12-84

DATA BOOK

8-/16-Bit Peripheral LSI

1986

HITACHI

HITACHI MICROCOMPUTER DATA BOOK

8-BIT·16-BIT PERIPHERAL LSI

HITACHI

MEDICAL APPLICATIONS

Hitachi's products are not authorized for use in medical applications, including, but not limited to, use in life support devices without the written consent of the appropriate officer of Hitachi's sales company. Buyers of Hitachi's products are requested to notify Hitachi's sales offices when planning to use the products in the applications which involves medical applications.

NOTICE

The information in this document has been carefully checked. However, no responsibility is assumed for inaccuracies.

The example of an applied circuit or combination with other equipment shown herein indicates characteristics and performance of a semiconductor-applied products. The Company shall assume to responsibility for any problem involving a patent caused when applying the descriptions in the example.

CONTENTS

GENERAL INFORMATION

- Quick Reference Guide
- Introduction of Packages
- Reliability and Quality Assurance
- Reliability Test Data of Microcomputer

QUICK REFERENCE GUIDE

■ 8-BIT MICROCOMPUTER PERIPHERAL

Division	Type No!	Old Type Name	LSI Characteristics: Process	Clock Frequency (MHz)	Supply Voltage (V)	Operating*** Temperature (°C)	Package†	Function	Compatibility	Reference Page
PIA	HD6321*		CMOS	1	5.0	−20~+75	DP-40 FP-54	Peripheral Interface Adapter		37
	HD63A21*			1.5						
	HD63B21*			2						
	HD6821	HD46821	NMOS	1	5.0	−20~+75	DP-40	Peripheral Interface Adapter	MC6821	37
	HD68A21	HD468A21		1.5					MC68A21	
	HD68B21	HD468B21		2					MC68B21	
PTM	HD6340*		CMOS	1	5.0	−20~+75	DP-28	Programmable Timer Module		59
	HD63A40*			1.5						
	HD63B40*			2						
	HD6840		NMOS	1	5.0	−20~+75	DP-28	Programmable Timer Module	MC6840	59
	HD68A40			1.5					MC68A40	
	HD68B40			2					MC68B40	
DMAC	HD6844	HD46504	NMOS	1	5.0	−20~+75	DP-40	Direct Memory Access Controller	MC6844	76
	HD68A44	HD46504-1		1.5					MC68A44	
	HD68B44	HD46504-2		2					MC68B44	
CRTC	HD6345*		CMOS	1	5.0	−20~+75	DP-40	CRT Controller (4.5 MHz High Speed Display) 6800 type bus timing		109
	HD63A45*			1.5						
	HD63B45*			2						
	HD6845	HD46505R	NMOS	1	5.0	−20~+75	DP-40	CRT Controller (3.0 MHz High Speed Display)	MC6845	148
	HD68A45	HD46505R-1		1.5					MC68A45	
	HD68B45	HD46505R-2		2					MC68B45	
	HD6845S	HD46505S-1	NMOS	1	5.0	−20~+75	DP-40	CRT Controller (3.7 MHz High Speed Display)		179
	HD68A45S	HD46505S-1		1.5						
	HD68B45S	HD46505S-2		2						
	HD6445-4*		CMOS	4	5.0	−20~+75	DP-40	CRT Controller (4.5 MHz High Speed Display) 80 type bus timing		147
ACIA	HD6350		CMOS	1	5.0	−20~+75	DP-24	Asynchronous Communications Interface Adapter		219
	HD63A50			1.5						
	HD63B50			2						
	HD6850	HD46850	NMOS	1	5.0	−20~+75	DP-24	Asynchronous Communications Interface Adapter	MC6850	219
	HD68A50	HD468A50		1.5					MC68A50	
SSDA	HD6852	HD46852	NMOS	1	5.0	−20~+75	DP-24	Synchronous Serial Data Adapter	MC6852	231
	HD68A52	HD468A52		1.5					MC68A52	
ADU	HD46508		NMOS	1	5.0	−20~+75	DP-24	Analog Data Aquisition Unit		245
	HD46508-1			1.5						
	HD46508A			1						
	HD46508A-1			1.5						
RTC	HD146818		CMOS	1	5.0	0~+70	DP-24 FP-24	Real Time Clock plus RAM	MC146818	536
DIPP	HD63084*		CMOS	10(max)	5.0	0~+70	DP-64S	Document Image Pre-Processor		265
DICEP	HD63085*		CMOS	32(max)	5.0	0~+70	PC-72	Document Image Compression and Expansion Processor		266
S-DPRAM	HD63310**		CMOS		5.0	0~+70	DP-48	Smart Dual Port RAM		299
LCTC	HD63645**		CMOS	2	5.0	−20~+75	FP-80	LCD Timing Controller		530
ACI	HD64941**		NMOS		5.0	0~+70	DP-24N	Asynchronous Communications Interface	SCN2641	533

* Preliminary ** Under development *** Wide temperature range (−40~+85°C) version is available.

†DP: Plastic DIP, FP: Flat Plastic Package, DC: Ceramic DIP PGA: Pin Grid Array, PC: Ceramic Pin Grid Array, CP: Plastic Leaded Chip Carrier

■ 16-BIT MICROCOMPUTER PERIPHERAL

Division	Type No.	LSI Characteristics					Function	Compatibility	Reference Page
		Process	Clock Frequency (MHz)	Supply Voltage (V)	Operating Temperature (°C)	Package			
PI/T	HD68230P8*	NMOS	8	5.0	0~+70	DP-48	Parallel Interface Timer	MC68230L8	267
	HD68230P10*		10					MC68230L10	
DMAC	HD63450-6*	CMOS	6	5.0	0~+70	DC-64 DP-64 DP-64S PGA-68 CP-68	Direct Memory Access Controller		301
	HD63450-8*		8						
	HD63450-10*		10						
	HD63450-12*		12.5						
	HD68450-4	NMOS	4	5.0	0~+70	DP-64 PGA-68	Direct Memory Access Controller	MC68450L4	348
	HD68450-6		6					MC68450L6	
	HD68450-8		8					MC68450L8	
	HD68450-10		10						
HDC	HD63463-4	CMOS	4	5.0	0~+70	DC-48 DP-48 CP-52	Hard Disk Controller		396
	HD63463-6		6						
	HD63463-8		8						
ACRTC	HD63484-4	CMOS	4	5.0	0~+70	DC-64 DP-64 CP-68	Advanced CRT Controller		464
	HD63484-6		6						
	HD63484-8		8						
GMIC	HD63485**	Hi-Bi CMOS		5.0	0~+70	DP-64S CP-68	Graphic Memory Interface Controller		523
GVAC	HD63486**	Hi-Bi CMOS		5.0	0~+70	DP-64S CP-68	Graphic Video Attribute Controller		525
DUSCC	HD68562**	NMOS	4 (max)	5.0	0~+70	DC-48	Dual Universal Serial Communications Controller	MC68562 SCN68562	527

* Preliminary ** Under development

■ 8/16 BIT MICROCOMPUTER PERIPHERAL

Division		Type No.	LSI Characteristics					Function	Compatibility	Reference Page
			Process	Clock Frequency (MHz)	Supply Voltage (V)	Operating Temperature*** (°C)	Package†			
Display Control	CRTC	HD6845	NMOS	1	5.0	−20~+75	DP-40	CRT Controller (3.0 MHz High Speed Display)	MC6845	148
		HD68A45		1.5					MC68A45	
		HD68B45		2					MC68B45	
		HD6845S	NMOS	1	5.0	−20~+75	DP-40	CRT Controller (3.7 MHz High Speed Display)		179
		HD68A45S		1.5						
		HD68B45S		2						
		HD6345*	CMOS	1	5.0	−20~+75	DP-40	CRT Controller (4.5 MHz High Speed Display) 6800 type bus timing		109
		HD63A45*		1.5						
		HD63B45*		2						
		HD6445-4*	CMOS	4	5.0	−20~+75	DP-40	CRT Controller (4.5 MHz High Speed Display) 80 type bus timing		147
	LCTC	HD63645**	CMOS	2	5.0	−20~+75	FP-80	LCD Timing Controller		530
	ACRTC	HD63484-4	CMOS	4	5.0	0~+70	DC-64	Advanced CRT Controller		464
		HD63484-6		6			DP-64			
		HD63484-8		8			CP-68			
	GMIC	HD63485**	Hi-Bi CMOS		5.0	0~+70	DP-64S CP-68	Graphic Memory Interface Controller		523
	GVAC	HD63486**	Hi-Bi CMOS		5.0	0~+70	DP-64S CP-68	Graphic Video Attribute Controller		525
Communication Interface	ACIA	HD6850	NMOS	1	5.0	−20~+75	DP-24	Asynchronous Communications Interface Adapter	MC6850	219
		HD68A50		1.5					MC68A50	
		HD6350	CMOS	1	5.0	−20~+75	DP-24	Asynchronous Communications Interface Adapter		219
		HD63A50		1.5						
		HD63B50		2						
	SSDA	HD6852	NMOS	1	5.0	−20~+75	DP-24	Synchronous Serial Data Adapter	MC6852	230
		HD68A52		1.5					MC68A52	
	DUSCC	HD68562**	NMOS	4 (max)	5.0	0~+70	DC-48	Dual Universal Serial Communications Control	SCN68562 MC68562	527
	ACI	HD64941**	NMOS		5.0	0~+70	DP-24N	Asynchronous Communications Control	SCN2641	533
Timer Control	PTM	HD6840	NMOS	1	5.0	−20~+75	DP-28	Programmable Timer Module	MC6840	59
		HD68A40		1.5					MC68A40	
		HD68B40		2					MC68B40	
		HD6340*	CMOS	1	5.0	−20~+75	DP-28	Programmable Timer Module		59
		HD63A40*		1.5						
		HD63B40*		2						
	RTC	HD146818	CMOS	1	5.0	0~+70	DP-24 FP-24	Real Time Clock Plus RAM	MC146818	536
	PI/T	HD68230P8*	NMOS	8	5.0	0~+70	DP-48	Parallel Interface Timer	MC68230L8	267
		HD68230P10*		10					MC68230L10	
File Control	HDC	HD63463-4	CMOS	4	5.0	0~+70	DC-48	Hard Disk Controller		396
		HD63463-6		6			DP-48			
		HD63463-8		8			CP-52			

* Preliminary ** Under development *** Wide temperature range (−40 ~ +85°C) version is available)

† DP: Plastic DIP, DC: Ceramic DIP, FP: Flat Plastic Package, PGA: Pin Grid Array, PC: Ceramic Pin Grid Array, CP: Plastic Leaded Chip Carrier

■ 8/16 BIT MICROCOMPUTER PERIPHERAL

Division		Type No.	LSI Characteristics					Function	Compatibility	Reference Page
			Process	Clock Frequency (MHz)	Supply Voltage (V)	Operating*** Temperature (°C)	† Package			
System Control	DMAC	HD6844	NMOS	1	5.0	−20~+75	DP-40	8-Bit Direct Memory Access Controller	MC6844	76
		HD68A44		1.5					MC68A44	
		HD68B44		2					MC68B44	
		HD68450-4	NMOS	4	5.0	0~+70	DC-64 PGA-68	16-bit Direct Memory Access Controller	MC68450L4	348
		HD68450-6		6					MC68450L6	
		HD68450-8		8					MC68450L8	
		HD68450-10		10					MC68450L10	
		HD63450-6*	CMOS	6	5.0	0~+70	DC-64 DP-64 DP-64S PGA-68 CP-68	16-Bit Direct Memory Access Controller		301
		HD63450-8*		8						
		HD63450-10*		10						
		HD63450-12*		12.5						
	S-DPRAM	HD63310**	CMOS		5.0	0~+70	DP-48	Smart Dual Port RAM		299
Peripheral Interface	PIA	HD6821	NMOS	1	5.0	−20~+75	DP-40	Peripheral Interface Adapter	MC6821	37
		HD68A21		1.5					MC68A21	
		HD68B21		2					MC68B21	
		HD6321*	CMOS	1	5.0	−20~+75	DP-40 FP-54	Peripheral Interface Adapter		37
		HD63A21*		1.5						
		HD63B21*		2						
	ADU	HD46508	NMOS	1	5.0	−20~+75	DP-40	Analog Data Acquisition Unit		245
		HD46508-1		1.5						
		HD46508A		1						
		HD46508A-1		1.5						
	PI/T	HD68230P8*	NMOS	8	5.0	0~+70	DP-48	Parallel Interface Timer	MC68230L8	267
		HD68230P10*		10					MC68230L10	
Document Image Processor	DIPP	HD63084*	CMOS	10 (max)	5.0	0~+70	DP-64S	Document Image Pre-Processor		265
	DICEP	HD63085*	CMOS	32 (max)	5.0	0~+70	PC-72	Document Image Compression and Expansion Processor		266

* Preliminary ** Under development *** Wide temperature range (−40 ~ +85°C) version is available.

† DP: Plastic DIP, DC: Ceramic DIP, FP: Flat Plastic Package PGA: Pin Grid Array, PC: Ceramic Pin Grid Array, CP: Plastic leaded chip carrier

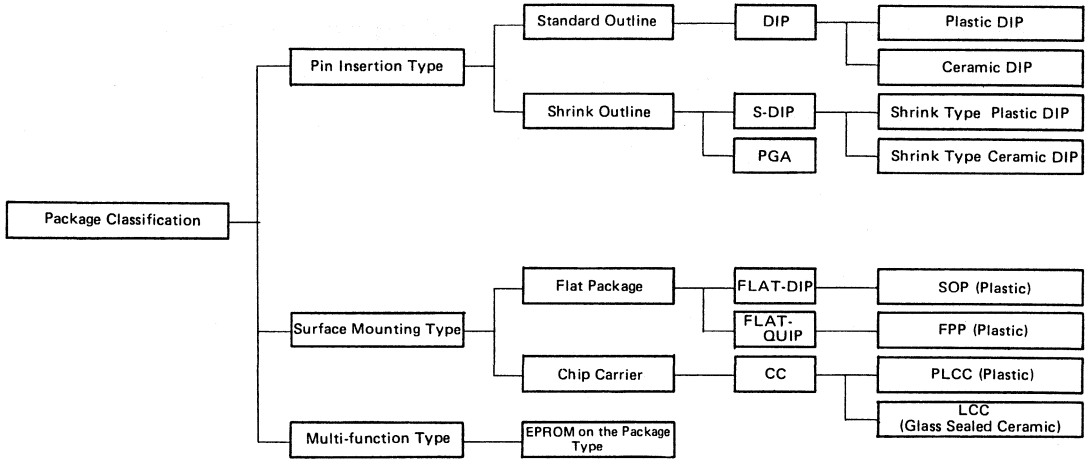
INTRODUCTION OF PACKAGES

Hitachi microcomputer devices include various types of package which meet a lot of requirements such as ever smaller, thinner and more versatile electric appliances. When selecting a package suitable for the customers' use, please refer to the following for Hitachi microcomputer packages.

1. Package Classification

There are pin insertion types, surface mounting types and multi-function types, applicable to each kind of mounting method. Also, plastic and ceramic materials are offered according to use.

Figure 1 shows the package classification according to the mounting types on the Printed Circuit Board (PCB) and the materials.



DIP; DUAL IN LINE PACKAGE
S-DIP; SHRINK DUAL IN LINE PACKAGE
PGA: PIN GRID ARRAY
FLAT-DIP; FLAT DUAL IN LINE PACKAGE
FLAT-QUIP; FLAT QUAD IN LINE PACKAGE
CC: CHIP CARRIER
SOP; SMALL OUTLINE PACKAGE
FPP; FLAT PLASTIC PACKAGE
PLCC; PLASTIC LEADED CHIP CARRIER
LCC ; LEADLESS CHIP CARRIER

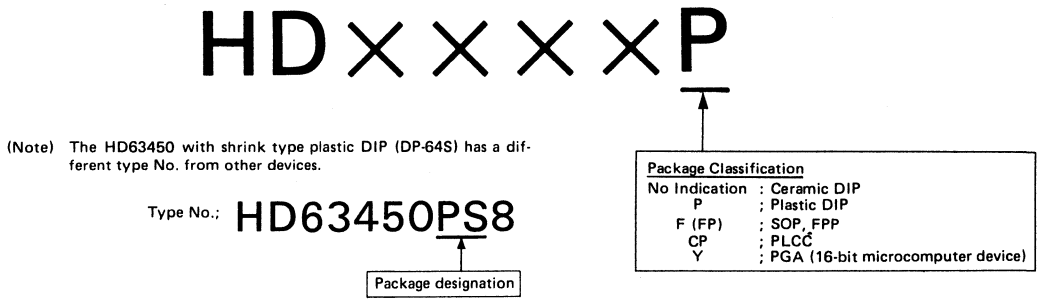
Figure 1 Package Classification according to the Mounting Type on the Printed Circuit Board and the Materials.

2. Type No. and Package Code Indication

Type No. of Hitachi 8/16 bit microcomputer peripheral device is followed by package material and outline specifications, as shown below. The package type used for each device is identified by code as follows, illustrated in the data sheet of each device.

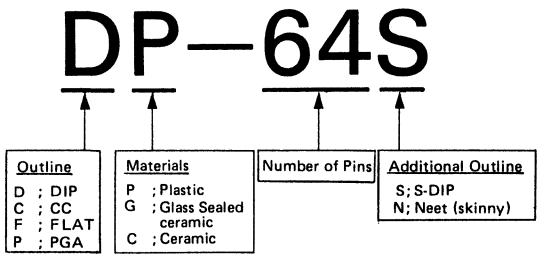
When ordering, please write the package code beside the type number.

Type No. Indication

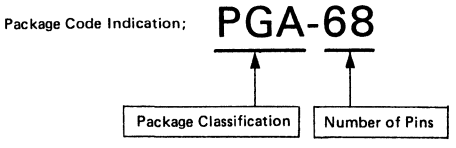


(Note) The HD63450 with shrink type plastic DIP (DP-64S) has a different type No. from other devices.

Package Code Indication



(Note) PGA packages of 16-bit microcomputer devices have a different indication.



3. Package Dimensional Outline

Hitachi multi-chip microcomputer device employs the packages shown in Table 1 according to the mounting method on the PCB.

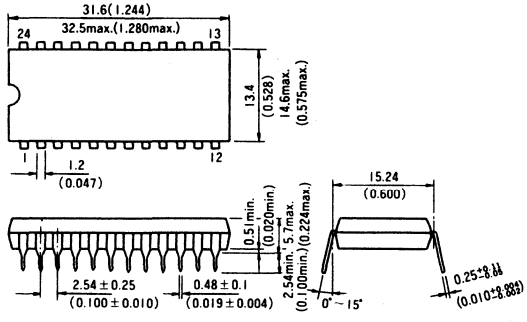
Table 1 Package List

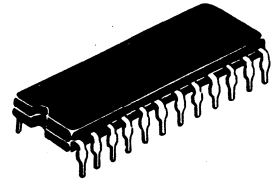
Method of Mounting	Package Classification		Package Material	Package Code
Pin Insertion Type	Standard Outline (DIP)		Plastic	DP-24 DP-24N DP-28 DP-40 DP-64
			Ceramic	DC-48 DC-64
	Shrink Outline	S-DIP	Plastic	DP-64S
		PGA	Glass Sealed Ceramic	PGA-68 PC-72
Surface Mounting Type	Flat Package	FLAT-DIP (SOP)	Plastic	FP-24
		FLAT-QUIP (FPP)		FP-54 FP-80

Plastic DIP

● **DP-24**

Scale : 1/1

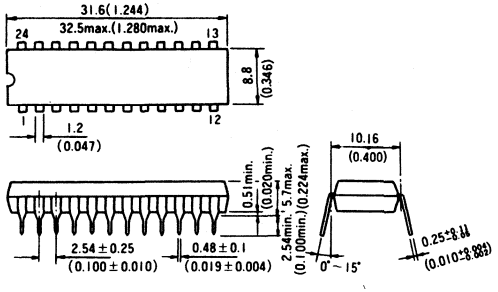


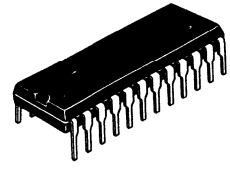


(Unit: mm)

● **DP-24N**

Scale: 1/1

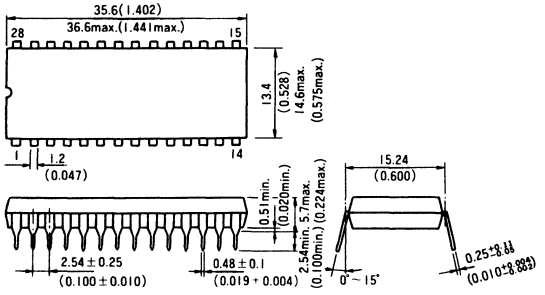


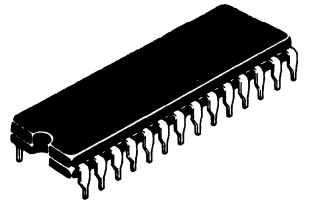


(Unit: mm)

• DP-28

Scale : 1/1

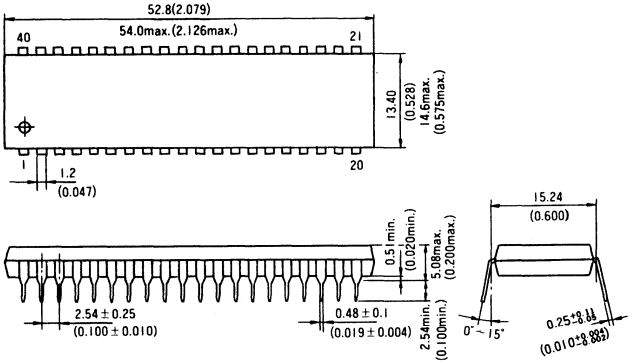


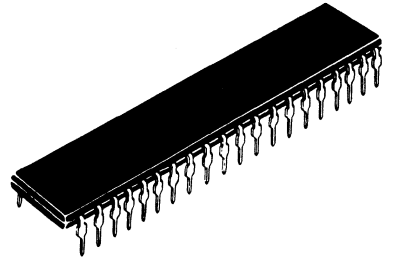


(Unit: mm)

• DP-40

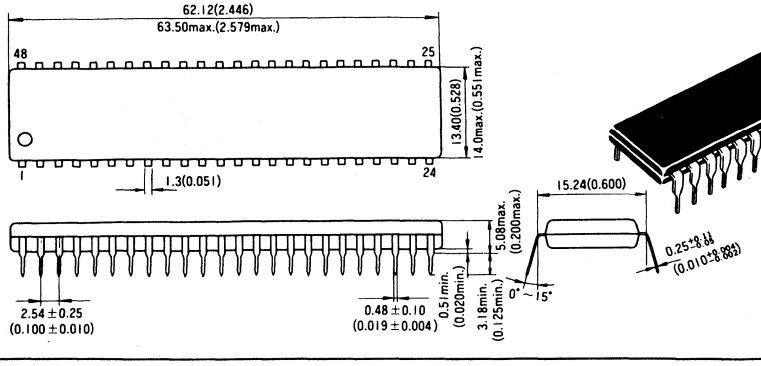
Scale : 1/1

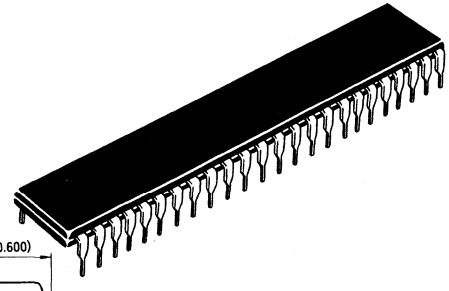




Unit (mm)

• DP-48

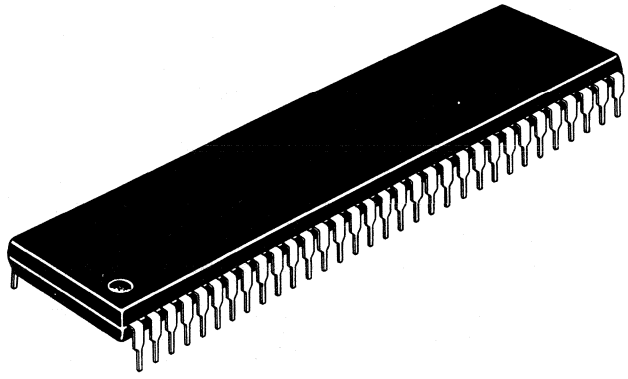


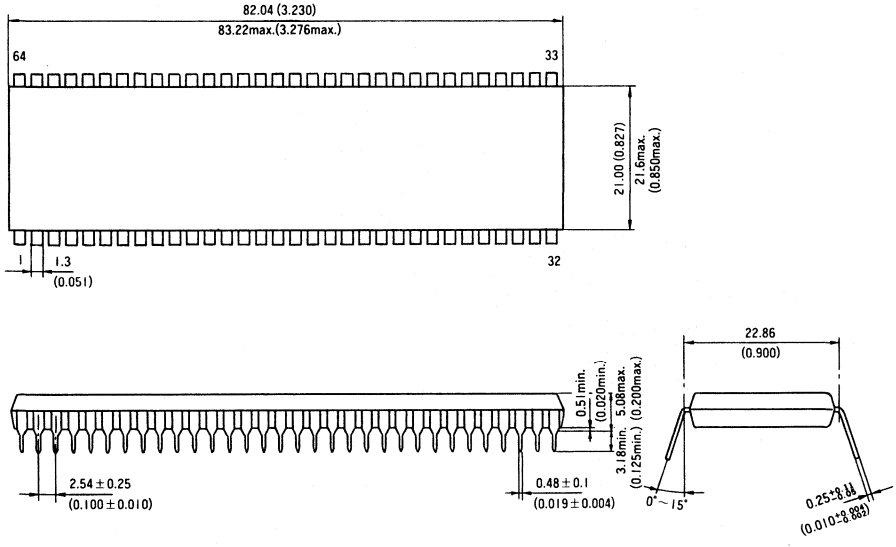


(Unit: mm)

● **DP-64**

Scale : 1/1



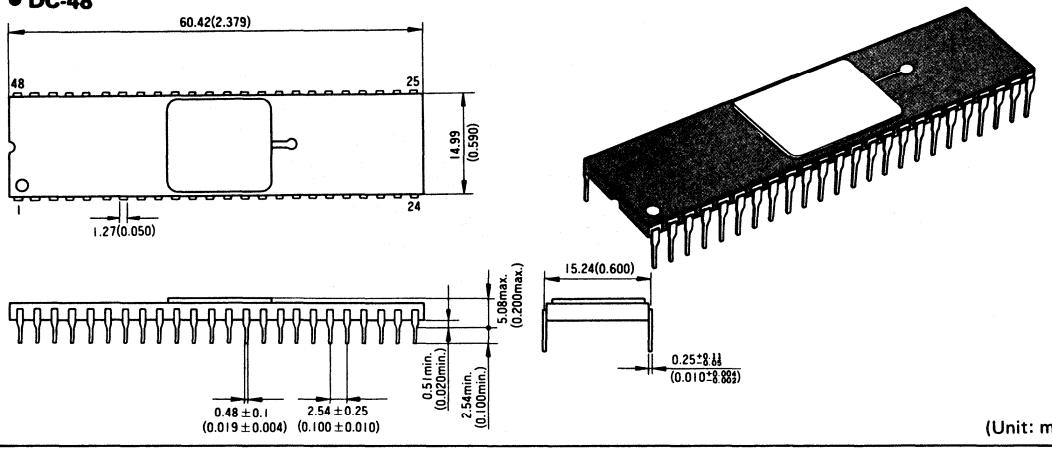


(Unit: mm)

Ceramic DIP

● DC-48

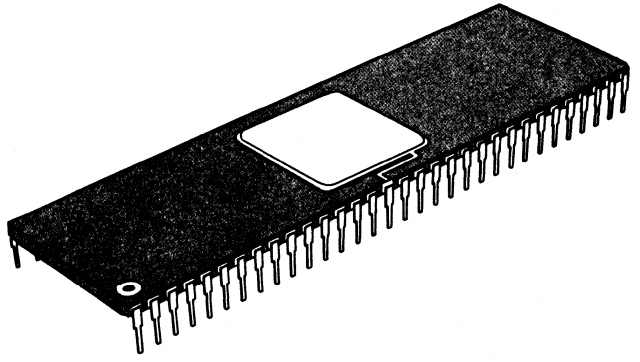
Scale : 1/1

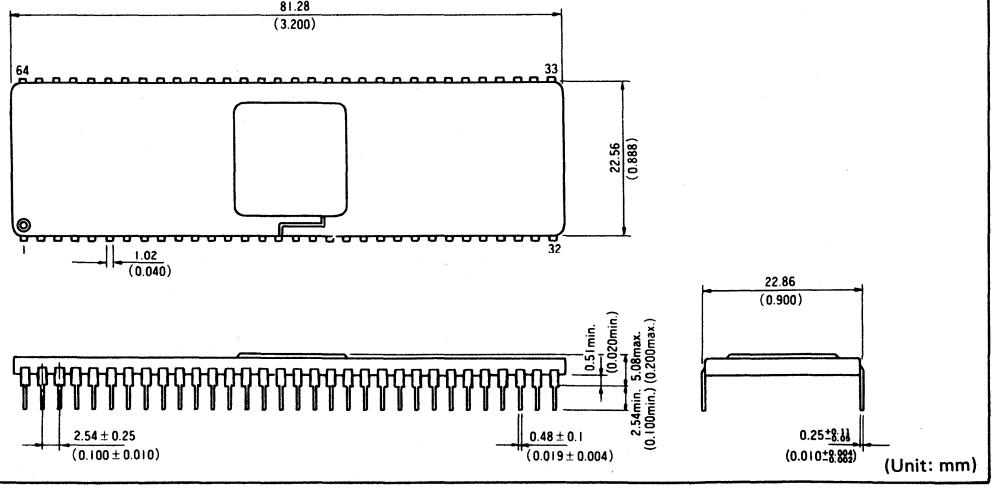


(Unit: mm)

● DC-64

Scale : 1/1



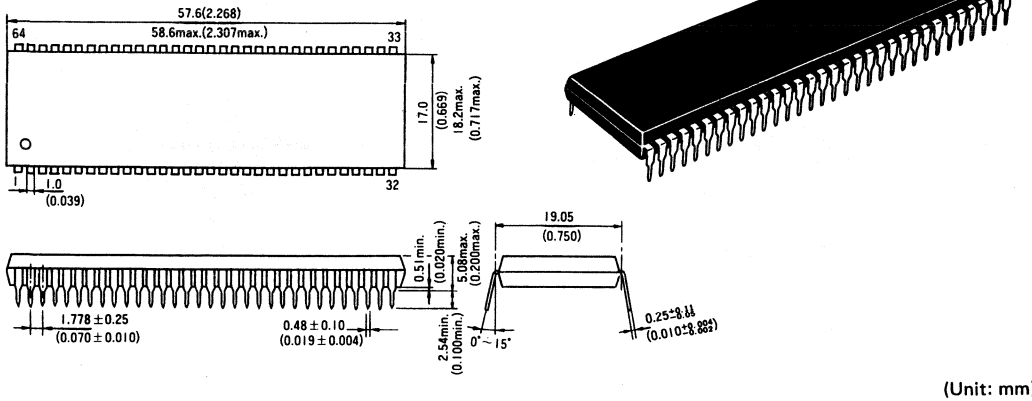


(Unit: mm)

Shrink Type Plastic DIP

• DP-64S

Scale: 1/1

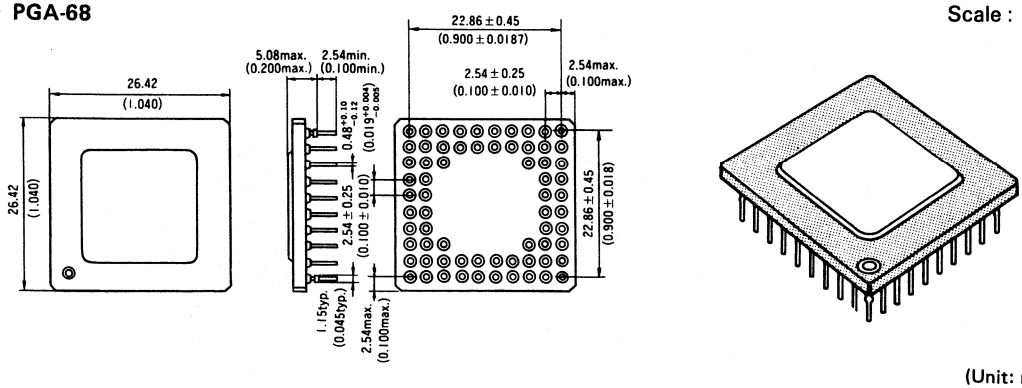


(Unit: mm)

Pin Grid Array

• PGA-68

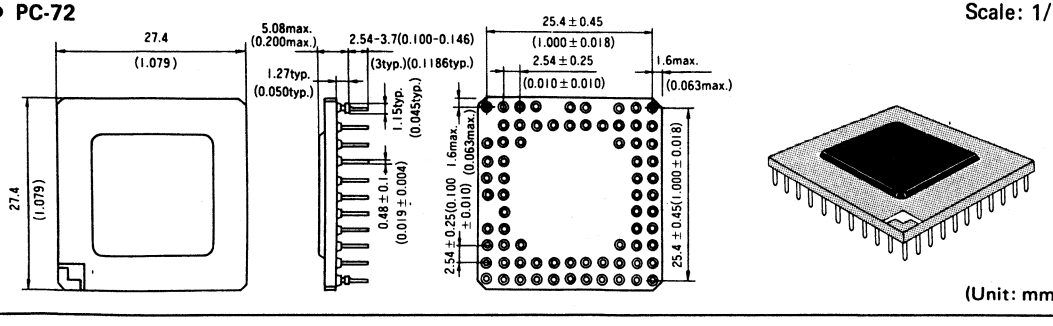
Scale : 1/1



(Unit: mm)

• PC-72

Scale: 1/1



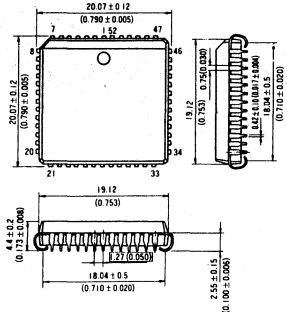
(Unit: mm)

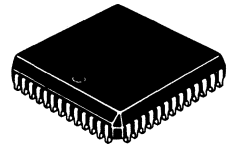
Plastic Leaded Chip Carrier

<PLCC>

• CP-52

Scale: 1/1

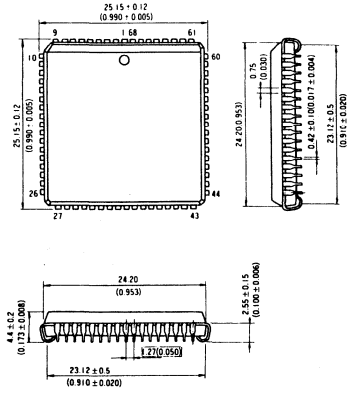


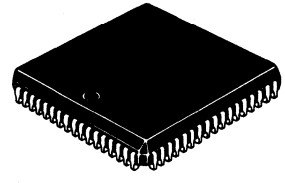


(Unit: mm)

• CP-68

Scale: 1/1





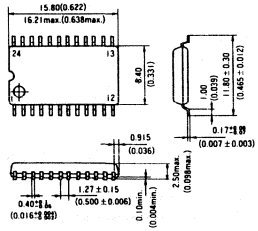
(Unit: mm)

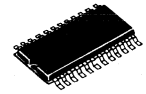
Flat Package

〈SOP〉

● FP-24

Scale : 1/1



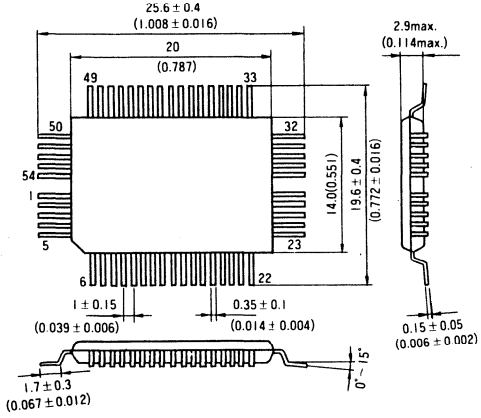


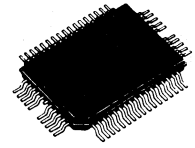
(Unit: mm)

〈FPP〉

● FP-54

Scale : 3/2

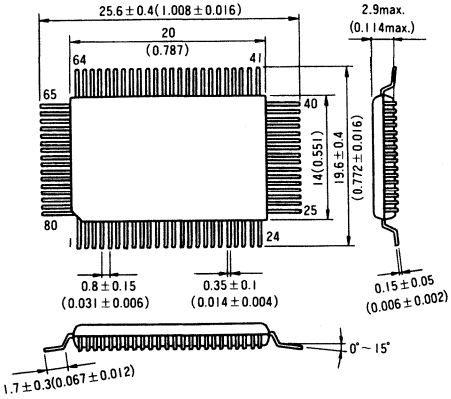


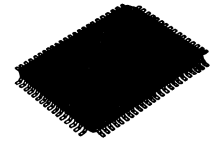


(Unit: mm)

● FP-80

Scale: 3/2





(Unit: mm)

4. Mounting Method on Board

Lead pins of the package have surface treatment, such as solder coating or solder plating, to make them easy to mount on the PCB. The lead pins are connected to the package by eutectic solder. The following explains the common connecting method of leads and precautions.

4.1 Mounting Method of Pin Insertion Type Package

Insert lead pins of the package into through-holes (usually about ϕ0.8mm) on the PCB. Soak the lead part of the package in a wave solder tub.

Lead pins of the package are held by the through-holes. Therefore, it is easy to handle the package through the process up to soldering, and easy to automate the soldering process. When soldering the lead part of the package in the wave solder tub , be careful not to get the solder on the package, because the wave solder will damage it.

4.2 Mounting Method of Surface Mounting Type Package

Apply the specified quantity of solder paste to the pattern on any printed board by the screen printing method, and put a package on it. The package is now temporarily fixed to the printed board by the surface tension of the paste. The solder paste melts when heated in a reflowing furnace, and the leads of the package and the pattern of the printed board are fixed together by the surface tension of the melted solder and the self alignment.

The size of the pattern where the leads are attached, partly depending on paste material or furnace adjustment, should be 1.1 to 1.3 times the leads' width.

The temperature of the reflowing furnace depends on package material and also package types. Fig. 2 lists the adjustment of the reflowing furnace for FPP. Pre-heat the furnace to 150°C. The surface temperature of the resin should be kept at 235°C max. for 10 minutes or less.

(1) The temperature of the leads should be kept at 260°C for 10 minutes or less.
(2) The temperature of the resin should be kept at 235°C for 10 minutes or less.
(3) Below is shown the temperature profile when soldering a package by the reflowing method.

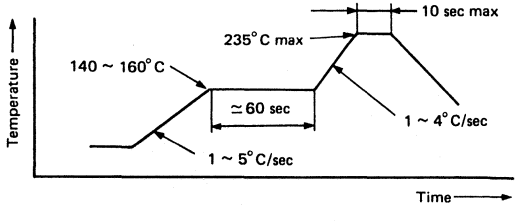


Figure 2 Reflowing Furnace Adjustment for FPP

Ensure good heater or temperature controls because the material of a plastic package is black epoxy-resin which damages easily. When an infrared heater is used, if the temperature is higher than the glass transition point of epoxy-resin (about 150°C), for a long time, the package may be damaged and the reliability lowered. Equalize the temperature inside and outside the packages by lessening the heat of the upper surface of the packages.

Leads of FPP may be easily bent under shipment or during handling and cannot be soldered onto the printed board. If they are, heat the bent leads again with a soldering iron to reshape them.

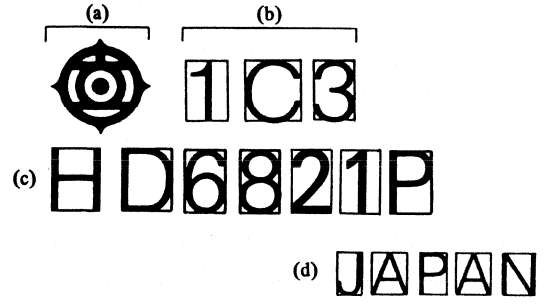
Use a rosin flux when soldering. Don't use a chloric flux because the chlorine in the flux tends to remain on the leads and lower the reliability of the product.

Even if you use a rosin flux, remaining flux can cause the leads to deteriorate. Wash away flux from packages with alcohol, chlorothene or freon. But don't leave these solvents on the packages for a long time because the marking may disappear.

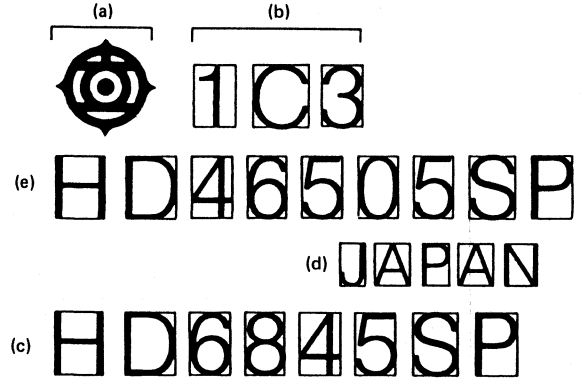
5. Marking

Hitachi trademark, product type No., etc. are printed on packages. Case I and Case II give examples of marks and Nos. Case I applies to products which have only a standard type No. Case II applies to products which have an old type No. and a standard type No.

Case I; Includes a standard type No.



Case II; Includes an old type No. and a standard type No.



Meaning of Each Mark

(a)	Hitachi Trademark
(b)	Lot Code
(c)	Standard Type No.
(d)	Japan Mark
(e)	Old Type No.

RELIABILITY AND QUALITY ASSURANCE

1. VIEWS ON QUALITY AND RELIABILITY

Basic views on quality in Hitachi are to meet individual user's purchase purpose and quality required, and to be at the satisfied quality level considering general marketability. Quality required by users is specifically clear if the contract specification is provided. If not, quality required is not always definite. In both cases, efforts are made to assure the reliability so that semiconductor devices delivered can perform their ability in actual operating circumstances. To realize such quality in manufacturing process, the key points should be to establish quality control system in the process and to enhance morale for quality.

In addition, quality required by users on semiconductor devices is going toward higher level as performance of electronic system in the market is going toward higher one and is expanding size and application fields. To cover the situation, actual bases Hitachi is performing is as follows;

(1) Build the reliability in design at the stage of new product development.
(2) Build the quality at the sources of manufacturing process.
(3) Execute the harder inspection and reliability confirmation of final products.
(4) Make quality level higher with field data feed back.
(5) Cooperate with research laboratories for higher quality and reliability.

With the views and methods mentioned above, utmost efforts are made for users' requirements.

2. RELIABILITY DESIGN OF SEMICONDUCTOR DEVICES

2.1 Reliability Targets

Reliability target is the important factor in manufacture and sales as well as performance and price. It is not practical to rate reliability target with failure rate at the certain common test condition. The reliability target is determined corresponding to character of equipments taking design, manufacture, inner process quality control, screening and test method, etc. into consideration, and considering operating circumstances of equipments the semiconductor device used in, reliability target of system, derating applied in design, operating condition, maintenance, etc.

2.2 Reliability Design

To achieve the reliability required based on reliability targets, timely sude and execution of design standardization, device design (including process design, structure design), design review, reliability test are essential.

(1) Design Standardization

Establishment of design rule, and standardization of parts, material and process are necessary. As for design rule, critical items on quality and reliability are always studied at circuit design, device design, layout design, etc. Therefore, as long as standardized process, material, etc. are used, reliability risk is extremely small even in new development devices, only except for in the case special requirements in function needed.

(2) Device Design

It is important for device design to consider total balance of process design, structure design, circuit and layout design. Especially in the case new process and new material are employed, technical study is deeply executed prior to device development.

(3) Reliability Evaluation by Test Site

Test site is sometimes called Test Pattern. It is useful method for design and process reliability evaluation of IC and LSI which have complicated functions.

1. Purposes of Test Site are as follows;
   - Making clear about fundamental failure mode
   - Analysis of relation between failure mode and manufacturing process condition
   - Search for failure mechanism analysis
   - Establishment of QC point in manufacturing
2. Effectiveness of evaluation by Test Site are as follows;
   - Common fundamental failure mode and failure mechanism in devices can be evaluated.
   - Factors dominating failure mode can be picked up, and comparison can be made with process having been experienced in field.
   - Able to analyze relation between failure causes and manufacturing factors.
   - Easy to run tests.

   etc.

2.3 Design Review

Design review is organized method to confirm that design satisfies the performance required including users' and design work follows the specified ways, and whether or not technical improved items accumulated in test data of individual major fields and field data are effectively built in. In addition, from the standpoint of enhancement of competitive power of products, the major purpose of design review is to ensure quality and reliability of the products. In Hitachi, design review is performed from the planning stage for new products and even for design changed products. Items discussed and determined at design review are as follows;

(1) Description of the products based on specified design documents.
(2) From the standpoint of specialty of individual participants, design documents are studied, and if unclear matter is found, sub-program of calculation, experiments, investigation, etc. will be carried out.
(3) Determine contents of reliability and methods, etc. based on design document and drawing.
(4) Check process ability of manufacturing line to achieve design goal.
(5) Discussion about preparation for production.
(6) Planning and execution of sub-programs for design change proposed by individual specialist, and for tests, experiments and calculation to confirm the design change.
(7) Reference of past failure experiences with similar devices, confirmation of method to prevent them, and planning and execution of test program for confirmation of them. These studies and decisions are made using check lists made individually depending on the objects.

3. QUALITY ASSURANCE SYSTEM OF SEMICONDUCTOR DEVICES

3.1 Activity of Quality Assurance

General views of overall quality assurance in Hitachi are as follows;

(1) Problems in individual process should be solved in the

process. Therefore, at final product stage, the potential failure factors have been already removed.

(2) Feedback of information should be made to ensure satisfied level of process ability.

(3) To assure reliability required as an result of the things mentioned above is the purpose of quality assurance.

The followings are regarding device design, quality approval at mass production, inner process quality control, product inspection and reliability tests.

3.2 Quality Approval

To ensure quality and reliability required, quality approval is carried out at trial production stage of device design and mass production stage based on reliability design described at section 2.

The views on quality approval are as follows;

(1) The third party performs approval objectively from the standpoint of customers.

(2) Fully consider past failure experiences and information from field.

(3) Approval is needed for design change and work change.

(4) Intensive approval is executed on parts material and process.

(5) Study process ability and fluctuation factor, and set up control points at mass production stage.

Considering the views mentioned above, quality approval shown in Figure 1 is performed.

3.3 Quality and Reliability Control at Mass Production

For quality assurance of products in mass production, quality control is executed with organic division of functions in manufacturing department, quality assurance department, which are major, and other departments related. The total function flow is shown in Figure 2. The main points are described below.

3.3.1 Quality Control of Parts and Material

As the performance and the reliability of semiconductor devices are getting higher, importance is increasing in quality control of material and parts, which are crystal, lead frame, fine wire for wire bonding, package, to build products, and materials needed in manufacturing process, which are mask pattern and chemicals. Besides quality approval on parts and materials stated in section 3.2, the incoming inspection is, also, key in quality control of parts and materials. The incoming inspection is performed based on incoming inspection specification following purchase specification and drawing, and sampling inspection is executed based on MIL-STD-105D mainly.

The other activities of quality assurance are as follows:

(1) Outside Vendor Technical Information Meeting

(2) Approval on outside vendors, and guidance of outside vendors

(3) Physical chemical analysis and test

The typical check points of parts and materials are shown in Table 1.

3.3.2 Inner Process Quality Control

Inner process quality control is performing very important function in quality assurance of semiconductor devices. The following is description about control of semi-final products, final products, manufacturing facilities, measuring equipments,

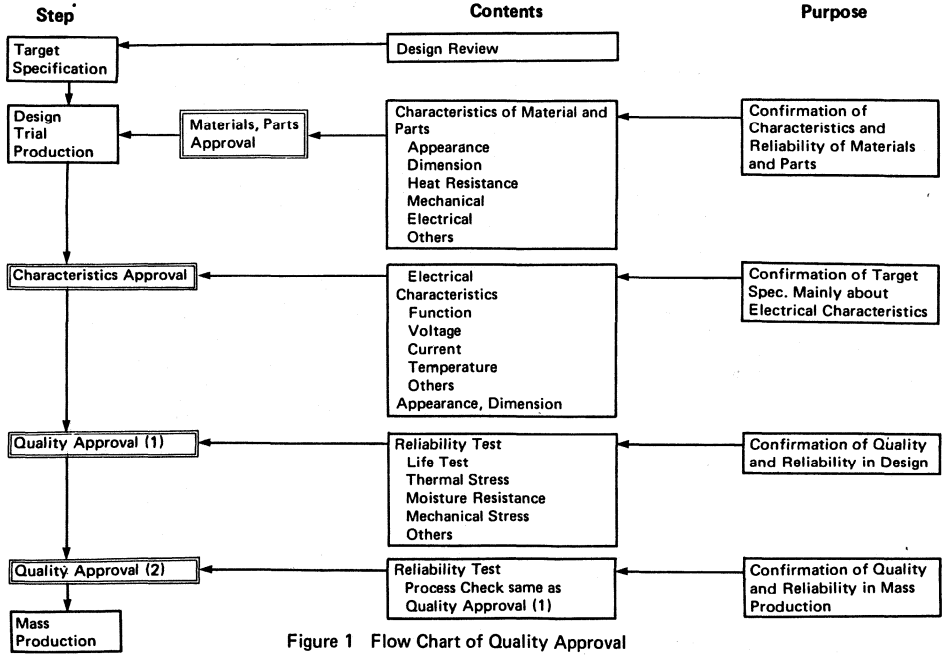


Figure 1 Flow Chart of Quality Approval

circumstances and sub-materials. The quality control in the manufacturing process is shown in Figure 3 corresponding to the manufacturing process.

(1) Quality Control of Semi-final Products and Final Products

Potential failure factors of semiconductor devices should be removed preventively in manufacturing process. To achieve it, check points are set-up in each process, and products which have potential failure factor are not transfer to the next process. Especially, for high reliability semiconductor devices, manufacturing line is rigidly selected, and the quality control in the manufacturing process is tightly executed – rigid check in each process and each lot, 100% inspection in appropriate ways to remove failure factor caused by manufacturing fluctuation, and execution of screening needed, such as high temperature aging and temperature cycling. Contents of inner process quality control are as follows;

- Condition control on individual equipments and workers, and sampling check of semifinal products.
- Proposal and carrying-out improvement of work
- Education of workers
- Maintenance and improvement of yield
- Picking-up of quality problems, and execution of counter-measures
- Transmission of information about quality

(2) Quality Control of Manufacturing Facilities and Measuring Equipment

Equipments for manufacturing semiconductor devices have been developing extraordinarily with necessary high performance devices and improvement of production, and are important factors to determine quality and reliability. In Hitachi, automatization of manufacturing equipments are promoted to improve manufacturing fluctuation, and controls are made to maintain proper operation of high performance equipments and perform the proper function. As for maintenance inspection for quality control, there are daily inspection which is performed daily based on specification related, and periodical inspection which is performed periodically. At the inspection, inspection points listed in the specification are checked one by one not to make any omission. As for adjustment and maintenance of measuring equipments, maintenance number, specification are checked one by one to maintain and improve quality.

(3) Quality Control of Manufacturing Circumstances and Sub-materials

Quality and reliability of semiconductor device is highly

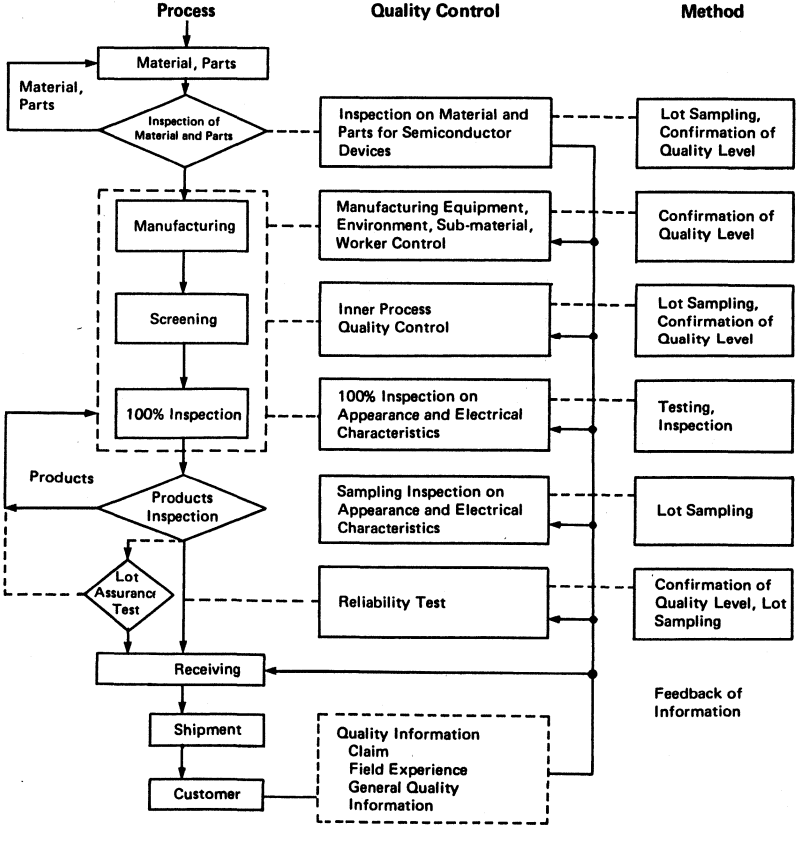


Figure 2 Flow Chart of Quality Control in Manufacturing Process

affected by manufacturing process. Therefore, the controls of manufacturing circumstances – temperature, humidity, dust – and the control of submaterials – gas, pure water – used in manufacturing process are intensively executed. Dust control is described in more detail below.

Dust control is essential to realize higher integration and higher reliability of devices. In Hitachi, maintenance and improvement of cleanness in manufacturing site are executed with paying intensive attention on buildings, facilities, air-conditioning systems, materials delivered-in, clothes, work, etc., and periodical inspection on floating dust in room, falling dusts and dirtiness of floor.

3.3.3 Final Product Inspection and Reliability Assurance

(1) Final Product Inspection

Lot inspection is done by quality assurance department for products which were judged as good products in 100% test, which is final process in manufacturing department. Though 100% of good products is expected, sampling inspection is executed to prevent mixture of failed products by mistake of work, etc. The inspection is executed not only to confirm that the products meet users' requirement, but to consider potential factors. Lot inspection is executed based on MIL-STD-105D.

(2) Reliability Assurance Tests

To assure reliability of semiconductor devices, periodical reliability tests and reliability tests on individual manufacturing lot required by user are performed.

Table 1 Quality Control Check Points of Material and Parts (Example)

Material, Parts	Important Control Items	Point for Check
Wafer	Appearance	Damage and Contamination on Surface
	Dimension	Flatness
	Sheet Resistance	Resistance
	Defect Density	Defect Numbers
	Crystal Axis	
Mask	Appearance	Defect Numbers, Scratch
	Dimension	Dimension Level
	Resistoration	
	Gradation	Uniformity of Gradation
Fine Wire for Wire Bonding	Appearance	Contamination, Scratch, Bend, Twist
	Dimension	
	Purity	Purity Level
	Elongation Ratio	Mechanical Strength
Frame	Appearance	Contamination, Scratch
	Dimension	Dimension Level
	Processing Accuracy	
	Plating	Bondability, Solderability
	Mounting Characteristics	Heat Resistance
Ceramic Package	Appearance	Contamination, Scratch
	Dimension	Dimension Level
	Leak Resistance	Airtightness
	Plating	Bondability, Solderability
	Mounting Characteristics	Heat Resistance
	Electrical Characteristics	
	Mechanical Strength	Mechanical Strength
Plastic	Composition	Characteristics of Plastic Material
	Electrical Characteristics	
	Thermal Characteristics	
	Molding Performance	Molding Performance
	Mounting Characteristics	Mounting Characteristics

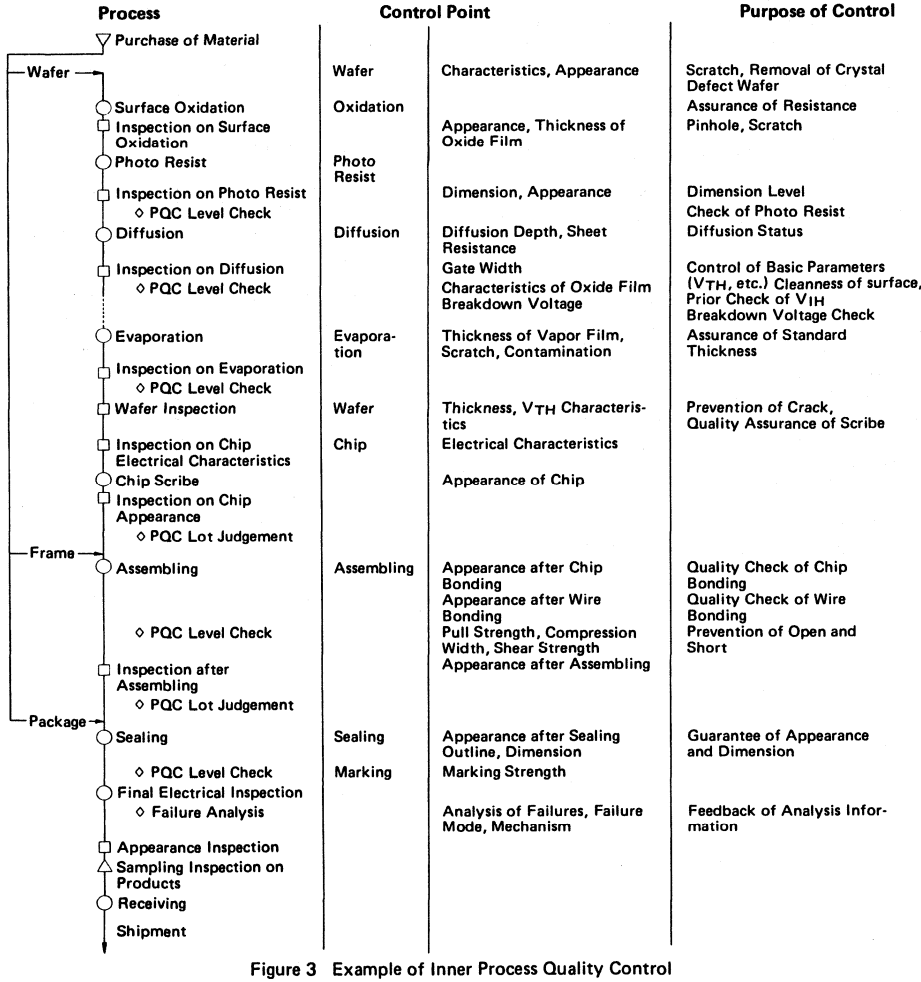


Figure 3 Example of Inner Process Quality Control

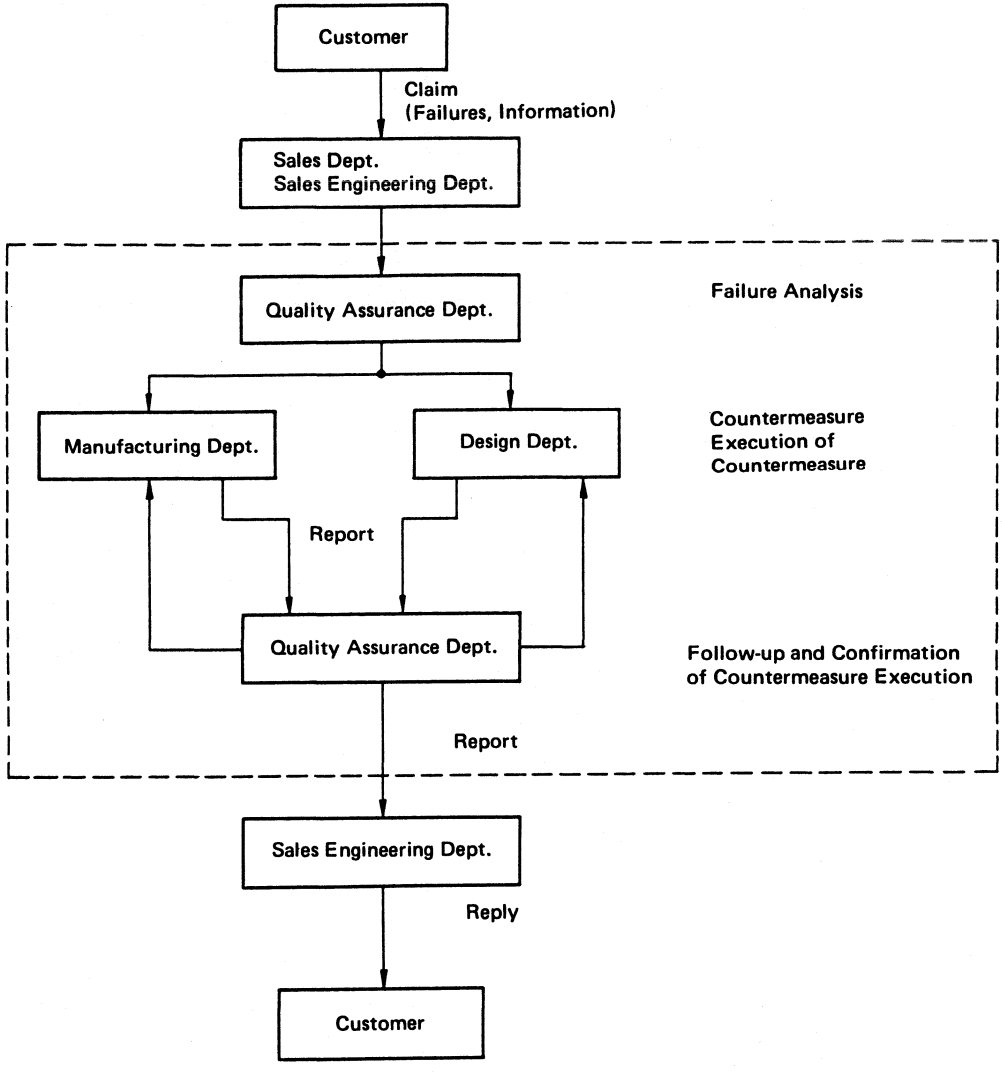


Figure 4 Process Flow Chart of Field Failure

RELIABILITY TEST DATA OF MICROCOMPUTER

1. INTRODUCTION

Microcomputer is required to provide higher reliability and quality with increasing function, enlarging scale and widening application. To meet this demand, Hitachi is improving the quality by evaluating reliability, building up quality in process, strengthening inspection and analyzing field data etc..

