400068

National Semiconductor

Mass Storage Handbook

A Corporate Dedication to Quality and Reliability

National Semiconductor is an industry leader in the manufacture of high quality, high reliability integrated circuits. We have been the leading proponent of driving down IC defects and extending product lifetimes. From raw material through product design, manufacturing and shipping, our quality and reliability is second to none.

We are proud of our success . . . it sets a standard for others to achieve. Yet, our quest for perfection is ongoing so that you, our customer, can continue to rely on National Semiconductor Corporation to produce high quality products for your design systems.

Charles E. Sporck
President, Chief Executive Officer
National Semiconductor Corporation

Wir fühlen uns zu Qualität und Zuverlässigkeit verpflichtet

National Semiconductor Corporation ist führend bei der Herstellung von integrierten Schaltungen hoher Qualität und hoher Zuverlässigkeit. National Semiconductor war schon immer Vorreiter, wenn es galt, die Zahl von IC Ausfällen zu verringern und die Lebensdauern von Produkten zu verbessern. Vom Rohmaterial über Entwurf und Herstellung bis zur Auslieferung, die Qualität und die Zuverlässigkeit der Produkte von National Semiconductor sind unübertroffen.

Wir sind stolz auf unseren Erfolg, der Standards setzt, die für andere erstrebenswert sind. Auch ihre Ansprüche steigen ständig. Sie als unser Kunde können sich auch weiterhin auf National Semiconductor verlassen.

La Qualité et La Fiabilité:

Une Vocation Commune Chez National Semiconductor Corporation

National Semiconductor Corporation est un des leaders industriels qui fabrique des circuits intégrés d'une très grande qualité et d'une fiabilité exceptionelle. National a été le premier à vouloir faire chuter le nombre de circuits intégrés défectueux et a augmenter la durée de vie des produits. Depuis les matières premières, en passant par la conception du produit sa fabrication et son expédition, partout la qualité et la fiabilité chez National sont sans équivalents.

Nous sommes fiers de notre succès et le standard ainsi défini devrait devenir l'objectif à atteindre par les autres sociétés. Et nous continuons à vouloir faire progresser notre recherche de la perfection; il en résulte que vous, qui êtes notre client, pouvez toujours faire confiance à National Semiconductor Corporation, en produisant des systèmes d'une très grande qualité standard.

Un Impegno Societario di Qualità e Affidabilità

National Semiconductor Corporation è un'industria al vertice nella costruzione di circuiti integrati di altà qualità ed affidabilità. National è stata il principale promotore per l'abbattimento della difettosità dei circuiti integrati e per l'allungamento della vita dei prodotti. Dal materiale grezzo attraverso tutte le fasi di progettazione, costruzione e spedizione, la qualità e affidabilità National non è seconda a nessuno.

Noi siamo orgogliosi del nostro successo che fissa per gli altri un traguardo da raggiungere. Il nostro desiderio di perfezione è d'altra parte illimitato e pertanto tu, nostro cliente, puoi continuare ad affidarti a National Semiconductor Corporation per la produzione dei tuoi sistemi con elevati livelli di qualità.

Charles E. Sporck
President, Chief Executive Officer
National Semiconductor Corporation

Mass Storage
HANDBOOK

1989 Edition

TRADEMARKS

Following is the most current list of National Semiconductor Corporation's trademarks and registered trademarks.

AbuseableTM
AnadigTM
ANS-R-TRANTM
APPSTM
ASPECTTM
Auto-Chem DeflasherTM
BCPTM
BI-FETTM
BI-FET IITM
BI-LINETM
BIPLANTM
BLCTM
BLXTM
Brite-LiteTM
BTLTM
CheckTrackTM
CIMTM
CIMBUSTM
CLASICTM
Clock✓ChekTM
COMBOTM
COMBO ITM
COMBO IITM
COPSTM microcontrollers
Datachecker®
DENSPAKTM
DIBTM
Digitalker®
DISCERNTM
DISTILLTM
DNR®
DPVMTM
ELSTARTM
Embedded System ProcessorTM
E-Z-LINKTM
FACTTM
FAIRCADTM
FairtechTM
FAST®
5-Star ServiceTM
GENIXTM
GNXTM
HAMRTM
HandiScanTM
HEX 3000TM
HPCTM
I3L®
ICMTM
INFOCHEXTM
Integral ISETM
IntelisplayTM
ISETM
ISE/06TM
ISE/08TM
ISE/16TM
ISE32TM
ISOPLANARTM
ISOPLANAR-ZTM
KeyScanTM
LMCMOSTM
M2CMOSTM
MacrobusTM
MacrocomponentTM
MAXI-ROM®
Meat✓ChekTM
MenuMasterTM
Microbus™ data bus
MICRO-DACTM
μtalkerTM
MicrotalkerTM
MICROWIRETM
MICROWIRE/PLUSTM
MOLETM
MSTTM
Naked-8TM
National®
National Semiconductor®
National Semiconductor Corp.®
NAX 800TM
Nitride PlusTM
Nitride Plus OxideTM
NMLTM
NOBUSTM
NSC800TM
NSCISETM
NSX-16TM
NS-XC-16TM
NTERCOMTM
NURAMTM
OXISSTM
P2CMOSTM
PC MasterTM
Perfect WatchTM
Pharma✓ChekTM
PLANTM
PLANARTM
Plus-2TM
PolycraftTM
POSilinkTM
POSitalkerTM
Power + ControlTM
POWERplanarTM
QUAD3000TM
QUIKLOOKTM
RATTM
RTX16TM
SABRTM
Script✓ChekTM
SCXTM
SERIES/800TM
Series 900TM
Series 3000TM
Series 32000®
Shelf✓ChekTM
SofChekTM
SPIRETM
STARTM
StarlinkTM
STARPLEXTM
Super-BlockTM
SuperChipTM
SuperScriptTM
SYS32TM
TapePak®
TDSTM
TeleGateTM
The National Anthem®
Time✓ChekTM
TINATM
TLCTM
TrapezoidalTM
TRI-CODETM
TRI-POLYTM
TRI-SAFETM
TRI-STATE®
TURBOTRANSCEIVERTM
VIPTM
VR32TM
WATCHDOGTM
XMOSTM
XPUTM
Z STARTM
883B/RETSTM
883S/RETSTM

IBM PC-AT® is a registered trademark of International Business Machines Corporation.

MULTIBUS® is a registered trademark of Intel Corporation.

PAL® is a registered trademark of and used under license from Monolithic Memories, Inc.

UNIX® is a registered trademark of AT&T Bell Laboratories.

DECTM, VAXTM and VMSTM are trademarks of Digital Equipment Corporation.

LIFE SUPPORT POLICY

NATIONAL'S PRODUCTS ARE NOT AUTHORIZED FOR USE AS CRITICAL COMPONENTS IN LIFE SUPPORT DEVICES OR SYSTEMS WITHOUT THE EXPRESS WRITTEN APPROVAL OF THE PRESIDENT OF NATIONAL SEMICONDUCTOR CORPORATION. As used herein:

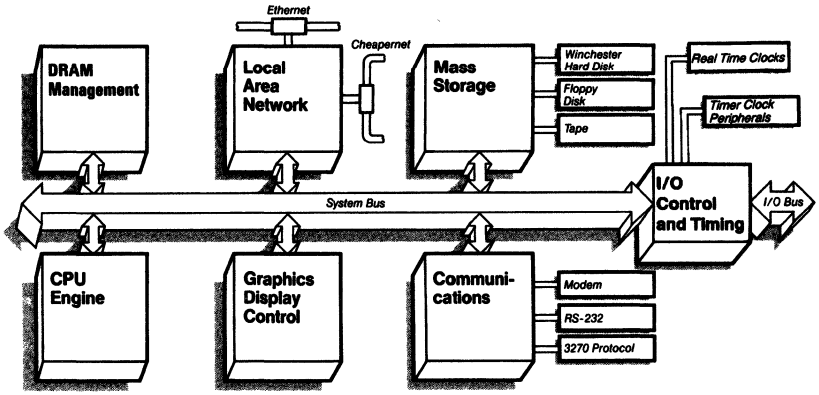
1. Life support devices or systems are devices or systems which, (a) are intended for surgical implant into the body, or (b) support or sustain life, and whose failure to perform, when properly used in accordance with instructions for use provided in the labeling, can be reasonably expected to result in a significant injury to the user.

2. A critical component is any component of a life support device or system whose failure to perform can be reasonably expected to cause the failure of the life support device or system, or to affect its safety or effectiveness.

National Semiconductor Corporation 2900 Semiconductor Drive, P.O. Box 58090, Santa Clara, California 95052-8090 (408) 721-5000
TWX (910) 339-9240

National Semiconductor

Introduction Advanced Peripherals



TL/XX/0058–1

National Semiconductor Advanced Peripherals products include complex VLSI peripheral circuits designed to serve a variety of applications. The Advanced Peripherals products are especially well suited for microcomputer and microprocessor systems such as graphics workstations, personal computers, and many others. National Semiconductor Advanced Peripherals devices are fully described in a series of databooks and handbooks.

Among the Advanced Peripherals books are the following titles:

MASS STORAGE

The National Semiconductor family of mass storage interface products offers the industry's highest performance and broadest range of products for Winchester hard disks and floppy disks. The Mass Storage Handbook includes complete product information and datasheets as well as a comprehensive design guide for disk controller systems.

DRAM MANAGEMENT

Today's large Dynamic Random Access Memory (DRAM) arrays require sophisticated high performance devices to provide timing access arbitration on board drive and control. National Semiconductor offers the broadest range of DRAM controllers with the highest "No-waitstate" performance available on the market. Controllers are available in Junction Isolated LS, Oxide Isolated ALS, and double metal CMOS for DRAMs from 64k bit through 4M bit devices, supporting memory arrays up to 64 Mbyte in size with only one LSI/VLSI device. For critical applications, National Semiconductor has developed several 16- and 32-bit Error Checking and Correction (ECC) devices to provide maximum data integrity. The Memory Support Handbook contains complete product information and several application notes detailing complete memory system design.

LOCAL AREA NETWORKS AND DATA COMMUNICATIONS

Today's computer systems have created a huge demand for data communications and Local Area Networks (LANs). National Semiconductor provides a complete three-chip solution for an entire IEEE 802.3 standard for Ethernet/Cheapernet LANs. National Semiconductor offers a completely integrated solution for the IBM 370 class mainframes, System 3X and AS/400 systems for physical layer front end and processing of the IBM 3270/3299 "coaxial" and 5250 "twinaxial" protocols. To drive the communications lines, National Semiconductor has drivers and receivers designed to meet all the major standards such as RS-232, RS-422, and RS-485. Datasheets and applications information for all these products are in the LAN/DATA COMM Handbook.

GRAPHICS

Sophisticated human interface is a mark of the newest computer systems designs. Today's personal computer may have better graphics display capability than engineering workstations of a few years ago. National Semiconductor has developed a new family of Advanced Graphics products to provide extremely high performance, high resolution color graphics displays. The graphics chip set is designed to provide the highest level of performance with minimum demands and loading on the system CPU. The graphics system may be expanded to any number of color planes with virtually unlimited resolution. The Graphics Databook lays it all out and makes the display system design easy.

REAL TIME CLOCKS

National offers a family of Real Time Clocks (RTCs) and advanced Timer Clock Peripherals (TCPs). The RTC family provides a simple μP bus compatible interface to any system requiring accurate, reliable, on-going real time and calender functions. The TCP family offers the RTC, RAM and two 16-bit programmable timers with fast μP bus handshake controls for chip select, read and write. The Real Time Clock handbook includes complete product information and datasheets as well as applications information.

National Semiconductor

Product Status Definitions

Definition of Terms

Data Sheet Identification	Product Status	Definition
Advance Information	Formative or In Design	This data sheet contains the design specifications for product development. Specifications may change in any manner without notice.
Preliminary	First Production	This data sheet contains preliminary data, and supplementary data will be published at a later date. National Semiconductor Corporation reserves the right to make changes at any time without notice in order to improve design and supply the best possible product.
No Identification Noted	Full Production	This data sheet contains final specifications. National Semiconductor Corporation reserves the right to make changes at any time without notice in order to improve design and supply the best possible product.

National Semiconductor Corporation reserves the right to make changes without further notice to any products herein to improve reliability, function or design. National does not assume any liability arising out of the application or use of any product or circuit described herein; neither does it convey any license under its patent rights, nor the rights of others.

Table of Contents

Table of Contents (Continued)

Alpha-Numeric Index

Alpha-Numeric Index (Continued)

Section 1

Rigid Disk Pulse Detectors

Section 1 Contents

National Semiconductor

DP8464B Disk Pulse Detector

General Description

The DP8464B Disk Pulse Detector utilizes analog and digital circuitry to detect amplitude peaks of the signal received from the read/write amplifier fitted with the heads of disk drives. The DP8464B produces a TTL compatible output which, on the positive leading edge, indicates a signal peak. Electrically, these peaks correspond to flux reversals on the magnetic medium. The signal from the read/write amplifier when reading a disk is therefore a series of pulses with alternating polarity. The Disk Pulse Detector accurately replicates the time position of these peaks.

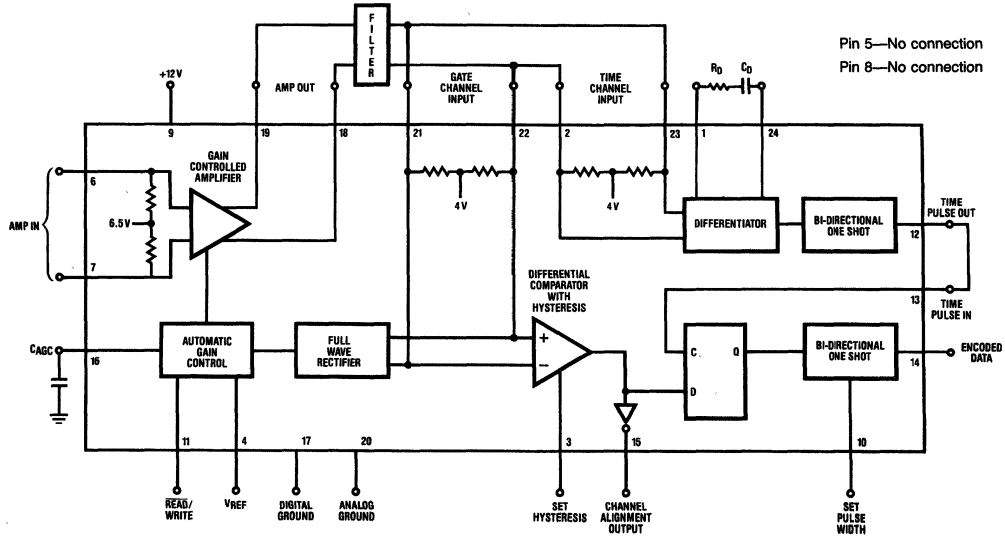
The DP8464B Disk Pulse Detector has three main sections: the Amplifier, the time channel and the gate channel. The Amplifier section consists of a wide bandwidth amplifier, a full wave rectifier and Automatic Gain Control (AGC). The time channel is made from the differentiator and its following bi-directional one shot, while the gate channel is made from the differential comparator with hysteresis, the D flip-flop and its following bi-directional one shot.

The Disk Pulse Detector is fabricated using an advanced oxide isolated Schottky process, and has been designed to function with data rates up to 15 Megabits/second. The DP8464B is available in either a 300 mil wide 24-pin dual-in-line package or a surface mount 28-pin plastic chip carrier package. Normally, it will be fitted in the disk drive, and its output may be directly connected to the DP8461 or the DP8465 Data Separator.

Features

- Wide input signal amplitude range—from 20 mVpp to 660 mVpp differential
- Data rates up to 15 Megabits/sec 2,7 code
- On-chip differential gain controlled amplifier, differentiator, comparator gating circuitry, and output pulse generator
- Input capacitively coupled directly from the disk head read/write amplifier
- Adjustable comparator hysteresis
- AGC and differentiator time constants set by external components
- TTL compatible digital Inputs and Outputs
- Encoded Data Output may connect directly to the DP8461 or DP8465 Data Separator
- Standard drive supply: 12V ± 10%
- Available in 300 mil wide 24-pin dual-in-line package, a surface mount 28-pin plastic chip carrier package, or a 40-pin TapePak® package

Block Diagram



TL/F/5283–7

Note: All pin numbers in this data sheet refer to the 24-pin dual-in-line package.

Absolute Maximum Ratings

If Military/Aerospace specified devices are required, please contact the National Semiconductor Sales Office/Distributors for availability and specifications.

	Pins	Limit
Supply Voltage	9	14V
TTL Input Voltage	11,13	5.5V
TTL Output Voltage	12,14,15	5.5V
Input Voltage	3,4	5.5V
Minimum Input Voltage	3,4	−0.5V
Differential Input Voltage	6-7, 21-22, 2-23	3V or −3V
ESD Susceptibility (see Note 5)		

Storage Temperature	−65°C to +150°C
Lead Temp. (Soldering, 10 seconds)	300°C
Maximum Power Dissipation at 25°C	
Molded DIP Package (derate 15.6 mW/°C above 25°C)	1950 mW
Plastic Chip Carrier Package (derate 12.5 mW/°C above 25°C)	1560 mW

Operating Conditions

Symbol	Parameter	Min	Typ	Max	Units
V_{CC}	Supply Voltage	10.8	12.0	13.2	V
T_A	Ambient Temperature	0		70	°C

DC Electrical Characteristics

Over Recommended Operating Temperature and Supply Range V_{REF} = 0.5V, Set Hysteresis = 0.3V. $\overline{\text{Read}}$/Write = 0.3V unless otherwise noted. All Pin Numbers Refer to 24 Pin Dual-In-Line Package.

Symbol	Pins	Parameter	Conditions	Min	Typ	Max	Units
AMPLIFIER							
$Z_{IN_{AI}}$	6,7	Amp In Impedance	T_A = 25°C (Note 1)	0.75	1.0	1.25	kΩ
A_{VMIN}	18,19	Min Voltage Gain	AC Output 4 Vpp Differential			6.0	V/V
A_{VMAX}	18,19	Max Voltage Gain	AC Output 4 Vpp Differential	200			V/V
$V_{C_{AGC}}$	16	Voltage on C_{AGC}	A_V = 6.0 A_V = 200	 2.8	4.5 3.7	5.5	V V
GATE CHANNEL							
$Z_{IN_{GCI}}$	21,22	Gate Channel Input Impedance	T_A = 25°C (Note 1)	1.75	2.5	3.25	kΩ
$I_{C_{AGC}-}$	16	Pin 16 Current which Charges C_{AGC}	$V_{PIN\ 16}$ = 3.9V $\lvert V_{PIN\ 21} - V_{PIN\ 22} \rvert$ = 1.3 V_{DC}	−1.5	−2.5	−3.5	mA
$I_{C_{AGC}+}$	16	Pin 16 Current which Discharges C_{AGC}	$V_{PIN\ 16}$ = 5V $\lvert V_{PIN\ 21} - V_{PIN\ 22} \rvert$ = 0.7 V_{DC}		1	5	μA
$I_{V_{REF}}$	4	V_{REF} Input Bias Current			−20	−100	μA
$V_{TH_{AGC}}$	22,21 4,16	AGC Threshold	(Note 2) $V_{PIN\ 16}$ = 4.2V	0.88	1.0	1.12	V
I_{SH}	3	Set Hysteresis Input Bias Current			−60	−100	μA
$V_{TH_{SH}}$	22,21 3,15	Set Hysteresis Threshold	(Note 3)	0.48	0.6	0.72	V
TIME CHANNEL							
$Z_{IN_{TC}}$	2,23	Time Channel Input Impedance	T_A = 25°C (Note 1)	3.5	5.0	6.5	kΩ
I_{C_d}	24	Current into Pin 1 and 24 that Discharges C_d		1.4	1.8	2.50	mA

DC Electrical Characteristics Over Recommended Operating Temperature and Supply Range V_{REF} = 0.5V, Set Hysteresis = 0.3V. $\overline{Read}$/Write = 0.3V unless otherwise noted. All pin numbers refer to the 24 pin dual-in-line package. (Continued)

Symbol	Pins	Parameter	Conditions	Min	Typ	Max	Units
WRITE MODE							
$Z_{IN_{AI}}$	6,7	Amp In Impedance in Write Mode	$V_{PIN\ 11}$ = 2.0V	50		250	Ω
$I_{C_{AGC-}}$	16	Pin 16 Current in Write Mode	$V_{PIN\ 11}$ = 2.0V $V_{PIN\ 16}$ = 3.9V $\lvert V_{PIN\ 21} - V_{PIN\ 22} \rvert$ = 1.3 V_{DC}		1	5	μA
DIGITAL PINS							
V_{IH}	11,13	High Level Input Voltage		2			V
V_{IL}	11,13	Low Level Input Voltage				0.8	V
V_I	11,13	Input Clamp Voltage	V_{CC} = Min I_I = −18 mA			−1.5	V
I_{IH}	11,13	High Level Input Current	V_{CC} = Max V_I = 2.7V			20	μA
I_I	11,13	Input Current at Maximum Input Voltage	V_{CC} = Max V_I = 5.5V			1	mA
I_{IL}	11,13	Low Level Input Current	V_{CC} = Max V_I = 0.5V			−200	μA
V_{OH}	12,14, 15	High Level Output Voltage	V_{CC} = Min I_{OH} = −40 μA (Note 4)	2.7			V
V_{OL}	12,14, 15	Low Level Output Voltage	V_{CC} = Min I_{OL} = 800 μA (Note 4)			0.5	V
I_{OS}	12,14, 15	Output Short Circuit Current	V_{CC} = Max V_O = OV			−100	mA
I_{CC}	9	Supply Current	V_{CC} = Max		54	75	mA

AC Electrical Characteristics

Over Recommended Operating Temperature and Supply Range unless otherwise noted

Symbol	Pins	Parameter	Conditions	Typ	Max	Units
DP8464B-2 t_{pp}	14	Pulse Pairing	(See Pulse Pairing Set Up)	±1.5	±3	ns
DP8464B-3 t_{pp}	14	Pulse Pairing	(See Pulse Pairing Set Up)	±2	±5	ns
DP8464B-1 t_{pp}	14	Pulse Pairing	(See Pulse Pairing Set Up) at 25°C V_{CC} = 12V only	±0.5	±1	ns

Note 1: The temperature coefficient of the input impedance is typically 0.05% per degree C.

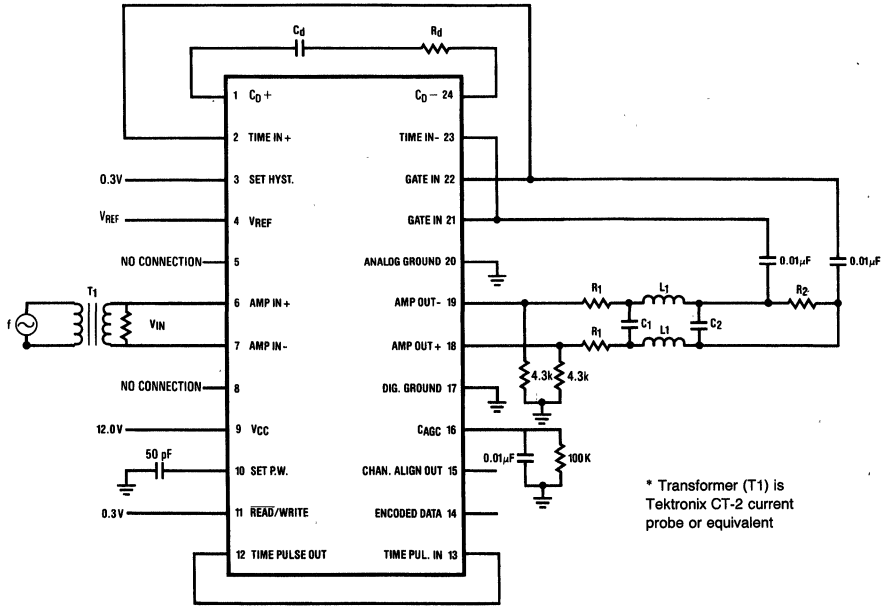
Note 2: The AGC Threshold is defined as the voltage across the Gate Channel Input (pins 21 and 22) when the voltage on C_{AGC} (pin 16) is 4.2V.

Note 3: The Set Hysteresis Threshold is defined as the minimum differential AC signal across the Gate Channel Input (pins 21 and 22) which causes the voltage on the Channel Alignment Output (pin 15) to change state.

Note 4: To prevent inductive coupling from the digital outputs to Amp In, the TTL outputs should not drive more than one ALS TTL load each.

Note 5: The following pins did not meet the 2000V ESD test with the human body model, 120 pF thru 1.5 kΩ: Pins 1, 2, 3, 10, 11, 12, 14, 21, 24.

Pulse Pairing Set Up



TL/F/5283–3

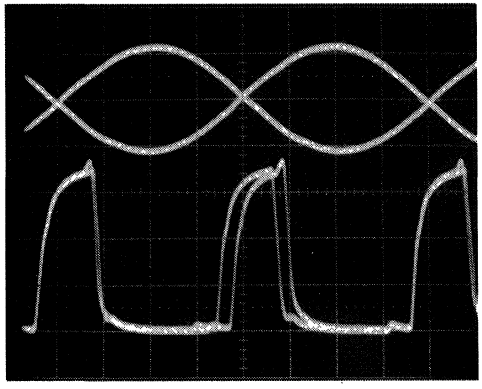
DP8464B

f = 2.5 MHz
V_{IN} = 40 mV_{pp} differential
V_{REF} = 0.50V
C_D = 50 pF
R_D = 430Ω

Filter

R1 = 240Ω R2 = 680Ω
C1 = 15 pF C2 = 100 pF
L1 = 4.7 μH

This is a 3 pole Bessel with the corner frequency at 7.5 MHz.



TL/F/5283–4

Pulse Pairing Measurement

Connect a scope probe to pin 14 (Encoded Data Out) and trigger off its positive edge. Adjust the trigger holdoff so the scope first triggers off the pulse associated with the positive peak and then off the pulse associated with the negative peak (as shown in the scope photo below). Pulse pairing is displayed on the second pair of pulses on the display. If the second pulses are separated by 4 ns, then the pulse pairing for this part is ±2 ns.

Circuit Operation

The output from the read/write amplifier is AC coupled to the Amp Input of the DP8464B. The amplifier's output voltage is fed back via an external filter to an internal fullwave rectifier and compared against the external voltage on the V_{REF} pin. The AGC circuit adjusts the gain of the amplifier to make the peak to peak differential voltage on the Gate Channel Input four times the DC voltage on V_{REF}. Typically the signal on Amp Out will be set for 4 Vpp differential. Since the filter usually has a 6 dB loss, the signal on the Gate Channel Input will be 2 Vpp differential. The user should therefore set 0.5V on V_{REF} which can be done with a simple voltage divider from the +12V supply.

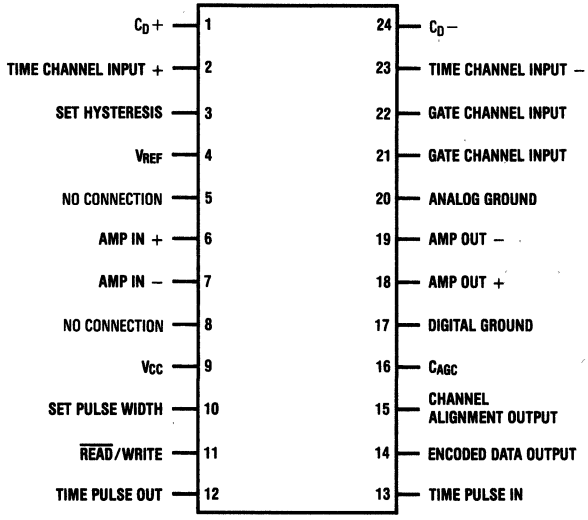
The peak detection is performed by feeding the output of the Amplifier through an external filter to the differentiator. The differentiator output changes state when the input pulse changes direction, generally this will be at the peaks. However, if the signal exhibits shouldering (the tendency to return to the baseline), the differentiator will also respond to noise near the baseline. To avoid this problem, the signal is also fed to a gating channel which is used to define a level either side of the baseline. This gating channel is comprised

Circuit Operation (Continued)

of a differential comparator with hysteresis and a D flip-flop. The hysteresis for this comparator is externally set via the Set Hysteresis pin. In order to have data out, the input amplitude must first cross the hysteresis level which will change the logic level on the D input of the flip-flop. The peak of the input signal will generate a pulse out of the differentiator and bi-directional one shot. This pulse will clock the new data at the D input through to the output. In this way, when the differentiator is responding to noise at the baseline, the output of the D flop is not changing since the logic level into the D input has not changed. The comparator circuitry is therefore a gating channel which prevents any noise near the baseline from contaminating the data. The amount of hysteresis is twice the DC voltage on the Set Hysteresis pin. For instance, if the voltage on the Set Hysteresis pin is 0.3V, the differential AC signal across the Gate Channel Input must be larger than 0.6V before the output of the comparator will change states. In this case, the hysteresis is 30% of a 2V peak to peak differential signal at the gate channel input.

Connection Diagrams

Dual-In-Line (DIP) Package

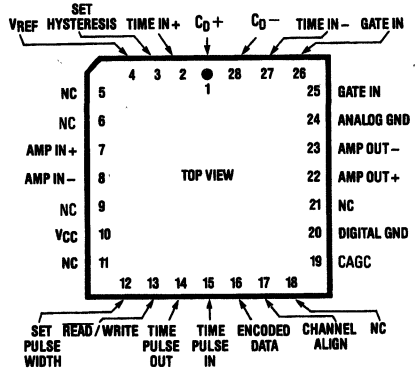


Top View

TL/F/5283–2

Order Number DP8464BN-3 or DP8464BN-2
See NS Package N24C

Plastic Chip Carrier (PCC) Package



TL/F/5283–30

Order Number DP8464BV-3, DP8464BV-2 or DP8464BV-1
See NS Package V28A

Pin Definitions

(All pin numbers refer to the 24 pin dual-in-line package)

Pin#	Name	Function
Power Supply		
9	V_{CC}	The supply is +12V±10%.
17	Digital Ground	Digital signals should be referenced to this pin.
20	Analog Ground	Analog signals should be referenced to this pin.
Analog Signals		
6	Amp In+	These are the differential inputs to the Amplifier. The output of the read/write head amplifier should be capacitively coupled to these pins.
7	Amp In−	
18	Amp Out+	These are the differential outputs of the Amplifier. These outputs should be capacitively coupled to the gating channel filter (if required) and to the time channel filter.
19	Amp Out−	
22	Gate	These are the differential inputs to the AGC block and the gating channel. These inputs must be capacitively coupled from the Amp Out.
21	Channel Inputs	
2	Time Channel Input+	These are the differential inputs to the differentiator in the time channel. In most applications, a filter between the Amp Out (pins 18 and 19) and these inputs is required to band limit the noise and to correct for any phase distortion introduced by the read circuitry. In all cases this input must be capacitively coupled to prevent disturbing the DC input level.
23	Time Channel Input−	
1	C_d+	The external differentiator network is connected between these two pins.
24	C_d−	
3	Set Hysteresis	The DC voltage on this pin sets the amount of hysteresis on the differential comparator. Typically this voltage can be established by a simple resistive divider from the positive supply.
4	V_{REF}	The AGC circuit adjusts the gain of the amplifier to make the differential peak to peak voltage on the Gate Channel Input equal to four times the DC voltage on this pin. This voltage can be established by a simple resistive divider from the positive supply.
5	No connection	
8	No connection	
16	C_{AGC}	The external capacitor for the AGC is connected between this pin and Analog Ground.

Pin#	Name	Function
Digital Signals		
10	Set Pulse Width	An external capacitor to control the pulse width of the Encoded Data Out is connected between this pin and Digital Ground.
11	$\overline{\text{Read}}$/Write	If this pin is low, the Pulse Detector is in the read mode and the chip is active. When this pin goes high, the pulse detector is forced into a standby mode. This is a standard TTL input.
12	Time Pulse Out	This is the TTL output from the bi-directional one shot following the differentiator. In most applications this can be connected directly to the Time Pulse In.
13	Time Pulse In	This is the TTL input to the clock of the D flip-flop. Usually this is connected directly to the Time Pulse Out pin.
15	Channel Alignment	This is the buffered output of the differential comparator with hysteresis. This is usually used in the initial system design and is not used in production.
14	Encoded Data Out	This is the standard TTL output whose leading edge, indicates the time position of the peaks.

Application Information

GENERAL DESCRIPTION

All pin numbers refer to 24 pin dual-in-line package.

The DP8464B Disk Pulse Detector utilizes analog and digital circuitry to detect amplitude peaks of the signal received from the Read/Write Amplifier. The analog signal from a disk is a series of pulses, the peaks of which correspond to 1's or flux reversals on the magnetic medium. The pulse detector must accurately determine the time position of these peaks. The peaks are indicated by the positive leading edge of a TTL compatible output pulse. This task is complicated by variable pulse amplitudes depending on the media type, head position, head type and read/write amplifier circuit gain. Additionally, as the bit density on the disk increases, the amplitude decreases and significant bit interaction occurs resulting in pulse distortion and shifting of the peaks.

The graph in *Figure 1* shows how the pulse amplitude varies with the number of flux reversals per inch (or recording density) for a given head disk system. The predominant disk applications are associated with the first two regions on this graph, Regions 1 and 2. Typical waveforms received by the pulse detector for these regions are shown next to the graph.

Application Information (Continued)

Region 1 is the high resolution area characterized by a large spread between flux reversals and a definite return to baseline (no signal) between these peaks. Pulses of this type are predominantly found in drives which use either thin film heads or plated media, or in drives which utilize run length limited codes (like the 2,7 code) which spread the distance between flux reversals.

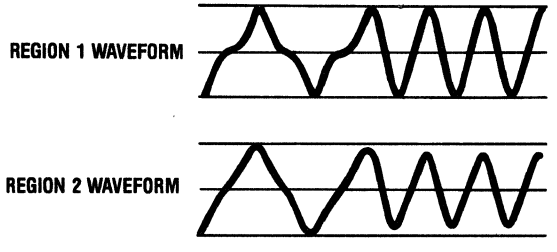
A Region 2 waveform will vary from a tendency to return to the baseline (called shouldering) to almost sinusoidal at the higher frequencies. These pulses come from drives which use limited frequency codes (such as MFM). The pulses may contain shouldering on the outer tracks of the disk and be nearly sinusoidal on the inner tracks since the flux density increases towards the inner track.

Detecting pulse peaks of waveforms of such variable characteristics requires a means of separating both noise and shouldering-caused errors from the true peaks. In the past, mild shoulder-caused errors were blocked by self-gating circuits (such as the "de-snaker"). These circuits fail when shouldering is extensive, hence the need for the DP8464B which includes a peak sensing circuit and an amplitude sensitive gating channel in parallel.

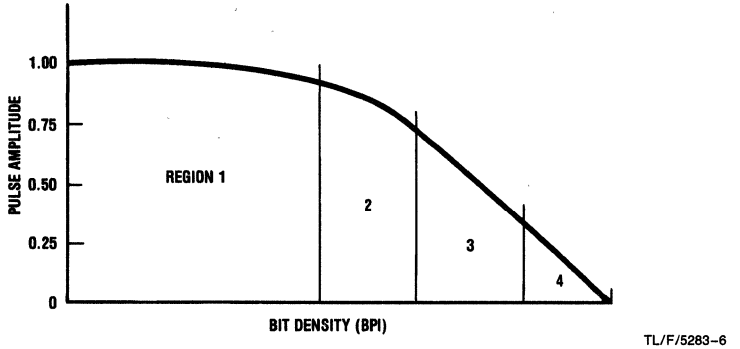
The main circuit blocks of the DP8464B are shown in *Figure 2*. The output from the read/write amplifier is fed directly to the Amp Input of the DP8464B. This is the input of a Gain Controlled Amplifier. The amplifier's output voltage is fed back via an external filter to an internal fullwave rectifier and compared against the external voltage on the V_{REF} pin. The AGC circuit adjusts the gain of the amplifier to make the peak-to-peak differential Gate Channel input voltage four times the DC voltage on V_{REF}.

The peak detection is performed by feeding the output of the Gain Controlled Amplifier through an external filter to the differentiator. The differentiator output changes state when the input pulse changes direction, generally this will be at the peaks. However, if the signal exhibits shouldering (the tendency to return to the baseline) as seen in Region 1 and the upper part of Region 2, the differentiator will also respond to noise near the baseline. To avoid this, the signal is also fed to a gating channel which is used to define a level either side of the baseline. This gating channel comprises a differential comparator with hysteresis and a D flip-flop. The hysteresis for this comparator is externally set via the Set Hysteresis pin. In order to have valid data out, the input amplitude must first cross the hysteresis level. This will change the logic level on the D input of the flip-flop. The peak of the input signal will generate a pulse out of the differentiator and bi-directional one shot. This pulse will clock in the new data on the D input, which will appear at the Q output. In this way, when the differentiator is responding to noise at the baseline, the output of the D flop is not changing since the logic level into the D input has not yet changed. The comparator circuitry is therefore a gating channel to prevent any noise near the baseline from contaminating the data.

The amount of hysteresis is twice the DC voltage on the Set Hysteresis pin. For instance, if the voltage on the Set Hysteresis pin is 0.3V, the differential Gate Channel Input must be larger than 0.6V (±0.3V) before the output of the comparator will change states. The Time Pulse Out, Encoded Data, and Channel Alignment Output are designed to drive 1 standard TTL gate.



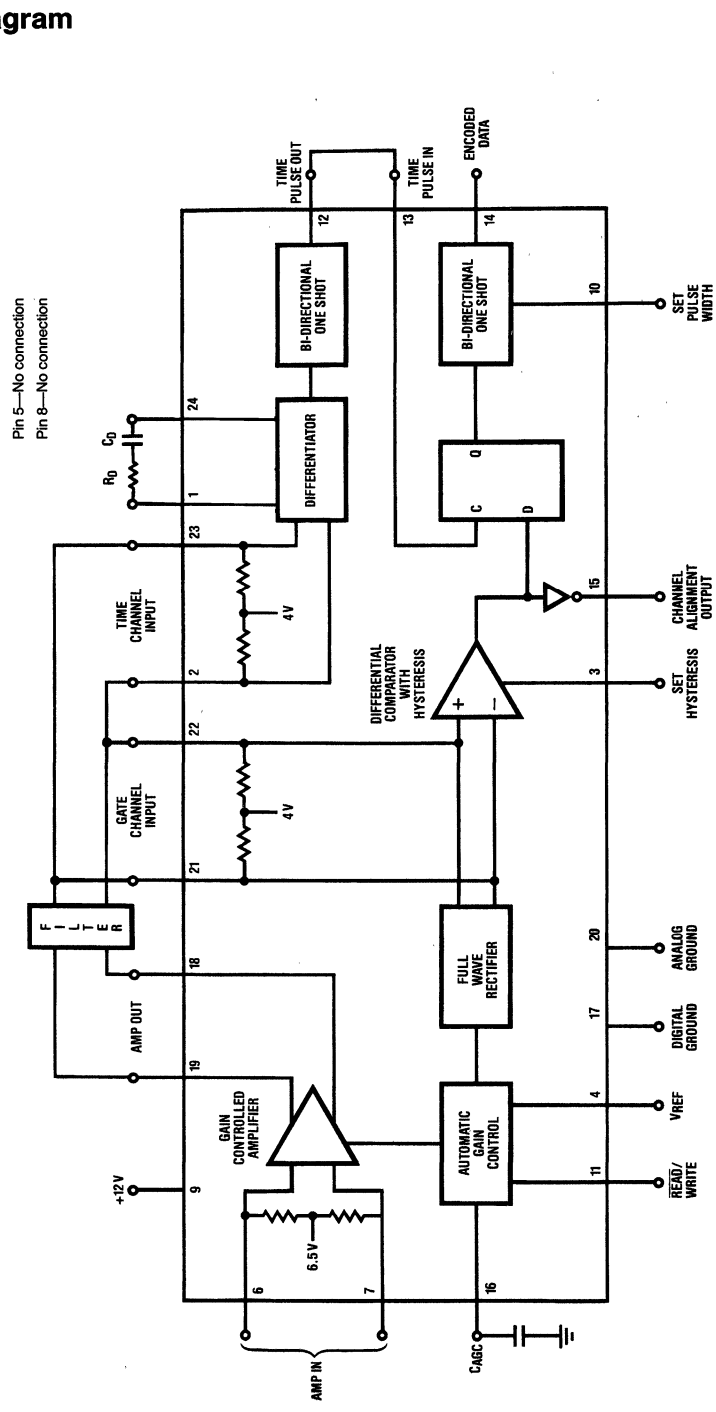
TL/F/5283–5



TL/F/5283–6

FIGURE 1. Pulse Amplitude vs. Bit Density with Typical Waveforms

Block Diagram



TL/F/5283–7

FIGURE 2. DP8464B Block Diagram, Region 1 Connection

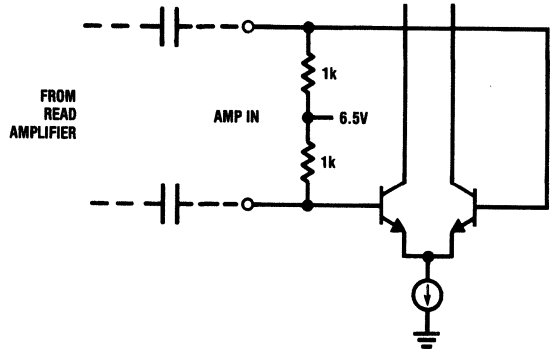
Application Information (Continued)

GAIN CONTROLLED AMPLIFIER

The purpose of the Gain Controlled Amplifier is to increase the differential input signal to a fixed amplitude while maintaining the exact shape of the input waveform. The Gain Controlled Amplifier is designed to accept input signals from 20 mVpp to 660 mVpp differential and amplify that signal to 4 Vpp differential. The gain is therefore from 6 to 200 and is controlled by the automatic gain control (AGC) loop. The amplifier output is actually capable of delivering typically 5 Vpp differential output but the parts are only tested and guaranteed to 4 Vpp.

The input to the Gain Controlled Amplifier is shown in *Figure 3*. The value of the input capacitors should be selected so that the pole formed by the coupling capacitor and the 1k bias resistor is a factor of 10 lower than the lowest signal frequency. These input bias resistors have a ±20% tolerance and a temperature coefficient of 0.05% per degree C. When the pulse detector is in the write mode, these bias resistors are automatically shunted by 425Ω resistors. This allows the input circuit to recover quickly from the large transients encountered during a write to read transition. The input impedance to the amplifier is therefore 1k during read operations and 300Ω during write operations.

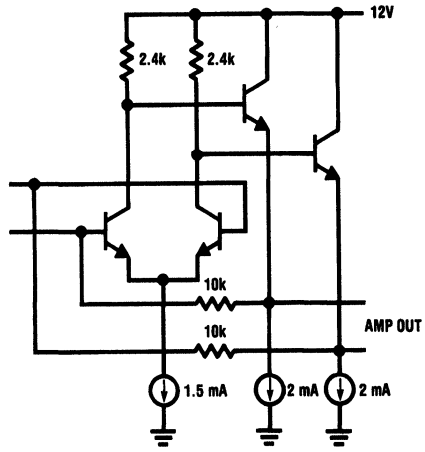
The output of the Gain Controlled Amplifier is shown in *Figure 4*. The outputs are biased at (12V − (0.75 mA × 2.4k) − 0.75V) or 9.5V. Since each output will swing ±1V (4 Vpp differential), each output pin will swing from 8.5V to 10.5V. If the total differential load placed on the output is 1k, (see *Figure 5*) then the circuit must supply 2V/1k or 2 mA. Since the output is class A, external resistors to ground must be used to provide the sink current. In this case, in order to sink 2 mA at the lowest voltage, then (8.5V/2 mA) or an external 4.3k resistor from each output to ground is required. Note that the circuit has additional margin since the internal 2 mA current sources were not included in the calculation. Typically the output impedance of the Gain Controlled Amplifier is 17Ω, and the −3 dB bandwidth is greater than 20 MHz.



TL/F/5263–8

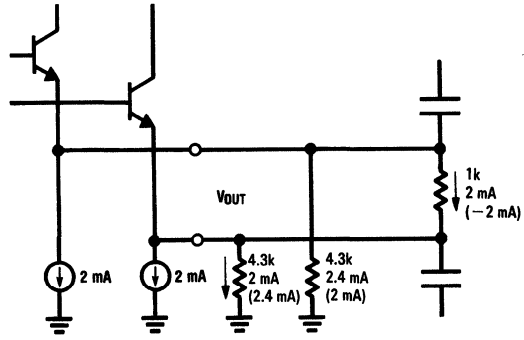
FIGURE 3. Input to Gain Controlled Amplifier

Application Information (Continued)



TL/F/5283–9

FIGURE 4. Output of Gain Controlled Amplifier



TL/F/5283–10

FIGURE 5. Output Stage with 1k Differential Load

Application Information (Continued)

AUTOMATIC GAIN CONTROL (AGC)

The Automatic Gain Control holds the signal level at the Gate Channel Input at a constant level by controlling the gain of the Gain Controlled Amplifier. This is necessary because the amplitude of the input signal will vary with track location, variations in the magnetic film, and differences in the actual recording amplitude. The Gain Controlled Amplifier is designed for a maximum 4 Vpp differential output. To prevent the Gain Controlled Amplifier from saturating, the V_{REF} level must be set so the maximum amplifier output voltage is 4 Vpp. The AGC will force the differential peak-to-peak signal on the Gate Channel Input to be four times the voltage applied to the V_{REF} pin. Normally some kind of filter is connected between the Gain Controlled Amplifier's output and the Gate Channel Input. Typically this filter has a 6 dB insertion loss in its pass band. Since the AGC holds the amplitude at the Gate Channel Input constant, this 6 dB loss through the Gate Channel filter will cause the Gain Controlled Amplifier's output to be 6 dB larger than the Gate Channel Input.

The AGC loop starts out in the high gain mode. When the input signal is larger than expected, the AGC loop will quickly reduce the amplifier gain so the peak-to-peak differential voltage on the Gate Channel Input remains four times the voltage on V_{REF}. If the input amplitude suddenly drops, the AGC loop will slowly increase the amplifier gain until the differential peak-to-peak Gate Channel Input voltage again reaches four times V_{REF}. The AGC loop requires several peaks to react to an increased input signal. In order to recover the exact peak timing during this transition, the V_{OUT} level must be set somewhat lower than the maximum of 4 Vpp. For instance, if the V_{REF} is 0.5V, and if the loss in the gate channel filter is 6 dB, then the Amp Output is 4 Vpp. If the Amp Input suddenly increases 30%, the amplifier may saturate and the timing for a few peaks may be disturbed until the AGC reduces the amplifier gain. If the peak detection is critical during this time, the system may fail. The proper operation, for this example, is to set the V_{REF} at 0.35V so the amplifier will not saturate if the input suddenly increases 30%.

A simplified circuit of the AGC block is shown in *Figure 6*. When the full wave rectified signal from the Gate Channel Input is greater than V_{REF}, the voltage on the collector of transistor T1 will increase and charge up the external capacitor C_{AGC} through T2. The typical available charging current is 2.5 mA. Conversely, if this input is less than V_{REF}, transistor T2 will be off, so the capacitor C_{AGC} will be discharged by the base current going into the Darlington T3 and T4. This discharge current is approximately 1 μA. The voltage across C_{AGC} controls the gain of the Gain Controlled Amplifier. This voltage will vary from typically 3.4V at the highest gain to 4.5V at the lowest gain.

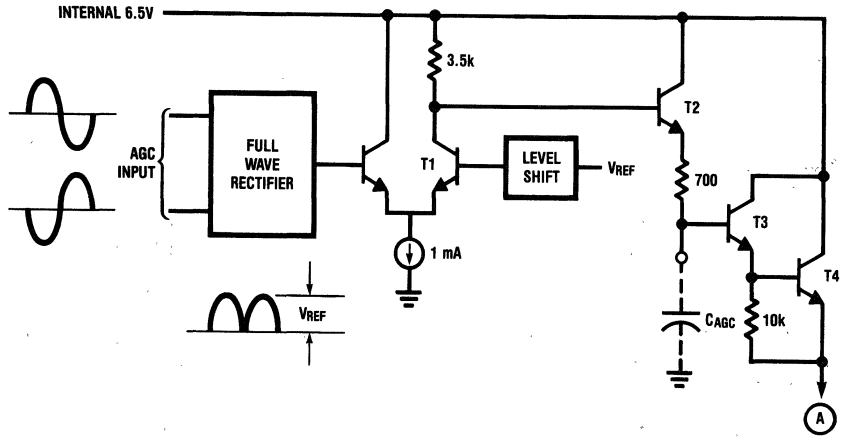
When the AGC circuit has not received an input signal for a long time, the base current of the Darlington will discharge the external C_{AGC} to 3.4V. The amplifier will now be at its highest gain. When a large signal comes in, the external C_{AGC} will be charged up with the 2.4 mA from T2 thereby reducing the gain of the amplifier. The formula, $I = C \times (dV/dt)$ can be used to calculate the time required for the amplifier to go from a gain of 200 to a gain of 6. For instance, if $C_{AGC} = 0.01$ μf, the charging current I is 2.4 mA, and the dV required for the amplifier to go through its gain range is 1.1V, then

$$dt = (0.01\ \mu F \times 1.1V)/(2.4\ mA) \text{ or } 4.6\ \mu s.$$

In reality, the gain does not change this quickly since the C_{AGC} would only be charging during a portion of the input waveform.

By using the same argument, the time required to increase the amplifier gain after the input has been suddenly reduced can be calculated. This time, the discharging current is only 1 μa so

$$dt = (0.01\ \mu F \times 1.1V)/1\ \mu A) \text{ or } 11\ ms.$$



TL/F/5283–11

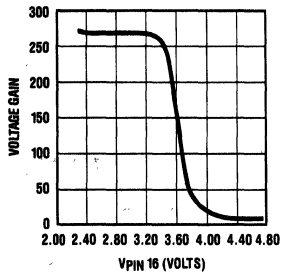
FIGURE 6. Simplified AGC Circuit

Application Information (Continued)

This time can be decreased by placing an external resistor across the C_{AGC}. For instance, if a 100k resistor is placed in parallel with C_{AGC}, then the discharge current is 40 μA. The time required to increase the amplifier gain is now 40 times faster or 275 μs. If this external resistor is made even smaller, say 10k, then the discharge time will go to 27.5 μs. Now however, there is another problem introduced. The response time of the AGC is so fast that it distorts the signal at the output of the Gain Controlled Amplifier. Distortion of the signal at the Amplifier Output can affect the time position of the peaks of this signal. Be sure to check this distortion over the range of input levels you expect to encounter, when choosing the external R and C values for the AGC.

If the value of the bleed resistor across the C_{AGC} is decreased (in order to equalize the AGC attack and decay times) the value of C_{AGC} must be increased in order to maintain an AGC response that does not distort the signal. There is a second order effect on the amplitude that results from this attack and decay time equalization. Referring to *Figure 2*, notice that the AGC is driven from a full wave rectified version of the Gate Channel Input signal. When the AGC is operated normally (ie. fast attack and slow decay) the voltage that appears across C_{AGC} is the peak detected value of this full wave rectified waveform. However, if you equalize the AGC attack and decay times the voltage across C_{AGC} is the RMS voltage (0.707 times the peak) of the full wave rectified waveform. Thus, the voltage across C_{AGC} is less and the amplitude out of the Gain Controlled Amplifier will consequently be 1.4 times larger.

It is possible to externally drive the C_{AGC} pin to control the gain of the amplifier. It must be noted that the gain of the amplifier is not always exactly 200 when the voltage on C_{AGC} is 3.4V. The transfer curve between the gain of the amplifier and the voltage on C_{AGC} is only approximate. This transfer curve will vary between parts and with temperature. Care should be taken to prevent the voltage on the C_{AGC} pin from going below ground or above 5.5V. *Figure 7* shows a typical curve of the Gain Controlled Amplifier Gain vs. the voltage across C_{AGC} (Vpin 16.)

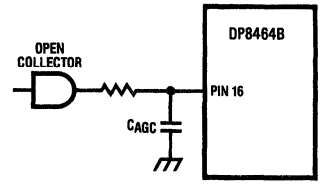


TL/F/5283–12

FIGURE 7. Gain Controlled Amplifier Gain vs. Vpin 16

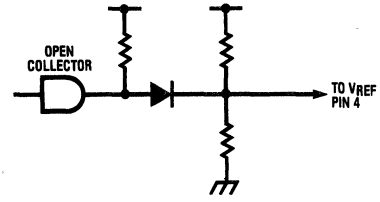
It is possible to change the time constant of the AGC circuit by switching in different external components at the desired times. For instance, as shown in *Figure 8*, an external open collector TTL gate and resistor can be added in parallel with C_{AGC} to decrease the AGC response time. Similarly, an external capacitor could be switched in to increase the response time. Since in the absence of an external resistor the discharge time of C_{AGC} is much longer than the attack time there may be some applications where it is desirable to switch in a parallel resistor to quickly discharge C_{AGC} then switch it out to force a quick attack. Because of the quick attack time, the AGC obtains the proper level quicker than it would had C_{AGC} simply been allowed to discharge to the new level.

There are some applications where it is desirable to hold the AGC level for a period of time. This can be done by raising the $\overline{\text{READ}}$/WRITE pin. This will shut off the input circuitry, and it will take time (about 2.5 μs) for the circuit to recover when going back into the read mode. *Figure 9* shows a method to hold the AGC level while remaining in the read mode (which could be used in embedded servo applications). If the voltage on V_{REF} is raised to 3V, then the amplifier output voltage cannot get large enough to turn on the circuitry to charge up C_{AGC}. For this to work properly, there can not be a large discharge current path (resistor in parallel with C_{AGC}) across C_{AGC}. The AGC block can be bypassed altogether by connecting V_{REF} to 3V. In this way, the user can use his own AGC circuit to drive the C_{AGC} pin directly.



TL/F/5283–13

FIGURE 8. Circuit to Decrease AGC Response Time



TL/F/5283–14

FIGURE 9. Circuit for AGC Hold

$\overline{\text{READ}}$/WRITE

In the normal read mode, the signal from the read/write head amplifier is in the range of 20 mVpp to 660 mVpp. However, when data is being written to the disk, the signal coming into the analog input of the pulse detector will be on the order of 600 mV. Such a large signal will disturb the AGC level and would probably saturate the amplifier. In addition, if a different read/write amplifier is selected, there will be a transient introduced because the offset of the preamplifiers are not matched. A $\overline{\text{READ}}$/WRITE input pin has been provided to minimize these effects to the pulse detector. This is a standard TTL input.

When the $\overline{\text{READ}}$/WRITE pin is low, the pulse detector is in the read mode. When the $\overline{\text{READ}}$/WRITE pin is taken high, three things happen. First, the 1k resistors across the AMP IN pins are shunted by 300Ω resistors, as described previously in the Gain Controlled Amplifier section. Next, the amplifier is squelched so there is no signal on the Amp Output.

Application Information (Continued)

Finally, the previous AGC level is held. This AGC hold function is accomplished by not allowing any current to charge up the external C_{AGC}. The voltage across this capacitor will slowly reduce due to the bias current into the Darlington (see *Figure 6*) or through any resistor placed in parallel with C_{AGC}. Therefore, as described in the Automatic Gain Control section, the gain of the amplifier will slowly increase. All of these three events happen simultaneously.

When the $\overline{\text{READ}}$/WRITE input is returned low, the pulse detector will go back to the read mode in a specific sequence. First of all, the input impedance at the Amp In is returned to 1k. Then, after approximately 1 μs, the Gain Controlled Amplifier is taken out of the squelch mode, and finally approximately 1 μs after that, the AGC circuit is turned back on. This return to the read mode is designed to minimize analog transients in order to provide stable operation after 2.5 μs. It is very important that the analog input be stable before the chip is returned to the read mode. It is recommended that other than when writing, the Pulse Detector be in the read mode at all times in order to prevent the 2.5 μs delay from slowing up the system. The $\overline{\text{READ}}$/WRITE pin may be connected to the Write Gate output of a controller (such as the DP8466 Disk Data Controller).

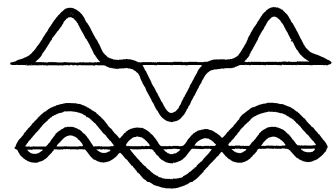
TIME CHANNEL FILTER

The peak detection is performed by feeding the output of the Gain Controlled Amplifier through an external filter to the differentiator. The differentiator output changes state when the input pulse changes direction, generally this will be at the peaks. The differentiator can also respond to noise near the baseline, in which case the comparator gating channel will inhibit the output. The purpose of the external filter is to bandwidth limit the incoming signal for noise considerations. Care must be used in the design of this filter to ensure the delay is not a function of frequency. For this reason, a high order Bessel filter with its constant group delay characteristics can be used in this application. Often, this filter must be specifically designed to correct errors introduced by the non-ideal phase characteristics of the input read head. The typcial −3 dB point for this filter is around 1.5 times the highest recorded frequency. The design of this filter is complex and will not be discussed here. However, the following discussion does give a feel for some of the considerations involved in the filter design. The reader is referred to reference #3 listed at the end of the Applications Notes for further filter design information.

Figure 10 shows a typical Region 1 waveform where there is no bit interaction. This waveform is primarily the sum of the fundamental frequency and its 3rd harmonic (higher odd harmonics are present when there is more shouldering).

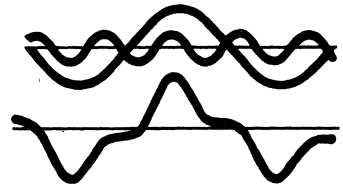
If the filter is to preserve this wave shape (this would be the case if no read/write head phase compensation were necessary) then the phase relationship between the fundamental frequency and its harmonics must not be altered. *Figure 11* shows the output when the 3rd harmonic has the proper magnitude, but the phase relationship is not maintained. The result is that the output waveform is not the same shape as the input (in a severe case it may be almost unrecognizable) and the time position of the peaks has been altered.

One electrical parameter which describes how well a filter will preserve a wave shape is called group delay. Group delay is defined as the change in phase divided by the change in frequency. If the group delay is constant over the



TL/F/5283–15

FIGURE 10. Typical Region 1 Waveform



TL/F/5283–16

FIGURE 11. Region 1 Waveform with the Incorrect Phase Relationship

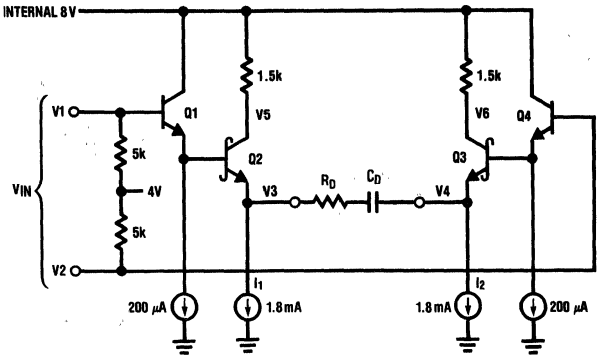
frequencies of interest, then the wave shape will be maintained. An MFM coded signal will contain three basic frequency components for the various digital patterns of data. For instance, a 10 Megabit/sec MFM signal will consist of analog frequencies of 2.5 MHz, 3.33 MHz and 5 MHz. On the outer track the bit density is the lowest and the 5 and 3.33 MHz signals will look sinusoidal while the 2.5 MHz signal will have a tendency to return to the baseline. This returning to the baseline is called shouldering and is illustrated in *Figure 10*. Since this shouldering is rich in 3rd harmonic—the 2.5 MHz signal will have a strong 7.5 MHz component. The 10 Megabit/sec MFM signal will therefore have 2.5 MHz, 3.33 MHz, 5 MHz, and 7.5 MHz components which must be filtered with constant group delay in order to reproduce the original waveform. For example, if the phase shift through the filter at 2.5 MHz is 33.3°, then at 3.33 MHz the phase shift must be 44.3°, at 5 MHz—66.6°, and at 7.5 MHz—99.9°. The group delay $\frac{d\theta}{dt}$ for this case is 13.32°/MHz. This can be better interpreted as a time delay. 33.3° of a 2.5 MHz signal is equivalent to (33.3/360) × (1/2.5 MHz) or 37 ns. Similarly, 66.6° on a 5 MHz signal is (66.6/360) × (1/5 MHz) = 37 ns.

The third order Bessel Filter as shown in the 10 Mbit/sec. pulse pairing measurement board on the data sheet is designed for a constant group delay and a −3 dB point of 7.5 MHz. At this frequency the delay through the filter is 35 ns. The Gain Controlled Amplifier of the DP8464B is designed for a group delay of a 7.8 ns ±0.5 ns for frequencies up to 7.5 MHz. The 7.8 ns delay in the Gain Controlled Amplifier and the 37 ns delay in the Bessel Filter do not introduce any timing error, only a delay of 44.3 ns from the Amp Input to the output of the filter.

DIFFERENTIATOR

A simplified circuit of the first stage of the differentiator is shown in *Figure 12*. The voltages at V3 and V4 are simply two diodes down from V1 and V2. Therefore the voltage

Application Information (Continued)



TL/F/5283–17

FIGURE 12. Simplified Differentiator First Stage

across the external differentiator network (C_d in series with R_d) is the differential input voltage V1 − V2. When R_d is zero, the current through C_d is $I = C \times (dV/dt)$ or $C_d \times (dV_{IN}/dt)$. The Q2 collector current is the sum of the 1.8 mA current source plus the current through C_d or

$$1.8\ mA + C_d \times (dV_{IN}/dt).$$

Similarly, the Q3 collector current is

$$1.8\ mA - C_d \times (dV_{IN}/dt).$$

Therefore, the differentiator output voltage, V5 − V6, is

$$1.5k \times 2 \times C_d \times (dV_{IN}/dt).$$

The input is at a peak when V5 − V6 = 0V.

The differentiator network (C_d and R_d) should be selected so the maximum current into the differentiator network is not greater than the minimum current of I1 and I2 over temperature. In the electrical specifications, the minimum current is specified for 1.4 mA (I_{Cd} Current into Pin 1 and 24 that discharges C_d). For example, the highest analog frequency in a 10 Megabit/sec, MFM signal is 5 MHz. Since the AGC loop has forced the input to the differentiator to 2 V_{PP} (which includes the 6 dB loss of the filter), then the voltage across the capacitor (assuming R_d is 0) is:

$$V_{IN} = 1 \times \sin(2 \times \pi \times 5E6 \times t)$$

and

$$dV_{IN}/dt = 1 \times 2 \times \pi \times 5E6 \times \cos(2 \times \pi \times 5E6 \times t)$$

and the maximum slope is

$$(dV_{IN}/dt)max = 1 \times 2 \times \pi \times 5E6 = 314E5\ V/sec.$$

For this example, C_d can now be calculated. Since $I = C \times (dV/dt)$, then for I = 1.4 mA, dV/dt = 314E5, then the maximum C_d must equal 45 pF. From this example, a following simple design equation for the value of C_d can be derived.

$$C_d = 445/(V_{IN} \times f_{max})$$

where

C_d is the maximum external differentiator capacitor in pF

V_{IN} is the peak to peak differential Time Channel input voltage

f_{max} is the maximum analog frequency in MHz

Note that this is the maximum value for the capacitor when the series resistor R_d is zero. The value of the capacitor can be increased if a series resistor is used, but the maximum current through the differentiator network must not exceed 1.4 mA. If too large a value for C_d is used, the delay through the differentiator will become dependent on frequency. This will not show up in a single frequency test such as a test for pulse pairing.

For the MFM code, the maximum analog frequency is ½ the data rate. For the ½(2,7) code, the maximum analog frequency is ⅓ the data rate. The above sinusoidal analysis is valid as long as the highest frequency on the outer track is nearly sinusoidal. If, however, there is significant shouldering of this signal then the value of C_d should be reduced accordingly.

The following table summarizes the value of C_d to use for a 2 V_{pp} differential signal to the time channel input.

Data Rate	Code	Maximum Frequency	C_d
5 mbits/sec	MFM	2.5 MHz	90 pF
5 mbits/sec	2,7	1.6 MHz	140 pF
10 mbits/sec	MFM	5.0 MHz	45 pF
10 mbits/sec	2,7	3.3 MHz	67 pF

As noted above, the value of the capacitor can be increased if a series resistor is used, but the maximum current through the differentiator network must not exceed 1.4 mA. For example, the components used in the Pulse Pairing Setup (see AC Electrical Specifications) are for a typical 10 Mbits/sec MFM drive. The combination of the C_d of 50 pF and the R_d of 430Ω gives a combined impedance of 768Ω at the highest frequency of 5 MHz. This gives a maximum current of 1.3 mA—well below the 1.4 mA limit.

A resistor is placed in series with C_d in order to bandlimit the differentiator response. This resistor also has an effect on the phase linearity of the differentiator. An ideal differentiator produces an output that is 90 degree phase shifted from the input regardless of the input frequency. The presence of the series resistor produces an output phase shift that is less than 90 degrees and changes with the input frequency. This resistor can be used to correct for frequency related phase problems encountered elsewhere in the read path.

Application Information (Continued)

To properly decode the information on the disk, the read channel must determine if there is a peak (or a "1") during a period of time called a detection window. The detection window for MFM and the (2,7) code is

$$1/(2 \times \text{data bit rate}).$$

This detection window must accommodate errors in many parts of the system including filters, data separator, and peak shift variations in the data pattern. The pulse pairing of the DP8464B should be included in the error budget calculation.

Unequal delays through the bi-directional one shots will contribute to pulse pairing. To minimize this effect, pin 2 should be connected to 22 and pin 23 should be connected to 21. If connected this way, the delays tend to cancel. For the PCC Package, Pin 26 to Pin 2, and Pin 25 to Pin 27.

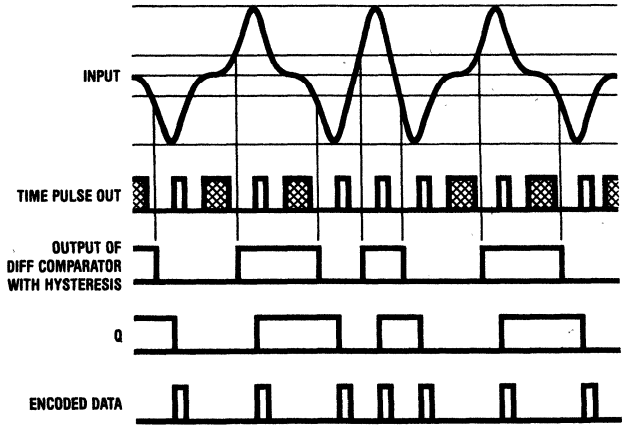
DIFFERENTIAL COMPARATOR WITH HYSTERESIS

The actual peak detection is done in the time channel with the differentiator. Unfortunately, the differentiator not only responds to signal peaks but also responds to noise at the baseline. In order to prevent this noise from generating false data, the signal at the output of the Gain Controlled Amplifier is also passed through a gating channel which prevents any output change before the input signal has crossed an established level. This gating channel comprises a differential comparator with hysteresis and a D flip-flop. The hysteresis for this comparator is set externally via the Set Hysteresis pin. The amount of hysteresis is twice the voltage on the Set Hysteresis pin. For instance, if the voltage on the Set Hysteresis pin is 0.3V, the differential input signal must be larger than 0.6V($\pm$0.3V) before the output of the comparator will change states. The 0.6V hysteresis represents 30% of a typical 2V differential input signal level to the gating channel. The hysteresis level is usually set between 15% to 40% of the differential input signal.

The operation of the gating channel is shown in *Figure 13*. At the top is a typical Region 1 waveform which exhibits shouldering on the lowest frequency and is almost sinusoidal on the highest frequency. In this example, this waveform is fed to both the timing and the gating channel. The hysteresis level (of about 25%) has been drawn on this waveform. The second waveform is the output of the differentiator and its bi-directional one shot. This is the waveform on the Time Pulse Out pin. While there is a positive edge pulse at each peak, there is also noise at the shoulders. In this example, the Time Pulse Out is connected directly to the Time Pulse In without any external delay. This output is therefore the clock for the D flip-flop.

The third waveform in *Figure 13* is the output of the Comparator with Hysteresis which goes to the D input of the flip-flop. The true peaks are the first positive edges of the Time Pulse Out which occur after the output of the comparator has changed states. The D flip-flop will "clock" in these valid peaks to the output bi-directional one shot. Therefore, the noise pulses (due to the differentiator responding to noise at the baseline) just "clock" in the old data through the flip-flop and the output does not change.

The Q output of the flip-flop drives the output bidirectional one-shot which generates the positive edges corresponding to the peaks. The width of the data pulses can be controlled by an external capacitor from the Set Pulse Width pin to ground. This pulse width can be adjusted from 20 ns to ½ the period of the highest frequency. Typical values for this capacitor are 20 pF for a 25 ns pulse width to 100 pF for a 100 ns pulse.



TL/F/5283–19

FIGURE 13. Time and Gate Channel Operation for Region 1 Signals

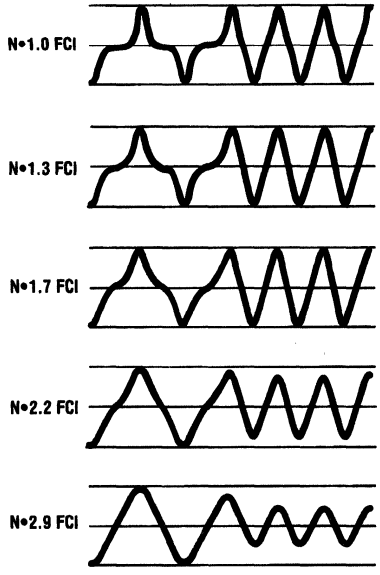
Application Information (Continued)

PULSE DETECTOR OPERATION IN REGIONS 1 AND 2

Figure 14 shows the input waveform for the lowest frequency followed by the highest frequency for an MFM code. In MFM the highest frequency is twice the lowest frequency. The outer track has the least flux changes per inch (FCI) and is illustrated in the waveforms at the top. There is so much room between the pulses that the signal returns to the baseline for the lowest frequency while there is shouldering at the highest frequency. As you go towards the inner track, the pulses become more crowded and bit interaction occurs. At the third curve down (N × 1.7 FCI), there is shouldering at the lowest frequency while the highest frequency is almost sinusoidal. At higher bit densities, the lowest frequency looks sinusoidal, while the highest frequency is decreasing in amplitude. In *Figure 14*, the first three waveforms are examples of Region 1 operation (very little change in amplitude with frequency). The last two waveforms are examples of Region 2 operation.

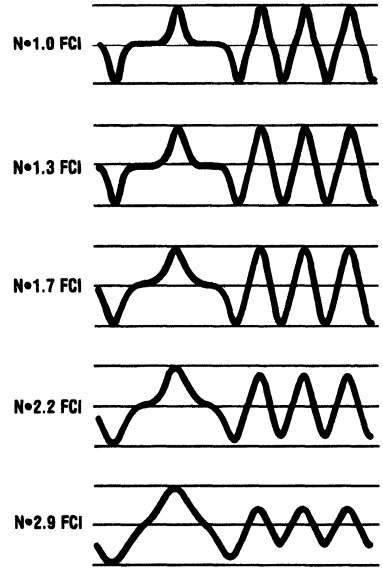
In a disk system, the bit density changes about a factor of 1.7 between the inner and the outer track. For instance, if the input waveform for the F-2F signal on the inner track of a system looks similar to waveform #4 in *Figure 14* (N × 2.2 FCI), then the outer track will have a bit density that is approximately N × 2.2/1.7 or N × 1.3 FCI. This is shown in the second waveform. Tracks half the way in will have a bit density of the average between the inner and outer tracks, in this case N × 1.7 FCI which is illustrated in the third waveform. Note that the analog waveforms change considerably with track location. Self-gating circuits ("desnakers") can be used in MFM systems which operate in these last three curves (from N × 1.7 FCI to N × 2.9 FCI). If the FCI becomes much less, the shouldering on the lowest frequency will let in too much noise. If the FCI is increased, the peak resolution gets very poor. Now we can compare these waveforms to longer run length limited codes.

Figure 15 shows the analog waveform for the lowest frequency followed by the highest frequency for a 2,7 code. In the 2,7 code, the frequency range is from F to 2.66 × F. Unlike the MFM code, there is no region where the self-gating "desnaker" will work on both the inner and outer tracks.



TL/F/5283–20

FIGURE 14. MFM F-2F Pulse Waveforms for Various Flux Changes per Inch



TL/F/5283–21

FIGURE 15. F-2.66 × F Pulse Waveforms for Various Flux Changes per Inch

Application Information (Continued)

The simplest operation is for systems operating entirely in Region 1, that is, no amplitude reduction between the highest and the lowest frequency at the inner track. The inner track is specified because the pulse interaction is most severe on the inner track. For Region 1 operation, only the Time Channel filter is required, so the Gate Channel Input is connected to the Time Channel Input. Since no external time delay is required to align the time and gate channels, the Time Pulse Out is connected directly to the Time Pulse In. The Region 1 connection is shown in *Figure 2*. The internal timing for this operation is shown in *Figure 13*.

If there is significant amplitude reduction at the highest frequency, the peak detection becomes more complex. If the worst case waveform is like the fourth waveform on *Figure 14*, then the Region 1 connection might still work satisfactorily. However, if the input begins to approach the fifth waveform, this system configuration will completely fail. One problem is that the AGC will respond to the frequency dependent amplitude modulation and distort the waveform.

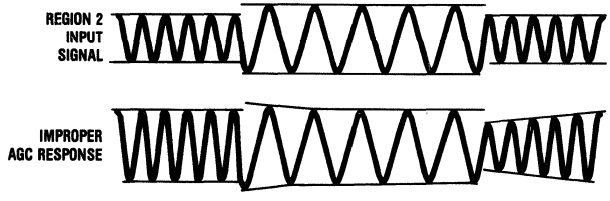
Figure 16 illustrates this problem which is encountered in systems operating in Region 2. If the input digital pattern suddenly shifts from a high frequency to a low frequency, the bit density may shift from the 70% level on the BPI curve of *Figure 1* to a point at 90% on the BPI curve. As shown, the AGC loop is correcting for this frequency-induced change in amplitude by quickly decreasing the amplifier gain. The situation gets worse if the input digital pattern shifts back to a high frequency. The AGC loop now cannot quickly increase the amplifier gain, so the output waveform will very slowly increase. The AGC response to frequency related amplitude change is not desirable since the AGC is now distorting the input waveform. This can be prevented by inserting a lead network between the Gain Controlled Amplifier's output and the AGC input, as shown in *Figure 17*. This will increase the amplitude of the higher frequency into the AGC, thereby preventing the AGC from changing gain.

Another problem encountered in Region 2 operation is that the amplitude of the highest frequency may be so low that it may not trip the hysteresis level. If this happens, these peaks would not be gated on to the output. This problem can also be corrected by placing a separate filter to the gating channel which will make the amplitude of the highest frequency equal the amplitude of the lowest frequency. This is illustrated in the following example.

Consider a disk system which uses the 2,7 code and has an input at the inner track which looks like the fifth waveform in *Figure 15*. Since the flux density on the outer track is 1/1.7 times the flux density of the inner track, the outer track waveform will look like the third waveform. One filter cannot perfectly compensate both these extremes, so we design to compensate a waveform between these two. The track which is ⅔ of the way in towards the inner track is a good compromise. The filter in this example is a single zero placed such that the lowest frequency followed by the highest frequency have the same amplitude on the track ⅔ of the way in. *Figure 18* shows the operation of the inner track of this example. While the gating channel filter has made the amplitudes of the two frequencies nearly the same, the time relationship to the Time Channel Input has not been preserved. The proper operation is to have the positive edge of the signal at the Time Pulse In pin, which corresponds to a peak, be the first positive edge after the output of the comparator has changed states. This can be accomplished either of two ways. One way is to insert an external delay between the Time Pulse Out and the Time Pulse In as shown in *Figure 18*. The required delay can be determined by comparing the Time Pulse Out to the Channel Alignment Output with both external filters in the circuit. Another way is to design the Time Channel Filter with more group delay. This will probably require additional poles.

Figure 19 shows the outer track operation of our example. Notice how the system has taken care of the shoulder-induced-noise on the Time Pulse Out. The external delay has shifted the Time Pulse In so the noise is not clocking in new data to the flip-flop. It is important to select this delay such that the positive edge corresponding to a signal peak is always the first positive edge after the output of the comparator has changed states.

While the gating filter has equalized the amplitudes between the highest and the lowest frequency, the amplitude between the inner and the outer track has not been held constant. This can be seen by comparing the Gate Channel Input between *Figure 18* and *Figure 19*. In order to avoid saturating the Gain Controlled Amplifier, the voltage on the V_{REF} pin must be set so that the voltage out of the Gain Controlled Amplifier is 4 Vpp or less for all tracks. The low frequency signal on the inner track contains far more fundamental frequency than the low frequency signal on the outer track. Consequently, the low frequency inner track signal will experience more attenuation than the low frequency outer track signal in passing through the gating channel filter which, for this example, has been optimized to pass higher frequencies. The AGC tends to hold the input to the gating channel constant for a fixed V_{REF} level. Therefore the largest output from the Gain Controlled Amplifier is for the low frequency inner track signal. The voltage on V_{REF} should be adjusted so that the differential output swing of the Gain Controlled Amplifier is 4 Vpp maximum for this signal. This means that the output voltage on the outer track will be less than 4 Vpp.



TL/F/5283–22

FIGURE 16. Improper AGC Response to Region 2 Signal

Application Information (Continued)

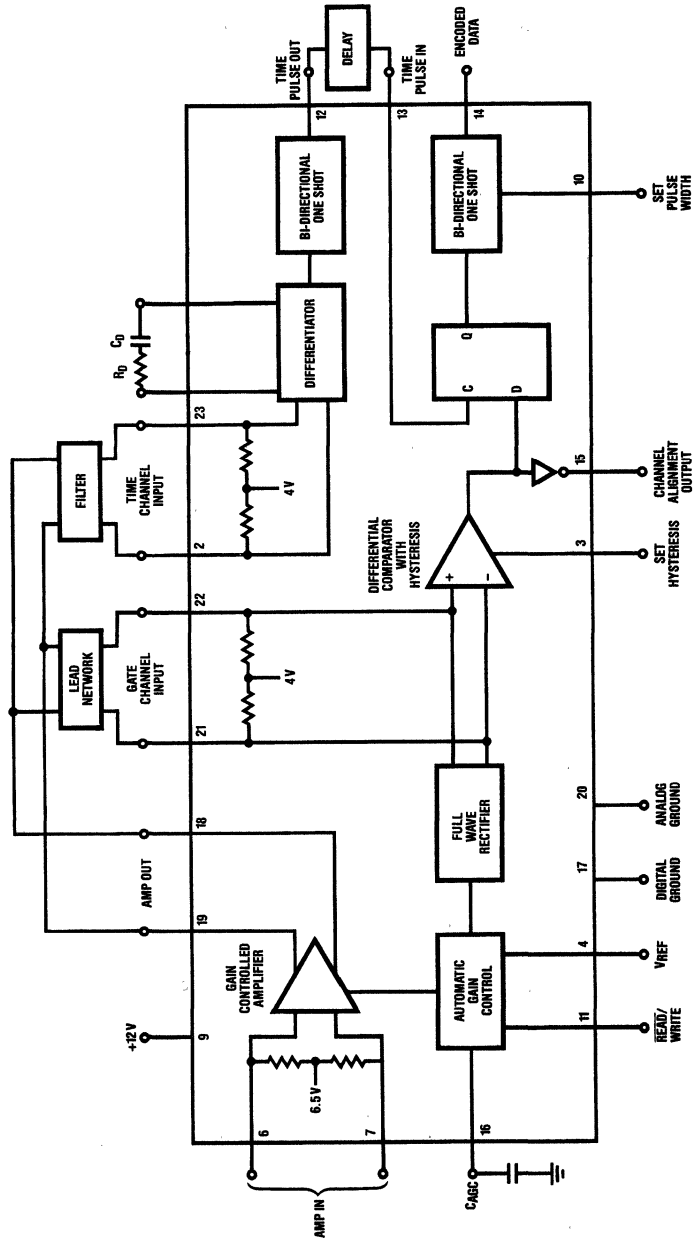
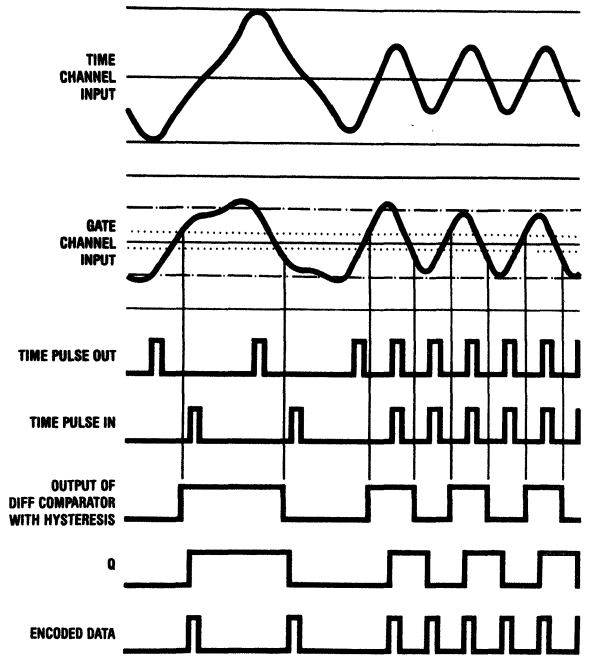


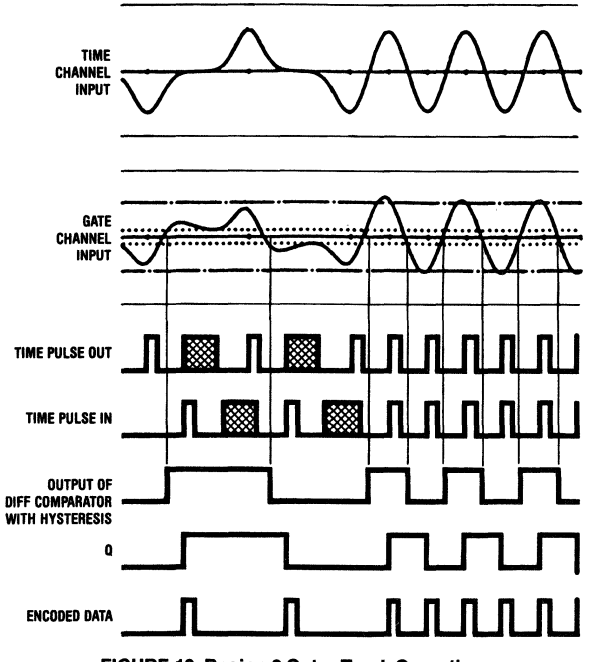
FIGURE 17. Circuit Connection for Region 2 Operation

Application Information (Continued)



TL/F/5283–24

FIGURE 18. Region 2 Inner Track Operation



TL/F/5283–25

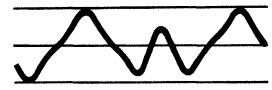
FIGURE 19. Region 2 Outer Track Operation

Application Information (Continued)

Another troublesome input pattern which should be investigated is a high frequency triplet surrounded by the lowest frequency as shown in *Figure 20*. Since the center bit of the triplet does not rise very much above the baseline, there is the possibility it will not trip the hysteresis level. This pattern should be checked to ensure the gating channel filter raises this center bit enough for the proper operation of the gating channel. The operation of the triplet in the previous example is shown in *Figure 21*.

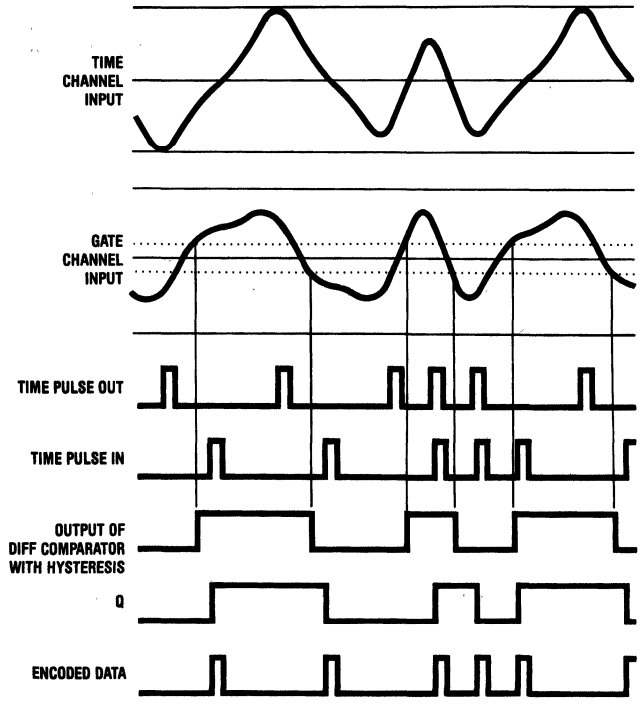
LAYOUT CONSIDERATIONS

Figure 22 is a top view of the component layout for the DP8464B application board whose schematic is shown in *Figure 23*. Care must be exercised in the board layout in order to isolate all digital signals from analog signals. The layout shown in *Figure 22* is a good example of what is required in this regard. In particular the Amp. In pins (pins 6 and 7) and the C_{DIFF} pins (pins 1 and 24) must be isolated from all digital signals. An analog ground plane will greatly aid in this isolation as will separate digital and analog grounds. The V_{CC} (pin 9) should have a 0.1 μf bypass capacitor to analog ground located close to the DP8464B. The component list is provided as an example. These components will need to be optimized for a specific read channel.



TL/F/5283–26

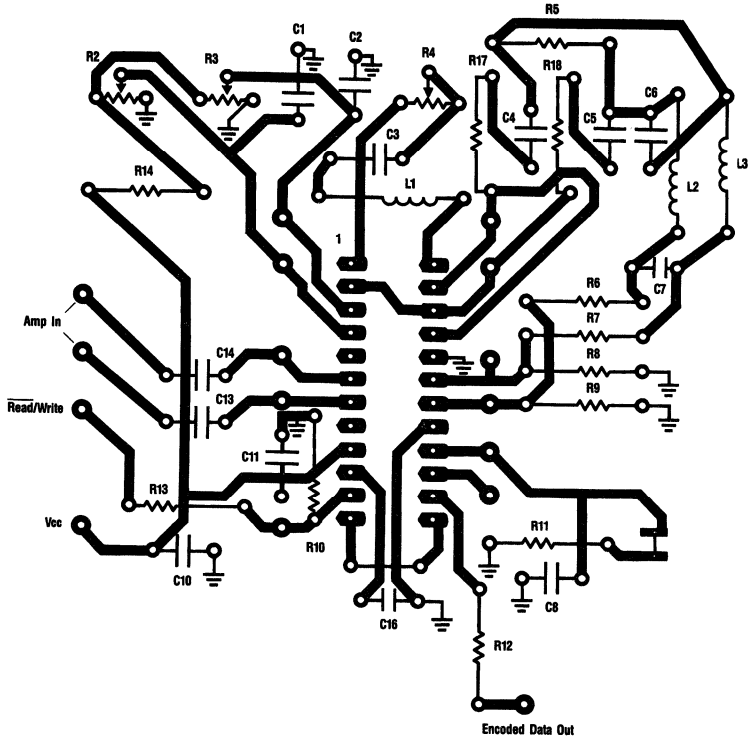
FIGURE 20. (2,7) Triplet



TL/F/5283–27

FIGURE 21. Region 2 Triplet Operation

Application Information (Continued)



TL/F/5283–28

FIGURE 22. DP8464B Component Layout—Top View

Application Information (Continued)

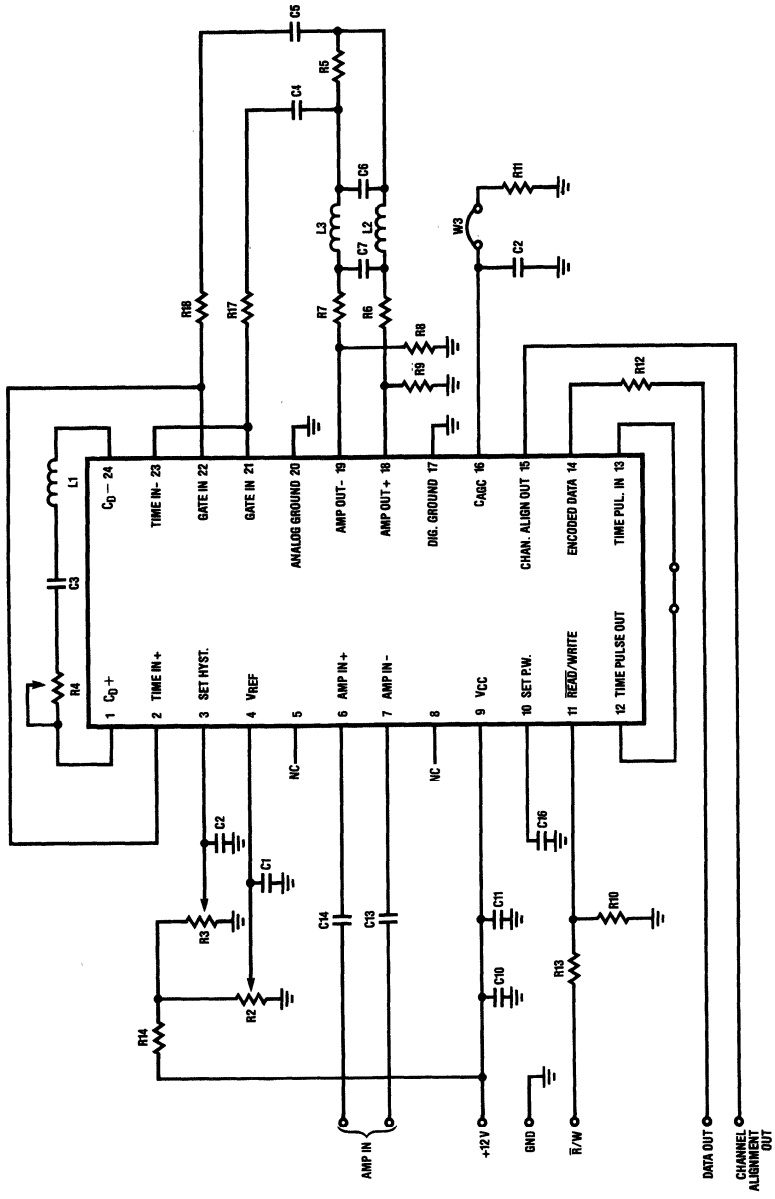


FIGURE 23. DP8464B Application Board Schematic

Application Information (Continued)

PARTS LIST FOR DP8464B BOARD

Component Name	Note #	Function	Value	Value for 5 Mbits/sec	Value for 10 Mbits/sec
R2	3	Adjustment for V_{REF} (AGC amplitude)	1k pot		
R3	3	Adjustment for Set Hyst. (threshold)	1k pot		
R4	2	Adjustment for differentiator network Q	5k pot		
R5	1	Low pass filter resistor	560Ω		
R6	1	Low pass filter resistor	240Ω		
R7	1	Low pass filter resistor	240Ω		
R8		Amp Out emitter bias resistor	4.3k		
R9		Amp Out emitter bias resistor	4.3k		
R10		Pull down resistor for Read/Write Pin	5.1k		
R11		Resistor in parallel with C_{AGC}	100k		
R12		Encoded Data Out damping resistor	51Ω		
R13		Read/Write damping resistor	51Ω		
R14		Divider network for Set Hyst. and V_{REF}	2.4k		
R17	6	Series resistor for Time Channel Input	Not required on DP8464B		
R18	6	Series resistor for Time Channel Input	Not required on DP8464B		
C1		V_{REF} cap	0.1 μF		
C2		Set Hyst. cap	0.1 μF		
C3	2	Differentiator cap		100 pF	50 pF
C4		Time and Gate Channel In coupling cap	0.01 μF		
C5		Time and Gate Channel In coupling cap	0.01 μF		
C6	1	Low pass filter cap		200 pF	100 pF
C7	1	Low pass filter cap		30 pF	15 pF
C8	4	C_{AGC} cap	0.01 μF		
C10		V_{CC} cap	1.0 μF		
C11		V_{CC} cap	0.1 μF		
C13	5	Amp In coupling cap	2200 pF		
C14	5	Amp In coupling cap	2200 pF		
C16		Set Pulse Width cap		100 pF	50 pF
L1	2	Differentiator inductor		3.6 μH	1.6 μH
L2	1	Low pass filter inductor		10 μH	4.7 μH
L3	1	Low pass filter inductor		10 μH	4.7 μH

BREADBOARD OPERATION NOTES

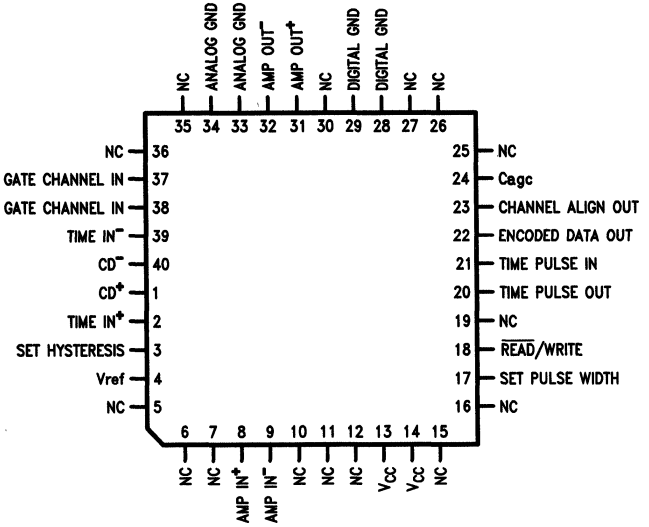
1. The low pass filter is a 3 pole Bessel with the corner frequency at 3.75 MHz for the 5Mbits/sec board (7.5 MHz for the 10 Mbits/sec board).
2. The differentiator is a simple RLC filter with the break frequency at 8.5 MHz for the 5 Mbits/sec board (17 MHz for the 10 Mbits/sec board). The resistor can be adjusted to correct for phase distortion in the channel.
3. The V_{REF} should be set at 0.5V. Since the low pass filter has a 6 dB loss, the signal on AMP OUT is 4 Vpp differential while the amplitude into the gate channel is 2 Vpp differential. The Set Hyst. should be nominally set at 0.3V.
4. The AGC attack time (the response to an increased input amplitude) is about 2 μs. To increase this time, increase the value of C8 (the AGC capacitor). The AGC decay time (the response to a decrease in amplitude) is about 10 ms. To increase this time, increase the value of R11. Care must be taken to not allow the response of the AGC loop to become too fast, otherwise loop instability may occur.
5. The input pole is set at 72 kHz (1k input impedance and a 2200 pF input coupling capacitor).
6. Pulse pairing (described in the differentiator section of this data sheet) can be caused by unequal delays through the Bi-directional one shots. To minimize this effect, pin 2 should be connected to pin 22, and pin 23 should be connected to pin 21. If connected this way, the delays tend to cancel.

REFERENCES

1. I. H. Graham, "Data Detection Methods vs. Head Resolution in Digital Recording," IEEE Transactions on Magnetics Vol. MAG-14, No. 4 (July 1978)
2. I. H. Graham, "Digital Magnetic Recording Circuits," to be published.
3. Anatol I., Zverev, Handbook of Filter Synthesis, John Wiley & Sons publisher, 1967.

Connection Diagrams (Continued)

TapePak®



TL/F/5283–31

Top View
Order Number DP8464BTP-3 or DP8464BTP-2
See NS Package Number TP40A

National Semiconductor

PRELIMINARY

DP8468B
Disk Pulse Detector + Embedded Servo Detector

General Description

The DP8468B Disk Pulse Detector + Servo utilizes analog and digital circuitry to detect amplitude peaks of the signal received from the read/write amplifier fitted with the heads of disk drives. The DP8468B produces a TTL compatible output which, on the positive leading edge, indicates a signal peak. Electrically, these peaks correspond to flux reversals on the magnetic medium. The signal from the read/write amplifier when reading a disk is therefore a series of pulses with alternating polarity. The Disk Pulse Detector accurately replicates the time position of these peaks.

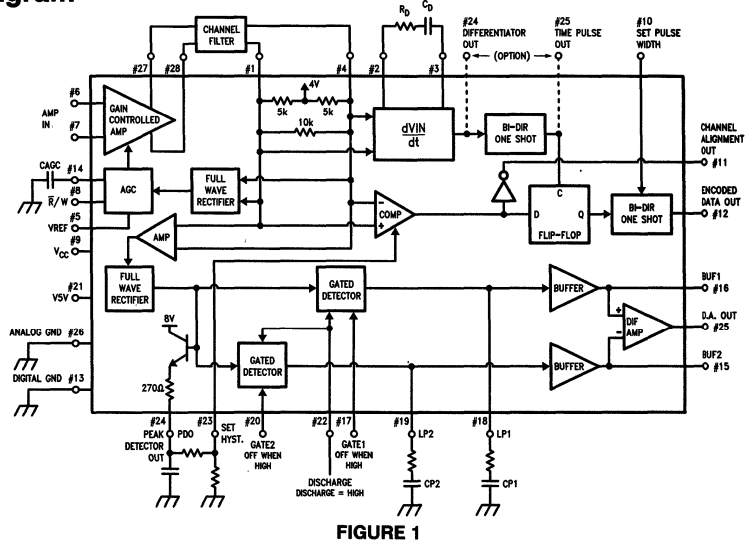
The DP8468B also incorporates two gated detectors which detect embedded servo information, used for head positioning. It is primarily intended for detecting a burst type servo pattern as shown in *Figure 16*. However, with external sync detection circuitry it will detect a tri-bit type servo pattern also shown in *Figure 16*. The DP8468B provides two buffered low impedance voltage outputs which represent the peak detected level of each servo burst. These voltages are suitable for digitizing by an analog to digital converter, as would typically be done in a system that utilizes stepper motors for head positioning. The DP8468B also provides a low impedance output that represents the difference voltage, centered about an external reference voltage, between the two servo channels. This voltage is useful in servo systems using a linear voice coil for head positioning.

The Disk Pulse Detector + Servo is fabricated using an advanced oxide isolated Schottky process, and has been designed to function with data rates up to 15 Megabits/second. The DP8468B is available in a surface mount 28-pin plastic chip carrier package. Normally, it will be fitted in the disk drive, and its output may be directly connected to any of National's Data Separators or Synchronizers.

Features

- Wide input signal amplitude range—from 20 mVpp to 660 mVpp differential
- Data rates up to 15 Megabits/sec ½(2,7) code
- On-chip differential gain controlled amplifier, differentiator, comparator gating circuitry, and output pulse generator
- Input capacitively coupled directly from the disk head read/write amplifier
- Adjustable comparator hysteresis
- Dynamic hysteresis tracks signal amplitude
- AGC and differentiator time constants set by external components
- TTL compatible digital Inputs and Outputs
- Encoded Data Output may connect directly to any of National's Data Separators or Synchronizers
- Standard drive supply: 12V ±10%, 5V ±5%
- Built in embedded servo detector
- On chip buffers provide low impedance servo output voltages
- User adjustable servo time constants
- Differentiator and time pulse outputs available as pins on special engineering parts
- Available in a surface mount 28-pin plastic chip carrier package or 40-pin TapePak® package

Block Diagram



TL/F/8828–1

FIGURE 1

Absolute Maximum Ratings

If Military/Aerospace specified devices are required, please contact the National Semiconductor Sales Office/Distributors for availability and specifications.

	Pins	Limit
Supply Voltage	9	14V
TTL Input Voltage	8, 17, 20, 22	5.5V
TTL Output Voltage	12, 11	5.5V
Input Voltage	23, 5	5.5V
Minimum Input Voltage	23, 5	−0.5V
Differential Input Voltage	6–7, 4–1,	3V or −3V

ESD susceptibility rating is to be determined.

	Limit
Storage Temperature	−65°C to 150°C
Lead Temp. (Soldering, 10 seconds)	300°C
Maximum Power Dissipation at 25°C Plastic Chip Carrier Package (derate 12.5 mW/°C above 25°C)	1560 mW

Operating Conditions

	Min	Typ	Max	Units
Supply Voltage, (V_{CC})	10.8	12.0	13.2	V
Logic Supply, (V5V)	4.75	5	5.25	V
Ambient Temperature, (T_A)	0		70	°C

DC Electrical Characteristics

Over Recommended Operating Temperature and Supply Range V_{REF} = 0.5V, Set Hysteresis = 0.3V. $\overline{Read}$/Write = 0.3V, $V_{PIN\ 17}$ = $V_{PIN\ 20}$ = 2V, $V_{PIN\ 22}$ = 0.3V, unless otherwise noted.

Symbol	Pins	Parameter	Conditions	Min	Typ	Max	Units
AMPLIFIER							
$Z_{IN_{AI}}$	6, 7	Amp In Impedance	T_A = 25°C (Note 1)	1.4	2.0	2.6	kΩ
A_{VMIN}	28, 27	Min Voltage Gain	AC Output 4 Vpp Differential			7.0	V/V
A_{VMAX}	28, 27	Max Voltage Gain	AC Output 4 Vpp Differential	200			V/V
$V_{C_{AGC}}$	14	Voltage on C_{AGC}	A_V = 7.0 A_V = 200	• 2.8	4.5 3.7	5.5	V V
CHANNEL							
$Z_{IN_{CI}}$	4, 1	Channel Input Impedance	T_A = 25°C (Note 1)		2.5		kΩ
$I_{C_{AGC}-}$	14	Pin 14 Current which Charges C_{AGC}	$V_{PIN\ 14}$ = 3.9V $\lvert V_{PIN\ 4} - V_{PIN\ 1}\rvert$ = 1.3V	−2.5	−4.2	−6	mA
$I_{C_{AGC}+}$	14	Pin 14 Current which Discharges C_{AGC}	$V_{PIN\ 14}$ = 5V $\lvert V_{PIN\ 4} - V_{PIN\ 1}\rvert$ = 0.7V		0.1	1	μA
$I_{V_{REF}}$	5	V_{REF} Input Bias Current			−20	−100	μA
$V_{TH_{AGC}}$	4, 1 5, 14	AGC Threshold	(Note 2), $V_{PIN\ 14}$ = 4.2V	0.915	1.04	1.165	V
I_{SH}	23	Set Hysteresis Input Bias Current			−40	−70	μA
$V_{TH_{SH}}$	4, 1 23, 11	Set Hysteresis Threshold	(Note 3)	0.48	0.6	0.72	V
I_{C_d}	2, 3	Current into Pin 2 and 3 that Discharges C_d		1.4	1.8		mA
WRITE MODE							
$Z_{IN_{AI}}$	6, 7	Amp In Impedance in Write Mode	$V_{PIN\ 8}$ = 2.0V		100	200	Ω
$I_{C_{AGC}-}$	14	Pin 14 Current in Write Mode	$V_{PIN\ 8}$ = 2.0V $V_{PIN\ 14}$ = 3.9V $\lvert V_{PIN\ 4} - V_{PIN\ 1}\rvert$ = 2.6V		0.1	1.0	μA

DC Electrical Characteristics Over Recommended Operating Temperature and Supply Range V_{REF} = 0.5V, Set Hysteresis = 0.3V. Read/Write = 0.3V, $V_{PIN\ 17}$ = $V_{PIN\ 22}$ = 0.3V, unless otherwise noted. (Continued)

Symbol	Pins	Parameter	Conditions	Min	Typ	Max	Units
DIGITAL PINS							
V_{IH}	8, 17, 20, 22	High Level Input Voltage		2			V
V_{IL}	8, 17, 20, 22	Low Level Input Voltage				0.8	V
V_I	8, 17, 20, 22	Input Clamp Voltage	V_{5V} = Min, I_I = −18 mA			−1.5	V
I_{IH}	8, 17, 20, 22	High Level Input Current	V_{5V} = Max, V_I = 2.7V			20	μA
I_I	8, 17, 20, 22	Input Current at Maximum Input Voltage	V_{5V} = Max, V_I = 5.5V			1	mA
I_{IL}	8, 17, 20, 22	Low Level Input Current	V_{5V} = Max, V_I = 0.5V			−200	μA
V_{OH}	12	High Level Output Voltage	V_{5V} = Min, I_{OH} = −40 μA (Note 4)	2.7			V
V_{OL}	12	Low Level Output Voltage	V_{5V} = Min, I_{OL} = 800 μA (Note 4)			0.5	V
I_{LH}	11	High Level Output Leakage Current	$V_{Pin\ 11}$ = 5.5V Measure Current into Pin 11			10	μA
V_{OL}	11	Low Level Output Voltage	$I_{PIN\ 11}$ = 5 mA			0.5	V
I_{OS}	12	Output Short Circuit Current	V_{5V} = Max, V_O = 0V			−100	mA
SERVO CHANNEL							
Z_{DIS}	18, 19	Discharge Impedence	$V_{PIN\ 22}$ = 2V Force 3V on Pins 18 or 19.	1.0	1.5	2.3	kΩ
V_{BOQ}	15, 16	Buffer Quiescent Output Level	$V_{PIN\ 17, 20, 22}$ = 0.3V V_{CI} = 0V, $V_{PIN\ 21}$ = 5V Pull 0 mA from Pins 15 and 16.	1.1	1.2	1.3	V
Av_Q	15, 16	Quiescent Output Level Gain to Pin 21	$V_{PIN\ 17, 20, 22}$ = 0.3V V_{CI} = 0V, $V_{PIN\ 21}$ = 4.5V Pull 0 mA from Pin 15 and 16. $Av_Q = \frac{V_{BO_Q} - V_{PIN\ 15\ or\ 16}}{5V - 4.5V}$		0.226		V/V
V_{PDOQ}	24	Peak Detector Quiescent Output Level	$V_{PIN\ 21}$ = 5V, V_{CI} = 0V, Pull 200 μA from Pin 24	0.14	0.36	0.64	V
I_L	18, 19	Gated Off Leakage Current	$V_{PIN\ 22}$ = 0.3V $V_{PIN\ 20}$ = $V_{PIN\ 17}$ = 2V Force 6.6V on Pin 18 or Pin 19	−1	0.2	1	μA
$V_{OS_{BO}}$	16, 15	Buffer Output Offset Voltage For V_{CI} = $1V_{pk\text{-}pk}$	$V_{PIN\ 17, 20, 22}$ = 0.3V $V_{PIN\ 1}$ = 4.25V $V_{PIN\ 4}$ = 3.75V Pull 0 mA from Pin 15 and 16. $V_{OS_{BO}}$ = $V_{PIN\ 16}$ − $V_{PIN\ 15}$		±1	±15	mV
$V_{OS_{SYS}}$	25, 21	System Output Offset Voltage For V_{CI} = $1V_{pk\text{-}pk}$	$V_{PIN\ 17, 20, 22}$ = 0.3V, $V_{PIN\ 21}$ = 5V V_{PIN1} = 4.25V, V_{PIN4} = 3.75V Pull 0 mA from Pin 25 $V_{OS_{SYS}}$ = $V_{PIN\ 25}$ − $V_{PIN\ 21}$		±5	±20	mV
$Av_{DA}\|_{2V}$	25, 21	Difference Amplifier Gain, 2V Differential Input	$V_{PIN\ 17, 20}$ = 2V, $V_{PIN\ 21}$ = 5V $V_{PIN\ 22}$ = 0.3V $V_{PIN\ 19}$ = 5.4V, $V_{PIN\ 18}$ = 3.4V $Av_{DA} = \frac{V_{PIN\ 21} - V_{PIN\ 25}}{V_{PIN\ 19} - V_{PIN\ 18}}$	0.42	0.49	0.57	V/V
$Av_{DA}\|_{1V}$	25, 21	Difference Amplifier Gain, 1V Differential Input	$V_{PIN\ 17, 20}$ = 4V, $V_{PIN\ 21}$ = 5V $V_{PIN\ 22}$ = 0.3V $V_{PIN\ 19}$ = 4.9V, $V_{PIN\ 18}$ = 3.9V $Av_{DA} = \frac{V_{PIN\ 21} - V_{PIN\ 25}}{V_{PIN\ 19} - V_{PIN\ 18}}$	0.43	0.5	0.57	V/V

DC Electrical Characteristics Over Recommended Operating Temperature and Supply Range V_{REF} = 0.5V, Set Hysteresis = 0.3V. $\overline{Read}$/Write = 0.3V, $V_{PIN\ 17}$ = $V_{PIN\ 22}$ = 0.3V, unless otherwise noted. (Continued)

Symbol	Pins	Parameter	Conditions	Min	Typ	Max	Units
SERVO CHANNEL (Continued)							
GL_{DA}	25	Difference Amplifier Gain Linearity	$GL_{DA} = \left\| \frac{Av_{da}\|_{2V}}{Av_{da}\|_{1V}} - 1 \right\| \times 100$ $V_{PIN\ 21}$ = 5V		0.5	2.5	%
$Z_{PDO\ Source}$	24	Peak Detector Out Output Impedence	$V_{PIN\ 17,\ 20,\ 22}$ = 0.3V, V_{CI} = 1V Measure $V_{PIN\ 24}$ with 200 µA and 3 mA Pulled out of the Pin. $Z_{PDO\ Source} = \frlvert{\text{change in } V_{PIN\ 24}}{3\ mA - 0.2\ mA}$	150	200	250	Ω
$Av_{gd}\|_{2V}$	15, 16	Gated Detector Gain For V_{CI} = $2V_{pk-pk}$	$V_{PIN\ 22,\ 20,\ 17}$ = 0.3V V_{PIN1} = 4.5V, V_{PIN4} = 3.5V $Av_{gd} = \frac{V_{PIN\ 15\ or\ 16} - V_{BOQ}}{V_{CI}}$	1.45	1.8	2.25	V/V
$Av_{gd}\|_{1V}$	15, 16	Gated Detector Gain For V_{CI} = $1V_{pk-pk}$	$V_{PIN\ 22,\ 20,\ 17}$ = 0.3V V_{PIN1} = 4.25V, V_{PIN4} = 3.75V $Av_{gd} = \frac{V_{PIN\ 15\ or\ 16} - V_{BOQ}}{V_{CI}}$	1.6	2	2.4	V/V
$AV_{PDO}\|_{2V}$	24	Peak Detector Output Voltage Gain For V_{CI} = $2V_{pk-pk}$	V_{PIN1} = 4.5V, V_{PIN4} = 3.5V $Av_{PDO} = \frac{V_{PIN\ 24\ pk-pk}}{V_{CI}}$	1.45	1.9	2.25	V/V
$AV_{PDO}\|_{1V}$	24	Peak Detector Output Voltage Gain For V_{CI} = $1V_{pk-pk}$	V_{PIN1} = 4.25V, V_{PIN4} = 3.75V $Av_{PDO} = \frac{V_{PIN\ 24_{AC}}}{V_{CI}}$	1.6	2.0	2.4	V/V
GL_{gd}	15, 16	Gated Detector Gain Linearity	$Gl_{gd} = \frac{\frac{\|V_{PIN\ 16} - V_{PIN\ 15}\|_{eighth}}{\|V_{PIN\ 16} - V_{PIN\ 15}\|_{quarter}} - 0.5}{0.5} \times 100$ $\|V_{PIN\ 16} - V_{PIN\ 15}\|_{eighth}$ = (Note 5) $\|V_{PIN\ 16} - V_{PIN\ 15}\|_{quarter}$ = (Note 6)		±0.5	±2.5	%
I_{CC}	9	V_{CC} Supply Current	V_{CC} = Max	25	46	65	mA
I_{5V}	21	5V Supply Current	V_{5V} = Max	3	7	11	mA

AC Electrical Characteristics Over Recommended Operating Temperature and Supply Range Refer to AC Test Setup. f = 2.5 MHz unless otherwise indicated.

Symbol	Pins	Parameter	Conditions	Min	Typ	Max	Units
t_{charge}	15, 16	Gated Detector Charge Time	V_{CI} = 2V pk-pk, $V_{PIN\ 22}$ = 0.3V, S1 & S2 = Closed. With LP1 and LP2 discharged, measure the time from Pin 17 or 20 going from 2V to 0.3V, to V_{bo_1} or V_{bo_2} respectively, reaching 90% of their final value.		11		µs
$t_{discharge}$	15, 16	Gated Detector Discharge Time	V_{CI} = 2V, pk–pk,S1 & S2 = Closed. With LP1 charged, measure the time from Pin 22 going from 0.3V to 2V, to the voltage at V_{bo_1} or V_{bo_2} reaching 90% of their final value.		56		µs
t_{ON}	18, 19	Gated Detector Turn On Time	V_{CI} = 1V DC, $V_{PIN\ 22}$ = 0.3V, S1 & S2 = Open. With LP1 discharged, measure the time from Pin 17 going from 2V to 0.3V, to the voltage on Pin 18 increasing 0.1V Do a similar measurement with LP2, Pin 20 and Pin 19.		8		ns

AC Electrical Characteristics Over Recommended Operating Temperature and Supply Range Refer to AC Test Setup. f = 2.5 MHz unless otherwise indicated. (Continued)

Symbol	Pins	Parameter	Conditions	Min	Typ	Max	Units
t_{OFF}	18, 19	Gated Detector Turn Off Time	V_{CI} = 1V DC, $V_{PIN\ 22}$ = 2V, S1 & S2 = Open. Measure the time from Pin 17 going from 0.3V to 2V, to the voltage on Pin 18 decreasing by 0.1V. Do a similar measurement with Pins 20 and 19.		10		ns
DP8468B-2 t_{PP}	12	Pulse Pairing (Note 7)	f = 2.5 MHz and f = 3.33 MHz V_{AI} = 60 mVpp differential.			±3	ns
DP8468B-3 t_{PP}	12	Pulse Pairing (Note 7)	f = 2.5 MHz and f = 3.33 MHz V_{AI} = 60 mVpp differential.			±5	ns

Note 1: The temperature coefficient of the input impedance is typically 0.05% per degree C.

Note 2: The AGC Threshold is defined as the voltage across the Channel Input (pins 4 and 1) when the voltage on C_{AGC} (pin 14) is 4.2V.

Note 3: The Set Hysteresis Threshold is defined as the minimum differential AC signal across the Channel Input (pins 4 and 1) which causes the voltage on the Channel Alignment Output (pin 11) to change state.

Note 4: To prevent inductive coupling from the digital outputs to Amp In, the TTL outputs should not drive more than one ALS TTL load each. Pin 11 is an open collector output which is tested with an external 1k pullup resistor to the 5V supply.

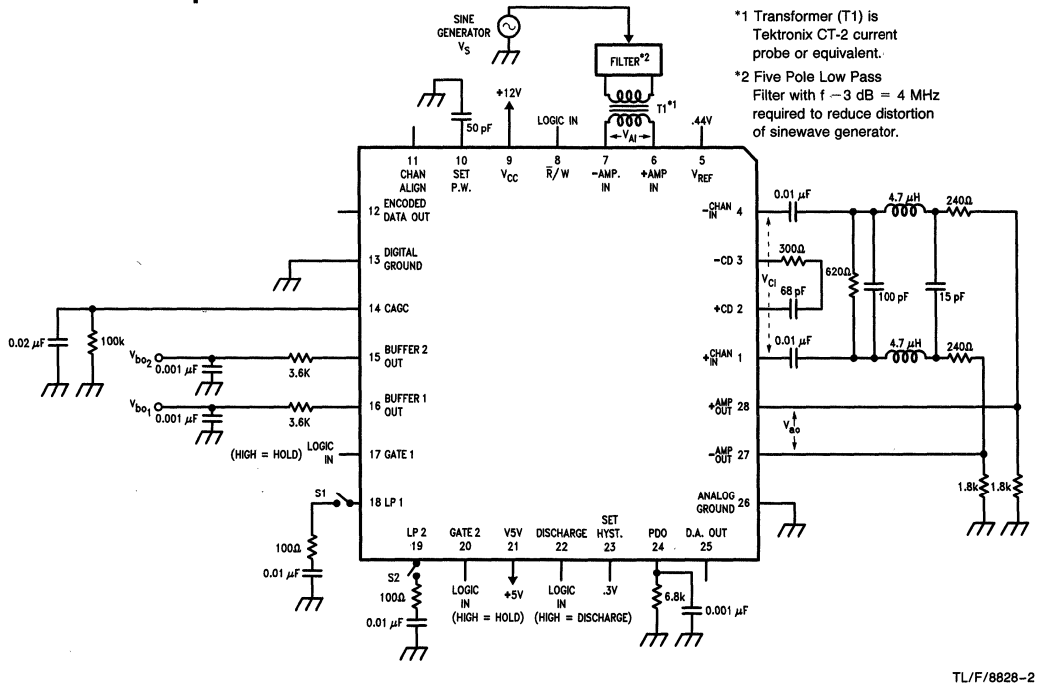
Note 5: $|V_{PIN\ 16} - V_{PIN\ 15}|_{eighth}$ = The difference in the buffer output voltages with the channel input level set to simulate the read head mispositioned by one eighth of a track, This is done by setting V_{CI} = 1.25V and measuring $V_{PIN\ 15}$ then set V_{CI} = 0.75V and measure $V_{PIN\ 16}$. The absolute value of the difference between Pins 15 and 16 is the quantity of interest. The part is also tested with Pin 16 measured at a V_{CI} = 1.25V and Pin 15 measured with V_{CI} = 0.75V.

$V_{PIN\ 17} = V_{PIN\ 18}$ = 0.3V

Note 6: $|V_{PIN\ 16} - V_{PIN\ 15}|_{quarter}$ = The difference in the buffer output voltages with the channel input level set to simulate the read head mispositioned by one quarter of a track, This is done by setting V_{CI} = 1.5V and measuring $V_{PIN\ 15}$ then set V_{CI} = 0.5V and measure $V_{PIN\ 16}$. The absolute value of the difference between Pins 15 and 16 is the quantity of interest. The part is also tested with Pin 16 measured at a V_{CI} = 1.5V and Pin 15 measured with V_{CI} = 0.5V.
$V_{PIN\ 17} = V_{PIN\ 18}$ = 0.3V

Note 7: For this Preliminary specification only, pulse pairing is measured and guaranteed at 25°C and 2.5 MHz.

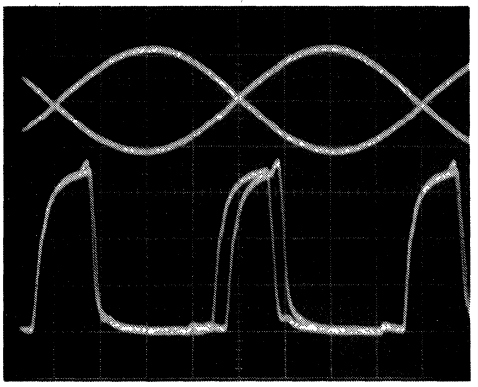
AC Test Set Up



TL/F/8828–2

AC Test Set Up (Continued)

The channel filter is a 3 pole Bessel with the corner frequency at 7.5 MHz which is similar to filters used in 10 Mbits/sec 2,7 drives.



TL/F/8828–3

Pulse Pairing Measurement

Connect a scope probe to pin 12 (Encoded Data Out) and trigger off its positive edge. Adjust the trigger holdoff so the scope first triggers off the pulse associated with the positive peak and then off the pulse associated with the negative peak (as shown in the scope photo below). Pulse pairing is displayed on the second pair of pulses on the display. If the second pulses are separated by 2 ns, then the pulse pairing for this part is ±1 ns.

Chip Operation

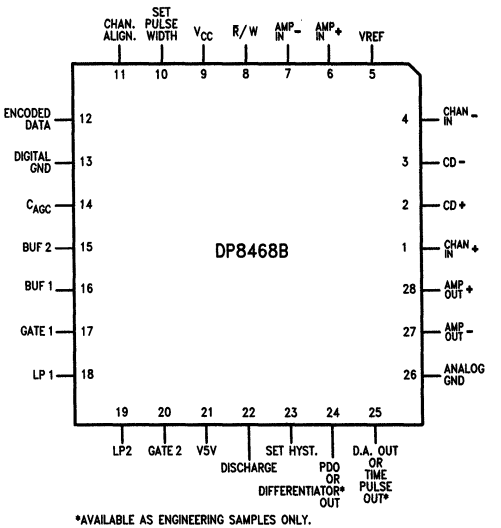
The output from the read/write amplifier is AC coupled to the Amp Input of the DP8468B. The amplifier's output voltage is fed back via an external filter to an internal fullwave rectifier and compared against the external voltage on the V_{REF} pin. The AGC circuit adjusts the gain of the amplifier to make the peak to peak differential voltage on the Gate Channel Input four times the DC voltage on V_{REF}. Typically the signal on Amp Out will be set for 4 Vpp differential. Since the filter usually has a 6 dB loss, the signal on the Gate Channel Input will be 2 Vpp differential. The user should therefore set 0.5V on V_{REF} which can be done with a simple voltage divider from the +12V supply.

The peak detection is performed by feeding the output of the Amplifier through an external filter to the differentiator. The differentiator output changes state when the input pulse changes direction, generally this will be at the peaks. However, if the signal exhibits shouldering (the tendency to return to the baseline), the differentiator will also respond to noise near the baseline. To avoid this problem, the signal is also fed to a gating channel which is used to define a level either side of the baseline. This gating channel is comprised of a differential comparator with hysteresis and a D flip-flop. The hysteresis for this comparator is externally set via the Set Hysteresis pin. In order to have data out, the input amplitude must first cross the hysteresis level which will change the logic level on the D input of the flip-flop. The peak of the input signal will generate a pulse out of the differentiator and bi-directional one shot. This pulse will clock the new data at the D input through to the output. In this way, when the differentiator is responding to noise at the baseline, the output of the D flop is not changing since the logic level into the D input has not changed. The comparator circuitry is therefore a gating channel which prevents any noise near the baseline from contaminating the data. The amount of hysteresis is twice the DC voltage on the Set Hysteresis pin. For instance, if the voltage on the Set Hysteresis pin is 0.3V, the differential AC signal across the Gate Channel Input must be larger than 0.6V before the output of the comparator will change states. In this case, the hysteresis is 30% of a 2V peak to peak differential signal at the gate channel input.

The signal at the Channel Input pins #1 and #4 is amplified and fed to a full wave rectifier. This full wave rectified signal is then presented to the inputs of two separately gated detectors which, when gated on, will generate a DC voltage level proportionate to the peak level of the full wave rectified signal. When recovering embedded servo information each channel is independently gated, thereby detecting a signal whose amplitude represents the relative positioning of the read head. The difference of the DC voltages from the gated detectors is the head positioning information. The DC voltages appear across external RC networks connected to pins 18 and 19. The output buffers level shift these DC voltages which then appear at the low impedance output pins 15 and 16. The voltages at these pins are differenced and appear at the low impedance output pin 25 centered about the external reference voltage applied to pin 21.

Connection Diagram

Plastic Chip Carrier (PCC) Package



*AVAILABLE AS ENGINEERING SAMPLES ONLY.

TL/F/8828–4

Plastic Chip Carrier (V)
Order Number DP8468BV-3 or DP8468BV-2
See NS Package Number V28A

Pin Definitions

Pin #	Name	Function
POWER SUPPLY		
9	V_{CC}	The supply is +12V ±10%.
21	V5V	Supplies internal chip logic and provides a reference voltage for the zero level of the difference amplifier output on pin 25. Supply tolerance is 5V ±5%.

Pin Definitions (Continued)

Pin #	Name	Function
POWER SUPPLY (Continued)		
26	Analog Ground	Analog signals should be referenced to this pin.
13	Digital Ground	Digital signals should be referenced to this pin.
ANALOG SIGNALS		
6 7	Amp. in+ Amp. in−	These are the differential inputs to the Amplifier. The output of the read/write head amplifier should be capacitively coupled to these pins.
28 27	Amp. out+ Amp. out−	These are the differential outputs of the Amplifier. These outputs should be capacitively coupled to the channel filter.
4 1	Channel Inputs	These are the differential inputs to the time, gating and servo channels. These inputs must be capacitively coupled to the channel filter at the amp. outputs.
2 3	C_d+ C_d−	The external differentiator network is connected between these two pins.
23	Set Hysteresis	The DC voltage on this pin sets the amount of hysteresis on the differential comparator.
24	PDO	This is a Peak Detector Output signal that is used in conjunction with the set hysteresis pin 23 to provide a dynamic hysteresis function.
5	V_{REF}	The AGC circuit adjusts the gain of the gain controlled amplifier to make the differential peak to peak voltage at the Channel Inputs equal to four times the DC voltage on this pin.
14	C_{AGC}	The external capacitor for the AGC is connected between this pin and Analog Ground.
18 19	LP1 LP2	The peak detected servo signal voltage appears across the RC networks connected from these pins to analog ground.
16 15	Buffer Out 1 Buffer Out 2	These low impedance pins, output the DC level at pins 18 and 19 respectively, level shifted down by two diode drops.
25	D.A. Out	This low impedance pin outputs the difference in voltage between pins 16 and 15 about a zero level set by the voltage on pin 21.
DIGITAL SIGNALS		
10	Set Pulse Width	An external capacitor to control the pulse width of the Encoded Data Out is connected between this pin and Digital Ground.
8	$\overline{\text{READ}}$/ WRITE	If this pin is low, the Pulse Detector is in the read mode and the chip is active. When this pin goes high, the pulse detector is forced into a stand-by mode. This is a standard TTL input.

Pin #	Name	Function
DIGITAL SIGNALS (Continued)		
11	Channel Alignment Output	This is the buffered, open collector, output of the differential comparator with hysteresis.
12	Encoded Data Out	This is the standard TTL output whose leading edge indicates the time position of the peaks.
17 20	Gate 1 Gate 2	These inputs accept TTL levels. When a low level is present the embedded servo signal is allowed to charge the RC network at pins 18 and 19 respectively. A high level will force a hold condition of the DC voltage across the RC network and will also disable the servo channel.
22	Discharge	This input accepts a TTL level. A high level connects a 1.5k internal resistor to ground on pins 18 and 19.

SPECIAL ENGINEERING PIN OUT AVAILABLE

On an engineering basis only, the Differentiator Output and the output of the time channel bi-directional one shot (referred to from now on as 'Time Pulse Output') will be brought out as pins. The Differentiator and Time Pulse Output pins will not be available on production parts. They are only available by special request on an engineering basis.

In order to bring out these pins it is necessary to eliminate two other pins. The pin trade off and operation details are as follows:

1. The Differentiator Output replaces the Peak Detector Output (PDO), pin 24. The Differentiator Output is buffered by an emitter follower which has a 3k resistor in series with the emitter connected to the output pin. This is shown in *Figure 12*. An external resistor to ground must be connected to this pin in order to bias the output emitter follower. The combination of the 3 kΩ output resistor and the external resistor pull down, form a voltage divider that attenuates the level of the differentiator output signal. Please note that the differentiator signal will only be linear near the differentiator output zero crossing because the signal peaks at the differentiator output are clamped by the Schottky diodes across the collectors of Q2 and Q3 as shown in *Figure 12*.
2. The Time Pulse Output replaces the Difference Amplifier Output (D.A. Out), pin 25. This pin is a standard TTL output capable of driving one ALS load.

DIFFERENCES BETWEEN THE DP8468B AND THE DP8464B

The DP8468B is a DP8464B type pulse detector in combination with two gated detectors which are used to detect embedded servo information in a Winchester disk drive. In order to fit into a 28-pin PCC package and provide the additional embedded servo detection functions, some of the pins on the DP8464B were eliminated. Other changes were made to reduce power dissipation. A summary of the differences between the two parts is given here for the benefit of those who are familiar with the DP8464B.

1. The Time Channel Inputs are now internally connected to the Gate Channel Inputs.

Pin Definitions (Continued)

2. The Time Pulse Out pin is now internally connected to the Time Pulse In pin, and not normally brought out as an output.
3. The Channel Alignment Output is now open collector and requires an external pull up resistor for use.
4. The internal logic is powered from an external 5V supply connected to pin 21.
5. The output impedance that drives C_{AGC}, pin 14, has been reduced from 700Ω to 350Ω. This allows you to double the external capacitance on this pin and still achieve the same attack time as with the DP8464B. The large capacitor allows for longer AGC hold times, which is useful during the embedded servo sectors.
6. The internal leakage current from pin 14 to ground has been reduced by a factor of 5.
7. The combined differential input impedance of the Gate Channel and the Time Channel has been increased from 1.67 kΩ to 2.5 kΩ.
8. The internal current sources on the Amp. Output pins have been eliminated. The current in the output emitter followers is now entirely set by the external pull down resistors on pins 27 and 28.
9. In addition to the embedded servo circuitry, a Dynamic Hysteresis function has been added. The hysteresis level can be set as before or the set hysteresis pin can be connected through an external RC network to the PDO output, pin 24, to implement the Dynamic Hysteresis function.
10. The gain controlled amplifier input impedance in read mode has been increased to 2 kΩ.

Application Information

GENERAL DESCRIPTION OF PULSE DETECTION

The DP8468B Disk Pulse Detector utilizes analog and digital circuitry to detect amplitude peaks of the signal received from the Read/Write Amplifier. The analog signal from a disk is a series of pulses, the peaks of which correspond to 1's or flux reversals on the magnetic media. The pulse detector must accurately determine the time position of these peaks. The peaks are indicated by the positive leading edge of a TTL compatible output pulse. This task is complicated by variable pulse amplitudes depending on the media type, head position, head type and read/write amplifier circuit gain. Additionally, as the bit density on the disk increases, the amplitude decreases and significant bit interaction occurs resulting in pulse distortion and shifting of the peaks.

The graph in *Figure 2* shows how the pulse amplitude varies with the number of flux reversals per inch (or recording density) for a given head disk system. The predominant disk applications are associated with the first two regions on this graph, Regions 1 and 2. Typical waveforms received by the pulse detector for these regions are shown next to the graph.

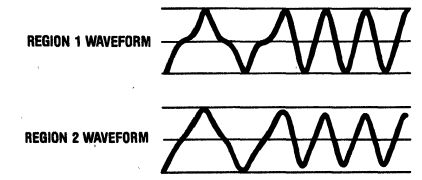
Region 1 is the high resolution area characterized by a large spread between flux reversals and a definite return to baseline (no signal) between these peaks. Pulses of this type are predominantly found in drives which use either thin film heads or plated media, or in drives which utilize run length limited codes (like the 2,7 code) which spread the distance between flux reversals.

A Region 2 waveform will vary from a tendency to return to the baseline (called shouldering) to almost sinusoidal at the higher frequencies. These pulses come from drives which use limited frequency codes (such as MFM). The pulses may contain shouldering on the outer tracks of the disk and be nearly sinusoidal on the inner tracks since the flux density increases towards the inner track.

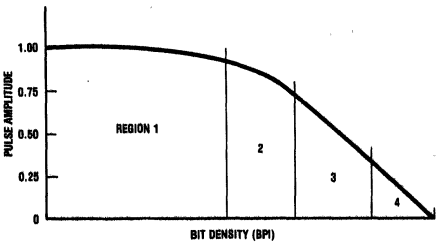
Detecting pulse peaks of waveforms of such variable characteristics requires a means of separating both noise and shouldering-caused errors from the true peaks. In the past, mild shoulder-caused errors were blocked by self-gating circuits (such as the "de-snaker"). These circuits fail when shouldering is extensive, hence the need for the DP8468B which includes a peak sensing circuit and an amplitude sensitive gating channel in parallel.

The main circuit blocks of the DP8468B are shown in *Figure 1*. The output from the read/write amplifier is fed directly to the Amp Input of the DP8468B. This is the input of a Gain Controlled Amplifier. The amplifier's output voltage is fed back via an external filter to an internal fullwave rectifier and compared against the external voltage on the V_{REF} pin. The AGC circuit adjusts the gain of the amplifier to make the peak-to-peak differential Channel input voltage four times the DC voltage on V_{REF}.

The peak detection is performed by feeding the output of the Gain Controlled Amplifier through an external filter to



TL/F/8828–5



TL/F/8828–6

FIGURE 2. Pulse Amplitude vs. Bit Density with Typical Waveforms

Application Information (Continued)

the differentiator. The differentiator output changes state when the input pulse changes direction, generally this will be at the peaks. However, if the signal exhibits shouldering (the tendency to return to the baseline) as seen in Region 1 and the upper part of Region 2, the differentiator will also respond to noise near the baseline. To avoid this, the signal is also fed to a gating channel which is used to define a level either side of the baseline. This gating channel comprises a differential comparator with hysteresis and a D flip-flop. The hysteresis for this comparator is externally set via the Set Hysteresis pin. In order to have valid data out, the input amplitude must first cross the hysteresis level. This will change the logic level on the D input of the flip-flop. The peak of the input signal will generate a pulse out of the differentiator and bi-directional one shot. This pulse will clock in the new data on the D input, which will appear at the Q output. In this way, when the differentiator is responding to noise at the baseline, the output of the D flop is not changing since the logic level into the D input has not yet changed. The comparator circuitry is therefore a gating channel to prevent any noise near the baseline from contaminating the data.

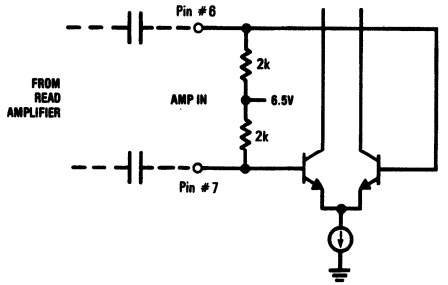
The amount of hysteresis is twice the DC voltage on the Set Hysteresis pin. For instance, if the voltage on the Set Hysteresis pin is 0.3V, the differential Channel Input must be larger than 0.6V (±0.3V) before the output of the comparator will change states. The Encoded Data Output is designed to drive 1 standard TTL gate. The Channel Alignment output is an open collector which requires a pull up resistor, if you want to monitor this point, otherwise this pin can be left floating.

GAIN CONTROLLED AMPLIFIER

The purpose of the Gain Controlled Amplifier is to increase the differential input signal to a fixed amplitude while maintaining the exact shape of the input waveform. The Gain Controlled Amplifier is designed to accept input signals from 20 mVpp to 660 mVpp differential and amplify that signal to 4 Vpp differential. The gain is therefore from 6 to 200 and is controlled by the automatic gain control (AGC) loop. The amplifier output is actually capable of delivering typically 5 Vpp differential output but the parts are only tested and guaranteed to 4 Vpp.

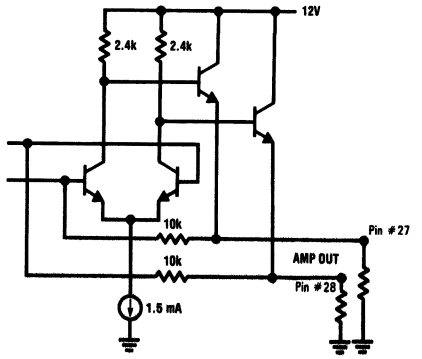
The input to the Gain Controlled Amplifier is shown in *Figure 3*. The value of the input capacitors should be selected so that the pole formed by the coupling capacitor and the 2k bias resistor is a factor of 10 lower than the lowest signal frequency. These input bias resistors have a ±20% tolerance and a temperature coefficient of 0.05% per degree C. When the pulse detector is in the write mode, these bias resistors are automatically shunted by 70Ω resistors. This allows the input circuit to recover quickly from the large transients encountered during a write to read transition. The input impedance to the amplifier is therefore 2k during read operations and 68Ω during write operations.

The output of the Gain Controlled Amplifier is shown in *Figure 4*. The outputs are biased at (12V − (0.75 mA × 2.4k) − 0.75V) or 9.5V. Since each output will swing ±1V (4 Vpp differential), each output pin will swing from 8.5V to 10.5V. If the total differential load placed on the output is 1k, (see *Figure 5*) then the circuit must supply 2V/1k or 2 mA. Since the output is class A, external resistors to ground must be



TL/F/8828–7

FIGURE 3. Input to Gain Controlled Amplifier



TL/F/8828–8

FIGURE 4. Output of Gain Controlled Amplifier

Application Information (Continued)

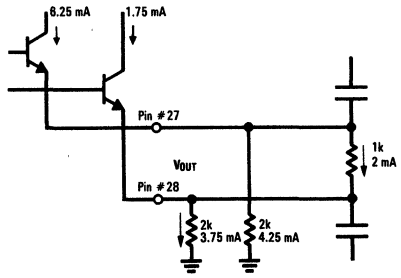
used to provide the sink current. In this case, in order to sink 2 mA at the lowest voltage and provide a 2 mA safety margin, then (8.5V/4 mA) or an external 2k resistor from each output to ground is required. The additional 2 mA margin insures that the output emitter followers never turn off. Typically the output impedance of the Gain Controlled Amplifier is 17Ω, and the −3 dB bandwidth is greater than 20 MHz.

AUTOMATIC GAIN CONTROL (AGC)

The Automatic Gain Control holds the signal level at the Channel Input at a constant level by controlling the gain of the Gain Controlled Amplifier. This is necessary because the amplitude of the input signal will vary with track location, variations in the magnetic film, and differences in the actual recording amplitude. The Gain Controlled Amplifier is designed for a maximum 4 Vpp differential output. To prevent the Gain Controlled Amplifier from saturating, the V_{REF} level must be set so the maximum amplifier output voltage is 4 Vpp. The AGC will force the differential peak-to-peak signal on the Channel Input to be four times the voltage applied to the V_{REF} pin. Normally some kind of filter is connected between the Gain Controlled Amplifier's output and the Channel Input. Typically this filter has a 6 dB insertion loss in its pass band. Since the AGC holds the amplitude at the Channel Input constant, this 6 dB loss through the Channel filter will cause the Gain Controlled Amplifier's output to be 6 dB larger than the Channel Input.

The AGC loop starts out in the high gain mode. When the input signal is larger than expected, the AGC loop will quickly reduce the amplifier gain so the peak-to-peak differential voltage on the Channel Input remains four times the voltage on V_{REF}. If the input amplitude suddenly drops, the AGC loop will slowly increase the amplifier gain until the differential peak-to-peak Channel Input voltage again reaches four times V_{REF}. The AGC loop requires several peaks to react to an increased input signal. In order to recover the exact peak timing during this transition, the V_{OUT} level must be set somewhat lower than the maximum of 4 Vpp. For instance, if the V_{REF} is 0.5V, and if the loss in the channel filter is 6 dB, then the Amp Output is 4 Vpp. If the Amp Input suddenly increases 30%, the amplifier may saturate and the timing for a few peaks may be disturbed until the AGC reduces the amplifier gain. If the peak detection is critical during this time, the system may fail. The proper operation, for this example, is to set the V_{REF} at 0.35V so the amplifier will not saturate if the input suddenly increases 30%.

A simplified circuit of the AGC block is shown in *Figure 6.* When the full wave rectified signal from the Channel Input is greater than V_{REF}, the voltage on the collector of transistor T1 will increase and charge up the external capacitor C_{AGC} through T2. The typical available charging current is 2.5 mA. Conversely, if this input is less than V_{REF}, transistor T2 will be off, so the capacitor C_{AGC} will be discharged by the base current going into the Darlington T3 and T4. This discharge



TL/F/8828–9

FIGURE 5. Output Stage with 1k Differential Load

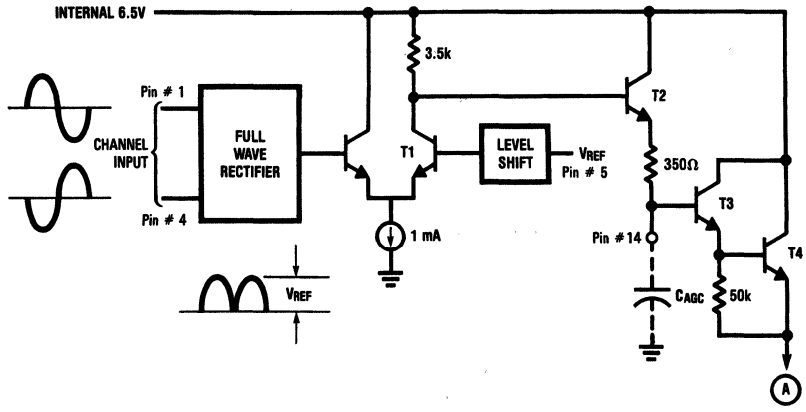


FIGURE 6. Simplified AGC Circuit

TL/F/8828–10

Application Information (Continued)

current is approximately 1 μA. The voltage across C_{AGC} controls the gain of the Gain Controlled Amplifier. This voltage will vary from typically 3.4V at the highest gain to 4.5V at the lowest gain.

When the AGC circuit has not received an input signal for a long time, the base current of the Darlington will discharge the external C_{AGC} to 3.4V. The amplifier will now be at its highest gain. When a large signal comes in, the external C_{AGC} will be charged up with 5 mA from T2 thereby reducing the gain of the amplifier. The formula, $I = C \times (dV/dt)$ can be used to calculate the time required for the amplifier to go from a gain of 200 to a gain of 6. For instance, if $C_{AGC} = 0.01\ \mu F$, the charging current I is 5 mA, and the dV required for the amplifier to go through its gain range is 1.1V, then

$$dt = (0.01\ \mu F \times 1.1V)/(5\ mA) \text{ or } 2.3\ \mu s.$$

In reality, the gain does not change this quickly since the C_{AGC} would only be charging during a portion of the input waveform.

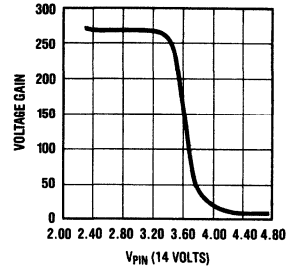
By using the same argument, the time required to increase the amplifier gain after the input has been suddenly reduced can be calculated. This time, the discharging current is only 1 μA so

$$dt = (0.01\ \mu F \times 1.1V)/1\ \mu A) \text{ or } 11\ ms.$$

This time can be decreased by placing an external resistor across the C_{AGC}. For instance, if a 100k resistor is placed in parallel with C_{AGC}, then the discharge current is 40 μA. The time required to increase the amplifier gain is now 40 times faster or 275 μs. If this external resistor is made even smaller, say 10k, then the discharge time will go to 27.5 μs. Now however, there is another problem introduced. The response time of the AGC is so fast that it distorts the signal at the output of the Gain Controlled Amplifier. Distortion of the signal at the Amplifier Output can affect the time position of the peaks of this signal. Be sure to check this distortion over the range of input levels you expect to encounter, when choosing the external R and C values for the AGC.

If the value of the bleed resistor across the C_{AGC} is decreased (in order to equalize the AGC attack and decay times) the value of C_{AGC} must be increased in order to maintain an AGC response that does not distort the signal. There is a second order effect on the amplitude that results from this attack and decay time equalization. Referring to *Figure 1*, notice that the AGC is driven from a full wave rectified version of the Channel Input signal. When the AGC is operated normally (ie. fast attack and slow decay) the voltage that appears across C_{AGC} is the peak detected value of this full wave rectified waveform. However, if you equalize the AGC attack and decay times the voltage across C_{AGC} is the RMS voltage (0.707 times the peak) of the full wave rectified waveform. Thus, the voltage across C_{AGC} is less and the amplitude out of the Gain Controlled Amplifier will consequently be 1.4 times larger.

It is possible to externally drive the C_{AGC} pin to control the gain of the amplifier. When properly filtered, the peak detector output, Pin 24, can be used in this regard. It must be noted that the gain of the amplifier is not always exactly 200 when the voltage on C_{AGC} is 3.4V. The transfer curve between the gain of the amplifier and the voltage on C_{AGC} is only approximate. This transfer curve will vary between parts and with temperature. Care should be taken to prevent the voltage on the C_{AGC} pin from going below ground or above 5.5V. *Figure 7* shows a typical curve of the Gain Controlled Amplifier Gain vs. the voltage across C_{AGC} (Vpin 14.)



TL/F/8828–11

FIGURE 7. Gain Controlled Amplifier Gain vs. Vpin 14

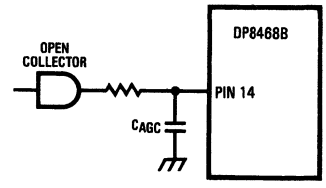
It is possible to change the time constant of the AGC circuit by switching in different external components at the desired times. For instance, as shown in *Figure 8*, an external open collector TTL gate and resistor can be added in parallel with C_{AGC} to decrease the AGC response time. Similarly, an external capacitor could be switched in to increase the response time. Since in the absence of an external resistor the discharge time of C_{AGC} is much longer than the attack time there may be some applications where it is desirable to switch in a parallel resistor to quickly discharge C_{AGC} then switch it out to force a quick attack. Because of the quick attack time, the AGC obtains the proper level quicker than it would had C_{AGC} simply been allowed to discharge to the new level.

There are some applications where it is desirable to hold the AGC level for a period of time. This can be done by raising the $\overline{READ}$/WRITE pin. This will shut off the input circuitry, and it will take time (about 2.5 μs) for the circuit to recover when going back into the read mode.

The AGC must be disabled during the servo sector. This is necessary in order to insure that the AGC does not respond to the servo signal and adjust the signal amplitude to the AGC threshold. The method of raising the $\overline{READ}$/WRITE pin voltage high will not work in this instance as the servo circuitry uses the input amplifier.

Figure 9 shows a method to hold the AGC level while remaining in the read mode (which could be used in embedded servo applications). If the voltage on V_{REF} is raised above 2V, then the amplifier output voltage cannot get large enough to turn on the circuitry to charge up C_{AGC}. For this to work properly, there cannot be a large discharge current path (resistor in parallel with C_{AGC}) across C_{AGC}. The scheme, as shown in *Figure 9*, removes the parallel resistor when the gate output is high.

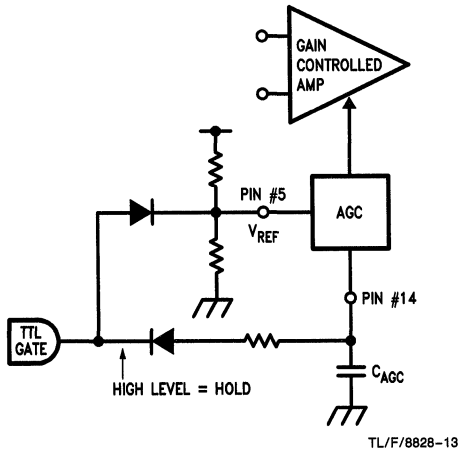
The AGC block can be bypassed altogether by connecting V_{REF} to 3V. In this way, the user can use his own AGC circuit to drive the C_{AGC} pin directly.



TL/F/8828–12

FIGURE 8. Circuit to Decrease AGC Response Time

Application Information (Continued)



TL/F/8828–13

FIGURE 9. AGC Hold Circuit

READ/WRITE

In the normal read mode, the signal from the read/write head amplifier is in the range of 20 mVpp to 660 mVpp. However, when data is being written to the disk, the signal coming into the analog input of the pulse detector will be on the order of 600 mV. Such a large signal will disturb the AGC level and would probably saturate the amplifier. In addition, if a different read/write amplifier is selected, there will be a transient introduced because the offset of the preamplifiers are not matched. A $\overline{\text{READ}}$/WRITE input pin has been provided to minimize these effects to the pulse detector. This is a standard TTL input.

When the $\overline{\text{READ}}$/WRITE pin is low, the pulse detector is in the read mode. When the $\overline{\text{READ}}$/WRITE pin is taken high, three things happen. First, the 1k resistors across the AMP IN pins are shunted by 70Ω resistors, as described previously in the Gain Controlled Amplifier section. Next, the amplifier is squelched so there is no signal on the Amp Output.

Finally, the previous AGC level is held. This AGC hold function is accomplished by not allowing any current to charge up the external C_{AGC}. The voltage across this capacitor will slowly reduce due to the bias current into the Darlington (see *Figure 6*) or through any resistor placed in parallel with C_{AGC}. Therefore, as described in the Automatic Gain Control section, the gain of the amplifier will slowly increase. All of these three events happen simultaneously.

When the $\overline{\text{READ}}$/WRITE input is returned low, the pulse detector will go back to the read mode in a specific sequence. First of all, the input impedance at the Amp In is returned to 1k. Then, after approximately 1 μs, the Gain Controlled Amplifier is taken out of the squelch mode, and finally approximately 1 μs after that, the AGC circuit is turned back on. This return to the read mode is designed to minimize analog transients in order to provide stable operation after 2.5 μs. It is very important that the analog input be stable before the chip is returned to the read mode. It is recommended that other than when writing, the Pulse Detector be in the read mode at all times in order to prevent the 2.5 μs delay from slowing up the system. The $\overline{\text{READ}}$/WRITE pin may be connected to the Write Gate output of a controller (such as the DP8466 Disk Data Controller).

CHANNEL FILTER

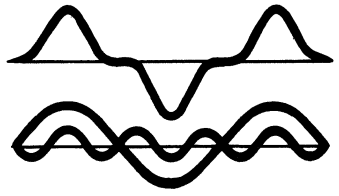
The peak detection is performed by feeding the output of the Gain Controlled Amplifier through an external filter to the differentiator. The differentiator output changes state when the input pulse changes direction, generally this will be at the peaks. The differentiator can also respond to noise near the baseline, in which case the comparator gating channel will inhibit the output. The purpose of the external filter is to bandwidth limit the incoming signal for noise considerations. Care must be used in the design of this filter to ensure the delay is not a function of frequency. For this reason, a high order Bessel filter with its constant group delay characteristics can be used in this application. Often, this filter must be specifically designed to correct errors introduced by the non-ideal phase characteristics of the input read head. The typcial −3 dB point for this filter is around 1.5 times the highest recorded frequency. The design of this filter is complex and will not be discussed here. However, the following discussion does give a feel for some of the considerations involved in the filter design. The reader is referred to reference #3 listed at the end of the Applications Notes for further filter design information.

Figure 10 shows a typical Region 1 waveform where there is no bit interaction. This waveform is primarily the sum of the fundamental frequency and its 3rd harmonic (higher odd harmonics are present when there is more shouldering).

If the filter is to preserve this wave shape (this would be the case if no read/write head phase compensation were necessary) then the phase relationship between the fundamental frequency and its harmonics must not be altered. *Figure 11* shows the output when the 3rd harmonic has the proper magnitude, but the phase relationship is not maintained. The result is that the output waveform is not the same shape as the input (in a severe case it may be almost unrecognizable) and the time position of the peaks has been altered.

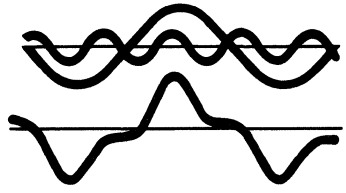
One electrical parameter which describes how well a filter will preserve a wave shape is called group delay. Group delay is defined as the change in phase divided by the change in frequency. If the group delay is constant over the frequencies of interest, then the wave shape will be maintained. An MFM coded signal will contain three basic frequency components for the various digital patterns of data. For instance, a 10 Megabit/sec MFM signal will consist of analog frequencies of 2.5 MHz, 3.33 MHz and 5 MHz. On the outer track the bit density is the lowest and the 5 and 3.33 MHz signals will look sinusoidal while the 2.5 MHz signal will have a tendency to return to the baseline. This returning to the baseline is called shouldering and is illustrated in *Figure 10*. Since this shouldering is rich in 3rd harmonic—the 2.5 MHz signal will have a strong 7.5 MHz component. The 10 Megabit/sec MFM signal will therefore have 2.5 MHz, 3.33 MHz, 5 MHz, and 7.5 MHz components which must be filtered with constant group delay in order to reproduce the original waveform. For example, if the phase shift through the filter at 2.5 MHz is 33.3°, then at 3.33 MHz the phase shift must be 44.3°, at 5 MHz—66.6°, and at 7.5 MHz—99.9°. The group delay $\frac{d\theta}{dt}$ for this case is 13.32°/MHz. This can be better interpreted as a time delay. 33.3° of a 2.5 MHz signal is equivalent to (33.3/360) × (1/2.5 MHz) or 37 ns. Similarly, 66.6° on a 5 MHz signal is (66.6/360) × (1/5 MHz) = 37 ns.

Application Information (Continued)



TL/F/8828–14

FIGURE 10. Typical Region 1 Waveform



TL/F/8828–15

FIGURE 11. Region 1 Waveform with the Incorrect Phase Relationship

The third order Bessel Filter as shown in the 10 Mbit/sec. pulse pairing measurement board on the data sheet is designed for a constant group delay and a −3 dB point of 7.5 MHz. At this frequency the delay through the filter is 35 ns. The Gain Controlled Amplifier of the DP8468 is designed for a group delay of a 7.8 ns for frequencies up to 7.5 MHz. The 7.8 ns delay in the Gain Controlled Amplifier and the 37 ns delay in the Bessel Filter do not introduce any timing error, only a delay of 44.3 ns from the Amp Input to the output of the filter.

DIFFERENTIATOR

A simplified circuit of the first stage of the differentiator is shown in *Figure 12*. The voltages at V3 and V4 are simply two diodes down from V1 and V2. Therefore the voltage across the external differentiator network (C_d in series with R_d) is the differential input voltage V1 − V2. When R_d is zero, the current through C_d is $I = C \times (dV/dt)$ or $C_d \times (dV_{IN}/dt)$. The Q2 collector current is the sum of the 1.8 mA current source plus the current through C_d or

$$1.8\ mA + C_d \times (dV_{IN}/dt).$$

Similarly, the Q3 collector current is

$$1.8\ mA - C_d \times (dV_{IN}/dt).$$

Therefore, the differentiator output voltage, V5 − V6, is

$$1.5k \times 2 \times C_d \times (dV_{IN}/dt).$$

The input is at a peak when V5 − V6 = 0V.

The differentiator network (C_d and R_d) should be selected so the maximum current into the differentiator network is not greater than the minimum current of I1 and I2 over temperature. In the electrical specifications, the minimum current is specified for 1.4 mA (I_{Cd} Current into Pin 1 and 24 that discharges C_d). For example, the highest analog frequency in a 10 Megabit/sec, MFM signal is 5 MHz. Since the AGC loop has forced the input to the differentiator to 2 V_{PP} (which includes the 6 dB loss of the filter), then the voltage across the capacitor (assuming R_d is 0) is:

$$V_{IN} = 1 \times \sin(2 \times \pi \times 5E6 \times t)$$

and

$$dV_{IN}/dt = 1 \times 2 \times \pi \times 5E6 \times \cos(2 \times \pi \times 5E6 \times t)$$

and the maximum slope is

$$(dV_{IN}/dt)max = 1 \times 2 \times \pi \times 5E6 = 314E5\ V/sec.$$

For this example, C_d can now be calculated. Since $I = C \times (dV/dt)$, then for I = 1.4 mA, dV/dt = 314E5, then the maximum C_d must equal 45 pF. From this example, a following simple design equation for the value of C_d can be derived.

$$C_d = 445/(V_{IN} \times f_{max})$$

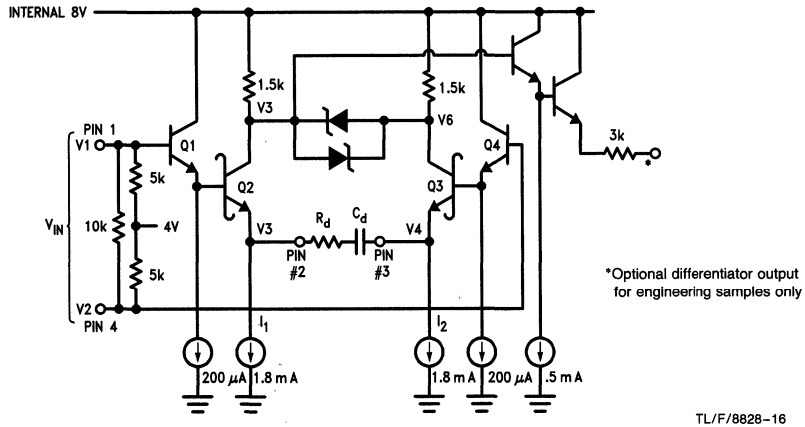
where

C_d is the maximum external differentiator capacitor in pF

V_{IN} is the peak to peak differential Channel input voltage

f_{max} is the maximum analog frequency in MHz

Note that this is the maximum value for the capacitor when the series resistor R_d is zero. The value of the capacitor can



TL/F/8828–16

FIGURE 12. Simplified Differentiator First Stage

Application Information (Continued)

be increased if a series resistor is used, but the maximum current through the differentiator network must not exceed 1.4 mA. If too large a value for C_d is used, the delay through the differentiator will become dependent on frequency. This will not show up in a single frequency test such as a test for pulse pairing.

For the MFM code, the maximum analog frequency is ½ the data rate. For the ½(2,7) code, the maximum analog frequency is ⅓ the data rate. The above sinusoidal analysis is valid as long as the highest frequency on the outer track is nearly sinusoidal. If, however, there is significant shouldering of this signal then the value of C_d should be reduced accordingly.

The following table summarizes the value of C_d to use for a 2 V_{pp} differential signal to the channel input.

Data Rate	Code	Maximum Frequency	C_d
5 mbits/sec	MFM	2.5 MHz	90 pF
5 mbits/sec	2,7	1.6 MHz	140 pF
10 mbits/sec	MFM	5.0 MHz	45 pF
10 mbits/sec	2,7	3.3 MHz	67 pF

As noted above, the value of the capacitor can be increased if a series resistor is used, but the maximum current through the differentiator network must not exceed 1.4 mA.

A resistor is placed in series with C_d in order to bandlimit the differentiator response. This resistor also has an effect on the phase linearity of the differentiator. An ideal differentiator produces an output that is 90 degree phase shifted from the input regardless of the input frequency. The presence of the series resistor produces an output phase shift that is less than 90 degrees and changes with the input frequency. This resistor can be used to correct for frequency related phase problems encountered elsewhere in the read path.

To properly decode the information on the disk, the read channel must determine if there is a peak (or a "1") during a period of time called a detection window. The detection window for MFM and the (2,7) code is

$$1/(2 \times \text{data bit rate}).$$

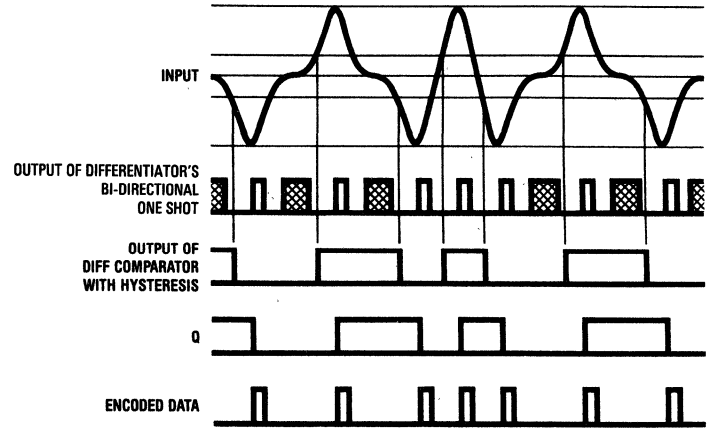
This detection window must accommodate errors in many parts of the system including filters, data separator, and peak shift variations in the data pattern. The pulse pairing of the DP8468 should be included in the error budget calculation.

DIFFERENTIAL COMPARATOR WITH HYSTERESIS

The actual peak detection is done in the time channel with the differentiator. Unfortunately, the differentiator not only responds to signal peaks but also responds to noise at the baseline. In order to prevent this noise from generating false data, the signal at the output of the Gain Controlled Amplifier is also passed through a gating channel which prevents any output change before the input signal has crossed an established level. This gating channel comprises a differential comparator with hysteresis and a D flip-flop. The hysteresis for this comparator is set externally via the Set Hysteresis pin. The amount of hysteresis is twice the voltage on the Set Hysteresis pin. For instance, if the voltage on the Set Hysteresis pin is 0.3V, the differential input signal must be larger than 0.6V(±0.3V) before the output of the comparator will change states. The 0.6V hysteresis represents 30% of a typical 2V differential input signal level to the gating channel. The hysteresis level is usually set between 15% to 40% of the differential input signal.

The operation of the gating channel is shown in *Figure 13*. At the top is a typical Region 1 waveform which exhibits shouldering on the lowest frequency and is almost sinusoidal on the highest frequency. In this example, this waveform is fed to both the timing and the gating channel. The hysteresis level (of about 25%) has been drawn on this waveform. The second waveform is the output of the differentiator and its bi-directional one shot. While there is a positive edge pulse at each peak, there is also noise at the shoulders. This waveform is the clock for the D flip-flop.

The third waveform in *Figure 13* is the output of the Comparator with Hysteresis which goes to the D input of the flip-flop. The true peaks are the first positive edges of the differentiator's bi-directional one shot output which occur after the output of the comparator has changed states. The D flip-flop will "clock" in these valid peaks to the output bi-directional one shot. Therefore, the noise pulses (due to the



TL/F/8828–17

FIGURE 13. Pulse Detector Operation for Region 1 Signals

Application Information (Continued)

differentiator responding to noise at the baseline) just "clock" in the old data through the flip-flop and the output does not change.

The Q output of the flip-flop drives the output bidirectional one-shot which generates the positive edges corresponding to the peaks. The width of the data pulses can be controlled by an external capacitor from the Set Pulse Width pin to ground. This pulse width can be adjusted from 20 ns to ½ the period of the highest frequency. Typical values for this capacitor are 20 pF for a 25 ns pulse width to 100 pF for a 100 ns pulse.

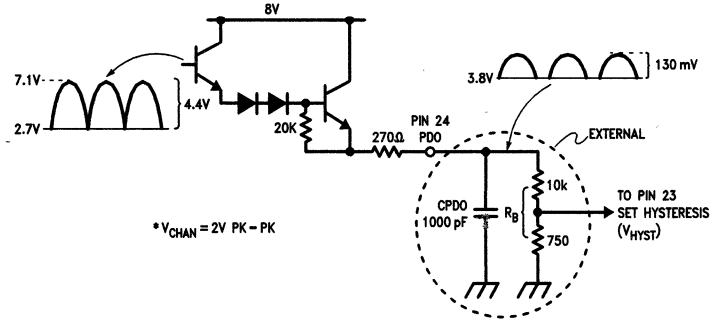
DYNAMIC HYSTERESIS

A dynamically changing hysteresis level (as a function of signal level at the Channel Input pins) can easily be implemented with the DP8468B. An amplified full wave rectified peak detecting output, pin 24, is made available for this function. A resistor capacitor network is connected from pin 24 to pin 23 as shown in *Figure 14*. The RC time constant determines the rate at which the hysteresis level can dynamically change as a function of the signal level out of the Gain Controlled Amplifier.

This time constant can be made to be much faster than the response time of the AGC of the Gain Controlled Amplifier. In this manner the hysteresis level can maintain a nearly constant percentage of the peak amplitude of the signal at the Channel Input Pins.

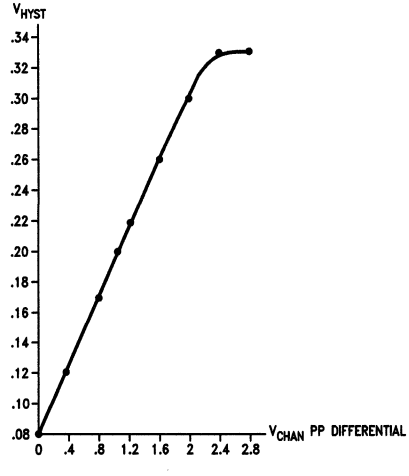
The charge time of the pin 24 capacitor, C_{PDO}, is set by the pin 24 output resistor. R_O, (approx. 270Ω) and the capacitor value. The discharge time is longer since the external bleed resistor, R_B, across the pin 24 capacitor is much larger than R_O. The charge time can be made longer by adding an external resistor in series with pin 24.

Figure 15 shows a plot of the hysteresis level obtained as a function of the peak to peak differential channel input level, pins 1 and 4, for the circuit shown in *Figure 14*. Note that V_{HYST} does not go to zero when there is no signal at the Channel Input. This is because pin 24 has a DC quiescent output level of approximately .8V. Consequently, a 50% reduction in the Channel Input level results in a 36% reduction in the hysteresis level in the linear region of *Figure 15*.



TL/F/8828–18

FIGURE 14. Dynamic Hysteresis Circuitry



TL/F/8828–19

Note: The circuit in *Figure 14* was used to generate this curve.

FIGURE 15. Dynamic Hysteresis Operation

Application Information (Continued)

EMBEDDED SERVO DETECTION SCHEMES

Figure 16 shows the two types of embedded servo patterns the DP8468B is capable of detecting. By far the simplest pattern to detect and to write onto the disk, is the Burst Pattern. Two servo bursts are written on alternate half tracks on the disk surface. These bursts are each generally 10 μs to 30 μs in duration.

When the read head is exactly centered between the two half tracks, the amplitude of each servo burst will be equal and one half the level that would be detected had the read head been exactly centered over the servo burst. If, for example, the read head is not centered but closer to the A burst, then the detected amplitude of the A burst will be proportionately larger than the detected amplitude of the B burst. If the A and B signals are gated into separate peak detectors then the levels of the A and B burst signals are detected. Comparing the relative levels of the resulting DC voltage yields a correction term for repositioning the read/write head.

The two burst scheme is particularly easy to write on the disk surface because the A and B patterns do not overlap. Consequently, an inexpensive servo writer can be used to lay down the servo pattern. The disadvantage of this two burst approach is that because of the physical separation of the A and B burst, an error term is introduced in the relative amplitudes when this pattern is used to detect the head position during a fast seek. For example if the head is moving across the A track in the direction of the B track, then by the time the A burst is detected the head will have moved closer to the B burst, resulting in an error term. This error term may or may not be important depending on the application.

This error term during a fast seek can be eliminated with the 'Interlaced Tri-Bit Pattern', also shown in *Figure 16*. This pattern uses accurately positioned A and B pulses written on alternate half tracks. When the read head is exactly centered between the two half tracks the recovered A and B amplitudes will be exactly equal. If, for example, the read head is positioned closer to the A half track, then the A pulse will be proportionately larger than the B pulse. Because the A and B pulses are so close together in time, a third negative going Sync pulse is required to synchronize

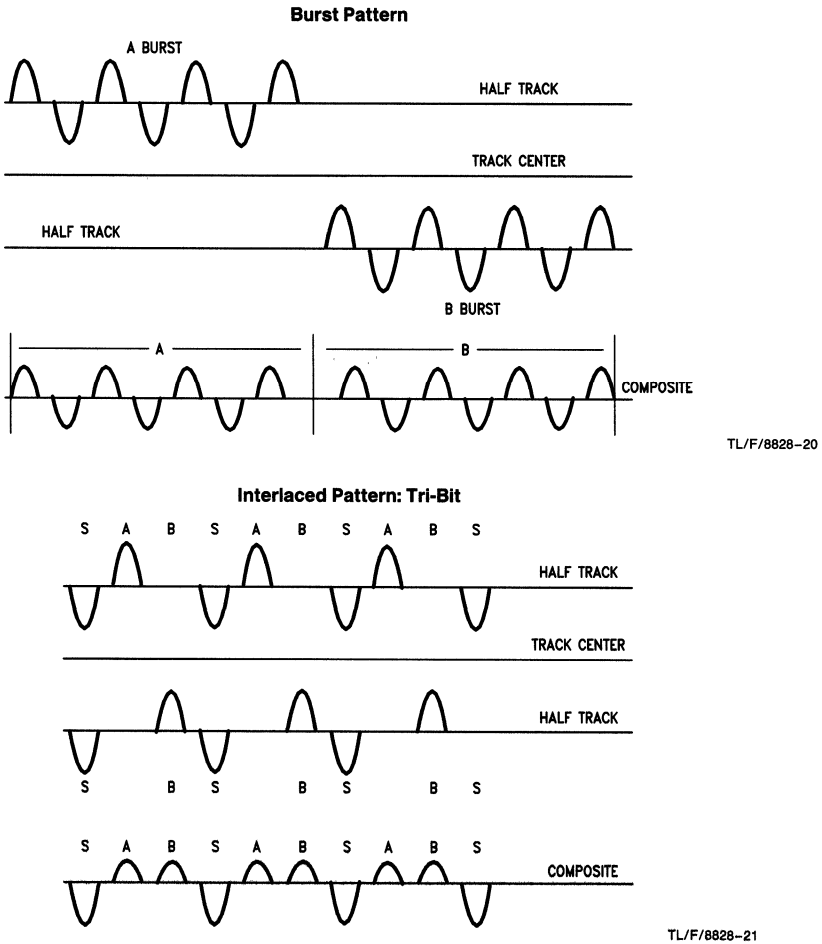


FIGURE 16. Servo Patterns

Application Information (Continued)

the gating of the A and B signals. Without the Sync pulse, there would be no way to gate the A and B pulses into the separate servo channels.

Because of the close proximity of the A and B pulses, the seek error term is eliminated. The price that is paid for this is high, however. Writing this servo pattern on the disk surface requires an expensive and very accurate servo writer. Also, external Sync detection circuitry must be added to the DP8468B. The Channel Alignment Output can be used for this purpose. Once the Sync pulse is detected then gating signals must be generated to control the Gated Detectors on the DP8468B.

GATED SERVO DETECTORS

Figure 17 is a simplified schematic of one of the gated detectors. The gated detectors require TTL gating signals which overlap the duration of the servo burst pattern. The gated detectors are basically peak detectors, but by adding a resistor in series with the peak detector capacitors, Cp1 and Cp2, filtering of the servo signal is accomplished. However, the RC time constant of this series combination should be shorter than the length of the servo burst. This allows the capacitor to fully charge up to the peak of the servo burst. Consequently, the longer the servo burst is, the longer RC time constant can be used and the less sensitive will be the Peak Detectors to noise spikes.

A separate TTL input is provided to implement the discharge of the peak detector capacitors. This discharge drains the capacitors of the charge they acquired during the previous servo sector. When discharge (pin 22) is high, approximately 1 kΩ is connected through two diodes and 1 schottky diode from the peak detector pins, 18 and 19, to ground. The discharge time will then depend on the RC time constant of the sum of the external resistor + 1 kΩ times the external capacitor.

The gated detectors are placed in a hold mode (i.e., gated off) when the gating pins 17 and 20 are high and pin 22 is low. In this mode a small leakage current (less than 1 μA) will slowly discharge the peak detector capacitors.

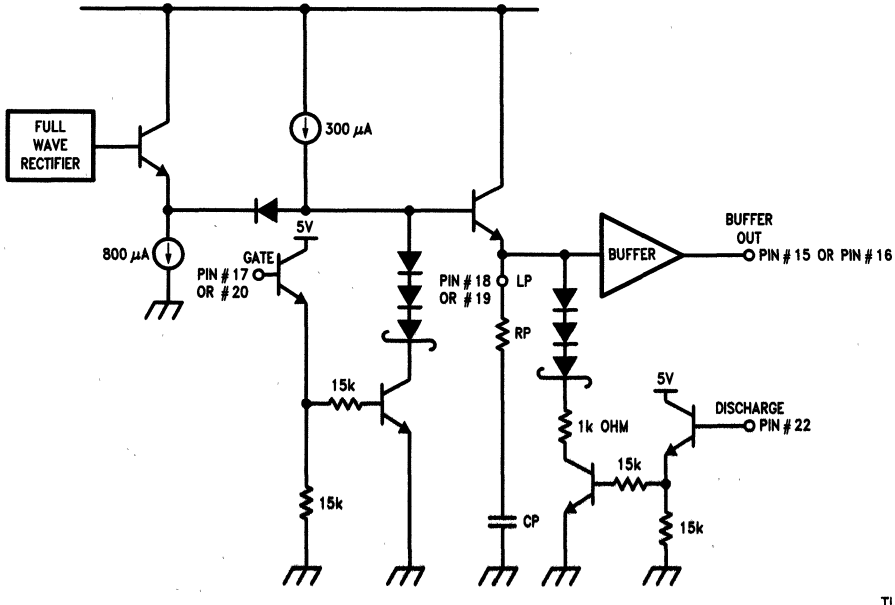
When centered on track, the channel input signal level from the servo burst should equal 1V peak to peak differential. Thus when on track, the servo burst is half the amplitude of the channel input signal in the data field. The channel filter should not be designed to increase the amplitude of the servo signal. The DP8468B has a built in amplifier that will boost the signal level. This amplifier saturates when the channel input signal exceeds 2V peak to peak differential. All that happens if this amplifier saturates is that the recovered DC level at the buffer outputs will not linearly track the amplitude of the channel input signal.

The output of the gated detectors drive two high input impedance unity gain buffers which provide a level shifted low impedance output voltage representation of the voltage across the peak detector capacitors. A simplified schematic of one of the buffers is shown in *Figure 18*. Though the output buffers do some level shifting, they do not level shift all the way down to a ground reference. In fact the minimum output voltage from the output buffers is approximately 1.2V for Vpin21 = 5V.

This minimum output voltage is designed to be a constant percentage of the 5V input level at pin 21 independent of temperature. This minimum output level is important when calculating an error correction term using the expression (A − B)/(A + B) where A and B represent the amplitudes of the servo burst signals. Since the A and B voltages at the buffer outputs do not go to zero, an additional term, C, must be accounted for. The modified expression then becomes:

$$\text{error} = (A - B)/(A + B - 2C)$$

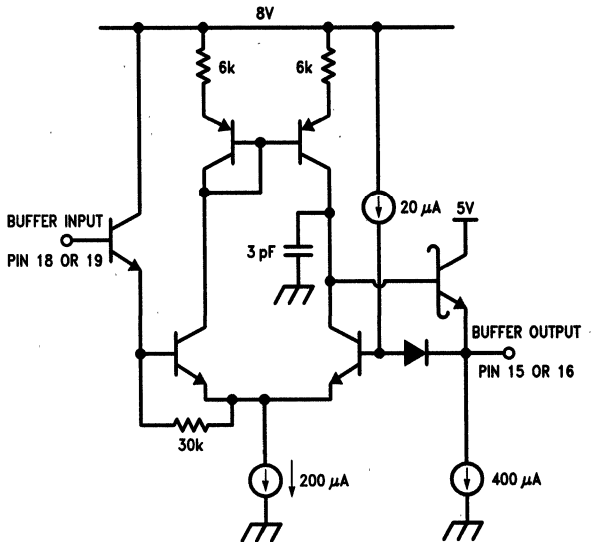
where C = minimum output level of each buffer.



TL/F/8828–22

FIGURE 17. Gated Detector

Application Information (Continued)



TL/F/8828–23

FIGURE 18. Buffer

If C is not accounted for, a gain error will result. As mentioned above, C is designed to be a constant percentage of the voltage applied to pin 21, independent of temperature. Thus,

$$C = K * Vpin\ 21$$

where K = a constant (approx. = 0.22).

Therefore, C will represent a constant number of bits in a system that digitizes the buffer outputs and uses the voltage applied to Vpin21 as an ADC (analog to digital converter) reference.

BUFFER OUTPUTS USED WITH ADC

The output voltage levels from the buffer have been specifically designed to allow them to be directly connected to a muxed A to D converter. Some muxed ADC's allow for a reduced span by tying one of the inputs to a level that will represent all 0's when converted. This can be accomplished with the above scheme by tying the ADC input to a resistor voltage divider from pin 21 to ground with the ratio of the resistors given by:

$$[R1/(R1 + R2)] = K$$

With this technique, the full resolution of the ADC can be utilized. The span of the voltage at the buffer outputs between minimum and maximum detected signals is approximately 3.5V. Because of this large voltage span at the buffer outputs, it may not be necessary to use the above reduced span technique. The 3.5V span allows for better than 7 bits of ADC resolution when digitizing with an 8-bit ADC between 0V and the ADC reference (which should be tied to pin 21).

The output buffers have been designed so that their output levels never exceed the voltage on pin 21. Thus the user need not be concerned that the buffer outputs will exceed the maximum allowed input voltage to the ADC.

The buffers are capable of sourcing up to 3 mA of current and can sink about 300 μA. This sink current can be increased by adding an external pull down resistor from the buffer outputs to ground.

DIFFERENCE AMPLIFIER USE WITH AN ADC

Though muxed ADCs are very common today and often available as part of a microprocessor, there may be some users who do not have muxed inputs. The DP8468B can still be used in these applications by using the difference amplifier output.

A simplified schematic of the difference amplifier is shown in *Figure 19*. When the voltage at the buffer outputs are equal (i.e., on track center) the voltage at the difference amplifier output will equal the voltage on pin 21. The DC transfer function for the difference amplifier output is given below and plotted in *Figure 20:*

$$V_{OUT} = (1/2)\,[A - B] + Vpin\ 21$$

Using the above equation the maximum and minimum voltage for V_{OUT} can be calculated. They are:

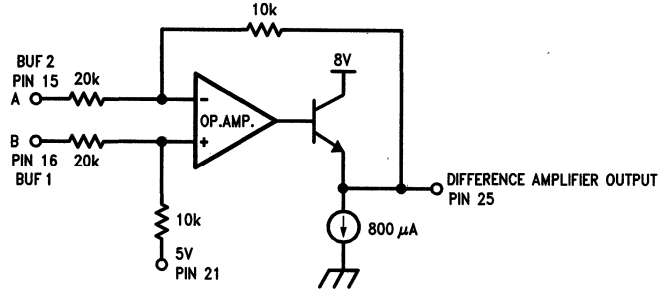
V_{OUT} max. = (1/2) ([(2 * 1.75) + 1.2] − 1.2) + 5 = 6.75V

V_{OUT} min. = (1/2) (1.2 − [(2 * 1.75] + 1.2)] + 5 = 3.25V

Since most ADCs can only digitize voltages below 5V, the difference amplifier's output must some how be level shifted down. The easiest way to accomplish this is to use a resistor voltage divider from the difference amplifier output to ground. In this case a divider ratio of 5/6.75 is required to insure that the center of the divider never goes above 5V. For a symetric span about the on channel voltage (on channel voltage = (5/6.75)*5V = 3.704V) the total output span will now be 2.592V. This is a large enough span to still allow for greater than 7 bits of accuracy from an 8-bit ADC.

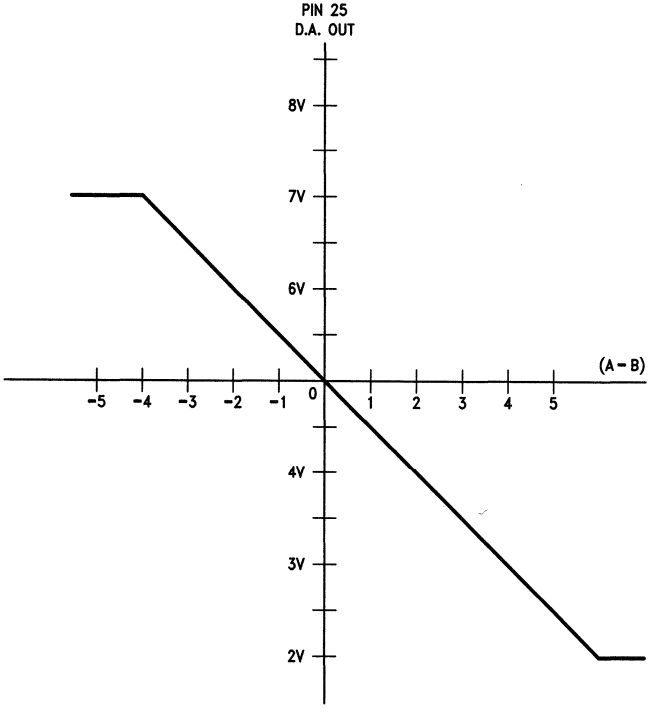
The above technique is only valid if the AGC has been allowed to settle to a known and constant level prior to the servo sector. Also the AGC must be put into a hold mode during the servo sector. If the AGC is allowed to adjust the signal levels to a known and constant level prior to the servo sector then the (A + B − 2C) denominator term is a constant independent of the track. Consequently, this term can be neglected from the gain error calculation and only the one difference output need be digitized.

Application Information (Continued)



TL/F/8828–24

FIGURE 19. Difference Amplifier



TL/F/8828–25

FIGURE 20. Difference Amplifier DC Transfer Curve

Application Information (Continued)

The output stage of the difference amplifier is capable of sourcing about 3 mA of current and can sink about 500 μA. The sink current can be increased by adding an external resistor from the DA output, pin 25, to ground.

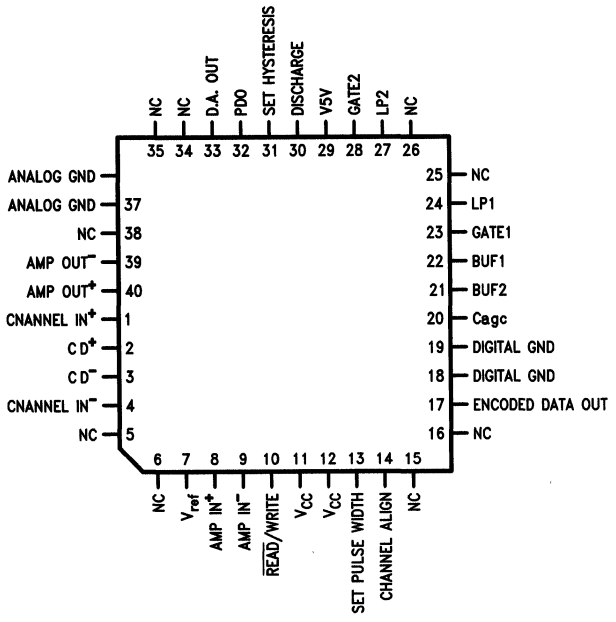
DIFFERENCE AMPLIFIER OUTPUT USED IN A LINEAR FEEDBACK SYSTEM

The difference amplifier provides an accurate difference between the two servo detector voltages. However, this difference is only an accurate representation of the positioning error after both servo channels have been updated. Consequently, the difference output is meaningless during the servo sector and only accurate after the servo sector. A linear system that is providing positioning feedback based on the output from the difference amplifier, will probably require an external sample and hold at the difference amplifier output in order to prevent a false correction during the servo sector. A sample would then be taken after the servo sector when both detectors are gated off.

REFERENCES

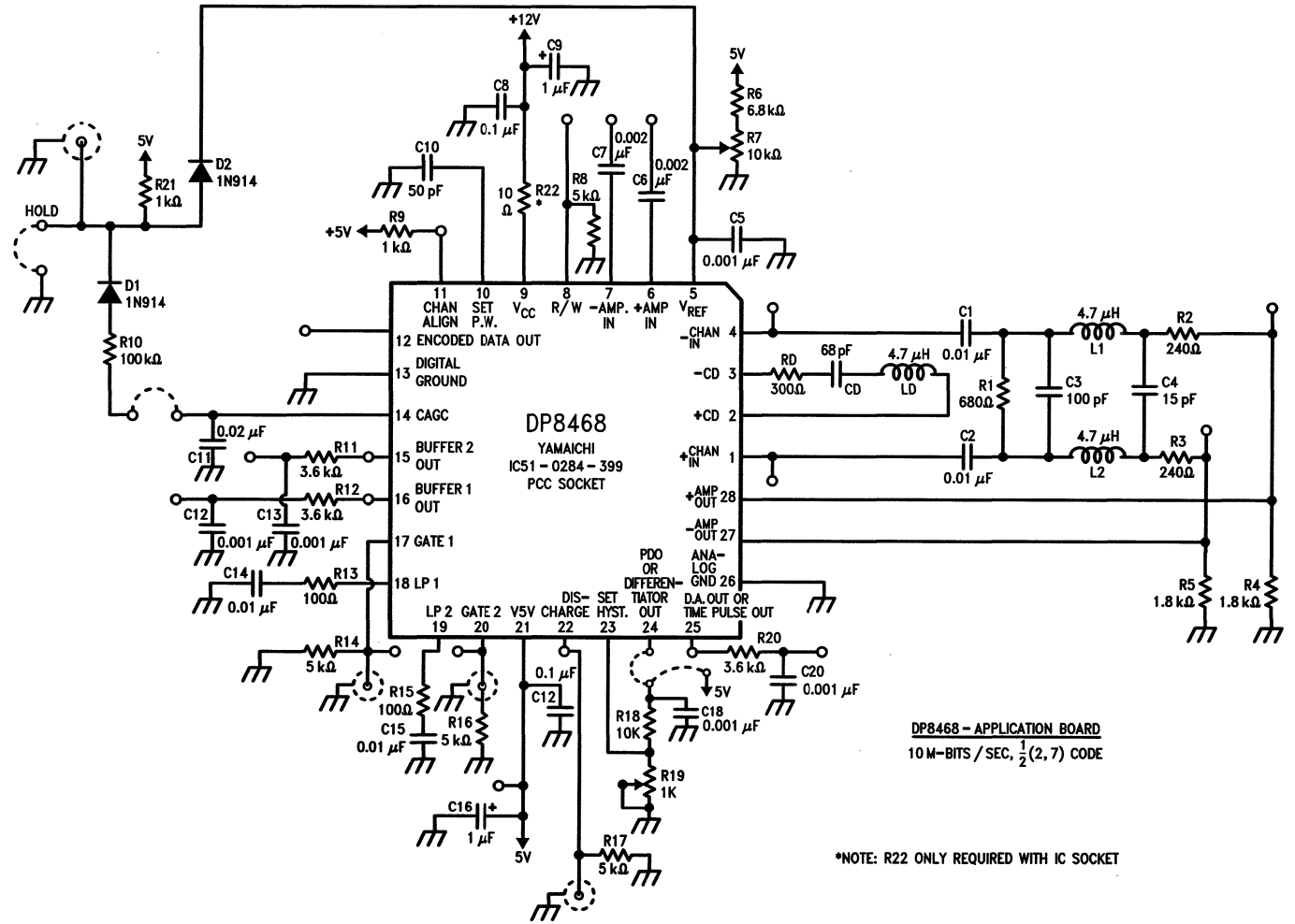
1. I. H. Graham, "Data Detection Methods vs. Head Resolution in Digital Recording", IEEE Transactions on Magnetics Vol. MAG-14, No. 4 (July 1978).
2. I. H. Graham, "Digital Magnetic Recording Circuits", to be published.
3. Anatol I. Zverev, Handbook of Filter Synthesis, John Wiley & Sons publisher, 1967.

TapePak®



TL/F/8828–28

Top View
Order Number DP8468BTP-3 or DP8468BTP-2
See NS Package Number TP40A



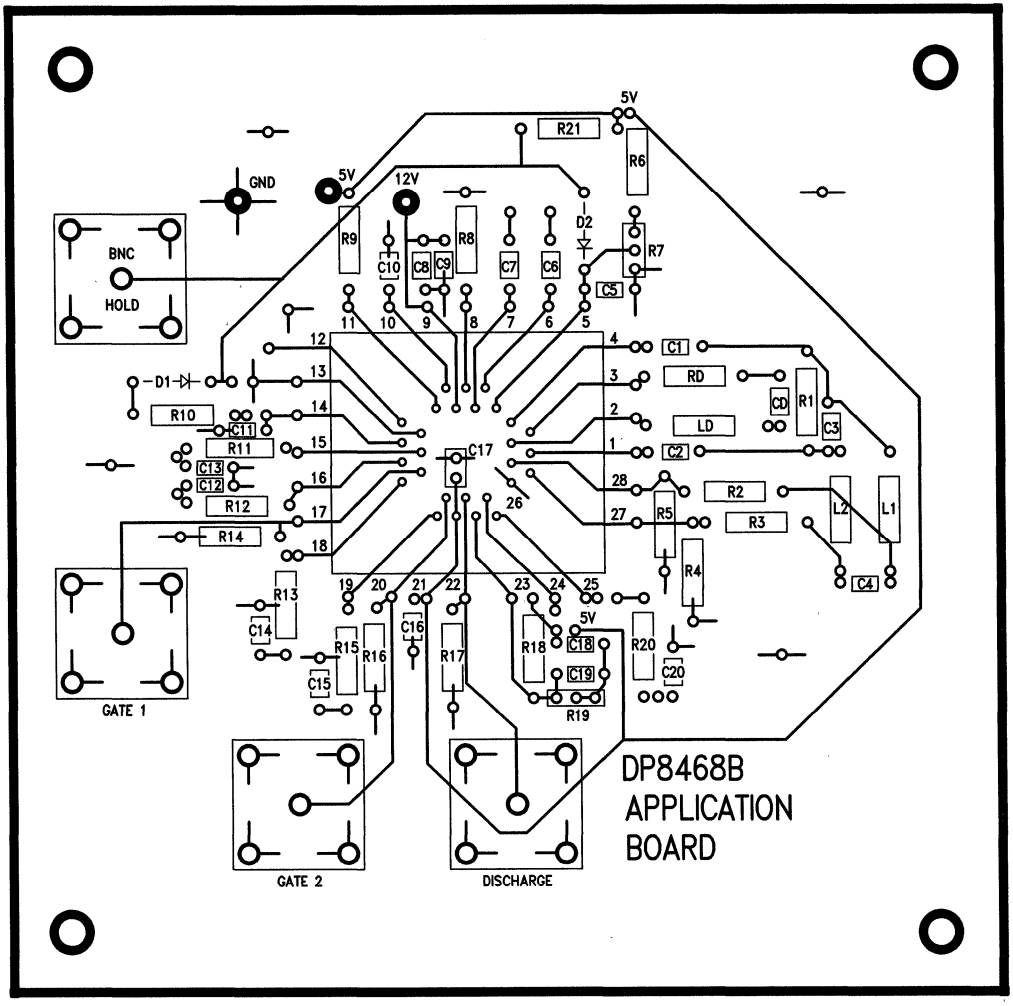
DP8468 – APPLICATION BOARD

10 M-BITS / SEC, $\frac{1}{2}$ (2, 7) CODE

*NOTE: R22 ONLY REQUIRED WITH IC SOCKET

TL/F/8828–26

Component Side



TL/F/8828–27

Section 2

Rigid Disk Data Separators/Synchronizers and ENDECs

Section 2 Contents

National Semiconductor

DP8461/65 Data Separator
DP8451/55 Data Synchronizer

General Description

DP8461/65

The DP8461/65 Data Separators are designed for applications in disk drive memory systems, and depending on system requirements, may be located either in the drive or in the controller. They receive digital pulses from a pulse detector circuit (such as the DP8464 Disk Pulse Detector) if situated in the drive, or from an ST506 type interface if situated in the controller. After locking on to the frequency of these input pulses, they separate them into synchronized data and clock signals. While in the non-read mode, both of these circuits employ a phase-frequency comparator to keep the VCO locked to the 2F input (this signal may be derived from a crystal or a servo track). The DP8465 switches to a phase only comparator when the read mode is entered. The DP8461 continues to use a phase-frequency comparator until the preamble detection circuit has detected two bytes of preamble. This feature thus restricts the DP8461 to use with codes employing the 1010 . . . preamble. MFM, and certain RLL Codes such as 1,7 and 1,8 employ such a preamble. If a Run Length Limited code is used or if the user wishes to do his own data separation, the synchronized data output is available to allow external circuitry to perform the data decoding function.

All of the digital input and output signals are TTL compatible and only a single +5V supply is required. The chip is housed in a narrow 24-pin dual-in-line package (DIP) and is fabricated using Advanced Schottky bipolar analog and digital circuitry. This high speed I.C. process allows the chip to work with data rates up to 20 Mbit/sec. There are two versions of the chip, each having a different decode window error specification. These two versions (−3, −4) are designed to operate from 2 to 20 Mbit/sec and are tested for their respective window tolerances, as specified in the Electrical Characteristics Table.

The DP8461/65 feature a phase-lock-loop (PLL) consisting of a phase-frequency comparator, pulse gate (to allow for phase-only operation in the read mode), charge pump, buffering amplifier, and voltage-controlled-oscillator (VCO). Pins are provided for the user to select the values of the external filtering components required for the VCO, and two current setting resistors for the charge pump. The DP8461/65 have been designed to be capable of locking onto the incoming preamble data pattern within the first two bytes, using an available high rate of charge pump current. Once lock-on has been achieved, the charge pump can be switched to a lower rate (both rates being determined by the external resistors) to improve bit-jitter immunity for the remainder of the read operation. At this time the READ CLOCK OUTPUT switches, without glitching, from half the 2F-CLOCK frequency to half the VCO CLOCK frequency. After lock-on, with soft sectored disks, the MISSING CLOCK DETECTED output indicates when a missing clock occurs so the controller can align byte boundaries to begin deserialization of the incoming data.

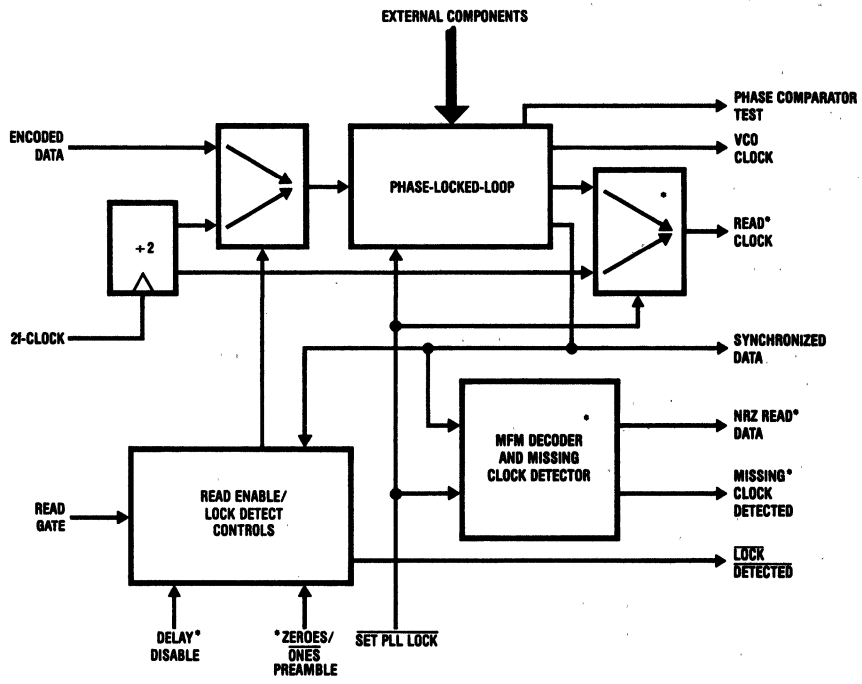
DP8451/55

The DP8451/55 perform the same data synchronization function of the DP8461/65 with no MFM related circuitry. As with the DP8461, the DP8451 continues in the phase-frequency comparison mode until two bytes of preamble are detected. The DP8451/55, which are packaged in 20 pin DIPs or 20-pin PCC's, exclude the READ CLOCK generating circuitry along with the MFM Decoder, Missing Clock Detector, and Read Enable Delay. Users who require only the SYNCHRONIZED DATA OUTPUT and VCO CLOCK OUTPUT can use the DP8451/55 as alternatives to the DP8461/65.

Features

- Operates at data rates up to 20 Mbit/sec
- Phase-Frequency comparison in non-read mode
- Phase-Frequency comparison in preamble—DP8461/51
- Separates MFM data into read clock and serial NRZ data (DP8461/65)
- 4 byte preamble-lock indication capability
- Preamble recognition of MFM encoded "0"s or "1"s
- User-determined PLL loop filter network
- PLL charge pump has two user-determined tracking rates
- External control of track rate switchover
- No glitch on READ CLOCK at switchover (DP8461/65)
- Synchronized data provided as an output (for RLL codes) (all four devices)
- ORed phase comparator outputs for monitoring bit-shift
- Missing clock detected for soft sectored disks
- Less then ½W power consumption
- Standard narrow 24-pin DIP or 28 pin Plastic Chip Carrier Package
- Single +5V supply

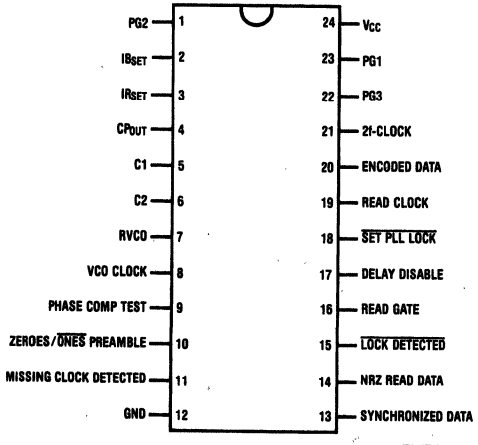
Simplified Block and Connection Diagrams



*Available only on DP8461/65

TL/F/8445–1

DP8461/65
Dual-In-Line Package

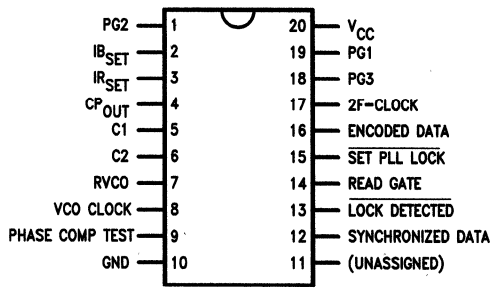


Top View

TL/F/8445–2

Order Number DP8461/65J or N
See NS Package Number J24F or N24C

DP8451/55
Dual-In-Line Package



Top View

TL/F/8445–3

Order Number DP8451/55N
See NS Package Number N20A

Pin Descriptions*

Power Supply

24 V_{CC} +5V ±5%

12 Ground

TTL Level Logic Inputs

16 READ GATE: This is an active high input signal that sets the DP8461/65 Data Separator into the Read Mode.

17 DELAY DISABLE: This input determines the delay from READ GATE going high to the time the DP8461/65 enters the Read Mode. If DELAY DISABLE is set high, this delay is within one cycle of the V_{CO}-CLOCK signal. If DELAY DISABLE is set low, the delay is thirty two-cycles of the VCO CLOCK, as shown in *Figure 1.*

18 $\overline{\text{SET PLL LOCK}}$: This input allows the user to control the on-chip PLL track rate. A high level at this input results in the PLL being in the high track rate. If this input is connected to the $\overline{\text{LOCK DETECTED}}$ output, the PLL will go into the low track rate mode immediately after lock is detected. A low level on this pin is also used to enable the MFM Decoder, the Missing Clock Detector, and to switch the Read Clock Multiplexer from half-2F-CLOCK to half-VCO.

10 ZEROES/$\overline{\text{ONES}}$ PREAMBLE: A high level on this input enables the MFM Decoder circuit to recognize an All Zeros data preamble. A low level results in the recognition of an All Ones data preamble.

20 ENCODED DATA: This input is connected to the output of the head amplifier/pulse-detecting network located in the disk drive. Each positive edge of the ENCODED DATA waveform identifies a change of flux on the disk. In the case of MFM encoded data, the input will be raw MFM.

21 2f-CLOCK: This is a system clock input, which is either a signal generated from the servo track (for systems utilizing servo tracks), or a signal buffered from a crystal. 2f CLOCK MUST ALWAYS BE APPLIED TO THIS INPUT FOR PROPER OPERATION.

TTL Level Logic Outputs

8 VCO CLOCK: This is the output of the on-chip VCO, transmitted from an Advanced Schottky-TTL buffer. It is synchronized to the MFM data output.

15 $\overline{\text{LOCK DETECTED}}$: This output goes active low only after both PLL Lock has occurred and 16 pulses of the preamble pattern have been recognized. It remains low until READ GATE goes inactive.

14 NRZ READ DATA: This is the NRZ (decoded MFM) data output, whose leading edges coincide with the trailing edge of READ CLOCK.

13 SYNCHRONIZED DATA: This output is the same encoded data that is input to the chip, but is synchronous with the negative edge of the VCO CLOCK.

11 MISSING CLOCK DETECTED: When an MFM missing clock is detected, this output will be a single pulse (of width equal to one cycle of READ CLOCK) occurring as shown in *Figure 2.*

19 READ CLOCK: This is half VCO CLOCK frequency when $\overline{\text{SET PLL LOCK}}$ is low; it is half 2f-CLOCK frequency at all other times. A deglitcher is utilized to ensure that no short clock periods occur during either switchover.

9 PHASE COMP TEST: This output is the logical "OR" of the Phase Comparator outputs, and may be used as a bit-shift indicator on for PLL analysis purpose.

Analog Signals

23, 22, PG1, PG3: The external capacitors and resistor of the Pulse Gate filter are connected to these pins. PG1 should be connected directly to the ground pin, pin 12.

1 PG2: This is the Pulse Gate current supply.

3 IRSET: The current into the rate set pin (V_{BE}/R_{RATE}) is used to set the charge pump output current for the low tracking rate.

2 IBSET: The current into the boost set pin (V_{BE}/R_{BOOST}) is used to set the amount by which the charge pump current is increased for the high tracking rate. (I_{INPUT} = I_{RATE} Set + I_{BOOST} Set).

4 CPOUT: CHARGE PUMP OUT/BUFFER AMP IN is available for connection of external filter components for the phase-lock-loop. In addition to being the charge pump output node, this pin is also the noninverting input to the Buffer Amplifier.

7 RVCO: The current at this pin determines the operating currents within the VCO.

5, 6 VCO C1, C2: An external capacitor connected between these pins sets the nominal VCO frequency.

*Pin Number Designations apply only to the DP8461/65. See Connection Diagram for DP8451/55.

Absolute Maximum Ratings

If Military/Aerospace specified devices are required, please contact the National Semiconductor Sales Office/Distributors for availability and specifications.

Supply Voltage	7V
TTL Inputs	7V
Output Voltages	7V
Input Current (CPOUT, IRSET, IBSET, RVCO)	2 mA
Storage Temperature	−65°C to 150°C

Operating Conditions

Symbol	Parameter	Conditions	Min	Typ	Max	Units
V_{CC}	Supply Voltage		4.75	5.00	5.25	V
T_A	Ambient Temperature		0	25	70	°C
I_{OH}	High Logic Level Output Current	VCO Clock Others			−2000 −400	μA
I_{OL}	Low Logic Level Output Current	VCO Clock Others			20 8	mA

Operating Conditions (Continued)

Symbol	Parameter	Conditions	Min	Typ	Max	Units
t_{DATA}	Input Data Rate		2.0		20	Mbit/sec
t_{WCK}	Width of 2f-CLOCK, High or Low		10			ns
t_{WPD}	Width of ENCODED DATA Pulse, (Note 2)	HIGH	5 ns + 0.10t			ns
		LOW	0.4t			
V_{IH}	High Logic Level Input Voltage		2			V
V_{IL}	Low Logic Level Input Voltage				0.8	V
t_{SETUP} (READ Gate)	Min. Amount of Time Which a Positive Edge of READ Gate Must Precede a Negative Edge of a VCO (Pin 8)		20			ns
t_{HOLD} (READ Gate)	Min. Time Required for a Positive Edge of a READ Gate to be Held after a Negative Edge of a VCO (Pin 8)		10			ns

DC Electrical Characteristics Over Recommended Operating Temperature Range

Symbol	Parameter	Conditions	Min	Typ	Max	Units
V_{IC}	Input Clamp Voltage	V_{CC} = Min., I_I = −18 mA			−1.5	V
V_{OH}	High Level Output Voltage	V_{CC} = Min., I_{OH} = Max.	V_{CC}−2V	V_{CC}−1.6V		V
V_{OL}	Low Level Output Voltage	V_{CC} = Min., I_{OL} = Max.			0.5	V
I_{IH}	High Level Input Current	V_{CC} = Max., V_I = 2.7V			20	μA
I_{IL}	Low Level Input Current	V_{CC} = Max., V_I = 0.4V			−200	μA
I_O	Output Drive Current (Note 1)	V_{CC} = Max., V_O = 2.125V	−12		−110	mA
I_{CC}	Supply Current	V_{CC} = Max.			100	mA
I_{OUT}	Charge Pump Output Current	500 μA ≤ I_{RSET} + I_{BSET} ≤ 2000 μA 200 μA ≤ I_{RSET} + I_{BSET} < 500 μA	I_{TYP} − .18 (I_R + I_B) − 30 μA I_{TYP} − .08 (I_R + I_B) − 80 μA	1.95 (I_{RSET} + I_{BSET}) − 70 μA 1.95 (I_{RSET} + I_{BSET}) − 70 μA	I_{TYP} + .18 (I_R + I_B) + 30 μA I_{TYP} + .08 (I_R + I_B) + 80 μA	μA

Note 1: This value has been chosen to produce a current that closely approximates one-half of the true short-circuit output current, I_{OS}.

Note 2: t is defined as the period of the encoded MFM data, or two times the VCO period.

AC Electrical Characteristics Over Recommended V_{CC} and Operating Temperature Range.

(All Parts unless stated otherwise) (t_R = t_F = 2.0 ns, V_{IH} = 3.0V, V_{IL} = 0V)

Symbol	Parameter	Min	Typ	Max	Units
t_{READ}	Positive READ CLOCK transitions from READ GATE set active until PLL Lock sequence begins (DELAY DISABLE low)		16	17	—
t_{READ}	Positive READ CLOCK transitions from READ GATE set active until PLL Lock sequence begins (DELAY DISABLE high)		1	1	—
$t_{DECODE\ NRZ}$	Number of READ CLOCK cycles required to output each decoded MFM data bit (Note 3, 4)	—	2	3	T-clock
$t_{TRANSMIT\ MFM}$	Positive READ CLOCK transitions required to transmit input MFM to output	1	2	3	—

Note: For Further Information Refer to Application Notes AN-414, AN-415, and AN-416.

AC Electrical Characteristics Over Recommended V_{CC} and Operating Temperature Range. (Continued)

(All Parts unless stated otherwise) ($t_R = t_F = 2.0$ ns, $V_{IH} = 3.0V$, $V_{IL} = 0V$)

Symbol	Parameter	Min	Typ	Max	Units
$t_{READ\ ABORT}$	Number of READ CLOCK cycles after READ GATE set low to read operation abort			2	T-clock
t_{WINDOW}	Variance of center of decode window from nominal (Note 7) DP84XX-3 / DP84XX-4			6 / 10	ns
$\phi_{LINEARITY}$	Phase range for charge pump output linearity (Note 2)	$-\pi$		$+\pi$	Radians
K_1	Phase Comparator—Charge Pump gain constant (Note 5) ($N = f_{VCO}/f_{INPUT\ DATA}$, $2 \le N \le 4$ for MFM)		$\frac{1.78V_{BE}}{N2\pi R}$		Amps/rad
$V_{CONTROL}$	Charge pump output voltage swing from nominal		± 100		mV
K_{VCO} ($=A \times K_2$)	VCO gain constant (ω_{VCO} = VCO center frequency in rad/s) (Note 1, 6)	$\frac{1.20\omega_C}{V_{BE}}$	$\frac{1.40\omega_C}{V_{BE}}$	$\frac{1.60\omega_C}{V_{BE}}$	rad/sec. V
f_{VCO}	VCO center frequency variation over temperature and V_{CC}	-5		$+5$	%
$f_{MAX\ VCO}$	VCO maximum frequency		60		MHz
t_{HOLD}	Time READ CLOCK is held low during changeover after lock detection has occurred (Note 3)			1½	T-clock
t_{PHL}	Prop. Delay. VCO Neg. Edge to Synchronized Data Neg. Edge		15	30	ns
t_{PLH}	Prop. Delay. VCO Negative Edge to Synchronized Data Positive Edge		10	25	ns
$t_{2F/RC}$	Delay from 2F positive edge to READ CLOCK positive on negative edge ($\overline{\text{SET PLL LOCK}}$ high)	10		35	ns

Note 1: A sample calculation of frequency variation vs. control voltage: $V_{IN} = \pm 0.1V$;

$$K_{VCO} = \frac{\omega OUT}{V_{IN}} = \frac{0.4\omega_C}{0.2V} = \frac{2.0\omega_C}{V} \frac{(rad/sec)}{(volt)}$$

Note 2: $-\pi$ to $+\pi$ with respect to 2f VCO CLOCK

Note 3: T-clock is defined as the time required for one period of the READ CLOCK to occur.

Note 4: This number remains fixed after PLL Lock occurs.

Note 5: With respect to VCO CLOCK; $I_{PUMP\ OUT} = 1.9\ I_{SET}$

$$I_{SET} = \frac{V_{BE}}{R_{SET}}$$

Note 6: Although specified as the VCO gain constant, this is the gain from the Buffer Amplifier input to the VCO output.

Note 7: This specification is guaranteed only for the conditions under which the parts were tested. However, significant variation from the formula is not expected for other data rates and filters. The filter values below were chosen for operation in an automatic test system (static window) environment. Different criteria may apply for choosing filter values in a disk system. See Loop Filter section for sample calculations of other filter values.

Static Window Margin Test Loop Filter Component Values

Part Type	Data Rate Tested	C_1	C_2	R_1	R_{RATE}	R_{BOOST}
DP8451/55/61/65-4	5 Mbit/Sec	0.02 μF	150 pF	200Ω	750Ω	1.6 kΩ
DP8451/55/61/65-3	10 Mbit/Sec	.082 μF	1600 pF	27Ω	820Ω	619Ω

External Component Selection (All Parts) (Note 1)

Symbol	Component	Min	Typ	Max	Units
R_{VCO}	VCO Frequency Setting Resistor (Note 2)	990		1010	Ω
C_{VCO}	VCO Frequency Setting Capacitor (Note 3, 4)	20		245	pF
R_{RATE}	Charge Pump I_{RATE} Set Resistor (Note 6)	0.4		4.0	kΩ
R_{BOOST}	Charge Pump (High Rate) I_{BOOST} Resistor (Note 6)	0.5		∞	kΩ
C_R	I_{RATE} Bypass Capacitor (Note 5)	.01			μF
C_B	I_{BOOST} Bypass Capacitor (Note 5)	.01			μF

Note 1: External component values for the Loop Filter and Pulse Gate are shown in tables 1 & 2.

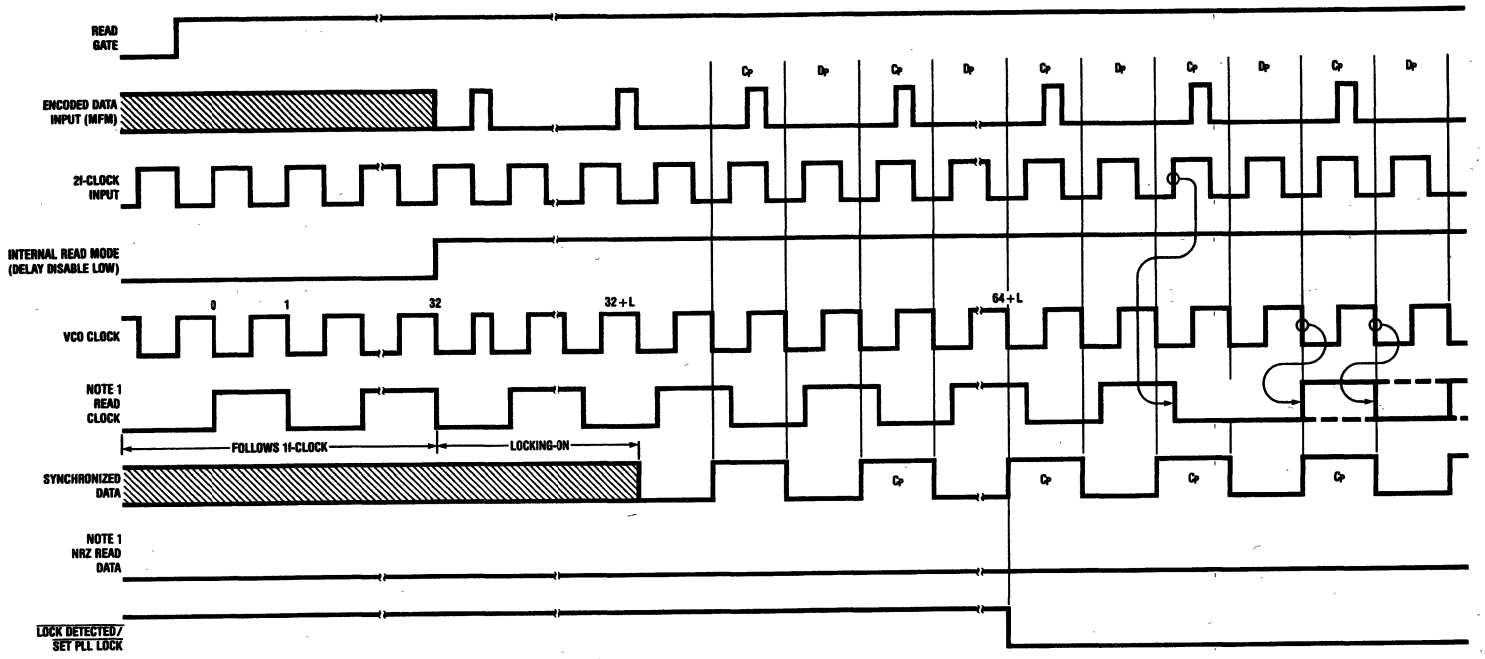
Note 2: A 1% Component Tolerance is Required.

Note 3: These MIN and MAX values correspond to the MAX and MIN data rates respectively.

Note 4: The Component Tolerance is system dependent on how much center frequency deviation can be tolerated.

Note 5: Component Tolerance 15%.

Note 6: The minimum value of the parallel combination of R_{RATE} and R_{BOOST} is 350Ω.



TL/F/8445–4

Note 1: Not included on the DP8451/55.

C_P, D_P = preamble clock and preamble data bits respectively.

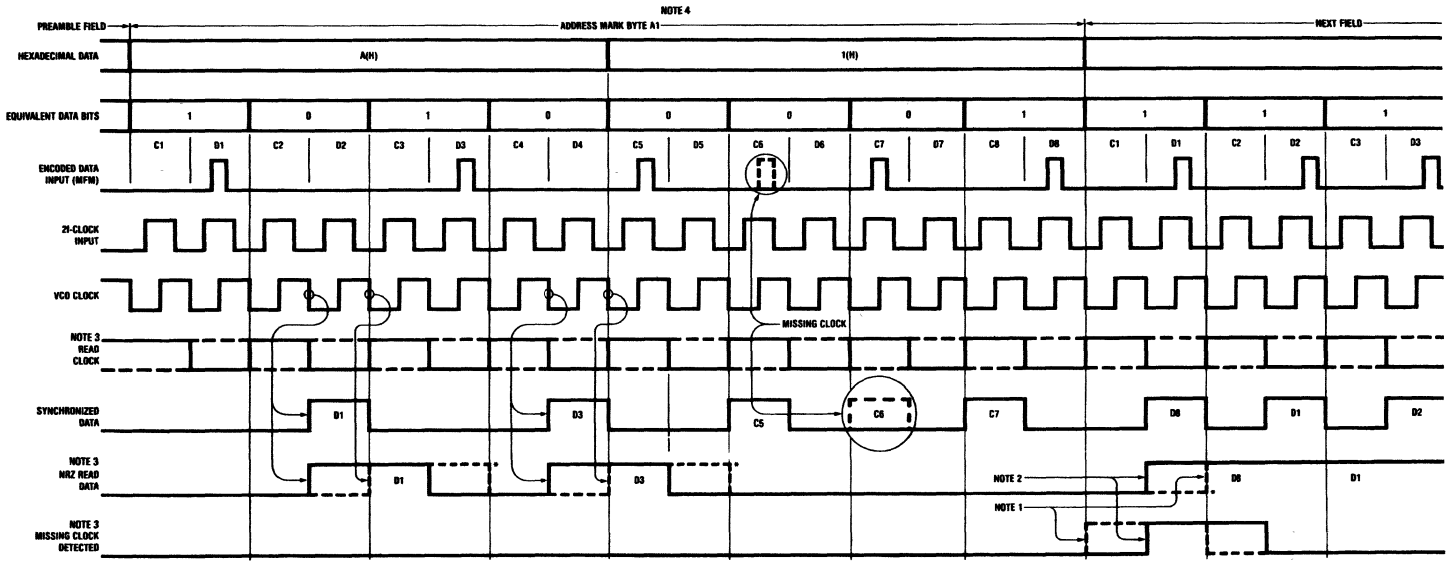
L = Number of 2f-clock cycles required for VCO to lock, determined by external loop filter component values

At 32 + L, VCO has just locked.

At 64 + L, circuit has confirmed lock (has been in lock for 16 MFM clock bits). This sequence shows the MFM all-zeros preamble pattern.

For DP8451/55 delay disable does not exist and part functions as if this input is always high.

FIGURE 1. Lock-on Sequence Waveform Diagram



TL/F/8445–5

* READ CLOCK and NRZ READ DATA may be delayed by one VCO clock period depending on the phase of the internal clock at activation of READ GATE input.

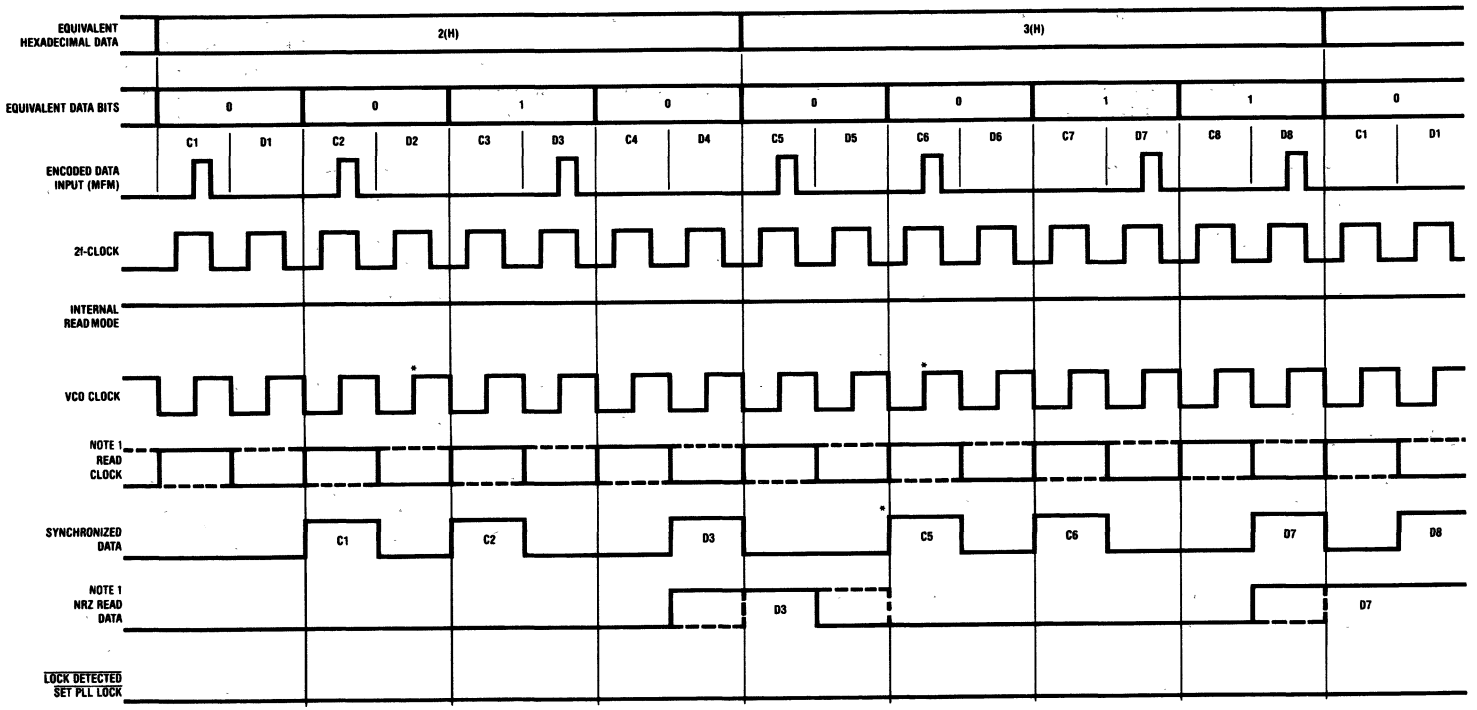
① MISSING CLOCK DETECTED is one READ CLOCK period ahead of the chip issuing D8 on the NRZ READ DATA output when READ CLOCK is delayed by one VCO clock period.

② MISSING CLOCK DETECTED is synchronous with the chip issuing D8 on the NRZ READ DATA Output when READ CLOCK is not delayed.

③ Not included on the DP8451/55.

④ The AI byte is shown only as an example address mark byte. *Any* missing clock bit which is framed by two existing clock bits will produce a missing clock detected pulse.

FIGURE 2. Missing Clock Detection Waveform Diagram

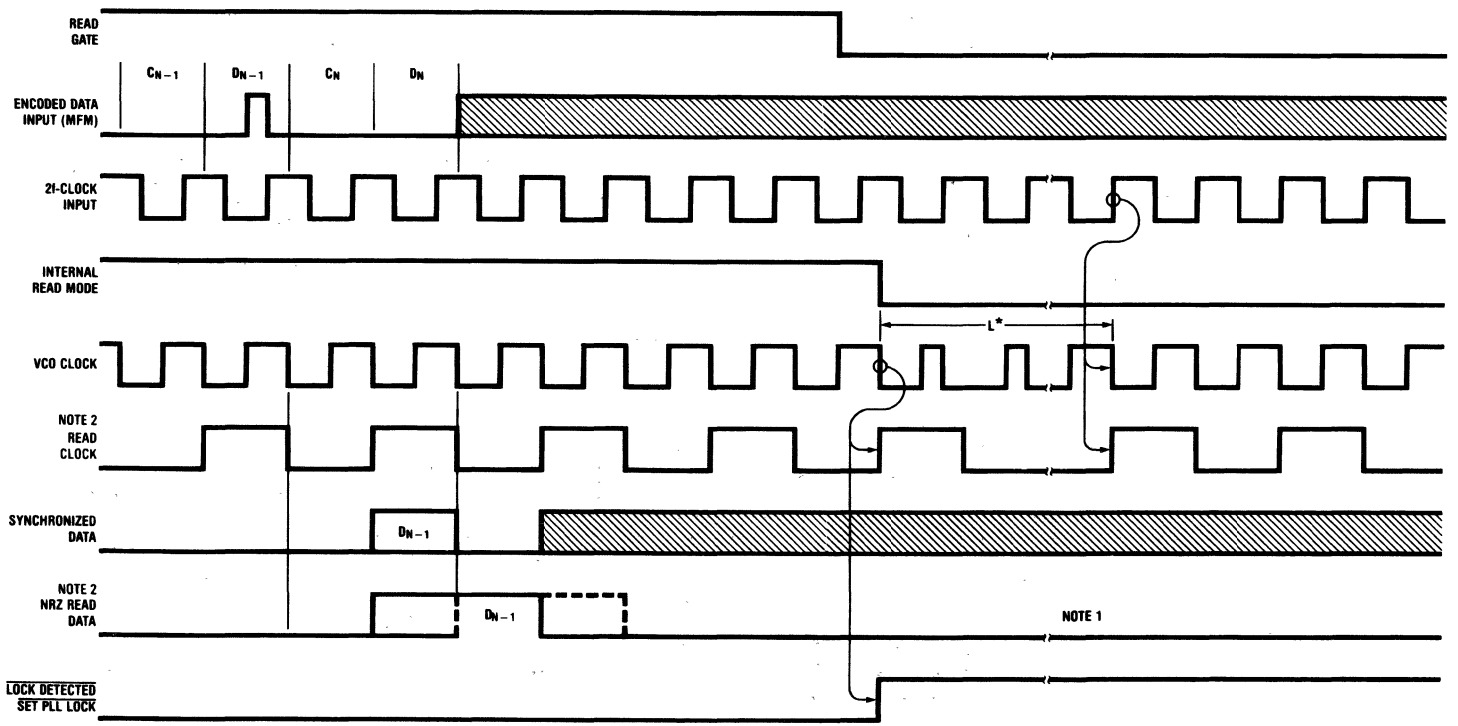


TL/F/8445–6

* **READ CLOCK and NRZ READ DATA may be delayed by one VCO clock period with respect to Synchronized Data depending on the phase of the internal clock at activation of READ GATE input.**

Note 1: Not included on the DP8451/55.

FIGURE 3. Locked-On Waveform Diagram



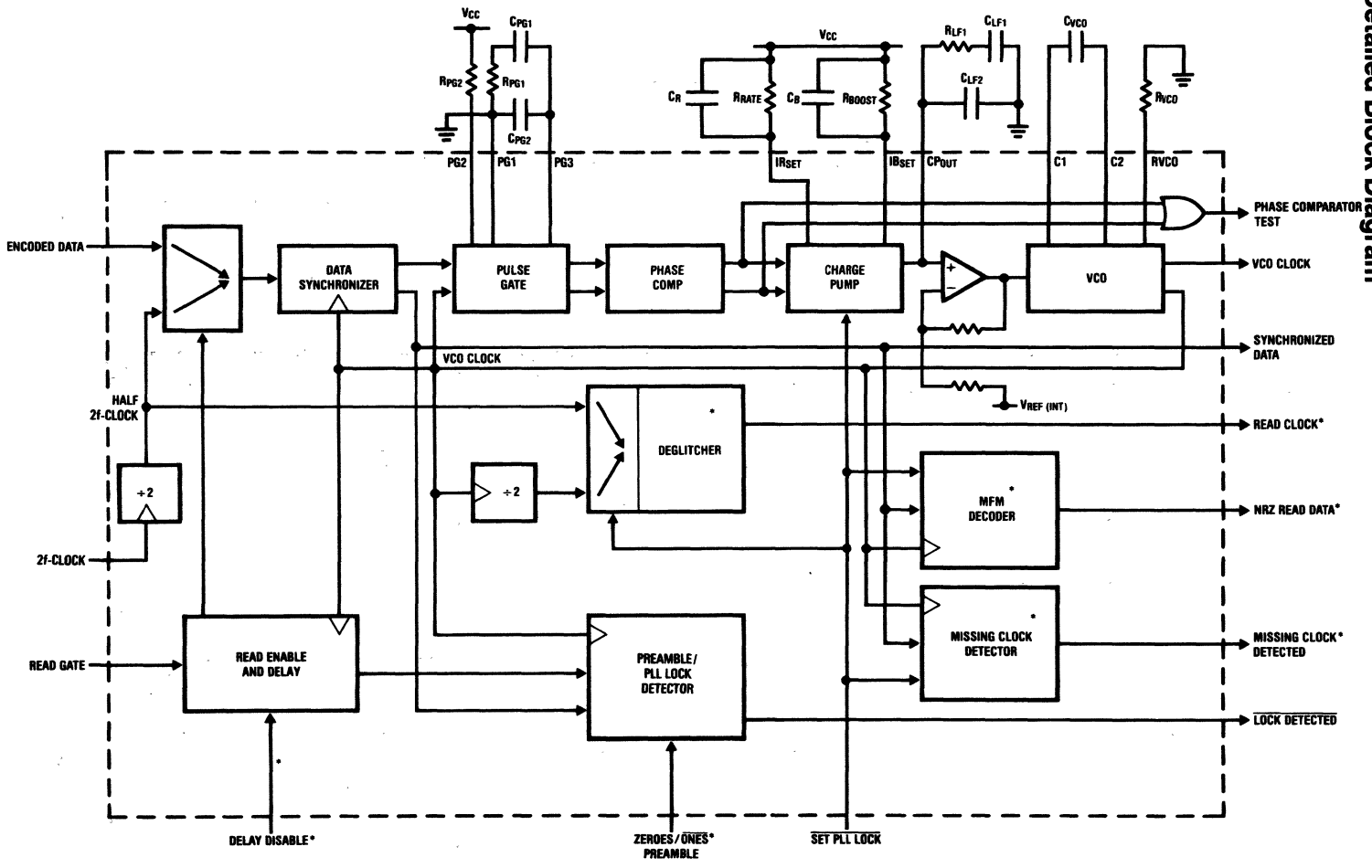
* L indicates the number of cycles required for the VCO to lock to the 2f-CLOCK.

Note 1: READ GATE going low will always result in NRZ READ DATA going low regardless of the state of the last bit.

Note 2: Not included on the DP8451/55.

FIGURE 4. Lock-Ending Sequence Waveform Diagram

Detailed Block Diagram



TL/F/8445–8

*Not included on the DP8451/55

Circuit Operation

When the READ GATE input goes high, the DP8461/65 will enter the read mode after a period determined by the state of the DELAY DISABLE pin. This may be either one or thirty two VCO CLOCK cycles. Once in the read mode the DP8465 switches from using a phase-frequency comparator to a phase-only comparator, i.e. the pulse gate is activitated. At this time, however, the DP8461 continues to use a phase-frequency comparator. Referring to *Figure 1*, as the read mode is entered, the phase-locked-loop reference signal is switched from 2F-CLOCK INPUT to the ENCODED DATA. The PLL, initially in the high-tracking rate mode, then attempts to lock onto the incoming encoded data stream. As soon as two bytes of the selected preamble are detected, (the selection is determined by the ZEROES/$\overline{\text{ONES}}$ PREAMBLE pin) the $\overline{\text{LOCK DETECTED}}$ OUTPUT goes low. At this time the DP8461 switches from using a phase-frequency comparator to using the pulse gate, thus beginning phase only comparisons. In a typical MFM disk drive application, the $\overline{\text{LOCK DETECTED}}$ OUTPUT is directly connected to the $\overline{\text{SET PLL LOCK}}$ INPUT. With this connection, track rate selection, clock output switchover, and data output enabling will occur after two consecutive preamble bytes have been detected by the chip. Typically it takes less than one byte time for the VCO to lock to the data sufficiently for preamble detection to begin following the start of the Read operation.

A low level on the $\overline{\text{SET PLL LOCK}}$ causes the PLL Charge Pump to switch from the high to low tracking rate. At the same time, the source of the READ CLOCK signal is switched from half the frequency of the 2F-CLOCK to half the VCO CLOCK. The MFM decoder also becomes enabled and begins to output decoded NRZ data. If a zeroes data preamble is present, the NRZ READ DATA OUTPUT will remain low until the end of the preamble. It will then output whatever NRZ data is present after the preamble field has ended, as shown in *Figures 2* and *3*.

When the READ GATE goes low, signifying the end of a read operation, the DP8461/65 will return to phase-frequency comparator operation. *Figure 4* shows the sequence when READ GATE goes low. The PLL reference signal is switched back to half the 2F-CLOCK and the $\overline{\text{LOCK DETECTED}}$ OUTPUT (and therefore the $\overline{\text{SET PLL LOCK}}$ INPUT) goes high. The PLL then returns to the high track rate, and the output signals return to their initial conditions. The 2F-CLOCK MUST BE APPLIED AT ALL TIMES to the DP8461/65 and DP8451/55 for proper operation.

Since the DP8461/51 employs a phase-frequency comparator until two bytes of the preamble (actually any 16 pulses within a 1010 . . . pattern) have been detected, care must be taken to ensure that when using this circuit the READ GATE is applied only within a field containing the 1010 . . . pattern. In soft sectored drives the head may be positioned anywhere on the track when initiating a read operation. Therefore, either a controller which only issues READ GATE when a high frequency synchronization field is present, or a simple external circuit between the controller and DP8461/51 to qualify the READ GATE, must be used.

CIRCUIT DESCRIPTION

1. Read Enable and Delay (DP8461/65 only): If the DELAY DISABLE input is connected low, then thirty two VCO CLOCK cycles after READ GATE goes active, the DP8461/65 will go into the read mode. If the DELAY DISABLE input is connected high, the chip will go into the read mode one VCO CLOCK cycle after READ GATE goes active. (The 32 cycle delay is permanently disabled in the DP8451/55).

2. Pulse Gate, including Multiplexer and Data Synchronizer: The Input Multiplexer selects the input to the phase-lock-loop. While the chip is in the bypassed (non-read) mode, the VCO frequency is phase and frequency locked to the 2F-CLOCK INPUT frequency. In the read mode the Input Multiplexer switches to the ENCODED DATA signal and the VCO CLOCK then begins to synchronize with the ENCODED DATA signal. Also, as soon as the read mode is entered, the DP8455/65 cease phase and frequency comparisons by employing the Pulse Gate.

In the DP8461/51 option, switchover from the phase-frequency comparator to the pulse gate (phase-only comparator) occurs after two bytes of the 1010 . . . pattern have been detected by the preamble pattern detector.

The Pulse Gate allows a reference pulse from the VCO into the Phase Comparator only after an ENCODED DATA bit has arrived. It utilizes a scheme which delays the incoming data by one-half the period of the 2F-CLOCK. This optimizes the position of the decode window and allows input jitter of approximately half the 2F-CLOCK period. The decode window error can be determined from the specification in the Electrical Characteristics Table.

3. Phase Comparator: The Phase Comparator receives its inputs from the Pulse Gate, and is edge-triggered from these inputs to provide charge-up and charge-down outputs.

4. Charge Pump: The high speed charge pump consists of a switchable constant current source and sink. The charge pump constant current is set by connecting external resistors to V_{CC} from the charge current rate set (IRSET) and current boost set (IBSET) pins. Before lock is indicated, the PLL is in the high tracking rate and the parallel combination of the resistors determines the current. In the low tracking rate after lock-on, only the IRSET resistor determines the charge pump current. The output of the charge pump subsequently feeds into external filter components and the Buffer Amplifier.

5. Buffer Amplifier: The input of the Buffer Amplifier is connected to the charge pump's constant current source/sink output as well as the external Loop Filter components. The Buffer Amplifier is configured as a high input impedance amplifier which allows for the connection of external PLL filter components to the Charge Pump output pin CPOUT. The output of the Buffer Amplifier is internally connected to the VCO control input.

6. VCO: The Voltage-Controlled-Oscillator requires a resistor from the RVCO pin to ground and a capacitor between pins C1 and C2, to set the center frequency. The VCO frequency can be varied from nominal by approximately ±20%, as determined by its control input voltage.

7. PLL Lock-on/Preamble Pattern Detector: To recognize preamble, the preamble pattern from the disk must consist exclusively of either MFM data bit zeroes (encoded into ..10.. MFM clock pulses) when the ZEROES/$\overline{\text{ONES}}$ PREAMBLE pin is set high, or MFM data bit ones (encoded into ..01.. MFM pulses) when set low (DP8461/65 only). The preamble pattern must be long enough to allow the PLL to lock, and subsequently for the Preamble Pattern Detector circuit to detect two complete bytes.

Once the chip is in the read mode, the VCO proceeds to lock on to the incoming data stream. The Preamble Pattern

Circuit Operation (Continued)

Detector then searches for a continuous pattern of 16 consecutive pulses at one-half the VCO frequency to indicate lock has been achieved.

The $\overline{\text{LOCK DETECTED}}$ output then goes low. At this time, in the DP8461/51 option, the PLL switches from using a phase-frequency comparator to employing a pulse gate and thus doing only phase comparisons. Any deviation from the above-mentioned one-zero pattern at any time before PLL Lock is detected will reset the PLL Lock Detector. The lock detection procedure will then start again.

8. MFM Decoder (DP8461/65 only): The MFM Decoder receives synchronized MFM data from the Pulse Gate and converts it to NRZ READ DATA. For run-length-limited codes the MFM Decoder and Missing Clock Detector will not be used.

9. Missing Clock Detector (DP8461/65 only): This block is only required for soft-sectored drives, and is used to detect a missing clock violation of the MFM pattern. The missing clock is inserted when writing to soft-sectored disks to indicate the location of the Address Mark in both the ID and the Data fields of each sector. Once PLL Lock has been indicated, the Missing Clock Detector circuit is enabled. MISSING CLOCK DETECTED will go active if at any time the incoming data pattern contains one suppressed clock bit framed by two adjacent clock bits. (This condition is not constrained to any particular byte pattern such as "A1.") The output signal goes high for one cycle of READ CLOCK.

10. Clock Multiplexer and Deglitcher (DP8461/65 only): When the $\overline{\text{SET PLL LOCK}}$ input changes state this circuit switches the source of the READ CLOCK signal between the half 2f-CLOCK frequency and the half VCO CLOCK frequency. A deglitcher circuit is utilized to ensure that no short clock periods occur during either switchover.

BIT JITTER TOLERANCE

The spec, t-window, as defined in the AC Electrical Characteristics table, describes the distance from the optimum window boundary a single shifted data bit may be placed (following complete PLL lock and stabilization) before it risks being interpreted as residing in the adjacent synchronization window. This is known as the **static window measurement**, which combines all contributing factors of window jitter and displacement within the data separator into a single specification.

The two options of the DP8451/55/61/65, the −4 and −3 offer decreasing static window errors (respectively) so that the parts may be selected for different data rates (up to 20 Mbit/sec). The −4 part will be used in most low data rate applications. As an example, at the 5 Mbit/sec MFM data rate of most 5¼ inch drives, the chip contributes up to ±10 ns of window error, out of the total available window of 100 ns. This allows the disk drive to have a margin of 40 ns of jitter from nominal bit position before an error will occur.

ANALOG CONNECTIONS

External passive components are required for the Pulse Gate, Charge Pump, Loop Filter and VCO as shown in *Figure 5*. The information provided here is for guidelines only. The user should select values according to his own system requirements. Phase-Locked Loops are complex circuits that require detailed knowledge of the specific system. Factors such as loop gain, stability, response to change of signal, lock-on time, etc are all determined by the external components. In many disk systems these factors are critical, and National Semiconductor recommends the designer be knowledgeable of phase-locked-loops, or seek the advice of an expert. Inaccurate design will probably result in excessive disk error rates. The phase-locked-loop in the DP8461/65 has many advantages over all but the most sophisticated discrete designs, and if the component values are selected correctly, it will offer significant performance advantages. This should result in a reduction of disk error rates over equivalent discrete designs. Please refer to the National Semiconductor Application Note AN-414, Precautions for Disk Data Separator Designs, AN-415, Designing with the DP8461, AN-416, Designing with the DP8465, and to the Disk Interface Design Guide and User's Manual, Chapter 1.

Circuit Operation (Continued)

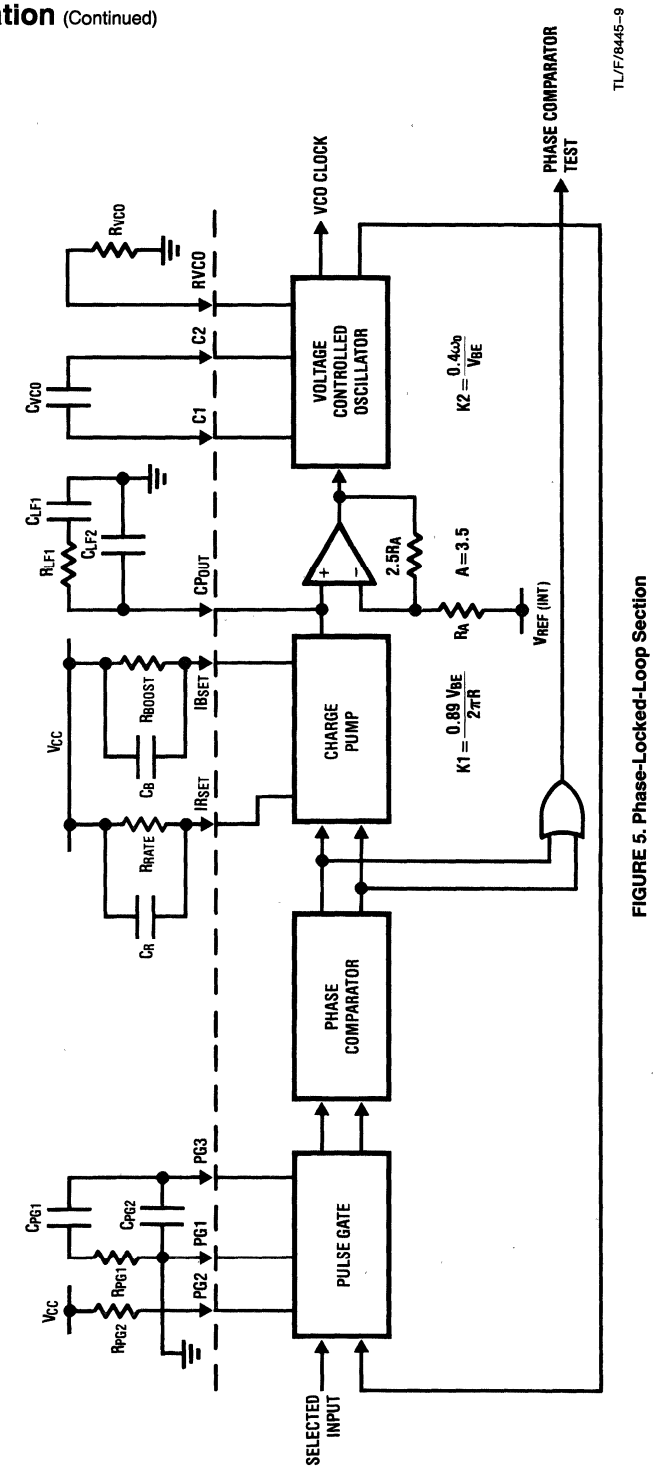


FIGURE 5. Phase-Locked-Loop Section

2

Circuit Operation (Continued)

Pulse Gate

There are four external components connected to the Pulse Gate as shown in *Figure 6* with the associated internal components. The values of R_{PG1}, R_{PG2}, C_{PG1}, and C_{PG2} are dependent on the data rate. C_{PG1} and C_{PG2} are proportional to the data rate, while R_{PG1} and R_{PG2} are inversely proportional. Table I shows component values for the data rates given. Component values are calculated by selecting R_{PG2} from Table I. Next calculate

$$C_{PG1} = \left(\frac{2.12 \times 10^5}{890 + R_{PG2}}\right)\left(\frac{1}{100 \times R_S}\right)^2$$

$$C_{PG2} = \frac{1}{10} C_{PG1}, \text{ and } R_{PG1} = \left(\frac{890 + R_{PG2}}{2.38 \times 10^5}\right)(100 \times R_S).$$

In the above equations R_S is the rotational speed and, for 3600 RPM, R_S = 60 Hz. A rotational speed of 3600 RPM was assumed for the calculations in Table I. For data rates not listed, R_{PG2} may be approximated as (30 kΩ/f_{DATA}) − 1.20 kΩ = R_{PG2} where f_{DATA} is the data rate in Megabits/second.

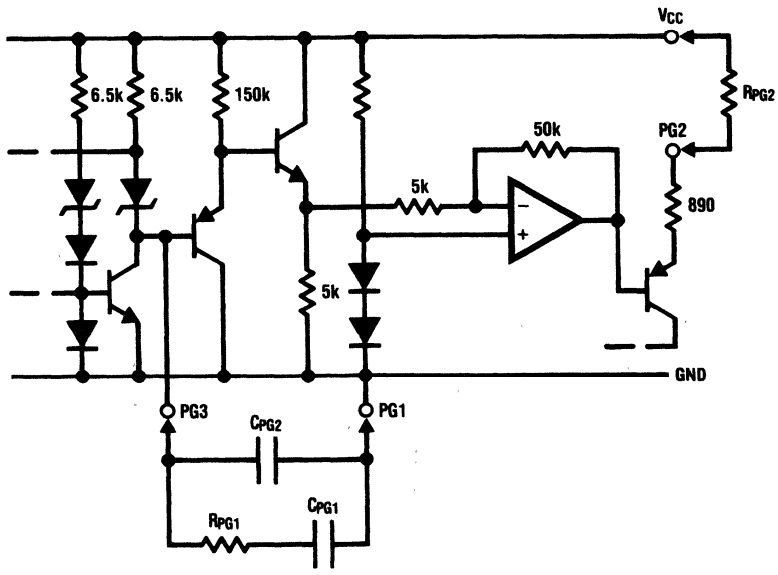
TABLE I. Pulse Gate Component Selection Chart
Components with 10% tolerance will suffice

Data Rate	R_{PG2}	R_{PG1}	C_{PG1}	C_{PG2}
2 Mbit/sec	15 kΩ	430Ω	.39 μF	.039 μF
5 Mbit/sec	4.7 kΩ	150Ω	1 μF	0.1 μF
10 Mbit/sec	1.8 kΩ	68Ω	2.2 μF	.22 μF
15 Mbit/sec	750Ω	39Ω	3.9 μF	.39 μF

Charge Pump

Resistors R_{RATE} and R_{BOOST} determine the charge pump current. The Charge Pump bidirectional output current is approximately 1.9 × the input current (See DC Electrical Characteristics for exact relationship). In the high tracking rate with $\overline{\text{SET PLL LOCK}}$ high, the input current is I_{BSET} + I_{RSET}, i.e., the sum of the currents through R_{BOOST} and R_{RATE} from V_{CC}. In the low tracking rate, with $\overline{\text{SET PLL LOCK}}$ low, this input current is I_{RSET} only.

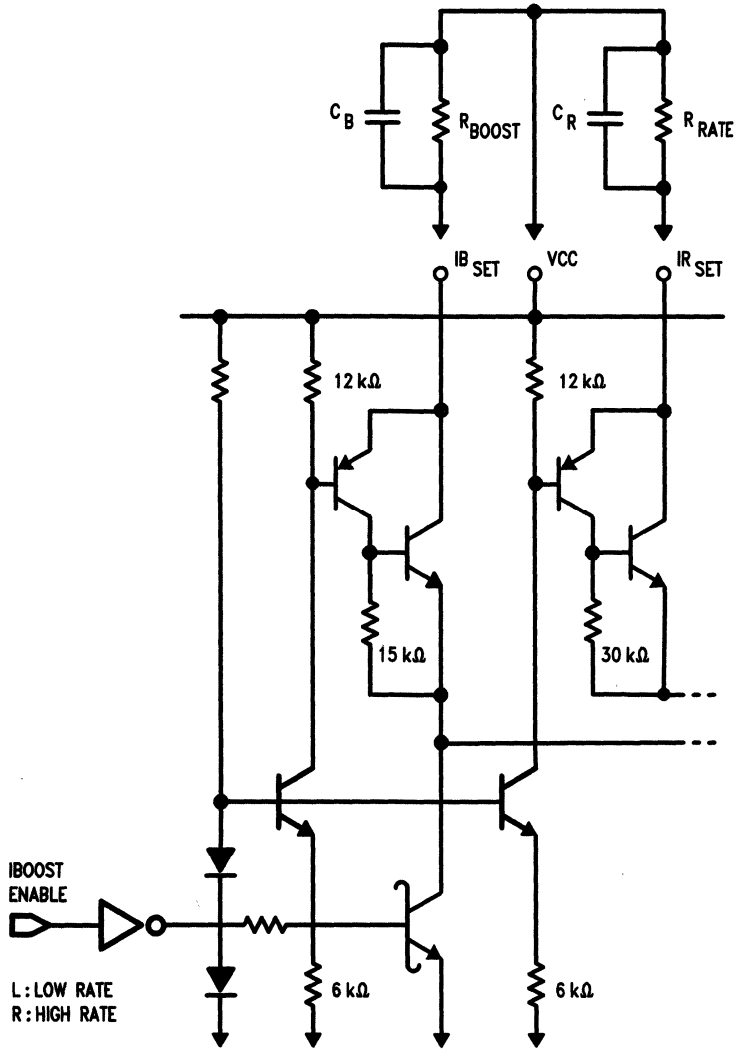
A recommended approach for selecting values for R_{RATE} and R_{BOOST} is described in the design example in the Loop Filter Section. A typical loop gain change of 2:1 for high to low tracking rate would require R_{BOOST} = R_{RATE}. Selecting R_{RATE} to be 820Ω would then result in R_{BOOST} equaling 820Ω. Referring to *Figure 7*, the input current is effectively V_{BE}/R_{RATE} in the low tracking rate, where V_{BE} is an internal voltage. This means that the current into or out of the loop filter is approximately (1.95 × V_{BE}/820) − 70 μA = 1.72 mA. Note that although it would seem the overall gain is dependant on V_{BE}, this is not the case. The VCO gain is altered internally by an amount inversely proportional to V_{BE}, as detailed in the section on the Loop Filter. This means that as V_{BE} varies with temperature or device spread, the gain will remain constant for a particular fixed set of values of R_{RATE} and R_{BOOST}. This alleviates the need for potentiometers to select values for each device. The tolerance required for these two resistors will depend on the total loop gain tolerance allowed, but 5% would be typical. Also V_{CC} bypass capacitors are required for these two resistors. A value of .01 μF is suitable for each.



TL/F/8445–10

FIGURE 6. Pulse Gate Controls

Circuit Operation (Continued)



TL/F/8445–11

FIGURE 7. I_{RATE} Set and I_{BOOST} Set

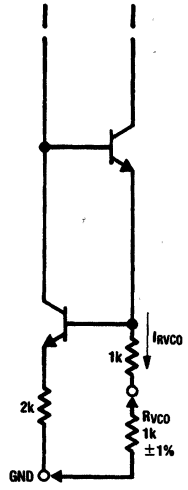
Circuit Operation (Continued)

VCO

The value of R_{VCO} is fixed at 1 kΩ ±1% in the External Component Limits table. *Figure 8* shows how R_{VCO} is connected to the internal components of the chip. This value was fixed at 1 kΩ to set the VCO operating current such that optimum performance of the VCO is obtained for production device spreads. This means fixed value components will be adequate to set the VCO center frequency for production runs. The value of C_{VCO} can therefore be determined from the VCO frequency f_{VCO}, using the equation: $C_{VCO} = [1/(R_{VCO})(f_{VCO})] - 5$ pF where f_{VCO} is twice the input data rate. As an example, for a 5 Mbit/sec data rate, $f_{VCO} = 10$ MHz, requiring that $C_{VCO} = 95$ pF. This does not take into account any inter-lead capacitance on the printed circuit board; the user **must** account for this. The amount of toler ance a particular design can afford on the center frequency will determine the capacitor tolerance. The capacitor is connected to internal circuitry of the chip as shown in *Figure 9*.

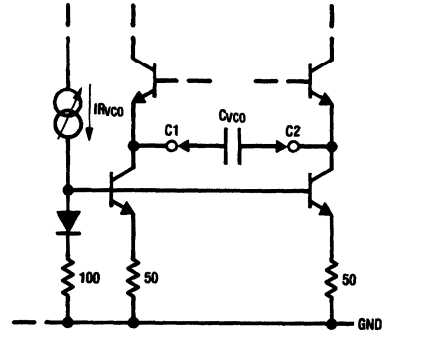
As the data rate increases and C_{VCO} gets smaller, the effects of unwanted internal parasitic capacitances influence the frequency. As a guide the graph of *Figure 10* shows approximately the value of C_{VCO} for a given data rate.

The VCO control input operational range (pin 4) lies at approximately 1.4 volts with a control swing of ±100 millivolts. The VCO itself is constrained to swing a maximum of approximately ±20% of its center frequency, and will remain clamped if the voltage at pin 4 exceeds its operational limit. The VCO center frequency may then be determined by: 1) holding pin 4 at ground potential and measuring the VCO frequency (−20% value); 2) holding pin 4 at approximately 3 volts and measuring the VCO frequency (+20% value); 3) averaging the two measured frequencies for the equivalent center frequency.



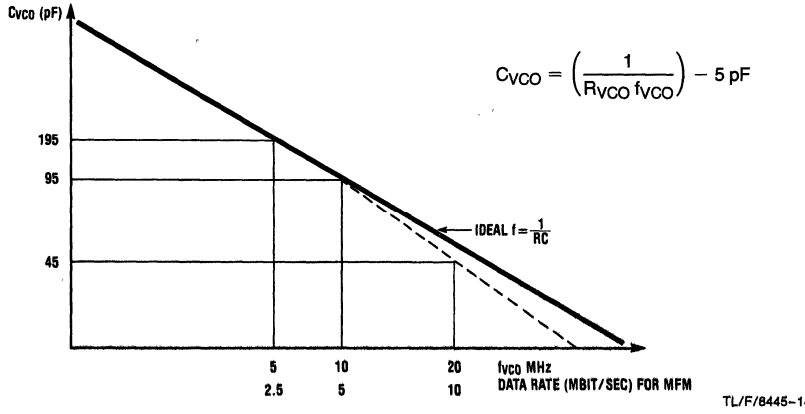
TL/F/8445–12

FIGURE 8. VCO Current Setting Resistor



TL/F/8445–13

FIGURE 9. VCO Capacitor



TL/F/8445–14

FIGURE 10. VCO Capacitor Value for Disk Data Rates

Circuit Operation (Continued)

Loop Filter

The input current into the Buffer Amplifier is offset by a matched current out of the Charge Pump, and even so is much less than the switching current in or out of the Charge Pump. It can therefore be assumed that all the Charge Pump switching current goes into the Loop Filter components R_1 and C_1 and C_2. The tolerance of these components should be the same as R_{RATE} and R_{BOOST}, and will determine the overall loop gain variation. The three components connected to the Charge Pump output are shown in *Figure 11*. Note the return current goes to analog GND, which should be electrically very close to the GND pin itself.

The value of capacitor C_1 determines loop bandwidth ... the larger the value the longer the loop takes to respond to an input change. If C_1 is too small, the loop will track any jitter on the ENCODED DATA input and the VCO output will follow this jitter, which is undesirable. The value of C_1 should therefore be large enough so that the PLL is fairly immune to phase jitter but not large enough that the loop won't respond to longer term data rate changes that occur on the disk drive.

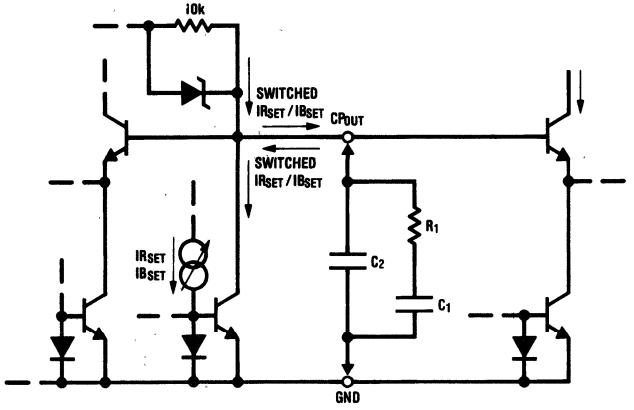
The damping resistor R_1 is required to regulate the second-order behavior of the closed-loop system (overshoot). A value of R_1 that would give a phase margin of around 45 degrees would be a reasonable starting point.

The main function of the capacitor C_2 is to smooth the action of the charge pump at the VCO input. Typically its value will be less than one tenth of C_1. Further effects of C_2 will be discussed later.

Figure 12 shows the relevant phase-locked-loop blocks that determine system response, namely the Phase Detector, Filter/Buffer Amplifier, and VCO. The Phase Detector (Phase Comparator and Charge Pump) produces an aggregate output current i which is proportional to the phase difference between the input signal and the VCO signal. The constant (K_1) is

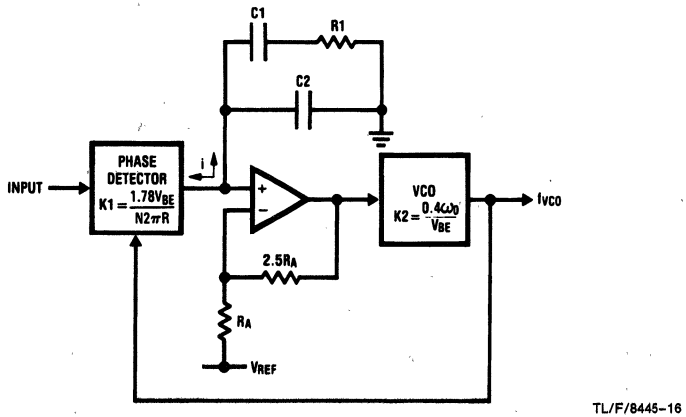
$$\frac{1.78\ V_{BE}}{N2\pi R} \text{ amps per radian, where } N = \frac{f_{VCO}}{f_{DATA}}.$$

R is either R_{RATE} or $R_{RATE} \parallel R_{BOOST}$. The amplified aggregate current feeds into or out of the filter impedance (Z), producing a voltage to the VCO that regulates the VCO frequency. The VCO gain constant is 0.4 ω_{VCO}/V_{BE} radians per second per volt. Under steady state conditions, i will be zero and there will be no phase difference between the input signal and the VCO. Any change of input signal will pro-



TL/F/8445–15

FIGURE 11. Charge Pump Out



TL/F/8445–16

FIGURE 12. Loop Response Components

Circuit Operation (Continued)

duce a change in VCO frequency that is determined by the loop gain equation. This equation is determined from the gain constants K_1, A and K_2 and the filter v/i response.

The impedance Z of the filter is:

$$\frac{1}{sC_2} \left\| \left(\frac{1}{sC_1} + R_1 \right) \right. = \frac{1 + sC_1R_1}{sC_1 \left(1 + \frac{C_2}{C_1} + sC_2R_1 \right)}$$

If $C_2 \ll C_1$ then the impedance Z approximates to:

$$\frac{1 + sC_1R_1}{sC_1 (1 + sC_2R_1)}$$

The overall loop gain is then

$$G(s) = \frac{K_1AK_2}{s} \times \frac{1 + sC_1R_1}{sC_1(1 + sC_2R_1)}$$

Let $G_{(K)} = K_1 A K_2$

$$F(s) = \frac{1 + SC_1R_1}{SC_1 (1 + SC_2R_1)}$$

The Overall Closed Loop Gain is:

$$\frac{\phi_{OUT}}{\phi_{IN}} = \frac{G_{(K)} F(s)}{s + G_{(K)} F(s)}$$

Substituting, We Get

$$\frac{\phi_{OUT}}{\phi_{IN}} = \frac{G_{(K)} (SC_1R_1 + 1)}{S^3 R_1 C_1 C_2 + S^2 C_1 + GK (SC_1R_1 + 1)}$$

$$= \frac{(G_{(K)}/C_1) (SR_1C_1 + 1)}{S^3R_1C_2 + S^2 + SG_{(K)}R_1 + G_{(K)}/C_1}$$

If $C_2 \ll C_1$, we can ignore the 3rd Order Component introduced by C_2 then:

$$\frac{\phi_{OUT}}{\phi_{IN}} = \frac{(G_{(K)}/C_1) (SR_1C_1 + 1)}{S^2 + SG_{(K)}R_1 + G_{(K)}/C_1}$$

This is a second Order Loop and can be solved as follows:

$$S^2 + SG_{(K)}R_1 + G_{(K)}/C_1 = S^2 + 2\zeta \omega_n S + \omega_n^2$$

$$\therefore C_1 = \frac{G_{(K)}}{\omega_n^2}$$

$$R_1 = \frac{2\zeta\omega_n}{G_{(K)}}$$

$\zeta = 1.0$ For Critically Damped Response

From the above equations:

$$\omega = \sqrt{\frac{G_{(K)}}{C_1}}$$

$$G_{(K)} = K_1 A K_2 = \frac{0.89 \times V_{BE}}{2\pi R} \times \frac{0.4 \times \omega_{VCO}}{V_{BE}} \times 3.5$$

MFM encoded data has a two to one frequency range within the data field. The expression $K = (0.89 \times V_{BE} / 2\pi R)$ is valid when the MFM data pattern is at its maximum frequency. In order to make this equation more general, it may be written as follows: $K = (1.78 \times V_{BE} / 2\pi RN)$ where N is defined as the V_{CO} frequency divided by the encoded data frequency, or, N is equal to F_{VCO}/F_{DATA} (N = 2 for maximum data rate i.e., MFM = 101010 ... and N = 4 for minimum data rate) i.e., MFM = 100010001 .... Now $G_{(K)}$ can be written as follows:

$$G_{(K)} = \frac{1.78 \times V_{BE}}{2\pi RN} \times \frac{0.4 \times \omega_{VCO}}{V_{BE}} \times 3.5$$

$$= \frac{2.5 \times F_{VCO}}{RN}$$

$$\omega_n = \sqrt{\frac{2.5 \times F_{VCO}}{C_1RN}}$$

$R = R_{RATE}$ in the low track rate

$R = R_{RATE} // R_{BOOST}$ in the high track rate

From the above equations:

$$\omega_n = \frac{R_1 G_{(K)}}{2\zeta}$$

$$G_{(K)} = C_1 \omega_n^2$$

$$\zeta = \text{(damping factor)} = \frac{R_1 \omega_n C_1}{2}$$

The damping factor should approach, but not fall below, 0.5 when ω_n is minimum. Response to bit shift is minimized when the damping factor is small; however, if the damping factor drops much below 0.5, the system tends to be oscillatory (underdamped).

Additionally, loop performance is poor (excessive phase acquisition times) if the damping factor becomes significantly greater than 1.0. Any increase in loop bandwidth (due to R decreasing in the high track rate) produces a proportional increase in the damping factor, and this should be limited to the point where the maximum damping factor does not significantly exceed 1.0. With the damping factor range established, loop design can now proceed. The following design example is for a 5 Mbit/sec MFM system.

A 1550 Krads/sec bandwidth in the non read mode results in a wide capture range; a 4% frequency difference between the crystal and recorded data would not cause an acquisition problem. (This bandwidth may seem excessive to some and if the user does not think it is necessary, he may design his filter with a more desirable bandwidth. For an in-depth discussion of this point, it is suggested that the reader refer to the Disk Interface Design Guide and User's Manual, chapter 1, sections 1.3 through 1.7.

This design example assumes that the $\overline{\text{SET PLL LOCK}}$ pin is tied to the $\overline{\text{PLL LOCK DETECTED}}$ pin. This results in the track rate being switched from high to low after two bytes of preamble are detected. As an alternative, the $\overline{\text{SET PLL LOCK}}$ pin may be tied to an inverted READ GATE signal, resulting in the track rate switching immediately to low when READ GATE is asserted. This is discussed further in the above mentioned reference material.

TABLE II.

Data Rate (NRZ)	Non-Read		Read		Charge Pump		Loop Filter		
	$\omega_{n(MAX)}$ rads/sec	ζ	$\omega_{n(MIN)}$ Rads/sec	ζ	R_{RATE} Ω	R_{BOOST} Ω	R_1 Ω	C_1 μF	C_2 pF
5 Mbit/sec	1550K	1.12	797K	0.55	820	820	120	0.012	300
5 Mbit/sec	903K	0.99	435K	0.48	1500	1300	100	0.022	390
5 Mbit/sec	659K	1.55	248K	0.52	1500	590	69	0.068	1500

Circuit Operation (Continued)

In the non read mode or high track rate.

$$\omega_n = \sqrt{\frac{2.5 \times F_{VCO}}{C_1 RN}}$$

Choose $R = R_{RATE} // R_{BOOST} = 410$

In the non-read mode $N = 2$

$$1550 \text{ Krads/sec} = \sqrt{\frac{2.5 \times 10^7}{C_1 \times 410 \times 2}}$$

$C_1 = 0.012\ \mu F$

In the preamble, after two bytes are detected and $\overline{\text{PLL LOCK DETECT}}$ goes low

$$\omega_n = \sqrt{\frac{2.5 \times F_{VCO}}{C_1 RN}}$$

$R = R_{RATE} = 820$

$N = 2$

$\omega_n = 1127$ Krads/sec

Again, in the data field, the minimum data frequency is equal to one half the preamble frequency. This means that $N = 4$ in the bandwidth equation. This reduces the bandwidth to:

$$\omega_{n(min)} = \frac{1}{\sqrt{2}} \times 1107 \text{ Krads/sec} = 797 \text{ Krads/sec}$$

Before, we stated that the minimum value of ζ should be 0.5; knowing $\omega_{n(min)}$ we can now solve for R_1

$$\zeta = \frac{\omega_n R_1 C_1}{2}$$

Choose $\zeta_{(min)} = 0.55$

$$R_1 = \frac{2\zeta}{\omega_n C_1}$$

$R_1 = 115\Omega$ (choose 120Ω)

The maximum damping value occurs in the high track rate;

$\zeta_{(max)} = \omega_{n(max)} R_2 C_1/2$

$= 1550 \text{ Krads/sec} \times 120 \times 0.012\ \mu F/2$

$\zeta_{(max)} = 1.12$

The maximum damping value in the read mode is as follows:

$\zeta_{(max-read)} = 1127 \text{ Krads/sec} \times 120 \times 0.012\ \mu F/2$

$\zeta_{(max-read)} = 0.81$

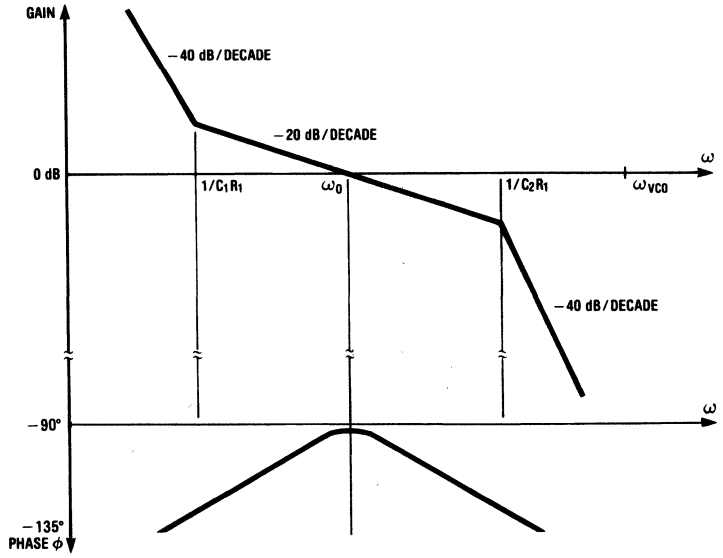
The continuous behavior (non-quantized) approximation used to predict loop performance assumes that the phase detector output is constantly proportional to the input phase difference. In reality, the phase detector output is a pulse applied for a period of time equal to the phase difference. The function of C_2 is to smooth the phase detector output (VCO control voltage) over each cycle. C_2 also adds a second pole to the filter transfer function. This pole should be far enough outside the loop bandwidth (at least one order of magnitude) that its phase and amplitude contribution is negligible in the loop bandwidth. If:

$C_2 = C_1/50 = 240$ pF (choose 300 pF)

The final loop component is R_{BOOST}. Since R_{RATE} and the parallel combination of R_{RATE} and R_{BOOST} are known, we can calculate R_{BOOST}.

$R_{BOOST} = (R_P)(R_{RATE}) / (R_{RATE} - R_P) = 820\Omega$

The above filter values and those for other bandwidths are listed on preceding page.



TL/F/8445–17

FIGURE 13. Bode Plot of Loop Response

Circuit Operation (Continued)

The calculated values are only a guide, the user should then empirically test the loop and determine stability, lock-on time, jitter tolerance, etc.

The desired Bode plot of gain and phase is shown in *Figure 13*, with 20 dB/decade slope at ω_0 for stability at unity gain.

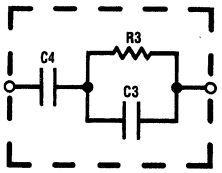
Capacitor C_2 governs the PLL's ability to reject instantaneous bit jitter. As C_2 increases in value, the effective jitter rejection will also increase. However, as the frequency of the pole R_1 and C_2 produce (while increasing C_2) decreases, loop stability will decrease, and the second-order approximation used to analyze the circuit becomes inaccurate. Thus, it is recommended that C_2 remain one tenth (or less) the value of C_1.

The value of resistor R_1 inversely effects the break frequencies on the Bode plot, and directly effects the loop's damping ratio (overshoot response). The capacitor C_1 governs the bandwith of the loop. Too high a value will slow down the response time, but make the PLL less prone to jitter or frequency shift whereas too low a value will improve response time while tending to increase the PLL's reaction to jitter.

Other filter combinations may be used, other than R_1 in series with C_1, all in parallel with C_2. For example the filter shown in *Figure 14* will also perform similarly, and in fact for some systems it will yield superior performance.

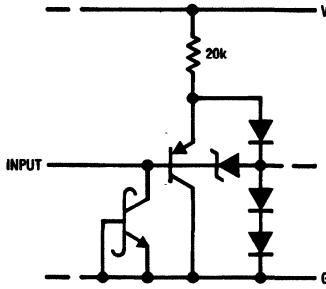
DIGITAL CONNECTIONS TO THE DP8461/65

Figure 17 shows a connection diagram for the DP8461/65 in a typical application. All logic inputs and outputs are TTL compatible as shown in *Figure 15* and *16*. The VCO CLOCK output is 74AS compatible. All other outputs are 74ALS compatible. All inputs are 74ALS compatible and therefore can be driven easily from any 74 series devices. The raw MFM from the pulse detector in the drive is connected to the ENCODED DATA input. The DELAY DISABLE input determines whether attempting lock-on will begin immediately after READ GATE is set or after 2 bytes. Typically in a hard-sectored drive, READ GATE is set active as the sector pulse appears, meaning a new sector is about to pass under the head. Normally the preamble pattern does not begin immediately, because gap bytes from the preceding sector usually extend just beyond the sector pulse. Allowing 2 bytes to pass after the sector pulse helps ensure that the PLL will begin locking on to preamble, and will not be chasing non-symmetrical gap bits. Thus DELAY DISABLE should be set low for this kind of disk drive.



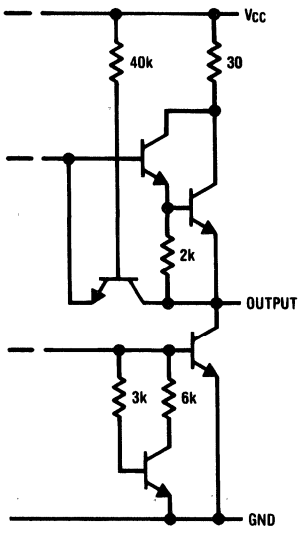
TL/F/8445–18

FIGURE 14. Alternate Loop Filter Configuration



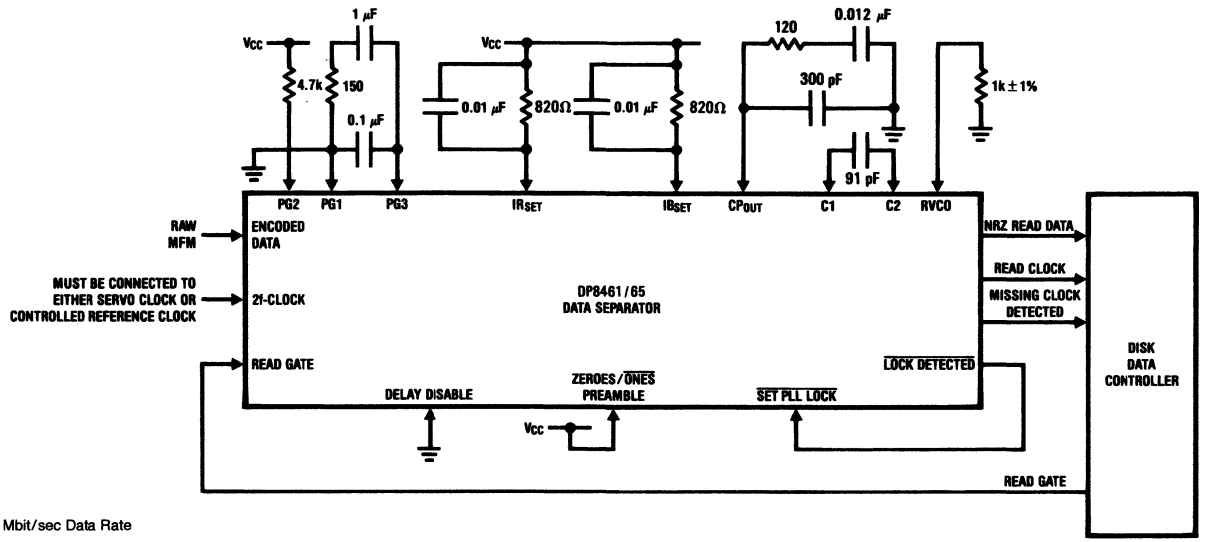
TL/F/8445–19

FIGURE 15. Logic Inputs



TL/F/8445–20

FIGURE 16. Logic Outputs

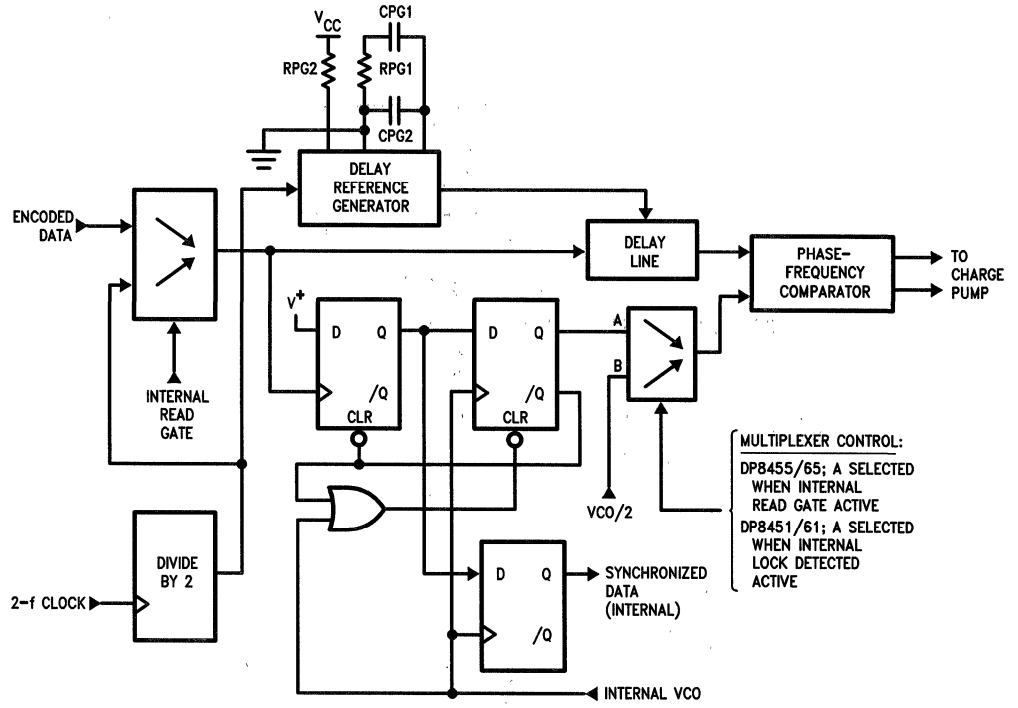


1) MFM Data Input, 5 Mbit/sec Data Rate
2) 32 Bit Delay to Enable
3) All Zeroes (NRZ) Preamble

TL/F/8445–21

FIGURE 17. Typical Connection to DP8461/65

Block Diagram



TL/F/8445–25

Circuit Operation (Continued)

For soft sectored drives, the controller normally will not wait for the index pulse before it attempts lock-on, so that READ GATE may go active at any time. Chances are the head will not be over a preamble field and therefore there is no need to wait 2 bytes before attempting lock-on. DELAY DISABLE can therefore be set high. If a non-preamble field is passing by as READ GATE goes active, the DP8461/65 will not indicate lock, and no data decoding will occur nor will MISSING CLOCK DETECTED go active. Normally, if lock-on has not been achieved after a certain time limit, the controller will de-activate READ GATE and then try again.

For MFM encoded disk drives, the $\overline{\text{LOCK DETECTED}}$ output will be connected back to the $\overline{\text{SET PLL LOCK}}$ input. As the PLL achieves lock-on, the DP8461/65 will automatically switch to the lower tracking rate and decoded data will appear at the NRZ READ DATA output. Also the READ CLOCK output will switch from half the 2F-CLOCK frequency to the disk data rate frequency. If a delay is required before the changeover occurs, a time delay may be inserted between the two pins.

Some drives have an all-ONES data preamble instead of all-ZEROES and the DP8461/65 must be set to the type being used before it can properly decode data. The ZEROES/$\overline{\text{ONES}}$ PREAMBLE input selects which preamble type the chip is to base its decoding phase on.

USE WITH RUN-LENGTH-LIMITED CODES (RLL)

If the drive uses a Run-Length-Limited Code (RLL) such as 1,7 or 1,8 instead of MFM, the user might choose to use the DP8451/55. These circuits contain the PLL portion of the DP8461/65 and thus perform the data synchronization function. RAW DATA is input to pin 16 and the 2F-CLOCK is applied to pin 17. Instead of supplying NRZ DATA, SYNCHRONIZED DATA OUTPUT is issued at pin 12. The VCO CLOCK, pin 8, is used to clock this data into external decoding circuitry. As long as the high frequency pattern of . . . 1010 . . . is used for the preamble, the user may choose the DP8451 if he desires to have the circuit perform phase and frequency comparisons until two bytes of preamble are detected by the on chip preamble pattern detector.

If a 2,7 code is being used the DP8465/55 may be used. Again, since the DP8465 MFM decoding function will not be used, the user may choose to use the DP8455. However, the National Semiconductor DP8462 is designed specifically for the 2,7 code. It is recommended that the user reviews the DP8462 specification for the added advantages the circuit offers with the 2,7 format.

Applications of the DP8461/65 Data Separator

The DP8461/65 are the first integrated circuits to place on one chip a PLL with features that offer the improved speed and reliability required by the disk industry. Not only does each chip simplify disk system design, but also provides fast lock-on to the incoming preamble. Once locked on, the loop is set into a more stable mode. This inherent loop stability allows for a sizeable amount of jitter on the data stream, such as is encountered in many disk systems. Once in the stable tracking rate, the SYNCHRONIZED DATA output represents the incoming ENCODED DATA and is synchronous with VCO CLOCK. If the disk is MFM encoded, then the chip can decode the synchronized data into NRZ READ DATA and READ CLOCK. These are available as outputs from the chip allowing the NRZ READ DATA to be deserialized using the READ CLOCK.

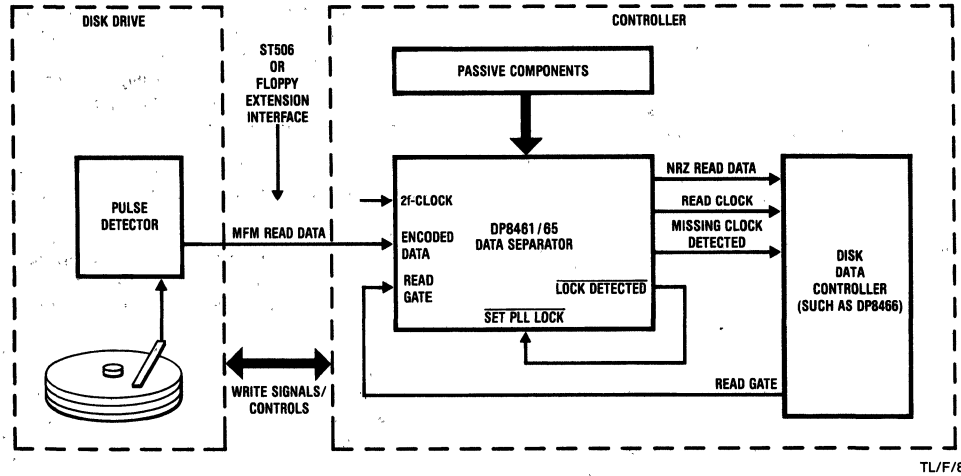
The DP8461/65 are capable of operating at up to 20 Mbits/sec data rates and so are compatible with a wide assortment of disk drives. The faster data rates of the 8-inch and 14-inch disk drives will mandate the selection of the DP8461/65–3 parts with their narrower window margins on the incoming data stream. This will also be the case when 5¼-inch drives achieve higher data rates. Some 8-inch and 14-inch disk drives incorporate the functions of the DP8461/65, but use many discrete ICs. In these cases, replacing these components with the DP8461/65 will offer reduced P.C. board area, lower cost, and improved performance while simplifying circuit testing.

Most 5¼-inch and many 8-inch and 14-inch disk drives manufactured at present do not incorporate any of the functions of the DP8461/65. This is so primarily because the PLL function is difficult to design and implement and requires circuitry which covers a large area of the printed circuit card. This is undesirable both from the drive size aspect and from the cost aspect (the cost includes soldering, testing, and adjusting the components). Consequently, most smaller disk drives output MFM encoded data so that the phase-locked-loop and data separation have to be performed by the controller. The DP8461/65 will therefore replace these functions in controller designs, as shown in *Figure 18*.

System design criteria has become more flexible because the DP8461/65 provide a one-chip solution, requiring only a few external passive components with fixed values. Each operates from a +5V supply, typically consumes about 0.3W, and is housed in a narrow 24-pin package. The circuitry has been designed so that the external resistors and capacitors need not be adjustable; the user chooses the values according to the disk drive requirements. Once selected, they will be fixed for that particular drive type. These features make it possible to transfer these functions to the disk drive, as shown in *Figure 19*. Apart from a slight increase in board area, the advantages outweigh the disadvantages. First, the components selected are fixed for each type of drive and this facilitates the problem of interchangeability of drives. At present, controllers are adjusted to function with each specific drive; with the DP8461/65 in the drive, component adjustment will no longer be required. Second there is often a problem of reliability of data transfer. The data returning from the disk drive is susceptible to noise, bit shift, etc. Soft errors will occur when the incoming disk data bit position is outside the Pulse Gate window as it is being synchronized to the VCO clock in the phase-locked-loop. Obviously, the nearer the PLL is to the data source, the less chance there is that extraneous noise or transmission line imbalances will cause errors to occur. Thus placing the DP8461/65 in the drive will increase the reliability of data transfer within the system.

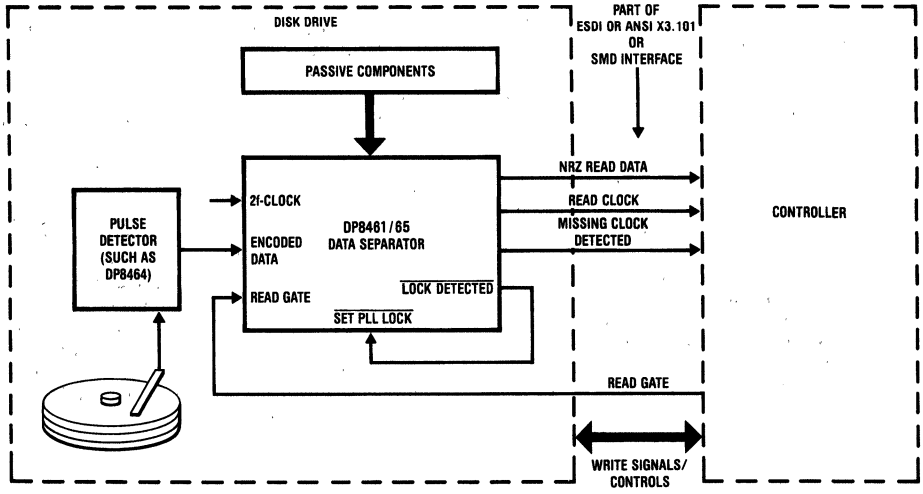
A third advantage is data rate upgrading. Most 5¼-inch drives have 5 Mbit/sec data rate because the early drives were made with this data rate. This meant the controllers had to be designed with PLLs which operate at this data rate. It is therefore difficult for drive manufacturers to introduce new drives that are not compatible with existing controllers. Since no new standard data rate has emerged, they must continue to produce drives at this data rate to be compatible with the controllers on the market. With the DP8461/65 in the drive, and associated components set for the drive's data rate, it no longer becomes a problem to increase the data rate, assuming the controllers digital circuitry can accommodate the change. This will allow the manufactures to increase the bit density and therefore the capacity of their drives.

Applications of the DP8461/65 Data Separator (Continued)



TL/F/8445–22

FIGURE 18. DP8461/65 in the Controller



TL/F/8445–23

FIGURE 19. DP8461/65 in the Disk Drive

PRECAUTIONS IN BREADBOARDING AND PCB LAYOUT

The DP8461/65 contains a high performance analog PLL and certain precautions must be taken when breadboarding or designing a PCB layout. The following guidelines should be adhered to when working with the DP8461/65:

1) Do not wire wrap.
2) Keep component lead lengths short, place components as close to pins as possible.This applies to R1, C1, R2, C_{VCO}, R_{RATE}, R_{BOOST}, C_{RATE}, C_{BOOST}, RPG1, RPG2, and CPG1.
3) Provide a good ground plane and use a liberal amount of supply bypassing. The quieter a PLL's environment, the happier it is.
4) Avoid routing any digital leads within the vicinity of the analog leads and components.
5) Keep inter-pin capacitance to a minimum; i.e., avoid running traces or planes between pins.
6) Minimize digital output pin capacitive loading to reduce current transients.

NSC has used a PC board approach to breadboarding the DP8461/65 that gives an excellent ground plane and keeps component lead lengths very short. With this setup very stable and reliable operation has been observed. Illustration of component layout is shown in *Figure 20*.

Applications of the DP8461/65 Data Separator (Continued)

ADDITIONAL NOTES

1. PG1 should be grounded to improve noise immunity.
2. 2F clock must be applied at all times; without the 2F clock, the pulse gate circuitry will not operate properly making it impossible to lock onto the incoming data stream.
3. The programming capacitor for the V_{CO} can be calculated as:

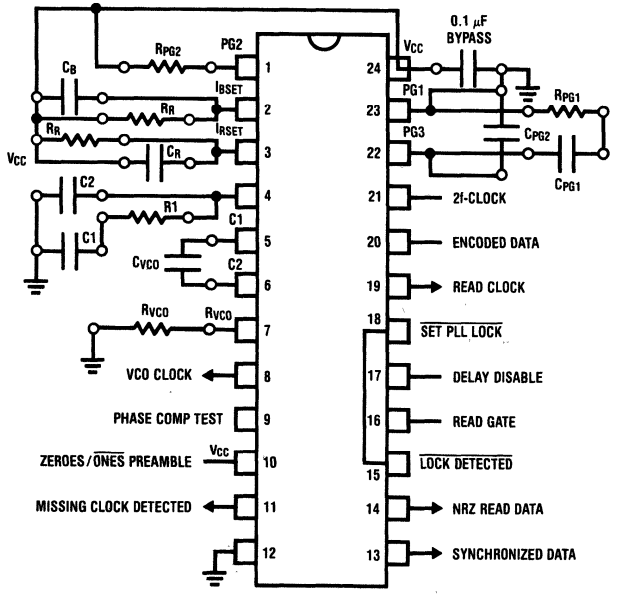
$$C_{VCO} = 1/(f_{VCO} \times R_{VCO}) - 5\ pF$$

The 5 pF value is due to parasitic and pin to pin capacitance. An additional accomodation must also be made for PC board capacitance.

4. Care must be taken in final PC board layout to minimize pin to pin capacitance, particularly in multi-layer printed circuit boards.
5. Please refer also to Precautions for Disk Data Separator Designs, NSC Application Note AN-414.

Connection Diagrams

DP8461/65

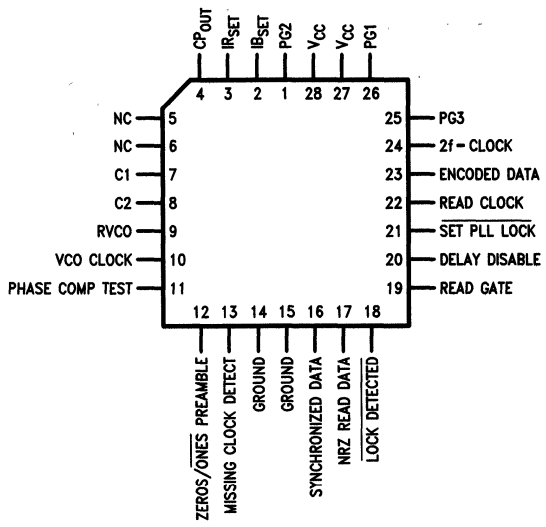


TL/F/8445-24

FIGURE 20. Recommended Component Layout

Connection Diagrams (Continued)

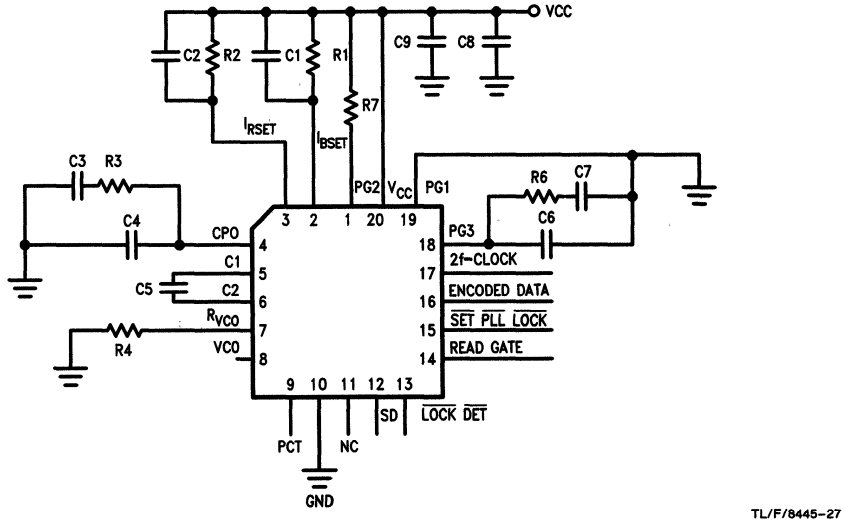
Plastic Chip Carrier



TL/F/8445–26

Order Number DP8461V or DP8465V
See NS Package Number V28A

DP8451/DP8455



TL/F/8445–27

National Semiconductor

DP8459 All-Code Data Synchronizer

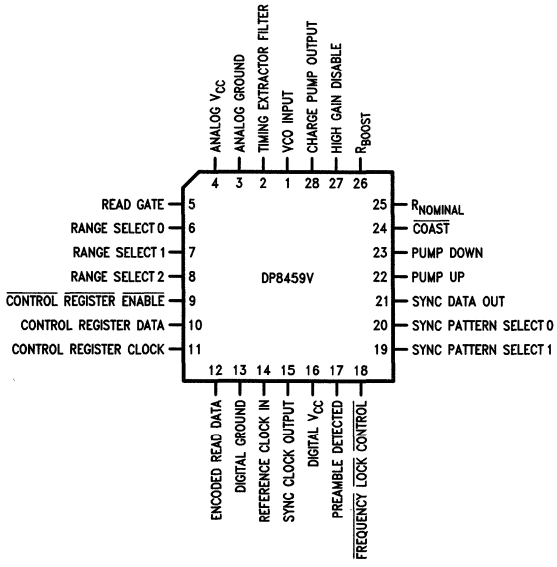
General Description

The DP8459 Data Synchronizer is an integrated phase locked loop circuit which has been designed for application in magnetic hard disk, flexible (floppy) disk, optical disk, and tape drive memory systems for data re-synchronization and clock recovery with any standard recording code, operating to 25 Mb/s. The DP8459 is provided in a 28-pin PCC package. Zero phase start is employed during both data and reference clock lock sequences for rapid acquisition. An optional (Customer-controlled) synchronization field frequency-acquisition feature guarantees lock, accommodating the preamble types used with GCR (Group Code Recording), MFM (Modified Frequency Modulation), the [1,N] run length limited (RLL) codes, and either of the standard 2,7 RLL codes. Precise synchronization window generation is achieved via an internal, self-aligning delay line which remains accurate independent of temperature, power supply, external component and IC process variations. The DP8459 also incorporates a digitally controlled (MICROWIRE™ bus compatible) strobe function with 5-bit resolution which allows for margin testing, error recovery routines, and precise window calibration. The PLL filter resides external to the chip, with two ports provided to allow significant design flexibility. Synchronization pattern detection circuitry issues a PREAMBLE DETECTED signal when a pre-determined length of the user-selected pattern is encountered. All digital input and output signals are TTL compatible and a single, +5V power supply is required. The DP8459V is offered as a DP8459V-10 (250 Kbit/sec thru 10 Mbits/sec) or DP8459V-25 (250 Kbits/sec thru 25 Mbit/sec), see AC Electrical Characteristics.

Features

- Fully integrated dual-gain PLL
- Zero phase start lock sequence
- 250 Kbit/sec–25 Mbit/sec data rate range
- Frequency lock capability (optional) for all standard recording codes
- Digital window strobe control, 5-bit resolution
- Two-port PLL filter network
- PLL free-run (Coast) control for optical disk defects
- Synchronization pattern (preamble lock) detection
- Non-glitching multiplexed read/write clock output
- +5V supply
- DP8459 supplied in 28-pin plastic chip carrier (PCC) and 40-pin TapePak packages

Connection Diagrams



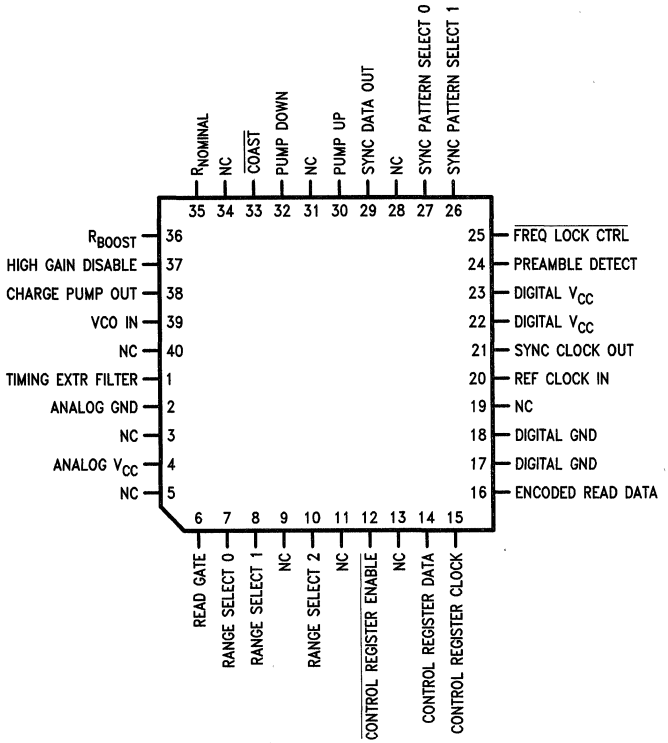
TL/F/9322–6

FIGURE 1. DP8459 in 28-Pin Plastic Chip Carrier (PCC) V-Type Package Order Number DP8459V-10 or DP8459V-25

Contents

Connection Diagrams (Continued)

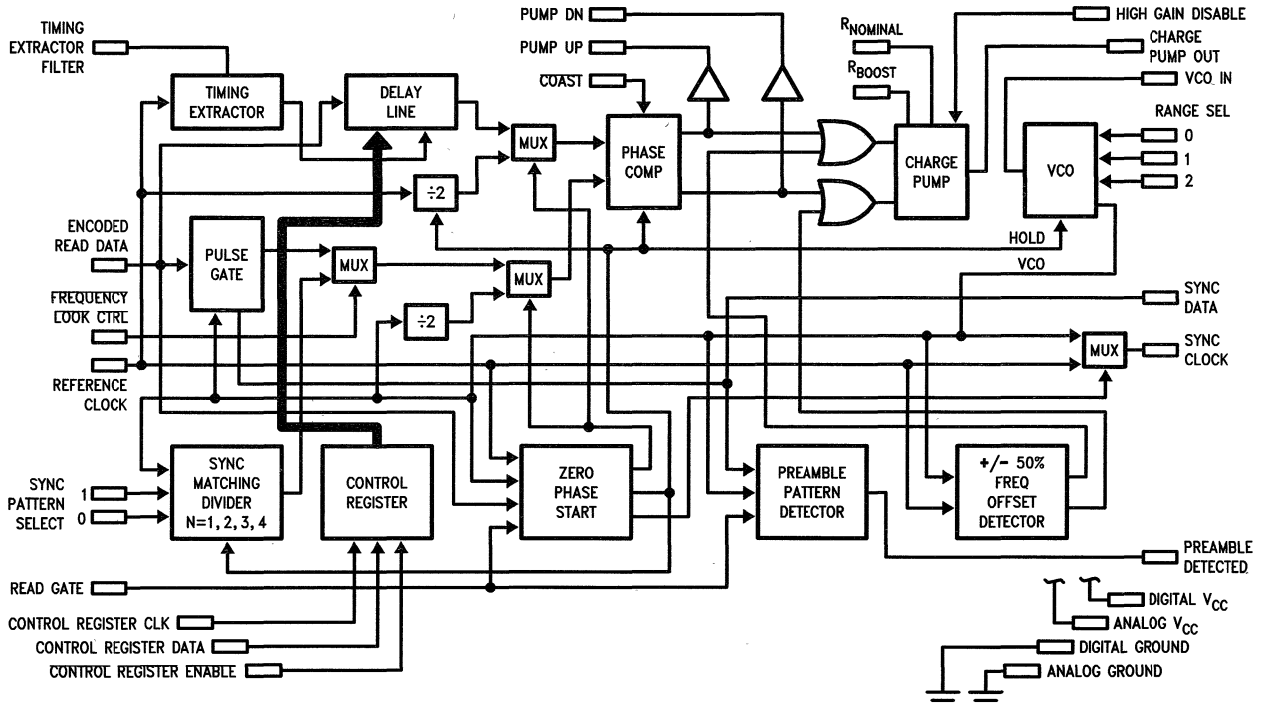
TapePak®



TL/F/9322–39

Top View

Order Number DP8459TP-10 or DP8459TP-25
See NS Package TP40A



TL/F/9322–8

FIGURE 2. DP8459 System Block Diagram

1.0 Pin Descriptions

DP8459 28-pin PCC package

Pin #	
POWER SUPPLY	
16	**DIGITAL V_{CC}:** 5.0V ±5%. (Note 1)
4	**ANALOG V_{CC}:** 5.0V ±5%. (Note 1)
13	**DIGITAL GROUND.**
3	**ANALOG GROUND.**
TTL LEVEL LOGIC INPUTS	
5	**READ GATE (RG):** Read mode control input, active high (logical-one). Assertion causes the PLL to lock to the ENCODED READ DATA, employing a zero phase start routine. Deassertion causes the PLL to lock the REFERENCE CLOCK input, also employing a zero phase start routine. READ GATE timing is allowed to be fully asynchronous.
6, 7, 8	**RANGE SELECT 0, 1, 2 (RS0, RS1, RS2):** Control the operating frequency range of the VCO. A 2:1 continuously variable sub-range is available within each of 6 allowed selections, enabling the VCO to operate at any frequency within a 96:1 range from 500 kHz to 50 MHz.
9	**$\overline{\text{CONTROL REGISTER ENABLE}}$ ($\overline{\text{CRE}}$):** A logical Low level allows the CONTROL REGISTER CLOCK to clock data into the Control Register via the CONTROL REGISTER DATA input; a logical HIGH level latches the register data and issues the information to the appropriate circuitry.
10	**CONTROL REGISTER DATA (CRD):** Control Register data input.
11	**CONTROL REGISTER CLOCK (CRC):** Negative edge triggered Control Register clock input.
12	**ENCODED READ DATA (ERD):** Incoming TTL-level data derived from the storage media; issued from a pulse detector circuit. Each positive edge represents a single recorded code bit.
14	**REFERENCE CLOCK (RFC):** A reference frequency input **required** for DP8459 operation. The RFC frequency must be accurate and highly stable (crystal or servo derived) and equivalent to the 2F frequency for the MFM or [2,7] codes (i.e., equal to, but not derived from the VCO frequency).
18	**$\overline{\text{FREQUENCY LOCK CONTROL}}$ ($\overline{\text{FLC}}$):** Selects or de-selects the frequency lock function during a READ operation. Has no effect with READ GATE deasserted; frequency lock is automatically employed for the full duration of time READ GATE is deasserted regardless of the level of the $\overline{\text{FLC}}$ input. With READ GATE high and $\overline{\text{FLC}}$ low (logical-zero) the PLL is forced to lock to the pattern frequency selected via the SYNC PATTERN SELECT inputs. When high (logical-one) frequency lock action is terminated and the PLL employs a pulse gate to accommodate random disk data patterns. $\overline{\text{FLC}}$ may be tied to PREAMBLE DETECTED output pin for self-regulated frequency lock control. $\overline{\text{FLC}}$ timing is allowed to be fully asynchronous.
20 19	**SYNC PATTERN SELECT 0, 1 (SP0, SP1):** Control inputs for selection of the preamble type being employed. These inputs determine the pattern to which the PLL will frequency-lock during preamble acquisition (if frequency lock is employed) and for which the PREAMBLE DETECTED circuitry searches.
24	**$\overline{\text{COAST}}$ ($\overline{\text{CST}}$):** Control for Coast function. The Coast function may be activated when READ GATE is either high or low. When the $\overline{\text{COAST}}$ input is low (logical-zero), the phase comparator is disabled and held in a cleared state, allowing the VCO to coast regardless of ENCODED READ DATA input activity (READ GATE high) or REFERENCE CLOCK input activity (READ GATE low). No other circuit functions are disturbed. When high (logical-one), the phase comparator operates normally.
27	**HIGH-GAIN DISABLE (HGD):** Charge Pump gain switch control. When low (logical-zero), the charge pump input current is the combined value of the currents at both R_{BOOST} and $R_{NOMINAL}$ pins. When high (logical-one), charge pump input current is taken from the $R_{NOMINAL}$ pin only. HGD may be tied either to READ GATE or PREAMBLE DETECTED for self-regulated gain control.

Note 1: These pins should *always* be tied together; they are not intended to be used with separate power supplies.

1.0 Pin Descriptions (Continued)

DP8459 28-pin PCC package

Pin #	
TTL LEVEL LOGIC OUTPUTS	
15	**SYNCHRONIZED CLOCK (SCK):** Issues the VCO signal following READ GATE assertion and completion of zero phase start sequence; issues REFERENCE CLOCK input signal when READ GATE is deasserted. Multiplexer switching is achieved without glitches.
17	**PREAMBLE DETECTED (PDT):** Issues a high level (logical-one) following assertion of READ GATE, completion of the zero phase start sequence, and the detection of approximately 32 sequential pulses of 1T, 2T or 3T period preamble, or 16 sequential pulses of 4T period preamble, depending on state of SYNC PATTERN SELECT inputs (T = VCO period). Following preamble detection, the output remains latched high until de-assertion of READ GATE. The PDT output will be at a logical zero state whenever READ GATE is inactive.
21	**SYNCHRONIZED DATA (SD):** A reconstructed replica of the ENCODED READ DATA signal, time-stabilized and synchronized to the SYNCHRONIZED CLOCK output.
22	**PUMP UP (PU):** Active HIGH whenever the phase comparator issues a pump-up signal to the charge pump. The PU pin is an open-emitter output requiring an external passive pull down resistor whenever in active use. The output should be allowed to float when not needed.
23	**PUMP DOWN (PD):** Active HIGH whenever the phase comparator issues a pump-down signal to the charge pump. The PD pin is an open-emitter output requiring an external passive pull down resistor whenever in active use. The output should be allowed to float when not needed.
ANALOG SIGNAL PINS	
28	**CHARGE PUMP OUTPUT:** The output of the high-speed, switching bi-directional current source circuitry of the charge pump. The external, passive PLL filter network is established between this pin, the VCO INPUT pin, and ground.
1	**VCO INPUT:** The high-impedance control voltage input to the voltage controlled oscillator (VCO). The external, passive PLL filter network is established between this pin, the CHARGE PUMP OUTPUT pin, and ground.
2	**TIMING EXTRACTOR FILTER:** A pin for the connection of external, passive components employed to stabilize the delay line timing extraction circuitry. Delay accuracy is not a function of external component values or tolerances.
25	**$R_{NOMINAL}$:** A resistor is tied between this pin and V_{CC} to set the charge pump *nominal* operating current. The current is internally multiplied by 2 for charge pump use.
26	**R_{BOOST}:** A resistor is tied between this pin and V_{CC} to set the charge pump *boost* (or adder) current. The R_{BOOST} resistor is effectively paralleled with the $R_{NOMINAL}$ resistor when the HIGH GAIN DISABLE input is inactive (logical-zero); thus the sum of the resistor currents sets the total input current. The input current is multiplied by 2 within the charge pump circuitry.

2.0 Circuit Operation

In the non-Read mode, the DP8459 PLL is locked to the REFERENCE CLOCK signal. This permits the VCO to remain at a frequency very close to the encoded data clock rate while the PLL is "idling" and thus will minimize the frequency step and associated lock time encountered at the initiation of lock to ENCODED READ DATA. Frequency acquisition is employed in the non-Read mode to ensure lock.

Note: The REFERENCE CLOCK signal is employed by circuitry which sets the time delay of the internal delay line. This requires the REFERENCE CLOCK signal to be present *at all times* at a stable and accurate frequency for proper DP8459 operation.

At the assertion of READ GATE, which is allowed to be done asynchronously (no timing requirements), and following the completion of two subsequent VCO cycles, the DP8459 VCO is stopped momentarily and restarted in accurate phase alignment with the second data bit which arrives following the VCO pause. This minimization of phase misalignment between the ENCODED READ DATA and the VCO (referred to as zero phase start, or ZPS) significantly reduces data lock acquisition time.

The DP8459 incorporates a preamble-specific frequency acquisition feature which may be employed at the user's option. The frequency acquisition feature is intended specifically for use within hard or pseudo-hard sectored systems where READ GATE is asserted only within a preamble. With the READ GATE active (logical-one) and the $\overline{\text{FREQUENCY LOCK CONTROL}}$ ($\overline{\text{FLC}}$) input active (logical-zero), the DP8459 will be forced to lock to the exact preamble frequency selected at the SYNC PATTERN SELECT inputs. The frequency discriminating action of the PLL provided in this mode produces a lock-in range equivalent to the available VCO operating range and thus eliminates the possibility of fractional-harmonic lock. Windowing (pulse gate action; see Pulse Gate, Section 2.1) is not employed in the frequency acquisition mode and thus quadrature lock is prevented (see National Semiconductor Application Note AN-414, APPS Mass Storage Handbook #1, 1986, for an explanation of typical false lock modes). The DP8459 will remain in the frequency acquisition mode until the $\overline{\text{FLC}}$ input is deactivated (logical-one). In ordinary hard sectored or pseudo-hard sectored operation, the PREAMBLE DETECTED (PDT) output is tied to the $\overline{\text{FLC}}$ input for automatic switching from frequency acquisition to phase lock following internal detection of the selected preamble by the DP8459. The Customer may choose to intervene in this path and extend the frequency lock period. However, the DP8459 *must* be placed in the phase lock mode ($\overline{\text{FLC}}$ deactivated—logical-one) prior to encountering the end of the preamble, or loss of lock will result. Switching of the $\overline{\text{FLC}}$ input may be done asynchronously (no set-up or hold timing requirements).

The PREAMBLE DETECTED (PDT) output will become active (logical-one) following READ GATE assertion, completion of the ZPS sequence and the subsequent detection of approximately 32 ENCODED READ DATA (ERD) pulses of the 1T, 2T or 3T preamble types, or 16 ENCODED READ DATA (ERD) pulses of the 4T preamble type (see specification tables), and will remain active (logical-one) until deassertion of READ GATE.

The Customer has the option of employing an elevated PLL bandwidth during preamble acquisition (or at any other time) for an extended capture range. An R_{BOOST} pin is provided to allow for an increase in charge pump gain above the level set by the $R_{NOMINAL}$ pin. When the HIGH GAIN DISABLE pin (HGD) is inactive (logical-zero), the R_{BOOST} resistor is electrically paralleled with the $R_{NOMINAL}$ for an elevated charge pump gain. When HIGH GAIN DISABLE is active (logical-one), only the $R_{NOMINAL}$ resistor is employed to set the pump current. The Charge Pump throughput gain is $I_{CPO} = 2 \times I_{Rp}$ where $I_{Rp} = 0.25V_{CC}/R_p$, $R_p = R_{NOM}$ with HGD high, and $R_p = R_{NOM} || R_{BOOST}$ with HGD low. The Customer may choose to configure the system for high gain prior to DP8459 preamble detection by tying the HGD pin to the PDT output pin, or for high gain only during REFERENCE CLOCK lock by tying the HGD pin to the READ GATE pin. Other configurations may be employed, if desired.

The DP8459 issues a clock waveform from the SYNCHRONIZED CLOCK output which is derived from the REFERENCE CLOCK input when the READ GATE is inactive (logical-zero), and from the VCO signal following READ GATE assertion (logical-one) and completion of the zero phase start sequence. The REFERENCE CLOCK signal is issued from the SYNCHRONIZED CLOCK output during non-Read activity and may be used as a write clock, if desired. Once data lock is achieved and the SYNCHRONIZED CLOCK output is issuing VCO, the SYNCHRONIZED DATA output and the SYNCHRONIZED CLOCK output are held in a fixed, specified timing relationship for use by decoding/deserializing circuitry. The SYNCHRONIZED CLOCK output multiplexer switching is achieved without glitches, i.e., no pulse is narrower than 50% of the VCO or REFERENCE CLOCK period.

The DP8459 provides a $\overline{\text{COAST}}$ control input which serves to clear the phase comparator and disable charge pump action whenever taken to an active, logical-zero level. This function is made available to allow the PLL to be set to free-run, undisturbed, while a detectable defect is being read from the media in a region where re-initiation of the lock procedure is impractical (e.g., data field). External data controller circuitry is responsible for the detection of the defect and issuance of the $\overline{\text{COAST}}$ command. The primary application of this feature is expected to be optical disk bright-spot avoidance, though it will lend itself to other applications as well.

As in the previous family of National Semiconductor data separators/synchronizers, the DP8459 provides phase comparator activity information to the Customer. The phase comparator's pump-up and pump-down outputs are brought out to separate pins, PUMP UP (PU) and PUMP DOWN (PD). The outputs are of the open-emitter type, requiring an external "pull-down" resistor when in active use. These outputs serve to indicate the relative displacement of the current data bit with respect to the internal VCO phase (window center). When in completely stabilized lock with no bit displacement, the output(s) will issue a pulse of a finite, minimum-valued width for each arriving data pulse. If any data pulse is displaced with respect to the VCO phase, the corresponding output pulse will widen by an amount equivalent to the bit displacement. These output signals may be integrat-

2.0 Circuit Operation (Continued)

ed over time and employed to determine the average magnitude of media bit shift. Additionally, the pulse widening/narrowing effect bit displacement has on the PU/PD outputs produces an amplitude modulation of the output's waveform. The waveform envelope, when observed with a relatively slow oscilloscope time base, can be employed for observation of PLL dynamics. This is particularly useful if intrusive probing of the PLL filter nodes is not desirable.

It is strongly recommended that the PU/PD outputs be left "floating" (unconnected to any net or circuit element, including the output pull-down resistor) in any application where they are not specifically needed. This will serve to minimize unnecessary, spurious digital switching transients in the vicinity of the DP8459, and thus improve noise performance.

The DP8459 provides a wide operating data rate range to facilitate use within a broad base of applications, including multiple data rate systems or constant density recording (CDR). In order to achieve the specified 250 kbit /sec to 25 Mbit/sec span, the operation of the VCO has been divided into 6 contiguous frequency sub-ranges, with approximately a 2:1 ratio between adjacent range selections. Three inputs are provided for selecting of the sub-ranges, RANGE SELECT 0, 1 and 2. Some code type restrictions have been placed on the higher ranges of operating VCO frequency. See *Figure 3* for the operating data rate truth table and allowed code type versus VCO range selection.

The DP8459 allows for flexible synchronization window strobe control. The inputs CONTROL REGISTER DATA (CRD), CONTROL REGISTER CLOCK (CRC), and CONTROL REGISTER ENABLE ($\overline{CRE}$) are configured to permit interfacing of the DP8459 to the MICROWIRE™ (or equivalent) bus for entry of strobe information. Information is serially shifted into the CONTROL REGISTER via the CRD and CRC pins whenever the $\overline{CRE}$ pin is active (logical-zero). When the $\overline{CRE}$ pin is inactive (logical-one), CRD and CRC are ignored. The strobe function allows the Customer to shift the synchronization window in 31 equal steps of magnitude $t_S = M \times [1.8\% \times \tau_{VCO}]$ from approximately 27% early to 27% late with respect to nominal window position. This function may be employed for margin testing (eg., approximately ±12%) or error recovery read re-try operations (eg., approximately ±2% to ±3%). Additionally, this feature allows the Customer to align the center of the synchronization window to within one half strobe step of ideal, regardless of the initial performance or specification of the DP8459. This window centering function may be performed completely within the drive system itself (auto-alignment) given the employment of an intelligent window alignment routine. Such a routine would be configured to determine the maximum error free early and late window positions via the strobe function, and then would fix the DP8459 window in the arithmetic mean position (Section 4.3.3). See *Figure 4* for a window strobe truth table.

Note: In all DP8459 applications, provision must be made to load the appropriate information into the Control Register.

RANGE SELECT Input (Note 1)			VCO Range MHz	Equivalent NRZ Data Rate	Minimum N (Allowed Code Type)			
2	1	0		MFM or 2,7 (Mbit/sec)	1 (GCR)	2 (MFM; 1, N)	3 (2,7)	4 (2,7)
1	1	X	0.50 ≤ Fvco ≤ 1.25	0.250≤ Fnrz ≤ 0.625	√	√	√	√
1	0	1	1.25 < Fvco ≤ 2.5	0.625< Fnrz ≤ 1.25	√	√	√	√
1	0	0	2.5 < Fvco ≤ 5	1.25< Fnrz ≤ 2.5	√	√	√	√
0	1	1	5 < Fvco ≤ 10	2.5< Fnrz ≤ 5	√	√	√	√
0	1	0	10 < Fvco ≤ 20	5< Fnrz ≤ 10	N/A	√	√	√
0	0	X	20 < Fvco ≤ 50 (Note 3)	10< Fnrz ≤ 25	N/A	√	√	√

Note 1: N/A—Not Allowed.

Note 2: Operation slightly beyond listed range boundaries may be acceptable in some applications. At or near range boundaries, range selection should be made to place the operating frequency near the UPPER boundary; e.g., use RS2 = 0, RS1 = 1, and RS0 = 0 for 10 Mb/s.

Note 3: 20 MHz < Fvco ≤ 38 MHz for 1, N codes.

FIGURE 3. Code Type Allowance Versus VCO Frequency Range

2.0 Circuit Operation (Continued)

Strobe Bit 4	3	2	1	0	Strobe Word M	Window Strobe T_S (Typical)
0	1	1	1	1	−15	$-0.270 \times \tau_{VCO}$
0	1	1	1	0	−14	$-0.252 \times \tau_{VCO}$
0	1	1	0	1	−13	$-0.234 \times \tau_{VCO}$
0	1	1	0	0	−12	$-0.216 \times \tau_{VCO}$
0	1	0	1	1	−11	$-0.198 \times \tau_{VCO}$
0	1	0	1	0	−10	$-0.180 \times \tau_{VCO}$
0	1	0	0	1	−9	$-0.162 \times \tau_{VCO}$
0	1	0	0	0	−8	$-0.144 \times \tau_{VCO}$
0	0	1	1	1	−7	$-0.126 \times \tau_{VCO}$
0	0	1	1	0	−6	$-0.108 \times \tau_{VCO}$
0	0	1	0	1	−5	$-0.090 \times \tau_{VCO}$
0	0	1	0	0	−4	$-0.072 \times \tau_{VCO}$
0	0	0	1	1	−3	$-0.054 \times \tau_{VCO}$
0	0	0	1	0	−2	$-0.036 \times \tau_{VCO}$
0	0	0	0	1	−1	$-0.018 \times \tau_{VCO}$
0	0	0	0	0	0	0
1	0	0	0	0	0	0
1	0	0	0	1	1	$0.018 \times \tau_{VCO}$
1	0	0	1	0	2	$0.036 \times \tau_{VCO}$
1	0	0	1	1	3	$0.054 \times \tau_{VCO}$
1	0	1	0	0	4	$0.072 \times \tau_{VCO}$
1	0	1	0	1	5	$0.090 \times \tau_{VCO}$
1	0	1	1	0	6	$0.108 \times \tau_{VCO}$
1	0	1	1	1	7	$0.126 \times \tau_{VCO}$
1	1	0	0	0	8	$0.144 \times \tau_{VCO}$
1	1	0	0	1	9	$0.162 \times \tau_{VCO}$
1	1	0	1	0	10	$0.180 \times \tau_{VCO}$
1	1	0	1	1	11	$0.198 \times \tau_{VCO}$
1	1	1	0	0	12	$0.216 \times \tau_{VCO}$
1	1	1	0	1	13	$0.234 \times \tau_{VCO}$
1	1	1	1	0	14	$0.252 \times \tau_{VCO}$
1	1	1	1	1	15	$0.270 \times \tau_{VCO}$

FIGURE 4. Window Strobe Truth Table

Customers who employ the DP8459 in a system without a MICROWIRE™ (or functionally equivalent) bus configuration and who wish to fix the synchronization window in the nominal position while deselecting the test mode need only load all-zero's into the Control Register following power-up; this may be easily achieved in some system configurations (requiring no additional hardware) by tying $\overline{CRE}$ to RG, tying CRC to ERD and tying CRD to ground, providing the necessary waveforms are present for register loading prior to the first read operation.

The DP8459 provides two pins for PLL filtering purposes, CHARGE PUMP OUTPUT (CPO) and VCO INPUT (VCOI). These provide the Customer with great flexibility in fliter design, permitting high-order filter functions for optimization of PLL lock characteristics and bit jitter rejection. For basic 3rd order applications, CPO and VCOI may be tied together (single-node) with a simple lead-lag, $C||(R+C)$ filter tied between these pins and ground. More esoteric filter designs may be implemented if the pins are electrically separated and a two-port filter network is established between CPO, VCOI, and ground. National Semiconductor supplies initial PLL filter recommendations for the single-node configuration within this data sheet with the qualifying statement that they are very general in nature, intended primarily for production testing of static window margin, and are NOT optimized for any particular disk system. For optimum performance, the Customer should pursue a filter design which is individualized and tailored to the requirements of the specific system involved. This is particularly true for the two-port filtering technique. See *Figure 5* for initial single-node filter design recommendations.

2.0 Circuit Operation (Continued)

Code	MFM	MFM	MFM	2,7	2,7	Units
Rate	0.500	2	5	10	20	Mbit/sec
VCO freq.	1	4	10	20	40	MHz
Sync bytes	12	12	12	12	12	bytes
pulses/byte	8	8	8	4	4	flux tran's
sync length	192	48	19.2	9.6	4.8	μs
sync freq	0.500	2	5	5	10	MHz
N_{sync}	2	2	2	4	4	none
N_{max}/N_{min}	4/2	4/2	4/2	8/3	8/3	none
ζ_{min}	0.5	0.5	0.5	0.5	0.5	none
ζ_{max}	0.7	0.7	0.7	0.8	0.8	none
ζ_{sync}	0.7	0.7	0.7	0.7	0.7	none
ω_{sync}	35	144	353	606	1230	Krad/sec
C1	0.5	0.12	0.05	0.018	8200 pF	μF*
R1	82	82	82	150	150	Ω
C2	0.01 μF	2700	1000	510	200	pF

Note 1: Preamble (sync) natural frequency chosen yields phase error $\leq$ 0.01 radians at sync field end, given a 1% frequency step at READ GATE assertion. Rnom = Rboost = 2.4k for all above loop filter selections. HGD is tied to RG, $\overline{FLC}$ is tied to PD and CPO is tied to VCOI as well as to the loop filter components.

Note 2: Component values are listed for purposes of window specification testing and correlation. These values do not necessarily yield optimum performance in actual system applications. PLL dynamics and code characteristics are presented for Customer information and convenience only. See Section 3.1.

*Unless otherwise noted.

FIGURE 5. Test Conditions and Component Values for Static Window Truncation Testing

The DP8459 VCO is constrained at all times to operate within a frequency swing of approximately $\pm$50% of the frequency present at the REFERENCE CLOCK input. Internal frequency detector/comparator circuitry senses when the VCO overruns the 50% boundary and forces the charge pump to move the VCO back toward the REFERENCE CLOCK frequency until the 50% constraint is again satisfied—thus preventing VCO runaway in the event of loss of lock or during extended periods where ENCODED READ DATA is not present. Additionally, this technique causes the filter node voltage to behave as if a voltage clamp were present at the Charge Pump Output, preventing the control voltage, in the event of loss of lock, from drifting outside of its operating range and inadvertently extending lock recovery time.

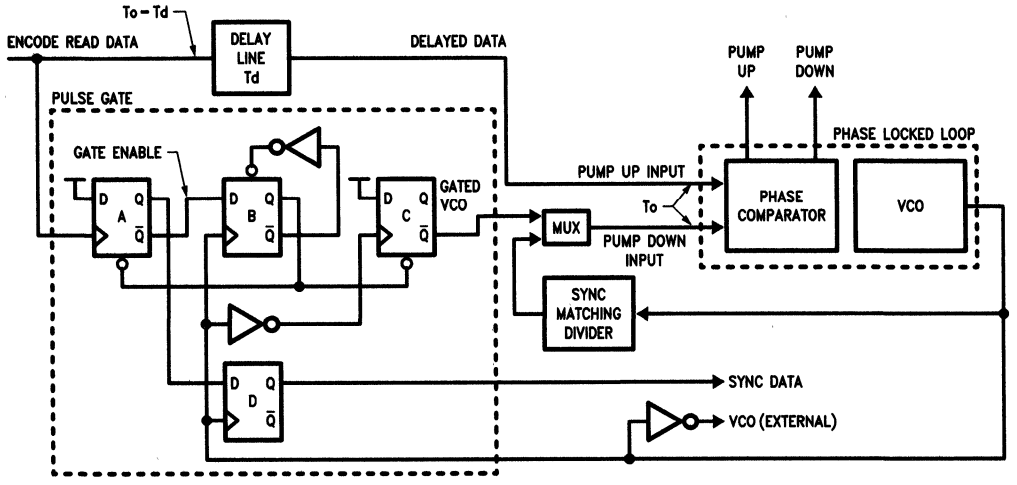
A special test mode feature has been incorporated into the DP8459 which allows a specific input pin to change function and act as an excitation source (substitute VCO) for clocking internal logic circuitry. When the last bit in the CONTROL REGISTER is taken to a logical ONE, the VCO is stopped, and the HGD input is redirected to act as a clock source for the VCO divider circuitry. Additionally, the Delay Line and Timing Extractor blocks are disabled when the Test Mode is entered, and thus the device will not function normally and should not be operated in this mode for purposes other than internal gate exercising. Further information regarding application of the Test Mode will be furnished at the Customer's request: contact National Semiconductor Logic Marketing Group or Logic Applications Group.