This chapter describes reliability and quality assurance data for Hitachi 8-bit and 16-bit microcomputer Peripheral based on test and failure analysis results. More detail data and new information will be reported in another reliability data sheet.

2. PACKAGE AND CHIP STRUCTURE

2.1 Package

The reliability of plastic molded type has been greatly improved, recently their applications have been expanded to automobiles measuring and control systems, and computer terminal equipment operated under relatively severe conditions and production output and application of plastic molded type will continue to increase.

To meet such requirements, Hitachi has considerably improved moisture resistance, operation stability, and chip and plastic manufacturing process.

Plastic and ceramic package type structure are shown in Figure 1 and Table 1.

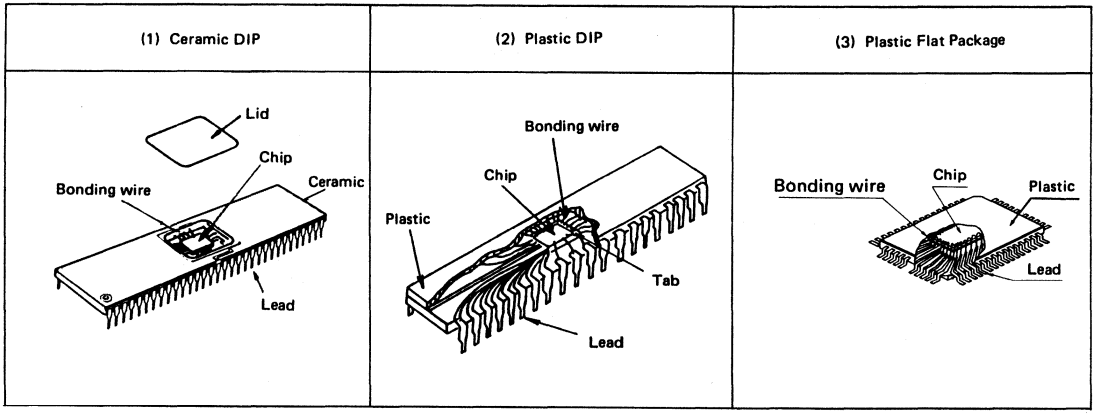


Figure 1 Package Structure

Table 1 Package Material and Properties

Item	Ceramic DIP	Plastic DIP	Plastic Flat Package
Package	Alumina	Epoxy	Epoxy
Lead	Tin plating Brazed Alloy 42	Solder dipping Alloy 42 or Cu	Solder plating Alloy 42
Seal	Au-Sn Alloy	N.A	N.A
Die bond	Au-Si	Au-Si or Ag paste	Au-Si or Ag paste
Wire bond	Ultrasonic	Thermo compression	Thermo compression
Wire	Al	Au	Au

2.2 Chip Structure

Hitachi microcomputers are produced in NMOS E/D technology or low power CMOS technology. Si-gate process is used in both types because of high reliability and high density.

Chip structure and basic circuit are shown in Figure 2.

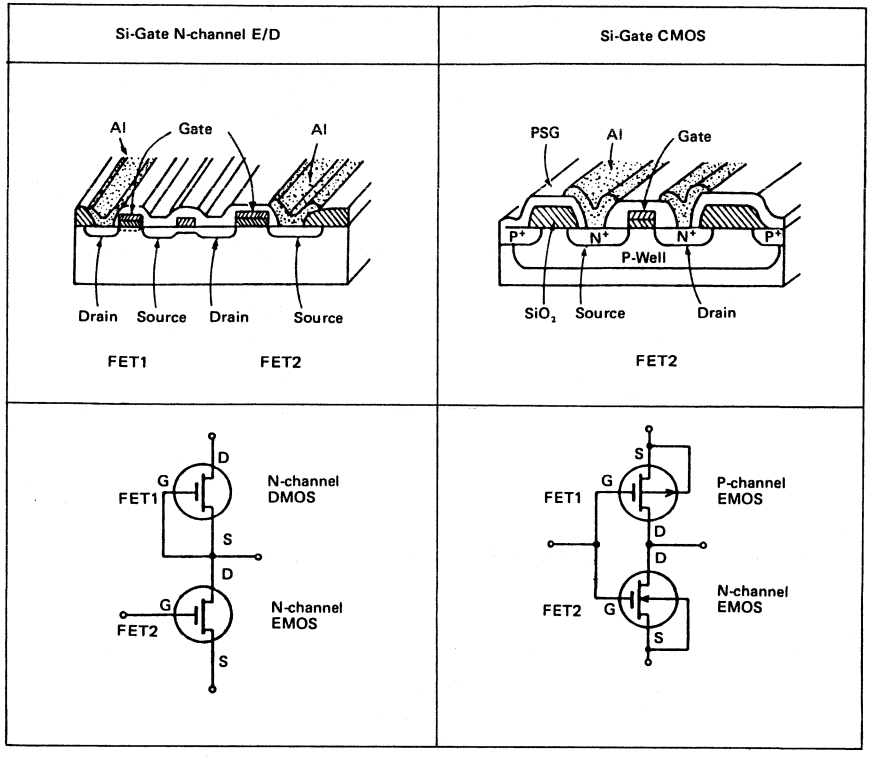


Figure 2 Chip Structure and Basic Circuit

3. QUALITY QUALIFICATION AND EVALUATION

3.1 Reliability Test Methods

Reliability test methods shown in Table 2 are used to qualify and evaluate the new products and new process.

Table 2 Reliability Test Methods

Test Items	Test Condition	MIL-STD-883B Method No.
Operating Life Test	125°C, 1000hr	1005,2
High Temp, Storage	Tstg max, 1000hr	1008,1
Low Temp, Storage	Tstg min, 1000hr	
Steady State Humidity	65°C 95%RH, 1000hr	
Steady State Humidity Biased	85°C 85%RH, 1000hr	
Temperature Cycling	-55°C ~ 150°C, 10 cycles	1010,4
Temperature Cycling	-20°C ~ 125°C, 200 cycles	
Thermal Shock	0°C ~ 100°C, 100 cycles	1011,3
Soldering Heat	260°C, 10 sec	
Mechanical Shock	1500G 0.5 msec, 3 times/X, Y, Z	2002,2
Vibration Fatigue	60Hz 20G, 32hrs/X, Y, Z	2005,1
Variable Frequency	20~2000Hz 20G, 4 min/X, Y, Z	2007,1
Constant Acceleration	20000G, 1 min/X, Y, Z	2001,2
Lead Integrity	225gr, 90° 3 times	2004,3

3.2 Reliability Test Result

Reliability Test Result of 8-bit microcomputer Peripheral devices is shown in Table 3 to Table 7, that of 16-bit microcomputer Peripheral devices in Table 8, Table 9. There is little difference according to device series, as the design and production process, etc. are standardized.

Table 3 Dynamic Life Test (8-bit microcomputer Peripheral)

Device Type	Sample Size	Component Hours	Failures
HD6821	399	266368	1*
HD6850	158	158000	0
HD6852	170	125816	0
HD6846	69	69000	0
HD6843	66	66000	0
HD6844	80	69000	0
HD6845S	88	55000	0
HD6840	64	64000	0
HD46508	140	140000	0
HD146818	44	44000	0
HD6350	70	70000	0
HD6321	45	45000	0
Total	1,393	1,172,184	1

*leakage current

Estimated Field Failure Rate
= 0.01% / 1000 hrs at Ta = 75°C
(Activation Energy = 0.7eV, Confidence Level 60%)

Table 4 High Temperature, High Humidity Test (8-bit multi-chip microcomputer) (Moisture Resistance Test)

(1) 85°C 85%RH Bias Test

Device Type	Vcc Bias	168 hrs	500 hrs	1000 hrs
HD6850P	5.5V	0/45	0/45	0/45
HD6852P	5.5V	0/22	0/22	0/22
HD6843P	5.5V	0/22	0/22	0/22
HD6844P	5.5V	0/22	0/22	0/22
HD6845SP	5.5V	0/137	0/137	0/137
HD6840P	5.5V	0/22	0/22	0/22
HD46508P	5.5V	0/22	0/22	0/22
HD146818P	5.5V	0/22	0/22	0/22
HD6321P	5.5V	0/22	0/22	0/22
Total		0/336	0/336	0/336

2) High Temperature-High Humidity Storage Life Test

Device Type	Condition	168 hrs	500 hrs	1000 hrs
HD6850P	65°C 95%RH	0/135	0/135	0/135
HD6850P	80°C 90%RH	0/22	0/22	0/22
HD6852P	85°C 95%RH	0/22	0/22	0/22
HD6844P	80°C 90%RH	0/22	0/22	0/22
HD6845SP	80°C 90%RH	0/22	0/22	0/22
HD6840P	65°C 95%RH	0/22	0/22	0/22
HD46508P	65°C 95%RH	0/70	0/70	0/70
HD146818P	65°C 95%RH	0/45	0/45	0/45
HD6350P	65°C 95%RH	0/70	0/70	0/70
HD6321P	65°C 95%RH	0/70	0/70	0/70

3) Pressure Cooker Test
(Condition ; 2atm 121°C)

Device Type	40 hrs	60 hrs	100 hrs	200 hrs
HD6821P	0/44	0/44	0/44	–
HD6850P	0/22	0/22	0/22	–
HD6843P	0/22	0/22	0/22	–
HD6845SP	0/43	0/43	0/43	1*/43
HD46508P	0/45	0/45	0/45	–
HD146818P	0/22	0/22	0/22	–
HD6350P	0/22	0/22	0/22	–
HD6321P	0/22	0/22	0/22	–

*Aluminum corrosion

(4) MIL-STD-883B Moisture Resistance Test
(Condition; 65°C ~ −10°C, over 90%RH, Vcc = 5.5V)

Device Type	10 cycles	20 cycles	40 cycles
HD6821P	0/25	0/25	0/25

Table 5 Temperature Cycling Test (8-bit microcomputer Peripheral) (-55°C ~ 25°C ~ 150°C)

Device Type	10 cycles	100 cycles	200 cycles
HD6821P	0/420	0/44	1*/44
HD6850P	0/151	0/38	0/38
HD6852P	0/149	0/38	0/38
HD6843P	0/247	0/44	0/44
HD6844P	0/150	0/44	0/44
HD6845SP	0/358	0/76	0/76
HD6840P	0/148	0/22	0/22
HD46508P	0/207	0/44	0/44
HD146818P	0/103	0/22	0/22
HD6350P	0/150	0/45	0/45
HD6321P	0/120	0/45	0/45

* Static damage

Table 6 High Temperature, Low Temperature Storage Life Test (8-bit microcomputer peripheral)

Device	Temperature	168 hrs	500 hrs	1000 hrs
Peripheral total	150°C	0/110	0/110	0/110
	-55°C	0/88	0/88	0/88

Table 7 Mechanical and Environmental Test (8-bit microcomputer peripheral)

Test Item	Condition	Plastic DIP		Flat Plastic Package	
		Sample Size	Failure	Sample Size	Failure
Thermal Shock	0°C ~ 100°C 10 cycles	110	0	100	0
Soldering Heat	260°C, 10 sec.	164	0	20	0
Salt Water Spray	35°C, NaCl 5% 24 hrs	110	0	20	0
Solderability	230°C, 5 sec. Rosin flux	159	0	34	0
Drop Test	75cm, maple board 3 times	110	0	20	0
Mechanical Shock	1500G, 0.5 ms 3 times/X, Y, Z	110	0	20	0
Vibration Fatigue	60 Hz, 20G 32 hrs/X, Y, Z	110	0	20	0
Vibration Variable Freq.	100 ~ 2000 Hz 20G, 4 times/X, Y, Z	110	0	20	0
Lead Integrity	225 g, 90° Bonding 3 times	110	0	20	0

Table 8 Dynamic Life Test (16-bit microcomputer peripheral)

Device Type	Condition		168 hrs	500 hrs	1000 hrs
	Ta	Vcc			
HD68450	125°C	5.5V	0/44	0/44	0/44
	150°C	5.5V	0/32	0/32	0/32
HD63484	125°C	5.5V	0/45	0/45	0/45

Estimated Field Failure Rate
= 0.013%/1000 hrs at Ta = 75°C
(Activation Energy 0.7eV, Confidence Level 60%)

Table 9 Mechanical and Environmental Test (16-bit microcomputer peripheral)

Test Item	Condition	Device Type
		HD68450
High Temperature Storage	Ta = 295°C, 1000 hrs	0/22
Low Temperature Storage	Ta = −55°C, 1000 hrs	0/42
Temperature Cycling (1)	−55°C ~ 25°C ~ 150°C 10 cycles	0/45
Temperature Cycling (2)	−20°C ~ 25°C ~ 125°C 500 cycles	0/22
Thermal Shock	−55°C ~ 125°C 15 cycles	0/22
Soldering heat	260°C, 10 sec	0/22
Solderability	230°C, 5 sec	0/22
Mechanical Shock	1500G, 0.5 msec 3 times/X, Y, Z	0/22
Vibration Variable Freq.	20 ~ 2000 Hz, 20G 3 times/X, Y, Z	0/22
Constant Acceleration	20000G 1 min/X, Y, Z	0/22

4. PRECAUTION

4.1 Storage

It is preferable to store semiconductor devices in the following ways to prevent detrioration in their electrical characteristics, solderability, and appearance, or breakage.

(1) Store in an ambient temperature of 5 to 30°C, and in a relative humidity of 40 to 60%.
(2) Store in a clean air environment, free from dust and active gas.
(3) Store in a container which does not induce static electricity.
(4) Store without any physical load.
(5) If semiconductor devices are stored for a long time, store them in the unfabricated form. If their lead wires are formed beforehand, bent parts may corrode during storage.
(6) If the chips are unsealed, store them in a cool, dry, dark, and dustless place. Assemble them within 5 days after unpacking. Storage in nitrogen gas is desirable. They can be stored for 20 days or less in dry nitrogen gas with a dew point at -30°C or lower. Unpacked devices must not be stored for over 3 months.
(7) Take care not to allow condensation during storage due to rapid temperature changes.

4.2 Transportation

As with storage methods, general precautions for other electronic component parts are applicable to the transportation of semiconductors, semiconductor-incorporating units and other similar systems. In addition, the following considerations must be given, too:

(1) Use containers or jigs which will not induce static electricity as the result of vibration during transportation. It is desirable to use an electrically conductive container or aluminium foil.
(2) In order to prevent device breakage from clothes-induced static electricity, workers should be properly grounded with a resistor while handling devices. The resistor of about 1 M ohm must be provided near the worker to protect from electric shock.
(3) When transporting the printed circuit boards on which semiconductor devices are mounted, suitable preventive measures against static electricity induction must be taken; for example, voltage built-up is prevented by shorting terminal circuit. When a belt conveyor is used, prevent the conveyor belt from being electrically charged by applying some surface treatment.
(4) When transporting semiconductor devices or printed circuit boards, minimize mechanical vibration and shock.

4.3 Handling for Measurement

Avoid static electricity, noise and surge-voltage when semiconductor devices are measured. It is possible to prevent breakage by shorting their terminal circuits to equalize electrical potential during transportation. However, when the devices are to be measured or mounted, their terminals are left open to provide the possibility that they may be accidentally touched by a worker, measuring instrument, work bench, soldering iron, belt conveyor, etc. The device will fail if it touches something which leaks current or has a static charge. Take care not to allow curve tracers, synchroscopes, pulse generators, D.C. stabilizing power supply units etc. to leak current through their terminals or housings.

Especially, while the devices are being tested, take care not to apply surge voltage from the tester, to attach a clamping circuit to the tester, or not to apply any abnormal voltage through a bad contact from a current source.

During measurement, avoid miswiring and short-circuiting. When inspecting a printed circuit board, make sure that no soldering bridge or foreign matter exists before turning on the power switch.

Since these precautions depend upon the types of semiconductor devices, contact Hitachi for further details.

4.4 Soldering

Semiconductor devices should not be left at high temperatures for a long time. Regardless of the soldering method, soldering must be done in a short time and at the lowest possible temperature. Soldering work must meet soldering heat test conditions, namely, 260°C for 10 seconds and 350°C for 3 seconds at a point 1 to 1.5 mm away from the end of the device package.

Use of a strong alkali or acid flux may corrode the leads, deteriorating device characteristics. The recommended soldering iron is the type that is operated with a secondary voltage supplied by a transformer and grounded to protect from lead current. Solder the leads at the farthest point from the device package.

4.5 Removing Residual Flux

To ensure the reliability of electronic systems, residual flux must be removed from circuit boards. Detergent or ultrasonic cleaning is usually applied. If chloric detergent is used for the plastic molded devices, package corrosion may occur. Since cleaning over extended periods or at high temperatures will cause swollen chip coating due to solvent permeation, select the type of detergent and cleaning condition carefully. Lotus Solvent and Dyfron Solvent are recommended as a detergent. Do not use any trichloroethylene solvent. For ultrasonic cleaning, the following conditions are advisable:

- Frequency: 28 to 29 kHz (to avoid device resonation)
- Ultrasonic output: 15W/ℓ
- Keep the devices out of direct contact with the power generator.
- Cleaning time: Less than 30 seconds

DATA SHEETS

8-BIT · 16-BIT PERIPHERAL LSI

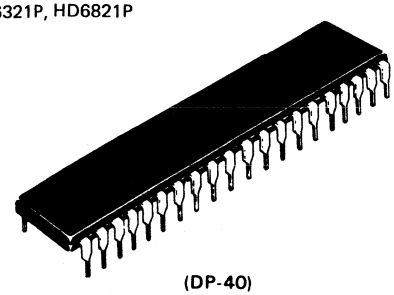
HD6321/HD6821

PIA (Peripheral Interface Adapter)

The HD6321/HD6821 Peripheral Interface Adapter provides the universal means of interfacing peripheral equipment to the HD6800 Microprocessing Unit (MPU). This device is capable of interfacing the MPU to peripherals through two 8-bit bi-directional peripheral data buses and four control lines. No external logic is required for interfacing to most peripheral devices.

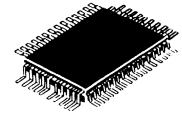
The functional configuration of the PIA is programmed by the MPU during system initialization. Each of the peripheral data lines can be programmed to act as an input or output, and each of the four control/interrupt lines may be programmed for one of several control modes. This allows a high degree of flexibility in the over-all operation of the interface.

HD6321P, HD6821P



(DP-40)

HD6321FP



(FP-54)

The specifications of the HD6321 are for preliminary and may change hereafter. Please make an inquire at sales office upon adoption of the HD6321.

FEATURES

- Two Bi-directional 8-bit Peripheral Data Bus for interface to Peripheral devices
- Two programmable control, Data Direction Registers
- Four Individually Controlled Interrupt Input Lines: Two Usable as Peripheral Control Outputs
- Handshake Control Logic for Input and Output Peripheral Operation
  CA_1, CA_2 ..... Port A ($PA_0 \sim PA_7$)
  CA_2, CB_2 ..... Port B ($PB_0 \sim PB_7$)
- High-Impedance 3-State and Direct Transistor Drive Peripheral Lines
- Two TTL Drive Capability on All A and B Side Buffers

–HD6321–

- Low-Power, High-Speed, High-Density CMOS
- Wide Range Operating Voltage (Vcc = 5V ± 10%)
- Compatible with NMOS PIA (HD6821) (Refer to Electrical Specification as to Minor difference)

–HD6821–

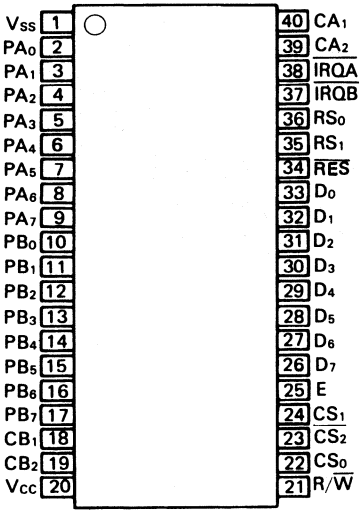
- Compatible with MC6821, MC68A21 and MC68B21

■ TYPE OF PRODUCTS

Type No.	Process	Clock Frequency	Package
HD6321	CMOS	1.0 MHz	DP-40
HD63A21		1.5 MHz	FP-54
HD63B21		2.0 MHz	
HD6821	NMOS	1.0 MHz	DP-40
HD68A21		1.5 MHz	
HD68B21		2.0 MHz	

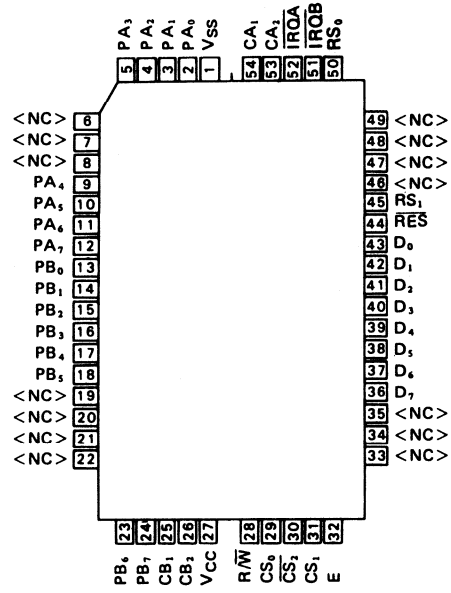
■ PIN ARRANGEMENT

● HD6321P, HD6821P



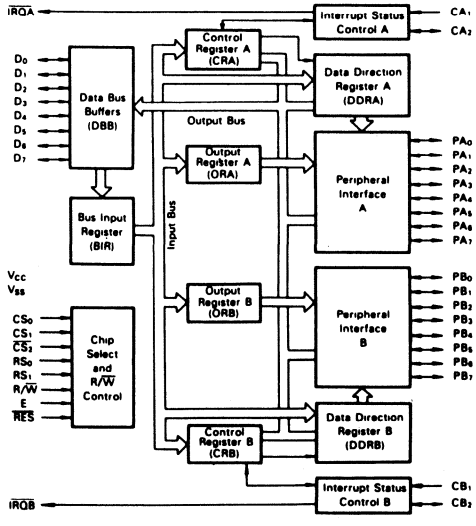
(Top View)

● HD6321FP



(Top View)

■ BLOCK DIAGRAM



■ ABSOLUTE MAXIMUM RATINGS

Item	Symbol	Value		Unit
		HD6321	HD6821	
Supply Voltage	V_{CC} *	−0.3 ~ +7.0	−0.3 ~ +7.0	V
Input Voltage	V_{in} *	−0.3 ~ +7.0	−0.3 ~ +7.0	V
Maximum Output Current	$\|I_o\|$ **	10	–	mA
Maximum Total Output Current	$\|\Sigma I_o\|$ ***	100	–	mA
Operating Temperature	T_{opr}	−20 ~ +75	−20 ~ +75	°C
Storage Temperature	T_{stg}	−55 ~ +150	−55 ~ +150	°C

* With respect to V_{SS} (SYSTEM GND)

** Maximum output current is the maximum current which can flow in or flow out from one output terminal and I/O common terminal, (PA_0~PA_7, CA_2, PB_0~PB_7, CB_2, D_0~D_7)

*** Maximum total output current is the total sum of output current which can flow in or flow out simultaneously from output terminals and I/O common terminals. (PA_0~PA_7, CA_2, PB_0~PB_7, CB_2, D_0~D_7)

(NOTE) Permanent LSI damage may occur if maximum ratings are exceeded. Normal operation should be under recommended operating conditions. If these conditions are exceeded, it could affect reliability of LSI.

■ RECOMMENDED OPERATING CONDITIONS

Item		Symbol	HD6321			HD6821			Unit
			min.	typ	max.	min.	typ	max.	
Supply Voltage		V_{CC} *	4.5	5.0	5.5	4.75	5.0	5.25	V
Input "Low" Voltage		V_{IL} *	−0.3	–	0.8	−0.3	–	0.8	V
Input "High" voltage	D_0~D_7, PA_0~PA_7, CA_1, CA_2 PB_0~PB_7 CB_1, CB_2	V_{IH} *	2.2	–	V_{CC}	2.0	–	V_{CC}	V
	E, R/$\overline{W}$, CS_0, $\overline{CS_2}$, CS_1, RS_0, RS_1, $\overline{RES}$		3.0 **		V_{CC}				
Operating Temperature		T_{opr}	−20	25	75	−20	25	75	°C

* With respect to V_{SS} (SYSTEM GND)

** Characteristics will be improved.

■ ELECTRICAL CHARACTERISTICS

● DC CHARACTERISTICS (HD6321; V_{CC} = 5V ±10%, HD6821; V_{CC} = 5V ±5%, V_{SS} = 0V, Ta = −20 ~ +75°C, unless otherwise noted.)

Item		Symbol	HD6321 Test Condition	HD6321 min.	HD6321 typ*	HD6321 max	HD6821 Test Condition	HD6821 min	HD6821 typ*	HD6821 max	Unit
Input "High" Voltage	D_0~D_7, PA_0~PA_7, CA_1 CA_2, PB_0~PB_7, CB_1, CB_2	V_{IH}		2.2	—	Vcc		2.0	—	Vcc	V
	E, R/$\overline{W}$, CS_0, $\overline{CS_2}$ CS_1, RS_0, RS_1, $\overline{RES}$			3.0**	—	Vcc					
Input "Low" Voltage	All Inputs	V_{IL}		−0.3	—	0.8		−0.3	—	0.8	V
Input Leakage Current	R/$\overline{W}$, $\overline{RES}$, $\overline{RS_0}$, RS_1, CS_0, CS_1, $\overline{CS_2}$, CA_1, CB_1, E	I_{in}	V_{in} = 0 ~ V_{cc}	−2.5	—	2.5	V_{in} = 0 ~ V_{cc}	−2.5	—	2.5	μA
Three State (Off State) Input Current	PA_0~PA_7, CA_2	I_{TSI}	V_{in} = 0.4~ V_{cc}	−10	—	10	V_{in} = 0.4~2.4V	−10		10	μA
	D_0~D_7, PB_0~PB_7, CB_2										
Input "High" Current	PA_0~PA_7, CA_2	I_{IH}					V_{IH} = 2.4V	−200	—	—	μA
Input "Low" Current	PA_0~PA_7, CA_2	I_{IL}					V_{IL} = 0.4V	—	—	−2.4	mA
Output "High" Voltage	D_0~D_7	V_{OH}	I_{OH} = −400μA	4.1	—	—	I_{OH} = −205μA	2.4	—	—	V
			$I_{OH} \leq$ −10μA	V_{cc} −0.1	—	—					
	PA_0~PA_7, CA_2 PB_0~PB_7, CB_2		I_{OH} = −400μA	4.1	—	—					
			I_{OH} < −10μA	V_{cc} −0.1	—	—					
	PA_0~PA_7, CA_2						I_{OH} = −200μA	2.4***	—	—	
	PB_0~PB_7, CB_2						I_{OH} = − 10μA	V_{cc} −0.1	—	—	
							I_{OH} = −200μA	2.4	—	—	
Output "Low" Voltage	D_0~D_7, $\overline{IRQA}$, $\overline{IRQB}$	V_{OL}	I_{OL} = 1.6mA	—	—	0.4	I_{OL} = 1.6mA	—	—	0.4	V
	PA_0~PA_7, CA_2		I_{OL} = 3.2mA	—	—	0.6	I_{OL} = 1.6mA	—	—	0.4	
	PB_0~PB_7, CB_2						I_{OL} = 3.2mA	—	—	0.6	
Output "High" Current	D_0 ~ D_7	I_{OH}					V_{OH} = 2.4V	−205	—	—	μA
	PA_0~PA_7, CA_2						V_{OH} = 2.4V***	−200	—	—	μA
	PB_0~PB_7, CB_2						V_{OH} = 1.5V	−1.0	—	−10	mA
Output Leakage Current (Off State)	$\overline{IRQA}$, $\overline{IRQB}$	I_{LOH}	V_{OH} = V_{cc}	—	—	10	V_{OH} = V_{cc}	—	—	10	μA
Input Capacitance	PA_0~PA_7, PB_0~PB_7 CA_2, CB_2, D_0~D_7	C_{in}	V_{in} = 0V Ta = 25°C f = 1.0MHz	—	—	12.5	V_{in} = 0V Ta = 25°C f = 1.0MHz	—	—	12.5	pF
	R/$\overline{W}$, $\overline{RES}$, $\overline{RS_0}$, RS_1 CS_0, CS_1, $\overline{CS_2}$, CA_1 CB_1, E			—	—	10		—	—	10	
Output Capacitance	$\overline{IRQA}$, $\overline{IRQB}$	C_{out}	V_{in} = 0V Ta = 25°C f = 1.0MHz	—	—	10	V_{in} = 0V Ta = 25°C f = 1.0MHz	—	—	10	pF
Supply Current ****	•PA_0~PA_7, CA_2, PB_0~PB_7, CB_2 are specified as input. •Chip is not selected. •Input level V_{IH} min = V_{cc} − 0.8V V_{IL} max = 0.8V	I_{cc}	E = 1.0MHz	—	—	300					μA
			E = 1.5MHz	—	—	400					
			E = 2.0MHz	—	—	500					
	•PA_0~PA_7, CA_2 and PB_0~PB_7, CB_2 are specified as input. •Under Data Bus R/$\overline{W}$ operation.		E = 1.0MHz	—	—	4					mA
			E = 1.5MHz	—	—	5					
			E = 2.0MHz	—	—	6					
Power Dissipation		P_D						—	260	550	mW

* Ta = 25°C, V_{CC} = 5.0V

** Characteristics will be improved.

*** HD68B21; V_{OH} = 2.2V min (PA_0~PA_7, CA_2)

**** Supply current is defined on the condition that there is no current flow from output terminals. Supply current will be increased when the current from output terminal exists. Also the current will be increased for charging and discharging the capacitive load. Please take this case into consideration in estimating system power.

• AC CHARACTERISTICS (HD6321; V_{CC} = 5V ± 10%, HD6821; V_{CC} = 5V ± 5%, V_{SS} = 0V, Ta = –20 ~ +75°C unless otherwise noted)

1. PERIPHERAL TIMING

Item		Symbol	Test Condition	HD6321		HD63A21		HD63B21		HD6821		HD68A21		HD68B21		Unit
				min	max	min	max	min	max	min	max	min	max	min	max	
Peripheral Data Setup Time		t_{PDSU}	Fig. 1	100	–	100	–	100	–	200	–	135	–	100	–	ns
Peripheral Data Hold Time		t_{PDH}	Fig. 1	0	–	0	–	0	–	0	–	0	–	0	–	ns
Delay Time, Enable negative Transition to CA_2 negative transition	Enable → CA_2 Negative	t_{CA2}	Fig. 2, Fig. 3	–	200	–	200	–	200	–	1000	–	670	–	500	ns
Delay Time, Enable negative transition to CA_2 positive transition	Enable→CA_2 Positive	t_{RS1}	Fig. 2	–	200	–	200	–	200	–	1000	–	670	–	500	ns
Rise and Fall Times for CA_1 and CA_2 input signals	CA_1, CA_2	t_r, t_f	Fig. 3	–	100	–	100	–	100	–	1000	–	1000	–	1000	ns
Delay Time from CA_1 active transition to CA_2 positive transition	CA_1 → CA_2	t_{RS2}	Fig. 3	–	300	–	300	–	300	–	2000	–	1350	–	1000	ns
Delay Time, Enable negative transition to Peripheral Data Valid	Enable→Peripheral Data	t_{PDW}	Fig. 4, Fig. 5	–	300	–	300	–	300	–	1000	–	670	–	500	ns
Delay Time, Enable negative transition to Peripheral CMOS Data Valid	Enable → Peripheral Data PA_0~PA_7, CA_2	t_{CMOS}	V_{CC}–30% V_{CC} Fig. 4							–	2000	–	1350	–	1000	ns
Delay Time, Enable positive transition to CB_2 negative position	Enable → CB_2	t_{CB2}	Fig. 6, Fig. 7	–	200	–	200	–	200	–	1000	–	670	–	500	ns
Delay Time, Peripheral Data Valid to CB_2 negative transition	Peripheral Data → CB_2	t_{DC}	Fig. 5	20	–	20	–	20	–	20	–	20	–	20	–	ns
Delay Time, Enable positive transition to CB_2 positive transition	Enable → CB_2	t_{RS1}	Fig. 6	–	200	–	200	–	200	–	1000	–	670	–	500	ns
Peripheral Control Output Pulse Width, CA_2/CB_2	CA_2 CB_2	PW_{CT}	Fig. 2, Fig. 6	550	–	375	–	250	–	550	–	550	–	500	–	ns
Rise and Fall Time for CB_1 and CB_2 input signals	CB_1, CB_2	t_r, t_f	Fig. 7	–	100	–	100	–	100	–	1000	–	1000	–	1000	ns
Delay Time, CB_1 active transition to CB_2 positive transition	CB_1 → CB_2	t_{RS2}	Fig. 7	–	300	–	300	–	300	–	2000	–	1350	–	1000	ns
Interrupt Release Time, $\overline{IRQA}$ and $\overline{IRQB}$	$\overline{IRQA}$, $\overline{IRQB}$	t_{IR}	Fig. 9	–	800	–	800	–	800	–	1600	–	1100	–	850	ns
Interrupt Response Time	$\overline{IROA}$, $\overline{IROB}$	t_{RS3}	Fig. 8	–	400	–	400	–	400	–	1000	–	1000	–	1000	ns
Interrupt Input Pulse Width	CA_1, CA_2, CB_1, CB_2	PWI	Fig. 8	1E cycle	–	1E cycle	–	1E cycle	–	500**	–	500**	–	500**	–	ns
Reset "Low" Time	$\overline{RES}$*	t_{RL}	Fig. 10	200	–	200	–	200	–	1000	–	660	–	500	–	ns

* The Reset line must be "High" a minimum of 1.0μs before addressing the PIA.

** At least one Enable "High" pulse should be included in this period.

2. BUS TIMING

1) READ

Item		Symbol	Test Condition	HD6321		HD63A21		HD63B21		HD6821		HD68A21		HD68B21		Unit
				min	max	min	max	min	max	min	max	min	max	min	max	
Enable Cycle Time		$t_{cyc}E$	Fig. 11	1000	–	666	–	500	–	1000	–	666	–	500	–	ns
Enable Pulse Width, "High"		PW_{EH}	Fig. 11	450	–	280	–	220	–	450	–	280	–	220	–	ns
Enable Pulse Width, "Low"		PW_{EL}	Fig. 11	430	–	280	–	210	–	430	–	280	–	210	–	ns
Enable Pulse Rise and Fall Times		t_{Er}, t_{Ef}	Fig. 11	–	25	–	25	–	20	–	25	–	25	–	25	ns
Setup Time	Address, R/$\overline{W}$-Enable	t_{AS}	Fig. 12	80	–	60	–	60*	–	140	–	140	–	70	–	ns
Address Hold Time		t_{AH}	Fig. 12	10	–	10	–	10	–	10	–	10	–	10	–	ns
Data Delay Time		t_{DDR}	Fig. 12	–	290	–	180	–	150	–	320	–	220	–	180	ns
Data Hold Time		t_{DHR}	Fig. 12	20	100	20	100	20	100	10	–	10	–	10	–	ns

* Characteristics will be improved.

2) WRITE

Item		Symbol	Test Condition	HD6321		HD63A21		HD63B21		HD6821		HD68A21		HD68B21		Unit
				min	max	min	max	min	max	min	max	min	max	min	max	
Enable Cycle Time		$t_{cyc}E$	Fig. 11	1000	–	666	–	500	–	1000	–	666	–	500	–	ns
Enable Pulse Width, "High"		PW_{EH}	Fig. 11	450	–	280	–	220	–	450	–	280	–	220	–	ns
Enable Pulse Width, "Low"		PW_{EL}	Fig. 11	430	–	280	–	210	–	430	–	280	–	210	–	ns
Enable Pulse Rise and Fall Times		t_{Er}, t_{Ef}	Fig. 11	–	25	–	25	–	20	–	25	–	25	–	25	ns
Setup Time		t_{AS}	Fig. 13	80	–	60	–	60*	–	140	–	140	–	70	–	ns
Address Hold Time	Address, R/$\overline{W}$-Enable	t_{AH}	Fig. 13	10	–	10	–	10	–	10	–	10	–	10	–	ns
Data Setup Time		t_{DSW}	Fig. 13	165	–	80	–	60	–	195	–	80	–	60	–	ns
Data Hold Time		t_{DHW}	Fig. 13	10	–	10	–	10	–	10	–	10	–	10	–	ns

* Characteristics will be improved.

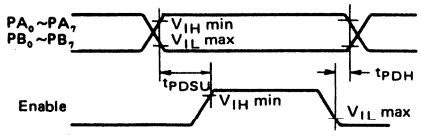
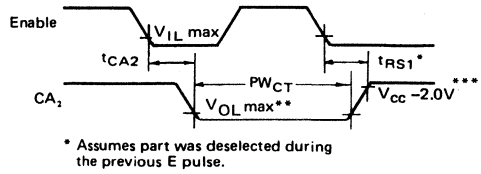


Figure 1 Peripheral Data Setup and Hold Times (Read Mode)



* Assumes part was deselected during the previous E pulse.

Figure 2 CA_2 Delay Time
(Read Mode; CRA5=CRA3=1, CRA4=0)

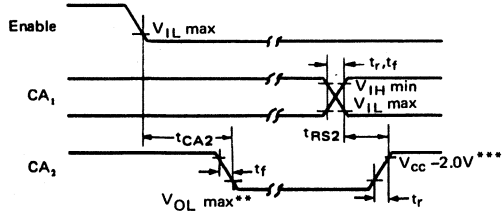


Figure 3 CA_2 Delay Time
(Read Mode; CRA5=1, CRA3=CRA4=0)

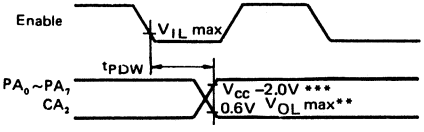
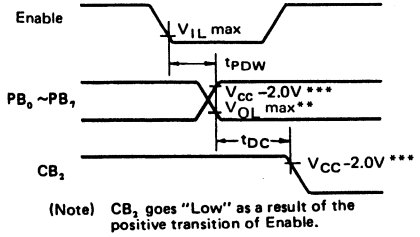
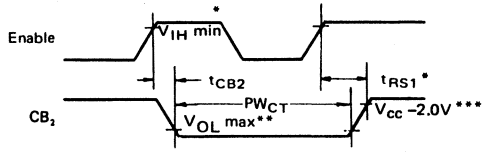


Figure 4 Peripheral Data Delay Times
(Write Mode; CRA5=CRA3=1, CRA4=0)



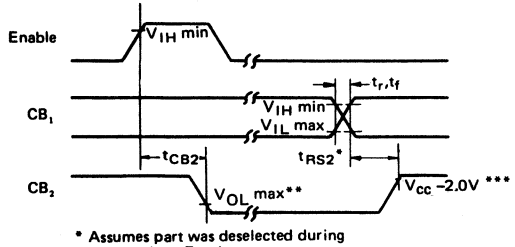
(Note) CB_2 goes "Low" as a result of the positive transition of Enable.

Figure 5 Peripheral Data and CB_2 Delay Times
(Write Mode; CRB5=CRB3=1, CRB4=0)



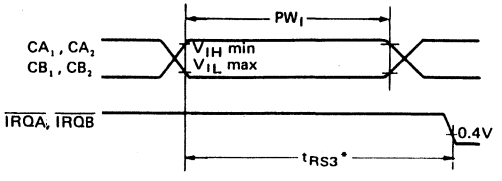
* Assumes part was deselected during the previous E pulse.

Figure 6 CB_2 Delay Time
(Write Mode; CRB5=CRB3=1, CRB4=0)



* Assumes part was deselected during any previous E pulse.

Figure 7 CB_2 Delay Time
(Write Mode; CRB5=1, CRB3=CRB4=0)



* Assumes Interrupt Enable Bits are set.

Figure 8 Interrupt Pulse Width and $\overline{IRQ}$ Response

** 0.6V for HD6321, 0.4V for HD6821

*** 2.4V for HD6821

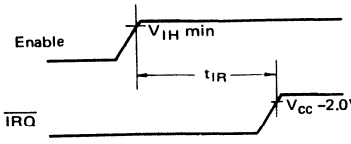
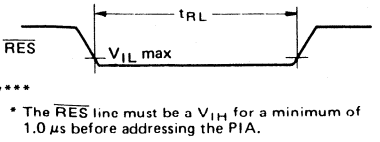


Figure 9 $\overline{IRQ}$ Release Time



* The $\overline{RES}$ line must be a V_{IH} for a minimum of 1.0 μs before addressing the PIA.

Figure 10 $\overline{RES}$ Low Time

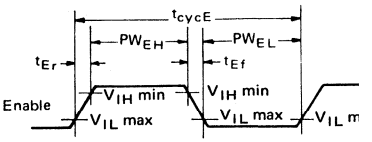


Figure 11 Enable Signal Characteristics

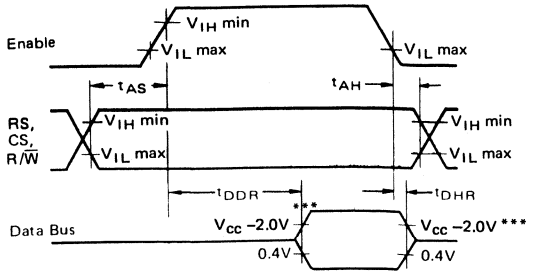


Figure 12 Bus Read Timing Characteristics (Read Information from PIA)

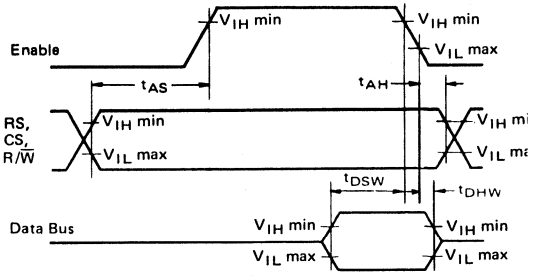
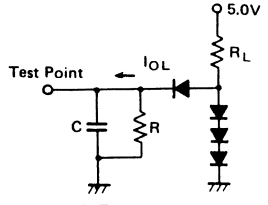


Figure 13 Bus Write Timing Characteristics (Write Information into PIA)

** 0.6V for HD6321, 0.4V for HD6821.
*** 2.4V for HD6821.

LOAD A
(PA_0~PA_7, PB_0~PB_7, CA_2, CB_2)

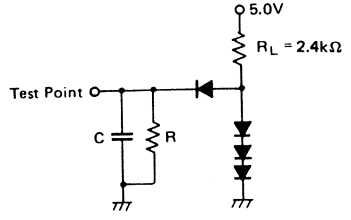


C=40pF
R=10kΩ (HD6321), 12kΩ (HD6821)

All diodes are1S2074Ⓗor equivalent.

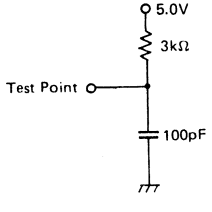
Adjust R_L so that I_{OL} = 1.6mA, then test V_{OL}
Adjust R_L so that I_{OL} = 3.2mA, then test V_{OL}

LOAD B
(D_0~D_7)



C=130pF
R=10kΩ (HD6321), 11kΩ (HD6821)

LOAD C
($\overline{IRQ}$ Only)



LOAD D (HD6821)

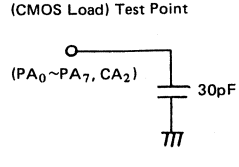


Figure 14 Bus Timing Test Loads

■ PIA INTERFACE SIGNALS FOR MPU

The PIA interfaces to the HD6800 MPU with an eight-bit bi-directional data bus, three chip select lines, two register select lines, two interrupt request lines, read/write line, enable line and reset line. These signals, in conjunction with the HD6800 VMA output, permit the MPU to have complete control over the PIA. VMA should be utilized in conjunction with an MPU address line into a chip select of the PIA.

- **Bi-Directional Data ($D_0 \sim D_7$)**

Input	Pin No. 33 ~ 26 (DP-40) Pin No. 43 ~ 36 (FP-54)

The bi-directional data lines ($D_0 \sim D_7$) allow the transfer of data between the MPU and the PIA. The data bus output drivers are three-state devices that remain in the high-impedance (off) state except when the MPU performs a PIA read operation. The R/$\overline{W}$ line is in the Read ("High") state when the PIA is selected for a Read operation.

- **Enable (E)**

Input/Output	Pin No. 25 (DP-40) Pin No. 32 (FP-54)

The enable pulse, E, is the only timing signal that is supplied to the PIA. Timing of all other signals is referenced to the leading and trailing edges of the E pulse. This signal will normally be a derivative of the HMCS6800 System ϕ_2 Clock. This signal must be continuous clock pulse.

- **Read/Write (R/$\overline{W}$)**

Input	Pin No. 21 (DP-40) Pin No. 28 (FP-54)

This signal is generated by the MPU to control the direction of data transfers on the Data Bus. A "Low" state on the PIA line enables the input buffers and data is transferred from the MPU to the PIA on the E signal if the device has been selected. A "High" on the R/$\overline{W}$ line sets up the PIA for a transfer of data to the bus. The PIA output buffers are enabled when the proper address and the enable pulse E are present.

- **Reset ($\overline{RES}$)**

Input	Pin No. 34 (DP-40) Pin No. 44 (FP-54)

The active "Low" $\overline{RES}$ line is used to reset all register bits in the PIA to a logical zero "Low". This line can be used as a power-on reset and as a master reset during system operation.

- **Chip Select (CS_0, CS_1 and $\overline{CS_2}$)**

Input	Pin No. 22, 24, 23 (DP-40) Pin No. 29, 31, 30 (FP-54)

These three input signals are used to select the PIA. CS_0 and CS_1 must be "High" and $\overline{CS_2}$ must be "Low" for selection of the device. Data transfers are then performed under the control of the E and R/$\overline{W}$ signals. The chip select lines must be stable for the duration of the E pulse. The device is deselected when any of the chip selects are in the inactive state.

- **Register Select (RS_0 and RS_1)**

Input	Pin No. 36, 35 (DP-40) Pin No. 50, 45 (FP-54)

The two register select lines are used to select the various registers inside the PIA. These two lines are used in conjunction with internal Control Registers to select a particular register that is to be written or read.

The register and chip select lines should be stable for the duration of the E pulse while in the read or write cycle.

- **Interrupt Request ($\overline{IRQA}$ and $\overline{IRQB}$)**

Input	Pin No. 38, 37 (DP-40) Pin No. 52, 51 (FP-54)

The active "Low" Interrupt Request lines ($\overline{IRQA}$ and $\overline{IRQB}$) act to interrupt the MPU either directly or through interrupt priority circuitry. These lines are "open drain" (no load device on the chip). This permits all interrupt request lines to be tied together in a wire-OR configuration.

Each $\overline{IRQ}$ line has two internal interrupt flag bits that can cause the $\overline{IRQ}$ line to go "Low". Each flag bit is associated with a particular peripheral interrupt line. Also four interrupt enable bits are provided in the PIA which may be used to inhibit a particular interrupt from a peripheral device.

Servicing an interrupt by the MPU may be accomplished by a software routine that, on a prioritized basis, sequentially reads and tests the two control registers in each PIA for interrupt flag bits that are set.

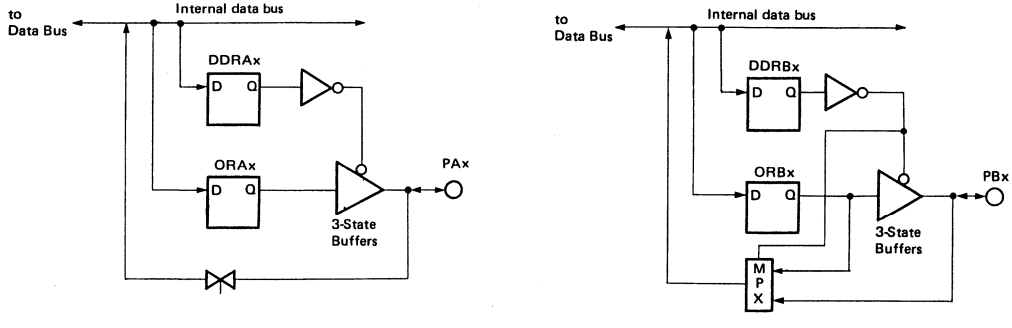
The interrupt flags are cleared (zeroed) as a result of an MPU Read Peripheral Data Operation of the corresponding data register. After being cleared, the interrupt flag bit cannot be enabled to be set until the PIA is deselected during an E pulse. The E pulse is used to condition the interrupt control lines (CA_1, CA_2, CB_1, CB_2). When these lines are used as interrupt inputs at least one E pulse must occur from the inactive edge to the active edge of the interrupt input signal to condition the edge sense network. If the interrupt flag has been enabled and the edge sense circuit has been properly conditioned, the interrupt flag will be set on the next active transition of the interrupt input pin.

■ PIA PERIPHERAL INTERFACE LINES

The PIA provides two 8-bit bi-directional data buses and four interrupt/control lines for interfacing to peripheral devices.

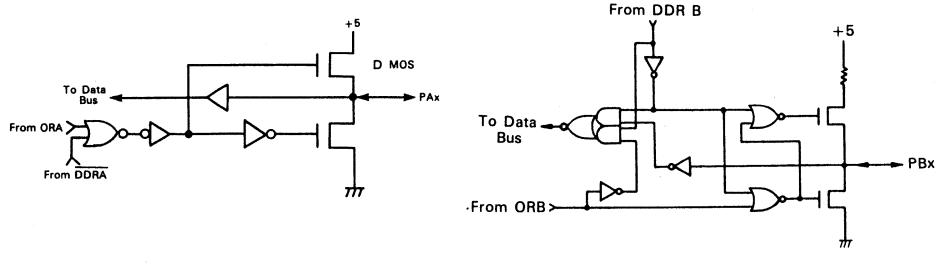
There is difference between HD6821 and HD6321 in Port structure. Fig. 15 shows the block diagram of Port A and Port B in HD6321. The output drivers of Port A and Port B consist of three-state drivers, allowing them to enter a High-impedance state when the peripheral data line is used as an input. Port A and Port B have the same output buffer. But the circuit configuration is slightly different and this makes the difference on data flow when MPU reads Port A and Port B in the case each Port is specified as output. As shown in Fig. 15, the output of the peripheral data A is transferred to internal data bus when used as output. On the other hand, in the case of Port B the contents of output register (ORB) is directly transferred to internal data bus through the multiplexor.

Secondly the equivalent circuit of the port in HD6821 is shown in Fig. 16. The output circuits of A port is different from that of B port. When the port is used as input, the input is pullup to V_{CC} side through load MOS in A port and B port becomes "Off" (high impedance).



(a) Port A (b) Port B

Figure 15 Block Diagram of Port A and Port B (HD6321)



(a) Port A (b) Port B

Figure 16 Circuit of Port A and Port B (HD6821)

- **Port A Peripheral Data (PA_0~PA_7)**

Input/Output	Pin No. 2~ 9 (DP-40) Pin No. 2~ 5, 9~ 12 (FP-54)

Each of the peripheral data lines can be programmed to act as an input or output. This is accomplished by setting a "1" in the corresponding Data Direction Register bit for those lines which are to be outputs. A "0" in a bit of the Data Direction Register causes the corresponding peripheral data line to act as an input. During an MPU Read Peripheral Data Operation, the data on peripheral lines programmed to act as inputs appears directly on the corresponding MPU Data Bus lines.

The data in Output Register A will appear on the peripheral data lines that are programmed to be outputs. A logical "1" written into the register will cause a "High" on the corresponding peripheral data line while a "0" results in a "Low". Data in Output Register A may be read by an MPU "Read Peripheral Data A" operation when the corresponding lines are programmed as outputs.

But concerning HD6821, this data will be read properly if the voltage on the peripheral data lines is greater than 2.0 volts for a logic "1" output and less than 0.8 volt for a logic "0" output. Loading the output lines such that the voltage on these lines does not reach full voltage causes the data transferred into the MPU on a Read operation to differ from that contained in the respective bit of Output Register A.

- **Port B Peripheral Data (PB_0 ~ PB_7)**

Input/Output	Pin No. 10 ~ 17 (DP-40) Pin No. 13 ~ 18, 23 ~ 24 (FP-54)

Each of the Port B peripheral data bus can be programmed to act as an input or output like PA_0 ~ PA_7.

PB_0 ~ PB_7 are in High-impedance condition because they are three-state outputs just like PA_0 ~ PB_0 when the peripheral buses are used as inputs, when programmed as outputs, MPU read of Port B make it possible to read the output register regardless of PB_0 ~ PB_7 loads and concerning HD6821, these line may be used as a source of up to 2.5 milliampare (typ.) at 1.5 volt to directly drive the base of transistor switch.

- **Interrupt Input (CA_1 and CB_1)**

Input	Pin No. 40, 18 (DP-40) Pin No. 54, 25 (FP-54)

The peripheral Input lines CA_1, and CB_1 are input only lines that set the interrupt flags of the control registers. The active transition for these signals is also programmed by the two control registers.

- **Peripheral Control (CA_2)**

Input/Output	Pin No. 39 (DP-40) Pin No. 53 (FP-54)

The peripheral control line CA_2 can be programmed to act as an interrupt input or as a peripheral control output.

The function of this signal is programmed by the Control Register A. When used as an input, this signal is in High-impedance state.

- **Peripheral Control (CB_2)**

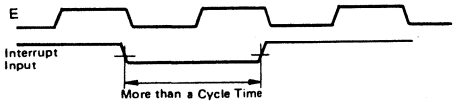
Input/Output	Pin No. 19 (DP-40) Pin No. 26 (FP-54)

The peripheral Control line CB_2 may also be programmed to act as an interrupt input or peripheral control output.

This line is programmed by Control Register B.

When used as an input, this signal is in High-impedance.

(NOTE) 1. Pulse width of interrupt inputs CA_1, CA_2, CB_1 and CB_2 shall be greater than a E cycle time. In the case that "High" time of E signal is not contained in Interrupt pulse, an interrupt flag may not be set.



■ INTERNAL CONTROLS

There are six locations within the PIA accessible to the MPU data bus: two Peripheral Registers, two Data Direction Registers, and two Control Registers. Selection of these locations is controlled by the RS_0 and RS_1 inputs together with bit 2 in the Control Register, as shown in Table 1.

Table 1 Internal Addressing

RS_1	RS_0	Control Register Bit CRA2	Control Register Bit CRB2	Location Selected
0	0	1	x	Peripheral Register A*
0	0	0	x	Data Direction Register A
0	1	x	x	Control Register A
1	0	x	1	Peripheral Register B*
1	0	x	0	Data Direction Register B
1	1	x	x	Control Register B

x = Don't Care
* Peripheral interface register is a generic term containing peripheral data bus and output register.

• Initialization

A "Low" reset line has the effect of zeroing all PIA registers. This will set $PA_0 \sim PA_7$, $PB_0 \sim PB_7$, CA_2 and CB_2 as inputs, and all interrupts disabled. The PIA must be configured during the restart program which follows the reset.

Details of possible configurations of the Data Direction and Control Register are as follows.

• Data Direction Registers (DDRA and DDRB)

The two Data Direction Registers allow the MPU to control the direction of data through each corresponding peripheral data line. A Data Direction Register bit set at "0" configures the corresponding peripheral data line as an input; a "1" results in an output.

• Control Registers (CRA and CRB)

The two Control Registers (CRA and CRB) allow the MPU to control the operation of the four peripheral control lines CA_1, CA_2, CB_1 and CB_2. In addition they allow the MPU to enable the interrupt lines and monitor the status of the interrupt flags. Bits 0 through 5 of the two registers may be written or read by the MPU when the proper chip select and register select signals are applied. Bits 6 and 7 of the two registers are read only and are modified by external interrupts occurring on control lines CA_1, CA_2, CB_1 or CB_2. The format of the control words is shown in Table 2.

Table 2 Control Word Format

	7	6	5	4	3	2	1	0
CRA	IRQA1	IRQA2	CA_2 Control			DDRA Access	CA_1 Control	
	7	6	5	4	3	2	1	0
CRB	IRQB1	IRQB2	CB_2 Control			DDRB Access	CB_1 Control	

Data Direction Access Control Bit (CRA2 and CRB2)

Bit 2 in each Control register (CRA and CRB) allows selection of either a Peripheral Interface Register or the Data Direction Register when the proper register select signals are applied to RS_0 and RS_1.