2.1 Functional Block Description

PULSE GATE

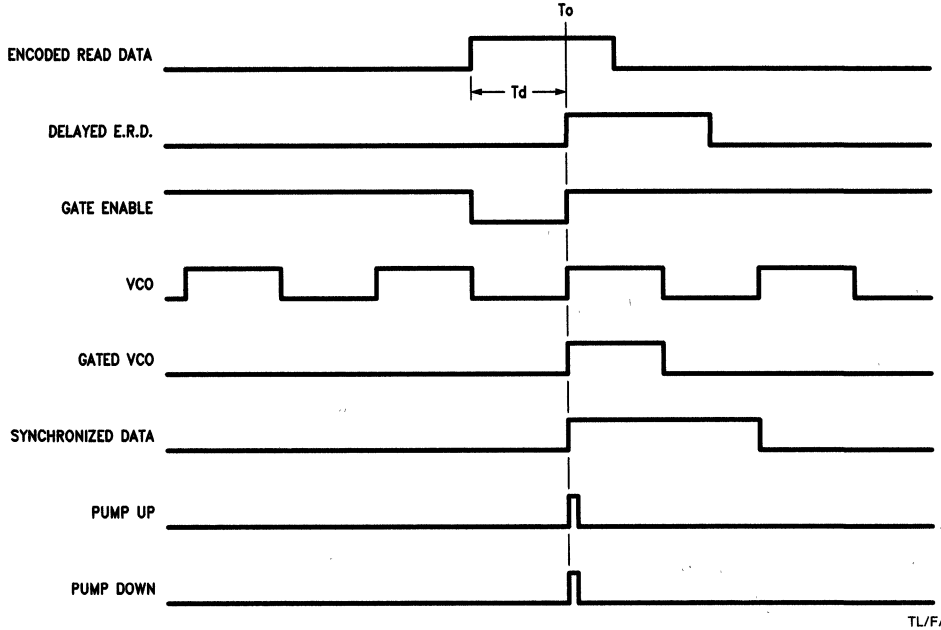
The function of the Pulse Gate within the DP8459 is twofold. First, the block contains the ECL flip-flop which captures each arriving ENCODED READ DATA bit and transmits the bit to the SYNCHRONIZED DATA output. The very high switching speed of the bit-capture ECL flip-flop minimizes the portion of window margin loss caused by flip-flop metastability at window boundaries. Second, the Pulse Gate regulates the transmission of the VCO waveform into the Phase Comparator, allowing only one VCO pulse to pass with each arriving ENCODED READ DATA pulse. See *Figure 6* for a simplified logical representation of the Pulse Gate block. The one-to-one data/VCO pulse ratio produced by the Pulse Gate permits the multiple-harmonic nature of encoded data to be accommodated by the phase/frequency comparator. During the non-Read mode or during the portion of the Read mode within which the Customer has set the $\overline{FREQUENCY\ LOCK\ CONTROL}$ pin to a logical-zero (low), the Pulse Gate is inactive (bypassed) and the VCO frequency is divided as appropriate to match the incoming frequency source (ENCODED READ DATA or the REFERENCE CLOCK input).

2.1 Functional Block Description (Continued)



TL/F/9322–11

FIGURE 6. Simplified Diagram of Window Generation Circuitry



TL/F/9322–12

FIGURE 7. Capture of Nominally Positioned ENCODED READ DATA Pulse

2.1 Functional Block Description (Continued)

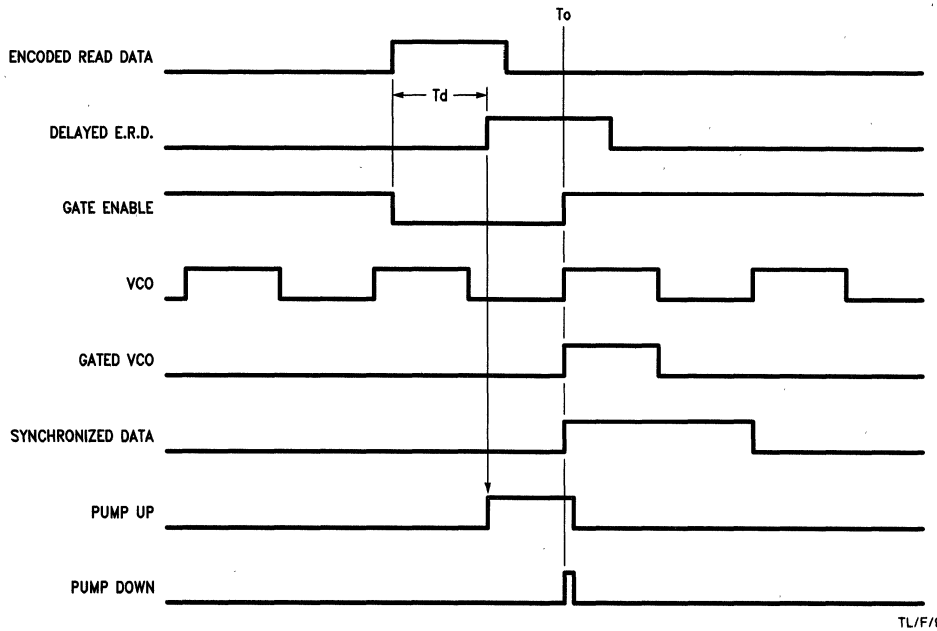


FIGURE 8. Capture of Early-Shifted ENCODED READ DATA Pulse

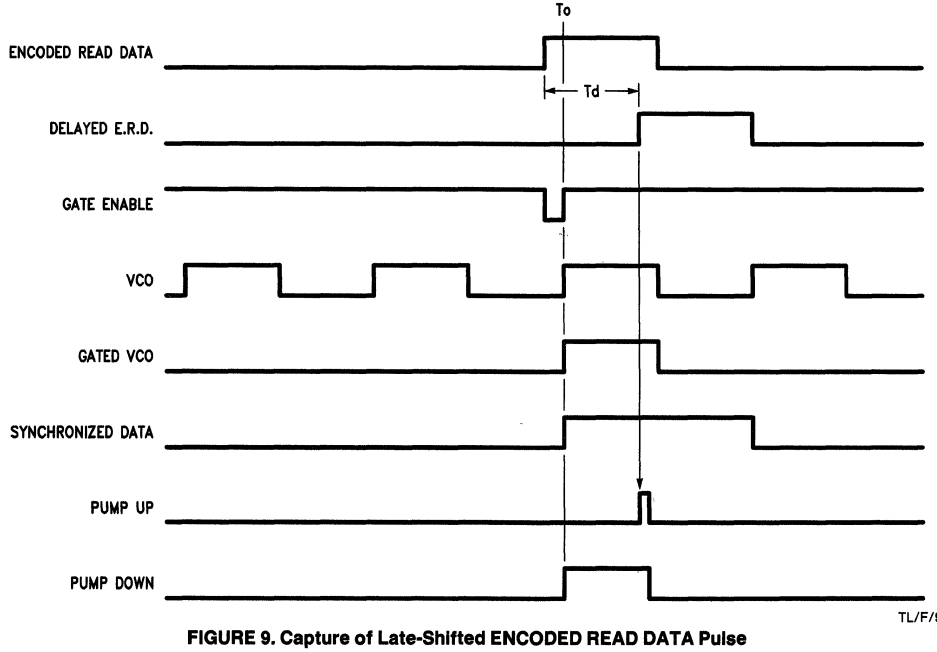


FIGURE 9. Capture of Late-Shifted ENCODED READ DATA Pulse

2.1 Functional Block Description (Continued)

DELAY LINE

The DP8459 employs an internal silicon delay line to establish synchronization window alignment. The delay is nominally equivalent to one half of the period of the REFERENCE CLOCK waveform, and is variable in fine increments via the Control Register in order to achieve the window strobe function. The Timing Extractor circuitry derives realtive timing information soley from the REFERENCE CLOCK signal and regulates the magnitude of the delay within the Delay Line. The Delay Line thus remains insensitive to the external components associated with the extractor as well as to supply voltage, temperature, and IC process variations.

TIMING EXTRACTOR

This block extracts timing information from the REFERENCE CLOCK input for use by the variable silicon delay line. External passive components (tied to the Timing Extractor Filter pin) are associated with this block, although the accuracy of the circuit's function remains independent of the general value and tolerance of the components. The resistor-capacitor net is employed by the Timing Extractor for stabilization purposes—no monostable multivibrator (one-shot) circuitry is employed by the DP8459. Note that the performance of the delay line is directly dependent upon the accuracy of the REFERENCE CLOCK input waveform. Either a crystal reference generator or a stable servo clock source must be applied to this input. Multiplexing of the REFERENCE CLOCK waveform between read operations (within multiple data rate systems) is acceptable, although sufficient Timing Extractor stabilization time must be allowed following any perturbation at this pin before a read operation may be performed (see *Figure 10* for timing table).

PHASE COMPARATOR

The DP8459 employs a digital Phase Comparator (non-harmonic discriminator circuit) which has the capability of forcing the frequency of the PLL VCO toward the frequency of the reference input regardless of the magnitude of the frequency difference. The function of the Phase Comparator circuit can be represented in a diagrammatically simplified form as in *Figure 11*.

The Phase Comparator's action can be disabled at any time (cleared) via the $\overline{\text{COAST}}$ input pin, allowing the VCO to free-run.

CHARGE PUMP

The Charge pump is a high speed, switching, dual-gain, bi-directional current source whose current flow is controlled by the digital Phase Comparator circuit. The current pulses at the CHARGE PUMP OUTPUT (CPO) pin thus reflect the magnitude and sign of the phase error seen at the input of the Phase Comparator. The CPO pin is connected externally to a passive component network whose impedance translates the aggregate current into a voltage for the VCO INPUT while providing a low-pass filter function for the PLL. The matched source and sink current generators' operating currents are set via the $R_{NOMINAL}$ and R_{BOOST} pins, which are supplied current from V_{CC} through external resistors. The bias voltages at the $R_{NOMINAL}$ and R_{BOOST} pins are set to $0.75 \times V_{CC}$; the current into each of these pins is internally multiplied by 2 for Charge Pump use. The CPO current is defined as follows:

$$I_{CPO} = (V_{CC}/2)/R_{NOM}$$

HIGH GAIN DISABLE high (logical-one)

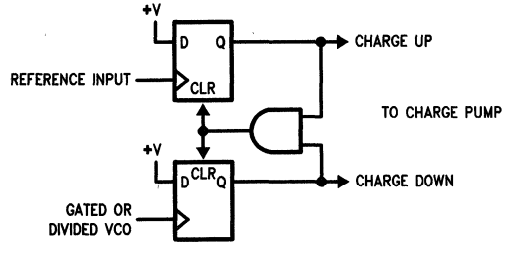
$$I_{CPO} = (V_{CC}/2)/(R_{NOM} \| R_{BOOST})$$

HIGH GAIN DISABLE low (logical-zero)

RFC Frequency	1	4	10	20	40	MHz
CT1	0.82	0.2	0.082	0.056	0.027	μF
RT1	68	68	68	68	68	Ω
Settling Time	192	96	19.2	9.6	4.6	μs

Values may be interpolated for intermediate data rates. Timing Extractor settling times are given which indicate time required for the DP8459 to accommodate a change of Strobe setting from nominal selection to either extreme (early/late), or vice versa, to within approximately 1% of final value.

FIGURE 10. TIMING EXTRACTOR FILTER Component Values for Various Data Rates



TL/F/9322–17

FIGURE 11. Simplified Digital Phase-Frequency Comparator

2.1 Functional Block Description (Continued)

VOLTAGE CONTROL OSCILLATOR (VCO)

The DP8459 VCO is comprised of two portions—a self contained, high frequency oscillator (no external components) whose frequency is regulated by the voltage at the VCO INPUT pin, and a programmable modulus digital divider. The oscillator is only required to operate over approximately a 2:1 frequency range; the divider modulus is programmable in factors of 2. The two blocks work in conjunction to achieve a continuous range of equivalent VCO operating frequencies from 500 kHz to 50 MHz. (See *Figure 12.*)

CONTROL REGISTER

Within the DP8459, the Control Register is a MICROWIRE compatible, 6-bit shift register block with bits 0 through 4 employed to control the window strobe function and bit 5 employed to regulate the device test mode (see *Figures 13* and *14*). Information is serially shifted into the Control Register via the CRD and CRC (negative edge clock) pins whenever the $\overline{CRE}$ pin is active (logical-zero). When the $\overline{CRE}$ pin is inactive (logical-one), CRD and CRC are ignored. *Figure 3* shows the truth table for the VCO range select function; *Figure 4* shows the truth table for the window strobe function.

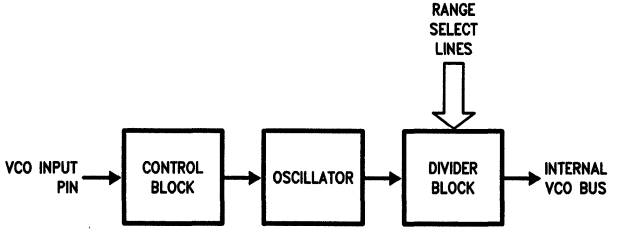


FIGURE 12

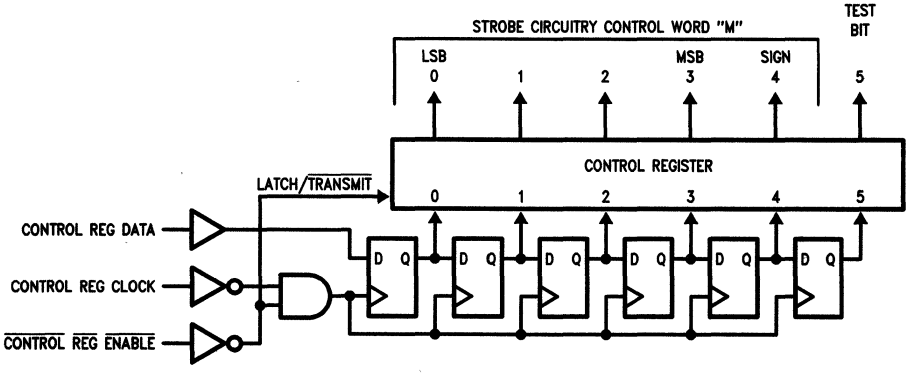


FIGURE 13. Control Register

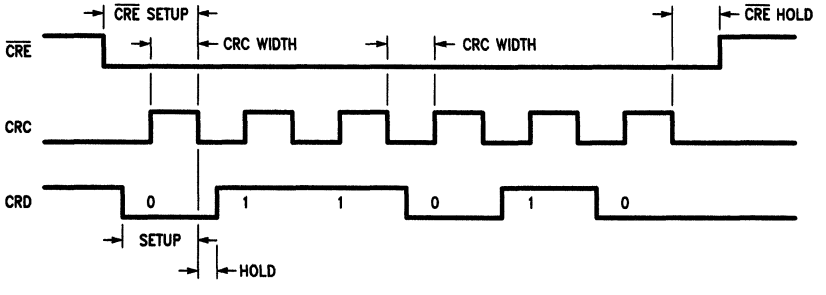


FIGURE 14. Microwire Compatible Control Register Serial Load Timing Diagram

2.1 Functional Block Description (Continued)

SYNCHRONIZATION FIELD MATCHING DIVIDER

The Synchronization field Matching Divider is a programmable modulus counter employed for implementation of the preamble frequency lock function. It is placed in the VCO feedback path to match the relative frequency of the VCO seen at the Phase Comparator to the frequency of the ENCODED READ DATA (preamble) during the read operation whenever the $\overline{\text{FREQUENCY LOCK CONTROL}}$ input is active (logic-zero). The modulus of the divider, M, is determined by the states of the SYNC PATTERN SELECT 0 and 1 inputs, as defined by the table in *Figure 15*.

Sync Pattern Select		Sync Matching Divider Modulus M	Expected Code Preamble
1	0		
0	0	1	GCR
0	1	2	MFM; 1,N
1	0	3	2,7
1	1	4	2,7

FIGURE 15. SYNC PATTERN SELECT Input Truth Table

Prior to the assertion of READ GATE, the divider is held in a known count state and is enabled at the end of the zero phase start sequence in correct phase relationship with the ENCODED READ DATA. Re-assertion (logical zero) of the $\overline{\text{FREQUENCY LOCK CONTROL}}$ pin within a read operation (following the normal $\overline{\text{FLC}}$ deassertion after lock is achieved) is permissible; however, it should be noted that the initial phase error of the Synchronization Field Matching Divider with respect to the ENCODED READ DATA at $\overline{\text{FREQUENCY LOCK CONTROL}}$ re-assertion may be as large as $M \times \tau_{VCO}$ in magnitude, possibly resulting in an extended PLL settling time.

ZERO PHASE START

The function of the zero phase start (ZPS) block is to clear the Phase Comparator and freeze the VCO in a known phase when a transition occurs at the READ GATE input (either high or low), and restart the VCO in a precise, controlled phase with respect to the newly selected input (ENCODED READ DATA or REFERENCE CLOCK ÷ 2, respectively). The ZPS circuit also resets the count state of the Synchronization field Matching Divider in anticipation of locking to specific preamble information (when frequency lock is being employed), and controls the operation of the REFERENCE CLOCK multiplexer. ZPS operation at READ GATE assertion is aimed at optimizing initial window alignment and thus minimizing initial phase step and the resulting phase lock acquisition time. ZPS is also employed at deassertion of READ GATE; however, the ZPS phase alignment for the REFERENCE CLOCK signal at READ GATE deassertion has been made less stringent than for ENCODED READ DATA at READ GATE assertion.

PREAMBLE PATTERN DETECTOR

The Preamble Pattern Detector block has a pattern-specific recognition circuit keyed to search the ENCODED READ DATA for the pattern selected at the SYNC PATTERN SELECT inputs. The pattern search begins following the assertion of READ GATE and the completion of the zero phase start sequence, and continues until approximately 32 uninterrupted ENCODED READ DATA pulses of the 1T, 2T or 3T pattern have been detected, or until 16 uninterrupted ENCODED READ DATA pulses of the 4T pattern have been detected (see specification tables). When this event occurs, the PREAMBLE DETECTED output becomes active high (logical-one). The output will then remain latched in the high state until READ GATE is deasserted. The PREAMBLE DETECTED output may be tied to the HIGH GAIN DISABLE input to regulate the gain of the PLL during the preamble lock sequence, and/or tied to the $\overline{\text{FREQUENCY LOCK CONTROL}}$ input for self-regulation of frequency acquisition in hard or pseudo-hard sectored systems.

±50% VCO FREQUENCY OFFSET DETECTOR

The Frequency Offset Detector is employed to constrain the VCO frequency swing, preventing VCO runaway associated with standard, wide-range voltage controlled oscillators. The circuitry will sense the relative difference between the REFERENCE CLOCK frequency and the VCO frequency, sending a "charge-up" signal to the Charge Pump to correct the VCO should a limit of approximately −50% in frequency differential (VCO w.r.t. REF CLOCK) be exceeded, and sending a "charge-down" signal to the Charge Pump to correct the VCO should a limit of approximately +50% in frequency differential be exceeded. The resulting voltage-clamping action at the filter node(s) also prevents out-of-range control voltage straying and thus speeds lock recovery.

SYNCHRONIZATION CLOCK OUTPUT MULTIPLEXER

This block issues the VCO signal following READ GATE assertion and completion of the zero phase start sequence, and issues the REFERENCE CLOCK input signal when the READ GATE is deasserted. Multiplexer switching is achieved without glitches. The output is intended to be used both for read and write clock purposes. (Please note output loading recommendations for this pin in Section 6.)

2.2 SPECIFICATION TABLES

Absolute Maximum Ratings

If Military/Aerospace specified devices are required, please contact the National Semiconductor Sales Office/Distributors for availability and specifications.

Supply Voltage	7V
TTL Inputs	7V
Output Voltages	7V
Input Current (R_{NOM}, R_{BOOST}, CPO, VCOI, TEF)	2 mA
Storage Temperature	−65°C to +150°C
Operating Temperature Range	0°C to +70°C
ESD Susceptibility (Note 3)	1500V

Operating Conditions

Symbol	Parameter	Conditions	Min	Typ	Max	Units
V_{CC}	Supply Voltage		4.75	5.00	5.25	V
T_A	Ambient Temperature		0	25	70	°C
I_{OH}	High Logic Level Output Current	SYNC CLOCK Others			−2000 −400	μA
I_{OL}	Low Logic Level Output Current (Note 1)	SYNC CLOCK Others			20 8	mA
V_{IH}	High Logic Level Input Voltage		2			V
V_{IL}	Low Logic Level Input Voltage				0.8	V
f_{NRZ}	Operating Data Rate Range		0.25		25	Mb/s
$t_{PW\text{-}RFC}$	Width of REFERENCE CLOCK, High or Low		8			ns
$t_{PW\text{-}ERD}$	Width of ENCODED READ DATA		12 High 18 Low			ns
$t_{PW\text{-}CRE}$	Width of CONTROL REGISTER ENABLE, High or Low (Note 2)		40			ns
$t_{SU\text{-}CRD}$	CONTROL REGISTER DATA Set-Up Time with Respect to CRC (Note 2)		20			ns
$t_{H\text{-}CRD}$	CONTROL REGISTER DATA Hold Time with Respect to CRC (Note 2)		10			ns
$t_{SU\text{-}\overline{CRE}}$	CONTROL REGISTER ENABLE Set-Up Time with Respect to CRC (Note 2)		20			ns
$t_{H\text{-}\overline{CRE}}$	CONTROL REGISTER ENABLE Hold Time with Respect to CRC (Note 2)		20			ns
$t_{PW\text{-}CRC}$	CONTROL REGISTER CLOCK Pulse Width Positive or Negative (Note 2)		40			ns
I_{CPIN}	Combined R_{NOM} & R_{BOOST} Input Current				1000	μA

Note 1: PUMP UP and PUMP DOWN outputs have no current sinking capability and thus are excluded from this specification.

Note 2: Parameter guaranteed by correlation to characterization data. No outgoing test performed.

Note 3: Human body model; 120 picofarads through 1.5 kΩ.

AC Electrical Characteristics Over recommended V_{CC} and operating temperature range.

Symbol	Parameter	Min	Typ	Max	Units
t_{STOP}	SYNC CLOCK Negative Transitions following READ GATE until Data Lock ZPS Sequence Begins (VCO Freezes)		2	3	—
$t_{RESTART}$	Positive ENCODED READ DATA Transitions following VCO Freeze until VCO Restarts		2		—
$t_{READ\ ABORT}$	Number of REF CLOCK Cycles following READ GATE Deactivation until REF CLOCK Lock ZPS Sequence Begins			4	—
t_T	Window Truncation (Half Window Loss); **DP8459V-10** 10 Mbit/sec (Note 1) **DP8459V-25** 20 Mbit/sec (Note 2)		 3% × τ_{VCO} 4% × τ_{VCO}	 3.0 2.5	 ns ns
ϕ Linearity	Phase Range for Charge Pump Linearity (wrt VCO)		$\pm\pi$		Radians
K_{VCO}	VCO Gain Constant	1.0 ω_O	1.2 ω_O	1.6 ω_O	Rad/Sec V
$f_{MAX\ VCO}$	VCO Maximum Frequency; RS0 = RS1 = RS2 = Logical ZERO	70			MHz
t_{SD0}	Time Skew between SYNC CLOCK Negative Edge and SYNC DATA Negative Edge	0		10	ns
t_{SD1}	Time Skew between SYNC CLOCK Negative Edge and SYNC DATA Positive Edge	0		10	ns
t_{ZPSR}	Zero Phase Start Trigger Bit Targeting Accuracy, READ GATE Activation (READ) (Note 4)		2		ns
t_{PWPC}	Width of PCT, PU or PD Outputs in Fully Stabilized Lock (ERD Free of Jitter); R-Pull-Down = 510Ω		10		ns
$\Delta f_{VCO}/f_{RFC}$	Automatic f_{VCO} Range Limiting		50		%
t_{HOLD}	SYNC CLOCK Rest Period (Logical One) at Assertion or De-Assertion of READ GATE	½		3	T_{VCO}
t_{PDT}	SCK Negative Edge to PREAMBLE DETECTED Positive Edge at End of Detection Sequence			25	ns
L_{PDT1}	Length of Valid 1T Preamble Pattern Required for Occurrence of PREAMBLE DETECTED	33	34	35	ERD Pulses
L_{PDT2}	Length of Valid 2T Preamble Pattern Required for Occurrence of PREAMBLE DETECTED	32	33	34	ERD Pulses
L_{PDT3}	Length of Valid 3T Preamble Pattern Required for Occurrence of PREAMBLE DETECTED	31	32	33	ERD Pulses
L_{PDT4}	Length of Valid 4T Preamble Pattern Required for Occurrence of PREAMBLE DETECTED	15	16	17	ERD Pulses
t_S	Window Strobe Time Step (M = Hex Value of Bits 0–3 in CONTROL REGISTER; Bit 4 = Sign Bit)		M × (1.8%) × t_{RFC}		ns
$t_{RFC-SCK1}$	Positive Transition Propagation Delay from REF CLOCK INPUT to SYNC CLOCK OUTPUT, READ GATE Low			15	ns
$t_{RFC-SCK0}$	Negative Transition Propagation Delay from REF CLOCK INPUT to SYNC CLOCK OUTPUT, READ GATE Low			15	ns

Note 1: The DP8459V-10 static window specification, t_T, applies only to the factory-tested 2,7-code data rate of 10 Mb/s (with RS0,1,2 = 010) and with the component values as listed in *Figures 5* and *10*, test configuration as shown in *Figure 23*, test procedure as shown in *Figure 24*, and strobe word M = −2. Significant variation in t_T as a percentage of the VCO period due to the use of other filters and data rates is not expected.

Note 2: The DP8459V-25 static window specification, t_T, incorporates the DP8459V-10 window specification and, in addition, the factory-tested 2,7-code data rate of 20 Mb/s (with RS0, 1, 2, = 000), with the component values as listed in *Figures 5* and *10*, test configuration as shown in *Figure 23*, test procedure as shown in *Figure 24*, and strobe word M = −3. Significant variation in t_T as a percentage of the VCO period due to the use of other filters and data rates is not expected.

Note 3: $I_{IN} = V_{CC}/(4 \times R_{IN})$. $R_{IN} = R_{NOM}$ (HGD High) or $R_{NOM} || R_{BOOST}$ (HGD Low).

Note 4: t_{ZPSR} (ZPS Read) gauges the accuracy with which the ZPS circuitry aligns the VCO to the triggering ERD bit internally (i.e., initial phase step) at the completion of a ZPS operation following READ GATE assertion.

DC Electrical Characteristics over recommended operating temperature range

Symbol	Parameter	Conditions	Min	Typ	Max	Units
V_{IC}	Input Clamp Voltage	V_{CC} = Min, I_I = −18 mA			−1.5	V
V_{OH}	High Level Output Voltage	V_{CC} = Min, I_{OH} = Max	V_{CC}−2V	V_{CC}−1.6V		V
V_{OL}	Low Level Output Voltage (Note 4)	V_{CC} = Min, I_{OL} = Max			0.5	V
I_{IH}	High Level Input Current	V_{CC} = Max, V_I = 2.7V			20	μA
I_{IL}	Low Level Input Current	V_{CC} = Max, V_I = 0.4V			−200	μA
I_O	Output Drive Current (Note 1)	V_{CC} = Max, V_O = 2.125V	−12		−110	mA
I_{CPO}	Charge Pump Output Current (K1)	100 ≤ I_{Rp} ≤ 1000 (Note 2)	1.7 I_{Rp}	2.0 I_{Rp}	2.5 I_{Rp}	μA
$I_{CPO-OFF}$	Charge Pump Output Inactive Current	100 ≤ I_{Rp} ≤ 1000 (Note 2)	−0.85		+0.85	μA
I_{VCOI}	VCOI Offset Current	VCOI Voltage 1.5V	−0.25		+0.25	μA
V_{RNOM}	Voltage across R-NOM Resistor	1.2 kΩ ≤ R-NOM ≤ 12 kΩ	Typ. −18%	0.26 V_{CC}	Typ. +18%	V
V_{RBST}	Voltage across R-BOOST Resistor	1.2 kΩ ≤ R-BOOST ≤ 12 kΩ	Typ. −18%	0.26 V_{CC}	Typ. +18%	V
I_{CC1}	Supply Current, Nominal Strobe	V_{CC} = Max (Note 3)			190	mA

Note 1: This value has been chosen to produce a current that closely approximates one-half of the true short-circuit output current, I_{OS}.

Note 2: $I_{Rp} = I_{NOM} + I_{BOOST}$.

Note 3: I_{CC1} is measured with the window strobe set at nominal timing (Strobe Bits 0 through 5 = 0,0,0,0,0,0); VCO operating at maximum allowed frequency within any given range selection. I_{CC} typically increases by 30 mA when the strobe is set at the maximum early position (M = −15). This is not a linear increase per step. Most of the increase occurs as the −15 step is approached. I_{CC} decreases as the window is moved late.

Note 4: PUMP UP and PUMP DOWN outputs have no current sinking capability and thus are excluded from this specification.

External Component Selection

Symbol	Parameter	Min	Typ	Max	Units
R_{NOM}	Charge Pump Nominal Operating Current Setting Resistor (Note 1)	1.2		12	kΩ
R_{BOOST}	Charge Pump Boost Current Setting Resistor (Note 1)	1.2		∞	kΩ
C_{NOM}	R_{NOM} Bypass Capacitor (Note 2)	0.01			μF
C_{BOOST}	R_{BOOST} Bypass Capacitor (Note 2)	0.01			μF
R_{PU}	PUMP UP Open Emitter Output Pull-Down Resistor	510			Ω
R_{PD}	PUMP DOWN Open Emitter Output Pull-Down Resistor	510			Ω

Note 1: The minimum allowed value for the parallel combination of R_{NOM} and R_{BOOST} is 1.2 kΩ.

Note 2: C_{NOM} and C_{BOOST} should be high quality, high frequency type.

3.0 PLL Applications: Loop Filter Design

In order to maintain greatest design flexibility for the Customer, all PLL filter components and Charge Pump gain setting elements reside external to the DP8459. All PLL dynamics are thus under the control of the system designer. The following is a brief analysis of the DP8459 PLL; Section 3.1 contains a derivation of component values based on projected requirements within an example hard disk drive system.

Figure 16 represents the DP8459 PLL in simplified form.

Mathematical gain representations for each block are:

$K_{PG} = 1/N$ Pulse Gate equivalent gain

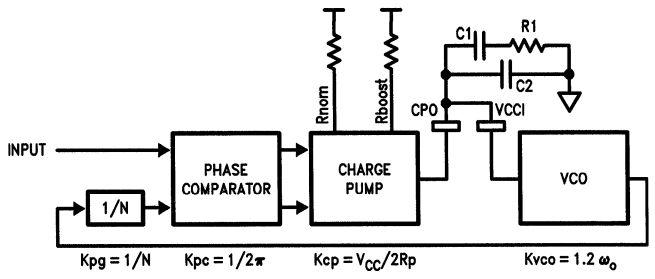
$K_{PC} = 1/(2\pi)$ Phase Comparator gain

$K_{CP} = V_{CC}/2R_p$ Charge Pump gain where

$R_p = R_{NOM}$, HGD high;

$R_p = R_{NOM} || R_{BOOST}$, HGD low

$K_{VCO} = 1.2\ \omega_O$ VCO gain (ω_O = operating center frequency)



TL/F/9322–18

FIGURE 16. Basic DP8459 Phase Locked Loop Block Diagram

3.0 PLL Applications: Loop Filter Design (Continued)

N is defined as the number VCO cycles per recorded ENCODED READ DATA pulse, or conversely, the ratio of the VCO frequency to the ENCODED READ DATA frequency. The aggregate block gain equation (excluding the loop filter) can be written as:

$$K_B = 1.2\, V_{CC}\, f_O/(2R_pN)$$

The impedance of the loop filter is

$$Z(s) = \frac{1}{sC_2} \| \left(\frac{1}{sC_1} + R_1\right) = \frac{1 + sR_1C_1}{sC_1(1 + C_2/C_1 + sR_1C_2)}$$

The open loop system response G(s) is given by

$$G(s) = \frac{K_B}{s} \times \frac{1 + sR_1C_1}{sC_1\,(1 + C_2/C_1 + sR_1C_2)}$$

This last equation reveals the PLL with this filter configuration is a third order system, which is typically difficult to analyze. However, if $C_2 << C_1$, it can be argued that the behavior of the third order loop closely resembles that of a second order system, allowing for a greatly simplified analysis.

If $C_2 << C_1$, the impedance Z(s) approximates to

$$\frac{1 + sR_1C_1}{sC_1}$$

The overall open loop gain (including the filter) is then

$$G(s) = \frac{K_B}{s} \times \frac{1 + sR_1C_1}{sC_1}$$

Substituting K_B into the equation,

$$G(s) = \frac{1.2\, f_o\, V_{CC}}{s2N} \times \frac{1 + sR_1C_1}{sR_pC_1}$$

$\tau_1 = R_pC_1$ and $\tau_2 = R_1C_1$ are the pole and zero, respectively, which govern the system response. The closed loop gain H(s) is

$$H(s) = \frac{\phi_{OUT}}{\phi_{IN}} = \frac{G(s)}{1 + G(s)}$$

Substituting,

$$H(s) = \frac{K_B\,(sR_1C_1 + 1)}{s^2C_1 + K_B\,(sR_1C_1 + 1)}$$

$$= \frac{(K_B/C_1)(sR_1C_1 + 1)}{s^2 + sK_BR_1 + K_B/C_1}$$

The second order characteristic equation can be written as follows:

$$s^2 + sK_BR_1 + K_B/C_1 = s^2 + s2\zeta\omega_n + \omega_n{}^2$$

Extracting the component values from these results,

$$C_1 = \frac{K_B}{\omega_n{}^2} = \frac{1.2\, V_{CC}\, f_o/(2R_pN)}{\omega_n{}^2}$$

$$R_1 = \frac{2\zeta\omega_n}{K_B} = \frac{2\zeta\omega_n}{1.2\, V_{CC}\, f_o/(2R_pN)}$$

$$C_2 \le (1/10)\, C_1$$

Thus, one is able to select component values in accordance with specific system requirements, i.e., with given VCO center frequency (equivalent to REFERENCE CLOCK frequency), R_p (in either high or low gain mode), N (the ratio of the VCO frequency to the ENCODED READ DATA frequency), the desired natural frequency of the loop, and the desired damping ratio.

The natural frequency and the damping ratio may be extracted from the component values to determine system behavior under various conditions (differing data patterns, i.e., varying N value; high gain or low gain; read or non-read mode):

$$\omega_n = [1.2\, V_{CC}\, f_o/2R_pNC_1)]^{0.5} \quad \text{Natural frequency}$$

$$\zeta = \omega_n\, R_1C_1/2 \quad \text{Damping ratio}$$

3.1 2,7 CODE, 10 MBIT/SEC LOOP FILTER DESIGN EXAMPLE

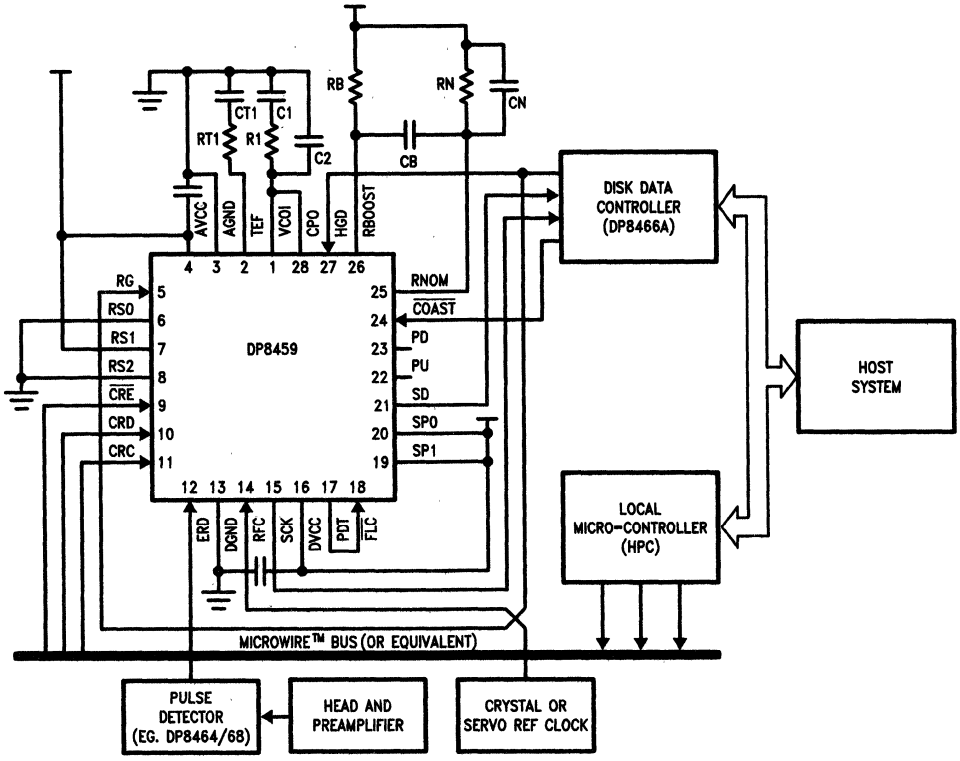
Initial Requirements and Definitions

This example illustrates a 10 MBit/sec 2,7 hard disk system employing a 4T preamble field (recorded at 1/4 the VCO frequency, i.e., N = 4). The component derivations are not meant to produce values which will be optimum for all systems employing this data rate, code, and preamble type; this exercise is for exemplary purposes only. (See National Semiconductor Advanced Peripheral Processing Solutions Mass Storage Handbook #1, 1986, AN-413, section 3.4, pages 1-43 through 1-48 for additional information regarding disk system PLL filter design.)

Although the DP8459 provides a frequency acquisition feature intended for use within the preamble, this design example will be approached so as to achieve PLL dynamics which will avoid the cycle-slipping phenomenon frequency-lock action is normally employed to accommodate. Thus, the design will be valid both for systems which do employ frequency lock as well as for those which do not. Advantages gained by the use of frequency-lock beyond that of extended lock-in range, however, such as harmonic false lock avoidance and quadrature lock avoidance, make the use of this feature strongly advisable even with the intrinsic lock-in range achieved by design in this example.

The DP8459 is configured here with the $\overline{\text{FREQ LOCK CONTROL}}$ input tied to the PREAMBLE DETECTED output, the HIGH GAIN DISABLE input tied to the READ GATE input, and the CHARGE PUMP OUTPUT tied to the VCO INPUT pin as well as to the external loop filter components (see *Figure 17*). This establishes self-regulated frequency lock control, READ GATE regulated Charge Pump gain, and single node loop filtering.

3.0 PLL Applications: Loop Filter Design (Continued)



TL/F/9322–19

FIGURE 17. DP8459 in a Typical System Configuration

3.0 PLL Applications: Loop Filter Design (Continued)

System constraints:

$f_{NRZ\ DATA}$ = 10 Mbit/sec

f_{VCO} = 20 MHz

$f_{REFERENCE\ CLOCK}$ = 20 MHz

Code type = ½ (2, 7)

N_{min} = 3 (highest recorded frequency)

N_{max} = 8 (lowest recorded frequency)

$N_{preamble}$ = 4 ($f_{preamble}$ = 5 MHz)

Preamble Length = 11 NRZ bytes (ESDI min.) = 8.8 μs (44 recorded pulses)

Disk formatting = pseudo hard sectored

The DP8459 provides a zero phase start function which minimizes the initial phase step encountered at the start of preamble lock acquisition and thus the phase stabilization time within the preamble is significantly reduced with respect to a fully random-phase lock sequence. However, the PLL will encounter a finite frequency step at the start of preamble acquisition due to variations in disk rotational velocity which may be as large as ±1% (more pronounced in exchangable media systems). The lock-in range of the PLL at the time of preamble acquisition must then be at least ±0.01 x $f_{preamble}$. Given that the PLL lock sequence involves only an adjustment to a frequency step, the following requirements will be set for final PLL dynamics within the filter design procedure:

1. Residual phase error θ_e at the end of the preamble (a full 11 NRZ bytes allowed for PLL stabilization) will be 2 ns or less (4% of the total synchronization window).
2. The lock-in range $\Delta\omega_L$ must be at least 1.5 times the expected frequency step range.
3. The minimum 3 dB bandwidth $\omega_{-3\ dB}$ in the data field must be twice the expected maximum mechanical vibration frequency (10 kHz).
4. The natural frequency of the loop ω_n and damping ratio ζ will be minimized in the data field in order to achieve a high level of jitter rejection. (Minimum damping ratio ζ will be 0.5 (phase margin of 52°) for adequate stability).
5. Re-lock time to the REFERENCE CLOCK will be minimized.

First, some definitions will be established. Regarding requirement #1, the equations for phase error due to a frequency step are[1]:

$$\theta_e(t) = [\Delta\omega/\omega_n]\,[1/(1-\zeta^2)^{0.5}\sin(1-\zeta^2)^{0.5}\omega_n t]\exp(-\zeta\omega_n t) \quad \text{for } \zeta < 1;$$

$$\theta_e(t) = [\Delta\omega/\omega_n]\,[\omega_n t]\exp(-\omega_n t) \text{ for } \zeta = 1;$$

$$\theta_e(t) = [\Delta\omega/\omega_n]\,[1/(\zeta^2-1)^{0.5}\sinh(\zeta^2-1)^{0.5}\omega_n t] \times \exp(-\zeta\omega_n t) \text{ for } \zeta > 1.$$

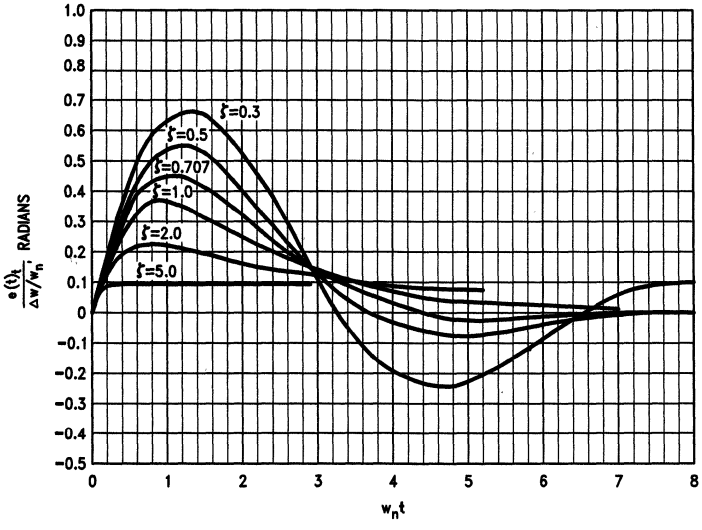
These equations are plotted in *Figure 18*. The equations for phase error due to a phase step are[1]:

$$\theta_e(t) = \Delta\theta\{\cos(1-\zeta^2)^{0.5}\omega_n t - [\zeta/(1-\zeta^2)^{0.5}]\sin(1-\zeta^2)^{0.5}\omega_n t\}\exp(-\zeta\omega_n t) \text{ for } \zeta < 1;$$

$$\theta_e(t) = \Delta\theta[1-\omega_n t]\exp(-\omega_n t) \text{ for } \zeta = 1;$$

$$\theta_e(t) = \Delta\theta\{\cosh(\zeta^2-1)^{0.5}\omega_n t - [\zeta/(\zeta^2-1)^{0.5}]\sinh(\zeta^2-1)^{0.5}\omega_n t\}\exp(-\zeta\omega_n t) \text{ for } \zeta > 1.$$

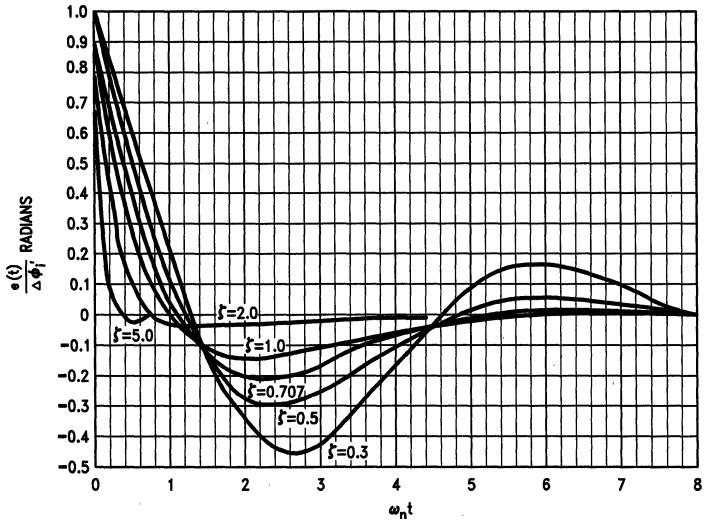
(These equations are plotted in *Figure 19* and are supplied for informational purposes only; an ideal zero phase start function would not produce a phase step at lock initiation.)



TL/F/9322-20

FIGURE 18. Transient Phase-Error Versus the Dimensionless Parameter $\omega_n t$ Due to a Step in Frequency for Various Loop Damping Factors, ζ (from Ref. 4 by Permission of L. A. Hoffman)

3.0 PLL Applications: Loop Filter Design (Continued)



TL/F/9322–21

FIGURE 19. Phase-Error Versus the Dimensionless Parameter $\omega_n t$ Due to a Step in Phase for Various Loop Damping Factors, ζ (from Ref. 4 by Permission of L. A. Hoffman)

Note that the phase error θ_e is measured with respect to the divided (or gated) VCO phase, i.e., 2π radians = N/(20 MHz) = 200 ns in this example.

Regarding requirement #2, the lock-in range (with no cycle-slipping) can be shown to be equal to the open loop transfer function multiplied by the loop filter impedance evaluated at infinite frequency[2]:

$$\Delta\omega_L \approx \pm K_B Z_f(s)|_{s \to \infty}$$

The 3 dB bandwidth for requirement #3 is defined by the equation[3]:

$$\omega_{-3\,dB} = \omega_n [2\zeta^2 + 1 + \{(2\zeta^2 + 1)^2 + 1\}^{0.5}]^{0.5}$$

Requirement #4 has been established in order to maximize the available window margin via PLL dynamics. Conceptually, window margin is preserved if the loop phase response to individually displaced bits (jitter) is not allowed to cause subsequent windows to be readily shifted from the "average" position. Any window movement from nominal position can readily degrade the window margin. It can be seen from *Figure 19* that systems employing low values of damping ratio exhibit a reduced instantaneous response to phase step and thus display improved jitter rejection with respect to higher damping ratio systems. Damping ratio, fortunately, is easily regulated by loop filter design. It also follows that a low natural frequency and its associated "slower" instantaneous phase response will assist in achieving the goal of jitter rejection. However, the minimum natural frequency limit for the PLL may actually be imposed on the system by the $\theta_e(t)$ settling time requirement, the $\Delta\omega_L$ requirement, or the $\omega_{-3\,dB}$ requirement. Whichever of these produces the highest minimum ω_n value must, by necessity, dominate in the design. The goal of minimizing the natural frequency in order to maximize jitter rejection, therefore, may have to defer to one of these other three criteria.

Requirement #5 is addressed in three ways: 1) the DP8459 itself engages the frequency discriminating action of the Phase Comparator whenever the READ GATE is deasserted and the PLL locks to the REFERENCE CLOCK signal, thus guaranteeing re-lock regardless of the initial frequency step; 2) tying the HIGH GAIN DISABLE pin to the READ GATE input places the Charge Pump in the high gain mode whenever the PLL is locked to the REFERENCE CLOCK, producing an elevated natural frequency and a more rapid locking action; 3) N = 2 whenever the READ GATE is deasserted, which, in this example, effectively increases the loop gain by another factor of 2 with respect to the gain within the preamble, where N = 4.

Determining PLL Response Characteristics

It is expected that the minimum value of ω_n will be determined by the residual phase error requirement of #1 rather

3.0 PLL Applications: Loop Filter Design (Continued)

than the lock-in range requirement of #2 or the $\omega_{-3\ dB}$ requirement of #3. This assumption will be checked at the end of the analysis. System requirements then are as follows:

1. $\theta_e(t) \leq (2\ ns) \times (2\pi\ rad/\ 200\ ns) = 0.063$ radians, where t = preamble length 8.8 μs
2. $\Delta\omega_L \approx \pm K_B Z_f(s)|_{s \to \infty} \geq 0.015 \times 5\ MHz \times 2\pi = 471$ Krad/sec
3. $\omega_{-3\ dB} = \omega_n[2\zeta^2 + 1 + \{(2\zeta^2 + 1)^2 + 1\}^{0.5}]^{0.5}$
   $\geq 2 \times 10\ kHz \times 2\pi = 126$ Kr/s

Requirement #1 calls for $\theta_e(8.8\ \mu s) \leq 0.063$ radians. Damping ratio ζ varies as the inverse square root of N (see the equation for Damping Ratio in Section 3.0) such that $\zeta_{PREAMBLE} = \sqrt{(N_{MAX}/N_{PREAMBLE})} \times \zeta_{MIN} = \sqrt{2} \times 0.5 = 0.707$. Solving the appropriate equation for $\theta_e(t)$ for various values of ω_n with $\zeta = 0.707$, t = 8.8 μs and an expected frequency step of $0.01 \times 5\ MHz \times 2\pi = 314$ Kr/s:

ω_n	θ_e (8.8 μs)	$\lvert t_e \rvert$
200 Kr/s	0.606 rad	19.29 ns
300 Kr/s	0.219 rad	6.97 ns
400 Kr/s	0.056 rad	1.78 ns
500 Kr/s	0.0012 rad	0.038 ns
600 Kr/s	−0.0098 rad	0.312 ns
700 Kr/s	−0.008 rad	0.026 ns

$\theta_e\ (8.8\mu s)|_{400\ Kr/s} = 0.056$ radian < 0.063 radian

$t_e = 0.056$ radian $\times$ 200 ns/2π radian $= 1.78$ ns < 2 ns

Thus 400 Kr/s is chosen as the desired natural frequency within the preamble to satisfy requirement #1.

If the assumption that $\theta_e(t)$ dominates the minimum natural frequency requirement is correct, then the $\Delta\omega_L$ requirement of #2 and the $\omega_{-3\ dB}$ requirement of #3 should be met by the ω_n obtained above. First, examining requirement #2,

$$Z_f(s)|_{s \to \infty} = R_1 \text{ } (C_2 \text{ neglected}).$$

Thus,

$$\Delta\omega_L = K_B R_1$$

Rearranging for R_1:

$$R_1 = \Delta\omega_L / K_B$$

The equation for R_1 previously derived shows

$$R_1 = 2\zeta\ \omega_n / K_B$$

Thus,

$$\Delta\omega_L / K_B = 2\zeta\omega_n / K_B$$

$$\Delta\omega_L = 2\ \zeta\omega n$$

In this case, $\omega_n = 400$ Kr/s and $\zeta = 0.707$ (preamble), thus

$\Delta\omega_L = 400\ Kr/s \times 2 \times 0.707 = 566\ Kr/s > 471\ Kr/s$

Thus, requirement #2 is met.

Examining requirement #3, where $\omega_{-3\ dB} \geq 2 \times 10\ kHz \times 2\pi$ when N equals its maximum value of 8 (minimum frequency data pattern; $\zeta = 0.5$):

$\omega_n(min) = \omega_n(preamble) \times 1/\sqrt{(N_{MAX}/N_{PREAMBLE})}$

$= 400\ Kr/s \times 1/\sqrt{2} = 283\ Kr/s$

$\omega_{-3\ dB} = \omega_{n(min)}\ [2\zeta^2 + 1 + \{(2\zeta^2 + 1)^2 + 1\}^{0.5}]^{0.5}$

$= 283\ Kr/s \times 1.817 = 514\ Kr/s$

$514\ Kr/s \div 2\pi = 82\ kHz > 2 \times 10\ kHz$

Thus requirements #1 through #3 are met, and #4 defers to the minimum ω_n established by #1.

Regarding requirement #5, the DP8459 has been configured externally in this example such that when the READ GATE is deasserted, the loop gain will be increased by a factor of 2 due to the Charge Pump gain switching ($R_{NOM} = R_{BOOST}$; HGD tied to RG) and by an additional factor of 2 due to the decrease in N from 4 (preamble) to a fixed internal value of 2. The resulting factor of 4 effective gain elevation results in an increase in both the natural frequency, ω_n, and the damping ratio, ζ, by $\sqrt{4} = 2$. Thus, when READ GATE is deasserted,

$$\omega_n = 2 \times 400\ Kr/s = 800\ Krad/s$$

$$\zeta = 2 \times 0.707 = 1.414$$

$$\Delta\omega_L = 2\zeta\omega_n = 2 \times 1.414 \times 800\ Krad/s = 2.3\ Mr/s$$

COMPONENT CALCULATIONS

The formulae for the filter components, derived previously, are

$$C_1 = \frac{K_B}{\omega_n^2} = \frac{1.2\ V_{CC}\ f_o/(2R_pN)}{\omega_n^2}$$

$$R_1 = \frac{2\zeta\omega_n}{K_B} = \frac{2\zeta\omega_n}{1.2\ V_{CC}\ f_o/(2R_pN)}$$

$$C_2 \leq (1/10)\ C_1$$

A 2:1 ratio of high-to-low Charge Pump gain was chosen for the derivation of R_{NOM} and R_{BOOST}. To achieve the 2:1 gain ratio, R_{NOM} must be equal to R_{BOOST} while the parallel combination $R_{NOM} \| R_{BOOST}$ must be equal to or greater than 1.2 kΩ as per specification. Note that in the equation for C_1 above, the capacitor value is inversely proportional to R_p. Thus, external field interference immunity can be achieved if C_1 is maximized through the minimizing of R_p. The selection of $R_{NOM} = R_{BOOST} = 2.4$ kΩ satisfies the requirements for the Charge Pump resistors and the gain ratio. R_p will be equal to R_{NOM} with READ GATE high, and thus

$C_1 = [1.2 \times 5 \times 20\ MHz/(2 \times 2.4k \times 4)]/(400\ Kr/s)^2$

$= 0.039\ \mu F$

R_1 can now be calculated:

$$\frac{2 \times 0.707 \times 400\ Kr/s}{1.2 \times 5 \times 20\ MHz/(2 \times 2.4k \times 4)} = 90\Omega$$

A standard value of 100Ω is chosen. Since $C_2 \leq 0.1 \times C_1$, C_2 will be chosen to be 510 pF. A table listing the dynamics of the PLL under standard operation conditions and with component values adjusted to industry standards is shown in *Figure 20*.

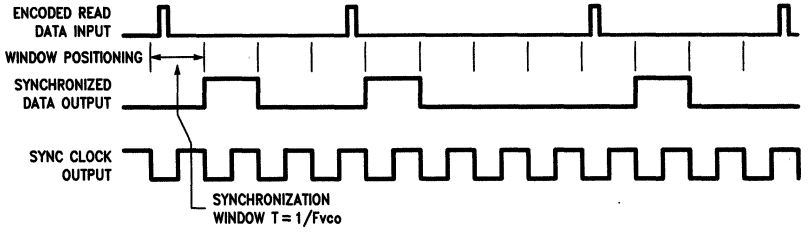
3.0 PLL Applications: Loop Filter Design (Continued)

Field	Preamble	Min Freq Data	Max Freq Data	Ref Clock
N	4	8	3	2
CP Gain	Low	Low	Low	High
Natural Freq. ω_n	400 Krad/s	283 Krad/s	462 Krad/s	800 Krad/s
Damping Ratio ζ	0.7	0.5	0.8	1.4

FIGURE 20. 2,7 Code, 10 Mbits/Sec Design Example PLL Dynamics

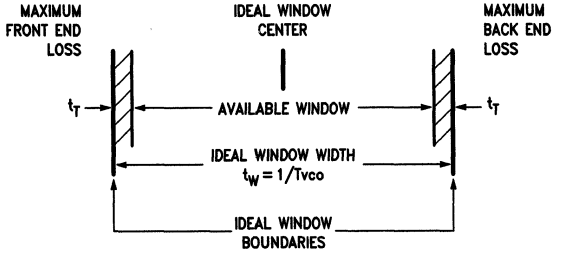
4.0 Window Margin and Bit Jitter Tolerance

A key performance specification for the DP8459 involves the integrity of the synchronization window. The **synchronization window** is defined as a continuously repeating time cell, nominally equal in span to the period of the VCO, within which an ENCODED READ DATA pulse will be recognized (captured) regardless of its position within the window (see *Figure 21*). The captured ERD bit is then transmitted to the SYNCHRONIZED DATA output on the next occurring SYNC CLOCK negative edge. The SYNCHRONIZED DATA and the SYNC CLOCK are held in a fixed, specified timing relationship for use by the data controller in deserialization and decoding. The synchronization window (with strobe setting at nominal position) is centered about the mean location of the ERD pulses via the delay line and the time-averaging action of the PLL. National Semiconductor specifies the **static window truncation** (t_T) of the DP8459 data synchronizer as the maximum expected loss of the synchronization window seen adjacent to the ideal window boundary following complete PLL stabilization with the strobe control setting at the M = −2 position (see *Figure 22*). Static lock conditions are defined as having been achieved when the PLL has been allowed to establish fully stabilized lock to a consistent preamble-type pattern of nominally positioned, non-shifted ERD pulses.



TL/F/9322–22

FIGURE 21. Synchronization Window



TL/F/9322–23

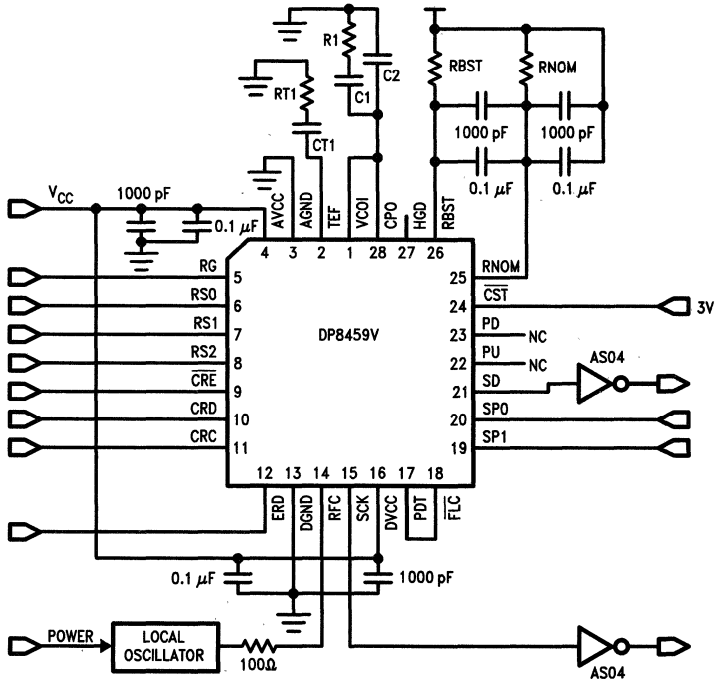
Figure 22. Window Specification Diagram

4.0 Window Margin and Bit Jitter Tolerance (Continued)

4.1 SYNCHRONIZATION WINDOW GENERATION

The DP8459 employs a pulse gate-delay line scheme in the generation of the synchronization window. *Figure 6* shows a simplified block diagram of the pulse gate and delay line circuitry coupled with the phase locked loop. All elements except the delay line are assumed to be delayless for simplicity of analysis. The pulse gate allows a single VCO edge to be transmitted to the pump down input of the phase comparator for each arriving ENCODED READ DATA pulse, while the delay line allows the ENCODED READ DATA pulse to open (enable) the pulse gate at a predetermined time (t_d) prior to the arrival of the ERD pulse at the pump up input of the phase comparator. *Figures 7, 8* and *9* show waveform diagrams of the capture of nominal, early and late ERD pulses, respectively. In normal operation where stable lock has been achieved, the time-integrating action of the PLL has established time alignment between the waveforms at the phase comparator inputs, i.e., both events occur at t_o, on average. If t_d is set equal to $0.5 \times \tau_{VCO}$, the nominal or average ERD pulse will open the pulse gate at $t_o - 0.5 \times \tau_{VCO}$, precisely the midpoint between VCO edges. ERD pulses are then free to shift to any position (ideally) between VCO edges, that is, they have an allowed displacement of $\pm 0.5\ \tau_{VCO}$ from the mean, while yet opening the pulse gate for the passing of the appropriate VCO edge to the phase comparator and at the same time being properly captured by the data synchronization latch (flip-flop D, *Figure 6*). The $\pm 0.5\ \tau_{VCO}$ region is referred to as the synchronization (capture) window.

Any variation in the value of the time delay t_d causes the time at which the pulse gate is enabled ($t_o - t_d$) to shift away from the VCO waveform midpoint, and thus produces a corresponding shift in the position of the synchronization (capture) window. This action, when done in a controlled fashion, is known as window strobing and is useful for purposes of window skew compensation, determination of system window margin, and recovery routines for non-readable data (see Section 4.3).



TL/F/9322-24

Notes: SD and SCK outputs are buffered by Advanced Schottky gates to provide standardized, typical loading conditions.

CRC, CRD, CRE, RG, and ERD are driven by a pattern generator providing the appropriate sequences both to load the control register with the appropriate strobe position information and to cycle the RG and ERD test routine as per *Figure 24*.

FIGURE 23. DP8459 Window Measurement Configuration

4.0 Window Margin and Bit Jitter Tolerance (Continued)

4.2 WINDOW TRUNCATION TESTING

The DP8459 static window truncation specification is an aggregate figure within which the window margin loss contributions from all relevant blocks in the data synchronization chain are combined into the single parameter, t_T.

The preliminary DP8459 static window specification, t_T, applies only to the factory-tested data rates of 10 Mb/s (with RS0,1,2 = 010) and 20 Mb/s (with RS0,1,2 = 000), with the component values as listed for each corresponding data rate in *Figures 5,* and *10,* test configuration as shown in *Figure 23,* test procedure as shown in *Figure 24,* and strobe word M = −2 for 10 Mbits/sec and M = −3 for 20 Mbits/sec. Significant variation in t_T due to the use of other filters and data rates is not expected.

The test algorithm employed in the outgoing factory measurement (screening) of t_T emulates an ENCODED READ DATA stream consisting of a long synchronization field with a single, movable test bit at its end. This method is referred to as *static* window testing, since the window in which the test bit is inserted is fully stabilized and unable to react instantaneously to the phase step introduced by the displaced bit. The standard screening procedure employed for determining DP8459 static window truncation is divided into two portions, one which determines the location of the leading (front) window boundary and one which determines the trailing (back) window boundary. The DP8459 is made to cycle through the read operation many times as a variable bit is moved, once per read cycle, from outside the target window across the ideal leading boundary and into the window. The bit is advanced toward the center of the target window until it resides in a position where it is able to be detected a large number of times consecutively, guaranteeing VCO jitter immunity. The time displacement between the bit's valid detection position and the ideal leading window boundary is recorded as t_{Tf} (front). (This value may be negative if the actual window boundary resides outside the ideal window.) The variable bit is then placed outside the trailing window boundary and the variable bit is again moved, once per read cycle, from outside the target window across the ideal boundary and into the window. The bit continues to advanced toward the center of the recognition region until it is in a position where it is able to be read a large number of times consecutively. The time displacement between the bit's valid detection position and the ideal trailing window boundary is recorded as t_{Tb} (back). (Again, the value may be negative if the actual window boundary resides outside the ideal window due to window encroachment.) The larger (more positive) of the two (t_{Tf}, t_{Tb}) values is taken as t_T. A flow chart of the test sequence is shown in *Figure 24.* Tables of external component values used for production screening of the DP8459 at various data rates are shown in *Figures 5* and *10.*

Window truncation evaluated within data patterns containing shifted bits is a direct function of PLL dynamics which are under Customer control, and thus is neither tested nor specified.

4.3 WINDOW STROBE

The DP8459 incorporates a window strobe function capable of shifting the synchronization window either early or late with respect to its nominal position in small, specified steps. The strobe step t_S is defined as the controlled time displacement of the DP8459 synchronization window from its nominal (strobe centered) position and is typically

$$t_S = M \times [1.8\% \times \tau_{VCO}]$$

where M is the value of the strobe control word (−15 through +15; see *Figure 4*) set by the first 5 bits within the Control Register. (Note that M is equivalent to the hexidecimal value of the five strobe control bits where bits 0 through 3 are the LSB through MSB and bit 4 is the sign bit.)

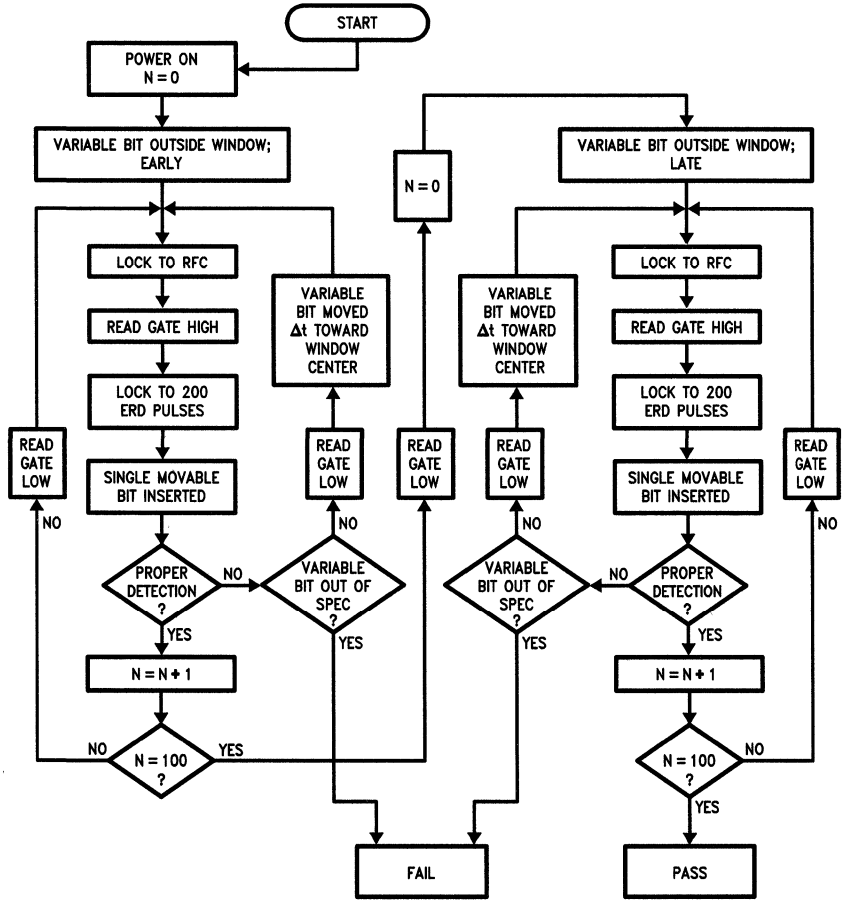
The changing of the strobe value t_S is not an instantaneous event following the changing of the control word in the Control Register. The response time of the strobe control circuitry to any change in strobe setting is a function of the timing elements connected to the TIMING EXTRACTOR FILTER pin and the data rate at which the device is being operated. A finite settling time must be allowed for the delay circuitry to respond following the loading and latching of the new control word (latching occurs and strobe changes begin at de-assertion of CONTROL REGISTER ENABLE, i.e., at transition to logical ONE). It is recommended that any changes to the strobe setting be done with READ GATE deasserted and with a sufficient allowance for settling time prior to the initiation of a subsequent read operation. Approximate settling times are given in *Figure 10* for various TEF component values at specific data rates. (Please refer to AN-578 Window Strobe Function.)

4.3.1 MARGIN TESTING

The read channel window margin of a disk/tape memory system is the portion of the synchronization window remaining after the subtraction of all possible sources of degradation such as media bit shift, head-amplifier anomalies, pulse detector anomalies, cable-induced skew, synchronizer losses, and extraneous noise. The remaining margin must be sufficient to allow the system to perform with an acceptable media error rate under all operating conditions. Acceptable media error rates will vary between systems depending on ECC codes, data redundancy, and other factors. The measured value of the synchronization window margin is often used as a performance criteria for HDA (head-disk assembly) and read channel qualification, and for gauging the probability of encountering data errors on the media.

The DP8459 strobe function can be readily used to measure the window margin within a drive system. Margin tests have been most frequently employed only during outgoing factory tests of storage media systems with specialized and costly test apparatus employed for the purpose; however, the DP8459 allows media/system qualification at any time in the factory or the field during the system's operational life, given the incorporation of an appropriate margin test algorithm within the disk system controller. The algorithm may be configured first to record the most bit-interactive (shift-producing) pattern possible with the recording code being employed (eg., a repeating hex **6D B6** pattern in MFM) in an area of the media where recording density is its highest (inner-most track in constant-angular velocity or constant data rate disk systems), and secondly to read the track repeatedly while incrementally advancing the degree of window "strobe" (controlled shift) first in the early direction until the data error rate crosses a pre-determined threshold and then in the late direction until the same threshold is again crossed. The smaller of the two DP8459 window strobe measurements (either the early or the late value) determined at the error rate threshold crossing points is then equal to the read channel window margin.

4.0 Window Margin and Bit Jitter Tolerance (Continued)



TL/F/9322–25

FIGURE 24. DP8459 Static Window Truncation Test Flow Chart

2

4.0 Window Margin and Bit Jitter Tolerance (Continued)

4.3.2 ERROR-BOUND SECTOR/TRACK DATA RECOVERY

A standard technique exists for attempting to recover illegible data from a sector or track within a disk system which involves the re-reading of the bad data while shifting the data synchronizer window a small amount early/late with respect to the nominal position. A typical early/late strobe value for data retrieval is in the range from approximately 2% to 3% of the total window width. The strobe step size produced by the DP8459 window control circuitry easily allows for this type of data recovery procedure, and is in fact small enough to feasibly permit more than one degree of window movement within the data recovery algorithm.

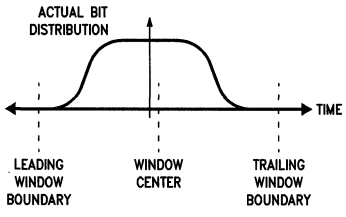
4.3.3 AUTO WINDOW ALIGNMENT (DE-SKEW ROUTINE)

It is possible to configure an intelligent drive system to employ the DP8459 strobe feature in a window auto-calibration (de-skew) routine implemented to center the detection window about the mean position of the bit distribution curve. The de-skew routine would maximize the read channel window margin and correspondingly minimize the bit error rate (BER). The auto-calibration routine would be configured as an extension of the window margin routine (Section 4.3.1), where the early and late strobe values determined at the error rate threshold crossing points would be numerically combined to determine the window center skew. For example, if at 10 Mb/s the strobe-until-error value in the "early" direction were found to be M = −8 and the "late" value M = 4, window skew would be determined as follows:

$$t_{skew} = 1.8 \times \tau_{VCO} \times [M_{early} + M_{late}]/2$$
$$= 0.9\ ns \times [-8 + 4]/2$$
$$= -1.8\ ns$$

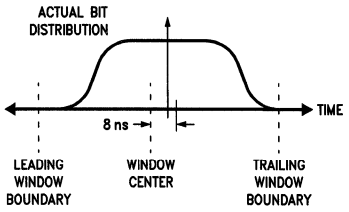
The window has an apparent shift of 1.8 ns in the late direction. The strobe setting in the DP8459 would then be set to compensate for the skew, centering the synchronization window and maximizing the available read channel window margin. In this case, the strobe setting would be M = −2. This routine could be executed at system power-up and perhaps on a regular, specified time schedule during system operation to maintain a fine-tuning of the read channel timing characteristics under varying operating conditions (conceivably eliminating the need for an error-strobe routine).

Original Window Position



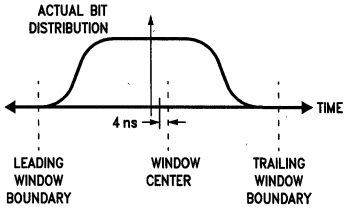
TL/F/9322–2

Early Strobe Window Position



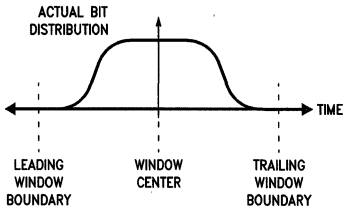
TL/F/9322–3

Late Strobe Window Position



TL/F/9322–4

De-Skewed Window Position



TL/F/9322–5

5.0 Multiple Data Rate Applications

The DP8459 may be rapidly and easily switched from one data rate to another, conceivably from its highest to its lowest specified data rate and vice versa, with a minimum of adaptation effort. This capacity facilitates the employment of the DP8459 for stepped data rate disk applications (constant density recording, or CDR), or for the employment of a single data synchronizer for multiple-media controllers as a cost and space conserving measure, e.g., allowing a controller to address tape, floppy disk and hard disk read channels on a multiplexed basis while employing a single data separator. DP8459 data rate changes require only the appropriate new REFERENCE CLOCK frequency be applied and the necessary new RANGE SELECT information be presented to the chip in cases where the Customer chooses to employ compromise loop and Timing Extractor filters. The Customer may alternatively choose to employ a transmission gate technique to multiplex between appropriate filter elements for various operating data rates should the frequencies be sufficiently different (e.g., streaming tape drive versus hard disk drive).

6.0 PC Board Layout Recommendations

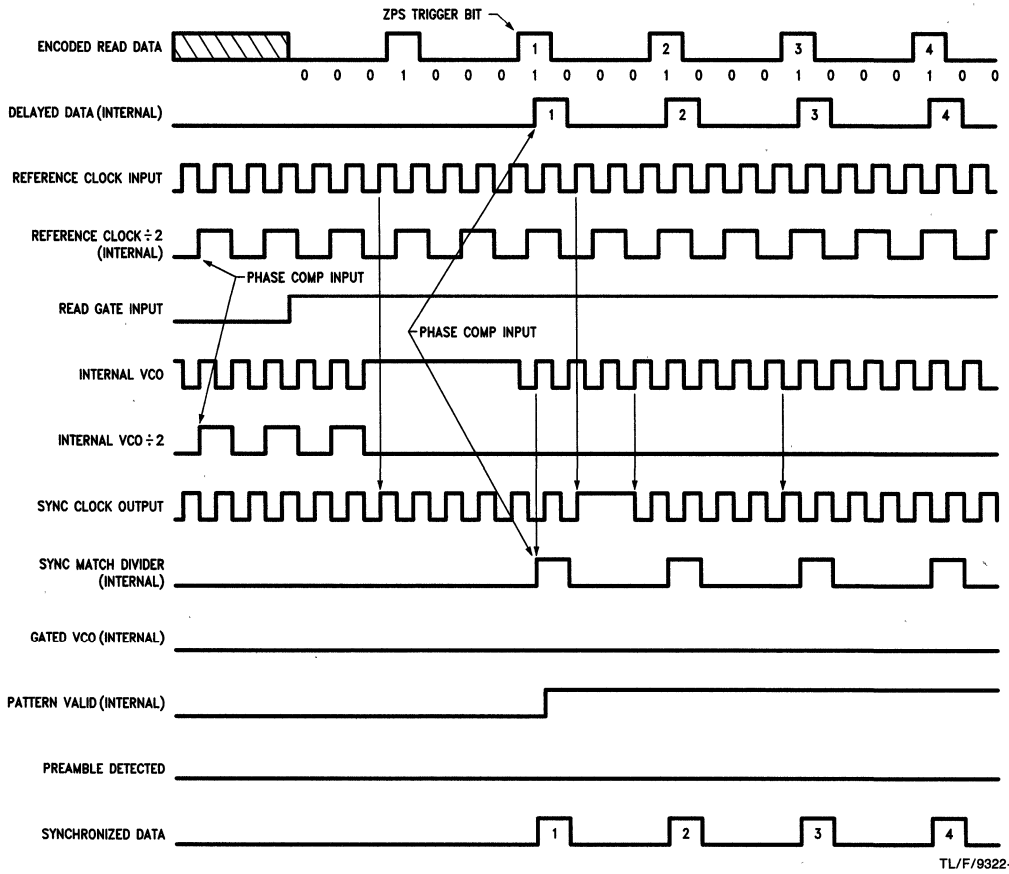
The DP8459 data synchronizer circuit has been designed to minimize the sensitivities normally associated with phase locked loops which operate within digital environments, and in particular those within disk and tape memory systems. A list of recommendations and precautions is made available here for the Customer, however, such that the DP8459 environment can be optimized and the best possible performance achieved with the device.

1. A localized V_{CC} supply net or island should be established for the device and all its associated passive components, supplied by but separated from the main V_{CC} plane. The local V_{CC} net should be tied to the main V_{CC} plane at only one point and bypassed to the ground plane at that point.
2. The DP8459 V_{CC} pins should be bypassed to ground through the shortest electrical path possible between the supply pins the ground pins themselves. Bypassing should be achieved with a 0.1 μF ceramic capacitor in parallel with a 1000 pF silver mica capacitor.
3. The main digital ground plane should be used for all grounding associated with the device. Both Analog and Digital ground pins should be tied to this plane.
4. All passive components associated with the DP8459 should be located as close to their respective device pins as possible. Lead length should be minimized.
5. External passive components should be oriented so as to minimize the length of the ground-return path between the component's ground plane tie point and the DP8459 Analog ground pin.
6. In order to minimize pin parasitic capacitances, planing (supply or ground) should not be placed between device pin eyelets.
7. Digital signal lines should not be run adjacent to external passive analog components associated with the device. Digital signal lines should not be run between analog signal pins or traces associated with the device.
8. Digital input noise experience by the device should be minimized, i.e., it may be advisable to condition input waveforms in order to reduce transient noise. This may be done with a series damping resistor at the REFERENCE CLOCK input (and perhaps at the ENCODED READ DATA input) in high frequency systems. This would terminate board traces and thus prevent underdamped, noise-producing switching transients at the device inputs.
9. Digital output loading should be minimized, i.e., if outputs must drive large loads or long traces, employ buffering. Pre-termination of PC traces driven by the SYNCHRONIZED CLOCK and SYNC DATA outputs may be advisable in high frequency systems (i.e., include series resistance equivalent to the characteristic impedance of the PC board trace).
10. All unused digital output pins should be allowed to float, unconnected to any trace.
11. The device should not be located in a region of the PC board where large V_{CC} or ground plane currents are expected, or where strong electric or magnetic fields may be present. The lowest ambient noise region of the board should be chosen for device location.
12. If device socketing is desired, a low-profile, low mutual capacitance, low resistance, forced-insertion socket type should be employed.
13. Wire-wrapping should not be employed, even in an evaluation set-up.
14. Capacitors used for the loop filter, the Timing Extractor filter, and all bypassing purposes should be ultra-stable monolithic ceramic capacitors or equivalent timing quality capacitors. Silver-mica capacitors should be employed for values 1000 pF and below.
15. In order to achieve very close proximity of passive components to the DP8459 device, it is acceptable to have axial-lead resistors standing upright; however, the shorter component lead should be connected to the device pins to obviate noise induction into sensitive nodes.

7.0 Application Support

It is National Semiconductor's policy to offer and maintain a high level of direct Customer support on all of its mass storage products. National's experience in supporting the disk data memory industry has allowed the DP8459 to be designed to directly address the unique challenges of serial data synchronization within the areas of magnetic and optical media data storage and local area networks, facilitating straightforward use of the device in a diverse range of applications. In the event that questions arise regarding the use of the DP8459 or any other associated NSC mass storage device, the Customer is encouraged to contact the Logic Applications Group or Logic Marketing Group at

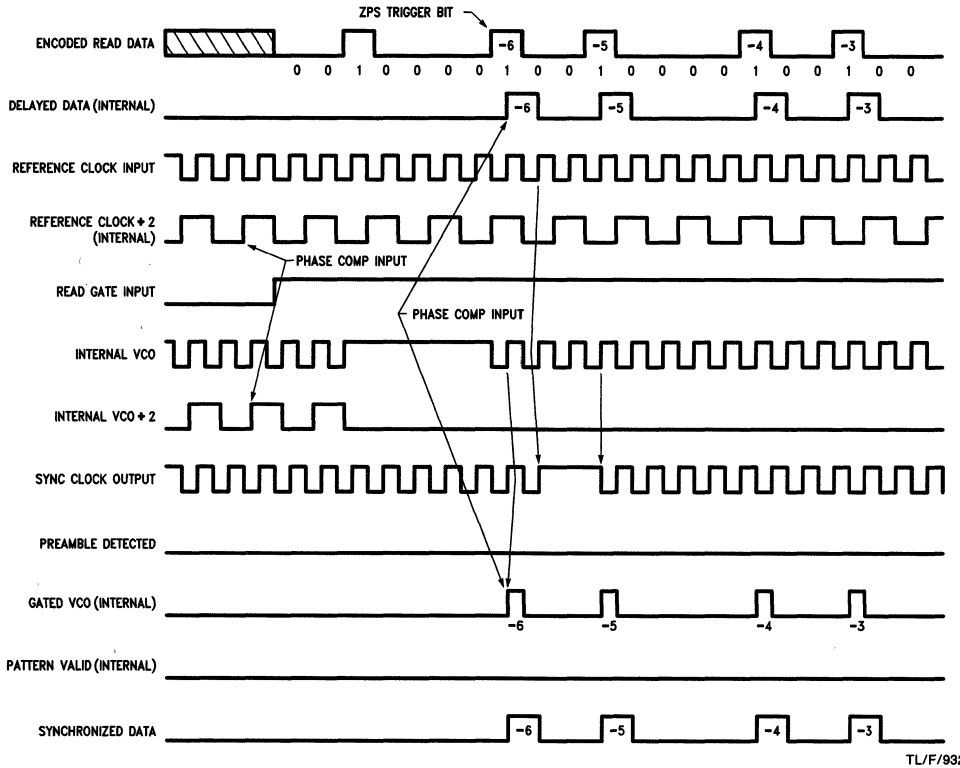
National Semiconductor Corporation
2900 Semiconductor Drive
P.O. Box 58090
Santa Clara, CA 95052-8090
Telephone (408) 721-5000



TL/F/9322–26

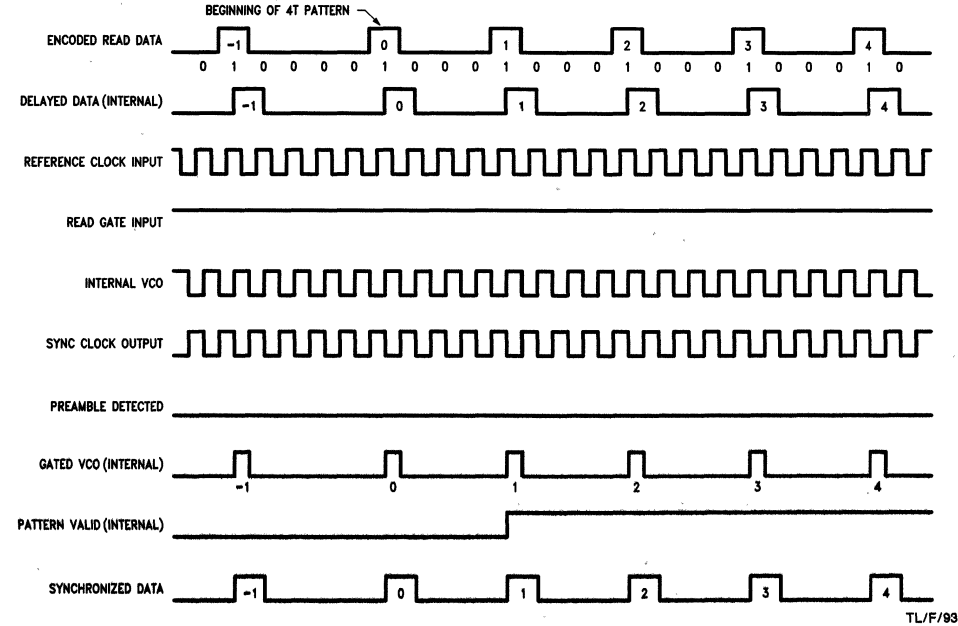
FIGURE 25. Zero Phase Start Lock Acquistion Sequence and Start of Preamble Detection; Frequency Lock Employed, 4T Pattern

7.0 Application Support (Continued)



TL/F/9322-27

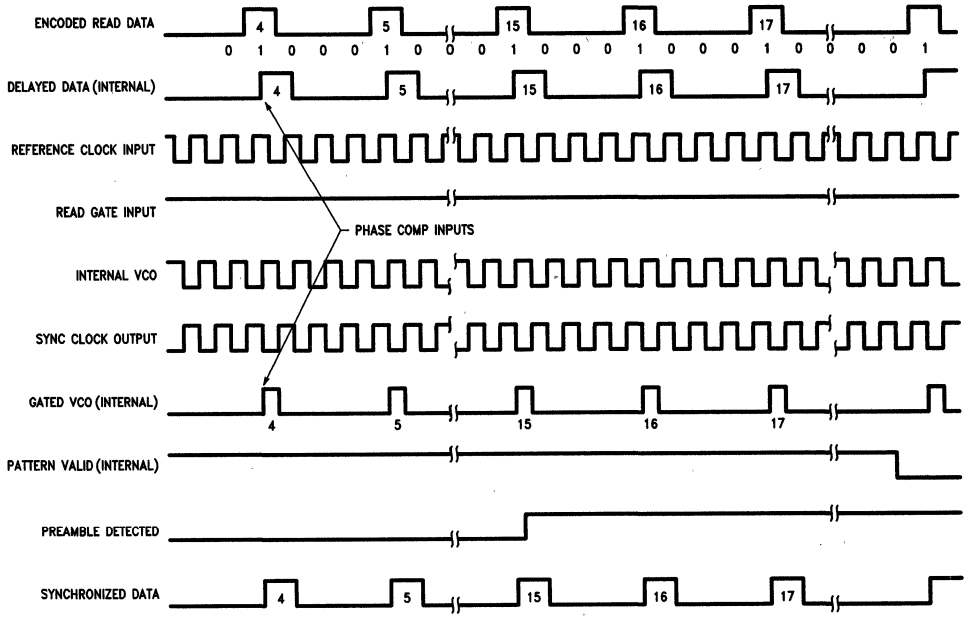
FIGURE 26. Zero Phase Start Lock Acquisition Sequence, Frequency Lock not Employed (Soft Sectoring)



TL/F/9322-28

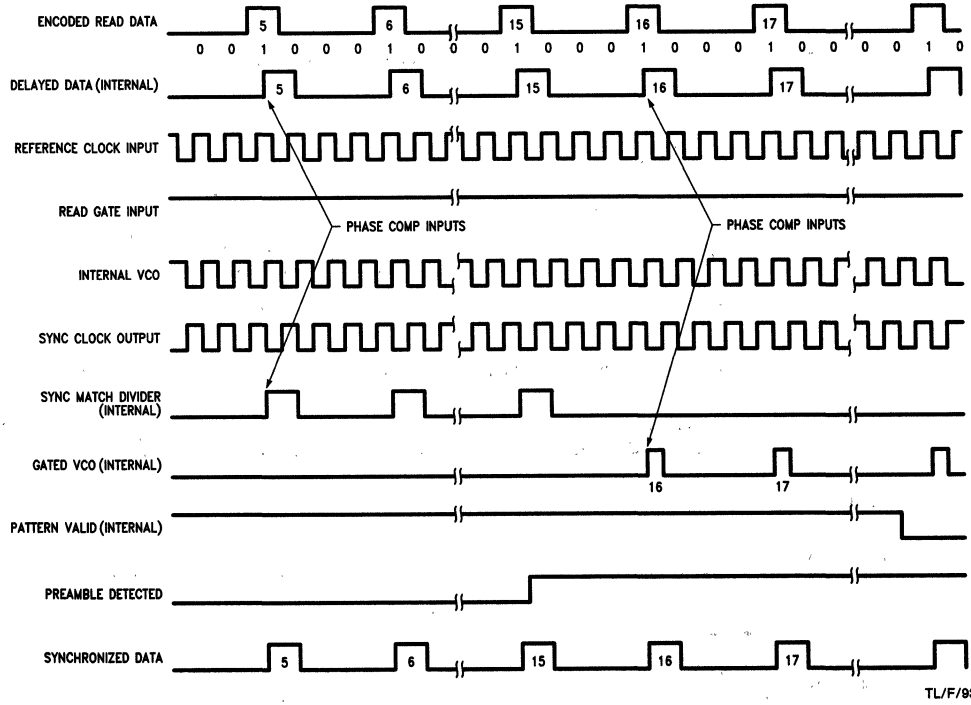
FIGURE 27. Start of Preamble Detection; 4T Pattern, Frequency Lock not Employed (Soft Sectoring)

7.0 Application Support (Continued)



TL/F/9322-29

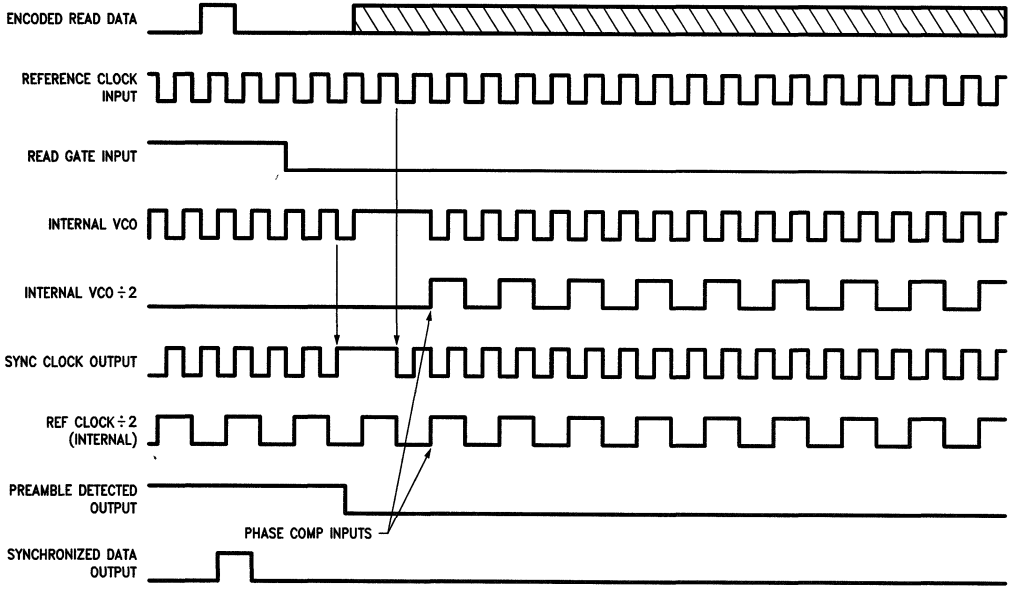
FIGURE 28. Occurrance of Preamble Detection; 4T Pattern, Frequency Lock not Employed (Soft Sectored)



TL/F/9322-30

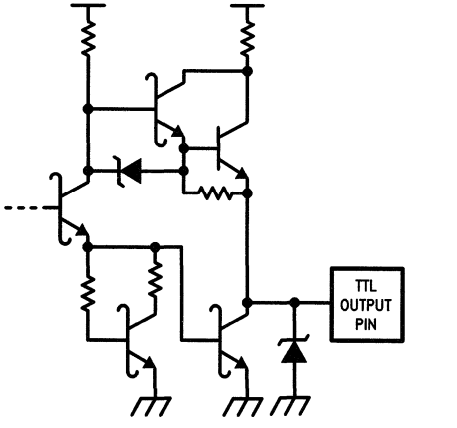
FIGURE 29. Occurrance of Preamble Detection, Frequency Lock Employed

7.0 Application Support (Continued)



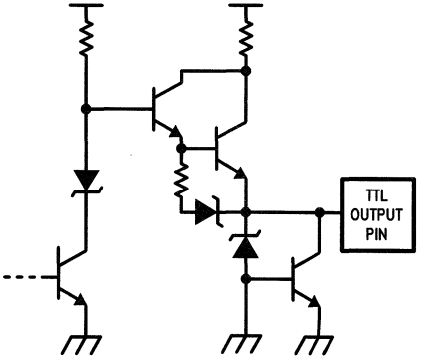
TL/F/9322–31

FIGURE 30. End of Read Cycle; REFERENCE CLOCK Lock Sequence



TL/F/9322–32

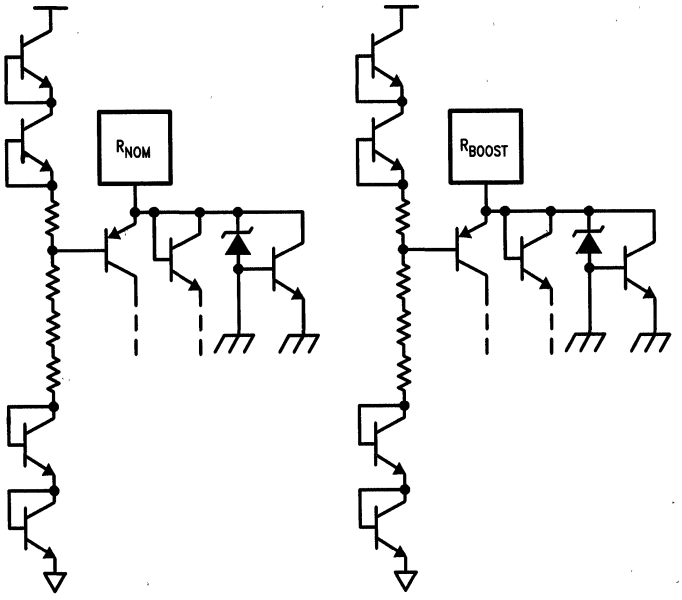
FIGURE 31. Typical TTL Digital Output



TL/F/9322–33

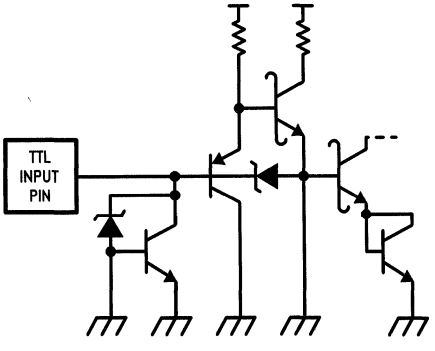
FIGURE 32. Open Emitter TTL Output (PU and PD Outputs)

7.0 Application Support (Continued)



TL/F/9322–34

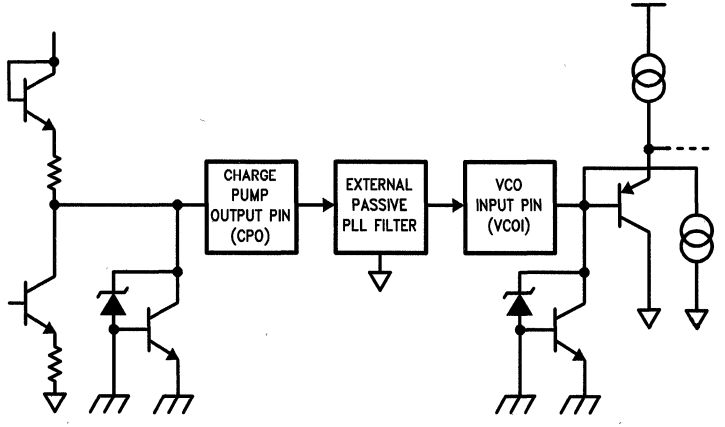
FIGURE 33. $R_{NOMINAL}$ and R_{BOOST} Pin Configurations



TL/F/9322–35

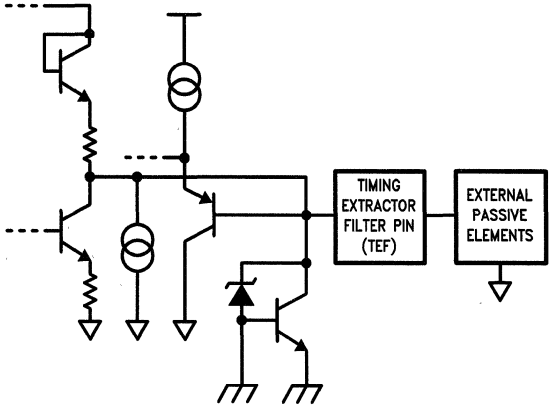
FIGURE 34. Typical TTL Digital Input

7.0 Application Support (Continued)



TL/F/9322–36

FIGURE 35. Charge Pump Output and VCO Input Circuit Configurations



TL/F/9322–37

FIGURE 36. Timing Extractor Filter Pin Circuit Configurations

References

1. *Phaselock Techniques*, Floyd M. Gardner, Second Edition, John Wiley & Sons, 1979, pp. 48.
2. ibid, pp. 70.
3. ibid, pp. 14.
4. *Receiver Design and the Phase Locked Loop*, L.A. Hoffman, Aerospace Corporation, El Segundo, Ca., May 1963.

National Semiconductor

DP8462 2,7 Code Data Synchronizer

General Description

The DP8462 Data Synchronizer is designed for application in disk drive memory systems employing Run Length Limited Codes using 1-0-0 or 1-0-0-0 preamble patterns, and depending on system requirements, may be located either in the drive or in the controller. It receives digital pulses from a pulse detector circuit (such as the DP8464 Disk Pulse Detector) if the DP8462 is situated in the drive, or from an interface if it is situated in the controller. In the read mode, the circuit locks onto and detects either a 100 or 1000 preamble pattern depending on the state of the pattern select input pin. The synchronized data and clock are then available for decoding and deserialization by a decoder circuit. All of the digital input and output signals are TTL compatible and only a single +5V supply is required. Although separate Analog and Digital V_{CC} and Ground pins are provided, they are expected to be tied together by the user. The chip is housed in a standard narrow 24-pin dual-in-line package (DIP) and is fabricated using Advanced Schottky bipolar analog and digital circuitry. This high speed I.C. process allows the chip to work with data rates up to 20 Mbits/sec. There are two versions of the chip, each having a different decode window error specification. These two versions (-3 and -4) will operate from 4 to 20 Mbits/sec, with respectively increasing window errors, as specified in the Electrical Characteristics Table.

The DP8462 features a phase-locked-loop (PLL) consisting of a pulse gate, phase comparator, charge pump, buffering amplifier, and voltage-controlled-oscillator (VCO). Pins are provided for the user to select the values of the external filtering components required for the pulse gate and PLL, the frequency setting components required for the VCO, two current setting resistors for the charge pump, and current setting resistors for the pulse gate that control the delay line.

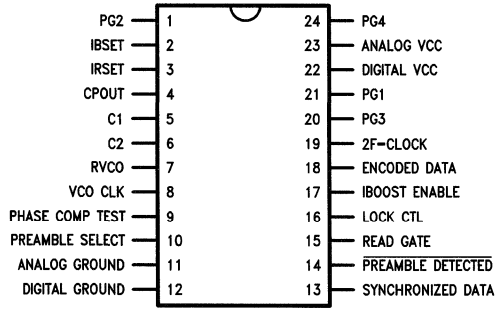
The on-board PLL's phase comparator has two modes of operation: phase and frequency comparison or phase only comparison. In the non-read mode, the comparator performs phase and frequency comparison, but once in the read mode, it switches to phase only comparison. The user selects whether this mode change occurs as soon as read mode is entered or after the preamble pattern is detected. The charge pump also has two modes of operation: high track rate—intended to be used in the non-read mode and in the read mode while acquiring lock, and low track rate—intended to be used in the read mode to retain lock. Both track rates are selected by the user with external components; the user is given control over when the track rate switch takes place.

Features

- Phase-Frequency PLL in non-read mode and during preamble if desired
- Operates at data rates up to 20 Mbit/sec
- Detects either 1-0-0 or 1-0-0-0 preamble patterns
- User determined PLL loop filter network
- PLL charge pump has two user-determined tracking rates
- External control of track-rate switchover
- External control of phase comparator switchover
- Delay line may be externally adjusted if desired
- ORed phase comparator outputs for monitoring bit-shift
- Standard narrow 24-pin DIP or 28-pin Plastic Chip Carrier package
- Less than ½W power consumption
- Single +5V supply

Connection Diagrams

Dual-In-Line Package

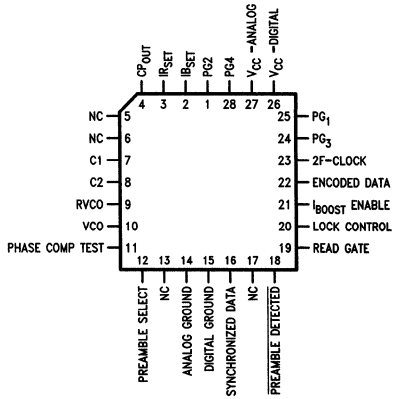


TL/F/8418-2

Top View

Order Number DP8462N
See NS Package Number N24C

Plastic Chip Carrier

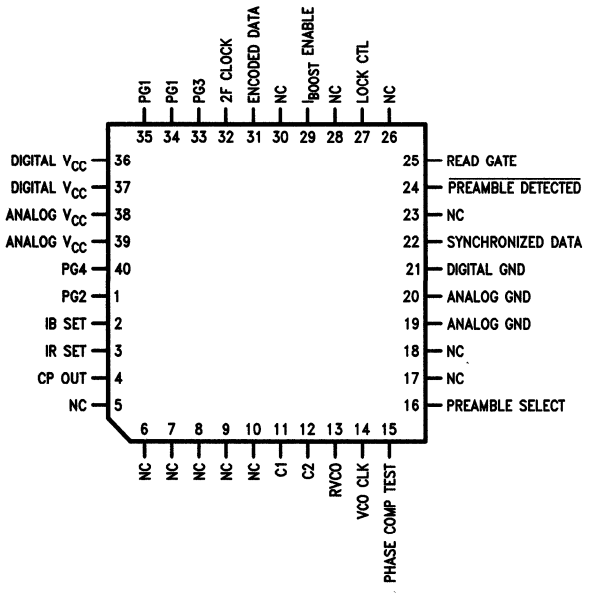


TL/F/8418-24

Order Number DP8462V
See NS Package Number V28A

Connection Diagrams (Continued)

TapePak®



TL/F/8418–25

Top View

Order Number DP8462TP
See NS Package Number TP40A

Pin Descriptions*

POWER SUPPLY

22,23 Digital and Analog V_{CC} = +5V ±5%
Should be tied together and bypassed by user.

11,12 Analog and Digital Ground
Should be tied together by user.

TTL Level Logic Inputs

15 READ GATE: When asserted, this signal sets the DP8462 into the Read Mode. The PLL then begins to lock onto the encoded data.

10 PREAMBLE SELECT: A high level on this input enables the circuit to recognize a 1-0-0-0 pattern while a low level results in the recognition of a 1-0-0 pattern. Also, in the non-read mode, if 1-0-0 is selected VCO/3 will lock onto 2F/3 while if 1-0-0-0 is selected VCO/4 will lock onto 2F/4.

16 LOCK CTL: This input allows the user to determine when the circuit will switch from Phase-Frequency comparison to Phase only comparison once in the Read Mode. A low level on this pin causes the circuit to switch from the phase-frequency comparison mode as soon as READ GATE is asserted while a high level means that the circuit will switch after 4 bytes of preamble have been detected and PREAMBLE DETECTED output has been asserted. (See the Truth Table at the end of this Section.)

17 IBOOST ENABLE: This input allows the user to control the PLL's track rate by turning Iboost current on and off. A high level at this input causes Iboost to be added to Irate—placing the PLL in the high track rate. In a typical system IBOOST ENABLE may be tied to READ GATE or PREAMBLE DETECTED.

18 ENCODED DATA: This input is for the incoming encoded data from the output of the head amplifier/pulse-detecting network located on the disk drive. Each positive edge of the ENCODED DATA waveform identifies a flux reversal on the disk.

19 2F-CLOCK: This is a system clock input, which is either a signal generated from the servo track, or a signal buffered from a crystal. It operates at twice the NRZ DATA rate. 2F-CLOCK MUST ALWAYS BE APPLIED TO THIS INPUT FOR PROPER OPERATION.

TTL Level Logic Outputs

8 VCO CLOCK: This is the output of the on-chip VCO, transmitted from an Advanced Schottky TTL buffer. It is synchronized to the SYNCHRONIZED DATA output so that it can be used by the encoder/decoder circuitry.

13 SYNCHRONIZED DATA: This is the same encoded data that is input to the chip, but is synchronous with the VCO CLOCK.

*Pin Number Designations apply for the 24 Pin DIP. See Connection Diagram for the Plastic Chip Carrier Pin Designations.

Pin Descriptions (Continued)

14 $\overline{\text{PREAMBLE DETECTED}}$: After READ GATE is asserted, this output goes low after detecting approximately 4 bytes of preamble and remains low until READ GATE goes inactive.

9 PHASE COMP TEST: This output is the logical "OR" of the Phase Comparator outputs, and may be used for the testing of the disk media.

Analog Signals

21,20 PG1, PG3: The external capacitors and resistor of the Pulse Gate filter are connected to these pins. PG1 should be tied directly to ground.

1 PG2: This is the Pulse Gate delay reference pin. The delay reference generator establishes a voltage at this pin; thereby producing the bias current for the Pulse Gate delay section in the resistor tied between this pin and V_{CC} for the DP8462-4 and in the current splitting network for the DP8462-3.

24 PG4: This is the Pulse Gate delay control pin. This pin can be tied to the PG2 pin if the user desires to adhere to the DP8462-4 standard synchronization window specification. For the DP8462-3 it should be tied to PG2 and V_{CC} through a "current splitting" network (see *Figure 6*) for optimal window positioning.

3 IRSET: The current into the rate set pin (Vbe/Rrate) is approximately half the charge pump output current for the low tracking rate.

2 IBSET: The current into the boost set pin (Vbe/Rboost) is approximately half the amount by which the charge pump current is increased for the high tracking rate.
(I_{hirate} = I_{rate} Set + I_{boost} Set).

4 CPOUT: This pin is the output node of the charge pump and also the noninverting input of the Buffer Amplifier. It is made available for connection of external filter components for the phase-locked-loop.

5,6 VCO C1, C2: An external capacitor connected across these pins sets the nominal frequency.

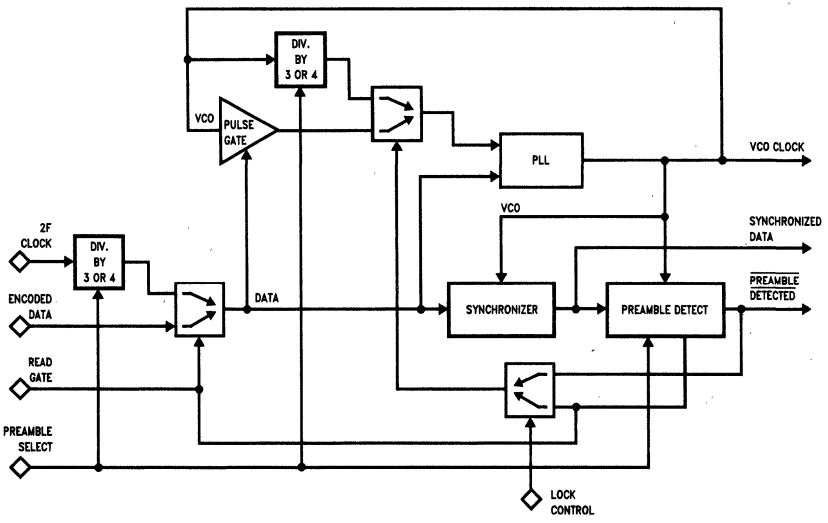
7 RVCO: The current into this pin determines the operating currents within the VCO.

Note: ANALOG and DIGITAL V_{CC} pins must be tied together by the user.
ANALOG and DIGITAL GND pins must also be tied together by the user.

Truth Table of Pulse-Gate's Modes

LOCK CTL (Pin 16)	READ GATE (Pin 15)	$\overline{\text{PREAMBLE DETECTED}}$ (Pin 14)	Pulse-Gate Comparison Mode	Comments
LO	LO	LO	N/A	N/A
LO	LO	HI	Phase and Frequency	Non-Read Mode
LO	HI	HI	Phase only	Read Mode
LO	HI	LO	Phase only	Read Mode
HI	LO	LO	N/A	N/A
HI	LO	HI	Phase and Frequency	Non-Read Mode
HI	HI	HI	Phase and Frequency	Read Mode
HI	HI	LO	Phase Only	Read Mode

Block Diagram



TL/F/8418–1

Absolute Maximum Ratings

If Military/Aerospace specified devices are required, please contact the National Semiconductor Sales Office/Distributors for availability and specifications.

Supply Voltage	7V
TTL Inputs	7V
Output Voltages	7V
Input Current (CPOUT, IRSET, IBSET, RVCO)	2 mA
Storage Temperature	−65°C to +150°C
Operating Temperature Range	0°C to +70°C

ESD Rating is to be determined.

Operating Conditions

Symbol	Parameter		Conditions	Min	Typ	Max	Units
V_{CC}	Supply Voltage			4.75	5.00	5.25	V
T_A	Ambient Temperature			0	25	70	°C
I_{OH}	High Logic Level Output Current		V_{CO} Clock Others			−2000 −400	μA
I_{OL}	Low Logic Level Output Current		V_{CO} Clock Others			20 8	mA
f_{DATA}	Input Data Rate			4.0		20	Mbit/sec
t_{WCK}	Width of 2f-CLOCK, High or Low			10			ns
t_{WPD}	Width of ENCODED DATA Pulse (Note 1)	High		18			ns
		Low		0.4t			ns
V_{IH}	High Logic Level Input Voltage			2			V
V_{IL}	Low Logic Level Input Voltage					0.8	V
t_{SU} Read Gate	Min Time Required for a Positive Edge of Read Gate to Occur Before a Negative Edge of VCO			20			ns
t_{HOLD} Read Gate	Min Time Required for a High Level on Read Gate to be Held After a Negative Edge of VCO			10			ns

Note 1: t is defined as the period of the NRZ data ($t = 2/F_{VCO}$).

DC Electrical Characteristics Over Recommended Operating Temperature Range

Symbol	Parameter	Conditions	Min	Typ	Max	Units
V_{IC}	Input Clamp Voltage	V_{CC} = Min, I_I = −18 mA			−1.5	V
V_{OH}	High Level Output Voltage	V_{CC} = Min, I_{OH} = Max	V_{CC} −2V	V_{CC} −1.6V		V
V_{OL}	Low Level Output Voltage	V_{CC} = Min, I_{OL} = Max			0.5	V
I_{IH}	High Level Input Current	V_{CC} = Max, V_I = 2.7V			20	μA
I_{IL}	Low Level Input Current	V_{CC} = Max, V_I = 0.4V			−200	μA
I_O	Output Drive Current (Note 1)	V_{CC} = Max, V_O = 2.125V	−12		−110	mA
I_{CC}	Supply Current	V_{CC} = Max			85	mA
I_{OUT}	Charge Pump Output Current	$200 \le I_{RATE} + I_{BOOST} \le 2000$	$0.9\ I_{TYP} - 25$	$2.0\ (I_{RATE} + I_{BOOST})$	$1.1\ I_{TYP} + 25$	μA

Note 1: This value has been chosen to produce a current that closely approximates one-half of the true short-circuit output current, I_{OS}.

AC Electrical Characteristics Over Recommended V_{CC} and Operating Temperature Range

(All Parts unless stated otherwise) ($t_R = t_F = 2.0$ ns, $V_{IH} = 3.0V$, $V_{IL} = 0V$) (Note 1)

Symbol	Parameter	Min	Typ	Max	Units
t_{READ}	Positive VCO CLOCK transitions from READ GATE set active until PLL Lock sequence begins (DELAY DISABLE high)		1	1	—
$t_{TRANSMIT}$	Positive VCO CLOCK transitions required to transmit input encoded data to output	1	2	3	—
$t_{READ\ ABORT}$	Number of VCO CLOCK cycles after READ GATE set low to read operation abort (Note 3)			2	T-clock
t_{WINDOW}	Variance of center of decode window from nominal (Note 6) DP8462-3 / DP8462-4			6 / 10	ns
ϕLINEARITY	Phase range for charge pump output linearity (Note 2)	$-\pi$		$+\pi$	Radians
K_1	Phase comparator—Charge Pump gain constant ($N = f_{VCO}/f$ input data) (Note 4)		$\frac{1.78\ V_{BE}}{N2\pi R}$		Amps/rad
$V_{CONTROL}$	Charge pump output voltage swing from nominal		±100		mV
K_{VCO} ($= A \times K_2$)	VCO gain constant (ω_{VCO} = VCO center frequency in rad/s) (Note 5)	$\frac{1.20\ \omega_C}{V_{BE}}$	$\frac{1.40\ \omega_C}{V_{BE}}$	$\frac{1.60\ \omega_C}{V_{BE}}$	rad/sec V
f_{VCO}	VCO center frequency variation over temperature and V_{CC}	−2		+2	%
$f_{MAX\ VCO}$	VCO maximum frequency		60		MHz
t_{PHL}	Propagation delay from VCO negative edge to synchronous DATA negative edge	2		18	ns
t_{PLH}	Propagation delay from VCO negative edge to synchronous DATA positive edge	4		20	ns

Note 1: A sample calculation of frequency variation vs. control voltage: $V_{IN} = \pm 0.1V$;

$$K_{VCO} = \frac{\omega OUT}{V_{IN}} = \frac{0.4\ \omega_C}{0.2V} = \frac{2.0\ \omega_C\ (rad/sec)}{V\ \ (volt)}.$$

Note 2: $-\pi$ to $+\pi$ with respect to 2f VCO CLOCK.

Note 3: T-clock is defined as the time required for one period of the VCO CLOCK to occur.

Note 4: With respect to VCO CLOCK; $I_{PUMP\ OUT} = 1.9\ I_{SET}$

$$I_{SET} = \frac{V_{BE}}{R_{SET}}$$

Note 5: Although specified as the VCO gain constant, this is the gain from the Buffer Amplifier input to the VCO output.

Note 6: This specification is guaranteed only for the conditions under which the parts were tested. However, significant variation from formula is not expected for other data rates and filters. The DP8462-4 specification is for the condition when PG2 and PG4 are tied together. The DP8462-3 specification is for the condition when a 330Ω resistor is tied from PG2 to PG4 and a 1.5 kΩ resistor is tied from PG4 to V_{CC}. External adjustment can be used to optimize the window as described in the pulse gate section. The filter values below were chosen for operation in an automatic test system (static window) environment. Different criteria may apply for choosing filter values in a disk system. See Loop Filter Section for sample calculations of other filter values.

Part Type	Data Rate Tested	Filter				
		C_1	C_2	R_1	R_{RATE}	R_{BOOST}
DP8462-4	5 Mbit/sec	0.03 μF	600 pF	100Ω	820Ω	1.5 kΩ
DP8462-3	10 Mbit/sec	0.022 μF	510 pF	81Ω	800Ω	1.8 kΩ

Note: For further informtion refer to Application Note AN-414

External Component Selection (All Parts) (Note 1)

Symbol	Component	Min	Typ	Max	Units
R_{VCO}	VCO Frequency Setting Resistor (Note 2)	990		1010	Ω
C_{VCO}	VCO Frequency Setting Capacitor (Notes 3,4)	20		120	pF
R_{RATE}	Charge Pump I_{RATE} Set Resistor (Note 6)	0.4		4.0	kΩ
R_{BOOST}	Charge Pump (High Rate) I_{BOOST} Resistor (Note 6)	0.5		∞	kΩ
C_R	I_{RATE} Bypass Capacitor (Note 5)	0.01			μF
C_B	I_{BOOST} Bypass Capacitor (Note 5)	0.01			μF

Note 1: External component values for the Loop Filter and Pulse Gate are given in Table II and Table I respectively.

Note 2: A 1% Component Tolerance is Required.

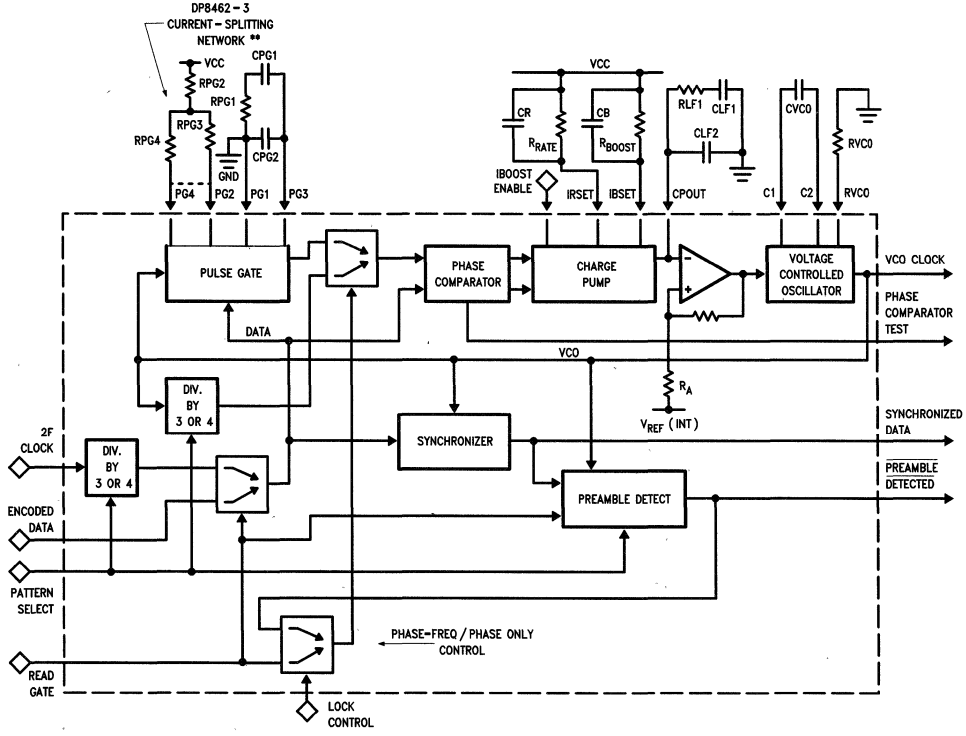
Note 3: These MIN and MAX values correspond to the MAX and MIN data rates respectively.

Note 4: The Component Tolerance is system dependent on how much center frequency deviation can be tolerated.

Note 5: Component Tolerance 15%.

Note 6: The minimum value of the parallel combination of R_{RATE} and R_{BOOST} is 350Ω.

Detailed Block Diagram



TL/F/8418–3

**For DP8462-4 window testing, RPG3 = RPG4 = 0Ω and RPG2 = 4.7 kΩ. For DP8462-3 window testing RPG4 = 0Ω, RPG3 = 330Ω and RPG2 = 1.5 kΩ.

Circuit Operation

In the non-read mode, the DP8462 Data Separator remains locked to the 2f CLOCK signal divided by 3 or 4 (depending upon the preamble used) in anticipation of a preamble when read mode is entered. When the READ GATE input goes high, the DP8462 enters the read mode after 1 VCO CLOCK cycle. Referring to *Figure 1*, once in the read mode, the PLL reference signal is switched from the 2f-divided-by-3-or-4 signal to the ENCODED DATA input. The PLL is at this point in the high-tracking rate mode and also in the Phase and Frequency Comparison mode. The PLL then attempts to

Circuit Operation (Continued)

quickly lock onto the incoming ENCODED DATA stream and starts looking for 16 consecutive preamble pulses—chosen by the user to be either 100 (PATTERN SELECT: LO) or 1000 (PATTERN SELECT: HI). If the user has chosen to switch to Phase Only Comparison as soon as read operation begins (LOCK CTL: LO), then the Phase Comparator will start to compare ENCODED DATA with VCO-gated-by-DATA immediately (see *Figure 2*); otherwise, it will keep comparing ENCODED DATA with VCO divided by 3 or 4—i.e., remain in Phase and Frequency Comparison mode until after 16 consecutive preamble pulses have been detected. At this time, $\overline{\text{PREAMBLE DETECTED}}$ output goes low and the circuit now starts to compare ENCODED DATA with VCO-gated-by-DATA (see *Figure 1*).

The user is given control over when to switch the charge-pump current rate through the use of the IBOOST ENABLE input. One way the user can accomplish this is by tying $\overline{\text{PREAMBLE DETECTED}}$ output to the IBOOST ENABLE input directly. Thus, once $\overline{\text{PREAMBLE DETECTED}}$ is asserted, the circuit will go into low track rate and Phase Only Comparison mode (if LOCK CTL: HI) so that a more stable lock can be retained. The incoming ENCODED DATA stream is now synchronized with the VCO CLOCK and appears at the SYNCHRONIZED DATA output (see *Figure 3*). (If the user wishes to switch to low track rate as soon as the circuit enters the read mode, then the READ GATE signal should be inverted and applied to the IBOOST ENABLE input).

Figure 4 shows the sequence when READ GATE goes low, signifying the end of a read operation. The PLL reference signal is switched back to 2f divided by 3 or 4 input and the $\overline{\text{PREAMBLE DETECTED}}$ output goes high (causing the charge-pump to go to the high tracking rate, if $\overline{\text{PREAMBLE DETECTED}}$ is tied to the IBOOST ENABLE input). Also, the Phase Comparator goes back to Phase and Frequency Comparison mode and the circuit attempts to lock onto the 2f divided by 3 or 4 signal, thus returning to the initial conditions.

CIRCUIT DESCRIPTION

1. Divide by 3 or 4: Depending on the preamble pattern being used, these circuits divide 2f CLOCK and internal VCO CLOCK signals by 3 or 4. During the non-read mode, the VCO remains phase and frequency locked to these divided signals so that when read mode is entered, the PLL can quickly acquire lock because the data stream that consists of the preamble pattern is very close in frequency to the VCO divided by 3 or 4.

2. Pulse Gate: Once in the read mode, the PLL has to lock the VCO CLOCK to the ENCODED DATA stream; outside of the preamble, however, the data signal is not cyclic like the VCO CLOCK and therefore cannot be frequency compared to the VCO. It is for this reason that the Pulse Gate is used to allow a reference signal from the VCO into the Phase Comparator only when an ENCODED DATA bit is valid. The Pulse Gate also utilizes a scheme which delays the incoming data by one-half the period of the 2f-CLOCK. This optimizes the position of the decode window and allows input jitter up to ± half the 2f-CLOCK period, assuming no error in the decode window position. The decode window error can be determined from the specification in the Electrical Characteristics Table.

3. Multiplexers at the Phase Comparator's inputs: These multiplexers are used to determine which signals the Phase Comparator will compare during different modes of operation. Either 2F divided by 3 or 4 or ENCODED DATA is compared with either VCO divided by 3 or 4 (Phase and Frequency Lock) or with VCO gated by DATA (Phase Only Lock).

4. Phase Comparator: The Phase Comparator receives its inputs from the Multiplexers mentioned above, and is edge-triggered from these inputs to provide charge-up and charge-down outputs.

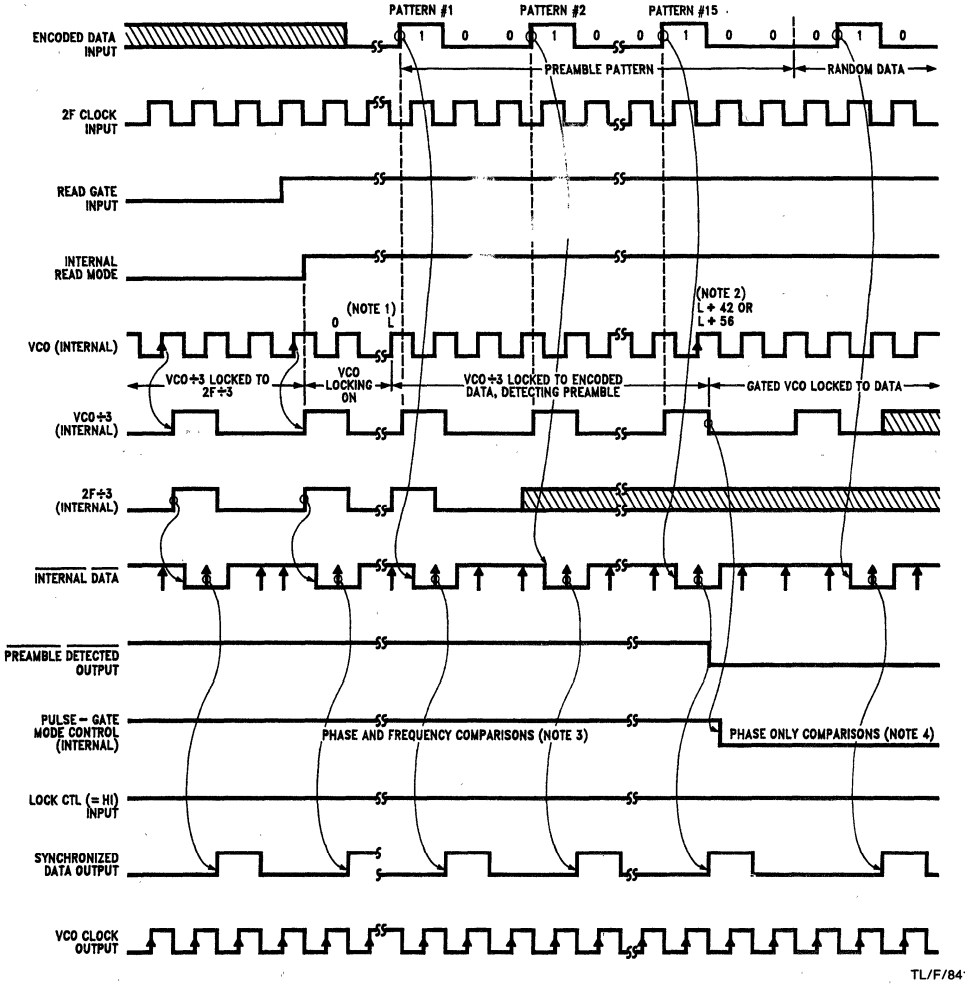
5. Charge Pump: The high speed charge pump consists of a switchable constant current source and sink. The charge pump constant current is set by connecting external resistors to V_{CC} from IRSET and IBSET pins. With IBOOST ENABLE HIGH, the PLL is in the high tracking rate and both resistors determine the current. With IBOOST ENABLE LOW, the PLL is in the low tracking rate and only the IRSET resistor determines the charge pump current. The output of the charge pump feeds into external filter components and the Buffer Amplifier. Thus, through the use of the IBOOST ENABLE pin, the user can determine when the circuit switches track rates.

6. Buffer Amplifier: The Buffer Amplifier is configured as a high input impedance amplifier which is inserted between the charge pump and the VCO, thus allowing connection of external PLL filter components to the charge pump output pin CPOUT. The output of the Buffer Amplifier is internally connected to the VCO control input.

7. VCO: The Voltage-Controlled-Oscillator requires a resistor from the RVCO pin to ground and a capacitor between pins C1 and C2, to set the center frequency. The VCO frequency can be varied from nominal by approximately ±20%, as determined by its control input voltage (CPOUT).

8. Preamble Pattern Detector: Two types of preamble patterns are commonly used in RLL 2,7 code disk systems—1-0-0 and 1-0-0-0. The user selects the preamble pattern to be used by setting PREAMBLE SELECT input either HI for the 1-0-0-0 pattern or LO for the 1-0-0 pattern. The DP8462 divides 2F Clock and VCO Clock signals by 3 or 4 depending upon whether 1-0-0 or 1-0-0-0 pattern is selected, respectively, and remains locked to this divided pattern in anticipation of a preamble. Once the chip is in the read mode, the VCO proceeds to lock onto the incoming data stream. The Preamble Pattern Detector then searches for 16 consecutive patterns (i.e., 100100100... or 100010001000...) to indicate lock has been achieved. The $\overline{\text{PREAMBLE DETECTED}}$ output then goes low. Any deviation from the above-mentioned continuous stream of patterns before 16 of these are detected will reset the Pattern Detector and the procedure will then start over again.

Circuit Operations (Continued)

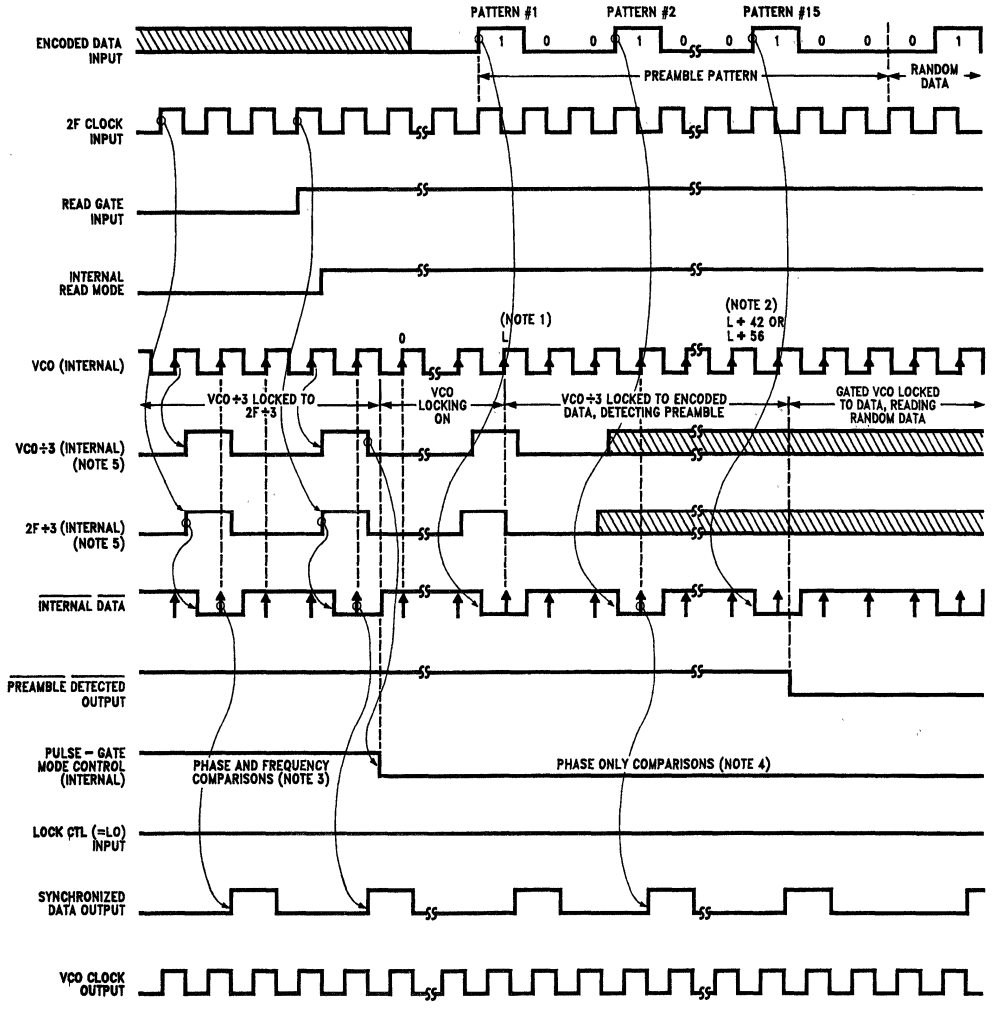


TL/F/8418–4

Note 1: L = Number of VCO cycles required for VCO to lock—typically 20 but determined by external component value.
Note 2: At L + 42 (Pattern = 1-0-0) or L + 56 (Pattern = 1-0-0-0), 15 patterns have been detected.
Note 3: VCO ÷3 (or 4) being compared with 2F ÷3 (or 4) in the non-read mode and Preamble in the Read Mode.
Note 4: VCO GATED BY DATA being compared with ENCODED DATA.
Note 5: PREAMBLE SELECT = LO; 100 pattern selected—so 2F & VCO are being divided by 3.

FIGURE 1. Lock-On Sequence Waveform Diagram—Pulse Gate Mode Switches after Preamble Detection

Circuit Operation (Continued)

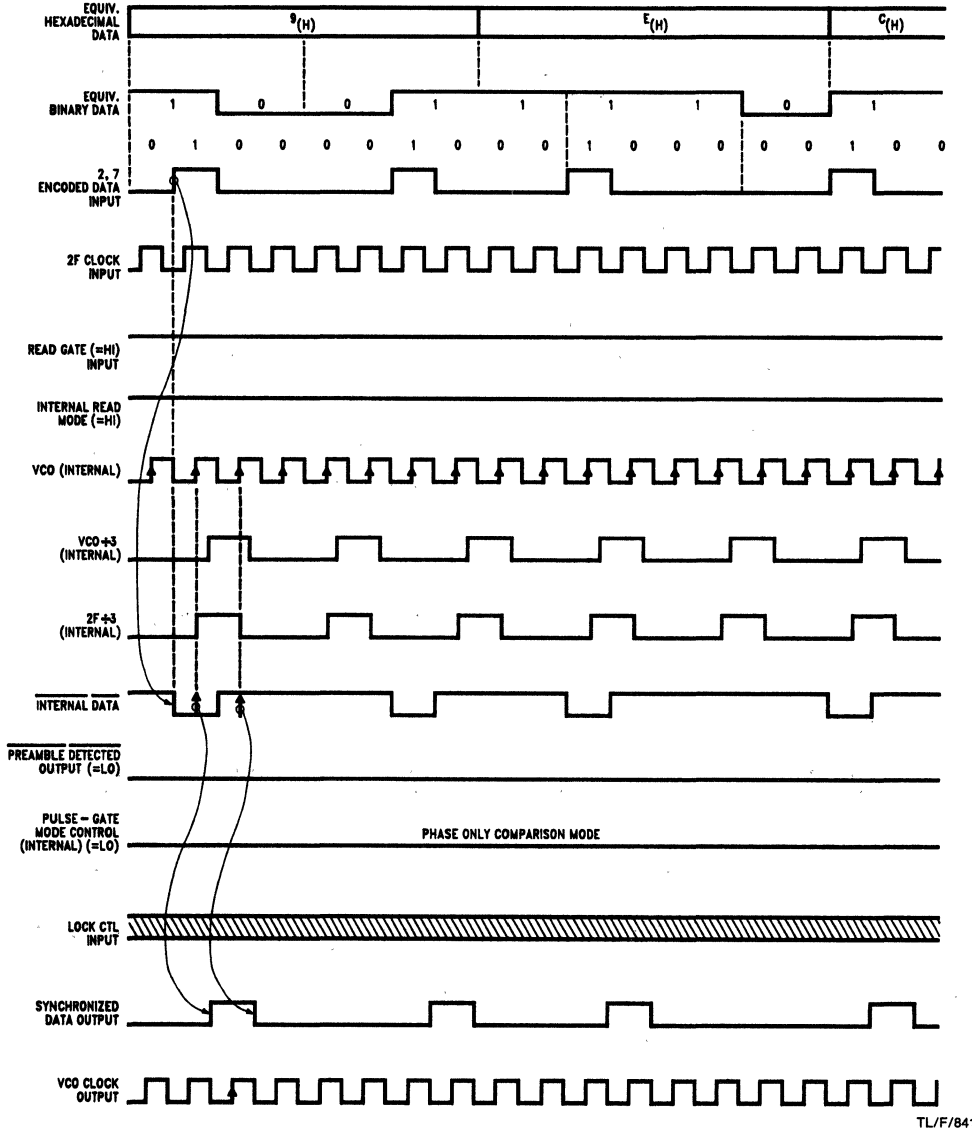


TL/F/8418–5

Note 1: L = Number of VCO cycles required for VCO to lock—typically 20 but determined by external component value.
Note 2: At L + 42 (Pattern = 1-0-0) or L + 56 (Pattern = 1-0-0-0), 15 patterns have been detected.
Note 3: VCO ÷ 3 (or 4) being compared with 2F ÷ 3 (or 4) in the non-read mode.
Note 4: VCO gated by DATA being compared with ENCODED DATA.
Note 5: PREAMBLE SELECT = LO; 1-0-0 pattern selected—so 2F & VCO are being divided by 3.

FIGURE 2. Lock-On Sequence Waveform Diagram—Pulse Gate Mode Switches Immediately After READ GATE is Asserted

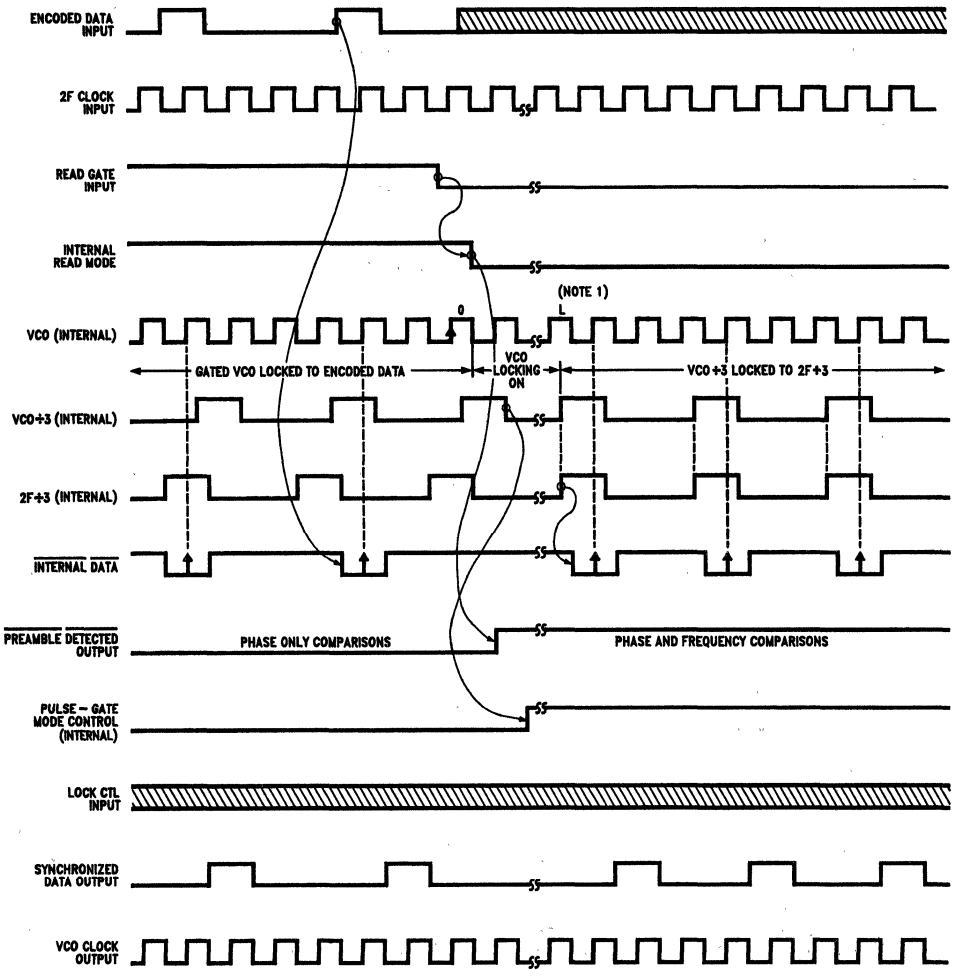
Circuit Operation (Continued)



TL/F/8418–6

FIGURE 3. Locked-On Waveform Diagram

Circuit Operation (Continued)



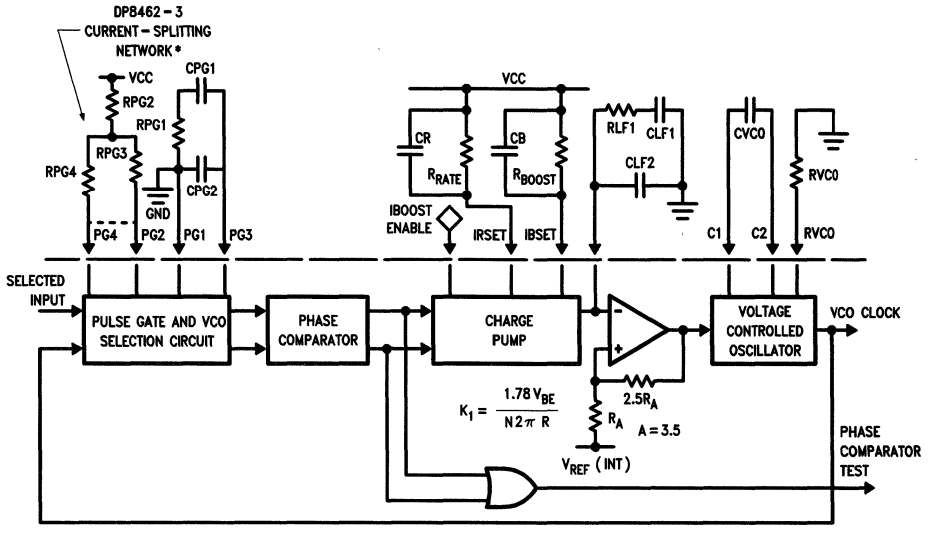
TL/F/8418–7

Note 1: L indicates the number of cycles required for the VCO to lock to the 2f-Clock.

Note 2: PREAMBLE SELECT = LO; 1-0-0 Pattern selected—so 2F & VCO being divided by 3.

FIGURE 4. Lock-Ending Sequence Waveform Diagram

Circuit Operation (Continued)



TL/F/8418–8

For DP8462-4 window testing RPG4 = RPG3 = 0Ω and RPG2 = 4.7 kΩ. For DP8462-3 window testing RPG4 = 0Ω, RPG3 = 330Ω and RPG2 = 1.5 kΩ.

FIGURE 5. Phase-Locked-Loop Section

BIT JITTER TOLERANCE

The two options of the DP8462, the -4 and -3 offer decreasing window errors (respectively) so that the parts may be selected for different data rates (up to 20 Mbit/sec). The -4 part will be used in most low data rate applications. As an example, at the 5 Mbit/sec data rate of most 5¼ inch drives, T = 200 ns so that from the Electrical Characteristics Table, t_{WINDOW} = 10 ns. The chip therefore contributes up to 10 ns of window error, out of the total allowable error of 50 ns (half the 2f-clock period of 100 ns). This allows the disk drive to have a margin of 40 ns of jitter on the transition position before an error will occur. The bit jitter tolerance can be improved by adjusting the window center using PG2 and PG4. A current splitting network consisting of RPG3 and RPG4 can be used to adjust the delay line. This adjustment is internally compensated for V_{CC} and temperature variation.

ANALOG CONNECTIONS TO THE DP8462

External passive components are required for the Pulse Gate, Charge Pump, Loop Filter and VCO as shown in *Figure 5*. The information provided here is for guidelines only. The user should select values according to his own system requirements. Phase-Locked Loops are complex circuits that require detailed knowledge of the specific system. Factors such as loop gain, stability, response to change of signal, lock-on time, etc. are all determined by the external components. In many disk systems these factors are critical, and National Semiconductor recommends the designer be knowledgeable of phase-locked-loops, or seek the advice of an expert. Inaccurate design will probably result in excessive disk error rates. The phase-locked-loop in the DP8462 has many advantages over all but the most sophisticated discrete designs, and if the component values are selected correctly, it will offer significant performance advantages. This should result in a reduction of disk error rates over equivalent discrete designs.

PULSE GATE

There are 6 external components connected to the Pulse Gate as shown in *Figure 6* with the associated internal components. Of these, RPG3 and RPG4 are optional and may be omitted if adjustment of the delay line is not desired. The values of RPG1, RPG2, RPG3, RPG4, CPG1, and CPG2 are dependent on the data rate. RPG1 and RPG2 are inversely proportional to the data rate, while CPG1 and CPG2 are proportional. Table I shows component values for the data rates given. Component values are calculated by selecting RPG2′ from Table I [RPG2′ = RPG2 + (RPG3//RPG4)]. If RPG4 = 0Ω, RPG2′ = RPG3 + RPG2. Next calculate

$$CPG1 = \left(\frac{2.12 \times 10^5}{890 + RPG2'}\right)\left(\frac{1}{100 \times Rs}\right)^2$$

$$CPG2 = \frac{1}{10} CPG1, \text{ and}$$

$$RPG1 = \left(\frac{890 + RPG2'}{2.38 \times 10^5}\right)(100 \times Rs).$$

In the above equation Rs is the rotational speed and, for 3600 RPM, Rs = 60 Hz. A rotational speed of 3600 RPM was assumed for the calculations in Table I. For data rates not listed, RPG2 may be approximated as (30 kΩ/fDATA) − 1.2 kΩ = RPG2 where fDATA is the data rate in Megabits per second. RPG3 and RPG4, in conjuction with RPG2, form a "current-splitting-network" that can be used to adjust the delay line; thus adjusting the decode window early or late. RPG2 should be made large with respect to RPG3 and RPG4 and a potentiometer can be used for RPG4—with its value centered around that of RPG3. For example, at Data Rate = 5 Mbits/sec., Table I dictates that RPG2′ should be 4.7k. If the delay line is to be made adjustable, then one could pick RPG2 = 4.3k and RPG3 = 800Ω. Now, using a 1.6 kΩ potentiometer for RPG4, RPG4 = 800Ω would give RPG2′ = 4.7 kΩ and would provide standard window synchronization; varying RPG4 high or low, however, would

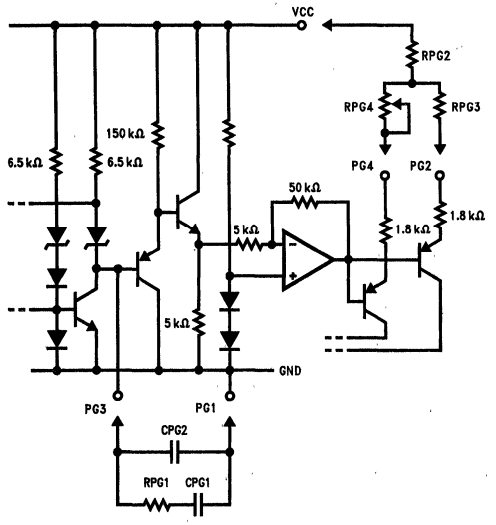
Circuit Operation (Continued)

shift the window late or early, respectively. If no adjustment is desired, then PG2 and PG4 should be tied together and only RPG2 should be used. Components with 5% tolerance will suffice.

TABLE I. Pulse Gate Component Selection Chart

Data Rate	RPG2'	RPG1	CPG1	CPG2
5 Mbit/sec	4.7 kΩ	150Ω	1 μF	0.1 μF
10 Mbit/sec	1.8 kΩ	68Ω	2.2 μF	0.22 μF
15 Mbit/sec	750Ω	39Ω	3.9 μF	0.39 μF

Where [RPG2' = RPG2 + RPG3//RPG4]



TL/F/8418–9

FIGURE 6. Pulse-Gate Controls

CHARGE PUMP

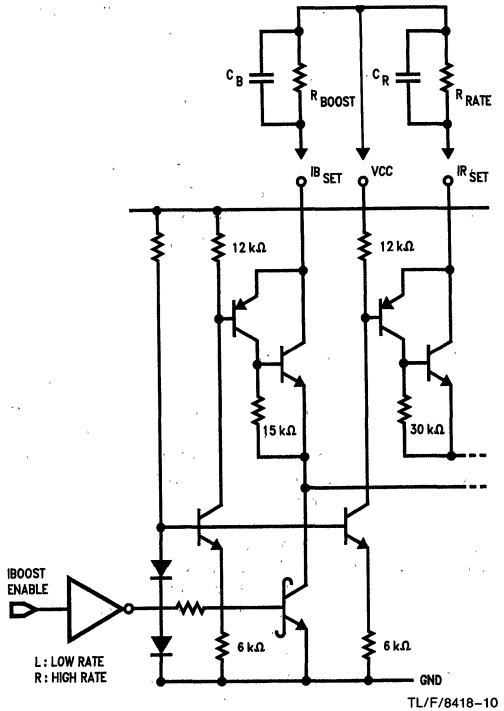
Resistors R_{RATE} and R_{BOOST} determine the charge pump current. The Charge Pump bidirectional output current is related to the input current according to the relationship specified in the DC Electrical Characteristics Table. In the high tracking rate with IBOOST ENABLE high, the input current is I_{BSET} + I_{RSET}, i.e., the sum of the currents through R_{BOOST} and R_{RATE} from V_{CC}. In the low tracking rate, with IBOOST ENABLE low, this input current is I_{RSET} only.

A recommended approach would be to select R_{RATE} first. The External Component Limits table allows R_{RATE} to be 0.4 kΩ to 4.0 kΩ, so for simplicity select R_{RATE} = 820Ω. A typical loop gain change of 2:1 for high to low tracking rate would require R_{BOOST} = R_{RATE} or 820Ω. Referring to *Figure 7* the input current is effectively V_{BE}/R_{RATE} in the low tracking rate, where V_{BE} is an internal voltage. This means that the current into or out of the loop filter is approximately 2.0 V_{BE}/R_{RATE}, or in this example approximately 1.8 mA. Note that although it would seem the overall gain is dependent on V_{BE}, this is not the case. The VCO gain is altered internally by an amount inversely proportional to V_{BE}, as detailed in the section on the Loop Filter. This means that as V_{BE} varies with temperature or device spread, the gain will remain constant for a particular fixed set of values of R_{RATE} and R_{BOOST}. This alleviates the need for potentiometers to select values for each device. The tolerance required for these two resistors will depend on the total loop gain tolerance allowed, but 5% would be typical. Also V_{CC} by-pass capacitors are required for these two resistors. A value of 0.01 μF is suitable for each.

VCO

The value of R_{VCO} is fixed at 1 kΩ ± 1% in the External Component Limits table. *Figure 8* shows how R_{VCO} is connected to the internal components of the chip. This value was fixed at 1 kΩ to set the VCO operating current such that optimum performance of the VCO is obtained for production device spreads. This means fixed value components will be adequate to set the VCO center frequency for production runs. The value of C_{VCO} can therefore be determined from the VCO frequency f_{VCO}, using the equation: C_{VCO} = $[1/(R_{VCO})(f_{VCO})]$ − 5 pF where f_{VCO} is twice the input data rate. As an example, for a 5 Mbit/sec data rate, f_{VCO} = 10 MHz, requiring that C_{VCO} = 95 pF. This does not take into account any lead capacitance on the printed circuit board; the user must account for this. The amount of tolerance a particular design can afford on the center frequency will determine the capacitor tolerance. The capacitor is connected to the internal circuitry of the chip as shown in *Figure 9*.

As the data rate increases and C_{VCO} gets smaller, the effects of unwanted parasitic capacitances influence the fre-



TL/F/8418–10

FIGURE 7. I_{RATE} Set and I_{BOOST} Set

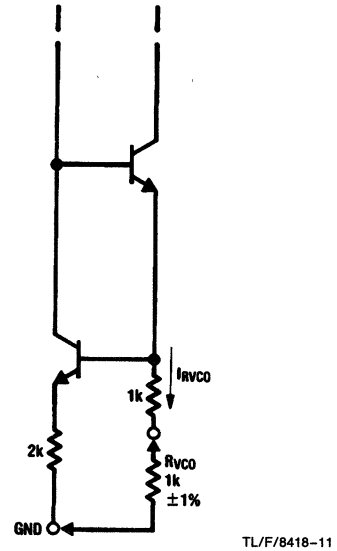
Circuit Operation (Continued)

quency. As a guide the graph of *Figure 10* shows approximately the value of C_{VCO} for a given data rate.

The VCO center frequency may be determined by: 1) holding pin 4 at ground potential and measuring the VCO frequency (−20% value); 2) holding pin 4 at approximately 3 volts and measuring the VCO frequency (+20% value); 3) averaging the two measured frequencies for the equivalent center frequency.

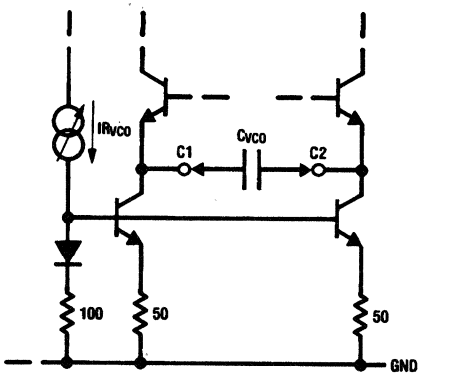
LOOP FILTER

The input current into the Buffer Amplifier is offset by a matched current out of the Charge Pump, and even so is much less than the switching current in or out of the Charge Pump. It can therefore be assumed that all the Charge Pump switching current goes into the Loop Filter components R_1 and C_1 and C_2. The tolerance of these compo-



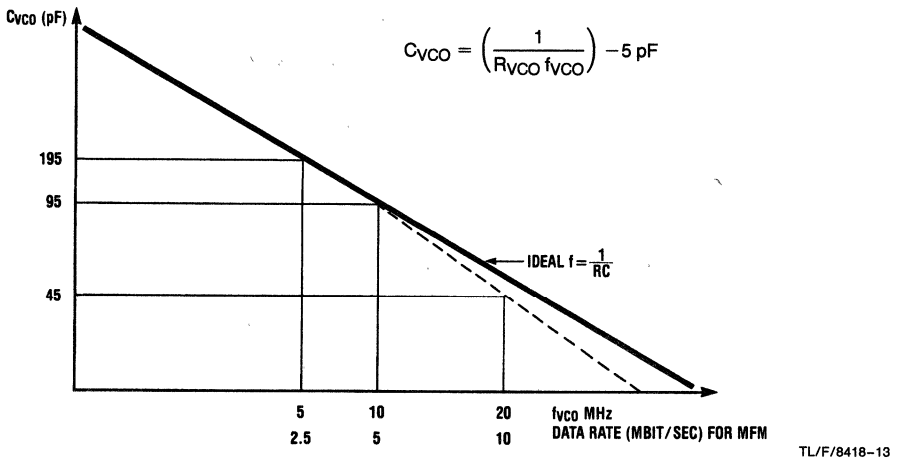
TL/F/8418–11

FIGURE 8. VCO Current Setting Resistor



TL/F/8418–12

FIGURE 9. VCO Capacitor



TL/F/8418–13

FIGURE 10. VCO Capacitor Value for Disk Data Rates

Circuit Operation (Continued)

nents should be the same as R_{RATE} and R_{BOOST}, and will determine the overall loop gain variation. The three components connected to the Charge Pump output are shown in *Figure 11*. Note the return current goes to analog GND, which should be electrically very close to the GND pin itself.

The value of capacitor C_1 determines loop bandwidth—the larger the value the longer the loop takes to respond to an input change. If C_1 is too small, the loop will track any jitter on the ENCODED DATA input and the VCO output will follow this jitter, which is undesirable. The value of C_1 should therefore be large enough so that the PLL is fairly immune to phase jitter but not large enough that the loop won't respond to longer term data rate changes that occur on the disk drive.

The damping resistor R_1 is required to damp any oscillation on the VCO input that would otherwise occur due to step function changes on the input. A value of R_1 that would give a phase margin of around 45 degrees would be a reasonable starting point.

The main function of the capacitor C_2 is to "smooth" the VCO input voltage. Typically its value will be less than one tenth of C_1.

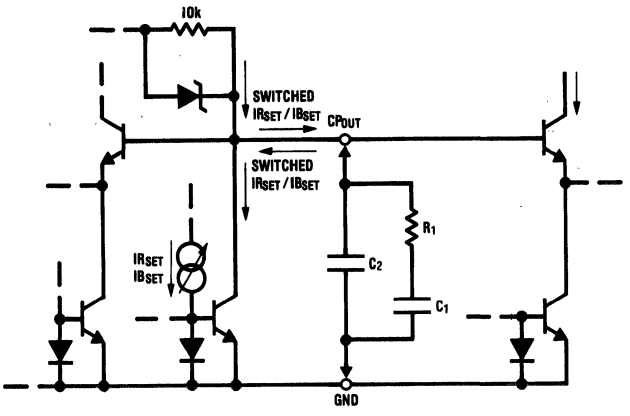
Figure 12 shows the relevant phase-locked-loop blocks that determine system response, namely the Phase Detector, Filter/Buffer Amplifier, and VCO. The Phase Detector (Phase Comparator and Charge Pump) produces an aggregate output current i which is proportional to the phase difference between the input signal and the VCO signal. The constant (K_1) is $\frac{1.78\ V_{BE}}{2\pi RN}$ amps per radian where $N = \frac{f_{VCO}}{f_{DATA}}$. R is either R_{RATE} or $R_{RATE} \parallel R_{BOOST}$. This aggregate current feeds into or out of the filter impedance (Z), producing a voltage to the VCO that regulates the VCO frequency. The VCO gain constant is 0.4 ω_{VCO}/V_{BE} radians per second per volt. Under steady state conditions, i will be zero and there will be no phase difference between the input signal and the VCO. Any change of input signal will produce a change in VCO frequency that is determined by the loop gain equation. This equation is determined from the gain constants K_1, A and K_2 and the filter v/i response.

The impedance Z of the filter is:

$$\frac{1}{sC_2} \parallel \left(\frac{1}{sC_1} + R_1\right) = \frac{1 + sC_1R_1}{sC_1\left(1 + \frac{C_2}{C_1} + sC_2R_1\right)}$$

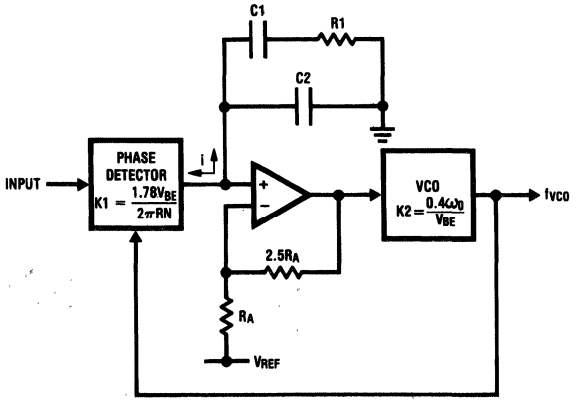
If $C_2 < C_1$ then the impedance Z approximates to:

$$\frac{1 + sC_1R_1}{sC_1\,(1 + sC_2R_1)}$$



TL/F/8418–14

FIGURE 11. Charge Pump Out



TL/F/8418–15

FIGURE 12. Loop Response Components

Circuit Operation (Continued)

The overall loop gain is then

$$G(s) = \frac{K_1 A K_2}{s} \times \frac{1 + sC_1R_1}{sC_1(1 + sC_2R_1)}$$

Let $G_{(K)} = K_1 A K_2$

$$F(s) = \frac{1 + SC_1R_1}{SC_1(1 + SC_2R_1)}$$

The Overall Closed Loop Gain is:

$$\frac{\phi OUT}{\phi IN} = \frac{G_{(K)} F(s)}{s + G_{(K)} F(s)}$$

Substituting, We Get

$$\frac{\phi OUT}{\phi IN} = \frac{G_{(K)}(SC_1R_1 + 1)}{S^3 R_1 C_1 C_2 + S^2 C_1 + GK(SC_1R_1 + 1)}$$

$$= \frac{(G_{(K)}/C_1)(SR_1C_1 + 1)}{S^3 R_1C_2 + S^2 + SG_{(K)}R_1 + G_{(K)}/C_1}$$

If $C_2 \ll C_1$, we can ignore the 3rd Order Component introduced by C_2 then:

$$\frac{\phi OUT}{\phi IN} = \frac{(G_{(K)}/C_1)(SR_1C_1 + 1)}{S^2 + SG_{(K)}R_1 + G_{(K)}/C_1}$$

This is a second Order Loop and can be solved as follows:

$$S^2 + SG_{(K)}R_1 + G_{(K)}/C_1 = S^2 + 2\zeta\omega_n S + \omega_n^2$$

$$\therefore C_1 = \frac{G_{(K)}}{\omega_n^2}$$

$$R_1 = \frac{2\zeta\omega_n}{G_{(K)}}$$

From the above equations:

$$\omega = (G(K)/C1)^{1/2}$$

$$G(K) = K1 \times A \times K2 =$$

$$[(0.89 \times V_{BE})/(2 \times Pi \times R)] \times [(0.4 \times W_{VCO})/V_{BE}] \times [3.5]$$

2,7 coded data has a 2.67 to 1.0 frequency range within the data field. The expression $K = (0.89 \times V_{BE}/2 \times Pi \times R)$ is valid when the VCO frequency is twice the ENCODED DATA frequency. In order to make this equation more general, it may be written as follows: $K = (1.78 \times V_{BE})/(2 \times Pi \times R \times N)$ where N is defined as the VCO frequency divided by the encoded data pulse frequency, or $N = F_{VCO}/F_{DATA}$ ($N = 3$ for maximum data rate and $N = 8$ for minimum data rate). Now G(K) can be written as follows:

$$G(K) = [(1.78 \times V_{BE})/(2 \times Pi \times R \times N)] \times [(0.4 \times \omega_{VCO})/V_{BE}] \times [3.5]$$

$$= (2.5 \times F_{VCO})/(R \times N)$$

$$\omega n = [(2.5 \times F_{VCO})/(C1 \times R \times N)]^{1/2}$$

where,

$R = R_{RATE}$ in the low track rate;

$R = (R_{RATE}//R_{BOOST})$ in the high track rate.

From the above equations:

$$\omega_N = (R1 \times G(K))/(2\zeta)$$

$$G(K) = C1 \times (\omega_N^2)$$

$$\zeta = (\text{damping factor}) = (R1 \times \omega_N \times C1)/2$$

The damping factor should be approximately 0.5 when ω_N is minimum. Response to bit shift is minimized when the damping factor is small; however, if the damping factor drops much below 0.5, the system tends to be oscillatory (underdamped). Additionally, loop performance is poor (excessive phase-acquisition times) if the damping factor becomes much larger than 1.0. Any increase in loop bandwidth (due to R decreasing in the high track rate) produces a proportional increase in the damping factor, and should be limited to the point where the maximum damping factor is 1.0. With the damping factor range established, loop design can proceed.

From the Disk Interface Design Guide And User's Manual Chapter 1, Section 1.3–1.7, it is shown that a 946 krads/sec loop bandwidth during acquisition results in a 7 byte crystal reference clock acquisition and data frequency acquisition (VCO settled to within 2 ns of window center). We recommend that these design guide sections be reviewed in conjunction with the DP8462 data sheet in order to obtain a more detailed explanation of the loop bandwidth selection used here, as well as for disk system PLLs in general.

This design example is for a 10 MBit/sec data rate and assumes that the IBOOST ENABLE pin is tied to the $\overline{\text{PREAMBLE DETECTED}}$ pin. This results in the track rate being switched from high to low after four bytes of preamble are detected. As an alternative, the IBOOST ENABLE pin may be tied to an inverted READ GATE signal, resulting in the track rate switching immediately to low when READ GATE is asserted. This is discussed further in the Design Guide.

We will assume a 1-0-0-0 . . . preamble. During acquisition we are in the high track rate and thus ω_N is at maximum value. In the read mode the highest frequency pattern we can encounter is 1-0-0 . . . ; however, ω_N will be lower since we will be in the low track rate.

$$\omega_N = [(2.5 \times F_{VCO})/(C1 \times R \times N)]^{1/2}$$

Choose $R_P = R_{RATE} // R_{BOOST} = 575\Omega$*

$$946 \times 10^3 = [(2.5 \times 20 \times 10^6)/(C1 \times 575 \times 4)]^{1/2}$$

$C1 = 0.028\ \mu F$ Choose $C1 = 0.022\ \mu F$

We don't want ζ to exceed 1.0. Therefore,

$$\zeta = \frac{\omega_N \times R1 \times C1}{2}$$

$$1.0 = \frac{(946 \times 10^3 \times R1) \times 0.022 \times 10^{-6}}{2}$$

$R1 = 96\Omega$

Choose $R1 = 100\Omega$

Note: Designing a PLL is an iterative procedure. For the DP8462, design values for R_{RATE} and R_{BOOST} typically range from 700Ω to 1.5 kΩ. The application note provides a more thorough discussion for choosing these values.

The continuous-behavior (non-quantized) approximation used to predict loop performance assumes that the phase detector output is constantly proportional to the input phase difference. In reality, the phase detector output is a pulse applied for a period of time equal to the phase difference. The function of C2 is to "smooth" the phase detector output (VCO control voltage) over each cycle. C2 also adds a second pole to the filter transfer function. This pole should be far enough outside the loop bandwidth (at least one order of magnitude) that its phase and amplitude contribution is negligable in the loop bandwidth. If

$$C2 = C1/50 = 390\ pF$$

the acquisition performance and the margin loss are not significantly changed from the predictions. If a larger C2 is used, the margin loss can be reduced at the expense of the acquisition time. This may be desirable for some systems. Please see the Disk Interface Design Guide And User's Manual Chapter 1, Sections 1.3–1.7 for a discussion of the function of C2.

Circuit Operation (Continued)

As soon as the PREAMBLE DETECTED output goes low we switch to the low track rate. To maintain stability we must ensure that $\zeta_{min} \geq 0.5$.

ζ_{min} occurs when ω_N is minimum; i.e., when we have seven consecutive zeroes (N = 8).

$$\zeta_{min} = \frac{(\omega_{Nmin} \times R1 \times C1)}{2}$$

$$0.5 = \frac{(\omega_{Nmin} \times 100 \times 0.022 \times 10^{-6})}{2}$$

$$\omega_{Nmin} = 454.5 \text{ krads/sec}$$

We can now calculate R_{RATE}

$$\omega_{Nmin} = [(2.5 \times F_{VCO})/(C1 \times R_{RATE} \times N)]^{1/2}$$

$$454.5 \times 10^3 = [(2.5 \times 20 \times 10^6)/(0.022 \times 10^{-6} \times R_{RATE} \times 8)]^{1/2}$$

Therefore, $R_{RATE} = 1.375\ k\Omega$

Choose, $R_{RATE} = 1.2\ k\Omega$

Now we calculate ω_{Nmax} and ζ_{max} in the low track rate

$$\omega_{Nmax} = [(2.5 \times 20 \times 10^6)/(0.022 \times 10^{-6} \times R_{RATE} \times 3)]^{1/2}$$

$$\omega_{Nmax} = 794 \text{ krads/sec}$$

$$\zeta_{max} = \frac{(\omega_{Nmax} \times R1 \times C1)}{2}$$

$$\zeta_{max} = 0.87$$

The final component to be determined is R_{BOOST}

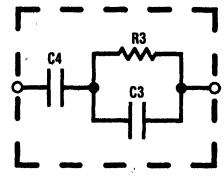
$$\text{Since, } Rp = \frac{R_{BOOST} \times R_{RATE}}{R_{BOOST} + R_{RATE}}$$

$$575 = \frac{R_{BOOST} \times 1.2 \times 10^3}{R_{BOOST} + 1.2 \times 10^3}$$

Therefore, $R_{BOOST} = 1.1\ k\Omega$

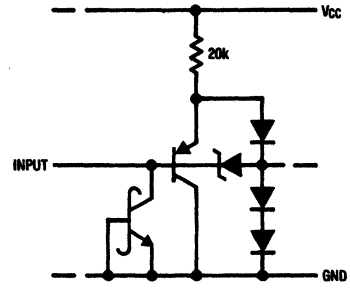
DIGITAL CONNECTIONS TO THE DP8462

Figure 16 shows a connection diagram for the DP8462 in a typical application. All logic inputs and outputs are TTL compatible as shown in *Figure 14* and *15*. The VCO Clock output is 74AS compatible. All other outputs are 74ALS compatible. All inputs are 74ALS compatible and therefore can be driven easily from any 74 series devices.



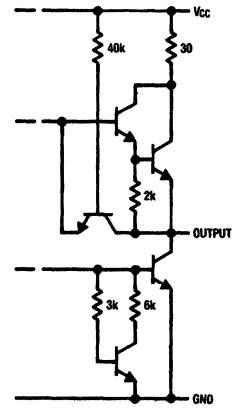
TL/F/8418–17

FIGURE 13. Alternate Loop Filter Configuration



TL/F/8418–16

FIGURE 14. Logic Inputs



TL/F/8418–18

FIGURE 15. Logic Outputs

TABLE II. Loop Filter External Component Values

Data Rate (NRZ)	Pulse Gate Components (Note 3)				Charge Pump (Note 1)		Loop Filter (Note 2)		
	R_{PG2}	R_{PG1}	C_{PG1}	C_{PG2}	R_{RATE}	R_{BOOST}	R_1	C_1	C_2
5 Mbit/sec	4.7k	150Ω	1.0 μF	0.1 μF	820Ω	1.5 kΩ	100Ω	0.03 μF	600 pF
10 Mbit/sec	1.8k	68Ω	2.2 μF	0.22 μF	1.2 kΩ	1.1 kΩ	100Ω	0.022 μF	390 pF
15 Mbit/sec	0.75k	39Ω	3.9 μF	0.39 μF	820Ω	2.7 kΩ	33Ω	0.082 μF	1600 pF

Note 1: Component tolerances are system dependent, they depend on how much loop gain deviation can be tolerated.

Note 2: Component tolerances are typically 5% but they depend on the amount of Loop Bandwidth tolerance that can be accepted.

These values have been altered from calculated values based on empirical tests of the loop.

Note 3: Component tolerances typically 10%, not critical.

Circuit Operation (Continued)

The incoming data from the pulse detector in the drive is connected to the ENCODED DATA input. PREAMBLE SELECT input is tied high or low depending on whether the user's system is employing 1000 or 100 preamble pattern. The LOCK CTL input is to be tied high or low depending on whether or not it is desired to keep the PLL in Phase and Frequency comparison mode while detecting preamble. Phase and Frequency comparison lock while detecting preamble will eliminate the chances of the PLL locking onto a harmonic of the preamble frequency when Read mode is first entered. (Susceptibility to a harmonic lock is increased when using the 1000 preamble). Since a high level on IBOOST ENABLE input puts the PLL in high track rate, it should be held high during Non-Read (standby) mode so that a quick lock is achieved upon entering Read mode. Once the PLL is locked onto the incoming data, however, this input should be taken low. Although the user is free to do this anytime, one possible method is to tie this input to the $\overline{\text{PREAMBLE DETECTED}}$ output of the chip—as shown in *Figure 16*. The READ GATE input is used to place the chip in and out of Read mode and therefore should be tied to the controller and/or a 2, 7 code Encoder/Decoder).

As for the outputs, SYNCHRONIZED DATA and VCO CLOCK may be tied to the Encoder/Decoder—which in turn would deseriallize and decode the data before sending it to the controller. $\overline{\text{PREAMBLE DETECTED}}$ output can be tied to the controller and/or the Encoder/Decoder to provide an indication when 4 consecutive bytes of preamble pattern have been detected. The only output that is not shown in *Figure 16* is the PHASE COMPARATOR TEST output. This output is the logical OR of the Phase Comparator's outputs (Charge-Up and Charge-Down inputs of the Charge-Pump). As such, pulses generated at this output provide information about the loop filter's behavior in that the envelope of the pulses generated at this output is a waveform that represents the loop filter's response to any phase difference detected by the Phase Comparator.

Finally, to improve noise immunity, Digital and Analog VCC pins should be tied together and also the Digital and Analog Ground pins should be tied together. PG1 pin should also be grounded.

Applications of the DP8462 Data Synchronizer

The DP8462 is part of National Semiconductor's DP8460 Series Disk Chip Set and therefore, is designed to work in conjunction with other members of this family; such as DP8464—the pulse detector, and DP8466—the disk data controller. A typical system application employing these components is shown in *Figure 17*. The DP8462 is based upon the proven circuitry of the DP8465 (Data synchronizer and separator for the MFM code)—the first integrated circuit to place on one chip a PLL with features that offer the improved speed and reliability required by the disk industry. Not only does the chip simplify disk system design, but also provides fast lock-on to the incoming preamble. Once locked on, the loop is set into a lower bandwidth mode. This inherent loop stability allows for a sizable amount of jitter on the data stream, such as is encountered in many disk systems. Once in the stable tracking rate, the SYNCHRONIZED DATA output represents the incoming ENCODED DATA and is synchronous with VCO CLOCK. This synchronized data is then deserialized by the ENDEC using the VCO CLOCK.

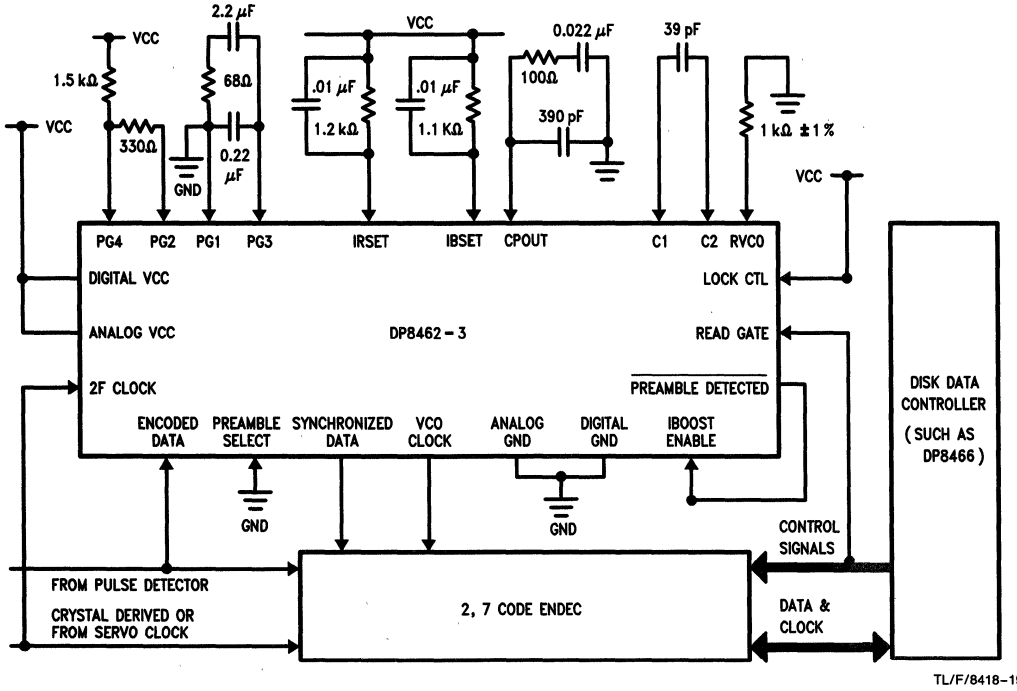
The DP8462 is capable of operating at up to a 20 Mbits/sec data rate and so is compatible with a wide assortment of disk drives. The faster data rates of the 8-inch and 14-inch disk drives will mandate the selection of the DP8462-3 parts with narrower window margin on the incoming data stream. This will also be the case when 5¼-inch drives achieve higher data rates. Some 8-inch and 14-inch disk drives incorporate the functions of the DP8462, but use many discrete ICs. In these cases, replacing these components with the DP8462 will offer reduced P.C. board area, lower cost, and improved performance while simplifying circuit testing.

Most 5¼-inch and many 8-inch and 14-inch disk drives manufactured at present do not incorporate any of the functions of the DP8462. This is so primarily because the PLL function is difficult to design and implement and requires circuitry which covers a large area of the printed circuit card. This is undesirable both from the drive size aspect and from the cost aspect (the cost includes soldering, testing, and adjusting the components). Consequently, most smaller disk drives output RLL encoded data so that the phase-locked-loop and data separation have to be performed by the controller. The DP8462 will therefore replace these functions in controller designs, as shown in *Figure 18a*.

System design criteria may now change because the DP8462 is a one-chip solution, requiring only a few external passive components with fixed values. It operates from a +5V supply, consumes about 0.5W, and is housed in a narrow 24-pin package. The circuitry has been designed so that the external resistors and capacitors need not be adjustable; the user chooses the values according to the disk drive requirements. Once selected, they will be fixed for that particular drive type. These features make it possible to transfer these functions to the disk drive, as shown in *Figure 18b*. Apart from a slight increase in board area, the advantages outweigh the disadvantages. First, the components selected are fixed for each type of drive and this facilitates the problem of interchangeability of drives. At present, components in the controller are adjusted to function with each specific drive; with the DP8462 in the drive, component adjustment will no longer be required. Second, there is often a problem of reliability of data transfer. The incoming data signal is susceptible to noise, bit shift, etc. Soft errors will occur when the incoming disk data bit position is outside the Pulse Gate window as it is being synchronized to the VCO clock in the phase-locked-loop. Obviously, the nearer the PLL is to the data source, the less chance there is that errors will occur. Thus placing the DP8462 in the drive will increase the reliability of data transfer within the system.

Applications of the DP8462 Data Synchronizer (Continued)

A third advantage is data rate upgrading. Most 5¼-inch drives have 5 Mbit/sec data rate because the early drives were made with this data rate. This meant the controllers had to be designed with PLLs which operate at this data rate. It is therefore difficult for drive manufacturers to introduce new drives that are not compatible with existing controllers. Since no new standard data rate has emerged, they must continue to produce drives at this data rate to be compatible with the controllers on the market. With the DP8462 in the drive, and its associated components set for the drive's data rate, it no longer becomes a problem to increase the data rate, assuming the controller's digital circuitry can accommodate the change. This will allow the manufacturers to increase the bit density and therefore the capacity of their drives.

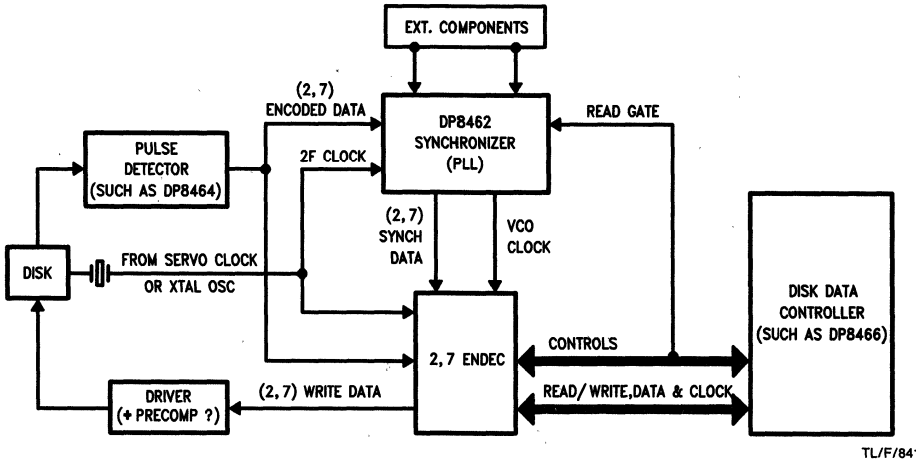


TL/F/8418–19

FIGURE 16. Typical Connection to DP8462 For:

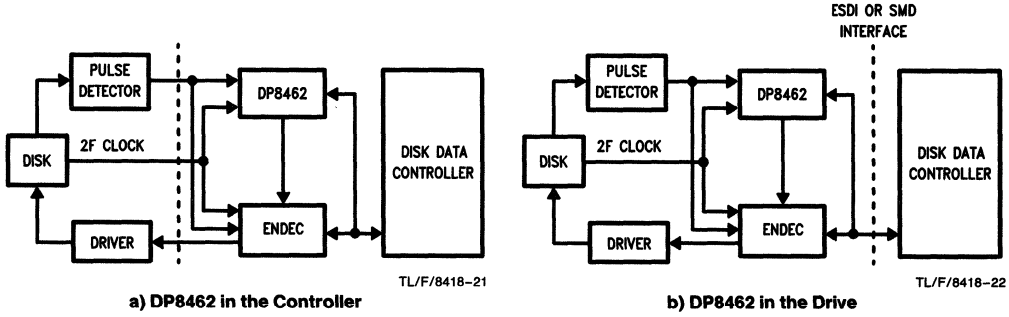
1) RLL (2,7 Code) Data Input, 10 Mbit/sec Data Rate
2) 1-0-0 Preamble Pattern
3) PLL to stay in Phase-Frequency Comparison mode until 4 bytes of Preamble Detected
4) PLL to stay in high Track Rate until $\overline{\text{PREAMBLE DETECTED}}$ asserted
5) Delay line left unadjusted (PG2 & PG4 shorted together)

Applications of the DP8462 Data Synchronizer (Continued)



TL/F/8418–20

FIGURE 17. Typical Application of DP8462 in a System Employing RLL (2,7) Code



TL/F/8418–21

a) DP8462 in the Controller

TL/F/8418–22

b) DP8462 in the Drive

FIGURE 18. Two Different Methods of Utilizing DP8462

PRECAUTIONS IN BREADBOARDING AND PCB LAYOUT

The DP8462 contains a high performance analog PLL and certain precautions must be taken when breadboarding or designing a PCB layout. The following guidelines should be adhered to when working with the DP8462:

1) Do not wire wrap.
2) Keep component lead lengths short, place components as close to pins as possible. This applies to R1, C1, R2, C_{VCO}, R_{RATE}, R_{BOOST}, C_{RATE}, C_{BOOST}, RPG1, RPG2, and CPG1.
3) Provide a good ground plane and use a liberal amount of supply bypassing. The quieter a PLL's environment, the happier it is.
4) Avoid routing any digital leads within the vicinity of the analog leads and components.

We have used a PC board approach to breadboarding the DP8462 that gives us an excellent ground plane and keeps component lead lengths very short. With this setup we have found very stable and reliable operation. Illustrations of component layout is shown in *Figure 19*. Note that the board layout is a recommendation not a requirement.

ADDITIONAL NOTES

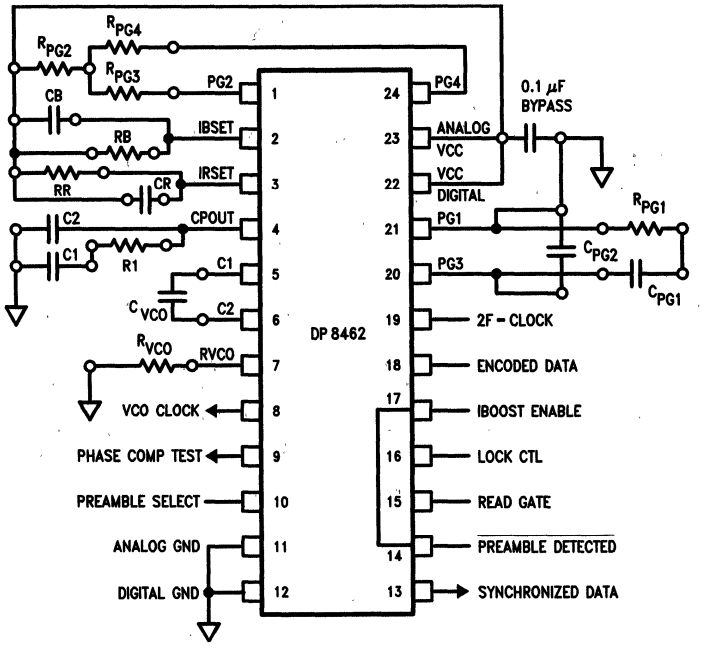
1) PG1 should be grounded to improve noise immunity.
2) 2F clock must be applied at all times; without the 2F clock, the pulse gate circuitry will not operate properly making it impossible to lock onto the incoming data stream.
3) The programming capacitor for the V_{CO} can be calculated as:

$$C_{VCO} = 1/(f_{VCO} * R_{VCO}) - 5\,pF$$

The 5 pF value is due to parasitic internal device capacitance.

4) Care must be taken in final PC board layout to minimize pin to pin capacitance, particularly in multi-layer printed circuit boards.
5) Please refer also to Precaution For Disk Data Separator Designs, NSC Application Note AN-414.

Connection Diagrams (Continued)



TL/F/8418–23

FIGURE 19. Recommended Component Layout

National Semiconductor

DP8463B (2,7) ENDEC

General Description

The DP8463B (2,7) ENDEC performs the encoding and decoding for a disk memory system using a (2,7) Run-Length-Limited (RLL) code. This code gives a disk system the ability to record up to 50% more message data in the same media space without any increase in the Flux Changes per Inch (FCI) density, when compared to a system using Modified Frequency Modulation (MFM) coding. The DP8463B also performs other functions of writing or reading format segments that can not be done by a disk data controller. These additional functions include the writing and reading of Preambles (PLL synchronization fields) and various soft-sector format Address Marks that are compatible with (2,7) RLL code. The user may also select different lengths of preamble to count before the DP8463B issues a Lock Detect signal. The encoded CODE OUT output is automatically resynchronized to the 2f CRYSTAL/SERVO CLOCK for perfect periodic writing regardless of the duty cycle of the WRITE CLOCK input and regardless of the phase relationship between the WRITE CLOCK and the 2f CRYSTAL/SERVO CLOCK. The READ/REFERENCE CLOCK output is switched between clock sources without generating any short pulses. In addition to the detecting of standard ESDI and SMD Address Marks, there is an optional noise tolerant mode that allows the recognizing of an address mark gap even with a bit or two of noise.

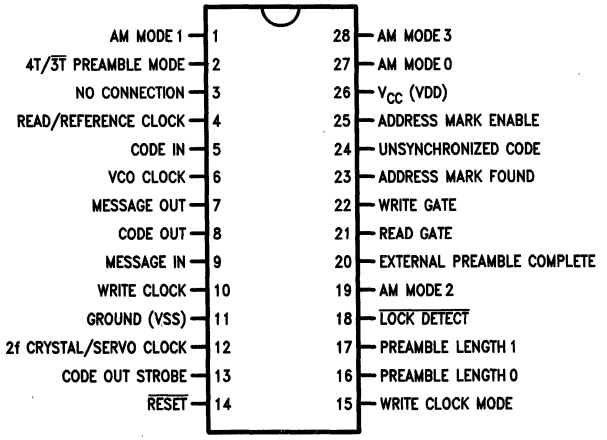
The DP8463B is compatible with the Storage Module Drive (SMD) and Enhanced Small Device Interface (ESDI) functional specifications, and has a format mode similar to the one used in ST-506 devices. The Input/Output (I/O) of the DP8463B are active-high, except LOCK DETECT, so inverting line drivers should be used for interfacing with the active-low I/O of ESDI. The term "Message" is used to designate unencoded data, also referred to as NRZ Data in disk literature. The term "Code" designates the encoded data.

Features

- Up to 50% increase in message data density over MFM
- Encodes and decodes using IBM (2,7) Message/Code Table
- Programmable Formats
  - — Hard Sector
  - — Soft Sector with Address Mark preceding Preamble
  - — Soft Sector with Address Mark following Preamble
- Programmable Address Marks
  - — ESDI 3-Byte transitionless gap, preceding Preamble
  - — ESDI 3-Byte transitionless gap, noise tolerant
  - — SMD 3-Byte transitionless gap, preceding Preamble
  - — SMD 3-Byte transitionless gap, noise tolerant
  - — IBM 2-Byte gap with three transitions, preceding Preamble
  - — N7V 2-Byte Address Mark with code word not in message/code table that does not violate (2,7) code constraints, following Preamble
    (Above gap lengths are in message-bytes)
- Programmable Preamble length counted before LOCK DETECT issued
  - — Externally determined (e.g., from DP8462 Data Synchronizer)
  - — 6 Message Bytes
  - — 8 Message Bytes
- Code output is resynchronized to 2f CRYSTAL/SERVO CLOCK
- Glitchless Multiplexer is used to switch between READ/REFERENCE CLOCK sources
- Strobe available to clock CODE OUT output into external register
- ADDRESS MARK FOUND appears after first "1" bit following Address Mark
- Message Data Rate to 20 Megabits/second (Code rate = 40 Mb/s)
- Compatible with ESDI
- Compatible with SMD
- Compatible with DP8462 Data Synchronizer
- Compatible with DP8466 Disk Data Controller
- 2-micron dual metal CMOS
- Single +5V Supply
- Packages
  - — 28-pin Dual-In-Line Package
  - — 28-pin Plastic Chip Carrier

Connection Diagrams

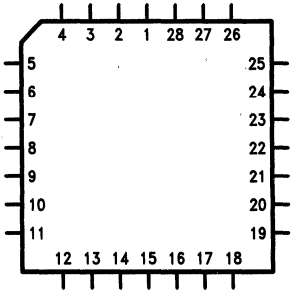
Dual-In-Line Package (DIP)



TL/F/9058–1

Order Number DP8463BN
See NS Package Number N28B

Plastic Chip Carrier
(Signal Assignments to the Pin Numbers are Identical to DIP)



TL/F/9058–2

Order Number DP8463BV
See NS Package Number V28A

Absolute Maximum Ratings

If Military/Aerospace specified devices are required, please contact the National Semiconductor Sales Office/Distributors for availability and specifications.

Supply Voltage	−0.5V to +7.0V
Input or Output Voltage	−0.5V to V_{CC} +0.5V
Storage Temperature	−65°C to +150°C
Lead Temperature	260°C

Recommended Operating Conditions

Temperature Range (T_A)	+0.0°C to +70°C

ESD rating is to be determined.

DC Electrical Characteristics

V_{CC} = +5V ±10%; Min./Max. limits apply across temperature range T_A unless otherwise specified.

Symbol	Parameter	Conditions	Min	Max	Units
V_{IH}	High Level Input Voltage		2.25		V
V_{IL}	Low Level Input Voltage			0.65	V
V_{OH1}	High Level Output Voltage	V_I = V_{CC} or GND I_O = 20 μA	V_{CC} − 0.1		V
V_{OH2}	High Level Output Voltage	V_I = V_{CC} or GND I_{OH} = −4.0 mA	3.5		V
V_{OL1}	Low Level Output Voltage	V_I = V_{CC} or GND I_O = 20 μA		0.1	V
V_{OL2}	Low Level Output Voltage	V_I = V_{CC} or GND I_{OL} = +4 mA		0.4	V
I_{IH}	High Level Input Current	V_I = V_{CC}		+10	μA
I_{CC} DY	Supply Current, Dynamic	V_I = V_{CC} or GND T_A = 25°C, f_{VCO} = 40 Mb/s		60	mA
I_{CC} SB	Supply Current, Standby	V_I = V_{CC} or GND T_A = 25°C, f_{VCO} = 1Mb/s		10	mA

AC Electrical Characteristics

(V_{CC} = 5V ±10%; Min./Max. limits apply across temperature unless otherwise specified. Output Load = 50 pF.)

Symbol	Parameter	Part No.	Min	Typ	Max	Units
f_{DATA}	Maximum Message Data Frequency (NRZ Data)	DP8463B-12	12			Mb/s
f_{VCO} $f_{C/S}$	Maximum VCO Frequency Maximum CRYSTAL/SERVO CLOCK frequency	DP8463B-12	24			Mb/s
t_{MISU}	Set-Up Time of MESSAGE IN Before WRITE CLOCK Positive Edge		10			ns
t_{MIH}	Hold Time of MESSAGE IN After WRITE CLOCK Positive Edge		10			ns
t_{MOSU}	Set-Up Time of MESSAGE OUT Before READ/REFERENCE CLOCK Positive Edge		24			ns
t_{MOH}	Hold Time of MESSAGE OUT After READ/ REFERENCE CLOCK Positive Edge		14			ns
t_{CISU}	Set-Up Time of CODE IN Before VCO CLOCK Positive Edge		7			ns
t_{CIH}	Hold Time of CODE IN After VCO CLOCK Positive Edge		10			ns
t_{COSU}	Set-Up Time of CODE OUT Before CODE OUT STROBE Positive Edge		10			ns
t_{COH}	Hold Time of CODE OUT After CODE OUT STROBE Positive Edge		5			ns
t_{PWUC}	Pulse Width of UNSYNCHRONIZED CODE		10			ns
t_{PWPC}	Pulse Width of EXTERNAL PREAMBLE COMPLETE		4			VCO Clock Periods
t_{pdIE}	Propagation Delay of IBM Encoder from WRITE CLOCK Positive Edge to CODE OUT		7		7	Code Bits
			+5		+45	ns
t_{pdIDC}	Propagation Delay of IBM Decoder from VCO CLOCK Postive Edge to READ/REFERENCE CLOCK Positive Edge		5.5		5.5	Code Bits
			+5		+62	ns
WCL	WRITE CLOCK Low		20%		80%	WRITE CLOCK Period
WCH	WRITE CLOCK High		20%		80%	

Note 1. Mb/s = Megabits/second
Code bit = period of 2f or VCO Clock

Pin Descriptions

Pin No	Description
POWER	
11	**GROUND (V_{SS})**
26	**V_{CC} (V_{DD})**
INPUT SIGNALS	
5	**CODE IN.** This is the encoded data output of the data synchronizer (e.g., DP8462). Each flux transition on the disk is a high level signal here with a width of one VCO clock period. The CODE IN is read by the DP8463B at the time of the positive going edge of the VCO CLOCK.
6	**VCO CLOCK.** This is the VCO clock output of the data synchronizer (e.g., DP8462). During the read mode, the VCO CLOCK is phase locked to the flux transitions on the disk.
9	**MESSAGE IN.** This is the unencoded "write data" from the disk data controller (e.g., DP8466). MESSAGE IN is read into the DP8463B by the positive going edge of the WRITE CLOCK input.
10	**WRITE CLOCK.** This clock strobes the MESSAGE IN "write data" into the encoder.
12	**2f CRYSTAL/SERVO CLOCK (2f C/S CLOCK).** This is the clock output of a disk drive's dedicated servo track or the buffered output of a crystal oscillator. This signal is the reference clock for generating the CODE OUT signal when the WRITE CLOCK MODE pin is low.
14	**$\overline{\text{RESET}}$.** An active low input resets various flip-flops when the next 2f C/S CLOCK positive edge occurs, so RESET should have a pulse width of two 2f C/S CLOCK periods. The DP8463B should be reset after each time power is applied.
20	**EXTERNAL PREAMBLE COMPLETE.** With PL1 and PL0 both low, an "active" level on this pin signals the DP8463B that the reading of the preamble, for phase locking purposes, is complete. The DP8463B then switches into its normal decoding mode and issues an active low LOCK DETECT signal. This mode could be used for short preamble lengths in conjunction with the DP8462. The DP8462 Lock Detect output would be connected to the EXTERNAL PREAMBLE COMPLETE (observing polarities, see pin 28). The DP8462 will issue a "Lock Detect" after counting 16 code ones (3 to 4 Message Bytes depending upon which preamble pattern is used, 3T or 4T). The DP8463B must receive at least two 4T preamble patterns before an EXTERNAL PREAMBLE COMPLETE signal is received. The "active" level of this pin is determined by the state of pin 28, ADDRESS MODE 3.

Pin No	Description
21	**READ GATE.** An active high level input places the DP8463B in the read mode. During this mode: the source of the READ/REFERENCE CLOCK is switched to the VCO CLOCK/2, the address mark is searched for (if selected), the length of the preamble is counted as programmed, and the CODE IN signal is decoded and output as MESSAGE OUT at the appropriate time per the programming of the mode pins.
22	**WRITE GATE.** An active high level input places the DP8463B in the write mode. During this mode: the MESSAGE IN data is encoded and output as CODE OUT, and during the programmed time (when selected), the preamble and address marks appear at CODE OUT.
24	**UNSYNCHRONIZED CODE.** This is the encoded data output of the pulse detector (e.g., DP8464B). A flux transition on the disk produces an active high pulse for this pin. This input is used to detect the ESDI, SMD, and IBM address marks.
25	**ADDRESS MARK ENABLE (AME).** An active high level, while WRITE GATE is also active high, will write the address mark prescribed by the AM MODE pins. The AME must be high for the complete address mark. An active high level of AME while WRITE GATE and READ GATE are both low causes the 8463B to search for an Address Mark when in the ESDI Mode (AM2 & AM1 low). An active high level of AME when READ GATE is active high causes the 8463B to search for an AM when in the SMD Mode (AM2 low, AM1 high).
PROGRAMMING INPUTS	
1	**ADDRESS MARK MODE 1 (AM MODE 1).** Defined in Table I. A logic "1" is a high level.
2	**4T/$\overline{\text{3T}}$ PREAMBLE MODE.** A high level places the DP8463B in the mode to generate and detect "4T" preamble patterns (i.e., 1000 in code which is four time periods). In this mode, the MESSAGE IN input is inverted in the DP8463B before being encoded so that a 4T preamble can be generated from an all zeros MESSAGE IN data stream. The output of the decoder is also inverted in this mode so the double inversion is transparent to the user. The double inversion is always done in this mode, not just during the preamble. A low level input on this pin places the DP8463B in the mode to accept the "3T" pattern as the preamble (3T pattern is 100 in code). The Input/Output is not inverted in this mode. If the 3T preamble pattern is used with the IBM code, a non-zero three-bit repeating input pattern is required which

Pin Descriptions (Continued)

Pin No	Description
PROGRAMMING INPUTS (Continued)	
2 (Cont.)	is typically impossible for a disk data controller to generate. Therefore there is a special mode during which the disk data controller's input to MESSAGE IN is ignored by the encoder section and the DP8463B internally generates a 3T code output; as long as the MESSAGE IN input is all zeros. This 3T preamble starts after the initiation of WRITE GATE going active (high) and continues until the first "1" is seen at the MESSAGE IN pin. This mode of internally generating a 3T preamble with IBM code is activated by having high levels on the two PREAMBLE LENGTH 0 and PREAMBLE LENGTH 1 pins (16 & 17).
3	**NO CONNECTION.** This pin must be left open-circuited or tied to V_{CC}.
15	**WRITE CLOCK MODE.** A low level enables the automatic resynchronization circuitry for the IBM encoder. The WRITE CLOCK is resynchronized to the 2f C/S CLOCK and used for clocking the IBM encoder. The same 2f C/S is used to clock a flip-flop that provides the CODE OUT signal. A high level WRITE CLOCK MODE input allows the WRITE CLOCK to directly clock the IBM encoder. In this mode the CODE OUT signal comes directly from the encoder and does not get strobed out by the 2f C/S CLOCK.
16	**PREAMBLE LENGTH 0 (PL0).** This input and PREAMBLE LENGTH 1 (PL1) determine the length of preamble that is read before an active low LOCK DETECT signal is issued. These two pins also control the internal generation of a 3T preamble pattern for use with an all zeros MESSAGE IN input.

PL1	PL0	LENGTH OF PREAMBLE/ OTHER FUNCTION
0	0	Length Set by External Preamble Complete
0	1	6 Message Bytes
1	0	8 Message Bytes
1	1	6 Message Bytes/Generate 3T Preamble Internally

17	**PREAMBLE LENGTH 1 (PL1).** See pin 16.
19	**ADDRESS MODE 2.** Defined in Table I.
27	**ADDRESS MODE 0.** Defined in Table I.
28	**ADDRESS MODE 3.** Defined in Table I. The level of this pin also determines the active level polarity of the pin 20 EXTERNAL PREAMBLE COMPLETE input. This is possible since the Table 1 function is a Don't Care for all except one type of the N7V Address Mark. If ADDRESS MODE 3 is high or open-circuited, then pin 20 is active high. If ADDRESS MODE 3 is low, then pin 20 is active low.

Pin No	Description
OUTPUT SIGNALS	
4	**READ/REFERENCE CLOCK.** This is the clock that is provided to the disk data controller where it is typically labelled "read clock". It is, however, necessary for both reading and writing. The source of the clock is different during reading compared to writing. When READ GATE is active (high) the READ/REFERENCE CLOCK is the VCO CLOCK divided-by-two. The MESSAGE OUT data is read by the disk data controller using the positive going edge of READ/REFERENCE CLOCK. When READ GATE is inactive (low), the READ/REFERENCE clock is the 2f C/S CLOCK divided-by-two. This clock is used by the disk data controller as the timing source for its WRITE DATA and WRITE CLOCK outputs.
7	**MESSAGE OUT.** This is the "Read Data" input to the disk data controller. A high level represents a "one" of decoded (NRZ) data. It is read by the controller using the positive going edge of the READ/REFERENCE CLOCK. MESSAGE OUT is held at a low level when READ GATE is inactive (low) and other intervals specified in Table I.
8	**CODE OUT.** A high level for a 2f clock period is output for each "1" in code that is to be written on the disk as a flux transition by a write amplifier containing a write flip-flop that changes state every time a positive going pulse edge is received. Since (2,7) code always has at least two zeros between adjacent ones, this output is a Return-to-Zero (RZ) code. If the CODE OUT is to be sent to another register, instead of directly to the Write Amplifier, it can be clocked out by using the CODE OUT STROBE.
13	**CODE OUT STROBE.** The positive going edge of this signal should be used as the clock input to an external shift register for applications where the CODE OUT is transformed before being sent to the Write Amplifier, e.g., for precompensation of some code patterns on some tracks.
18	**$\overline{\text{LOCK DETECT}}$.** A low level signifies that a minimum, uninterrupted length of the programmed preamble pattern has been read. The length of the preamble read is programmed by PREAMBLE LENGTH 0 and 1. The LOCK DETECT level returns to a high when READ GATE goes inactive (low).
23	**ADDRESS MARK FOUND (AMF).** A high level appears when an address mark has been sensed and, depending upon programming, other conditions may also be required before the AMF goes active (high). Table I specifies the various conditions under which AMF becomes active and inactive. Also see the Address Mark section under "Description of Format and Circuit Characteristics".

TABLE I. Address Mark Modes

AM Mode	Program Inputs				AM Written	AM Read	Inputs		Output	
	AM 3	AM 2	AM 1	AM 0			Read Gate During AM Search	AME During AM Search	AMF Goes Active When:	AMF Returns Inactive When:
ESDI, Fully Compatible	X	0	0	0	ESDI	ESDI	Inactive (L)	Active (H)	AM Found & 1st Code "1" Following AM	AME Goes Inactive
ESDI, Noise Tolerant	X	0	0	1	ESDI	ESDI or IBM	Inactive (L)	Active (H)		
SMD, Fully Compatible	X	0	1	0	SMD	SMD	Active (H)	Active (H)		
SMD, Noise Tolerant	X	0	1	1	SMD	SMD or IBM	Active (H)	Active (H)		
Hard Sector	X	1	0	0	None (ESDI/SMD Signals)	None	Don't Care	Don't Care	NA	NA
ESDI/SMD (8466 Comp.)	X	1	0	1	ESDI = SMD	ESDI = SMD	Don't Care	Don't Care	AM Found & 1st Code "1" Following AM	2nd Code "1" Following AM is Read
IBM	X	1	1	0	IBM	IBM	Don't Care	Don't Care		
N7V-A (Note 1)	1	1	1	1	N7V (In Header & Data Segments)	N7V (In Header & Data Segments)	Active (H)	Don't Care	AM Found & 1st Message "1" Following AM	Read Gate Goes Inactive (L)
N7V-B	0	1	1	1	N7V (In Header Only)	N7V (In Header Only)	Active (H)	Active (H)		

X = Don't Care NA = Not Applicable

(Continued on next page)

TABLE I. Address Mark Modes (Continued)

AM Mode	Program Inputs AM 3	AM 2	AM 1	AM 0	Phase Sync (Byte) Requirements:	Byte Sync (Byte) Requirements	MESSAGE OUT = 0 Until: (& Phase Sync Mode Ends) (Note 2)
ESDI, Fully Compatible	X	0	0	0	1. With 4T Preamble: Function done by all zeros preamble. 2. With 3T Preamble: Recommend writing 10111100 which is read as 00000000.	1. With 4T Preamble: Must have one or more "1"s; with one in leading position, preferably. 2. With 3T Preamble: Must have one or more "1"s; Leading two bits should be zeros.	1. With 4T Preamble: After 6 or Programmed number of bytes of preamble are read. 2. With 3T Preamble: After two 4T patterns in phase sync byte are read.
ESDI, Noise Tolerant	X	0	0	1			
SMD, Fully Compatible	X	0	1	0			
SMD, Noise Tolerant	X	0	1	1			
Hard Sector	X	1	0	0			
ESDI/SMD = (8466 Comp.)	X	1	0	1			
IBM	X	1	1	0			
N7V-A (Note 1) N7V-B	1 0	1 1	1 1	1 1	1. With 4T Preamble: Write 8 zeros. 2. With 3T Preamble: Write 8 ones.	3. With 4T Preamble: Use 00000010 & 00000011. Same as 1 and 2 Above.	[N7V AM is Detected] $\underbrace{(\text{N7V AM Detected} \bullet \text{AME})}_{\text{For Header}} + \underbrace{\text{N7V AM Detected} \bullet \overline{(\text{AME})}}_{\text{For Data}}$ Segment (Note 3)

X = Don't Care NA = Not Applicable

Note 1. Use only for 8463A compatibility where there is an N7V AM in both header & data segments.

Note 2. After the time MESSAGE OUT = 0, MESSAGE OUT = Decoded CODE IN

Note 3. Data segment has no AM

TABLE II. Message Data/Code Tables

IBM (2,7,1,2,3) Message Data/Code Table

Normal Message Data		Inverted Message Data		Code	
MSB	LSB	MSB	LSB	MSB	LSB
000		111		000100	
10		01		0100	
010		101		100100	
0010		1101		00100100	
11		00		1000	
011		100		001000	
0011		1100		00001000	

Most Significant Bit (MSB) is read/written first.

Use the Normal Message Data column when 4T/$\overline{3T}$ PREAMBLE MODE (pin 2) is low.

Use the Inverted Message Data column when 4T/$\overline{3T}$ PREAMBLE MODE is high.

NOTE: The IBM (2,7,1,2,3) Code and some implementations of it are patented by IBM. National Semiconductor Corporation (NSC) has a license agreement with IBM enabling NSC to incorporate a particular implementation of the IBM (2,7) Endec in an integrated circuit for sale to others. Also see the Patent Indemnification section in NSC's Standard Terms and Conditions for Sales.

Note: Definition of (2,7,1,2,3):

2 = minimum number of zeros between adjacent code ones

7 = maximum number of zeros between adjacent code ones

1 = ratio of message data bits to code bits with first number (1)

2 = being the number of message bits

3 = number of different lengths of message data words

Description of Format and Circuit Characteristics

1. Address/Sector Marks

1A. Hard Sector Format—Sector Mark

In this format the sector mark signal from the disk drive is sent directly to the disk data controller and the DP8463B is not involved.

1B. Soft Sector Format—Address Mark (AM)

1B-1. ESDI Address Mark

This is a gap on the disk without any flux transitions for a length of three message bytes. The gap appears at the start of each sector. It is written by having the ADDRESS MARK ENABLE (AME) active (high) for 3 message bytes while WRITE GATE is also active (high). The AM is detected when an interval of 16 message bit times passes without a flux transition. Table I shows three different modes for using the ESDI AM. The "ESDI, FULLY COMPATIBLE" is the fully-compatible-with-ESDI specifications mode where AME is active (high) while READ GATE and WRITE GATE are inactive (low) when looking for an AM. When the AM is found, ADDRESS MARK FOUND (AMF) goes active (high), the disk data controller receives this and responds by having READ GATE go active (high) and AME go inactive (low). The AME going inactive will cause the 8463B's AMF to go inactive (low). A second mode "ESDI, NOISE TOLERANT" has the ability to accept a few noise bits in the AM gap and still output an AMF. (See Noise Tolerant ESDI/SMD AM section for full explanation). A third mode of ESDI AM has the state of the AME, during an AM search, as a "don't care" for compatibility with the DP8466 controller. In all these modes, the AMF appears only after both the AM is sensed and the first code "1" bit of the preamble is detected.

1B-2. SMD Address Mark

This AM is the same 3 message byte gap as ESDI. The difference is the state of READ GATE during the search for an AM. For SMD compatibility, READ GATE must be active (high) during the AM search. The three SMD modes shown in Table I are analogous to the three ESDI modes.

1B-3. IBM Address Mark:

This AM is a gap of 32 code bits which has no flux transitions except for two transitions in bit positions 8 and 20. The first 7 bit positions are "don't cares" per IBM's definition. The DP8463B writes a "1" (transition) in positions 3, 8 and 20. The DP8463B detects these AMs by detecting two gaps of 10 bits with a "1" between the two gaps. If the first gap is larger than 16 bits, the detector will reset and begin again.

1B-4. Noise Tolerant ESDI/SMD AM:

These modes will accept a perfect 2-message-byte ESDI or SMD gap or a gap which contains some noise bits. The ESDI/SMD AM gap detector is ORed with the IBM AM detector so an AMF will appear if either a 2-message-byte gap (32 code bits), or two 10-code-bit gaps with a "1" in between, is detected in the 3-message-byte (48 code bits) gap. The 16-bit limit on the first gap of the IBM AM detector is disabled.

1B-5. N7V Address Mark (N7V = Non 7 Violation):

This AM is a unique code word that does not violate the (2,7) RLL constraints but can not be generated by any message input to the IBM encoder. The N7V AM is a two-message-byte AM that must follow the preamble, since the decoder must be in phase sync to read the AM properly. The first byte of the AM consists of 4T phase sync patterns, the second byte is the unique N7V pattern. If the disk data controller randomly asserts READ GATE, the possibility of the N7V AM being detected in the write splice or in the data, before phase sync has been achieved, must be avoided. This can be done by not routing the CODE IN input to the 8463B until several bytes of the preamble have been detected externally. For example, this is done simply with the DP8462 by ANDing its SYNCHRONIZED DATA output with its inverted LOCK DETECT output, since the LOCK DETECT only appears after 3 or 4 bytes of the preamble have been read.

See Table I for more detail of Inputs/Outputs during AM reading and writing.

Description of Format and Circuit Characteristics (Continued)

2. Phase Sync Pattern

The decoder must be able to distinguish between the odd and even code bits to decode properly. This is impossible to do with the maximum frequency "3T" code pattern (100) in a preamble because the "1" alternates between odd and even positions. The "4T" code pattern (1000) must be used, along with the knowledge of what message pattern was used to generate it. A "3T" preamble can be used if it is followed by two "4T" patterns before data is read. See *Figure 1* formats for examples, and Table II for encode/decode definitions.

3. Byte Sync Pattern:

The purpose of the Byte Sync (or Sync Byte) is to define the message byte boundary for the disk data controller. The Byte Sync message byte should consist of one or more "1"s. With "4T" preambles, the Byte Sync should, preferably, have a "1" in the leading bit position. With "3T" preambles, the Byte Sync should have zeros in the two leading positions (to buffer it in time from the phase sync byte).

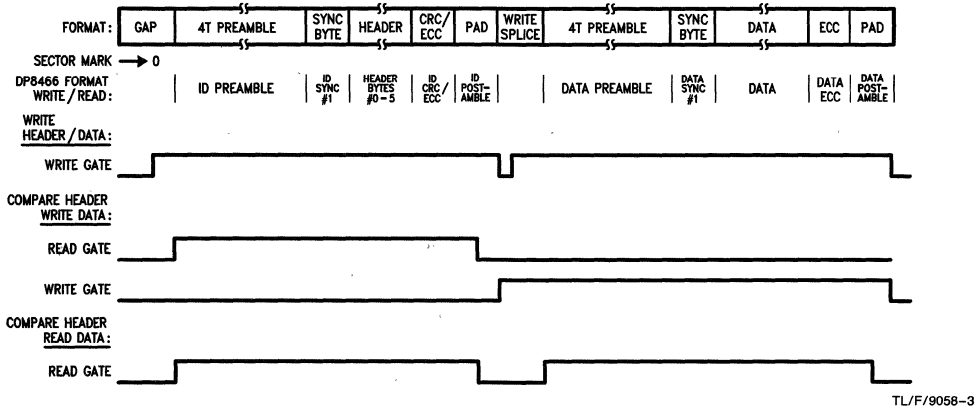
4. Error Propagation:

Since a single bit-shift error in a code word may be decoded as a different message word, there is error propagation. The longest error burst found for the IBM Code is 5 message bits. Therefore, the disk system must have Error Checking and Correcting (ECC) circuitry capable of correcting these errors.

5. Format Examples:

Figure 1 illustrates the sector formats and timing of various control signals for each of the Address Mark Modes and Preamble types for both reading and writing.

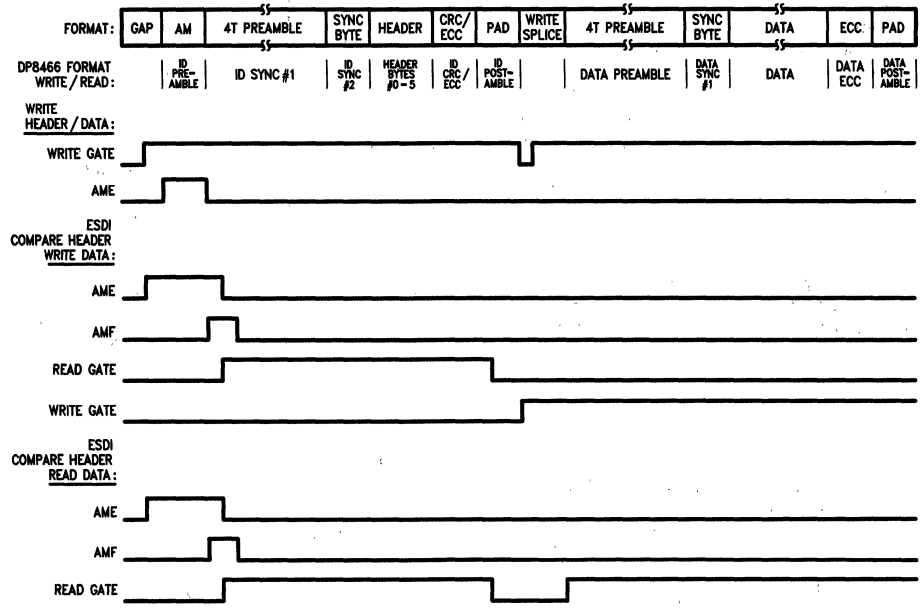
Formats



TL/F/9058–3

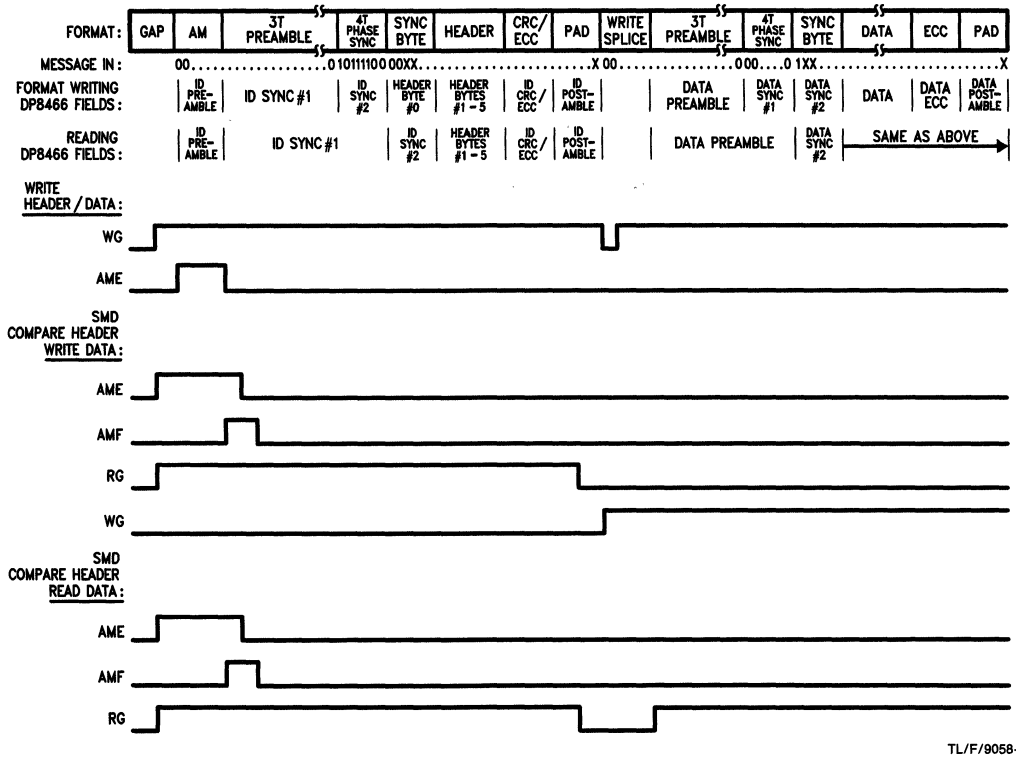
FIGURE 1-1. Hard Sector, 4T Preamble

Formats (Continued)



TL/F/9058–4

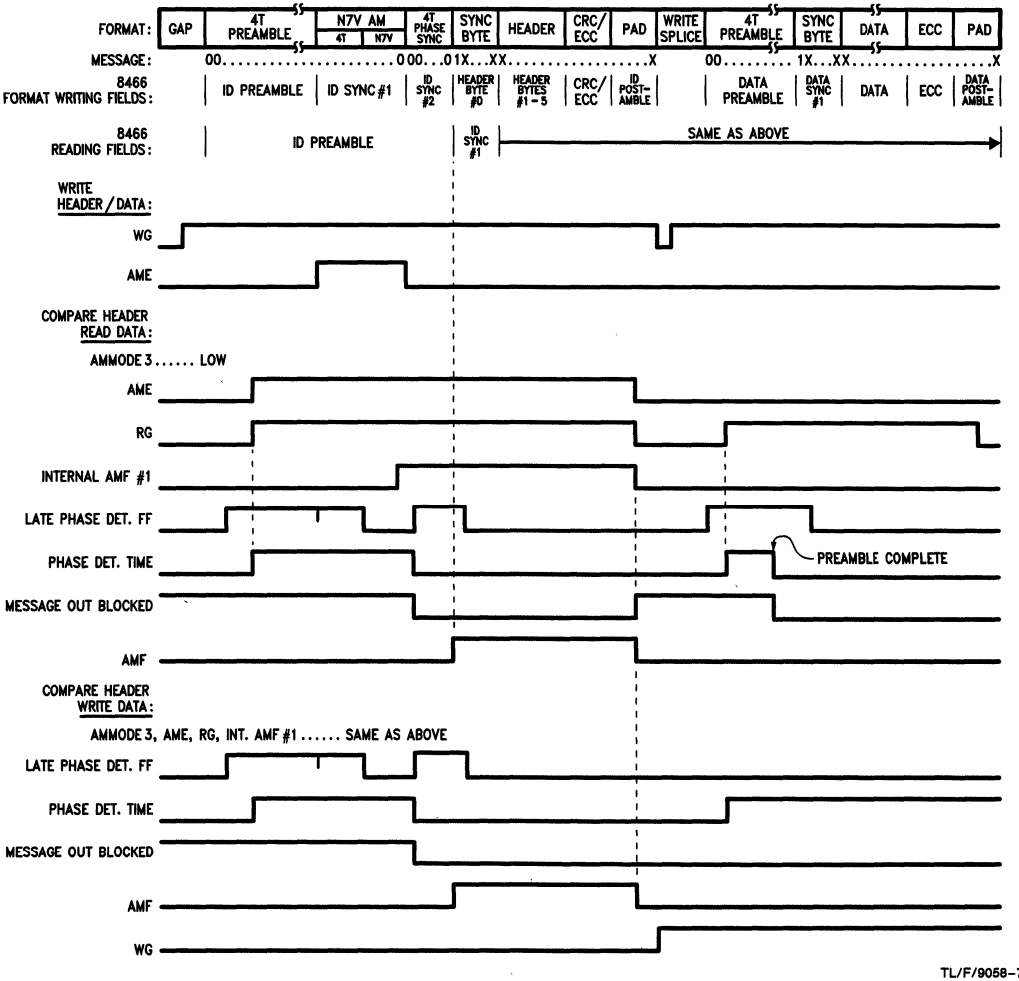
FIGURE 1-2. SMD, ESDI, IBM Address Mark, 4T Preamble



TL/F/9058–5

FIGURE 1-3I. ESDI, SMD, or IBM AM, 3T Preamble

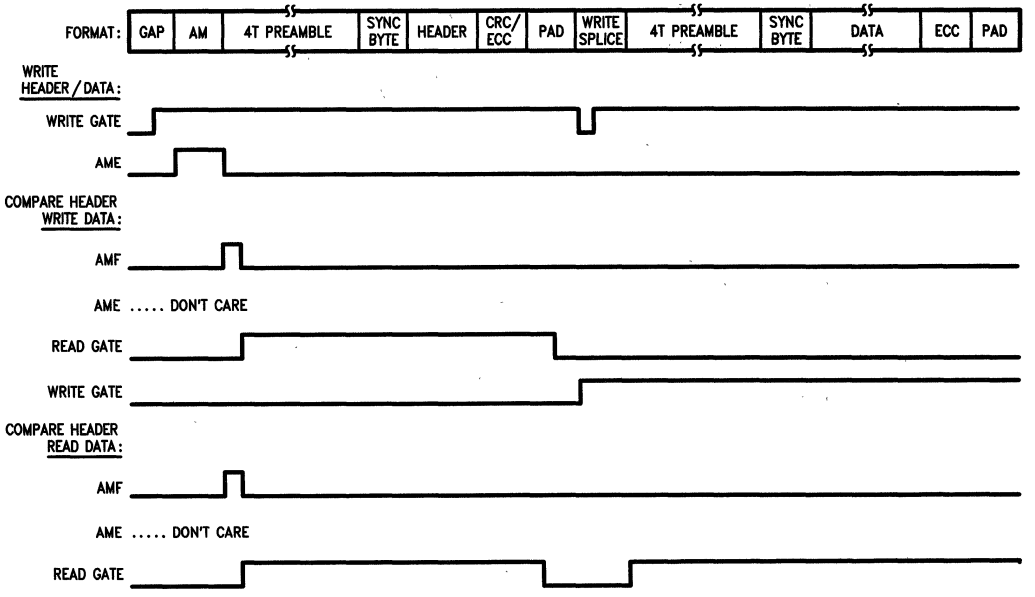
Formats (Continued)



TL/F/9058–7

FIGURE 1-4. 4T Preamble, N7V AM (N7V-B Mode)

Formats (Continued)

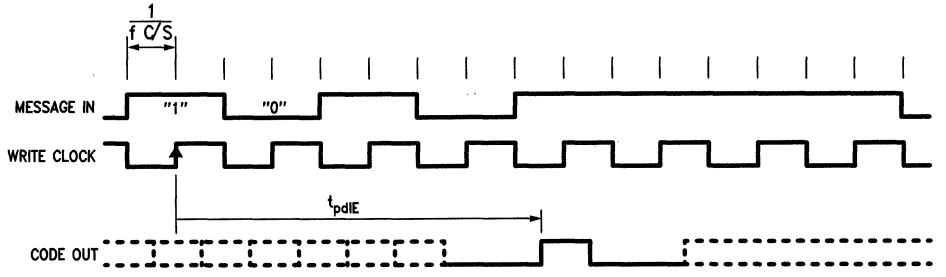


TL/F/9058–8

FIGURE 1-5. ESDI, SMD, or IBM AM; 4T Preamble (with Read Gate & AME as "don't cares" during search for AM)

Timing Waveforms

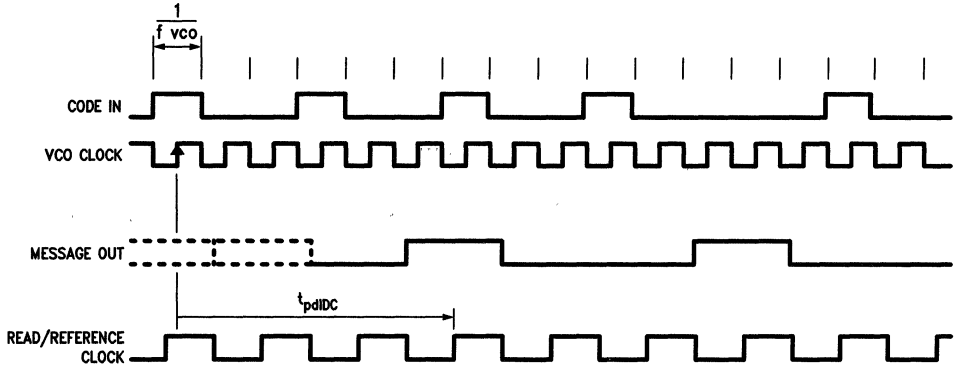
IBM Encoder



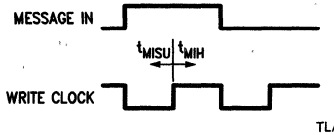
TL/F/9058–14

Timing Waveforms (Continued)

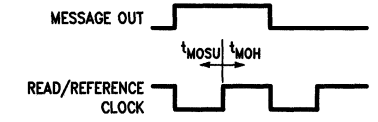
IBM Decoder



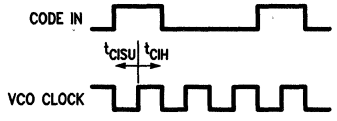
TL/F/9058–15



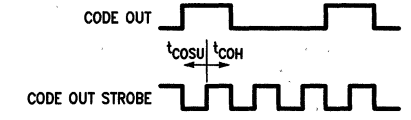
TL/F/9058–16



TL/F/9058–17



TL/F/9058–18

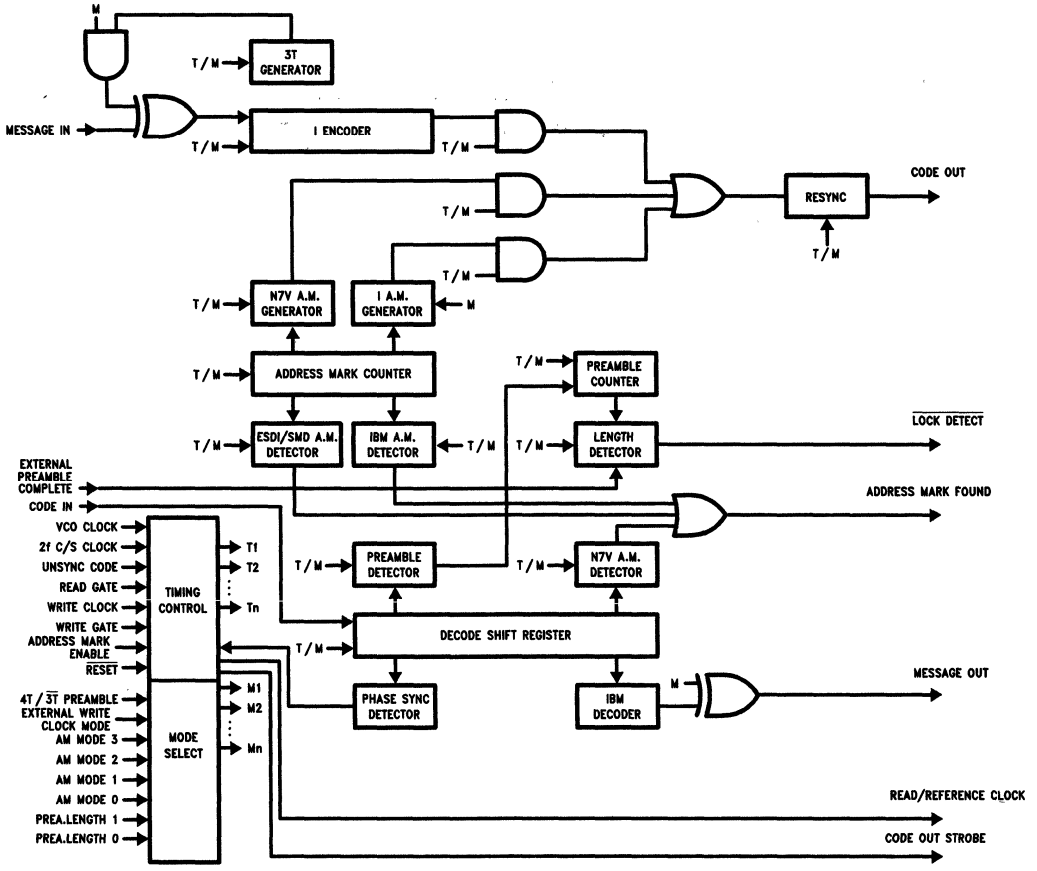


TL/F/9058–19

2

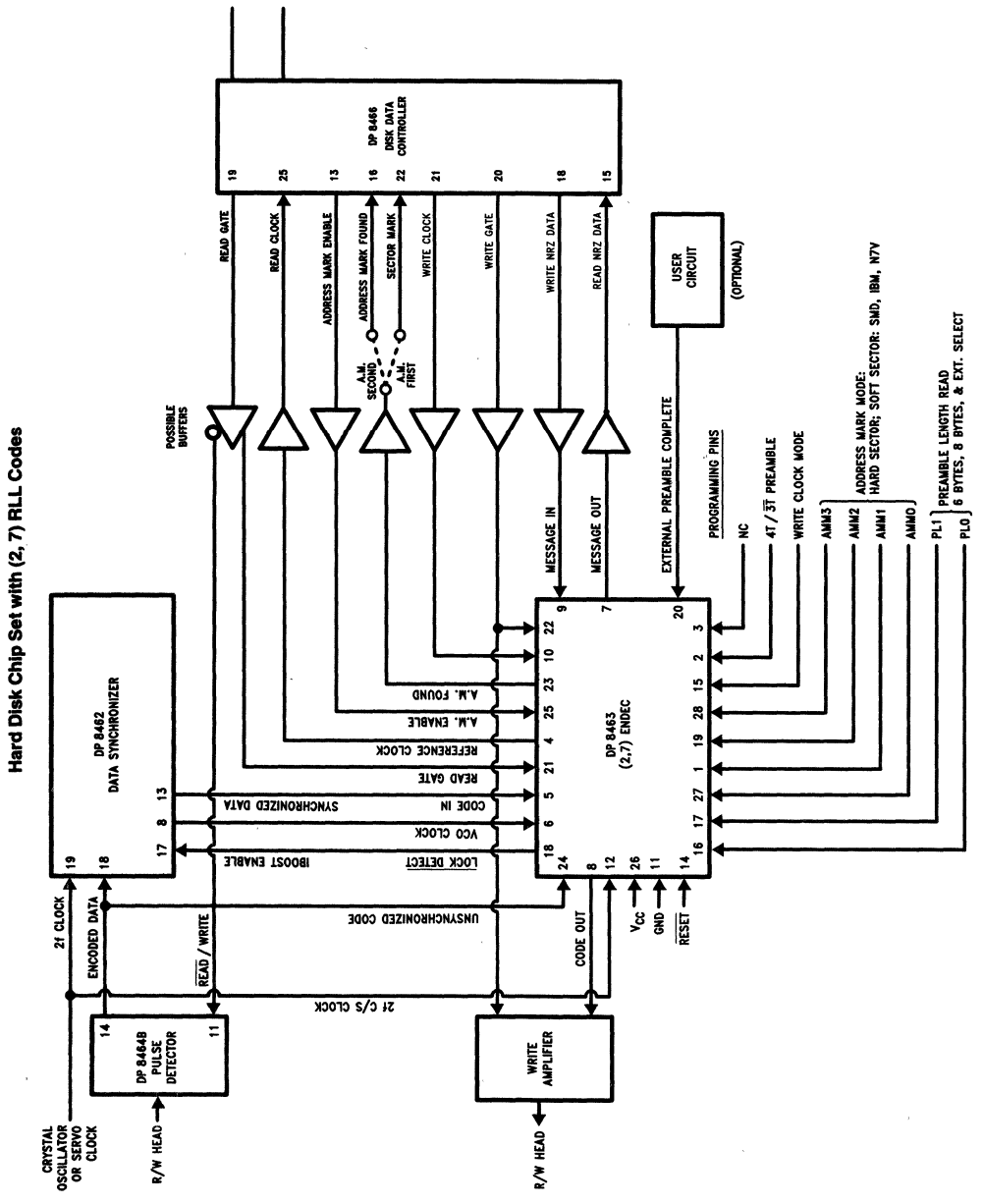
Typical Applications

DP8463B (2, 7) Endec Block Diagram



TL/F/9058–12

Typical Applications (Continued)



TL/F/9058–13

National Semiconductor

PRELIMINARY

DP8469 Synchronizer/2,7 Endec

General Description

The DP8469 data synchronizer/2,7 endec is intended for use in magnetic disk, optical disk, or tape drives during reading and writing operations. The device utilizes a fully integrated PLL to synchronize 2,7 serial code and convert data between one of several hard and soft sectored versions of 2,7 RLL (Run Length Limited) and serial NRZ code format. The DP8469 synchronizer/endec incorporates both the DP8459 synchronizer and the DP8463 2,7 code endec functions together in a 28-pin PCC package.

In the read mode, the device receives 2,7 RLL coded data from the drive's pulse detector, resynchronizes it, and then decodes the data to NRZ format for output to the controller.

In the write mode, the device receives NRZ data from the disk controller, encodes it in one of nine different 2,7 RLL hard/soft sectored formats, and then sends the data out to the drive with optional 3T precompensation adjustments.

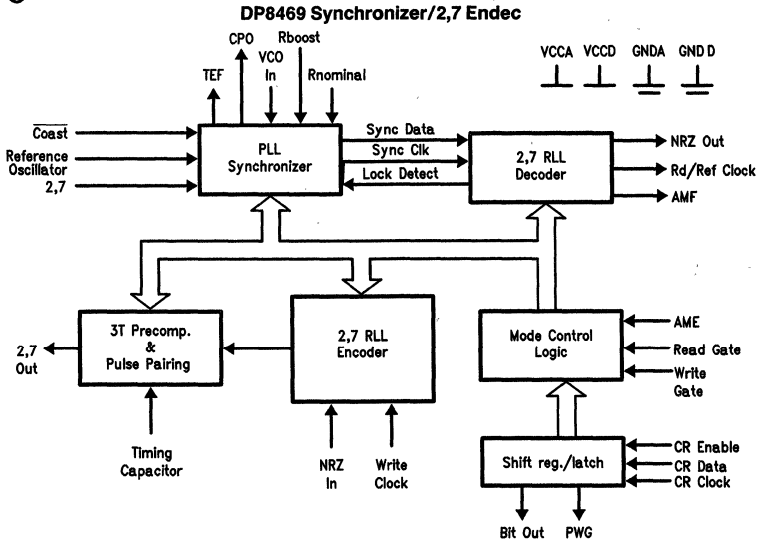
The device generates and recognizes the following 2,7 address mark formats; ESDI, ESDI noise tolerant, SMD, SMD noise tolerant, ST506(A), hard sector and three variations of ESDI, IBM, & ST506(B) optimized for the DP8466 controller. The address mark format is selected by 4 bits in a control register. A user defined variable-length preamble pattern can be used with any of the address mark modes. The pattern type, 3T or 4T, is set with one control register bit, and the preamble length is defined by the input NRZ data.

The synchronizer provides a dual gain phase locked loop which offers a high bandwidth mode for preamble lock acquisition and a low bandwidth mode for reading data. Two ports are provided for the PLL filter to enable use of higher order filter designs. The synchronizer has a Zero-Phase-Start feature which helps to minimize acquisition time in both read and write modes. A PHASE COMPARATOR TEST function is also provided for observation of PLL loop dynamics and determination of average media bit shift. The 2,7 OUTPUT pin provides the logical OR of the phase comparator's pump up and down outputs when programmed for Test Mode 1 operation. (Continued)

Features

- NRZ to 2,7 RZ RLL encoding/decoding
- 3T and 4T preamble generation/detection
- 1.5 Mbit/s to 24 Mbit/s data rates
- User specified preamble length
- ESDI, SMD, and ST506 soft sectoring
- Hard sectoring
- Fully integrated dual-gain PLL
- Zero-Phase-Start lock sequence
- Digitally controlled window strobe
- Digital write precompensation
- TTL compatible inputs and outputs
- +5V supply
- Packaging availability:
  — 28-pin Plastic Chip Carrier (PCC)
  — 40-pin TapePak

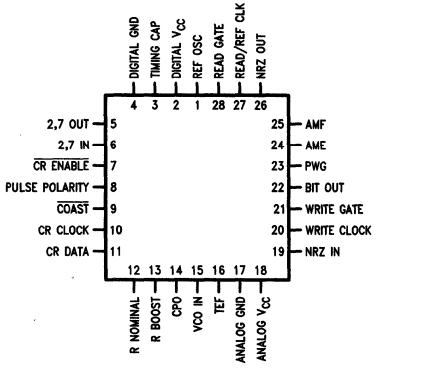
Block Diagram



TL/F/9386–2

Connection Diagrams

PCC

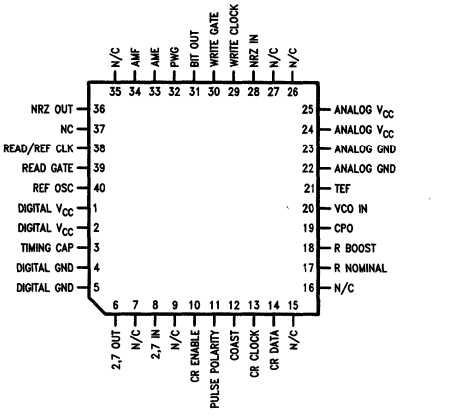


TL/F/9386–34

Top View

Order Number DP8469V
See NS Package Number V28A

TapePak®



TL/F/9386–1

Top View

Order Number DP8469TP
See NS Package Number TP40A

General Description (Continued)

A precise synchronization window is provided on chip using a self-aligned silicon delay line which remains accurate independent of temperature, power supply, external components and IC process variations. A strobe early/late function is provided which allows the synchronization window to be digitally adjusted to allow for error recovery or margin testing. The window can be shifted up to 20% in steps of 1.25% by 5 bits in a control register.

The synchronizer's data rate range is 1.5 Mbit/s to 24 Mbit/s. This range is divided into four operating regions each providing a 2 to 1 span in VCO frequency. Selection of one of the four data rate regions is controlled by two bits in a control register.

The READ/REFERENCE CLOCK provides both a read and write clock source for the controller. In read mode, once READ GATE has gone active and the Zero-Phase-Start sequence has been completed, the READ/REFERENCE CLOCK outputs the VCO divided by two. In the non-read mode, READ/REFERENCE CLOCK outputs REFERENCE OSCILLATOR divided by two. The circuitry incorporates a non-glitching multiplexer to ensure no erroneous clock pulses occur during the switch between input sources.

A digital precompensation feature is provided for write operations to compensate for bit shift due to data crowding on the media. The device will precompensate 3T code which is adjacent to greater than 3T code by selecting 1 of 6 bit shifted steps determined externally on the TC pin with an RC time constant set by the user.

The BIT-OUT output directly follows the logic level programmed in Control Register bit #10. By connecting BIT-OUT to the TC pin through a resistor, it is possible to have the controller select different pre-comp ranges via the control register.

Pin Descriptions

POWER SUPPLIES

ANALOG V_{CC}: Analog positive 5V supply, ±5%.

DIGITAL V_{CC}: Digital positive 5V supply, ±5%.

ANALOG GND: Analog negative supply pin.

DIGITAL GND: Digital negative supply pin.

INPUTS FROM CONTROLLER

NRZ IN (NRZI): NRZI input from the controller. Data is encoded and written to the disk in 2,7 format on the positive edge of WRITE CLOCK. NRZI is held LOW during the Preamble and Address fields, and transitioned HI at the start of data encoding.

Non-ST506 modes: NRZI must remain LOW throughout the address mark field and transition HI for a minimum of two NRZ bits to terminate preamble and start the controller sync byte.

ST506 modes: NRZI must remain LOW through both the address mark and preamble fields, and then transition HI for a minimum of one NRZ bit to start the controller sync byte.

WRITE CLOCK: Clock input from the controller synchronized with the NRZ IN data.

WRITE GATE (WG): A mode control input from the controller which allows the writing of header and data to the disk when active HI and prohibits writing of header or data when LOW.

ADDRESS MARK ENABLE (AME): AME must be held HI while writing an Address Mark. During an ESDI or SMD read operation, the AME pin must be held HI to search for an address mark. Termination of the AME HI level will reset AMF to the LOW state. The AME logic is not relevant during non ESDI and SMD read operations.

Pin Descriptions (Continued)

INPUTS FROM CONTROLLER (Continued)

READ GATE (RG): A mode control input from the disk controller. In ESDI and IBM modes, RG should not be transitioned HI until AMF is active HI. In SMD and ST506 modes, RG is qualified to the synchronizer internally as follows:

ST506: device must recognize 4 consecutive 3T or 4T preamble patterns.

SMD: AMF goes active HI.

In Hard Sector mode, RG should not be transitioned HI until the Index Sector Gap is found. After RG goes HI, the synchronizer locks to the 2,7 INPUT data rate using a Zero Phase Start frequency lock routine. When RG goes LOW, the synchronizer locks to the Reference Oscillator (REG OSC) using a Zero Phase Start frequency lock routine. RG timing is allowed to be fully asynchronous.

OUTPUTS TO CONTROLLER

ADDRESS MARK FOUND (AMF): An active HI output for the controller to indicate the first 2,7 pulse beyond a valid address mark has been found. In the ESDI and SMD modes, AMF remains HI until AME transitions LOW. In the ST506A mode, AMF stays HI until RG is deasserted. In the DP8466 modes, ESDI, IBM and ST506B, AMF returns LOW after the second 2,7 pulse is encountered.

NRZ OUTPUT (NRZO): Decoded output data for the controller that is strobed on the positive transition of READ REF CLK. Control register bit 11 selects either TRI-STATE® or totem pole output. Bit 11 = LOW sets NRZO to active totem pole output, and Bit 11 = HI sets NRZO to TRI-STATE output.

READ REF CLK (R/RCLK): This supplies the controller clock source. In read mode, after the Zero Phase Start sequence is completed, R/RCLK issues the VCO divided by two signal. The NRZO is synchronized with this clock. In the non-read mode, R/RCLK will issue the REF CLK input signal divided by two.

INPUTS/OUTPUTS TO DRIVE

2,7 OUTPUT: Output 2,7 Return-to-Zero (RZ) data for recording onto the storage media. Each positive edge represents a single recorded code bit. The 2,7 OUTPUT active transition edge can be shifted by specific controllable time steps by setting control register bits 7,6,5. Only certain minimum 3T patterns are affected.

2,7 INPUT: Incoming data derived from the storage media, issued from a pulse detector circuit. Each positive edge represents a single recorded bit.

PULSE POLARITY (PP): Input derived from DP8464/8 pulse detector's channel polarity output telling the pulse pairing circuitry which pulse to shift.

EXTERNAL SOURCES

REFERENCE OSCILLATOR (REF OSC): A reference frequency input *required* for DP8469 operation. The signal must be crystal or servo derived (accurate and highly stable), and at a frequency approximately equal to the 2,7 code rate (i.e., twice the NRZ data rate).

$\overline{\text{CR ENABLE}}$ ($\overline{\text{CRE}}$): When active LOW, $\overline{\text{CRE}}$ permits the loading of mode information via CRD and CRC. Data is latched into the part when $\overline{\text{CRE}}$ is transitioned HI. This input is also used to set test mode conditions as described in the Test Mode Operation section.

$\overline{\text{COAST}}$ ($\overline{\text{CST}}$): The control input for a coast function which may be activated when RG is either HI or LOW. When the $\overline{\text{CST}}$ input is LOW, the phase comparator is disabled and held in a cleared state, allowing the VCO to coast regardless of the 2,7 code input or reference oscillator activity. No other circuit functions are disturbed. When the $\overline{\text{CST}}$ is inactive HI, the synchronizer operates normally.

CR CLOCK (CRC): Positive edge triggered clock for the 24-bit control register. This input is also used to set test mode conditions as described in the Test Mode Operation section.

CR DATA (CRD): Data input for the 24 bit control register that selects the strobe window, precompensation and pulse pairing bit shift, data rate range, and address mark modes. This input is also used to set test mode conditions as described in the Test Mode Operation section.

ANALOG SIGNAL PINS

RNOMINAL (R_{NOM}): A resistor is tied from this pin to V_{CC} to set the *nominal* operating current. The current is internally multipled by 2 for charge pump use. This pin is also used to load the test modes as described in the Test Mode Operating section.

RBOOST (R_{BST}): A resistor is tied from this pin to V_{CC} to set the chrage pump *boost* (or adder) current, which is multiplied by 2 internally for use by the charge pump. The R_{BST} resistor is electrically paralleled with the R_{NOM} resistor until either RG is passed to the synchronizer, or preamble lock is acquired. This selection is made with control bit 18;

Bit 18 = HI: forces synchronizer switch to Low Gain on assertion of RG.

Bit 18 = LOW: forces synchronizer switch to Low Gain when preamble lock is acquired.

RBOOST (R_{BST}) (Continued): ST506 operating modes are special in that the switch to Low Gain is forced on the assertion of RG regardless of the state of bit 18. If no boost current is desired, a high value resistor must be tied to this pin to ensure its level is not allowed to drop below V_{IH} = 2V and activate the production test mode circuitry. This pin is also used for the test modes as described in the Test Mode Operating section.

CHARGE PUMP OUT (CPO): The output of the high-speed bi-directional current source switching circuitry of the charge pump. The external, passive PLL filter network is established between this pin, the VCO Input and ground.

VCO IN (VCOI): The high-impedance control voltage input to the voltage controlled oscillator (VCO). The external, passive PLL filter network is established between this pin, CPO pin and ground.

Pin Descriptions (Continued)

ANALOG SIGNAL PINS (Continued)

TIMING EXTRACTOR FILTER (TEF): Connection pin for external, passive components employed to stabilize the delay line timing extraction circuitry.

Note: The delay line accuracy *is not* a function of external component values or tolerances.

OTHER OUTPUTS

RC ADJ: An open collector Bipolar output which can be used to adjust the precompensation or pulse pairing external RC time constants. By connecting a resistor between the RC ADJ pin and the TC pin, a different RC timing constant can be used between precompensation and pulse pairing. RC ADJ is the logical true or complement of WG depending on the state of register bit 8:

Bit 8 = HI: RC ADJ is inverted from WG.

Bit 8 = LOW: RC ADJ follows WG.

BIT OUT: An undedicated open collector Bipolar output whose state is set using bit 10 in the control register. This output can be used for any purpose, including adding extra RC ranges to the precompensation and pulse pairing circuitry. Window strobe adjustments can be performed by connecting a resistor between BIT OUT and the CPO output pin.

Circuit Operation

CONTROL REGISTER OPERATION

The DP8469 is initialized by loading the desired mode selections, such as address mark format and 3T/4T Preamble pattern, via the CR DATA (CRD), CR CLOCK (CRC) and CR ENABLE ($\overline{CRE}$) inputs. Loading is accomplished by taking $\overline{CRE}$ active LOW, and clocking in the mode selection data on the positive going edge of CRC. The modes are latched in when $\overline{CRE}$ is transitioned HI. The selections are indicated in the 24-Bit Control Register section. The test modes are also loaded using CRD, $\overline{CRE}$, CRC as explained in the special Test Mode Operation section.

SYNCHRONIZER OPERATION

In non-read mode, the DP8469 PLL is locked to the REFERENCE OSCILLATOR signal (REF OSC). This permits the VCO to remain at a frequency very close to the media bitrate while the PLL is "idling" and thus will minimize the frequency step and associated lock time requirement encountered at the initiation of lock to 2,7 INPUT data. When READ GATE is transitioned LOW to terminate the Read mode, a Zero-Phase-Start and frequency acquisition sequence is employed to insure lock. The REF OSC signal is also used during this time to set the time delay of the internal delay line. Note that this requires the REF OSC signal to be present *at all times* at a stable and accurate frequency for proper DP8469 operation.

In non-ST506 modes of operation, after the assertion of READ GATE (fully asynchronous, with no timing requirements), and following the completion of two subsequent VCO cycles, the DP8469 VCO is stopped momentarily. The VCO is then restarted in accurate phase alignment with the second data bit which arrives subsequent to the VCO pause. This minimization of phase misalignment between the 2,7 READ DATA and the VCO (referred to as Zero-Phase-Start or ZPS) significantly reduces the data lock acquisition time.

The DP8469 incorporates a preamble-specific acquisition feature which is employed for all non-ST506 modes of operation where READ GATE is asserted only within a preamble. In these modes, after READ GATE is asserted HI, the device will be forced to lock to the exact 3T or 4T selected preamble frequency. The frequency discriminating action of the PLL provided in these modes produces a lock-in range equivalent to the available VCO operating range and thus eliminates the possibility of fractional-harmonic lock. Windowing (pulse-gate action; see Pulse Gate section) is prevented. (Application Note AN-414 has an explanation of typical false lock modes.)

In the ST506 modes of operation, at the assertion of READ GATE, the 2,7 IN data pattern is first sampled asynchronously and the synchronizer ZPS lock-on sequence is prevented until eight preamble patterns are recognized. The synchronizer only operates Low Bandwidth Gain, and in a phase lock mode (pulse gate action) during the ST506 operation.

In the non-ST506 modes, the user is provided, the option of an elevated PLL bandwidth during preamble acquisition for an extended capture range. An R_{BOOST} pin is provided to allow for an increase in charge pump gain over and above the level set by the $R_{NOMINAL}$ pin. The net current through either $R_{NOMINAL}$ or $R_{BOOST}//R_{NOMINAL}$ is multipled internally by 2 for use by the charge pump. The user should connect a high value resistor to R_{BOOST} if an elevated PLL bandwidth is not desired to ensure the pin does not fall below the 2V test inactive HI logic threshold.

The READ/REFERENCE CLOCK (R/R CLK) issues a waveform derived from the REF OSC input during the non-read mode (i.e., READ GATE inactive). In the read mode, following the assertion of READ GATE, the completion of the ZPS sequence, and in the case of ST506 modes, the recognition of a short 3T or 4T preamble pattern, R/R CLK issues a waveform derived from the VCO signal. Once data lock is achieved in the read mode and the first bit of the controller Byte Sync field is encountered, the NRZ OUT and R/R CLK outputs are held in a fixed, specified timing relationship. The R/R CLK output switches between input sources without glitches.

The DP8469 provides a $\overline{COAST}$ control input which serves to clear the phase comparator and disable charge pump action whenever taken to an active, logical-zero level. This function is made available to allow the PLL to be set to free-

Circuit Operation (Continued)

run, undisturbed, while a detectable defect is being read from the media in a region where re-initiation of the lock procedure is impractical (e.g., data field). External data controller circuitry is responsible for the detection of the defect and issuance of the $\overline{\text{COAST}}$ command. For a more detailed explanation of the synchronizer, and loop filter design, please see the DP8459 datasheet.

ENDEC OPERATION

2,7 Encoder: The data encoding path is responsible for generating an address field, preamble field, and the data field. This allows synchronous generation of preamble and data in all cases, insuring the fastest possible PLL lock during readback. The encoder has two modes of operation, ST506 and non-ST506 mode. The primary difference between these two modes is the placement of the fields:

Preamble Field	Address Field	Data Field

ST-506 Mode (Synchronous Type Address Mark)

Address Field	Preamble Field	Data Field

Non ST-506 Mode (Gap Type Address Mark)

In the ST506 modes, the controller issues a WRITE GATE (WG) command to begin the preamble generation. During the Preamble and Address Mark fields, NRZ IN is blanked. Preamble length is user defined since the ENDEC will continue to generate preamble until Address Mark Enable (AME) is asserted. Assertion of AME terminates the preamble, generates a Phase Sync (for decoding), and begins the N7V address mark. In both ST506 modes, the AME input *must be held high for exactly one Byte* (8 bits). The N7V address mark is one byte in length, does not violate the 2,7 code rules, and cannot be generated by the encoder. Immediately following the completion of the Address Mark, NRZ IN is unblanked, and encoded as data. This data must be the controller's sync byte. If the controller requires more then one byte of AME, then both AME and NRZ IN *must be held LOW* during the second byte to insure that during readback the first non-zero data presented to the controller will be a valid Sync Byte. The Sync Byte must begin with a leading "1", however the user has total freedom of the last 7 bits (1------). The encoder continues to encode the data according to the 2,7 code rules until WG is deasserted.

In non-ST506 modes, the controller issues WG to begin the header generation. Normally AME is also issued at the same time. If there is a delay between the assertion of WG and AME, the encoder will begin generating a preamble pattern. Assertion of AME generates the address gap. ESDI and SMD address marks require 3 bytes of AME while the IBM mark only requires 2 bytes of AME. Immediately upon the deassertion of AME, the encoder will begin generating the preamble field. NRZ IN is blanked while AME is asserted. The first non-zero NRZ data following the deassertion of AME *must be the controller's Sync Byte* as any NRZ IN data presented to the encoder after deassertion of AME terminates the preamble, and is encoded. The Sync Byte restrictions are as follows:

3T: 1-------

4T: 11------

For 3T preamble, the Customer has total freedom of the last 7 bits of the Sync Byte, however in the 4T preamble, the Customer has total freedom of only the last 6 bits.

NRZ IN data is internally blanked while AME is asserted as the DP8469 generates its own address mark data depending upon the type of address mark selected in control register bits 15 to 12. In the ST506 case, NRZ data presented to the DP8469 *immediately after* deassertion of AME *will* be encoded as data, while in the non-ST506 modes, the encoder will return to preamble generation *until the first non-zero* NRZ data following deassertion of AME. The preamble pattern is selected in control register bit 9; HIGH for 3T and LOW for 4T. Once the pattern is selected, it is transparent to the user.

2,7 Decoder: The data readback path is responsible for detecting the address field, preamble field, and finally the data field. The controller issues an active READ GATE (RG) to initiate clock synchronization and 2,7 decoding.

In ST506 mode, the controller first issues RG for the detection of the preamble field. The DP8469 employs a two phase preamble prequalifier on the internal RG. The prequalifier must first find several bytes of valid preamble pattern before it passes RG to the synchronizer and enables the standard preamble detector. In the event there is enough valid looking data to qualify as a preamble, the synchronizer is switched into phase-only low gain mode on the assertion of RG to guarantee that it can pass through any subsequent Write Splice without problems. The synchronizer employs a Zero Phase Start circuit to minimize its lock time.

Once preamble has been located (thirty-one 3T patterns or fifteen 4T patterns) the decoder begins searching for the N7V address mark. Phase-sync and address mark detection are accomplished at the same time, and the decoder then unblanks the NRZ OUT at the start of Controller's Sync Byte.

The non-ST506 modes (commonly called "Gap Type") breakdown into two groups:

a) ESDI and IBM modes where the controller does not assert RG until after AMF signifies the start of the Preamble.

b) SMD mode where the controller asserts RG prior to AMF, and the ENDEC is responsible for blocking RG to the synchronizer until the Preamble field has been found.

In both cases, the address mark precedes the preamble field. The controller instructs the DP8469 to search for the address mark, a 2 NRZ byte gap, transitionless ESDI and SMD gaps and an IBM gap with 3 specific transitions. Once the gap has been located, the AMF output signals the controller, and the controller responds with assertion of RG in ESDI and IBM. In the SMD mode, RG is internally qualified with AMF, providing the same function.

Once RG has been passed to the synchronizer, the same thirty-one 3T or fifteen 4T preamble pattern requirements must be met before the decoder can look for the Phase Sync, and unblank. The first non-zero NRZ OUT data is the controller's Sync Byte.

Circuit Operation (Continued)

Both the encoder and decoder follow the encoding rules as shown in the table below:

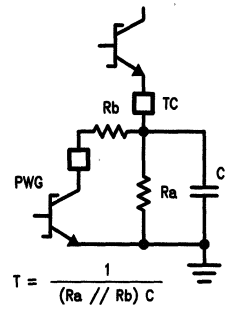
IBM 2,7 Conversion Table

NRZ				2,7 Code						
	0	0	0			0	0	0	1	0	0
		1	0					0	1	0	0
	0	1	0			1	0	0	1	0	0
0	0	1	0	0	0	1	0	0	1	0	0
		1	1					1	0	0	0
	0	1	1			0	0	1	0	0	0
0	0	1	1	0	0	0	0	1	0	0	0

2 = Minimum # of Zeros between adjacent Ones
7 = Maximum # of Zeros between adjacent Ones
1 = NRZ bits for relative conversion ratio
2 = Code bits for relative conversion
3 = # of different length code words

Precompensation

The Precomp circuitry uses the same basic delay-line technique to provide controllable time-delay steps. The user applies the necessary external RC components to the TC pin to provide the desired time step sizes. By using the BIT OUT and PWR outputs, different RC values can be used for different circumstances ... Programming Bit 8 = LOW sets the PWG output to follow WRITE GATE, while programming Bit 8 = HI sets PWG to the logical inverse of WRITE GATE. This allows complete freedom in setting the precomp time constants independently for different Write channel anomolies such as inner vs outer track recording densities.



TL/F/9386–14

Bit-crowding on the disk results in a maximum frequency pattern (3T) timing error to the data written on the disk. Whenever the 3T pattern is adjacent to a non-3T pattern, the two 3T bits are pushed apart, resulting in a time skew during readback. The precompensation feature allows the user to adjust the 3T pattern on a cycle-by-cycle basis to correct for this problem.

Precompensation Bit Patterns

Bit t−3	Bit t−2	Bit t−1	Target Bit t	Bit t+1	Bit t+2	Bit t+3
0	0	0	No Shift	0	0	0
1	0	0	No Shift	0	0	1
1	0	0	Shift Early	0	0	0
0	0	0	Shift Late	0	0	1

Test Mode Operation

The DP8469 provides 7 special test modes. With the exception of the Phase Comparator Test (PCT), SYNC DATA and SYNC CLK, these special test modes are used for production testing. The PCT data is the logical OR'ing between charge Pump Up and charge Pump Down, and can be used to examine the locking action of the PLL. The SYNC DATA and SYNC CLK outputs allow window strobe measurements. All together, they will allow the user to fine tune the application to insure minimum jitter during readback. The PCT function is present on the AMF output in test mode 1 (load 001, MSB first). SYNC DATA and SYNC CLK are present in test mode 7 (load 111, MSB first) on the READ/REF CLK and OUTPUT 2,7 outputs respectively.

The test modes are set by a 3 bit test register that is accessed via the R_{BOOST} and $R_{NOMINAL}$ inputs. In an application, both R_{BOOST} and $R_{NOMINAL}$ are held above the 2V threshold clearing the test register to mode 0, and allowing both HI and LOW gain PLL action. This insures that the device will always power-up in the normal operating mode. By pulling R_{BOOST} below the 2V threshold, the test register can now accept 3 bits of data that will decode to one of seven test modes (the eighth mode, 000, is normal operation). The $R_{NOMINAL}$ input is used to select between loading the 24 bit control register and the 3 bit test register; $R_{NOMINAL}$ = HI sends the data to the 24 bit register while $R_{NOMINAL}$ = LOW sends the data to the 3 bit test register. The test register is then loaded using $\overline{CRE}$, CRC, and CRD. After loading the test register, first return $\overline{CRE}$ = HI *to protect the data in both the Control Register, as well as the data in the Test Register.* It is required to set $\overline{CRE}$ = HI *before changing the state of any other control input* (including both $R_{NOMINAL}$ and R_{BOOST}) in order to prevent any internal switching from generating false clocks and changing the state of any register data. Next, return $R_{NOMINAL}$ HI to allow normal synchronizer operation. Maintain R_{BOOST} LOW to preserve the test mode. This combination will only allow low gain PLL action during testing. CRD and CRC must also be returned LOW, and held there during testing as they are used as special testing inputs.

The action of pulling $R_{NOMINAL}$ low disables the charge pump, and drives the VCO to the lower clamp limit. For any test where the PLL loop needs to run, such as window strobe tests, the test mode must be loaded before the loop is locked. First, load the test register and return $\overline{CRE}$ HI, and then $R_{NOMINAL}$ HI. Then reload the control register, and return $\overline{CRE}$ HI. Finally, exercise the device starting with RG deasserted, and running the device through the complete Read cycle. Note that for the device to remain in test mode, R_{BOOST} must be held low, and this eliminates the high gain mode of operation for the synchronizer. To observe the synchronizer in a high gain mode, replace $R_{NOMINAL}$ with a resistor equivalent to the parallel combination of $R_{NOMINAL}$ and $R_{NOMINAL}$. Do not exceed the 1 mA maximum total input current specification for the $R_{NOMINAL}$ pin.

DP8469 24 Bit Control Register

Control Register is Loaded MSB (Bit 23) First, and LSB (Bit 0) Last.

23 MSB	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0 LSB
Window Strobe					Gain Control	VCO Frequency Range Select		Address Mark Mode Select				NRZ Hi-Z Enable	Bit Out Hi	3T Preamble	RC Adjust	Precomp Step Select			Invert 2,7 Out	Edge Polarity	Pulse Pairing Step Select		

Pulse Pairing Delay Steps

3*	Pulse Pairing Early/Late 2	1	0	Delay Step % Bit Shift
X	0	0	0	Bypassed
X	0	0	1	0.0
0 / 1	0	1	0	± 2.5*
0 / 1	0	1	1	± 5.0*
0 / 1	1	0	0	± 7.5*
0 / 1	1	0	1	±10.0*
0 / 1	1	1	0	±12.5*
0 / 1	1	1	1	±15.0*

*Bit 3 selects which flux change from the Pulse Detector is Pulse Pairing Corrected.

Control Register Bit Definitions for Bits 4, 8, 9, 10, 11, 18

Bit 4	2,7 Output	0	True
		1	Inverted
Bit 8	RC ADJ Follows	0	$\overline{\text{WRT GATE}}$
		1	WRT GATE
Bit 9	3T/4T Preamble	0	4T
		1	3T
Bit 10	BIT OUT	0	Active LOW
		1	Inactive HI
Bit 11	NRZ OUT	0	Totem Pole
		1	TRI-STATE
Bit 12	PLL Switch to Low Gain	0	READ GATE
		1	LOCK DETECT

Precomp Delay Steps

Precomp Early/Late 2	1	0	Delay Step % Bit Shift
0	0	0	Bypassed
0	0	1	0.0
0	1	0	± 2.5*
0	1	1	± 5.0*
1	0	0	± 7.5*
1	0	1	±10.0*
1	1	0	±12.5*
1	1	1	±15.0*

Control Register # 15	14	13	12	Address Mark Mode
0	0	0	0	ESDI
0	0	0	1	SMD
0	0	1	0	ESDI*
0	0	1	1	N/A
0	1	0	0	ST506A
0	1	0	1	ST506B
0	1	1	0	HARD
0	1	1	1	IBM*
1	0	0	0	ESDI NT
1	0	0	0	SMD NT

*DP8466 Compatible

VCO Freq Range

Bits 15	14	Freq Range (Mbits/sec)
0	0	1.5–3
0	1	3–6
1	0	6–12
1	1	12–24

Control Register # 23	22	21	20	19	Strobe Word	Typical Window Strobe
0	1	1	1	1	−15	$-0.270\times T_{vco}$
0	1	1	1	0	−14	$-0.252\times T_{vco}$
0	1	1	0	1	−13	$-0.234\times T_{vco}$
0	1	1	0	0	−12	$-0.216\times T_{vco}$
0	1	0	1	1	−11	$-0.198\times T_{vco}$
0	1	0	1	0	−10	$-0.180\times T_{vco}$
0	1	0	0	1	−9	$-0.162\times T_{vco}$
0	1	0	0	0	−8	$-0.144\times T_{vco}$
0	0	1	1	1	−7	$-0.126\times T_{vco}$
0	0	1	1	0	−6	$-0.108\times T_{vco}$
0	0	1	0	1	−5	$-0.090\times T_{vco}$
0	0	1	0	0	−4	$-0.072\times T_{vco}$
0	0	0	1	1	−3	$-0.054\times T_{vco}$
0	0	0	1	0	−2	$-0.036\times T_{vco}$
0	0	0	0	1	−1	$-0.018\times T_{vco}$
0	0	0	0	0	0	0
1	0	0	0	0	0	0
1	0	0	0	1	1	$0.018\times T_{vco}$
1	0	0	1	0	2	$0.036\times T_{vco}$
1	0	0	1	1	3	$0.054\times T_{vco}$
1	0	1	0	0	4	$0.072\times T_{vco}$
1	0	1	0	1	5	$0.090\times T_{vco}$
1	0	1	1	0	6	$0.108\times T_{vco}$
1	0	1	1	1	7	$0.126\times T_{vco}$
1	1	0	0	0	8	$0.144\times T_{vco}$
1	1	0	0	1	9	$0.162\times T_{vco}$
1	1	0	1	0	10	$0.180\times T_{vco}$
1	1	0	1	1	11	$0.198\times T_{vco}$
1	1	1	0	0	12	$0.216\times T_{vco}$
1	1	1	0	1	13	$0.234\times T_{vco}$
1	1	1	1	0	14	$0.252\times T_{vco}$
1	1	1	1	1	15	$0.270\times T_{vco}$

Absolute Maximum Ratings

If Military/Aerospace specified devices are required, please contact the National Semiconductor Sales Office/Distributors for availability and specifications.

Supply Voltage	−0.5V to +7.0V
Inputs (Note 1)	
TTL	−0.5V to +7.0V
CMOS	−0.5V to +5.5V
Outputs (Note 1)	
TTL	−0.5V to +7.0V
CMOS	−0.5V to +5.5V
Input Current Maximum ($R_{NOMINAL}$, R_{BOOST}, CPO, VCOI, TEF)	±2 mA
Storage Temperature	−65°C to +150°C
Operating Temperature Range	0°C to +70°C
ESD Susceptibility	1500V

Recommended Operating Conditions

Symbol	Parameter	Conditions		Min	Typ	Max	Units
V_{CC}	Supply Voltage			4.75	5.00	5.25	V
T_A	Ambient Temperature			0	25	70	°C
V_{IH}	High Logic Level Input Voltage			2.0			V
V_{IL}	Low Logic Level Input Voltage					0.8	V
I_{OH}	High Logic Level Output Current	I_{OH1}	2,7 Output			−400	µA
		I_{OH2}	AMF, NRZ OUT, RD/REF CLK			−2.0	mA
		I_{OH3}	AMF, NRZ OUT, RD/REF CLK			−20	µA
I_{OL}	Low Logic Level Output Current	I_{OL1}	2,7 Output			20	mA
		I_{OL2}	RC ADJ, BITOUT			8.0	mA
			AMF, NRZ OUT, RD/REF CLK			4.0	mA
		I_{OL3}	AMF, NRZ OUT, RD/REF CLK			20	µA
$I_{RC\ ADJ}$	Pulse-Pairing/Pre-Compensation Maximum External Load Current			0.0		−16	mA
f_{NRZ}	Operating Data Rate Range			1.5		24	Mb/s
t_{mpwr}	Minimum Pulse Width of REF CLK, HIGH or LOW			10			ns
t_{mpwi}	Minimum Pulse Width of INPUT 2,7, HIGH			20			ns
t_{mpww}	Minimum Pulse Width of WRT CLK, HIGH or LOW			20			ns
t_{mpwcc}	Minimum Pulse Width of CRC, HIGH or LOW			40			ns
t_{mpwc}	CONTROL REGISTER CLOCK Minimum Pulse Width HIGH or LOW (Note 2)			40			ns
t_{scec}	CONTROL REGISTER CLOCK Setup-Time with respect to CRC (Note 2)			80			ns
t_{hcec}	CONTROL REGISTER ENABLE Hold-Time with respect to CRC (Note 2)			80			ns
t_{scdc}	CONTROL REGISTER DATA Setup-Time with respect to CRC (Note 2)			40			ns
t_{hcdc}	CONTROL REGISTER DATA Hold-Time with respect to CRC (Note 2)			40			ns
I_{CPIN}	Combined R_{NOM} and R_{BOOST} Input Current					1000	µA

Note 1: Bipolar Inputs: REF CLK, $\overline{COAST}$, R_{NOM}, R_{BOOST}
Bipolar Outputs: TC, 2,7 OUT, RC ADJ, BITOUT
CMOS Inputs: 2,7 IN, $\overline{CRE}$, CRD, CRC, PULSE POL, NRZ IN, WRT CLK, WRT GATE, AME, READ GATE
CMOS Outputs: AMF, NRZ OUT, RD/REF CLK
Analog Inputs: TEF, CPO, VCO IN

Note 2: Parameter guaranteed by correlation to characterization data. No outgoing test performed.

DC Electrical Characteristics

Symbol	Parameter	Conditions	Min	Typ	Max	Units
V_{IC}	Input Clamp Voltage (All Inputs)	V_{CC} = Min, I_I = −18 mA			−1.5	V
V_{OH1}	High Level Output Voltage	V_{CC} = Min, I_{OH} = I_{OH1}	V_{CC} − 2V	V_{CC} − 1.6V		V
		V_{CC} = Min, I_{OH} = I_{OH2}	3.5			V
		V_{CC} = Min, I_{OH} = I_{OH3}	V_{CC} − 0.1V			V
V_{OL1}	Low Level Output Voltage	V_{CC} = Min, I_{OL} = I_{OL1}			0.5	V
		V_{CC} = Min, I_{OL} = I_{OL2}			0.4	V
		V_{CC} = Min, I_{OL} = I_{OL3}			0.1	V
I_{OZ}	Maximum TRI-STATE Leakage (Note 1)	V_{CC} = Max, 0.4V ≤ V_O ≤ 2.7V			±2.0	μA
I_{CEX}	Open-Collector Leakage (Note 2)	V_{CC} = Max, V_O = Max			−100	μA
I_{IH}	High Level Input Current (Note 3)	V_{CC} = Max, V_I = 2.7V			20	μA
I_{IL}	Low Level Input Current (Note 3)	V_{CC} = Max, V_I = 0.4V			−200	μA
I_{IN}	Maximum Input Current (Note 4)	V_{CC} = Max, V_{IN} = V_{CC} or GND	−1		+1	μA
I_O (Note 5)	Output Drive Current OUTPUT 2,7	V_{CC} = Max, V = 2.125V	−12		−110	mA
	TC Pin (Note 6)	V_{CC} = Max, V = 2.125V	−36			mA
I_{CPO}	Charge Pump Output Current (K1)	100 ≤ I_{RP} ≤ 1000 (Note 7)	1.7 I_{RP}	2.0 I_{RP}	2.5 I_{RP}	μA
$I_{CPO-OFF}$	Charge Pump Output Inactive Current	100 ≤ I_{RP} ≤ 1000 (Note 7)	−0.85		+0.85	μA
I_{VCOI}	VCOI Input Leakage Current	VCOI Voltage 1.5V	−0.25		+0.25	μA
V_{RNOM}	Voltage Across R-NOM Resistor	1.2 kΩ ≤ R_{NOM} ≤ 12 kΩ	Typ − 18%	0.26 V_{CC}	Typ + 18%	V
V_{RBST}	Voltage Across R-BOOST Resistor	1.2 kΩ ≤ R_{BOOST} ≤ 12 kΩ	Typ − 18%	0.26 V_{CC}	Typ + 18%	V
I_{CC1}	Supply Current, Nominal Strobe	V_{CC} = Max (Note 8)			190	mA
I_{CC2}	Supply Current, Early Strobe	V_{CC} = Max, (Note 9)			TBD	mA

Note 1: Applies to the TRI-STATE output NRZ OUT with CTRL BIT 11 set HI, and WRT GATE HI.

Note 2: Applies to the Bipolar outputs: RC ADJ, BITOUT.

Note 3: Applies to the Bipolar inputs: REF CLK, $\overline{\text{COAST}}$, R_{NOM}, R_{BOOST}.

Note 4: Applies to the CMOS inputs: INPUT 2,7, $\overline{\text{CRE}}$, CRD, CRC, PULSE POL, NRZ IN, WRT CLK, WRT GATE, AME, READ GATE.

Note 5: This represents approximately one-half of the true short-circuit output current, I.

Note 6: This only applies to the TC Open Emitter output. Do not exceed 100 mA for more than 0.1 sec.

Note 7: I_{RP} = I_{NOM} + I_{BOOST}

Note 8: I_{CC1} is measured with the window strobe set at nominal timing (Register Bits 23 through 19 = 0,0,0,0,0); VCO operating at the maximum allowed frequency within any given range selection.

Note 9: I_{CC2} is measured with the window strobe set to the maximum early timing (Register Bits 23 through 19 = 0,1,1,1,1); VCO operating at the maximum allowed frequency within any given range selection.

External Component Selection

Symbol	Parameter	Min	Typ	Max	Units
R_{NOM}	Charge Pump Nominal Operating Current Setting Resistor (Note 1)	1.2		12	kΩ
R_{BOOST}	Charge Pump Boost Current Setting Resistor (Note 1)	1.2		∞	kΩ
C_{NOM}	R_{NOM} Bypass Capacitor (Note 2)	0.01			μF
C_{BOOST}	R_{BOOST} Bypass Capacitor (Note 2)	0.01			μF
R_{TC}	Pre-Comp/Pulse Pairing Timing Resistor (Note 3)	0.33		10	kΩ
C_{TC}	Pre-Comp/Pulse Pairing Timing Capacitor (Note 3)	33		10000	pF
$R_{RC\ ADJ}$	Secondary Pre-Comp/Pulse Pairing Timing Resistor (Note 4)	0.33		10	kΩ
$R_{BIT\ OUT}$	Secondary Pre-Comp/Pulse Pairing Timing Resistor (Note 4)	0.33		10	kΩ

Note 1: The minimum allowed value for the parallel combination of R_{NOM} and R_{BOOST} is 1.2 kΩ.

Note 2: C_{NOM} and C_{BOOST} should be high quality, high frequency type.

Note 3: R_{TC} and C_{TC} are both used to establish Pulse-Pairing and Pre-Comp timing on the TC pin.

Note 4: R_{RC} ADJ and R_{BIT} OUT modify the external timing on the TC pin for Read vs Write and inner track adjustability.

AC Electrical Characteristics

Symbol	Parameter	Min	Typ	Max	Units
t_{STOP}	READ/REF CLK Positive Transitions after READ GATE until Data Lock ZPS Sequence Begins (VCO Freezes) (Notes 1, 2)		2 (Note 4)	3 (Note 4)	—
$t_{RESTART}$	Positive 2,7 INPUT Transitions following VCO Freeze until the VCO Restarts (Note 2)		2	3	—
$t_{READ\ ABORT}$	Number of REF CLOCK Cycles following READ GATE Deactivation until REF CLOCK Lock ZPS Sequence begins (Note 2)			4	—
t_T	Window Truncation (Half Window Loss); 10 and 20 Mbit/s at Strobe Position M = −2 (Note 3)		4% × T_{VCO}		ns
ϕ Linearity	Phase Range for Charge Pump Linearity (wrt VCO) (Note 2)		$\pm\pi$		Radians
K_{VCO}	VCO Gain Constant (Note 6)	1.0 ω_o	1.2 ω_o	1.7 ω_o	rad/sec V
$f_{MAX\ VCO}$	VCO Maximum Frequency; Control Bits 17,16 = 11	70			MHz
t_{ZPSR}	Zero Phase Start Trigger Bit Targeting Accuracy, READ GATE Activation (READ) (Note 5)		2		ns
$\Delta f_{VCO}/f_{RFC}$	Automatic f_{VCO} Range Limiting (Note 2)		50		%
t_{RRCH}	RD/REF CLK Rest Period at Assertion or Deassertion of READ GATE (Note 2)	½		3	T_{VCO}
Encoder Delay	Serial Delay Time in REF CLK Cycles + ns Delay (Note 7)			10 + 100	Cyc + ns
Decoder Delay	Serial Delay Time in REF CLK Cycles + ns Delay			8 + 100	Cyc + ns

AC Electrical Characteristics (Continued)

Symbol	Parameter		Min	Typ	Max	Units
t_{plh} AMF Assertion Time	ST506A and B; After REF CLK following Address Detection (ESDI, SMD, and IBM Gap Type Address are Asynchronous)		50		100	ns
t_{phl} AMF Deassertion Time	ESDI and SMD (Spec & Noise Tolerant); Falling Edge of AME		50		100	ns
	DP8466 Compatible (ESDI and IBM); 2nd Input 2,7 Pulse		50		100	ns
	ST506A: 2nd Input 2,7 Pulse		50		100	ns
	ST506B; Next REF CLK Positive Edge after Falling RD GATE		50		100	ns
t_{mpw}	REF CLK Minimum Pulse Width t_{mpwr+} t_{mpwr-}		5			ns
	INPUT 2,7 Minimum Input Pulse Width t_{mpwi+}		15			ns
	WRT CLK Minimum Input Pulse Widths t_{mpww+} t_{mpww-}		10			ns
$t_{s1/0}$ $t_{h1/0}$	REF CLK Setup and Hold Times to WRT CLK				10	ns
	WRT CLK Setup and Hold Times to AME, NRZ IN, WRT GATE, READ GATE				20	ns
t_{plh}, t_{phl}	Propagation Delay from REF CLK to OUTPUT 2,7				100	ns
$t_{rfc-rrc}$	Propagation Delay from REF CLOCK to RD/REF CLK, READ GATE LOW				50	ns
$t_{skw\pm}$	RD/REF CLK Negative Edge to NRZ OUTPUT Transition				±5	ns
L_{PDT}	Length of Valid Preamble Pattern Required for Internal PREAMBLE DETECTED (Note 2)	3T Preamble	31	32	33	2,7 Input Pulses
		4T Preamble	15	16	17	
t_{Str}	Window Strobe Time Step (M = Hex Value of Bits 22–19 in CONTROL REGISTER; Bit 23 = Sign Bit			M × (1.8%) × T_{RFC}		ns

Note 1: ST506 and SMD modes utilize a Preamble Prequalification routine to guarantee that the synchronizer only locks to data during a valid Preamble Field. This adds a prequalification time to the T_{STOP} time parameters as follows:
ST506: Adds 8 full preamble fields PRIOR to the T_{STOP} parameter
SMD: Adds the time it takes to locate a valid Address Mark, and output a valid AMF

Note 2: Limits are guaranteed by design or correlation to characterization data; no outing testing is performed.

Note 3: *The preliminary DP8469 static window specification, IT, applies only to the factory-tested data rates of 10 Mb/s (Control Bits 17,16 = 01) and 20 Mb/s (17,16 = 10), with the component values as listed for each corresponding data rate in Figures 7* and *12*, test configuration as shown in *Figure 24*, test procedure as shown in *Figure 25*, and strobe word M = −2. Significant variation in IT due to the use of other filters and data rates is not expected.

Note 4: t_{ZPSR} (ZPS Read) gauges the accuracy with which the ZPS circuitry aligns the VCO to the triggering 2,7 INPUT bit internally (i.e., initial phase step) at the completion of a ZPS operation following READ GATE assertion.

Note 5: $I_{IN} = V_{CC}/(4 \times R_{IN})$. $R_{IN} = R_{NOM}$ (HGD High) or $R_{NOM} \| R_{BOOST}$ (HGD Low).

Note 6: Specification for 25°C only. Temperature coefficient = −0.4%/°C.

Note 7: Encoder Serial Delay specified for Pre-Comp set to Bypass Mode (bits 2,1,0 = 000). Non-Bypass Pre-Comp adds 5 REF CLK cycles plus 50 ns.

DP8469 Loop Filter Component Values

Preamble Type	10 Mbit/Sec		20 Mbit/Sec		Units
	3T	4T	3T	4T	
Ref Clk Frequency	20	20	40	40	MHz
NRZ Data Rate	10	10	20	20	Mbit/s
Sync Field Frequency	20	20	40	40	MHz
ζ Min	20	20	40	40	None
ζ Max	20	20	40	40	None
ζ Sync	20	20	40	40	None
Sync Field Frequency	6.7	5	13.3	10	MHz
ω Sync	20	20	40	40	Krad/s
C1 (Main Loop)	0.018	0.018	0.0082	0.082	μF
R1 (Main Loop)	150	150	150	150	Ω
C2 (Main Loop)	510	510	200	200	μF
CT1 (TEF)	0.056	0.056	0.0027	0.027	μF
RT1 (TEF)	68	68	68	68	Ω
TEF Settling Time	9.6	9.6	4.6	4.6	μs

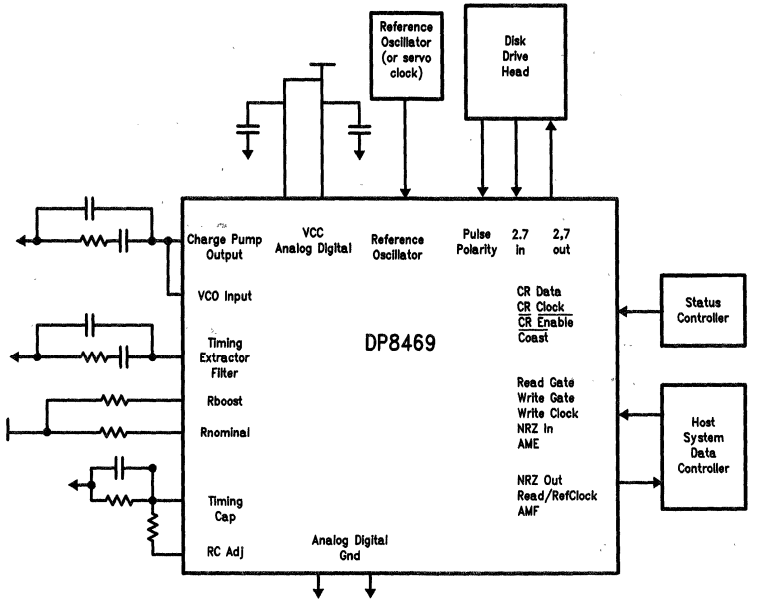
Loop Filter Component Values

Preamble Sync natural frequency has been chosen to yield phase error $\leq$ 0.063 radians (i.e., 1% $\times$ 2π) at sync field end, given a 1% frequency step at READ GATE assertion. $R_{NOM} = R_{BOOST} = 2.4k$ for all loop filter selections. These values apply for Bit 18 = HI, device switches to LOW Gain on assertion of READ GATE.

The TEF settling times are given which indicate time required for the DP8469 to accommodate a change of Strobe setting from nominal selection to either extreme (early/late), or vice versa, to within approximately 1% of final value.

The values listed in the Loop Filter Component Values table are the approximate filter values that are used for Static Window Truncation testing. These values represent a simple solution, yielding only average circuit behavior. Please see National Semiconductor's DP8459 datasheet for a more thorough discussion on loop filter component selection.

Application Diagram



TL/F/9386–3

Encoder Serial Delay Times

Encoder Delay Time Parameter Definitions

t_{es} = Encoder Start time from WRT GATE

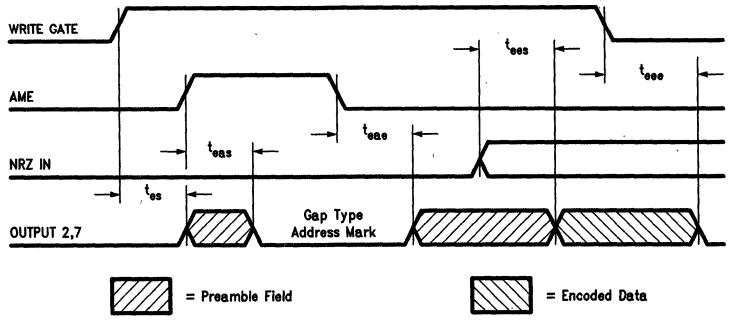
t_{eas} = Encoder Address Mark Start time from assertion of AME

t_{eae} = Encoder Address Mark End time from deassertion of AME

t_{ees} = Encoder Encyrption Start time from first NRZ IN bit

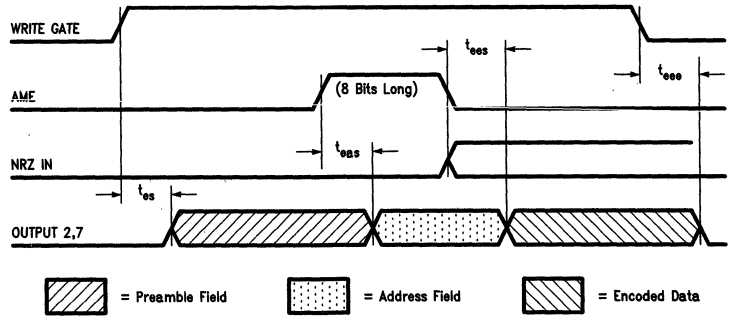
t_{eee} = Encoder Encryption End time from deassertion of WRT GATE

ESDI/SMD/IBM Modes (Gap Type Address Marks)



TL/F/9386–15

ST506A and B Modes (Synchronous Address Marks)



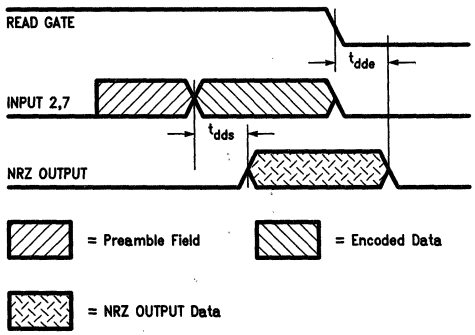
TL/F/9386–16

Decoder Serial Delay Times

Decoder Delay Time Parameter Definitions

t_{dds} = Decoder Decryption Start time from INPUT 2,7 data

t_{dde} = Decoder Decryption End Time from deassertion of READ GATE

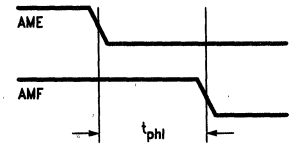


TL/F/9386–17

This parameter defines the time delay for a specific INPUT 2,7 pattern to enter the device, and its resulting NRZ output to be present at the NRZ OUTPUT pin.

AMF Assertion Delay Times by Mode

ESDI/SMD Spec and Noise Tolerant Modes

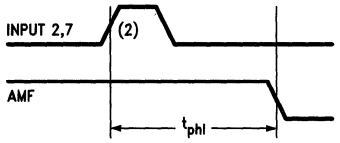


TL/F/9386–18

AMF deasserts on the FALLING edge of READ GATE.

AMF Assertion Delay Times by Mode (Continued)

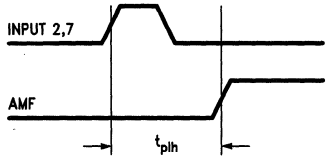
DP8466 Compatible Modes (ESDI and IBM)



TL/F/9386–19

AMF deasserts on the SECOND INPUT 2,7 pulse after assertion

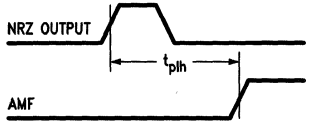
IBM and All ESDI/SMD Modes



TL/F/9386–20

AMF asserts on the FIRST INPUT 2,7 pulse after valid Address Mark

ST506 A and B Modes

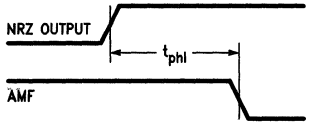


TL/F/9386–22

AMF asserts on the FIRST NRZ OUTPUT pulse after valid Address Mark

ESDI/SMD Spec and Noise Tolerent Modes

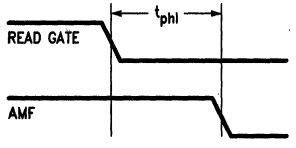
ST506A Mode



TL/F/9386–21

AMF deasserts on the SECOND NRZ OUTPUT pulse after assertion.

ST506B Mode

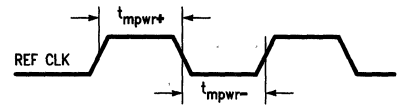


TL/F/9386–23

AMF deasserts on the FALLING edge of READ GATE.

Input Minimum Pulse Widths

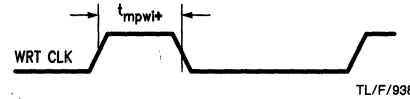
REF CLK Min Pulse Widths



TL/F/9386–24

REF CLK requires minimum input positive and negative pulse width times; including both mpw+ and t_{mpw-}

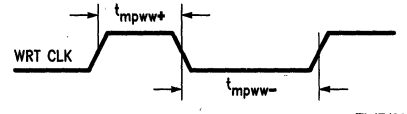
Input 2,7 Min Pulse Widths



TL/F/9386–26

INPUT 2,7 requires a minimum positive pulse width: t_{mpwi+}; there is no minimum negative pulse width spec.

WRT CLK Minimum Pulse Widths

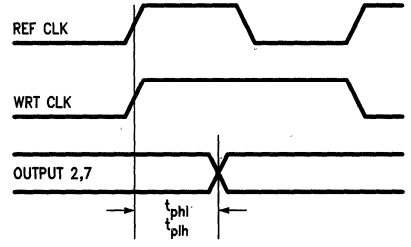


TL/F/9386–27

WRT CLK Input Minimum Positive and Negative Pulse Widths, t_{mpww+} and t_{mpww-}.

Output Propagation Delay Times

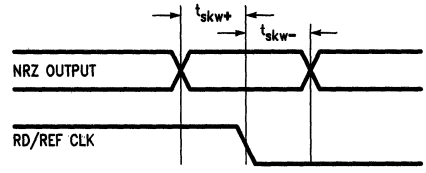
REF CLK to OUTPUT 2,7 Propagation Delay Time



TL/F/9386–25

REF CLK to OUTPUT 2,7 for t_{plh} and t_{phl}. Delay time is specified in both REF CLK cycles and ns of delay. WRT CLK edges coincident with REF CLK.

NRZ OUTPUT vs RD/REF CLK T_{skew} for all Read Modes

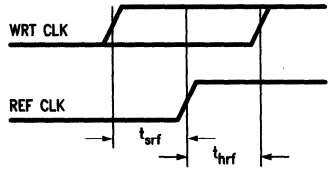


TL/F/9386–28

T_{skw+} defines data changing prior to the clock.
T_{skw-} defines data changing after the clock.

Input Setup and Hold Times

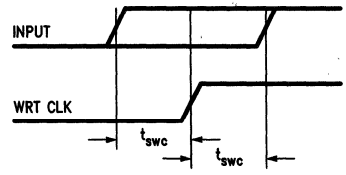
REF CLK Setup and Hold Times



TL/F/9386–29

Setup and Hold Times (t_{srf} and t_{hrf}) from REF CLK to WRT CLK.

WRT CLK Setup and Hold Times
NRZ IN, AME, WRT GATE, READ GATE

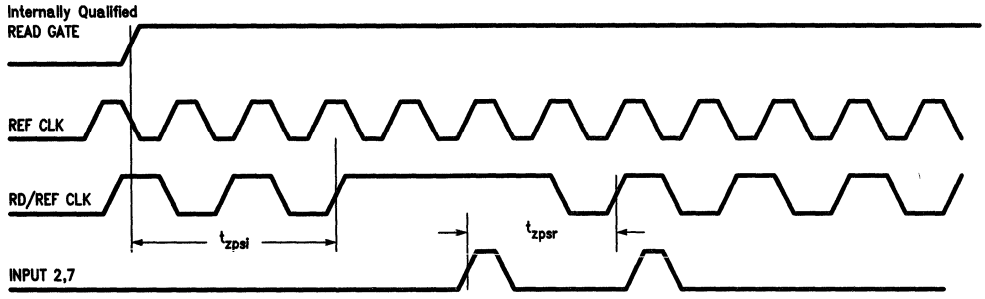


TL/F/9386–30

Setup and Hold Times (t_{swc} and t_{swc}) for the four inputs: AME, WG, RG and NRZ IN.

Zero Phase Start Timing Sequence

ZPS Stop and Restart Delay Time



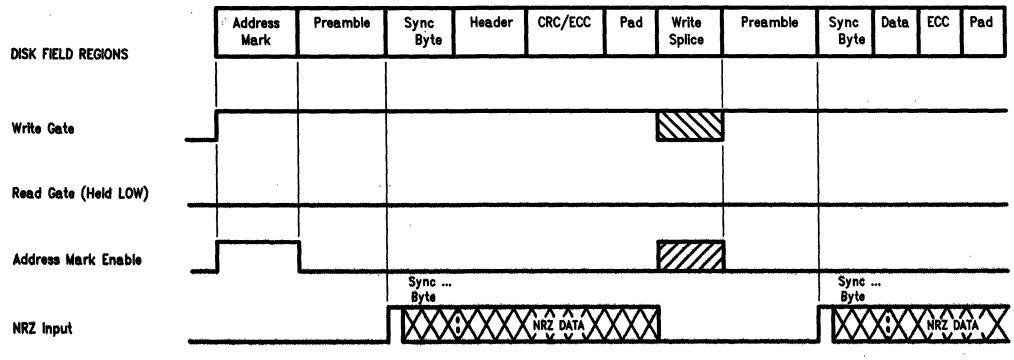
TL/F/9386–31

ZPS Delay Time measures the number of REF CLK cycles that pass from the time READ GATE is passed to the Synchronizer, until Zero Phase Start interrupts the VCO and stops RD/REF CLK. READ GATE is internally qualified in all SMD and ST506 modes to prevent the Synchronizer from attempting to lock when there is no data present (outside the Preamble Field). The basic t_{zpsi} and t_{zpsr} delay times are listed for the ESDI and IBM modes, while the SMD and ST506 modes have prequalifier adders. Delay times t_{zpsi} measures the time to interrupt the VCO, and t_{dzpsr} measure the time to Restart the VCO.

Address Mode	RG Qualified On:	t_{dzpsi} (in REF CLK) Cycles)	t_{dzpsr}
All ESDI Modes	N/A	≤3	≤ 4 after Falling RG
All SMD Modes	AMF Assertion	≤3 + AMF Detection	≤ 4 after Falling RG
All ST506 Modes	Preamble	≤3 + 12 Preamble Patterns	≤ 4 after Falling RG

ESDI (Std and Nt) MODE Control Waveforms

Write Cycle

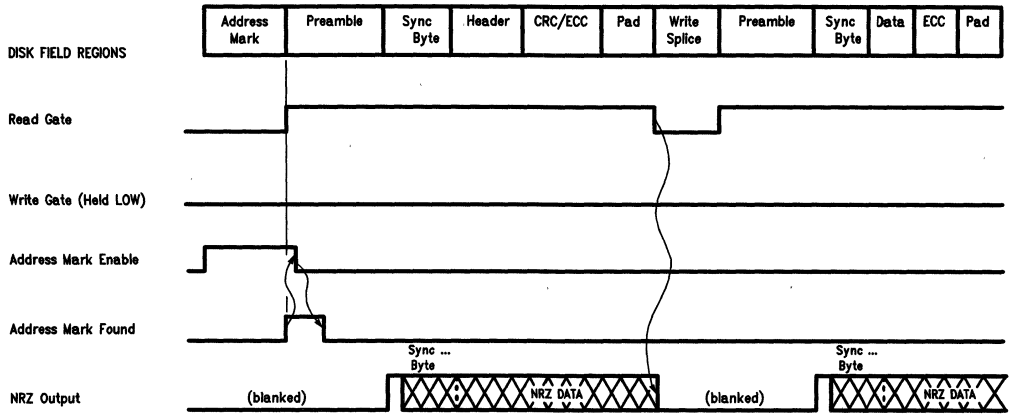


Note: Cycling either Write Gate, ▨ or AME ▨ resets the encoder, and starts the Preamble again.

TL/F/9386–4

ESDI (Std and Nt) MODE Control Waveforms (Continued)

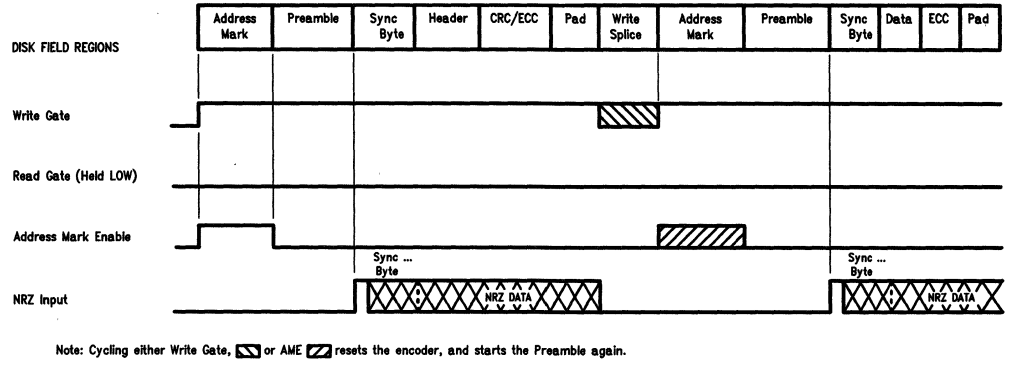
Read Cycle



TL/F/9386–5

SMD (Std and Nt) MODE Control Waveforms

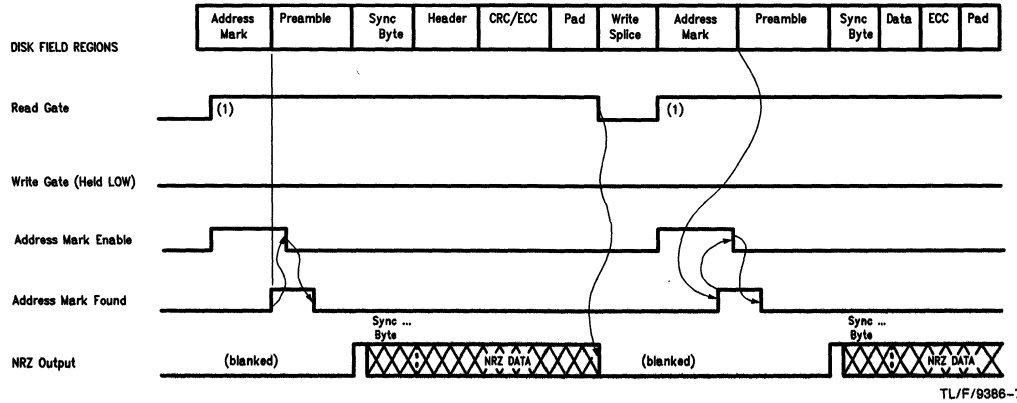
Write Cycle



Note: Cycling either Write Gate, ▨ or AME ▨ resets the encoder, and starts the Preamble again.

TL/F/9386–6

Read Cycle

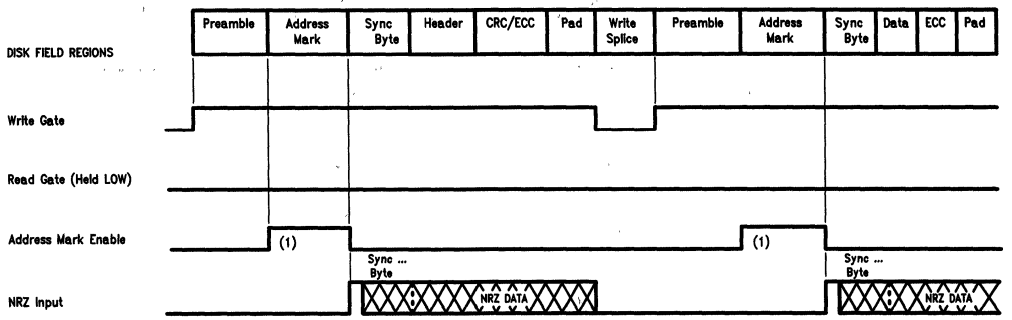


TL/F/9386–7

Note 1: Read Gate to the Synchronizer is internally qualified with AMF.

ST506A MODE Control Waveforms

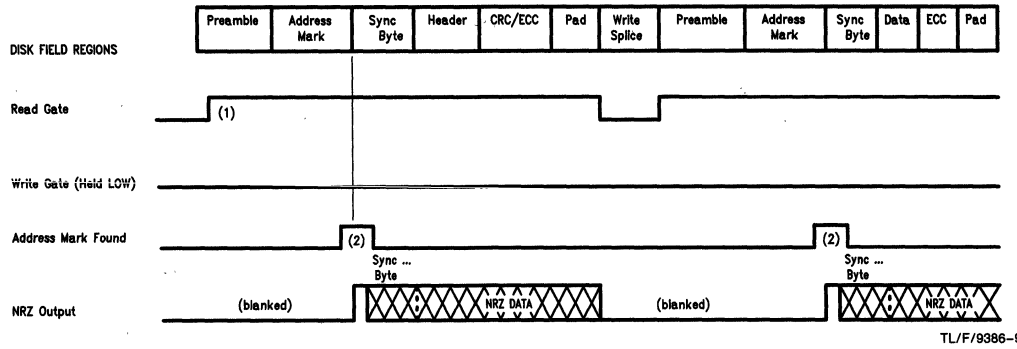
Write Cycle



TL/F/9386–8

Note 1: AME must be exactly 8 NRZ bits.

Read Cycle



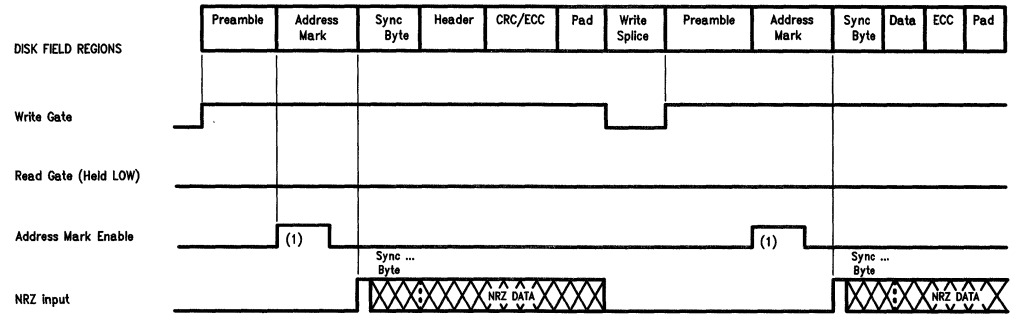
TL/F/9386–9

Note 1: The DP8469 employs a Preamble Prequalifier to ensure that the Read Gate is only passed to the synchronizer after the Preamble field has been found.

Note 2: AMF Terminated by the Second NRZ OUTPUT bit after NRZ OUTPUT unblanks.

ST506B MODE Control Waveforms

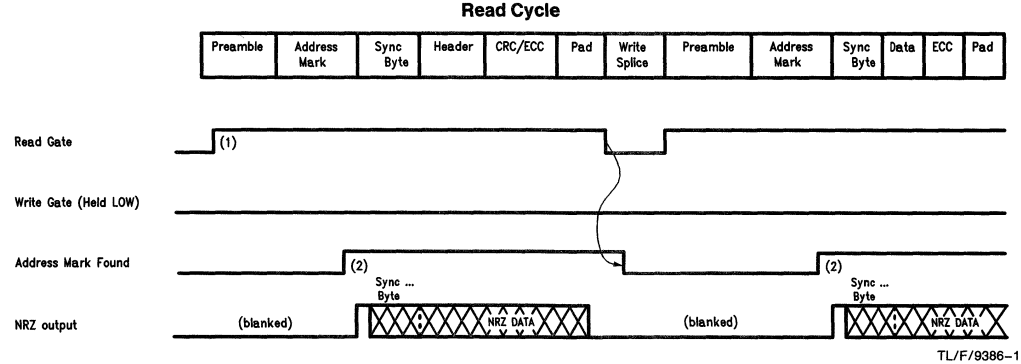
Write Cycle



TL/F/9386–10

Note 1: AME must be exactly NRZ bits.

Read Cycle



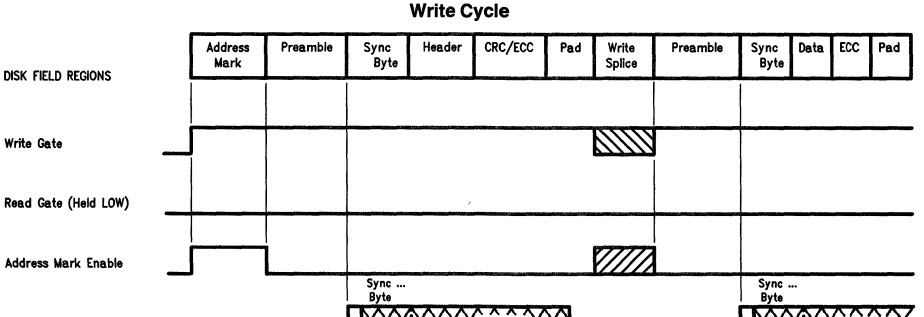
TL/F/9386–11

Note 1: The DP8469 employs a Preamble Prequalifier to ensure that the Read Gate is only passed to the synchronizer after the Preamble field has been found.

Note 2: AMF terminated by the deassertion of Read Gate.

IBM and DP8466 ESDI Mode Control Waveforms

Write Cycle



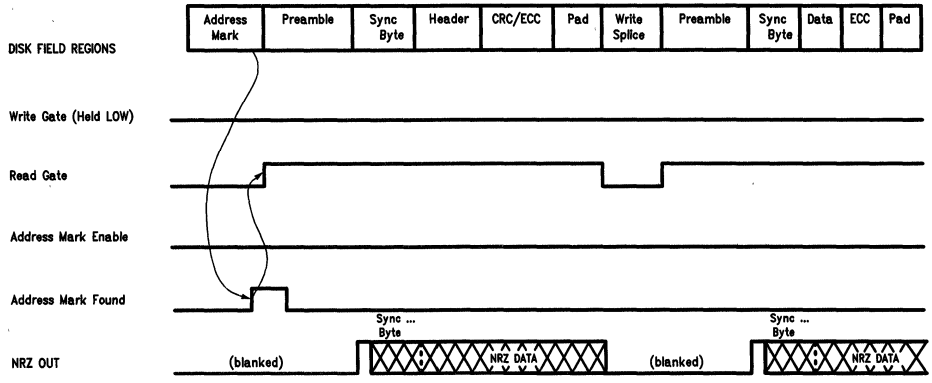
Note: Cycling either Write Gate, ▨ or AME ▨ resets the encoder, and starts the Preamble again.

IBM Address Mark is written as a 2 byte gap with 3 transitions, DP8466 ESDI uses standard 3 byte transitionless ESDI gap

TL/F/9386–32

IBM and DP8466 ESDI Mode Control Waveforms (Continued)

Read Cycle

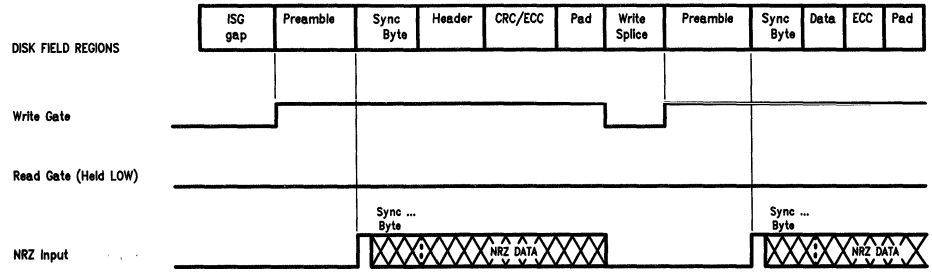


TL/F/9386–33

Note: AME is a DON'T CARE during Read Mode.

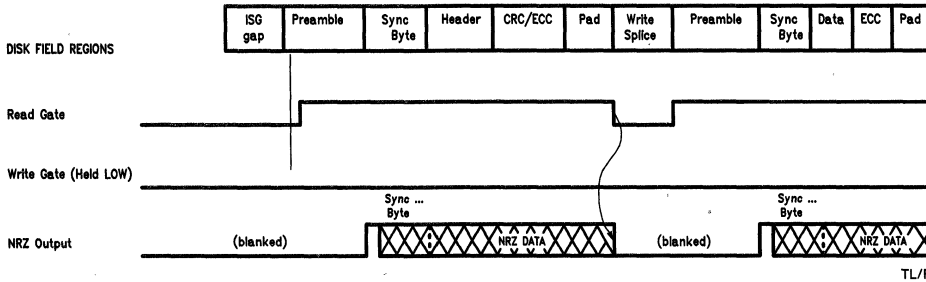
Hard Sector MODE Control Waveforms

Write Cycle



TL/F/9386–12

Read Cycle



TL/F/9386–13

Precautions for Disk Data Separator (PLL) Designs—How to Avoid Typical Problems

National Semiconductor
Application Note 414
William Llewellyn

The disk data separator/synchronizer PLL is subject to a unique set of concerns, all of which can be accommodated when adequate precautions are taken in system design.

FRACTIONAL HARMONIC LOCK

The frequency discrimination capacity of the digital phase detector within the data separator/synchronizer is suppressed whenever a pulse gate technique is employed. Although this pulse gating technique is a standard in disk drive applications and is necessary in order to allow the PLL to remain phase locked to randomly spaced disk data bits, it essentially causes the phase detector to behave as would an analog quadrature multiplier, i.e., the capture range of the loop takes on the finite value related to the loop bandwidth. Under ordinary circumstances, this is quite acceptable; however, it does permit the PLL to become susceptible to a form of quasi-stable false lock to fractional harmonics of the input frequency. (For example, a typical lock null for this phenomenon would be where the VCO stabilizes at 5/6 or 6/5 of its nominal frequency.) The conditions for occurrence of this are:

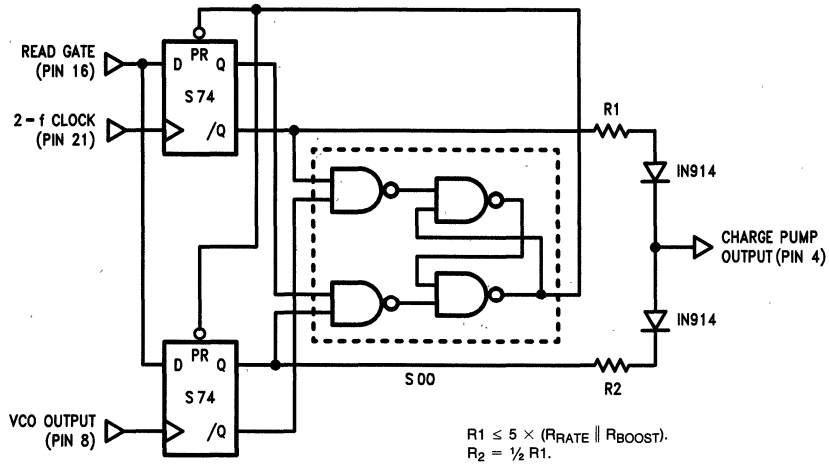
1) Pulse gate in use;
2) Periodic pattern is present (i.e., preamble);
3) Perturbation occurs either during or just prior to the periodic pattern, causing the VCO to swing outside of the dynamic capture range of the loop.

Since the capture range in a typical disk PLL configuration is on the order of ±2% of the data rate, it can be seen that harmonic lock could easily occur given an adequate perturbation of the loop. Typical causes of perturbations would be media defects and spurious noise pulses, among others, but the most commonly seen occurrence within soft-sectored systems is where an attempt is made to "read" through the write splice region on the disk (zone where the head write current is either switched on or off) during a sector search operation. Typical system-level symptoms of fractional harmonic lock are "sector not found" and "address mark not found" errors. Data (CRC or ECC) errors rarely are seen here because the phenomenon occurs primarily during the sector search routine.

Recovery from harmonic lock will occur readily when the read operation is terminated if:

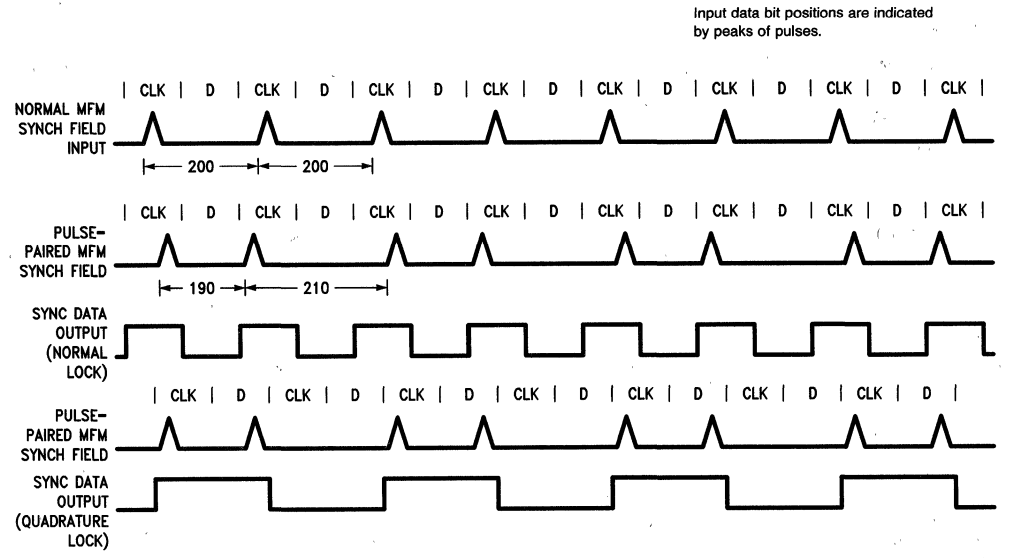
1) frequency discrimination is re-introduced as the PLL is re-locked to the reference clock, or
2) the PLL bandwidth is raised to a higher value (capture range is extended) as the PLL is re-locked to the reference clock, or
3) the phase transient experienced by the PLL as its input is switched back to the reference clock is enough simply to jar the PLL back to the correct frequency.

Item #1 is incorporated within all of National's current hard disk data separator/synchronizer circuits (the DP8460/50 are excepted, being replaced by the DP8465/55). Item #2 (user optional) is incorporated within all of National's hard disk PLL's. Systems which incorporate the frequency lock function (#1) along with a suitable sector search algorithm will rarely, if ever, encounter difficulty in this area. If the



TL/F/8598–1

FIGURE 1. External Phase-Frequency Comparator Circuit for the DP8460



TL/F/8598–2

FIGURE 2. Timing Diagram of PLL Quadrature Lock Within a Symmetrically Pulse-Paired Synch Field

system employs a PLL which does not incorporate frequency acquisition when locked to the reference signal (such as the DP8460/50 predecessor of the DP8465/55), either a simple external circuit may be added if desired to achieve the function (see *Figure 1*), or the PLL can be updated by inclusion of the DP8465 or DP8455. The DP8461 or DP8451 would provide the most reliable solution (frequency aquisition of both preamble and reference clock), but may be used only within hard or psuedo-hard sectored systems. (Note that the resistor values given in *Figure 1* are initial recommendations only; values may need to be adjusted to optimize system performance.)

Within systems where it becomes evident that the reading of write splices is consistently producing sector-not-found errors, while at the same time it is not possible to either modify the sector search algorithm (in order to avoid the splices) or to incorporate the lock support circuitry of *Figure 1*, the PLL can be made less sensitive to the write splice disturbance by the lowering of the loop bandwidth. This is recommended only as an interim solution until firmware or hardware accommodations can be made.

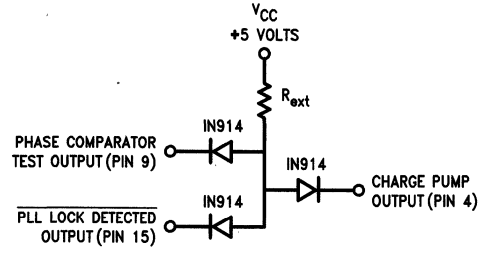
QUADRATURE LOCK

Another form of false lock may also occur (pulse gate in use) within a periodic disk pattern (preamble) given one additional condition; the periodic disk pattern being presented to the PLL exhibits a pulse-pairing phenomenon (typically introduced by the data channel electronics); see *Figure 2.* Within this particular pattern, PLL has the potential to lock to the correct frequency while remaining caught on a phase null 90 degrees from nominal. In this case, each pair of bits is interpreted by the PLL as residing in two directly adjacent windows (actually a violation of all standard disk codes) with the two subsequent windows empty. Although the bits appear to be greatly shifted within these windows, the phase corrections produced complement each other and average to a filtered DC value of zero. This repeating pattern is thus self-sustaining.

Quadrature lock is unique in that it is more likely to occur within a relatively well designed, noise-free system environment. The reason for this is that the randomizing effect noise ordinarily has on the data stream has been minimized, preserving the purity of the pulse paired pattern and thus increasing the probability of this form of lock. Again, this form of lock is generally only seen within the preamble, and may occur within either soft or hard sectored systems. The most typical symptoms are "address mark not found" or "ID error", with "sector not found" occurring, but less frequently. Easily recognized waveform patterns seen at the separator/synchronizer outputs would be (1) the Synchronized Data Output exhibits a 110011001100 . . pattern instead of the standard 1010101010 . . . preamble pattern; (2) the Phase Comparator Test output pulse width consistently remains at approximately half of the VCO period (nominal width should be 7–12 nanoseconds); (3) -Lock Detected does not become active (low).

The most robust solution to this phenomenon (as well as to harmonic false lock, as mentioned above) is to incorporate a hard or pseudo hard-sectored search algorithm in conjunction with a data separator/synchronizer which employs frequency acquisition within the preamble. The frequency acquisition mode allows no residual phase or frequency error within the PLL when locked, and thus the possibility of both quadrature and harmonic lock is eliminated.

Although the modified sector search algorithm of the first solution may be possible, certain system constraints may not allow it to be practical. A second, highly effective solution to quadrature lock involves the inclusion of four passive



Recommended value for R_{ext}:
$10[R_{rate} \| R_{boost}] \leq R_x \leq 20\ [R_{rate} \| R_{boost}]$

FIGURE 3. External Circuitry Used to Insure Correct PLL Lock to a Pulse-Paired Synchronization Field

elements external to the National disk PLL (see *Figure 3*) which will deliberately force the window to shift away from the 90 degree phase null when (and only when) quadrature lock occurs. The passive network is automatically disabled once the PLL detects preamble lock. Although a recommended value is given for the resistor in this support circuit, some experimenting may be required in determining an optimum value for use within any particular system.

VCO JITTER

The inherent purity of the VCO's operating frequency is a key element in the accuracy of the data separator/synchronizer window generation. Any "jitter" present in the VCO frequency (any modulation of the period of the waveform by noise or any other source) will degrade the performance of the PLL. Within National's initially released DP8460/50 data separators-synchronizers, it has been found that maintaining a value of R_{rate} at or below 820Ω has a stabilizing effect on the jitter performance of the VCO circuitry. Although this is primarily a characteristic of these two devices, we are recommending the following guidelines be followed in the selecting of charge pump resistors and loop filter components for all of the hard disk data separator/synchronizer circuits (see table I):

1) An 820Ω value resistor should be substituted for the originally recommended value of 1.5 kΩ.

2) Although this new R_{rate} value is below the original DP8460 specification limit, a substitute requirement has been placed on both R_{rate} and R_{boost} to maintain proper circuit operation:

$$R_{rate} \| R_{boost} \geq 350\Omega$$

(i.e., the parallel value of R_{rate} and R_{boost} should not fall below 350Ω.)

3) If the inclusion of an 820Ω value for R_{rate} means a component change within an existing system (i.e., the user had been employing some higher value), all other component values associated with the loop filter must also be modified in order to maintain the original PLL response characteristics within the disk data field:

Define: M = R_{rate}(old)/820 [eg., (1500/820)]. Then,

$$CLF1' = CLF1 * M$$
$$CLF2' = CLF2 * M$$
$$RLF1' = RLF1 / M$$

4) Additionally, in the cases where the external Phase-Frequency circuitry and/or the Quadrature lock circuitry are in use:

$$R1' = R1 / M$$
$$R2' = R2 / M$$
$$R_{ext}' = R_{ext} / M$$

Table I. Data Separator/Synchronizer Reference List

Device	Synchronized Codes	Separated Codes	Frequency Lock	Delay Trim
DP8461*	MFM; 1, N	MFM	Reference & Data	None
DP8462*	2, 7 MFM;1, N	None	Reference & Data (optional)	Optional
DP8465*	All	MFM	Reference	None
DP8451	MFM; 1, N	None	Reference & Data	None
DP8455	All	None	Reference	None

Note 1: "All" code synchronization does not include GCR.

Note 2: DP846X devices are in the 24-pin, 300 mil. package; DP845X devices are in the 20-pin, 300 mil. package.

Note 3: * Also available in 28-lead plastic chip carrier.

Note 4: DP8461 and DP8451 pinouts match the DP8465 and DP8455, respectively; for use with hard and psuedo-hard sectoring only.

Note 5: DP8462 incorporates optional frequency acquisition for 2, 7 synchronization fields, but may be used as a data synchronizer for any disk code.

Note 6: DP8451 and DP8455 also available in PCC package (20 pin).

PRINTED CIRCUIT BOARD LAYOUT RECOMMENDATIONS

The phase locked loop is inherently a sensitive device, and thus the environment in which it is operated should be optimized wherever possible to improve reliability. The following list applies for National's family of hard disk data separator/synchronizer circuits:

1) Establish a local V_{cc} island or net, separate from the main V_{CC} plane, to which the device and its associated passive components can be connected. V_{CC} supply filtering should be liberal and in very close proximity to the chip. The electrical lead length of the filter capacitance between the V_{CC} and ground pins themselves should be as short as possible (minimizing lead inductance). Inclusion of a quality high-frequency capacitor, such as a 1000 pF silver-mica capacitor, in parallel with a ceramic 0.1 μF capacitor, is recommended. (Note: the chip is particularly sensitive to inadequately filtered switching supply noise.)
2) Effective capacitive bypassing of the R_{boost} and R_{rate} pins (#2 and #3) directly to the V_{CC} pin is very important. Again, use quality, high-frequency capacitors and maintain the shortest possible electrical lead length.
3) Use the main digital ground plane for all grounding associated with the device. The ground pin and the PG1 pin should tie directly to this plane.
4) Do not locate the chip in a region of the PC board where large ground plane currents are expected.
5) Locate all passive components associated with the chip as close to their respective device pins as possible.
6) Orient the chip's external passive components so as to minimize the length of the ground-return path between each component's ground plane tie point and the chip's ground pin. (Ground noise at the loop filter components, RLF1, CLF1 and CLF2, which is not identically present at the ground pin (common mode), is coupled through the filter components into the VCO control voltage pin.)
7) Include no planing whatsoever (V_{CC} or ground) directly between adjacent pins. This will minimize parasitic capacitance at each pin. Planing between the two pin rows, however, is recommended (directly beneath the package).
8) Avoid running signal traces between pins.
9) Run no digital signal lines between or adjacent to the analog pins or signal traces (pins 1 through 7 and PG3) in order to avoid capacitive coupling of digital transients.
10) Minimize the total lead length of the C_{VCO} capacitor. Inductance in this path degrades VCO performance, as does parasitic pin capacitance.
11) Do not place any bypass filtering at the R_{VCO} pin (minor coupling of the VCO waveform into this pin is normal and acceptable).
12) Eliminate negative-going voltage transients (undershoot) at the digital input pins (pre-termination of driving lines may be necessary) to avoid drawing transient input-clamp-diode current from the device pins.
13) Minimize digital output loading; i.e., if outputs must drive large loads or long lines, employ buffers.
14) Allow unused digital output pins to float, unconnected to any net.
15) Avoid locating the chip within strong electromagnetic fields. If possible, choose the "quietest" region of the board.
16) If chip socketing is desired, use a low-profile, low mutual capacitance, low resistance, forced-insertion type (socket-strips are recommended). Avoid the use of "ZIP-DIP's".
17) Do not use wire-wrap interconnect, even in an evaluation set-up.
18) Make allowance for pin-to-pin capacitance when determining C_{VCO} (Typically 4–5 pF) from data sheet formula.

Designing with the DP8461

National Semiconductor
Application Note 415
Kern Wong

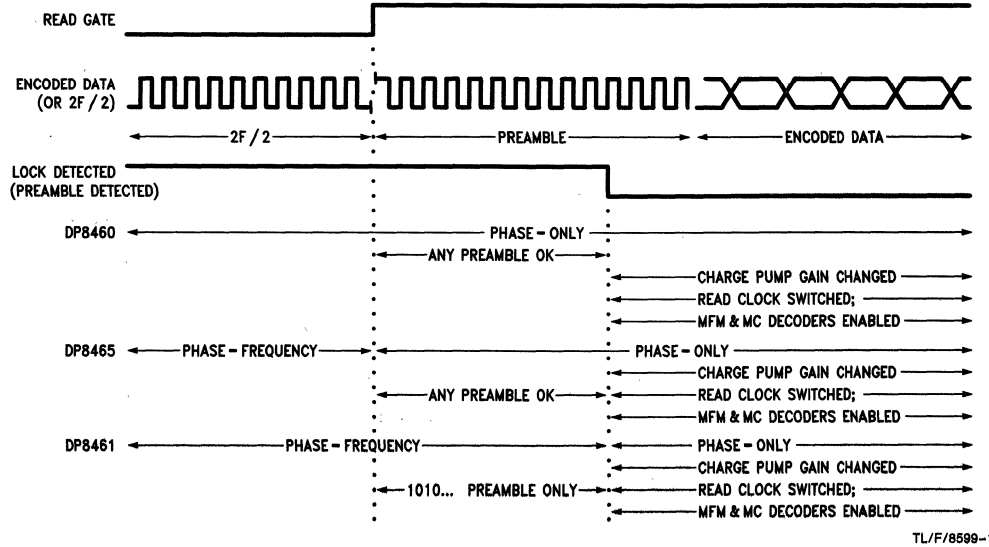
GENERAL DESCRIPTION

The DP8461 is one of the second generation data separator/synchronizer products introduced following the highly successful DP8460 single chip PLL circuit for applications in rotational memory storage systems. The DP8461 consists of the same basic functional blocks as the first generation device (DP8460). It has a proprietary pulse-gate which features an accurate silicon delay line, an edge-triggered digital phase comparator, a high speed matched charge pump, high impedance buffer amplifier, and a temperature compensated stable voltage-controlled-oscillator (VCO). It also contains MFM decoder, missing clock detector, and lock-detect control circuitry for maximum design flexibility.

Like the DP8465, the DP8461 performs PHASE-FREQUENCY COMPARISONS in the non-read mode (READ GATE is "Low"). This enhancement eliminates the possibility of false lock to the reference signal during a power-up sequence or when returning from a read operation. Furthermore, the DP8461 has been designed to allow PHASE-FREQUENCY COMPARISONS to continue into the preamble field during read mode (READ GATE is "High"). This feature eliminates the possibility of a Quadrature Lock or Harmonic Lock problem occurring in the PLL synchronization field. In order to take advantage of phase-frequency comparison during preamble detection, the DP8461 requires a "Qualified Read-Gate" (that is the READ GATE shall be asserted only within the preamble or maximum frequency field span). Since the DP8461 looks for a 1010... encoded data pattern while doing PHASE and FREQUENCY COMPARISONS in the read mode, it must be used only with a code employing this preamble such as MFM, (1,7) or (1,8). The DP8461 is pin-for-pin compatible with the DP8460 and DP8465 parts; and is also functionally equivalent to them with the exceptions of extended Phase-Frequency Comparison during preamble and a "Qualified Read-Gate" requirement. The DP8461 can be used as a synchronizer for MFM, (1,7), and (1,8) codes or as a data separator for MFM only. *Figure 1* shows a diagrammatic comparison of the key functional features of the three part types mentioned above.

CIRCUIT OPERATION

The DP8461 is in the non-read mode whenever the READ GATE is deasserted (Low). The 2F REFERENCE CLOCK INPUT is divided by two and transmitted to the READ CLOCK OUTPUT via a multiplexer. In this mode the VCO DIVIDED BY TWO is locked onto the 2F CLOCK DIVIDED BY TWO, keeping the VCO close to the data frequency in anticipation of locking onto the actual data stream. While in the non-read mode, PHASE-FREQUENCY COMPARISONS are always employed to eliminate any possibility of false lock.



TL/F/8599–1

FIGURE 1. DP8460/65/61 Modes of Operation Comparison Diagram

Since the DP8461 continues to make PHASE-FREQUENCY COMPARISONS when the read mode is entered, the assertion of READ GATE should only occur over a PREAMBLE or 1010... pattern. The DP8461 enters the read mode after a selectable delay time which may be either one or thirty-two VCO cycles. The 2-byte (32 VCO cycles) delay is useful in hard-sectored drives for allowing a gap pattern to pass before the PLL locks onto the data pattern. Soft-sectored drives do not need this delay. Once in the read mode, the PLL reference input is switched from the 2F CLOCK source to the ENCODED DATA INPUT. The PLL remains in the high track rate mode and continues to perform PHASE-FREQUENCY COMPARISONS to quickly lock onto the repetitive encoded preamble.

By careful selection of the loop filter components, it takes less than one byte time for the VCO to lock onto the data stream sufficiently for preamble detection to begin. As soon as 2 bytes (16 consecutive pulses) of the selected (ones or zeroes pattern) preamble are detected, the $\overline{\text{LOCK DETECTED}}$ OUTPUT goes low and this causes the PLL circuit to switch from PHASE-FREQUENCY comparisons to PHASE-ONLY comparisons. In a typical disk drive application, the $\overline{\text{LOCK DETECTED}}$ OUTPUT may be directly connected to the $\overline{\text{SET PLL LOCK INPUT}}$. When a low level is present on the $\overline{\text{SET PLL LOCK INPUT}}$, the CHARGE PUMP changes from a high to low tracking rate, the source of the READ CLOCK signal switches from the 2F CLOCK INPUT to the VCO CLOCK, and the MFM decoder becomes enabled and begins to output decoded NRZ data. If the DP8461 is employed as a data separator for MFM encoded data, the READ CLOCK OUTPUT and the NRZ READ DATA OUTPUT (which is synchronized to the READ CLOCK) should be used. These TTL compatible signals can be connected directly to a Disk Data Controller such as the DP8466 which controls Winchester or floppy disk drives. The MISSING CLOCK DETECTED OUTPUT can also be utilized for MFM-encoded data for soft-sectored disk drives. It should be noted, however, the circuit is designed only to recognize a missing MFM clock-bit which is framed by two existing clock bits. In order to insure the detection of an address mark, simultaneous monitoring of the NRZ output for an "A1" hexidecimal code and the MISSING CLOCK DETECTED OUTPUT for a single pulse within the same byte time is necessary.

When the READ GATE goes low, signifying the end of a read operation, the PLL reference signal is switched back to the 2F CLOCK and the $\overline{\text{LOCK DETECTED}}$ OUTPUT goes high, causing the VCO gating circuitry within the PULSE GATE to be bypassed thus allowing PHASE and FREQUENCY comparisons to occur. The PLL also returns to the high tracking rate and the output signals return to their initial conditions.

If the chip is used as a data-synchronizer (on-chip decoding not necessary) for MFM or other popular RLL codes employing a 1010... preamble, the SYNCHRONIZED DATA OUTPUT and the VCO CLOCK OUTPUT should be used. External decoding can be accomplished either in commercially available controller chips or encoder/decoder circuits, or by the customer's proprietary design.

PHASE ONLY VS. PHASE-FREQUENCY COMPARISON OPERATION

As stated earlier, the function of the PLL is to maintain phase and frequency lock between the reference signal (2F CLOCK or ENCODED DATA) and the feedback signal (VCO). A comparator that performs only phase comparison is mandatory during read-mode in the data field in order to handle the non-periodic nature of various coding schemes. With this type of detector, the phase-locked-loop functions as a feedback loop in which it responds only to the phase differences between the input and the feedback waveforms. As long as the reference and VCO singals have their edges aligned (are in phase lock), the PLL is insensitive to their frequency relationship.

During the non-read mode the PLL is required to lock onto the 2F CLOCK, a specific frequency reference that is close to the data rate. If a disturbance is somehow introduced in the system which results in cycle slipping or prolonged transient behavior of the reference clock, false lock may occur if a PHASE-ONLY comparator is being used. Similarly, in the preamble field during read mode, the PLL tries to lock onto a periodic pattern. If pulse-pairing occurs in this PLL synchronization field due to asymmetry in disk drive electronics, quadrature lock may result if a PHASE-ONLY comparator is being used. Under these circumstances PHASE comparison alone may be inadequate, since it discriminates only phase and not frequency information. A PHASE-FREQUENCY-COMPARATOR, therefore, is recommended during these modes of operation whenever false lock presents a potential problem. The DP8461 implements such a comparator. It performs identically to the PHASE-COMPARATOR in the case when both inputs to the comparator have the same frequency; however, if the inputs exhibit the slightest frequency offset, the PHASE-FREQUENCY-COMPARATOR also provides a frequency-sensitive error correction signal to ensure frequency acquisition.

As mentioned in the device description, the DP8461 requires a "Qualified Read-Gate" for proper operation in soft-sectored environments. This is necessary to accommodate phase-frequency comparison into the preamble field. If the READ GATE is allowed to be asserted randomly, it might be asserted in the data field or in the write-splice area. With the DP8461, prior to preamble detected, the PLL is operating in the phase-frequency mode. If it encounters a low frequency pattern in the data field, the VCO will try to lock onto it and thus shift its frequency. Similarly, if READ GATE becomes active in a write-splice area, the PLL may be pushed to either of its limiting frequency excursions. By employing a "Qualified Read-Gate" with the DP8461, the READ GATE will always be asserted over a repetitive 1010... pattern and thus avoid any of these problems.

PULSE GATE

The PULSE GATE has two important functions. It ensures a continuous PLL lock in the presence of random patterns encountered on the media and in the bit stream. It also provides a precise time delay (independent of process and

external component variations) necessary to align the incoming data with the center of the decoding window. The delay is exactly one-half the period of the 2F CLOCK and the delay generator is referenced to the 2F CLOCK. This allows input bit jitter up to ±one-half the 2F CLOCK period.

The PULSE GATE incorporated in the DP8461 has two multiplexers which allow the circuit to switch from PHASE-FREQUENCY COMPARISON to PHASE-ONLY COMPARISON when the LOCK DETECTED signal becomes active (Low). *Figure 2* is a block diagram of the PULSE GATE which details how this is accomplished. When both the INTERNAL READ GATE and the INTERNAL LOCK DETECTED are inactive (non-read mode) the 2F CLOCK DIVIDED BY TWO and the VCO DIVIDED BY TWO signals are selected by MULTIPLEXER-1 and MULTIPLEXER-2 respectively. In this configuration phase and frequency comparisons are made between them and the possibility for a false lock occurrence is eliminated. When the INTERNAL READ GATE is active while the INTERNAL LOCK DETECTED remains inactive (read-mode, preamble detection) the ENCODED DATA and the VCO DIVIDED BY TWO signals are selected by the multiplexers. Again, PHASE-FREQUENCY COMPARISONS continue to be performed to ensure the PLL locks exactly to the data rate frequency. After sixteen pulses of consecutive preamble pattern are detected, the INTERNAL LOCK DETECTED line becomes active. MULTIPLEXER-2, under the control of the INTERNAL LOCK DETECTED singal, then switches from the VCO DIVIDED BY TWO to the GATED VCO signal. Through the circuit configured by the D-type flip-flops and the OR gate, the comparator effectively performs PHASE-ONLY comparisons (an INTERNAL VCO pulse is allowed to the input of MUX-2 only when an ENCODED DATA pulse is sensed). Thus, the DP8461 guarantees proper frequency lock of the VCO to the 2F REFERENCE CLOCK during the non-read mode and to the preamble synchronization pattern during the read mode. The circuit performs the necessary phase-only comparison in the data field during read mode operation.

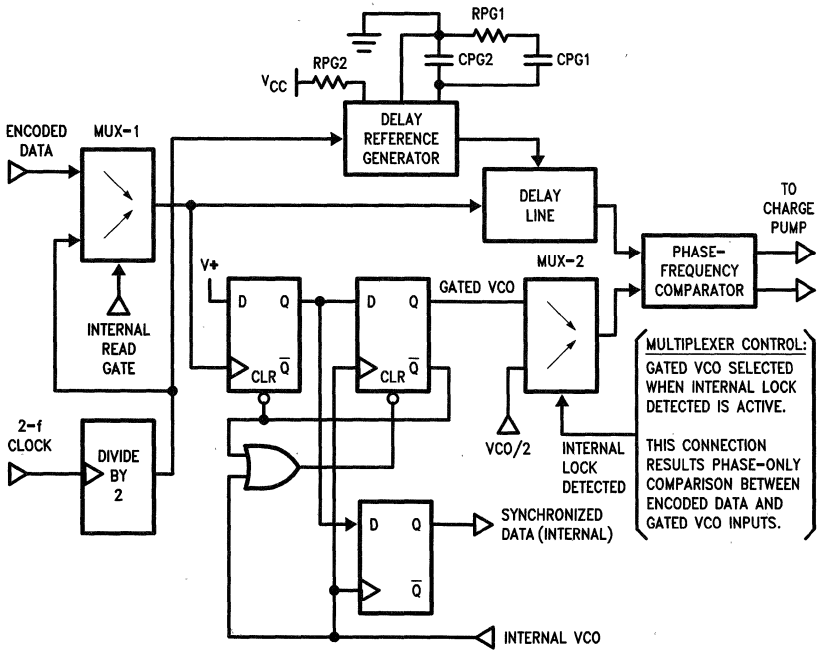
DATA SEPARATOR APPLICATION PROBLEMS

Following are some common application problems for many data separator circuit designs. The purpose of this application note is to identify these problems and to propose simple solutions thus enabling our DP8461 users to avoid these potential application problems.

FORMS OF FALSE LOCK

Two types of pseudo-lock can typically occur in a PLL within a disk drive system. A periodic input waveform must be present, such as a disk synchronization field, in conjunction with the suppression of frequency information (pulse-gate type PLL) in order for either to occur.

Pulse Gate Block Diagram



TL/F/8599–2

FIGURE 2

I. The first is herein termed simply "false lock", or more accurately, **fractional harmonic** lock. This occurs when the PLL is disturbed, forcing the VCO outside of the capture range (determined by the loop bandwidth; typically ±2% of the data rate) of the PLL for a period of time. The PLL is then able to acheive a pseudo-lock if (1) the ratio of the VCO frequency vs. the input frequency is an integer fraction, such as 5:6 or 6:5, and (2) the difference frequency between the VCO and the input is greater than the PLL capture range. Ideally, the error signal generated at the VCO control input, which is at the "beat" (difference) frequency of the two signals, would correct the false lock. However, this error signal is suppressed because it lies outside the frequency range of the low pass loop filter. The loop will, however, produce a self-sustaining error signal and thus will remain on the false lock null.

II. The second form of pseudo-lock is called **quadrature lock**. In this case, the PLL is able to lock to the correct frequency, but is caught on a narrow phase null which is positioned 90 degrees (w. r. t. the NRZ data period) from nominal. This phase null can occur only when there exists a pairing of periodic disk data pulses which originates in the disk drive itself (see *Figure 3*). The quadrature lock is perpetuated because the net error signal generated at the VCO input by the displaced, complimentary pulses lies well outside of the loop bandwidth and is averaged to a self-sustaining correction signal.

LOSS OF LOCK DURING READ MODE

In some systems the controller asserts the READ GATE randomly along a formatted track. If the READ GATE is asserted over a write splice, which usually contains unintelligible information, the PLL might false lock to some harmonic of the data or it might be pushed to either extreme of its allowed frequency swing. Similarly, when the READ GATE is asserted over a data field the PLL might lock to a harmonic of the data.

This problem can be completely avoided with the DP8461, which is used in conjunction with a "Qualified Read-Gate" technique. As an example, a good PLL controller algorithm that only allows assertion of the READ GATE over a preamble or similar high frequency pattern is listed below.

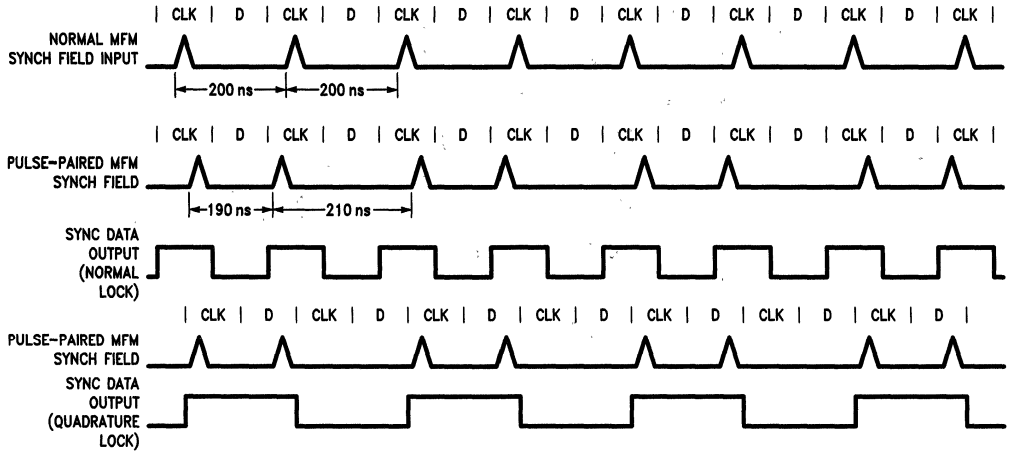
1) Deassert READ GATE—allow a 4 byte time minimum for the PLL to lock to the 2F-REFERENCE CLOCK.
2) Wait for 2.5 bytes of valid preamble pattern.
3) Assert READ GATE.
4) If valid preamble continues for 5 or more bytes then go to 5; otherwise go to 1.
5) "LOCK DETECTED" becomes active, AM search begins.
6) If AM is found, then continue the read routine; otherwise go to 1.

FALSE LOCK IN THE NON-READ MODE

The DP8461 **has been specifically designed to eliminate the possibility of false lock during the non-read mode.** This is accomplished by the use of a phase-frequency comparator in the non-read mode as was described in the PULSE GATE section.

False lock during the non-read mode can occur by two means in systems using phase-only comparisons in the non-read mode. When the power supply of the PLL circuit is switched on for the first time, the VCO ramps toward the reference frequency. The acquisition process may lock the VCO to some harmonic of the 2F REFERENCE CLOCK if the bandwidth (capture range) is not high enough. False lock can also occur in the non-read mode after an aborted read operation as described above. If the VCO has either lost lock or has been driven far from its center frequency while trying to read, false lock might occur during relock to the crystal if the capture range is narrow.

Example shows 5 Mbit/sec MFM synchronization field.
Input data bit positions are indicated by peaks of pulses.



TL/F/8599–3

FIGURE 3. Timing Diagram of PLL Quadrature Lock within a Symetrically Pulse-Paired Synch Field

QUADRATURE LOCK

The DP8461 **has been specifically designed to eliminate the possibility of both quadrature lock and harmonic lock in the preamble field during read mode** by extending the phase-frequency comparison technique during preamble detection in the read mode until LOCK DETECTED occurs.

Within the standard synchronization field which precedes the data field, bits are recorded at a constant frequency for a time sufficient to allow the PLL to acquire lock. With normal recording and read-circuit behavior, this synchronization information reaches the PLL as a continuous, periodic data stream. In some disk drives, if an offset has somehow been induced into the recorded information, or if read-channel asymmetry exists within the drive electronics which skews the flux reversal zero-crossing point, the synchronization field waveform which reaches the PLL may appear in the form of periodic pulse-pairs. This condition only arises when a repetitive pattern is present, giving rise to the possibility of quadrature lock. Note that quadrature lock is actually more prone to occur within systems where a low-noise design has minimized the randomizing effect which noise has on bit position.

MINIMUM PULSE WIDTH REQUIREMENT

The DP8461, as with other members of the DP8460 family of data synchronizers, has a minimum pulse width requirement on the ENCODED DATA input for proper operation. As there is no uniform pulse width specification on "Raw Read Data" outputs from disk and tape drive manufacturers, it has been found that certain drive systems output too narrow a pulse width for the DP8460 family of circuits to accept. Our recommended minimum positive pulse width is 6 ns and the minimum negative pulse width is 80% of the VCO period; this allows a maximum positive pulse width of 120% of the VCO period. Some drives utilize a bidirectional one-shot to shape the read data output pulse. The output pulse width from such drives can be readily readjusted from an RC timing network to attain acceptable minimum pulse width requirements for the PLL circuits.

Note: The chip is particularly sensitive to inadequately filtered switching supply noise.

SUMMARY

The DP8461 is another one of National's second generation single chip high performance PLL circuits for application in disk memory systems. This device features a comparator with both phase-frequency and phase-only comparison capabilities. Additionally, the PHASE-FREQUENCY COMPARISON circuit in the DP8461 has been designed to allow its operation in the preamble field during read mode so that employing it with a "Qualified Read-Gate" will eliminate all potential false and quadrature lock problems associated with soft-sectored systems. The DP8461 offers significant savings of cost and time in production, test, and maintenance since only a few fixed passive components are required for operation. The need to trim any external components has been eliminated and, since no external components determine window accuracy, the performance will not be sensitive to external variations. The chip requires a single +5 volt supply and it is housed in a narrow 24-pin dual-in-line package (also available in 28 pin PCC package). The DP8461 has the same pinout as the DP8460 and the DP8465, and thus, can be used in their designed applications provided the READ GATE has been qualified. The DP8461 can be used as a data synchronizer for MFM or any of the existing RLL codes employing a 1010. . . preamble, or as a data separator for MFM.

For further information, the reader should also refer to the National Semiconductor Application Note 414, Precautions for Disk Data Separator (PLL) Designs.

Designing with the DP8462

National Semiconductor
Application Note 494
Kern Wong

GENERAL DESCRIPTION

This literature assumes the reader has thoroughly read the DP8462 datasheet and is familiar with the basic circuit operation discussed therein. The objective of this note is to give the customer comprehensive application information for the design and testing of the device. It also offers detailed guidance in the selection of appropriate component values, use of optional device features, and considerations for system configuration. The following is a brief description of key features of the device.

The DP8462 is one of National Semiconductor's second generation disc data synchronizers designed for application in disc drive memory storage systems. It features 3T (1-0-0) or 4T (1-0-0-0) preamble detection circuitry which makes it especially convenient for systems employing run-length-limited codes such as the popular 2, 7 code. Of course, it may also be used as a data synchronizer for other conventional coding methods such as MFM, (1, 7), (1, 8), etc., but the internal (3T/4T) preamble detector function will then be bypassed.

This Phase-Locked-Loop (PLL) chip contains three customer programmable input control pins (TLL logic levels) to offer significant design flexibility. The PREAMBLE SELECT programming input determines whether the preamble detector looks for a 3T or 4T preamble pattern. The 4T preamble has been the most widely chosen pattern because of its ability to attain phase-sync naturally (refer to DP8463B (2, 7) ENDEC datasheet for detailed discussion). The LOCK-CONTROL input pin can be set to allow the phase comparator to function in one of two modes of operation: Phase and Frequency comparison or Phase-Only comparison. During non-read mode, phase and frequency comparison is employed within the DP8462 to lock onto the 2F reference clock. However, in the read mode the phase-frequency detection circuitry changes to phase-only comparison. Via the LOCK CONTROL input, the user may select when this mode change occurs, either immediately after Read Gate is asserted or after preamble detection has occurred. The charge pump also has two modes of operation: high track rate is used for fast acquisition to the 2F reference clock in the non-read mode and to the preamble in the read mode; low track rate is used in the read mode to allow smooth, stable lock to the data field. Both track rates are selectable by the customer via the I-BOOST ENABLE input. In a typical design the PREAMBLE DETECTED output may be directly connected to the I-BOOST ENABLE input to accomplish the desired track rate switch-over.

PHASE-ONLY OR PHASE-FREQUENCY COMPARISON

Depending on system characteristics and sector formats employed, such as soft or hard sectoring, encoding method, acquisition time requirement, etc., the system designers may be required to select only one of the two modes of comparison during read (preamble acquisition) or they may be free to choose either comparison mode to achieve optimum system performance. The following discussion illustrates the differences between the harmonic phase detector and the non-harmonic phase detector techniques incorporated in the DP8462 PLL chip.

In the non-read mode the PLL is always set to Phase-Frequency comparison mode (non-harmonic) to guarantee that it will attain frequency lock to the 2F Clock reference signal; thus, the possibility of a false lock occurrence is eliminated. In the read mode, if the Phase-Only comparison mode is chosen (via LOCK CTL: L0), then the phase detector will engage in phase comparison (harmonic mode) between ENCODED DATA and VCO gated by data pulses immediately upon Read Gate assertion. This must be done in a true soft-sectored system where there is no guarantee of Read Gate assertion occurring only within preamble field. Furthermore, the maximum phase step seen by the PLL at the assertion of Read Gate is at most one-half of one decode window period since the Pulse Gate allows the incoming data pulse to be compared with the nearest occurring VCO pulse. However, there is always a probabilistic condition that an initial burst of pulses can occur at the input during Read Gate switching which may violate normal coding methods or nominal pulse width specifications, resulting in synchronization errors. To avoid false lock possibility with Phase-Only mode in synchronization field when Lock Control is set "LO", it is recommended that the loop's natural frequency bandwidth, Wn, be low. Normally the loop filter component values for C1 and R1 will need to be recalculated in accordance with the lower Wn value selected. Optimizing the selection of the Pulse Gate network (RPG2 and RPG4) may also improve the loop's performance as this allows better centering of the data bit position with respect to its decode window and optimizes accurate locking to the data frequency.

The customer may also configure a soft-sectored write, and pseudo-hard sectored read system, which retains the advantages of formatting flexibility while capitalizing on the Phase-Frequency comparison feature in the read mode (with LOCK CTL: HI). Such a system configuration is realized with a Read Gate qualifying technique which ensures Read Gate is asserted only within the preamble field, thus removing any lock problems associated with reading through write-splice gaps or data field induced false address mark (AM) or false preamble patterns. With this method of operation, the PLL remains in the Phase-Frequency comparison mode in the preamble field until 16 consecutive preamble pulses have been successfully detected; then, it switches to Phase-Only mode. Employing a Phase-Frequency comparator during preamble acquisition ensures that the possibility of Quadrature Lock, a form of false lock due to pulse pairing in the read channel, is eliminated. Similarly, a true hard-sectored system will utilize these same techniques to avoid all the potential lock problems mentioned above.

The user should be aware of the fact that when employing frequency lock, the VCO divided by either 3 or 4 is compared with the data pulse. Therefore, depending on the initial logic state of the divider (which is random) and the initial phase difference incurred at the time of Read Gate assertion, the maximum phase error can be 3 or 4 VCO cycles long (depending on whether 3T or 4T preamble is selected) which is 6 or 8 times greater than when the Phase-Only mode is employed. This requires the PLL synchronization

bandwidth to be set adequately high in order to guarantee the phase error has settled to less than one-half of one decode window before the PLL switches out of frequency lock mode. Otherwise, a second phase step is introduced when the device switches out of frequency lock mode which lengthens lock times and may result in synchronization errors.

BANDWIDTH CONSIDERATIONS AND LOOP FILTER SELECTION

The DP8462 PLL is a frequency selective feedback circuit that can synchronize with a selected input signal and track the frequency and/or phase changes associated with it. A finite phase error is necessary to generate a corrective voltage to shift the VCO output frequency in compliance with the input signal variation. Once locked, the synchronizer can track a slowly changing input signal. It rejects transient changes (via loop filter) which tend to disturb the PLL. The phase detector (a bi-directional charge pump) generates a proportional error correction current pulse and is followed by a passive low-pass filter network generally with a sufficiently narrow bandwidth to suppress any noise and high frequency components from the phase-detector and charge pump stages. More importantly, the filter network determines the dynamic performance of the loop which includes capture and lock ranges, bandwidth, and transient response time. Of equal significance, the passive filter also yields the characteristic poles and zeroes for stability. There are numerous approaches to calculate a suitable set of filter component values; they may be estimated from a capture range point of view, from the required time and dynamic response profile of the loop, or from noise bandwidth considerations, etc.

LOOP FILTER COMPONENT VALUE SELECTION

The first step when selecting component values is to determine an appropriate value for the natural angular frequency, Wn, of the PLL. (Wn, commonly referred to as the loop's natural frequency should not be confused with the center frequency, Wo or Fo, of the VCO.) Wn is related to the loop gain and the RC time constants imposed by the low-pass loop filter. It should be noted that if the available loop gain is high and a relatively low natural frequency required, very large time constants will result. The size or value of the passive components for the loop filter may become impractical if not difficult to obtain. Therefore, judicious choice of Wn and filter components is necessary to allow good performance while at the same time using only standard component values. Optimum filter design may involve an iterative engineering effort from empirical data to arrive at the desired results. Recommended values in the datasheet are all valid suggestions but may not be the optimized values for every system.

As suggested in the datasheet, a (degenerated) 2nd order passive filter is quite adequate for most applications. The filter network consists of C1 connected in series with R1 and C2 connected in parallel with R1 and C1. The characteristic equation is 3rd order by inclusion of C2 in the highest order term. In practice, if the pole determined by R1 and C2 is more than a decade above the zero, the 3rd order term can be ignored and the equation can be rewritten as a 2nd order loop (refer to the DP8462 datasheet for details).

Two approaches for the determination of Wn are illustrated; component calculations are the same in each case. These examples allow digital designers to start estimating a loop filter without having to derive its poles/zero, and centroid, or construct Bode plots.

Example #1: 10 Mb/s data rate, (2, 7) encoding, 4T preamble

Assume it is desired to track the information recorded on a rotating or travelling medium being modulated by the spindle or capstan velocity, and assume there will be a 1.5% offset in the motor speed. It has been shown that a capture range greater than the input frequency offset, ΔW, will ensure that lock will occur quickly. Choose Rrate = 820Ω (an optimal value for the charge pump).

The initial frequency offset is then:

$$\Delta W = (1.5\%)\,(10\text{ Mb/s})\,(2\pi) = 942\text{ krad/s}$$

In most system designs, an optimally flat frequency-transfer function of the loop is the objective. The Bode diagram of such a transfer characteristic shows this can be achieved with $\zeta = 0.707$ (see reference: F. Gardner). The expression of capture range $\cong 2\,\zeta Wn$ is a reasonable approximation also found in reference materials (i.e., R. Best). A damping of $\zeta = 0.7$ is chosen as a design criterion for optimal response during the 4T pattern acquisition.

By equating capture range and ΔW for acquisition of 4T data pattern, (that is, one data pulse present within 4 VCO periods):

$2\,\zeta Wn\,(4T) \geq \Delta W$

$Wn\,(4T) \geq \Delta W/(2\zeta) = (942\text{ krad/s})/(2)\,(0.7) = 666\text{ krad/s}$

This is defined as Wn(max) during preamble or Xtal acquisition, where the PLL is locking onto the 4T preamble or the Xtal divided by 4 signal.

Wn is proportional to the square root of the data frequency (Refer to the datasheet:

$Wn = \sqrt{(2.5)\,(Fvco)/(C1)\,(R)\,(N)}$,

where N is defined as Fvco/Fdata (i.e., N = 8/1). R is the effective gain setting resistor which determines the charge pump reference current.)

For minimum data frequency tracking, for example the 8T pattern (which is the lowest allowed data frequency for the 2, 7 RLL code where one data pulse occurs within an 8 VCO clock period span),

$Wn(min) \equiv Wn(8T) = \sqrt{4T/8T}\,[Wn(4T)] = 471\text{ krad/s}$

Similarly at maximum data frequency, as in a 3T pattern, Wn(max) in data field becomes 769 krad/s.

To calculate for the loop filter component values, the above general expression for Wn is transposed to obtain the following equation:

1. $C1 = (2.5)(Fvco)/[(Rrate)(N)(Wn(4T))^2]$
   $= (2.5)(Fvco)/[(820)(4)(666E3)^2]$
   $= 0.034E\text{-}6$
   C1 = 0.03 μF, which is a standard capacitor value.
2. $R1 = 2\,\zeta/(C1)(Wn) = 2[\zeta(4T)]/(C1)[Wn(4T)]$
   $= 2(0.7)/(0.03E-6)(666E3)$
   $= 70.07$
   R1 = 68Ω, is a standard resistor value.
3. Choose C2 ≅ C1/50. In general, selecting C1/50 < C2 < C1/10 will yield non-dominant parasitic pole and still allows C2 to integrate the current pulses. (Capacitor may be film or monolithic ceramic.)
   C2 = 560 pF

4. Recalculate Wn(8T) at minimum data frequency with standard component values: (For loop stability ζ(min) should not be allowed to be less than 0.5. ζ(min) occurs when Wn is at a minimum.)

$$Wn(8T) = \sqrt{(2.5)(20E6)/(820)(8)(0.03E-6)}$$
$$= 504 \text{ krad/s}$$
$$\zeta(8T) = (504E3)(68)(0.03E-6)/2$$
$$= 0.51$$

5. Recalculate Wn(4T) for data frequency in preamble acquisition with standard component values:

$$Wn(4T) = \sqrt{(2.5)(20E6)/(820)(4)(0.03E-6)}$$
$$= 713 \text{ krad/s}$$
$$\zeta(4T) = (713E3)\ (68)\ (0.03E-6)/2$$
$$= 0.73$$

6. Calculate the loop parameters for locking to the crystal frequency divided by 4 signal (N=4). Normally a higher loop gain is employed. With the use of an additional charge pump current setting resistor (Rb) to boost loop gain during non-read mode, the damping factor may be chosen with $\zeta(4T-Xtal) \cong 1.0$ for critical damping during crystal acquisition.

$$Wn(4T-Xtal) = \zeta(4T-Xtal)(2)/(R1)(C1)$$
$$= (1.0)(2)/(68)(0.03E-6)$$
$$= 980 \text{ krad/s}$$

7. Calculate Rb from Wn(4T−Xtal):

$$Wn(4T-Xtal) = \sqrt{(2.5)(Fvco)/(Rr//Rb)(C1)(N)}$$
$$980E3 = \sqrt{(2.5)(20E6)/(820//Rb)(0.03E-6)(4)}$$
$$820//Rb = (2.5)(20E6)/(0.03E-6)(4)(980E3)^2$$
$$= 433.8$$
$$= (820)(Rb)/(820 + Rb)$$
$$Rb = 921.2$$

choose Rb = 910Ω, a standard resistor value.

Rp = 820//910 = 431.33, the effective charge pump current setting resistor value.

8. Recalculate Wn(4T−Xtal) with standard component values:

$$Wn(4T-Xtal) = \sqrt{(2.5)(20E6)/(431.33)(0.03E-6)(4)}$$
$$= 983 \text{ krad/s}$$
$$\zeta(4T-Xtal) = Wn(4T-Xtal)(Rp)(C1)/2 = 1.02$$

In general the customer may choose any suitable gain ratio between fast lock and stable data tracking. A 2:1 ratio is a popular choice; for example, Rr = Rb = 820Ω. In the above example this will make the damping factor and Wn(max) at crystal reference lock slightly larger. Note that we have chosen zeta, ζ, to be between 0.5 and 1.0 as a general guideline such that the system will not be excessively underdamped nor subject to excessive acquisition times. However, an experienced designer can choose other appropriate design values to tailor toward a specific system's requirements.

Note: There is a minimum value for Rr//Rb (parallel combination value) allowed for the DP8462. A maximum reference current limit for the charge pump is imposed such that Vbe/(Rr//Rb) ≤ 2 mA. This translates to Rr//Rb ≥ Vbe/2 mA ≅ 350Ω, typically, for Vbe ≡ 0.7V.

Hence, the above calculations yield the loop filter component values and their corresponding loop parameters for example #1 as follows:

10 Mb/s (2,7) Code	Rr	Rb	R1	C1	C2
	820Ω	910Ω	68Ω	0.03 μF	560 pF

Wn(8T)	ζ(8T)	Wn(4T)	ζ(4T)	Wn(4T−Xtal)	ζ(4T−Xtal)
504 krad/s	0.51	713 krad/s	0.73	983 krad/s	1.02

Another frequently used method for Wn determination is by graphical means. If damping ratio, ζ, and lock time are given, Wn can be arrived at via the normalized phase error versus Wnt plots. Alternatively, if the desired system response has been defined as specified in the example given below, Wn can also be readily determined.

Example #2: 7.5 Mb/s data rate with (2, 7) encoding and 4T preamble

Maximum peak overshoot $< 30\%$

Settling time $\cong$ 11 μs with phase error $\leq \pm 5\%$

VCO capture range within $\pm 20\%$ of 15 MHz and follow excursion of $\pm 5\%$ of Fo

Charge pump gain ratio = 2:1 between non-read and read modes, (Rr = Rb = 820Ω).

It is helpful to know the initial frequency offset for the derivation of Wn. But this information may be available to the customer and the exact value is not necessary to make a trial estimate for Wn. Fortunately, the maximum velocity variation of commercial drive mechanisms are well within 2% and current disc drives have guaranteed rotational tolerance of less than 1%. Hence, initial frequency step between non-read and read modes in normal operation does not contribute appreciably to the total phase-error transient resulting from a maximum phase step assumed in the calculation.

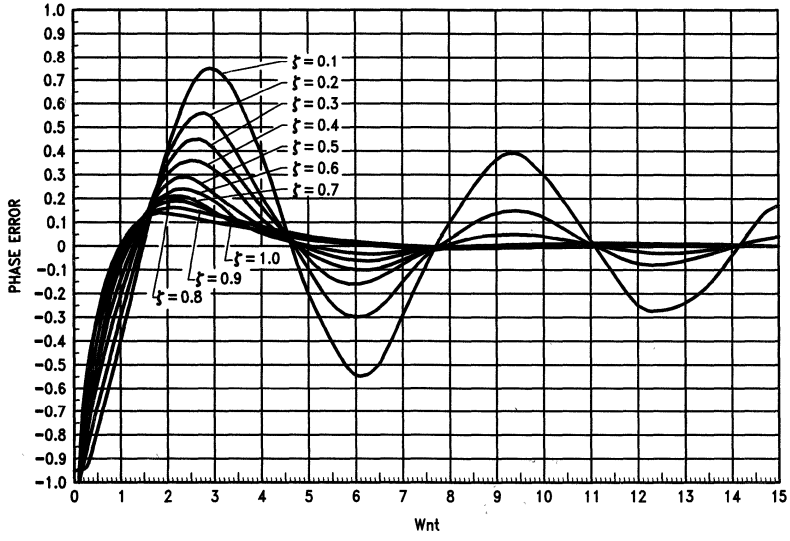
Wn can be derived from the Wnt vs. $\theta e(t)$ curves due to a phase step. *Figure 1* shows that the $\zeta = 0.5$ parametric curve meets the above requirement of maximum peak overshoot less than 30%. The residual phase error will be approximately -5% at Wnt = 6 rad. Therefore, settling to $\pm 5\%$ (or about 3 ns) in t = 11 μs yields:

Wnt = 6.0 rad

Wn(4T) = 6.0 rad/11 μs = 545E3 rad/s

This is the computed BW for preamble acquisition, which is then used to calculate the loop filter component values as in the previous example. The loop filter and parameter values are presented below:

Transient Response of 2nd Order Loop due to a Step in Phase



TL/F/9331–1

FIGURE 1

7.5 Mb/s	Rr	Rb	R1	C1	C2
(2, 7) Code	820Ω	820Ω	68Ω	0.039 μF	820 pF
Wn(8T)	ζ(8T)	Wn(4T)	ζ(4T)	Wn(4T−Xtal)	ζ(4T−Xtal)
383 krad/s	0.51	541 krad/s	0.72	766 krad/s	1.01

Note: The derived Wn value may be checked to verify for adequate capture range in the read mode. The value chosen does satisfy the expression $2\zeta(4T)Wn(4T) > \Delta W$(of 1.5%).

The above techniques are useful in estimating suitable loop filters which satisfy basic capture and lock conditions in non-read mode and read mode in preamble field. The conservatively derived component values should allow good performance in most applications. However, there are situations where significant improvement in system performance may be gained if the loop bandwidth is optimized with respect to a particular system. For example, to accommodate certain sensitivities to cylinder formats, media imperfections, or noise in gaps, etc., it may be necessary to raise or lower Wn considerably from the suggested values used in the above computations. In addition to the NSC suggested values in the datasheet and this note, the customer is strongly encouraged to modify those values or tailor a filter to achieve minimum error-rate.

The following table lists additional sets of loop filter component values which have performed well in the field or have been designed into systems:

LOOP BANDWIDTH OPTIMIZATION AND DESIGN CONSIDERATIONS

Numerous "optimized" loops have long been derived for various types and orders of PLL systems. Optimum loop transfer functions were formulated for different types of input stimuli. It is practically impossible to make a perfect loop as this will require continuous adaptation of the filter to compensate for the input changes as a function of time. The 2nd order loop (due to its ease of analysis) is by far the most popular and of the greatest interest to design engineers. The following list presents some design trade-off issues and several frequently encountered design and device evaluation considerations.

1. Narrow Bandwidth: Most designers will wish to keep loop BW low to improve noise rejection. This will decrease the loop's response to input jitter at the expense of having a smaller capture range. This can be very helpful in soft-sectored reading to desensitize the effect of write-splice induced perturbations.

2. Wide Bandwidth: If capture range exceeds frequency range within which the input signal is expected to vary, it increases the probability of spontaneous lock. Raising the loop's BW is recommended if the input frequency offset is sufficiently large.

3. Damping Factor: It is suggested that this should be set approximately equal to:

   0.5 for Wn(min) in read mode,
   0.7 for Wn(max) in read mode, and
   1.0 for Wn(Xtal) in non-read mode.

   (If a 4T preamble is employed, then Wn(max) in data field of 3T pattern is larger and the damping factor becomes slightly greater than 0.7.)

   A larger damping factor increases capture range which improves the probability of acquiring lock. It also increases total acquisition time.

   A smaller damping factor minimizes response to bit shift, but excessive underdamping will cause instability.

4. The open loop response of the chip can be modelled as a 2nd order PLL (highest power of s=2 in the denominator of the loop transfer function) with a parasitic 3rd pole.

$$G(s) = \frac{2.5(Fvco)\,(1+sR1C1)}{(N)(s^2)(Rr)(C1)(1+sR1C2)}$$

   where Fvco = center frequency and N = Fvco/Fdata.

   If the chosen physical parameters and component values are faithfully modelled by such an expression, phase-margin of the loop can be reasonably predicted. Phase-margin should be between 30° and 70° at unity gain. 45° is good compromise in general; this is the criterion for loop stability. Larger phase-margin produces a more stable loop, but it slows the response, increases VCO output unwanted sidebands, and reduces loop VCO-noise suppression capability.

TABLE I. Loop Filter Component Values

Data Rate (NRZ)	Rr	Rb	R1	C1	C2
5 Mb/s	820Ω	1.5 KΩ	75Ω	0.056 μF	820 pF
	820	2.2K	95	0.047	820
	370	∞	42	0.10	1600
7.5 Mb/s	820Ω	820Ω	62Ω	0.047 μF	1000 pF
10 Mb/s	820Ω	1.2 KΩ	82Ω	0.020 μF	510 pF
	820	820	62	0.039	820
	910	910	51	0.056	820

5. The effects of the loop filter component values can be summarized as follows for component optimization considerations:

   Increasing C1: (Lowers BW) desensitizes PLL from input bit jitter or frequency shift.

   Reduces possibility of spontaneous transient response to spurious signals in read/write gaps. Slows down loop response time. Reduces capture range.

   Increases loop sensitivity to VCO noise.

   Increasing C2: Reduces the rippling effect of the charge pump switching current which allows less loop response to shifted bits. Reduces stability margin.

   Increases (slightly) loop response to VCO noise. Increases response to write splice perturbation.

   Increasing R1: Increases stability margin. Decreases loop response to VCO noise.

6. An often asked question on PLL design parameters regards lock-up time. There is no simple expression that will adequately predict acquisition or lock times because of many factors that influence them and ambiguity of definition of lock. Lock time may be estimated roughly from "Wnt vs. phase-error" normalized curves if loop parameters are defined or known (see *Figure 1*). The following lists several factors that contribute most to the variation in lock times. This information offers some insight into lock time interaction with respect to the DP8462 PLL and its environment. Customer should be aware of them in order to realize good designs.

   a) Initial phase difference between input & VCO comparison pulses.

   b) Initial state of the VCO divided by 3 or 4 counter.

   c) Initial input frequency offset.

   d) Low pass loop filter characteristics (BW and damping).

   e) Center frequency (VCO) accuracy and symmetry.

   f) VCO noise and sideband signals.

   Note: Although customer has no control over a) and b), their impact can be minimized through loop filter design.

7. Another frequently asked question is how to measure lock ranges and capture ranges. A few methods are suggested below:

   Lock range is the maximum input frequency-ramp excursion (upper and lower bound) permissible while the loop still maintains lock. To measure lock range, apply a square wave pulse train of known frequency to the ENCODED DATA input of the PLL chip. Allow the PLL to be stably locked onto the periodic input waveform. While in read mode measure either the SYNCHRONIZED DATA or VCO output for good synchronization to the input as its frequency varies until lock is lost. It is recommended to set Read Gate low momentarily after each measurement.

   For the capture range determination, continuously cycle Read Gate as in above (always allow sufficient time for the PLL to lock to the 2F reference clock). Start with the PLL in lock, gradually increase the frequency of the input data stream until the PLL will not lock, then readjust the frequency towards the center frequency until locking is achieved again at a particular frequency. This frequency is the upper capture frequency. Repeat this procedure for the lower boundary to determine the lower capture frequency.

   Note: A noisy analog instrumentation tuning mechanism may produce undersirable transients that disturb the PLL and give false data. Always make very slow and gradual adjustments. Improper toggling of the Read Gate or insufficient deassertion time may result in lock problems during read mode.

OPTIMIZING THE DECODE WINDOW:

In the past, the Twindow A.C. specification in the datasheet was not well understood and has caused misinterpretation by some customers. This specification simply indicates the maximum truncation of the available half decode-window width (one side of nominal window). For a given data rate, an ideal window boundary is defined for the data bit. The entire (ideal) window width can rarely be made available due to noise, device tolerance, and process or other related physical imperfections. The PLL chip, therefore, contributes a total window loss equal to twice the Twindow time stated in the Twindow specification. Basically, there are two mechanisms that are responsible for this loss. First, is the dynamic uncertainty arising from the superposition or modulation of noise and phase movement from internal and external sources that effectively narrows the usable window size. The second, and dominant mechanism, is the deviation of chip produced window center from the theoretical center of the data window. This is caused by non-ideal on-chip device matching due to process related variations during device fabrication. This deviation is sometimes called "window-margin". Our Twindow specification, in effect, lumps all contributing factors of window jitter and displacement within the PLL data synchronizer into a single specification which corresponds to the amount of maximum half-window loss. This figure can be determined empirically using a specific set of external passive components and test conditions following complete PLL lock and stabilization. (This will be explained in the next section.)

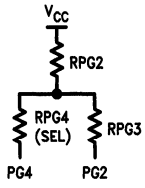
As suggested in the datasheet, bit jitter tolerance can be improved by adjusting the window center via the PG2 and PG4 pins. There have been no fewer than 3 different current splitting networks employed to adjust the internal silicon delay line to improve the phase margin. (The adjustments are internally compensated for Vcc and temperature variations.) The various network configurations, whether chosen for test convenience or for layout topolgy reasons, are fundamentally equivalent and achieve the necessary bias. They should yield the same final result of centering the data bits within their respective windows. It should be pointed out that we are by no means requiring the user to employ the extra task of trimming during manufacturing to accomplish margin improvements as may be misinterpreted from drawings depicted in the datasheet. Our intention is to illustrate to the users the fact that by selecting an appropriate resistor (fixed) value for RPG4, as in *Figures 2a* and *2b*, one can obtain better Twindow tolerance than specified in the datasheet. For example, the DP8462-3 part type guarantees at most 6 ns of static window loss at each window boundary. But, via an RPG4 of approximately 2.2k to 3.4 kΩ, the window loss can be significantly reduced by 60% or more.

We do see customers who must regain as much margin as possible from up-stream erosions that may be hard to control or correct. This is particularly the case for OEM customers whose electrical and/or mechanical recording compo-

nents in the read channel are usually off-the-shelf items and their integrated spec may not be as optimal as desired. To alleviate such constraint, an extra, but simple, step is added to perform a fine RPG4 selection. For 10 Mb/s operation, 1.5 ns or less of static window loss from each side (from a 50 ns window width) can be achieved with data bits perfectly centered around the expected data window. The graph in *Figure 3* is a plot (represents several lots of DP8462-3 units) of typical RPG4 values vs. the amount of bit shift allowable. The data rate is 10 Mb/s and (2, 7) code with 4T preamble is used. The chip operates under a specific set of operating conditions and support components. The curves show that the absolute value of the average decode window width is 47 ns (sum of T-early and T-late times for any RPG4 value shown.) The cross-over point indicates where T-early equals T-late. The RPG4 value at this point will allow the data bit to be symmetrically centered about its decoding window.

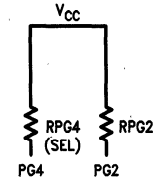
Note that only the phase-shift is being gauged, that is, the actual deviation of the data pulse under test from the theoretical center of the data window. Hence, the plot also indicates that the customer may "program" (via an external current source or resistors and switches) a wide range of bias conditions on PG4 pin to perform margin tests on the read channel for final test in manufacturing or for Early/Late strobing techniques used in certain data recovery procedures.

In general the RPG configuration for window deskewing as depicted by *Figure 2C* is preferred. RPG2 is fixed at 1.5 kΩ and RPG3 may be selected proximately between 300Ω to 500Ω. This configuration offers optimum component tracking between the RPG resistors.



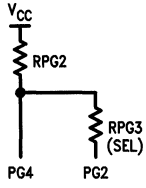
TL/F/9331–2

FIGURE 2a



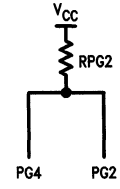
TL/F/9331–3

FIGURE 2b



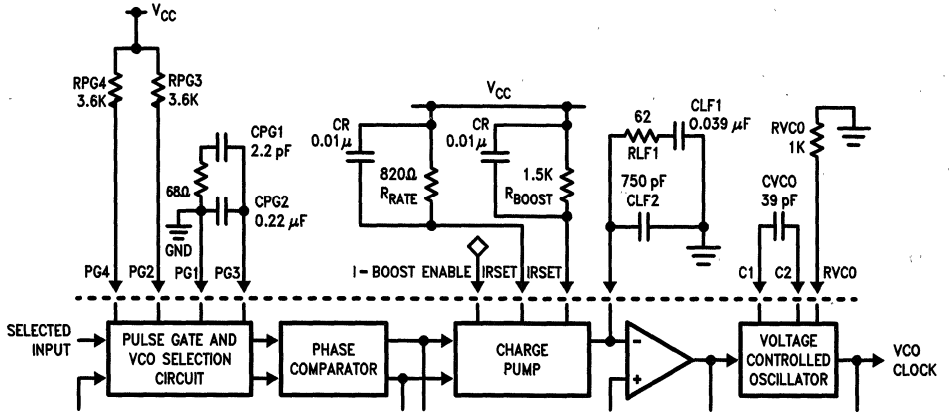
TL/F/9331–4

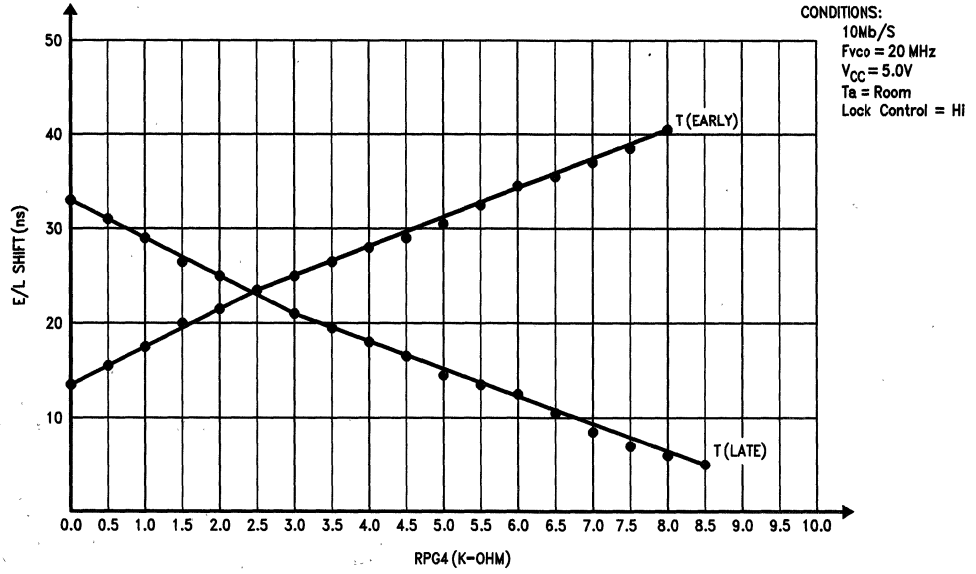
FIGURE 2c



TL/F/9331–5

FIGURE 2d





TL/F/9331–6

FIGURE 3. DP8462 Early/Late Window Shift vs RPG4

Note: There is a datasheet change to the DP8462-3 part type (Aug. 1, 1986) which pertains to the test method used to qualify the Twindow spec. The parametric value which appears in the AC Electrical Characteristics table of the datasheet has not been changed. For testing the window, the original configuration required the PG2 and PG4 pins to be tied together and connected to a 1.8 kΩ, RPG2, resistor to Vcc *(Figure 2d)*. This configuration is still used for testing the DP8462-4. For testing the DP8462-3, one 300Ω resistor will be palced between the PG2 and PG4 pins and PG4 will be tied to Vcc through a 1.5 kΩ resistor *(Figure 2c)*. Note 6 in the datasheet will now read:

"This specification is guaranteed only for the conditions under which the parts were tested. However, significant variation from formula is not expected for other data rates and filters. The DP8462-4 specification is for the condition when PG2 and PG4 are tied together. The DP8462-3 specification is for the condition when a 300Ω resistor is tied from PG2 to PG4 and a 1.5 kΩ resistor is tied from PG4 to Vcc. External adjustment can be used to optimize Twindow as described . . . "

PRINTED CIRCUIT BOARD LAYOUT GUIDELINES:

Other than a true Digital Phase Locked Loop (DPLL), any type of analog PLL is inherently a sensitive device. Hence, the environment in which it is operated should be optimized wherever possible to improve reliability. The following is a list of PCB layout recommendations that should help to minimize the effects of VCO jitter or mislock ocurrence that can be caused by various external sources of disturbance:

1. Establish a local Vcc island or net, separate from the main Vcc plane, to which the device and its associated passive components can be connected to remove unwanted noise coupling. Vcc supply filtering should be liberal and in very close proximity to the PLL chip. The device is sensitive to inadequately filtered switching supply noise. All lead lengths of the filter capacitors between Vcc and ground pins should be as short as possible to minimize lead inductance. Inclusion of a quality high-frequency capacitor, such as a 1000 pF silver mica capacitor, in parallel with a 0.1 μF ceramic capacitor is recommended.
2. Effective capacitive bypassing of the Rrate and Rboost pins (#2 and #3) directly to the Vcc pin is very important. Again, use quality high-frequency capacitors and maintain the shortest possible electrical lead length.
3. Use the main digital ground plane for all grounding associated with the device. The ground pins and the PG1 pin should tie directly to this plane.
4. Do not locate the chip in a region of the PC board where large ground plane currents are expected.
5. Locate all passive components associated with the chip as close to their respective device pins as possible.
6. Orient the chip's external passive components so as to minimize the length of the ground-return path between each component's ground plane tie point and the chip's ground pin.
   Note: Ground noise at the loop filter components, R1, C1, and C2 which is not identically present at the ground pin (common mode), is coupled through the filter components into the VCO control voltage pin.
7. Include no planing whatsover (Vccc or ground) directly between adjacent pins. This will minimize parasitic capacitance at each pin. Planing between the two pin rows, however, is recommended (directly beneath the package).
8. Avoid running signal traces between pins to avoid unwanted noise coupling.
9. Run no digital signal lines between or adjacent to the analog pins or signals traces (pins #1 to #7 and PG3) in order to avoid capacitive coupling of digital transients.
10. Minimize the total lead length of the Cvco capacitor. Inductance in the path degrades VCO performance, as does parasitic pin capacitance.
11. Do not place any bypass filtering at the Rvco pin (minor coupling of the VCO waveform into this pin is normal and acceptable).
12. Eliminate negative-going voltage transients (undershoot) at the digital input pins (pre-termination of driving lines may be necessary) to avoid drawing transient input-clamp-diode current from the device pins. Negative undershoot at the 2F Clock and the Encoded Data inputs should be removed to avoid excess VCO phase movement and window margin degradation. A damping resistor in series from the driving circuitry is recommended (typically 200Ω will be sufficient depending on the characteristic impedance of the path).
13. Minimize digital output loading (VCO, Phase Comparator Test, and Synchronized Data); i.e., if such outputs must drive large loads or long lines, employ buffers.
14. Allow unused digital output pins to float, unconnected to any net.
15. Avoid locating the chip within strong electromagnetic fields. If possible, choose the "quietest" region of the PC board.
16. If chip socketing is desired, use a low-profile, low mutual capacitance, low resistance, forced-insertion type (gold plated socket-strips are recommended). In general, avoid the use of "ZIP-DIP", zero-insertion-pressure sockets. If there is a need for them such as in batch testing of devices, the new low-profile "ZIP" sockets may still be used with acceptable results. The supporting passive components must be soldered directly under the socket pins with virtually zero lead length allowed.
17. Do not use wire-wrap interconnect, even in an evaluation set-up. Point-to-point wiring is acceptable, but the prototype board must have (or improvise) a decent ground plane or strips and Vcc nets.
18. Make allowance for pin-to-pin capacitance when determining Cvco (typically 4 pF to 5 pF) from datasheet formula.
19. When using PCC (plastic chip carrier) package, space for passive component connection to the device will be considerably tighter. It is acceptable to have axial-lead resistors (not capacitors) standing upright, but, the shorter lead must be connected to the device pins to obviate noise induction to sensitive nodes.
20. In multi-layer PCB layout, as in miniature board designs for small drive systems, do not allow any power planes to run into the device region to prevent introducing unwanted noise into the chip.
21. The crystal oscillators may be designed with CMOS inverters such as the 74HC04. If the device gain is too high for the circuit, proper oscillation may not occur. As an alternative, use National's unbuffered 74HCU04 instead (which may require different capacitor values for the oscillator).

EVALUATION BOARD FOR THE DP8462

An example of a good PCB layout is portrayed by the artwork for the DP8462 evaluation PC board in *Figure 4*. This hardware simplifies the task wherein customers evaluate and correlate devices. This evaluation/demonstration board is a compact, convenient, self-contained test set-up with the following features:

— Housed in a compact, pocket-sized PCB

— On-Board programmable data pattern generator

— Self-contained movable-bit generator with continuous adjustable bit delay time and pulse width. Operates to 25 Mb/s data rate.

— Two (2) independent 2F frequency oscillator sources on board for 2F Reference Clock and for clocking the data pulses.

— Four (4) selectable Early/Late strobe settings for window shifting tests; one range is made variable for fine tuning experiments.

— Programmable Lock-Control and Preamble-Select (3T/4T) selections.

— Reference bit and sync-bit pulse output for convenient measurement and triggering needs.

— Direct outputs to ratio counter for monitoring during window margin tests.

First, within the following discussion the method used by National for window testing (manufacturing) will be explained. Secondly, it will be shown how this compact hardware has the ability to perform similar evaluations in a laboratory bench set-up.

WINDOW TEST PHILOSOPHY

The test method used to determine the data synchronizer's tolerance to bit shift is explained below. Test procedures and test hardware have been developed to measure the device A.C. performance. These can easily be ascertained and correlated by any customer regardless of their application specifics. In customers' applications, signal sources may be quite different; they can range from magnetic or optical media, to transmission lines (as in token rings). In each case, the method of pulse detection requires different techniques. Therefore, in order for every customer to be able to agree upon a representative figure of merit, we stipulate a "Static Window Margin Test" method. Consequently, for correlation purposes, the window margin test of the PLL chip must not be tested in the system environment. Furthermore, the input signals to the PLL should be supplied from a stable word generator or equivalent source and not from the read outputs of any disc or tape drive system (the PLL chip, not the signal source, is under investigation). In addition, a fixed set of passive components is used for each data rate of interest. In this manner, a controlled condition is established which removes any ambiguity; correlation of Twindow data can thus be performed confidently.

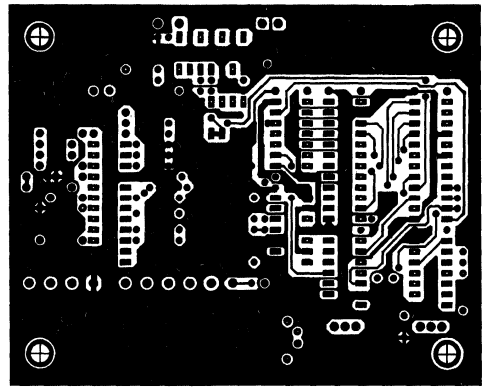
Refer to the timing and test sequence flow diagrams in *Figures 5* and *6*. The A.C. screening test is as follows: The device under test is first powered-on and set in the non-read mode. It will remain in this mode for 200 VCO cycles to allow the PLL to acquire a stable lock to a crystal 2F reference source. The device is then switched to the read mode by asserting the Read Gate input to a logic high level. This causes the internal input multiplexer to switch the PLL input from the crystal 2F reference source to the encoded preamble data pattern which is asynchronously derived from a signal pattern generator. (The preamble pattern is a string of 1-0-0-0 . . . , 4T preamble.) The PLL usually can lock onto the encoded data in less than two bytes of sync pattern. When 16 consecutive preamble bits are validated by the internal detection circuitry, the Preamble-Detected output becomes active-low, indicating lock has been achieved. The preamble continues for a minimum of 200 VCO cycle times after Read Gate assertion to ensure the device will be very stable prior to the window margin test.

When inserting the movable test bits following the last preamble bit, it is necessary to suppress at least as many data pulses (missing bits) as the coding scheme requires before inserting either the Early or Late test bit to avoid interference from an adjacent pulse. The code must not be violated even if the movable bit is allowed to be shifted into an adjacent window. In the 5 Mb/s example, in *Figure 6*, the amount of decode-window loss is determined by shifting a test bit toward the window center starting from 10 ns outside the nominal window boundary. The bit will be moved across the window boundary toward the nominal window center in incremental steps until it resides in a position where it will be recognized by the device under test as being in the appropriate window for a large number of sequential read operations. For a device which passes specification, the movable bit will move no closer to the nominal window center than 40 ns (the ideal half-window width (50 ns) minus the Twindow spec. (10 ns)), before the bit falls into the proper decode window. Hence, the Early test bit begins at 10 ns away from (outside) the left side of the nominal window boundary and moves toward the center of the window by 1 ns increments until the corresponding Synchronized Data output is found at its expected location. Similarly, the Late test bit approaches from the right hand side of the nominal window boundary starting at 10 ns outside the boundary and shifts toward the window center. An automated tester used in manufacturing verifies that the Synchronized Data output and the shifted test bit input waveforms are in their respective valid locations for 100 consecutive successful operations (a test time versus confidence compromise). Parts that exhibit less than 6 ns of window loss from both the front (Early bit) and the back (Late bit) window tests at a 10 Mb/s data rate are designated −3 graded parts. Devices with window truncation of less than 10 ns at 5 Mb/s data rate are designated as the −4 grade part-type.

Device specifications are:

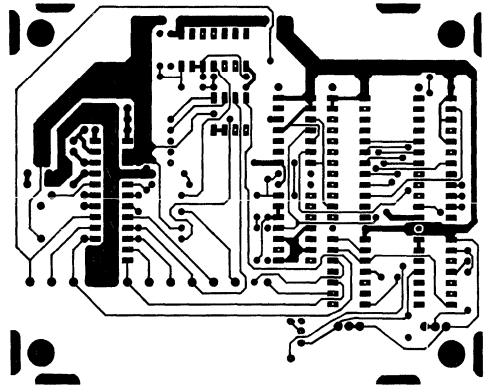
DP8462-4 → Twindow Tolerance = 10.0 ns
DP8462-3 → Twindow Tolerance = 6.0 ns

DP8462 Evaluation PC Board Layout



TL/F/9331–7

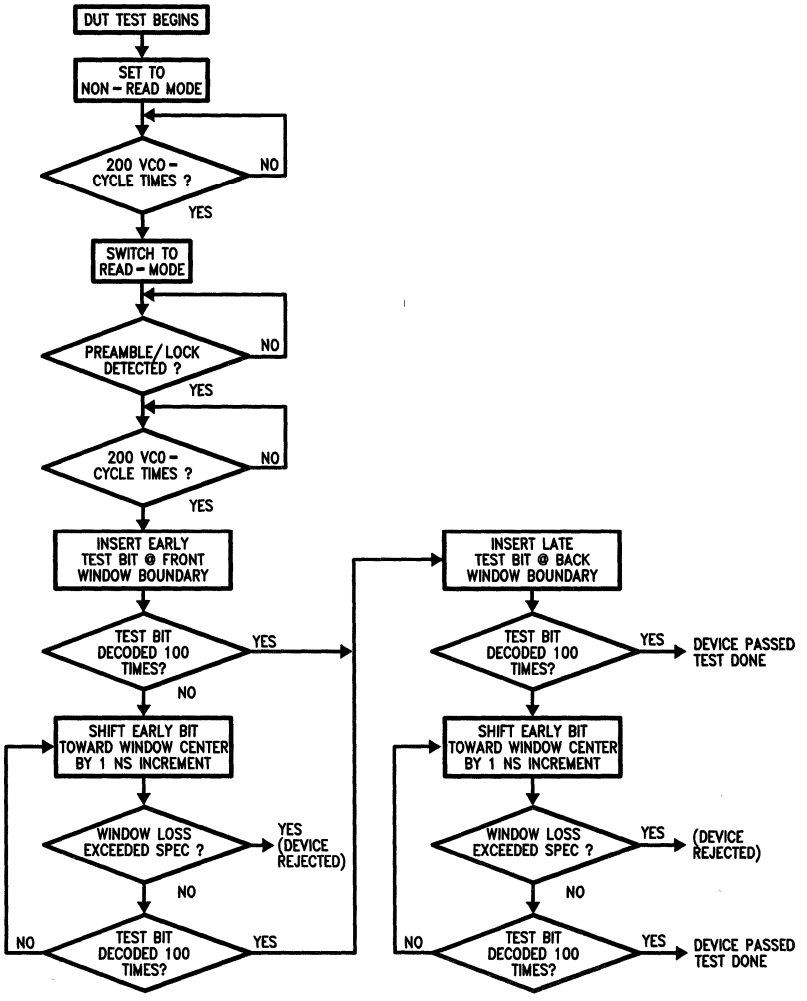
FIGURE 4a. Component Side with Ground-Plane



TL/F/9331–8

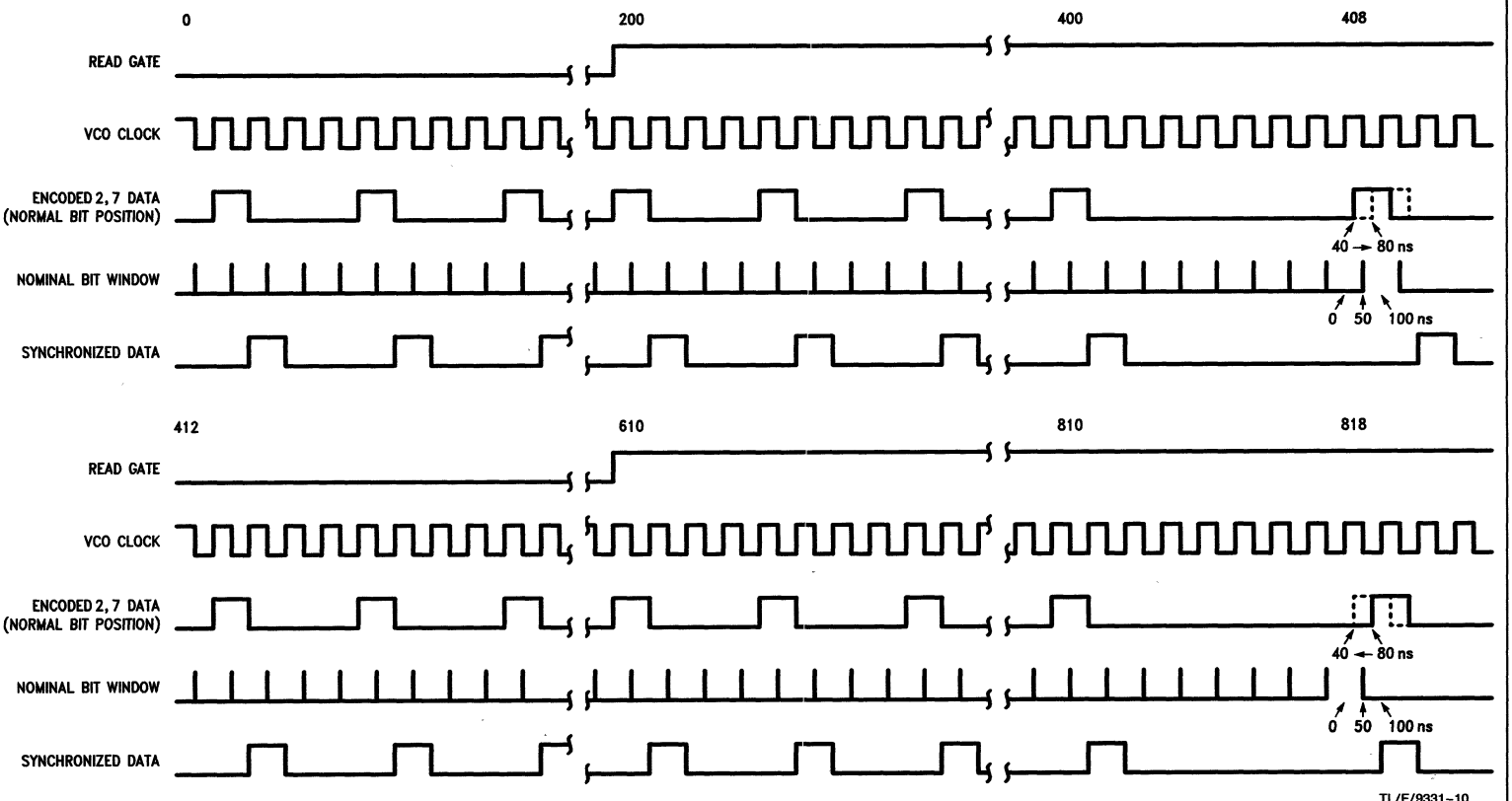
FIGURE 4b. Trace Side, Bottom View

AC Test Sequence Flow Diagram



TL/F/9331–9

FIGURE 5



***** DP8462 AC TEST TIMING DIAGRAM (5 Mbit/s) *****

VCO = 10 MHz, Preamble is 4T
Rr = 750 Rb = 1600 R1 = 200
C1 = 0.02 μF C2 = 150 pF

FIGURE 6

DP8462 EVALUATION BOARD CIRCUIT OPERATION

The DP8462 evaluation board mentioned above performs the equivalent test sequence as in automatic testers. However, it does not require elaborate pattern generators and measurement systems, or a software contorlled sequencer. Only a calibrated oscilloscope and possibly a digital ratio counter are needed to monitor the device under test. Refer to the schematic diagram in *Figure 7* and the timing diagram in *Figure 8*.

One half of U4 and U6 together form a 9-bit synchronous binary counter to cycle the address lines of U8, a read-only memory that contains all the appropriate test patterns for the Twindow test. U7 is a high speed AS (Advanced Schottky) gate used to buffer and gate the U8 ouputs to the PLL chip. U2 and U5 are CMOS inverters used to construct two crystal oscillators. One oscillator generates the 2F Reference Clock while the other sequences the rest of the logic chips. U9, a monostable multivibrator, is used to set the pulse width of a single movable bit within the Encoded Data pattern string. Q1 and its associated passive components realize a linearly variable threshold adjustment that is capable of continuously skewing the movable bit about its window center position. The other half of U4, with an "OR" gate of U7 and together witth U8, set up a detection window for the synchronized bit associated with the corresponding movable test bit. If the test bit is within the correct decode window position, pin #6 of U7 will yield 2 pulses in every read cycle. If the test bit falls out of its expected window only a single pulse is produced from this output of U7. Hence, this output and Read Gate can be connected to the ratio counter inputs for an accurate display of when the bit is within or outside of its expected window. The time measurement system in a laboratory oscilloscope can provide accurate Twindow readings.

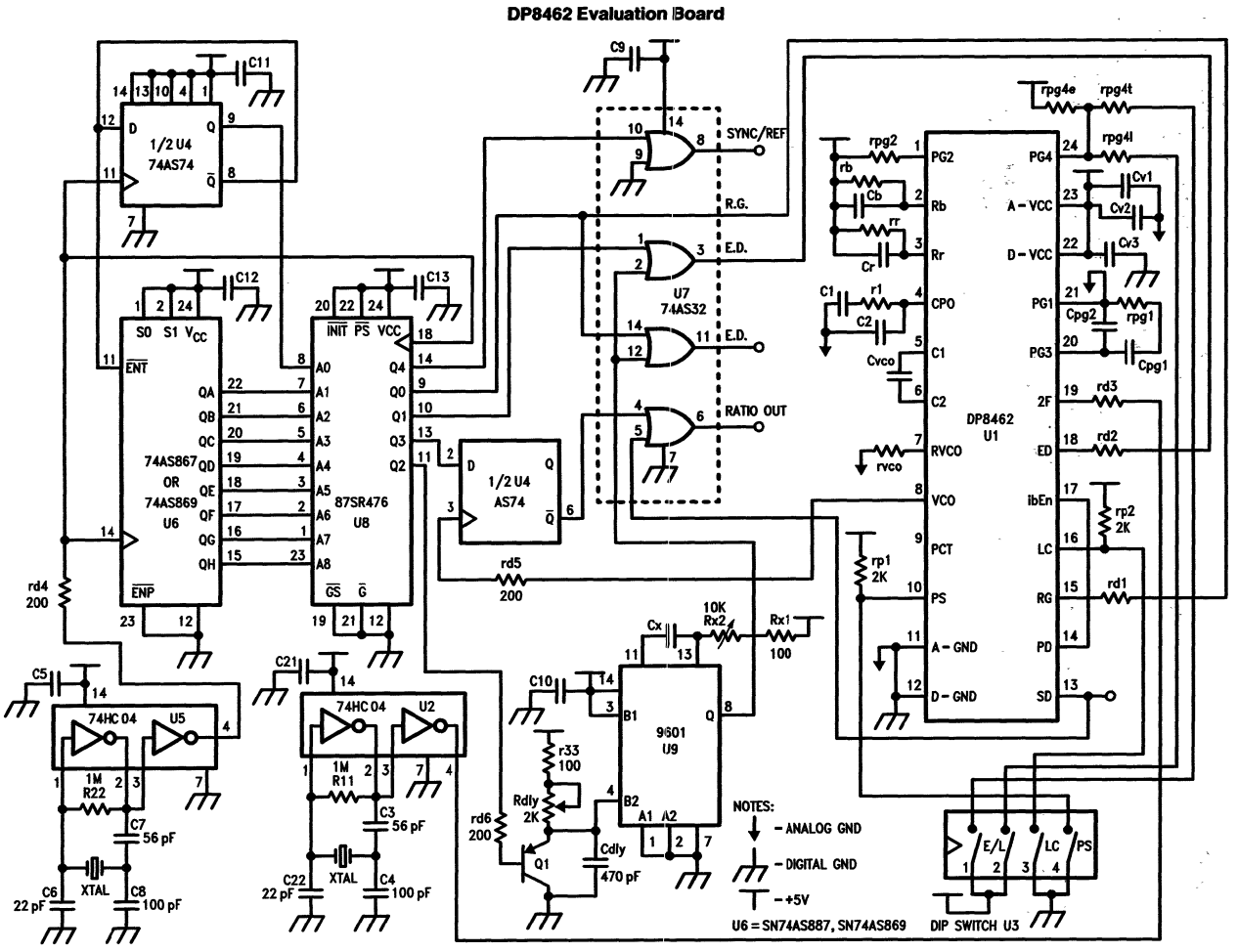
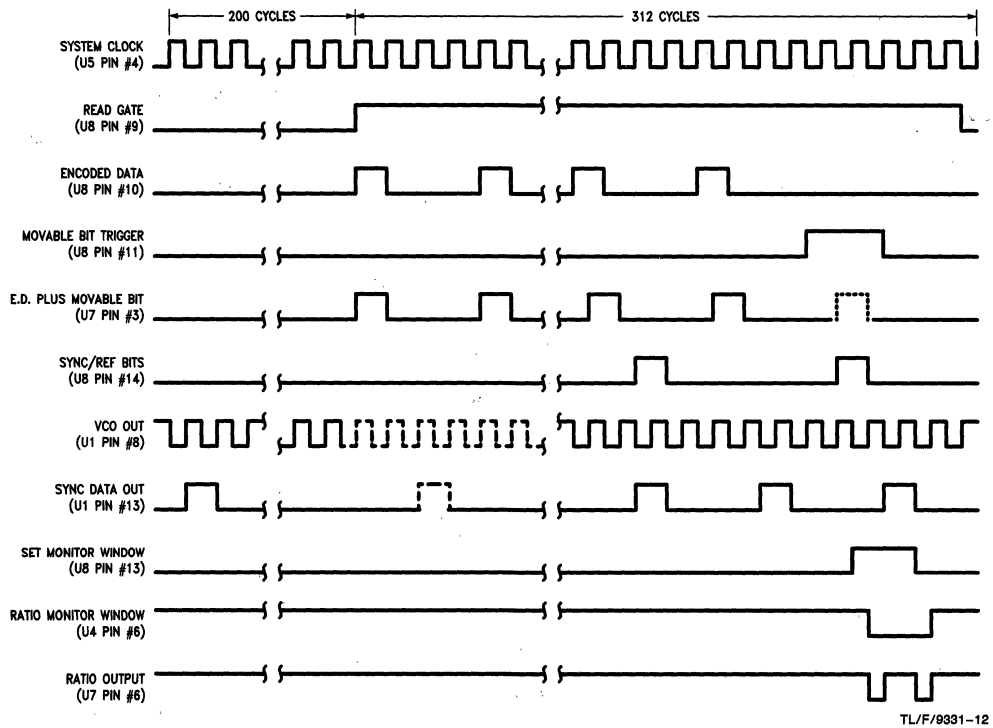
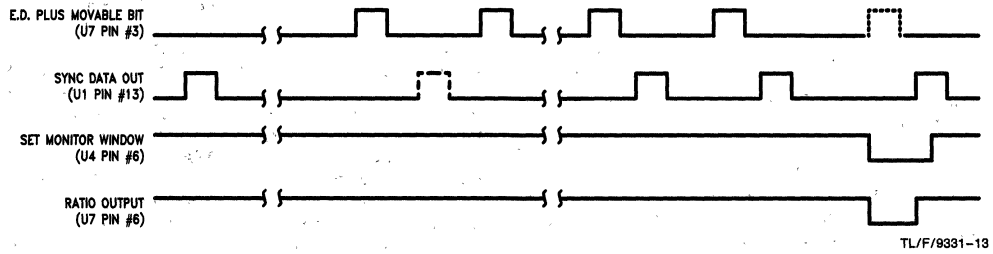

DP8462 Evaluation Board
1/2 U4 74AS74
74AS867 OR 74AS869 U6
87SR476 U8
U7 74AS32
DP8462 U1
74HC04 U5
74HC04 U2
9601 U9
SYNC/REF
R.G.
E.D.
E.D.
RATIO OUT
rd4 200
rd5 200
rd6 200
rvco
rp1 2K
rp2 2K
rd1
rd2
rd3
Rx1 100
10K Rx2
Cx
r33 100
Rdly 2K
Cdly 470 pF
Q1
1M R22
1M R11
C7 56 pF
C3 56 pF
C6 22 pF
C8 100 pF
C22 22 pF
C4 100 pF
XTAL
XTAL
rpg2
rb
Cb
rr
Cr
r1
C1
C2
Cvco
rpg4e
rpg4t
rpg4l
Cv1
Cv2
Cv3
Cpg2
rpg1
Cpg1
NOTES:
- ANALOG GND
- DIGITAL GND
- +5V
U6 = SN74AS887, SN74AS869
DIP SWITCH U3
E/L
LC
PS
TL/F/9331-11

FIGURE 7



TL/F/9331–12

If movable bit is shifted outside of its expected decode window position:



TL/F/9331–13

***Note:** Dotted waveforms depict movable bit signal positions; broken waveforms indicate outputs not yet stabilized after read gate assertion.

FIGURE 8. Essential Waveform Diagramatic Representation of the DP8462 Evaluation Board

The variable resistor, "Rdly", is used to shift the test bit about its window center. "Rx2" is used to adjust the pulse width of the movable bit, which should be set equal to the width of the Encoded Data pulses from the pattern generator ROM. U3 consists of 4 switches used to set Lock Control, Preamble Select, and to select different currents supplied to PG4 to control window centering or to introduce a predetermined amount of early or late shift of the window.

EVALUATION BOARD PREPARATION

Below are listed the component values currently being used by NSC for production testing at 10 Mb/s. (Values may change without notice.)

Rvco = 1.00 kΩ	Rr = 800Ω
Cvco = 39 pF	Rb = 1.8 kΩ
Cv1 = 0.1 μF	R1 = 82Ω
Cv2 = 1000 pF (Silver Mica)	C1 = 0.02 μF
Cv3 = 1000 pF (Silver Mica)	C2 = 510 pF
CPG1 = 2.2 μF	Cr = 0.01 μF
CPG2 = 0.22 μF	Cb = 0.01 μF
RPG1 = 68Ω	
RPG2 = 1.8 kΩ (DP8462-4), = 1.5 kΩ (DP8462-3)	(See *Figures 2c & d*)
RPG3 = 0.0Ω (DP8462-4), = 300Ω (DP8462-3)	
RPG4 = 0.0Ω	

The circuit described in *Figure 7*, for evaluation of the DP8462 PLL, uses the same components as above except for RPG2 = 3.6k, RPG3 = 0, and RPG4 = Rselect. Refer to *Figure 2b*.

The center frequency of the VCO may be checked according to the datasheet (under VCO section description). It can also be conveniently confirmed by another simple technique. First, set CPO (pin #4) at ground potential and measure the VCO frequency (minimum frequency). Then apply approximately 3V to pin #4 and again measure the VCO frequency (maximum frequency). The arithmetic mean of these two measured frequency values yields the equivalent center freqency (choose appropriate Cvco value to ensure proper VCO center frequency).

Before powering up the evaluation board, notice that the circuit has been designed so that it is possible to provide each section of the test circuit, the DP8462 (via Vcc1), the 2F Clock oscillator (Vcc2), and the rest of the digital circuitry (Vcc3), with an independent +5V regulated supply. After power(s) has been on, first check U2 and U5 for proper 2F frequency outputs. If either one or both crystal oscillators appear to exhibit excessive jitter, abnormal oscillation or is being perturbed by the other oscillator, it is possible that the gain of the inverters used is too high. Replace the 74HCO4 device or use the unbuffered device, the 74HCU04, instead; (may require new capacitor values). Next monitor U8 to make certain that all the test patterns are present (refer to *Figure 8*). Trigger an oscilloscope with the SYNC/REF BITS output and display this waveform on the CRT also. Pin #8 of U7 should present two pulses, a sync bit followed by a reference bit from Q4 of U8. Q0 of U8 is the Read Gate sequence, which should be high for 312 2F Clock cycles and low for 200 2F Clock cycles. Pin #3 of U7 outputs the Encoded Data pattern from Q1 of U8. Q1 of U8 consists of a train of 4T patterns starting when Read Gate is asserted. At the end of the 4T patterns there should be an isolated pulse (movable bit). Adjust the Rx2 potentiometer so that the width of this isolated pulse matches those of the 4T pattern (or the reference bit). Also adjusting "Rdly" should be able to shift this pulse. Position this bit such that its leading edge aligns perfectly with the leading edge of the reference bit (refer to *Figure 8*). This alignment procedure puts the movable bit in the nominal center of its decode window. Q5 and Q6 (pins #15 and #16) of U8 are optional sync signals for triggering the oscilloscope to monitor other areas of the test sequence.

TEST PROCEDURES

Set "DIP" switches: SW-3 for LOCK CTL to (H) for phase-frequency mode during preamble acquisition; SW-4 for Preamble Select to (H) for 4T preamble; SW-1 and SW-2 both set to (L), this selects RPG4E = 3 kΩ on the board (or whatever value the user chooses in testing). Check SYNCHRONIZED DATA and VCO outputs, they should be stable and in lock by the arrival of the SYNC/REF bits (at least 200+ VCO cycles after Read Gate assertion).

To test the available half-window width (complement of the half-window loss), begin turning 'Rdly" slowly while monitoring ENCODED DATA, SYNC/REF BITS, SYNCHRONIZED DATA or RATIO OUT. (It is recommended to connect READ GATE and RATIO OUT to a digital ratio counter). The ratio counter normally reads a perfect 2.0000 ratio when the test bit is within its proper decode window. Move the test bit away from its centered position (with respect to the Reference Bit positive edge) in either direction. As long as the test bit remains inside the available decode window, a ratio of two is displayed on the counter. When the test bit begins to fall outside of its decode window even occasionally, the ratio count will deviate from a perfect two ratio, indicating the window boundary has been reached. Measure the time span between the leading edge of the movable bit and that of the reference bit which corresponds to the available half-window width. Subtracting this figure from 25 ns (for 10 Mb/s data rate) yields the Twindow number (half-window truncation). Repeat the same procedure for the other half of the decode window. If a ratio counter is not used one can monitor the SYNCHRONIZED DATA or RATIO OUT node on the oscilloscope screen. As the pulse is shifted towards the adjacent window position it will begin to fade; this is the point where the window measurement should be made. This alternate test also gives a reasonably accurate estimate of the Twindow specification.

Switches #1 and #2 of U3 along with the various RPG4 components have been selected for the user who desires to experiment with early/late strobe or select RPG4 in a system design for minimum window offset. To experiment with these different options this procedure is as follows (for 10 Mb/s transfer rate): Refer to *Figure 9*.

1. Choose RPG4E ≅ 6.2 kΩ, RPG4T ≅ 620Ω, and for RPG4L select a 5.0 kΩ potentiometer. (These values may be slightly different for other data rates.)
2. Set switches #1 and #2 to (H), this selects RPG4E only. With a relatively large resistor value, a low level bias current is allowed to the PG4 pin. This produces a wider LATE window (w.r.t. nominal bit position) and a correspondingly smaller EARLY window.

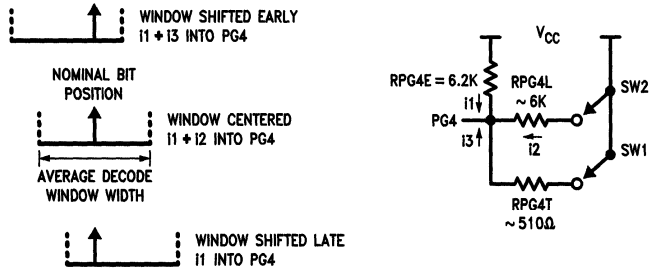


FIGURE 9

3. Set switches #1 to (L) and #2 to (H), this selects RPG4E and RPG4T in parallel. The effective RPG4 value will be the smallest of all the combinations of RPG4 resistors, hence, a larger bias current into PG4. This yields a wider EARLY window and a narrower LATE window (w.r.t. nominal bit position).
4. Set switches #1 to (H) and #2 to (L), this will select RPG4E and RPG4L; the combined value nulls out the window offset (on-time). For an easier set-up, one can utilize a single potentiometer for RPG4L (delete RPG4E and RPG4T) to determine an optimum RPG4 resistor value to nullify any inherent window offset.

SUMMARY

The DP8462 PLL chip is the third circuit in the family of National's second generation high performance data synchronizers. It combines the features of the DP8461 and DP8465 for non-harmonic phase detection in the non-read mode (DP8461 and DP8465) and in read mode during preamble acquisition (DP8461), and harmonic phase detection in read mode. Furthermore, in the DP8462, these mode selections are user controlled. Analog and digital Vcc and ground are brought out to separate device pins which further minimizes phase jitter and internal interference. Window center offset is also reduced or can be nulled out in design with a fixed external resistor. These improvements have significantly eased the task of design and volume production of higher transfer rate systems (data rates of 10 Mb/s to 15 Mb/s), where the data windows are proportionally smaller. The DP8462 device is part of National Semiconductor's DP8460 Series Disc System Data Path Chip Set. It has been designed into many disc drives and controller systems worldwide in conjunction with other members of this family such as the DP8464B pulse detector, the DP8463B (2, 7) ENDEC CHIP; the DP8466 DDC, and the DP8475 SCSI controller.

For further information regarding the DP8462 and the DISK DATA PATH chip set, please refer to National's "APPS Handbook Volume 1: Mass Storage".

ACKNOWLEDGEMENTS:

The author wishes to thank Bill Llewellyn and Pat Tucci for their constructive criticisms and diligent proofreading on this application note.

Designing with the DP8465

National Semiconductor
Application Note 416
Kern Wong

GENERAL DESCRIPTION

The DP8465 is a second generation of the successful DP8460 high performance PLL integrated circuit family of data separators/synchronizers. Like its predecessor, the DP8460, it consists of a proprietary pulse-gate which features an accurate silicon delay line, an edge-triggered digital phase comparator, a high speed matched charge pump, high impedance buffer amplifier, and a temperature compensated stable voltage controlled oscillator (VCO). The DP8465 also contains MFM decoder, missing clock detector, and lock-detect control circuitry for added flexibility to the system designer. There is one difference between the DP8460 and the DP8465. The DP8465 has been designed to perform PHASE FREQUENCY COMPARISONS during the non-read mode and switches to phase comparisons only when in the read mode, whereas the DP8460 employs only phase comparisons in both the read and non-read modes. This enhancement eliminates the possibility of false lock to the reference signal during a power-up sequence or when returning from a read operation. The DP8465 is 100% pin-for-pin and function-for-function compatible with the DP8460. It is a direct replacement for the DP8460 part type.

CIRCUIT OPERATION

The DP8465 is in the non-read mode whenever the READ GATE is deasserted. The 2F REFERENCE CLOCK input is divided by two and transmitted to the READ CLOCK output via a multiplexer. In this mode the VCO is locked onto the 2F CLOCK, keeping the VCO close to the data frequency in anticipation of locking onto the actual data stream. During the non-read mode PHASE-FREQUENCY COMPARISONS are employed, thus eliminating any possibility of false lock.

When the READ GATE input goes high, the DP8465 enters the read mode after a selectable delay time. This may be either one or thirty-two VCO clock cycles. The 2-byte delay is useful in hard-sectored drives for allowing a gap pattern to pass before the PLL locks onto the data. Soft-sectored drives do not need this delay. Once in the read mode, the PLL reference input is switched from the 2F CLOCK source to the ENCODED DATA input. The PULSE GATE allows a reference signal from the VCO into the PHASE COMPARATOR only when an ENCODED DATA bit is valid, thus PHASE-ONLY comparisons are made. The PLL, initially in the high-tracking mode, then attempts to quickly lock onto the repetitive encoded preamble.

By careful selection of the loop filter components, it takes less than one byte time for the VCO to lock onto the data stream sufficiently for preamble detection to begin. As soon as 2 bytes of the selected (ones or zeroes pattern) preamble are detected, the $\overline{\text{LOCK DETECTED}}$ output goes low. In a typical disk drive application, the $\overline{\text{LOCK DETECTED}}$ output may be directly connected to the $\overline{\text{SET PLL LOCK}}$ input. A low level on the $\overline{\text{SET PLL LOCK}}$ input causes the PLL CHARGE PUMP to switch from a high to low tracking-rate. At the same time the source of the READ CLOCK signal is switched from the 2F CLOCK input to the VCO clock. The MFM decoder also becomes enabled and begins to output decoded NRZ data. If the DP8465 is employed as a data-separator for MFM encoded data, the READ CLOCK output and the NRZ READ DATA output (which is synchronized to the READ CLOCK) may be used. These signals can be connected directly to a Disk Data Controller such as the DP8466 which controls Winchester or floppy disk drives. The MISSING CLOCK DETECTED output can also be utilized for MFM-encoded data in soft-sectored disk drives. It should be noted, however, the circuit is designed only to recognize a missing MFM clock-bit which is framed by two existing clock bits. In order to insure the detection of an address mark, simultaneous monitoring of the NRZ output for an "A1" hexidecimal code and the MISSING CLOCK DETECTED output for a single pulse within the same byte time is necessary.

When the READ GATE goes low, signifying the end of a read operation, the PLL reference signal is switched back to the 2F CLOCK, the $\overline{\text{LOCK DETECTED}}$ output goes high, and the VCO gating circuitry within the PULSE GATE is bypassed thus allowing PHASE and FREQUENCY comparisons to occur. The PLL then returns to the high tracking rate and the output signals return to their initial conditions.

If the chip is used as a data-synchronizer for MFM (on-chip data decoding not necessary) or other popular RLL codes, the SYNCHRONIZED DATA OUTPUT and the VCO CLOCK OUTPUT should be used. External decoding can be accomplished either in commercially available controller chips or via an encoder-decoder circuit, or by the customer's proprietary design.

PHASE ONLY VS. PHASE-FREQUENCY COMPARISON OPERATION

As mentioned above, the function of the PLL is to maintain phase and frequency lock between the reference signal (2F CLOCK or ENCODED DATA) and the feedback signal (VCO). A comparator that performs only phase comparison is mandatory during read-mode in order to handle the non-periodic nature of various coding schemes. With this type of detector, the phase-locked-loop functions as a feedback loop in which it responds only to the phase differences between the input and the feedback waveforms. As long as the reference and VCO signals have their edges aligned (are in phase lock) the PLL is insensitive to their frequency relationship.

During the nonread mode the PLL is required to lock onto the 2F CLOCK, a specific frequency reference that is close to the data rate. If a disturbance is somehow introduced in the system which results in cycle slipping or prolonged transient behavior of the reference clock, false lock may occur if a PHASE-ONLY comparator is being used. Under these circumstances PHASE comparison alone may be inadequate, since it discriminates only phase and not frequency information. A PHASE-FREQUENCY-COMPARATOR, therefore is employed within the DP8465 during this mode of operation. This comparator performs identically to the PHASE-COMPARATOR in the case when both inputs to the comparator have the same frequency; however, if the inputs exhibit the slightest frequency offset, the PHASE-FREQUENCY-COMPARATOR also provides a frequency-sensitive error correction signal to ensure frequency aquisition.

The PULSE GATE has two important functions. It ensures a continuous PLL lock in the presence of bit gaps encountered on the media and in the bit stream. It also provides a

precise time delay (independent of process and external component variations) necessary to align the incoming data with the center of the decoding window. The delay is exactly one half the period of the 2F CLOCK and the delay generator is referenced to the 2F CLOCK. This allows input bit jitter up to ±half the 2F CLOCK period.

The PULSE GATE incorporated in the DP8465 has two multiplexers which allow the circuit to switch from PHASE-FREQUENCY comparison to PHASE-ONLY comparison as the circuit switches from non-read mode to read mode. *Figure 1* is a block diagram of the PULSE GATE and details how this is accomplished. The delayed output of MUX-1 is shown to be compared with either the GATED VCO or the VCO DIVIDED BY TWO. The two VCO signals are multiplexed, with the INTERNAL READ GATE as the control signal, and the output is connected to the PHASE-FREQUENCY-COMPARATOR. When INTERNAL READ GATE is inactive (non-read mode) the 2F CLOCK DIVIDED BY TWO and the VCO DIVIDED BY TWO signals are selected by MULTIPLEXER-1 and MULTIPLEXER-2, respectively. In this configuration, phase and frequency comparisons are made between them and the possibility for a false lock occurence is eliminated. When the INTERNAL READ GATE is active (read mode), however, the ENCODED DATA and the GATED VCO signals are selected by the multiplexers. Through the circuit configured by the D-type flip-flops and the OR gate, the comparator effectively performs PHASE-ONLY comparisons (an INTERNAL VCO pulse is allowed to reach the input of MUX-2 only when an ENCODED DATA pulse is sensed). Thus, the DP8465 chip guarantees proper frequency lock of the VCO to the 2F REFERENCE CLOCK during the non-read mode, and it performs the necessary phase-only comparison during the read mode.

DATA SEPARATOR APPLICATION PROBLEMS

Following are some common application problems for many data separator circuit designs. The purpose of this application note is to identify these problems and to propose simple solutions. Thus, our DP8465 users will be able to avoid these potential application problems.

A) Loss of lock during read mode

In some systems the controller asserts the READ GATE randomly along a formatted track. If the READ GATE is asserted over a write splice, which usually contains unintelligible information, the PLL might false lock to some harmonic of the data, or it might be pushed to either extreme of its allowed frequency swing. Similarly, when the READ GATE is asserted over a data field, the PLL might lock to a harmonic of the data.

To recover from this problem a recovery routine must be implemented by the disk controller. This routine should toggle READ GATE so that the PLL can lock back to the 2F REFERENCE CLOCK and, after waiting a sufficient amount of time (to frequency lock to the crystal), activate READ GATE to retry the read operation.

A superior controller PLL algorithm only allows assertion of the READ GATE over a preamble or similar high frequency pattern. An example of such an algorithm is as follows:

1) Deassert READ GATE—allow a 4 byte time minimum for the PLL to lock to the 2F-REFERENCE CLOCK.
2) Wait for 2.5 bytes of valid preamble pattern.
3) Assert READ GATE
4) If valid preamble continues for 5 or more bytes then go to 5; otherwise go to 1.
5) "LOCK DETECTED" becomes active, AM search begins.
6) If AM is found, then continue the read routine; otherwise go to 1.

B) False lock in the non-read mode

The DP8465 **has been specifically designed to eliminate the possibility of false lock during the non-read mode.** This is accomplished by the use of a phase-frequency comparator in the non-read mode as was described in the PULSE GATE section.

False lock during the non-read mode can occur by two means in systems using phase only comparisons in the non read mode. When the power supply of the PLL circuit is switched on for the first time, the VCO ramps toward the reference frequency. The aquisition process may lock the VCO to some harmonic of the 2F REFERENCE CLOCK if the bandwidth (capture range) is not high enough. False lock can also occur in the non-read mode after an aborted read operation as described above. If the VCO has either lost lock or has been driven far from its center frequency while trying to read, then while relocking to the crystal, if the capture range is not wide enough, false lock might occur.

C) Quadrature Lock

Quadrature lock is a phenomenon which may occur when the periodic pulses in the PLL synchronization field become distorted such that they appear as periodic pulse-pairs as shown in *Figure 3*. This phenomenon is usually caused by the read channel electronics or recording components in the disk drive and may give rise to a false lock condition in the PLL known as quadrature lock.

Within the standard synchronization field which precedes the data field, bits are recorded at a constant frequency for a time sufficient to allow the PLL to acquire lock. With normal recording and read-circuit behavior, this synchronization information reaches the PLL as a continuous, periodic data stream. In some disk drives, if an offset has somehow been induced into the recorded information, or if a read-channel asymmetry exists within the drive electronics which skews the flux reversal zero-crossing point, the synchronization field waveform which reaches the PLL may appear in the form of periodic pulse-pairs. This condition only arises when a repetitive pattern is present, and gives rise to the occurance of quadrature lock. Note that quadrature lock is actually more prone to occur within systems where a low-noise design has minimized the randomizing effect which noise has on bit position.

Optional External Quadrature Lock Circuitry

To eliminate the possibility of a quadrature lock condition, a simple circuit (4 passive components) solution may be employed to prevent its occurrence. The circuit shown in *Figure 2* has the effect of forcing a misalignment of the data synchronization window with respect to the input pulse pattern should the quadrature condition occur. This circuit does not affect PLL operation once proper lock has occurred, and it is disabled once PLL LOCK has been detected by the DP8465. Although a recommended value is given for the resistor in the support circuit, some experimentation may be required in determining an optimum value for use within any particular system. *Figure 3* shows a diagrammatic representation of the quadrature lock waveforms.

D) VCO Jitter

The recommended starting value for the charge pump current setting resistor, R_{RATE}, was initially 1.5 kΩ. It has been found that maintaining a value of R_{RATE} at or below 820Ω has a stabilizing effect on the jitter performance of the VCO circuitry. Thus, we recommend that this 820Ω value be substituted for the originally recommended value of 1.5 kΩ.

As shown in the DP8465 data sheet, the minimum value of R_{RATE} is 400Ω. When choosing values for R_{RATE} and R_{BOOST}, the only requirement is that the total charge pump input current is less than or equal to 2 mA. This requirement can be met by adhering to the following requirement on the parallel combination of R_{RATE} and R_{BOOST}.

$$R_{RATE} \| R_{BOOST} \geq 350\Omega$$

(i.e., the parallel value of R_{RATE} and R_{BOOST} should not fall below 350Ω.)

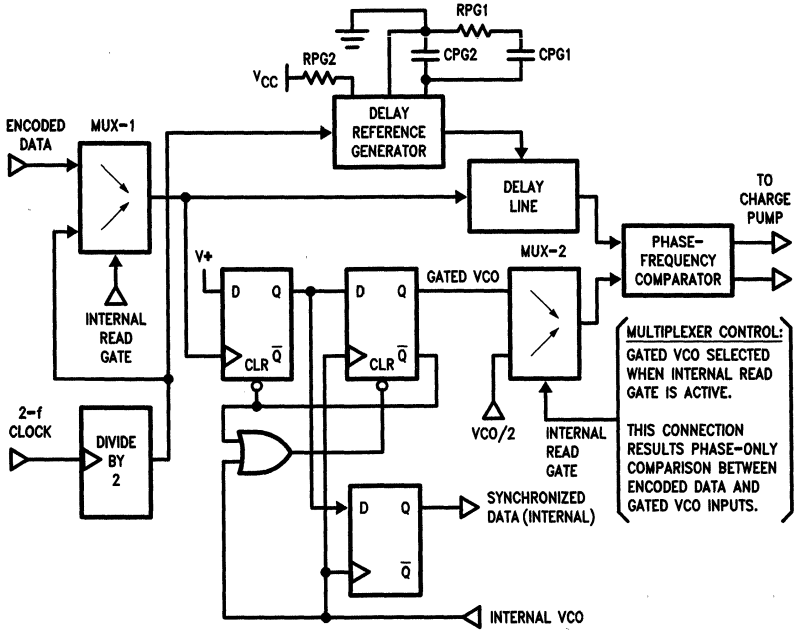
When the R_{RATE} value adjustment is implemented, all other component values associated with the loop filter must also be modified in order to maintain the original PLL response characteristics within the disk data field. The DP8465 Data Sheet shows a sample filter calculation and also several sets of loop filter component values for different values of R_{RATE}.

SUMMARY

The DP8465 is one of National's second generation single-chip high performance PLL circuits for application in disk memory systems. It features a comparator with both phase-frequency and phase-only comparison capabilities. The DP8465 offers significant savings of cost and time in production, test, and maintenance since only a few fixed passive components are required for operation. The need to trim any external components has been eliminated and since no external components determine window accuracy, the performance will not be sensitive to external variations. The chip requires a single +5V supply and it is housed in a narrow 24-pin dual-in line package (also available in 28-pin PCC package). The DP8465 is a direct replacement for the DP8460 and it may be used either as a data synchronizer for MFM or any of the existing Run-Length-Limited codes, or as a data separator for MFM.

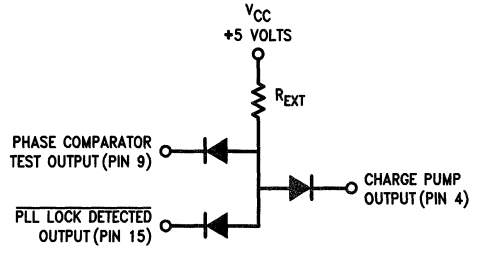
For further information, the reader should also refer to the National Semiconductor Application Note 414, Precautions for Disk Data Separator (PLL) Designs.

Pulse Gate Block Diagram



TL/F/8600–1

FIGURE 1.



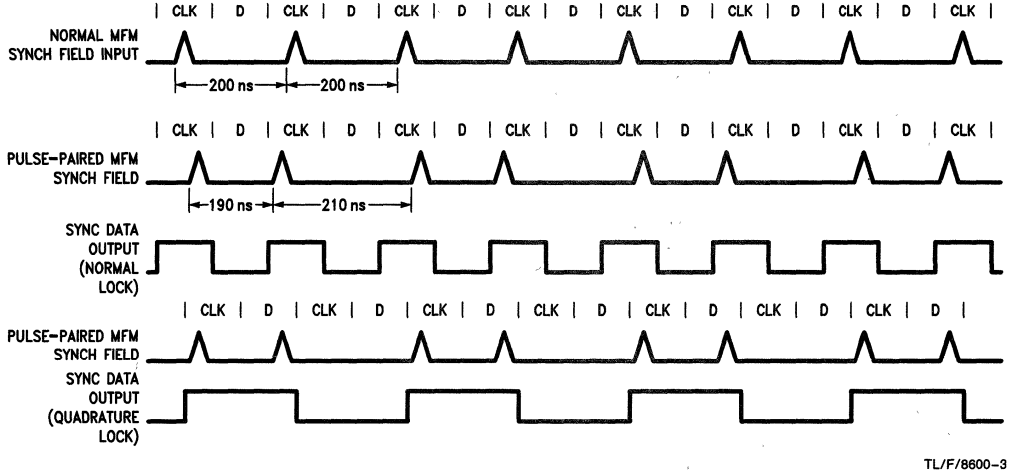
TL/F/8600–2

Recommended value for R_{EXT}: 10 $[R_{RATE} \| R_{BOOST}] \leq R_{EXT} \leq 20[R_{RATE} \| R_{BOOST}]$.

Diodes must be carefully chosen for minimal zero-bias capacitance and reverse leakage current (2 pF and 100 nA or better are recommended values, respectively).

Recommended diode types: 1N4448
1N4148
1N914

FIGURE 2. External Circuitry Used to Insure Correct PLL Lock to a Pulse-Paired Synchronization Field



TL/F/8600–3

FIGURE 3. Timing Diagram of PLL Quadrature Lock Within a Symmetrically Pulse-Paired Synch Field

Application Issues for the DP8465 Family of Data Synchronizers

National Semiconductor
Application Note 581
William Llewellyn

This application note covers National's first generation hard disk data synchronizers and separators, i.e., the DP8465, DP8462, DP8461, DP8455 and DP8451, citing some common PLL application concerns and describing some techniques to enhance performance.

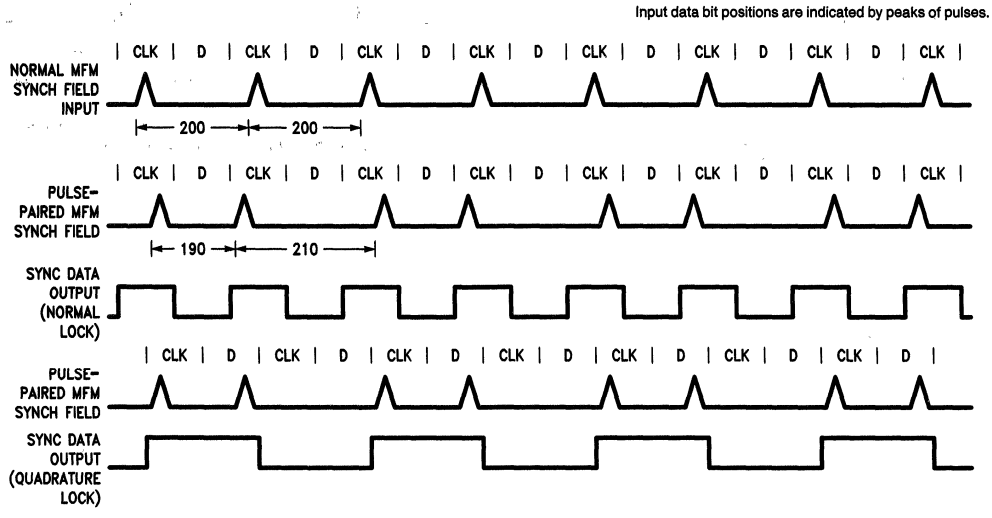
FRACTIONAL HARMONIC LOCK

The frequency discrimination capacity of the digital phase detector within the data separator/synchronizer is suppressed whenever a pulse gate technique is employed. Although this pulse gating technique is a standard in disk drive applications and is necessary in order to allow the PLL to remain phase locked to randomly spaced disk data bits, it essentially causes the phase detector to behave as would an analog quadrature multiplier, i.e., the capture range of the loop takes on the finite value related to the loop bandwidth. Under ordinary circumstances, this is quite acceptable; however, it does permit the PLL to become susceptible to a form of quasi-stable false lock to fractional harmonics of the input frequency. (For example, a typical lock null for this phenomenon would be where the VCO stabilizes at 5/6 or 6/5 of its nominal frequency.) The conditions for occurrence of this are:

1) Pulse gate in use;
2) Periodic pattern is present (i.e., preamble);
3) Perturbation occurs either during or just prior to the periodic pattern, causing the VCO to swing outside of the dynamic capture range of the loop.

Since the capture range in a typical disk PLL configuration is on the order of ±2% of the data rate, it can be seen that harmonic lock could easily occur given an adequate perturbation of the loop. Typical causes of perturbations would be media defects and spurious noise pulses, among others, but the most commonly seen occurrence within soft-sectored systems is where an attempt is made to "read" through the write splice region on the disk (zone where the head write current is either switched on or off) during a sector search operation. Although recovery from harmonic lock will occur readily if the read operation is terminated (all National's currently released disk data synchronizers incorporate frequency discrimination when locking to the reference input), systems which exhibit fractional harmonic lock during a sector search may experience extended latency time as the search routine repeatedly tries to find the desired sector.

The soundest solution to harmonic data lock is the incorporation of a sector search routine which asserts Read Gate only within a preamble field, in conjunction with the use of a PLL which employs frequency acquisition during the lock sequence. National's DP8451 and DP8461 are designed for this type of application, employing frequency lock for the standard MFM preamble (2T, when T is equal to the VCO period). The DP8462 allows optional frequency lock for both 2, 7 preamble types (3T or 4T). National's most recent stand-alone data synchronizer, the DP8459, can accommodate frequency lock (also optional) with any standard preamble type (1T, 2T, 3T or 4T). The DP8459 also incorporates zero-phase start at the assertion of Read Gate, minimizing the phase disturbance at the beginning of a read operation.



TL/F/10337–1

FIGURE 1. Timing Diagram of PLL Quadrature Lock within a Symmetrically Pulse-Paired Synch Field

Within systems where it becomes evident that the reading of write splices is consistently producing sector-not-found errors, while at the same time it is not possible to either modify the sector search algorithm (in order to avoid the splices) or to make hardware modifications, the PLL can be made less sensitive to the write splice disturbance by the lowering of the loop bandwidth. This is recommended only as an interim solution until firmware or hardware accommodations can be made.

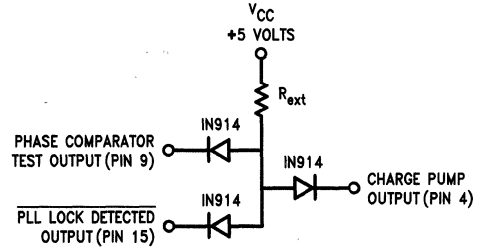
QUADRATURE LOCK

Another form of false lock may also occur (pulse gate in use) within a periodic disk pattern (preamble) given one additional condition; the periodic disk pattern being presented to the PLL exhibits a pulse-pairing phenomenon (typically introduced by the data channel electronics); see *Figure 1*. Within this particular pattern, PLL has the potential to lock to the correct frequency while remaining caught on a phase null 90 degrees from nominal. In this case, each pair of bits is interpreted by the PLL as residing in two directly adjacent windows (actually a violation of most recording codes) with the two subsequent windows empty. Although the bits appear to be greatly shifted within these windows, the phase corrections produced complement each other and average to a filtered DC value of zero. This repeating pattern is thus self-sustaining.

Quadrature lock is unique in that it is more likely to occur within a relatively well designed, noise-free system environment. The reason for this is that the randomizing effect noise ordinarily has on the data stream has been minimized, preserving the purity of the pulse paired pattern and thus increasing the probability of this form of lock. Again, this form of lock is generally only seen within the preamble, and may occur within either soft or hard sectored systems. Easily recognized waveform patterns seen at the separator/synchronizer outputs would be (1) the Synchronized Data Output exhibits a 110011001100 . . pattern instead of the standard 1010101010 . . . preamble pattern; (2) the Phase Comparator Test output pulse width consistently remains at approximately half of the VCO period (nominal width should be 7–12 nanoseconds); (3) -Lock Detected does not become active (low).

The most robust solution to this phenomenon (as in the section on harmonic false lock, above) is to incorporate a hard or pseudo hard-sectored search algorithm in conjunction with a data separator/synchronizer which employs frequency acquisition within the preamble as do the DP8451, DP8461, DP8462 and DP8459. The frequency acquisition mode allows no residual phase or frequency error within the PLL when locked, and thus the possibility of both quadrature and harmonic lock is eliminated.

A "bootstrap" hardware technique to avoid quadrature lock can be incorporated with the DP8465/55 devices, which do not incorporate preamble frequency lock internally (see *Figure 2*). This technique involves the inclusion of four passive elements external to the chip, which will deliberately force the window to shift away from the 90 degree phase null when (and only when) quadrature lock occurs. The passive network is automatically disabled once the PLL detects preamble lock. Although a recommended value is given for the resistor in this support circuit, some experimenting may be required in determining an optimum value for use within any particular system.



TL/F/10337–2

Recommended value for R_{ext}:
$10[R_{rate} \| R_{boost}] \leq R_X \leq 20\ [R_{rate} \| R_{boost}]$

FIGURE 2. External Circuitry Used to Insure Correct PLL Lock to a Pulse-Paired Synchronization Field

VCO JITTER

The inherent purity of the VCO's operating frequency is a key element in the accuracy of the data separator/synchronizer window generation. Any "jitter" present in the VCO frequency (any modulation of the period of the waveform by noise or any other source) will degrade the performance of the PLL. Within National's initially released DP8451/55/61/62/65 data separators-synchronizers, it has been found that maintaining a value of R_{rate} at or below 820Ω has a stabilizing effect on the jitter performance of the VCO circuitry. We recommend the following guidelines be followed in the selecting of charge pump resistors and loop filter components for these circuits (see Table I):

1) An 820Ω value resistor should be substituted for the originally recommended value of 1.5 kΩ.
2) Although this new R_{rate} value is below the original DP8451/55/61/62/65 specification limit, a substitute requirement has been placed on both R_{rate} and R_{boost} to maintain proper circuit operation:

$$R_{rate} \parallel R_{boost} \geq 350\Omega$$

(i.e., the parallel value of R_{rate} and R_{boost} should not fall below 350Ω.)

3) If the inclusion of an 820Ω value for R_{rate} means a component change within an existing system (i.e., the user had been employing some higher value), all other component values associated with the loop filter must also be modified in order to maintain the original PLL response characteristics within the disk data field:

Define: $M = R_{rate}(old)/820$ [eg., (1500/820)]. Then,

$$CLF1' = CLF1 * M$$
$$CLF2' = CLF2 * M$$
$$RLF1' = RLF1 / M$$

4) Additionally, in the cases where the Quadrature lock circuitry are in use:

$$R_{ext}' = R_{ext} / M$$

TABLE I. Data Separator/Synchronizer Reference List

Device	Synchronized Codes	Separated Codes	Frequency Lock	Window Centering Trim
DP8461	MFM; 1, N	MFM	Reference & 2T (MFM)	None
DP8462	2, 7 MFM;1, N	None	Reference; 2, 7 (3T or 4T, Optional)	Optional
DP8465	2,7 MFM; 1, N	MFM	Reference	None
DP8451	MFM; 1, N	None	Reference, 2T (MFM)	None
DP8455	2, 7 MFM; 1, N	None	Reference	None
DP8459	1, N; 2, 7; MFM; GCR	None	All (1T–4T, Optional)	5-Bit Digital Strobe

Note 1: DP8461 and DP8451 pinouts match the DP8465 and DP8455, respectively; for use with hard and psuedo-hard sectoring only.

Note 2: DP8462 incorporates optional frequency acquisition for 2, 7 synchronization fields, but may be used as a data synchronizer for any disk code.

Note 3: DP8451 and DP8455 also available in PCC package (20 pin).

PRINTED CIRCUIT BOARD LAYOUT RECOMMENDATIONS

As with any high-frequency analog circuitry, care should be taken in PC board layout when using the DP8465 family of data synchronizers. The following is a list of practical guidelines intended to help insure sound, trouble-free operation with these devices.

1) Establish a local V_{cc} island or net, separate from the main V_{CC} plane, to which the device and its associated passive components can be connected. V_{CC} supply filtering should be liberal and in very close proximity to the chip. The electrical lead length of the filter capacitance between the V_{CC} and ground pins themselves should be as short as possible (minimizing lead inductance). Inclusion of a quality high-frequency capacitor, such as a 1000 pF silver-mica capacitor, in parallel with a ceramic 0.1 μF capacitor, is recommended. (Note: the chip is particularly sensitive to inadequately filtered switching supply noise.)
2) Effective capacitive bypassing of the R_{boost} and R_{rate} pins directly to the V_{CC} pin is very important. Again, use quality, high-frequency capacitors and maintain the shortest possible electrical lead length.
3) Use the main digital ground plane for all grounding associated with the device. The ground pin and the PG1 pin should tie directly to this plane.
4) Do not locate the chip in a region of the PC board where large ground plane currents are expected.
5) Locate all passive components associated with the chip as close to their respective device pins as possible.
6) Orient the chip's external passive components so as to minimize the length of the ground-return path between each component's ground plane tie point and the chip's ground pin. (Ground noise at the loop filter components, RLF1, CLF1 and CLF2, which is not identically present at the ground pin (common mode), is coupled through the filter components into the VCO control voltage pin.)
7) Include no planing whatsoever (V_{CC} or ground) directly between adjacent pins. This will minimize parasitic capacitance at each pin. Planing between the two pin rows, however, is recommended (directly beneath the package).
8) Avoid running signal traces between pins.
9) Run no digital signal lines between or adjacent to the analog pins or signal traces (pins 1 through 7 and PG3) in order to avoid capacitive coupling of digital transients.
10) Minimize the total lead length of the C_{VCO} capacitor. Inductance in this path degrades VCO performance, as does parasitic pin capacitance.
11) Do not place any bypass filtering at the R_{VCO} pin (minor coupling of the VCO waveform into this pin is normal and acceptable).
12) Eliminate negative-going voltage transients (undershoot) at the digital input pins (pre-termination of driving lines may be necessary) to avoid drawing transient input-clamp-diode current from the device pins.
13) Minimize digital output loading; i.e., if outputs must drive large loads or long lines, employ buffers.
14) Allow unused digital output pins to float, unconnected to any net.
15) Avoid locating the chip within strong electromagnetic fields. If possible, choose the "quietest" region of the board.
16) If chip socketing is desired, use a low-profile, low mutual capacitance, low resistance, forced-insertion type (socket-strips are recommended). Avoid the use of "ZIP-DIP's".
17) Do not use wire-wrap interconnect, even in an evaluation set-up.
18) Make allowance for pin-to-pin capacitance when determining C_{VCO} (Typically 4–5 pF) from data sheet formula.

DP8459 Window Strobe Function

National Semiconductor
Application Note 578
Kern Wong

INTRODUCTION

This note explains in detail the Strobe Function incorporated on the DP8459 ALL-Code Data Synchronizer. It is recommended that the reader reviews the data sheet prior to reading this note. The Strobe Function within the DP8459 chip is considerably more intricate and versatile than any existing Strobe Function on commercially available data synchronizers, thus, this application note is intended to point out the significance of this device feature and to convey important information on the proper use of it. It is also the intent of this writing to offer an explanation on the concept of Strobing and associated terminology for customers who may not be familiar with the subject. Further, the ease with which the Strobe Function can be employed to optimize system performance and to realize cost effective manufacturing of products is discussed.

DESCRIPTION

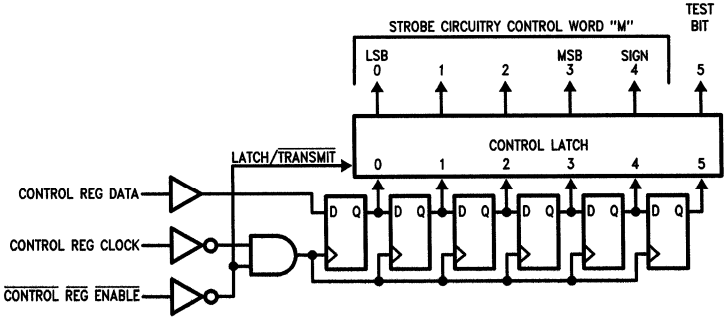
The Strobe Function implemented in the DP8459 chip provides a powerful and convenient means for the synchronization window to be shifted either Early or Late with respect to its nominal position. By definition the Strobe step, t_S is the digitally programmable time displacement of the (DP8459) synchronization window from its nominal position and is expressed as:

$$t_S = M \times (1.8\% \times t_{VCO})$$

where "M" is the value of the Strobe control word, having a range from −15 to +15. The fine resolution of the individual strobe step (LSB) in conjunction with the thirty-one steps of movement provided by the DP8459 is unprecedented among commercially available devices, which have at most a few fixed strobe positions if any at all. The strobe control word, "M", is set by five of the six binary bits within the Control Register as shown in *Figure 1*. Bit #4 is the sign bit which determines Early or Late strobe movement. The last bit (#5) in the control word is the test bit, which when set to high is used for factory testing. This bit is always the first bit serially loaded into the shift register. The following truth table (refers to Table I) maps "t_S" to the corresponding Strobe word representations.

TABLE I. Window Strobe Truth Table

Strobe Bit					Strobe	Window Strobe
4	3	2	1	0	Word M	t_S (Typical)
0	1	1	1	1	−15	$-0.270 \times \tau_{VCO}$
0	1	1	1	0	−14	$-0.252 \times \tau_{VCO}$
0	1	1	0	1	−13	$-0.234 \times \tau_{VCO}$
0	1	1	0	0	−12	$-0.216 \times \tau_{VCO}$
0	1	0	1	1	−11	$-0.198 \times \tau_{VCO}$
0	1	0	1	0	−10	$-0.180 \times \tau_{VCO}$
0	1	0	0	1	−9	$-0.162 \times \tau_{VCO}$
0	1	0	0	0	−8	$-0.144 \times \tau_{VCO}$
0	0	1	1	1	−7	$-0.126 \times \tau_{VCO}$
0	0	1	1	0	−6	$-0.108 \times \tau_{VCO}$
0	0	1	0	1	−5	$-0.090 \times \tau_{VCO}$
0	0	1	0	0	−4	$-0.072 \times \tau_{VCO}$
0	0	0	1	1	−3	$-0.054 \times \tau_{VCO}$
0	0	0	1	0	−2	$-0.036 \times \tau_{VCO}$
0	0	0	0	1	−1	$-0.018 \times \tau_{VCO}$
0	0	0	0	0	0	0
1	0	0	0	0	0	0
1	0	0	0	1	1	$0.018 \times \tau_{VCO}$
1	0	0	1	0	2	$0.036 \times \tau_{VCO}$
1	0	0	1	1	3	$0.054 \times \tau_{VCO}$
1	0	1	0	0	4	$0.072 \times \tau_{VCO}$
1	0	1	0	1	5	$0.090 \times \tau_{VCO}$
1	0	1	1	0	6	$0.108 \times \tau_{VCO}$
1	0	1	1	1	7	$0.126 \times \tau_{VCO}$
1	1	0	0	0	8	$0.144 \times \tau_{VCO}$
1	1	0	0	1	9	$0.162 \times \tau_{VCO}$
1	1	0	1	0	10	$0.180 \times \tau_{VCO}$
1	1	0	1	1	11	$0.198 \times \tau_{VCO}$
1	1	1	0	0	12	$0.216 \times \tau_{VCO}$
1	1	1	0	1	13	$0.234 \times \tau_{VCO}$
1	1	1	1	0	14	$0.252 \times \tau_{VCO}$
1	1	1	1	1	15	$0.270 \times \tau_{VCO}$

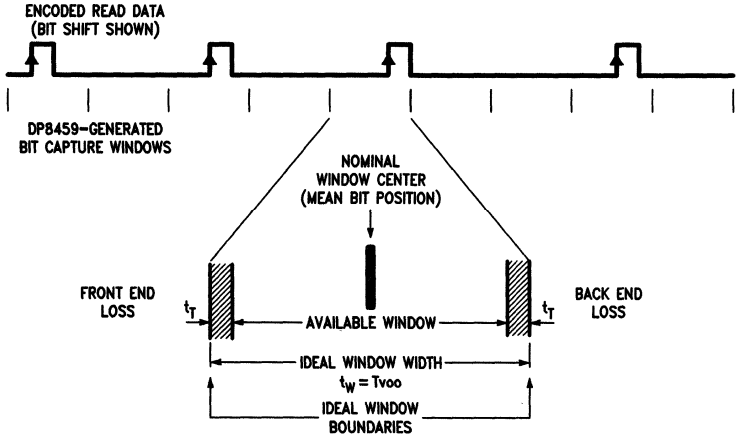


TL/F/10251–1

FIGURE 1. Strobe Control Register Diagram

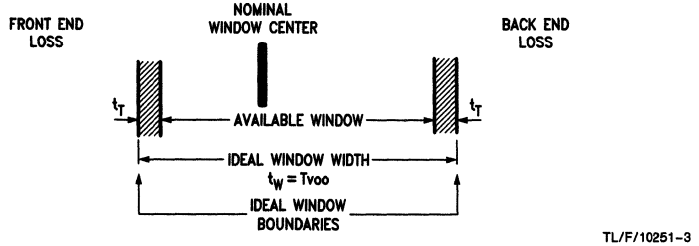
The Strobe Function is also referred to as Window Strobe, because strobing is used to adjust the relative position of the synchronization window. To better illustrate the relationship of strobe to the synchronization window, please refer to *Figure 2a*. This diagram depicts the concept of synchronization window. The synchronization window is defined as a continuously repeating time cell, which has a nominal time span equal to the period of the VCO. Ideally, an Encoded Read Data pulse will be captured and correctly interpreted regardless of its position within the window boundaries. However, the ideal window width cannot be realized in practice, due to noise and device non-idealities such as VCO jitter. Thus, a small fraction of the usuable window is truncated, leaving a correspondingly narrower available window width. If device mismatch and other asymmetric phenomena are also present, another window eroding phenomenon called window shift results. This phenomenon causes the average data window center to shift with respect to the expected mean bit position. Referring to *Figure 2b*, note that the static window width in this case appears congruent to that in *Figure 2a*. However, early shifted data bits in *Figure 2b* actually have a greater probability of falling outside the window since it is shifted late. The Strobe function is a simple to use, yet powerful feature which, as will be explained later, allows the user to offset, within ½ LSB, the undesirable window shift and improve system performance.

Synchronization Window Diagram



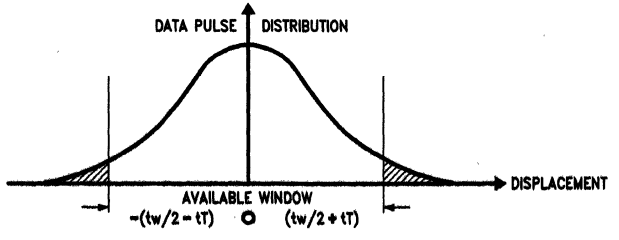
TL/F/10251–2

FIGURE 2a. Synchronization Window



TL/F/10251–3

FIGURE 2b. Window Shifted Late



TL/F/10251–4

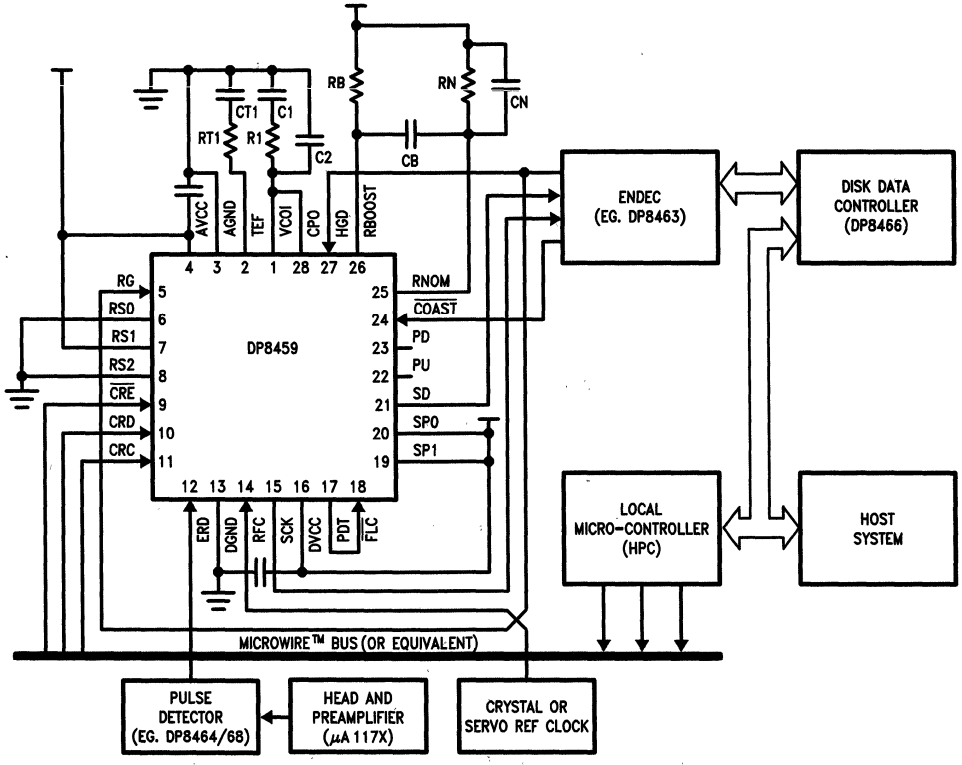
FIGURE 2c. Bitshirt vs Probability of Its Occurrence

BYPASSING STROBE FEATURE

To fully harness the Strobe feature as in an intelligent interface environment, the MICROWIRE™ bus may be interfaced to a processor or controller function to program the internal control register during initialization or between read operations. The information stored in the control register, bits #0 through #4, defines the amount and sense of window displacement. (Important, bit #5 must be set to "0" for normal operation, since a "1" at this location will place the chip into a test mode used during production testing.) A typical system diagram with the MICROWIRE bus is shown in *Figure 3*.

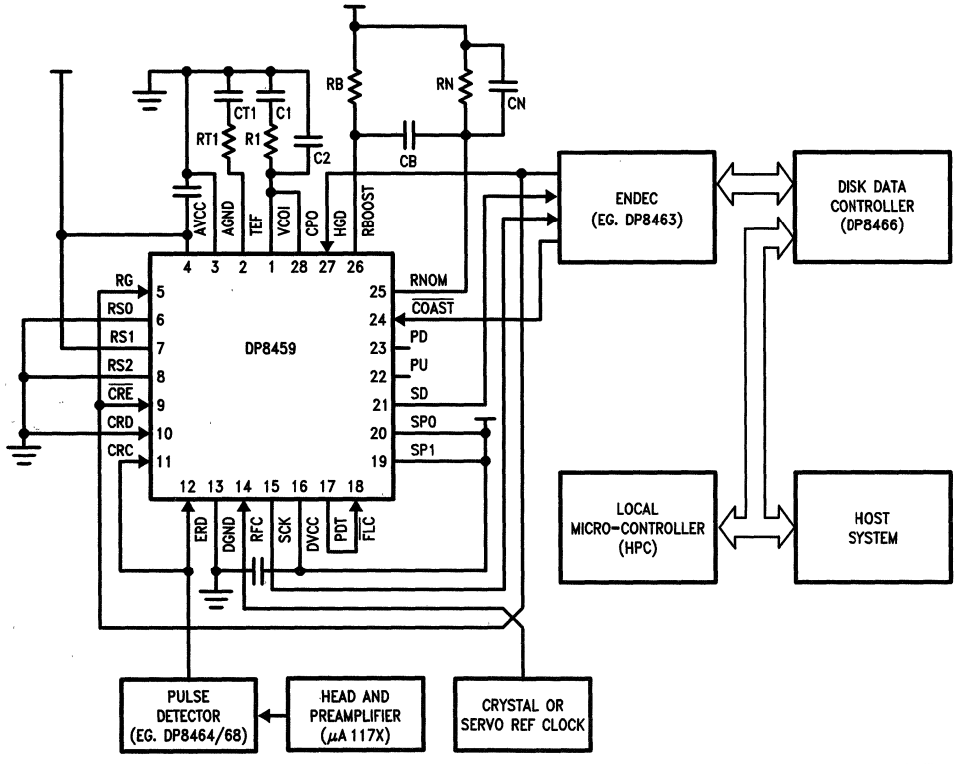
If the customer does not wish to use the Strobe feature (strobe word may be fixed at the nominal setting of all bits equal to 0), the MICROWIRE bus can be bypassed. By appropriate hardwiring of the control register inputs, as depicted in *Figure 4*, the control register content is always set to the nominal strobe upon Read Gate assertion. Please note that the window truncation is not specified at the "Nominal" Strobe setting. Therefore, optimum window performance may not be realized at this setting. This and similar device configurations are usually employed when the prime interest is to construct a simple setup for general device evaluation purposes. Of course it can also be used in lower data rate systems and, particularly, in tape and floppy drives where the window margin requirement is generally less stringent.

Although the technique just described can serve as a convenient means to set the control register content, it is very important to note that it requires the user to correctly sequence the Read Gate. For example the chip must be powered up in the non-read mode, this is with Read Gate low. Since Read Gate is tied to the "CRE" pin, a logic low enables the shift register portion of the control register to serially enter data while the control latch is held in its previous state. Concurrently, the E.R.D. pulses must be present which serve as clock pulses to shift the all "zeroes" data. Finally, the Read Gate must then be switched to the high state, thus inhibiting data entry into the shift register and allowing the new data just entered into the shift register to be transferred to the latch. Whether loading the control registers with the method as shown in *Figure 4* (this diagram represents a typical system setup where the RG also controls the HDG and CRE pins) or employing the same scheme via external circuitry, if the Read Gate is tied to CRE, the synchronizer chip must be powered up with Read Gate low. After the control data has been loaded, Read Gate must be set high. (The need to pulse the Read Gate was not mentioned in the March 1988 datasheet).



TL/F/10251–5

FIGURE 3. Typical System with the MICROWIRE Bus



TL/F/10251–6

FIGURE 4. Typical System without the MICROWIRE Bus

WINDOW TRUNCATION SPECIFICATION

The Strobe function is also used to qualify the window truncation specification. For example, the DP8459 half-window loss, "t_T", represents the sum of static window truncations from all relevant circuits and mechanisms in the data synchronizer, e.g. window shift, VCO jitter, etc. At 10 Mb/s, "t_T" is specified as 3.0 ns (6% of VCO period) maximum with M = −2. If the strobe is not used, i.e. M = 0, then "t_T" maximum would typically be 4.8 ns. This, however, is not tested and therefore not guaranteed.

The static window truncation consists of two separate major contributors to window loss, namely "T_S" (which is due to window shift) and "T_N" (which is truncation due to noise and other relevant mechanisms). The window strobe feature can be used to minimize "T_S" to ½ LSB of the strobe step. At 10 Mb/s "T_S" can be minimized to typically 0.45 ns. The remaining loss is due to "T_N".

The DP8459 window specifications are valid over the entire operating temperature and power supply ranges per datasheet. Thus the window specification "T_T" is actually comprised of "T_N", "T_S", temperature guardbanding, and measurement system guardbanding. The "t_T" parameter is tested at 4.75V V_{CC} and 5.25V V_{CC}. The device is checked at 10 Mb/s and 20 Mb/s data rates for window specification compliance as part of the final test in manufacturing. It should be noted that the data sheet recommended strobe position settings are determined by statistical averaging of many units during device characterization. In high performance drive system applications, users can take advantage of this Strobe feature to individually "tune" the synchronizer for optimal detection window symmetry.

STROBE FUNCTION

There are several reasons why one wishes to alter the inherent average window position generated by the phase-locked oscillator within the synchronizer. First, the average data window position may not be perfectly centered about the expected mean data bit position. Second, deliberately skewing the window position can serve to recover malshifted data bits. Third, shifting the window can introduce excessive error rate for testing and calibration purposes.

There could be a substantial amount of random displacement (jitter) of individual bits dependent on the design of the read/write head, media, signal processing electronics, and the result of bit interaction. This is exemplified in *Figure 2c*, which shows that the data bit displacement (also known as bitshift) versus its probability of occurrence is a Gaussian distribution. In bitshift theory, total bitshift refers to the movement of the magnetic transitions with respect to the position where they are recorded. There are three dominant factors which contribute to the loss of window margin, i.e. the difference between the farthest shifted bit and the actual window boundary. The intersymbol interference, which is a function of recording components and code used, the pulse detector imperfection, which is primarily caused by

equalization and differentiation errors, and the phase locked loop accuracy, which involves its inherent window skew and jitter. The first two contribute to bitshift while the last one reduces the effective window width. Fortunately, the fraction of window loss due to the PLL window skew can be substantially nullified, and excessive bitshifts may be compensated during read mode via the strobe function in high performance data synchronizers such as the DP8459.

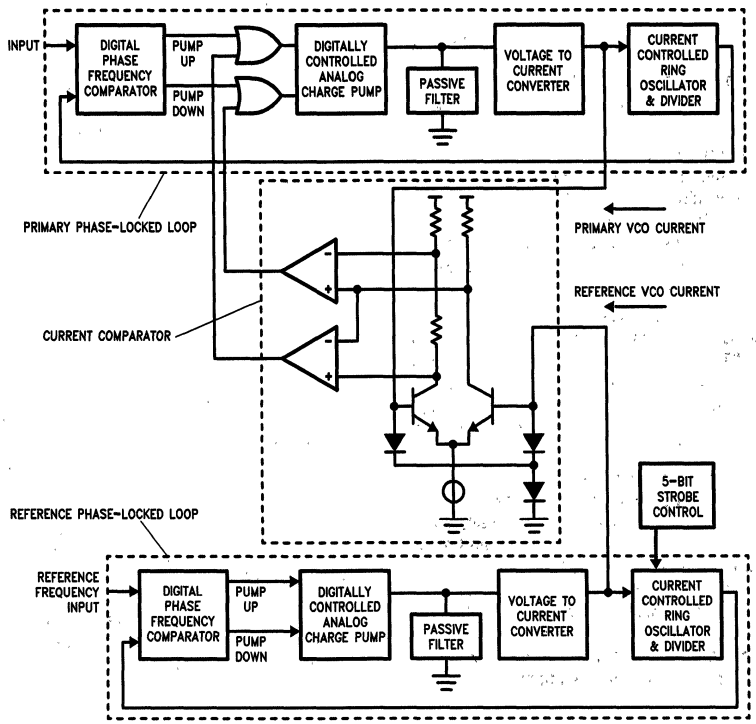
The first three LSBs of the Strobe Control Word produce strobe steps that have shown to track quite well with respect to the predicted values. For example, at 10 Mb/s data rate, typical characterization data indicates less than one nanosecond deviation between the measured strobe readings and the corresponding calculated strobe values. The data is taken with the Strobe Control Word range from M = 0 to M = ±6 and with the chip operating at 4.75V V_{CC} and 5.25V V_{CC}. It is recommended that this strobe range (M = 0 to M = ±6) be used for applications requiring relatively accurate strobe step control, such as for window alignment, data recovery, and window margin test. The higher strobe ranges are not as accurate due to the cumulative error when more bits of the control word are turned on. Thus for $|M| \geq 7$, this range is perhaps more suitable for inducing excessive soft errors in system analysis and experimentation purposes. System designers can thus perform real time system optimization studies to identify and to correct anomalies within the read channel chain. For example, after creating a significant amount of soft errors, any component within the read channel may be changed or modified to determine if error reduction can be achieved.

DESIGN CONSIDERATIONS

It is important to note that changing the strobe setting requires a finite response time of the control circuitry. In addition to the time required to load the 5-bit Strobe Control Word to the internal register the user must account for the settling time associated with any change of the strobe. This is a function of the Timing Extractor Filter (TEF) components and the data rate at which the data synchronizer is being operated. It is highly recommended that any change to the strobe setting be done with the Read Gate deasserted and with sufficient time allowed for settling prior to the initiation of another read sequence.

The Time Extractor Filter is used in a second PLL, the reference phase-locked loop, within the chip. This loop stably locks to a crystal reference oscillator (or a servo derived) frequency reference and it is responsible for producing a delay time of exactly one-half of the VCO period. Via this delay, the synchronizer data window is accurately centered about the mean bit position such that optimal capturing of the data becomes possible in the presence of jittery data pulses. Furthermore, strobing, which modifies the one-half VCO period delay, is achieved by programming a small amount of change in current (sourcing or sinking) to the current controlled oscillator of the reference phase-locked loop.

The VCO circuit of the reference PLL is constructed identical to that of the primary PLL. To prevent VCO frequency-control-voltage runaway in the primary loop, a comparator circuit is connected between the two loops to sense when the primary oscillator current crosses thresholds, which are placed 50% above and below the reference current. (Please refer to the simplified block diagram in *Figure 5.*) If



TL/F/10251–7

FIGURE 5. Reference and Primary PLLs within the DP8459

either one of the preset threshold is crossed, the comparator directs a correction signal to the primary PLL and limits further excursion of its VCO control voltage.

Since the reference PLL is part of the feedback loop that regulates the primary PLL, the Time Extractor Filter components also affect the synchronizer performance, but not the strobe step amount. In the (March 1988) DP8459 datasheet, *Figure 10*, presents a table of the estimated settling times due to a phase step for different (TEF) loop component values operating at various data rates. The calculated numbers in that table reflect the idealized settling time of the reference PLL block only. Assumptions made in the calculation, as empirical data was not available at the time of the publication, were no initial frequency offset, the values of the loop gain and the loop's natural frequency were approximately equal, and the effect of any parasitic conditions were neglected. Current lab data shows the TEF settling time from a maximum strobe movement (a step from M = −15 to M = +15) is nearly 15 times longer than the figures projected in the datasheet. [i.e. at RFC = 10 MHz, T_{SETTLE} ≅ 290 μs (measured) instead of 19.6 μs as predicted; and at RFC = 20 MHz, T_{SETTLE} measured is approximately 140 μs versus 9.6 μs as calculated.] In practice it is expected that customers would normally use the strobe range of $|M| \leq 6$ for reasons as mentioned in the previous section. Typical lab data for a strobe movement from M = −6 to M = +6 are also included here for reference: with RFC = 10 MHz, T_{SETTLE} ≅ 130 μs and at RFC = 20 MHz, T_{SETTLE} ≅ 54 μs.

Complete settling of the synchronized outputs at the primary loop may require slightly longer time. Although this settling time can be substantially reduced by either raising the bandwidth of the TEF or increasing the damping of the filter via RT1, this usually degrades the static window margin (by up to a few percent of the ideal window width). Since the window margin is an important parameter, it should not be compromised with the settling time. In a multi-data rate system it is recommended to employ the TEF components associated with the lowest data rate. In general the TEF settling time is not a critical parameter except perhaps for test time considerations when multiple strobing is involved (i.e. in margin testing in the production line). Other than the highest performance and the fastest systems, the latency time due to such system factors as soft-sectoring, command instruction delays, error correction/retrys, etc. in typical hard disk storage drives may be much longer than the TEF settling time. Therefore changing strobe settings from sector to sector is not practical in many hard disk environments. It is advisable that even for systems that are not limited by their latency time between read executions to allow a minimum of one revolution time (16.7 ms) between strobing. It should be mentioned also that the customer who wants to change the TEF settling time must observe stability criteria as in any PLL system.

APPLICATIONS

Individual "Trimming" of the PLL

For mass storage systems operating at low data rates (e.g. 5 Mb/s or less), two or three nanoseconds of window loss due to skew may be an acceptable specification. But, at higher data rates, storage systems can not tolerate such figures. As mentioned in the previous section, a portion of the window loss is ascribed to window skew. Although most of this loss may be recovered, undertaking the trimming of the synchronizer block is often an unpleasant manufacturing issue; presently this involves a technician in the assembly line adjusting a potentiometer while monitoring an oscilloscope. In high performance drives it is always desirable to regain as much margin as practical in production. The ability to perform in-system window deskewing is an extremely attractive attribute because it can attain the potential window margin available by optimizing each drive individually. It should be noted that this is superior to having the PLL chip statically adjusted prior to system integration. Of course such an in-system trimming procedure usually tags on a relatively high premium, such as added cost of time, labor, and material. Other than discrete designs, monolithic PLL devices generally afford limited or no adjustment to nullify window skew. The DP8459 synchronizer's Strobe feature presents a viable solution to perform window adjustment easily and inexpensively. The digitally controlled strobing within the chip allows "trimming" of the detection window without the need to "tweak" any external components. As mentioned, its 5-bit resolution strobe function makes it possible to deskew the inherent window offset with subnanosecond precision at 10 Mb/s and higher data rates.

Window Centering Algorithm

Truncation due to inherent window shift can be mostly nullified via the window strobing technique. Hence, an intelligent drive system can greatly benefit from the DP8459 Strobe feature because a window centering algorithm can be installed with no extra hardware or adjustment required. Moreover, the task of window deskewing can be readily automated with the DP8459 via the MICROWIRE bus. For example, some of the industrial interface standards such as the Rev-2 "ESDI" interface standard already supports a 4-step Early/Late strobe option in the command level, independent of the strobe step size to be implemented. The following describes a strobing alogorithm that can be employed for window centering.

A typical window centering routine should establish some higher than nominal error rate thresholds by strobing the data window in the early direction, then repeating the same process in the late direction. The early and late strobe settings which yield the equivalent error distribution are then stored. From the strobe excursion information, the window center skew can thus be determined and the appropriate strobe word is set to correct the window skew. To implement an effective routine one should employ a periodic test pattern such that the average decode window is made more stable. The system error rate can be deliberately made more responsive if some constant (time independent) source of window degradation functions such as maximum bit crowding or reduced signal to noise ratio is introduced during testing, i.e. performing the test at the inner recording cylinders of the Winchester type disk drive. Such a routine could execute during system power-up and produce optimal centering of the window. The system would function like having a built-in tester to perform window auto-calibration at power up or at any scheduled maintenance interval.

Data Recovery

Another valuable application of the DP8459 Strobe function is to service error-bound data recovery. Infrequently, the need arises to rescue vital, but marginally recorded, data

information from a removable data cartridge, and in particular from a (magnetically) damaged or defective storage module. Usually, such an operation requires repetitively reading through one block of data at a time and exhaustively shifting through some range of window strobe steps in an attempt to correctly retrieve the data of interest. However, employing the DP8459 in conjunction with a suitable data retrieval algorithm, could offer any mass storage system a powerful tool for speedy recovery of error-bound data. The DP8459 device is capable of delivering typically a 1.8% window shift resolution, furthermore, its thirty-one steps of strobe setting is programmable via the MICROWIRE bus. Such features make it possible to incorporate sophisticated utility programs such as one for data recovery.

Window Margin Test

The technique of Strobing employed in window margin testing is not a new practice. Shifting the detection window is one of two acceptable means to "marginalize" the window (modify the VCO generated data window) for checking the merit of the read channel. Although it has been gaining popularity in the test equipment sector, the theory behind it may be unfamiliar to some drive users and manufacturers alike. Window margin testing is an extension of the window centering process as discussed above. Please refer to Appendix A for a discussion of window shifting and window narrowing techniques in error-rate analysis.

Window strobing can be a very time-saving method to analyze the drive system's error rate characteristics. Unlike conventional testing, it does not consume hours to transfer data and to perform data integrity comparison. Hence, the DP8459 Strobe Function is very convenient for this purpose. For example, in a manufacturing environment, a new data storage drive system will be thoroughly characterized for its error rate profile with independent test techniques. Then it will be followed by a series of window shift induced error rate tests. The accelerated error rate profile thus generated is next compared to those produced from other test methods. Error distribution, statistical correlation, and acceptable thresholds are thus established. In production then, the correlated error rate figures and their corresponding Strobe settings are stored permanently in the memory of the drive/system during final checkout. These statistics are subsequently utilized as criteria for performance acceptance or rejection such as in QA and incoming inspections. Furthermore, systems in the field can be routinely interrogated for their current window margin status.. This, for example, can take the form of an embedded system maintenance routine to reduce the potential hazard of an unexpected system crash predicament. An effective window margin diagnostic test routine should employ the most bitshift sensitive test pattern and operate at the maximum bit crowding region of the media.

SUMMARY

The Window Strobe feature of National's latest PLL data synchronizer chip, the DP8459, has been presented along with the background information necessary for its utilization. The versatility and power embedded in this digitally controlled Strobe function is unparalleled. Although the prime intent of strobing is to deskew the inherent window asymmetry thus improving the device window margin, it also lends itself to a host of important system applications. They include in-system calibration of window centering which also allows adjustment of the individual drives to maximize their performance margin; system window margin analysis in design optimization and system maintenance; and data recovery on damaged media without the need of dedicated test equipment. The DP8459 strobe feature provides an economic and reliable solution to enhance the value and performance of disk drive designs, at the same time making the products more cost effective to manufacture.

Acknowledgement

The author would like to acknowledge the constructive criticisms, suggestions, and encouragement offered by Patrick Tucci in the preparation of the manuscript. He also thanks Gray Tietz for his comments to improve the presentation of this note.

References

1. "Effects of Bitshift Distribution on Error Rate in Magnetic Recording", E. R. Katz and T. G. Campbell, *IEEE Transactions on Magnetics* Vol. MAG-15, No. 3, May 1979.
2. "Phase Margin Analysis", M. Monett, Memory Technology Inc., 1980.

Appendix A

The purpose of this appendix is to present some basic information on a few commonly used tools and methodologies in window margin analysis. It will derive a mathematical expression describing the relationship between window shifting versus error rate probability. It will be shown that this expression is equivalent to that which relates to the window narrowing method, which is a well publicized technique to analyze the detection window margin.

There are two prevalent methods in the industry to qualify the window margin of disk drives, the margin that remains when the worst case jitter on the Encoded Read Data bits is subtracted from the non-ideal window (shifted and/or truncated). One method is to ascertain the average pulse distribution from the encoded data stream output. It usually involves a precision time-interval acquisition apparatus which measures the average pulse separation read from a particular track that has been preconditioned with a worst case data pattern. Another method is to measure the accelerated soft error rate induced by modifying the data window derived from an accurate (discrete designed) PLL circuit. Most disk drive testers employ either one of these methods for window margin test. It should be pointed out that testers used in research and development are usually built with the variable window width design, because they lend more sophisticated testing and render additional useful information such as system signal to noise ratio, media defects, and resolution of the head/disk components. It is also interesting to note that because an external PLL system is required, such a tester is generally not capable of checking the data synchronizer block in a disk data storage system.

Since a PLL system with programmable detection window width is expensive and difficult to build, nearly all commercial disk drive testers for manufacturing and end user applications do not employ such technique. Instead, they employ other methods such as measuring the average pulse distribution or the average synchronization window width. Although strobing is also incorporated in some drive testers, they contain but only a few strobe steps. Therefore, in window margin analysis their result can not be compared to the more extensive data gathered from the variable window width method. This trend, however, is changing as engineers are turning to more sophisticated strobing in both disk drive and drive tester designs. If a window shifting technique can mimic window narrowing in error rate response, it can be applied to window margin analysis also. The following discussion presents an accepted model used in the industry to describe error rate probability versus window width and shows that an equivalent expression also holds for the case of a shifted window.

As mentioned in the Strobe Function section, a Gaussian distribution describes error rates due to bit shift. Two such distributions are needed to define the error probability. One distribution is associated with the positive bitshifts and one is associated with negative bitshifts. This is due to the fact that for adjacent bits, one bit is pushed in the Early direction and one is pushed in the Late direction. The residual bitshift is denoted as offset or "To". The width of each of the distributions is defined as "Tne" which describes the broadening of the distributions due to the random noise of the environment (please refer to *Figure 6*). Hence, the appropriate Gaussian function, defined as N, for the two distributions can be written as:

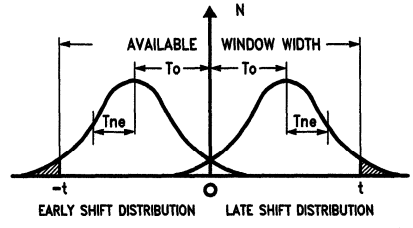
$$N = \frac{No(K)}{2}\left[\exp\left(-\frac{(t-To^2}{2\,(Tne)^2}\right) + \exp\left(-\frac{(t+To)^2}{2(Tne)^2}\right)\right] \quad (1)$$

where "No" is the number of bits read in a revolution of the disc, "K" is the normalizing constant to normalize each of the resulting error constants to unity when the distributions are integrated, and "t" is the time associated with the bitshift distribution. The equation for the error rate, which is simply the area of the tails of the bitshift distribution beyond the available window width, can be written as:

$$\text{Error Rate (Window-Narrowed)} = \frac{K(No)}{2}\left\{\left[\int_t^{\infty} \exp\left(-\frac{(t-To)^2}{2(Tne)^2}\right)\right] dt \quad (2)$$

$$+ \left[\int_{-\infty}^{t} \exp\left(-\frac{(t+To)^2}{2(Tne)^2}\right)\right] dt\right\}$$

$$= No/2\left\{\text{erfc}\left[\frac{t-To}{Tne}\right] + \text{erfc}\left[\frac{t+To}{Tne}\right]\right\} \quad (3)$$



TL/F/10251–8

FIGURE 6. Error Probability from Early/Late Shift Distributions

where "erfc" is the complementary error function and "t" is the half-window width at which the error rate is observed. An error rate versus window width curve takes the form represented in *Figure 7*. This is obtainable for the case where the data window is modified by narrowing the window while keeping it centered about the phase-locked oscillator expected window. This technique is typical of engineering drive testers employing an external PLL system for window margin analysis.

Consider if the window is shifted, instead of being narrowed, the limits on the second integration of equation (2) are changed and the corresponding "erfc" term becomes:

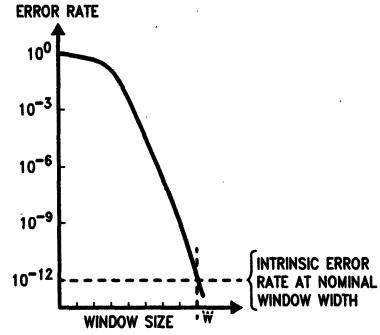
Error Rate
(Shifted Window)

$$= \frac{K(No)}{2}\left\{\int_{t}^{\infty}\left[\exp\left(-\frac{(t'-To)^2}{2(Tne)^2}\right)dt'\right]\right. \quad (4)$$

$$\left.+\int_{-\infty}^{t-T_\omega}\left[\exp\left(-\frac{(t'+To)^2}{2(Tne)^2}\right)dt'\right]\right\}$$

$$= No/2\left\{\text{efrc}\left[\frac{t-To}{Tne}\right]+\text{erfc}\left[\frac{Tw-t+To}{Tne}\right]\right\} \quad (5)$$

This corresponds to the bit error distribution evaluated with the average window having nominal width of "Tw", and "t" being the time position of the window's right-side boundary. In theory, the techniques with window narrowing and that of window shifting are equivalent. Empirical results from these two approaches should correlate if the incremental change in window displacement or window size employed is the same in both cases. Typical error rate versus the amount of shifted window with respect to the mean bit distribution should be similar to the profile depicted in *Figure 7*. A means to implement the shifted window scheme via the DP8459 Strobe function for margin testing is discussed in the text. Unlike the variable PLL-window technique, this method presents a bootstrap test methodology wherein the PLL synchronizer in the system is included in the window margin test.



TL/F/10251–9

FIGURE 7. Error Rate Distribution vs Window Width

DP8459 Zoned Bit Recording

National Semiconductor
Application Note 599
Kern Wong

Driven by current progress in micro-processors, small information processing machines with mainframe-like power are rapidly moving toward storage intensive areas such as sophisticated file servers, photorealistic graphics systems, and engineering work stations. Disk drive makers are constantly trying to keep up with the increasing demand for higher capacity modules. The quest to store more and more data within the same hard disk assembly form factor has reached a point where advances in head and disk technology alone are not adequate to raise the areal density as quickly and economically as the industry wants.

Most of the high performance 5¼ in. Winchester drives today employ up to 15 Mb/s data transfer rates (RLL codes) and some even use ultra high resolution head/media in order to achieve a 700+ Mbytes capacity. Nevertheless, disk drive makers continue to seek out new ways and viable techniques to reach even higher capacity storage modules. While new methods of recording and higher performance recording components are being researched, other means of improvement using current disk technology are being investigated. One of these schemes is zoned-bit recording (ZBR) which is showing very respectible results. This method allows for packing more data on a disk surface using conventional head/disk components. The following discussion reviews the basis of ZBR and presents a design application using the DP8459 data synchronizer to simplify ZBR implementation.

ZBR BOOSTS DENSITY BY RECAPTURING VAST DISK AREA INEFFICIENTLY UTILIZED

ZBR came from an earlier idea called constant (linear) density recording, or (CDR) which pioneers of the disk community had toyed with nearly two decades ago. Conventional 5½ in. Winchesters usually employ a single data transfer rate such as 5 Mb/s, 10 Mb/s, 15 Mb/s, etc. The linear density of a given design is dictated by the maximum number of flux transitions that can be placed end-to-end on the inner-most data track of the disk surface. Since the circumferencial length of any track is proportional to its radius, the outer tracks become increasingly more loosely packed with data bits. As a result, a large portion (over 90%) of the disk surface in typical drives is not effectively utilized; i.e., valuable areal density is wasted.

The goal of CDR was to write the same linear bit density (constant flux change per unit length) to every track on the disk data surface. The reading and writing of data occur at frequencies which increase as a function of the diameter of the disk. This requires that multiple data rates be used, with each track operating at a different transfer rate, the outer tracks employing higher data rates than the inner ones. True CDR implementation can bear a substantial design/manufacturing overhead, e.g., sophisticated sector management is required since the number of sectors are different in each track, also non-conventional servo data is needed to accommodate the different data rate used for each track. Such a system demands additional firmware and software design. And because there are as many data rates used as the number of tracks employed in the disk drive, more elaborate tests are involved during manufacturing. Furthermore, the minute gain in recording capacity for using slightly different data rates in adjacent tracks might not be practical from a system's point of view. System overhead for a CDR drive thus might not be cost effective for current disk drive systems.

Zoned Bit Recording, on the other hand, is an engineering compromise to CDR. By adopting a more conservative approach, the ZBR method divides the disk data surface into a number of concentric bands called zones, with each one consisting of several tens to hundreds of tracks. The drive thus writes more data in zones on the outer edge of the disk (employing higher data rates) than in zones toward the spindle center. This is a design compromise where a substantial gain in areal density can be achieved with a much lower design and manufacturing overhead. Please refer to *Figure 1a* and *1b* which depict the bit distribution from a pre-recorded periodic pattern over different tracks.

The reemergence of CDR or ZBR was facilitated by current advances in the LSI read channel electronics. In particular, the monolithic data synchronizer/data separator integrated circuits have been a significant driving factor. About five years ago such a function (for 5 Mb/s data rate only) would have typically occupied a good portion of the electronic card cage in a drive system. Today many chip design houses can produce the core of the data synchronizer function in a single chip. However, few commercially available devices could meet the needs for high performance CDR and ZBR implementations. The DP8459 data synchronizer stands out with unique features specially designed to address this application.

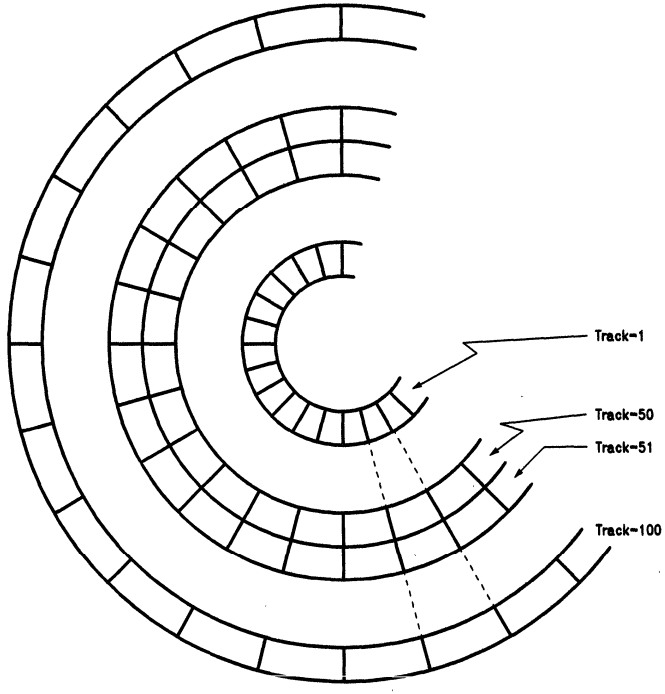
DP8459 INCORPORATES FEATURES FOR ZBR

The key features of the DP8459 which make it ideal for multiple data rate operation reside in the Timing Extractor and VCO sections of the chip. They enable a broad range of data rate operation, plus the external support components require minimal programming. (Please refer to *Figure 2.*)

Conventional synchronizer chips may require a passive delay line to perform the task of window alignment. For ZBR designs that means many delay modules are needed to be multiplexed for operation requiring different data rates. Furthermore, they may have to be custom made to obtain those precise and non-standard delay values. For higher performance data synchronizer circuits and particularly for applications involving higher data rates, some PLL chipmakers use external resistors to control duty cycle of the VCO waveform. This approach utilizes the opposite (rising and falling) edges of the VCO output as a delay line. But, such chips may still require the trimming of an external resistor to attain optimum duty cycle symmetry for the desired delay time. Moreover, they would invariably require the switching of an external bank of precision resistors in ZBR.

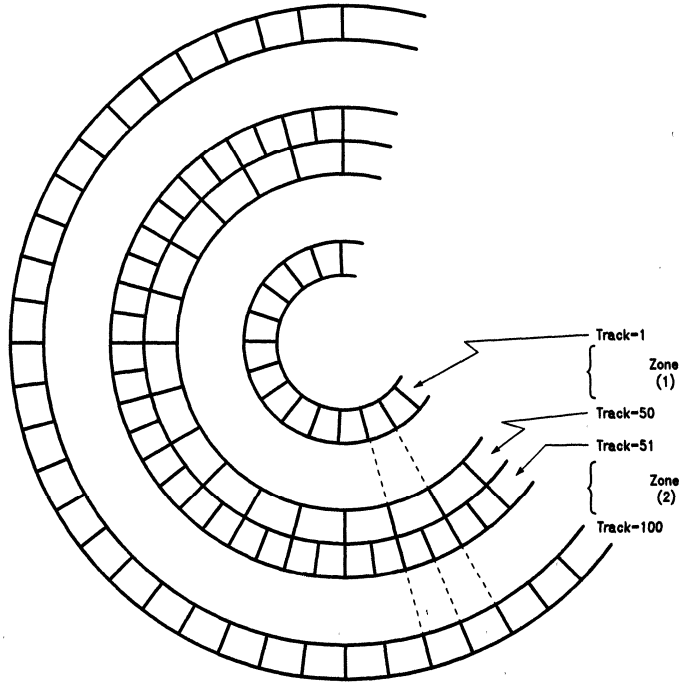
The timing extractor block of the DP8459 represents a second PLL operating in tandem with the main PLL of the device. It is an accurate variable active delay line whose function is to establish synchronization window alignment so that the expected data bit is optimally centered about its decode window. The resultant delay is nominally equivalent to one half of the period of the reference clock.

The DP8459's built-in tracking silicon delay line completely obviates the need to switch external components. It takes



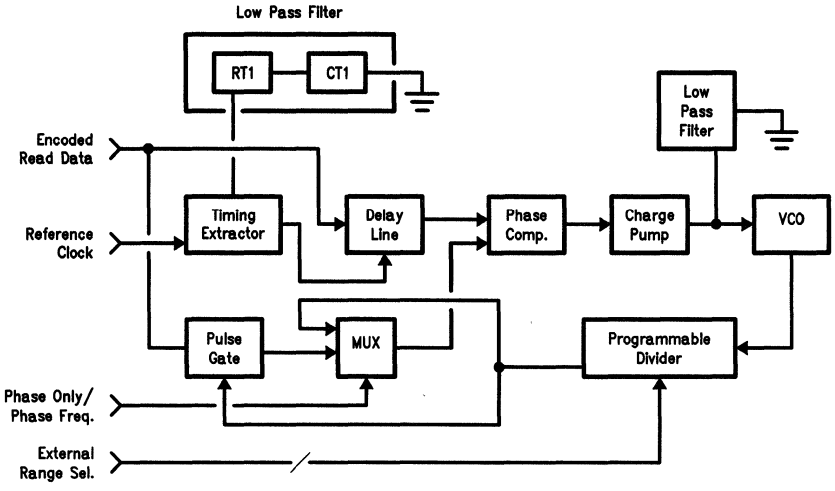
TL/F/10379–1

FIGURE 1a. Single Data Rate



TL/F/10379–2

FIGURE 1b. Multiple Data Rates



TL/F/10379–3

FIGURE 2. Simplified Block Diagram of the DP8459 PLL

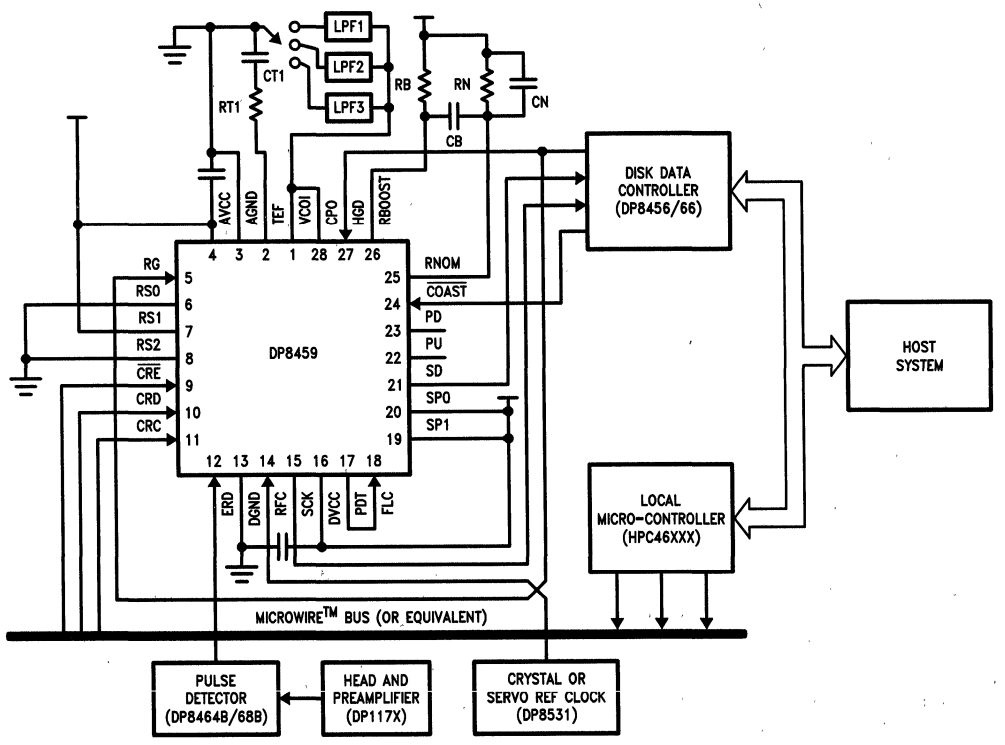
full advantage of the stable timing relationship from the reference clock input. The reference clock can be a crystal oscillator, a closed loop servo clock, or a programmable clock generator chip such as National Semiconductor's DP8531. The resistor-capacitor (Timing Extractor Filter) shown in *Figure 2* is employed for stabilizing the secondary PLL. The delay line performance is strictly a function of the reference clock accuracy, and is insensitive to the external components associated with the extractor as well as to supply voltage, temperature and IC process variations. Furthermore, the Time Extractor Filter (RT1 and CT1) components need not be changed for different data rates in ZBR. Simply choose the manufacturer's suggested values associated with the lowest data rate employed.

The DP8459's unique "auto ranging VCO" design provides another important advantage for ZBR. The reference clock frequency also sets the "center frequency" of the VCO automatically for the selected data rate, thus, no external components are needed for the VCO. The VCO section of the synchronizer consists of a high frequency oscillator and a programmable modulus divider. This arrangement provides a continuous range of VCO operating frequencies from 500 kHz to 50 MHz (range selection is via a 3-bit word input). The data rate range extends from 250 Kb/s to 25 Mb/s with MFM, [2, 7], and [1, 7] codes and from 250 Kb/s to 10 Mb/s for GCR codes. The DP8459 again simplifies multiple data rate applications as it requires no adjustment of any external VCO resistor/capacitor, LRC tank, or separate VCO circuitry. This not only facilitates design and testing of products, it also reduces unwanted noise from coupling to sensitive nodes. In addition, with ZBR designs operating over a 2:1 VCO frequency range, users may not need to change the range select control word, which further simplifies the design.

ZBR DESIGN CONSIDERATIONS

Unlike the controller section of a ZBR design, whose firmware and software can become more complex in order to handle a different system of sectoring, the DP8459 PLL bears practically no additional overhead. Whether one wishes to produce a system with 8 zones or 64 zones, it is important to recognize that the DP8459 synchronizer eases drive system designs by reducing the number of external components required and eliminating the need for trimming and switching critical components. ZBR implementation is simple with the DP8459, as is shown by the system block diagram in *Figure 3*. For ZBR operation, the Reference Clock input frequency must be changed whenever a different data rate (zone) is used. This can be accomplished with a crystal based programmable clock chip, such as National's DP8531. If the input is derived from a PLO (closed loop servo clock) the Reference Clock frequency would automatically update to the proper frequency as the servo head positions to a different zone.

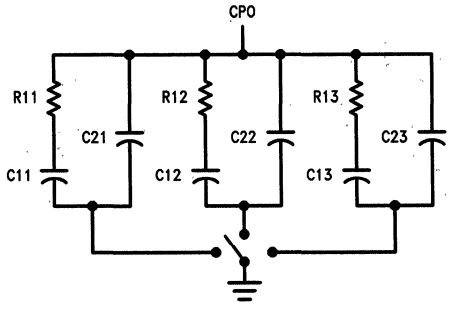
In general it is not necessary to employ different loop filter components for every zone, especially if the adjacent zones are designed to operate over a small range of data rates. A single (compromised) 2nd order low pass loop filter should deliver acceptable performance margins, even when operating with a 2:1 ratio of data rates, for most of the drive designs. Of course there might be some designers who prefer to use more exotic filter configurations and/or to switch several loop filters to optimize system performance. If one requires switching filters, the following discussion is presented for consideration.



TL/F/10379–4

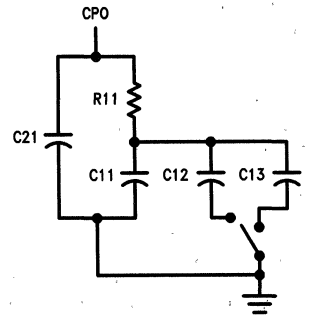
FIGURE 3. Sample ZBR Drive System Employing the DP8459

Figure 4a shows multiplexing 3 complete sets of simple loop filter components via a switch. Alternatively one can modify the loop filter's characteristic BW or damping by changing only one of its resistor or capacitor values (see *Figure 4b*). Such a configuration can also produce an effective compromised loop filter which can handle a large number of data rates. (This scheme can help trim cost and minimize circuit board real estate.)



TL/F/10379–5

FIGURE 4a. Switching of 3 Complete Lead-Lag Loop Filters



TL/F/10379–6

FIGURE 4b. Modifying a Simple Filter with a Capacitor

Multiplexing several loop filters may be accomplished to advantage with micro-miniature mechanical relays. This is due to their superior isolation, lower leakage, and lower contact resistance compared to electronic switches. However, these advantages must be weighed against their slower response time and the need to suppress the relay coil's back EMF. Although "wet" contact relays are speedier and provide debouncing action, they can cost up to 10 times over conventional types. Analog (FET) switches have certain merits that make them suitable for switching of loop filters; these include long term reliability, fast response time, and high contact density per device. Furthermore, they lack the

back EMF hazards and do not consume much power to maintain closed "contacts". These features make them good candidates for multiplexing a large number of filter networks as in a ZBR application. However, some important device specifications must be examined before making a device selection. Commercial grade analog switches have a typical on-resistance of less than 40Ω, a terminal parasitic capacitance of 20 pF or less, and a leakage current of less than one nano-amp (but this spec can increase up to 50 times at high temperatures!). Nevertheless, they may offer acceptable performance as signal switches. The important issue is to determine the switch's performance under actual drive operating conditions and that typical loop filter characteristics can be maintained in the presence of parasitic elements. Another point to be aware of is that analog switches generally require bipolar power supplies, which may be unavailable in most drive systems. There are some instrumentation solid state relays which offer superior specs with attributes resembling those of mechanical relays, but their cost and contact density per device may not be cost effective for most present drive designs.

Since the DP8459 has provisions for a dual-port PLL filter network, higher order filters and active filter designs may also be used. This provides a third means to alter the loop filter characteristics conveniently via active filter implementation.

ZBR AND SINGLE DATA RATE DESIGN COMPARISON

Presently, high performance 5¼ in. Winchester companies are actively trying, with advanced head/media, to push recording densities to over 30k bit/in., track densities to over 1600 tracks/in., and are employing higher data transfer rates from 20 Mb/s to 24 Mb/s to achieve a Giga-byte capacity drive. It can be shown that without resorting to radically different technology and future generation recording components, this goal can be met by using ZBR techniques with conventional head/media.

The following design example compares a ZBR drive system and one using a single data rate; both assume moderate design parameters that are within current-generation disk technology:

The data rates and number of tracks used in each of the 9 zones for the ZBR design example above is listed below:

Zone	Tracks	Data Rate	Loop Filters
1	71	15.000 Mb/s	LPF1
2	71	15.600 Mb/s	LPF1
3	77	16.224 Mb/s	LPF1
4	80	16.873 Mb/s	LPF2
5	84	17.547 Mb/s	LPF2
6	87	18.249 Mb/s	LPF2
7	91	18.779 Mb/s	LPF3
8	94	19.738 Mb/s	LPF3
9	869	20.528 Mb/s	LPF3

Compared to the 15 Mb/s single data rate design, the ZBR design operating between 15 Mb/s and 20 Mb/s offers approximately a 30% increase in capacity. Operating the same system at a uniform data rate of 20 Mb/s is not feasible because the selected head/media characteristics (resolution limitations) would not have yielded acceptable performance over 40% of the inner tracks. At this increased data rate, higher grade head/media must be used to raise the drive's storage capacity. Although this ZBR design example needs to use higher data rates, the required read channel electronics to handle it are available and bear a lower system overhead compared to choosing the higher cost, grade, and less available (high bit density) media and (higher resolution and lower flying height) recording heads of today to support greater than 15 Mb/s single data rate Winchester disk drives.

Zoned-Bit Recording	RLL Codes	Single Data Rate Design (RLL)
Linear Density	20,404 Bits/In.	20,404 Bits/In. (Max)
Track Density	1,400 Tracks/In.	1,400 Tracks/In.
Data Surfaces	15	15
Data Rates	15 Mb/s to 20 Mb/s	15 Mb/s
Data Rate Spread	4% on Adjacent Zones	Not Applicable
No. of Zones	9 Zones	1 Zone
Loop Filters	3 Sets	1
Unformatted Capacity	1,016.3 Mbytes	721.5 Mbytes

CONCLUSION

The Zone-bit Recording concept is rapidly gaining popularity for the next generation of high capacity disk drive designs. The DP8459 high performance data synchronizer chip is truly designed to address the special needs of ZBR applications. It eases ZBR designs by eliminating cumbersome adjustment and switching of critical resistor/capacitor components for the VCO and delay line sections of the data synchronizer circuit. The combination of using the DP8459 device and ZBR techniques provides an economic and reliable solution to enhance the value and performance of disk drives, making them more cost effective to manufacture. This design methodology can achieve a significant increase in storage capacity while using existing disk technology. The DP8459 can be used in the manufacture of high performance, high capacity, multiple data rate flexible disk, optical disk, and giga-byte capacity 5¼ in. Winchester drives.

DP8459 Evaluation Board

National Semiconductor
Application Note 502
Kern Wong and Michelle Wong

I. INTRODUCTION

This literature describes the hardware and function of a printed circuit board for the evaluation of National Semiconductor's DP8459—All Code PLL Data Synchronizer. The purpose of the board is to assist the user of the chip in becoming familiar with the device features and in understanding its operation. The board is configured (socketed) for the 28-pin PLCC package (DP8459V) device. It contains the necessary connections and circuitry which greatly simplify the set-up of a test apparatus and facilitates the task of device evaluation.

II. CIRCUIT BOARD DESCRIPTION

Please refer to the schematic and layout captions in *Figures 1* to *4*. The circuit board layout follows the general recommendations discussed in Section 5 of the DP8459 data sheet. The entire circuit on the board can be described by the four subsections in the following text.

1. RESET FUNCTION: D1, Q1, Q2, and ½ of U1 (2-input NAND gates) form a single-shot circuit which generates a reset pulse during power-on. This signal clears the shift register, counters, and D-type flip-flops on board. Subsequently, it allows the selected STROBE circuitry control word (bits #0 to #4) and the test bit #5 to be loaded into the PLL's internal control register. The board also provides a manual reset function, which consists of switch SW-1 (SPDT momentary or toggle switch) and the remaining half of U1. It provides a reset signal while the chip is powered on.

2. LOAD AND SHIFT FUNCTION: U2 (2-input NAND gates), U3 (D-type flip-flops connected in toggle mode), U4 (8-bit parallel-in serial-out Shift Register), and U5 (divide-by-6 counter) constitute a 6-bit word programmable data generator. It issues a set of MICROWIRE™ equivalent outputs to program the control register in the PLL. U5 pin #8 is the Control Register Enable (CRE) signal. A logic LOW level allows data to be strobed into the registers. A logic HIGH level latches the register data and issues the information to the appropriate circuitry in the chip. U4 pin #9 to the Control Register Data (CRD) input shifts out a selectable 6-bit word. U3 pin #8 is input to the Control Register Clock (CRC) whose negative edge clocks the CRD pattern into the register.

3. REFERENCE FREQUENCY CLOCK: U6 (CMOS Inverter HC04) and a crystal form a stable reference oscillator which is required by the PLL for proper operation. This signal is applied to the RFC input of the DUT.

4. DIP SWITCHES: Three sets of toggle DIP switches are used to manually program the PLL input pins for the different modes of operation and for the various ranges of data rates and window strobe features.

(i) SW2—Switches #1 to #6 select the 32 combinations or steps of Early/Late window strobe position which is useful for the purpose of window skew compensation or recovery routines of marginally readable data. For normal device operation (non-test mode), the test bit (switch #6) must be set logic LOW.

The control register bit and toggle switch correspondence is shown below. Whenever the reset function is activated, the current SW2 setting is loaded into the control register. For detailed window step vs. strobe bit, refer to *Figure 4* in the data sheet.

SW2: Switches	#1	#2	#3	#4	#5	#6
Control Register Bit:	0	1	2	3	4	Test

(ii) SW3—The top three switches are used to select the VCO frequency range via the Range Select pins RS0, RS1, and RS2. The fourth switch is not used. The relative VCO frequency versus the 3 bit code is tabulated below:

Switch			VCO Range
#3 RS2	#2 RS1	#1 RSO	
1	1	X	0.50 MHz–1.25 MHz
1	0	1	1.25 MHz–2.5 MHz
1	0	0	2.5 MHz–5.0 MHz
0	1	1	5.0 MHz–10.0 MHz
0	1	0	10.0 MHz–20.0 MHz
0	0	X	20.0 MHz–48.0 MHz

Important—If the desired VCO frequency is located at the boundary of 2 ranges (overlapping), select the range which places that frequency at the higher end of the spectrum, (e.g. Fvco = 20.0 MHz, choose RS code = 010). This will optimize the performance of the chip.

(iii) SW4—The functions of the 8 switches are detailed as follows:

Switches #1 to #3 control the Frequency Lock Control (FLC). At most one and only one of the 3 switches should be engaged or closed.

Switch #1 closed—FLC connects to PDT

Switch #2 closed—FLC connects to RG

Switch #3 closed—FLC connects to GND

Switches #4 and #5 program the Sync Pattern Select Pins SP0, SP1 for the internal divider modulus to detect either the 1T, 2T, 3T, or 4T preamble pattern. (T = VCO period; e.g. 3T = 100100...preamble).

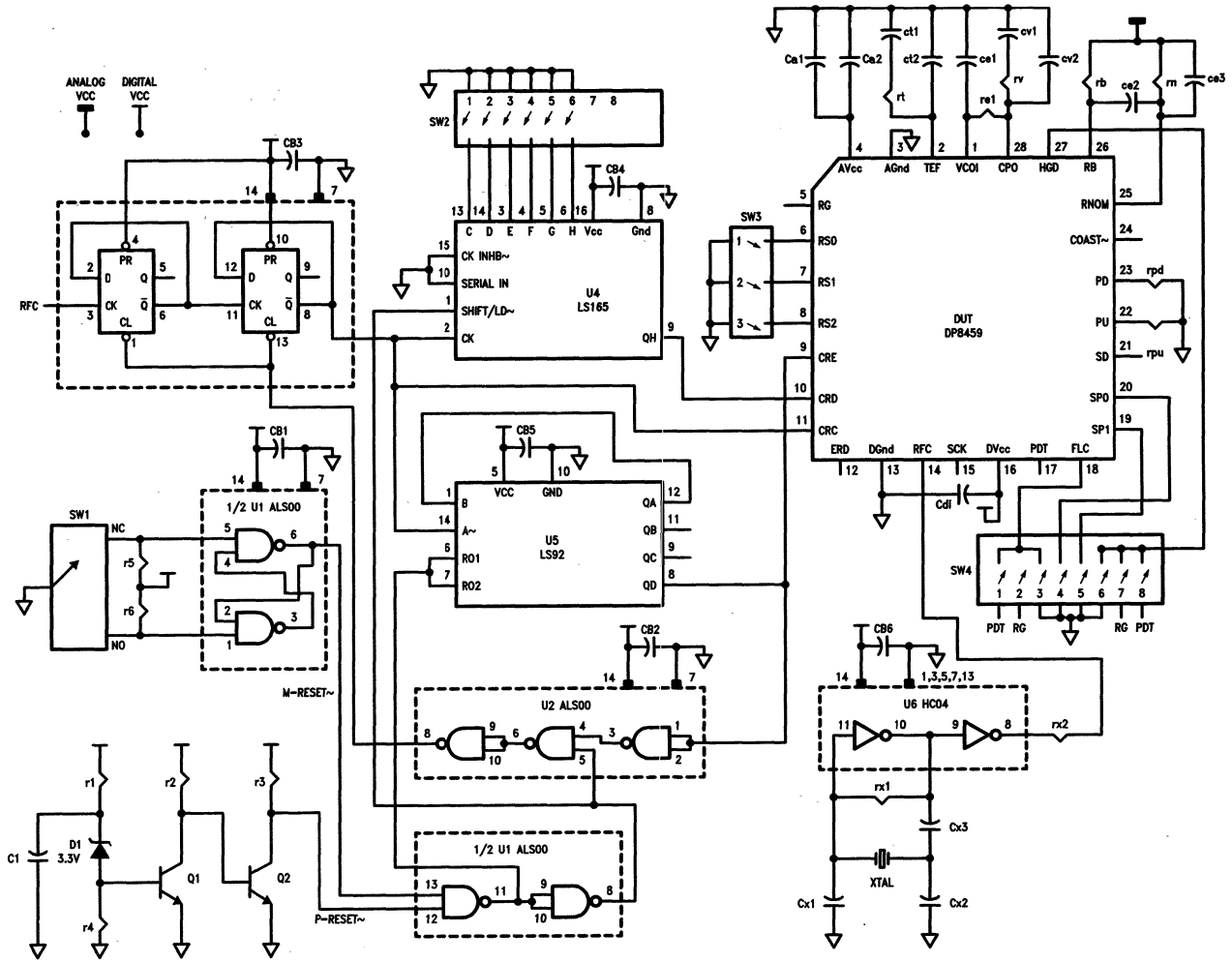
Switch		Sync Matching Divider Modulus M
#5 SP1	#4 SP0	
0	0	1
0	1	2
1	0	3
1	1	4

Switches #6 to #8 control the High Gain Disable (HGD) pin. Again, only one of these 3 switches should be closed at any time.

Switch #6 closed—HGD connects to GND

Switch #7 closed—HGD connects to RG

Switch #8 closed—HGD connects to PDT



TL/F/9385-1

FIGURE 1

III. TEST HARDWARE

1. EXTERNAL COMPONENTS: All integrated circuits are socket mounted to make it easy for the purpose of interchange or replacement. Passive components for the loop filters, charge pump resistors, and the crystal oscillator sections are also provided with sockets to facilitate alteration of components for different data rates.

2. POWER SUPPLY: Three separate supply terminals are provided on board. One ties to the Analog V_{CC} and a second ties to the Digital V_{CC} of the PLL chip. A third post makes connection to the rest of the digital support circuitry. Normally two +5V supplies are used, one to supply the V_{CC} pins of the PLL device under test, another to power the support circuitry. It should be noted that in order to take advantage of the power on reset function, the power of the PLL chip must be turned on first, then apply power to the support circuitry.

3. ENCODED READ DATA (ERD) AND READ GATE (RG): A word pattern generator is a convenient signal source to provide the desired TTL logic level signals to these inputs.

4. LOOP FILTER COMPONENTS: Normally components "Re1" should be replaced by a short-circuited connection and "Ce1" should be open-circuited for a second order loop filter implementation. These optional components are intended for higher order filter realization.

IV. DP8459 EVALUATION BOARD PARTS LIST: (10 Mbit/s 2, 7 code operation)

DUT = DP8459 (PLCC package)
U1 = 74ALS00
U2 = 74ALS00
U3 = 74ALS74
U4 = 74LS165
U5 = 74LS92
U6 = 74HC04

D1 = Zener Diode (Vz = 3.3V)
Q1 = 2N2369 (NPN)
Q2 = 2N2369 (NPN)

Xtal = 20 MHz

SW1 = SPDT Momentary Switch
SW2 = 16-Pin DIP Toggle Switch
SW3 = 8-Pin DIP Toggle Switch
SW4 = 16-Pin DIP Toggle Switch

Bypass Capacitors:
Cb1 = 0.1 μF
Cb2 = 0.1 μF
Cb3 = 0.1 μF
Cb4 = 0.1 μF
Cb5 = 0.1 μF
Cb6 = 0.1 μF
Ca1 = 0.1 μF
Ca2 = 1000 pF
Cdi = 0.1 μF

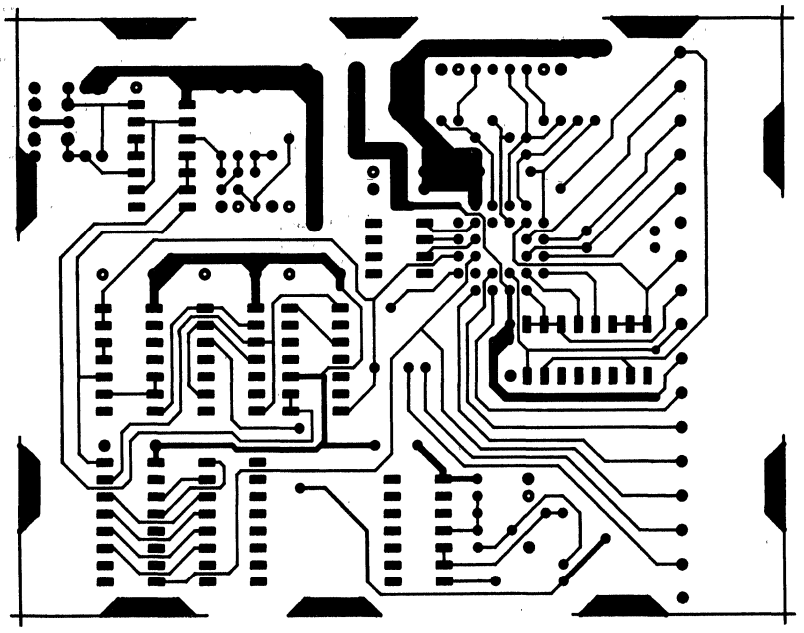
Loop Filters:
Ct1 = 0.05 μF
Ct2 = NC*
Rt = 68Ω
Cv1 = 0.022 μF
Cv2 = 510 pF
Rv = 150Ω

Crystal Oscillator:
Cx1 = 22 pF
Cx2 = 100 pF
Cx3 = 56 pF
Rx1 = 1 MΩ
Rx2 = 100Ω

Other External Components:
Ce1 = NC*
Re1 = Short Circuit It
Rn = 2.4 kΩ
Rb = 2.4 kΩ
Ce2 = NC*
Ce3 = 1000 pF
Rpu = 510Ω
Rpd = 510Ω

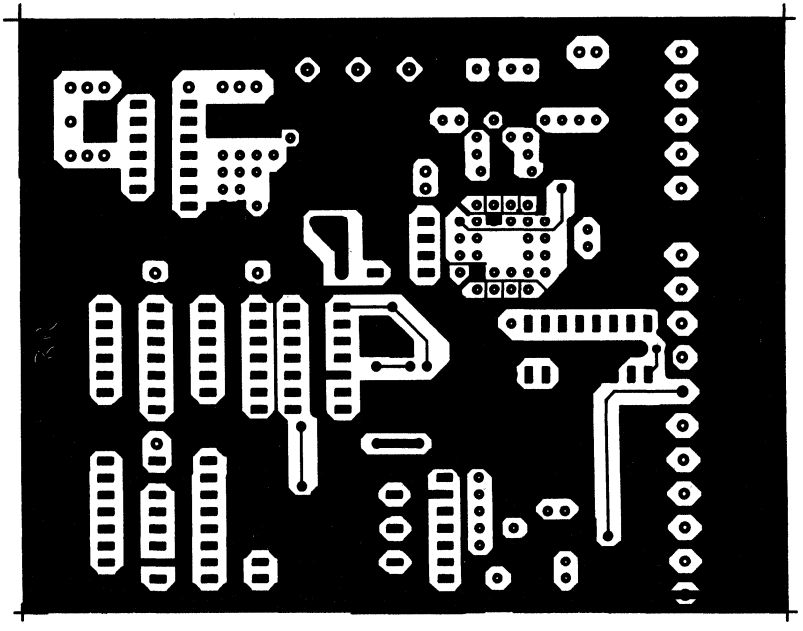
Reset Circuitry:
R1 = 2 kΩ
R2 = 10 kΩ
R3 = 10 kΩ
R4 = 10 kΩ
R5 = 1 kΩ
R6 = 1 kΩ
C1 = 10 μF

*NC means no component required there (leave nodes open).



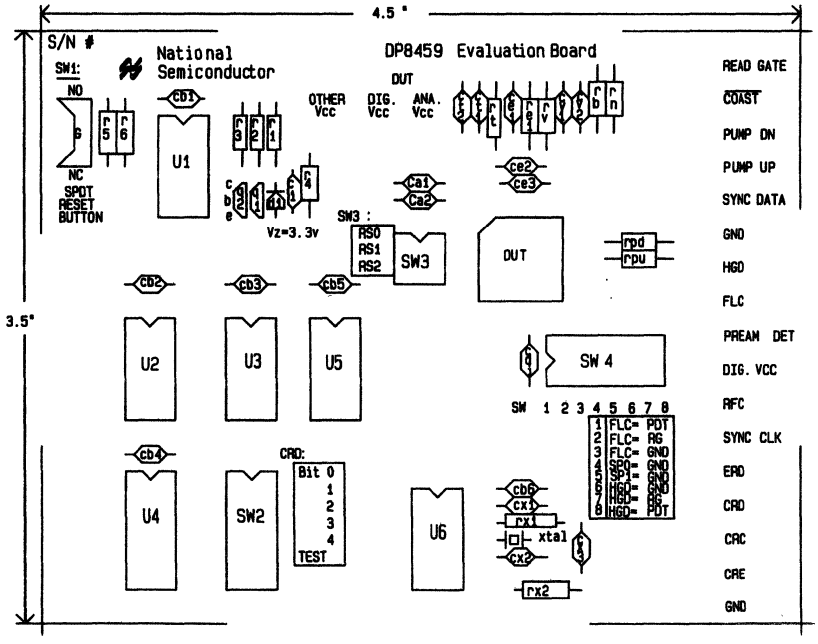
TL/F/9385–2

FIGURE 2. Trace Side (Bottom View)



TL/F/9385–3

FIGURE 3. Component Side with Ground Plane



TL/F/9385–4

FIGURE 4. Component Location (Top View)

Section 3

Rigid Disk Data Controller

Section 3 Contents

National Semiconductor

DP8466A Disk Data Controller

General Description

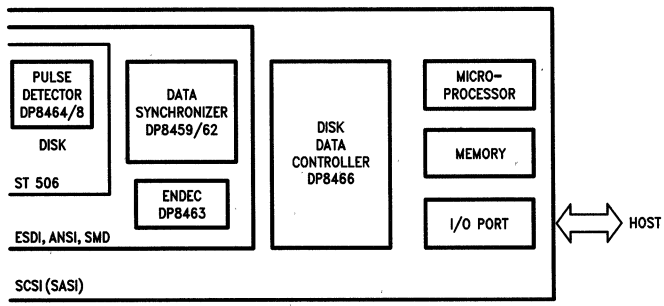
The DP8466A Disk Data Controller (DDC) is an intelligent peripheral which interfaces Winchester or Floppy disk drives to microprocessor based systems. It transfers data between a buffer memory or host system and the serial bit data stream with disk rates up to 25M-bits per second. High speed system data transfer is possible with full on-chip DMA control of buffer or main memory. The 16-bit system I/O interface allows use with any popular 8-bit, 16-bit or 32-bit microprocessor. Programmable track format enables reconfiguration of the DDC for different drive types in a multiple drive environment. Using other National DP8460 series disk data path chips, the DP8466A conforms to ST506, SMD and ESDI standard drive interfaces, as well as to intelligent standard interfaces such as SCSI (SASI) and IPI.

The DP8466A is available in three performance versions DP8466AN-12, DP8466AN-20 and DP8466AN-25.

Part Number	Max Disk Data Rate	Max DMA Transfer Rate
DP8466AN-25	25 Mbit/sec	10 Mbyte/sec
DP8466AN-20	20 Mbit/sec	8 Mbyte/sec
DP8466AN-12	12 Mbit/sec	6 Mbyte/sec

Features

- Easily conforms to any standard drive interface
- Compatible with floppy, hard and optical disk drives
- Compatible with 8, 16 or 32-bit microprocessor systems
- Programmable disk format
- Sector lengths up to 64k bytes, with up to 255 sectors per track
- Programmable 32 or 48-bit ECC polynomial
- Internal ECC correction in less than a sector time
- Disk data rate to 25M bits per second
- Multiple sector transfer capability
- 32 byte internal FIFO data buffer with interleavable burst capability
- 8 or 16-bit wide data transfers
- Single 32-bit or dual 16-bit DMA channel addresses
- Up to 10M bytes per second DMA transfer rate
- +5V supply, 48 pin DIP, microCMOS process



TL/F/5282–1

FIGURE 1. Typical System Configuration

Table of Contents

1.0 Introduction

National's DP8466A Disk Data Controller (DDC) chip is designed to concentrate only on the data aspects of a disk system, leaving the control signals to either a low cost single chip controller or an I/O port from a microprocessor. For this reason, the DDC will work with any standard drive interface.

The DP8466A is an advanced VLSI chip, fabricated in National's latest 2 μ CMOS technology, that allows for operation with disk data rates from the slowest floppy to the fast Winchester and Optical data rates of 25 megabits per second.

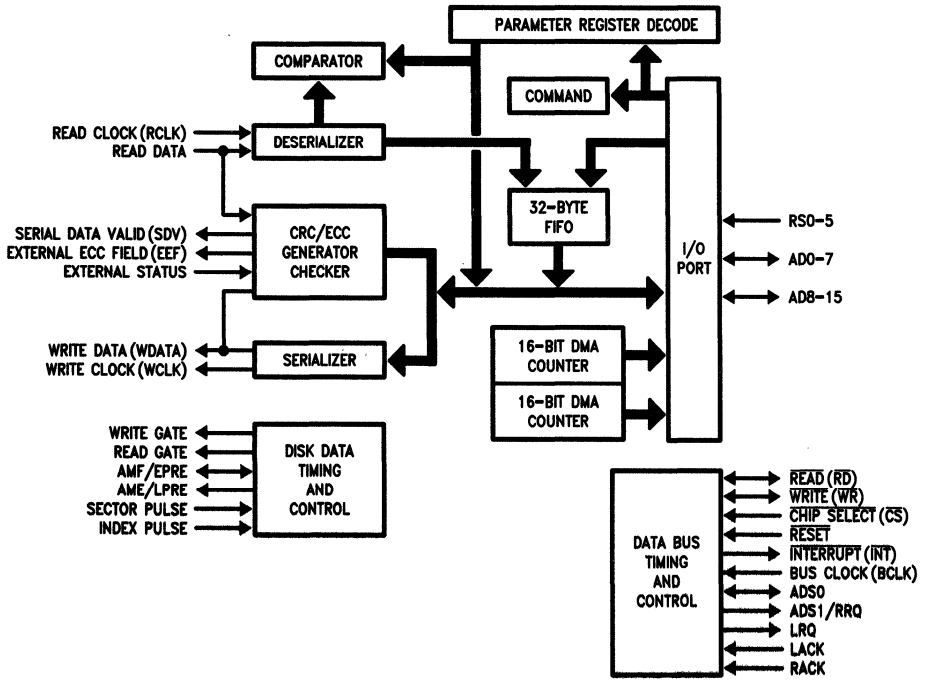
The CMOS design significantly helps the system designer because of reduced power consumption. The chip typically consumes 100 mW.

The DDC is designed for maximum programmability that not only allows the user to select any drive type he wishes, but also allows for different types of drives to be used on the same system. The chip contains 64 registers that can be loaded at any time by a microprocessor connected to the chip's bus. These registers determine the number of bytes in each field of the format, and the byte pattern that each of these fields will repeat. The number of data bytes per sector is selectable from 1 byte to 64k bytes. Finally, both the header field and the data field can each be appended with either a Cyclic Redundancy Check (CRC) field (the 16-bit code used on floppies) or a programmable Error Check and Correct (ECC) field.

The DDC allows the user to load in any 32 or 48-bit ECC polynomial from the microprocessor along with the format parameters. Once an error has been detected, the microprocessor decides whether to re-read the sector during the next revolution of the disk, or to attempt a correction. The DDC can correct errors in a time shorter than that required to read the next sector.

Key blocks in the DDC include a 32-byte FIFO and two 16-bit DMA channels that give the chip a 10 megabyte per second memory transfer capability. This high system data throughout is needed for the high speed drives now becomming available. The small FIFO allows for bursts of data to take place on the bus, thereby leaving the bus free for useful periods of time. The threshold for FIFO data storage is selectable to allow for some degree of system latency. The DDC allows for bursts of 2, 8, 16 or 24 bytes of data to be transferred between the FIFO and memory. The width of the data bus is selectable for either 8 or 16-bit transfers. The system designer selects the threshold so that when the FIFO contains the selected amount of data, the DDC will issue a request. The CPU can continue its operation and then stop to acknowledge the DDC, which then bursts the data between FIFO and memory, before the FIFO has time to overflow or underflow. With a 10 megabit per second disk data rate and a 10 megabyte per second memory transfer cycle, the bus will only be occupied for one-eighth of the time transferring data between FIFO and memory. This leaves the bus free for microprocessor usage for over 80% of the time.

Block Diagram

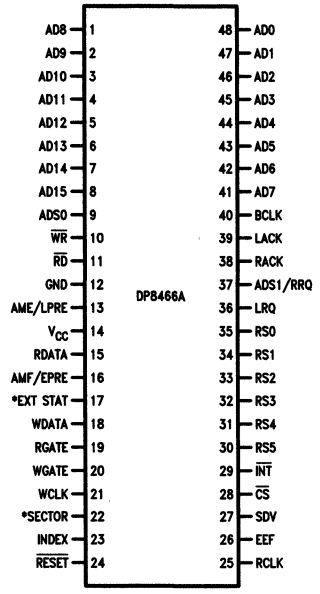


TL/F/5262-2

FIGURE 2. DDC

Connection Diagrams

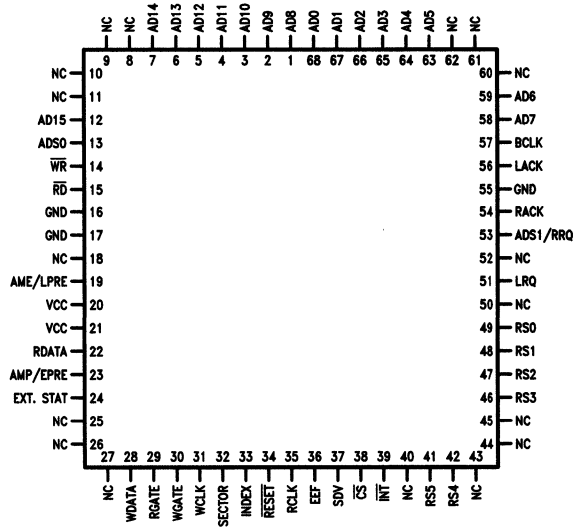
Dual-In-Line Package



TL/F/5282-3

Top View

*This pin must be grounded if not used.



TL/F/5282-76

Top View

Order Number DP8466AN
See NS Package Number N48A

Order Number DP8466AV
See NS Package Number V68A

FIGURE 3

2.0 Pin Descriptions

2.1 BUS INTERFACE PINS

Symbol	DIP Pin No.	PCC Pin No.	Type	Function
$\overline{CS}$	28	38	I	**CHIP SELECT:** Sets DDC as a standard I/O port for reading and writing registers. Configures $\overline{RD}$ and $\overline{WR}$ pins as inputs when DMA is inactive. This pin is ignored if on-chip DMA is enabled and performing a transfer.
$\overline{INT}$	29	39	O	**INTERRUPT:** An interrupt can be generated on any error, or after completion of a command, a correction cycle or any header operation.
$\overline{RESET}$	24	34	I	**RESET:** Clears FIFO, Status and Error registers. Halts DMA immediately. Halts disk read and write immediately. Does not affect parameter and most count and command registers. On power-up, must be held low for at least 32 RCLK cycles and 4 BCLK cycles. Note that both RCLK and BCLK must be active for the reset cycle to complete.
$\overline{RD}$	11	15	I/O	**READ:** • MICROPROCESSOR ACCESS MODE, with $\overline{CS}$ pin low and DMA inactive (RACK AND LACK low): Places data from FIFO or register as selected by pins RS0–5 onto the AD0–7 bus. • SLAVE MODE, with LACK pin high: Places data from FIFO onto the AD0–7/AD0–15 bus. • MASTER MODE: When DMA is active, $\overline{RD}$ pin enables data from the addressed device onto the address/data bus.
$\overline{WR}$	10	14	I/O	**WRITE:** • MICROPROCESSOR ACCESS MODE, with $\overline{CS}$ low and DMA inactive (RACK and LACK low): Latches data from AD0–7 bus to internal registers selected by RS0–5. • SLAVE MODE, with LACK pin high: Latches data from AD0–7/AD0–15 bus to FIFO. • MASTER MODE: When DMA is active, $\overline{WR}$ pin enables data from the address/data bus to the addressed device.

2.0 Pin Descriptions (Continued)

2.1 BUS INTERFACE PINS (Continued)

Symbol	DIP Pin No.	PCC Pin No.	Type	Function
BCLK	40	57	I	**BUS CLOCK:** Used as a reference clock when DDC is bus master. Used only during reset and DMA operations. Maximum ratio of RCLK/BCLK is 4 for Word Mode, and 2 for Byte Mode.
RACK	38	54	I	**REMOTE DMA ACKNOWLEDGE:** System input granting use of the bus for a remote DMA bus cycle. If RACK is de-asserted during a transfer, the current transfer cycle will complete.
LACK	39	56	I	**LOCAL DMA ACKNOWLEDGE:** System input granting use of bus for a local DMA bus cycle. If LACK is deasserted during a transfer, the current transfer cycle will complete. LACK has priority over RACK.
RS0–5	35-30	41, 42 46-49	I	**REGISTER SELECT:** Used as address inputs to select internal registers when $\overline{CS}$ pin is low.
AD0–7	48-41	58, 59 63–68	I/O	**ADDRESS/DATA 0–7:** These pins float if $\overline{CS}$ pin = 1 and DMA is inactive. • STANDARD I/O PORT, With DMA inactive and $\overline{CS}$ pin low: Command, Parameter, Count and Status register data is transferred. • SLAVE MODE, with external DMA controller active and LACK pin high: D0–7 are transferred between FIFO and memory. • MASTER MODE, with internal DMA active, and LACK pin high: A16–23, A0–7 and D0–7 are transferred depending on DMA mode and bus phase.
LRQ	36	51	O	**LOCAL DMA REQUEST:** Requests are automatically generated when the FIFO needs to have data transferred.
AD8–15	1–8	1–7 12	I/O	**ADDRESS/DATA 8–15:** • STANDARD I/O PORT, with DMA inactive and $\overline{CS}$ pin low: These pins are driven high. • SLAVE MODE, with external DMA active and LACK pin high: D8–15 are transferred between FIFO and memory. • MASTER MODE, with internal DMA active and LACK pin high: A24–31, A8–15 and D8–15 are transferred, depending on DMA mode and bus phase.
ADS0	9	13	I/O	**ADDRESS STROBE 0:** • INPUT with DMA inactive: ADS0 latches RS0–5 inputs when low. When high, data present on RS0–5 will flow through to internal register decoder. • OUTPUT: ADS0 latches low order address bits (A0–15) to external memory during DMA transfers.
ADS1/RRQ	37	53	O	**ADDRESS STROBE 1/REMOTE REQUEST:** In 32-bit DMA Mode, ADS1 latches high order address bits (A16–31) to external memory. For remote DMA modes, RRQ pin is active high when SRI or SRO bits in the OC register are set in non-tracking mode, or during a remote transfer in tracking mode. (See RT register description in DMA REGISTERS Section.)

2.2 DISK INTERFACE PINS

Symbol	DIP Pin No.	PCC Pin No.	Type	Function
RCLK	25	35	I	**READ CLOCK:** Disk data rate clock. When RGATE is high, RCLK input will be the recovered/separated clock from the recorded data and is used to strobe data into the DDC. When RGATE is low, this input should become the referenced clock which will be delayed and is used as WCLK to strobe data to the drive. The transition between the recovered/separated clock and reference clock must be made with no short pulses. Short pulses are pulses that are less than the specified minimum RCLK pulse widths which are specified in the AC timing section as rcl and rch. In the event of any short pulses on RCLK or if RCLK is inactive for greater than 10 μs, then the DDC could go into an indeterminant state. If this happens, then the DDC needs to be reset and the format parameters must be updated to ensure normal operation. Maximum ratio of RCLK/BCLK is 4 for word mode, and 2 for byte mode.

2.0 Pin Descriptions (Continued)

2.2 DISK INTERFACE PINS (Continued)

Symbol	DIP Pin No.	PCC Pin No.	Type	Function
RGATE	19	29	O	**READ GATE:** Set active high during any disk read operation. This pin commands data separator to acquire lock. Enables RDATA input pin.
RDATA	15	22	I	**READ DATA:** Accepts NRZ disk data from the data separator/decoder.
WCLK	21	31	O	**WRITE CLOCK:** Used when NRZ data is on WDATA pin. Also active when MFM data is used, but normally not utilized. WCLK frequency follows RCLK pin.
WGATE	20	30	O	**WRITE GATE:** When writing data onto a disk, WGATE is asserted high with the first bit of data and deasserted low after the last bit of data. WGATE is also de-asserted on reset or on detection of an error.
WDATA	18	28	O	**WRITE DATA:** During any write operation, MFM or NRZ encoded data is output to disk, dependent upon MFM bit status in the DF register. This pin is inactive low when WGATE is low.
AMF/EPRE	16	23	I/O	**ADDRESS MARK FOUND/EARLY PRECOMPENSATION:** Address mark input is monitored if the HSS bit in the DF register is low (for soft sectoring). If the MFM bit in the DF register and the EP bit in the OC register are both set, then this pin becomes the EPRE control. If both functions are used, WGATE pin determines the function as follows: • WGATE asserted: EPRE output. • WGATE de-asserted: AMF input.
AME/LPRE	13	19	O	**ADDRESS MARK ENABLE/LATE PRECOMPENSATION:** If the MFM bit in the DF register is low, AME will indicate that an address mark byte(s) is being output on WDATA pin. If the MFM bit in the DF register and the EP bit in the OC register are both set, LPRE control is output (if internal MFM encoding is used).
SECTOR	22	32	I	**SECTOR PULSE:** In hard sectored drives, this signal comes from the start of a sector. In a soft sectored drive this pin must be tied low.
INDEX	23	33	I	**INDEX PULSE:** This signal comes from the disk drive, indicating the start of a track.
SDV	27	37	O	**SERIAL DATA VALID:** Asserted when the DDC is either issuing or receiving header field, internal header CRC/ECC, data field, or internal data CRC/ECC information. Mainly used for external ECC and diagnostics.
EEF	26	36	O	**EXTERNAL ECC FIELD:** Only used if the External ECC Byte Count register(s) are non-zero. Asserted when external ECC check bits are being generated (WGATE high) and checked (RGATE high).
EXT STAT	17	24	I	**EXTERNAL STATUS:** *IMPORTANT NOTE: This pin MUST be tied low if it is not to be used.* This pin has three functions: • 1: If EEW bit in the RT register is set, the read and write strobes are extended for both remote and local transfers as long as this pin is high. This is the External Wait State function. • 2: If the EEW bit in the RT register is low, this pin will accept a pulse granting valid byte alignment on the last bit of the synch byte before header or data bytes. This is an OR function with the internal synch detect. • 3: External ECC Check. Only used if External ECC Byte Count register(s) are non-zero, and EEW bit in the RT register is low. After the last byte of external ECC, this pin will accept a pulse confirming that there has been no error. A CRC/ECC error will be flagged if this pulse is not received.
V_{CC} GND	14, 12	20, 21 16, 17		**POWER, GROUND:** +5V DC is required. It is suggested that a decoupling capacitor be connected between these pins. It is essential to provide a path to ground for the GND pin with the lowest possible impedance. Otherwise any voltage spikes resulting from transient switching currents will be reflected in the logic levels of the output pins.

3.0 Internal Registers of the DDC

The numerous registers within the DDC are presented below, grouped according to their function. A key is given as an aid for the use of each register. The key data is only suggested for common operation, and should not be considered as an absolute requirement. Following this listing is a description of each register, in the order of which they are listed below. The HA column at the left of this listing gives the Hex Address of each register.

KEY

- D May be updated when a different drive type is selected
- C May be updated before each command
- R May be read at any idle time
- F Used during formatting
- I Used during initialization
- NO Operation is not possible

COMMAND

HA	Register	Bits	Write	Read
10	Drive Command Register (DC)	8	C	NO
11	Operation Command Register (OC)	8	C	NO
35	Disk Format Register (DF)	8	D	NO
00	Status Register (S)	8	NO	R
01	Error Register (E)	8	NO	R
12	Sector Counter (SC)	8	C	R
13	Number of Sector Operations Counter (NSO)	8	C	R
0F	Header Byte Count (HBC)/Interlock	3	F	R
36	Header Diagnostic Readback (HDR)	8	NO	R

DMA

HA	Register	Bits	Write	Read
37	DMA Sector Counter (DSC)	8	NO	R
37	Remote Transfer Register (RT)	8	I	NO
36	Local Transfer Register (LT)	8	I	NO
1A	Remote Data Byte Count (L)	8	C	R
1B	Remote Data Byte Count (H)	8	C	R
1C	DMA Address Byte 0	8	C	R
1D	DMA Address Byte 1	8	C	R
1E	DMA Address Byte 2	8	C	R
1F	DMA Address Byte 3	8	C	R

FORMAT (See Note)

HA	Register	Bits	Write	Read
21	ID Preamble Byte Count	5	D	R
31	ID Preamble Pattern	8	D	R
22	ID Synch #1 (AM) Byte Count	5	D	R
32	ID Synch #1 (AM) Pattern	8	D	R
23	ID Synch #2 Byte Count	5	D	R
33	ID Synch #2 Pattern	8	D	R
24	Header Byte 0 Control Register (HC0)	5	D	R
14	Header Byte 0 Pattern	8	D	R
25	Header Byte 1 Control Register (HC1)	5	D	R
15	Header Byte 1 Pattern	8	D	R

FORMAT (Continued)

HA	Register	Bits	Write	Read
26	Header Byte 2 Control Register (HC2)	5	D	R
16	Header Byte 2 Pattern	8	D	R
27	Header Byte 3 Control Register (HC3)	5	D	R
17	Header Byte 3 Pattern	8	D	R
28	Header Byte 4 Control Register (HC4)	5	D	R
18	Header Byte 4 Pattern	8	D	R
29	Header Byte 5 Control Register (HC5)	5	D	R
19	Header Byte 5 Pattern	8	D	R
2B	ID External ECC Byte Count	5	D	R
2C	ID Postamble Byte Count	5	D	R
3C	ID Postamble Pattern	8	D	R
2D	Data Preamble Byte Count	5	D	R
3D	Data Preamble Pattern	8	D	R
2E	Data Synch #1 (AM) Byte Count	5	D	R
3E	Data Synch #1 (AM) Pattern	8	D	R
2F	Data Synch #2 Byte Count	5	D	R
3F	Data Synch #2 Pattern	8	D	R
3B	Data Format Pattern	8	F	R
38	Sector Byte Count L	8	D	R
39	Sector Byte Count H	8	D	R
2A	Data External ECC Byte Count	5	D	R
20	Data Postamble Byte Count	5	D	R
30	Data Postamble Pattern	8	D	R
34	Gap Byte Count	8	F	R
3A	Gap Pattern	8	F	R

CRC/ECC

HA	Register	Bits	Write	Read
02	ECC SR Out 0	8	NO	R
03	ECC SR Out 1	8	NO	R
04	ECC SR Out 2	8	NO	R
05	ECC SR Out 3	8	NO	R
06	ECC SR Out 4	8	NO	R
07	ECC SR Out 5	8	NO	R
02	Polynomial Preset Byte 0 (PPB0)	8	D	NO
03	Polynomial Preset Byte 1 (PPB1)	8	D	NO
04	Polynomial Preset Byte 2 (PPB2)	8	D	NO
05	Polynomial Preset Byte 3 (PPB3)	8	D	NO
06	Polynomial Preset Byte 4 (PPB4)	8	D	NO
07	Polynomial Preset Byte 5 (PPB5)	8	D	NO
08	Polynomial Tap Byte 0 (PTB0)	8	D	NO
09	Polynomial Tap Byte 1 (PTB1)	8	D	NO
0A	Polynomial Tap Byte 2 (PTB2)	8	D	NO
0B	Polynomial Tap Byte 3 (PTB3)	8	D	NO
0C	Polynomial Tap Byte 4 (PTB4)	8	D	NO
0D	Polynomial Tap Byte 5 (PTB5)	8	D	NO
0E	ECC/CRC Control (EC)	8	D	NO
08	Data Byte Count L	8	NO	R
09	Data Byte Count H	8	NO	R

3.0 Internal Registers of the DDC (Continued)

DUAL-PURPOSE REGISTERS

Some of the above listed registers have dual functions depending on whether they are being written to or read from. These registers are repeated below to help clarify their operation.

HA	Register	Bits	Write	Read
02	ECC SR Out 0	8	NO	R
02	Polynomial Preset Byte 0 (PPB0)	8	D	NO
03	ECC SR Out 1	8	NO	R
03	Polynomial Preset Byte 1 (PPB1)	8	D	NO
04	ECC SR Out 2	8	NO	R
04	Polynomial Preset Byte 2 (PPB2)	8	D	NO
05	ECC SR Out 3	8	NO	R
05	Polynomial Preset Byte 3 (PPB3)	8	D	NO
06	ECC SR Out 4	8	NO	R
06	Polynomial Preset Byte 4 (PPB4)	8	D	NO
07	ECC SR Out 5	8	NO	R
07	Polynomial Preset Byte 5 (PPB5)	8	D	NO
08	Polynomial Tap Byte 0 (PTB0)	8	D	NO
08	Data Byte Count (0)	8	NO	R
09	Polynomial Tap Byte 1 (PTB1)	8	D	NO
09	Data Byte Count (1)	8	NO	R
36	Header Diagnostic Readback (HDR)	8	NO	R
36	Local Transfer Register (LT)	8	I	NO
37	DMA Sector Counter (DSC)	8	NO	R
37	Remote Transfer Register (RT)	8	I	NO

Format Note: It is recommended that the Format Registers be reloaded after the following events:

1. A hardware or software reset of the chip
2. A Sector Not Found error
3. A Sector Overrun error
4. A Data Sync Error

3.1 COMMAND REGISTERS

DRIVE COMMAND (DC) Hex Address (10) Write Only

The locations within this register, when written to, initiate disk commands and chip functions. For a disk operation, after the DDC has been configured, this register is loaded to initiate command execution.

Loading the DC register constitutes the initiation of a disk operation and will hence generate an operation complete interrupt.

DO2	DO1	H02	H01	FMT	MSO	SAIS	RED
7	6	5	4	3	2	1	0

RED: Re-enable DDC

A 1 should be written into this location during the power up initialization process (see POWER UP AND INITIALIZATION Section), or after an error has been encountered in order to re-enable the DDC to accept commands. (NOTE: If the RES bit in the OC register has been set, a 0 should be written to that location before this operation is performed.) If no error has been encountered, and a command is being issued, a zero should be written to this bit. The Re-enable is an operation by itself and hence an interrupt will be generated on completion of the operation.

SAIS: Start at Index or Sector

0 Operation begins only upon receipt of an index pulse.

1 Operation begins on either an index pulse or sector pulse for hard sector drives or immediately for soft sector drives.

MSO: Multi-sector Operation

0 Single-sector operation.

1 Multi-sector operation using NSO register.

FMT: Format Mode

0 No Format Operation.

1 When set, along with other DC register bits, will initiate disk formatting upon receipt of an index pulse.

HO1, 2: Header Operation Bits:

H02 H01

0 0 *IGNORE HEADER:* associated data transfer operation will take place with any valid sector encountered.

0 1 *COMPARE HEADER:* Normal mode used to find a specific sector. The Header Pattern registers contain the comparison pattern.

1 0 *WRITE HEADER (Write ID):* Normally used only during Format mode to write ID patterns to disk.

1 1 *READ HEADER (Read ID):* Reads header information from disk for diagnostic purposes.

DO1, 2: Data Operation Bits:

D02 D01

0 0 *NO OPERATION:* Can be used only with an Ignore Header command. No disk operation is performed with this combination, and it can be used along with the RED command to re-enable the DDC (see OPERATING MODES).

0 1 *CHECK DATA:* No DMA action and no data movement between disk and FIFO. CRC/ECC checks are calculated and interrupts, if enabled, are asserted on proper conditions. DFE bit in Error register will be set if a data CRC/ECC error occurs unless in Interlock Mode.

1 0 *WRITE DATA:* Initiates local DMA action to fill the FIFO. Writes data to disk with the proper pre and post appendages in the data field. FIFO is replenished by local DMA.

1 1 *READ DATA:* Data enters FIFO from disk, and local DMA transfer is initiated when the FIFO contains the number of bytes specified by the Burst Length in the LT register.

The following table shows a list of valid commands combining the H01, H02, D01, D02, FMT bits from the DC register and the FTF bit in the DF register. No other DC register combinations are allowed.

3.0 Internal Registers of the DDC (Continued)

Valid DDC Commands

DC Register D02	D01	H02	H01	FMT	DF Reg FTF	Operation
0	0	0	0	0	X	No Operation
0	1	0	1	0	X	Check Data, Compare Header
0	1	1	0	0	X	Check Data, Write Header
0	1	1	1	0	X	Check Data, Read Header
1	0	0	0	0	X	Write Data, Ignore Header
1	0	0	1	0	X	Write Data, Compare Header, (normal write)
1	0	1	0	0	X	Write Data, Write Header
1	0	1	0	1	0	Write Data,Write Header, Format with No FIFO Table
1	0	1	0	1	1	Write Data, Write Header, FIFO Table Format
1	1	0	0	0	X	Read Data, Ignore Header, (recover data)
1	1	0	1	0	X	Read Data, Compare Header, (normal read)
1	1	1	1	0	X	Read Data, Read Header

OPERATION COMMAND (OC)

Hex Address (11) **Write Only**

The fields within this register enable on-chip operations. In non-tracking mode, a remote DMA operation will be initiated by loading the SRO or SRI bits in this register.

IR	SCC	EP	SRO	SRI	EHI	EI	RES
7	6	5	4	3	2	1	0

RES: Reset DDC

0 Clears a previously set RES function. Allows normal operation.

1 DDC immediately enters a stand-by mode. The FIFO is reset, Status and Error registers are cleared and all operations in progress are stopped. DDC is placed in the Reset mode (see OPERATING MODES). RGATE and WGATE pins are de-asserted if active. All DMA counters are cleared. Format Parameter, DMA Address and ECC registers are unaffected.

EI: Enable Interrupts

0 Disabled, INT pin remains inactive high.

1 Enables interrupts generated by the following:

- Correction cycle complete.
- Error which sets ED bit in Status register.
- Command successfully completed (including independent remote DMA transfer).

EHI: Enable Header Interrupt

EI bit must be set if this bit is set.

0 Disabled.

1 Interrupt issued at start of ID postamble field when:

- Header matches in Compare Header operation.
- Header finished in Read, Write or Ignore Header operation.

SRI, SRO: Start Remote Input, Start Remote Output

These bits are only operational in non-tracking mode. The Remote Start Address and Remote Data Byte Count registers must be loaded first.

SRI SRO

0 0 Remote DMA operation unchanged.

0 1 *START REMOTE OUTPUT*: Asserts RRQ pin and RCB flag in Status register, to begin a remote DMA operation from memory to I/O Port.

1 0 *START REMOTE INPUT*: Asserts RRQ pin and RCB flag in Status register, to begin a remote DMA operation from I/O Port to local memory.

1 1 *STOP CURRENT REMOTE OPERATION*: RRQ pin is de-asserted and RCB flag is reset in Status register.

EP: Enable Precompensation

0 Early and late precompensation signals are forced low during a disk write operation.

1 Permits precompensation signals to be output to external precompensation circuitry (see MFM ENCODED DATA). This bit is only valid if the MFM bit is set in the DF register.

SCC: Start Correction Cycle

0 No correction is attempted.

1 Setting this command will begin the internal correction cycle. The CCA flag in the Status register is set and drive commands should not be issued during this time. At the completion of the cycle, an interrupt is issued.

IR: Interlock Required (Interlock Mode)

0 No interlock function.

1 The interlock (HBC) register must be written to after the header operation has completed and before the DDC encounters the data postamble field. This allows updating of header bytes during a Format operation or changing of drive commands during a multi-sector operation. Normally used with the header interrupt enabled.

3.0 Internal Registers of the DDC (Continued)

DISK FORMAT (DF) Hex Address (35) Write Only

ID2	1D1	IH2	1H1	FTF	HSS	SAM	MFM
7	6	5	4	3	2	1	0

MFM: MFM Encode

(See MFM Encoded Data section.)

0 NRZ data is output on the WDATA pin when WGATE is active.

1 MFM data is output on the WDATA pin when WGATE is active. Also configures AMF/EPRE and AME/LPRE pins as EPRE and LPRE outputs when Write Gate is active. Precompensated outputs are enabled by the EP bit in the OC register.

SAM: Start with Address Mark

(See Formatting section)

0 Address Marks will be generated in the synch #1 fields if MFM bit = 1, or AME will be generated if MFM bit = 0.

1 Address Mark Enable will be generated in ID preamble if MFM bit = 0.

HSS: Hard or Soft Sectored

(See Hard Sector vs. Soft Sector Operation).

0 Sets DDC for soft sectored operation.

1 Sets DDC for hard sectored operation.

FTF: FIFO Table Format

0 Formatting is done without the use of DMA.

1 The local DMA channel loads the correct number of header bytes (HBC register) per sector into the FIFO from local memory. This data is then substituted for the header bytes during a format operation.

IH1, 2: Internal Header Appendage

IH2 IH1

0 0 No CRC/ECC is internally appended, but external ECC must be attached.

0 1 16-bit CRC CCITT polynomial is appended.

1 0 32-bit programmable ECC code is appended.

1 1 48-bit programmable ECC code is appended.

External ECC may be used with any internal CRC/ECC selection. 1 to 31 bytes of external ECC may be added.

ID1, 2: Internal Data Appendage

ID2, ID1

0 0 No CRC/ECC internally appended.

0 1 16-bit CRC CCITT polynomial is appended.

1 0 32-bit programmable ECC code is appended.

1 1 48-bit programmable ECC code is appended.

External ECC can be appended to any of the four cases dependent upon the Data External ECC Byte Count register.

STATUS (S) Hex Address (00) Read Only

The RESET pin and the RES bit in the OC register reset all of the bits in this register.

ED	CCA	LCB	RCB	LRQ	HMC	NDC	HF
7	6	5	4	3	2	1	0

HF: Header Fault

This bit is valid after a Compare Header or Read Header operation.

SET CRC/ECC error detected in a header field.

RESET This bit is reset when the DDC begins the next disk operation after a new disk command has been issued.

All ID fields entering the DDC during the operation are checked. The HF bit will be set if an error is detected in any header field encountered. However, if the header being sought is found and has no CRC/ECC error, the HF bit is reset. This bit does not produce an error that will stop operation, assert an interrupt, or set the ED bit in the Status register in a compare header operation, but will in a read header operation.

This bit could provide useful diagnostic information if a Sector Not Found error occurs (see Error Register in this section).

NDC: Next Disk Command

SET DDC will accept a new command into the DC register. The header operation is completing the last sector being operated on.

RESET On receipt of a new disk command.

HMC: Header Match Completed

For each of the following, this bit is set and the interrupt is generated at the start of the header postamble field.

Compare Header Operation:

SET Header field correctly matched with no CRC/ECC error.

RESET At beginning of subsequent header operation.

Read Header Operation:

SET Header field has been read with no CRC/ECC error.

RESET At beginning of subsequent header operation.

Ignore Header or Write Header Operation:

SET Always set at end of header field.

RESET At beginning of subsequent header operation.

LRQ: Local Request

This bit follows the LRQ pin, and allows application of the DDC in a polled mode.

SET LRQ pin is asserted.

RESET LRQ pin is not asserted.

RCB: Remote Command Busy

Non-Tracking Mode:

SET When OC register is loaded with a DMA instruction.

RESET Upon completion of the instruction or upon internal or external reset.

3.0 Internal Registers of the DDC (Continued)

Tracking Mode:

SET When RRQ pin is first asserted in a disk write mode, or when the Drive Command register is loaded in a disk read mode.

RESET Upon completion of the instruction or upon internal or external reset.

LCB: Local Command Busy

SET When command requiring local DMA is loaded.

RESET Upon completion of the last local or remote DMA transfer (in tracking mode) or upon internal or external reset.

CCA: Correction Cycle Active

SET On asserting SCC bit in the OC register.

RESET At the end of the correction cycle, simultaneously with the INT pin, if enabled.

ED: Error Detected

SET On assertion of one or more bits in the Error register.

RESET Upon internal or external reset.

ERROR(E) Hex Address (01) Read Only

Any bit set in this register generates an interrupt (if EI bit in the OC register is set) and stops the current operation. The RESET pin and the RES bit in the OC register reset all of the bits in this register.

LI	CF	FDL	NDS	SO	SNF	DFE	HFASM
7	6	5	4	3	2	1	0

HFASM: Header Failed Although Sector Number Matched

(See HFASM description in ADDITIONAL FEATURES)

SET The header bytes(s) marked with the EHF bit in the corresponding HC register(s) matched correctly, but other header bytes were in error.

RESET Upon internal or external reset.

DFE: Data Field Error

SET On detection of a data field CRC/ECC error in a Read Data or Check Data operation. This bit may be set when another error occurs; especially an error occurring during a Write operation. These errors would be Sector Overrun or FIFO Data Lost.

RESET Upon internal or external reset.

The RED command must be loaded into the DC register if error correction is to be attempted.

SNF: Sector Not Found

SET When header cannot be matched for two consecutive index pulses in any Compare Header operation.

RESET Upon internal or external reset.

SO: Sector Overrun

SET If RGATE is active and FIFO is being written to when a sector or index pulse is received. If WGATE is active, this bit is set when a sector or index pulse is received.

RESET Upon internal or external reset.

An SO error will not occur during a Format operation.

NDS: No Data Synch

SET If a sector or index pulse occurs while the DDC is waiting to byte align on the first data synch field (synch #1 or synch #2), or if the DDC byte aligns to the first synch word of the data field but does not match to subsequent bytes (synch #1 or synch #2).

RESET Upon internal or external reset.

FDL: FIFO Data Lost

SET During a disk read operation if the FIFO overflows, or during a disk write operation if the FIFO is read when it is empty.

RESET Upon internal or external reset.

CF: Correction Failed

SET If correction is attempted (SCC bit set in OC register) and correction failed.

RESET Upon internal or external reset.

LI: Late Interlock

Will only occur if IR bit in OC register is set.

SET Controlling logic has failed to write to the Interlock (HBC) register before the end of the data field of the present sector.

RESET Upon internal or external reset.

SECTOR COUNTER (SC)

Allowable Value 0–255 **Hex Address (12) Read/Write**

In a multi-sector operation, the SC register is first loaded with the starting sector number. It is incremented after each header operation is completed. The contents of the SC register will replce any header Byte if the SSC bit is set in the corresponding HC register.

NUMBER OF SECTOR OPERATIONS COUNTER (NSO)

Allowable Value 0–255 **Hex Address (13) Read/Write**

In a multisector operation, the NSO register is loaded with the number of sectors to be operated on. It is decremented after every header operation. When zero, the command is finished. This counter must be reloaded after a reset of the DDC.

HEADER BYTE COUNT (HBC)/INTERLOCK

Allowable Value 2–6 **Hex Address (0F) Read/Write**

This register loads the DMA with the number of header bytes to expect in a Read Header, or a Format operation where FIFO table formatting is used. This register is also used in interlock mode to signal completion of update. The upper five bits of this register are pulled low when read.

HEADER DIAGNOSTIC READBACK (HDR)

Hex Address (36) Read Only

If a Compare Header/Check Data operation is performed and an HFASM error occurs, the header bytes for that sector will have been loaded into the FIFO. By consecutively reading this address, the header bytes are read from the FIFO to the microprocessor. Data will be valid for only the number of header bytes specified in the parameter RAM. (NOTE: This is a dual function register, sharing operation with the Local Transfer register, see DMA REGISTER.)

SECTOR BYTE COUNT REGISTER (L, H)

Allowable Value 1–64k **Hex Address (38, 39) Read/Write**

The two bytes (most and least significant) that comprise this register are loaded during initialization, and define the data

3.0 Internal Registers of the DDC (Continued)

field size for each sector. The number of bytes transferred with local DMA is always equal to what has been loaded into this register. Loading *both* with zero is not allowed.

3.2 DMA REGISTERS

LOCAL TRANSFER (LT) Hex Address (36) Write Only

This is a dual function register, sharing operation with the Header Diagnostic Readback (HDR) register (see COMMAND REGISTERS). IMPORTANT NOTE: If any internal DMA is being used, or if the Remote Data Byte Count registers will be read by the processor, the LT (and RT) register must be loaded before the Sector Byte Count and Remote Data Byte Count register pairs.

LBL2	LBL1	LTEB	LA	LSRW	RBO	LWDT	SLD
7	6	5	4	3	2	1	0

SLD: Select Local DMA Mode

0 *SLAVE MODE*: External DMA must be used in place of on-chip DMA.

1 *NON-TRACKING MODE*: Local DMA is enabled. Whenever local transfers are needed, the DDC becomes the bus master.

TRACKING MODE: Local and remote DMA are enabled. DMA transfers are interleaved (see DMA in DATA TRANSFER section).

LWDT: Local Word Data Transfer

0 Address increments by 1, 8 bit wide transfers.

1 Address increments by 2, 16 bit wide transfers. Address, A0, remains unchanged as it was set by the DMA address.

RBO: Reverse Byte Order

Valid if LWDT bit is set.

0 First byte to/from FIFO is mapped onto the AD0–7 bus.

1 First byte to/from FIFO is mapped onto AD8–15 bus (e.g. 68000).

LSRW: Local Slow Read And Write

0 DMA cycles are four clock periods.

1 DMA cycles are five clock periods. RD and WR strobes are widened by one clock period.

LA: Long Address

Valid only if SLD = 1, and SRD = 0 in Remote Transfer register.

0 16 address bits are issued and strobed by the ADS0 pin. ADS1/RRQ is available for use by the remote DMA.

1 32 address bits are issued, the lower 16 are strobed by ADS0 pin. The most significant 16 address lines are only issued when a rollover from the least significant 16 address lines occurs, or after loading the upper half of the 32-bit address. When the upper 16 address lines are issued, that DMA cycle is five clock cycles long if no internal or external wait states are used.

LTEB: Local Transfer Exact Burst

0 When DMA tranfer is needed, the FIFO will be filled when writing to disk or emptied when reading from disk.

1 When DMA tranfer is needed, the FIFO will receive (when writing) or deliver (when reading) an exact burst of data.

LBL1, 2: Local Burst Length

LBL2	LBL1	
0	0	1 word (2 byte)
0	1	4 word (8 byte)
1	0	8 word (16 byte)
1	1	12 word (24 byte)

When reading from disk, these bits select the number of bytes needed in the FIFO in order to generate an LRQ signal. When writing, these bits select the number of bytes that need to be removed from a full FIFO in order to generate an LRQ. In either case, if the LTEB bit is set, this bit pair indicate how many data transfers will be allowed before LRQ is removed.

Note: Please refer to Section 17, Helpful Hints #29.

REMOTE TRANSFER (RT) Hex Address (37) Write Only

This is a dual function register, sharing operation with the DMA Sector Counter (DSC) (see DSC at the end of this section). If any internal DMA is being used, or if Remote Data Byte Count registers will be read by the processor, the RT (and LT) register must be loaded before the Sector Byte Count and Remote Data Byte Count register pairs.

RBL2	RBL1	RTEB	TM	RSRW	EEW	RWDT	SRD
7	6	5	4	3	2	1	0

SRD: Select Remote DMA

0 Remote DMA inhibited, ADS1/RRQ pin is configured as ADS1.

1 Remote DMA enabled. This is necessary but not sufficient to start remote transfer.

RWDT: Remote Word Data Transfer

0 Remote address increments by 1.

1 Remote address increments by 2. Address A0 remains unchanged as it was set by the starting DMA address.

EEW: Enable External Wait

0 No external wait states acknowledged. Functions 2 and 3 of EXT STAT pin are enabled (see PIN DESCRIPTIONS).

1 The EXT STAT pin will lengthen RD and WR strobes during DMA transfers as long as it is maintained at a high level.

RSRW: Remote Slow READ/WRITE

0 Remote DMA cycles are four clock periods long.

1 Remote DMA cycles are five clock periods long, if external wait states are not asserted.

TM: Tracking Mode

See Tracking Mode description in DATA TRANSFER Section.

0 DMA channels are independent and addresses are allowed to overlap.

1 DMA channel addresses are not allowed to overlap.

RTEB: Remote Transfer Exact Burst

0 If a remote transfer has been initiated, the RRQ pin will remain asserted until the number of bytes specified by the Remote Data Byte Count registers has been transferred, or until the oper-

3.0 Internal Registers of the DDC (Continued)

ation is reset or SRI and SRO bits in the OC register are both set when in non-tracking mode, or when DMA sector counter reaches zero when in tracking mode.

1 If a remote transfer has been initiated, the RRQ pin will remain asserted until the exact number of bytes specified by RBL1 and RBL2 has been transferred, or if any of the conditions described in the previous paragraph occur.

RBL1, 2: Remote Burst Length

LBL2	LBL1	
0	0	1 word (2 byte)
0	1	4 word (8 byte)
1	0	8 word (16 byte)
1	1	12 word (24 byte)

REMOTE DATA BYTE COUNT (L, H)

Allowable Value 0–64k Hex Address (1A, 1B) READ/WRITE

This pair of registers specifies the number of bytes in one remote transfer using the 16-bit address of the remote DMA channel. In the non-tracking mode, the remote DMA can transfer 1–64k bytes independent of the local DMA. Loading both registers with zero will be interpreted as a 64k byte count. These registers are ignored in tracking mode.

DMA ADDRESS BYTE 0–3

Allowable Value 0–255 Hex Address (1C–1F) READ/WRITE

These address bytes are configured dependent on the current DMA mode. In *32-bit mode*, all four bytes form the physical address with 1F containing the most significant byte. In *16-bit mode*, bytes 0 and 1 form the low and high bytes of the local DMA channel, and bytes 2 and 3 form the low and high of the remote DMA channel, if enabled.

DMA SECTOR COUNTER (DSC)

Hex Address (37) Read Only

This counter is only valid during tracking mode and holds the difference between the number of sectors transferred by the local and remote DMA channels. In tracking mode, when DSC = 0, remote transfer is disabled in a disk read operation so invalid data is not exchanged between local and host memory. This is a dual function register, sharing operation with the Remote Transfer (RT) register described earlier in this section.

3.3 FORMAT REGISTERS

The disk format is defined by using the format pattern and control registers. Generally, these registers are set up in pairs. In each pair, one register is loaded with an appropriate 8-bit pattern that will be written to the disk during a Format or Write command, or will be used during a Read or Compare command for byte alignment or a comparison in locating a sector. Refer to *Figure 4*, below, for a listing of the format registers, and the manner in which they are paired. The FORMAT, READ AND WRITE Section contains a listing and description of each of the format fields.

The other register in the pair is used to control the use of the corresponding pattern register. These Byte Count registers are loaded with a 5-bit binary number indicating the number of times the associated pattern will be repeated, therefore defining the size of that particular field (0–31 bytes). The Gap Byte Count register is the only one with 8 bits, allowing a field of up to 255 bytes in length.

The External ECC Count registers do not perform any pattern repetition. The external ECC appendage is provided from outside the DDC, and must be fit into the field whose length is defined by these registers (0–31 bytes). If any field is to be excluded from the disk format, the Byte Count register associated with that field must be loaded with zero. This is particularly important with the External ECC Byte Count registers. If these are non-zero, the EXT STAT pin will expect a pulse for each external ECC field during a Read operation. If these pulses are not supplied, the operation will be aborted in an error condition. Also, no more than two consecutive format fields may be deleted at one time.

The Header Byte Control registers also do not perform any pattern repetition, nor do they define field size. They are provided for controlling the function of each corresponding header byte.

HEADER CONTROL (HC0–5)

Hex Address (24–29) Read/Write

There is one HC register for each of six Header Byte pattern registers.

NU	NCP	EHF	SSC	HBA
4	3	2	1	0

HBA: Header Byte Active

0 The corresponding Header Byte is not included in the header byte field and will not be used in the ID operation. All other bits in each HC register in which this bit is set to zero must also be set to zero. A minimum of two Header Bytes must be enabled out of six, with no more than two disabled consecutively.

1 The corresponding Header Byte contains valid data and will be used in the ID operation.

SSC: Substitute Sector Counter

0 The corresponding Header Byte as stored in the pattern register is directly written to the disk for a Write Header command, and will be compared for Compare Header command.

1 The contents of the Sector Counter (SC) are substituted for this Header Byte during a Write Header command and compared during a Compare Header command. This is normally used in multisector operations.

EHF: Enable HFASM Function

See HFASM function description in ADDITIONAL FEATURES.

0 HFASM function is disabled.

1 HFASM function is enabled. The corresponding Header Byte is designated as that byte that must match in order to generate an HFASM error, typically the sector number.

NCP: Not Compare

0 The corresponding Header Byte will be compared normally.

1 A valid comparison will always be assumed, regardless of the true outcome.

NU: Not Used

This bit must be set to zero. If set to 1 unspecified operations may occur.

3.0 Internal Registers of the DDC (Continued)

Pattern Register	Hex Addr	Pattern Source	Control Function	Hex Addr	Control Register
ID Preamble	31	*Internal*	*Repeat 0–31x*	21	ID Preamble Byte Count
ID Synch #1 (AM)	32			22	ID Synch #1 (AM) Byte Count
ID Synch #2	33			23	ID Synch #2 Byte Count
Header Byte 0	14		*Define/Control*	24	Header Byte 0 Control
Header Byte 1	15			25	Header Byte 1 Control
Header Byte 2	16			26	Header Byte 2 Control
Header Byte 3	17			27	Header Byte 3 Control
Header Byte 4	18			28	Header Byte 4 Control
Header Byte 5	19			29	Header Byte 5 Control
ID External ECC	*	*External*	*0–31 Bytes*	2B	ID External ECC Byte Count
ID Postamble	3C	*Internal*	*Repeat 0–31x*	2C	ID Postamble Byte Count
Data Preamble	3D			2D	Data Preamble Byte Count
Data Synch #1 (AM)	3E			2E	Data Synch #1 (AM) Byte Count
Data Synch #2	3F			2F	Data Synch #2 Byte Count
Data Format	3B		*Field Size 1–64k Bytes*	38	Sector Byte Count L
				39	Sector Byte Count H
Data External ECC	*	*External*	*0–31 Bytes*	2A	Data External ECC Byte Count
Data Postamble	30	*Internal*	*Repeat 0–31x*	20	Data Postamble Byte Count
Gap	3A		*Repeat 0–255x*	34	Gap Byte Count

*These are not pattern registers.

FIGURE 4. Format Registers

3.4 CRC/ECC REGISTERS

The following registers are for programming and controlling the CRC/ECC functions of the DDC. Many of these registers have dual functions, depending on whether they are being written to or read from. Take care in noting which these are, to avoid confusion later. Only a basic functional description of these are provided here. Detailed instructions on their use can be found in the CRC/ECC section.

ECC SR OUT 0–5 Hex Address (02–07) Read Only

The syndrome bytes for performing a correction are available from these registers, and are externally XOR'ed with the errored data bytes. These are dual function registers, sharing operation with the Polynomial Preset Bytes.

POLYNOMIAL PRESET BYTES 0–5 (PPB0–5)

Hex Address (02–07) Write Only

The ECC shift registers can be preset by loading a bit pattern into these registers. These are dual function registers, sharing operation with the ECC SR Out registers.

POLYNOMIAL TAP BYTES (PTB0–5)

Hex Address (08–0D) Write Only

These registers are used for programming the taps for the internal 32 or 48-bit ECC polynomial. PTB0 and PTB1 are dual function registers, sharing operation with the Data Byte Counters.

DATA BYTE COUNTER 0, 1 (LS, MS)

Hex Address (08, 09) Read Only

The Data Byte Counters indicate the location of the byte in error after an ECC cycle. These are dual function registers, sharing operation with the Polynomial Tap Bytes 0 & 1. The Sector Byte Count Register must be reloaded with the sector length plus the number of ECC bytes before the start of a correction cycle. If the CF bit in the Error register is reset after a correction, the Data Byte Counter will contain an offset pointing to the first byte in error.

ECC/CRC Control (EC)
Hex Address (OE) Write Only

DNE	IDI	IEO	HNE	CS3	CS2	CS1	CS0
7	6	5	4	3	2	1	0

CS0–CS3: Correction Span Selection Bits

These four bits program the number of bits that the ECC circuit will attempt to correct. Errors longer than the correction span will be treated as non-correctable. The allowable correction span is 3–15 bits. If a span outside this range is loaded, the DDC will automatically default to a span of three bits.

For example, a five bit correction span would load as:

CS3	CS2	CS1	CS0
0	1	0	1

HNE: Header Non-Encapsulation

0 Header address mark and/or synch fields are encapsulated in the CRC/ECC calculation.

1 Header address mark and/or synch fields are not encapsulated in the CRC/ECC calculation.

NOTE: The SAM bit in the DF register must be reset when performing a Compare or Read Header operation, and the HNE bit is active low. If this is not done, the CRC/ECC calculation will begin at the synch word of the header, resulting in a Header Fault that will abort a Read operation or a Sector Not Found error for a Compare Header operation.

IEO: Invert ECC Out

See note under IDI bit, below.

0 Checkbits exiting ECC/CRC shift register are unaltered.

1 Checkbits exiting ECC/CRC shift register are inverted.

3.0 Internal Registers of the DDC (Continued)

IDI: Invert Data In

0 Data and checkbits entering the ECC/CRC shift register are unaltered.

1 Data and checkbits entering the ECC/CRC shift register are inverted.

NOTE: This inversion option has been included for compatability with a few systems that require ECC input and/or output inversion.

DNE: Data Non-Encapsulation

0 Data address mark and/or synch fields are encapsulated in the CRC/ECC calculation.

1 Data address mark and/or synch fields are not encapsulated in the CRC/ECC calculation.

4.0 DDC Operation

4.1 MICROPROCESSOR ACCESS

The DDC requires microprocessor control to initiate operations and commands, and to check chip status. All registers in the DDC appear as unique memory or I/O locations. Each can be randomly accessed and operated on. When the DMA is not performing a memory transfer, the chip can be accessed as a memory location or standard I/O port. Only eight bits of data may be transferred at this time, using pins AD0–7 (the upper 8 bits of a 16 bit microprocessor are not used). Six dedicated address pins (RS0–5) individually select all of the DDC's internal registers. By using these dedicated lines with an address strobe input (ADS0), the chip can be used in both multiplexed and demultiplexed address bus environments. The ADS0 and RS0–5 pins operate as a fall through type latch. By asserting CS active low, the DDC recognizes it has to be a slave and allows RD and WR to effect the internal registers. With multiplexed address and data lines, a positive strobe pulse on ADS0 will latch the address. The ADS0 line may be derived from a microprocessor address strobe such as ALE. In systems with a dedicated address bus (demultiplexed), ADS0 may be pulled high to allow address information to flow through the latch. Finally, by applying CS and a RD or WR strobe, any of the 64 internal locations can be accessed. It is important to note that most registers are read or write only. Some registers, however, change function dependent on whether they are being read from or written to (see Dual Function register list in INTERNAL REGISTERS).

4.2 OPERATING MODES

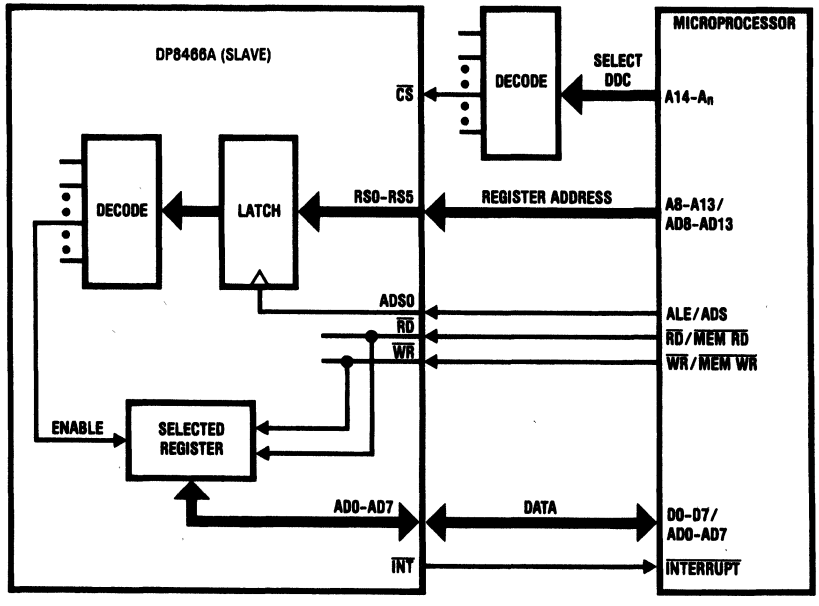
The DDC can be thought of as operating in four modes: *RESET, COMMAND ACCEPT, COMMAND PERFORM* and *ERROR*. These modes are given here in order to provide a functional operating description of the DDC, particularly when an error has been encountered.

Mode 1 *RESET*: All functions are stopped, and no command can be issued. During power up and before initalization, the DDC is held in this mode. To leave this mode, pin 24 (RESET) must be high, a 0 must be written to the RES location in the OC register, and a RED command loaded into the DC register. This places the DDC into MODE 2.

Mode 2 *COMMAND ACCEPT*: The DDC is free and ready to receive the next command (NDC bit set in Status register). Upon receipt of a command, the DDC will enter MODE 3.

Mode 3 *COMMAND PERFORM*: The directed operation is performed. If no error is encountered, the DDC will return to MODE 2. An error will put the DDC into MODE 4.

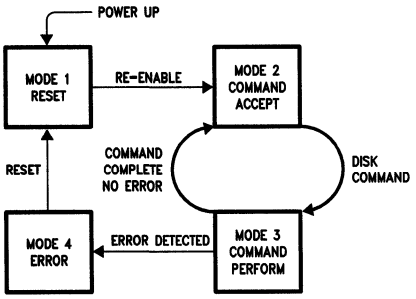
Mode 4 *ERROR*: The error needs to be serviced, and then the DDC can be reset by MODE 1.



TL/F/5282–4

FIGURE 5. Microprocessor Access to DP8466A

4.0 DDC Operation (Continued)

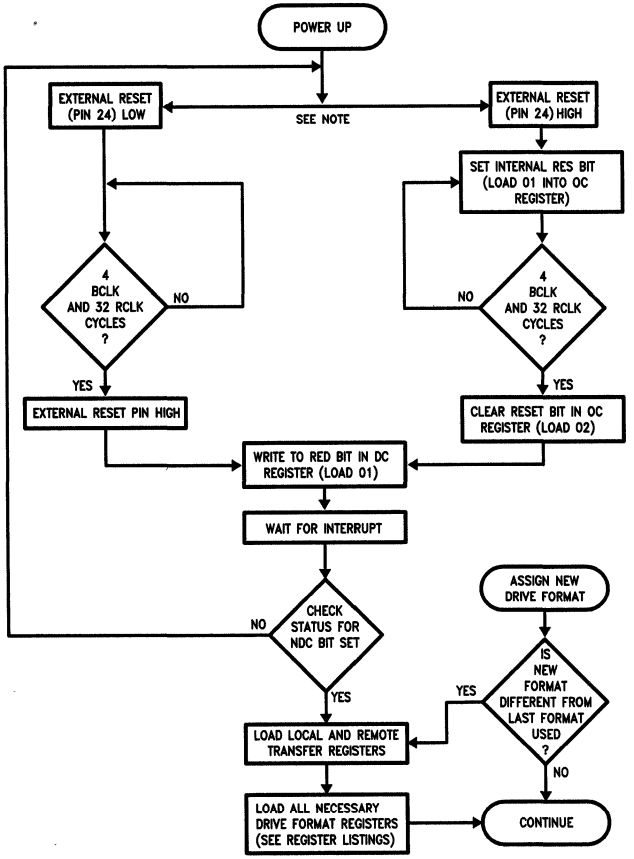


TL/F/5282–5

FIGURE 6. DDC Operating Modes

4.3 POWER UP AND INITIALIZATION

In powering up the DDC, the counters and registers must be initialized before a drive can be assigned and the appropriate information loaded. This can be done by either holding pin 24 (RESET) low, or by setting the internal RES bit in the OC register. Both require that the DDC be held in the reset condition for a minimum of 32 RCLK periods and 4 BCLK periods before the reset condition can be cleared. *Figure 7* shows a general algorithm for both methods. After power up, and whenever a new drive is assigned, the appropriate drive format registers need to be loaded before any drive operation is performed.



TL/F/5282–6

Note 1: If the RE-ENABLE operation is accomplished by polling the status register and not enabling interrupt, then it should be polled for NDC bit set. When set, it should remain set for at least 30 RCLKs before RE-ENABLE can be considered complete. The REN operation under worst case condition could take as long as 270 RCLKs.

Note 2: As shown various methods are possible for power up, and it is up to the user as to which is more suitable. The DDC should be reset and RCLK and BCLK should be applied after power up, otherwise it may draw an excessive amount of current, and may cause bus contention.

FIGURE 7. Power Up and Initialization Algorithm

5.0 Format, Read and Write

5.1 DISK FORMATTING

The formatting process is carried out through the format parameter and pattern registers (see FORMAT REGISTERS). These registers should be loaded during the initialization process for the particular drive in use. The pattern registers are loaded with the specific 8-bit pattern to be written to the disk. The count registers specify the number of times each 8-bit pattern is to be written. In loading these registers, several things need to be kept in mind:

- If any byte count register is loaded with zero, that field will be excluded, and no pattern for the corresponding pattern register need be loaded.
- At least two header bytes must be used, with no more than two consecutive unused header bytes. This also applies to all the fields in the format, where no more than two consecutive fields may be deleted. The one exception is the internal header ECC and external header ECC field. At least one of these fields must be present.
- If the disk is hard sectored, no gap byte count needs to be loaded. See Hard Vs. Soft Sector Operation in the FORMAT, READ AND WRITE Section.

The sector format options that are provided with the DDC are shown in *Figure 8.* The fields common to the ID and data fields, such as the preamble, Synch, CRC/ECC and postamble fields, perform similar functions, and are briefly discussed below.

PREAMBLE: Allows the PLL in the data separator to achieve phase lock.

SYNCH #1 and 2: Synch #1 contains the missing clock address mark for use with soft sectored disks. Generally, this field is not used in hard sectored disks. The synch #1 field can be used to extend the preamble or the synch fields in hard sectored mode. Synch #1 and #2 fields allow for byte alignment of the DDC.

HEADER BYTES: Used to uniquely identify each sector. Examples are sector number, cylinder number, track number, etc.

DATA: Information to be stored.

CRC/ECC: This field is generated and checked internally.

EXT.ECC: Used with external ECC circuitry. Provides space for externally generated ECC bytes.

POSTAMBLE: Allows read gate turn off time for the PLL to unlock. Provides a pad so that the write splice does not occur at the end of the CRC.

GAP 3: Provides protection against speed variation. In soft sectored mode, its length is determined by the Gap Byte Count register. In hard sectored mode, this gap will continue until the next sector pulse.

Format operations always start with an index pulse, and end with the next index pulse, thus making one track. The DDC has three approaches for formating disks:

Internal Sequential
FIFO Table
Interlock Type

INTERNAL SEQUENTIAL

This mode is used where the sector number is incremented for each physically adjacent sector, that is, for an interleave of one. This mode may be used on a multi-sector operation to format a whole track of sequential sectors. The header bytes other than the sector number, such as cylinder number and head number, are loaded. The Sector Counter (SC) is loaded with the first sector number desired on the track and the HC register with SSC = 1. The Number of Sector Operations (NSO) counter is loaded with the number of sectors per track. Finally, the FMT bit is set in the DC register in addition to bits for a Write Header/Write Data, multi-sector operation. Formatting begins upon loading the DC register. The last sector number written will therefore be [SC] + [NSO] − 1.

FIFO TABLE

This approach is ideal for sector interleaving and offers the minimum of microprocessor intervention during the format operation. The microprocessor sets up the header bytes of each sector, contiguously in memory. The local DMA channel or external DMA is used to transfer the header byte sets into the FIFO. Each set transferred is used once for each header field. The local DMA transfers a new set for each sector. The number of sectors transferred is determined by the NSO register.

The format operation follows the sequence below:

(1) Before the format operation, a full track of header byte sets is loaded into a memory area accessible to the local DMA channel. Each header byte set must contain an even number of bytes. If it contains an odd number of bytes, an extra "dummy" byte must be inserted so that each header byte set will be contained in an even byte boundary.

(2) The DMA address is loaded with the location of the first byte of the first header byte set.

ID FIELD

ID PREAMBLE	ID SYNCH #1 (AM)	ID SYNCH #2	HEADER BYTES	ID CRC/ECC*	ID EXT ECC*	ID POSTAMBLE
0–31 Bytes	*0–31 Bytes*	*0–31 Bytes*	*2–6 Bytes*	*0, 2, 4 or 6 Bytes*	*0–31 Bytes*	*0–31 Bytes*

DATA FIELD

DATA PREAMBLE	DATA SYNCH #1 (AM)	DATA SYNCH #2	DATA FORMAT PATTERN	DATA CRC/ECC	DATA EXT ECC	DATA POSTAMBLE	GAP 3
0–31 Bytes	*0–31 Bytes*	*0–31 Bytes*	*1–64k Bytes*	*0, 2, 4 or 6 Bytes*	*0–31 Bytes*	*0–31 Bytes*	*0–255 Bytes*

Note 1: The ID CRC/ECC field and the ID EXT ECC field must not be set to zero simultaneously.

Note 2: The ID and DATA preamble fields need to be at least 3 bytes for proper operation.

FIGURE 8. Sector Format Fields

5.0 Format, Read & Write (Continued)

(3) The Header Byte Count (HBC) is loaded with the number of header bytes in each sector (2–6 bytes).

(4) The Disk Format (DF) register is loaded with the FTF bit set.

(5) The Drive Command (DC) register is loaded for a Write Header/Write Data, multi-sector, format operation.

INTERLOCK TYPE

This approach offers the most versatility, but requires fast microprocessor intervention. It may be used to format a whole track of interleaved sectors. It can also be used for creating files of varying sector length, but this can be very tricky. The DDC can format sectors with data lengths from 1 to 64k bytes with single byte resolution.

Interlock type formatting uses the interlock mode and the header complete interrupt to enable the microprocessor to directly update any format parameter bytes. The Operation Command (OC) register is loaded wth IR (Interlock Mode), EHI and EI bits set. The Disk Format (DF) register should be loaded with the FTF bit reset. The header byte pattern for each selected header byte must be loaded into the relevant register. The NSO register is loaded with the number of sectors to be formatted. The DC register is then loaded for a Write Header/Write Data, multi-sector, format operation.

After the header field is written in the first sector, the DDC issues the header complete interrupt. With interlock mode set, the controlling microprocessor has the block of time until the preamble field of the next sector to read status, load the next sector's header bytes into the DDC registers and confirm this had been accomplished by writing to the Interlock (HBC) register. This must be done after the HMC interrupt for every sector, including the last sector of the operation. If this is not done, a Late Interlock error will occur when a subsequent command is loaded in the DC register.

In a non-format operation, the user has only until the end of the data field to write to the HBC register (see Data Recovery Using The Interlock Feature in ADDITIONAL FEATURES). This operation is repeated until the NSO register decrements to zero. An interrupt will then be issued indicating that the operation has completed.

5.2 READ AND WRITE

For initiating Read/Write operations, the necessary format registers need to be loaded with the appropriate information to enable the DDC to identify the desired sector. Multi-sector operations will also require the Number of Sector Operations (NSO) counter and the Sector Counter (SC). Algorithms outlining the read/write operations are shown in *Figures 10* and *11*. For each of these, it is assumed that the parameters for the desired sector(s) have been loaded, and that the head is positioned over the proper track.

READ

During a read operation, header data passing under the disk head is compared to the header bytes in the DDC parameter RAM. If a match is found after a read command is issued, the data field of the identified sector will start filling the FIFO. Once the selected threshold data level (burst length) is reached, the Local DMA Request (LRQ) pin will be asserted, signaling that a transfer is required. When the LACK pin grants the bus, either the exact burst length or the entire FIFO contents are transferred to memory. The FIFO continues filling, and this process repeats until the entire data field has been transferred to memory.

WRITE

A similar process occurs in reverse for a write operation. The DMA fills the FIFO, and when the correct sector is found, this data begins to be written to disk. When the data in the FIFO falls by an amount equal to the burst length, a transfer request is issued on LRQ. When LACK is granted, the DMA either fills the FIFO or transfers the exact number of bytes specified in the burst length. This process continues until a number of bytes specified by the Sector Byte Count register has been written to the disk.

Multi-sector operations follow the same procedure, but the operation is repeated on the number of sectors specified in the Number of Sector Operations (NSO) counter, with an interrupt being generated on completion of the last sector.

5.3 HARD SECTOR vs. SOFT SECTOR OPERATION

The choice between hard and soft sectored operation is made through the use of the HSS bit in the Drive Format register. This bit, in conjunction with other control bits can set the DDC to perform a number of functions depending on whether a read, write or format operation is to be enacted. HSS = 0 sets the DDC for soft sectored operation, and HSS = 1 sets the DDC for hard sectored operation.

FORMAT

In hard sectored operation, the DDC assumes that sector pulses are present, and will ignore the gap count. Gap bytes will be written until a pulse is detected on the SECTOR pin.

In soft sectored operation, the gap count will be used for every sector except the last. The Gap Byte Count register determines the Gap 3 length. For the last sector, gap bytes will be written until an index pulse is received.

READ

When reading, the need for the AMF input pulse is determined by the HSS bit. For soft sectoring, the AMF input is required for at least one bit time within the Synch #1 fields in both the ID and Data sections of the sector. For hard sectoring, the AMF input is not required.

The HSS bit in the DF register, and the SAIS command in the DC register define when RGATE is asserted for various sector formats. This is outlined below.

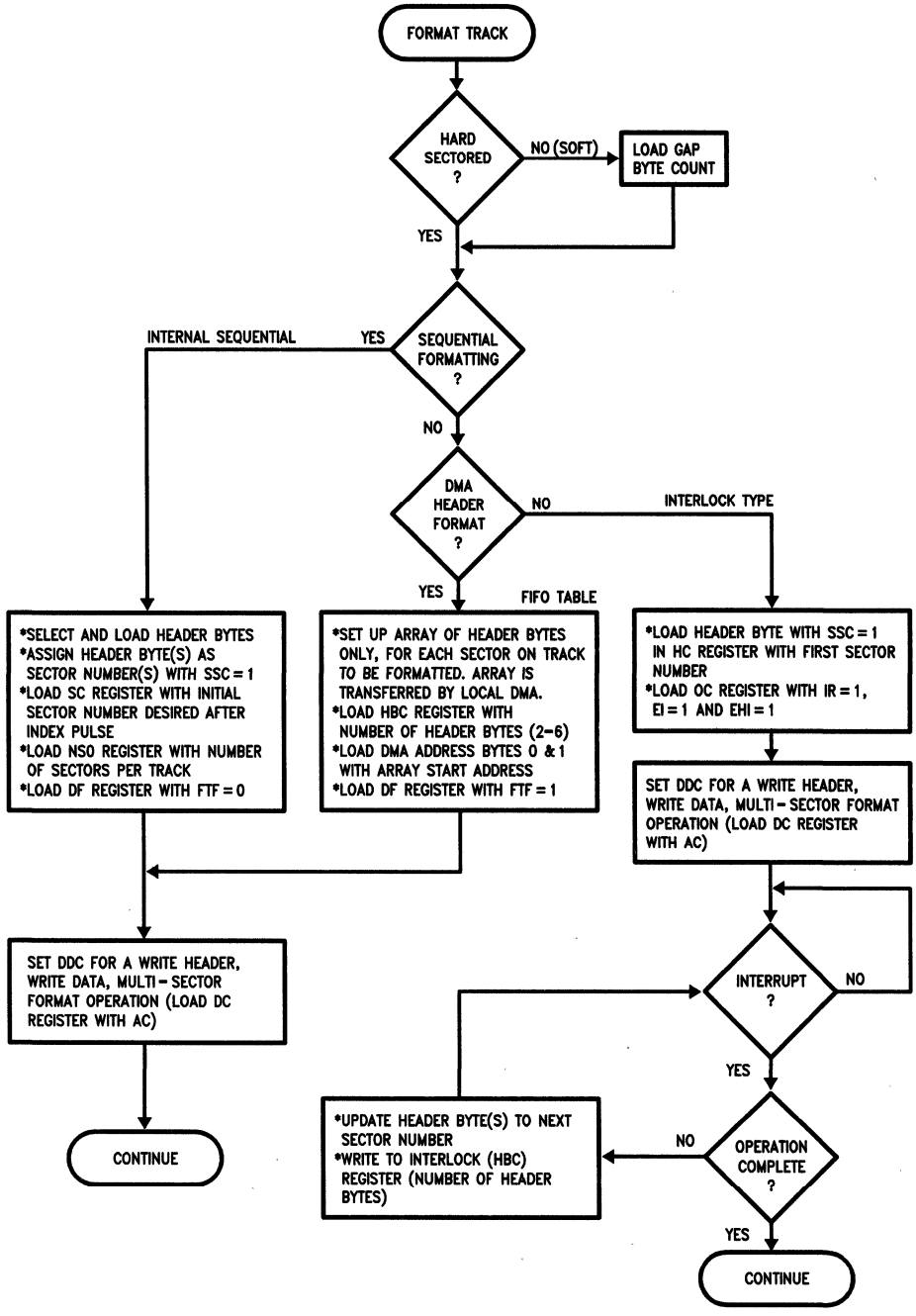
HSS	SAIS	RGATE ASSERTED:
0	0	On index pulse
0	1	On receipt of instruction
1	0	On index pulse
1	1	On index or sector pulse

WRITE

The HSS, MFM and SAM bits in the DF register determine the use of the address mark and the AME pin as follows:

HSS	MFM	SAM	FUNCTION
0	0	0	AME pin activated during ID and data synch #1 fields.
X	0	1	AME pin activated during ID preamble.
0	1	X	Missing clocks inserted in ID and data synch #1 fields. AME pin indicates LPRE (if enabled).
1	0	0	AME pin disabled.
1	1	X	Synch #1 fields written without missing clock pulse.

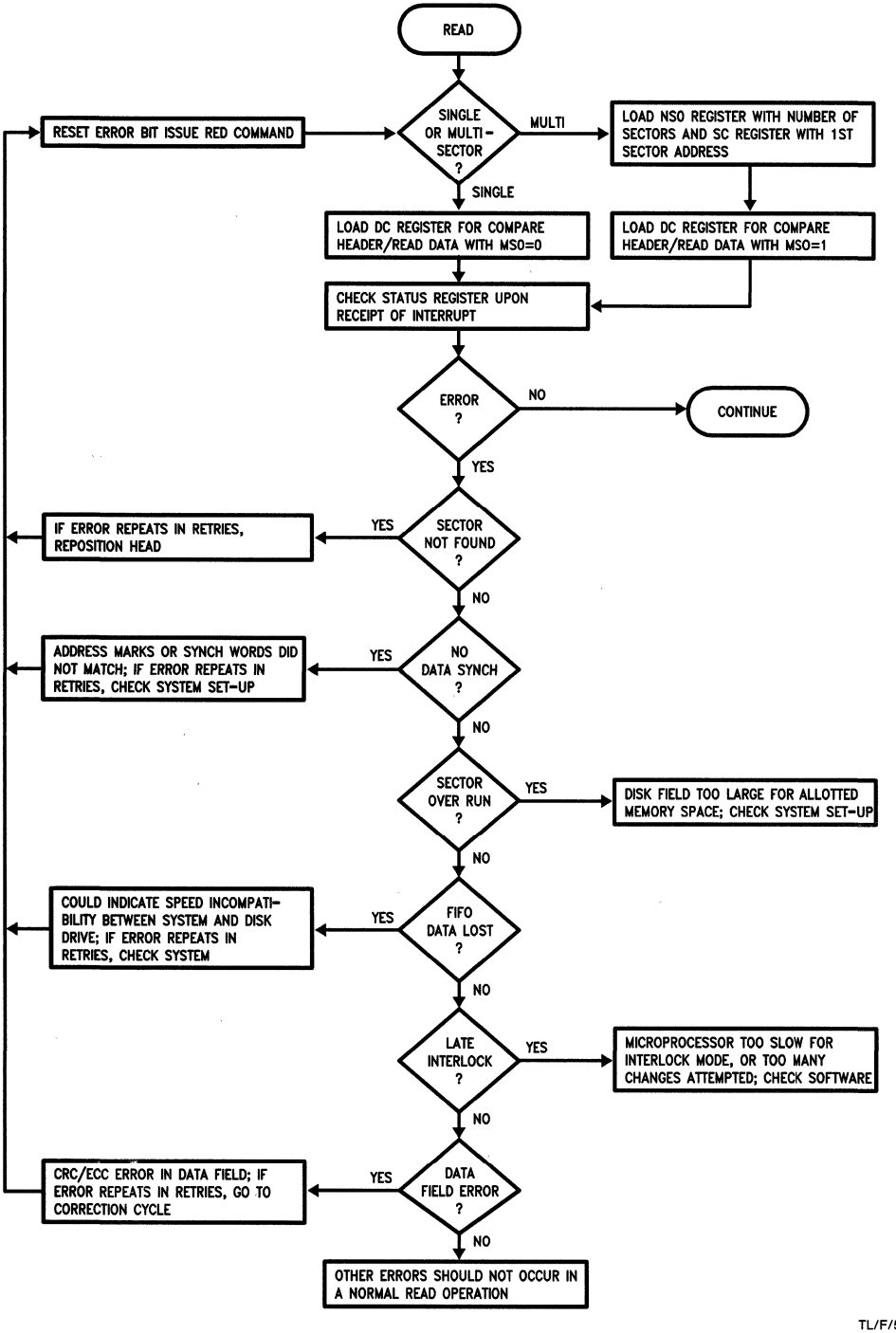
5.0 Format, Read & Write (Continued)



TL/F/5282-7

FIGURE 9. Format Track Algorithm

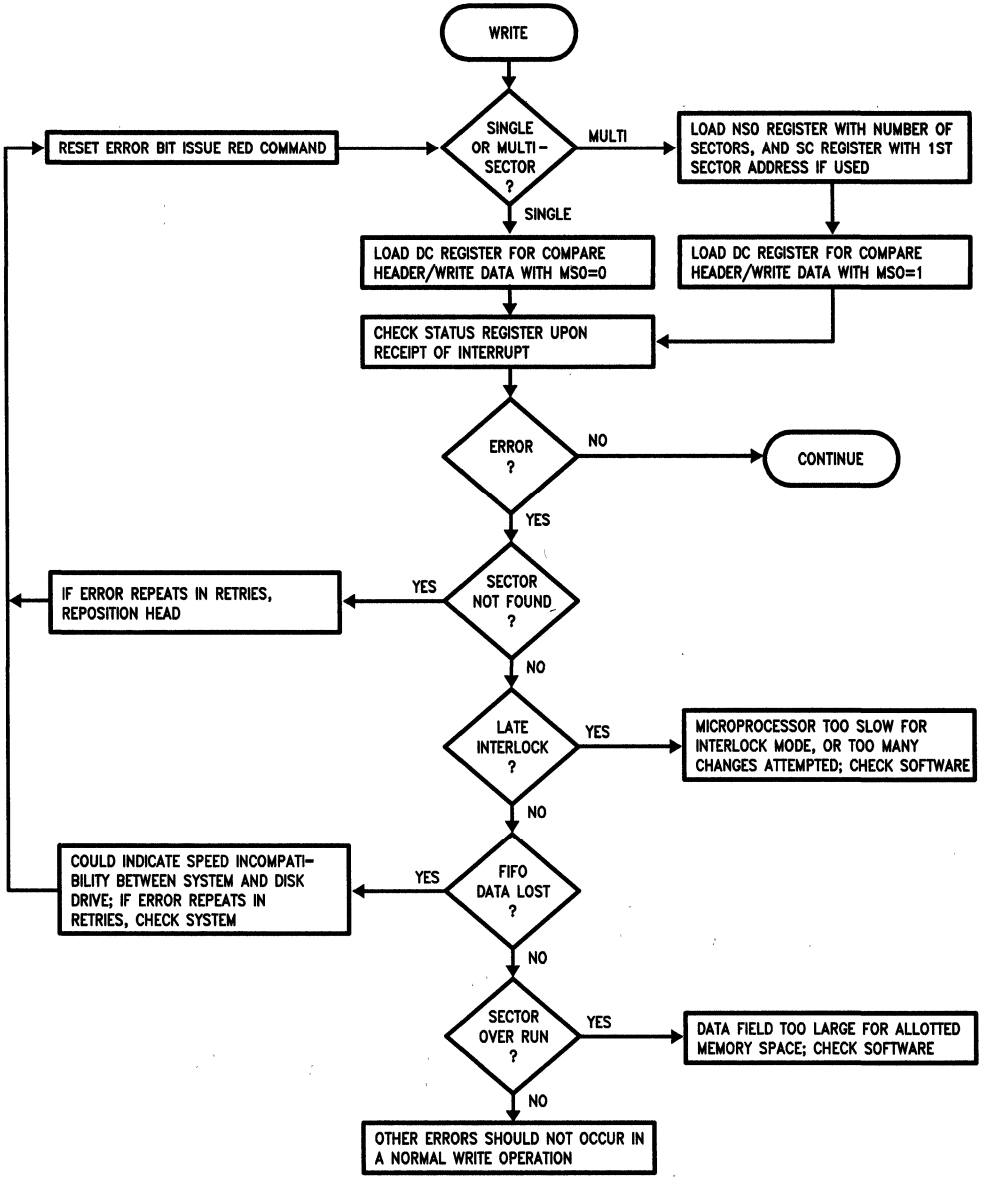
5.0 Format, Read & Write (Continued)



TL/F/5282–8

FIGURE 10. Simple Read Operation

5.0 Format, Read & Write (Continued)



TL/F/5282–9

FIGURE 11. Simple Write Operation

5.0 Format, Read & Write (Continued)

5.4 MFM ENCODED DATA

MFM encoding of write data is controlled by the MFM bit in the DF register MFM = 1 sets the DDC to write MFM data to the disk. MFM = 0 sets the DDC to write NRZ data to the disk.

PRECOMPENSATION OF MFM ENCODED DATA

When the MFM bit in the DF register and the EP bit in the OC register are set, precompensation will be indicated on the EPRE and LPRE pins. Precompensation is issued for the middle bit of a 5-bit field. In the DP8466A, early and late precompensation will be enacted for all of the combinations as shown below. All other patterns will not require precompensation. Precompensation can be disabled by setting the EP bit in the OC register inactive low.

EPRE NRZ PATTERNS	LPRE NRZ PATTERNS
00 0 10	00 1 10
00 0 11	00 1 11
01 1 00	10 0 00
01 1 01	10 0 01
11 1 00	10 1 10
11 1 01	10 1 11

Precompensation outputs are aligned to provide symmetrical set-up and hold times relative to the rising edge of the WDATA outputs. This gives a half period of RCLK set-up time on precompensation outputs. This is shown in *Figure 12*. Two bits of zero precede the preamble fields at the leading edge of the write gate when writing MFM data due to MFM encoded delays.

5.5 ADDRESS MARK PATTERNS, MISSING CLOCK

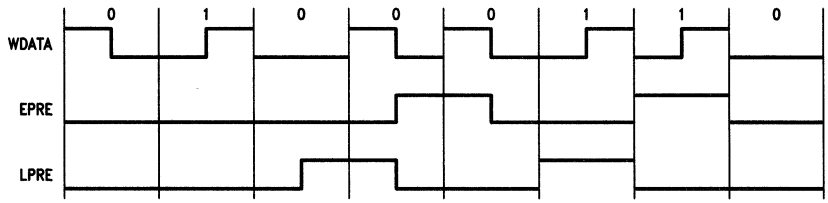
During writing and formatting a sector with MFM encoding enabled, a clock violation, or missing clock pulse, will be inserted in the synch #1 field. This indicates the address mark. For an example of this, refer to *Figure 13*.

When writing MFM encoded data with precompensation enabled, only the following hex values are allowed to be loaded into the synch #1 pattern registers:

A1, C2, C3, E1, 84, 85, 86, 87

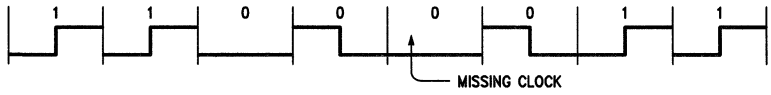
With no precompensation, any pattern containing 100001 is valid.

During a soft sectored read operation, an AMF pulse will be expected on the AMF/EPRE pin during each byte of the synch #1 field.



TL/F/5282–10

FIGURE 12. Example of EPRE and LRPE Outputs



TL/F/5282–11

FIGURE 13. Missing Clock Example

6.0 CRC/ECC

6.1 PROGRAMMING CRC

The DDC is set for internal CRC by programming the disk Format (DF) and ECC/CRC Control (EC) registers. The CRC-CCITT polynomial used by the DDC for the CRC code is given below:

$$P(x) = x^{16} + x^{12} + x^5 + 1$$

The DDC uses the pattern preset to all 1's for the CRC calculation. *Note:* If no CRC/ECC is used for the ID fields, an external ECC must be used.

6.2 PROGRAMMING ECC

There are two sets of six registers used to program the ECC. One set of six is used to program the polynomial taps, while the other set is used to establish a preset pattern (typically all 1's). Bits contained in the ECC Control (EC) register are used to control the correction span. The DF register contains bits for choosing the desired type of appendage: Either 32 or 48-bit programmable ECC polynomials, or the 16-bit CCITT CRC polynomial is possible. A 48-bit computer generated polynomial is also available from National Semiconductor free of charge.

PROGRAMMING POLYNOMIAL TAPS

To program a polynomial into the shift register, each tap position used in the code must be set to 0, and all unused taps should be set to 1. The bit assignment for these registers in 48 and 32-bit modes is shown in the tables that follow. It is important that for 32-bit codes, PTB2 and PTB3 all be set to 1's. Failure to do so will result in improper operation. Also, x^{48} and x^{32} are implied, i.e., a 32-bit ECC will always contain the x^{32} term and a 48-bit ECC will always contain the x^{48} term. For both ECC's, the term x^0 (or 1) is also implied, even though this bit is accessible.

Tap Assignment 48-Bit Mode

REG#	ADDR	*BIT NUMBER*							
		DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
PTB0	*08*	x^7	x^6	x^5	x^4	x^3	x^2	x^1	x^0
PTB1	*09*	x^{15}	x^{14}	x^{13}	x^{12}	x^{11}	x^{10}	x^9	x^8
PTB2	*0A*	x^{23}	x^{22}	x^{21}	x^{20}	x^{19}	x^{18}	x^{17}	x^{16}
PTB3	*0B*	x^{31}	x^{30}	x^{29}	x^{28}	x^{27}	x^{26}	x^{25}	x^{24}
PTB4	*0C*	x^{39}	x^{38}	x^{37}	x^{36}	x^{35}	x^{34}	x^{33}	x^{32}
PTB5	*0D*	x^{47}	x^{46}	x^{45}	x^{44}	x^{43}	x^{42}	x^{41}	x^{40}

Tap Assignment 32-Bit Mode

REG#	ADDR	*BIT NUMBER*							
		DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
PTB0	*08*	x^7	x^6	x^5	x^4	x^3	x^2	x^1	x^0
PTB1	*09*	x^{15}	x^{14}	x^{13}	x^{12}	x^{11}	x^{10}	x^9	x^8
PTB2	*0A*	1	1	1	1	1	1	1	1
PTB3	*0B*	1	1	1	1	1	1	1	1
PTB4	*0C*	x^{23}	x^{22}	x^{21}	x^{20}	x^{19}	x^{18}	x^{17}	x^{16}
PTB5	*0D*	x^{31}	x^{30}	x^{29}	x^{28}	x^{27}	x^{26}	x^{25}	x^{24}

PROGRAMMING PRESET PATTERN

To program the preset pattern that the shift registers will be preset to, PPB0–PPB5 must be initialized. As in the polynomial taps, x^{48}, x^{32}, and x^0 are implied. The assignment of the bits for 48 and 32 bit modes is shown in the tables on the following pages.

The value programmed into each register will be the preset pattern for the eight bits of the corresponding shift register. For typical operation, these will be programmed to all 1's. All unused presets must be set to 0. In 32-bit mode, PPB2 and PPB3 must be set to all 0's. Failure to do so will result in improper operation.

Preset Bit Assignment 48-Bit Mode

REG#	ADDR	*BIT NUMBER*							
		DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
PPB0	*02*	x^7	x^6	x^5	x^4	x^3	x^2	x^1	x^0
PPB1	*03*	x^{15}	x^{14}	x^{13}	x^{12}	x^{11}	x^{10}	x^9	x^8
PPB2	*04*	x^{23}	x^{22}	x^{21}	x^{20}	x^{19}	x^{18}	x^{17}	x^{16}
PPB3	*05*	x^{31}	x^{30}	x^{29}	x^{28}	x^{27}	x^{26}	x^{25}	x^{24}
PPB4	*06*	x^{39}	x^{38}	x^{37}	x^{36}	x^{35}	x^{34}	x^{33}	x^{32}
PPB5	*07*	x^{47}	x^{46}	x^{45}	x^{44}	x^{43}	x^{42}	x^{41}	x^{40}

Preset Bit Assignment 32-Bit Mode

REG#	ADDR	*BIT NUMBER*							
		DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
PPB0	*02*	x^7	x^6	x^5	x^4	x^3	x^2	x^1	x^0
PPB1	*03*	x^{15}	x^{14}	x^{13}	x^{12}	x^{11}	x^{10}	x^9	x^8
PPB2	*04*	0	0	0	0	0	0	0	0
PPB3	*05*	0	0	0	0	0	0	0	0
PPB4	*06*	x^{23}	x^{22}	x^{21}	x^{20}	x^{19}	x^{18}	x^{17}	x^{16}
PPB5	*07*	x^{31}	x^{30}	x^{29}	x^{28}	x^{27}	x^{26}	x^{25}	x^{24}

RECOMMENDED POLYNOMIAL AS AN EXAMPLE

To program the 32-bit polynomial of the form:

$$x^{32} + x^{28} + x^{26} + x^{19} + x^{17} + x^{10} + x^6 + x^2 + 1$$

with a preset of all 1's, a correction span of 5-bits with no header/data encapsulation, the following registers would be programmed as shown. Note that PTB2 and PTB3 must be all 1's and PPB2 and PPB3 must be all 0's in 32-bit mode.

6.0 CRC/ECC (Continued)

Polynomial Taps

REG#	BIT NUMBER 7	6	5	4	3	2	1	0
PTB0	1	0	1	1	1	0	1	0
PTB1	1	1	1	1	1	0	1	1
PTB2	1	1	1	1	1	1	1	1
PTB3	1	1	1	1	1	1	1	1
PTB4	1	1	1	1	0	1	0	1
PTB5	1	1	1	0	1	0	1	1

Preset Pattern

REG#	BIT NUMBER 7	6	5	4	3	2	1	0
PTB0	1	1	1	1	1	1	1	1
PTB1	1	1	1	1	1	1	1	1
PTB2	0	0	0	0	0	0	0	0
PTB3	0	0	0	0	0	0	0	0
PTB4	1	1	1	1	1	1	1	1
PTB5	1	1	1	1	1	1	1	1

ECC Control Register

BIT#	7	6	5	4	3	2	1	0
SET	1	0	0	1	0	1	0	1

6.3 OPERATION DURING CORRECTION

The DDC can be set to correct an error any time one has been detected and before another operation has begun. The user decides when to initiate the correction. The sector in question can be re-read several times to insure that the error is repeatable. If so, the error can be considered a hard error on the disk and a correction can be attempted. Since the DDC does not contain drive control circuitry, it is the user's responsibility to provide the programming for the execution of any re-read operations and the associated decision making.

The syndrome bytes in the ECC shift register will contain the bit error information. The bytes in error will already have been transferred to memory. Once initiated, the correction is performed internal to the DDC, leaving the bus free for other operations. An interrupt will be issued within the time it takes to read a sector, indicating whether the error was corrected or not. During this time, the erroneous sector in memory will remain unchanged.

Error correction time is determined by the error's location in the sector. The nearer to the start of the sector, the longer the DDC takes to locate the error. This time can be determined using the formula shown at right. It should be noted that this is internal correction time only; more time is required for the microprocessor to perform additional operations.

Before initiating a correction operation, the DDC needs to be reset, and re-enabled (see Operating Modes in DDC OPERATION). The Sector Byte Count registers must be initialized to *[sector length]* + 4 for 32-bit mode or *[sector length]* + 6 for 48-bit mode. The correction command should be issued when the counter has been updated.

The DDC will issue an interrupt after the correction cycle is complete. Other activities (such as completion of remote DMA) may issue interrupts before this happens. These interrupts should be serviced to allow the Correction Cycle Complete interrupt to be issued. The CCA bit in the Status register will be high during the entire correction cycle. It will be reset when the cycle has completed. The ED bit in the Status register will remain active throughout the correction cycle.

If after an interrupt, the Status register is read and the CCA bit is low, the Error register is read to see if the correction was successful. If the CF bit is set, this signifies that the error was non-correctable. This usually means that two errors have occurred with extremities exceeding the selected correction span. Failure to correct an error is serious and the system should be notified that the data from that sector is erroneous.

If the CF bit was not set, the error was corrected. The microprocessor then computes the address of the first byte in the data field that contains the error. That address is: *[current value of DMA Address Bytes 0 & 1] - [Sector Length] + [Data Byte Count L & H]* − 1.

Errors are corrected by XOR'ing syndrome bytes (ECC SR Out 0–5) with the bytes in the data record in memory that contain the error. The Data Byte Count can be used to determine whether the error is in the ECC or data field. If the Data Byte Count is greater than the maximum sector length, the error is in the ECC field and no correction should be attempted. If the Data Byte Count is less than the sector length, the error is in the data field (or it may straddle the data and ECC fields) and may be corrected.

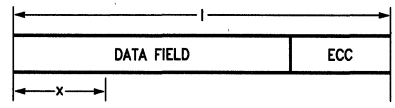
For performing a correction with 32-bit ECC, the following shift registers should be read sequentially to obtain the syndrome byte pattern:

ECC SR Out 1, ECC SR Out 4, ECC SR Out 5

ECC SR Out 2 and 3 are not used in 32-bit mode and will contain 0's if read. ECC SR Out 0 will contain all 0's if the error is correctable, and may contain some set bits if it is not.

ECC SR Out 1 will always contain the first bits in error. The succeeding bits will be contained in ECC SR Out 4 and 5. If the maximum span of 15 bits is used, all three registers may be needed, depending on where the first bit occurs.

To correct the error, the syndrome bits in these registers are XOR'ed with the data bits contained in buffer memory. The corrected data is then written back to the buffer memory, replacing the data in error. The address of the first byte in error is computed by the microprocessor as described above.



TL/F/5282–12

Approximate Correction Time = $(I - x)/f$

I = Entire length of data field and ECC appendage (in bits)

x = Distance from least significant bit to first error location (in bits)

f = read clock frequency (in hertz)

FIGURE 14. Calculating Correction Time

6.0 CRC/ECC (Continued)

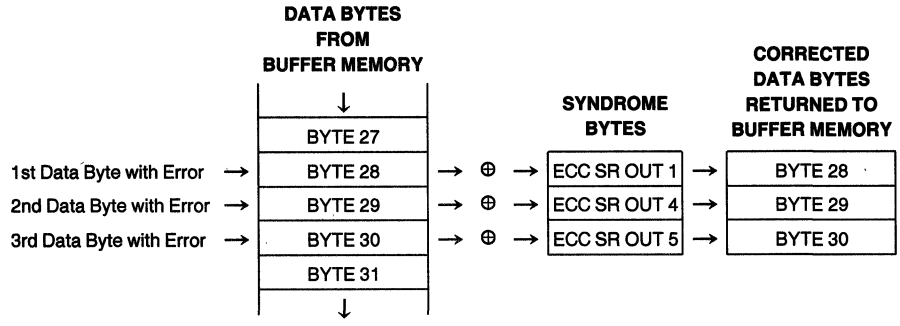


FIGURE 15. 32-Bit ECC Correction Process

To perform a 48-bit ECC correction, the following registers should be read sequentially:

ECC SR Out 1, ECC SR Out 2, ECC SR Out 3

ECC SR Out 0, 4 and 5 are not used for outputting syndrome bits for correction in 48-bit mode and will contain 0's for a correctable error. If the error is non-correctable, these registers may contain some set bits. Syndrome bit location and error correction is performed as in 32-bit mode.

EXAMPLE OF A 32-BIT CORRECTION

Shown in *Figure 17*, is a record with several bits read in error from disk. Bits D4, D11, D13 and D14, now located in memory, were incorrectly and need to be corrected. As can be seen, the correction pattern provided in ECC SR Out 1 and 2 can be used to correct bits D4, D11, D13 and D14. The CPU reads the Data Byte Count and computes that it points to the first byte read from disk. This byte is XOR'ed with ECC SR Out 1 and is written back to memory. The second byte read from the disk is XOR'ed with ECC SR Out 4 and then written back. ECC SR Out 5 need not be used since it contains all 0's.

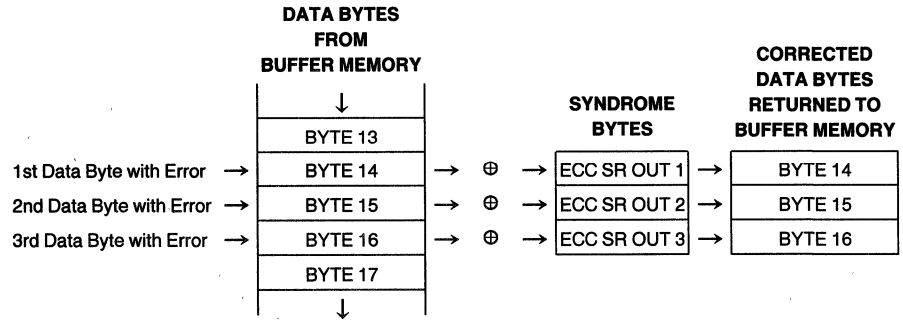


FIGURE 16. 48-Bit ECC Correction Process

Syndrome Pattern

REGISTER	*BIT NUMBER* 7	6	5	4	3	2	1	0
ECC SR OUT 1	0	0	0	1	0	0	0	0
ECC SR OUT 4	0	1	1	0	1	0	0	0
ECC SR OUT 5	0	0	0	0	0	0	0	0

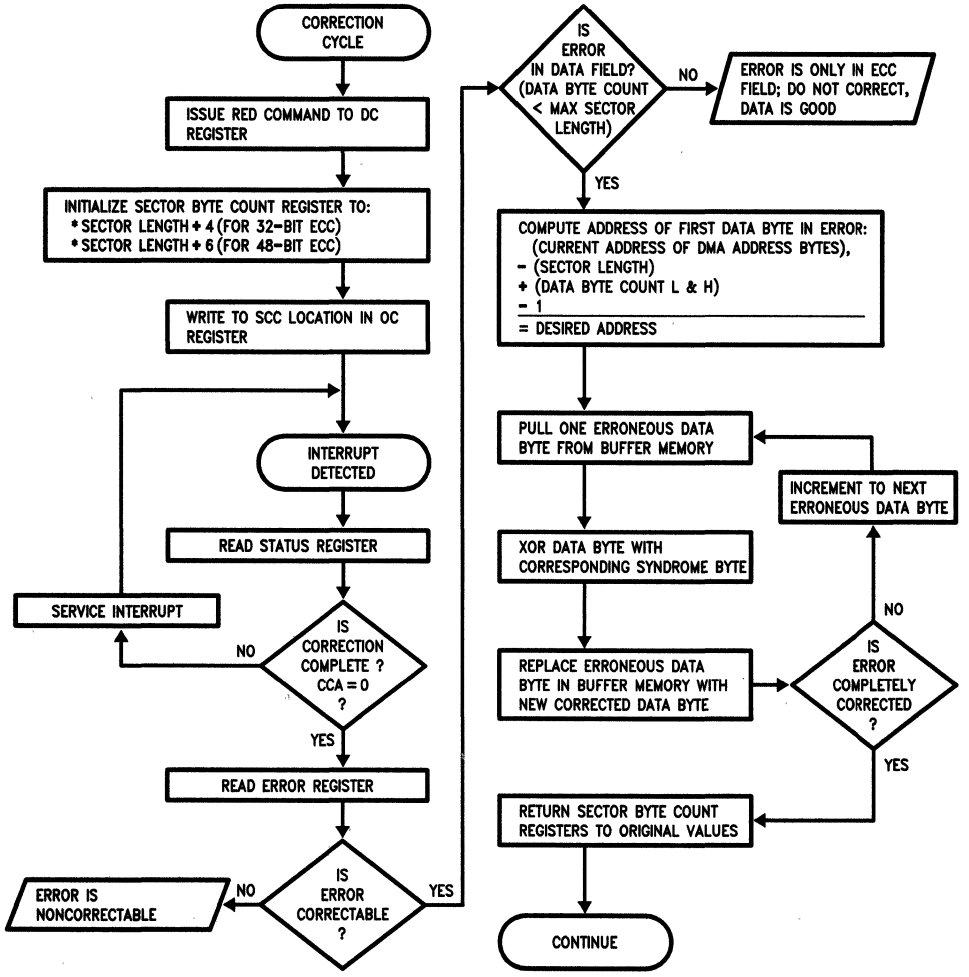
Buffer Memory

CORRESPONDING BUFFER DATA BIT PATTERN							
D7	D6	D5	*	D3	D2	D1	D0
D15	*	*	D12	*	D10	D9	D8
D23	D22	D21	D20	D19	D18	D17	D16

* = location of bits in error

FIGURE 17. Example of a 32-Bit Correction

6.0 CRC/ECC (Continued)



TL/F/5282-13

FIGURE 18. Correction Cycle Algorithm

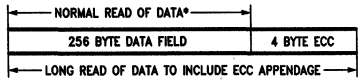
THIS CYCLE CAN ONLY BE INITIATED AFTER A READ DATA OPERATION HAS BEEN COMPLETED

6.0 CRC/ECC (Continued)

A note of caution: If the DDC is in the tracking DMA mode when a data error occurs, the remote DMA channel will transfer the sector in error to its destination in the system. The DDC will still interrupt to indicate that it has detected an error. It is then up to the system to get the DDC to correct the error in buffer memory and retransfer the corrected data to the system.

6.4 ECC CHECK USING LONG READ AND LONG WRITE

During a normal read or write operation, the size of the data field is specified by the Sector Byte Count register pair. If the data field is extended during a readback, the ECC appendage can be read in as data and analyzed outside the DDC. This is what is known as a *long read*.



TL/F/5282–14

*Read length defined by Sector Byte Count register pair.

FIGURE 19. Example of a Long Read

Likewise, an externally generated ECC appendage can be added to the data and written to the disk as data with or without the onboard CRC/ECC generator enabled. This is known as a *long write*.

By using long reads and long writes in conjunction with external software used to produce data fields and external CRC/ECC appendages, various diagnostic programs can be devised to test the DDC's internal correction functions and ECC generation circuitry. These tests could be incorporated in the initialization algorithm to test the chip each time it is powered up.

7.0 Data Transfer

7.1 DIRECT MEMORY ACCESS (DMA)

The DDC is designed to work efficiently in two major system configurations:

(1) A single system bus with shared data buffer/system memory (see *Figure 20*).

(2) A dual bus environment with a local microprocessor, buffer memory and DP8466A on a local bus interfacing the host system bus through an I/O port (see *Figure 21*).

All DMA activity is supported by the following three features:

PROGRAMMABLE BURST LENGTH (THRESHOLD)

Here, the transfer of data between the 32-byte FIFO on the DDC and the external memory (local or main) involves the use of internal or external local DMA channel. While writing to the disk, the DDC will initiate a transfer when the FIFO has been depleted by the burst length. It will also initiate a transfer while reading from the disk when the FIFO fills to the burst length. This length is selectable from 2, 8, 16 or 24 bytes, allowing for the variations in bus latency time encountered in most systems.

At the start of a write operation, the FIFO will be filled up in a series of bursts of the programmed length.

If the exact burst option is not selected, the FIFO will be completely filled (if writing to disk) or emptied (if reading from disk) in one DMA operation. The burst length is always the threshold at which the transfer will be requested and is independent of the DMA mode, including slave.

At the end of a sector or an operation, the local burst counter does not reset. This means that the first burst of a subsequent sector will not be what was programmed in the LTR if the burst length was not an exact multiple of the data length. The data length is equal to the sector length times the number of sectors. The DDC would have to be reset between operations if resetting the local burst counter is desired. It is not recommended to count bursts in order to monitor the amount of data transferred.

8-BIT/16-BIT WIDE TRANSFERS

Byte or word wide data transfer can be selected for both local and remote DMA channels. Word wide tranfers with local DMA use the AD0–15 pins, and byte wide use the AD0–7 pins. Both the local and the remote DMA addresses are incremented by 2 for word wide transfers, and 1 for byte wide transfers. Commands and DDC parameter registers are loaded and read only 8-bits at a time, using AD0–7.

REVERSE BYTE ORDER

This option is only valid for 16-bit wide transfers using the local DMA channel. This should not be used for 8 bit wide transfers. It enables the two bytes being transferred to be mapped with the high order byte to AD0–7 and the low order byte to AD8–15, or vice-versa.

The DDC has provisions to accommodate five DMA modes. These are as follows:

EXTERNAL DMA:	1. Slave Mode
INTERNAL DMA, Single Bus:	2. 16-Bit Local Mode
	3. 32-Bit Local Mode
Multiple Bus:	4. Non-Tracking Mode
	5. Tracking Mode

All five modes accommodate the three configurations just described. All DMA modes, except external slave, use an incrementing address. Local channel transfers always have priority over remote channel transfers unless externally reprioritized. If the local channel is used, its transfer length is always automatically loaded from the Sector Byte Count register pair.

7.2 EXTERNAL DMA

SLAVE MODE

In this mode, no on-chip DMA control is used. LRQ and LACK pins are connected to an external DMA controller. After LACK has been granted, I/O RD and I/O WR from the DMA controller are used to strobe data between the internal FIFO and the DDC I/O port. 8-bit and 16-bit wide data transfers are possible. Throughout this data sheet, reference has been made to the use of on-chip DMA for the transfer of data. It is important to note here that external DMA can be used in place of this if so desired.

7.3 INTERNAL DMA

The following four modes all use on-chip DMA control with at least the local channel serving as bus master for data transfers between the internal FIFO and memory.

SINGLE BUS SYSTEMS

The following two modes support a single bus and a single shared buffer/system memory. Bus access should be guaranteed before the FIFO overflows or empties during a disk transfer operation. A FIFO Data Lost error (FDL bit in Error register) will be flagged and the operation aborted if this fails

7.0 Data Transfer (Continued)

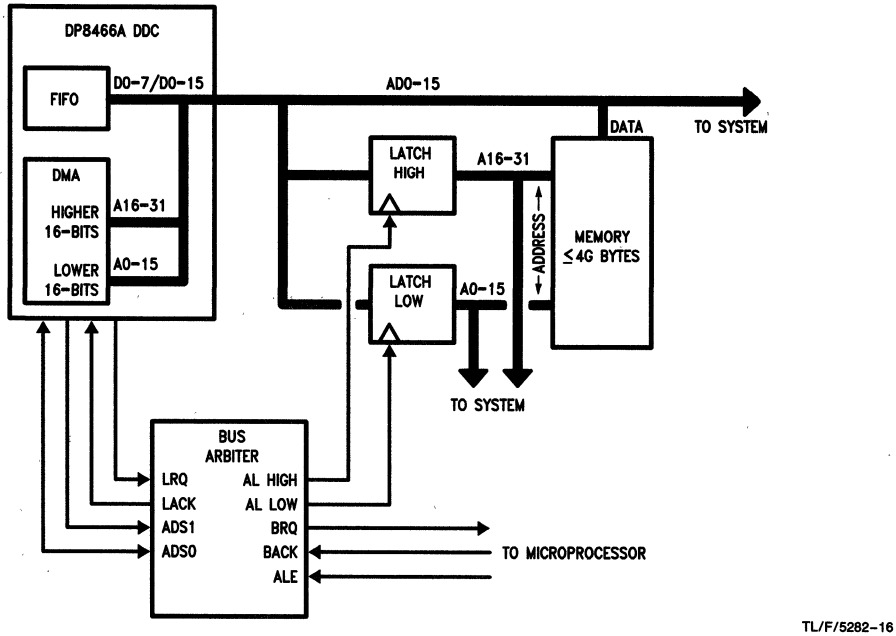


FIGURE 20. Single System Bus, 32-Bit Address DMA

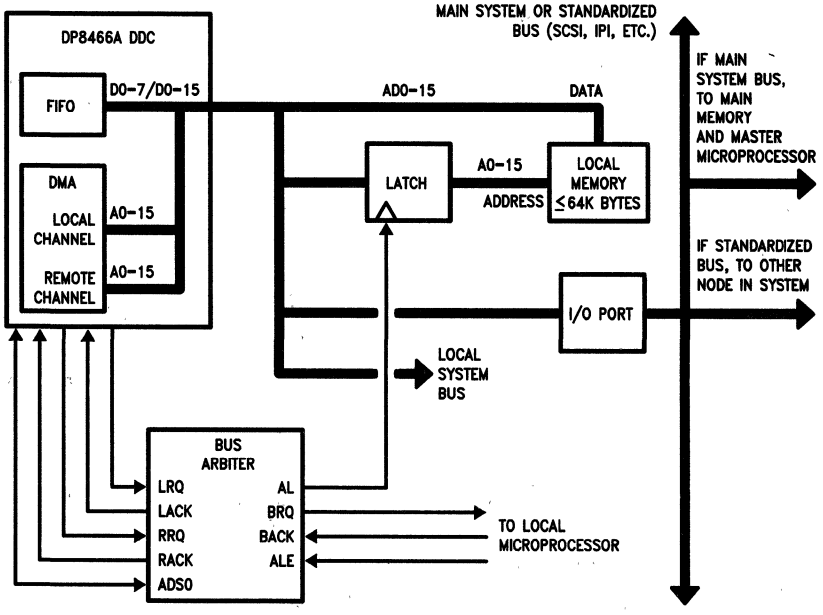


FIGURE 21. Dual System Bus, 16-Bit Address DMA

7.0 Data Transfer (Continued)

to happen. Different system latency times can be accommodated by the selectable burst length.

16-BIT LOCAL MODE

SLD bit is set and LA bit is reset in the LT register. Only the 16-bit local DMA channel is enabled. 64k bytes are directly addressable by the DDC. Address data is presented on AD0–15 and latched with ADS0. Transfers always take 4 BCLK cycles if no wait states are issued.

32-BIT LOCAL MODE

SLD bit and LA bit are both set in the LT register. SRD bit in the RT register must be reset. The local DMA channel is now set to issue 32-bit addresses using the remote DMA channel as the upper 16-bit address register. 4 G bytes are addressable by the DDC. During the first DMA cycle of a newly programmed address, or after a roll-over of the lower 16-bit address counter occurs, ADS1 strobes a new high order word (A16-31) into the external address latches. Each time this happens, the DMA cycle is 5 BCLK periods long. When a new high order address is not needed, the DMA cycle is 4 BCLK periods long. ADS0 is used as an output to latch the low order word (A0–15) from the AD0–15 pins into the address latch.

MULTIPLE BUS SYSTEMS

The following two modes support a dual bus environment, where a local microprocessor, buffer memory and the DP8466A interface to the host through an I/O port. The difference between tracking and non-tracking mode is whether the DDC or the controlling microprocessor ensures that an attempt to read data from buffer memory does not occur before data has been written there. Basic algorithms for both are shown in *Figures 22* and *23*.

TRACKING MODE

SLD bit set and LA bit reset in the LT register. SRD bit and TM bit set in the RT register. The DDC ensures that data is not overwritten by data transferred from the FIFO.

This mode effectively turns the buffer memory into a large FIFO. This is accomplished through the use of the DMA Sector Counter (DSC), which keeps track of the difference between sectors read/written to the disk and the sectors transferred to/from the host system. Each time the source transfers a sector of data into buffer memory (length determined by the Sector Byte Count register pair), the DSC register is incremented. It is decremented each time the destination has transferred a sector of data. Whenever the DSC register contents become zero, destination transfers are inhibited. This mode facilitates multi-sector operations.

Example: Tracking Mode, Disk Read

- Source is local DMA
- Destination is remote DMA
- DSC register is reset automatically upon start of operation
- Local and remote start address, SC, NSO, OC and finally DC registers are loaded. Other registers may need to be updated, but this is a minimum set.

A sector is read from the disk and is transferred in bursts from the FIFO to the buffer memory by local DMA. The DSC register then increments and the remote channel can begin transferring the first sector from the buffer memory to the host system. Burst transfers can be interleaved with local DMA, remote DMA and microprocessor all sharing the bus. The local channel bursts have priority over remote bursts. If the remote channel manages to transfer a sector before the local channel has completed the next sector, the DSC register will decrement to zero. Further remote transfers are inhibited until the local channel completes another sector and increments the DSC. In other words, each time a local sector has been transferred, the DSC is incremented and each time a remote sector completes, the DSC is decremented. Therefore, the DDC prevents further buffer memory contents that have not been previously loaded with valid data by the local DMA from being transferred to the host system. The remote channel continues operation until the last byte from the buffer memory has been transferred. An interrupt is issued upon completion of the operation.

NON-TRACKING MODE

SLD bit set and LA bit reset in the LT register. SRD bit set and TM bit reset in the RT register. The remote and local channel addresses are completely independent. The controlling microprocessor must insure that the data to be transferred by the remote channel is not over-written by the local channel and vice-versa. DMA address and count registers are set up independently. Remote start address (DMA Address Bytes 2 and 3) and Remote Data Byte Count registers must be loaded before SRI or SRO bits are set in the OC register. Local or remote transfers may already be in progress when the other channel is started. The local channel has priority over the remote channel. Local bus utilization is then interleaved between the local channel, the remote channel and the controlling microprocessor.

By setting both SRI and SRO simultaneously, any non-tracking remote DMA operation will stop. The present remote address and remote data byte count will be retained and the local DMA will be unaffected. Loading the original OC instruction (input or output) will restart the original instruction from the last remote DMA address.

DMA Mode Select Table

DMA Mode	LT Register SLD	LT Register LA	RT Register SRD	RT Register TM
SLAVE	0	0	0	0
16-BIT LOCAL	1	0	0	0
32-BIT LOCAL	1	1	0	0
TRACKING	1	0	1	1
NON-TRACKING	1	0	1	0

NOTE: In either tracking or non-tracking mode, if either channel is loaded with an odd byte transfer count, the DDC will transfer the next higher even number of bytes. For example, if 511 was loaded into the Remote Data Byte Count registers, 512 bytes would be transferred, with valid data only in the first 511 bytes.

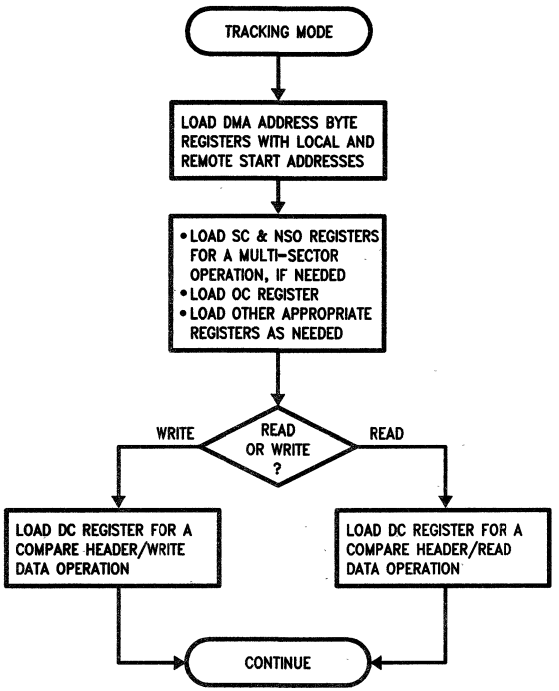
DMA WAIT STATES

INTERNAL

Both DMA channels can independently be set to lengthen the RD and WR strobes by one clock cycle (LSRW bit in the LT register and RSRW bit in the RT register). This lengthens each transfer from 4 cycles to 5 cycles of the BCLK.

EXTERNAL

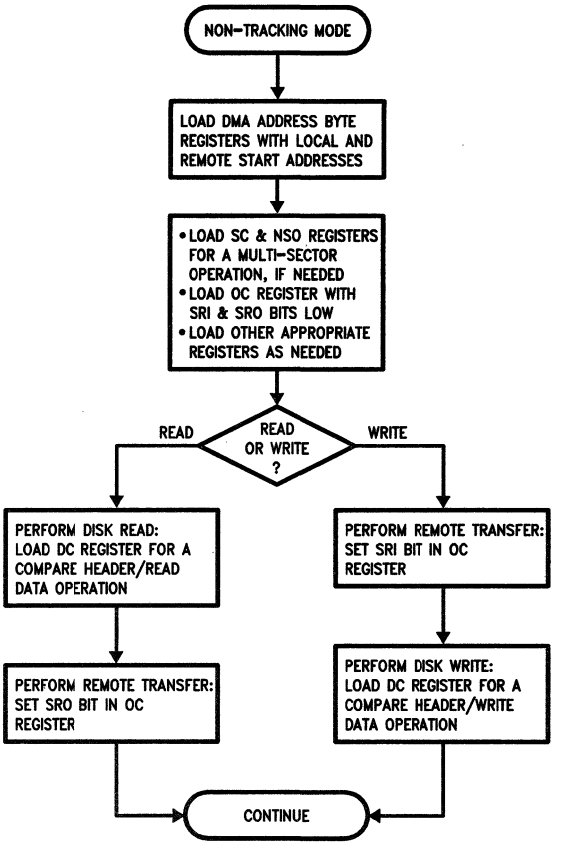
By enabling the external wait states (in the RT register), the EXT STAT pin is configured to insert wait states in each RD and WR pulse as long as this input is high. This is valid for both the local and remote DMA channels.



TL/F/5282–17

NOTE: DMA operation is completely automatic for the duration of the command. For example, when reading disk, local DMA empties/fills the FIFO and remote DMA transfers data at least one sector behind the local channel to an I/O port. For disk write, local channel will be at least one sector behind the remote channel.

FIGURE 22. Tracking Mode for Normal Disk Read/Write



TL/F/5282–18

NOTE: This is the most basic of non-tracking mode operations, and unlimited, more versatile algorithms can be built up from this.

FIGURE 23. Non-Tracking Mode for Normal Disk Read/Write

8.0 Interrupts

Interrupts can only occur if the EI bit in the OC register is set. If it is not set, the INT pin is always de-asserted high. 16 RCLK periods (3.2 μs at 5 Mbit/sec data rate) must pass before servicing an interrupt (i.e. reading Status). Failure to do this will result in servicing the same interrupt twice. There are four general conditions that may cause an interrupt to occur:

Operation Complete
Header Complete
Error
Correction Cycle Complete

OPERATION COMPLETE

This interrupt indicates that the current DDC operation has completed and the DDC is ready to execute a new command. Commands can be loaded sooner by setting EHI bit in the OC register. The Next Disk Command (NDC) bit in the Status register is set coincident with the Header Complete interrupt. New disk commands can be loaded before DMA operation is finished if NDC is set. If the command is a multi-sector operation, the end of operation interrupt will occur only after the operation is completed in the last sector of operation. The INT pin is asserted low when:

- Disk operation is completed for any command that is not a disk read operation.
- A read operation in the tracking DMA mode after the remote transfer is complete.
- A read operation in the non-tracking DMA mode after the local transfer is complete.
- A non-tracking mode remote DMA transfer is completed. This is independent of the disk operation or the local DMA.

HEADER COMPLETE:

If the EHI and EI bits are set in the OC register, an interrupt will occur when any header operation is complete. Multi-sector operations will generate an interrupt after each header in each sector has been operated on. It is asserted two bit times into the ID postamble. This function allows the changing of header bytes (and parameter RAM in general) *on the fly*. The Header Complete interrupt can be used in conjunction with the Interlock Required (IR) bit in the OC register set to insure that changes have been completed before the next sector is encountered (see Interlock Type formatting). Another normal mode of use would be to notify the controlling microprocessor when the next disk command can be loaded. This interrupt is coincident with the Next Disk Command (NDC) bit being set in the Status register.

ERROR

Any bit set in the Error register sets the ED bit in the Status register and causes an interrupt.

CORRECTION CYCLE COMPLETE

An interrupt will occur at the end of an internal correction cycle, regardless of whether the error was corrected or not. If the error was non-correctable, the CF bit will be set in the Error register. This will not generate two interrupts.

CLEARING INTERRUPTS

The INT pin will be forced inactive high any time the Status register is being read. If an interrupt condition arises during a status read, this condition will assert INT as soon as the status read is finished.

Interrupts can also be cleared by setting the internal RES bit, or by asserting the external RESET pin.

9.0 Additional Features

9.1 DATA RECOVERY USING THE INTERLOCK FEATURE

The potential use of the interlock feature is in recovering data from a sector with an unreadable header field. It is assumed that the number of the sector physically preceding the bad sector on the disk is known. A single-sector operation will be performed on these sectors, and the Drive Command register will be changed in between them. The following steps will recover the data:

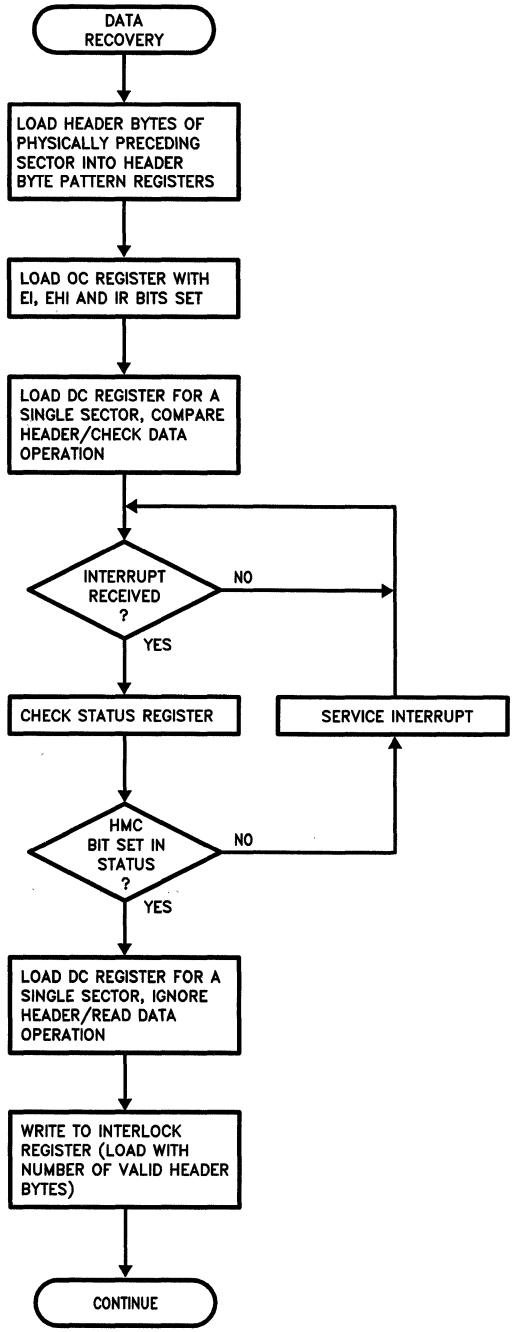
- The header bytes of the physical sector preceding the desired sector are loaded into the relevant byte pattern registers.
- The OC register must be loaded with the EI, EHI and IR bits set. This enables the Header Complete interrupt as well as the interlock feature.
- The DC register is loaded for a single-sector, Compare Header/Check Data operation.
- After the Header Complete interrupt, the DC register must be loaded with an Ignore Header/Read Data operation, and the Interlock (HBC) register written to. If the controlling microprocessor fails to write to the HBC register before the end of the data field of the first sector, a Late Interlock error (LI bit in Error register) will be flagged, and the operation will be terminated with an interrupt.
- When the HMC interrupt occurs on the second sector, the Interlock (HBC) register must be written to again in order to avoid LI error.
- The operation will terminate normally when the data from the badly labeled sector has been read.

9.2 HFASM FUNCTION

The Header Failed Although Sector number Matched (HFASM) function on the DDC can be used to perform maintainance and diagnostic functions, both of which will be briefly outlined here.

The HFASM function is enabled by setting the EHF bit in at least one of the Header Control registers, with a Compare Header command loaded into the DC register. More than one header byte may have its EHF bit set. If any one of the header byte(s) with it's EHF bit set matched, but any other header byte(s) (regardless of the state of their EHF bit) don't match, an HFASM error will occur.

9.0 Additional Features (Continued)



TL/F/5282–19

FIGURE 24. Data Recovery Algorithm

9.0 Additional Features (Continued)

In this way, the HFASM function performs a maintenance type function, and can often indicate that the head is positioned over the wrong track. It is independent of whether or not a CRC failure has occurred. An HFASM failure will not stop operation until the header CRC bytes have been compared and the CRC check is completed.

To perform a diagnostic function, the header can be read and analyzed. This can be done only during a Compare Header/Check Data operation with HFASM enabled. This causes the header patterns coming from the disk to be written into the FIFO. We must assume that the FIFO is empty (or has been reset before the operation) in order for this operation not to interfere with data transfers. If an HFASM error occurs during a Header Compare, the FIFO will be left intact and the header with the error can be read out of the FIFO from the Header Diagnostic Readback (HDR) register. (Note: LWDT of the local transfer register must be set to match the bus width of the accessing MP for this function.)

If an HFASM error did not occur, the FIFO will be cleared and the header patterns that were stored there will be lost. This process can only be enabled for one disk command. The Compare Header/Check Data command will enable this function. Any other command will disable it.

10.0 Typical System Configurations

10.1 LOW COST SYSTEM

In a single bus system, the DDC can directly address 4G bytes of main memory. The 16-bit I/O port (AD0–15) is externally demultiplexed and buffered with the octal latches and drivers. The main microprocessor, through a separate disk drive control I/O block, is responsible for commands like Head Select, Seek, TRK 000, Drive select, etc. Bus access must be guaranteed before the FIFO overflows or empties. A short burst length (LT and RT registers) accommodates longer bus latency times and helps to insure this. The burst capability allows for other bus operations to be interleaved while the FIFO is filling (during a read) or emptying (during a write). If long, important CPU operations are required, the next configuration must be used.