Interrupt Flags (CRA6, CRA7, CRB6, and CRB7)

The four interrupt flag bits are set by active transitions of signals on the four Interrupt and Peripheral Control lines when those lines are programmed to be inputs. These bits cannot be set directly from the MPU Data Bus and are reset indirectly by a Read Peripheral Data Operation on the appropriate section.

Control of CA_1 and CB_1 Interrupt Lines (CRA0, CRB0, CRA1, and CRB1)

The two lowest order bits of the control registers are used to control the interrupt input lines CA_1 and CB_1. Bits CRA0 and CRB0 are used to enable the MPU interrupt signals $\overline{IRQA}$ and $\overline{IRQB}$, respectively. Bits CRA1 and CRB1 determine the active transition of the interrupt input signals CA_1 and CB_1 (Table 3).

Control of CA_2 and CB_2 Peripheral Control Lines (CRA3, CRA4, CRA5, CRB3, CRB4, and CRB5)

Bits 3, 4 and 5 of the two control registers are used to control the CA_2 and CB_2 Peripheral Control lines. These bits determine if the control lines will be an interrupt input or an output control signal. If bit CRA5 (CRB5) is "0" CA_2 (CB_2) is an interrupt input line similar to CA1 (CB1) (Table 4). When CRA5 (CRB5) is "1", CA_2 (CB_2) becomes an output signal that may be used to control peripheral data transfers. When in the output mode, CA_2 and CB_2 have slightly different characteristics (Table 5 and 6).

Table 3 Control of Interrupt Inputs CA_1 and CB_1

CRA1 (CRB1)	CRA0 (CRB0)	Interrupt Input CA_1 (CB_1)	Interrupt Flag CRA7 (CRB7)	MPU Interrupt Request $\overline{IRQA}$ ($\overline{IRQB}$)
0	0	↓ Active	Set "1" on ↓ of CA_1 (CB_1)	Disabled – $\overline{IRQ}$ remains "High"
0	1	↓ Active	Set "1" on ↓ of CA_1 (CB_1)	Goes "Low" when the interrupt flag bit CRA7 (CRB7) goes "1"
1	0	↑ Active	Set "1" on ↑ of CA_1 (CB_1)	Disabled – $\overline{IRQ}$ remains "High"
1	1	↑ Active	Set "1" on ↑ of CA_1 (CB_1)	Goes "Low" when the interrupt flag bit CRA7 (CRB7) goes "1"

(Notes)
1. ↑ indicates positive transition ("Low" to "High")
2. ↓ indicates negative transition ("High" to "Low")
3. The Interrupt flag bit CRA7 is cleared by an MPU Read of the A Peripheral Register and CRB7 is cleared by an MPU Read of the B Peripheral Register.
4. If CRA0 (CRB0) is "0" when an interrupt occurs (Interrupt disabled) and is later brought "1", $\overline{IRQA}$ ($\overline{IRQB}$) occurs after CRA0 (CRB0) is written to a "1".

Table 4 Control of CA_2 and CB_2 as Interrupt Inputs – CRA5 (CRB5) is "0"

CRA5 (CRB5)	CRA4 (CRB4)	CRA3 (CRB3)	Interrupt Input CA_2 (CB_2)	Interrupt Flag CRA6 (CRB6)	MPU Interrupt Request $\overline{IRQA}$ ($\overline{IRQB}$)
0	0	0	↓ Active	Set "1" on ↓ of CA_2 (CB_2)	Disabled – $\overline{IRQ}$ remains "High"
0	0	1	↓ Active	Set "1" on ↓ of CA_2 (CB_2)	Goes "Low" when the interrupt flag bit CRA6 (CRB6) goes "1"
0	1	0	↑ Active	Set "1" on ↑ of CA_2 (CB_2)	Disabled – $\overline{IRQ}$ remains "High"
0	1	1	↑ Active	Set "1" on ↑ of CA_2 (CB_2)	Goes "Low" when the interrupt flag bit CRA6 (CRB6) goes "1"

(Notes)
1. ↑ indicates positive transition ("Low" to "High")
2. ↓ indicates negative transition ("High" to "Low")
3. The interrupt flag bit CRA6 is cleared by an MPU Read of the A Peripheral Register and CRB6 is cleared by an MPU Read of the B Peripheral Register.
4. If CRA3 (CRB3) is "0" when an interrupt occurs (Interrupt disabled) and is later brought "1", $\overline{IRQA}$ ($\overline{IRQB}$) occurs after CRA3 (CRB3) is written to a "1".

Table 5 Control of CB_2 as an Output – CRB5 is "1"

CRB5	CRB4	CRB3	CB_2 Cleared	CB_2 Set
1	0	0	"Low" on the positive transition of the first E pulse after MPU Write "B" Data Register operation.	"High" when the interrupt flag bit CRB7 is set by an active transition of the CB_1 signal. (See Figure 16)
1	0	1	"Low" on the positive transition of the first E pulse after an MPU Write "B" Data Register operation.	"High" on the positive edge of the first "E" pulse following an "E" pulse which occurred while the part was deselected. (See Figure 16)
1	1	0	"Low" (The content of CRB3 is output on CB_2)	
1	1	1	"High" (The content of CRB3 is output on CB_2)	

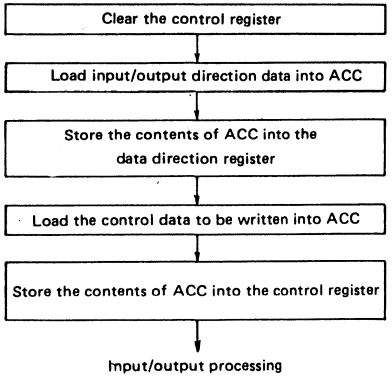
Table 6 Control of CA_2 as an Output – CRA5 is "1"

CRA5	CRA4	CRA3	CA_2	
			Cleared	Set
1	0	0	"Low" on negative transition of E after an MPU Read "A" Data Operation.	"High" when the interrupt flag bit CRA7 is set by an active transition of the CA_1 signal. (See Figure 16)
1	0	1	"Low" on negative transition of E after an MPU Read "A" Data operation.	"High" on the negative edge of the first "E" pulse which occurs during a deselect. (See Figure 16)
1	1	0	"Low" (The content of CRA3 is output on CA_2)	
1	1	1	"High" (The content of CRA3 is output on CA_2)	

■ PIA OPERATION

• Initialization

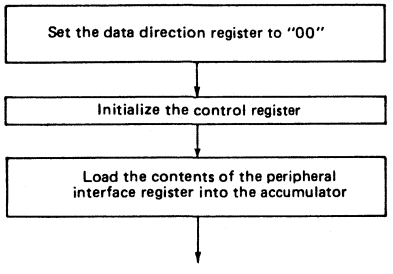
When the external reset input $\overline{RES}$ goes "Low", all internal registers are cleared to "0". Periperal data port (PA_0~PA7, PB_0~PB_7) is defined to be input and control lines (CA_1, CA_2, CB_1 and CB_2) are defined to be the interrupt input lines. PIA is also initialized by software sequence as follows.



- Program the data direction register access bit of the control register to "0" to allow to access the dada direction register.
- The data of the control line function is set into the accumulator, of which Data Direction Register Access Bit shall be programmed to "1".
- Transfer the control data from the accumulator into the control register.

• Read/Write Operation Not Using Control Lines

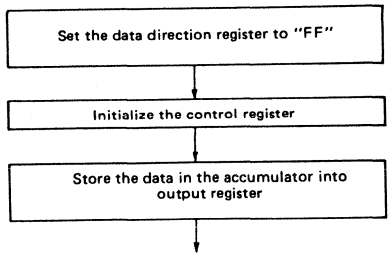
<Read Operation>




```
CLR   CRA
CLR   DDRA
LDAA  #$04
STAA  CRA
      .
      .
LDAA  PIRA
```

- Clear the DDRA access bit of the control register to "0".
- Clear all bits of the dada direction register.
- Set DDRA access bit of the control register to "1" to allow to access the peripheral interface register.

<Write Operation>



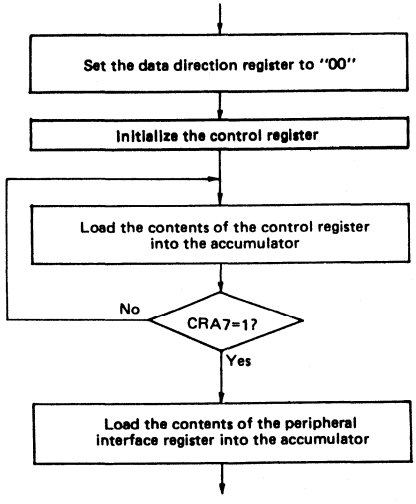

```
CLR   CRA     • Set DDRB access bit of the control register to "0".
LDAA  #$FF    • Set all bits of the data direction register to "FF".
STAA  DDRB
LDAA  #$04    • Set DDRB access bit of the control register to "1" to allow to
STAA  CRB       access the peripheral interface register.
LDAA  DATA    • Write the data into the peripheral interface register.
STAA  PIRB
```

Read/Write Operating Using Control Lines

Read/write request from peripherals shall be put into the control lines as an interrupt signal, and then MPU reads or writes after detecting interrupt request.

< Read >

The following case is that Port A is used and that the rising edge of CA_1 indicates the request for read from peripherals.



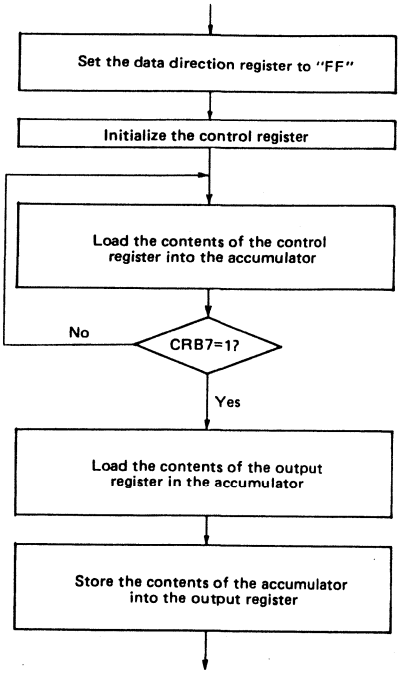

```
CLR   CRA     • Set the DDRA access bit to "0".
CLR   DDRA    • Set all bits of the data direction register to "0".
LDAA  #$06    • Program the rising edge of CA1 to be active. IRQA is masked
STAA  CRA       and DDRA access bit is set to 1.
LOOP  LDAA CRA
      BPL LOOP  • Check whether the read request comes from peripherals
                  or not.
LDAA  PIRA    • Load the data from the peripheral interface register into the
                accumulator. CRA flag is reset after this read operation.
```

To read the peripheral data, the data is directly transfered to the data buses D_0~D_7 through PA_0~PA_7 or PB_0~PB_7 and they are not latched in the PIA. If necessary, the data should be held in the external latch until MPU completes reading it.

When initializing the control register, interrupt flag bit (CRA7, CRA6, CRB7, CRB6) cannot be written from MPU. If necessary the interrupt flag must be reset by dummy read of Peripheral Register A and B.

<Write>

Write operation using the interrupt signal is as follows. In this case, B port is used and interrupt request is input to CB_1. And the $\overline{IRQ}$ flag is set at the rising edge of CB_1.




```
CLR   CRB
LDAA  #$FF
STAA  DDRB
```
• Set DDRB access bit to "0"
• Set all bits of DDRB to output "1".

```
LDAA  #$06
STAA  CRB
```
• Program the rising edge of CB_1 to be active. $\overline{IRQB}$ is mask and DDRB access bit is set to "1".

```
LOOP  LDAA  CRB
      BPL   LOOP
```
• Check whether the write request comes from peripherals or not.

```
LDAA  PIRB
```
• Reset the CRB7 flag by the dummy read of the peripheral interface register.

```
STAB  PIRB
```
• Store the data of the accumulator B to the peripheral interface register.

Interrupt request flag bits (CRA7, CRA6, CRB7 and CRB6) cannot be written and they cannot be also reset by write operation to the peripheral interface register. So dummy read of peripheral interface register is needed to reset the flags.

To accept the next interrupt, it is essential to reset indirectly the interrupt flag by dummy read of peripheral interface register.

Software poling method mentioned above requires MPU to continuously monitor the control register to detect the read/write request from peripherals. So other programs cannot run at the same time. To avoid this problem, hardware interrupt may be used. The MPU is interrupted by $\overline{IRQA}$ or $\overline{IRQB}$ when the read/write request is occurred from peripherals and then MPU analyzes cause of the interrupt request during interrupt processing.

- **Handshake Mode**

The functions of CRA and CRB are similar but not identical in the hand-shake modes. Port A is used for read hand-shake operation and Port B is used for write hand-shake mode.

CA_1 and CB_1 are used for interrupt input requests and CA_2 and CB_2 are control outputs (answer) in hand-shake mode.

Fig. 17, Fig. 18 and Fig. 19 show the timing of hand-shake mode.

< Read Hand-shake Mode>

CRA5="1", CRA4="0" and CRA3="0"

① A peripheral device puts the 8-bit data on the peripheral data lines after the control output CA_2 goes "Low".

② The peripheral requests MPU to read the data by using CA_1 input.

③ CRA7 flag is set and CA_2 becomes "High" (CA_2 automatically becomes "High" by the interrupt CA_1). This indicates the peripheral to maintain the current data and not to transfer the next data.

④ MPU accepts the read request by $\overline{IRQA}$ hardware interrupt or CRA read. Then MPU reads the peripheral register A.

⑤ CA_2 goes "Low" on the following edge of read Enable pulse. This informs that the peripheral can set the next data to port A.

<Write Hand-shake>

CRB5 = "1", CRB4 = "0" and CRB3 = "0"

① A peripheral device requests MPU to write the data by using CB_1 input. CB_2 output remains "High" until MPU write data to the peripheral interface register.

② CRB7 flag is set and MPU accepts the write request.

③ MPU reads the peripheral interface register to reset CRB7 (dummy read).

④ Then MPU write data to the peripheral interface register. The data is output to port B through the output register.

⑤ CB_2 automatically becomes "Low" to tell the peripheral that new data is on port B.

⑥ The peripheral read the data on Port B peripheral data lines and set CB_1 to "Low" to tell MPU that the data on the peripheral data lines has been taken and that next data can be written to the peripheral interface register.

<Pulse mode>

CRA5 = "1", CRA4 = "0" and CRA3 = "1"
CRB5 = "1", CRB4 = "0" and CRB3 = "1"
This mode is shown in Figure 17, Figure 20 and Figure 21.

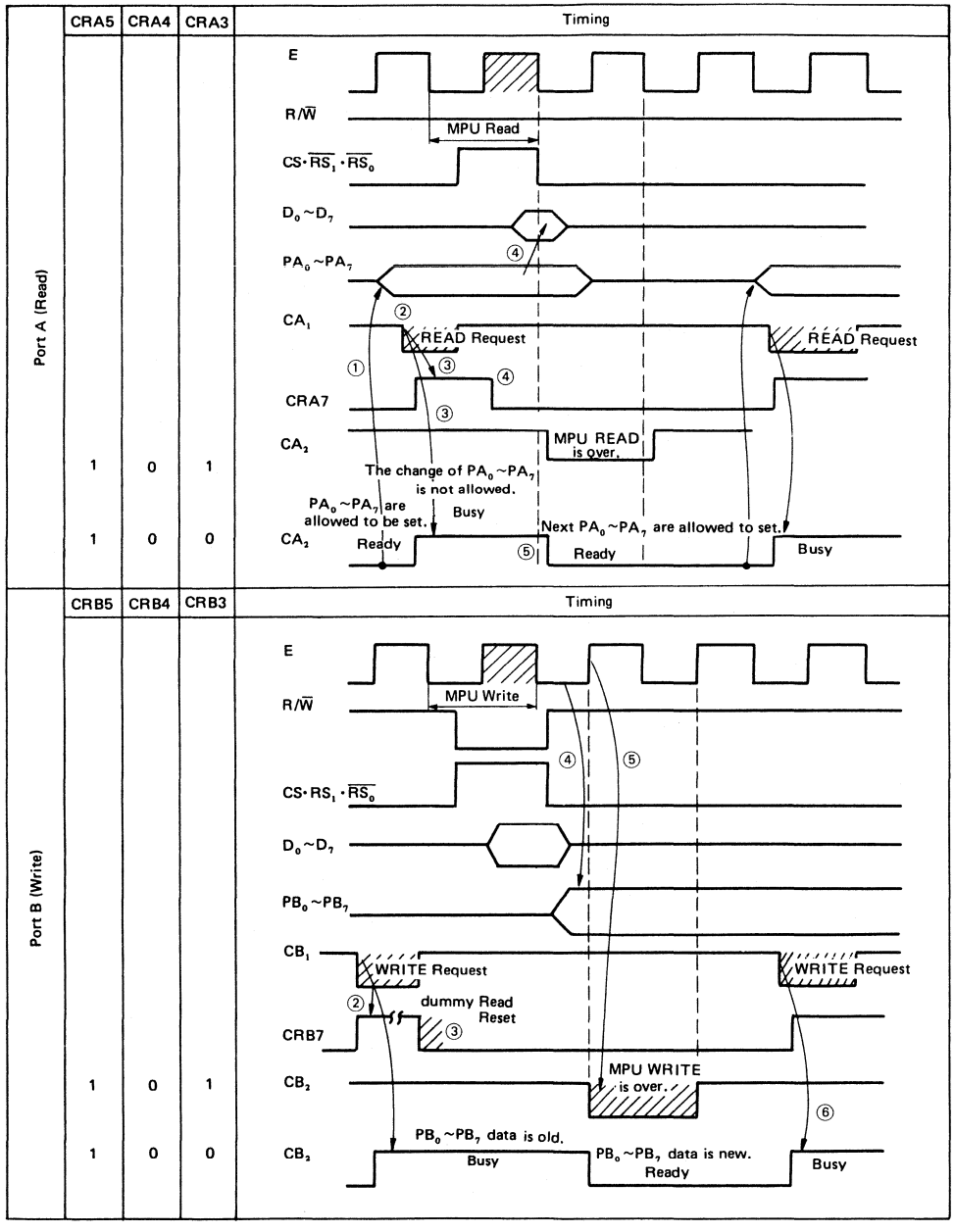


Figure 17 Timing of Hand-shake Mode and Pulse Mode

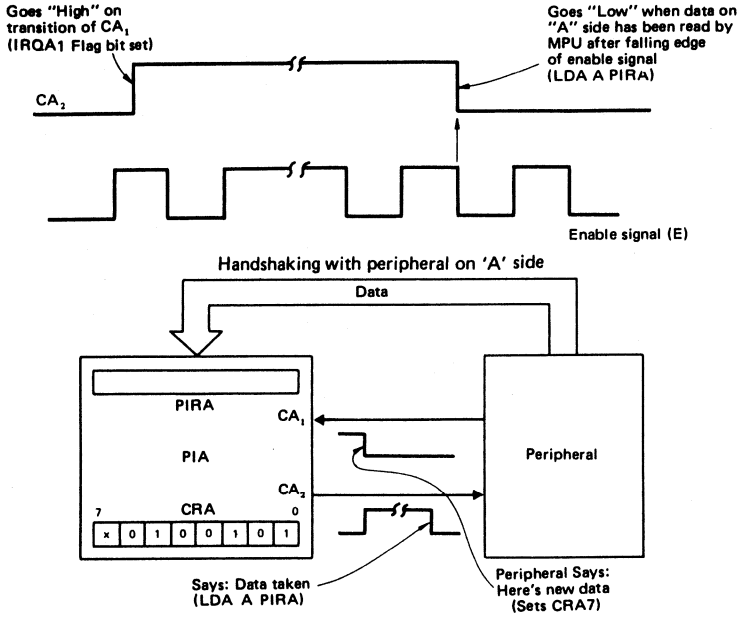


Figure 18 Bits 5, 4, 3 of CRA = 100 (Hand-shake Mode)

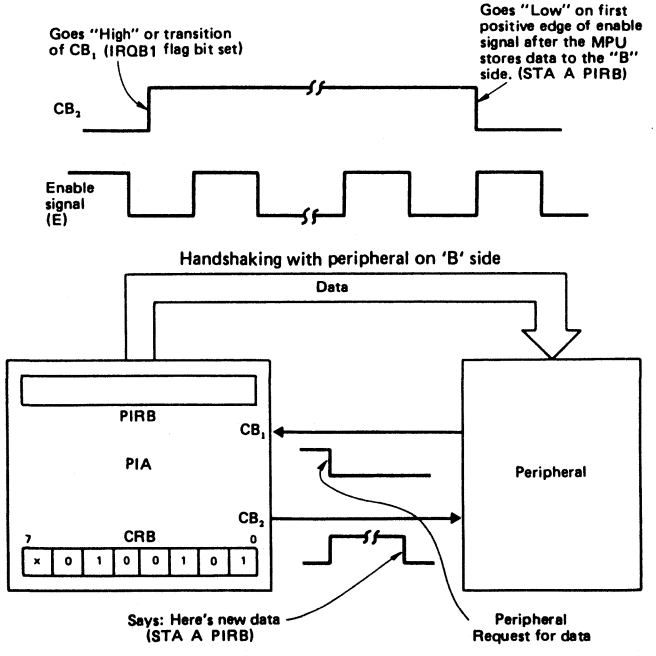


Figure 19 Bits 5, 4, 3 of CRB = 100 (Hand-shake Mode)

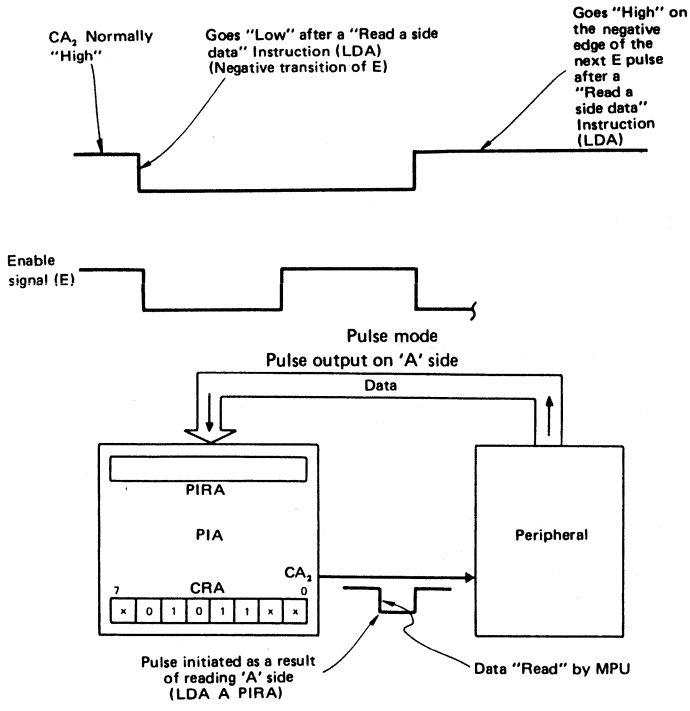


Figure 20 Bits 5, 4, 3 of CRA = 101 (Pulse Mode)

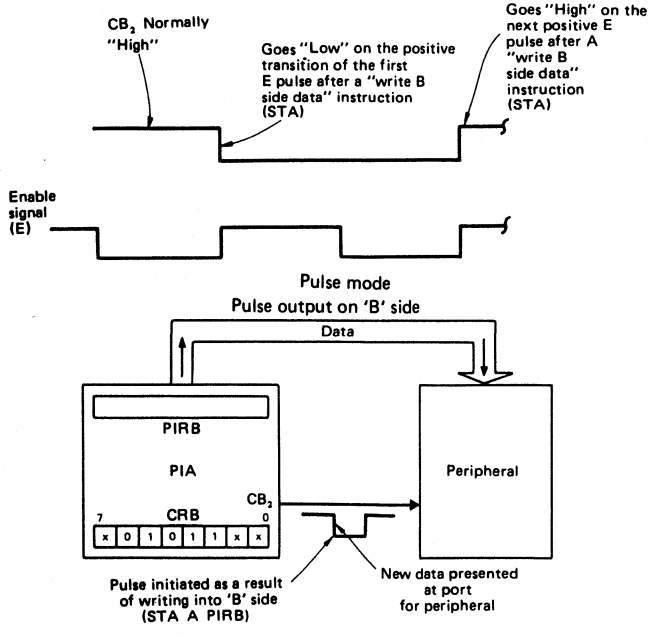


Figure 21 Bits 5, 4, 3 of CRB = 101 (Pulse Mode)

■ SUMMARY OF CONTROL REGISTERS CRA AND CRB

Control registers CRA and CRB have total control of CA_1, CA_2, CB_1, and CB_2 lines. The status of eight bits of the control registers may be read into the MPU. However, the MPU can only write into Bit 0 through Bit 5 (6 bits), since Bit 6 and Bit 7 are set only by CA_1, CA_2, CB_1, or CB_2.

• Addressing PIAs

Before addressing PIAs, the data direction (DDR) must first be loaded with the bit pattern that defines how each line is to function, i.e., as an input or an output. A logic "1" in the data direction register defines the corresponding line as an output while a logic "0" defines the corresponding line as an input. Since the DDR and the peripheral interface resister have the same address, the control register bit 2 determines which register is being addressed. If Bit 2 in the control register is a logic "0", then the DDR is addressed. If Bit 2 in the control register is a logic "1", the peripheral interface register is addressed. Therefore, it is essential that the DDR be loaded first before setting Bit 2 of the control register.

<Example>

Given a PIA with an address of 4004, 4005, 4006, and 4007. 4004 is the address of the A side peripheral interface register. 4005 is the address of the A side control register. 4006 is the address of the B side peripheral interface register. 4007 is the address of the B side control register. On the A side, Bits 0, 1, 2, and 3 will be defined as inputs, while Bits 4, 5, 6, and 7 will be used as outputs. On the B side, all lines will be used as outputs.

```
PIA1AD = 4004        (DDRA, PIRA)
PIA1AC = 4005        (CRA)
PIA1BD = 4006        (DDRB, PIRB)
PIA1BC = 4007        (CRB)

1.  LDA A #%11110000   (4 outputs, 4 inputs)
2.  STA A    PIA1AD    (Loads A DDR)
3.  LDA A #%11111111   (All outputs)
4.  STA A    PIA1BD    (Loads B DDR)
5.  LDA A #%00000100   (Sets Bit 2)
6.  STA A    PIA1AC    (Bit 2 set in A control register)
7.  STA A    PIA1BC    (Bit 2 set in B control register)
```

Statement 2 addresses the DDR, since the control register (Bit 2) has not been loaded. Statements 6 and 7 load the control registers with Bit 2 set, so addressing PIA1AD or PIA1BD accesses the peripheral interface register.

• PIA Programming Via The Index Register

The program shown in the previous section can be accomplished using the Index Register.

```
1.  LDX   #$F004
2.  STX   PIA1AD      $F0→PIA1AD;$04→PIA1AC
3.  LDX   #$FF04
4.  STX   PIA1BD      $FF→PIA1BD;$04→PIA1BC
```

Using the index register in this example has saved six bytes of program memory as compared to the program shown in the previous section.

• Active Low Outputs

When all the outputs of given PIA port are to be active "Low" (True ≦ 0.4 volts), the following procedure should be used.

a) Set Bit 2 in the control register.
b) Store all 1s ($FF) in the peripheral interface register.
c) Clear Bit 2 in the control register.
d) Store all 1s ($FF) in the data direction register.
e) Store control word (Bit 2 = 1) in control register.

<Example>

The B side of PIA1 is set up to have all active low outputs. CB_1 and CB_2 are set up to allow interrupts in the HANDSHAKE MODE and CB_1 will respond to positive edges ("Low"-to-"High" transitions). Assume reset conditions. Addresses are set up and equated to the same labels as previous example.

```
1.  LDA A #4
2.  STA A PIA1BC     Set Bit 2 in PIA1BC (control register)
3.  LDA B #$FF
4.  STA B PIA1BD     All 1s in peripheral interface register
5.  CLR PIA1BC       Clear Bit 2
6.  STA B PIA1BD     All 1s in data direction register
7.  LDA A #$27
8.  STA A PIA1BC     00100111→→ control register
```

The above procedure is required in order to avoid outputs going "Low", to the active "Low" TRUE STATE, when all ls are stored to the data direction register as would be the case if the normal configuration procedure were followed.

• Interchanging RS_0 And RS_1

Some system applications may require movement of 16 bits of data to or from the "outside world" via two PIA ports (A side + B side). When this is the case it is an advantage to interconnect RS_1 and RS_0 as follows.

RS_0 to A1 (Address Line A1)
RS_1 to A0 (Address Line A0)

This will place the peripheral interface registers and control registers side by side in the memory map as follows.

Table	Example Address	
PIA1AD	$4004	(DDRA, PIRA)
PIA1BD	$4005	(DDRB, PIRB)
PIA1AC	$4006	(CRA)
PIA1BC	$4007	(CRB)

The index register or stackpointer may be used to move the 16-bit data in two 8-bit bytes with one instruction. As an example:

```
LDX PIA1AD      PIA1AD →→ IXH: PIA1BD →→ IXL
```

• PIA – After Reset

When the $\overline{RES}$ (Reset Line) has been held "Low" for a minimum of one microsecond, all registers in the PIA will be cleared.

Because of the reset conditions, the PIA has been defined as

follows.

1. All I/O lines to the "outside world" have been defined as inputs.
2. CA_1, CA_2, CB_1, and CB_2 have been defined as interrupt input lines that are negative edge sensitive.
3. All the interrupts on the control lines are masked. Setting of interrupt flag bits will not cause $\overline{IRQA}$ or $\overline{IRQB}$ to go "Low".

■ SUMMARY OF CA_1-CB_1 PROGRAMMING

Bits 1 and 0 of the respective control registers are used to program the interrupt input control lines CA_1 and CB_1.

b1	b0	
0	0	b1 = Edge (0 = –, 1 = +)
0	1	b0 = Mask (0 = Mask, 1 = Allow)
1	0	
1	1	

Note that this is the same logic as Bits 4 and 3 for CA_2-CB_2 when CA_2-CB_2 are programmed as inputs.

■ SUMMARY OF CA_2-CB_2 PROGRAMMING

Bits 5, 4, and 3 of the control registers are used to program the operation of CA_2-CB_2.

	b5	b4	b3		
CA_2–CB_2 Input Mode →	0	0(–)	0	(Mask)	CA_2–CB_2 Input Mode
	0	0(–)	1	(Allow)	b4 = Edge (0 = –, 1 = +)
	0	1(+)	0	(Mask)	b3 = Mask (0 = Mask,
	0	1(+)	1	(Allow)	1 = Allow)
CA_2–CB_2 Output Mode →	1	0	0	0 – Handshake Mode	
	1	0	1	1 – Pulse Mode	
	1	1	0	b3 Following Mode	
	1	1	1	b3 Following Mode	

I/O As Follow:
Control Lines:
CA_1 – Positive Edge, Allow Interrupt
CA_2 – Pulse Mode
CB_1 – Negative Edge, Mask Interrupt
CB_2 – Hand Shake Mode

Assume Reset Condition
PIA1AD
PIA1AC
PIA1BD
PIA1BC

PIA Configuration Solution

LDA A #$BC	10111100
STA A PIA1AD	I/O to DDRA
LDA A #$FF	1111 1111
STA A PIA1BD	I/O to DDRB
LDA A #$2F	0010 1111
STA A PIA1AC	To "A" Control
LDA A #$24	0010 0100
STA A PIA1BC	To "B" Control

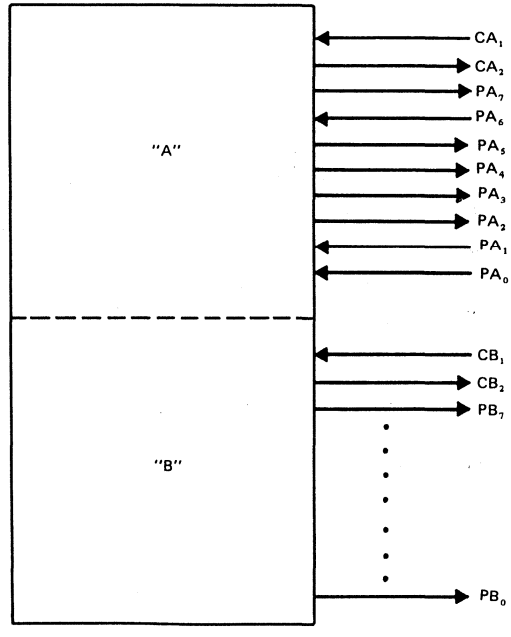
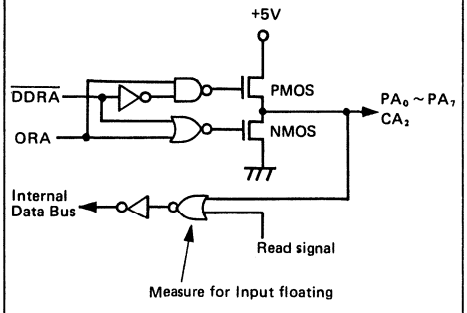
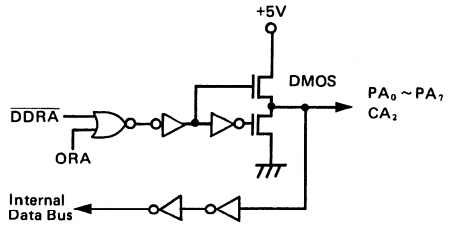
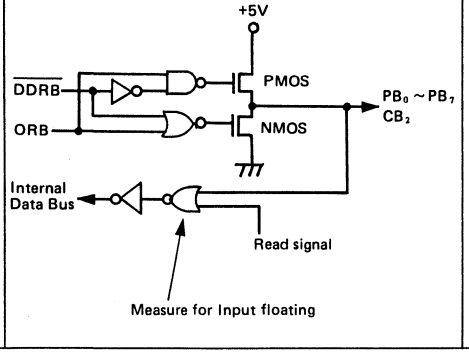
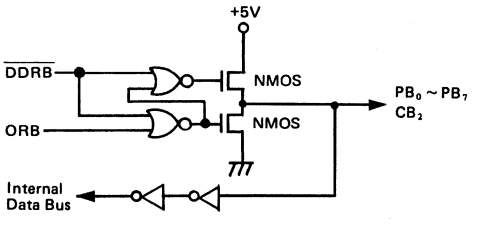


Figure 22 PIA Configuration Problem

■ NOTE FOR USE

Compatibility with NMOS PIA (HD6821)

Table 7 Comparison CMOS PIA (HD6321) with NMOS PIA (HD6821)

Item	CMOS PIA (HD6321)	NMOS PIA (HD6821)
Port A Output Buffer	Three-state output 	Pull-up output 
Port B Output Buffer	Three-state output 	Three-state output 

There is no difference between CMOS PIA and NMOS PIA in pin arrangement.

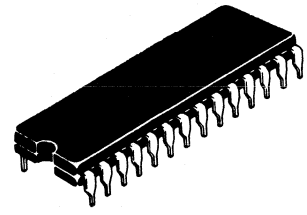
HD6340/HD6840

PTM (Programmable Timer Module)

The HD6340/HD6840 (PTM) is a programmable subsystem component of the HMCS6800 family designed to provide variable system time intervals.

The PTM has three 16-bit binary counters, three corresponding control registers and a status register. These counters are under software control and may be used to cause system interrupts and/or generate output signals. The PTM may be utilized for such tasks as frequency measurements, event counting, interval measuring and similar tasks. The device may be used for square wave generation, gated delay signals, single pulses of controlled duration, and pulse width modulation as well as system interrupts.

HD6340P, HD6840P



(DP-28)

The specifications of the HD6340 are for preliminary and may change hereafter.

Please make an inquire at sales office upon adoption of the HD6340.

■ FEATURES

- Operates from a Single 5 volts Power Supply
- Single System Clock Required (E)
- Selectable Prescaler on Timer 3 Capable of 4 MHz for the HD6340/HD6840, 6 MHz for the HD63A40/HD68A40 and 8 MHz for the HD63B40/HD68B40.
- Programmable Interrupts ($\overline{IRQ}$) Output to MPU
- Readable Down Counter Indicates Counts to Go until Time-out
- Selectable Gating for Frequency or Pulse-Width Comparison
- Three Asynchronous External Clock and Gate/Trigger Input Internally Synchronized
- Three Maskable Outputs

– HD6340 –

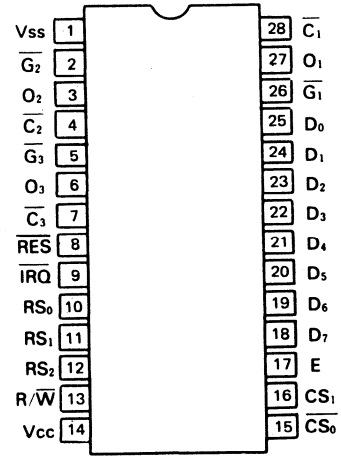
- Wide Range Operating Voltage (V_{CC} = 5V ±10%)
- Low-Power, High-Speed, High-Density CMOS
- Compatible with NMOS PTM (HD6840)

– HD6840 –

- Compatible with MC6840, MC68A40 and MC68B40

■ TYPE OF PRODUCTS

Type	Process	Clock Frequency	Package
HD6340	CMOS	1.0 MHz	DP-28
HD63A40		1.5 MHz	
HD63B40		2.0 MHz	
HD6840	NMOS	1.0 MHz	DP-28
HD68A40		1.5 MHz	
HD68B40		2.0 MHz	



(Top View)

■ ABSOLUTE MAXIMUM RATINGS

Item	Symbol	Value		Unit
		HD6340	HD6840	
Supply Voltage	V_{cc}*	−0.3~+7.0	−0.3~+7.0	V
Input Voltage	V_{in}*	−0.3~+7.0	−0.3~+7.0	V
Maximum Output Current	$\|I_o\|$**	10		mA
Operating Temperature	T_{opr}	−20~+75	−20~+75	°C
Storage Temperature	T_{stg}	−55~+150	−55~+150	°C

* With respect to V_{ss} (SYSTEM GND)

** Maximum output current is the maximum currents which can flow out from one output terminal or I/O common terminal. (D_0 ~ D_7, O_1 ~ O_3, $\overline{IRQ}$)

(NOTE) Permanent LSI damage may occur if maximum ratings are exceeded. Normal operation should be under recommended operating conditions. If these conditions are exceeded, it could affect reliability of LSI.

■ RECOMMENDED OPERATING CONDITIONS

Item		Symbol	HD6340			HD6840			Unit
			min	typ	max	min	typ	max	
Supply Voltage		V_{cc}*	4.5	5.0	5.5	4.75	5.0	5.25	V
Input "Low Voltage		V_{IL}*	0	–	0.8	−0.3	–	0.8	V
Input "High" Voltage	E_1 R/$\overline{W}$	V_{IH}*	2.6**	–	V_{cc}	2.2	–	V_{cc}	V
	Other Inputs		2.2	–	V_{cc}				
Operating Temperature		T_{opr}	−20	25	75	−20	25	75	°C

* With respect to V_{ss} (SYSTEM GND)

** Characteristics to be improved.

■ ELECTRICAL CHARACTERISTICS

• DC CHARACTERISTICS (HD6340; V_{CC} = 5V ±10%, HD6840; V_{CC} = 5V ±5%, V_{SS} = 0V, T_a = −20~+75°C, unless otherwise noted.)

Item	Symbol	HD6340 Test Condition		HD6340 min	HD6340 typ*	HD6340 max	HD6840 Test Condition		HD6840 min	HD6840 typ*	HD6840 max	Unit
Input "High" Voltage	E, R/$\overline{W}$	V_{IH}		2.6**	–	V_{CC}	–		2.2	–	V_{CC}	V
	Other Inputs			2.2	–	V_{CC}						
Input "Low" Voltage	V_{IL}			−0.3	–	0.8			−0.3	–	0.8	V
Input Leakage Current	I_{in}	V_{in} = 0 ~ V_{CC} (Except $D_0 \sim D_7$)		−2.5	–	2.5	V_{in}=0~V_{CC} (Except $D_0 \sim D_7$)		−2.5	–	2.5	μA
Three-State Input Current (Off-state)	I_{TSI}	V_{in} = 0.4 ~ V_{CC}, V_{CC} = 5.5V ($D_0 \sim D_7$)		−10	–	10	V_{in} = 0.4~2.4V V_{CC} = 5.25V ($D_0 \sim D_7$)		−10	–	10	μA
Output "High" Voltage	V_{OH}	I_{LOAD} = −400μA ($D_0 \sim D_7$)		4.1	–	–	I_{LOAD} = −205μA ($D_0 \sim D_7$)		2.4	–	–	V
		$I_{LOAD} \leq$ 10μA ($D_0 \sim D_7$)		V_{CC} −0.1	–	–						
		I_{LOAD} = −400μA (Other Outputs)		4.1	–	–	I_{LOAD} = −200μA ($I_{Other\ Outputs}$)					
		$I_{LOAD} \leq$ 10μA (Other Outputs)		V_{CC}−0.1	–	–						
Output "Low" Voltage	V_{OL}	I_{LOAD} = 1.6mA ($D_0 \sim D_7$)		–	–	0.4	I_{LOAD} = 1.6mA ($D_0 \sim D_7$)		–	–	0.4	V
		I_{LOAD} = 3.2mA ($O_1 \sim O_3$, $\overline{IRQ}$)					I_{LOAD} = 3.2mA, ($O_1 \sim O_3$, $\overline{IRQ}$)					
Output Leakage Current (Off-state)	I_{LOH}	V_{OH} = V_{CC} ($\overline{IRQ}$)		–	–	10	V_{OH} = 2.4V ($\overline{IRQ}$)		–	–	10	μA
Supply Current	I_{CC}	• Chip is not selected. • All counter latches are preset. • $O_1 \sim O_3$ outputs are masked. • Input level (Except E) { V_{IH} min=V_{CC}−0.8V, V_{IL} max = 0.8V	E = 1.0 MHz	–	–	1.0						mA
			E = 1.5 MHz	–	–	1.5						
			E = 2.0 MHz	–	–	2.0						
		• Chip is not selected • Counters are operating. • $O_1 \sim O_3$ operating with load. • Input level (Except E) { V_{IH} min = V_{CC}−0.8, V_{IL} max = 0.8V	E = 1.0 MHz	–	–	3.0						
			E = 1.5 MHz	–	–	4.0						
			E = 2.0 MHz	–	–	6.0						
		• Data bus in R/W operation. • Counters are operating. • $O_1 \sim O_3$operating with load.	E = 1.0 MHz	–	–	5.0						
			E = 1.5 MHz	–	–	8.0						
			E = 2.0 MHz	–	–	10.0						
Power Dissipation	P_D								–	330	550	mW
Input Capacitance	C_{in}	V_{in} = 0V, T_a = 25°C f = 1 MHz	$D_0 \sim D_7$	–	–	12.5	V_{in} = 0V, T_a = 25°C f = 1.0 MHz	$D_0 \sim D_7$	–	–	12.5	pF
			Other Input			7.5		Other Input	–	–	7.5	
Output Capacitance	C_{out}	V_{in} = 0V, T_a = 25°C, f = 1 MHz	$\overline{IRQ}$	–	–	5.0	V_{in} = 0V T_a = 25°C f = 1.0 MHz	$\overline{IRQ}$	–	–	5.0	pF
			O_1, O_2, O_3	–	–	10.0		O_1, O_2, O_3	–	–	10.0	

* T_a = 25°C, V_{CC} = 5.0V ** V_{IH} = 2.2V at V_{CC} = 5V ±5%, T_a = 0~70°C, Characteristics to be improved.

• AC CHARACTERISTICS (HD6340; V_{CC} = 5V ±10%, HD6840; 5V ±5%, V_{SS} = 0V, Ta = −20~+75°C, unless otherwise noted.)

1. MPU READ TIMING

Item	Symbol	Test Condition	HD6340		HD63A40		HD63B40		HD6840		HD68A40		HD68B40		Unit
			min	max	min	max	min	max	min	max	min	max	min	max	
Enable Cycle Time	t_{cycE}	Fig. 1	1000	10000	666	10000	500	10000	1000	10000	666	10000	500	10000	ns
Enable "High" Pulse Width	PW_{EH}		450	9500	280	9500	220	9500	450	4500	280	4500	220	4500	ns
Enable "Low" Pulse Width	PW_{EL}		430	9500	280	9500	210	9500	430	–	280	–	210	–	ns
Enable Rise and Fall Time	t_{Er}, t_{Ef}		–	25	–	25	–	20	–	25	–	25	–	25	ns
Address Set-up Time	t_{AS}		80	–	60	–	40	–	140	–	140	–	70	–	ns
Data Delay Time	t_{DDR}		–	290	–	180	–	150	–	320	–	220	–	180	ns
Data Hold Time	t_{HR}		20	100	20	100	20	100	10	–	10	–	10	–	ns
Address Hold Time	t_{AH}		10	–	10	–	10	–	10	–	10	–	10	–	ns
Data Access Time	t_{ACC}		–	370	–	240	–	190	–	480	–	360	–	250	ns

2. MPU WRITE TIMING

Item	Symbol	Test Condition	HD6340		HD63A40		HD63B40		HD6840		HD68A40		HD68B40		Unit
			min	max	min	max	min	max	min	max	min	max	min	max	
Enable Cycle Time	t_{cycE}	Fig. 2	1000	10000	666	10000	500	10000	1000	10000	666	10000	500	10000	ns
Enable "High" Pulse Width	PW_{EH}		450	9500	280	9500	220	9500	450	4500	280	4500	220	4500	ns
Enable "Low" Pulse Width	PW_{EL}		430	9500	280	9500	210	9500	430	–	280	–	210	–	ns
Enable Rise and Fall Time	t_{Er}, t_{Ef}		–	25	–	25	–	20	–	25	–	25	–	25	ns
Address Set-up Time	t_{AS}		80	–	60	–	40	–	140	–	140	–	140	–	ns
Data Set-up Time	t_{DSW}		165	–	80	–	60	–	195	–	80	–	60	–	ns
Data Hold Time	t_{HW}		10	–	10	–	10	–	10	–	10	–	10	–	ns
Address Hold Time	t_{AH}		10	–	10	–	10	–	10	–	10	–	10	–	ns

3 TIMING OF PTM SIGNAL

Item		Symbol	Test Condition	HD6340		HD63A40		HD63B40		HD6840		HD68A40		HD68B40		Unit
				min	max	min	max	min	max	min	max	min	max	min	max	
Input Rise and Fall Time	$\overline{C}$, $\overline{G}$, $\overline{RES}$	t_r, t_f	Fig. 3, Fig. 4	–	1000*	–	666*	–	500*	–	1000*	–	666*	–	500*	ns
Input "Low" Pulse Width	$\overline{C}$, $\overline{G}$, $\overline{RES}$	PW_L	Fig. 3 Asynchronous Mode	t_{cycE} +tSU +tHD	–	t_{cycE} +tSU +tHD	—	t_{cycE} +tSU +tHD	—	t_{cycE} +tSU +tHD	—	t_{cycE} +tSU +tHD	—	t_{cycE} +tSU +tHD	–	ns
Input "High" Pulse Width	$\overline{C}$, $\overline{G}$	PW_H	Fig. 4 Asynchronous Mode	t_{cycE} +tSU +tHD	–	t_{cycE} +tSU +tHD	–	t_{cycE} +tSU +tHD	–	t_{cycE} +tSU +tHD	–	t_{cycE} +tSU +tHD	–	t_{cycE} +tSU +tHD	–	ns
Input Setup Time	$\overline{C}$, $\overline{G}$, $\overline{RES}$	t_{SU}	Fig. 5 Synchronous Mode	200	–	120	–	75	–	200	–	120	–	75	–	ns
	($\overline{C_3}$ ÷ 8 Pre-scaler Mode)			200	–	170	–	170	–	200	–	170	–	170	–	ns
Input Hold Time	$\overline{C}$, $\overline{G}$, $\overline{RES}$	t_{HD}	Fig. 5 Synchronous Mode	50	–	50	–	50	–	50	–	50	–	50	–	ns
	$\overline{C_3}$ (÷ 8 Pre-scaler Mode)			50	–	50	–	50	–	50	–	50	–	50	–	ns
Input Pulse Width	$\overline{C_3}$ (÷ 8 Pre-scaler Mode)	PW_L PW_H	Asynchronous Mode	120	–	80	–	60	–	125	–	84	–	62.5	–	ns
Output Delay Time	O_1 ~ O_3	t_{co}	Fig. 6	–	200	–	200	–	200	–	700	–	460	–	340	ns
		t_{cm}								–	450	–	450	–	340	ns
		t_{cmos}								–	2000	–	1350	–	1000	ns
Interrupt Release Time		t_{IR}	Fig. 7	–	1200	–	900	–	700	–	1200	–	900	–	700	ns

* t_r, $t_f \leq t_{cycE}$

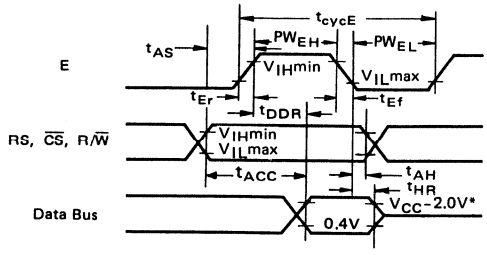


Figure 1 Bus Read Timing
(Read Information from PTM)

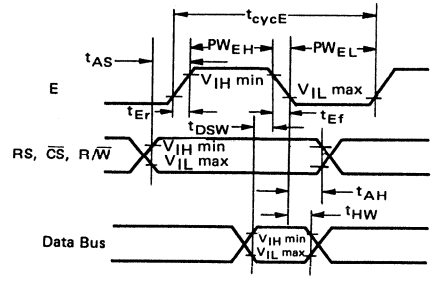


Figure 2 Bus Write Timing
(Write Information into PTM)

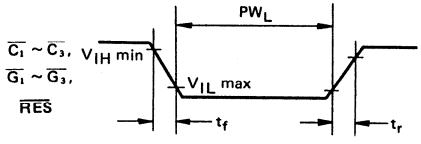


Figure 3 Input Pulse Width "Low"

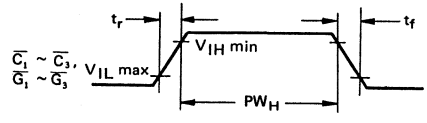


Figure 4 Input Pulse Width "High"

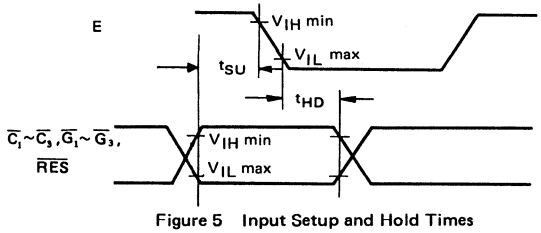


Figure 5 Input Setup and Hold Times

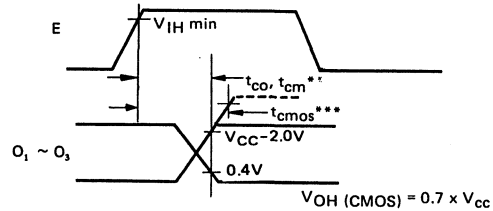


Figure 6 Output Delay

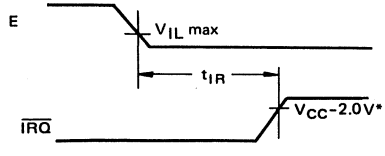


Figure 7 $\overline{IRQ}$ Release Time

* 2.4V for HD6840
, * HD6840 only

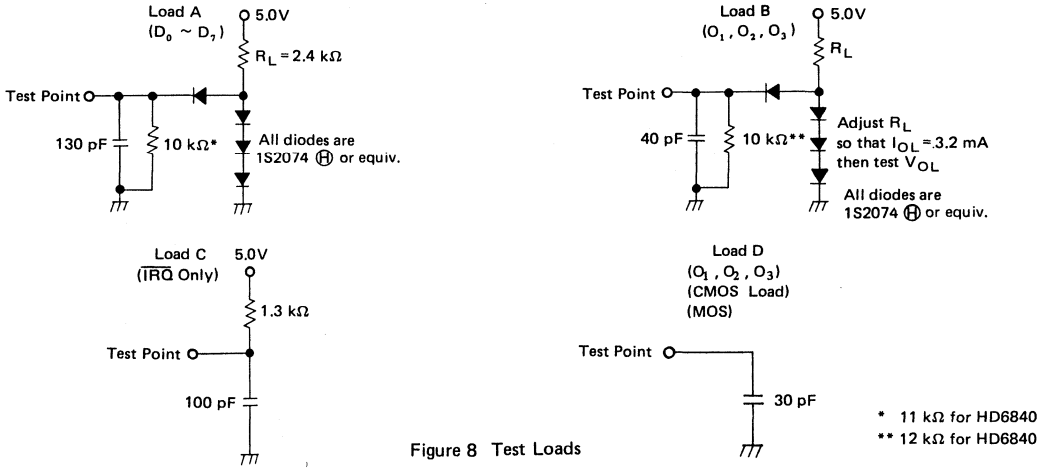


Figure 8 Test Loads

■ GENERAL DESCRIPTION

The PTM is part of the HMCS6800 microprocessor family and is fully bus compatible with HD6800 systems. The three timers in the HD6340/HD6840 operate independently and in several distinct modes to fit a wide variety of measurement and synthesis applications.

The PTM is an integrated set of three distinct counter/timers. It consists of three 16-bit data latches, three 16-bit counters (clocked independently), and the comparison and enable circuitry necessary to implement various measurement and synthesis functions. In addition, it contains interrupt drivers to alert the processor that a particular function has been completed.

In a typical application, a timer will be loaded by first storing two bytes of data into an associated Counter Latch. This data is then transferred into the counter via a Counter initialization cycle. If the counter is enabled, the counter decrements on each subsequent clock period which may be an external clock, or Enable (E) until one of several predetermined conditions causes it to halt or recycle. The timers are thus programmable, cyclic in nature, controllable by external inputs or the MPU program, and accessible by the MPU at any time.

■ PTM INTERFACE SIGNALS FOR MPU

The Programmable Timer Module (PTM) interfaces to the HMCS6800 Bus with an eight-bit bidirectional data bus, two Chip Select lines, a Read/Write line, an Enable (System ϕ_2) line, an Interrupt Request line, an external Reset line, and three Register Select lines. These signals, in conjunction with the HD6800 VMA output, permit the MPU to control the PTM. VMA should be utilized in conjunction with an MPU address line into a Chip Select of the PTM, when the HD6800, HD6802 are used.

• Bidirectional Data ($D_0 \sim D_7$)

Input/Output	Pin No. 25 ~ 18

The bidirectional data lines ($D_0 \sim D_7$) allow the transfer of data between the MPU and PTM. The data bus output drivers are three-state devices which remain in the high-impedance (off) state except when the MPU performs a PTM read operation (Read/Write and Enable lines "High" and PTM Chip Selects activated).

• Chip Select ($\overline{CS_0}$, CS_1)

Input	Pin No. 15, 16

These two signals are used to activate the Data Bus interface and allow transfer of data from the PTM. With $\overline{CS_0}$ = "Low" and CS_1 = "High", the device is selected and data transfer will occur.

• Read/Write (R/$\overline{W}$)

Input	Pin No. 13

This signal is generated by the MPU to control the direction of data transfer on the Data Bus. With the PTM selected, a "Low" state on the PTM R/$\overline{W}$ line enables the input buffers and data is transferred from the MPU to the PTM on the trailing edge of the Enable (System ϕ_2) signal. Alternately, (under the same conditions) R/$\overline{W}$ = "High" and Enable "High" allows data in the PTM to be read by the MPU.

• Enable (E)

Input	Pin No. 17

This signal synchronizes data transfer between the MPU and the PTM. It also performs an equivalent synchronization function on the external clock, reset, and gate inputs of the PTM.

• Interrupt Request ($\overline{IRQ}$)

Output (open drain)	Pin No. 9

The active "Low" Interrupt Request signal is normally tied directly (or through priority interrupt circuitry) to the $\overline{IRQ}$ input of the MPU. This is an "open drain" output (no load device on the chip) which permits other similar interrupt request lines to be tied together in a wire-OR configuration.

The $\overline{IRQ}$ line is activated if, and only if, the Composite Interrupt Flag (Bit 7 of the Internal Status Register) is asserted. The

conditions under which the $\overline{IRQ}$ line is activated are discussed in conjunction with the Status Register.

- **Reset ($\overline{RES}$)**

Input	Pin No. 8

A "Low" level at this input is clocked into the PTM by the Enable (System ϕ_2) input. Two Enable pulses are required to synchronize and process the signal. The PTM then recognizes the active "Low" or inactive "High" on the third Enable pulse. If the $\overline{RES}$ signal is asynchronous, an additional Enable period is required if setup times are not met. The $\overline{RES}$ input must be stable "High"/"Low" for the minimum time stated in the AC Characteristics.

Recognition of a "Low" level at this input by the PTM causes the following action to occur:

a. All counter latches are preset to their maximal count values.
b. All Control Register bits are cleared with the exception of CR10 (internal reset bit) which is set.
c. All counters are preset to the contents of the latches.
d. All counter outputs are reset and all counter clocks are disabled.
e. All Status Register bits (interrupt flags) are cleared.

- **Register Select Lines (RS_0, RS_1, RS_2)**

These inputs are used in conjunction with the R/$\overline{W}$ line to select the internal registers, counters and latches as shown in Table 1.

Input	Pin No. 10, 11, 12

It has been previously stated that the PTM is accessed via MPU Load and Store operations in much the same manner as a memory device. The instructions available with the HMCS6800 family of MPUs which perform operations directly on memory should not be used when the PTM is accessed. These instructions actually fetch a byte from memory, perform an operation, then restore it to the same address location. Since the PTM used the R/$\overline{W}$ line as an additional register select input, the modified data may not be restored to the same register if these instructions are used.

■ PTM ASYNCHRONOUS INPUT/OUTPUT SIGNALS

Each of the three timers within the PTM has external clock and gate inputs as well as a counter output line. The inputs are high impedance, TTL compatible lines and outputs are capable of driving two standard TTL loads.

- **Clock Inputs ($\overline{C_1}$, $\overline{C_2}$, $\overline{C_3}$)**

Input	Pin No. 28, 4, 7

Input pins $\overline{C_1}$, $\overline{C_2}$, and $\overline{C_3}$ will accept asynchronous TTL voltage level signals to decrement Timers 1, 2, and 3, respectively. The "High" and "Low" levels of the external clocks must each be stable for at least one system clock period plus the sum

Table 1 Register Selection

Register Select Inputs *			Operations	
RS_2	RS_1	RS_0	R/$\overline{W}$ = "Low"	R/$\overline{W}$ = "High"
L	L	L	CR20 = "0" Write Control Register #3 CR20 = "1" Write Control Register #1	All bits "0"
L	L	H	Write Control Register #2	Read Status Register
L	H	L	Write MSB Buffer Register	Read Timer #1 Counter
L	H	H	Write Timer #1 Latches	Read LSB Buffer Register
H	L	L	Write MSB Buffer Register	Read Timer #2 Counter
H	L	H	Write Timer #2 Latches	Read LSB Buffer Register
H	H	L	Write MSB Buffer Register	Read Timer #3 Counter
H	H	H	Write Timer #3 Latches	Read LSB Buffer Register

* L; "Low" level, H; "High" level

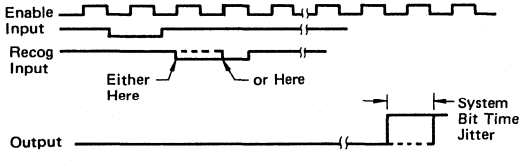
of the setup and hold times for the inputs. The asynchronous clock rate can vary from dc to the limit imposed by Enable (System ϕ_2) Setup, and Hold time.

The external clock inputs are clocked in by Enable (System ϕ_2) pulses. Three Enable periods are used to synchronize and process the external clock. The fourth Enable pulse decrements the internal counter. This does not affect the input frequency, it merely creates a delay between a clock input transition and internal recognition of that transition by the PTM. All references to $\overline{C}$ inputs in this document relate to internal recognition of the input transition. Note that a clock "High" or "Low" level which does not meet setup and hold time specifications may require an additional Enable pulse for recognition. When observing recurring events, a lack of synchronization will result in "jitter" being observed on the output of the PTM when using asynchronous clocks and gate input signals. There are two types of jitter. "System jitter" is the result of the input signals being out of synchronization with the Enable (System ϕ_2), permitting signals with marginal setup and hold time to be recognized by either the bit time nearest the input transition or the subsequent bit time.

"Input jitter" can be as great as the time between input signal negative going transitions plus the system jitter, if the first transition is recognized during one system cycle, and not recognized the next cycle, or vice versa.

External clock input $\overline{C_3}$ represents a special case when Timer #3 is programmed to utilize its optional ÷8 prescaler mode. The maximum input frequency and allowable duty cycles for this case are specified under the AC Characteristics. The output of the ÷8 prescaler is treated in the same manner as the previously discussed clock inputs. That is, it is clocked into the counter by Enable pulses, is recognized on the fourth Enable pulse (provided setup and hold time requirements are met), and must

produce an output pulse at least as wide as the sum of an Enable period, setup, and hold times.



- Gate **Inputs ($\overline{G_1}$, $\overline{G_2}$, $\overline{G_3}$)**

Input	Pin No. 26, 2, 5

Input pins $\overline{G_1}$, $\overline{G_2}$, and $\overline{G_3}$ accept asynchronous TTL-compatible signals which are used as triggers or clock gating functions to Timers 1, 2, and 3, respectively. The gating inputs are clocked into the PTM by the Enable (System ϕ_2) signal in the same manner as the previously discussed clock inputs. That is, a $\overline{Gate}$ transition is recognized by the PTM on the fourth Enable pulse (provided setup and hold time requirements are met), and the "High" or "Low" levels of the $\overline{Gate}$ input must be stable for at least one system clock period plus the sum of setup and hold times. All references to $\overline{G}$ transition in this document relate to internal recognition of the input transition.

The $\overline{Gate}$ inputs of all timers directly affected the internal 16-bit counter. The operation of $\overline{G_3}$ is therefore independent of the ÷8 prescaler selection.

- **Timer Outputs (O_1, O_2, O_3)**

Output	Pin No. 27, 3, 6

Timer outputs O_1, O_2, and O_3 are capable of driving up to two TTL loads and produce a defined output waveform for either Continuous or Single-Shot Timer modes. Output waveform definition is accomplished by selecting either Single 16-bit or Dual 8-bit operating modes. The single 16-bit mode will produce a square-wave output in the continuous timer mode and will produce a single pulse in the Single-Shot Timer mode. The Dual 8-bit mode will produce a variable duty cycle pulse in both the continuous and single shot Timer modes. "1" bit of each Control Register (CRX7) is used to enable the corresponding output. If this bit is cleared, the output will remain "Low" (V_{OL}) regardless of the operating mode.

If it is cleared while the output is high the output will go low during the first enable cycle following a write to the Control Register.

The Continuous and Single-Shot Timer Modes are the only ones for which output response is defined in this data sheet. Signals appear at the outputs (unless CRX7 = "0") during Frequency and Pulse Width comparison modes, but the actual waveform is not predictable in typical applications.

■ CONTROL REGISTER

Each timer in the HD6340 has a corresponding write-only Control Register. Control Register #2 has a unique address space (RS0="High", RS1="Low", RS2="Low") and therefore may be written into at any time. The remaining Control Registers (#1 and#3) share the Address Space selected by a "Low" level on all Register Select inputs.

- **CR20**

The least-significant bit of Control Register #2 (CR20) is used as an additional addressing bit for Control Registers #1 and #3. Thus, with all Register selects and R/$\overline{W}$ inputs at "Low" level. Control Register #1 will be written into if CR20 is a logic "1". Under the same conditions, control Register #3 can also be written into after a $\overline{RES}$ "Low" condition has occurred, since all control register bits (except CR10) are cleared. Therefore, one may write in the sequence CR3, CR2, CR1.

- **CR10**

The least-significant bit of Control Register #1 is used as an internal Reset bit. When this bit is a logic "0", all timers are allowed to operate in the modes prescribed by the remaining bits of the control registers. Writing a "1" into CR10 causes all counters to be preset with the contents of the corresponding counter latches, all counter clocks to be disabled, and the timer outputs and interrupt flags (Status Register) to be reset. Counter Latches and Control Registers are undisturbed by an Internal Reset and may be written into regardless of the state of CR10.

- **CR30**

The least-significant bit of Control Register #3 is used as a selector for a ÷8 prescaler which is available with Timer #3 only. The prescaler, if selected, is effectively placed between the clock input circuitry and the input to Counter #3. It can therefore be used with either the internal clock (Enable) or an external clock source.

- **CRX1 ~ CRX7 (X=1~3)**

The functions depicted in the foregoing discussions are tabulated in Table 2 for ease of reference.

Control Register Bits CR10, CR20, and CR30 are unique in that each selects a different function. The remaining bits (1 through 7) of each Control Register select common functions, with a particular Control Register affecting only its corresponding timer.

- **CRX1**

Bit 1 of Control Register #1 (CR11) selects whether an internal or external clock source is to be used with Timer #1. Similarly, CR21 selects the clock source for Timer #2, and CR31 performs this function for Timer #3. The function of each bit of Control Register "X" can therefore be defined as shown in the remaining section of Table 2.

- **CRX2**

Control Register Bit 2 selects whether the binary information contained in the Counter Latches (and subsequently loaded into the counter) is to be treated as a single 16-bit word or two 8-bit bytes. In the single 16-bit Counter Mode (CRX2=0) the counter will decrement to zero after N + 1 enabled ($\overline{G}$="Low") clock periods, where N is defined as the 16-bit number in the Counter Latches. With CRX2 = 1, a similar Time Out will occur after (L + 1)·(M + 1) enabled clock periods, where L and M, respectively, refer to the LSB and MSB bytes in the Counter Latches.

- **CRX3 ~ CRX7**

Control Register Bits 3, 4, and 5 are explained in detail in the Timer Operating Mode section. Bit 6 is an interrupt mask bit which will be explained more fully in conjunction with the Status Register, and bit 7 is used to enable the corresponding Timer Output. A summary of the control register programming modes is shown in Table 3.

■ STATUS REGISTER/INTERRUPT FLAGS

The PTM has an internal Read-Only Status Register which contains four Interrupt Flags. (The remaining four bits of the register are not used, and default to "0"s when being read.) Bits 0, 1, and 2 are assigned to Timers 1, 2, and 3, respectively, as individual flag bits, while Bit 7 is a Composite Interrupt Flag. This flag bit will be asserted if any of the individual flag bits is

Table 2 Control Register Bits

CONTROL REGISTER #1		CONTROL REGISTER #2		CONTROL REGISTER #3	
CR10	Internal Reset Bit	CR20	Control Register Address Bit	CR30	Timer #3 Clock Control
"0" All timers allowed to operate "1" All timers held in preset state		"0" CR #3 may be written "1" CR #1 may be written		"0" T3 Clock is not prescaled "1" T3 Clock is prescaled by ÷ 8	
CRX1* "0" "1"		Timer #X Clock Source TX uses external clock source on $\overline{CX}$ input TX uses Enable clock			
CRX2 "0" "1"		Timer #X Counting Mode Control TX configured for normal (16-bit) counting mode TX configured for dual 8-bit counting mode			
CRX3 CRX4 CRX5		Timer #X Counter Mode and Interrupt Control (See Table 3)			
CRX6 "0" "1"		Timer #X Interrupt Enable Interrupt Flag masked on $\overline{IRQ}$ Interrupt Flag enabled to $\overline{IRQ}$			
CRX7 "0" "1"		Timer #X Counter Output Enable TX Output masked on output OX TX Output enabled on output OX			

* Control Register for Timer 1, 2, or 3, Bit 1.

set while Bit 6 of the corresponding Control Register is at a logic "1". The conditions for asserting the Composite Interrupt Flag bit can therefore be expressed as:

$$INT = I_1 \cdot CR16 + I_2 \cdot CR26 + I_3 \cdot CR36$$

where INT = Composite Interrupt Flag (Bit 7)
I_1 = Timer #1 Interrupt Flag (Bit 0)
I_2 = Timer #2 Interrupt Flag (Bit 1)
I_3 = Timer #3 Interrupt Flag (Bit 2)

STATUS REGISTER

7	6	5	4	3	2	1	0
INT	/	/	/	/	I_3	I_2	I_1

An interrupt flag is cleared by a Timer Reset condition, i.e., External $\overline{RES}$ = "Low" or Internal Reset Bit (CR10) = "1". It will also be cleared by a Read Timer Counter Command provided that the Status Register has previously been read while the interrupt flag was set. This condition on the Read Status Register – Read Timer Counter (RS–RT) sequence is designed to prevent missing interrupts which might occur after the status register is read, but prior to reading the Timer Counter.

An Individual Interrupt Flag is also cleared by a Write Timer Latches (W) command or a Counter Initialization (CI) sequence, provided that W or CI affects the Timer corresponding to the individual Interrupt Flag.

■ COUNTER LATCH INITIALIZATION

Each of the three independent timers consists of a 16-bit addressable counter and 16 bits of addressable latches. The counters are preset to the binary numbers stored in the latches. Counter initialization results in the transfer of the latch contents to the counter. See notes in Table 5 regarding the binary number N, L, or M placed into the Latches and their relationship to the output waveforms and counter Time-Outs.

Since the PTM data bus is 8-bits wide and the counters are 16-bits wide, a temporary register (MSB Buffer Register) is provided. This "write only" register is for the Most Significant Byte of the desired latch data. Three addresses are provided for the MSB Buffer Register (as indicated in Table 1), but they all lead to the same Buffer. Data from the MSB Buffer will automatically be transferred into the Most Significant Byte of Timer #X when a Write Timer #X Latches Command is performed. So it can be seen that the PTM has been designed to allow transfer of two bytes of data into the counter latches provided that the MSB is transferred first.

In many applications, the source of the data will be as HMCS6800 MPU. It should be noted that the 16-bit store operations of the HMCS6800 microprocessors (STS and STX etc.) transfer data in the order required by the PTM. A Store Index Register Instruction, for example, results in the MSB of the X register being transferred to the selected address, then the LSB of the X register being written into the next higher location. Thus, either the index register or stack pointer may be transfered directly into a selected counter latch with a single instruction.

A logic "Low" at the $\overline{RES}$ input also initializes the counter latches. In this case, all latches will assume a maximum count of $(65,536)_{10}$. It is important to note that an Internal Reset (Bit 0 of Control Register 1 Set) has no effect on the counter latches.

■ COUNTER INITIALIZATION

Counter Initialization is defined as the transfer of data from the latches to the counter with subsequent clearing of the Individual Interrupt Flag associated with the counter. Counter Initialization always occurs when a reset condition ($\overline{RES}$ = "Low" or CR10 = "1") is recognized. It can also occur – depending on Timer Mode – with a Write Timer Latches command or recognition of a negative transition of the $\overline{Gate}$ input.

Counter recycling or re-initialization occurs when a negative transition of the clock input is recognized after the counter has reached an all-zero state. In this case, data is transferred from the Latches to the Counter.

■ TIMER OPERATING MODES

The PTM has been designed to operate effectively in a wide variety of applications. This is accomplished by using three bits of each control register (CRX3, CRX4, and CRX5) to

defined different operating modes of the Timers. These modes are divided into Wave Synthesis and Wave Measurement modes, and outlined in Table 3.

Table 3 Operating Modes

Control Register			Timer Operating Mode	
CRX3	CRX4	CRX5		
0	*	0	Continuous	Wave Synthesis
0	*	1	Single-Shot	
1	0	*	Frequency Comparison	Wave Measurement
1	1	*	Pulse Width Comparison	

* Defines Additional Timer Functions.

One of the WAVE SYNTHESIS modes is the Continuous Operating mode, which is useful for cyclic wave generation. Either symmetrical or variable duty-cycle waves can be generated in this mode. The other wave synthesis mode, the Single-Shot mode, is similar in use to the Continuous operating mode, however, a single pulse is generated, with a programmable preset width.

The WAVE MEASUREMENT modes include the Frequency Comparison and Pulse Width Comparison modes which are used to measure cyclic and singular pulse widths, respectively.

In addition to the four timer modes in Table 3, the remaining control register bit is used to modify counter initialization and enabling or interrupt conditions.

■ WAVE SYNTHESIS MODES

• Continuous Operating Mode (Table 4)

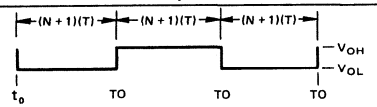
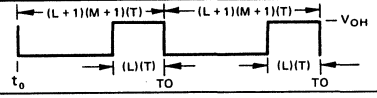
The continuous mode will synthesize a continuous wave with a period proportional to the preset number in the particular timer latches.

Any of the timers in the PTM may be programmed to operate in a continuous mode by writing "0"s into bits 3 and 5 of the corresponding control register. Assuming that the timer output is enabled (CRX7 = "1"), either a square wave or a variable duty cycle waveform will be generated at the Timer Output, OX. The type of output is selected via Control Register Bit 2.

Either a Timer Reset (CR10 = "1" or External $\overline{RES}$ = "Low") condition or internal recognition of a negative transition of the $\overline{Gate}$ input results in Counter Initialization. A Write Timer Latches command can be selected as a Counter Initialization signal by clearing CRX4.

The counter is enabled by an absence of a Timer Reset condition and a "Low" level at the $\overline{Gate}$ input. In the 16-bit mode, the counter will decrement on the first clock cycle during or after the counter initialization cycle. It continues to decrement on each clock signal so long as $\overline{G}$ remains "Low" and no reset condition exists. A Counter Time Out (the first clock after all

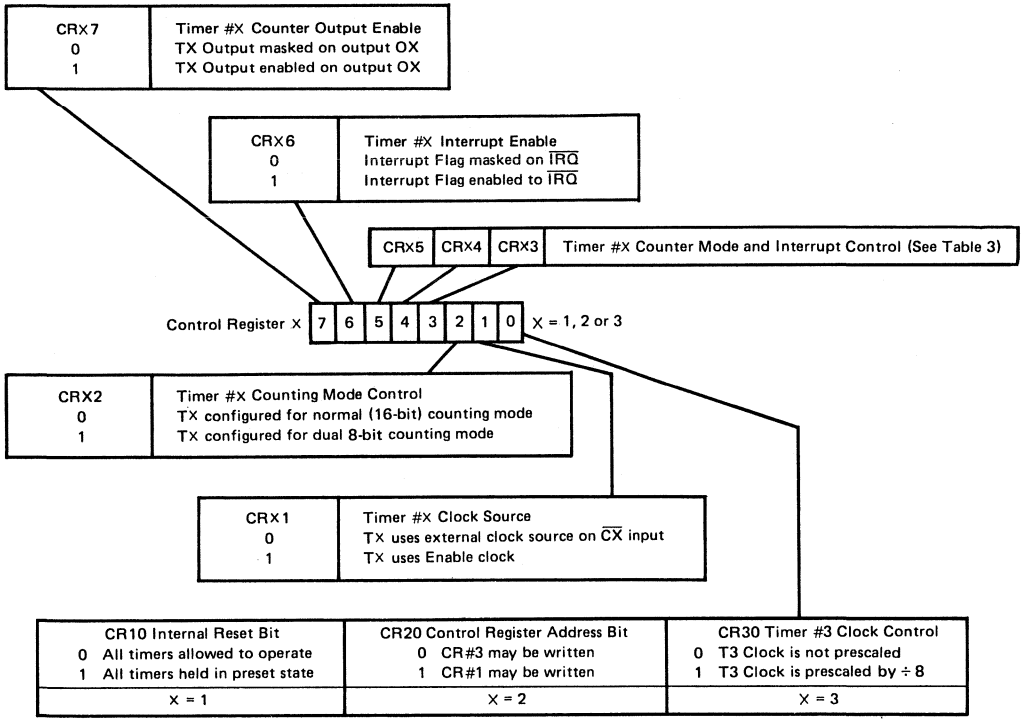
Table 4 Continuous Operating Modes

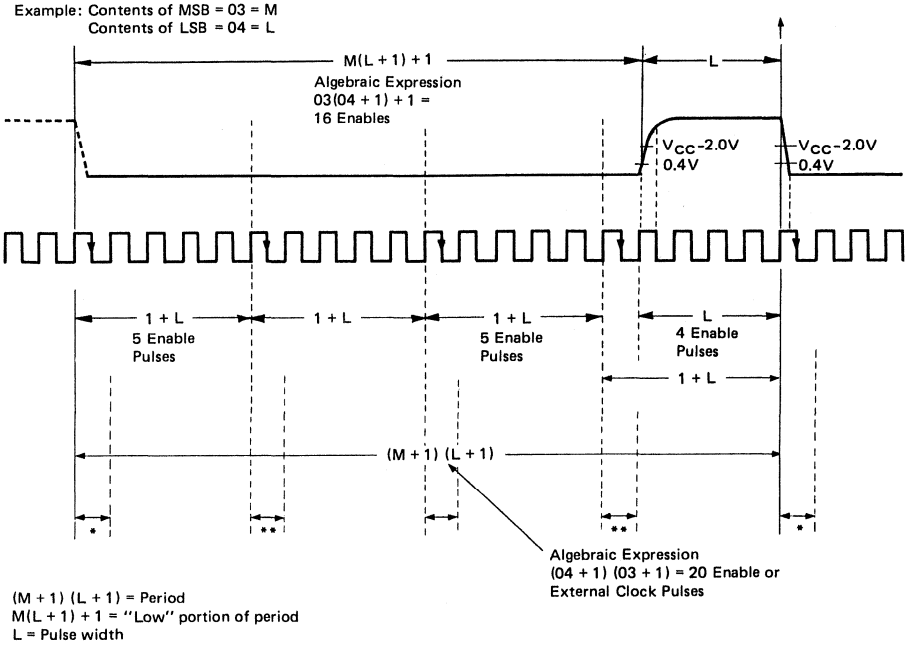
CONTINUOUS MODE (CRX3 = "0", CRX5 = "0")			
Control Register		Initialization/Output Waveforms	
CRX2	CRX4	Counter Initialization	*Timer Output (OX) (CRX7 = "1")
0	0	$\overline{G}$↓+W+R	
0	1	$\overline{G}$↓+R	
1	0	$\overline{G}$↓+W+R	
1	1	$\overline{G}$↓+R	

$\overline{G}$↓ = Negative transition of $\overline{Gate}$ input.
W = Write Timer Latches Command.
R = Timer Reset (CR10 = "1" or External $\overline{RES}$ = "Low")
N = 16-Bit Number in Counter Latch.
L = 8-Bit Number in LSB Counter Latch.
M = 8-Bit Number in MSB Counter Latch.
T = Clock Input Negative Transitions to Counter.
t_0 = Counter Initialization Cycle.
TO = Counter Time Out (All Zero Condition).

* All time intervals shown above assume the $\overline{Gate}$ ($\overline{G}$) and $\overline{Clock}$ ($\overline{C}$) signals are synchronized to Enable (System ϕ_2) with the specified setup and hold time requirements.

Control Register Bits





* Preset LSB and MSB to Respective Latches on the negative transition of the E.
** Preset LSB to LSB Latches and Decrement MSB by one on the negative transition of the E.

Figure 9 Timer Output Waveform Example
(Continuous Dual 8-Bit Mode using Internal Enable)

counter bits = "0") results in the Individual Interrupt Flag being set and re-initialization of the counter.

In the dual 8-bit mode (CRX2= "1") [Refer to the example in Fig. 9] the MSB decrements once for every full countdown of the LSB + 1. When the LSB = "0", the MSB is unchanged; on the next clock pulse the LSB is reset to the count in the LSB Latches and the MSB is decremented by 1 (one). The output, if enabled, remains "Low" during and after initialization and will remain "Low" until the counter MSB is all "0"s. The output will go "High" at the beginning of the next clock pulse. The output remains "High" until both the LSB and MSB of the counter are all "0"s. At the beginning of the next clock pulse the defined Time Out (TO) will occur and the output will go "Low". In the Dual 8-bit mode the period of the output of the example in Fig. 9 would span 20 clock pulses as opposed to the 1546 clock pulses using the Normal 16-bit mode.

A special time-out condition exists for the dual 8-bit mode (CRX2 = "1") if L = "0". In this case, the counter will revert to a mode similar to the single 16-bit mode, except Time Out occurs after M+1 clock pulses. The output, if enabled, goes "Low" during the Counter Initialization cycle and reverses state at each Time Out. The counter remains cyclical (is re-initialized at each Time Out) and the Individual Interrupt Flag is set when Time Out occurs. If M = L = "0", the internal counters do not change, but the output toggles at a rate of 1/2 the clock frequency.

The discussion of the Continuous Mode has assumed that the application requires an output signal. It should be noted that the Timer operates in the same manner with the output disabled (CRX7 = "0"). A Read Timer Counter command is valid regardless of the state of CRX7.

- **Single-Shot Timer Mode**

This mode is identical to the Continuous Mode with three exceptions. The first of these is obvious from the name – the output returns to a "Low" level after the initial Time Out and remains "Low" until another Counter Initialization cycle occurs. The waveforms available are shown in Table 5.

As indicated in Table 5, the internal counting mechanism remains cyclical in the Single-Shot Mode. Each Time Out of the counter results in the setting of an Individual Interrupt Flag and re-initialization of the counter.

The second major difference between the Single-Shot and Continuous modes is that the internal counter enable is not dependent on the $\overline{\text{Gate}}$ input level remaining in the "Low" state for the Single-Shot mode.

Another special condition is introduced in the Single-Shot mode. If L = M = "0"(Dual 8-bit) or N = "0" (Single 16-bit), the output goes "Low" on the first clock received during or after Counter Initialization. The output remains "Low" until the Operating Mode is changed or nonzero data is written into the Counter Latches. Time Outs continue to occur at the end of each clock period.

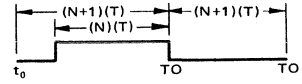
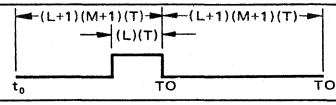
The three differences between Single-Shot and Continuous Timer Modes can be summarized as attributes of the Single-Shot

mode:

1. Output is enabled for only one pulse until it is reinitialized.
2. Counter Enable is independent of $\overline{Gate}$.
3. L = M = "0" or N = "0" disables output.

Aside from these differences, the two modes are identical.

Table 5 Single-Shot Operating Modes

Single-Shot Mode (CRX3 = "0", CRX7 = "1", CRX5 = "1")			
Control Register		Initialization/Output Waveforms	
CRX2	CRX4	Counter Initialization	Timer Output (OX)
0	0	$\overline{G}\downarrow$+W+R	
0	1	$\overline{G}\downarrow$+R	
1	0	$\overline{G}\downarrow$+W+R	
1	1	$\overline{G}\downarrow$+R	

Symbols are as defined in Table 5.

■ WAVE MEASUREMENT MODES

The Wave Measurement Modes are the Frequency (period) Measurement and Pulse Width Comparison Modes, and are provided for those applications which require more flexibility of interrupt generation and Counter Initialization. Individual Interrupt Flags are set in these modes as a function of both Counter Time Out and transitions of the $\overline{Gate}$ input. Counter Initialization is also affected by Interrupt Flag status.

A timer's output is normally not used in a Wave Measurement mode, but it is defined. If the output is enabled, it will operate as follows. During the period between reinitialization of the timer and the first Time Out, the output will be a logical zero. If the first Time Out is completed (regardless of its method of generation), the output will go "High". If further TO's occur, the output will change state at each completion of a Time-Out.

The counter does operate in either Single 16-bit or Dual 8-bit modes as programmed by CRX2. Other features of the Wave Measurement Modes are outlined in Table 6.

Table 6 Wave Measurement Modes

CRX3 = "1"			
CRX4	CRX5	Application	Condition for Setting Individual Interrupt Flag
0	0	Frequency Comparison	Interrupt Generated if $\overline{Gate}$ Input Period (1/F) is less than Counter Time Out (TO)
0	1	Frequency Comparison	Interrupt Generated if $\overline{Gate}$ Input Period (1/F) is greater than Counter Time Out (TO)
1	0	Pulse Width Comparison	Interrupt Generated if $\overline{Gate}$ Input "Down Time" is less than Counter Time Out (TO)
1	1	Pulse Width Comparison	Interrupt Generated if $\overline{Gate}$ Input "Down Time" is greater than Counter Time Out (TO)

• Frequency Comparison or Period Measurement Mode (CRX3 = "1", CRX4 = "0")

The Frequency Comparison Mode with CRX5 = "1" is straightforward. If Time Out occurs prior to the first negative transition of the $\overline{Gate}$ input after a Counter Initialization cycle, an Individual Interrupt Flag is set. The counter is disabled, and a Counter Initialization cycle cannot begin until the interrupt flag is cleared and a negative transition on $\overline{G}$ is detected.

If CRX5 = "0", as shown in Table 6 and Table 7, an interrupt is generated if $\overline{Gate}$ input returns "Low" prior to a Time Out. If Counter Time-Out occurs first, the counter is recycled and continues to decrement. A bit is set within the timer on the initial Time Out which precludes further individual interrupt generation until a new Counter Initialization cycle has been completed. When this internal bit is set, a negative transition of the $\overline{Gate}$ input starts a new Counter Initialization cycle. (The condition of $\overline{G}\downarrow \cdot \overline{I} \cdot TO$ is satisfied, since a Time Out has occurred and no individual Interrupt has been generated.)

Any of the timers within the PTM may be programmed to compare the period of a pulse (giving the frequency after calculations) at the Gate input with the time period requested for Counter Time-Out. A negative transition of the $\overline{Gate}$ input enables the counter and starts a Counter Initialization cycle – provided that other conditions as noted in Table 7 are satisfied. The counter decrements on each clock signal recognized during or after Counter Initialization until an Interrupt is generated, a Write Timer Latches command is issued, or a Timer Reset condition occurs. It can be seen from Table 7 that an interrupt condition will be generated if CRX5 = "0" and the period of the pulse (single pulse or measured separately repetitive pulses) at the $\overline{Gate}$ input is less than the Counter Time Out period. If CRX5 = "1", an interrupt is generated if the reverse is true.

Assume now with CRX5 = "1" that a Counter Initialization has occurred and that the $\overline{\text{Gate}}$ input has returned "Low" prior to Counter Time Out. Since there is no Individual Interrupt Flag generated, this automatically starts a new Counter Initialization Cycle. The process will continue with frequency comparison being performed on each $\overline{\text{Gate}}$ input cycle until the mode is changed, or a cycle is determined to be above the predetermined limit.

- **Pulse Width Comparison Mode (CRX3 = "1", CRX4 = "1")**

This mode is similar to the Frequency Comparison Mode except for a positive, rather than negative, transition of the $\overline{\text{Gate}}$ input terminates the count. With CRX5 = "0", an Individual Interrupt Flag will be generated if the "Low" level pulse applied to the $\overline{\text{Gate}}$ input is less than the time period required for Counter Time Out. With CRX5 = "1", the interrupt is generated when the reverse condition is true.

As can be seen in Table 8, a positive transition of the $\overline{\text{Gate}}$ input disables the counter. With CRX5 = "0", it is therefore possible to directly obtain the width of any pulse causing an interrupt. Similar data for other Time Interval Modes and conditions can be obtained, if two sections of the PTM are dedicated to the purpose.

Table 7 Frequency Comparison Mode

CRX3 = "1", CRX4 = "0"				
Control Reg Bit 5 (CRX5)	**Counter Initialization**	**Counter Enable Flip-Flop Set (CE)**	**Counter Enable Flip-Flop Reset (CE)**	**Interrupt Flag Set (I)**
0	$\overline{G}\downarrow \cdot \overline{I} \cdot (\overline{CE}+TO)+R$	$\overline{G}\downarrow \cdot \overline{W} \cdot \overline{R} \cdot \overline{I}$	W+R+I	$\overline{G}\downarrow$ Before TO
1	$\overline{G}\downarrow \cdot \overline{I}+R$	$\overline{G}\downarrow \cdot \overline{W} \cdot \overline{R} \cdot \overline{I}$	W+R+I	TO Before $\overline{G}\downarrow$

I represents the interrupt for a given timer.

Table 8 Pulse Width Comparison Mode

CRX3 = "1", CRX4 = "1"				
Control Reg Bit 5 (CRX5)	**Counter Initialization**	**Counter Enable Flip-Flop Set (CE)**	**Counter Enable Flip-Flop Reset (CE)**	**Interrupt Flag Set (I)**
0	$\overline{G}\downarrow \cdot \overline{I}+R$	$\overline{G}\downarrow \cdot \overline{W} \cdot \overline{R} \cdot \overline{I}$	W+R+I+G	$\overline{G}\uparrow$ Before TO
1	$\overline{G}\downarrow \cdot \overline{I}+R$	$\overline{G}\downarrow \cdot \overline{W} \cdot \overline{R} \cdot \overline{I}$	W+R+I+G	TO Before $\overline{G}\uparrow$

G = Level sensitive recognition of $\overline{\text{Gate}}$ input.

Table 9 Control Register Programming

7	6	5	4	3	2	1	0
X	X	X	X	X	X	X	↕

	Register 1	Register 2	Register 3
"0"	All Timers Operate	Reg #3 May Be Written	T3 Clk ÷ 1
"1"	All Timers Preset	Reg #1 May Be Written	T3 Clk ÷ 8

7	6	5	4	3	2	1	0
X	X	X	X	X	X	↕	X

"0" External Clock ($\overline{CX}$ Input)
"1" Internal Clock (Enable)

7	6	5	4	3	2	1	0
X	X	X	X	X	↕	X	X

"0" Normal (16-Bit) Count Mode
"1" Dual 8-Bit Count Mode

7	6	5	4	3	2	1	0
X	X	0	0	0	X	X	X

Continuous Operating Mode: $\overline{Gate}$ ↓ or Write to Latches or Reset Causes Counter Initialization

7	6	5	4	3	2	1	0
X	X	0	0	1	X	X	X

Frequency Comparison Mode: Interrupt if $\overline{Gate}$  is < Counter Time Out

7	6	5	4	3	2	1	0
X	X	0	1	0	X	X	X

Continuous Operating Mode: $\overline{Gate}$ ↓ or Reset Causes Counter Initialization

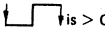
7	6	5	4	3	2	1	0
X	X	0	1	1	X	X	X

Pulse Width Comparison Mode: Interrupt if $\overline{Gate}$  is < Counter Time Out

7	6	5	4	3	2	1	0
1	X	1	0	0	X	X	X

Single Shot Mode: $\overline{Gate}$ ↓ or Write to Latches or Reset Causes Counter Initialization

7	6	5	4	3	2	1	0
X	X	1	0	1	X	X	X

Frequency Comparison Mode: Interrupt If $\overline{Gate}$  is > Counter Time Out

7	6	5	4	3	2	1	0
1	X	1	1	0	X	X	X

Single Shot Mode: $\overline{Gate}$ ↓ or Reset Causes Counter Initialization

7	6	5	4	3	2	1	0
X	X	1	1	1	X	X	X

Pulse Width Comparison Mode: Interrupt If $\overline{Gate}$ is > Counter Time Out

7	6	5	4	3	2	1	0
X	↕	X	X	X	X	X	X

"0" Interrupt Flag Masked ($\overline{IRQ}$)
"1" Interrupt Flag Enabled ($\overline{IRQ}$)

7	6	5	4	3	2	1	0
↕	X	X	X	X	X	X	X

"0" Timer Output Masked
"1" Timer Output Enable

(NOTE) Reset is Hardware or Software Reset ($\overline{RES}$ = "Low" or CR10 = "1").

■ NOTE FOR USE (HD6340 only)

Input signal, which is not necessary for user's application, should be used fixed to "High" or "Low" level. This is applicable to the following signal pins.

$\overline{C_1}, \overline{C_2}, \overline{C_3}, \overline{G_1}, \overline{G_2}, \overline{G_3}$

■ RESTRICTION FOR USE

• Notes for the $O_1 - O_3$ Outputs Noise

(1) Phenomenon

When the excessive load capacitance is connected to data bus and GND wiring impedance is not neglectable in the system using HD6340, the noise appears in $O_1 - O_3$ outpus in the read cycle as indicated in Fig. 10 which may cause the erroneous operation of the system.

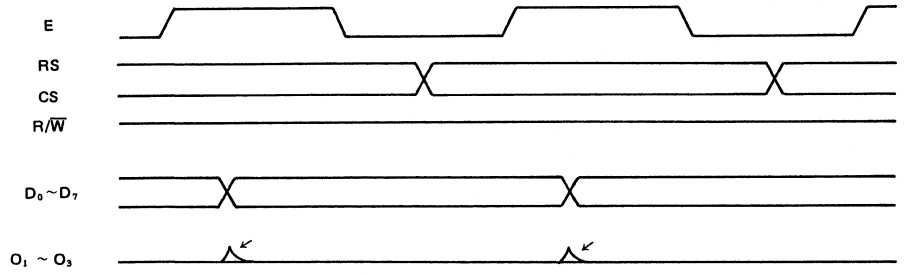


Figure 10 The $O_1 - O_3$ Outputs Noise in the MPU Read Cycle

(2) Cause

When the data buffer turns from "H" to "L", the excessive transient current runs to the GND (the discharge current of the data bus load capacity). Therefore, the noise occurs in the GND pin of the LSI because of the impedance of the GND wiring (resistance and inductance). See Fig. 11 for the details.

Fig. 12 indicates the dependence of the noise voltage upon each parameter.

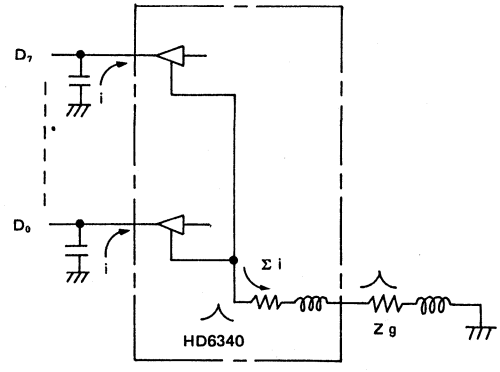
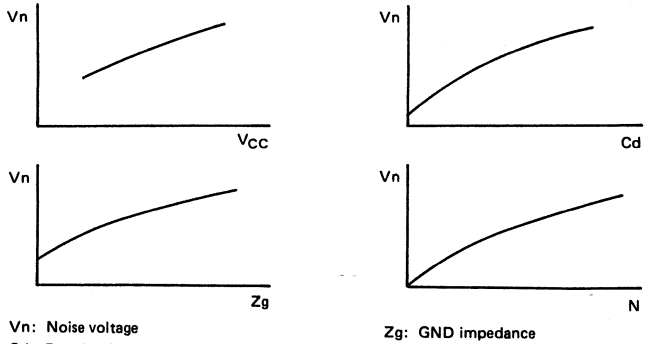


Figure 11 Cause of the Noise



Vn: Noise voltage
Cd: Data bus load capacitance
Zg: GND impedance
N : Number of data bus which changes according to H → L.

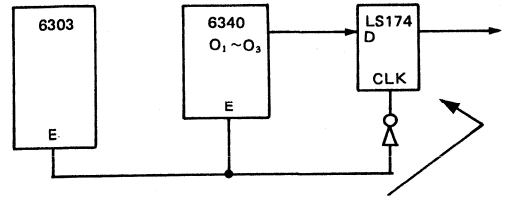
Figure 12 The Dependence of the Noise Voltage upon Each Parameter

However, it is important to consider the fact that the noise voltage varies according to the type of parameter as indicated in Fig. 12.

(3) Countermeasures

When the noise cause the erroneous operation of the system, the countermeasures to be taken are as follows.

(a) Latch the $O_1 - O_3$ outputs by the falling edge of the signal "E".



The latch added to prevent the noise (74LS174).

■ **Precautions when using Timer 3 (HD6340 only)**

When using the HD63B40P Timer 3 under the conditions

1) external clock mode (CR31 = 0)
2) ÷ 8 prescaler unused (CR30 = 0)

and changing the bits of the control register #3 except for the CR30 bit (e.g. in a case where the interrupt mask bit and O_3 output enable bit are changed and the CR30 is not), there is the possibility that one decrement clock may be omitted.

This phenomenon occurs when t_{DSW} (data setup time; standard spec. 60ns minimum) is less than 80ns, and does not occur when t_{DSW} is greater than 80ns.

Therefore, please avoid to use the HD63B40P in the above status when t_{DSW} is less than 80ns.

(This phenomenon doesn't occur in the HD6340P and HD63A40P.)

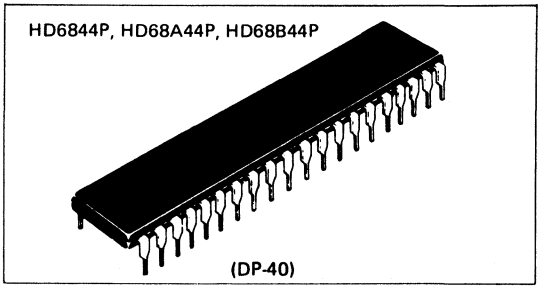
HD6844, HD68A44, HD68B44
DMAC (Direct Memory Access Controller)

The HD6844 Direct Memory Access Controller (DMAC) performs the function of transferring data directly between memory and peripheral device controllers. It controls the address and data buses in place of the MPU in bus organized systems such as the HMCS6800 Microprocessor System.

The bus interface of the HD6844 includes select, read/write, interrupt, transfer request/grant, and bus interface logic to allow the data transfer over an 8-bit bidirectional data bus. The functional configuration of the DMAC is programmed via the data bus. The internal structure provides for control and handling of four individual channels, each of which is separately configured. Programmable control registers provide control for the transfer location and length, individual channel control and transfer mode configuration, priority of servicing, data chaining, and interrupt control. Status and control lines provide control to the peripheral controllers.

The mode of transfer for each channel can be programmed as cycle-stealing or a burst transfer mode.

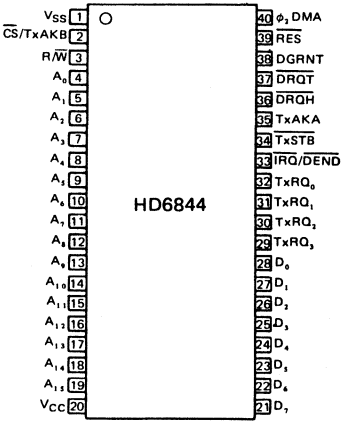
Typical applications would be with the Floppy Disk Controller (FDC), etc..



■ FEATURES

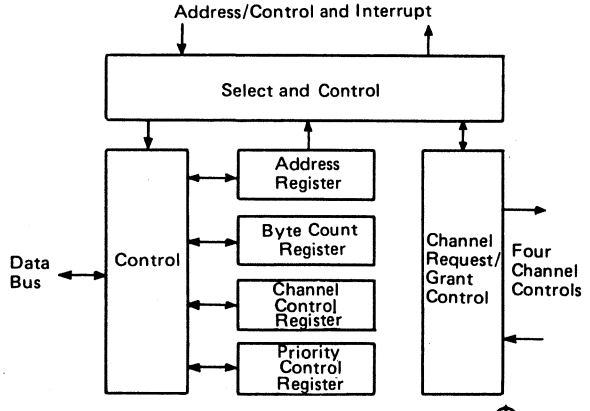
- Four DMA Channels, Each Having a 16-Bit Address Register and a 16-Bit Byte Count Register
- 1 M Byte/Sec (HD6844), 1.5 M Byte/Sec (HD68A44), 2.0 M Byte/Sec (HD68B44) Maximum Data Transfer Rate
- Selection of Fixed or Rotating Priority Service Control
- Separate Control Bits for Each Channel
- Data Chain Function
- Address Increment or Decrement Update
- Programmable Interrupts and DMA End to Peripheral Controllers
- Compatible with MC6844, MC68A44, MC68B44

■ PIN ARRANGEMENT



(Top View)

■ BLOCK DIAGRAM



HITACHI

■ ABSOLUTE MAXIMUM RATINGS

Item	Symbol	Value	Unit
Supply Voltage	V_{CC}*	-0.3 ~ +7.0	V
Input Voltage	V_{in}*	-0.3 ~ +7.0	V
Operating Temperature	T_{opr}	-20 ~ +75	°C
Storage Temperature	T_{stg}	-55 ~ +150	°C

* With respect to V_{SS} (SYSTEM GND)

(NOTE) Permanent LSI damage may occur if maximum ratings are exceeded. Normal operation should be under recommended operating conditions. If these conditions are exceeded, it could affect reliability of LSI.

■ RECOMMENDED OPERATING CONDITIONS

Item	Symbol	min	typ	max	Unit
Power Supply Voltage	V_{CC}*	4.75	5.0	5.25	V
Input Voltage	V_{IL}*	-0.3	–	0.8	V
	V_{IH}*	2.0	–	V_{CC}	V
Operating Temperature	T_{opr}	-20	25	75	°C

* With respect to V_{SS} (SYSTEM GND)

■ ELECTRICAL CHARACTERISTICS (V_{CC}=5V±5%, V_{SS}=0V, Ta=-20~+75°C, unless otherwise noted.)

• DC CHARACTERISTICS

Item		Symbol	Test Condition	min	typ*	max	Unit
Input "High" Voltage		V_{IH}		2.0	–	V_{CC}	V
Input "Low" Voltage		V_{IL}		-0.3	–	0.8	V
Input Leakage Current	TxRQ$_0$~$_3$, ϕ_2 DMA, $\overline{RES}$, DGRNT	I_{in}	V_{in}=0~5.25V	-2.5	–	2.5	μA
Three-State (off state) Leakage Current	A_0~A_{15}, D_0~D_7, R/$\overline{W}$	I_{TSI}	V_{in}=0.4~2.4V	-10	–	10	μA
Output "High" Voltage	D_0~D_7	V_{OH}	I_{OH}=-205μA	2.4	–	–	V
	A_0~A_{15}, R/$\overline{W}$		I_{OH}=-145μA	2.4	–	–	
	All Other Outputs		I_{OH}=-100μA	2.4	–	–	
Output "Low" Voltage		V_{OL}	I_{OL}=1.6mA	–	–	0.4	V
Source Current	$\overline{CS}$/TxAKB	I_{CSS}	V_{in}=0V, Fig. 10	–	10	16	mA
Power Dissipation		P_D		–	500	1000	mW
Input Capacitance	ϕ_2 DMA	C_{in}	V_{in}=0V, T_a=25°C f=1.0MHz	–	–	20	pF
	D_0~D_7, $\overline{CS}$, A_0~A_4, R/$\overline{W}$			–	–	12.5	
	TxRQ$_0$~$_3$, $\overline{RES}$, DGRNT			–	–	10	
Output Capacitance		C_{out}	V_{in}=0V, T_a=25°C, f=1MHz	–	–	12	pF

* V_{CC}=5.0V, T_a=25°C

• AC CHARACTERISTICS (Load Condition Fig. 9)

1. CLOCK TIMING

Item		Symbol	Test Condition	HD6844			HD68A44			HD68B44			Unit
				min	typ	max	min	typ	max	min	typ	max	
ϕ_2 DMA Cycle Time		$t_{cyc\phi}$	Fig. 2	1000	–	–	666	–	–	500	–	–	ns
ϕ_2 DMA Pulse Width	"High" Level	$PW_{\phi H}$	Fig. 2	450	–	–	280	–	–	235	–	–	ns
	"Low" Level	$PW_{\phi L}$	Fig. 2	400	–	–	230	–	–	210	–	–	ns
ϕ_2 DMA Rise and Fall Time		$t_{\phi r}$, $t_{\phi f}$	Fig. 2	–	–	25	–	–	25	–	–	25	ns

2. DMA TIMING (Load Condition Fig. 9)

Item		Symbol	Test Condition	HD6844			HD68A44			HD68B44			Unit
				min	typ	max	min	typ	max	min	typ	max	
TxRQ Setup Time	ϕ_2 DMA Rising Edge	t_{TQS1}	Fig. 3	120	–	–	120	–	–	120	–	–	ns
	ϕ_2 DMA Falling Edge	t_{TQS2}		210	–	–	210	–	–	155	–	–	
TxRQ Hold Time	ϕ_2 DMA Rising Edge	t_{TQH1}		20	–	–	10	–	–	10	–	–	ns
	ϕ_2 DMA Falling Edge	t_{TQH2}		20	–	–	10	–	–	10	–	–	
DGRNT Setup Time	DGRNT	t_{DGS}	Fig. 4	155	–	–	125	–	–	115	–	–	ns
DGRNT Hold Time	DGRNT	t_{DGH}		10	–	–	10	–	–	10	–	–	
Address Output Delay Time	A_0~A_{15}, R/$\overline{W}$, $\overline{TxSTB}$	t_{AD}	Fig. 6	–	–	270	–	–	180	–	–	160	ns
Address Output Hold Time	A_0~A_{15}, R/$\overline{W}$	t_{AHO}	Fig. 6	30	–	–	20	–	–	20	–	–	ns
	$\overline{TxSTB}$		Fig. 7	35	–	–	35	–	–	35	–	–	
Address Three-State Delay Time	A_0~A_{15}, R/$\overline{W}$	t_{ATSD}	Fig. 7	–	–	270	–	–	270	–	–	270	ns
Address Three-State Recovery Time	A_0~A_{15}, R/$\overline{W}$	t_{ATSR}	Fig. 7	–	–	270	–	–	270	–	–	270	ns
Delay Time	$\overline{DRQH}$, $\overline{DRQT}$	t_{DQD}	Fig. 5	–	–	375	–	–	250	–	–	210	ns
TxAK Delay Time	ϕ_2 DMA Rising Edge	t_{TKD1}	Fig. 5	–	–	400	–	–	310	–	–	250	ns
	DGRNT Rising Edge	t_{TKD2}	Fig. 8	–	–	190	–	–	160	–	–	150	
$\overline{IRQ}$/$\overline{DEND}$ Delay Time	ϕ_2 DMA Falling Edge	t_{DED1}	Fig. 6	–	–	300	–	–	250	–	–	210	ns
	DGRNT Rising Edge	t_{DED2}	Fig. 8	–	–	190	–	–	160	–	–	125	

3. BUS TIMING

1) READ TIMING

Item		Symbol	Test Condition	HD6844			HD68A44			HD68B44			Unit
				min	typ	max	min	typ	max	min	typ	max	
Address Setup Time	A_0~A_4, R/$\overline{W}$, $\overline{CS}$	t_{AS}	Fig. 2	140	–	–	140	–	–	70	–	–	ns
Address Input Hold Time	A_0~A_4, R/$\overline{W}$, $\overline{CS}$	t_{AHI}		10	–	–	10	–	–	10	–	–	ns
Data Delay Time	D_7~D_7	t_{DDR}		–	–	320	–	–	220	–	–	180	ns
Data Access Time	D_0~D_7	t_{ACC}		–	–	460	–	–	360	–	–	280	ns
Data Output Hold Time	D_0~D_7	t_{DHR}		10	–	–	10	–	–	10	–	–	ns

2) WRITE TIMING

Item		Symbol	Test Condition	HD6844			HD68A44			HD68B44			Unit
				min	typ	max	min	typ	max	min	typ	max	
Address Setup Time	$A_0 \sim A_4$, R/$\overline{W}$, $\overline{CS}$	t_{AS}	Fig. 2	140	–	–	140	–	–	70	–	–	ns
Address Input Hold Time	$A_0 \sim A_4$, R/$\overline{W}$, $\overline{CS}$	t_{AHI}		10	–	–	10	–	–	10	–	–	ns
Data Setup Time	$D_0 \sim D_7$	t_{DSW}		195	–	–	80	–	–	60	–	–	ns
Data Input Hold Time	$D_0 \sim D_7$	t_{DHW}		10	–	–	10	–	–	10	–	–	ns

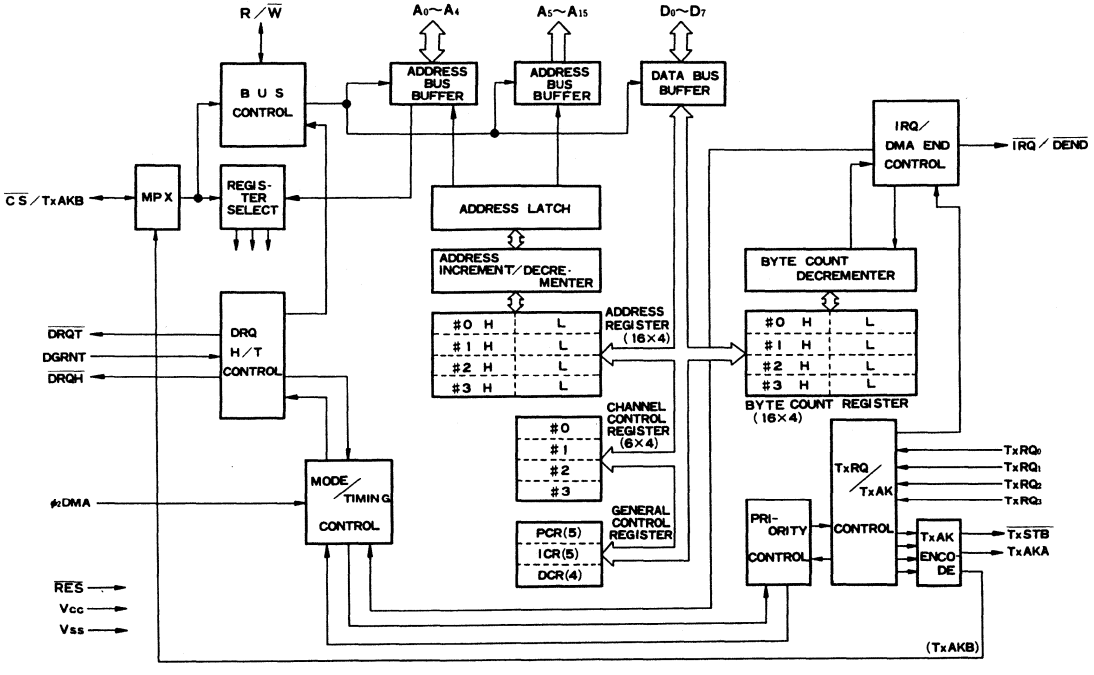


Figure 1 Expanded Block Diagram

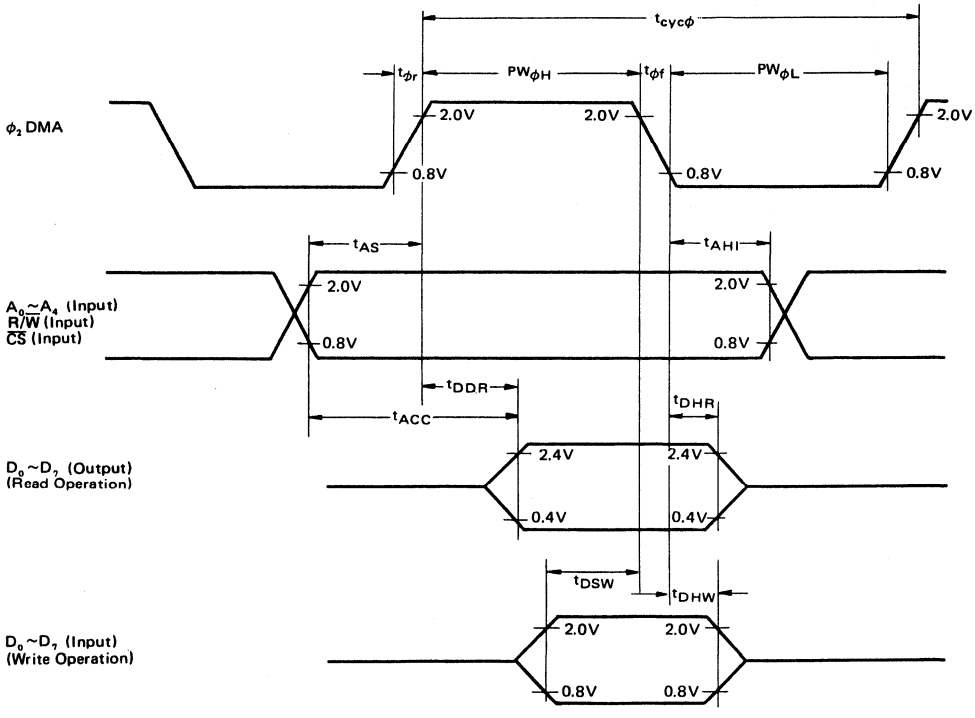


Figure 2 Read/Write Sequence

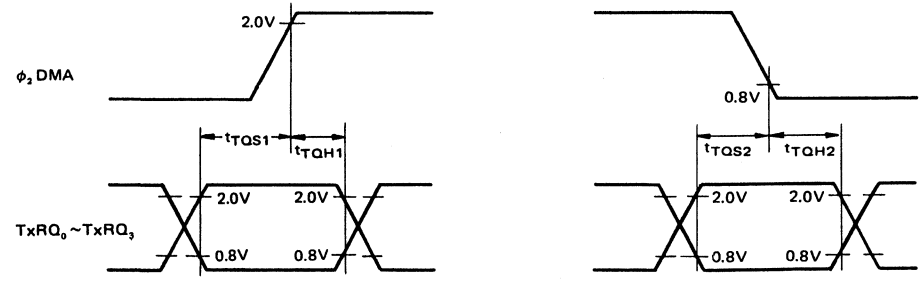


Figure 3 Timing of TxRQ Input

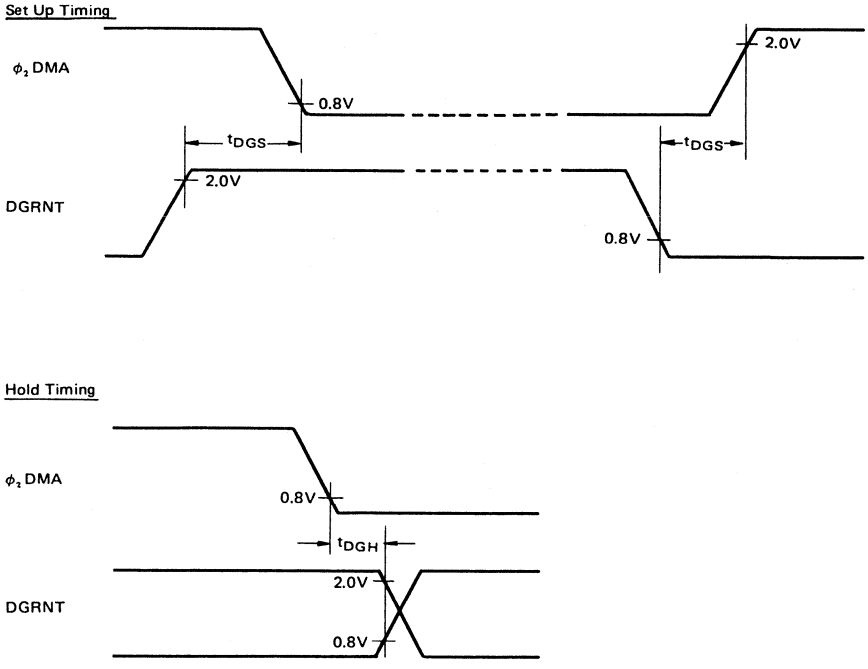


Figure 4 Timing of DGRNT Input

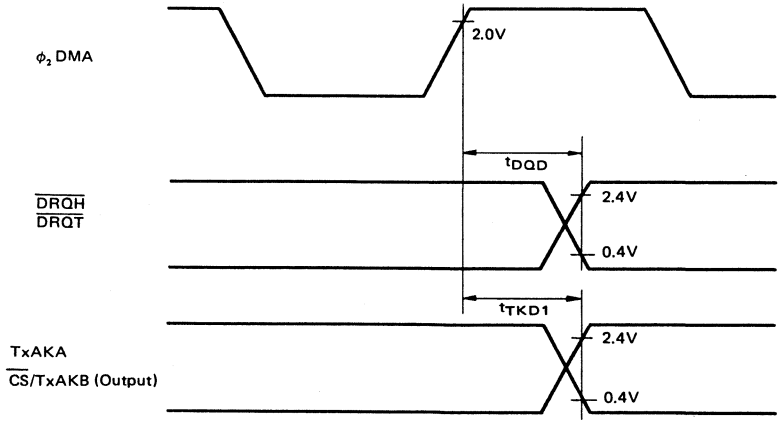


Figure 5 Timing of $\overline{DRQH}$, $\overline{DRQT}$, TxAK Outputs

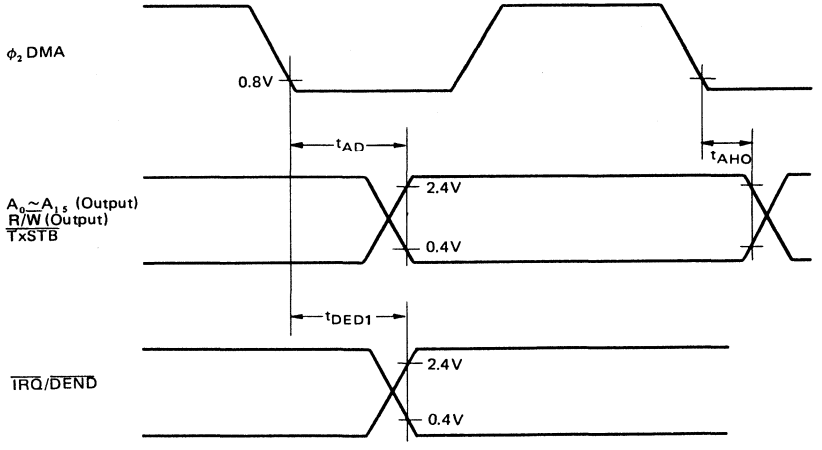


Figure 6 Timing of Address and IRQ/DEND Outputs

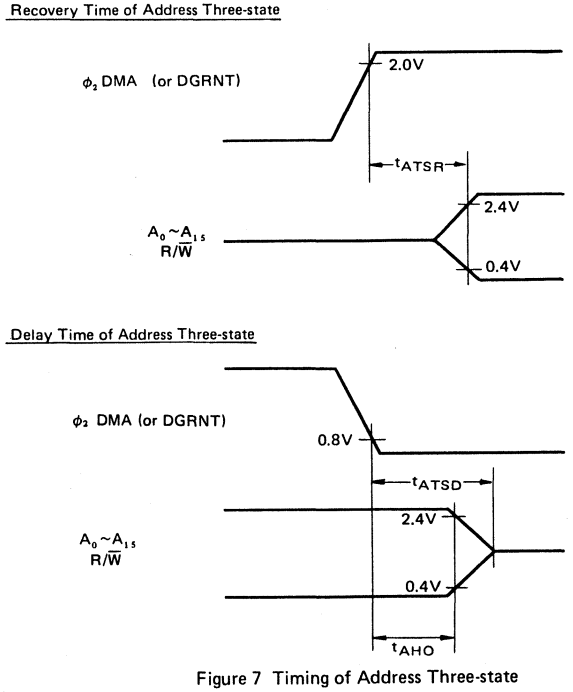


Figure 7 Timing of Address Three-state

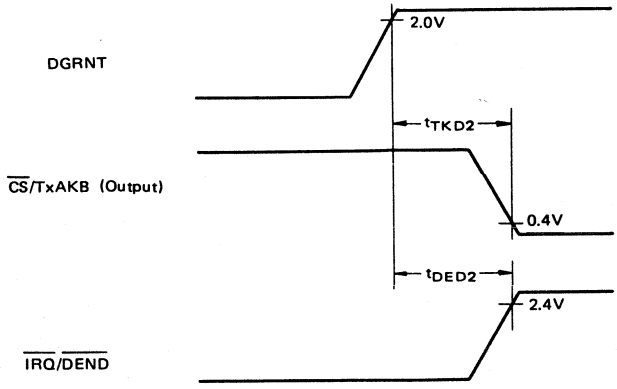
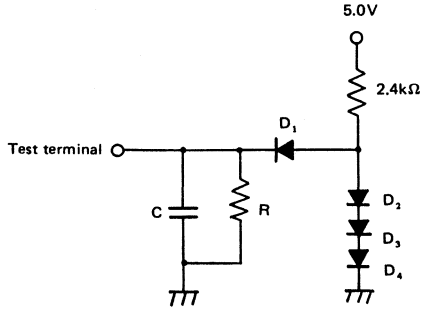


Figure 8 Timing of Synchronous DGRNT Output



Test terminal	C	R
D_0~D_7	130 pF	11 kΩ
A_0~A_{15}, R/$\overline{W}$	90 pF	16 kΩ
$\overline{CS}$/TxAKB	50 pF	24 kΩ
All other outputs	30 pF	24 kΩ

D_1~D_4 : 1S2074 Ⓗ or equivalent.