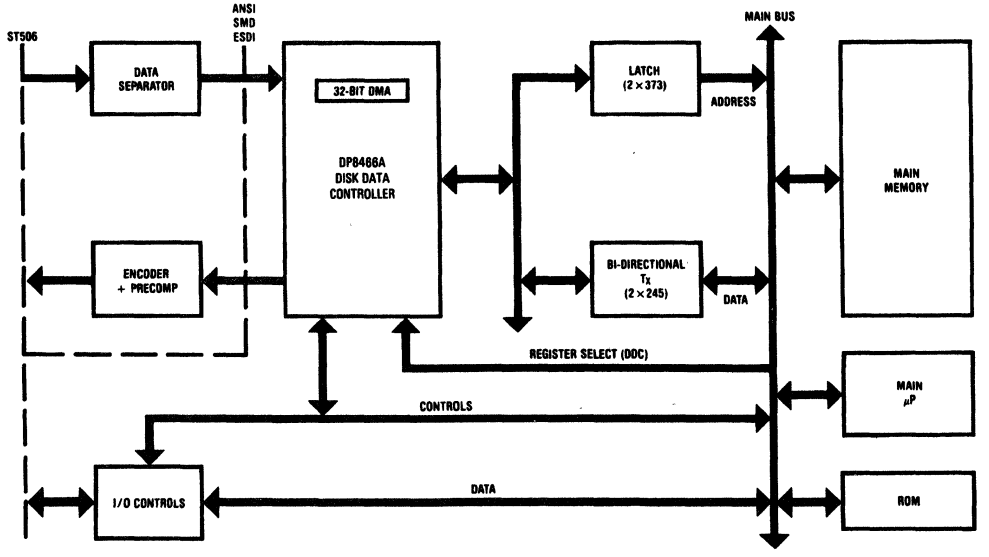


FIGURE 25. Low Cost System Configuration

TL/F/5282–20

10.0 Typical System Configuration (Continued)

10.2 HIGH PERFORMANCE SYSTEM

This configuration provides a local bus for the DDC to share with the local microprocessor and a buffer memory. Here, whole blocks of data can be transferred between the DDC and buffer memory without interfering with the system bus. This leaves the main CPU to perform important operations and to allow data transfers when it is ready. This configuration is also used in intelligent drives or systems that comply to SCSI or IPI specifications. A local bus, dedicated microprocessor and buffer memory are main characteristics of an intelligent disk interface. The buffer memory can be used as a cache for track or file buffering and command lists can be down-loaded for execution by the microprocessor. The two DMA channels can both directly address 64k bytes of buffer memory. The local DMA channel transfers data between the buffer memory and the internal FIFO. The remote DMA channel transfers data between the buffer memory and the host I/O port. With the addition of a bi-directional buffer isolating the DDC from the microprocessor, simultaneous drive operations can be accomplished. While the DDC is transferring data via DMA with the buffer memory, the local microprocessor can issue drive control commands.

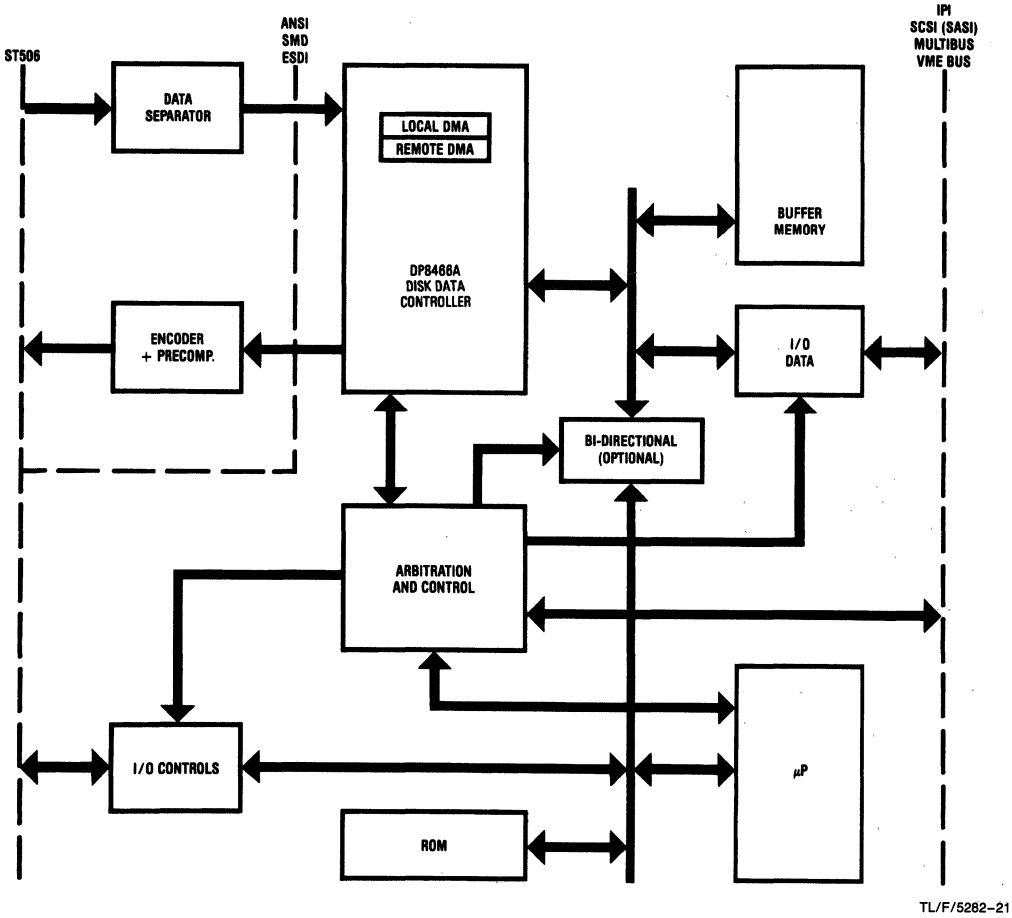


FIGURE 26. High Performance System

INDEX

11.0 Absolute Maximum Ratings*

If Military/Aerospace specified devices are required, please contact the National Semiconductor Sales Office/Distributors for availability and specifications.

Supply Voltage (V_{CC})	−0.5 to +7.0V
DC Input Voltage (V_{IN})	−0.5 to V_{CC}+0.5V
DC Output Voltage (V_{OUT})	−0.5 to V_{CC}+0.5V
Storage Temperature Range (TSTG)	−65°C to +150°C
Power Dissipation (PD)	500 mW
Lead Temperature (TL) (Soldering 10 sec.)	260°C
ESD Tolerance: C_{ZAP} = 100 pF, R_{ZAP} = 1500Ω	1600V

*Absolute Maximum Ratings are those values beyond which damage to the device may occur.

12.0 DC Electrical Characteristics (V_{CC} = 5V ±10%, unless otherwise specified) T_A = 0°C to +70°C

Symbol	Parameter	Conditions	Typ	Limit	Units
V_{IH}	Minimum High Level Input Voltage	(Note 1)		2.0	V
V_{IL}	Maximum Low Level Input Voltage	(Note 1)		0.8	V
V_{OH1}	Minimum High Level Output Voltage (Note 2)	$\|I_{OUT}\|$ = 20 μA		V_{CC} − 0.1	V
V_{OH2}		ADS0, ADS1 $\|I_{OUT}\|$ = 4.0 mA For All Other Outputs $\|I_{OUT}\|$ = 2.0 mA		3.5	V
V_{OL1}	Maximum Low Level Output Voltage (Note 2)	$\|I_{OUT}\|$ = 20 μA		0.1	V
V_{OL2}		ADS0, ADS1 $\|I_{OUT}\|$ = 4.0 mA For All Other Outputs $\|I_{OUT}\|$ = 2.0 mA		0.4	V
I_{IN}	Maximum Input Current	V_{IN} = V_{CC} or GND		±1	μA

12.0 DC Electrical Characteristics

(V_{CC} = 5V ±10%, unless otherwise specified) T_A = 0°C to +70°C (Continued)

Symbol	Parameter	Conditions	Typ	Limit	Units
I_{OZ}	Maximum TRI-STATE Output Leakage Current	$V_{OUT} = V_{CC}$ or GND		±10	μA
I_{CC}	Average Supply Current DP8466AN-12 (Note 3)	$V_{IN} = V_{CC}$ or GND BCLK = RCLK = 12 MHz I_{OUT} = 0 μA	12	30	mA
	Average Supply Current DP8466AN-20 (Note 3)	$V_{IN} = V_{CC}$ or GND RCLK = 20 MHz BCLK = 16 MHz, I_{OUT} = 0 μA	20	40	mA
	Average Supply Current DP8466AN-25 (Note 3)	$V_{IN} = V_{CC}$ or GND BCLK = 20 MHz RCLK = 25 MHz I_{OUT} = 0 μA	25	45	mA

Note 1: Limited functional test patterns are performed at these levels. The majority of functional test patterns are performed with input levels of 0V and 3V for AC Timing Verification.

Note 2: Outputs are "conditioned" for Tested States by normal functional test patterns. Device clocks are disabled and a purely static measurement is performed.

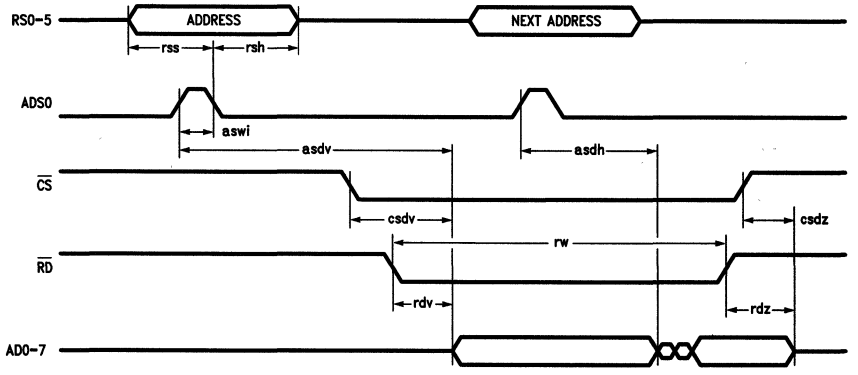
Note 3: Device is in normal operating mode and is measured with bypass capacitor of 0.1 μF between V_{CC} and Ground.

13.0 AC Electrical Characteristics & Timing Diagrams

NATIONAL SEMICONDUCTOR PRELIMINARY TIMING FOR THE DP8466A

Note: Refer to 11.4 for AC Timing Test Conditions.
Refer to 11.5.6 for derating factor.

13.1 REGISTER READ (Latched Register Select: ADS0 Active)



TL/F/5282–22

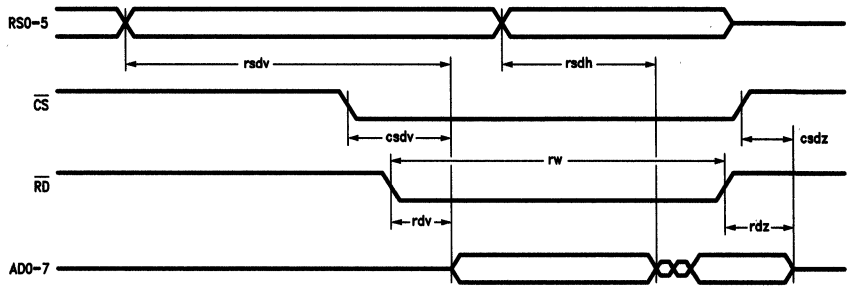
Symbol	Parameter	DP8466A-25/20		DP8466A-12		Units
		Min	Max	Min	Max	
rss	Register Select Setup to ADS0 Low	10		15		ns
rsh	Register Select Hold to ADS0 Low	10		15		ns
aswi	Address Strobe Width In	20		30		ns
asdv	Address Strobe to Data Valid (Note 1)		150		200	ns
csdv	Chip Select to Data Valid		125		150	ns
rdv	Read Strobe to Data Valid		125		150	ns
rw	Read Strobe Width		10		10	μs
csdz	Chip Select to Data TRI-STATE (Note 2)	20	80	20	90	ns
rdz	Read Strobe for Data to TRI-STATE (Note 2)	20	80	20	90	ns
asdh	Data Hold from ADS0 (Note 1)	20		20		ns

Note 1: asdv and asdh timing is referenced to the leading edge of ADS0 or the leading edge of valid address, whichever comes last.

Note 2: TRI-STATE note: These limits include the RC delay inherent in our test method. These signals typically turn off within 15 ns, enabling other devices to drive this line with no contention.

13.0 AC Electrical Characteristics & Timing Diagrams (Continued)

13.2 REGISTER READ (Non-Latched Register Select: ADS0 = 1)



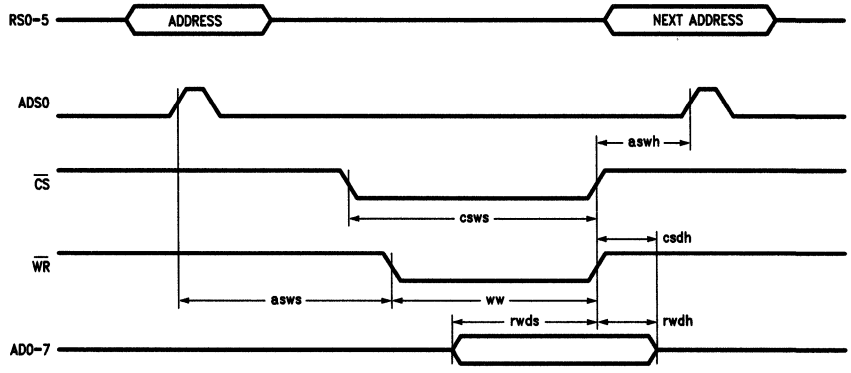
TL/F/5282–23

Symbol	Parameter	DP8466A-25/20		DP8466A-12		Units
		Min	Max	Min	Max	
rsdv	Register Select to Data Valid (ADS0 = 1) (Note 1)		150		200	ns
csdv	Chip Select to Data Valid		125		150	ns
rdv	Read Strobe to Data Valid		125		150	ns
rw	Read Strobe Width		10		10	μs
csdz	Chip Select to Data TRI-STATE (Note 2)	20	80	20	90	ns
rdz	Read Strobe for Data to TRI-STATE (Note 2)	20	80	20	90	ns
rsdh	Data Hold from Register Select Change (Note 1)	20		20		ns

Note 1: rsdv and rsdh timing assumes that ADS0 is true when RS0–5 changes.

Note 2: TRI-STATE note: These limits include the RC delay inherent in our test method. These signals typically turn off within 15 ns, enabling other devices to drive this line with no contention.

13.3 REGISTER WRITE (Latched Register Select: ADS0 Active)



TL/F/5282–24

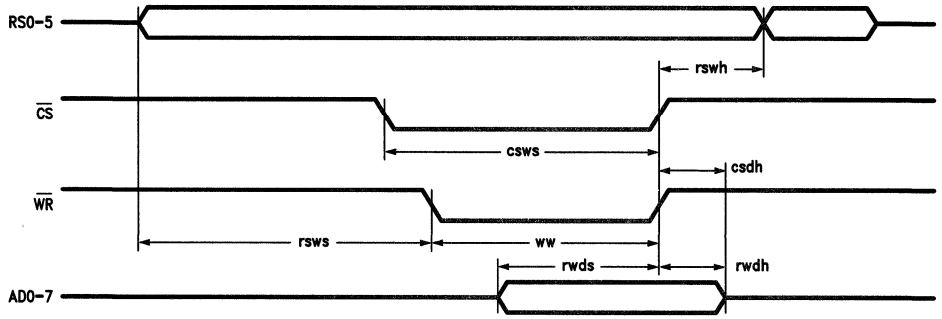
Symbol	Parameter	DP8466A-25/20		DP8466A-12		Units
		Min	Max	Min	Max	
asws	Address Strobe to Write Setup (Note 1)	15		20		ns
csws	Chip Select to Write Setup	50		70		ns
csdh	Chip Select Data Hold (Note 2)	7		10		ns
rwds	Register Write Data Setup	40		50		ns
rwdh	Register Write Data Hold (Note 2)	3		5		ns
ww	Write Strobe Width	50		70		ns
aswh	ADS0 Hold from Write (Note 1)	10		15		ns

Note 1: asws and aswh timing is referenced to the leading edge of ADS0 or the leading edge of valid address, whichever comes last.

Note 2: Minimum data hold time for a register write is referenced to $\overline{CS}$ or $\overline{WR}$, whichever goes inactive high first.

13.0 AC Electrical Characteristics & Timing Diagrams (Continued)

13.4 REGISTER WRITE (Non-Latched Register Select: ADS0 = 1)



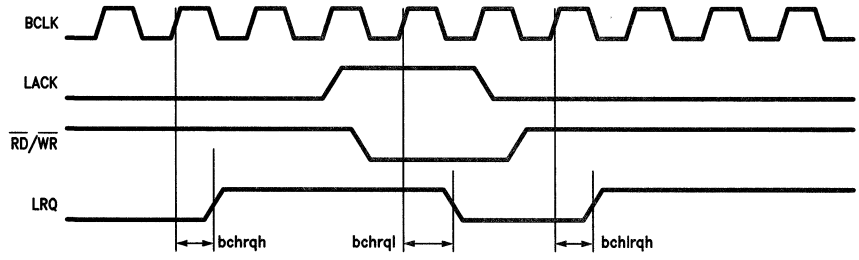
TL/F/5282–25

Symbol	Parameter	DP8466A-25/20		DP8466A-12		Units
		Min	Max	Min	Max	
rsws	Register Select to Write Setup (Note 1)	10		15		ns
csws	Chip Select to Write Setup	50		70		ns
csdh	Chip Select to Data Hold (Note 2)	7		10		ns
rwds	Register Write Data Setup	40		50		ns
rwdh	Register Write Data Hold (Note 2)	3		5		ns
ww	Write Strobe Width	50		70		ns
rswh	Register Select Hold from Write (Note 1)	15		20		ns

Note 1: rsws and rswh assume that ADS0 is true when RS0–5 changes.

Note 2: Minimum data hold time for a register write is referenced to $\overline{CS}$ or $\overline{WR}$, whichever goes inactive high first.

13.5 LRQ TIMING WITH EXTERNAL DMA



TL/F/5282–26

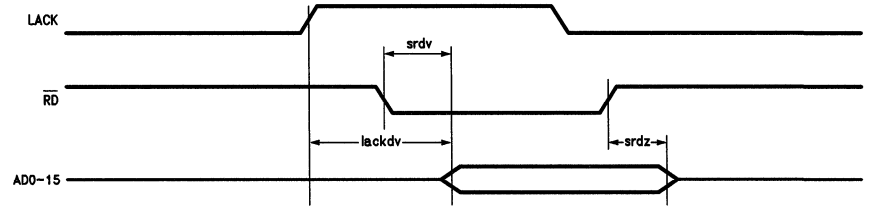
Symbol	Parameter	DP8466A-25/20		DP8466AN-12		Units
		Min	Max	Min	Max	
bchrqh	BCLK High to LRQ High		75		100	ns
bchrql	BCLK High to LRQ Low		75		100	ns

Note 1: The "ON" condition for the slave mode DMA, once the LRQ is active, is when both LACK and the $\overline{RD}$ or $\overline{WR}$ strobes are active. The LRQ is then removed after the next BCLK as shown. The "OFF" condition for the slave mode DMA is determined by the $\overline{RD}$ or $\overline{WR}$ strobe becoming inactive and the LRQ could be deasserted from the next BCLK rising edge. Lack does not play a role in determining the "OFF" condition.

Note 2: National recommends to use the same clock that generates the external $\overline{RD}$ & $\overline{WR}$ strobes for BCLK.

13.0 AC Electrical Characteristics & Timing Diagrams (Continued)

13.6A READING FIFO DATA IN DMA SLAVE MODE



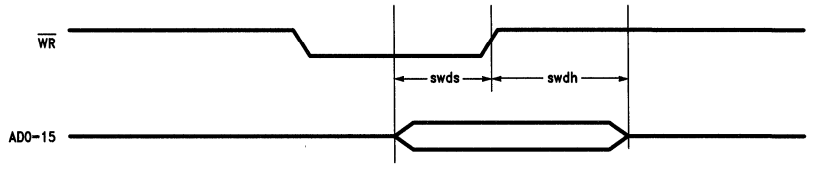
TL/F/5282–27

Symbol	Parameter	DP8466A-25/20		DP8466A-12		Units
		Min	Max	Min	Max	
lackdv	LACK to Data Valid		80		90	ns
srdv	Slave Read Strobe to Data Valid		60		70	ns
srdz	Slave Read Strobe to Data TRI-STATE (Note 3)	20	80	20	90	ns

Conditions: Disk read operation, DMA disabled, LRQ output true.

Note 3: TRI-STATE Note: These limits include the RC delay inherent in our test method. These signals typically turn off within 15 ns, enabling other devices to drive this with no contention.

13.6B WRITING FIFO DATA IN DMA SLAVE MODE

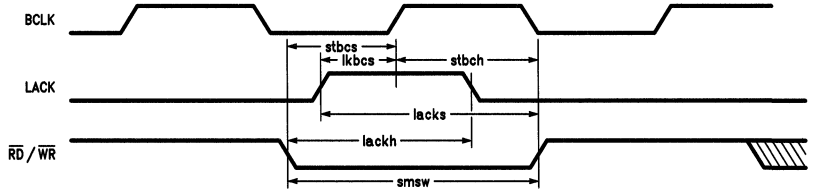


TL/F/5282–28

Symbol	Parameter	DP8466A-25/20		DP8466A-12		Units
		Min	Max	Min	Max	
swds	Slave Write Data Setup	5		10		ns
swdh	Slave Write Data Hold	20		28		ns

Conditions: Disk write operation, DMA disabled, LRQ output true.

13.7 ADDITIONAL SLAVE MODE DMA TIMING



TL/F/5282–78

Symbol	Parameter	DP8466A-25/20		DP8466A-12		Units
		Min	Max	Min	Max	
smsw	Slave Mode Strobe Width	40	(Note 2)	50	(Note 2)	ns
lacks	Lack to Strobe Setup	50		60		ns
lackh	Strobe to Lack Hold	10		15		ns
lkbcs	Lack to Bus Clock Setup	35		40		ns
stbch	Strobe from Bus Clock Hold	10		15		ns
stbcs	Strobe to Bus Clock Setup	20		25		ns

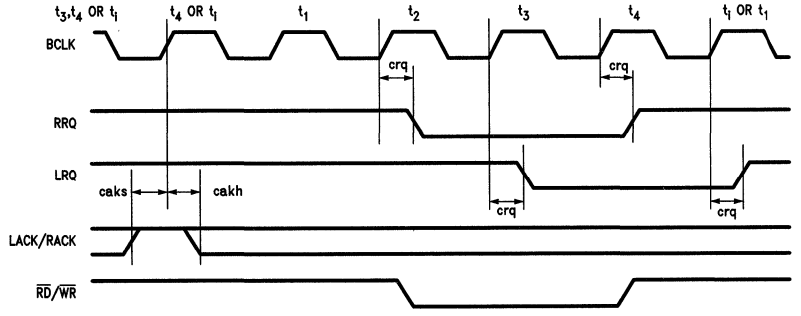
Conditions: Disk read or disk write operation, internal DMA disabled, and LRQ output active.

Note 1: The Read or Write Cycle begins when Lack and ($\overline{WR}$ or $\overline{RD}$) are true. From this point Lack must be held true for lackh and $\overline{WR}$ or $\overline{RD}$ must remain true for smsw.

Note 2: Disk Read or Write Byte Transfer Rate cannot exceed DMA Byte Transfer Rate. The inactive $\overline{RD}/\overline{WR}$ pulse width must be at least 2 BCLK cycles.

13.0 AC Electrical Characteristics & Timing Diagrams (Continued)

13.8 LOCAL AND REMOTE DMA ACKNOWLEDGE



TL/F/5282–29

Symbol	Parameter	DP8466A-25/20		DP8466A-12		Units
		Min	Max	Min	Max	
crq	Bus Clock to Request (Notes 5, 6)		85		100	ns
caks	Acknowledge Setup to Clock	20		25		ns
cakh	Bus Clock to Remote Status	10		15		ns

Note 1: The Local and Remote Acknowledges are sampled at the beginning of bus cycles t4 and t1.

Note 2: Local Acknowledge has internal priority over Remote Acknowledge.

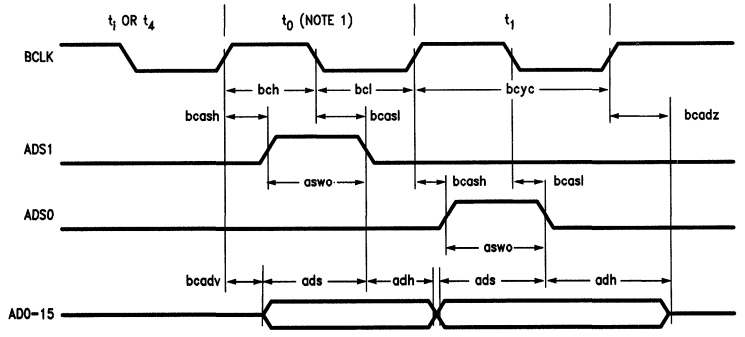
Note 3: Local and Remote Acknowledge are ignored if their respective Request output line is false.

Note 4: Above timing is for 16 bit address updates. For 32 bit Local address mode, cycle t0 occurs on the first transfer of an operation or when the lower 16 bits of the address rollover.

Note 5: crq is implied to be the same for both assertion and deassertion of LRQ or RRQ.

Note 6: LRQ will deassert on t2 for the final deassertion.

13.9 DMA ADDRESS GENERATION



TL/F/5282–30

Symbol	Parameter	DP8466A-25/20		DP8466A-12		Units
		Min	Max	Min	Max	
bcyc	Bus Clock Cycle Time (Notes 2, 3)	50	10,000	80	10,000	ns
bch	Bus Clock High Time (Note 3)	22.5	10,000	32	10,000	ns
bcl	Bus Clock Low Time (Note 3)	22.5	10,000	32	10,000	ns
bcash	Bus Clock to Address Strobe High		45		55	ns
bcasl	Bus Clock to Address Strobe Low		50		60	ns
aswo	Address Strobe Width Out	bch		bch		ns
bcadv	Bus Clock to Address Valid		60		70	ns
bcadz	Bus Clock to Address TRI-STATE (Note 4)	20	80	20	90	ns
ads	Address Setup to ADS0/1 Low	bch − 17		bch − 22		ns
adh	Address Hold from ADS0/1 Low	bcl − 10		bcl − 10		ns

Note 1: Cycle t0 occurs only on the first transfer of an operation or when the lower 16 bits of the address rolls over.

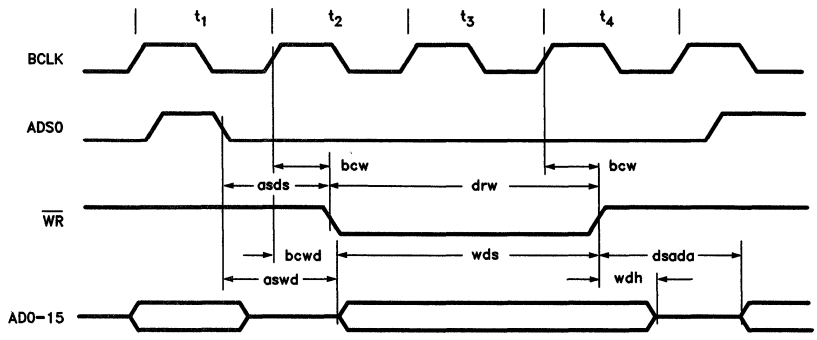
Note 2: The rate of bus clock must be high enough that data will be transferred to and from the FIFO faster than the data being transferred to and from the disk.

Note 3: For DP8466A-20, minimum bcyc = 60 ns minimum bch = bcl = 28 ns.

Note 4: TRI-STATE note: These limits include the RC delay inherent in our test method. These signals typically turn off within 15 ns, enabling other devices to drive this line with no contention.

13.0 AC Electrical Characteristics & Timing Diagrams (Continued)

13.10 DMA MEMORY WRITE



TL/F/5282–31

Symbol	Parameter	DP8466A-25/20		DP8466A-12		Units
		Min	Max	Min	Max	
bcw	Bus Clock to Write Strobe		50		60	ns
wds	Data Setup to $\overline{WR}$ High (Note 1)	2bcyc − 35		2bcyc − 45		ns
wdh	Data Hold from $\overline{WR}$ high (Note 1, 3)	8	50	8	60	ns
bcwd	Data Valid from t2 Clock (Note 1)		75		90	ns
asds	Address Strobe to Data Strobe (Note 2)		bcl + 10		bcl + 20	ns
aswd	Address Strobe to Write Data Valid		bcl + 40		bcl + 60	ns

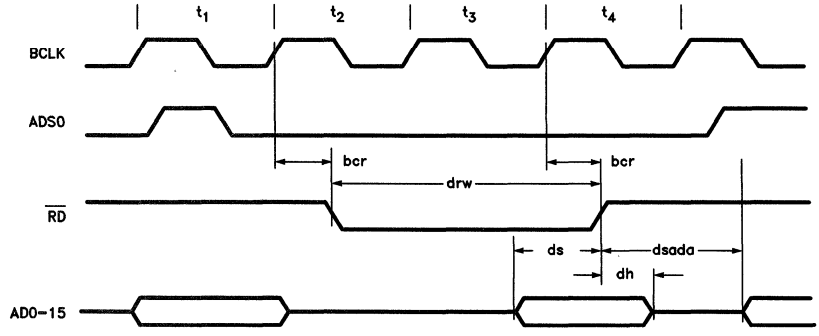
Conditions: DMA write, Local or Remote transfer, internal DMA.

Note 1: Data is enabled on AD0–15 only in local DMA transfers.

Note 2: Data strobe is either $\overline{RD}$ or $\overline{WR}$ out.

Note 3: TRI-STATE Note: These limits include the RC delay inherent in our test method.

13.11 DMA MEMORY READ



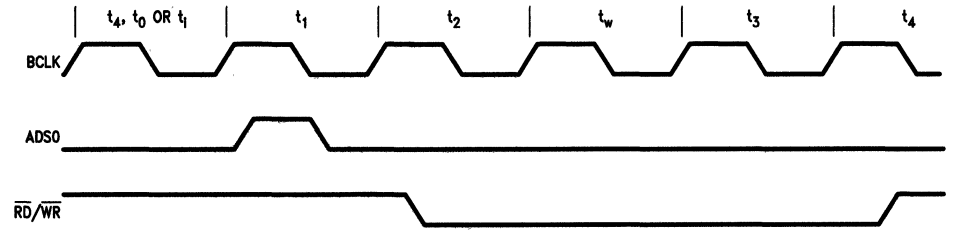
TL/F/5282–32

Symbol	Parameter	DP8466A-25/20		DP8466A-12		Units
		Min	Max	Min	Max	
bcr	Bus Clock to Read Strobe		50		60	ns
ds	Data Setup to Read Strobe High	30		35		ns
dh	Data Hold from Read Strobe High	0		0		ns
drw	DMA Data Strobe Width Out	2bcyc − 10		2bcyc − 15		ns
dsada	DMA Data Strobe to Address Bus Active	bcyc − 10		bcyc − 10		ns

Note 1: ds and dh timing are for Local transfers only.

13.0 AC Electrical Characteristics & Timing Diagrams (Continued)

13.12 DMA WITH INTERNAL WAIT STATES



TL/F/5282–33

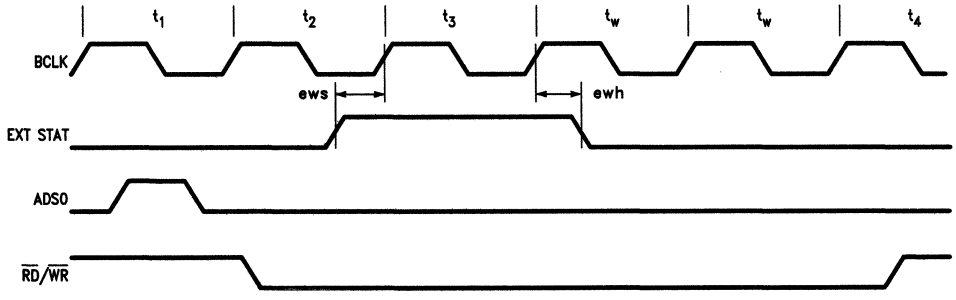
Conditions: Local or Remote DMA transfer, read or write, internal DMA.

Note 1: Addition of an internal wait state will lengthen $\overline{RD}/\overline{WR}$ strobes by an additional bus clock cycle.

Note 2: Internal wait states are enabled by setting the Slow Read/Write bits in the Local and Remote Transfer registers.

Note 3: If used, external wait states will be added between cycles t3 and t4.

13.13 DMA WITH EXTERNAL WAIT STATES



TL/F/5282–34

Symbol	Parameter	DP8466A-25/20		DP8466A-12		Units
		Min	Max	Min	Max	
ews	External Wait Setup to t3 Clock	15		20		ns
ewh	External Wait Hold after tw Clock	10		15		ns

Conditions: Read or write, internal DMA mode. Local or Remote transfer.

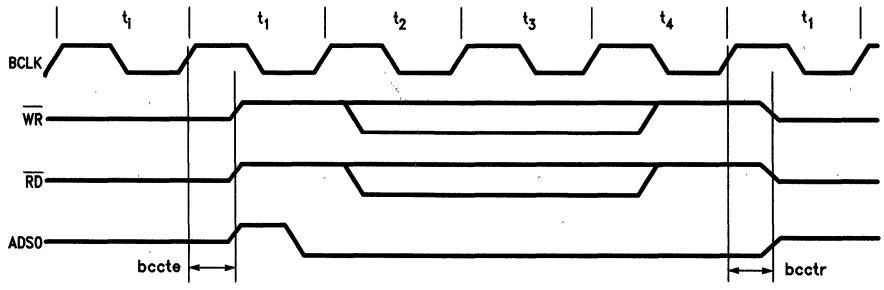
Note 1: Addition of external wait states will extend $\overline{RD}/\overline{WR}$ strobes by an integral number of bus clock cycles.

Note 2: If enabled, an internal wait state is added between cycles t2 and t3.

Note 3: EXT STAT is sampled upon entering states t3 and tw, and adds wait states one bus clock cycle later.

13.0 AC Electrical Characteristics & Timing Diagrams (Continued)

13.14 DMA CONTROL SIGNALS

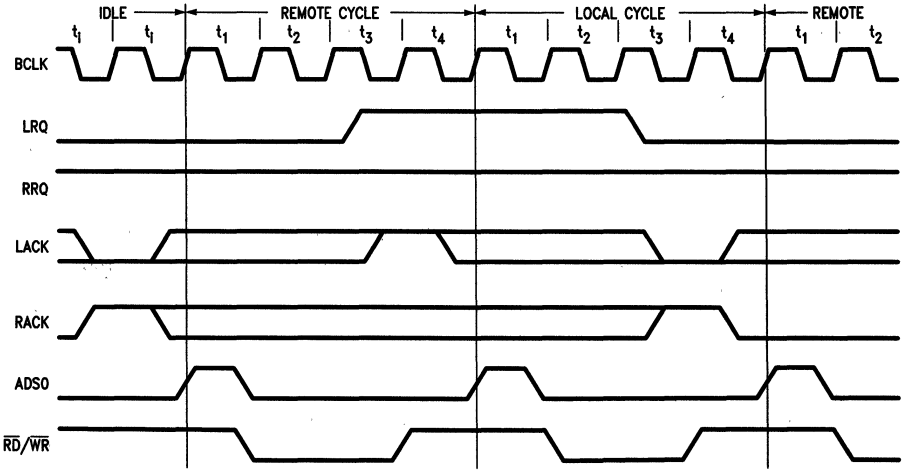


TL/F/5282–35

Symbol	Parameter	DP8466A-25/20		DP8466A-12		Units
		Min	Max	Min	Max	
bccte	Bus Clock to Control Enable (WR, RD, ADS0)		55		70	ns
bcctr	Bus Clock to Control Release (WR, RD, ADS0) (Note 1)		60		70	ns

Note 1: TRI-STATE note: These limits include the RC delay inherent in our test method. These signals typically turn off within 15 ns, enabling other devices to drive this line with no contention.

13.15 LOCAL AND REMOTE DMA INTERLEAVING



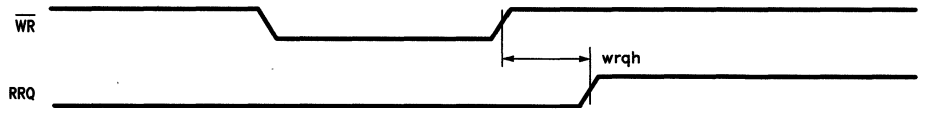
TL/F/5282–36

Note 1: Timing of the acknowledge pulses are used for illustration. Acknowledges need only to be set up with respect to t4 and t1 clock cycle.

Note 2: If both LACK and RACK are asserted with both LRQ and RRQ pending, a local DMA transfer will be performed.

13.0 AC Electrical Characteristics & Timing Diagrams (Continued)

13.16 RRQ ASSERTION AFTER WRITING TO OC REGISTER FOR REMOTE TRANSFER

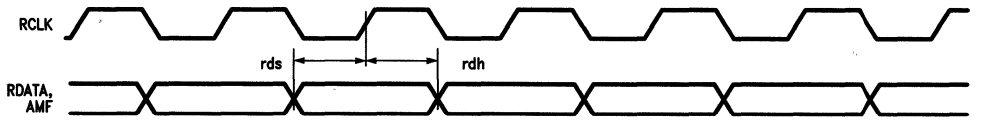


TL/F/5282–37

Symbol	Parameter	DP8466A-25/20		DP8466A-12		Units
		Min	Max	Min	Max	
wrqh	Write Strobe to Remote Request High		100		150	ns

Conditions: Non-tracking mode, writing "Start Remote Input/Output" to the Operation Command register.

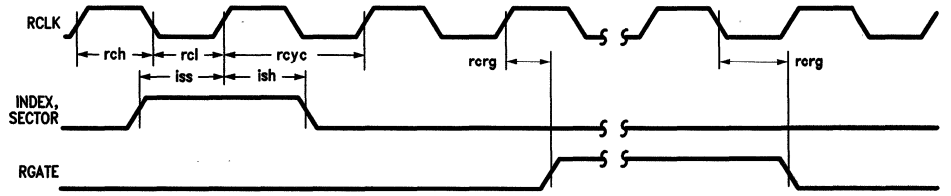
13.17 READ DATA TIMING



TL/F/5282–38

Symbol	Parameter	DP8466A-25/20		DP8466A-12		Units
		Min	Max	Min	Max	
rds	Read Data/AMF Setup to Read Clock	10		15		ns
rdh	Read Data/AMF Hold to Read Clock	10		15		ns

13.18 RGATE ASSERTION FROM INDEX OR SECTOR PULSE INPUT



TL/F/5282–39

Symbol	Parameter	DP8466A-25/20		DP8466A-12		Units
		Min	Max	Min	Max	
rcyc	Read Clock Cycle Time (Notes 2, 3)	39	10,000	80	10,000	ns
rch	Read Clock High Time (Note 2)	16	10,000	32	10,000	ns
rcl	Read Clock Low Time (Note 2)	16	10,000	32	10,000	ns
iss	Index/Sector Setup to Read Clock	10		15		ns
ish	Index/Sector Pulse Hold	10		15		ns
rcrg	Read Clock to Read Gate		55		70	ns

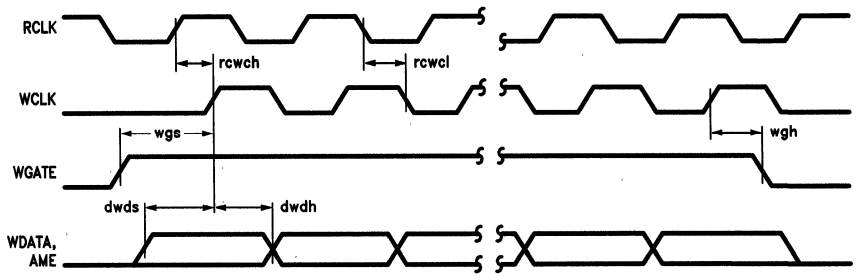
Note 1: INDEX/SECTOR low must meet iss/ish timing for proper INDEX/SECTOR pulse detection.

Note 2: For DP8466A-20, minimum rcyc = 50 ns, minimum rch and rcl = 20 ns.

Note 3: For DP8466A-25, this parameter is not tested directly, but is guaranteed through correlation.

13.0 AC Electrical Characteristics & Timing Diagrams (Continued)

13.19 WRITE DATA TIMING FOR NRZ TYPE DATA

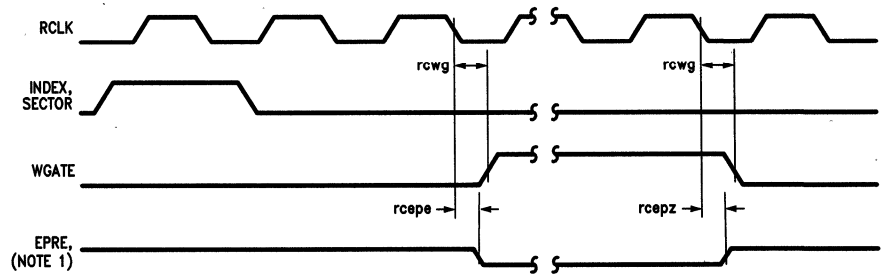


TL/F/5282–40

Symbol	Parameter	DP8466A-25/20		DP8466A-12		Units
		Min	Max	Min	Max	
rcwch	Read Clock to Write Clock High Delay		30		40	ns
rcwcl	Read Clock to Write Clock Low Delay		30		40	ns
rcwcs	Absolute Value of (rcwcl — rcwch)		6		7	ns
dwds	Drive Write Data Setup to Write Clock	rcl − 10		rcl − 15		ns
dwdh	Drive Write Data Hold to Write Clock	rch − 5		rch − 8		ns
wgs	Write Gate Setup to Write Clock	rcl − 10		rcl − 15		ns
wgh	Write Gate Hold to Write Clock	rch		rch		ns

Note 1: rcl and rch are described in Timing Diagram 13.18.

13.20 WGATE ASSERTION FROM INDEX OR SECTOR PULSE INPUT



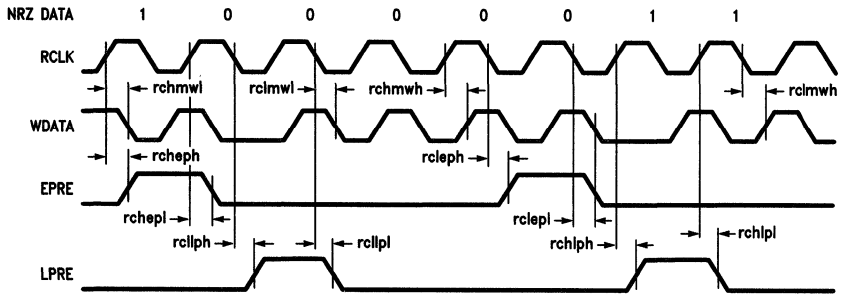
TL/F/5282–41

Symbol	Parameter	DP8466A-25/20		DP8466A-12		Units
		Min	Max	Min	Max	
rcwg	Read Clock to Write Gate		40		50	ns
rcepe	Read Clock to Early Precomp Enabled		50		60	ns
rcepz	Read Clock to Early Precomp TRI-STATE		50		60	ns

Note 1: Early Precompensation (EPRE) is used as an output only when writing MFM data.

13.0 AC Electrical Characteristics & Timing Diagrams (Continued)

13.21 WRITE DATA TIMING FOR MFM TYPE DATA



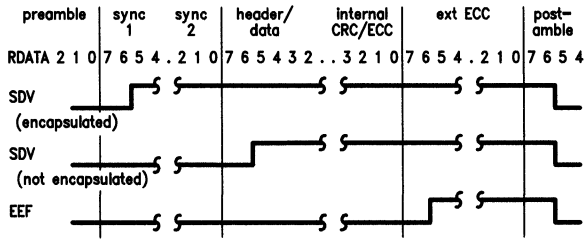
TL/F/5282–42

Symbol	Parameter	DP8466A-25/20		DP8466A-12		Units
		Min	Max	Min	Max	
rchmwh	RCLK High to MFM WDATA High		40		50	ns
rchmwl	RCLK High to MFM WDATA Low		40		50	ns
rclmwh	RCLK Low to MFM WDATA High		40		50	ns
rclmwl	RCLK Low to MFM WDATA Low		40		50	ns
rcheph	RCLK High to EPRE High		40		50	ns
rchepl	RCLK High to EPRE Low		40		50	ns
rcleph	RCLK Low to EPRE High		40		50	ns
rclepl	RCLK Low to EPRE Low		40		50	ns
rchlph	RCLK High to LPRE High		40		50	ns
rchlpl	RCLK High to LPRE Low		40		50	ns
rcllph	RCLK Low to LPRE High		40		50	ns
rcllpl	RCLK Low to LPRE Low		40		50	ns

13.0 AC Electrical Characteristics & Timing Diagrams (Continued)

13.22 POSITIONAL TIMING FOR SDV AND EEF

Read operation (Compare Header, Read Header, Compare Data or Read Data)

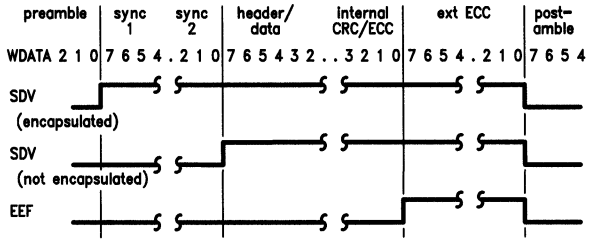


TL/F/5282–43

Note 1: Data should be delayed 2 bit times before entering external ECC circuitry in order for it to properly align correctly with SDC and EEF.

Note 2: Encapsulation is controlled by the HEN and DEN bits in the EC register, and causes the sync patterns to be included in the CRC/ECC calculation.

Write operation (Write Header, Write Data or Format Track)

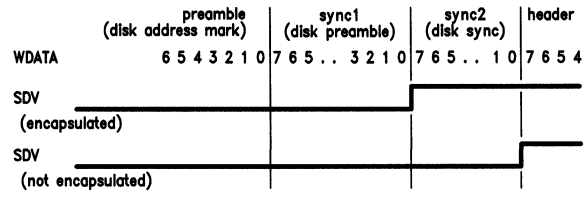


TL/F/5282–44

Note 1: Write operation shown is for NRZ data. For MFM encoding, Write data is delayed two bit times relative to NRZ data.

Note 2: Encapsulation is controlled by the HEN and DEN bits in the EC register, and causes the sync patterns to be included in CRC/ECC calculation.

Write header operation (Start with Address Mark)



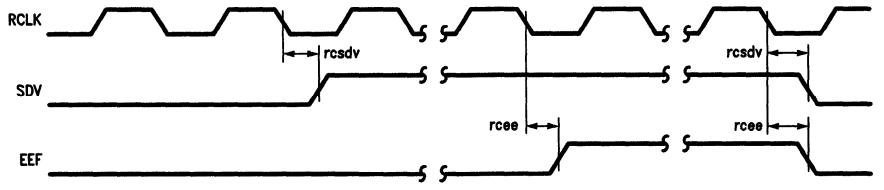
TL/F/5282–45

Note 1: Field names within parenthesis are the names of the fields on disk.

Note 2: Encapsulation is controlled by the HEN bit in the EC register, and causes the sync patterns to be included in CRC/ECC calculation.

13.0 AC Electrical Characteristics & Timing Diagrams (Continued)

13.23 FIELD ENVELOPE TIMING



TL/F/5282–46

Symbol	Parameter	DP8466A-25/20		DP8466A-12		Units
		Min	Max	Min	Max	
rcsdv	Read Clock to Serial Data Valid		35		50	ns
rcee	Read Clock to External ECC		35		50	ns

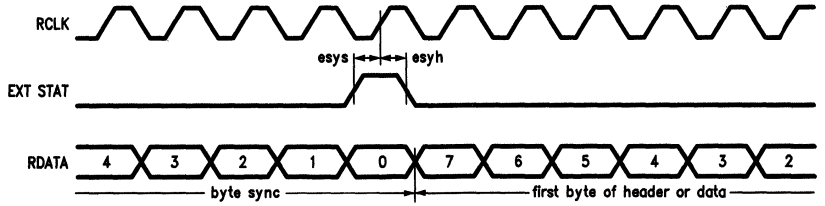
Note 1: SDV is asserted after sync fields, and is deasserted at the start of the postamble field. If sync field encapsulation is enabled, SDV is asserted at the start of the sync fields.

Note 2: EEF is asserted at the start of the external ECC field, and is deasserted at the start of the postamble field.

Note 3: When the DDC is receiving data from the disk, the SDV and EEF are delayed by two bit times from incoming read data due to internal delays.

Note 4: If the external ECC count is set to zero, no EEF output will be generated.

13.24 EXTERNAL STATUS TIMING WHEN USING EXTERNAL BYTE SYNC



TL/F/5282–47

Symbol	Parameter	DP8466A-25/20		DP8466A-12		Units
		Min	Max	Min	Max	
esys	External Byte Sync Setup to Rising Edge of Bit Clock 0 of Byte Sync	15		20		ns
esyh	External Byte Sync Hold to Rising Edge of Bit Clock 0 of Byte Sync	10		15		ns

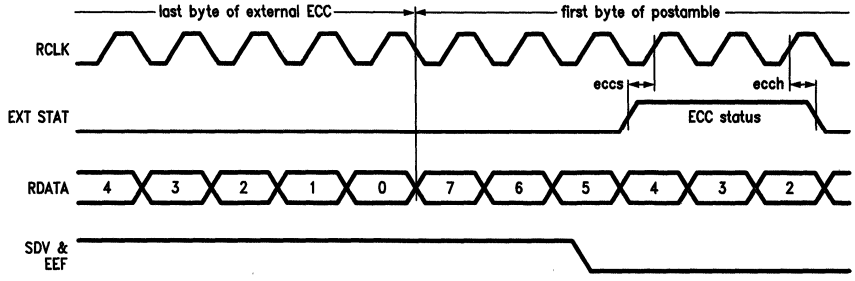
Note 1: The external sync feature can only be used if the Enable External Wait states (EEW) bit of the Remote Transfer (RT) register is not set.

Note 2: External circuitry is needed to feed the DDC with NRZ zeros until the external sync signal has been generated to prevent the DDC from trying to detect sync.

Note 3: If External Sync and External Wait states are not being used, the EXT. STAT. pin must be false during preamble and sync fields.

13.0 AC Electrical Characteristics & Timing Diagrams (Continued)

13.25 EXTERNAL STATUS TIMING WHEN USED FOR EXTERNAL ECC



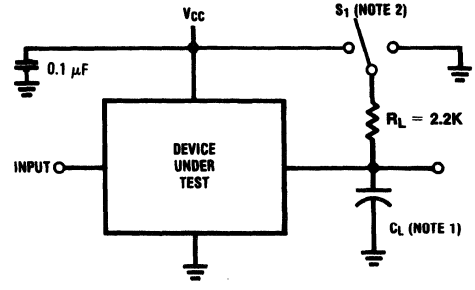
TL/F/5282–48

Symbol	Parameter	DP8466A-25/20		DP8466A-12		Units
		Min	Max	Min	Max	
eccs	External ECC Status Setup to Rising Edge of Bit Clock 4 of Postamble	15		20		ns
ecch	External ECC Status Hold to Rising Edge of Bit Clock 2 of Postamble	10		15		ns

Note 1: The external ECC error detection feature can only be used if the Enable External Wait states (EEW) bit of the Remote Transfer register (RT) is zero.

14.0 AC Timing Test Conditions

Input Pulse Levels	GND to 3.0V
Input Rise and Fall Times	5 ns
Input and Output Reference Levels	1.3V
TRI-STATE Reference Levels	Float (ΔV) ± 0.5V
Output Load (See Figure 27)	



TL/F/5282–77

FIGURE 27

Note 1: C_L = 50 pF, includes scope and jig capacitance

Note 2: S1 = Open for Push Pull Outputs

S1 = V_{CC} for High Impedance to active low and active low to High Impedance measurements.

S1 = GND for High Impedance to active high and active high to High Impedance measurements.

Capacitance (T_A = 25°C, f = 1MHz)

Parameter	Description	Typ	Max	Unit
C_{IN}	Input Capacitance	7	12	pF
C_{OUT}	Output Capacitance	7	12	pF

Note: This parameter is sampled and not 100% tested.

15.0 Miscellaneous Timing Information

15.1 STATUS REGISTER TIMING

HEADER FAULT: This bit is set at the start of the Header Postamble field of a header with a CRC/ECC error. It is reset at the start of the Header Postamble of the header requested, or upon receipt of a new disk command. No interrupt is generated.

NEXT DISK COMMAND: This bit is set at the start of the Header Postamble of the last sector of an operation, and is reset upon loading the Drive Command register. No interrupt is generated.

HEADER MATCH COMPLETED: This bit is set at the start of the Header Postamble field of the header of interest. This bit is reset when the DDC begins the next header operation. An interrupt is generated if enabled.

LOCAL REQUEST: This bit has the same timing as the Local Request pin. When the FIFO requires servicing, this bit is set. When service is no longer required, this bit is cleared. No interrupt is generated.

REMOTE COMMAND BUSY: In the tracking mode, this bit is set 3–5 RCLK's after receipt of a drive command. In the non-tracking mode, this bit is set when either a Start Remote Input or Start Remote Output command is received in the Operation Command register. This bit is reset and interrupt is generated upon completion of the initiating operation.

LOCAL COMMAND BUSY: This bit is set 3–5 RCLK's after receipt of a drive command which requires the use of the local channel. It is reset after the last transfer of the local channel if in the non-tracking mode or writing the disk, or after the last transfer of the remote channel if in the tracking mode and reading disk. Interrupt is generated upon completion of the initiating operation.

CORRECTION CYCLE ACTIVE: This bit is set upon receipt of the Start Correction Cycle in the Operation Command register, and is reset at the end of the correction operation. An interrupt is generated at the end of the correction cycle.

ERROR DETECTED: This bit is a logical OR function of all the bits in the Error register. An interrupt is generated when an error is detected.

15.2 ERROR REGISTER TIMING

HFASM ERROR: If while in the HFASM mode the sector address matches and another header byte does not, this bit will be set at the start of the Header Postamble field.

DATA FIELD ERROR: If the Data field contains a CRC/ECC error, this bit will be set at the start of the Data Postamble field.

SECTOR NOT FOUND: If the header of the desired sector is not located before two index pulses are received, this bit will be set upon receipt of the second index pulse.

SECTOR OVERRUN: If an index or sector pulse is detected while reading the Header or Data field, or while writing and not in the Gap field, this bit will be set upon receipt of the sector/index pulse.

NO DATA SYNC: If an index or sector pulse is received before data sync is detected, this bit is set upon receipt of the sector/index pulse. If there is a data sync error after the first sync byte has been detected, this bit will be set during the byte following the byte in error.

FIFO DATA LOST: If a transfer between the disk and FIFO causes the FIFO to underrun or overrun, this bit will be set within the next byte time creating a write splice if write gate was on. This is reflected as an ECC error and can be removed if sector is rewritten.

CORRECTION FAILED: This bit is set at the end of the correction cycle if the error is non-correctable.

LATE INTERLOCK: This bit is set at the start of Data Postamble field for Read operations and at the end of the postamble field for non-format Write operations. While formatting, this bit is set at the end of the Gap field.

15.3 GENERAL TIMING FOR READ GATE

Whenever the DDC is reading, comparing, or in some cases, ignoring information, RGATE is asserted. The use of RGATE can be separated into three groups: Header search (soft sectored mode), header examination, and data examination.

SEARCHING FOR HEADERS

When the DDC is searching for a header in the soft-sectored mode, RGATE is asserted in a somewhat random location in the format. After being asserted, if the DDC does not recognize the address mark pattern within eight bit times of detecting a one, RGATE will be de-asserted in 18½ RCLK's. RGATE will then remain low for 17½ RCLK's before another search attempt is made.

In modes where the DDC starts a Read, Compare or Ignore Header operation at an index or sector pulse, RGATE will be asserted 3–4 RCLK cycles from detection of the index or sector pulse.

DATA OPERATIONS

After the header operation has completed, RGATE will be removed two bits after the start of the Header Postamble. If a Read or Check Data operation is to follow, RGATE will be reasserted 11½ bits after the Header Postamble.

At the end of the Data field, RGATE will be removed two bits into the start of the Data Postamble.

15.4 WRITE GATE TIMING

Whenever the DDC is writing information, WGATE is asserted. WGATE can be separated into three uses: Writing header, writing data or track formatting.

WRITING HEADERS

When the DDC writes the header, the write operation does not begin until the receipt of an index or sector pulse. After the pulse is detected, WGATE will be asserted 2½–3½ RCLKs from the detection of the pulse. WGATE will stay true until the end of the Header Postamble, unless the Data field is to be written. If the Data field is to be written, WGATE will not be de-asserted between the Header and Data fields.

WRITING DATA

After a header operation has properly completed, WGATE will be asserted 3 bit times into the Data Preamble. The WGATE will remain active until the end of the Data Postamble. Because of internal delays within the DDC, the Write Data operation is delayed three bit times from the header patterns.

15.0 Miscellaneous Timing Information (Continued)

FORMAT TRACK

In a format track operation, WGATE is asserted 2½–3½ RCLK's from the detection of the index pulse. WGATE will remain active until the next index pulse is detected, and will then be removed.

Note: Detection of an index or sector pulse is defined as the rising edge of the RCLK where index/sector input has met the setup time.

15.5 NORMAL INTERRUPTS

Interrupts are generated by the DDC for a variety of reasons, but they all fall into one of three categories: Either they signal normal completion, a synchronization point, or an error condition. If an interrupt is generated because of an error, the interrupt will have timing as described in the Error register timing section.

The Header Operation Complete interrupt is used for synchronization, and is enabled with the Enable Header Interrupt bit of the Operation Command register. This interrupt will occur when the DDC finishes the header operation, and starts the data operation. For Read, Compare, Write, or Ignore Header operations, the interrupt will be generated at the start of the Header Postamble field.

The normal Operation Complete interrupt is dependent on the operation being performed. If the operation is to Check Data, the interrupt is generated at the start of the Data Postamble field. For Write Data operations, an interrupt will be generated at the end of the Data Postamble. When the DDC is formatting, the interrupt will be delayed by the length of the Header Preamble after the format has finished. The fourth event is further defined by the DMA mode used. For all local channel operations except for tracking mode disk read, the interrupt will be generated during the last transfer of data from the FIFO. In the configuration, tracking mode disk read, the interrupt will be delayed until the last transfer is made by the remote DMA. For all non-tracking remote DMA operations, the interrupt will be generated during the last transfer of the remote DMA.

When a correction operation is being performed, an interrupt is generated at the end of the correction cycle, regardless of the outcome.

15.6 DERATING FACTOR

Output timings are measured with a purely capacitive load for 50 pF. The following correction factor can be used for other loads:

DP8466A-25/20	$C_L \geq 50$ pF:	+.13 ns/pF (ADS0, ADS1) +.20 ns/pF (all other outputs)
DP8466A-12	$C_L \geq 50$ pF:	+.18 ns/pF (ADS0, ADS1) +.25 ns/pF (all other outputs)

16.0 DP8466A Functional Status

Introduction

This section is intended to provide some relevant information on the functional status of the Disk Data Controller, DP8466A. Several problems have been identified in the DP8466 from the numerous beta sites. All of these were investigated and many were rectified resulting in the DP8466A as the production version. However there are still a few shortcomings which are outlined below for reference:

1.0 Correction Cycle Failure

If a correction cycle is attempted when an ECC/CRC error occurs in a multisectored disk operation with the sync word being encapsulated, then it will always fail because the ECC shift register gets preset. In order to ensure proper correction, a single sector retry must be attempted on the erroneous sector before correction cycle is initiated.

2.0 Error-Correction Handling Feature in Tracking Mode (Remote transfer of data conditional on Data ECC Error)

During tracking-mode read data operation, data will be transferred to local memory and then to a remote port. The DMA should prevent a remote transfer of the data until the DDC has checked for a data ECC error. Hence if correction is to be attempted, then it can be done in the local memory and then remote transfer can continue. However, the bad data will be sent to remote system memory without regard to its integrity and hence it's the responsibility of the user to correct the data in his system memory or send the correct data block again.

3.0 Odd Byte Remote DMA Transfer

Odd byte remote transfers are not allowed by the DMA mode. Therefore if only one transfer is desired to the remote port, it cannot be done. The only way to overcome this problem is to do a transfer of two bytes and ignore the second byte by reloading the remote data byte counter, etc.

4.0 Parameter RAM Registers Losing Contents

If at anytime the Read Clock input sees a glitch, then there is a good probability for some of the registers in the parameter RAM to lose their contents, e.g., ID sync #1, ID sync #2 etc. Whenever the Read Clock goes below the minimum specifications of 'rch' (read clock high time) and 'rcl' (read clock low time), it is considered as glitching the Read Clock. Hence it is the users responsibility to ensure that there are no glitches in the Read Clock input. In the future version, redesign will be attempted to decrease or totally remove the susceptibility of the chip to glitches on the Read Clk.

5.0 Remote DMA Interrupt Handling

In the non-tracking-mode remote DMA operation the operation complete interrupt could be held off or remain asserted despite servicing attempts whenever it happens while the disk header search is being attempted simultaneously. This will have to be taken care of in software. More details of this situation are provided in Chapter 2.

6.0 AME/AMF Handshake for ESDI (Softsectored Drives)

The DDC does not incorporate the handshake for ESDIsoft sectored disk operation. The DDC generates the AME signal only during the format operation and not during the read/write operation, when in the hardsectored, NRZ data mode. In the ESDI spec. Address Mark Found, AMF, responds only after AME is asserted. If AME is not asserted then AMF from the drive will not occur and the beginning of the sector will not be determined. The external logic and software methods needed to implement this handshake protocol is discussed in the design guide application note (AN413), in the MASS STORAGE data book.

7.0 Post Index/Sector Gap Field

The DDC has no defined field to implement the post index or post sector pulse gap. This can however be still imple-

16.0 DP8466A Functional Status
(Continued)

mented using a combination of software manipulation and external circuitry, as outlined in the design guide application note (AN413) in the MASS STORAGE data book.

8.0 Write Clock with Respect to Write Gate

In the DDC, Write Clock is generated 0.5 bit times after Write Gate is asserted. However in case of the SMD and ESDI drives they expect write clock to be active 250 ns (worst case) before write gate is asserted. This would have to be accomplished using external circuitry if desired.

17.0 Some Helpful Hints, When Designing a Disk Controller Subsystem with National Semiconductor's DP8466A, Disk Data Controller (DDC)

The following section provides some useful hints/application information for designing with the DP8466A. The suggestions given in this document are the results of situations encountered while debugging the designs at National and also from the feedback provided by the numerous beta site designs during their debug stages. This is an unending list and users are welcome to add their experiences, for they may save someone else a lot of trouble in the process. It should be understood that some of the situations outlined may be dependent on that particular system design approach and may not necessarily present itself in a different system environment. Hence National assumes no guarantees regarding these situations. A lot of the suggestions are explanations of inherent operational rules that may not be very evident in the chips documentation. For a detailed technical reference for design purposes, users are recommended to consult the DP8466 design guide in the MASS STORAGE handbook, while the DATA SHEET gives the features and timing specifications of the chip.

Sync #1 and Sync #2 Pattern Restrictions

When the DDC is in the read mode, i.e., read/compare/ignore header/or read/ check data, then it starts out looking for the sync byte. The data separator usually sends out zeroes when it is attempting to lock and when it has, it sends out the data coming off the disk. Hence the DDC is looking for the first non-zero bit to initiate sync byte comparison. If the DDC is programmed in the soft sectored mode then it basically attempts to do a compare for eight clocks before it asserts the abort address mark function internally and recycles Read Gate. In the hard sectored mode it will essentially be waiting for the sync match forever, till two revolutions of the disk, after which it gives a SNF error. Therefore it is not advisable to use a pattern of zeroes for the sync #1 or sync #2 bytes as that would result in an immediate sync byte alignment when read gate is asserted, as the serializer has been cleared to all zeroes. However, when writing information on the disk the sync #1 and sync #2 fields could be used to write a pattern of zeroes. This would probably be the case when some software manipulation is being attempted with the various fields of the DDC to implement some additional function like the post index gap, etc. *Hence, a pattern of zeroes is not recommended for sync #1 and sync #2 fields during a READ operation.*

Most Significant Bit of Sync Byte

When the sync byte is included in the CRC/ECC calculation, i.e., the encapsulated mode, controlled by the ECC Control Register, then it is **mandatory** that the most significant bit of the first sync byte be a 1. Hence, the most significant bit of the sync byte must be a 1.

Proper Sequence for Reset and Renable

The proper reset sequence for the chip consists of holding the RESET line active or the reset bit set in the OC register for 32 RCLKS and 4 BCLKS. Then this is deactivated and the RENABLE operation is initiated with a 01 in the DC register. It is possible, although not necessary for the renable operation to take as long as 260 RCLKS after which the operation complete interrupt would be generated. In case the status register is polled to detect operation completion, then the status register should be polled for the NDC bit set. Once set, it should be read after 30 RCLKS. If the NDC bit is still set then it signals the proper completion of the RENABLE operation.

Read/Write Registers

In the DDC some of the registers are defined as read only, while some are defined as write only. Care should be taken that read only registers should not be written to and write only registers should not be read from.

Write Header—Write Data Operation Variations

For the WRITE HEADER-WRITE DATA operation the DDC will fetch the data for the ID and DATA fields from the on-chip parameter RAM if the FMT bit is set in the DC register, (this also constitutes a format operation). The same scenerio with the FTF bit set in the DISK FORMAT register will fetch the ID and DATA information from the local buffer memory by the local DMA channel, and constitutes a full format operation. If however, the FMT bit is reset then the information is fetched from the local buffer memory by the local DMA channel and the operation is a regular one.

Status Reads on Interrupts

The STATUS register is read when an interrupt occurs to determine its cause and also serves to reset the interrupt line. However, if the status is read before 16 RCLKS after the interrupt, then the interrupt line will not be reset by the status register read. If the STATUS register is being constantly polled in the software, then it must not go faster then once in 16 RCLKS.

LCB Bit Behaviour

Whenever a disk operation is initiated, the LCB bit in the status register is set until the operation is complete. However if the operation is truncated due to any error condition in the header or the data fields, the LCB bit will remain set in the status register.

Resetting the DDC

After a normal reset of the DDC, none of the registers in the parameter RAM are affected. The STATUS and ERROR registers are cleared. The internal counters are reset and the FIFO pointers point to the beginning of the FIFO. There are, however, two other conditions that need to be taken into account.

If the DDC is reset while it is reading, writing, or formatting, the entire format RAM is potentially corrupted. For this reason, hex addresses 14–33, 38, 39, 3B–3F need to be reloaded after such a reset.

17.0 Some Helpful Hints, When Designing a Disk Controller Subsystem with National Semiconductor's DP8466A, Disk Data Controller (DDC) (Continued)

There is a certain relationship between BCLK and $\overline{WR}$ that can cause the DDC to be fooled into thinking it is in a DMA cycle for 2 BCLKs following a software reset (setting the reset bit in the Operation Command Register). This is also true if BCLK and $\overline{RST}$ have a certain relationship. The implications of this are that ADS0 will be asserted on the first BCLK and $\overline{WR}$ will be asserted on the second. Both will be cleared by the third. Internally, the following registers may change:

HA	Register
0F	Header Byte Count
13	NSO Counter
1A	Remote Data Byte Count (L)
1B	Remote Data Byte Count (H)
1C	DMA Address Byte 0
1D	DMA Address Byte 1
1E	DMA Address Byte 2
1F	DMA Address Byte 3
38	Sector Byte Count (L)
39	Sector Byte Count (H)

For this reason all of the above registers should be reloaded after a reset if it is not known that the timing relationship is such that the problem will not occur. Additionally, if $\overline{CS}$ is active at the time of the false $\overline{WR}$, then the register selected by the RS0–5 signals will be altered. In order to avoid a problem here, putting a zero on these lines will cause the write to go to the Status Register which cannot be written to, so no destructive write will occur.

Queing Disk Commands

After header match is successfully accomplished in a disk operation, also indicated by the header match complete interrupt if enabled, the NDC bit is set in the status register, indicating that the DDC is ready to accept the next disk command. Hence the next disk operation could be queued by a load of the DC register, however, it should be noted that the operation will not commence until the previous one has completed. Hence, care should be taken that registers being used while the data segment of the previous operation is in progress should not be changed, e.g., the registers associated with the DMA etc.

Assertion/Deassertion of LRQ/RRQ

In the burst DMA mode the request (LRQ/RRQ) is asserted when the set threshold is reached in the FIFO/LOCAL BUFFER MEMORY, and deasserted after the set burst is transferred even if the threshold has been reached again for the next transfer. The request is then reasserted for the next transfer.

Causes of Interrupts

There is only a single interrupt line on the DDC. There may be more than one source for the interrupt at times. It is hence recommended that every time an interrupt is serviced all the possibilities be checked to safeguard against more than one completion condition occurring at the same time.

HMC Bit in the Status Register

The HMC bit in the STATUS register is functional even if the header match complete interrupt is not enabled in the OC register. In a similar context it should be noted that even if the interrupts are not enabled in the OC register, the interrupt condition is generated internally when it happens. This EN bit in the OC register essentially controls the physical availability of the interrupt on the pin to the outside world.

Correction Cycle Initiation Sequence

When a CRC/ECC error occurs in a disk operation, the DDC has to be reset before a correction cycle can be attempted. On completion of the correction cycle the chip needs to be reset only if the correction cycle failed and hence an error condition resulted. In general, the DDC should be reset following any operation terminating in an error condition.

DFE (Data Field Error) Exceptions

Usually the Data Field Error condition in the DDC is terminal and the operation is aborted with an interrupt. However there is one exception to the rule. This is if the operation is a multisector check data operation in the interlock mode, then the DFE error will set the bit in the ERROR register but will not generate an interrupt and hence will not terminate the operation.

Read Header—Check Data Operation Exception

Normally the operation complete interrupt comes at the end of the data field for the disk operation. It is usually signified by the LCB bit reset in the STATUS register in case of non-tracking mode or by the LCB and RCB bit reset in the tracking mode. However there is one exception to the rule. In the case of a read header-check data operation, because local DMA transfers only the header and there is no DMA activity for the data field, the LCB bit is reset just after the header and the operation complete interrupt is generated. There is no interrupt at the end of the check data unless there is an ECC error in which case the operation is terminated with the error signalled through the STATUS and ERROR registers.

Header Fault Exceptional Behaviour

The HF (header fault) bit in the STATUS register is a passive error condition bit which is set if there is a CRC/ECC error in the header. This does not generate an interrupt nor terminate the operation normally. In a normal operation this bit is set if there is a header fault in the header, while searching for a sector and its gets reset, only if there is no CRC/ECC error in the header of the sector being sought. It should be noted that this behaviour is exhibited even when the DDC is searching for headers. However there is one exception to the rule. In case of a Read Header operation, if there is a CRC/ECC error in the header, then an interrupt is generated, the operation is terminated and the STATUS register will have the ED bit and the HF bit set while the ERROR register will read zeroes.

SC and NSO Counter Updates

The Number of Sector operations counter (NSO) in the DDC should be handled with care. Although addressed as one register, internally it is downloaded into two separate counters; one for the disk side and the other for the DMA logic. Whenever a read is done of the NSO counter, the value read back is the contents of the disk side NSO counter. The disk side NSO counter is decremented just after the header match complete interrupt, while the DMA side NSO counter is decremented while the local DMA channel is transferring

17.0 Some Helpful Hints, When Designing a Disk Controller Subsystem with National Semiconductor's DP8466A, Disk Data Controller (DDC) (Continued)

the last byte of the data field. If the SC and NSO counters have to be read/written by the microprocessor for some reason, care should be taken that they are not read/written when the DDC is accessing them internally, otherwise they might be zeroed. So it is recommended that they be read or updated about 1 μs after the HMC bit is set in the STATUS register. By the same token, this applies to other registers like the Remote Data Byte Count registers, Sector Byte Count registers, and DMA address registers. Also if the NSO counter has to be updated before the operation is completed to fool the DDC to go on some more without reloading the command then certain precautions need to be observed. Firstly the NSO register can be written to only after the DMA side NSO has been decremented. Secondly, the update cannot be done after the NSO on the DMA side has decremented to a 1, in other words the update cannot be done after the second to last sector, and hence has to be done at the latest before two sectors remain for the completion of the current multisector operation.

LT and RT Register Loading Restrictions

It is mandatory that the LT (RT) register must be loaded before the Sector Byte Count (Remote Data Byte) register pairs for any of the following situations.

a) If any internal DMA is being used or **b)** if the Remote Data Byte Count registers are going to be read by the processor, or **c)** if one needs to rewrite the LT or RT registers at anytime, (like when one wants to shift from tracking to non-tracking mode etc).

DMA Burst Mode Behaviour

One of the features of the DDC is that it can be programmed to do DMA transfers in the burst mode. The size of the burst is selectable through the LT & RT registers. Internally the burst value is downloaded to the burst counter which will reload itself only when the terminal count is reached or if the DDC is reset. The size of the DMA transfer is the length of the sector in case of a single sector transfer while in case of a multisector operation the DDC looks at it like one big transfer of length equal to the sector length times the number of sectors requested for the operation. This value is divided by the burst length which determines the number of bursts in the total transfer. If the total transfer length were not an even multiple of the burst length, then the very last burst would be less than the burst length selected. Control logic in the FIFO ensures that the remainder bytes are transferred even though it is less than the burst threshold. However, the internal burst counter remains at that lesser number and does not get reinitialized to the original burst value at the end of the operation. Hence the length of the first burst transfer of the next DMA operation may not be the same as that specified in the LT & RT registers.

Glitches on the Read Clock Input

The DDC has a minimum specification for the RCLK high time (rch) and the RCLK low time (rcl). Any RCLK not within these specifications is taken as a clock with the glitch. If such a situation is presented to the DDC then a number of things happen. This glitch results in throwing the Disk Sequencer in an unknown state, away from the standby state. Hence, in order for the DDC to be able to accept commands the chip has to be reset and reenabled in order to bring the sequencer to standby. A glitch on the RCLK can also potentially cause a situation leading to the altering of some register contents in the parameter RAM. Hence it is the designers responsibility to ensure that there are no possibilities of a glitch as defined by the specs on the RCLK line to the DDC and if it does reach the DDC, he should be aware of what to expect.

The DDC doesn't tolerate glitches on the RCLK input.

Remote DMA Completion Interrupt

The DMA on board the DDC is controlled by a separate sequencer. This DMA sequencer is responsible for generating the DMA completion interrupts and also controlling the LCB & RCB bits in the STATUS register. It is oblivious to the disk sequencer, in terms of the errors on the disk etc. However the interrupt generating mechanism for the remote DMA uses a clock from the disk PLA for synchronization purposes. This clock becomes inactive at certain times referred to as the freeze condition for the disk sequencer. This happens whenever a command is loaded in the DC register and the sequencer is waiting for a sync match in a disk read operation. Hence in the non-tracking mode if the remote DMA finishes at an instant when the disk sequencer is frozen then the remote DMA completion interrupt is held off till the next header comes along where the sequencer comes out of the frozen state and the clock is available. So this is more apt to happen when the remote DMA is under way while the disk sequencer is off looking for a header match. The other instances where the disk sequencer freezes is in a multisector operation; 1) the time after the header CRC and before the sync match for the data field occurs; 2) the time after the data field and just before the sync match of the header of the next field. Hence, if the remote DMA finishes around those instances then the completion interrupt could be delayed. The more serious implication of this situation is if the remote happened to finish just before the disk sequencer was entering the freeze mode, than the remote DMA completion interrupt would be held active till the sequencer comes out of the freeze state. Until then all efforts to service the interrupt by doing a status read will not deactivate the interrupt.

Hence the recommendation would be to initiate a remote operation only after a header match has occurred and to wait for the remote DMA completion before issuing another disk command. The other alternative would be to accommodate in software to look for such a situation and work around it. Software polling could be used to determine remote DMA completion and the interrupts from it ignored.

LRQ/RRQ Synchronization and Hold Off

In the DDC, the acknowledge signal in response to a request is sampled at the T4 state of the DMA transfer cycle. The chip does not require cycling of the acknowledge signal with the request from the chip. Also the initial assertion of the LRQ/RRQ signals is not synchronous to the BCLK.

Read Gate Algorithm for Harmonic Lock

If the read head was turned on over a write splice, the data separator may go into harmonic lock, which will prevent it from detecting the preamble pattern it is looking for. This forces zeroes data out of the data separator to the DDC and hence the DDC allows read gate to remain asserted, indefinitely. This is a lock up situation which must be avoided using external hardware.

17.0 Some Helpful Hints, When Designing a Disk Controller Subsystem with National Semiconductor's DP8466A, Disk Data Controller (DDC) (Continued)

Write Splice During a Disk Write

If a genuine FDL error occurs during a disk write function, the write gate will be deasserted as soon as the FIFO gets over-read. If this happens in the middle of a sector, it will result in a write splice to occur.

Read Gate Timing

Usually in most drives, when write gate is asserted, data actually gets written after 8 RCLKS, because of write driver delays etc. Hence the exists a write splice. In the DDC for a write data operation, the write gate is asserted 3 bit times into the data preamble. The read gate is asserted 8.5 bit times after the write gate, which is just sufficient to ensure assertion of read gate beyond the write splice associated with the write gate assertion. However at the beginning of the sector, both read and write gate are asserted 2–4 bit times from the index or sector pulse, hence resulting in the read gate being asserted in the write splice. External circuitry must be implemented to prevent this from happening.

FIFO Table Format

In the FIFO TABLE format mode the local DMA loads the correct number of header bytes (given by the HBC register) per sector into the FIFO from the local buffer memory. This data is then substituted for the header bytes during a format operation. It should be noted that each header byte set must contain an even number of bytes. If it contains an odd number of bytes, an extra dummy byte must be inserted so that each header byte set starts at an even byte boundary.

Two Interrupts in a Read Disk Operation

In a read disk operation, there is a potential for the controller μP to see two interrupts from the DDC, if a DFE error occurred during the operation. One of them is due to the error condition reflecting the DFE error, while the other reflects the operation completion by the local DMA, i.e., when the local DMA has finished transferring the data. Depending on the local DMA speed and bus latency, this could occur before or after the DFE interrupt, and they could be within 16 RCLKS or further apart. If the two interrupts are within 16 RCLKS of each other, the μP sees only one interrupts, while if they occur more than 16 RCLKS apart, then there is a potential of two interrupts being presented to the controller μP. This situation should be kept in mind and handled in firmware accordingly.

ADS0 Glitch During First DMA Transfer

The ADS0 line is a bidirectional line. Hence when the DMA transfer is initiated, the ADS0 line changes from an input to an output. It is released from the input mode into a tristate condition. When released for the DMA operation, it tends to touch the high level and goes low when the address needs to be latched in the t1 cycle of the first DMA transfer. It has been observed that just prior to that instant due to an internal race condition it is possible that the ADS0 may momentarily go low in a glitch fashion. This does not really hurt the system because it will go low at the appropriate time to latch the correct address in the t1 cycle, however if the trailing edge of the strobe is monitored to initiate some operation in a system design, then this could pose a problem. This needs to be kept in mind while designing.

Restrictions for the 2 Byte Exact Burst DMA Transfer Mode

The two byte exact burst mode was intended to be used for systems with very fast BCLKS relative to the RCLKS, such as when using the DDC to write a floppy as back up. The 2 byte exact mode is not needed for quick bus access since this can always be accomplished by the arbitration logic (in any burst mode) by deasserting LACK and waiting a minimum of 4 BCLKS. This is a better response than the 2 byte exact burst mode when waiting for the LRQ to be deasserted.

The performance degradation of the DDC when in the exact burst mode is due to the following sequence of events.

1. A burst of data is transferred causing the LRQ to go inactive.
2. Because the FIFO is still in a condition which requires more data to be transferred, the LRQ must be reasserted.
3. This reassertion of LRQ is held off until both the FIFO address counters match in parity; that is until both counters are either odd or even. This results in the LRQ being held off until the disk strobe occurs which in some cases (see table below) will allow the DMA to transfer only at the same data rate as the disk.

The most exaggerated effect of this problem is when in the 2 byte exact burst mode when the data bus is in the byte mode. In this mode the following ratios of BCLK to RCLK must be observed for the corresponding DMA to disk transfer rates.

Byte Mode

BCLK/RCLK Ratio	Max DMA Transfer Rate
< 1/1.6	Will not be able to keep up with disk rate, will get FDL. Can only transfer at the disk rate, therefore any bus sharing will result in depleting the FIFO, with no ability to refill it.
> 1/1.6 but < 1/0.6	
> 1/0.6	Can transfer at least at 2X the disk rate. Can easily refill the FIFO if depleted.

Word Mode

< 1/1	Can transfer only at the disk rate. Depleted FIFO cannot be refilled.
> 1/1	Can transfer at least at 2X the disk rate.

Lost BCLK Cycles in DMA Burst Mode

During DMA burst mode operation, LRQ or RRQ is deasserted for two BCLK cycles between bursts of local or remote DMA, i.e., When a remote burst is followed by another remote burst, an extra BCLK cycle occurs between t4 of the prior burst and the t1 of the subsequent burst. Likewise this is true for a local burst followed by another local burst, with the exception that here there is a possibility of two dummy BCLK cycles being inserted between t4 and t1. However, if a remote burst is followed by a local burst or vice versa, no dummy BCLK cycles are introduced.

18.0 Appendix

18.1 DDC REGISTERS, INDEX BY HEX ADDRESS

The following is a repeat of what can be found in the DDC INTERNAL REGISTERS Section. This listing is arranged numerically by hex address, and is provided as a quick reference. The section numbers provided indicate where the best description for the particular register can be located. For an explanation of the information contained in the WR and RD columns, refer to the key in the INTERNAL REGISTERS Section.

COLUMN KEY:

HA: *Hex Address* **#B**: *Number of bits* **WR**: *Write* **RD**: *Read* **SC**: *Section*

HA	REGISTER	#B	WR	RD	SC
00	Status Register (S)	8	NO	R	3.1
01	Error Register (E)	8	NO	R	3.1
02	ECC SR Out 0	8	NO	R	3.4
02	Polynomial Preset Byte 0 (PPB0)	8	D	NO	3.4
03	ECC SR Out 1	8	NO	R	3.4
03	Polynomial Preset Byte 1 (PPB1)	8	D	NO	3.4
04	ECC SR Out 2	8	NO	R	3.4
04	Polynomial Preset Byte 2 (PPB2)	8	D	NO	3.4
05	ECC SR Out 3	8	NO	R	3.4
05	Polynomial Preset Byte 3 (PPB3)	8	D	NO	3.4
06	ECC SR Out 4	8	NO	R	3.4
06	Polynomial Preset Byte 4 (PPB4)	8	D	NO	3.4
07	ECC SR Out 5	8	NO	R	3.4
07	Polynomial Preset Byte 5 (PPB5)	8	D	NO	3.4
08	Data Byte Count (0)	8	NO	R	3.4
08	Polynomial Tap Byte 0 (PTB0)	8	D	NO	3.4
09	Data Byte Count (1)	8	NO	R	3.4
09	Polynomial Tap Byte 1 (PTB1)	8	D	NO	3.4
0A	Polynomial Tap Byte 2 (PTB2)	8	D	NO	3.4
0B	Polynomial Tap Byte 3 (PTB3)	8	D	NO	3.4
0C	Polynomial Tap Byte 4 (PTB4)	8	D	NO	3.4
0D	Polynomial Tap Byte 5 (PTB5)	8	D	NO	3.4
0E	ECC CONTROL (EC)	8	D	NO	3.4
0F	Header Byte Count (HBC)/Interlock	3	F	R	3.1
10	Drive Command Register (DC)	8	C	NO	3.1
11	Operation Command Register (OC)	8	C	NO	3.1
12	Sector Counter (SC)	8	C	R	3.1
13	Number of Sector Operations Counter (NSO)	8	C	R	3.1
14	Header Byte 0 Pattern	8	C	R	3.3
15	Header Byte 1 Pattern	8	C	R	3.3
16	Header Byte 2 Pattern	8	C	R	3.3
17	Header Byte 3 Pattern	8	C	R	3.3
18	Header Byte 4 Pattern	8	C	R	3.3
19	Header Byte 5 Pattern	8	C	R	3.3
1A	Remote Data Byte Byte Count (L)	8	C	R	3.2
1B	Remote Data Byte Byte Count (H)	8	C	R	3.2
1C	DMA Address Byte 0	8	C	R	3.2

HA	REGISTER	#B	WR	RD	SC
1D	DMA Address Byte 1	8	C	R	3.2
1E	DMA Address Byte 2	8	C	R	3.2
1F	DMA Address Byte 3	8	C	R	3.2
20	Data Postamble Byte Count	5	D	R	3.3
21	ID Preamble Byte Count	5	C	R	3.3
22	ID Sync #1 (AM) Byte Count	5	D	R	3.3
23	ID Sync #Byte 2 Count	5	D	R	3.3
24	Header Byte 0 Control	5	D	R	3.3
25	Header Byte 1 Control	5	D	R	3.3
26	Header Byte 2 Control	5	D	R	3.3
27	Header Byte 3 Control	5	D	R	3.3
28	Header Byte 4 Control	5	D	R	3.3
29	Header Byte 5 Control	5	D	R	3.3
2A	Data External ECC Byte Count	5	D	R	3.3
2B	ID External ECC Byte Count	5	D	R	3.3
2C	ID Postamble Byte Count	5	D	R	3.3
2D	Data Preamble Byte Count	5	D	R	3.3
2E	Data Sync #1 (AM) Byte Count	5	D	R	3.3
2F	Data Sync #2 Byte Count	5	D	R	3.3
30	Data Postamble Pattern	8	D	R	3.3
31	ID Preamble Pattern	8	D	R	3.3
32	ID Sync #1 (AM) Pattern	8	D	R	3.3
33	ID Sync #2 Pattern	8	D	R	3.3
34	Gap Byte Count	8	F	R	3.3
35	Disk Format Register (DF)	8	D	NO	3.1
36	Header Diagnostic Readback (HDR)	8	NO	R	3.1
36	Local Transfer Register	8	I	NO	3.2
37	DMA Sector Counter (DSC)	8	NO	R	3.2
37	Remote Transfer Register	8	I	NO	3.2
38	Sector Byte Count 0	8	D	R	3.2
39	Sector Byte Count 1	8	D	R	3.2
3A	Gap Pattern	8	F	R	3.3
3B	Data Format Pattern	8	F	R	3.3
3C	ID Postamble Pattern	8	D	R	3.3
3D	Data Preamble Pattern	8	D	R	3.3
3E	Data Sync #1 (AM) Pattern	8	D	R	3.3
3F	Data Sync #2 Pattern	8	D	R	3.3

18.0 Appendix (Continued)

18.2 ALPHABETICAL MNEMONIC GLOSSARY AND INDEX

Listed on the following pages are the majority of the abbreviations used within this data sheet as mnemonics to describe portions or functions of the DDC. The section numbers referenced indicate where the terms are first defined. Mnemonics from the specifications section are not included here.

MNEMONIC	DESCRIPTION	SECTION
AD0–7	Address/Data 0–7 (pins 41–48)	2.0
AD8–15	Address/Data 8–15 (pins 1–8)	2.0
ADS0	Address Strobe 0 (pin 9)	2.0
ADS1	Address Strobe 1 (attached to RRQ, pin 37)	2.0
AME	Address Mark Enable (attached to LPRE, pin 13)	2.0
AMF	Address Mark Found (attached to EPRE, pin 16)	2.0
BCLK	Bus Clock (pin 40)	2.0
CCA	Correction Cycle Active (bit in Status register)	3.1
CF	Correction Failed (bit in Error register)	3.1
$\overline{CS}$	Chip Select (pin 28)	2.0
CS0–3	Correction Span Selection (bits in EC register)	3.4
DC	Drive Command register	3.1
DNE	Data Non-Encapsulation (bit in EC register)	3.4
DF	Disk Format register	3.2
DFE	Data Field Error (bit in Error register)	3.1
D01, 2	Data Operation bits (command in DC register)	3.1
DSC	DMA Sector Counter	3.2
E	Error register	3.1
EC	ECC Control register	3.4
ED	Error Detected (bit in Status register)	3.1
EEF	External ECC Field (pin 26)	2.0
EEW	Enable External Wait (bit in RT register)	3.2
EHF	Enable HFASM Function (bit in HC0–5 registers)	3.3
EHI	Enable Header Interrupts (command in OC register)	3.1
FTF	FIFO Table Format (bit in DF register)	3.1
HBA	Header Byte Active (bit in HC0–5 registers)	3.3
HBC	Header Byte Count register	3.1
HC0–5	Header Byte 0–5 Control registers	3.3
HDR	Header Diagnostic Readback register	3.1
HNE	Header Non-Encapsulation (bit in EC register)	3.4
HF	Header Fault (bit in Status register)	3.1
HFASM	Header Failed Although Sector number Matched (bit in Error register)	2.0
HMC	Header Match Completed (bit in Status register)	3.1
H01, 2	Header Operation bits (command in DC register)	3.1
HSS	Hard or Soft Sectored (bit in DF register)	3.1
ID1, 2	Internal Data Appendage (bits in DF register)	3.1
IDI	Invert Data In (bit in EC register)	3.4
IH1, 2	Internal Header Appendage (bits in DF register)	3.1
$\overline{INT}$	Interrupt (pin 29)	2.0
LA	Long Address (bit in LT register)	3.2
LACK	Local DMA Acknowledge (pin 39)	2.0
LBL1, 2	Local Burst Length (bits in LT register)	3.2
LCB	Local Command Busy (bit in Status register)	3.1
LI	Late Interlock (bit in Error register)	3.1
LPRE	Late Precompensation (attached to AME, pin 13)	2.0
LRQ	Local DMA Request (pin 36)	2.0
LRQ	Local Request (bit in Status register)	3.1
LSRW	Local Slow Read/Write (bit in LT register)	3.2
LT	Local Transfer register	3.2
LTEB	Local Transfer Exact Burst (bit in LT register)	3.2
LWDT	Local Word Data Transfer (bit in LT register)	3.2
MFM	MFM Encode (bit in DF register)	3.1
MSO	Multi-Sector Operation (command in DC register)	3.1
NCP	Not Compare (bit in HC0–5 registers)	3.3
NDC	Next Disk Command (bit in Status register)	3.1
NDS	No Data Synch (bit in Error register)	3.1
NSO	Number of Sector Operations counter	3.1
OC	Operation Command register	3.1
PPB0–5	Polynomial Preset Byte 0–5	3.4
PTB0–5	Polynomial Tap Byte 0–5	3.4
RACK	Remote DMA Acknowledge (pin 38)	2.0
RBL1, 2	Remote Burst Length (bits in RT register)	3.2
RBO	Reverse Byte Order (bit in LT register)	3.2
RCB	Remote Command Busy (bit in Status register)	3.1
RCLK	Read Clock (pin 25)	2.0
$\overline{RD}$	Read (pin 11)	2.0
RDATA	Read Data (pin 15)	2.0
RED	Re-Enable DDC (command in DC register)	3.1
RES	Reset DDC (bit OC register)	3.2
RGATE	Read Gate (pin 19)	2.0
RRQ	Remote Request (attached to ADS1, pin 37)	2.0
RS0–5	Register Select 0–5 (pins 30–35)	2.0
RSRW	Remote Slow Read/Write (bit in RT register)	3.2
RT	Remote Transfer register	3.2
RTEB	Remote Transfer Exact Burst (bit in RT register)	3.2
RWDT	Remote Word Data Transfer (bit in RT register)	3.2
S	Status register	3.1
SAIS	Start At Index or Sector (command in DC register)	3.1
SAM	Start at Address Mark (bit in DF register)	3.1
SC	Sector Counter	3.1
SCC	Start Correction Cycle (command in OC register)	3.1
SDV	Serial Data Valid (pin 27)	2.0
SLD	Select Local DMA (bit in LT register)	3.2
SNF	Sector Not Found (bit in Error register)	3.1
SO	Sector Overrun (bit in Error register)	3.1
SRD	Select Remote DMA (bit in RT register)	3.2
SRI	Start Remote Input (command in OC register)	3.1
SRO	Start Remote Output (command in OC register)	3.1
SSC	Substitute Sector Counter (bit in HC0–5 registers)	3.3
TM	Tracking Mode (bit in RT register)	3.2
WCLK	Write Clock (pin 21)	2.0
WDATA	Write Data (pin 18)	2.0
WGATE	Write Gate (pin 20)	2.0
$\overline{WR}$	Write (pin 10)	2.0

Designing an ESDI (Enhanced Small Device Interface) Disk Controller Subsystem with National's DP8466A (Disk Data Controller)

National Semiconductor
Application Note 500

1.0 INTRODUCTION

The ESDI (Enhanced Small Disk Interface) is designed to handle a variety of 5¼" Winchester disk, tape and optical drives. It opens the door to higher performance system designs by incorporating more intelligence onto the drives, and by allowing higher data transfer rates—10 Mbits/s to 24 Mbits/s. This is achieved by incorporating data separation, data encoding and decoding in the drive itself and the smarter interface protocol allows dissemination of more information between the drive and the controller. Thus by removing the restrictions placed by ST506 on 5¼" hard disk transfer rates, the ESDI interface clears the way for higher recording densities and ultimately, higher storage capacities in the 5¼" form factor, up to 700 Mbytes and beyond.

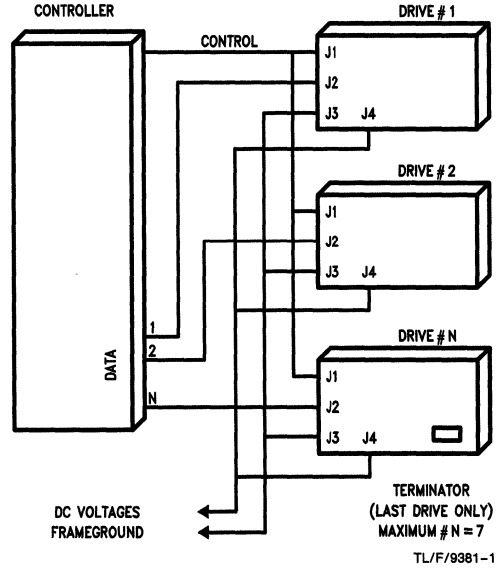
National Semiconductor's DP8466A Disk Data Controller integrates a number of functions originally supported by discrete logic in conventional disk controller designs. This results in a decrease in complexity and parts count in a disk controller design. The DP8466A is a data path controller and hence can support the various disk interfaces viz. ST506, ESDI, SMD, etc. By its versatility and programmability, it greatly simplifies the task of designing a disk controller. There are basically two types of ESDI drives viz. hard sectored ESDI drives and soft sectored ESDI drives. This application note discusses the ESDI interface and the various steps involved in designing an ESDI disk controller with the DP8466A. The emphasis is predominantly on the disk side as that is of utmost relevance with respect to intefacing the DP8466A.

2.0 ENHANCED SMALL DEVICE INTERFACE (ESDI)

The ESDI consists of a control cable and a data cable. The control cable allows for a daisy chain connection of up to seven drives with only the last drive being terminated. The data cable must be attached in a radial fashion. *Figure 1* shows a typical connection in a multiple drive system. All control lines are digital in nature (open collector TTL) and either provide signals to the drive (input) or signals to the controller (output). The data transfer signals are differential in nature and provide data either to, (write) or from, (read), the drive.

2.1 Control Cable

The control cable definition is shown in *Figure 2*. It basically consists of some input lines and some output lines. The control input signals are of two kinds; those to be multiplexed in a multiple drive system and those intended to do the multiplexing. The control input signals to the drive to be multiplexed are WRITE GATE, READ GATE, HEAD SELECT 2^0, 2^1, 2^2, 2^3, TRANSFER REQUEST and COMMAND DATA. The signals to do the multiplexing are the DRIVE SELECT 1,2,3. ADDRESS MARK ENABLE (AME) is a control input on the radial data cable and is not multiplexed. The drive select lines accept a binary input combination decoded internally to allow 1–7 drives. Decode 000 is a no select condition. The four Head select lines allow selection of each individual read/write head in a binary coded sequence. Heads are numbered 0 thru 15. Write Gate and Read Gate are control signals which initiate writing and reading of data respectively at the disk.



TL/F/9381–1

FIGURE 1. Typical Connection in a Multiple Drive System

Control Cable		Data Cable	
Pin No.	Signal	Pin No.	Signal
2	Head Select 2^2	1	Drive Selected
4	Head Select 2^1	2	Sector/AMF
6	Write Gate	3	Command Complete
8	Config/Status	4	Address Mark Enable
10	Transfer ACK	7,8	Write Clock ±
12	Attention	10, 11	Read/Reference CLK ±
14	Head Select 2^0	13,14	NRZ Write Data ±
16	Sector/AMF	17,18	NRZ Read Data ±
18	Head Select 2^3	20	Index
20	Index	Ground = 5,6,9,12,15,16,19	
22	Ready		
24	Transfer Request		
26	Drive Select 1		
28	Drive Select 2		
30	Drive Select 3		
32	Read Gate		
34	Command Data		
Ground = All Odd # Pins			

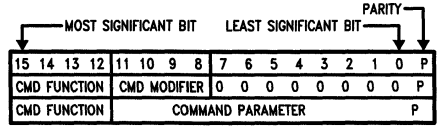
FIGURE 2. ESDI Cable Assignments

The ESDI is an intelligent interface and provides for certain commands, to do certain specific functions, thereby freeing the controller from a number of mundane tasks, and also from being tied to a particular drive. These commands are presented to the drive as 16 information bits of serial data, plus a parity bit. The transfer of this data is controlled by the handshake protocol with the TRANSFER REQUEST and TRANSFER ACKNOWLEDGE signals. Upon receipt of this serial data, the drive will perform the required function as specified by the bit configuration. *Figure 3* lists the various commands supported by ESDI.

The Address Mark Enable (AME) signal behaves differently for soft and hard sectored drives. In soft sectored drives this signal, when active with Write Gate, writes an Address Mark on the disk. When AME is asserted without Write Gate or Read Gate, it causes a search for Address Marks. The address mark written is usually a gap of no flux transitions, exactly 24 bits long. In case of hard sectored drives, the AME does not cause an Address Mark to be written on the media. The trailing edge of AME with Write Gate asserted, initiates the writing of the ID PLO sync field.

The output control signals are driven with an open collector output stage capable of sinking a maximum current of 48 mA. They consist of the DRIVE SELECTED, READY, ATTENTION, INDEX, SECTOR, TRANSFER ACKNOWLEDGE, and CONFIGURATION & STATUS. COMMAND COMPLETE is a control output which allows the host to monitor the drive's command completion status. The ATTENTION line is activated whenever there is an erroneous condition at the drive. In response to the Request Configuration and Request Status commands, the drive provides some status information which is sent to the controller in a

Command Data Word Structure



TL/F/9381–2

Command Data Definition

Command Function Bit 15	14	13	12	Command Function Definition	Command Modifier Applicable Bits 11–8	Command Parameter Applicable Bits 11–0	Status Data Returned to Host
0	0	0	0	Seek	No	Yes	No
0	0	0	1	Recalibrate	No	No	No
0	0	1	0	Request Status	Yes	No	Yes
0	0	1	1	Request Config	Yes	No	Yes
0	1	0	0	Select Head Group*	No	Yes	No
0	1	0	1	Control	Yes	No	No
0	1	1	0	Data Strb Offset*	Yes	No	No
0	1	1	1	Track Offset	Yes	No	No
1	0	0	0	Init Diagnostics*	No	Yes	No
1	0	0	1	Set Bytes/Sector*	No	Yes	No
1	0	1	0	Reserved			
1	0	1	1	Reserved			
1	1	0	0	Reserved			
1	1	0	1	Reserved			
1	1	1	0	Set Config*	No	Yes	No
1	1	1	1	Reserved			
*Optional Commands					All Unused Bits Set to Zero		

FIGURE 3. ESDI Command Structure

serial manner over the CONFIGURATION & STATUS line using a handshake protocol between TRANSFER REQUEST and TRANSFER ACKNOWLEDGE.

Index and sector are interface signals from the drive which indicate the start of a track and sector respectively. In case of a soft sectored drive there are no sector pulses, but if an address mark is found then it signals the end of an address mark, indicated by the AMF signal from the drive.

2.2 Data Cable

All lines associated with the transfer of data between the drive and the host system are differential in nature and may not be multiplexed. Four pairs of balanced signals are used for the transfer of data and clock signals: NRZ Write Data, NRZ Read Data, Write Clock, and Read/Reference Clock. Differential drivers and receivers are needed to interface the controller to the drive, like National's DS8922A/DS8923A. Connection details are shown in *Figure 13*. The NRZ Write Data is clocked by the Write Clock signal while the NRZ Read Data is clocked by the Read Clock signal. The Reference Clock signal from the drive will determine the data transfer rate. The transitions from Reference Clock to Read Clock must be performed without glitches, i.e. the clock should not violate the minimum allowable specifications of the controller chip.

2.3 ESDI Format Rules

The ESDI essentially supports a fixed sector implementation, (Drive hard sectored) and a soft sectored implementation, (Drive soft sectored). The record format on the disk is under control of the controller, however, the ESDI standard recommends a certain format structure which must be implemented. In a hard sectored drive the index pulse signifies the start of a track, while the sector pulse signifies the start of a sector. *Figure 4* shows a fixed sector format and associated timings. In a soft sectored drive, the index signifies the beginning of a track while the beginning of each sector is defined by an Address Mark, followed by the ID field which contains the header information. The AME/AMF handshake is utilized to detect these address marks. *Figure 5* shows the soft sectored format and associated timings. In a hard sectored drive, the beginning of the ID PLO sync field is specified by the trailing edge of the AME, when Write Gate is active. From *Figures 4, 5* and *6* it can be seen that there are some minor differences between the DDC's format and the ESDI recommended format. The ESDI recommended format supports a post index/sector gap field and a write splice field between the ID and DATA fields, which is not directly supported by the DDC. Also Write Gate needs to be optionally deasserted in the write splice area between the ID and DATA segments as shown in *Figures 4* and *5*. These shortcomings can however be overcome through a combination of hardware and software considerations as discussed in the following sections.

3.0 CONTROLLER DESIGN—DISK SIDE

Perhaps of greatest significance to system designers is the fact that the ESDI drives provide the data separation function internally. The performance benefits attained by putting the data separator on the drive more than offset the cost in terms of system efficiency and reliability. Data is transferred over the interface in NRZ format. This results in the use of high density encoding schemes to be implemented in the drive, like 2,7 RLL, etc. These factors greatly simplify the task of designing a controller for ESDI drives. This task is even more simplified with the availability of VLSI disk controller IC's like the DP8466A, which integrates numerous

ESDI Hard Sectored Format

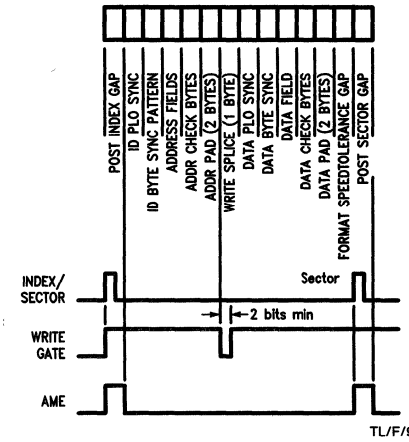


FIGURE 4. ESDI Fixed Sector Format and Relevant Timing

functions in the disk path. The disk side involves the interface of the DDC to a disk interface, like the ESDI. This is made up of two main paths—the data path and the control path. The control path is responsible for the disk related functions like sending commands, etc (as discussed in Section 2.1), while the data path is responsible for the data transfer, (refer to Section 2.2). The DP8466A is a disk data path controller which does not involve itself with the slower tasks of the control path. It features full format programmability, fully programmable ECC, 16-bit dual channel DMA and a 32-byte FIFO. Data is transferred from the FIFO in selectable bursts, which minimizes bus occupancy and can thereby accommodate some degree of latency. For more details refer to AN-413. As mentioned in the previous section, the disk formats suggested by the ESDI standard produce some compatibility problems with the DP8466A, mostly in the area of control line timing with the drive. The techniques to handle them with minimum overhead is discussed below for the two types of ESDI drive systems.

ESDI Soft Sectored Format

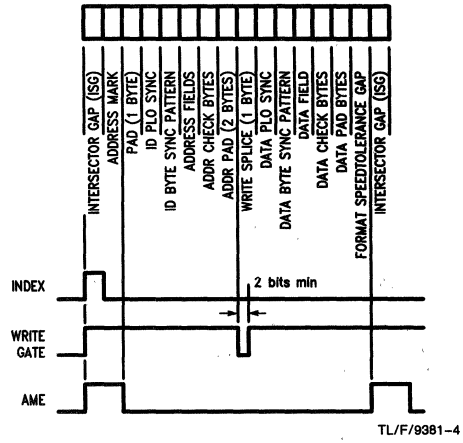


FIGURE 5. ESDI Soft Sectored Format and Relevant Timing

3.1 ESDI Hard Sectored Drive

For an ESDI drive which is hard sectored, the specification calls for an Inter-Sector Gap (ISG) which is to precede and follow the index/sector pulse,also referred to as the post index/sector pulse. This gap is needed to provide the drive with an area for the embedded servo (if used) and also gives the controller time to assert read gate. While formatting the drive, the end of the ISG (post index/sector gap), is indicated by the trailing edge of AME. This is needed by the drive to indicate the beginning of the PLO sync (preamble) field, necessary when the drive encodes the PLO sync field with a non-standard preamble pattern, e.g. as in 2,7 encoding with 3T or 4T preambles. The DDC generates the AME signal with the necessary timing during a format operation. To do so the DDC must be programmed to be in the hard sectored mode and have the start with address mark bit enabled in the Disk Format Register, (HSS and SAM bits in the DF register). *Figure 6* illustrates the manner in which the DDC format parameter registers need to be manipulated for a format operation and a read/write operation. It should be noted that the ISG field is implemented by the DDC's ID preamble field while the PLO sync is implemented by the ID sync #1 field. This feature is needed only during formatting, and hence should be disabled at other times. On the other hand when a disk read/write operation is attempted, the header field is compared (or read) usually. The DDC asserts read gate just after the index or sector pulse, which initiates the data separator to start locking the PLL to the preamble. Since there is the ISG field after the index/sector pulse, the read gate to the drive needs to be delayed until after the ISG. This can be achieved in two ways. One technique is to delay index and sector to the DDC by the length of the ISG, so that the DDC would be asserting Read Gate in the preamble area. *Figure 6* shows the format and control signals for this situation. This is the technique adopted for the combined solution proposed in section 3.3 to implement both hard and soft sectored ESDI.

Alternate Technique

The other technique for implementing the post index sector gap is explained below and could be used when only ESDI hard sectored drives are under consideration. Essentially when formatting the drive, the DDC is set up to use ID preamble field as the ISG, the ID sync #1 as the PLO synch and the ID sync #2 as the sync byte. Hence when reading or writing, the DDC is set up to have ID preamble as the PLO sync, ID sync #1 is skipped and ID sync #2 as sync byte. External hardware is used to delay the read gate at the beginning of every sector, by the length of the ISG so that it gets asserted over the preamble on the drive. It should however be noted that the read gate needs to be delayed only at the beginning of the sector and not in the middle of the sector before the data field. *Figure 7* outlines the format manipulating and control signals behaviour.

3.2 ESDI Soft Sectored Drive

The soft sectored specification of ESDI provides two major stumbling blocks for the DDC. First there is a need for providing the handshaking for the AME and AMF lines. The controller needs to raise AME when it wishes to be notified of the start of a sector. The controller asserts Read Gate on detecting AMF, generated by the drive on finding an Address Mark of 24 bits. After seeing the drive assert AMF, the controller removes AME, finishing the handshake. Since the DCC does not provide the AME/AMF handshake, external hardware is required to do so when the DDC is not formatting. As the Address Mark is at the beginning of every sector, the AMF signal is seen by the DDC as a sector pulse. The DDC is programmed to believe that it is looking at a hard sectored drive, and thereby the soft sectored ESDI can be handled in this pseudo fashion.

The second stumbling block is that the soft sectored ESDI drives generates an AMF signal for each sector on the disk, including the first. Because the DDC starts its operation on either index or sector, (AMF instead of sector in this case), it will believe that there is a sector between index and the first AMF. External circuitry is needed to eliminate the first AMF pulse and replace it with a delayed version of index.

One other problem with soft sectored ESDI, is formatting the drive. In these drives there is usally a ISG following the index pulse. In addition to this is the Address Mark field, which fits between the ISG and the PLO sync fields. The DDC has only two fields (ID preamble and ID sync #1) before the sync byte, with which these three fields have to be created. One way to implement it with external circuitry is to use the ID preamble field of DDC for writing the ISG and AM. The external circuitry delays the AME from the DDC by the length of the ISG. The address mark (AM) field will hence be the length of the ID preamble less the delay and the PLO sync is implemented using the ID sync #1 field. *Figure 8* shows the format manipulation required and the control signals behaviour.

3.3 Combined Solution

A combined solution to the above problems, (for a system supporting both soft and hard sectored ESDI), can be provided by a single PAL® device and something to provide a delay (possibly another PAL device). The interface solutions can be grouped into two main areas, implemented as state machines.

1. Index and sector pulse generation to the DDC
2. AME/AMF handshaking between the DDC and the drive.

3.3.1 The Index/Sector Machine

There are essentially 4 types of drive operations which are encoded using the two PAL inputs (soft/hard/ and format). These are outlined below:

1. Hard sectored drive, read or write data operation
   PAL Inputs: soft/hard/ = 0 and format = 0

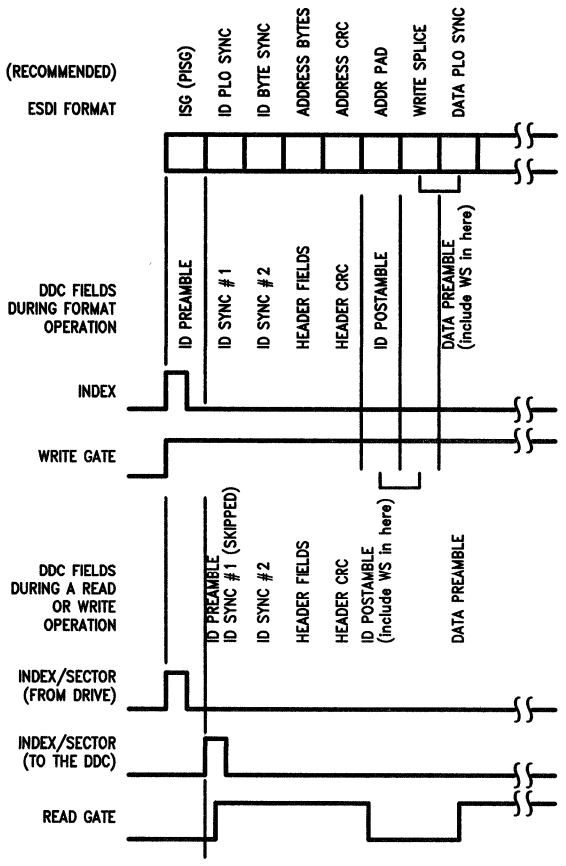
In this case the index/sector machine essentially delays both the index and sector pulse from the drive by the length of the ISG (post index/sector gap) and then present it to the DDC. *Figure 10* shows the state diagram and truth table for this part of the index/sector machine. Three state variables are used: F1 and F2 to indicate the state, and G1 to signal if an index pulse has been received from the drive. *Figure 9* shows the state diagram, truth tables and timing relationships for this part of the machine.

2. Hard sectored drive format operation
   PAL inputs: soft/hard/ = 0 and format = 1

In this case the index/sector machine essentially lets the index and sector pulses from the drive flow through to the DDC.

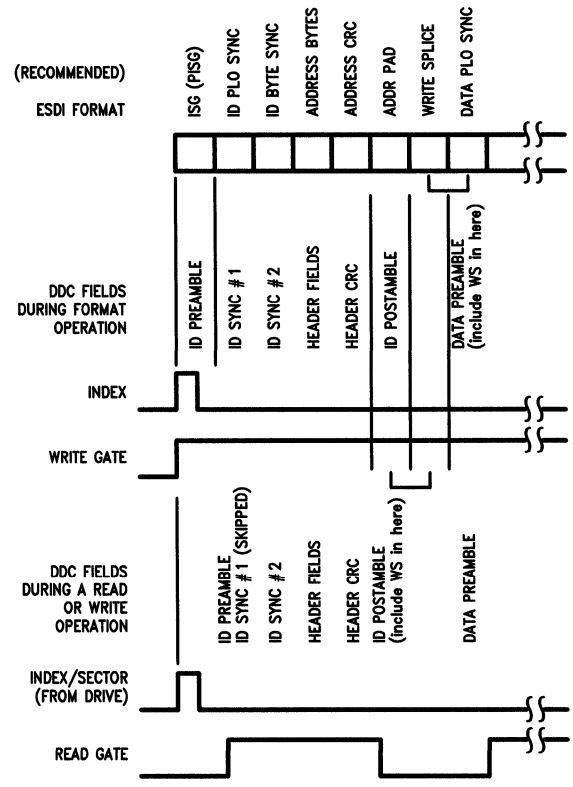
3. Soft sectored drive, read or write operation
   PAL inputs: soft/hard/ = 1 and format = 0

In this case the index/sector machine essentially follows the AMF from the drive, translating it as the sector input to the DDC; however if an index pulse occurs from the drive then it waits for an AMF from the drive and then generates an index and no AMF/sector pulse to the DDC. This is the scenario which happens in the sector just after the index



TL/F/9381–5

FIGURE 6. Programming the DDC Format Parameters in the Case of an ESDI Hard Sectored Implementation



TL/F/9381–6

FIGURE 7. Programming the DDC Format Parameters in the Case of an ESDI Hard Sectored Implementation (Alternative Technique)

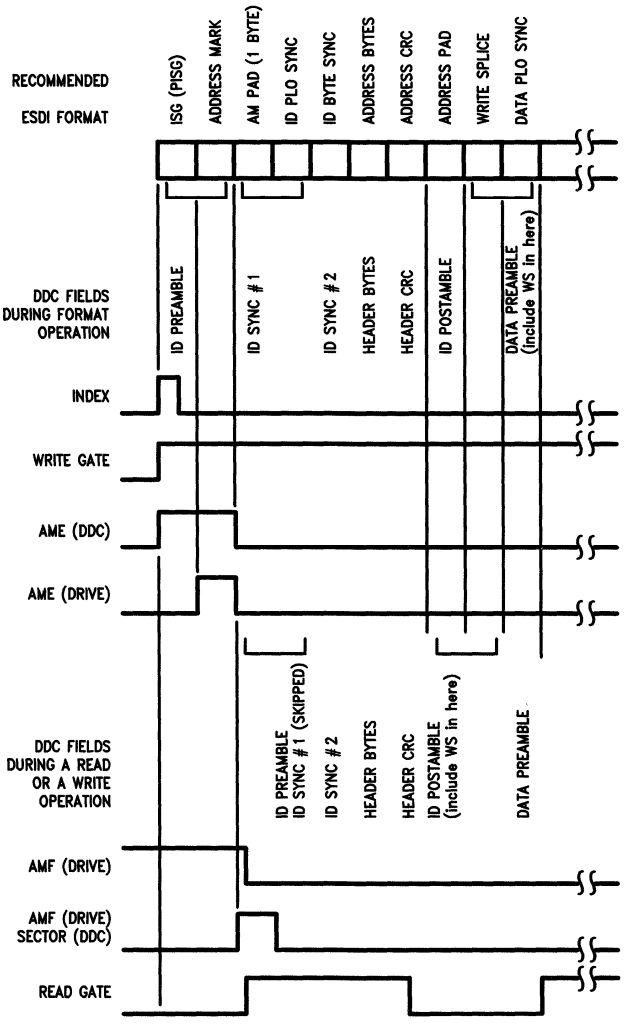
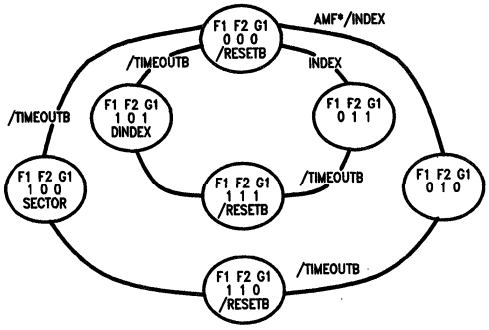


FIGURE 8. Programming the DDC Format Parameters in the Case of an ESDI Soft Sectored Implementation

State Machine

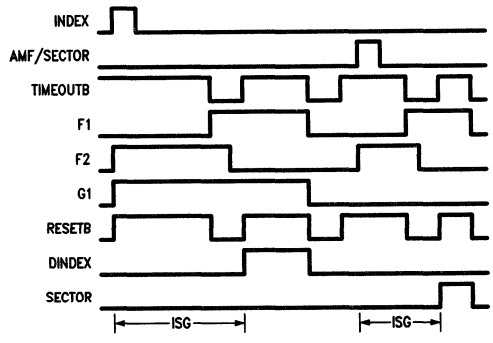


TL/F/9381–8

Truth Table

F1	F2	G1	Index	AMF	TIMEOUTB	F1	F2	G1	RESETB	DINDEX	Sector
Present State			Inputs			Next State			Outputs		
0	0	X	0	0	X	0	0	0	0	0	0
0	0	0	1	X	X	0	1	1	1	0	0
0	0	0	0	1	X	0	1	0	1	0	0
0	1	0	X	X	1	0	1	0	1	0	0
0	1	1	X	X	1	0	1	1	1	0	0
0	1	0	X	X	0	1	1	0	0	0	0
0	1	1	X	X	0	1	1	1	0	0	0
1	1	0	X	X	X	1	0	0	1	0	0
1	1	1	X	X	X	1	0	1	1	1	0
1	0	0	X	X	1	1	0	0	1	0	1
1	0	1	X	X	1	1	0	1	1	1	0
1	0	X	X	X	0	0	0	0	1	0	0

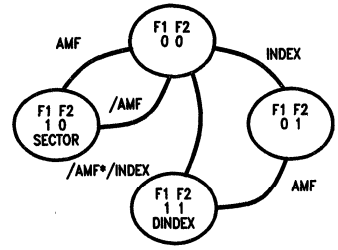
Index/Sector Machine Timing Diagrams



TL/F/9381–9

FIGURE 9. Index/Sector Machine Hard Sectored ESDI (Non-Format) State Diagram, Truth Tables and Timing

State Diagram



TL/F/9381–10

Truth Table

F1	F2	Index	AMF	F1	F2	DINDEX	Sector
Present State		Inputs		Next State		Outputs	
0	0	0	0	0	0	0	0
0	0	1	0	0	1	0	0
0	1	X	0	0	1	0	0
0	1	X	1	1	1	1	0
1	1	1	X	1	1	1	0
1	1	X	1	1	1	1	0
1	1	0	0	0	0	0	0
0	0	X	1	1	0	0	1
1	0	X	1	1	0	0	1
1	0	X	0	0	0	0	0

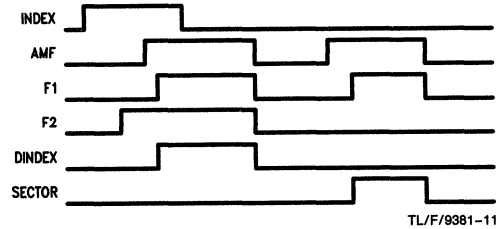
hole. *Figure 10* shows the state diagram, truth table and timing relationships for this section of the Index/Sector machine. It uses two state variables, F1 and F2.

4. Soft sectored drive, format operation
   PAL inputs: soft/hard/ = 1 and format = 1

In this case the index/sector machine essentially follows the index pulse from the drive and passes it on to the DDC while it forces the sector input to the DDC to be inactive (low).

3.3.2 The Address Mark Machine

The Address Mark Machine consists of a pair of multiplexers which feed the AME input to the drive and translate the AMF from the drive as the sector pulse input to the DDC. Once again as before there are 4 types of drive operations, encoded using the two PAL inputs soft/hard/ and format.



TL/F/9381–11

FIGURE 10. Index/Sector Machine Soft Sectored ESDI (Non-Format) State Diagram, Truth Table and Timing

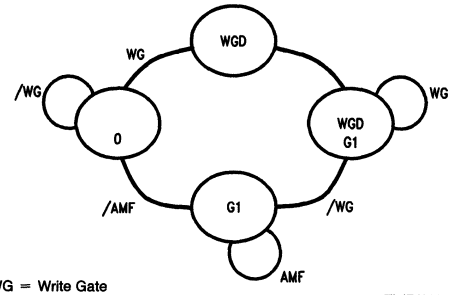
The working of the address mark machine in these cases is outlined below.

1. Hard sectored drive, Read and Write operation
   PAL inputs: soft/hard/ = O and format = O.

The AME to the drive is kept inactive (low).

2. Hard sectored drive, format operation
   PAL inputs: soft/hard/ = O and format = 1

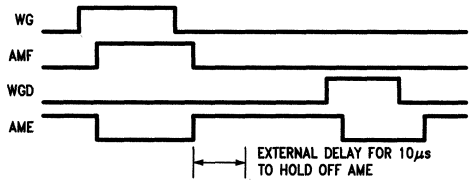
State Diagram



WG = Write Gate
WGD = Write Gate Delayed by One Clock

TL/F/9381–12

Timing Diagrams—Soft Sectored (Non-Format) Operation



TL/F/9381–13

In this case the AME to the drive follows the DDC generated AME.

3. Soft sectored drive, read or write operation
   PAL inputs: soft/hard/ = 1 and format = O

In this case the PAL generates AME and handshakes with the AMF from the drive, translating it to the DDC as the sector input.

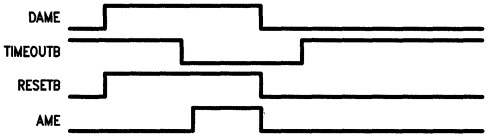
4. Soft sectored drive format operation
   PAL inputs: soft/hard/ = 1 and format = 1

In this case the leading edge of AME generated by the DDC is delayed by the length of the ISG (post index/sector gap) and presented to the drive.

Note: The trailing edge is not delayed.

Figure 11 shows the state diagram and timing relationships for the Address Mark Machine. Given below are the PAL equations for this control PAL implemented in a 16R4. These include some simplifications from the above information. In particular the hard sector non-format equations for F1, F2 and G1 can drop the use of the term /FORMAT, since during the format operation it is quite acceptable to have these output behaving as for non-format since the index to the DDC follows the index from the drive. The equations are written in PLAN format.

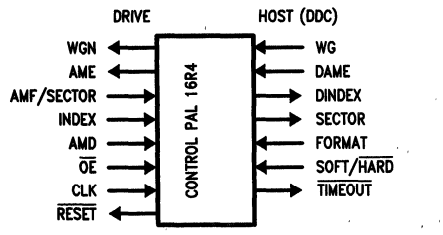
Timing Diagrams—Soft Sectored Format Operation



TL/F/9381–14

FIGURE 11. Address Mark Machine Details

Control PAL Pin Description



TL/F/9381–15

Control PAL Equations (Plan Format)
Version 1.3 Dated: 05/12/86

```
/wgd := /wg

gl   := (soft*wgd*wg)+(soft*wgd*gl)+
        (soft*/wgd*/wg*amf)+
        (/soft*/gl*index)+(/soft*gl*/fl)+
        (/soft*gl*f2)+(soft*gl*/timeout)

fl   := (soft*/format*f2*amf)+
        (soft*/format*fl*ff2*index)+
        (soft*/format*fl*/f2*amf)+
        (soft*/format*/fl*/f2*/index*amf)+
        (/soft*/fl*f2*timeout)+(/soft*fl*f2)+
        (/soft*fl*/f2*/timeout)

f2   := (soft*/format*/fl*/f2*index)+
        (soft*/format*/fl*f2)+
        (soft*/format*fl*f2*amf)+
        (soft*/format*fl*f2*index)+
        (/soft*/fl*/f2*index)+
        (/soft*/fl*/f2*amf)+
        (soft*/fl*f2)

if (vcc) /ame   = (soft*/format*gl)+
                  (soft*/format*wg)+
                  (soft*format*/dame)+
                  (soft*format*/timeout)+
                  (/soft*/format)+
                  (/soft*format*/dame) + (amd)

if (vcc) reset  = (soft*/format)+
                  (soft*format*/dame)+
                  (/soft*/format*/fl*/f2)+
                  (/soft*/format*fl*f2)+
                  (/soft*format)

if (vcc) /dindex =(soft*/format*/fl)+
                  (soft*/format*/f2)+
                  (format*/index)+
                  (/soft*/format*/fl)+
                  (/soft*/format*f2)+
                  (/soft*/format*/gl)+
                  (/soft*/format*timeout)

if (vcc)/sector  =(/format*/fl)+
                  (soft*/format*f2)+
                  (soft*format)+
                  (/soft*/format*f2)+
                  (/soft*/format*gl)+
                  (/soft*/format*timeout)+
                  (/soft*format*/amf)
```

3.3.3 Timer PAL

This is essentially the PAL used to generate the delay. When the RESET input to the PAL is low it does not count. It starts counting and when the desired time delay is reached it produces an active low output ($\overline{\text{TIMEOUT}}$). The value of the delay is variable from 1 to 32 byte times. The counter is clocked by the disk's RCLK which is divided by 8 to provide a byte-rate clock to a five-stage counter. When $\overline{\text{RESET}}$ is low (active), the inverse of the values GA–GE are loaded into the counter, and one byte-time after the counter reaches 11111 the TIMEOUT output goes low and counting stops. The value on GA–GE (GA is least significant bit) should be set to (number of bytes delay—1). Given below are the equations for this PAL 20X10). Note that PLAN requres the use of a dummy term to complete the OR function before the XOR. Hence for example, in the equation for F1 the term "F1*/F1" is put in, which always equals to 0.

Timer PAL Equations (Plan Format)
Version 1.3 Dated: 05/12/86

```
f3 := (/timeout*/reset)+
      (f3*/f3) :+:(f3*/reset)

f2 := (/timeout*/reset*f3)+
      (f2*/f2) :+:(f2*/reset)

f1 := (/timeout*/reset*f2*f3)+
      (f1*/f1) :+:(f1*/reset)

qa := (/reset*f3*f2*f1)+
      (reset*ga) :+:(qa*/reset)

qb := (/reset*f3*f2*f1*/qa)+
      (reset*gb) :+:(qb*/reset)

qc := (/reset*f3*f2*f1*/qa*/qb)+
      (reset*gc) :+:(qc*/reset)

qd := (/reset*f3*f2*f1*/qa*/qb*/qc)+
      (reset*gd) :+:(qd*/reset)

timeout :=
   (/reset*f3*f2*f1*/qa*/qb*/qc*/qd*/qe)+
   (timeout*/timeout) :+:(timeout*/reset)
```

3.3.4 Some Other Timing Considerations

The ESDI specification imposes some additional timing restrictions which have to be accommodated for with external logic. These are outlined below:

1. AME to the drive cannot be asserted till at least 10 μs, after deassertion of Write Gate to the drive.
2. In the PAL solution discussed above, when the drive drops AMF at the end of the handshake, the PAL reasserts AME. The MAXTOR doesn't seem to like that and hence the AME to the drive must be held off for at least 8 μs, from the trailing edge of AMF.
3. On a similar token, the MAXTOR drive doesn't like AME to be active when Read Gate is asserted, (it usually activates ATTENTION on the drive). Hence AME needs to be held off at least 8 μs from trailing edge of Read Gate. This is accomplished using a mono shot, which generates a disable signal (Address Mark Disable, AMD), for 10 μs and while this is active the control PAL disables AME to the drive. Also since we are presenting the drive like a hard sectored one to the DDC, we could safely assume that AME need not be active with Read Gate. This is accomplished by gating the AME to the drive with the read gate. Hence if read gate is active, AME gets disabled to the drive. These could be incorporated within the control PAL if desired.

3.4 Handling the Optional ESDI Format Specifications

The ESDI has certain optional specifications which are not directly supported by the DDC. These are discussed below with explanations of the way they can be implemented.

3.4.1 Optional Deassertion of Write Gate between the ID and DATA Fields in a Format Operation

This option is not supported by the DDC, as once its starts the format operation, the Write Gate remains asserted for the entire track. The purpose of the deassertion of Write Gate between the ID and Data fields is usually to indicate to the 2,7 RLL encoder, the start of the data preamble. This enables the encoder to substitute the 3T or 4T preamble pattern. This feature could be implemented in two ways. Using external logic, the trailing edge of the SDV (Serial Data Valid) is used to gate the logic, count until the header postamble has been written and then force the Write Gate low for the required two bit times. This problem could also be circumvented in software by incorporating a two pass format. The first pass involves a regular format operation which will write the headers for all the sectors, but since there will be no de-assertion of Write Gate before the data fields, the proper data preamble will not be written. In the second pass a compare header-write data operation is done, where the Write Gate edge is used to initiate the drive generated preamble. It should be noted (as also pointed out in the ESDI specification), that this is necessary only if a read will be attempted after a format.

3.4.2 Handling the Write Splice Field between the ID and DATA Fields in the ESDI Format

The ESDI format specification recommends a 2-byte header postamble and a 1-byte write splice. The DDC does not have a separate field to implement the write splice. It is accomplished by software manipulation as follows. The format is programmed to have a 2-byte header postamble and a data preamble which is one byte longer than the desired length. This byte is taken as the write splice (a floating Byte). During a write operation, this floating byte is considered as part of the data preamble. As Write Gate is asserted 3 RCLKS into the data preamble, the "write splice" associated with Write Gate assertion, due to write driver turn on time, etc, occurs during the 1 byte of the data preamble which is the floating byte. When the sector is being read, this byte is attached to the header postamble. Since Read Gate is reasserted 11.5 RCLKS into the data preamble, this ensures that it doesn't get asserted in the splice. From the above data it can be concluded that for a normal operation in the DDC, Read Gate and Write Gate assertion in the data field are separated by 8.5 RCLKS in the data preamble, hence automatically taking care of the write splice as the first byte of the data preamble.

3.5 Critical Read and Write Parameters

There are a number of drive dependent parameters which must be met in order to ensure proper operation with an ESDI drive. These are summarized below, for consideration during actual design.

3.5.1 Read Function Parameters

1. A read operation may not be initiated until 15 μs following head switch.
2. Read Gate may not be asserted during a write splice or within ± 1 bit time of a write splice.
3. Read gate must be asserted within 16 bit times from the write gate assertion point when the current field was written.
4. Data (read) at the interface could be delayed by up to 9 bit times from the data recorded on the disk media.
5. RCLK and RDATA are valid within the number of PLO sync field bytes specified by the drive configuration after read enable and a PLO sync field is encountered.

3.5.2 Write Function Parameters

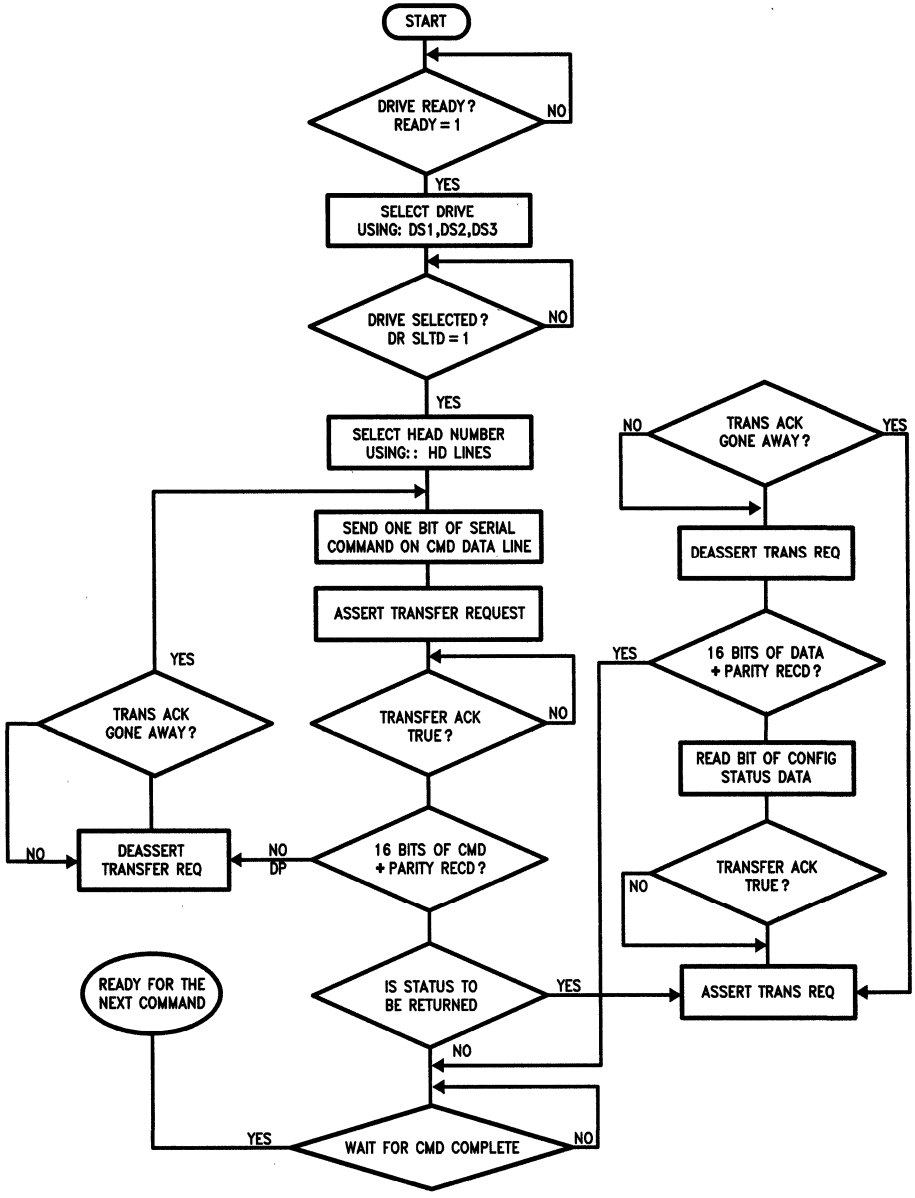
1. Assuming head selection is stabilized, the time lapse from deassertion of Read Gate to assertion of Write Gate shall be five reference clock periods minimum.
2. Write Clocks must precede Write Gate by a minimum of two and a half Reference Clock periods.
3. Write driver plus data-encoder turn on time (write splice width) is between 3 and 7 reference clock periods.
4. To account for data-encoding delays, write gate must be held on for at least 2 byte times after the last bit of information to be recorded. This implies a minimum data postamble length of 2 bytes.
5. The time lapse before Read Gate or AME can be asserted after deassertion of Write Gate is defined by the "ISG bytes after index/sector" in the configuration data response (10 μs).
6. Write Gate must be deasserted at least 1 μs before a head change and shall not be asserted until 15 μs after a head change or command complete.
7. Write data received at the I/O connector will be delayed by the encoder by up to 8 bits maximum prior to being recorded on the media.

Some of these parameters were accommodated in the combined PAL solution, while others need to be accommodated in firmware. The above discussion covered the hardware and relevant firmware considerations for designing the disk side data path section of the controller subsystem. The other aspect of the disk side design is the control path which is discussed in the following sections.

3.6 Disk Side Control Path Design

Since the DDC is a data path controller, the control path functions of the drive have to be controlled by the local intelligence in the disk controller subsystem. This offers more versatility and is less of a handicap, for it allows the DDC to be used with any of the disk interface standards. Also the control path functions like seeks, etc, are very slow operations and by not handling them, the DDC is able to achieve maximum operating speeds of 25 Mbits/sec. Any μP or μC with simple I/O ports would suffice for the control path functions. This usually is no extra overhead, because a local μP or μC is necessary to handle the protocol over the system bus and accordingly set up and activate the DDC anyway.

The control signals are sent over the drive's "A" control cable. Details of the respective signals are discussed in section 2.1. These signals are sent/received, to/from the drive through industry standard open collecter drivers (DP8311) and receivers (74ALS240), with appropriate terminations in accordance with the ESDI specification. These are also shown in the complete design, schematic *(Figure 13)*. In the system design under consideration, the control path functions are controlled by the NSC800, through the NSC810 programmable I/O port. The ideal flow of operations for the control path is shown in *Figure 12* and is self-explanatory.



TL/F/9381–16

FIGURE 12. Control Path Flow of Operations

The above discussion covered the hardware and relevant firmware considerations for designing the disk side of the disk controller subsystem. The other aspect of the disk controller subsystem design is the system side. This essentially involves the interface of the DDC to the local buffer memory and interface of this local buffer over the system bus to the host system. Discussion of the system side design follows in the next section.

4.0 CONTROLLER DESIGN—SYSTEM SIDE

4.1 System Side Hardware Considerations

The system side architecture in controller subsystems usually consists of two buses. A local bus accommodates the DDC, local microprocessor or microcontroller and the buffer memory while the remote bus is usually a standard bus like VME, STD100, MULTIBUS®, SCSI, etc, which connects the controller to the host system. The buffer memory on-board is a disk buffer which could have a maximum size of 64k. The DDC supports two 16-bit DMA channels, local and remote DMA. The local DMA transfers the data between the on-board FIFO and local buffer memory, while the remote DMA transfers data between the local buffer memory and the host over the system bus.

In the controller subsystem under consideration *(Figure 13)*, the local μP is the NSC800™. This can access the registers of the DDC in the peripheral mode and executes program code from an EPROM. All the peripheral chips are memory mapped and the 74HC138 decodes the address lines to generate the various chip selects. A PAL has been designed to provide arbitration between the DDC and the NSC800 for use of the local bus. This design was intended to provide a good exerciser for the DP8466 and hence a terminal is connected to the μP (NSC800) through a UART (NSC858). A monitor has been developed which allows exercise of the DDC in a lot of modes and interacts with the user through the terminal. Hence the remote DMA is not really used. Also in order to be able to do both byte-wide and word-wide transfers, a detection logic was implemented. Only an 8k local buffer RAM is used, accommodating 8k bytes or 8k words. LEDs are provided for visual indications of certain disk parameters.

As mentioned above, from a general design point of view, this could be easily extended to a standard system bus like VME, MULTIBUS II, etc, using the remote DMA channel of the DDC and using the local microprocessor to handle the communications protocol to the DDC and to the host operation system over the system bus.

4.2 System Side Programming Considerations

The firmware in conjunction with the microprocessor is essentially responsible for deciphering the protocol sent over the system bus and then based on the requested operations, set up the control path through the I/O ports and set up the DDC to initiate the disk operation. The driver routines to handle the control path are usually very simple and need to do the defined task as outlined in Section 3.6. The driver would consist of various blocks to implement the different ESDI commands and interpret the status reported; however, the flow would be as shown in *Figure 12*. Once the drive is positioned at the right track and ready to start the operation, control is handed over to the DDC. Before the DDC can be instructed to initiate a disk operation it has to be prepared, i.e., all registered set up to achieve the desired task.

Figure 14 shows the sequence of actions to be done with parameter RAM is initialized with the pattern and count values in accordance with the format desired. The DMA registers should be set up next starting with the LT and RT registers followed by the DMA Address Byte #, indicating start boundaries in memory and finally the sector byte count # and remote data byte count # registers. The ECC registers (preset, taps and control) are set up and the DF register is set up accordingly. Having set up the various registers, the format registers are manipulated if desired, sector counter, NSO counters initialized, and DC register loaded to initiate a disk operation. The interrupts are monitored with reads of the status register to determine the results of the disk operation. If desired, certain fancy operations like error correction cycle, interlock mode of operations, FIFO table formatting could be implemented with additional firmware.

An effort has been made in this application note to introduce the designer to the ESDI interface, and explain the intricacies of designing to its specifications with the DDC. The emphasis was on the disk side, as system side requirements may vary with different design situations. A representative design has been included which was built and tested at National (shown in *Figure 13*). For more details on the ESDI standard, refer to the official specifications provided by the ESDI committee. For more details on the DP8466A refer to AN-413 and the data sheet.

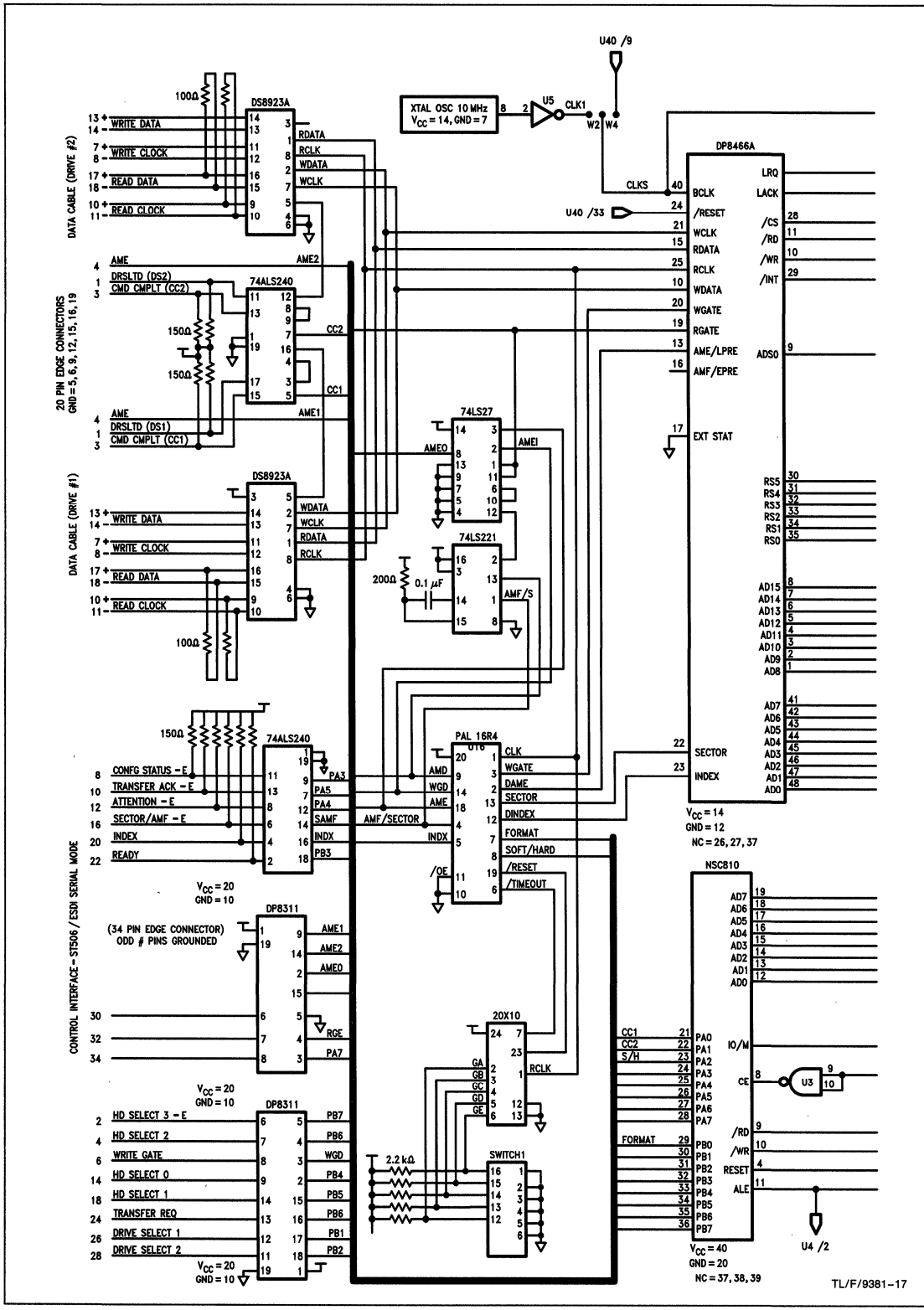
APPENDIX
Schematic of a Representative Design of an ESDI Disk Controller

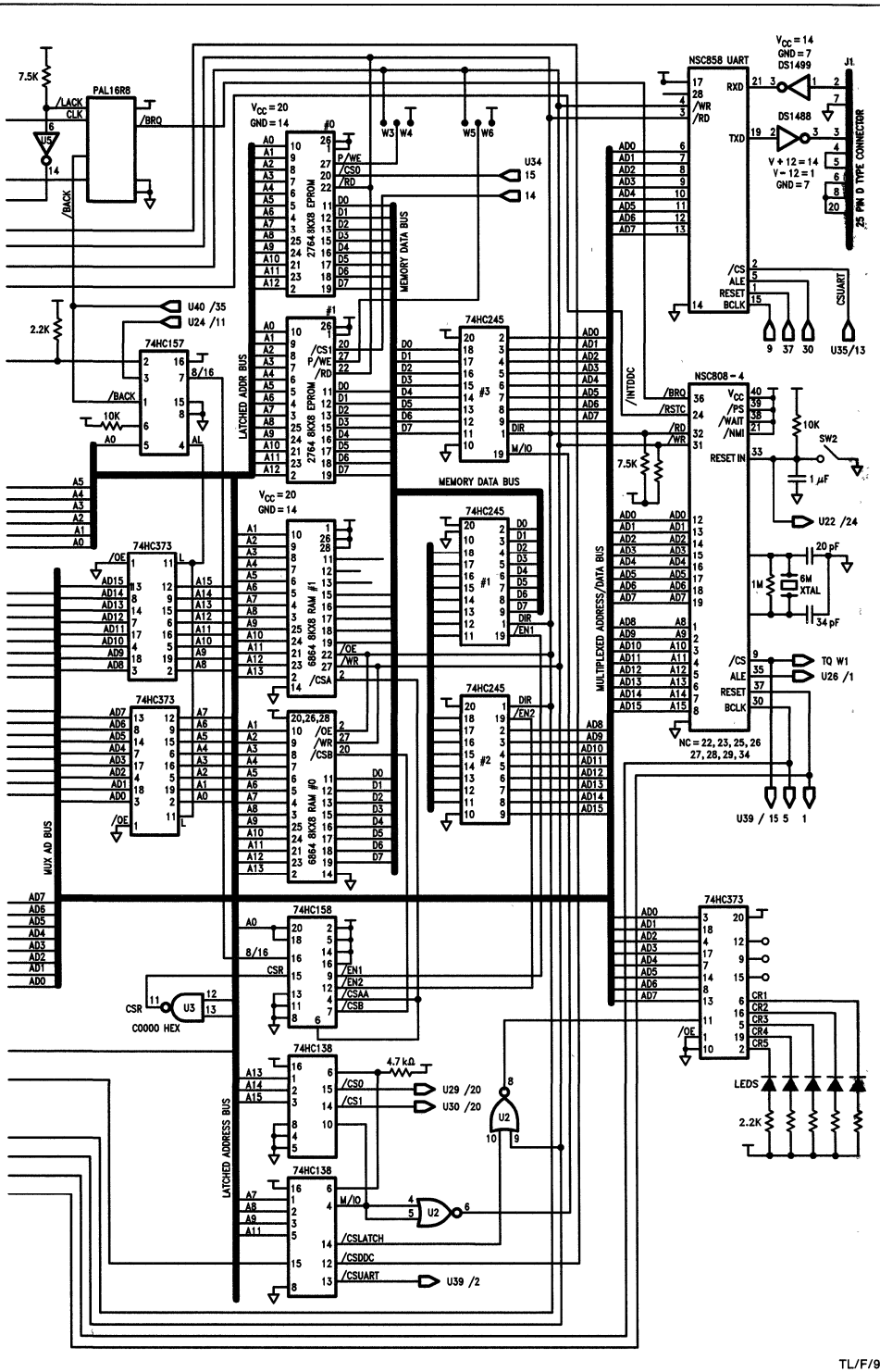
Shown in the attached schematic is the design of an ESDI disk controller system. Emphasis has been placed on the disk side of the design, as that is of foremost concern with respect to the DDC. The design incorporates the PALs as described in the previous sections, to implement the shortcomings of the DDC, in order to completely support the Enhanced Small Device Interface specifications.

The control signals are generated using the NSC810 programmable I/O port. They are connected to the "A" cable through industry standard open collector drivers, (DP8311) and receivers, (74ALS240). The receivers support a termination of 150Ω. The Data path signals are directly controlled by the DDC. They are connected to the "B" cable through industry standard differential line drivers and receivers (DS8923A). The appropriate terminations required are shown in the schematic. The design supports two drives and the appropriate data cable is enable by the drive selected line as shown.

The local microprocessor used is the NSC800, which is responsible for controlling the NSC810 to generate the appropriate control signals, and also programming the DDC to initiate the desired disk operation. This design is essentially a demonstration system and hence supports a UART (NSC858), through which the user is allowed to interact through a special monitor to initiate specific operations for the DDC. The memory design has been done so that the system could be operated in both the byte-wide transfer mode and also the word-wide transfer mode. The arbitration PAL is a simple one to arbitrate the bus between the DDC's DMA activity and the microprocessor accesses of the DDC.

It should be noted that the DDC has the capability of two 16-bit DMA channels, local and remote. The local DMA is used to transfer data between the FIFO and the local buffer memory, while the remote DMA is used to transfer data between the local buffer memory and the host system over a system bus like VME, MULTIBUS, SCSI, etc. Hence the design could easily be extended to these situations, if desired.


U40 /9
XTAL OSC 10 MHz
VCC = 14, GND = 7
U5
CLK1
W2 W4
DP8466A
CLKS
U40 /33
BCLK
/RESET
WCLK
RDATA
RCLK
WDATA
WGATE
RGATE
AME/LPRE
AMF/EPRE
EXT STAT
SECTOR
INDEX
LRQ
LACK
/CS
/RD
/WR
/INT
ADS0
RS5
RS4
RS3
RS2
RS1
RS0
AD15
AD14
AD13
AD12
AD11
AD10
AD9
AD8
AD7
AD6
AD5
AD4
AD3
AD2
AD1
AD0
VCC = 14
GND = 12
NC = 26, 27, 37
DATA CABLE (DRIVE #2)
DATA CABLE (DRIVE #1)
WRITE DATA
WRITE CLOCK
READ DATA
READ CLOCK
100Ω
DS8923A
RDATA
RCLK
WDATA
WCLK
20 PIN EDGE CONNECTORS
GND = 5, 6, 9, 12, 15, 16, 19
AME
DRSLTD (DS2)
CMD CMPLT (CC2)
DRSLTD (DS1)
CMD CMPLT (CC1)
74ALS240
150Ω
CC2
CC1
AME2
AME1
74LS27
AMEO
AMEI
74LS221
200Ω
0.1 µF
AMF/S
CONTROL INTERFACE – ST506/ESDI SERIAL MODE
CONFG STATUS – E
TRANSFER ACK – E
ATTENTION – E
SECTOR/AMF – E
INDEX
READY
PA3
PA5
PA4
SAMF
INDX
PB3
PAL 16R4
U16
CLK
WGATE
DAME
SECTOR
DINDEX
FORMAT
SOFT/HARD
/RESET
/TIMEOUT
/OE
AMD
WGD
AME
AMF/SECTOR
INDX
VCC = 20
GND = 10
DP8311
(34 PIN EDGE CONNECTOR)
ODD # PINS GROUNDED
RGE
PA7
20X10
GA
GB
GC
GD
GE
RCLK
2.2 kΩ
SWITCH1
HD SELECT 3 – E
HD SELECT 2
WRITE GATE
HD SELECT 0
HD SELECT 1
TRANSFER REQ
DRIVE SELECT 1
DRIVE SELECT 2
PB7
PB6
WGD
PB4
PB5
PB6
PB1
PB2
NSC810
CC1
CC2
S/H
FORMAT
PA0
PA1
PA2
PA3
PA4
PA5
PA6
PA7
PB0
PB1
PB2
PB3
PB4
PB5
PB6
PB7
IO/M
CE
U3
/RD
/WR
RESET
ALE
U4 /2
VCC = 40
GND = 20
NC = 37, 38, 39
TL/F/9381–17


7.5K
PAL16R8
/LACK
CLK
/BRQ
/BACK
U5
14
2.2K
U40 /35
U24 /11
74HC157
/BACK
10K
A0
8/16
AL
LATCHED ADDR BUS
A5
A4
A3
A2
A1
A0
74HC373
/OE
AD15
AD14
AD13
AD12
AD11
AD10
AD9
AD8
A15
A14
A13
A12
A11
A10
A9
A8
74HC373
AD7
AD6
AD5
AD4
AD3
AD2
AD1
AD0
/OE
MUX AD BUS
V_{CC} = 20
GND = 14
#0
#1
2764 8KX8 EPROM
P/WE
/CS0
/CS1
/RD
D0
D1
D2
D3
D4
D5
D6
D7
W3 W4
W5 W6
MEMORY DATA BUS
U34
15
14
74HC245
#3
DIR
M/IO
MEMORY DATA BUS
V_{CC} = 20
GND = 14
6864 8KX8 RAM #1
/OE
/WR
/CSA
74HC245
#1
DIR
/EN1
20,26,28
/CSB
6864 8KX8 RAM #0
74HC245
#2
DIR
/EN2
MULTIPLEXED ADDRESS/DATA BUS
74HC158
8/16
CSR
/EN1
/EN2
/CSAA
/CSB
CSR
U3
C0000 HEX
74HC138
4.7 kΩ
/CS0
/CS1
U29 /20
U30 /20
U2
LATCHED ADDRESS BUS
74HC138
M/IO
U2
/CSLATCH
/CSDDC
/CSUART
U39 /2
/INTDDC
NSC858 UART
RXD
/WR
/RD
TXD
AD0
AD1
AD2
AD3
AD4
AD5
AD6
AD7
V_{CC} = 14
GND = 7
DS1499
J1
DS1488
V + 12 = 14
V − 12 = 1
GND = 7
25 PIN D TYPE CONNECTOR
/CS
ALE
RESET
BCLK
CSUART
9 37 30 U35/13
NSC808 – 4
/BRQ
/RSTC
/RD
/WR
V_{CC}
/PS
/WAIT
/NMI
RESET IN
10K
SW2
1 µF
U22 /24
20 pF
1M
6M
XTAL
34 pF
/CS
ALE
RESET
BCLK
TQ W1
U26 /1
NC = 22, 23, 25, 26
27, 28, 29, 34
U39 / 15 5 1
74HC373
CR1
CR2
CR3
CR4
CR5
LEDS
2.2K

TL/F/9381–18

Interfacing National's DP8466A to the SMD Storage Module Device Interface Standard (Hard Sectored Drive)

National Semiconductor
Application Note 501
Ramachandran Gopalan

1.0 INTRODUCTION

With the advent of computer technology, the demand for high performance memory devices has been increasing. Before the introduction of 5¼ and 8 inch Winchester disk drives in the late 1970's, minicomputers and mainframes were the only systems that utilized rigid disks. Storage capacity, data transfer rates and to some extent cost, are the main factors which determine the performance of Winchester drives. The data transfer rate is highly dependent on the interface protocol adopted by the drive and also the electronics. The defacto industry standard drive interface for low end systems, ST506 by Seagate Technology, supports a maximum data transfer rate of 5 Mbits/sec and requires the disk controller to take care of data separation (synchronization and decoding). With the intention of supporting higher data transfer rates, newer standards were defined, like the ESDI (Enhanced Small Device Interface), which incorporates data separation on the drive itself and supports a data transfer rate of 10 Mbits/sec. The intelligent disk interface standards like SCSI and IPI, incorporate an interface to the host system with a well established high level communication protocol, however they also need to incorporate a drive-level interface like ESDI, SMD, etc.

With the intention of standardization of a common interface and to prevent product obsolescence, Control Data Corporation developed an intelligent interface called the Storage Module Device (SMD). This interface allows different drives to use the same hardware signals, even though their capacities and physical sizes were different. Variations of the SMD were also introduced. SMD started out at 9.667 Mbits/s data transfer rate and has since gone through several upgrades to a 24 Mbit/s option (SMD-E), introduced recently by Control Data Corporation. The SMD disk interface is a high quality proven attachment and is well on its way to becoming a defacto industry standard. It incorporates error recovery facilities, includes power sequencing for multiple units and is adaptable to many different rigid-disk storage units. Although it requires two very bulky and expensive cables, in high-end products this is an acceptable drawback and SMD still remains a popular choice for the higher capacity 5¼ and 8 inch Winchesters. It offers several advantages over the ST506 type interface in the high capacity arena (like parallel seek instead of serial step pulses and better status reporting), however, it is not a trivial interface when it comes down to designing a controller.

This application note looks into the definition of the SMD interface and the various design aspects of building a Disk Controller for a hard sectored drive supporting the SMD interface, using National Semiconductor's Disk Data Controller IC, DP8466A (DDC). Emphasis is laid on the disk interface design as that is of relevance to the DDC. The DDC is the most versatile LSI disk controller in the market today, offering fully programmable format features, maximum range in data rate (50 Kbits/s to 25 Mbits/s, dual DMA capability with a transfer rate of 10 Mbytes/sec, programmable error checking and correction and many other features which makes the design of a disk controller simpler and less complex. To be able to support high performance drives in the future, disk controller IC's must be capable of handling data transfer rates >20 Mbits/s. The DDC concentrates on the high speed data path signals, while the slower drive control operations are left to an inexpensive microcontroller, or the local (on board) microprocessor. The DDC in conjunction with this local intelligence, can achieve the Disk Controller function which was accomplished by 100–150 SSI/MSI integrated circuits in earlier Disk Controller systems, minimizing complexity, cost and system overhead.

2.0 SMD INTERFACE DEFINITION

The SMD disk interface standard started out as a dominant de-facto standard for 14" OEM Winchester drives and is virtually the basis of 8"-14" OEM disk drive industry today, with an eye out for the 5¼" OEM market in the future. It was approved by the ANSI committee (ANSI X3.91), in 1982. The interface consists of a 60-pin twisted pair control (A) cable and a 20-pin data (B) cable with a molded foil ground plane on one side. The 'A' cable is responsible for all head movements, status reporting and issuing commands, while the 'B' cable is used for reading and writing NRZ data. The 'A' cable assignments are shown in *Figure 1.* Address and Control functions are transferred on ten BUS OUT lines. The significance of the information on these lines is indicated by one of the six TAG functions as shown in *Figure 2.* Status for real time control, device identity and current sector status are returned on eight BUS IN lines. Drives are selected by separate UNIT SELECT lines on the 'A' cable, which have their own strobe line, UNIT SELECT TAG. All TAG lines except UNIT SELECT TAG are gated by the UNIT SELECTED signal, unit referring to the drive. The 'B' cable (Data) I/O signal assignments are shown in *Figure 1.* This cable essentially handles the transmission of data and clock information, which could be at very high transfer rates. The 'A' interface cable may be connected to the controller in a star-chained mode or daisy-chained mode. Most of the drive manufacturers support a dual channel option, where the drive could be accessed by two controllers. This is controlled by three special signals on the 'A' cable—PRIORITY SELECT, BUSY and RELEASE lines.

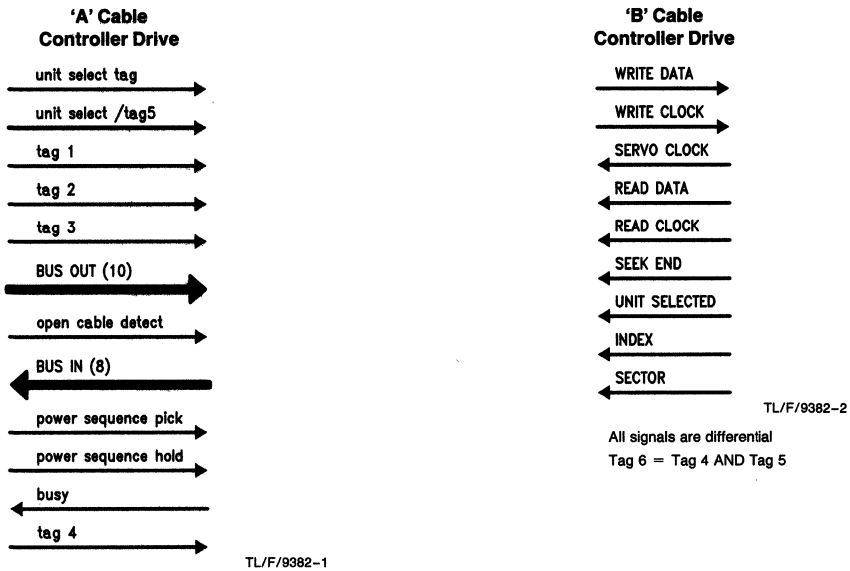


FIGURE 1. SMD Interface Cable Assignments

All input and output signals are differential in nature, and utilize industry standard transmitters and receivers. When used with properly shielded cables, this interface provides a terminated, differential transmission system for long distances, up to 50 feet, in noisy electrical environments. The maximum number of drives connected to the 'A' cable is 16. The recommended TTL differential drivers and receivers are the MC3453 and MC3550. The appropriate terminations for the driver/receiver combinations are shown in *Figure 3.* The detailed schematic in the appendix also shows them in the design. There are certain drives like CDC's 9772-XMD drive (24 MHz), where the high data rates and timing requirements at the drive necessitate the use of ECL drivers and receivers, with the appropriate terminations. The schematic in the appendix outlines the connection requirements for these with the appropriate terminations.

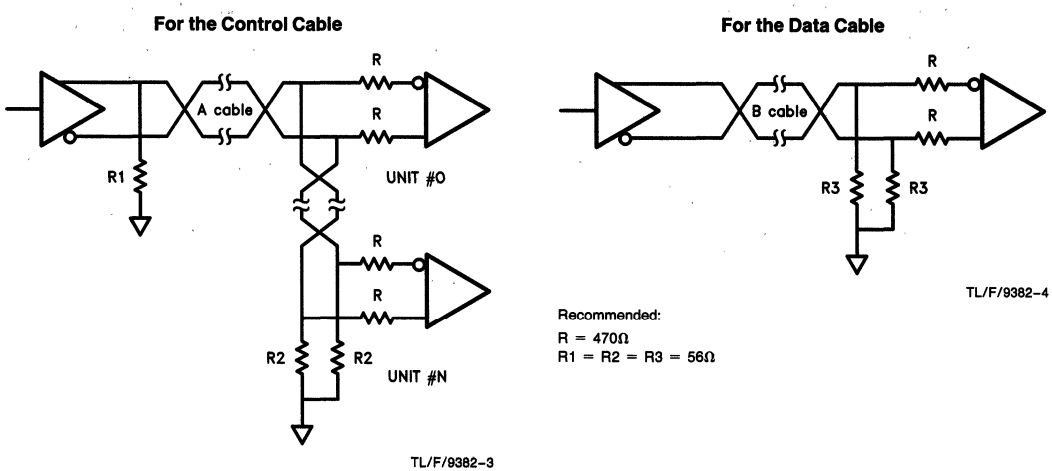
3.0 DISK SIDE INTERFACING

In the SMD interface it is fairly obvious that a lot of control is necessary to perform even a simple operation. The disk controller must perform simultaneous operations on both cables, as well as monitor status signals to determine if the command was executed properly. The disk controller board essentially provides a connection from the drive-level interface to the system. Its main functions are to handle the disk's control and data path, to transfer data between the disk and the system. The Disk Controller interfaces to the disk drive at one end and the main system at the other end. On the disk side the DDC interface to the drive can be broken down into two main paths—control path and data path. The control path essentially comprises of the 'A' cable signals on the SMD interface. These are differential in nature and require differential drivers/receivers to drive the cable between the chip and the interface connectors. The local μP has to activate and monitor the control lines in a certain sequence to achieve a desired operation, defined by the SMD protocol. The interface is activated by asserting the OPEN CABLE DETECT signal. This signal is sent through two drivers paralleled and the regular termination resistor is omitted. The first task then is to select the drive using the UNIT SELECT lines, latching them with the UNIT SELECT TAG. UNIT SELECTED is asserted by the drive indicating the selection of the drive. The unit is checked to see if it is ready for the next operation by monitoring the READY line. Once ready, the head and cylinder addresses are provided to place the drive's head assembly at the desired position on the media, and then the desired sector of data is sought. On arriving at the desired sector, the read/write operation is performed. In case of an unsuccessful operation, an error condition is flagged which could be determined by reading the status and monitoring other signal lines. *Figure 4* gives a detailed flow chart of the sequence of drive control operations performed by the local μP for the SMD interface standard. The status information presented on the interface is a function of the various TAG lines as seen in *Figure 2.*

Bus Out Bit	Unit Select Tag	Tag1 (cyl se.l)	Tag2 (head and upp cyl sel)	Tag3 (control sel)	Tag4 (curr sector)	Tag5 (extnd status)				Tag6 (Device Type)
0		2^0	2^0	Write Gate	0	0	1	0	1	0
1		2^1	2^1	Read Gate	0	0	0	1	1	0
2		2^2	2^2	Servo Offset +	0	0	0	0	0	0
3		2^3	2^3	Servo Offset −	0	0	0	0	0	0
4		2^4	2^4	Fault Clear	0	0	0	0	0	0
5		2^5		AME	0	0	0	0	0	0
6		2^6		Return to Zero	0	0	0	0	0	0
7		2^7	2^{10}	dztz str (Early)	0	0	0	0	0	0
8		2^8	2^{11}	dztz str (Late)	0	0	0	0	0	0
9		2^9		Release	0	0	0	0	0	0
						①	②	③	④	

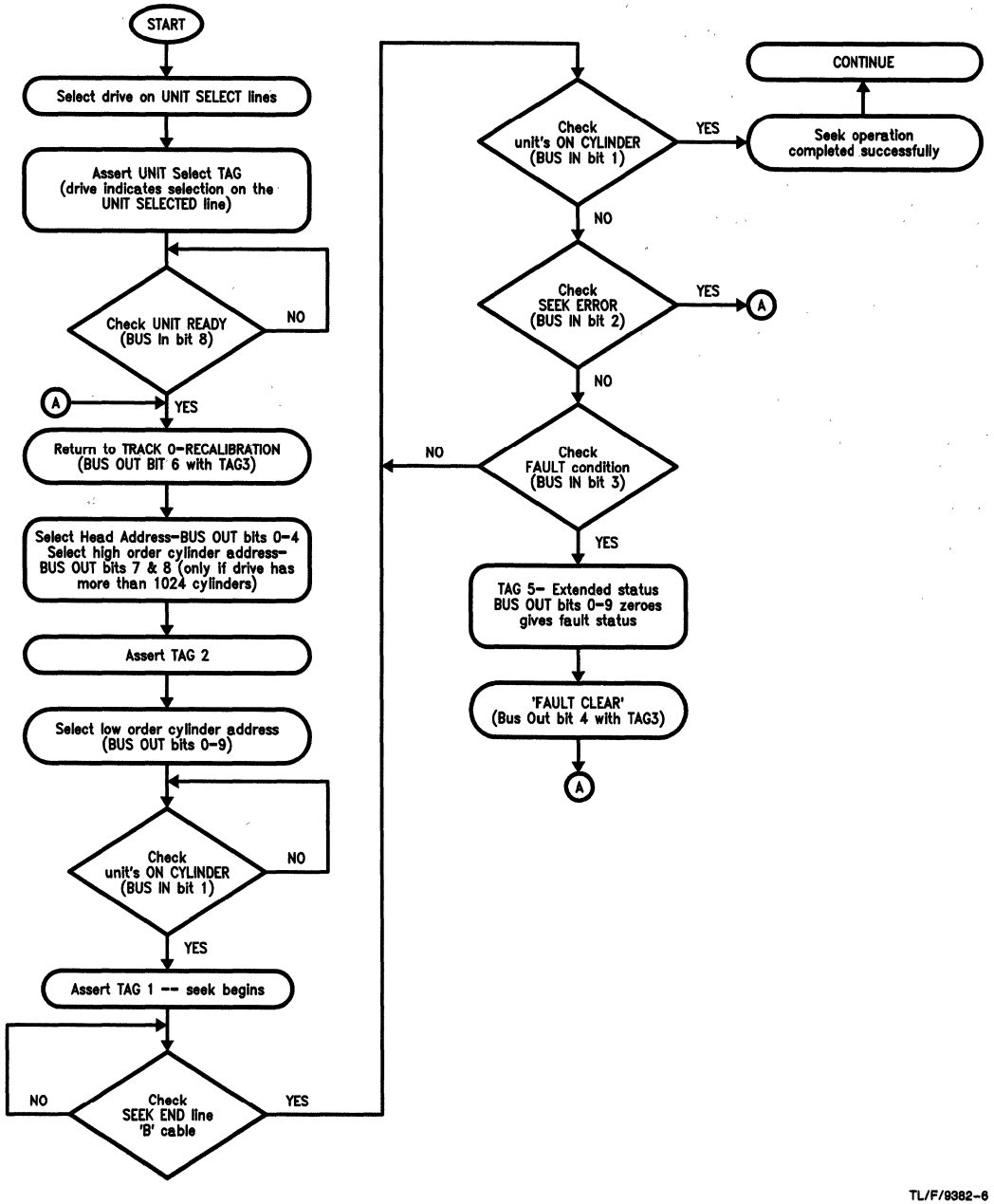
Bus in Bits	Drive Status	curr sector cnt	Fault Status①	Operating Status②	Diagnostic Status③	Diagnostic Execute Status④	Device Type Status (User Defined)
0	Unit Ready	2^0	RD * WR				
1	On Cylinder	2^1	(RD + WR) * oncyl				
2	Seek Error	2^2	First Seek				
3	Fault	2^3	Write				
4	WR Protect	2^4	WR * WR Protect				
5	AMF	2^5	Head Select				
6	Index Mark	2^6	Voltage				
7	Sector Mark	2^7	Valid Status Available			tst	

FIGURE 2. Tag Bus Decode Information Available on the 'A' Cable



TL/F/9382-3

FIGURE 3. Driver/Receiver Combinations with the Appropriate Terminations for 'A' and 'B' Cable



TL/F/9382-6

FIGURE 4. Control Path—Flow of Operations

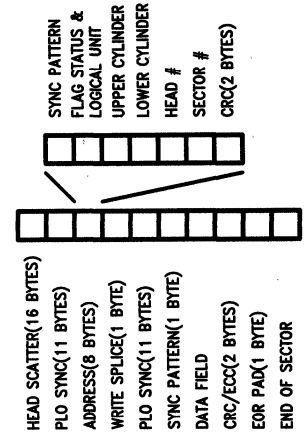
The other component of the Disk Controller system is the data path. The SMD 'B' cable essentially contains the data path signals to the drive. It consists of lines to transmit serial NTZ data to and from the drive. Associated with the read/write lines are the clocks: Read Clock for reading data, Write Clock for writing data and a Servo Clock synchronous with the rotation of the spindle, for reference. Additional signals help in determining the status of each drive on the bus. The main component of the data path in a disk controller system is the Disk Data Controller IC, like the DP8466A. The DDC can be programmed to operate in accordance to the SMD interface specification for the associated operations like formatting, reading and writing. Read Gate and Write Gate are the two main read/write control signals in the data path. Write Gate enables the write operation and is validated only when UNIT READY, ON CYLINDER, SEEK END are false. If Write Gate is asserted under any other conditions, a fault occurs and writing is inhibited. There are also certain drive dependent constraints which affect the read/write timing and must be taken care of. Listed below are some of them. Details can be found in the SMD specification.

1. write circuit turn-on delay
2. head select transients
3. read after write transients
4. read/write—encoding/decoding delays
5. write after read transients
6. PLL synchronization time

The DDC supports most of the specifications of the SMD standard, however there are certain specifications that the DDC does not directly support. These can be taken care of through software or with the help of external logic. A discussion of these considerations is given in the following sections.

4.0 HANDLING THE SMD RECOMMENDED FORMAT AND INTERFACE SPECIFICATIONS

Figure 5 shows the recommended format for SMD drives. There are certain fields in the format which are not directly supported by the DDC, which however can be implemented as discussed below.



TL/F/9382–7

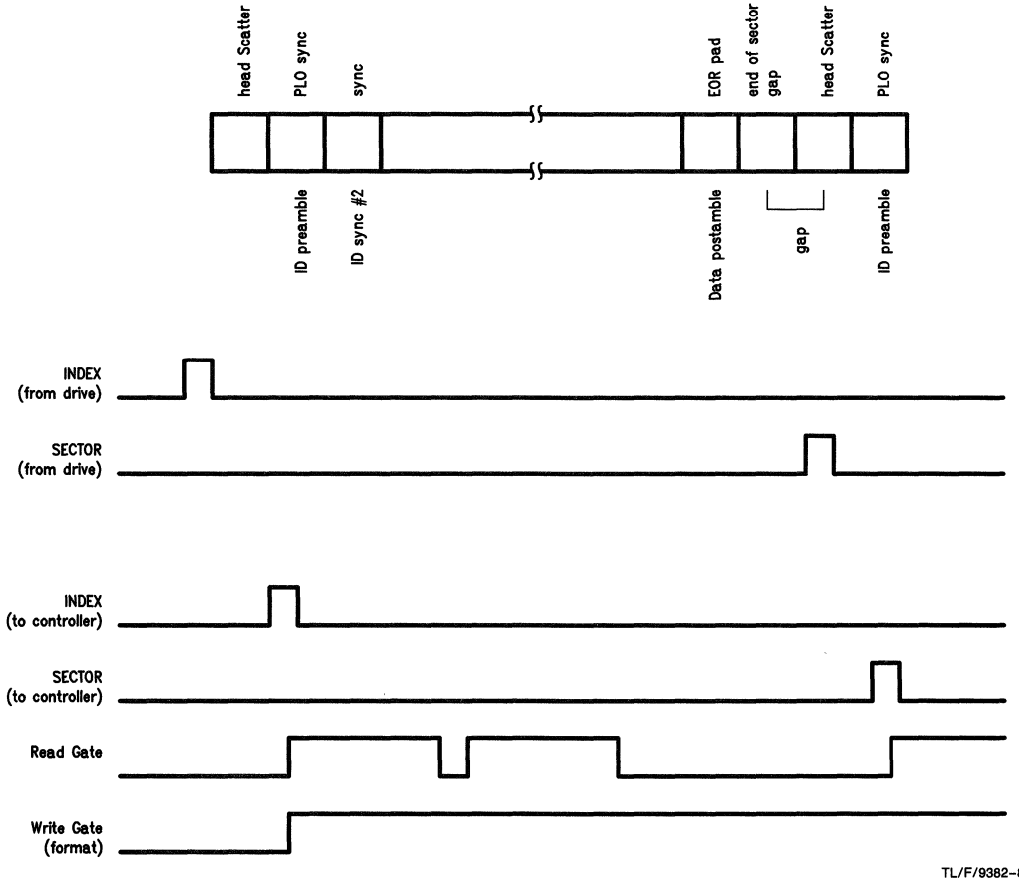
FIGURE 5. SMD Recommended Format (Hardsectored)

4.1 The Write Splice Field

The SMD format specification recommends a zero byte header postamble, a one byte write splice field and an eleven byte data preamble, (PLO sync field). The DDC does not support a separate register for the write splice field between the header and the data segments in its format parameter RAM. However this can be easily taken care of by the programmable format feature of the DDC. The format parameter registers are programmed to have a 12 byte data preamble, one byte longer than the desired length. This byte is considered for practical purposes as the write splice. Hence when formatted this will have the same pattern as the data preamble. During a write operation, this byte is kept as part of the data preamble and with the standard requirement of 8 clocks write propagation delay in the drive (due to write circuitry), the data gets written on the media beyond the write splice field, at the correct place in the preamble. In case of a read operation, this byte is considered to be part of the header postamble. Since the DDC asserts read gate 11.5 bits into the data preamble, it will never be asserted in the write splice. In this manner the write splice field can be implemented to support the recommended SMD format specification. It should be noted however that it is mandatory to use at least a one byte header postamble with the DDC for proper operation.

4.2 The Post Index/Sector Gap Field (Head Scatter)

The recommended format in the SMD standard supports a gap felled after the index/sector pulse, referred to as the post index/sector gap or head scatter. This is necessary mainly to accommodate drive dependent transients as mentioned earlier. The DP8466A does not support a separate field for this gap and also uses the index/sector pulse as a reference for the read/write control signals. To implement this gap field, an external counter is used to delay the index/sector pulse from the drive, by the desired gap count, before presentation to the DDC. The circuitry to achieve this is shown in the schematic in Appendix A. Since the index/sector pulse is presented to the DDC delayed by the length of the Head Scatter field, at the time it starts writing the PLO field etc., it's at the right area on the media. The gap field at the end of the sector is written till a sector pulse is received by the DDC. Hence the gap pattern gets written for the Head Scatter field of the next sector. *Figure 6* shows the details of this technique and the manipulation of the parameter fields.



TL/F/9382–8

The disk controller writes the end of sector gap over the head scatter area, hence achieving the writing of the post index/sector gap. The last sector overlaps to write the head scatter for the first sector.

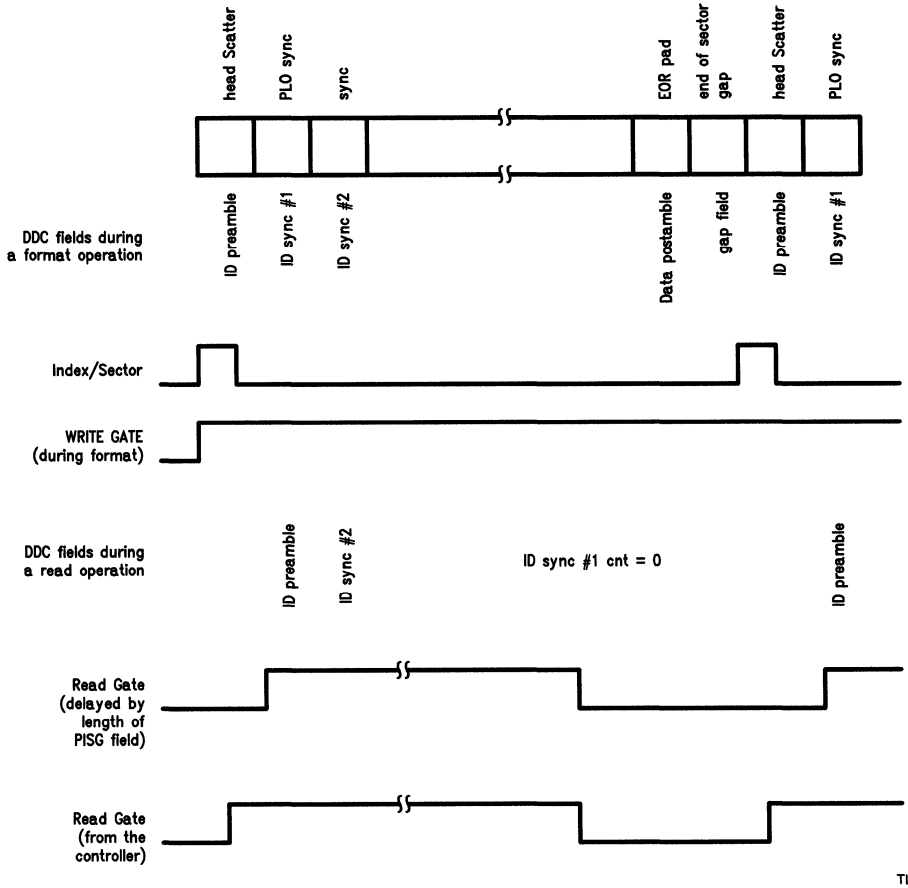
FIGURE 6. PISG Implementation (Technique 1)

4.3 Read/Write Gate Timing Restrictions

On assertion of Write Gate, there is a write splice before data is actually written on the media. Hence when reading the data, it must be ensured that Read Gate is asserted sometime after the write splice. In the DDC the delay between Write and Read Gate in the Data field is 8.5 bit times, which satisfactorily covers the write splice. However at the beginning of the sector it is only 0.5 bit times, hence Read Gate would have to be delayed to prevent assertion of the Read Gate during the write splice. This would be a problem only in the first sector after the index pulse, as during a format operation Write Gate remains asserted till the Index pulse is encountered again. This is done by delaying the Read Gate by 8 bits from the DDC to the drive, using a counter as shown in the schematic in the appendix. This results in the read gate assertion delayed by a byte even in the data field which does not make any difference to the performance.

4.4 Alternative Technique to Implement the Shortcomings in Section 4.2 and 4.3

An alternative technique to implement the Post index/sector gap and the problem of asserting read gate in splice at the beginning of the sector is to have a single delay circuit which delays the read gate qualified by index/sector, assertion to the drive by the length of this gap. The manipulations of the DDC fields to achieve this is shown in *Figure 7.*



TL/F/9382–9

Note: Read Gate assertion needs to be delayed only at the index or sector pulse.

FIGURE 7. PISG Implementation (Technique 2)

4.5 MUXing the Two Clocks on the SMD INTERFACE 'B' CABLE

The 'B' cable has two clock outputs— the SERVO CLOCK and the READ CLOCK. The SERVO CLOCK signal is a phase locked clock, (frequency dependent on the drive), generated from the servo track bits and is available at all times. The READ CLOCK signal defines the beginning of a data cell and is synchronous with the detected data. This signal is derived from the SERVO CLOCK. At the start of a read operation, READ GATE is asserted by the controller. This initiates the PLL on the drive to begin locking on the data from the media. Till this point, the clock sent to the controller is the reference clock (similar to the SERVO CLOCK frequency). When the PLL achieves phase lock, the clock transmitted on the READ CLOCK line to the controller is the one in phase sync with the data. An undefined clock may be transmitted at the point of obtaining phase sync, upon initiating or ceasing a read operation. Read Clock is in phase sync within 2.5 μs after Read Gate is asserted, (worst case). Also the WRITE CLOCK generated by the DP8466A is essentially the SERVO CLOCK retransmitted to the drive synchronized to the NRZ Write Data. The DP8466A has only one clock input line (READ CLOCK). Hence the switching of the SERVO or READ CLOCK to the DP8466A must be done externally. Also since the DP8466A's clock input cannot accommodate short pulses, this switching must be done without short pulses. The external circuitry to multiplex the two clocks and the deglitcher is shown in the schematic. This can safely operate up to clock frequencies of 25 MHz. To take care of the undefined pulse occurring on obtaining phase sync, the Read Gate signal delayed by 3 μs (to accommodate worst case lock time), is used as the 'switch' control input.

4.5.1 GLITCHLESS CLOCK MULTIPLEXER CIRCUIT

With Read Gate active (high), the source of the 8466 Read Clock is the Read Clock input signal. When Read Gate is inactive, the 8466 Read Clock will come from the Servo Clock input. If the switching between the two clocks would result in a less-than-normal-width pulse in a simple multiplexer, this glitchless circuit allows the currently active clock to finish a full one or two clock pulses before the output goes low and waits one or two clock times until the new clock appears at the "8466 Read Clock" output. So there may be one or two missing pulses but there will never be a "glitch" (narrow pulse width).

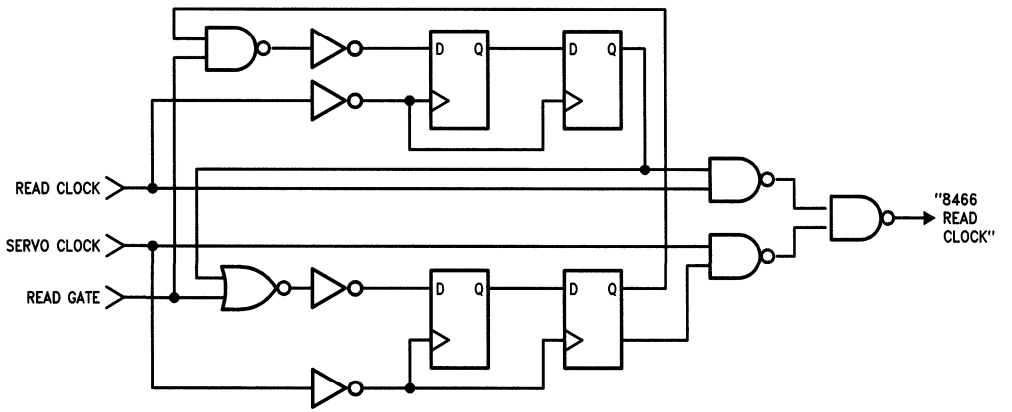
If the disk system also uses the National Semiconductor DP8463B (2, 7) RLL ENDEC, no external circuitry is required as this glitchless multiplexer is already incorporated into the DP8463B.

5.0 SYSTEM CONSIDERATIONS AND CONCLUSIONS

The DP8466A supports a dual channel onboard DMA controller which simplifies the task of interfacing to any system bus. System design considerations are flexible to the designer as outlined in the Application Note: "Designing an ESDI Disk Controller system with National's DP8466A" and AN-413. An effort has been made in this application note to introduce the designer to the SMD interface, and explain the intricacies of designing to its specifications with the DP8466A. The emphasis was on the disk side design and a representative design of the disk side is included in the appendix. This is the disk side for a disk controller designed for the 25 MHz XMD drive from Control Data Corporation. The board was built and tested in the Labs at National and was found to perform satisfactorily. For more details on the SMD standard, refer to the ANSI document on SMD interface specifications. For more details on the DP8466A refer to AN-413 and the datasheet.

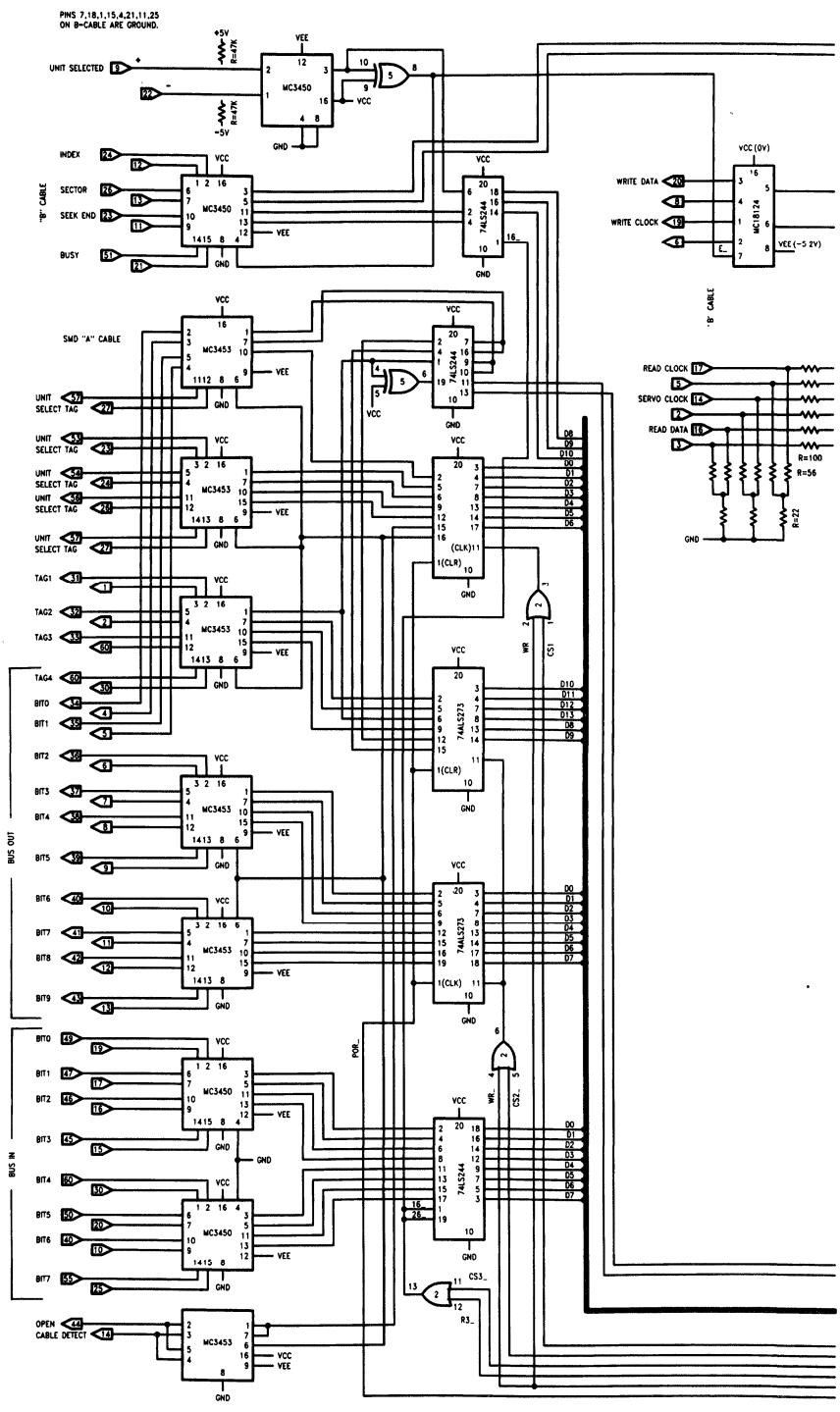
GLITCHLESS MULTIPLEXER

Block Diagram



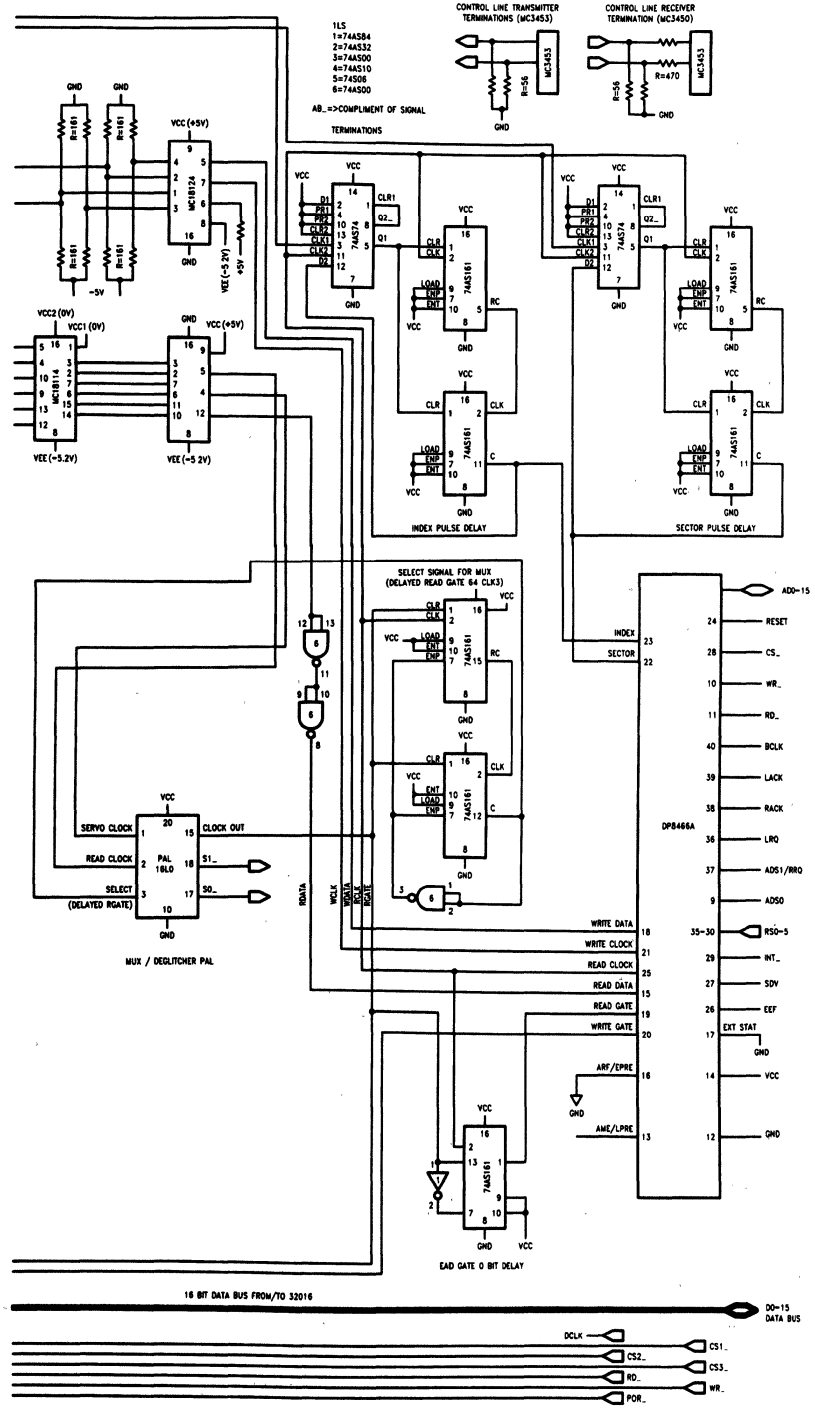
TL/F/9382–10

FIGURE 8. Glitchless Multiplexer



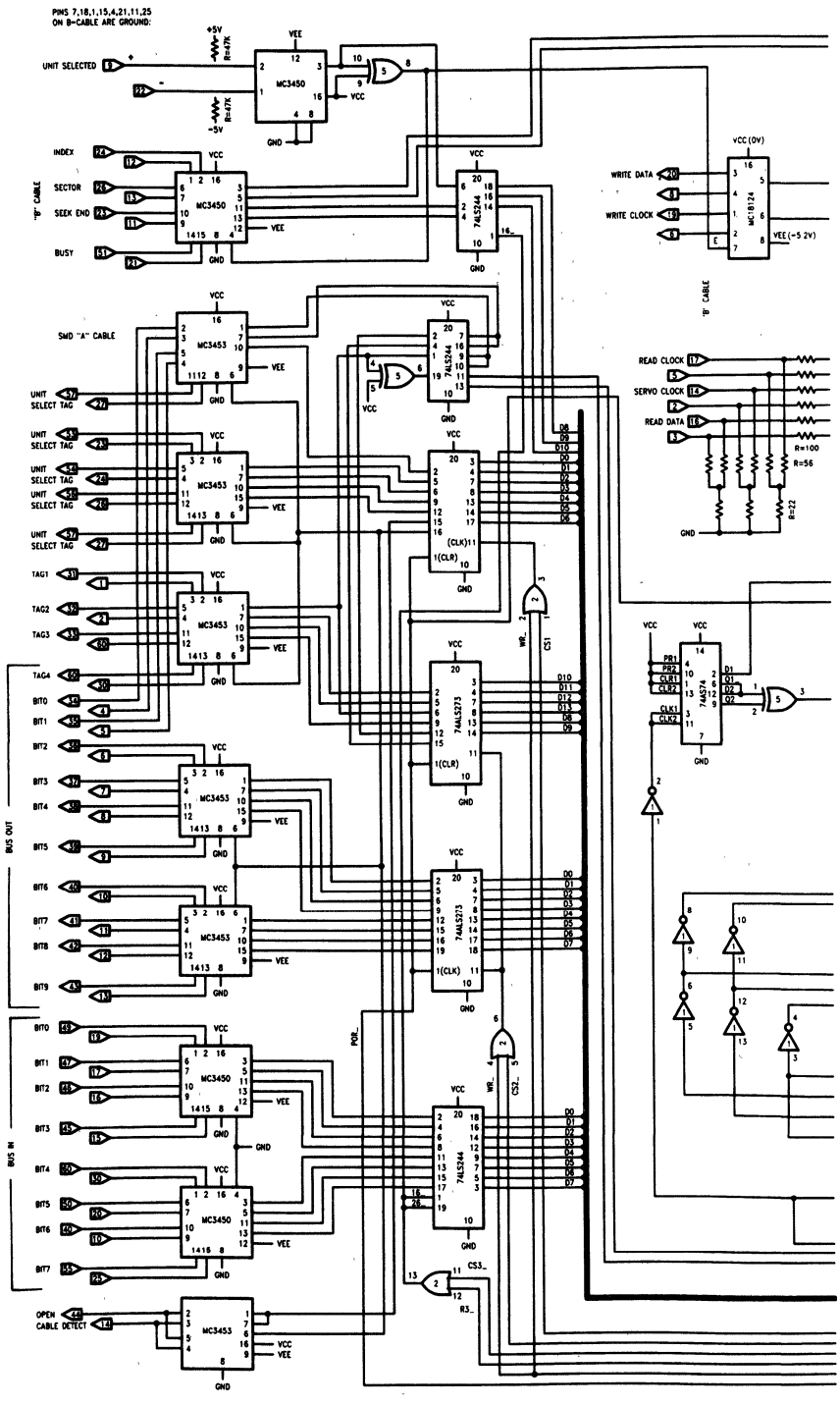
TL/F/9382–13

FIGURE 9



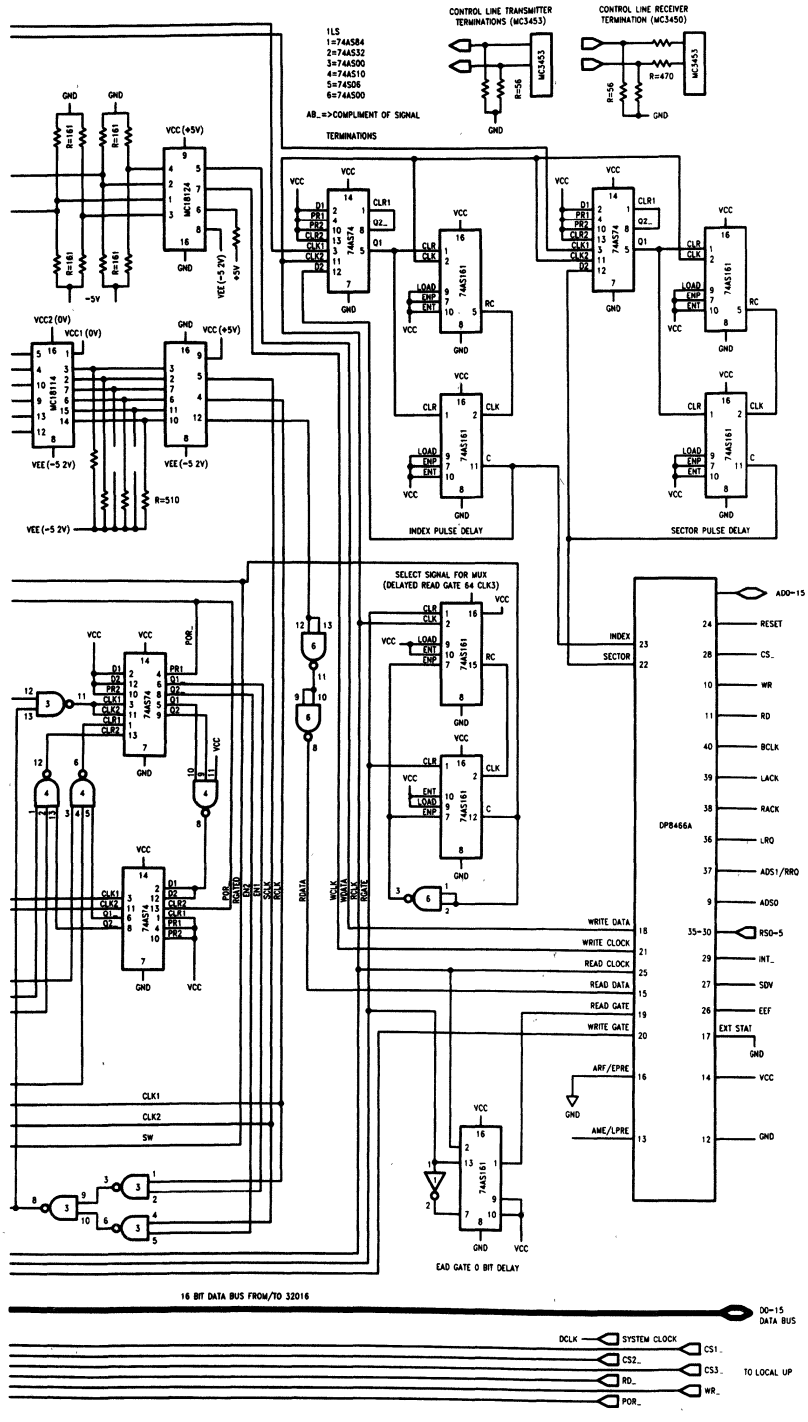
TL/F/9382-14

FIGURE 9 (Continued)



TL/F/9382–15

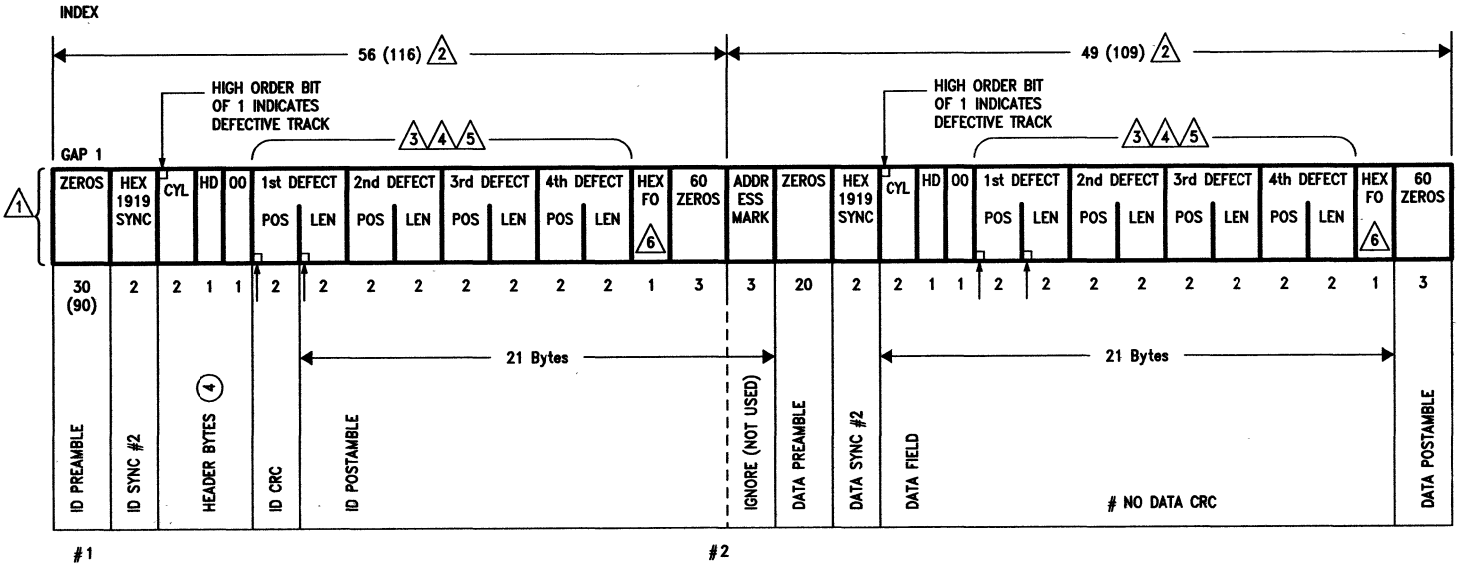
FIGURE 10



TL/F/9382-16

FIGURE 10 (Continued)

Defect List Format (SMD Drives)



TL/F/9382–12

Notes:

No data CRC is selected, OPERATION is IGNORE HEADER—READ DATA

If in case of a defect, so that #1 field becomes 90 and #2 field becomes 60 then it poses no complications for reading the defect information. However if the defect format is to be written again then it might cause complications as the max. byte count for a field is 32 bytes, hence cannot be done unless delayed physically.

Under no conditions can sync #1 be used with a pattern of zeroes. Its count must be kept zero.

Note that this technique to enable the DP8466 to be able to read the defect list format is possible only if written in two parts originally. HITACHI, NEC and CDC support the above but *not FUJITSU.*

Section 4

SCSI Bus Interface Circuits

Section 4 Contents

The DP8490 and DP5380 Comparison Guide

National Semiconductor
Application Brief 39
Desmond Young

OVERVIEW

National Semiconductor has released two products to support the Small Computer System Interface (SCSI). These two products offer significant features over existing asynchronous SCSI devices such as:

- Lower power (25 mW vs 700 mW)
- Lower cost
- Higher speed (up to 4 Mbytes/sec)
- More functionality
- Compatibility (with existing NCR5380)

FEATURES OF NATIONAL's DP8490 & DP5380

Low Power

The DP8490 Enhanced Asynchronous SCSI Interface (EASI) and the DP5380 Asynchronous SCSI Interface (ASI) are CMOS devices. The current drawn by the NCR5380 is 140 mA while both the EASI and the ASI draw 4 mA max—*and* the EASI/ASI are specified with SCSI pins terminated with SCSI spec values!

Low Cost

National's marketing strategy will ensure the EASI and ASI are the most competitively priced parts on the market.

High Speed

The high speed CMOS process used for the EASI and ASI means that DMA rates of up to 4 Mbytes per second are achievable. Currently there are no low cost high speed 8-bit DMA controllers that will provide the necessary speed. A common solution is to provide the simple DMA functionality as part of ASIC circuitry normally found in each application. National has published an application note on how to build a 4 Mbytes/sec asynchronous SCSI interface using the EASI/ASI devices to assist designers.

The user should be aware that with this increase in speed comes the need and care on the user's part in board design and layout. The EASI and ASI have reduced: read access times from 140 ns to 50 ns, write data hold times from 30 ns to 10 ns, and all other parameters correspondingly. This means the designer should ensure he has correctly generated chip strobes. In particular pay attention to the generation of DACK, and the period of hand-over of the bus between the DMA and CPU.

Improved Functionality

The EASI provides new and enhanced functions to ease the firmware overhead. The major features are:

new interrupts
new interrupt status and mask structure
extended arbitration
loopback and other testing facilities
microprocessor data bus parity
improved SCSI and DMA timing

New Interrupts

The EASI adds interrupts for: arbitration complete, any phase mismatch, true end of DMA and microprocessor parity error. The arbitration complete interrupt allows the firmware to program the EASI to arbitrate for the bus and interrupt when done. True End of DMA interrupt overcomes the fault in the DP5380 when using it to *send* data on SCSI.

New Interrupt Structure

The DP5380 cannot be fully interrupt driven (e.g., no arbitration complete interrupt)—the DP8490 can be! In the DP5380 the firmware must read 2 registers and then process the results to determine the cause of an interrupt. Not all interrupt conditions have a status bit in the DP5380. The DP8490 provides an interrupt status register which gathers all status into one register, with a dedicated status bit for each interrupt. An interrupt mask register provides individual masking control of each interrupt. Finally, the design of the interrupt logic ensures interrupts will not be lost (they can be on the DP5380).

Extended Arbitration

The DP5380 device must be polled by the μP during arbitration. Since this process does not start until the bus is free, the μP may be polling for many milliseconds. The DP8490 provides an interrupt on arbitration complete to allow the μP to handle other tasks such as cacheing data, overlapped seeks etc. The DP8490 also implements the 2.2 μs SCSI arbitration delay before interrupting the μP.

Loopback Testing

The DP8490 provides a loopback test mode. Each SCSI pin driver is disabled and looped back internally. The firmware may exercise the EASI to verify correct part operation. SCSI signal driving and monitoring, interrupts and DMA can all be tested. The EASI also provides programmable parity polarity (EVEN/ODD) for both μP and SCSI buses. These bits may be used to verify parity circuitry on all boards in a SCSI system.

Microprocessor Parity

The PCC-packaged DP8490 provides an extra pin for μP data bus parity. This enables checked data to be maintained throughout a system—even including controller buffer memory.

Improved Timing

The DP5380 has some timing anomalies just like the NCR5380 on the SCSI and μP interfaces. The DP8490 fixes these.

COMPATIBILITY

The DP5380 is *completely* compatible with existing NCR5380 type devices—but offers higher speed and lower power.

The DP8490 is also completely compatible with existing NCR5380 type devices except for one register bit. An unused "test-mode" bit in the NCR5380 is replaced by an "Enhanced Mode" bit in the DP8490. Until this bit is set the DP8490 functions as a DP5380. Once set the enhanced features of the DP8490 are accessible. Since the DP8490 powers up in DP5380 mode *all DP5380 sockets should be able to use a DP8490!*

For more information see Section 8 of the DP8490 datasheet.

National Semiconductor

DP8490 Enhanced Asynchronous SCSI Interface (EASI)

General Description

The DP8490 EASI is a CMOS device designed to provide a low cost, high performance Small Computer Systems Interface. It complies with the ANS X3.131-1986 SCSI standard as defined by the ANSI X3T9.2 committee. It can act as both INITIATOR and TARGET, making it suitable for any application. The EASI supports selection, reselection, arbitration and all other bus phases. High-current open-drain drivers on chip reduce application chip count by interfacing direct to the SCSI bus. An on-chip oscillator provides all timing delays.

The DP8490 is pin and program compatible with the NMOS NCR5380 and CMOS DP5380 devices. NCR5380, DP5380 or AM5380 applications should be able to use it with no changes to hardware or software. The DP8490 includes new features which make this part more attractive for new designs and performance upgrades. These new features include μP data bus parity, programmable parity for both SCSI and μP busses, loopback test mode, improved arbitration support, faster timing and extended interrupt control logic. The DP8490 is available in a 40-pin DIP or a 44-pin PCC.

The EASI is intended to be used in a microprocessor based application, and achieves maximum performance with a DMA controller. The device is controlled by reading and writing several internal registers. A standard non-multiplexed address and data bus easily fits any μP environment.

Data transfers can be performed by programmed-I/O, pseudo-DMA or via a DMA controller. The EASI easily interfaces to a DMA controller using normal or Block Mode. The EASI can be used in either a polled or interrupt-driven environment. The EASI includes enhanced features for interrupt control.

Features

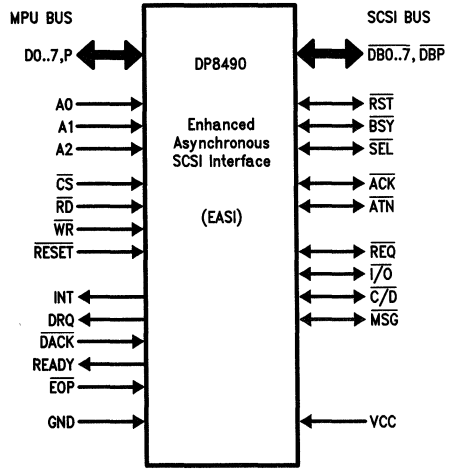
SCSI Interface

- Supports TARGET and INITIATOR roles
- Parity generation with optional checking
- Programmable parity polarity (ODD/EVEN)
- Arbitration support—can interrupt when done
- Direct control/monitoring of all SCSI signals
- High current outputs drive SCSI bus directly
- Faster and improved timing
- Very low SCSI bus loading

μP Interface

- Memory or I/O-mapped control transfers
- Programmed-I/O or DMA data transfers
- Normal or Block-mode DMA
- Fast DMA handshake timing
- Individually maskable interrupts
- Active interrupts identified in one register
- Optional data bus parity generation/checking
- Programmable parity polarity (ODD/EVEN)
- Loopback test mode

Connection Diagram



TL/F/9387–1

Table of Contents

1.0 Functional Description

1.1 OVERVIEW

The EASI is designed to be used as a peripheral device in a μP-based application and appears as a number of read/write registers. Write registers are programmed to select desired functions. Status registers provide indication of operating conditions. In an application extensive use of interrupts is desirable. The EASI incorporates an improved interrupt structure which enables fully interrupt-driven operation. In the enhanced mode interrupts can be individually masked or enabled, and a status register identifies all active interrupt requests.

For best performance a DMA controller can be easily interfaced directly to the EASI. The EASI provides request/acknowledge and wait-state signals for the DMA interface.

The SCSI bus is easily controlled via the EASI registers. Any bus signal may be asserted or deasserted via a bit in the appropriate register, and the state of every signal is available by reading registers. This direct control over SCSI signals allows the user to implement all or part of the protocol in firmware. The EASI provides hardware support for much of the protocol, and all speed-critical steps are handled by the EASI.

The EASI provides the following SCSI support:

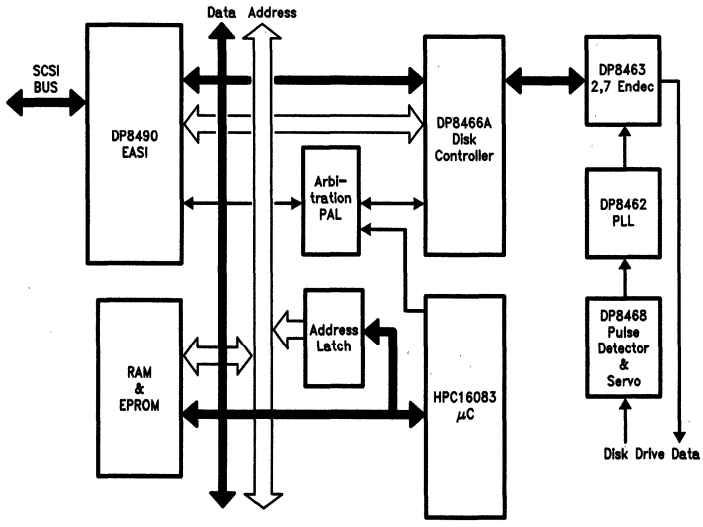
- Programmed-I/O transfers for all eight information transfer types, with or without parity.
- Data transfers via DMA, in either block or non-block mode. The DMA interface supports most devices.
- Individual setting/resetting and monitoring of every SCSI bus signal.
- Automatic release of the bus for BSY loss from a TARGET, SCSI RST, and lost arbitration.
- Automatic bus arbitration with an optional interrupt upon completion—the μP has only to check for highest priority. The 2.2 μs arbitration delay can be optionally performed by the EASI.
- Selection or Reselection of any bus device. The EASI will respond to both Selection and Reselection.
- Optional automatic monitoring of the $\overline{\text{BSY}}$ signal from a TARGET with an interrupt after releasing control of the bus.
- Optional parity polarity selection. Default after reset is ODD, but EVEN generation and checking can be programmed for diagnostic purposes and to determine whether a device supports parity when first making a connection.

Figure 1 shows an EASI in a typical application, a low cost embedded SCSI disk controller. In this application the 8051 single-chip μP acts as the controller and the dual DMA channels in the DP8475 allow one for the disk data and the other for SCSI data. The PAL provides chip selection as well as determining who has control of the bus. The advantage of using a μP with on-board ROM is that there is more free time on the external bus.

1.2 μP INTERFACE

Figure 2 shows a block diagram of the EASI. Key blocks within the EASI are Read/Write registers with associated decode and control logic, interrupt and DMA logic, SCSI bus arbitration logic, SCSI drivers/receivers with parity and the SCSI data input and output registers. The EASI has three interfaces, one to SCSI, one to a DMA controller and the third to a μP. The internal registers control all operation of the EASI.

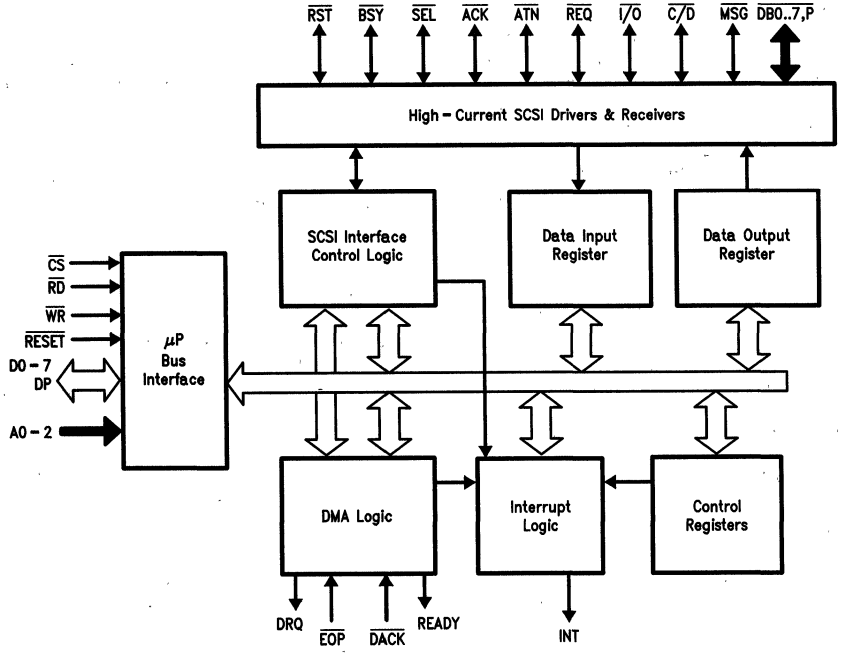
The μP interface consists of non-multiplexed address and data busses with associated control signals. The data bus can be programmed to use either ODD or EVEN parity. Address decode logic selects a register for reading or writing. The address lines A0–A2 select the register for μP accesses while for DMA accesses the address lines are ignored. The decode logic also selects different registers or functions to be mapped into address 7, according to the programmed mode (see Section 8).



TL/F/9387–2

FIGURE 1. EASI Application

1.0 Functional Description (Continued)



TL/F/9387–3

FIGURE 2. EASI Block Diagram

The register bank consists of twelve registers mapped into an address space of eight locations. Upon an external chip reset the registers are cleared (all zeroes)—the same as the NCR5380. Once the ENHANCED MODE bit in the INITIATOR COMMAND REGISTER is set, three new registers can be accessed to utilize the extra features of the DP8490 EASI.

1.3 DMA INTERFACE

The DMA logic interfaces to single-cycle, block mode, flow-through or fly-by controllers. Single byte transfers are accomplished via the DRQ/$\overline{\text{DACK}}$ handshake signals. Block mode transfers use the READY output to control the speed (insert wait-states). An End Of Process ($\overline{\text{EOP}}$) input from the DMA controller signals the EASI to halt DMA transfers. An interrupt can be generated for DMA completion or an error (see Section 5). All DMA data passes through the SCSI data input and output registers, automatically selected during DMA cycles.

1.4 SCSI INTERFACE

The EASI contains all logic required to interface directly to the SCSI bus. Direct control and monitoring of all SCSI signals is provided. The state of each SCSI signal may be determined by reading a register which continuously reflects the state of the bus. Each signal may be asserted by writing a ONE to the appropriate bit.

The EASI includes logic to automatically handle SCSI timing sequences too fast for µP control. In particular there is hardware support for DMA transfers, bus arbitration, selection/reselection, bus phase monitoring, $\overline{\text{BSY}}$ monitoring for bus disconnection, bus reset and parity generation and checking.

The EASI arbitration logic controls arbitration for use of the SCSI bus. The µP programs the SCSI device ID into the EASI, then sets the ARBITRATE bit. The EASI will interrupt the µP when one of three events occurs: arbitration is lost; arbitration has completed and the ID priorities need to be checked; or arbitration is complete and the 2.2 µs SCSI Arbitration delay has expired. Arbitration can be invoked with the enhanced feature of an interrupt on completion or the expiration of the SCSI Arbitration delay. These extra steps are programmed via the EXTRA MODE REGISTER (EMR). The INITIATOR COMMAND REGISTER (ICR) is read to determine whether arbitration has been won or lost.

The $\overline{\text{BSY}}$ signal is continuously monitored to detect bus disconnection and bus free phases. The EASI incorporates an on-board oscillator to determine Bus Settle, Bus Free and Arbitration Delays. The oscillator tolerance guarantees all timing to be within the SCSI specification.

The EASI incorporates high-current drivers and SCHMITT trigger receivers for interfacing directly to the SCSI bus. This feature reduces the chip count of any SCSI application. The driver/receivers also incorporate loopback logic which is enabled by an EMR bit. The Loopback mode enables testing of all EASI functions without interfering with the SCSI bus.

1.5 PARITY

The EASI provides for parity protection on both the µP and SCSI interfaces. Each data bus has eight data bits and one parity bit (only the PCC part provides µP parity, both the DIP and PCC provide SCSI parity). In each case the parity may

1.0 Functional Description (Continued)

be enabled via a register bit. A parity error can be programmed to cause an interrupt. Additionally the parity may be programmed to be either ODD or EVEN. This has a particular use on the SCSI interface where programming EVEN parity allows diagnostics, or determining whether a device supports parity. The inclusion of μP parity allows development of controllers that maintain data integrity right from the media to the host system.

1.6 INTERRUPTS

The EASI is intended to be used in an interrupt-driven environment. Each function can be programmed to cause an interrupt. In ENHANCED MODE two registers are used to control interrupts—the INT STATUS REGISTER (ISR) and the INT MASK REGISTER (IMR). Each interrupt can be masked from interrupting via the IMR. When an interrupt is recognized by the μP, reading the ISR will display all active interrupt sources. The ISR contents remain unchanged until an interrupt reset is programmed. A shadow register behind the ISR guarantees that interrupts occurring while others are serviced will not be lost.

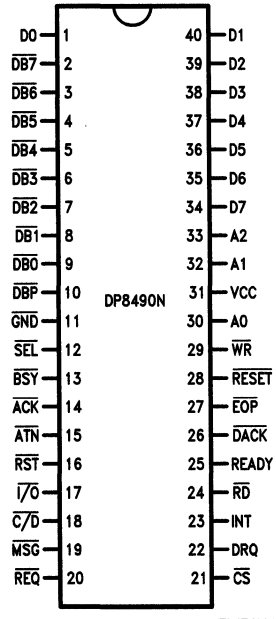
2.0 Pin Descriptions

Symbol	DIP	PCC	Type	Function
$\overline{CS}$	21	24	I	**Chip Select:** an active low enable for read or write operations, accessing the register selected by A0–A2.
A0–A2	30, 32 33	33, 36 37	I	**Address 0–2:** these three signals are used with $\overline{CS}$, $\overline{RD}$, and $\overline{WR}$ to address a register for read or write.
$\overline{RD}$	24	27	I	**Read:** an active low enable for reading an internal register selected by A0–A2 and enabled by $\overline{CS}$. It also selects the Input Data Register when used with $\overline{DACK}$.
$\overline{WR}$	29	32	I	**Write:** an active low enable for writing an internal register selected by A0–A2 and enabled by $\overline{CS}$. It also selects the Output Data Register when used with $\overline{DACK}$.
$\overline{RESET}$	28	31	I	**Reset:** an active low input with a Schmitt trigger. Clears all internal registers. (SCSI $\overline{RST}$ unaffected.)
D0–D7, P	1, 40–34	2, 44–38 1	I/O	**Data 0–7, P:** bidirectional TRI-STATE® signals connecting the active high μP data bus to the internal registers. The PCC part offers an optional parity on the μP data bus. If the parity option is not enabled the pin is TRI-STATE.
INT	23	26	O	**Interrupt:** an active high output to the μP when an error has occurred, an event requires service or has completed.
DRQ	22	25	O	**DMA Request:** an active high output asserted when the data register is ready to be read or written. DRQ occurs only if DMA mode is enabled. The signal is cleared by $\overline{DACK}$.
$\overline{DACK}$	26	29	I	**DMA Acknowledge:** an active low input that resets DRQ and addresses the data registers for input or output transfers. $\overline{DACK}$ is used instead of $\overline{CS}$ by the DMA controller.
READY	25	28	O	**Ready:** an active high output used to control the speed of block mode DMA transfers. Ready goes active when the chip is ready to send/receive data and remains inactive after the transfer until the byte is sent or until the DMA mode bit is reset.
$\overline{EOP}$	27	30	I	**End of Process:** an active low signal that terminates a block of DMA transfers. It should be asserted during the transfer of the last byte.
$\overline{DB0}$–$\overline{DB7}$ $\overline{DBP}$	9–2, 10	10–3, 11	I/O	**DB0–DB7, DBP:** SCSI data bus with parity. $\overline{DB7}$ is the MSB and is the highest priority during arbitration. Parity is default ODD but can be programmed EVEN. Parity is always generated and can be optionally checked. Parity is not valid during arbitration.
$\overline{RST}$	16	18	I/O	**Reset:** SCSI reset, monitored and can be set by EASI.
$\overline{BSY}$	13	15	I/O	**Busy:** indicates the SCSI bus is being used. Can be driven by TARGET or INITIATOR.

2.0 Pin Descriptions (Continued)

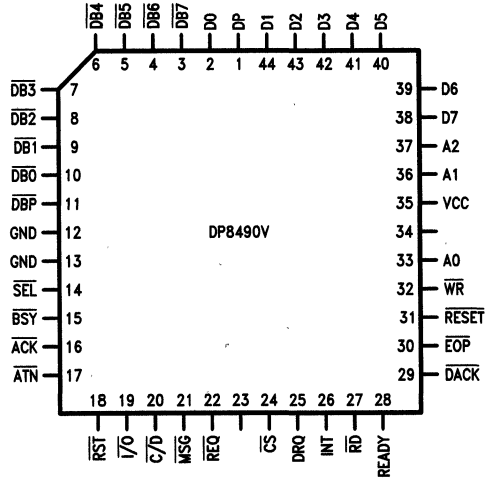
Symbol	DIP	PCC	Type	Function
$\overline{SEL}$	12	14	I/O	**Select:** used by the INITIATOR to select a TARGET or by the TARGET to reselect an INITIATOR.
$\overline{ACK}$	14	16	I/O	**Acknowledge:** driven by the INITIATOR and received by the TARGET as part of the $\overline{REQ}/\overline{ACK}$ handshake.
$\overline{ATN}$	15	17	I/O	**Attention:** driven by the INITIATOR to indicate an attention condition to the TARGET.
$\overline{REQ}$	20	22	I/O	**Request:** driven by the TARGET and received by the INITIATOR as part of the $\overline{REQ}/\overline{ACK}$ handshake.
$\overline{I/O}$	17	19	I/O	**Input/Output:** driven by the TARGET to control the direction of transfers on the SCSI bus. This signal also distinguishes between selection and reselection.
$\overline{C/D}$	18	20	I/O	**Command/Data:** driven by the TARGET to indicate whether command or data bytes are being transferred.
$\overline{MSG}$	19	21	I/O	**Message:** driven by the TARGET during message phase to identify message bytes on the bus.
V_{CC} GND	31 11	35 12, 13	—	**V_{CC}, GND:** $+5$ V_{DC} is required. Because of very large switching currents, good decoupling and power distribution is mandatory.

2.1 Connection Diagrams



TL/F/9387–4

Order Number DP8490N
See NS Package Number N40A



TL/F/9387–5

Order Number DP8490V
See NS Package Number V44A

3.0 Register Description

3.1 GENERAL

The DP8490 EASI is a register-based device with eight addressable locations used to access twelve registers. Some addresses have dual functions depending upon whether they are being read from or written to. Two basic operating "modes" result in differences in the registers accessed through address 7. Device operation is described in Section 4 but mode differences are highlighted in this section and Section 8.

The EASI operates in one of two modes—NORMAL (MODE N) and ENHANCED (MODE E). Switching between the modes is performed by setting or resetting bit 6 in the Initiator Command Register.

In MODE N, EASI registers appear the same as the DP5380. In MODE E, address 7 accesses enhanced logic features. To help identify these differences the register description is split into two subsections. The first describes the registers in MODE N. The second describes the register differences when in MODE E.

Figure 3.1 summarizes the register map in MODE N. Note that for registers reading or writing SCSI signals the SCSI name is used for each bit. Although the SCSI bus is active low, the registers invert the SCSI bus. This means an active SCSI signal is represented by a ONE in a register and an inactive signal by a ZERO.

Hex Adr	Register	Mnemonic	Bits	R/W
0	Output Data Register	ODR	8	WO
0	Current SCSI Data	CSD	8	RO
1	Initiator Command Register	ICR	8	RW
2	Mode Register 2	MR2	8	RW
3	Target Command Register	TCR	4	RW
4	Select Enable Register	SER	8	WO
4	Current SCSI Bus Status	CSB	8	RO
5	Bus and Status	BSR	8	RO
5	Start DMA Send	SDS	0	WO
6	Start DMA Target Receive	SDT	0	WO
6	Input Data Register	IDR	8	RO
7	Start DMA Initiator Receive	SDI	0	WO
7	Reset Parity/Interrupts	RPI	0	RO

FIGURE 3.1. Normal Mode Registers

3.2 NORMAL MODE REGISTERS

OUTPUT DATA REGISTER (ODR)

8 Bits HA 0 Write Only

This is a transparent latch used to send data to the SCSI bus. The register can be written by µP cycles or via DMA. DMA writes automatically select the ODR at Hex Address 0 (HA 0). This register is also written with the ID bits required during arbitration and selection/reselection phases. Data is latched at the end of the write cycle.

Bit 7 **Bit 0**

$\overline{DB7}$	$\overline{DB6}$	$\overline{DB5}$	$\overline{DB4}$	$\overline{DB3}$	$\overline{DB2}$	$\overline{DB1}$	$\overline{DB0}$

Output Data Register

CURRENT SCSI DATA (CSD)

8 Bits HA 0 Read Only

This register enables reading of the current SCSI data bus. If SCSI parity checking is enabled it will be checked at the beginning of the read cycle. The register is also used for µP accesses of SCSI data during programmed-I/O or ID checking during arbitration. Parity is not valid during arbitration. DMA transfers select the IDR (HA 6) instead of the CSD register.

Bit 7 **Bit 0**

$\overline{DB7}$	$\overline{DB6}$	$\overline{DB5}$	$\overline{DB4}$	$\overline{DB3}$	$\overline{DB2}$	$\overline{DB1}$	$\overline{DB0}$

Current SCSI Data

INITIATOR COMMAND REGISTER (ICR)

8 Bits HA 1 Read/Write

This register is used to control the INITIATOR and some other SCSI signals, and to monitor the progress of bus arbitration. Most of the SCSI signals may also be asserted in TARGET mode. Bits 5 to 0 are reset when $\overline{BSY}$ is lost (see MR2 description).

Bit 7 **Bit 0**

$\overline{RST}$	AIP/ MODE	LA/ DIFF	$\overline{ACK}$	$\overline{BSY}$	$\overline{SEL}$	$\overline{ATN}$	DBUS

Initiator Command Register

DBUS: Assert Data Bus Bit 0

0 Disable SCSI data bus driving.

1 Enable contents of Output Data Register onto the SCSI data bus. SCSI parity is also generated and driven on DBP.

This bit should be set when transferring data out of the EASI in either TARGET or INITIATOR mode, for both DMA or programmed-I/O. In INITIATOR mode the drivers are only enabled if: Mode Register 2 TARGET MODE bit is 0, and $\overline{I/O}$ is false, and $\overline{C/D}$, $\overline{I/O}$, $\overline{MSG}$ match the contents of the Target Command Register (phasematch is true). In TARGET mode only the MR2 bit needs to be set with this bit.

Reading the ICR reflects the state of this bit.

$\overline{ATN}$: Assert Attention Bit 1

0 Deassert $\overline{ATN}$

1 Assert SCSI $\overline{ATN}$ signal. The MR2 TARGET MODE bit must also be false to assert the signal.

Reading the ICR reflects the state of this bit.

$\overline{SEL}$: Assert Select Bit 2

0 Deassert $\overline{SEL}$

1 Assert SCSI $\overline{SEL}$ signal. Can be used in INITIATOR or TARGET mode.

Reading the ICR reflects the state of this bit.

$\overline{BSY}$: Assert Busy Bit 3

0 Deassert $\overline{BSY}$.

1 Assert SCSI $\overline{BSY}$ signal. Can be used in INITIATOR or TARGET mode.

Reading the ICR reflects the state of this bit.

3.0 Register Description (Continued)

$\overline{ACK}$: Assert Acknowledge Bit 4

0 Deassert $\overline{ACK}$.

1 Assert SCSI $\overline{ACK}$ signal. The MR2 TARGET MODE bit must also be false to assert the signal.

Reading the ICR reflects the state of this bit.

DIFF: Differential Enable Bit 5 Write

0 This bit must be reset to 0.

LA: Lost Arbitration Bit 5 Read

0 Normally reset to 0 to show arbitration not lost or not enabled.

1 Will be set when the EASI loses arbitration, i.e. when SEL is true during arbitration AND the Assert $\overline{SEL}$ bit of this register is false.

A 1 in this bit means the EASI has arbitrated for the bus, asserted $\overline{BSY}$ and its ID on the data bus and another device has asserted $\overline{SEL}$. The ARBITRATE bit in MR2 or the EMR must be set to enable arbitration.

MODE: Operating Mode Bit 6 Write

0 Normal Mode (MODE N) is selected.

1 Enhanced Mode (MODE E) is selected.

AIP: Arbitration In Progress Bit 6 Read

0 Normally 0 to show no arbitration in progress.

1 Set when the EASI has detected BUS FREE phase and asserted $\overline{BSY}$ and the Output Data Register contents onto the SCSI data bus. This bit remains set until arbitration is disabled.

$\overline{RST}$: Assert $\overline{RST}$ Bit 7

0 Deassert $\overline{RST}$.

1 Assert SCSI $\overline{RST}$ signal. $\overline{RST}$ is asserted as long as this bit is 1, or until a μP Reset ($\overline{RESET}$).

After this bit is set the INT pin goes active and internal registers are reset (except for the interrupt latch, MR2 TARGET MODE bit, and this bit). Reading the ICR reflects the state of this bit.

MODE REGISTER 2 (MR2)

8 Bits HA 2 Read/Write

This register is used to program basic operating conditions in the EASI. Operation as TARGET or INITIATOR, DMA mode and type as well as some interrupt controls are set via this register. This is a read/write register and when read the value reflects the state of each bit.

Bit 7							Bit 0
BLK	TARG	PCHK	PINT	EOP	BSY	DMA	ARB

Mode Register 2

ARB: Arbitrate Bit 0

0 Disable arbitration

1 Enable arbitration. The EASI will wait for a BUS FREE phase then arbitrate for the bus. Before setting this bit the Output Data Register should contain the SCSI device ID—a single bit set only. The status of the arbitration process is given in the AIP and LA bits (6,5) in the Initiator Command Register.

DMA: DMA Mode Bit 1

0 Disable DMA mode

1 Enable DMA operation. This bit should be set then one of address 5 to 7 written to start DMA. The TARGET MODE bit in the ICR and the phase lines in the TCR should have been set appropriately. The DBUS bit in the ICR must be set for DMA send operations. $\overline{BSY}$ must be active in order to set this bit. The phase lines must match the contents of the TCR during the actual transfers. In DMA mode EASI logic automatically controls the $\overline{REQ}$/$\overline{ACK}$ handshakes.

This bit should be reset by a μP write to stop any DMA transfer. An $\overline{EOP}$ signal will not reset this bit. During DMA, $\overline{CS}$ and $\overline{DACK}$ should not be active simultaneously.

This bit will be reset if $\overline{BSY}$ is lost during DMA mode.

BSY: Monitor Busy Bit 2

0 Disable $\overline{BSY}$ monitor.

1 Monitor SCSI $\overline{BSY}$ signal and interrupt when $\overline{BSY}$ goes inactive. When this bit goes active the lower 6 bits of the ICR are reset and all signals removed from the SCSI bus. This is used to check for valid TARGET connection.

EOP: Enable $\overline{EOP}$ Interrupt Bit 3

0 No interrupt for $\overline{EOP}$.

1 Interrupt after valid $\overline{EOP}$ condition.

PINT: Enable SCSI Parity Interrupt Bit 4

0 No Interrupt on SCSI parity error.

1 When SCSI parity is enabled via the PCHK bit, setting this bit enables an interrupt upon a SCSI parity error.

PCHK: Enable SCSI Parity Checking Bit 5

0 No SCSI parity checking.

1 Enable checking of SCSI parity during read operations. This applies to either programmed I/O or DMA mode.

TARG: Target Mode Bit 6

0 Initiator Mode

1 Target Mode

BLK: Block Mode DMA Bit 7

0 Non-block DMA

1 When set along with DMA bit (1), enables block mode DMA transfers. In block mode the READY line is used to handshake each byte with the DMA controller instead of the DRQ/$\overline{DACK}$ handshake used in non-block mode.

TARGET COMMAND REGISTER (TCR)

4 Bits HA 3 Read/Write

This register is used to control TARGET SCSI signals and to program the desired phase during INITIATOR mode. During DMA transfers the SCSI phase lines ($\overline{C/D}$, $\overline{MSG}$, $\overline{I/O}$) must match the contents of the TCR for transfers to occur. A phase mismatch halts DMA transfers and generates an interrupt.

Bit 7							Bit 0
X	X	X	X	$\overline{REQ}$	$\overline{MSG}$	$\overline{C/D}$	$\overline{I/O}$

Target Command Register

3.0 Register Description (Continued)

This is a read/write register and the value read reflects the state of each bit, except bits 4–7 which always read 0.

I/O: Assert I/O Bit 0

0 Deassert I/O.

1 Assert SCSI I/O signal. The MR2 TARGET MODE bit must also be active.

C/D: Assert C/D Bit 1

0 Deassert C/D

1 Assert SCSI C/D signal. The MR2 TARGET MODE bit must also be active.

MSG: Assert MSG Bit 2

0 Deassert MSG.

1 Assert SCSI MSG signal. The MR2 TARGET MODE bit must also be active.

REQ: Assert REQ Bit 3

0 Deassert REQ.

1 Assert SCSI REQ signal. The MR2 TARGET MODE bit must also be active. This bit is used to handshake SCSI data via programmed-I/O.

SELECT ENABLE REGISTER (SER)

8 Bits HA 4 Write Only

This write-only register is used to program the SCSI device ID for the EASI to respond to during Selection or Reselection phases. Only one bit in the register should be set. When SEL is true, BSY false and the SER ID bit active an interrupt will occur.

This interrupt is reset or can be disabled by writing zero to this register. Parity will also be checked during Selection or Reselection if the PCHK bit in MR2 is set.

Bit 7							Bit 0
DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0

Select Enable Register

CURRENT SCSI BUS STATUS (CSB)

8 Bits HA 4 Read Only

This read-only register is used to monitor SCSI control signals and the SCSI parity bit. The SCSI lines are monitored during programmed-I/O transfers and after an interrupt in order to determine the cause. A bit is 1 if the corresponding SCSI signal is active.

Bit 7							Bit 0
RST	BSY	REQ	MSG	C/D	I/O	SEL	DBP

Current SCSI Bus Status

BUS AND STATUS REGISTER (BSR)

8 Bits HA 5 Read Only

This read-only register is used to monitor SCSI signals not included in the CSB, and internal status bits. This register is read after an interrupt (in MODE N) to determine the cause of an interrupt. Bit 0 or 1 are set to 1 if the SCSI signal is active.

Bit 7							Bit 0
EDMA	DRQ	SPER	INT	PHSM	BSY	ATN	ACK

Bus and Status Register

ACK: Acknowledge Bit 0

This bit reflects the state of the SCSI ACK Signal.

ATN: Attention Bit 1

This bit reflects the state of the SCSI ATN Signal.

BSY: Busy Error Bit 2

0 No error

1 The SCSI BSY signal has become inactive while the MR2 BSY (Monitor BSY) bit is set. This will cause an interrupt, remove all EASI signals from the SCSI bus and reset the DMA MODE bit in MR2.

PHSM: Phase Match Bit 3

0 Phase Match. The SCSI C/D, I/O and MSG phase lines are continuously compared with the corresponding bits in the TCR. The result of this comparison is reflected in this bit. This bit must be 1 (phase matches) for DMA transfers. A phase mismatch will stop DMA transfers and cause an interrupt.

INT: Interrupt Request Bit 4

0 No interrupt

1 Interrupt request active. Set when an enabled interrupt condition occurs. This bit reflects the state of the INT pin. INT may be reset by performing a Reset Parity/Interrupt (RPI) function.

SPER: SCSI Parity Error Bit 5

0 No SCSI parity error

1 SCSI parity error occurred. This bit remains set once an error occurs until the RPI function clears it. The PCHK bit in MR2 must be set for a parity error to be checked and registered.

DRQ: DMA Request Bit 6

0 No DMA request

1 DMA request active. This bit reflects the state of the DRQ pin. DRQ is reset by asserting DACK during a DMA cycle or by resetting the DMA bit in MR2. A Busy error will reset the MR2 DMA bit and thus will also clear DRQ. A phase mismatch will not reset DRQ.

EDMA: End of DMA Bit 7

0 Not end of DMA

1 Set when DACK, EOP and either RD or WR are active simultaneously. Normally occurs when the last byte is transferred by the DMA. During DMA send operations the last byte transferred by the DMA may not have been transferred on SCSI so REQ and ACK should be monitored to verify when the last SCSI transfer is complete. This bit is reset when the MR2 DMA bit is reset.

Note: In MODE E the EASI presents a true EDMA bit in bit 7 of the TCR. This feature removes the need to poll the REQ and ACK signals.

START DMA SEND (SDS)

0 Bits HA 5 Write Only

This write-only register is used to start a DMA send operation. A write of don't care data should be the last thing done by the μP. The MR2 DMA, BLK and TARG bits must have been programmed previously.

Bit 7							Bit 0
X	X	X	X	X	X	X	X

Start DMA Send

3.0 Register Description (Continued)

START DMA TARGET RECEIVE

0 Bits HA 6 Write Only

This write-only register is used to start a DMA Target Receive operation. Same comments as SDS apply.

INPUT DATA REGISTER (IDR)

8 Bits HA 6 Read Only

This read-only register contains the SCSI data last latched during a DMA receive. Each byte from SCSI is latched into this register automatically by the EASI DMA logic. A DMA read ($\overline{DACK}$ and $\overline{RD}$) automatically selects this register. Programmed I/O SCSI data reads should use the CSD (HA 8).

START DMA INITIATOR RECEIVE (SDI)

0 Bits HA 7 Write Only

This write-only register is used to start DMA INITIATOR Receive Operation. Same comments as SDS apply. An alternative method of performing the SDI function is available in the Enhanced Mode Register.

RESET PARITY/INTERRUPT (RPI)

0 Bits HA 7 Read Only

This read-only register is used to reset the parity and interrupt latches. Reading this register resets the SCSI parity, μP parity, Busy Loss and Interrupt Request latches. It also resets the interrupt latches presented in the Interrupt Status Register (available in MODE E).

An alternative method of performing the Reset Parity Interrupt function is available in the Enhanced Mode Register. In MODE E writing a value of 01 to bits 2 and 1 of the EMR will reset the same bits as a read from HA 7 in MODE N. The EMR RPI will also reset enhanced logic that has bits set in the Interrupt Status Register.

3.3 ENHANCED MODE REGISTERS

Addresses 0 to 6 remain the same as MODE N except for bit 7 of TCR, as described below. Address 7 is the SDI and RPI functions in MODE N, but in MODE E it directly accesses the Enhanced Mode Register (EMR) and indirectly accesses the Interrupt Mask Register (IMR) and the Interrupt Status Register (ISR).

When bit 6 in the ICR (HA 1) is set, HA 7 accesses the read/write EMR. The SDI and RPI functions performed when writing/reading HA 7 in MODE N are disabled. To perform these functions the EMR is used instead.

Note that EMR functions are intended to be used in an interrupt-driven environment. Reading this register reflects the state of each bit.

TARGET COMMAND REGISTER (TCR)

5 Bits HA 3 Read/Write

This is the same as MODE N except for bit 7 which is described below. Note bits 4–6 always read 0.

Bit 7 **Bit 0**

(true) EDMA	X	X	X	$\overline{REQ}$	$\overline{MSG}$	$\overline{C/D}$	$\overline{I/O}$

Target Command Register

EDMA: True End of DMA Bit 7

0 Not End of DMA

1 Set when the last byte of data has been transferred. This bit is not set until $\overline{REQ}$ and $\overline{ACK}$ both go inactive following the DMA cycle during which $\overline{EOP}$ was asserted. Note that unlike the BSR EDMA bit, this bit reflects the true completion of DMA transfers.

ENHANCED MODE REGISTER (EMR)

8 Bits HA 7 Read/Write

This register is accessed at HA 7 when the ICR MODE bit (6) is set. The register controls operation of enhanced logic and timing. Normally the application will leave the EASI permanently in MODE N or MODE E.

Bit 7 **Bit 0**

APHS	MPEN	MPOL	SPOL	LOOP	EFN1	EFN0	ARB

Enhanced Mode Register

ARB: Extended Arbitration Bit 0

0 Disable extended arbitration

1 Enable extended arbitration. This is an alternative bit to the MR2 ARB function. The EASI waits for a BUS FREE phase then arbitrates for the bus, asserting the contents of the ODR onto the SCSI data bus and asserting $\overline{BSY}$. The EASI will then wait the 2.2 μs SCSI Arbitration Delay, and set the Arbitration Complete bit in the ISR and cause an interrupt. As for the MR2 ARB function, the ICR LA and AIP bit can be examined to determine arbitration status.

EFN1,0: Enhanced Function Bits 2,1

00 No operation—these bits ALWAYS read 00.

01 When this pattern is written the Parity and Interrupt latches are reset. This pattern should be followed by any other pattern to remove the reset (usually 00). This function replaces the RPI function performed when reading HA 7 in MODE N. Only the latches with ISR bits set to 1 will be reset.

10 Start DMA Initiator Receive. At the end of the write cycle the SDI function is performed. There is no need to follow with another pattern as required with the RPI value (01). This function replaces the SDI function performed when writing HA 7 in MODE N.

11 Read/Write ISR/IMR. When this pattern is written the NEXT read of HA 7 will access the ISR, or the NEXT write to HA 7 will access the IMR. This state ONLY lasts for the ONE following read OR write cycle. Further cycles will then access the EMR.

LOOP: Loopback Mode Bit 3

0 Normal operation

1 When set: SCSI drivers are disabled and the SCSI I/O's looped back inside the EASI, and both the TARGET and INITIATOR signals may be driven simultaneously. This enables the μP to check EASI operation without affecting the SCSI bus.

SPOL: SCSI Parity Polarity Bit 4

0 SCSI parity is ODD (as per SCSI specification).

1 SCSI parity is EVEN. This allows diagnostics to be performed.

3.0 Register Description (Continued)

MPOL: μP Parity Polarity Bit 5

0 μP parity is ODD.

1 μP parity is EVEN.

MPEN: μP Parity Enable Bit 6

0 μP parity checking and generation disabled.

1 Enable checking of μP data bus parity during μP and DMA writes. Generate parity during μP and DMA reads. Parity errors will cause an interrupt and set the ISR MPE bit if not masked.

APHS: Any Phase Mismatch Bit 7

0 Disable phase mismatch detection

1 Detect SCSI requests with a phase mismatch present. Is set when $\overline{REQ}$ goes active AND the SCSI phase lines do not match the contents of the TCR. Can be used in INITIATOR mode to interrupt on TARGET phase changes.

INTERRUPT STATUS REGISTER (ISR)

8 Bits HA 7 (EFN = 11) Read Only

This register is accessed during the first read cycle after the EFN bits in the EMR have both been set to 1. Once read, successive accesses of HA 7 go to the EMR. This register provides all interrupt status within one register. This is intended to make determination of interrupt sources easier in MODE E. In MODE N two registers must be read—the BSR and CSB. In MODE E only the ISR needs to be read. Additionally each interrupt (except SCSI $\overline{RST}$) has a corresponding status bit in the ISR.

When the ISR is read all unmasked, enabled, active interrupt sources set their corresponding ISR bits to a ONE. The interrupt status is sampled at the beginning of the ISR read cycle. When an RPI function is performed via the EMR, only each interrupt latch with an ISR bit set will be reset. This means interrupts occurring since the last ISR read will not be lost.

ISR bits may be individually masked via their corresponding bits in the IMR.

Bit 7							Bit 0
SPE	MPE	EDMA	DPHS	APHS	BSY	SEL	ARB

Interrupt Status Register

ARB: Arbitration Complete Bit 0

This bit is set when arbitration enabled by EMR ARB bit (0) has completed. Completion occurs in two ways: when the EASI loses arbitration and has asserted the LA bit in the ICR; or when the EASI has asserted the ID contained in the ODR onto the data bus, asserted $\overline{BSY}$, then waited for the 2.2 μs SCSI Arbitration Delay.

SEL: Selection/Reselection Bit 0

This bit is set when $\overline{BSY}$ is false, $\overline{SEL}$ is active, and any SER bit set to 1 has an active corresponding SCSI data bus bits. This situation occurs during Selection or Reselection phases.

BSY: Busy Loss Bit 2

This bit is the same as BSR bit 2. Set when the SCSI $\overline{BSY}$ signal becomes inactive while the MR2 BSY (monitor $\overline{BSY}$) bit is set.

APHS: Any Phase Mismatch Bit 3

Set when a $\overline{REQ}$ occurs while the SCSI phase lines do not match. Bit is set to enable detection of the contents of the TCR. EMR APHS (bit 7) must be 1 to allow this bit to be set.

DPHS: DMA Phase Mismatch Bit 4

Set when a SCSI DMA mode operation occurs with a phase mismatch. Similar to the APHS condition but restricted to DMA mode only. In MODE N this condition is not as easily determined.

EDMA: End of DMA Bit 5

Set when $\overline{REQ}$ and $\overline{ACK}$ are both false following the DMA cycle during which $\overline{EOP}$ was asserted. This represents the true end of DMA operation.

MPE: μP Parity Error Bit 6

Set when μP parity error is detected at the end of a μP or DMA write cycle. The MPEN bit in the EMR must be set for μP parity to be checked.

SPE: SCSI Parity Error Bit 7

Set when a SCSI parity error is detected when reading the CSD, during selection/reselection, or when the IDR is loaded during DMA operation. The MR2 PCHK bit must be set for SCSI parity to be checked, and the MR2 PINT bit must be set to enable the interrupt.

INTERRUPT MASK REGISTER (IMR)

8 Bits HA 7 (EFN = 11) Write Only

This register is accessed during the first write cycle after the EFN bits in the EMR have been set to 11. Once written, successive reads or writes at HA 7 access the EMR.

This register has the same bit definition as the ISR. If a bit in the IMR is set to 1 that interrupt will be masked. The interrupt will be captured internally (if enabled) but will not cause an active INT signal. A bit reset to 0 will enable that interrupt to occur (if enabled) and will enable the ISR bit to be set to 1.

Bit 7							Bit 0
SPE	MPE	EDMA	DPHS	APHS	BSY	SEL	ARB

Interrupt Mask Register

4.0 Device Operation

4.1 GENERAL

This section describes overall operation of the EASI. More detailed information of data transfers, interrupts and reset conditions are covered in later sections. The operation description covers μP accesses, SCSI bus monitoring, arbitration, selection, reselection, programmed-I/O, DMA interrupts. Programming and timing details are covered.

For information regarding interfacing to μPs and DMA controllers refer to Section 9.

In the descriptions following, program examples are given in pseudo-C. This processor-independent approach should be clearest. These are backed up by flow charts in Appendix A1.

For each section where appropriate the description is split into two, MODE N and MODE E.

4.0 Device Operation (Continued)

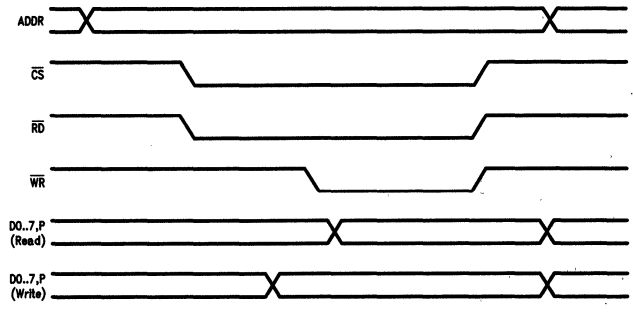


FIGURE 4.2. μP Cycles

4.2 μP ACCESSES

The μP accesses the EASI via the $\overline{CS}$, $\overline{RD}$, $\overline{WR}$ and address and data lines in order to read/write the registers. *Figure 4.1* shows typical timing. Note the use of non-multiplexed address and data lines.

4.3 SCSI BUS MONITORING/DRIVING

The SCSI bus may be monitored or driven at any time. Each bus signal is buffered and inverted by the EASI and can be read via the CSB, BSR and CSD registers. An active SCSI signal reads a 1 in the status registers.

Each SCSI signal may be asserted by setting a bit in the TCR or ICR. Setting the bit to 1 asserts the SCSI signal.

The following code demonstrates a byte transferred via programmed-I/O in INITIATOR mode.

```
{    /* Transfer one byte as Initiator */
     while (NOT (TCR: REQ)) ;
      /* wait till TARGET asserts REQ */
     data = input (CSD) ;
      /* parity is checked if enabled */
     output (ICR, Assert_ACK) ;
     while (TCR:REQ) ;
      /* wait till TARGET deasserts REQ */
     output (ICR, 0) ;
      /* deassert ACK, ready for next byte */
}
```

4.4 ARBITRATION

This sub-section describes the arbitration support provided by the EASI and how to program it.

4.4.1 MODE N Arbitration

Since the SCSI arbitration process requires signal sequencing too fast for μPs, hardware support is provided by the EASI. The arbitration process is enabled by bit 0 MR2 (ARB). Prior to setting this bit the ODR should be programmed with the device's SCSI ID—a single bit.

The EASI will monitor the bus for a BUS FREE phase. The $\overline{BSY}$ signal is continuously monitored. If continuously inactive for at least a SCSI Bus Settle Delay (400 ns) and $\overline{SEL}$ is inactive, a valid Bus Free Phase exists. After a period of SCSI Bus Free Delay (800 ns) the EASI asserts $\overline{BSY}$ and the ODR onto the SCSI data bus. The μP should poll the ICR to determine when arbitration has started. The AIP bit in the ICR is set when the Bus Free Phase is detected and the EASI is beginning the Bus Free Delay. Following the Bus Free Delay a 2.2 μs SCSI Arbitration Delay is required before examining the data bus to resolve the priorities of the ID bits. This delay must be implemented in firmware. The ICR Lost Arbitration (LA) bit must be examined to determine whether arbitration is lost. The LA bit is set if another device asserts $\overline{SEL}$ during arbitration. If the LA bit is 0 the data bus is read via the CSD register. The data is examined to resolve ID priorities. If this device is the highest ID, assert $\overline{SEL}$ by setting ICR bit 2 to a 1. After waiting Bus Clear + Bus Settle Delays (1200 ns), the Selection Phase begins. These 2 delays must be implemented in firmware.

4.4.2 MODE E Arbitration

The extended arbitration in MODE E is enabled by bit 0 of the EMR (ARB). This alternative offers two significant advantages over MODE N. First the 2.2 μs SCSI Arbitration Delay is implemented by the EASI. Second the arbitration process may be interrupt driven.

In MODE N the EASI must be polled to see when the ICR AIP bit is set. After the appropriate delays, the LA bit and data bus are examined. If arbitration is lost or this device is not the highest priority the MR2 ARB bit must be reset to 0, then set to 1 again and the whole process restarted. This means the EASI MUST be polled until arbitration is won—potentially many SECONDS, typically ms. This ties up the host μP.

In MODE E when the EMR ARB bit is set the EASI will: wait for BUS FREE phase; delay SCSI Bus Free Delay; assert $\overline{BSY}$ and the ODR onto $\overline{DB0}$–$\overline{DB7}$; delay SCSI Arbitration Delay; interrupt the μP.

The μP should read the ISR and if the ISR bit 0 (ARB complete) is set examine ICR bits 5 and 6 (LA and AIP) to determine whether arbitration is lost. If not lost the data bus is examined to resolve ID priorities. As for MODE N if arbitration has failed the EMR ARB bit should be reset and then set again after first resetting the ARB interrupt. Note that the EMR ARB bit allows the μP to carry on with other tasks while the EASI arbitrates. This means there is NO NEED to poll the EASI.

4.5 SELECTION/RESELECTION

The EASI can be used to select or reselect a device. The EASI will also respond to selection or reselection. Selecting or reselecting a device is the same in MODE N or MODE E. Response to selection or reselection can differ between the bus modes.

4.0 Device Operation (Continued)

4.5.1 Selecting/Reselecting

Selection requires programming the ODR with the desired and own device IDs; the data bus via ICR DBUS (bit 0); asserting $\overline{ATN}$ if required via ICR bit 1; asserting $\overline{SEL}$ via ICR bit 2; then resetting the MR2/EMR ARB bit.

The SER should have been cleared to zero before Selection/Reselection to ensure the EASI does not respond. If Reselection is desired the $\overline{I/O}$ line should also be asserted before $\overline{SEL}$ via TCR bit 0.

Resetting the ARB bit causes the EASI to remove $\overline{BSY}$ and the ODR from the data bus. Thus the ICR Assert data bus bit is required to assert the bits for desired and own device IDs.

$\overline{BSY}$ is then monitored to determine when the device has responded to (re)selection. If the device fails to respond an error handler should sequence the EASI off the bus. If the device responds the ICR DBUS and $\overline{SEL}$ bits should be reset to remove these signals. If this is a Reselection the ICR $\overline{BSY}$ bit (3) should be set before removing the other signals.

The bus is now ready to handle Information Transfer Phases.

4.5.2 MODE N (Re)Selection Response

The EASI responds to Selection or Reselection when the SER is non-zero. A (re)selected interrupt is generated when $\overline{BSY}$ is false for at least a Bus Settle Delay (400 ns); and $\overline{SEL}$ is true AND any non-zero bit in the SER has its corresponding SCSI data bus bit active. A Selection is disabled by zeroing the SER. If parity is supported it should be valid during (re)selection so it must be checked via the SPE bit (5) in the BSR. SCSI specification states that (re)selection is not valid if more than 2 data bits are active. This condition is checked by reading the CSD.

When the selection interrupt occurs it is determined by reading the BSR and CSB registers. There is no dedicated status bit for (re)selection in MODE N, it must be determined by the absence of other interrupts, and the active state of the $\overline{SEL}$ signal. Reselection occurs when $\overline{I/O}$ is also active. See Section 6.

4.5.3 MODE E (Re)Selection Response

The same conditions for valid (re)selection apply as for MODE N. A (re)selection interrupt is generated as per MODE N. The difference is that the interrupt sets the SEL bit (1) in the ISR. In MODE E, reading the ISR enables exact determination of interrupt sources. This interrupt can be masked via bit 1 of the IMR. The interrupt is reset by programming the RPI function to the EMR. See Section 6 for further details.

4.6 MONITORING BSY

While an INITIATOR is connected to a TARGET the TARGET must maintain an active $\overline{BSY}$ signal. During DMA operations the $\overline{BSY}$ signal is monitored by the EASI and will halt operations if it goes inactive. To enable $\overline{BSY}$ to be monitored at other times the MR2 BSY bit (2) should be set. An interrupt will be generated if $\overline{BSY}$ goes inactive while MR2 BSY is set.

In MODE N this interrupt sets bit 2 in the BSR. In MODE E this interrupt sets bit 2 in the BSR and bit 2 in the ISR. The interrupt may be masked via bit 2 of the IMR.

4.7 COMMAND/MESSAGE/STATUS TRANSFERS

Command, message and status bytes are transferred using programmed-I/O. The SCSI $\overline{REQ}/\overline{ACK}$ handshake is accomplished by monitoring and setting lines individually. Data is output via the ODR and read in via the CSD register.

The following code shows INITIATOR and TARGET programming for two of these cases. See Appendix A1 for flowcharts.

Initiator Command Send

```
{
  MR2 = monitor BSY
  TCR = Command Phase  /* 02h */
  while (bytes) to do) {
        while (REQ) inactive)
              idle; /* CSB bit 5 = 0 */
        if (BSR: phase_match==0)
              phase error;
        else  {
              ODR = data byte;
              ICR = Assert_ACK;
              while (REQ active)
              idle; /* CSB bit 5==1 */
              ICR = deassert_ACK
              /* byte transfer complete */
              byte count - -;
        }
  }
  goto data phase;
}
```

Target Message Receive

```
{
  /* assumed Assert_BSY already set in ICR */
  MR2 = TARG MODE OR PARITY CHECK
        OR PARITY INTERRUPT;
  TCR = Message_Out phase; /* 06h */
  delay (Bus Settle);
  TCR = Assert_REQ;
  while (ACK inactive)
        idle; /* BSR bit 0 */
  data = CSD; /* parity is latched */
  if (BSR: parity_error)
        error routine;
  else  {
        TCR = deassert_REQ;
        while (ACK active)
              idle;
  }
    /* message done, can change to next
       phase*/
}
```

4.8 NON-BLOCK DMA TRANSFERS

Data transfers may be effected by DMA. This method should be used for optimum performance. Two methods of DMA are available—block and non-block mode. This section describes non-block mode transfers. MODE N operation is covered first followed by MODE E.

The interface to the DMA controller uses the DRQ, $\overline{DACK}$, $\overline{EOP}$ lines in non-block mode. Each byte is requested (DRQ) and ack'd ($\overline{DACK}$). Representative timing for a DMA read is shown in *Figure 4.8.1*.

4.0 Device Operation (Continued)

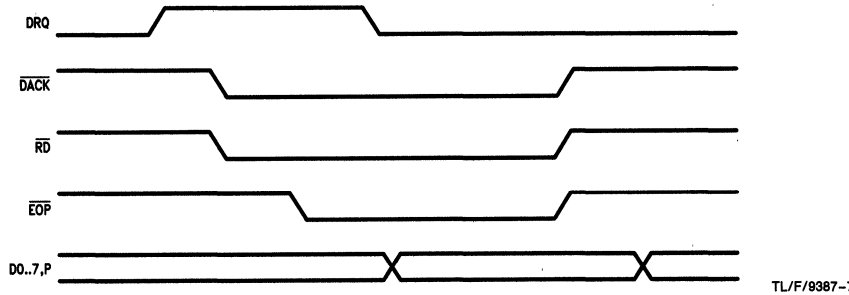


FIGURE 4.8.1. Non-Block DMA Timing

4.8.1 MODE N Non-Block DMA

DMA operation involves programming the EASI with the set-up parameters, initiating the DMA cycles and checking for correct operation when the completion interrupt is received. The DMA controller should be programmed with the data byte count and the memory start address. Methods of halting a DMA operation are covered in Section 4.11.

Setting up the EASI requires enabling or disabling the following: Data bus driving, DMA mode enable, BSY monitoring, EOP interrupt, parity checking, parity interrupt, TARGET Mode, bus phase.

Once set up, DMA should be initiated by writing to address 5, 6 or 7 as appropriate. The DMA controller should assert EOP during the transfer of the last byte, although this may be done by the μP if the DMA transfers (n−1) bytes and the μP transfers the last byte. See the application guide for more details (Section 9).

Upon completion the μP should check the following as required: End of DMA, Parity Error, Phase Match, Busy Error. In MODE N the end of DMA occurs as a response to EOP. SCSI transfers may still be underway so REQ and ACK must still be checked to establish when the final byte is finished.

The code below shows programming of the EASI in each of the four DMA cases. One of these cases is shown in a flow diagram in Appendix A.

```
Initiator Send   /* DATA OUT PHASE */
{
  Program DMA Controller;
  TCR = 00h;     /* phase */
  ICR = 01h;     /* Assert_DBUS */
  MR2 = 0Eh;
  SDS = 00;      /* Start DMA Send */
  while (NOT interrupt)
        idle;
  while (CSD:REQ)
        idle;    /* wait for last SCSI byte
                    transfer so phase
                    is checked */
  if(BSR:Busy_error OR NOT(BSR:End_of_DMA))
        error routine;
  else {         /* DMA END */
        MR2 = 04h;  /* reset DMA bit */
        ICR = 0;
  }
}
```

```
Initiator Receive  /* DATA IN PHASE */
{
  Program DMA Controller;
  TCR = 01h;     /* phase */
  MR2 = 3Eh;
  SDI = 0;       /* Start DMA Init Rx */
  while (NOT interrupt)
        idle;
  /* no need to wait for last SCSI handshake
    done since DMA done implies it is
    checked */
  if (BSR:parity_error OR BSR:busy_error
        or NOT (BSR End of DMA))
              do error routines;
  else  {        /* End of DMA */
        while (CSD:REQ)
              idle;  /* wait for REQ inactive
                to deassert ACK */
        MR2 = 04h;
  }
}

Target Receive  /* DATA OUT PHASE */
{
  Program DMA controller;
  TCR = 0;       /*phase*/
  ICR = 08h;
  MR2 = 7Ah;     /*check parity*/
  SDT = 0;       /*Start DMA Targ Rx*/
  while (not interrupt)
        idle;
  /*when End of DMA occurs the last byte
    has been read and checked*/
  if(BSR:parity_error OR NOT(BSR:End_of_DMA)
        error routine;
  else  {        /* End of DMA */
  while (BSR:ACK)
        idle;
  /*Not True End of DMA in MODE N, so wait
    until SCSI bus inactive before
    changing phase */
  MR2 = 40h;
  change phase as required;
  }
}
```

4.0 Device Operation (Continued)

```
Target Send  /* DATA IN PHASE */
{
  Program DMA Controller;
  TCR = 01h;   /* phase */
  ICR = 09h;
  MR2 = 4Ah;
  SDS = 0;     /* Start DMA Send */
  while (NOT interrupt)
        idle;
  if (NOT(BSR:END_of_DMA))
        error;
  else  {      /* DMA end */
        repeat {
              while (CSB:REQ OR BSR:ACK)
              loop count = 3;
              loop count - -;}
              /* decrement */
        until (loop count == 0);
        MR2 = 40h;
  Change phase as required;
  }
}
```

Some explanation of the final part of Target Send is required. In this type of DMA operation it is very difficult to exactly determine the True End of DMA in MODE N. Simply detecting $\overline{REQ}$ and $\overline{ACK}$ simultaneously inactive is not enough.

Reference to *Figure 4.8.2* will help to understand the following text.

As shown in *Figure 4.8.2,* $\overline{ACK}$ going active causes the DRQ for the next byte and also $\overline{REQ}$ to go inactive. $\overline{ACK}$ going inactive allows $\overline{REQ}$ to go active for the next byte. If the INITIATOR is slow removing $\overline{ACK}$ the μP may sample the SCSI bus after the $\overline{EOP}$ interrupt at point A. Here both $\overline{REQ}$ and $\overline{ACK}$ will be inactive, but there is one more byte to transfer on SCSI. Due to chip timing delays this condition will not last more than 200 ns. A safe way to determine the True End of DMA is to sample $\overline{REQ}$ and $\overline{ACK}$ and ONLY when both are inactive in three successive samples will the μP be at point B in the figure.

In MODE E, True End of DMA is correctly decoded and the End of DMA interrupt occurs as a result of this True condition, thus there is no need to sample $\overline{REQ}$ and $\overline{ACK}$. For this and other reasons operation in MODE E is strongly recommended.

4.8.2 MODE E Non-Block DMA

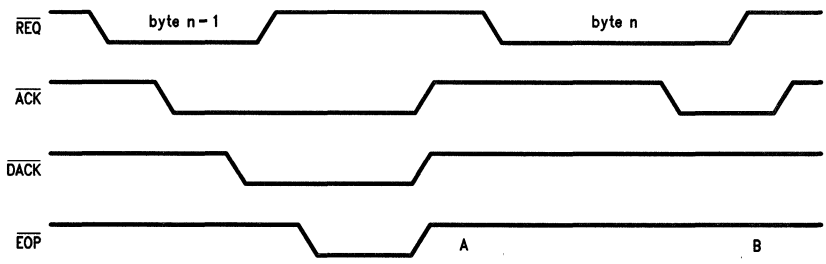
Operation for the Non-Block DMA in MODE E is essentially the same as MODE N. A primary difference is that in MODE E True End of DMA is decoded internally and this causes an interrupt. This feature removes the need to check for DMA End before moving to the next phase. Additionally, the use of the ISR allows easier determination of the end condition. For more information on interrupts see Section 6.

Examples of code are given below.

```
Initiator Receive  /* DATA IN PHASE */
{
  Program DMA controller;
  TCR = 01h;   /* phase */
  ICR = 40h;   /* MODE E */
  MR2 = BEh;
  EMR = 04h;   /* Start DMA INIT Rx */
  while (NOT interrupt)
        idle;
  if (ISR ! = 20h)
        error;
else          /* DMA end */
  MR2 = 04h;
}

Target Send    /* DATA IN PHASE */
{
  Program DMA controller;
  TCR = 01h;   /* phase */
  ICR = 49h;   /* MODE E */
  MR2 = CAh;
  SDS = 0;     /* Start DMA Send */
  while (NOT interrupt)
        idle;
  if (ISR! = 20h)
        error;
  else  {      /* DMA end */
  MR2 = 40h;
  Change phase as required;
  }
}
```

4.9 BLOCK MODE DMA TRANSFERS

In Block Mode the DMA interface uses the DRQ, $\overline{DACK}$, $\overline{EOP}$ and READY lines, DRQ is asserted once at the begin-



TL/F/9387–8

FIGURE 4.8.2. Target Send DMA

4.0 Device Operation (Continued)

ning of transfers and deasserted once $\overline{DACK}$ is received. $\overline{DACK}$ should be asserted continuously for the duration of all the transfer. $\overline{EOP}$ should be asserted during the last DMA byte signal when the next DMA byte transfers. The EASI asserts the READY signal when the next DMA byte should be transferred. As for non-block mode the End of DMA interrupt is just $\overline{EOP}$ in MODE N, but is a True End of DMA in MODE E.

The block mode is intended for systems where the overhead of handing the system busses to and from the μP and DMA controller is too great. The block mode handshake is not necessarily faster than non-block (it may be) but the overall transfer rate is improved once the bus exchange overhead is removed. Of course the μP is prevented from executing for the whole DMA operation.

If a phase mismatch occurs the READY signal is left in the inactive state. The DMA controller must hand back the bus to the μP and the inactive READY signal may need to be gated off. For more detail see Section 9.

When performing DMA as an INITIATOR the $\overline{EOP}$ signal does not deassert $\overline{ACK}$ on the SCSI bus in MODE N. Firmware must determine when $\overline{REQ}$ is inactive after the last SCSI transfer then reset the MR2 DMA bit to deassert $\overline{ACK}$. In MODE E the EASI correctly handles the deassertion of $\overline{ACK}$ after True End of DMA.

Programming the EASI in block mode is the same as non-block mode except bit 7 in MR2 should also be set.

4.10 PSEUDO DMA

The system design can utilize EASI DMA logic for non data transfers. This removes the need to poll $\overline{REQ}$/$\overline{ACK}$ and program the assertion/deassertion of the handshake signal. The μP can emulate a DMA controller by asserting $\overline{DACK}$ and $\overline{EOP}$ signals. DRQ may be sampled by reading the BSR. In most cases the chip decode logic can be adapted to this use for little or no cost. See Section 9 for further details.

4.11 HALTING A DMA OPERATION

There are three ways to halt a DMA operation apart from a chip or SCSI reset. These methods are: $\overline{EOP}$, phase mismatch and resetting the DMA MODE bit in MR2.

4.11.1 End of Process

$\overline{EOP}$ is asserted for a minimum period during the last DMA cycle. The $\overline{EOP}$ signal generates the End of DMA interrupt in MODE N, and enables the True End of DMA to occur later in MODE E. $\overline{EOP}$ does not cause the MR2 DMA mode bit to be reset.

4.11.2 DMA Phase Mismatch

If a $\overline{REQ}$ goes active while there is a phase mismatch the DMA will be halted and an interrupt generated. The EASI will stop driving the SCSI bus when the mismatch occurs. A phase mismatch is when the TCR phase bits do not match the SCSI bus values.

4.11.3 DMA Mode Bit

If $\overline{EOP}$ is not used, the best method is to reset the MR2 DMA Mode bit. This bit may be reset at any time, and should be reset after an End of DMA interrupt or a phase mismatch. Resetting the bit disables all DMA logic and thus should only be reset at the True End of DMA condition. Additionally, all DMA logic is reset, so this bit must be reset then set again to carry out the next DMA phase.

5.0 Interrupts

5.1 OVERVIEW

The EASI is intended to be used in an interrupt driven environment. MODE E has greatly enhanced the use of interrupts to relieve firmware overhead. This section describes the conditions for and use of each interrupt. Each description explains MODE N then MODE E operation.

Before individually describing each interrupt, an explanation of the use of interrupts is required.

5.2 USING INTERRUPTS

MODE N: In this mode interrupts are controlled by bits in MR2 if control is provided. Not all interrupts can be disabled under software control. When an interrupt occurs, both the BSR and CSD register must be read and analyzed to determine the source of interrupt. Since status is NOT provided for each interrupt, great care should be exercised when determining the interrupt source.

MODE E: In this mode every interrupt can be individually masked and enabled or disabled. In addition, when an interrupt occurs a single register (ISR) can be read to determine the source(s) of interrupt. An associated register (IMR) allows a mask bit to be set for each interrupt. Finally, the design of the logic prevents loss of interrupts occurring after the INT signal goes Active and before the Reset Parity/Interrupt function. In MODE N loss of interrupts can occur.

5.3 SCSI PARITY ERROR

MODE N: If SCSI parity checking is enabled via MR2 bit 5, an interrupt can occur as a result of a read from CSD, a selection/(re)selection, or a DMA receive operation. The parity error bit (bit 5) in the BSR will be set if checking is enabled. An interrupt will occur if Enable Parity Interrupt (bit 4) of MR2 is set. The interrupt is reset by reading HA 7. Following an interrupt the BSR and CSB should contain the values shown below.

Bit 7 **Bit 0**

X	X	1	1	X	X	X	X
EDMA	DRQ	SPER	INT	PHSM	BSY	$\overline{ATN}$	$\overline{ACK}$

BSR

Bit 7 **Bit 0**

0	1	X	X	X	X	0	X
$\overline{RST}$	$\overline{BSY}$	$\overline{REQ}$	$\overline{MSG}$	$\overline{C/D}$	$\overline{I/O}$	$\overline{SEL}$	$\overline{DBP}$

CSB

5.4 μP PARITY ERROR

MODE N: μP parity is not available under this mode.

MODE E: If bit 6 of the EMR is 1, μP parity will be checked on a write to the EASI via the μP or DMA controller. The parity polarity is determined by bit 5 of the EMR. If bit 6 of the IMR is zero, a μP parity error will set bit 6 of the ISR and cause an interrupt. The interrupt may be reset by writing 01 to bits 2 and 1 of the EMR followed by any other pattern.

5.5 END OF DMA

MODE N: If $\overline{EOP}$ is asserted during a DMA transfer, bit 7 of the BSR will be set and an interrupt generated if bit 3 of MR2 is 1. $\overline{EOP}$ is recognized when $\overline{EOP}$, $\overline{DACK}$ and either $\overline{IOR}$ or $\overline{IOW}$ are all simultaneously active for a minimum period. The interrupt may be reset by reading HA 7. Following an interrupt the BSR and CSB should contain the values shown below.

5.0 Interrupts (Continued)

Bit 7							Bit 0
1	X	X	1	X	X	0	X
EDMA	DRQ	SPER	INT	PHSM	BSY	$\overline{ATN}$	$\overline{ACK}$

BSR

Bit 7							Bit 0
0	1	X	X	X	X	0	X
$\overline{RST}$	$\overline{BSY}$	$\overline{REQ}$	$\overline{MSG}$	C/$\overline{D}$	I/$\overline{O}$	$\overline{SEL}$	$\overline{DBP}$

CSB

5.6 DMA PHASE MISMATCH

MODE N: When the SCSI $\overline{REQ}$ goes active during a DMA operation the contents of the TCR are compared with the SCSI phase lines C/$\overline{D}$, $\overline{MSG}$ and I/$\overline{O}$. If the two do not match an interrupt is generated. This interrupt will occur as long as the MR2 DMA bit is set (bit 1), i.e., it cannot be masked in MODE N. The mismatch removes the EASI from driving the SCSI data bus. The interrupt may reset by reading HA 7. Following an interrupt the BSR and CSB should contain the values shown below.

Bit 7							Bit 0
X	0	X	1	0	X	X	X
EDMA	DRQ	SPER	INT	PHSM	BSY	$\overline{ATN}$	$\overline{ACK}$

BSR

Bit 7							Bit 0
0	X	X	X	X	X	0	X
$\overline{RST}$	$\overline{BSY}$	$\overline{REQ}$	$\overline{MSG}$	C/$\overline{D}$	I/$\overline{O}$	$\overline{SEL}$	$\overline{DBP}$

CSB

MODE E: Bit 4 of the ISR will be set by a DMA phase mismatch—the same conditions as MODE N. The interrupt and setting if ISR bit 4 may be masked by setting bit 4 of the IMR. The interrupt may be reset by writing 01 to bits 2 and 1 of the EMR followed by any other pattern.

5.7 ANY PHASE MISMATCH

MODE N: This feature is not available under this mode.

MODE E: This condition is similar to DMA Phase Mismatch except that it applies to all operations—not just DMA. If the TCR contents do not match the SCSI phase lines when $\overline{REQ}$ goes active an interrupt is generated and bit 3 set in the ISR. The ISR bit and the interrupt may be masked by setting bit 3 in the IMR. The interrupt may be reset by writing 01 to bits 2 and 1 of the EMR followed by any other pattern.

5.8 BUSY LOSS

MODE N: If bit 2 in MR2 is set the SCSI $\overline{BSY}$ signal is monitored and an interrupt is generated if $\overline{BSY}$ is continuously inactive for at least a BUS SETTLE DELAY (400 ns). This interrupt may be reset by reading HA 7. Following an interrupt the BSR and CSB should contain the values shown below, where usually CSB = 00.

Bit 7							Bit 0
X	X	X	1	X	1	X	X
EDMA	DRQ	SPER	INT	PHSM	BSY	$\overline{ATN}$	$\overline{ACK}$

BSR

Bit 7							Bit 0
0	0	X	X	X	X	X	X
$\overline{RST}$	$\overline{BSY}$	$\overline{REQ}$	$\overline{MSG}$	C/$\overline{D}$	I/$\overline{O}$	$\overline{SEL}$	$\overline{DBP}$

CSB

MODE E: Bit 2 in MR2 performs the same function as in MODE N. When an interrupt is generated bit 2 in the ISR will be set. The ISR bit and the interrupt may be masked by setting bit 2 in the IMR. The interrupt may be reset by writing 01 to bits 2 and 1 of the EMR followed by any other pattern.

5.9 (RE)SELECTION

MODE N: An interrupt will be generated when: $\overline{SEL}$ is active, $\overline{BSY}$ is inactive, and the device ID is true. The device ID is determined by the value in the SER. If ANY non-zero bit in the SER has its corresponding SCSI data bit active during selection, the device ID is true. If I/$\overline{O}$ is active this is a reselection. The interrupt is disabled by writing all zeroes to the SER, and reset by reading HA 7.

If SCSI parity checking is enabled it will be checked and should be valid. Following an interrupt the BSR and CSB should contain the values shown below.

Bit 7							Bit 0
0	0	0	1	X	0	X	0
EDMA	DRQ	SPER	INT	PHSM	BSY	$\overline{ATN}$	$\overline{ACK}$

BSR

Bit 7							Bit 0
0	0	0	0	0	0	1	X
$\overline{RST}$	$\overline{BSY}$	$\overline{REQ}$	$\overline{MSG}$	C/$\overline{D}$	I/$\overline{O}$	$\overline{SEL}$	$\overline{DBP}$

CSB

MODE E: The functioning of the (re)selection logic is the same as for MODE N. Additionally, bit 1 in the ISR will be set upon a (re)selection interrupt. The interrupt and setting of bit 1 in the ISR may be masked by setting bit 1 in the IMR. The interrupt may be reset by writing all zeroes to the SER then writing 01 to bits 2 and 1 of the EMR followed by any other pattern.

5.10 ARBITRATION COMPLETE

MODE N: No interrupt is generated in MODE N.

MODE E: When bit 0 of the EMR is set to 1 the EASI monitors $\overline{BSY}$ and $\overline{SEL}$ to determine a BUS FREE Phase. The EASI then carries out all steps required for bus arbitration. When either arbitration is lost or the arbitration process has completed, bit 0 in the ISR is set and an interrupt generated. The interrupt and bit 0 in the ISR can be masked by setting bit 0 in the IMR. Note that arbitration is not affected by the IMR, just the interrupt.

6.0 Reset Conditions

6.1 GENERAL

There are three ways to reset the EASI; μP chip $\overline{RESET}$, SCSI bus reset applied externally, SCSI bus reset issued by the EASI.

6.2 CHIP RESET

When the $\overline{RESET}$ signal is asserted for the required duration the EASI clears ALL internal registers and therefore resets all logic. This action does not create an interrupt or generate a SCSI reset. Since all registers contain zeroes,

6.0 Reset Conditions (Continued)

the EASI is in MODE N under any of the three reset conditions.

6.3 EXTERNAL SCSI RESET

When an SCSI $\overline{RST}$ is applied externally the EASI resets all registers and logic and issues an interrupt. The only register bits not affected are the Assert RST bit (bit 7) in the ICR and the TARGET Mode bit (bit 6) in MR2. Note that the ISR will contain all zeroes.

6.4 SCSI RESET ISSUED

When the μP sets the Assert RST bit in the ICR, the $\overline{RST}$ signal goes active. Since the EASI monitors $\overline{RST}$ also, the same reset actions as in 6.2 apply. The SCSI $\overline{RST}$ signal will remain active as long as bit 7 in the ICR is set—i.e., until programmed 0 or a chip $\overline{RESET}$ occurs.

7.0 Loopback Testing

7.1 GENERAL

The DP8490 EASI features loopback testing, enabled by bit 3 of the EMR. When the LOOP bit is set in the EMR all SCSI drivers are disabled at the pads and the signals looped back internally. Additionally, both TARGET and INITIATOR signals may be simultaneously asserted—this is not possible in normal operation. All this enables testing of EASI operation without affecting the SCSI bus.

7.2 SCSI SIGNAL DRIVING/MONITORING

Since each SCSI signal is looped back, testing is accomplished by asserting a signal via the ICR or TCR, then reading back via CSB and BSR to check its state. The code below provides examples. The first example tests SCSI control signals and the second the SCSI data bus. Note that loopback mode must be enabled prior to asserting any signals if they are not to drive the SCSI bus.

```
SCSI_signal_test
{
  ICR = 40h;   /* MODE E */
  EMR = 08h;   /* LOOPback */
  ICR = 5Eh;   /* drive INIT controls */
  TCR = 0Fh;   /* drive TARG controls */
  if((CSB==7Eh) AND (BSR==0Fh))
        ok;
  else error;
}

SCSI_data_test
{
  ICR = 40h;   /* MODE E */
  EMR = 08h;   /* LOOPback */
  ICR = 41h;   /* assert data bus */
  MR2 = 30h;   /* parity check & interrupt */
  ODR = 0AAh;
  if(CSD==0AAh)  ok;
  else  error;
  ODR = 55h;
  if(CSD==55h)  ok;
  else  error;
  if(interrupt)error;  /* no parity errors */
  else ok;
}
```

7.3 (RE)SELECTION AND ARBITRATION

Both these features may be tested in loopback. Note that when checking BSY via the CSB register the "debounced" version of BSY is presented and will be active for 400 ns–800 ns after BSY goes inactive. Logic within the EASI continuously monitors BSY when it becomes inactive to detect a valid Bus Free Phase. One of the outputs of this logic is a clean version of BSY which is accessed through the CSB.

7.4 DATA TRANSFERS

Both programmed-I/O and DMA transfers may be performed. When doing DMA transfers the MR2 TARGET MODE bit (6) must be programmed according to the type of DMA—i.e., set to 1 for TARGET Send or Receive, reset to 0 for INITIATOR Send or Receive. Additionally, the actions of the other SCSI device must be programmed. For example, when testing INITIATOR operations the BSY signal must be set via ICR to simulate a TARGET connected, and the REQ signal must be programmed active and inactive to perform the handshake. The code below shows a single byte transfer as INITIATOR Send.

```
DMA_test
{
  program DMA controller;
  ICR = 40h;    /* MODE E */
  EMR = 08h;    /* LOOPback */
  ICR = 49h;    /* BSY & data bus on */
  MR2 = 0Ah;    /* DMA & EOP interrupt */
  TCR = 08h;    /* assert REQ */
  SDS = 0;      /* Start DMA Send */
  /* DMA cycle with EOP is done here */
  TCR = 0;      /* deassert REQ */
  if(IDR==data byte)  ok;
  else  error;
  if(BSR==90h)  ok;
  else  error;
  /* also should be an EOP interrupt */
}
```

8.0 Extra Features/Compatibility

8.1 OPERATING MODES

This section is intended to clearly indentify the differences between the DP8490 and DP5380. The description covers registers, signals, timing, "bugs" and enhancements. For a more detailed description of register programming and bit functions refer to Sections 3–7.

Before discussing differences a review of DP8490 operation is required. The EASI can be operated in two modes—Normal Mode (MODE N) and Enhanced Mode (MODE E). The EASI is in one or the other mode at any time. MODE E is selected when bit 6 in the INITIATOR COMMAND REGISTER (ICR) is set to 1. A SCSI or external chip reset clears all registers (except MODE REGISTER 2 bit 6) so MODE N is selected as the default. MODE N may also be selected at any time by clearing bit 6 of the ICR to zero. If MODE E

8.0 Extra Features/Compatibility (Continued)

functions have been invoked, selecting MODE N does not in general affect their operation. This means MODE E is used to enable selection of enhanced features which can be used in either mode. Normally an application will choose to remain in MODE E always since all functions are accessible in this mode. MODE N is only required for downward compatibility with a standard 5380 device. The DP5380 operates ONLY in MODE N.

The MODE selection via bit 6 of the ICR enables existing 5380 firmware to run unaltered with the EASI. On the 5380 this is a TEST MODE bit which disables ALL output drivers—NOTHING ELSE! No logic is "tested" and no changes are made to internal logic. Since the μP data bus drivers are also disabled the internal state of the 5380 is inaccessible. This means the TEST MODE bit has very limited application for the end user. By comparison the LOOPBACK bit in the EASI enables thorough testing of device operation.

In summary, current operating firmware should not be using bit 6 of the ICR so the DP8490 uses this bit to enable enhanced operation. *As long as this bit is not set in the EASI the DP8490 appears the same as the NCR, AMD or DP5380. Programming, device operation, and timing sequences are all the same as a 5380.*

8.2 INTERNAL REGISTERS

Figure 8.1 shows the register map of the EASI. Note that in MODE N hex address 7 (HA 7) causes a Parity/Interrupt Reset when read and a Start DMA Initiator Receive when written. In MODE E the read/write EXTRA MODE REGISTER (EMR) is mapped into HA 7. Since the two original functions at HA 7 would conflict, their functions are reproduced by writing to bits 1 and 2 of the EMR. This removes any need to switch between modes. The INTERRUPT MASK REGISTER (IMR) and the INTERRUPT STATUS REGISTER (ISR) are accessed indirectly via the EMR. Setting both bits 1 and 2 of the EMR to 1 enables the next read of address 7 to access the ISR or the next write to access the IMR. Once the access is made subsequent uses of address 7 use the EMR.

Hex Addr	Read Register	Write Register
00	Current SCSI Data	Output Data Register
01	Initiator Command	Initiator Command
02	Mode Register 2	Mode Register 2
03	Target Command	Target Command
04	Current SCSI Bus Status	Select Enable Register
05	Bus and Status	Start DMA Send
06	Input Data Register	Start DMA Targ Rx
07	Reset Parity/Interrupt (MODE N)	Start DMA Init Rx (MODE N)
07	Extra Mode Register (MODE E)	Extra Mode Register (MODE E)
07	Interrupt Status Reg (MODE E and bits 1 and 2 of EMR = 1)	Interrupt Mask Reg (MODE E and bits 1 and 2 of EMR = 1)

FIGURE 8.1. EASI Register Map

8.3 ENHANCEMENT MODE REGISTERS

MODE E provides three new registers to provide control for the extra features of the EASI. The EXTRA MODE REGISTER (EMR) is a read/write register which enables or disables the extra functions. Setting an EMR bit to 1 enables the corresponding function while programming a 0 disables it. Note on power-up or reset the EMR is zeroed so all extra functions are disabled. The INTERRUPT MASK REGISTER (IMR) provides the ability to individually mask out interrupts. The INTERRUPT STATUS REGISTER (ISR) shows the status of the interrupt system at any time. When an interrupt occurs the ISR will contain 1's for interrupts that are active, enabled and not masked. *Figure 8.2* shows the format of the three registers.

8.4 EMR FEATURES

8.4.1 Arbitration

In MODE N, arbitration is enabled via bit 0 of MR2. However this requires polling the device for potentially many milliseconds. The μP polls the ICR for the ARBITRATION IN PROGRESS bit (6) which is set once the ASI/EASI detects a bus free phase and enters arbitration. The μP must then wait for the 2.2 μs ARBITRATION DELAY before checking IDs on the bus. No interrupt is given for any of these events.

In MODE E, EASI arbitration is enabled by bit 0 of either MR2 or EMR. The EMR bit enables extended arbitration. The EASI will wait for bus free, arbitrate, wait the 2.2 μs ARBITRATION DELAY and the INTERRUPT the μP if not masked in the IMR. An interrupt occurs if arbitration is complete or has been lost. This feature removes the need to poll the EASI.

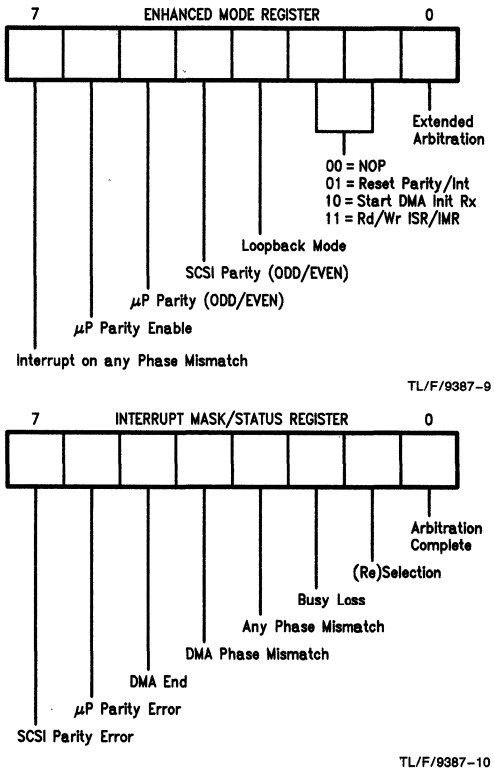


FIGURE 8.2. MODE E Registers

8.0 Extra Features/Compatibility (Continued)

8.4.2 Loopback Mode

When bit 3 is set in the EMR the EASI disables all SCSI drivers and loops back the signals internally. SCSI Data Out is linked to SCSI Data In. $\overline{BSY}$, $\overline{SEL}$, $\overline{ATN}$, $\overline{RST}$, $\overline{I/O}$, $\overline{C/D}$, $\overline{MSG}$, $\overline{REQ}$ and $\overline{ACK}$ outputs are fed back to their own inputs. This enables testing of EASI operations, including DMA.

8.4.3 SCSI Parity

SCSI parity may be enabled and also cause an interrupt (or be masked). Bit 4 of the EMR allows the polarity of SCSI parity to be set to either ODD or EVEN, with a default of ODD. This feature allows checking of SCSI devices and cable. It also allows an INITIATOR to automatically detect whether a device supports parity. (Send a parity error and check it is reported).

8.4.4 μP Parity

The EASI includes parity for the μP data bus. This enables controllers to validate data while it passes through their data buffers. In common with SCSI parity the μP parity may be: enabled/disabled, interrupt or be masked, be EVEN or ODD polarity.

8.4.5 Phase Mismatch

In MODE N the EASI will only give a Phase Mismatch interrupt during DMA (normally data) transfers. In MODE E an interrupt can also be programmed for a phase mismatch during any phase. The mismatch is detected if $\overline{REQ}$ is active and the phase lines do not match the phase expected as specified in the TCR bits 0 to 2. This feature allows completely interrupt-driven operation of the EASI.

8.5 INTERRUPTS

The EASI has an improved interrupt structure to ease programming. In MODE N the structure is the same as a standard 5380. In this mode interrupts can occur which may only be determined by the lack of other interrupts (e.g., Selection and Phase Mismatch). Interrupts can also be missed if they occur while servicing others. To determine the cause of an interrupt requires reading the BSR and CSB registers then interpreting the results, knowing what the 5380 was currently doing.

In MODE E interrupts can be individually masked via the IMR and active ones determined by reading the ISR. Any unmasked interrupts sets an ISR bit to 1. After the interrupt service routine the μP should perform a Reset Parity/Interrupt function via the EMR—this will ONLY RESET interrupts which had a 1 in the ISR when it was last read. This feature means interrupts will not be lost.

8.6 TIMING

The NCR5380 timing has some aspects which have been improved for MODE E of the EASI. In MODE N timing details are the same as the 5380. Of course the DP5380 ASI remains the same as the NCR5380. The timing improvements are listed below.

1. **True End of DMA:** In MODE N the end of DMA is when the last byte is transferred by the DMA controller, not when the final SCSI transfer is done. In MODE E bit 7 of the TCR shows a true end of DMA status—i.e., the last byte transferred by the later of the two events. This bit is compatible with the NCR CMOS 53C80.
2. **SCSI Handshake after EOP:** in MODE N, INITIATOR receive when a $\overline{REQ}$ is received after $\overline{EOP}$ has been given an $\overline{ACK}$ will be generated although no valid data exists since no DRQ was issued. The EASI will NOT generate this invalid $\overline{ACK}$ while in MODE E.

μP and DMA accesses have relaxed timing on both the EASI and ASI. Data setup/hold and read access times are reduced. Faster handshaking on the SCSI bus, along with faster response to the DMA signals means that higher transfer rates are possible with both devices—typically over 3 MBytes/s.

9.0 Application Guide

This section is intended to show the interface between the μP, EASI and DMA controller (DMAC). *Figure 9.1* shows a general interface when the EASI and DMAC are I/O-mapped devices. This configuration will implement a 2 to 2.5 MBytes/s SCSI port using 2 cycle compressed timing from the 5 MHz DMAC.

Using a faster DMAC and memory may allow the EASI to operate at a higher rate—but of course any system will be limited by the available DMA rate from the SCSI device currently connected to. The interface shown has several features that are examined more closely in the following text.

All the interface signal requirements are satisfied by a PAL device. The memory interface is not shown, only the relevant DMAC and μP lines are included.

The EASI data and address lines connect directly to the μP/DMAC busses. The DRQ output from the EASI goes direct to the DMAC. The $\overline{EOP}$ output from the DMAC goes to the EASI input, via the PAL, but can also be asserted via the PAL since the DMAC output is open-drain.

The PAL is programmed so that the μP can access the EASI in three ways. The three access types are: Register R/W, DMA R/W, DMA with $\overline{EOP}$. Examination of the PAL equations below shows how the μP may perform any of the three basic access types simply by accessing the EASI at different I/O address slots. This enables the μP to simulate a DMAC (pseudo-DMA). DMA mode may then be used for all information transfer phases.

In DMA mode the EASI generates all SCSI handshakes. At all other times the μP is responsible for $\overline{REQ}/\overline{ACK}$ handshakes. Using pseudo-DMA may reduce μP overhead.

When doing DMA transfers via BLOCK MODE and an error occurs, the EASI may not deassert the READY signal. For some DMA controllers this may lock the bus, so the PAL asserts READY and $\overline{EOP}$ to the DMA if an interrupt occurs while READY is false. This completes the current DMA cycle and prevents further DMA for the rest of the block thus allowing the bus to be handed back to the μP for servicing.

The PAL generates $\overline{IOR}$ and $\overline{IOW}$ strobes while the μP is bus master, but the DMAC provides the strobes while it is bus master so the PAL outputs are TRI-STATE.

The PAL details are shown in *Figure 9.2* with the signal definitions and equations following.

9.0 Application Guide (Continued)

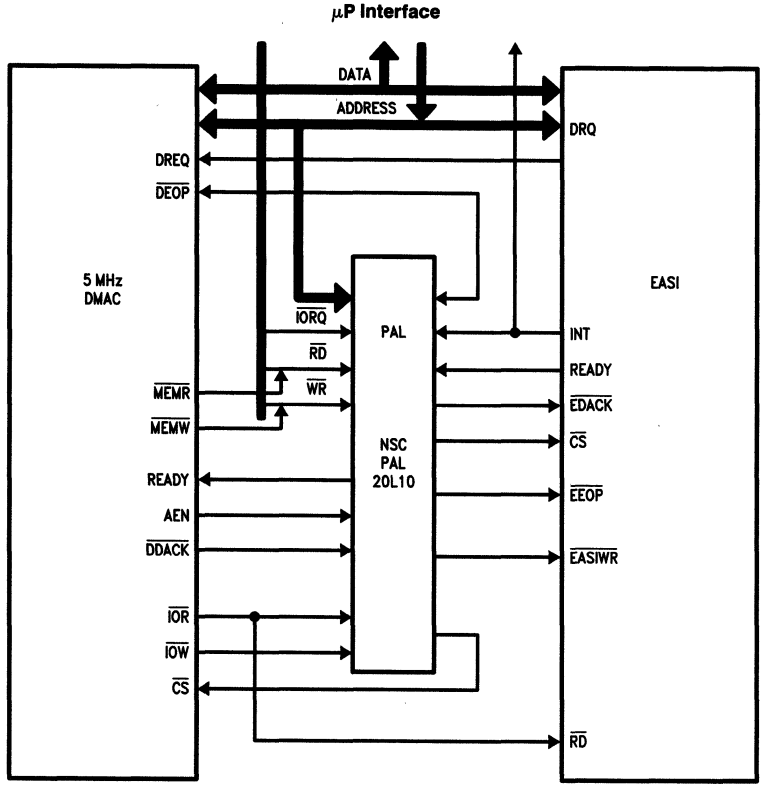


FIGURE 9.1. μP/EASI/DMA Interface

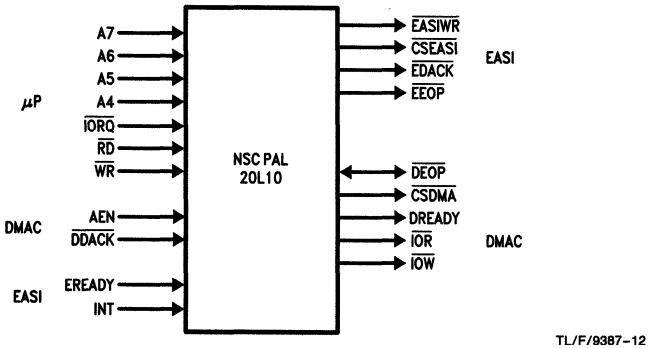


FIGURE 9.2. Interface PAL

4

9.0 Application Guide (Continued)

```
/CSEASI = /IORQ*/A7*/A6*/A5*/A4*/AEN          ; EASI reg R/W chip select
/EDACK =  /IORQ*/A7*/A6*/A5* A4*/RD           ; µP pseudo-DMA cycle
          /IORQ*/A7*/A6*/A5* A4*/WR
          +/IORQ*/A7*/A6* A5*/A4*/RD          ; µP pseudo-DMA with EOP
          +/IORQ*/A7*/A6* A5* A4*/WR
          +/DDACK                             ; DMAC DMA cycle
/EEOP = /IORQ*/A7*/A6* A5*/A4*/RD*/AEN        ; µP pseudo-DMA with EOP
        +/IORQ*/A7*/A6* A5*/A4*/WR*/AEN
        +/DEOP*EREADY                         ; Prevents EASI from seeing
                                              ; EOP until READY goes high.
IF(/DDACK*/EREADY*INT)/DEOP = /DDACK*/EREADY*INT; on error this will terminate the DMA
transfer.
/CSDMA = /IORQ*/A7*/A6*A5*A4                  ; DMAC register R/W
/DREADY = /EREADY*/INT                        ; EASI not READY and not INT
        +/EREADY*/DDACK                       ; EASI not READY and DMA cycle active
IF(/AEN) /IOR   = /IORQ*/RD                   ; µP I/O Read cycle
IF(/AEN) /IOW   = /IORQ*/WR                   ; µP I/O Write cycle
/EASIWR = /IORQ* WR*/AEN+/IOW*EREADY*AEN      ; Prevents SCSI data being
                                              ; changed before EASI is
                                              ; READY for next byte.
```

FIGURE 9.3. PAL Equations

The µP and DMA signals are defined below

A7–A4	Address bus
$\overline{\text{IORQ}}$	Memory I/O cycle select
$\overline{\text{RD}}$	Read Strobe
$\overline{\text{WR}}$	Write Strobe
AEN	High DMA address enable asserted by DMAC
$\overline{\text{DDACK}}$	DMAC DMA Acknowledge
$\overline{\text{CSDMA}}$	DMA Chip Select
DREADY	Ready signal to DMAC—inserts wait-states when low
$\overline{\text{IOR}}$, $\overline{\text{IOW}}$	I/O data strobes to/from DMAC
$\overline{\text{EASIWR}}$	EASI write strobe.

10.0 Absolute Maximum Ratings*

If Military/Aerospace specified devices are required, please contact the National Semiconductor Sales Office/Distributors for availability and specifications.

Supply Voltage (V_{CC})	−0.5V to +7.0V
DC Input Voltage (V_{IN})	−0.5V to V_{CC} + 0.5V
DC Output Voltage (V_{OUT})	−0.5V to V_{CC} + 0.5V
Storage Temperature Range (T_{STG})	−65°C to +150°C
Power Dissipation (P_D)	500 mW
Lead Temperature (T_L) (Soldering, 10 seconds)	260°C
Electro-Static Discharge Rating	2 kV

*Absolute maximum ratings are those values beyond which damage to the device may occur.

11.0 DC Electrical Characteristics

T_A = 0°C to +70°C, V_{CC} = 5.0V ±5% unless otherwise specified

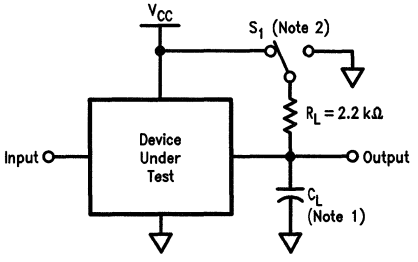
Symbol	Parameter	Conditions	Typ	Limit	Units
V_{IH}	Minimum High Level Input Voltage			2.0	V
V_{IL}	Maximum Low Level Input Voltage			0.8	V
V_{OH1} V_{OH2}	Minimum High Level Output Voltage	$\|I_{OUT}\|$ = 20 μA $\|I_{OUT}\|$ = 4.0 mA		V_{CC} − 0.1 2.4	V V
V_{OL1} V_{OL2} V_{OL3}	Maximum Low Level Output Voltage	SCSI Bus Pins: $\|I_{OL}\|$ = 48 mA Other Pins: $\|I_{OL}\|$ = 20 μA $\|I_{OL}\|$ = 8.0 mA		0.5 0.1 0.4	V V V
I_{IN}	Maximum Input Current	V_{IN} = V_{CC} or GND		±1	μA
I_{OZ}	Maximum TRI-STATE Output Leakage Current	V_{OUT} = V_{CC} or GND		±10	μA
I_{CC}	Supply Current	V_{IN} = V_{CC} or GND SCSI Inputs = 3V	2.5	4	mA

Capacitance T_A = 25°C, f = 1 MHz

Symbol	Parameter (Note 3)	Typ	Units
C_{IN}	Input Capacitance	5	pF
C_{OUT}	Output Capacitance	7	pF

AC Test Conditions

Input Pulse Levels	GND to 3.0V
Input Rise and Fall Time	6 ns
Input/Output Reference Levels	1.3V
TRI-STATE Reference Levels (Note 2)	active low + 0.5V
	active high − 0.5V



TL/F/9387–13

Note 1: C_L = 50 pF including jig and scope capacitance.

Note 2: S1 = Open for push-pull outputs.
S1 = V_{CC} for active low to TRI-STATE.
S1 = GND for active high to TRI-STATE.

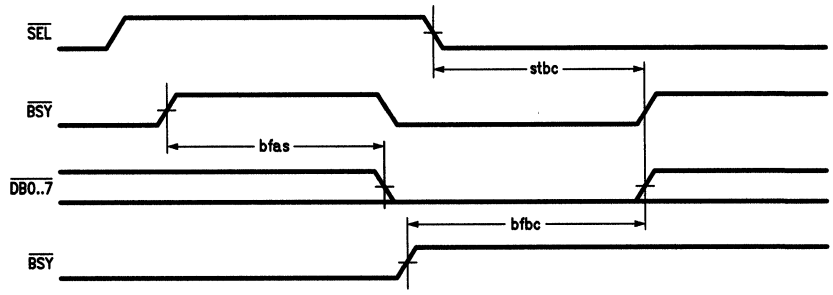
Note 3: This parameter is not 100% tested.

12.0 AC Electrical Characteristics

All parameters are preliminary and subject to change without notice

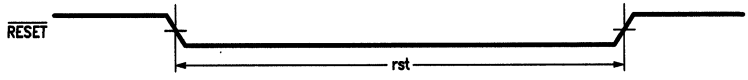
Symbol	Parameter	DP8490			Units
		Min	Typ	Max	
bfas	BSY false to arbitrate start	1200		2200	ns
bfbc	BSY false to bus clear			800	ns
stbc	SEL true to bus clear			500	ns
rst	RESET pulse width	100			ns

12.1 ARBITRATION



TL/F/9387–14

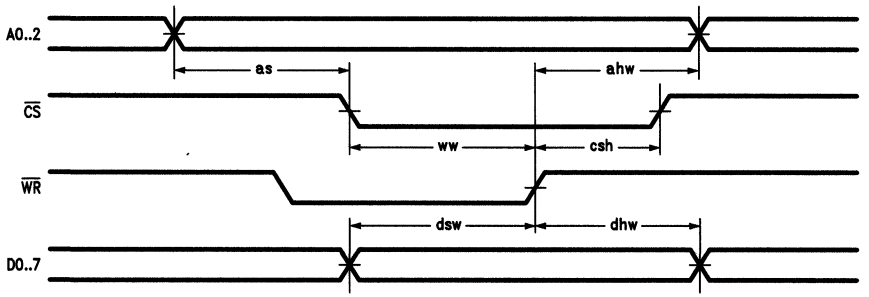
12.2 μP RESET



TL/F/9387–15

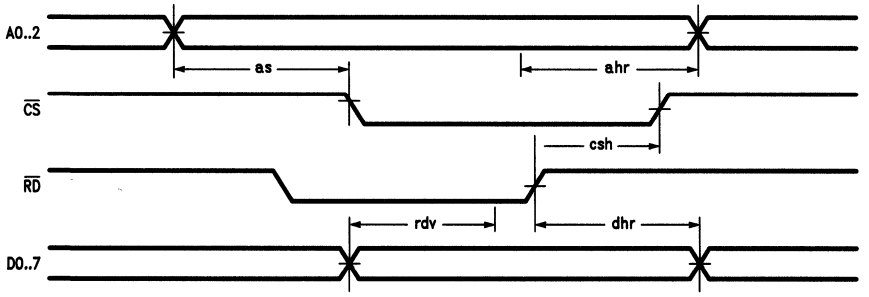
12.0 Electrical Characteristics

12.3 μP WRITE



TL/F/9387–16

12.4 μP READ



TL/F/9387–17

Symbol	Parameter	DP8490			Units
		Min	Typ	Max	
ahr	Address Hold from and of Read Enable (Note 1)	5			ns
ahw	Address Hold from End of Write Enable (Note 2)	5			ns
as	Address Setup to Read or Write Enable (Notes 1,2)	5			ns
csh	CS Hold from End of RD or WR	0			ns
dhr	Data Hold from End of Read Enable (Notes 1,3)	20		60	ns
dhw	μP Data Hold Time from End of WR	10			ns
dsw	Data Setup to End of Write Enable (no μP Parity) (Note 2) (with μP Parity)	35 35			ns ns
rdv	Data Valid from Read Enable (Note 1)			50	ns
ww	Write Enable Width (Note 2)	40			ns

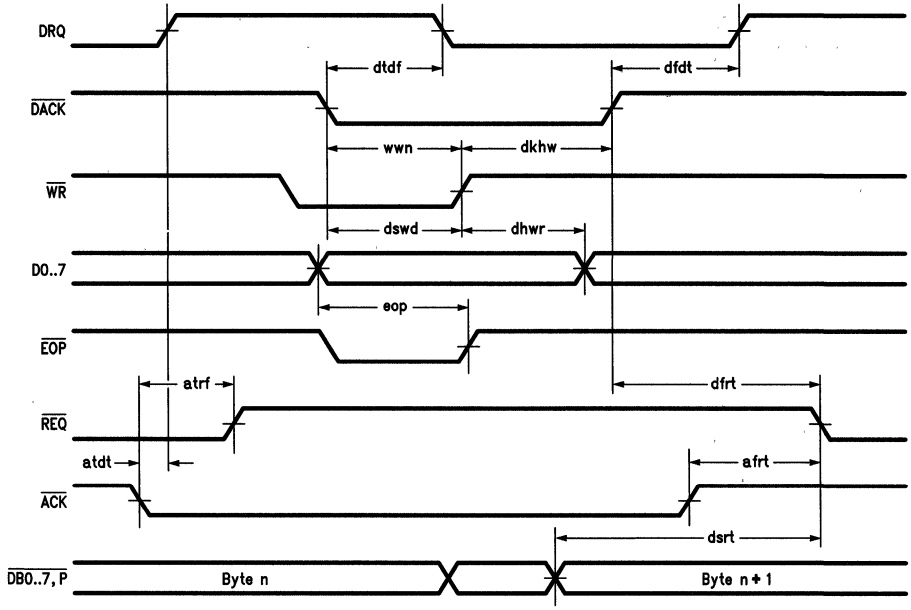
Note 1: Read enable (μP) is CS and RD active.

Note 2: Write enable (μP) is CS and WR active.

Note 3: This includes the RC delay inherent in the test's method. These signals typically turn off after 25 ns, enabling other devices to drive these lines with no contention.

12.0 AC Electrical Characteristics (Continued)

12.5 DMA WRITE (NON-BLOCK MODE) TARGET SEND



TL/F/9387–18

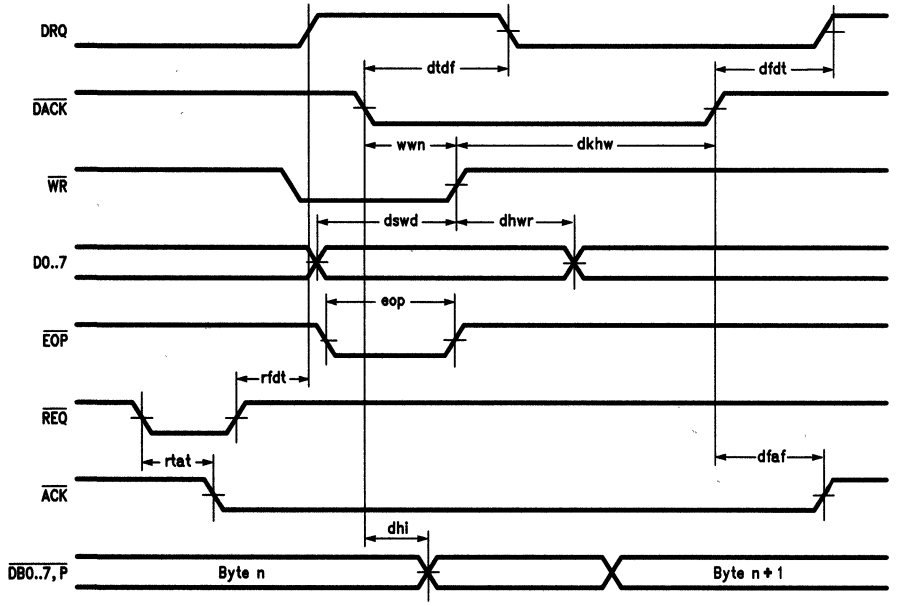
Symbol	Parameter	DP8490			Units
		Min	Typ	Max	
afrt	ACK False to REQ True (DACK or WR False)			75	ns
atdt	ACK True to DRQ True			55	ns
atrf	ACK True to REQ False			100	ns
dfdt	DACK False to DRQ True	30	90		ns
dfrt	DACK False to REQ True (ACK False)			75	ns
dhwr	DMA Data Hold Time from End of WR	10			ns
dkhw	DACK Hold from End of WR	0			ns
dsrt	SCSI Data Setup to REQ True	25			ns
dswd	Data Setup to End of DMA Write Enable (no μP Parity) (Note 1) (with μP Parity)	35 35			ns ns
dtdf	DACK True to DRQ False			45	ns
eop	Width of EOP Pulse (Note 2)	25			ns
wwn	DMA Non-block Mode Write Enable Width (Note 2)	40			ns

Note 1: Write enable (DMA) is DACK and WR active.

Note 2: EOP, DACK, RD/WR must all be true for recognition of EOP.

12.0 AC Electrical Characteristics (Continued)

12.6 DMA WRITE (NON-BLOCK MODE) INITIATOR SEND



TL/F/9387–19

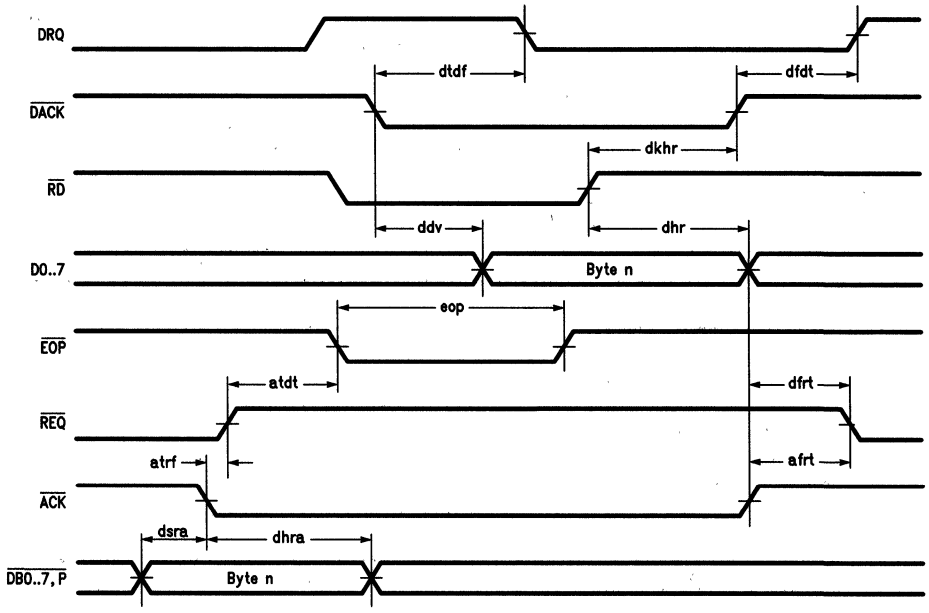
Symbol	Parameter	DP8490			Units
		Min	Typ	Max	
dfaf	DACK False to ACK False			90	ns
dfdt	DACK False to DRQ True	30	90		ns
dhi	SCSI Data Hold from Write Enable-Initiator	15			ns
dhwr	DMA Data Hold Time from End of WR	10			ns
dkhw	DACK Hold from End of WR	0			ns
dswd	Data Setup to End of DMA Write Enable (no μP Parity) (Note 1) (with μP Parity)	35 35			ns ns
dtdf	DACK True to DRQ False			45	ns
eop	Width of EOP Pulse (Note 2)	25			ns
rfdt	REQ False to DRQ True			60	ns
rtat	REQ True to ACK True			80	ns
wwn	DMA Write Enable Width (Note 1)	40			ns

Note 1: Write enable (DMA) is DACK and WR active.

Note 2: EOP, DACK, RD/WR must all be true for recognition of EOP.

12.0 AC Electrical Characteristics (Continued)

12.7 DMA READ (NON-BLOCK MODE) TARGET RECEIVE



TL/F/9387–20

Symbol	Parameter	DP8490			Units
		Min	Typ	Max	
afrt	ACK False to REQ True (DACK or WR False)			75	ns
atdt	ACK True to DRQ True			55	ns
atrf	ACK True to REQ False			100	ns
ddv	DMA Data Valid from Read Enable (Note 1)			40	ns
dfdt	DACK False to DRQ True	30	90		ns
dfrt	DACK False to REQ True (ACK False)			75	ns
dhr	Data Hold from End of Read Enable (Notes 1, 2)	20		60	ns
dhra	SCSI Data Hold from ACK True	15			ns
dkhr	DACK Hold from End of RD	0			ns
dsra	SCSI Data Setup Time to ACK True	10			ns
dtdf	DACK True to DRQ False			45	ns
eop	Width of EOP Pulse (Note 3)	25			ns

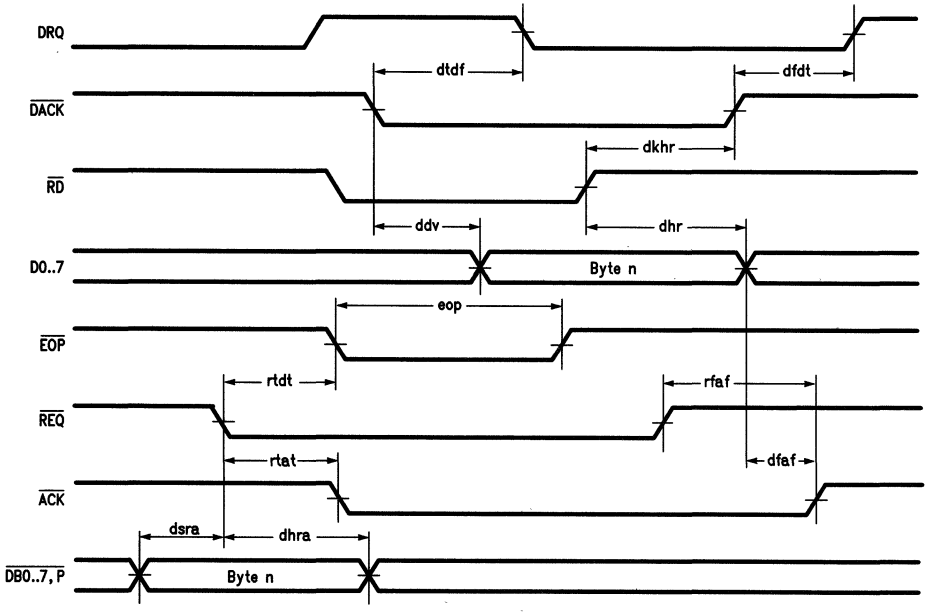
Note 1: Read enable (DMA) is DACK and RD active.

Note 2: This includes the RC delay inherent in the test's method. These signals typically turn off after 25 ns enabling other devices to drive these lines with no contention.

Note 3: EOP, DACK, RD/WR must all be true for recognition of EOP.

12.0 AC Electrical Characteristics (Continued)

12.8 DMA READ (NON-BLOCK MODE) INITIATOR RECEIVE



TL/F/9387–21

Symbol	Parameter	DP8490			Units
		Min	Typ	Max	
ddv	DMA Data Valid from Read Enable (Note 1)			40	ns
dfaf	DACK False to ACK False (REQ False)			90	ns
dfdt	DACK False to DRQ True	30	90		ns
dhr	Data Hold from End of Read Enable (Notes 1, 2)	20		60	ns
dhra	SCSI Data Hold from REQ True	15			ns
dkhr	DACK Hold from End of RD	0			ns
dsra	SCSI Data Setup Time to REQ True	10			ns
dtdf	DACK True to DRQ False			45	ns
eop	Width of EOP Pulse (Note 3)	25			ns
rfaf	REQ False to ACK False (DACK False)			90	ns
rtat	REQ True to ACK True			80	ns
rtdt	REQ True to DRQ True			70	ns

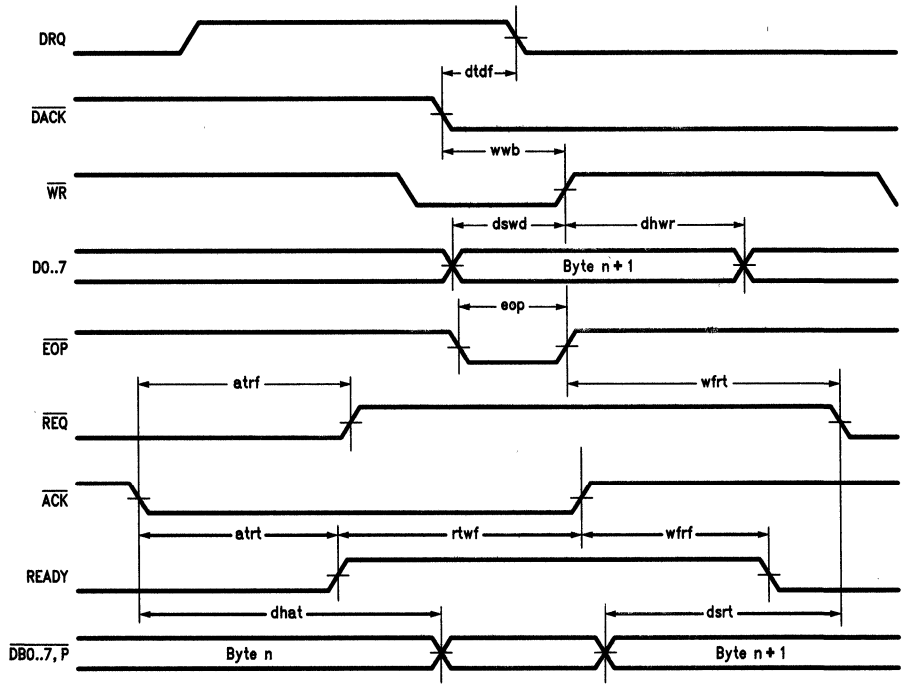
Note 1: Read enable (DMA) is DACK and RD active.