Figure 9 Load Circuit

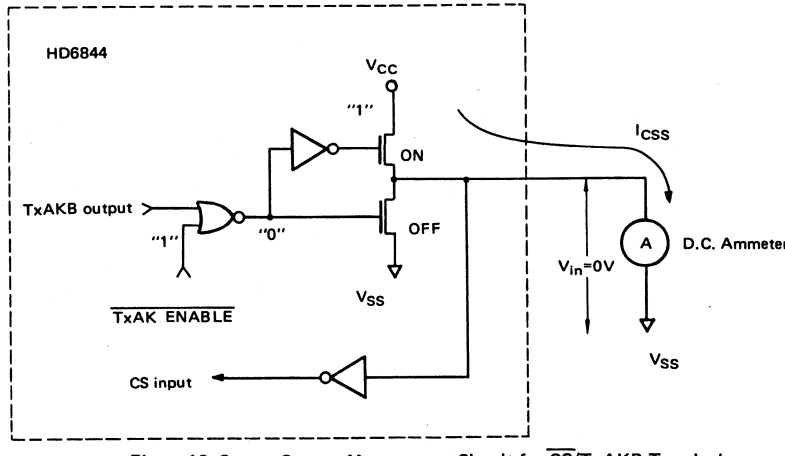


Figure 10 Source Current Measurement Circuit for $\overline{CS}$/TxAKB Terminal

■ DEVICE OPERATION

The DMAC has fifteen addressable registers, eight of them are sixteen bits in length. Each channel has a separate Address Register and a Byte Count Register, each of which is sixteen bits. There are also four Channel Control Registers. The three General Control Registers common to all four channels are the Priority Control Register, the Interrupt Control Register, and the Data Chain Register.

To prepare a channel for DMA, the Address Registers must be loaded with the starting memory address and the Byte Count Register loaded with the number of bytes to be transferred. The bits in the Channel Control Register establish the direction of the transfer, the mode, and the address increment or decrement after each cycle. Each channel can be set for one of three transfer modes: Three-State Control (TSC) Steal, Halt Steal, or Halt Burst. Two read-only status bits in the Channel Control Register indicate when the channel is busy transferring data and when the DMA transfer is completed.

The Priority Control Register enables the transfer requests from the peripheral controllers and establishes either a fixed priority or rotating priority scheme of servicing these requests.

When the DMA transfer for a channel is complete (the Byte Count Register is zero), a $\overline{\text{DMA End}}$ signal is directed to the peripheral controller and an $\overline{\text{IRQ}}$ goes to the MPU. Enabling of these interrupts is done in the interrupt Control Register. The IRQ flag bit is read from this register.

Chaining of data transfers is controlled by the Data Chain Register. When enabled, the contents of the Address and Byte Count Registers for channel #3 are put into the registers of the channel selected for chaining when its Byte Count Register becomes zero. This allows for repetitively reading or writing a block of memory.

During the DMA mode, the DMAC controls the address bus and data bus for the system as well as providing the R/$\overline{\text{W}}$ line and a signal to be used as VMA. When a peripheral device controller desires a DMA transfer, it is requested by a Transfer Request. Assuming this request is enabled and meets the test of highest priority, the DMAC will issue a DMA Request. When the DMAC receives the DMA Grant, it gives a Transfer Acknowledge to the peripheral device controller, at which time the data is transferred. When the channel's Byte Count Register equals zero, the transfer is complete and a $\overline{\text{DMA End}}$ is given to the peripheral device controller, and an $\overline{\text{IRQ}}$ is given to the MPU.

• Initialization

During a power-on sequence, the DMAC is reset via the $\overline{\text{RES}}$ input. All registers, with the exception of the Address and Byte Count Registers, are set to a logic "0" state. This disables all requests and the Data Chain function while masking all interrupts. The Address, Byte Count, and Channel Control Registers must be programmed before the respective transfer request bit is enabled in the Priority Control Register.

• Transfer Modes

There are three ways in which a DMA transfer may be done. The one used is determined by the data transfer rate required, the number of channels attached, and the hardware complexity allowable. Refer to Figures 12, 16 and 17.

Two of the modes, TSC Steal and Halt Steal, are done by cycle-stealing from the MPU. The Three-State Control (TSC) Steal mode is initiated by the DMAC bringing the $\overline{\text{DRQT}}$ line "Low". This line goes to the system clock driver which returns a "High" on DGRNT on the rising edge of the system ϕ_1 clock. The DGRNT signal must cause the address control and data lines to go to the high impedance state. The DMAC now supplies the address from the Address Register of the channel requesting. It also supplies the R/$\overline{\text{W}}$ signal as determined from the Channel Control Register. After one byte is transferred, control is returned to the MPU. This method stretches the ϕ_1 and ϕ_2 clocks while the DMAC uses the memory.

The second method of cycle-stealing is the Halt Steal mode. This method actually halts the MPU instead of stretching the ϕ_1 clock for the transfer period. This mode is initiated by the DMAC bringing the $\overline{\text{DRQH}}$ line "Low". This line connects to the MPU $\overline{\text{HALT}}$ input. The MPU Bus Available (BA) line is the DGRNT input to the DMAC. While the MPU is halted, its Address Bus, Data Bus, and R/$\overline{\text{W}}$ are in the high impedance state. The DMAC now supplies the address and R/$\overline{\text{W}}$ line. After one byte is transferred, the $\overline{\text{HALT}}$ line is returned "High" and the MPU regains control. In this mode, the MPU stops internal activity and is removed from the system while the DMAC uses the memory.

The third mode of transfer is the Halt Burst mode. This mode is similar to the Halt Steal mode, except that the transfer does not stop with one byte. The MPU is halted while an entire block of data is transferred. When the channel's Byte Count Register equals zero, the transfer is complete and control is returned to the MPU. This mode gives the highest data transfer rate, at the expense of the MPU being inactive during the transfer period.

■ INPUT/OUTPUT FUNCTIONS

• DMAC Interface Signals for the MPU

The DMAC interfaces with the HMCS6800 MPU through the eight-bit bidirectional data bus, the $\overline{\text{CS}}$ line, five address lines, an $\overline{\text{IRQ}}$ line, the Read/Write line, and the $\overline{\text{RES}}$ line. These signals, in conjunction with the HMCS6800 VMA output, permit the MPU to have access to the DMAC. Four other lines associated with the MPU and the clock driver are the $\overline{\text{DRQT}}$, $\overline{\text{DRQH}}$, DGRNT, and the ϕ_2 DMA.

Bidirectional Data ($D_0 \sim D_7$)

The Bidirectional Data lines ($D_0 \sim D_7$) allow for data transfer between the DMAC and the MPU. The data bus output drivers are three-state devices that remain in the high impedance state except when the MPU performs DMAC read operations.

Chip Select/Transfer Acknowledge B ($\overline{\text{CS}}$/T x AKB)

This line is multiplexed, serving both as an input and an output. $\overline{\text{CS}}$/TxAKB is an output in the four-channel mode during the DMA transfer. At all other times, it is a high impedance TTL compatible input used to address the DMAC. The DMAC is selected when $\overline{\text{CS}}$/TxAKB is "Low". VMA must be used in generating this input to insure that false selects will not occur. Transfers of data to and from the DMAC are then performed under the control of the ϕ2 DMA, Read/Write, and $A_0 \sim A_4$ address lines. In the four-channel mode when TxAKB is needed, the $\overline{\text{CS}}$ gate must have an open-collector output (a pull-up resistor should not be used). In the two-channel mode, $\overline{\text{CS}}$/TxAKB is always an input.

Address Lines ($A_0 \sim A_4$)

Address lines $A_0 \sim A_4$ are both input and output lines. In the MPU mode, these are high impedance inputs used to address the DMAC registers. In the DMA mode, these lines are outputs which are set to the contents of the Address Register of the channel being processed.

Interrupt Request/DMA End ($\overline{IRQ}/\overline{DEND}$)

$\overline{IRQ}/\overline{DEND}$ is a TTL compatible, active "Low" output that is used to interrupt the MPU and to signal the peripheral controller that the data block transfer has ended. If the Interrupt has been enabled, the $\overline{IRQ}/\overline{DEND}$ line will go "Low" after the last DMA cycle of a transfer. An open collector gate must be connected to DGRNT and $\overline{IRQ}/\overline{DEND}$ to prevent false interrupts from the DEND signal when interrupts are not enabled. Refer to the section of "DMA End Control".

Read/Write (R/$\overline{W}$)

Read/Write is a TTL compatible line that is a high impedance input in the MPU mode and an output in the DMA mode. In the MPU mode, it is used to control the direction of data flow through the DMAC's input/output data bus interface. When Read/Write is "High" (MPU read cycle) and the chip is selected, DMAC data output buffers are turned on and a selected register is read. When it is "Low", the DMAC output drivers are turned off and the MPU writes into a selected register.

In the DMA mode, Read/Write is an output to drive the memory and peripheral controllers. Its state is determined by bit 0 of the Channel Control Register for the channel being serviced. When Read/Write is "High", the memory is read and the data from the memory is written into the peripheral controller. When it is "Low", the peripheral controller is read and its data stored in the memory. In the DMA mode, the DMAC data buffers are off.

Reset ($\overline{RES}$)

The $\overline{RES}$ input provides a means of resetting the DMAC from an external source. In the "Low" state, the $\overline{RES}$ input causes all registers, with the exception of the Address and Byte Count Registers, to be reset to the logic "0" state. This disables all transfer requests, masks all interrupts, disables the data chain function, and puts each Channel Control Register into the condition of memory write, Halt Steal transfer mode, and address increment.

- **Transfer Signals to the MPU**

Two DMA request output lines and a DMA Grant input line, together with the system clock, synchronize the DMAC with the MPU system.

DMA Request Three-State Control Steal ($\overline{DRQT}$)

This active "Low" output requests a DMA transfer for a channel configured for the TSC Steal transfer mode. This line is connected to the system clock driver, requesting a ϕ_1 clock stretch. It will remain in the "Low" state until the transfer has begun.

DMA Request Halt ($\overline{DRQH}$)

This active "Low" output requests a DMA transfer for a channel programmed for the Halt Steal or Halt Burst mode transfer. This line is connected directly to the MPU $\overline{HALT}$ input and remains "Low" until the last byte has begun to be transferred.

DMA Grant (DGRNT)

This is a high impedance input to the DMAC, giving it control of the system busses. For the TSC Steal mode, the signal comes from the system clock drive circuit (DMA Grant), indicating that the clock is being stretched. For either of the Halt modes, this signal is the Bus Available from the MPU, indicating that the MPU has halted and turned control of its busses over to the DMAC. For a design involving TSC Steal and Halt mode transfers, this input must be the OR of the clock driven DMA Grant and the MPU BA.

ϕ_2 DMA

Transferring in and out of the DMAC registers, sampling of channel request lines and gating of other control signals to the system is done internally in conjunction with the ϕ_2 DMA high impedance input. This input must be the system memory clock (non-stretched ϕ_2 clock).

- **Transfer Signals From the Peripheral Controller**

Transfer Request ($TxRQ_0 \sim TxRQ_3$)

Each of the four channels has its own high impedance input request for transfer line. The peripheral controller requests a transfer by setting its TxRQ line "High" (a logic "1"). The lines are sampled according to the priority and enabling established in the Priority Control Register. In the Steal mode and the first byte of the Halt Burst mode, the TxRQ signals are tested on the positive edge of ϕ_2 DMA and the highest priority channel is strobed. Once strobed, the TxRQs are not tested until that channel's data transfer is finished. In the succeeding bytes of the Halt Burst mode transfer, the TxRQ is tested on the negative edge of ϕ_2 DMA, and data is transferred on the next ϕ_2 DMA cycle if TxRQ is "High".

- **Transfer Signals to the Peripheral Controller**

Two encoded lines select the channel to be serviced. A strobe line acknowledges the request and performs the transfer. The $\overline{DEND}$ line signals to the peripheral controller that the DMA transfer is completed.

Transfer Acknowledge A (T x AKA)

The Transfer Acknowledge A (TxAKA) is a TTL compatible output used in conjunction with the $\overline{CS}$/TxAKB line to select the channel to be strobed for transfer and to give the DMA End Signal. In the two-channel mode, only TxAKA is used to select channel 0 or channel 1, and $\overline{CS}$/TxAKB is always an input.

Chip Select/Transfer Acknowledge B ($\overline{CS}$/TxAKB)

In the DMA mode, this dual purpose line is encoded together with TxAKA to select the channel being serviced. Table 1 shows the encoding order.

Table 1 Encoding Order

$\overline{CS}$/TxAKB	TxAKA	Channel #
0	0	0
0	1	1
1	0	2
1	1	3

Transfer Strobe ($\overline{TxSTB}$)

The $\overline{TxSTB}$ causes acknowledgement to be given to the peripheral controller and transfers the data to or from the memory. This line is also intended to be the VMA signal for the system in the DMA mode. In a one-channel system, $\overline{TxSTB}$ may be inverted and run to the peripheral controller's Acknowledge input. In a two or four-channel system, $\overline{TxSTB}$ enables the decode of TxAKA and $\overline{CS}$/TxAKB to select the device controller to be acknowledged.

Interrupt Request/DMA End ($\overline{IRQ}/\overline{DEND}$)

In the DMA mode, this dual purpose line is "Low" for the last byte of transfer, indicating a DMA End. This occurs when the Byte Count register decrements to zero.

This line, through the decode of TxAKA and $\overline{CS}$/TxAKB, can be used to strobe a DMA End to each device controller.

- **Address Lines to the Memory**

Address Lines ($A_0 \sim A_{15}$)

These output lines are in the high impedance state during the MPU mode. In the DMA mode, these lines are outputs which are set to the contents of the Address Register of the channel being processed.

■ THE DMAC REGISTERS

The HD6844 (DMAC) has Address Register (ADR), Byte Count Register (BCR), Channel Control Register (CHCR), and General Control Register (GCR).

General Control Register (GCR) is composed of Priority Control Register (PCR) that controls priority among the channels, Interrupt Control Register (ICR) that controls interrupt and Data Chain Control Register (DCR) that controls data chain function. Refer to Table 2 and Figure 1.

These are Read/Write registers and MPU can exchange the data with DMAC when $\overline{CS}$ is at "Low" level. $A_0 \sim A_4$ specifies the address of the registers. How to specify the registers is shown in Table 2.

2-byte ADR and BCR can be read or written by one instruction, using 2-byte instruction of the MPU.

- **Function of Internal Registers**

ADR (Address Register)

Each channel has 16-bit Address Register. Initial address of memory used for DMA transfer is programmed to this register. The contents of ADR are output to address bus ($A_0 \sim A_{15}$) during DMA transfer operation. When 1-byte transfer has completed, the 16-bit address is incremented or decremented by one.

The address which the MPU reads out is the renewed one, that is, the memory address for the next transfer. When 1-block transfer has completed, final memory address +1 or -1 is read out.

Table 2 Internal Registers of the DMAC

Register Name	Symbol	Channel	Address Bus Signal					Address (Hexadecimal)
			A_4	A_3	A_2	A_1	A_0	
Address Register	ADRH	0	0	0	0	0	0	00
	ADRL	0	0	0	0	0	1	01
Byte Count Register	BCRH	0	0	0	0	1	0	02
	BCRL	0	0	0	0	1	1	03
Address Register	ADRH	1	0	0	1	0	0	04
	ADRL	1	0	0	1	0	1	05
Byte Count Register	BCRH	1	0	0	1	1	0	06
	BCRL	1	0	0	1	1	1	07
Address Register	ADRH	2	0	1	0	0	0	08
	ADRL	2	0	1	0	0	1	09
Byte Count Register	BCRH	2	0	1	0	1	0	0A
	BCRL	2	0	1	0	1	1	0B
Address Register	ADRH	3	0	1	1	0	0	0C
	ADRL	3	0	1	1	0	1	0D
Byte Count Register	BCRH	3	0	1	1	1	0	0E
	BCRL	3	0	1	1	1	1	0F
Channel Control Register	CHCR	0	1	0	0	0	0	10
	CHCR	1	1	0	0	0	1	11
	CHCR	2	1	0	0	1	0	12
	CHCR	3	1	0	0	1	1	13
Priority Control Register	PCR	–	1	0	1	0	0	14
Interrupt Control Register	ICR	–	1	0	1	0	1	15
Data Chain Control Register	DCR	–	1	0	1	1	0	16

(NOTE) 1) All the registers can be accessed by Read/Write operation. Unused bit of the register is read out "0".
2) H/L of ADR and BCR means the higher (H) 8 bits/the lower (L) 8 bits of a 16-bit register.
3) Being allocated to continuous address, 16-bit ADR and BCR can be read or written by one instruction, using MPU's 2-byte LOAD/STORE instruction.

```
                 Register Address
e.g. LDX   $ * * 0C   ............ [(ADRH 3) → (Index Register H)]
           Address of DMAC         [(ADRL 3) → (Index Register L)]
```

BCR (Byte Count Register)

Each channel has a 16-bit Byte Count Register. Number of DMA transfer words is programmed into this register. The content of the Byte Count Register is decremented by one everytime one-byte transfer has completed. When it becomes "0", $\overline{DEND}$ output goes "Low" level and informs I/O controller of the end of one-block DMA transfer. When $\overline{IRQ}$ is not masked, $\overline{IRQ}$ output goes "Low" level and MPU is interrupted to be informed of the end of DMA transfer. Moreover, $\overline{IRQ}$ and $\overline{DEND}$ signals are output, multiplexed with $\overline{IRQ}/\overline{DEND}$ pin.

CHCR (Channel Control Register)

Each channel has Channel Control Register. This register is used to program the control information of its corresponding channel. Structure of CHCR is shown in Table 3.

(1) R/W Control (specifies the direction of transfer)

Bit – CHCR Bit 0

This bit controls the direction of DMA transfer. When it is at "1", R/$\overline{W}$ signal of DMAC goes "High" level during DMA transfer operation. This means to read out memory and write into I/O controller, that is, data is transferred from memory to I/O controller.

When it is at "0", R/$\overline{W}$ output goes "Low" level and data is transferred from I/O controller to memory.

Table 3 Bit Structure of CHCR (Channel Control Register)

Bit No.	Name	Read/Write	Function	
			"1"	"0"
0	R/W	R/W	Transfer from memory to I/O controller (R/$\overline{W}$ output = "High")	Transfer from I/O controller to memory (R/$\overline{W}$ output = "Low")
1	Burst/Cycle Steal	R/W	Burst Mode	Cycle Steal Mode*
2	TSC/HALT	R/W	TSC Mode	HALT Mode*
3	Address down/up	R/W	Address: -1	Address: +1
4	Not used	–	–	–
5	Not used	–	–	–
6	Busy/Ready Flag	R	Busy (DMA Transfer Operation)	Ready (No DMA Transfer Operation)
7	DEND Flag	R	DMA End & Interrupt	No Interrupt

* Burst transfer in TSC mode is prohibited. R: Read, W: Write

Note that during DMA transfer operation, the function of R/$\overline{W}$ signal is accommodated to the memory Read/Write operation. Therefore, on the side of I/O device during DMA transfer operation, R/$\overline{W}$ input should be interpreted in inverse of the MPU Read/Write. That is, data should be output when R/$\overline{W}$ input is at "Low" level (In the case of MPU's read operation, I/O device outputs the data when it is at "High" level).

This arises from that during DMA transfer operation, I/O side performs data transfer independently instead of MPU. Moreover, such family LSI as HD6843 (FDC), etc. has this function and R/$\overline{W}$ signal is automatically interpreted inversely.

(2) Burst/Cycle Steal Bit – CHCR Bit 1

This bit is used to decide that DMA transfer should be performed in burst mode or cycle steal mode. When it is at "1", it specifies burst mode. That is, once DMA transfer is performed, MPU remains stopped until one-block data transfer is completed.

When this bit is "0", it specifies cycle steal mode. That is, everytime one-byte transfer has completed, MPU takes back the bus control, and DMA transfer and MPU operation are performed in time sharing.

(NOTE) Only in the case of HALT mode, burst mode can be specified. In TSC mode, burst mode cannot be specified.

(3) TSC/HALT Mode Bit – CHCR Bit 2

This bit is used to decide that DMA transfer should be performed by using MPU's TSC function or HALT function. When it is at "0", DMA transfer request signal is output from $\overline{DRQH}$ of DMAC.

When it is at "1", DMA transfer request signal is output from $\overline{DRQT}$ of DMAC.

(4) Address down/up Bit – CHCR Bit 3

This bit is used to decide that the address of memory region used for DMA transfer should be renewed up (increment of address) or down (decrement of address). When it is at "1", the address is decremented by one after one-byte transfer. When it is at "0", the address is incremented by one.

(5) Busy/Ready Flag Bit – CHCR Bit 6

This bit is a status flag to indicate whether its corresponding channel is performing DMA transfer or not. (READ only)

When it receives the first TxRQ of its corresponding channel, it goes to "1". When one-block transfer is completed and BCR becomes "0", it is reset to "0".

Also this flag is cleared when corresponding TxRQ Enable Bit in the PCR becomes "0".

(6) DEND Flag Bit – CHCR Bit 7

This bit is an interrupt flag to indicate that one-block DMA transfer of its corresponding channel has completed. (READ only).

When one-block transfer of its corresponding channel is completed and BCR becomes "0", it goes to "1". As soon as this flag is read out, i.e. CHCR of this channel is read

out, it is reset to "0".

Moreover, this bit is connected to $\overline{IRQ}$ output. When it is at "1" and IRQ enable bit (within ICR register described later) is at "1", $\overline{IRQ}$ output goes "Low" level.

PCR (Priority Control Register)

Priority Control Register is a 5-bit register to decide the operation mode of priority control circuit. Structure of PCR is shown in Table 4.

Table 4 Bit Structure of PCR (Priority Control Register)

Bit No.	Name	Read /Write	Function	
			"1"	"0"
0	TxRQ Enable #0 ($TxEN_0$)	R/W	TxRQ of Channel 0 is accepted.	TxRQ of Channel 0 is not accepted.
1	TxRQ Enable #1 ($TxEN_1$)	R/W	TxRQ of Channel 1 is accepted.	TxRQ of Channel 1 is not accepted.
2	TxRQ Enable #2 ($TxEN_2$)	R/W	TxRQ of Channel 2 is accepted.	TxRQ of Channel 2 is not accepted.
3	TxRQ Enable #3 ($TxEN_3$)	R/W	TxRQ of Channel 3 is accepted.	TxRQ of Channel 3 is not accepted.
4	Not used	–	–	–
5		–	–	–
6		–	–	–
7	Rotate Control	R/W	Rotate Mode	The order of priority is fixed at numerical order.

R: Read, W: Write

(1) TxRQ Enable Bit ($TxEN_0$~$TxEN_3$) – PCR Bit 0~3

Each channel has this TxRQ Enable bit. When it is at "1", TxRQ input of its corresponding channel is accepted to perform DMA transfer. When it goes to "0", TxRQ of its corresponding channel is masked not to be received and TxAK is not output. During DMA transfer operation, when this bit goes to "0" before BCR becomes "0", following TxRQ input is not accepted and DMA transfer is interrupted. Then contents of ADR and BCR remain unchanged. When it rises to "1" again, DMA transfer is reopened. Therefore, in the case of cycle steal DMA, it is possible for the program to change the priority of the specific channel temporarily by manipulating this bit.

(2) Rotate Control Bit – PCR Bit 7

When this bit is at "0", the order of priority among DMA channels is fixed at numerical order. That is, Channel 0 is given a first priority and then is followed by Channel $1 \rightarrow 2 \rightarrow 3$.

When this bit is at "1", priority control is due to rotate mode. That is, the channel that ended in the first time is given a first priority and the channel ended in the last time is controlled to be given a last priority.

ICR (Interrupt Control Register)

Interrupt Control Register is a 5-bit register to control $\overline{IRQ}$ output. Its structure is shown in Table 5.

(1) IRQ Enable Bit – ICR Bit 0~3

Each channel has IRQ Enable Bit. When this bit is at "1" and DEND Flag of its corresponding channel is set to "1", $\overline{IRQ}$ output goes "Low" level. But when it is at "0", $\overline{IRQ}$ output is masked not to be output even if DEND Flag is set to "1".

These bits enable to control to output only a necessary channel to $\overline{IRQ}$.

(2) IRQ Flag – ICR Bit 7

This is a read-only bit and the status of $\overline{IRQ}$ output is directly reflected on it. That is, when $\overline{IRQ}$ output goes to "Low" level, it becomes "1".

$\overline{IRQ}$ output of DMAC is output as logical OR of 4-channel DEND Flag according to the following equation.

$IRQ = (DEND_0 \cdot IRQ\ Enable_0) + (DEND_1 \cdot IRQ\ Enable_1) + (DEND_2 \cdot IRQ\ Enable_2) + (DEND_3 \cdot IRQ\ Enable_3)$

DCR (Data Chain Control Register)

Data Chain Control Register is a 4-bit register and three of those bits are used to control data chain function. Remaining one bit is used to specify 2-channel/4-channel mode.

Structure of DCR is shown in Table 6.

(1) Data Chain Enable Bit – DCR Bit 0

When this bit is at "1", data chain function of DMAC is enabled. That is, when DMA transfer of a specified channel has completed and BCR goes to "0", the contents of ADR and BCR of Channel #3 are automatically transferred to ADR and BCR of the specified channel.

(2) Data Chain Channel Bit – DCR Bit 1~2

These bits are used to specify which channel should be used for the data chain. How to specify the channel is shown in Table 7. Data Chain Channel bit specifies the channel to which data should be transfered from Channel #3. Channel #3 contains the data for replacement. Channel #3 is fixed and cannot be changed.

(3) 2/4-channel Mode Bit – DCR Bit 3

This bit has no relation to the data chain function.

It is used to specify whether $\overline{CS}$/TxAKB is used for only input pin or I/O pin. When this bit is "0", $\overline{CS}$/TxAKB becomes $\overline{CS}$ input pin in 2-channel mode since TxAKB output is not necessary for application up to 2-channel.

When this bit is "1", $\overline{CS}$/TxAKB becomes I/O pin in 4-channel mode (See Fig. 11).

Table 5 ICR (Interrupt Control Register)

Bit No.	Name	Read /Write	Function	
			"1"	"0"
0	IRQ Enable #0	R/W	$\overline{IRQ}$ of Channel 0 is able to be output.	$\overline{IRQ}$ output of Channel 0 is masked.
1	IRQ Enable #1	R/W	$\overline{IRQ}$ of Channel 1 is able to be output.	$\overline{IRQ}$ output of Channel 1 is masked.
2	IRQ Enable #2	R/W	$\overline{IRQ}$ of Channel 2 is able to be output.	$\overline{IRQ}$ output of Channel 2 is masked.
3	IRQ Enable #3	R/W	$\overline{IRQ}$ of Channel 3 is able to be output.	$\overline{IRQ}$ output of Channel 3 is masked.
4	Not used	–	–	–
5		–	–	–
6		–	–	–
7	IRQ Flag	R	$\overline{IRQ}$ output "Low"	$\overline{IRQ}$ output "High" (off state)

R: Read, W: Write

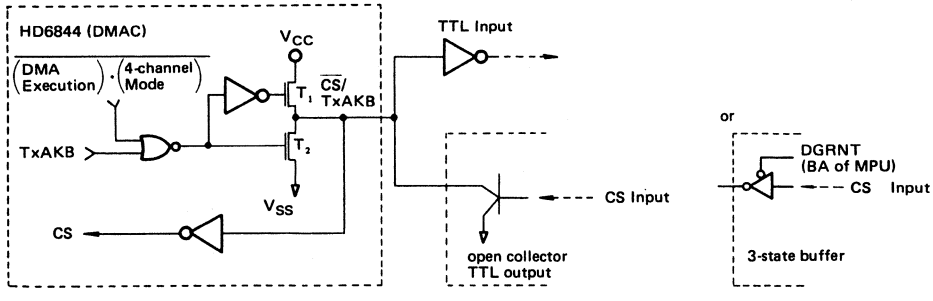
Table 6 Bit Structure of DCR (Data Chain Control Register)

Bit No.	Name	Read /Write	Function	
			"1"	"0"
0	Data Chain Enable	R/W	Data Chain is performed.	Data Chain is not performed.
1	Data Chain Channel	R/W	The channel which performs Data Chain is specified. (The channel where contents of ADR and BCR of Channel #3 are loaded.)	
2		R/W		
3	2/4-Channel Mode	R/W	4-Channel Mode ($\overline{CS}$/TxAKB is I/O pin.)	2-Channel Mode ($\overline{CS}$/TxAKB is designated to only input pin.)
4	Not used	–	–	–
5		–	–	–
6		–	–	–
7		–	–	–

R: Read, W: Write

Table 7 How to specify Data Chain Channel

DCR Bit 1	DCR Bit 2	Specified Channel
0	0	Channel #0
1	0	Channel #1
0	1	Channel #2
1	1	–



In $\overline{CS}$ input mode T1 turns ON and T2 turns OFF. T1 functions as pull-up resistance.

Figure 11 How to Use CS/TxAKB Pin

■ OPERATION OF THE DMAC

• Transfer Mode of the DMAC

There are three DMA transfer modes such as HALT Cycle Steal, HALT Burst and TSC Cycle Steal. Operation in each mode is explained in the following.

HALT Cycle Steal Mode

This is a basic DMA transfer mode utilizing HALT state of MPU. In this mode, everytime 1-byte transfer has completed, MPU takes back the bus control and executes instruction cycle. That is, DMA transfer and MPU operation are performed in time sharing.

Timing chart is shown in Fig. 12 and flow chart is shown in Fig. 13. Procedure of transfer operation is the following. (No. ① ~ ⑪ in Fig. 12 correspond to the following items.)

① $TxRQ_0$~$TxRQ_3$ input is checked at the rising edge of ϕ2DMA. When it is at "High" level, it gets into the following operation.
② $\overline{DRQH}$="Low" is output and MPU is requested to stop its operation.
③ TxAKA is driven (Level output).
④ MPU stops its operation and DMAC waits until DGRNT goes to "High" level.
⑤ When DGRNT goes to "High" level, DMAC drives TxAKB, A_0~A_{15} and R/$\overline{W}$ lines.
⑥ $\overline{TxSTB}$ is given to perform DMA transfer.
⑦ Address is incremented or decremented by one and number of transfer words is decremented by one.
⑧ When $\overline{DRQH}$ rises to "High" level, MPU gets into Instruction Cycle again.
⑨ TxRQ falls to "Low" level.
⑩ A_0~A_{15} and R/$\overline{W}$ get into high impedance state again.
⑪ DGRNT falls to "Low" level.

[Note] $TxRQ_0$~$TxRQ_3$ input in HALT cycle steal is, in principle as shown in Fig. 12, set to "High" every 1-byte transfer on account of I/O request. When $\overline{TxSTB}$ of the DMAC is driven, it is reset to "Low". Take care not to be against this principle, or the following states may happen.

(1) In the case where TxRQ becomes "High", but it is reset to "Low" before DGRNT becomes "High". In this case, the DMAC is in the wait state without sending out $\overline{TxSTB}$ until TxRQ rises to "High" again. As $\overline{DRQH}$ remains "Low" the MPU is forced to be stopped, and the system is in dead lock state until TxRQ rises to "High" again (Fig. 14).

(2) In the case where TxRQ is not reset to "Low" though $\overline{TxSTB}$ has been driven.
In this case, unless TxRQ returns to "Low" by the time ϕ_2DMA rises after $\overline{TxSTB}$ has risen to "High", it is considered as a new I/O request, which leads the above-mentioned operation ①,② —→ .
If TxRQ falls to "Low" immediately after that, the same state as (1) happens (Fig. 15).

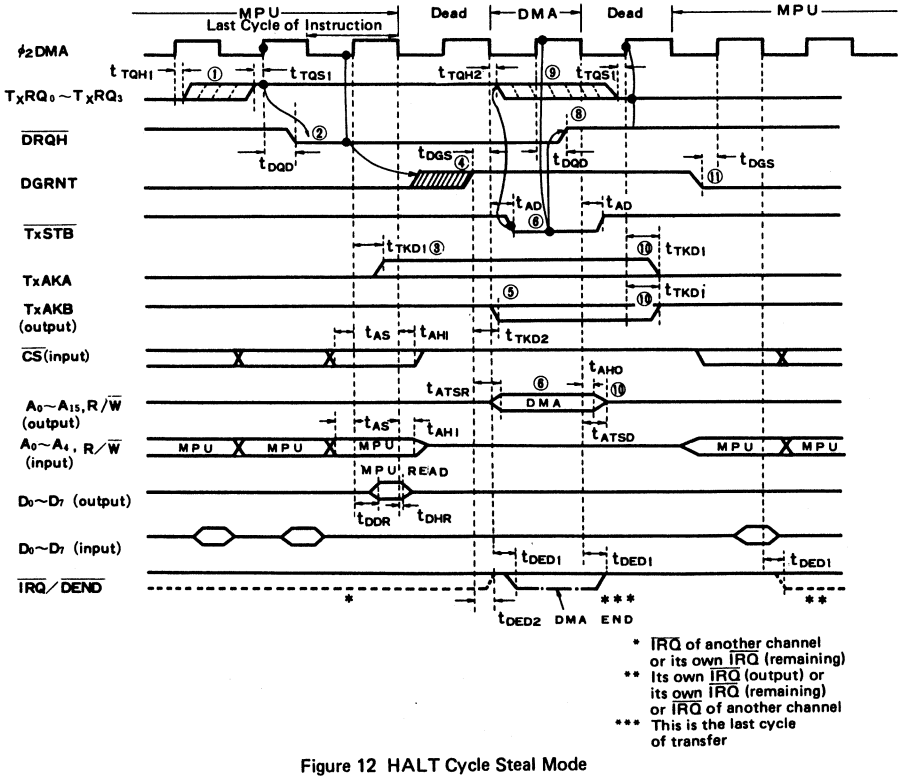


Figure 12 HALT Cycle Steal Mode

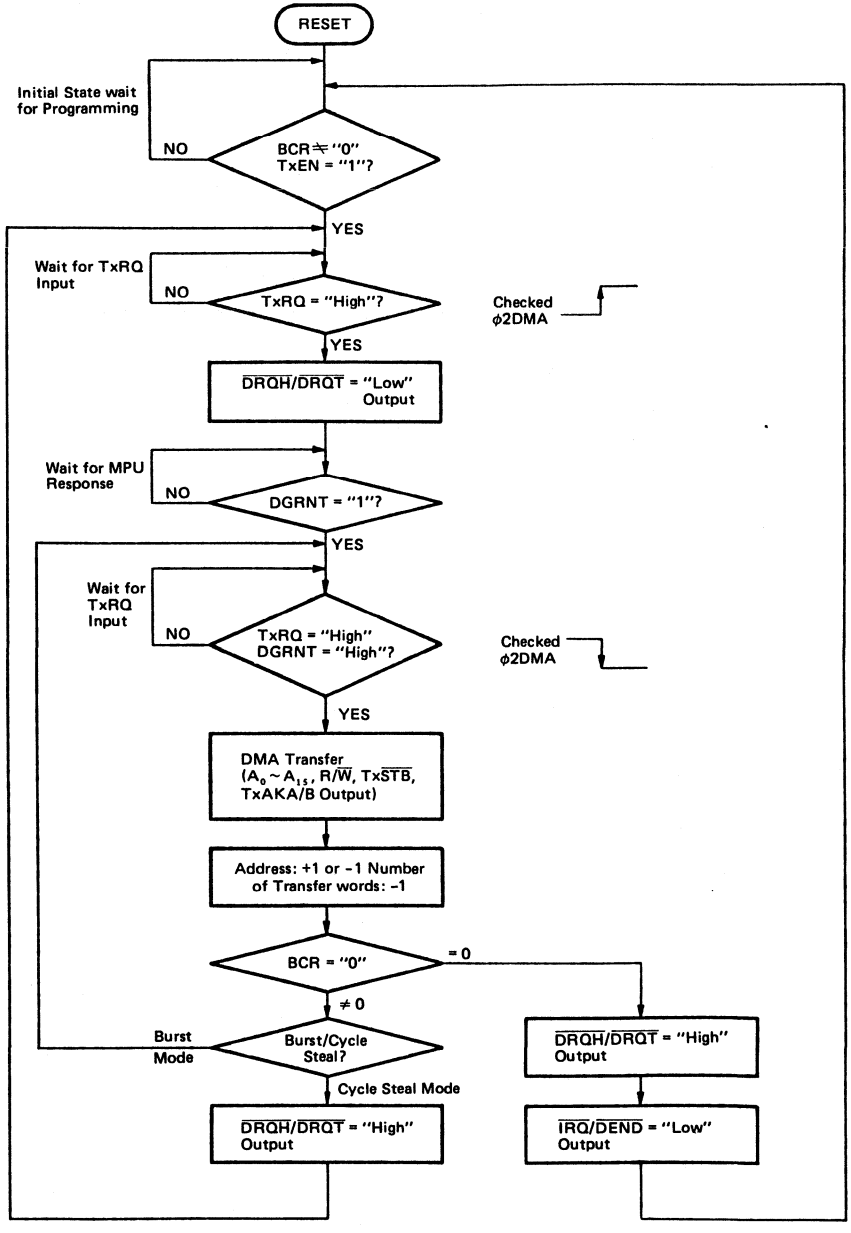


Figure 13 Flow Chart of DMAC Operation

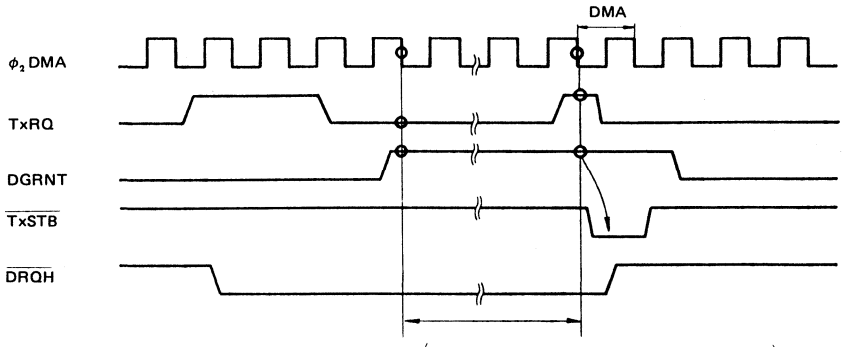


Figure 14 Extraordinary TxRQ Input (1)

(In the case where TxRQ is reset to "Low" before the transfer)

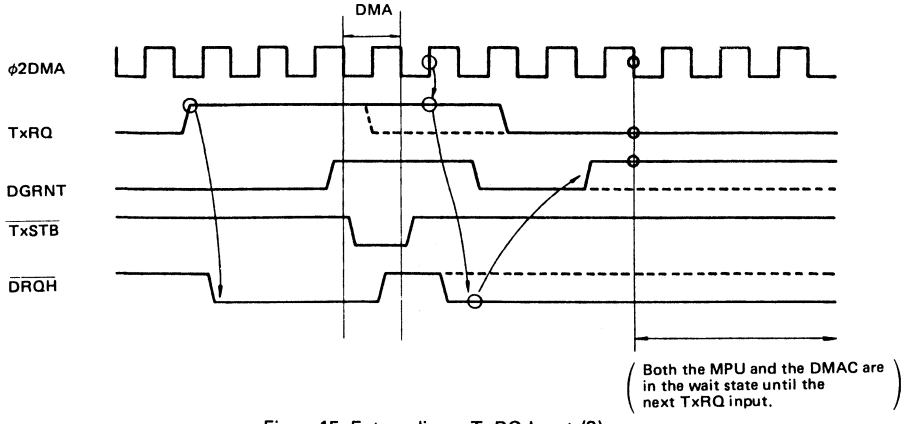


Figure 15 Extraordinary TxRQ Input (2)

(In the case where TxRQ doesn't fall to "Low" after the transfer has been completed.)

HALT Burst Mode

In the case of cycle steal mode, MPU gets into Instruction Cycle everytime 1-byte transfer has completed. But in the case of burst mode, MPU remains stopped until 1-block transfer is finished. That is, $\overline{DRQH}$ continues to be output "Low" level until BCR becomes "0".

Its timing chart and flow chart are shown in Fig. 16 and Fig. 13 respectively. Procedure of transfer is the following (No. ① ~ ⑭ in Fig. 16 correspond to the following items).

① TxRQ input is checked at the rising edge of ϕ_2 DMA. When it is at "High" level, it gest into the following operation.
② $\overline{DRQH}$="Low" level is given and MPU is requested to stop its operation.
③ TxAKA is driven.
④ MPU stops and DMAC waits for DGRNT rising "High" level.
⑤ When DGRNT rises "High" level, DMAC drives TxAKB, A_0 ~A_{15}, and R/$\overline{W}$ lines.
⑥ $\overline{TxSTB}$ is sent out to perform DMA transfer.
⑦ Address is incremented or decremented by one and number of transfer words is decremented by one.
⑧ TxRQ falls to "Low" level.
⑨ When number of transfer words is 0, from ⑪ to ⑭ operations are performed.
⑩ When BCR is not "0", TxRQ is checked at the falling edge of ϕ_2 DMA.When TxRQ is at "High" level, DMA transfer is performed through ⑥ ~ ⑧ again. When TxRQ is not at "High" level, DMAC waits for becoming "High" level.
⑪ $\overline{IRQ/DEND}$ output goes to "Low" level.
⑫ $\overline{DRQH}$ output rises to "High" level and MPU gets into Instruction Cycle again.
⑬ A_0 ~ A_{15} and R/$\overline{W}$ get into high impedance state.
⑭ DGRNT falls to "Low" level.

The transfer of the first byte (① ~ ⑥) is performed in the same way as that in HALT cycle steal mode. But in the second-byte and subsequent transfer, TxRQ is checked at the falling edge of ϕ_2 DMA and if TxRQ is at "High" level, DMA transfer is performed at the following cycle. Therefore, a high-speed response (MAX. 1 byte/1 cycle) is feasible.

In burst mode, TxRQ should be also, in principle, set to "High" when I/O request is asserted, and reset to "Low" when $\overline{TxSTB}$ goes to "Low". If TxRQ is asserted as level input without being reset, DMA transfer is performed at all cycles of ϕ_2 DMA since TxRQ is always at "High" level at the falling edge of ϕ_2 DMA. Its example is shown in the second-byte and the third-byte transfer in Fig. 16.

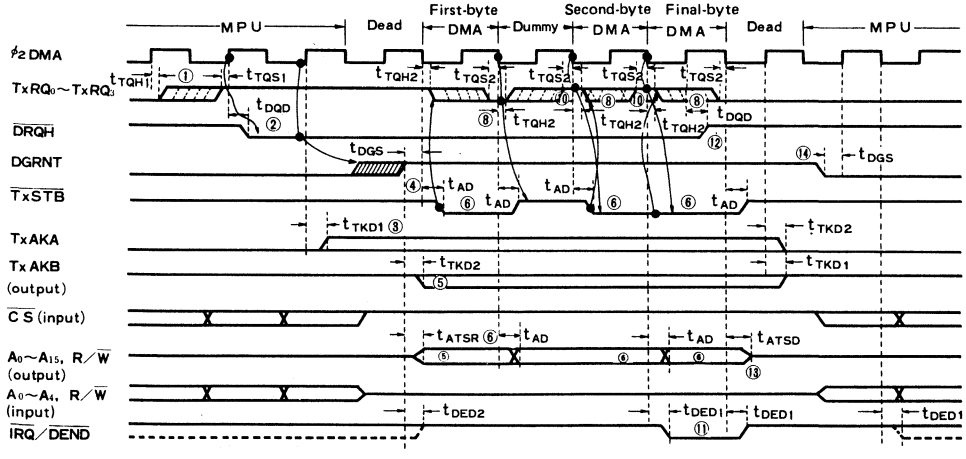


Figure 16 HALT Burst Mode

TSC Cycle Steal Mode

In the above-mentioned modes, DMA is performed by using the HALT function of the MPU. In TSC cycle steal mode, DMA is performed by using the TSC function of the MPU.

Its timing chart and flow chart are shown in Fig. 17 and Fig. 13 respectively.

Basic operation of the DMAC is the same as that in HALT cycle steal mode, but the detailed timing is different. The difference is explained in the following.

(1) $\overline{DRQT}$ is used for DMA transfer request instead of $\overline{DRQH}$.
(2) $\overline{DRQT}$ is sent to the external clock control circuit to extend clock E (ϕ_2) of MPU.
(3) To DGRNT, the external clock control circuit inputs response signals.

In TSC mode, there isn't a burst mode. Because the MPU clock cannot be extended for a long time because MPU performs dynamic operation. When TSC mode is specified, $\overline{DRQT}$ returns to "High" and the MPU gets into the instruction cycle everytime 1-byte transfer has finished.

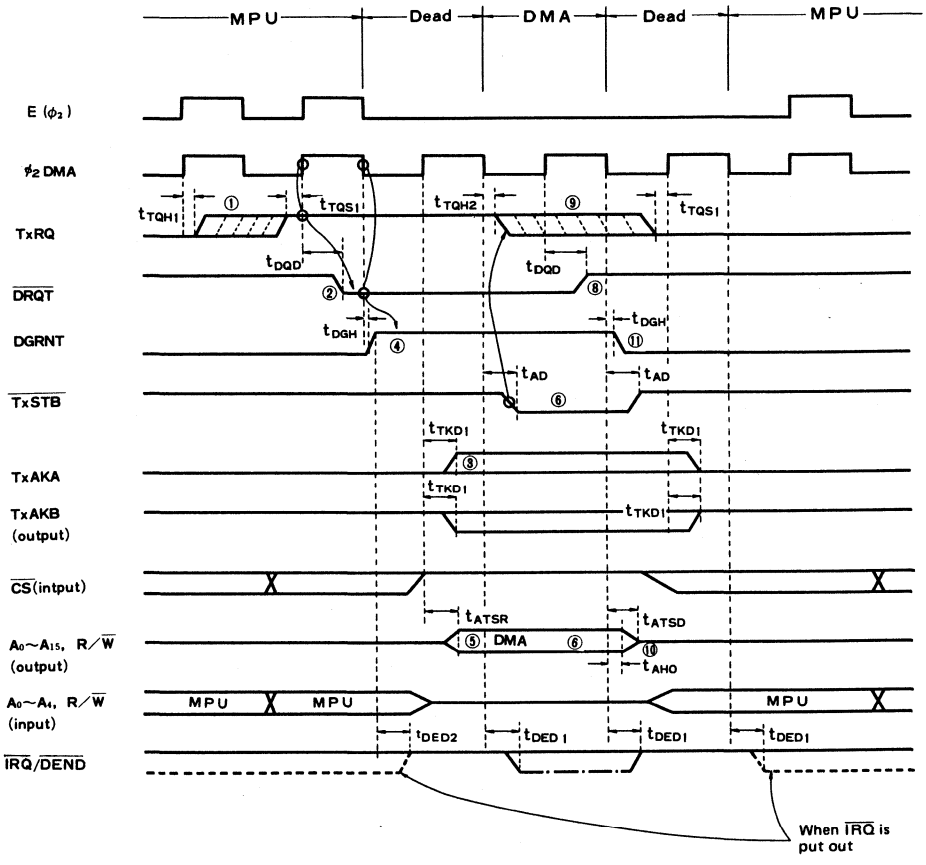


Figure 17 TSC Cycle Steal Mode

- **Priority Control**

Basic priority Control

There are two kinds of the DMAC priority control function. One is to mask TxRQ on each channel by TxRQ Enable bit of PCR. The other is priority-order-determining-circuit which the DMAC has as a hardware.

Moreover, the priority-order-determining-circuit has two operation modes (the rotate mode and the normal mode).

Structure of the priority control circuit is shown in Fig. 18. As shown in Fig. 18, TxRQ of the channel whose TxRQ Enable bit is at "1" level becomes an input of the priority-order-determining-circuit. Then it is checked whether TxRQ is at "High" level or not.

(Note) In this case, ZERO flag needs to be at "1" level. ZERO flag will be described later.

If one of $TxRQ_0 \sim TxRQ_3$ is at "High" level, its channel is selected, being given a first priority. Then it is latched by an executing-channel-number-latch-circuit to perform DMA transfer. Once an executing channel is determined and latched, it is unchanged until its DMA transfer has been completed. That is, the channel number strobe signal of DMAC doesn't go to "1" and the contents of the channel-number-latch-circuit are unchanged. In the cycle steal mode, the channel is fixed until 1-byte transfer has completed. In the burst mode, it is fixed until BCR becomes "0".

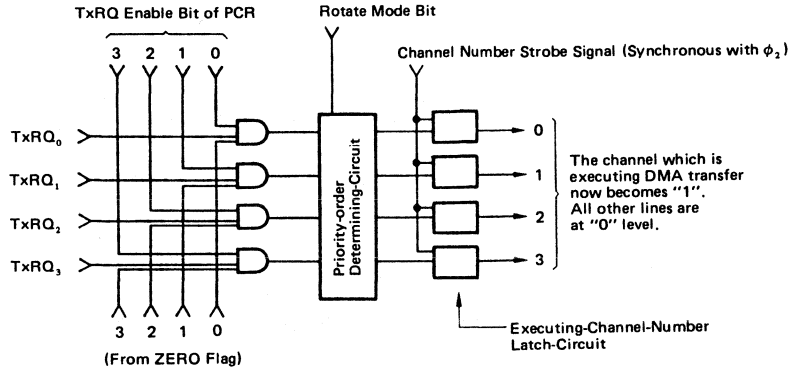


Figure 18 Structure of Priority Control Circuit

Therefore, once a long-period DMA transfer of a channel is performed in the burst mode, other channels need to wait until it has completed even if they have higher priority than the channel. Take much care to this point in designing response time to TxRQ of DMA channel.

(Note) As explained above, TxRQ input is latched internally. So once it is accepted and latched, the channel number cannot be changed even though it returns to "Low". But as explained in HALT Cycle Steal Mode, DMA transfer is not performed unless TxRQ rises to "High" again.

Strobe timing of executing-channel-number-latch-circuit which allow modification or decision of executing channel is shown in Fig. 19.

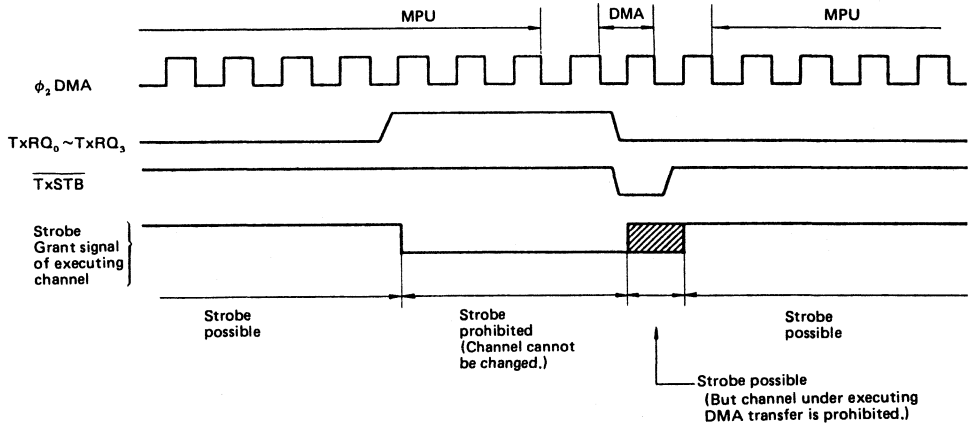


Figure 19 Strobe Timing of Executing-Channel-Number-Latch-Circuit (the cycle steal mode)

But, as shown in Fig. 19, only the channel under executing DMA transfer is prohibited to accept TxRQ during DMA transfer operation, in order that one more byte transfer may not be performed when the reset timing of TxRQ is delayed. Strobe timing in the burst mode is shown in Fig. 20.

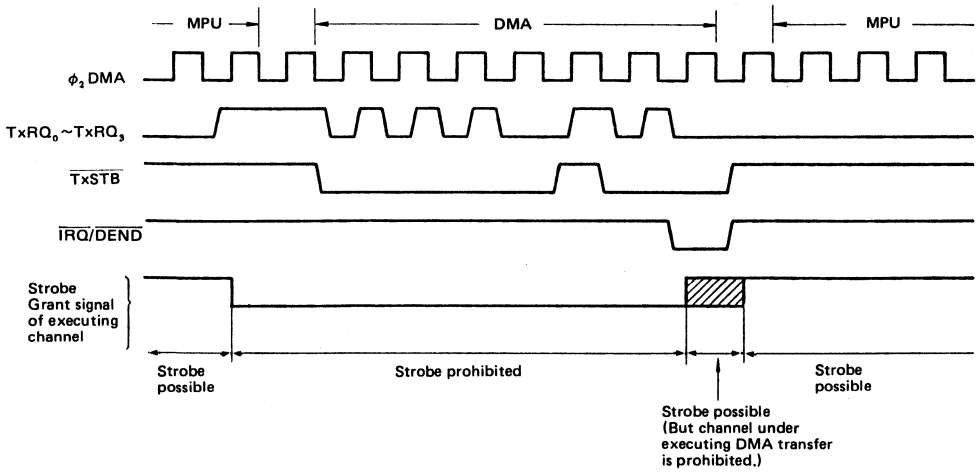


Figure 20 Strobe Timing of Executing-Channel-Number-Latch-Circuit (the burst mode)

Rotate Mode

There are two operation modes in priority-order-determining circuit. These are Normal Mode and Rotate Mode. In the normal mode, the order of priority is fixed at numerical order. (Channel 0 is given a first priority and then is followed by Channel 1 → 2 → 3.) In the rotate mode, the channel next to the channel with which DMA was executed in the last sequence, is given a first priority and the channel in the last sequence is given a last priority. But immediately after it gets into the reset state, the order of priority is the following: Channel 0 → 1 → 2 → 3.

An example of the rotate mode is shown in Fig. 21.

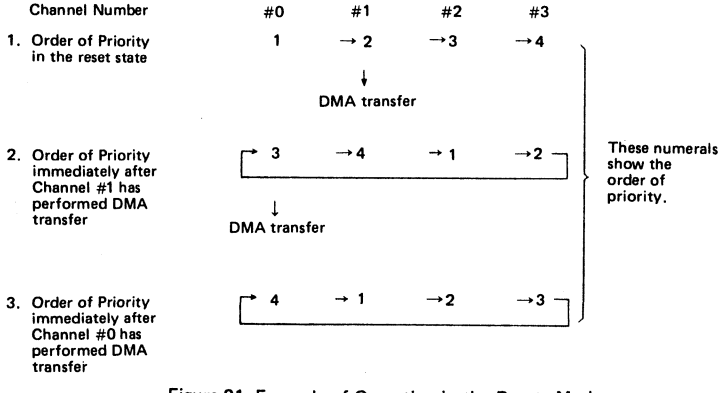
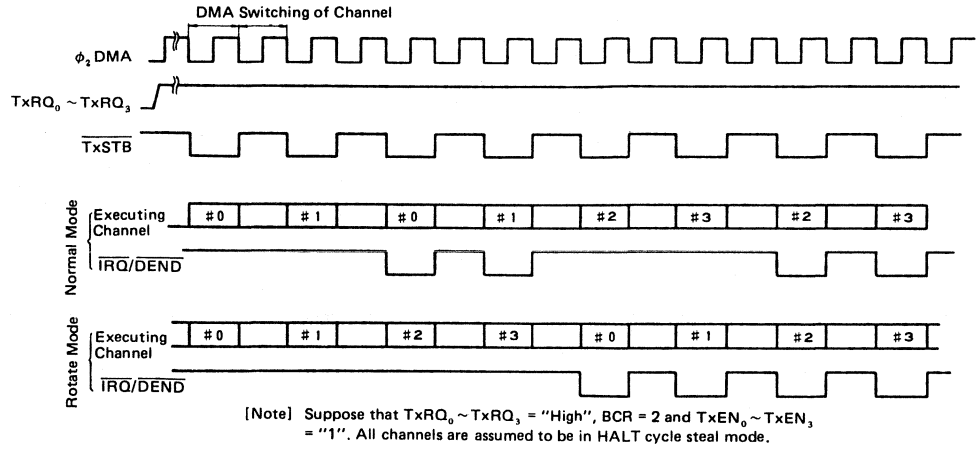


Figure 21 Example of Operation in the Rotate Mode

Next, Fig. 22 shows an example of the difference between the operation in the rotate mode and that in the normal mode. In this example, TxRQ of all channels is always at "High" level. Moreover, BCR=2 and TxEN=1 are assumed. As a transfer mode, HALT cycle steal mode is used.



[Note] Suppose that $TxRQ_0 \sim TxRQ_3$ = "High", BCR = 2 and $TxEN_0 \sim TxEN_3$ = "1". All channels are assumed to be in HALT cycle steal mode.

Figure 22 Difference between the operation in the rotate mode and that in the normal mode

The reason why the order of priority is not #0 → #0 → #1 → #1 → ––– in the normal mode is that during DMA transfer operation, TxRQ of an executing channel is prohibited from being accepted.

DMA Operation Timing with priority control

When more than 2 channels perform DMA transfer in parallel, the abovementioned priority-order-determining-circuit is used to determine the priority. The channel with lower priority waits until the channel with higher priority completes the transfer. Then it gets into DMA transfer operation. In this case, The following combinations of transfer modes are conceivable.

(1) From HALT mode to HALT mode (Fig. 23)
(2) From TSC mode to TSC mode (Fig. 24)
(3) From HALT mode to TSC mode
(4) From TSC mode to HALT mode } (Fig. 25)

In changing from HALT mode to HALT mode, only one dead cycle is intervened. That is, even in the cycle steal mode, DMA transfer of the next channel is performed without returning the bus control to the MPU ($\overline{DRQH}$ remains "Low").

In changing from TSC mode to TSC mode, DMA transfer of the next channel is performed, after returning the bus control to MPU for one cycle.

In the case of HALT → HALT, it doesn't return the bus control to MPU in order not to increase the response time of DMA transfer and dead cycles of the system.

On the other hand, in the case of TSC → TSC mode, same mean cannot be applicable because MPU clock cannot remain stopped for a long time as in the case of HALT mode.

Both in the case of HALT → TSC mode and in the case of TSC → HALT mode, DMA operation timing is based on the same idea as the above two kinds of mode change. (In detail, see Fig. 25).

The timing in the case where the next byte is transfered without changing the channel is shown in Fig. 26. This is the case of HALT → HALT mode. In this case, the bus control returns to MPU, before the next byte is transfered. In the case of TSC → TSC mode, its timing is almost the same as than in Fig. 24, that is, after 1-byte transfer has completed, MPU executes the Instruction Cycle for one clock and then DMAC executes 1-byte transfer again.

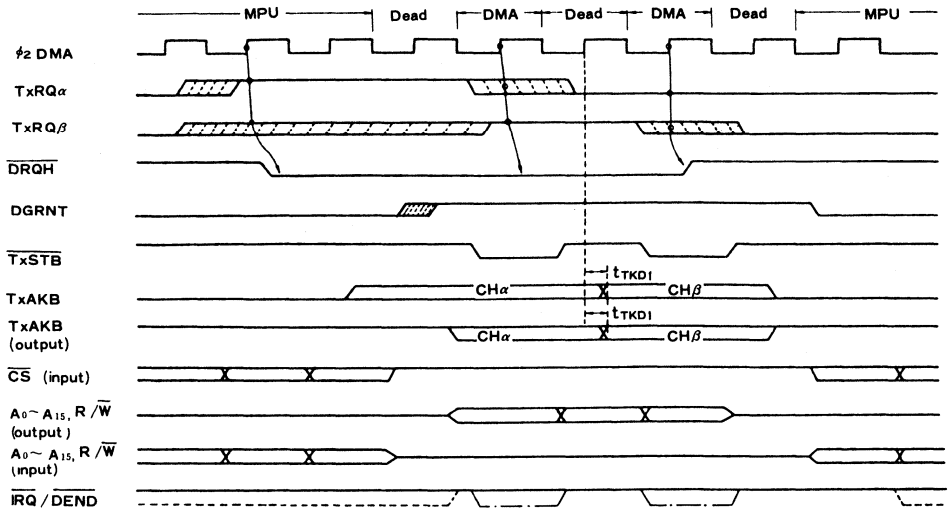


Figure 23 Channel Change (HALT Mode → HALT Mode)

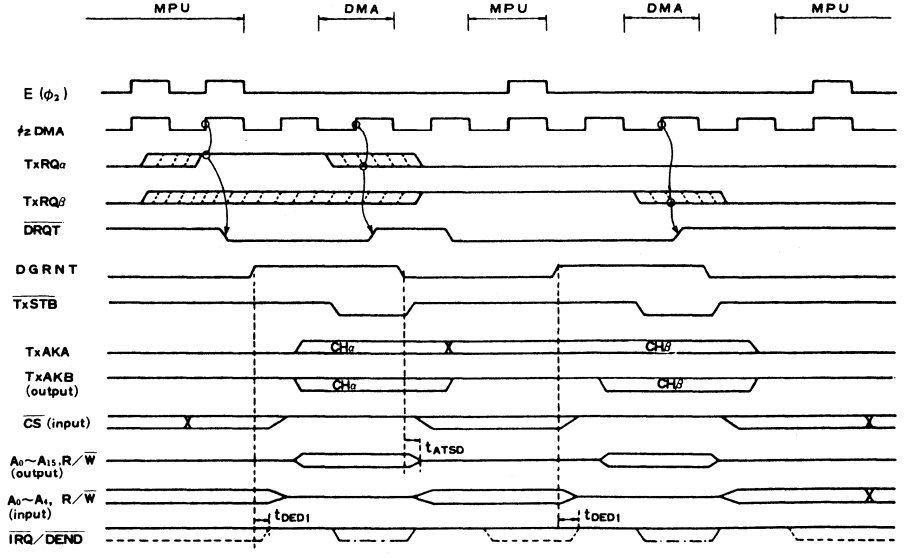


Figure 24 Channel Change (TSC Mode → TSC Mode)

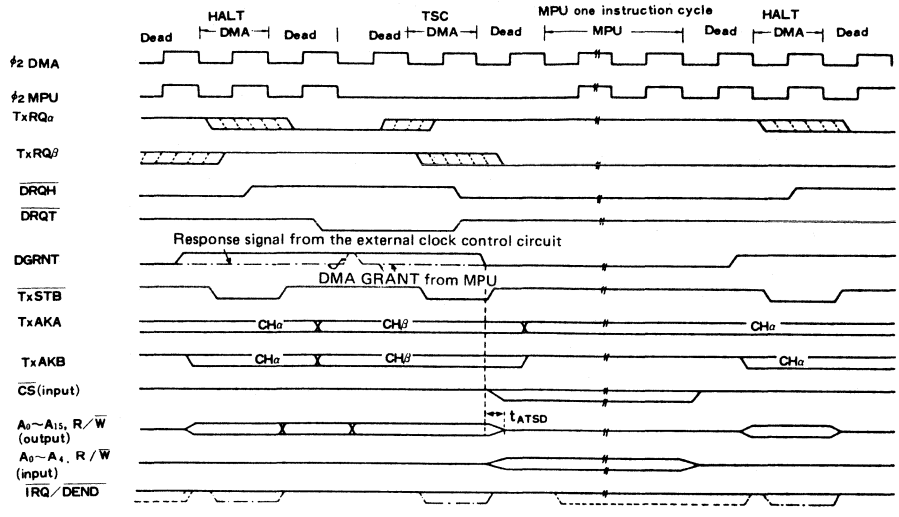
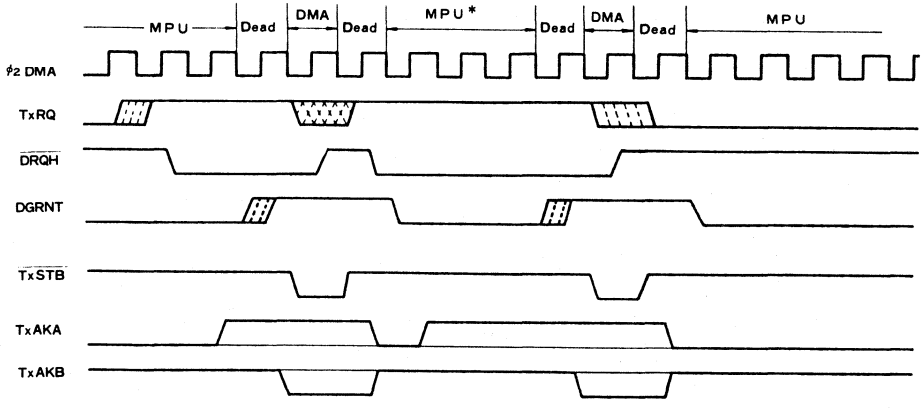


Figure 25 Channel Change (HALT Mode → TSC Mode → HALT Mode)



* Executing Period of One Instruction

Figure 26 Successive 2-byte Transfer of One Channel (HALT Cycle Steal Mode)
HALT → HALT (by one channel)

- **Status Flag**

DMAC has BUSY Flag, DEND Flag and ZERO Flag on each channel. The former two of these flags can be read out by MPU, but ZERO Flag cannot be read out. Set and reset timing of each flag are shown in Fig. 27.

BUSY/READY Flag

This flag is set to "1" when it accepts the first-byte TxRQ of its corresponding channel. After 1-block transfer has completed and BCR becomes "0", it is reset to "0". Therefore, while this flag is "1", that is, its corresponding channel is being used, the next block transfer cannot be performed.

Also this flag is cleared when corresponding TxRQ Enable Bit in the PCR becomes "0".

DEND Flag

This is the interrupt flag to indicate the end of DMA transfer of its corresponding channel. After 1-block transfer has completed and BCR becomes "0", this flag is set to "1". This flag is reset to "0" immediately after the Channel Control Register having this flag is read out.

ZERO Flag

This is the internal flag to indicate whether the data stored in the BCR is "0" or not (It cannot be read out).

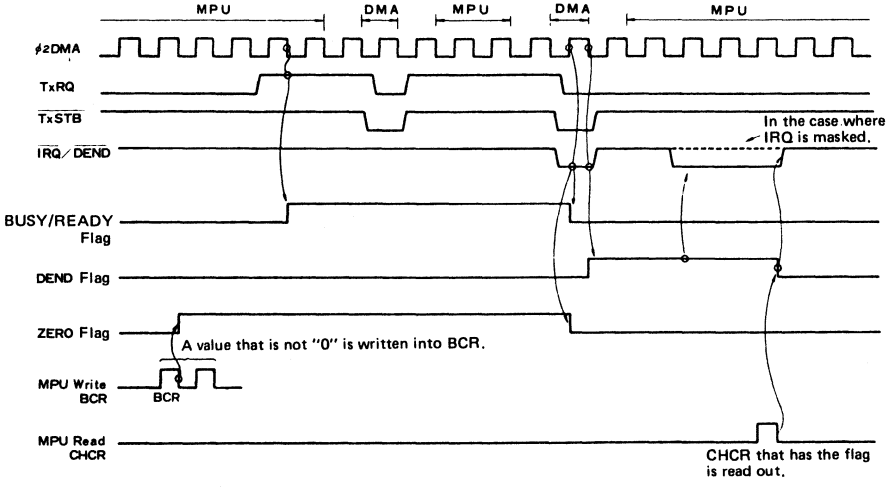


Figure 27 Timing of Status Flag (Suppose that BCR is 2 in the initial state)

When BCR is "0", ZERO Flag is "0". When BCR is not "0", it is "1".

In the reset state, this flag is "0". If data that is not "0" is written into BCR, this flag is set to "1". When BCR becomes "0" after 1-block data transfer has completed, or MPU writes "0" into BCR, this flag is reset to "0".

The function of ZERO Flag is to prohibit accepting TxRQ of its corresponding channel while this flag is "0" (that is, BCR is "0") (See Fig. 18). While ZERO Flag is "0", TxRQ is not accepted even if TxEN is "1". This function avoids an false operation even if "High" input is provided to TxRQ before the initialization of the register.

When $\overline{RES}$ pin goes to "Low", this flag becomes "0", but the number in BCR is not reset to "0". Therefore, the state of this flag and BCR are not the same. In this case new data should be written into BCR (Then ZERO Flag becomes "1").

- **DMA End Control**

Function of $\overline{IRQ}$/$\overline{DEND}$ Pin

DMAC has $\overline{IRQ}$ output and $\overline{DEND}$ output to perform DMA End Control. These are multiplexed outputs to $\overline{IRQ}$/$\overline{DEND}$ pin.

The function of $\overline{DEND}$ output is to inform I/O controller of the end of 1-block transfer. After 1-block transfer has been completed and BCR becomes "0", $\overline{DEND}$ output provides "Low" pulse whose cycle is one clock, being synchronous with the final 1-byte data transfer. 4 channels have only one $\overline{DEND}$ output in common, so each channel determines whether $\overline{DEND}$ output is its own output or not, combining with TxAK signal. When TxAK of the channel is "High" and $\overline{DEND}$ is "Low", it shows that the cycle is the last one of DMA (See Fig. 29 and 30).

The function of $\overline{IRQ}$ output is to inform MPU of the end of 1-block transfer by interrupting it. As shown in Fig. 28, $\overline{IRQ}$ output is logical AND-OR of the interrupt flag (DEND Flag) and IRQ Enable bit of each channel.

$\overline{IRQ}$ and $\overline{DEND}$ outputs are multiplexed. $\overline{IRQ}$/$\overline{DEND}$ pin is used as $\overline{DEND}$ output during DMAC cycle and $\overline{IRQ}$ output during MPU cycle. Moreover, DGRNT signal separates $\overline{DEND}$ and $\overline{IRQ}$ by its "High" or "Low". In detail, see Fig. 29 and Fig. 30.

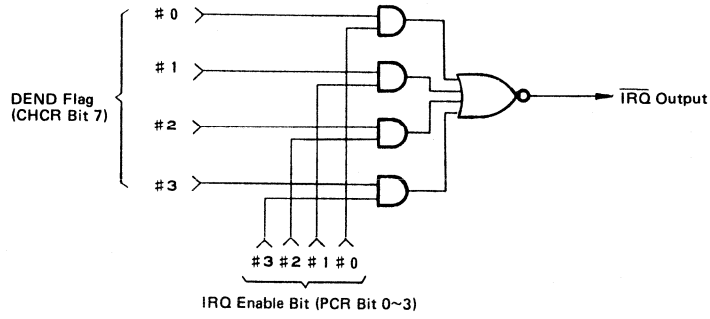


Figure 28 Logic of $\overline{IRQ}$ Output

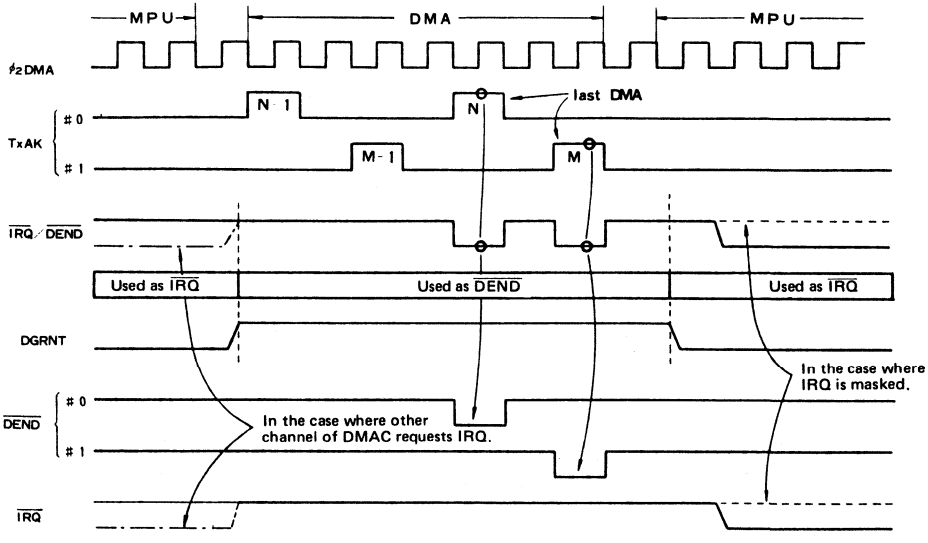


Figure 29 Timing of $\overline{IRQ}/\overline{DEND}$ Output

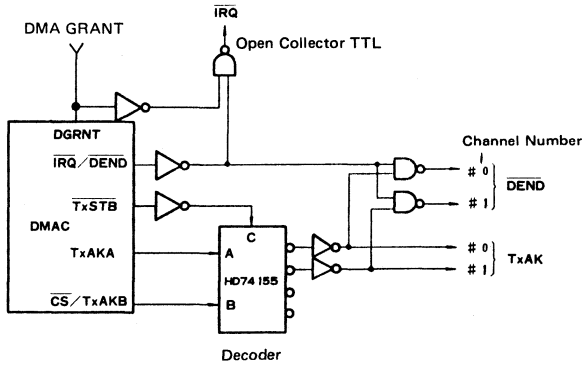


Figure 30 How to Use $\overline{IRQ}/\overline{DEND}$ Output Signal

Unusual DMA End

Following section describes how to terminate or change normal sequence of DMA transfer.

(1) When "0" is written into BCR
When "0" is written into BCR before it becomes "0", subsequent TxRQ are not accepted and this causes the termination of the DMA transfer since the internal ZERO Flag is reset to "0". In this case, note that $\overline{DEND}$ pulse is not provided.

(2) When "1" is written into BCR
When "1", instead of "0", is written into BCR, only the next TxRQ is accepted and 1-byte DMA transfer is performed. In this case, $\overline{DEND}$ pulse is provided, being synchronous with the last transfer.

(3) When another value is written into ADR & BCR during the transfer
When the data in ADR & BCR are changed during the transfer, the following transfer is performed according to the change of the data.

(4) When "0" is written into TxRQ Enable bit
When TxEN is reset to "0" during the transfer, this causes TxRQ comes not to be accepted and the transfer halts. But the state is different from that in the case (1), the number in BCR remains unchanged. Therefore, when TxEN is set to "1" again, the transfer is performed again.

(5) When $\overline{RES}$ pin is set to "Low"
When $\overline{RES}$ is provided during the transfer, the transfer stops.
Then all of the control registers and their internal flags are reset to "0". But the data in ADR & BCR are not reset.

(Supplement)

It is only in the cycle steal mode that DMAC registers such as BCR and ADR can be read or written during the transfer. In the burst mode, it is usually impossible (But special external circuits enable it).

- **Data Chain Function**

The data chain function of DMAC is to transfer the contents of ADR & BCR of Channel #3 to ADR & BCR of a specified channel automatically and renew the data of them after the channel has completed 1-block transfer.

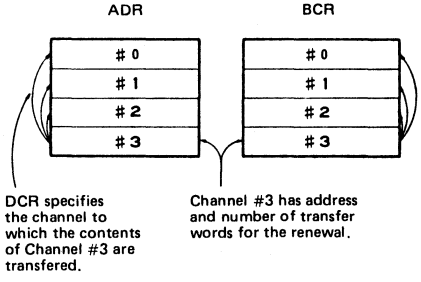


Figure 31 Data Chain Operation

Its detailed timing is shown in Fig. 32 and Fig. 33. As shown in these figures the contents of ADR & BCR of Channel #3 are transfered to the channel during the clock cycle next to the last one of 1-block transfer (which provides DEND pulse). Then $\overline{DRQH}$ or $\overline{DRQT}$ provides "Low" output for one more clock cycle than in the normal case. Therefore, MPU takes back the bus control again 1-clock later than in the normal case, that is, after the data renewal of the specified channel by the data chain from Channel #3.

In the TSC mode, the stretching period of clock ϕ_1 is longer than in the normal case.

The contents of ADR & BCR of Channel #3 remain unchanged as long as new data are not written by MPU, even if the data chain is executed.

As for $\overline{DEND}$ output, DEND Flag and BUSY Flag in the case of data chain execution, they function in the same way as in the normal case. They provide $\overline{DEND}$ pulse everytime 1-block transfer has completed, and then DEND Flag is set to "1". Therefore, in the case where more than 3-block data chain is needed, DEND Flag is used for the execution. Its sequence is shown in Fig. 34. First, DEND Flag="1" that shows the end of the first-block data chain is read out. Next, the data of ADR & BCR for the third-block data chain need to be written into Channel #3, in parallel with the execution of the second-block data chain. (This data chain is feasible only in the cycle steal mode.)

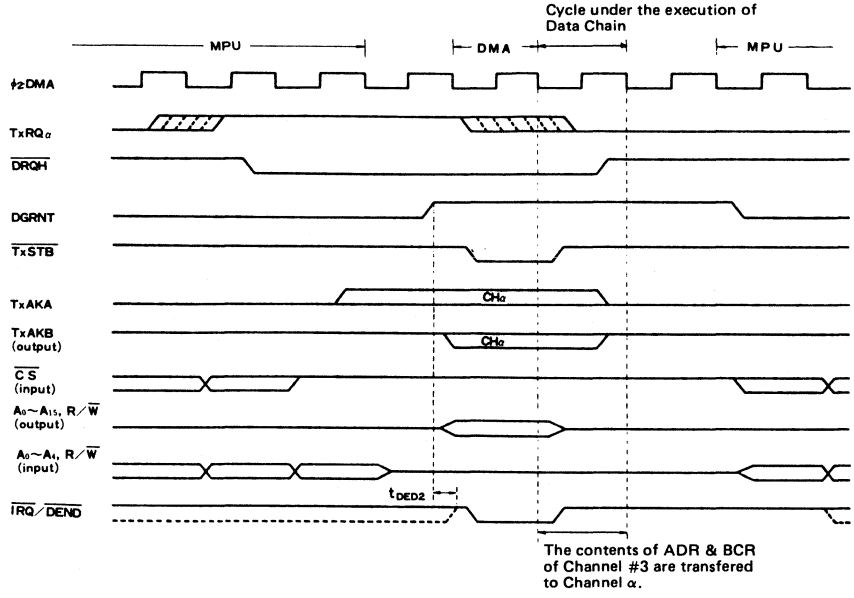


Figure 32 Data Chain Operation (HALT Mode)

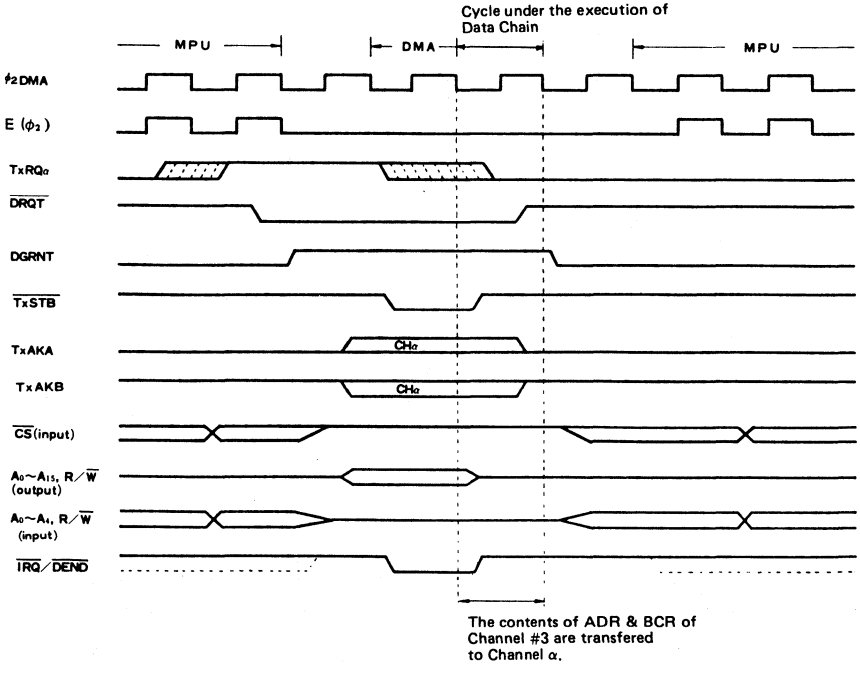


Figure 33 Data Chain Operation (TSC Mode)

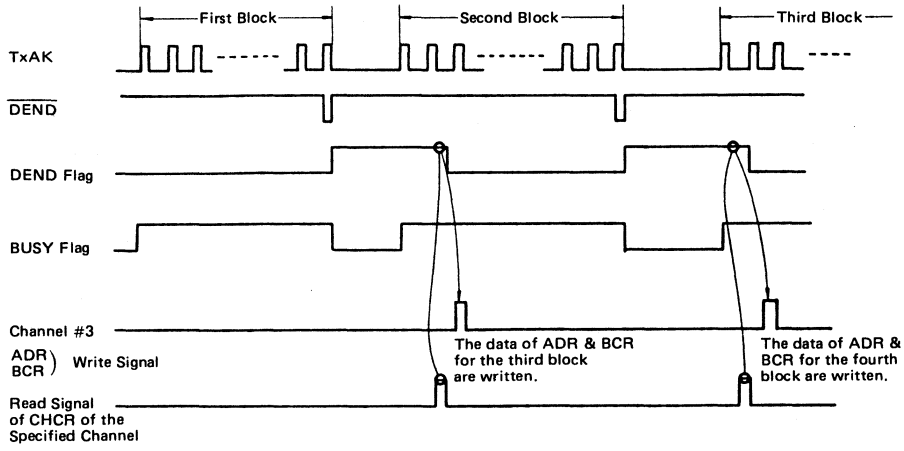


Figure 34 Sequence of More than 3-block Data Chain

■ DMAC PROGRAMMING

Preparation of a channel for a DMA transfer requires:

1) Load the starting address into the Address Register.
2) Load the number of bytes into the Byte Count Register.
3) Program the Channel Control Register for the transfer characteristics: direction (bit 0), mode (bits 1 and 2), and the address update (bit 3).

The channel is now configured. To enable the transfer request, set the appropriate enable bit (bits 0~3) of the Priority Control Register, as well as the Rotate Control bit.

If an interrupt on DMA End is desired, the enable bit (bits 0~3) of the Interrupt Control Register must be set.

If data chaining for the channel is necessary, it is programmed into the Data Chain Register and the appropriate data must be written into the Address and Byte Count Registers for channel #3.

Table 8 DMAC Programming Model

Register	Address (Hex)	Register Content							
		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Channel Control	1x*	DMA End Flag (DEND)	Busy/Ready Flag	Not Used	Not Used	Address Up/Down	TSC/ Halt	Burst/ Steal	Read/Write (R/W)
Priority Control	14	Rotate Control	Not Used	Not Used	Not Used	TxRQ Enable #3 (TxEN3)	TxRQ Enable #2 (TxEN2)	TxRQ Enable #1 (TxEN1)	TxRQ Enable #0 (TxEN0)
Interrupt Control	15	IRQ Flag	Not Used	Not Used	Not Used	IRQ Enable #3 (IE3)	IRQ Enable #2 (IE2)	IRQ Enable #1 (IE1)	IRQ Enable #0 (IE0)
Data Chain	16	Not Used	Not Used	Not Used	Not Used	Two/Four Channel Select (2/4)	Data Chain Channel Select B	Data Chain Channel Select A	Data Chain Enable

* The x represents the binary equivalent of the channel desired.

A comparison of the response times and maximum transfer rates is shown in Table 9. The data are shown for a system clock rate of 1 MHz.

The two 8-bit bytes that form the registers in Table 10 are placed in consecutive memory locations, making it very easy to use the MPU index register in programming them.

Fig. 38 shows an example of its minimum structure (1 channel, HALT mode, combination with FDC). Fig. 39 shows an example of its maximum structure. (but only one DMAC is used.)

Table 9 Maximum Transfer Speed & Response Time of the DMAC when $t_{cyc\phi}$ equals 1 μsec.

Mode		Maximum Transfer Speed (μsec/byte)	Response Time (μsec)	
			maximum	minimum
HALT Cycle Steal		(executing time of one instruction) + 3	(executing time of one instruction) $+3.5 - t_{TQH1}$	$3.5 + t_{TQS1}$
HALT Burst	first byte	1		
	since second byte		$2 - t_{TQH2}$	$1 + t_{TQS2}$
TSC Cycle Steal		4	$3.5 - t_{TQH1}$	$2.5 + t_{TQH1}$

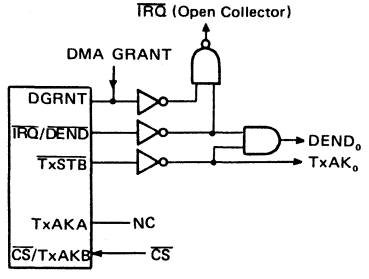


Figure 35 One Channel

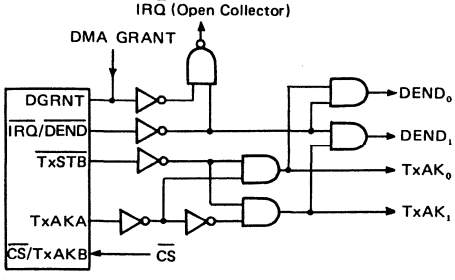


Figure 36 Two Channel

Table 10 Address and Byte Count Registers

Register	Channel	Address (Hex)
Address High	0	0
Address Low	0	1
Byte Count High	0	2
Byte Count Low	0	3
Address High	1	4
Address Low	1	5
Byte Count High	1	6
Byte Count Low	1	7
Address High	2	8
Address Low	2	9
Byte Count High	2	A
Byte Count Low	2	B
Address High	3	C
Address Low	3	D
Byte Count High	3	E
Byte Count Low	3	F

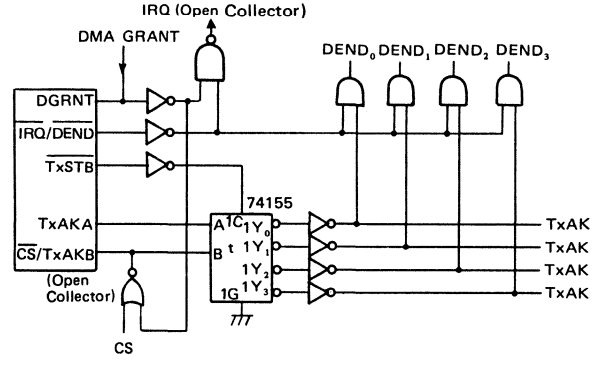


Figure 37 Four-Channel

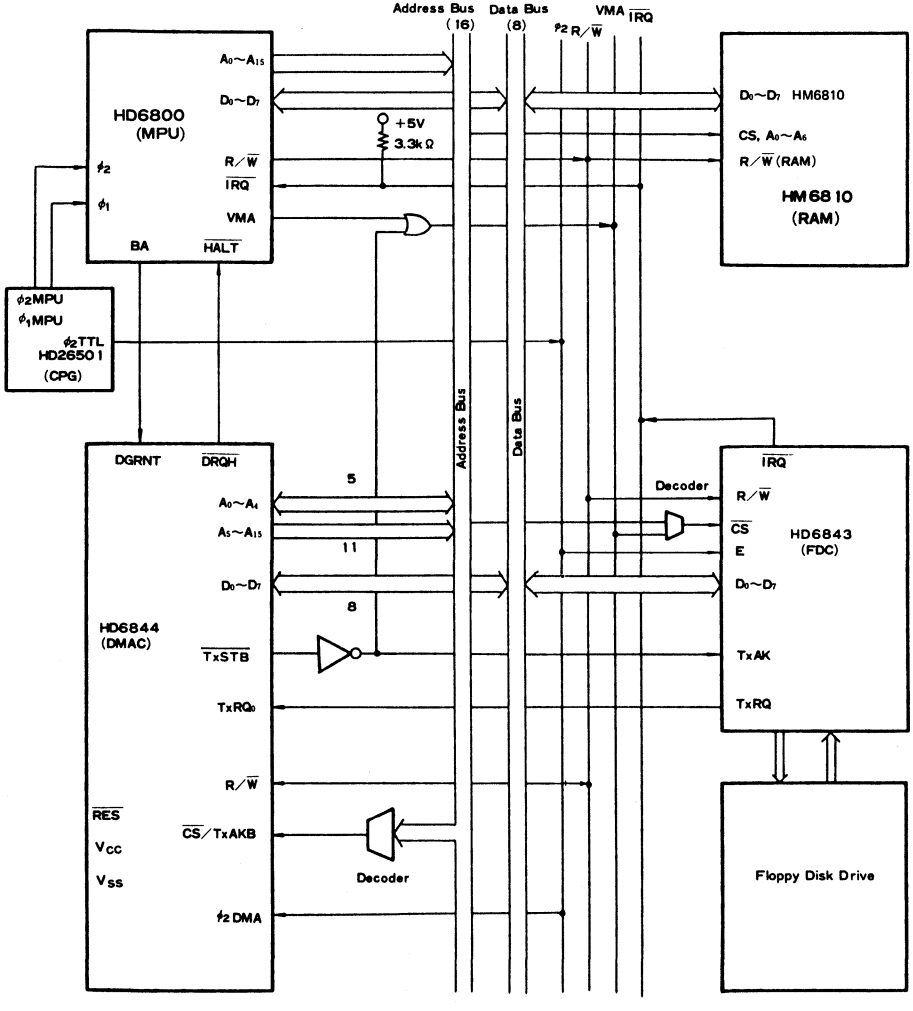


Figure 38 Example of DMA System Structure (1) (minimum)

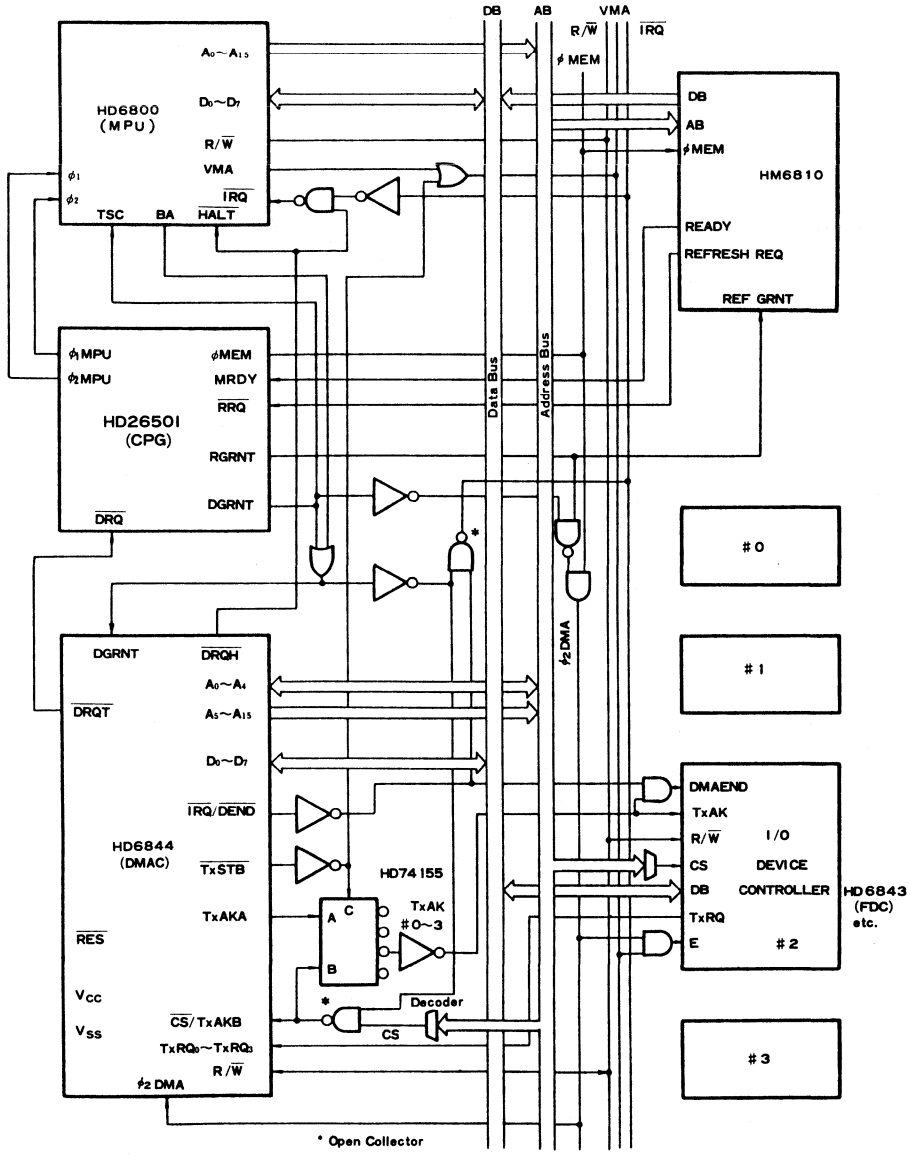
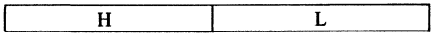


Figure 39 Example of DMA System Structure (2) (maximum)

■ APPENDIX

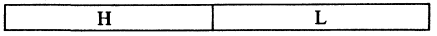
Contents of the DMAC Registers

(1) ADR0 ~ ADR3 (Address Register) (1 ADR on each channel)



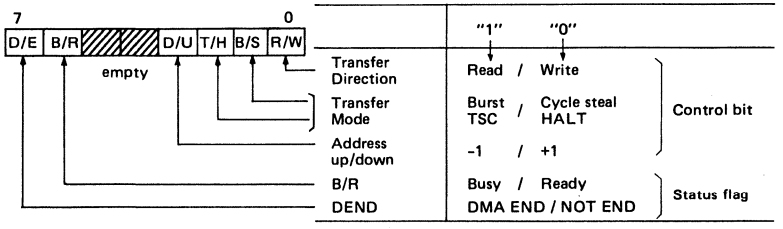
16 bit x 4

(2) BCR0 ~ BCR3 (Byte Count Register) (1 BCR on each channel)

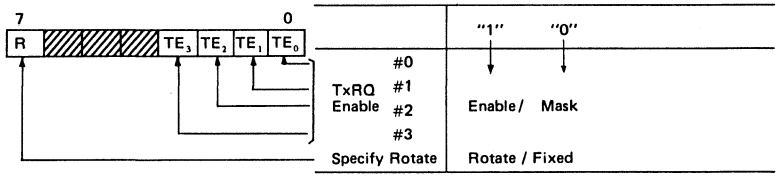


16 bit x 4

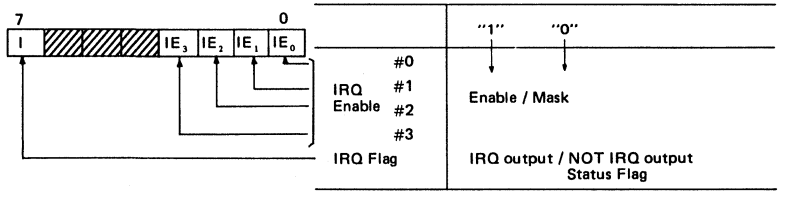
(3) CHCR0 ~ CHCR3 (Channel Control Register) (1 CHCR on each channel) (6 bit x 4)



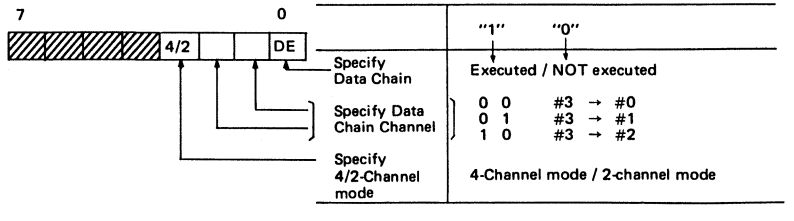
(4) PCR (Priority Control Register) (5 bit x 1)



(5) ICR (Interrupt Control Register) (5 bit x 1)



(6) DCR (Data Chain Control Register) (4 bit x1)



HD6345
CRTC-II (CRT Controller)

—PRELIMINARY—

The HD6345 CRTC-II provides an interface between MPU and a raster scan CRT display. It is upward-compatible with the NMOS CRTC HD6845S in pin and software. A power dissipation is lowered by adopting the CMOS process.

The HD6345 offers a variety of functions under MPU control, such as programmable timing signal outputs for CRT monitor and display screen control operation. It can be widely applied to the various types of CRT display systems.

■ FEATURES

FLEXIBLE SCREEN FORMAT
- Programmable numbers of characters per screen and rasters per character row
- Programmable horizontal/vertical sync signals and display timing signals
- Up to 16k words refresh memory (14-bit) addressable
- Programmable raster scanning modes: Non-Interlace, Interlace sync, or Interlace sync and video modes
- Up to 256 character rows per field
- High-speed display operation at 4.5 MHz character clock
- Double-size vertical display by raster interpolation

VERSATILE DISPLAY FUNCTIONS
- Screen split (max.4 screens configurable, horizontally)
- Paging and scrolling for each screen
- Smooth scrolling
- Two cursors with programmable width
- Programmable refresh memory width

FACILITATED SYSTEM CONFIGURATION
- 68 system bus interface
- Three-state control of memory address and raster address
- External synchronization in Master-slave or TV sync modes
- Interrupt request by vertical blanking or light pen strobe detection
- Programmable timing signal for dual-port RAM in DPRAM mode

PIN AND SOFTWARE UPWARD-COMPATIBLE WITH HD6845S

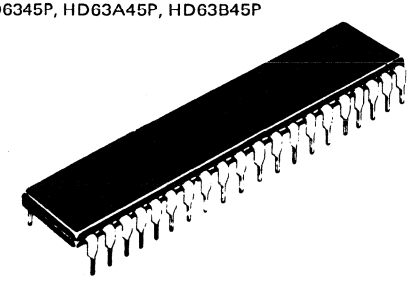
SINGLE +5 V POWER SUPPLY

CMOS PROCESS

■ TYPE OF PRODUCTS

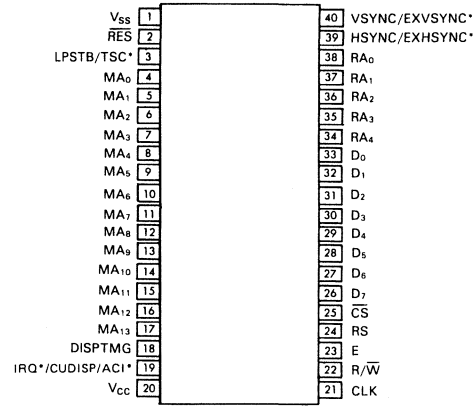
Type No.	Bus Timing	CRT Display Timing
HD6345	1.0 MHz	4.5 MHz Max.
HD63A45	1.5 MHz	
HD63B45	2.0 MHz	

HD6345P, HD63A45P, HD63B45P



(DP-40)

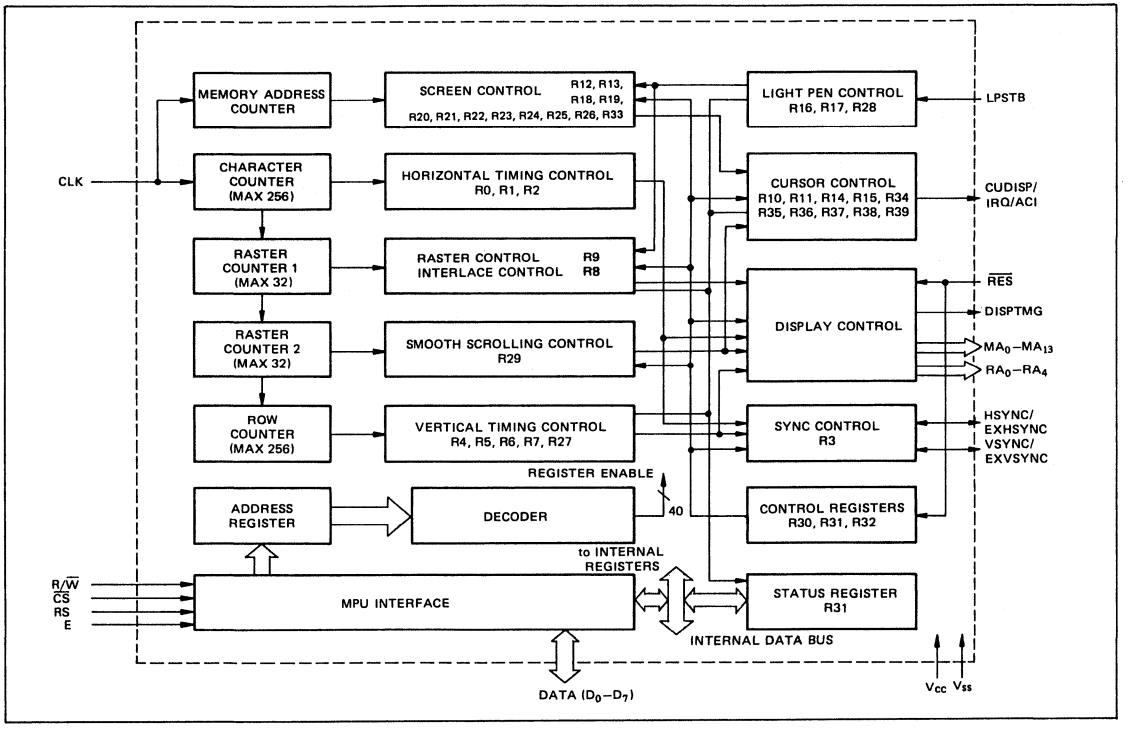
■ PIN ARRANGEMENT



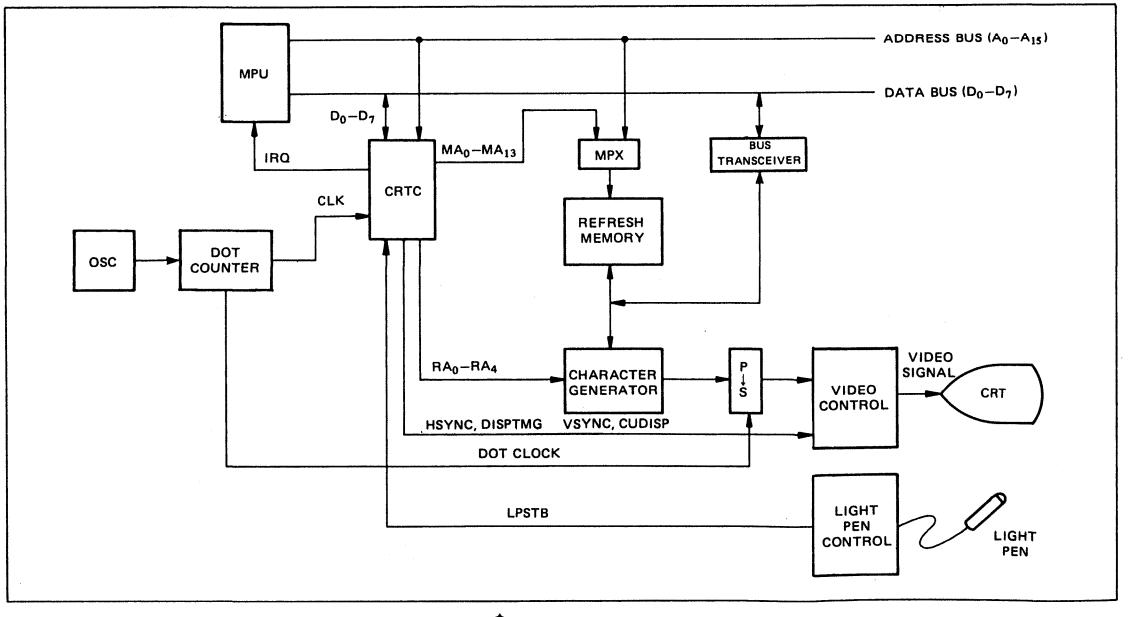
(Top View)

* Providing additional functions to the HD6845S.

■ INTERNAL BLOCK DIAGRAM



■ SYSTEM BLOCK DIAGRAM



■ FUNCTION TABLE

Item	Descriptions	Remarks
Programmable Screen Format	Horizontal scanning cycle Vertical scanning cycle (by row) Vertical scanning cycle (Adjust) Number of displayed chars. / row Number of char. rows / screen Number of rasters / char. row Horizontal display position Vertical display position Vertical sync position (Adjust) HSYNC pulse width VSYNC pulse width DISPTMG skew	Programmable by char. clock time Programmable by char. row time Programmable by raster time Enabled by programming sync signal output timings Programmable by raster time 1 or 2 character skew
Screen Split	4 split-screens start positions programmable	Discretely programmable (Unit: row) 2/3/4 screens format selectable
Cursor Control	Cursor display position Cursor height Cursor width Cursor blink Simultaneous output of 2 cursors (Only 1 available in DPRAM mode) Cursor display mode CUDISP skew	Two 14-bit Cursor registers 1 or 2 cursors displayed Display start/end rasters programmable within a row Programmable by char. clock time 1/16 or 1/32 field rate selectable Discretely programmable OR/EOR mode selectable 1 or 2 character skew
Raster Scanning Mode	Non-Interlace mode Interlace sync mode Interlace sync and video mode	Either one of three modes selectable
Memory Format	Memory width set	Memory width programmable wider than display width (Unit: char.)
Smooth Scrolling	Display start raster address set Target screen set	Programmable by char. clock time Any screen selectable
Raster Interpolation	Double-size vertical display Vertical scanning cycle doubled	Same raster address supplied twice
External Synchronization	Synchronization with external sync signals	Superimposed display enhabled on other CRT or TV screens
Interrupt Request	Interrupt request signal caused by vertical blanking period or light pen detection (Disabled in DPRAM mode)	Interrupt request mode programmable
Light Pen	14-bit Light pen register Light pen raster register	Light pen raster address detected
Refresh Memory Addressing	14-bit refresh memory address output Four 14-bit screen start regs. (Display start address programmable for each screen)	Up to 16k words refresh memory accessible Paging and scrolling enabled for each screen
Three-State Control	Three-state control on MA and RA	Controlled by TSC pin input
Programmable Timing Output	Programmable timing signal supplied from access inhibit pin	In DPRAM mode

■ PIN FUNCTION

Pin No.	Symbol	Pin Name	Input/ Output	Functions
1	V_{SS}	V_{SS}	—	Ground (GND) pin
2	$\overline{RES}$	RESET	Input	Performs external reset on CRTC $\overline{RES}$ assertion causes CRTC: (1) Clear all the internal counters (2) Set all the output signals at "L" (D_0–D_7 are excluded) (3) Clear registers R30 (Control 1), R31 (Control 2 / Status), and R32 (Control 3) (Other registers are not affected at all) $\overline{RES}$ is valid only while LPSTB is "L"
3*	LPSTB	LIGHT PEN STROBE	Input	Informs light pen strobe pulse detection
	TSC	THREE STATE CONTROL	Input	Performs three-state control on memory and raster addresses
4–17	MA_0–MA_{13}	MEMORY ADDRESS 0–13	Output	Supplies memory address for periodical memory refresh
18	DISPTMG	DISPLAY TIMING	Output	Indicates a screen display period
19*	CUDISP	CURSOR DISPLAY	Output	Displays cursor on a screen Enabled during DISPTMG is "H"
	ACI	ACCESS INHIBIT	Output	Supplies DPRAM access inhibit timing (programmable)
	IRQ	INTERRUPT REQUEST	Output	Indicates interrupt request to MPU Enabled during DISPTMG is "L"
20	V_{CC}	V_{CC}	—	Power supply (+5V) pin
21	CLK	CHARACTER CLOCK	Input	Receives character clock timing
22	R/$\overline{W}$	READ/WRITE	Input	Controls data transfer direction between MPU and CRTC
23	E	ENABLE	Input	Enables register read/write strobe signals from MPU
24	RS	REGISTER SELECT	Input	Selects either of address register or other registers Address reg. selected when at "L", and others at "H" Normally, requested to connect to "A_0" of MPU address bus
25	$\overline{CS}$	CHIP SELECT	Input	Performs addressing on CRTC MPU read/write upon CRTC registers enabled when $\overline{CS}$ is "L"
26–33	D_0–D_7	DATA BUS 0–7	Input/ Output	Bidirectional bus for data transfer between MPU and CRTC
34–38	RA_0–RA_4	RASTER ADDRESS 0–4	Output	Supplies raster address for selecting raster on character generator
39*	HSYNC	HORIZONTAL SYNC	Output	Supplies horizontal sync signal
	EXHSYNC	EXTERNAL HORIZONTAL SYNC	Input	Receives external horizontal sync signal
40*	VSYNC	VERTICAL SYNC	Output	Supplies vertical sync signal
	EXVSYNC	EXTERNAL VERTICAL SYNC	Input	Receives external vertical sync signal

Note: *-marked pin function is alterable according to the register setting.

■ INTERNAL REGISTER ASSIGNMENT

$\overline{CS}$	RS	Register Address 5 4 3 2 1 0	Reg. No.	Register Name	Program Unit	Symbol	R/W	Data Bit 7	6	5	4	3	2	1	0
1	x	x x x x x x	–	–	–	–	–								
0	0	x x x x x x	AR	ADDRESS REGISTER	–	–	W								
0	1	0 0 0 0 0 0	R0	HORIZONTAL TOTAL CHARACTERS	character	Nht	W								
0	1	0 0 0 0 0 1	R1	HORIZONTAL DISPLAYED CHARACTERS	character	Nhd	W								
0	1	0 0 0 0 1 0	R2	HORIZONTAL SYNC POSITION	character	Nhsp	W								
0	1	0 0 0 0 1 1	R3	SYNC WIDTH	*	Nvsw, Nhsw	W	Wv_3	Wv_2	Wv_1	Wv_0	Wh_3	Wh_2	Wh_1	Wh_0
0	1	0 0 0 1 0 0	R4	VERTICAL TOTAL ROWS	row	Nvt	W								
0	1	0 0 0 1 0 1	R5	VERTICAL TOTAL ADJUST	raster	Nadj	W								
0	1	0 0 0 1 1 0	R6	VERTICAL DISPLAYED ROWS	row	Nvd	W								
0	1	0 0 0 1 1 1	R7	VERTICAL SYNC POSITION	row	Nvsp	W								
0	1	0 0 1 0 0 0	R8	INTERLACE MODE AND SKEW	–	–	W	C_1	C_0	D_1	D_0			V	S
0	1	0 0 1 0 0 1	R9	MAX RASTER ADDRESS	raster	Nr	W								
0	1	0 0 1 0 1 0	R10	CURSOR 1 START	raster	Ncs_1	W		B_1	P_1					
0	1	0 0 1 0 1 1	R11	CURSOR 1 END	raster	Nce_1	W								
0	1	0 0 1 1 0 0	R12	SCREEN 1 START ADDRESS (H)	memory address	–	R/W								
0	1	0 0 1 1 0 1	R13	SCREEN 1 START ADDRESS (L)	memory address	–	R/W								
0	1	0 0 1 1 1 0	R14	CURSOR 1 ADDRESS (H)	memory address	–	R/W								
0	1	0 0 1 1 1 1	R15	CURSOR 1 ADDRESS (L)	memory address	–	R/W								
0	1	0 1 0 0 0 0	R16	LIGHT PEN (H)	–	–	R								
0	1	0 1 0 0 0 1	R17	LIGHT PEN (L)	–	–	R								
0	1	0 1 0 0 1 0	R18	SCREEN 2 START POSITION	row	–	R/W								
0	1	0 1 0 0 1 1	R19	SCREEN 2 START ADDRESS (H)	memory address	–	R/W								
0	1	0 1 0 1 0 0	R20	SCREEN 2 START ADDRESS (L)	memory address	–	R/W								
0	1	0 1 0 1 0 1	R21	SCREEN 3 START POSITION	row	–	R/W								
0	1	0 1 0 1 1 0	R22	SCREEN 3 START ADDRESS (H)	memory address	–	R/W								
0	1	0 1 0 1 1 1	R23	SCREEN 3 START ADDRESS (L)	memory address	–	R/W								
0	1	0 1 1 0 0 0	R24	SCREEN 4 START POSITION	row	–	R/W								
0	1	0 1 1 0 0 1	R25	SCREEN 4 START ADDRESS (H)	memory address	–	R/W								
0	1	0 1 1 0 1 0	R26	SCREEN 4 START ADDRESS (L)	memory address	–	R/W								
0	1	0 1 1 0 1 1	R27	VERTICAL SYNC POSITION ADJUST	raster	Nvad	W								
0	1	0 1 1 1 0 0	R28	LIGHT PEN RASTER	–	–	R	DP							
0	1	0 1 1 1 0 1	R29	SMOOTH SCROLLING	raster	Nss	R/W								
0	1	0 1 1 1 1 0	R30	CONTROL 1	–	–	W	VE	VS	IB	IL	SY	TV	SP_1	SP_0
0	1	0 1 1 1 1 1	R31	CONTROL 2	–	–	W	SS_4	SS_3	SS_2	SS_1	RI			
				STATUS	–	–	R	AI					E	SB	SL
0	1	1 0 0 0 0 0	R32	CONTROL 3	–	–	W	CM	C_2	CW_1	CW_2	MW	TC	DR	
0	1	1 0 0 0 0 1	R33	MEMORY WIDTH OFFSET	character	Nof	R/W								
0	1	1 0 0 0 1 0	R34	CURSOR 2 START	raster	Ncs_2	W		B_2	P_2					
0	1	1 0 0 0 1 1	R35	CURSOR 2 END	raster	Nce_2	W								
0	1	1 0 0 1 0 0	R36	CURSOR 2 ADDRESS (H)	memory address	–	R/W								
0	1	1 0 0 1 0 1	R37	CURSOR 2 ADDRESS (L)	memory address	–	R/W								
0	1	1 0 0 1 1 0	R38	CURSOR 1 WIDTH	character	Ncw_1	R/W								
0	1	1 0 0 1 1 1	R39	CURSOR 2 WIDTH	character	Ncw_2	R/W								

Notes 1) *: Vertical : raster / Horizontal : character

2) ▒: "0" is to be set, since these bits may be used in the future.

■ FUNCTIONAL DESCRIPTION

PROCRAMMABLE SCREEN FORMAT

Figure 1 illustrates the screen format example, in Non-Interlace mode, when programming CRTC registers as listed in Table 1. Figure 2 shows the relation between memory address (MA_0–MA_{13}), raster address (RA_0–RA_4) and the location on the CRT screen.

The timing charts of CRT interface signals are shown in Figure 3, and those details are partially shown in Figure 4 and 5.