Note 2: This includes the RC delay inherent in the test's method. These signals typically turn off after 25 ns enabling other devices to drive these lines with no contention.

Note 3: EOP, DACK, RD/WR must all be true for recognition of EOP.

12.0 AC Electrical Characteristics (Continued)

12.9 DMA WRITE (BLOCK MODE) TARGET SEND



TL/F/9387–22

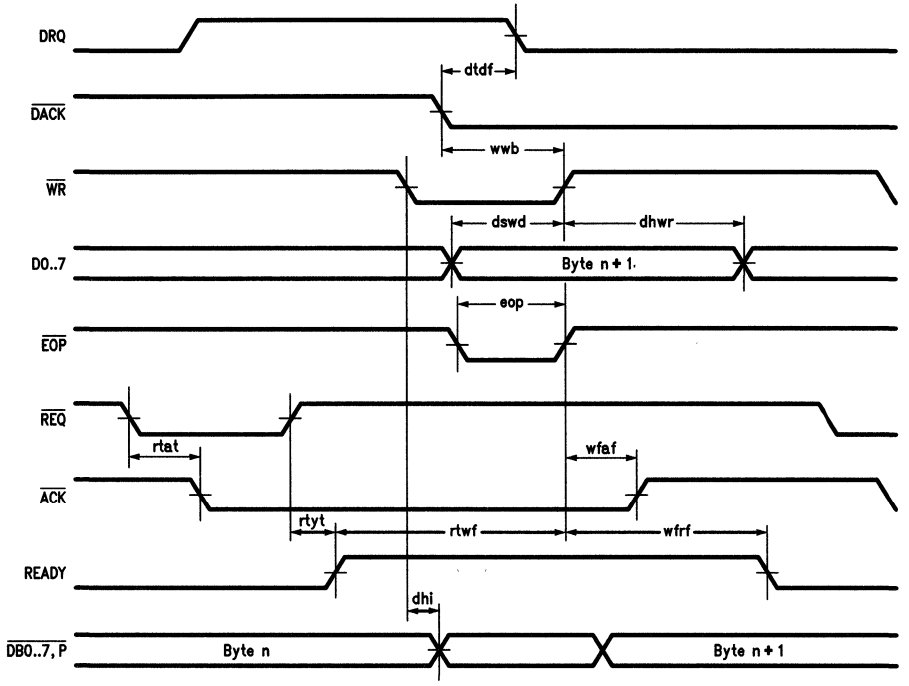
Symbol	Parameter	DP8490			Units
		Min	Typ	Max	
afrt	ACK False to REQ True (DACK or WR False)			75	ns
atrf	ACK True to REQ False			100	ns
atrt	ACK True to READY True			50	ns
dhat	SCSI Data Hold from ACK True	40			ns
dhwr	DMA Data Hold Time from End of WR	10			ns
dsrt	SCSI Data Setup to REQ True	35			ns
dswd	Data Setup to End of DMA Write Enable (no μP Parity) (Note 1) (with μP Parity)	35 35			ns ns
dtdf	DACK True to DRQ False			45	ns
eop	Width of EOP Pulse (Note 2)	25			ns
rtwf	READY True to WR False	40			ns
wfrf	WR False to READY False			50	ns
wfrt	WR False to REQ True (ACK False)			80	ns
wwb	DMA Write Enable Width (Note 1)	40			ns

Note 1: Write enable (DMA) is DACK and WR active.

Note 2: EOP, DACK, RD/WR must all be true for recognition of EOP.

12.0 AC Electrical Characteristics (Continued)

12.10 DMA WRITE (BLOCK MODE) INITIATOR SEND



TL/F/9387–23

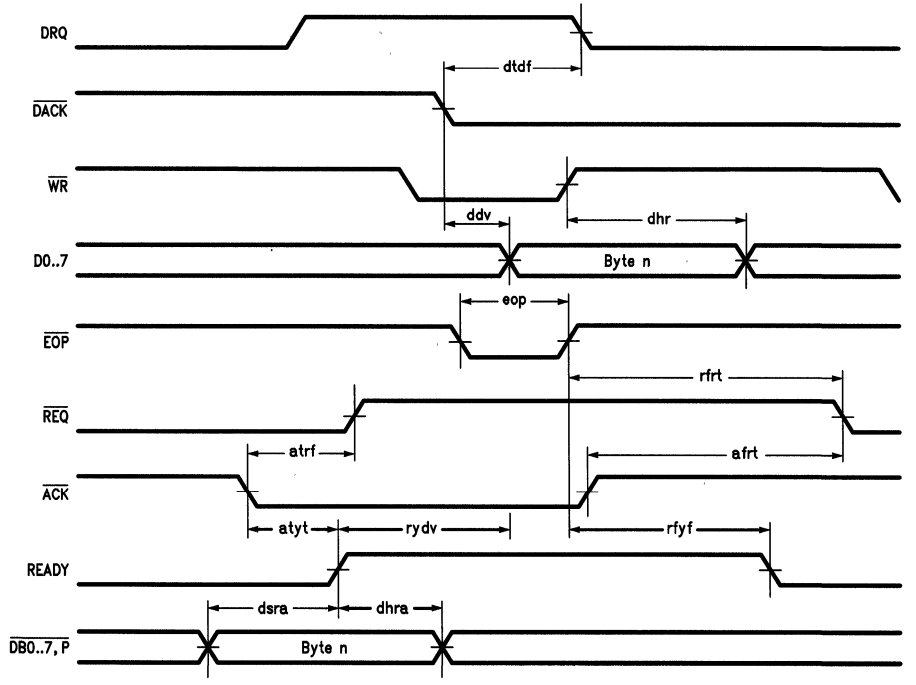
Symbol	Parameter	DP8490			Units
		Min	Typ	Max	
dhi	SCSI Data Hold from Write Enable	15			ns
dhwr	DMA Data Hold Time from End of WR	10			ns
dswd	Data Setup to End of DMA Write Enable (no μP Parity) (Note 1) (with μP Parity)	35 35			ns ns
dtdf	DACK True to DRQ False			45	ns
eop	Width of EOP Pulse (Note 2)	25			ns
rfyt	REQ False to READY True			60	ns
rtat	REQ True to ACK True			80	ns
rtwf	READY True to WR False	40			ns
wfaf	WR False to ACK False (REQ False)			95	ns
wfrf	WR False to READY False			50	ns
wwb	DMA Write Enable Width (Note 1)	40			ns

Note 1: Write enable (DMA) is DACK and WR active.

Note 2: EOP, DACK, RD/WR must all be true for recognition of EOP.

12.0 AC Electrical Characteristics (Continued)

12.11 DMA WRITE (BLOCK MODE) TARGET RECEIVE



TL/F/9387–24

Symbol	Parameter	DP8490			Units
		Min	Typ	Max	
afrt	ACK False to REQ True (DACK or WR False)			75	ns
atrf	ACK True to REQ False			100	ns
atyt	ACK True to READY True			50	ns
ddv	DMA Data Valid from Read Enable (Note 1)			40	ns
dhr	Data Hold from End of Read Enable (Notes 1,2)	20		60	ns
dhra	SCSI Data Hold from ACK True	15			ns
dsra	SCSI Data Setup Time to ACK True	10			ns
dtdf	DACK True to DRQ False			45	ns
eop	Width of EOP Pulse (Note 3)	25			ns
rfrt	RD False to REQ True (ACK False)			75	ns
rfyf	RD False to READY False			45	ns
rydv	READY True to Data Valid			20	ns

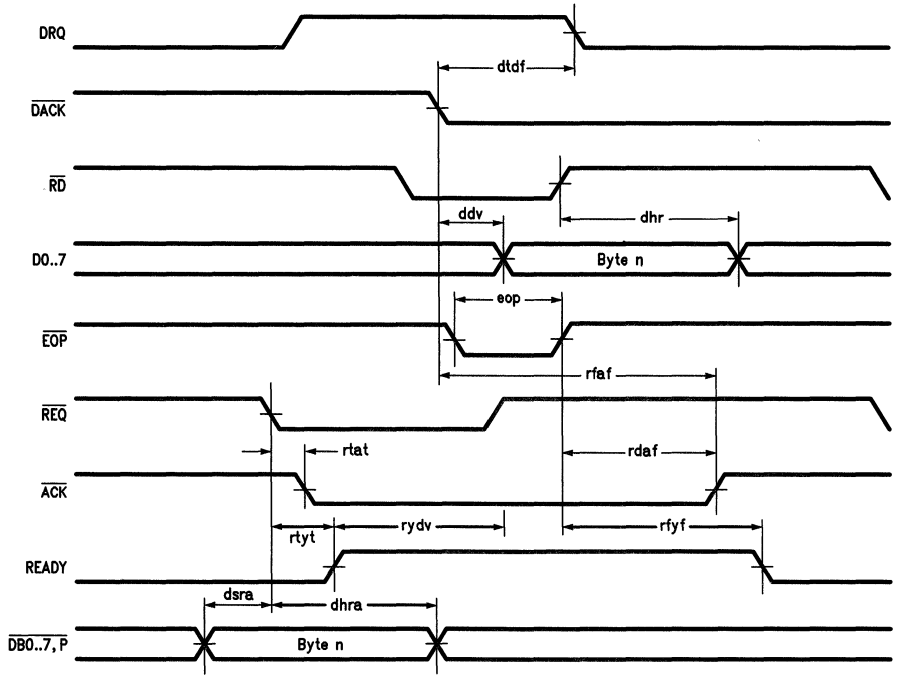
Note 1: Read enable (DMA) is DACK and RD active.

Note 2: This includes the RC delay inherent in the test's method. These signals typically turn off after 25 ns enabling other devices to drive these lines with no contention.

Note 3: EOP, DACK, RD/WR must all be active for recognition of EOP.

12.0 AC Electrical Characteristics (Continued)

12.12 DMA READ (BLOCK MODE) INITIATOR RECEIVE



TL/F/9387–25

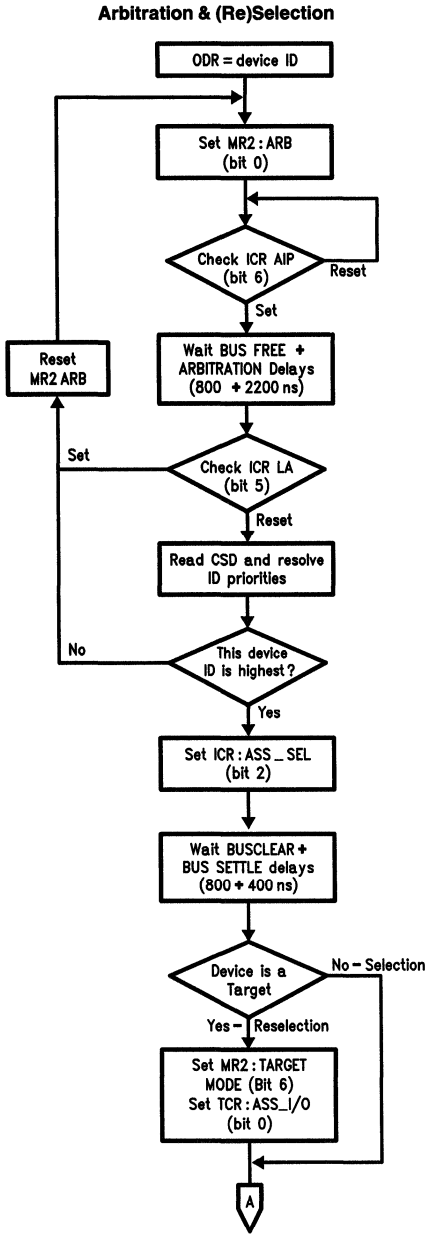
Symbol	Parameter	DP8490			Units
		Min	Typ	Max	
ddv	DMA Data Valid from Read Enable (Note 1)			40	ns
dhr	Data Hold from End of Read Enable (Notes 1,2)	20		60	ns
dhra	SCSI Data Hold from REQ True	15			ns
dsra	SCSI Data Setup Time to REQ True	10			ns
dtdf	DACK True to DRQ False			45	ns
eop	Width of EOP Pulse (Note 3)	25			ns
rdaf	RD False to ACK False (REQ False)			120	ns
rfaf	REQ False to ACK False (DACK False)			90	ns
rfyf	RD False to READY False			45	ns
rtat	REQ True to ACK True			80	ns
rtyt	REQ True to READY True			65	ns
rydv	READY True to Data Valid			20	ns

Note 1: Read enable (DMA) is DACK and RD active.

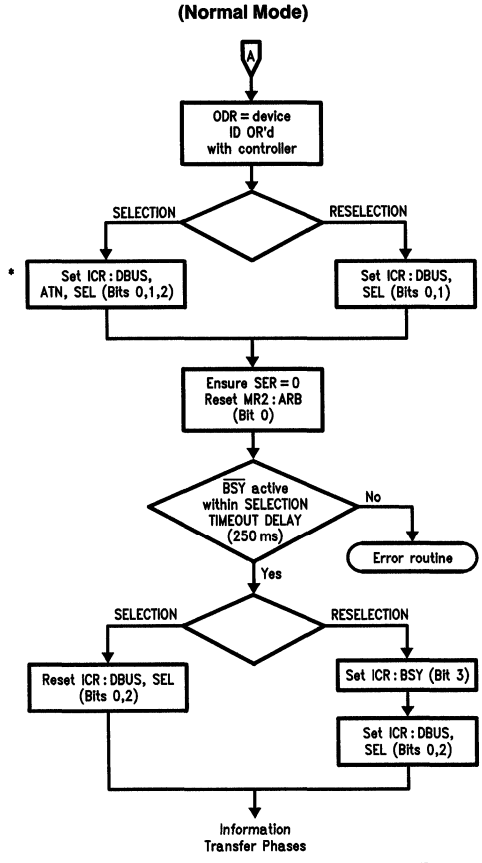
Note 2: This includes the RC delay inherent in the test's method. These signals typically turn off after 25 ns enabling other devices to drive these lines with no contention.

Note 3: EOP, DACK, RD/WR must all be active for recognition of EOP.

Appendix A1



TL/F/9387–26

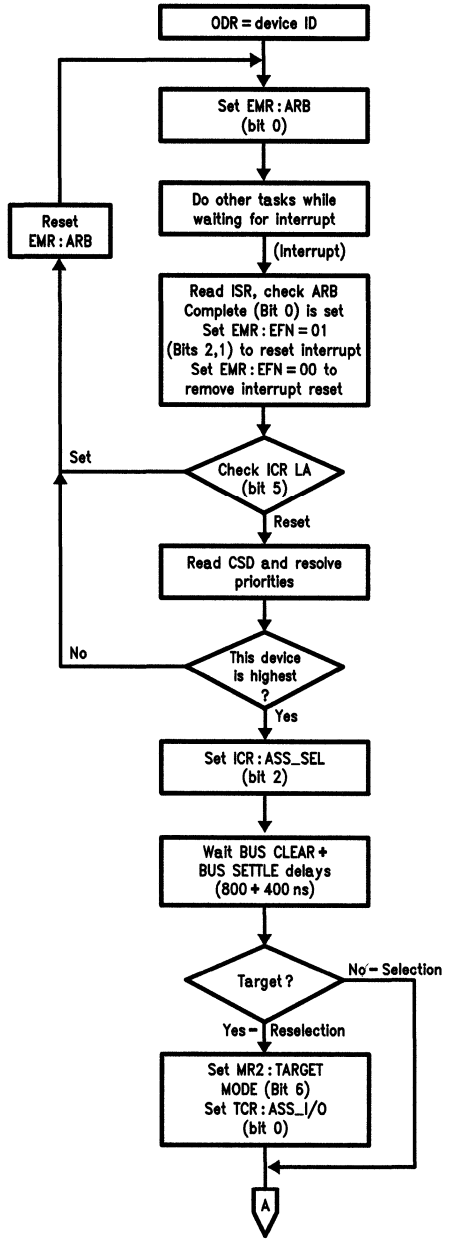


TL/F/9387–27

*Only set ATN if Select with ATN is desired.

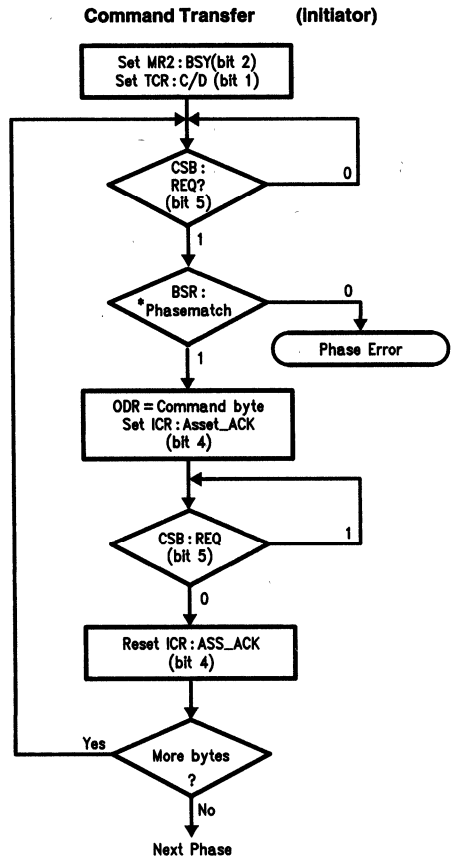
Appendix A1 (Continued)

Arbitration & (Re)Selection (Enhanced Mode)



TL/F/9387–28

Appendix A1 (Continued)

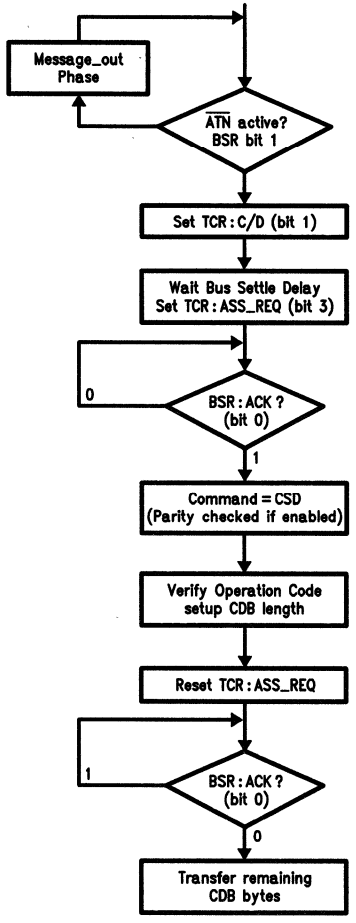


TL/F/9387–29

*This step unnecessary in MODE E if the EMR : APHS (bit 7) is enabled. Logic automatically checks the phase on an error.

Appendix A1 (Continued)

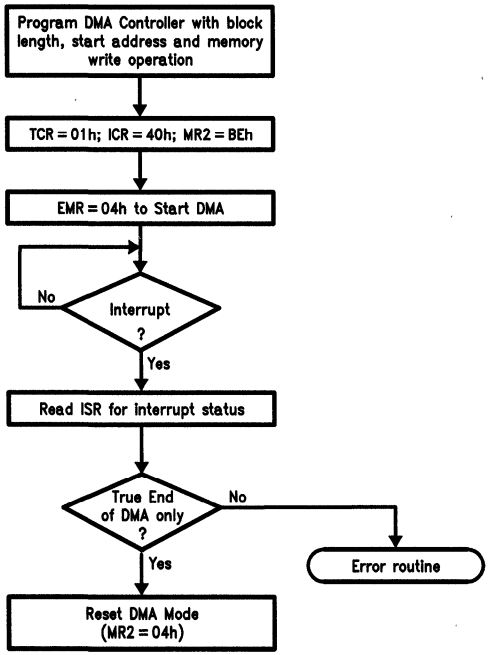
Command Transfer (Target)



TL/F/9387–30

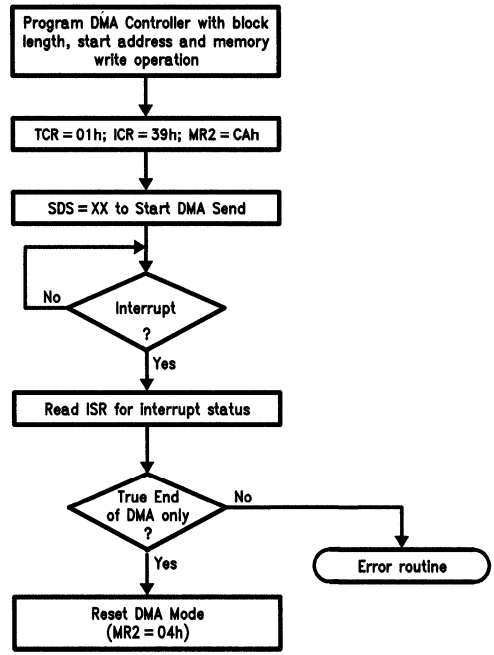
Appendix A1 (Continued)

Block Mode DMA Transfer
Initiator Receive (MODE E)



TL/F/9387–31

Block Mode DMA Transfer
Target Send (MODE E)



TL/F/9387–32

Appendix A2

Register Chart

Read

Current SCSI Data (CSD)

Bit 7							Bit 0
$\overline{DB7}$	$\overline{DB6}$	$\overline{DB5}$	$\overline{DB4}$	$\overline{DB3}$	$\overline{DB2}$	$\overline{DB1}$	$\overline{DB0}$

Initiator Command Register (ICR)

Bit 7							Bit 0
$\overline{RST}$	AIP	LA	$\overline{ACK}$	$\overline{BSY}$	$\overline{SEL}$	$\overline{ATN}$	DBUS

Mode Register 2 (MR2)

Bit 7							Bit 0
BLK	TARG	PCHK	PINT	EOP	BSY	DMA	ARB

Target Command Register (TCR)

Bit 7							Bit 0
0	0	0	0	$\overline{REQ}$	$\overline{MSG}$	$\overline{C/D}$	$\overline{I/O}$

Current SCSI Bus Status (CSB)

Bit 7							Bit 0
$\overline{RST}$	$\overline{BSY}$	$\overline{REQ}$	$\overline{MSG}$	$\overline{C/D}$	$\overline{I/O}$	$\overline{SEL}$	$\overline{DBP}$

Bus and Status Register (BSR)

Bit 7							Bit 0
EDMA	DRQ	SPER	INT	PHSM	BSY	$\overline{ATN}$	$\overline{ACK}$

Input Data Register (IDR)

Bit 7							Bit 0
$\overline{DB7}$	$\overline{DB6}$	$\overline{DB5}$	$\overline{DB4}$	$\overline{DB3}$	$\overline{DB2}$	$\overline{DB1}$	$\overline{DB0}$

Reset Parity/Interrupt (RPI)—MODE N

Bit 7							Bit 0
X	X	X	X	X	X	X	X

Enhanced Mode Register (EMR)

Bit 7							Bit 0
APHS	MPEN	MPOL	SPOL	LOOP	EFN1	EFN0	ARB

Interrupt Status Register (ISR)

Bit 7							Bit 0
SPE	MPE	EDMA	DPHS	APHS	BSY	SEL	ARB

Target Command Register (TCR)—MODE E

Bit 7							Bit 0
(true) EDMA	0	0	0	$\overline{REQ}$	$\overline{MSG}$	$\overline{C/D}$	$\overline{I/O}$

X = Unknown

Write

Output Data Register (ODR)

Bit 7							Bit 0
$\overline{DB7}$	$\overline{DB6}$	$\overline{DB5}$	$\overline{DB4}$	$\overline{DB3}$	$\overline{DB2}$	$\overline{DB1}$	$\overline{DB0}$

Initiator Command Register (ICR)

Bit 7							Bit 0
$\overline{RST}$	MODE E	DIFF EN	$\overline{ACK}$	$\overline{BSY}$	$\overline{SEL}$	$\overline{ATN}$	DBUS

Mode Register 2 (MR2)

Bit 7							Bit 0
BLK	TARG	PCHK	PINT	EOP	BSY	DMA	ARB

Target Command Register (TCR)

Bit 7							Bit 0
X	X	X	X	$\overline{REQ}$	$\overline{MSG}$	$\overline{C/D}$	$\overline{I/O}$

Select Enable Register (SER)

Bit 7							Bit 0
$\overline{DB7}$	$\overline{DB6}$	$\overline{DB5}$	$\overline{DB4}$	$\overline{DB3}$	$\overline{DB2}$	$\overline{DB1}$	$\overline{DB0}$

Start DMA Send (SDS)

Bit 7							Bit 0
X	X	X	X	X	X	X	X

Start DMA Target Receive (SDT)

Bit 7							Bit 0
X	X	X	X	X	X	X	X

Start DMA Initiator Receive (SDI)—MODE N

Bit 7							Bit 0
X	X	X	X	X	X	X	X

Enhanced Mode Register (EMR)

Bit 7							Bit 0
APHS	MPEN	MPOL	SPOL	LOOP	EFN1	EFN0	ARB

Interrupt Mask Register (IMR)

Bit 7							Bit 0
SPE	MPE	EDMA	DPHS	APHS	BSY	SEL	ARB

Target Command Register (TCR)—MODE E

Bit 7							Bit 0
X	X	X	X	$\overline{REQ}$	$\overline{MSG}$	$\overline{C/D}$	$\overline{I/O}$

X = Don't Care

National Semiconductor

DP5380 Asynchronous SCSI Interface (ASI)

General Description

The DP5380 ASI is a CMOS device designed to provide a low cost, high performance Small Computer Systems Interface. It complies with the ANS X3.131-1986 SCSI standard as defined by the ANSI X3T9.2 committee. It can act as both INITIATOR and TARGET, making it suitable for any application. The ASI supports selection, reselection, arbitration and all other bus phases. High-current open-drain drivers on chip reduce application chip count by interfacing direct to the SCSI bus. An on-chip oscillator provides all timing delays.

The DP5380 is pin and program compatible with the NMOS NCR5380 device. NCR5380 or AM5380 applications can use it with no changes to hardware or software. The DP5380 is available in a 40-pin DIP or a 44-pin PCC.

The ASI is intended to be used in a microprocessor based application, and achieves maximum performance with a DMA controller. The device is controlled by reading and writing several internal registers. A standard non-multiplexed address and data bus easily fits any μP environment.

Data transfers can be performed by programmed-I/O, pseudo-DMA or via a DMA controller. The ASI easily interfaces to a DMA controller using normal or Block Mode. The ASI can be used in either a polled or interrupt-driven environment.

Features

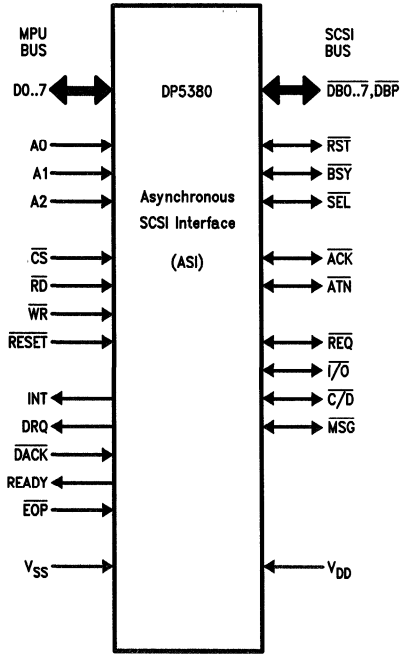
SCSI Interface

- Supports TARGET and INITIATOR roles
- Parity generation with optional checking
- Arbitration support
- Direct control/monitoring of all SCSI signals
- High current outputs drive SCSI bus directly
- Faster and improved timing
- Very low SCSI bus loading

μP Interface

- Memory or I/O-mapped control transfers
- Programmed-I/O or DMA data transfers
- Normal or Block-mode DMA
- Fast DMA handshake timing

Connection Diagram



TL/F/9756–1

Table of Contents

1.0 Functional Description

1.1 OVERVIEW

The ASI is designed to be used as a peripheral device in a μP-based application and appears as a number of read/write registers. Write registers are programmed to select desired functions. Status registers provide indication of operating conditions.

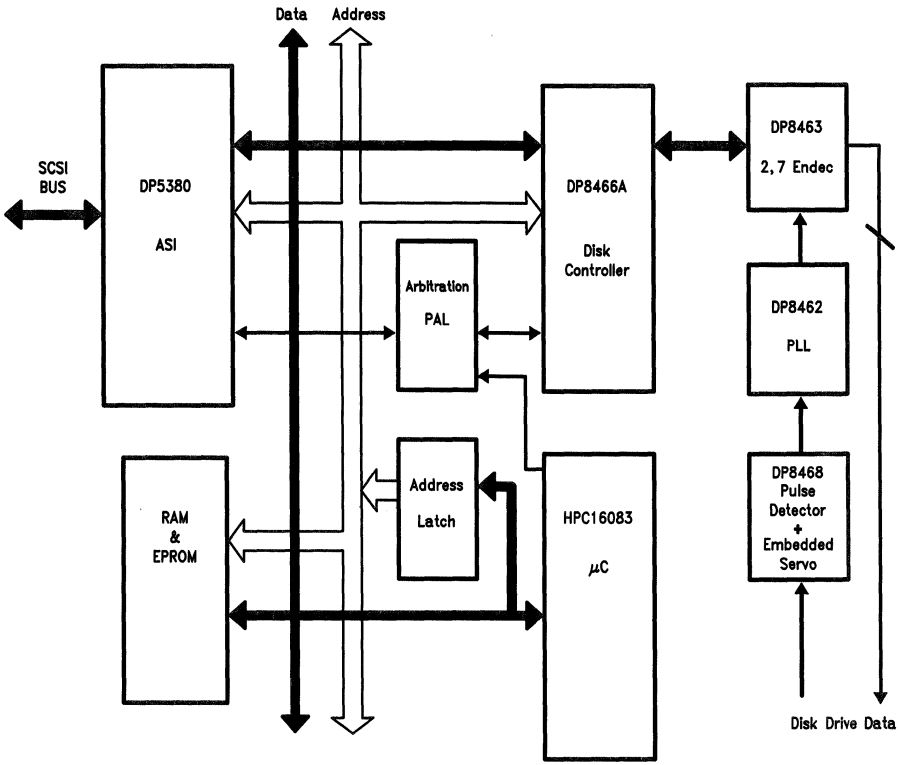
For best performance a DMA controller can be easily interfaced directly to the ASI. The ASI provides request/acknowledge and wait-state signals for the DMA interface.

The SCSI bus is easily controlled via the ASI registers. Any bus signal may be asserted or deasserted via a bit in the appropriate register, and the state of every signal is available by reading registers. This direct control over SCSI signals allows the user to implement all or part of the protocol in firmware. The ASI provides hardware support for much of the protocol.

The ASI provides the following SCSI support:

- Programmed-I/O transfers for all eight information transfer types, with or without parity.
- Data transfers via DMA, in either block or non-block mode. The DMA interface supports most devices.
- Individual setting/resetting and monitoring of every SCSI bus signal.
- Automatic release of the bus for BSY loss from a TARGET, SCSI RST, and lost arbitration.
- Automatic bus arbitration—the μP has only to check for highest priority.
- Selection or Reselection of any bus device. The ASI will respond to both Selection and Reselection.
- Optional automatic monitoring of the $\overline{\text{BSY}}$ signal from a TARGET with an interrupt after releasing control of the bus.

Figure 1 shows an ASI in a typical application, a low cost embedded SCSI disk controller. In this application the 8051 single-chip μP acts as the controller and the dual DMA channels in the DP8475 allow one for the disk data and the other for SCSI data. The PAL® provides chip selection as well as determining who has control of the bus. The advantage of using a μP with on-board ROM is that there is more free time on the external bus.



TL/F/9756–2

FIGURE 1. ASI Application

1.0 Functional Description (Continued)

1.2 μP INTERFACE

Figure 2 shows a block diagram of the ASI. Key blocks within the ASI are Read/Write registers with associated decode and control logic, interrupt and DMA logic, SCSI bus arbitration logic, SCSI drivers/receivers with parity and the SCSI data input and output registers. The ASI has three interfaces, one to SCSI, one to a DMA controller and the third to a μP. The internal registers control all operations of the ASI.

The μP interface consists of non-multiplexed address and data busses with associated control signals. Address decode logic selects a register for reading or writing. The address lines A0–2 select the register for μP accesses while for DMA accesses the address lines are ignored.

The register bank consists of twelve registers mapped into an address space of eight locations. Upon an external chip reset the registers are cleared (all zeroes).

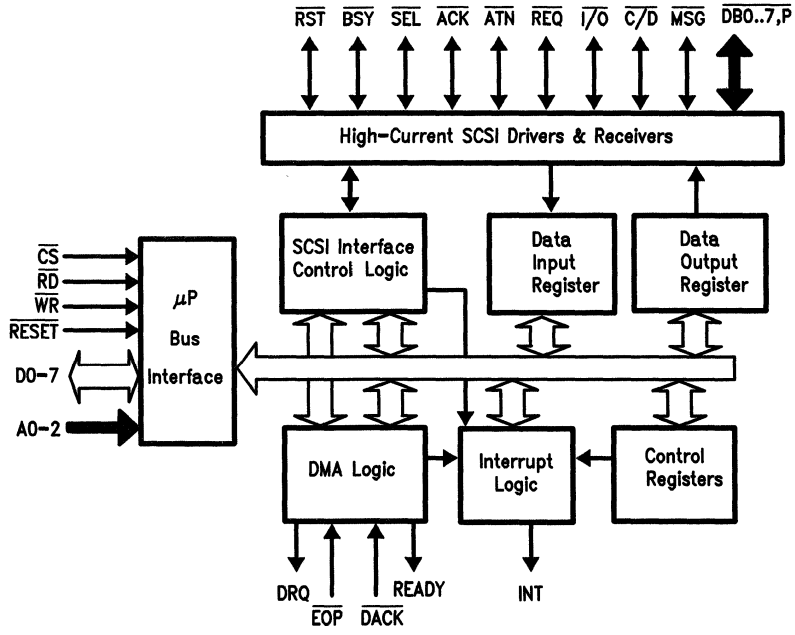
1.3 DMA INTERFACE

The DMA logic interfaces to single-cycle, block mode, flow-through or fly-by controllers. Single byte transfers are accomplished via the DRQ/$\overline{\text{DACK}}$ handshake signals. Block mode transfers use the READY output to control the speed (insert wait-states). An End Of Process ($\overline{\text{EOP}}$) input from the DMA controller signals the ASI to halt DMA transfers. An interrupt can be generated for DMA completion or an error (see Section 5.0). All DMA data passes through the SCSI data input and output registers, automatically selected during DMA cycles.

1.4 SCSI INTERFACE

The ASI contains all logic required to interface directly to the SCSI bus. Direct control and monitoring of all SCSI signals is provided. The state of each SCSI signal may be determined by reading a register which continuously reflects the state of the bus. Each signal may be asserted by writing a ONE to the appropriate bit.

The ASI includes logic to automatically handle SCSI timing sequences too fast for μP control. In particular there is hardware support for DMA transfers, bus arbitration, selection/reselection, bus phase monitoring, $\overline{\text{BSY}}$ monitoring for bus disconnection, bus reset and parity generation and checking.



TL/F/9756–3

FIGURE 2. ASI Block Diagram

1.0 Functional Description (Continued)

The ASI arbitration logic controls arbitration for use of the SCSI bus. The μP programs the SCSI device ID into the ASI, then sets the ARBITRATE bit. The INITIATOR COMMAND REGISTER (ICR) is read to determine when arbitration has started and whether it is won or lost.

The $\overline{BSY}$ signal is continously monitored to detect bus disconnection and bus free phases. The ASI incorporates an on-board oscillator to determine Bus Settle, Bus Free and Arbitration Delays. The oscillator tolerance guarantees all timing to be within the SCSI specification.

The ASI incorporates high-current drivers and SCHMITT trigger receivers for interfacing directly to the SCSI bus. This feature reduces the chip count of any SCSI application.

1.5 PARITY

The ASI provides for parity protection on the SCSI interface. The data bus has eight data bits and one parity bit. The parity may be enabled via a register bit. A parity error can be programmed to cause an interrupt.

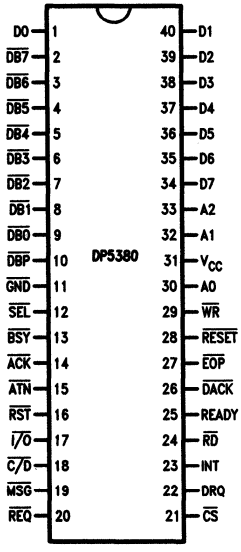
2.0 Pin Descriptions

Symbol	DIP	PCC	Type	Function
$\overline{CS}$	21	24	I	**Chip Select:** an active low enable for read or write operations, accessing the register selected by A0 . . . 2.
A0 . . . 2	30, 32, 33	33, 36, 37	I	**Address 0 . . . 2:** these three signals are used with $\overline{CS}$, $\overline{RD}$, and $\overline{WR}$ to address a register for read or write.
$\overline{RD}$	24	27	I	**Read:** an active low enable for reading an internal register selected by A0 . . . 2 and enabled by $\overline{CS}$. It also selects the Input Data Register when used with $\overline{DACK}$.
$\overline{WR}$	29	32	I	**Write:** an active low enable for writing an internal register selected by A0 . . . 2 and enabled by $\overline{CS}$. It also selects the Output Data Register when used with $\overline{DACK}$.
$\overline{RESET}$	28	31	I	**Reset:** an active low input with a Schmitt trigger. Clears all internal registers. (SCSI $\overline{RST}$ unaffected).
D0 . . . 7	1, 40–34	2, 44–38	I/O	**Data 0 . . . 7:** bidirectional TRI-STATE® signals connecting the active high μP data bus to the internal registers.
INT	23	26	O	**Interrupt:** an active high output to the μP when an error has occurred, an event requires service or has completed.
$\overline{DRQ}$	22	25	O	**DMA Request:** an active high output asserted when the data register is ready to read or written. DRQ occurs only if DMA mode is enabled. The signal is cleared by $\overline{DACK}$.
$\overline{DACK}$	26	29	I	**DMA Acknowledge:** an active low input that resets DRQ and addresses the data registers for input or output transfers. $\overline{DACK}$ is used instead of $\overline{CS}$ by the DMA controller.
READY	25	28	O	**Ready:** an active high output used to control the speed of block mode DMA transfers. Ready goes active when the chip is ready to send/receive data and remains inactive after the transfer until the byte is sent or until the DMA mode bit is reset.
$\overline{EOP}$	27	30	I	**End Of Process:** an active low signal that terminates a block of DMA transfers. It should be asserted during the transfer of the last byte.
$\overline{DB0 . . . 7}$ $\overline{DBP}$	9 . . . 2, 10	10 . . . 3, 11	I/O	**DB0 . . . 7, DBP:** SCSI data bus with parity. $\overline{DB7}$ is the MSB and is the highest priority during arbitration. Parity is ODD. Parity is always generated and can be optionally checked. Parity is not valid during arbitration.
$\overline{RST}$	16	18	I/O	**Reset:** SCSI reset, monitored and can be set by ASI.
$\overline{BSY}$	13	15	I/O	**Busy:** indicates the SCSI bus is being used. Can be driven by TARGET or INITIATOR.
$\overline{SEL}$	12	14	I/O	**Select:** used by the INITIATOR to select a TARGET or by the TARGET to reselect an INITIATOR.
$\overline{ACK}$	14	16	I/O	**Acknowledge:** driven by the INITIATOR and received by the TARGET as part of the REQ/$\overline{ACK}$ handshake.
$\overline{ATN}$	15	17	I/O	**Attention:** driven by the INITIATOR to indicate an attention condition to the TARGET.

2.0 Pin Descriptions (Continued)

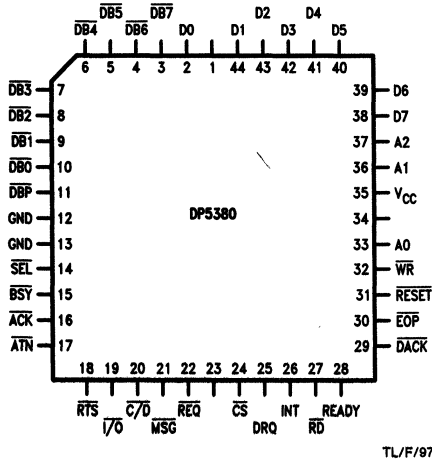
Symbol	DIP	PCC	Type	Function
$\overline{REQ}$	20	22	I/O	**Request:** driven by the TARGET and received by the INITIATOR as part of the $\overline{REQ}$/$\overline{ACK}$ handshake.
$\overline{I/O}$	17	19	I/O	**Input/Output:** driven by the TARGET to control the direction of transfers on the SCSI bus. This signal also distinguishes between selection and reselection.
$\overline{C/D}$	18	20	I/O	**Command/Data:** driven by the TARGET to indicate whether command or data bytes are being transferred.
$\overline{MSG}$	19	21	I/O	**Message:** driven by the TARGET during message phase to identify message bytes on the bus.
VCC GND	31 11	35 12, 13	—	**VCC, GND:** +5V DC is required. Because of very large switching currents good decoupling and power distribution is mandatory.

2.1 Connection Diagrams



TL/F/9756–4

Order Number DP5380N
See NS Package Number N40A



TL/F/9756–5

Order Number DP5380V
See NS Package Number V44A

3.0 Register Description

3.1 GENERAL

The DP5380 ASI is a register-based device with eight addressable locations. Some addresses have dual functions depending upon whether they are being read from or written to. Device operation is described in Section 4.

Figure 3.2 summarises the register map. Note that for registers reading or writing SCSI signals the SCSI name is used for each bit. Although the SCSI bus is active low the registers invert the SCSI bus. This means an active SCSI signal is represented by a ONE in a register and an inactive signal by a ZERO.

3.2 REGISTERS

OUTPUT DATA REGISTER (ODR)

8 Bits HA 0 Write-Only

This is a transparent latch used to send data to the SCSI bus. The register can be written by µP cycles or via DMA. DMA writes automatically select the ODR at Hex Address 0 (HA 0). This register is also written with the ID bits required during arbitration and selection/reselection phases. Data is latched at the end of the write cycle.

Bit 7							Bit 0
$\overline{DB7}$	$\overline{DB6}$	$\overline{DB5}$	$\overline{DB4}$	$\overline{DB3}$	$\overline{DB2}$	$\overline{DB1}$	$\overline{DB0}$

Output Data Register

CURRENT SCSI DATA (CSD)

8 Bits HA 0 Read-Only

This register enables reading of the current SCSI data bus. If SCSI parity checking is enabled it will be checked at the beginning of the read cycle. The register is also used for µP accesses of SCSI data during programmed-I/O or ID checking during arbitration. Parity is not valid during arbitration. DMA transfers select the IDR (HA 6) instead of the CSD register.

Bit 7							Bit 0
$\overline{DB7}$	$\overline{DB6}$	$\overline{DB5}$	$\overline{DB4}$	$\overline{DB3}$	$\overline{DB2}$	$\overline{DB1}$	$\overline{DB0}$

Current SCSI Data

INITIATOR COMMAND REGISTER (ICR)

8 Bits HA 1 Read-Write

This register is used to control the INITIATOR and some other SCSI signals, and to monitor the progress of bus arbitration. Most of the SCSI signals may also be asserted in TARGET mode. Bits 5 to 0 are reset when $\overline{BSY}$ is lost (see MR2 description).

Bit 7							Bit 0
$\overline{RST}$	TEST	LA/DIFF	$\overline{ACK}$	$\overline{BSY}$	$\overline{SEL}$	$\overline{ATN}$	DBUS

Initiator Command Register

DBUS: Assert Data Bus **Bit 0**

0 Disable SCSI data bus driving.

1 Enable contents of Output Data Register onto the SCSI data bus. SCSI parity is also generated and driven on DBP.

This bit should be set when transferring data out of the ASI in either TARGET or INITIATOR mode, for both DMA or programmed-I/O. In INITIATOR mode the drivers are only enabled if: Mode Register 2 TARGET MODE bit is 0, and $\overline{I/O}$ is false, and $\overline{C/D}$, $\overline{I/O}$, $\overline{MSG}$ match the contents of the Target Command Register (phasematch is true). In TARGET mode only the MR2 bit needs to be set with this bit.

Reading the ICR reflects the state of this bit.

$\overline{ATN}$: Assert Attention **Bit 1**

0 Deassert $\overline{ATN}$.

1 Assert SCSI $\overline{ATN}$ signal. The MR2 TARGET MODE bit must also be false to assert the signal.

Reading the ICR reflects the state of this bit.

$\overline{SEL}$: Assert Select **Bit 2**

0 Deassert $\overline{SEL}$.

1 Assert SCSI $\overline{SEL}$ signal. Can be used in INITIATOR or TARGET mode.

Reading the ICR reflects the state of this bit.

$\overline{BSY}$: Assert Busy **Bit 3**

0 Deassert $\overline{BSY}$.

1 Assert SCSI $\overline{BSY}$ signal. Can be used in INITIATOR or TARGET mode.

Reading the ICR reflects the state of this bit.

Hex Adr	Register	Mnemonic	Bits	R/W
0	Output Data Register	ODR	8	WO
0	Current SCSI Data	CSD	8	RO
1	Initator Command Register	ICR	8	RW
2	Mode Register 2	MR2	8	RW
3	Target Command Register	TCR	4	RW
4	Select Enable Register	SER	8	WO
4	Current SCSI Bus Status	CSB	8	RO
5	Bus and Status	BSR	8	RO
5	Start DMA Send	SDS	0	WO
6	Start DMA Target Receive	SDT	0	WO
6	Input Data Register	IDR	8	RO
7	Start DMA Initiator Receive	SDI	0	WO
7	Reset Parity/Interrupts	RPI	0	RO

FIGURE 3.2. Registers

3.0 Register Description (Continued)

$\overline{ACK}$: Assert Acknowledge **Bit 4**

0 Deassert $\overline{ACK}$.

1 Assert SCSI $\overline{ACK}$ signal. The MR2 TARGET MODE bit must also be false to assert the signal.

Reading the ICR reflects the state of this bit.

DIFF: Differential Enable **Bit 5 Write**

0 This bit must be reset to 0.

1 Do not use. Reserved for future use on a differential pair device.

LA: Lost Arbitration **Bit 5 Read**

0 Normally reset to 0 to show arbitration not lost or not enabled.

1 Will be set when the ASI loses arbitation, i.e. when SEL is true during arbitration AND the Assert $\overline{SEL}$ bit of this register is false.

A 1 in this bit means the ASI has arbitrated for the bus, asserted $\overline{BSY}$ and its ID on the data bus and another device has asserted $\overline{SEL}$. The ARBITRATE bit in MR2 must be set to enable arbitration.

TEST: Test Mode Enable **Bit 6 Write**

0 Output drivers are enabled.

1 Output drivers disabled.

AIP: Arbitration In Progress **Bit 6 Read**

0 Normally 0 to show no arbitration in progress.

1 Set when the ASI has detected BUS FREE phase and asserted $\overline{BSY}$ and the Output Data Register contents onto the SCSI data bus. This bit remains set until arbitration is disabled.

$\overline{RST}$: Assert $\overline{RST}$ **Bit 7**

0 Deassert $\overline{RST}$.

1 Assert SCSI $\overline{RST}$ signal. $\overline{RST}$ is asserted as long as this bit is 1, or until a μP Reset ($\overline{RESET}$).

After this bit is set the INT pin goes active and internal registers reset (except for the interrupt latch, MR2 TARGET MODE bit, and this bit. Reading the ICR reflects the state of this bit.

MODE REGISTER 2 (MR2)
8 Bits HA2 Read-Write

This register is used to program basic operating conditions in the ASI. Operation as TARGET or INITIATOR, DMA mode and type as well as some interrupt controls are set via this register. This is a Read/Write register and when read the value reflects the state of each bit.

Bit 7 **Bit 0**

BLK	TARG	PCHK	PINT	EOP	BSY	DMA	ARB

Mode Register 2

ARB: Arbitrate **Bit 0**

0 Disable arbitration.

1 Enable arbitration. The ASI will wait for a BUS FREE phase then arbitrate for the bus. Before setting this bit the Output Data Register should contain the SCSI device ID—a single bit set only. The status of the arbitration process is given in the AIP and LA bits (6, 5) in the Initiator Command Register.

DMA: DMA Mode **Bit 1**

0 Disable DMA mode.

1 Enable DMA operation. This bit should be set then one of address 5 to 7 written to start DMA. The TARGET MODE bit in the ICR and the phase lines in the TCR should have been set appropriately. The DBUS bit in the ICR must be set for DMA operations. $\overline{BSY}$ must be active in order to set this bit. The phase lines must match the contents of the TCR during the actual transfers. In DMA mode ASI logic automatically controls the $\overline{REQ}$/$\overline{ACK}$ handshakes.

This bit should be reset by a μP write to stop any DMA transfer. An $\overline{EOP}$ signal will not reset this bit. During DMA, $\overline{CS}$ and $\overline{DACK}$ should not be active simultaneously.

This bit will be reset if $\overline{BSY}$ is lost during DMA mode.

BSY: Monitor Busy **Bit 2**

0 Disable $\overline{BSY}$ monitor.

1 Monitor SCSI $\overline{BSY}$ signal and interrupt when $\overline{BSY}$ goes inactive. When this bit goes active the lower 6 bits of the ICR are reset and all signals removed from the SCSI bus. This is used to check for valid TARGET connection.

EOP: Enable $\overline{EOP}$ Interrupt **Bit 3**

0 No interrupt for $\overline{EOP}$.

1 Interrupt after valid $\overline{EOP}$ condition.

PINT: Enable SCSI Parity Interrupt **Bit 4**

0 No interrupt on SCSI parity error.

1 When SCSI parity is enabled via the PCHK bit, setting this bit enables an interrupt upon a SCSI parity error.

PCHK: Enable SCSI Parity Checking **Bit 5**

0 No SCSI parity checking.

1 Enable checking of SCSI parity during read operations. This applies to either programmed I/O or DMA mode.

TARG: Target Mode **Bit 6**

0 Initiator Mode.

1 Target Mode.

BLK: Block Mode DMA **Bit 7**

0 Non-block DMA.

1 When set along with DMA bit (1) enable block mode DMA transfers. In block mode the READY line is used to handshake each byte with the DMA controller instead of the DRQ/$\overline{DACK}$ handshake used in non-block mode.

TARGET COMMAND REGISTER (TCR)
4 Bits HA 3 Read-Write

This register is used to control TARGET SCSI signals and to program the desired phase during INITIATOR mode. During

3.0 Register Description (Continued)

DMA transfers the SCSI phase lines (C/D, MSG, I/O) must match the contents of the TCR for transfers to occur. A phase mismatch halts DMA transfers and generates an interrupt.

Bit 7							Bit 0
x	x	x	x	REQ	MSG	C/D	I/O

Target Command Register

This is a read/write register and the value read reflects the state of each bit, except bit 4–7 which always read 0.

I/O: Assert I/O **Bit 0**

0 Deassert I/O.

1 Assert SCSI I/O signal. The MR2 TARGET MODE bit must also be active.

C/D: Assert C/D **Bit 1**

0 Deassert C/D.

1 Assert SCSI C/D signal. The MR2 TARGET MODE bit must also be active.

MSG: Assert MSG **Bit 2**

0 Deassert MSG.

1 Assert SCSI MSG signal. The MR2 TARGET MODE bit must also be active.

REQ: Assert REQ **Bit 3**

0 Deassert REQ.

1 Assert SCSI REQ signal. The MR2 TARGET MODE bit must also be active. This bit is used to handshake SCSI data via programmed-I/O.

SELECT ENABLE REGISTER (SER)
8 Bits HA 4 Write-Only

This write-only register is used to program the SCSI device ID for the ASI to respond to during Selection or Reselection Phases. Only one bit in the register should be set. When SEL is true, BSY false and the SER ID bit active an interrupt will occur.

This interrupt is reset or can be disabled by writing zero to this register. Parity will also be checked during Selection or Reselection if the PCHK bit in MR2 is set.

Bit 7							Bit 0
DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0

Select Enable Register

CURRENT SCSI BUS STATUS (CSB)
8 Bits HA 4 Read-Only

This read-only register is used to monitor SCSI control signals and the SCSI parity bit. The SCSI lines are monitored during programmed-I/O transfers and after an interrupt in order to determine the cause. A bit is 1 if the corresponding SCSI signal is active.

Bit 7							Bit 0
RST	BSY	REQ	MSG	C/D	I/O	SEL	DBP

Current SCSI Bus Status

BUS AND STATUS REGISTER (BSR)
8 Bits HA 5 Read-Only

This read-only register is used to monitor SCSI signals not included in the CSB, and internal status bits. This register is read after an interrupt to determine the cause of an interrupt. Bit 0 or 1 are set to 1 if the SCSI signal is active.

Bit 7							Bit 0
EDMA	DRQ	SPER	INT	PHSM	BSY	ATN	ACK

Bus & Status Register

ACK: Acknowledge **Bit 0**

This bit reflects the state of the SCSI ACK Signal.

ATN: Attention **Bit 1**

This bit reflects the state of the SCSI ATN Signal.

BSY: Busy Error **Bit 2**

0 No Error.

1 This SCSI BSY signal has become inactive while the MR2 BSY (Monitor BSY) bit is set. This will cause an interrupt, remove all ASI signals from the SCSI bus and reset the DMA MODE bit in MR2.

PHSM: Phase Match **Bit 3**

0 Phase Match. The SCSI C/D, I/O and MSG phase lines are continuously compared with the corresponding bits in the TCR. The result of this comparison is reflected in this bit. This bit must be 1 (phase matches) for DMA transfers. A phase mismatch will stop DMA transfers and cause an interrupt.

INT: Interrupt Request **Bit 4**

0 No Interrupt.

1 Interrupt request active. Set when an enabled interrupt condition occurs. This bit reflects the state of the INT pin. INT may be reset by performing a Reset Parity/Interrupt (RPI) function.

SPER: SCSI Parity Error **Bit 5**

0 No SCSI parity error.

1 SCSI parity error occurred. This bit remains set once an error occurs until the RPI function clears it. The PCHK bit in MR2 must be set for a parity error to be checked and registered.

DRQ: DMA Request **Bit 6**

0 No DMA request.

1 DMA request active. This bit reflects the state of the DRQ pin. DRQ is reset by asserting DACK during a DMA cycle or by resetting the DMA bit in MR2. A Busy error will reset the MR2 DMA bit and thus will also clear DRQ. A phase mismatch will not reset DRQ.

EDMA: End of DMA **Bit 7**

0 Not end of DMA.

1 Set when DACK, EOP and either RD or WR are active simultaneously. Normally occurs when the last byte is transferred by the DMA. During DMA send operations the last byte transferred by the DMA may not have been transferred on SCSI so REQ and ACK should be monitored to verify when the last SCSI transfer is complete. This bit is reset when the MR2 DMA bit is reset.

3.0 Register Description (Continued)

START DMA SEND (SDS)
0 Bits HA 5 Write-Only

This write-only register is used to start a DMA send operation. A write of don't-care data should be the last thing done by the μP. The MR2 DMA, BLK and TARG bits must have been programmed previously.

Bit 7							Bit 0
x	x	x	x	x	x	x	x

Start DMA Send

START DMA TARGET RECEIVE (SDT)
0 Bits HA 6 Write-Only

This write-only register is used to start a DMA Target Receive operation. Same comments as SDS apply.

INPUT DATA REGISTER (IDR)
8 Bits HA 6 Read-Only

This read-only register contains the SCSI data last latched during a DMA receive. Each byte from SCSI is latched into this register automatically by the ASI DMA logic. A DMA read ($\overline{DACK}$ and $\overline{RD}$) automatically selects this register. Programmed-I/O SCSI data reads should use the CSD (HA8)

START DMA INITIATOR RECEIVE (SDI)
0 Bits HA 7 Write-Only

This write-only register is used to start a DMA INITIATOR Receive Operation. Same comments as SDS apply.

RESET PARITY/INTERRUPT (RPI)
0 Bits HA 7 Read-Only

This read-only register is used to reset the parity and interrupt latches. Reading this register resets the SCSI parity, Busy Loss and Interrupt Request latches.

4.0 Device Operation

4.1 GENERAL

This section describes overall operation of the ASI. More detailed information of data transfers, interrupts and reset conditions are covered in later sections. The operation description covers μP accesses, SCSI bus monitoring, arbitration, selection, reselection, programmed-I/O, DMA interrupts. Programming and timing details are covered.

For information regarding interfacing to μP's and DMA controllers refer to Section 7.0.

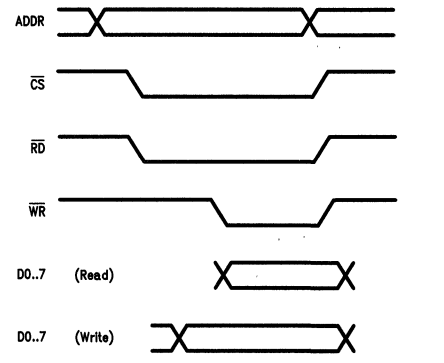
In the descriptions following program examples are given in pseudo-C. This processor-independent approach should be clearest. These are backed up by flow charts in Appendix A.1.

4.2 μP ACCESSES

The μP accesses the EASI via the $\overline{CS}$, $\overline{RD}$, $\overline{WR}$ and address and data lines in order to read/write the registers. *Figure 4.2* shows typical timing. Note the use of non-multiplexed address and data lines.

4.3 SCSI BUS MONITORING/DRIVING

The SCSI bus may be monitored or driven at any time. Each bus signal is buffered and inverted by the ASI and can be read via the CSB, BSR and CSD registers. An active SCSI reads a 1 in the status registers.



TL/F/9756–6

FIGURE 4.2. μP Cycles

Each SCSI signal may be asserted by setting a bit in the TCR or ICR. Setting the bit to 1 asserts the SCSI signal.

The following code demonstrates a byte transferred via programmed-I/O in INITIATOR mode.

```
{     /*Transfer one byte as Initiator*/
   while (NOT (TCR:REQ));
      /* wait till TARGET asserts REQ */
   data = input (CSD);
      /* parity is checked if enabled*/
   output (ICR, Assert ACK);
   while (TCR:REQ);
      /* wait till TARGET deasserts REQ */
   output (ICR, 0);
      /* deassert ACK, ready for next byte */
}
```

4.4 ARBITRATION

This sub-section describes the arbitration support provided by the ASI and how to program it.

Since the SCSI arbitration process requires signal sequencing too fast for μP's, hardware support is provided by the ASI. The arbitration process is enabled by bit 0 MR2 (ARB). Prior to setting this bit the ODR should be programmed with the device's SCSI ID—a single bit.

The ASI will monitor the bus for a BUS FREE phase. The $\overline{BSY}$ signal is continuously monitored. If continuously inactive for at least a SCSI Bus Settle Delay (400 ns) and $\overline{SEL}$ is inactive, a valid Bus Free Phase exists. After a period of SCSI Bus Free Delay (800 ns) the ASI asserts $\overline{BSY}$ and the ODR onto the SCSI data bus. The μP should poll the ICR to determine when arbitration has started. The AIP bit in the ICR is set when the Bus Free Phase is detected and the EASI is beginning the Bus Free Delay. Following the Bus Free Delay a 2.2 μs SCSI Arbitration Delay is required before examining the data bus to resolve the priorities of the ID bits. This delay must be implemented in firmware. The ICR Lost Arbitration (LA) bit must be examined to determine whether arbitration is lost. The LA bit is set if another

4.0 Device Operation (Continued)

device asserts $\overline{SEL}$ during arbitration. If the LA bit is 0 the data bus is read via the CSD register. The data is examined to resolve ID priorities. If this device is the highest ID assert $\overline{SEL}$ by setting ICR bit 2 to a 1. After waiting Bus Clear + Bus Settle Delays (1200 ns) the Selection Phase begins. These 2 delays must be implemented in firmware.

4.5 SELECTION/RESELECTION

The ASI can be used to select or reselect a device. The ASI will also respond to selection or reselection.

4.5.1 Selecting/Reselecting

Selection requires programming the ODR with the desired and own device ID's; the data bus via ICR DBUS (bit 0); asserting $\overline{ATN}$ if required via ICR bit 1; asserting $\overline{SEL}$ via ICR bit 2; then resetting the MR2 ARB bit.

The SER should have been cleared to zero before Selection/Reselection to ensure the ASI does not respond. If Reselection is desired the $\overline{I/O}$ line should also be asserted before $\overline{SEL}$ via TCR bit 0.

Resetting the ARB bit causes the ASI to remove $\overline{BSY}$ and the ODR from the data bus. Thus the ICR Assert data bus bit is required to assert the bits for desired and own device ID's.

$\overline{BSY}$ is then monitored to determine when the device has responded to (re)selection. If the device fails to respond an error handler should sequence the ASI off the bus. If the device responds the ICR DBUS and $\overline{SEL}$ bits should be reset to remove these signals. If this is a Reselection the ICR $\overline{BSY}$ bit (3) should be set before removing the other signals.

The bus is now ready to handle Information Transfer Phases.

4.5.2 (Re)Selection Response

The ASI responds to Selection or Reselection when the SER is non-zero. A (re)selected interrupt is generated when $\overline{BSY}$ is false for at least a Bus Settle Delay (400 ns); and $\overline{SEL}$ is true AND any non-zero bit in the SER has its corresponding SCSI data bus bit active. A Selection is disabled by zeroing the SER. If parity is supported it should be valid during (re)selection so must be checked via the SPE bit (5) in the BSR. SCSI specification states that (re)selection is not valid if more than 2 data bits are active. This condition is checked by reading the CSD.

When the selection interrupt occurs it is determined by reading the BSR and CSB registers. There is no dedicated status bit for (re)selection so it must be determined by the absence of other interrupts, and the active state of the $\overline{SEL}$ signal. Reselection occurs when $\overline{I/O}$ is also active. See Section 6.0.

4.6 MONITORING BSY

While an INITIATOR is connected to a TARGET the TARGET must maintain an active $\overline{BSY}$ signal. During DMA operations the $\overline{BSY}$ signal is monitored by the ASI and will halt operations if it goes inactive. To enable $\overline{BSY}$ to be monitored at other times the MR2 BSY bit (2) should be set. An interrupt will be generated if $\overline{BSY}$ goes inactive while MR2 BSY is set.

This interrupt sets bit 2 in the BSR.

4.7 COMMAND/MESSAGE/STATUS TRANSFERS

Command message and status bytes are transferred using programmed-I/O. The SCSI $\overline{REQ}$/$\overline{ACK}$ handshake is accomplished by monitoring and setting lines individually. Data is output via the ODR and read in via the CSD register.

The following code shows INITIATOR and TARGET programming for two of these cases. See Appendix A.1 for flowcharts.

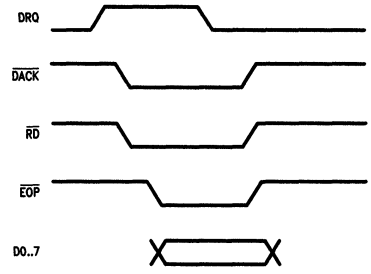
```
Initiator Command Send
{
   MR2 = monitor BSY
   TCR = Command Phase  /*02h*/
   while (bytes) to do) {
      while (REQ) inactive)
        idle; /*CSB bit 5 = 0*/
      if (BSR: phase match == 0)
        phase error;
      else {
        ODR = date byte;
        ICR = Assert ACK;
        while (REQ active)
          idle; /*CSB bit 5 == 1*/
        ICR = deassert ACK
        /* byte transfer complete */
        byte count --;
      }
   }
   goto data phase;
}

Target Message Receive
{
   /* assumed Assert BSY already set in ICR */
   MR2 = TARG MODE OR PARITY CHECK OR
         PARITY INTERRUPT;
   TCR = Message Out phase; /*06h*/
   delay (Bus Settle);
   TCR = Assert REQ;
   while (ACK inactive)
         idle; /* BSR bit 0 */
   data = CSD; /* parity is latched */
   if (BSR: parity error)
         error routine;
   else  {
         TCR = deassert REQ;
         while (ACK active)
            idle;
   }
    /* message done, can change to next
phase */
}
```

4.8 NON-BLOCK DMA TRANSFERS

Data transfers may be effected by DMA. This method should be used for optimum performance. Two methods of DMA are available-block and non-block mode. This section describes non-block mode transfers.

4.0 Device Operation (Continued)

The interface to the DMA controller uses the DRQ, $\overline{DACK}$, EOP lines in non-block mode. Each byte is requested ($\overline{DRQ}$) and ack'd ($\overline{DACK}$). Representative timing for a DMA read is shown in *Figure 4.8.1.*



TL/F/9756–7

FIGURE 4.8.1. Non-Block DMA Timing

4.8.1. NON-BLOCK DMA

DMA operation involves programming the ASI with the set-up parameters, initiating the DMA cycles and checking for correct operation when the completion interrupt is received. The DMA controller should be programmed with the data byte count and the memory start address. Methods of halting a DMA operation are covered in Section 4.11.

Setting up the ASI requires enabling or disabling the following: Data bus driving, DMA mode enable, $\overline{BSY}$ monitoring, $\overline{EOP}$ interrupt, parity checking, parity interrupt, TARGET Mode, bus phase.

Once set up DMA should be initiated by writing to address 5, 6, or 7 as appropriate. The DMA controller should assert $\overline{EOP}$ during the transfer of the last byte, although this may be done by the μP if the DMA transfers (n − 1) bytes and the μP transfers the last byte. See the application guide for more details (Section 7.0).

Upon completion the μP should check the following as required: End of DMA, Parity Error, Phase Match, Busy Error. The end of DMA occurs as a response to $\overline{EOP}$. SCSI transfers may still be underway so $\overline{REQ}$ and $\overline{ACK}$ must still be checked to establish when the final byte is finished.

The code below shows programming of the ASI in each of the four DMA cases. One of these cases is shown in a flow diagram in Appendix A.

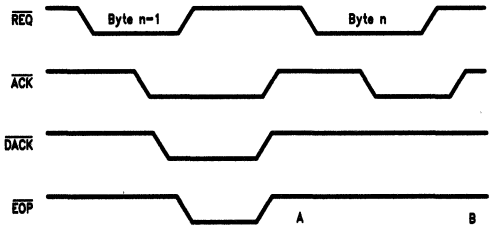
```
Initiator Send              /*DATA OUT PHASE*/
{
    Program DMA Controller;
    TCR = 00h;              /*phase*/
    ICR = 01h;              /*Assert_DBUS*/
    MR2 = 0Eh;
    SDS = 00;               /*Start DMA Send*/
    while (NOT interrupt)
      idle;
    while (CSD:REQ)
      idle                  /*wait for last
                              SCSI byte
                            transfer so phase
                              is checked*/
    if(BSR:Busy error OR NOT
        (BSR:End_of_DMA))
      error routine;
    else {                  /*DMA End*/
      MR2 = 04h;            /*reset DMA bit*/
      ICR = 0;
    }
}

Initiator Receive           /*DATA IN PHASE*/
{
    Program DMA Controller;
    TCR = 01h;              /*phase*/
    MR2 = 3Eh;
    SDI = 0;                /*Start DMA Init
                              Rx*/
    while (NOT interrupt)
        idle;
/*no need to wait for last SCSI handshake
  done since DMA done implies it is
  checked*/
    if(BSR:parity_error OR BSR: busy_error
     or NOT (BSR End of DMA)
       do error routines;
    else {                  /*End of DMA*/
     while (CSD:REQ)
      idle; /*wait for REQ inactive
          to deassert ACK*/
     MR2 = 04h;
    }
}

Target Receive              /*DATA OUT PHASE*/
{
    Program DMA Controller;
    TCR = 0;                /*phase*/
    ICR = 08h;
    MR2 = 7Ah;              /*check parity*/
    SDT = 0;                /*Start DMA Targ Rx*/
    while (not interrupt)
        idle;
    /*when End of DMA occurs the last byte
     has been read and checked*/
    if(BSR:parity_error OR NOT(BSR: End_of_DMA)
      error routine;
       {      else          /*End of DMA*/
      while (BSR:ACK)
            idle;
    /*Not True End of DMA, so wait until SCSI
     bus inactive before changing phase*/
    MR2 = 40h;
    change phase as required;
    }
}
```

4.0 Device Operation (Continued)

```
Target Send                /*DATA IN PHASE*/
{
    Program DMA Controller;
    TCR = 01h;             /*phase*/
    ICR = 09h;
    MR2 = 4Ah;
    SDS = 0;               /*Start DMA Send*/
    while (NOT interrupt)
       idle;
    if(NOT(BSR:End_of_DMA)
       error;
    else {   /*DMA end*/
     repeat {
      while (CSB:REQ OR BSR:ACK)
      loop count = 3;
      loop count --;} /*decrement*/
     until (loop count == 0);
     MR2 = 40h;
     Change phase as required;
     }
}
```

Some explanation of the final part of Target Send is required. In this type of DMA operation it is very difficult to exactly determine the True End of DMA simply detecting $\overline{REQ}$ and $\overline{ACK}$ simultaneously inactive is not enough.

Reference to *Figure 4.8.2* will help to understand the following text.



TL/F/9756–8

FIGURE 4.8.2. Target Send DMA

As shown in *Figure 4.8.2* $\overline{ACK}$ going active causes the DRQ for the next byte and also $\overline{REQ}$ to go inactive. $\overline{ACK}$ going inactive allows $\overline{REQ}$ to go active for the next byte. If the INITIATOR is slow removing $\overline{ACK}$ the μP may sample the SCSI bus after the $\overline{EOP}$ interrupt at point A. Here both $\overline{REQ}$ and $\overline{ACK}$ will be inactive, but there is one more byte to transfer on SCSI. Due to chip timing delays this condition will not last more than 200 ns. A safe way to determine the True End of DMA is to sample $\overline{REQ}$ and $\overline{ACK}$ and ONLY when both are inactive in three successive samples will the μP be at point B in the figure.

4.9 BLOCK MODE DMA TRANSFERS

In Block Mode the DMA interface uses the DRQ, $\overline{DACK}$, $\overline{EOP}$ and READY lines, DRQ is asserted once at the beginning of transfers and deasserted once $\overline{DACK}$ is received. $\overline{DACK}$ should be asserted continuously for the duration of all the transfer. $\overline{EOP}$ should be asserted during the last DMA byte signal when the next DMA byte transfers. The ASI asserts the READY signal when the next DMA byte should be transferred.

As for non-block mode the End of DMA interrupt is just $\overline{EOP}$, also in block mode receive the ASI does not return READY to an active signal after $\overline{EOP}$. This means external logic must gate off READY if the μP is not to be locked up. For more details see Section 7.0.

The block mode is intended for systems where the overhead of handing the system busses to and from the μP and DMA controller is too great. The block mode handshake is not necessarily faster than non-block (it may be) but the overall transfer rate is improved once the bus exchange overhead is removed. Of course the μP is prevented from executing for the whole DMA operation.

If a phase mismatch occurs the READY signal is left in the inactive state. The DMA controller must hand back the bus to the μP and the inactive READY signal may need to be gated off.

When performing DMA as an INITIATOR the $\overline{EOP}$ signal does not deassert $\overline{ACK}$ on the SCSI bus. Firmware must determine when $\overline{REQ}$ is inactive after the last SCSI transfer then reset the MR2 DMA bit to deassert $\overline{ACK}$.

Programming the ASI in block mode is the same as non-block mode except bit 7 in MR2 should also be set.

4.10 PSEUDO DMA

The system design can utilize ASI DMA logic for non-data transfers. This removes the need to poll $\overline{REQ}$/$\overline{ACK}$ and program the assertion/deassertion of the handshake signal. The μP can emulate a DMA controller by asserting $\overline{DACK}$ and $\overline{EOP}$ signals. DRQ may be sampled by reading the BSR. In most cases the chip decode logic can be adapted to this use for little or no cost. See Section 7.0 for further details.

4.11 HALTING A DMA OPERATION

There are three ways to halt a DMA operation apart from a chip or SCSI reset. These methods are: $\overline{EOP}$, phase mismatch and resetting the DMA MODE bit in MR2.

4.11.1 End Of Process

$\overline{EOP}$ is asserted for a minimum period during the last DMA cycle. The $\overline{EOP}$ signal generates the End of DMA interrupt. $\overline{EOP}$ does not cause the MR2 DMA mode bit to be reset.

4.11.2 DMA Phase Mismatch

If a $\overline{REQ}$ goes active while there is a phase mismatch the DMA will be halted and an interrupt generated. The ASI will stop driving the SCSI bus when the mismatch occurs. A phase mismatch is when the TCR phase bits do not match the SCSI bus values.

4.11.3 DMA Mode Bit

If $\overline{EOP}$ is not used the best method is to reset the MR2 DMA Mode bit. This bit may be reset at any time, and should be reset after an End of DMA interrupt or a phase mismatch.

4.0 Device Operation (Continued)

Resetting the bit disables all DMA logic and thus should only be reset at the True End of DMA condition. Additionally all DMA logic is reset so this bit must be reset then set again to carry out the next DMA phase.

5.0 Interrupts

5.1 OVERVIEW

Before individually describing each interrupt an explanation of the use of interrupts is required.

5.2 USING INTERRUPTS

Interrupts are controlled by bits in MR2 if control is provided. Not all interrupts can be disabled under software control. When an interrupt occurs both the BSR and CSD register must be read and analysed to determine the source of interrupt. Since status is NOT provided for each interrupt great care should be exercised when determining the interrupt source.

5.3 SCSI PARITY ERROR

If SCSI parity checking is enabled via MR2 bit 5 an interrupt can occur as a result of a read from CSD, a selection/(re)selection, or a DMA receive operation. The parity error bit (bit 5) in the BSR will be set if checking is enabled. An interrupt will occur if Enable Parity Interrupt (bit 4) of MR2 is set. The interrupt is reset by reading HA7. Following an interrupt the BSR and CSD should contain the values shown below.

Bit 7							Bit 0
x	x	1	1	x	x	x	x
EDMA	DRQ	SPER	INT	PHSM	BSY	$\overline{ATN}$	$\overline{ACK}$

BSR

Bit 7							Bit 0
0	1	x	x	x	x	0	x
$\overline{RST}$	$\overline{BSY}$	$\overline{REQ}$	$\overline{MSG}$	C/$\overline{D}$	I/$\overline{O}$	$\overline{SEL}$	$\overline{DBP}$

CSD

5.4 END OF DMA

If $\overline{EOP}$ is asserted during a DMA transfer bit 7 of the BSR will be set and an interrupt generated if bit 3 of MR2 is 1. $\overline{EOP}$ is recognized when $\overline{EOP}$, $\overline{DACK}$ and either $\overline{IOR}$ or $\overline{IOW}$ are all simultaneously active for a minimum period. The interrupt may be reset by reading HA 7. Following an interrupt the BSR and CSD should contain the values shown below.

Bit 7							Bit 0
1	x	x	1	x	x	0	x
EDMA	DRQ	SPER	INT	PHSM	BSY	$\overline{ATN}$	$\overline{ACK}$

BSR

Bit 7							Bit 0
0	1	x	x	x	x	0	x
$\overline{RST}$	$\overline{BSY}$	$\overline{REQ}$	$\overline{MSG}$	C/$\overline{D}$	I/$\overline{O}$	$\overline{SEL}$	$\overline{DBP}$

CSD

5.5 DMA PHASE MISMATCH

When the SCSI $\overline{REQ}$ goes active during a DMA operation the contents of the TCR are compared with the SCSI phase lines C/$\overline{D}$, $\overline{MSG}$ and I/$\overline{O}$. If the two do not match an interrupt is generated. This interrupt will occur as long as the MR2 DMA bit is set (bit 1), i.e. it cannot be masked. The mismatch removes the ASI from driving the SCSI data bus. The interrupt may reset by reading HA 7. Following an interrupt the BSR and CSD should contain the values shown below.

Bit 7							Bit 0
x	0	x	1	0	x	x	x
EDMA	DRQ	SPER	INT	PHSM	BSY	$\overline{ATN}$	$\overline{ACK}$

BSR

Bit 7							Bit 0
0	x	x	x	x	x	0	x
$\overline{RST}$	$\overline{BSY}$	$\overline{REQ}$	$\overline{MSG}$	C/$\overline{D}$	I/$\overline{O}$	$\overline{SEL}$	$\overline{DBP}$

CSD

5.6 BUSY LOSS

If bit 2 MR2 is set the SCSI $\overline{BSY}$ signal is monitored and an interrupt is generated if $\overline{BSY}$ is continuously inactive for at least a BUS SETTLE DELAY (400 ns). This interrupt may be reset by reading HA 7. Following an interrupt the BSR and CSD should contain the values shown below, where usually CSD = 00.

Bit 7							Bit 0
x	x	x	1	x	1	0	x
EDMA	DRQ	SPER	INT	PHSM	BSY	$\overline{ATN}$	$\overline{ACK}$

BSR

Bit 7							Bit 0
0	0	x	x	x	x	x	x
$\overline{RST}$	$\overline{BSY}$	$\overline{REQ}$	$\overline{MSG}$	C/$\overline{D}$	I/$\overline{O}$	$\overline{SEL}$	$\overline{DBP}$

CSD

5.7 (RE)SELECTION

An interrupt will be generated when: $\overline{SEL}$ is active, $\overline{BSY}$ is inactive, and the device ID is true. The device ID is determined by the value in the SER. If ANY non-zero bit in the SER has its corresponding SCSI data bit active during selection the device ID is true. If I/$\overline{O}$ is active this is a reselection. The interrupt is disabled by writing all zeros to the SER, and reset by reading HA 7.

5.0 Interrupts (Continued)

If SCSI parity checking is enabled it will be checked and should be valid. Following an interrupt the BSR and CSD should contain the values shown below.

Bit 7							Bit 0
0	0	0	1	x	0	x	0
EDMA	DRQ	SPER	INT	PHSM	BSY	$\overline{ATN}$	$\overline{ACK}$

BSR

Bit 7							Bit 0
0	0	0	0	0	0	1	x
$\overline{RST}$	$\overline{BSY}$	$\overline{REQ}$	$\overline{MSG}$	$\overline{C/D}$	$\overline{I/O}$	$\overline{SEL}$	$\overline{DBP}$

CSD

6.0 Reset Conditions

6.1 GENERAL

There are three ways to reset the ASI; µP chip $\overline{RESET}$, SCSI bus reset applied externally, SCSI bus reset issued by the ASI.

6.2 CHIP RESET

When the $\overline{RESET}$ signal is asserted for the required duration the ASI clears ALL internal registers and therefore resets all logic. This action does not create an interrupt or generate a SCSI reset.

6.3 EXTERNAL SCSI RESET

When a SCSI $\overline{RST}$ is applied externally the ASI resets all registers and logic and issues an interrupt. The only register bits not affected are the Assert RST bit (bit 7) in the ICR and the TARGET Mode bit (bit 6) in MR2.

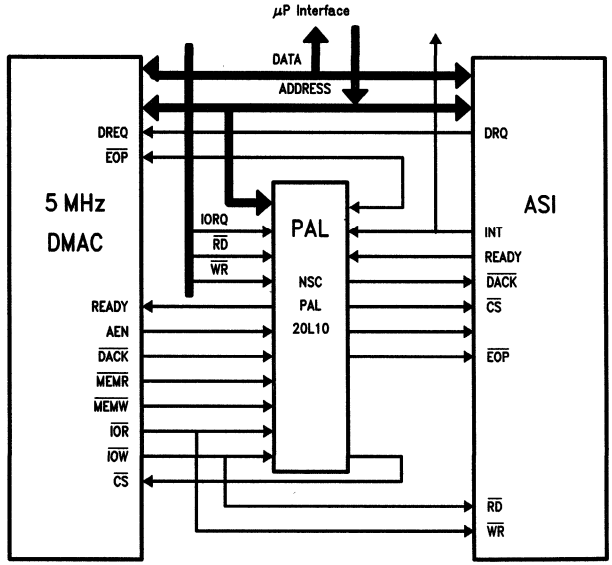
6.4 SCSI RESET ISSUED

When the µP sets the Assert RST bit in the ICR the $\overline{RST}$ signal goes active. Since the ASI monitors $\overline{RST}$ also the same reset actions as in 6.3 apply. The SCSI $\overline{RST}$ signal will remain active as long as bit 7 in the ICR is set—i.e. until programmed 0 or a chip $\overline{RESET}$ occurs.

7.0 Application Guide

This section is intended to show the interface between the µP, ASI and DMA controller (DMAC). *Figure 7.1* shows a general interface when the ASI and DMAC are I/O-mapped devices. This configuration will implement a 2 to 2.5M Bytes/sec SCSI port using 2 cycle compressed timing from the 5 MHz DMAC.

Using a faster DMAC and memory may allow the ASI to operate at a higher rate—but of course any system will be limited by the available DMA rate from the SCSI device currently connected to. The interface shown has several features that are examined more closely in the following text.



TL/F/9756–9

FIGURE 7.1. µP/ASI/DMA Interface

7.0 Application Guide (Continued)

All the interface signal requirements are satisfied by a PAL device. The memory interface is not shown, only the relevant DMAC and μP lines are included.

The ASI data and address lines connect directly to the μP/DMAC busses. The DRQ output from the ASI goes direct to the DMAC. The $\overline{EOP}$ output from the DMAC goes to the ASI input, but can also be asserted via the PAL since the DMAC output is open-drain.

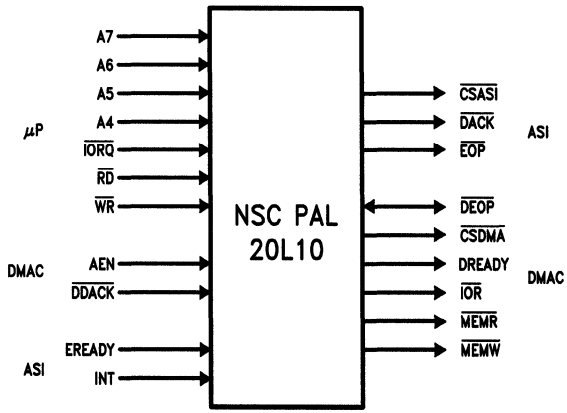
The PAL is programmed so that the μP can access the ASI in three ways. The three access types are: Register R/W, DMA R/W, DMA with $\overline{EOP}$. Examination of the PAL equations below shows how the μP may perform any of the three basic access types simply by accessing the ASI at different I/O address slots. This enables the μP to simulate a DMAC (pseudo-DMA). DMA mode may then be used for all information transfer phases.

In DMA mode the ASI generates all SCSI handshakes. At all other times the μP is responsible for $\overline{REQ}/\overline{ACK}$ handshakes. Using pseudo-DMA may reduce μP overhead.

When doing DMA transfers via BLOCK MODE and an error occurs, the ASI may not deassert the READY signal. For some DMA controllers this may lock the bus, so the PAL asserts READY and $\overline{EOP}$ to the DMA if an interrupt occurs while READY is false. This completes the current DMA cycle and prevents further DMA for the rest of the block thus allowing the bus to be handed back to the μP for servicing.

The PAL generates $\overline{RD}$ and $\overline{WR}$ strobes while the μP is bus master, but the DMAC provides the strobes while it is bus master so the PAL outputs are TRI-STATE.

The PAL details are shown in *Figure 7.2* with the signal definitions and equations following.



TL/F/9756–10

FIGURE 7.2. Interface PAL

7.0 Application Guide (Continued)

```
CSASI = IORQ*A7*A6*A5*A4*AEN; ASI reg R/W chip select
ADACK = IORQ*A7*A6*A5*A4*RD;  µP pseudo-DMA cycle
        IORQ*A7*A6*A5*A4*WR
        +IORQ*A7*A6*A5*A4*RD; µP pseudo-DMA with EOP
        +IORQ*A7*A6*A5*A4*WR
        +DDACK;               ; DMAC DMA cycle
IF(AEN)AEOP = IORQ*A7*A6*A5*A4*RD; µP pseudo-DMA with EOP
            +IORQ*A7*A6*A5*A4*WR + DEOP* AREADY
IF(DDACK*AREADY*INT)DEOP = DDACK*AREADY*INT
                         ;DMA cycle with error
CSDMA = IORQ*A7*A6*A5*A4     ;DMAC register R/W
DREADY = AREADY*INT          ;ASI not READY and not INT
       +AREADY*DDACK         ;ASI not READY and DMA cycle active
IF(AEN)IOR  = IORQ*RD        ;µP I/O Read cycle
IF(AEN)IOW  = IORQ*WR        ;µP I/O Write cycle
IF(AEN)MEMR = IORQ*RD        ;µP memory Read cycle
IF(AEN)MEMW = IORQ*WR        ;µP memory Write cycle
```

FIGURE 7.3. PAL Equations

The µP and DMA signals are defined below

A7–A4	Address bus
IORQ	Memory I/O cycle select
RD	Read Strobe
WR	Write Strobe
AEN	High DMA address enable asserted by DMAC
DDACK	DMAC DMA Acknowledge
CSDMA	DMA Chip Select
DREADY	Ready signal to DMAC—inserts wait-states when low
IOR, IOW	I/O data strobes to/from DMAC
MEMR, MEMW	Memory data strobe from DMAC

8.0 Absolute Maximum Ratings*

If Military/Aerospace specified devices are required, please contact the National Semiconductor Sales Office/Distributors for availability and specifications.

Supply Voltage (V_{CC})	−0.5V to +7.0V
DC Input Voltage (V_{IN})	−0.5V to V_{CC} + 0.5V
DC Output Voltage (V_{OUT})	−0.5V to V_{CC} + 0.5V
Storage Temperature Range (T_{STG})	−65°C to +150°C
Power Dissipation (P_D)	500 mW
Lead Temperature (T_L) (Soldering, 10 sec)	260°C
Electro-Static Discharge Rating	2 kV

*Absolute maximum ratings are those values beyond which damage to the device may occur.

9.0 DC Electrical Characteristics (V_{CC} = 5.0V ± 10% unless otherwise specified) T_A = 0°C to +70°C

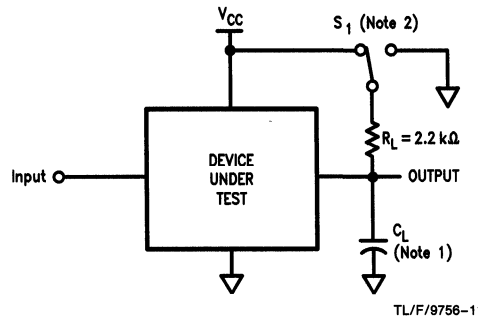
Symbol	Parameter	Conditions	Typ	Limit	Units
V_{IH}	Minimum High Level Input Voltage			2.0	V
V_{IL}	Maximum Low Level Input Voltage			0.8	V
V_{OH1} V_{OH2}	Minimum High Level Output Voltage	$\|I_{OUT}\|$ = 20 μA $\|I_{OUT}\|$ = 4.0 mA		V_{CC} − 0.1 2.4	V V
V_{OL1} V_{OL2} V_{OL3}	Maximum Low Level Output Voltage	SCSI Bus Pins: $\|I_{OL}\|$ = 48 mA Other Pins: $\|I_{OL}\|$ = 20 μA $\|I_{OL}\|$ = 8.0 mA		0.5 0.1 0.4	V V V
I_{IN}	Maximum Input Current	V_{IN} = V_{CC} or GND		±1	μA
I_{OZ}	Maximum TRI-STATE Output Leakage Current	V_{OUT} = V_{CC} or GND		±10	μA
I_{CC}	Supply Current	V_{IN} = V_{CC} or GND SCSI Inputs = 3V	2.5	4	mA

Capacitance T_A = 25°C, f = 1 MHz

Symbol	Parameter (Note 3)	Typ	Units
C_{IN}	Input Capacitance	5	pF
C_{OUT}	Output Capacitance	7	pF

AC Test Conditions

Input Pulse Level	GND to 3.0V
Input Rise and Fall Times	6 ns
Input/Output Reference Levels	1.3V
TRI-STATE Reference Levels (Note 2)	Active Low + 0.5V Active High − 0.5V



TL/F/9756–11

Note 1: C_L = 50 pF including jig and scope capacitance.

Note 2: S1 = Open for push-pull outputs.

S1 = V_{CC} for active low to TRI-STATE.

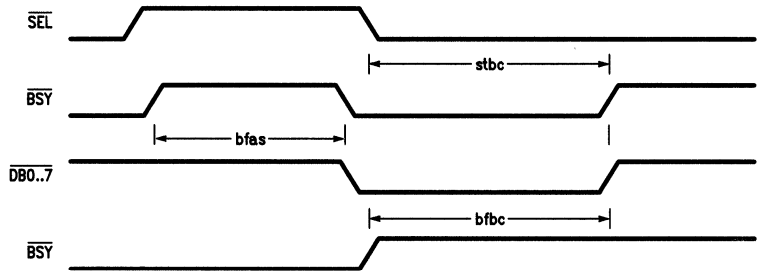
S1 = GND for active high to TRI-STATE.

Note 3: This parameter is not 100% tested.

10.0 AC Electrical Characteristics all parameters are preliminary and subject to change without notice

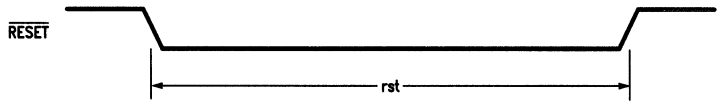
Symbol	Parameter	DP5380			Units
		Min	Typ	Max	
bfas	BSY False to Arbitrate Start	1200		2200	ns
bfbc	BSY False to Bus Clear			800	ns
rst	RESET Pulse Width	150			ns
stbc	SEL True to Bus Clear			500	ns

10.1 ARBITRATION



TL/F/9756–12

10.2 μP RESET



TL/F/9756–13

10.0 AC Electrical Characteristics

all parameters are preliminary and subject to change without notice (Continued)

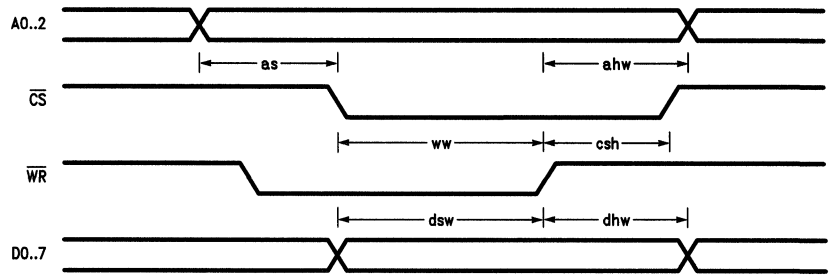
Symbol	Parameter	DP5380			Units
		Min	Typ	Max	
ahr	Address Hold from End of Read Enable (Note 1)	10			ns
ahw	Address Hold from End of Write Enable (Note 2)	10			ns
as	Address Setup to Read or Write Enable (Notes 1, 2)	10			ns
csh	CS Hold from End of RD or WR	0			ns
dhr	Data Hold from End of Read Enable (Notes 1, 3)	10		60	ns
dhw	μP Data Hold Time from End of WR	20			ns
dsw	Data Setup to End of Write Enable	50			ns
rdv	Data Valid from Read Enable (Note 1)			100	ns
ww	Write Enable Width (Note 2)	60			ns

Note 1: Read enable (μP) is CS and RD active.

Note 2: Write enable (μP) is CS and WR active.

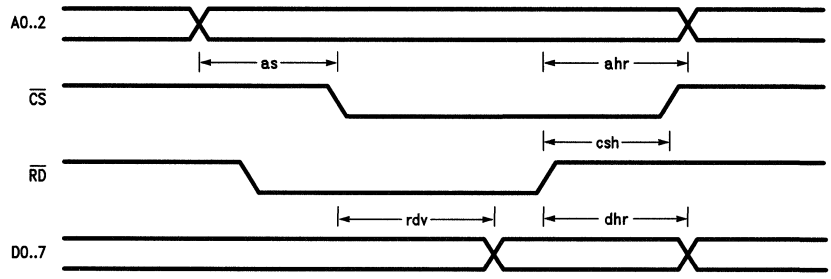
Note 3: This includes the RC delay inherent in the tests' method. These signals typically turn off after 25 ns enabling other devices to drive these lines with no contention.

10.3 μP WRITE



TL/F/9756–14

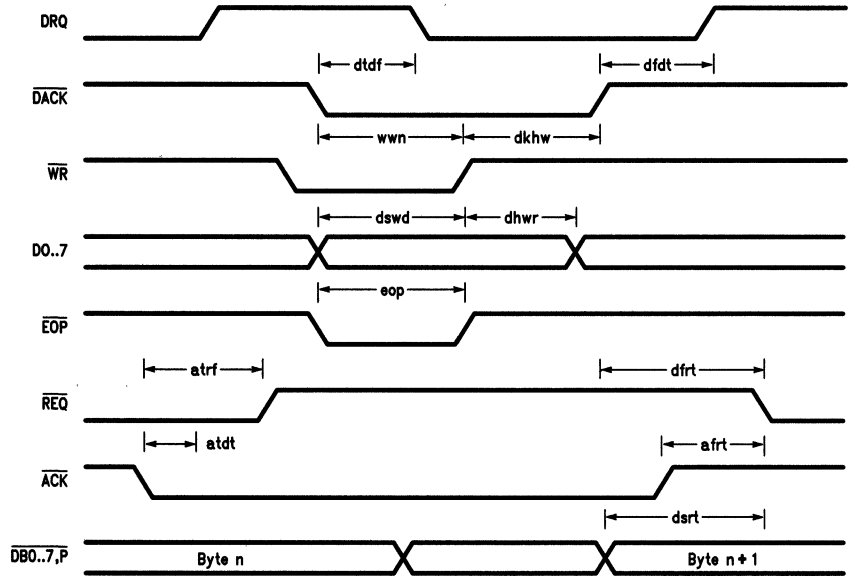
10.4 μP READ



TL/F/9756–15

10.0 AC Electrical Characteristics (Continued)

10.5 DMA WRITE (NON-BLOCK MODE) TARGET SEND



TL/F/9756–16

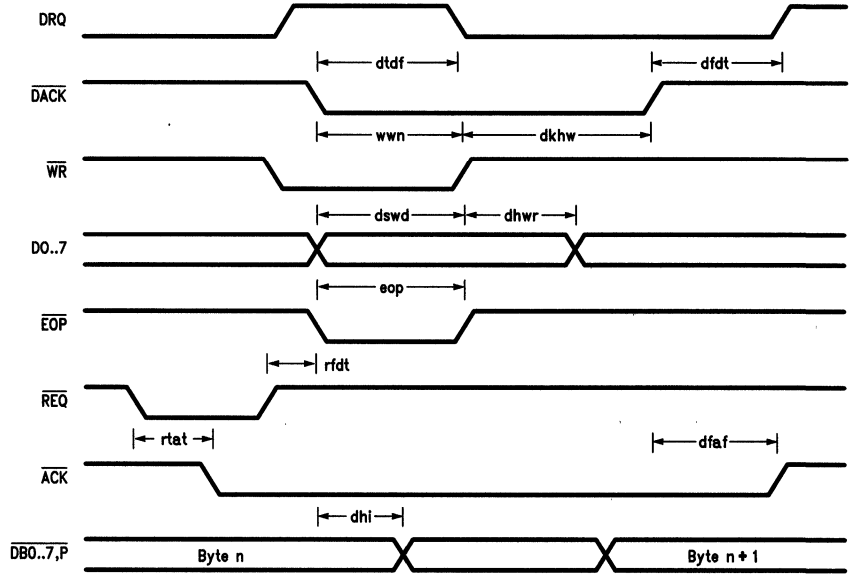
Symbol	Parameter	DP5380			Units
		Min	Typ	Max	
afrt	ACK False to REQ True (DACK or WR False)			120	ns
atdt	ACK True to DRQ True (Target)			90	ns
atrf	ACK True to REQ False (Target)			115	ns
dfdt	DACK False to DRQ True	30	90		ns
dfrt	DACK False to REQ True (ACK False)			110	ns
dhwr	DMA Data Hold Time from End of WR	30			ns
dkhw	DACK Hold from End of WR	0			ns
dsrt	SCSI Data Setup to REQ True (Target Send) (Note 1)	40			ns
dswd	Data Setup to End of DMA Write Enable	50			ns
dtdf	DACK True to DRQ False			100	ns
eop	Width of EOP Pulse (Note 2)	40			ns
wwn	DMA Non-Block Mode Write Enable Width (Note 3)	60			ns

Note 1: EOP, DACK, RD/WR must all be true for recognition of EOP.

Note 2: Write enable (DMA) is DACK and WR active.

10.0 AC Electrical Characteristics (Continued)

10.6 DMA WRITE (NON-BLOCK MODE) INITIATOR SEND



TL/F/9756–17

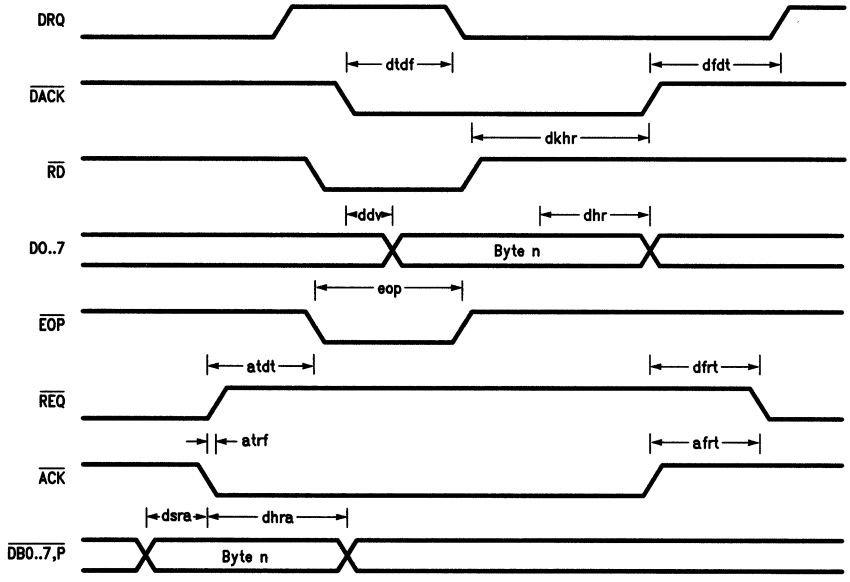
Symbol	Parameter	DP5380			Units
		Min	Typ	Max	
dfaf	DACK False to ACK False (Non-Block Initiator Send)			120	ns
dfdt	DACK False to DRQ True	30	90		ns
dhi	SCSI Data Hold from Write Enable—Initiator	15			ns
dhwr	DMA Data Hold Time from End of WR	30			ns
dkhw	DACK Hold from End of WR	0			ns
dswd	Data Setup to End of DMA Write Enable	50			ns
dtdf	DACK True to DRQ False			100	ns
eop	Width of EOP Pulse (Note 1)	40			ns
rfdt	REQ False to DRQ True			100	ns
rtat	REQ True to ACK True (Initiator Send)			100	ns
wwn	DMA Non-Block Mode Write Enable Width (Note 2)	60			ns

Note 1: EOP, DACK, RD/WR must all be true for recognition of EOP.

Note 2: Write enable (DMA) is DACK and WR active.

10.0 AC Electrical Characteristics (Continued)

10.7 DMA READ (NON-BLOCK MODE) TARGET RECEIVE



TL/F/9756–18

Symbol	Parameter	DP5380			Units
		Min	Typ	Max	
afrt	ACK False to REQ True (DACK or WR False)			120	ns
atdt	ACK True to DRQ True (Target)			90	ns
atrf	ACK True to REQ False (Target)			115	ns
ddv	DMA Data Valid from Read Enable (Note 1)			90	ns
dfdt	DACK False to DRQ True	30	90		ns
dfrt	DACK False to REQ True (ACK False)			110	ns
dhr	Data Hold from End of Read Enable (Notes 1, 2)	10		60	ns
dhra	SCSI Data Hold from REQ or ACK True (Receive)	30			ns
dkhr	DACK Hold from End of RD	0			ns
dsra	SCSI Data Setup Time to REQ or ACK True (Receive)	20			ns
dtdf	DACK True to DRQ False			100	ns
eop	Width of EOP Pulse (Note 4)	40			ns

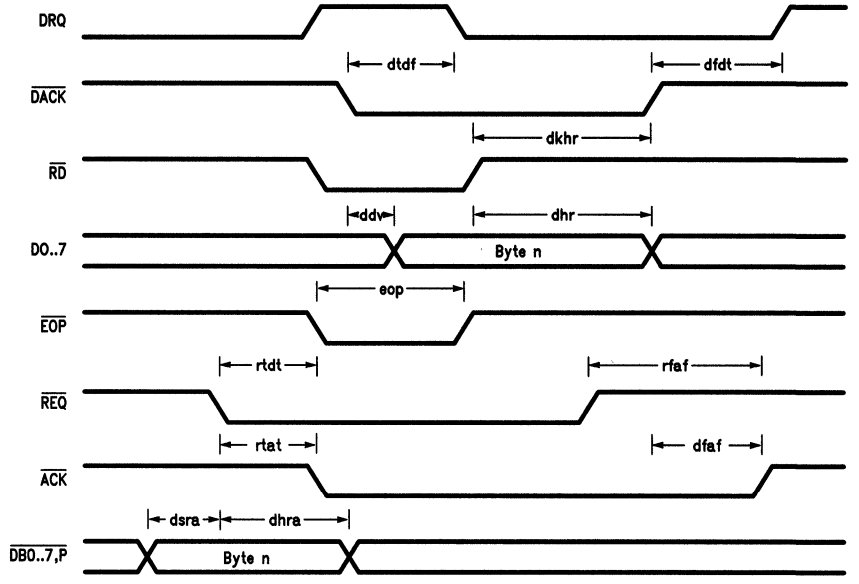
Note 1: Read enable (DMA) is DACK and RD active.

Note 2: This includes the RC delay inherent in the tests' method. These signals typically turn off after 25 ns enabling other devices to drive these lines with no contention.

Note 3: EOP, DACK, RD/WR must all be true for recognition of EOP.

10.0 AC Electrical Characteristics (Continued)

10.8 DMA READ (NON-BLOCK MODE) INITIATOR RECEIVE



TL/F/9756–19

Symbol	Parameter	DP5380			Units
		Min	Typ	Max	
ddv	DMA Data Valid from Read Enable (Note 1)			90	ns
dfaf	DACK False to ACK False (REQ False, Non-block, In rx)			120	ns
dfdt	DACK False to DRQ True	30	90		ns
dhr	Data Hold from End of Read Enable (Notes 1, 2)	10		60	ns
dhra	SCSI Data Hold from REQ or ACK True (Receive)	30			ns
dkhr	DACK Hold from End of RD	0			ns
dsra	SCSI Data Setup Time to REQ or ACK True (Receive)	20			ns
dtdf	DACK True to DRQ False			100	ns
eop	Width of EOP Pulse (Note 3)	40			ns
rfaf	REQ False to ACK False (DACK False)			100	ns
rtat	REQ True to ACK True (Initiator Receive)			100	ns
rtdt	REQ True to DRQ True			120	ns

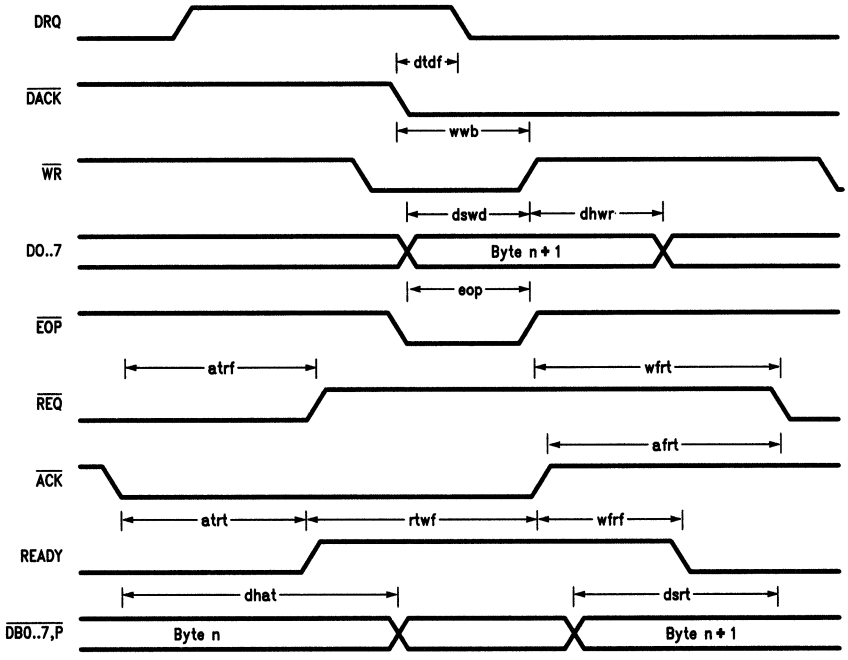
Note 1: Read enable (DMA) is DACK and RD active.

Note 2: This includes the RC delay inherent in the tests' method. These signals typically turn off after 25 ns enabling other devices to drive these lines with no contention.

Note 3: EOP, DACK, RD/WR must all be true for recognition of EOP.

10.0 AC Electrical Characteristics (Continued)

10.9 DMA WRITE (BLOCK MODE) TARGET SEND



TL/F/9756–20

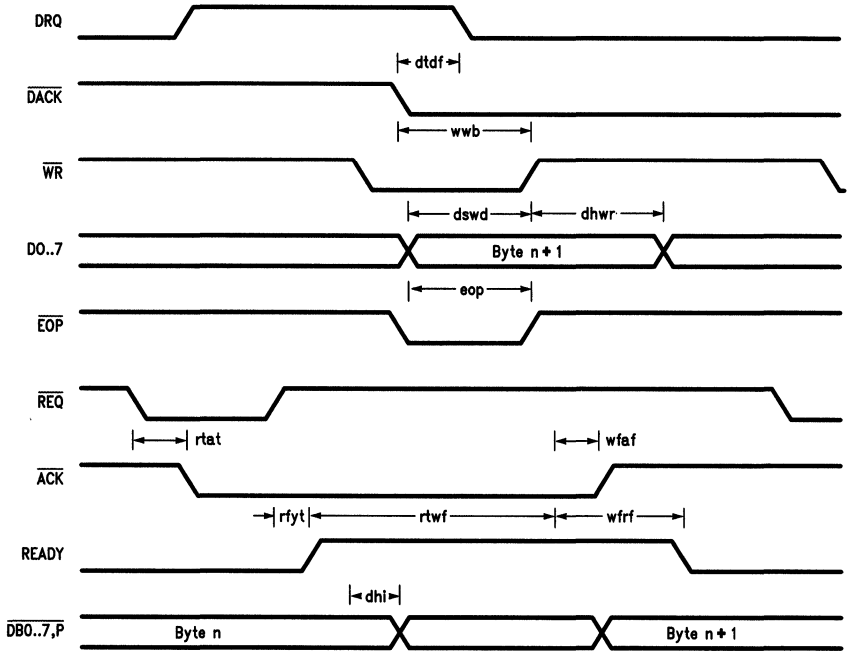
Symbol	Parameter	DP5380			Units
		Min	Typ	Max	
afrt	ACK False to REQ True (DACK or WR False)			120	ns
atrf	ACK True to REQ False (Target)			115	ns
atrt	ACK True to READY True (Block Mode Target Send)			110	ns
dhat	SCSI Data Hold from ACK True	40			ns
dhwr	DMA Data Hold Time from End of WR	30			ns
dsrt	SCSI Data Setup to REQ True	50			ns
dswd	Data Setup to End of DMA Write Enable	50			ns
dtdf	DACK True to DRQ False			100	ns
eop	Width of EOP Pulse (Note 1)	40			ns
rtwf	READY true to WR False	60			ns
wfrf	WR False to READY False			100	ns
wfrt	WR False to REQ True (ACK False)			120	ns
wwb	DMA Write Enable Width (Note 2)	60			ns

Note 1: EOP, DACK, RD, WR must all be true for recognition of EOP.

Note 2: Write enable (DMA) is DACK and WR active.

10.0 AC Electrical Characteristics (Continued)

10.10 DMA WRITE (BLOCK MODE) INITIATOR SEND



TL/F/9756–21

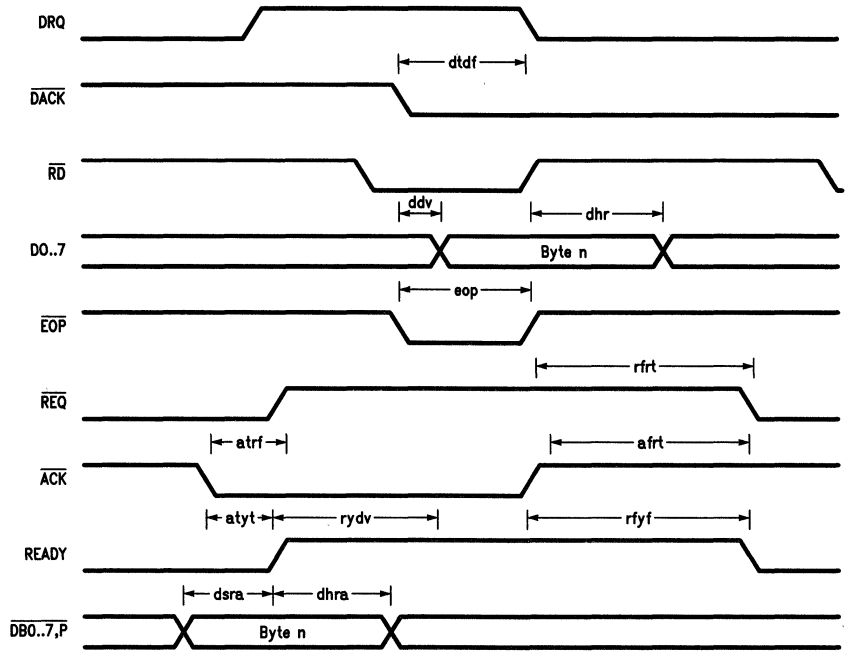
Symbol	Parameter	DP5380			Units
		Min	Typ	Max	
dhi	SCSI Data Hold from Write Enable—Initiator	15			ns
dhwr	DMA Data Hold Time from End of WR	30			ns
dswd	Data Setup to End of DMA Write Enable	50			ns
dtdf	DACK True to DRQ False			100	ns
eop	Width of EOP Pulse (Note 2)	40			ns
rfyt	REQ False to READY True			80	ns
rtat	REQ True to ACK True			100	ns
rtwf	READY True to WR False	60			ns
wfaf	WR False to ACK False (REQ False)			120	ns
wfrf	WR False to READY False			100	ns
wwb	DMA Write Enable Width (Note 1)	60			ns

Note 1: Write enable (DMA) is DACK and WR active.

Note 2: EOP, DACK, RD/WR must all be true for recognition of EOP.

10.0 AC Electrical Characteristics (Continued)

10.11 DMA READ (BLOCK MODE) TARGET RECEIVE



TL/F/9756–22

Symbol	Parameter	DP5380			Units
		Min	Typ	Max	
afrt	ACK False to REQ True (DACK or WR False)			120	ns
atrf	ACK True to REQ False			115	ns
atyt	ACK True to READY True			110	ns
ddv	DMA Data Valid from Read Enable (Note 1)			90	ns
dhr	Data Hold from End of Read Enable (Notes 1, 2)	10		60	ns
dhra	SCSI Data Hold from REQ or ACK True	30			ns
dsra	SCSI Data Setup Time to REQ or ACK True	20			ns
dtdf	DACK True to DRQ False			100	ns
eop	Width of EOP Pulse (Note 3)	40			ns
rfrt	RD False to REQ True (ACK False)			100	ns
rfyf	RD False to READY False			110	ns
rydv	READY True to Data Valid			35	ns

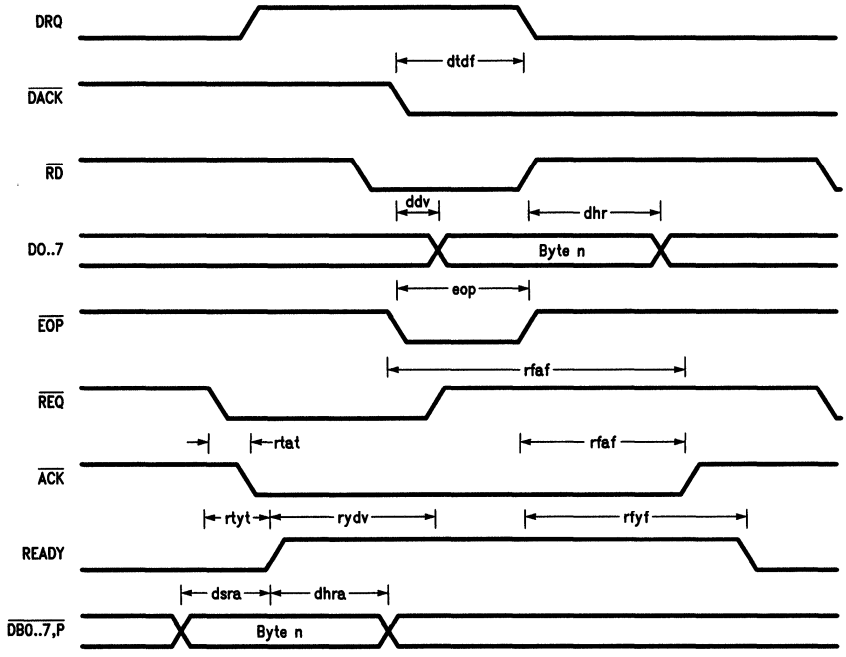
Note 1: Read enable (DMA) is DACK and RD active.

Note 2: This includes the RC delay inherent in the tests' method. These signals typically turn off after 25 ns enabling other devices to drive these lines with no contention.

Note 3: EOP, DACK, RD/WR must all be active for recognition of EOP.

10.0 AC Electrical Characteristics (Continued)

10.12 DMA READ (BLOCK MODE) INITIATOR RECEIVE



TL/F/9756–23

Symbol	Parameter	DP5380			Units
		Min	Typ	Max	
ddv	DMA Data Valid from Read Enable (Note 1)			90	ns
dhr	Data Hold from End of Read Enable (Notes 1, 2)	10		60	ns
dhra	SCSI Data Hold from REQ or ACK True	30			ns
dsra	SCSI Data Setup Time to REQ or ACK True	20			ns
dtdf	DACK True to DRQ False			100	ns
eop	Width of EOP Pulse (Note 3)	40			ns
rdaf	RD False to ACK False (REQ False)			125	ns
rfaf	REQ False to ACK False (DACK False)			100	ns
rfyf	RD False to READY True False			110	ns
rtat2	REQ True to ACK True			100	ns
rtyt	REQ True to READY True			75	ns
rydv	READY True to Data Valid			35	ns

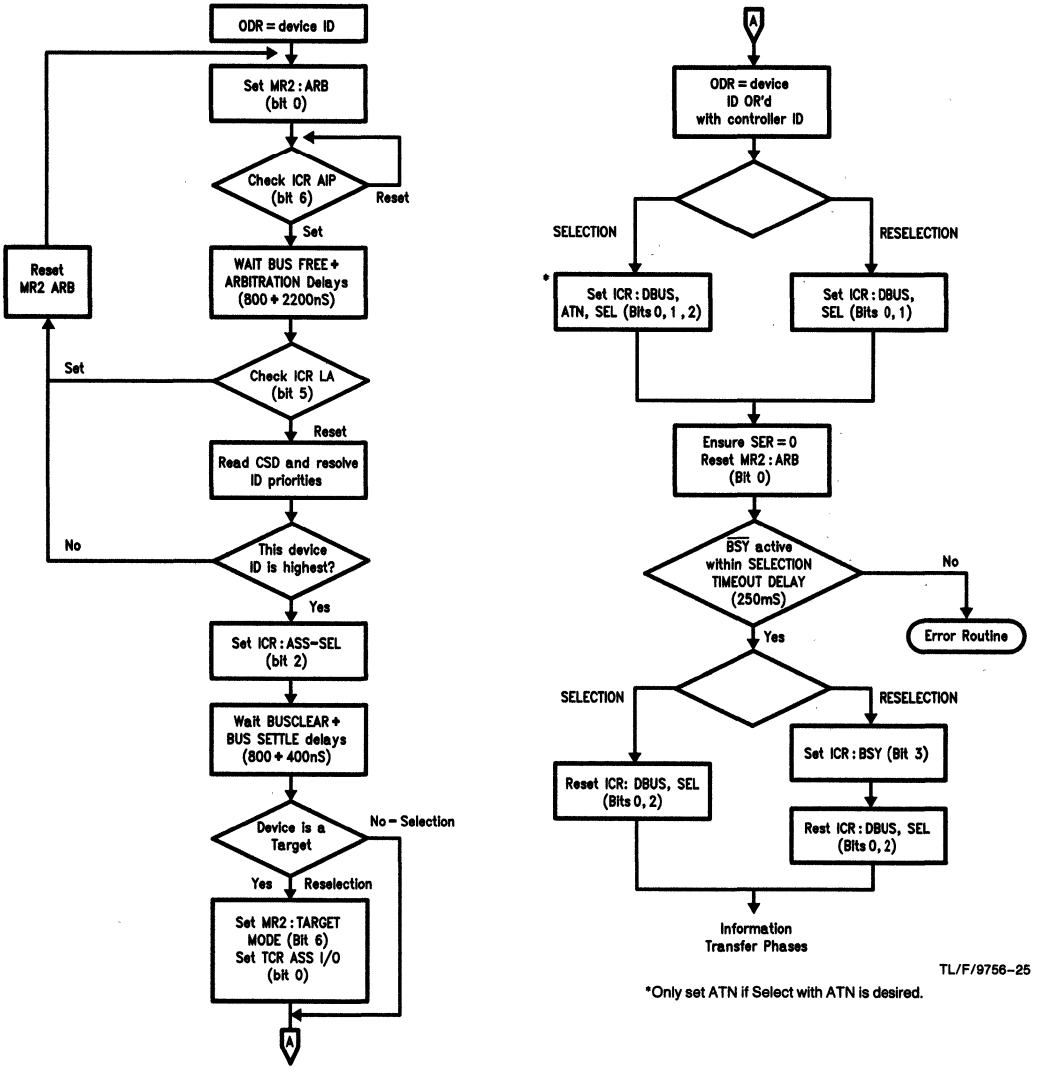
Note 1: Read enable (DMA) is DACK and RD active.

Note 2: This includes the RC delay inherent in the tests' method. These signals typically turn off after 25 ns enabling other devices to drive these lines with no contention.

Note 3: EOP, DACK, RD/WR must all be active for recognition of EOP.

Appendix A1

Arbitration and (Re)Selection



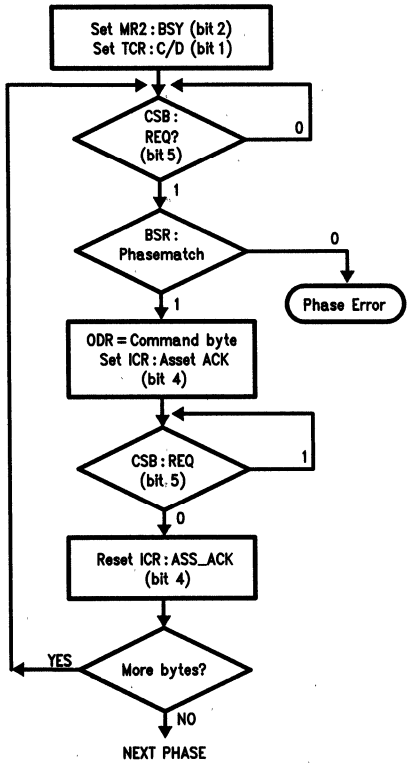
TL/F/9756–24

TL/F/9756–25

*Only set ATN if Select with ATN is desired.

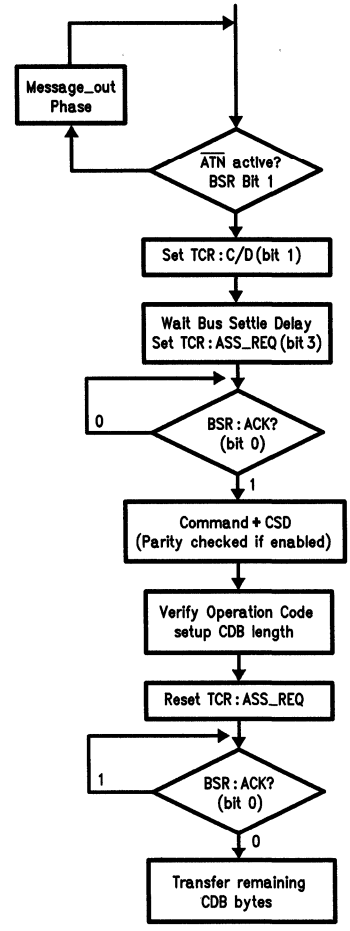
Appendix A1 (Continued)

Command Transfer (Initiator)



TL/F/9756–26

Command Transfer (Target)



TL/F/9756–27

Appendix A2

Register Chart

READ

Current SCSI Data (CSD)

Bit 7							Bit 0
$\overline{DB7}$	$\overline{DB6}$	$\overline{DB5}$	$\overline{DB4}$	$\overline{DB3}$	$\overline{DB2}$	$\overline{DB1}$	$\overline{DB0}$

Initiator Command Register (ICR)

Bit 7							Bit 0
$\overline{RST}$	AIP	LA	$\overline{ACK}$	$\overline{BSY}$	$\overline{SEL}$	$\overline{ATN}$	DBUS

Mode Register 2 (MR2)

Bit 7							Bit 0
BLK	TARG	PCHK	PINT	EOP	BSY	DMA	ARB

Target Command Register (TCR)

Bit 7							Bit 0
0	0	0	0	$\overline{REQ}$	$\overline{MSG}$	$\overline{C/D}$	$\overline{I/O}$

Current SCSI Bus Status (CSB)

Bit 7							Bit 0
$\overline{RST}$	$\overline{BSY}$	$\overline{REQ}$	$\overline{MSG}$	$\overline{C/D}$	$\overline{I/O}$	$\overline{SEL}$	$\overline{DBP}$

Bus and Status Register (BSR)

Bit 7							Bit 0
EDMA	DRQ	SPER	INT	PHSM	BSY	$\overline{ATN}$	$\overline{ACK}$

Input Data Register (IDR)

Bit 7							Bit 0
$\overline{DB7}$	$\overline{DB6}$	$\overline{DB5}$	$\overline{DB4}$	$\overline{DB3}$	$\overline{DB2}$	$\overline{DB1}$	$\overline{DB0}$

Reset Parity/Interrupt (RPI)—Mode N

Bit 7							Bit 0
x	x	x	x	x	x	x	x

X = Unknown

WRITE

Output Data Register (ODR)

Bit 7							Bit 0
$\overline{DB7}$	$\overline{DB6}$	$\overline{DB5}$	$\overline{DB4}$	$\overline{DB3}$	$\overline{DB2}$	$\overline{DB1}$	$\overline{DB0}$

Initiator Command Register (ICR)

Bit 7							Bit 0
$\overline{RST}$	TEST	DIFF EN	$\overline{ACK}$	$\overline{BSY}$	$\overline{SEL}$	$\overline{ATN}$	$\overline{DBUS}$

Mode Register 2 (MR2)

Bit 7							Bit 0
BLK	TARG	PCHK	PINT	EOP	BSY	DMA	ARB

Target Command Register (TCR)

Bit 7							Bit 0
x	x	x	x	$\overline{REQ}$	$\overline{MSG}$	$\overline{C/D}$	$\overline{I/O}$

Select Enable Register (SER)

Bit 7							Bit 0
$\overline{DB7}$	$\overline{DB6}$	$\overline{DB5}$	$\overline{DB4}$	$\overline{DB3}$	$\overline{DB2}$	$\overline{DB1}$	$\overline{DB0}$

Start DMA Send (SDS)

Bit 7							Bit 0
x	x	x	x	x	x	x	x

Start DMA Target Receive (SDT)

Bit 7							Bit 0
x	x	x	x	x	x	x	x

Start DMA Initiator Receive (SDI)—Mode N

Bit 7							Bit 0
x	x	x	x	x	x	x	x

X = Don't Care

Realize the Speed of Asynchronous SCSI

National Semiconductor
Application Note 575
Andrew M. Davidson

National Semiconductor's new SCSI Interface devices raise the standard of low cost SCSI.

National Semiconductor has two new low cost, high performance Small Computer Systems Interface (SCSI) devices; the DP5380 and DP8490. The DP5380 Asynchronous SCSI Interface (ASI) is pin and program compatible with the NMOS NCR5380 type device, but since it is manufactured in CMOS it offers speed and power advantages. The DP8490 Enhanced Asynchronous SCSI Interface (EASI) is pin compatible with the ASI, and program compatible until its enhanced mode is set. This mode offers features and architectural improvements which can greatly increase a system's throughput. However a major factor in favor of both devices is the improvement in their achievable data transfer rate.

Users who have replaced their NMOS 5380 type devices for National devices, in an existing application, have found improvements of between 10 and 15 percent in data transfer speeds. That is with no other changes to their application. For new designs, or performance upgrades, the gains possible can be much more dramatic.

In many applications the limiting factor in a data transfer is the DMA controller. This presented a problem to users i.e., "How fast **can** the National devices transfer data?" To determine this National designed an asynchronous DMA controller, using a programmable gate array, which can go as quickly as the SCSI device will allow it. **In these tests the National devices were up to 140% faster than the NCR5380 equivalent.**

SYSTEM OVERVIEW

A general SCSI application is shown in *Figure 1*. This shows the hardware required to interface a SCSI bus to a peripheral controller, without the devices to control the peripheral itself. The Central Processing Unit (CPU), the only "intelligent" device, controls all operations. It communicates with the SCSI bus through the EASI or ASI, which both support selection, reselection, arbitration and all other bus phases detailed in the ANS X3.131-1986 SCSI standard defined by the ANSI X3T9.2 committee. Since these devices can act as both target and initiator this could be the core of a peripheral controller or a host adaptor.

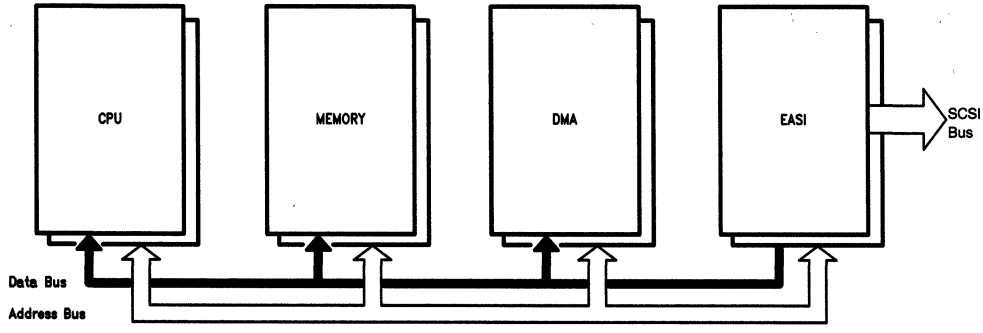
For optimum performance a DMA controller is used to transfer data between the SCSI device and memory. This memory is used as a cache for the final destination, which could be main system memory, a hard disk, floppy disk, printer, etc. By adding the appropriate interface the core shown in *Figure 1* can be adapted for the particular application.

TEST SYSTEM

To fully test the achievable DMA rate National built two boards of the form shown in *Figure 1*. The first board configures itself as an initiator, the second acts as a target. The initiator arbitrates for the SCSI bus, until successful, then selects the target and sends a Write command. This causes a DMA transfer from the initiator to the target.

The initiator arbitrates for the bus again, selects the target and sends a Read command. The data sent to the target is returned to the initiator, under DMA control, and checked. This process is constantly repeated, with the transmitted data varying between "01,FF,00,01 etc." and "FF,01,00,FF etc." This checks every data bit and varies the parity bit.

The two boards continually transmit blocks of data, alternating between Initiator Send/Target Receive and Initiator Receive/Target Send modes.



TL/F/10104–1

FIGURE 1. Core of a SCSI Controller

TEST CIRCUIT

The two boards have identical circuitry, shown in *Figure 2*. The CPU is an NSC800™ 8-bit CMOS microprocessor, which requires an octal latch to demultiplex the address and data bus. The boards configuration, initiator or target, depends on the system software stored in the EPROM. Fast static RAM is required for the cache, CMOS RAM with an access time of 45 ns was selected. A PAL® is used to decode the chip selects. The SCSI controller can be either a 5380 or a DP8490.

The DMA controller was implemented in a Xilinx Programmable Gate Array. These devices offer a quick, easily adaptable method of prototyping. The design is entered as a schematic, then converted to the gate array format by a software package. Routing is largely automatic, but complex designs require a certain amount of interaction. In a speed critical design, such as the DMA controller, manual control of routing delays is vital.

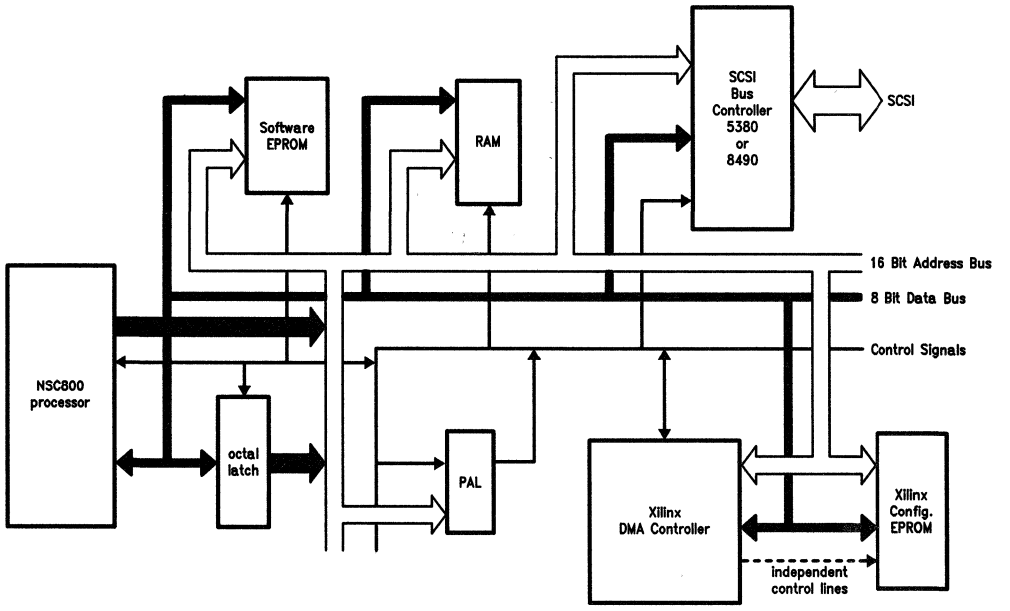
The Xilinx device can automatically load its configuration from an EPROM on power-up. This could be the same EPROM that contains the boards software, but for simplicity during prototyping a separate device was used.

DMA DESIGN

The DMA controller required was highly application specific and could therefore be relatively simple. It basically consists of a counter, a register and control logic *(Figure 3)*. The counter is loaded with the start address of the DMA block and the register with the end address. On a subsequent DMA request the controller has to take control of the microprocessor bus and transfer data until the address in the counter equals the address in the end register.

The DMA is enabled by setting a bit in the control register. The other control bit determines whether the transfer is a memory read or a memory write.

Once enabled, a subsequent DMA request (DRQ) will cause the device to issue a bus request ($\overline{BRQ}$). When it receives a bus acknowledge ($\overline{BACK}$) from the processor it asserts the counter output onto the address bus and issues a DMA acknowledge ($\overline{DACK}$). The DMA controller also takes control of the I/O and memory read and write strobes. This device only operates in block mode, so $\overline{DACK}$ is asserted throughout the transfer. The SCSI device deasserts DRQ when $\overline{DACK}$ goes active. This is used to start the first transfer. Subsequent transfers are triggered by the throttle control, READY, going high.



TL/F/10104–2

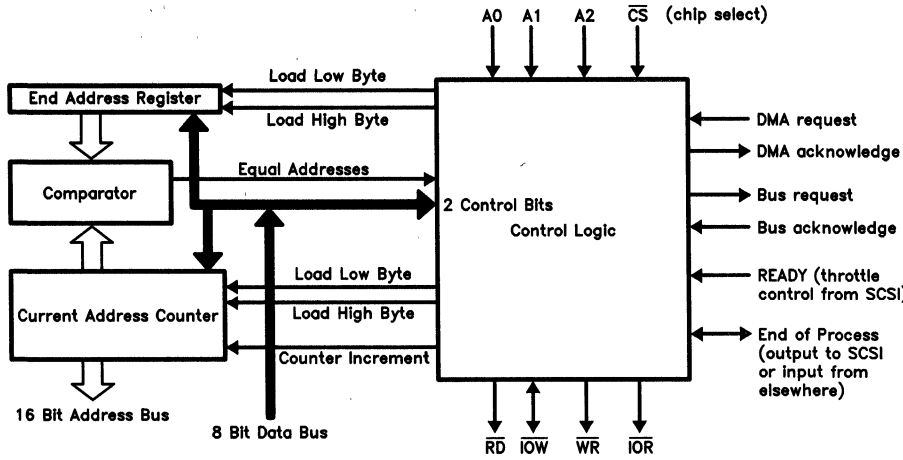
FIGURE 2. Circuit of SCSI Test Board

For a memory read, $\overline{RD}$ is held active throughout the transfer and $\overline{IOW}$ is strobed. For a memory write, both $\overline{IOR}$ and $\overline{WR}$ are strobed. The basis of the control logic is shown in *Figure 4*.

Figure 5 shows the timings of the circuit in *Figure 4*. The I/O or memory strobe pulses low for the time difference between READY going high and the STROBEIN input causing CLOCK to go high. Thus the external delay line controls the strobe signals pulse widths. The back edge of CLOCK is used to increment the address counter.

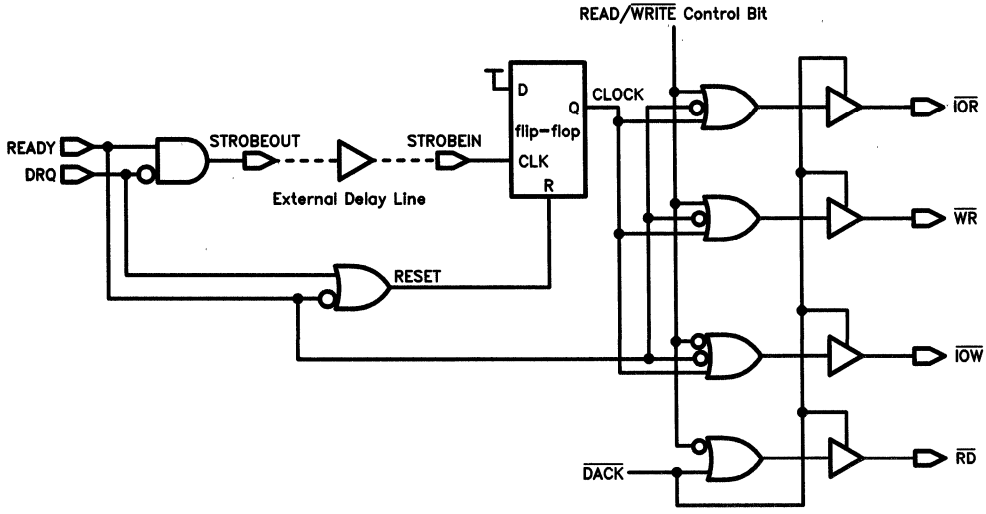
The controller continues as in *Figure 5*, until the $\overline{EOP}$ (End of Process) signal is asserted. This is either asserted externally, to prematurely end a transfer, or by the DMA controller itself, once the current address counter equals the end address register.

The speed that the DMA controller transfers data is determined by the READY line. The SCSI controller deasserts READY after an $\overline{IOW}$ or $\overline{IOR}$, then asserts it when it is ready to transfer the next byte. This high transition triggers the next transfer of the DMA controller. Varying delay line time varies the $\overline{IOR}$, $\overline{IOW}$ and $\overline{WR}$ pulse widths, so different speeds of devices can be accommodated by this controller.



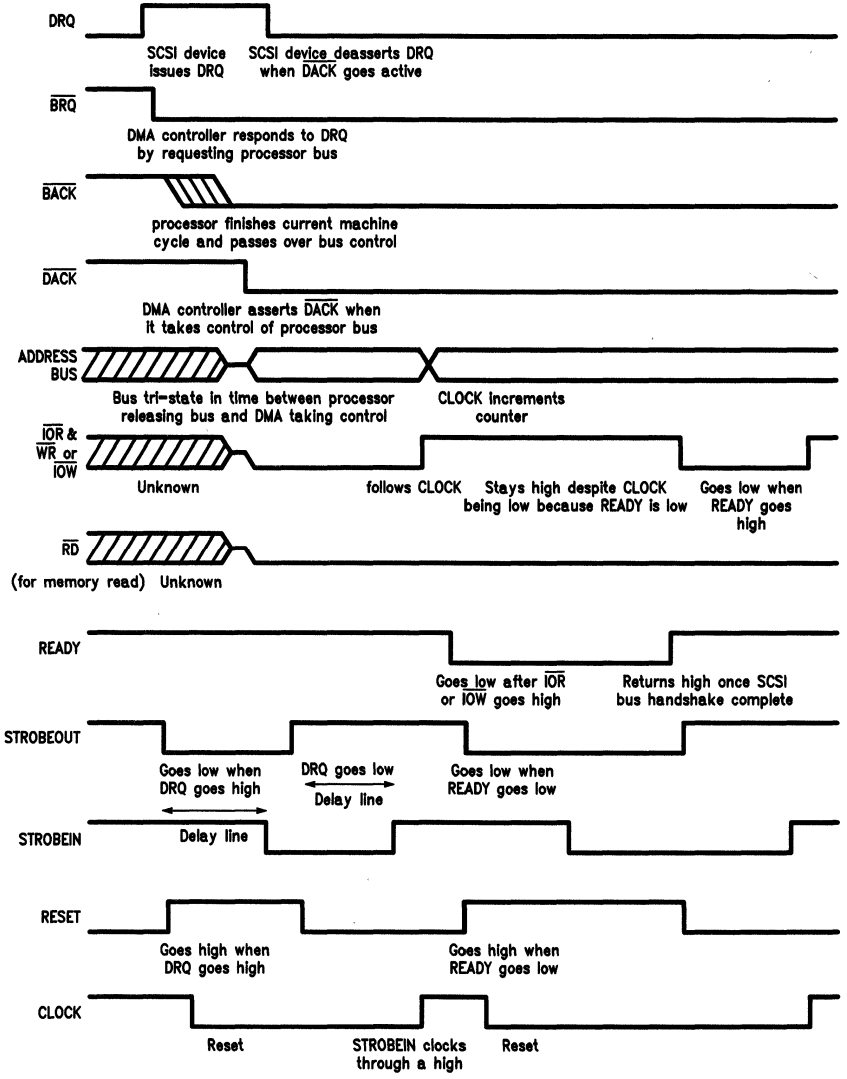
TL/F/10104–3

FIGURE 3. Block Diagram of DMA Controller



TL/F/10104–4

FIGURE 4. DMA Control Logic



TL/F/10104–5

FIGURE 5. DMA Controller Timings

BENCHMARKING

The system used for the benchmarking is shown in *Figure 6*.

The two boards were connected by three different lengths of cable; 50 cm, 3m and 6m (the maximum length of single ended cable allowed). 50-way STD IDC ribbon cable was used, wired as per SCSI specification with every second line grounded. All SCSI signal lines were terminated on both boards, with a 330Ω resistor to ground and a 220Ω resistor to V_{CC}.

A Hewlett Packard 16500A Logic Analysis system was used to monitor the control signals and measure the burst DMA rate. The 10 ns sampling rate gave sufficient accuracy to measure handshaking speeds which were all below 5 Mbytes/s.

The first byte of a transfer is delayed, as the initiator takes time to follow the phase set by the target. After this the DMA controller reaches its peak transfer rate, which is maintained until the end of the transfer. The frequency of the SCSI bus handshaking signal, REQ, was measured, to give the data transfer rate.

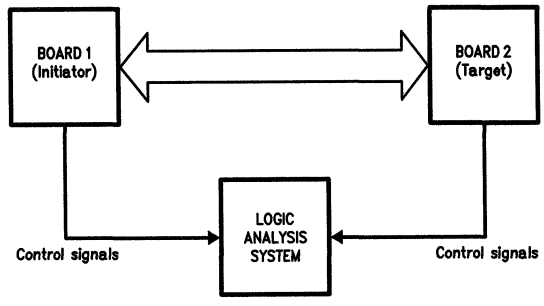
BENCHMARKED DEVICES

Initially it was planned to benchmark three manufacturers devices, that is National Semiconductor, NCR and Logic Devices.

However it was found that the Logic Devices part caused problems in the test board and would not function in conjunction with any other manufacturers device. The DMA controller uses DRQ to generate the first read and write strobes (see DMA Design section), but expects the signal to stay inactive for the rest of the transfer. The Logic Devices 5380 produces a spurious DRQ during a block mode transfer. This corrupts the read and write strobes produced by the DMA controller. The Logic Devices 5380 would not interface to other manufacturers devices because of its parity checking method. National and NCR only check the bus parity as they load their input data register. Logic Devices check parity when $\overline{ACK}$ is active, for Target Receive or when REQ is active for Initiator Receive. However, the National and NCR devices write the next byte of data before the $\overline{REQ}/\overline{ACK}$ handshake for the previous byte is complete. As parity is decoded through logic the data will be on the bus before the parity bit is valid. Logic Devices detect this as a parity error, which causes the transfer to be terminated.

The National and NCR devices were fully interchangeable.

Different Lengths of SCSI Bus



TL/F/10104–6

FIGURE 6. Benchmarking System

BENCHMARK RESULTS

The transfer rate during a data phase was measured for a sample of National and NCR devices, over three lengths of cable. Measurements were made for Initiator Send/Target Receive mode and for Initiator Receive/Target Send mode. When a NCR device was used the delay line time was increased, to allow for its slower reading and writing times. *Figures 7a, b* and *c* show the results.

These tables show the dramatic speed advantages of the National device over NCR. The National device maintains a significant advantage over all three lengths of cable. It may appear that the NCR device maintains its speed better over longer lengths of cable, but only because the added propagation delays of the cable are a smaller percentage of the transfer time.

The Initiator Receive/Target Send mode is inherently faster, which makes the specification of one DMA rate for the device a highly debatable subject, even if the length of cable for the measurement is agreed. However, customers using the SCSI device for a disk application will appreciate the fact that this mode is faster, since approximately 80% of disk accesses are reads, while running most software.

Even if only one National device is used there is a significant improvement in performance, but to fully realize the potential of the SCSI interface two National devices are required. *Figure 8* shows just how much of an improvement this is.

a. 10cm Cable

Initiator / Target	Speed in Mbytes/second	
	National	NCR
National	**3.5** **4.9**	2.6 2.7
NCR	1.8 2.3	**1.5** **2.0**

b. 3m Cable

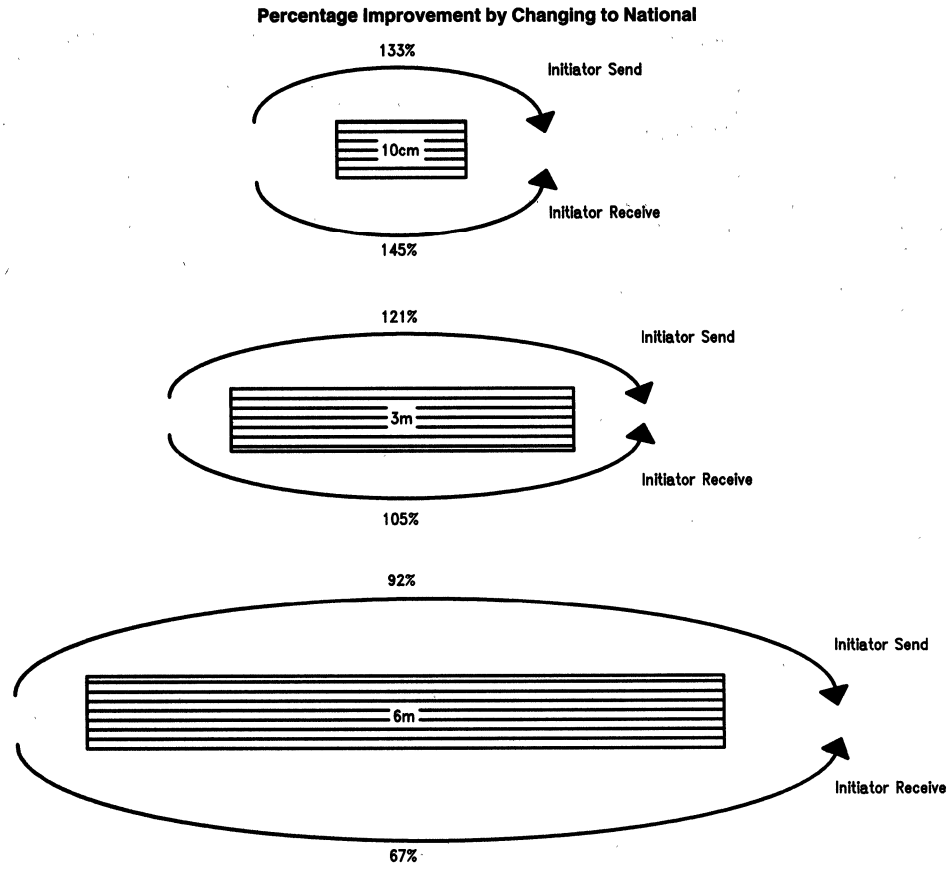
Initiator / Target	Speed in Mbytes/second	
	National	NCR
National	**3.1** **3.9**	2.3 2.5
NCR	1.6 2.2	**1.4** **1.9**

c. 6m Cable

Initiator / Target	Speed in Mbytes/second	
	National	NCR
National	**2.5** **3.0**	2.0 2.2
NCR	1.4 2.0	**1.3** **1.8**

Note: The upper value shows Initiator Send/Target Receive while the lower shows Initiator Receive/Target Send.

FIGURE 7. Measured DMA Transfer Rate



TL/F/10104–7

FIGURE 8

CONCLUSIONS

The new asynchronous SCSI interface devices from National offer an instant improvement to old designs. By changing over to National devices the user can upgrade an existing SCSI interface by upwards of 10%, with no further effort. For greater performance improvements a fast DMA controller is required, at which point the asynchronous SCSI approaches synchronous SCSI speeds.

For new designs the DP8490 EASI is the ideal choice. In addition to the fast DMA transfer rate, this device offers architectural improvements which can greatly boost a systems throughput. In a 5380 the arbitration time is dead to the system, as the device must be polled to determine when this phase has started. By interrupt driving arbitration the DP8490 EASI frees many milliseconds, or even seconds, to be used at the system designers discretion. This time may be utilized for disk cacheing, overlapped seeks, printing etc. depending on the particular application. The entire interrupt structure has been revamped, making software for this device quicker to develop and quicker to run. To further ease software overheads the troublesome bugs of the 5380 have been fixed.

By improving the speed of data transfers, lowering the power consumption and offering new additions to a familiar SCSI interface device, National have rejuvenated asynchronous SCSI.

DP8490: E.A.S.I. Does It!

National Semiconductor
Application Note 562
Andrew M. Davidson

National's Enhanced Asynchronous SCSI Interface (EASI) offers features that can yield increases in system performance over designs incorporating a 5380 type device.

When looking for a low cost Small Computer Systems Interface (SCSI) controller many users adopt the 5380 Asynchronous SCSI Interface (ASI) despite its many documented flaws and undocumented difficulties in operation, familiar to any past user. This device tended to be selected due to the lack of an alternative, but the DP8490 EASI from National Semiconductor will change this. The EASI is pin compatible with the ASI, and software compatible until an enhanced mode bit is set. This enhanced mode offers features which can increase the performance of a system currently using a 5380 type device. This is achieved through improvements to speed and architecture, while eliminating the 5380's inherent bugs.

SYSTEM OVERVIEW

The DP8490 and DP5380 support selection, reselection arbitration and all other bus phases as detailed in the ANS X3.131–1986 SCSI standard defined by the ANSI X3T9.2 committee. As the devices can act as both TARGET and INITIATOR they are suitable for any SCSI bus application. A typical application will look like *Figure 1*.

The C.P.U. controls all operations by reading and writing registers in the I/O devices; to achieve a high performance a member of the HPC16000 microcontroller family can be used. For optimum performance a DMA controller handles data transfers between the EASI and memory. The memory is used as a data cache for the final destination, which may be a hard disk, RAM disk, printer etc. *Figure 1* shows the basic system but not the hardware required to interface to such a peripheral. The EASI has high-current open-drain drivers which interface directly to the SCSI bus.

SPEED AND POWER IMPROVEMENTS

National's DP8490, and DP5380, are implemented in low voltage silicon gate microCMOS, which gives power and speed improvements over existing NMOS 5380 parts. The National devices have a maximum supply current of 4 mA, compared to the NCR5380 145 mA, and DMA rates of over 3 Mbytes/second, compared to 1.5 Mbytes/second. National's EASI device has reduced timing parameters; read access times are reduced from 130 ns to 50 ns, write data hold times, from 30 ns to 10 ns and all other parameters correspondingly improved.

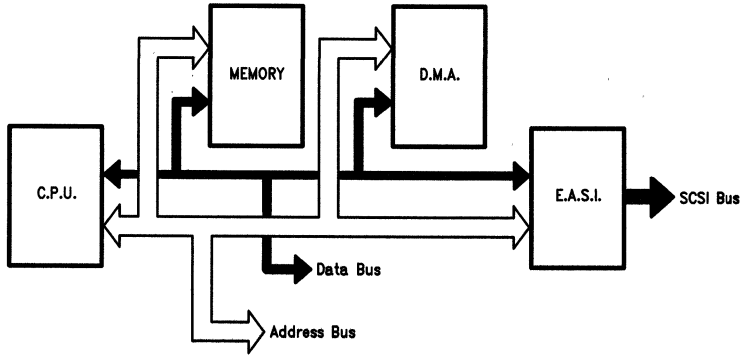
ENHANCED INTERRUPTS

The DP8490 enhanced mode interrupt structure is considerably different to that of the 5380. The source of interrupt is available by reading one register, the Interrupt Status Register (ISR). In order to find the source of interrupt in the 5380 two registers must be read. In addition all interrupts are flagged in enhanced mode. With the 5380 a selection interrupt has to be determined by the absence of other flags while the select line is active. Since selection is the means of establishing contact with other devices on the SCSI bus it is a common occurrence. The DP8490 supplies an interrupt flag for this and all other interrupts. This enhanced interrupt structure should simplify system software, thus increasing the speed of interrupt servicing. Simplified code is easier to understand, and therefore easier to adapt or repair.

All interrupts are maskable, using the Interrupt Mask Register (IMR), but interrupts cannot be lost since resetting the interrupts only resets those bits that were active on the last read of the ISR. This achieves a greater error tolerance.

INTERRUPT DRIVEN ARBITRATION

An enhancement which will have a great effect on many system's performance is the addition of an arbitration interrupt. In the 5380, arbitration is polled. On a busy SCSI bus, arbitration will take typically many milliseconds, and potentially many seconds. Therefore, time which could have been spent doing overlapped seeks, or filling a data buffer is instead used reading a 5380 register waiting for a flag to go active. The enhanced mode of the DP8490 offers interrupt driven arbitration, allowing the user to utilize this time in data cacheing, data manipulation, etc., thus increasing the system throughput. This increase in system performance can become particularly effective in low priority devices which will typically have to arbitrate for the bus the longest.



TL/F/10081–1

FIGURE 1

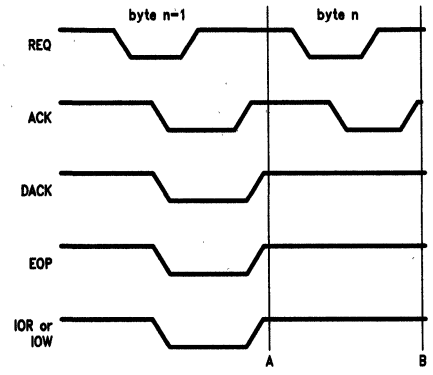
INCREASED TESTABILITY

The enhanced mode is set using a register bit which, in the 5380 initiates the 'Test Mode'. In Test Mode all output drivers on the device are disabled, so although the device can still be written, no data can be read. This makes it unsuitable for any application. In addition, DMA and interrupt requests are tri-stated, which may cause system problems.

In contrast, the enhanced mode of the DP8490 offers a loopback facility, where the SCSI drivers are disabled, and the SCSI I/O's loop back inside the EASI. This loopback test mode allows all device signals to be fully tested, including a DMA transfer. Using this, a system can perform a full self diagnostic test, without affecting the SCSI bus. Good diagnostic testing simplifies system error detection and checking.

TRUE END OF DMA DETECTION

Another interrupt improvement in the enhanced mode is seen during DMA operation. The 5380 will generate an end of DMA interrupt when it sees a concurrent EOP, DACK and IOR or IOW, even though the SCSI bus transfer may not be complete. To overcome this the user must examine REQ and ACK, to determine a true end of transfer. Consider *Figure 2*:



TL/F/10081–2

FIGURE 2

If an initiator is slow in removing the ACK, the processor could sample REQ and ACK after the EOP interrupt, at point A. At this point REQ and ACK are both inactive, although there is still one byte to transfer. The user must see REQ and ACK inactive on three successive transfers to be confident the transfer is complete (point B). Thus a great deal of time is wasted ensuring the last byte of every DMA transfer is successful. The EASI eliminates this inefficiency by detecting, and interrupting on, true end of DMA, when ACK goes inactive.

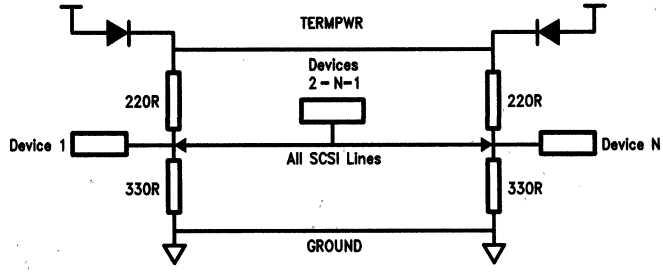
TIMING ENHANCEMENTS

Some of the major bugs in the 5380 are experienced in DMA mode. In initiator receive mode, if a REQ is received, after a valid EOP, an ACK will be generated although no valid data exists. This could be a particular problem for users who have to split a block transfer into two, perhaps to prevent crossing a page boundary. Part of the second block of data could be lost by the generation of these spurious ACK's. Another problem experienced by an initiator in DMA mode is ACK being left asserted after receipt of a valid EOP. After receiving the end of DMA interrupt the processor must 'manually' deassert ACK by writing to the relevant register.

These problems make the DMA mode in a 5380 extremely difficult to use. Extra software is required to ensure correct operation, making boards incorporating a 5380 run slower. **None of these problems exist in the enhanced mode of the DP8490.**

BUS TERMINATION

The most common SCSI bus is a 50 way ribbon cable, of up to six meters long, with SCSI devices daisy chained along it. The devices at either end of the bus should terminate all SCSI lines with a 330Ω resistor to ground and a 220Ω to power. To ensure the bus can operate correctly, even if the device at one end is not powered up, the power to these terminators can be made available on the SCSI bus. This allows the terminators to always receive power. The bus configuration is shown in *Figure 3*. Terminator power is fed through a Schottky diode to prevent a backflow of power into either of the devices.



TL/F/10081–3

FIGURE 3

Some manufacturers' CMOS devices will not work in this configuration, since when not receiving power they pull the SCSI line low. Therefore all devices on the bus must be powered up. Both National's DP5380 and DP8490 have a special input protection which makes this configuration acceptable.

EXTENDED FEATURES

The DP5380 and DP8490 are available in a standard 40 pin DIP, or in a 44 pin PLCC, which is pin compatible with existing 5380 PLCCs. The DP8490 PLCC has a microprocessor parity pin. In the enhanced mode, microprocessor bus parity checking can be enabled, with parity polarity optional. Since parity checking is optional, the PLCC DP8490 can still be used in a system where the microprocessor does not support parity. However, use of this feature will help the confirmation of data validity throughout the system.

Enhanced mode (for both types of packaging) offers other features, including programmable SCSI parity polarity. Although the specified SCSI polarity is ODD, by enabling EVEN parity bus diagnostics can be carried out. In this way the error detection and error handling capabilities of other devices on the bus can be determined.

The enhanced mode also has a general phase mismatch interrupt. The 5380 only checks phase mismatch during DMA. This means that an initiator following the phase set by a target must poll the device to determine when it changes. Only a change during DMA produces an interrupt. In the enhanced mode, an option available gives an interrupt on any phase mismatch. This allows for quicker detection of a change of phase, thus decreasing the dead time on the SCSI bus.

The wide range of new features available in the DP8490 makes this part a first choice in any new SCSI designs, or upgrades requiring increased performance. For users not wishing to make the software changes, required to utilize the enhanced features of the DP8490, the DP5380 is available, offering speed and power improvements over existing parts, at a highly competitive cost. Every user that has tried replacing existing NMOS 5380s with either the DP5380 or DP8490, at both ends of their existing application, have experienced data transfer speed improvements of between 10% and 15%. That is with no other changes!

A SCSI Printer Controller Using Either the DP8490 EASI or DP5380 ASI and Users Guide

National Semiconductor
Application Note 563
Andrew M. Davidson

The DP8490 Enhanced Asynchronous SCSI Interface and DP5380 Asynchronous SCSI Interface are CMOS devices, which offer a low cost high performance Small Computer Systems Interface. These devices are pin compatible, and software compatible until an enhanced mode bit is set in the DP8490. This enhanced mode offers many new features which can yield increases in system performance through software, in addition to the improvements in speed and power shown by both devices over existing NMOS devices.

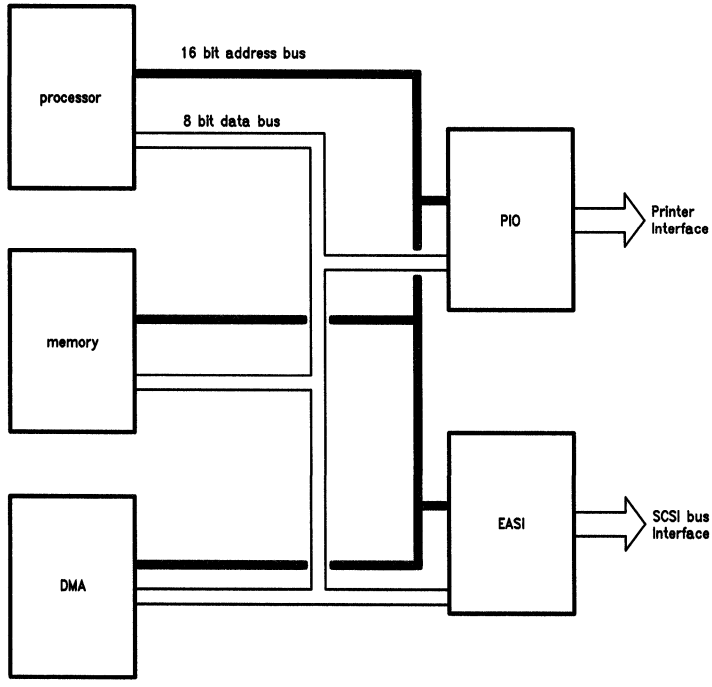
This application note shows how the hardware and software can be designed for a SCSI Printer Controller (SPC) so that it can incorporate either the DP5380 or DP8490. Since the software automatically detects which device is inserted either can be used, although the enhanced mode of the DP8490 offers a better system in terms of throughput and error tolerance. All of the software discussed is available on a floppy disk.

1.0 Introduction

The SCSI Printer Controller *(Figure 1.1)* consists of five main parts:

1. Microprocessor NSC800™
2. Printer Interface NSC831 PIO (Parallel Input/Output)
3. SCSI Interface DP8490 or DP5380
4. Memory
   CMOS EPROM NMC27C256
   CMOS Static RAM 62256
5. DMA Controller 9517 or 8237

The NSC800 is an eight bit CMOS microprocessor which is the central processing unit of the National Semiconductor



TL/F/10082–1

FIGURE 1.1

1.0 Introduction (Continued)

NSC800 microcomputer family. It is capable of addressing 64 kbytes of memory and 256 I/O devices using a multiplexed address and data bus. The instruction set is fully compatible with that of the Z80.

The NSC831 is the parallel input/output (PIO) device of the NSC800 family. This provides 20 I/O bits which can be individually programmed to be inputs or outputs. These are configured as two eight bit ports and one four bit port. The PIO is used as the interface between the microprocessor and the printer.

The DP5380 and DP8490 comply with the ANS X3.131-1986 SCSI standard as defined by the ANSI X3T9.2 committee. They can act as both Initiator and Target supporting all bus phases. Due to the on-chip high-current open-drain drivers the devices interface directly to the SCSI bus.

The 64 kbyte memory space is split between a 32 kbyte EPROM, containing the run-time software, and a 32 kbyte static RAM.

Data transfers between memory and the EASI can be controlled by the DMA controller, which supplies all read and write strobes for a transfer in either direction. This is a synchronous device which uses the 4 MHz clock output from the microprocessor.

The microprocessor is normally in control of the internal busses, giving it the ability to read or write memory or I/O devices. During a DMA transfer the DMA takes control of these busses and can pass data between the EASI and memory. As the microprocessor is the only device with 'intelligence' it must control these transfers. It commences and controls operations by setting registers in the DMA and EASI.

2.0 Hardware

2.1 DEVICES

2.1.1 Introduction

This section is intended to explain the hardware of the SPC referring to the circuit diagram given in Appendix A. This will describe the devices used and the signals generated but not the way in which they are programmed. What will be discussed is how the devices relate to each other, their relative timings, and any extra hardware required. A more exact description of the internal registers of the EASI, DMA and PIO, and their operation, will be given in the diagnostics section.

2.1.2 Microprocessor

The NSC800 is an eight bit microprocessor which multiplexes the eight bit data bus with the lower half of the address bus to create a sixteen bit address bus. An octal latch, MM74HCT373, is required to hold the address on the bus during a memory read or write, with the ALE (Address Latch Enable) output from the NSC800 strobing the latch.

Since the bus has devices which use both TTL and CMOS levels pull-up resistors are required. The $\overline{IOM}$ output signifies whether the processor is on a memory or an I/O cycle and is used along with the $\overline{RD}$ and $\overline{WR}$ outputs to strobe bus data.

This processor allows control of the bus to be passed to an I/O device, using the $\overline{BRQ}$ (bus request) input and $\overline{BACK}$ (bus acknowledge) output. On this board the DMA is the only device which may request control of the bus.

When the EASI issues a DMA request the DMA signals that it needs control of the bus by issuing a $\overline{BRQ}$. The processor acknowledges this by issuing $\overline{BACK}$, and then drives the bus and all related control signals to their high impedance state. The MM74HCT373 must TRI-STATE® its outputs; the output enable is driven by the DMA signal AEN (address enable), which it asserts when it has control of the bus. At the end of its operations the DMA releases AEN (enabling the latch outputs) and negates $\overline{BRQ}$. The processor de-asserts $\overline{BACK}$ and retakes control of the bus by enabling its outputs. To prevent spurious signals being generated when the bus is not driven the control signals must have pull-up resistors.

The NSC800 provides five hardware interrupts of which only $\overline{RSTA}$ is used. The others are tied inactive. $\overline{RSTA}$ is driven by the EASI interrupt output, with an inversion between them to allow for the difference in active levels. An interrupt on this pin causes a software reset to a particular address in memory (more details of interrupt servicing will be given in the diagnostic software section).

The system clock is generated using an 8 MHz crystal, with the frequency divided by two for use by the processor and the DMA. There is also power-on reset circuitry, which resets the processor, and causes a reset output to the other devices when power is first applied.

2.1.3 DMA Controller

The 9517 and 8237 are compatible direct memory access devices, controlled by setting internal registers. These registers are selected using a chip select, with the particular register selected by the lower nibble of the address bus, and data strobed by $\overline{IOR}$ (I/O read) or $\overline{IOW}$ (I/O write). These registers control the type of data transfer, the number of bytes transferred and the memory address the transfer is to/from.

The two modes of transfer used are single mode and block mode. In single mode the DRQ (DMA request) input and the $\overline{DACK}$ (DMA acknowledge) output are used to "handshake" every byte of data transferred. In block mode, after an initial DRQ, $\overline{DACK}$ must be active until after the last byte has been transferred. The rate of transfer is controlled by the READY input.

Another device required in conjunction with the DMA is an MM74HCT74 'D' type flip-flop. This is used to overcome the metastability problem introduced by the asynchronous READY signal driving a synchronous device. The flip-flop synchronizes the input with the system clock.

The DMA supplies all the necessary control signals to move data from memory to EASI or vice-versa. When all data has been transferred the DMA drives $\overline{EOP}$ (end of process) active. $\overline{EOP}$ can also be used as an input to the DMA, prematurely terminating any transfer. It is configured as an open-drain driver, so requires a pull-up resistor, of an advised value of 4.7 kΩ.

As in the processor the DMA has a multiplexed address and data bus, also requiring an MM74HCT373 to latch the address. In this case it is the upper byte of the address bus which is multiplexed with the data bus. The strobe signal is similar to ALE but is generated in the DMA and called ADSTB (address strobe). The TRI-STATE for this latch is driven by the inverse of AEN, since the device must drive the bus only when the DMA has control.

2.0 Hardware (Continued)

The 9517 and 8237 have four DMA channels, of which only one is used. The DRQ inputs for the other three channels are tied inactive. The RESET input causes the control registers to be cleared and the DRQ inputs to be masked.

2.1.4 EASI

The EASI is controlled by setting internal registers, written by an $\overline{IOW}$ and chip select, read by an $\overline{IOR}$ and chip select. The particular register selected is determined by the lowest three bytes of the address bus. The EASI has an eight bit microprocessor data bus, and on the PLCC part a microprocessor bus parity pin. However the NSC800 does not support parity checking so this bit is tied low. The other microprocessor controlled pin is the $\overline{RESET}$. This causes the EASI to clear all registers and therefore reset all logic.

The DRQ output and $\overline{DACK}$ input "handshake" single mode DMA, while the READY output is used to control the speed of a block mode transfer. $\overline{EOP}$ is an input which terminates DMA, and can be used to cause an interrupt. This interrupt output INT can interrupt the microprocessor if the EASI detects an error, or has completed some task.

SCSI uses an eight bit data bus with a parity bit; which must support odd parity during all bus transactions except arbitration. Other SCSI signals are the bus selection control signals $\overline{SEL}$ and $\overline{BSY}$, phase control signals $\overline{MSG}$, $\overline{C/D}$ and $\overline{I/O}$, data transfer handshakes $\overline{REQ}$ and $\overline{ACK}$ and the message flag $\overline{ATN}$. Further details on the use of these signals will be given in the software sections of this document. The SCSI reset $\overline{RST}$ is similar to a chip reset, but generates an interrupt.

The EASI interfaces directly to the SCSI bus using high-current open-drain drivers. This bus is a 50-way ribbon cable (maximum length 6.0 meters) on which all SCSI devices are daisy-chained. The devices at the end of this cable must terminate the SCSI signals i.e., a 220Ω resistor to power, and a 330Ω to ground. This power can be V_{CC} or TERMPWR (terminator power).

TERMPWR is V_{CC} fed through a Schottky barrier diode. This can be fed to pin 26 of the connector allowing the SCSI device at one end of the bus to be powered down without affecting the bus. Since the terminators are receiving power the bus can operate effectively. To make this optional TERMPWR is fed to the connector through a jumper link. The Schottky diode must be included to prevent backflow of power into the printer controller board.

Other manufacturers CMOS devices can not be used in a configuration like this, as they pull the bus low when not powered. The DP5380 and DP8490 have a special input protection to overcome this problem.

All other pins on the connector should be tied to ground, except pin 25 which must be left floating.

2.1.5 PIO

The printer controller uses a NSC831 PIO to interface between the processor and the printer. The NSC831 has 20 individually programmable I/O bits which are arranged as three ports; A, B and C. As it is part of the NSC800 family the NSC831 has a compatible multiplexed address and data bus with an ALE input, to strobe the address into an internal latch. This eight bit address can then be used to access an internal register, in conjunction with an $\overline{IOR}$ or $\overline{IOW}$ and chip select. The device has two chip selects, one of which is tied active (low), with the other coming from the address decode PAL®. The RESET input causes the three ports to become all inputs.

In this application the three ports are set up so that each port is all input or all output. Port A is an output, which drives the printer data bus through an MM74HCT241 octal buffer. Port B is an input, with the lower nibble coming from the printer outputs, driven by half an MM74HCT240 octal inverting buffer. The pull-down resistors on the $\overline{ERROR}$ and PE (paper error) inputs give a proper error signal if the printer is unconnected i.e., the same as when the printer is off-line. The upper three bytes of Port B are connected to switches, which are used to set the board's SCSI I.D. This will be further explained in the SCSI diagnostics section. Only three bytes of Port C are used, all as outputs, driven by the same device used by Port B. Bits 0 and 2 are printer outputs, while bit 3 drives a LED. This displays the 'health' of the board i.e., when the board is operational the LED is on, when non-operational it is off and when there is a known hardware error a message is 'flashed'.

The printer connector is a standard IBM® 25 way 'D' range, with all unused pins grounded, except 13.

2.2 PAL EQUATIONS

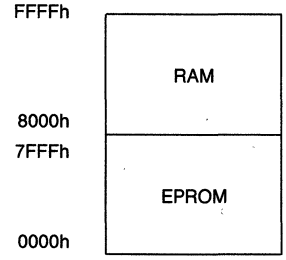
2.2.1 Introduction

PAL's are used in this board to generate chip selects, overcome potential timing problems and to invert signals for devices with different active levels. The PAL's used are National Semiconductor's PAL16L8. The equations are written in a form compatible with PLAN (Programmable Logic Analysis by National) Ver. 2.00.

2.2.2 Decode PAL

The decode PAL supplies the five main devices with their chip selects.

Memory is split into two 32k byte blocks, each of which represents one device. These devices are an EPROM, at the base of memory, and RAM in the upper half. Thus the memory map and chip select equations are as follows:



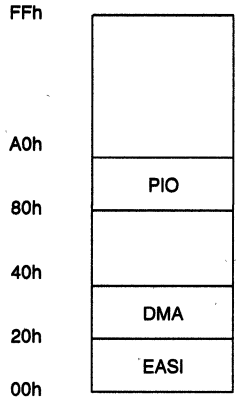
/EPROMZ = /A15*BACKZ*/IOMZ

/RAMZ = A15*BACKZ*/IOMZ + A15 * AEN

EPROMZ (the Z at the end of the name shows it is active low) can only be selected by the processor, while RAMZ is available to the DMA.

2.0 Hardware (Continued)

The DMA, PIO and EASI fit into the I/O map shown, which results in the equations following. These equations show that the devices can only be selected by the microprocessor on an I/O cycle.



/EASIZ = /A7 * /A6 * /A5 * IOMZ * BACKZ
/DMAZ = /A7 * /A6 * A5 * IOMZ * BACKZ
/PIOZ = A7 * /A6 * /A5 * IOMZ * BACKZ

The other three outputs are used as inverters:

/BRQZ = HRQ
/AENOZ = AEN
/HLDA = BACKZ

The two spare inputs are tied low.

2.2.3 Control PAL

The NSC800 uses $\overline{IOM}$ to distinguish between memory and I/O cycles. This allows $\overline{IOR}$ and $\overline{IOW}$, for processor cycles to be generated.

/IORZ = /RDZ * IOMZ
IORZ.TRST = BACKZ
/IOWZ = /WRZ * IOMZ
IOWZ.TRST = BACKZ

The second line of these equations shows that the output is TRI-STATE when the processor relinquishes control of the bus.

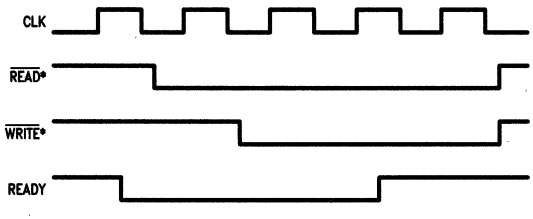
Before examining the next five equations it is important to fully understand the relative signal sequencing involved in a DMA transfer. When the transfer is initiated the EASI issues a DRQ, causing the DMA to respond with $\overline{DACK}$, and take control of the bus by asserting $\overline{BRQ}$. The processor asserts $\overline{BACK}$ and allows the bus and relevant control signals to go TRI-STATE. The DMA then asserts AEN, to show it is in control of the bus, causing the microprocessor address latch to TRI-STATE its outputs and the DMA latch to enable its outputs.

For single mode each byte transferred must have a DRQ and a $\overline{DACK}$. For block mode DRQ returns inactive after the DMA responds with a $\overline{DACK}$, but the $\overline{DACK}$ must remain active until the transfer is complete. The READY line controls the rate of block mode DMA transfer i.e., when READY is low the byte transfer is 'frozen', until READY returns high (*Figures 2.1a* and *2.1b*).

The READY signal is asynchronous but the DMA is synchronous, testing the READY signal on the negative edge of each clock. For this reason the READY input must be synchronized, using a flip-flop which updates its output on the positive edge of the clock.

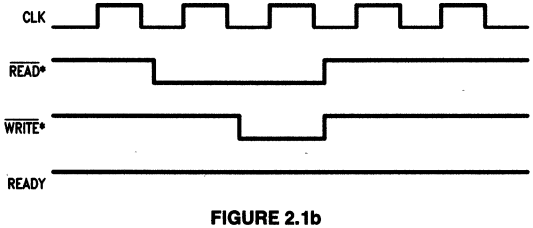
When READY is detected as being inactive on the negative edge of the clock an extra cycle is inserted to the read and write.

While data is being transferred a register keeps track of how many bytes are left. When this reaches zero an $\overline{EOP}$ signal is generated, concurrent with the last read and write, causing the EASI to set an EOP flag.



TL/F/10082–2

FIGURE 2.1a



TL/F/10082–3

FIGURE 2.1b

***Note:**

Of these read and write signals, one will refer to memory the other I/O, depending on the direction of data transfer.

2.0 Hardware (Continued)

The tranfer is now complete so $\overline{BRQ}$, AEN and $\overline{DACK}$ are all driven inactive. AEN going inactive causes the address latches to swop back over; the microprocessor latch outputs are driven, the DMA latch outputs are TRI-STATE. When the microprocessor detects that $\overline{BRQ}$ is inactive it deasserts $\overline{BACK}$ and retakes control of the bus. The microprocessor then clecks flags in the EASI to determine the success, or otherwise, of the transfer.

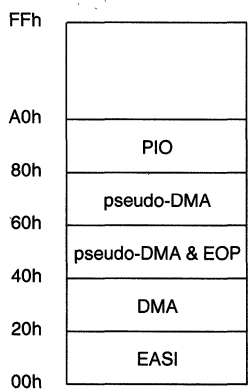
The other form of DMA available is pseudo-DMA; that is the EASI 'thinks' it is a proper DMA transfer but the processor is still in control of the bus. To do this the microprocessor must set the DMA to mask off any DRQ, initialize the EASI for single mode DMA and monitor the EASI flags for DRQ going active. At this point the processor must generate an $\overline{IOR}$ or an $\overline{IOW}$ and a $\overline{DACK}$, to properly simulate DMA. This can be done by causing an I/O read or write at a certain address, to generate a $\overline{DACK}$, and if wanted an $\overline{EOP}$. The particular address does not matter since during DMA the EASI ignores the address bus.

The following equations generate pseudo-DMA signals, while allowing the true DMA signals to operate normally:

```
/DACKOZ = /IORZ * BACKZ * /A7 * A6
        + /IOWZ * BACKZ * /A7 * A6
        + /DACKIZ

/EOP2 = /IORZ * BACKZ * /A7 * A6 * /A5
      + /IOWZ * BACKZ * /A7 * A6 * /A5
      + /EOP1 * EREADY
```

DACKIZ is the $\overline{DACK}$ from the DMA, which the output normally follows. The rest of the equation generates the pseudo-DMA $\overline{DACK}$.

$\overline{EOP2}$ is the input to the EASI, $\overline{EOP1}$ is the output from the DMA. The first two lines of this equation generate an $\overline{EOP}$ for the pseudo-DMA cycle, on the lower half of the address space that also generates the pseudo DACK. This means the final I/O map now is as follows:



The third part of the $\overline{EOP1}$ equation is to overcome a problem that occurs during a block mode DMA transfer. On the last byte the READY signal from the EASI (this will be called EREADY to distinguish it from the READY into the DMA, DREADY) may be driven low to freeze the transfer. However $\overline{DACK}$, $\overline{EOP}$ and $\overline{IOR}$ or $\overline{IOW}$ will all be active causing a valid EOP condition. This is not a problem until we consider the next equation.

```
/DREADY = /EREADY * /INT * /DACKIZ
```

This equation causes the DMA READY input to go high if any of EREADY, SCSI interrupt INT or DACKIZ go high. This overcomes a potential bus lockup, caused by an error occurring during a block mode DMA transfer. On a phase or parity error the EASI will stop the transfer and generate an interrupt. If DREADY is left inactive the DMA will keep control of the microprocessor bus. This equation holds DREADY high on interrupt, allowing the DMA to pass control of the bus back to the microprocessor.

Although this equation prevents an error it also introduces a problem in the $\overline{EOP2}$ equation. As explained previously, during the last byte of a block mode transfer although READY is low a valid EOP condition may exist, which will cause an interrupt. This interrupt will then cause DREADY to be driven high, allowing the DMA to finish the transfer, but lose the last byte since the EASI is not ready. To overcome this the $\overline{EOP2}$ equation gates $\overline{EOP1}$ with EREADY, so the EASI does not see a valid EOP condition until it is able to transfer the last byte. **This problem does not occur in MODE E since it generates a true end of DMA interrupt.**

Another problem is introduced by the DREADY equation. If an error occurs during a DMA transfer the EASI will generate an interrupt and DREADY will be forced high. This allows the DMA to 'run free', writing garbage into MEMORY, and wasting SCSI bus time. To prevent this an external $\overline{EOP}$ must be applied to the DMA on error. The next equation does this:

```
/EOP1 = /DACKIZ * /EREADY * INT
EOP1.TRST = /DACKIZ * /EREADY * INT
```

$\overline{EOP1}$ is an I/O pin which normally acts as input from the DMA, so the output is TRI-STATE unless DACKIZ and EREADY are low, with INT active. When this error condition occurs an $\overline{EOP}$ is output to the DMA, terminating the data transfer. A prerequisite of this equation working properly is the existence of the DREADY equation, since an externally applied $\overline{EOP}$ will have no effect on the DMA if READY is held low.

The next equation is also required to prevent a fault occurring during a block mode DMA transfer:

```
/EASIWRZ = /IOWZ * BACKZ
         + /IOWZ * EREADY * /BACKZ
```

When the processor is in control of the bus a straightforward $\overline{IOW}$ can write to the EASI, but if the DMA is in control EREADY must be high to allow the write. The inclusion of EREADY is required because during DMA the EASI is a flowthrough latch; an $\overline{IOW}$ passes the data from the processor bus onto the SCSI bus. Therefore if the DMA reaches its next byte before the data on the SCSI bus has been transferred, the DMA will overwrite the data. To prevent this $\overline{EASIWR}$ can only be allowed when EREADY is high.

The final PAL output is used to invert the SCSI interrupt signal, to make this active high output compatible with the processor's active low input.

```
/SCSIINTZ = INT
```

3.0 Diagnostic Software

3.1 INTRODUCTION

The diagnostic software resides at the base of RAM, with the purpose of checking and initializing the printer controller board after power up and every hard reset. It must be at location zero, since after any hardware reset the program counter is cleared. The interrupt service routine is included here, since it must exist at an exact location in memory.

3.0 Diagnostic Software (Continued)

The software uses the NSC800 instruction set (fully Z80® compatible) which makes the required low level board operations faster, and allows the exact positioning of code in memory. The assembler and linking loader are from Microtec Research.

3.2 DEVICE CHECKING AND INITIALIZATION

This section will refer to the diagnostic software PRINTER.SRC and the files PRNSYM.SRC and EASISYM.SRC which contain the constants used. A full listing of all programs described in this document are available on floppy disk.

3.2.1 Memory Check and Interrupt Servicing

An 'org' statement can be used to position this program at the base of ROM, address 0. After a jump to the ROM check, the next byte stores a label defining which version of the software is installed. On an EPROM the version number can be read from address 0002h. The upper and lower nibbles should be considered as two numbers i.e., '10' defines version 1.0.

The interrupts are disabled, since the service routines are not initialized, and the EPROM is checked; simply consisting of reading the same address twice and ensuring the same value returns both times. If this test fails the system halts since a drastic error must have occurred. If the EPROM passes the test the program jumps to the RAM check.

On interrupt the NSC800 stops before its next instruction, pushes the program counter onto the stack and loads it with a new address, depending on the highest priority interrupt channel active. Interrupt A, the only channel used, causes a jump to location 003Ch. The interrupt service routine at this location pushes all of the processor registers onto the stack, and tests for a SCSI reset. SCSI reset must be checked, as this is a special condition, causing the board to be reset by a general restart routine (this routine is in EASIO.SRC and explained in section 3.5).

If the interrupt is not a SCSI reset the software makes a call to the bottom of RAM, where a jump command should be followed by a public variable RESETA. This variable can be loaded with the starting address of the routine to service the interrupt. On return from this routine the processor registers are popped back off the stack, and program control returns from the interrupt. Since RESETA is public any external software may use it, and therefore control which routine services an interrupt. RAM must be verified before this jump table is set up.

The RAM test simply checks each BIT can contain a zero and a one, then clears every byte. Once RAM has been verified the stack pointer may be initialized, allowing call statements to be included. It should be remembered that although the interrupt routine is between the ROM and RAM tests, when the program runs it will jump straight from the ROM test to the RAM test. Since the interrupt jump table is in volatile RAM the code for a jump instruction must be written into RAM after every reset.

3.2.2 PIO Initialization

The file PRNSYM.SRC contains the port addresses for all of the I/O devices. The upper nibble contains the location the device takes within the I/O map, the lower nibble selects the register within the device to be accessed.

The PIO has five types of registers which control data transfers through the device:

1. **Data Register**—Each port has an eight bit data register containing the data passed between the PIO and the processor. These are either read or write registers, depending on whether the port bits are inputs or outputs.
2. **Data Direction Registers**—Each port also has a data direction register which controls whether each of the 20 bits is an input or an output (an input is defined by a zero, an output by a one).
3. **Mode Register**—There is one three bit mode register which selects which of the four modes the device is in. This board will always require the PIO to be in Mode 0 which is the basic I/O mode. The alternatives use Port C for handshaking.
4. **Bit Clear Register**—Each port has an eight bit, bit clear register which clears any output bit whose corresponding bit in this register is high.
5. **Bit Set Register**—These are similar to the above, but allow the output bits to be set.

The PIO is set for Mode 0, with Ports A and C outputs and Port B an input. Port A drives the printer data bus, while Ports B and C read and write the printer control signals. The four printer outputs, to the Port B input, are $\overline{\text{ACK}}$, BUSY, PE and $\overline{\text{ERROR}}$. $\overline{\text{ERROR}}$ goes low when there is no paper, the printer is off-line or an error occurs; PE goes high when the printer is out of paper; BUSY goes high to show the printer can accept data; $\overline{\text{ACK}}$ pulses low to acknowledge data after a byte has been transferred. Since BUSY and $\overline{\text{ACK}}$ are both handshake signals the user must decide which to use. This software will use BUSY.

The top three bytes of Port B are used to read in the SCSIID from a block of switches. The SCSIID is the boards identifier on the SCSI bus, used in selection and arbitration, consisting of a single active bit. This is read in as a three bit number, converted to the correct bit pattern and stored in a public variable called SCSIID.

Port C is an output, driving the printer signals $\overline{\text{INIT}}$ and $\overline{\text{STROBE}}$. A low pulse on $\overline{\text{INIT}}$ of 50 μs causes the printer to be initialized; a 500 ns low pulse on $\overline{\text{STROBE}}$ causes the printer to read the data on the bus. The highest bit of Port C drives a LED, which is on during board operation, and can display an error message by occulting a fixed code. Errors are displayed by the routine *ERROR,* a public function which displays the error number as a four bit code. It does this by occulting the LED for 1 second to show a 1, ½ a second for a 0.

Since none of the PIO registers are true read/write the device cannot be tested, only initialized.

3.2.3 DMA Test and Initialization

The DMA consists of address, word-count and control registers for four channels, of which only one is used. There follows a description of the registers used, with the addresses shown in PRNSYM.SRC.

1. **Word-Count Registers**—There are two word-count registers; a sixteen bit write only base word-count register and a sixteen bit read only current word-count register. As with all sixteen bit registers in this device the two bytes of data are accessed by two successive selections of the same address. An internal flip-flop determines which byte is read or written, with the lower byte selected first after a reset. To transfer the correct number of bytes the word-count register must be written with the number of bytes to be transferred minus one.

3.0 Diagnostic Software (Continued)

2. **Address Registers**—The address registers are also sixteen bit, and called base and current. The base address register is written with the address the transfer must start at, while the current address register contains the next address to be written or read. *The DMA is tested by writing a value to the base address and reading it back from the current address.*
3. **Control Registers**—There are three types of control register which will be discussed only in the way they are used by this software. The master clear register (DMAMCL) is the software equivalent of a hardware reset; all registers are cleared and DRQ is masked off. Masking of DRQ is controlled by the parallel mask register (DMAMSK), in this case always used to allow DRQ on Channel 0 and mask off the others. The final type of register used is the mode register (DMAMOD) which controls whether a transfer is block or single mode, memory read or memory write. Each of the four channels has a six bit mode register, all at the same address, with the bottom two bits of the data bus determining which is selected.

The DMA test program initializes the device for a one byte, block mode, memory write, to an address TSTBYT. The DMA is initialized in this particular manner for use in the EASI loopback DMA test routine.

An error in DMA test will cause error signal 0.

3.3 EASI TEST AND INITIALIZATION

3.3.1 Introduction

This document will show how the registers within the DP5380 and DP8490 are used in an application. For a full register description refer to the DP5380 and DP8490 datasheets. Register names and their locations are given in PRNSYM.SRC.

3.3.2 DP8490 and DP5380 Test

The DP5380 and DP8490 are completely pin and hardware compatible, and also software compatible until the enhanced mode bit is set in the DP8490. The DP8490 powers up in normal mode (MODE N which is software compatible with the 5380) and is in MODE N after any chip reset. The initial test is therefore the same for both devices, writing to the Mode Register 2 (EASIMR2) and reading back the data. The EASIMR2 is selected since it has the most read/write bits that do not directly affect the bus.

An error in EASI test will cause error signal 1.

For the DP5380 testing is now complete, but the DP8490 enhanced mode offers a loopback test facility, where the SCSI drivers are disabled and the SCSI I/O's looped back inside the EASI. **Using this feature the user can fully check the device, and by doing a DMA transfer fully check the board.**

At this stage it is therefore neccessary to determine which device is inserted. In the DP5380 bit 6 of the Initiator Command Register (EASIICR) selects the 'test mode', which disables all output drivers on the device, making it invisible to the system. Although the device can still be written to no data can be read back, making applications very limited. In the DP8490 this bit selects the enhanced mode (MODE E) of the device.

In MODE E addresses 0–6 access the same registers as in MODE N, but address 7 is different. Instead of accessing the Reset Parity/Interrupt (EASIRPI) and Start DMA Initiator receive (EASISDI) registers, address 7 directly accesses the Enhanced Mode Register (EASIEMR) and indirectly accesses the Interrupt Mask Register (EASIIMR) and Interrupt Status Register (EASIISR). The only other difference to registers occurs in the Target Command Register (EASITCR) where one of the previously unused bits becomes a flag (explained in section 3.3.4).

To test which device is inserted EASI test sets the enhanced/test mode bit, writes data to address 7 and reads back from the same address. If the device is a DP5380 it will be in 'test mode' and the data read will be 0FFh due to the pull-up resistors on the data bus. If it is a DP8490 the data read back will be the data written, providing the only bits set are read/write. For a DP5380 the program jumps to initialization for selection; if it is a DP8490 loopback testing follows.

3.3.3 DP5380 Initialization for Selection

The DP5380 initialization involves programming the device to respond to a selection. First, the public variable DP8490 (which other programs should treat as a constant) is set FALSE to indicate that a DP8490 is not inserted. For the DP5380 to be selected it must have the SCSIID of the device in its Select Enable Register (EASISER) and parity must be enabled. The processor must enable its interrupts and set up the jump table; *intA* is an external routine which sets the processor's interrupt mask to only allow interrupts on A, and enables the interrupts (see EASIO.SRC); *main__* is an external program which responds to a SCSI selection (further explained in the Run-time software section).

If BSY is inactive for a bus settle delay (400 ns), SEL is true and the bit on the data bus corresponding to the SCSIID is active, a SCSI selection interrupt is generated. This causes a processor interrupt, taking the program from the continuous 'jr Now' instruction to the interrupt service routine, and through the jump table to *main__*. When *main__* has finished servicing the interrupt it will return to the interrupt servicing routine and then return from the interrupt. Thus the board's 'idle' state is to continually execute the 'jr Now' instruction.

3.3.4 DP8490 Loopback Test

The loopback test mode of the DP8490 allows all signals to be fully tested, including a DMA transfer, without affecting the SCSI bus. To allow this DMA transfer loopback allows the user to drive both initiator and target signals simultaneously.

After setting the DP8490 flag true all initiator signals, and then target signals, are asserted and checked. This includes a check on the data bus by writing a test value to it and reading it back. The data bus test value has an odd number of bits, and since the specified SCSI parity is ODD, the parity bit must be inactive. One of the new features offered in MODE E is programmable SCSI parity, which is tested by ensuring the parity bit becomes active when EVEN parity is enabled, and inactive when a test value with an even number of active bits is written. Parity is then returned ODD, and the parity bit should become active.

An error in Loopback testing causes error signal 2.

The next test in loopback mode is a DMA target receive transfer to a location in memory, TSTBYT. This not only tests the EASI, but also the interrupt servicing, the DMA and memory. Although the device is in loopback the software must carry out the transfer as it would normally i.e., the bus phase must be correct, $\overline{\text{BSY}}$ must be active and the SCSI bus must be asserted. The EASIMR2 must be properly set

3.0 Diagnostic Software (Continued)

up for block mode DMA, with interrupt on $\overline{EOP}$ or parity error. Since this test is interrupt driven the interrupt jump table must be loaded with the address of the routine which will service the DMA loopback test, and the interrupts must be enabled. The DMA has already been initialized for this.

Before enabling interrupts the SCSI interrupts should be reset, which in MODE N would involve reading address 7. Since the EASIEMR is now at this address the resetting of interrupts, and the start of DMA initiator receive are initiated by writing to function bits of the EASIEMR.

When the Start DMA Target receive (EASISDT) register is written the EASI will issue a $\overline{REQ}$ to show it is ready to receive data on the SCSI bus. At this point the device at the other end of the bus would normally assert $\overline{ACK}$ to show it has data available. In this case there is no other device so the user must wait for $\overline{REQ}$ to go active and then assert $\overline{ACK}$. Thus both initiator and target signals must be asserted simultaneously. The user waits for $\overline{REQ}$ to go inactive, and deasserts $\overline{ACK}$ to show the bus transfer is complete.

The program then goes into a continuous loop awaiting a SCSI interrupt. This interrupt will occur because the EASI will have issued a DRQ, when $\overline{ACK}$ went active, and the DMA wil have transferred the last test byte written to the EASI to TSTBYT, finishing with an $\overline{EOP}$. On interrupt the program will jump to address 003Ch, where all the processor registers are pushed onto the stack, a call is made to the base of RAM, and from there it will jump to the subroutine *DIAGA*.

One of the problems with DMA in a DP5380 is that end of DMA is flagged when $\overline{DACK}$, $\overline{EOP}$ and $\overline{IOR}$ or $\overline{IOW}$ are simultaneously active; although the data may not yet have been transferred on the SCSI bus. To overcome this the software must examine the SCSI handshake signals $\overline{REQ}$ and $\overline{ACK}$, both of which must be inactive on three successive samples for a true end of DMA. This is more fully explained in the DP5380 and DP8490 datasheets. **MODE E of the DP8490 detects true end of DMA, after $\overline{ACK}$ goes inactive, before generating an interrupt.**

DIAGA responds to the interrupt after the loopback DMA by checking all the correct flags have been set. The DP5380 only uses four bits of EASITCR so MODE E uses the free bit 7 as a flag, to show true end of DMA. This is the first flag checked. Address 7 not only directly addresses the EASIEMR it also indirectly addresses the EASIIMR and EASIISR. After writing the correct code to the function bits of the EASIEMR the next access of address 7 will be to the EASIISR, if it is a read, or the EASIIMR, if it is a write. **The advantage of the EASIISR over the DP5380 registers is that all interrupt information is available in one register, and every interrupt is flagged.** To check DMA the user need only read the EASIISR and ensure that the only flag active is end of DMA. The user should note that SCSI reset causes the device to revert to MODE N, from which the EASIISR can not be read, so is not flagged.

The final EASI flag test is the 'conventional' end of DMA flag in the Bus and Status Register (EASIBSR). This flag, the interrupt flag and the flag to show no phase mismatch has occurred must be the only bits active. The final loopback DMA test is to ensure that memory location TSTBYT contains the correct data.

An error in Loopback DMA test causes error signal 3.

3.3.5 DP8490 Initialization for Selection

The DP8490 initialization is very similar to that of the 5380, even calling the same routine, *main__*, to respond to selection. However, this device stays in MODE E and uses these enhancements to only allow interrupts on selection and parity by setting the EASIIMR. Selection and parity are the only valid interrupts at this point.

At the end of the DIAGA routine program control returns to the 'jr Here' instruction, which it will continually enact until a parity or selection interrupt causes a jump to *main__*.

3.4 ERROR HANDLING

Throughout all software for this board the error handling is the same for errors considered non-recoverable, which includes all errors in diagnostics. On error the board continually displays an error number, as a four bit binary code, using the LED. The subroutine *ERROR* carries out this function.

On error register 'I' should be loaded with the error number and routine *ERROR* called. This routine then occults the LED for ½ second to display a zero, 1 second to display a one, with the LED on for ½ second between flashes. The four bits, most significant bit first, are repeatedly displayed between 2 second intervals, during which time the LED is on. The timing delays are generated using a routine *DELAY*, which gives a number of ¼ second delays, the number of delays being determined by the value in the 'I' register.

The following list shows the possible errors in diagnostics. The run-time software section contains a similar listing of its error codes.

Error 0
The DMA can not be accessed.

Error 1
The EASI can not be accessed.

All other diagnostic errors concern a DP8490

Error 2
An error has occurred in the assertion of SCSI signals in loopback test mode.

Error 3
An error has occurred during the loopback DMA transfer.

3.5 EASIO.SRC

This file contains public assembly language routines, which can be called by either the diagnostics or run-time software, to implement low level commands. As these routines may need to be called by routines written in 'C' any variables passed to the routines are passed in the 'hl' register pair, then in 'de', then 'bc' and then on the stack. *ERROR* and *DELAY* both use the 'I' register to pass a variable. Data passed back to the calling routine is returned in the accumulator.

Functions *'read'* and *'write'* implement general purpose I/O register accesses, while *'dmaread'* and *'dmawrite'* handle the special case of the 16-bit DMA registers. These require two accesses of the same address. *'intA'* initializes the processor interrupt mask, with a 'pseudo' I/O write, which selects a register internal to the processor, setting the mask to only allow interrupts on RSTA. This function also enables the interrupts, which can be done by *'eni'*, with *'dsi'* disabling interrupts.

'IDtest' is used by the run-time software, during selection, to read the number of bits active on the bus. This routine checks the number of high bits in the byte passed in the 'I' register, returning the value in the accumulator.

3.0 Diagnostic Software (Continued)

Function *'restrt'* is a general purpose reset routine, which can be called on an error condition. This causes every device on the board to be reset, and the diagnostics to be re-run, thus clearing all memory and reinitializing the stack.

4.0 Run-Time Software

4.1 INTRODUCTION

Following the diagnostic software must be a program which will control all SCSI bus transfers, beginning with selection. This program is written in 'C', using a PARAGON 'C' cross compiler for an NSC800. The relevant files for this section are SCSI.C, COMMAND.LIB, PROCESS.LIB, DMA.LIB, PRINT.LIB, ARBITRAT.LIB, SYM.H, CONST.H and COMMANDS.H, all of which are on the supplied floppy disk.

SCSI.C contains the main program, called *'main'* by 'C' and *'main_'* by assembler. The '.LIB' files contain the functions called by *main* and the '.H' files contain the constant values used by all of these files.

4.2 MANDATORY PHASES

This section outlines all the bus phases and transfers which a target must support. It would be perfectly legal for a target to respond to a selection, fetch a command block from the initiator, and then returns status and message bytes before releasing the bus. Although no process would be actuated this would be a legal sucession of events.

4.2.1 Selection Response

main() is the program jumped to when an interrupt is generated as the program circles the continuous loop at the end of diagnostics. This routine should only be entered after a selection interrupt, if any other interrupt is active it is considered an error. The function *select()* checks an interrupt to ensure it is valid.

The type of device installed is checked by reading the 'DP8490' flag, and this determines how the interrupt is verified. In a DP5380 a selection interrupt is determined by the absence of any other interrupt, with $\overline{SEL}$ active. The user must check the EASIBSR to ensure that no error flags are active, only the INT and PHSM (phase match) bits.

Any other flag will cause error number 4 to be displayed.

After reading the EASIBSR and finding no interrupt flags the user knows the interrupt should be a selection. The only other unflagged interrupt is a reset, which would have been handled by the low level interrupt service routine. Therefore the EASICSB register must also be read, and if $\overline{SEL}$ is inactive an error condition exists.

This causes error number 5 to be displayed.

In MODE E the user need only read the EASIISR to determine which interrupt is active, including selection. All interrupts, other than parity and selection, should be masked off, so any error concerns only these two flags.

A SCSI parity error is displayed as error number 8, if the selection flag is not active error number 9 is displayed.

The common tests for MODES E and N both concern the EASICSD; error 6 shows that the correct SCSIID bit was not active on the bus; error 7 shows there was more than two bits active on the bus. During selection an initiator is only allowed to assert two bits on the bus, its own ID and the target ID.

The target must assert $\overline{BSY}$ to show it has recognized the selection, then when the initiator deasserts $\overline{SEL}$ the selection phase is complete.

4.2.2 Command Phase

A selection phase is followed by a command phase, where the target reads a command block from the initiator. This command block specifies the actions the initiator requires the target to execute, plus the length of any data transfers requested. This board only allows six byte command blocks, which are transferred into the target by function *fetch_cmd()*.

The command block is transferred using programmed I/O; that is each byte is individually handshaken under processor control. The bus phase must be Command Out (out and in always refer to the initiator, so Command Out is a command block sent by the initiator), bus phase being set in the EASITCR. To transfer a byte (see *Figure 4.1*) the user must assert $\overline{REQ}$, then wait for $\overline{ACK}$ to go active to show data is available. On $\overline{ACK}$ the target can read the data on the bus, then deassert $\overline{REQ}$ to show it has received the data. When the initiator deasserts $\overline{ACK}$ the byte transfer is complete.

4.2.3 Status Phase

The target must send a status byte to the initiator during the status phase, at the termination of each command. A list of status codes is given in COMMANDS.H. *status()* is the function which enters a Status In phase, and transfers the code.

The EASITCR must be written with the Status In phase and the EASIODR with the status code. This code should be asserted onto the bus, remembering to keep $\overline{BSY}$ asserted, and if neccessary the MODE E bit. $\overline{REQ}$ is then asserted to show data is available. The initiator should assert $\overline{ACK}$ when it has read the data, allowing the target to deassert $\overline{REQ}$ and take the data off the bus. The transfer is complete when the initiator deasserts $\overline{ACK}$ (See *Figure 4.2*).

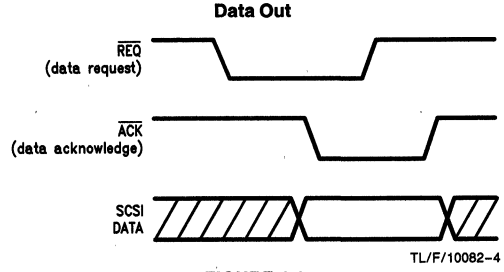


FIGURE 4.1

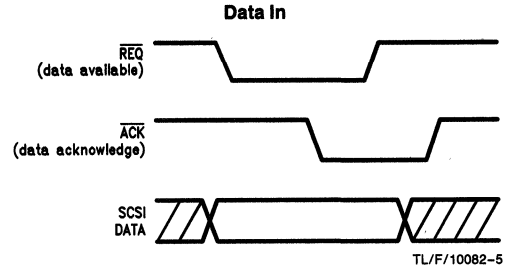


FIGURE 4.2

4.0 Run-Time Software (Continued)

4.2.4 Message Phases

A selection must be terminated by the target entering the Message In phase, and sending a relevant message code (as listed in COMMANDS.H). The target follows the Status phase with the Message In phase, usually to send the COMMAND COMPLETE message. This indicates valid status exists, so the target can release $\overline{BSY}$ to free the bus. *messin()* enters the Message In phase and sends the message using programmed I/O, as in *status()*.

The only message which must be supported is COMMAND COMPLETE. If a device on the bus supports other messages it indicates this by responding to or asserting $\overline{ATN}$. An initiator asserts $\overline{ATN}$ if it has a message for the target. It is common for an initiator to assert $\overline{ATN}$ during selection and, if the target responds by entering a Message Out phase, it sends the IDENTIFY message. This establishes whether the target can respond to a greater set of messages, and whether the initiator supports disconnection. It shows this by setting bit 6 of the message. In a disk controller IDENTIFY would also establish the path by including a Logical Unit Number (LUN), but the printer controller uses all LUN's at this SCSIID.

getmes() enters a Message Out phase and fetches a message using programmed I/O. Although this software only supports single byte messages it must be prepared to accept messages from the initiator of up to the maximum length, 256 bytes. If an initiator wishes to send further bytes after the first it must keep $\overline{ATN}$ asserted, only deasserting after the final byte has been transferred. *getmes()* will handshake up to 256 bytes, after which if $\overline{ATN}$ is asserted it will be considered an error. Only the first byte is used in determining the message sent.

The use of messages during disconnection, arbitration, reselection and in error handling will be discussed in those sections of this document.

4.2.5 Command Termination

After sending status and message codes to the initiator, the target should then release $\overline{BSY}$ to free the SCSI bus. However, before allowing a bus free phase the target must initialize itself for the next selection, as in function *reset()*.

The interrupt jump table is loaded with *main()* and the EASISER with the SCSIID. For a MODE E device the interrupt mask is set. For both types of device the interrupts must be reset, and $\overline{BSY}$ deasserted. Interrupts are disabled at the start of this function, so must be enabled after calling *reset()*.

4.3 COMMAND PROCESSING

After the command block has been fetched the command has to be determined and executed. The printer command set is shown in COMMANDS.H.

process_cmd() reads the first byte of the command block which contains the operation code. This determines what actions are taken. If a command has been sent which this software does not recognize, a status of CHECK CONDITION is returned, with sense set to ILLEGAL REQUEST.

4.3.1 Test Unit Ready

TEST UNIT READY will be sent by an initiator before a print to ensure the printer is on, on-line, has paper and is not in an error condition. On this command the function *ck_printer()* is called to read the printer signals through Port B of the PIO and check for errors.

If the error line is not high the printer is operational, so the status is GOOD, and the sense is set to NO SENSE (no error). If there is an error the paper error line must be checked. If this is low (active low input) the printer is out of paper and status of CHECK CONDITION is returned, with sense set to MEDIUM ERROR. If it is not a paper error the printer is assumed to be off or off-line, so status is CHECK CONDITION with sense UNIT ATTENTION.

4.3.2 Request Sense

An initiator will send a REQUEST SENSE command after the target has returned a status of CHECK CONDITION. The sense data is sent to the initiator in an effort to understand an error condition, and if possible recover from it. Byte 4 of this command block contains the length of an extended sense message, which must not be greater than 4, or sense is set to ILLEGAL REQUEST and CHECK CONDITION status returned. This software only supports four bytes of sense data.

Sense data is sent to the initiator using single mode DMA. The DMA is initialized by *send_sense()*; it is reset by a master clear, the mode set, the DMA mask written and the address and word-count loaded. The function *single_dma_in()* (explained in section 4.4.1) enters the correct phase (Data In) and transfers the data.

4.3.3 Reserve Unit

In a multi-initiator system a printer controller must be reserved before a print commences, or the data from two different initiators may be mixed. This command stores the initiator's ID in a variable called 'reserved', and on subsequent selections only this initiator may execute commands.

Any initiator wishing to reserve the unit must put its own ID on the bus during selection, along with the target's ID, so this software knows which initiator is reserving the unit. If the initiator's ID is not on the bus it can not reserve the unit, and since this is a prerequisite of printing, it cannot use the printer.

If an initiator attempts to reserve this unit without making its ID available sense is set to ILLEGAL REQUEST and status of CHECK CONDITION returned. If another device attempts to reserve the board when it is already reserved status returned is RESERVATION CONFLICT.

4.3.4 Release Unit

This is the reciprocal command to the previous, freeing the printer for other initiators after a print has been completed. If an initiator other than the reserver attempts this command a status of RESERVATION CONFLICT is returned; if this is attempted when there is no reservation current the returned status is CHECK CONDITION with sense set to ILLEGAL REQUEST.

4.0 Run-Time Software (Continued)

4.3.5 Print

Since transferring data to a printer is a very slow process, typical Epson Fx range 80 cps–160 cps, the print command transfers the data to a buffer, leaving it to be printed later. Bytes 2, 3 and 4 of the command block contain the data length, byte 2 the most significant. The use of three bytes to define the size means that in theory block transfers of up to 16 MB are allowed. This software could support blocks of up to 64 kB, bytes 3 and 4 are read, but blocks are limited to a size BUFFLIM, which will be determined in section 4.4.3.

Possible errors in a print occur when the unit is not reserved, the data length is set to zero, or the block size is too large. The block size is too large if byte 2 of the command block is active or if the data length is greater than BUFFLIM. In response to an error the transfer is cancelled and status of CHECK CONDITION is returned with sense set to ILLEGAL REQUEST.

The data is not transferred unless sense is set to NO SENSE. If an error has occurred in the printer sense will have been set to indicate the error source. The error condition must be rectified and TEST UNIT READY sent to reset the sense data.

print_cmd( ) is the function which transfers data into a buffer using block mode DMA. The print buffer is a circular queue, allowing the user to take data off the front and put data on the rear (see *Figure 4.3*).

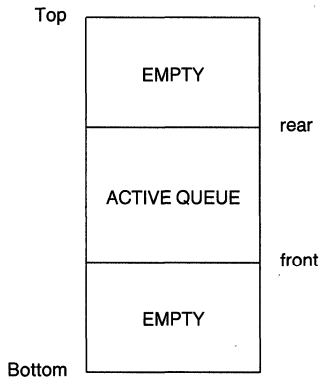


FIGURE 4.3

front points to the next byte to be printed, rear points to the next available byte for entering data. When front equals rear the queue is empty, when rear is one less than front the queue is full.

The buffer limits are called top and bottom. When either the front or rear pointers reach the top the next increment takes them to the bottom. Thus the queue may look like *Figure 4.4.*

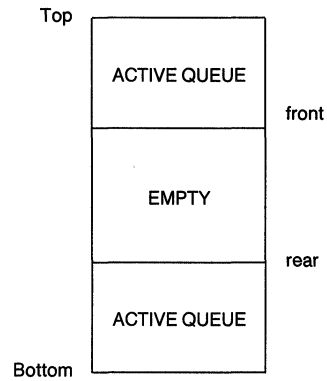


FIGURE 4.4

When *print_cmd( )* is first called after a reset it must set up the queue, defining top and bottom, and setting front equal to rear equal to bottom. Before any transfer the data length must be checked to ensure that the DMA will not take rear past top. If it would the data must be transferred in two blocks; one to the top of the queue, and one starting at the bottom of the queue. This routine should not be called unless there is sufficient free space for the size of transfer. Function *dma_data( )* sets up the EASI and DMA for a block mode transfer, enters the Data Out phase and calls *dma( )* to handle the transfer. This will be more fully explained in section 4.4.

4.3.6 Flush Buffer

The purpose of this command is to allow an initiator to terminate an unwanted print. An initialization pulse is sent to the printer, in case it has its own buffer, and the SPC buffer is cleared. This is done by setting front equal to rear equal to bottom.

4.4 DATA TRANSFERS

This section is concerted with the way in which software controls DMA transfers, both block and single mode, and how the print data block length is determined.

4.4.1 Single Mode DMA

The four bytes of sense data are sent to the initiator using single mode DMA. *send_sense( )* sets up the DMA to transfer four bytes of data from the sense buffer to the EASI, leaving function *single_dma_in( )* to complete the transfer.

The phase, which in this case is Data In, is determined by the calling function, and the EASIMR2 initialized for a single mode transfer. Parity checking is enabled with interrupt on $\overline{\text{EOP}}$ or parity error. The data bus is asserted and the routine *dma( )* is called to handle the transfer.

Since both single and block mode DMA transfers, either in or out of the initiator, have large portions of code that are common, a function can be written for general purpose DMA handling. This function, *dma( )*, sets up the interrupt response and makes a write to the register which initiates the required type of transfer, which could be target receive or target send etc. *dma( )* checks the success of the transfer when the transfer is complete.

4.0 Run-Time Software (Continued)

In MODE E the interrupt mask can be set to only allow parity, $\overline{EOP}$ and DMA phase mismatch interrupts. The interrupt jump table is loaded with the starting address of a service routine, which sets a flag to show an interrupt has occurred. The program can then sit in a loop waiting for this flag to go active.

After the interrupt the cause has to be checked to ensure the transfer was successful, but this checking depends on the device installed. For a DP8490 the user can read the EASIISR and if any flag other than end of DMA is active an error has occurred. On such an error, status is set to CHECK CONDITION with a sense of ABORTED COMMAND. The interrupt mask can be reset, masking off end of DMA and DMA phase mismatch, and the DMA mode in the EASIMR2 disabled. The interrupt service routine is set to jump to the general interrupt handler *gen_int()*. The interrupt must then be reset and the processor's interrupts enabled.

In MODE E true end of DMA is detected, but in MODE N the end of DMA interrupt is generated when $\overline{EOP}$, $\overline{DACK}$ and $\overline{IOR}$ or $\overline{IOW}$ are active concurrently. True end of DMA, after the transfer is complete, must be detected by software. $\overline{REQ}$ and $\overline{ACK}$ must both be inactive on three successive samples for true end of DMA. **This additional code makes the MODE N DMA routine slower**. After detecting this the user can check the end of DMA flag is active in the EASIBSR, giving status CHECK CONDITION and setting sense ABORTED COMMAND if it is not. The DMA bit is then reset and *gen_int()* called to ensure there was no parity or phase errors, and to reset the interrupt service routine.

On return to the function *single_dma_in()* it will deassert the SCSI bus and return to the calling routine.

4.4.2 Block Mode DMA

blk_dma_in() is the equivalent routine to *single_dma_in()* initiating a block mode transfer to the initiator. *send_sense()* could call this routine, after setting the DMA for block mode, and transfer the data using block mode, with no difference to the other software. The only difference between these two routines is the setting of the EASIMR2 BLK mode bit.

print_cmd() is used to transfer the data to be printed from the initiator. This function checks that the transfer will not exceed the constraints of the queue and calls *dma_data()* to initiate a block mode transfer. *dma_data()* sets the EASI and DMA registers for a block mode target receive, with the data size set by *print_cmd()* and the destination starting address equal to rear. *dma()* is again used to handle the interrupts.

4.4.3 Determination of Data Block Length

The block of data to be transferred for a print has been specified as a maximum length of BUFFLIM, but the value of this constant has to be evaluated. The two main considerations in this are the latencies of the bus and the SPC.

Since this board is a printer controller it will be of a very low priority. Thus the device should not hold the bus for too great a time per transfer, as this would slow down initiator's accesses of a high priority peripheral. The second consideration is the time that the target takes from selection to entering a data phase, the data block must take longer than this to transfer, or the command is inefficient. Measurements taken on various models of personal computers showed that, with an Advanced Storage Concepts ASC-88™ SCSI Host Adaptor, the time taken from the initiator asserting $\overline{SEL}$ to the target commencing the data transfer is approximately 3 ms.

The block mode DMA rate was measured as between 200 kB/s and 500 kB/s limited by the DMA in the PC. A block length of 2 kB was selected, since this would take between 4 ms and 10 ms to transfer. Thus the data transfer time is greater than the selection to data phase time, but the overall time on the bus in not too long.

4.5 PRINTING DATA

Due to the inherent slowness of printers data can not be printed while the controller is in command of the SCSI bus. To prevent tying up the bus data is stored in a buffer and printed after $\overline{BSY}$ has been released. The function *printit()* handles the transfer of data to the printer.

printit() checks the printer is not in an error condition, and transfers the byte of data at address front to the printer. The data is transferred through Port A of the PIO with BUSY and $\overline{STROBE}$ used to handshake the data. BUSY must be high and $\overline{STROBE}$ pulsed low a minimum of 0.5 μs for the printer to accept data. front is incremented, to point to the next byte to be transferred. To print out the queue *printit()* is called until front equals rear.

If the function does detect an error it sets sense to the appropriate value, and implements any outstanding reconnection. The printer is continually polled to determine when it comes back on-line, at which time the print continues.

While the board is printing it must still be available for selection. An initiator can reset every device on the SCSI bus if a target does not respond to a selection within a Selection Timeout, normally 250 ms. The interrupt service routine has been set to jump to *main()*, but the program is currently in *main()*, so this function must be re-entrant.

main() is written in such a way that it will only process commands if it is unreserved and not printing, or reserved by the current initiator but not printing. If the board is reserved, but not by the current initiator, a status of RESERVATION CONFLICT is returned. If the board is printing, and the reserving initiator attempts to send it another command a status of BUSY can be returned.

However, the controller has a 30 kB print buffer, of which only 2 kB would be used at a time. It would be much more efficient to continue executing commands until the buffer is full. One method of achieving this can be seen in *outbuf()*. In this function a flag, called next, is set if the free space in the queue is greater than BUFFLIM. If a selection occurs while a print is on, this flag is checked and the command processed if it is active. If it is inactive a status of BUSY is returned. This method has the disadvantage that the initiator must continually poll the SPC to determine when it is ready to accept data. This 'loads' the SCSI bus and slows down the print, since the SPC will be responding to selection. It also restricts other device's use of the bus. The alternative method of utilizing the buffer space is to use disconnection and reconnection.

4.0 Run-Time Software (Continued)

4.6 DISCONNECTION AND RECONNECTION

4.6.1 Disconnection

If an initiator sends the IDENTIFY message to the target, it can indicate that it supports disconnection by setting bit 6 of this message. If this bit is not set, or the message not sent, the initiator is assumed not to support disconnection, and no disconnection is attempted. This software uses disconnection in two places; 1) the board is reserved, currently printing and selected by the reserving initiator; 2) the board has been released, but not finished printing.

If the board is busy printing it can be advantageous to disconnect from the reserved initiator after reading in the command block, and then reconnect having ensured that the command can be implemented i.e., there is enough free space in the queue. For any other initiator the response is the same as before, the status of RESERVATION CONFLICT is returned until the unit is released. After this the controller will disconnect from any initiator attempting to select it, until it has enough free space in its buffer.

disconnect() is the routine which carries out the disconnection procedure. This can only take place after the command phase and before the data phase, with an initiator that sent the IDENTIFY message with the disconnection bit set. The target follows the Command Out phase with a Message In phase and sends the DISCONNECT message, after which it can drop $\overline{BSY}$, to release the bus. If the DISCONNECT message is not sent the initiator will treat the deasserting of $\overline{BSY}$ as an illegal termination of command. Before releasing $\overline{BSY}$ *exchange()* is called to store the ID of the initiator and the command it wants to execute. *reset()* is again used to initialize the interrupts and release $\overline{BSY}$.

disconnect() stores in a variable, recon_data, the amount of print buffer which will be required to execute the command. For a print, this depends on the length of data to be transferred, for any other command the amount is zero. recon_data is used to determine when the target can reconnect to the initiator i.e., once there is enough space free in the buffer.

After disconnection the program will return to printing, at which point it may be selected by another initiator. As it can only disconnect from one initiator at a time it must return a status of BUSY.

In *outbuf()* it can be seen that reconnection takes place when the free space in the print buffer is greater than the number of bytes to be transferred. *reconnect()* calls the functions which action the correct reconnection procedure, beginning with arbitration.

4.6.2 Arbitration

Arbitration requires two functions, one for MODE E another for MODE N, due to the fundamental difference in their methods of arbitration i.e., **MODE E is interrupt driven, MODE N is polled**. These two functions, *Narbitrate()* and *Earbitrate()*, carry out the same basic operation, arbitrate for the bus until successful, but do so in distinctly different ways.

To arbitrate for the bus a device must wait for a Bus Free Phase, when $\overline{BSY}$ is continuously inactive for 400 ns with $\overline{SEL}$ inactive. After a Bus Free Delay of 800 ns the SCSI device should assert $\overline{BSY}$, and assert its SCSIID bit on the bus. After a further 2.2 μs Arbitration Delay the data bus should be examined, and the device with the highest priority SCSI ID bit asserted wins arbitration.

In MODE N the EASIODR should be written with the SCSIID and the arbitration bit in the EASIMR2 set. The interrupts must be initialized to cause a jump to routine *servn()*. The EASI will wait for a Bus Free Phase then, after a Bus Free Delay, it will assert $\overline{BSY}$, and put the contents of the EASIODR onto the bus. The user must poll the AIP bit of the EASIICR to determine when arbitration has begun and check the LA bit to ensure that arbitration has not been lost. The LA bit is set if another initiator asserts $\overline{SEL}$ during arbitration. If AIP is active and LA inactive the user must examine the EASICSD to determine whether it is the highest priority device arbitrating. If arbitration is lost the arbitration bit in the EASIMR2 must be reset and the whole procedure begun again. The device shows it has won arbitration by asserting $\overline{SEL}$. An interrupt during MODE N arbitration is treated as a selection, so *servn()* enables parity, since there is no parity checking during arbitration, and calls *main()*.

In MODE E arbitration is interrupt driven, allowing the board to continue printing while waiting to arbitrate. For a busy SCSI bus typically many milliseconds, and potentially many seconds, can be taken up arbitrating. In MODE E this time can be utilized, thus increasing the system throughput. For this application the time gained is used to continue printing, in other applications, such as a disk controller, it could be used in data cacheing, overlapped seeks, etc.

In MODE E arbitration the EASIODR must be written with the SCSIID and parity checking cancelled in the EASIMR2. The interrupts are set to allow selection and arbitration interrupts, with the jump table loaded with *serva()*. This sets a flag to show an interrupt has occurred. The enhanced arbitration is initiated, with a write to the arbitration bit of the EASIEMR. This causes the EASI to wait for a Bus Free Phase; delay a Bus Free Delay; assert $\overline{BSY}$ and the EASIODR; delay an Arbitration Delay then interrupt the processor. The interrupt causes the flag to be set that shows the state of arbitration should be examined. While waiting for this interrupt the printer carries on printing out data, until front equals rear.

After the interrupt is detected the EASIISR can be read to determine the cause. If it is not an arbitration interrupt parity checking is enabled and the interrupt treated as a selection, by calling *main()*. If the interrupt is signalling the commencement of arbitration, the procedure is the same as in MODE N, with the LA bit of the EASIICR being examined, and priority determined by reading the EASICSD. If arbitration is lost, interrupts must be reset, the arbitration bit in the EASIEMR reset, and the whole procedure begun again. If arbitration is successful the user asserts $\overline{SEL}$.

Function *printarb()* is used to print data during arbitration in the same way as *printit()* does normally. The difference here is that the arbitration interrupt should be serviced as quickly as possible.

This routine must be left if an interrupt occurs. Instead of waiting for the printer to come on-line if an error occurs this routine simply does not send the character.

4.0 Run-Time Software (Continued)

4.6.3 Reselection

When arbitration has been won the disconnected initiators ID and command block must be restored, and the initiator reselected. This reselection is carried out by function *reselect( )*.

The EASISER is cleared to stop it responding to the reselection and the initiator ID bit written into the EASIODR along with the SCSIID. I/O must be asserted to show this is a reselection. The EASIODR is asserted onto the bus and the relevant arbitration bit, depending on MODE reset. This deasserts $\overline{BSY}$. Parity checking is enabled, and the board waits a selection timeout delay of 250 ms for the initiator to respond, by asserting $\overline{BSY}$. If it does not, $\overline{RST}$ is asserted, resetting the whole SCSI bus.

If the initiator does respond, the target must assert $\overline{BSY}$, then deassert $\overline{SEL}$ and take the EASIODR off the bus. For MODE E the interrupt mask can be set, and the arbitration interrupt reset. Reselection is now complete.

4.6.4 Reconnection

After *reselect( )* the IDENTIFY message is sent by the target with the disconnect bit set. The command that was previously sent is processed, if this message is received successfully. After processing the command status and the COMMAND COMPLETE message are returned. *reset( )* is again used to set the interrupts and release $\overline{BSY}$.

If a print was not current at the time of reconnection, and the reconnected command was PRINT, this is handled within *reconnect( )*. As long as the status is GOOD the print is carried out. However, *reconnect( )* will be mostly called from *outbuf( )*, when the queue has enough free space to process the outstanding command. Any reconnection missed by *outbuf( )* is captured in *main( )*.

4.7 ERROR HANDLING

Throughout this document the handling of errors has been discussed as the errors arose, but there are some remaining to be discussed. These are phase or parity errors during command transfers and message errors.

4.7.1 General Error Handling

While this board is connected to the SCSI bus, with $\overline{BSY}$ active, *gen_int( )* is generally used to handle interrupts. This routine responds to an unexpected interrupt by checking the parity and phase flags, in the EASIBSR for MODE N, the EASIISR for MODE E. It then resets the interrupts, before setting appropriate flags. Sense is set to ABORTED COMMAND with a status of CHECK CONDITION if an error has occurred.

After calling *select( )*, to respond to selection, *main( )* then calls function *set_up( )*, which sets the mask and jump table to respond to an interrupt. This is also the routine which checks to see if $\overline{ATN}$ is asserted. If it is *getmes( )* is called to enter the Message Out phase and fetch the IDENTIFY message. *messout( )* processes the message. This is where the target determines if the initiator supports disconnection.

If an error occurs during *select( )* or *set-up( )*, detected by *gen_int( )*, the board sends relevant status and a message of COMMAND COMPLETE, before releasing the bus. No attempt is made to recover from the error. Similarly, if a phase error occurs during the command phase, status and sense are returned. However, if a parity error occurs during a command phase, the target can attempt a recovery by sending the RESTORE POINTERS message. This message instructs the initiator to reset the command pointer to the beginning of the command block, allowing the target to re-enter the Command Out phase and re-transfer the command block. If there is a parity error again, status is returned, along with the COMMAND COMPLETE message. If the second transfer is successful, the command execution continues as normal.

4.7.2 Message Errors

If an initiator wishes to respond to a message sent by the target it indicates this by asserting $\overline{ATN}$, before releasing $\overline{ACK}$ to finish the transfer. The target should then enter the Message In phase, and transfer message bytes until $\overline{ATN}$ goes inactive, up to 256 bytes. If the initiator attempts to send more than this the target will send the MESSAGE REJECT message and terminate the command with status of CHECK CONDITION and sense set to ABORTED COMMAND. If a parity error occurs during the message transfer, the target must wait until the transfer is complete, then instruct the initiator to resend all previous message bytes, by asserting $\overline{REQ}$ before changing phase. If the parity error occurs again the command will be terminated with status of CHECK CONDITION and sense ABORTED COMMAND.

The relevant functions for message phases are *messin( )*, *getmes( )* and *messout( )*. *messin( )* is used to send a message to the initiator, while *getmes( )* fetches a message from the initiator. *messout( )* processes the message sent by the initiator. The supported messages are now explained.

IDENTIFY: establishes the use of a greater message set, indicates the ability to support, or not support, disconnection and gives a LUN number, if necessary.

ABORT: if the reserving initiator sends this message to the target it causes the board to be reset.

BUS DEVICE RESET: This is similar to ABORT, except any initiator can implement it, resetting the board.

MESSAGE PARITY ERROR: On receiving this message the target attempts to resend the last message sent, and if this fails, terminates the command with status of CHECK CONDITION and sense set to HARDWARE ERROR.

MESSAGE REJECT: the targets response to this message is determined by the message being rejected. If the last message sent was COMMAND COMPLETE, the MESSAGE REJECT is ignored, since the initiator must support the mandatory message. A MESSAGE REJECT in reply to a DISCONNECT causes the disconnection to be cancelled. If it was in reply to a MESSAGE REJECT sent, the command is terminated with a status of CHECK CONDITION and sense set to HARDWARE ERROR. If the message sent was IDENTIFY (during reconnection) the target immediately goes to the Bus Free Phase and aborts the command. No Status or Message In phases are attempted, though sense is set to HARDWARE ERROR.

4.0 Run-Time Software (Continued)

It should be noted that *messin( )* has been written to only allow *getmes( )* to be called, from inside *messin( )*, once. Alternatively on parity error the target and initiator could eternally cycle, sending each other the MESSAGE PARITY ERROR message.

4.7.3 Non-Recoverable Errors

The following are errors which occur **during selection**, so severe that system operation is terminated, with an error code displayed on the LED.

The first two concern a DP5380.

Error 4
Wrong interrupt flags active.

Error 5
SEL inactive.

These errors concern either a DP5380 or a DP8490.

Error 6
SCSIID bit not active on bus.

Error 7
More than two bits active on bus.

The final errors are for a DP8490 only.

Error 8
SCSI parity error.

Error 9
Select flag inactive.

5.0 User Guide

The SCSI Printer Controller (SPC) is a Small Computer System Interface target board, which can use either the DP5380 Asynchronous SCSI interface (ASI) or DP8490 Enhanced Asynchronous SCSI Interface (EASI). It can be selected and used by an initiator as described in the ANSX3.131-1986 SCSI standard as defined by the ANSI X3T9.2 committee. This document will explain the installation of SPC, and show how it can be used in a system. By way of example a description is given of how the SPC can be used with an Advanced Storage Concepts ASC-88 IBM PC-SCSI Manager II™ host adaptor.

5.1 INSTALLATION

This section explains how the board must be set up before use, and the type of connectors required to interface to it. Any reference made to an EASI also applies for an ASI.

5.1.1 POWER

The SPC can be installed in a Personal Computer (PC) where it takes power from the backplane. There are two connections to +5V and two to ground. These are the only connections made to the backplane. Alternatively power can be taken from the available connector block *(Figure 5.1)*.

When the board receives power the LED will come on, and stay on if the board passes its self diagnostics. If it fails an error message will be displayed by the LED, indicating the source of error. This will be further explained in section 5.1.5. If the LED does not come on some fatal error has occurred.

5.1.2 SCSI CONNECTOR

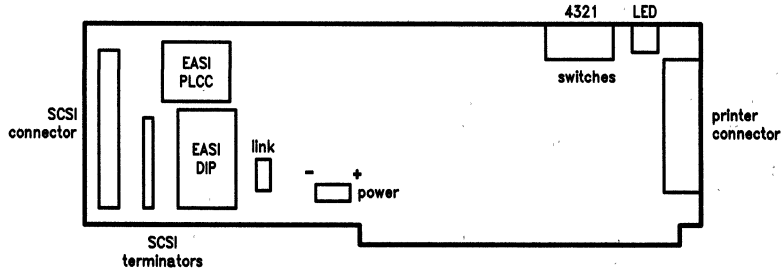
The SCSI bus should consist of a 50 way flat ribbon cable a maximum of 6.0 meters long. Both ends of this cable should have all SCSI lines terminated, with a 330Ω resistor to ground and a 220Ω resistor to power. SCSI devices are daisy chained along this cable, with SCSI signals common to all devices.

The SPC contains sockets for two DP8490 or DP5380 devices, one a PLCC socket the other DIL. **Only one of these sockets should contain a device.**

To comply with the regulations on terminating resistors six SIL resistor packs are available between the SCSI connector and the EASI DIL package *(Figure 5.1)*. If this board is not to be used at the end of the SCSI cable all of these six packs must be removed. The resistors have been socketed for this purpose.

An option available in the SCSI standard is to supply terminator power on the cable, so terminators at either end of the bus can use the same power. If the device at one end of the bus is unpowered its terminators can receive power from the cable, and the bus will operate correctly. It should be noted that other manufacturer's CMOS devices will not work in this configuration since if not powered they pull the SCSI lines low. National Semiconductor's DP5380 and DP8490 have special input protection to prevent this.

Pin 26 on the SCSI connector is the terminator power pin, which can be left floating by leaving the jumper in position 2, see *Figure 5.2*. By moving the jumper to position 1 the terminator power is supplied to this pin. *Figure 5.3* shows the two possible configurations. Terminator power is fed through a Schottky barrier diode to prevent a backflow of power into the board.



TL/F/10082–6

FIGURE 5.1

5.0 User Guide (Continued)

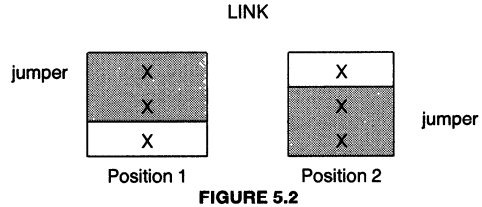
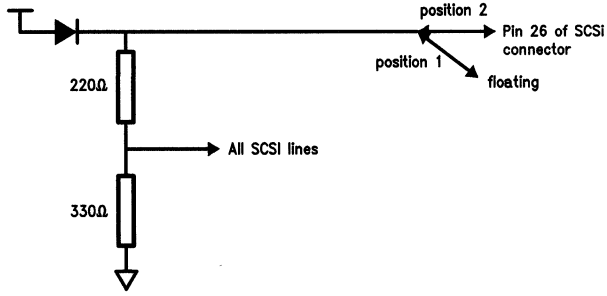


FIGURE 5.2



TL/F/10082–7

FIGURE 5.3

5.1.3 SWITCH BLOCK SETTINGS

The switch block is included to allow the user to select the SCSI ID of the SPC. This is used to identify the board during selection phases and determine its priority in arbitration. The SCSI ID is read in as a three bit binary number and converted in software to an eight bit pattern, with one bit active. For an ID of 0 the least significant bit is active, for an ID of 7 the most significant bit is active.

The three bit number is taken from the switch block (see *Figure 5.1*), using switches 1, 2 and 3. 1 is the most significant bit, 4 is unused. These switches should be set to give a unique ID for this bus. An ID of 0 is suggested, making the printer the lowest priority device on the bus. These switches must be set up before power is applied, as they are only checked during the board diagnostics.

5.1.4 PRINTER CONTROLLER

The printer connector is a standard IBM 25 way 'D' type connector. The SPC software controls data transfers to the printer using the BUSY and $\overline{\text{STROBE}}$ signals.

5.1.5 ERROR MESSAGES

Non-recoverable errors cause a four bit binary number to be displayed by the LED. This indicates the source of error.

The LED displays the number by occulting for 1 second to show a 1, ½ second to show a 0. The code is displayed most significant bit first, with the LED on for 1 second between bits. The error number is repeatedly displayed, between breaks of 2 seconds when the LED is on.

The error numbers, and their cause, are shown below. The first four errors are generated during the board diagnostics, the others occur when a selection fails. The SCSI controller will be referred to as an EASI, unless the error is specific to either a DP5380 or DP8490. These possible differences are due to the DP8490's more extensive testing, and because it handles a selection differently.

Error 0

The DMA could not be accessed. This is possibly a damaged device.

Error 1

EASI can not be accessed. Device could be damaged, or not properly terminated.

Error 2

DP8490 has failed loopback test. Device could be damaged.

Error 3

DP8490 has failed the loopback DMA test. In this test a DMA transfer to memory from the DP8490 is attempted. Failure here could indicate an error in memory, processor, DMA, DP8490 or PALs.

These first four errors concern problems internal to the board. The following errors indicate a bus problem that occurred while the board was waiting for selection.

Error 4

DP5380 error flag active.

Error 5

DP5380 found select line inactive. Possible selection timeout or bus error.

Error 6

EASI ID bit not active on bus. Possible selection timeout or bus error.

Error 7

EASI detected more than two bits active on bus. During a selection phase the initiator can only assert its own ID and the targets ID on the bus. This indicates a bus error.

Error 8

DP8490 SCSI parity error. Error on bus.

Error 9

DP8490 select flag inactive. This is board hardware error, possibly in DP8490 or PAL, possibly interrupt line.

For any of these problems the user should try switching the device on and off again. For bus errors the cable and terminators should be checked along with any other devices on the bus.

5.0 User's Guide (Continued)

5.2 DRIVER SOFTWARE

By way of example it will be shown how software can be written to drive the SPC from an ASC-88 host adaptor, installed in a PC. The ASC-88, like many commercially available host adaptors, handles all low level SCSI signal controls. The user controls the command to be implemented by means of a Job Control Block, JCB, which is passed to it.

The software required to use an ASC-88 is on the supplied floppy disk. This disk contains both the source code, explained later, and the executable code. This is called ***printout.exe*** and can be implemented in the form:

printout *filename*

The *filename* supports the MS-DOS use of directories and paths:

e.g., printout a: /ASC /ASC.C

The second executable command, called ***stoprint.exe***, can be used to terminate a print. When this command is executed the SPC flushes the print queue and initializes the printer.

5.2..1 SEQUENCE OF COMMANDS

Although the SPC requires no prescribed sequence of SCSI commands **an initiator must reserve the unit before a print can take place**. Otherwise the sequence of commands specified here indicate how the SPC could be used.

Any print should begin with a TEST UNIT READY command. On receiving this command the SPC tests the printer to ensure it has paper and is not in an error condition. If the status of GOOD is returned the user should then attempt RESERVE UNIT. If the command is successful the SPC will only execute commands from this initiator. This prevents print data from two or more sources being mixed.

The user then sends PRINT commands with a maximum length of data per transfer of 2 kB. If the file to be printed is larger than 2 kB the data must be sent in several blocks. This limit, set to minimize bus latency, is more fully explained in Section 4.4.3.

These PRINT commands transfer the data to the SPC print buffer. To check if an error occurred when the SPC attempted to transfer the data to the printer the user can send a REQUEST SENSE command. If the returned data is NO SENSE the printer is still operational. Any other sense indicates an error.

The final command sent is RELEASE UNIT. This allows the SPC to be used by other initiators.

5.2.2 ASC-88 SOFTWARE

All ASC-88 software is written for a Microsoft® C compiler. File ASC.C contains the main run-time software. ASC STRUC.LIB sets up the structure which is used as a Job Control Block, while ASCCOM.LIB defines the JCBs for particular SCSI commands. The routines used in ASC.C, other than SCSI command calls, are in ASCROT.LIB and UTIL.LIB. CONSTANT.H contains the constants used throughout these files. If the user wishes the SCSI ID of the target board to be anything other than zero they should change the value of *TARGET_ID* in this file, and recompile the code for both commands.

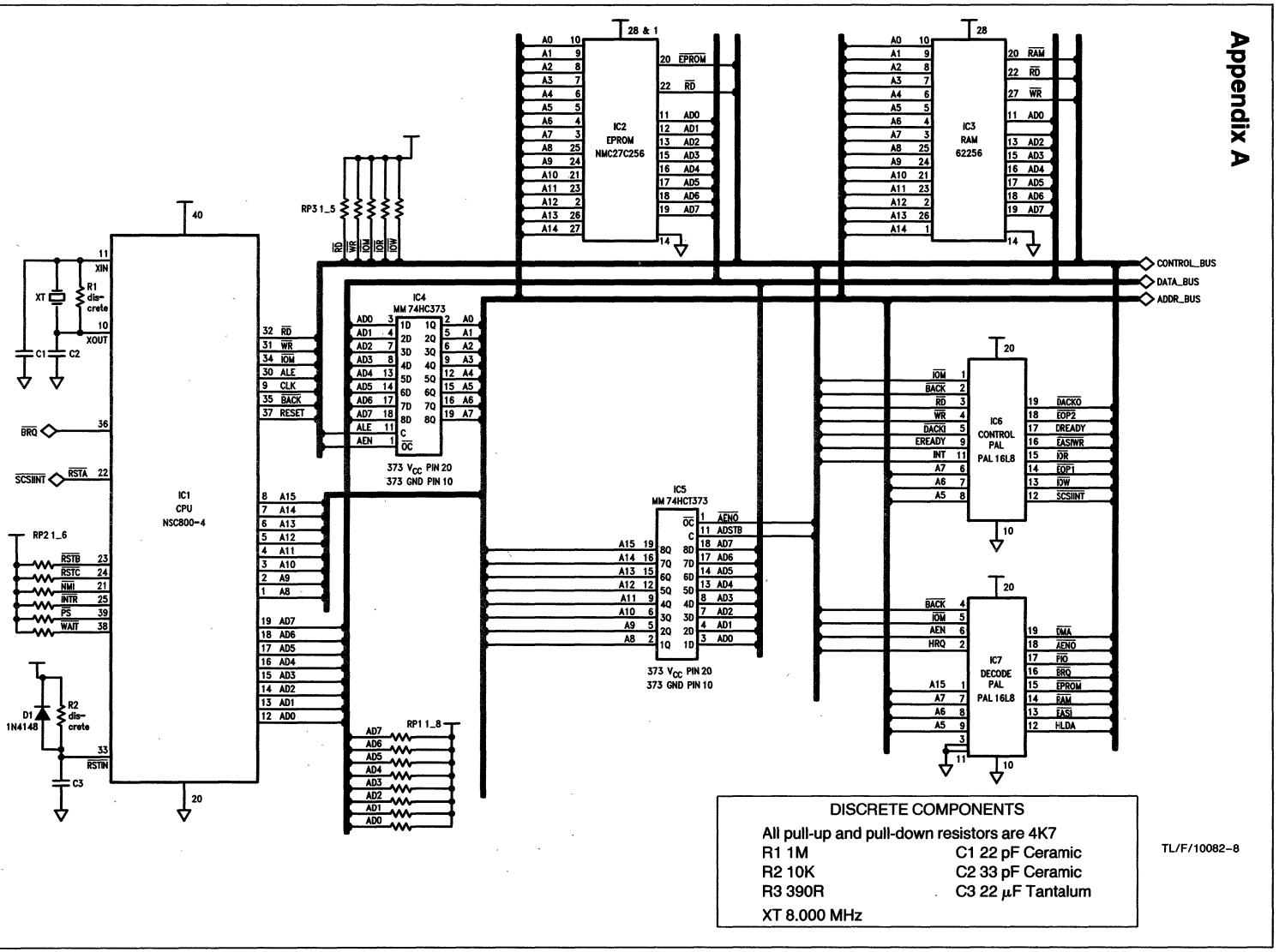
STDIO.H is a Microsoft library containing constants, macro definitions and function declarations for I/O stream operations. This controls opening files, reading files, detecting file ends and closing files. DOS.H, also a Microsoft library, handles the interface to MS-DOS®. This allows the user to set up registers and execute interrupts. The final Microsoft library CONIO.H allows the user to fetch information from the keyboard.

The SCSI-BIOS™ EPROM in the ASC-88 is accessed by generating interrupt number forty. The value in the 'AH' register determines what SCSI-BIOS software is used. SCSI-PRO™ implements the SCSI command specified in the JCB whose starting address is passed in the 'BX' and 'ES' registers.

ASC.C must be compiled and loaded to produce a file printout.exe. When executed this code opens the file specified in the command line and instigates the sequence of commands, outlined in the previous section, to print it out. After these commands the file is closed.

If the printer enters an error state while communicating with the initiator a message will be displayed on screen and the user is given the option of terminating the print. If the print is to be continued the user must correct the printer error and press a key to restart the print. If an error occurs when the SPC has already received all the data from the PC the board will wait until the print error is corrected and continue printing.

STOPRINT.C contains the run-time software required to terminate a print. This uses the library CHECK.LIB, which checks the success of a FLUSH BUFFER command. To terminate an unwanted long print the user can: take the printer off-line; press escape to leave *printout*; send the *stoprint* command.

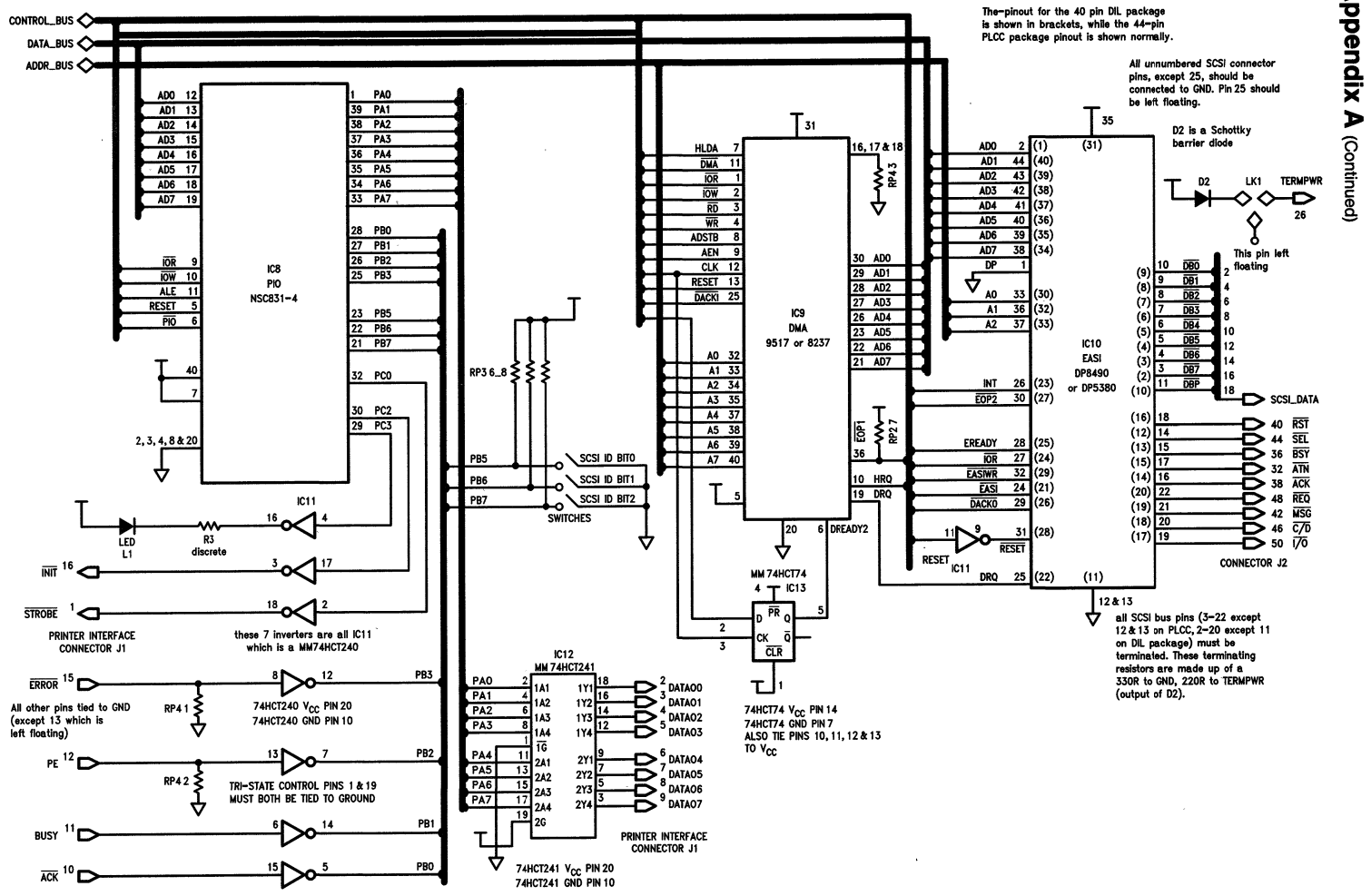


DISCRETE COMPONENTS

All pull-up and pull-down resistors are 4K7

R1 1M	C1 22 pF Ceramic
R2 10K	C2 33 pF Ceramic
R3 390R	C3 22 μF Tantalum
XT 8.000 MHz	

TL/F/10082–8


CONTROL_BUS
DATA_BUS
ADDR_BUS
IC8
PIO
NSC831-4
IC9
DMA
9517 or 8237
IC10
EASI
DP8490
or DP5380
IC11
IC12
MM74HCT241
IC13
MM74HCT74
SCSI ID BIT0
SCSI ID BIT1
SCSI ID BIT2
SWITCHES
LED
L1
R3
discrete
PRINTER INTERFACE
CONNECTOR J1
these 7 inverters are all IC11
which is a MM74HCT240
All other pins tied to GND
(except 13 which is
left floating)
74HCT240 VCC PIN 20
74HCT240 GND PIN 10
TRI-STATE CONTROL PINS 1 & 19
MUST BOTH BE TIED TO GROUND
74HCT241 VCC PIN 20
74HCT241 GND PIN 10
74HCT74 VCC PIN 14
74HCT74 GND PIN 7
ALSO TIE PINS 10, 11, 12 & 13
TO VCC
The pinout for the 40 pin DIL package
is shown in brackets, while the 44-pin
PLCC package pinout is shown normally.
All unnumbered SCSI connector
pins, except 25, should be
connected to GND. Pin 25 should
be left floating.
D2 is a Schottky
barrier diode
This pin left
floating
CONNECTOR J2
SCSI_DATA
all SCSI bus pins (3–22 except
12 & 13 on PLCC, 2–20 except 11
on DIL package) must be
terminated. These terminating
resistors are made up of a
330R to GND, 220R to TERMPWR
(output of D2).
TL/F/10082–9

Section 5

Floppy Disk Controller

Section 5 Contents

National Semiconductor

DP8473 Floppy Disk Controller PLUS-2™

General Description

This controller is a full featured floppy disk controller that is software compatible with the μPD765A, but also includes many additional hardware and software enhancements. These enhancements include additional logic specifically required for an IBM® PC, PC-XT®, PC-AT®, or PS/2® design.

This controller incorporates a precision analog data separator, that includes a self trimming delay line and VCO. Up to three external filters are switched automatically depending on the data rate selected. This provides optimal performance at the standard PC data rates of 250/300 kb/s, and 500 kb/s. It also enables optimum performance at 1 Mb/s (MFM). These features combine to provide the lowest possible PLL bandwidth, with the greatest lock range, and hence the widest window margin.

This controller includes write precompensation circuitry. A shift register is used to provide a fixed 125 ns early-late precompensation for all tracks at 500k/300k/250 kb/s (83 ns for 1 MB/s), or a precompensation value that scales with

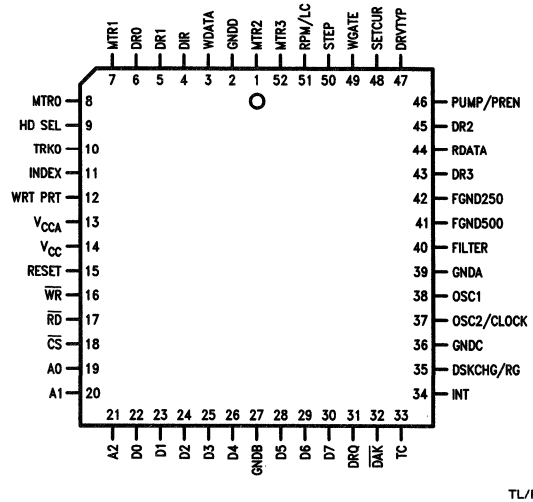
(Continued)

Features

- Fully μPD765A and IBM-BIOS compatible
- Integrates all PCXT®, PCAT®, and most PS/2® Logic
  - On chip 24 MHz Crystal Oscillator
  - DMA enable logic
  - IBM compatible address decode of A0–A2
  - 12 mA μP bus interface buffers
  - 40 mA floppy drive interface buffers
  - Data rate and drive control registers
- Precision analog data separator
  - Self-calibrating PLL and delay line
  - Automatically chooses one of three filters
  - Intelligent read algorithm
- Two pin programmable precompensation modes
- Other enhancements
  - up to 1 Mb/s data rate
  - Implied seek up to 4000 tracks
  - IBM or ISO formatting
- Low power CMOS, with power down mode

Connection Diagrams

Plastic Leaded Chip Carrier

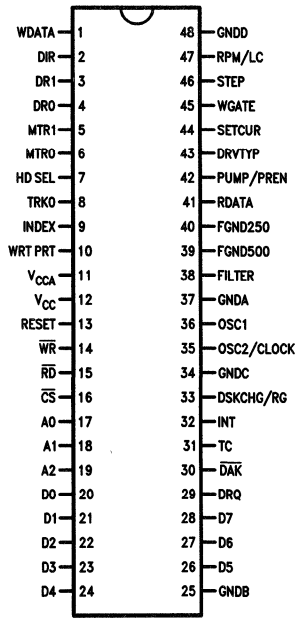


TL/F/9364–1

Top View

Order Number DP8473V
See NS Package Number V52A

Dual-In-Line Package



TL/F/9384–2

Top View

Order Number DP8473N
See NS Package Number N48A

General Description (Continued)

the data rate, 83 ns/125 ns/208 ns/ 250 ns for data rates of 1.0M/500k/300k/250 kb/s respectively.

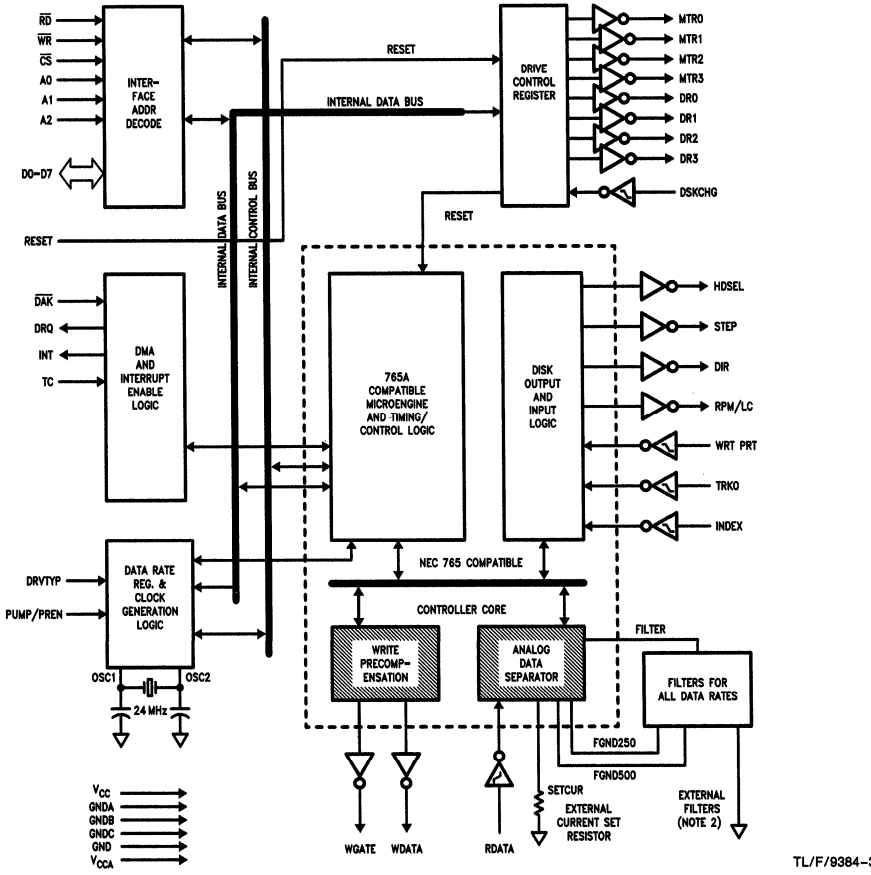
Specifically to support the PC-AT and PC-XT design, the Floppy Disk Controller PLUS-2 includes address decode for the A0–A2 address lines, the motor/drive select register, data rate register for selecting 250/300/500 kb/s, Disk Changed status, dual speed spindle motor control, low write current and DMA/interrupt sharing logic. The controller also supports direct connection to the μP bus via internal 12 mA buffers. The controller also can be connected directly to the disk drive via internal open drain high drive outputs, and Schmitt inputs.

In addition to this logic the DP8473 includes many features to ease design of higher performance drives and future controller upgrades. These include 1.0 Mb/s data rate, extended track range to 4096, Implied seeking, working Scan Commands, motor control timing, both standard IBM formats as well as Sony 3.5" (ISO) formats, and other enhancements.

This device is available in a 52 pin Plastic Chip Carrier, and in a 48 pin Dual-In-Line package.

Table of Contents

Block Diagram



TL/F/9384–3

Note 1: The MTR2, MTR3, DR2, and DR3 are not available on the 48 pin DIP (DP8473N, J) versions.

Note 2: See *Figure 4* for filter description.

Note 3: Total transistor count is 29,700 (approx).

FIGURE 1. DP8473 Functional Block Diagram

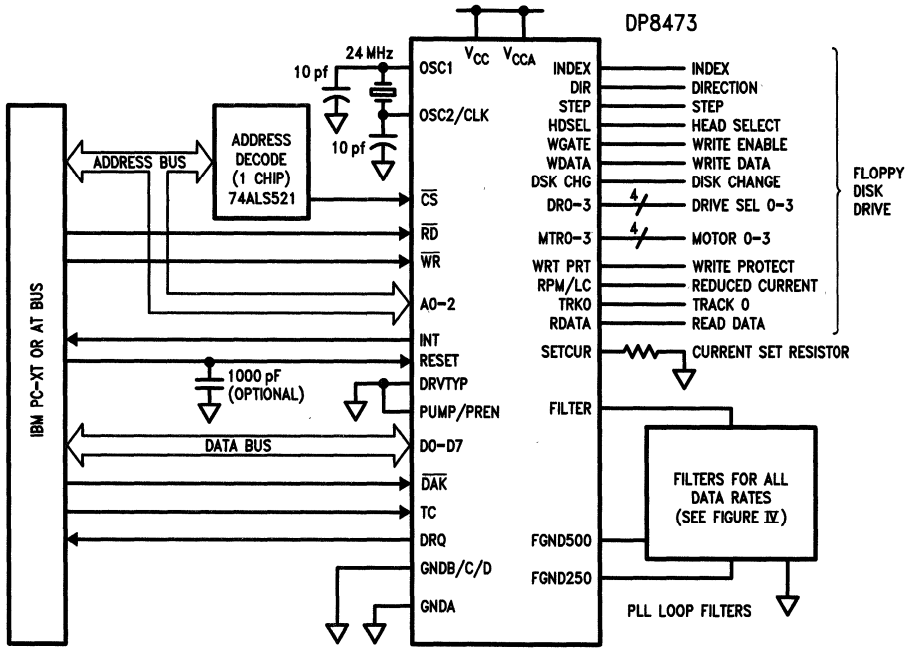
Pin Descriptions

Symbol	DP8473 PCC	DP8473 DIP	Function
MTR2	1	—	This is an active low motor enable line for drive 2, which is controlled by the Drive Control register. This is a high drive open drain output.
GNDD	2	48	This pin is the digital ground for the disk interface output drivers.
WDATA	3	1	This is the active low open drain write precompensated serial data to be written onto the selected disk drive. This is a high drive open drain output.
DIR	4	2	This output determines the direction of the head movement (low = step in, high = step out). When in the write or read modes, this output will be high. This is a high drive open drain output.
DR1	5	3	This is an active low drive select line for drive 1 that is controlled by the Drive Control register bits D0, D1. The Drive Select bit is ANDed with the Motor Enable of the same number. This is a high drive open drain output.
DR0	6	4	This is an active low drive select similar to DR1 line except for drive 0.
MTR1	7	5	This is an active low motor enable line for drive 1. Similar to MTR2.
MTR0	8	6	This is an active low motor enable line for drive 0. Similar to MTR2.
HD SEL	9	7	This output determines which disk drive head is active. Low = Head 1, Open (high) = Head 0. This is a high drive open drain output.
TRK0	10	8	This active low Schmitt input tells the controller that the head is at track zero of the selected disk drive.
INDEX	11	9	This active low Schmitt input signals the beginning of a track.
WRT PRT	12	10	This active low Schmitt input indicates that the disk is write protected. Any command that writes to that disk drive is inhibited when a disk is write protected.
V_{CCA}	13	11	This pin is the 5V supply for the analog data separator circuitry.
V_{CC}	14	12	This pin is the 5V supply for the digital circuitry.
RESET	15	13	Active high input that resets the controller to the idle state, and resets all the output lines to the disk drive to their disabled state. The Drive Control register is reset to 00. The Data Rate register is set to 250 kb/s. The Specify command registers are not affected. The Mode Command registers are set to the default values. Reset should be held active during power up. To prevent glitches activating the rest sequence, a small capacitor (1000 pF) should be attached to this pin.
$\overline{WR}$	16	14	Active low input to signal a write from the microprocessor to the controller.
$\overline{RD}$	17	15	Active low input to signal a read from the controller to the microprocessor.
$\overline{CS}$	18	16	Active low input to enable the $\overline{RD}$ and $\overline{WR}$ inputs. Not required during DMA transfers. This should be held high during DMA transfers.
A0, A1, A2	19–21	17–19	Address lines from the microprocessor. This determines which registers the microprocessor is accessing as shown in Table IV in the Register Description Section. Don't care during DMA transfers.
D0–D4	22-26	20–24	Bi-directional data lines to the microprocessor. These are the lower 5 bits and have buffered 12 mA outputs.
GNDB	27	25	This pin is the digital ground for the 12 mA microprocessor interface buffers. This includes D0–D7, INT, and DRQ.
D5–D7	28–30	26–28	Bi-directional data lines to the microprocessor. These upper 3 bits have buffered 12 mA outputs.
DRQ	31	29	Active high output to signal the DMA controller that a data transfer is needed. This signal is enabled when D3 of the Drive Control Register is set.
$\overline{DAK}$	32	30	Active low input to acknowledge the DMA request and enable the $\overline{RD}$ and $\overline{WR}$ inputs. This signal is enabled when D3 of the Drive Control Register is set.
TC	33	31	Active high input to indicate the termination of a DMA transfer. This signal is enabled when the DMA Acknowledge pin is active.
INT	34	32	Active high output to signal that an operation requires the attention of the microprocessor. The action required depends on the current function of the controller. This signal is enabled when D3 of the Drive Control Register is set.

Pin Descriptions (Continued)

Symbol	DP8473 PCC	DP8473 DIP	Function
DSKCHG/RG	35	33	This latched Schmitt input signal is inverted and routed to D7 of the data bus and is read when address xx7H is enabled. When the RG bit in the Mode Command is set, this pin functions as a Read Gate signal that when low forces the data separator to lock to the crystal, and when high it locks to data for diagnostic purposes.
GNDC	36	34	This pin is the digital ground for the controller's digital logic, including all internal registers, micro-engine, etc.
OSC2/CLOCK	37	35	One side of the external 24 MHz crystal is attached here. If a crystal is not used, a TTL or CMOS compatible clock is connected to this pin.
OSC1	38	36	One side of an external 24 MHz crystal is attached here. This pin is tied low if an external clock is used.
GNDA	39	37	This pin is the analog ground for the data separator, including all the PLLs, and delay lines.
FILTER	40	38	This pin is the output of the charge pump and the input to the VCO. One or more filters are attached between this pin and the GNDA, FGND250 and FGND500 pins.
FGND500	41	39	This pin connects the PLL filter for 500k(MFM)/250k(FM) b/s to ground. This is a low impedance open drain output.
FGND250	42	40	This pin connects the PLL filter for 250k(MFM)/125k(FM) b/s or 300k(MFM)/150k(FM) b/s to ground. This is a low impedance open drain output.
DR3	43	—	This is the same as DR0 except for drive 3.
RDATA	44	41	The active low raw data read from the disk is connected here. This is a Schmitt input.
DR2	45	—	This is the same as DR0 except for drive 2.
PUMP/PREN	46	42	When the PU bit is set in Mode Command this pin is an output that indicates when the charge pump is making a correction. Otherwise this pin is an input that sets the precomp mode as shown in Table VI. If pin is configured as PUMP, PREN is assumed high.
DRV TYP	47	43	This is an input used by the controller to enable the 300 kb/s mode. This enables the use of floppy drives with either dual or single speed spindle motors. For dual speed spindle motors, this pin is tied low. When low, and 300 kb/s data rate is selected in the data rate register, the PLL actually uses 250 kb/s. This pin is tied high for single speed spindle motor drives (standard AT drive). When this pin is high and 300 kb/s is selected 300 kb/s is used. (See also RPM/LC pin).
SETCUR	48	44	An external resistor connected from this pin to analog ground programs the amount of charge pump current that drives the external filters. The PLL Filter Design section shows how to determine the values.
WGATE	49	45	This active low open drain high drive output enables the write circuitry of the selected disk drive. This output has been designed to prevent glitches during power up and power down. This prevents writing to the disk when power is cycled.
STEP	50	46	This active low open drain high drive output will produce a pulse at a software programmable rate to move the head during a seek operation.
RPM/LC	51	47	This high drive open drain output pin has two functions based on the selection of the DRVTYP pin. 1. When using a dual speed spindle motor floppy drive (DRVTYP pin low), this output is used to select the spindle motor speed, either 300 RPM or 360 RPM. In this mode this output goes low when 250/300 kb/data rate is chosen in the data rate register, and high when 500 kb/s is chosen. 2. When using a single speed spindle motor floppy drive (DRVTYP pin high), this pin indicates when to reduce the write current to the drive. This output is high for high density media (when 500 kb/s is chosen).
MTR3	52	—	This is an active low motor enable line for drive 3.

Typical Application



TL/F/9384–4

Recommended Plastic Chip Carrier Socket:
AMP P/N 821551-1 or equivalent.

FIGURE 2. DP8473 Typical Application

Functional Description

This section describes the basic architectural features of the DP8473, and many of the enhancements provided. Refer to *Figure 1.*

765A COMPATIBLE MICRO-ENGINE

The core of the DP8473 is a μPD765A compatible microcoded engine. This engine consists of a sequencer, program ROM, and disk/misc registers. This core is clocked by either a 4 MHz, 4.8 MHz or 8 MHz clock selected in the Data Rate Register. Upon this core is added all the glue logic used to implement a PC-XT or AT, or PS/2 floppy controller, as well as the data separator and write precompensation logic.

The controller consists of a microcoded engine that controls the entire operation of the chip including coordination of data transfer with the CPU, controlling the drive controls, and actually performing the algorithms associated with reading and writing data to/from the disk. This includes the read algorithm for the data separator.

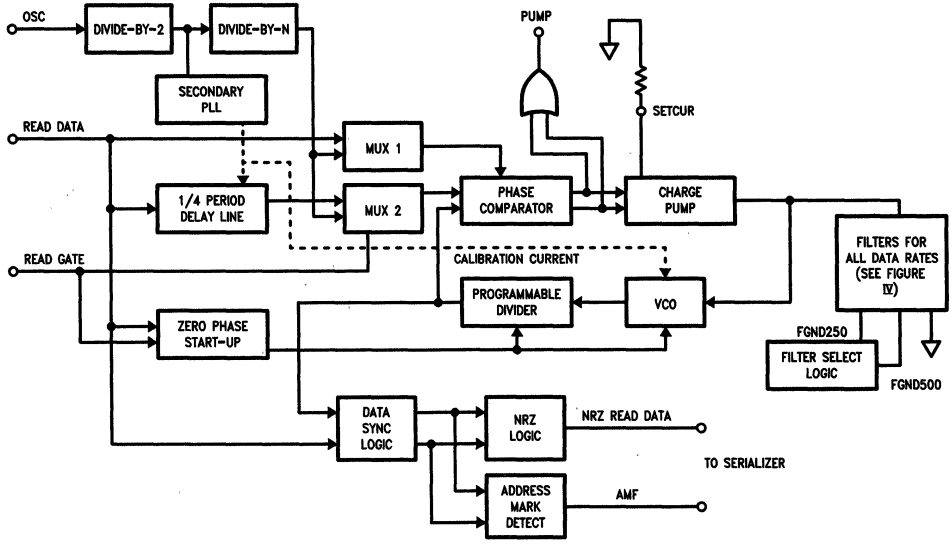
Like the μPD765A, this controller takes commands and returns data and status through the Data Register in a byte serial fashion. Handshake for command/status I/O is provided via the Main Status Register. All of the μPD765A commands are supported, as are many other enhanced commands.

DATA SEPARATOR

The internal data separator consists of an analog PLL and its associated circuitry. The PLL synchronizes the raw data signal read from the disk drive. The synchronized signal is used to separate the encoded clock and data pulses. The data pulses are de-serialized into bytes and then sent to the μP by the controller.

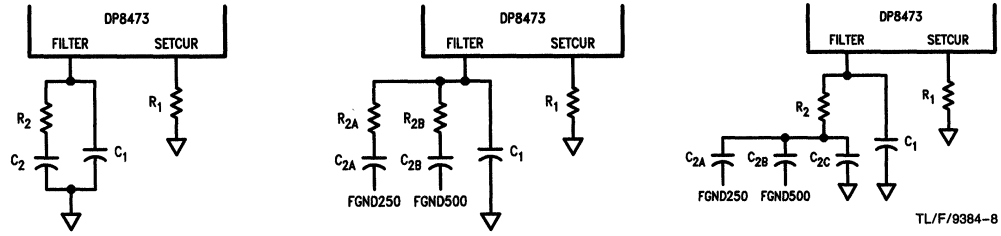
The main PLL consists of four main components, a phase comparator, a filter, a voltage controlled oscillator (VCO), and a programmable divider. The phase comparator detects the difference between the phase of the divider's output and the phase of the raw data being read from the disk. This phase difference is converted to a current which either charges or discharges one of the three external filters. The resulting voltage on the filter changes the frequency of the VCO and the divider output to reduce the phase difference between the input data and the divider's output. The PLL is "locked" when the frequency of the divider is exactly the same as the average frequency of the data read from the disk. A block diagram of the data separator is shown in *Figure 3.*

Functional Description (Continued)



TL/F/9384–5

FIGURE 3. Block Diagram of DP8473's Data Separator



Note: For all filter configurations, 250kb/s and 300 kb/s share the same filter.

FIGURE 4. Typical Configuration for Loop Filters for the DP8473 Showing Component Labels

To ensure optimal performance, the data separator incorporates several additional circuits. The quarter period delay line is used to determine the center of each bit cell. A secondary PLL is used to automatically calibrate the quarter period delay line. The secondary PLL also calibrates the center frequency of the VCO.

To eliminate the logic associated with controlling multiple data rates the DP8473 supports the connection of three filters to the chip via the FGND250, and FGND500 pins (filter ground switches). The controller chooses which filter components to use based on the value loaded in the Data Rate Register. If 500 k(MFM) is being used then the FGND500 is enabled (FGND250 is disabled). If 250 k(MFM) or 300 k(MFM) is being used the FGND250 pin is enabled, and FGND500 is disabled. For 1 Mb/s (MFM) both FGND pins are disabled.

Note for FM encoding: Sometimes, after a reset, the DP8473 will consistently return an error in the Result Phase after an FM read command. If this occurs, simply reset the DP8473 and retry the operation. This may have to be done more than once, as many as five times. Resetting and repeating will prevent soft errors being reported prematurely. This technique is used by MS-DOS.

Figure 4 shows several possible filter configurations. For a filter to cover all data rates (*Figure 4c*), the DP8473 has a 1 Mb/s filter always connected and other capacitor filter components for the other data rates are switched in parallel to this filter. The actual loop filter for 500 kb/s is the parallel combination of the two capacitors, C_{2C} and C_{2B}, attached to the FGND500 pin and to ground. The 250/300 kb/s filter is the parallel combination of the capacitors, C_{2C} and C_{2A}, attached to the FGND250, and ground. If 1 Mb/s need not be supported then the filter configuration of *Figure 4b* can be used. This configuration allows more optimal performance for both 500k and 250/300 kb/s. *Figure 4a* is a simple filter configuration primarily for a single data rate (or multiple data rates with a performance compromise). Table II shows some typical filter values. Other filter configurations and values are possible, these result in good general performance.

While the controller and data separator support both FM and MFM encoding, the filter switch circuitry only supports

Functional Description (Continued)

the IBM standard MFM data rates. To provide both FM and MFM filters external logic may be necessary.

The controller takes best advantage of the internal data separator by implementing a sophisticated ID search algorithm. This algorithm, shown in *Figure 5,* enhances the PLL's lock characteristics by forcing the PLL to relock to the crystal any time the data separator attempts to lock to a non-preamble pattern. This algorithm ensures that the PLL is not thrown way out of lock by write splices or bad data fields.

TABLE II. Typical Filter Values for the Various Data Rates (Assuming ±6% Capture Range)

Data Rate (MFM b/s)	C_2	R_2	C_1	R_1
Filter Values when Using All 3 Data Rates				
1.0M 500k 250/300k	C_{2C} = 0.012 μF C_{2B} = 0.015 μF C_{2A} = 0.033 μF	560Ω	510 pF	5.6 kΩ
Filter Values when Using 250/300 and 500 kb/s				
500k 250/300k	C_{2B} = 0.027 μF C_{2A} = 0.047 μF	560Ω 560Ω	1000 pF	5.6 kΩ
Filter Using Only One Data Rate				
1.0M	C_2 = 0.012 μF	560Ω	510 pF	5.6 kΩ
500k	C_2 = 0.027 μF	560Ω	1000 pF	5.6 kΩ
300/250k	C_2 = 0.047 μF	560Ω	2000 pF	5.6 kΩ

(These values are preliminary and thus are subject to change.)

TABLE III. Data Rates (MFM) versus VCO Divide-By Factor

Data Rate	N
1 Mb/s	4
500 kb/s	8
300 kb/s	16
250 kb/s	16

PLL DIAGNOSTIC MODES

In addition, the DP8473 has two diagnostic modes to enable filter optimization, 1) enabling the Charge Pump output signal onto the PUMP/PREN pin, and 2) providing external control of the Read Gate signal to the data separator. Both modes are enabled in the last byte of the Mode Command.

The Pump output signal indicates when the charge pump is making a phase correction, and hence whether the loop is locked or not.

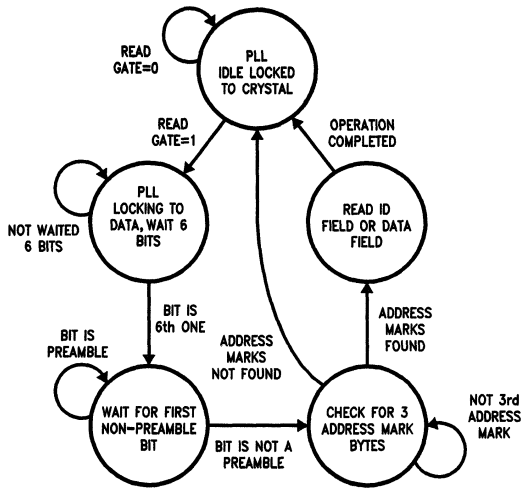
The Read Gate function, when enabled, allows the designer to manually force the data separator to lock to the incoming data or back to the reference clock. This enables easy verification of the lock characteristics of the PLL, by monitoring the FILTER pin, and the Pump signal.

PLL FILTER DESIGN

This section provides information to enable design of the data separator's external filter and charge pump set resistor. This discussion is for a single data rate filter, and can be easily extrapolated to the other filters of *Figure 4.* Table II shows some typical filter component values, but if a custom filter is desired, the following parameters must be considered:

R_1: Charge pump current setting resistor. The current set by this resistor is multiplied by the charge pump gain, K_P which is ~2.5. Thus the charge pump current is:

I_{PUMP} = (2.5) 1.2V/R_1. R_1 should be set to between 3–12 kΩ. This resistor determines the gain of the phase detector, which is $K_D = I_{PUMP}/2\pi$.



TL/F/9384–9

FIGURE 5. Read Algorithm-State Diagram for Data

Functional Description (Continued)

C_2: Filter capacitor in series with R_2. With pump current this determines loop bandwidth.

R_2: Filter resistor. Determines the PLL damping factor.

C_1: This filter capacitor improves the performance of the PLL by providing additional filtering of bit jitter and noise.

K_{VCO}: The ratio of the change in the frequency of the VCO output due to a voltage change at the VCO input. $K_{VCO} \approx 25$ Mrad/s/V. The VCO is followed by a divider to achieve the desired frequency for each data rate. VCO center frequency is 4 MHz for data rates of 1 Mb/s, 500 kb/s, and 250 kb/s (MFM), and is 4.8 MHz for 300 kb/s (MFM).

K_{PLL}: This is the gain of the internal PLL circuitry, and is the product of $V_{REF} \times K_{VCO} \times K_P$. This value is specified in the Phase Locked Loop Characteristics table.

ω_n: This is the bandwidth of the PLL, and is given by,

$$\omega_n = \sqrt{\frac{K_{PLL}}{2\pi C_2 N R_1}}$$

where N is the number of VCO cycles between two phase comparisons. The value of N for the various data rates are shown in Table III.

ζ: The damping factor is set to 0.7 to 1.2 and is given by,

$$\zeta = \frac{\omega_n R_2 C_2}{2}$$

The trade off, when choosing filter components is between acquisition time while the PLL is locking and jitter immunity while reading data. To select the proper components for a standard floppy disk application the following procedure can be used:

1. Choose FM or MFM, and data rate. Determine N from Table III. Determine preamble length (MFM = 12). The PLL should lock within ½ the preamble time.

2. Determine loop bandwidth (ω_n) required, and set the charge pump resistor R_1.

3. Calculate C_2 using:

$$C_2 = \frac{K_{PLL}}{2\pi R_1 N \omega_n^2}$$

4. Choose R_2 using:

$$R_2 = \frac{2\zeta}{\omega_n C_2}$$

6. Select C_1 to be about $\frac{1}{20}$th of C_2.

The above procedure will yield adequate loop performance. If optimum loop performance is required, or if the nature of the loop performance is very critical, then some additional consideration must be given to choosing ω_n and the damping factor. (For a detailed description on how to choose ω_n and ζ, see: **AN-505 Floppy Disk Data Separator Design Guide for the DP8473).**

WRITE PRECOMPENSATION

The DP8473 incorporates a single fixed 3-bit shift register. This shift register outputs are tapped and multiplexed onto the write data output. The taps are selected by a standard precompensation algorithm. This precompensation value can be selected from the PUMP/PREN pin. When this pin is low 125 ns precomp is used for all data rates except 1 Mb/s which uses 83 ns. When PREN is tied high, the precompensation-value scales with data rate at 250 kb/s its 250 ns, for 300 kb/s its 208 ns, at 500 kb/s its 125 ns, and at 1.0 Mb/s its 83 ns. These values are shown in Table VI.

PC-AT AND PC-XT LOGIC BLOCKS

This section describes the major functional blocks of the PC logic that have been integrated on the controller. Refer back to *Figure 1,* the block diagram.

DMA Enable Logic: This is gating logic that disables the DMA lines and the Interrupt output, under the control of the DMA Enable bit in the Drive control register. When the DMA Enable bit is 0 then the INT, and DRQ are held TRI-STATE, and $\overline{DAK}$ is disabled.

Drive Output Buffers/Input Receivers: The drive interface output pins can drive 150Ω ±10% termination resistors. This enables connection to a standard floppy drive. All drive interface inputs are TTL compatible schmitt trigger inputs with typically 250 mV of hysteresis. *The only functional differences between the 52 pin PLCC and the 48 pin DIP version are that the MTR2 and 3, and DR2 and 3 pins have been removed in order to accommodate the 48 pin package.*

Bus Interface-Address Decode: The address decode circuit allows software access to the controller, Drive Control Register, and Data Rate Register (see Table IV for the memory map) using the same address map as is used in the XT, AT, or PS/2. The decoding is provided for A0–A2, so only a single address decoder connected to the chip select is needed to complete the decode. The bus interface logic includes the 8-bit data bus and DRQ/INT signals. The output drive for these pins is 12 mA.

TABLE IV. Address Memory Map for DP8473

A2	A1	A0	R/W	Register
0	0	0	X	None (Bus TRI-STATE)
0	0	1	X	None (Bus TRI-STATE)
0	1	0	W	Drive Control Register
0	1	1	X	None (Bus TRI-STATE)
1	0	0	R	Main Status Register
1	0	1	R/W	Data Register
1	1	0	X	None (Bus TRI-STATE)
1	1	1	W	Data Rate Register
1	1	1	R	Disk Changed Bit*

*When this location is accessed only bit D7 is driving, all others are held TRI-STATE.

Drive Control Register: This 8-bit write only register controls the drive selects, motor enables, DMA enable, and Reset. See Register Description.

Reset Logic: The reset input pin is active high, and directly feeds the Drive Control Register and the Data Rate Register. After a hardware reset the Drive Control Register is reset to all zeros, and the Data Rate Register is set to 250 kb/s data rate. The controller is held reset until the software sets the Drive Control reset bit, after which the controller may be initialized. A software reset to the controller core can be issued by resetting then setting this bit. A software reset does not reset the Drive Control Register, or the Data Rate Register.

Functional Description (Continued)

Data Rate Register and Clock Logic: This is a two bit register that controls the data rate that the controller uses. See Register Description. This register feeds logic that selects the data rates by programming a prescaler that divides the crystal or clock input by either 3, 5, or 6. This causes either 4 MHz, 4.8 MHz and 8 MHz to be input as the master clock for the controller core. If the Drive Type pin is high and a 300 kb/s data rate is chosen, 4.8 MHz is used to generate 300 kb/s, but when the DRVTYP pin is low and 300 kb/s is selected, 4 MHz is used, and the actual data rate is 250 kb/s. See Table VI.

Low Power Mode Logic: This logic is an enhancement over the standard XT, AT, PS/2 design. In the Low Power Mode the crystal oscillator, controller and all linear circuitry are turned off. When the oscillator is turned off the controller will typically draw about 100 μA. The internal circuitry is disabled while the oscillator is off because the internal circuitry is driven from this clock. The oscillator will turn back on automatically after it detects a read or a write to the Main Status or Data Registers. It may take a few milli-seconds for the oscillator to stabilize and the μP will be prevented from trying to access the Data Register during this time through the normal Main Status Register protocol. (The Request for Master bit in the Main Status Register will be inactive.) There are two ways to go into the low power mode. One is to command the controller to switch to low power immediately. The other method is to set the controller to automatically go into the low power mode 500 ms after the beginning of the idle state (based on a 500 kb/s (MFM) data rate). This would be invisible to the software. The low power mode is programmed through the Mode Command.

The Data Rate Register and the Drive Control Register are unaffected by the power down mode. They will remain active. It is up to the user to ensure that the Motor and Drive select signal are turned off.

TABLE V. Truth Table for Drive Control Register

D7	D6	D5	D4	D1	D0	Function
X	X	X	1	0	0	Drive 0 Selected (DR0 = 0)
X	X	1	X	0	1	Drive 1 Selected (DR1 = 0)
X	1	X	X	1	0	Drive 2 Selected (DR2 = 0)
1	X	X	X	1	1	Drive 3 Selected (DR3 = 0)

Crystal Oscillator: The DP8473 is clocked by a single 24 MHz signal. An on-chip oscillator is provided, to enable the attachment of a crystal, or a clock. If a crystal is used, a 24 MHz fundamental mode, parallel resonant crystal should be used. This crystal should be specified to have less than 150Ω series resistance, and shunt capacitance of less than 7 pF. Typically a series resonant crystal can be used, it will just oscillate in parallel mode 30–300 ppm from its ideal frequency.

If an external oscillator circuit is used, it must have a duty cycle of at least 40–60%, and minimum input levels of 2.4V and 0.4V. The controller should be configured so that the clock is input into the OSC2 pin, and OSC1 is tied to ground.

Crystals: Staytek: CX1-SM1-24 MHz(B)
SaRonix: SRX 3164

Register Description

This section describes the register bits for all the registers that are directly accessible to the μP. Table IV (previous page) shows the memory map for these registers. Note that in the PC some of the registers are partially decoded, this is not the case here. All registers occupy only their documented addresses.

MAIN STATUS REGISTER (Read Only)

The read only Main Status Register indicates the current status of the disk controller. The Main Status Register is always available to be read. One of its functions is to control the flow of data to and from the Data Register. The Main Status Register indicates when the disk contoller is ready to send or receive data. It should be read before each byte is transferred to or from the Data Register except during a DMA transfer. No delay is required when reading this register after a data transfer.

D7 Request for Master: Indicates that the Data Register is ready to send or receive data from the μP. This bit is cleared immediately after a byte transfer and will become set again as soon as the disk controller is ready for the next byte.

D6 Data Direction: Indicates whether the controller is expecting a byte to be written to (0) or read from (1) the Data Register.

D5 Non-DMA Execution: Bit is set only during the Execution Phase of a command if it is in the non-DMA mode. In other words, if this bit is set, the multiple byte data transfer (in the Execution Phase) must be monitored by the μP either through interrupts, or software polling as described in the Processor Software Interface section.

D4 Command in Progress: Bit is set after the first byte of the Command Phase is written. Bit is cleared after the last byte of the Result Phase is read. If there is no result phase in a command, the bit is cleared after the last byte of the Command Phase is written.

D3 Drive 3 Seeking: Set after the last byte of the Command Phase of a Seek or Recalibrate command is issued for drive 3. Cleared after reading the first byte in the Result Phase of the Sense Interrupt Command for this drive.

D2 Drive 2 Seeking: Same as above for drive 2.

D1 Drive 1 Seeking: Same as above for drive 1.

D0 Drive 0 Seeking: Same as above for drive 0.

DATA REGISTER (Read/Write)

This is the location through which all commands, data and status flow between the CPU and the DP8473. During the Command Phase the μP loads the controller's commands into this register based on the Status Register Request for Master and Data Direction bits. The Result Phase transfers the Status Registers and header information to the μP in the same fashion.

Register Description (Continued)

TABLE VI. Data Rate and Precompensation Programming Values

D1	D0**	DRVTYP Pin	Data Rate MFM (kb/s)	Normal Precomp* (ns)	Alternate Precomp* (ns)	FGND Pin Enabled	RPM/LC Pin Level
0	0	X	500	125	125	FGND500	High
0	1	0	250	125	250	FGND250	Low
0	1	1	300	208	208	FGND250	Low
1	0	0	250	125	250	FGND250	Low
1	0	1	250	125	250	FGND250	Low
1	1	0	1000	83	83	None	High
1	1	1	1000	83	83	None	Low

*Normal values when PUMP/PREN pin set low; Alternate values when PUMP/PREN pin set high.

**D0 and D1 are Data Rate Control Bits.

DRIVE CONTROL REGISTER (Write Only)

D7 Motor Enable 3: This controls the Motor for drive 3, MTR3. When 0 the output is high, when 1 the output is low. (Note this signal is not output to a pin on 48 pin DIP version.)

D6 Motor Enable 2: Same function as D7 except for drive 2's motor. (Note this signal is not brought out to a pin on DIP.)

D5 Motor Enable 1: This bit controls the Motor for drive 1's motor. When this bit is 0 the MTR1 output is high.

D4 Motor Enable 0: Same as D5 except for drive 0's motor.

D3 DMA Enable: When set to a 1 this enables the DRQ, DAK, INT pins. A zero disables these signals.

D2 Reset Controller: This bit when set to a 0 resets the controller, and when a 1 enables normal operation. It does not affect the Drive Control or Data Rate Registers which are reset only by a hardware reset.

D1–D0 Drive Select: These two pins are encoded for the four drive selects, and are gated with the motor enable lines, so that only one drive is selected when it's Motor Enable is active. (See Table V.)

DATA RATE REGISTER (Write Only)

D7–D2: Not used.

D1, D0 Data Rate Select: These bits set the data rate and the write precompensation values for the disk controller. After a hardware reset these bits are set to 10 (250 kb/s). They are encoded as shown in Table VI.

DISK CHANGED REGISTER (Read Only)

D7 Disk Changed: This bit is the latched complement of the Disk Changed input pin. If the DSKCHG input is low this bit is high.

D6–D0: These bits are reserved for use by the hard disk controller, thus during a read of this register, these bits are TRI-STATE.

Result Phase Status Registers

The Result Phase of a command contains bytes that hold status information. The format of these bytes are described below. Do not confuse these register bytes with the Main Status Register which is a read only register that is always available. The Result Phase status registers are read from the Data Register only during the Result Phase.

STATUS REGISTER 0 (ST0)

D7–D6 Interrupt Code:

00 = Normal Termination of Command.

01 = Abnormal Termination of Command. Execution of Command was started, but was not successfully completed.

10 = Invalid Command Issue. Command Issued was not recognized as a valid command.

11 = Ready changed state during the polling mode.

D5 Seek End: Seek or Recalibrate Command completed by the Controller. (Used during Sense Interrupt command.)

D4 Equipment Check: After a Recalibrate Command, Track 0 signal failed to occur. (Used during Sense Interrupt command.)

D3 Not Used: 0

D2 Head Address (at end of Execution Phase).

D1, D0 Drive Select (at end of Execution Phase).

00 = Drive 0 selected. 01 = Drive 1 selected.

10 = Drive 2 selected. 11 = Drive 3 selected.

STATUS REGISTER 1 (ST1)

D7 End of Track: Controller transferred the last byte of the last sector without the TC pin becoming active. The last sector is the End Of Track sector number programmed in the Command Phase.

D6 Not Used: 0

D5 CRC Error: If this bit is set and bit 5 of ST2 is clear, then there was a CRC error in the Address Field of the correct sector. If bit 5 of ST2 is set, then there was a CRC error in the Data Field.

D4 Over Run: Controller was not serviced by the μP soon enough during a data transfer in the Execution Phase.

Result Phase Status Registers (Continued)

TABLE VII. Maximum Time Allowed to Service an Interrupt or Acknowledge a DMA Request in Execution Phase

Data Rate	Time to Service
125	62.0 μs
250	30.0 μs
500	14.0 μs
1000	6.0 μs

Time from rising edge of DRQ or INT to trailing edge of $\overline{DAK}$ or $\overline{RD}$ or $\overline{WR}$.

D3 Not Used: 0

D2 No Data: Three possible problems: 1) Controller cannot find the sector specified in the Command Phase during the execution of a Read, Write, or Scan command. An address mark was found however so it is not a blank disk. 2) Controller cannot read any Address Fields without a CRC error during Read ID command. 3) Controller cannot find starting sector during execution of Read A Track command.

D1 Not Writable: Write Protect pin is active when a Write or Format command is issued.

D0 Missing Address Mark: If bit 0 of ST2 is clear then the disk controller cannot detect any Address Field Address Mark after two disk revolutions. If bit 0 of ST2 is set then the disk controller cannot detect the Data Field Address Mark.

STATUS REGISTER 2 (ST2)

D7 Not Used: 0

D6 Control Mark: Controller tried to read a sector which contained a deleted data address mark during execution of Read Data or Scan commands. Or, if a Read Deleted Data command was executed, a regular address mark was detected.

D5 CRC Error in Data Field: Controller detected a CRC error in the Data Field. Bit 5 of ST1 is also set.

D4 Wrong Track: Only set if desired sector not found, and the track number recorded on any sector of the current track is different from that stored in the Track Register.

D3 Scan Equal Hit: "Equal" condition satisfied during any Scan Command.

D2 Scan Not Satisfied: Controller cannot find a sector on the track which meets the desired condition during Scan Command.

D1 Bad Track: Only set if the desired sector is not found, and the track number recorded on any sector on the track is different from that stored in the Track Register and the recorded track number is FF.

D0 Missing Address Mark in Data Field: Controller cannot find the Data Field Address Mark during Read/Scan command. Bit 0 of ST1 is also set.

STATUS REGISTER 3 (ST3)

D7 Not Used: 0

D6 Write Protect Status

D5 Not Used: 1

D4 Track 0 Status

D3 Not Used: 0

D2 Head Select Status

D1, D0 Drive Selected:

00 = Drive 0 selected. 01 = Drive 1 selected.

10 = Drive 2 selected. 11 = Drive 3 selected.

Processor Software Interface

Bytes are transferred to and from the disk controller in different ways for the different phases in a command.

COMMAND SEQUENCE

The disk controller can perform various disk transfer, and head movement commands. Most commands involve three separate phases.

Command Phase: The μP writes a series of bytes to the Data Register. These bytes indicate the command desired and the particular parameters required for the command. All the bytes must be written in the order specified in the Command Description Table. The Execution Phase starts immediately after the last byte in the Command Phase is written. Prior to performing the Command Phase, the Drive Control and Data Rate Registers should be set.

Execution Phase: The disk controller performs the desired command. Some commands require the μP to read or write data to or from the Data Register during this time. Reading data from a disk is an example of this.

Result Phase: The μP reads a series of bytes from the data register. These bytes indicate whether the command executed properly and other pertinent information. The bytes are read in the order specified in the Command Description Table.

A new command may be initiated by writing the Command Phase bytes after the last bytes required from the Result Phase have been read. If the next command requires selecting a different drive or changing the data rate the Drive Control and Data Rate Registers should be updated. If the command is the last command, then the software should deselect the drive. *(Note as a general rule the operation of the controller core is independent of how the μP updates the Drive Control and Data Rate Registers. The software must ensure that manipulation of these registers is coordinated with the controller operation.)*

During the Command Phase and the Result Phase, bytes are transferred to and from the Data Register. The Main Status Register is monitored by the software to determine when a data transfer can take place. Bit 6 of the Main Status Register must be clear and bit 7 must be set before a byte can be written to the Data Register during the Command Phase. Bits 6 and 7 of the Main Status Register must both be set before a byte can be read from the Data Register during the Result Phase.

If there is information to be transferred during the Execution Phase, there are three methods that can be used. The DMA mode is used if the system has a DMA controller. This allows the μP to do other things during the Execution Phase data transfer. If DMA is not used, an interrupt can be issued for each byte transferred during the Execution Phase. If interrupts are not used, the Main Status Register can be polled to indicate when a byte transfer is required.

Processor Software Interface (Continued)

DMA MODE

If the DMA mode is selected, a DMA request will be generated in the Execution Phase when each byte is ready to be transferred. To enable DMA operations during the Execution Phase, the DMA mode bit in the Specify Command must be enabled, and the DMA signals must be enabled in the Drive Control Register. The DMA controller should respond to the DMA request with a DMA acknowledge and a read or write strobe. The DMA request will be cleared by the active edge of the DMA acknowledge. After the last byte is transferred, an interrupt is generated, indicating the beginning of the Result Phase. During DMA operations the Chip Select input must be held high. TC is asserted to terminate an operation. Due to the internal gating TC is only recognized when the $\overline{DAK}$ input is low.

INTERRUPT MODE

If the non-DMA mode is selected, an interrupt will be generated in the Execution Phase when each byte is ready to be transferred. The Main Status Register should be read to verify that the interrupt is for a data transfer. Bits 5 and 7 of the Main Status Register will be set. The interrupt will be cleared when the byte is transferred to or from the Data Register. The μP should transfer the byte within the time allotted by Table VII. If the byte is not transferred within the time allotted, an Overrun Error will be indicated in the Result Phase when the command terminates at the end of the current sector.

An interrupt will also be generated after the last byte is transferred. This indicates the beginning of the Result Phase. Bits 7 and 6 of the Main Status Register will be set and bit 5 will be clear. This interrupt will be cleared by reading the first byte in the Result Phase.

SOFTWARE POLLING

If the non-DMA mode is selected and interrupts are not suitable, the μP can poll the Main Status Register during the Execution Phase to determine when a byte is ready to be transferred. In the non-DMA mode, bit 7 of the Main Status Register reflects the state of the interrupt pin. Otherwise, the data transfer is similar to the Interrupt Mode described above.

Command Description Table

READ DATA

Command Phase

MT	MFM	SK	0	0	1	1	0
IPS	X	X	X	X	HD	DR1	DR0
Track Number							
Drive Head Number							
Sector Number							
Number of Bytes per Sector							
End of Track Sector Number							
Intersector Gap Length							
Data Length							

Note 1

Result Phase

Status Register 0
Status Register 1
Status Register 2
Track Number
Head Number
Sector Number
Bytes/Sector

READ ID

Command Phase

0	MFM	0	0	1	0	1	0
X	X	X	X	X	HD	DR1	DR0

Result Phase

Status Register 0
Status Register 1
Status Register 2
Track Number
Head Number
Sector Number
Bytes/Sector

FORMAT A TRACK

Command Phase

0	MFM	0	0	1	1	0	1
X	X	X	X	X	HD	DR1	DR0
Number of Bytes per Sector							
Number of Sectors per Track							
Intersector Gap Length							
Data Pattern							

Result Phase

Status Register 0
Status Register 1
Status Register 2
Track Number
Head Number
Sector Number
Bytes/Sector

Command Description Table (Continued)

READ DELETED DATA

Command Phase

MT	MFM	SK	0	1	1	0	0
IPS	X	X	X	X	HD	DR1	DR0
Track Number							
Drive Head Number							
Sector Number							
Number of Bytes per Sector							
End of Track Sector Number							
Intersector Gap Length							
Data Length							

Result Phase

Status Register 0
Status Register 1
Status Register 2
Track Number
Head Number
Sector Number
Bytes/Sector

WRITE DATA

Command Phase

MT	MFM	0	0	0	1	0	1
IPS	X	X	X	X	HD	DR1	DR0
Track Number							
Drive Head Number							
Sector Number							
Number of Bytes per Sector							
End of Track Sector Number							
Intersector Gap Length							
Data Length							

Result Phase

Status Register 0
Status Register 1
Status Register 2
Track Number
Head Number
Sector Number
Bytes/Sector

SCAN EQUAL

Command Phase

MT	MFM	SK	1	0	0	0	1
IPS	X	X	X	X	HD	DR1	DR0
Track Number							
Drive Head Number							
Sector Number							
Number of Bytes per Sector							
End of Track Sector Number							
Intersector Gap Length							
Sector Step Size							

Result Phase

Status Register 0
Status Register 1
Status Register 2
Track Number
Head Number
Sector Number
Bytes/Sector

READ A TRACK

Command Phase

0	MFM	SK	0	0	0	1	0
IPS	X	X	X	X	HD	DR1	DR0
Track Number							
Drive Head Number							
Sector Number							
Number of Bytes per Sector							
End of Track Sector Number							
Intersector Gap Length							
Data Length							

Result Phase

Status Register 0
Status Register 1
Status Register 2
Track Number
Head Number
Sector Number
Bytes/Sector

WRITE DELETED DATA

Command Phase

MT	MFM	0	0	1	0	0	1
IPS	X	X	X	X	HD	DR1	DR0
Track Number							
Drive Head Number							
Sector Number							
Number of Bytes per Sector							
End of Track Sector Number							
Intersector Gap Length							
Data Length							

Result Phase

Status Register 0
Status Register 1
Status Register 2
Track Number
Head Number
Sector Number
Bytes/Sector

SCAN LOW OR EQUAL

Command Phase

MT	MFM	SK	1	1	0	0	1
IPS	X	X	X	X	HD	DR1	DR0
Track Number							
Drive Head Number							
Sector Number							
Number of Bytes per Sector							
End of Track Sector Number							
Intersector Gap Length							
Sector Step Size							

Result Phase

Status Register 0
Status Register 1
Status Register 2
Track Number
Head Number
Sector Number
Bytes/Sector

Command Description Table (Continued)

SCAN HIGH OR EQUAL

Command Phase

MT	MFM	SK	1	1	1	0	1
IPS	X	X	X	X	HD	DR1	DR0
Track Number							
Drive Head Number							
Sector Number							
Number of Bytes per Sector							
End of Track Sector Number							
Intersector Gap Length							
Sector Step Size							

Result Phase

Status Register 0
Status Register 1
Status Register 2
Track Number
Head Number
Sector Number
Bytes/Sector

SEEK

Command Phase

0	0	0	0	1	1	1	1
X	X	X	X	X	X	DR1	DR0
New Track Number							
MSB of Track				0	0	0	0

Note 2

RECALIBRATE

Command Phase

0	0	0	0	0	1	1	1
0	0	0	0	0	0	DR1	DR0

SENSE INTERRUPT

Command Phase

0	0	0	0	1	0	0	0

Result Phase

Status Register 0							
Present Track Number (PTN)							
MSN PTN				0	0	0	0

Note 2

SENSE DRIVE STATUS

Command Phase

0	0	0	0	0	1	0	0
X	X	X	X	X	HD	DR1	DR0

Result Phase

Status Register 3

SPECIFY

Command Phase

0	0	0	0	0	0	1	1
Step Rate Time				Motor Off Time			
Motor On Time							DMA

MODE

Command Phase

0	0	0	0	0	0	0	1
TMR	IAF	IPS	0	LW	PR	1	ETR
0	0	0	0	0	0	0	0
1	1	0	WLD	Head Settle			
0	0	0	0	0	RG	0	PU

Note 3

SET TRACK

Command Phase

0	$\overline{R}$/W	1	0	0	0	0	1
0	0	1	1	0	MSB	DR1	DR0
New Track Number							

Result Phase

Value

Note 3

INVALID COMMAND

Command Phase

Invalid Op Codes

Result Phase

Status Register 0

Note 1: The IPS bit is only enabled if the IPS bit in the mode command is set. Otherwise this bit is a don't care.

Note 2: Shaded byte only written or read if the extended track range mode is enabled in the Mode Command (ET) = 1.

Note 3: These commands are additional enhanced commands.

Note: Mnemonic Definitions

X = DON'T CARE

MFM = Data Encoding Scheme

MSN PTN = Most Significant Nibble Present Track Number

MT = Multi-Track

IPS = Implied Seek (In individual commands this bit is a don't care unless the IPS bit in the mode command is set.)

SK = Skip Sector

HD = Head Number

DRn = Drive to Select (encoded)

TMR = Motor/Head Timer Mode

IAF = Index Address Field

LW PR = Low Power Mode

ETR = Extended Track Range

WLD = Wildcard in Scan

RG = Enables the Read Gate Input on the DSKCHG pin for the Data Separator.

PU = Enables Charge Pump PUMP signal to be output on the PUMP/PREN pin.

MSB = Selects whether the most significant or least significant byte of the track is read. 1 = MSB.

$\overline{R}$/W = Selects whether the track is written or read (Read = 0, Write = 1).

Command Description

READ DATA

The Read Data op-code is written to the data register followed by 8 bytes as specified in the Command Description Table. After the last byte is written, the controller starts looking for the correct sector header. Once the sector is found the controller sends the data to the μP. After one sector is finished, the Sector Number is incremented by one and this new sector is searched for. If MT (Multi-Track) is set, both sides of one track can be read. Starting on side zero, the sectors are read until the sector number specified by End of Track Sector Number is reached. Then, side one is read starting with sector number one.

In DMA mode the Read Data command continues to read until the TC pin is set. This means that the DMA controller should be programmed to transfer the correct number of bytes. TC could be controlled by the μP and be asserted when enough bytes are received. An alternative to these methods of stopping the Read Data command is to program the End of Track Sector Number to be the last sector number that needs to be read. The controller will stop reading the disk with an error indicating that it tried to access a sector number beyond the end of the track.

The Number of Data Bytes per Sector parameter is defined in Table VIII. If this is set to zero then the Data Length parameter determines the number of bytes that the controller transfers to the μP. If the data length specified is smaller than 128 the controller still reads the entire 128 byte sector and checks the CRC, though only the number of bytes specified by the Data Length parameter are transferred to the μP. Data Length should not be set to zero. If the Number of Bytes per Sector parameter is not zero, the Data Length parameter has no meaning and should be set to FF (hex).

If the Implied Seek Mode is enabled by both the Mode command and the IPS bit in this command, a Seek will be performed to the track number specified in the Command Phase. The controller will also wait the Head Settle time if the implied seek is enabled.

After all these conditions are met, the controller searches for the specified sector by comparing the track number, head number, sector number, and number bytes/sector given in the Command Phase with the appropriate bytes read off the disk in the Address Fields.

If the correct sector is found, but there is a CRC error in the Address Field, bit 5 of ST1 (CRC Error) is set and an abnormal termination is indicated. If the correct sector is not found, bit 2 of ST1 (No Data) is set and an abnormal termination is indicated. In addition to this, if any Address Field track number is FF, bit 1 of ST2 (Bad Track) is set or if any Address Field track number is different from that specified in the Command Phase, bit 4 of ST2 (Wrong Track) is set.

After finding the correct sector, the controller reads that Data Field. If a Deleted Data Mark is found and the SK bit is set, the sector is not read, bit 6 of ST2 (Control Mark) is set, and the next sector is searched for. If a deleted data mark is found and the SK bit is not set, the sector is read, bit 6 of ST2 (Control Mark) is set, and the read terminates with a normal termination. If a CRC error is detected in the Data Field, bit 5 is set in both ST1 and ST2 (CRC Error) and an abnormal termination is indicated.

If no problems occur in the read command, the read will continue from one sector to the next in logical order (not physical order) until either TC is set or an error occurs.

If a disk has not been inserted into the disk drive, there are many opportunities for the controller to appear to hang up. It does this if it is waiting for a certain number of disk revolutions for something. If this occurs, the controller can be forced to abort the command by writing a byte to the Data register. This will place the controller into the Result Phase.

TABLE VIII. Sector Size Selection

Bytes/Sector Code	Number of Bytes in Data Field
0	128
1	256
2	512
3	1024
4	2048
5	4096
6	8192

An interrupt will be generated when the Execution Phase of the Read Data command terminates. The values that will be read back in the Result Phase are shown in Table IX. If an error occurs, the result bytes will indicate the sector being read when the error occurred.

READ DELETED DATA

This command is the same as the Read Data command except for its treatment of a Deleted Data Mark. If a Deleted

TABLE IX. Result Phase Termination Values with No Error

MT	HD	Last Sector	ID Information at Result Phase			
			Track	Head	Sector	B/S
0	0	< EOT	NC	NC	S+1	NC
0	0	= EOT	T+1	NC	1	NC
0	1	< EOT	NC	NC	S+1	NC
0	1	= EOT	T+1	NC	1	NC
1	0	< EOT	NC	NC	S+1	NC
1	0	= EOT	NC	1	1	NC
1	1	< EOT	NC	NC	S+1	NC
1	1	= EOT	T+1	0	1	NC

EOT = End of Track Sector Number from Command Phase

NC = No Change in Value

S = Sector Number last operated on by controller

T = Track Number programmed in Command Phase

Command Description (Continued)

Data Mark is read, the sector is read normally. If a Regular Data Mark is found and the SK bit is set, the sector is not read, bit 6 of ST2 (Control Mark) is set, and the next sector is searched for. If a Regular Data Mark is found and the SK bit is not set, the sector is read, bit 6 of ST2 (Control Mark) is set, and the read terminates with a normal termination.

WRITE DATA

The Write Data command is very similar to the Read Data command except that data is transferred from the μP to the disk rather than the other way around. If the controller detects the Write Protect signal, bit 1 of ST1 (Not Writable) is set and an abnormal termination is indicated.

WRITE DELETED DATA

This command is the same as the Write Data Command except a Deleted Data Mark is written at the beginning of the Data Field instead of the normal Data Mark.

READ A TRACK

This command is similar to the Read Data command except for the following. The controller starts at the index hole and reads the sectors in their physical order, not their logical order.

Even though the controller is reading sectors in their physical order, it will still perform a comparison of the header ID bytes with the Data programmed in the Command Phase. The exception to this is the sector number. Internally, this is initialized to a one, and then incremented for each successive sector read. Whether or not the programmed Address Field matches that read from the disk, the sectors are still read in their physical order. If a header ID comparison fails, bit 2 of ST1 (No Data) is set, but the operation will continue. If there is a CRC error in the Address Field or the Data Field, the read will also continue.

The command will terminate when it has read the number of sectors programmed in the EOT parameter.

READ ID

This command will cause the controller to read the first Address Field that it finds. The Result Phase will contain the header bytes that are read. There is no data transfer during the Execution Phase of this command. An interrupt will be generated when the Execution Phase is completed.

FORMAT A TRACK

This command will format one track on the disk. After the index hole is detected, data patterns are written on the disk including all gaps, address marks, Address Fields, and Data Fields. The exact details of the number of bytes for each field is controlled by the parameters given in the Format A Track command, and the IAF (Index Address Field) bit in the Mode command. The Data Field consists of the Fill Byte specified in the command, repeated to fill the entire sector.

To allow for flexible formatting, the μP must supply the four Address Field bytes (track, head, sector, number of bytes) for each sector formatted during the Execution Phase. In other words, as the controller formats each sector, it will request four bytes through either DMA requests or interrupts. This allows for non-sequential sector interleaving. Some typical values for the programmable GAP size are shown in Table X.

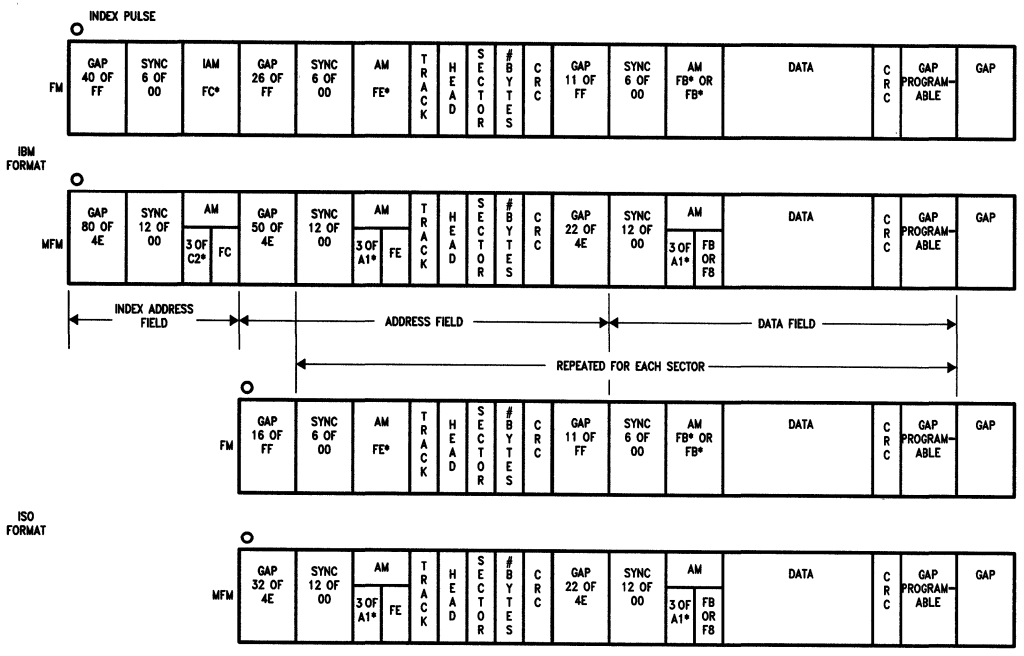
The Format Command terminates when the index hole is detected a second time, at which point an interrupt is generated. Only the first three status bytes in the Result Phase are significant.

TABLE X. Gap Length for Various Sector Sizes and Disk Types

Mode	Sector Size	Sector Code	EOT	Gap	Format Gap
8" Drives (360 RPM, 500 kb/s)					
FM	128	00	1A	07	1B
	256	01	0F	0E	2A
	512	02	08	1B	3A
	1024	03	04	47	8A
	2048	04	02	C8	FF
	4096	05	01	C8	FF
MFM	256	01	1A	0E	36
	512	02	0F	1B	54
	1024	03	08	35	74
	2048	04	04	99	FF
	4096	05	02	C8	FF
	8192	06	01	C8	FF
5.25" Drives (300 RPM, 250 kb/s)					
FM	128	00	12	07	09
	128	00	10	10	19
	256	01	08	18	30
	512	02	04	46	87
	1024	03	02	C8	FF
	2048	04	01	C8	FF
MFM	256	01	12	0A	0C
	256	01	10	20	32
	512	02	08	2A	50
	1024	03	04	80	F0
	2048	04	02	C8	FF
	4096	05	01	C8	FF
3.5" Drives (300 RPM, 250 kb/s)					
FM	128	00	0F	07	1B
	256	01	09	0E	2A
	512	02	05	1B	3A
MFM	256	01	0F	0E	36
	512	02	09	1B	54
	1024	03	05	35	74

Note: Format Gap is the gap length used only for the Format command.

Command Description (Continued)



TL/F/9384–10

Notes:

FE* = Data pattern of FE, Clock pattern of C7
FC* = Data pattern of FC, Clock pattern of D7
FB* = Data pattern of FB, Clock pattern of C7
F8* = Data pattern of F8, Clock pattern of C7
A1* = Data pattern of A1, Clock pattern of 0A
C2* = Data pattern of C2, Clock pattern of 14

All byte counts in decimal.
All byte values in hex.
CRC uses standard polynomial $x^{16} + x^{12} + x^5 + 1$.

FIGURE 6. IBM and ISO Formats Supported by the Format Command

SCAN COMMANDS

The Scan Commands allow data read from the disk to be compared against data sent from the μP. There are three Scan Commands to choose from:

Scan Equal	Disk Data = μP Data
Scan Less Than or Equal	Disk Data ≤ μP Data
Scan Greater Than or Equal	Disk Data ≥ μP Data

Each sector is interpreted with the most significant bytes first. If the Wildcard mode is enabled from the Mode command, an FF(hex) from either the disk or the μP is used as a don't care byte that will always match equal. After each sector is read, if the desired condition has not been met, the next sector is read. The next sector is defined as the current sector number plus the Sector Step Size specified. The Scan command will continue until the scan condition has been met, or the End of Track Sector Number has been reached, or if TC is asserted.

If the SK bit is set, sectors with deleted data marks are ignored. If all sectors read are skipped, the command will terminate with D3 of ST2 set (Scan Equal Hit). The result phase of the command is shown in Table XI.

TABLE XI. Scan Command Termination Values

Command	Status Register 2		Conditions
	D2	D3	
Scan Equal	0	1	Disk = μP
	1	0	Disk ≠ μP
Scan Low or Equal	0	1	Disk = μP
	0	0	Disk < μP
	1	0	Disk > μP
Scan High or Equal	0	1	Disk = μP
	0	0	Disk > μP
	1	0	Disk < μP

5

Command Description (Continued)

SEEK

There are two ways to move the disk drive head to the desired track number. Method One is to enable the Implied Seek Mode. This way each individual Read or Write command will automatically move the head to the track specified in the command.

Method Two is using the Seek Command. During the Execution Phase of the Seek Command, the track number to seek to is compared with the present track number and a step pulse is produced to move the head one track closer to the desired track number. This is repeated at the rate specified by the Specify Command until the head reaches the correct track. At this point an interrupt is generated and a Sense Interrupt Command is required to clear the interrupt.

During the Execution Phase of the Seek Command the only indication via software that a Seek Command is in progress is bits 0–3 (Drive Busy) of the Main Status Register. Bit 4 of the Main Register (Controller Busy) is not set. While the internal microengine is capable of multiple seeks on 2 or more drives at the same time since the drives are selected via the Drive Control Register in software, software should ensure that only one drive is seeking at one time. No other command except the Sense Interrupt Command should be issued while a Seek Command is in progress.

If the extended track range mode is enabled, a fourth byte should be written in the Command Phase to indicate the four most significant bits of the desired track number. Otherwise, only three bytes should be written.

RECALIBRATE

The Recalibrate Command is very similar to the Seek Command. It is used to step a drive head out to track zero. Step pulses will be produced until the track zero signal from the drive becomes true. If the track zero signal does not go true before 77 step pulses are issued, an error is generated. If the extended track range mode is enabled, an error is not generated until 3917 pulses are issued.

Recalibrations on more than one drive at a time should not be issued for the same reason as explained in the Seek Command. No other command except the Sense Interrupt Command should be issued while a Recalibrate Command is in progress.

SENSE INTERRUPT STATUS

An interrupt is generated by the controller when any of the following conditions occur:

1. Upon entering the Result Phase of:
   a. Read Data Command
   b. Read Deleted Data Command
   c. Write Data Command
   d. Write Deleted Data Command
   e. Read a Track Command
   f. Read ID Command
   g. Format Command
   h. Scan Commands
2. During data transfers in the Execution Phase while in the Non-DMA mode
3. Internal Ready signal changes state (only occurs immediately after a hardware or software reset).
4. Seek or Recalibrate Command termination

An interrupt generated for reasons 1 and 2 above occurs during normal command operations and are easily discernible by the μP. During an execution phase in Non-DMA Mode, bit 5 (Execution Mode) in the Main Status Register is set to 1. Upon entering Result Phase this bit is set to 0. Reasons 1 and 2 do not require the Sense Interrupt Status command. The interrupt is cleared by reading or writing information to the data register.

Interrupts caused by reasons 3 and 4 are identified with the aid of the Sense Interrupt Status Command. This command resets the interrupt when the command byte is written. Use bits 5, 6 and 7 of ST0 to identify the cause of the interrupt as shown in Table XII.

TABLE XII. Status Register 0 Termination Codes

Status Register 0			Cause
Interrupt Code		Seek End	
D7	D6	D5	
1	1	0	Internal Ready Went True
0	0	1	Normal Seek Termination
0	1	1	Abnormal Seek Termination

TABLE XIII. Step, Head Load and Unload Timer Definitions (500 kb/s MFM)

Timer	Mode 1		Mode 2		Unit
	Value	Range	Value	Range	
Step Rate	(16 − N)	1–16	(16 − N)	1–16	ms
Head Unload	N × 16	0–240	N × 512	0–7680	ms
Head Load	N × 2	0–254	N × 32	0–4064	ms

Issuing a Sense Interrupt Status Command without an interrupt pending is treated as an invalid command.

If the extended track range mode is enabled, a third byte should be read in the Result Phase which will indicate the four most significant bits of the Present Track Number. Otherwise, only two bytes should be read.

SPECIFY

The Specify Command sets the initial values for each of the three internal timers. The timer programming values are shown in Table XIII.

The Head Load and Head Unload timers are artifacts of the μPD765A. These timers determine the delay from loading the head until a read or write command is started, and unloading the head sometime after the command was completed. Since the DP8473's head load signal is now the software controlled Motor lines in the Drive Control Register, these timers only provide some delay from the initiation of a command until it is actually started. These times can be extended by setting the TMR bit in the Mode Command.

The Step Rate Time defines the time interval between adjacent step pulses during a Seek, Implied Seek, or Recalibrate Command.

The times stated in the table are affected by the Data Rate. The values in the table are for 500 kb/s MFM (250 kb/s FM) and 1 Mb/s MFM (500 kb/s FM). For a 300 kb/s MFM data rate (150 kb/s FM) these values should be multiplied by 1.6667, and for 250 kb/s MFM (125 kb/s FM) these values should be doubled.

The choice of DMA or Non-DMA operation is made by the NON-DMA bit. When this bit is 1 then Non-DMA mode is selected, and when this bit is 0, the DMA mode is selected.

This command does not generate an interrupt.

Command Description (Continued)

LOW PWR (LOW PoWeR mode)

SENSE DRIVE STATUS

This two byte command obtains the status of a disk drive. Status Register 3 is returned in the result phase and contains the drive status. This command does not generate an interrupt.

MODE

This command is used to select the special features of the controller. The bits for the command phase bytes are shown in the command description table, and their function is described below. The defaults after a hardware or software reset are shown by the "bullets" to the left of each item.

- **TMR=0 (motor TiMeR):** Timers for motor on and motor off are defined for Mode 1. (See Specify Command)
  TMR=1: Timers for motor on and motor off are defined for Mode 2. (See Specify Command)

LW PR (LoW PoweR)

- 00 Completely disable the low power mode. (default)
  01 Go into low power mode 500 ms after the head unload timer times out.
  10 Go into low power mode now.
  11 Not Used.
- **IAF=0 (Index Address Format):** The controller will format tracks with the Index Address Field included. (IBM Format)
  IAF=1: The controller will format tracks without including the Index Address Mark Field. (ISO Format)
- **IPS=0 (ImPlied Seek):** The implied seek bit in the command is ignored.
  IPS=1: The implied seek bit in the command is enabled so that if the bit is set in the command, a Seek will be performed automatically.
- **ETR=0 (Extended Track Range):** Header format is the IBM System 34 (double density) or System 3740 (single density).
  ETR=1: Header format is the same as above but there are 12 bits of track number. The MSB's of the track number are in the upper four bits of the head number byte.
- **WLD=0 (scan WiLD card):** An FF(hex) from either the μP or the disk during a Scan Command is interpreted as a wildcard character that will always match true.
  WLD=1: The Scan commands do not recognize FF(hex) as a wildcard character.
  Head Settle: Time allowed for head to settle after an Implied Seek. Time = N $\times$ 4 ms, (0 ms–60 ms). (Based on 500 kb/s and 1 Mb/s MFM data rates. Double for 250 kb/s.)
  PU (PUMP Pulse Output): When set enables a signal that indicates when the Data Separator's charge pump is making a phase correction. This is a series of pulses. This signal is output on the PUMP/PREN pin when this bit is set.
  This is intended as a test mode to aid in evaluation of the Data Separator. (Default mode is off)
  RG (Read Gate): Like the PUMP output, when this bit is set it enables a pin (the DSKCHG pin) to act as an external Read Gate signal for the Data Separator. This is intended as a test mode to aid in evaluation of the Data Separator. (Default mode is off)

SET TRACK

This command is used to inspect or change the value of the internal Present Track Register. This could be useful for recovery from disk mis-tracking errors, where the real current track could be read through the Read ID command and then the Set Track Command can set the internal present track register to the correct value.

The first byte of the command contains the command opcode and the R/W bit. If the R/W bit is low, a track register is to be read. In this case, the result phase contains the value in the internal register specified, and the third byte of the command is a dummy byte.

If the R/W bit is high, data is written to a track register. In this case the 3rd byte of the command phase is forced into the specified internal register, and the result phase contains the new byte value written.

The particular track register chosen to operate on is determined by the least significant 3 bits of the second byte of the command. The two LSB's select the drive (DR1, DR0), and the next bit (MSB) determines whether the least significant byte (MSB=0) or the most significant byte (MSB=1) of the track register is to be read/written. When not in the extended track range mode, only the LSB track register need be updated. In this instance, the MSB bit is set to 0.

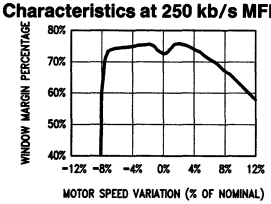
This command does not generate an interrupt.

INVALID COMMAND

If an invalid command (i.e., a command not defined) is received by the controller, the controller will respond with ST0 in the Result Phase. The Controller does not generate an interrupt during this condition. Bits 6 and 7 in the Main Status Register are both set to one's indicating to the processor that the Controller is in the Result Phase and the contents of ST0 must be read. When the system reads ST0 it will find an 80(hex) indicating an invalid command was received.

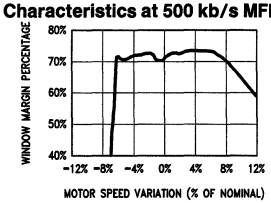
Typical Performance Characteristics

Typical Window Margin Performance Characteristics at 250 kb/s MFM



TL/F/9384–18

Typical Window Margin Performance Characteristics at 500 kb/s MFM



TL/F/9384–19

Absolute Maximum Ratings (Notes 1 and 2)

If Military/Aerospace specified devices are required, please contact the National Semiconductor Sales Office/Distributors for availability and specifications.