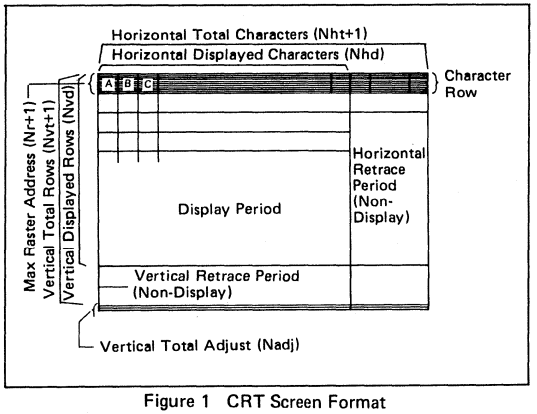
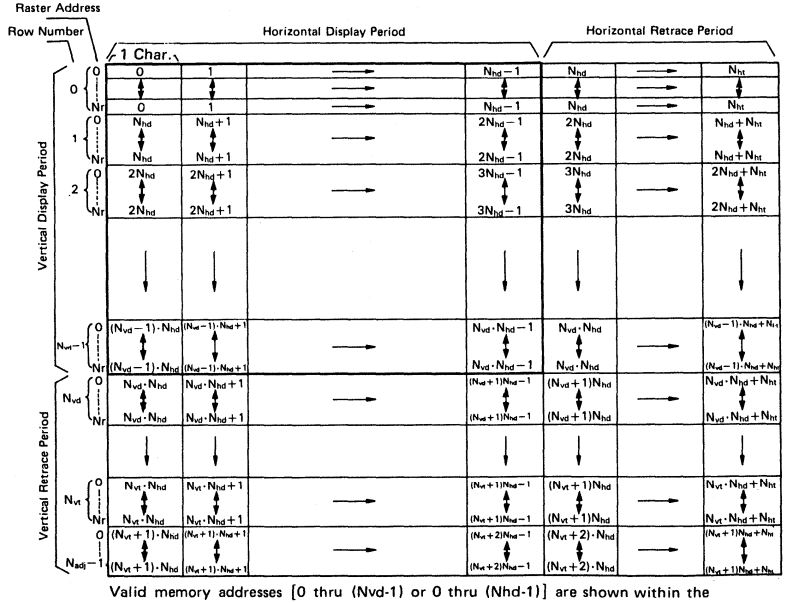


Figure 1 CRT Screen Format

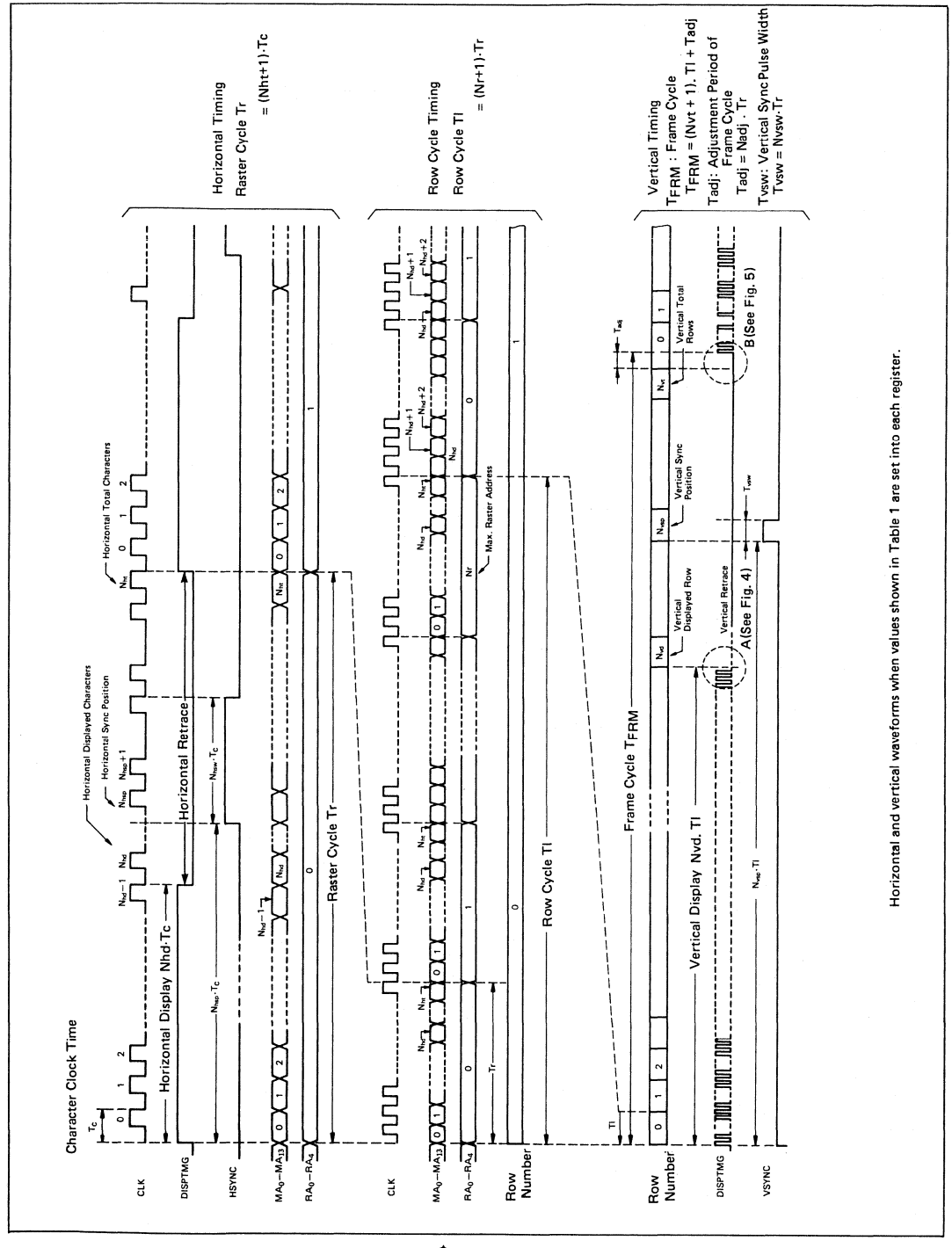
Table 1 Programmed Values in Each Register

Register No.	Register Name	Programmed Values
R0	HORIZONTAL TOTAL CHARACTERS	Nht
R1	HORIZONTAL DISPLAYED CHARACTERS	Nhd
R2	HORIZONTAL SYNC POSITION	Nhsp
R3	SYNC WIDTH	Nvsw, Nhsw
R4	VERTICAL TOTAL ROWS	Nvt
R5	VERTICAL TOTAL ADJUST	Nadj
R6	VERTICAL DISPLAYED ROWS	Nvd
R7	VERTICAL SYNC POSITION	Nvsp
R9	MAX. RASTER ADDRESS	Nr
R12	SCREEN 1 START ADDRESS (H)	0
R13	SCREEN 1 START ADDRESS (L)	0
R30	CONTROL 1	0
R31	CONTROL 2 / STATUS	0
R32	CONTROL 3	0

Note 1) Nhd < Nht, Nvd < Nvt
2) R30, R31, and R32 are cleared by a device reset.



Valid memory addresses [0 thru (Nvd-1) or 0 thru (Nhd-1)] are shown within the thick-line square. Memory addresses are provided even during horizontal and vertical retrace period. This is an example in the case where the programmed value of start address register is 0.



Horizontal and vertical waveforms when values shown in Table 1 are set into each register.

Figure 3 CRTC Timing Chart (Non-Interlace mode)

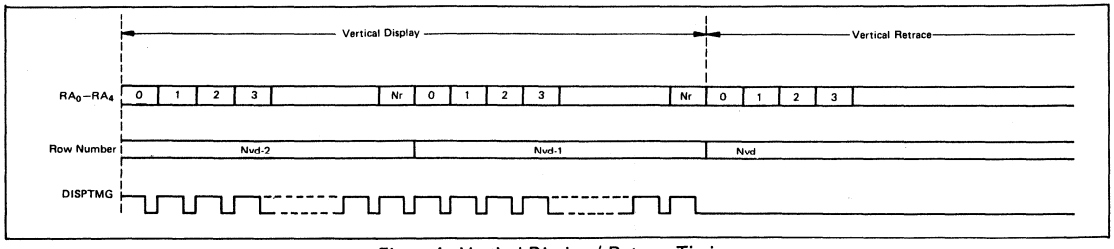


Figure 4 Vertical Display / Retrace Timing
(A detail drawing of Fig. 3 A)

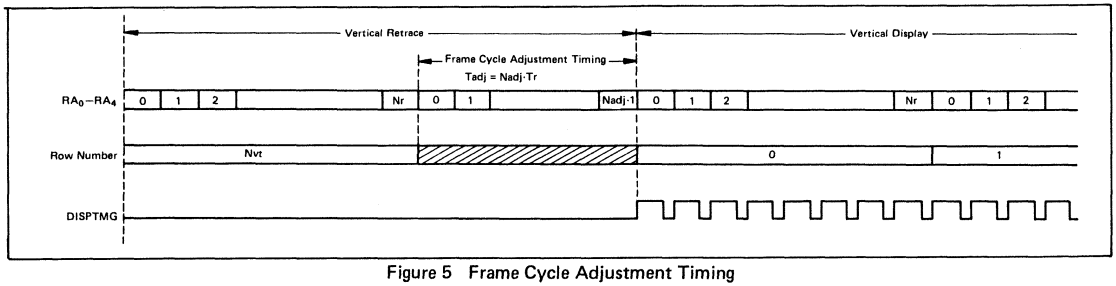


Figure 5 Frame Cycle Adjustment Timing
(A detail drawing of Fig.3 B)

SCREEN SPLIT

A display screen can be divided into up to four parts in the horizontal direction. Divided four parts are defined as split-screen 1, split-screen 2, split-screen 3, and split-screen 4. Split-screens are controlled by using bits SP_0 and SP_1 of the control 1 register (R30) and screen start position registers (R18, R21, R24).

Starting positions of each split-screen are determined in the number of character row. Split-screen 1 is the base screen, and always starts at row 0, while the other three split-screens start at any row except row 0. Paging or scrolling (by character) is performed in each split-screen independently.

The following is the example of screen split:

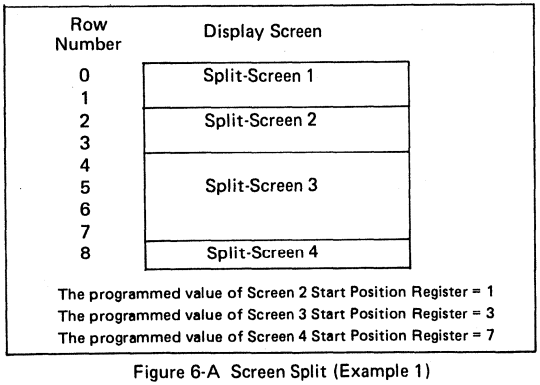


Figure 6-A Screen Split (Example 1)

When the same value are programmed into more than one screen start position registers, split-screens corresponding to these registers are not displayed.

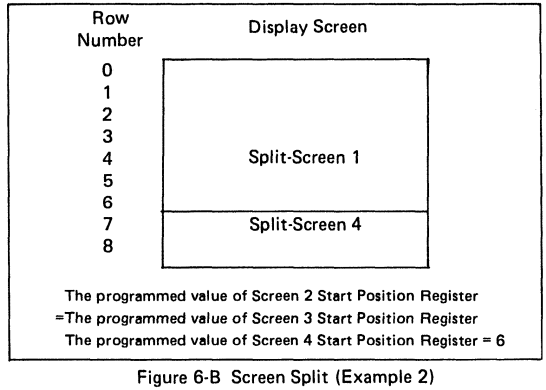


Figure 6-B Screen Split (Example 2)

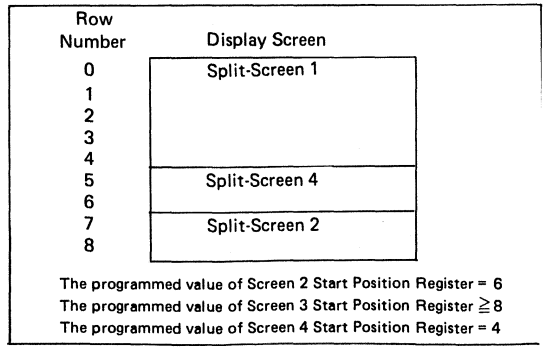


Figure 6-C Screen Split (Example 3)

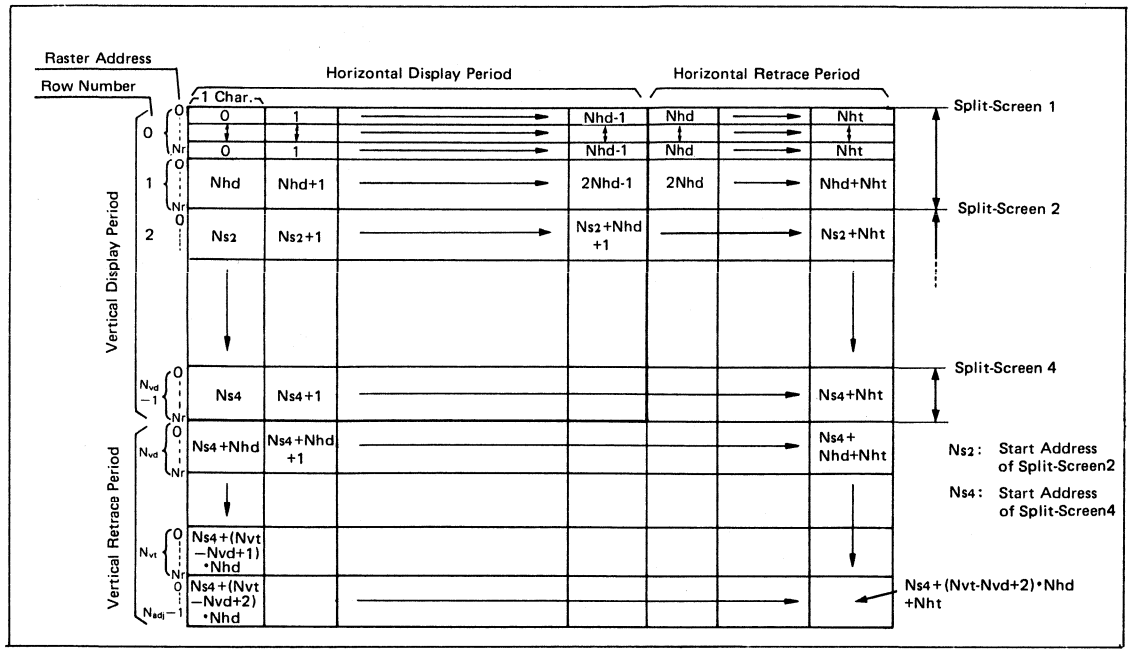


Figure 7 Memory Address and Raster Address in Split-Screens

CURSOR CONTROL

The HD6345 can display two separate cursors (cursor 1, cursor 2) simultaneously on the screen. These two cursors are controlled independently. The cursor 1 is always valid, while the cursor 2 becomes valid by setting the C_2 bit of the control 3 register (R32).

In the DPRAM mode, the cursor 2 cannot be displayed.

The HD6345 controls cursors as follows:

1) Starting Position
   Starting position is controlled by using the cursor 1 address registers (R14, R15), and the cursor 2 address registers (R36, R37).
2) Cursor Heights
   The heights of the cursor 1 and the cursor 2 can be specified independently in units of rasters by using the cursor start registers (R10, R34), and the cursor end registers (R11, R35).
3) Cursor Widths
   The widths of the cursor 1 and the cursor 2 can be specified independently in units of characters by using the cursor width registers (R38, R39), and bits CW_1 and CW_2 of the control 3 register (R32).
   If the cursor width extends over the following row, the cursor in the following row is not displayed.
4) Cursor Blink
   Cursor display, non-display, and blink rate can be controlled by using bits B_1 and P_1 of the cursor 1 start register (R10), and bits B_2 and P_2 of the cursor 2 start register (R34).
5) Cursor Display Mode
   When the cursor 1 and the cursor 2 are overlapped on the screen, cursor display mode in the overlapped area can be specified by the CM bit of the control 3 register (R32), as shown in Figure 9.

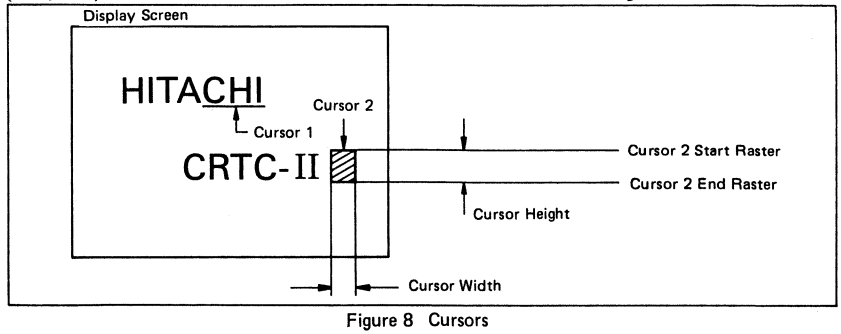


Figure 8 Cursors

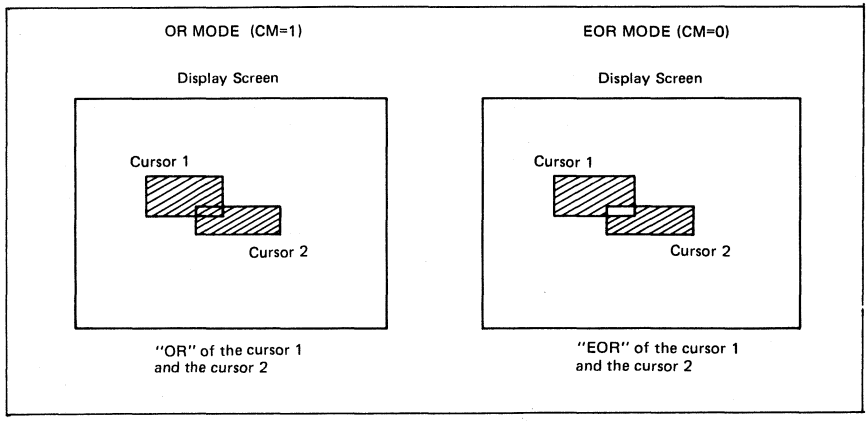


Figure 9 Cursor Display Mode

RASTER SCANNING MODE

The HD6345 performs a character display in three types of raster scanning modes: Non-Interlace mode, Interlace sync mode, and Interlace sync and video mode. The bits V and S of the interlace mode and skew register (R8) control these modes.

The period that the raster scans across a screen and returns to the top of the raster line is designated as 'one field'. In the Non-Interlace mode, one field configures a single frame (Figure 10). In the Interlace sync mode and the Interlace sync and video mode, a single frame period is shared between two alternating, even and odd, fields (Figure 10).

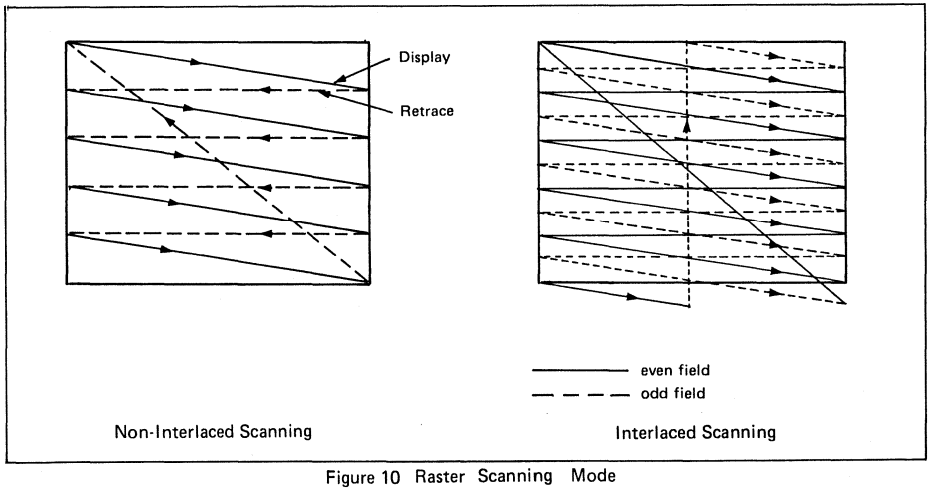


Figure 10 Raster Scanning Mode

In the Interlace sync mode, the scanning lines in the odd field are placed downward by 1/2 raster line space from those in the even field because of the difference in HSYNC/VSYNC phases between two alternating fields.

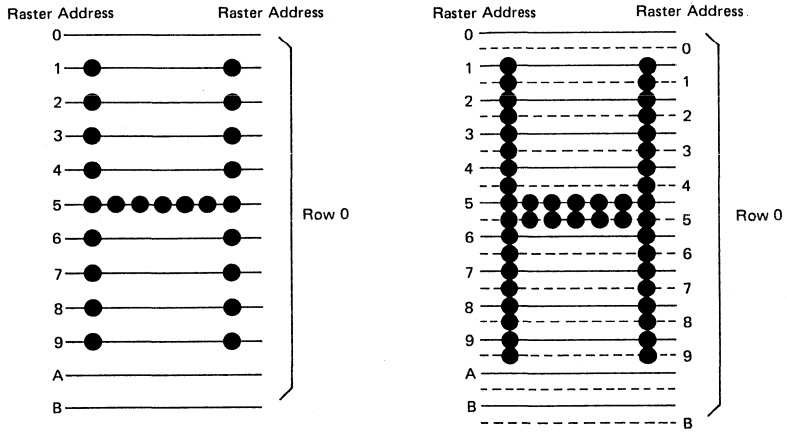
In the Interlace sync and video mode, the placement of the scanning lines is the same as in the Interlace sync mode. However, the alternating even and odd raster lines are displayed in the alternating even and odd fields. For a given number of rasters per character, this mode allows twice as many characters to be displayed in the vertical direction as the Non-Interlace and the Interlace sync modes. Note that the raster address is supplied in the different way according to the total number of rasters in a row, even or odd, as shown in Table 2.

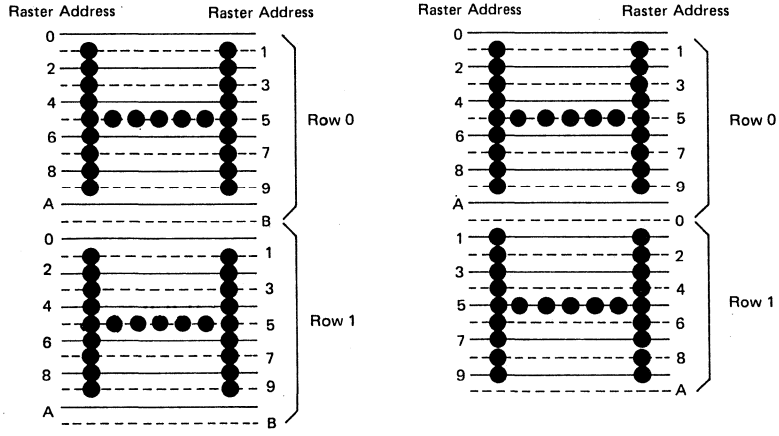
Table 2 Start Raster Address for Each Row (In Interlace sync and video mode)

Total Number of Rasters in a Row \ Field		Even Field	Odd Field
Even		Even address	Odd address
Odd	Even Char. Row*	Even address	Odd address
	Odd Char. Row*	Odd address	Even address

* The start row address is assumed to be "0" (even).

Figure 11 illustrates the raster scanning example in each mode.





Note: Refer to Table 2 for raster address output in the Interlace sync and video mode.

even field

odd field

Figure 11 Raster Scanning Example

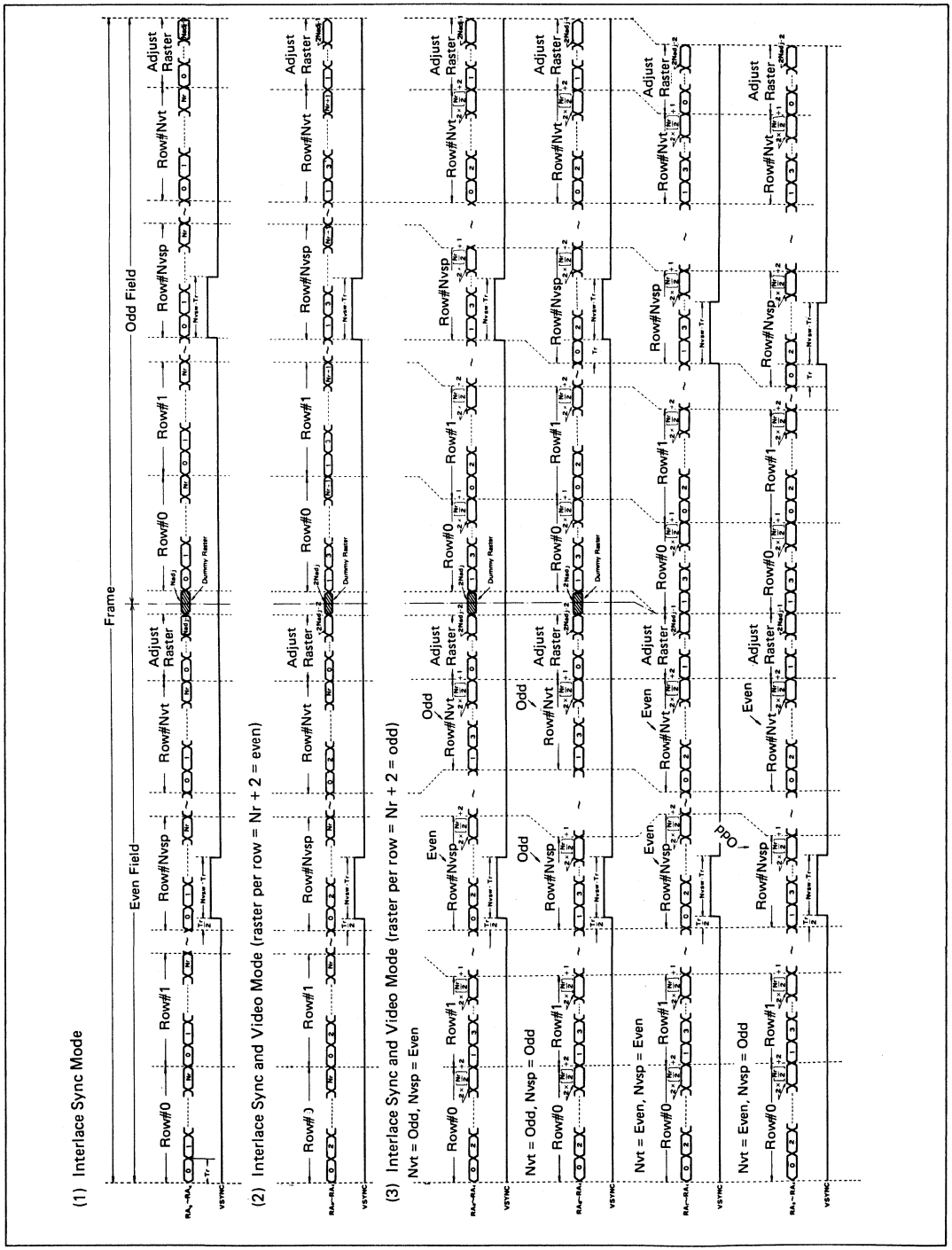


Figure 12 Raster Scanning Timing

MEMORY WIDTH SETTING

The offset value which is the difference between the display screen width and the display memory width in the horizontal direction can be specified in units of characters. (See Fig. 13)

Scrolling in any direction can be accomplished in units of characters, by setting the display memory width (horizontal direction) and the offset value, and by changing the start addresses of split-screens 1–4. This is performed by the memory width offset register (R33) and the MW bit of the control 3 register (R32).

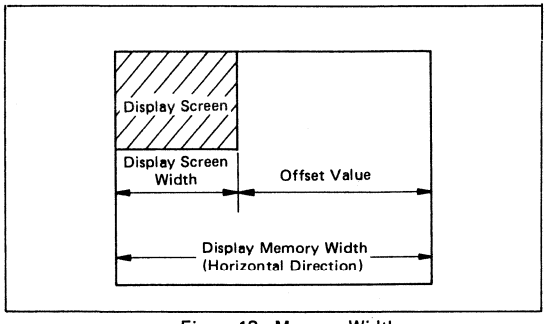


Figure 13 Memory Width

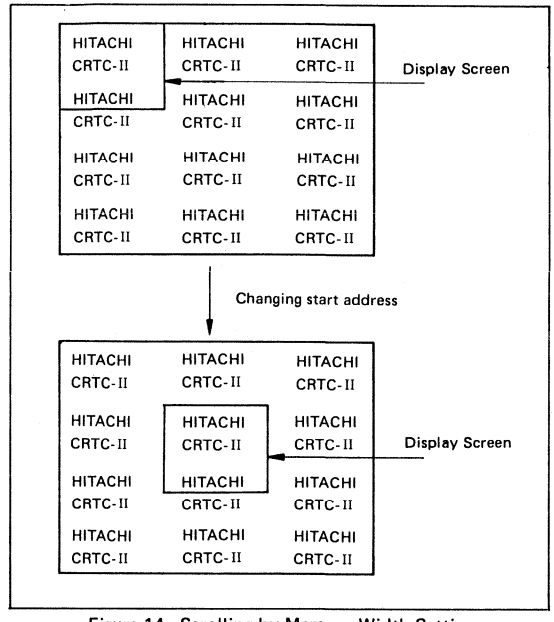
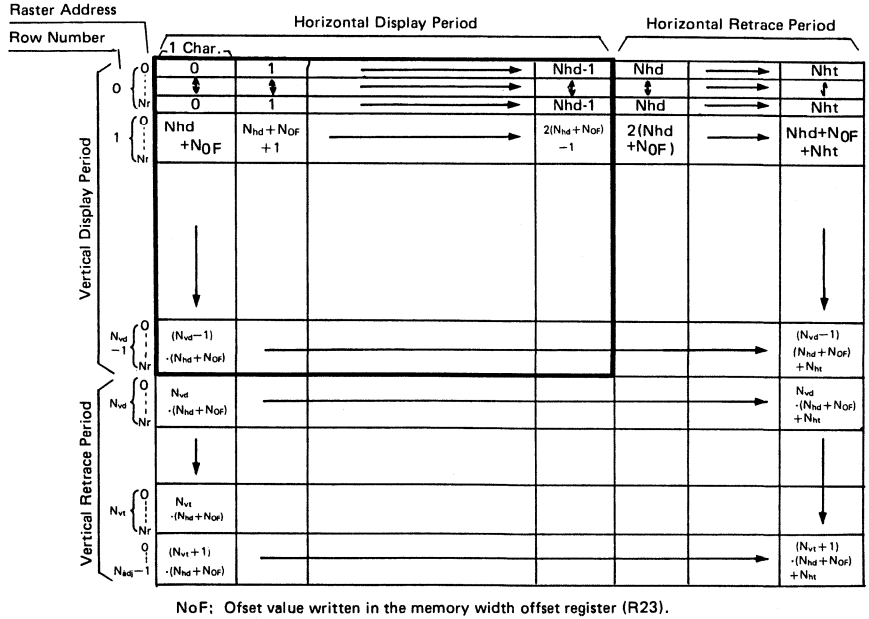


Figure 14 Scrolling by Memory Width Setting



NoF: Ofset value written in the memory width offset register (R23).

Figure 15 Memory Address and Raster Address in Memory Width Setting

SMOOTH SCROLLING

Smooth scrolling in the vertical direction can be accomplished by changing the start raster address in a character row. Whether scrolling in each split-screen is available or not can be selected. Selected split-screens scroll in the same way up to four split-screens simultaneously.

Smooth scrolling is performed by bits SS_1–SS_4 of the control 2 register (R31), and the smooth scrolling register (R29).

Smooth scrolling can be used in the Non-Interlace mode and the Interlace sync mode, but not in the Interlace sync and video mode.

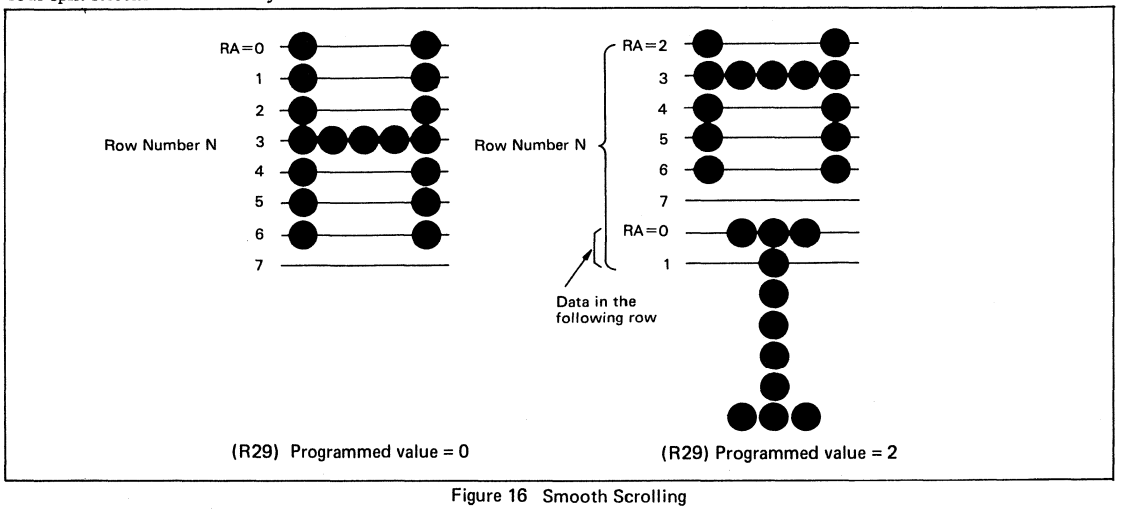


Figure 16 Smooth Scrolling

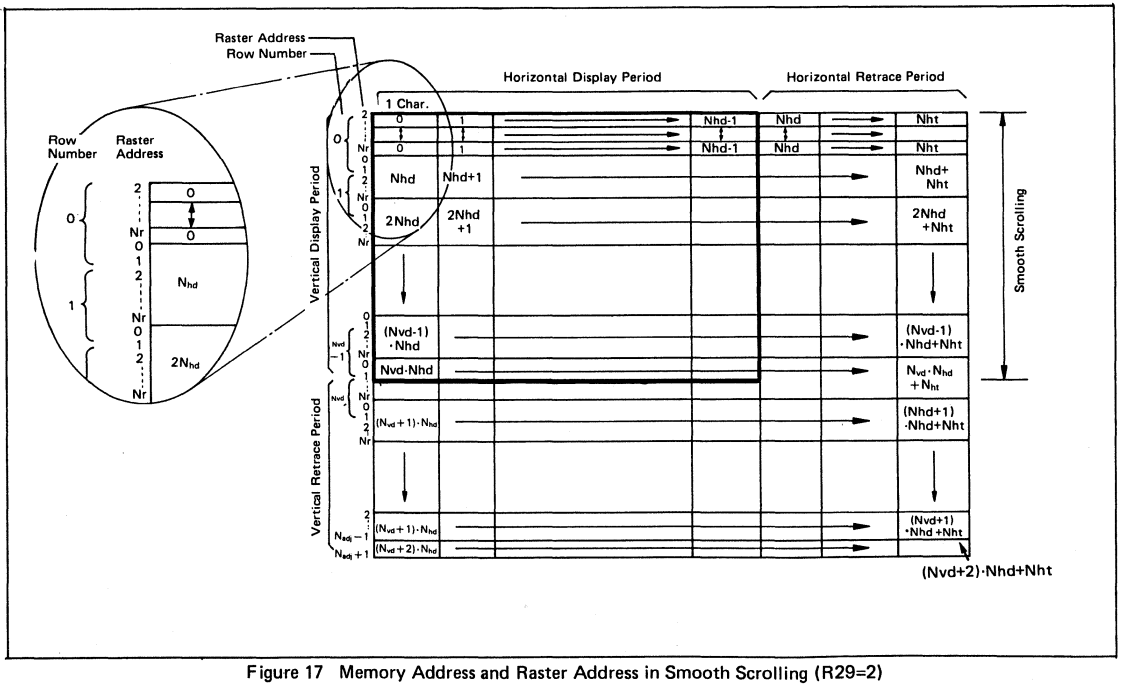


Figure 17 Memory Address and Raster Address in Smooth Scrolling (R29=2)

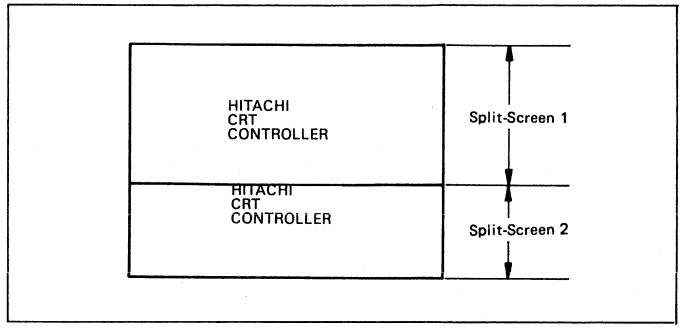


Figure 18 Smooth Scrolling in the Split-Screens
(Split-Screen 2 is scrolling.)

RASTER INTERPOLATION

In raster interpolation, the raster address is incremented every two rasters, thus the displayed image is doubled in the vertical direction. The vertical scanning cycle is also doubled.

Raster interpolation is performed by the RI bit of the control 2 register (R31).

This function can be used in the Non-Interlace mode and the Interlace sync mode, but not in the Interlace sync and video mode.

Figure 19 is an example of display with raster interpolation.

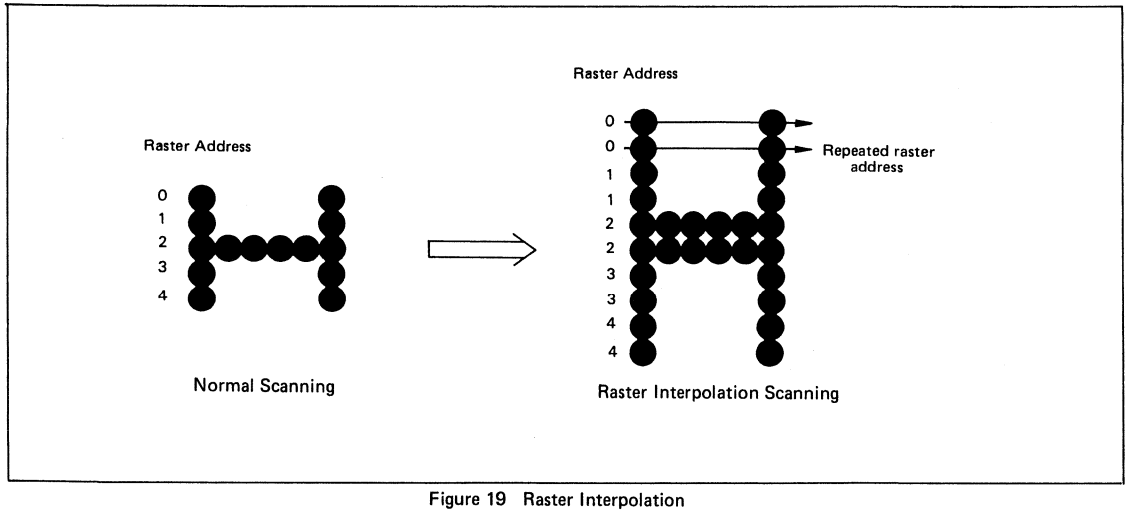


Figure 19 Raster Interpolation

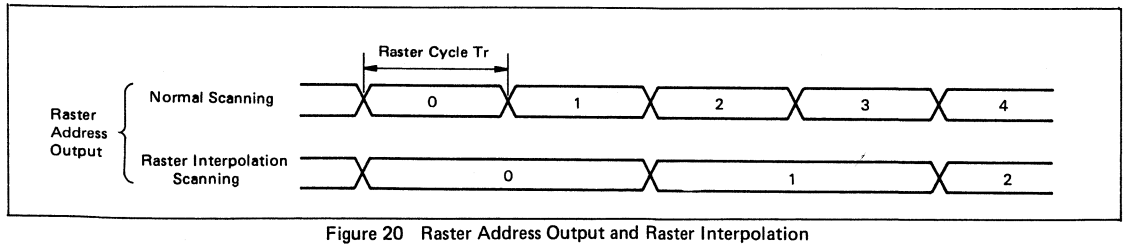


Figure 20 Raster Address Output and Raster Interpolation

EXTERNAL SYNCHRONIZATION

External synchronization (EX sync) has master-slave mode and TV sync mode.

The EX sync mode is controlled by bits VE, VS, and TV of the control 1 register (R30).

Master-slave mode is used to synchronize slave CRTCs with a master CRTC by VSYNC of a master CRTC. When superimposing a master screen with slave screens on the same CRT, clocks of a master and slave CRTCs can operate in different frequency on conditions shown below.

(1) Phase of master CRTC clock matches with slave CRTC clock at rising edge of VSYNC.
(2) Both master and slave CRTCs have the same horizontal/vertical scanning cycle.

Figure 21 illustrates the system configuration. In the Interlace sync mode and Interlace sync and video mode, the control 1 register must be set as to provide VSYNC output in odd fields of master CRTC.

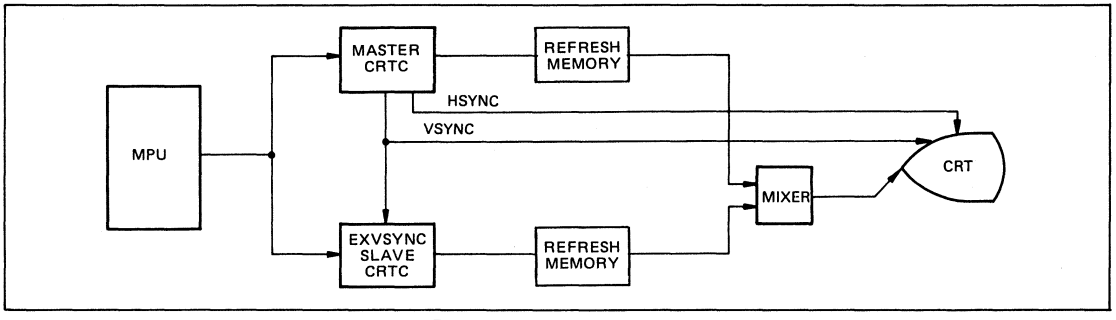


Figure 21 Master-Slave Mode

TV sync mode is used to synchronize the CRTC with the HSYNC and VSYNC signals of a TV's video signal.

In the TV sync mode, VSYNC/EXVSYNC and HSYNC/EXHSYNC pins function as input pins. The length of horizontal back porch is specified by the bits 0–3 of the sync width register (R3) and determines the display position in the horizontal direction.

In the Interlace sync and video mode, the TV sync mode cannot be used.

Figure 22 illustrates the system configuration.

In TV sync mode, when performing raster interpolation of slave CRTCs, Interlace sync mode or Interlace sync and video mode must not be set in a master CRTC; this causes the screen to move up and down by one raster.

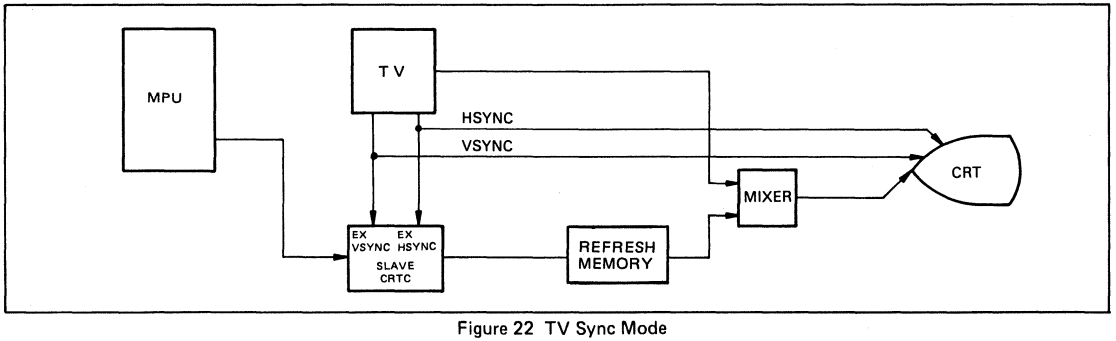


Figure 22 TV Sync Mode

External sync is valid on the conditions shown in Table 3.

Table 3 External Sync (V: valid, I: invalid, —: program inhibited)

Sync Mode	Slave Mode / Master Mode	Non-Interlace	Interlace Sync Mode	Interlace Sync and Video Mode
Master-Slave Mode	Non-Interlace Mode	V	I	I
	Interlace Sync Mode	I	V	V
	Interlace Sync and Video Mode	I	V	V
TV Sync Mode	Non-Interlace Mode	V	—	—
	Interlace Sync Mode	V	—	—
	Interlace Sync and Video Mode	V	—	—

Note) Slave CRTCs are always Non-Interlace mode in TV sync mode.

INTERRUPT REQUEST

An interrupt request signal to the MPU is output in the timing shown in Fig. 23. Interrupt request is generated by the vertical blanking period, or the light pen input.

Reading the status register (R31) clears interrupt request signal. Thus, if MPU does not read the status register (R31) when the interrupt request is generated, an interrupt request signal is output during the horizontal and vertical retrace period.

In the DPRAM mode, an interrupt request signal is not output.

This function is controlled by using bits IB and IL of the control 1 register (R30).

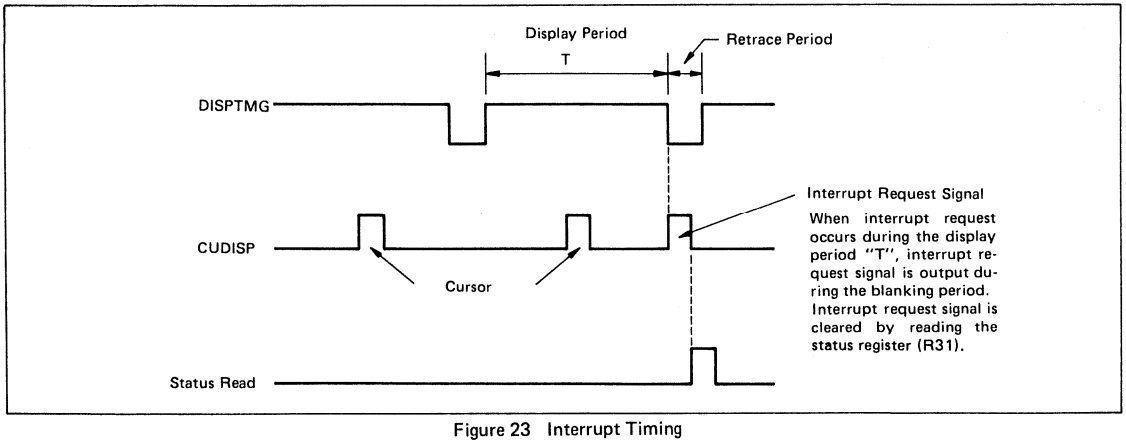


Figure 23 Interrupt Timing

THREE-STATE CONTROL OF MA/RA

Memory address (MA) and raster address (RA) outputs can be three-stated, using the TSC input pin. Three-state control is enabled by setting the TC bit of the control 3 register (R32).

By using this function, multiplexer (MPX) which selects address lines from MPU and CRTC for refresh memory is not needed.

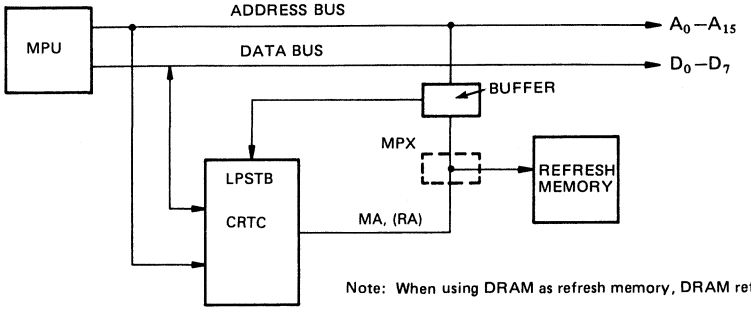


Figure 24 Three-State Control

DPRAM MODE

When the DPRAM mode is selected, the HD6345 generates a programmable timing signal from the access inhibit pin. This signal, shown in Fig. 25 as access inhibit period, provides the timing for the MPU to access to dual port memory.

In the DPRAM mode, interrupt request signal is not output, and the cursor 2 is not displayed.

This timing signal is available by using the DR bit of the control 3 register (R32), and the cursor 2 width register (R39).

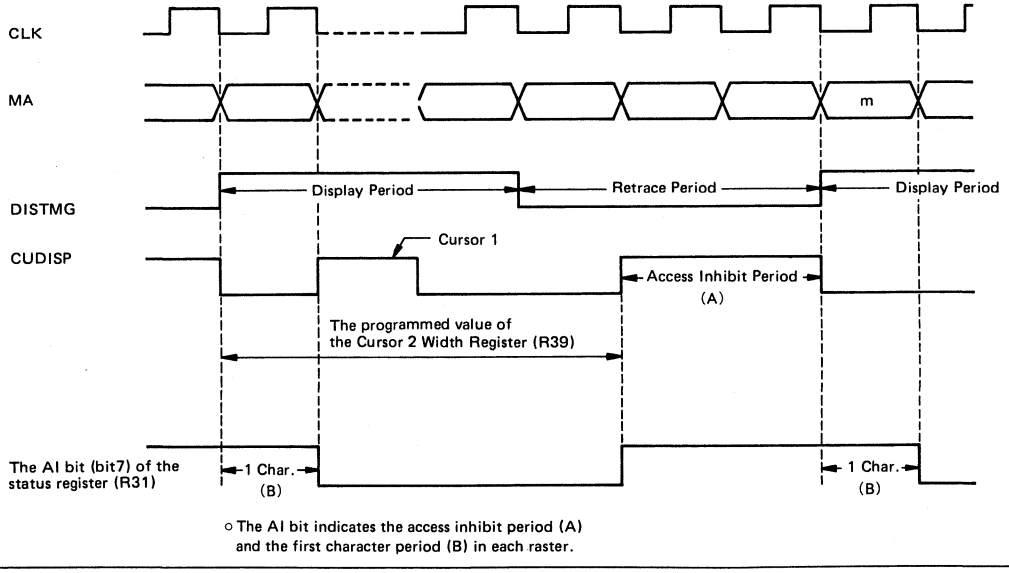


Figure 25 DPRAM Mode Output Timing

■ INTERNAL REGISTERS

HD6345 has one address register (AR) and forty data registers (R0 – R39). One register out of 40 data registers is selected by writing the address number of the register into the address register. Then the MPU can transfer data to or from the selected data register.

Write "0" to unused data bits (appear as [—] in the register table), since these bits are reserved for the future extension.

ADDRESS REGISTER (AR)

Data Bit								Program Unit	R/W
7	6	5	4	3	2	1	0	——	W
–	–	Register Address							

This register specifies the address number of the data register to be accessed. When both RS and $\overline{CS}$ are at "L" level, this register is selected. Programming the data from 40 to 63 produces no result.

HORIZONTAL TOTAL CHARACTERS REGISTER (R0)

Data Bit								Program Unit	R/W
7	6	5	4	3	2	1	0	Character	W
Nht (No. of Char. –1)									

This 8-bit register determines the horizontal scanning cycle. The programmed value is the total number of displayed and non-displayed characters per raster, minus one.

In the Interlace sync mode, the programmed value, Nht, must be odd.

HORIZONTAL DISPLAYED CHARACTERS REGISTER (R1)

Data Bit								Program Unit	R/W
7	6	5	4	3	2	1	0	Character	W
Nhd (Displayed Characters)									

This 8-bit register specifies the number of displayed characters per row. Any number less than the total number of characters can be programmed into this register.

HORIZONTAL SYNC POSITION REGISTER (R2)

Data Bit								Program Unit	R/W
7	6	5	4	3	2	1	0	Character	W
Nhsp (Horizontal Sync Position – 1)									

This 8-bit register is used to program horizontal sync position as a multiple of the character clock time. The programmed value is the character number of horizontal sync position, minus one. Any number equal to or less than the horizontal total characters register value (R0) can be programmed. When the programmed value of this register is increased, the display position on the screen is shifted to the left. When the programmed value is decreased, the display position is shifted to the right. Thus the optimum horizontal position can be determined.

SYNC WIDTH REGISTER (R3)

Data Bit								Program Unit	R/W
7	6	5	4	3	2	1	0	H: Character V: Raster	W
Wv_3	Wv_2	Wv_1	Wv_0	Wh_3	Wh_2	Wh_1	Wh_0		

This register determines the widths of both horizontal sync pulse and vertical sync pulse. The horizontal sync pulse width is programmed from 1-to-15 character clock times in the low-order four bits. "0" cannot be programmed. The vertical sync pulse width is programmed from 1-to-16 raster times in the high-order four bits. When "0" is programmed in the high-order four bits, a vertical sync pulse width is 16 raster times (16H).

Table 4 Horizontal Sync Pulse Width

HSW				Pulse Width
2^3	2^2	2^1	2^0	
0	0	0	0	Not Used
0	0	0	1	1 CH
0	0	1	0	2
0	0	1	1	3
0	1	0	0	4
0	1	0	1	5
0	1	1	0	6
0	1	1	1	7
1	0	0	0	8
1	0	0	1	9
1	0	1	0	10
1	0	1	1	11
1	1	0	0	12
1	1	0	1	13
1	1	1	0	14
1	1	1	1	15

CH: Character Clock Time

Table 5 Vertical Sync Pulse Width

VSW				Pulse Width
2^7	2^6	2^5	2^4	
0	0	0	0	16 H
0	0	0	1	1
0	0	1	0	2
0	0	1	1	3
0	1	0	0	4
0	1	0	1	5
0	1	1	0	6
0	1	1	1	7
1	0	0	0	8
1	0	0	1	9
1	0	1	0	10
1	0	1	1	11
1	1	0	0	12
1	1	0	1	13
1	1	1	0	14
1	1	1	1	15

H: Raster Time

VERTICAL TOTAL ROWS REGISTER (R4)

Data Bit								Program Unit	R/W
7	6	5	4	3	2	1	0	Row	W
Nvt (No. of Char. Rows – 1)									

This 8-bit register determines the vertical scanning cycle. The programmed value is the total number of character rows in a field, minus one.

VERTICAL TOTAL ADJUST REGISTER (R5)

Data Bit								Program Unit	R/W
7	6	5	4	3	2	1	0	Raster	W
–	–	–	Nadj (No. of Rasters)						

This register determines the number of additional rasters to adjust total number of rasters in a field. The optimum number from 0-to-31 can be programmed. Thus fine adjustment of vertical scanning cycle is performed.

VERTICAL DISPLAYED ROWS REGISTER (R6)

Data Bit								Program Unit	R/W
7	6	5	4	3	2	1	0	Row	W
Nvd (Displayed Char. Rows)									

This 8-bit register specifies the number of displayed character rows in each field. Any number less than the total number of character rows can be programmed into this register.

VERTICAL SYNC POSITION REGISTER (R7)

Data Bit								Program Unit	R/W
7	6	5	4	3	2	1	0	Row	W
Nvsp (Vertical Sync Position – 1)									

This 8-bit register is used to program vertical sync position on the screen as a multiple of the character row time. The programmed value is the character number of vertical sync position, minus one. Any number equal to or less than the vertical total rows register value (R4) can be programmed. When the programmed value of this register is increased, the display position is shifted up. When the programmed value is decreased, the display position is shifted down. Thus the optimum vertical position can be determined.

INTERLACE MODE AND SKEW REGISTER (R8)

Data Bit								Program Unit	R/W
7	6	5	4	3	2	1	0	——	W
C_1	C_0	D_1	D_0	–	–	V	S		

This register selects the raster scanning mode and controls the skew (delay) of CUDISP and DISPTMG outputs.

- **Raster Scanning Mode (V, S)**

The low-order two bits select the raster scanning mode.

V	S	Raster Scanning Mode
0	0	Non-Interlace mode
0	1	Interlace sync mode
1	0	Non-Interlace mode
1	1	Interlace sync and video mode

- **Skew (C_1, C_0, D_1, D_0)**

The bits D_1/D_0 and C_1/C_0 specify the skew of DISPTMG and CUDISP outputs, respectively. Skew control makes these output timings match with the serial video signals by delaying these outputs as programmed, in order to assure the access time to refresh memory and character generator.

D_1	D_0	DISPTMG
0	0	No skew
0	1	One character skew
1	0	Two character skew
1	1	Not available

C_1	C_0	CUDISP
0	0	No skew
0	1	One character skew
1	0	Two character skew
1	1	Not available

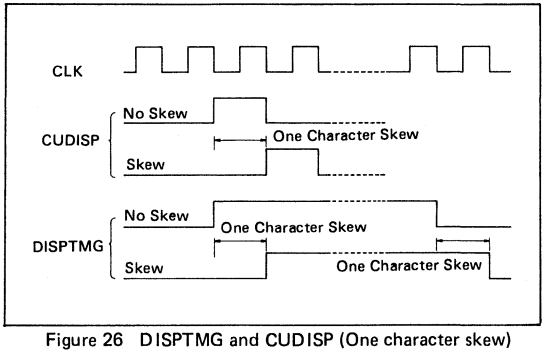


Figure 26 DISPTMG and CUDISP (One character skew)

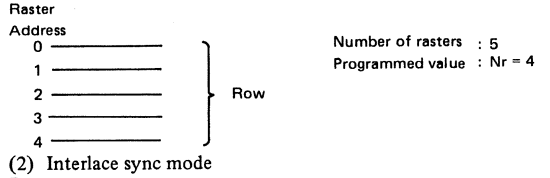
MAXIMUM RASTER ADDRESS REGISTER (R9)

Data Bit								Program Unit	R/W
7	6	5	4	3	2	1	0	Raster	W
–	–	–	Nr						

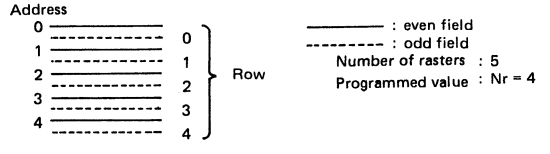
This 5-bit register determines the number of rasters per character row. When n means the number of rasters, the programmed value is as follows:

- Non-Interlace mode, Interlace sync mode : n – 1
- Interlace sync and video mode : n – 2

(1) Non-Interlace mode

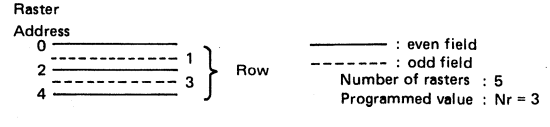


(2) Interlace sync mode



In the Interlace sync mode, a half number of rasters per row minus one should be programmed.

(3) Interlace sync and video mode



In the Interlace sync and video mode, the sum of rasters per row minus two should be programmed.

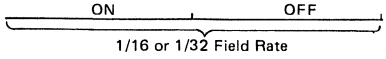
CURSOR 1 START REGISTER (R10)

Data Bit								Program Unit	R/W
7	6	5	4	3	2	1	0	Raster	W
–	B_1	P_1	Ncs_1 (Raster Address)						

This register determines the start raster address and selects the cursor blink mode for the cursor 1. The low-order five bits determines the start raster address. Bits B_1 and P_1 select the cursor blink mode.

B_1	P_1	Cursor Blink Mode
0	0	No blink
0	1	No cursor
1	0	Blink, 1/16 field rate
1	1	Blink, 1/32 field rate

Blink Rate



CURSOR 1 END REGISTER (R11)

Daata Bit								Program Unit	R/W
7	6	5	4	3	2	1	0	Raster	W
–	–	–	Nce_1 (Raster Address)						

This register determines the end raster address for the cursor-1.

SCREEN 1 START ADDRESS REGISTER (H, L) (R12, R13)

(R12)

Data Bit								Program Unit	R/W
7	6	5	4	3	2	1	0	Memory Address	R/W
–	–	Memory Address (H)							

(R13)

Data Bit								Program Unit	R/W
7	6	5	4	3	2	1	0	Memory Address	R/W
Memory Address (L)									

These registers determine the start memory address for the split-screen 1 display. Paging or scrolling is enabled by renewing these registers. The high-order two bits of R12 are always read as "0"s.

CURSOR 1 ADDRESS REGISTER (H, L) (R14, R15)

(R14)

Data Bit								Program Unit	R/W
7	6	5	4	3	2	1	0	Memory Address	R/W
–	–	Memory Address (H)							

(R15)

Data Bit								Program Unit	R/W
7	6	5	4	3	2	1	0	Memory Address	R/W
Memory Address (L)									

These registers determine the cursor 1 display memory address. The high-order two bits of R14 are always read as "0"s.

LIGHT PEN REGISTER (H, L) (R16, R17)

(R16)

Data Bit								Program Unit	R/W
7	6	5	4	3	2	1	0	—	R
–	–	Memory Address (H)							

(R17)

Data Bit								Program Unit	R/W
7	6	5	4	3	2	1	0	—	R
Memory Address (L)									

These registers store the light pen detection address. The high-order two bits of R16 are always read as "0"s.

Note that the stored address will be different from the actual address due to the following delays: address output delay, video signal output delay, light pen detection to LPSTB delay, and LPSTB to internal recognition delay. The relations between the LPSTB input and the memory address, raster address are shown in Figures 32, 33 in electrical characteristics.

SCREEN 2 START POSITION REGISTER (R18)

Data Bit								Program Unit	R/W
7	6	5	4	3	2	1	0	Row	R/W
Start Row – 1									

This register determines the start row of the split-screen 2. When the start row is to be "n"th, "n–1" must be set into this register.

If the split-screen 3 (or 4) start position register (R21 (or R24)) has already been programmed with the identical data, both of the split-screens 2 and 3 (or 4) will be disabled.

SCREEN 2 START ADDRESS REGISTER (H, L) (19, R20)

(R19)

Data Bit								Program Unit	R/W
7	6	5	4	3	2	1	0	Memory Address	R/W
–	–	Memory Address (H)							

(R20)

Data Bit								Program Unit	R/W
7	6	5	4	3	2	1	0	Memory Address	R/W
Memory Address (L)									

These registers determine the start memory address for the split-screen 2 display. Paging or scrolling is enabled by renewing these registers. The high-order two bits of R19 are always read as "0"s.

SCREEN 3 START POSITION REGISTER (R21)

Data Bit								Program Unit	R/W
7	6	5	4	3	2	1	0	Row	R/W
Start Row – 1									

This register determines the start row of the split-screen 3. When the start row is to be "n"th, "n–1" must be set into this register.