Supply Voltage (V_{CC})	−0.5V to +7V
DC Input Voltage (V_{IN})	−0.5V to V_{CC} + 0.5V
DC Output Voltage (V_{OUT})	−0.5V to V_{CC} + 0.5V
Storage Temperature Range (T_{STG})	−65°C to +165°C
Package Power Dissipation (P_D)	750 mW
Lead Temperature (T_L) (Soldering, 10 seconds)	260°C
$\lvert V_{CC}-V_{CCA}\rvert$	0.6V

Operating Conditions

	Min	Max	Units
Supply Voltage (V_{CC})	4.5	5.5	V
Operating Temperature (T_A)	0	+70	°C
ESD Tolerance: C_{ZAP} = 100 pF, R_{ZAP} = 1.5 kΩ (Note 5)	1500		V

DC Electrical Characteristics V_{CC} = 5V ±10% unless otherwise specified (Note 3)

Symbol	Parameter	Conditions	Min	Max	Units
V_{IH}	High Level Input Voltage	(except OSC2/CLK)	2.0		V
V_{IL}	Low Level Input Voltage	(except OSC2/CLK)		0.8	V
I_{IN}	Input Current (except OSC pins)	V_{IN} = V_{CC} or GND		±1.0	μA
I_{CCA}	Average V_{CCA} Supply Current	V_{IN} = 2.4V or 0.5V, I_O = 0 mA (Note 4)		20.0	mA
	Quiescent V_{CCA} Supply Current in Low Power Mode	V_{IN} = V_{CC} or GND, I_O = 0 mA (Note 4)		400	μA
I_{CC}	Average V_{CC} Supply Current	V_{IN} = 2.4V or 0.5V, I_O = 0 mA (Note 4)		20.0	mA
	Quiescent V_{CC} Supply Current in Low Power Mode	V_{IN} = V_{CC} or GND, I_O = 0 mA (Note 4)		2	mA
OSCILLATOR PINS (OSC2/CLK)					
I_{OSC}	OSC2 Input Current (OSC1 = GND)	V_{IN} = V_{CC} or GND	±1.6		mA
V_{IH}	OSC2 High Level Input Voltage	OSC1 = GND	2.4		V
V_{IL}	OSC2 Low Level Input Voltage	OSC1 = GND		0.4	V
MICROPROCESSOR INTERFACE PINS (D0–D7, INT, DAK, TC, DRQ, RD, WR, CS, A0–A3)					
V_{OH}	High Level Output Voltage	I_{OUT} = −20 μA I_{OUT} = −4.0 mA	V_{CC} − 0.1 3.5		V V
V_{OL}	Low Level Output Voltage	I_{OUT} = 20 μA I_{OUT} = 12 mA		0.1 0.4	V V
I_{OZ}	Output TRI-STATE® Leakage Current	V_{OUT} = V_{CC} or GND		±10.0	μA
DISK DRIVE INTERFACE PINS (MTR0–3, DR0–3, WDATA, WGATE, RDATA, DIR, HDSEL, TRK0, WRTPRT, RPM, STEP, DSKCHG, INDEX)					
V_H	Input Hysteresis		250 Typical		mV
V_{OL}	Low Level Output Voltage	I_{OUT} = 48 mA		0.4	V
I_{LKG}	Output High Leakage Current	V_{OUT} = V_{CC} or GND		±100	μA
V_{IH}	High Level Input Voltage		2.2		V
V_{IL}	Low Level Input Voltage			0.8	V

Note 1: Absolute Maximum Ratings are those values beyond which damage to the device may occur.

Note 2: Unless otherwise specified all voltages are referenced to ground.

Note 3: These DC Electrical Characteristics are measured statically, and not under dynamic conditions.

Note 4: I_{CC} is measured with a 0.1 μF supply decoupling capacitor to ground.

Note 5: Value based on test complying with NSC SOP5-028 human body model ESD testing using the ETS-910 tester.

Phase Locked Loop Characteristics V_{CC} = 5V ±10%, F_{XTAL} = 24 MHz unless otherwise specified

Symbol	Parameter	Conditions	Typ		Units
V_{REF}	SETCUR Pin Reference Voltage	R_1 = 5.6 kΩ, V_{CC} = 5V	1.1		V
K_{VCO}	VCO Gain (Note 5)	t_{DATA} = 1 μs ±10%	25		Mrad/s/V
R_1	Recommended Pump Resistor Range		3–12		kΩ
$K_{P(UP)}$	Charge Pump Up Current Gain ($I_{REF}/I_{P(UP)}$) (Note 6)	R_1 = 5.6 kΩ	2.50		(none)
$K_{P(DWN)}$	Charge Pump Down Current Gain ($I_{REF}/I_{P(DWN)}$) (Note 6)	R_1 = 5.6 kΩ	2.25		(none)
K_{PLL}	Internal Phase Locked Loop Gain (Note 7)	(R_1 = 5.6 kΩ) Pump Up Pump Down	 75 70		 Mrad Mrad
T_{SW}	Static Window (Note 8)		**Early**	**Late**	
		(R_1 = 5.6 kΩ) 250 kb/s 500 kb/s 1.0 Mb/s	 1075 530 259	 872 440 234	 ns ns ns
T_{DW}	Dynamic Window Margin	(Note)	70		%

Note: Measurements made with a repeating 'DB6" data pattern with reverse write precompensation, using recommended filter values for the configuration shown in Figure 4c. 25°C, 5.0V, 0% MSV.

Note 5: The VCO gain is measured at the 1.0 Mb/s data rate by forcing the data period over a range from 900 ns to 1100 ns, and measuring the resulting voltage on the filter pin. The best straight line gain is fit to the measured points.

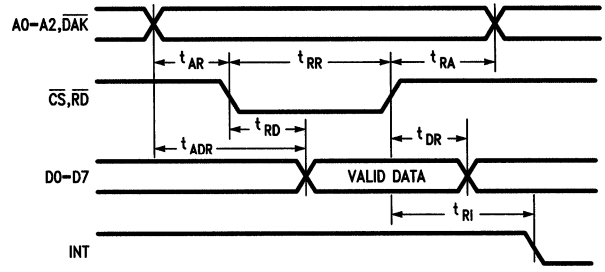
Note 6: This is the current gain of the charge pump, which is defined as the output current divided by the current through R_1.

Note 7: This is the product of: $V_{REF} \times K_P \times K_{VCO}$. The total variation in this specification indicates the total loop gain variation contributed by the internal circuitry. The K_{VCO} portion of this specification is measured at the 1.0 Mb/s data rate by forcing the data period over a range of 900 ns to 1100 ns, and measuring the resultant K_{VCO}. K_P is measured by forcing the Filter pin to 2.1V and measuring the ratio of the charge pump current over the input current.

Note 8: The DP8473 is guaranteed to correctly decode a single shifted clock pulse at the end of a long series of non-shifted preamble bits as long as the single shifted pulse is shifted less than the amount specified in T_{SW}. The length of the preamble is long enough for the PLL to lock. The filter components used are those in Table II.

AC Electrical Characteristics

MICROPROCESSOR READ TIMING

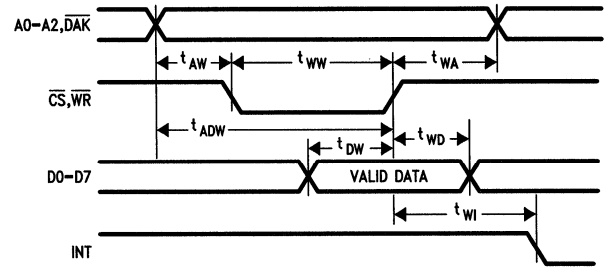


TL/F/9384–11

Symbol	Parameter	Min	Max	Units
t_{AR}	Address Valid prior to Read Strobe	10		ns
t_{RA}	Address Hold from Read Strobe	0		ns
t_{RR}	Read Strobe Width	75		ns
t_{RD}	Read Strobe and Chip Select to Data Valid		75	ns
t_{ADR}	Address Valid to Read Data		85	ns
t_{DR}	Data Hold from Read Strobe to High Impedance (TRI-STATE Note)	5	60	ns
t_{RI}	Clear INT from Read Strobe		65	ns

TRI-STATE Note: This limit includes the RC delay inherent in our test method. This signal will typically turn off within 15 ns, enabling other devices to drive this signal with no contention.

MICROPROCESSOR WRITE TIMING

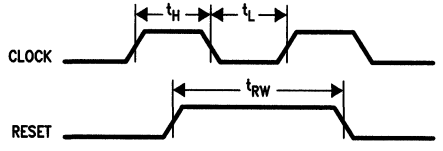


TL/F/9384–12

Symbol	Parameter	Min	Max	Units
t_{AW}	Address Valid to Leading Edge of Write Strobe	10		ns
t_{WA}	Address Hold from Write Strobe	0		ns
t_{WW}	Write Strobe Width	25		ns
t_{ADW}	Address Valid to Trailing Edge of Write Strobe	35		ns
t_{DW}	Data Setup to End of Write Strobe or Chip Select	20		ns
t_{WD}	Data Hold from Write Strobe	12		ns
t_{WI}	Clear INT from Write Strobe		65	ns

AC Electrical Characteristics (Continued)

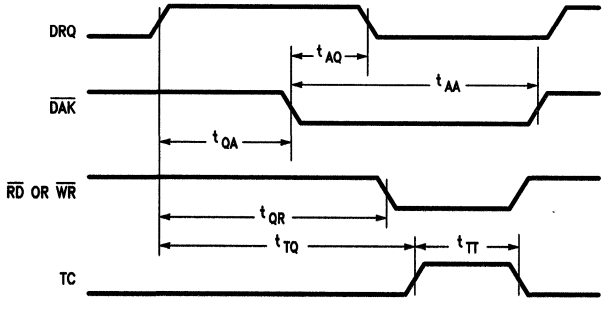
OSC2/CLOCK AND RESET TIMING



TL/F/9384–13

Symbol	Parameter	Min	Max	Units
t_H	Clock High Time	16		ns
t_L	Clock Low Time	16		ns
t_{RW}	Reset Pulse Width	100		ns

DMA TIMING (Note 9)



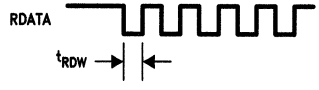
TL/F/9384–14

Symbol	Parameter	Min	Max	Units
t_{AQ}	End of DRQ from DAK		115	ns
t_{QA}	DAK Assertion from DRQ	10		ns
t_{AA}	DAK Pulse Width	75		ns
t_{QR}	DRQ to Read or Write Strobe	10		ns
t_{TT}	TC Strobe Width	50		ns
t_{TQ}	Time after Last DRQ That TC Must Be Asserted By		(Note 10)	ns

Note 9: DMA Acknowledge is sufficient to acknowledge a data transfer. Read or Write Strobes are neccessary only if data is to be presented to the data bus. If Read/Write Strobes are applied, then they and the Acnowledge must be removed within 1 μs of each other.

Note 10: TC is is the terminal count pin which terminates the data transfer operation. There are several constraints placed on the timing of TC. 1) TC is enabled by $\overline{DAK}$, so TC must be pulsed while $\overline{DAK}$ is low. 2) TC must occur before ((1/data rate x 8) − 1 μs). Data rate is the exact data transfer rate being used.

DRIVE READ TIMING

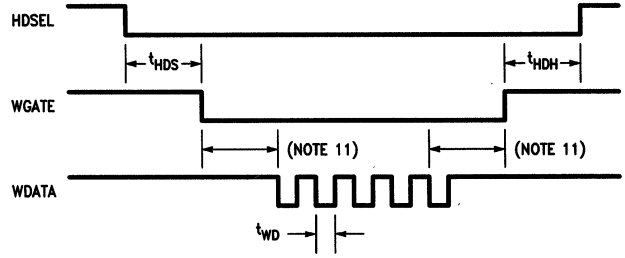


TL/F/9384–15

Symbol	Parameter	Min	Max	Units
t_{RDW}	Read Data Pulse Width	25		ns

AC Electrical Characteristics (Continued)

DRIVE WRITE TIMING

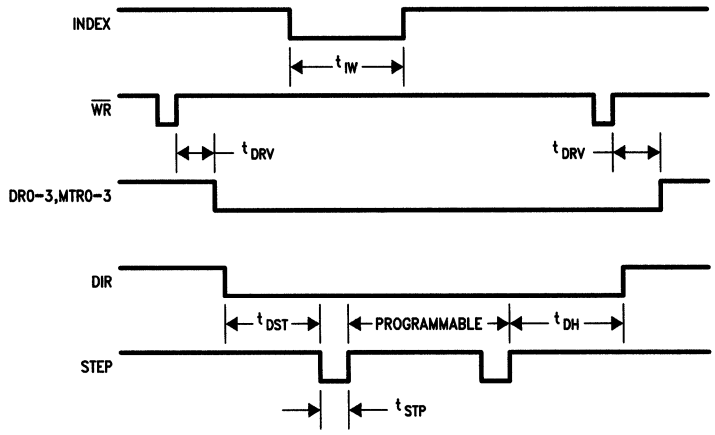


TL/F/9384–16

Symbol	Parameter	Conditions	Min	Max	Units
t_{WD}	Write Data Pulse Width	250 kb/s (MFM)	500		ns
		300 kb/s (MFM)	416		ns
		500 kb/s (MFM)	250		ns
		1000 kb/s(MFM)	225		ns
t_{HDS}	Head Select Setup to Write Gate Assertion		50		μs
t_{HDH}	Head Select Hold from Write Gate		15		μs

Note 11: Whenever WGATE is asserted the WDATA line is active. At the end of each write one dummy byte is written before WGATE is deasserted.

DRIVE TRACK ACCESS TIMING



TL/F/9384–17

Symbol	Parameter	Min	Max	Units
t_{DST}	Direction Setup prior to Step	6		μs
t_{DH}	Direction Hold from End of Step	1 step time		
t_{STP}	Step Pulse Width	8		μs
t_{IW}	Index Pulse Width	100		ns
t_{DRV}	Drive Select or Motor Time from Write Strobe		100	ns

AC Test Conditions (Notes 11, 12, 13)

Input Pulse Levels	GND to 3V
Input Rise and Fall Times	6 ns
Input and Output Reference Levels	1.3V
TRI-STATE Reference Levels	Active High − 0.5V
	Active Low + 0.5V

Note 11: C_L = 100 pF, includes jig and scope capacitance.

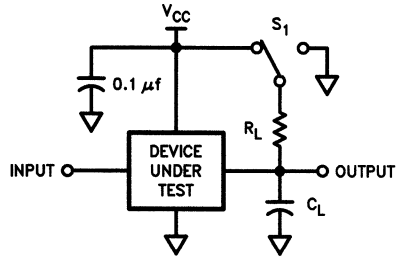
Note 12: S1 = open for push-pull outputs. S1 = V_{CC} for high impedance to active low and active low to high impedance measurements. S1 = GND for high impedance to active high and active high to high imepedance measurements. R_L = 1.0 kΩ for μP interface pins.

Note 13: For the Open Drain Drive Interface Pins S1 = V_{CC} and R_L = 150Ω.

Capacitance T_A = 25°C, f = 1 MHz (Note 14)

Symbol	Parameter	Typ	Units
C_{IN}	Input Capacitance	5	pF
C_{OUT}	Output Capacitance	8	pF

Note 14: This parameter is not 100% tested.



TL/F/9384–20

Floppy Disk Data Separator Design Guide for the DP8473

National Semiconductor
Application Note 505
Bob Lutz, Paolo Melloni
and Larry Wakeman

Table of Contents

1.0 INTRODUCTION

Due to the increase in CMOS processing capabilities it is now possible to integrate both the analog and digital circuitry to achieve a high performance monolithic data separator. The choice of CMOS technology also enables the integration of an analog data separator function with good performance onto a floppy disk controller, resulting in National's DP8473 integrated floppy data separator/controller.

This paper discusses the functionality of the DP8473 data separator blocks, after a brief introduction to floppy disk data separator theory. It then delves into the detail of PLL design theory, providing design equations and considerations that enables the user to optimize the performance of the PLL for various applications.

2.0 THE FUNCTION OF A DATA SEPARATOR

2.1 Encoding Techniques

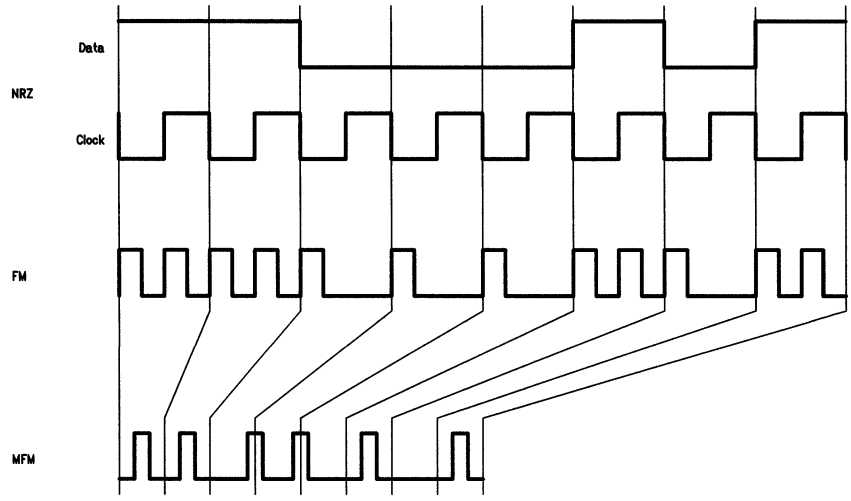
The floppy disk controller writes data to the floppy disk drive in a bit serial fashion as a series of encoded pulses. These pulses are then converted by the drive into magnetic flux reversals on the floppy disk media. The pulses can be later read by the drive and converted back to encoded pulses which can be decoded by the controller into the original data.

Since data is one serial set of bits, and because "real world" imperfections in the writing/reading process can cause the serial information to vary and jitter, the clocking information is embedded into the data stream, which enables synchronization to the data by the circuitry in charge of reading the data.

It is the purpose of the Data Separator circuit to take the encoded data from the disk, and to recover and separate out the clock signal. The separated clock and data signals are then sent to the controller's deserializer which converts the data to bytes of data suitable for microprocessor manipulation.

The two most popular encoding schemes used on floppy disks are: FM (Frequency Modulation), and MFM (Modified Frequency Modulation). FM defines a bit cell for each bit of data. Each cell contains a position for a clock pulse and a position for a data pulse. Each of these positions are referred to as windows. The clock pulse is present in every cell and a data pulse is present only if the data bit for that cell is a one. When this data is read back from a disk, a read clock can be generated from the clock pulses of the signal. An example of FM encoded data is shown in *Figure 1.*

FM encoding was the first method used for recording data on a floppy disk. It is still used in some low cost systems where storage capacity is not a critical issue. This method works very well and requires relatively simple circuitry to separate the clock pulses from the data pulses when the data is read back. However, only 50% of the useful disk space is used for recording data. The other 50% is used to record clock pulses.



TL/F/9419–1

FIGURE 1. Examples of how clock and data information is encoded into FM and MFM formats. Notice the increased density of MFM over FM.

MFM encoding allows 100% of the useful disk space for storing data, and is currently the most widely used recording format used for floppy disks. MFM defines a bit cell for each bit of data, similar to FM. Again, each cell contains a position for a clock pulse (clock window) and a position for a data pulse (data window). A data pulse is present if the data bit is a one. A clock pulse is present only if the data bit is a zero and the data bit in the previous bit window was a zero.

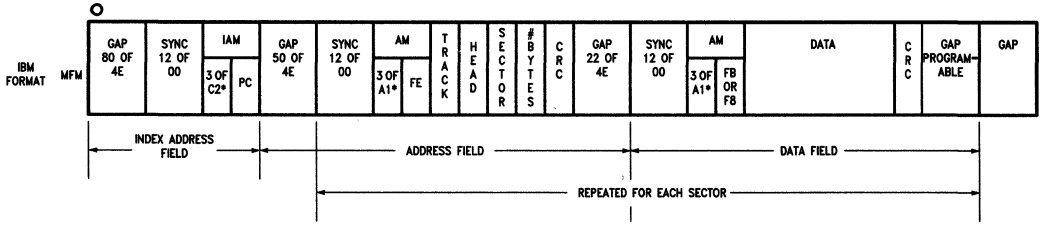
A comparison of FM and MFM can be seen in *Figure 1*. Because MFM requires fewer pulses to encode the same amount of data, the information can be stored in half the area required for FM encoded data. The only drawback of MFM is that it requires better read/write head and accompanying electronics. It also requires a higher precision data separator than FM requires. This is to resolve the location of each pulse more precisely than with FM.

2.2 Typical Floppy Format

A disk consists of many separate tracks. These tracks are configured as a set of concentric circles. Each track contains a set of many sectors. Each sector contains one Address Field and one Data Field. *Figure 2* shows the most common track format used on floppy disks today. It is the IBM double density standard.

The Address Field within a sector is used to identify what sector the following data field belongs to. It begins with a synchronization field or preamble to allow the data separator (which will be described soon) to synchronize to the speed at which the data is being read from the disk.

Next an address mark uniquely identifies this field as an Address Field. The address marks are encoded with a unique illegal pattern that is an MFM encoding rule violation. The violation is a missing clock pulse from a particular location within the byte. This illegal pattern guarantees that this is an address mark field and not a data pattern from some other area of the disk. The following four bytes identify the sector being read. This is followed by a CRC (Cyclic Redundancy Check). The CRC allows the controller to verify that the information read is free of errors. This is followed by a gap which is simply a series of bytes that physically separates the Address Field from the Data Field.



TL/F/9419–2

FIGURE 2. Typical format of a floppy disk. This is the IBM MFM standard.

Notes:

C2* = Data Pattern of C2, Clock Pattern of 14
A1* = Data Pattern of A1, Clock Pattern of 0A

The Data Field contains the data that the sector represents. It begins with a preamble and data field address mark, similar to the beginning of the Address Field. The actual sector data follows this. The data is followed by a CRC and then a gap, that separates this sector from the next.

2.3 Obstacles in Reading Data

Since MFM is the most popular floppy disk data encoding method the following discussion will refer specifically to MFM.

The floppy controller must be able to decode the data read from a disk drive. Theoretically, this could be a fairly easy process. The controller must first synchronize to the clock pulses in the preamble field of a sector. After that, it is just a matter of checking when the next encoded data pulse arrives. The pulse can arrive, 1, 1.5 or 2 bit periods later. Using this information the controller can decode this and all subsequent bits, reconstructing the original data from this information.

Unfortunately, this simple method is not so simple. The data pulses read back from a disk drive will generally be somewhat different from the data originally written.

There are three major sources of data degradation.

1. **Bit Shift**—As data is written, magnetic interaction of adjacent bits cause the data to be shifted in time from its nominal position. When these flux transitions are recorded close to each other, the superposition of their magnetic fields tends to move their apparent position. Thus when they are read, the floppy drive's peak detector moves the peak of these flux transitions apart from each other. This is the major cause of instantaneous bit shift.

   (This type of data degradation is mostly predictable and can be partially compensated for by shifting the data as it is written to the disk in the opposite direction that the bit is predicted to shift. This is called Write Precompensation. The DP8473 contains circuitry required to perform this function. The write precompensation circuit intercepts the serial data being written to the disk and shifts the data early, late, or none, based on the data pattern.)

   Several other factors can contribute to jitter. The drive's peak detector may be unbalanced, resulting in a bit shift similar to that described above. This could cause positive going peaks to appear earlier than negative going peaks or vice-versa. Also, a long cable between the disk drive and the disk controller may contribute to bit shift.

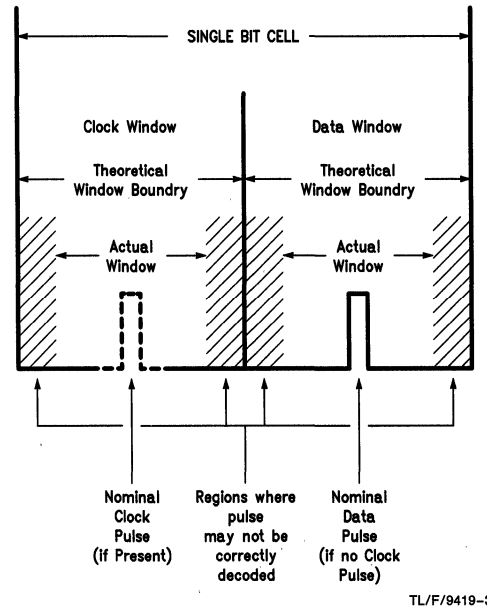
2. **Motor Speed Variation (MSV)**—This is an error in the spindle motor speed from the nominal, and causes the data rate to vary typically 1–2% for each drive. For design purposes this value is doubled since drive media is interchangeable. Thus a slow drive can record data that is read on a fast drive.

3. **Instantaneous Speed Variation (ISV)**—This is an additional speed error that is a constantly changing affect due to disk-jacket friction, and mechanical resonances. Usually this variation has a frequency component less than 1 kHz and causes the data rate to vary an additional 1–2% (again doubled).

While bit shift is the primary cause of decoding problems, the speed variations create difficulty in locking to the frequency of the data stream, and degrade jitter tolerance. The data separator must be able to synthesize the average frequency of the data coming in, and if the disk data rate differs from the nominal value then synchronization is more difficult.

2.4 Performance Measures of a Data Separator

There are several measures of data separator performance. The most universal one is window margin. Window margin measurements themselves can be subdivided into two categories, Dynamic, and Static (described shortly). Window margin is defined as the amount of bit shift that can be tolerated by a data separator without mis-decoding the data.



TL/F/9419–3

FIGURE 3. Window Margin Timing Definition

Figure 3 shows a timing diagram of a typical bit cell, and its composite data and clock windows. Theoretically a pulse (either clock or data) could be shifted in time either early or late relative to its nominal position by up to ¼ of a bit period and still be decoded correctly. This is the theoretical window boundary region in *Figure 3*. If the pulse is shifted more than this, then it would fall into another pulse's window. In reality, due to the limitations of practical data separator implementations, the actual window boundary in which data will be directly decoded is less than the full ¼ period. This is shown in *Figure 3* as the actual window region. Typically this actual window size is measured as either a percentage of the theoretical maximum or in terms of nanoseconds. The former is the more popular method and will be used here as well. In equation form:

$$WM\% = \frac{\text{(Actual Window Size)}}{\text{(Nominal Theoretical Window Size)}} \times 100\%$$

Window margin should be measured with a specified amount of ISV and MSV, at a specific data rate, with a specified data field and format pattern, and a known bit shift algorithm. If any of these are unspecified, then a true comparison of data separator performance is more difficult. Window margin is usually specified as a percentage of the nominal frequency window (as opposed to the nominal frequency plus the MSV).

There are two basic types of window margin tests. One test is where the data separator and controller must read a sector of data, in which all the bits are shifted until the data separator cannot read the sector correctly. This is called dynamic window margin. A second test is to present a long sequence of perfectly centered MFM clock pulses except for one bit. This bit is shifted until an error occurs. This second test is called static window margin.

Do Not Confuse the Two Measurements. The first one more correctly reflects the "Real World". The second one is an indicator of the accuracy of the circuits that compose the PLL, but does not include most of the errors due to the response of the PLL.

As an example of a dynamic window margin test: A data separator has a 70% window margin at 500 Kb/s, with a total ±1.5% MSV, and ±1% ISV. The encoding is MFM, and the data pattern is a repeating DB6DB6 . . . (HEX) pattern. A reverse write precompensation algorithm is used for pattern dependent bit jitter (all bits are jittered). These conditions are one of the worst case conditions for analog data separators. This means that the data separator will correctly decode a pulse so long as it is shifted no more than ±350 ns from its nominal position (the theoretical window at 500 Kb/s is ±500 ns) over the full MSV and ISV range.

Another data separator performance measurement is Bit Error Rate (BER). This is a measure that is defined as ratio of the number of bit errors during long term reading divided by the total number of bits read. A small bit error rate is desirable. BER is better for evaluation of total system performance, since the performance of the whole system effects on this figure. Thus as a final system checkout the manufacturer can specify the media, drives, data separator, and determine the error rate of this system. It is relatively difficult to isolate the bit errors due solely to the data separator. This specification is a less exact performance measurement than window margin for a data separator.

Why is Window Margin Important?

The greater the window margin the lower the error rate. For example if a bit is read with too much bit jitter for the data separator, then that data cannot be read and the whole sector (or file) is lost. This is especially fatal since floppies do not have error correction capability.

Another area where window margin is important, is manufacturing yields. A larger window margin ensures that when intermixing best/worst case drives and controllers there is minimum fallout. Thus larger volume vendors tend to try to optimize window margin to improve yields as well as data integrity.

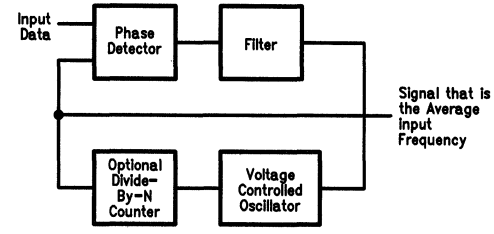
Each designer needs to decide for himself what margin requirements are necessary. In general, many high quality analog designs typically achieve 60–65% window margin under worst case conditions, with the best designs approaching 70%.

Later in sections 4.2 and 5.1 theoretical calculation and practical measurements of window margin are described.

2.5 Analog Data Separator Basics

The job of a data separator is to produce a read clock that follows the slow data rate change caused by the drive motor variation (MSV and ISV), but not track the instantaneous bit jitter. This read clock then is used to clock in the serial data into some type of deserializer. Generating a read clock for MFM encoded data is potentially difficult. Because of the MFM encoding rules, many clock pulses are missing from clock or data windows. As a matter of fact, in a long string of one's, there are no clock pulses at all. A data separator must use both clock pulses and data pulses to synchronize to the encoded signal. The most popular method to do this is with a Phase Locked Loop (PLL).

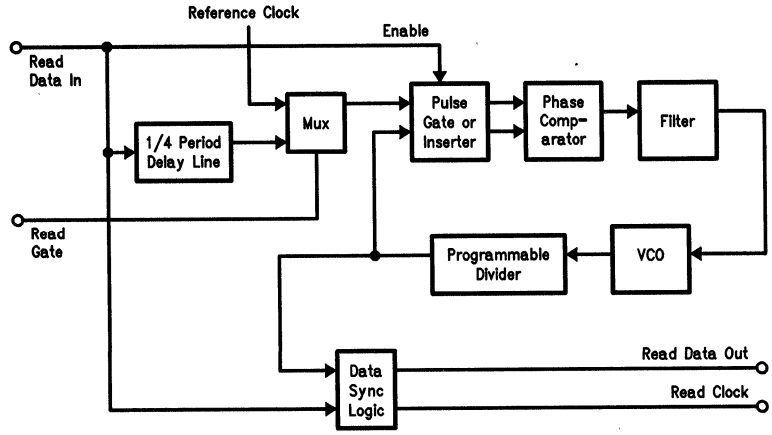
A PLL consists of three main components, a phase detector, a filter, and a voltage controlled oscillator (VCO), as shown in *Figure 4.* Also in most cases a divider is used to divide the VCO frequency as needed. The basic operation of a PLL is fairly straight-forward. The phase detector detects the difference between the phase of the VCO (or divider) output and the phase of a periodic input signal. This phase difference is converted to a current which either charges or discharges a filter. The resulting filter voltage changes the frequency of the VCO (and also the divider) output in an attempt to reduce the phase difference between the two phase detector input signals. A PLL is "locked" when the frequency of the two phase detector input signals is the same.



TL/F/9419–4

FIGURE 4. Simplified Block Diagram of Phase Locked Loop

With a slight modified version of the basic PLL, an MFM data separator can be made. A modification is required because MFM encoded data is not a periodic signal. A phase comparison can only be made when a pulse arrives from the disk. When there is no clock or data pulse, the PLL should continue generating the frequency it was generating before the missing pulse. This is called a phase only comparison, and it is the usual method of tracking an MFM signal.



TL/F/9419–5

FIGURE 5. Simplified Block Diagram of Typical Data Separator

A typical data separator is shown in *Figure 5*. In addition to the components of a typical PLL, it includes a quarter period delay line and either a pulse gate or pulse inserter (not both). Both of these blocks enable the pulse gate or pulse inserter to decide when the phase comparison should be made. The quarter period delay line delays the incoming data pulses a quarter of a bit cell, and this feeds the pulse gate or pulse inserter. The pulse gate will disable phase comparisons when a VCO pulse occurs but read data pulses are missing. The pulse inserter will insert fake read data pulses into the phase detector when there is a VCO pulse but no read data pulse. These components are required to determine the proper timing of the phase comparisons for MFM encoded data. The need for these blocks can be demonstrated by referring to *Figure 6*. Only the use of the pulse gate is described since this is what is implemented in the DP8473 data separator.

Figure 6a shows two MFM bit cells, each with a clock pulse. The VCO output provides two clocks per cell since an MFM pulse can appear in either of the two windows that compose the bit cell. (Note for simplicity the divider block is ignored.) To achieve lock, the data separator tries to line up the rising edge of the input pulses with the rising edges of the VCO output cycles. MFM encoded data is not periodic, that is some of the cells are missing pulses. The data separator must decide when to make a valid phase comparison. This can be seen from *Figure 6a* where the phase detector first makes a comparison to an early pulse, which is correct, but then on the next VCO cycle the phase detector now compares this VCO edge even though no input pulses are present. Hence, there must be a mechanism for fooling the phase detector into not making a comparison. The method chosen in the DP8473 is to use a pulse gate to eliminate the unwanted VCO edge.

However, disabling the phase detector's input does not completely solve the problem, as shown in *Figure 6b*. Here the first early pulse is compared correctly. At the beginning the next VCO cycle the data separator does not know whether to do a phase comparison, since it does not know whether the pulse is missing or just late. Thus by the time the next pulse does arrive the PLL is lost.

Therefore, the DP8473 data separator uses a 1/4 period (1/2 bit window) long delay line with the pulse gate. Now with this delay line, all phase comparisons are made to the delayed data. Thus the PLL is operating 1/4 of a period behind the data coming from the disk, but this allows the phase comparison enable logic to determine whether a pulse will occur in a bit cell or not, and make the proper comparison.

Figure 6c shows how this works. For an early bit, the data input enables a phase comparison, and the phase detector compares the delayed data bit to the VCO edge. In the case of this early bit, the proper pump up is generated. On the next VCO cycle, the quarter period delay has detected no pulse, and so no comparison is made. For the late bit, the comparison is enabled prior to the VCO clock, so a pump down is generated until the delayed data bit is seen by the phase detector.

At nominal frequency, a delay of 1/4 of a bit ensures that the phase detector will be properly enabled even if the data bit is late all the way to the edge of its clock or data window. (Remember one bit cell contains a clock and a data window. A data pulse will appear within either (but not both) of these windows. Therefore the theoretical maximum amount of bit shift is 1/4 of a bit cell.)

The quarter period delay line solves this problem of forecasting the future. It causes the MFM encoded data pulses to be delayed by a quarter of a bit period. This allows the pulse gate to determine when data pulses exist ahead of time and thus enable the phase detector only at the appropriate times.

It is important that the quarter period delay line be accurate. If the quarter period delay line is not accurate (ie. it's too long or too short), then the window margin performance of the data separator will be reduced. This performance reduction is due to the PLL's inability to correctly resolve bit shift near the edge of a bit window. For example, if at 500 Kb/s the delay line were shorter than it should be, say 400 ns long instead of 500 ns, then a bit shifted 450 ns from its nominal position is incorrectly decoded. The window margin in this case is immediately reduced 20% from it's ideal. The same degradation occurs when the delay line is too long.

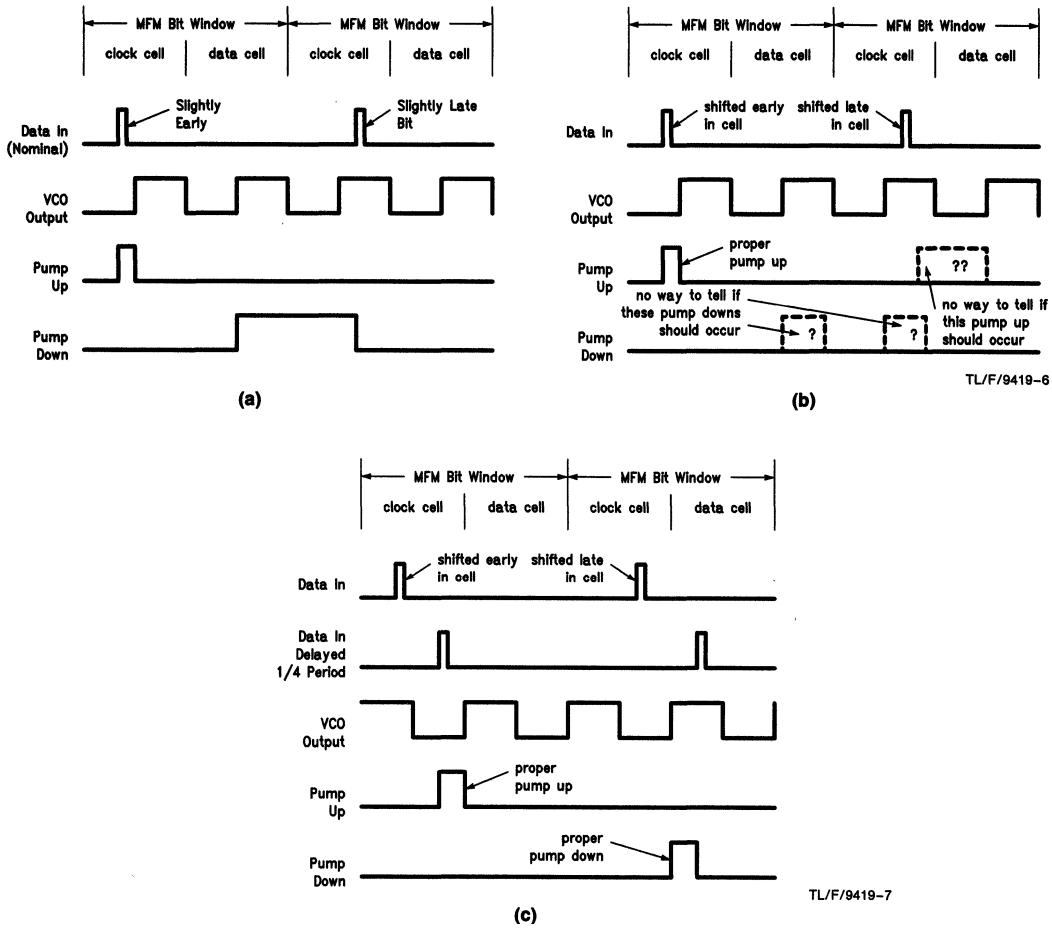


FIGURE 6. This shows why a ¼ period delay line is needed. a) Shows phase comparisons that occur if only a phase detector is used; b) Shows the data separator's need to predict the arrival of a pulse; and c) Shows how the ¼ period delay fixes this.

2.6 Operation of an Analog Data Separator

A data separator can be described as operating in one of three phases during each read cycle: Idle Phase, Initial Locking Phase, and the Tracking Phase.

Initially, when the data separator is not being used to read data from the disk, it is in the Idle Phase. While in the Idle Phase, the PLL is both phase and frequency locked to a reference frequency. (Frequency comparison is implemented by forcing a phase comparison every VCO clock.) The PLL must eventually lock to both clock and data pulses of the encoded data when it is read from the disk, so the reference frequency is generally two times the data rate frequency.

When data is to be read from the disk, the PLL switches from the reference frequency to the incoming data stream. Because the encoded data read from the disk is not a periodic signal, only phase comparisons are made. Since the PLL was initially locked to a frequency very close to twice the actual data rate, the time required for the PLL to lock onto the data read from the disk is minimized.

To further minimize this locking time, the beginning of each Address Field and Data Field starts with a preamble (or synchronization field). The preamble is a series of bytes with a zero data pattern (all clock pulses and no data pulses). When read, the preamble will produce a periodic signal with little bit jitter. The data separator can lock to this signal with the least chance of an error. It would be ideal for the floppy controller to switch the data separator from the Idle Phase to the Initial Locking Phase at the beginning of a preamble to enable the maximum amount of lock time.

Once the PLL is locked to the average frequency of the data being read from the disk, it should simply track the data frequency. This means tracking the slow data rate speed variations caused by the drive motor, yet ignoring instantaneous bit jitter. This is the Tracking Phase. The data separator then allows the controller's deserializer to start decoding the incoming data.

2.7 Digital Data Separators

A second method of separating clock and data information is to use a digital data separator. While the circuits for the analog solution has evolved significantly, digital data separators have also improved somewhat, in a (less than successful) attempt to match the performance of the analog approach. These circuits are described below.

First Generation Digital Data Separator (DDS)—This circuit is a very convenient all digital data separator. Its primary advantages are simplicity, and low external parts count. This circuit usually consists of a set of counter timing circuits and some control logic to count times between individual pulses, and thus determine whether a pulse is clock or data. The SMC9216 is representative of this technology. Its major disadvantage is performance, and the inability to optimize the window margin for various lock ranges. The dynamic window margin for these types of circuits is usually around 55% with no MSV and as low as 30% with a ±3% total MSV.

Second Generation DDS—A sophisticated digital data separator can be compared functionally to an analog data separator. The ideal digital separator consists of a sampler (phase detector), a ROM look up table, with memory (filter), and a programmable counter (VCO). The pulse gate can be implemented as an extension of the ROM look up table. These circuits are typically much better than 1st generation circuits, but still far short of the analog approach. These circuits have dynamic window margins of 50–55% over a ±6% lock range (total MSV variation).

3.0 DP8473 DATA SEPARATOR FUNCTIONAL DESCRIPTION

The integrated floppy disk data separator from National Semiconductor combine the performance of an analog PLL and the ease of use of a digital data separator. It does not require any external trimmed components, and it has a data rate range from 125 Kbits/sec up through 1.0 Mbits/sec. It is built using CMOS technology to achieve good linear performance as well as low power operation. A block diagram for the data separator is shown in *Figure 7*.

3.1 Block Diagram Description

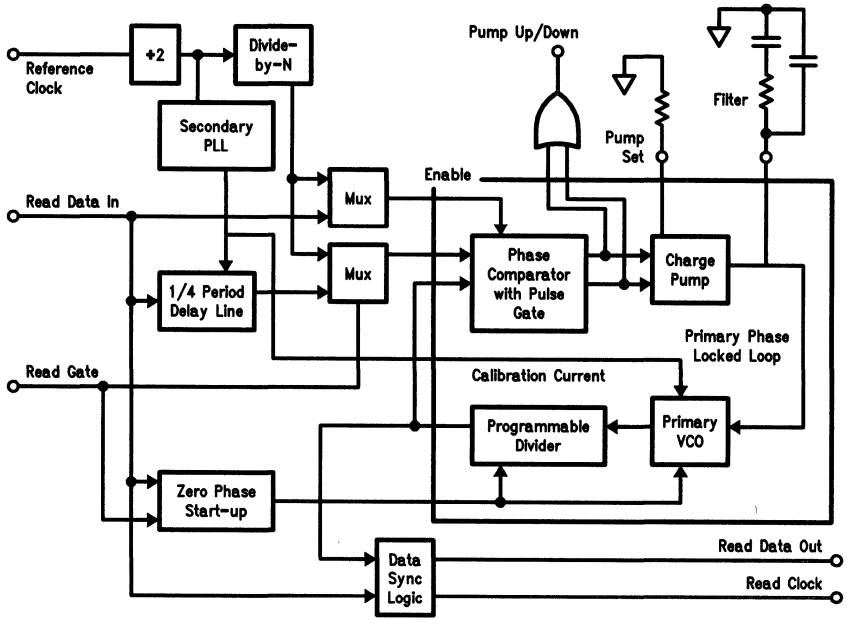
The heart of the DP8473 data separator is the main PLL. The main PLL consists of the VCO, programmable divider, phase detector, and the charge pump. The entire operation of the PLL and data separator logic is based on the Reference Clock which should be an accurate reference frequency. The Reference Clock is divided by two, then feeds the Secondary PLL, and the divide-by-N counter. As discussed later the Secondary PLL is used to calibrate the operation of the quarter period delay and Primary VCO. The Reference Clock's Divide-By-N counter and the Programmable Divider are both programmable counters whose divide by factor is determined by the data rate selected. The output of the divide-by-N and the Programmable divider is always twice the data rate. The output of the divide-by-N is used as a reference frequency for the PLL to lock to when the PLL is idle. The output of the Programmable Divider is the separated clock that is used to strobe the incoming pulses into the controller's deserializer.

Note: Throughout this discussion, the Reference Clock as shown in *Figure 7* is the master clock for the data separator block. This Reference Clock also generates several other clock frequencies that are used by the data separator sub-sections. In the following discussions the term Reference Clock refers only to the signal in *Figure 7* that feeds the divide-by-2 and divide-by-N blocks. Also the term Divide-By-N counter is used for the counter driven by the Reference Clock, whereas the Programmable Divider refers to the counter driven by the VCO.

In the DP8473, the Reference Clock of *Figure 7* is derived from a prescaler circuit that is operating at 24 MHz. The output of this prescaler circuit is 8 MHz for all data rates except 300 Kb/s. At 300 Kb/s the equivalent prescaler output is 9.6 MHz. The 24 MHz is intended to be a fixed frequency, but could be scaled lower for unique applications if desired.

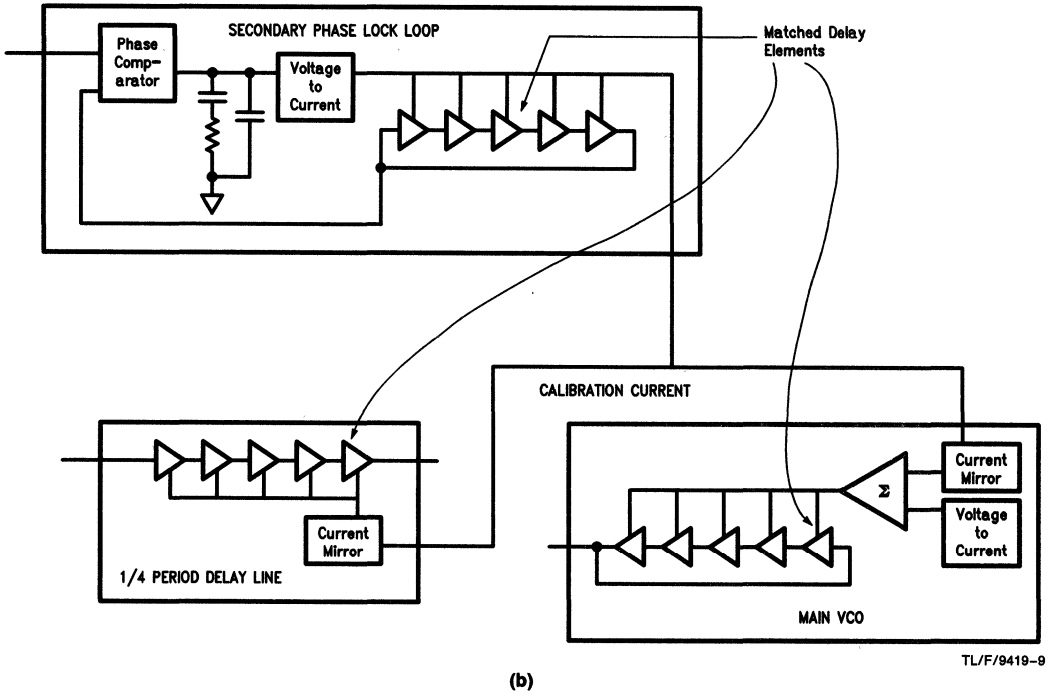
Under normal operation the Secondary PLL and the Primary VCO run at one half the Reference Clock frequency. Thus the Primary VCO's output is 4 MHz (except at 300 Kb/s where the VCO output is 4.8 MHz.) Operation at different data rates is accomplished by changing the Divide-By-N and Programmable Divider.

The basic operation of the Phase Locked Loop is fairly standard although there are several added features in the DP8473 PLL. The phase detector determines the phase



TL/F/9419–8

(a)



TL/F/9419–9

(b)

FIGURE 7. A more detailed Block Diagram of the DP8473 data separator, a) showing the zero phase start up, secondary PLL, control logic block and external R's and C's. b) Also showing a more detailed diagram of the secondary PLL, main VCO and delay line. This highlights the self calibration technique.

difference between two inputs. One input is always the divided output of the Programmable Divider.

The other input is either a reference frequency (derived from the Reference Clock) of twice the data rate while the data separator is in the idle mode, or when reading the disk the encoded data from the drive after it has passed through the quarter-period delay line.

While in the idle mode, the phase detector determines both phase and frequency information. This is accomplished by forcing the pulse gate to enable all phase comparisions. This state is set by the top two multiplexers of *Figure 7* which in idle mode select clocks generated by the Reference Clock to enable the pulse gate on every clock edge (hence all clocks are compared by the phase detector). By locking to the Reference Clock generated frequencies in both phase and frequency, the PLL is preventing from locking back to a harmonic of this frequency.

When the data separator is told to read the incoming data pulses, Read Gate is asserted. This changes the signals selected by the multiplexers. The top multiplexer switches to inputting the "raw" read data pulses into the pulse gate, and the bottom multiplexer sends read data delayed by a quarter bit period to the phase detector input. When this switch occurs, the Zero Phase Start-Up logic synchronizes the Programmable Divider's output to be in phase with the very next arriving data pulse. This causes the PLL to acquire lock to the data quicker since it is starting with a "near zero" phase error between the Programmable Divider and the encoded data.

The quarter-period delay line consists of a series of voltage controlled delay elements. The encoded data from the disk drive enters the beginning of the delay line. The output is derived from the output of one of the delay elements. The delay element used for the output depends upon the data rate used.

When locked to either the read data pulses or the reference, the phase detector issues either a pump up signal or a pump down signal depending upon whether the VCO should increase or decrease its frequency. The length of this pump signal is proportional to the amount of phase difference between the two input signals. When locked, the phase difference between the VCO and the delayed data will be small. In this case the width of the pump signal could be so small that the rise time may prevent the signal from ever being recognized by the charge pump. To ensure that the charge pump can recognize even the smallest pump signal, both pump up and pump down signals are asserted at each phase comparison and the appropriate signal is extended by an amount proportional to the phase difference between the two input signals. The pump signals are then subtracted from each other by the charge pump. Therefore, the rise time of the pump up or down signals will not degrade the performance of the charge pump.

The charge pump simply adds or removes an amount of charge proportional to the length of the pump signal to or from an external filter. The voltage of the external filter determines the frequency produced by the VCO.

Finally a synchronized clock and serial data signal is sent to the controller's deserializer by the Data Synch Logic Block. This circuit takes the output of the Programmable Divider and the Read Data pulses, and synchronizes these two signals, by centering the read data pulse in the appropriate Programmable Divider's clock cycle. This allows the controller to easily deserialize and decode the data pulses properly.

An additional block not shown here, but used on the DP8473 is the filter selection logic. This logic is used to select different filters for different data rates. The description and use of this circuitry is described in section 5.3.

3.2 Self Calibration

Normally, most VCO implementations would need an external precision capacitor (maybe trimmed) to set its center frequency. Also, the quarter period delay line would require an external trimming resistor to set the delay to exactly a quarter of the data rate. The actual delay of the delay elements used in these functions would normally vary from one part to another due to normal process variations. However, the DP8473 has been designed to eliminate the need for these external trims. There are actually two PLLs in the DP8473; the Primary PLL, and a Secondary PLL. The Primary PLL is used for data separation. The Secondary PLL is used to calibrate all of the delay elements used in the chip. This includes the quarter-period delay line and the main VCO.

The VCO of the Secondary PLL is a ring oscillator of delay elements. The amount of delay that each inverter produces is regulated by a control voltage which is the internally connected output of the Secondary PLL. The Secondary PLL is locked to the same frequency as the Primary VCO, half of the reference clock frequency. The delay elements used in the secondary VCO are identical to those used in the Primary VCO and are regulated by the same control voltage. There are also the same number of delay elements in each. Therefore, the center frequency of the Primary VCO is internally trimmed to exactly half of the reference frequency. Since the delay elements in the Secondary PLL have a known delay, any number of identical elements that are set by the calibration voltage will have a very accurate delay. Thus the quarter period delay line is just a chain of these delay elements that have the desired total length.

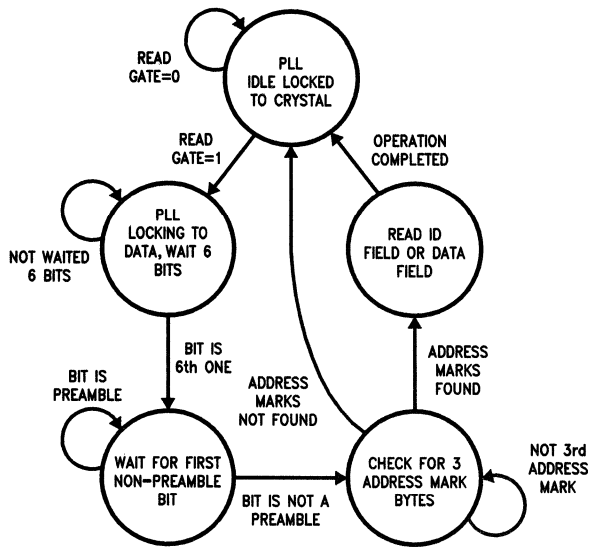
The delay elements used in the quarter-period delay line are also of the same type used in the secondary VCO. Because the delay of each element is accurately set by the Secondary PLL, there is no need for any trimmed tuning components for any of these circuits. Essentially, the only external passive components required for the DP8473 are for the filter(s) (two capacitors and a resistor per data rate), and a resistor to set the gain of the charge pump (ie. the amount of charge pump current).

3.3 Data Separator Read Algorithm

Since the DP8473 floppy disk controller incorporates the analog data separator, it takes advantage of close proximity of the controller and data separator blocks to implement a read algorithm that is much more sophisticated than previous floppy controller integrated circuits.

Before describing the details of the algorithm, a brief discussion of a disk read (or write) is necessary. When the controller is issued a read (or write) command, it is asked for a specified sector. The controller starts to look at the incoming MFM information. It scans this information, trying to locate the proper address field. In order to do this, the data separator is first told to lock to the disk data, and once locked the controller looks at the incoming information. In most controllers, the data separator is told to look at the data continuously until the correct sector address is found. This makes the data separator susceptible to being thrown out of lock, since the controller is not "watching" the data separator to see if it has maintained lock through the search process (which it can easily lose). Once the correct address is found the controller must then ensure that the associated data area is read, by re-locking to the incoming signal and then reading (or writing) the data.

Figure 8 shows in state diagram form the algorithm used by the DP8473 controller to ensure that disk data is correctly read (or written). This algorithm is much more sophisticated than previous generation controllers. The following describes the operation. When the controller is idle, the data separator is locked to the crystal/clock reference in a frequency comparison mode. (Frequency comparisons are made by disabling the pulse gate, and ensuring that all reference and VCO cycles are compared.)



TL/F/9419–10

FIGURE 8. State Diagram for a Data Separator Synchronization and Read Operation

When a read command is issued to the controller the controller asserts an internal Read Gate signal to the data separator. This causes the PLL to switch from locking to the reference to locking to the data with the pulse gate enabled, and the primary VCO/divider started in phase with the next incoming pulse. The PLL waits 6-bit times to lock to the data. When the seventh bit arrives the data separator assumes that this bit is a preamble bit (thus an MFM clock bit). The controller/data separator then continues looking at the data until a non-preamble pattern is detected (ie. an MFM data pulse). It then checks to see if it has now encountered an address mark with the proper rule violation. If it has not, read gate is deasserted. The data separator returns to the idle state for 6 byte times, and then starts all over again.

If three address mark bytes are found then the data separator remains locked to the data while the controller looks to see if it has found the right address field. If the controller discovers that this field is not the correct address field then it deasserts read gate for 6 bytes, and tries again.

If the correct address field is encountered, the controller deasserts read gate during the gap between the address and data fields. It then re-asserts read gate, and follows the state diagram to read the data field (ie. looking for preamble, address marks etc.).

This comparison is done on a bit-by-bit basis, therefore ensuring that the PLL never tries to lock on an unwanted field for more than one bit time. In other words, the PLL will never loose lock. This algorithm provides a very fast lock to the data stream, and ensures that the data separator never falls out of lock while reading the data. Both of these features reduce the need to do retries of operations to ensure correct execution.

4.0 DESIGNING WITH THE DP8473 DATA SEPARATOR

The following section is a fairly in-depth description of the design characteristics of the PLL in the DP8473 controller. *(National Semiconductor cannot be responsible for the sanity of any one who ventures into this section. Hence we recommend using the filter values supplied in the datasheets.)*

Two elements determine the overall performance of a Phase Locked Loop: the loop gain and the loop filter design. When using the DP8473 both of these elements are controlled by the user. The amount of current in the charge pump circuit can be set with an external resistor. This will set the overall gain of the PLL. The filter is external to the DP8473 and is user definable. This gives the user the possibility of tailoring the data separator performance to his own application requirements and design criteria. The following information will present some tradeoffs that apply in choosing the external components for typical applications.

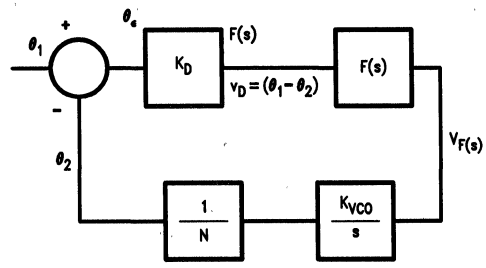
4.1 Basic Phase Lock Loop Theory

This section will first start with the basic control systems model for a second order PLL, and then apply these basic equations to the individual blocks that compose the data separator in the DP8473 Controller.

Initially Locked Model

In order to understand the behavior of the data separator and to discuss the tradeoffs of the different design parameters, some background in the theory of PLLs will be presented.

Figure 9 shows a diagram of a PLL which is assumed to be in a locked state. Each box contains the phase transfer function of the corresponding block and each node has the relative phase signal. The PLL is locked, which means that the VCO output and the input signal are at the same frequency and in-phase with each other (the phase error is a constant). This model is useful for understanding the phase locking process when the PLL is switched from the reference frequency to the incoming data.



TL/F/9419–11

FIGURE 9. The Block Diagram for an Initially Locked PLL Control System, Showing the Transfer Functions for Each Block

The frequency variations that this model take into account are assumed small enough so that the loop stays locked in frequency. Therefore, only the effect on the phase difference is considered. Also, it is easier to refer to bit shift tolerance in terms of phase. Hence the phase transfer functions yield the most appropriate information.

The concept of phase is very much related with the concept of time. The advantage of phase information is that it is independent of frequency of the signal and it is measured as a pure number (radians).

A simple RC filter model will be used to simplify the math. This is actually a good model because the effect of a second capacitor is only seen at high frequencies. This simplification allows the use of second order PLL theory that is easily available in literature. (See for example: R.E. Best, Phase Locked Loops, McGraw-Hill, 1984)

From *Figure 9* the closed loop phase transfer function for the loop can be derived using standard control system theory techniques, and reduces to:

$$H(s) = \frac{\theta_2(s)}{\theta_1(s)} = \frac{K_D K'_{VCO} F(s)}{s + K_D K'_{VCO} F(s)} \quad (1)$$

where $K'_{VCO} = K_{VCO}/N$, N being the number of VCO cycles between phase comparisons due to the divider that typically is inserted between the VCO and phase detector. The closed loop phase error function can be written as:

$$H_\epsilon(s) = \frac{\theta_\epsilon(s)}{\theta_1(s)} = 1 - H(s) =$$

$$\frac{\theta_1 - \theta_2}{\theta_1} = \frac{s}{s + K_D K'_{VCO} F(s)} \quad (2)$$

Now we'll evaluate the variables that appear in expressions (1) and (2).

The Charge Pump

In the phase detector/charge pump circuit a current is generated in the correct direction (positive or negative) every time the edges of the VCO/divider output and the incoming data pulses are not coincident. The current is a pulse with amplitude equal to I_{PUMP} and length equal to the phase error between the two signals. The pump current is zero for the rest of the period, so the average current is:

$$I_Z = \frac{I_{PUMP}\theta_\epsilon}{2\pi} \quad (3)$$

where θ_ϵ is the phase error between the VCO/divider and input pulses. The phase detector and charge pump gain is:

$$K_D = \frac{I_Z}{\theta_\epsilon} = \frac{I_{PUMP}}{2\pi} \quad (4)$$

where $I_{PUMP} = K_P \times I_R$. I_R is the current set by an external resistor at SETCUR pin. The current at this pin is: $I_R = V_{REF}/R_1$, $K_P = 2.5$, and $V_{REF} \cong 1.2V$. Thus combining equations:

$$I_{PUMP} = \frac{(2.5)(1.2)}{R_1} \quad (5)$$

Note that the maximum current that can flow to or from the charge pump is 500 μA, which corresponds to a resistor value of 3 kΩ. The minimum current is limited to 125 μA by stability and leakage constraints on the internal reference circuits. So R_1 must be smaller than 12 kΩ. In conclusion, the charge pump current and resistor can be set in the following range:

$$125\ \mu A \leq I_{PUMP} \leq 500\ \mu A$$
$$3\ k\Omega \leq R_1 \leq 12\ k\Omega$$

Any value in this range can be chosen, and usually the choice is dependent on the PLL filter capacitor's mechanical size and cost. We have chosen in the datasheet a value of 5.6 kΩ since it represents a good compromise of all these considerations.

The VCO and Programmable Divider

The VCO gain is defined as the ratio between a frequency change at the output vs. a voltage change at the input (FILTER pin). This value cannot be set by the user and has been designed to be immune to process, temperature and voltage variation. There is a variation of less than ±20% between different parts, and the typical value of K_{VCO} is:

$$K_{VCO} = 25\frac{Mrad/sec}{volt} \quad (6)$$

The actual value K'_{VCO} for expressions (1) and (2) differs by a factor of N. N is the ratio between the frequency of the internal VCO and the "instantaneous" frequency of the data. This takes into account both the factor due to the way the data is encoded, and the factor due to the internal programmable divider used for the data rate selection. The following table gives the value of N for different codes, data rates, and data patterns. K'_{VCO} can be derived from these values of N.

TABLE I. VCO Gain Reduction Factor for the DP8473 with a 24 MHz Crystal/Clock

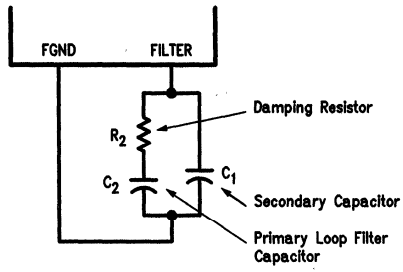
Data Rate	Code	Data Patterns	N
1 Mb/s	MFM	all 0's, all 1's 010101 . . .	4 8
500 Kb/s	MFM	all 0's, all 1's 010101 . . .	8 16
	FM	all 0's all 1's	8 4
300 Kb/s	MFM	all 0's, all 1's 010101 . . .	16 32
250 Kb/s	MFM	all 0's, all 1's 010101 . . .	16 32
	FM	all 0's all 1's	16 8
125 Kb/s	FM	all 0's all 1's	32 16

So for a 250 Kb/s MFM data rate N = 16 and $K'_{VCO} = 1.56$ Mrad/set/volt.

The loop filter calculation is made assuming lock and acquisition during a preamble (all 0's pattern), so these values of N are used in the bandwidth and damping calculations shown later.

The PLL Loop Filter

Inside the data separator, the charge pump output is connected directly to the VCO input. A filter is attached externally to this point. The typical configuration of this filter is shown in *Figure 10*. The output of the phase detector/charge pump circuit is basically a current generator with a very high output impedance (hundreds of kΩ). This high impedance combined with the external capacitor, C_2, of the filter provide a small (close to 0) steady phase error after a frequency step in the input signal. The charge pump setting along with C_2 sets the bandwidth on the PLL. The DP8473's charge pump circuit eliminates the need for an external active filter. The resistor R_2 is the damping resistor and it controls the stability of the loop.



TL/F/9419–12

FIGURE 10. Simple Schematic of Typical DP8473 Data Separator Filter Configuration

The filter design is usually improved by adding another capacitor in parallel, C_1. This second capacitor is intended to improve the low-pass filtering action of the PLL. In our subsequent filter discussions, C_1 is ignored initially since its value will be much smaller than C_2.

In the DP8473, the input of the filter is a current from the phase detector/charge pump, the output is a voltage to the VCO. Therefore, the transfer function of the filter of *Figure 9* is simply its impedance:

$$F(s) = Z(s) = \frac{1 + sR_2C_2}{sC_2} \tag{7}$$

(As mentioned, we are ignoring the effect of C_1 for now.) Substituting these equations into (1) and (2) produces:

$$H(s) = \frac{\dfrac{K'_{VCO}K_D(sR_2C_2 + 1)}{C_2}}{\left(s^2 + s(K'_{VCO}R_2K_D) + \dfrac{K'_{VCO}K_D}{C_2}\right)} \tag{8}$$

This then reduces to a standard second order equation of the form:

$$H(s) = \frac{(2s\zeta\omega_n + \omega_n^2)}{(s^2 + 2\zeta\omega_n s + \omega_n^2)}$$

and similarly the error function has the form:

$$H_\epsilon(s) = \frac{s^2}{(s^2 + 2\zeta\omega_n s + \omega_n^2)} \tag{9}$$

In this discussion we won't be making use of the error function equation, however it is the basis of much of the acquisition equations which are used to derive *Figures 11* and *12*. The complete analysis is beyond the scope of this paper, but is discussed in detail by most of the references. From equation 8, and the standard second order equation we can solve for the bandwidth, ω_n and damping, ζ. The natural frequency is:

$$\omega_n^2 = \frac{K'_{VCO}K_D}{C_2}$$

$$\omega_n = \sqrt{\frac{I_{PUMP}K'_{VCO}}{2\pi C_2}} \tag{10}$$

By combining K_{VCO} and I_{PUMP} constants, the design equations for the PLL can be simplified, by introducing K_{PLL}, which is defined as the product of $V_{REF} \times K_P \times K_{VCO}$. By using the K_{PLL} constant the above equation becomes:

$$\omega_n = \sqrt{\frac{K_{PLL}}{2\pi N R_1 C_2}} \tag{11}$$

where N is the VCO divider as defined in Table I. The damping factor is given by:

$$2\zeta\omega_n = K'_{VCO}R_2K_D\frac{K'_{VCO}R_2K_DC_2}{C_2}$$

$$2\zeta\omega_n = \omega_n^2R_2C_2$$

$$\zeta = \frac{\omega_nR_2C_2}{2} \tag{12}$$

These two parameters (equations 11 and 12) will allow us to calculate the PLL filter components based on bandwidth, and damping. The closed loop phase transfer function shows that the PLL behaves like a low-pass filter. It passes signals for input phase signals whose frequency spectrum is between 0 and ω_n. This means that a second-order PLL is able to track a phase and frequency modulation of the input signal up to a frequency equal to $\omega_n/2\pi$, and it will not follow input variations of higher frequencies.

4.2 System Performance and Filter Design

The system performance of a data separator can be described by three main criteria.

1) Acquisition Time—ability to guarantee lock during a preamble.
2) Window Margin—ability to recognize data shifted in time from its ideal position (bit jitter) without incorrectly decoding it.
3) Tracking of Disk Data—ability to follow slow (<1 kHz) disk data speed variations.

The filter design must meet the requirements set by these performance characteristics. The two conflicting requirements are that the bandwidth must be large enough to ensure proper locking to the data stream, but as small as possible while tracking the data to maximize bit shift rejection and window margin. Primarily the filter sets the bandwidth, and it is determined by the required acquisition time as shown later.

To illustrate this, a numerical example (for 500 Kb/s data rate) is presented following the design considerations step by step. After we have completed the paper design a discussion of performace measurements is provided. Once the initially calculated paper values are chosen, then real measurements must be made, and adjustments to these values are decided upon.

Acquisition to the Data Stream

Acquisition means to achieve phase lock and to bring the phase error of the VCO to zero, or close to it. This includes acquisition of phase lock to either the data input or to the reference frequency. The lock mechanisms for these two cases may be different.

At the moment just before Read Gate is asserted, it will be assumed that the VCO is locked to the reference frequency. It has been locked for a relatively long period of time, therefore the phase error between the VCO output and the reference frequency is nearly zero. Because the system is initially locked, the initially locked model can be used.

When Read Gate is asserted by the controller, the input to the phase detector is switched from the reference frequency to the data stream. This is an instantaneous change of both phase and frequency to the input of the phase detector. The loop must be designed to assure that it can achieve both phase and frequency lock to the incoming data. Phase and Frequency Lock implies that the steady state phase and frequency error at the phase detector input is near zero.

The goal is to lock to the data stream within the length of the preamble, very often half of the preamble to increase the probability of locking successfully. In fact, during a preamble the data pulses are relatively free of bit shift and the frequency is constant. There are two basic requirements to ensure that the data separator correctly locks to the data stream in the required amount of time.

1. The loop bandwidth must be large enough to ensure that phase and frequency error of the VCO goes to zero within the required time (usually within ½ the length of the preamble). This implies that the shorter the preamble the larger ω_n.
2. The filter must also be designed to guarantee that a data pulse will never fall out of the data window during the lock process. The peak phase error during acquisition must be less than ½ of a data or clock window (i.e. or $< \pi/2$). If the filter is incorrectlly designed and the data pulse falls outside the window (called cycle slipping) during acquisition, the loop may never lock within the desired acquisition time and the encoded data will not be decoded correctly. This requires that to guarantee lock over a wider variation of data rate, a larger ω_n is required.

Both of these requirements can be approximately derived from *Figures 11* and *12*. These curves plot relative phase error normalized to ω_n versus time in units normalized to ω_n. This period of time, and the amount of phase error present is dependent upon ω_n and damping.

For the first requirement, the phase error settles close to 0 in about:

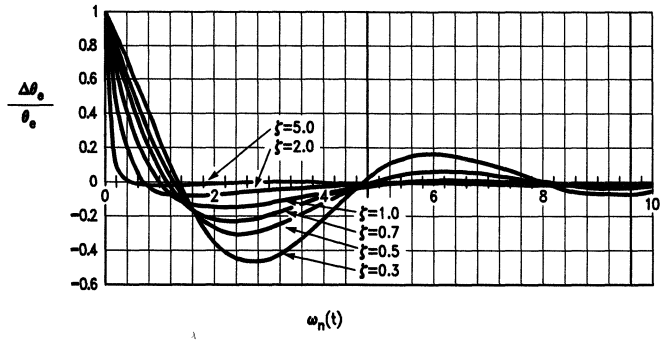
$$t_{acq} = \frac{5}{\omega_n} \tag{13}$$

This equation yields a minimum starting bandwidth, and is valid for systems where the speed variation of the incoming data is small (±1–2%).

For the second requirement, the peak phase error, $\theta_{e(PEEK)}$, during acquisition can be determined from *Figures 11* and *12*. The design must ensure that the sum of this peak phase error due to a phase step, $\theta_{e(PHASE)}$, the peak phase error due to a frequency step, $\theta_{e(FREQ)}$, and the phase error due to PLL noise and non-linearities, $\theta_{e(PLL)}$, must all be less than $\pi/2$. In equation form:

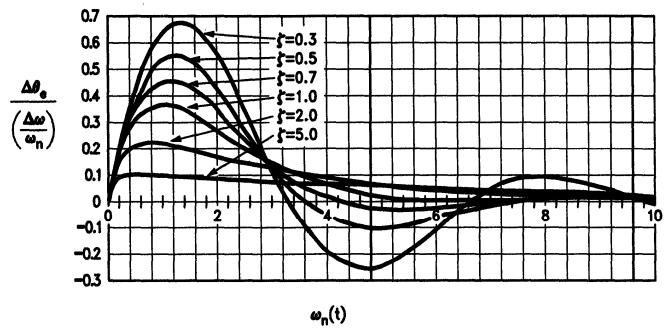
$$\theta_{e(PEAK)} = \theta_{e(FREQ)} + \theta_{e(PHASE)} + \theta_{e(PLL)} < \frac{\pi}{2} \tag{14}$$

Thus the sum of the peak phase errors for a phase step and a frequency step must be calculated.



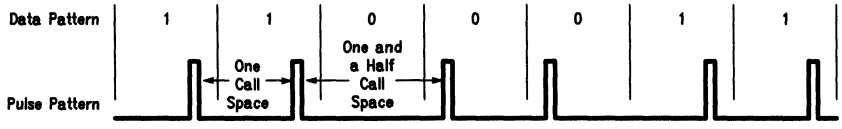
TL/F/9419–13

FIGURE 11. A Plot of Normalized Phase Error of a PLL to a Phase Step Input



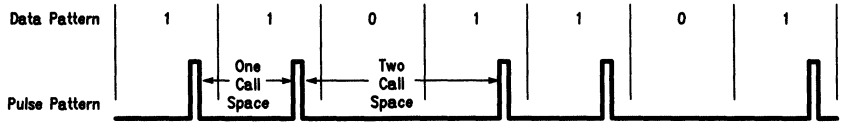
TL/F/9419–14

FIGURE 12. A Plot of Normalized Phase Error of a PLL to a Frequency Step



TL/F/9419–15

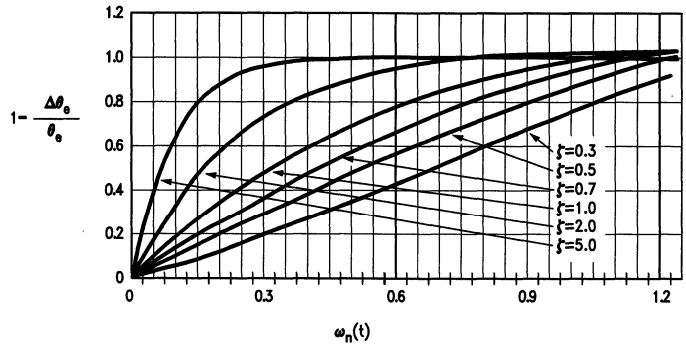
(a)



TL/F/9419–16

(b)

FIGURE 13. Two "Worst Case" Data Field Patterns for Measurement of PLL Bit Shift Tolerance a) "11000" Pattern with 2/3 μs Pulse Spacing (at 500 Kb/s), and b) A "DB6" ("110") Pattern with 2/4 μs Pulse Spacing (at 500 Kb/s)



TL/F/9419–17

FIGURE 14. A Plot of Normalized Bit Shift Resistance Versus Time for a PLL

For determining the peak phase step error, the value of $\theta_{e(PHASE)}$ is the maximum Y-axis value of the chosen curve multiplied by the input phase step, and the result is in radians. For example with a damping of 0.7 a maximum error of -0.20 occurs at about 3 ω_n. If the maximum phase step is $\pi/2$ when switching to the data, then the peak phase error is 0.1 π radians. The choice of which curve to use depends on damping factor desired.

Since a frequency step is also present at the transition of READ GATE, the peak phase error of the frequency step given by the normalized plot in *Figure 12* must also be derived. The phase error can be calculated by reading the peak Y-axis value from the desired curve and using:

$$\theta_{e(FREQ)} = Y\frac{\Delta\omega}{\omega_n} \qquad (15)$$

to determine $\theta_{e(FREQ)}$. Where Y is the value read from the Y axis, and $\Delta\omega$ is the maximum frequency step times 2π. The maximum frequency step is the worst case frequency variation of the data being read from the disk drive, which is the sum of the MSV and the ISV.

With the equations for loop bandwidth, damping factor, and the relationship between acquisition time and bandwidth, the following example demonstrates the first steps at arriving at the loop filter components.

Example 1: Design a data separator using the DP8473. Determine the loop bandwidth, dampening factor, and C_2, R_1, R_2 component values for a data separator that decodes MFM data at a data rate of 500 kbits/sec. The preamble is 12 bytes long, and the total MSV/ISV is ±6%.

Select a value of pump current resistor. For example 5.6 kΩ.

Find out the minimum acquisition time required. Generally, half preamble which is 6 bytes. Thus:

$$t_{acq} = ((6 \times 8)\text{ bits}) \times 2\ \mu\text{s/bit} = 96\ \mu\text{s}$$

Next calculate ω_n based on the larger of the two acquisition requirements. The first requirement for ω_n is: $5/\omega_n < t_{acq}$; Thus:

$$\omega_n > 52.5\ \text{Krad/sec.}$$

Calculate ω_n for the second acquisition requirement, i.e. ensuring the maximum phase error is less than $\pi/2$. Due to the Zero Phase start-up block within the Data Separator, the maximum phase step when switching to the data is $\pi/8$. We'll choose a damping of 0.7. From *Figure 11* the maximum overshoot is 0.2 ($\pi/8$) = 0.08 rad (Note 1.0 rad = 314 ns at 500 kb/s). Assume that the data separator contributes a total 0.1 rad noise error. This is for charge pump asymmetry, delay line variation, and VCO jitter.

Using equation 18:

$\theta_{e(PEAK)} = \theta_{e(FREQ)} + 0.08$ rad $+ 0.1$ read $< \pi/2$ or . . .

500 ns > 25 ns + 31 ns + $\theta_{e(FREQ)}$ which results in:

$\theta_{e(FREQ)} < 443$ ns or 1.41 rad

This yields the maximum tolerable phase error due to a frequency step (which is dependent on ω_n).

5

First find the maximum frequency step in radians that the data separator must undergo. The design requirement was 6%, but in order to account for gain variations in the data separator some margin on top of this is required, so we will design to 8% total speed variation. Thus:

$$\Delta\omega = 0.08\ (500k)\ 2\pi = 251\ \text{Krad}$$

The relative phase step error from *Figure 12* using a damping factor of 0.7, is Y = 0.45, plugging into equation 15:

$$\omega_n \geq \frac{\Delta\omega}{\theta_e} = \frac{0.45\ (251\ \text{Krad})}{1.41} = 80\ \text{Krad/s}.$$

The larger of the two calculated ω_n's is 80 Krad/sec, so that is the chosen bandwidth.

$$C_2 = \frac{K_{PLL}}{2\pi R_1 N \omega_n^2} \qquad (16)$$

$$= \frac{75\ \text{Mrad}}{2\pi(8)(5.6\ \text{k}\Omega)(80\ \text{k}^2)} \approx 0.041\ \mu\text{F}$$

We will round down to the next lowest standard value of 0.039 μF.

$$R_2 = \frac{2\zeta}{\omega_n C_2} \qquad (17)$$

$$R_2 = \frac{2\zeta}{C_2\omega_n} \approx \frac{2(0.7)}{(0.039)\ (80K)} \approx 450\Omega$$

In the above example we have calculated a set of component values for the acceptable bandwidth based on acquisition. This calculation yields a good starting point from which experimental measurements can be made, but may not be the optimum values. Depending on other considerations we may decide to chose a value of ω_n that is slightly different depending on window margin, or bit shift performance; as will be shown.

Theoretical Dynamic Window Margin Determination

Previously in section 2.4 Window Margin was discussed in terms of distortions of the data that degrade the window margin. Here, using a model for these distortions we will arrive at a calculation of the expected Dynamic Window Margin for the DP8473 analog PLL. The effects of Window error, VCO jitter, and PLL response to a previous or present bit shift all cause a reduction of the available window margin available. (Remember the goal is to maximize the total available bit window.) The following analysis provides a feel for the amount of degradation due to various parameters, and serves to provide an indicator of the expected PLL performance. The following is a list of parameters that cause loss of window:

Internal Window Error (or static phase error): The window error is related to the accuracy of the internal delay line in the data path before the phase detector. As explained previously, this delay line is automatically trimmed using the crystal frequency as reference. The static phase error is the sum of two factors. One of these factors is the difference between the data stream frequency and the nominal and unavoidable internal mismatches. Another contributing factor is charge pump leakage. This factor causes a perceived phase error that is equivalent to varying the delay line length. This is usually $\geq$ 2%–4%.

VCO Jitter: The VCO jitter is caused by the modulation of the VCO frequency with secondary VCO frequency, crystal oscillator and other noise. This can account for another 2–5% percent of error.

PLL Response: The PLL response to a data bit shifted from its nominal position because of noise or jitter is directly translated in a margin loss for the bits following any shifted bit. For a highly accurate PLL circuit this is the primary source of error, and it typically results in a window loss of up to 20%, depending on data pattern and frequency variation constraints.

All of these degradations are summed into the Window Margin specification.

$$\theta_{wm} = (\tfrac{1}{2}\ \text{Bit Window}) - \theta_{e(PLL)} - \theta_{e(SWL)} \qquad (18)$$

This yields the margin loss, where θ_{wm} is the total window margin or the total amount of a half bit cell in which a data pulse will be properly recognized, $\theta_{e(PLL)}$ is the error due to PLL response, and $\theta_{e(SWL)}$ is the total error contributed by imperfections of the PLL circuitry (including delay line accuracy, leakage, and noise).

The window margin loss contributed by the device accuracies are relatively straightforward, and are supplied by National. The more difficult task is to determine the window margin loss due to the PLL response.

Tracking of the Disk Data

The bit shift produced by an average disk depends on the pattern of encoded pulses recorded on the media. Pulses that are placed close together appear to push each other apart when they are read back. This is the primary cause of bit shift.

The PLL tolerance to bit shift is dependent on the amount of data bits that are shifted, and the data pattern. It can tolerate more bit shift by individual bits if only a few of the bits within a pulse stream are shifted. However, if most of the data read by the PLL is shifted (both early and late), the loop is constantly correcting itself and is never really phase locked. Because it is not phase locked the maximum tolerable bit shift is less.

Some data patterns are better at determining PLL performance than others. These are patterns that are particularly difficult for a PLL remain locked to while the data pulses are shifted early and late. For example a bit pattern of "11000" is difficult to decode since it has pulses alternately spaced by 1 and 1.5 bit cells. See *Figure 13a*. This is difficult to decode in some cases because under maximum bit shift conditions all pulses are equally spaced.

Another bit pattern that is difficult to decode is a repeated bit pattern triplet of "110" (or "101" also referred to as a "DB6" pattern), which has a pair of pulses one bit cell apart, and the next pair two bit cells apart. This pattern is particularly difficult to decode because it contains two pulses of minimum spacing, followed by two maximum spaced pulses. This pulse pattern is shown in *Figure 13b*.

Unlike the acquisition process described earlier, the PLL must largely ignore individual bit shifts during the tracking phase. The PLL should only follow the longer term average

data rate. The desired PLL response during the tracking phase is somewhat different from the response required during the acquisition phase. Instead of a high bandwidth to decrease the lock time, a low bandwidth is preferred to prevent the PLL from following individual bit shifts. When choosing the filter bandwidth, the lowest possible value should be used that still satisfies the acquisition time requirement.

Figure 14 can be used to determine the theoretical window margin. This curve of phase bit shift resistance plots the amount of error introduced in the PLL's VCO by a phase error. The following example will show the steps involved in calculating expected window margin, and illustrate some concepts of tracking data.

Example 2: Determine the total dynamic window margin for a loop with a bandwidth of 90 Krad/sec, a damping of 0.7, and at a 500 Kb/s data rate. The intrinsic circuit errors amount to 0.1 radians of the window. Calculate the margin for both the 110 and 11000 pattern.

First step is to calculate the bandwidth of the PLL while tracking these data patterns. The bandwidth is the square root of the ratio of the pulses per bit cell relative to preamble data, multiplied by the bandwidth. Preamble data has 1 pulse/bit cell, and a 110 pattern has 2 pulses per 3 bit cells, while a 1100 pattern has 4 pulses for every 5 cells. Thus:

$$\omega_{n(110)} = (90\ \text{Krad/sec}) \times \sqrt{0.66}$$
$$= 72\ \text{Krad/sec}$$
$$\omega_{n(11000)} = (90\ \text{Krad/sec}) \times \sqrt{0.8}$$
$$= 81\ \text{Krad/sec}$$

To analyze the problem, we need to look at two bits. The present bit which has an early phase step, and the next bit which has an equal late phase step. We must determine how large a shift in the first bit can be tolerated such that the same amount of shift in the second bit will still fall within the proper window. In equation form:

$$\theta_{e(2)} + K_{WM}\Delta\theta_{e(1)} + \theta_{PLL} \geq \theta_T \qquad (19)$$

where $\Delta\theta_{e(1)}$, the shift of the first bit, is multiplied by K_{WM}, which is the affect the first bit has on the VCO when then next bit arrives. θ_T is the total window available (in this case ±500 ns). θ_{PLL} is the static error degradation due to the DP847x and is 10% of 500 ns or 50 ns. $\Delta\theta_{e(2)}$ is the maximum phase error of the second bit. We can solve for $\Delta\theta_{e(2)}$ assuming that $\Delta\theta_{e(1)} = \Delta\theta_{e(2)}$:

$$\Delta\theta_{e(2)} = \frac{\theta_T - \theta_{PLL}}{1} + K_{WM} \qquad (20)$$

The value of K_{WM} is the value read off the y-axis of *Figure 14*, at a time normalized to ω_n. This time is the time between two bits or:

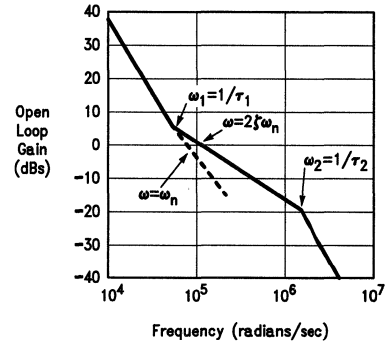
For a "110" pattern $\omega_n(t)$ = (4 μs) (72 Krad/s) = 0.29, resulting in a value from *Figure 14* of K_{WM} = 0.37. And for a "11000" pattern $w_n(t)$ = (3 μs) (81 Krad/s) = 0.24, which results in K_{WM} = 0.28. Using 0.37, the window margin is:

$$\Delta\theta_{e(2)} = \frac{500\ \text{ns} - 31\ \text{ns}}{1.37} = 342\ \text{ns}$$

In terms of percentage this is 100*(342/500)% = 68%. Note that this is the window margin at nominal frequency.

Open Loop Bode Plots and the Second Capacitor

Figure 15 shows the open loop gain Bode plot for the second order PLL. This plot is useful as a double check to make sure that we have a stable design and is important to show the affect of the second capacitor in the filter. In this plot, it is assumed that the charge pump output impedance is infinity.



TL/F/9419–18

FIGURE 15. Bode Plot of Open Loop Gain of DP8472/3/4 Using a Typical Filter at 500 Kb/s (from Example 1)

As can be seen, the gain starts off with a slope of 40 dB per decade due to the two poles of the filter and VCO. The phase angle starts at −180°C. The stabilizing zero is introduced at $\omega_1 = 1/R_2C_2$, and causes the slope to change to 20 dB per decade. ω_n is the extrapolation of 40 dB/dec line to 0 gain, and the actual crossing point is $\omega = 2\zeta\omega_n$. Example 4 discusses how to plot this curve.

The further reduce the effect of unwanted changes in the VCO phase the second capacitor can be added to the filter. This capacitor, C_1, introduces a pole, *Figure 15*, between the loop natural frequency and the data rate frequency. The pole due to the second capacitor occurs at $\omega_2 = 1/\tau_2 = 1/R_2C_1$. This capacitor provides further attenuation of bit shift caused frequency components, and the pump pulse noise, both of which have frequency components that are around the data frequency. The only considerations in choosing the value of this capacitor are related to the stability of the loop, and inadvertently affecting ω_n. A good criterion for stability is that the Bode plot of the open loop gain, *Figure 15* must cross the 0 dB gain with a slope of 20 dB/dec, ie. before the break caused by C_1's pole.

To determine a simple method for deriving C_1, we must look at the open loop gain of the PLL along with the transfer function of the loop filter. The open loop gain is:

$$\frac{\theta_2}{\theta_1} = \frac{K_D K'_{VCO}}{s} F(s) \quad (21)$$

Where F(s) is the filters transfer function:

$$F(s) = Z(s) = \frac{\left(\frac{1+sR_2C_2}{sC_1}\right)}{\left(\frac{sR_2C_1C_2}{(C_1+C_2)}\right)} \quad (22)$$

Combining these two equations and manipulating we find that the second pole occurs at:

$$\omega_P = \frac{1}{R_2C_1} \text{ (confirming } \textit{Figure 15}\text{)} \quad (23)$$

This is assuming $C_2 >> C_1$ (which we ought to assume to maintain the validity of previous filter assumptions).

The zero introduced by C_2 and R_2 should be designed to be close to the 0 dB gain crossing. Its frequency is $\omega_z = 1/R_2C_2$. The frequency of the pole due to R_2 and C_1 is approximately $\omega_P = 1/R_2C_1$. This pole must not significantly change the slope around the 0 dB line. If we choose $C_1 = C_2/20$ the effect on the slope of the transfer function is less than 1 dB/decade at the frequencies around the 0 dB gain line crossing. Thus as a guide:

$$C_1 \leq \frac{C_2}{20} \quad (24)$$

Example 3: For example 1, determine C_1.

Very simply for example 1a:

$$C_1 \leq \frac{0.039\ \mu F}{20} \cong 2000\ pF$$

The 1/20 factor provides the approximate value for C_1.

Example 4: Plot the Bode diagram of the open loop gain for the DP8472 with $C_2 = 0.027$, $C_1 = 1000$ pF, $R_1 = 5.6\ k\Omega$, $R_2 = 545\Omega$.

There are two easy methods of doing this. One method involves determining the open loop gain at a low frequency, where the poles and zeros don't have any affect, and then using this gain point at a start drawing the properly sloped lines to the break point frequencies. A second method is to calculate the ω_n, and $2\zeta\omega_n$ frequencies.

For the first method, first calculate the locations of τ_1 and τ_2 in *Figure 15*. Thus

$$\omega_1 = \frac{1}{\tau_1} = \frac{1}{R_2C_2} = \frac{1}{(545\Omega)(0.027\ \mu F)} = 6.8 \times 10^4$$

$$\omega_2 = \frac{1}{\tau_2} = \frac{1}{R_2C_1} = \frac{1}{(545\Omega)(1000\ pF)} = 1.8 \times 10^6$$

Now pick a point that is below ω_1, and calculate the open loop gain, which is:

$$K_{LOOP} = 20 \log \left[\left(\frac{K_{VCO}K_PV_{REF}}{2\pi R_1 N \omega} \right) \left(\frac{1}{\omega C_2} \right) \right]$$

At $\omega = 10^4$, the K_{LOOP} is:

$$K_{LOOP} = 20 \log \left(\frac{12 \times 10^6}{R_1C_2\omega^2 N} \right) = 39\ dB$$

Now draw a line from $\omega = 10^4$ to ω_1 with a 40 dB/decade slope. Then at ω_1 draw a line to ω_2 with a 20 dB/decade slope, and finally draw a line from ω_2 with a 40 dB/decade slope.

To understand the affect of C_1, the additional attenuation introduced can be determined as using:

$$A_P(w) = \frac{1}{1 + \frac{\omega}{\omega_p}} \quad (25)$$

Using our previous example 1:

$$\omega_P = \frac{1}{1000\ pF\ 545\Omega} = 1834\ Krad, \text{ and}$$

$$\omega = 2\pi\ (500\ kHz) = 3142\ Krad/sec.$$

Thus

$$A_P(w) = \frac{1}{1 + \frac{3142}{1834}} = 0.37$$

which yields an additional 9 dB of attenuation.

Choosing Component Tolerances and Types

One of the most often asked questions is how accurate should the filter and charge pump resistors and capacitors be? The answer depends on how accurate a data separator is required. For a good performance design, the following criteria can be followed for each component:

R_1: The pump set resistor's tolerance affects the loop bandwidth, ω_n. The loop bandwidth directly affects window margin. Due to the square root relationship, a 5% change in this resistor changes ω_n by 2.5% which in turn affects the window margin by 1–2%. It is thus recommended that R_1 be a 1% resistor. A standard carbon or metal film resistor with a low series inductance should be chosen.

C_2: The main filter capacitor also affects ω_n in the same way as R_1 so it too should be relatively accurate. 5% is recommended. This capacitor should be a very good quality capacitor, with good high frequency response and low dielectric absorption. Mica is a good choice although maybe too expensive. Polypropalene and metal film are good as well. Avoid Mylar or Polystyrene.

R_2: This resistor has a much lower affect on window margin, and thus standard 5% resistors can be used.

C_1: The second capacitor's accuracy in not critical 10%–20%, but its high frequency characteristics should be quite good, similar to a good high frequency power supply decoupling capacitor.

5.0 ADVANCED TOPICS

The following sections discuss several specialized areas of evaluation and design of the PLL for the DP8473 controller. Also a short discussion of the crystal oscillator design considerations is given.

5.1 Design and Performance Testing

Testing data separators can get rather complicated. Once the PLL circuitry, gain, bandwidth, and damping are set, then data separator lock range testing, window margin testing, and finaly bit error rate testing may be undertaken. This testing can require some fairly sophisticated setups, and is time consuming. To help the designer get started, a rigorous approach to floppy disk PLL design verification is described with reference to desired performance and available equipment. If the designer is not concerned about optimum performance for custom applications, then the values for the PLL filter and pump resistor provided in the DP8473 datasheet should prove more than adequate.

Step 1—Calculating the Filter/Pump Resistor: Following the examples above, the optimum "paper design" filter components should be calculated (or use the values provided in the datasheets and tweak them).

Step 2—Testing Lock Range and Damping: For characterization of the PLL's acquisition, a fairly simple setup can be used, which utilizes a pulse generator to provide a pulse train that simulates a preamble to be input into the data separator's read data input. A second synchronous square-wave that is 20–50 times the period of the data pulses is applied to read gate. The frequency of the read data pulses should be varied from the nominal data rate to the limits of the desired lock range, and the lock range requirement should be verified by monitoring the Filter pin, and the Pump outputs. *Figure 16* shows a typical setup, and some typical waveforms on the pump and filter pins during acquisition. *Figure 16b* shows a proper locking PLL which does not exhibit "cycle slipping". Cycle slipping is denoted by the saw tooth waveform on the filter pin, which can be seen by the locking shown in *Figure 16c*.

In both *Figure 16b* and *c* the total amplitude, ΔV, of the filter pin waveform is typically less than 200 mV.

The object is to adjust the PLL bandwidth to ensure that when locking over the desired range of data rates (for example 500 Kb/s ±6%) that no cycle slipping occurs. If slipping does occur then the bandwidth should be increased.

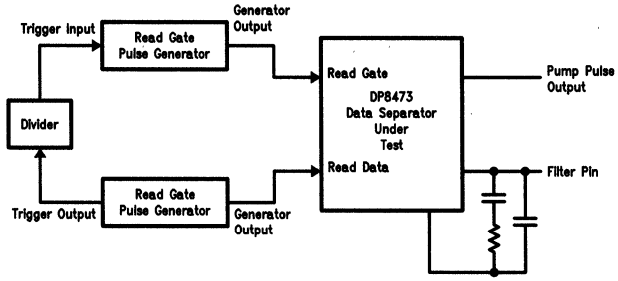
A second piece of information that these curves provide is verification of the damping of the PLL. As in *Figure 16b* the filter pin waveform should slightly overshoot, and then (maybe) slightly undershoot the eventual locked voltage. If there is very little overshoot then the loop may be overdamped, and if the filter pin voltage "rings" for a few cycles the loop is probably underdamped. For example, *Figure 16c* not only cycle slips, but does not overshoot, therefore the loop bandwidth may be fine, and the loop is just too heavily damped.

Step 3—Window Margin Evaluation: Usually to perform this test a disk simulator should be used to simulate the worst case drive read data conditions, and measure the error performance of the data separator. This simulator can be used to vary the following parameters:

1. Motor Speed Variation
2. Instantaneous Speed Variation
3. Instantaneous bit shift
   (to determine the window edge)
4. Data pattern.

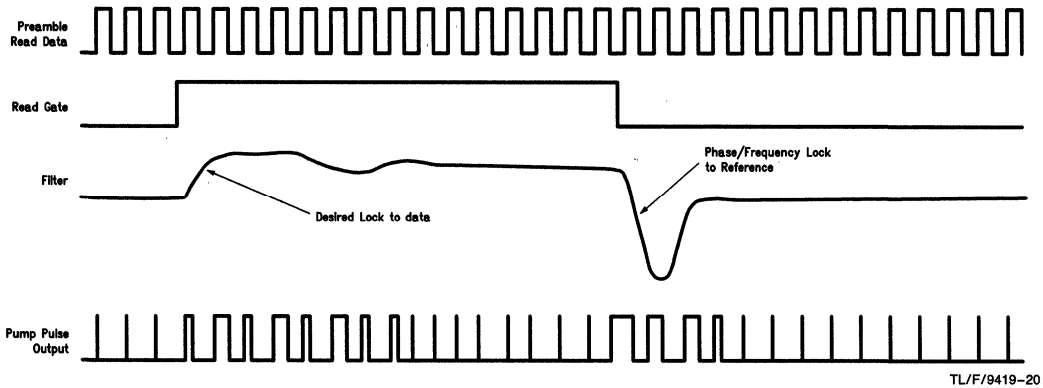
The test setup shown in *Figure 17* can accomplish some of this testing. The disk simulator looks like a formatted disk to the controller/data separator. To test the data separator, software in the host computer performs a repeated series of read operations over a period of time, while the designer programs the disk simulator to vary the data rate from one end of the lock range to the other. At each data rate the bit shift is increased until the error rate increases above a minimal threshold.

This testing should measure window margin over the entire lock range, and under conditions as described in section 2.4. Typically this process is a trial-and-error process. The bandwidth and damping can be adjusted based on the results of these tests.



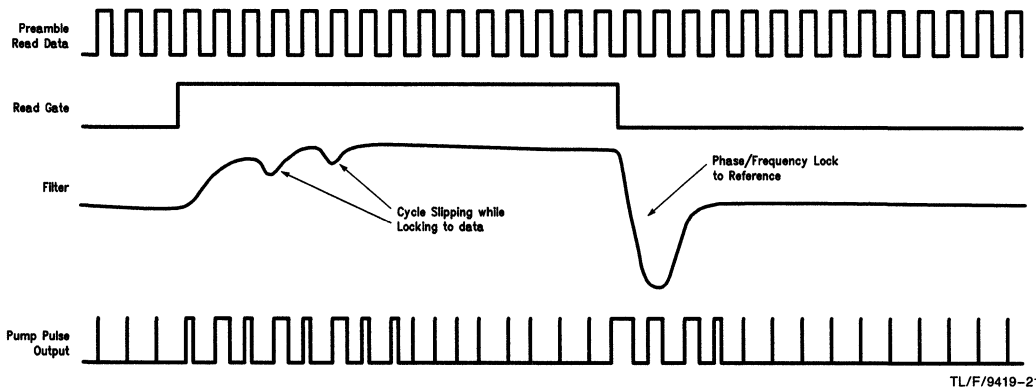
TL/F/9419–19

(a)



TL/F/9419–20

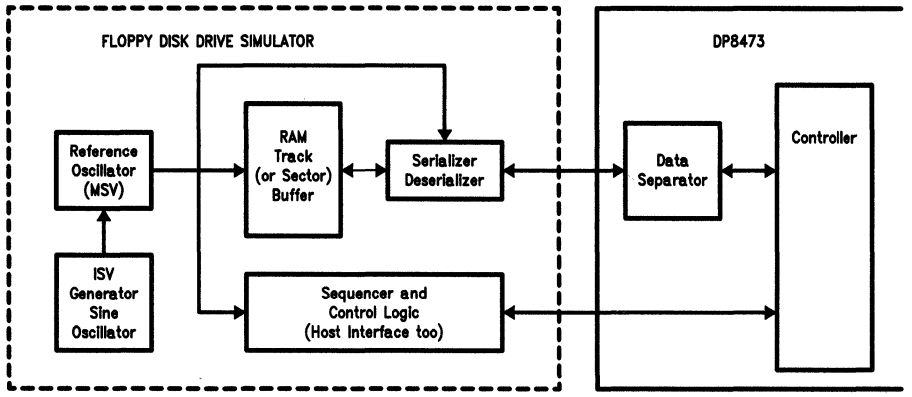
(b)



TL/F/9419–21

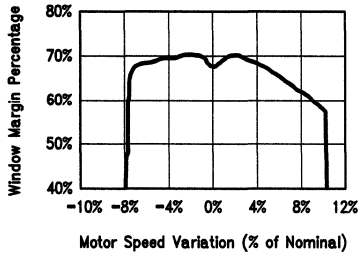
(c)

FIGURE 16. Simple PLL lock/performance testing; a) Typical frequency generation hardware to generate read gate and preamble for various frequencies. b) Using this hardware a typical lock acquisition showing key signals. This is a proper lock waveform. c) This shows an unreliable lock to a frequency beyond the lock range of the PLL. Cycle slipping occurs because the PLL is unable to respond quickly enough to the frequency step.



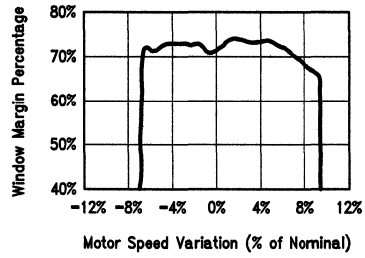
TL/F/9419–22

FIGURE 17. Block Diagram of Connection of Disk Simulator to Data Separator to be Tested



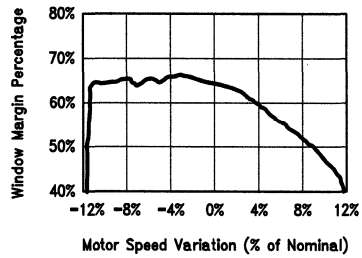
TL/F/9419–23

(a)



TL/F/9419–24

(b)



TL/F/9419–25

(c)

FIGURE 18. Various window margin graphs. a) Data separator with slightly long delay line and fairly normal lock range; b) data separator with short delay line; c) a relatively poor data separator with a long delay line, and wide bandwidth.

ISV may also be simulated. Unfortunately most disk simulators do not easily test for this. Therefore it is very likely that the window margin measurement will be made with only MSV variation initially. This may also be desirable since MSV-only testing will yield a better understanding of the PLL's lock range. If MSV-only testing is done initially then the design of the PLL and total frequency range for evaluation should be the sum of the desired MSV and ISV. For example, a design requirement of ±3% ISV and ±3% MSV can be approximated by ±6% MSV.

If an MSV-only test is done, it may be useful to then follow this with a simple ISV test just to double check that the loop will follow the ISV properly with little degradation in window margin. It is also sometimes useful to vary the ISV frequency until the window margin degrades, just to give an indication of how high the ISV frequency can go. A typical PLL's ISV performance should not degrade until the frequency is greater than about 800 Hz.

Step 4—Considering Temperature, Supply and Device Variations: The next step is to take the above "optimum" filter and simulate variations in gain induced by temperature, supply and processing. This is accomplished with the same disk simulator setup as in step 3. To simulate these variations, the designer need only vary the pump resistor's value. This will affect the open loop gain identically to device variations. The above tests should be re-run with a minimum and maximum resistor. If the performance "falls-off-a-cliff" then some compromises and adjustments to the filter may be required.

For relatively small temperature ranges a resistor variation of ±10–15% should be adequate. For full 0°–70° simulation, resistor variation of ±20% is a more accurate reflection of the DP8473 performance. If for some reason the performance of the data separator cannot be maintained at the desired window margin, then trimming the pump resistor may be needed to meet performance over the full process spread.

The final simulation involves varying the delay line length. It is possible to simulate variations in the delay line by forcing a leakage current onto the filter pin (at the VCO input) using a pull up or pull down resistor. This leakage current will cause a phase error within the loop and will cause the loop to act as if the delay line length were changed. Before actually running dynamic window margin tests, the designer must determine the actual length of the delay line, and then adjust this length to the limits specified in the datasheet. Then at each limit the dynamic window margin test can be performed.

In order to measure the length of the delay line a static window margin test can be performed. This test uses a disk simulator with a format that has all 0's or 1's data fields. Within the data field one bit is shifted until the PLL mis-decodes the data. Using the maximum tolerable bit shift number, the delay line length can be extrapolated. This same measurement can be done with various filter pull up/down resistors. By proper resistor selection the limits of the delay line length can be simulated. Using these simulated delay line techniques a dynamic window margin test can be run, and the resultant PLL performance can be characterized.

Step 5a—Bit Error Rate Measurements: This test is performed as a verification of the final total system. It consists of putting together a complete floppy drive, floppy media, separator, and controller system then running long term read/write tests randomly across the disk media. For a known number of read/writes and the resultant value of read errors, a number can be derived that is the ratio of bit errors to total number of bits read. This test proves the integrity of the entire system, and should be performed over some manufacturing spread of products. While this test is useful to verify the complete system integration, the data separator is only one small part of the total contribution to the total system error rate.

Step 5b—Real World Worst-Case Tests: As a final analysis and proof that the data separator is solid it is often useful to test the data separator on a known "worst case" drive and disk. Generally the evaluation is done with a disk that is recorded off speed and off track. This ensures that a maximum amount of bit shift and speed variation is present. The main problem is to ensure that the disk still has acceptable data. If excessive errors are encountered, evaluation of the types of error that are occurring can be used to determine whether the bandwidth of the PLL needs to be increased (acquisition related errors) or decreased (bit shift related errors).

Alternate Simple In-System Data Separator Evaluation

Provided here are some quick tips on evaluating the data separator without any test equipment, but just by running long term in-system tests.

1. If after some initial testing a lot of sector or ID address mark or data address mark not found errors are given by the controller, then in all likelihood the data separator is not locking properly. The bandwidth of the loop should be increased, or possibly the loop is too heavily overdamped.
2. If the disk controller is experiencing a large amount of address or data field CRC errors, then probably the loop is being sent out of lock by bit shift noise. Thus, the gain of the loop may be too high or the loop is underdamped.

5.2 Understanding the Window Margin Curves

Careful analysis of a window margin curve can yield quite a bit of information about the data separator characteristics. *Figures 18a*, *b*, and *c* show some typical window margin plots. These curves were made using a disk simulator that outputs a "reverse write precompensated" bit shift pattern to the data separator.

These curves plot the maximum bit shift tolerance (vertical axis) versus motor speed variation (MSV-only) (horizontal axis). The controller was programmed to perform a repetitive single sector read, while the simulator outputs a formatted track at the programmed MSV and bit shift amount. All the data read with MSV and bit shift amounts that fall under and within the curve (shaded area) can be consistently read correctly. All data read with MSV and bit shift amounts outside and above the curve either could not be correctly located by the controller, or had errors in it.

The first thing to note is that as the MSV variation from the nominal frequency increases, there is a point at which the PLL performance drops to zero bit shift (the vertical lines of the curve). This is an indication of the lock range of the PLL. Thus for *Figure 18a* the lock range is −8% to +10%. This asymmetry in the lock range is typical of the DP847x series PLLs and is due to a slight skew in the charge pump. This

skew adds some bit shift rejection improvement. *Figure 18b* has a lock range of −7% to +9%, while *Figure 18c* has a much wider lock range of nearly ±12%. This is an indication of the loop bandwidth. Here *Figure 18a* has the lowest bandwidth, then *Figure 18b*, and finally *Figure 18c* has the highest bandwidth.

Another characteristic of each of these curves is the downward slope in the positive MSV direction. The start of this slope is the point at which the delay line becomes longer than the ¼ period of the data rate. For example, in *Figure 18a* the slope starts at about 0 MSV (ignoring the dip at zero MSV for the moment which is due to some internal noise). This indicates that the delay line is a quarter period of the nominal data rate. Unfortunately, this causes a degradation in performance at higher MSV. In *Figure 18b* the slope starts at about +3%–4% MSV, and so the delay line is about 3%–4% short at a nominal data rate. This is an optimal delay line length to maximize performance over a ±6%–8% lock range. In *Figure 18c* the delay line is about 3%–4% too long and so there is quite a bit of degradation in window margin in the +MSV portion of the curve.

A final observation is that the wider the bandwidth the lower the window margin, assuming other things like data rate remain constant.

Another interesting fact to note is the type of errors that occur when the bit shift/MSV exceeds the PLL performance. Generally to the left or right of the curve (extreme MSV) the controller will give "Address Mark not Found" errors which is an indication of the PLLs' inability to lock properly. Errors for bit shift that exceeds the curve but for MSV within the PLL's lock range are generally CRC errors, although at very high bit shift a mixture of CRC and Address Mark Errors are expected.

5.3 DP8473 Filter Switching Design Considerations

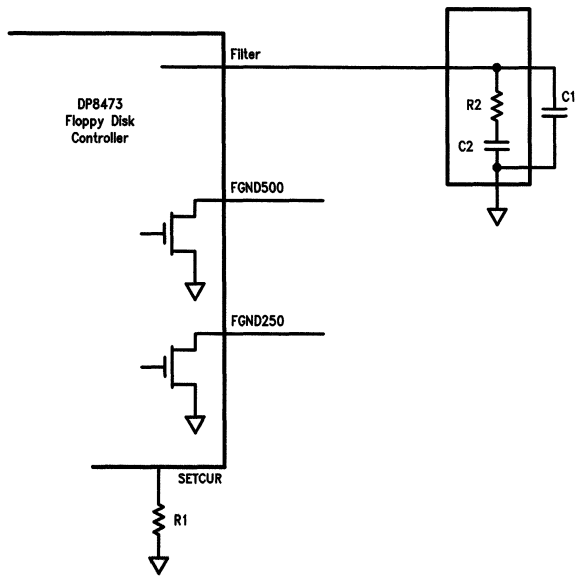
Due to the desire to handle multiple data rates, the DP8473 incorporates on-chip data rate selection logic, and also filter switching logic. In this section we will discuss how this logic works and how to design a set of filters to maximize performance at various combinations of data rates.

Designing with a Single Filter

Previous design examples have dealt with optimization of a single filter at a single data rate. If the DP8473 is to be used at one data rate, the circuit connection is shown in *Figure 19*, and its design is straight forward. It is possible to use a single filter and obtain a reasonable performance at two data rates (i.e., 250 Kb/s and 500 Kb/s or 500 Kb/s and 1 Mb/s). This can be accomplished since the loop bandwidth is scaled by the PLL's divider. Since the divider value increases by a factor of two when going from the high to the low data rate, the bandwidth scales by:

$$\omega_{n250} = \frac{\omega_{n500}}{\sqrt{2}}$$

This scaling is probably not enough to optimize the lower data rate and the damping is affected too, but the single filter approach can still provide acceptable performance in many instances.



TL/F/9419–26

FIGURE 19. DP8473 Filter Configuration for a Single Data Rate Filter, May Be Used as Compromise Filter for Any Two Data Rates

Designing for 250K/300K/500K MFM

This next filter configuration allows fewer compromises (and hence better performance) when using all 3 PCAT data rates. This configuration is shown in *Figure 20*. This circuit uses two pairs of R's and C's that compose two independent filters. One filter is connected to the FGND250, and the other to FGND500.

To implement the design of *Figure 20*, first design single optimum filters at 250 Kb/s, 300 Kb/s, and 500 Kb/s. Use a single pump resistor for all data rates. Verify and tweak the performance of these filters individually. The 500 Kb/s filter can be directly used. The 250 Kb/s and 300 Kb/s individual filter values need to be compromised into a single R and C filter. C_1 should be $1/20^{th}$ of C_2, and if desired C_3 can be added with a value that is $1/20^{th}$ of C_4.

Designing for 1.0 Mb/s and a 2nd Data Rate

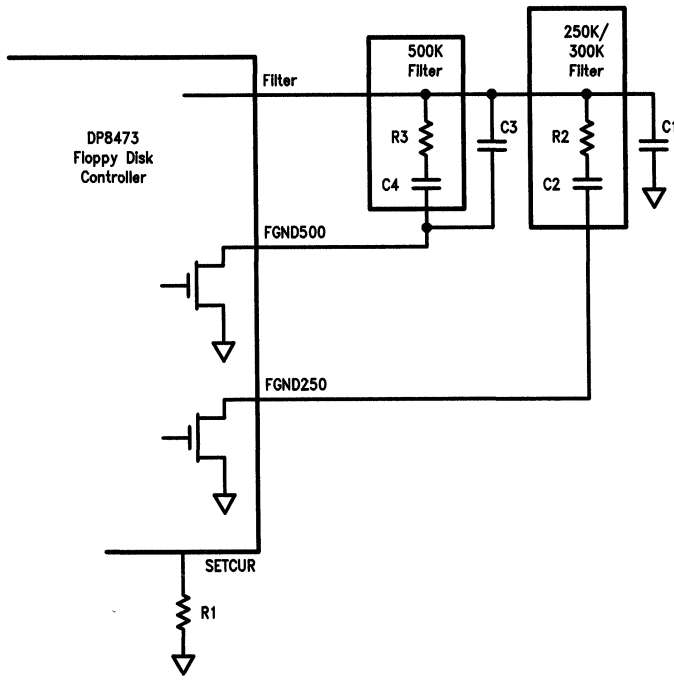
By adding an additional capacitor to *Figure 19* 1.0 Mb/s data rate can be supported, as shown in *Figure 21*. In this figure a single damping resistor is used, but depending on the chosen data rate, one or both capacitors is selected. This configuration allows the bandwidth to be adjusted more flexibly when designing for the various data rates. The design process is, however, a little more complex.

To implement the design of *Figure 21*, first design single optimum filters at 1.0 Mb/s and 500 Kb/s (or 250 Kb/s). Use a single pump resistor for all data rates. Verify and tweak the performance of these filters individually. Using these values, choose a value for R_2 (damping resistor) that is a good compromise for all data rates. Next choose C_2 to be the optimum value from the initial individual design verification of the 1.0 Mb/s design. Next choose C_3 such that the sum of C_2 and C_3 equals the value for the optimum 500 Kb/s (or 250 Kb/s) design. Finally, choose C_1 to be $1/20^{th}$ of C_3.

Designing for All Possible Data Rates

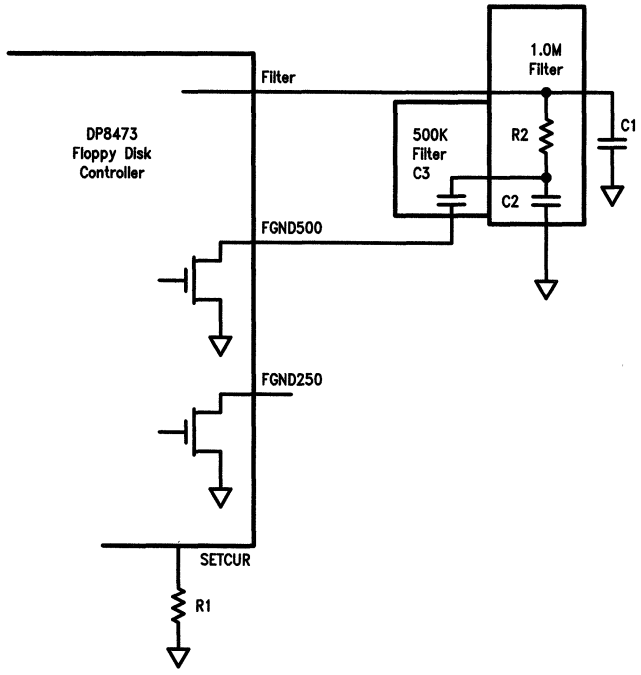
To support all possible data rates the simplest circuit configuration is one similar to *Figure 20*, but with an additional capacitor selected for the 1 Mb/s data rate.

To implement the design of *Figure 22*, first design single optimum filters at 250 Kb/s, 500 Kb/s, (300 Kb/s also if needed), and 1 Mb/s. Use a single pump resistor for all data rates. Verify and tweak the performance of these filters individually. Using all of these values, choose a value for R_2 (damping resistor) that is a good compromise for all data rates. Next choose C_2 to be the optimum value from the initial individual design verification for 1 Mb/s data rate filter. Next choose C_3 from the 500 Kb/s initial design. C_3 should be chosen such that $C_2 + C_3$ equals the optimum 500 Kb/s filter value. Then the 250 Kb/s (and 300 Kb/s if used) filter capacitor, C_4, must be chosen in a similar manner. C_4 should be chosen such that $C_4 + C_2$ equals the optimum 250 Kb/s filter capacitor value, or if using 300 Kb/s it must equal the best compromise 250/300 Kb/s filter capacitor. C_1 should be chosen to be $1/20^{th}$ of C_2.



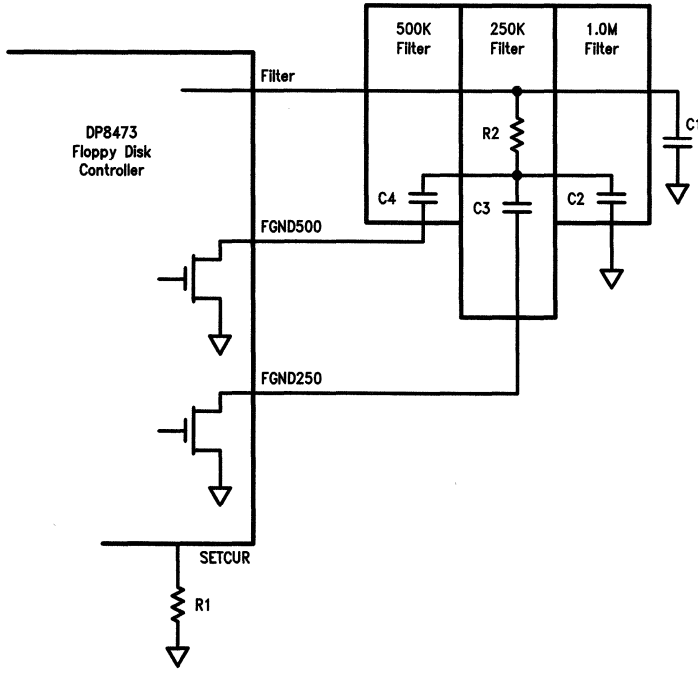
TL/F/9419–27

FIGURE 20. DP8473 Filter Configuration for Optimum 250/300/500 Kb/s (2 Filter) Design



TL/F/9419–28

FIGURE 21. DP8473 Filter Configuration for a Slight Tradeoff Filter Design at 1 Mb and 500 Kb/s



TL/F/9419–29

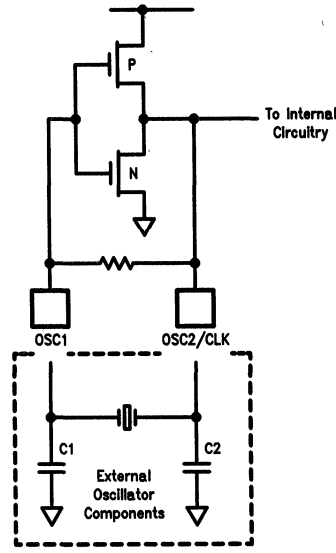
FIGURE 22. DP8473 Filter Configuration for All Data Rates

The previous filter does, however, have some minor performance tradeoffs if all data rates are to be implemented. If the best performance is desired, then the configuration shown in *Figure 23* should be used. In this figure, 3 individual filters are used for each of the data rates. The 1.0 Mb/s filter must be switched via some external circuitry, labeled "ground switch" in the figure. The circuit should enable this filter only when 1 Mb/s data is used, and this filter should be disabled when any other data rate is needed. The circuit to accomplish this could be as simple as an open collector gate derived from the RPM/LC pin, or an alternative that uses no additional hardware is to use an unused drive select output. If the latter option is chosen, then software will have to select the 1 Mb/s filter prior to using this data rate by enabling this bit in the Drive Control Register.

The design of this filter network is very straightforward. Simply design and optimize each filter individually, and use these filter values directly. (Again if 300 Kb/s is also used the 250 Kb/s filter used will be a compromise between the optimal 250 Kb/s and 300 Kb/s filters derived individually.)

5.4 DP8473 Oscillator Design

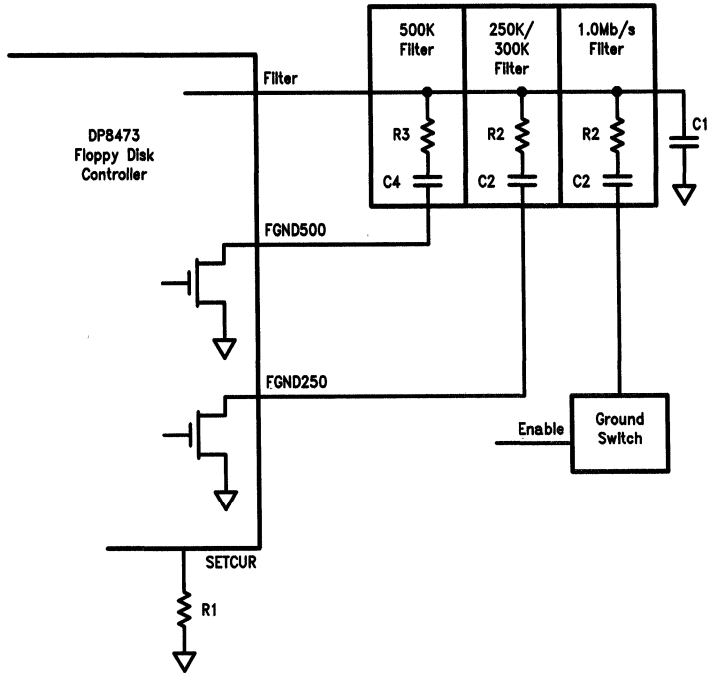
Figure 24 shows the schematic of the crystal oscillator used on the floppy disk controller. This circuit consists of a simple inverter whose impedance has been optimized for use as an oscillator. The inverter is biased into its linear operating region by a high value (>1 MΩ) resistor that is in parallel with the crystal. This biasing allows the inverter to operate as a simple inverting linear gain element.



TL/F/9419–31

FIGURE 24. Simplified Schematic of Oscillator Circuit for DP8473

The DP8473 oscillator is intended to be used with a fundamental mode parallel resonant crystal. The only external components required are the crystal and two external capacitors. These capacitors are usually very small (picofarads).



TL/F/9419–30

FIGURE 23. Filter Configuration for Optimal Filter Design at All Data Rates

TABLE II. Important Parameters for Crystal Selection

Parameter	DP8473
Crystal Frequency	24 MHz
Oscillatory Mode	Fundamental
Oscillator Resonance	Parallel
Accuracy	<0.5%
Series Resistance	<100Ω
Shunt Capacitance	<7 pF
External Parallel Capacitors (include parasitics)	10 pF

Table II shows the important parameters to check for when selecting a crystal to use with the DP8473. While the recommended resonance mode is parallel, a series resonant crystal can be used. It will just oscillate in parallel mode 30 ppm–300 ppm from its ideal frequency.

If an external oscillator circuit is used, it must have a duty cycle of at least 40%–60%, and minimum input levels of 2.4V and 0.4V. The controller should be configured so that the clock is input into the OSC2 pin, and OSC1 is tied to ground.

5.5 Trimming for Perfection

The integrated data separator was designed to achieve excellent performance. However, product, temperature, and power supply variations can degrade performance somewhat. This can lead up to a 10% variation in window margin performance. While this is still exceptional for any analog design, it is possible to trim out this variability.

The two major factors that contribute to data separator performance degradation are: 1) Loop Gain variation; 2) ¼ period delay line length variation.

Trimming the Loop Gain

The loop gain variation can be trimmed by replacing the pump resistor, R_1, with a variable resistor. This resistor should be trimmed based on the ideal lock range of the PLL desired. For example, if a ±6% lock range is desired, then during final board product test, a tester can be used to measure the total lock range and R_1 can be adjusted larger to reduce lock range, or smaller to increase it.

Trimming the Quarter Period Delay Line

The perceived length of the quarter period delay line can be modified by causing a static phase error in the loop to compensate for the quarter period delay line's error. This is accomplished by placing a pull up or pull down resistor on the filter pin. This resistor can be adjusted by measuring the Static Window margin for both an early and late single bit shift. Based on these measurements the delay line can effectively be adjusted by changing the value of the filter pull up/down resistor.

5.6 Initially Unlocked Model

This section is provided in order to complete the full discussion of the theoretical operation of a data separator. It is useful to discuss how the controller locks back to the crystal/clock reference when it needs to. This operation is taken care of by the controller so that the user need not concern himself with the design aspects of this section. However, if the user desires a more complete understanding of the entire lock process that the data separator goes through, this section is presented.

Another model is used to analyze the behavior of the PLL in an unlocked state. It is assumed in this model that the loop is not locked, and the VCO frequency is different from the input frequency. This model can be used to evaluate how the PLL re-locks to the reference clock after reading bad data and being thrown off frequency.

The first operation of the PLL is to frequency lock, so for this model each block is described as a function in terms of frequency, not phase. An equation can be derived for the frequency error that is similar to equation (2) for the phase error:

$$K_e = \frac{\Omega_e}{\Omega_1} = \frac{1}{[1 + K_{DF} K'_{VCO} F(s)]} \tag{26}$$

In the initially unlocked model, the phase detector has a key role. The phase detector compares the VCO output with the input signal. If the VCO output rising edge is leading the input signal, a pump-down signal is generated from this edge of the VCO to the next rising edge of the input signal. If the VCO is lagging the input signal, a pump-up signal is generated from the edge of the input signal to the rising edge of the VCO.

There is no overshoot of the VCO frequency. Only one type of pump signal is generated, up or down, to bring the VCO frequency toward the input frequency. For example, if the VCO frequency is higher than the input frequency, only pump-down signals are generated. It can also be seen that the larger the frequency difference between the VCO and the input, the longer the pump pulses become. The average current flowing from the charge pump is roughly proportional to the frequency difference of the signals at the input of the phase (and now also frequency) detector. The phase detector gain is:

$$K_{DF} = \frac{I_{PUMP}}{\Delta\omega} \tag{27}$$

for $\omega_2 < 2\omega_1$, where $\Delta\omega = \omega_2 - \omega_1$, and

$$K_{DF} = -I_{PUMP} \tag{28}$$

for $\omega_2 > 2\omega_1$. Therefore, if ω_2 is not too far from ω_1, the expression (8) can be written for our PLL as:

$$K_e(s) = \frac{\dfrac{s\Delta\omega\, C_2}{K'_{VCO}\, I_{PUMP}}}{1 + s\left(R_2 C_2 + \dfrac{\Delta\omega\, C_2}{K'_{VCO}\, I_{PUMP}}\right)} \tag{29}$$

This expression will allow the calculation of the time that the loop requires to lock back to the reference after a read operation goes through a bad data field or write splice.

Acquisition to the Crystal

After the completion of a read attempt, it is important to ensure that under the worst case conditions the PLL will properly re-lock itself to the reference clock. In order to achieve the required performance during the acquisition to the data stream, the PLL must have reached the lock to the crystal before it is allowed to lock back to the data.

If the PLL attempts to lock to a write splice the VCO may be pulled way off frequency. To prevent this, read gate should be deasserted as soon as a wrong or bad data field is detected. This will prevent the VCO frequency from being pulled too far away. The DP8473 read algorithm has been optimized to prevent this.

If the PLL is locked to the data stream, when read gate is deasserted, the lock mechanism to lock back to the reference frequency is quite similar to locking to the data. If the frequency of the VCO has been swept way off frequency because of a bad data field, noise, write splice, or missing data, the unlocked PLL model must be used. Since when locking to the crystal the phase-frequency comparison is always enabled, the phase detector acts as a frequency discriminator. Switching to the crystal imposes a frequency step to the PLL. It can be demonstrated that the frequency error generated is an exponential function of time, going to 0 with the time constant of:

$$T_P = R_2C_2 + \frac{C_2\Delta\omega R_2 N}{K_{PLL}} \qquad (30)$$

Thus T_P can be assumed to be the worst case acquisition time to the reference clock.

Example 3: From the previous example 1, we have chosen the values for the components of: $C_2 = 0.039\ \mu F$, $R_2 = 535\Omega$. We would like to find the worst case time required to re-lock to the crystal. Assume that the maximum frequency range of the VCO is ±30%.

If the VCO is pulled 30% off center, then:

$$\Delta\omega = 2\pi(500\text{ kHz})\ (0.30) = 942\text{ Krad/sec}$$

Using equation (10) for example 1, we can obtain:

$$T_P = (0.039\ \mu F)(545\Omega) + \frac{(0.039\mu)(942K)(5.6K\Omega)(8)}{(75\text{ Mrad})}$$

$$T_P = 41\ \mu s$$

This is about 4 byte times. Thus, read gate must be deasserted for this length of time before re-asserted to assure that the PLL has relocked to the reference. Most floppy controllers de-assert read gate much longer than this, and the DP8473 deasserts its internal read gate for 6 bytes.

Bibliography

Gardner, Floyd M. PhD., *Phase Locked Techniques*, 2nd edition. New York: John Wiley and Sons, 1979

Best, Roland E., *Phase Locked Loops*, New York: McGraw-Hill, 1984

DP8470 Phase Locked Loop Data Sheet, National Semiconductor Corp. 1986,7

DP8472 Floppy Disk Controller PLUS Data Sheet, National Semiconductor Corp. 1986,7

DP8473 Floppy Disk Controller PLUS/2 Data Sheet, National Semiconductor Corp. 1987

Design Guide for DP8473 in a PC-AT

National Semiconductor
Application Note 631
Robert Lutz

OVERVIEW

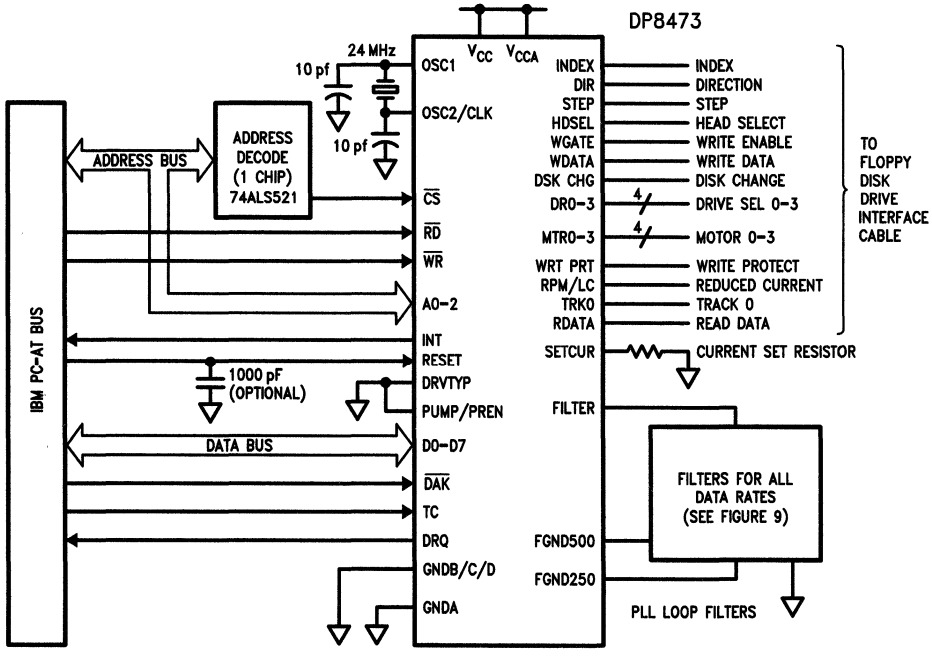
When designing a floppy interface circuit for a PC-AT in the past, there was very little flexibility given to the design engineer. The NEC765, which was designed into the original IBM PC, was the only floppy controller available that would guarantee compatibility. Compatibility is extremely important in a PC design.

There were many design issues that had to be resolved when using the NEC765 to produce a fully functional floppy controller interface. A data separator had to be selected or designed. A write precompensation circuit needed to be included. A whole score of miscellaneous logic had to be designed to handle all of the unique PC functions that the NEC765 does not handle itself.

The DP8473 from National Semiconductor was developed to eliminate all of these design problems. All of the extra functions and logic normally required for an XT or an AT design were included inside the chip. Even an analog data separator, which is classically the hardest function to design, has been integrated into the DP8473.

Compatibility has been completely retained. The DP8473 is software compatible with the NEC765A. We have not found any current software available for the PC that fails to work properly with the DP8473. This includes software running under DOS and OS/2.

This application note will discuss some of the issues involved in a floppy controller design with the DP8473. Even though on the surface, there may not seem to be many options when designing a floppy controller for a PC, there really are quite a few. Some of these options include: signal swapping in the floppy cable, different types of floppy drives, and data separator filter selection.



TL/F/10458–1

FIGURE 1. Schematic of Typical DP8473 Floppy Controller Design

HARDWARE ENVIRONMENT

A typical floppy controller design with the DP8473 will look something like the schematic shown in *Figure 1*. You may be surprised that the entire schematic for the floppy controller design fits on less than one page. Especially if you consider that the schematic for a similar function in the IBM PC-XT technical reference manual takes four full pages.

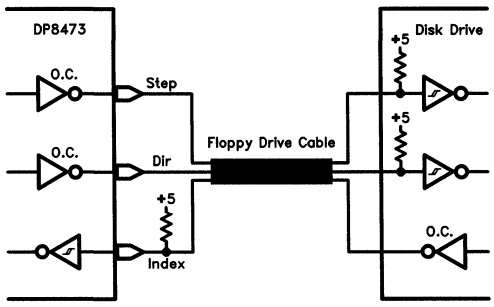
The heart of the design is, of course, the DP8473. Most of the interface pins to and from the DP8473 go directly to the peripheral bus or the disk drive cable without additional logic or buffering.

DRIVE CABLE INTERFACE

The DP8473 disk interface signals connect directly to the drive cable. Most disk drives terminate the drive cable with resistors. Termination is required because the output buffers of the floppy controller are open-collector.

Terminated signals are used because historically, relatively long cables have been used to connect the floppy controller to the disk drives. The cable termination will decrease the amount of crosstalk and noise on the drive cable signals.

A typical disk interface circuit consists of an open-collector output buffer at the signal source, and a termination resistor and a Schmitt input buffer at the signal destination. For example, the STEP output pin on the DP8473 is an open-collector output buffer that is capable of sinking up to 48 mA (See Note 1). If the output is off, the buffer is disabled, and the termination resistor on the disk drive will pull the signal high. If the STEP output is on, the DP8473 buffer will pull the signal low. An example of the cable termination logic is shown in *Figure 2*.



TL/F/10458–2

Note 1: The DP8473 actually contains open-drain output buffers due to its CMOS design. The end result is the same as TTL open-collector buffers.

FIGURE 2. Example of Buffers and Terminators Used for Floppy Drive Interface

The termination resistors used with 5.25″ drives or 8″ drives typically have a value of 150Ω. With this resistor value, the output buffers must be capable of sinking about 35 mA each in order to pull the drive signal to a logic low level. The DP8473 is able to sink this current without external buffers.

The termination resistors used with 3.5″ drives are often 1 kΩ. 1 kΩ termination resistors are also sometimes found on low power 5.25″ drives. Drive manufacturers have recognized that the floppy interface cable used in a PC is relatively short. Also, the drives are installed in the same grounded enclosure as the PC and the floppy controller. This reduces the amount of noise introduced on the floppy interface cable.

If both 3.5″ drives and 5.25″ drives are to be used in an application, the termination resistors used with the DP8473 must be chosen carefully. The termination resistor value used with the DP8473 must be the larger of the termination resistors used on the drives. For example, if the 5.25″ drive has 150Ω termination resistors and the 3.5″ drive has 1 kΩ termination resistors, the termination resistors used on the inputs to the DP8473 should be 1 kΩ.

The termination resistors for the inputs to the DP8473 should be placed near the DP8473. The termination resistors for the outputs for the DP8473 to the disk drive are contained in the disk drive itself.

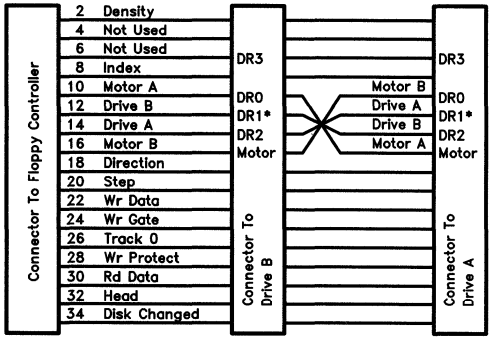
Additional disk interface buffering is not normally required when using the DP8473. It can sink up to 48 mA for each disk output signal. This is more than enough capacity for a typical floppy design.

DRIVE CONFIGURATION

A PC-XT can typically interface to up to four disk drives. A PC-AT, however, is usually limited to two disk drives. The two drive limitation is due to the ROM BIOS used in most AT's. More than two drives can be used if special software drivers are written.

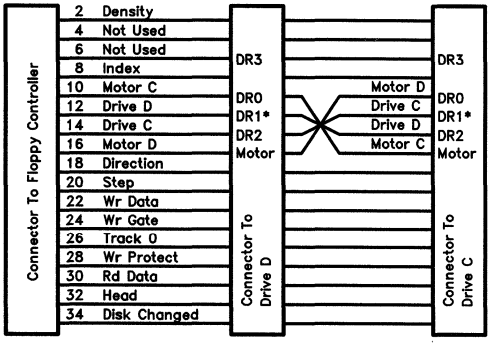
The connection between the floppy drives and the floppy controller in a PC is slightly different than the SA450 standard used in non-PC's. The advantage to the method used in a PC is that each disk drive installed in the PC can be configured identically. Even the Drive Select strap is the same. Each drive is configured as drive 1 (or B). Even if four drives are all connected, they are each strapped as drive 1.

The trick used to accomplish this feat is cable wire swapping. The drive cable is cut up and wires are moved around. The cable swapping re-routes the four DRIVE select signals (DR0, DR1, DR2, DR3) to the DR1 signal of each individual drive. For example, DR0 is routed to drive A's DR1 input. DR1 is routed to Drive B's DR1 input, and so on. In a similar manner, the cable swapping also re-routes the four MOTOR signals (MTR0, MTR1, MTR2, MTR3) to the MOTOR signal of each drive. *Figure 3a* demonstrates how the cable is configured for two floppy disk drives. A second cable would be used if more than two drives are required as shown in *Figure 3b*.



TL/F/10458–3

FIGURE 3a. Cable Swapping Used for Drives A and B



TL/F/10458–4

FIGURE 3b. Cable Swapping Used for Drives C and D

Note: The asterisk (*) next to DR1 indicates that this drive is strapped as Drive 1. In the PC, all drives are strapped as Drive 1.

If more than one disk drive is attached to the floppy controller, there may be more than one drive terminated. This may cause a current overloading problem. The controller may not be able to drive an active signal low.

It is easy to prevent current overloading in a two disk drive PC. Simply make sure that only one drive is terminated. Ideally the terminated drive should be the drive at the end of the drive cable, although it could be either drive.

If both drives are terminated, the output buffers will be driving too much current. The system will be out of specification. This is a common situation and is largely ignored by PC manufacturers. The output buffers can usually handle the additional load.

It three or four drives are to be used, things become more complex. For example, if four 150Ω terminated drives are all attached to the DP8473, the DP8473 will have to sink 139 mA for each drive interface signal. The DP8473 is guaranteed to sink up to 48 mA. Therefore, this configuration would not work without additional buffering. Please refer to the How to Calculate the Maximum Current Required for Output Buffers section for a description on current calculation.

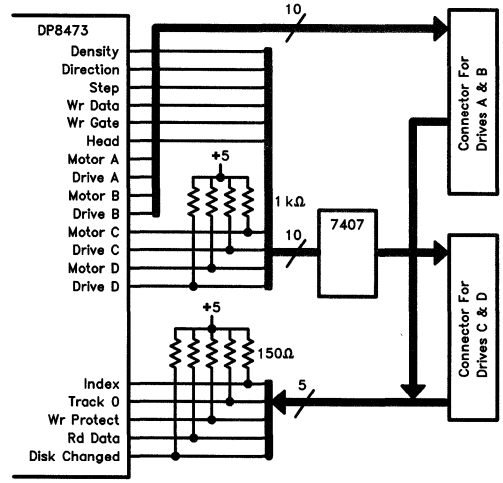
There are three techniques that can be used to prevent overloading the output buffers:

Technique 1:

Using larger termination resistors can reduce the load on the DP8473 to acceptable levels. If 1 kΩ resistors can be used instead of 150Ω resistors. In the worst case where all four disk drives are terminated, only 21 mA will be generated instead of 140 mA. This is well within the specification of the DP8473. 1 kΩ resistors are commonly used with 3.5" drives.

Technique 2:

The most direct technique that can be used is simply adding additional buffers for the extra disk drives. Drives A and B can be driven directly by the DP8473. The outputs to drives C and D can pass through an open-collector buffer such as the 7407. This is shown in *Figure 4*. 1k pullup resistors are required for some of the DP8473 outputs because they are not terminated by drives A or B.

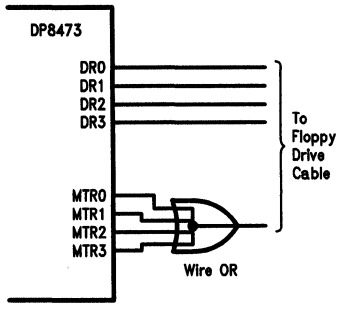


TL/F/10458–5

FIGURE 4. Extra Buffers Required for Four Drive System

Technique 3:

Daisy chain the floppy drives with the controller on one end of the drive interface cable, and one terminated drive at the other end. One to three additional non-terminated drives can be added in the middle as shown in *Figure 6*. With this technique, the four Motor signals from the DP8473 should be wire-ORed together as shown in *Figure 5*. Each drive must be strapped for the proper drive select (0–3).



TL/F/10458–6

FIGURE 5. Wire OR Required for Daisy Chain Connection

How to Calculate the Maximum Current Required for Output Buffers.

Since a floppy controller design may not work correctly due to current overloading, it is important to understand exactly how to calculate the maximum current required by the floppy controller for each output signal to the disk drive. This is largely determined by the termination resistors used by the disk drives. A formula that can be used is:

$$\frac{(V_{CC}) + (\text{Max } V_{CC} \text{ Variation}) - (V_{OL(max)})}{(\text{Termination Resistor}) \bullet \frac{1}{N} \bullet (1 - \text{Resistor Accuracy})}$$

$V_{CC} = 5.0$

Max V_{CC} Variation = Power supply variation (0.5V)

$V_{OL(Max)}$ = Maximum active low output voltage of buffer (0.8V)

Termination Resistor = Termination on disk drive

N = Number of terminated disk drives

Resistor Accuracy = Accuracy of termination resistors (10% or 0.10)

Example 1:

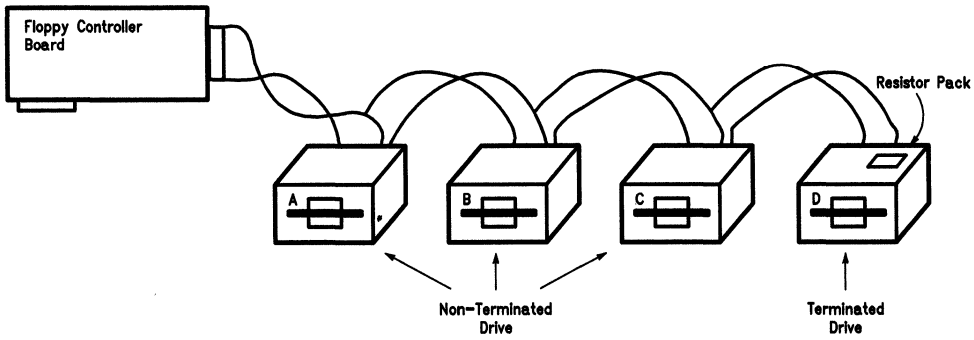
One terminated drive with 150 termination resistors.

$$\frac{5.0 + 0.5 - 0.8}{150 \bullet 1 \bullet 0.9} = 34.8 \text{ mA}$$

Example 2:

Four terminated drives with 1k termination resistors.

$$\frac{5.0 + 0.5 - 0.8}{1000 \bullet \frac{1}{4} \bullet 0.9} = 20.9 \text{ mA}$$



TL/F/10458–7

FIGURE 6. Daisy Chain of Four Drives

DRIVE TYPES

There are many types of disk drives that can be connected to the DP8473 floppy disk controller. The DP8473 is compatible with 8", 5.25", and 3.5" floppy disk drives, although 8" drives are rarely used today.

Other types of peripherals may be connected to the floppy controller as well. A streaming tape drive that is used to back up the hard disk is often connected to the floppy controller. A tape drive of this type is very specialized. It has been designed to look like a floppy disk drive from an electrical interface point of view. It does not perform exactly like a disk drive, however. Special software is usually required to make it work correctly. The STEP signal is often used to issue commands to the tape drive. For example, to rewind the tape, four step pulses may need to be issued. To start a read, six step pulses might be issued.

All of these different drive types have one thing in common, a similar electrical interface. This allows them all to be connected to a common drive interface cable. For example, the READ DATA signal on pin 30 of the floppy interface cable is the MFM encoded serial stream of data from the disk. The INDEX signal on pin 8 of the cable identifies the beginning of a track.

However, there are some minor differences between drive tapes that must be considered. The DENSITY signal is a good example of a difference. This signal is active for *high* density transfers on a dual density 3.5" drives. But, this signal is active for *low* density transfers on a dual density 5.25" drive. This difference makes the floppy system design more complex when 5.25" dual density and 3.5" dual density drives are both used in the same PC.

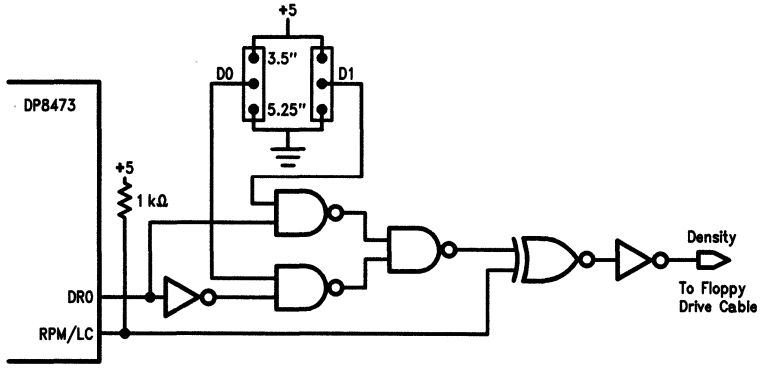
The DP8473 has a signal called RPM/LC that normally connects to the Density or Low Current input on a dual density 5.25" drive. If a 3.5" drive is used, the RPM/LC output should be inverted.

A design such as the one shown in *Figure 7* could be used to create the proper DENSITY signal for both 5.25" and 3.5" drives. The jumpers and logic allow the user to select between drive types for each individual drive.

Another solution is simply to use a 3.5" drive that contains a built-in jumper to vary the polarity of the DENSITY signal directly on the drive. This option eliminates the need for external logic and jumpers.

Another signal that is drive type dependent is DISK CHANGED. This signal exists on low and dual density 3.5" drives and also on dual density 5.25" drives. It does not exist on low density 5.25" drives. If a low density 5.25" drive is to be used, the DISK CHANGED signal should be held active (low level). It may be held active by the drive by itself. If not, a pull-down resistor could be used to activate the non-driven signal.

One thing to consider while choosing drive types is media compatibility. Of course, you can't put a 3.5" disk in a 5.25" drive. But, there are more subtle incompatibilities even within similar media types. For example, a low density 5.25" disk written to in low density mode by a dual density drive cannot be read reliably by a low density drive. Table I lists the compatabilities between different drives and media types.



TL/F/10458–8

FIGURE 7. Density Select Logic for 5.25" or 3.5" Drives

TABLE I. Drive and Media Compatibility

Drive Type	Mode	Media Type			
		3.5″	3.5″ HD	5.25″ DD	5.25″ HD
3.5″ LD	720k	R/W			
3.5″ LD,HD	720k 1.44M	R/W	R/W		
5.25″ LD	360k			R/W	
5.25″ LD,HD	360k 1.2M			R Only	R/W

Notes:

LD = Low Density
HD = High Density
R/W = Readable & Writable
R Only = Readable Only

DATA RATES

Different drive types may use different data rates. The data rate is specified by the number of bits that are transferred in a second. For example, 250 kb/s is translated to 250 thousand bits per second.

The data rate used by a disk drive is determined by the electronics of the drive and the specifications of the media. Low Density media require data to be transferred at 250 kb/s. High Density media is twice as fast at 500 kb/s.

There is a complication with Low Density 5.25″ media in a Dual Density drive. The 5.25″ Dual Density drive spins the disk faster (360 RPM) than a Low Density drive (300 RPM). When a Low Density disk is read from a Dual Density drive, the data rate will be 300 kb/s instead of 250 kb/s because of the rotational speed difference.

The DP8473 can operate at all the data rates required for a PC. This includes 250, 300, and 500 kb/s. In addition, the DP8473 can operate at 1 Mb/s. This high data rate is starting to appear in both floppy disks and streaming tape drives.

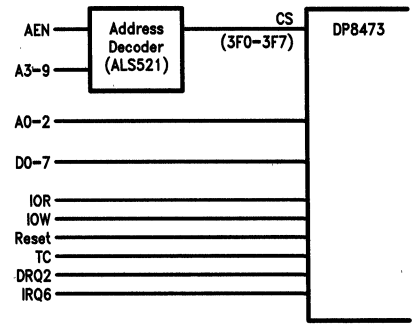
μP INTERFACE

It is hard to imagine how the interface between the DP8473 and the μP bus could be made any simpler than it is. Only one interface function is not integrated in the DP8473. That function is address decoding. The typical μP connections are shown in *Figure 8*.

The DP8473 requires a CS (chip select) enable signal that is generated elsewhere. Typically, this could be generated by an ALS521 8-bit comparator or a similar circuit. Address decoding of the three least significant bits of the address is performed by the DP8473. The AEN (address enable) signal from the μP bus should be included in the CS logic. This will prevent DMA transfers from generating a CS.

The eight bit data bus from the DP8473 connects directly to the data bus of the μP. Bus transceivers are not required because 12 mA buffers are built into the DP8473.

The IOR (I/O read), IOW (I/O write), RESET, TC (terminal count), DRQ (DMA request), and IRQ (interrupt request) signals can be connected directly to the DP8473. No additional buffering or gating is required. The logic required to TRI-STATE® the DRQ and INT pins is integrated in the DP8473.



TL/F/10458–9

FIGURE 8. μP Interface to DP8473

FILTER SELECTION

The internal data separator in the DP8473 requires an external filter to operate correctly. This filter is part of an analog Phase-Locked Loop (PLL) that is used by the data separator integrated into the DP8473. This is commonly referred to as an analog data separator due to the analog nature of the PLL's filter and Voltage Controlled Oscillator (V_{CO}).

As the floppy controller changes from one data rate to another, the filter used by the PLL must change also. This is done automatically in the DP8473. Two or three filters are connected externally to different pins of the DP8473. The correct filter is selected and activated by the DP8473 itself.

The filter selection is performed by grounding or forcing TRI-STATE the appropriate filters. For example, a two-filter arrangement is shown in *Figure 9*. If 500 kb/s is selected, the FGND500 pin is grounded and the FGND250 pin is at TRI-STATE. From the filter pin's point of view, this appears as filter F2 in parallel with capacitor C1. F1 is electrically out of the picture. When 250 kb/s is selected, it is the opposite. The FGND500 pin is at TRI-STATE and the FGND250 pin is grounded. Since 300 kb/s data rate is close to 250 kb/s, the same filter is used for both data rates.

The two-filter arrangement shown in *Figure 9* would be used in most PC applications. It supports 250, 300, and 500 kb/s data rate. Other filter combinations may be used for specialized applications. These filter combinations are described in the DP8473 data sheet.

DATA SEPARATOR PERFORMANCE

One of the most important features of the DP8473 is the high level of data separator performance. This performance translates directly to reduced disk I/O errors. There is quite a bit of information that can be discussed on data separator performance. A lot of this information is presented in an application note titled "Floppy Disk Data Separator Design Guide for the DP8473, AN-505".

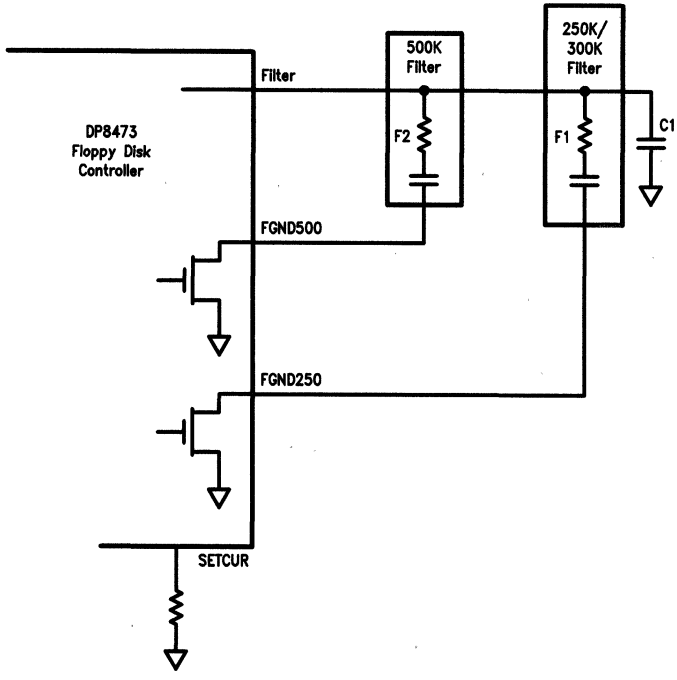
It might seem desirable to specify the maximum error rate of the DP8473. This could be expressed in terms of the number of bits that can be read correctly before an error occurs (10^{12}, for example). This is not a practical parameter to specify, however. One problem is the amount of time this type of measurement would take. A test of 10^{12} could take well over 400 days to measure.

Another problem is defining the test conditions. Is the test performed under ideal disk conditions? Are bits jittered or is motor speed variation simulated? There are so many variables in this type of a test that it would be difficult for two different people to produce the same results. In addition, the error rate is related to other factors beside the DP8473 such as the disk drive or the floppy media.

There are many different types of measurements that can be made with a data separator. Most of these measurements indicate how well only one particular section of the data separator performs. For example, the gain of the VCO (Voltage Controlled Oscillator) or the accuracy of the ¼ period delay line. They don't, however, give a good indication of how well the data separator will perform under real-life conditions.

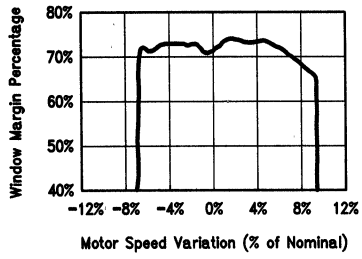
There is one measurement that produces meaningful data. This measurement is called "Dynamic Window Margin". Dynamic window margin attempts to indicate total data separator performance under real-life conditions. It measures how much bit jitter the data separator can handle while reading a worst case data pattern (DB6 hex) with a drive that has a motor speed of varying accuracy. The data is jittered in a manner similar to real world jitter with a reverse write precompensation pattern. The measurement is taken at the worst point over a motor speed variation of ±6%. An example of a dynamic window margin measurement is shown in *Figure 10.*

The typical dynamic window margin is specified in the DP8473 data sheet. National Semiconductor has made measurements of the dynamic window margin of many data separators, and the results show that no other data separator performs as well as the DP8473s.



TL/F/10458–10

FIGURE 9. Typical Filter Connection for 250, 300, and 500 kb/s



TL/F/10458–11

FIGURE 10. Typical Dynamic Window Margin for DP8473

LAYOUT AND COMPONENT CONSIDERATIONS

The DP8473 contains a combination of digital and analog circuitry. Because of the analog nature of the data separator, some precautions should be taken when designing a board with the DP8473.

A good DP8473 board layout design will distinguish between analog and digital, V_{CC} and Ground. The supply pins used by the digital circuitry are labeled V_{CC}, GNDB, GNDC, and GNDD. The supply pins used for the analog data separator are labeled V_{CC}A and GNDA. Standard digital decoupling techniques should be used with the digital supply pins. This typically involves 0.1 μF capacitors connected between V_{CC} and GND.

The analog supply pins require a bit more consideration than the digital supply pins. Any noise or ripple on the analog supplies will degrade the data separator performance. It is recommended to minimize this noise as much as possible. Less than 50 mV noise would be good.

There are many methods that can be used to minimize noise on the analog supply pins. One of the best methods is a 5.0V voltage regulator dedicated to the analog section. This guarantees a very clean signal. The voltage regulator can be driven by the 12V power supply.

Another method is to place the DP8473 on the board close to the entry point of the power supply. At the very least, separate supply lines should be dedicated from the power supply entry point to the DP8473 V_{CC}A and GNDA.

In addition to the analog supply, any noise or crosstalk introduced to the external filters will adversely effect the performance of the data separator. The data separator filters should be positioned as close as possible to the DP8473. A ground plane surrounding the filters would also be advisable. The resistor attached to the SETCUR pin should also be close to the DP8473.

If a 24 MHz crystal is used, it should be placed close to the DP8473 as well as the 10 pF capacitor attached to both sides of the crystal.

The component types and tolerances may also effect the data separator performance. The accuracy of the resistor attached to the SETCUR pin is important. It should have a 1% tolerance rating. A 5% could be used, but the accuracy may effect the data separator performance.

The capacitor in series with the resistor attached to the FILTER pin is also critical. It should have a 5% tolerance. The series resistor in the same network is not as critical as the SETCUR resistor. Therefore, a normal 5% tolerance can be used here.

Finally, the capacitor in parallel with the filter network can be rated as low as 10%–20%. It does not effect the data separator very much.

The component tolerances mentioned here are only recommendations. The DP8473 will work properly with a wide range of filter values. These recommendations should be followed if data separator performance is an important issue in a particular design (which is normally true).

TROUBLE-SHOOTING

If the floppy controller does not appear to operate correctly, there are some key areas that can be looked at for the source of the problem.

Drive's in Use light remains on at all times.

- Drive Interface cable plugged in backward.
- Drive Interface signals not properly routed.

DOS returns a "Not ready error reading drive X".

This error can be caused by many different problems. At this point it would be best to run the "Floppy Demo Program" as described later in this section.

DOS Directory command returns an old directory.

- Disk Changed signal improperly routed.

POST produces a "601" error while booting.

- Drive Interface cable unplugged.
- Hardware problem with μP interface.

PC locks up while booting.

- Hardware problem with μP interface.

Parity Error.

- DMA interface problem.

Many advanced diagnostics may be used while running the "Floppy Demo Program" which is available from National Semiconductor. This program allows you to issue individual floppy commands such as Read Data, Format Track, and Seek. The Result Phase of these commands can be analyzed to help determine where a problem exists. The following list describes some likely sources of problems based on what is observed with the "Floppy Demo Program".

"Missing Address Mark in Address Field" error.

This indicates that the floppy controller could not find any valid data on the track being read. No sectors could be found.

- Track has not been formatted. Could be a blank disk. The controller does recognize Index Pulses, however. This indicates that the drive cable interface is at least partially intact.
- Read Data drive interface signal not properly routed.
- General data separator problem. See the data separator discussion later.

"Did not receive interrupt" error.

This is usually accompanied with all FF's in the Result Phase of the command. This indicates that the controller could not read anything from the disk drive and Index pulses were not seen. Be sure to reset the floppy controller after this error.

- Disk not inserted in drive or drive door open.
- Wrong drive selected in command.
- Drive Interface cable unplugged.

MSR (Main Status Register) does not return to 80 (hex) after a reset.

- Floppy controller not inserted in socket properly.
- Hardware problem with address decode (CS) or μP interface signals.
- Crystal or external clock not operating properly.
- Floppy controller is bad.

"No Data" error or "CRC error".

- Bad disk media.
- General data separator problem.

Fewer bytes read than requested or error in Result Phase.

- Noise on Reset pin. Insert capacitor between Reset and digital GND as specified in the data sheet.

Long term read produces some errors.

- General data separator problem.

Many of the problems described above refer to a general data separator problem. There are many things that can be looked at for data separator problems.

- Filter wired incorrectly.
- Incorrect filter component values.
- Filter layed-out poorly.
- Too much noise on analog $V_{CC}A$ or GNDA.

FLOPPY CONTROLLER DESIGN WITH NEC765A

In order to appreciate the amount of circuitry integrated into the DP8473, it may be useful to analyze a typical floppy controller design for the PC-AT using the standard NEC765A floppy disk controller. To simplify the analysis, only the block diagram will be presented. The actual schematic would require many pages. The block diagram is shown in *Figure 11*.

The NEC765A was developed many years ago originally for 8" floppy disk drives. It became very popular because it was designed into the original IBM PC. The NEC765A performs many functions, but there are also many functions that it does not perform.

A data separator isolates the individual pulses read from the disk drive and allows the floppy controller to distinguish between MFM clock pulses and data pulses. It typically incorporates an analog PLL. An analog data separator design could require as many as 10 chips to design. A single chip discrete digital data separator could also be used, although the performance will not be as good.

The write precompensation circuit shifts the MFM encoded data as it is being written to the disk. This shifting compensates for known bit shifts that will occur due to the magnetic influences of the individual bits recorded on the disk. This is typically designed with a shift register and a multiplexer.

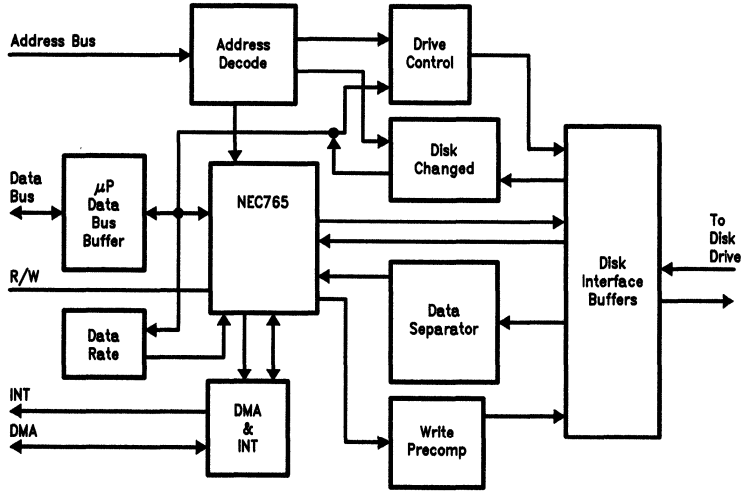
The NEC765 cannot interface directly to the Drive Interface cable. Separate 48 mA buffers are required for each output signal. This requires three 7406's. Also, the inputs from the disk drive required Schmitt inverters.

Additional buffering is also required for the μP data bus.

The PC-AT requires that the Disk Changed signal be read from a particular port. This involves address decoding and buffering.

The only method available to vary the data rate used by the NEC765A is by altering the input clock frequency to the chip. This must be done with a complex divider circuit that generates the different frequencies required for 250, 300, and 500 kb/s data rates.

Due to timing incompatabilities in the NEC765A, the Drive Select and Motor On signals must be generated by an external port that is controlled by software. This port also controls the software reset and DMA and INT enable circuitry.



TL/F/10458–12

FIGURE 11. Block Diagram of Floppy Controller Design with NEC765

DMA transfers must be slowed down due to handshaking problems with the NEC765A. This delay is performed with an external shift register.

This entire design easily requires at least a couple of dozen chips. The amount of board space used is quite large. A considerable amount of current is also consumed. This compares to the DP8473 solution which only requires two chips. It is easy to see that the DP8473 solution is much more economical.

CONCLUSION

This design guide was created to answer the most common questions encountered while designing with the DP8473. Any new design can be based on the information given in this guide. A two drive system can be created or more drives can be added if required. A variety of disk drives may be used including 3.5" drives.

The address decoding is the only function not integrated into the DP8473. However, the integrated data separator requires external filtering which should be carefully layed-out on the board.

If problems arise, there are many items that can be looked at to help identify where the problem exists. It may be useful to obtain a floppy controller diagnostic program similar to the "Floppy Demo Program" available from a National Semiconductor sales office.

If more information is desired concerning the performance of the data separator or the trade-offs of designing a custom filter for the PLL, please read the application note titled "Floppy Disk Data Separator Design Guide for the DP8473", AN-505.

National Semiconductor

DS8922/22A/DS8923/23A TRI-STATE® RS-422 Dual Differential Line Driver and Receiver Pairs

General Description

The DS8922/22A and DS8923/23A are Dual Differential Line Driver and Receiver pairs. These devices are designed specifically for applications meeting the ST506, ST412 and ESDI Disk Drive Standards. In addition, the devices meet the requirements of the EIA Standard RS-422.

These devices offer an input sensitivity of 200 mV over a ±7V common mode operating range. Hysteresis is incorporated (typically 70 mV) to improve noise margin for slowly changing input waveforms. An input fail-safe circuit is provided such that if the receiver inputs are open the output assumes the logical one state.

The DS8922A and DS8923A drivers are designed to provide unipolar differential drive to twisted pair or parallel wire transmission lines. Complementary outputs are logically ANDed and provide an output skew of 0.5 ns (typ.) with propagation delays of 12 ns.

Both devices feature TRI-STATE outputs. The DS8922/22A have independent control functions common to a driver and receiver pair. The DS8923/23A have separate driver and receiver control functions.

Power up/down circuitry is featured which will TRI-STATE the outputs and prevent erroneous glitches on the transmission lines during system power up or power down operation.

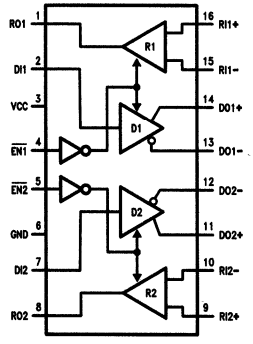
The DS8922/22A and DS8923/23A are designed to be compatible with TTL and CMOS.

Features

- 12 ns typical propagation delay
- Output skew—±0.5 ns typical
- Meets the requirements of EIA Standard RS-422
- Complementary Driver Outputs
- High differential or common-mode input voltage ranges of ±7V
- ±0.2V receiver sensitivity over the input voltage range
- Receiver input fail-safe circuitry
- Receiver input hysteresis—±70 mV typical
- Glitch free power up/down
- TRI-STATE outputs

Connection Diagrams

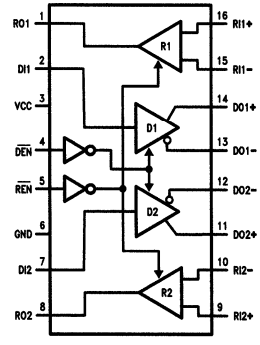
DS8922A Dual-In-Line



TL/F/8511–1

Order Number DS8922N, J, M, DS8922AN, AJ, AM
See NS Package Number N16A, J16A or M16A

DS8923A Dual-In-Line



TL/F/8511–2

Order Number DS8923N, J, M DS8923AN, AJ, AM
See NS Package Number N16A, J16A or M16A

Truth Tables

DS8922/22A

$\overline{EN1}$	$\overline{EN2}$	RO1	RO2	DO1	DO2
0	0	ACTIVE	ACTIVE	ACTIVE	ACTIVE
1	0	HI-Z	ACTIVE	HI-Z	ACTIVE
0	1	ACTIVE	HI-Z	ACTIVE	HI-Z
1	1	HI-Z	HI-Z	HI-Z	HI-Z

DS8923/23A

$\overline{DEN}$	$\overline{REN}$	RO1	RO2	DO1	DO2
0	0	ACTIVE	ACTIVE	ACTIVE	ACTIVE
1	0	ACTIVE	ACTIVE	HI-Z	HI-Z
0	1	HI-Z	HI-Z	ACTIVE	ACTIVE
1	1	HI-Z	HI-Z	HI-Z	HI-Z

For complete specifications see the Interface Databook.

National Semiconductor

DS8921/DS8921A Differential Line Driver and Receiver Pair

General Description

The DS8921, DS8921A are Differential Line Driver and Receiver pairs designed specifically for applications meeting the ST506, ST412 and ESDI Disk Drive Standards. In addition, these devices meet the requirements of the EIA Standard RS-422.

The DS8921A receiver offers an input sensitivity of 200 mV over a ±7V common mode operating range. Hysteresis is incorporated (typically 70 mV) to improve noise margin for slowly changing input waveforms. An input fail-safe circuit is provided such that if the receiver inputs are open the output assumes the logical one state.

The DS8921A driver is designed to provide unipolar differential drive to twisted pair or parallel wire transmission lines. Complementary outputs are logically ANDed and provide an output skew of 0.5 ns (typ.) with propagation delays of 12 ns.

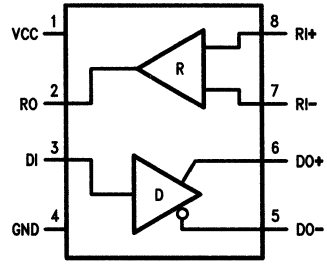
Power up/down circuitry is featured which will TRI-STATE® the outputs and prevent erroneous glitches on the transmission lines during system power up or power down operation.

The DS8921A is designed to be compatible with TTL and CMOS.

Features

- 12 ns typical propagation delay
- Output skew - 0.5 ns typical
- Meet the requirements of EIA Standard RS-422
- Complementary Driver Outputs
- High differential or common-mode input voltage ranges of ±7V
- ±0.2V receiver sensitivity over the input voltage range
- Receiver input fail-safe circuitry
- Receiver input hysteresis-70 mV typical
- Glitch free power up/down

Connection Diagram



TL/F/8512–1

Order Number DS8921M, DS8921N, DS8921AM, DS8921AN, DS8921J or DS8921AJ
See NS Package Number J08A, M08A or N08E

Truth Table

Receiver		Driver		
Input	V_{OUT}	Input	V_{OUT}	$\overline{V_{OUT}}$
$V_{ID} \geq V_{TH}$ (MAX)	1	1	1	0
$V_{ID} \leq V_{TH}$ (MIN)	0	0	0	1

For complete specifications see the Interface Databook.

Section 6 Contents

Section 6

Disk Drive Interface Circuits

National Semiconductor

DS26C31C CMOS Quad TRI-STATE® Differential Line Driver

General Description

The DS26C31 is a quad differential line driver designed for digital data transmission over balanced lines. The DS26C31 meets all the requirements of EIA standard RS-422 while retaining the low power characteristics of CMOS. This enables the construction of serial and terminal interfaces while maintaining minimal power consumption.

The DS26C31 accepts TTL or CMOS input levels and translates these to RS-422 output levels. This part uses special output circuitry that enables the individual drivers to power down without loading down the bus. The DS26C31 also includes special power up and down circuitry which will TRI-STATE the outputs during power up or down, preventing spurious glitches on its outputs. This device has enable and disable circuitry common to all four drivers. The DS26C31 is pin compatible to the AM26LS31 and the DS26LS31.

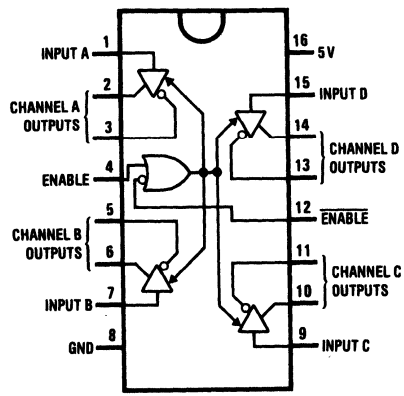
All inputs are protected against damage due to electrostatic discharge by diodes to V_{CC} and ground.

Features

- TTL input compatible
- Typical propagation delays: 6 ns
- Typical output skew: 0.5 ns
- Outputs won't load line when V_{CC} = 0V
- Meets the requirements of EIA standard RS-422
- Operation from single 5V supply
- TRI-STATE outputs for connection to system buses
- Low quiescent current
- Available in surface mount

Connection Diagram

Dual-In-Line Package



TL/F/8574–1

Top View

Order Number DS26C31CJ, DS26C31CM or DS26C31CN
See NS Package Number J16A, M16A or N16A

Truth Table

Active High Enable	Active Low Enable	Input	Non-Inverting Output	Inverting Output
L	H	X	Z	Z
All other combinations of enable inputs		L	L	H
		H	H	L

L = Low logic state
H = High logic state
X = Irrelevant
Z = TRI-STATE (high impedance)

6

National Semiconductor

DS26C32AC Quad Differential Line Receiver

General Description

The DS26C32A is a quad differential line receiver designed to meet the RS-422, RS-423, and Federal Standards 1020 and 1030 for balanced and unbalanced digital data transmission, while retaining the low power characteristics of CMOS.

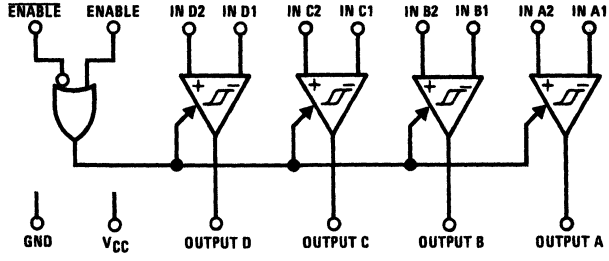
The DS26C32A has an input sensitivity of 200 mV over the common mode input voltage range of ±7V. Each receiver is also equipped with input fail-safe circuitry, which causes the output to go to a logic "1" state when the inputs are open.

The DS26C32A provides an enable and disable function common to all four receivers, and features TRI-STATE® outputs with 6 mA source and sink capability. This product is pin compatible with the DS26LS32A and the AM26LS32.

Features

- Low power CMOS design
- ±0.2V sensitivity over the entire common mode range
- Typical propagation delays: 19 ns
- Typical input hysteresis: 60 mV
- Input fail-safe circuitry
- Inputs won't load line when V_{CC} = 0V
- Meets the requirements of EIA standard RS-422
- TRI-STATE outputs for connection to system buses
- Available in Surface Mount

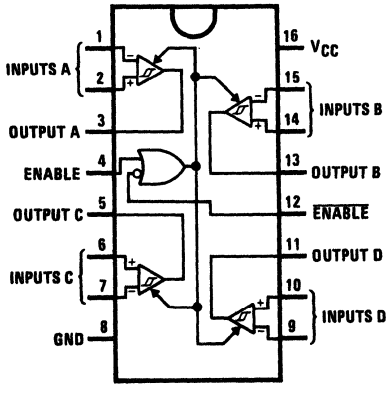
Logic Diagram



TL/F/8764–1

Connection Diagram

Dual-In-Line Package



TL/F/8764–2

Top View

Order Number DS26C32ACJ, DS26C32ACM or DS26C32ACN
See NS Package J16A, M16A or N16A

Truth Table

ENABLE	$\overline{\text{ENABLE}}$	Input	Output
0	1	X	Hi-Z
See Note Below		$V_{ID} \geq V_{TH}$ (Max)	1
		$V_{ID} \leq V_{TH}$ (Min)	0
		Open	1

Hi-Z = TRI-STATE

Note: Input conditions may be any combination not defined for ENABLE and $\overline{\text{ENABLE}}$.

National Semiconductor

DS34C86
Quad CMOS Differential Line Receiver

General Description

The DS34C86 is a quad differential line receiver designed to meet the RS-422, RS-423, and Federal Standards 1020 and 1030 for balanced and unbalanced digital data transmission, while retaining the low power characteristics of CMOS.

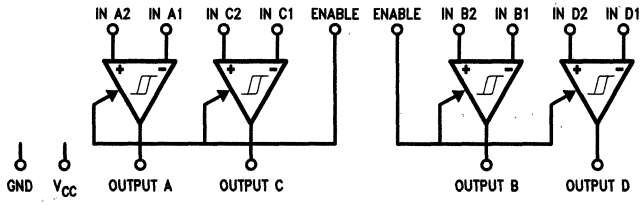
The DS34C86 has an input sensitivity of 200 mV over the common mode input voltage range of ±7V. Hysteresis is provided to improve noise margin and discourage output instability for slowly changing input waveforms.

Separate enable pins allow independent control of receiver pairs. The TRI-STATE® outputs have 6 mA source and sink capability. The DS34C86 is pin compatible with the DS3486.

Features

- Low power CMOS design
- ±0.2V sensitivity over the entire common mode range
- Typical propagation delays: 19 ns
- Typical input hysteresis: 60 mV
- Inputs won't load line when V_{CC} = 0V
- Meets the requirements of EIA standard RS-422
- TRI-STATE outputs for connection to system buses
- Available in surface mount

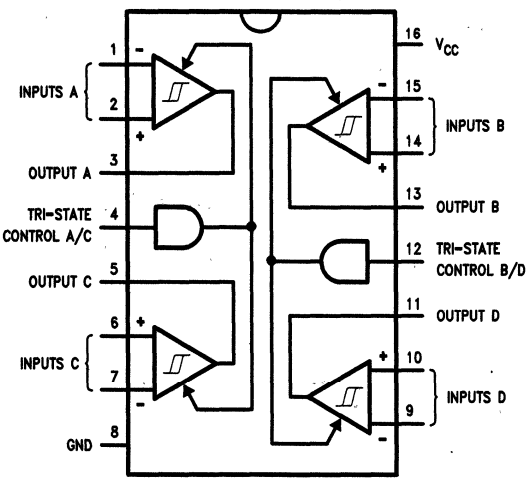
Logic Diagram



TL/F/8699–1

Connection Diagram

Dual-In-Line Package



TL/F/8699–2

Top View

Order Number DS34C86J, DS34C86M, and DS34C86N
See NS Package Number J16A, M16A and N16A

National Semiconductor

DS34C87 CMOS Quad TRI-STATE® Differential Line Driver

General Description

The DS34C87 is a quad differential line driver designed for digital data transmission over balanced lines. The DS34C87 meets all the requirements of EIA standard RS-422 while retaining the low power characteristics of CMOS. This enables the construction of serial and terminal interfaces while maintaining minimal power consumption.

The DS34C87 accepts TTL or CMOS input levels and translates these to RS-422 output levels. This part uses special output circuitry that enables the individual drivers to power down without loading down the bus. The DS34C87 also includes special power up and down circuitry which will TRI-STATE the outputs during power up or down, preventing spurious glitches on its outputs. This device has separate enable circuitry for each pair of the four drivers. The DS34C87 is pin compatible to the DS3487.

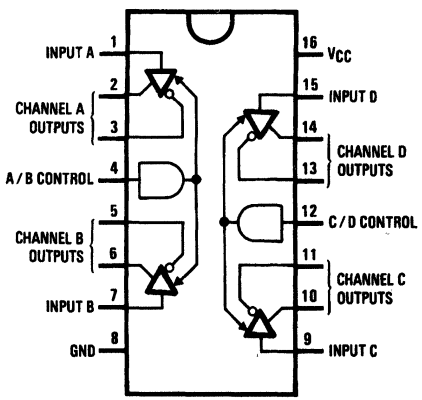
All inputs are protected against damage due to electrostatic discharge by diodes to V_{CC} and ground.

Features

- TTL input compatible
- Typical propagation delays: 6 ns
- Typical output skew: 0.5 ns
- Outputs won't load line when V_{CC} = 0V
- Meets the requirements of EIA standard RS-422
- Operation from single 5V supply
- TRI-STATE outputs for connection to system buses
- Low quiescent current
- Available in surface mount

Connection and Logic Diagrams

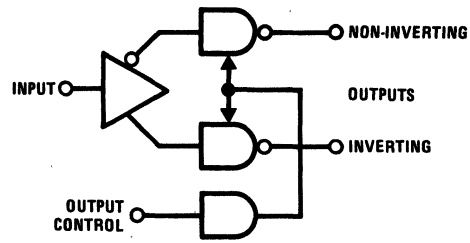
Dual-In-Line Package



TL/F/8576–1

Top View

Order Number DS34C87J, DS34C87M or DS34C87N
See NS Package Number J16A, M16A or N16A



TL/F/8576–2

Truth Table

Input	Control Input	Non-Inverting Output	Inverting Output
H	H	H	L
L	H	L	H
X	L	Z	Z

L = Low logic state
H = High logic state
X = Irrelevant
Z = TRI-STATE (high impedance)

National Semiconductor

DS3695A/DS3695AT
Multipoint RS485/RS422 Transceiver

General Description

The DS3695A is a high speed differential TRI-STATE® bus/line transceiver designed to meet the requirements of EIA standard RS485 with extended common mode range (+12V to −7V), for multipoint data transmission. In addition, it is compatible with the requirements of RS422.

The driver and receiver outputs feature TRI-STATE capability. The driver outputs remain in TRI-STATE over the entire common mode range of +12V to −7V. Bus faults that cause excessive power dissipation within the device trigger a thermal shutdown circuit, which forces the driver outputs into the high impedance state.

The receiver incorporates a fail safe feature which guarantees a high output state when the inputs are left open.

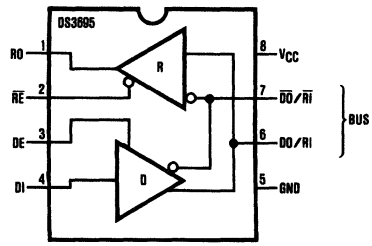
Both AC and DC specifications are guaranteed over the 0 to 70°C temperature and 4.75V to 5.25V supply voltage range.

Features

- Meets EIA standard RS485 for multipoint bus transmission and is compatible with RS-422
- 15 ns driver propagation delays with 2 ns skew (typical)
- Single +5V supply
- −7V to +12V bus common mode range permits ±7V ground difference between devices on the bus
- Thermal shutdown protection
- Power-up/down glitch-free driver outputs permit live insertion or removal of transceivers
- High impedance to bus with driver in TRI-STATE or with power off, over the entire common mode range allows the unused devices on the bus to be powered down
- Combined impedance of a driver output and receiver input is less than one RS485 unit load, allowing up to 32 transceivers on the bus
- 70 mV typical receiver hysteresis

Connection and Logic Diagram

Molded Package, Small Outline (M)



TL/F/5272–1

Top View

Order Number DS3695AM DS3695ATM
See NS Package Number M08A

Section 7

Rigid Disk Preamplifiers and Servo Control Circuits

Section 7 Contents

National Semiconductor

DP117-X/DP117-XR/μA117-X/μA117-XR Series Winchester Disk Read/Write Preamplifiers

General Description

The DP117-X/DP117-XR, μA117-X/μA117-XR Series High Performance Read/Write Preamplifiers are intended for use in Winchester disk drives which employ center tapped ferrite or manganese-zinc read/write heads. The circuit can interface with up to eight read/write heads which makes it ideal for multi-platter disk drive designs. Designed to reside in the Head/Disk Assembly (HDA) of Winchester disk drives, the Read/Write Preamplifiers provide termination, gain, and output buffering for the disk heads as well as switched write current. Certain write fault conditions are detected and reported to protect recording integrity. The parts are available with internal damping resistor (DP117-R) and without internal damping resistor (DP117).

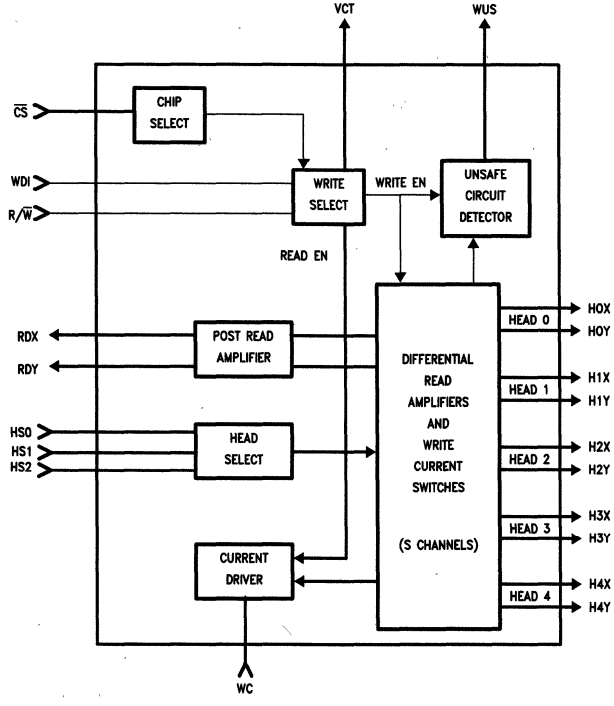
Features

- Wide bandwidth, high gain, low noise
- Up to eight read/write channels
- Internal write fault condition detection
- 5.0V and 12V power supply voltages
- Independent read and write data lines
- TTL control and data logic levels
- Externally programmable write current
- Available with internal damping resistor
- Compatible with SSI 117 family

Part Selection

Device Code	Channels
μA117-2	2
μA117-4	4
μA117-6	6

Block Diagram (Typical, DP117-X)



TL/F/9406–7

Absolute Maximum Ratings All voltages referenced to GND

If Military/Aerospace specified devices are required, please contact the National Semiconductor Sales Office/Distributors for availability and specifications.

Storage Temperature Range	
Ceramic	−65°C to +175°C
Plastic	−65°C to +150°C
Operating Junction Temperature Range	+25°C to +135°C
Lead Temperature	
Ceramic (Soldering, 60 seconds)	300°C
Plastic (Soldering, 10 seconds)	265°C
Internal Power Dissipation (Notes 1 & 2)	
28L-Ceramic DIP	2.50W
24L-Ceramic DIP	1.95W
18L-Ceramic DIP	1.58W
24L-Brazed Flatpak	0.97W
24L-Ceramic Flatpak	0.90W
28L-PLCC	1.39W
Supply Voltage (V_{CC1})	6.0V
Supply Voltage (V_{CC2})	15V
Write Current (IWC)	70 mA
Input Voltage Range	
Head Select (HS0, HS1, HS2)	−0.4V to V_{CC1} + 0.3V
Write Current (WC) Voltage in read and idle modes. (Write mode must be current limited to −70 mA)	−0.3V to V_{CC1} + 0.3V
Chip Select (CS)	−0.4V to V_{CC1} + 0.3V
Read/Write (R/W)	−0.4V to V_{CC1} + 0.3V
DC Supply Voltage	
(V_{DD1})	−0.3V to +14V
(V_{DD2})	−0.3V to +14V
(V_{CC})	−0.3V to +6.0V
Digital Input Voltage Range	
(V_{IN})	−0.3V to V_{CC} + 0.3V
Head Port Voltage Range	
(V_H)	−0.3V to V_{DD} + 0.3V
WUS Port Voltage Range	
(V_{WUS})	−0.3V to +14V
Write Current (I_W)	60 mA
Output Current (I_O)	
RDX, RDY	−10 mA
VCT	−60 mA
WUS	+12 mA

Recommended Operating Conditions

DC Supply Voltage	
(V_{DD1})	12V ±10%
(V_{DD2})	6.5V to V_{DD1}
(V_{CC})	5.0V ±10%
Head Inductance (Lh)	5.0 μH to 15 μH
Damping Resistor (External) (RD)	500Ω to 2000Ω
RCT Resistor (RCT)	90Ω ±5.05 (½W)
Write Current (I_W)	25 mA to 50 mA
RDX, RDY Output Current (I_O)	0 μA to 100 μA

Note 1: $T_{J\,MAX}$ = 150°C for the Plastic, and 175°C for the Ceramic.

Note 2: Ratings apply to ambient temperature at 25°C. Above this temperature, derate the 28L-Ceramic DIP at 16.7 mW/°C, the 24L-Ceramic DIP at 13 mW/°C, the 18L-Ceramic DIP at 10.5 mW/°C, the 24L-Brazed Flatpak at 6.5 mW/°C, the 24L-Ceramic Flatpak at 6.0 mW/°C, and the 28L-PLCC at 11.2 mW/°C.

DC Characteristics 25°C ≤ T_J < 125°C, V_{DD1} = 12V, V_{CC} = 5.0V, unless otherwise specified

Symbol	Parameter		Conditions		Min	Max	Units
I_{CC}	Supply Current		Read/Idle Mode			25	mA
			Write Mode			30	
I_{DD}	Supply Current		Idle Mode			25	mA
			Read Mode			50	
			Write Mode			30 + I_W	
P_C	Power Consumption		T_J = 125°C	Idle Mode		400	mW
				Read Mode		600	
				Write Mode, I_W = 50 mA RCT = 90Ω RCT = 0Ω		850 1050	
V_{IL}	Digital Inputs	Input Voltage LOW			−0.3	0.8	V
V_{IH}		Input Voltage HIGH			2.0	V_{CC} + 0.3	V
I_{IL}		Input Current LOW	V_{IL} = 0.8V		−0.4		mA
I_{IH}		Input Current HIGH	V_{IH} = 2.0V			100	μA
V_{OL}	WUS Output		I_{OL} = 8.0 mA			0.5	V
I_{OH}			V_{OH} = 5.0V			100	μA
V_{CT}	Center Tap Voltage		Read Mode		4.0 (typ)		V
			Write Mode		6.0 (typ)		V

Write Characteristics

V_{DD1} = 12V, V_{CC} = 5.0V, I_W = 45 mA, Lh = 10 μH, f (Data) = 5.0 MHz, CL (RDX, RDY) ≤ 20 pF, $R_{D\ EXT}$ = 750Ω or $R_{D\ INT}$, unless otherwise specified

Parameter	Conditions	Min	Max	Units
Write Current Range		10	50	mA
Write Current Constant "K"		133	147	V
Differential Head Voltage Swing		5.7		V (pk)
Unselected Differential Head Current			2.0	mA (pk)
Differential Output Capacitance			15	pF
Differential Output Resistance	Without Internal Resistors	10k		Ω
	With Internal Resistors	538	1.0k	
WDI Transition Frequency	WUS = LOW	400 (typ)		kHz
I_{WC} to Head Current Gain		18 (typ)		mA/mA

Read Characteristics

V_{DD1} = 12V, V_{CC} = 5.0V, Lh = 10 μH, f (Data) = 5.0 MHz, CL (RDX, RDY) ≤ 20 pF, (V_{in} is referenced to V_{CT}), $R_{D\ EXT}$ = 750Ω or $R_{D\ INT}$, unless otherwise specified

Parameter	Conditions		Min	Max	Unit
Differential Voltage Gain	V_{IN} = 1.0 mV_{p-p} at 300 kHz RL (RDX), RL (RDY) = 1.0 kΩ		80	120	V/V
Dynamic Range	Input Voltage, V_I, where gain falls by 10%. V_{IN} = V_I + 0.5 mV_{p-p} at 300 kHz		−2.0	2.0	mV
Bandwidth (−3 dB)	I Zs I < 5.0Ω, V_{IN} = 1.0 mV_{p-p}		30		MHz
Input Noise Voltage	BW = 15 MHz, Lh = 0, Rh = 0			2.1	$nV/\sqrt{Hz}$
Differential Input Capacitance	f = 5.0 MHz			23	pF
Differential Input Resistance	f = 5.0 MHz	Without Internal Resistors	2k		Ω
		With Internal Resistors	440	850	
Input Bias Current				45	μA
Common Mode Rejection Ratio	V_{CM} = V_{CT} + 100 mV_{p-p} at 5.0 MHz		50		dB
Power Supply Rejection Ratio	100 mV_{p-p} at 5.0 MHz on V_{DD1}, V_{DD2} or V_{CC}		45		dB
Channel Separation	Unselected Channels: V_{IN} = 100 mV_{p-p} at 5.0 MHz and Selected Channel: V_{IN} = 0 mV_{p-p}		45		dB
Output Offset Voltage			−480	480	mV
Common Mode Output Voltage			5.0	7.0	V
Single Ended Output Resistance	f = 5.0 MHz			35	Ω
Internal Damping Resistor			560	1070	Ω

7

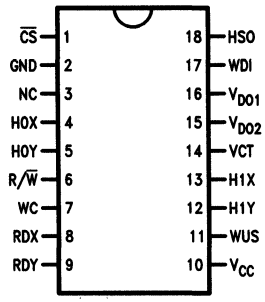
Switching Characteristics

V_{DD1} = 12V, V_{CC} = 5.0V, T_J = 25°C, I_W = 45 mA, Lh = 10 μH, f (Data) = 5.0 MHz, $R_{D\ EXT}$ = 750Ω or $R_{D\ INT}$, unless otherwise specified

Symbol	Parameter	Conditions	Min	Max	Units
R/$\overline{W}$	R/$\overline{W}$ to Write	Delay to 90% of Write Current		1.0	μs
	R/$\overline{W}$ to Read	Delay to 90% of 100 mV, 10 MHz Read Signal Envelope or to 90% Decay of Write Current		1.0	
$\overline{CS}$	$\overline{CS}$ to Select	Delay to 90% of Write Current or to 90% of 100 mV, 10 MHz Read Signal Envelope		1.0	μs
	$\overline{CS}$ to Unselect	Delay to 90% Decay of Write Current		1.0	
HS0 HS1 HS2	to Any Head	Delay to 90% of 100 mV, 10 MHz Read Signal Envelope		1.0	μs
WUS	Safe to Unsafe—TD1	I_W = 50 mA	1.6	8.0	μs
	Unsafe to Safe—TD2	I_W = 20 mA		1.0	
Head Current	Propagation Delay—TD3, TD4	Lh = 0 μH, Rh = 0Ω from 50% Points		25	ns
	Asymmetry	WDI has 50% Duty Cycle and 1 ns Rise/Fall Time		2	
	Rise/Fall Time	10%–90% Points		20	

Connection Diagrams

18-Lead Molded DIP

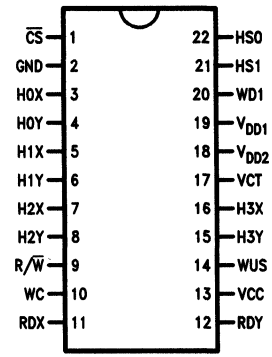


TL/F/9406–1

Top View

†Order Number μA1172DC or μA1172RDC
††See NS Package Number N18A

22-Lead Molded DIP



TL/F/9406–2

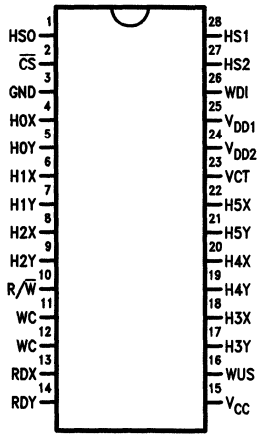
Top View

†Order Number μA1174PC or μA1174RPC
††See NS Package Number N22A

†For most current order information, contact your local sales office.
††For most current package information, contact product marketing.

Connection Diagrams (Continued)

28-Lead DIP

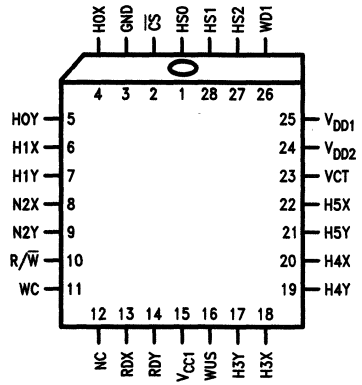


TL/F/9406–6

Top View

†Order Number μA1176PC or μA1176RPC
††See NS Package Number N28B

28-Lead PLCC



TL/F/9406–5

Top View

†Order Number μA1176QC or μA1176RQC
††See NS Package Number V28A

7

†For most current order information, contact your local sales office.
††For most current package information, contact product marketing.

Functional Description

In the Write mode, the DP117-X/DP117-XR, μA117-X/μA117-XR Series accepts TTL compatible write data pulses on the WDI lead. On the falling edge of each write data pulse, a current transition is made in the selected head. Head selection is accomplished via TTL input signals: HS0, HS1, HS2 (see Table II). Internal circuitry senses the following conditions:

1. Absence of data transitions.
2. Open circuit head connection.
3. Absence of write current.
4. Short circuit head connection.
5. Idle or read mode.

Any or all of the above conditions would result in a high level on the write unsafe (WUS) output signal.

During read operations, the DP117-X amplifies the differential voltages appearing across the selected R/W head lead and applies the amplified signal differentially to data lines RDX and RDY.

Pin Descriptions

Lead	Name	Function
$\overline{CS}$	Chip Select	Chip Select High disables the read/write function of the device and forces idle mode. (TTL)
R/$\overline{W}$	Read/Write Select	A Logic High places the devices in read mode and a Logic Low forces write mode. Refer to Table I. (TTL)
H0X, Y through H5X, Y	Read/Write Head Connections	The DP117 has five pairs of read/write connections. The X and Y phases are made consistent with the read output, RDX and RDY, phases. (Differential)
RDX, Y	Read Data Outputs	The chip has one pair of read data outputs which is multiplexed to the appropriate head connections. (Differential)
HS0 through HS2	Head Select Inputs	The eight read/write heads are addressed with the head select inputs. Refer to Table II. (TTL)
WC	Write Current Input	This lead sets the current level for the write mode. An external resistor is connected from this lead to ground, and write current is determined by the value of this resistor divided into the write current constant K, which is typically 140V.
WDI	Write Data Input	The write data input toggles the write current between the X and Y selected head connections. Write current is switched on the negative edge of WDI. The initial direction for write current is the X side of the switch and is set upon entering read or idle mode. (TTL)
V_{DD2}	Resistor Center Tap	In some versions (determined by lead availability) of the DP117-X series, a resistor may be connected between RCT and V_{DD1} to reduce internal power dissipation. If this resistor is not used, RCT must be connected externally to V_{DD1}.
VCT	Center Tap Voltage	The center tap output provides bias voltage for the head inputs in read and write mode. It should be connected to the center tap of the read/write heads.
WUS	Write Unsafe	A high logic level at the write unsafe output indicates a fault condition during write. Write unsafe will also be high during read and idle mode. (Open collector)

TABLE I. Read/Write Select

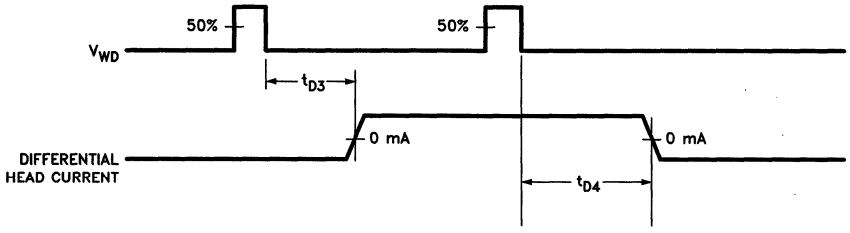
Operating Modes		
Chip Select $\overline{CS}$	**Read/Write R/$\overline{W}$**	**Mode**
1	X	Idle
0	1	Read
0	0	Write

TABLE II. Head Select Inputs

Head Selection			
HS0	**HS1**	**HS2**	**Head Selected (Note 1)**
0	0	0	0
1	0	0	1
0	1	0	2
1	1	0	3
0	0	1	4
1	0	1	5

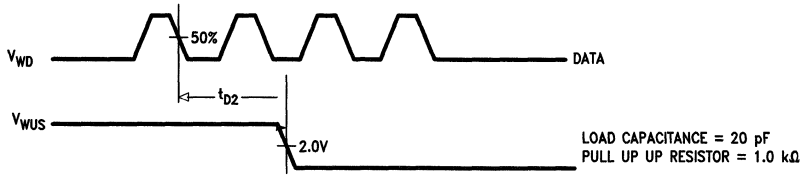
Note 1: If selected head is beyond the capacity of the DP117-X model, the open input condition on the selected input will be reported as an unsafe level at the WUS output.

Timing Diagrams



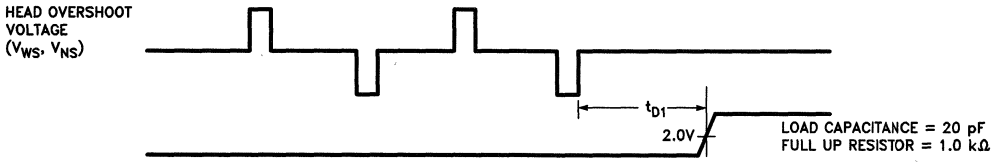
TL/F/9406–8

FIGURE 1. Head Current Timing



TL/F/9406–9

FIGURE 2a. Unsafe to Safe Timing



TL/F/9406–10

FIGURE 2b. Safe to Unsafe Timing

National Semiconductor

DP501X/DP501XR/μA501X/μA501XR Series 6 or 8 Channel Read/Write Circuit

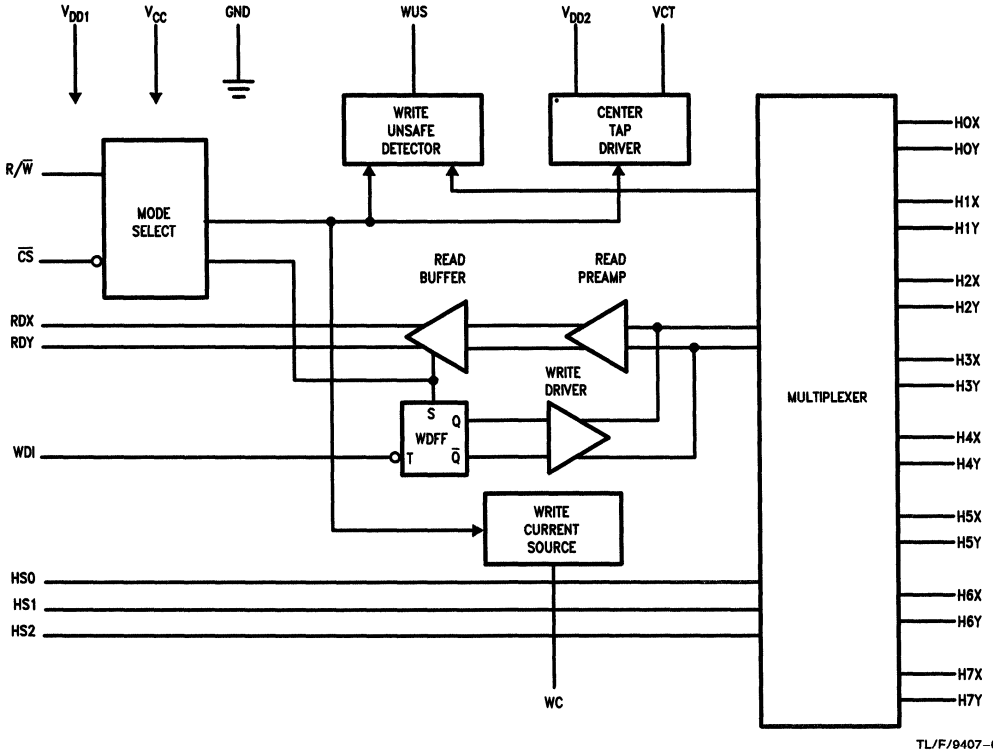
General Description

The μA501X/μA501XR devices are bipolar monolithic integrated circuits designed for use with center-tapped ferrite recording heads. They provide a low noise read path, write current control, and data protection circuitry for eight channels. The μA501X/μA501XR requires +5.0V and +12V power supplies and is available in a variety of packages. The μA501XR differs from the μA501X by having internal damping resistors.

Features

- +5.0V, +12V power supplies
- Single- or multi-platter Winchester drives
- Designed for center-tapped ferrite heads
- Programmable write current source
- Easily multiplexed for larger systems
- Includes write unsafe detection
- TTL compatible control signals

Block Diagram



TL/F/9407–6

Note: Caution: Use handling procedures necessary for a static sensitive component.

Absolute Maximum Ratings (Note 1)

If Military/Aerospace specified devices are required, please contact the National Semiconductor Sales Office/Distributors for availability and specifications.

Storage Temperature Range	
Ceramic DIP and Flatpak	−65°C to +175°C
Molded DIP and PLCC	−65°C to +150°C
Operating Temperature Range	0°C to +70°C
Lead Temperature	
Ceramic DIP and Flatpak (Soldering, 60 seconds)	300°C
Molded DIP and PLCC (Soldering, 10 seconds)	265°C
Internal Power Dissipation (Notes 2 & 3)	
28L-Ceramic DIP	2.50W
28L-Plastic DIP	1.92W
32L-Brazed Flatpak	1.88W
40L-Ceramic DIP	2.65W
40L-Plastic DIP	2.5W
28L-Plastic LCC	1.39W
44L-Plastic LCC	1.92W
DC Supply Voltage	
V_{DD1} and V_{DD2}	−0.3V to +14V
V_{CC}	−0.3V to +6.0V
Digital Input Voltage Range	−0.3V to V_{CC} + 0.3V
Head Port Voltage Range	−0.3V to V_{DD} + 0.3V
WUS Port Voltage Range	−0.3V to +14V
1 Write Current	60 mA
Output Current	
RDX and RDY	−10 mA
VCT	−60 mA
WUS	+12 mA

Note 1: All voltages referenced to GND.

Note 2: $T_{J\ MAX}$ = 150°C for the Plastic, and 175°C for the Ceramic.

Note 3: Ratings apply to ambient temperature at 25°C. Above this temperature, derate the 28L-Ceramic DIP at 16.7 mW/°C, the 28L-Plastic DIP at 15.3 mW/°C, the 32L-Brazed Flatpak at 12.5 mW/°C, the 40L-Ceramic DIP at 20.1 mW/°C, the 40L-Plastic DIP at 20 mW/°C, the 28L-Plastic LCC at 11.2 mW/°C, and the 44L-Plastic LCC at 15.3 mW/°C.

Recommended Operating Conditions

DC Supply Voltage	
V_{DD1}	12V ±10%
V_{CC}	5V ±10%
Head Inductance (Lh)	5.0 μH to 15 μH
Damping Resistor (External)	
RD (DP501X Only)	500Ω to 2000Ω
RCT Resistor	90Ω ±5.0% (½W)
Write Current (I_W)	25 mA to 50 mA

DC Electrical Characteristics

V_{DD1} = 12V ±10%, V_{CC} = 5.0V ±10%, 0°C ≤ T_A ≤ +70°C, unless otherwise specified

Symbol	Parameter		Conditions		Min	Max	Units
I_{CC}	Supply Current		Read/Idle Mode			25	mA
			Write Mode			25	
I_{DD}	Supply Current		Idle Mode			20	mA
			Read Mode			40	
			Write Mode			20 + I_W	
P_C	Power Consumption		25°C ≤ T_J ≤ 135°C	Idle Mode		400	mW
				Read Mode		650	
				Write Mode, I_W = 50 mA, RCT = 90Ω		880	
				Write Mode, I_W = 50 mA, RCT = 0Ω		1060	
V_{IL}	Digital Inputs:	Input Voltage LOW			−0.3	0.8	V
V_{IH}		Input Voltage HIGH			2.0	V_{CC} + 0.3	V
I_{IL}		Input Current LOW	V_{IL} = 0.8V		−0.4		mA
I_{IH}		Input Current HIGH	V_{IH} = 2.0V			100	μA
V_{OL}	WUS Output		I_{OL} = 8.0 mA			0.5	V
I_{OH}			V_{OH} = 5.0V			100	μA
V_{CT}	Center Tap Voltage		Read Mode		4.0 (typ)		V
			Write Mode		6.0 (typ)		V

Write Characteristics

V_{DD1} = 12V ±10%, V_{CC} = 5.0V ±10%, 0°C ≤ T_A ≤ +70°C, I_W = 45 mA, Lh = 10 μH, Rd = 750Ω (DP501X only), f(Data) = 5.0 MHz, CL (RDX, RDY) ≤ 20 pF, unless otherwise specified

Parameter	Conditions	Min	Max	Units
Write Current Range		10	50	mA
Write Current Constant "K"		129	151	V
Differential Head Voltage Swing		7.5		V (pk)
Unselected Head Transient Current	5.0 μH ≤ Lh ≤ 9.5 μH		2.0	mA (pk)
Differential Output Capacitance			15	pF
Differential Output Resistance	Without Internal Resistors	10k		Ω
	With Internal Resistors	560	940	
WDI Transition Frequency	WUS = LOW	250		kHz
Head Current Gain to I_{WC} $\left(\frac{I_W}{I_{WC}}\right)$		20 (typ)		mA/mA
Unselected Head Leakage	Sum of X and Y Side Current		85	μA

Read Characteristics

V_{DD1} = 12V ±10%, V_{CC} = 5.0V ±10%, I_W = 45 mA, CL (RDX, RDY) ≤ 20 pF, (V_{IN} is referenced to V_{CT}), 0°C ≤ T_A ≤ + 70°C, Lh = 10 μH, Rd = 750Ω, f(Data) = 5.0 MHz unless otherwise specified

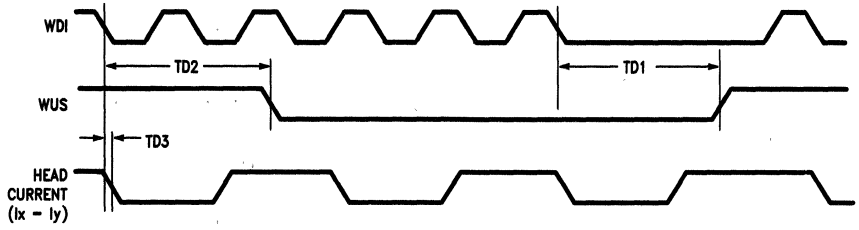
Characteristic	Condition		Min	Max	Unit
Differential Voltage Gain	V_{IN} = 1.0 mV_{PP} at 300 kHz RL (RDX), RL (RDY) = 1.0 kΩ (AC coupled)		80	120	V/V
Dynamic Range	Input Voltage, V_I, where Gain Falls by 10% V_{IN} = V_I + 0.5 mV_{PP} at 300 kHz		−3.0	3.0	mV
Bandwidth (−3 dB)	|Zs| < 5.0Ω, V_{IN} = 1.0 mV_{PP}		30		MHz
Input Noise Voltage	BW = 15 MHz, Lh = 0, Rh = 0			1.5	nV/√Hz
Differential Input Capacitance	f = 5.0 MHz			23	pF
Differential Input Resistance	f = 5.0 MHz, V_{IN} ≤ 6 mV_{PP}	Without Internal Resistors	2k		Ω
		With Internal Resistors	530	790	
Input Bias Current (per Side)				100	μA
Common Mode Rejection Ratio	V_{CM} = V_{CT} + 100 mV_{PP} at 5.0 MHz		50		dB
Power Supply Rejection Ratio	100 mV_{PP} at 5.0 MHz on V_{DD1}, V_{DD2}, or V_{CC}		45		dB
Channel Separation	Unselected Channels: V_{IN} = 100 mV_{PP} at 5.0 MHz and Selected Channel: V_{IN} = 0 mV_{PP}		45		dB
Output Offset Voltage			−480	480	mV
Common Mode Output Voltage		Read Mode	5.0	7.0	V
		Write/Idle Mode	4.3 (typ)		
Single Ended Output Resistance	f = 5.0 MHz			30	Ω
External Resistive Load (AC Coupled to Output)	Per Side to GND		100		Ω
Leakage Current (RDX, RDY)	5.0 < RDX, RDY < 8.0V Write or Idle Mode		−50	50	μA
Center Tap Output Impedance	0 ≤ f ≤ 5.0 MHz			150	Ω
Output Current	AC Coupled Load RDX to RDY		2.0		mA

Switching Characteristics

V_{DD1} = 12V ±10%, V_{CC} = 5.0V ±10%, 0°C ≤ T_A ≤ +70°C, I_W = 45 mA, Lh = 10 μH, Rd = 750Ω, f(Data) = 5.0 MHz, unless otherwise specified

Symbol	Parameter	Conditions	Min	Max	Units
R/$\overline{W}$	R/$\overline{W}$ to Write	Delay to 90% of Write Current		600	ns
	R/$\overline{W}$ to Read	Delay to 90% of 100 mV, 10 MHz Read Signal Envelope or to 90% Decay of Write Current		600	
$\overline{CS}$	$\overline{CS}$ to Select	Delay to 90% of Write Current or to 90% of 100 mV, 10 MHz Read Signal Envelope		600	ns
	$\overline{CS}$ to Unselect	Delay to 90% Decay of Write Current		600	
HS0 HS1 HS2	to Any Head	Delay to 90% of 100 mV, 10 MHz Read Signal Envelope		600	ns
WUS	Safe to Unsafe—TD1	I_W = 50 mA	1.6	8.0	μs
	Unsafe to Safe—TD2	I_W = 20 mA		1.0	
Head Current	Propagation Delay—TD3	Lh = 0 μH, Rh = 0Ω from 50% Points		30	ns
	Asymmetry	WDI has 50% Duty Cycle and 1 ns Rise/Fall Time		2	
	Rise/Fall Time	10%–90% Points		20	

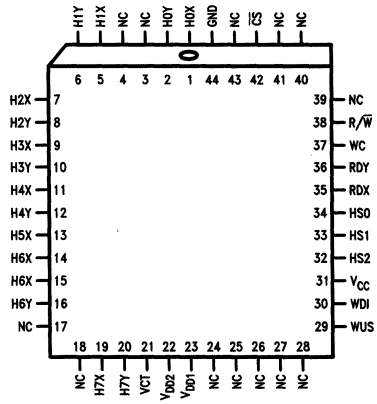
Write Mode Timing Diagram



TL/F/9407–8

Connection Diagrams

44-Lead PLCC

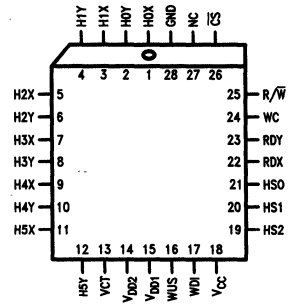


TL/F/9407-2

Top View

Order Number μA5018QC or μA5018RQC
See NS Package Number V44A

28-Lead PLCC

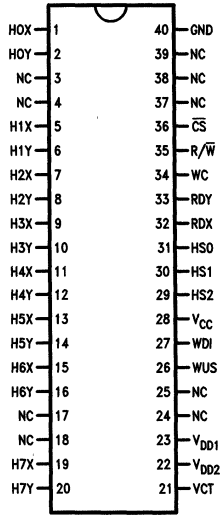


TL/F/9407-3

Top View

Order Number μA5016QC or μA5016RQC
See NS Package Number V28A

40-Lead DIP



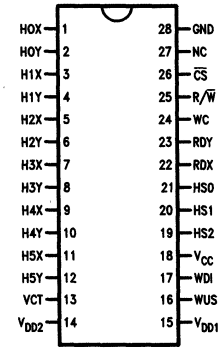
TL/F/9407-1

Top View

Ceramic DIP
***Order Number μA5018DC or μA5018RDC**
****See NS Package Number J40A**

Molded DIP
***Order Number μA5018PC or μA5018RPC**
****See NS Package Number N40A**

28-Lead DIP



TL/F/9407-5

Top View

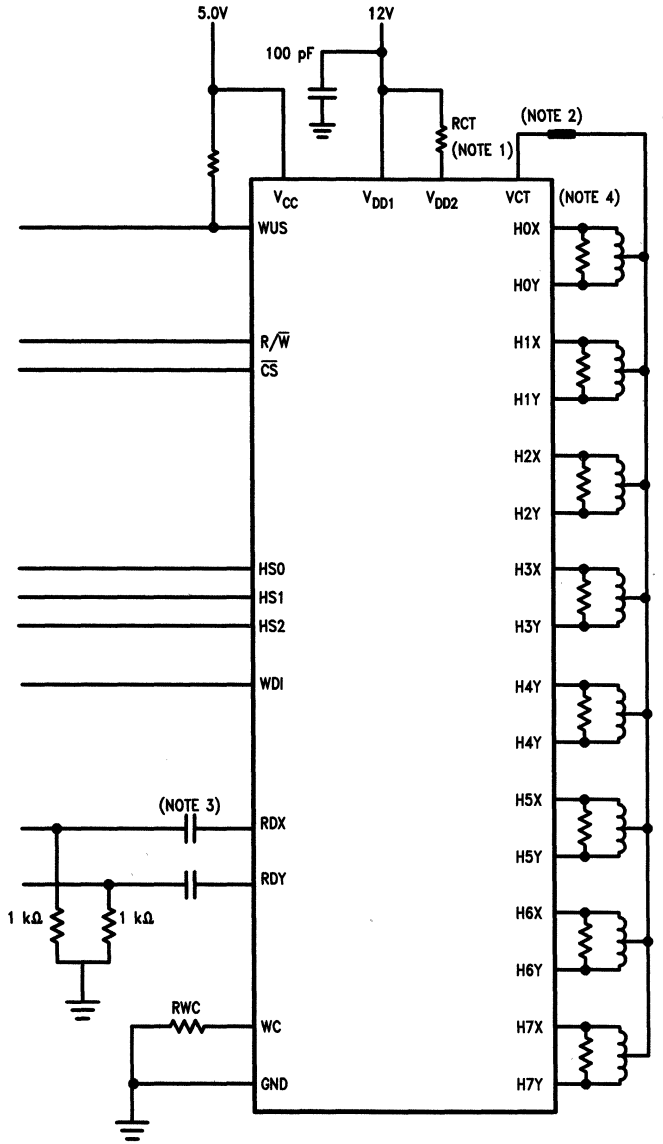
Order Number μA5016DC or μA5016RDC
See NS Package Number J28A

Order Number μA5016PC or μA5016RPC
See NS Package Number N28B

*For most current order information, contact your local sales office.

**For current package information, contact product marketing.

Application Information



TL/F/9407–7

Note 1: An external ½W resistor, RCT, given by RCT = 90 (50/I_W)Ω, where I_W is in mA can be used to limit internal power dissipation. Otherwise connect V_{DD2} to V_{DD1}.

Note 2: A ferrite bead (Ferroxcube 5659065/4A6) can be used to suppress write current overshoot and ringing induced by flex cable parasitics.

Note 3: Limit DC current from RDX and RDY to 100 μA and load capacitance to 20 pF.

Note 4: Damping resistors required on DP501X only.

Pin Descriptions

TABLE I. Description of Lead Functions

Name	Functions
HS0–HS2	Head Select
$\overline{CS}$	Chip Select: a low level enables device.
R/$\overline{W}$	Read/Write: a high level selects read mode.
WUS	Write Unsafe: a high level indicates an unsafe writing position.
WDI	Write Data In: a negative transition toggles the direction of the head current.
H0X–H7X H0Y–H7Y	X, Y Head Connections
RDX, RDY	X,Y Read Data: differential read signal out.
WC	Write Current: used to set the magnitude of the write current.
VCT	Voltage Center Tap: voltage source for head center tap.
V_{CC}	+5.0V
V_{DD1}	+12V
V_{DD2}	Positive power supply for the center tap voltage source.
GND	Ground

Circuit Operation

The μA510X/μA501XR functions as a write driver or as a read amplifier for the selected head. Head selection and mode control are described in Tables II and III. Both R/$\overline{W}$ and $\overline{CS}$ have internal pull-up resistors to prevent an accidental write condition.

WRITE MODE

The Write mode configures the μA510X/μA501XR as a current switch and activates the Write Unsafe Detector. Head current is toggled between the X- and Y-side of the recording head on the falling edges of WDI, Write Data Input. Note that a preceding read operation initializes the Write Data Flip-Flop, WDFF, to pass current through the X-side of the head. The magnitude of the write current, given by

I_W = K/Rwc, where K = Write Current Constant

is set by the external resistor, Rwc, connected from lead WC to GND.

TABLE II. Mode Select

$\overline{CS}$	R/$\overline{W}$	Mode
0	0	Write
0	1	Read
1	X	Idle

TABLE III. Head Select

HS2	HS1	HS0	Head
0	0	0	0
0	0	1	1
0	1	0	2
0	1	1	3
1	0	0	4
1	0	1	5
1	1	0	6
1	1	1	7

0 = Low Level

1 = High Level

Any of the following conditions will be indicated as a high level on the Write Unsafe, WUS, open collector output.

- Head open
- Head center tap open
- WDI frequency too low
- Device in Read mode
- Device not selected
- No write current

After the fault condition is removed, two negative transitions on WDI are required to clear WUS.

READ MODE

In the Read mode the μA510X/μA501XR is configured as a low noise differential amplifier, the write current source and the write unsafe detector are deactivated, and the write data flip-flop is set. The RDX and RDY outputs are driven by emitter followers and are in phase with the "X" and "Y" head ports. They should be AC coupled to the load.

Note that the internal write current source is deactivated for both the Read and the chip deselect mode. This eliminates the need for external gating of the write current source.

National Semiconductor

DP24H80/μA24H80
Winchester Disk Servo Preamplifier

General Description

The DP24H80/μA24H80 provides termination, gain, and impedance buffering for the servo read head in Winchester disk drives. It is a differential input, differential output design with fixed gain of approximately 100. The bandwidth is guaranteed greater than 30 MHz.

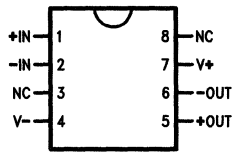
The internal design of the DP24H80/μA24H80 is optimized for low input noise voltage to allow its use in low input signal level applications. It is offered in 8-lead DIP, 10-lead flatpak, or SO-8 package suitable for surface mounting.

Features

- Low input noise voltage
- Wide power supply range (8V to 13V)
- Internal damping resistors (1.3 kΩ)
- Direct replacement for SSI 101A, with improved performance

Connection Diagrams

8-Lead DIP and SO-8 Package



TL/F/9408–1

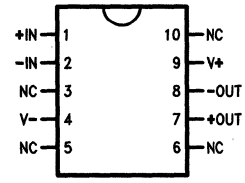
Top View

Ceramic DIP
† **Order Number μA24H80RC**
‡ **See NS Package Number J08A**

Molded Surface Mount
† **Order Number μA24H80SC**
‡ **See NS Package Number M08A**

Molded DIP
† **Order Number μA24H80TC**
‡ **See NS Package Number N08E**

10-Lead Ceramic Flatpak



TL/F/9408–2

Top View

† **Order Number μA24H80FC**
‡ **See NS Package Number F10B**

Pin Descriptions

Name	Description of Functions
V+	Positive Differential Supply with Respect to V−
V−	Negative Differential Supply with Respect to V+
+IN	Positive Differential Input
−IN	Negative Differential Input
+OUT	Positive Differential Output
−OUT	Negative Differential Output
NC	No Connection

† For most current order information, contact your local sales office.

‡ For current package information, contact product marketing.

Absolute Maximum Ratings

If Military/Aerospace specified devices are required, please contact the National Semiconductor Sales Office/Distributors for availability and specifications.

Storage Temperature Range	
Ceramic DIP and Flatpak	−65°C to +175°C
Molded DIP and SO-8	−65°C to +150°C
Operating Temperature Range	0°C to +70°C
Lead Temperature	
Ceramic DIP and Flatpak (Soldering, 60 seconds)	300 °C
Molded DIP and SO-8 (Soldering, 10 seconds)	265°C
Internal Power Dissipation (Notes 1 & 2)	
8L-Ceramic DIP	1.30W
8L-Molded DIP	0.93W
SO-8	0.81W
10L-Flatpak	0.79W
Supply Voltage	15V
Output Voltage	15V
Differential Input Voltage	±10V

Note 1: $T_{J\,MAX}$ = 150°C for the Molded DIP and SO-8, and 175°C for the Ceramic DIP and Flatpak.

Note 2: Ratings apply to ambient temperature at 25°C. Above this temperature, derate the 8L-Ceramic DIP at 8.7 mW/°C, the 8L-Molded DIP at 7.5 mW/°C, the SO-8 at 6.5 mW/°C, and the Flatpak at 5.3 mW/°C.

Electrical Characteristics T_A = 25°C, V_{CC} = 8V to 13.2V, unless otherwise noted

Symbol	Parameter	Conditions	Min	Typ	Max	Units
G	Gain (Differential) (Note 4)	R_p = 130Ω, V_{CC} = 12V	80	100	120	
		R_p = 130Ω, V_{CC} = 12V T_A = 0°C to +70°C	70		130	
BW	Bandwidth (3.0 dB) (Note 2)	V_I = 0.5 mV_{p-p}	30	65		MHz
R_I	Input Resistance		1040	1300	1560	Ω
C_I	Input Capacitance			3		pF
V_I	Input Dynamic Range (Differential)	R_p = 130Ω, V_{CC} = 12V	3			mV_{p-p}
I_S	Supply Current	V_{CC} = 12V		20	25	mA
ΔV_O	Output Offset (Differential)	R_p = 130Ω, R_s = 0Ω			200	mW
V_n	Equivalent Input Noise (Notes 2 & 3)	R_s = 0Ω, BW = 4 MHz		1.5	2	μV
PSRR	Power Supply Rejection Ratio (Note 1)	R_s = 0Ω, f = 5 MHz	55	70		dB
ΔG/ΔV	Gain Sensitivity (Supply)	R_p = 130Ω, ΔV_{CC} = ±10%			±0.5	%/V
ΔG/ΔT	Gain Sensitivity (Temp)	R_p = 130Ω, T_A = 25°C to +70°C		−0.1		%/°C
CMR	Common Mode Rejection (Note 1) (Input)	f = 5 MHz	60	75		dB

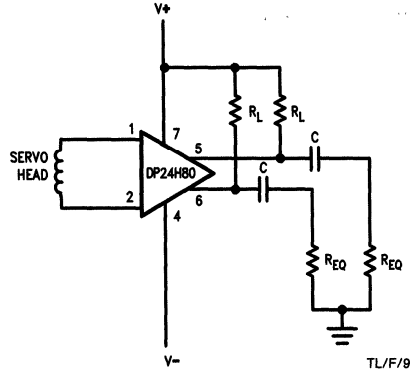
Note 1: Tested at DC, guaranteed at frequency.

Note 2: Guaranteed, but not tested in production.

Note 3: Equivalent input noise (additional specification):

Typ	Max	Unit	Condition
3	4	μV	BW = 15 MHz[2]
0.85	1.0	$nV/\sqrt{Hz}$	BW = 15 MHz[2]

Typical Applications



TL/F/9408–3

Note 1: Leads shown for 8-lead DIP.

Note 2: R_{eq} is equivalent load resistance.

Note 3: $R_p = \frac{R_L \bullet R_{eq}}{R_L + R_{eq}}$

Note 4: G = 0.77 R_p
Where R_p = value from Note 3 (above) in ohms.

National Semiconductor

DP2580/μA2580
Winchester Disk Servo Preamplifier

General Description

The DP2580 provides termination, gain, and impedance buffering for the thin film servo read head in Winchester disk drives. It is a differential output design with fixed gain of approximately 250. The bandwidth is guaranteed greater than 30 MHz.

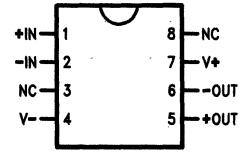
The internal design of the DP2580 is optimized for low input noise voltage to allow its use in low input signal level applications. It is offered in 8-lead ceramic DIP, 10-lead Flatpak, and an SO-8 package suitable for surface mounting.

Features

■ Low input noise voltage	Typ. 0.5 nV/√Hz
■ Wide power supply range	8V to 13V
■ Internal damping resistors	1 kΩ

Connection Diagrams

8-Lead DIP and SO-8 Package



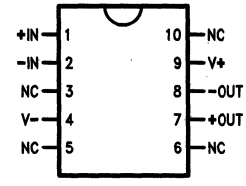
TL/F/9409-1

Top View

†Order Number μA2580DC
††See NS Package Number N08E

†Order Number μA2580SC
††See NS Package Number M08A

10-Lead Ceramic Flatpak



TL/F/9409-2

Top View

†Order Number μA2580FC
††See NS Package Number F10B

Pin Description

Name	Function
+IN	Positive Differential Input
−IN	Negative Differential Input
NC	No Connection
V^-	Negative Differential Supply with Respect to V_{CC}
+OUT	Positive Differential Output
−OUT	Negative Differential Output
V^+	Positive Differential Supply with Respect to V_{CC}
NC	No Connection

†For most current order information, contact your local sales office.
††For most current package information, contact product marketing.

Absolute Maximum Ratings

If Military/Aerospace specified devices are required, please contact the National Semiconductor Sales Office/Distributors for availability and specifications.

Storage Temperature Range	
Ceramic DIP and Flatpak	−65°C to +175°C
SO-8	−65°C to +150°C
Operating Temperature Range	0°C to +70°C
Lead Temperature	
Ceramic DIP and Flatpak	
(Soldering, 60 seconds)	300°C
SO-8	
(Soldering, 10 seconds)	265°C
Internal Power Dissipation (Notes 1 and 2)	
8L—Ceramic DIP	1.3W
10L—Flatpak	0.79W
SO-8	15V
Supply Voltage	15V
Output Voltage	15V
Differential Input Voltage	±1V

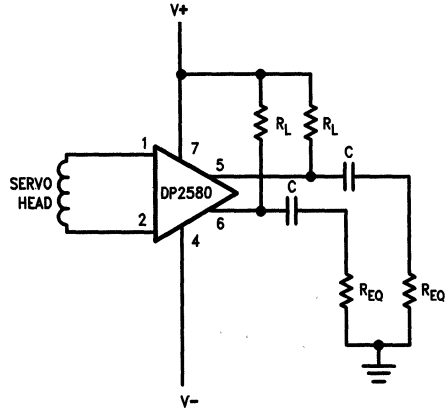
Note 1: T_J Max = 150°C for the SO-8, and 175°C for the Ceramic DIP and Flatpak.

Note 2: Ratings apply to ambient temperature at 25°C. Above this temperature, derate the 8L—Ceramic DIP at 8.7 mW/°C, the 10L—Flatpak at 5.3 mW/°C, and the SO-8 at 6.5 mW°C.

Electrical Characteristics T_A = 25°C, V+ − V− = 8V to 13.2V, unless otherwise specified

Symbol	Parameter	Conditions	Min	Typ	Max	Units
G	Gain (Differential)	R_P = 100Ω, (V+) − (V−) = 12V		250		
BW	Bandwidth (3 dB)	V_I = 0.5 mV_{p-p}	30	65		MHz
R_I	Input Resistance			300		Ω
C_I	Input Capacitance			35		pF
V_I	Input Dynamic Range (Differential)	R_P = 100Ω, (V+) − (V−) = 12V			1	mV_{PP}
I_S	Supply Current	(V+) − (V−) = 12V		28	40	mA
ΔV_O	Output Offset (Differential)	R_S = 0Ω, R_P = 100Ω	600		600	mV
V_n	Equivalent Input Noise	BW = 4 MHz		0.6		$nV/\sqrt{Hz}$
PSRR	Power Supply Rejection Ratio	R_S = 0Ω, f = 5 MHz	50	65	0.90	dB
ΔG/V	Gain Sensitivity (Supply)	Δ (V+) − (V−) ±10%, R_P = 100Ω			0.5	%/V
ΔG/T	Gain Sensitivity (Temp.)	T_A = 25°C to 70°C, R_P = 100Ω		0.16		%/°C
CMR	Common Mode Rejection (Input)	f = 5 MHz	60	70		dB

Typical Applications (Notes 1–4)



TL/F/9409–3

Note 1: Leads shown for 8-lead DIP.

Note 2: R_{EQ} is equivalent load resistance.

Note 3: $R_p = \frac{R_L \bullet R_{EQ}}{R_L + R_{EQ}}$

Note 4: G = 2.5 R_P
Where R_P = value Note 3 (above) in Ω.

National Semiconductor

DP2460/DP2461, μA2460/μA2461 Servo Control Chips

General Description

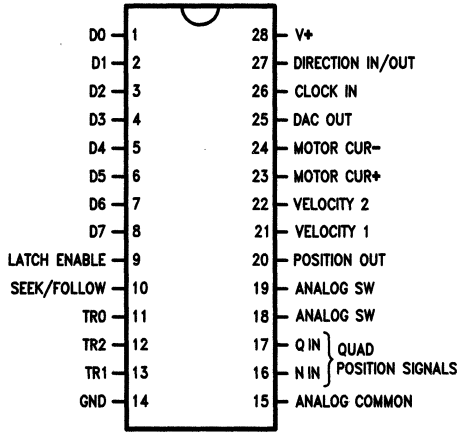
The DP2460 and DP2461 provide the analog signal processing required between a drive resident microprocessor and the servo power amplifier for Winchester disk closed loop head positioning. The DP2460 and DP2461 receive quadrature position signals from the servo channel; and from these, derive actual head seek velocity as well as position-mode off-track error. In the seek mode, the Digital to Analog Converter (DAC) is used to command velocity, while actual velocity is obtained by differentiating the quadrature position signals provided at V1 for external processing. The velocity signal (V2), obtained by integrating the motor current, is also available for extra damping, if desired. Further, the DAC may be used for detenting the head off-track for any purpose such as thermal compensation or soft-error retries.

Features

- Microprocessor compatible interface
- Quadrature di-bit compatible
- On board DAC
- Velocity V1 derived from position signal
- Velocity V2 derived from motor current
- Quarter-Track-Crossing signal outputs
- Minimal external components
- Compatible with DP2470 demodulator

Connection Diagrams

28-Lead Ceramic DIP

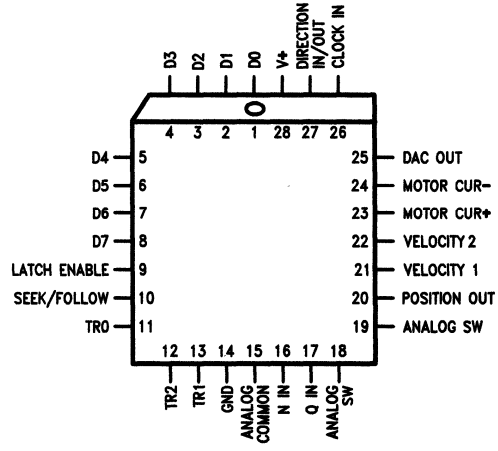


TL/F/9410–1

Top View

†Order Number μA2460DC or μA2461DC
††See NS Package Number J28A

28-Lead PLCC



TL/F/9410–2

Top View

†Order Number μA2460QC or μA2461QC
††See NS Package Number V28A

†For most current order information, contact your local sales office.
††For most current package information, contact product marketing.

7

Absolute Maximum Ratings

If Military/Aerospace specified devices are required, please contact the National Semiconductor Sales Office/Distributors for availability and specifications.

Storage Temperature Range
- Ceramic DIP: −65°C to +175°C
- PLCC: −65°C to +150°C

Operating Temperature Range: 0°C to +70°C

Lead Temperature
- Ceramic DIP (Soldering, 60 sec.): 300°C
- PLCC (Soldering, 10 sec.): 265°C

Internal Power Dissipation (Notes 1 and 2)
- 28L—Ceramic DIP: 2.50W
- 28L—PLCC: 1.39W

Supply Voltage: 15V Max
- Analog Common Voltage: 8.0V Max
- All Inputs: V_{supply} Max

Note 1: T_J max = 150°C for the PLCC, and 175°C for the Ceramic DIP.

Note 2: Ratings apply to ambient temperature at 25°C. Above this temperature, derate the 28L—Ceramic DIP at 16.7 mW/°C, and the 28L—PLCC at 11.2 mW/°C.

Electrical Characteristics

T_A = 0°C to 70°C, V_{CC} = 12V, f_{CLK} = 2.0 MHz, Analog Common = 5.0V, unless otherwise specified

Symbol	Parameter	Conditions	Min	Typ	Max	Units
Digital I/O	Input Voltage LOW				0.8	V
	Input Voltage HIGH		2.0			
	Output Voltage LOW	I_{OL} = 2.5 mA			0.45	
	Output Voltage HIGH	I_{OH} = 40 μA	2.4			
	Input Load Current	V_I = 0V to V_{CC}			0.2	mA
Clock Input	Input Comparator Reference Level		2.0	2.5	3.0	V
	Input Impedance		15	20		kΩ
DAC	Linearity (Note 1)		−1		1	LSB
	Resolution			8.0		bits
	Differential Nonlinearity		Monotonicity Guaranteed			
	Full Scale Output Voltage	Direction in High	7.25	7.35	7.45	V
		Direction in Low	2.55	2.65	2.75	
	Zero Scale Voltage			5.0		
	Output Offset Voltage				±10	mV
	Settling Time (Notes 2, 4)	To ½ LSB All bits ON or OFF				μs
Position Inputs	Input Voltage Range		1.0		9.0	V
	Input Impedance		15	20		kΩ
Analog Switch	On Resistance	V_{CM} = 0V to 12V		100	200	Ω
	Off Leakage (Note 3)			2.0	100	nA
Position Output	Output Voltage Swing	R_L = 15k Follow Mode	1.0		9.0	V
	Voltage Gain		0.9		1.1	—
	Output Offset Voltage				±20	mV
Velocity Outputs	Output Voltage Swing	R_L = 15k	1.0		9.0	V
	Output Offset Voltage V2				±20	mV
	Output Offset Voltage V1				15	
I_{CC}	Positive Supply	V_{CC} = 13.2V		10	15	mA
I_{SS}	Negative Supply	V_{CC} = 13.2V	−15	−10		mA
I_{AC}	Analog Common I		−2.0	0	2.0	mA
V1—Differentiator	Linearity	f_{CLK} = 1.0 MHz to 4.0 MHz; $f_{N/Q} \leq$ 10 kHz		0.25		%
V2—Integrator	Linearity	f_{CLK} = 1.0 MHz to 4.0 MHz		1.0		%

Note 1: DAC Linearity is a function of the Clock frequency; Linearity at 1.0 MHz is typically ±½ LSB.

Note 2: DAC Settling Time is approx. 5.0 μs, plus a delay of maximum 32 × Clock period i.e., 5 + 32 μs at Clock = 1.0 MHz Minimum could be 5.0 μs.

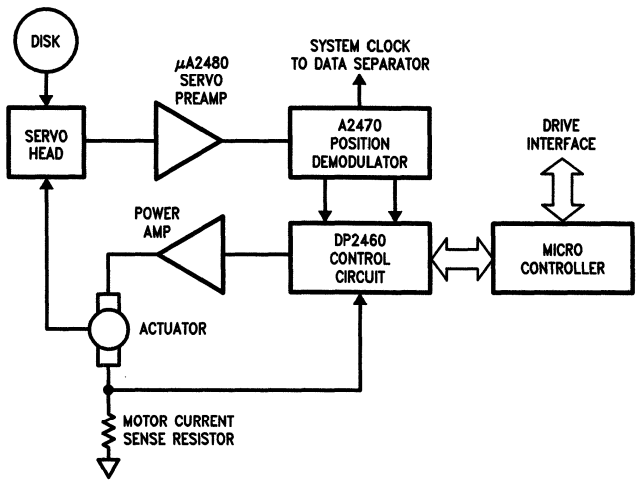
Note 3: Equivalent to 50 MΩ.

Note 4: Guaranteed, but not tested in production.

Pin Description

Pin No.	Name	Function
INPUTS		
1–8	DAC Input Word (D_0–D_7)	Programs DAC output, 00000000 = Analog Command Lead 1 = LSB Lead 8 = MSB
9	Latch Enable	Allows present DAC input word to be latched.
10	Seek/Follow Mode	Configures the feedback loop for either seeking or track-following. (High = Seek, Low = Follow)
14	Ground	
15	Analog Common	Analog signal reference input level (5.0V)
16	N	Normal position input signal.
17	Q	Quadrature position input signal.
23	Motor Current +	Motor current sense input to motor current integrator.
24	Motor Current −	
26	Clock	4.0 MHz (maximum) input square wave.
27	Direction In/Out	Changes the polarity of DAC output from positive to negative consistent with the desired direction of head motion.
28	V+	12V supply

Pin No.	Name	Function
OUTPUTS		
11	Track 2^0 (TR0)	TTL signal whose frequency is 8 times N (or Q).
12	Track 2^2 (TR2)	TTL signal indicating N > Q (for DP2460). TTL signal whose frequency is 2 times N (or Q) (for DP2461).
13	Track 2^1 (TR1)	TTL signal indicating $\overline{N}$ > Q (for DP2460). TTL signal whose frequency is 4 times N (or Q) (for DP2461).
18	Analog Switch	Analog switch to be used externally for changing from seek to follow.
19	Analog Switch	
20	Position Output	Analog signal representing sensed off track ampitude.
21	Velocity 1	Analog output representing velocity processed from position signals N and Q.
22	Velocity 2	Analog output representing the integral of motor current.
25	DAC Output	Used to command velocity and position.

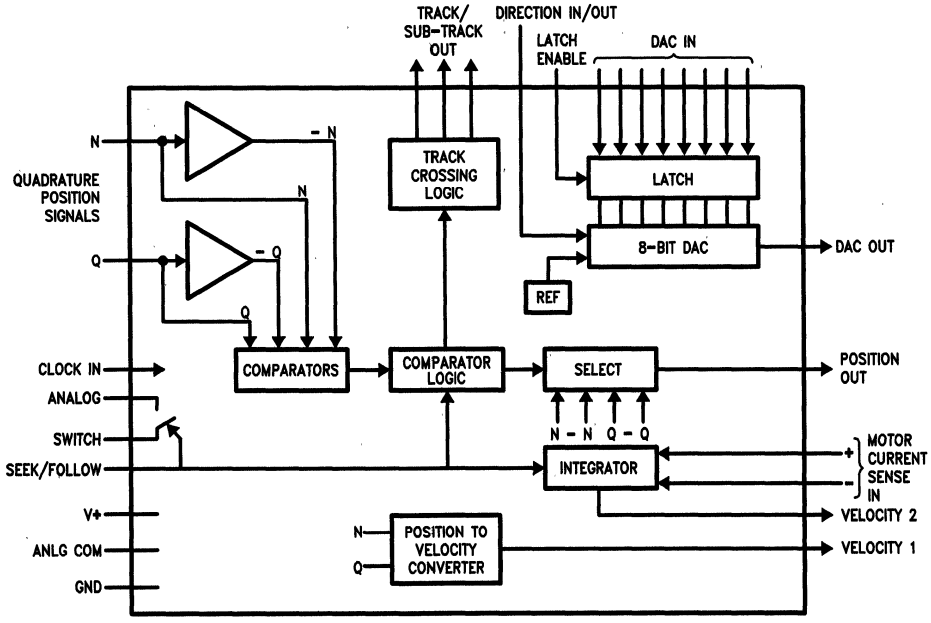


TL/F/9410–3

FIGURE 1. Head Actuator Control System

7

Functional Description



TL/F/9410–4

FIGURE 2. Block Diagram

Figure 2 shows a block diagram of the DP2460/DP2461 Servo Controller.

POWER SUPPLY AND REFERENCE REQUIREMENTS

The DP2460/DP2461 is designed to operate from a single supply of 10V to 12V. Also required is a reference voltage of 5.0V called Analog Common which serves two functions; all analog signals will be referenced to this voltage and in addition the internal DAC will use it to set full scale.

A clock signal must be provided as a reference for the internal switched capacitor position differentiator and motor current integrator. The clock signal should be a sine or square wave between Analog Common and ground at a maximum frequency of 4.0 MHz.

All digital inputs and outputs are TTL compatible levels referenced to ground.

INPUT SIGNALS AND TRACK CROSSING OUTPUTS

The input format selected for position feedback is consistent with a large class of sensors that generate two cyclical output signals displaced in space phase by 90 degrees (quadrature signal pairs). These sensors include resolvers, inducto-syns, optical encoders, and most importantly, servo demodulators designed for rigid disk head position sensing.

The input signals N and Q are quadrature quasi triangular waveforms with amplitudes of ±2.5V nominal referenced to Analog Common. The periods of the input signals are subdivided by internal comparators and logic and sent to the Track Crossing outputs T_0, T_1, and T_2. The relationship of these outputs to the inputs N and Q is shown in *Figure 3a* (for DP2460) and *Figure 3b* (for DP2461).

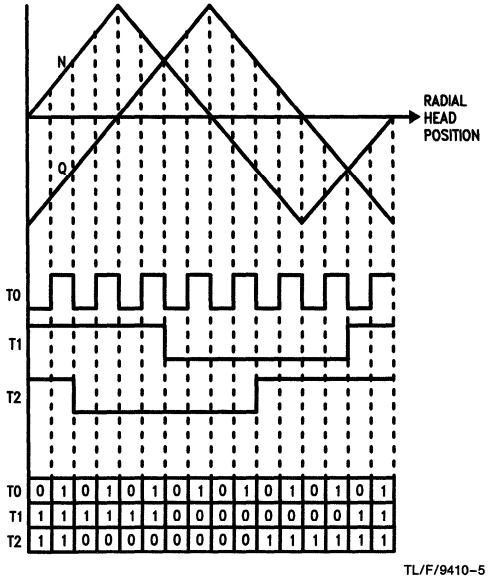
Note that different servo patterns may yield different numbers of track centerlines for each period of the quadrature signal pair. The relationship of T_0, T_1, and T_2 to N and Q is independent of track centerlines, leaving the correct interpretations to the microcontroller.

DAC

The DAC is an 8-bit, buffered input, voltage output digital to analog converter. The output voltage with an input code of all zeros is equal to Analog Common. Full scale is equal to Analog Common ±2.35V. The polarity depends on the Direction In Signal; Direction In High will result in a positive DAC output.

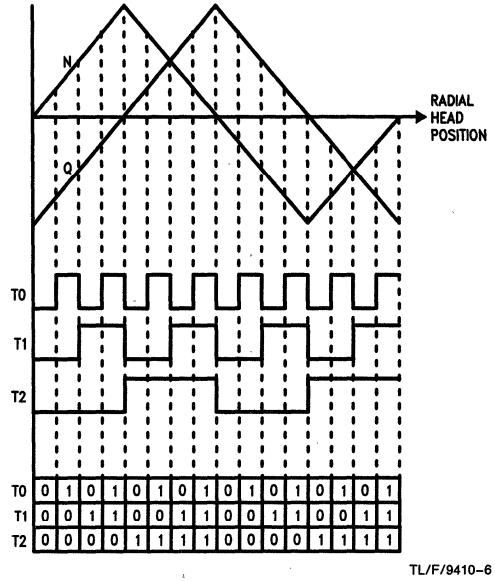
The DAC enable line when high will cause the DAC's input buffer to become transparent, i.e. input data will affect the output voltage immediately. When DAC enable is brought

Functional Description (Continued)



TL/F/9410–5

FIGURE 3a. Track Crossing Outputs (for DP2460)



TL/F/9410–6

FIGURE 3b. Track Crossing Outputs (for DP2461)

low the data present on the input lines will be latched and any further changes to the input data will not change the output voltage. The DAC functions in both Seek and Follow Mode. During Seek Mode the DAC output is used as a velocity reference. In Follow Mode the DAC output can be summed into the position reference signal to offset the heads from track center.

ANALOG SWITCH

An uncommitted single pole single throw analog switch with an ON resistance of approximately 100Ω is provided. This switch is ON during Follow Mode.

MODE SELECT

The two major intended operating modes for the DP2460 are controlled by the microcontroller via the SEEK/FOLLOW input. Mode Select input high enables Seek Mode, low enables Track Follow Mode.

SEEK, when asserted by the microcontroller along with DIRECTION and a non-zero VELOCITY value as inputs, causes the actuator system to accelerate in the requested direction. During the ensuing motion, the actuator system will come under velocity feedback control. The velocity feedback signal is created by differentiation of the quadrature position signals and, additionally, by integration of motor current.

FOLLOW, the negation of SEEK, changes the feedback loop to a track-following or position mode. Position servos are typically second order systems and without loop compensation are potentially unstable. External components are used, along with the DP2460, to achieve stable track following performance. Velocity information (V1) is made available as an output in this mode to aid in stabilizing certain loops. If non-zero data is supplied to the velocity latches in this mode, it will result in a track offset in the direction indicated by DIRECTION IN/OUT. *Figure 4* shows typical seek operation.

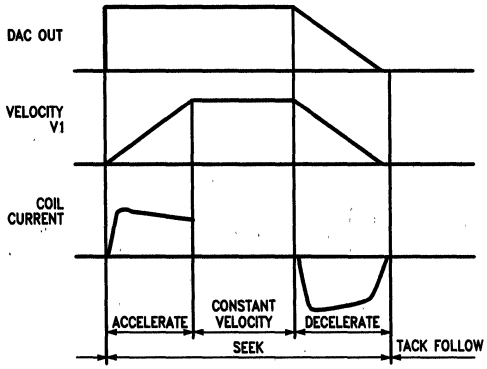
POSITION OUTPUT

When the DP2460/DP2461 is set to Seek Mode the signal from Position Output lead is shown in *Figure 5*. This signal is made by switching the position inputs, (N and Q) through an inverter if required, ($\overline{N}$ and $\overline{Q}$) to the output using the track crossing signals. It can be used, if desired, to interpolate between DAC steps by attenuating it and summing it with the DAC output.

Track Follow Mode is entered when the heads are near the end of a seek, usually within one half to one track away from the target track centerline. The final setting to the track center is done by the position loop.

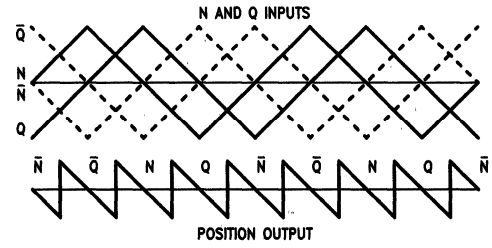
When the device is switched to Follow Mode, the position input signal (N, $\overline{N}$, Q or $\overline{Q}$) that is currently selected to the output is latched and the Position Out signal follows the selected position input signal until the device is switched back to Seek Mode. This implies that the switch to Follow Mode must not be made until the signal that will be the correct Position error signal for the target track is present at the output. If track centers are defined as the zero crossings of both N and Q this means that the switch to Follow Mode must be made less than one-half track away from the target track. (This is with respect to the convention of 4 tracks per encoder cycle, so switching must be done within 90° of the period of N or Q.)

Functional Description (Continued)



TL/F/9410–7

FIGURE 4. Typical Seek



TL/F/9410–8

FIGURE 5. Position Output during Seek Mode

VELOCITY OUTPUTS

There are two analog signal outputs representing velocity. The first (V1) is derived by differentiating the position input signals. The entire differentiator is on-chip, using switched capacitor techniques and requires no external components.

The transfer function of the differentiator is:

$$V_O = dv/dt\,(input) \times 14.3/f\,(clock)\ Hz$$

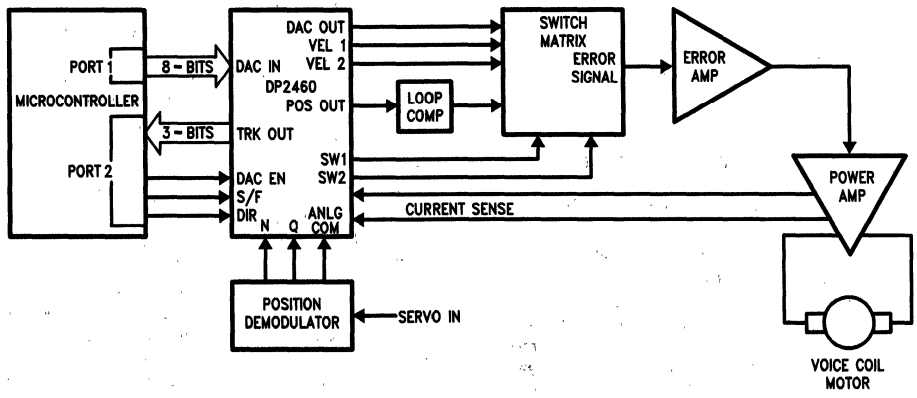
As an example; a 10 kHz triangular signal pair into N and Q of 6.0V peak-to-peak amplitude (dv/dt = 120 kV/s) would result in a velocity voltage output of 1.716V referenced to Analog Common with a clock of 1.0 MHz. The polarity will be positive if N is leading Q by 90 degrees and negative if Q is leading N. This block functions during both Seek and Follow modes.

The second velocity output is obtained by integrating a voltage proportional to the current in the motor using the following function:

$$dv/dt\,(out) = V\,(+I_{in} - -I_{in}) \times 2 \times 10^{-4} f\,(clock)\ Hz.$$

The motor current integrator output is clamped to Analog Common during Follow Mode and is released at the initiation of a seek.

Figure 6 shows a typical application setup for the Servo Control chip.



TL/F/9410–9

FIGURE 6. Typical Application Setup

National Semiconductor

2470A Servo Demodulator

General Description

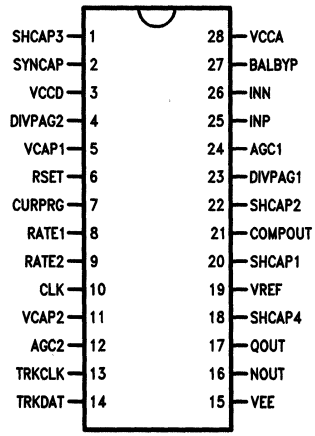
The new 2470A servo demodulator decodes the quadrature di-bit pattern from the dedicated servo surface providing position and data information.

Features

- Quadrature positions signals
- Phase locked to servo pattern with embedded lock indication
- Track data and track clock for data encoding
- AGC amplifier with 36 dB range
- Servo fields to 400 kHz
- Compatible with the 24H80 servo preamp and 2460 servo control chip
- Standard 5V and 12V supplies
- New phase detector eliminates jitter due to dropped sync's
- New lock detector uses sync pulse location to determine sync. Dropped pulses are not out of sync conditions.
- New ±20% VCO with extended frequency capability (>30 MHz)
- New totem pole TTL outputs
- New sync detector eliminates one shot multivibrator setting
- New sample and hold circuits eliminate output droop and glitching of the quadrature circuits
- New reference centers the quadrature outputs in the 12V supply
- New sync window controller prevents erroneous pulses from reaching the phase detector for a second level of jitter prevention

Connection Diagrams

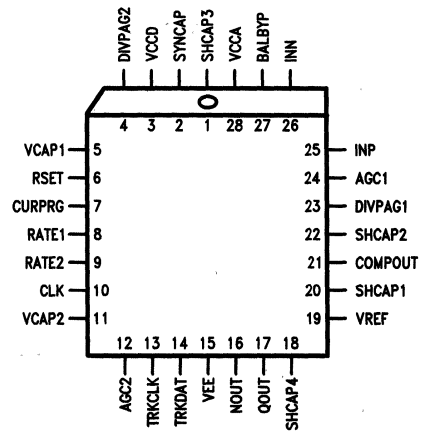
28-Pin Molded DIP



TL/F/9411–1

Top View
† Order Number 2470PC
‡ See NS Package Number N28B

28 PLCC



TL/F/9411–2

Top View
† Order Number 2470QC
‡ See NS Package Number V28A

† For most current order information, contact your local sales office.

‡ For most current package information, contact product marketing.

Absolute Maximum Ratings

If Military/Aerospace specified devices are required, please contact the National Semiconductor Sales Office/Distributors for availability and specifications.

Storage Temperature	−65°C to +175°C
Operating Temperature	0°C to +70°C
Lead Temperature Ceramic DIP (10 sec.)	300°C
Internal Power Dissipation	2.5W
Supply Voltage V_{CCD}	6V
Supply Voltage V_{CCA}	15V

2470A Electrical Specification $T_A = 25°C$, $V_{CCD} = 5V$, $V_{VCCA} = 12V$

Parameter	Conditions	Min	Typ	Max	Units
AGC AMPLIFIER					
Max Voltage Gain	Input Freq. = 1 MHz	40	46		dB
AGC Range	Input Freq. = 1 MHz	20	36		dB
Frequency Response			10		MHz
Input Voltage Range		30		300	mV
Output Voltage		3.0	3.3	3.6	V_{PP}
Common Mode Voltage			8.2		V
QUADRATURE OUTPUTS (Referred to 6V ref; R_L = 20k)					
Output Voltage	R_L = 20k	3.0	3.3	3.6	V_{PP}
Output Impedance				100	Ω
Output Offset Voltage			±5	+20	mV
Output Current	(Note: Out Impedance)		5	6	mA
VOLTAGE REFERENCE					
Output Voltage		5.88	6.00	6.12	V
Output Current			5	6	mA
V_{CO}					
Max Frequency V_{CO} (Ctr)			30		MHz
PLL System Performance using sine³ Waveform as Servo Reference. Frame(center) = V_{CO}(center)/divider ratio.					
Acquisition Range		±10%	±15%		frame(ctr)
Dropped Sync Endurance		15	40	—	frames
Maximum Frame Rate		400			kHz
LOGIC					
Input Voltage Low				0.8	V
Input Voltage High		2.0			V
Output Voltage Low				0.5	V
Output Voltage High		2.7			V
Risetime	10%–90%		9	20	ns
Falltime	10%–90%		4	14	ns

DIVIDER TABLE Ratio = V_{CO} Frequency ÷ Frame Rate

DIVPAG1	DIVPAG2	RATIO
0	0	32
1	0	64
0	1	96
1	1	128

Power Supply Ratings				
V_{CCD} (5V)		80	100	mA
V_{CCA} (12V)		40	60	mA

Features of the 2470A Servo Demodulator

1) The sync detecting operation is based on the servo disk's own timing and eliminates the need to precisely set a resistor-capacitor time constant for the di-bit detecting one shot timer. The new circuit uses a single low precision capacitor.
2) The phase detector has a linear phase vs. output detection scheme as an improvement over the one shot scheme. The circuit performs no detection for dropped sync pulses and when in lock as defined by the lock detector, it will only detect in a predefined window. These features eliminate jitter caused by dropped pulses and/or bad servo areas on the disk. Also eliminated are the phase detector external components.

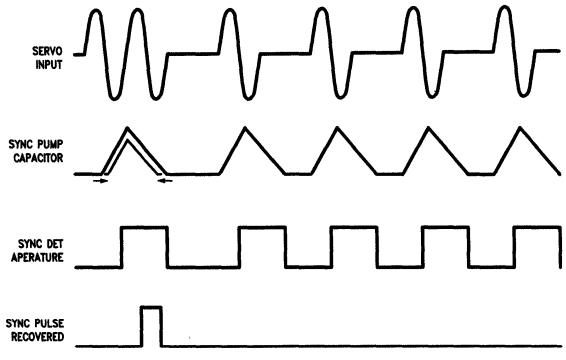
   Out of lock conditions require acquisition aids to achieve lock. Should a sync pulse show outside the sync window (2 of 32 counts in a servo field), aperature control circuits realign the sync pulse with the sync window by resetting the decoder and enlarge the next window to find a sync pulse with the VCO's ±20% tuning range. The limited range on the VCO prevents 2X locks. The aperature control prevents the dropped pulse ignoring phase detector from achieving non-integral false locks. The window realignment and enlargement is disabled during lock to prevent erroneous sync pulses from upsetting the decoder.
3) The new lock detector ignores dropped pulses in testing for in and out of lock conditions. Should a sync pulse appear the detector records whether or not it appeared in the normal sync window. The lock detector uses four consecutive sync pulses either all out or all in the sync window to determine lock status. The lock detector enables and disables the aperature control for the phase detector and the sync data detector.
4) The 2470A has a VCO with improved performance. It has > 30 MHz operation and a restricted tuning range of ±20%. Tuning circuits will reduce jitter due to parasitic couplings into the VCO.
5) New sample hold circuits for the N and Q decoders eliminate the droop in the N and Q outputs. The sample holds are opened immediately after the peak detection is complete. This eliminates droop induced offsets and glitching.
6) TTL totem pole outputs eliminates the need for resistive pullup for the output. Switching times of 10 ns are achieved.
7) The analog reference is 6V. Centering in the 12V supply lines is easier. The 6V reference maintains compatibility with the 2460 servo controller and the 24H80 preamp.

List of Lead Functions

Lead	Name	Function
INPUT SIGNALS		
23 4	DIVPAG1 DIVPAG2	Programs the prescaler for the VCO Divide ratios are 32, 64, 96 and 128
7	CURPRG	Voltage sets PLL charge pump bias current
15	V_{EE}	Ground 0V
3	V_{CCD}	+5V supply
28	V_{CCA}	+12V supply
25 26	INP INN	Composite inputs to the AGC amplifier
OUTPUTS		
13	TRKCLK	Clock output for data during lock, TTL
14	TRKDAT	Data from dropped sync pulses TTL
10	CLK	VCO output TTL
21	COMPOUT	Output of AGC amplifier @8.2V CM
19	V_{REF}	6V reference for N and Q outputs
16	N_{OUT}	Normal position signal @6V CM
17	Q_{OUT}	Quadrature position signal @6V CM

List of Lead Functions (Continued)

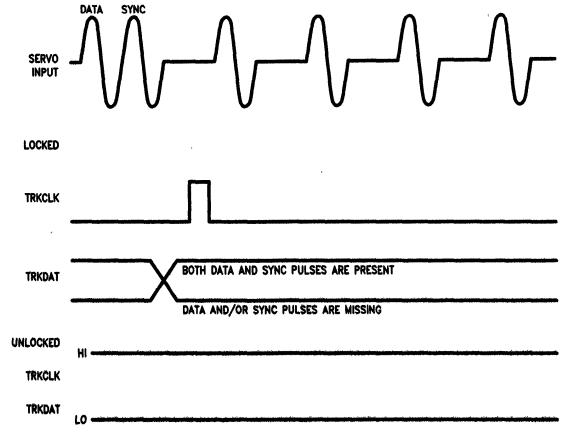
Lead	Name	Function
EXTERNAL COMPONENTS		
2	SYNCAP	Timing capacitor for the sync detector
5–11	V_{CAP} 1 & 2	VCO timing capacitor
8–9	Rate 1 & 2	PLL loop filter
27	BALBYP	DC offset restore filter capacitor.
24	AGC1	AGC system loop filter
12	AGC2	AGC2 Pin includes an amplitude control function. This pin has a nominal voltage of 5V. The amplitude increases according to the formula: $\frac{V_{(COMPOUT\ P\text{-}P)} - V_{(COMPOUT\ NOM\ P\text{-}P)}}{V_{(AGC2)} - V_{(AGC2\ NOM)}} = -0.7$ AGC2 is Pin 12 and COMPOUT is Pin 21.
6	R_{SET}	Sets the VCO bias currents I < 2 mA
20, 22, 1, 18	SHCAP 1 . . . 4	Four sample hold capacitors



TL/F/9411–3

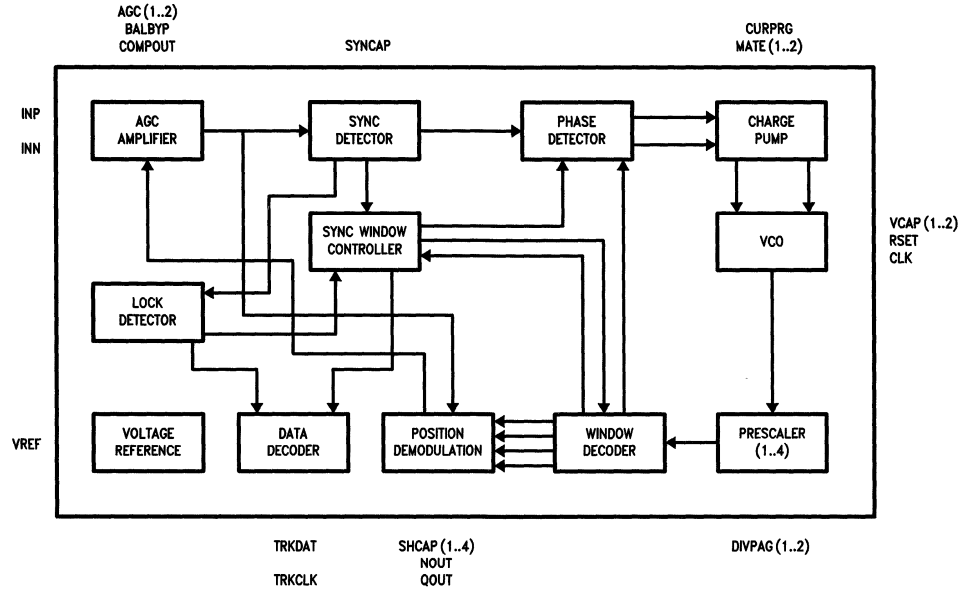
The sync pulse gate is triggered by the sync det aperature and is locked open until the sync goes to zero. The locking mechanism prevents clipping the negative edge of the sync.

FIGURE 1. Sync Detector Diagram



TL/F/9411–4

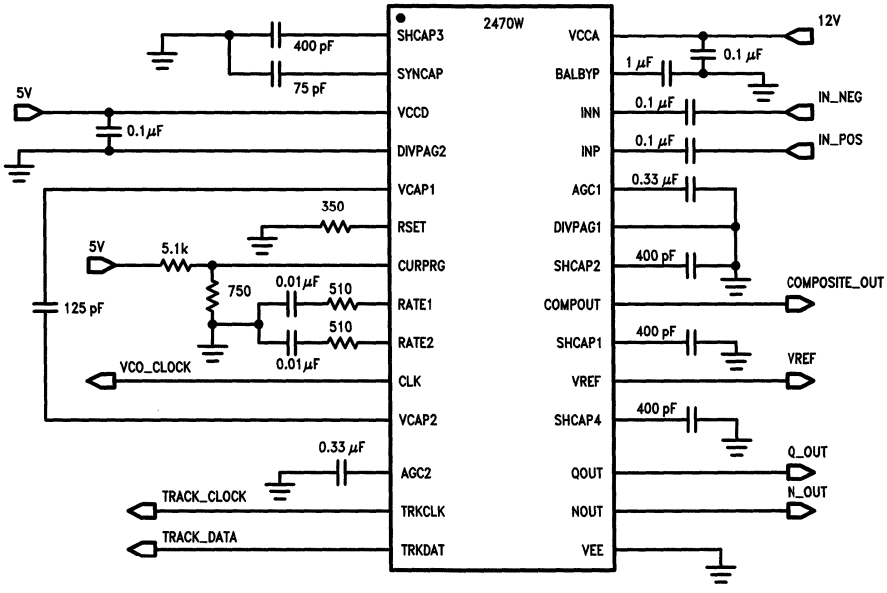
FIGURE 2. Track Data Output Information



TL/F/9411–5

FIGURE 3. 2470A Block Diagram

This test circuit runs at about a 136 kHz frame rate



TL/F/9411–6

FIGURE 4. 2470A Test Circuit

Section 8

Disk Drive Microcontroller Circuits

Section 8 Contents

National Semiconductor

The 16-Bit HPC™ Family: Optimized for Mass Storage System Performance

Key Applications

- Asynchronous/Synchronous SCSI Controller System
- Intelligent Drive Applications in Rigid Disk, Optical Disk, High Density Floppy Disk and Tape Drives

The high-performance HPC16083 16-bit microcontroller, for example, is ideally suited for synchronous SCSI (Small Computer System Interface) controller system's CPU and repository for the firmware. The device's intruction set, including fast multiply and divide, customizes it to real-time control applications. Spare CPU capacity enables designers to implement other software tasks, such as power on/off, disk spindle control, head positioning, and an expanded SCSI Common Command Set.

Key Features

- World's first 16-bit CMOS microcontroller
- World's fastest CMOS microcontroller
- 67 ns instruction-cycle time at 30 MHz
- Full 16-bit architecture and implementation
- 64 kbyte address space
- High code efficiency with single-byte, multiple-function instructions
- 16 x 16-bit multiply, 32 x 16-bit divide
- Eight vectored interrupt sources
- Watchdog logic monitors
- 16-bit timer/counters
- Up to 52 general-purpose high-speed I/O lines
- On-chip ROM to 16 kbytes
- On-chip RAM to 512 bytes
- On-chip peripherals
  - — DMA
  - — HDLC
  - — Timers
  - — Input-capture registers
  - — A/D converter
  - — UART
  - — User-programmable memory
  - — High speed SRAM
- M²CMOS fabrication
- MICROWIRE/PLUS™ serial interface
- ROMless versions available
- Wide operating voltage range: +4.5V to +5.5V
- Military temp range available (−55°C to +125°C)
- MIL-STD-883C versions available
- 68-pin PGA, PLCC, LDCC packages and 84-pin Tape-Pak®

National's High Performance Controller (HPC) family is not only the world's first 16-bit CMOS microcontroller family, but also the world's fastest.

Currently operating at a clock rate of 30 MHz, the HPC is fabricated in scalable M²CMOS™, allowing die-shrinks ultimately, to submicron levels. Meaning the HPC will be operating at much higher frequencies in the future.

The HPC is designed for high-performance applications. With its 67 ns instruction cycle and its 16 x 16-bit multiply and 32 x 16-bit divide, the HPC is appropriate for compute-intensive environments that used to be the sole domain of the microprocessor.

The Powerful HPC Core

The HPC is an "application-specific" microcontroller.

Based on a common, high-performance CPU "core", each HPC family member can be "customized" to meet the exact needs of a particular application.

The core, based on a microprocessor-like von Neumann architecture, contains seven key functional elements:

1. Arithmetic Logic Unit (ALU)
2. 6 working registers
3. 8 interrupts
4. 3 timers
5. Control logic
6. Watchdog circuitry
7. MICROWIRE/PLUS interface

The internal data paths, registers, timers, and ALU are all 16 bits wide.

So the HPC can directly address up to 64 kbytes of "external" memory.

The external data bus, however, is configurable as 8 or 16 bits, allowing it to efficiently interface with a variety of peripheral devices.

Efficient Instruction Set

The HPC family achieves much of its performance through its unique, highly optimized instruction set. Unlike the instruction set of a typical microprocessor, the HPC instruction set is designed for maximum code efficiency. Because ROM-space is necessarily limited on a single-chip solution, programs must be compact and economical.

The HPC instruction set supports nine addressing modes, like a high-performance 16-bit microprocessor. And each instruction in the set is designed to execute a number of individual functions, so the same operations can be executed with tighter code.

As a result, the typical HPC instruction cycle is only 67 ns at 30 MHz. And the typical HPC 16-bit multiply or divide takes less than 4 μs.

To achieve the same level of performance in other 16-bit and high-end 8-bit microcontrollers, as indicated by recent benchmark studies, would require up to *two times the memory space* as the HPC.

Low Power Operation

The HPC uses power as efficiently as it uses memory space.

The HPC draws only 47 mA of current at 20 MHz. And its even less at lower clock rates.

In addition, the HPC has two software-selectable power-down modes:

1. IDLE, which stops all operations except for the oscillator and one timer, thereby maintaining all RAM, registers, and I/O in a static state.

2. HALT, which stops all operations including the oscillator and timers, but holds RAM, registers, and I/O stable.

High Level Language Support

A C compiler is already available for software development on standard platforms: the IBM PC running DOS or UNIX® or the DEC™ VAX™ running VMS™ or UNIX.

With powerful tools such as these, the HPC can be quickly and efficiently programmed for any high-performance application.

HPC Family of Microcontrollers

Commercial Temp Version 0°C to +70°C	Industrial Temp Version −40°C to +85°C	Military Temp Version −55°C to +125°C	Memory		Features						
			ROM (Bytes)	RAM (Bytes)	I/O		Interrupt	Stack	Timer Base Counters	Size (Pins)	Other*
					I/O Pins	Serial I/O					
HPC46003	HPC36003	HPC16003	ROMless	256	52	YES	8 Sources	In RAM	8	68	4 ICR's
HPC46004	HPC36004	HPC16004	ROMless	512	52	YES	8 Sources	In RAM	8	68	4 ICR's
HPC46064	HPC36064	HPC16064	16.0k	512	52	YES	8 Sources	In RAM	8		4 ICR's
HPC46083	HPC36083	HPC16083	8.0k	256	52	YES	8 Sources	In RAM	8	68	4 ICR's
HPC46104	HPC36104	HPC16104	ROMless	512	52	YES	8 Sources	In RAM	8		4 ICR's & 8 CH A/D
HPC46164	HPC36164	HPC16164	16.0k	512	52	YES	8 Sources	In RAM	8	68	4 ICR's & 8 CH A/D
HPC46400	HPC36400	HPC16400	N/A	256	56	YES	8 Sources	In RAM	4	68	HDLC & DMA

*ICR = Input Capture Registers

HDLC = High-Level Data Link Control

PEARL = Port Expanded and Recreation Logic

National Semiconductor

PRELIMINARY

HPC16083/HPC26083/HPC36083/HPC46083/ HPC16003/HPC26003/HPC36003/HPC46003 High-Performance microControllers

General Description

The HPC16083 and HPC16003 are members of the HPC™ family of High Performance microControllers. Each member of the family has the same core CPU with a unique memory and I/O configuration to suit specific applications. The HPC16083 has 8k bytes of on-chip ROM. The HPC16003 has no on-chip ROM and is intended for use with external direct memory. Each part is fabricated in National's advanced microCMOS technology. This process combined with an advanced architecture provides fast, flexible I/O control, efficient data manipulation, and high speed computation.

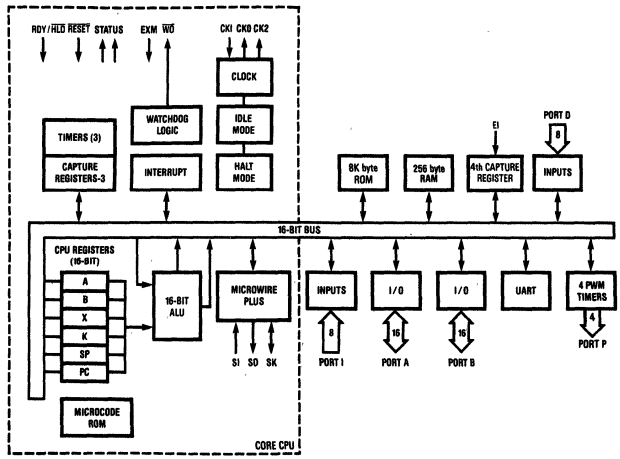
The HPC devices are complete microcomputers on a single chip. All system timing, internal logic, ROM, RAM, and I/O are provided on the chip to produce a cost effective solution for high performance applications. On-chip functions such as UART, up to eight 16-bit timers with 4 input capture registers, vectored interrupts, WATCHDOG™ logic and MICROWIRE/PLUS™ provide a high level of system integration. The ability to address up to 64k bytes of external memory enables the HPC to be used in powerful applications typically performed by microprocessors and expensive peripheral chips. The term "HPC16083" is used throughout this datasheet to refer to the HPC16083 and HPC16003 devices unless otherwise specified.

The microCMOS process results in very low current drain and enables the user to select the optimum speed/power product for his system. The IDLE and HALT modes provide further current savings. The HPC is available in 68-pin PLCC, LCC, LDCC, PGA and 84-Pin TapePak® packages.

Features

- HPC family—core features:
  - 16-bit architecture, both byte and word
  - 16-bit data bus, ALU, and registers
  - 64k bytes of external direct memory addressing
  - FAST—200 ns for fastest instruction when using 20.0 MHz clock, 134 ns at 30 MHz
  - High code efficiency—most instructions are single byte
  - 16 x 16 multiply and 32 x 16 divide
  - Eight vectored interrupt sources
  - Four 16-bit timer/counters with 4 synchronous outputs and WATCHDOG logic
  - MICROWIRE/PLUS serial I/O interface
  - CMOS—very low power with two power save modes: IDLE and HALT
- UART—full duplex, programmable baud rate
- Four additional 16-bit timer/counters with pulse width modulated outputs
- Four input capture registers
- 52 general purpose I/O lines (memory mapped)
- 8k bytes of ROM, 256 bytes of RAM on chip
- ROMless version available (HPC16003)
- Commercial (0°C to +70°C), industrial (−40°C to +85°C), automotive (−40°C to +105°C) and military (−55°C to +125°C) temperature ranges

Block Diagram (HPC16083 with 8k ROM shown)



TL/DD/8801-1

8

National Semiconductor

PRELIMINARY

HPC16164/26164/36164/46164
HPC16104/26104/36104/46104
HPC16064/26064/36064/46064
HPC16004/26004/36004/46004
High-Performance microControllers with A/D

General Description

The HPC16164, HPC16104, HPC16064 and HPC16004 are members of the HPC™ family of High Performance microControllers. Each member of the family has the same core CPU with a unique memory and I/O configuration to suit specific applications. The HPC16164 and HPC16104 have 16k bytes of on-chip ROM. The HPC16104 and HPC16104 have no on-chip ROM and is intended for use with external memory. Each part is fabricated in National's advanced microCMOS technology. This process combined with an advanced architecture provides fast, flexible I/O control, efficient data manipulation, and high speed computation.

The HPC devices are complete microcomputers on a single chip. All system timing, internal logic, ROM, RAM, and I/O are provided on the chip to produce a cost effective solution for high performance applications. On-chip functions such as UART, up to eight 16-bit timers with 4 input capture registers, vectored interrupts, WATCHDOG logic and MICROWIRE/PLUS™ provide a high level of system integration. The ability to address up to 64k bytes of external memory enables the HPC to be used in powerful applications typically performed by microprocessors and expensive peripheral chips. The term "HPC16164" is used throughout this datasheet to refer to the HPC16164, HPC16104, HPC16064 and HPC16004 devices unless otherwise specified.

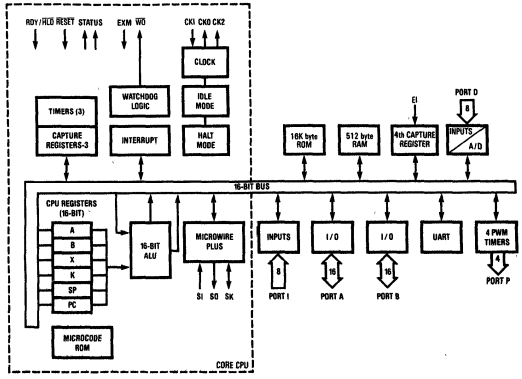
The HPC16164 and HPC16104 have, as an on-board peripheral, an 8-channel 8-bit Analog-to-Digital Converter. This A/D converter can operate in single-ended mode where the analog input voltage is applied across one of the eight input channels (D0–D7) and AGND. The A/D converter can also operate in differential mode where the analog input voltage is applied across two adjacent input channels. The A/D converter will convert up to eight channels in single-ended mode and up to four channel pairs in differential mode. The HPC16064 and HPC16004 do not have onboard A/D.

The microCMOS process results in very low current drain and enables the user to select the optimum speed/power product for his system. The IDLE and HALT modes provide further current savings. The HPC is available in 68-pin PLCC, LCC, LDCC, PGA and 84-pin TapePak® packages.

Features

- HPC family—core features:
  - 16-bit architecture, both byte and word
  - 16-bit data bus, ALU, and registers
  - 64k bytes of external direct memory addressing
  - FAST—200 ns for fastest instruction when using 20.0 MHz clock
  - High code efficiency—most instructions are single byte
  - 16 x 16 multiply and 32 x 16 divide
  - Eight vectored interrupt sources
  - Four 16-bit timer/counters with 4 synchronous outputs and WATCHDOG logic
  - MICROWIRE/PLUS serial I/O interface
  - CMOS—very low power with two power save modes: IDLE and HALT
- A/D—8-channel 8-bit analog-to-digital converter with conversion time minimum 6.6 μs for single conversion
- A/D—supports conversions in "quiet mode"

Block Diagram (HPC16164 with 16k ROM shown)



TL/DD/9682–1

Section 9

Disk Interface Design Guide

Section 9 Contents

Disk Interface Design Guide and Users Manual

National Semiconductor
Application Note 413

CHAPTER 1 DISK DRIVE TECHNOLOGY—OVERVIEW

1.0 INTRODUCTION—WINCHESTER DRIVES

From the start, digital computers have required some form of data storage as an adjunct to their relatively sparse main-storage facilities. Some of the early forms of storage were punched cards, paper tape and the magnetic tape storage. This was the principal storage medium, until faster-transfer, higher-capacity media became available and a direct link was established between the computer's main memory and the mass storage device. This link was the rotating memories, commonly referred to as disks.

Disk technology started a quarter-century ago, with the introduction of a large cumbersome fixed disk unit with 50 rotating surfaces 24″ in diameter, a single read/write head assembly, 600 ms seek time and a modest capacity of 5 megabytes. Half a decade later, capacities had increased by tenfold. Multiple head assemblies, one for each surface, introduced the concept of a "cylinder", providing simultaneous access to multiple tracks, one above the other, with a single head movement. Packing densities increased, resulting in increased storage capacity up to 100 megabytes. Head designs became more sophisticated; bits per inch increased by an order of 10; tracks per inch doubled.

Contamination-free Winchester technology was introduced by IBM in 1973. In addition to a controlled environment that eliminated dust collection on the disk surface, Winchester innovations included lightly loaded heads, an oriented iron-oxide coating to support higher flux reversal densities, and a silicone or wax coating that permitted heads to slide directly on the surface during "takeoff" and "landing"—eliminating the need for complex head loading mechanisms. The Winchester technology offers a number of advantages; device reliability, data integrity, faster transfer rates and a broader range of capacities. By the early 80's, fixed disk 14″ Winchester capacities were approaching 600 megabytes. Drives with capacities of 3 to 6 gigabytes are now on the immediate horizon. Winchester innovations also served as the springboard for miniaturized rigid disk systems. First came compact single or double-platter, non-removeable 14″ units with capacities down to 10 megabytes. Then around 1975 the 8″ Winchesters appeared, closely followed by the 5¼″ units, suitable for smaller desktop computers. Today the market boasts of a continuous spectrum of small to medium Winchester sizes: 3½, 5¼, 8, 10½ and 14 inches. Capacities begin at 5 Mbytes to 900 Mbytes.

The disk drive consists of one or more platters and heads, and the control mechanism with its associated electronics. The disk is essentially a platter made of aluminum or other base material, coated with iron-oxide or other magnetizable material. Each side of the disk consists of a number of thin annular regions called *tracks*. Each track is divided into blocks referred to as *sectors*. Data and other identification information is stored in the sectors. There are two types of sectoring: hard sectored discs and soft sectored discs. The hard sectored discs have sectors demarcated by the manufacturer and are identified by a sector pulse at the start of each sector while the soft sectored discs have only an index pulse signifying the start of a track.

The more recent hard disk drives have a number of platters on the same spindle, with one head per surface. In such cases similar track position on each platter constitutes a cylinder, e.g. cylinder 0 is the cylinder corresponding to track 0 on both sides of all the platters. The reading or writing of data is accomplished by the read/write head. This head is positioned on the required track by the drives positioning control system. This process is commonly referred to as seeking and is usually less than 17 ms. The quantity of data that can be stored on a disk depends on how much of its surface area is magnetized for the storage of a bit. On a typical low cost Winchester disk track densities are around 400 tracks per inch, while flux densities range around 9000 flux transitions per inch (implying recording densities of 9000 bits per inch). The rate at which data is written on the disk or read from it is termed as *transfer rate* and ranges from 5 Mbits/sec to 24 Mbits/sec and greater. The speed at which a particular sector is found for the writing or reading of data is gauged by the access time. First the head must be positioned over the proper track referred to as seek time. Then the proper sector of the track must come under the head which is referred to as the latency time. These are some of the common terms associated with the disk drive system.

The disk selection process is a function of several factors like storage capacity, upward mobility, transfer rate, etc. Data capacity is, perhaps, the most difficult decision to make in the selection process. All questions, present and future, must be considered in the context of the application. A fail, safe option, of course, would be to select a drive design with enough potential capacity to meet any future storage requirements. Disk technology has been striving to increase capacity, with future increases taking the form of increased data densities. There is considerable room for growth. Better head and disk material techniques are being used to raise track densities. Higher track densities have resulted in replacing the head positioning stepper motors by solenoid type "voice coil" actuators with theoretically infinite track following resolution. Developments in disk technology can also influence the transfer rate. The transfer rate directly affects system throughput. It is the average transfer rate that counts, and again this is a function of the application.

If write/read accesses are scattered because of varied reasons, track-seeking and sector-searching delays will reduce the effective transfer rate to a fraction of the theoretical value determined from data density and rotational speed. A series of application-dependent cost-performance tradeoffs

must be individually evaluated. Higher rotational speeds reduce the latency time as the system waits for a desired sector to pass under the write/read heads. Multiple heads reduce both the number of head repositions and the distance that must be travelled. Lower cost stepper motor actuators are normally open loop—moving the heads from track to track at a constant, relatively slow rate. Voice coil actuators are more expensive but inherently faster, accelerating and decelerating in response to feedback signals from a closed loop servo system.

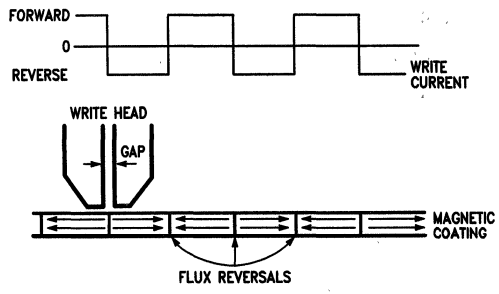
1.1 DISK STORAGE BASICS

Magnetic writing—the recording of data in a magnetic medium, is based on the principle that if a current flows in a coil of wire, it produces a magnetic field. The field is largely confined in a ring-shaped core of magnetic material, around which the wire is wound. A narrow slot is cut in the magnetic material and the field in the vicinity of the slot magnetizes the magnetic medium on the disk surface. Thus it creates alternating north-south magnets in the coated surface of the rotating disk. Thereby data is written, refer to *Figure 1.1(a).*

The head that writes the data can also be used to read it. This is done based on the principle of induction wherein a voltage is induced in an open circuit (like a loop of wire) by the presence of a changing magnetic field. In the case of a head positioned above a spinning magnetic disk on which data has been written, the magnetic fields emanate from the magnetized regions on the disk. During the time the head is over a single magnetized region, the field is more or less uniform. Hence no voltage develops across the coil that is part of the head. When a region passes under the head in which the magnetization of the medium reverses from one state to the other, i.e. a flux reversal, there is a rapid change in the field, developing a voltage pulse, refer to *Figure 1.1(b).* In this way the digital data are read as an analog signal, which can be readily converted back to digital form. The shape of this pulse and its ability to be recovered depends on various spacings. *Figure 1.1(c)* shows the spread of the coupling effect as a function of the width of the read-head gap and, equally important, the distance from the gap. The latter is, in turn, a function of both the head-surface separation and the depth of the flux reversal within the magnetic coating.

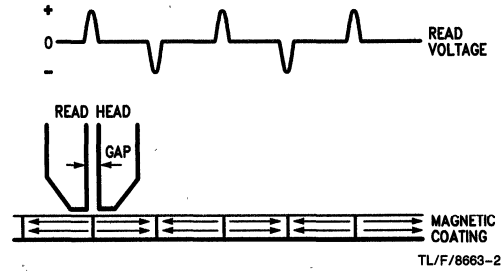
The quality of writing and reading of data depends of course on the magnetic properties of both the medium in which the data are stored and the head that writes and reads them. The common method of disk manufacture is to coat an aluminum disk with a slurry containing the gamma form of iron oxide. The iron atoms in the needle-like particles have their own minute magnetic fields and act like bar magnets with a dipole. The overall magnetization in any given region of the disk is the sum of the fields of these particles within it.

The core of most read/write heads is a ceramic consisting of spherical ferrite particles. The design of the head must conform to the design of the disk. In the case of the floppy disk(or flexible disket), which is a thin sheet of mylar plastic on which the gamma form of iron oxide is coated, the head makes contact with the surface, resulting in higher error rates and greater wear of the medium. In high performance disk drives, the magnetic medium is the coating on a rigid aluminum disk, and the head is kept from touching the medium by the so-called air-cushion effect. Consider a head that is nearly in contact with the surface of a hard disk spinning at 3600 revolutions per minute. If the length of the head along the direction of relative motion is two orders of magnitude longer than the separation between the head and the



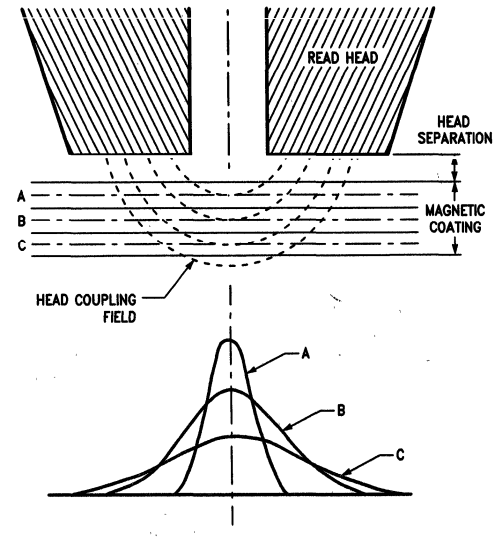
TL/F/8663–1

(a) Flux Reversals Produced by Write Current



TL/F/8663–2

(b) Read Pulses Generated by Flux Reversals



TL/F/8663–3

(c) Output as a Function of Saturation

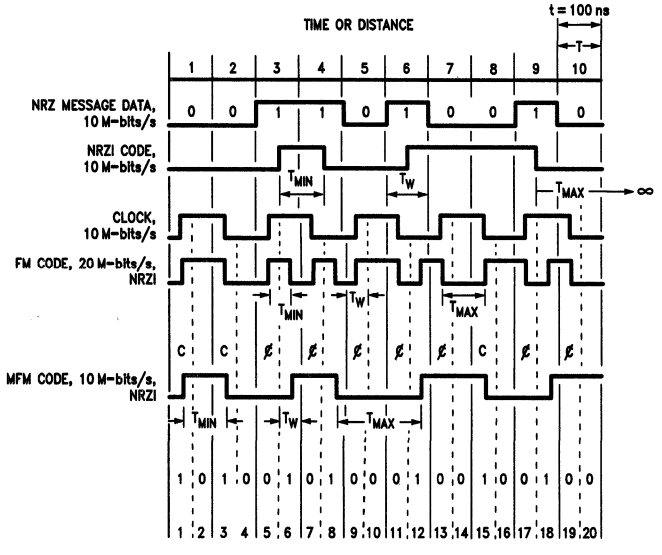
FIGURE 1.1 Flux Reversals as They Relate to "Writing To" and "Reading From" the Platter

medium, the flow of air between the head and the medium provides support for a head weighing up to several grams. Therefore because of the high cost of producing a hard disc along with its large storage capacity, it is used even if it has a bad track or sector while a floppy could be discarded. The bad sector is detected by using error checking and correction codes.

Optical Disk Technology

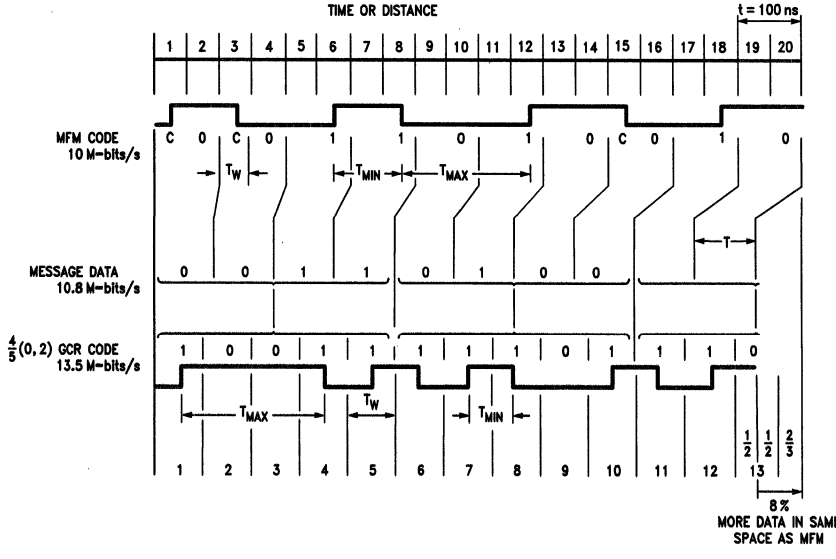
Disk drive improvements have resulted in faster data rates caused by increasing the density of the magnetic particles for greater storage. In the case of rotating magnetic memories, the strength of the signal depends on the strength of the medium's remnant magnetization. Recent advances in laser technology have resulted in digital optical disks becoming the last word in data storage and retrieval. Here, the laser beam itself provides the energy, hence the head is not in contact with the medium and it is protected, resulting in reduced errors and minimum medium wear. The advantages offered by optical disk technology are increased storage capacity, long data life, low cost per bit, noncontact read/write and easy physical mass replication. The optical disks initially developed could be written to only once. Read/write optical technology is being developed. Applications for optical disks are many and varied. On the interface level it is no different from Winchester drives and SCSI seems to be one of the most suitable of several possible choices. Another magnetic disk technology, "vertical" recording, is done with north-south magnetic poles perpendicular to the disk surface instead of end-to-end along the track.

FM & MFM Codes



TL/F/8663–4

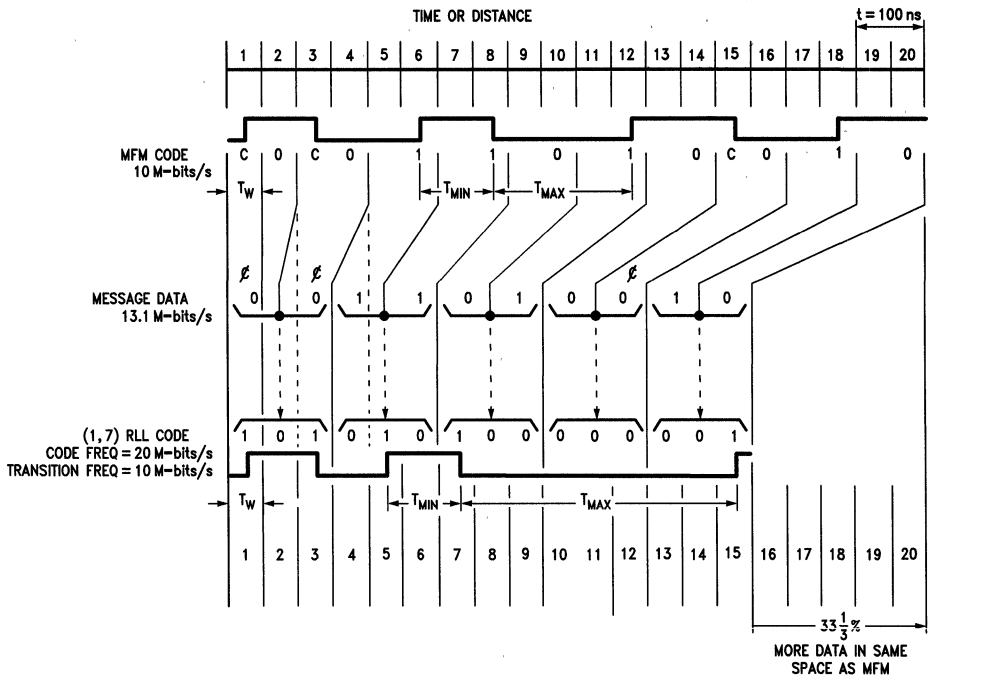
$\frac{4}{5}$ (0,2) GCR Code



TL/F/8663–5

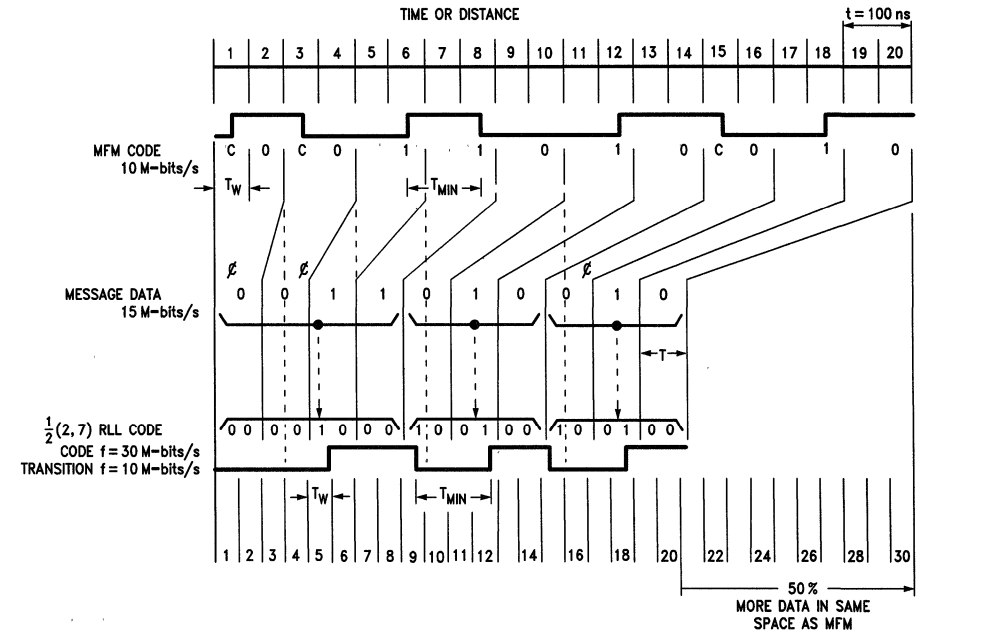
FIGURE 1.2 (a). FM, MFM and GCR Encoding

$\frac{2}{3}$ (1,7) RLL Code



TL/F/8663–6

$\frac{1}{2}$ (2,7) RLL Code



TL/F/8663–7

FIGURE 1.2 (b). RLL Encoding Schemes

1.2 DATA ENCODING/DECODING

Disk Data Encoding is the specific technique by which data is written to the disk, whereas decoding of data is necessary while reading from the disk. Data encoding removes the need of having clock information added to the track. Encoding also assists the controller in resynchronizing the data to the correct byte alignment, by allowing code violations for special data and address marks on the track. Considering the demand for ever-increasing data densities, it is understandable that the selection of a particular code is based largely on the efficiency with which flux reversals are converted into binary information, ZEROs and ONEs. In the ideal case, there should be the fewest flux reversals relative to the number of data bits they represent. Ideally, too, the code itself should provide its own "clock" for identifying the bit-cell intervals. Lacking this feature, a separate clock track may be required—or an extremely accurate oscillator must be provided to maintain the bit-cell divisions during intervals without flux reversals. The two requirements tend to be contradictory. An efficient code in terms of flux reversals will not be self-clocking. A self-clocking code will be wasteful of flux reversals. Nearly all of the widely used codes represent a compromise between these two extremes. *Figure 1.2(a)* shows details of FM, MFM and GCR encoding schemes, while *Figure 1.2(b)* shows details of some RLL encoding schemes. The commonly used encoding methods are discussed below in brief.

NRZ (NON-RETURN TO ZERO)

This is a telecommunication code and by far the most efficient. "Zero" refers to the transmission signal level. Instead of discrete pulses for each data bit, the signal rises or falls only when a ZERO bit is followed by a ONE bit or a ONE by a ZERO. NRZ coding reduces signal bandwidth by at least half. It also requires precise synchronization between source and destination in order to maintain bit cell divisions during the transmission of long strings of ZEROs or ONEs. NRZ could be used to transmit serial data to or from a magnetic recording device, disk or tape. But the extended intervals which can occur between flux reversals limit its usefulness as a recording technique.

NRZI (NRZ CHANGE ON ONEs)

This is the next most efficient code. It is widely used for tape recording and, to an increasing degree, disk recording. All ONEs are clocked, but special steps must be taken to compensate for the absence of flux reversals during strings of ZEROs. In the case of parallel-bit recording (tape), parity-bit ONEs serve as clock when all other bits in the byte are ZERO. In the case of serial-bit recording, data can be converted to RLL code (discussed below) which restricts the number of successive, unclocked ZEROs.

PE (PHASE ENCODED)

This is the least efficient of the coding methods but is completely self-clocking. The direction of a flux reversal at the middle of each cell indicates whether the bit is a ZERO or a ONE. Either one or two flux reversals occur, therefore, during each bit cell interval. The effect is to shift the "phase" of the signal by 180 degrees each time there is an NRZ type transition between ZEROs and ONEs.

FM (FREQUENCY MODULATION)

The FM method of encoding is equivalent to the PE technique and was the first choice for early disk-recording systems. It is generally only used for older floppy drives. Every bit cell interval is clocked by a flux reversal at the start of the cell. ONEs are marked by an additional flux reversal at the middle of the cell, doubling (modulating) the frequency of flux reversals for a series of ONEs compared to a series of ZEROs. A constant bit cell reference, provided by the clock bit, simplifies encoding and decoding with this scheme.

MFM (MODIFIED FREQUENCY MODULATION ENCODING)

With available head and media technology, MFM encoding is the most easily implemented encoding scheme and by far the most popular for floppy drives. It is used in the IBM System/34 and in available double-density LSI controller chips. MFM encoding doubles the data capacity over FM by eliminating the clock transitions (used in FM encoding) with data bits, refer to *Figure 1.2(b)*. Clock bits are still used, but are written only when data bits are not present in both the preceding and the current bit cell. As a result there is a maximum of one flux change per bit cell. Clock bits are written at the beginning of the bit cell, while data bits are written in the middle of the bit cell.

To decode data bits in MFM encoding, a data separator must generate a data window and a data window complement for a clock window. Because not every bit cell has a clock pulse, the data/clock windows cannot be timed from the clock pulse. Instead, the data separator must continuously analyze the bit position inside the windows so that the data/clock windows remain synchronous with the data/clock bits. Ideally, the clock transitions should appear at the center of the window. However, clock edges data bits can shift due to bit-shift effects. Present LSI controller chips can handle the drive interface, double density encoding function, and bit-shift pattern detection and compensation. National's DP8466 takes care of all these functions and needs only the data separator DP8465. Despite these constraints, disc controller design for MFM is simpler than that for either of the following encoding schemes.

M²FM RECORDING SCHEME (MODIFIED-MODIFIED FREQUENCY MODULATION ENCODING SCHEME)

Until recently, M²FM has been used as a double density encoding scheme, because the resolution of the medium and the read/write head was not adequate for the sizes of data window used in MFM. In M²FM, a clock is written only if no data or clock bit is present in the preceding bit cell, and no data bit occurs in the current cell. Because clock pulses are relatively isolated on the medium, the effect of bit shift on clock pulses is minimal. Therefore, a narrower clock window can be used to decode the clock pulse. The width of the data window can thus be increased by 20%, which allows more margin for shifted data bits. Today's ceramic-based read/write heads have much better resolution than those used in the past. This head design reduces the effects of bit shift, and makes the window margin provided by M²FM unnecessary. Additionally, M²FM is subject to a droop problem, which occurs in the read amplifier circuit when a low frequency pattern is read.

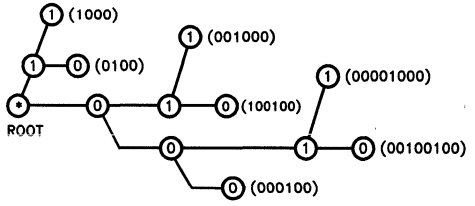
GCR (GROUP CODED RECORDING) ENCODING SCHEME

GCR encoding evolved from methods used in magnetic tape recorders. This method translates four data bits into a 5-bit code during a write. During a read, the 5-bit code is retranslated to four data bits; no clock bits are generated. Using data rates specified by drive manufacturers, this scheme is less dense than MFM. This method requires more circuits to code and decode, requiring necessary look-up tables, and costs more than either of the other two encoding schemes. For example; 1101 is encoded into a serial bit stream 01101 according to GCR encoding rules. To de-

code, a data window is generated around the expected position of each bit. The result is serial read data of 01101, which must be decoded to 1101 by lookup tables.

RLL (RUN LENGTH LIMITED) ENCODING SCHEMES

These recently popular encoding schemes are used in big 14" drives from IBM, CDC and DEC, and are starting to make an appearance in the small 5¼" drive market. The RLL encoding schemes have an excellent encoding efficiency, up to 50% higher than MFM. It is, however, considerably more complex to generate, requires a much better data separation unit to recover recorded data and is more suceptible to wider error bursts. The encoding rules for RLL depend on the RLL scheme chosen. The most common one is the 2,7 RLL code which refers to the maximum number of consecutive 0s, refer to *Figure 1.2(b)*. A standard encoding tree is defined and the data is encoded on the basis of those rules, as shown in *Figure 1.3*. The data bit stream is taken and the encoding tree is traversed, starting at the root, where the nodes traversed are the data bit stream in the sequence they arrive. On reaching the leaf of the tree, the code there is then the 2,7 RLL code for that data stream, e.g. if there is a data stream 100011010, then on traversing the tree, a data bit stream of 10 has a code of 0100, while the next bits 0011 are encoded as 00001000.



TL/F/8663–8

FIGURE 1.3. Encoding Tree—2,7 RLL Code

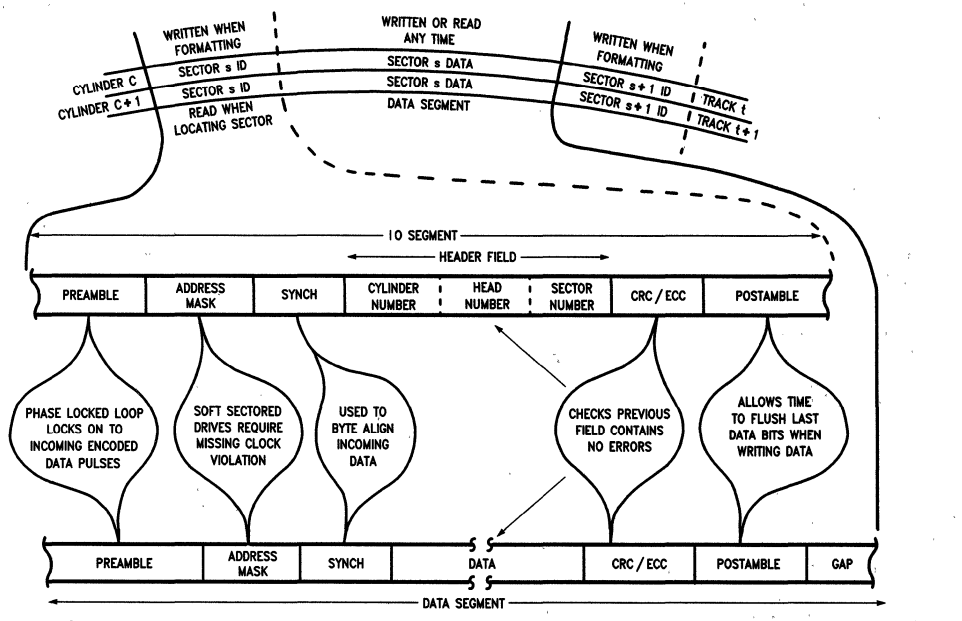
1.3 MEDIA FORMATTING

Media formatting provides the user with a reliable means of data retrieval using the magnetic recording surface of the track. There are many different formats but most of them are variations of the same basic structure. The formatting process is different for hard sectored and soft sectored disks. In hard sectored disks the sectors are defined by the manufacturers because the start of each sector is identified by the sector pulse generated by the drive. In soft sectored disks the drive issues only an Index pulse at the beginning of the track and the user can define all details of how information will be stored on the track, allowing more flexibility. *Figure 1.4* shows the basic format used. It consists of two segments—the ID segment and the data segment. The ID segment contains unique header information for the sector and the data segment contains the actual data. When the system requests a particular sector on a disk, the head must be positioned over the selected track, and the desired sector on that track must be found. This requires electronics to lock on to the data stream and then decode it. The beginning of a track is indicated by the Index pulse while the beginning of the sector is indicated by the sector pulse. This is followed by gap before the start of the sector on the track, which is referred to as the **Post Index/Sector Gap**. The explanation of the various fields are given below:

1.3.1 ID Segment

PREAMBLE OR PLL SYNCH FIELD

This is a field of repetitive clocked data bits usually 10 to 13 bytes long. The preamble normally will be all zeroes of NRZ data (encoded as 1010. . . in MFM). During the ID preamble, the signal Read Gate will go active, indicating that the incoming data pattern has to be locked on to.



TL/F/8663–9

FIGURE 1.4. Typical Sector Format Showing the Various Fields Within the ID and Data Segments

ADDRESS MARK FIELD

Address Mark (AM) is required on soft sectored drives to indicate the beginning of a sector, because this type of drive does not have a sector pulse at the start of each sector. This address mark byte contains a missing clock code violation, typically in MFM. The violation is detected by circuitry to indicate the start of a sector. The first decoded byte that does not contain all 0s after the preamble will be the address mark. The first 1 to be received is then used to byte align after the all zeroes preamble. Some formats have one ID address mark byte, while others have several.

ID SYNCH FIELD

For a hard sectored disk, byte alignment begins with the synch field that follows the preamble. The Synch bytes constitute a bit pattern that enables control circuitry to determine the byte boundaries of the incoming data, bit synchronization. Synch field usually follows the address mark on soft sectored drives and the AM is used for byte alignment also. Some formats use two synch fields: synch #1 and synch #2.

HEADER FIELD

The Header Field format varies between drive types, but typically has two cyclinder number bytes, a sector number byte, and a head number byte. It is generally 3 to 6 bytes long and one of the bytes may contain bits for bad sector or bad track recognition.

HEADER CRC/ECC FIELD

CRC (Cyclic Redundancy Checking) code or ECC (Error Checking and Correcting) code is appended to the header field. If CRC is used it consists of two bytes of the standard CRC-CCITT polynomial. The code detects errors in the header field. If ECC code is used, it is normally the same ECC polynomial that is used for the data field. This appendage is basically a protection field to make sure that the ID field contains valid information.

POSTAMBLE

This field may be used to give the disk controller time to interpret the data found in the ID field and to act upon it. It provides slack for write splicing that occurs between the ID and Data segment. A Write splice occurs when the read/write head starts writing the data field. A splice is created each time a sector's data segment is written to. The slight variations in the rotational speeds cause the first flux change to occur in different positions for each write operation. It also allows time in a write disk operation for the read/write circuitry to be switched from read to write mode. Finally it allows time for the PLL circuit to re-lock on to a fixed reference clock before it returns to synchronize to the preamble of the data field.

1.3.2 Data Segment

PREAMBLE FIELD

The Data Preamble field is necessary when reading a sector's data. It ensures that the PLL circuit locks on to the Data segment data rate. Initially, the ID segment and the data segment of every sector will be written when formatting the disk, but the Data segment will be written over later. Due to drive motor speed variations within the tolerance specified, the ID and Data segments will have slightly different data rates because they are written at different times. This implies that the PLL must adjust its frequency and phase in order to lock on to the data rate of the Data segment before the incoming preamble field has finished. Hence the need for a second preamble field in the sector.

DATA ADDRESS MARK FIELD AND DATA SYNCH FIELD

Following the Data Preamble will be the Data Address Mark for soft sectored drives, and Data synch, both similar to the ID segment equivalents.

DATA FIELD

The Data field is transferred to or from external memory. It is usually from 128 bytes to 64 kbytes per sector.

DATA CRC/ECC

A CRC/ECC appendage usually follows the Data field. CRC/ECC generating (when writing to the disk) and checking (when reading from the disk) are performed on the Data field. Errors may therefore be detected, and, depending on the type of error and if an ECC polynomial is used, they may also be corrected.

DATA POSTAMBLE FIELD

This has the same function as the ID Postamble field.

GAP FIELD

This is sometimes referred to as Gap 3, and is the final field of the sector. It allows slack between neighboring sectors. Without this gap, whenever a data segment is written to a sector, any reduction in drive motor speed at the instance of writing to the disk would cause an overlap of the data segment and the succeeding ID segment of the next sector. This field is only written when formatting the disk.

A final gap field is added from the end of the last sector until the INDEX pulse occurs and this gap is often termed Gap 4. It takes up the slack from the end of the last sector to the Index pulse.

1.4 THE DISK SYSTEM—DRIVE AND CONTROLLER

The Disk system essentially consists of two main paths: 1)the Disk Data Path, 2) the Disk Control Path, refer to *Figure 1.5*. The disk control path is responsible for controlling the disk drive with respect to positioning the head at the desired track and control the associated control signals (these are a function of the disk interface). The various disk interfaces are discussed later. The other component of the Disk System is the Disk Data Path which is responsible for data transfer from and to the disk.

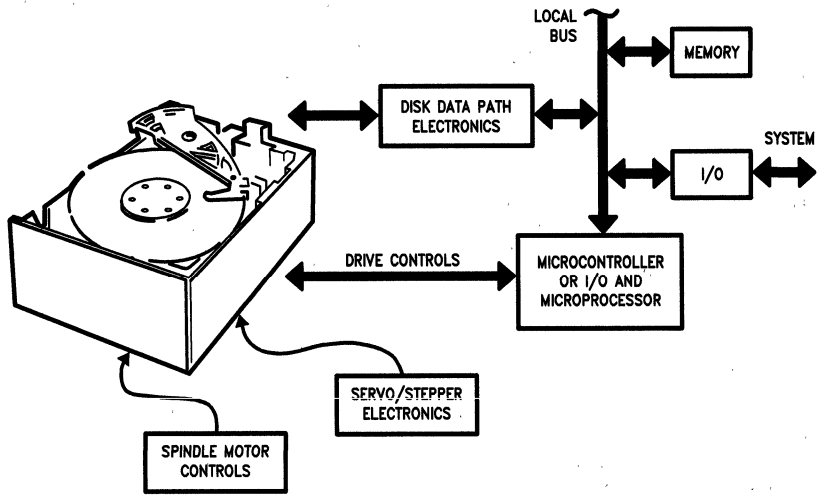
1.4.1 Reading Data from the Disk

Reading data from the disk to the system memory is a complex process and involves a number of operations enroute, as shown in *Figure 1.6(a)*. To initiate a read operation—a command is sent to the disk drive indicating the track and sector from which data is to be read. The seek operation moves the head to the desired track on the disk. Eventually the desired sector is identified by the header ID segment and the various fields are checked depending on the formatting rules used. The flux reversals are recorded by the head and are of the order of 500 microvolts. These pulses are then amplified by the read/write amplifier to about 10 mV. The signal from the read/write amplifier when reading a disk is therefore a series of pulses with alternating polarity. These pulses are passed through a Pulse Detector, like the DP8464. Electrically, these peaks correspond to flux reversals on the magnetic medium. The Disk Pulse Detector accurately replicates the time position of these peaks. The Disk Pulse Detector utilizes analog and digital circuitry to detect amplitude peaks of the signal received from the read/write amplifier associated with the heads of disk drives. A TTL compatible output is produced which on the positive leading edge indicates a signal peak.

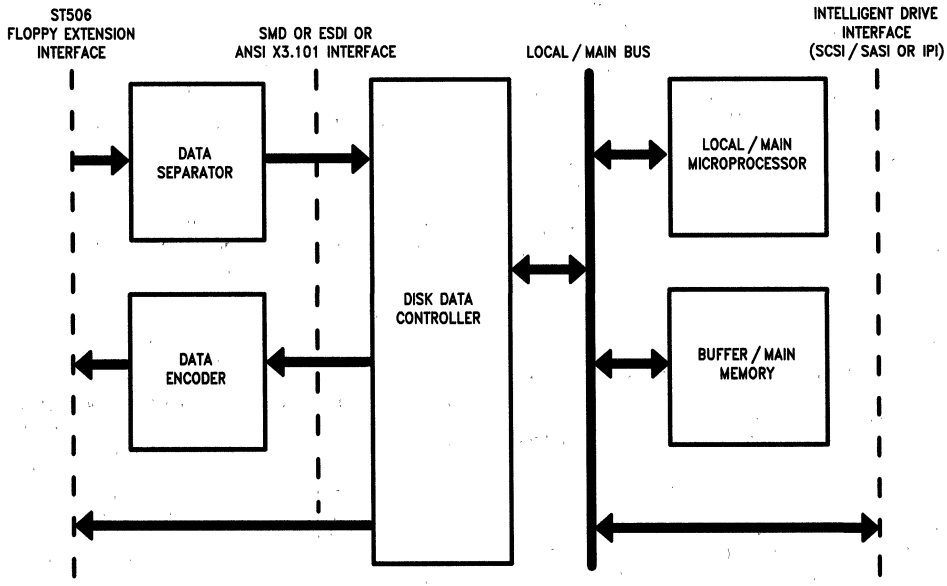
The raw data coming from the pulse detector consists of composite clock and data bits depending on the encoding scheme used. This encoded data has to be synchronized and decoded. These functions are performed by the Data Separator, like the DP8465. Due to bit shifting and distortion of the read pulses, the Pulse Detector issues non-synchronous pulses. For reliable decoding this jittery bit stream must be synchronized. The data separator has a Phase Locked Loop which attempts to lock on to the bit stream and synchronize it.

In hard sectored drives, the sector pulse indicates the beginning of the sector. Normally the preamble pattern does not begin immediately, because gap bytes from the preceding sector usually extend just beyond the sector pulse. Allowing two bytes to pass after the sector pulse helps ensure that the PLL will begin locking on to the preamble and will not be chasing non-symmetrical gap bits. For soft sectored drives, the controller normally will not wait for the Index pulse before it attempts lock-on. Chances are the head will not be over a preamble field and therefore there is no need to wait two bytes before attempting lock-on.

Disk Data Controller in a Disk System

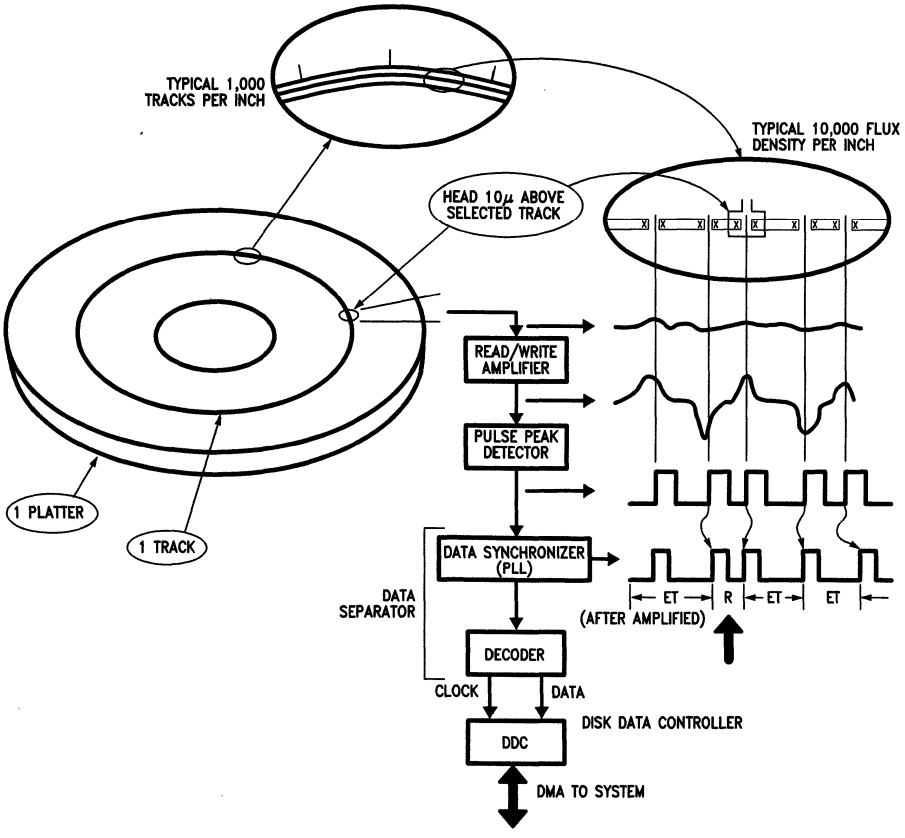


TL/F/8663–10



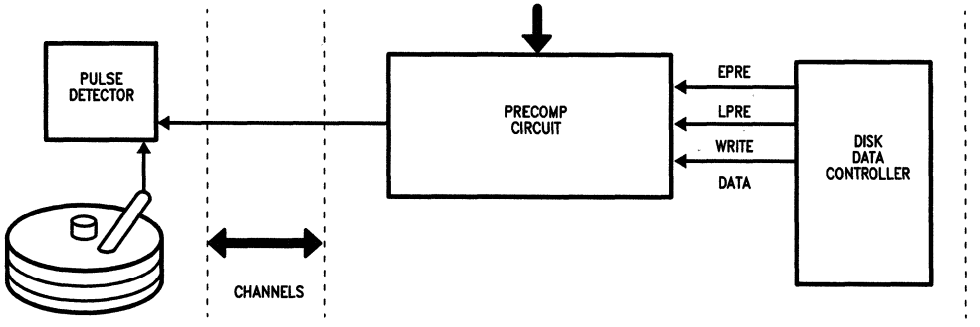
TL/F/8663–11

FIGURE 1.5. Disk System—Data and Control Paths



TL/F/8663–12

(a) Reading From Disk



TL/F/8663–13

(b) Writing to Disk (MFM Encoding)

FIGURE 1.6

Having locked-on to the bit stream the data synchronizer circuit must first determine the nominal position of clock and data bits and then generate an appropriate clock and data window that is centered around the bit positions. However there are many causes for bits to shift from the position where they are written. Erroneous data could be issued if bit jitter is beyond the tolerance computed. Therefore, special design considerations must be given to the type and resolution of the Data Separator used in reading data bits from the disk. The more accurately the bit position can be determined and the tighter the resolution of the data window, the lower is the soft error rate of the disk. Essentially the Data Separator's Phase Locked Loop locks on to the basic frequency of data bits read from the disk, and determines nominal bit positions for data and clock bits by sampling every bit (clock and data). It uses the phase relationship between a bit and its window to vary the position of the window. By sampling each bit, the phase-lock loop determines the phase error between a bit and the frequency being generated. To determine the nominal bit position around which to center the window, the data separator must track data bit frequency changes, yet ignore jitter. In this manner, even if an unpredictable bit shift occurs, the data separator can adjust the window's position to compensate for the change. Otherwise the shifted bit could be positioned outside the window. To remain within the typical error rate specified by the system, not more than 1 in 10^{10} bits can appear outside the window. With the present media technology, only a data separator based on an analog phase-lock loop technique can provide the necessary reliability.

Once the bit stream read from the disk has been synchronized and decoded to NRZ data, it is directly sent to the Disk Data Controller block, DDC, like the DP8466. In the DDC, the serial data is converted to parallel data (in terms of bytes), by the deserializer block. The main task is to recognize the byte boundaries accurately. In soft sectored drives this can be done by detecting a "missing clock" signal, which provides a fixed reference in the bit stream to set the byte boundary. Upon receipt of this signal the divide-by-eight circuit is set, to allow subsequent stages of the controller to acquire the bytes correctly. Hard sectored drives use a preset bit pattern in the synch field to determine byte alignment. Once the data is in parallel form it is stored in a temporary register in the controller. Transfer of data from this register to the system memory is achieved by DMA (Direct Memory Access) transfer. In this fashion data are read from the disk and transferred to the system.

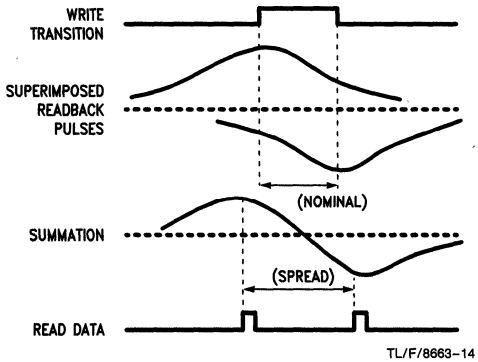
1.4.2 Writing Data to the Disk

The process of writing data to the disk is similar to the read operation in the reverse direction, with some changes. The write operation is initiated after the appropriate Seek command has been issued to the drive and the head is positioned over the desired track/sector. *Figure 1.6(b)* shows the basic write path blocks. Data is transferred from the system to the Controller using the DMA. The parallel data is converted to serial data by the serializer in the controller. This operation is conceptually easier to do, as the controller already has the right byte boundaries in the data and knows exactly where to insert the address mark. Most disk Controllers, like National's DP8466, provide either NRZ encoded data or MFM encoded data.

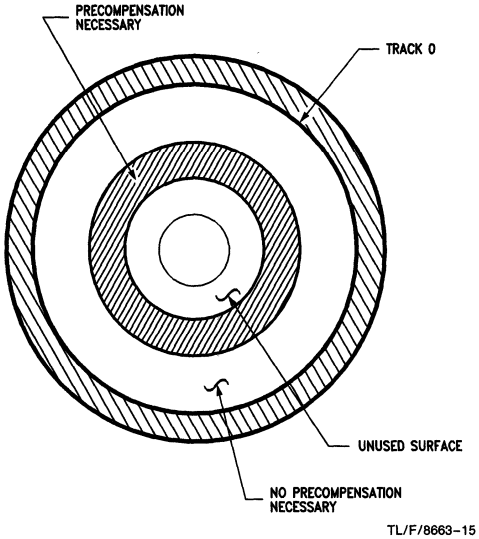
As mentioned in the previous section, predictable bit shift effects result from normal read/write head operation. Data are written when the read/write head generates a flux change in the media. In reading, a current is induced into the read/write head when a flux transition on the medium is encountered. The current change is not instantaneous, since it takes a finite time to build up to the peak and then to return to zero, refer to *Figure 1.7(a)*. If flux transitions are close together, the signal buildup after one flux transition declines, but it does not reach zero before a second transition begins. So when the flux changes are detected by the read/write head the peaks are shifted. A negative flux change, for example, may appear late because it has been added to the remnant of a positive transition. Narrower spacing between bits results in greater bit shift on the inner tracks. Hence compensation is needed on the inner tracks to minimize bit shift while no compensation is required on the outer tracks as bit shift is negligible, *Figure 1.7(b)*. Two methods currently being used are *precompensation* and *postcompensation*.

With precompensation, bits are deliberately shifted in the direction opposite to that of the expected shift. As data are being written, the controller detects bit patterns. From these bit patterns, the controller calculates which bit will shift in which direction. For example, a 4-bit pattern of 0110 on an inner track would cause the third bit to appear a few nanoseconds later than its nominal position. The controller chip, after detecting this late bit shift pattern, would generate an early signal, indicating that the third bit should be written earlier to make it appear closer to its nominal position when read. Conversely, if the third bit were going to appear early, a late signal would be generated so that the bit could be written later. How early or late the bit should be written is a function of its position in the data pattern, track position, and media, among other factors. Most Controllers provide signals to indicate what type of compensation is necessary. External circuitry is used to provide the actual delay as shown in *Figure 1.7(b)*.

The encoded precompensated data is then sent to the read/write amplifier where the stream of pulses is recorded on the disk as magnetic flux reversals. Postcompensation can be used when reading, usually as filter components around the pulse detector.



(a)



(b)

FIGURE 1.7. Bit Shifting

1.4.3 DMA (Direct Memory Access) Transfer/Data Buffering

The DMA block is responsible for the transfer of data between the host system and the disk controller. This is done because it is inefficient to dedicate a special communication channel to the task of transferring data between the disk controller and the system. The DMA system takes control of the control lines associated with a system's address and data buses, and exercises them in such a way as to transfer data in an appropriate direction from one device to another. It is also generally optimally efficient in using the available bus bandwidth whenever it is on the bus. The DMA capability is built-in for some disk controllers while in some an external one is required. National's DP8466 supports a single or dual channel DMA with capability of using an external DMA instead, if desired.

Data buffering is the temporary storage of some or all of the data to be transferred between the disk and the system memory. Any centrally intelligent system benefits from minimizing the bus occupancy. This is because the system has a lot of other tasks to perform, and if the bus is too heavily used, the system will miss performing some timely tasks. Therefore to prevent this, the data from the disk is transferred to a FIFO (First In First Out buffer). In a dual-DMA system, the local channel transfers the data from the FIFO to a local buffer memory while the remote DMA channel optimally transfers data from the local to the remote main memory over the system bus. This minimizes bus bandwidth use by the Disk I/O channel. The size of the FIFO is a function of different factors like: 1) the rate at which the system picks up the blocks of data, 2) the data rate from the disk and, 3) the burst transfer rate of the DMA. The way this is incorporated may differ in disk controllers. The buffer memory optimizes bus bandwidth. A system utilizing a single channel DMA would transfer data from the FIFO directly to the Host.

1.4.4 Error Detection/Error Correction

There are a number of factors which contribute to disk errors, viz. electrical noise, crosstalk, inadequately erased signals from previous recording, offtrack error in positioner, pin holes, inclusions, media thinning, and pattern induced errors. Of these, media defects are permanent errors. In general, ECC (error checking and correcting code), ensures reliable data storage and recovery. Generation of the ECC polynomial involves a detailed understanding of the mathematics of coding theory, and a cookbook approach to designing ECC logic. The basic idea behind ECC is the concept of irreducible polynomials. Take an irreducible polynomial (prime) and multiply it by the data pattern. Store the resulting remainder on the disk after all the data has been sent through the polynomial, *Figure 1.8(a)*. When reading the data back, divide the data coming off the disk into the polynomial. The reciprocal of the result should be equal to the check bytes on the disk, *Figure 1.8(b)*. If not, there is an error. The way error correction works is shown in *Figure 1.8(c)*. If there were no errors, then the sequence follows the straight path and the shift register contains all zeroes. If an error occurred, then at the point of occurrence the sequence vectors off, and at the end the shift register contains the pattern which caused the error. This helps in tracing back to the point of occurrence. The correction span is the number of contiguous bits in error which could be corrected. The probability of miscorrection is given by:

$$P_{mc} = (2^{(C-1)}) \times S/2^{A}$$

where C = correction span in bits

S = no. of bits in the sector

A = no. of bits in ECC appendage

Some codes have a higher miscorrection probability due to pattern sensitivities. A 48-bit ECC with an 11-bit correction span is recommended for 1,7 or 2,7 Run Length Limited encoded disks, while for MFM a 32-bit ECC with a 5-bit correction span results in low miscorrection probabilities. There are different types of codes:

FIRE CODES

Used in older systems and some current chips on the market. Fire codes have a high miscorrection probability related to double-burst errors (errors at two locations separated by more than the detection span) and are not recommended by National. Some examples of fire codes are:

32-bit FIRE CODE $(x^{21}+1)*(x^{11}+x^2+1)$

48-bit FIRE CODE $(x^{13}+1)*(x^{35}+x^{23}+x^8+x^2+1)$

56-bit FIRE CODE $(x^{22}+1)*(x^{11}+x^7+x^6+x+1)*$
$(x^{12}+x^{11}+\ldots+x^2+x+1)*$
$(x^{11}+x^9+x^7+x^6+x^5+x+1)$

COMPUTER GENERATED CODES

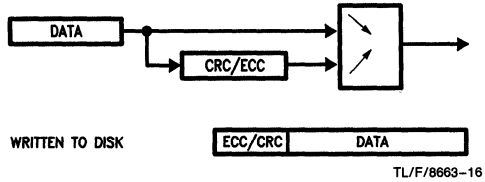
These have a very good reliability and are specifically chosen to guarantee not miscorrecting a specified worst case double burst error. The reliability can be calculated by the equation for miscorrection probability. National recommends the use of these codes for disk systems.

DOUBLE BURST REED SOLOMON CODES

Reed Solomon codes can handle longer bursts, multiple burst error (two burst error within a sector) correction capability and would be necessary for use with some optical media because of the high error rates. Typically RS codes for optical media are as long as a quarter of the sector. An example of the Reed Solomon code is given below.

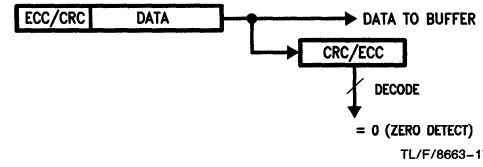
$$(x + a^5)*(x + a^6)*(x + a^7)*(x + a^8)*(x + a^9)$$

How Error Detection Works



TL/F/8663–16

(a) Writing to Disk



TL/F/8663–17

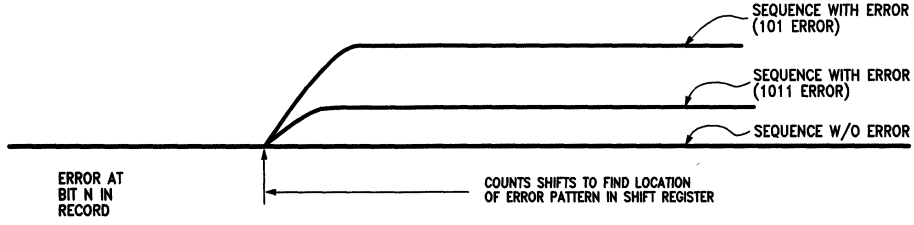
(b) Reading From Disk

$$\frac{\text{Data}}{\text{Polynomial}} = \text{Quotient (Discard)} + \frac{\text{Remainder}}{\text{(append as ECC)}}$$

$$\frac{\text{Data + Remainder}}{\text{Polynomial}} = \frac{\text{Data}}{\text{Polynomial}} + \frac{\text{Remainder}}{\text{Polynomial}}$$

$$= \frac{\text{Remainder}}{\text{Polynomial}} = \phi$$

If CRC Read = CRC Written



TL/F/8663–18

(c) How Correction Works

FIGURE 1.8. ECC/CRC

CYCLIC REDUNDANCY CODE

These can only detect errors and will not correct. They are generally used for header appendage and in floppy drives. The most widely used code is the CRC-CCITT code given below.

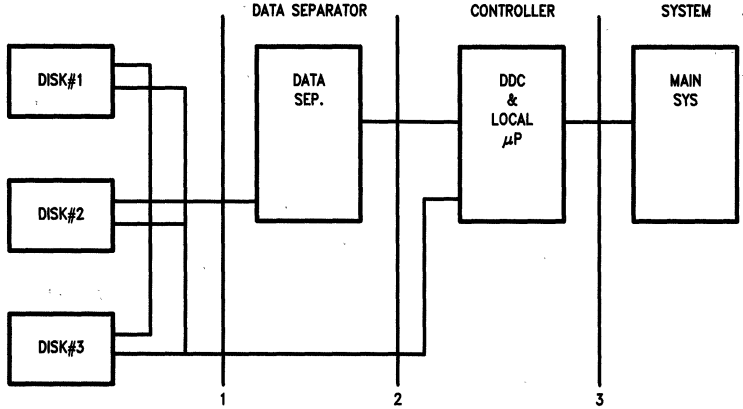
CRC-CCITT 16 bit $x^{16} + x^{12} + x^5 + 1$

Selecting the correct CRC/ECC is a function of the parameters being evaluated.

Detection Ability: The ability of the CRC/ECC to detect errors in the data transferred, measured as the number of bits affected and number of distinct bursts, the important measure being the guaranteed value.

Correction Ability: The ability of the ECC to restore erroneous data to its original state. Again, like the detection ability, this is measured as number of bits and number of bursts, the important measure being the guaranteed value.

Operating Environment: This involves factors like encoding scheme, data rate, data block size, technology on disk, product environment, and compatibility.



TL/F/8663–19

(a) CONCEPTUAL REPRESENTATION OF THE DISK SYSTEM WITH POTENTIAL INTERFACE POINTS. 1 = ST506/ST412; 2 = ESDI, SMD; 3 = SCSI, IPI

Interface	Year	Data Rate	Connectors	Drives	Status
SMD*	1975	≤15 Mb/s	60-Pin, 26-Pin	Hi Perf 8", 14"	Upgrading Now
SA1000	1978	4.3 Mb/s	34-Pin, 20-Pin	Low Cost 8"	Limited Future
ST506	1980	5 Mb/s	34-Pin, 20-Pin	Most 5¼", 3½"	Still Popular
ST412HP	1983	10 Mb/s	34-Pin, 20-Pin	Low Cost 8"	
ESDI*	1983	10–15 Mb/s	34-Pin, 20-Pin	Mid-Hi Perf 5¼"	New Standard
Future	1985–6	24 Mb/s	34-Pin, 20-Pin	Hi-Perf 5¼", 8"	

*Data separator on the drive.

(b) POPULAR HARD DISK DRIVE INTERFACES

Interface	Year	Data Rate	Data Bus	CTL Bus	System	Status
SASI	1981	≤1–2 Mb/s	8-Bit + P	9-Bit	Low-End	Superseded by SCSI
SCSI Asynchronous	1982	≤1–2 Mb/s	8-Bit + P	9-Bit	Low/Mid-End	ANSI Standard
SCSI Synchronous	1984	≤4 Mb/s	8-Bit + P	9-Bit	Mid-End	Recent ANSI Standard
IPI-3	1984	≤10 Mb/s	8-Bit + P 16-Bit + 2P	6-Bit	High-End	Almost ANSI Standard

(c) POPULAR INTELLIGENT DISK SYSTEM INTERFACES

FIGURE 1.9. Drive Interface Standards

1.5 THE DISK DRIVE CONTROL PATH

The disk drive control path essentially consists of the various control signals defined by the drive interface for drive control and data path control. The control path in a disk system has a number of potential interface points, *Figure 1.9(a)*. The popular hard disk drive interfaces which define the physical connections of the controller with the drive are given in *Figure 1.9(b)*. These are at the interface points 1 and 2. At these points data is still in the serial format. With the advent of sophisticated controllers many Intelligent disk system interfaces have come into being, *Figure 1.9(c)*. These essentially incorporate the complete controller on the drive and interface to the outside world, through an 8- or 16-bit standard bus, interface specific. The physical interface with the disk is usually one of the standard hard disk drive interfaces mentioned in *Figure 1.9(b)*.

INTERFACE STANDARDS

Interface standards are the definition of the connection between parts of the disk unit, controller, and system. Interfaces can be defined on several levels, viz.

- Electrical specification of signal levels.
- Timing relationships between signal lines.
- Physical specification of cabling, connectors etc.
- Functional specifications of tasks the standard performs.
- Command descriptor specification of the standard.

Some interface standards require only a subset of the above definition categories. For example, there is no necessity for a command descriptor segment to the ST506 defacto standard, as no provisions are made for command communication other than the simple control lines described in the functional specifications.

Standards are important as they allow numerous manufacturers to cater to the same market segment, thus creating healthy competition. For example, the ST506 interface is an industry standard simply because it is being used by a lot of drive manufacturers. The factors which affect the choice of the standard are: data rate, flexibility, popularity, performance, and cost.

1.5.1 Popular Hard Disk Drive Interfaces

There are a number of disk interface standards. It is important to realize the implications of the various contenders for the interface point. For example, if the data separator is placed on the drive, the cost of the drive increases, the cost of the controller board decreases, the speed of the interface can increase if desired, but the system has to cope with this increase. Some of the most commonly used standards (disk interfaces) are discussed briefly in the following sections.

FLOPPY DISK INTERFACE

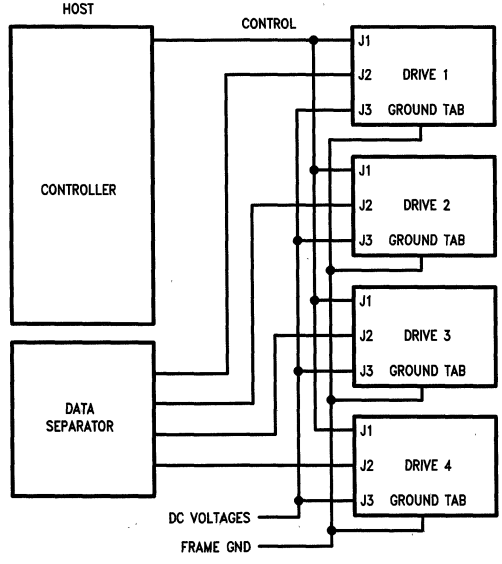
This is a relatively "dumb", single connector, serial data and control interface. There are two lines which carry the read/write data, and several control signals. This interface is positioned at point 1 in *Figure 1.9(a)*. The data rate for such interfaces is comparatively slow, around 100 to 500 kBits per second and the data capacity of floppy disks is not very large. The head is positioned by issuing step pulses to the drive. Read and write operations are initiated by asserting signals called Read Gate or Write Gate.

INTERFACE SIGNALS

Head load, Index, Sector (hard sectored drives only), Ready, Drive Select (usually 4), Step, Direction, Write Gate, Track 0, Write protect.

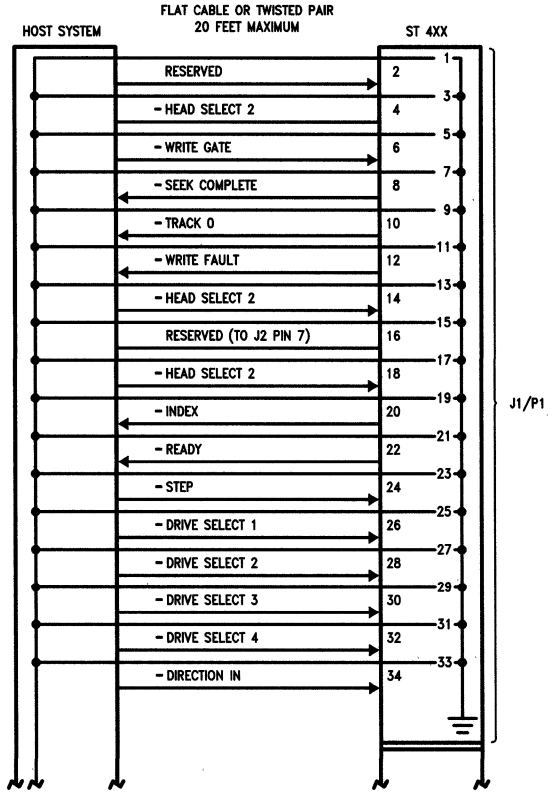
ST506/ST412 DISK INTERFACE STANDARD

This is also sometimes referred to as the floppy extension interface and is one of the most commonly used interface standards. The data rate is defined to be 5 Mbits per second, and the code is MFM. The interface is divided into two cables—a 34-pin control cable and a 20-pin data cable. The control cable allows for a daisy chain connection of up to four drives with only the last drive being terminated, *Figure 1.10*. The data cable must be attached in a radial configuration. This interface is at point 1 and, hence, the data separator is a part of the controller.



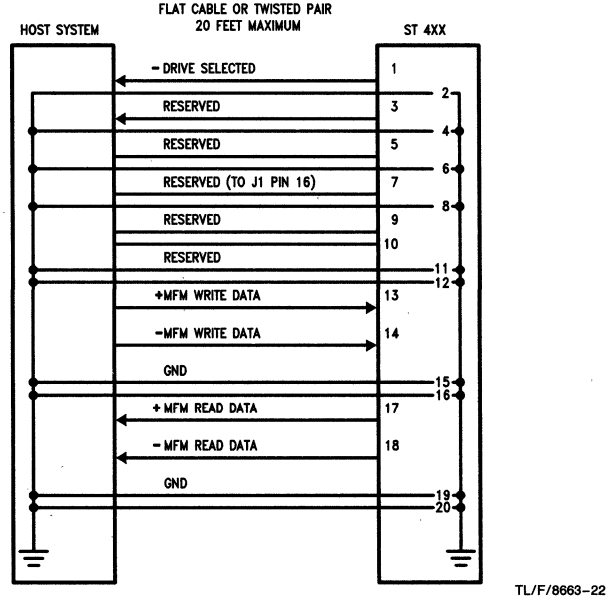
(a) Typical Connection, 4 Drive System

FIGURE 1.10. ST506/412 Configurations



TL/F/8663–21

(b) Control Signals Cable



TL/F/8663–22

(c) Data Signals Cable

FIGURE 1.10. ST506/412 Configurations (Continued)

Functional Operations

DRIVE SELECTION

Drive selection occurs when one of the DRIVE SELECT lines is activated. Only the selected drive will respond to the input signals, and only that drive's output signals are then gated to the controller interface.

TRACK ACCESSING

Read/Write head positioning is accomplished by:

a) Deactivating WRITE GATE line.

b) Activating the appropriate DRIVE SELECT line.

c) Being in the READY condition with SEEK COMPLETE true.

d) Selecting the appropriate direction.

e) Pulsing the STEP line.

Each step pulse will cause the head to move either one track in or one track out depending on the level of the direction line. A low level on the DIRECTION line will cause a seek inward toward the spindle, a high, outward toward track 0. Some drives have buffered seeks where the drive stores the pulses until the last one is received, then executes the seek as one continuous movement.

HEAD SELECTION

Any of the heads can be selected by placing the head's binary address on the Head Select lines.

READ OPERATION

Reading data from the disk is accomplished by:

a) Deactivating the WRITE GATE line.

b) Activating the appropriate DRIVE SELECT line.

c) Assuring the drive is READY.

d) Selecting the appropriate head.

WRITE OPERATION

Writing data onto the disk is accomplished by:

a) Activating the appropriate DRIVE SELECT line.

b) Assuring the drive is READY.

c) Selecting the proper head.

d) Insuring no WRITE FAULT conditions exist.

e) Activating WRITE GATE and placing data on WRITE DATA line.

Electrical Interface

The interface to the ST506/ST412 family can be separated into three categories, each of which is physically separated.

1. Control Signals.
2. Data Signals.
3. DC Power.

All control lines are single ended and digital in nature (open collector TTL) and either provide signals to the drive (input) or signals to the controller (output) via interface connection J1/P1. The data transfer signals are differential in nature and provide data either to (write) or from (read) the drive via J2/P2. *Figure 1.10* shows the connector pin assignments for this interface.

Since the data separator is on the controller, the ST506/ST412 drive must have a transfer rate of 5 M bit/sec. The bit density cannot be increased as the data rate and disc rotational speed are fixed. The only way to increase drive capacity is to increase the number of tracks, which does not allow large increases of capacity. Despite this limitation, it has a strong future as it moves into lower cost systems and smaller 3½" drives.

ST412 HP INTERFACE

This standard was designed to provide an upgrade path from the ST506 and is very similar. This interface is also at point 1. The main differences from the ST506 family are:

- One additional control line in the daisy chain . . . Recovery mode.
- Reduced write current is not part of the interface.
- The data rate is 10 Mbits/sec.
- The encoding scheme is not tightly specified, but suggested to be MFM.
- The maximum repetition rate of step pulses has been increased.

The major benefit of this interface is the higher data rate compared to ST506 drives, however, a much more careful design is needed to keep the bit error rate the same and for this reason may not be popular. Since the data separator is located on the controller, the data transfer rate must be exactly the 10 Mbits/sec rate and still be MFM encoded.

Recovery Mode

Recovery mode has been added in response to higher track density. It is asserted by the controller in response to bad data. In this mode the controller issues up to eight step pulses, and the drive steps through its own micropositioning algorithm. After each pulse, the controller tries to reread data and, if it fails again, after the eighth try it abandons the procedure. This drive interface emerged as higher data rate embellishment to the ST412.

ESDI (ENHANCED SMALL DEVICE INTERFACE)

This interface is at point 2 on *Figure 1.9*. This standard was a proposal by Maxtor Corporation, subsequently modified by an experienced working committee, and is finding growing acceptance largely because it is a sensible proposal. It has control and data cables like the ST506/412 interface but adds a driver and receiver on the data cable for the clock information, as shown in *Figure 1.11*. The implication is that the data separator resides on the drive, which means fewer design problems for drive users, and that certain status and command information is transmitted in serial, which means more control circuitry on both sides of the interface. The data rate is allowed to be several frequencies, dependent upon options, with the maximum rate probably reaching 24 Mbits/sec.

Features

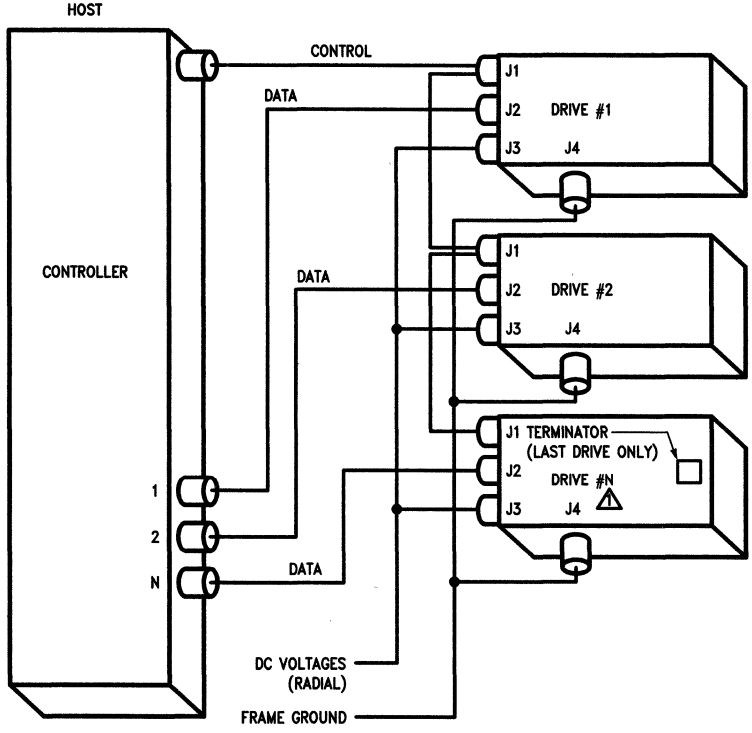
- Low cost, high performance interface suitable for smaller, high performance drives.
- Two protocols supported: serial and step mode.
- Supports up to 7 drives in the serial mode and 3 drives in the step mode.
- Maximum data rate of 24 Mbits/sec.
- Supports cable lengths of up to 3 meters.
- Serial mode of operation utilizes NRZ data transfer along with serial commands and serial configuration and status reporting across the command cable.

- Step mode implementation utilizes the same NRZ data transfer; however, the step and direction lines are used to cause actuator motion. Hence, with this mode configuration and status reporting are unavailable over the interface.

The ESDI interface puts the data separator on the drive and its output is NRZ data with a synchronous clock. This results in the data rate, and therefore the bit density, not being rigidly defined. The controller speed is governed by the synchronous clock coming from the drive, not from a data separator as in the ST412/ST506 interfaces. The drive is code independent as the data across the interface is always NRZ (or decoded) format. This enables the use of codes like RLL which put more data on the disk for the same bit density (flux reversals per inch). Moreover the use of NRZ encoding results in decreased errors due to electrical transients on the interface cable. This lowers practical bit error rates and allows the use of higher speeds.

Step Mode

The ESDI step mode is essentially similar to the step mode in the ST506/412 family of drives, except for the NRZ data transfer. Only two of the seventeen signals change function in the control cable between ESDI step and ST412HP. READ GATE being added is the important change which enables the data separator on the drive to the controller. The data cable is considerably different. Differential drivers and receivers are used for signals like Write Clock and Read Clock and a few single ended lines are added like Cartridge Changed (for tapes) and other lines like Seek Complete, Index etc. The step mode pulse timings are comparable to that of the ST412HP. This enables switching between the two interfaces under software control, in the controller design.

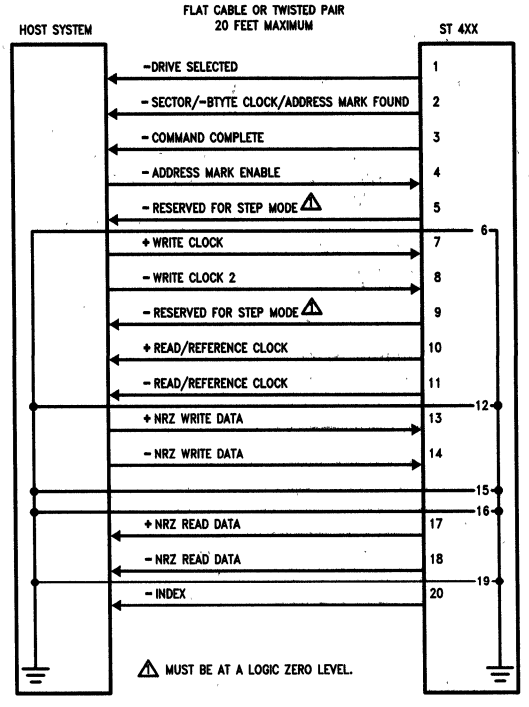


⚠ NOTE: IN STEP MODE, MAXIMUM NUMBER OF DRIVES = 3
IN SERIAL MODE, MAXIMUM NUMBER OF DRIVES = 7

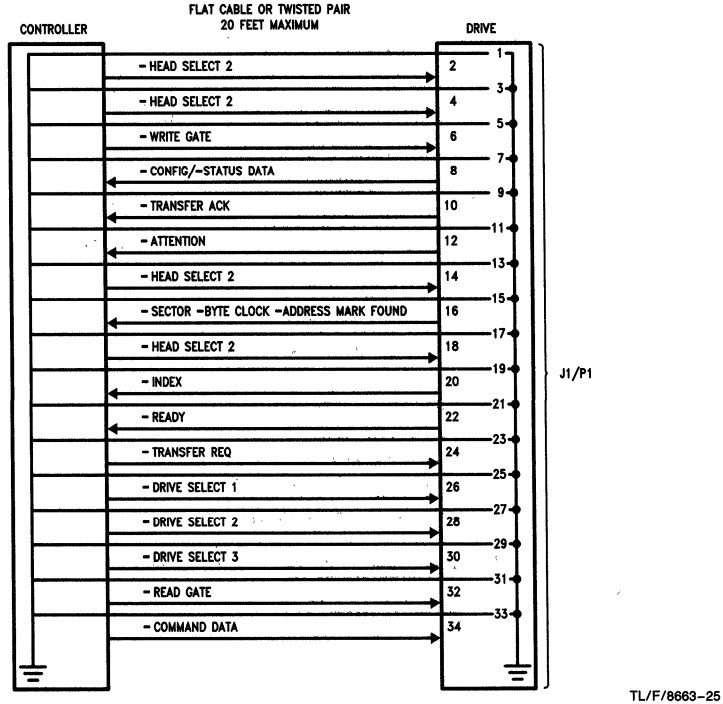
TL/F/8663–24

(a) Typical Connection, Multiple Drive System

FIGURE 1.11. ESDI (Enhanced Small Device Interface)



(b) Data Cable (J2/P2) Signals (Disk Implementation—Serial Mode)



(c) Control Cable (J1/P1) Signals (Disk Implementation—Serial Mode)

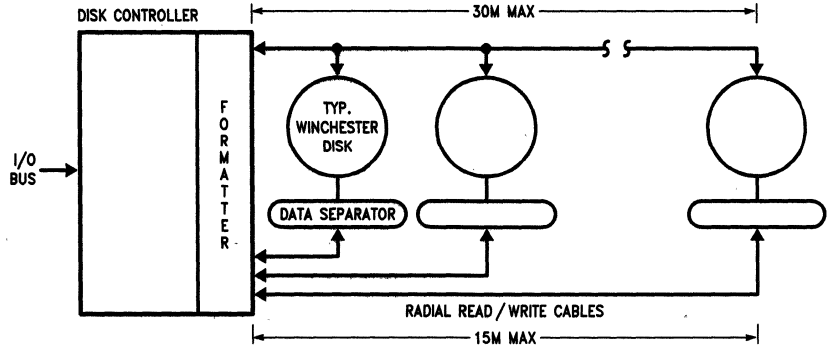
FIGURE 1.11. ESDI (Enhanced Small Device Interface) (Continued)

Serial Mode

Serial mode ESDI is a definite improvement over the interfaces discussed. As the name implies, communication from the controller to the drive takes place on the COMMAND DATA line of J1/P1 in conjunction with the handshake signals TRANSFER REQUEST and TRANSFER ACKNOWLEDGE. Communication from the drive to the controller takes place on the CONFIG-STATUS line of J1/P1 in conjunction with the handshake signals. Each bit of the 16-bit command or status word is handshaked across the interface. The hardware changes between EDSI serial and step modes, have several control lines redefined. The disk drive's microprocessor interprets commands like SEEK (seek to a cylinder), RECALIBRATE (seek to track 0), REQUEST STATUS and REQUEST CONFIGURATION, which provide the controller with standard status and configuration information of the drive like the number of heads, number of tracks, sectors per track, bytes per track, command data parity fault, write fault etc. Hence the controller can configure itself to the drive connected to it and can send the data to the host if desired. Thus ESDI serial mode offers big benefits and is rapidly gaining popularity in higher performance hard disk drives.

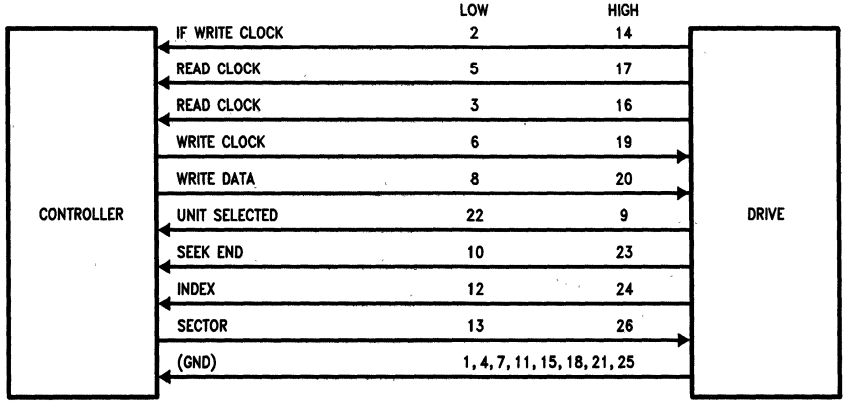
STORAGE MODULE (SMD) INTERFACE (ANS X3.91M 1982)

The Storage Module Interface was originated by Control Data Corporation around 1972. It has been extremely popular with 8"–14" drives. However, as it is expensive and hardware intensive and because of competition due to ESDI and SCSI, it is not very popular with 5¼" drives. *Figure 1.12* gives the data and control cable assignment.



TL/F/8663–26

(a) Typical Connection

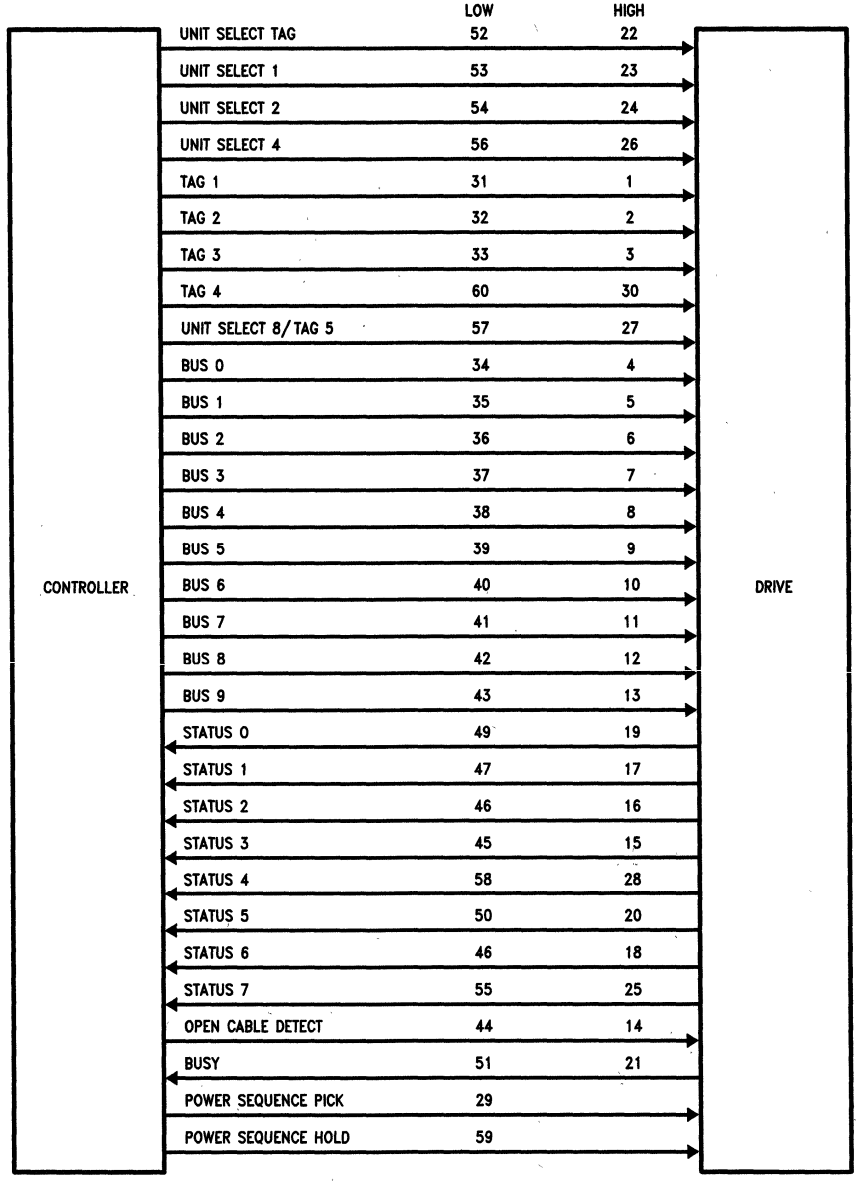


TL/F/8663–28

B = Cable

(b) Data Cable

FIGURE 1.12. Storage Module (SMD) Interface (ANS X3.91M 1982)



TL/F/8663–27

A = Cable

(c) Control Cable

Storage Module (SMD) Interface (ANS X3.91M 1982) (Continued)

Features

- Bit serial digital data transfer (Data Separator in drive)
- Relatively high transfer rate (9.67 Mbits/s is common in older 14″ drives and newer 8″ drives, new 10.5″ and 14″ drives are typically about 15 Mbits/s)
- Dominant de-facto standard for 14″ OEM disk drives; virtual basis of OEM disk controller industry. Widely used by minicomputer system manufacturers.
- Differential signals
- 23 required plus 8 optional control bus signals, 7 required plus 2 optional read/write cables
- Parallel control bus, but radial read/write cables, one per drive
- Incorporates error recovery facilities
- Includes power sequencing for multiple units
- Approved ANS X3.91 1982

1.5.2 Intelligent Disk System Interfaces

These are high level interfaces which result in the complete disk controller being situated on the drive and the interface to the host is through a special bus. Their chief advantage is nearly complete device transparency to system hardware and software, also lower system overhead for disk control and higher speeds. A well defined protocol is used for communication with the host system. Some of the popular Intelligent disk system interfaces are discussed below in brief.

SHUGART ASSOCIATES SYSTEM INTERFACE (SASI)

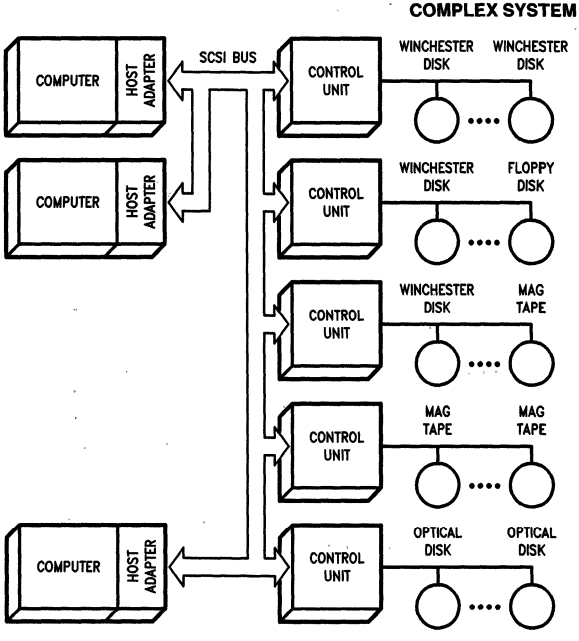
SASI was introduced by Shugart around 1980. The overall objective was to make it easier for computer systems to talk to disk drives. SASI defines the logical level and all lower interface levels, down to the 50-pin connectors and ribbon cable. Eighteen lines are used for signals, nine for data and nine for control. The data lines consist of a single octet with an odd parity bit. The control lines include a two-wire handshake and various lines to put the bus in different transfer modes or phases. The interface is always in one of the five phases:

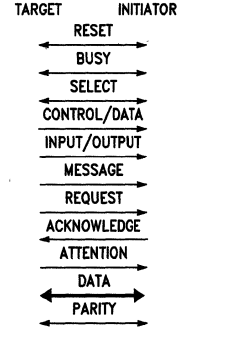
Bus Free, Arbitration, Selection, Reselection, Data

Eight devices are allowed but only one host or "initiator" is allowed. So a maximum SASI system will consist of an Initiator and seven target devices. All signals in the interface are open collector driven. The ANSI standard version of this interface is the SCSI (Small Computer System Interface).

SMALL COMPUTER SYSTEM INTERFACE (SCSI)

The Small Computer System Interface (SCSI) was formed from the SASI framework and ANSI has standardized it under X3T9.2. The interface consists of a single cable that is daisy-chained to other SCSI units. It will accommodate not only disk drives but also tapes, printers and other devices and is potentially a universal peripheral port for small systems. The SCSI system could potentially be a single initiator - single target system or a single initiator - multiple target system or a multiple initiator - multiple target system as shown in *Figure 1.13(a)*. The SCSI bus signals are shown in *Figure 1.13(b)*. The cable consists of transfer handshaking and status signals in addition to an 8-bit data bus. Information is exchanged on the bus via a set of higher level commands sent by the host.





Up to 8 SCSI DEVICES can be supported by the SCSI bus. They can be any combination of host CPUs and intelligent controllers.

(a) Multiple Target-Multiple Initiator **(b) SCSI Cable**

FIGURE 1.13. Small-Computer System Interface

Features

- Connects up to 8 computers and peripheral controllers
- Maximum rate up to 1.2 Mbyte/sec asynchronous, 4 Mbytes/sec synchronous: suitable for floppy disks, all 5.25" and 8" Winchester disks, medium performance 8" and 14" disks, and tape drives
- Relatively high level peripheral command set
- Single ended version: 50-conductor flat ribbon cable, up to 6 meters, 48 mA drivers
- Differential version: 50-conductor flat or twisted pair cable, up to 25 meters, EIA RS-485
- Distributed bus arbitration
- Includes command sets for common peripherals
- Products now widely available include: disk drives with integral controllers, SCSI to ST506, SMD, Floppy, SCSI to S-100, Multibus®, IBM PC™, VME™, Unibus™, TRS-80™, and Q-BUS™ Adapters, and VLSI bus protocol and disk controller chips

SCSI makes no hardware changes compared to the SASI but adds several features which are discussed below.

Arbitration

This allows multiple Initiators to talk to multiple targets in any order, to a maximum of eight nodes only.

Reselection

This allows a target to disconnect from the Bus while it is getting data and reconnect to the proper Initiator when it has found it. This results in efficient utilization of the bus because other nodes can use it during the relatively long seek time of the disk drive or search time of a tape drive.

Synchronous Mode Transfer

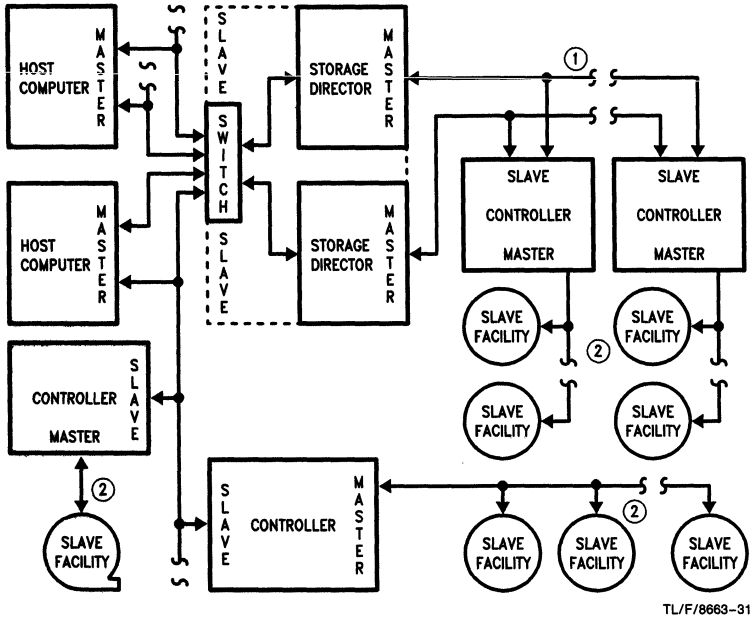
This speeds data transfer to a maximum of 4 Mbytes/sec (from an asynchronous maximum of approximately 1.2 Mbytes/sec).

Differential Transceivers

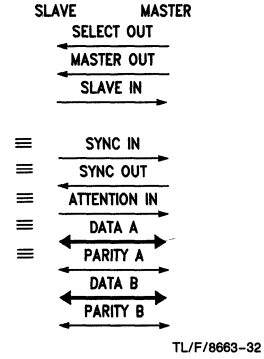
These boost the maximum length of the interface from 6m to 25m.

Extended Command Set

This set includes expanded large block addressing (from 2^{21} to 2^{32} blocks) and "Inquiry" type commands for self configuring controllers. It also handles tape drives, printers, processors, optical disks and read-only optical disks. There is also room for a "vendor unique" command set, i.e. commands unique to the device.



(a) IPI Configuration



(b) IPI Cable Signals

FIGURE 1.14. Intelligent Peripheral Interface

INTELLIGENT PERIPHERAL INTERFACE(IPI)

This is the ANSI standard X3T9.3 and is an additional peripheral bus, having higher performance than SCSI. *Figure 1.14(a)* shows the orientation of the IPI system and the IPI port signals are shown in *Figure 1.14(b)*.

Features

- Connects master to a maximum of 8 slaves
- Can be used at various levels in the system as shown
- Two 8-bit buses (in and out) for commands and status speed protocol and status presentation for fast path switching
- 8- or 16-bit parallel transfers
- 24 signals
- Several electrical options; fastest allows 10 Mbytes/sec through a 75 meter cable
- Offers both "intelligent" and "device level" command definitions, command and data handshaking
- 50-pin cable ground increases noise immunity

The IPI interface comprises four levels. Level 0 consists of cables, connectors and drivers/receivers. Level 1 consists of state machine and bus protocol. Features of Level 2 are device specific commands, timing critical, physical addressing, physical volumes, command parameters and bus control commands. Features of Level 3 are device generic commands, timing independent, buffered, command stacking, queing, limited specific commands, logical addressing and physical volumes. Messages are transmitted in packets.

IPI derives its higher performance from a faster handshake and a wider data bus. Two octets, each with a parity line, make up the data interface. Six control lines fill out the interface of 24 signals. There is one master allowed and up to eight slaves on a daisy-chained cable. This master to slave interface is a parallel one and hence IPI 3 could be used, (point 1) in *Figure 1.14(a)*. Each Slave can address up to 16 Facilities, like disk drives. The Slave-to-Facility interface may be IPI 2, (point 2) in *Figure 1.13(a)*, or a lower level interface such as ESDI. Data can be moved at 5 Mbytes/sec in asynchronous mode, 10 Mbytes/sec in synchronous mode. The interface supports various driver options with maximum cable lengths ranging from 5 meters to 125 meters.

1.5.3 Other Disk Interfaces

There are many other ANSI standardized interfaces which were the outcome of the interface standards discussed above. Some of these are disk level while some are intelligent interfaces. A brief discussion follows, also refer to *Figure 1.15*. Detailed descriptions can be found in the appropriate ANSI document.

FLEXIBLE DISK INTERFACE (ANS X3.80)

Features

- American National Standard Interface between flexible disk cartridge drives and their host controllers
- 50-wire flat ribbon cable (8″ disk) or 34-wire (5¼″ disk)
- Bit serial encoded FM or MFM data transfer from/to Read/Write electronics (data separator in controller)
- Modest data transfer rate (100 kbyte/s)
- Very widely used by the industry; based on a de facto standard. Supported by most 8″ and 5¼″ drives; also used by some, but not all micros (less than 4″) floppy drives.
- Seeks track by track, one step per pulse
- Single ended signals
- Change has been submitted to identify high density 5¼″ drives
- Approved ANS X3.80 - 1981

RIGID DISK INTERFACE (ANS X3.101—1983)

With the emergence of the 8″ rigid-disk drive, there was a strong industry push for a new interface standard with broader applicability than the SMD, which would allow for self-applicability than the SMD, which would allow for self-reconfiguring controllers, as different devices were attached. As a result the ANSI Rigid Disk Interface came into existence.

Features

- Optimized for relatively high performance small winchester disks
- Bit serial digital data transfer (data separator in drive)
- Relatively high transfer rate possible (up to 10 Mbits/s with low cost option, up to 16 Mbits/s with high performance option)
- 50 conductor ribbon cable
- Class A: 24 and 40 mA single-ended control bus signals, 20 mA differential serial data and clock lines
- Class B: 100 mA single-ended control bus signals, 40 mA differential serial data and clock lines
- 22 single-ended plus 4 differential signals
- Byte parallel command bus
- Relatively high level command set
- Approved ANS X3.101—1983

Peripheral Bus Interface

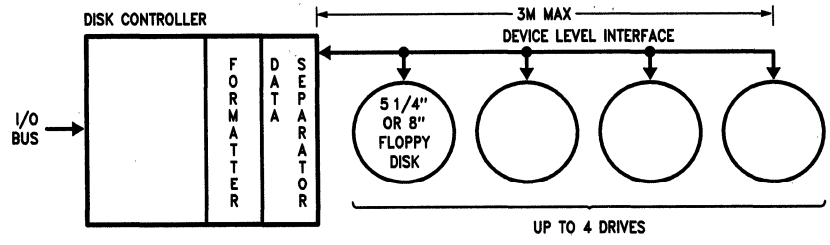
- Connects computer to peripheral different controllers
- Block transfer rather than word transfer orientation
- No provision for memory address on bus
- Longer distances than backplane
- Interface hides many device characteristics from software
- Peer to peer multi-master protocol may be called a "system bus"

Device Interface

- Specific to particular device type
- Between controller and device
- Often serial data transfer
- User device interchangeability not always certain
- Very widely used by industry

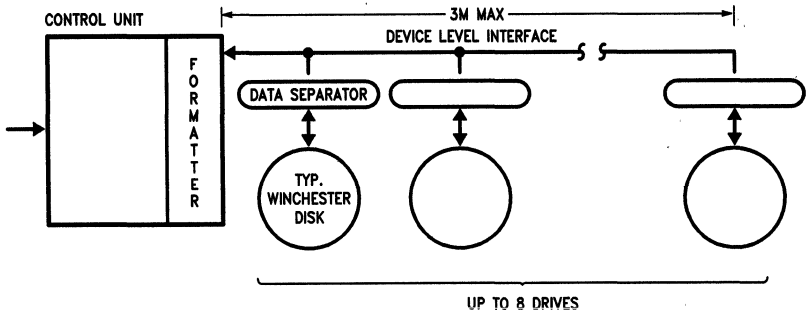
This interface has not gained acceptance and has been superseded by ESDI

Flexible Disk Interface (ANS X3.80—1981)

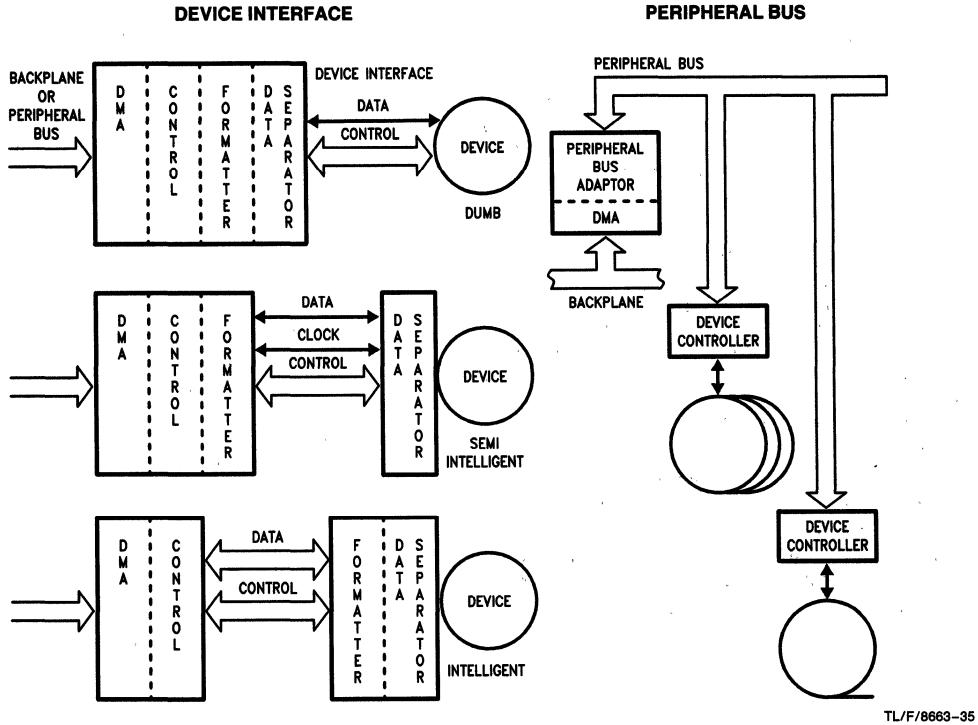


TL/F/8663–33

Rigid Disk Interface (ANS X3.101—1983)



TL/F/8663–34



TL/F/8663–35

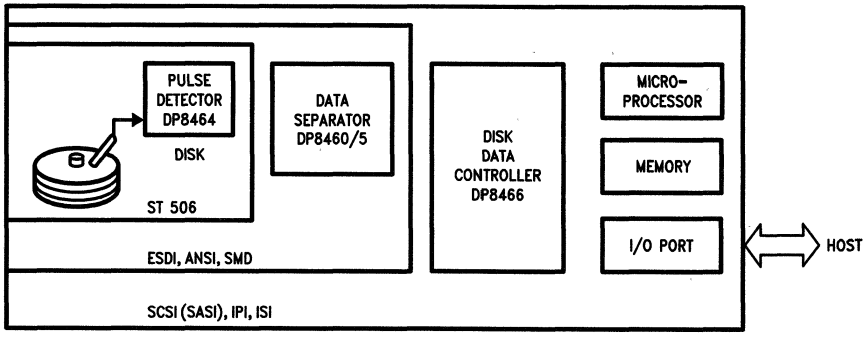
FIGURE 1.15. Other Drive Interfaces—Typical Connections

1.5.4 Universal Plug

Although the concept of a standard interface is appealing, the desire to gain industry recognition and lock in customers, coupled with valid technical improvements, will continue to spawn new interfaces at regular intervals. In systems where a maximum of one or two drives are required or where configuration flexibility justifies the higher cost of standard attachments, high-level interfaces will attach to a "universal" system plug. Here controllers will be integrated into the devices themselves, while in cost effective multiple drive systems, the connection of device level interfaces to system specific controllers is expected to continue.

1.6 ELEMENTS OF DISK CONTROLLING ELECTRONICS

Disk controller chips in the market today are complex VLSI chips and perform a multitude of functions. In fact they take care of most of the tasks besides the task of data separation. National's Disk Data Controller DP8466 is one such chip. It takes care of serialization, deserialization, data encoding, DMA transfer, error detection and correction, and pattern recognition to determine type of compensation required. The Disk Data Controller DP8466 is discussed in detail in the following sections. National's Data Separator chip DP8465, together with the Disk Pulse Detector DP8464, comprise the chip set for the disk controlling electronics. If RLL encoding is used, then National's DP8463 2,7 ENDEC could be integrated into the system. *Figure 1.16* shows the place of these chips in the disk data and control path. It also shows the separation lines of the components on the drive and controller for the various interface standards.



TL/F/8663–36

FIGURE 1.16. Typical Disk Controller System Configuration Showing the Interface Points with Respect to the Various Standards

CHAPTER 2 DP8464B HARD DISK PULSE DETECTOR

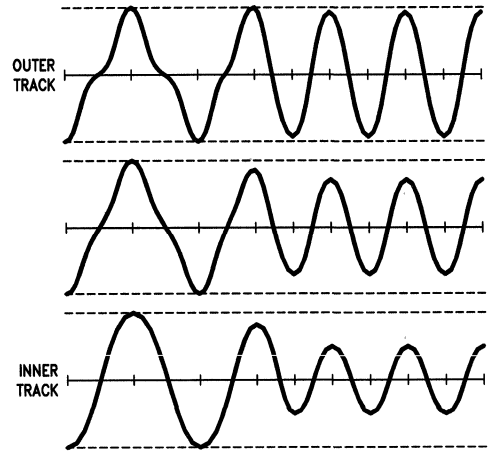
2.0 INTRODUCTION

The standard Winchester disk drive available today uses a magnetic film platter, a ferrite head and MFM coding. This combination produces relatively wide pulses off the disk which can be detected using a simple time-domain filter technique. However, as disk manufacturers strive for higher density and data rates, they must turn to new technologies such as plated media, thin film heads and run length limited codes (such as the 2,7 code). Unfortunately, these technologies produce more complex pulses off the disk which require more sophisticated pulse detection techniques. The DP8464B utilizes a separate time and gate channel which can detect the peaks in these complex waveforms.

2.1 BACKGROUND OF PULSE WAVEFORM DETECTION

Data on the disk is stored as a series of magnetic domains recorded on concentric circular tracks. To read the data, the head arm assembly brings the head directly above the track on the rotating disk. As previously recorded flux reversals pass under the head, a small signal will be induced. The signal from the disk is therefore a series of pulses, each of which are caused by flux reversals on the magnetic medium. The pulse detector must accurately replicate the time position of the peaks of these pulses. This task is complicated by variable pulse amplitudes depending on the media type, head position, head type and the gain of the Read/Write amplifier. Pulse amplitudes may vary on any one track if the distance between the head and the media varies as the disk rotates. Additionally, as the bit density on the disk increases, significant bit interaction occurs resulting in decreased amplitude, pulse distortion and peak shift.

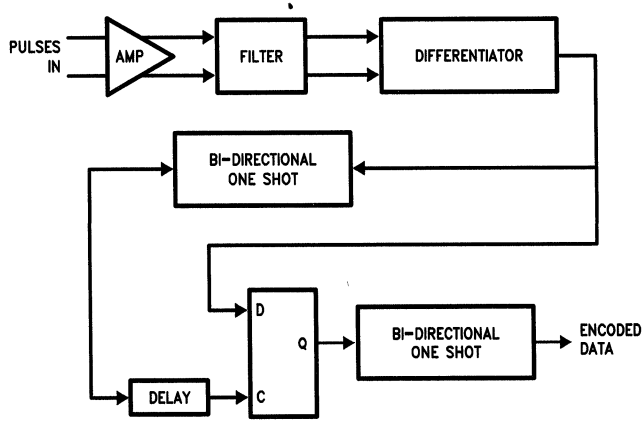
Traditionally MFM code has been used to encode digital information on the disk surface. MFM code uses the limited frequency range of F to 2F as illustrated in *Figure 2.1.* Such a system can use a special self gating circuit for the pulse detector as shown in *Figure 2.2.* Pulses from the inner track (the bottom waveform in *Figure 2.1*) are almost sinusoidal so the peak detector can simply differentiate the waveform to determine the peak positions. On the outer track however, the pulses have a small amount of shouldering as shown in the top waveform in *Figure 2.1.* The problem is that any noise occurring during these very narrow shoulders can be incorrectly interpreted as signal peaks.



TL/F/8663–37

Typical MFM waveforms on oxide media with ferrite heads. Since there is only slight shouldering on the outer track, the traditional self gating (de-snaker) pulse detector can be used.

FIGURE 2.1



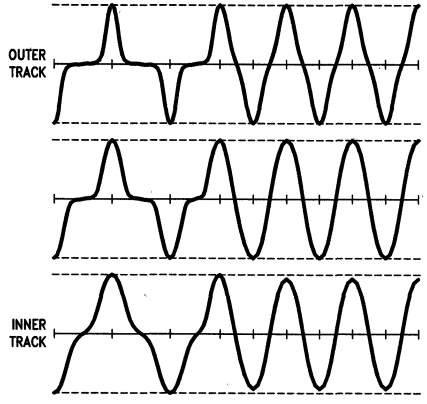
TL/F/8663–38

Self gating circuit (de-snaker) traditionally used for pulse detection of MFM code on oxide media with ferrite heads. The amount of delay in the clock line must exceed the maximum amount of shouldering.

FIGURE 2.2

The self gating circuit places a fixed delay and bi-directional one shot between the output of the differentiator and the clock input. The amount of this delay is selected to be longer than the worst case shouldering. When shoulder-induced noise occurs, the D input to the flip-flop will change states. However, by the time the clock occurs, the noise will no longer be present and the D input will have returned to its previous state. The flip-flop will therefore "clock" in the previous data. The output Q will not change states due to these narrow shoulder-induced noise pulses. This fixed delay in the clock line is called a time domain filter. This circuit is also known as a "de-snaker", named after the snake appearance of the waveform which exhibits slight shouldering. This "de-snaker" circuit works very well with waveforms which exhibit only slight shouldering. Unfortunately, the new methods used to increase the disk capacity do not produce such simple pulse patterns.

There are several methods of increasing the disk capacity. This includes plated media, thin film heads, and run length limited codes. All of these techniques result in narrow pulses with increased shouldering. An example of these slimmer pulses is shown in *Figure 2.3*. Instead of the slight shouldering present in the MFM example, the signal returns to the baseline between pulses. Since the shouldering is so extensive, the "de-snaker" technique simply will not work here. If a long delay were used to correct the shouldering present in the top waveform, it would not capture the pulses at the highest frequency.



TL/F/8663–39

Pulse waveforms for high resolution technologies. The large shouldering on the outer tracks precludes the use of the de-snaker pulse detector.

FIGURE 2.3

Detecting pulse peaks of waveforms of such variable characteristics requires a means of separating both noise and shouldering-induced errors from the true peaks. The old self gating circuits (such as the "de-snaker") will not work with the new techniques to increase the disk capacity. Hence the need for a circuit that includes a peak sensing circuit with an amplitude sensitive gating channel in parallel. Such a circuit is a key feature of the DP8464B Pulse Detector.

2.2 DP8464B FEATURES

Certainly a key feature of the DP8464B is the combination of a peak sensing circuit with an amplitude sensitive gating channel in parallel which allows the DP8464B to accurately detect the peak of waveforms that preclude the use of the traditional "de-snaker" circuit. The DP8464B, however, has many other features that make it ideal for the disk drive read channel.

Another key feature of the DP8464B is a wide bandwidth automatic gain controlled (AGC) amplifier. The automatic gain control removes the signal level variations of the read signal. The amplifier's wide bandwidth (20 MHz) insures that timing errors will not be introduced by the amplifier's pole.

The DP8464B offers considerable flexibility to the user, allowing him to tailor various operating characteristics to his specific needs. In particular, the user can set the frequency response of the differentiator, the width of the pulses on the encoded data output, the amount of hysteresis in the gating channel, the signal amplitude at the output of the gain controlled amplifier and the overall frequency response of the system and AGC. This kind of flexibility is provided by strategically placing pinouts at key points throughout the circuit. Differential signal paths were utilized whenever possible to minimize effects of power supply noise and external noise pickup. The IC can be effectively disabled when the disk drive is in a write mode, thus preventing saturation of the input amplifier and preventing disturbance of the AGC level.

The IC is powered from a single +12V supply (which is standard in most drives) and has an internal regulator and separate analog and digital grounds in order to properly isolate the sensitive analog circuitry. It is presently offered in a 24-pin DIP but will soon be made available in a 28-pin PCC surface mount package.

The DP8464B is fabricated on an advanced low power Schottky process which allows the part to handle data rates up to 15 Megabits/sec., 2, 7 Code.

2.3 THE DP8464B HARD DISK PULSE DETECTOR OPERATION

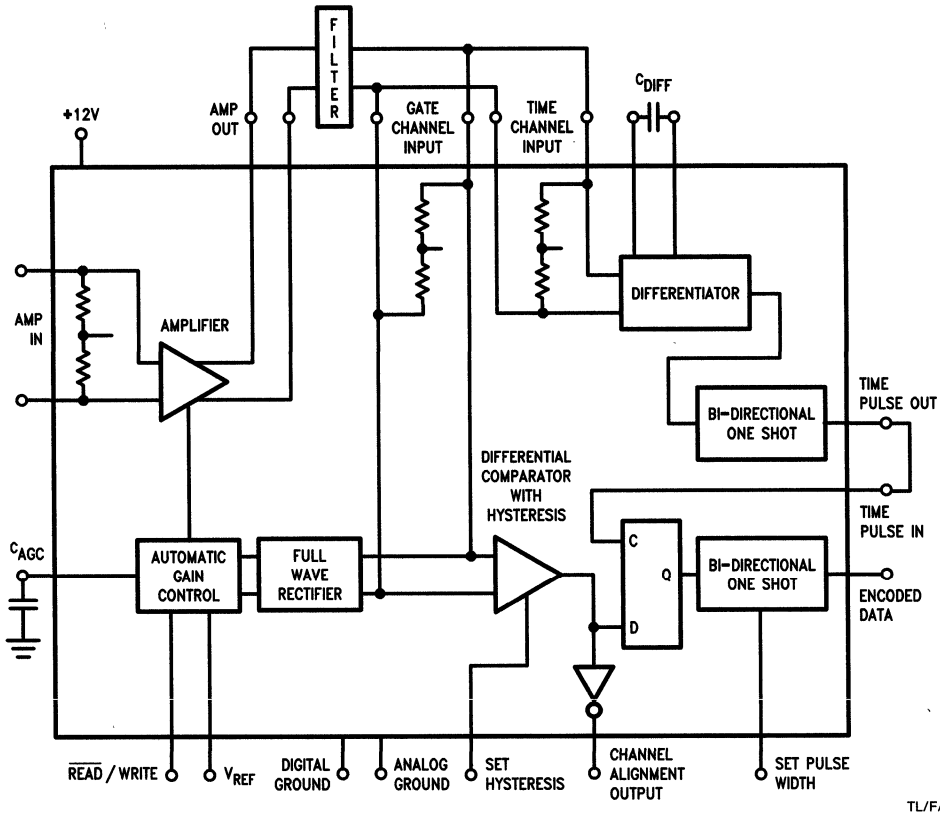
The main circuit blocks of the DP8464B are shown in *Figure 2.4*. The circuit consists of three main sections: the Amplifier, the Time Channel and the Gate Channel. The Amplifier section consists of a wide bandwidth amplifier, a full wave rectifier and the Automatic Gain Control (AGC). The Time Channel is the differentiator and its associated bi-directional one shot, while the Gate Channel is made from the Differential Comparator with Hysteresis, the D flip-flop and its following bi-directional one shot. Also, there is special circuitry for the Write mode. To better understand the circuit operation, let's discuss each section separately.

2.3.1 Gain Controlled Amplifier

The purpose of the Amplifier is to increase the differential input signal to a fixed amplitude while maintaining the exact shape of the input waveform. The Amplifier is designed to accept input signals from 20 mV_{pp} to 660 mV_{pp} differential and amplify the signal to 4 V_{pp} differential. The gain is therefore from 6 to 200 and is controlled by the Automatic Gain Control (AGC) loop.

2.3.2 Time Channel

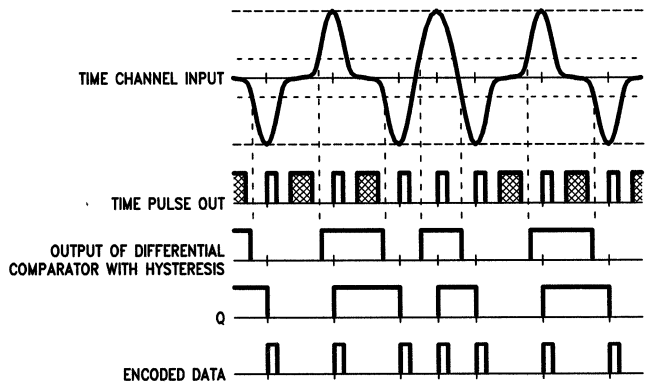
The peak detection is performed by feeding the output of the Amplifier through an external filter to the Differentiator. The Differentiator output changes state when the input pulse changes direction, generally this will be at the peaks. The Differentiator can also respond to noise near the baseline, in which case the Gating Channel will inhibit the output pulse (as discussed in the Gate Channel section). The purpose of the external filter is to bandwidth limit the incoming signal for noise considerations. Care must be used in the design of this filter to ensure the filter delay is not a function of frequency. The output of the Differentiator drives a bi-directional one shot which creates the Time Pulse Out.



TL/F/8663–40

Main circuit blocks of DP8464B. The three main sections are the Amplifier (amplifier and AGC), the Gating Channel (comparator with hysteresis and D flip-flop), and the Time Channel (differentiator and bi-directional one shot).

FIGURE 2.4



TL/F/8663–41

These signal waveforms illustrate the operation of the DP8464B. The noise in the Time Pulse Out (which occurs during the shouldering) simply clocks in old data present at the output of the Differential Comparator with Hysteresis. A bi-directional one shot at the Encoded Data output provides a rising edge representing the relative time position of the peaks at the Time Channel Input.

FIGURE 2.5

2.3.3 Gate Channel

While the actual peak detection is done in the Time Channel with the Differentiator, there is the problem of preventing the output data from being contaminated when the Differentiator responds to noise at the baseline. To prevent this, the signal is also passed through a Gate Channel which prevents any output pulse before the input signal has crossed an established level. This Gate Channel comprises a Differential Comparator with Hysteresis and a D flip-flop. The hysteresis for this comparator is set externally via the Set Hysteresis pin.

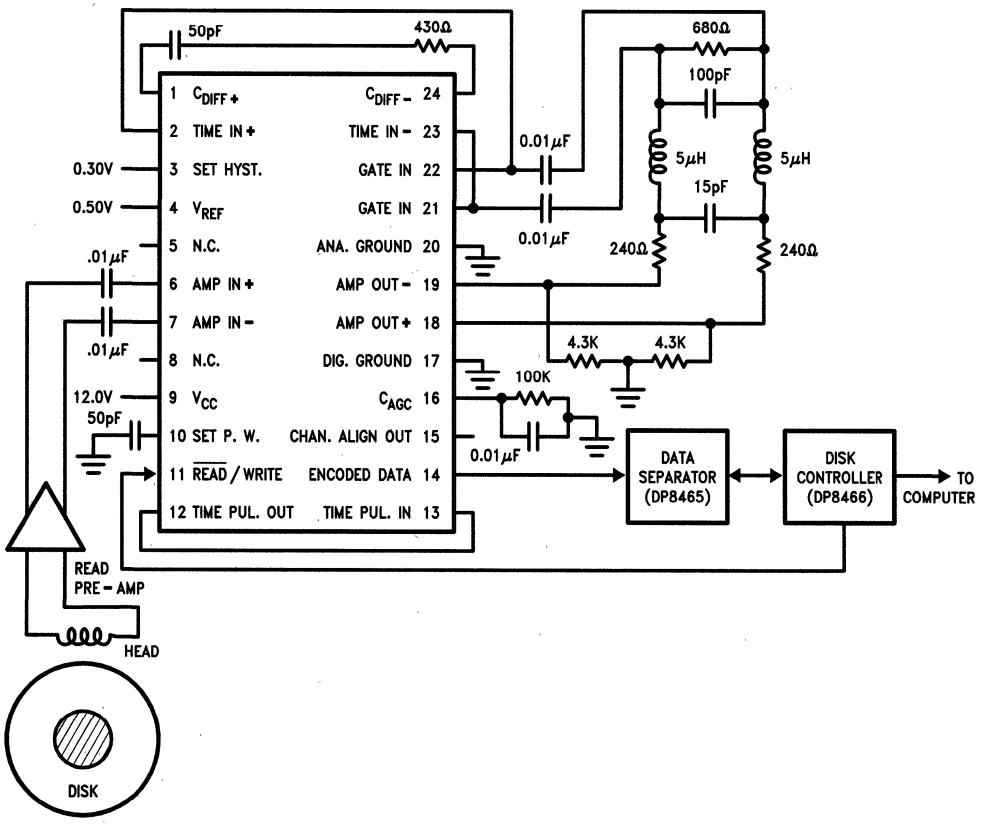
The operation of this Gate Channel is shown in *Figure 2.5*. At the top is a typical waveform which exhibits shouldering at the lowest frequency, and is almost sinusoidal at the highest frequency. This waveform is fed to both the Time and the Gate Channel. The hysteresis level (of abut 30%) has been drawn on this waveform. The second waveform is Time Pulse Out. While there is a positive edge pulse at each peak, there is also noise at the shoulders. Since Time Pulse Out is externally connected to the Time Pulse In, this output is therefore the clock for the D flip-flop.

The third waveform is the output of the Comparator with Hysteresis which goes to the D input of the flip-flop. The true peaks are the first positive edges of the Time Pulse Out which occur after the output of the comparator has changed states. The D flip-flop will "clock" in these valid peaks to the output bi-directional one shot. Therefore, the noise pulses due to the Differentiator responding to noise at the baseline just "clock" in the old data through the flip-flop so there is no noise pulse on Encoded Data.

The Q output of the flip-flop drives the bi-directional one shot which generates the pulses on Encoded Data. The positive edges on Encoded Data correspond to the signal peaks. The width of the data pulses can be controlled by an external capacitor from the Set Pulse Width pin to ground. This pulse width can be adjusted from 20 ns to 1/2 the period of the highest frequency.

Design Example

A typical system implementation of the DP8464B is shown in *Figure 2.6*. The DP8464B is driven from the Read/Write amplifier which is generally located very close to the actual Read/Write head. This amplifier is fixed gain and pre-amplifies the weak readback signal picked up by the head from the surface of the disk.



TL/F/8663–42

Shown here is a typical 10 Mbit/sec MFM disk drive application. Signals are picked off the disk by the Read/Write head and amplified by the pre-amplifier. The DP8464B further amplifies the read signal and accurately represents the time position of the peaks by the rising edge of the signal at the Encoded Data output pin. The Data Separator locks up to the signal at the Encoded Data output and provides decoded MFM data to the Disk Data Controller. The Disk Data Controller handles the interface between the disk drive and the computer.

FIGURE 2.6

The output of the DP8464B drives a data separator (such as the DP8465) which extracts the digital data from the encoded data supplied by the DP8464B. The data separator utilizes a phase-locked loop that locks onto the leading edge of the encoded data signal from the DP8464B.

In addition to extracting the digital data from the encoded data, the data separator synchronizes the digital data thereby removing any timing jitter present in the encoded data signal. The data separator implements this last function by opening up timing windows that bracket the encoded data signal. The encoded data signal need only appear within the window to be detected. For 10 Mbit MFM, the window is only 50 ns wide.

In order to guarantee that the drives have error rates on the order of 1 per 10 to the 10th power, bits read (industry standard), the leading edge of the encoded data must fall well within the 50 ns window. Because of this stringent criteria, there is little room for error with regard to the accuracy that the DP8464B can extract the relative time position of the peaks of the read back signal.

One form of timing error is jitter of time position of the leading edge of the encoded data signal. This jitter can be a result of noise output from the differentiator or noise pickup in other portions of the read channel. These timing errors will significantly affect the Encoded Data signal causing an increase in the error rate.

The filter that drives the differentiator is important in reducing the noise input to the differentiator. For this reason, a high order Bessel filter with its constant group delay characteristic can be used in this application. The constant group delay characteristic insures that the filter does not introduce any timing errors by distorting the signal and moving the position of the peaks. Often, this filter must be specifically designed to correct phase errors introduced by the non-ideal characteristics of the input read head. The typical −3 dB point for this filter is around 1.5 times the highest recorded frequency. Reducing the noise input to the differentiator will ultimately reduce the amount of noise jitter on the encoded data output.

Another way to reduce noise jitter is to limit the bandwidth of the differentiator with a series combination of resistor, capacitor and inductor in the external differentiator network and to use as large a differentiator capacitor as possible, thereby maximizing the differentiator gain. In order to prevent saturation of the differentiator, Schottky diode clamps were added to the differentiator output thus allowing the use of a larger differentiator capacitor.

An automatic gain control (AGC) circuit is used to maintain a constant input level to the gating channel (which is typically tied directly to the input of the differentiator). By maintaining a constant signal level at this point, we insure not only a large input level to the differentiator but also a constant level of hysteresis of the signal to the gating channel. Gain control is also necessary because the amplitude of the input signal will vary with track location, variations in the magnetic film, and differences in the actual recording amplitude. The peak-to-peak differential amplitude on the Gate Channel Input is four times the voltage set by the user on the V_{ref} pin.

The actual dynamics of the AGC loop are very important to the system operation. The AGC must be fast enough to respond to the expected variations in the input amplitude, but not so fast as to distort the actual data. A simplified circuit of the AGC block is shown in *Figure 2.7*. When the full wave rectified signal from the Amplifier is greater than V_{ref}, the voltage on the collector of transistor T1 will increase and charge up the external capacitor C_{agc} through T2. The maximum available charging current is 3 mA. Conversely, if this input is less than V_{ref}, transistor T2 will be off, so the capacitor, C_{agc}, will be discharged by the base current going into the Darlington T3 and T4. This discharge current is approximately 1 μA. The voltage on the emitter of T4 controls the gain of the Amplifier.

If the AGC circuit has not received an input signal for a long time, the base current of the Darlington will discharge the external C_{agc}. The Amplifier will now be at its highest gain. If a large signal comes in, the external C_{agc} will be charged by the 3 mA from T2, thereby reducing the gain of the Amplifier. The formula, $I = C^*(dV/dt)$ can be used to calculate the time required for the Amplifier to go from a gain of 200 to a gain of 6. For instance, if $C_{agc} = 0.05$ μF, the charging current I is 3 mA, and the dV required for the Amplifier to go through its gain range is 1V, then

$$dt = (0.05\ \mu F^*\ 1V)/(3\ mA) \text{ or } 17\ \mu s.$$

By using the same argument, the time required to increase the Amplifier gain after the input has been suddenly reduced can be calculated. This time, the discharging current is only 1 μA, so

$$dt = (0.05\ \mu F^*\ 1V)/(1\ \mu A) \text{ or } 50\ ms.$$

This time can be decreased by placing an external resistor across C_{agc}.

2.4 READ/WRITE

In the normal read mode, the signal from the read/write head amplifier is in the range of 20 mV_{pp} to 660 mV_{pp}. However, when data is being written to the disk, the signal coming into the analog input of the pulse detector will be on the order of 600 mV. Such a large signal will disturb the AGC level and would probably saturate the amplifier. In addition, if a different read/write amplifier is selected, there will be a transient introduced because the offset of the preamplifiers are not matched.

A TTL-compatible $\overline{\text{READ}}$/WRITE input pin has been provided to minimize these effects to the pulse detector. When the $\overline{\text{READ}}$/WRITE pin is taken high, three things happen. First, the 1k resistors across the AMP IN pins are shunted by 300Ω resistors. Next, the amplifier is squelched so there is no signal on the Amp Output. Finally, the previous AGC level is held. This AGC hold function is accomplished by not allowing any current to charge up the external C_{agc}. The voltage across this capacitor will slowly decrease due to the bias current into the Darlington (see *Figure 2.7*) or through any resistor placed in parallel with C_{agc}. Therefore, the gain of the amplifier will slowly increase. All of these three events happen simultaneously.

When the $\overline{\text{READ}}$/WRITE input is returned low, the pulse detector will go back to the read mode in a specific sequence. First, the input impedance at the Amp In is returned to 1k. Then, after approximately 1.2 μs, the Amplifier is taken out of the squelch mode, and finally approximately 1.2 μs after that, the AGC circuit is turned back on. This return to the read mode is designed to minimize analog transients in order to provide stable operation after 2.4 μs.

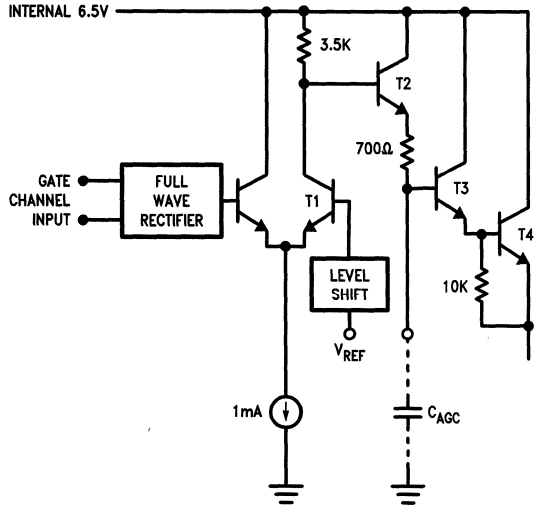
2.5 CONCLUSION

The push to higher disk capacities in increasingly smaller drives has forced drive manufactures to utilize different media, heads and encoding. Each of these changes render the traditional de-snaker pulse detector unusable. The DP8464B pulse detector utilizes a new detection technique that overcomes the limitations of the de-snaker. Furthermore, the DP8464B provides both gain controlled amplification of the pre-amplified readback signal and the ability to disable the circuit during write operations. The DP8464B is easily adapted to a wide variety of applications through selection of external components.

References

1. I. H. Graham, *"Data Detection Methods vs. Head Resolution in Digital Recording,"* IEEE Transactions on Magnetics Vol. MAG-14, No. 4 (July 1978).

2. I. H. Graham, *"Digital Magnetic Recording Circuits,"* available through the University of Santa Clara, (California), bookstore (408) 554-4491.



TL/F/8663–44

The AGC Circuit senses the level of the signal at the Gate Channel Input and compares it to an externally set reference voltage. The signal that results from this comparison (at the collector of T1) charges C_{AGC}. The voltage across C_{AGC}, when buffered by T3 and T4, provides the gain control voltage to the input amplifier.

FIGURE 2.7

CHAPTER 3 DISK DATA SEPARATOR OVERVIEW (DP8460/61/62/65 AND DP8451/55)

3.0 INTRODUCTION—THE DATA SEPARATOR

As was discussed in the chapter on Disk Drive Technology (overview), the disk information which is recovered during a read operation ordinarily would have no defined phase relationship with respect to the timing within the host system. In order to establish a method of reconstructing a clock waveform with which the disk data may be entered into a shift register for deserialization and decoding, clock information is imbedded into the recorded bit pattern in any of a number of different ways by the various encoding schemes discussed in Chapter 1. The schemes vary in their efficiency of use of disk surface (bit density) and ease of recovery (challenge to the data separator), but they all are employed to achieve a mixture of clock and data within the same serial bit stream. It is then the function of the data separator to accurately extract this clock information from the bit stream and reconstruct a stabilized replica of the data, while at the same time remaining essentially immune to the random displacement of individual bits due to noise, media defects, pulse crowding and anomalies in the data channel.

From a "black box" standpoint, the data separator is fed a logic-level digital signal from a pulse detector (DP8464) within the disk head electronics (with positive transitions representing flux reversals on the media) and a read gate signal from the controller, and produces a reconstructed clock waveform along with a re-synchronized data output derived from the incoming disk pulse stream (see *Figure 3.1*). The regenerated clock and data signals have fixed timing relationships with respect to one another for use by subsequent shift register circuitry.

3.0.1 Separators and Synchronizers

The term "data separator" actually applies to a device which both regenerates a clock waveform from the bit stream as well as decodes (separates) the original NRZ data from the encoded disk data. This would include National's DP8461 and DP8465, both of which perform data synchronization with MFM-to-NRZ data separation, while including slight functional variations between devices. (The DP8460 initially released device is being replaced by the fully pin-for-pin compatible DP8465. The DP8461 pinout matches the DP8465, but is intended for use with hard and pseudo-hard sectoring only. Further details will be discussed later.) A device which performs clock regeneration and data synchronization without the separation function is simply called a data synchronizer. This would include the DP8451, DP8455, and the DP8462; again, there are functional differences between the devices, which will be discussed later. Additionally, the DP8461 and DP8465 also have outputs available which allow them to serve as data synchronizers, if desired (See Table 3.1).

The complete data separator circuit eliminates the need for external decoding circuitry but is dedicated to only a single code type. The data synchronizer requires an external decoding network but has the capacity to be used with any coding scheme. Since the MFM environment is being addressed in this design guide, the discussion in the remainder of this chapter will deal primarily with the integrated, MFM-type data separator. The PLL fundamentals being presented apply to all of the circuits.

3.0.2 Window

The data separator must establish what is called a "window" around the expected position of bits within the disk data stream. Windows are laid end-to-end in time by the data separator at a repetition rate (equal to the separator's VCO frequency) known as the disk code rate. Each window is an allotment of time within which a disk bit, if detected, will be captured and interpreted as if it had occurred exactly at the window center. This allows for a random displacement (jitter) of individual bits within the boundaries of the window with no apparent effect on the accuracy of the data recovery (error rate). Bits are displaced from a nominal position in a fashion which could be represented by a bell-shaped probability curve (see *Figure 3.2*), and it could be easily seen that for optimum performance, the window must be accurately centered about the mean of this curve. This is traditionally a difficult goal to achieve, and is essentially the primary responsibility of the data separator. Although various techniques have been employed to attain this goal, the phase-locked loop (incorporated within all National data separator/synchronizers) has proven to date to yield the most reliable and satisfactory results among all the synchronization methods, and its use is the standard approach taken within the disk industry.

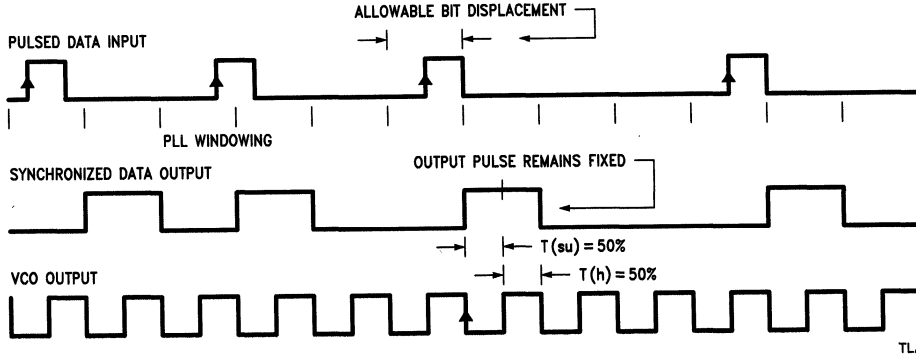
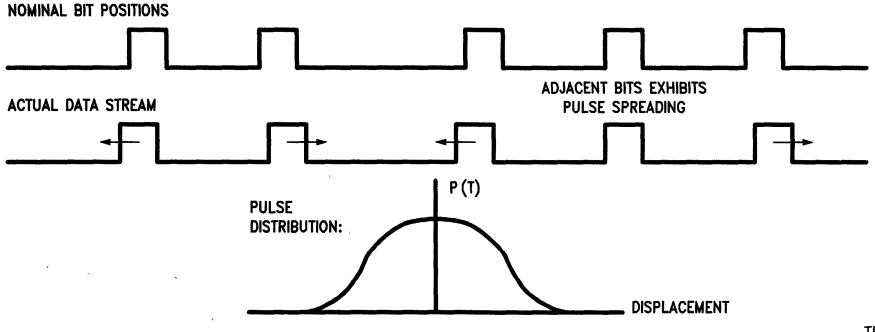


FIGURE 3.1. DP8460-Series Data Synchronizer Timing Relationships

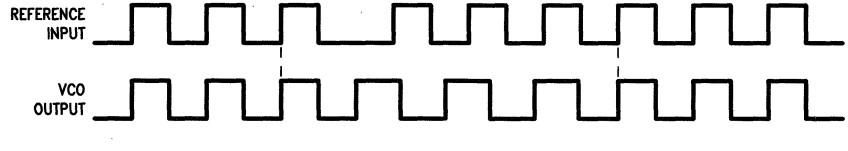
3.1 PHASE-LOCKED LOOP OVERVIEW

A phase-locked loop is a closed-loop control system which forces the phase from the output of a controlled oscillator to track the phase of an external reference signal. It consists of three essential elemental blocks; a phase detector, a loop filter, and a voltage controlled oscillator (VCO) (see *Figure 3.3*). The phase detector compares the phase of the reference input with that of the VCO output and generates an "error" signal at its output which is proportional to the sensed phase difference. This error signal is filtered by the loop filter (low-pass) to suppress any unwanted high-frequency components within the error signal, and is then fed to the VCO control voltage input. If the phase of the reference signal leads that of the VCO signal, the phase detector develops a positive error voltage across the loop filter, and the VCO responds by increasing its frequency (advancing phase). This continues until the phase error is eliminated and the control voltage returns to a quiescent value. When the reference signal lags that of the VCO, the adjustment occurs in the opposite direction until equilibrium (phase lock) is again obtained.

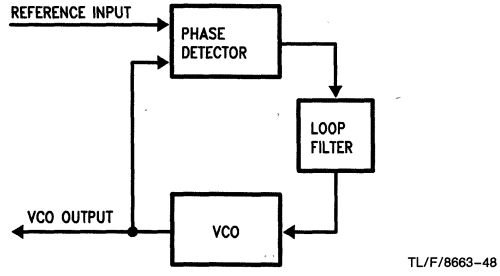


TL/F/8663–46

FIGURE 3.2. Data Jitter



TL/F/8663–47



TL/F/8663–48

A phase-locked loop is a system which forces the phase of the output from a voltage controlled oscillator (VCO) to track the phase of a reference signal.

FIGURE 3.3

3.1.1 PLL Dynamics

Much of the performance of the PLL, given adequate design in the major functional blocks, is determined by the loop filter. This includes (1) the ability of the PLL to rapidly (or slowly) track phase or frequency changes in the reference signal, (2) the ability of the loop to re-aquire lock after encountering a large frequency step at the reference input, and (3) the ability of the loop to exhibit stable behavior during operation.

TRACKING

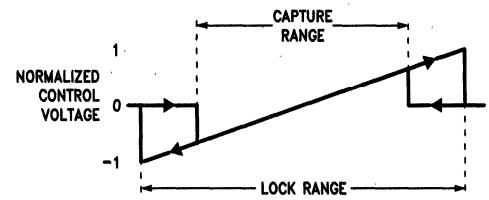
In many PLL applications, such as FM demodulation and frequency shift keying demodulation, rapid PLL tracking (high bandwidth) is a necessity. However, in the disk drive application, where (1) frequency changes (disk rotational speed variations) are gradual with respect to the data rate and (2) it is desirable to suppress response to instantaneous bit shift (jitter), a very slow tracking rate (low bandwidth) is necessary.

CAPTURE RANGE

The classical PLL is well able to maintain relative phase lock to a reference signal, but is unable to pass true difference-frequency information through its phase detector (this is true for the standard analog four-quadrant multiplier technique as well as for the gated-VCO technique employed in disk drive data separation applications). With this being the case, the PLL has a limited ability to re-establish lock when an instantaneous input frequency change occurs. The new frequency must lie inside a relatively narrow band on either side of the current VCO frequency, or re-lock will not occur (see *Figure 3.4*). This band is known as the capture range, and is a direct function of the passband of the loop filter, or more accurately, of the bandwidth of the PLL as a whole. A digital frequency discrimination technique, however, is employed in National Semiconductor's disk drive PLL's which provides an extended capture range and guarantees successful lock to the reference clock input and, as a chip dependent option, to the data as well. Consequences of having a limited capture range are discussed in National Semiconductor Application Notes AN414, AN415, AN416 (Also see "Frequency Lock" in section 3.2 of this chapter).

STABILITY

Mathematical analysis of the functional blocks of the PLL show a 1/s factor in the VCO; i.e., it behaves as a integrator, adding a pole to the transfer function. The loop filter adds at least one additional pole, resulting in a system which is, at minimum, second order in nature. Since the PLL is then a closed loop, minimum second order system, it has the potential for instability if improperly implemented. All of the dynamic characteristics of the PLL, however, can be controlled by loop filter selection, including bandwidth and capture range along with stability; thus great care must be taken in the selection of the loop filter in order to achieve the desired performance within the specific application.



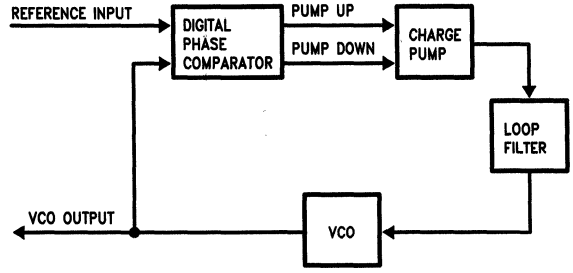
TL/F/8663–49

Capture range in a PLL is determined by the loop filter bandwidth.

FIGURE 3.4. Capture and Lock Range

3.2 THE PLL WITHIN A DISK DRIVE SYSTEM

Many methods have been employed in implementing each of the individual blocks within the phase locked loop (PLL), with techniques being customer-tailored to specific applications. Design methods include analog and digital configurations or a combination of both. For the application of a PLL within a disk drive data separator (and specifically regarding National's family of data separator/synchronizers), a combination of digital and analog block design has been found to provide the most efficient and reliable solution (see *Figure 3.5*). Here, since the waveforms to be compared are digital signals and the phase relationships are indicated by logical transitions (positive edges), the phase detector is comprised of a simple set of cross-coupled latches which produce "pump-up" (reference leads VCO) and "pump-down" (reference lags VCO) digital outputs. Since the filtering, however, is most easily and flexibly performed with passive analog components, the pump-up and pump-down signals are converted into gated sourcing and sinking currents, respectively, via an analog "charge pump" circuit, which is used to develop an error voltage across a capacitive loop filter (see *Figure 3.6*). This error voltage is used as a control potential for a variable rate relaxation oscillator (emitter coupled multivibrator VCO), whose oscillation is converted to digital signal levels again for use both at the phase detector input and as the regenerated clock waveform.

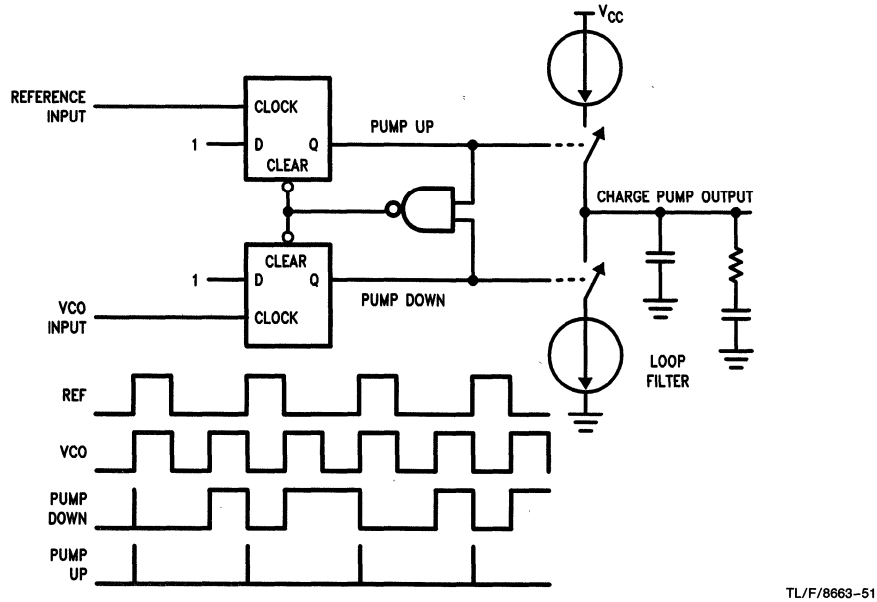


TL/F/8663–50

Goals:

1. Edge sensitive detector which eliminates dependence on waveform shape.
2. Unlimited capture range to ensure phase and frequency lock.
3. Zero phase difference when in lock to improve lock range.

FIGURE 3.5. Basic Disk System PLL



TL/F/8663–51

FIGURE 3.6. Digital Phase Detector

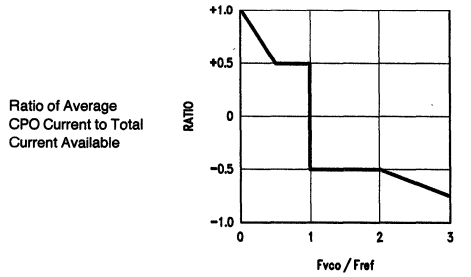
PHASE DETECTOR

The digital phase detector employed within National's data separator/synchronizer circuits is actually a true phase/frequency discriminator block capable of allowing a theoretically infinite capture range for the PLL (see *Figures 3.7–3.10*). Essentially, the capture range is limited only by the design constraints placed on the VCO's frequency excursion. In all of National's current disk PLL circuits (excluding the DP8460/50), this extended lock capability is employed while the circuit is in the non-read mode and the PLL is locked to a constant reference signal, guaranteeing proper lock recovery from any given mislock which may occur during a read operation.

PULSE GATE

The data returning from the disk is not a periodic waveform, but instead has the possibility of bits either appearing or not appearing within assigned positions (windows) in the data stream (see *Figure 3.1*). The PLL is required to achieve and maintain lock to this pseudo-random pattern, despite the missing bits. This is analogous to the placement of teeth in a gear (data separator) which are ready to mesh with another gear (data stream), regardless of whether or not some teeth on the second gear (data) are occasionally missing (random data patterns).

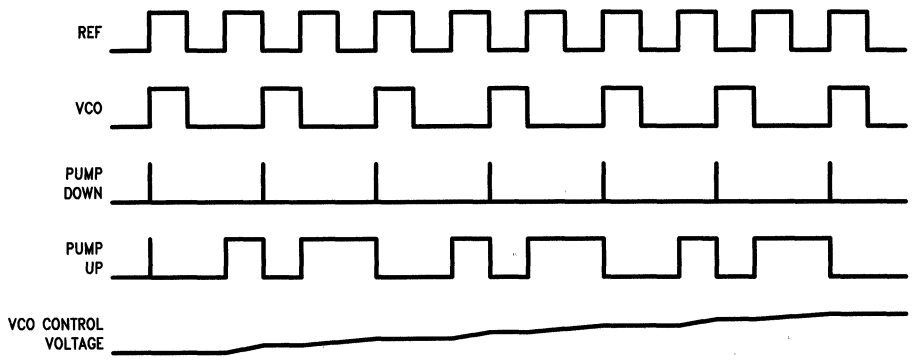
In order to allow for the missing bits, the PLL employs a Pulse Gate circuit, which functions as follows: a data bit arriving at the PLL is sent to its corresponding input on the phase detector, and at the same time trips a gate which allows the next occurring VCO edge into the phase detector; the gate then closes following transmission of the VCO edge. If no data bit arrives, no comparison occurs and the VCO holds its frequency. Essentially, the PLL is attempting to align each data bit with the nearest occurring VCO edge, thus maintaining phase lock while frequency discrimination is suppressed (see *Figures 3.11* and *3.12*).



TL/F/8663–52

Ensures unlimited capture range. The digital phase detector ensures frequency acquisition by forcing the charge pump to always pump in the direction needed to make F_{VCO} equal to F_{REF}.

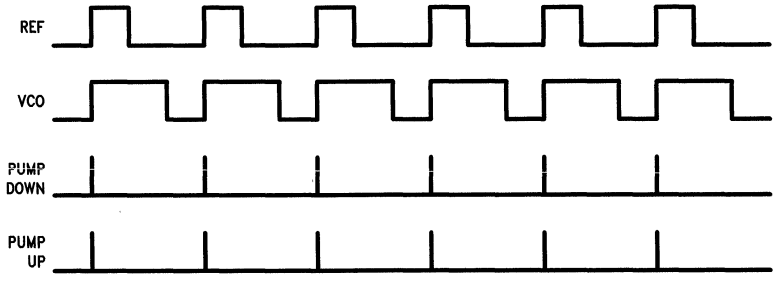
FIGURE 3.7. Digital Phase Detector



TL/F/8663–53

When the digital phase-lock loop is out of lock, the output of the phase comparator has a duty cycle which varies between 0 and 100%. The charge pump is active more than 50% of the average time but it only pumps current in the direction necessary to lock the VCO phase.

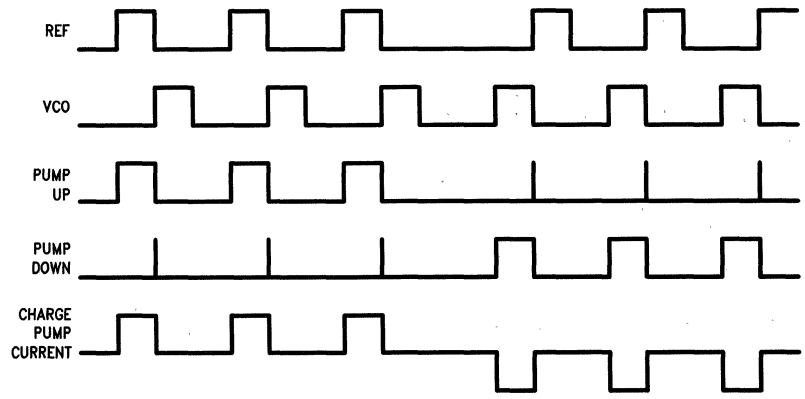
FIGURE 3.8. Digital Phase Detector



TL/F/8663–54

Once the VCO is locked to the reference signal, the only phase difference which occurs will be that required to pump enough charge to compensate for any leakage current in the charge pump, loop filter or bias current in the VCO control input.

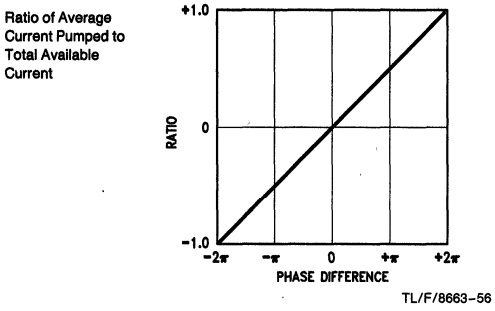
FIGURE 3.9. Digital Phase Detector



TL/F/8663–55

Effect of phase difference on digital phase detector when the VCO and reference frequencies are equal. The net charge pumped during each period is equal to the product of the charge pump current and the time difference between the phase comparator inputs.

FIGURE 3.10. Digital Phase Detector

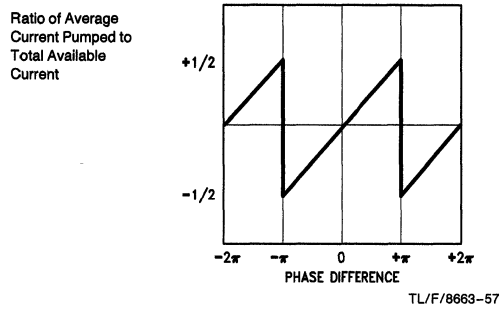


Phase detector gain when in lock

When the loop is in lock the net charge (Q) pumped during each period is equal to the product of the charge pump current and the time difference between the phase comparator inputs.

$$Q = I_{cpo}(T_{up} - T_{down})$$

FIGURE 3.11. Digital Phase Detector Gain without Pulse Gate



Each vertical transition represents a window boundary.

FIGURE 3.12. Digital Phase Detector Gain with Pulse Gate

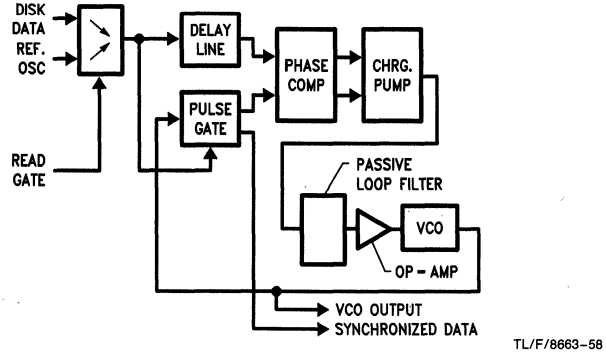


FIGURE 3.13. Disk Data PLL

DELAY LINE

In the generation of the symmetrical bit-detection window mentioned previously, a delay line is employed as shown in *Figure 3.13*. The incoming data is allowed to trip the VCO pulse gate immediately upon arrival, but is allowed into the phase detector only after it traverses the delay line. To understand the purpose for the delay line, first consider the case where the delay line is not present; given proper PLL lock, the VCO would align its edge to occur exactly at the same time as the arrival of each data bit. Any bits shifted early would cause a small VCO phase-advancing correction within the loop, which would be desirable. However, any bits shifted even a very slight amount late would arrive after the current VCO edge had passed and been suppressed by the pulse gate. The bit would then have to be compared to the subsequent VCO edge instead of to the current edge, producing an erroneous phase correction. With the delay line in place and set to delay the data bit by one-half of the VCO period, the data bit would first trip the VCO gate and then spend one-half of the VCO period traversing the delay line before it reached the phase detector. Given the loop is in lock, both the delayed bit and the VCO edge arrive at the phase detector at exactly the same time. If the bit were early up to one-half VCO cycle, it would still gate the appropriate VCO edge through, and produce an appropriate phase correction at the phase detector. Also, if the bit were late up to one-half of the VCO cycle, it would again still gate the appropriate VCO edge through to the phase detector, as well as produce the appropriate phase correction. Thus, the net effect of the delay line is to allow the incoming data bit to shift either one-half cycle early or one-half cycle late while yet maintaining a proper comparison to the appropriate VCO edge.

WINDOW ACCURACY

As mentioned previously, the integrity of the window alignment is crucial in maintaining an acceptable system error rate. It can be easily seen that accuracy in the delay line is critical in achieving this alignment. This has traditionally made the implementation of a delay line a costly design challenge. Within National's data separator/synchronizer

circuits, a proprietary technique has been employed which extracts precise timing information from the 2F CLOCK input waveform, and uses this information to regulate the timing within an on-chip silicon delay line. This in itself has unique advantages since the 2F source is either (1) a highly accurate crystal oscillator source, or (2) derived from the disk servo clock which tracks the data rate and consequently allows the delay line to adjust its delay accordingly. Completely dependent on this 2F signal for timing information, the delay remains independent of variations in its associated external components, power supply, temperature, and silicon processing. It requires no adjustment (although fine tuning is optional on the DP8462), and its accuracy is guaranteed within the window tolerance specification for the device.

FREQUENCY LOCK

The frequency discriminating capability of the phase detector within National's data PLL circuits can be employed to great advantage if used appropriately. It is brought into play simply by the internal bypassing of the pulse gate circuit. The advantages achieved are (1) the avoidance of mis-lock to the reference clock input, (2) rapid and guaranteed lock recovery from an aborted read operation, and (3) avoidance of mis-lock within the disk PLL synchronization field (preamble).

Items #1 and #2 above are easily attained by pulse gate bypassing when the PLL is locked to the reference signal in the non-read mode. Pulse gate bypassing allows the digital phase detector to perform unrestricted frequency comparison and thus guarantees lock. Both items are employed within all of National's currently released data separator/synchronizer circuits (see Table 3.1).

Incorporation of item #3 (employed within the DP8461 and DP8451, and optional within the DP8462) is highly dependent on preamble type and places specific requirements on the controller's sector search algorithm. First, there are several common preamble types currently in use on disk drives; (1) the MFM and 1,N type, (2) the 2,7 high frequency preamble, the (3) the 2,7 low frequency preamble:

Code Type	VCO Cycles (Code Positions or Windows) Per Recorded Preamble Bit
GCR*	1
MFM	2
1,7	2
1,8	2
2,7	3
2,7	4

***Note:** GCR (Group Code Recording) is used almost exclusively in tape drive systems; it is mentioned here for comparative purposes only.

Since each preamble is recorded at a different frequency with respect to the VCO operating frequency, the VCO must be internally divided down to equal the preamble frequency for the particular code in use before being fed into the phase detector along with the data pattern. (This function is performed internally within specific National disk PLLs listed in Table 3.1.) It is then the responsibility of the PLL to detect the occurrence of frequency lock and revert back to the pulse gate mode prior to leaving the preamble and encountering random data patterns. Second, while in the frequency acquisition mode, the controller must allow a read operation to begin (initial PLL lock to the data) only during the presence of the appropriate field, i.e., the system must employ a hard-sectored or pseudo hard-sectored PLL control algorithm which will guarantee the PLL read gate will only be asserted at the start of the preamble on the disk, otherwise serious PLL mislock problems will result.

DP846X EXPOSITION

Because of the varied requirements and applications which exist for the data separator/synchronizer, National provides an assortment of disk PLL circuits, including versions which provide frequency lock for specific preamble types, as mentioned above.

TABLE 3.1. Data Separator/Synchronizer Reference List

Device	Synchronized Codes	Separated Codes	Frequency Lock	Delay Trim
DP8461*	MFM; 1,N	MFM	Reference & Data	None
DP8462*	2,7 1,N MFM	None	Reference (Optional for Data)	Optional
DP8465*	All	MFM	Reference	None
DP8451	MFM; 1,N	None	Reference & Data	None
DP8455	All	None	Reference	None

Note 1: "All" code synchronization does not include GCR.

Note 2: DP846X devices are in the 24-pin, 300 mil. package; DP845X devices are in the 20-pin, 300 mil. package.

Note 3: *Also available in 28-lead plastic chip carrier.

Note 4: DP8461 and DP8451 pinouts match the DP8465 and DP8455, respectively; for use with hard and pseudo-hard sectoring only.

Note 5: DP8451 and DP8455 are also available in 20-pin plastic chip carrier.

3.3 SYSTEM DYNAMICS—LOOP FILTER DESIGN

The key element contained within the PLL system for governing the loop dynamics and overall performance is the loop filter. It is at this point that the user has the greatest flexibility and control regarding the behavior of the PLL. As previously mentioned, there are several requirements placed on the dynamics of the loop, some of which tend to conflict with others. Table 3.2 lists some of the issues at hand, and where they lie with respect to one another.

TABLE 3.2

	High Bandwidth	Low Bandwidth	High Damping Factor	Low Damping Factor
Lock Time	Good	Poor	Good	Poor
Jitter Rejection	Poor	Good	Poor	Good
Capture Range	Good	Poor	Good	Poor
Noise Immunity	Poor	Good	Poor	Good
Stability	(No Relationship)		Good	Poor

Although the designations of "good" and "poor" are very general in nature, they apply fairly well here for comparative purposes. An ideal PLL would be able to lock to any frequency and/or phase step in a very short time with no possibility of missed or false lock, settle quickly to a highly stable state, and track any frequency variations encountered in the data stream while at the same time rejecting all bit jitter and extraneous noise. While all this is not possible with any single loop filter, acceptable performance can be achieved via careful compromise, with the design biased to accommodate the more critical parameters.

3.4 OVERVIEW OF FILTER DESIGN OBJECTIVES

The first design objective to be discussed is the minimization of acquisition time. This includes both acquisition of phase lock to the data stream as well as acquisition of phase lock to the crystal frequency (or servo track). Both of these acquisition times impact the length of the preamble field which precedes the address mark. Since longer preambles result in more overhead per data sector, a decrease in formatted disk capacity may result from excessively long acquisition times. Acquisition times are directly controlled by the phase locked loop filter.

The second design objective is the maximization of data margin. Data margin measures the ability of the data separator to allow the data bit to move from its expected time position without a resulting data error. This movement of the data bit away from its expected time position is caused by noise, read channel asymmetry, magnetic domain interference, and other factors in the head, media, and channel portion of the drive system. The data separator generates a window around the expected data position; however, the window accuracy is affected by some factors in the data separator. These factors include delay inaccuracy, VCO jitter, phase detector inaccuracy, and phase locked loop response to bit movement which occurred in preceding windows. The last of these factors, loop response, is controlled by the phase locked loop filter.

The final objective to be checked is the tracking of disk data. The rate of change of phase between the VCO and the read data is modulated by various mechanical phenomena in the drive. Instantaneous variations in disk speed as well as head vibration contribute to this modulation. The maximum frequency of these mechanical resonances tends to be in the 10 kHz or 64 Krads/sec range. Phase-locked loop bandwidth must be wide enough to allow this modulation to be tracked. This objective tends to be encompassed by the acquisition time objective. However, it is conceivable that a system which allows relatively long acquisition times may come up against this barrier.

The loop filter design process may start with any one of these objectives. If the disk format has been established, or a certain disk capacity is desired, the acquisition performance may dictate the loop filter design. If data reliability and error rates are of primary importance, the design may start with margin loss considerations. In any case, all aspects of the loop performance must be checked and the final design is usually a tradeoff between the desired performance and the achievable performance.

3.5 ACQUISITION PERFORMANCE

The read acquisition time is the time between the assertion of READ GATE and the reading of the address mark. Also of concern is the time required for the loop to acquire lock to the crystal frequency. Many application-specific system parameters impact this portion of the loop design. Some of these parameters which will be discussed include sector search algorithms in soft sectored systems and frequency differences between the crystal and the data in removable media systems.

Before the READ GATE is asserted, the VCO is locked to the crystal. When READ GATE is asserted, the phase difference between the VCO and the read data is random. The first portion of the acquisition is where the loop captures phase alignment. In the worst case, the initial phase alignment is such that the data bit is positioned at the edge of the window which gives the proper polarity of error signal, however, the loop cannot keep the bit in the window since it started so close to the edge. One of the results is that the incoming data will appear to be different than what is actually being read. Any system which desires to immediately monitor the read data must wait for this initial cycle slip to occur before reading. The second result is that a series of error signals of the wrong polarity occurs after this initial cycle slip while the phase aligns to the window center. The duration of this slip and phase acquisition is approximately $1/\omega n$ for damping factors between 0.7 and 1.0. See Section 3.7 for acquisition plots. Note that ωn is the loop bandwidth or the natural frequency of the loop and that the phase error is zero at $\omega nt = 1$.

The next period of the acquisition is where the loop begins to capture frequency. There is also some overshoot from the phase acquisition during this period. This analysis assumes that the frequency acquisition begins where $\omega nt = 1$ and is superimposed upon the phase acquisition for $t > 1/\omega n$.

In a fixed media system the difference between the crystal frequency and the read data frequency can be about 1%. In a removable media system the data may be written in one drive and read in another. The total difference between the read data frequency and the crystal frequency can be twice the rotational speed difference of the drive. For example, if the rotational speed variation of the drive is +1% when the disk is written, and then −1% when the disk is read, the read back data frequency will be 2% slower than the crystal frequency. Since a frequency difference becomes a phase ramp, the phase error will initially grow during read acquisition while the loop attempts to hold the phase error to zero (see Section 3.7). If the peak phase error which results exceeds the window tolerance, the loop will not capture within the desired acquisition time. The data will slip out of its proper window into the neighboring window and a series of error signals of the wrong polarity will result.

When the data rate is 5 Mbit/sec, the window period is 100 ns and the peak phase error allowed is ideally 50 ns. For a damping factor of 0.7, the peak phase error is given by:

$$\text{peak error} = (0.45)\,(\Delta\omega/\omega n)\,(200\text{ ns}/2\pi) + 0.21(\Delta\theta)$$

where $\Delta\theta$ is the worst case initial phase (50 ns). The $200\text{ ns}/2\pi$ term converts radians to seconds. $\Delta\omega$ is the

worst case initial frequency difference which is determined by the rotational speed variation of the drive. For a removable media drive with 1% variation:

$$\Delta\omega = (2\pi/200\text{ ns})\ (0.02) = 628\text{ Krads/sec}$$

The results are as follows

ωn = 600 Krads/sec → peak phase error = 25.5 ns

ωn = 500 Krads/sec → peak phase error = 28.5 ns

ωn = 400 Krads/sec → peak phase error = 33.0 ns

ωn = 300 Krads/sec → peak phase error = 40.5 ns

For ωn = 300 Krads/sec, with a damping factor of 0.7, the loop will just barely capture. If a removable media system were to use a 300 Krads/sec loop bandwidth, the capture range analysis should be performed very carefully since the numbers given here are very approximate. Capture performance improves for larger damping factors, however, total acquisition time increases.

There is no precise definition of phase lock. At the end of the preamble, there will be some residual phase error. The loop is locked if this phase error contributes an acceptable amount of margin loss during the reading of the address mark. It is recommended that the residual error be about 2 ns in a 5 Mbit/sec data rate system:

$$2.0\text{ ns}\ (2\pi/200\text{ ns}) = 0.062\text{ radians}$$

If the damping factor is 0.7, the following results

ωn = 600 Krads/sec → error = 0.062 at t = 7 μs

ωn = 500 Krads/sec → error = 0.062 at t = 9 μs

ωn = 400 Krads/sec → error = 0.062 at t = 12 μs

ωn = 300 Krads/sec → error = 0.062 at t = 17 μs

The total read acquisition time includes the initial phase acquisition as follows:

ωn = 600 Krads/sec → 7 μs + 1.7 μs
= 9 μs = 5.5 bytes

ωn = 500 Krads/sec → 9 μs + 2.0 μs
= 11 μs = 6.5 bytes

ωn = 400 Krads/sec → 12 μs + 2.5 μs
= 15 μs = 9.0 bytes

ωn = 300 Krads/sec → 17 μs + 3.3 μs
= 20 μs = 12.5 bytes

3.5.1. Crystal Acquisition

Analysis of the crystal acquisition time is similar to the read acquisition time. In the case of the National DP8465 data separator, however, a high bandwidth mode is provided to decrease the acquisition time. The high bandwidth is activated when $\overline{\text{SET PLL LOCK}}$ is deasserted. This increases the phase detector gain (increases the charge pump current). When the phase detector gain increases, both the loop bandwidth and the damping factor are increased. Loop performance is poor if the damping factor gets much larger than 1.0 and therefore the increase in loop bandwidth should be limited to the point where the damping factor is 1.0. This means that:

$$\omega n(\text{high track}) = \omega n(\text{low track})\ (1.0/0.7)$$

Repeating the acquisition analysis for the four bandwidths used before:

ωn = 600 Krads/s (1/0.7) = 857 Krads/s → acq = 5 bytes

ωn = 500 Krads/s (1/0.7) = 714 Krads/s → acq = 6 bytes

ωn = 400 Krads/s (1/0.7) = 571 Krads/s → acq = 7 bytes

ωn = 300 Krads/s (1/0.7) = 429 Krads/s → acq = 9 bytes

3.5.2. Margin Loss Due to PLL Response

Fast acquisition is desirable to minimize preamble lengths. However, the wider the loop bandwidth, the larger is the loop response to shifted data. Loop response to shifted data results in margin loss. The data is shifted by noise and other factors which contain no information about data frequency changes. When the loop responds to this bit shift it moves the windows for subsequent data bits thereby reducing the amount of shift allowed for these bits.

For a 5 Mbit/sec system with a maximum shifted early bit, the following formula gives the loop response:

$$\text{loss} = 40[1 - (\cos\sqrt{1-\zeta^2}\,\omega nt - \zeta/\sqrt{1-\zeta^2}\,\sin\sqrt{1-\zeta^2}\,\omega nt)\exp(-\zeta\omega nt)] \quad \text{Note 1.}$$

where 40 is the phase step (in ns) due to the early bit, ωn is the loop bandwidth, and ζ is the damping factor. The phase detector output is active for 40 ns and the amount of loss is determined by setting t = 240 ns which is the time to the far edge of the next window where a bit may appear.

The ωn and ζ for this calculation will be the same as used in the data acquisition analysis as long as t does not exceed 2 VCO cycles. If t is much greater than 2 cycles, the effective phase detector gain is reduced and the ωn is reduced also (see design example of ωn formula). This calculation also assumes that the second pole in the filter is well outside the loop bandwidth.

ωn(max freq data) = 600 Krads/s → loss = 7.7 ns

ωn(max freq data) = 500 Krads/s → loss = 6.4 ns

ωn(max freq data) = 400 Krads/s → loss = 5.2 ns

ωn(max freq data) = 300 Krads/s → loss = 3.9 ns

There are techniques for reducing this margin loss without heavily impacting the acquisition performance. See Section 3.7 for details.

DESIGN EXAMPLE FOR 5 Mbit/s DATA RATE

Although there is no real standard, most of the track formats for small Winchesters are using about 12 bytes of preamble. There is a formated gap after the data ECC field but it cannot be assumed that any of this gap is available for PLL acquisition. Some of this gap will be lost when the sector is updated due to rotational speed variation and the remainder is required for write-to-read recovery of the read channel.

Note 1: Phaselock Techniques; Floyd M. Gardner, Second Edition, John Wiley & Sons; pg. 48.

As discussed, in a soft sectored disk drive, acquisition to the crystal, as well as acquisition to the read data, must occur within the preamble. (In a hard sectored drive some of the preamble may be lost to uncertainty in sector pulse detection but usually most of the preamble is available for data acquisition.) For this reason, the loop bandwidth during acquisition must be in the 600 Krads/sec range for a soft sectored drive. The 600 Krads/sec bandwidth gives 5 byte crystal acquisition and 5.5 byte data acquisition. The margin loss can be held to about 6.0 ns + 7.7 ns = 13.7 ns with the DP8465-3. (This margin loss can be reduced if a longer preamble/lower bandwidth were used or by using some of the techniques discussed in the 10 Mbit/sec design example.)

For the DP8465 with $\overline{\text{SET PLL LOCK}}$ asserted:

$$\omega n = \sqrt{(2.5)\,(F_{VCO})/(N)\,(C1)\,(R_{rate})}$$

Rrate should be set at 820Ω since the current in this resistor does have a small effect on the VCO stability and 820Ω has been determined to be the optimum value. F_{VCO} is the center frequency of the VCO in hertz. F_{VCO} is 10 MHz for 5 Mbit/sec data rates. N is the number of VCO cycles per data bit. MFM preamble data has 2 cycles per data bit. This gives:

$$C1 = (2.5)\,(F_{vco})/(N)\,(Rrate)\,(\omega n^2)$$
$$= (2.5)\,(10E6)/(2)\,(820)\,(3.6E11)$$
$$= 0.042E\text{-}6 \text{ (use } 0.039\ \mu F)$$

C1 should be an ultra-stable monolithic ceramic capacitor or equivalent timing quality capacitor.

In the data field, the MFM data frequency can be half the preamble frequency. This means that N = 4 in the bandwidth equation. This reduces the bandwidth by $1/\sqrt{2}$:

$$\omega n \text{ (min)} = (1/\sqrt{2})\,(6.25.1 \text{ Krads/s}) = 442.1 \text{ Krads/s}$$

where 625.2 Krads/sec is the computed bandwidth in the preamble with C1 = 0.039 μF. Since there should be no mechanical resonances anywhere near this frequency, the loop will be able to track the data.

The damping factor should be about 0.5 when ωn is minimum. Response to bit shift is minimized when the damping factor is small; however, if the damping factor drops much below 0.5 the system tends to be oscillatory (underdamped):

$$\zeta = (\omega n)\,(R1)\,(C1)/2 \rightarrow R1 = (2)\,(\zeta)/(\omega n)\,(C1)$$
$$R1 = 1/(442.1E3)\,(0.039E\text{-}6) = 58 \text{ (use } 56\Omega)$$

The actual damping during acquisition is then:

$$\zeta(acq) = (625.2)\,(56)\,(0.039E\text{-}6)/2 = 0.68$$

The linear approximation used to predict loop performance assumes that the phase detector output is constantly proportional to the input phase difference. In reality, the phase detector output is a pulse applied for a period of time equal to the phase difference. The function of C2 is to smooth the phase detector output over the cycle. C2 adds a second pole to the filter transfer function as discussed in section 3.7. This pole should be far enough outside the loop bandwidth that its phase and amplitude contribution is negligible to the loop bandwidth. If:

$$C2 = C1/50 = 789 \text{ pF (use } 820 \text{ pF)}$$

the acquisition performance and the margin loss are not significantly changed from the predictions. If a larger C2 is used, the margin loss can be reduced at the expense of the acquisition. This may be desirable for some systems. See section 3.7 for discussion of the function of C2.

Note 1: Ibid.

The final loop component is R_{boost}. When $\overline{\text{SET PLL LOCK}}$ is deasserted, R_{boost} is in parallel with R_{rate} to set the charge pump current. If the parallel combination of R_{rate} and R_{boost} is called R_p:

$$\omega n \text{ (high)} = \sqrt{(2.5)\,(F_{VCO})/(N)\,(C1)\,(R_p)}$$

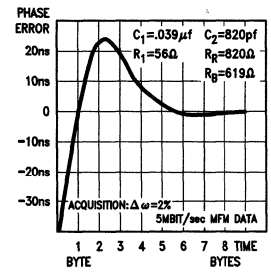
Since the total set current, I_{set}, into the charge pump may not exceed 2 mA, the parallel resistance R_{boost} and R_{rate} (R_p) should also be restricted to:

$$R_p \geq V_{be}/I_{set} = 0.7V/2 \text{ mA} = 350\Omega$$

Solving for R_{boost}:

$$R_{boost} = (R_p)\,(R_{rate})/(R_{rate} - R_p)$$
$$= (350)\,(820)/(820 - 350) = 610.9 \text{ (use } 619\Omega)$$

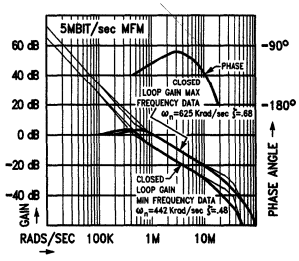
Remember, R_{boost} is switched-in whenever $\overline{\text{SET PLL LOCK}}$ is deasserted. This design example assumes that $\overline{\text{SET PLL LOCK}}$ is deasserted whenever READ GATE is deasserted.



TL/F/8663–59

ωn (acquisition) = 625 Krads/s ζ = 0.68
Lock time ≈ 5 bytes.
ωn (write mode) = 957 Krads/s ζ = 1.04
Lock time ≈ 4 bytes.
ωn (min data) = 442 Krads/s ζ = 0.48

FIGURE 3.14a. 5 Mbit/sec Design Example



TL/F/8663–60

FIGURE 3.14b. 5 Mbit/sec Design Example

DESIGN EXAMPLE FOR 10 MBIT/SEC MFM DATA RATE

At 10 Mbit/sec the window period is reduced to 50 ns and margin loss becomes a much more important design parameter than at 5 Mbit/sec. The total bit shift allowed is half the window (25 ns). If the National DP8465-3 is used, there will be a maximum static loss (independent of the loop filter components) of 6 ns. This part could be used as the starting point for a 10 Mbit/sec design.

The margin loss due to loop response should be kept as low as possible. The maximum allowed bit shift with the DP8465-3 is (25 ns − 6 ns) or about 19 ns. The loss approximation formula is then:

$$\text{loss} = 19[1 - (\cos\sqrt{1-\zeta^2}\,\omega nt$$
$$- \zeta/\sqrt{1-\zeta^2}\,\sin\sqrt{1-\zeta^2}\,\omega nt)\exp(-\zeta\omega nt)] \quad \text{Note 1.}$$

where 19 is the phase step (in ns) due to an early bit. Evaluation should be done with t = 120 ns:

ωn = 600 Krads/s and ζ = 0.66 → loss = 1.9 ns

This 1.9 ns loss should be acceptable for most designs. The total loss would then be 6 ns + 1.9 ns = 7.9 ns for a total window of 34.2 ns with the DP8465-3. As discussed in Section 3.7, the actual margin loss due to loop response will be less than 1.9 ns due to the roll off of the second pole in the loop filter.

The acquisition performance of a 600 Krad/s loop was analyzed in the 5 Mbit/sec design example. With a 2% frequency difference between the read data and the crystal, the peak phase error was about 24 ns. At 10 Mbit/sec $\Delta\omega$ is doubled and the conversion term becomes 100 ns/2π. The initial phase misalignment is 25 ns. The peak phase error is then just over 20 ns. This is pushing the capture range of the DP8465-3 since its allowed shift is only 19 ns. If the system is fixed media with 1% or better rotational speed variation there is no problem since the peak phase error is reduced by a factor of two in this case. If capture is needed at 2%, $\overline{\text{SET PLL LOCK}}$ can be deasserted during read acquisition as discussed later.

Notice that the 5 Mbit/sec lock time was given in bytes (i.e. 5.5 bytes). At 10 Mbit/sec the read acquisition time will remain about the same but that means that the number of bytes is doubled to 11 bytes. This read acquisition time can be reduced without increasing the margin loss if the disk controller will hold $\overline{\text{SET PLL LOCK}}$ deasserted (high) during read acquisition. This increases the damping factor to 1.04 and the bandwidth to 940 Krads/sec. The result is a read acquisition time of 8 bytes. The crystal acquisition time is also 8 bytes so the total preamble length is 16 bytes in a soft sectored disk drive. (This assumes zero monitoring overhead. If x bytes are required to determine that the read data is not preamble, the total preamble length would be 16 + x bytes.) Of course the price paid for this reduction in acquisition time is that the controller must now control the $\overline{\text{SET PLL LOCK}}$ input during acquisition.

The calculation of the loop components is similar to the 5 Mbit/sec data rate example:

$C1 = (2.5)(F_{VCO})/(N)(R_{rate})(\omega n^2)$

$= (2.5)(20E6)/(2)(820)(3.6E11)$

$= 0.085E\text{-}6$ (use 0.082 μF (5%))

C1 should be an ultra-stable monolithic ceramic capacitor or equivalent timing quality capacitor. Evaluation of the bandwidth formula with C1 = 0.082 μF gives:

$\omega n = \sqrt{(2.5)(20E6)/(2)(820)(0.082E\text{-}6)} = 609.8$ Krads/s

The target damping factor at this bandwidth is 0.66:

$R1 = (2)(\zeta)/(\omega n)(C1)$

$= (2)(0.66)/(609.8E3)(0.082E\text{-}6) = 26.4$ (use 27Ω)

The total set current into the charge pump through the parallel combination of R_{boost} and R_{rate}, (R_p), must not exceed 2 mA. For $I_{set} \leq Vbe/R_p$, then:

$$R_p > Vbe/I_{set} = 0.7V/2\text{ mA} = 350\Omega$$

Solving for R_{boost}:

$R_{boost} = (RP)(R_{rate})/(R_{rate} - R_p)$

$= (350)(820)/(820-350) = 610.6$ (use 619Ω)

This design example assumes that R_{boost} is switched in during crystal acquisition and data acquisition.

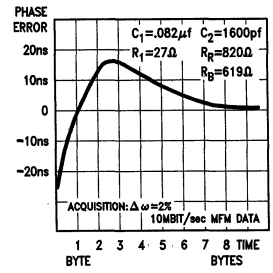
If the margin loss due to loop response of 1.9 ns is acceptable and acquisition performance is to remain unaffected:

$C2 \ll C1/50 = 0.082\ \mu F/50 = 1640$ pF (use 1600 pF)

A smaller value of C2 is chosen for this example since the 1.9 ns of margin loss seems very acceptable whereas it is undesirable to impact the capture range or the acquisition time. Larger values of C2 may be used, however, the additional pole in the filter may begin to affect frequencies inside the loop bandwidth. The major impact of large C2 is on the capture range although some degradation in read acquisition time can be seen for larger values of C2. See section 3.7 for further discussion.

3.6 COMMENTS ON OTHER CODES

MFM is a 1,3 RLL code. This means that a minimum of one empty window will occur between two windows which each contain a disk data bit, and that a maximum of three empty windows will occur between windows which each contain a disk data bit. The most popular of the newer RLL codes are the 2,7 codes, in which there are a minimum of 2 and a maximum of 7 windows between windows containing a disk data bit. Although the ratio of encoded disk data bit positions (windows) on the disk to non-encoded data (NRZ) bits for both MFM and 2,7 code is 2:1, the two codes differ in the actual number of recorded pulses required to store a given number of NRZ bits (their NRZ-bit versus disk-bit ratio, or efficiency). The 2,7 code requires fewer recorded bits than MFM on average for disk encoding of the same amount of information and has a 50% larger minimum bit spacing than MFM. These allow, on a given disk, a theoretical increase in data storage of 50% when 2,7 encoding is chosen over MFM while the minimum flux transition spacing is kept constant.



TL/F/8663–61

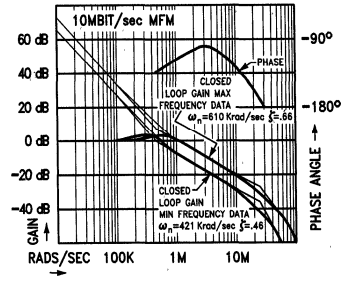
Controller must keep $\overline{\text{SET PLL LOCK}}$ deasserted for 8 bytes after READ GATE is asserted.

ωn (acquisition) = 933 Krads/s ζ = 1.03

ωn (min data) = 431 Krads/s ζ = 0.48

ωn (max data) = 610 Krads/s ζ = 0.66

FIGURE 3.15a. 10 Mbit/sec Design Example

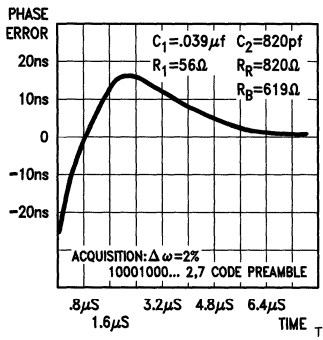


TL/F/8663–62

FIGURE 3.15b. 10 Mbit/sec Design Example

The impact of a 2,7 code on the data separator is significant. Loop bandwidth is dependent upon the sample rate into the phase detector. With MFM it has been seen that the bandwidth in minimum frequency data is $1/\sqrt{2}$ times the bandwidth in maximum frequency data. This is a ratio of 1:0.707. In a 2,7 code the ratio is $\sqrt{3/8}$ or 1:0.612. The damping factor follows the bandwidth so a 2,7 code system must be more carefully designed to avoid underdamped or overdamped response.

Another complexity of 2,7 codes is that most systems are not using the maximum frequency data in the preamble. The 100100 .... encoded data does not decode to a data pattern with byte alignment. The 10001000 ... encoded data pattern decodes to all ones byte aligned data and is being used more often. The lower frequency data increases the read acquisition time and increases the probability of harmonic lock. Channel induced pulse pairing (from channel asymmetry), coupled with an initial phase alignment which puts the data bit at the extreme window edge, may allow the loop to stabilize out of phase. There are two techniques which are used to eliminate this problem. The first is to start the VCO in phase with the read data (zero phase start-up). The second technique is to perform phase and frequency comparison (i.e. do not window the data) during read acquisition. The second technique is used on the National DP8462 data separator which is specifically designed for 2,7 codes.



Controller must keep $\overline{\text{SET PLL LOCK}}$ deasserted for 6 μs after READ GATE is asserted.

ωn (acquisition) = 956 Krads/s ζ = 1.04

ωn (min data) = 442 Krads/s ζ = 0.48

ωn (max data) = 722 Krads/s ζ = 0.79

FIGURE 3.16a. 10 Mbit/sec 2,7 Code Example

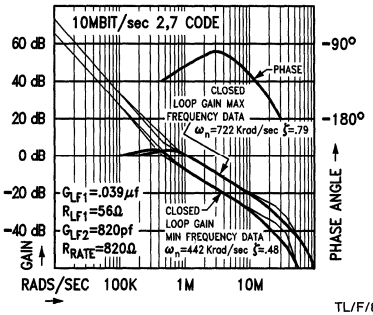


FIGURE 3.16b. 10 Mbit/sec 2,7 Code Example

Note 1. Ibid.

In conclusion, loop filter design for 2,7 codes is more difficult than for MFM. In hard sectored drives, or pseudo hard sectored drives with dc erased gaps, the problem is simply the wide range of damping factors and can be easily solved.

3.7 LOOP FILTER DETAILS

Time domain response for a second order feedback control system is well known. The response to a phase step and a frequency step is shown in the graphs in *Figure. 3.17.* The phase locked loop system is normally designed to have a damping factor between 0.7 and 1.0 during acquisition so these curves show the performance boundaries. The equations for the phase step response are:

$$\zeta < 1 \rightarrow \theta(t) = \Delta\theta(\cos\sqrt{1-\zeta^2}\,\omega nt - \zeta/\sqrt{1-\zeta^2}\,\sin\sqrt{1-\zeta^2}\,\omega nt)\exp(-\zeta\,\omega nt)$$

$$\zeta = 1 \rightarrow \theta(t) = \Delta\theta(1-\omega nt)\exp(-\omega nt) \qquad \text{Note 1}$$

The equations for the frequency step response are:

$$\zeta < 1 \rightarrow \theta(t) = (\Delta\omega/\omega n)\,(1/\sqrt{1-\zeta^2}\,\sin\sqrt{1-\zeta^2}\,\omega nt)\exp(-\zeta\,\omega nt)$$

$$\zeta = 1 \rightarrow \theta(t) = (\Delta\omega/\omega n)\,(\omega nt)\exp(-\omega nt) \qquad \text{Note 1.}$$

These equations were used to derive the margin loss due to bit jitter (i.e. phase steps) as well as the acquisition performance in the previous design examples. The second order time domain response is the usual starting point for loop filter design.

Disk data separation is complicated by the fact that the input data stream is not a single frequency. Missing data bits must not be allowed to generate error signals to the loop since the loop bandwidth would have to be set unacceptably low to filter out this erroneous information. As a result, most phase detectors for disk data separators do not generate a continuous voltage dependent upon the input phase difference. The discontinuity in the input data stream is dealt with by only generating a phase detector output when a data bit has arrived and only during the period of time corresponding to the input phase difference.

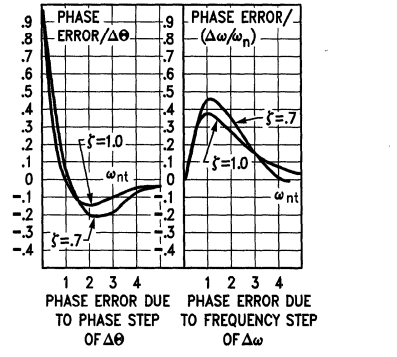
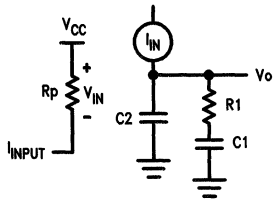


FIGURE 3.17. Phase Error as a Function of Damping and Phase Step

There are some problems which result from a pulsed phase detector output. First, oscillator response can be affected by the phase relationship between these pulses and its internal charging and discharging cycle. Second, the response of the oscillator occurs in a quantum jump when the error pulse is generated, reducing margin for later bits. These problems are reduced by adding one or more poles to the loop filter. These poles smooth the phase detector output and reduce the loop response to bit jitter. As long as the added poles do not add significant gain loss and phase shift within the loop bandwidth, the system time domain response will not differ appreciably from a pure second order feedback control system.

The voltage output of the National DP8465 phase detector is converted to a current which is sourced into the filter:



TL/F/8663–66

FIGURE 3.18 Charge Pump and Loop Filter Equivalent Circuitry.

If Zf is the input impedance of the loop filter:

$$Zf = (1/sC2) \,//\, (1/sC1 + R1) =$$

$$(1 + sC1R1)/sC1(sC2R1 + C2/C1 + 1)$$

$$Vo = (Iin)(Zf) =$$

$$(Vin/R_p)(1 + sC1R1)/sC1(sC2R1 + C2/C1 + 1)$$

$$Vo/Vin = (sC1R1 + 1)/sC1R_p(sC2R1 + C2/C1 + 1)$$

The effect of C2 is to introduce a pole into the transfer function of the loop filter. The pole location is where $s = (1 + C2/C1)/C2R1 = 1/C2R1$ if $C2 << C1$.

The open loop gain is the product of the phase detector, the filter, and the oscillator transfer functions. The phase detector transfer function is simply a constant, K_{pd}, in volts/radian. The oscillator transfer function is K_o/s where the 1/s term represents phase as the integral of frequency and K_o is in units of radians/(volt x sec). The open loop transfer function is then:

$$G(s) = (K_oK_{pd})\,[st2 + 1]/[st1(st3 + 1)s]$$

where $t1 = C1R_p$, $t2 = C1R1$, and $t3 = C2R1$. The transfer function has a zero at 1/t2, a pole at 1/t3, and two poles at the origin.

Note: For the National DP8465 the K_oK_{pd} product is (2.5) (F_{VCO})/N where F_{vco} is the oscillator frequency in Hertz and N is the number of oscillator cycles between data bits.

In the frequency domain, the open loop gain falls at 40 dB/decade until equal to $K_oK_{pd}/t1$ at $j\omega = 1$. The 40 dB/decade slope continues until the zero at $j\omega = 1/t2$. At the zero the slope changes to 20 dB/decade. The slope returns to 40 dB/decade when the pole breaks at $j\omega = 1/t3$. The phase shift begins at −180 degrees and asymptotically approaches −90 degrees. The phase is equal to −135 degrees at $j\omega = 1/t2$. The phase plot turns around and starts back toward −180 degrees as $j\omega$ approaches 1/t3 such that at $j\omega = 1/t3$ the phase equals −135 degrees.

If a second pole were in the filter around $j\omega = 1/t3$, the phase would equal −225 degrees at $j\omega = 1/t3$ and stability would require that the gain be below 0 dB before the phase reached −180 degrees. This sometimes limits how closely the poles can be moved to the loop bandwidth. When the gain is 0 dB, the difference between the actual phase shift and 180 degrees is referred to as the phase margin. The open loop phase margin is related to the damping factor of the second order system. Note that there is a pole in the buffer amplifier of the DP8465 between the filter and the VCO. This pole is at 5 MHz or about 31.4 Mrads/s and could affect the phase margin in a very wide band loop.

The additional pole in the loop filter helps to improve read margin because it lowers the loop gain at the frequency of the bit jitter. The fundamental frequency content of the bit jitter is slightly below the bit frequency since the pump up error begins before the end of the window and the pump down error ends after the end of the window. At 5 Mbit/sec, the bit jitter frequency is 1/(200 ns + 80 ns) = 22.4 Mrads/sec for 40 ns bit shifts. The 615 Krads/sec loop designed earlier for 5 Mbit/sec data would have a gain of −28 dB at $j\omega = 22.4$ Mrads/s if C2 were not in the loop. With C2 = 820 pF, the gain is reduced to about −31 dB at the bit jitter frequency. If the additional pole were added at $j\omega = 6.15$ Mrads/s (10 ωn) the gain would be −38 dB at the bit jitter frequency for an extra 10 dB of noise rejection. C2 would be 0.0027 μF in this case.

It is difficult to analytically predict the effect of C2 on the acquisition performance. Since C2 is moving close to the loop bandwidth, the system behavior is not purely second order. There are some computer programs which allow time domain response to be predicted for third order systems but normally it is not necessary to use these tools. It is usually sufficient to start with a second order analysis and experimentally measure the system performance as the third pole (or poles) is brought closer to the loop bandwidth.

CHAPTER 4 DP8466 Disk Data Controller Overview

4.0 INTRODUCTION

National's Disk Data Controller (DDC) chip, DP8466, performs many of the functions in the disk data electronics path of either disk controllers or intelligent disk drives. The primary function of the chip is to correctly identify the selected sector on disk and then to transfer that sector's data to or from memory.

The DDC performs serialization and deserialization of disk data, CRC/ECC generation, checking and correction, data buffering with a 16-word (32-byte) FIFO, and single or dual channel DMA addressing. It can write NRZ or MFM encoded data to the disk. The data separation required in the disk data path electronics can be obtained by using one of National's Data Separator chips, DP8460 or DP8461/5. If 2,7 is used instead of MFM, 2,7 ENDEC chip could be used in conjunction with the DP8462 2,7 Data Synchronizer. The DDC is fabricated using the dual layer metal 2μ microCMOS process, which allows complex functions to be implemented with high operating speeds and modest power consumption. Internal gate delays of less than 2 ns allow the DDC to function with disk data rates up to 25 Megabits/sec. This enables the DDC to be used not only with $3\frac{1}{2}$-inch, $5\frac{1}{4}$-inch and 8-inch drives, either Winchester or Floppy (or both), but also with high-end drives such as 14-inch Winchester drives, vertically recorded drives, and optical drives.

The DDC interfaces with drives compatible with the ST506, ST412HP, ESDI, SMD and other interfaces. Also the DDC may instead be part of an intelligent disk drive that has a SCSI (SASI), or an IPI type interface. Refer to chapter one where the block level boundaries of the various disk interface standards are shown.

4.1 THE DDC ARCHITECTURE AND BASIC OPERATION

An architectural block diagram of the DDC is shown in *Figure 4.1.* The 64 internal registers consist of control, command, pattern and count registers. These registers are initially preloaded with information such as header or synch bytes, ECC polynomial bytes, preamble or postamble patterns, or address marks (for soft sectored drives) etc. Some of the registers will be programmed each time the DDC starts an operation, for example the command register.

The DP8466 has a range of commands that enable reading and writing of both data, and header fields, checking for header fields, formatting with either hard or soft sectored formats, and aborting. Each of these operations can be performed in various modes and an abundance of formats. Most operations can be performed as single or multi-sector operations.

In a typical disk read or write operation, the desired sector (where the data information is to be read from or written to) is first located by comparison of the header bytes. To achieve this comparison, the incoming serial data from the external data separator is deserialized into byte-wide data that is fed both to a comparator and FIFO. The comparator checks the address mark (if present) and the synch bytes to align the incoming bytes. Once the incoming data stream has been byte aligned, header comparison for the desired sector then begins. As each header byte is deserialized it is compared with the next preloaded header byte. If any of the header bytes do not match, the desired sector has not been located, but the DP8466 still performs a CRC/ECC check on the header and waits for the ID segment of the next

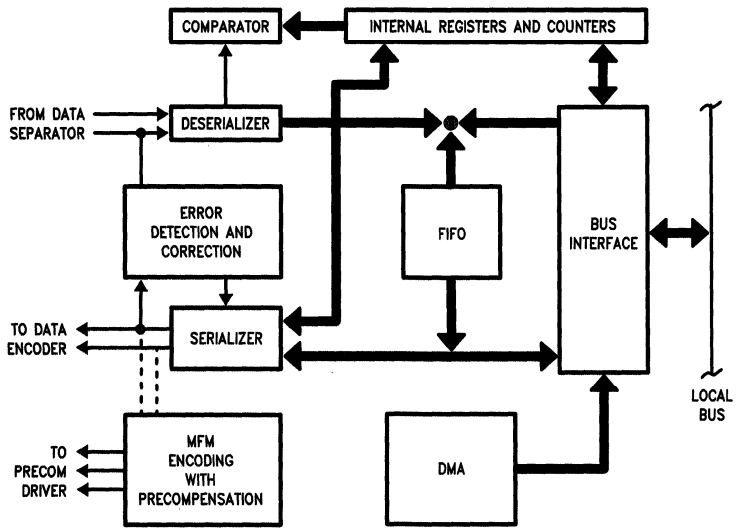


FIGURE 4.1. The DDC Data Path Architecture

sector. If after two disk revolutions the header is not found the DDC will abort the operation. Once a header match is detected, the DDC prepares to transfer data to or from the data segment of the sector.

1. When writing a data segment to the disk, the DDC first inserts the preamble pattern field, address mark (for soft sectored drives) and synch fields, each byte repeated a specified number of times. These fields are followed by the data field bytes that are provided sequentially through the FIFO and external memory. Internal CRC or ECC (or external ECC) check bits are generated from the bits in the data field and subsequently appended to it. The write operation ends with the postamble. As each byte is serially transmitted, the next byte becomes available to be serialized and output. The serial output may be either NRZ data with the associated write clock or MFM encoded data.

2. When reading a data segment from the disk, the DDC deserializes the incoming data and byte aligns with the address mark (if present) and synch fields using the comparator. It then transfers the data field bytes into FIFO. At the same time it checks incoming serial data using an internal CRC code or ECC code.

For both read or write operations, disk data goes to or from the internal FIFO. Once the FIFO has filled or emptied to the selected threshold level, data may be transferred to or from the external buffer memory in the selected burst length by means of a DMA channel. A second DMA channel is available to transfer data to or from buffer memory to the system (for systems that utilize a buffer memory).

If the operation terminates properly an interrupt is issued, and the user may check status. If an error results during the operation the DDC will also interrupt the microprocessor, and the user must determine the appropriate action.

The DDC can be configured in three different modes; peripheral, master and slave. In the peripheral mode, the microprocessor accesses internal register to read or write data. The DDC acts like a peripheral when it is being configured, and when the microprocessor issues a command. During the execution of a command and when the on-chip DMA has been granted access to the bus for local and remote transfers, the DDC goes into its master mode, and becomes bus master. If during the command an external DMA controls data transfer the DDC will go into a slave mode.

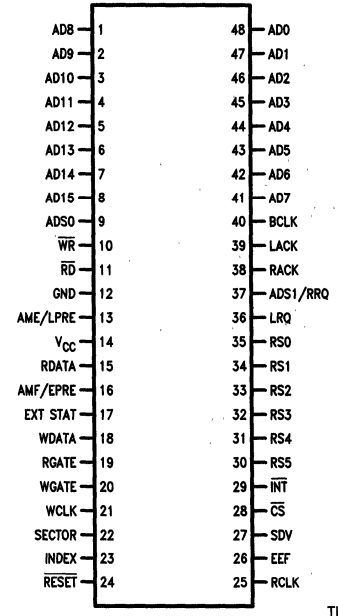
4.2 PIN ASSIGNMENT AND DESCRIPTION

In this section a complete pin description is presented. The pin assignment diagrams are shown in *Figure 4.2(a)* and *4.2(b)*. Specific timing information for these signals can be found in the DP8466's Datasheet.

4.2.1 Bus Interface

Chip Select ($\overline{CS}$): When the DDC is in the peripheral mode the chip select signal must be asserted low to access enable microprocessor access. In the peripheral mode pins RS0–5 are address inputs and pins AD0–7 are set for 8-bit transfer of data between the DDC and microprocessor. $\overline{CS}$ has no effect if on-chip or external DMA is performing a transfer. (DDC in slave or master modes.)

Bus Clock (BCLK): The DDC uses BUS CLOCK input as the reference clock when the DDC is bus master. It is used only during RESET and DMA operations and is independent of the disk data rate. BCLK may be the microprocessor clock and must be at least 1/4 the rate of READ CLOCK, RCLK.



TL/F/8663–68

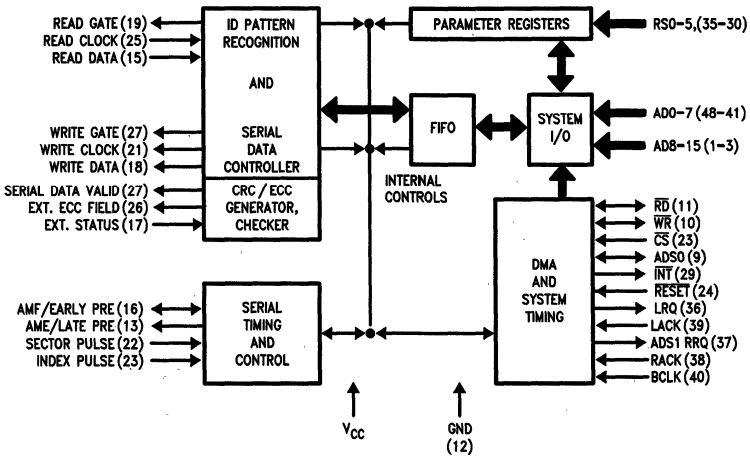
FIGURE 4.2(a). The DDC (DP8466) Connection Diagram

Address/Data (AD0–7): This 8-bit data/address bus port has one microprocessor associated function and four memory transfer associated functions. When the DDC is in the peripheral mode and $\overline{CS}$ is set low, this port transfers data between the internal sections of the DDC and the microprocessor. When external DMA is active (i.e. the DDC is in slave mode) with LACK (local acknowledge) set low, data bits D0–7 are transferred between the FIFO and memory.

When the DDC is controlling the bus (i.e. when on-chip DMA is active), the AD0–7 bus is multiplexed between DMA address and FIFO data bits. Using the single DMA mode, A0–7 are issued on this port as are A16–23. When using dual DMA, these lines are used to transfer both the local and remote DMA address bits, A0–7. In either dual or single channel mode, the data bits D0–7 are transferred between FIFO and external memory through this port.

Address/Data (AD8–15): This 8-bit I/O port has four memory transfer functions (in the peripheral mode with $\overline{CS}$ low, these pins remain indeterminant low impedance). In the slave (external DMA active) mode with LACK set low, data bits D8–15 are transferred between the FIFO and memory when 16-bit transfers are enabled. When the DDC is controlling the bus, (master mode) it issues address bits A8–15 on this port and can also issue address A24–31 if it is in single channel DMA mode.

Register Select (RS0–5): In the peripheral mode, these 6 inputs are used to select the internal registers to be accessed by the microprocessor. These inputs feed "fall through" latches that are controlled by the ADS0 input. The RS0–5 inputs fall through and are decoded by the DDC when the input level on ADS0 pin is high. The RS0–5 inputs are stored on the falling edge of ADS0. This enables easy connection to either multiplexed or non-multiplexed buses.



TL/F/8663-69

FIGURE 4.2(b). The DDC Block Diagram with Pin Assignment

Address/Data Strobe 0 (ADS0): This I/O strobe has two functions. In the peripheral mode, ADS0 becomes an input and may be used as a microprocessor address strobe input. In this mode when ADS0 is high, the address bits on RS0–5 enter the Register Select Latch and are latched on low going transition.

With the bus under DDC control, ADS0 becomes an output that issues the address strobe to external memory at the start of the DMA transfer cycle. The low going transition of ADS0 coincides with the DMA address bits A0–15 being valid on pins AD0-15. (Note: ADS0 when an output will still latch "data" into the RS0–5 latches. Normally this is random data and is of no consequence since $\overline{CS}$ is high. However when the system wants to access the DDC after a DMA, the proper address must be loaded into the latches or ADS0 must be high prior to $\overline{CS}$ going low.)

Address/Data Strobe 1 / Remote Request (ADS1/RRQ): This output pin can be configured to have one of two functions. If the DDC has been configured for 32-bit address single channel DMA, the pin becomes ADS1. This address strobe is issued either at the start of the very first memory transfer cycle of a new disk operation or when the lower 16 address bits have just rolled over. In either case the address on pins AD0–15 is A15–31 at the instance of low going transition of ADS1.

If the DDC is configured to perform remote DMA transfers in its dual channel mode, the pin becomes RRQ, or Remote DMA Request. The DDC will assert the RRQ high whenever it is ready to transfer data between buffer memory and the system I/O port. The RRQ will be reset at the end of the selected burst transfer length.

Remote DMA Acknowledge (RACK): This input pin must be asserted high after RRQ has been set high, when the external I/O device is ready to transfer data. Data will be transferred between external buffer memory and the remote I/O device until the DDC sets RRQ low, or until RACK is set low externally. If RACK is removed during a transfer any cycle in progress at this time will complete.

Local DMA Request (LRQ): This output pin is low when no data transfer is required between FIFO and the external buffer memory. The DDC asserts LRQ high when FIFO requires data to be transferred to or from the external buffer memory for both dual and single DMA mode. LRQ will return low during the last cycle of the burst transfer or on emptying or filling up the FIFO.

Local DMA Acknowledge (LACK): Once the DDC has set LRQ high, and the external circuitry subsequently sets the LACK input high the DDC becomes bus master. Data transfers between the on-chip FIFO and external memory will now proceed until the DDC sets LRQ low, or until LACK is set low externally, in which case any cycle in progress at this time will complete.

$\overline{\text{Read}}$ ($\overline{\text{RD}}$): The $\overline{RD}$ strobe I/O pin has two functions. In the peripheral mode $\overline{RD}$ is an input that when low causes data from a DDC register (selected through pins RS0–5) to be output on pins AD0–7 when CS is low. Pins AD0–7 will be high impedance before and after the READ strobe.

When the DDC has bus control, with either LACK or RACK active, the $\overline{RD}$ strobe is an active low output from the DDC to be used by external memory. It enables data to be transferred from the selected memory location to either the on-chip FIFO for local transfers or to an external I/O device for remote transfers.

$\overline{\text{Write}}$ ($\overline{\text{WR}}$): The $\overline{WR}$ strobe I/O pin has two functions. In the peripheral mode WR is an input that when set low will cause data present on pins AD0–7 to be loaded into a DDC register (selected through pins RS0–5, and when $\overline{CS}$ is low).

When the DDC has bus control, with either LACK or RACK active, the $\overline{WR}$ strobe is an active low output used to write data to the selected memory location from either the on-chip FIFO for local transfers or from an external device for remote transfers.

Interrupt (INT): The INT output is set active low whenever the DDC wishes the controlling microprocessor to check status. It is set high when the microprocessor services the interrupt by setting the $\overline{CS}$ input low and reading the Status Register.

Reset: The RESET pin is an input that is normally set high. When RESET is set low, the DDC goes into the internal reset mode. It clears the FIFO contents, the Status and Error registers and also deactivates LRQ, RRQ, WRITE GATE and READ GATE.

4.2.2 Disk Interface

Write Gate (WGATE): When the DDC writes data to the disk during either a Write Data, Write Header or Format operation, it asserts the WRITE GATE output pin active high at the start of the operation. The transition coincides with the first WRITE DATA bit being issued. WRITE GATE remains high until either the last data bit of the sequence to be written has ended, or when the DDC aborts or resets.

Write Data (WDATA): During a write disk operation, this pin outputs serial disk data. The DDC can be configured to output either NRZ or MFM encoded data on the WRITE DATA pin. If NRZ data is selected its bit rate has the same period as the WRITE CLOCK output. When the DDC has been configured to issue MFM encoded data to the disk, data pulses will be output on WRITE DATA as determined by the MFM encoding rules. In either configuration, when WRITE GATE is inactive low, so is WRITE DATA.

Write Clock (WCLK): When the DDC is configured to write NRZ data, a synchronous clock is provided at the WRITE CLOCK output. The WRITE CLOCK frequency is the same as READ CLOCK. When the DDC is configured to output MFM encoded data, clock information is not needed however this output will still toggle.

Address Mark Found/Early Precompensation (AMF/EPRE): This pin has two modes. When the DDC is configured to read from a soft sectored disk, this pin is ADDRESS MARK FOUND, an active high input. Normally this input will be low but whenever external circuitry detects an Address Mark (such as a missing clock, or blank information) AMF should go high for at least one period of READ CLOCK. This will indicate to the DP8466 that a valid address mark has been located.

When the DDC is configured to write MFM encoded data, this pin becomes the output EARLY PRECOMPENSATION. When it is high, the MFM pulse appearing on the WRITE DATA output requires early precompensation. When low, the MFM pulse does not require early precompensation.

If both functions are being used in the system, WRITE GATE is used to determine the function of this pin. When WRITE GATE is active high, this pin is an LPRE output, otherwise it is an AMF input. External demultiplexing circuitry can be used.

Address Mark Enable/Late Precompensation (AMF/LPRE): This pin has two modes of operation depending on whether NRZ or MFM data is written to the disk. When the DDC is configured to write NRZ data on a soft sectored disk, this output pin is ADDRESS MARK ENABLE. AME is normally low and will remain low when the DDC is configured for a hard sectored disk (bit HSS in disk format register is set). On the other hand, for the soft sector configuration, the DDC will set AME active high during the time any Address Mark byte is serially output on WRITE DATA pin.

When the DDC is configured to write MFM encoded data, this pin becomes LATE PRECOMPENSATION output. In this configuraton if LATE PRECOMPENSATION is high, then late precompensation is required on the MFM pulse being output on WRITE DATA. If LATE PRECOMPENSATION is low, late precompensation is not required. If both EARLY PRECOMPENSATION and LATE PRECOMPENSATION output pins are set low, no precompensation is required.

Read Gate (RGATE): When the DDC is set to read the disk, such as in a Read Header, Compare Header, Ignore Header, Read Data or Ignore Data operation, READ GATE will go active high. This informs external data separator that it can begin locking on to incoming disk data. If the data separator fails to achieve locking, the DDC will set READ GATE inactive for 18 bit times before another locking attempt is made. READ GATE is also set inactive either at the end of the specified operation or if the DDC aborts or resets.

Read Data (RDATA): Once READ GATE has been set active high and external circuitry has locked on to the incoming encoded disk data, the encoded data must be separated into clock and NRZ data. The NRZ data connects to the READ DATA input of the DDC, and is clocked into the DDC on the positive edge of READ CLOCK. When READ GATE is set low, the READ DATA input will be ignored.

Read Clock (RCLK): READ CLOCK is a clock input that may have slightly differing frequencies, depending on the READ GATE control pin. When READ GATE is inactive, this clock should be derived from either a servo clock or a crystal clock to produce a clock with a period close to the bit rate of the disk data. After READ GATE has been set active, and external circuitry has locked on to the incoming encoded disk data, the READ CLOCK input must switch frequency (without any short pulses or glitches) to a period identical to the READ DATA signal.

Sector Pulse (SPULSE): In hard sectored drives the SECTOR PULSE input goes high as the start of each sector passes under the drive head. Once the DDC detects this high signal (for at least one period of the READ CLOCK input), it interprets this to indicate a sector operation can begin. In soft sectored drive there is no sector pulse and the start of each sector must be indicated by an Address Mark byte or bytes, this pin should be tied to ground.

Index Pulse (IPULSE): All drives have an index pulse output that goes active high as the beginning of any track passes under the drive head. Once the DDC detects this high signal (for at least one period of the READ CLOCK input), it assumes an INDEX PULSE has occurred. The DDC uses the INDEX PULSE input to begin various operations.

Serial Data Valid (SDV): This output pin goes high whenever the DDC is issuing or receiving either header field bytes and internal header CRC or ECC bytes, or data field bytes and internal data CRC or ECC bytes. It is set high synchronous with the first header or data bit appearing on the WRITE DATA output pin, or the READ DATA input pin. If the encapsulation mode is set then SCV is set high synchronously with the first sync byte (address mark). If the start with address mark bit is set and encapsulation is enabled SDV will be set high at the first sync #2 byte. (See Chapter 5 and 6.2.) It is set low synchronous with the last bit of internal CRC or ECC field ending on the WRITE DATA output pin or the READ DATA input pin. If internal CRC or ECC is not selected, it will be set low synchronous with the last bit of the header field or data field. The SERIAL DATA VALID pin may be used to select external ECC circuitry or for diagnostics in checking the lengths of the fields.

External ECC Field (EEF): This output pin is normally low, but will go high at specified times if external ECC has been selected. This will be during the time the external ECC field check bits need to be generated (with WRITE GATE high) or checked (with READ GATE high). It will be deasserted (synchronous with SERIAL DATA VALID output going low) after the last bit of the external ECC field has ended.

External Status (EXT STAT): The EXTERNAL STATUS input pin has three possible functions: Enabling wait states for the DMA, or Supplying external synchronization information and/or ECC information to the DDC. The user selects either the first alternative, or the other two. In other words, generating wait states is mutually exclusive with external synch and ECC. If the wait state alternative is selected, the use of this status pin is limited to only supplying wait states to the DMA bus cycle. If the latter two alternatives are selected, input signals on EXTERNAL STATUS may provide synchronization at the start of a header or data field, and external ECC error status at the end of the external ECC field.

4.2.3 Power Supply

V_{CC}, GND: The supply pins require a standard +5V ±10% regulated supply. As with any high speed controller that must connect to high speed buses, output switching transients can cause supply noise glitches which can affect other circuitry within the IC. Thus, a good ceramic decoupling capacitor is recommended to be connected across these pins. This capacitor should be ≥0.1 μf and should be located in close proximity to the V_{CC} and ground pins. Good GND and VCC planes are also recommended. Both of these precautions are to minimize the effects of current switchings on the chip affecting sensitive sections of the chip. With inadequate decoupling or GND and VCC planes, inexplicable behavior of the chip may result.

4.3 DDC FUNCTIONAL DESCRIPTION

This section is intended to provide a block level functional overview of the DDC. The detailed operational information is given in Chapter 7. A block diagram of the DDC is shown in *Figure 4.1*. The DDC is composed of a bus interface unit which communicates with the microprocessor and memory. It also is composed of a serializer and a deserializer that communicates with the disk. A single/dual channel DMA block provides intelligent on-chip data transfer. This DMA controller transfers data to and from the internal multi-mode FIFO block. A 32/48 bit ECC or 16-bit CRC correction block is included for error generation and checking of disk data. The functional description of each block follows.

4.3.1 Bus Interface

This block of the DDC provides an interface between the DDC and system bus through its two input/output data bus ports (AD0–7, AD8–15) and one input port (RS0–5). In the peripheral mode, the internal registers of the DDC are selected through pins RS0–5 and data is transferred between microprocessor and the internal registers through the I/O port AD0–7.

When the DDC is controlling the bus, two I/O ports (AD0–7, AD8–15) provide 16-bit address both for the local and remote DMA data transfers. In single channel DMA mode, an address up to 32 bits could be obtained to access memory up to 4 Gigabytes (see the DMA block description section). Multiplexed along with the DMA address information on these two ports is data information. In the 8-bit transfer mode only AD0–7 is used, and the interface logic contains a multiplexer to convert 16 internal data bits to 8 bits. For 16-bit transfers, AD0–7 transfers the lower 8 bits and AD8–15 transfers the upper 8 bits between the FIFO and the system bus.

4.3.2 Internal Registers

The DDC has 64 internal registers including parameter, pattern and count registers. Some of these registers are read-only, some write-only and the remainder read/write. These registers can be classified in four categories:

1) Command and Control Registers
2) ECC/CRC Registers
3) Format Registers
4) DMA Registers

Each of the above mentioned classes is described in the following paragraphs. A list of the DDC's internal registers with their hexadecimal addresses is given in Table 4.1.

COMMAND AND CONTROL REGISTERS

The Command and Control registers are the key registers of the DDC. They control basic functions and operations of the chip. The registers which can be included in this category are Drive Command (address 10H), Operation Command (address 11H), Status (address 00H), Error (address 01H), Disk Format (address 35H), Sector Counter (address 12H), Number of Sector Operations (address 13H), Header Byte Count/Interlock (address 0FH), and Header Diagnostic Readback (address 36H). Table 4.2 lists these registers along with a short description.

The Drive Command register basically determines the operations to be performed on the disk data. Also it can be used to set the DDC to format drives and to abort any operation in progress. The operations determined by the drive command register can then be controlled through the Operation Command register. The Operation Command register enables the DDC to issue certain interrupt and acknowledge signals during different operations. The Status register gives the status of the operation while the Error register indicates errors which may occur during these operations.

The DDC is adapted to the selected drive format through the Format register which determines the format of the information to be written to the disk. The Start Sector, Number of Operations, and Header Byte count registers, in conjunction with the Drive Command register, allow the DDC to perform multisector operations. The Header Diagnostic Readback register on the other hand enables the DDC to perform a readback operation on the header bytes present in the FIFO.

TABLE 4.1. The DDC Internal Registers In Numerical Order

Hex Address	Name
00	Status Register
01	Error Register
02	ECC Shift Register Out0/Polynomial Preset Byte0
03	ECC Shift Register Out1/Polynomial Preset Byte1
04	ECC Shift Register Out2/Polynomial Preset Byte2
05	ECC Shift Register Out3/Polynomial Preset Byte3
06	ECC Shift Register Out4/Polynomial Preset Byte4
07	ECC Shift Register Out5/Polynomial Preset Byte5
08	Polynomial Tap Byte0
09	Polynomial Tap Byte1
0A	Polynomial Tap Byte2
0B	Polynomial Tap Byte3
0C	Polynomial Tap Byte4
0D	Polynomial Tap Byte5
0E	ECC Control
0F	Header Byte Count/Interlock
10	Drive Command Register
11	Operation Command Register
12	Start Sector Number
13	Number of Sector Operations
14	Header Byte0 Pattern
15	Header Byte1 Pattern
16	Header Byte2 Pattern
17	Header Byte3 Pattern
18	Header Byte4 Pattern
19	Header Byte5 Pattern
1A	Local Data Byte Count L
1B	Remote Data Byte Count H
1C	DMA Address Byte0
1D	DMA Address Byte1
1E	DMA Address Byte2
1F	DMA Address Byte3

Hex Address	Name
20	Data Postamble Byte Count
21	ID Preamble Byte Count
22	ID Address Mark Byte Count
23	ID Synch Byte Count
24	Header Byte0 Control Register
25	Header Byte1 Control Register
26	Header Byte2 Control Register
27	Header Byte3 Control Register
28	Header Byte4 Control Register
29	Header Byte5 Control Register
2A	Data External ECC Byte Count
2B	ID External ECC Byte Count
2C	ID Postamble Byte Count
2D	Data Preamble Byte Count
2E	Data Address Mark Byte Count
2F	Data Synch Byte Count
30	Data Postamble Pattern
31	ID Preamble Pattern
32	ID Address Mark Pattern
33	ID Synch Pattern
34	Gap Byte Count
35	Disk Format Register
36	Local Transfer Reg/Header Diagnostic Readback
37	Remote Transfer Register
38	Sector Byte Count L
39	Sector Byte Count H
3A	Gap Pattern
3B	Data Format Pattern
3C	ID Postamble Pattern
3D	Data Preamble Pattern
3E	Data Address Mark Pattern
3F	Data Synch Pattern

TABLE 4.2. Summary of Control Registers

Address	Register Name	General Operations
00H 01H 0FH	Status Register Error Register Header Byte Count	Disk Operation, DMA Status (Read) Error Determination of Operation
10H 11H	Drive Command Operation Command	Start Disk Operation Reset, Remote DMA, INTR Operation
12H 13H	Start Sector No. of Sector Operations	Can Contain Sector Number Used in Multi-Sector Operation
35H 36H	Disk Format Header Diagnostic	MFM, Hard Sector, External ECC

THE ECC/CRC REGISTERS

The ECC/CRC registers (addresses 02H to 0EH) are used to set up the DDC for desired error detection and correction configuration. Registers 02H to 07H are read-write and contain the preset pattern for the internal ECC. The preset pattern is the data that the ECC shift register is initialized to prior to an operation. Typically this is all ones. During a correcton cycle, reading these registers provides the syndrome bytes to correct the erroneous data. Registers 08H to 0DH are write-only and are used to load in the internal ECC polynomial required for the drive format selected. The ECC Control register (address 0EH) is used for selecting the internal ECC Correction Span, inversion of input or output check bits to the ECC register and ECC encapsulation (the mode that includes sync bytes (address marks) in the check bit calculation). These registers are listed in Table 4.3.

THE FORMAT REGISTERS

The Format registers, Table 4.4, (addresses 20H to 2FH, 30H to 35H and 38H to 3FH) determine and control the format of the fixed fields for the selected drive type according to the selected format. (*Figure 4.3* shows the Sector Format fields incorporated in the DDC). Registers 20H to 2FH contain the byte count of the fixed fields along with the 6 Header Control registers while Registers 30H to 35H and 38H to 3FH contain the patterns of the fixed fields along with the Inter Sector Gap Count and the Sector Byte Count registers. These Registers are shown in Table 4.4. Since almost every pattern register has an associated count register (which controls the repetition number of its field) or a control register, Table 4.4 is organized to show both together.

THE DMA REGISTERS

The DMA registers consist of the Local Transfer register (address 36H), Remote Transfer register (address 37H), and count and address registers, 1AH to 1FH. The Local Transfer register controls the data transfers between the DDC and buffer memory by controlling data and address bus lengths, byte ordering, memory cycle and the burst length. The Remote Transfer register, on the other hand, controls the data transfers between the buffer memory and the system I/O port in dual bus architectures. In addition to the data and address bus lengths, the memory cycle length, and determining the burst lengths, it also controls the transfers in the dual channel DMA mode.

Registers 1AH and 1BH determine the byte count required in a remote data transfer while registers 1CH to 1FH are DMA Address bytes 0 to 3 for local and remote transfers.

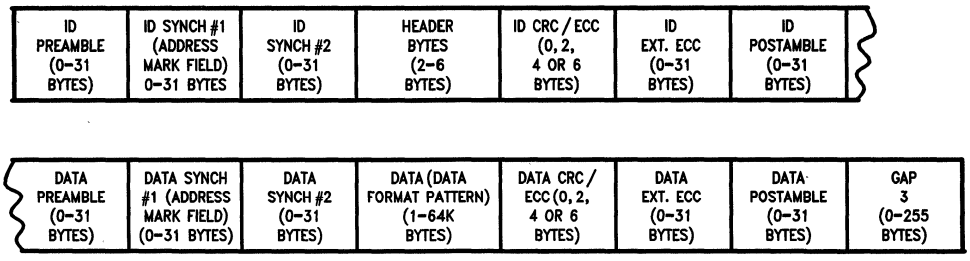
Table 4.5 lists the DMA control and address registers, and their function when the DDC is used in either dual channel or single channel mode.

TABLE 4.3. ECC Control Registers

Shift Reg/ Polynomial Preset	Polynomial Tap	Register Description
02	08	ECC Shift Reg/Poly Preset and Tap 0
03	09	ECC Shift Reg/Poly Preset and Tap 1
04	0A	ECC Shift Reg/Poly Preset and Tap 2
05	0B	ECC Shift Reg/Poly Preset and Tap 3
06	0C	ECC Shift Reg/Poly Preset and Tap 4
07	0D	ECC Shift Reg/Poly Preset and Tap 5
0EH		ECC Control
08H, 09H		Data Byte Count

TABLE 4.4. Format Count, Control and Pattern Registers

Count/ Control Reg	Pattern Reg	Header/ID Field Register Descriptions	Count Reg	Pattern Reg	Data Field/ECC Format Register Description
20	30	Data Postamble	2A	—	Data External ECC
21	31	ID Preamble	2B	—	ID External ECC
22	32	ID Synch Field 1	2C	3C	ID Postamble
23	33	ID Synch Field 2	2D	3D	Data Preamble
			2E	3E	Data Address Mark
24	14	Header Byte 0	2F	3F	Data Synch
25	15	Header Byte 1	34	3A	Post Sector Gap
26	16	Header Byte 2			
27	17	Header Byte 3	—	3B	Data Format
28	18	Header Byte 4	38	—	Sector Byte (LSB)
29	19	Header Byte 5	39	—	Sector Byte (MSB)



TL/F/8663–70

FIGURE 4.3. Sector Format Fields Incorporated in the DDC

TABLE 4.5. DMA Registers and Functions in Dual or Single Channel Mode

Addrs	Register Name	Single Channel	Dual Channel
1AH	Remote Count	NA	LSB Data Xfer Count
1BH	Remote Count	NA	MSB Data Xfer Count
1CH	DMA Addr 0	A0–A7	A0–A7 Local DMA
1DH	DMA Addr 1	A8–A15	A8–A15 Local DMA
1EH	DMA Addr 2	A16–A23	A0–A7 Remote DMA
1FH	DMA Addr 3	A24–A32	A8–A15 Remote DMA
36H	Local Transfer	Configures Local or Single Channel	
37H	Remote Transfer	Configures Remote (Dual Channel Mode)	
37H	DMA Sector Count	DMA Sector Counting (Dual Channel)	

4.3.3 The FIFO

The primary function of the DDC is to transfer data between disk and the system. The DDC has been configured so that during a disk data transfer operation, it does not occupy the bus for the whole disk transfer. Instead, it allows burst transfers so that the bus is free between the bursts for normal system usage. Systems with a main microprocessor and main memory will interface directly to the DDC. In this type of application, burst data transfers will require occupancy of the main system bus, so use of the bus must be granted at the discretion of the system. For example, if the system is performing a higher priority operation, it must not relinquish the bus for disk data transfer.

Once the bus has been relinquished, the system is held from performing other operations and a burst of data is then transferred. For the DDC, these requirements mean first, that some degree of data buffering is necessary to store the continuous arrival or removal of disk data, and second, that when the bus is granted, transfer must be fast. The amount of data buffering will be dependent on the system, but the majority of low-end systems should be able to respond to a data transfer request from the DDC within 50 μs. Most disk drives in this kind of application run at a data rate of 5 Mbits/sec, or one bit every 200 ns. Typically then, the data buffer must be able to store around 250 bits. A 32-byte FIFO has been included on the DDC enabling it to operate with most bus systems.

The data is transferred between the FIFO and the local memory (dual channel mode) or system memory in different burst thresholds, 1, 4, 8 or 12 words for 16-bit wide word transfers or 2, 8, 16 or 24 bytes for byte wide transfers. In a disk read operation when the FIFO fills to the selected threshold level with disk data, the DMA controller issues a data transfer request. The FIFO continues to fill. Whenever the DMA gets access to the system bus, it transfers data in selected bursts. These bursts may be of fixed length or until the FIFO empties. If the DMA request is not acknowledged, and the disk data fills the FIFO before it reaches to its maximum 32-bytes capacity, the FIFO Data Lost error occurs and the operation is aborted.

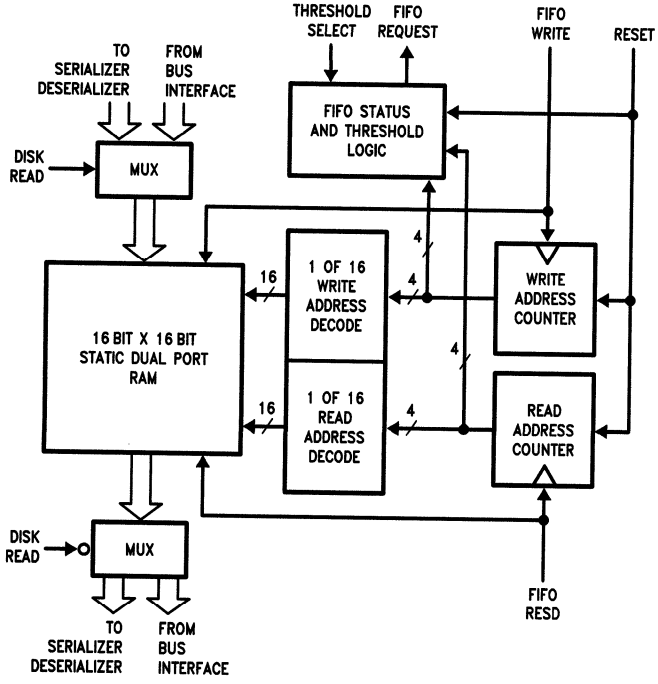
Conversely, in a disk write operation, the FIFO is first filled. It then requests a new data burst when it empties to below the selected threshold level. Depending on burst mode, the DMA will then request a fixed number of bytes or fill the FIFO. If the FIFO completely empties during the operation, a FIFO Data Lost error is again generated.

Figure 4.4 shows the basic blocks that compose the FIFO. It consists of a 16 x 16 bit dual port RAM array, which is addressed by a read counter and a write counter. These counters are decoded to address the array and also to feed the status logic which takes the difference between these pointers to generate the threshold signals. The input and output data ports of the FIFO can be connected to the serializer/deserializer or bus interface block. The direction of the FIFO is determined by the disk operation, read or write. An 8-bit bus interface is supported in 8-bit transfer mode by treating the FIFO as two interleaved banks of 8 bits.

4.3.4 The DMA

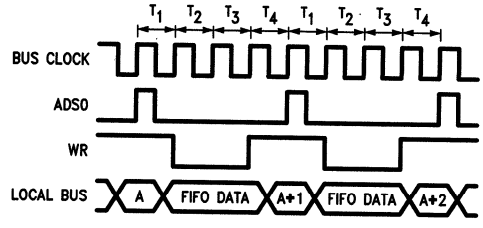
The DDC has an important feature that helps both in saving external ICs and in increasing data throughput, namely powerful DMA capability. With on-chip DMA capability, there is no need to dedicate a channel of a DMA controller for disk transfers. This offers two advantages: first, it may alleviate the need for a DMA controller chip, and second, memory transfer time will be faster because DMA controllers are relatively slow, usually around 2 Mbytes/sec maximum throughput. The DDC can transfer data much faster than this, especially when selected to transfer 16 bits each cycle. This faster transfer rate offers a much lower bus occupancy time, freeing the bus sooner for other usage.

When using the DMA capability, the DDC becomes bus master during the data transfer operation. In bus master mode the DDC issues incrementing address information at the start of each memory cycle. Each read or write memory cycle takes four clock periods, using a similar sequence to a four clock cycle microprocessor with multiplexed address and data bus. In some cases a five clock cycle sequence is used when two address words must be multiplexed to form the DMA address. *Figure 4.5* shows a typical read or write cycle of 4.



TL/F/8663–71

FIGURE 4.4. Simplified Block Diagram of DDC's FIFO



TL/F/8663–72

FIGURE 4.5. DDC-to-Memory Word Transfers (16-Bit Address)

The block diagram on the DMA section is shown in *Figure 4.6*. The heart of the DMA is a sequencer/PLA that uses inputs from the FIFO (FIFO Request) and disk control logic (DC Ready) for local DMA. It also generates signals for the remote channel to determine if a transfer is necessary. It then issues the requests, and once the acknowledge is received, the PLA sequences through the DMA cycle by placing the address onto the data/address bus (via the bus interface block) and manipulating ADS0, ADS1, read, and write strobes. The various counters are then incremented. In addition to the address counters, there are counters to keep track of bytes per sector (local and remote channel), bytes per header (local and remote), number of total sector operations, and remote counter burst length. The DMA sector counter is used in the tracking mode (described later), and enables the destination DMA whenever it is not zero. The remote transfer counter is decremented after each remote transfer and is used to set the total length of the transfer. When it reaches zero the sequencer halts remote DMA operations.

The DDC can be configured into two DMA modes, Single Channel and Dual Channel. The Dual Channel Mode has two sub-modes: Tracking and Non-Tracking. The general operation of these modes is described below.

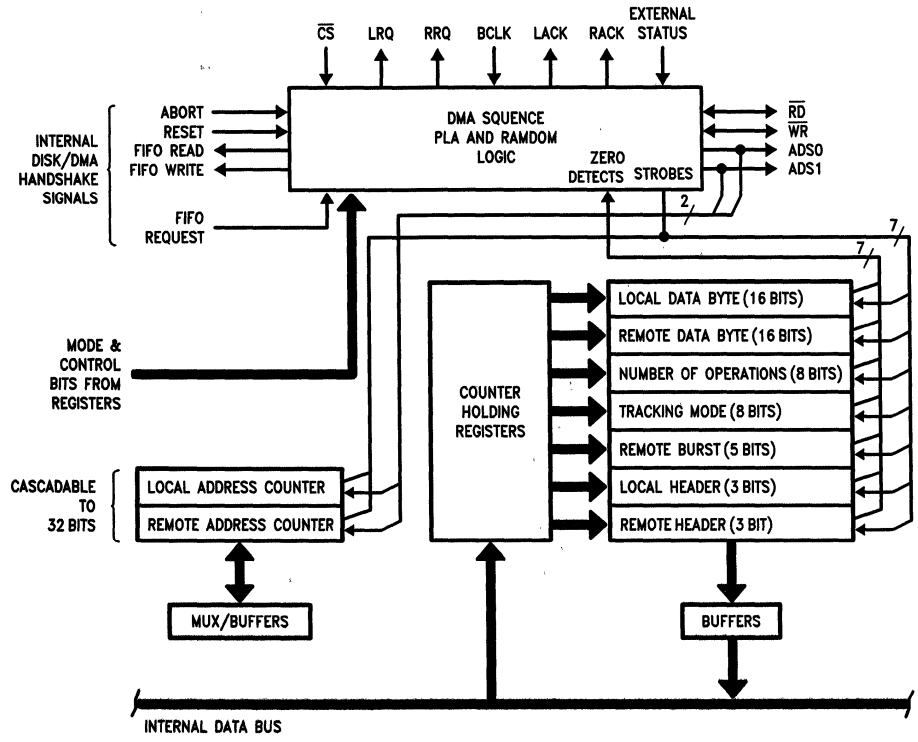
SINGLE CHANNEL

In single channel DMA mode the DDC interfaces directly to main memory having 32 address bits available to access up to 4 Gigabytes of memory. *Figure 4.7(a)* shows the DDC in single channel DMA configuration. The lower 16 address bits are normally issued at the start of each memory cycle so that most memory cycles comprise four clocks. The upper 16 bits are issued at the start of an operation or if the lower 16 bits rollover. In these cases, the memory cycle becomes 5 clock periods, the upper 16 bits are issued during first clock period and the cycle then completes the next four clock periods as in normal read or write operation. The upper two bytes of address information should be latched during the first clock period.

DUAL CHANNEL

Some systems may require the DDC to interface to a local bus with its own dedicated buffer memory before it interfaces to the main system. Such an application would be in intelligent disk drives that comply with the SCSI (SASI) or IPI interfaces. Intelligent drives may receive or transmit data whenever the controlling unit is ready. Another application could be in higher end systems, where the main memory is hooked onto a main bus such as the Q-BUS™, MULTIBUS®, VME or Future Bus. These buses are usually very busy and often impose high latency times while the main processor is performing important tasks. Once the bus is free, it is advantageous to be able to transfer all the disk information in as short a time as possible to minimize bus occupancy. For these types of applications, the dual channel capability of the DDC is ideal.

In the dual channel mode, the DMA generates a 16-bit address for both the local and remote transfers. The local channel controls the data transfers between the FIFO and the local buffer memory. The remote channel, on the other hand, controls data transfers between the addressed local buffer memory and the main system bus through an I/O port. A DMA channel in the system DMA controller could



TL/F/8663–73

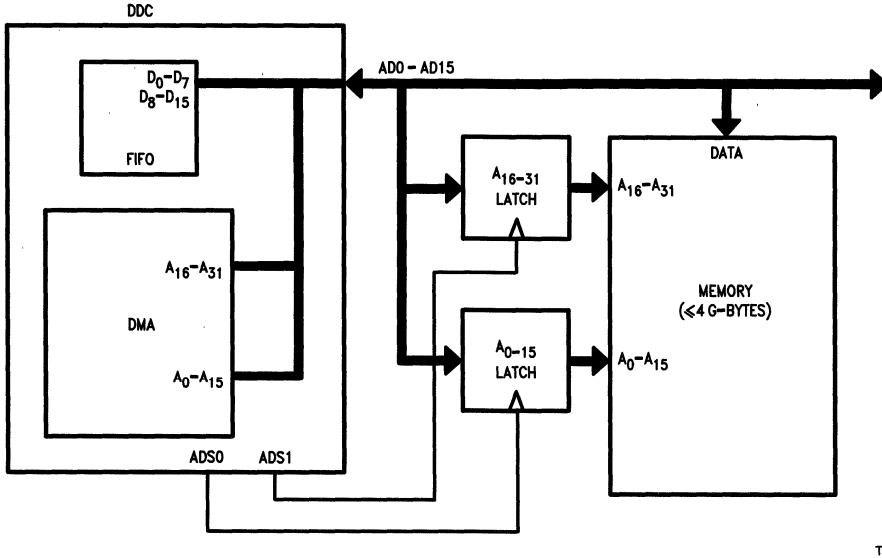
FIGURE 4.6. Block Diagram for Dual/Single Channel DMA Controller

then transfer the data from the port to system memory. Refer to *Figure 4.7(b)*. The local request and remote request are issued when the DDC requires a transfer, the microprocessor and bus arbiter respond by acknowledging the requests. If both channels are requesting, the system should arbitrate the acknowledge, however the local DMA has a higher priority if both requests are acknowledged.

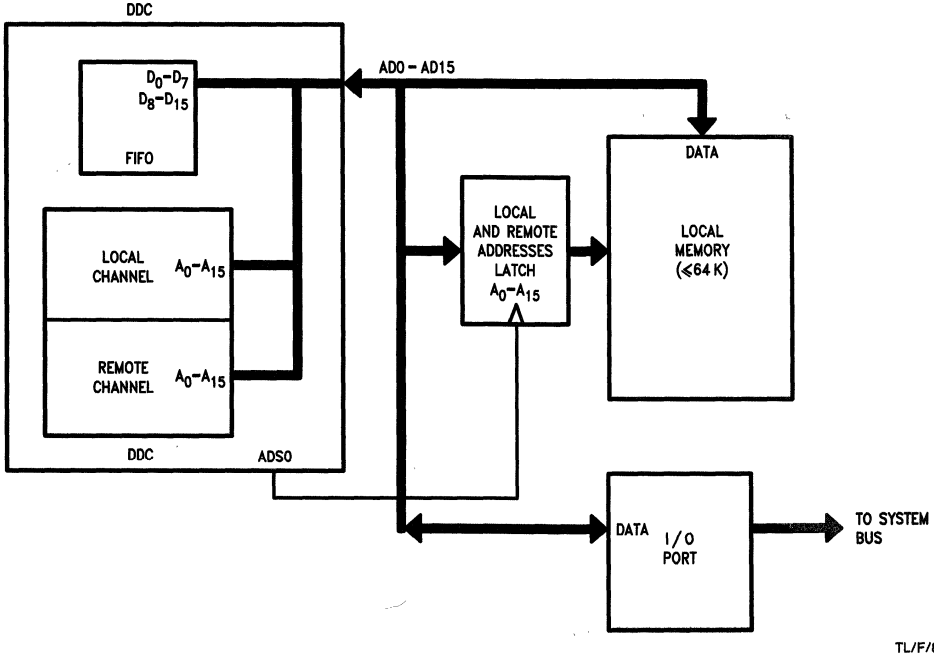
The dual channel mode can be further divided into two modes, Tracking (non-overlapping) and Non-Tracking (overlapping). These dual channel modes are described below.

TRACKING OR NON-OVERLAPPING MODE

In Tracking or Non-Overlapping Dual DMA mode the DMA controls the local and remote transfers in such a way that the local buffer memory appears to the system as a large



(a). The DDC in Single Channel DMA Mode



(b). The DDC in Dual Channel DMA Mode

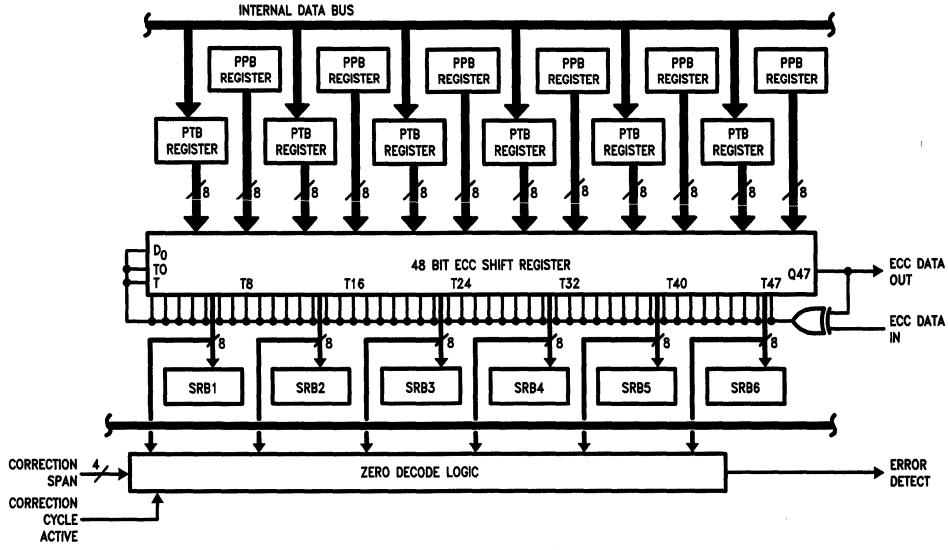
FIGURE 4.7

FIFO. This allows the system to transfer data to or from the local buffer memory whenever the system is ready, but with protection against overlapping of disk data. Basically this mode of operation is more applicable to multi-sector operations where the DDC efficiently interleaves bursts of data into and out of the buffer memory using both channels. Each channel has 16 address bits, allowing up to 65k of buffer memory to be used by the two channels. While doing a remote transfer in a disk read operation, data is read (bursted out) from the same local memory area where it was written to (bursted in) during the local transfer. Similarly, in a disk write operation, data is read from the same local memory area where it was written to during the remote transfer. In both cases, buffer memory addresses for local and remote transfers are issued such that data is never overlapped. If the two channels track very closely, then large amounts of contiguous data can be transferred, making the buffer memory appear to be a multi-megabyte FIFO.

The protection against overlapping of disk data is enabled by use of the DMA Sector Transfer Counter in the DDC. The counter is initially reset at the start of the operation. It is then incremented each time the source has transferred a sector of data into the buffer memory, and is decremented each time the destination has transferred a sector of data from the buffer memory. Whenever the count is zero, destination transfers are inhibited, so preventing the destination from catching up and overtaking the source transfers.

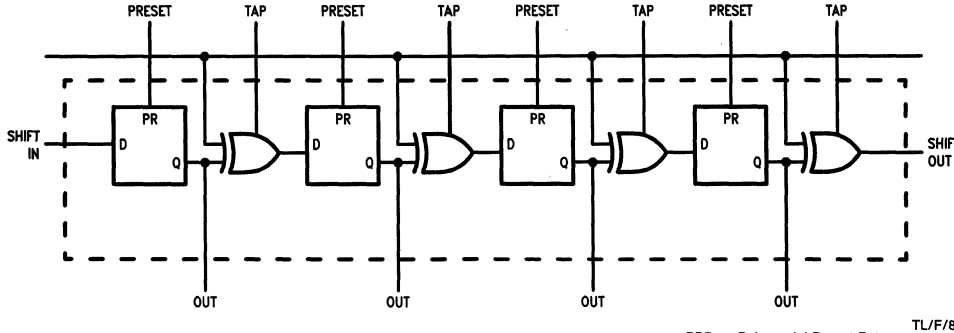
NON-TRACKING OR OVERLAPPING MODE

Some systems require that the two DMA channel are completely independent of each other. For this type of application the DDC can be configured to set up both DMA channels independently. The Local and Remote operations may be from different areas of memory or common areas. The local DMA may already be performing an operation when the Remote DMA is instructed to begin an operation. Likewise a Remote operation can be in progress when the Local operation is initialized. One operation can be for reading memory and the other for writing. This puts the burden on the user to protect from overwriting the buffer memory. In other words, the controlling microprocessor has the responsibility of ensuring that no memory overwriting occurs when both local and remote transfers are in progress. This mode



(a). ECC Shift Register Logic

TL/F/8663–76



(b). Detail of 4-Bit Section of Shift Register

TL/F/8663–77

PPB = Polynomial Preset Byte
PTB = Polynomial Tap Byte
SRB = Shift Register Buffer

FIGURE 4.8

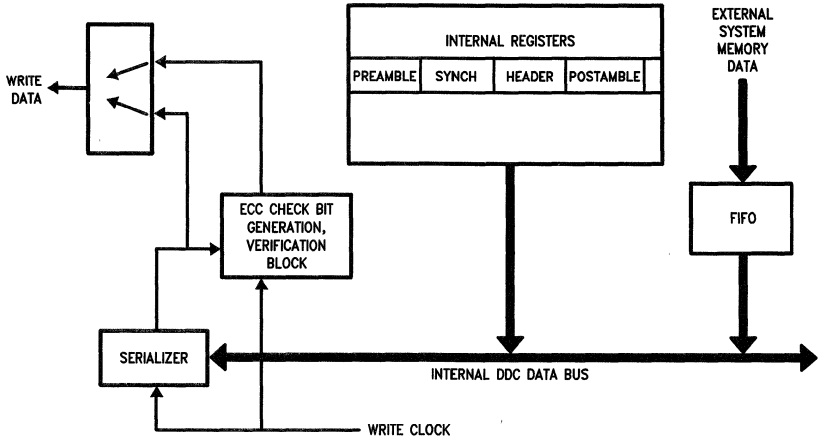
also gives the user the freedom to use the remote DMA controller with no restrictions, even for general non-disk transfers, such as for high level commands or status tranfers.

4.3.5 Error Detection and Correction

A fixed CRC code for detecting errors only, and a programmable ECC code for detecting and correcting errors can be used via the on-chip Error Detection and Correction block. The DDC has full polynomial programmability for 32 bits or 48 bits of ECC appendage, along with a programmable correction span from 3 to 15 bits and a programmable preset. The DDC can also be easily interfaced to external ECC circuitry if desired.

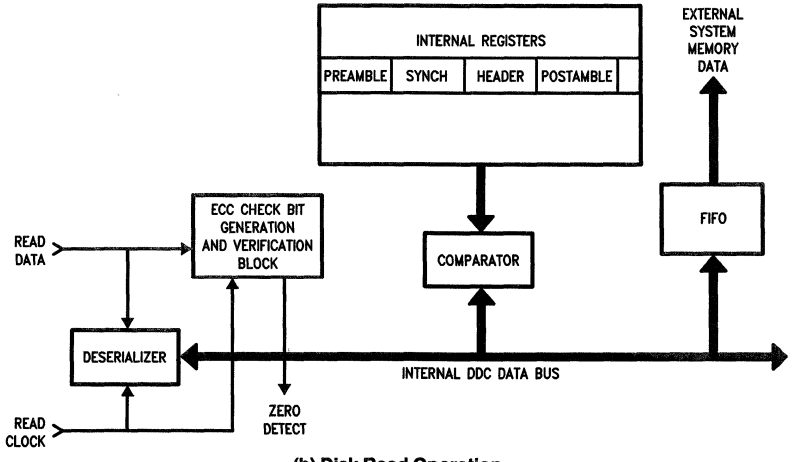
The Error Detection and Correction block mainly consists of a 48-bit shift register with XOR taps. It generates and then appends the check bits to header and data fields. The CRC uses the standard CCITT polynomial that provides 16 generated check bits. The CRC-CCITT code is hardware implemented in the DDC. The ECC code may be a Fire code or a computer generated code with either 32 or 48 generated check bits. The selected 32- or 48-bit ECC polynomial can be implemented by means of the 48-bit shift register and the Polynomial Tap and Preset Byte registers (addresses 02H thru 0DH).

The ECC Shift register logic is shown in *Figure 4.8*. In this figure the internal data bus connects to the Polynomial Preset Byte Registers (PPB). These register bits feed into the shift register latches. Any bit set in the PPBs will preset a corresponding flip-flop before an ECC operation begins, while all others will not be set. The Polynomial Tap Byte Registers feed the XOR gates in the ECC Shift Register. When a PTB bit is reset, the associated XOR gate is enabled for a particular ECC register bit. This effectively creates the ECC polynomial tap. The outputs of each shift register flip-flop bit input to a set of output buffers which drive the internal DDC data bus. This enables reading of the ECC registers. The ECC outputs also go to a combinational logic block that decodes the contents of the ECC shift register and the correction span. If at the end of a detection cycle the ECC shift register contains zero then no error was detected.



TL/F/8663–78

(a) Disk Write Operation



TL/F/8663–79

(b) Disk Read Operation

FIGURE 4.9

INTERNAL CHECK BIT GENERATION AND CHECKING

When writing to the disk, the CRC/ECC shift register is preset from the Preset ECC registers. At the same time that the DDC is outputting either the Header field or Data field bytes as a serial data stream through the Serializer, it is feeding them serially into the CRC/ECC shift register as shown in *Figure 4.9(a)*. When the last bit of the Header or Data field has been transmitted out of the DDC, the DDC begins shifting out the generated check bits from the CRC/ECC shift register starting with the MSB and ending with the bit 0. After the specified number of check bits have been appended the DDC internally switches to the next field.

When reading from the disk, the shift register is first preset from the Preset ECC registers before the read data operation begins. The incoming Header or Data field is serially fed into the CRC/ECC shift register as shown in *Figure 4.9(b)*. When all the Header or Data field bits and all the generated check bits have entered the CRC/ECC shift register, the status of the bits in the CRC/ECC shift register is checked for the all zeroes condition. If this condition is met, it signifies the field contains no errors. If any of the CRC/ECC shift register bits are high, the field contains an error. The Header and the Data field errors are indicated by the Status and Error registers respectively.

INTERNAL ERROR CORRECTION

The DDC is capable of correcting from 3 to 15 contiguous bits in error for selected 32- or 48-bit polynomials. The value desired is set in the ECC Control Register, in other words, it can correct a span of the selected amount. The DDC can be put in the Correction Mode through the Operation Command register. The CRC/ECC shift register contains a non-zero 32- or 48-bit pattern which is used to determine the location of the bytes in error and the error pattern. The most significant 3 to 15 bits of the 32 or 48 bits are selected as the Syndrome bits, while the rest are checked for a zero detect. During the correction mode, the CRC/ECC shift register is reverse shifted. Also, reverse shifting guarantees that the correction cycle will be completed within the time it takes to read one sector of the disk.

When the reverse shifting of the shift register begins, the Data Byte Counter register begins decrementing from its preloaded value of the number of data and ECC bytes in the sector. Another 3-bit counter is used to keep track of byte boundaries in the serial bit stream of the whole sector. Reverse shifting continues until all zeroes are detected in the (32-C) or (48-C) bits of the CRC/ECC shift register (where C is the correction span selected). When this occurs and the 3-bit counter contains all zeroes, the clock is stopped. At this point the C syndrome bits contain the bit error pattern of the byte indicated by the Data Byte Counter register. If the 3-bit counter count was not zero when the zero-detect occurred, then the CRC/ECC shift register has to undergo further reverse shifts to byte align the right byte in error. If the Data Byte counter register count goes to zero and the zero-detect is not obtained, then the error is non-correctable. If either the zeroes condition is determined or the Data Byte counter decrements to zero, an INTERRUPT is issued to indicate to the microprocessor that the correction cycle has finished.

The results of the correction cycle are indicated by the Status register. In the case of a correctable error, the error must be in either the Data field or the check bits of the ECC field or overlapping both fields. If the error is only in the ECC field then the memory data is correct and no further action is needed to complete the correction. But if the error is in the Data field then it can be corrected by XORing the C syndrome bits in the CRC/ECC shift register with the contents of the relevant memory location determined from the final Data Byte Counter register count.

EXTERNAL ECC

Some users may wish to use an ECC polynomial code with a different number of check bits. Some encoding schemes require a wider error correction span, or some users may prefer some high integrity ECC codes such as Reed-Solomon code. For these reasons DDC has been configured to interface easily to external ECC circuitry.

When the DDC is configured to utilize an external circuit for ECC code, the external ECC code may use any polynomial that generates from 1 to 31 bytes of check bits. The external circuitry is informed by the DDC when data is valid and when to generate check bits (for writing) or detect (when reading) through SDV, EEF and EXT STAT pins. Refer to Section 4.2 for pin description. The external ECC may be used to encapsulate internal CRC/ECC field as a confirmation of error detection.

4.3.6 The Serializer-Deserializer

This section of the DDC interfaces to the disk. The Serializer takes byte wide data either from the internal registers or the FIFO into a shift register and serially outputs the bits in a continuous bit stream, starting with the most significant bit. This serial data is then fed into the Error Detection and Correction block for check bit generation. When the CRC/ECC appendage is about to begin, the serializer stops shifting out and the Error Detection and Correction block begins shifting out the check bits, again most significant bit first. At the end of the appendage the Serializer starts shifting out further information to finish the segment. The serial data passes through the MFM Encoding and Precompensation block, if selected, or is output to the external encoder.

During a read operation, the incoming serial NRZ data feeds into the Deserializer and the CRC/ECC block, most significant bit first. The Comparator continually checks the incoming data for a synchronizing pattern that matches the pattern loaded into the internal pattern registers. Once a match occurs, the DDC then knows the byte boundary such that all further bytes from the deserializer are byte synchronized to the boundary first established. CRC/ECC fields, postamble, and preamble fields are not required to be deserialized, and do not enter the deserializer. Once the address mark and/or synch byte have aligned, the header bytes preloaded into the internal registers are sequentially output to the Comparator as each incoming byte is ready. The Comparator checks all the header bytes in turn for a match. If a full match is detected, the DDC checks the CRC/ECC appendage and prepares for the following data field. Finally, the data field is read, and serial data bytes are converted to parallel. They then enter the FIFO from the Deserializer to be transferred by the DMA.

The basic blocks associated with the Serializer/Deserializer are shown in *Figure 4.10*. For Serialization, data from either the Pattern Registers, Sector Counter, or FIFO is multiplexed to a holding latch. The holding latch will load the Serializer/Deserializer shift register at the appropriate time

(determined by the disk controller's PLA). This data is shifted through the EEC/Data MUX and the MFM or NRZ logic to the Serial Data output pin. When serializing data the write clock feeds the shift register.

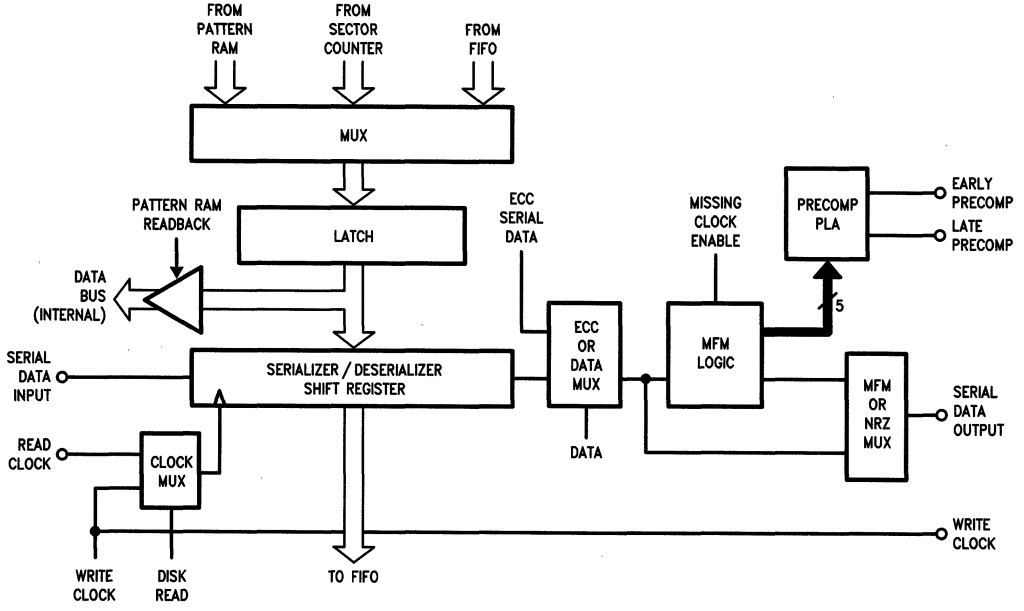
For Deserialization of data, Read Data and Read Clock feeds the internal data bus. Once byte alignment is determined, a byte clock controls loading data into the FIFO.

MFM ENCODING AND PRECOMPENSATION

The DDC can be set to output MFM encoded data and Precompensation information in a disk write operation. The MFM Encoding and Precompensation block of the DDC consists of a 5-bit shift register, and MFM and Precompensation encode logic. The NRZ data coming out of the Serializer passes through the shift register and then is fed into the MFM and Precomp encode logic. The MFM Encode logic converts this NRZ data into MFM and also inserts the missing clocks when Address Mark fields are required to be written to the disk.

The DDC can be programmed to output two control signals, EARLY PRECOMP or LATE PRECOMP, if precompensation is desired. The information on these output pins is then used by the external Precomp Circuitry (MUX, Delay logic, etc.) to perform the actual Precompensation. These pins perform an algorithm that compensates for the bit shifting that occurs when the magnetic flux transitions are recorded on the disk.

The MFM Encoder and precompensation block are shown in *Figure 4.11* also. After serialization, logic performs the MFM encoding. 5 bits from the encoder monitor previous and subsequent data to determine whether early, late, or no precompensation is needed.



TL/F/8663–80

FIGURE 4.10. Serializer/Deserializer Block Diagram

CHAPTER 5 The DDC Registers and Commands

5.0 INTRODUCTION

In this Chapter, the DDC's (DP8466) registers and its command operation will be described in more detail. Initially, a general overview of the registers is given. This section can be used to supplement the data sheet register descriptions. Then details on the various methods of formulating commands and operations are described. Understanding the wide variety of operating modes presented in this section will enable a better understanding of which modes are suitable for which applications.

5.1 INTERNAL REGISTERS AND COUNTERS

In this section a detailed description of internal Command, ECC/CRC, Format and DMA pattern, control and count registers, and various counters is provided. A summary description of the registers is given below, as well as a description of important command bits. Additional details of the DDC's registers are given in the DP8466's Data Sheet.

5.1.1 Command Registers and Counters

The Command registers and counters are listed in *Figure 5.1* and explained in the following paragraphs. These registers include the drive and Operations Command registers, Disk Format register, Error and Status registers, Sector Counter and Sector Operations Counter registers.

Name	Hex Address
Drive Command Register	10H
Operation Command Register	11H
Disk Format Register	35H
Status Register	00H
Error Register	01H
Sector Counter	12H
Number Of Sector Operations	13H
Header Byte Count Register	0FH
Interlock Complete Signal	
Header Diagnostic Readback	36H

FIGURE 5.1. Command Registers and Counters

DRIVE COMMAND (DC) REGISTER (Address = 10H)

The drive command register is shown in *Figure 5.2.* This register is loaded when the DDC is required to perform a command. Disk operations are started after loading this write-only register. This register can be loaded to start a new command when the Next Disk Command bit is set in the Status register. The bit descriptions are given below.

Re-Enable (RED)

When loading a disk command a zero should be written to this bit to permit normal operation. A one should be written to the bit to re-enable the DP8466 after either a reset via the RESET pin or RES bit in the Operation Command Register.

Start Operation (SAIS)

Bit SAIS determines when the operation for the command being written to the drive command register shall begin. If SAIS is high, the operation will begin when the DDC detects either an INDEX or SECTOR PULSE for hard sectored drives, or immediately for soft sectored drives. If SAIS is low the operation only begins when the DDC detects an INDEX PULSE.

Single/Multi Sector Operation (MSO)

Bit MSO of the drive command register indicates whether the operation is for just one sector or a number of sectors. If MSO is set high then the DDC can perform a multi-sector operation. Multi-sector operations usually are handled on logically (by sector address) contiguous rather than physically contiguous sectors. The DDC can perform both types of multi-sector operations.

D0	RED	RE-ENABLE	0 = No Action 1 = Re-Enabled
D1	SAIS	START OPERATION	0 = Start on Index Pulse 1 = Start on Index/Sector Pulse or Immediately
D2	MSO	SINGLE/MULTI-SECTOR OPERATION	0 = Single 1 = Multi
D3	FMT	FORMAT MODE	0 = Normal 1 = Format
D4 D5	HO1 HO2	HEADER OPERATIONS	00 = Ignore Header 01 = Compare Header 10 = Write Header 11 = Read Header
D6 D7	DO1 DO2	DATA OPERATIONS	00 = Invalid* 01 = Check Data 10 = Write Data 11 = Read Data

*Unless used with an Ignore Header operation. When HO1, HO2, DO1 and DO2 are written as 0 then no operation is performed. This is useful when Re-Enabling the DDC by setting the RED bit, for example.

FIGURE 5.2. Drive Command Register Bit Assignments

Format Mode (FMT)

The DDC can be set to format a disk by setting bit FMT high. The DO2, DO1, HO2, and HO1 bits must be set for a write-header/write-data operation, 1010. (For details please refer to the disk formatting section).

HEADER OPERATIONS (HO2, HO1)

Bits HO2 and HO1 determine the operation to be performed on the ID fields. The details are given below.

Ignore Header (HO2 = 0, HO1 = 0)

In an Ignore Header operation, after the byte alignment of Address Mark and/or Synch fields, the header bytes comparison and ECC/CRC checks are not performed. This results in reading or writing the associated data with respect to any sector encountered.

Compare Header (HO2 = 1, HO1 = 1)

Compare Header operation is the normal mode of header operation for locating the selected sector. In this operation the header bytes are compared with the corresponding values in the pattern registers and CRC/ECC checks are also carried out.

Write Header (HO2 = 1, HO1 = 0)

Write Header operation is normally performed during disk formatting. The Header bytes are written to the disk either from the pattern register or the buffer memory through FIFO depending upon the FIFO Table Format (FTF) bit of the Disk Format register.

Read Header (HO2 = 1, HO1 = 1)

Read Header operation performs CRC/ECC checks and transfers the header bytes into buffer memory through the FIFO for diagnostic purposes. If during this operation, a header containing a CRC/ECC error is encountered, the operation is aborted immediately and the header fault bit (Status Register) is set. An interrupt is generated, but no Error register bits are set.

DATA OPERATIONS (DO2, DO1)

Bits DO2 and DO1 determine operations to be performed on data fields.

Check Data (DO2 = 0, DO1 = 1)

After the preceding header operation and the byte alignment, the Check Data operation performs the CRC/ECC checks on the data fields. It does not transfer the data field to the FIFO and hence no data is transferred to memory via the DMA channels.

Read Data (DO2 = 1, DO1 = 0)

Read Data operation, on the other hand, transfers data to external memory via FIFO after performing CRC/ECC checks.

Write Data (DO2 = 1, DO1 = 1)

In Write Data operation, after the associated header operation, data bytes are written to the selected sector from the FIFO using the DMA channel.

OPERATION COMMAND (OC) REGISTER
(Address = 11H)

Operation command register, shown in *Figure 5.3,* is a write-only register and may be updated before each disk operation. This register controls some of the basic DDC operating modes, such as interrupts, starting remote DMA, starting a correction, and precompensation.

Reset (RES)

When RES is set high, the DDC enters the standby mode and remains in this mode until this bit is reset and a one is written to the RED bit in the Drive Command register. To properly reset the DDC this bit must remain set for 32 read clock periods and 4 bus clock periods (with both clocks applied). This bit has the same effect as setting the RESET input pin low. Read and Write Gate are deasserted, the FIFO is cleared, DMA requests are removed, the Error register is cleared and status register is cleared except the Next Disk Command is set, and the abort bit is reset. The parameter registers, sector counter and number of sectors registers are not affected.

Enable Interrupts (EI)

Bit EI, when set high, enables the DDC to issue interrupts when certain conditions are met, such as upon successful completion of a command.

Enable Header Complete Interrupts (EHI)

Setting bit EHI high will enable the DDC to issue an interrupt at the completion of a header operation so that it can be loaded with new information before the next ID operation begins. New commands can be loaded or some pattern information could be updated during the data operation of a sector.

D0	RES	RESET	0 = Normal Operation 1 = Reset DDC
D1	EI	ENABLE INTERRUPTS	0 = Disabled 1 = Enabled
D2	EHI	ENABLE HEADER COMPLETE INTERRUPTS	0 = Disabled 1 = Enabled
D3	SRI	START REMOTE INPUT	0 = No Remote Read 1 = Start Remote Read
D4	SRO	START REMOTE OUTPUT	0 = No Remote Write 1 = Start Remote Write
D5	EP	ENABLE PRECOMPENSATION	0 = Disabled 1 = Enabled
D6	SCC	START CORRECTION CYCLE	0 = No Cycle 1 = Start Cycle
D7	IR	INTERLOCK MODE	0 = No Interlock 1 = Interlock Mode

FIGURE 5.3. Operations Command Register

Start Remote Input (SRI)

The DDC initiates the transfer of data from the system to local memory when SRI is set high. This enables the start of remote transfers (non-tracking dual DMA mode) to the local buffer.

Start Remote Output (SRO)

The DDC initiates the transfer of data from local memory to system when bit SRO is set high. This starts remote transfers from the local buffer.

Enable Precompensation (EP)

The DDC will allow precompensation if bit EP of the operation command register is set high. This bit is valid only if the AMF/EARLY PRECOMP and AME/LATE PRECOMP pins are configured as write precompensation control pins.

Start Correction Cycle (SCC)

The internal correction cycle is initiated by setting bit SCC high (see ECC section).

Interlock Mode (IR)

In some situations, if it is desired to update the header bytes (during disk formatting) or issue a new drive command before the next command, the DDC must be put in Interlock mode by setting bit IR high.

In interlock mode, after every header operation, the DDC issues an interrupt after Header Match Complete flag, (HMC), in the status register goes high, indicating that the information can be updated before the beginning of the next header operation (or during the current data operation). Within this time, microprocessor has to update the information (header bytes or drive command register) and then write the header byte count to the Header Byte count register to indicate completion of update.

DISK FORMAT (DF) REGISTER (Address 35H)

Disk Format register, shown in *Figure 5.4,* is a write-only register and is usually updated when a different drive type is selected. This register controls some of the major format features of a disk, such as MFM, type of ECC/CRC, and configuring address marks.

D0	MFM	NRZ/MFM ENCODE	0 = NRZ 1 = MFM
D1	SAM	START WITH ADDRESS MARK	0 = Start with Preamble 1 = Start with AM
D2	HSS	HARD OR SOFT SECTOR	0 = Soft 1 = Hard
D3	FTF	FIFO TABLE FORMAT	0 = Use Registers 1 = Use FIFO
D4 D5	IH1 IH2	INTERNAL HEADER APPENDAGE	00 = None 01 = 16-bit CRC 10 = 32-bit ECC 11 = 48-bit ECC
D6 D7	ID1 ID2	INTERNAL DATA APPENDAGE	00 = None 01 = 16-bit CRC 10 = 32-bit ECC 11 = 48-bit ECC

FIGURE 5.4. Disk Format Register

MFM/NRZ Encode (MFM)

When writing to the disk, the DDC can output either MFM encoded data if MFM is high, or NRZ data if MFM is low.

Start with Address Mark (SAM)

If SAM is low, the format begins with the Preamble field followed by the Address Mark field. If SAM is high, the first field is Address Mark followed by the Preamble field. This supports ESDI or SMD drive formats.

Hard or Soft Sector (HSS)

The DDC can be configured for soft or hard sectored drives by setting bit HSS low or high, respectively.

FIFO Table Format (FTF)

If bit PTF is low, the header bytes are taken from the internal pattern registers during the disk formatting. If FTF is high, these bytes will be written from the FIFO through local DMA channel. This bit is used only during disk formatting.

Internal Header and Data Appendages (IH1, IH2, ID1, ID2)

The Internal Header Appendage (IH1 and IH2) and the Internal Data Appendage (ID1 and ID2) bits of the Disk Format register control CRC/ECC appendages for header and data fields. The appendage options which could be selected are no appendage (00), 16 bit CRC CCITT polynomial (01), and 32- and 48-bit programmable ECC codes (10, 11 respectively). If none of these internal ECC or CRC codes is selected, then an external header ECC code must be appended. Also, even if the internal codes are being appended, an external ECC code of up to 31 bytes may be added to encapsulate header, data and internal CRC/ECC fields.

STATUS (S) REGISTER (Address = 00H)

Figure 5.5 shows the flags in the Status register, which is a read-only register. This register provides status on current operation of the DDC. This includes DMA local and remote status, correction cycle status, operation error, and ready for next command's status. The flags are set or reset by conditions detected by the DDC. The flags are also reset when either the RESET input pin or RES bit in the operation command register are set. The flags in the Status register either provide the status of different operations in progress or the results of these operations.

D0	HF	HEADER FAULT
D1	NDC	NEXT DISK COMMAND
D2	HMC	HEADER MATCH COMPLETE
D3	LRG	LOCAL REQUEST
D4	RCB	REMOTE COMMAND BUSY
D5	LCB	LOCAL COMMAND BUSY
D6	CCA	CORRECTION CYCLE ACTIVE
D7	ED	ERROR DETECTED

FIGURE 5.5. Status Register

Header Fault (HF)

The Header Fault flag (HF) is set when a header field error is detected after a Compare Header operation. This is set when an ECC or CRC error is detected in any header field read. This may or may not be on the header that the DDC was looking for. During a disk operation if a header error was detected, and subsequently the correct sector was found, this bit will be reset. If the correct sector was not found, the DDC will timeout with the HF bit set. It is reset when the DDC is reset or when a new command is issued.

The HF will abort the operation immediately if the operation is a read header, and any header read has a CRC/ECC error. In this case no Error register bit is set.

Next Disk Command (NDC)

The Next Disk Command flag, when set, shows that the DDC is ready to receive a header byte update and another disk command. It is reset when a new disk command is issued to the DDC.

Header Match Completed (HMC)

In a Compare Header operation, after a header match, the Header Match Completed flag is set. This bit is reset at the end of the data operation. This flag is automatically set in Ignore and Write header operations or when any header field is encountered after a Read Header operation.

Local Request (LRQ)

The Local Request flag follows the LRQ exactly. It is set coincident with the LRQ output when the FIFO first requires a data transfer. The flag is reset whenever the LRQ pin is deasserted.

Remote Command Busy (RCB)

The Remote Command Busy flag is set at the start of a remote transfer operation and is reset at the completion of the last memory transfer. This can be used to determine if the remote DMA channel is in operation.

Local Command Busy (LCB)

The Local Command Busy flag remains set through the entire period the local DMA channel is busy in transferring data between the FIFO and buffer memory. This is the same function as the RCB except local DMA.

Correction Cycle Active (CCA)

The Correction Cycle Active flag is set at the beginning of a Correction Cycle (when the Start Correction Cycle bit is set in the Operation Command Register) and is reset at the end of the cycle whether the error is located or not.

Error Detected (ED)

Error Detected flag is set if any of the error flags in the Error register is set. This is the logical ORing of all the Error register bits.

ERROR (E) REGISTER (Address = 01H)

Error register, shown in *Figure 5.6,* is also a read-only register. The flags of this register are set by conditions within the DDC and reset by the next new command to the Drive Command register. The flags are also set low when either the RESET input pin is set low, or the Reset bit (RES) in the Operation register is set.

D0	HFASM	HEADER FAILED ALTHOUGH SECTOR MATCHED
D1	DFE	DATA FIELD ERROR
D2	SNF	SECTOR NOT FOUND
D3	SO	SECTOR OVERRUN
D4	NDS	NO DATA SYNCH
D5	DL	FIFO DATA LOST
D6	CF	CORRECTION FAILED
D7	LI	LATE INTERLOCK

FIGURE 5.6. Error Register

Header Failed Although Sector Matched (HFASM)

The HFASM (Header Failed Although Sector Matched) flag, when set, indicates that the Sector byte(s) of the header field match correctly but there is an error in other header byte(s). This flag can only be set if the Enable HFASM Function (EFH) bit of at least one of the Header Control registers is set high during a Compare Header operation. This bit will be set if any one of the header byte(s) with its EHF bit set matches but any other header bytes don't match. For example, assume a 6 byte header with the first two bytes having their EHF bit set. If during a compare header operation the first byte matched, but any of the 2nd through 6th bytes don't match this HFASM bit is set.

When executing a Compare Header-Check Data command, and this flag is set, the operation is aborted allowing the header bytes to be read from the FIFO for disk diagnostics. If this bit is set during a Compare Header-Read (or Write) Data, the command is aborted, but the header is not stored in the FIFO.

Data Field Error (DFE)

After a successful header match, if an internal CRC/ECC or external ECC error is detected during a Read Data or Check Data operation, the Data Field Error flag will be set.

Sector Not Found Error (SNF)

If the Header Match Completed flag of the Status register is not set for two consecutive index pulses in a Compare Header operation (i.e., the correct header was not found), then the Sector Not Found bit is set to indicate that the desired sector cannot be found. The operation is aborted and an interrupt is issued.

Sector Overrun (SO)

If during the time when data is being transferred between disk and the FIFO, either the SECTOR PULSE or INDEX PULSE inputs go active, then the sector is assumed to have overrun and the Sector Overrun flag (SO) is set. Operation is aborted. RGATE or WGATE are deactivated, and an interrupt is generated.

No Data Synch (NDS)

If an INDEX PULSE (hard or soft sectored drives) or a SECTOR PULSE (hard sectored only) is encountered while the DDC is looking to byte align on the first data synch byte (synch 1 or 2), this bit is set. Also if the DDC recognizes the first synch byte but not subsequent synch bytes then this bit is also set.

FIFO Data Lost (FDL)

This bit is set if the FIFO overflows during a Read Data operation. This normally would occur when the host does not allow the DMA to empty the FIFO faster than the Disk Data is being read. FIFO Data Lost is also set during a Write Data operation when the DDC empties the FIFO writing to the disk, and attempts to read the empty FIFO again. In either case the operation will be aborted.

Correction Failed (CF)

If by the end of a Correction Cycle (Data Byte Counter decrements to zero) the error has not been located, then the error is not correctable and the Correction Failed flag is set.

Late Interlock (LI)

If the Interlock Complete register is not written to by the time the next header field arrives, and the DDC is in Interlock mode, then the Late Interlock flag will be set.

START SECTOR (SC) REGISTER (Address = 12H)

The Start sector (and Number of Sector Operations Counter) facilitates multi-sector operations. This counter can be programmed to replace the header byte designated by the user to be the sector number. Thus, in a multi-sector operation, the Sector Counter is initialized with the sector number to start on. As a header is compared, and its data field is read or written, the sector counter is incremented at the end of that sector's header operation. This enables immediate operation on the next logical sector. Operation continues until the Number of Sector Operations Counter decrements to zero. The sector counter is enabled if bit substitute sector counter bit of any Header Control register is set high, and the contents of Sector Counter will be substituted for the corresponding Header Byte.

NUMBER OF SECTOR OPERATIONS COUNTER (NSO) REGISTER (Address = 13H)

In a multi-sector operation, the Sector Operations Counter is preset to the logical number of sectors to be consecutively operated on. It is decremented after every sector's header operation and when decremented to zero, terminates the active command.

HEADER BYTE COUNT (HBC)/INTERLOCK REGISTER (Address = 0FH)

This 4-bit read-write register, normally used during formatting, is loaded with the number of header bytes to be written to (or read from) disk. The allowable number of header bytes is from 2 to 6. On read-back, only the three least significant bits of this register are valid.

Another important function of this register is when the DDC is in the Interlock mode (explained in formatting section). During a multi-sector operation, if it is desired to update any header byte (for example in the case of disk formatting) or if the next drive command has to be changed, then this register must be written with the actual header byte count value after updating the header bytes. This will basically strobe the internal hardware to recognize that interlock (update) has occurred.

FIFO HEADER DIAGNOSTIC READBACK (HDR) REGISTER (Address = 36H)

This is a read-only register and allows the FIFO contents to be read one byte at a time. Normally, data or header bytes may be read for diagnostic purposes through this register (described later). There is no way to write to the FIFO except under DMA control. In order to read the header bytes in the same order as they are read from the disk, the Reverse Byte Ordering bit in the local transfer register must be reset.

5.1.2 Error Correction/Cyclic Redundancy (ECC/CRC) Registers and Counters

The ECC/CRC registers and counters are listed in *Figure 5.7.* These registers enable programming of various modes of ECC, the ECC pattern, and access to the ECC shift register for performing correction cycles. They are explained in the following paragraphs.

Name	Hex Address
ECC Shift Register Out0 Register	02H
ECC Shift Register Out1 Register	03H
ECC Shift Register Out2 Register	04H
ECC Shift Register Out3 Register	05H
ECC Shift Register Out4 Register	06H
ECC Shift Register Out5 Register	07H
Polynomial Preset (Byte0) Register	02H
Polynomial Preset (Byte1) Register	03H
Polynomial Preset (Byte2) Register	04H
Polynomial Preset (Byte3) Register	05H
Polynomial Preset (Byte4) Register	06H
Polynomial Preset (Byte5) Register	07H
Polynomial Tap (Byte0) Register	08H
Polynomial Tap (Byte1) Register	09H
Polynomial Tap (Byte2) Register	0AH
Polynomial Tap (Byte3) Register	0BH
Polynomial Tap (Byte4) Register	0CH
Polynomial Tap (Byte5) Register	0DH
ECC Control Register	OEH
Data Byte Count (LS) Register	08H
Data Byte Count (MS) Register	09H

FIGURE 5.7 ECC/CRC Registers and Counters

ECC SHIFT REGISTER OUT0–OUT5 REGISTERS (Address = 02–07H)

The 48-bit long CRC/ECC shift register of the DDC can be read through these 6 read-only registers at any time. If 32 byte ECC is used then only registers 0, 1, 4, and 5 are used. After a correction cycle has occurred, these registers contain the ECC syndrome bits. The memory address of the sector in error and the data bit in error are calculated by the μP, and then the bits of these registers are XORed with the data in order to correct the error (assuming the error is correctable).

POLYNOMIAL PRESET (PPB) 0–5 REGISTERS (Address = 02–07)

The selected ECC polynomial preset pattern is loaded into the ECC/CRC shift register from these six Polynomial Preset registers. These are write only registers. The preset bit pattern could be all ones, all zeroes or a combination. This is the value the ECC shift regsiter is loaded with prior to shifting in the ECC pattern. The most significant bit of PPB5 is the most significant polynomial bit, X^{47}, and the least significant shift register tap is the least significant bit of PTB0. For 32-bit ECC, PPB2 and PPB3 are set to all zeroes, and are not used.

POLYNOMIAL TAP BYTE (PTB) 0–5 REGISTERS (Address = 08–0DH)

The ECC shift register is tapped at every bit by an XOR element. Wherever an exclusive-OR tap is required into the 32-bit or 48-bit shift register a zero should be set in the corresponding PTB0–5 bits. All polynomial elements not in the equation (and hence not tapped) must be disabled by setting all these bits to a one. The tap X^{32} (or X^{48} for 48-bit polynomial) is always present and is not programmable. The taps X^{31} (or X^{47}) to X^0 are fully programmable. The MSB of PTB5 corresponds to the most significant tap, X^{47}, and the LSB of PTB0 is the least significant tap, X^0. For 32-bit ECC, PTB2 and PTB3 are not used and must be set to all ones. In this case PTB5's MSB becomes X^{31}.

ECC CONTROL (EC) REGISTER (Address = 0EH)

The ECC Control register, shown in *Figure 5.8,* is a write-only register. This register works in conjunction with the Disk Format register to set the ECC modes. The bit description is given below.

D0 D1 D2 D3	CS0 CS1 CS2 CS3	CORRECTION SPAN SELECT 0011 thru 1111	 3 bit span thru 15 bit span
D4	HE	ENCAPSULATION HEADER	0 = Encapsulated 1 = Not Encapsulated
D5	IEO	INVERT ECC OUT	0 = Normal 1 = Inverted
D6	IDI	INVERT DATA IN	0 = Normal 1 = Inverted
D7	DEN	DATA ENCAPSULATION	0 = Encapsulated 1 = Not Encapsulated

FIGURE 5.8. ECC Control Register

Correction Span Select Bits (CS0–CS3)

The number of bits which the ECC circuit attempts to correct, generally known as the correction span, is determined by CS0–CS3. Errors longer than the correction span will be

treated as non correctable errors. The allowable correction span for 32-bit ECC is 3 to 15 bits and for 48-bit ECC it is 3–15 bits. Setting the CS bits to any correction span that is outside the maximum allowable range of 3–15 bits causes the CRC/ECC to default to a 3 bit correction span.

Header Encapsulation (HEN)

When this bit is reset, the bit patterns of the Synch and/or Address Mark fields are included in ECC/CRC calculations. Some disk formats want these bytes included in check bit calculations (i.e., IBM 3740 floppy format). When this bit is set the Synch and/or Address Mark fields are excluded from the CRC/ECC calculation, and only the header field bytes are included.

Invert ECC Data Out (IEO)

When the shift register data out bit is set high, all the data and check bits coming out of the ECC shift register are inverted in a disk write operation. Otherwise the ECC data is not inverted.

Invert Data in (IDI)

This bit controls data and check bits when they enter the ECC shift register during a disk read operation. When this bit is set high, both data and check bits will be inverted. When low true data is input.

Data Encapsulation (DEN)

The DEN bit performs the same function for the data field as the HEN bit does for the Header field. When reset DEN will include the Synch and Address Mark fields in the CRC or ECC calculations. If set these fields are not included in the check bit calculations.

DATA BYTE COUNTER REGISTERS
(Address = 08H, 09H)

The Data Byte Counter registers are used during a correction cycle, and are preset by the μP prior to starting a correction cycle. They are set to the sum of the number of bytes in the data and ECC fields of the sector just read. During the correction cycle, the data byte count is decremented after shifting by 8 bits in the ECC shift register each byte. At the completion of the correction cycle, and if the error is correctable, the contents of the data byte counters are added to the starting address of the sector in error to determine the location of the memory byte or bytes in error. Details on this, are provided later.

5.1.3 Format Pattern and Count Registers

The Pattern, Count and Control registers used during disk formatting are listed in *Figure 5.9* and explained in the following paragraphs.

PATTERN REGISTERS
(Address = 30–33H, 3A–3FH, 14–19H)

The pattern registers hold byte information for the various fields of a formatted disk. These registers are written to the disk during a format operation. The Synch or Address Mark, Header, and ID postable pattern are read and compared to the pattern registers during a Compare Header operation. The Data Address Mark, and Data Synch are compared when doing a data field read, and are written to the disk during a disk data write or format. Associated with each pattern register (except the header pattern registers) is a byte repetition counter register that sets the field length, described below.

All the pattern registers listed in *Figure 5.9* are preloaded with the value of their respective fields such as ID and Data fields. The fields which are allowed in the DDC pattern registers are ID and Data Preamble, Address Mark, Synch, Postamble and Gap. Up to six header byte patterns can be programmed, thus enabling a header field of six bytes (excluding synch and preamble).

Name	Hex Address
ID Preamble Register	31H
ID Preamble Byte Count Register	21H
ID Synch #1 (AM) Pattern Register	32H
ID Synch #1 (AM) Byte Count Register	22H
ID Synch #2 Pattern Register	33H
ID Synch #2 Pattern Register	23H
Header (Byte0) Pattern Register	14H
Header (Byte0) Control Register	24H
Header (Byte1) Pattern Register	15H
Header (Byte1) Control Register	25H
Header (Byte2) Pattern Register	16H
Header (Byte2) Control Register	26H
Header (Byte3) Pattern Register	17H
Header (Byte3) Control Register	27H
Header (Byte4) Pattern Register	18H
Header (Byte4) Control Register	28H
Header (Byte5) Pattern Register	19H
Header (Byte5) Control Register	29H
ID External ECC Byte Count Register	2BH
ID Postamble Pattern Register	3CH
ID Postamble Byte Count Register	2CH
Data Preamble Pattern Register	3DH
Data Preamble Byte Count Register	2DH
Data Address Mark Pattern Register	3EH
Data Address Mark Byte Count Register	2EH
Data Synch Pattern Register	3FH
Data Synch Byte Count Register	2FH
Data Format Pattern Register	3BH
Sector Byte Count (L) Register	38H
Sector Byte Count (H) Register	39H
Data External ECC Byte Count Register	2AH
Data Postamble Pattern Register	30H
Data Postamble Byte Count Register	20H
Gap Pattern Register	3AH
Gap Byte Count Register	34H

FIGURE 5.9. Format Registers and Counters

During an operation these registers must not be read, as this will interfere with the DDC's internal access to these registers. This could cause internal PLA's to misinterpret these registers, and lead to sporadic misbehavior of the DDC. These registers may be written to any time. If written to during an operation they will take effect immediately.

One data byte pattern register is provided. This pattern is used during a format operation as the data field byte. It is repeated for the length of the data field in the sector.

BYTE COUNT REGISTERS
(Address = 20–23H, 2A–2FH)

The Byte Count registers determine the number of times each field's pattern can be repeated. All of ID and Data Preamble, Address Mark, Synch, Postamble, External ECC, and Gap patterns can be repeated for maximum 31 times. The Gap pattern, on the other hand, can be repeated for 255 times. As mentioned earlier, there can only be six Header bytes and 64K data bytes in any format for the selected

drive. The length of Header and Data bytes is controlled by the Header Control and Sector Byte Count registers, respectively (described below).

During an operation these registers must not be read, as this will interfere with the DDC's internal access to these registers. This could cause internal PLA's to misinterpret these registers, and lead to sporadic misbehavior of the DDC. These registers may be written to any time. If written to during an operation they will take effect immediately.

HEADER BYTE CONTROL REGISTER
(Address = 24–29H)

These six read/write registers control the associated six header bytes. Each of the six registers is 4-bits long and performs the same functions. One of these is shown in *Figure 5.10* and the functional description of each bit is given in the following.

D0	HBA	HEADER BYTE ACTIVE	0 = Header Byte Disabled 1 = Header Byte Enabled
D1	SSC	SUBSTITUTE SECTOR NUMBER	0 = Header Byte Used 1 = Sector Counter Used
D2	EHF	ENABLE HFASM FUNCTION	0 = Header Used Normally 1 = Header Interpreted as Sector Number
D3	NCP	NOT COMPARE	0 = Header Used for Compare 1 = Header Disabled

FIGURE 5.10. Header Byte Control Register (One of Six)

Header Byte Active (HBA)

This bit determines whether the corresponding Header byte is to be included in the Header field or not. If set low, the corresponding header byte will be omitted from the header field and setting it high will include the corresponding byte in the header field. Only 4 out of 6 header bytes can be disabled. Also, only two consecutive header bytes can be disabled.

Note: All the other bits in this register must also be set to zero if the header byte is to be disabled.

Substitute Sector Counter (SSC)

This bit when set high, enables the DDC to substitute the Sector Counter register's contents in the header byte pattern register instead of the actual header byte during a write operation, or to compare the Sector Counter register's contents with the corresonding header byte during a read operation. This is normally done in a multi-sector operation to enable automatically incrementing the sector number.

Enable Header Failed Although Sector Matched Function (EHF)

If bit EHF of any header control register is set high, then the associated header byte is designated as that byte that must match in order to enable generation of an HFASM. In this mode, if this header byte matches but any of the other header bytes don't, then an HFASM error and an interrupt is generated. In a Compare Header-Data operation, the header bytes are loaded into the FIFO and can be examined by the host by reading the FIFO Diagnostic Register (Address = 36H). This can also be used during a Compare Header-Read Data, but the FIFO will not store the header bytes, see Error Register description, HFASM.

When this bit is reset the corresponding header byte is compared normally.

Not Compare (NCP)

When this bit is set low, the Header byte will be written and compared normally. On the other hand, if this bit is set high, the corresponding Header byte will always be declared matched regardless of the actual comparison results. In other words the comparison is disabled. This can be used to read a group of sectors.

5.1.4 DMA Registers and Counters

The DMA registers and counters enable programming of the DMA start address (in single or dual channel mode), transfer length, and the various modes of operation. The Operation Command register controls actual starting of the Remote DMA operation. The DMA registers are listed in *Figure 5.11* and explained in the following paragraphs.

Name	Hex Address
DMA Sector Counter	37H
Local Transfer Register	36H
Remote Transfer Register	37H
Remote Data Byte Counter (L)	1AH
Remote Data Byte Counter (H)	1BH
DMA Address (Byte 0) Counter	1CH
DMA Address (Byte 1) Counter	1DH
DMA Address (Byte 2) Counter	1EH
DMA Address (Byte 3) Counter	1FH

FIGURE 5.11. The DMA Registers and Counters

DMA SECTOR COUNTER
(Address = 37H)

This read only register is used only when the DDC is configured in the dual channel tracking mode. This counter keeps track of the number of sectors transferred by the remote DMA channel (local RAM to/from system), and the local DMA channel (DDC to/from local RAM). When this register is 0, remote channel transfers are inhibited. This counter ensures that the source channel will not overtake the remote channel. This eliminates the chances of overwriting while transferring data to or from the local memory, or transferring non-data.

LOCAL TRANSFER (LT) REGISTER
(Address = 36H)

This write-only register, shown in *Figure 5.12*, controls the data transfers between the DDC and buffer memory, using the local DMA channel or single channel mode. It is configured at the time of initialization and normally need not be written to again. The Local Transfer register is not affected by reset or abort operations.

Local DMA Enable (SLD)

This bit when set high, enables the local DMA channel, Tracking or Non-Tracking. If this bit is not set high, the on-chip DMA will not transfer data. This bit is used to enable control of starting/stopping a DMA operation. If it is permanently disabled, external DMA circuitry can be used.

Local Word Data Transfer (LWDT)

This bit determines the length of the data word to be transferred between the DDC and buffer memory. When set to 0, single 8-bit bytes are transferred each DMA cycle and the address increments by 1. If this bit is set, 16-bit words are transferred each DMA cycle and the address increments by 2.

Reverse Byte Ordering (RBO)

This bit controls the order of bytes in a 16-bit word transfer. When RBO is set low, the first byte to be read from the disk will be placed in the least significant half of the word (AD0–AD7) and when RBO is set high, the LS byte will be mapped to AD8–AD15. This is only valid for data entering the FIFO. This byte should be reset for 8 bit transfers.

Local Slow/Fast Read and Write (LSRW)

This bit can add one wait state cycle to the DMA transfers. When this bit is reset the Read and Write cycle is 4 clock periods, but if this bit is set, one wait state is added and the read or write cycle is 5 clock periods. This extends the RD and WR strobes by one bus clock period, and allows the DDC to access slower memories.

Long Address (LA)

This bit determines the Local DMA address bus width and is only valid if Local DMA is enabled and the Remote channel is disabled. When Long Address is low, 16 address bits are issued and strobed by ADS0 pin. When Long Address is high, 32 address bits are issued, the lower 16 bits are strobed by ADS0 pin, the upper 16 address bits are strobed by ADS1/RRQ pin. The most significant 16 address bits are only issued when a rollover from the least significant 16 address lines occurs or on the first DMA cycle of a multi-byte transfer. When the most significant 16 address bits are issued, that DMA cycle is 5 clock periods long if no internal or external wait states are used.

Dump FIFO/Exact Burst (LTEB)

This bit controls how the data is burst to/from the FIFO. If this bit is reset the FIFO will fill or empty completely. When the disk is being read, the FIFO will wait until the FIFO is filled to the programmed threshold. At this time the DDC will completely empty the FIFO to buffer RAM even if more data entered the FIFO during the burst. During a disk write, when the FIFO is emptied to the programmed threshold, the DMA will fill the FIFO.

When this bit is set, the DMA will only transfer a fixed number of bytes to/from the system. For reading the disk, when the FIFO fills to the programmed threshold, the DMA will burst the exact number of bytes that are in the FIFO. Any bytes entering the FIFO after the burst begins will be transferred at the next burst. For a disk write, when the FIFO empties to the selected threshold, an exact number of bytes will be DMAed to the FIFO, whether the FIFO has emptied more or not.

Local Burst Length Select (LBL1, LBL2)

Bits LBL1 and LBL2 offer different burst lengths and the thresholds according to when data will be transferred to or from the FIFO. These burst lengths/thresholds could be 2, 8, 16 or 24 bytes, if these bits are programmed, 00, 01, 10, or 11, respectively.

REMOTE TRANSFER REGISTER (Address = 37H)

This write-only register, shown in *Figure 5.13*, determines the mode of transfers between the local buffer and the system I/O port (remote DMA channel) if the DDC is in dual channel mode. The register should be loaded at initialization and normally need not be written to again. It is not affected by reset or abort.

Remote DMA Enable (SRD)

This bit, when set high, configures the DDC in dual channel DMA mode, and enables the remote DMA channel.

Remote Word Data Transfer (RWDT)

The data bus may be configured to be either 8 or 16 bits during remote transfers between buffer memory and main system. If RWDT is high, transfers are 16 bits wide, and the remote address information is incremented by 2 each memory cycle and bit A0 remains low. If RWDT is low transfers are 8 bits wide, and the remote address increments by 1.

Enable External Wait (EEW)

If EEW is high, the EXTERNAL STATUS pin of the DDC will be enabled to supply wait states in both the local and remote DMA bus cycles. A side effect using this feature is that external synchronization and external ECC cannot be used. When EEW is low, no external wait states can be inserted, and external synchronization and external ECC may be used.

Remote Slow Read/Write (RSRW)

This bit allows for slower memory or slower system cycle time and is only valid if the Remote DMA Enable bit is high. In this case, if RSRW is set, each remote DMA cycle becomes five clock periods rather than four, and both the $\overline{RD}$ and $\overline{WR}$ strobes are widened by one clock period.

D0	SLD	SELECT LOCAL DMA	0 = Disabled 1 = Enabled
D1	LWDT	LOCAL WORD DATA TRANSFER	0 = 8-Bit 1 = 16-Bit
D2	RBO	REVERSE BYTE ORDER	0 = AD0–7 = LSB, AD8–15 = MSB 1 = AD0–7 = MSB, AD8–15 = MSB
D3	LSRW	LOCAL SLOW READ AND WRITE	0 = 4 Clock Transfer 1 = 5 Clock Transfer
D4	LA	LONG ADDRESS	0 = 16-Bit 1 = 32-Bit
D5	LTEB	LOCAL TRANSFER EXACT BURST	0 = Transfer Until FIFO Empty 1 = Transfer Exact Burst
D6 D7	LBL1 LBL2	LOCAL BURST LENGTH (FIFO THRESHOLD)	00 = 1 Word/2 Bytes 01 = 4 Words/8 Bytes 10 = 8 Words/16 Bytes 11 = 12 Words/24 Bytes

FIGURE 5.12. Local Transfer Control Register

Tracking Mode (TM)

This bit configures the DDC in Tracking or Non-Tracking DMA modes when it is set high or low, respectively. In Non-Tracking mode, the DMA channels are independent and addresses are allowed to overlap, while in Tracking mode, channel addresses are maintained at least one sector apart.

Remote Transfer Exact Bursts (RTEB)

When the DDC is performing a remote transfer, the condition of this bit determines when RRQ is de-asserted after the exact number of words or bytes, specified by RBL1 and RBL2, have been transferred. If RTEB is low, RRQ wil remain asserted until the whole count, specified by the Remote Data Byte Counter, has been transferred.

Remote Burst Length (RBL1, RBL2)

These bits select the burst transfer lengths during a remote transfer operation between buffer memory and a system I/O port if RTEB is set high. These length could be 2-byte (1-word), 8-byte (4-word), 16-byte (8-word) and 32-byte (16-word).

D0	SRD DMA	SELECT REMOTE	0 = Disabled 1 = Enabled
D1	RWDT	REMOTE WORD DATA TRANSFER	0 = 8-Bit 1 = 16-Bit
D2	EEW	ENABLE EXTERNAL WAIT STATE	0 = Disabled 1 = Enabled
D3	RSRW	REMOTE SLOW READ AND WRITE	0 = 4 Clock Transfer 1 = 5 Clock Transfer
D4	TM	TRACKING MODE	0 = Non-Tracking 1 = Tracking
D5	RTEB	REMOTE TRANSFER EXACT BURST	0 = Transfer Whole Count 1 = Transfer Exact Burst
D6 D7	RBL1 RBL2	REMOTE BURST LENGTH	00 = 1 Word/2 Bytes 01 = 4 Words/8 Bytes 10 = 8 Words/16 Bytes 11 = 12 Words/24 Bytes

FIGURE 5.13. Remote Transfer Control Register

REMOTE DATA BYTE COUNT REGISTERS

(Address = 1AH (LSB), 1BH(MSB))

These registers determine the byte count required in a remote data transfer. They are preloaded with a maximum count equal to the desired total DMA transfer, and are decremented by the DDC after each transfer until a count of zero is reached. This counter is 16 bits wide, therefore up to a total of 65,536 bytes can be transferred. Presetting this register to all zeroes transfers 65,536 bytes. The count can be read at any time during the transfer.

This register is also used in tracking mode DMA to keep track of whether a sector is ready to be transferred by the remote channel.

DMA ADDRESS (BYTE0-3) REGISTERS

(Address = 1CH–1FH)

The DMA address registers issue dual channel local and remote addresses during the local and remote transfers or single channel addresses. When using the dual channel DMA mode, registers 0 (LSB) and 1 (MSB) are used for holding and incrementing the local DMA channel address. Registers 2 (LSB) and 3 (MSB) contain the address information for remote DMA transfers. These registers can be preset to any address (within 64k) and can be read at any time.

In single channel DMA mode, the 4 registers are concatenated to form a 32-bit address, thus the single DMA channel can address up to 4 Gigabytes.

5.2 DDC COMMANDS

The DDC can be configured to perform various disk and related operations. These include disk read and write operations, error correction operation, formatting operations, and DMA operations. DMA and error correction commands are considered separately later. To understand the operation better, it is useful to break up the DDC's execution of a command into three phases:

1. Command Entry — μP loads bytes and command word prior to execution.
2. Command Execution — Once loaded the DDC performs the operation.
3. Result — After execution is terminated the μP reads the DDC to determine whether an error terminated the operation.

Since the DDC is primarily reading and writing to the disk drive, it has a rich set of disk read and write functions and each of these can be used is several modes. This creates a large array of commands that provides versatility. However there are about 10–15 commands/modes that would normally be used. This section will present the basic command operations first, then common commands are shown as combinations of the operations and finally a simplifying list is created.

A typical read/write command is composed of two operations; ID field operations and disk data field operations. Bits DO2, DO1, HO2, HO1; FMT in the Drive Command Register, determine the command to be executed. The two least significant bits enable some specific options to the commands. A list of all the possible commands is given in *Figure 5.13.* These commands are executed by setting up all the registers with the desired modes and header information, and as the last step the Drive Command register is loaded with the command. Once loaded the DDC will execute the command.

The header and data operations are described individually below. They repeat some of the information given in the register bit description in Chapter 4. Following the header and data operations, the more useful combinations of these are described.

5.2.1 Header Operations

Ignore Header

The DDC will use the preamble for PLO locking, and will check for the ID field's byte synch fields. The DDC will ignore the actual header contents, and treat any values as a match, and a data field operation will proceed on the subsequent data field.

DO2	DO1	HO2	HO1	FMT	
0	0	0	0	0	No Operation, No Format (No Operation)
0	1	0	0	0	Ignore Header, Check Data, No Format
1	0	0	0	0	Ignore Header, Write Data, No Format
1	1	0	0	0	Ignore Header, Read Data, No Format (Recover Data)
0	1	0	1	0	Compare Header, Check Data, No Format
1	0	0	1	0	Compare Header, Write Data, No Format (Normal Write)
1	1	0	1	0	Compare Header, Read Data, No Format (Normal Read)
0	1	1	0	0	Write Header, Check Data, No Format
1	0	1	0	0	Write Header, Write Data, No Format
1	0	1	0	1	Write Header, Write Data, Format (Normal Format)
0	1	1	1	0	Read Header, Check Data, No Format (Get Header Info)
1	1	1	1	0	Read Header, Read Data, No Format
0	0	0	1	X	Compare Header, No Data Operation *** ILLEGAL COMMAND ***
0	0	1	0	X	Write Header, No Data Operation *** ILLEGAL COMMAND ***
1	1	1	0	X	Write Header, Read Data *** ILLEGAL COMMAND ***
0	0	1	1	X	Read Header, No Data Operation *** ILLEGAL COMMAND ***
1	0	1	1	X	Read Header, Write Data *** ILLEGAL COMMAND***

FIGURE 5.14. The DDC Commands

Compare Header

This operation usually preceeds a normal disk read or write in which a particular sector is operated on. After preamble and synch fields are compared, the DDC compares every byte in the header to the pattern registers. Only if a complete match of the header bytes and no CRC/ECC error occurs, then the data field operation is executed.

Write Header

The DDC will write header information typically when formatting the disk, but operations allow the DDC to write individual headers in a hard sectored disk as a method of correcting a header. In this operation the entire ID field is written, preamble, both byte synch fields, header bytes, and CRC/ECC bytes.

Read Header

This is used when the host desires to know what a header is, usually when trying to determine where a disk drive head is located. This operation will compare the synch fields and then read the header bytes into the FIFO. CRC/ECC is checked.

5.2.2 General Data Operations

No Data Operation

This bit combination is valid only when the header operation is ignore header. Using this with other header operations will cause "unpredictable" results. Using this with the Ignore Header function is a NOP command and can be used when the user wishes to change non-command bits in the command register without executing a disk command.

Check Data

This is essentially a data NOP. If executed the DDC will read the data field just like a read data operation, but no data will be transferred to the system. The data field CRC/ECC is checked at the end of the operation.

Read Data

To fetch data from the disk drive this operation is used. The DDC first checks the data synch field to byte align and then it reads the data from the disk drive and sends it to the FIFO for transfer to external memory. After all bytes have been transferred, the data field's CRC/ECC bytes are checked.

Write Data

This is the inverse of the read data command. The DDC will begin by writing the data preamble field and synch fields. Data is input from the external memory into the FIFO, and this data is written to the disk. During the writing of data, CRC/ECC is being generated, and is appended to the data field.

5.2.3 When a Command Starts

As a command is loaded, the microprocessor can decide when the DDC should start the operation. There are two choices, which are programmed by loading bit SAIS in the Drive Command Register, Start at Index or Sector.

If this bit is set, the operation will be started at the beginning of the next sector (if hard sector) or immediately (if soft sectored). This is normally used for a normal read or write operation. Since the operation starts asynchronously to where the head is located over the track, starting immediately means that the sector will be read/written within one disk revolution worst case. If the command is started on an index pulse, it is likely that the head will pass the sector before it gets to the index to start the command, wasting a full revolution.

If this bit is reset when a command is loaded, the DDC will wait for the disk to revolve until the index pulse is seen. This mode would usually be used for track oriented operations. For example, a format and/or multi-sector operation would be most useful if started at the beginning of a track rather than the middle.

It is possible to execute any valid command starting either at the next sector or at the index pulse, thus many specialized command combinations are possible.

5.2.4 Multi-Sector Versus Single Sector Operations

All of the disk command op codes listed below have a multi-sector bit that determines whether the operation is a multi-sector or single sector operation. The DDC can be programmed to read one specific sector, several sectors, or an entire track in one operation. In a multi-sector operation the DDC will read a group of logically or physcially contiguous sectors. Logically contiguous means the DDC reads sectors with sector numbers that are in numerical order, but need not appear physically in order on the drive's track. For example, sectors are numbered physically on the drive as 5, 4, 6, 1, 3, 2, will be read 1, 2, 3, 4, 5, 6. This enables interleaving sectors. In order to do this the multi-sector bit must be set and several other registers and modes must be determined.

Usually a single sector operation can be considered the default operation, and additional steps must be taken to execute a multi-sector command. To review, a single sector operation will first execute a header operation followed by a data operation and then terminate. For a normal read or write, the header bytes are compared to the header pattern registers, and when the desired header is found the data field is operated on.

In a multi-sector operation, the DDC will re-execute the same command over again. The number of times the command is executed depends on the value programmed into the Number of Sector Operations Register (Address = 13H) with a maximum of 255. For most read or write commands, multi-sector operations will also need to select and program the Sector Counter (although some commands won't). For a given disk ID format one of the header bytes is the sector number. The header byte that has been designated the sector number must have its control register programmed to substitute the sector counter for the header pattern register. The sector counter is then programmed with the number of the first sector to be read/written.

After the command starts, the first sector is operated on. Then the Sector Counter increments and the Number of Sector Operations Counter decrements. If this has not reached zero the command is re-executed until the Number of Sector Operations Counter does reach zero. Programming a 1 into the Number of Operations Counter will cause the command to execute once.

The preceding describes how a multi-sector read or write would normally be executed. An example of when the Sector Counter may not need to be enabled might be a track dump command which is a read header-read data. Since sector numbers are not compared all the header fields the data fields can be sequentially read.

5.2.5 Interlock Mode Operation

The Interlock mode can be used to enable a microprocessor to update commands or parameters on the fly just after the previous command finishes with the header operation. This enables the μP to execute a series of commands on contiguous sectors. These commands may be the same one repeated, a format for example, or a different one executed sequentially. Interlock is enabled by setting the Interlock bit, and enabling the header interrupt in the Operation Command register, or polling the Next Disk Command status bit.

In the Interlock mode, and when a command is first issued, an interrupt at the end of the header operation informs the μP that the DDC is ready to accept a new command. The μP updates the header, or command registers, and lastly writes to the Header Byte Counter Register. This update must take place before the end of the CRC/ECC field of the present sector. If the Header Byte Counter is not written to before the header of the following sector, the DDC will set the Late Interlock bit in the Error Register, and abort the operation.

The Interlock mode can be used either in single sector or multi-sector operations. In a single sector operation, the μP updates all the header and control registers. It then writes the new command into the Disk Command register, and finally writes to the Header Byte Counter. This enables different commands to act on physically sequential sectors, i.e. read to one, write from the next.

In a multi-sector operation, the μP sets the Number of Sector Operations to the number of sectors to be read or written. The μP writes the command to the Disk Command register once at the beginning of an operation. Then the μP updates the header and mode registers after each interrupt. Finally, before the end of each data ECC field the Header Byte Counter is loaded. The DDC will automatically repeat the command until the Number of Sector Operations counter reaches zero. This command mode is useful for operating on physically sequential sectors that are not logically sequential. For example, formatting a track with interleaved sector numbers.

While executing in interlock mode, any pattern or count registers may be written to. However, it is recommended to not write to data pattern or count registers, as timing of the μP write relative actual disk data being operated on will determine whether the write will effect the present or next sector. During the operation, the pattern or count registers **must not** be read, as this will cause spurious operation.

Actually, the Interlock mode is somewhat misleading. For any command being executed, the DDC will be ready for the next command after it has successfully completed a header operation. The only action the Interlock mode takes is enabling an Error bit that tells the μP when it didn't update the DDC in time.

5.2.6 Command Termination, Resetting, and Re-enabling

Once a command starts execution, it will perform its desired task, or an error will be encountered that will prevent the command from executing. These errors could result from reading the disk, losing data while not transferring it fast enough, or not finding the header it is looking for after two index pulses have occurred. In these cases, the DDC will terminate the operation, set an error flag, and set itself into an error state from which it must be reset before the next command can be executed.

The errors that the DDC recognizes are listed in the Error register. Errors caused by corrupted disk data or format are:

Data Field Error
No Data Synch
Sector Not Found
Sector Overrun

Errors caused by not transferring data or parameters to the disk fast enough are:

FIFO Data Lost
Late Interlock

Errors that occur because the disk controller is looking for a sector header that is non-existent, or the disk head is on the wrong track are:

Sector Not Found
Header Failed Although Sector Matched

Additionally, during a correction cycle an error will be indicated if the correction failed and a Correction Failed error is set.

When an error occurs the DDC will terminate the command, and will issue an interrupt (if enabled). Once the error is flagged, the CPU must read the error register and then reset the DDC.

To reset the DDC, the CPU first must set the Reset bit in the Operation Command register. Then it must reset the Reset bit. This has reset the DDC into the default state. Now the DDC can be re-enabled by setting the Re-Enable bit high, and the DDC is then ready to receive the next command.

5.2.7 Summarizing Most Useful DDC Commands

As one can see there are a multitude of possible commands that the DDC can implement, and the header-data operations with the various modes tend to be very cryptic. To try to simplify the commands, *Figure 5.15* lists most of the common commands, a mnemonic, and the command op code. These commands assume the disk format is fairly standard, with the header at least containing one byte for the sector number. These are by no means all of the commands. Some specialized ones may be desirable, and can be assembled from the previous descriptions. Or, if the user decides to be creative with the header format other commands or modes may be useful, and maybe encryption/decryption of the header for data security could be implemented.

These commands are header-data operations with the multi-sector and start on sector or index bit configured to their most common way. For example, Read Sector (SRD) starts on a sector pulse (or immediately) and is not multi-sector. Multi-Sector Read Track is multi-sector and normally would start on the index pulse (but doesn't have to) so that the entire track's data can be read starting at the physical beginning of the track.

The read, write, format track commands assume that the Number of Sector Operations register is loaded with the number of sectors per track.

The logical multi-sector read, write commands assume that either the Sector Counter is enabled, or the Interlock mode is used.

Command Name		Op Code	
Read Single Sector	RDSS	11010010	D2H
Read Sector ID	RDID	01110010	72H
Read Multi-Sector Logical	RDMS	11010110	D6H
Read Track	RDTK	11010100	D4H*
Read Track Blind	RDTB	11000100	C4H*
Read ID Multi-Sector	RDIM	01110100	74H
Read Track Data/ID	RDDI	11110100	F4H*
Write Single Sector	WRSS	10010010	92H
Write Multi-Sector Logical	WRMS	10010110	96H
Write Track	WRTK	10000100	84H*
Format Track	FMTK	10101100	ACH
Format Track No Gap	FMNG	10100100	A4H
Find ID	FNID	01010010	52H
Find ID Multi-Sector	FNMS	01010110	54H
Recover Header	RCID	01100010	62H
Re-Enable Controller	RENB	00000001	01H
No Operation	NOP	00000000	00H

***Note:** For an entire track operation, the Number of Sector Operations Counter should be set to the number of sectors per track.

FIGURE 5.15. Common Configurations of the Command Bits

SINGLE SECTOR READ (Compare Header-Read Data)
Op Code = 11010010
This command is used to perform a normal disk read operation by disabling the multi-sector operation and starting immediately/sector pulse. The header bytes loaded into the DDC are compared to header information read off the disk drive. The DDC continues to scan the drive until a match is found. Once a header has matched, data in the subsequent data field is read from the disk and transferred to the system via the internal FIFO and DMA operations.

READ SECTOR ID (Read Header-Check Data)
Op Code = 01110010
This is a single sector command, which starts immediately. It will read the first sector header that the drive head passes over, and transfers it to the external memory. This is useful if the system gets lost and would like to know where the drive head is without recalibrating the drive to track zero.

READ MULTI-SECTOR LOGICAL
(Compare Header-Read Data)
Op Code = 11010110
This is a multi-sector command that starts immediately. There are two modes that can be used for this command. One is to use the sector counter to sequentially read logical

sectors. The second is to use the Interlock mode. If the sector counter is used then the logical sectors are read sequentially by sector number. In the Interlock mode the logical sectors can be read in any logical sequence, depending on μP update of header bytes.

This command can be modified to do a single sector read in multi-sector mode, by setting the Number of Operations to one and, preferrably, changing the Start bit from start on index to start immediately.

READ TRACK (Compare Header-Read Data)
Op Code = 11010100
The Read Track command has the same op code as the Multi-Sector Logical Read, except that the command is started on an index pulse. This will cause a read of all sectors ensuring that all other header bytes are compared and header CRC/ECC is checked. Of course the Number of Sector Operations counter must be set to the number of sectors per track. If the track is to be read in a logical order, then the sector number header byte can be compared to the Start Sector register. For a physically contiguous read the sector number header byte can be set to not compare.

READ TRACK BLIND (Ignore Header-Read Data)
Op Code = 11000010
This command will not compare the header field for a match, but will read the first data field that the DDC encounters and all subsequent data fields no matter what the header contains. The DDC will read in the data from the drive and DMA it to external memory.

READ ID MULTI-SECTOR (Read Header-Check Data)
Op Code = 01110100
This is a multi-sector command that starts at an index pulse and reads every header on the track (assuming the Number of Sector Operations equals the number of sectors on the track). This can tell the system what the entire track's header format.

READ TRACK ID AND DATA (Read Header-Read Data)
Op Code = 11110100
The read track command uses the Read Header-Read Data command in multi-sector mode, starting on the index pulse, and setting the Number of Sector Operations counter to the number of sectors on a track. This will read both the header and data fields of all sectors on a track. This can be used as a diagnostic tool to dump the entire contents of the disk drive's track.

WRITE SINGLE SECTOR (Compare Header-Write Data)
Op Code = 10010010
This command is used to perform a normal disk write operation to an individual sector. The multi-sector bit is reset and the command is started on a sector pulse or immediately. The header bytes are compared as in a disk read. Once the header matches, data is written to the associated data field by the DDC.

WRITE MULTI-SECTOR LOGICAL
(Compare Header-Write Data)
Op Code = 10010110
This is the same as the Multi-Sector Logical Read command. This one starts immediately, and can use the Interlock mode or Sector counter to change the header information. The sector counter enables logically sequential sector writing, and Interlock mode enables μP to update sectors on the fly.

As in the read command, by programming the Number of Sector operations counter to 1 results in a multi-sector mode single sector write. Also the start operation bit should be set to start on a sector pulse.

WRITE TRACK (Compare Header-Write Data)
Op Code = 10010100
The Track Write is similar to the Multi-Sector Logical Write command, but several modes are set up differently. First this command starts on an index pulse, and second the header byte corresponding to the sector number is programmed to not compare. This causes every sector to be written to in a physically sequential order.

FORMAT TRACK (Write Header-Write Data)
Op Code = 10101100
This command is commonly used when disk formatting is to be performed. The format bit in the disk command register normally should be set to execute this command. The entire ID Field and Data Field, one or more sectors, are written to the disk either from the header byte registers or from the FIFO, depending on which mode of disk formatting is selected (refer to chapter on Disk Formatting). The remaining header and data fields are written from the respective pattern registers.

FORMAT TRACK NO GAP (Write Header-Write Data)
Op Code = 10100100
There is a special format option which can be used in hard sectored drives by not setting the format bit in the Disk Command register and using a multi-sector operation. This enables writing of a format without intersector gaps. This may be useful if the drive puts servo information in the gaps. This would be overwritten with the normal format.

FIND HEADER/ID (Compare Header-Check Data)
Op Code = 01010010
This command is normally used to perform a diagnostic operation, operating almost like the Compare Header-Read Data command. It will first scan and compare the header information. The data field is read but not transferred.

FIND ID MULTI-SECTOR (Compare Header-Check Data)
Op Code = 01010100
This command is a multi-sector version of the Find Header, and can be used to verify all the headers on a track are valid.

RECOVER HEADER (Write Header-Check Data)
Op Code = 01100010
This command will start immediately, and rewrite the header field for the first sector encountered. This can be used as a method of recovering a damaged or unreadable ID field. To do so actually requires the interlock mode executing this command following a Find Header command. To recover the sector in error, the sector physically preceeding is found and the Find header is executed followed immediately by this command. The new header is then written over the damaged one.

RE-ENABLE (Re-enable bit set)
Op Code = 00000001
This command is used as part of the Reset/Re-Enable operation which is normally used to recover from an error condition. First the Reset bit in the Operation Command register is set, then reset. To finally enable the DDC for another command, the Re-Enable bit is set (see previous section).

NO OPERATION (Ignore Header-No Data Operation)
Op Code = 00000000
As mentioned previously, this is a NOP command. The DDC will not perform any operation.

ILLEGAL COMMANDS

Figure 5.14 lists several commands as illegal commands. They are not implemented by the internal PLA, and executing these commands will cause erroneous results.

CHAPTER 6 System and Disk Interfacing with DDC

6.0 INTRODUCTION

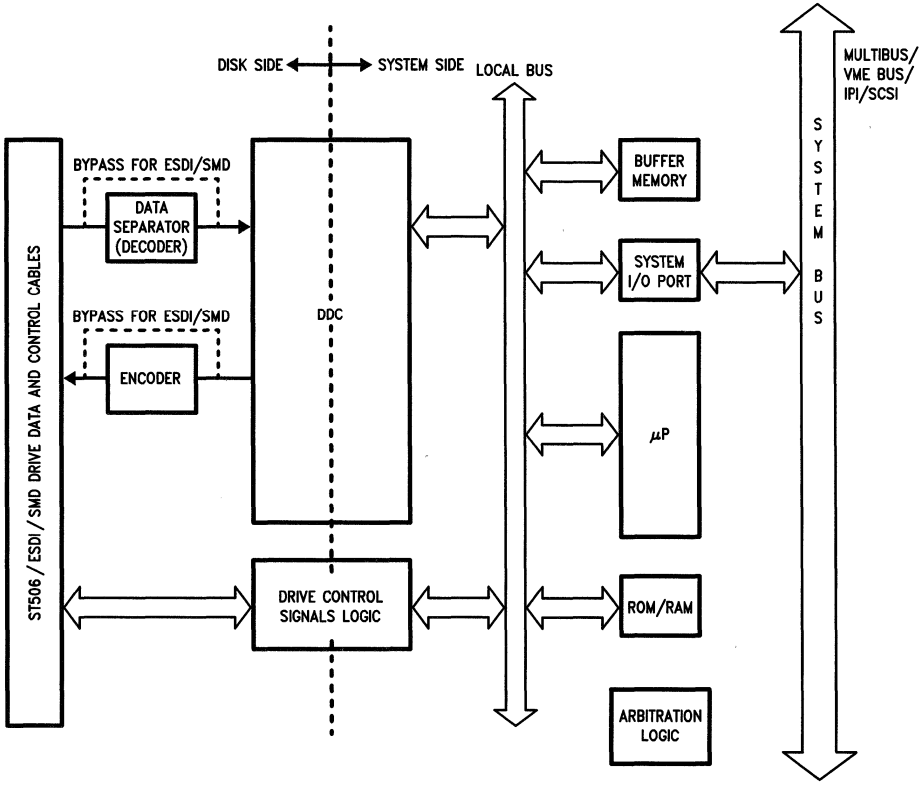
In a typical disk controller design, the DDC interfaces to the controlling microprocessor, memory and/or main system bus (such as MULTIBUS®, VME bus or IPI and SCSI) on the system side, and to the disk drives (via various disk interfaces such as ST506, ESDI and SMD) on the disk side. *Figure 6.1* shows the DDC in a typical disk controller design. As mentioned in previous chapters, the DDC (disk data controller) controls only the disk data path. It takes 8- or 16-bit wide data from the system bus, serializes it and then writes it to the disk in a disk write operation. While reading the disk data from the disk, the DDC deserializes the data and puts it back to the system bus. Before any disk operation can be performed, a track seek operation must be performed. The drive control signals required to perform a seek operation, can be generated using additional simple circuitry.

In this chapter, various DDC hardware interfaces to the host system and disk drive will be discussed in detail. DDC programming algorithms for carrying out disk operations are discussed in detail in chapter 7. The hardware interfacing is divided into two parts, system side and disk side.

6.1 SYSTEM SIDE INTERFACE

The DDC goes through three configuration modes when it performs a disk operation. When a microprocessor accesses the DDC to initialize it for a particular disk operation, it becomes a peripheral to the microprocessor (peripheral mode). While performing a disk operation and whenever the FIFO requires a data transfer to or from the external memory, the DDC becomes the bus master (master mode) and uses its on-chip DMA channels to perform the desired data transfers. If the desired data transfers are carried out by an external DMA controller, the DDC becomes a slave to the external DMA (slave mode controller).

In the following sub-sections the DDC's hardware interfacing with the microprocessor (peripheral mode), memory (master mode), and external DMA (slave mode) are discussed. The logic needed for arbitration of bus control between the DDC and microprocessor, and, to provide drive control signals are also discussed. In addition to the hardware connections, typical timing for various signals is also discussed.



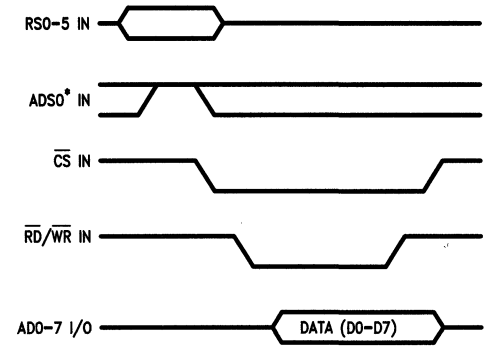
TL/F/8663–81

FIGURE 6.1. A Typical DDC-Based Disk Controller

6.1.1 Microprocessor-DDC Interface

The DDC must be initialized by a microprocessor (or microcontroller) in order to perform various disk operations. *Figure 6.2(a)* shows a basic microprocessor-DDC interface. When the microprocessor accesses the DDC, all 64 internal registers appear to it as unique memory or I/O locations. Each register can be randomly accessed and operated on. Only eight bits of data can be transferred to or from these registers using pins AD0–7. All the registers can be individually selected using six register select lines, RS0–5. Using these dedicated lines with an address strobe input, ADS0, the chip can be used in both multiplexed and demultiplexed address bus environments. Basically, the ADS0 and RS0–5 together act like a flow through latch.

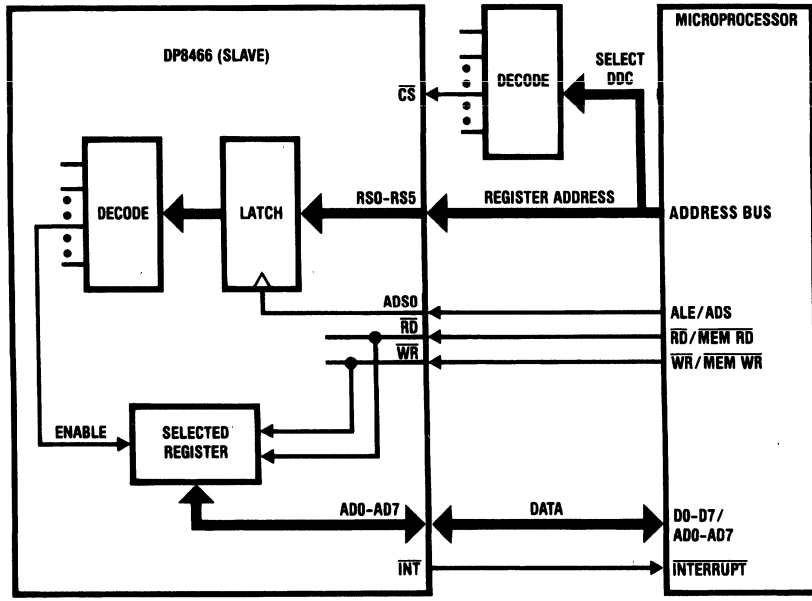
With multiplexed address and data lines, a positive strobe pulse on ADS0 will latch the address. The ADS0 line may be derived from a microprocessor address strobe line such as ALE. In systems with a dedicated address bus (demultiplexed), ADS0 may be pulled high to allow address information to flow through the latch. Finally, by applying $\overline{CS}$ and $\overline{RD}$ or $\overline{WR}$ strobes, the selected register is accessed. *Figure 6.2(b)* shows a typical timing for a multiplexed and demultiplexed system bus when the DDC is in Peripheral mode.



TL/F/8663–83

FIGURE 6.2(b). DDC Register Read/Write in Peripheral Mode

*Latched Register Select: ADS0 = Active
Non-Latched Register Select: ADS0 = Tied High



TL/F/8663–82

FIGURE 6.2(a). A Simplified Microprocessor Interface in Peripheral Mode

6.1.2 Bus Arbitration Logic

When the FIFO fills up (during a read operation) or empties (during a write operation) to the progammed threshold level, the DDC issues a local request (LRQ) to carry out local buffer memory transfers. Similarly, the DDC issues a remote request (RRQ) when local buffer memory needs a data transfer on the remote channel. Upon receiving the request (LRQ or RRQ), the current bus master should generate an acknowledge (LACK or RACK) to transfer bus control to the DDC. The request essentially puts the microprocessor (current bus master) on hold. The DDC keeps LRQ or RRQ asserted for the entire selected data burst transfer. This enables the DDC to remain bus master during this time while the microprocessor is on hold. See *Figure 6.3(a)* for typical RRQ-RACK and LRQ-LACK timing.

Generally, the LRQ-LACK or RRQ-RACK pins of the DDC are connected to the HOLD-HOLDA type (or BUSREQ-BUSACK type) of pins on the microprocessor through some additional combinational logic. This additional logic would be responsible for providing a smooth bus arbitration between the DDC and other possible bus users (like microprocessor). It must remove the possibility of any bus contention and may also prioritize DDC's DMA requests with other requests which may be present in the system. See *Figure 6.3(b)* for typical bus arbitration logic connections.

The Address strobes (ADS0 and ADS1) from the DDC and similar address strobe signals from the microprocessor may also be used in the bus arbitration logic, to generate a common set of address strobes for the system. This is useful if the demultiplexing address latches are to be shared between the DDC and the μP.

The DDC samples the RACK and LACK inputs during T_i(idle) or T_4 for each memory transfer. In general, the arbitration logic should leave LACK/RACK high for the duration of the burst. It may also toggle LACK/RACK so long as LACK/RACK are high during T_4 when the local channel has transferred one sector (in tracking mode-dual DMA), or when the remote DMA is enabled (in non-tracking mode).

6.1.3 The DDC-Memory Interface

After becoming bus master, the DDC communicates with the buffer or system memory if its on-chip DMA is chosen for data transfers. This communication takes place via a 16-bit multiplexed address/data bus, AD0–15, and supported by $\overline{RD}$, $\overline{WR}$, ADS0 and ADS1 strobes. The DDC is programmed in one of the three basic DMA modes; single channel, dual channel non-tracking and dual channel tracking. (See chapter 7 for detailed explanation on DMA programming). The hardware aspects of using the DDC in one of these modes are discussed below.

SINGLE CHANNEL DMA

In single channel DMA mode, the DDC takes care of data transfers between the FIFO and external memory using its local DMA channel. In this mode, the DDC can be set to generate either 16-bit or 32-bit of address. This gives system architects a great deal of flexibility. With 32-bit single channel DMA, up to 4 GBytes of memory can be accessed which is ideal in a system where the buffer memory is located within the main system memory. Factors like sector length, number of sectors per track and DMA burst length should be considered in determining the appropriate buffer memory size.

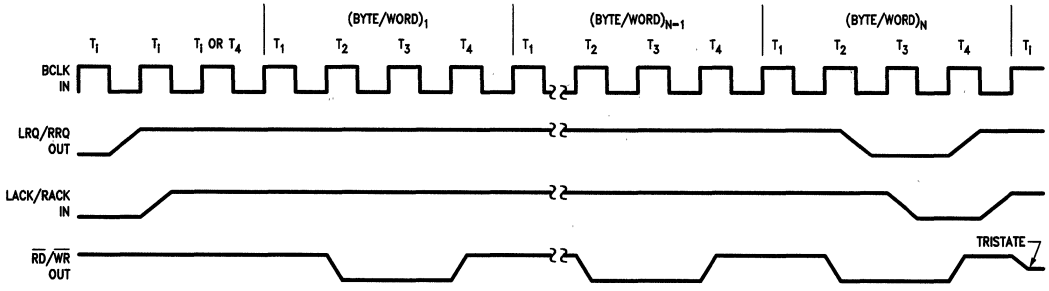


FIGURE 6.3(a). A Typical LRQ/RRQ-LACK/RACK Timing for a Data Burst Transfer

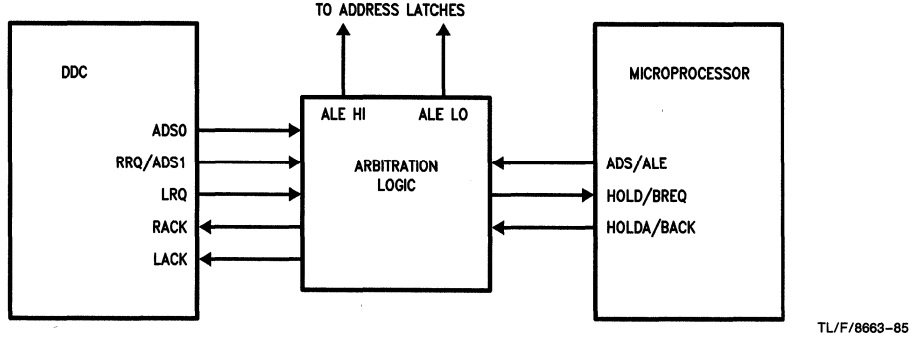


FIGURE 6.3(b). Arbitration Logic Connections

DUAL CHANNEL DMA

For systems where disk data is first buffered in a local memory before being transferred to the host memory via a host I/O port, the DDC offers two channels of DMA to handle the necessary data transfers. The "local" DMA channel controls data transfer between the FIFO and local buffer memory, while the "remote" DMA channel controls data transfer between the local buffer memory and the host I/O port. The two DMA channels can be operated independently of each other or the remote channel can be made to track the local channel. In either case, both channels can address a 64k address space. In the rare cases where a dual bus architecture and more than 64k bytes of local buffer memory are required, the DDC should be programmed for single channel, 32-bit addressing mode to handle FIFO-to-buffer transfers, leaving the buffer-to-host transfers to an external DMA controller.

For the purpose of the following discussion, it is helpful to introduce the concept of "source" and "destination" channels. In a disk read operation, the source channel is the local DMA channel, and the destination channel is the remote DMA channel. Conversely, in a disk write operation, the source channel is the remote DMA channel and the destination channel is the local DMA channel. In both cases, the source channel controls data transfers from the data source to buffer memory, while the destination channel controls data transfers from buffer memory to data destination. As an example, in a disk read operation, the source channel is the local DMA channel and controls data transfers from the FIFO to the buffer memory, the remote channel becomes the destination channel and controls transfers from buffer memory to the host.

Non-Tracking Dual Channel DMA Mode

In this mode, the local and remote channels operate as two independent DMA channels. This allows the microprocessor to tightly control data movement between the FIFO, buffer memory and the host I/O port. It also allows the microprocessor to process disk data in the buffer memory before transferring them to the destination, for example. This extra degree of control imposes an extra burden on the microprocessor. It is now responsible for preventing any destination channel transfers that may result in a data overrun situation. This can occur since the destination channel will continue to transfer data out of the buffer memory even though that section of memory has not been written to by the source channel.

Tracking Dual Channel DMA Mode

In this mode, the DDC keeps track of data transfers occurring over the two DMA channels. It forces the two DMA channels to track each other appropriately to prevent data overruns.

The tracking mechanism is implemented through a DMA sector counter, or DSC, which keeps track of the difference between the number of sectors transferred via the source and destination channels. The DSC is incremented every time a sector of data has been transferred into the buffer memory via the source channel, and decremented each time a sector of data has been transferred out of the buffer memory via the destination channel. The destination channel will not initiate the transfer of a new sector of data unless the DSC is non-zero. Thus the destination channel will not initiate the transfer of a new sector until it has been completely transferred into memory by the source channel. With appropriate choice of DMA data burst lengths, the local buffer memory then appears to the DDC as an extension of the internal FIFO.

DATA TRANSFER TIMING

Memory Read/Write Cycles

A standard DMA memory cycle consists of 4 BCLK periods, with the exception of the extended DMA memory cycle associated with the 32-bit addressing single channel DMA mode.

Referring to *Figure 6.4(a)*, a standard memory cycle consists of four bus states, T_1 to T_4, each lasting for one BCLK period. The cycle begins with the rising edge of BCLK in T_1, at which time the 16-bit memory address is put out on the address/data bus. ADS0 is also asserted during T_1 and can be used by the memory to strobe in the address on its negative edge. The low order address bits (A0–A7) are put out on port AD0–7, and the high order address bits (A8–A15) on port AD8–15. The address remains on these ports for the remainder of T_1. At the start of T_2 the address is removed and, depending on whether a read or write access is required, either the Read Strobe ($\overline{RD}$) or the Write Strobe ($\overline{WR}$) is asserted. These strobes stay asserted until the end of T_3. If a write access to buffer memory is required, the DDC will put its FIFO data on the address/data ports at the beginning of T_2 until the end of T_3. If a read access to buffer memory is required, then the DDC expects valid memory data to be placed on these ports during T_2 to T_3 so that the required data setup time referenced to the positive edge of $\overline{RD}$ is met. At the beginning of T_4, the appropriate acknowledge input (LACK/RACK) is sampled. If the sampling occurred when the acknowledge input is high, then the next memory cycle will be permitted to start at the end of T_4. Otherwise the DDC will relinquish control of the bus. If the acknowledge input is de-asserted prior to the rising edge of BCLK in T_4, the current memory cycle will be completed before the DDC frees up the bus.

The extended DMA memory cycle is used only in the 32-bit addressing single channel DMA mode. Since there are only 16 address/data lines available, the 32-bit addresses must be split into two groups of 16 bits and multiplexed onto the address/data ports. The low order 16-bit address is handled exactly as for the 16-bit addressing DMA modes. Additionally, a separate address strobe (ADS1) is provided so that the high order address bits can be stored in an external address latch.

As shown in *Figure 6.4(b)*, the extended DMA memory cycle is essentially the standard DMA memory cycle with an additional state T_0 inserted prior to T_1. At the start of T_0, the high order address bits are placed on the address/data ports for the duration of T_0. The address strobe ADS1 is also asserted and can be used by the memory system to latch in the high order address on its negative edge. Events occurring from T_1 to T_4 are exactly the same as in the case of the 16-bit addressing DMA modes.

Once initialized at the beginning of a DMA block transfer, the external high order address latch only needs to be updated whenever the lower order 16-bit address rolls over. Thus the extended DMA memory cycle is only required at the beginning of a block transfer, and each time the lower order address rolls over. Consequently, even when 32-bit addressing is required, the vast majority of DMA memory cycles will be the standard 4-clock cycle.

Prior to commencing any DMA transfers, the DDC must request and be granted control of the address/data bus. Whenever a DMA channel (local or remote) is ready for a transfer, the corresponding request output (LRQ or RRQ) is asserted to request control of the address/data bus. When such control is granted by the system via the appropriate acknowledge input (LACK or RACK), the DMA cycles will

commence. *Figures 6.5(a)* and *6.5(b)* illustrate the signal timings for a burst DMA transfer over the local and remote DMA channels.

Referring to *Figure 6.5(a),* a local transfer is requested by the DDC by asserting LRQ high when the FIFO threshold is reached. The DDC samples the LACK input at each positive edge of BCLK thereafter until a logic high is detected on the LACK input. At which point the DDC assumes control of the bus and starts a burst transfer. The burst spans over N memory cycles (where N is equal to the FIFO threshold in byte mode or half the FIFO threshold in word mode), or until the FIFO is full or empty. In the last transfer cycle of a burst, LRQ is de-asserted at the start of T_2 and the DDC relinquishes control of the bus at the end of T_4.

The remote transfer operation can be programmed to transfer data in bursts or to transfer the entire block in one stream. If burst mode is selected, the remote channel will transfer data in bursts with the user-programmed number of bytes per burst as illustrated in *Figure 6.5(b).* As in the case of local transfers, the DDC first requests bus control by asserting remote request (RRQ) high. The remote acknowledge input (RACK) is then sampled at each positive edge of BCLK until it (RACK) goes high. The DDC then starts the first of a series of N transfer cycles, where N is the programmed number of bytes or words per burst. In the last cycle of the burst, RRQ is de-asserted at the beginning of T_2 and the DDC gives up control of the bus at the end of T_4 if the entire block has been transferred. Otherwise, RRQ is re-asserted at the start of T_4 to request bus control for the next burst. This allows other peripherals with equal or higher priority to gain access to the bus between bursts. The bus arbitration logic must de-assert RACK prior to T_4 to ensure that the DDC will not initiate another memory cycle. It should also hold off RACK to the DDC until such peripherals have completed their transfers.

If the remote channel is programmed to transfer an entire block in a continuous stream, then RRQ will not be de-asserted until T_2 of the last transfer cycle. If the system cannot tolerate such prolonged bus usage by the DDC, then the bus arbitration logic may de-assert RACK to force the DDC to relinquish bus control. If RACK is de-asserted prior to T_4, the DDC will complete the current transfer cycle and relinquish bus control. Otherwise the following transfer cycle will be completed before the DDC will free up the bus. When this technique is used to gain access to the bus, the arbitration logic must delay the granting of bus access to another device for time T_d from the de-assertion of RACK, where T_d is given by:

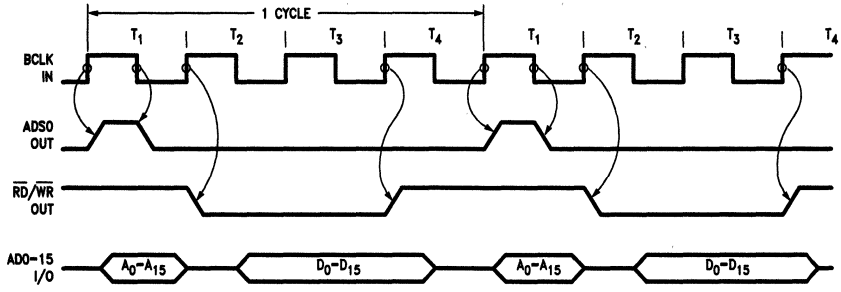
$$T_d = (\text{Period of BCLK}) + (\text{Duration of one DMA memory cycle}).$$

Note that the duration of a DMA memory cycle may vary depending on the number of wait states that may be inserted, as explained in a later section.

In both tracking and non-tracking modes, the local channel has priority over the remote channel for bus access.

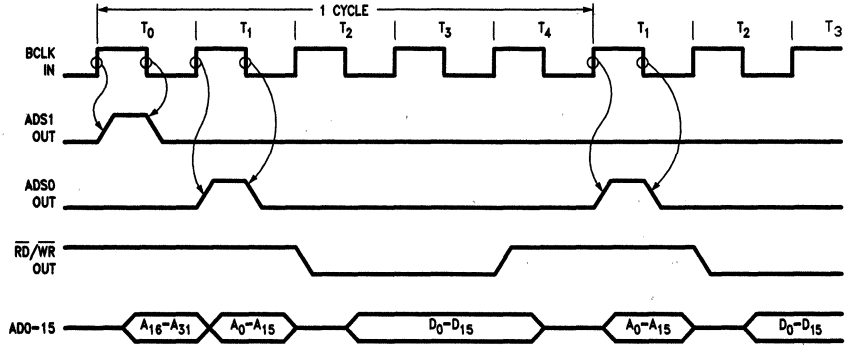
Bus Latency

The DDC can be operated at a 20 MHz Bus Clock (BCLK). With this frequency data can be transferred between the FIFO, local memory and system I/O port at the rate of 5 Mega-Transfers/sec (or 10 MBytes/sec) assuming a 4 clock periods memory cycle. With 15 Mbits/sec disk data rate, the 32-byte FIFO can be filled in approximately 17 μs.



TL/F/8663–86

FIGURE 6.4(a). 4-Clock DMA Memory Cycle (DDC in 16-Bit Single and Dual Channel DMA Modes)



TL/F/8663–87

FIGURE 6.4(b). 5-Clock Extended DMA Memory Cycle (DDC in 32-Bit Single Channel DMA Mode)

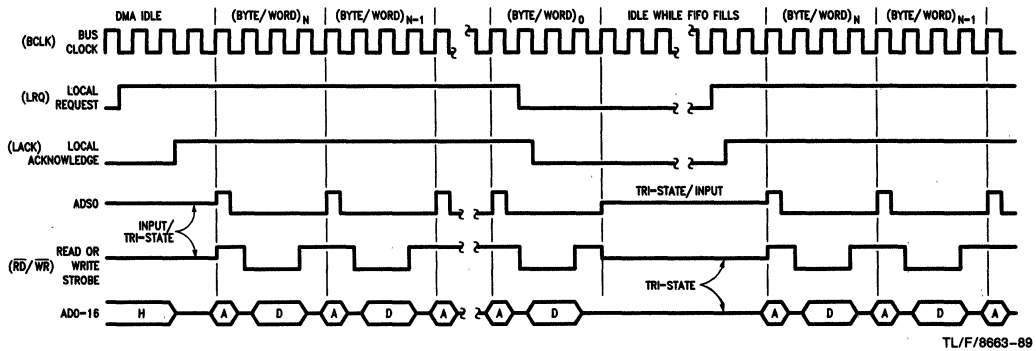


FIGURE 6.5 (a). Typical Local DMA Burst Transfer Timing
(H = Host, A = DDC Address, D = DDC Data)

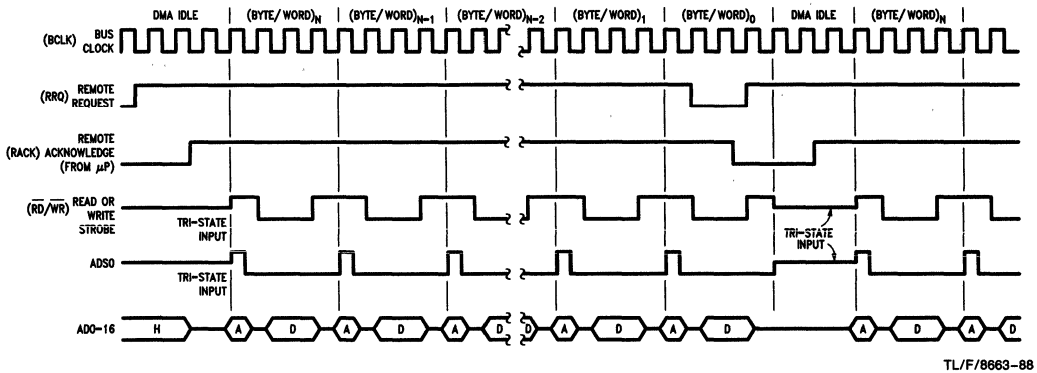


FIGURE 6.5 (b). Typical Remote DMA Burst Transfer Timing
(H = Host, A = DDC Address, D = DDC Data)

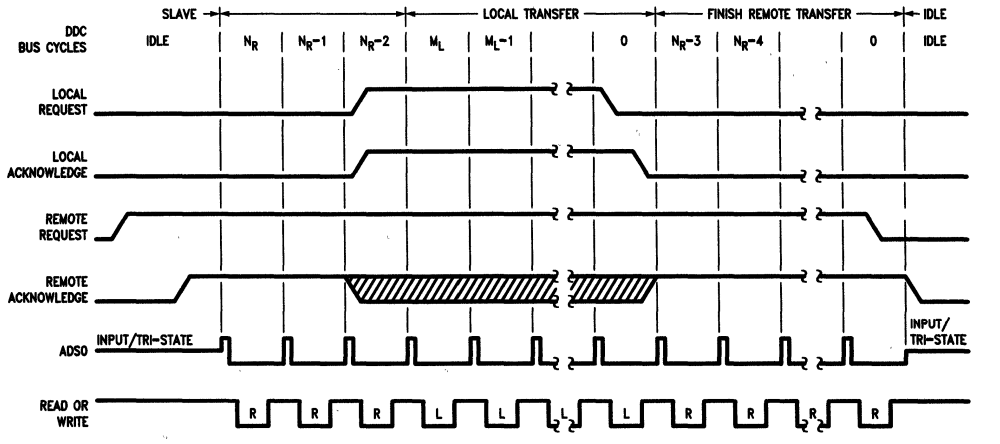


FIGURE 6.5 (c). Interleaved Remote and Local Transfer Timing

Note N_R = Remote DMA Transfer Cycle Number N
M_L = Local DMA Transfer Cycle Number M

At 20 MHz bus clock (i.e. clock period of 50 ns and memory cycle of 200 ns) and assuming word-wide DMA transfers, the DDC will take 16 × 200 ns = 3.2 μs to empty the FIFO. This essentially shows that microprocessor bus is free for 17 − 3.2 = 13.8 μs (81.2% of the time). In other words, the DDC will need bus control after every 17 μs.

The selection of the FIFO threshold level should be based on the maximum amount of time the system takes to respond to a DMA request, often called bus latency. If the system has a longer latency, then a smaller threshold should be programmed. This will allow greater time for the system to respond without overflowing the FIFO. The disadvantage to programming lower FIFO thresholds is that more requests are made, tying up the system bus more often. For example, with an 8 byte threshold a request would be made about every 4.5 μs.

A second consideration is whether to have an exact burst, or burst until the FIFO empties. If the system has significant latencies then the FIFO should be emptied. If the system must not relinquish the bus for too long then a fixed burst size must be chosen.

The bus latency should be calculated for a given disk and DMA transfer rate to determine appropriate FIFO threshold. Programming the DMA for various burst transfer options is discussed in depth in chapter 7.

Local and Remote Interleave Timing

In dual channel DMA mode, the local and remote transfers can be interleaved using the DDC in tracking mode (see chapter 7 for details). A typical local and remote data transfer interleave timing is shown in *Figure 6.5(c)*. In this case a remote transfer is interrupted by the higher priority local transfer. Taking the same example given in the previous sub-section, the remote transfers can be carried out during the 81.2% of time when the local channel is not using the bus.

Slow Read/Write

The Read or Write strobes ($\overline{RD}$ or $\overline{WR}$) can be extended by adding extra wait state(s). It can be done internally by setting the LSRW bit in Local Transfer register (and) or the RSRW bit in Remote Transfer register. This will generate an extra cycle (T_W) between the T_2 and T_3 clock cycles as shown in *Figure 6.6(a)*. These strobes can also be extended by setting the EEW bit in the Remote Transfer register in conjunction with driving the EXT STAT input high. This will add extra wait states between clock periods T_3 and T_4, as shown in *Figure 6.6(b)*. The DDC samples the EXT STAT input at the positive edge of T_3 during each DMA (local or remote) memory cycle. If EXT STAT is sensed high and the EEW bit of the Remote Transfer Register is set, then a wait state (T_W) of a bus BCLK period will be inserted after T_3. During such wait states, the DDC continues to sample EXT STAT at the positive edges of BCLK. If EXT STAT is sampled high, a new wait state will be inserted at the end of the current one. If EXT STAT is sampled low, then the next state will be T_4.

6.1.4 DDC-External DMA Interface

If the on-chip DMA is not to be used, an external DMA controller must be used to service the FIFO. A typical DDC-external DMA interface is shown in *Figure 6.7(a)*. The LRQ asserted by the DDC is acknowledged by the external DMA and the DDC becomes a slave to the DMA controller. The DMA controller carries out the local transfers and deasserts LACK after LRQ is deasserted by the DDC. A typical FIFO Read/Write sequence by the external DMA is shown in *Figure 6.7(b)*.

6.1.5 Drive Control Signals Logic

The drive control signals generation is not incorporated on the DDC which makes it interfaceable with any type of disk

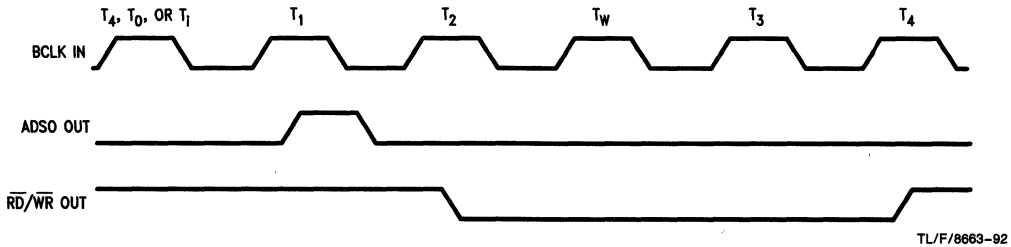


FIGURE 6.6 (a). DMA with Internal Wait States

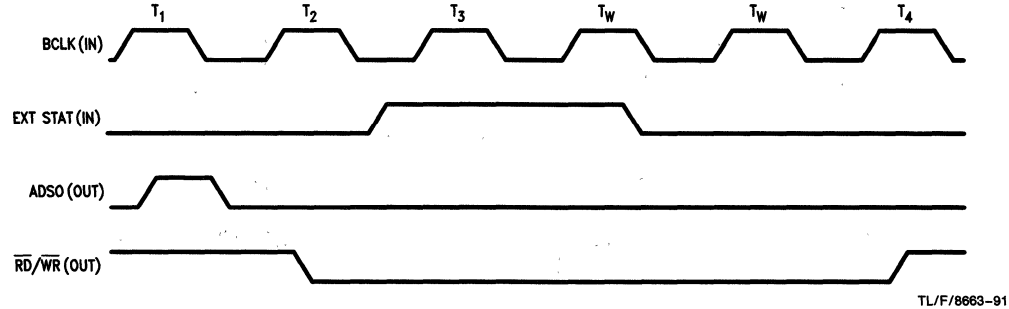


FIGURE 6.6 (b). DMA with External Wait States

interface. These signals can be generated simply with an I/O port associated with the controlling microprocessor. The microprocessor prepares the control signals for a particular drive interface and writes it to the I/O port. In high performance multi-drive, multi-interface systems, a dedicated microcontroller (COPS™, 8048 or HPC) or microprocessor (NSC800™, Series 32000® etc.) may be used to provide control signals for various drives. In this case, the controlling microprocessor (or host) instructs the microcontroller to perform a complete disk seek operation.

6.1.6 DDC in Typical System Configurations

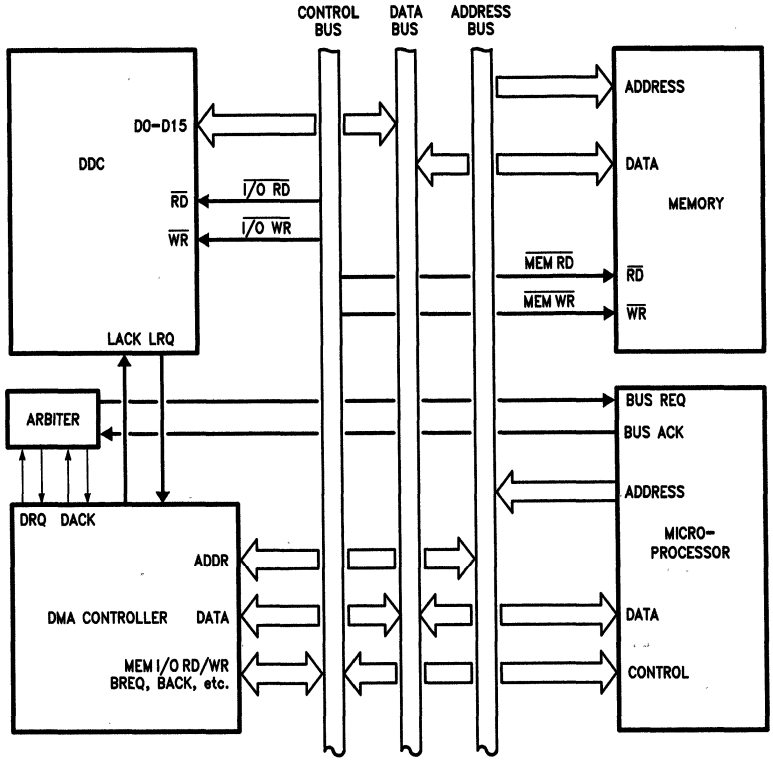
The DDC can typically be used in two types of system architectures, i.e. a single bus system or a dual bus system.

SINGLE BUS SYSTEM

Single bus systems usually are standalone systems controlled by a microprocessor. In such system, each component of the system communicates with the rest of the system through a single bus. For example, in a single board microcomputer, all the functional components like memory, I/O ports etc. communicate with the controlling microprocessor via main system bus.

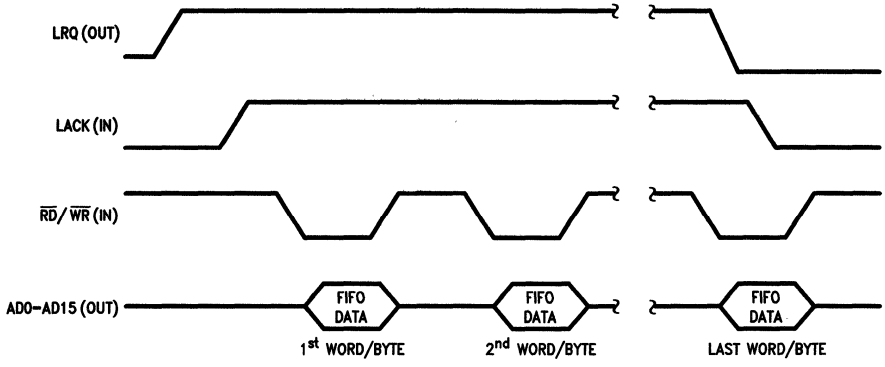
The DDC when used in such a system, becomes another component that communicates with the controlling microprocessor via the main system bus. It also communicates directly with the system memory using its on-chip DMA capability (usually the local channel). A single bus system generally requires a large system memory, so using the DDC in its single channel DMA mode becomes very worthwhile as it can access up to 4 GBytes of memory in this mode. The DDC in a typical signal bus system is shown in *Figure 6.8(a)*. The controlling microprocessor sets up the drive signal control logic to perform track seek operations and the DDC to perform disk operations.

In high performance single bus systems, it may be desirable to use a dedicated microcontroller to control the DDC and to generate the drive control signals instead of involving the main system microprocessor in such tasks. *Figure 6.9* shows the disk controller design in a 32-bit single bus system. Here an HPC or other single chip microcontroller is used to program the DDC for various disk operations and to generate drive control signals compatible with the desired interface. As mentioned above, the DDC is typically configured in single channel DMA mode when used in a single bus system environment. There is no restriction on using the DDC in dual channel DMA mode except total address capability. *Figure 6.10* shows a single bus system with the DDC in dual channel DMA mode, transferring data between the FIFO and 64k-pages of 16 MBytes system memory. External latches were added to extend the address range to 24 bits. To use them the host μP would load the most significant 8 bits with the 64k page to transfer data to/from. In this design the local and remote DMA channels are restricted to operating in the same page. To overcome this an additional latch and some logic gating the LACK and RACK signals could be used to enable operation of each channel in different pages.



TL/F/8663-93

FIGURE 6.7 (a). A Typical External DMA-DDC Interface



TL/F/8663–94

FIGURE 6.7 (b). DDC FIFO Read/Write by External DMA

DUAL BUS SYSTEM

Systems usually have two or more buses associated to them. One set of buses for local communication and another main system bus for communication with the host. These systems contain sub-systems (or modules) of the main system (the host). The purpose of the local bus architecture is to allow peripheral processors to control specific tasks, offloading the host CPU. Thus the local bus has in addition to peripherals, such as the DDC, a local microprocessor, memory and an interface to the main system. All the sub-systems access the main system through the I/O channels connected to the main system bus (such as MULTIBUS®, VME etc.).

The DDC easily fits into a dual bus system when configured in the dual channel DMA mode. The local DMA channel communicates with local memory and the remote DMA channel is used to transfer data between local memory and a system I/O port. The main system's DMA controller then takes the data to/from the I/O port from/to main memory. The DDC in a typical dual bus system is shown in *Figure 6.8(b)*. The local microprocessor receives commands from the host using DDC's remote channel and then sets up the drive control logic and the DDC for different disk operations. The arbitration block arbitrates the bus between the DDC (while in bus master mode) and the local microprocessor. *Figure 6.11* shows a dual bus system with local microprocessor using 48 kbytes of local memory space as ROM and RAM, and allowing only 16 kbytes of local buffer.

Figure 6.11 shows a possible local bus implementation. Here a local CPU such as the NSC800, 32008, 80188, 64180, etc. controls both the DDC and the disk control port. The local CPU is also in charge of receiving commands from the main system, caching disk sectors, and performing any logic to physical sector address translations.

In this design we have optionally segmented a special 16k for sector data buffers. This is not necessary, but does ensure the DDC will not access the local CPU's data memory inadvertently (this could be done in software as well). The HC646 and some read/write logic form the pass thru port for the commands/data from the main system. The DMA external controller, which may reside on the CPU (80188, 64180) or in the main system, controls transfers from main memory to/from the port. The DDC's remote DMA controls data transfers between local memory and the pass thru port.

INTELLIGENT PERIPHERAL BUS

A special case of the Dual Bus architecture are intelligent peripheral buses, such as SCSI and IPI. In this case the local bus is the intelligent disk drives bus, and the second bus is the SCSI or IPI bus which will connect to a main system through a host adapter. The dual channel DMA mode of the DDC is ideally suited to this application. The local DMA channel can manage transfers between the DDC's FIFO and the drives buffer RAM, while the remote DMA is passing information to/from the SCSI or IPI bus. The design of the system interface is very similar to *Figure 6.11*, except that the read/write interface logic block and the 'HC646 is replaced by a SCSI or IPI interface, and the systems remote DMA controller is located at the system end of the bus and not on the drive. With the DDC's two channel capability eliminates the need for an external DMA controller on the SCSI or IPI drive.

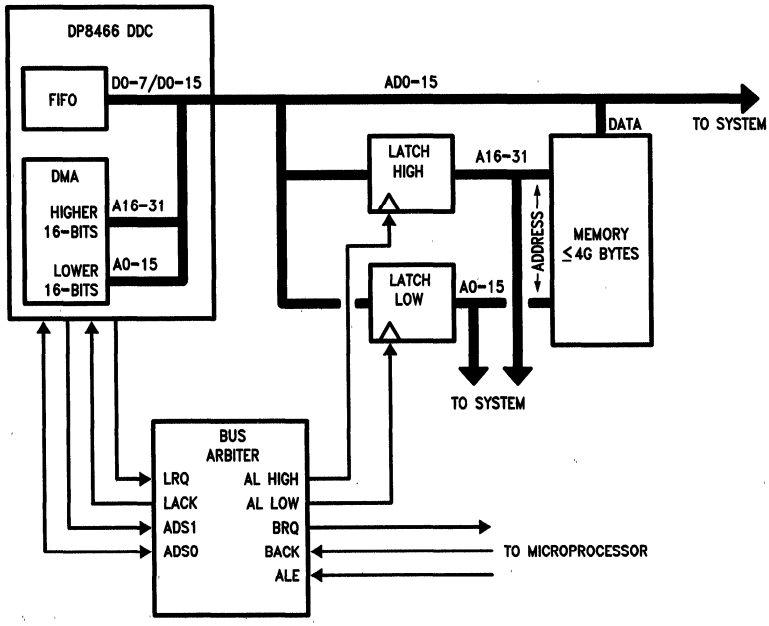


FIGURE 6.8(a). The DDC in a Single Bus System

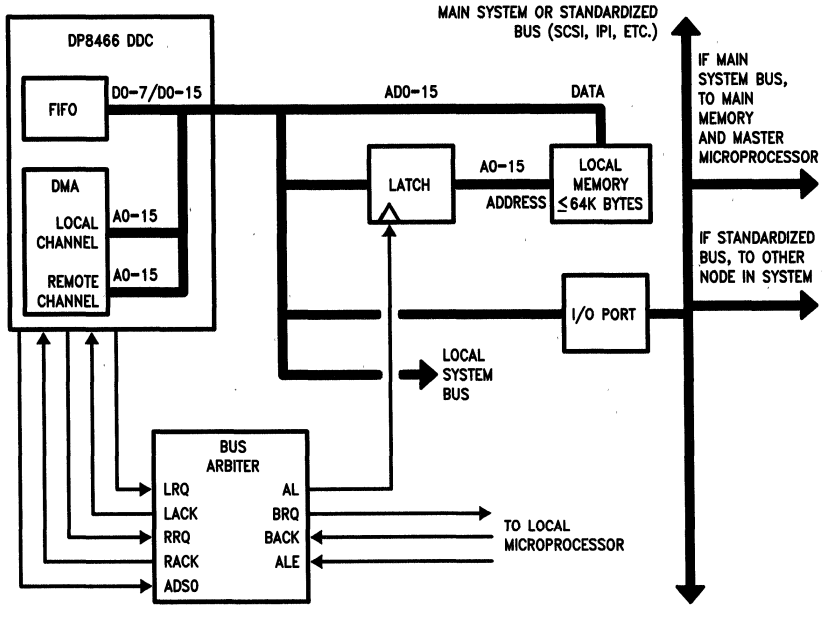
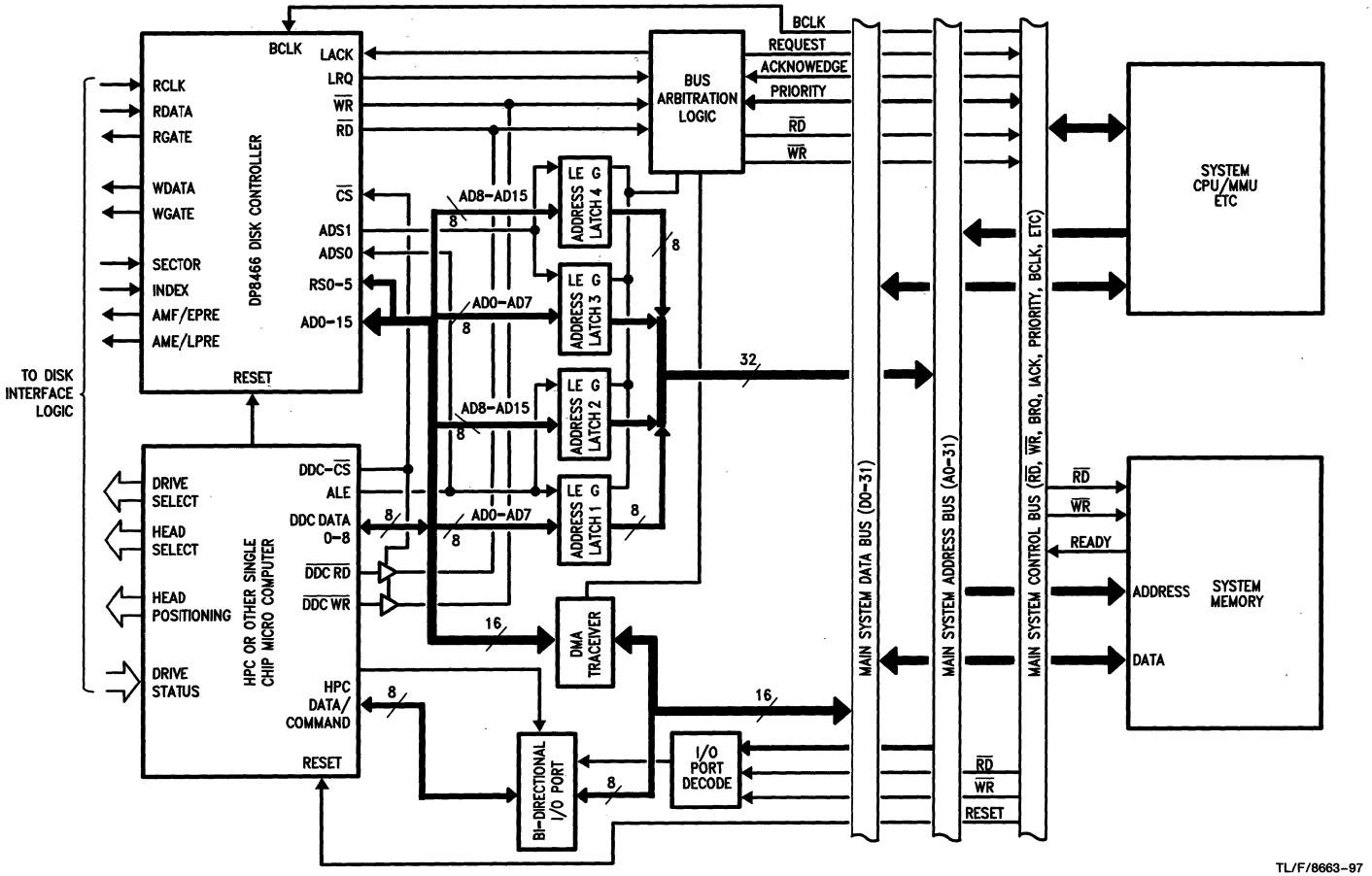
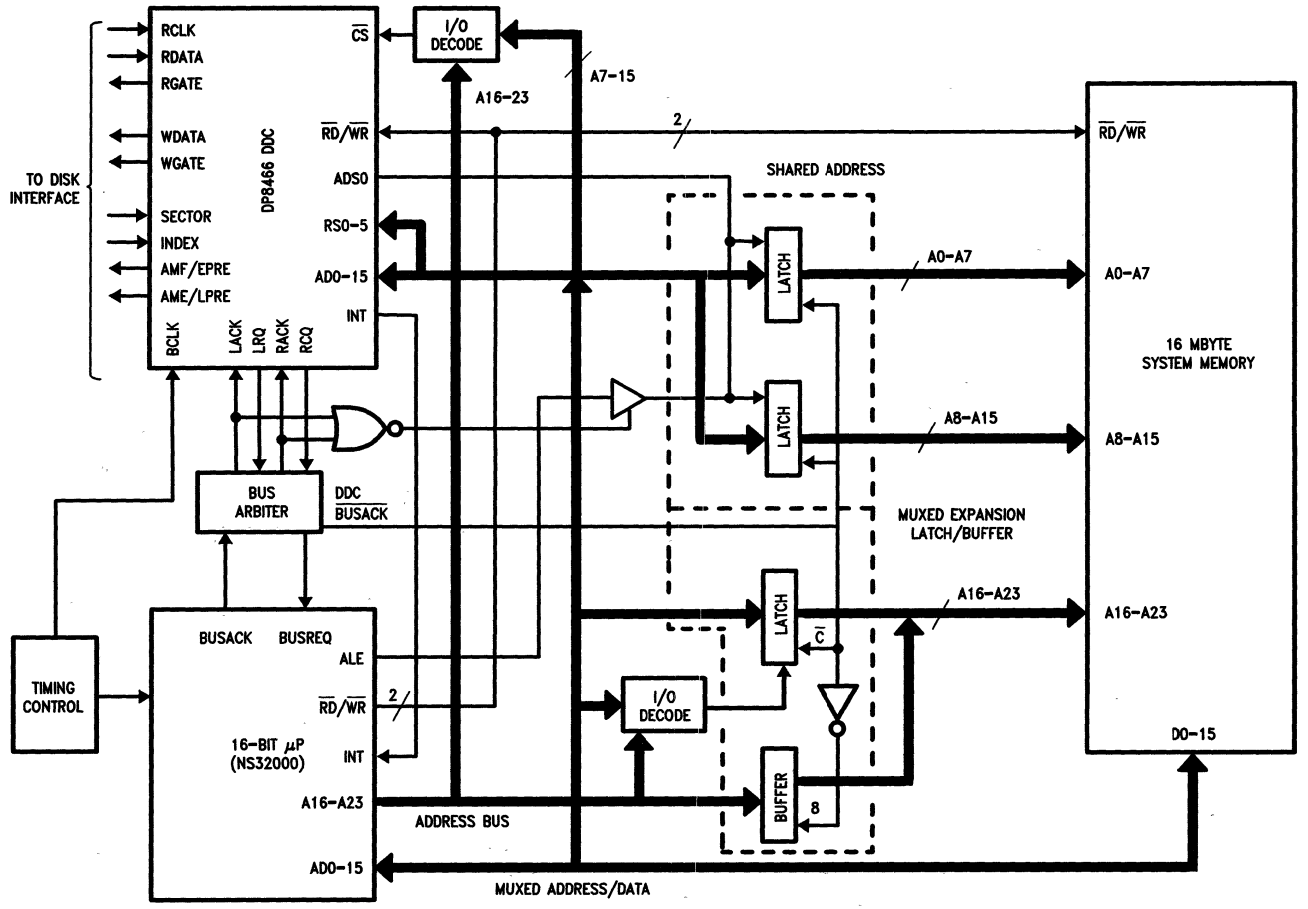


FIGURE 6.8(b). The DDC in a Dual Bus System



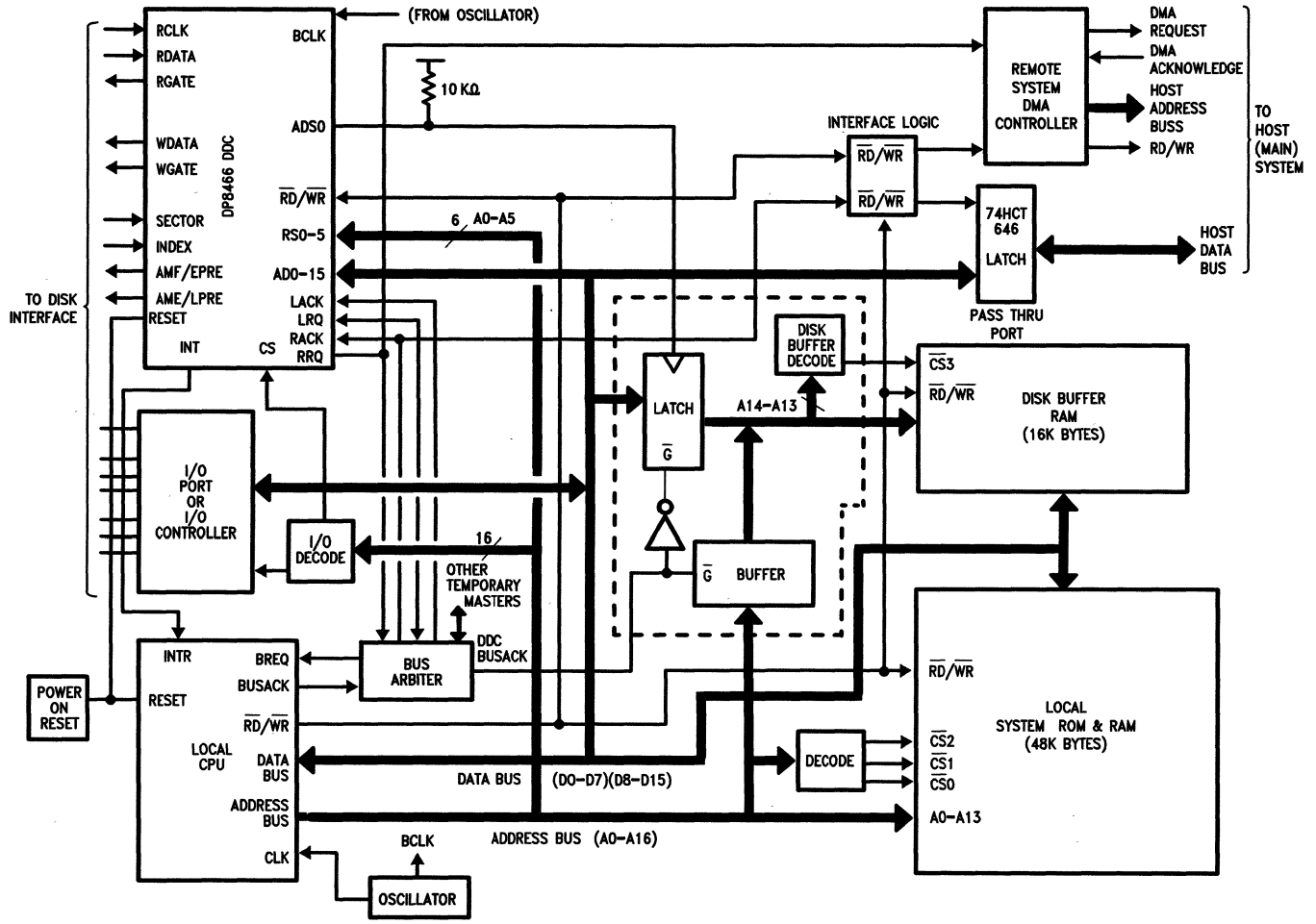
TL/F/8663–97

FIGURE 6.9. Conceptual Design of DP8466 Using Single Channel DMA and Controlled by a Microcontroller That Controls DDC and Drive Control Signals



TL/F/8663-98

FIGURE 6.10. Simplified design showing DP8466 with expanded dual direct memory addressing and 16-Bit multiplexed address/data CPU. Lower address latches are shared while the upper address lines are multiplexed.



TL/F/8663–99

FIGURE 6.11. Simplified design of DP8460 with a local 8-bit μP. Latches or microcontroller for disk control signals and limited disk buffer is included.

6.2 DISK SIDE INTERFACE

The disk side hardware connections/orientation is a function of the disk interface standard used, the encoding/decoding scheme and the local intelligence used for disk control signals. The Disk System Controller essentially consists of a data separator (DP8465), the Disk Data Controller (DP8466) and some local microprocessor for disk control like the NSC800 or a microcontroller like the HPC. The Disk Pulse Detector (DP8464) is situated on the drive for all the interface standards, while the data separator is on the drive for the ESDI and SMD interfaces. *Figure 1.9(a)* in chapter 1 also shows the interface points for the various standards. If MFM encoding is used it can be programmed to be part of the DP8466, details can be found in chapter 7; whereas in case of 2,7 RLL code, the DP8463 (2,7 ENDEC) could be used in conjunction with the DP8462 (2,7 data synchronizer).

6.2.1 Generalized Disk Interface with DP846X Chip Set

The DP8464 pulse detector receives signals from the disk's read amplifier, and converts these signals to a digital pulse train. The DP8465 Data Separator receives digital pulses from a pulse detector circuit (such as the DP8464). After locking on to the frequency of these input pulses, the DP8465 separates them into synchronized data and clock signals. If the input pulses are MFM encoded data, the data is made available as decoded NRZ data to be deserialized directly by the disk data controller DP8466. If the RLL code is used, the synchronized data output is available to allow external circuitry to perform the data decoding function. All the digital input and output signals are TTL compatible. The raw MFM from the pulse detector in the drive is connected to the ENCODED DATA input of the DP8465. The DELAY DISABLE input determines whether attempting lock-on will begin immediately after READ GATE is set or after two bytes. Typically in a hard sectored drive, READ GATE is set active as the sector pulse appears, meaning a new sector is about to pass under the head. Normally the preamble pattern does not begin immediately, because gap bytes from the preceding sector usually extend just beyond the sector pulse. Allowing two bytes to pass after the sector pulse helps ensure that the PLL will begin locking on to the preamble, and will not be chasing non-symmetrical gap bits. Attempting to lock-on to a fixed preamble pattern speeds up lock-on, and after another two bytes the PLL will nominally have locked-on. Thus DELAY DISABLE should be set low for this kind of disk drive. For soft sectored drives, the controller normally will not wait for the index pulse before it attempts lock-on, so that READ GATE may go active at any time. Chances are the head will not be over a preamble field and therefore there is no need to wait two bytes before attempting lock-on. DELAY DISABLE can therefore be set high. If a non preamble field is passing by as READ GATE goes active. The DP8465 will not indicate lock, and so no data decoding will occur nor will MISSING CLOCK DETECTED go active. Normally, if lock-on has not been achieved after a certain time limit, the controller should de-activate READ GATE and then try again. For MFM encoded disk drives, the LOCK DETECTED output will be connected back to the SET PLL LOCK input. As the PLL achieves lock-on, the DP8465 will automatically switch to the slower tracking rate and decoded data will appear at the NRZ READ DATA output. Also the READ CLOCK output will switch from half the 2f clock frequency to the disk data rate frequency. If a delay is required before the changeover occurs, a time delay may be inserted between the two pins. The ZEROES/ONES PREAMBLE input selects which preamble the chip is to lock-on to. *Figure 6.12* gives schematics of the data separator in a disk system—when in the controller and when on the drive. For more specific details on the DP8465 refer chapter 3.

If the drive uses the RRL code such as "2,7", instead of MFM, the PLL function of the DP4862 may be used in conjunction with the 2,7 ENDEC (DP8463), as shown in *Figure 6.13*. The DP8463 performs encoding of NRZ data to RLL encoded data, and RLL encoded data back to NRZ data. It uses the SYNCHRONIZED DATA output of the DP8462 along with VCO CLOCK to lock-on to the preamble and then decode data. For more specific details on the DP8463 refer to it's data sheet.

The most important component in the disk side is the physical disk interface itself. We shall discuss the interface details for some major interface standards viz., STxxx, ESDI etc. Higher level interfaces, like SCSI, incorporate the whole controller board on the drive and interface with the host through a host adapter on to the system bus. However the DDC as such interfaces to the host microprocessor for commands and status.

6.2.2 Interfacing the DDC to the ST506/ST412HP Standard

The schematic in *Figure 6.14* shows the interfacing of the DDC to a ST506/ST412HP disk interface. The ST506/ST412HP interface standard has a control cable which is daisy chained to the drives (assuming a multiple drive system). On selecting the drive all the signals on the control cable are associated with the drive selected. The data cable is radial in nature and is multiplexed at the controller using the drive select output on the data cable. The ST506/ST412HP interface standard supports MFM encoding and soft-sectored formatting only. Since the DDC does not take care of the disk control signals, this is done using some local intelligence, indicated as the Disk Signals Controller block, DSC. The drivers are open collecter drivers as per the requirements of the interface standard. The DDC requires NRZ data, hence the MFM encoded data from the drive is sent to the data separator, which synchronizes the data and decodes it to NRZ. The missing clock detect output of the data separator is used to trigger the address mark found (AMF) input of the DDC. While writing data to the disk, the DDC provides MFM encoded data, and the necessary precompensation is provided using an external delay line in conjunction with the early and late precompensation outputs (EPRE/LPRE) of the DDC. The DDC and the DSC could interface to the system bus of the host system or could interface to the host system through an intelligent interface like SCSI or IPI, as mentioned in the previous section.

The basic operation of the interface is as follows. The DSC first selects the drive select lines. Then the head selected by the head select lines is positioned over the desired track by issuing step pulses in conjunction with the direction line. Once the head is positioned, indicated by the seek complete line, the DDC is ready to initiate a read/write operation. This interface's read path is through the data separator and hence a lot depends on the selection of the data separator. The data separator then feeds the controller. The write path consists of write data out of the DDC going to a precompensation block. The output of the precompensation block goes over the ST506 interface to the drive.

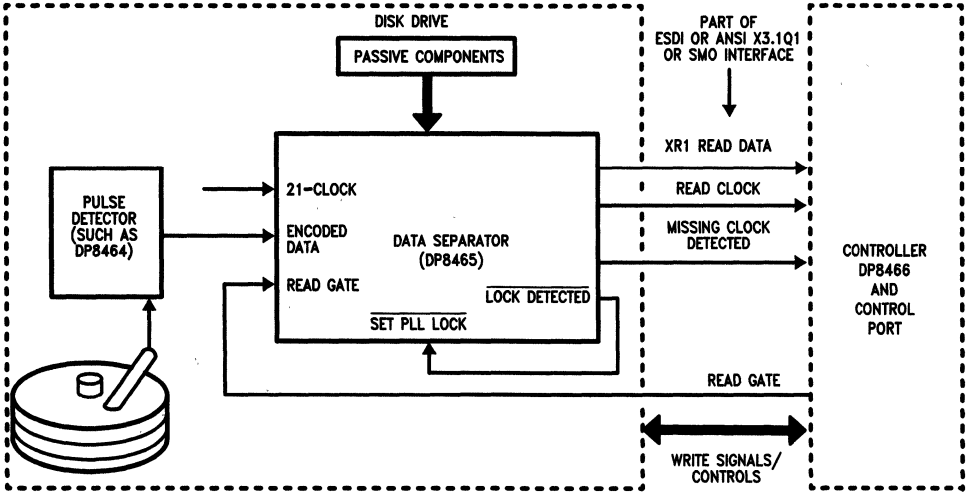


FIGURE 6.12 (a). Data Separator Residing in the Controller

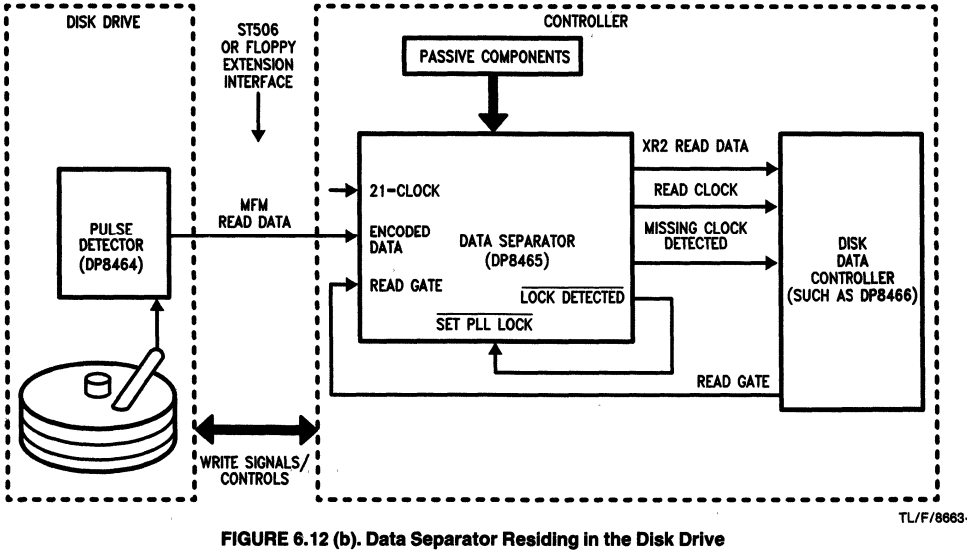


FIGURE 6.12 (b). Data Separator Residing in the Disk Drive

Interfacing the DP8463 2,7 ENDEC in a Disk System

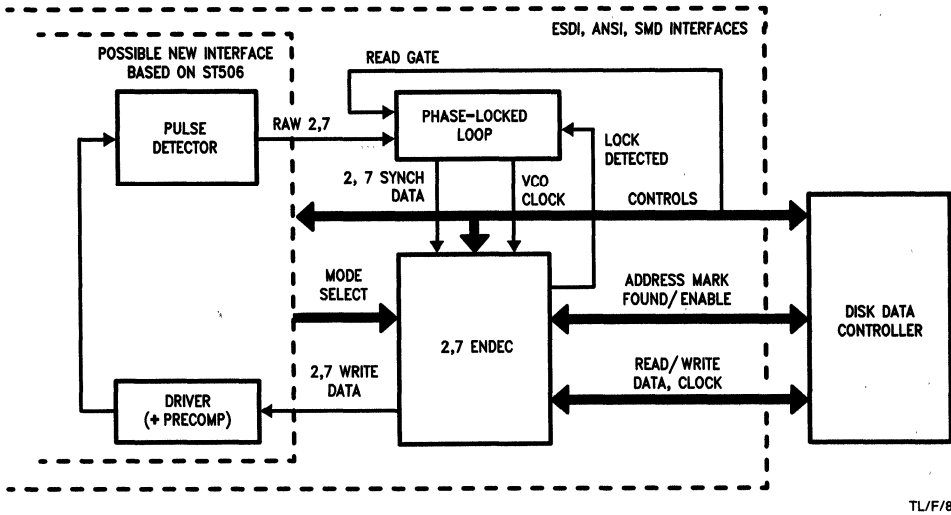


FIGURE 6.13. DDC and DP8463-2,7 ENDEC (a) Generalized Disk System Block Diagram

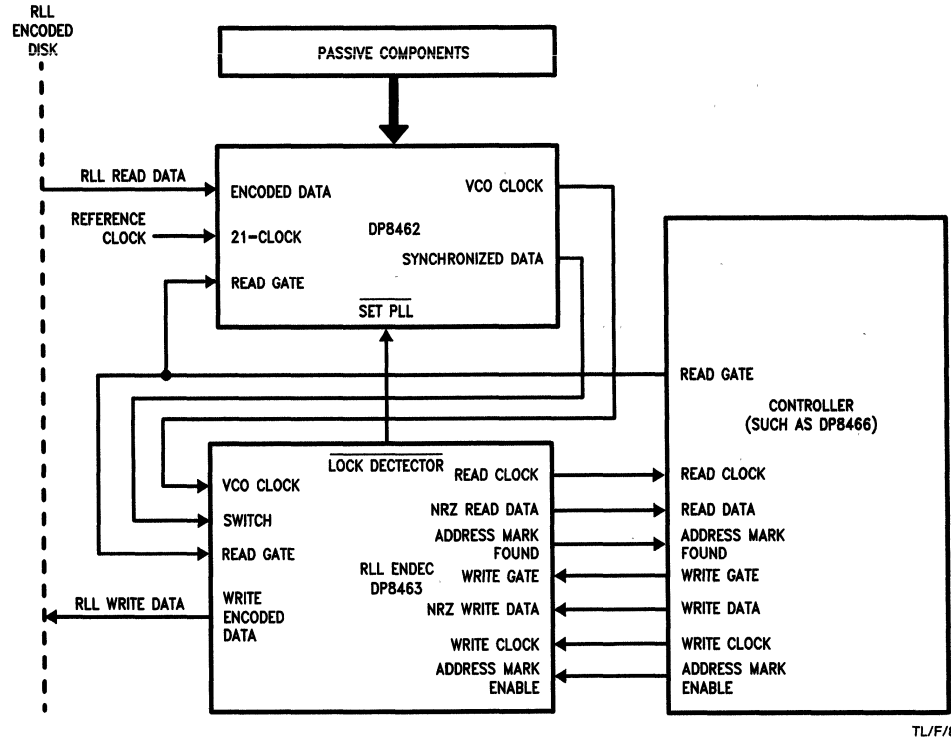
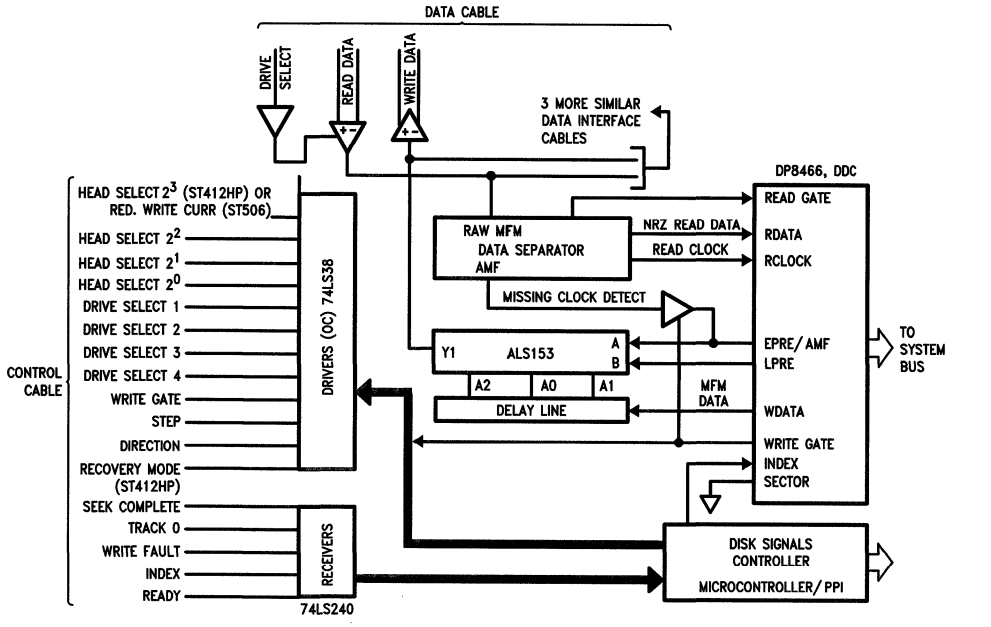


FIGURE 6.13. DDC and DP8463-2,7 ENDEC (b) Specific DP846X Solution



TL/F/8663–A4

FIGURE 6.14. Disk Data and Control Paths (ST506/ST412HP)

The necessary sequence of events (with associated timing restrictions) for proper read/write operation of the STxxx drive are shown in *Figure 6.15*. The DDC specifications are in compliance with the above requirements. The DDC has an on board encoder for MFM encoding and provides the precompensation outputs EPRE/LPRE, which are used by some external logic to generate the delays, as shown in *Figure 6.16(a)* and *(b)*

WRITE DATA PATH—PRECOMPENSATION

This consists of a differential pair that defines the transitions to be written on the track. The transition of the +MFM WRITE DATA line going more positive than the −MFM WRITE DATA line will cause a flux reversal on the track provided WRITE GATE is active. To ensure data integrity at the error rate specified for the interface, the write data presented by the DDC must be precompensated on the inner tracks. The optimum amount of precompensation is drive dependent, but usually is around 12 ns for both early and late written bits. In the DP8466 precompensation will be indicated on the EPRE and LPRE pins. Precompensation is issued for the middle bit of a 5-bit field. In the DDC, early and late precompensation will be enacted for all the combinations as shown below. All other patterns will not require precompensation. The center bit column is the present bit being output. The left two bit column is the bits previously shifted out, and the right two bit column is the two bits that will be shifted out next. In the following table a "bit" is a clock or data bit.

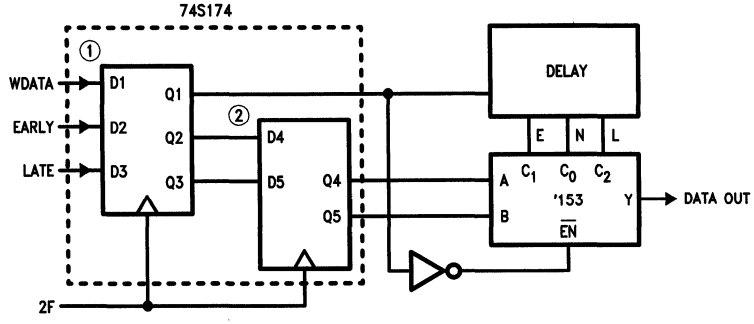
EPRE Patterns	LPRE Patterns
00 1 10	00 1 10
00 0 11	00 1 11
01 1 00	10 0 00
01 1 01	10 0 01
11 1 00	10 1 10
11 1 01	10 1 11

READ DATA PATH—READ GATE

For the read data path the data separator is the critical block. Its design was discussed in chapter 3. In addition how the controller cycles the read gate will affect performance. In case of conventional soft sectored drives the Read Gate cycling by the DDC is different for different conditions. In the case when the address mark is not detected after lock has occurred (abort address mark function—internal to the chip), the DDC deasserts Read Gate 19 RCLKS after getting a non-zero bit on the Read Data line, where it has been receiving an all zeroes data. It will then reassert the Read Gate 17.5 RCLKS later. In the situation where there is a sync failure, Read Gate is deasserted 10 RCLKS after the failed sync word, and reasserts it 17.5 RCLKS later. In case of a Header failure or a CRC failure, Read Gate is deasserted 2 RCLKS after the last check byte and is reasserted 25.5 RCLKS later. *Figure 6.15(c)* shows the Read Gate timing details.

HANDLING THE READ GATE IN SOFT SECTORED ST506 DRIVES

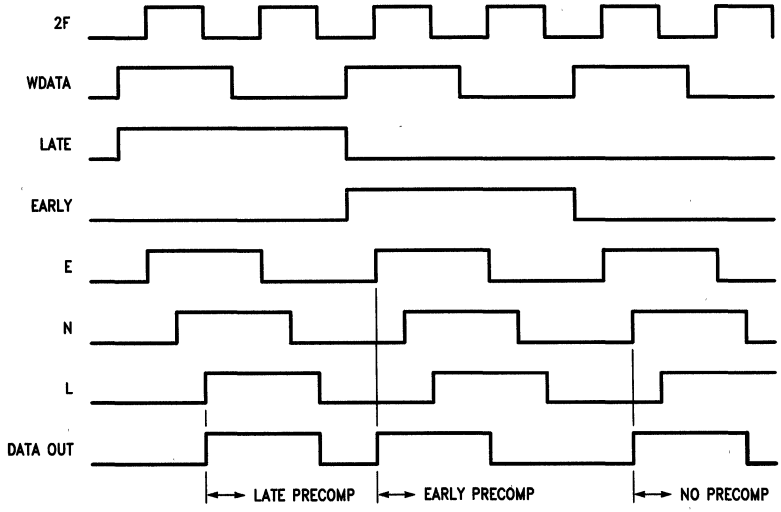
When reading soft-sectored drives, it is difficult to predict when Read Gate will be asserted. Data patterns can appear as preamble and the PLL will lock to these patterns. The controller is able to determine that the field is not a preamble by the address mark signal not being asserted because no missing clock violation was detected and deasserts Read Gate. However, Read Gate might be asserted over a write splice, which may result in the Data Separator failing to lock properly. It is usually up to the data separator design to ensure proper lock. The DDC does not implement a read gate on/off cycling algorithm. If the data separator can be thrown out of lock, the data separator should incorporate this algorithm. If not, this may lead to failure when accessing a sector in certain soft sectored drives. This is only a problem for drives using the conventional soft sectored format



TL/F/8663–12

FIGURE 6.16 (a). Precompensation Circuitry (MFM Encoding)

Note: Latch ② essentially ensures that the mux is enabled before the early and late signals arrive.
The early, normal, and late compensated data are only relative to each other.
Precomp time is drive dependent, usually 12 ns.



TL/F/8663–13

FIGURE 6.16 (b). Timing

(preamble, address mark, sync for both the ID and DATA segment), which is typical of lower performance, low capacity drives, operating at data rates of 5Mbits/sec and below, like the STxxx family. Newer standards replace the A1 (Hex) missing clock address mark currently used in soft sectored drives with an address mark which is a gap of no transitions at the beginning of the sector. Such a format is referred to as "start with address mark" format. The SMD and ESDI interfaces currently specify this type of operation and should be typical of those drives operating at 10 Mbits/s or greater.

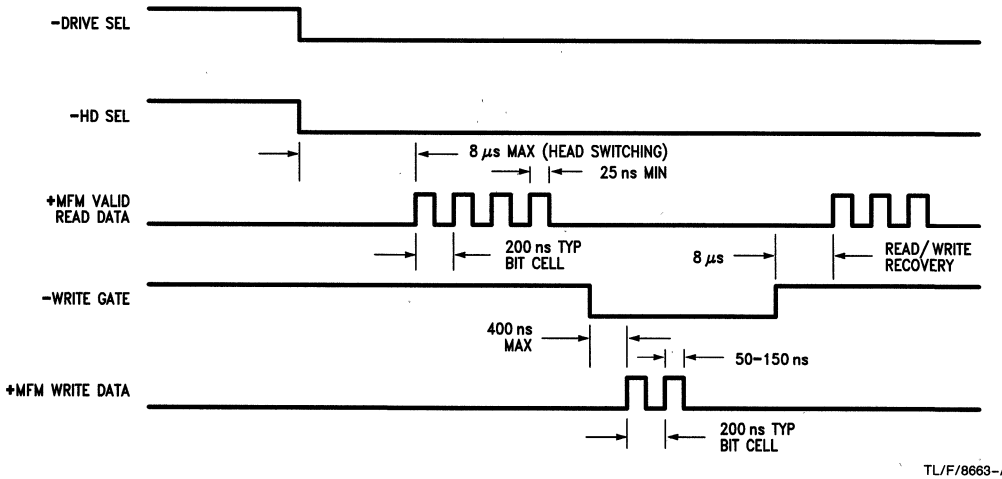
Operation of the DDC Read Gate in Soft Sectored Drives

In a soft sectored drive, the DDC asserts Read Gate at an arbitrary point over the track. This initiates a lock sequence on the Data Separator and the DDC begins to monitor the pattern entering its shift register. To avoid issuing garbage to the controller while it is locking up, the data separator usually will issue zeroes to the controller as it is looking for the first bit of non-zero data signifying lock. If using the 8465, when SET PLL LOCK is asserted, the Data Separator will begin to issue decoded data. In most applications, the LOCK DETECT output is connected to the SET PLL LOCK input, so that when the Data Separator has locked, it will issue data. However, if the Data Separator were to fail to lock and in some manner gets hung up where it will never be able to detect a preamble pattern, the system will be deadlocked since the DDC still is being given all zeroes from the Data Separator. The read operation will be aborted only after 2 revolutions. This type of failure can occur if Read Gate is asserted over write splice areas. The problem is really twofold, the Data separator is not issuing non-zero data to the DDC so that it can deassert Read Gate (the DDC thinks that the Data separator is still looking for the preamble) and the DDC is not placing a timeout on a response from the Data Separator.

Suggested External Logic to Timeout the Read Gate

If necessary a simple external circuit can perform the time out function. When Read Gate is asserted a counter is started. After 3 to 4 bytes (allowing time for the Data Separator to acquire lock), the SET PLL Lock on the Data Separator is driven low. This enables data to be sent from the Data Separator to the DDC. The DDC monitors the data and if a '1' propagates through the deserializer without matching the first pattern programmed for the format, Read Gate is deasserted. The DDC deasserts Read Gate for approximately 19 bit times, *Figure 6.17(a)*, hence the Data separator should be able to resynchronize to the 2f clock within this time. Hence even in the worst case, the amount of preamble lost by this scheme is 6–7 bytes, (1 byte for the DDC to deassert Read Gate after receiving a '1', pattern not matched, 2 bytes of Read Gate deasserted time and 3–4 bytes of timeout, essentially the lock time of the Data Separator). The suggested circuit is shown in *Figure 6.17(b)*. This circuitry could also be implemented in a PAL.

Read/Write Data Timing



TL/F/8663–A5

FIGURE 6.15. ST506 Read/Write Data Timing

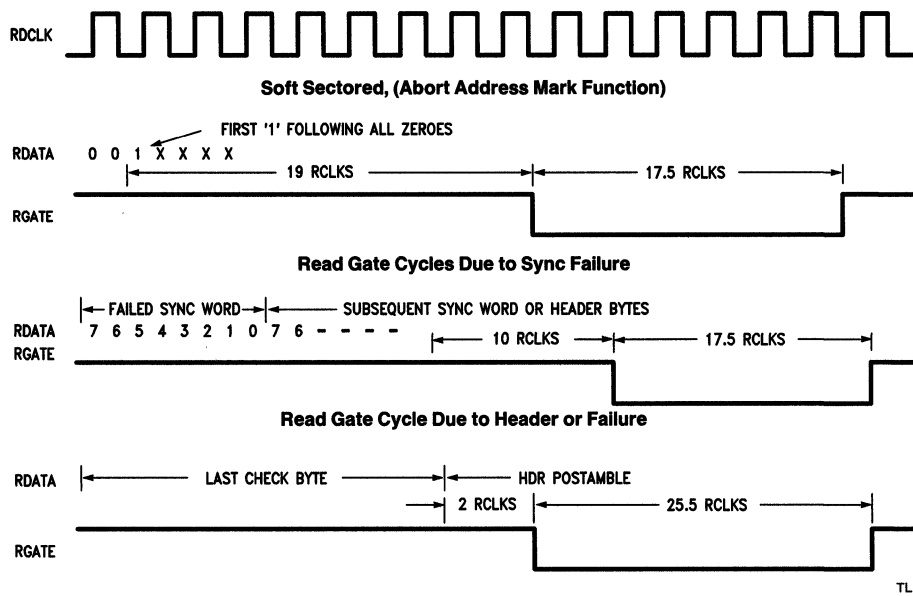
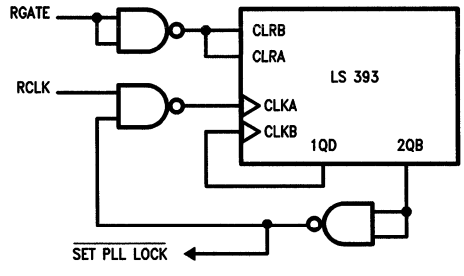


FIGURE 6.17 (a). Read Gate Timing (Cycling)



Counter timer using dual 4 bit binary counters cascaded. Count is stopped by decoding bit 2QB which is reached after 32 RCLKS, counter is reset when RGATE is deasserted.

FIGURE 6.17 (b). Read Gate Timeout Circuit

6.2.3 Interfacing the DDC to the ESDI Standard (Serial Mode)

The Enhanced Small Device Interface (ESDI), is a low cost, high performance interface suitable for the smaller memory devices currently on the market. It supports higher data transfer rate, 10–15 Mbits/s and provides for additional performance features desirable in higher performance systems. Two modes of implementation are possible: Serial mode of operation, utilizing NRZ data transfer along with serial commands and serial configuration/status reporting across the command cable (J1). Step mode implementation utilizes the same NRZ data transfer; however, the STEP and DIRECTION lines are used to cause actuator motion. This is similar to the ST506 type of interface as far as interfacing is concerned.

The ESDI interface consists of a 34-pin control cable and a 20-pin data cable. The control cable is attached in a daisy chain configuration while the data cable must be attached in a radial configuration. Hence all the control signals are associated with the drive selected. All the control lines are digital in nature (open collector TTL) and either provide signals to the drive (input) or signals to the host (output). The control signals and the serial commands are handled by the Disk Signals Controller, which is some local intelligence, as shown in *Figure 6.18*.

The ESDI interface can support both Hard and Soft sector drives. The interfacing of the, DDC is slightly different for hard versus soft sectored drives. *Figure 6.18* shows the interfacing of the DDC to a Hard sectored ESDI drive in the serial mode. In this configuration the sector pulse from the drive is used to identify the start of a sector. All lines associated with the transfer of data between the drive and the host system are differential in nature and may not be multiplexed. These lines are provided at the J2/P2 connectors on all drives. Four pairs of balanced signals are used for the transfer of data and clock.

NRZ Write Data: This defines the data to be written on the disk and is clocked by the WRITE CLOCK signal. This defi-

nition is compatible with that of the DDC. The Write data and Write Clock timings are shown in *Figure 6.19*.

NRZ Read Data: The data recovered by reading previously written information is transmitted to the host system via the differential pair of NRZ Read Data lines. This data is clocked by the READ CLOCK signal. These lines must be held at a zero level until PLL synch has been obtained and data is valid. This is compatible with what the DDC expects. One note of caution, when the PLL is locking on to the preamble, the DDC is receiving 0's and is looking for a non-zero synch byte. If the PLL goes off to harmonic lock or never locks and continuously outputs NRZ 0's, the DDC will assume it is forever looking at the preamble and will finally abort the command after two index pulses, flagging a sector not found error. It is up to the drive designer to prevent the PLL from going into harmonic lock. The DP8465, DP8461 and DP8462 are designed to remove harmonic lock. *Figure 6.19* shows the timing requirements for the read data.

Read/Reference Clock: *Figure 6.19* depicts the necessary sequence of events (with associated timing restrictions) for proper read/write operation of the ESDI drive. The Reference Clock signal from the drive will determine the data transfer rate. The transitions from Reference Clock to Read Clock must be performed without glitches. Read Clock and Read Data are valid within the number of PLL sync field bytes specified by the drive configuration after read enable and a PLL sync field is encountered. The interface Read/Reference Clock line may contain no transitions for up to two Reference Clock periods for transitions between reference and read clocks. The transition period will also be one-half of a Reference Clock period minimum with no shortened pulse widths. This is compatible with the specifications of the DDC, which has setup and hold times typically of the order of 15 ns. Reference Clock is valid when Read Gate is inactive while Read Clock is valid when Read Gate is active & PLL sync has been established.

Write Clock: Write Clock is provided by the DDC and must be at the bit data rate. This clock frequency is dictated by the Read/Reference clock during write operations. The DDC complies with the ESDI standard which requires the Write Clock to be active when Write Gate is active. See *Figure 6.19* for timing.

READ AND WRITE TIMING

Write Gate

The active state of this signal enables write data to be written on the disk. The low to high transition of this signal often creates a write splice and then initiates the writing of the data PLL sync field by the drive. The timing restrictions on Write Gate in the ESDI specification are as follows:

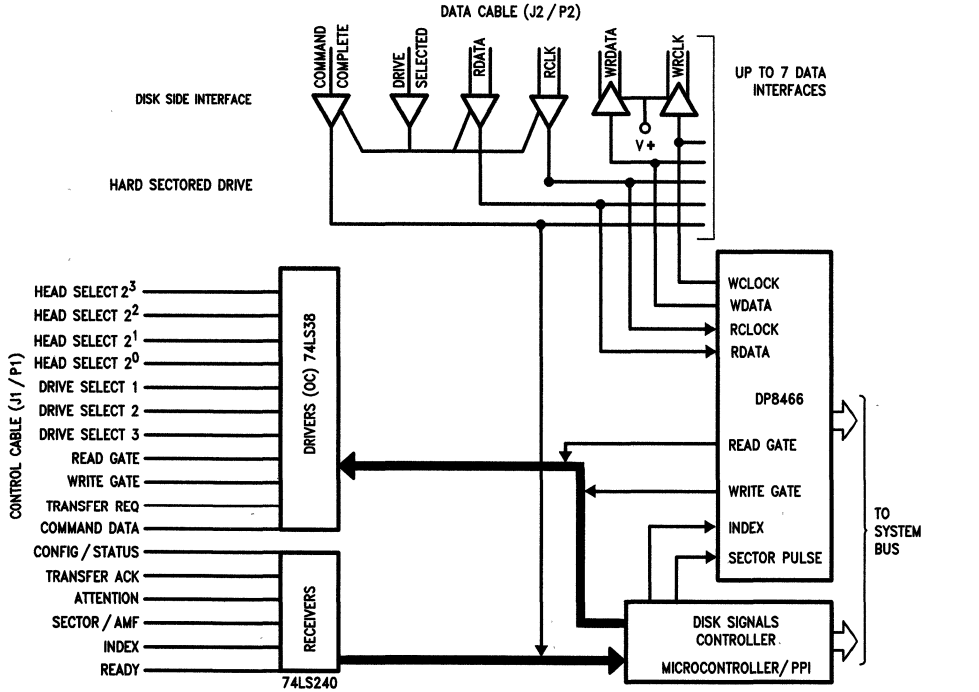
1) When formatting, Write Gate should be deactivated for 2 bit times minimum between address area and the data area to identify to the drive the beginning of the data PLL sync field.

2) It should be asserted at least two and a half reference clock periods after Write Clock.

3) The time lapse from deactivating Read Gate to activating Write Gate should be a minimum of 5 reference/read clock periods.

4) To account for data-encoding delays, Write Gate must be held on for at least two byte times after the last bit of information to be recorded.

5) It should be deactivated at least 1 μs before a head change and may not be activated until 15 μs after a head change.



TL/F/8663–A9

FIGURE 6.18. Data and Control Paths (ESDI-Serial Interface) Hard Sectored Drive

Read Gate

The active state of this signal, or low level, enables data to be read from the disk. This signal should become active only during a PLO sync field and at least the number of bytes defined by the drive prior to the ID or Data Sync bytes. The timing restrictions on Read Gate for the ESDI specification are as follows:

1) Read Gate must be false when passing over a write splice area. It must be deactivated 1 bit time min. before a Write Splice area and may be enabled 1 bit time min. after a Write Splice area.

2) The time lapse before Read Gate can be activated after deactivating the Write Gate is 10 μs.

The Read/Write Gate timings for format, write and read operations are discussed below with reference to the DDC, demonstrating its compatibility with the ESDI specifications.

Format Sector

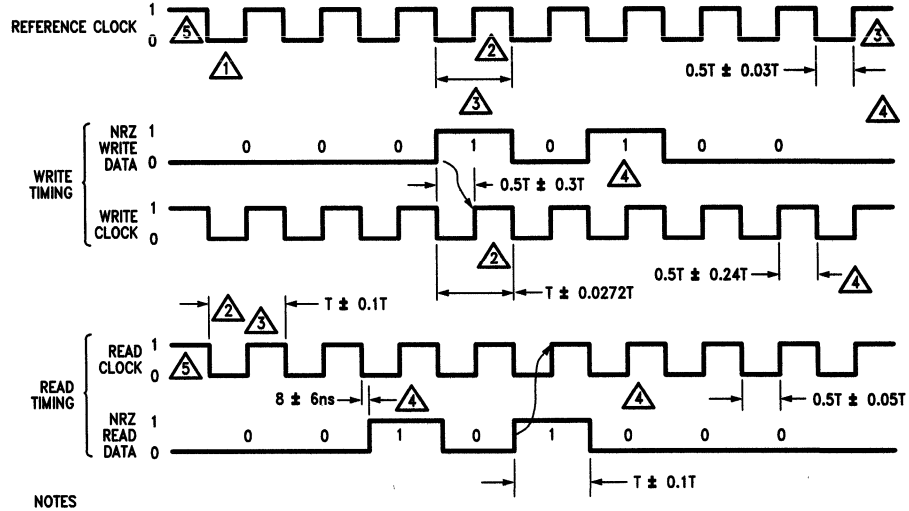
In the case of the DDC, during a format operation, it will do a continuous operation on the whole track. Thus after the index hole is sensed the DDC will assert WRITE GATE within 3.5 bit times. Write Gate will remain asserted until the index hole is sensed again. When each sector pulse is sensed (index for sector 0) the DDC will immediately start writing the various fields. After the data postamble is written the DDC will fill the rest of the sector with the gap until the next sector pulse is sensed. *Figure 6.20* shows the basic timing and the ESDI recommended format.

Write Sector

In case of a compare header-write data operation, the DDC will assert Read Gate within 3.5 Read Clocks from the rising edge of the Index or sector pulse. Read Gate is de-asserted 2 Read Clock periods after the ID check field, (plus a small propagation delay). Read Gate will remain de-asserted for the entire postamble. Due to internal delays, Write Gate is asserted 3 bit times into the data preamble field. Hence this would meet the 5 bit time minimum spec. between the read gate de-assertion and write gate assertion assuming at least one ID postamble byte is programmed. At the end of the write operation, Write Gate is removed 0 bit times after the data postamble. Hence a 3 bit time 'pad' is created between header postamble and data preamble updating data. This pad will contain data preamble as written during format operation. Refer to *Figure 6.20* for basic timing. It should be noted that a write splice of 8 bits is associated with the assertion of Write Gate. Hence if a write header operation is involved, then the read gate should not be asserted in the splice area.

Read Sector

In case of a compare header-read data operation, the DDC will assert Read Gate within 3.5 Read Clock cycles from the rising edge of the Index or sector pulse. Read gate is de-asserted 2 Read Clock cycles after the ID check field. Read Gate is re-asserted 11.5 bit times from the data preamble. This is 8.5 bit times from the point where Write Gate is as-



NOTES

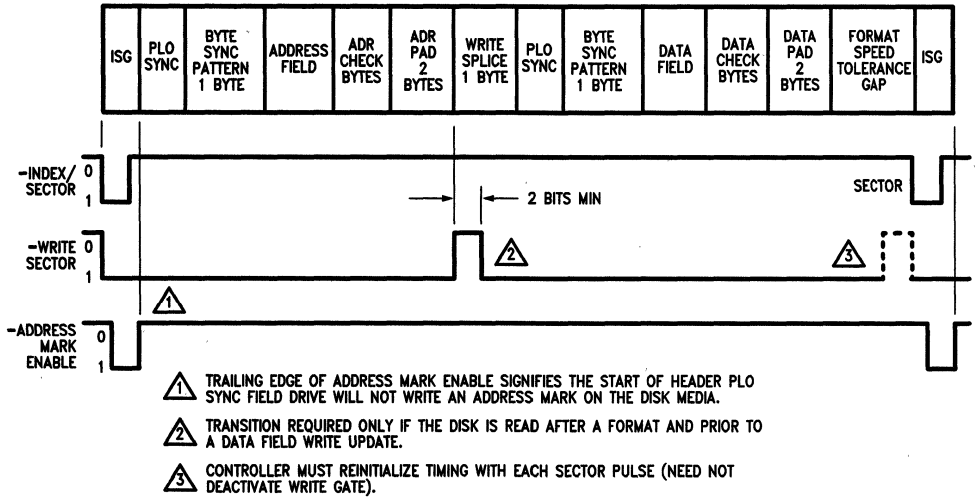
1. ALL TIMES IN ns MEASURED AT I/O CONNECTOR OF THE DRIVE T IS THE PERIOD OF THE CLOCK SIGNALS AND IS THE INVERSE OF THE REFERENCE OR READ CLOCK FREQUENCY.
2. SIMILAR PERIOD SYMMETRY SHALL BE IN ± 4 ns BETWEEN ANY TWO ADJACENT CYCLES DURING READING OR WRITING.
3. EXCEPT DURING A HEAD CHANGE OR PLO SYNCHRONIZATION THE CLOCK VARIANCES FOR SPINOLE SPEED AND CIRCUIT TOLERANCES SHALL NOT VARY MORE THAN −5.5% TO +5.0% PHASE RELATIONSHIP BETWEEN REFERENCE CLOCK AND NRZ WRITE DATA OR WRITE CLOCK IS NOT DEFINED.
4. TIMING APPLICABLE DURING READING OR WRITING.
5. REFERENCE CLOCK IS VALID WHEN READ GATE IS INACTIVE READ CLOCK IS VALID WHEN READ GATE IS ACTIVE AND PLO SYNCHRONIZATION HAS BEEN ESTABLISHED.

TL/F/8663–B0

FIGURE 6.19. NRZ Read/Write Data Timings

serted. This accommodates the 8 bit time write splice generated due to write driver turn on time as required in the ESDI specification. Read Gate is de-asserted 2 bit times into the data postamble. Of particular importance is that the read operation avoid reading the write splice. As can be seen from the write sector discussion, the write splice will occur 3 bits into the preamble, and the read will occur after 11.5 bits. Thus Read Gate is disabled during the actual write splice. When the read and write operations use the same format parameters, Write Gate will cause a splice 8.5 bits before Read Gate is asserted assuming insignificant delay in the read path from the media. However this provides a maximum of 8.5 bit times the write data out could be delayed by the write data encoder, in the drive, prior to being written on the media. *Figure 6.20* gives the basic timing. The user may have to account for additional delays specific to the drive.

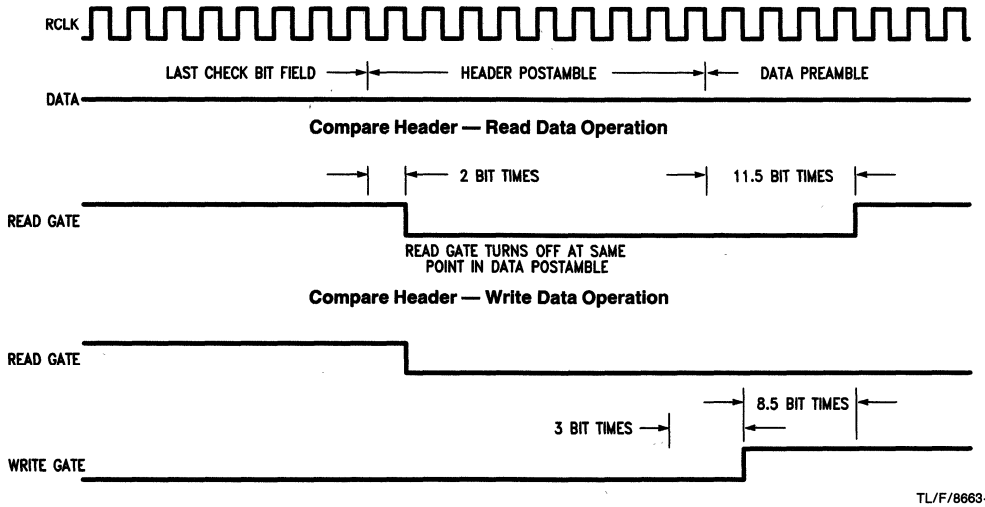
Write Gate in ESDI with Index/Sector, AME



TL/F/8663–B1

FIGURE 6.20 (a). Read/Write Gate Timing for DDC

Read/Write Gate Timing at End of ID; Beginning of Data Field



TL/F/8663–B2

FIGURE 6.20 (b). Read/Write Gate Timing for DDC

6.2.4 Special Consideration for ESDI Drives

SOFT SECTORED

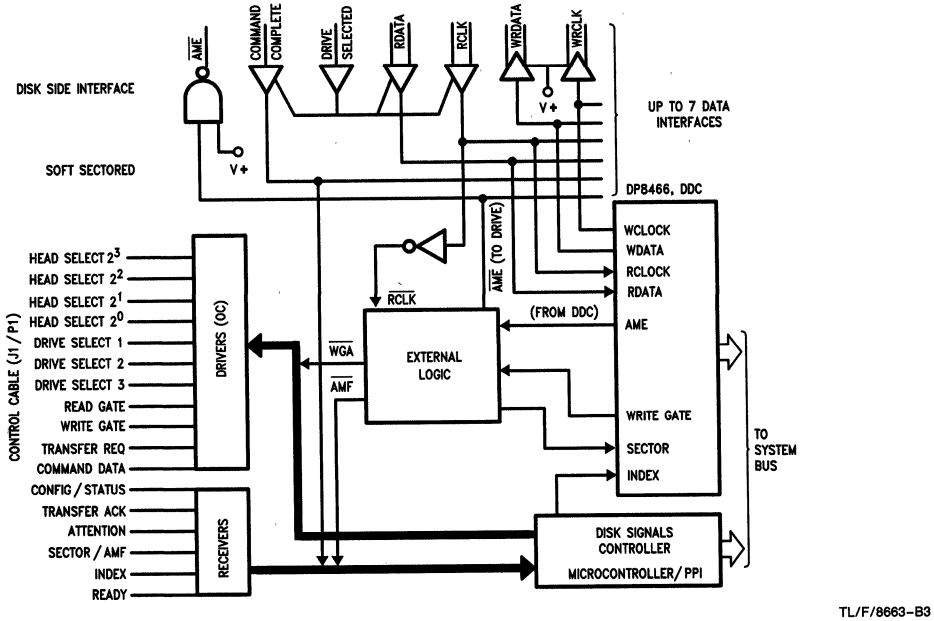
Interfacing the DDC, to a soft sectored ESDI drive is slightly different. There are no sector pulses demarcating sectors. An Address Mark pattern of no flux transitions is used to identify the start of the sector. The ideal soft sectored format (as supported by the DDC) consists of the Preamble field followed by the Address Mark (AM) and Sync fields. However in the case of ESDI the soft sectored format specification supports an AM at the beginning of the sector followed by the Preamble and Sync fields. This is a special field of no flux transitions which is essentially used to mark the start of the sector. Address Mark detection and generation is done using a handshake protocol between AME and AMF on the drive cable. Since the AMF on the DDC is geared to look for a missing clock violation in the address mark pattern, it cannot be used in the ESDI handshake. Also the DDC generates AME only during the format operation. Hence to achieve successful operation of the ESDI soft sectored drive, the DDC formats the drive as a soft sectored drive and while reading, the AMF signal from the drive is used to generate a sector pulse, as the DDC is configured to operate in the hard sectored mode. It also requires manipulation of the format parameters as shown in *Figure 6.22*. *Figure 6.21* shows the schematic of the interface at a block level.

Let us first consider manipulation of the format parameters to achieve proper operation. *Figure 6.22* gives the sequence of events. Consider the first three fields of the format. The DDC views them as Preambie, AM and Sync, *Figure 6.22(a)*. The DDC format is programmed as AM, Preamble and Sync for the first three fields, *Figure 6.22(b)*. If the format operation is initiated with SAM (start on address mark) bit set in the Disk Format register, the recording on the media is shown in *Figure 6.22(c)*. The AME output on the DDC is asserted during the Address Mark (field 1). During a Compare header-read/write operation the first field of the format parameter RAM in the DDC is programmed to be the Preamble. The count for the second field is set to zero, so that it is skipped, while the third field is progammed to be the Sync field. When data is read, the field 1 of no transitions (AM) results in AMF becoming active which generates the sector pulse for the DDC, *Figure 6.22(d)*.

In the ESDI specification, AME (address mark enable) line, when active with Write Gate causes an Address Mark to be written on the media. When AME is active without Write Gate or Read Gate, it causes a search for Address Marks. On detection of the end of Address Mark, AMF (address mark found) responds. The trailing edge of AME with Write Gate true initiates the writing of the header PLO Sync field.

To incorporate the AME/AMF handshake, external logic is required with the DDC. Since the DDC generates AME only during a format operation, the circuit would have to generate AME to the drive during a read operation and incorporate the handshake with AMF from the drive. This AMF is used as a sector pulse input to the DDC. This technique then results in a sector pulse corresponding to each address mark pattern demarcating each sector. However at sector 0 this poses a problem. The index pulse is followed by the post index gap and then the address mark field of sector 0 which would generate a sector pulse. As per ESDI requirements this circuit would have a delay and present the index pulse over the sector 0 pulse and block out the sector 0 pulse to the DDC. The other constraints to be maintained in order to comply with ESDI spec requirements are:

1) AME should be asserted min 100 ns after min write gate is asserted and be deasserted 100 ns before write gate is deasserted.
2) AME can be asserted again at least 10 μs after write gate deassertion.



TL/F/8663–B3

FIGURE 6.21. Data and Control Paths (ESDI-Serial) Soft Sectored Drive

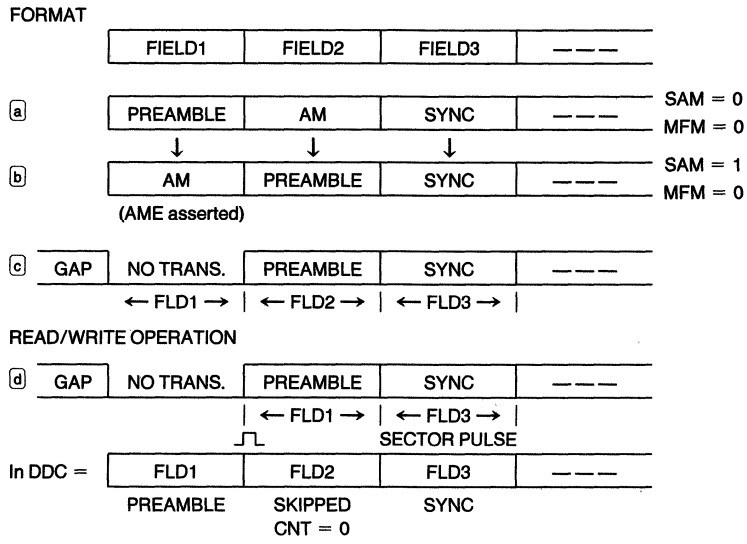


FIGURE 6.22. Manipulation of Format Parameters in DDC for ESDI Soft Sectored Operation

HARD SECTORED ESDI

For an ESDI drive which is hard sectored, the ESDI specification calls for an Inter Sector Gap (ISG) which is to precede and follow the index/sector pulses. The gap is needed to provide the drive with an area for the embedded servo (if used), and gives the controller time to assert read gate. While formatting the drive, the end of the ISG is indicated by the removal of the address mark enable signal (AME). This signal is needed by the drive to indicate the beginning of the PLL field, necessary when the disk encodes the PLL field with a non-standard pattern (as with 2,7 encoding with 3t preambles). The DDC is capable of generating the AME signal with the necessary timing. The DDC needs to be in the hard sectored mode, and have the Start with Address Mark bit (SAM bit of the DISK FORMAT register) enabled. The DDC ID Preamble field now becomes the ISG following the index/sector pulse, and the ID Sync 1 field becomes the PLL preamble field. While not formating, this feature is not needed, and should be disabled.

When the header field of an ESDI drive is read (or compared), the read gate to the drive needs to be delayed until after the ISG. The DDC generates read gate only after receiving a sector or index pulse, so by delaying the sector and index signals to the DDC the read gate will be delayed.

COMBINED SOLUTION

A solution to the above problems can be provided by 1 PAL device and something to provide a delay (possibly another PAL device) *Figure 6.23(b)*. The interface solutions can be grouped into two main areas: Address marks and Index/Sector.

The address mark control needs to provide the following:

(1) Direct connection of AME to the drive and AMF to DDC sector input when formating a hard sectored drive.

(2) Delay the leading edge of the AME by the width of post index ISG when formating a soft sectored drive.

(3) Provide AME/AMF handshaking when not writing the disk, and properly change from reading to writing and back with soft sectored drives.

The index/sector control needs to provide the following:

(1) Delay the index and sector pulses from a hard sectored drive when not formating.

(2) Generate index and sector pulses to the DDC from AMF and Index when using a soft sectored drive while not formating.

The Address Mark Machine

The address mark machine consists of a pair of multiplexers which feed the AME input to the disk drive and the AMF input to the DDC. A state machine ensures the proper sequence of events, while a timer provides delay. The address mark machine works in the following way:

When in the soft sectored mode and not formating, it will assert AME to the drive. When AMF is detected, AME is removed until AMF is no longer detected. This uses states 0 and 3 of the address mark state machine, as shown in the diagrams, *Figure 6.23(a)*.

When write gate is detected, AME is removed, and write gate to the drive is generated a short time later. When the write operation is ended, write gate is removed from the drive, and AME is enabled a short time later.

In all other modes of operation, the state machine is deactivated and write gate is delayed to the drive by one bit clock time (circuit convenience). The multiplexer continues to provide the drive and DDC with proper signals.

The Index/Sector Machine

The index/sector machine consists of a pair of multiplexers which feed the index and sector inputs of the DDC. While formating, the multiplexers connect the index and sector signals from the drive to the DDC.

When not formating a hard sectored drive, the state machine waits for either a sector or index pulse. When this is detected, the machine waits for a delay and then generates a sector or index pulse. A flip flop, borrowed from the address mark machine, is used to record whether an index or sector pulse should be generated (the address mark machine is disabled when in the hard sectored mode). The delayed index or sector pulse is generated for another delay period of time.

With a soft sectored drive, the state machine waits for either an index or AMF signal. If index is detected, the machine will wait until an AMF is detected. An index pulse will then be sent to the DDC without a sector pulse. If a AMF pulse is detected without index, the state machine will generate a sector pulse.

6.2.5 Interfacing the DDC to the SMD Interface Standard

The Storage Module Device (SMD) interface is a high performance interface, extremely popular with 8″–14″ drives. It provides features similar to the ESDI interface, with some differences. It supports higher data rates from 10 Mbits/s to 24 Mbits/s and utilizes NRZ data transfer along with parallel command and status reporting across the command cable. The SMD interface consists of a 60-pin control (A) cable and a 26-pin data (B) cable. The control cable is attached in a daisy chain configuration while the data cable must be attached in a radial configuration. The control signals are handled by a separate Disk Signals Controller block, which is some local intelligence, as shown in *Figure 6.24*. Only the interfacing of the data path signals are discussed as they are of relevance to the DDC. The control signals could be easily done by a local microprocessor. All lines associated with the transfer of data between the drive and the host system are differential in nature and may not be multiplexed. These lines are provided on the B cable of all drives and are briefly discussed below:

Write Data

This line carries NRZ data, to be written on the disk surface and must be synchronized with Write Clock. This definition is compatible with that of the DDC.

Write Clock

This is a retransmitted clock signal of the servo clock (IF Write Clock) issued by the controller.

Servo Clock

This signal is used by the control unit in the drive to synchronize Write Data with the Clock. Servo Clock may be available during unit ready status except during read operations, or at all times.

Read Clock

This line transmits Read Clock. The Read Data is synchronized with 1F Read Clock. This line may be valid only during a read operation, or may be multiplexed with the Servo Clock signal at other times.

Read Data

This line transmits the recovered data in the form of NRZ data synchronized with 1F Read Clock.

Figure 6.25 shows the basic timing requirements for the above signals for the Fujitsu M2311 micro disk drive hereafter referred to as MDD. In general "SMD" type drives have similar timing requirements.

READ AND WRITE GATE

Read and Write Gate are the two signals which are used to read/write data from/to the specified track/sector. In the SMD interface, they are present on the Bus Out, (bit 1—read gate, bit 0—write gate), when enabled by tag 3. Write Gate enables the write operation and is validated only when Unit Ready, On Cylinder and Seek End are true and Seek Error, Fault, File Protect, Offset are false. If Write Gate is turned on in cases other than the above conditions, fault occurs and writing is inhibited. At this juncture it would be appropriate to mention that there are certain drive dependent constraints which must be taken care of while interfacing to the SMD drives. These are drive dependent, and representative values observed in most of the drives are given below:

Write circuit turn-on delay: approximately 8 bit times.

Head select transient: A 5 μs delay minimum must be provided between head select and initiating read gate. Normally this is provided by selecting the appropriate length for the gap after the data postamble and the gap after the index/sector pulse.

Read-after-Write transients: A minimum delay of 10 μs must be provided between the trailing edge of write gate and the leading edge of read gate. This could also be done by adjusting the lengths of the gap after the data postamble and the gap after the index/sector pulse.

Read/Write Encoding/Decoding delays: Through encoding and decoding circuitry, a read data signal will be delayed by approximately one byte against a write data.

Write-after Read transient: A minimum delay of 0.3 μs must be provided between the trailing edge of read gate and the leading edge of write gate. This is accomplished by having at least one byte of header postamble.

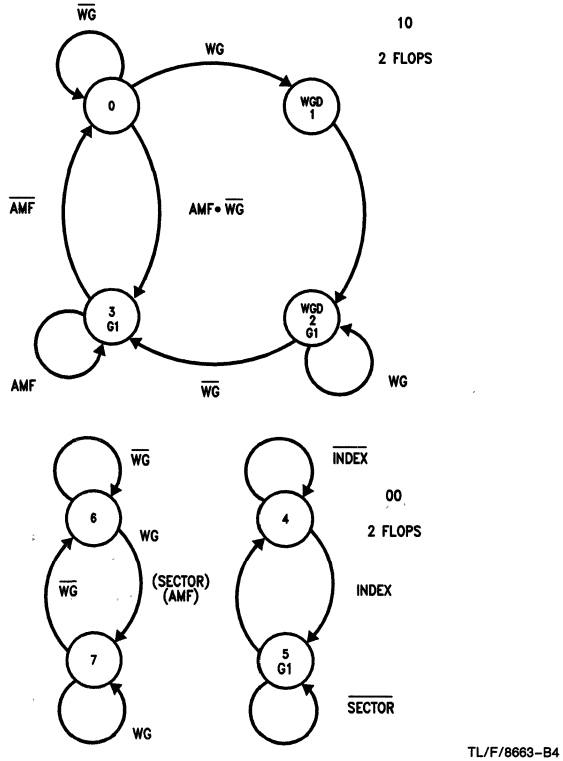


FIGURE 6.23 (a). AME/AMF/Index/Sector Logic State Diagrams

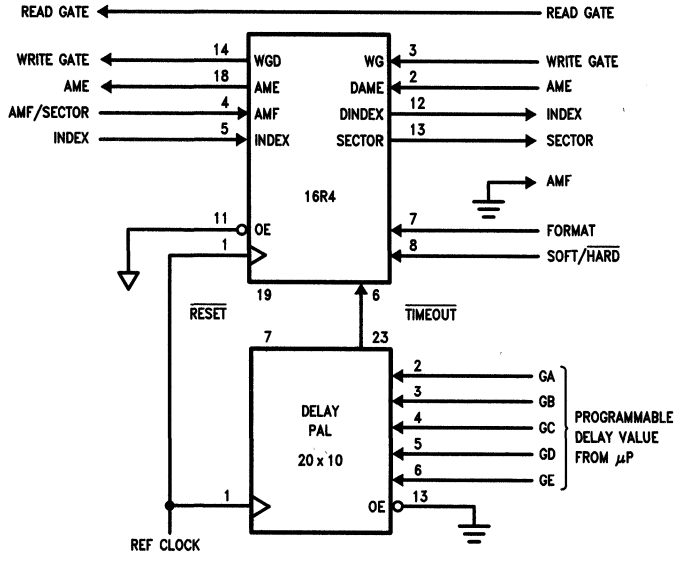
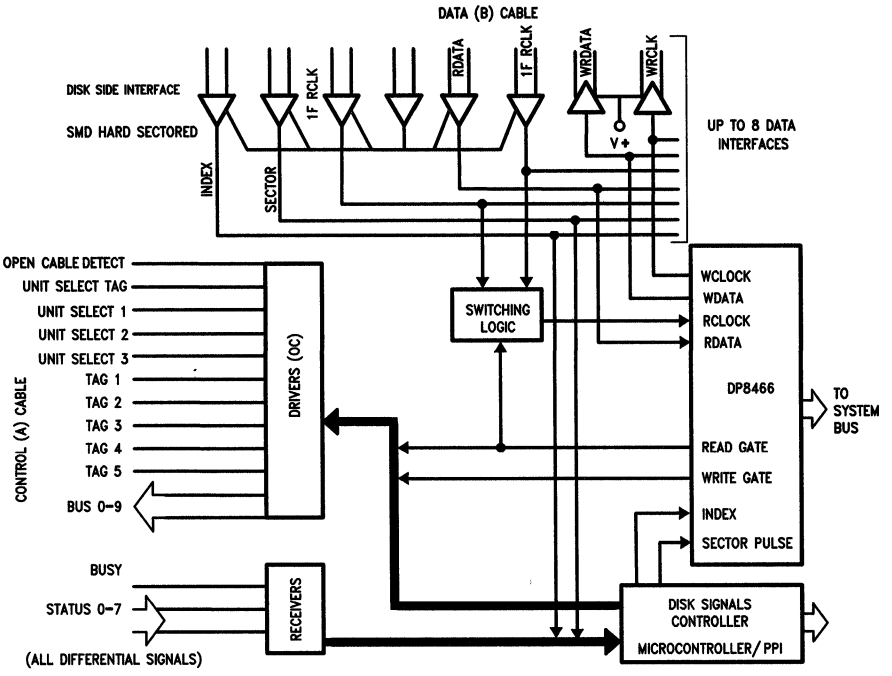


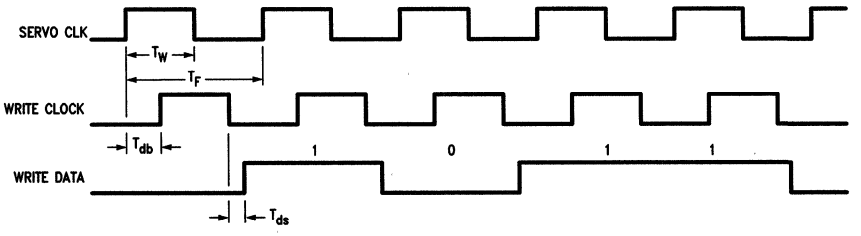
FIGURE 6.23 (b). ESDI AME/AMF Handshake Logic and State Diagram



TL/F/8663–B6

FIGURE 6.24. Data and Control Paths — SMD (Hard Sectored Drive)

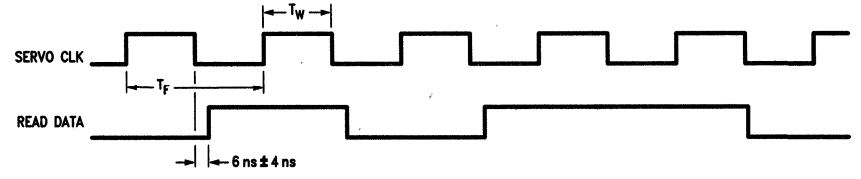
1F Write Clock, Write Data/Write Clock Timing



T_W = TF/2
TF = (Function of Transfer Rate)
Tdb = Continuous delay within 2 bits
Tds = 0 ±10 ns

TL/F/8663–B7

FIGURE 6.25 (a). Write Clock/Write Data Timing



T_W = TF/2

TL/F/8663–B8

FIGURE 6.25 (b). Read Clock/Read Data Timing

Preamble length: The synchronization time required to allow the PLL to synchronize is 11 bytes before the sync pattern of address and data fields.

READ/WRITE TIMING

Representative read/write timing for format write, data read and data write operations are shown in *Figure 6.26*. The corresponding timings for the DDC are given in *Figure 6.20*.

FORMAT WRITE

During a format operation or write header operation, the drive requires that the Write Gate be asserted by 600 ns (max.), from the index/sector pulse and Write Gate must be de-asserted at least 1 byte after the check byte field in the header, refer *Figure 6.26(b)*. In the case of DDC, Write Gate is asserted 3.5 bit times after the Index pulse in a format operation or after the index/sector pulse in a write header operation (hard sectored drive). Write Gate is de-asserted at the end of header postamble field. Hence if the header postamble is kept at least one byte long, the DDC should satisfy the requirements of the drive. *Figure 6.26(a)* shows the recommended format used by "SMD" drive manufacturers like CDC, etc.

DATA WRITE

During a data write operation (essentially a compare header-write data operation) read gate is asserted by the DDC 3.5 bit times from the rising edge of the index/sector pulse. The recommended format for the MDD has a post index/sector gap and requires the Read Gate to be asserted 6 μs (approx. 8 bytes) from the index/sector pulse. This can be done in a similar fashion as outlined for the ESDI spec. (refer section—Handling the Post Index/Sector Gap in the ESDI format). Read Gate is de-asserted 2 bit times after the ID check bits field, which satisfies the requirement of 8 bit times maximum. Write Gate is then asserted 3 bit times after the header postamble, which implies that the header postamble can be a maximum of 3 bytes long in order to maintain the requirement of 4 bytes max. by which Write Gate must be asserted after the header check field. There is also a condition that Write Gate must be asserted at least 300 ns after Read Gate is de-asserted. This is satisfied by the DDC as seen from *Figure 6.20(b)*, the minimum time of de-assertion being 9 bit times assuming at least a 1 byte postamble. Write Gate is de-asserted at the end of the data postamble, hence, the data postamble must be at least 4 bytes long. *Figure 6.26(c)* gives the timing requirements for a data write operation.

DATA READ

During a read operation (essentially a compare header-read data operation), DDC asserts the Read Gate 3.5 bit times after the index/sector pulse. Hence the post index/sector gap has to be handled in a similar fashion as discussed in the Data Write section. Read Gate is de-asserted by the DDC 2 bit times after the ID check bit field, which is well within the MDD requirement of 8 bits max. If the continuing operation is Read data, then Read Gate is re-asserted by the DDC 11.5 bit times from the data preamble, *Figure 6.20(b)*, which certainly satisfies the requirement of a 1 byte minimum before re-assertion of the Read Gate. Read Gate is de-asserted by the DDC 2 bit times after the data check bit field. *Figure 6.26(d)* gives the timing requirements of the data read operation.

SPECIAL CONSIDERATIONS FOR SMD SOFT SECTORED DRIVES

For a soft sectored "SMD" drive, the recommended format is similar to the one in ESDI. The sector starts with the address mark field (three bytes of no flux transitions). The AME signal is available on the bit 5 of the bus with Tag 3 active while the AMF signal is available on bit 5 of the Status bus with both Tag 4 and Tag 5 inactive. The AME/AMF handshake is handled in a similar fashion as discussed in section 6.2.4.

HANDLING THE SEPARATE CLOCKS FOR READ AND WRITE OPERATIONS IN THE SMD DATA CABLE

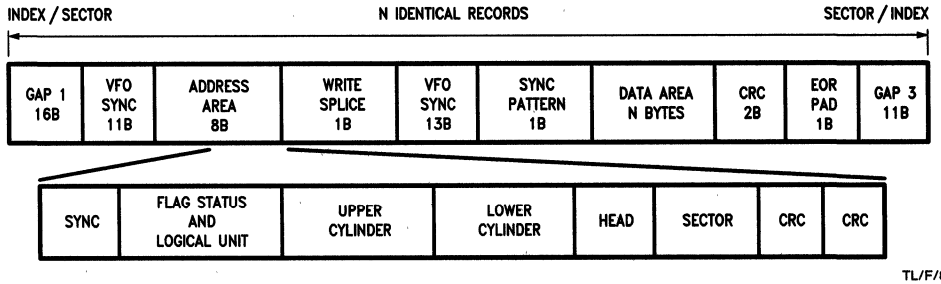
The SMD interface has two reference clocks, one for write (1F WCLK/SERVO CLOCK) and one for read (1F RCLK/READ CLOCK). The DDC requires that the Read/reference clock be provided on the same line (RCLK) and that the switching between the reference clock and the PLL locked frequency clock be such that there are no glitches on the line going to the DDC. To accomplish this the two clock signals, 1F Read Clock/READ CLOCK and 1F Write Clock/Servo Clock need to be multiplexed during read and write operations to switch the appropriate clock signal going to the DDC. It should also be made sure that there are no glitches or short pulses in the process of switching. Since the PLL has a certain finite time for lock, the actual switch of the clocks occurs after a finite lock time from Read Gate assertion. Hence the Read Gate used to mux the two clocks must be delayed by the lock time (worst case) before it is sent to the switching logic. This is shown as the box 'switching logic' in *Figure 6.24* and given in detail in *Figure 6.27*. Besides these other timing requirements are compatible with those of the DDC. This circuitry has also been realized in a PAL.

6.2.6 Miscellaneous ESDI/SMD Considerations

As with most standards, actual devices that follow the standards have their own idiosyncracies. ESDI and SMD standards are no exception. In addition some standards define design constraints which do not necessarily exist with actual devices. Some of these have been discussed earlier (for example de-assertion of Write Gate between ID and Data fields in ESDI). The following subsections describe some additional SMD & ESDI considerations that may be necessary depending on actual drive implementation.

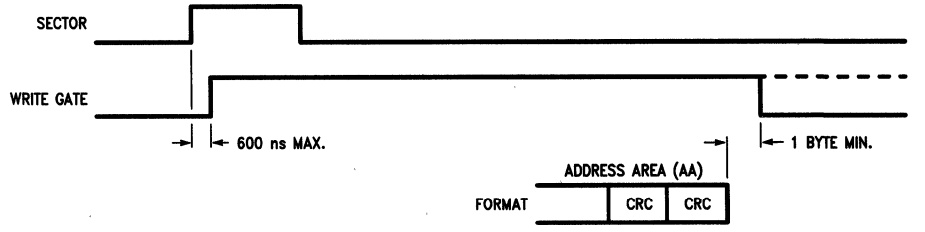
HANDLING THE OPTION OF DE-ASSERTION OF WRITE GATE BETWEEN THE ID AND THE DATA FIELDS, IN THE ESDI FORMAT SPECIFICATION

The option of de-asserting Write Gate between the ID and the Data field is not directly supported by the DDC. The purpose of this may be indication to the encoder, of the start of the data preamble field in case of RLL encoding, so that the encoder can send actual data rather than the encoded data for the preamble pattern. To support this, external logic can be used. This logic would be gated from the trailing edge of the DDC's Serial Data Valid signal, count until the 2 byte pad, (header postamble) has been sent (at the end of the ID segment), then force Write Gate low for the desired time, and then re-enable it. This problem can also be effectively circumvented by first doing a two pass format operation. This first pass does a format operation as above, then a second pass-write data operation is done on all of the sectors. If the Write Gate pulse is used to initiate a drive generated preamble, then by performing a compare header-write data operation the correct data preamble will then be written, and the data field can then be properly read, refer *Figure 6.18* for timing with respect to the DDC.



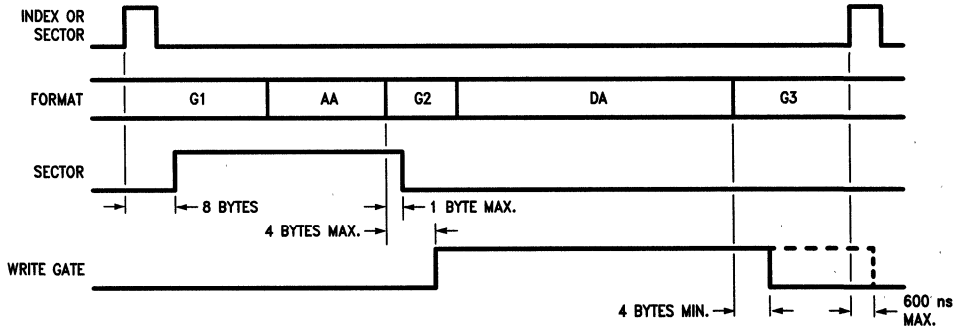
TL/F/8663–B9

(a) Format



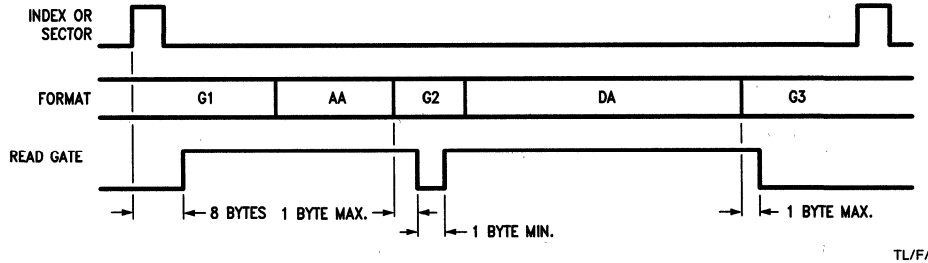
TL/F/8663–C0

(b) Format Write Timing



TL/F/8663–C1

(c) Data Write Timing



TL/F/8663–C2

(d) Data Read Timing

FIGURE 6.26. Read/Write Timing (SMD)

A note of caution to be observed for ESDI hard sectored drives during format operation for RLL encoding. Since the leading edge of the write gate may be used by the encoder to trigger the generation of the preamble, this would be a problem as Write Gate is asserted 3.5 bit times after the index pulse and there would be no de-assertion of Write Gate with each sector pulse. The two pass format operation is a good software solution. In hardware the inverted sector pulse could be ANDed with Write Gate to generate the Write Gate to the encoder. Generally after a format write, then read is necessary to determine defective sectors, and initiate some sector substitutions.

HANDLING THE WRITE SPLICE FIELD BETWEEN THE ID AND DATA SEGMENTS IN THE ESDI/SMD FORMAT

The ESDI/SMD format specification recommends a two byte header postamble and a one byte write splice. The DDC format parameters indirectly support a write splice field between the ID and data segments. Consider normal operation of the DDC. The format is programmed to have a 2 byte header postamble and a data preamble one byte longer than the desired length. This byte is taken as the write splice, (a floating byte). During a write operation this floating byte is considered as part of the data preamble, so write gate is asserted 3 bit times into the data preamble i.e., the 'write splice' and data would be written on the media after taking into effect the write propagation delay. In case of a read operaion this byte is taken to be part of the header postamble. Hence as Read gate is asserted after the header postamble, it would never be asserted in the write splice. Normal operation could be achieved with the DDC without physically having a write splice field between the ID and the Data segment, rather creating one by adjusting other field lengths.

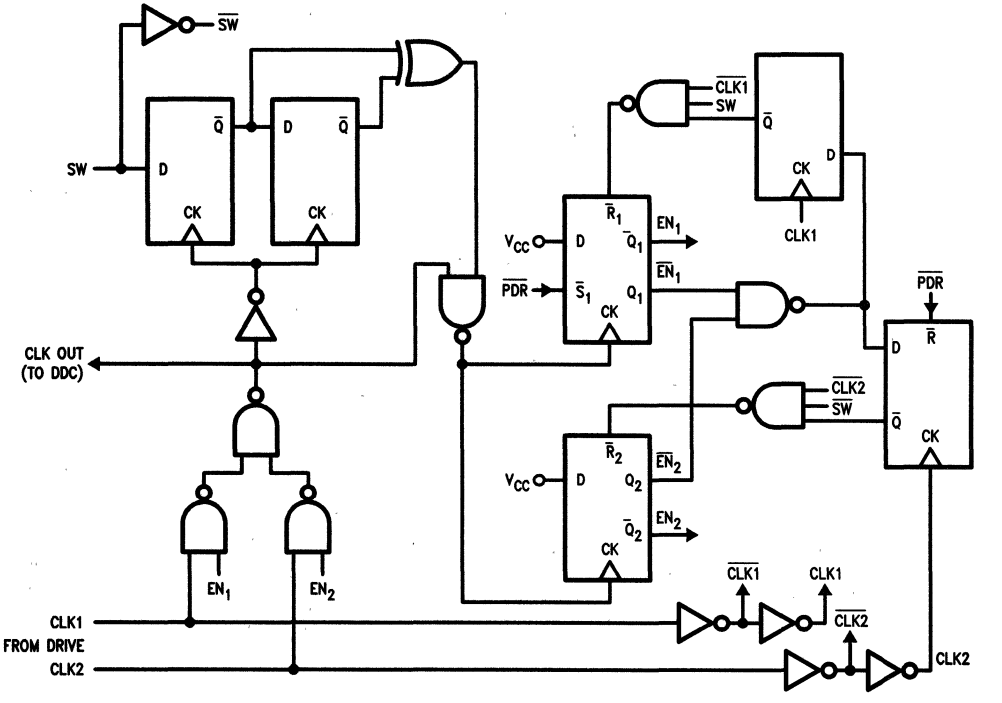


FIGURE 6.27 (a). MUX and Deglitcher Circuitry to Switch between Read Clock and Reference Clock

Note 1: SW is low at start up, (L selects CLK 2, H selects CLK 1)

Note 2: POR = Power on reset ($EN_2 \rightarrow 1$, $EN_1 \rightarrow 0$)

Note 3: Worst case latency (SW to actual switching) = 1.5 periods of clock switching from +2 periods of clock switching.

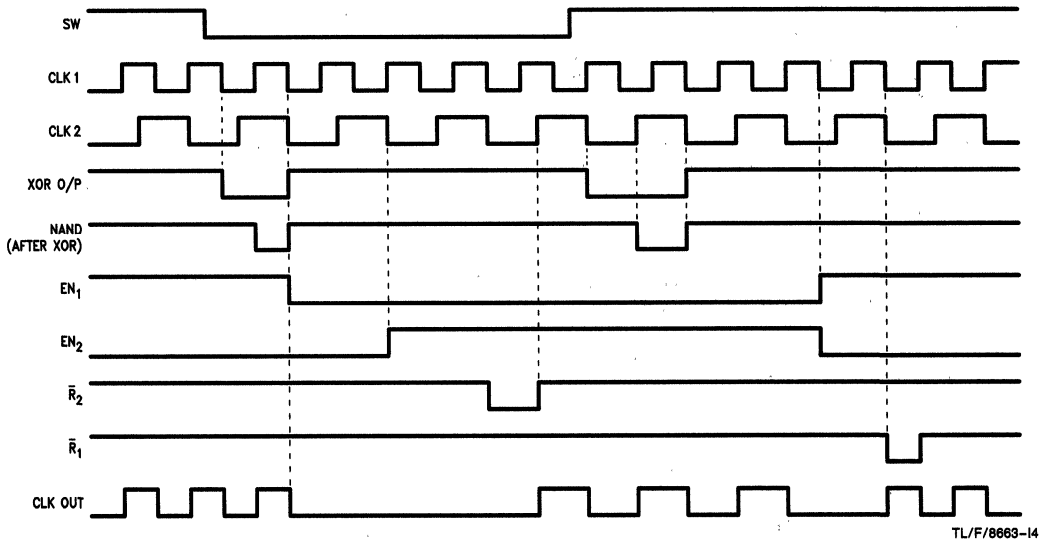


FIGURE 6.27 (b). Timing Diagram for Switching Logic

HANDLING THE POST INDEX/SECTOR GAP FIELD IN THE ESDI/SMD FORMAT SPECIFICATION

In the recommended format of the ESDI/SMD specification there is a gap after the index/sector pulse, referred to as the post index/sector gap. This is necessary mainly to accommodate head transients, read-to-write transients, write-to-read transients, etc. In the DDC, there is no format parameter to implement this field. Hence to implement this, external logic is needed, whereby the index/sector pulse to the DDC is delayed from the index/sector pulse from the drive by the desired gap count using counters. This is done for all format, read and write operations. The gap pattern for the intersector gap is then written for this post index/sector gap field also. Refer *Figure 6.28*.

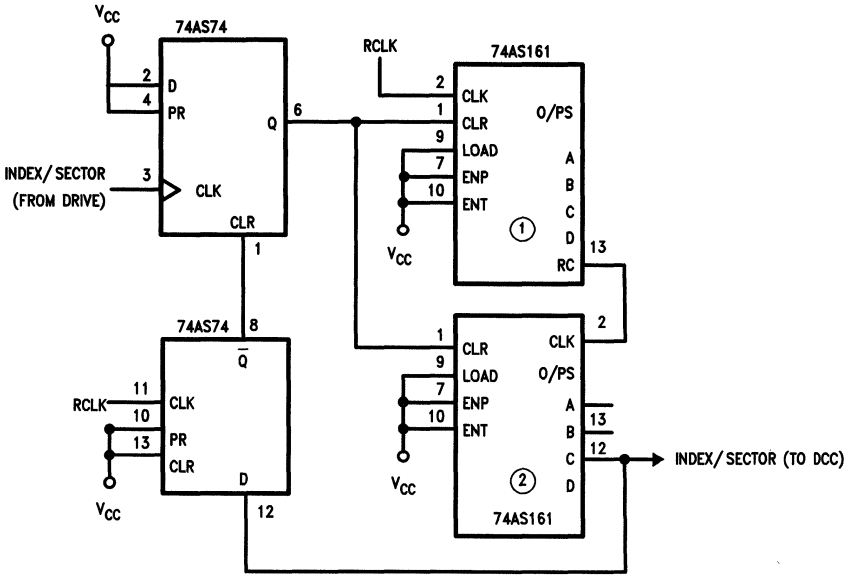
READ GATE DELAY

As discussed earlier, the separation between assertion of Read Gate and Write Gate at the beginning of the sector is 0.5 to −0.5 bit times. This may not accommodate the write splice associated with the Write Gate due to write driver turn on time etc., which is generally about 8 bit times from write gate assertion. Hence with the existing timing Read Gate may get asserted in the write splice area while reading the header. This might be a problem during the Format operation, in the very first sector of the track, because Write Gate is asserted after the Index pulse and remains asserted through the track till the Index pulse is encountered again after one revolution of the disk. Hence for the very first sector, Read Gate would have to be delayed to avoid the write splice. it would not affect other sectors.

An important point to note here is that if a Write Header operation is done on any sector, then the Read Gate may have to be delayed for that sector also.

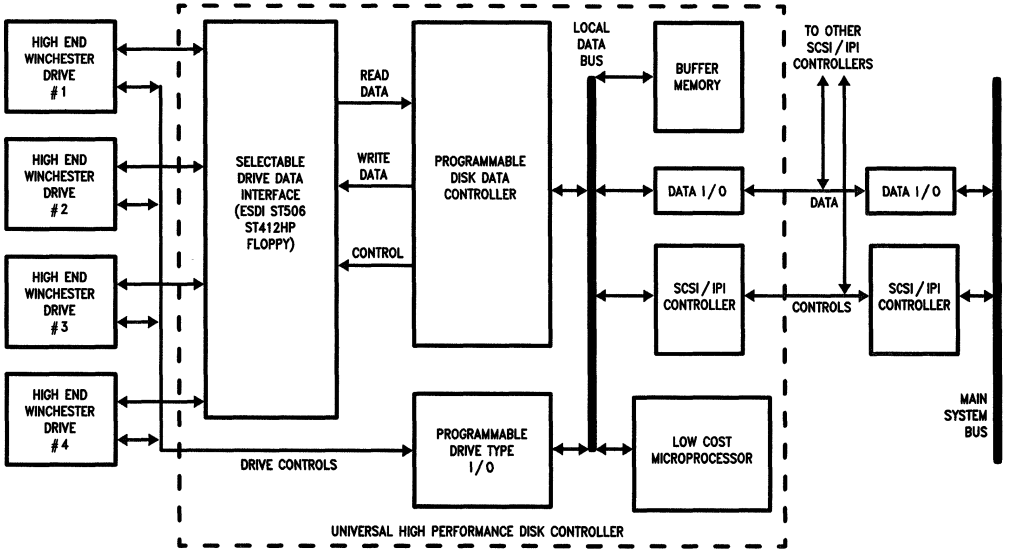
6.2.7 Intelligent Disk Interfaces

The overall objective of interfaces in this category, (like SCSI, IPI), was to make it easier for computer systems to talk to disk drives while ensuring minimum overhead for the host system. The Controller would incorporate a drive level interface on the disk side and a well defined interface to the host bus. The controller board has a local microprocessor which essentially controls the disk controller (data path), disk control signals (control path) and communication with the host system. Actually the DDC interfaces directly to the local bus. The microprocessor essentially controls the transfer of data using the DDC's DMA capability, from the local memory to the system memory, etc. *Figure 6.29* shows a block diagram of a high performance mass storage system, incorporating a SCSI peripheral bus.



TL/F/8663–A6

FIGURE 6.28. Logic to Implement Post Index/Sector Gap Field



TL/F/8663–C4

FIGURE 6.29. High Performance Mass Storage System

6.3 CIRCUIT BOARD LAYOUT AND SUPPLY ROUTING

There are several considerations to PC board design that should be followed. These guidelines serve to minimize problems that can occur with any high speed digital device. Since the DDC can operate at speeds approaching 30 MHz on the disk side, and up to 20 MHz on the system side, typical high speed design techniques should be employed to reduce noise, transmission line, and crosstalk effects. These are described below.

6.3.1 General Layout Considerations

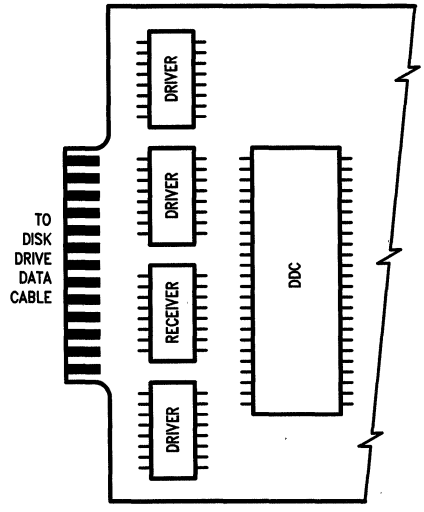
The DP8466 has two design areas of routing and loading on its signal lines, namely: disk interface and bus interface signals. For the disk signals, Read Data, Read Clock, Write Data, and Write Clock can be very high speed signals. It is recommended that when interfacing these signals to the interface line drivers/receivers, these lines be kept short as possible. Also the data cable interface devices should be located close to the data connector. It is especially important to minimize noise, propagation delay skews and jitter on these lines when in MFM mode because excessive skew, noise and jitter can lead to increased error rates. A generalized PCB chip layout is shown in *Figure 6.30*.

The bus signals generally should be treated the same as any VLSI's bus interface. The Address/Data bus provides 16 lines capable of 2 mA DC drive, and can drive variable loads up to about 100–120 pF at 20 MHz (larger loads can be driven although not at full speed). If more than 150 pF or 2 mA load needs to be driven, the data bus should be buffered, as shown in *Figure 6.31*. The one "trick" is to always connect the address de-multiplexing latches directly on the DDC, this ensures maximum address latch strobe setup time. The buffers can then drive the rest of the system. As with any high speed bus good layout practices should be followed to minimize crosstalk and reflections.

6.3.2 Decoupling, and V_{CC} and Ground Routing Guidelines

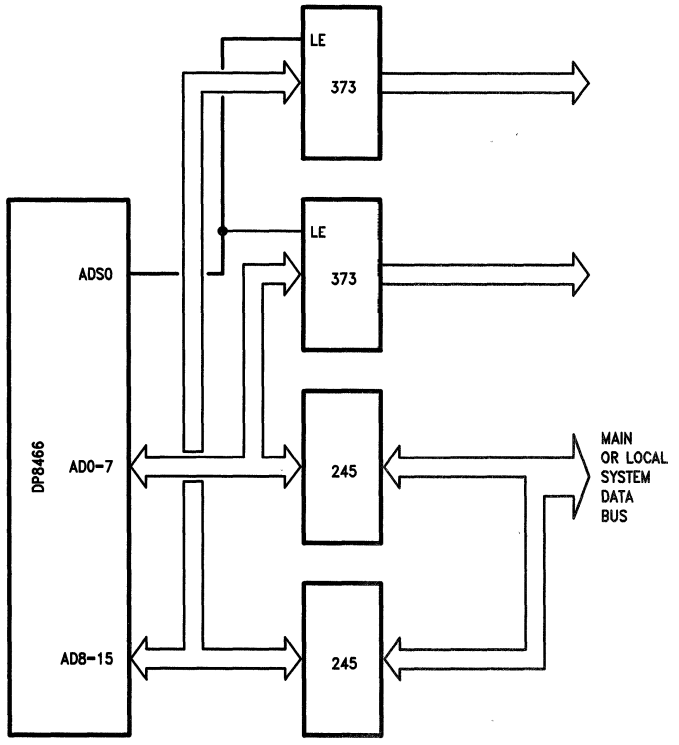
Due to the combined high speed and wide data bus of the DDC it is important to follow good power supply layout practices. Any noise generated by toggling of outputs can cause noise on V_{CC} and this in turn can reflect noise back into other inputs or outputs. The result is noise and glitches appearing on other inputs or outputs. This can be especially true when the address/data bus is toggling many lines simultaneously. In this case it is not unusual to have peak current spikes up to 300 mA being generated when toggling 16 or more outputs.

V_{CC} and ground noise problems can be easily minimized by following some simple rules when laying out the PCB. In general, multi-layer printed circuit layouts should include V_{CC} and ground planes. If a simple two-sided board, then wide V_{CC} and ground traces should be used, and an effort to layout V_{CC} and ground planes on the foil and component sides should be made. Most importantly, the DDC should have a decoupling capacitor placed as close to its V_{CC} and ground pins as possible, as shown in *Figure 6.32*. This capacitor should be a low series inductance type ceramic capacitor. Additionally, it may be desirable to add a 3–10 μF tantalum capacitor in parallel with the ceramic to offer further decoupling.



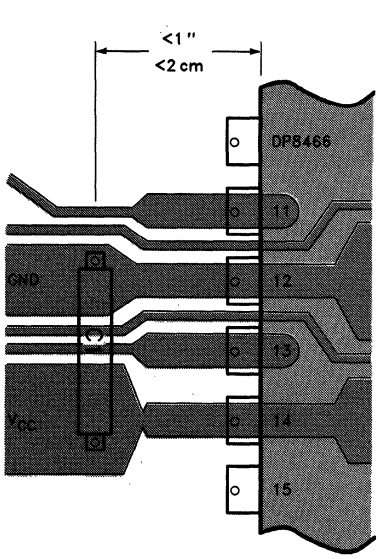
TL/F/8663–I8

FIGURE 6.30. Conceptual Component Placement to Ensure Short PCB Traces between Connector and DDC



TL/F/8663–I5

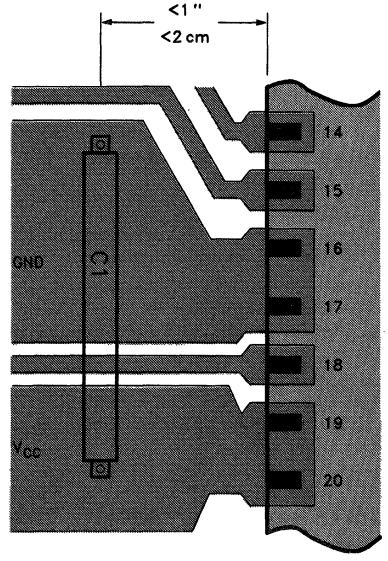
FIGURE 6.31. Proper Buffering of DDC's Address/Data Bus by Latching Address In



TL/F/8663–I6

C1 = 0.1 μF Ceramic

(a) Dual-In-Line Package



TL/F/8663–I7

C1 = 0.1 μF Ceramic

(b) Plastic Chip Carrier

FIGURE 6.32. Typical PC Board Layout for Decoupling the Power Supply (X-Ray View of Foil Side from Component Side)

Chapter 7 DDC Functional Operations

7.0 INTRODUCTION

In this chapter, all the main DDC functional operations such as Disk Formatting, DMA Data transfers, Error detection and Correction, and basic Disk Read/Write are discussed in detail from software point of view. Also, the power-up, chip initialization procedure, and interrupt servicing is described.

7.1 OPERATING MODES OF THE DDC

The DDC can be thought of as operating in one of the four modes; Reset, Command Accept, Command Perform or Error. *Figure 7.1* shows a flow chart for these modes.

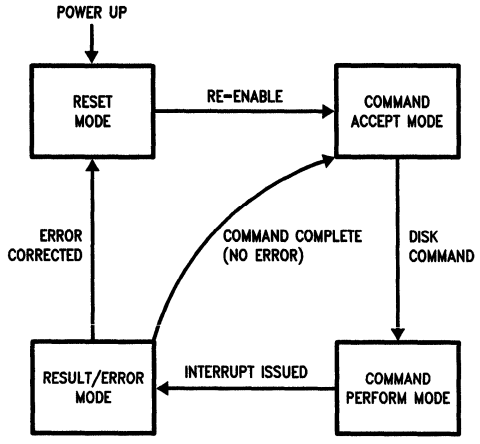


FIGURE 7.1. DDC Operating Modes

RESET MODE:

DDC is put in the reset mode after power-up or prior to starting a new operation if the previous operation was aborted. After the DDC has been reset and the DDC has been re-enabled, it moves to the Command Accept mode.

COMMAND ACCEPT MODE:

The DDC is free and ready to receive a command. Various command and pattern registers and counters may be loaded to perform a disk operation, such as format, read or write etc.

COMMAND PERFORM MODE:

In this mode, the DDC executes the disk command that was loaded into it in the Command Accept mode. It carries out DMA operations. On a successful or unsuccessful completion of the operation, the DDC will generate an interrupt (if the interrupts were enabled, EI bit in OC register), and enters the Result/Error mode.

Note: If interrupts were not enabled, then the Status register should be polled in order to find out the result.

RESULT/ERROR MODE:

In this mode, the Status and Error registers should be read to find out the result of the operation or the type of error that occurred. If the operation was completed successfully, the DDC will go back to Command Accept mode. If an error occurred during the execution of the command, the DDC will abort the operation and the Error register will indicate the type of error that occurred. To perform the operation again or to correct the error, the DDC must be reset and loaded for a particular operation, hence it goes back to the reset mode.

7.2 INITIALIZATION

After the DDC is hooked up in the system, it can be powered up and initialized to perform the desired disk operation. The chip power-up reset, operation initialization, and register programming are discussed in the following paragraphs. A flow chart shown in *Figure 7.2* describes a basic algorithm for a power-up, reset and initialization procedure.

7.2.1 Power-up and Reset

After the chip power-up, the DDC must be held reset for a duration of at least 4 BCLK and 32 RCLK periods (with these clocks active) before it could be assigned a disk format or it could be set for any disk operation. The DDC can be reset by asserting the RESET pin low or by setting the internal RES bit in the OC register high. After the DDC has been reset (for the time indicated above), The external RESET pin must be deasserted and internal RES bit in OC register must be cleared. When the system is powered, the DDC should be reset immediately, even if it will not be used right away. This is because its internal sequencers may be randomly powered on into a state that could draw some excessive I_{CC} currents.

7.2.2 Disk Operation Initialization

After a reset, the DDC must be re-enabled by setting RED bit high in the Drive Command (DC) register before other registers are loaded for a particular operation. Once the DDC is reenabled, it is ready to perform a disk operation such as read, write, or format. Various parameter and command registers and counters can then be loaded depending on the type of operation. In most of the operations, the Drive Command (DC) register is loaded the last except when the DDC is configured in a Non-Tracking DMA mode.

7.2.3 Register Programming

In order to perform an operation, related registers can be loaded in any order keeping the following restrictions in mind.

- The Drive Command (DC) register must be the last register to be loaded for any DDC operation. There is one exception to this. In non-tracking DMA mode, a remote DMA operation may be initiated by loading the operation command (OC) register after a disk command has been started.
- If the on-chip DMA is being used, or if the Remote Data Byte Count registers will be read back, the Local and Remote Transfer registers must be loaded before the Sector Byte Count and Remote Data Byte Count registers are loaded.
- The Number of Sector Operations (NSO) counter must be loaded after an external RESET or internal RESET (in OC register) are both inactive. Other registers can be loaded while reset is active.

- During the execution of an operation (format, read or write), the pattern and count registers **must not** be read. Reading these registers will interfere with the DDC's operation, and could cause some bizarre results. These registers may be written to at anytime. When data is written it takes effect immediately. Thus the only caution is to not change a pattern or count register randomly as the system may not know whether the change will apply to the current sector or the next. The Interlock mode with the Header Complete interrupt should be used to synchronize register updates.

7.3 DISK FORMATTING

The DDC can be programmed to format a disk with any type of sector format. A versatile and flexible sector format with various formatting techniques are incorporated in the DDC. Various formatting features, methods and the DDC sector format options are discussed below.

7.3.1 Key Features

MFM/NRZ Data:

The DDC can be programmed to output MFM or NRZ data while writing to the disk (bit MFM in the Disk Format register). In case of MFM data, some external circuitry will be required, as indicated in section 6.2.

Hard/Soft Sectored Drives:

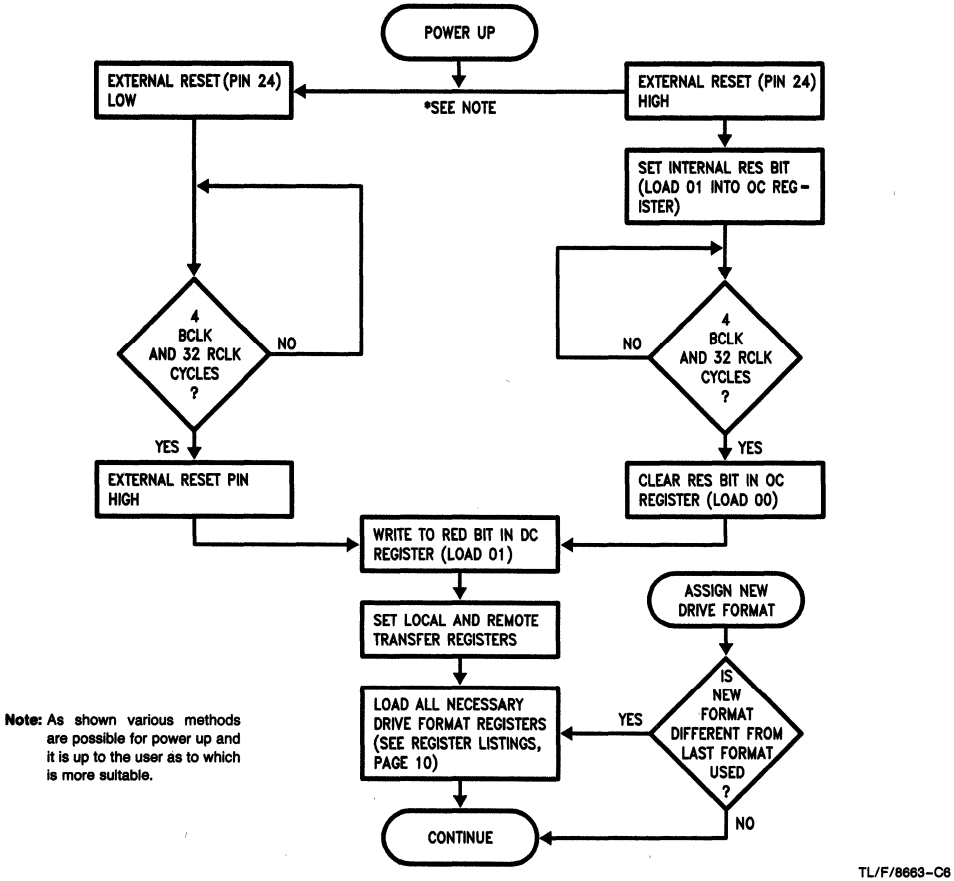
The DDC can be interfaced to both hard and soft sectored drives. This is done through HSS bit in Disk Format register. See section 6.2 and 7.3.3 for implementation of various hard and soft sector formats with the DDC.

7.3.2 Sector Format Options

The DDC offers a versatile sector format which can accommodate most of the currently used disk formats. The DDC sector format options are shown in *Figure 7.3* and the associated registers to program various format fields are shown in *Figure 7.4*. Various ID and Data fields are described below. Discussion on implementation of the popular disk formats using the DDC is given in the next section.

ID And Data Preamble:

There are two fields provided for ID and Data Preambles, which are required in hard and soft sector formats. The ID Preamble field can also be used as an Address Mark (field of no transitions) in case of formats with sector mark (ESDI or SMD type formats). Up to 31 bytes each for ID and Data preambles are allowed. These fields can be programmed using ID and Data Preamble Pattern and Byte Count registers (addresses 31H and 21H for ID, and addresses 3DH and 2DH for the Data).



TL/F/8663–C6

FIGURE 7.2. Power-up and Initialization Algorithm

ID And Data Synch #1:

This field will normally be used for writing an ID and Data Address Mark in case of soft sector drives. For hard sector drives this field may either be skipped or be used to extend the ID Preamble or ID Synch #2 fields. Up to 31 bytes can be written in this field for both ID and Data using ID Synch #1 Pattern and Byte Count registers, addresses 32H and 22H for ID, and addresses 3EH and 2EH for data.

ID And Data Synch #2:

The ID and Data Synch #2 fields, used by the DDC for byte alignment, can also have up to 31 bytes each for ID and Data. These fields can be programmed using ID and Data Synch #2 Pattern and Byte Count registers (addresses 33H and 23H for ID and addresses 3FH and 2FH for data).

HEADER BYTES: Header bytes are used to specify the header information of a particular sector such as sector number, cylinder number, track number etc. At least 2 and maximum 6 bytes can be written using 6 Header Byte Pattern and associated control registers (addresses 14H, 15H, 16H, 17H, 18H, 19H and 24H, 25H, 26H, 27H, 28H, 29H respectively). See description of Header Byte Control register in Chapter 5.

ID and Data CRC/ECC:

The DDC can be programmed for a 2 bytes of internal CRC or up to 6 bytes of internal ECC appendage using the Disk Format register, for both ID and Data fields. The CRC appendage is internal to the DDC and no pattern or count

ID FIELD

ID PREAMBLE	ID SYNCH #1 (AM)	ID SYNCH #2	HEADER BYTES	ID CRC/ECC	ID EXT ECC	ID POSTAMBLE
0–31 Bytes	0–31 Bytes	0–31 Bytes	2–6 Bytes	0, 2, 4 or 6 Bytes	0–31 Bytes	0–31 Bytes

DATA FIELD

DATA PREAMBLE	DATA SYNCH #1 (AM)	DATA SYNCH #2	DATA FORMAT PATTERN	DATA CRC/ECC	DATA EXT ECC	DATA POSTAMBLE	GAP 3
0–31 Bytes	0–31 Bytes	0–31 Bytes	1–64K Bytes	0, 2, 4 or 6 Bytes	0–31 Bytes	0–31 Bytes	0–255 Bytes

FIGURE 7.3. Sector Format Options

Pattern Register	Hex Addr	Pattern Source	Control Function	Hex Addr	Control Register
ID Preamble	31	Internal	Repeat 0-31 Bytes	21	ID Preamble Byte Count
ID Synch #1 (AM)	32			22	ID Synch #1 (AM) Byte Count
ID Synch #2	33			23	ID Synch #2 Byte Count
Header Byte 0	14		Define/Control	24	Header Byte 0 Control
Header Byte 1	15			25	Header Byte 1 Control
Header Byte 2	16			26	Header Byte 2 Control
Header Byte 3	17			27	Header Byte 3 Control
Header Byte 4	18			28	Header Byte 4 Control
Header Byte 5	19			29	Header Byte 5 Control
ID CRC/ECC	* *		16-BIT CRC/ 32-BIT / 48-BIT ECC	35	Disk Format (DF) PPB0-5 / PTB0-5
ID External ECC	*	External	0-31 Bytes	2B	ID External ECC Counter
ID Postamble	3C	Internal	Repeat 0-31 Bytes	2C	ID Postamble Byte Count
Data Preamble	3D			2D	Data Preamble Byte Count
Data Synch #1 (AM)	3E			2E	Data Synch #1 (AM) Byte Count
Data Synch #2	3F			2F	Data Synch #2 Byte Count
Data Format	3B		Field Size 1-6k Bytes	38	Sector Byte Count 0
				39	Sector Byte Count 1
Data CRC/ECC	* *		16-BIT/ 32-BIT / 48-BIT ECC	35	Disk Format
Data External ECC	*	External	0-31 Bytes	2A	Data External ECC Counter
Data Postamble	30	Internal	Repeat 0-31 Bytes	20	Data Postamble Byte Count
Gap	3A		Repeat 0-255 Bytes	34	Gap Byte Count

*These are not pattern registers.

**CRC polynomial is built into the DDC and does not require any registers. ECC requires PPB0-5 and PTB0-5 registers.

FIGURE 7.4. Pattern Registers/Counters for Sector Format Fields

register is used besides the disk format register. In case of an ECC appendage, the Polynomial Preset and Tap Byte (0–5) registers (addresses 2H–DH) must be used. Also in case of ID CRC/ECC appendage, if an internal CRC/ECC appendage is not to be used then an external ECC must be used. An external ECC is not necessary if an internal CRC/ECC was not used for a Data CRC/ECC appendage.

ID and Data External ECC:

An external ECC may be appended to encapsulate and internal CRC or ECC, for diagnostic purposes, using an external ECC circuitry. Up to 31 bytes can be appended using ID and Data Ext. Byte Count registers (addresses 2BH and 2AH).

ID and Data Postamble:

Up to 31 bytes can be used for ID and Data postamble using ID and Data Postamble Pattern and Byte Count registers, addresses 3CH and 2CH (for ID) and addresses 30H and 20H (for data).

Data Format Pattern:

Data format is programmed by the Data Pattern register (address 3B) and then can be repeated up to 64K times through the Sector Byte Count registers (addresses 38H and 39H). The number of times data format is repeated depends on the sector size.

Gap 3:

Up to 255 bytes can be written using Gap Pattern and Byte Count registers (address 3AH and 34H). In soft sectored drive operation, the Gap 3 bytes are written for each sector (determined by Gap 3 Byte Count register) except the last sector. For the last sector, gap bytes will be written until an Index pulse is received. In case of hard sectored drives, the Gap 3 Byte Count register is only used while formatting the disk. In normal disk write operation, the DDC writes gap bytes until a Sector pulse is received, ignoring the contents of the Gap byte count register.

Considerations for Pattern and Count Register Programming for Format Operations

- If any Byte Count register is loaded with zero, that field will be excluded and no pattern for the corresponding Pattern register needs to be loaded. Similarly if any of six Header Byte Control registers is set with all bits equal to zero, no pattern for that byte needs to be loaded.
- Maximum two consecutive fields can be excluded from a sector format. This includes the six header bytes which may be thought of as six fields.
- Format operations always start with an index pulse and end with the next index pulse, thus making one track. The DDC can only be programmed to format one track at a time.

7.3.3 Implementing Some Popular Sector Formats

There are three general sector formats which are commonly used in various disk systems; soft sector format (Floppy, ST506 type formats), hard sector format (hard or fixed formats used in ESDI/SMD type drives), and format with sector mark (soft or variable formats used in ESDI/SMD type drives). These three types are shown in *Figure 7.5* and their implementation using the DDC is discussed below.

SOFT SECTOR (FLOPPY/ST506 TYPE) FORMATS

The field shown in *Figure 7.5 (a)* is the most commonly used sector format used in soft sectored drives such as Floppy and ST506/412/419 type winchester drives. A double density floppy format recommended by the IBM and a Seagate's ST506/412/419 type format are shown in *Figure 7.6* as examples. It can be seen from *Figure 7.5* that the fields used in these formats are in direct correspondance with the ones supported by the DDC except the Post Index Gap in floppy format or Gap 1 in ST506 format. The Post Index Gap field is not supported by the DDC and may be eliminated as part of the disk format. If it must be generated external hardware may be added as discussed in section 6.2. The implementation of rest of the format is very straightforward and can be achieved using the registers shown in *Figure 7.4*. Implementation of the ST506/412/419 type format is shown in *Figure 7.7* and Table 7.1.

TABLE 7.1. Implementation of ST506 Formats

ST506			DDC		
Field	**Pattern**	**Byte Count**	**Field**	**Pattern Register Address**	**Byte Count Register Address**
Gap 1	4E	16	*	*	*
SYNC	00	13	ID Preamble	31H	21H
ID AM	A1, FE	2	ID Synch #1	32H	22H
CYL	#	2	Header Byte	14H–5H	24H–5H
HD	#	1	Header Byte	7H	7H
SEC	#	1	Header Byte	19H	29H
CRC	*	2	ID CRC/ECC	**	**
Gap 2	00	3	ID Postamble	3CH	2CH
Gap 2	00	13	Data Preamble	3DH	2DH
Data AM	A1, F8	2	Data Synch #1	3EH	2EH
Data	——	256	Data Format	3BH	38H, 39H
CRC	****	2	Data CRC/ECC	****	****
Gap 3	00	3	Data Postamble	30H	20H
Gap 3	4E	15	Gap	3AH	34H
Gap 4	4E	352	Gap **	3AH	34H

*Gap 1, Post Index Gap, is not supported by the DDC. See section 6.2 and the hard sector format section in this chapter.
**The CRC is internal to the DDC and Disk Format register is used to select it.
***The DDC only allows 256 bytes for the Gap field. During a format the gap for the last sector of the track will be written until the pulse is received.

TABLE 7.2. Implementation of ESDI Hard Sector Format

ESDI FORMAT			DDC REGISTERS		
Field	Pattern	Byte Count	Field	Pattern Register Address	Count Register Address
	(See Note†)				
Inter-Sector Gap	00	‡	*	*	*
Address PLO Sync	00	‡	ID Preamble	31H	21H
(No Field)		0	ID Sync #1	32H	22H
Byte Sync Pattern	—	≥ 1	ID Sync #2	33H	23H
Cylinder	xxxx	2	Header #1/2	14/15H	24/25H
Head	xx	1	Header #3	16H	26H
Sector	xx	1	Header #4	17H	27H
Flag/Status	xx	1	Header #5	18H	28H
(No Field)		0	Header #6	19H	29H
Address Check Bytes	**	**	ID CRC/ECC	**	**
Address Pad	00	—	ID Postamble	3CH	2CH
Write Splice	00	1	***	***	***
Data PLO Sync	00	—	Data Preamble	3DH	2DH
(No Field)		0	Data Sync #1	3EH	2FH
Byte Sync Pattern	—	≥ 1	Data Sync #2	3FH	2FH
Data	xx	—	Data Pattern@	3BH	38/39H
Data Check Bytes	**	**	Data CRC/ECC	**	**
Data Pad	00	≥ 2	Data Postamble	30H	20H
Format Tol.	00	—	Gap	3AH	34H
Inter Sector Gap	00	‡	Gap	*	*

†Where dashed entries appear in these columns the ESDI standard does not specifically define these fields but leaves this up to the user.

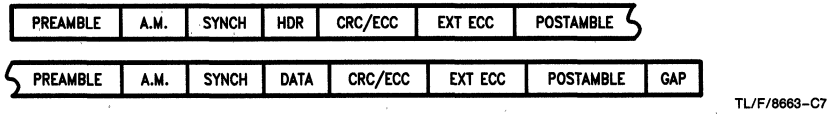
‡Defined by the ESDI specification using a specific formula.

*The standard Inter Sector Gap field is not supported by the DDC. See Hard sector section in this chapter, and section 6.2 for details in generating this field. Chapter 6's hardware actually uses the DDC's gap field to generate the ISG field for the next sector.

**The 16 bit CRC, 32 bit or 48 bit ECC is programmed internally see section 7.6 for details.

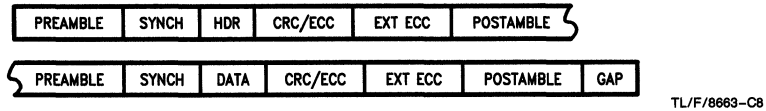
***The write splice field is not supported by the DDC. It should be included as part of the Data preamble field for format and write operations, and part of the ID postamble for read operations.

@Used only to format the data field.



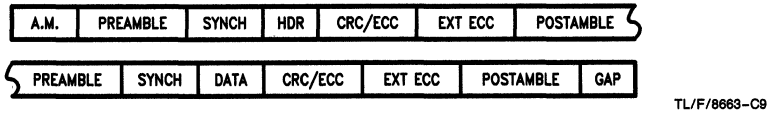
TL/F/8663–C7

(a) Soft Sector Format



TL/F/8663–C8

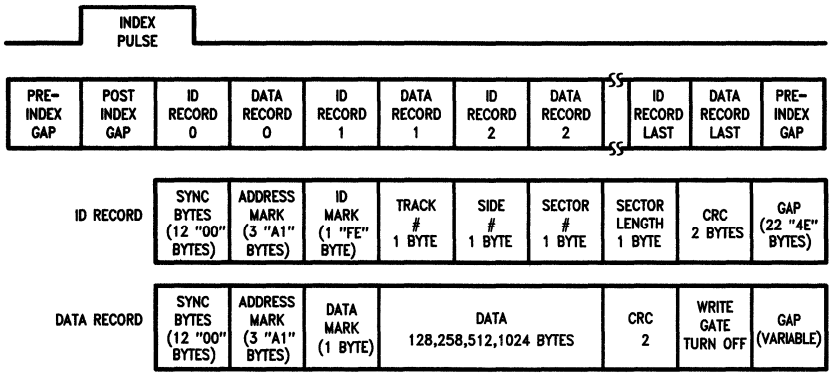
(b) Hard Sector Format



TL/F/8663–C9

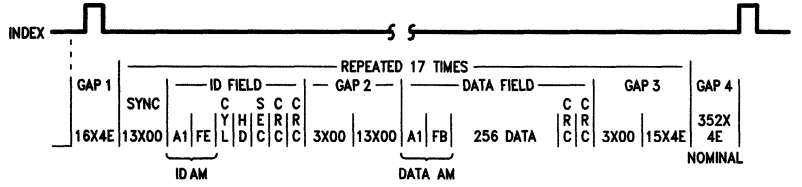
(c) Format with Sector Mark

FIGURE 7.5. Generalized Common Sector Formats



TL/F/8663–D0

(a) Double Density Floppy Format



TL/F/8663–D1

(b) ST506/412/419-type Sector Format Recommended by Seagate

FIGURE 7.6. Soft Sector Formats

HARD SECTOR FORMATS (Setting up Pattern and Count registers for Read/Write and Formatting)

The second sector format shown in *Figure 7.5* is the general hard sector format. In this format the beginning of each sector is marked by a sector pulse. This format is typically used in ESDI and SMD type drives. The format for these types of drives is fairly straight forward, and Table 7.3 shows the DDC registers, patterns and count lengths to perform and ESDI format. Table 7.4 shows Control Data Corp. recommended format for an SMD specification.

In both formats the Inter Sector Gap of ESDI and the Head Scatter Gap, commonly referred to as the Post Index or Post Sector Gap in SMD are not directly supported by the DDC. Additional counters and logic as shown in Section 6.2 previously are needed if these are to be supported. This solution would be appropriate for ESDI or SMD hard sectored drives. The philosophy behind it being that the index/sector pulse from the drive is presented to the DDC delayed by the external logic by the length of the post index/sector gap.

Another field in both SMD and ESDI that is not directly supported is the write splice field. This field is intended for turning off and on the read/write head without interfering with the data preamble. This field is easily generated by including it in the data preamble for formatting. Then during read operations including it in the ID postamble, and for write operations including it in the Data preamble.

For ESDI drives during the format operation the standard calls out for optionally deasserting Write Gate for two bit times during the write splice. This option is useful if the drive is performing some data encoding (such as 2–7). Pulsing Write Gate for two bit times informs the encoder/decoder on the drive to start the data preamble field. The DDC does not directly support this deassertion of Write Gate, but can format the drive easily anyway, by performing a two pass format operation. For the first pass the DDC will be set up to format the whole track, however only the ID fields will actually be formatted due to the encoder. The second pass should be a multi-sector write data operation, which will format the data fields.

FORMATS WITH SECTOR MARKS (Setting up Pattern and Count registers for Read/Write and Formatting)

The third method of formatting a disk is with sector marks. A simplified typical format is shown in *Figure 7.5(c)*. This format is most common with soft sectored ESDI, and a few SMD drives. Basically this format uses a field of no flux transitions and some special drive hardware to enable the drive to generate a sector pulse-like signal called Address Mark. The Address Mark signal thus signifies the start of a new sector.

To implement this using the DDC, requires some manipulation of the format parameter RAM in conjunction with the use of the ESDI control PAL hardware described in section 6.2. Other external hardware designs may require altering the format parameters, however, much of the discussion is still applicable. While formatting, the ID preamble field of the DDC is set with the pattern of the Post index gap. The count length of this field is set to the length of the gap plus length of address mark (usually 3 bytes). The ID synch #1 field contains the pattern of the preamble while its length is increased by one to accommodate the Address Mark pad field. This ID Sync #2 contains the byte synch pattern.

The SAM bit in the Disk Format register is set. Hence when formatting is initiated, the DDC generates AME and the start of its preamble. This is taken by the PAL and external hardware, and delayed by the length of the gap (which is being written), to when the address mark (3 bytes of no transitions) is written. This is then followed normally by the preamble and synch fields.

TABLE 7.3. Implementation of SMD Hard Sector Format

SMD FORMAT			DDC REGISTERS		
Field	**Pattern**	**Byte Count**	**Field**	**Pattern Register Address**	**Count Register Address**
Head Scatter Bytes	00	16	*	*	*
Address PLO Sync	00	11	ID Preamble	31H	21H
(No Field)		0	ID Sync #1	32H	22H
Byte Sync Pattern	—	1	ID Sync #2	33H	23H
Flag/Status	—	1	Header #5	14H	24H
Cylinder	—	2	Header #2/3	15/16H	25/26H
Head	—	1	Header #3	17H	27H
Sector	—	1	Header #4	18H	28H
(No Field)		0	Header #6	19H	29H
Address Check Bytes	CRC/ECC	2–6	ID CRC/ECC	**	**
(No Field)		0	ID Postamble	3CH	2CH
Write Splice	00	1	***	***	***
Data PLO Sync	00	11	Data Preamble	3DH	2DH
(No Field)		0	Data Sync #1	3EH	2FH
Byte Sync Pattern	—	1	Data Sync #2	3FH	2FH
Data	xx	—	Data Pattern@	3BH	38/39H
Data Check Bytes	CRC/ECC	2–6	Data CRC/ECC	**	**
EOR Pad	—	1	Data Postamble	30H	20H
End of Sector	—	10	Gap	3AH	34H

†Where dashed entries appear in these columns the SMD standard does not specifically define these fields but leaves this up to the user.

‡The Data Pattern is used during format operations only.

*The standard Inter Sector Gap (Head Scatter Bytes) field is not supported by the DDC. See Hard sector section in this chapter, and section 6.2 for details in generating this field. The hardware in chapter 6 uses the DDC's gap filed to generate the Head Scatter Bytes for the next sector.

**The 16 bit CRC, 32 bit or 48 bit ECC is programmed internally see section 7.6 for details.

***The write splice field is not supported by the DDC. It should be included as part of the Data Preamble field for format and write operations, and part of the ID postamble for read operations.

@Used only to format the data field.

When reading from the disk the DDC looks on the drive like a hard sectored one. The Address Mark is detected, and the drive asserts AMF which is decoded by the PAL into a sector pulse for the DDC. Hence the DDC format parameter RAM should be changed to the ID preamble field containing the pattern of the actual preamble and count length. The ID synch #1 field count is set to zero so that this field is skipped and the ID synch #2 contains the byte synch. (The read gate is delayed by 8 bits to accommodate the 1 byte write splice field). This ensures a successful read operation. With the DDC working in this mode it would see an index and a sector pulse from the sector 0 AMF separated by the Post Index Gap length. The DDC takes the index to be the sector 0 pulse also. The ESDI control PAL of chapter 6 takes care of delaying the index pulse over the sector 0 AMF pulse and suppresses the sector pulse to the DDC from the sector 0 AMF.

If a Write Header operation is desired at any time then it would be possible to do so only if a regular format operation had been done earlier. The AMF from the drive generates a sector pulse to the DDC to start writing the header. Also the Read Gate assertion is delayed externally by a byte to accommodate the Write Splice associated with Write Gate assertion. The ESDI field names byte values and lengths along with the DDC's equivalent registers are shown in Table 7.5.

The DDC does not directly support the ESDI write splice field as before, however, this is easily remedied by including this byte as part of the postamble during formats and write operations as part of the data preamble during read operations. This ensures that reading of the write splice is avoided which is the purpose of this field.

As in the hard sector format, ESDI has an optional deassertion of the Write Gate in between the ID and Data fields. The DDC does not support this option, but by performing a two pass format as described above a complete format can be accomplished.

MODIFICATIONS TO SECTOR FORMATS

While the previously discussed "standard" formats are a useful starting point, they are generally not strictly adhered to when actually formatting a disk. This is due usually to the users desire to optimize the drive for various parameters. These include optimizing data integrity, data separator per-

TABLE 7.4. Implementation of ESDI Sector Mark Format

ESDI FORMAT			DDC REGISTERS		
Field	Pattern (See Note†)	Byte Count	Field	Pattern Register Address	Count Register Address
Inter-Sector Gap	00	‡	*	*	*
Pad	00	1	ID Preamble	31H	21H
Address PLO Sync	00	—			
Address Mark	†††	3	ID Sync #1	32H	22H
Byte Sync Pattern	—	≥ 1	ID Sync #2	33H	23H
Cylinder	xxxx	2	Header #1/2	14/15H	24/25H
Head	xx	1	Header #3	16H	26H
Sector	xx	1	Header #4	17H	27H
Flag/Status	xx	1	Header #5	18H	28H
(No Field)		0	Header #6	19H	29H
Address Check Bytes	**	**	ID CRC/ECC	**	**
Address Pad	00	≥ 2	ID Postamble	3CH	2CH
Write Splice	00	1	***	***	***
Data PLO Sync	00	—	Data Preamble	3DH	2DH
(No Field)		0	Data Sync #1	3EH	2FH
Byte Sync Pattern	—	≥ 1	Data Sync #2	3FH	2FH
Data	xx	—	Data Pattern@	3BH	38/39H
Data Check Bytes	**	**	Data CRC/ECC	**	**
Format Tol.	00	‡	Gap	*	*
ISG	00	‡	Gap	*	*

†Where dashed entries appear in these columns the ESDI standard does not specifically define these fields but leaves this up to the user.

‡Defined by the ESDI specification using a specific formula.

†††For the Address mark the pattern does not matter, as an area of no flux transitions is recorded.

@The Data Pattern register is used only for format operations.

*The standard Inter Sector Gap field is not supported by the DDC. See ESDI control PAL hardware solution in chapter 6.2 for details in generating this field.

**The 16 bit CRC, 32 bit or 48 bit ECC is programmed internally see section 7.6 for details.

***The write splice field is not supported by the DDC. It should be included as part of the data preamble field for format and write operations, and part of the ID postamble for read operations.

formance, access speed, sector defect sparing algorithms, and total storage capacity. Some of these considerations are discussed below.

Error Detection/Correction And Data Field Length

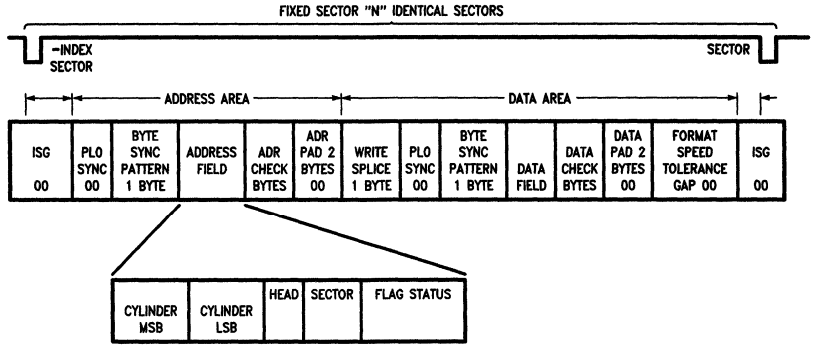
Generally, ST506 type drive recommended formats utilize a 16 bit CRC, however, this generally does not offer the type of data integrity that is needed in the data field. Thus generally ST506 type drives should utilize a 32 ECC for data. The ID field usually can be a CRC check field since the header is so short, and since recovery of the header address can be achieved by reading the previous sector header.

In general the type of ECC field to choose for the data field depends on the length of the data field, the defects on the disk media itself and the tolerance the designer places on his system for detecting errors, and the probability limits for miscorrection.

The choice for the length of the data field is determined in part by the type of ECC chosen (or vice versa), but also by system performance criteria. For a given media, a longer data field, and fewer sectors per track can maximize total storage capacity, but it requires better ECC. In many cases where the disk is to store many small files that leave many partially filled sectors, a larger sector size will waste disk space. However, if a few very large files are stored when large data field maximize storage. Thus usually a good tradeoff is to have sector sizes between 256 to 1024 bytes.

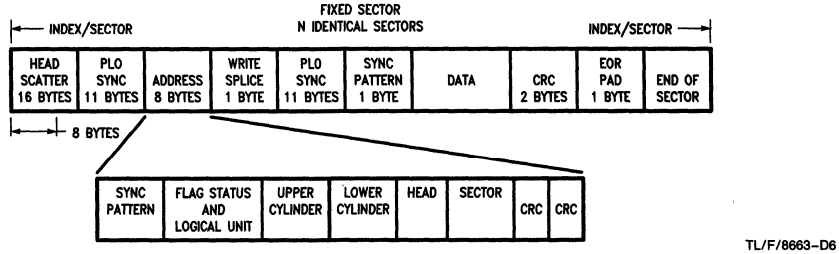
ID And Data Preamble

These field lengths are used to lock the data separator to incoming data during a read operation. The lengths of these fields determine the lock time requirements of the data separator. For individual ESDI and SMD drives, where the data separator is on the drive, the preamble should follow the manufacturers recommendations. For ST506 drives, and drives with imbedded controllers, the data separator is part of the controller and the preamble length should be set based on the separators performance goals.



TL/F/8663–D5

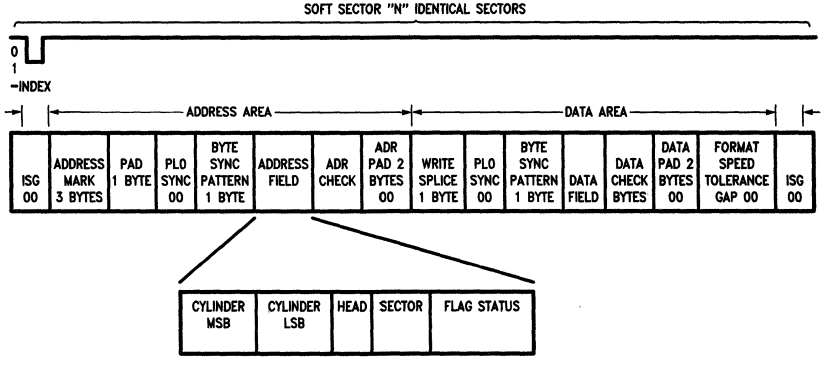
(a) An EDSI Hard Sector Format Recommended by Maxtor



TL/F/8663–D6

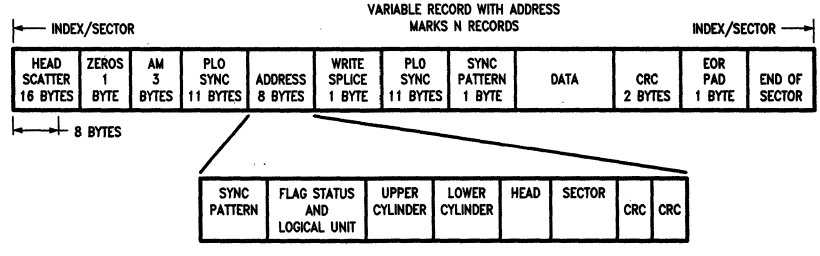
(b) An SMD Hard Sector Format Recommended by CDC

FIGURE 7.7 Hard Sector Formats



TL/F/8663–E0

(a) An ESDI Soft Sector Format Recommended by the Maxtor



TL/F/8663–E1

(b) An SMD Soft Sector Format Recommended by the CDC

FIGURE 7.8. Formats with Sector Mark

7.3.4 Formatting Methods

The disk formatting can be carried out using one of the three methods supported by the DDC; Internal Sequential, FIFO Table and Interlock Mode. These three disk formatting methods are explained in the following paragraphs and also summarized in a flow chart in *Figure 7.9.*

1. INTERNAL SEQUENTIAL METHOD

This method of disk formatting is adopted when sectors are to be physically contiguous. The DDC can be set for a multi-sector operation to format a whole track of sequential sectors. The steps required to perform the Internal Sequential method are explained in the following paragraphs. See *Figure 7.9.*

The DDC in Command Accept Mode

Step 1. All the Pattern and Byte Count registers for various sector format fields are loaded. All the Header byte's Pattern and Control registers (such as the one for cylinder number, head number etc.) are loaded except the one containing sector number information. The Pattern register for this header byte need not be loaded with anything, but the Control register should be loaded with 3H (i.e. SSC = 1 and HB = 1). With SSC = 1, the sector number for each sector will be loaded into this Header Byte Pattern register, automatically, from the Sector Counter.

Step 2. The Sector Counter is loaded with the first sector to be formatted. The contents of Sector Counter are loaded into the Header Byte Pattern register reserved for sector number, written to the disk, and then incremented for the next sector. The Number of Sector Operation (NSO) Counter is loaded with the number of sectors per track.

Step 3. The Disk Format register is loaded with FTF = 0, desired internal CRC/ECC appendage and other information (such as Hard/soft sector, NRZ/MFM data, etc.). The ECC/CRC control register (address 0EH) should also be loaded for desired options, such as inverting the serial data. The ECC polynomial and tap registers should be programmed if ECC is chosen.

Step 4. The Operation Command register is loaded to enable interrupts. Finally the Drive Command register is loaded with the Format Track command (i.e. ACH). Refer to Table 5.6 for various DDC commands.

The DDC in Command Perform Mode

Step 5. The DDC will start the operation when it receives an index pulse indicating the start of a track and will end the operation when it encounters another index pulse.

The DDC in Result/Error Mode

Step 6. An interrupt will be generated after a successful or unsuccessful execution of Format Track command, if the interrupts were enabled. The Status and (or) Error registers will indicate the result or the type of error occurred. In case of successful completion of formatting, the DDC could be initialized to format the next track and steps 1 thru 5 will be repeated. If an error has occurred, the interrupt should be serviced properly and the DDC should be reset. Refer to section 7.7 for interrupt servicing, and section 7.2.2 for resetting.

2. FIFO TABLE METHOD

This method is ideal if sector interleaving is required. The sectors may be written to the disk in any order using this method which offers the minimum amount of microprocessor involvement during the format operation. In this method the header bytes are written on the disk from the memory (via FIFO) instead of the Header Byte registers. This essentially eliminates the need of the microprocessor to update header bytes for each sector as could be done in the interlock mode. All other format pattern and count registers are loaded once by the microprocessor remain valid for the entire operation. The header bytes for each sector (with or without interleaving) are set up contiguously in sets as a table in the memory and then read by the DDC, one set for each sector, from the memory using the local DMA channel.

The steps required to perform the FIFO table method are explained below. Also see *Figure 7.9.*

Step 1. Sets of header bytes (one for each sector) for the entire track are stored contiguously in a memory area accessible to the local DMA. Each header byte set must contain an even number of bytes and start on an even byte boundary. If the header byte set contains an odd number of bytes, an extra dummy byte must be inserted at the end of each set so that each header byte set will start on an even byte boundry, for DMA considerations.

The DDC in Command Accept Mode

Step 2. Address of the first byte of the first header byte set is loaded in the DMA Address Byte (0, 1) registers.

Step 3. Sector Counter (SC), Number of Sector Operations (NSO) counter and Header Byte Count registers are loaded with initial sector number, number of sectors to be operated on, and number of header bytes (2–6 bytes), respectively.

Step 4. All other Format patterns and count registers (such as Preamble, Postamble, ECC/CRC etc.) according to the selected sector format are loaded with appropriate information. See *Figure 7.4.*

Step 5. The Desk Format (DF) register is loaded with the FTF bit set. The DF register is also loaded with MFM/NRZ, hard sector/soft sector, and ID, Data CRC/ECC appendage etc.

Step 6. The Operation Command (OC) register is loaded to enable interrupts. Finally, the Drive Command register is loaded with Format track command (ACH). See Table 5.6 for the DDC commands.

The DDC in Command Perform Mode

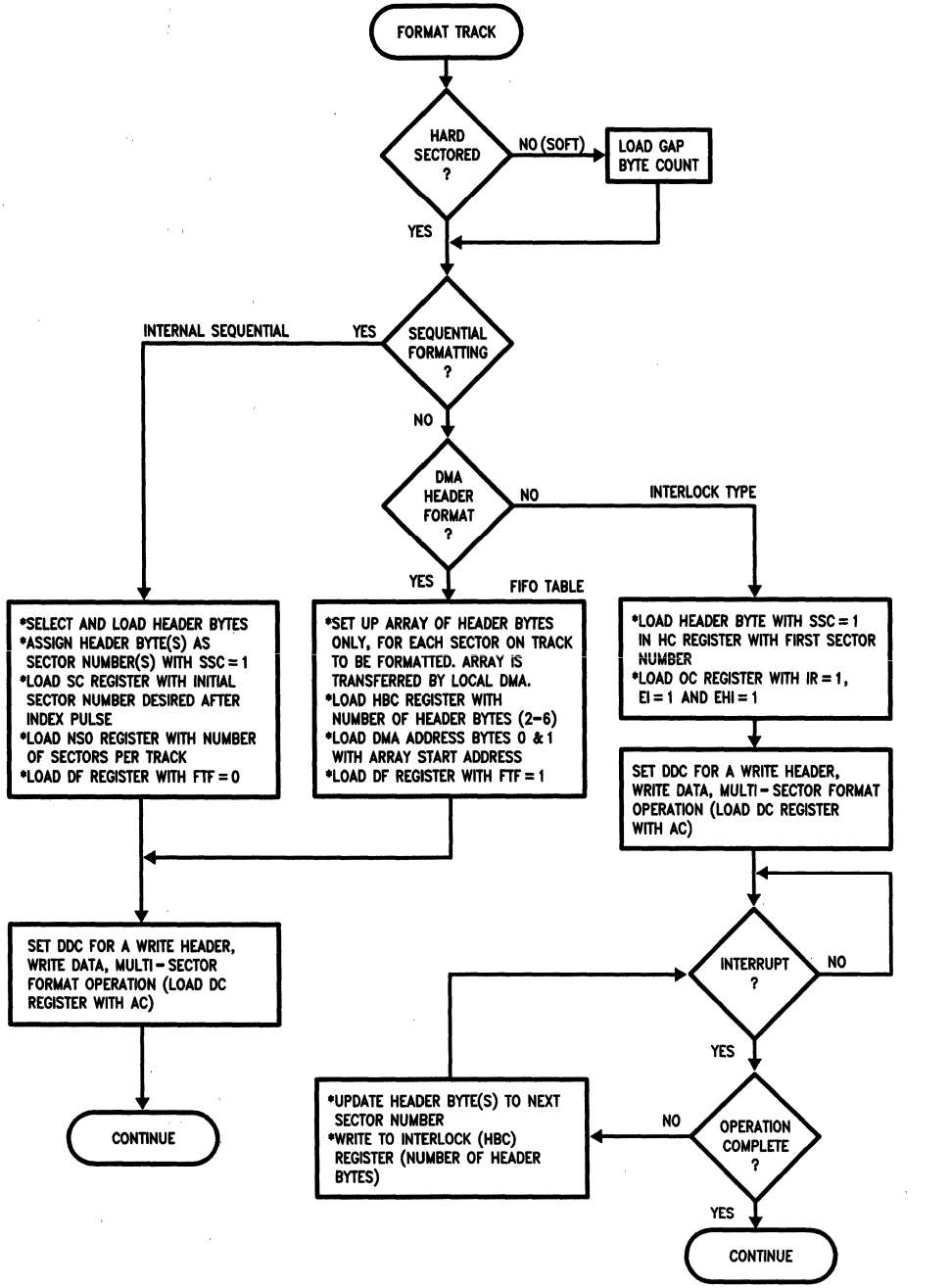
Step 7. The DDC starts the operation when it receives the index pulse and ends it on the occurrence of next index pulse. As the header bytes are needed they are DMA'd from memory. After a successful or unsuccessful completion of the operation, the DDC will generate an interrupt.

The DDC in Result/Error Mode

Step 8. The Status and (or) Error registers are read to find out the cause of interrupt. In case of a successful completion of the operation, steps 1 thru 7 may be repeated or the DDC may be initialized for another disk operation. In case of an error interrupt, the interrupt is serviced properly. See section 7.2.2 and 7.7.

3. INTERLOCK METHOD

This method is the most versatile of the three disk formatting methods. But it requires fast microprocessor involvement. This method may be used to format a whole track of interleaved sectors or a single sector. Using this method to reformat a single sector is ideal. Formatting single sectors is useful for remapping bad sectors. This method can also be used for creating tracks with varying sector field lengths.



TL/F/8663-E6

FIGURE 7.9. Track Formatting Methods

The Interlock type disk formatting uses the interlock mode and header complete interrupt to enable the microprocessor to directly update any format parameter bytes. In a write header-write data operation, after the header bytes are written to the disk, the DDC issues the header complete interrupt. With interlock mode set and header complete interrupt issued, the controlling microprocessor has the time (until the preamble field of the next sector) to read status, load the next sector's header bytes, and write the interlock (HBC) register. Writing to the interlock (HBC) register confirms that microprocessor has updated all the required parameters. This must be done following each header match complete interrupt for every sector, including the last sector of the operation. If this is not done, the DDC assumes that the microprocessor did not properly update the parameter registers in time for the next sector and therefore generates an interrupt indicating the Late Interlock error when a subsequent command is loaded in the DC register. A 'step by step' procedure of the Interlock type formatting is explained below. Also see *Figure 7.9*.

DDC in Command Accept Mode

Step 1. All the format registers (shown in *Figure 7.4*) are loaded with respective pattern and count values. The header byte control register for the sector number is loaded with SSC = 1. This is done only for the first sector. Later on, depending upon the interleave factor, the header byte reserved byte for sector number may be loaded with a new value and the associated control register with SSC=0.

Step 2. The Sector Counter is loaded with the sector number to start with and NSO (number or sector operations) Counter with number of sectors to be formatted.

Step 3. In Disk Format register, FTF is set to zero and other relevent bits such as HSS, MFM, IH's and ID's are also set or reset.

Step 4. The DDC is set to be in Interlock Mode by setting IR = 1 in the Operation Command register. Also header complete and other interrupts are enabled by setting EHI = 1 and EI = 1 in the same register.

Step 5. Finally, the Drive Command register is loaded with the Format Track command, ACH. See Table 5.6 for various DDC commands.

DDC in Command Perform Mode

Step 6. The format operation starts when the DDC receives an index pulse. An interrupt is expected at the completion of the Write Header operation. When this interrupt is received it is tested for Command Complete Error or Header Complete Status Bits set. If it's a Header Complete, then parameter, count and control registers are updated. The Interlock register is written to. This is repeated until the Command Complete or Error Interrupt occurs.

DDC in Result/Error Mode

Step 7. On the occurrence of an interrupt, the Status register is read. If the interrupt was an operation complete interrupt then steps 1 through 5 are repeated for the rest of the sectors to be formatted without changing the contents of the NSO counter. In case of an error, the interrupt is serviced properly. See section 7.7 on interrupts.

7.4 READ AND WRITE OPERATIONS

Once the disk has been formatted, various disk read and write operations can be performed. These commands are listed in Table 7.5 and are discussed briefly in section 5.2.7.

Generally, the read operation is taken as reading data from the disk and can therefore be performed by executing the Read Sector (single or multi-sector) and Read Track commands. Similarly, write operation is considered as writing data to the disk and hence could be achieved by executing the Write Sector (single or multi-sector) and Write Track commands. Other read and write commands imply reading or writing ID with or without data from (to) the disk. The DDC programming procedure for all the read and write commands basically is the same.

A general register programming procedure to perform read and write operations is given below.

Figure 7.10 shows a generalized flow chart for performing a read operation. To generalize both multi-sector and single sector operations, the continue block would go to the next sector operation if multi-sector, and would go to other μP tasks if single sector. Refer also to *Table 7.5* which shows the command codes for the various operations.

Command Name		Op Code	
Read Single Sector	RDSS	11010010	D2H
Read Sector ID	RDID	01110010	72H
Read Multi-Sector Logical	RDMS	11010110	D6H
Read Track	RDTK	11010100	D4H*
Read Track Blind	RDTB	11000100	C4H*
Read ID Multi-Sector	RDIM	01110100	74H
Read Track Data/ID	RDDI	11110100	F4H*
Write Single Sector	WRSS	10010010	92H
Write Multi-Sector Logical	WRMS	10010110	96H
Write Track	WRTK	10000100	84H*
Format Track	FMTK	10101100	ACH
Format Track No Gap	FMNG	10100100	A4H
Find ID	FNID	01010010	52H
Find ID Multi-Sector	FNMS	01010110	54H
Recover Header	RCID	01100010	62H
Re-Enable Controller	RENB	00000001	01H
No Operation	NOP	00000000	00H

***Note:** For an entire track operation, the Number of Sector Operations Counter should be set to the number of sectors per track.

Table 7.5. Common Configurations of the Command Bits

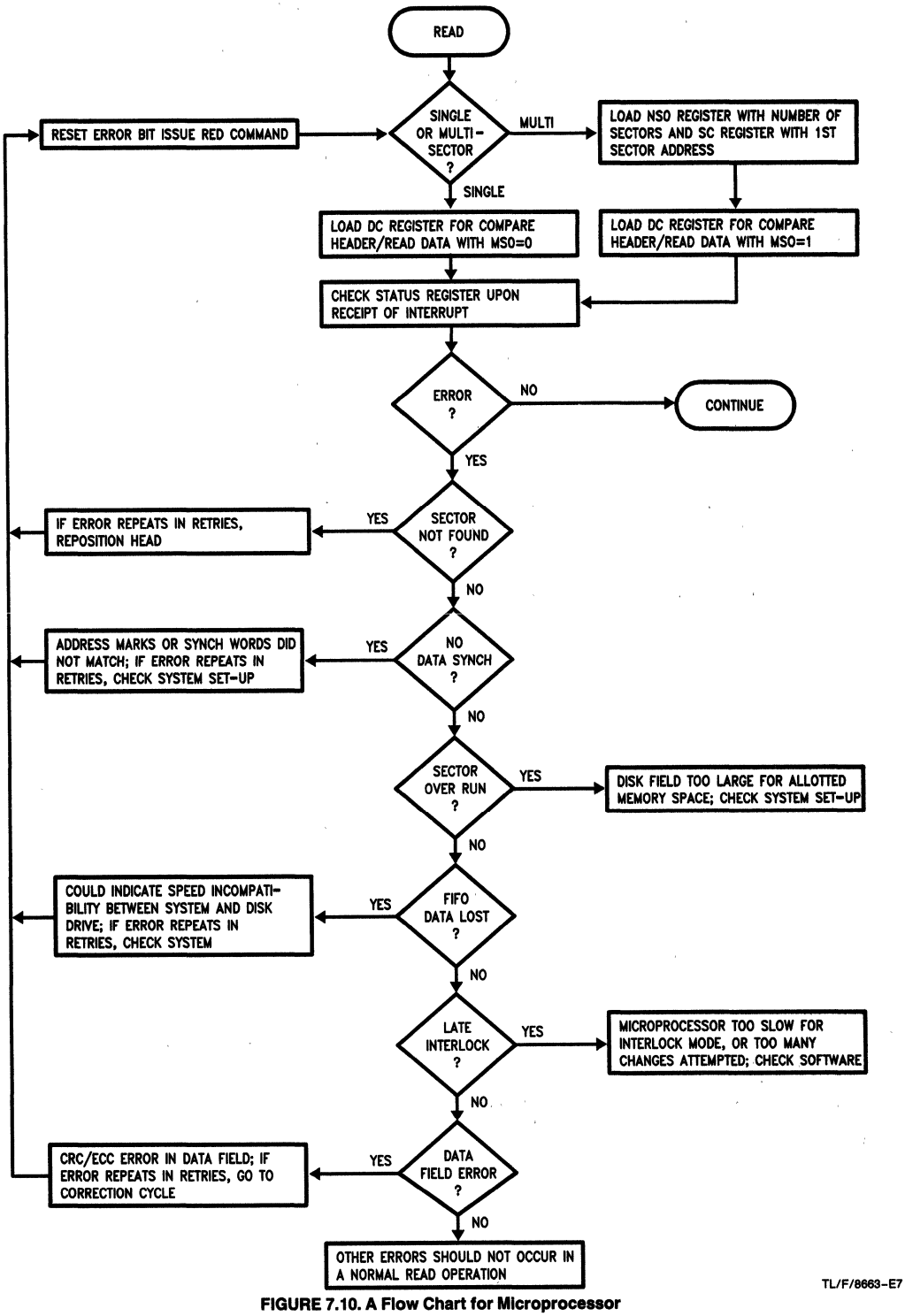
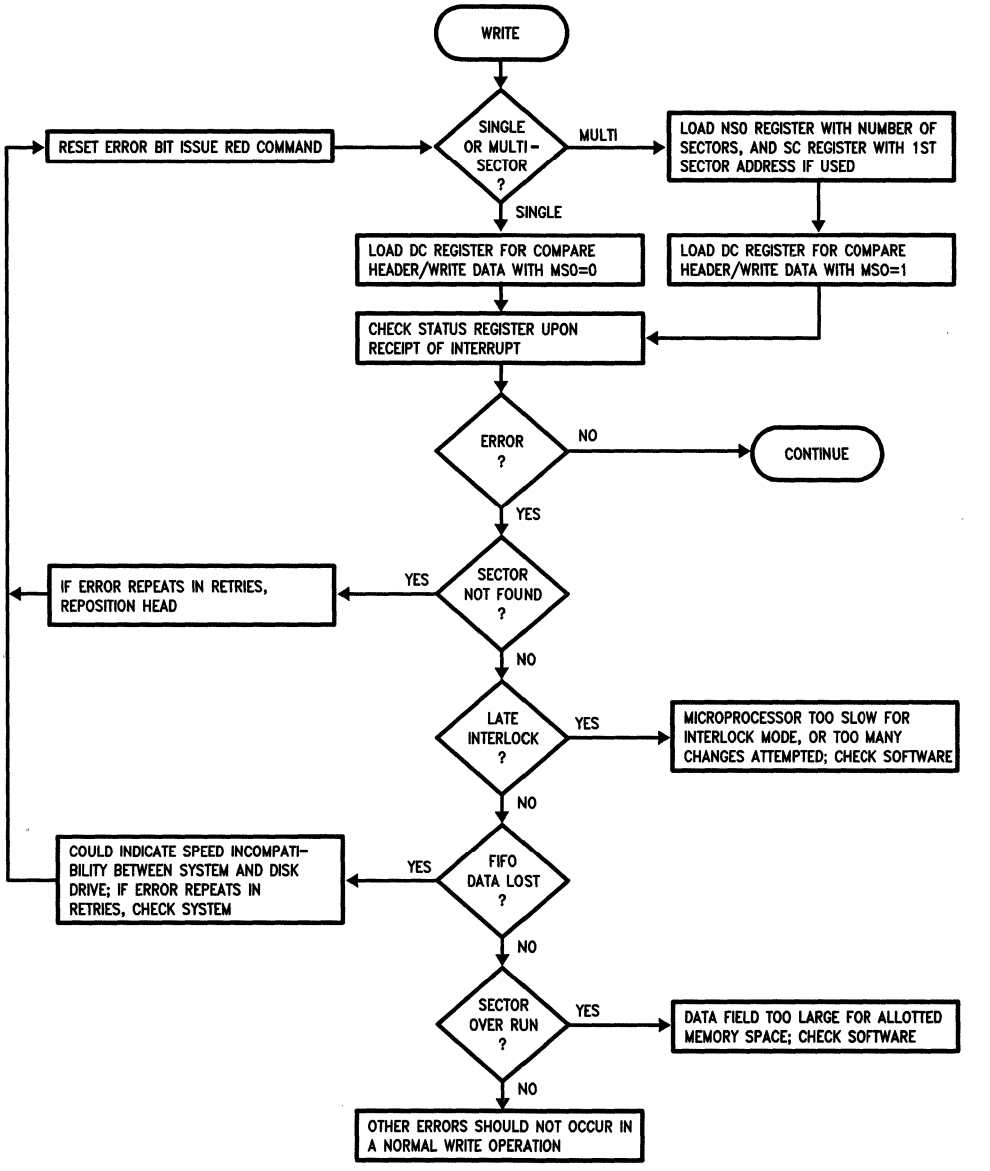


FIGURE 7.10. A Flow Chart for Microprocessor Initiation and Servicing of a Read Operation



TL/F/8663–E8

FIGURE 7.11. A Flow Chart for Microprocessor Initiation and Servicing of a Write Operation

SINGLE SECTOR READ AND WRITE OPERATIONS

This operation involves setting the Disk Command register to perform a Compare Head/Read Data Operation. Only one sector is transferred.

DDC in Command Accept Mode

Step 1. The Parameter and Count registers must be loaded with the proper format information (if not already loaded with the correct information). The Header byte pattern registers must be loaded with the exact header information for the sector to be transferred.

Step 2. The Local DMA channel must be initialized. The Local transfer register should be configured to the desired DMA mode. The DMA address registers are loaded with the data transfer address. If the Remote DMA is used in coordination with the local transfer, especially in tracking mode, the Remote Transfer register should be initialized, as should its address, and length information. Interrupts should be enabled, in order to enable determination of operation completion.

Step 3. Finally the Drive Command register is loaded with the desired Read or Write Command. If a remote DMA operation is to be performed, the Operation Command register should be loaded to start the remote DMA.

DDC in Command Perform Mode

Step 4. The DDC will perform the operation by looking for the correct sector header, then acquiring the system bus, and transferring the appropriate data.

DDC in Result/Error Mode

Step 5. At the end of the command an interrupt is generated. If the interrupt is the Operation Complete interrupt the command terminated properly, and the μP can read this from the Status and Error registers and can proceed to the next command. If an error occurred the microprocessor should service this by either retrying the operation, or correcting the error condition.

MULTI SECTOR LOGICAL READ AND WRITE OPERATIONS

Multi sector logical operations transfer sectors sequentially based on their sector number, whether these sectors are scattered around the disk or not. Multi sector operations use the Number of Sector Operations Counter to determine the number of sectors to transfer. Logical operations are most typically done by using the Sector Counter.

DDC in Command Accept Mode

Step 1. The Parameter and Count registers must be loaded with the proper format information (if not already loaded with the correct information). The Header byte pattern registers must be loaded with the exact header information for the sector to be transferred. The header byte that contains the sector number must have the Sector Counter substituted for it. The Sector Counter is set to the number of the first sector to be transferred. The Number of Sector Operations Counter should be loaded with the number of sectors to be transferred.

Step 2. The Local DMA channel must be initialized. The Local transfer register should be configured to the desired DMA mode. The DMA address registers are loaded with the data transfer address. If the Remote DMA is used in coordination with the local transfer, especially in tracking mode, the Remote Transfer register should be initialized, as should its address, and length information. Interrupts should be enabled, in order to enable determination of operation completion.

Step 3. Finally the Drive Command register is loaded with the desired Logical Read or Write Command. If a remote DMA operation is to be performed, the Operation Command register should be loaded to start the remote DMA.

DDC in Command Perform Mode

Step 4. The DDC will perform the operation locating the first sector, acquiring the system bus, and transferring that sector's data. The Number of Sector Operations Counter is decremented, and the Sector Counter is incremented. Step 4 is repeated until the Number of Sector Operations Counter reaches zero. The command may be terminated early if the DDC could not find one of the correct sectors being sought.

DDC in Result/Error Mode

Step 5. At the end of the command when the NSO counter equals zero, an interrupt is generated. If the interrupt is the Operation Complete interrupt the command terminated properly, and the μP can read this from the Status and Error registers and can proceed to the next command. If an error occurred the microprocessor should service this by either retrying the operation, or correcting it.

MULTI SECTOR PHYSICAL READ AND WRITE OPERATIONS

These operations are very similar to the logical read and write commands, except that rather than looking for the sectors in numerical order, they are read from the disk in the exact order that they pass under the read/write head. Only the differences between these and the logical commands are described.

DDC in Command Accept Mode

The only difference in setting up for the command is that since the basic command will ignore the header bytes for comparison (Ignore Header-Read/Write Data) the Header pattern registers do not need to be updated. If doing a track operation, the command should start on an index pulse by setting the SAIS bit, and the Number of Sector Operations Counter should be loaded with the number of sectors on the track.

DDC in Command Perform Mode

This is the same as previous operation.

DDC in Result/Error Mode

This is the same as the previous operation.

OTHER MULTI-SECTOR PHYSICAL OPERATIONS—INTERLOCK MODE

The host microprocessor can perform many related but different operations on physically consecutive sectors, by using the Interlock mode. Several possible command sequences are briefly outlined:

1. Header Re-writing—If an ID is detected to have an error in its header bytes a sequence of commands can be executed to re-write the header. The rewritten ID may attempt to fix the error or mark the sector as bad. This involves doing two commands in sequence:

a) First find the header of the sector located physically prior to the Bad ID field.

b) Second, do a write header operation to re-write the ID field.

2. Recovery of Data—This is like one above except no attempt is made to correct the ID field, just obtain the data field information.

a) First find the header of the sector located physically prior to the defective sector.

b) Second, do an ignore header operation to The data field is to be recovered, so a read data operation should be executed, and the data can be read to memory.

3. Sparing a defective sector—This is a more complicated version of 1 and 2 above, and is performed with the following commands. Steps a and b and steps c and d should be performed in the interlock mode, whereas between steps b and c interlock mode may not be desirable since ECC correction may be necessary.

a) First find the header of the sector located physically prior to the defective sector.

b) Second, do a write header operation to re-write the ID field, indicating a bad sector. If the data field is to be recovered the write header operation could be accompanied by a read data operation, and the data can be read to memory.

c) Third the DDC does a compare header operation to find the sector prior to the spare sector (if located on the same track). (If the spare has a known header a Compare Header is all that is necessary.)

d) Finally the spare sector ID is rewritten and the old sector's data is written to the spare sector.

In general, Interlocked mode operation is required only when unique operations on physically adjacent sectors is necessary. General multi-sector operations not performed on adjacent sectors can be cascaded without using the interlock mode.

DDC in Command Accept Mode

Step 1. The Parameter and Count registers must be loaded with the proper format information (if not already loaded with the correct information). The Header byte pattern registers must be loaded with the exact header information for the sector to be that will be operated on. Many Interlock mode command sequences may not need the Number of Sector Operations Counter loaded, but if needed it should be loaded with the number of sectors to be transferred.

Step 2. The Local DMA channel must be initialized. The Local transfer register should be configured to the desired DMA mode. The DMA address registers are loaded with the data transfer address. If the Remote DMA is used in coordination with the local transfer, especially in tracking mode, the Remote Transfer register should be initialized, as should its address, and length information. The header complete and command complete interrupts should be enabled, in order to enable determination of when to load the subsequent sector's information and command. This is done by setting the EI, EIH, IR bits in the Operation Command Register.

Step 3. Finally the Drive Command register is loaded with the desired first Command. If a remote DMA operation is to be performed, the Operation Command register should be loaded to start the remote DMA.

DDC in Command Perform Mode

Step 4. The DDC will perform the operation locating the first sector, acquiring the system bus, and transferring that sector's data. As soon as the first sector's header has been located, then the DDC issues a header complete interrupt, and the host must update all desired registers, and finally loading the Drive Command register with the new information. (Note: if the interlocked operation was a multi-sector (NSO not equal 0) operation then the Drive command is not updated, since the operation is a continuation of the multi-sector operation.) Finally the Interlock Register is written to, and step 4 is repeated until the microprocessor is done writing commands. During the last operation, the Interlock register must be written to avoid a late interlock error.

The command may be terminated early if the DDC could not find one of the correct sectors being sought, or the Interlock register is not written to prior to the beginning of the next sector.

DDC in Result/Error Mode

Step 5. At the end of each header field an interrupt is issued. The operation has completed when the Header Interrupt and Operation Complete is received after the last command. If the command terminated properly, and the μP can read this from the Status and Error registers and can proceed to the next command. If an error occurred the microprocessor should service this by either retrying the operation, or correcting the problem.

GENERAL CHAINING OF DISK COMMANDS

Various commands can be executed one right after the other, like the interlock mode described above, except without the Interlock timing constraints. This is done by enabling the Header Complete Interrupt executing a disk command, and when the interrupt is received by the CPU, it checks to make sure it is the Header Complete interrupt, and that the Next Disk Command bit is set. If it is set the CPU can then execute a new operation, while the old operation is completing. In this way fast access to the data on the disk track can be achieved.

7.5 DMA OPERATIONS

In this section, the DDC's data transfer operations using on-chip or external DMA will be discussed in depth. Discussion of DMA as part of the above disk operations has been omitted in favor of a separate discussion. It is important for the designer to keep in mind that while the type of DMA operation won't affect the individual commands, it can be a very important factor in overall system through-put, and bus utilization. In general, once a disk sector buffering scheme and method of transferring data to/from the system has been designed, the DMA mode selection is obvious, and usually remains fixed. The DDC-system interface connections for different system applications are discussed in chapter 6.

7.5.1 Data Transfer Features

All DMA operations are supported by the following four features. These features are valid for all types of DMA modes described in section 7.5.2 including the Slave mode (external DMA).

PROGRAMMABLE BURST LENGTHS

The data transfer from/to the DDC to/from the system is fully programmable. In single bus systems, the data from/to the FIFO to/from the memory, can either be transferred in 32-byte bursts or in smaller bursts of 2, 8, 16 or 24 bytes (or 1, 4, 8 or 12 words). In dual bus systems, data can be transferred either up to 64 Kbytes in a single operation or in smaller operations. The programmable burst lengths feature accommodates the variations in bus latency time usually present in all systems (see Chapter 6).

The DDC is programmed for the desired data transfer mode through LTEB, LBL1, and LBL2 bits in the Local Transfer Register and the RTEB, RBL1, RBL2 bits in the Remote Transfer Register. For Remote transfers, the DMA Byte Count registers are also used.

8-BIT OR 16-BIT WIDE TRANSFERS

Data can be transferred either byte wide or word wide. This is achieved through LWDT and RWDT bits in the Local and Remote Transfer Registers, respectively. The DMA address counters are incremented by one for byte wide and by two for word wide transfers.

SLOW READ/WRITE

For slow memory or other devices, the normal DMA memory read/write cycle of four periods can be extended to five cycles for all DMA modes (including external DMA), using bit LSRW and bit RSRW in the Local and Remote Transfer Registers respectively. The read/write cycles can also be extended to an infinite length by using the External Status input (pin 17) of the DDC in conjunction with EEW bit in the Remote Transfer Register.

REVERSE BYTE ORDER

This option is only valid for 16-bit wide transfers using the Local DMA channel. It enables the two bytes being transferred to be mapped with the high order byte to AD0–7 and the low order byte to AD8–15, or vice-versa. This could be achieved through RBO bit in the Local Transfer Register.

(Note: This option is still functional in 8 bit mode, however it performs no useful function. When reading, the first byte DMA'd was the second byte read, the second DMA'd byte the first read, the third DMA'd byte the fourth, the fourth DMA'd byte the third, and so on. Similar order occurs for a write.)

7.5.2 DMA Modes

Various data transfers are carried out by configuring the DDC in one of the three main DMA modes: single channel, dual channel or external DMA. Some of this has been discussed in Chapter 6.

SINGLE CHANNEL (LOCAL DMA) MODE

In the local DMA mode, only three DMA registers/counters are used; the Local Transfer Register, DMA Address Byte 0 and 1. The Sector Byte Count 0 and 1 registers determines the sector size. A local transfer operation can be carried out following the steps below:

DDC in Command Accept Mode

Step 1. The DMA Address Bytes (0 and 1) Counters are initialized with local (or main) memory address to/from where the data is to be transferred from/to the FIFO.

Step 2. The Local Transfer Register is set for enabling the local DMA channel (bit SLD), 8- or 16-bit transfer (bit LDWT), reverse-byte order (optional in 16-bit data transfer mode, using bit RB0). Slow Read/Write cycles (bit LSRW), Long Address (bit LA), and the burst length (bits LTEB, LBL1, LBL2).

Step 3. The Operation Command Register is loaded for enabling interrupts, if desired (bit EI).

Step 4. Finally, the Drive Command (DC) Register is loaded with the desired DDC command. See Table 7.5. The DDC enters the command perform mode immediately after the DC register is loaded. (This step is the same as Step 3 in previous read/write operations discussions.)

DDC in Command Perform Mode

Step 5. The DDC should be granted bus control on the occurrence of an LRQ. The DDC will generate an interrupt after the completion of the operation or on the occurrence of an error. (Same as previous read/write operations Step 4.)

DDC In Result/Error Mode

Step 6. On the occurrence of an interrupt, the Status Register is read. In case of an operation complete (the NDC bit), steps 1 through 6 may be repeated, or the DDC may be initialized for a new operation. If an error was occurred, appropriate actions should be taken, as discussed in section 7.7.

DUAL CHANNEL TRACKING (LOCAL AND REMOTE) MODE

In the dual DMA mode, all the DMA registers are used (see Chapter 5). The DDC can further be set for either a Tracking Dual DMA or a NON-Tracking Dual DMA mode.

In Tracking mode, data is transferred from the on-chip FIFO to the system I/O port through the local buffer memory, and vice versa. The entire DMA operation is controlled by the DDC and external arbitration logic that synchronizes the DDC's remote operation with the external DMA for the system, after initialized by the microprocessor. Basically, local and remote transfers are dependent on each other and the DDC keeps track of both transfers in order to avoid any possible data overlapping in the local buffer memory.

This mode effectively turns the buffer memory into a large FIFO. This is accomplished through the use of DMA Sector Counter (DSC), which keeps track of the difference between sectors read/written from/to the disk and sectors transferred to/from the host system. Each time the source transfers a sector or data into buffer memory (length is determined by the Sector Byte Count Register pair), the DSC register is incremented. It is decremented each time the destination has transferred a sector of data. Whenever the DSC register contents become zero, destination transfers are inhibited. Note that in a Disk Read operation, local DMA (or the FIFO) is the source and remote DMA (or system I/O port) is the destination. Similarly, in a Disk Write operation, the remote DMA (or the system I/O port) is the source and local DMA (or the FIFO) is the destination. A detailed step-by-step disk read/write operation, in the tracking dual DMA mode, is given below and the flow chart is given in *Figure 7.12.*

DDC in Command Accept Mode

Step 1. The DMA Address Bytes (0, 1, 2, 3) registers are loaded with the same local and remote start address in the local buffer memory.

Step 2. Bits SLD and SRD in the Local and Remote Transfer Registers are set to enable both local and remote DMA channels. The DDC is configured for Tracking Mode by setting the TM bit in the Remote Transfer Register. Other options such as 8-/16-bit transfer, slow read/write cycles, and

burst lengths for both FIFO and local buffer memory are selected through LWDT, LSRW, LTEB, LBL1 and LBL2 bits in the Local Transfer Register and RWDT, RSRW or EEW, RTEB, RBL1 and RBL2 bits in the Remote Transfer Register. Bit LA in the Local Transfer Register must be reset for 16-bit address mode.

Step 3. The Number of Sector Operations (NSO) counter and Sector Counter (SC) are loaded for multi-sector operation. Only SC should be loaded for a single sector operation.

Step 4. Interrupts are enabled using EI bit in the Operation Command (OC) Register.

Step 5. Finally, the desired Read/Write command (see Table 7.6). (This is the same as Step 3 in the previous Read/Write command descriptions.)

DDC in Command Perform Mode

Step 6. The DDC will start performing the desired operation after the DC register is loaded. An LRQ, RRQ or Interrupt should be expected. The DDC should be given the bus control when LRQ or RRQ occurs. (Same as Step 4 in previous Read/Write command descriptions.)

DDC in Result/Error Mode

Step 7. If an interrupt is generated, it should be serviced properly (see section 7.7 for interrupt servicing). If the operation was completed successfully, steps 1 through 6 may be repeated for a new operation. (Same as Step 5 in previous Read/Write command descriptions.)

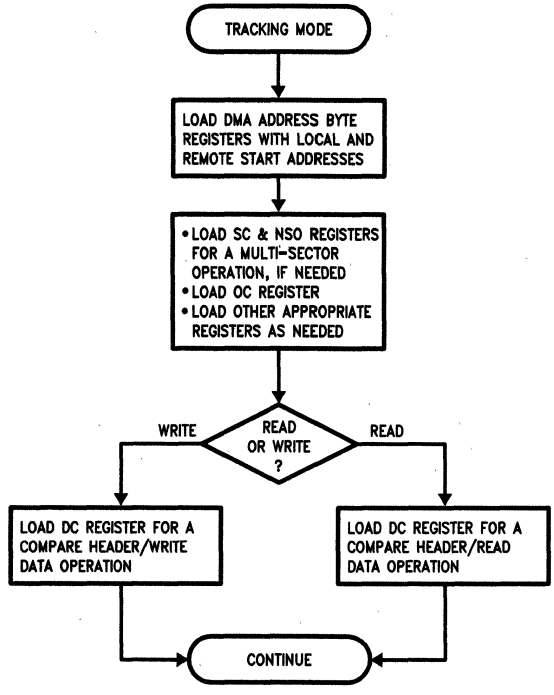
DUAL DMA NON-TRACKING (LOCAL AND REMOTE) MODE

In the non-tracking dual channel DMA mode, the Local and Remote transfers are independent of each other. The controlling microprocessor has to keep track of both transfers to avoid any possible data overlapping in the local buffer memory. The DMA Address (bytes 0–3) Registers are set up independently for Local and Remote transfers. All the necessary steps needed to perform a data transfer between the FIFO and system I/O port with the DDC in this mode are explained below and also shown in a flow chart in *Figure 7.16.*

DDC in Command Accept Mode

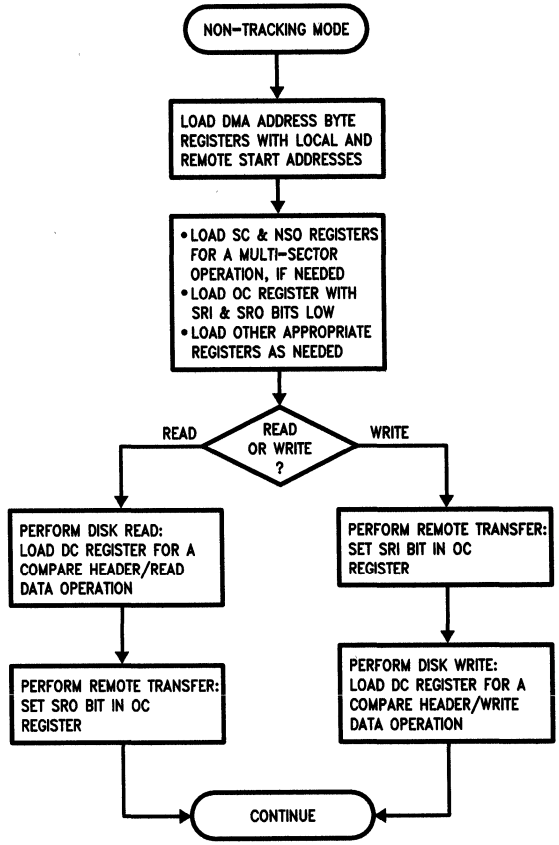
Step 1. The DMA Address (byte 0–3) are set up for the desired Local and Remote addresses. Any address within 64k memory space could be loaded.

Step 2. The DDC is configured in the non-tracking mode, first by enabling Local and Remote DMA channels through SLD and SRD bits in the Local and Remote transfer Registers. The LA bit is also set to zero for 16-bit address for both DMA channels. Other DMA options such as 8-/16-bit data transfer slow read/write, reverse byte ordering, and burst length, are selected through LWDT and RWDT; LSRW, RSRW and EEW; RBO; and LTEB, LBL1, LBL2, RTEB, RBL1, and RBL2 bits in the Local and Remote Transfer Registers.



TL/F/8663–E9

FIGURE 7.12. Dual DMA Tracking Mode μP Programming Flow Chart



TL/F/8663–F0

FIGURE 7.13. Dual DMA Non-tracking Mode μP Programming Flow Chart

Step 3. The Number of Sector Operation (NSO) Counter and Sector Counter (SC) are loaded for multi-sector operation. Only SC is loaded for a single sector operation.

Step 4. Write Operation: Before having the DDC to perform a write operation, data is transferred from the system I/O port to the buffer memory by enabling the Remote channel through SRI bit in the Operation Command Register. Interrupts are also enabled using the EI bit.

Read Operation: The Drive Command (DC) Register is loaded with the desired Read command (refer to Table 5.6). If a multisector operation is selected, the SAIS bit may be set to zero for the operation to start at the Index pulse.

DDC Performs the Remote Transfer

Write Operation: The DDC will transfer the remote data to local memory and will issue an operation complete interrupt. Remote transfer operations could be repeated to fill the local memory before performing a disk operation. The DDC now should be initialized for the actual disk write operation.

Read Operation: The DDC will complete the disk read just like a normal operation. See Step 7. Now the data may be transferred to the system I/O or any remote locations.

Step 5. Finally, the Operation Command Register is loaded to enable data transfer from the local buffer memory to the system I/O, with SRO bit set. The interrupts may also be enabled with the EI bit, if required.

The DDC will start the operation when the OC Register is loaded. The RRQ must be acknowledged.

Step 6. The Drive Command register is loaded with the desired write command. If a multi-sector operation is desired, the SAIS bit may be reset for the operation to start at the Index pulse. (This is the same as Step 3 for Read/Write operations discussed previously.)

DDC in Command Perform Mode

Step 7. The DDC will start performing the desired write operation immediately and will issue a local request, LRQ. Upon receiving an LACK, it completes the write operation. (This is the same as Step 4 for previously discussed Read/Write operations.)

DDC in Result/Error Mode

Step 8. The DDC will issue an interrupt which should be serviced properly (refer section 7.7 for interrupt servicing). In case of an operation complete interrupt, steps 1 through 5 may be repeated for a new operation.

IMPORTANT NOTES

1. By setting both SRI and SRO simultaneously, any non-tracking DMA operation will stop. The current remote address and remote data byte count will be retained, and the local DMA will be unaffected. Loading the original OC instruction (input or output) will restart the original instruction from the last remote DMA address.

2. In either Tracking or Non-tracking mode, if either channel is loaded with an odd byte transfer count, the DDC will transfer the next higher even byte. For example, if 511 was loaded in Remote Data Byte Count Registers, 512 bytes would be transferred, with the valid data only in the first 511 bytes.

3. In the Tracking mode the DDC keeps track of the data in the buffer memory. The Remote Transfer follows the Local transfer by a sector length and the DDC makes sure that the correct data is transferred to the system memory. However in the Non-tracking mode the remote channel is independent of the disk operation and hence the remote channel can follow the local channel as closely as possible. The microprocessor is responsible of preventing overlap of data.

4. Even though normally the remote channel would be used for transfers from system to buffer memory (and vice versa) and the local channel for transfers from the buffer memory to the FIFO (and vice versa), the remote channel could also be used for some other purpose that is independent of the DDC's other operations.

EXTERNAL DMA

In external DMA mode, the data transfer between the on-chip FIFO and external memory (local buffer or system) is controlled by the external DMA. The DDC is programmed to perform a disk read/write operation without the internal DMA. Whenever the FIFO needs any data transfer, the DDC asserts LRQ. At this point, external DMA takes control and completes that particular data transfer. The following steps illustrate the necessary actions to perform a disk read/write operation using external DMA i.e. DDC in the slave mode.

DDC in Command Accept Mode

Step 1. The registers are initialized.

Step 2. Interrupts are enabled using EI bit in the Operation Command Register.

Step 3. Finally, the Drive Command Register is loaded for the desired Read/Write operation.

DDC in Command Perform Mode

Step 4. The DDC will start performing the operation and LRQ will be asserted when the FIFO requires the data transfer. The LRQ must be acknowledged by the external DMA in order to complete the operation.

DDC in Result/Error Mode

Step 5. If an interrupt is issued, it must be serviced, (refer section 7.7). In case of an operation complete interrupt, steps 1 through 3 may be repeated for a new operation.

7.6 ERROR DETECTION AND CORRECTION

The Disk Data Controller, DDC has comprehensive and versatile error detecting and correcting capabilities. It features a fully programmable ECC;

- Programmable Preset Pattern
- Programmable Polynomial Taps
- Programmable Correction Spans
- Programmable Assignment of CRC/ECC on Header or Data

There are essentially two internal codes available; a fixed Cyclic Redundancy Checking (CRC) code for detecting errors only, which uses a CRC-CCITT polynomial that provides 16 generated check bits for appending to the Header fields and/or Data fields. The other type is the ECC code which may be a Fire code or a Computer generated code with 32 or 48 generated check bits that may be appended to the Header field and Data field. National Semiconductor recommends a computer generated polynomial called the Glover 140A0443 code with a correction span of 5-bits for MFM encoded drives. The designation represents the hexadecimal equivalent of the forward polynomial and it requires a preset of all 1's. The code has two polynomials; the forward one for checkbit generation and checking and the reverse one for error location. The error detection span is 32-bits while the correction span is 5-bits. The number of bytes in the sector determines the integrity of the code. The maximum sector length the code can handle is 1024 bytes of data and 4 bytes of ECC, which is within the limits of most disk formats. The completely programmable feature of the DDC with respect to ECC offers a lot of flexibility to the user. In case a user prefers to use his own high integrity code, the DDC can be configured to interface easily with external ECC circuitry and the DDC can be programmed to operate in the external ECC mode, as discussed in chapter 4. There are essentially three kinds of operations associated with the ECC circuitry: 1) checkbit generation, 2) checkbit verification, and 3) error location.

7.6.1 Error Detection

Internal Checkbit Verification—Write

This operation occurs when the controller is performing a write operation. The ECC shift register is downloaded with the preset value stored in the Preset Register. The code length is selected independently for Header and Data appendage. Bits being shifted out of the SERDES (serializer/deserializer) to the disk, are also shifted into the ECC Shift Register. When the last bit of the Header or Data field has been transmitted out of the DDC, the generated check bits in the ECC shift register are directly shifted out and onto the disk, starting with the MSB and ending with bit 0. After the ECC bits have been appended, the DDC switches to the next field.

Internal Checkbit Verification—Read

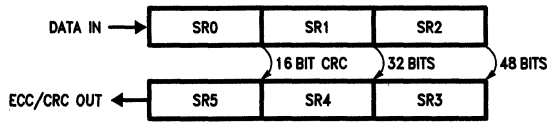
This function will occur concurrently with a read data operation from the disk. The ECC Shift Register is first preset from the Preset Registers. The incoming Header or Data field is serially fed into the same ECC Shift Register that is used to generate checkbits. When all the Header or Data field bits and all the generated checkbits have entered the ECC Shift Register, the status of the bits in it is checked for an all zeroes condition. If it is true then the field contains no errors, else if any of the ECC Shift Register's bits are high, the field contains an error. In the case of a Header field error, the Header Fault bit (SO), in the Status Register is set, while in case of a Data field error, the Data Field Error bit (E1), of the Error Register is set.

System Alternatives on Error Detection

Once an error has been detected by the ECC logic, the Re-enable (REN) bit must be reset via the Drive Command Register, before proceeding. If a Header field error is detected, the DDC will react differently depending on the Header operation involved. The various options are discussed under the Drive Command Register description. If an error is detected in the Data field, a re-read is initiated (by the system), to overrule a soft error. If the data is still not corrected after several re-reads, it implies the detection of a hard error. By re-reading the sector in question and comparing the syndromes to previous retries, a certain level of confidence can be reached, that the error is media induced and ECC correction can be attempted. The syndrome bytes in the ECC Shift Register will contain the bit error information, although the bytes in error have been transferred to memory.

7.6.2 Error Correction

The DDC has a maximum correction span of 15 bits, i.e. it can correct up to 15 contiguous bits in error, or a span of errors 15 bits or less. Of course correction can only be attempted if internal ECC checkbits were appended to the data field when written to the disk. The first step in the correction process is to load the Data Count Register with the data count, (sector byte count) plus the number of bytes of checkbits, i.e. sector byte count registers must be initialized to sector length plus 4 or 6 for 32 bit mode or 48 bit mode ECC respectively. Then the correction cycle is initiated by setting the Start Correction Cycle bit of the Operation Command Register. This should be done before any further Drive Command operation is issued to the DDC. This prevents the destruction of the stored syndromes in the ECC Shift Register. Also while the correction cycle is in progress,



TL/F/8663–F1

FIGURE 7.14(a). Hardware Configuration of ECC Shift Register

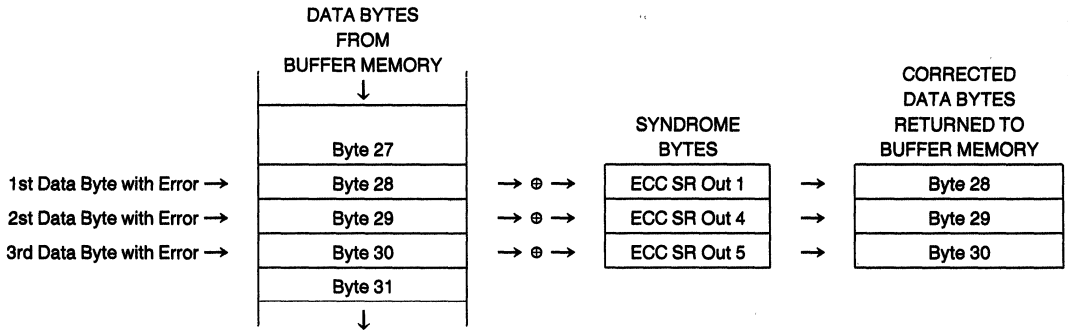


FIGURE 7.14(b). 32-Bit ECC Correction Process

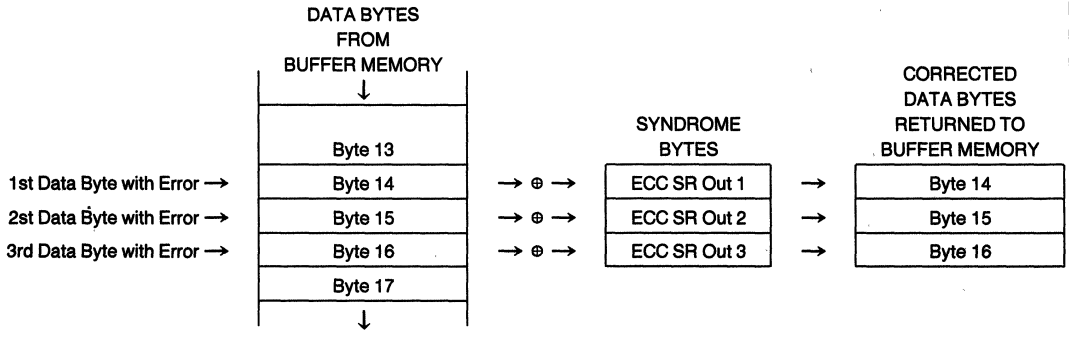
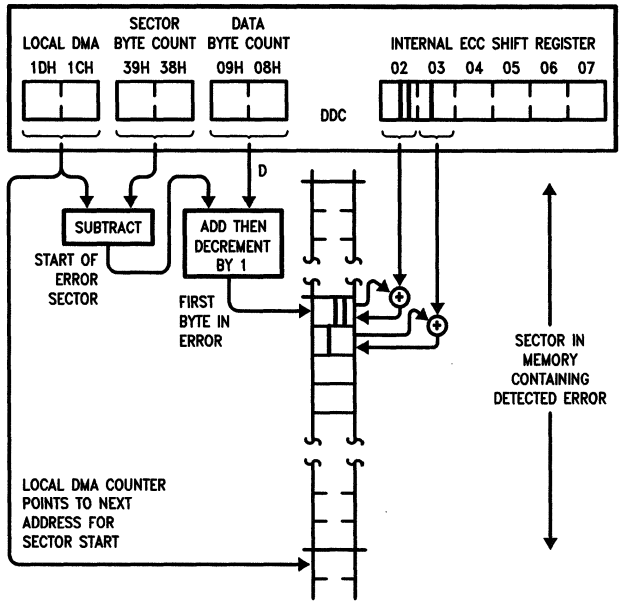


FIGURE 7.14(c). 48-Bit ECC Correction Process

MP Operations after Error Located in DDC



TL/F/8663–F2

FIGURE 7.15. Location of Bytes in Error (in Memory) for the Correction Process

the DDC ignores any drive command loaded into the Drive Command Register. On initiation of the Correction Cycle, the Correction Cycle Active flag, (bit 6 of the Status register), will go high.

The ECC Shift Register contains encoded information with regards to both the location of the bytes in error and the error pattern. The ECC Shift Register's contents are transposed which sets up a reverse shift without actually reversing the direction of shift in the shift register. The advantage of reverse shifting is that a non-correctable error is determined much quicker than if forward shifting is used. It also guarantees the completion of the correction cycle within the time it takes to read one sector of the disk. The ECC logic begins shifting, looking for a zero detect, i.e. detection of all zeroes in the upper (32-C) or (48-C) bits of the ECC Shift Register, where C is the correction span selected. After 8 shifts, the Data Count Register begins decrementing, with one down count for every 8 shifts of the ECC Shift Register. When the zero detect condition occurs, the control logic will stop decrementing the Data Count Register and its state indicates the byte that is in error. If the Data Byte Counter decrements to zero before the selected most significant bits of the ECC Shift Register are all zeroes, the error is non-correctable. In case of this condition or the zero detect condition of the ECC Shift Register, an interrupt is issued to indicate to the host microprocessor that the correction cycle has finished, indicated by the CCA flag (bit 6 of the Status Register being reset).

During the correction cycle other operations like completion of remote DMA etc., may issue an interrupt which should be serviced to enable recognition of the interrupt on completion of the correction cycle. The Error Register bit CF is examined, which if set signifies a non correctable error. If the bit is not set, then the error is correctable and must be either in the data field or the checkbits of the ECC field or overlapping both fields.

At the instant when the 'zero detect' condition occurs in the ECC Shift Register, the status of the Data Count Register indicates the byte in error. For example—if the data count register shows 515, then the 515th byte of the data field is in error. If there were only 512 bytes in the data field, then 515 means that the 3rd byte of the checkbit field is in error. The syndrome bytes in the ECC Shift register should be aligned so that the Most Significant Bit of the syndrome field align with the Most Significant Bit of the byte 515. However, if the syndrome spans a field of two bytes, then it will align with byte 515 and 516. When the data byte in error is located, the ECC logic makes sure that the syndrome bits are aligned properly on a bit by bit basis with that byte in error. Therefore, it will continue to shift until this has happened. To facilitate the speed restraints of the process, the syndrome

will get shifted one full byte beyond where one would expect to see it in the ECC Shift Register. The syndrome bits will be located starting at the second byte of the ECC Shift Register. *Figure 7.14 (a)* shows the orientation of the ECC Shift Register for various sizes of polynomials selected.

Errors are corrected by XOR'ing syndrome bytes (ECC SR 0–5) with the bytes in the data record in memory that contain the error. The address of the first byte of the data field, in error is computed as follows: [current value of DMA address bytes 0 and 1] − [sector byte count] + [data byte count] − 1. For performing a correction with 32 bit ECC, ECC SR1, ECC SR4, and ECC SR5 contain the syndrome pattern in that sequence. ECC SR2 and ECC SR3 are not used in 32-bit mode and will contain 0's if read. ECC SR0 will contain all 0's if the error is correctable, and may contain some set bits if not. The bytes in error (in the memory) are located as shown in *Figure 7.15* while the correction process is shown in *Figure 7.14 (b)* and *(c)*. To perform a 48 bit correction ECC SR1, ECC SR2 and ECC SR3 should be read sequentially for the syndrome bits. ECC SR0, ECC SR4 and ECC SR5 are not used and will contain 0's for a correctable error. *Figure 7.16* shows an example of the correction process. *Figure 7.17* gives a flow chart of the correction cycle operations.

EXAMPLE OF A 32-BIT CORRECTION

Shown in *Figure 7.18* is a record with several bits read in error from the disk. Bits D4, D11, D13 and D14, now located in memory were read incorrectly and need to be corrected. As can be seen, the correction pattern provided in ECC SR1 and ECC SR2 can be used to correct bits D4, D11, D13 and D14. The CPU reads the Data Byte Count and computes the address of the first data byte in error, read from the disk. This byte is XOR'ed with ECC SR1 and is written back to memory. The second byte read form the disk is XOR'ed with ECC SR4 and then written back. ECC SR5 need not be used since it contains all 0's.

7.6.3 Programming the ECC

There are two sets of six registers used to program the ECC. One set of six is used to program the polynomial taps, while the other set is used to establish a preset pattern. Bits contained in the ECC Control Register are used to control the correction span. The Data Format Register contains bits for choosing the desired type of appendage: either 32 or 48 bit programmable ECC polynomials, or the 16 bit CCITT CRC polynomial.

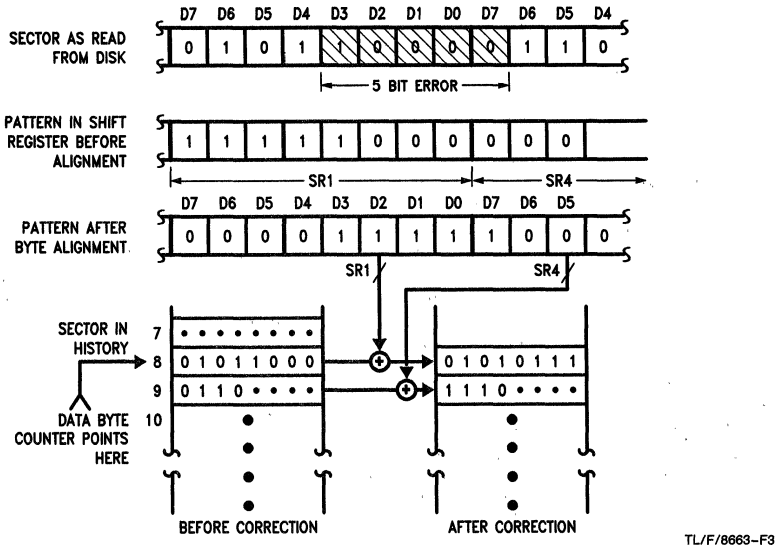
PROGRAMMING POLYNOMIAL TAPS

To program a polynomial into the shift register, each tap position used in the code must be set to 0, and all unused taps should be set to 1. The bit assignment for these registers in 48 and 32-bit modes is shown in the tables, *Figure 7.21 (a)*. It is important to note that for 32-bit codes, PTB2 and PTB3 must be set to all 1's. Failure to do so will result in improper operation. Also x^{48} for 48-bit and x^{32} for 32-bit ECC are implied and so is x^0, even though this bit is accessible.

PROGRAMMING PRESET PATTERN

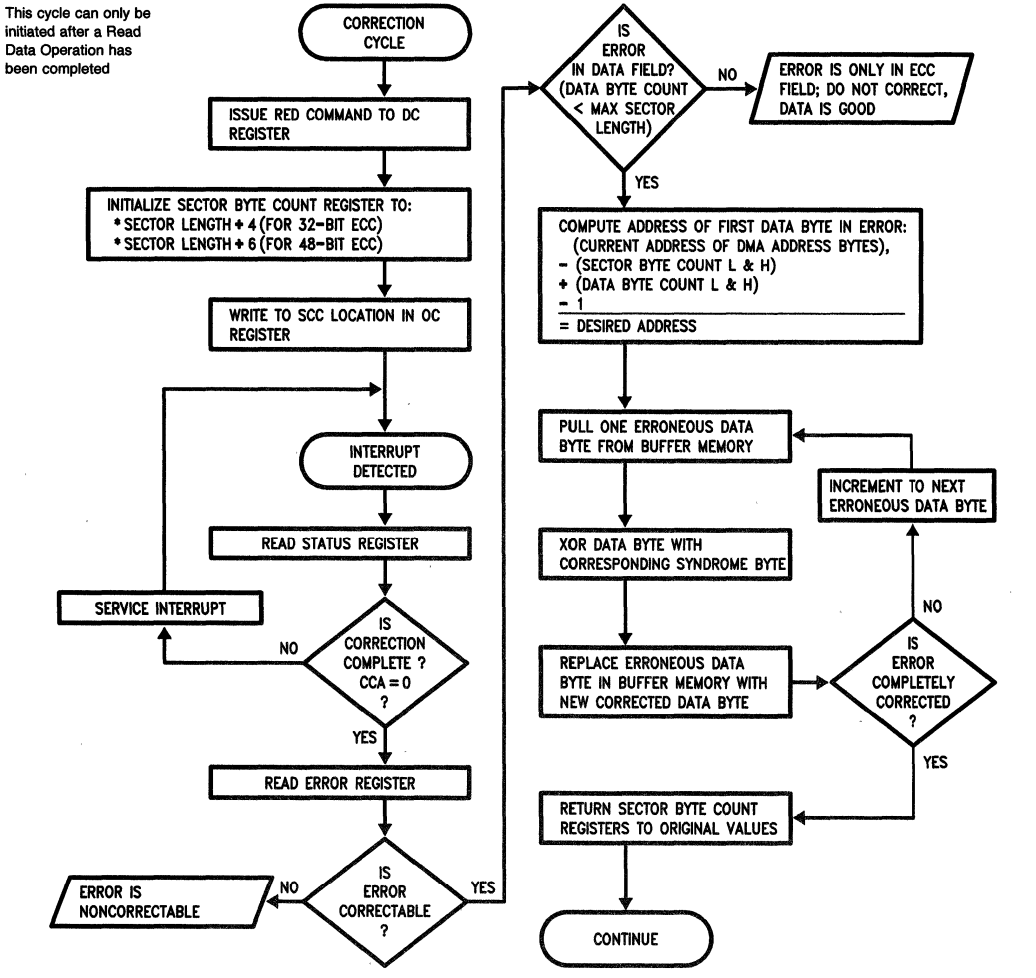
PPB0–PPB5 must be initialized to program the preset pattern that the shift registers will be preset to. As in the polynomial taps, x^{48}, x^{32}, and x^0 are implied. The assignment of the bits for 48 and 32-bit modes is shown in *Figure 7.21 (b)*. The value programmed into each register will be the preset pattern for the eight bits of the corresponding shift register. For typical operation, these will be programmed to all 1's. All unused presets should be set to 0. In 32-bit mode, PPB2 and PPB3 must be set to all 0's. Failure to do so will result in improper operation.

An Example of ECC in Action



TL/F/8663–F3

FIGURE 7.16. How Error Correction Works



TL/F/8663–F4

FIGURE 7.17. Flow Chart of the Correction Cycle Operation

ECC CONTROL REGISTER

The ECC Control Register controls a number of functions. The correction span can be programmed using four bits of this register. Errors longer than the correction span are treated as non-correctable. The allowable correction span is 3–15 bits. If a span outside this range is loaded, then the DDC defaults to a span of three bits. There is a bit (HEN) to indicate whether Header address mark and/or synch fields are encapsulated in the CRC/ECC calculation. There is also a bit (DEN) for indicating whether data address mark and/or synch fields are encapsulated in the CRC/ECC calculation. Facility for inverting data entering and leaving the ECC Shift Register is also provided. For selecting the internal 16-bit CRC polynomial, the appropriate bits in the Disk Format register are set, the ECC Control Register is programmed as desired.

Syndrome Pattern

Register		Bit Number							
		7	6	5	4	3	2	1	0
ECC	SR1	0	0	0	1	0	0	0	0
ECC	SR4	0	1	1	0	1	0	0	0
ECC	SR5	0	0	0	0	0	0	0	0

Buffer Memory

Corresponding Buffer Data Bit Pattern							
D7	D6	D5	*	D3	D2	D1	D0
D15	*	*	D12	*	D10	D9	D8
D23	D22	D21	D20	D19	D18	D17	D16

* = location of bits in error

Figure 7.18 Example of Correction Syndrome Bits relating to Data Bit Patterns

Tap Assignment 32-Bit Mode

REG#	ADDR	Bit Number							
		DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
PTB0	08	X^7	X^6	X^5	X^4	X^3	X^2	X^1	X^0
PTB1	09	X^{15}	X^{14}	X^{13}	X^{12}	X^{11}	X^{10}	X^9	X^8
PTB2	0A	1	1	1	1	1	1	1	1
PTB3	0B	1	1	1	1	1	1	1	1
PTB4	0C	X^{23}	X^{22}	X^{21}	X^{20}	X^{19}	X^{18}	X^{17}	X^{16}
PTB5	0D	X^{31}	X^{30}	X^{29}	X^{28}	X^{27}	X^{26}	X^{25}	X^{24}

Preset Bit Assignment 32-Bit Mode

REG#	ADDR	Bit Number							
		DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
PPB0	02	X^7	X^6	X^5	X^4	X^3	X^2	X^1	X^0
PPB1	03	X^{15}	X^{14}	X^{13}	X^{12}	X^{11}	X^{10}	X^9	X^8
PPB2	04	0	0	0	0	0	0	0	0
PPB3	05	0	0	0	0	0	0	0	0
PPB4	06	X^{23}	X^{22}	X^{21}	X^{20}	X^{19}	X^{18}	X^{17}	X^{16}
PPB5	07	X^{31}	X^{30}	X^{29}	X^{28}	X^{27}	X^{26}	X^{25}	X^{24}

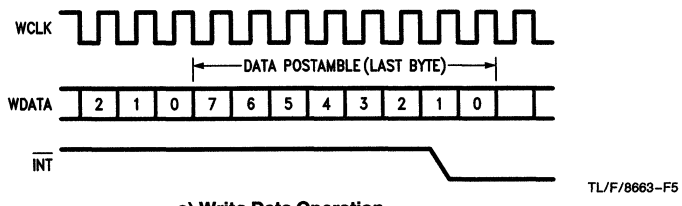
Tap Assignment 48-Bit Mode

REG#	ADDR	Bit Number							
		DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
PTB0	08	X^7	X^6	X^5	X^4	X^3	X^2	X^1	X^0
PTB1	09	X^{15}	X^{14}	X^{13}	X^{12}	X^{11}	X^{10}	X^9	X^8
PTB2	0A	X^{23}	X^{22}	X^{21}	X^{20}	X^{19}	X^{18}	X^{17}	X^{16}
PTB3	0B	X^{31}	X^{30}	X^{29}	X^{28}	X^{27}	X^{26}	X^{25}	X^{24}
PTB4	0C	X^{39}	X^{38}	X^{37}	X^{36}	X^{35}	X^{34}	X^{33}	X^{32}
PTB5	0D	X^{47}	X^{46}	X^{45}	X^{44}	X^{43}	X^{42}	X^{41}	X^{40}

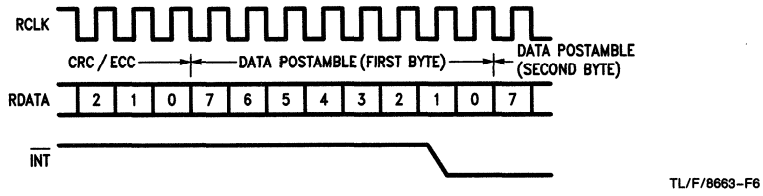
Preset Bit Assignment 48-Bit Mode

REG#	ADDR	Bit Number							
		DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
PPB0	02	X^7	X^6	X^5	X^4	X^3	X^2	X^1	X^0
PPB1	03	X^{15}	X^{14}	X^{13}	X^{12}	X^{11}	X^{10}	X^9	X^8
PPB2	04	X^{23}	X^{22}	X^{21}	X^{20}	X^{19}	X^{18}	X^{17}	X^{16}
PPB3	05	X^{31}	X^{30}	X^{29}	X^{28}	X^{27}	X^{26}	X^{25}	X^{24}
PPB4	06	X^{39}	X^{38}	X^{37}	X^{36}	X^{35}	X^{34}	X^{33}	X^{32}
PPB5	07	X^{47}	X^{46}	X^{45}	X^{44}	X^{43}	X^{42}	X^{41}	X^{40}

FIGURE 7-19. Programming the Presets of Taps; the Tap and Preset Register Configurations

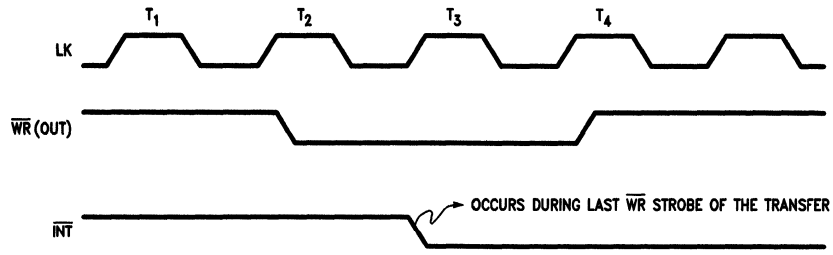


a) Write Data Operation

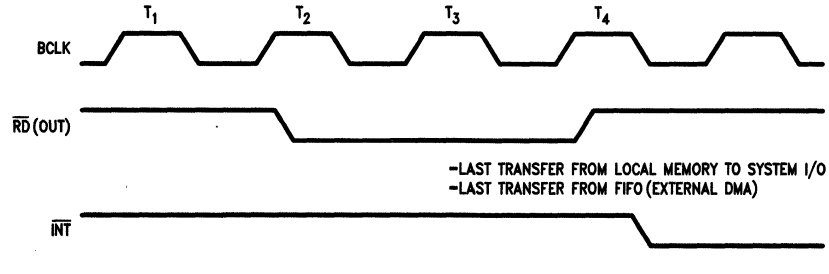


b) Check Data Operation

FIGURE 7.20. Operation Complete Interrupt Timing



c) Read Data, in Single or Dual Channel (Non-tracking) Move



d) Read Data, in Dual Channel (Tracking) or in an External DMA Mode

FIGURE 7.20. Operation Complete Interrupt (Continued)

Example of programming the ECC registers

Objective: To program the 32-bit polynomial of the form; (This is National Semiconductor's recommended polynomial)

$x^{32} + x^{28} + x^{26} + x^{19} + x^{17} + x^{10} + x^6 + x^2 + x^0$

with a preset of all 1's, a correction span of 5-bits with no header/data encapsulation. The registers would be programmed as given below. Note that as defined earlier, PTB2 and PTB3 must be all 1's and PPB2 and PPB3 must be all 0's.

Polynomial Taps Registers

REG#	7	6	5	4	3	2	1	0
PTB0	1	0	1	1	1	0	1	0
PTB1	1	1	1	1	1	0	1	1
PTB2	1	1	1	1	1	1	1	1
PTB3	1	1	1	1	1	1	1	1
PTB4	1	1	1	1	0	1	0	1
PTB5	1	1	1	0	1	0	1	1

Polynomial Preset Registers

REG#	7	6	5	4	3	2	1	0
PPB0	1	1	1	1	1	1	1	1
PPB1	1	1	1	1	1	1	1	1
PPB2	0	0	0	0	0	0	0	0
PPB3	0	0	0	0	0	0	0	0
PPB4	1	1	1	1	1	1	1	1
PPB5	1	1	1	1	1	1	1	1

ECC Control Register

BIT#	D7	D6	D5	D4	D3	D2	D1	D0
	1	0	0	1	0	1	0	1

7.6.4 Internal ECC Diagnostics

The DDC has a diagnostic capability for validating the internal ECC function. By loading the Data Byte Count Register with the number of bytes in the sector plus the number of bytes of ECC appendage for the Data field. The internal CRC/ECC appendage for the Data field is set to zero so that no CRC/ECC will append the data field. Next the microprocessor sets up a data pattern in memory of all zeroes for the nominal sector length, except for bit positions where simulated errors are desired. Also, the microprocessor appends to this data the ECC appendage for an all zeroes data field by setting the Drive Command Register to perform a Compare Header-Write Data operation. In this way the DDC executes a diagnostic write function. In this mode, the data field from memory is written as in a normal write operation to the data field of the selected sector. Then the 32-bits or 48-bits of ECC check are also issued, where these check bits are falsely generated as if from an all zeroes data field. The selected sector now contains an all zeroes data field with simulated error bits followed by an ECC appendage representing checkbits generated from an all zeroes data field. The Data Byte Count is now re-loaded with the normal sector length and the correct ECC appendage length selected. A subsequent Read Data operation should produce an error indication. A correction cycle can then be implemented and the syndromes can be examined along with the Data Byte Counter contents. The microprocessor can then compare these syndromes with the positions of the simulated error bits previously written in the data field. This offers the user a diagnostics capability that simulates errors easily, merely by writing the data field with all zeroes except where the simulated error locations are desired.

7.6.5 Encapsulation of Internal ECC with External ECC

The external ECC field may be used to encapsulate the internal ECC/CRC field as a confirmation of error detection. The advantages of this scheme are that both external and internal ECC must agree on 1) the existence of the error, 2) location of the error, and 3) the error pattern. If an error is detected either internally or externally, the DDC will operate as if an internal error were detected.

7.7 INTERRUPTS

The DDC will interrupt the microprocessor only if the Interrupt Enable bit (EI) in the Operation Command register is set high. If it is not set, the INTERRUPT output is always forced high.

7.7.1 Types of Interrupt

The interrupts generated by the DDC can be divided into four categories:

1) Operation Complete Interrupt
2) Header Complete Interrupt
3) Error Interrupt
4) Correction Cycle Complete Interrupt

Each of the above mentioned types is explained in the following.

OPERATION COMPLETE INTERRUPT

The DDC will interrupt the microprocessor when it completes any one of the legal header-data disk operations listed in Table 5.13. The interrupt will also indicate that the DDC is ready to execute a new command. Some interrupt generation situations are explained below.

1) An interrupt will occur when the remote transfer is completed during a disk read operation in Tracking mode.

2) An interrupt will occur when the local transfer is completed during a disk read operation in Non-tracking mode.

3) In Non-tracking mode, if remote DMA channel is enabled, an interrupt will occur after the remote transfer is completed independent of the disk operation or the local transfer.

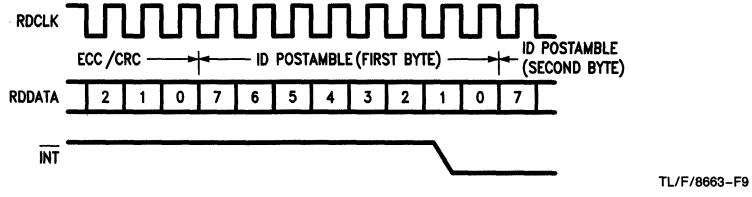
4) If the operation was a multi-sector operation, an interrupt will only occur on the completion of the last operation.

The Operation Complete interrupt generation for various Header and Data operation is shown in *Figure 7.20*. In disk write operations, the operation complete interrupt is generated when last byte of data postamble is being output by the DDC. In Header-Check data operation, operation complete interrupt occurs when first byte of data postamble enters the DDC. In disk read operations when the DDC is using only its local DMA channel (single channel DMA and non-tracking DMA modes), the operation complete interrupt is generated when the last byte (or word) is transferred to the memory. Basically it is coincident with the last WR\ strobe. When the DDC is in dual channel DMA mode, the operation complete interrupt is issued during the last RD\ strobe i.e. when last byte (or word) is transferred from local memory to system I/O. Similarly, when an external DMA is used, the operation complete interrupt is generated during the last RD\ strobe i.e. when last byte (or word) is transferred from the DDC to external memory.

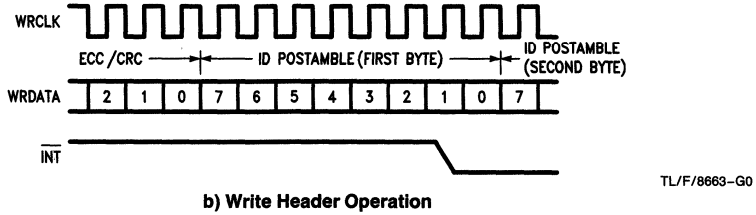
HEADER COMPLETE INTERRUPT

In all legal DDC operations listed in Table 5.13, an interrupt will be generated after a header operation only if the Enable Header Complete Interrupt bit (EHI) in the operation command register is set high. In case of multi-sector operation, this interrupt will be generated after each header of a sector has been operated on. The header complete interrupt feature is commonly used when the DDC is in Interlock Mode (Refer to section 5.2.5). On interrupt, the ID and Data fields for the next sector can be changed, if desired, before the next sector operation starts.

The header complete interrupt is coincident with the Next Disk Command bit (NDC) being set in the Status register. Thus, the controlling microprocessor can be notified to load the DDC with the next disk command. In other words, the DDC could be run continuously for any length of time by loading a new disk command whenever next disk command flag is set. The generation of header complete interrupt is shown in *Figure 7.21*. In Compare, Read and Ignore Header



a) Compare, Read or Ignore Header Operations



b) Write Header Operation

FIGURE 7.21. Header Complete Interrupt Generation

operations, the interrupt is generated when first byte of ID Postamble is being read by the DDC whereas in Write Header operation, interrupt is generated when first byte of ID Postamble is being written to the disk by the DDC.

ERROR INTERRUPT

An interrupt will be generated if any bit in the Error register is set, which in turn sets the Error Detected bit (ED) in the Status register. Refer to description of Error and Status registers in section 5.1.1. Also an error interrupt will be issued.

CORRECTION CYCLE COMPLETE INTERRUPT

An interrupt will occur at the end of an internal correction cycle independent of the result of the correction cycle. If the error was not correctable, another interrupt will not be generated, only the Correction Failed flag (CF) in Error register will be set.

7.7.2 Interrupt Servicing

As explained earlier, the DDC issues an interrupt on, an operation complete, the header operation complete, the occurrence of an error and the completion of a correction cycle. Whenever an interrupt is generated, the Status and Error register should be read in order to find out which one of the four situations has happened. In the status register, flag NDC indicates the completion of an operation, flag HMC indicates the completion of a header operation, and flag ED indicates the occurrence of an error. Only when the ED flag is set, the Error register is read to find out the type of error that caused an interrupt. Also, the CF flag in the Error register indicates result of the correction cycle. The interrupt servicing for various interrupts is described below. *Figure 7.22* shows a flow chart for servicing interrupts.

Operation Complete Interrupt: In case of operation complete interrupt, the NDC flag in the Status register gets set indicating that the DDC is ready for next command. The DDC is brought to the Command Accept mode and the desired command is loaded with all other related registers initialized. Refer to sections 7.1 through 7.6.

Header Complete Interrupt: The HMC flag in the Status register, gets set in case of header complete interrupt. This basically indicates that the header operation (ignore, compare, read or write) has completed. The information in the DDC's registers can be changed before the start of next header operation i.e. during the time when the data operation for the current sector is in progress. If the DDC is in Interlock mode, the HBC/interlock register is also written to during this time. See sections 5.2.5 and 7.2 (interlock format method).

Error Interrupt: In case of an Error interrupt, the ED flag in the Status register gets set. The Error register should be read next, to find out the error that caused the interrupt. For the description of various error flags, refer to Error register description in chapter 5. The HFASM function is explained in detail in section 7.8. Also see description of Header Byte control register in chapter 5. The DFE (data field error) is caused by an ECC/CRC error in the data field. Generally, retrying the operation takes care of this error. If this error does repeat on retries, a correction cycle should be performed. See section 7.6. The SNF (sector not found) error could also be resolved by retrying the operation. If it does repeat on retries, then the head should be repositioned. The SO (sector over run) occurs while reading or writing more data than what has been allotted on the disk and could be taken care of by checking system software. The NDS (no data synch) occurs because of a mismatch in address mark or synch fields and could be resolved by retries or system check-up. The FDL (FIFO data lost) could occur due to speed incompatibility between system and the disk drive and could be resolved by retrying or checking the system. The CF (correction cycle failed) can be taken care of by retrying the correction cycle again, if still not resolved, then that means the error is not correctable. See section 7.7 on ECC/CRC. The LI (late interlock) could also be resolved by retrying. See section 5.2.5 for details on interlock operation.

Correction Cycle Complete: The Error register is read. If CF flag indicates that the correction cycle failed then it can be performed again. After retry if it still fails, then the error is not correctable. See section 7.6 for details on correction cycle.

7.7.3 Interrupt Clearing

The INT pin will be forced inactive high any time the status register is read. If an interrupt condition arises during a status read, an interrupt will be generated as soon as the status read is finished. INT pin will also be deactivated by setting the internal Reset bit (RES) or asserting the external RESET pin. Clearing the RED bit in Drive Command register will not deactivate an interrupt.

7.8 ADDITIONAL OPERATIONS

7.8.1 Data Recovery Using the Interlock Feature

The potential use of the interlock feature is in recovering data from a sector with an unreadable header or ID field. It is assumed that the number of the sector physically preceding the bad sector on the disk is known. A single-sector operation will be performed on these sectors, and the Drive Command Register will be changed in between them. The following steps will recover the data.

Step 1. The header bytes of the physical sector preceding the desired sector are loaded into the relevant header byte pattern registers.

Step 2. The OC Register must be loaded with the EI, EHI, IR bits set. This enables the Header Complete interrupt as well as the interlock feature.

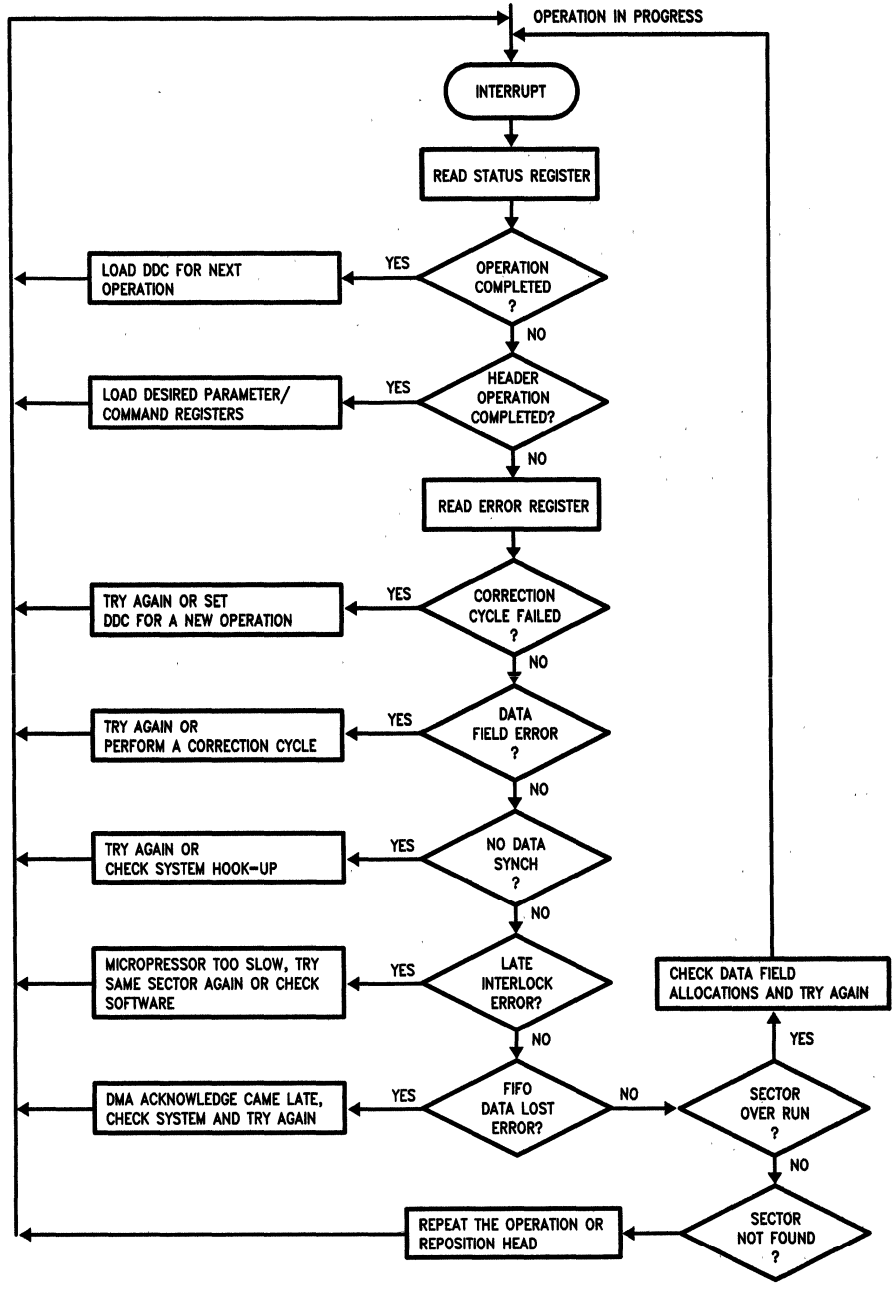
Step 3. The DC register is loaded for a single-sector, compare header/check data operation.

Step 4. After the header complete interrupt, the DC register must be loaded with an Ignore Header/Read data operation, and the Interlock Register (HBC) written to. If the controlling microprocessor fails to write to the HBC register before the end of the data field of the first sector, a Late Interlock error (LI bit in the Error Register) will be flagged, and the operation will be terminated with an interrupt.

Step 5. When the HMC interrupt occurs on the second sector, the Interlock (HBC) Register must be written to again to avoid a LI error.

Step 6. The operation will terminate normally when the data from the badly labeled sector has been read.

Figure 7.23 shows the data recovery algorithm.



TL/F/8663–G1

FIGURE 7.22 Interrupt Servicing

7.8.2 The Header Failed Although Sector Matched (HFASM) Function

The Header Failed Although Sector Matched, HFASM function can be used to perform some disk maintenance and diagnostic functions. The HFASM function (pronounced H-fazzzm) has been described very basically in section 5.1.1 and 5.1.3. In this section, an attempt is made to provide a more detailed description and some example uses.

The HFASM function is essentially an Error that can be enabled by setting the EHF bit in any one or more Header Control Registers. When this bit for any header byte is enabled, and a Compare Header Operation is performed, the HFASM Error will be generated if certain conditions are met. The Error is generated if a header byte pattern register matches with its disk header byte, and that header byte had its EHF bit, and any other byte in the header fails to match. If multiple header bytes have been enabled only one need to match, while *any* other header byte does not match in order to generate an HFASM error. The other header bytes may or may not have their EHF bit set. Thus, this error can tell the system when a particular type of header has been found, even though the exact header did not match.

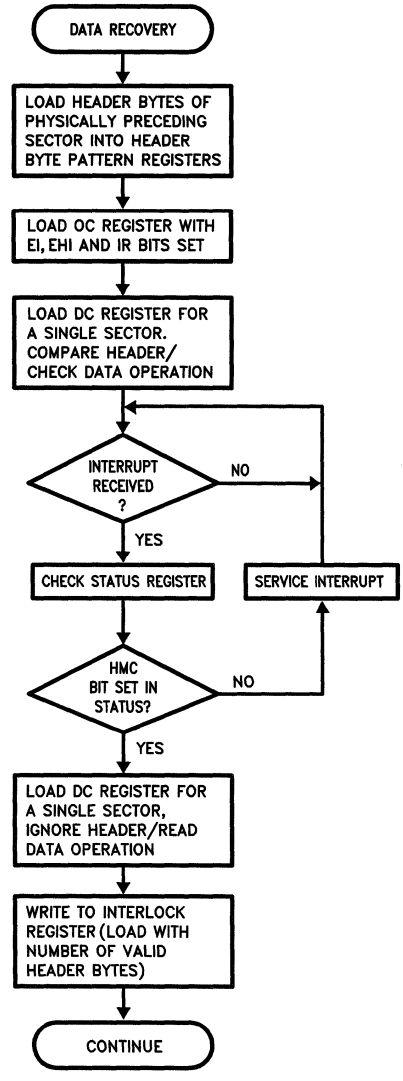
The HFASM error is generated only when execute a command that has a Compare header operation. Write Header, Read Header and Ignore Header operations will not generate an HFASM error. If a Compare Header Operation, and a Check Data Operation form the command is executed and an HFASM error is generated, no data is transferred to the system, but the DDC will load the Header into the FIFO. If a multi-sector operation was in progress the HFASM Error being set will terminate the operation. If the HASM Error is set, but the sector has a CRC error as well, the DDC will terminate the command with both bits set.

This command can be used for various tasks. For example, if the sector's sector number byte has the EHF bit set for all read and write operations, no HFASM error should occur. If one does occur then the system knows that the correct sector number was found, and the reason the correct sector was not found was because of a seek error (head on wrong track); the header was marked bad (and the DDC is looking only for good headers); the wrong head was selected; or some other header parameter was incorrect.

Another example suppose that the header byte that is designated as the sector's sector number has its EHF bit set, as before. The system wants to find sector one, but does not know the other header information. If a Compare Header-Check Data Command is executed, an HFASM error the FIFO will be loaded with the actual header that has a sector number of one. The system can then determine what the status/flag information is (is it a bad sector or a good one etc?) or which cylinder the head is on.

A third possibility, would be to find specific sectors that might have their flag header bytes indicating a bad sector. The EHF bit of the header byte designated for the flag should have its EHF bit set. In this case if a Compare header-Check data is performed using the interlock mode and starting on the sector, all the bad sectors can be identified.

In general the HFASM function is a subtle but powerful tool to enable some diagnostics, and provides to a limited degree the ability for a user definable error condition.



TL/F/8663–G2

FIGURE 7.23. Data Recovery Using Interlock Feature

APPENDIX: DISK DATA CONTROLLER COMMAND FLOW CHARTS

The design guide has covered the general ways that the DDC can execute the various operations. However, it is impossible to account for all possible design situations in the guide. To attempt to provide information which the designer can utilize to determine how the controller might behave in various situations, the figures in this section outline the command flow for the disk operations. All header and data operations are covered, and DMA operations are excluded except to the extent that information must be loaded/unloaded from the FIFO to ensure no overflow or underflow occurs. Other than this DMA operation occurs independently (and concurrently) to the actual disk operations.

The command flow that the DDC executes is divided into header and data operations. In addition to this there are several operations that are performed concurrently, these are shown in the flow charts as special sections and are labelled as concurrent operations. *Figures A.1* to *A.13* cover the header operation and *Figures A.14* to *A.18* cover the data operations. These are described below.

A.1 START OF COMMAND

After a reset, the DDC is in the standby mode waiting for a command. In this mode the DDC is waiting for a command, and once a command is loaded the controller starts its operation. The first major task is to start the DMA transfer to the FIFO if a write data operation is desired. This will start a concurrent task for the DMA controller that will be executed throughout the write data operation. Note that this is shown in *Figure A.2* as a separate dotted outlined block. Otherwise the rest of *Figure A.1* sets up for the header operation, including when to start this operation.

A.2 WRITE HEADER OPERATION

In *Figure A.3* a Write Header operation is assumed. The first test is whether the data output is to be MFM or NRZ. If MFM, the precompensation may be enabled if NRZ, *Figure A.4* is executed. The first steps taken by the DDC are to write the preamble, address mark bytes, and the sync bytes if desired. Also, if the CRC/ECC is to encapsulate both header and sync fields than it is turned on, otherwise it will be turned on later.

Figure A.4, A.5, and *A.6* complete the write header operation. First the CRC/ECC is enabled, then the header bytes are written. There are various options on how to do this and they are shown in this figure. Finally the CRC/ECC fields are written to the disk. This includes the internal one, and/or an external field if desired. The last task is to write the header postamble field.

At entry point D, a standard end to the header operation is shown. This is used by other header operations. Here the header match bit is set and if interrupts are enabled, the Header completion interrupt is issued. If the operation is a multi-sector operation, then the Number of Sector Operations counter and the Start Sector register are updated. If the last operation then the DDC is ready for the next command and will start the data operation.

A.3 NON-WRITE HEADER OPERATION

For a read or ignore header operation, the entry point in *Figure A.7* is H. First the operation waits to start based on the internally programmed mode and the receipt of an index or sector pulse. Then Read Gate is asserted, and the DDC searches for the sync or address mark fields. Once all the sync and address mark fields match, the DDC can perform the Header operation. *Figure A.5.* If encapsulation of the CRC/ECC was enabled then CRC/ECC calculation will begin at the address mark field.

If not already active, the CRC/ECC is activated, and the DDC starts operating on the header information. If the data operation is a check data or the header operation is a read header (the later is shown in *Figure A.11*), then the header bytes are loaded into the FIFO. If the operation is a compare header, the header bytes are compared to the registers or start sector register. Next the CRC/ECC is checked. If it is in error then the header fault flag is set, however the operation is not aborted. If there is a CRC/ECC error or the header did not match, the DDC then looks for the next sector. If the correct header is found then, the DDC will deassert read gate to avoid the write splice, and will proceed to execute a data operation.

A.4 WRITE DATA OPERATION

For a write data operation, the write gate is asserted, and an algorithm similar to the write header operation is executed. MFM or NRZ data output is configured, and the start with address mark mode is checked. Address mark, preamble and sync fields are written *Figure A.14.* If needed the CRC/ECC generator is enabled when the Address Mark is written.

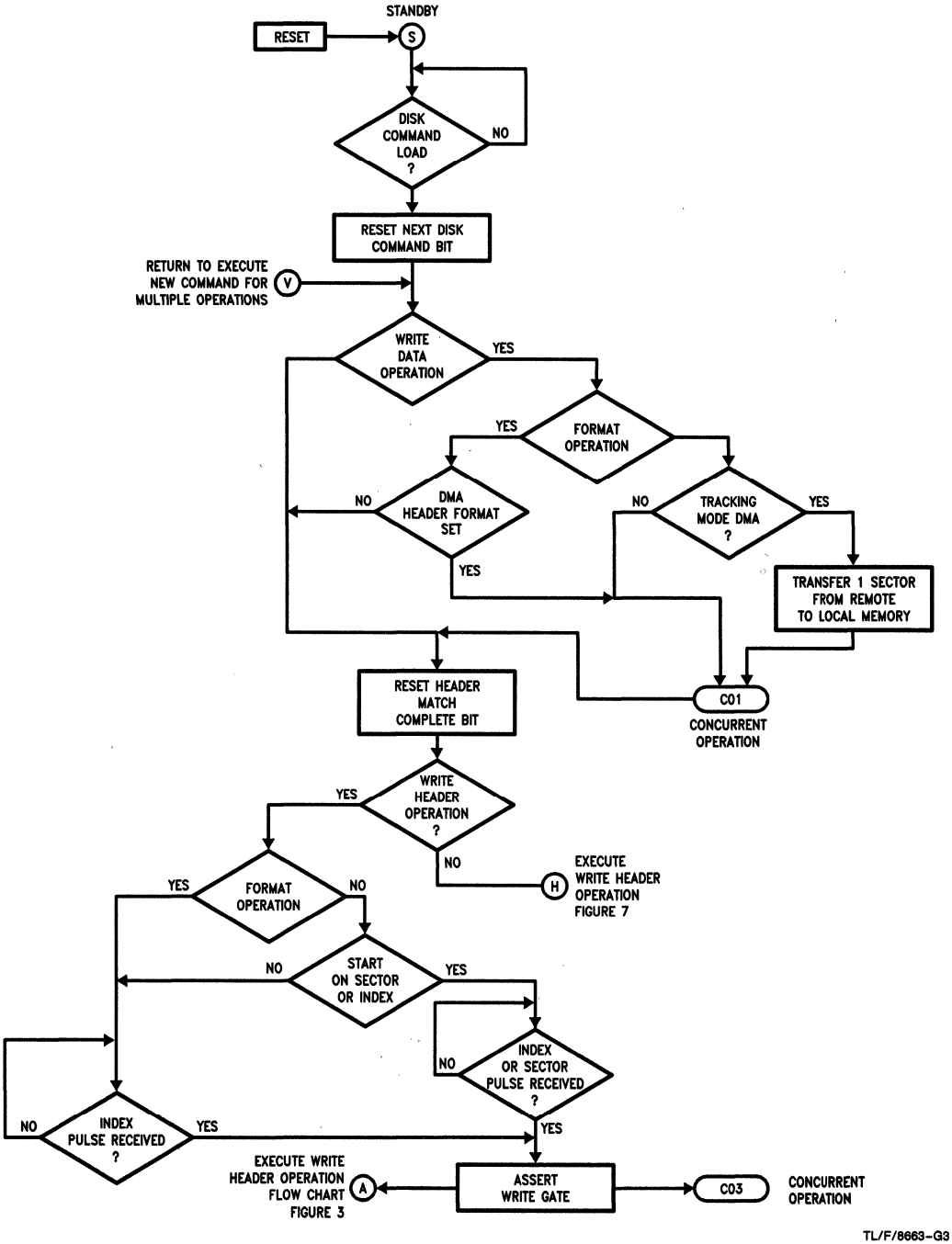
Just prior to writing data, the CRC/ECC will be enabled, if it is not already. Then data is written to the disk from the FIFO or the Format pattern register. Once the data is written, the CRC/ECC is written, followed by the data postamble. If the operation is not a format operation then write gate is deasserted, and interrupts may be generated. If the operation is a format operation, the post sector gap is written, as shown in *Figure A.16.*

The operation then is completed and the DDC will check to see if the command was a multisector one or if a new command has been entered. If so then a new command is started immediately, and this is shown by jumping to entry point V in *Figure A.1.*

A.5 NON-WRITE DATA OPERATION

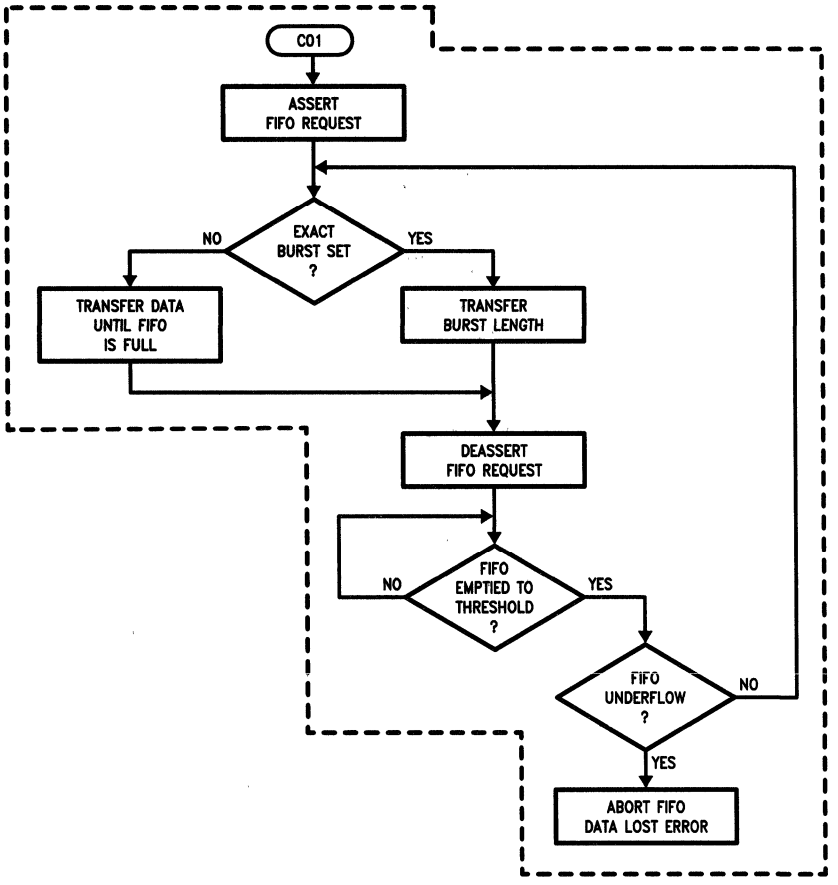
The flow chart for a read data or check data operation is shown in *Figure A.17.* After the header postamble, read gate is asserted. The DDC first configures when the CRC/ECC calculation should begin. It then begins checking the address mark and sync fields. If an error occurs the operation is aborted. Otherwise if the operation is a read data, the data is sent to the FIFO, and the CRC/ECC is checked.

If the operation is a check data command, the data field is not transferred or checked, and the DDC just counts this field and checks the CRC/ECC. At exit point U on *Figure A.18* the end of the data operation jumps to *Figure A.16,* where the interrupts may be generated if enabled and the DDC checks for another command.

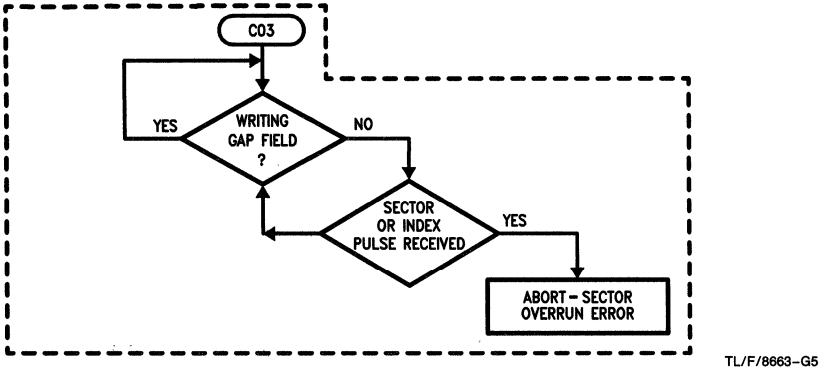


TL/F/8663–G3

FIGURE A.1. Flow Chart for Start of a Command

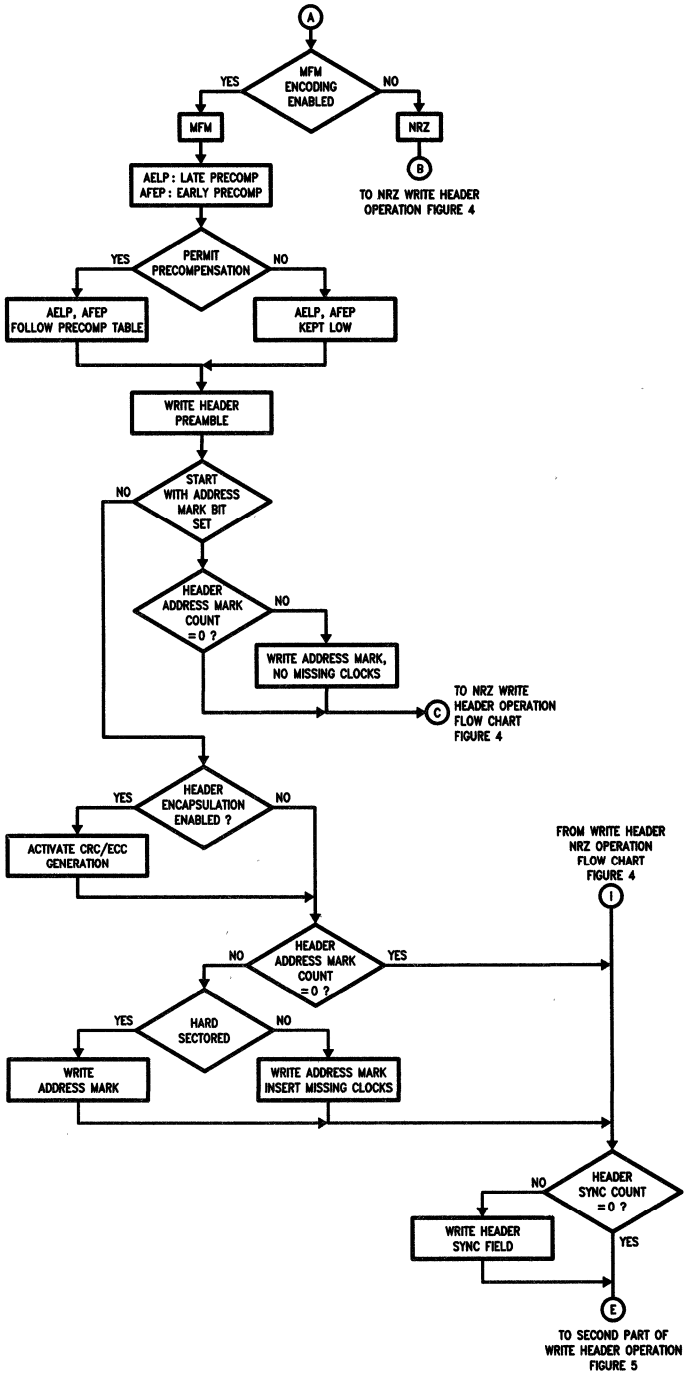


(a) Concurrent Operation 1



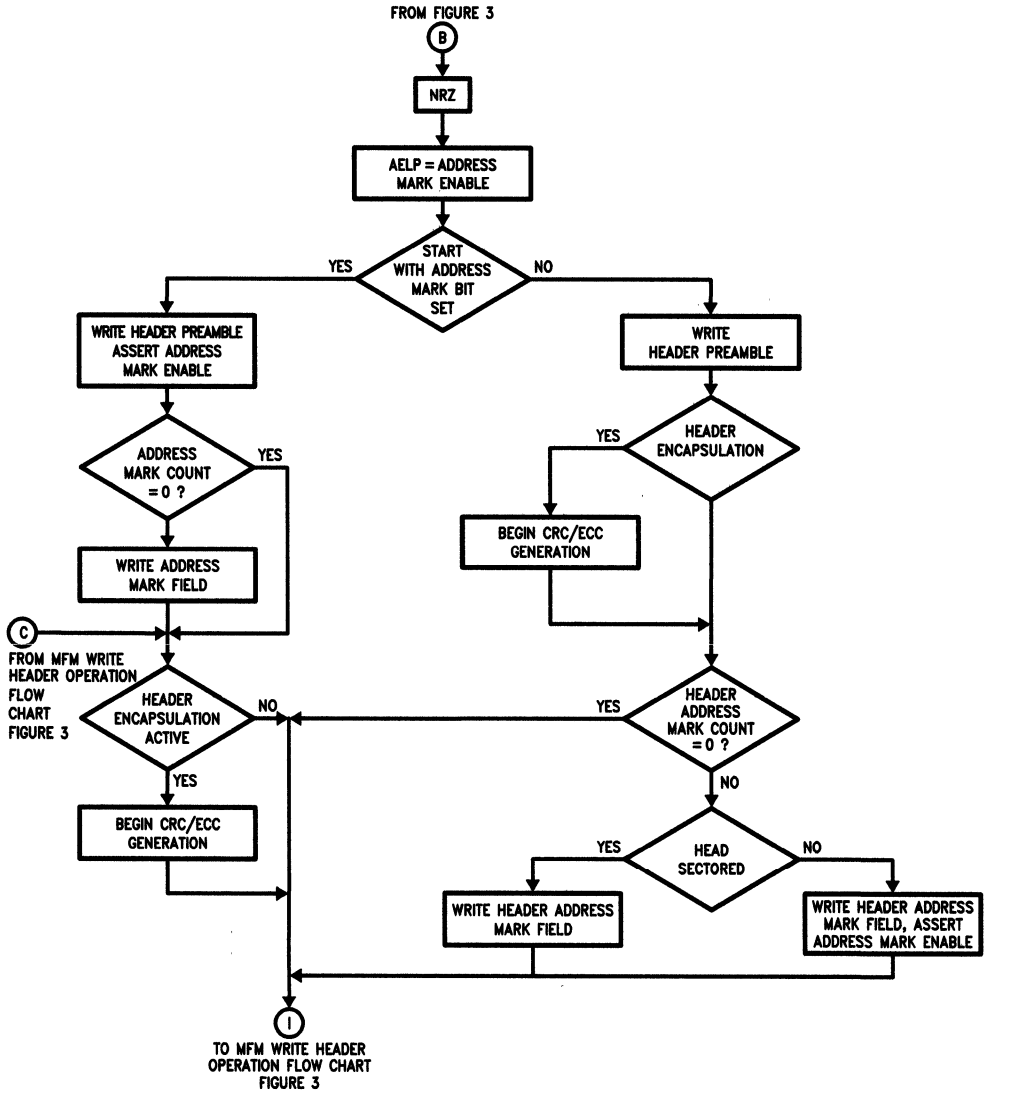
(b) Concurrent Operation 3

FIGURE A.2. Concurrent Operations for the Start of a Command



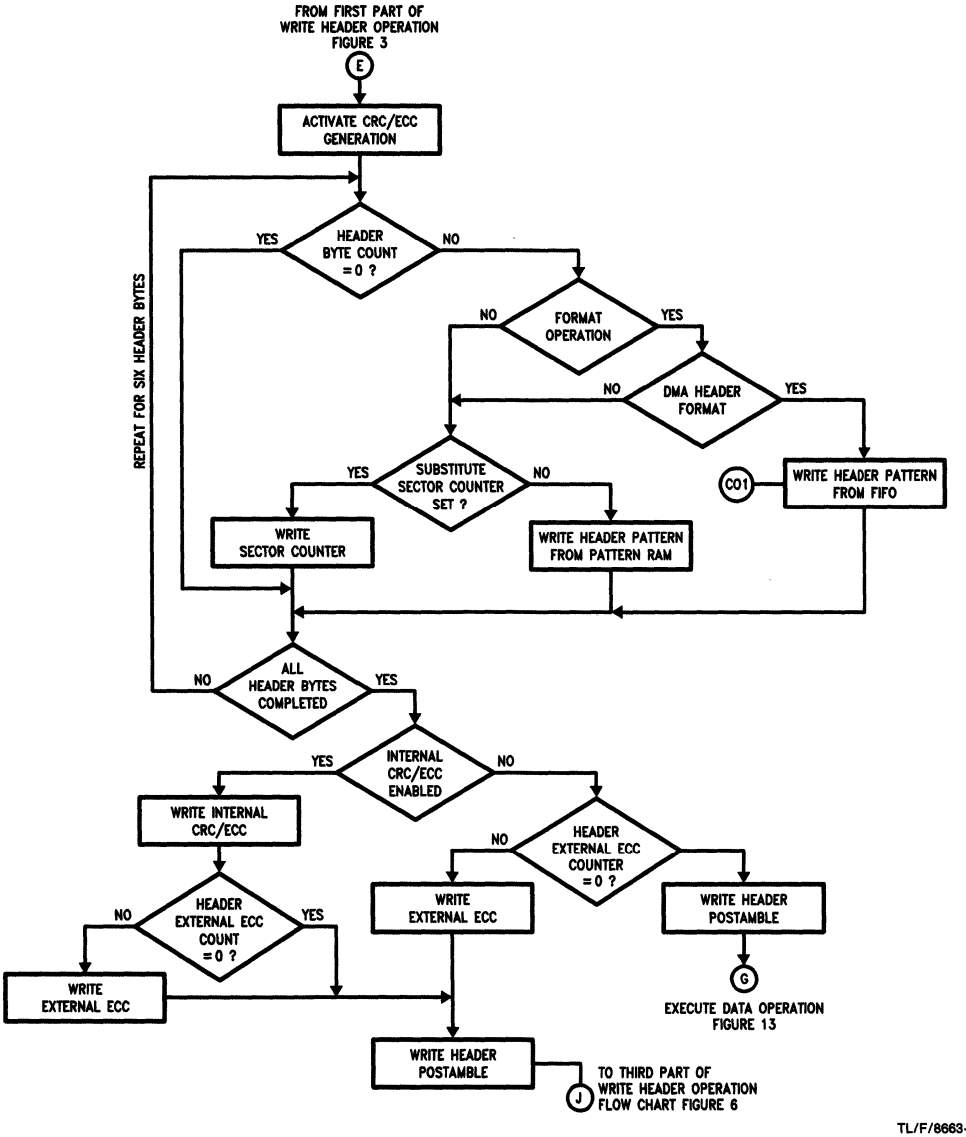
TL/F/8663–G6

FIGURE A.3. Flow Chart for First Half of Write Header Operation



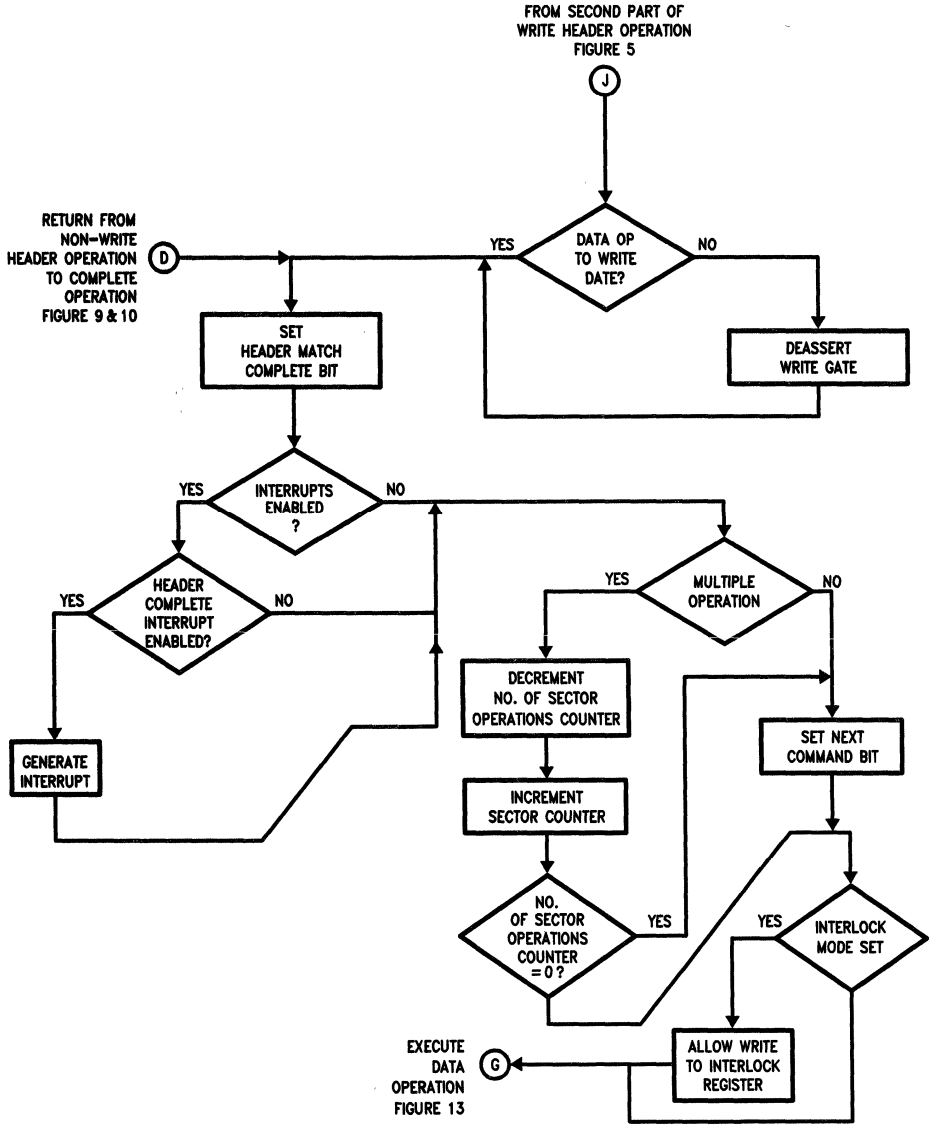
TL/F/8663–G7

FIGURE A.4. Flow Chart for First Half of Write Header NRZ Operation



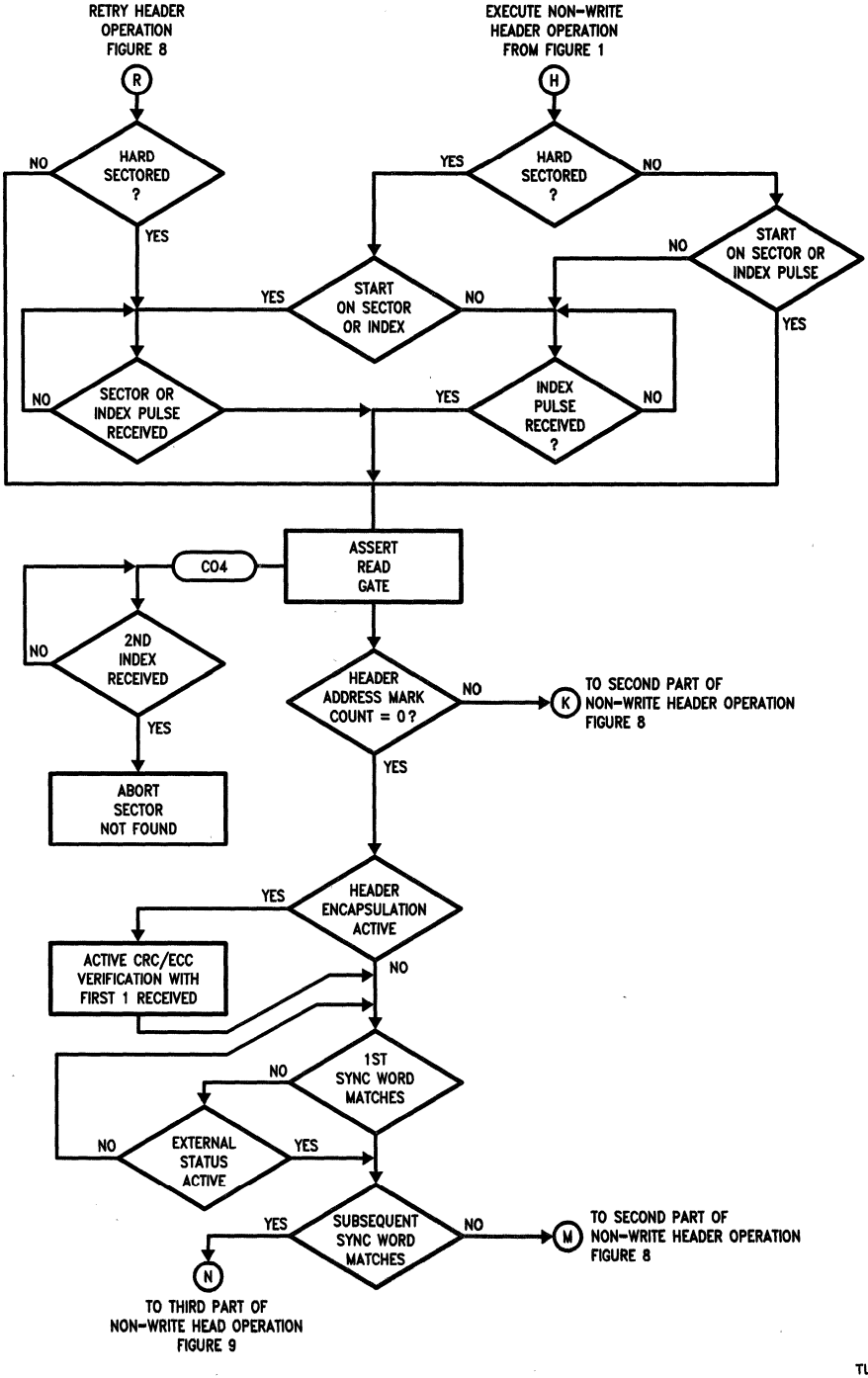
TL/F/8663–G8

FIGURE A.5. Flow Chart for Second Part of a Write Header Operation



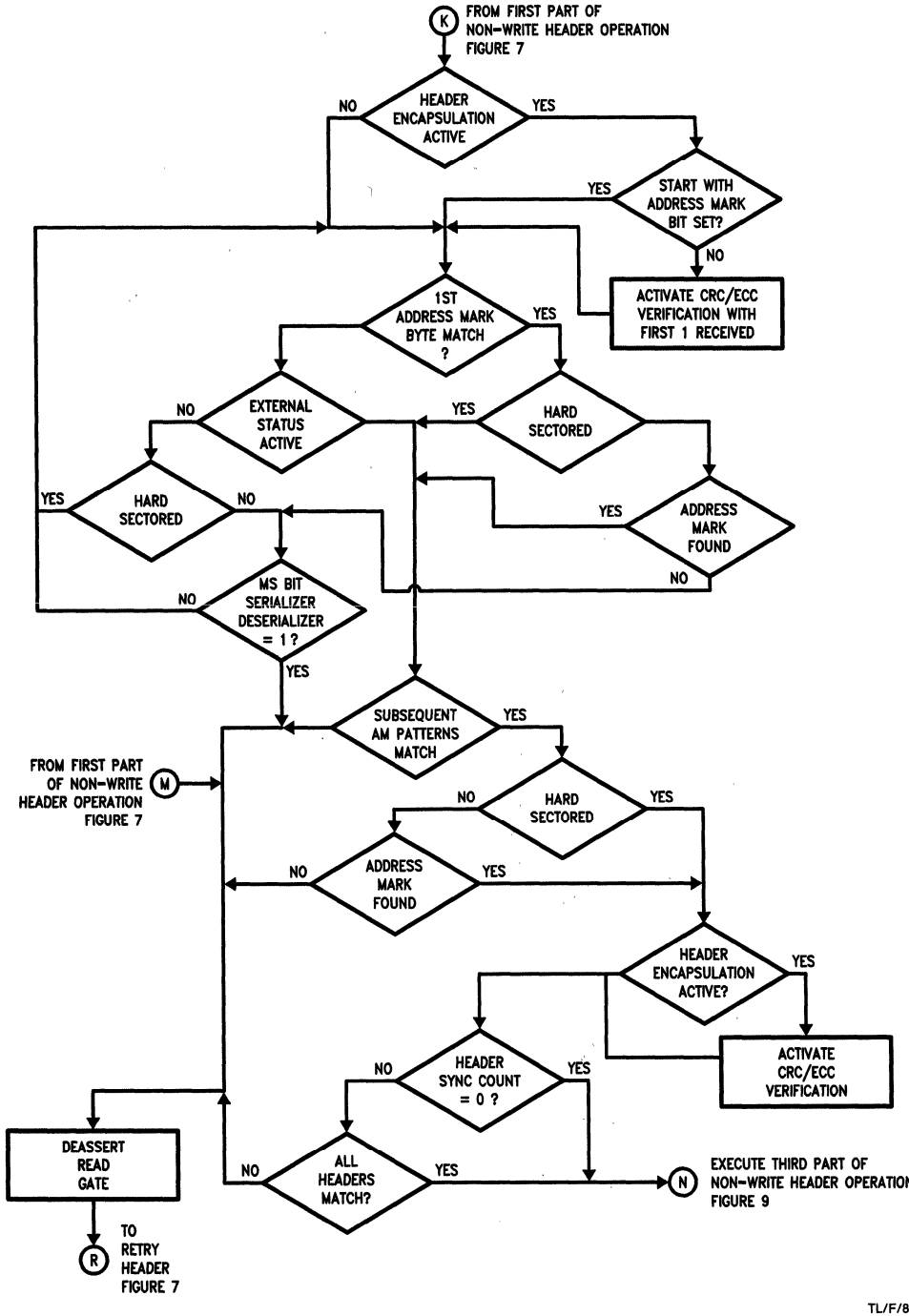
TL/F/8663–G9

FIGURE A.6. Flow Chart for Third Part of a Write Header Operation and Ending Sequence for Other Header Operations



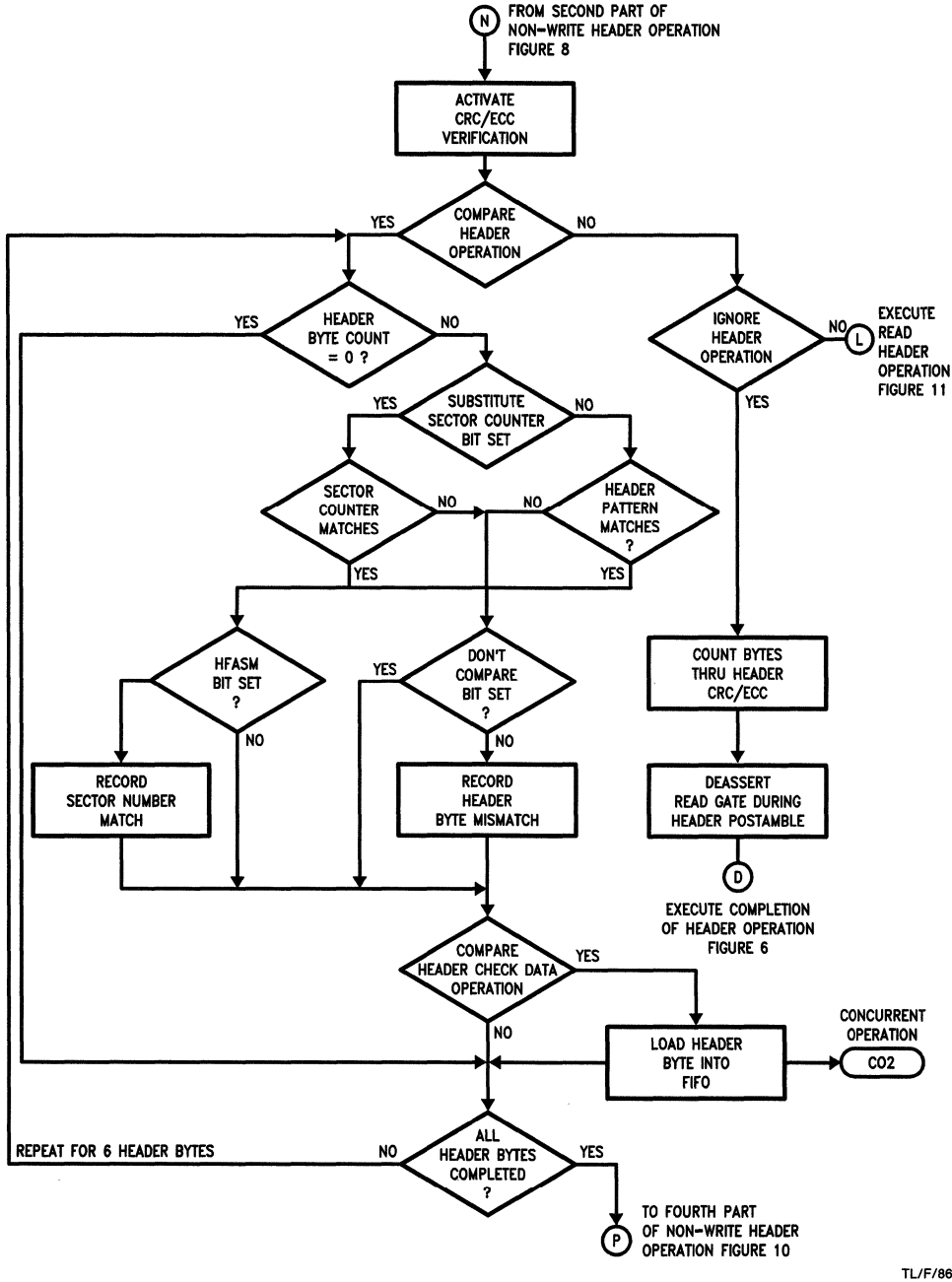
TL/F/8663–H0

FIGURE A.7. Flow Chart for First Part of Non-Write Header Operation and Retry Header



TL/F/8663–H1

FIGURE A.8. Flow Chart for Second Part of Non-Write Header Operation



TL/F/8663–H2

FIGURE A.9. Flow Chart for Third Part of Non-Write Header Operation (Except Compare Header)

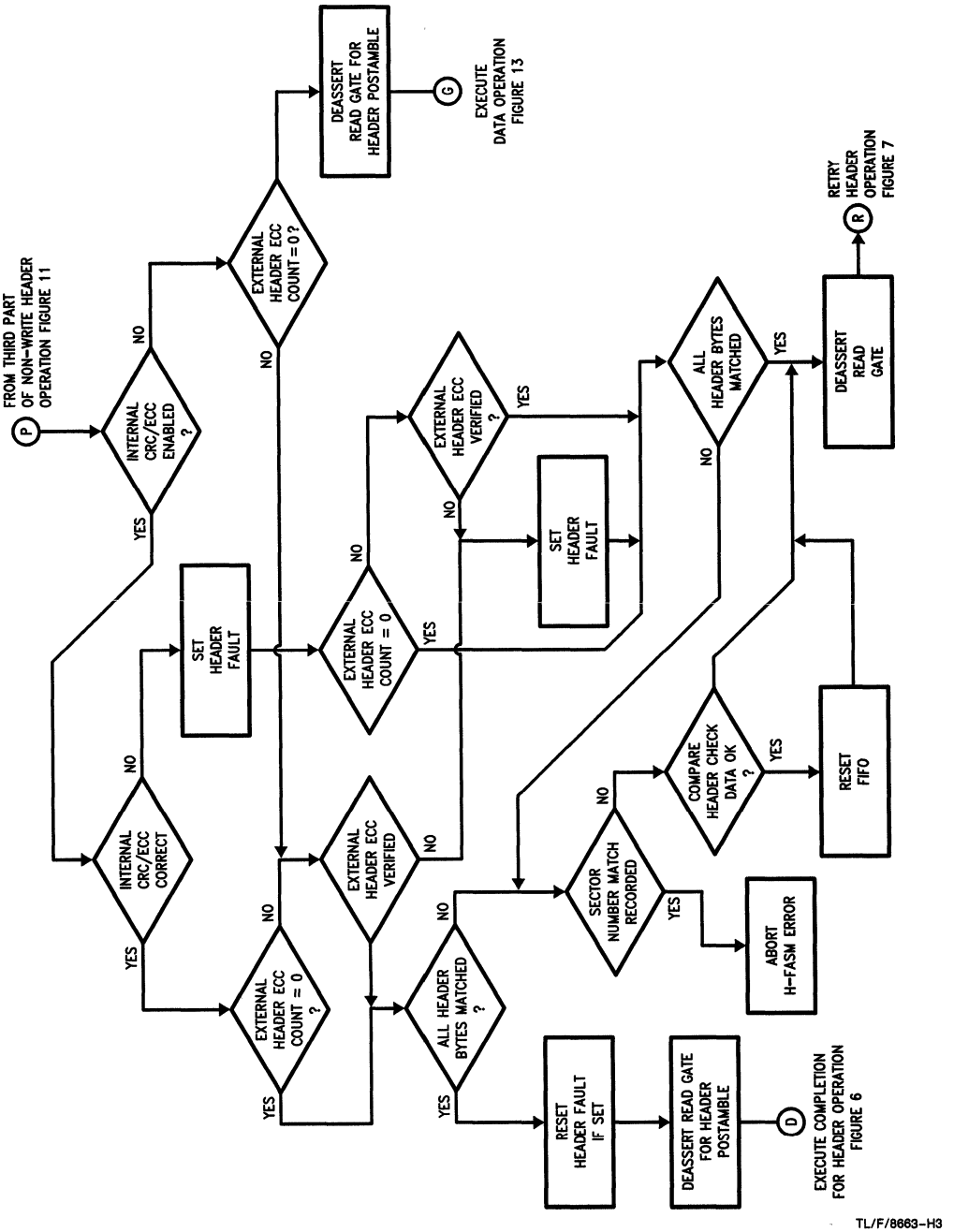
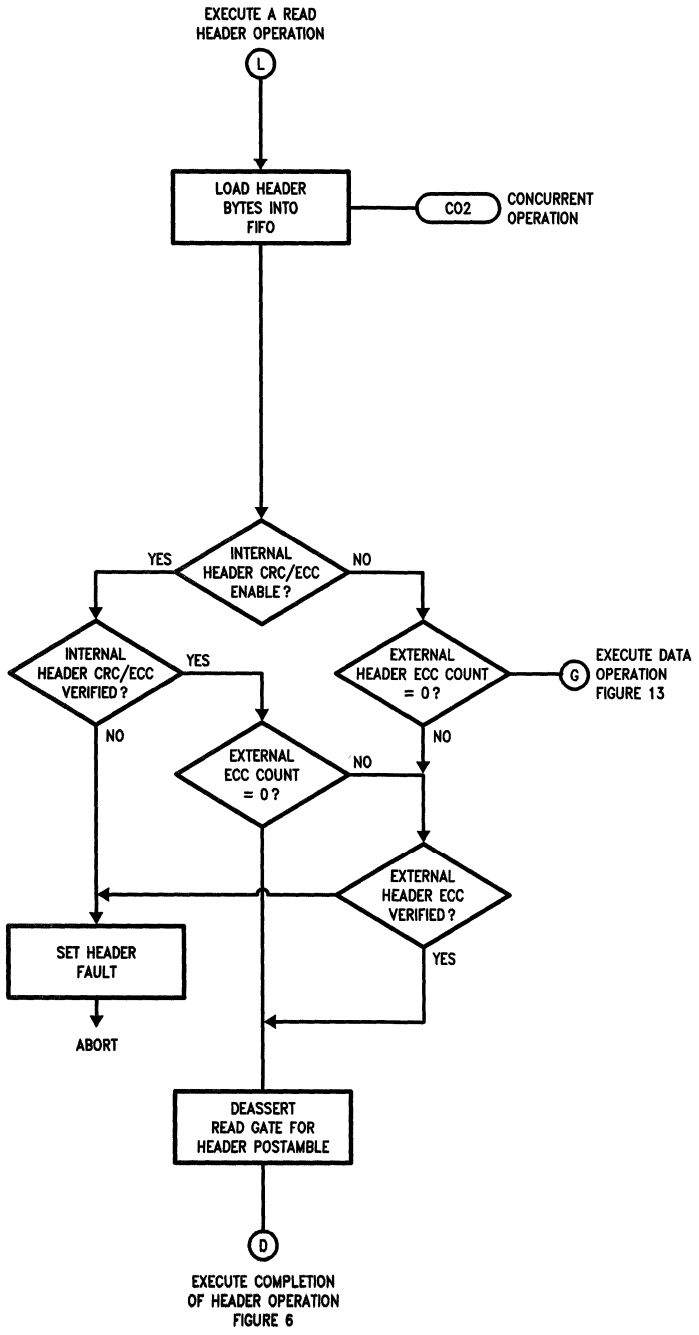
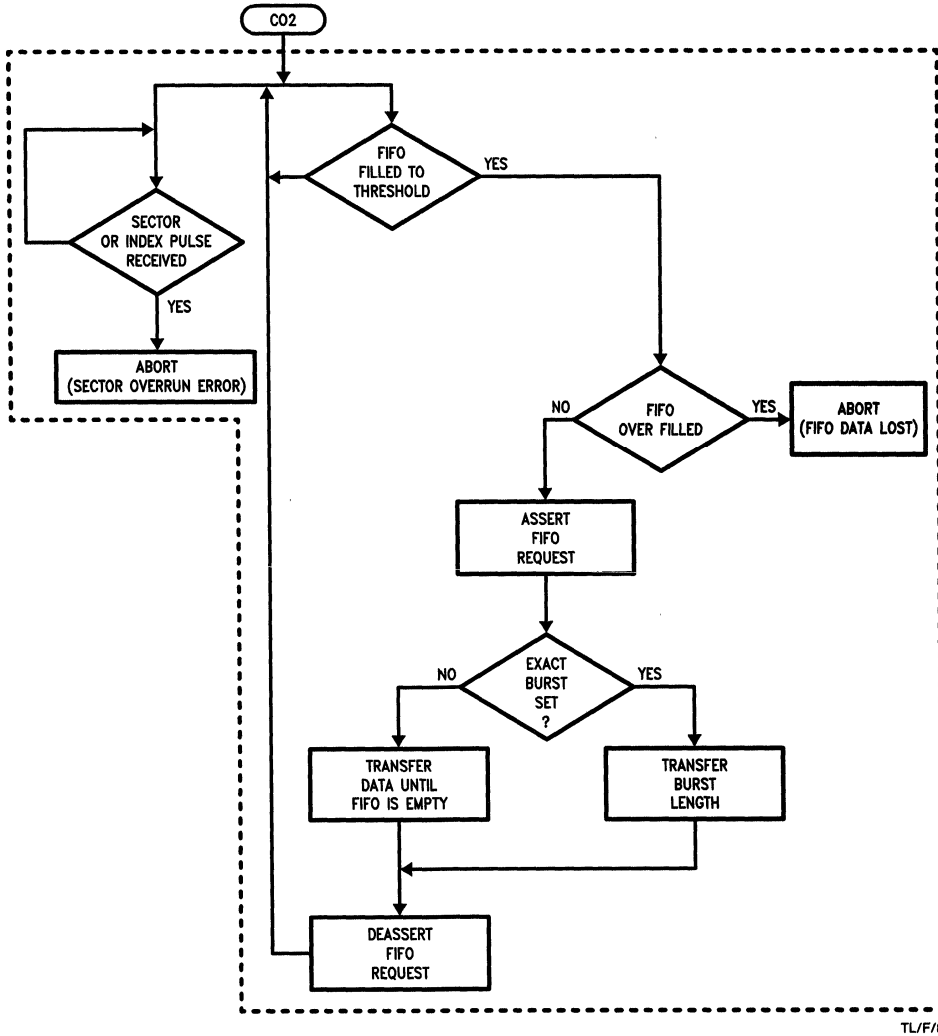


FIGURE A.10. Flow Chart for Fourth Part of Non-Write Header Operation (Except Compare Header)



TL/F/8663–H4

FIGURE A.11. Flow Chart for Compare Header Operation



TL/F/8663–H5

FIGURE A.12. Flow Chart for Compare Header Concurrent Operation

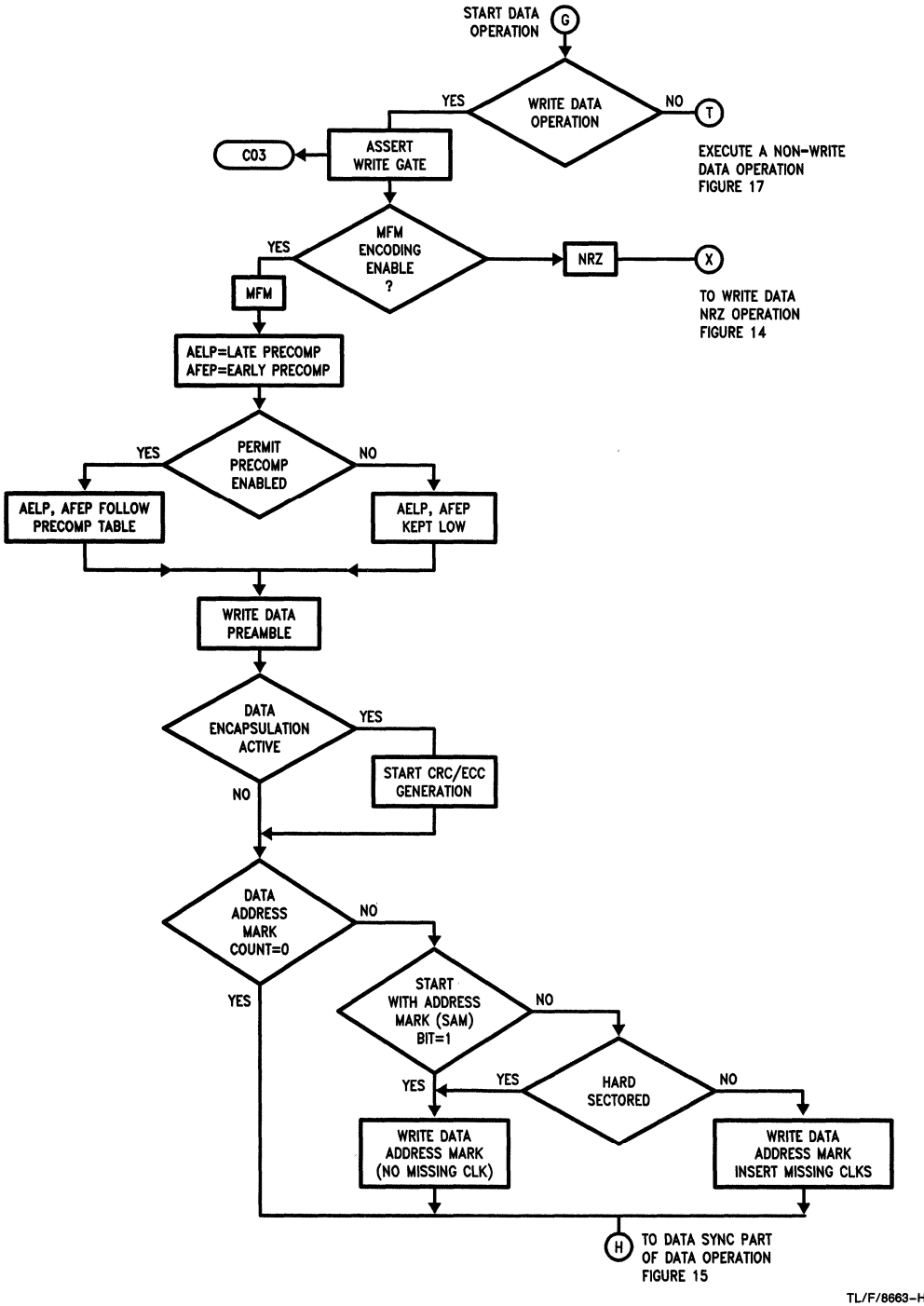


FIGURE A.13. First Part of Data Operation Flow Chart, MFM Mode (Up to Data Sync Field)

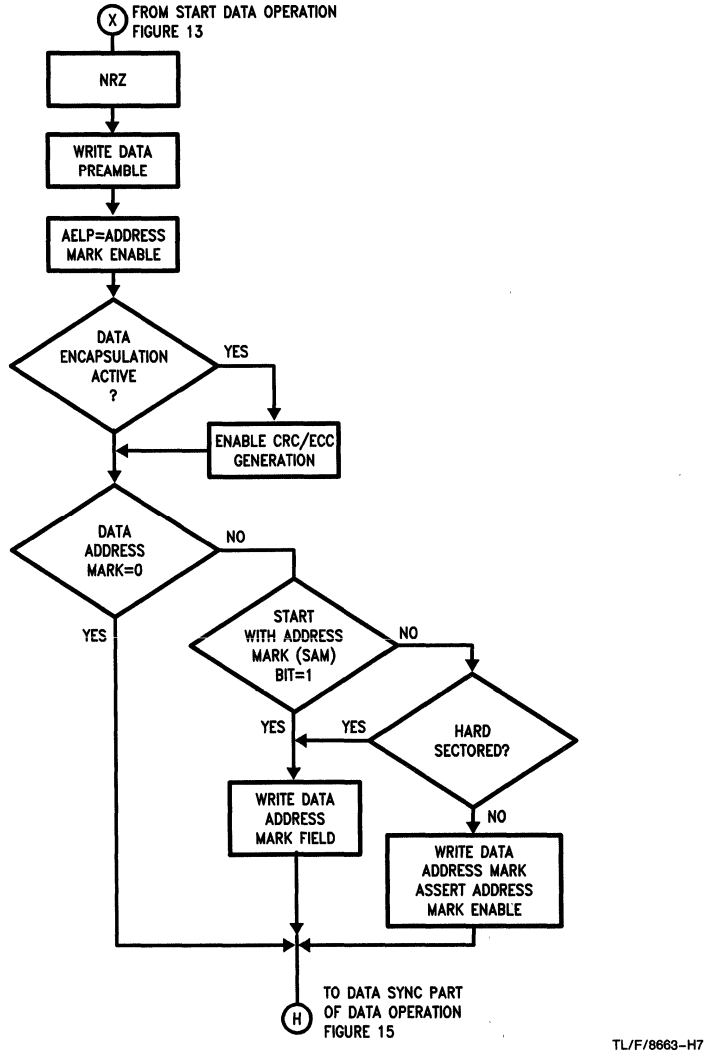
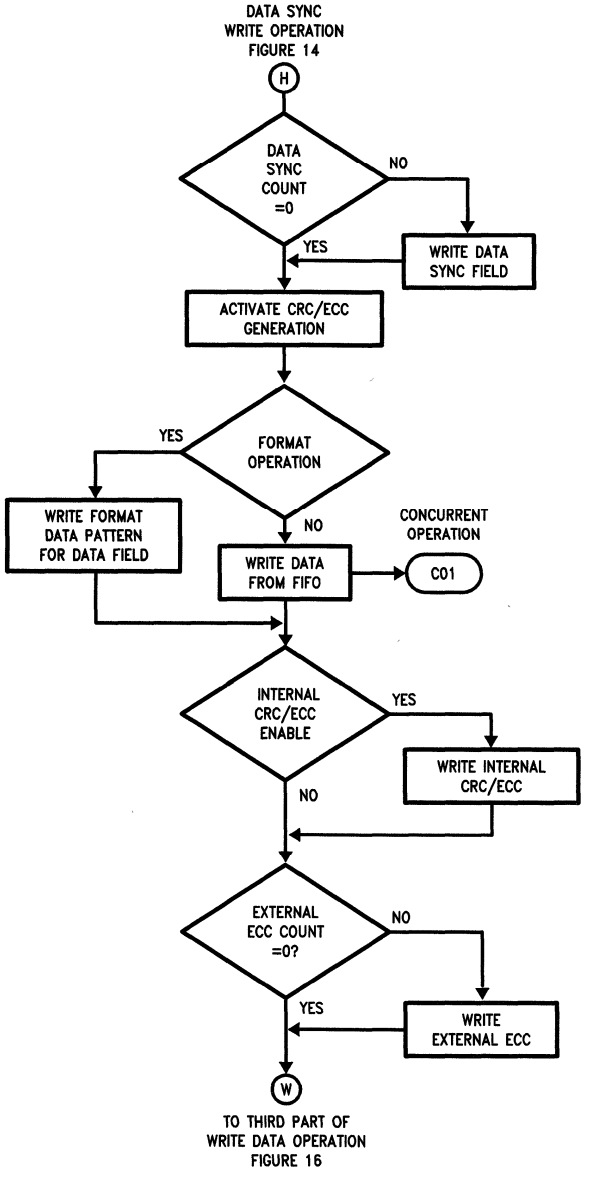
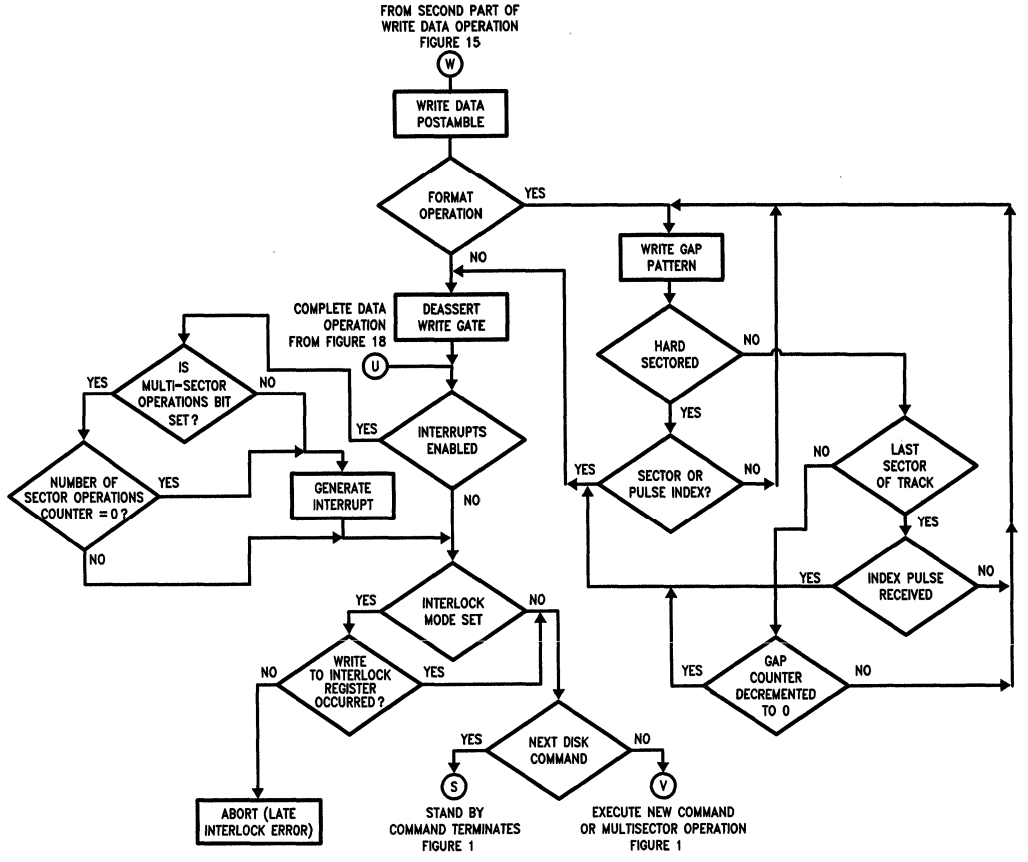


FIGURE A.14. First Part of Data Operation Flowchart, NRZ Mode (Up to Data Sync Field)



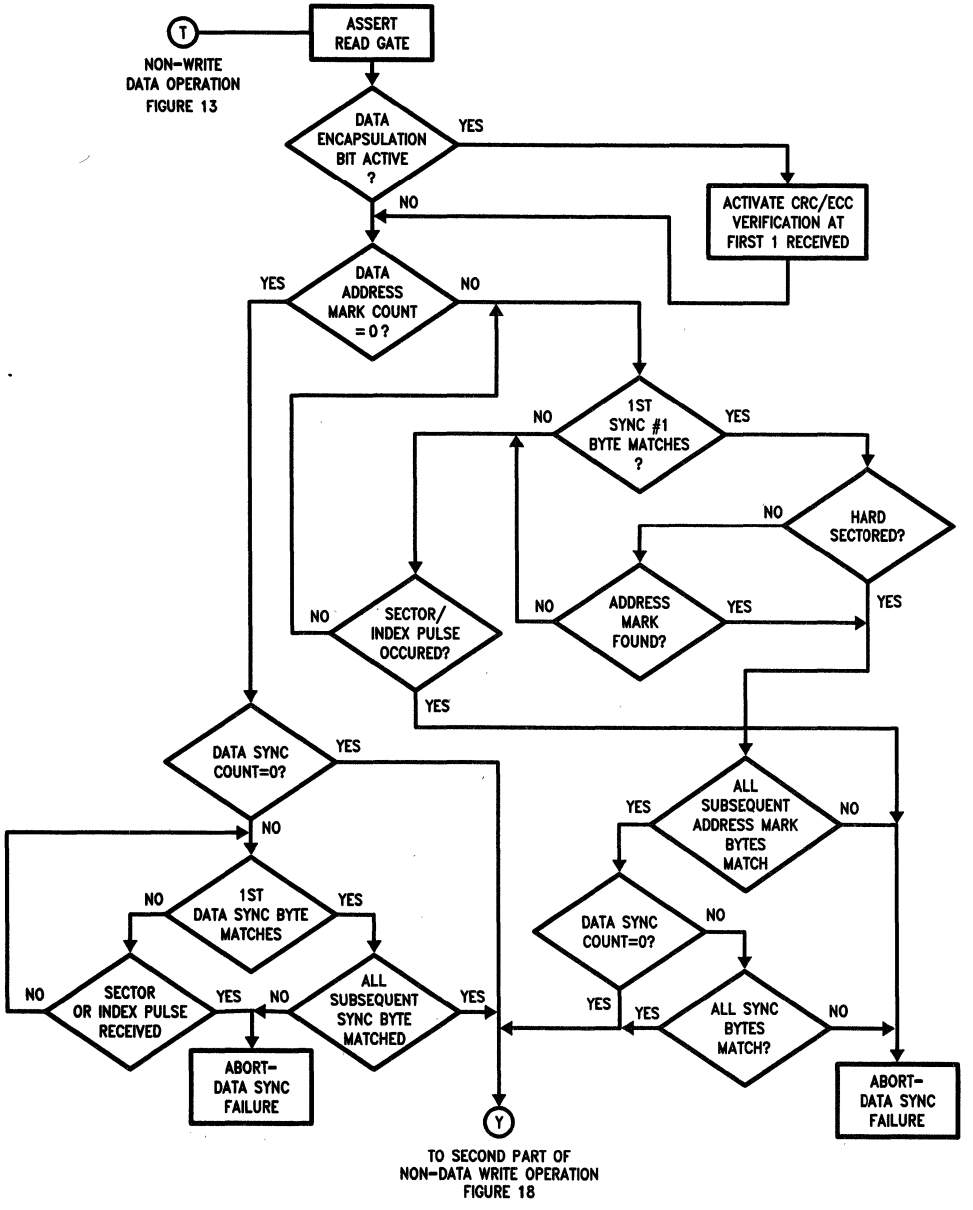
TL/F/8663–H8

FIGURE A.15. Second Part of Data Operation (for Write Data Operation)



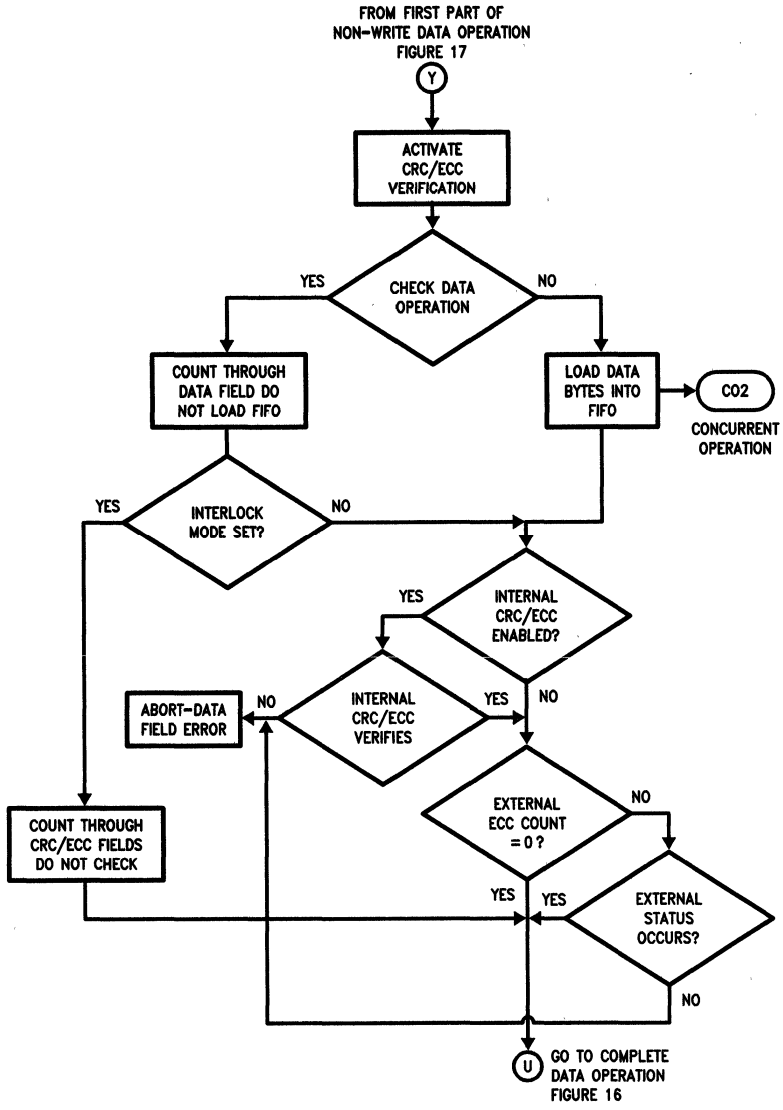
TL/F/8663–H9

FIGURE A.16. Flow Chart for Third Part of Write Data Operation



TL/F/8663-10

FIGURE A.17. First Part of Data Operation Flow Chart for Non-Write Operation



TL/F/8663–I1

FIGURE A.18. Second Part of Data Operation Flowchart for Non-Write Operation

Section 10

Physical Dimensions/ Appendices

Section 10 Contents

National Semiconductor

TapePak®

The latest generation in VLSI packaging, TapePak is the package of the future—low-cost, reliable, high-leadcount packaging that's easy to handle, easy to test, and easy to mount. It's also compatible with existing surface-mount technology.

TapePak uses tape-automated bonding technology and a unique, patented outer ring to protect the leads and, at the same time, provide an effective test interface.

This outer ring is molded at the same time as the body of the package and creates test points outside the package leads. The test ring is discarded along with the tape as the package is excised by the automatic pick-and-place machine at the point of assembly.

During testing, the leads themselves never come in contact with the test socket, so lead damage and coplanarity problems are eliminated. The test ring also allows burn-in to be performed on each device.

Not only does this ring protect the leads during handling, testing and assembly, but it also allows leads to be placed on centers of 0.012 inch–0.020 inch (0.3 mm–0.5 mm), while the test points are placed on standard centers of 0.020 inch (0.5 mm), 0.025 inch (0.65 mm) or 0.050 inch (1.27 mm). That way, the test points are compatible with existing automatic test equipment.

TapePak packages are significantly smaller than conventional and alternative surface-mount packages. TapePak lead counts range from 40 to greater than 360, yet the largest package measures only 1.1 inches (28 mm) square.

Comparison of TapePak and Conventional Packages

	40L DIP	44L PLCC	40L TapePak
Lead thickness (mils)	10.0	10.0	2.8
Lead pitch (mils)	100	50	20
Package length (mils)	2050	650	350
Package width (mils)	600	650	350
Package thickness (mils)*	175	180	71
Volume ratio	24.4	9.1	1

	40L DIP		44L PLCC		40L TapePak	
	Long	Short	Long	Short	Long	Short
Lead length (in)	1.0	0.3	0.35	0.25	0.1	0.1
Resistance (mOhm)	7	4	4	3	2.4	2.4
Inductance (nH)	22	6.0	6.5	5	1.2	1.2
Capacitance (pF) (lead to lead)	0.5	0.2	0.3	0.2	0.2	0.1

*Measured from seating plane to the top of the package.

A TapePak device can be less than 1/10 the size of a traditional DIP and 1/3 the size of other surface-mount packages such as a PLCC.

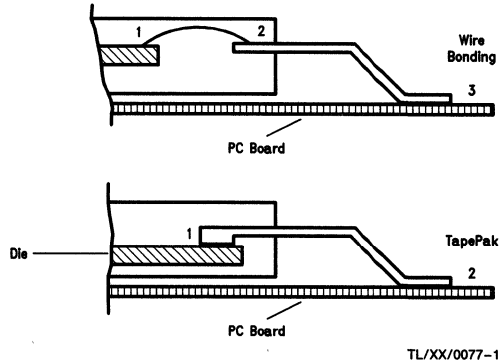
TapePak technology was designed to take full advantage of automatic assembly systems with their high speed and precision. It can be used with existing precision surface-mount assembly equipment with minimal modification. The only requirement is an accessory for removing the test ring and forming the leads at the point of assembly.

TapePak also provides a significant improvement in the electrical characteristics of each package. Lead capacitance and inductance, for example, can be reduced up to ten times that of other packages. Signal propagation time is also reduced, and thermal characteristics are improved. Because of the tremendous space savings, TapePak technology offers much greater power density per unit area as compared to DIP or alternative surface-mount packages.

Performance and reliability are improved because there are one-third fewer connections between the die and the PC board. Low-stress molding compounds also improve package reliability. TapePak devices pass stringent environmental tests, including autoclaving at 121°C at 15 psi and thermal shock from −65°C to +150°C for 1000 cycles.

No other package takes similar advantage of materials technology to provide the combination of low cost, high density, testability, damage resistance, and reliability.

TapePak has been accepted as an industry standard by the Joint Electronic Device and Engineering Council (JEDEC) and registration is in progress with the Electronic Industries Association of Japan (EIAJ). TapePak technology has also been licensed to other manufacturers for their own proprietary devices.



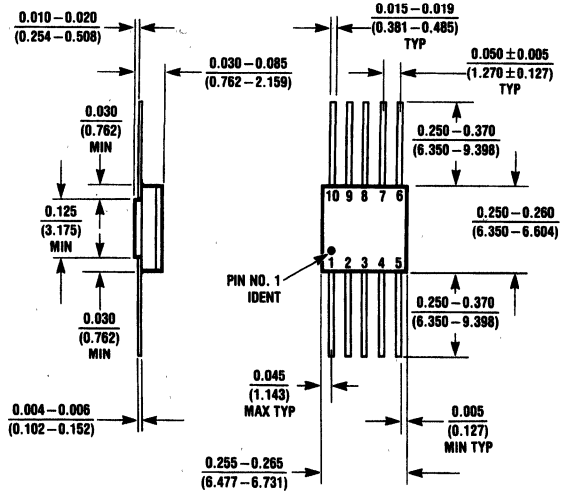
TL/XX/0077–1

With TapePak, there are one-third fewer connections between die and board than with traditional wire bonding.

National Semiconductor

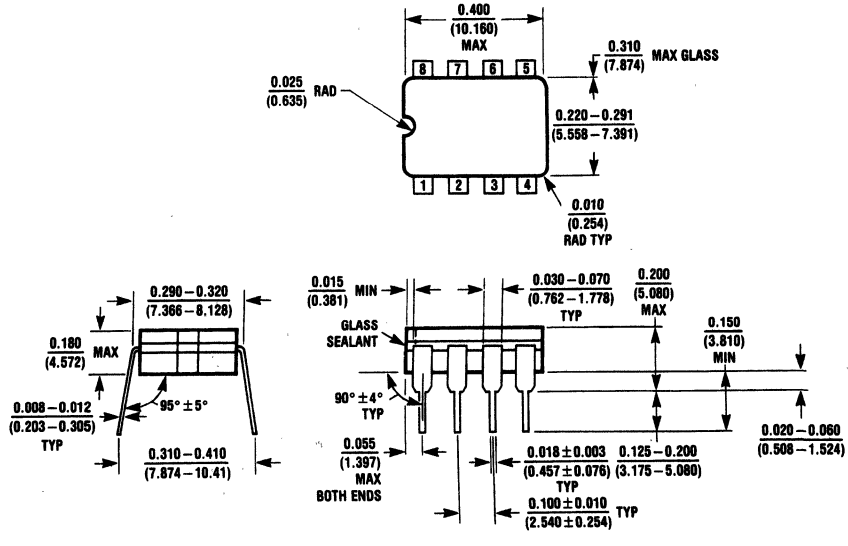
All dimensions are in inches (millimeters)

10 Lead Ceramic Flatpack Package (F)
NS Package Number F10B



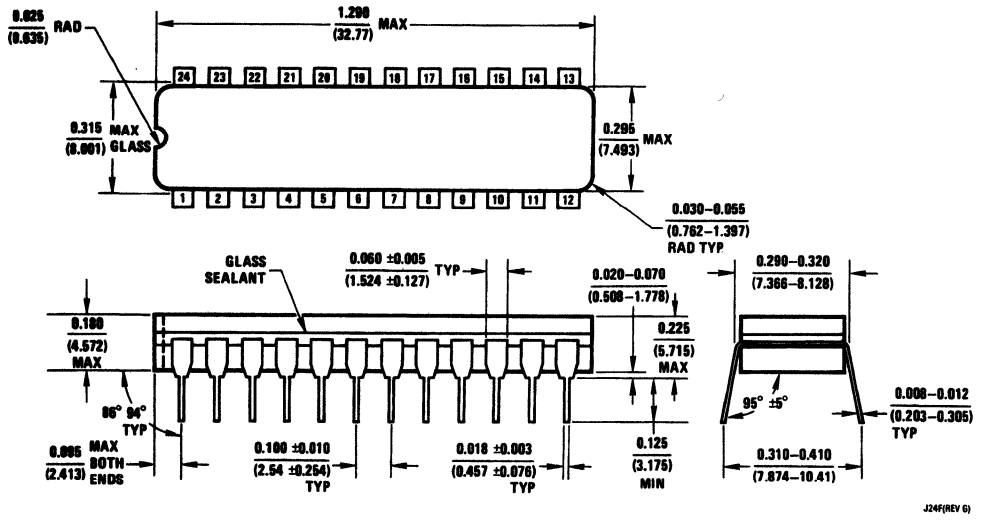
F10B (REV F)

8 Lead Ceramic Dual-In-Line Package (J)
NS Package Number J08A

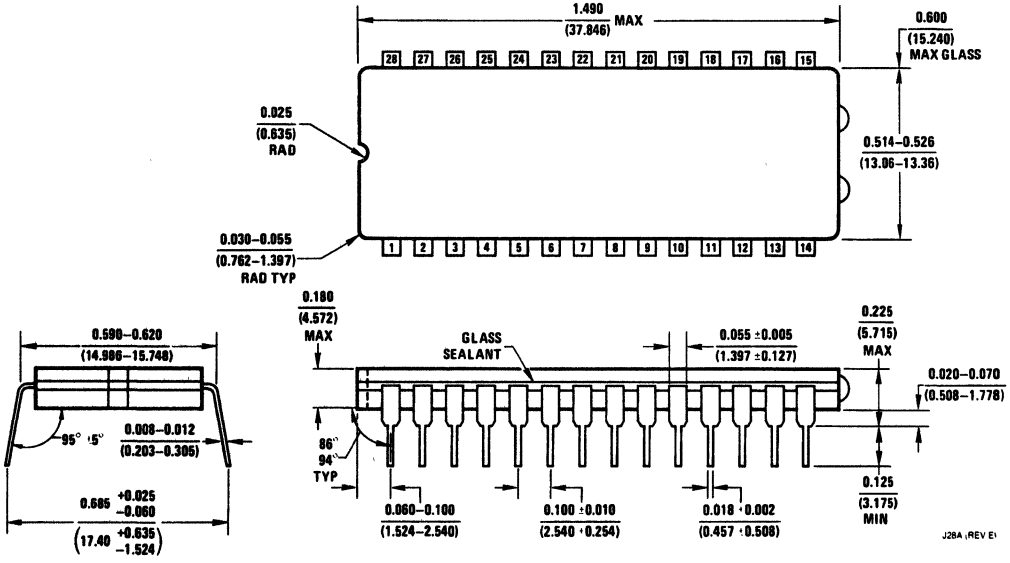


J08A (REV J)

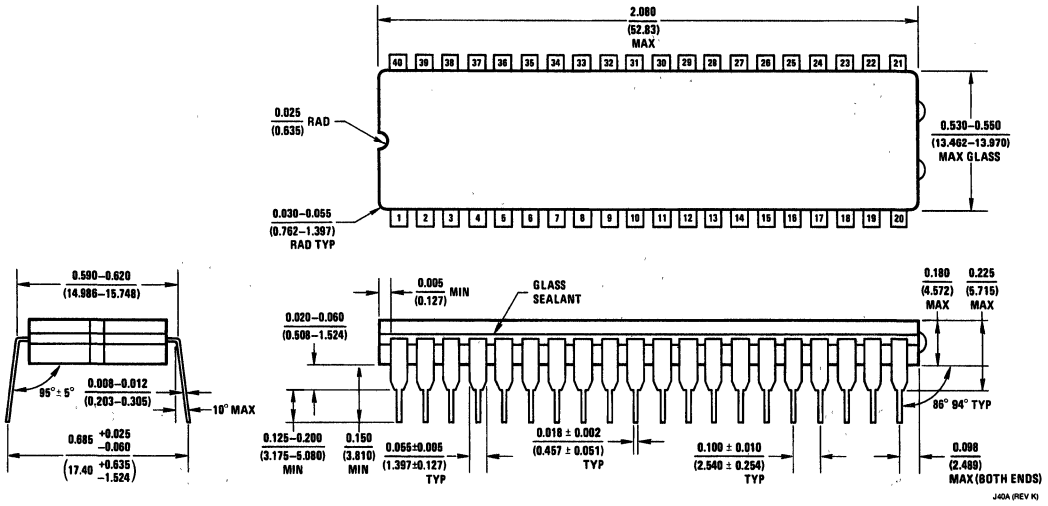
24 Lead Ceramic Dual-In-Line Package (J)
NS Package Number J24F



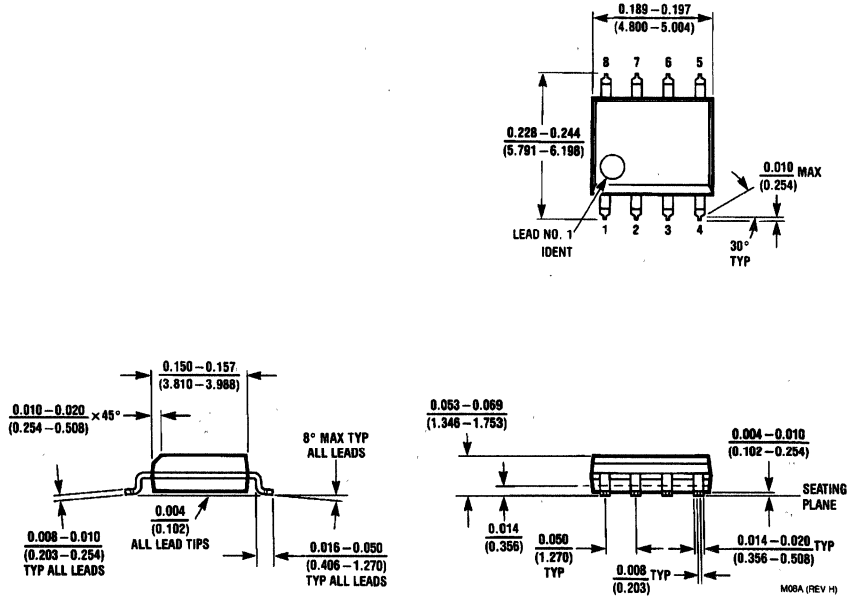
28 Lead Ceramic Dual-In-Line Package (J)
NS Package Number J28A



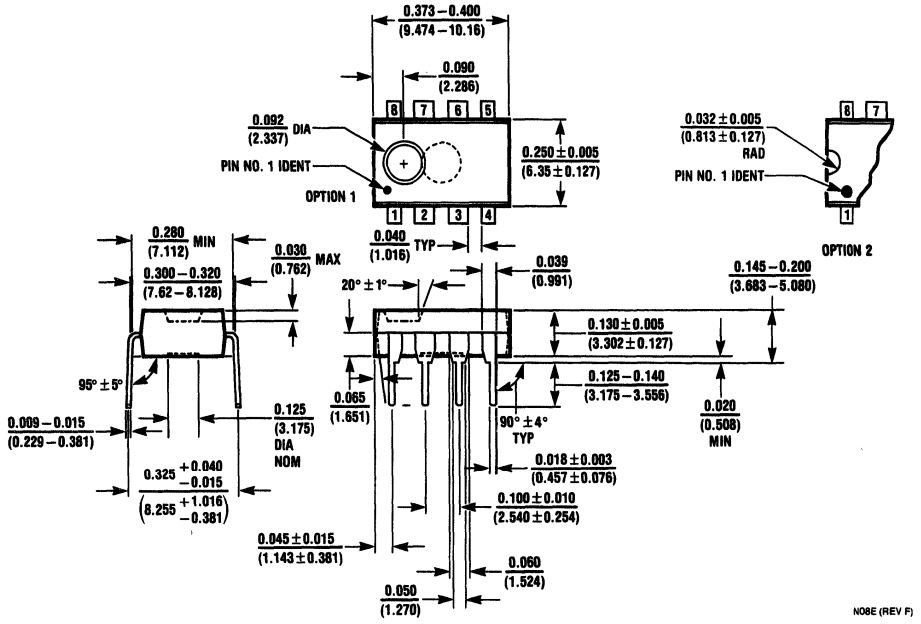
40 Lead Ceramic Dual-In-Line Package (J)
NS Package Number J40A



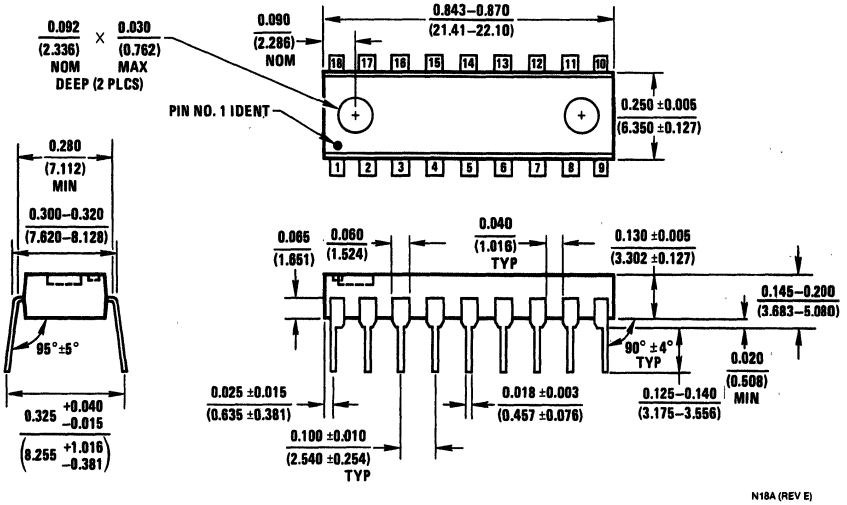
8 Lead (0.150" Wide) Molded Small Outline Package (M)
NS Package Number M08A



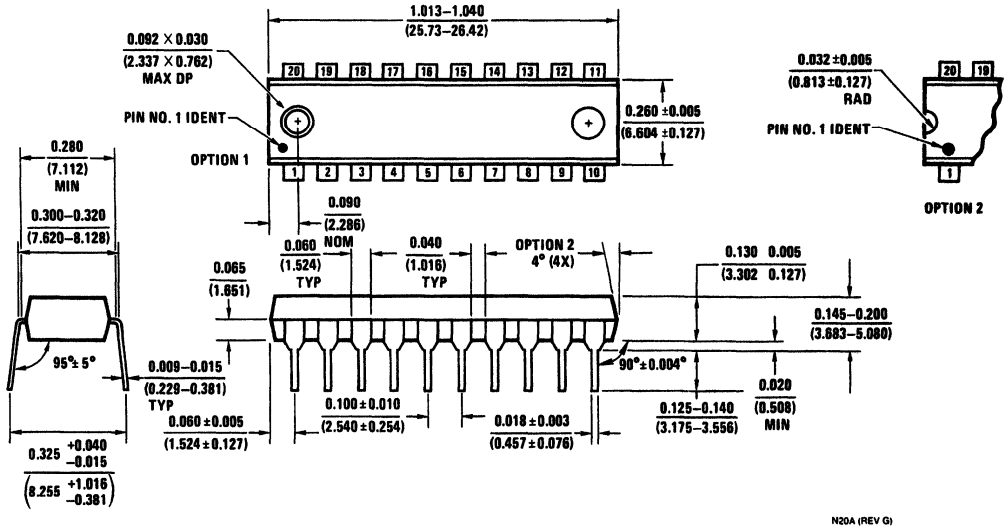
8 Lead Molded Dual-In-Line Package (N)
NS Package Number N08E



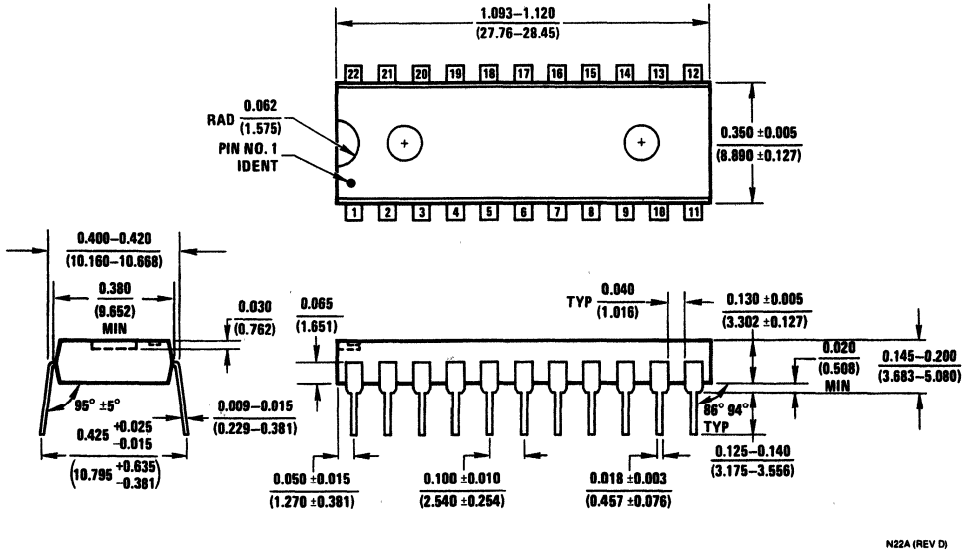
18 Lead Molded Dual-In-Line Package (N)
NS Package Number N18A



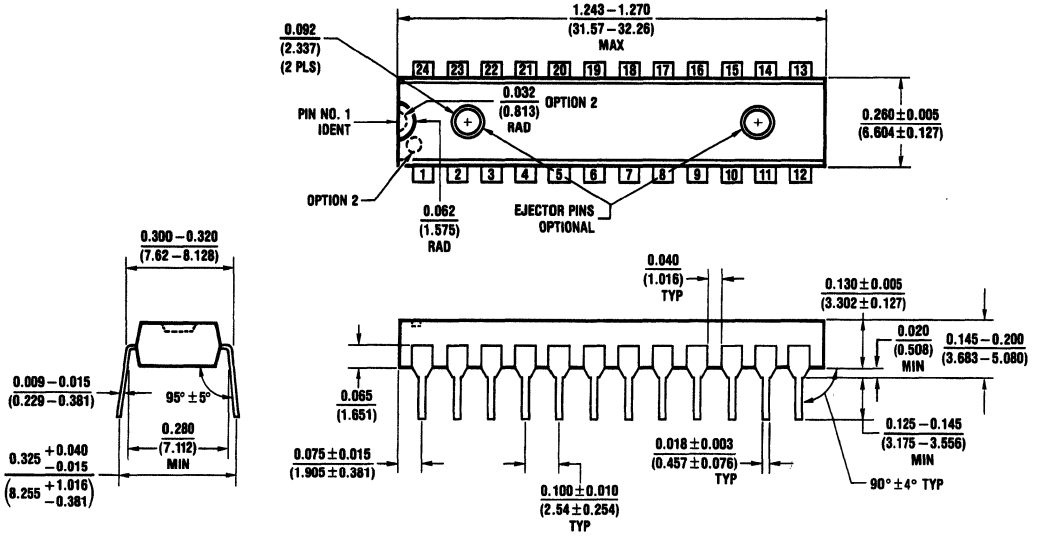
20 Lead Molded Dual-In-Line Package (N)
NS Package Number N20A



22 Lead Molded Dual-In-Line Package (N)
NS Package Number N22A

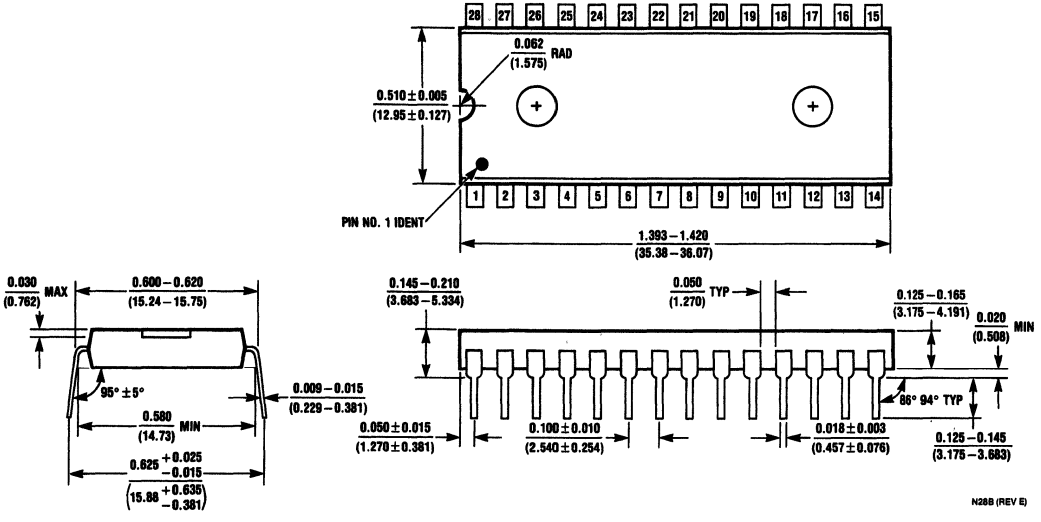


24 Lead Skinny Dual-In-Line Package (0.300" Centers Molded) (N)
NS Package Number N24C



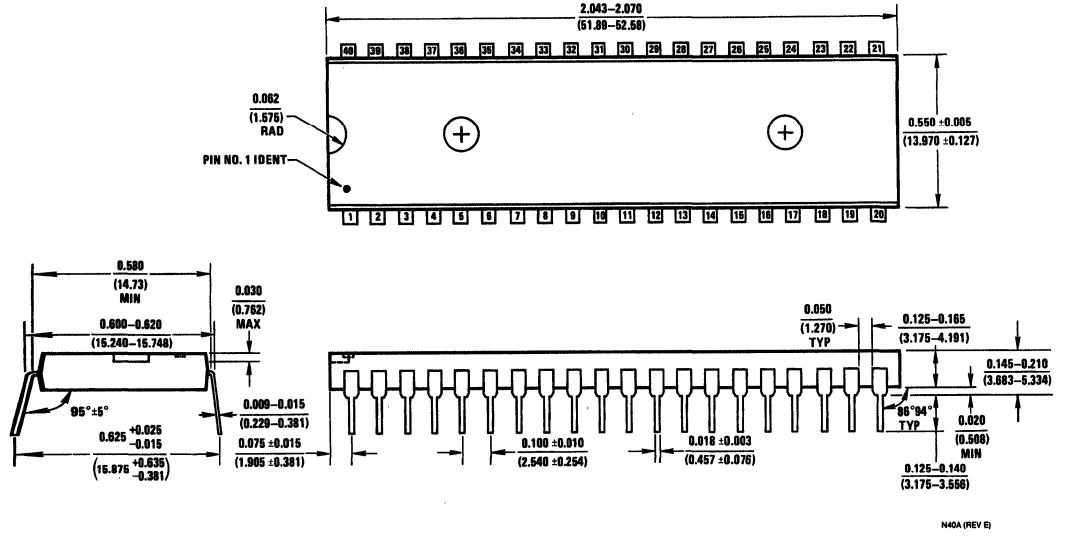
N24C (REV F)

28 Lead Molded Dual-In-Line Package (N)
NS Package Number N28B

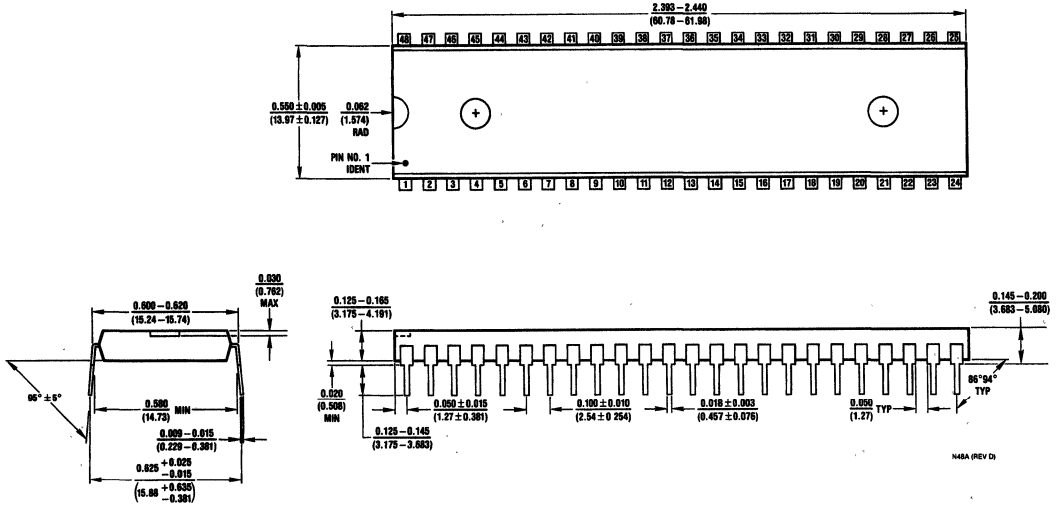


N28B (REV E)

40 Lead Molded Dual-In-Line Package (N)
NS Package Number N40A

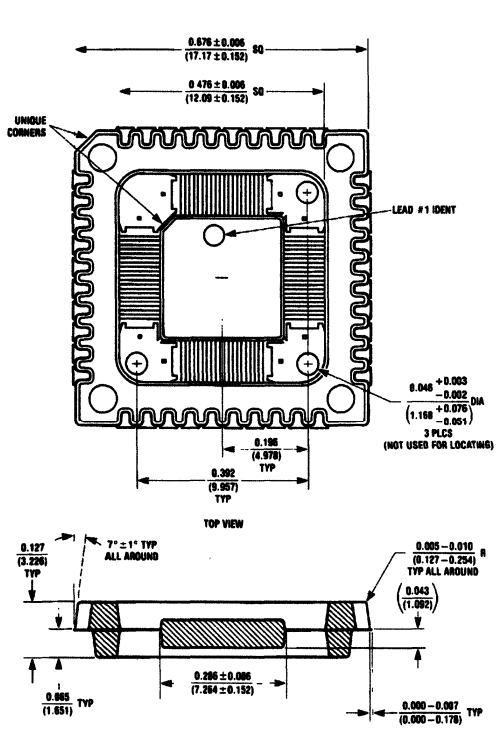


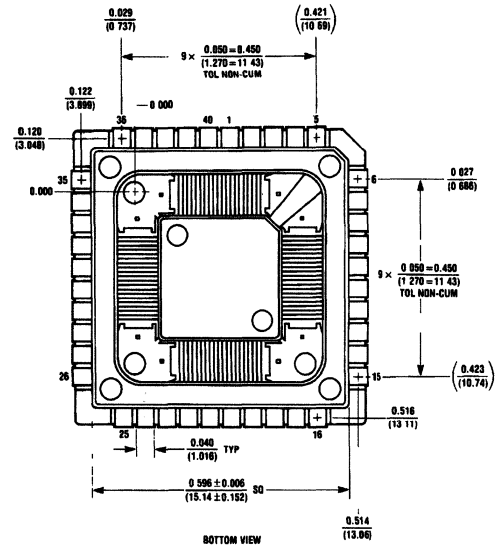
48 Lead Molded Dual-In-Line Package (N)
NS Package Number N48A



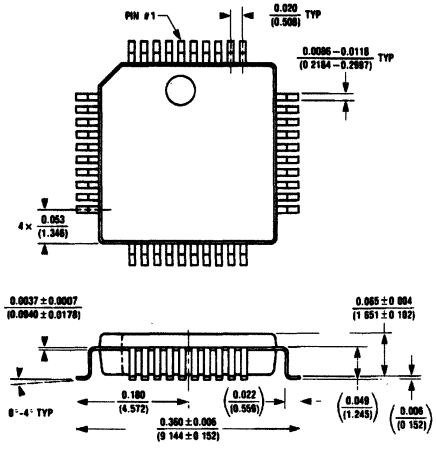
40 Lead TapePak® Package (TP)
NS Package Number TP40A

PACKAGE CONFIGURATION AS SHIPPED

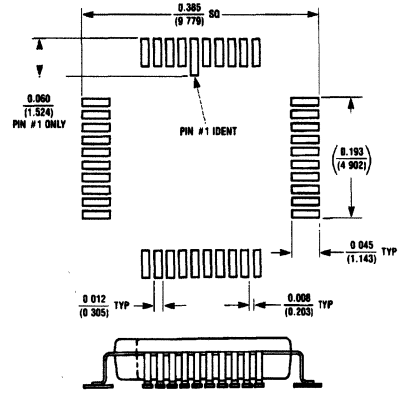




RECOMMENDED FORMED AND EXCISED PACKAGE OUTLINE

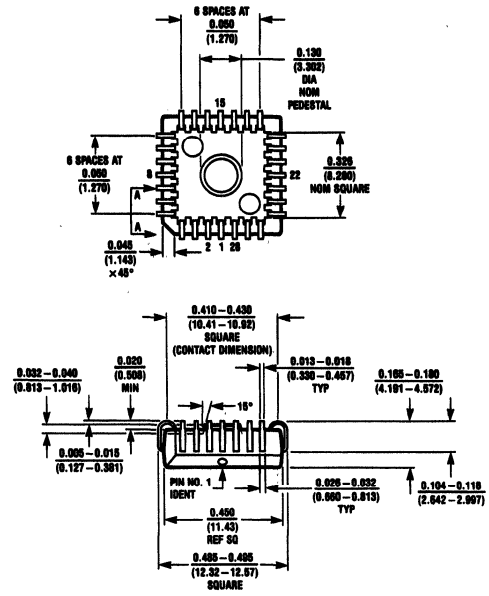


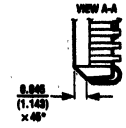
RECOMMENDED FOOTPRINT



TP40A (REV A)

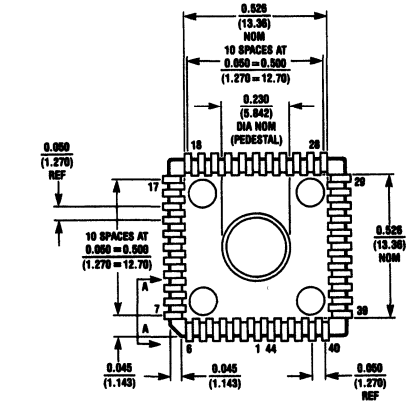
28 Lead Plastic Chip Carrier (V)
NS Package Number V28A

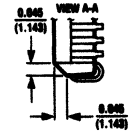


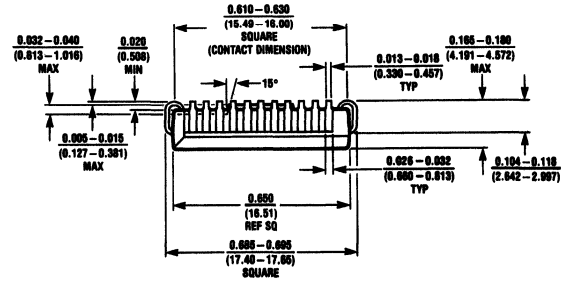


V28A (REV G)

44 Lead Plastic Chip Carrier (V)
NS Package Number V44A

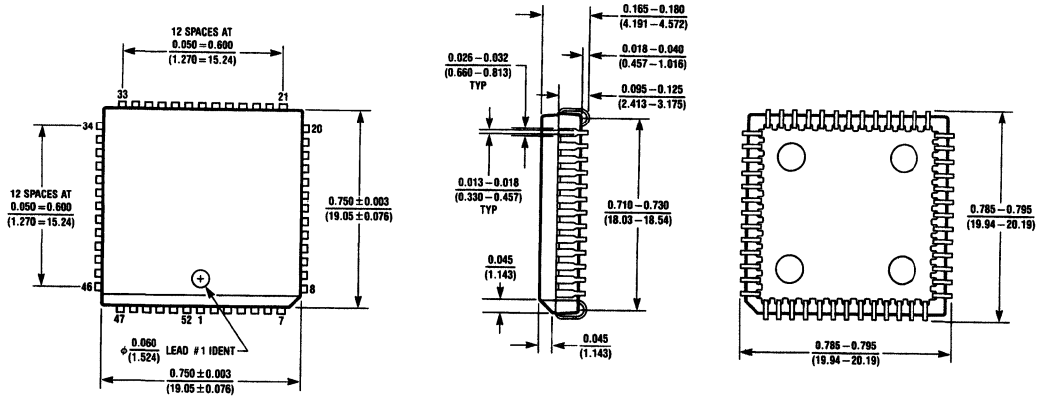






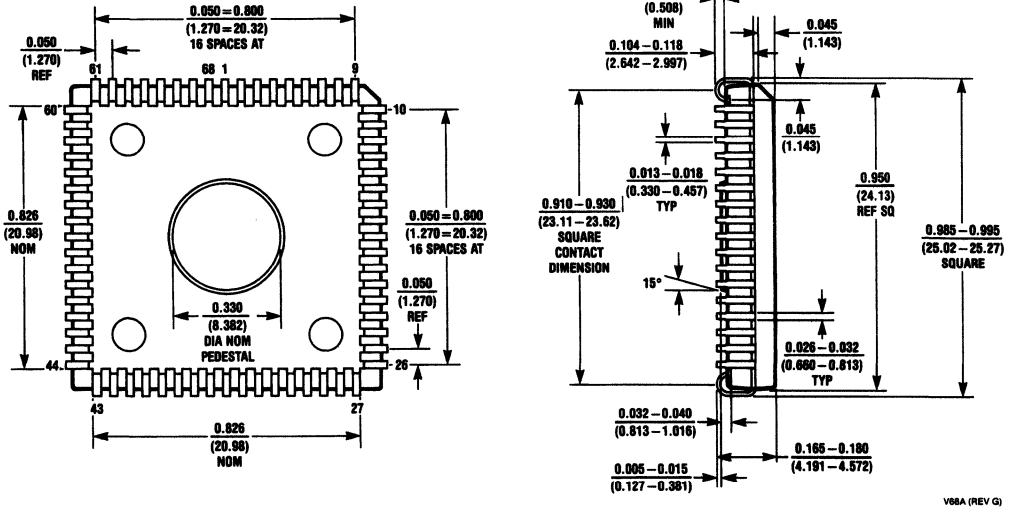
V44A (REV H)

52 Lead Plastic Chip Carrier (V)
NS Package Number V52A



V52A (REV A)

68 Lead Plastic Chip Carrier (V)
NS Package Number V68A



V68A (REV G)

NOTES

NOTES

NOTES