If the split-screen 2 (or 4) start position register (R18 (or R24)) has already been programmed with the identical data, both of the split-screens 3 and 2 (or 4) will be disabled.

SCREEN 3 START ADDRESS REGISTER (H, L) (R22, R23)

(R22)

Data Bit								Program Unit	R/W
7	6	5	4	3	2	1	0	Memory Address	R/W
–	–	Memory Address (H)							

(R23)

Data Bit								Program Unit	R/W
7	6	5	4	3	2	1	0	Memory Address	R/W
Memory Address (L)									

These registers determine the start memory address for the split-screen 3 display. Paging or scrolling is enabled by renewing these registers. The high-order two bits of R22 are always read as "0"s.

SCREEN 4 START POSITION RECISTER (R24)

Data Bit								Program Unit	R/W
7	6	5	4	3	2	1	0	Row	R/W
Start Row – 1									

This register determines the start row of the split-screen 4. When the start row is to be "n"th, "n–1" must be set into this register.

If the split-screen 2 (or 3) start position register (R18 (or R21)) has already been programmed with the identical data, both of the split-screens 4 and 2 (or 3) will be disabled.

SCREEN 4 START ADDRESS REGISTER (H, L) (R25, R26)

(R25)

Data Bit								Program Unit	R/W
7	6	5	4	3	2	1	0	Memory Address	R/W
–	–	Memory Address (H)							

(R26)

Data Bit								Program Unit	R/W
7	6	5	4	3	2	1	0	Memory Address	R/W
Memory Address (L)									

These registers determine the start memory address for the split-screen 4 display. Paging or scrolling is enabled by renewing these registers. The high-order two bits of R25 are always read as "0"s.

VERTICAL SYNC POSITION ADJUST REGISTER (R27)

Data Bit								Program Unit	R/W
7	6	5	4	3	2	1	0	Raster	W
–	–	–	Nvad (No. of Rasters)						

This register performs a fine adjustment on the vertical sync signal output in units of rasters. The VSYNC signal is supplied after the delay of specified number of rasters. The programmable number "n" is equal to or less than the maximum raster address register value (R9).

This register is enabled when "SY (bit 3) = 1" is set into the control 1 register (R30). If an adjustment is not required, the SY bit or this register (R27) is requested to be set as "0".

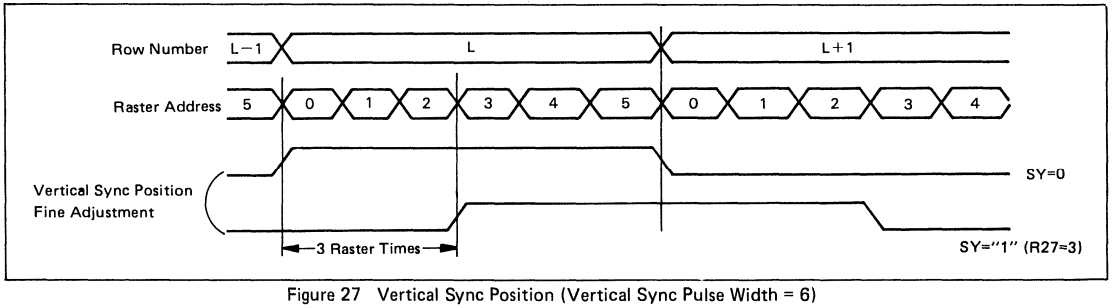


Figure 27 Vertical Sync Position (Vertical Sync Pulse Width = 6)

LIGHT PEN RASTER REGISTER (R28)

Data Bit								Program Unit	R/W
7	6	5	4	3	2	1	0	—	R
DP	–	–	Raster Address						

This register stores the light pen detection raster address and the detection period. The raster address is detected and stored into this register when the LPSTB input is asserted. The DP (bit 7) indicates the period in which the light pen strobe is detected. "DP = 1" is stored when the LPSTB is asserted during the display period, and "DP = 0" is stored when it occurs during the blanking period.

SMOOTH SCROLLING REGISTER (R29)

Data Bit								Program Unit	R/W
7	6	5	4	3	2	1	0	Raster	R/W
–	–	–	Nss (Raster Address)						

This register determines the start raster address within a row. By renewing this register, a smooth scrolling is provided for the screen specified by the bits SS_1 – SS_4 of the control 2 register. The programmable number is equal to or less than the maximum raster address register value (R9).

This register is valid only in the Non-Interlace mode and the Interlace sync mode.

CONTROL 1 REGISTER (R30)

Data Bit								Program Unit	R/W
7	6	5	4	3	2	1	0	—	W
VE	VS	IB	IL	SY	TV	SP_1	SP_0		

This register controls the following by the corresponding bits. A device reset clears all bits of this register.

VE, VS, TV : External synchronization control
IB, IL : Interrupt control
SY : Vertical sync position adjust control
SP_1, SP_0 : Screen split control

- **External Synchronization Control (VE, VS, TV)**

(1) Operation mode alteration

VE	Function
0	EXVSYNC/VSYNC corresponds to VSYNC output
1	EXVSYNC/VSYNC corresponds to EXVSYNC input

VS	Function
0	DISPTMG activated
1	DISPTMG keeps "L"

TV	Function
0	EXHSYNC/HSYNC corresponds to HSYNC output Sync width register (R3) defines the HSYNC pulse width
1	EXHSYNC/HSYNC corresponds to EXHSYNC input Sync width register (R3) defines the horizontal back porch period

(2) EXVSYNC and VSYNC control

< VE = 1 >

VS	TV	EXVSYNC
0	0	Signals supplied to EXVSYNC is ignored
0	1	EXVSYNC/VSYNC corresponds to EXVSYNC input and synchronized with the external signal
1	0	
1	1	

< VE = 0 >

	VSYNC		
VS	Non-Interlace	Interlace sync	Interlace sync and video*
0	VSYNC signal supplied	VSYNC signal supplied	
1		VSYNC signal supplied only in odd field scanning	

Note) * Attention to the limitation item described in Note) of Table 6 for Interlace sync and video mode.

Table 6 External Synchronization Control

TV	VE	VS	Function
0	0	0	Set as master CRTC in master-slave mode*
0	0	1	Set as master CRTC in master-slave mode, in Interlace sync mode or Interlace sync and video mode, upon synchronization
0	1	0	Set as slave CRTC in master-slave mode
0	1	1	Set as slave CRTC in master-slave mode, upon synchronization
1	0	0	Program inhibited
1	0	1	Program inhibited
1	1	0	Set as slave CRTC in TV sync mode**
1	1	1	Set as slave CRTC in TV sync mode, upon synchronization**

Note) * "000" is to be set also when not performing the external synchronization.

** In TV sync mode, DISPTMG is supplied after the back porch period.

- **Interrupt Control (IB, IL)**

Control 1 Register IB	IL	Function
0	0	IRQ signal not supplied
0	1	IRQ signal supplied by light pen strobe
1	0	IRQ signal supplied by vertical blanking
1	1	IRQ signal supplied by light pen strobe or vertical blanking

Note) The IRQ signal is supplied from the IRQ pin while the DISPTMG is "L".

- **Vertical Sync Position Adjust Control (SY)**

Control 1 Register SY	Function
0	Vertical sync position adjust register (R27) disabled
1	Vertical sync position adjust register (R27) enabled

- **Screen Split Control (SP_1, SP_0)**

Control 1 Register SP_1	SP_0	Function
0	0	Screen start position regs. (R18, R21, R24) disabled
0	1	Screen 2 start position register (R18) enabled
1	0	Screen 2 and 3 start position regs. (R18, R21) enabled
1	1	Screen 2 to 4 start position regs. (R18, R21, R24) enabled

CONTROL 2 / STATUS REGISTER (R31)

Data Bit								Program Unit	R/W	
7	6	5	4	3	2	1	0			
SS_4	SS_3	SS_2	SS_1	RI	–	–	–	—	W	Control 2
AI	0	0	0	0	E	SB	SL		R	Status

During a write transaction, this register specifies the screen to be scrolled smoothly and provides the double-size vertical display with the raster interpolation function. A device reset clears the bits of the control 2 register.

During a read transaction, this register indicates the status such as display field, vertical blanking, light pen strobe and access inhibit in DPRAM mode. A device reset clears the SB bit and SL bit of the status register.

Refer to Table 10 Reset State of Internal Registers for details.

< Control 2 Register >

SS_4, SS_3, SS_2, SS_1 : Smooth scrolling control

RI : Raster interpolation control

< Status Register >

AI, E, SB, SL : Status indication

- **Smooth Scrolling Control (SS1–SS4)**

Setting the bits SS1 to SS4 enables smooth scrolling on the split-screens 1 to 4, respectively. The smooth scrolling register (R29) is enabled for the specified split-screen.

		Smooth Scrolling
SS_1	0	Disabled on the split-screen 1
	1	Enabled on the split-screen 1
SS_2	0	Disabled on the split-screen 2
	1	Enabled on the split-screen 2
SS_3	0	Disabled on the split-screen 3
	1	Enabled on the split-screen 3
SS_4	0	Disabled on the split-screen 4
	1	Enabled on the split-screen 4

- **Raster Interpolation Control (RI)**

Setting "1" into this bit performs a raster interpolation. The raster address is incremented every two rasters, doubling the vertical scanning cycle.

This function is provided only in the Non-Interlace mode and the Interlace sync mode.

- **Status Indication (AI, E, SB, SL)**

Access Inhibit Status Bit : AI

AI	Status
0	Refresh memory allowed to be accessed.
1	Refresh memory inhibited to be accessed.

Display Field Status Bit : E

E	Status
0	During odd field display
1	During even field display

Note) E is always "0" in the Non-Interlace mode.

Vertical Blanking Status Bit : SB

SB	Status
0	Other than during vertical blanking
1	During vertical blanking

Light Pen Strobe Status Bit : SL

SL	Status
0	Light pen strobe not detected.
1	Light pen strobe detected.

Reading this register clears the SL, but not SB. Also, the IRQ output signal goes "L" upon read access of this register. The low-order three bits of the control register 2 are requested to always be written with "0"s. The bits 3-6 of the Status register are always read as "0"s.

CONTROL 3 REGISTER (R32)

Data Bit								Program Unit	R/W
7	6	5	4	3	2	1	0	—	W
CM	C_2	CW_1	CW_2	MW	TC	DR	–		

This register controls the following by the corresponding bits. A device reset clears this register.

CM : Cursor display mode control
C_2 : Cursor 2 enable
CW_1 : Cursor 1 width control
CW_2 : Cursor 2 width control
MW : Memory width control
TC : Three-state control
DR : DPRAM mode selection

- **Cursors Display Mode Control (CM)**

CM	Cursors Display Mode
0	EOR mode
1	OR mode

- **Cursor 2 Enable (C_2)**

C_2	Cursor 2
0	Cursor 2 disabled
1	Cursor 2 enabled

- **Cursor 1 Width Control (CW_1)**

CW_1	Cursor 1 Width
0	Cursor 1 width register (R38) disabled 1-character width specified as cursor 1 width
1	Cursoer 1 width register (R38) enabled Set value in R38 specified as cursor 1 width

- **Cursor 2 Width Control (CW_2)**

CW_2	Cursor 2 Width
0	Cursor 2 width register (R39) disabled 1-character width specified as cursor 2 width
1	Cursor 2 width register (R39) enabled Set value in R39 specified as cursor 2 width

- **Memory Width Control (MW)**

MW	Memory Width
0	Memory width offset register (R33) disabled Linear address supplied as memory address
1	Memory width offset register (R33) enabled Memory width definable

- **Three-State Control (TC)**

TC	MA and RA Outputs
0	Three-state control disabled
1	Memory address supplied on MA_{0-13} and RA_{0-4} when TSC is "L"
	MA_{0-13} and RA_{0-4} stated into high-impedance when TSC is "H"

- **DPRAM Mode Selection (DR)**

DR	DPRAM Mode Selection
0	DPRAM control signal not supplied
1	Enters DPRAM mode

MEMORY WIDTH OFFSET REGISTER (R33)

Data Bit								Program Unit	R/W
7	6	5	4	3	2	1	0	Character	R/W
Nof (Offset Address)									

This register specifies the offset value to be supplemented to the memory address, in units of characters, in order to define the start memory address of the next row. Adding the offset value to the memory address makes the memory width wider than the display width. Renewing the start memory address enables the display screen to be scrolled in any direction within a memory space by character. If the offset value "M" is set into this register, the start address of the next row is to be the last displayed character address + "M + 1".

This register is enabled when "MW (bit 3) = 1" is set into the control 3 register (R32). If an offset value is not required, the MW bit of R32 or this register (R33) is requested to be set as "0".

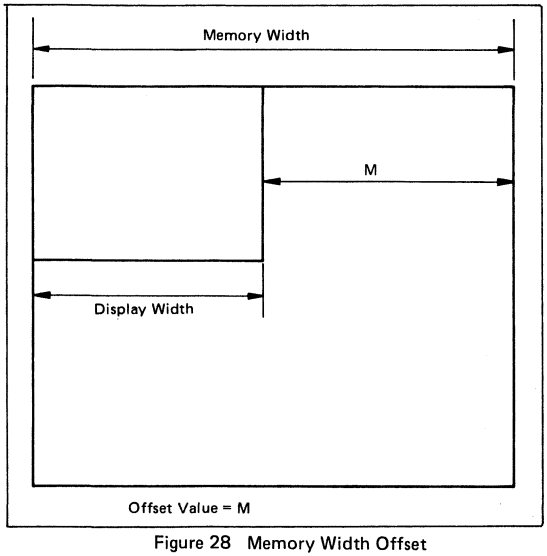


Figure 28 Memory Width Offset

CURSOR 2 START REGISTER (R34)

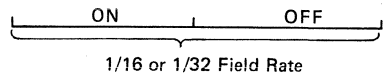
Data Bit								Program Unit	R/W
7	6	5	4	3	2	1	0	Raster	W
–	B_2	P_2	Ncs_2 (Raster Address)						

This register determines the start raster address and selects the cursor blink mode for the cursor 2. The low-order 5 bits determines the start raster address. Bits B_2 and P_2 select the cursor blink mode.

If the C_2 bit of the control 3 register is "0", or in the DPRAM mode, this register is invalid.

B_2	P_2	Cursor Blink Mode
0	0	No blink
0	1	No cursor
1	0	Blink, 1/16 field rate
1	1	Blink, 1/32 field rate

Blink Rate



CURSOR 2 END REGISTER (R35)

Data Bit								Program Unit	R/W
7	6	5	4	3	2	1	0	Raster	W
–	–	–	Nce_2 (Raster Address)						

This register determines the end raster address for the cursor 2.

If the C_2 bit of the control 3 register is "0", or in the DPRAM mode, this register is invalid.

CURSOR 2 ADDRESS REGISTER (H, L) (R36, R37)

(R36)

Data Bit								Program Unit	R/W
7	6	5	4	3	2	1	0	Memory Address	R/W
–	–	Memory Address (H)							

(R37)

Data Bit								Program Unit	R/W
7	6	5	4	3	2	1	0	Memory Address	R/W
Memory Address (L)									

These registers determine the cursor 2 display memory address. The high-order two bits of R36 are always read as "0"s.

This register is disabled in DPRAM mode or when "C_2 = 0" is set in the control 3 register.

CURSOR 1 WIDTH REGISTER (R38)

Data Bit								Program Unit	R/W
7	6	5	4	3	2	1	0	Character	R/W
Ncw_1 (Number of Characters)									

This 8-bit register specifies the cursor 1 width in units of characters. Writing "0" into this register disables the cursor 1 display.

This register is enabled when "CW_1 (bit 5) = 1" is set into the control 3 register (R32).

CURSOR 2 WIDTH REGISTER (R39)

Data Bit								Program Unit	R/W
7	6	5	4	3	2	1	0	Character	R/W
Ncw_2 (Number of Characters)									

This 8-bit register specifies the cursor 2 width in units of characters. Writing "0" into this register disables the cursor 2 display.

This register is enabled when "CW_2 (bit 4) = 1" is set into the control 3 register (R32).

In DPRAM mode, this register specifies the ACI signal output timing, not affected by the CW_2 bit at all.

■ LIMITATION FOR PROGRAMMING

The register programmed value is limited as listed in the table below.

Table 7 Limitation on Register Programmed Value

Function	Register Programmed Value Range	Associated Reg.
Screen Format	$1 < Nhd < Nht + 1 \leqq 256$	R1, R0
	$0 < Nvd < Nvt + 1 \leqq 256$	R6, R4
	$0 \leqq Nhsp \leqq Nht$	R2, R0
	$0 \leqq Nvsp \leqq Nvt$*	R7, R4
	$2 \leqq Nr \leqq 30$ (Interlace sync and video mode)	R9
	$Nvad < Nr$	R27, R9
	$3 \leqq Nht$ (Interlace sync mode and Interlace sync and video mode) $5 \leqq Nht$ (Non-Interlace mode)	R0
Cursor Control	$0 \leqq Ncs_1 \leqq Nce_1$	R10, R11
	$Nce_1 \leqq Nr$ (Non-Interlace mode and Interlace sync mode) $Nce_1 < Nr$ (Interlace sync and video mode)	R11, R9
	$0 \leqq Ncs_2 \leqq Nce_2$	R34, R35
	$Nce_2 \leqq Nr$ (Non-Interlace mode and Interlace sync mode) $Nce_2 < Nr$ (Interlace sync and video mode)	R35, R9
	$0 \leqq Ncw_1 \leqq 255$	R38
	$0 \leqq Ncw_2 \leqq 255$	R39
Smooth Scrolling	$Nss \leqq Nr$	R29, R9
Memory Width Setting	$0 \leqq Nof \leqq 255$	R33

Notes 1) * In the Interlace mode, if the vertical sync signal assertion strides over the next field, the signal pulse width is alternately increased or decreased by 1/2 raster period in the following fields.
2) Refer to INTERNAL REGISTER ASSIGNMENT for symbols.

■ RESET

The $\overline{RES}$ functions as a reset input signal only while the LPSTB is "L". "Reset" is definable in two stages.

(1) "During a reset state" indicates the period that the $\overline{RES}$ remains "L".

(2) "After a reset state" indicates the state after the $\overline{RES}$ transition from "L" to "H".

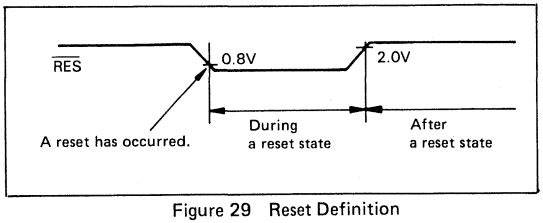


Figure 29 Reset Definition

The note for a reset is listed in Table 8, and the pin status during a reset state is in Table 9.

Table 8 Note for a Reset

	Note
During a Reset State	1. HD6345 sets "6845S mode*" Control registers R30, R31, and R32 cannot be programmed
After a Reset State	1. HD6345 remains "6845S mode" until control registers R30, R31, R32 are programmed 2. In external sync mode, the additional circuit is required to prevent the contention between sync signals (VSYNC, HSYNC) of a master and those of a slave

The $\overline{RES}$ assertion at power-on does not define the internal registers. The internal operation remains undefined until all the internal registers have been programmed.

Note) * "6845S mode"
The 6845S mode causes the HD6345 to implement the HD6845S functions. The HD6345 is software-compatible with the HD6845S, and is provided with the extended functions of the HD6845S. Programming the control registers enables the extended functions. During a reset state, the control registers are initialized.
The 6845S mode is enabled by programming the internal registers R0 — R17 during a reset state after a power is supplied.

Table 9 Pin Status during a Reset State

Pin No.	Symbol	Pin Name	Input/ Output	Pin Status
1	Vss	Vss	–	——
2	$\overline{RES}$	RESET	Input	——
3	LPSTB	LIGHT PEN STROBE	Input	"L" level signal requested to be supplied
	TSC	THREE STATE CONTROL	Input	
4 – 17	$MA_0 - MA_{13}$	MEMORY ADDRESS 0 – 13	Output	Goes "L" immediately after reset
18	DISPTMG	DISPLAY TIMING	Output	Goes "L" immediately after reset
19	CUDISP	CURSOR DISPLAY	Output	Goes "L" immediately after reset
	ACI	ACCESS INHIBIT	Output	
	IRQ	INTERRUPT REQUEST	Output	
20	Vcc	Vcc	–	——
21	CLK	CHARACTER CLOCK	Input	Not affected
22	R/$\overline{W}$	READ/WRITE	Input	Not affected
23	E	ENABLE	Input	Not affected
24	RS	REGISTER SELECT	Input	Not affected
25	$\overline{CS}$	CHIP SELECT	Input	Not affected
26 – 33	$D_0 - D_7$	DATA BUS 0 – 7	Input/ Output	Not affected
34 – 38	$RA_0 - RA_4$	RASTER ADDRESS 0 – 4	Output	Goes "L" immediately after reset
39	HSYNC	HORIZONTAL SYNC	Output	Corresponds to HSYNC until external sync mode is set into the control register after reset
	EXHSYNC	EXTERNAL HORIZONTAL SYNC	Input	
40	VSYNC	VERTICAL SYNC	Output	Corresponds to VSYNC until external sync mode is set into the control register after reset
	EXVSYNC	EXTERNAL VERTICAL SYNC	Input	

Table 10 Reset State of Internal Registers

Reg. No.	Register Name	R/W	Data Bit (Reset State)							
			7	6	5	4	3	2	1	0
AR	ADDRESS REGISTER	W								
R0	HORIZONTAL TOTAL CHARACTERS	W								
R1	HORIZONTAL DISPLAYED CHARACTERS	W								
R2	HORIZONTAL SYNC POSITION	W								
R3	SYNC WIDTH	W								
R4	VERTICAL TOTAL ROWS	W								
R5	VERTICAL TOTAL ADJUST	W								
R6	VERTICAL DISPLAYED ROWS	W								
R7	VERTICAL SYNC POSITION	W								
R8	INTERLACE MODE AND SKEW	W								
R9	MAX. RASTER ADDRESS	W								
R10	CURSOR 1 START	W								
R11	CURSOR 1 END	W								
R12	SCREEN 1 START ADDRESS (H)	R/W								
R13	SCREEN 1 START ADDRESS (L)	R/W								
R14	CURSOR 1 ADDRESS (H)	R/W								
R15	CURSOR 1 ADDRESS (L)	R/W								
R16	LIGHT PEN (H)	R								
R17	LIGHT PEN (L)	R								
R18	SCREEN 2 START POSITION	R/W								
R19	SCREEN 2 START ADDRESS (H)	R/W								
R20	SCREEN 2 START ADDRESS (L)	R/W								
R21	SCREEN 3 START POSITION	R/W								
R22	SCREEN 3 START ADDRESS (H)	R/W								
R23	SCREEN 3 START ADDRESS (L)	R/W								
R24	SCREEN 4 START POSITION	R/W								
R25	SCREEN 4 START ADDRESS (H)	R/W								
R26	SCREEN 4 START ADDRESS (L)	R/W								
R27	VERTICAL SYNC POSITION ADJUST	W								
R28	LIGHT PEN RASTER	R								
R29	SMOOTH SCROLLING	R/W								
R30	CONTROL 1	W	0	0	0	0	0	0	0	0
R31	CONTROL 2	W	0	0	0	0	0			
	STATUS	R	0					*	1	0
R32	CONTROL 3	W	0	0	0	0	0	0	0	
R33	MEMORY WIDTH OFFSET	R/W								
R34	CURSOR 2 START	W								
R35	CURSOR 2 END	W								
R36	CURSOR 2 ADDRESS (H)	R/W								
R37	CURSOR 2 ADDRESS (L)	R/W								
R38	CURSOR 1 WIDTH	R/W								
R39	CURSOR 2 WIDTH	R/W								

Note) ☐ : not affected
(After power-on, the value is not fixed until it is programmed.)
0 : becomes "0"
1 : becomes "1".
* : becomes "0" in the Non-Interlace mode
becomes "1" in the Interlace sync mode and Interlace sync and video mode
(After power-on, its status is not fixed until the raster scanning mode is set.)
▒ : not used

■ NOTES AND LIMITATIONS FOR HD6345

NOTES

(1) The CRTC-II HD6345 is the CMOS LSI. The pin assignment of the HD6345 is the same as that of the HD6845S, but it should be noted, peculiar to CMOS LSIs, that the input pins of the HD6345 must not be left disconnected, etc.

(2) Refer to the RESET section for notes on a reset at power-on.

LIMITATION FOR PROGRAMMING

Refer to "Table 7 Limitation on Register Programmed Value" for details.

■ ANOMALOUS OPERATION BY REGISTER RENEWAL DURING SCREEN DISPLAY

The temporary erroneous operation may occur if renewing the internal register during a screen display period. Generally, the device starts the newly specified operation on and after the following field after a renewal.

Whether or not the register renewal is allowed during a display is shown by the symbols O, Δ, and X in the following table.

Register renewal is:

- O : Allowable
- Δ : Allowable with some conditions
  A temporary flicker may occur upon deviation of the conditions.
- X : Not recommended
  A temporary flicker may occur upon register renewal.

(These are presented only for reference, not guaranteed by Hitachi Ltd.)

Reg. No.	Register Name	Phenomenon and Renewal Recommended Period	
R0	Horizontal Total Characters	Horizontal scanning cycle is irregularized.	X
R1	Horizontal Displayed Characters	DISPTMG width may be set shorter than specified only during 1 raster period because of a momentary misrecognition of this register data.	O
R2	Horizontal Sync Position	HSYNC will not be supplied as required, or a noise may occur. It may be supplied as programmed on and after the following field.	X
R3	Sync Width	Sync pulse width may be set shorter than specified upon register renewal during "H" of HSYNC and VSYNC.	Δ
R4	Vertical Total Rows	Vertical scanning cycle may be irregularized upon renewal during the last raster scanning period within a row.	Δ
R5	Vertical Total Adjust	The specified number of adjust rasters will not be supplemented upon renewal within the last character clock time during adjust raster scanning period.	Δ
R6	Vertical Displayed Rows	Raster scanning may be suspended (DISPTMG goes "L".) immediately after a renewal within a field. The programmed display is enabled on and after the following field.	O
R7	Vertical Sync Position	VSYNC will not be supplied as required, or a noise may occur. It may be supplied as programmed on and after the following field.	X
R8	Interlace Mode and Skew	Renewing scanning mode bit irregularizes vertical scanning cycle. Renewing skew bit neglects the programmed position for screen and cursor displays.	X
R9	Maximum Raster Address	Vertical scanning cycle is irregularized.	X
R10 R11	Cursor 1 Start Cursor 1 End	Cursor raster scanning may be irregularized or blink period be temporarily set shorter upon renewal within the last character clock time during raster scanning period.	Δ
R12 R13	Screen 1 Start Address (H) Screen 1 Start Address (L)	Except during the last raster scanning period within a row, register renewal is allowable. Horizontal/vertical display period is especially recommended for renewal. * If R12 and R13 are separately renewed in the different fields, a screen display will temporarily start from the half-renewed address.	O
R14 R15	Cursor 1 Address (H) Cursor 1 Address (L)	The cursor will not temporarily be displayed at the specified address upon renewal during display period. Horizontal/vertical retrace period is especially recommended for renewal. * If R14 and R15 are separately renewed in the different fields, a cursor will be temporarily displayed on the half-renewed address.	O
R16 R17	Light Pen (H) Light Pen (L)	——	–
R18	Screen 2 Start Position	Except during raster scanning period prior to the split-screen 2 start row, renewal is allowable. Horizontal/vertical retrace period is especially recommended for renewal.	O
R19 R20	Screen 2 Start Address (H) Screen 2 Start Address (L)	Except during raster scanning period prior to the split-screen 2 start row, renewal is allowable. Horizontal/vertical retrace period is especially recommended for a renewal. * If R19 and R20 are separately renewed in the different fields, a screen display will temporarily start from the half-renewed address.	O
R21	Screen 3 Start Position	Except during raster scanning period prior to the split-screen 3 start row, renewal is allowable. Horizontal/vertical retrace period is especially recommended for renewal.	O

Reg. No.	Register Name	Phenomenon and Renewal Recommended Period	
R22 R23	Screen 3 Start Address (H) Screen 3 Start Address (L)	Except during raster scanning period prior to the split-screen 3 start row, renewal is allowable. Horizontal/vertical retrace period is especially recommended for renewal. * If R22 and R23 are separately renewed in the different fields, a screen display will temporarily start from the half-renewed address.	○
R24	Screen 4 Start Position	Except during raster scanning period prior to the split-screen 4 start row, renewal is allowable. Horizontal/vertical retrace period is especially recommended for renewal.	○
R25 R26	Screen 4 Start Address (H) Screen 4 Start Address (L)	Except during raster scanning period prior to the split-screen 4 start row, renewal is allowable. Horizontal/vertical retrace period is especially recommended for renewal. * If R25 and R26 are separately renewed in the different fields, a screen display will temporarily start from the half-renewed address.	○
R27	Vertical Sync Position Adjust	The programmed position for VSYNC output will not be satisfied.	×
R28	Light Pen Raster	——	–
R29	Smooth Scrolling	For a screen not performing smooth scroll, renewal is allowable except during the last raster scanning period within each row. For a screen performing smooth scroll, renewal is allowable except during raster scanning period on the address of "programmed value – 1". Horizontal/vertical retrace period is especially recommended for renewal.	Δ
R30	Control 1 • VE	Renewal is allowable except at VSYNC output.	○
	• TV	Renewal is allowable except at HSYNC output when DISPTMG is "L".	○
	• VS, IB, IL	Renewal is allowable.	○
	• SY	VSYNC will not be supplied as required, or a noise may occur. It may be supplied as programmed on and after the following field.	Δ
	• SP_0, SP_1	Temporary disturbance may occur on a screen upon renewal during display period. Vertical retrace period is especially recommended for renewal.	Δ
R31	Control 2 • SS_1 – SS_4	Temporary disturbance may occur on a screen upon renewal during display period. Vertical retrace period is especially recommended for renewal.	Δ
	• RI	Vertical raster scanning cycle is irregularized.	×
	Status	——	–
R32	Control 3 • CM, C_2 CW_1, CW_2	Temporary disturbance may occur on a cursor upon renewal during display period. Vertical retrace period is especially recommended for renewal.	Δ
	• MW	Temporary disturbance may occur on a screen upon renewal during display period. Vertical retrace period is especially recommended for renewal.	Δ
	• TC, DR	Renewal is inhibited.	×
R33	Memory Width Offset	The specified address for cursor display will not temporarily be satisfied upon renewal during display period. Horizontal/vertical retrace period is especially recommended for renewal.	○
R34 R35	Cursor 2 Start Cursor 2 End	Cursor raster scanning may be irregularized or blink period be temporarily set shorter upon renewal within the last character clock time during raster scanning period.	Δ
R36 R37	Cursor 2 Address (H) Cursor 2 Address (L)	The specified address for cursor display will not temporarily be satisfied upon renewal during display period. Horizontal/vertical retrace period is especially recommended for renewal. * If R36 and R37 are separately renewed in the different fields, a cursor will be temporarily displayed at the half-renewed address.	○
R38 R39	Cursor 1 Width Cursor 2 Width	The specified cursor width will not temporarily be satisfied upon renewal during display period. Horizontal/vertical retrace period is especially recommended for renewal.	Δ

■ ABSOLUTE MAXIMUM RATINGS

Item		Symbol	Rating	Unit
Supply Voltage		V_{CC}*	−0.3 ~ +7.0	V
Input Voltage		V_{in}*	−0.3 ~ V_{CC}+0.3	V
Operating Temperature		T_{opr}	−20 ~ +75	°C
Storage Temperature		T_{stg}	−55 ~ +150	°C
Allowable Output Current	Data Bus	$\|I_o\|$**	5	mA
	Others		3	mA
Total Allowable Output Current		$\|\Sigma I_o\|$***	60	mA

* This value is in reference to V_{SS} = 0V.
** The allowable output current is the maximum current that may be drawn from, or flow out to, one output pin or one input/output common pin.
*** The total allowable output current is the total sum of currents that may be drawn from, or flow out to, output pins or input/output common pins.
Note) Using an LSI beyond its maximum ratings may result in its permanent destruction. LSI's should usually be under recommended operating conditions. Exceeding any of these conditions may adversely affect its reliability.

■ RECOMMENDED OPERATING CONDITIONS

Item	Symbol	Min	Typ	Max	Unit
Supply Voltage	V_{CC}*	4.75	5.0	5.25	V
Input "Low" Level Voltage	V_{IL}*	−0.3	–	0.8	V
Input "High" Level Voltage	V_{IH}*	2.0	–	V_{CC}	V
Operating Temperature	T_{opr}	−20	25	75	°C

* This value is in reference to V_{SS} = 0V.

■ ELECTRICAL CHARACTERISTICS

DC CHARACTERISTICS (V_{CC} = 5.0V±5%, V_{SS} = 0V, Ta = −20 ~ +75°C unless otherwise noted)

Item		Symbol	Measuring Condition	Min	Typ*	Max	Unit
Input "High" Level Voltage		V_{IH}		2.0	–	V_{CC}	V
Input "Low" Level Voltage		V_{IL}		−0.3	–	0.8	V
Input Leak Current	Inputs except D_0 – D_7	I_{in}	V_{in} = 0 ~ 5.25 V	−2.5	–	2.5	μA
Three State (Off State) Input Current	D_0 – D_7 Memory Address Raster Address	I_{TSI}	V_{in} = 0.4 ~ 2.4 V V_{CC} = 5.25 V	-10	–	10	μA
Output "High" Level Voltage	D_0 – D_7	V_{OH}	I = −205 μA	2.4	–	–	V
	Others		I = −100 μA				
Output "Low" Level Voltage		V_{OL}	I = 1.6 mA	–	–	0.4	V
Input Capacity	D_0 – D_7 EXVSYNC EXHSYNC	C_{in}	V_{in} = 0 V Ta = 25°C f = 1.0 MHz	–	–	12.5	pF
	Others			–	–	10.0	pF
Output Capacity		C_{out}	V_{in} = 0 V Ta = 25°C f = 1 MHz	–	–	10.0	pF
Power Dissipation		P_D	f_{CLK} = 4.5 MHz f_E = 2 MHz V_{CC} = max, No Load V_{IH} = V_{CC} − 1.0 V V_{IL} = 0.8 V	–	50	–	mW

* Ta = 25°C, Vcc = 5.0V

AC CHARACTERISTICS (V_{CC} = 5V±10%, V_{SS} = 0V, Ta = −20 ~ +75°C, unless otherwise noted.)

1. TIMING OF CRTC SIGNAL

Item	Symbol	Test Condition	Min	Typ	Max	Unit
Clock Cycle Time	t_{cycC}	Fig. 30	220	–	–	ns
Clock "High" Pulse Width	PW_{CH}		100	–	–	ns
Clock "Low" Pulse Width	PW_{CL}		100	–	–	ns
Rise and Fall Time for Clock Input	tcr, tcf		–	–	20	ns
Memory Address Delay Time	t_{MAD}		–	–	80	ns
Raster Address Delay Time	t_{RAD}		–	–	80	ns
DISPTMG Delay Time	t_{DTD}		–	–	120	ns
CUDISP Delay Time	t_{CDD}		–	–	120	ns
Horizontal Sync Delay Time	t_{HSD}		–	–	100	ns
Vertical Sync Delay Time	t_{VSD}		–	–	120	ns
Light Pen Strobe Pulse Width	PW_{LPH}		60	–	–	ns
Light Pen Storobe Uncertain Time of Acceptance	t_{LPD1}	Fig. 32, 33	–	–	70	ns
	t_{LPD2}		–	–	0	ns
Memory Address Three-State Off Time	t_{MAZ}	Fig. 31	–	–	50	ns
Raster Address Three-State Off Time	t_{RAZ}		–	–	50	ns

2. EXTERNAL SYNC TIMING

Item	Symbol	Test Condition	Min	Typ	Max	Unit
Clock Halt Time	t_{CLKST}	Fig. 34	100	–	–	ns
External Horizontal Sync Pulse Width	t_{PWHS}		1000	–	–	ns
External Horizontal Sync Rise and Fall Time	t_{Hr}		–	–	20	ns
	t_{Hf}		–	–	20	ns
External Vertical Sync Pulse Width*	t_{PWVS}		1660	–	–	ns
External Vertical Sync Rise and Fall Time	t_{Vr}		–	–	20	ns
	t_{Vf}		–	–	20	ns

* : External Vertical Sync Pulse Width t_{PWVS} = 1000 ns + 3·t_{cyc} C

3. MPU BUS TIMING

Item	Symbol	Test Condition	6345		63A45		63B45		Unit
			Min	Max	Min	Max	Min	Max	
Enable Cycle Time	t_{cycE}	Fig. 35 Fig. 36	1000	–	666	–	500	–	ns
Enable "High" Pulse Width	PW_{EH}		450	–	280	–	220	–	ns
Enable "Low" Pulse Width	PW_{EL}		400	–	280	–	210	–	ns
Enable Rise and Fall Time	t_{Er}, t_{Ef}		–	20	–	20	–	20	ns
Address Set Up Time	t_{AS}		80	–	80	–	40	–	ns
Data Set Up Time	t_{DSW}		195	–	80	–	60	–	ns
Data Delay Time	t_{DDR}		–	200	–	140	–	120	ns
Data Hold Time	t_H		10	–	10	–	10	–	ns
Address Hold Time	t_{AH}		10	–	–	–	10	–	ns
Data Access Time	t_{ACC}		–	280	–	220	–	160	ns
Input Signal Rise and Fall Time ($\overline{RES}$, LPSTB, RS, $\overline{CS}$, R/$\overline{W}$)	tr, tf		–	100	–	100	–	100	ns

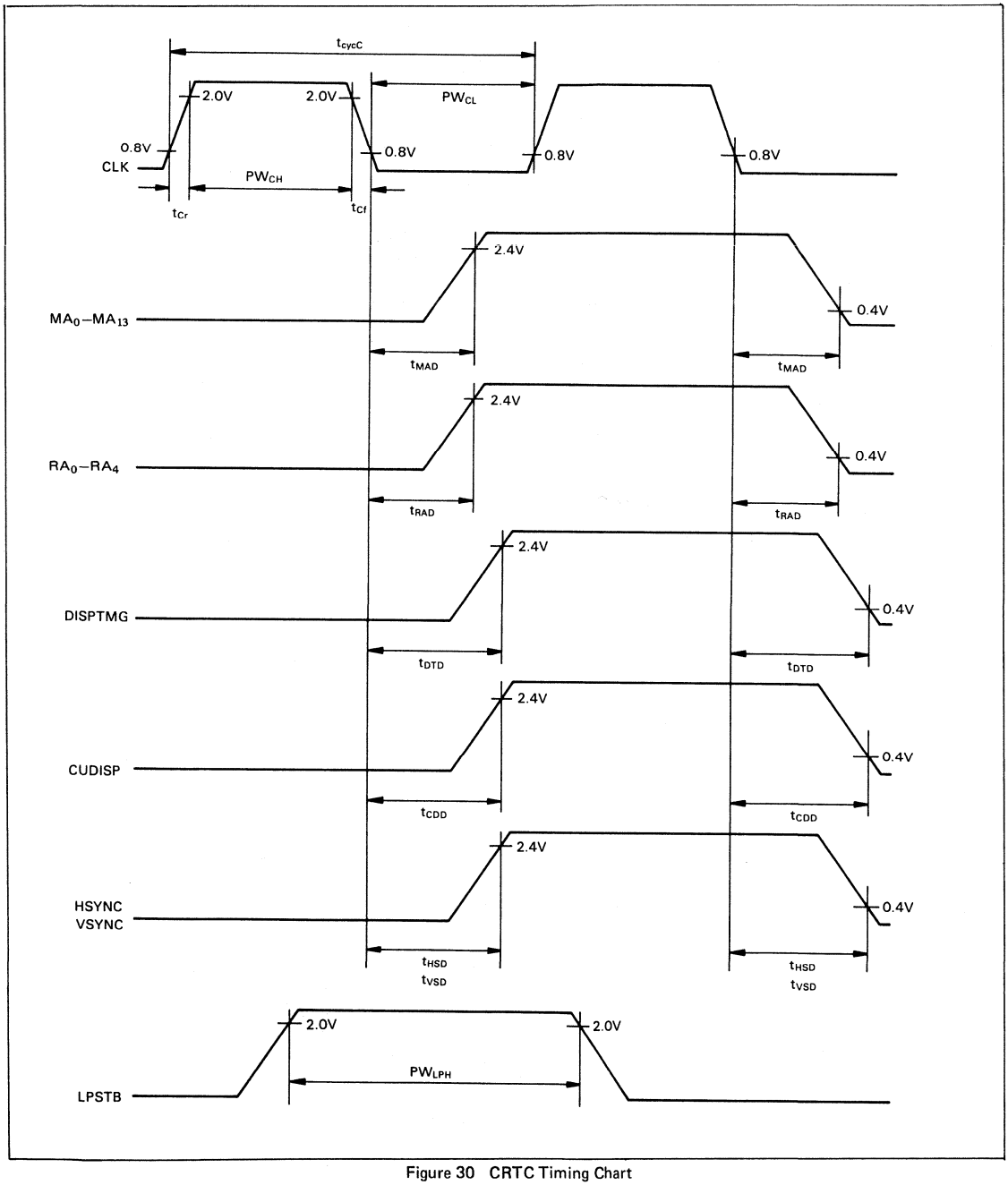
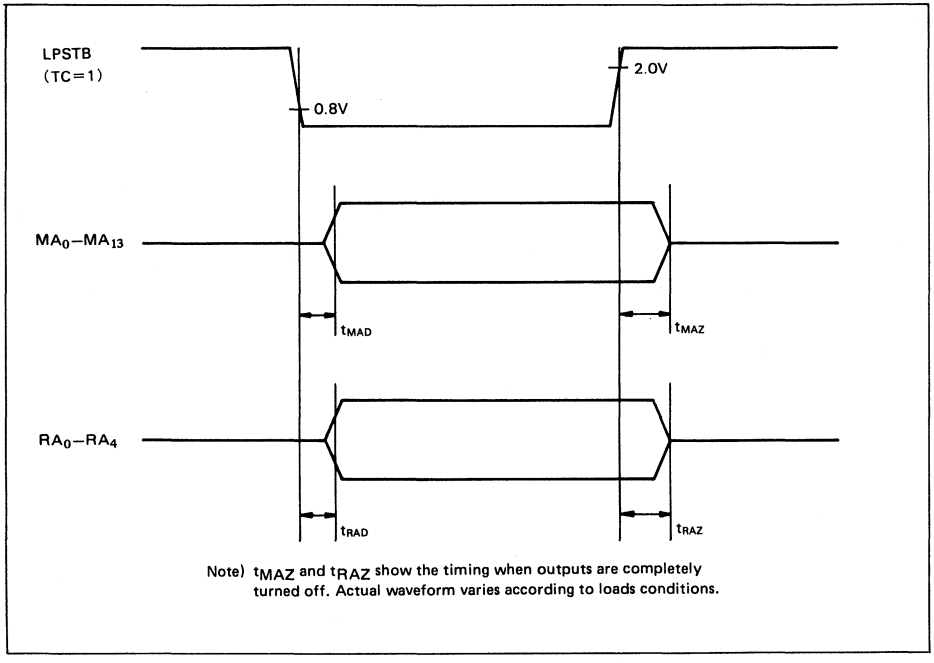


Figure 30 CRTC Timing Chart



Note) t_{MAZ} and t_{RAZ} show the timing when outputs are completely turned off. Actual waveform varies according to loads conditions.

Figure 31 Three-State Delay Timing (Three-state mode: TC = 1)

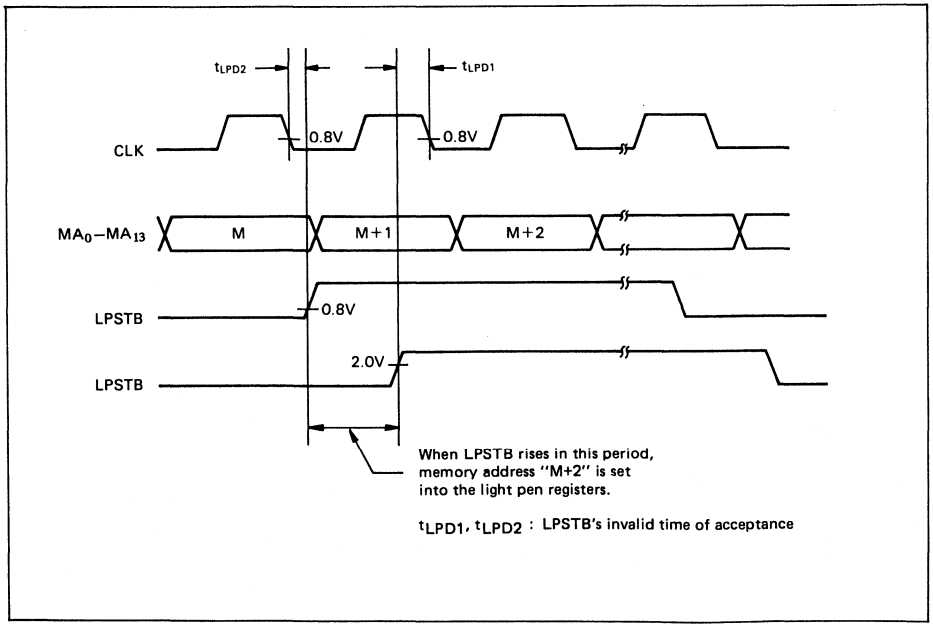


Figure 32 CRTC-CLK, MA0–MA13, and LPSTB Timing

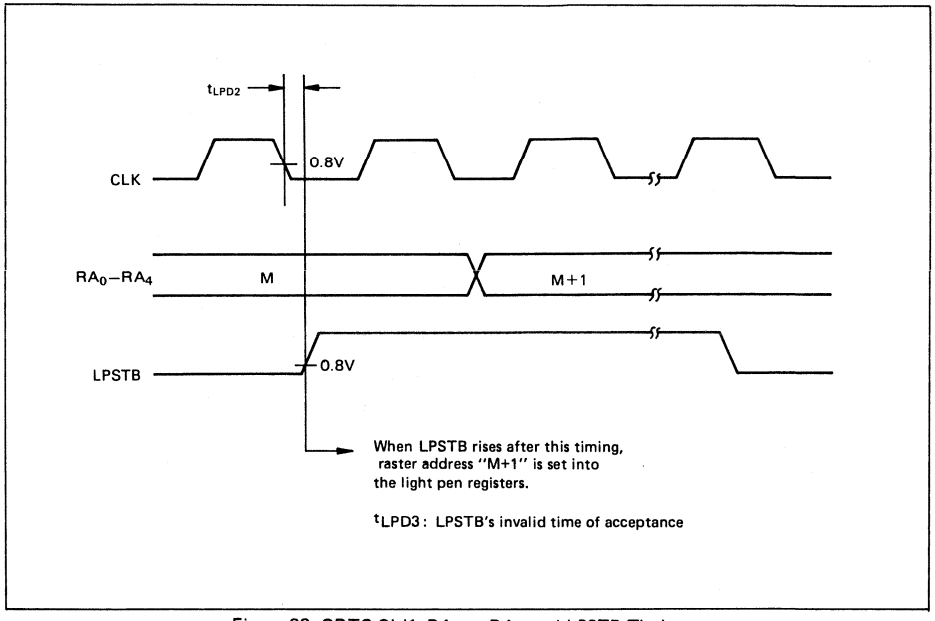


Figure 33 CRTC-CLK, RA_0 – RA_4 and LPSTB Timing

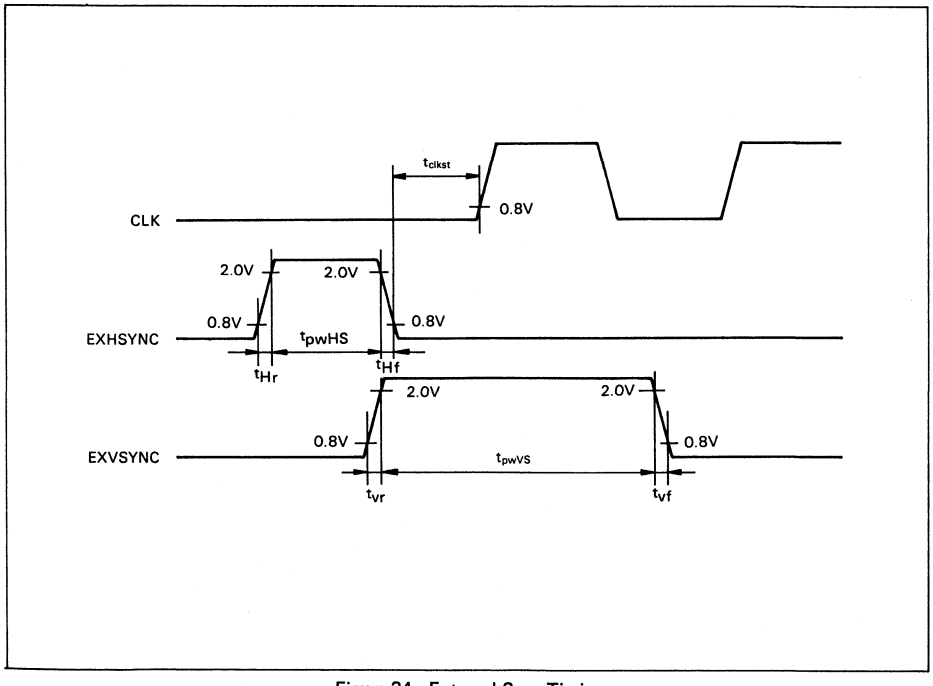


Figure 34 External Sync Timing

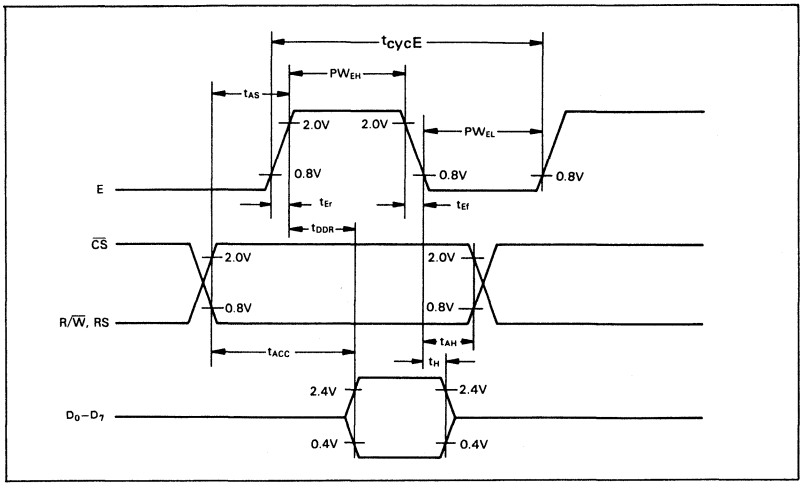


Figure 35 Read Sequence

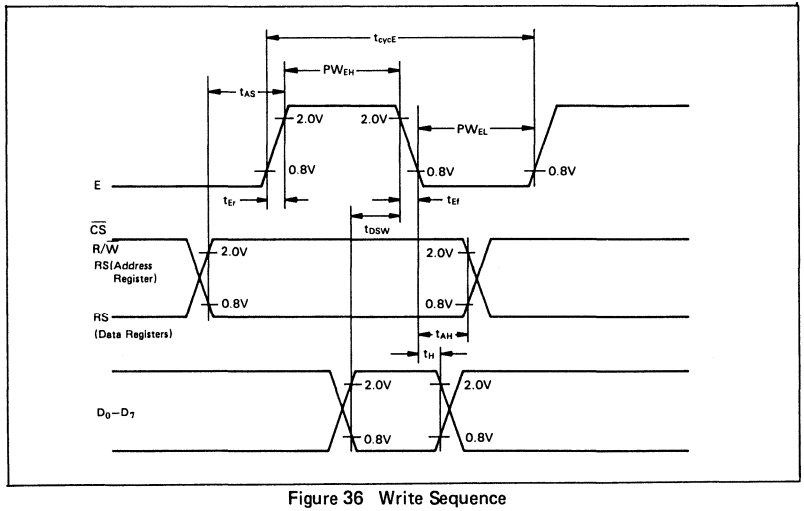


Figure 36 Write Sequence

TEST LOAD

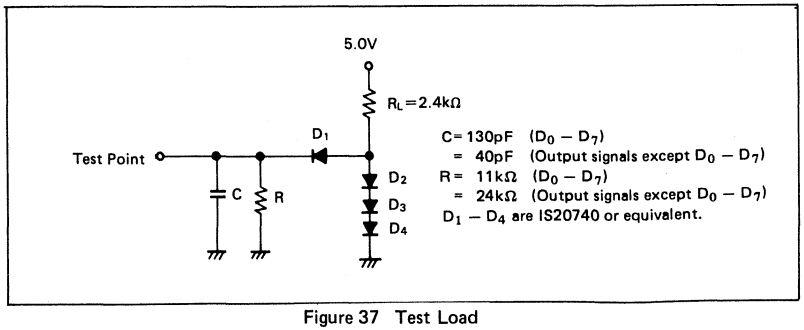


Figure 37 Test Load

■ COMPARISON BETWEEN HD6345 AND HD6845S

HD6345	HD6845S
CLK: 4.5 MHz CMOS 6800 System Bus Interface	CLK: 3.7 MHz NMOS 6800 System Bus Interface
1. Refresh Memory Address (16k words)	1. Refresh Memory Address (16k words)
2. Paging, Scrolling	2. Paging, Scrolling
3. Light Pen	3. Light Pen
4. TTL Compatible	4. TTL Compatible
5. Software Programmable: Number of displayed characters on screen Number of rasters per character row Horizontal/Vertical sync signal Raster scanning mode Cursor	5. Software Programmable: Number of displayed characters on screen Number of rasters per character row Horizontal/Vertical sync signal Raster scanning mode Cursor
6. Screen Split (Up to 4)	
7. Smooth Scrolling	
8. External Synchronization	
9. Interrupt Request	
10. Raster Interpolation	
11. Sync Position Adjustment	
12. Light Pen Raster Address	
13. Second Cursor	
14. Display Memory Width Setting	
15. Up to 256 Character Rows	
16. Timing Signal for Dual Port RAM	
17. Three-State Control of Memory Address and Raster Address	

■ CHARACTERISTICS DIFFERENCES BETWEEN HD6345 AND HD6845S

No.	Item	Symbol	HD6345			HD6845S			Unit
			Min	Typ	Max	Min	Typ	Max	
1	Power Dissipation	P_D	–	50	–	–	600	1000	mW
2	Clock Cycle Time	$tcyc_C$	220	–	–	270	–	–	ns
3	Clock "High" Pulse Width	PW_{CH}	100	–	–	130	–	–	ns
4	Clock "Low" Pulse Width	PW_{CL}	100	–	–	130	–	–	ns
5	Memory Address Delay Time	t_{MAD}	–	–	80	–	–	160	ns
6	Raster Address Delay Time	t_{RAD}	–	–	80	–	–	160	ns
7	Display Timing Delay Time	t_{DTD}	–	–	120	–	–	250	ns
8	Horizontal Sync Delay Time	t_{HSD}	–	–	100	–	–	200	ns
9	Vertical Sync Delay Time	t_{VSD}	–	–	120	–	–	250	ns
10	Cursor Display Delay Time	t_{CDD}	–	–	120	–	–	250	ns
11	Enable Cycle Time	$tcyc_E$	500	–	–	1000	–	–	ns
12	Enable "High" Pulse Width	PW_{EH}	220	–	–	450	–	–	ns
13	Enable "Low" Pulse Width	PW_{EL}	210	–	–	400	–	–	ns
14	Enable Rise and Fall Time	t_{Er}, t_{Ef}	–	–	20	–	–	–	ns
15	Address Set Up Time	t_{AS}	40	–	–	140	–	–	ns
16	Data Set Up Time	t_{DSW}	60	–	–	195	–	–	ns
17	Data Delay Time	t_{DDR}	–	–	120	–	–	320	ns
18	Data Access Time	t_{ACC}	–	–	160	–	–	460	ns
19	Input Signal Rise and Fall Time	t_r, t_f	–	–	100	–	–	–	ns

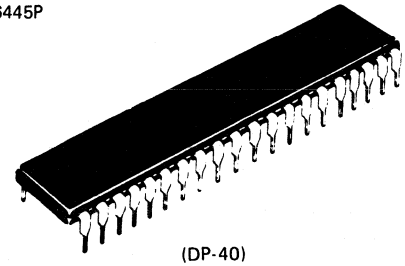
HD6445
CRTC-II (CRT Controller)

—PRELIMINARY—

The HD6445 CRTC-II provides an interface between MPU and a raster scan CRT display. It is upward-compatible with the NMOS CRTC HD6845S in software. A power dissipation is lowered by adopting the CMOS process.

The HD6445 offers a variety of functions under MPU control, such as programmable timing signal outputs for CRT monitor and display screen control operation. It can be widely applied to the various types of CRT display systems.

HD6445P



(DP-40)

■ FEATURES

FLEXIBLE SCREEN FORMAT
- Programmable numbers of characters per screen and rasters per character row
- Programmable horizontal/vertical sync signals and display timing signals
- Up to 16k words refresh memory (14-bit) addressable
- Programmable raster scanning modes: non-interlace, interlace sync, or interlace sync and video modes
- Up to 256 character rows per field
- High-speed display operation at 4.5 MHz character clock
- Double-size vertical display by raster interpolation

VERSATILE DISPLAY FUNCTIONS
- Screen split (max.4 screens configurable, horizontally)
- Paging and scrolling for each screen
- Smooth scrolling
- Two cursors with programmable width
- Programmable refresh memory width

FACILITATED SYSTEM CONFIGURATION
- 80 system bus interface
- Three-state control of memory address and raster address
- External synchronization in master-slave or TV sync modes
- Interrupt request by vertical blanking or light pen strobe detection
- Programmable timing signal for dual-port RAM in DPRAM mode

SOFTWARE UPWARD-COMPATIBLE WITH HD6845S

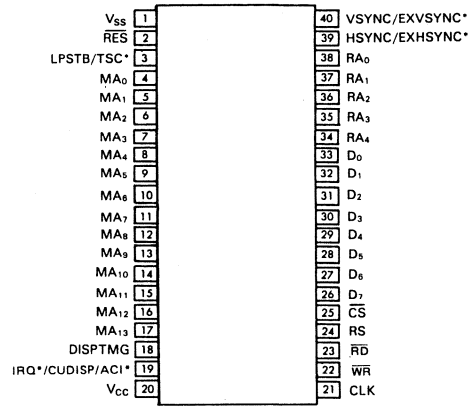
SINGLE +5 V POWER SUPPLY

CMOS PROCESS

■ TYPE OF PRODUCTS

Type No.	Bus Timing	CRT Display Timing
HD6445–4	4.0 MHz	4.5 MHz Max.

■ PIN ARRANGEMENT



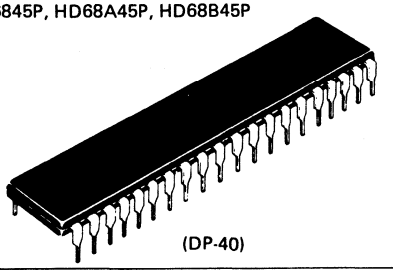
(Top View)

* Providing additional functions to the HD6845S.

HD6845, HD68A45, HD68B45
CRTC (CRT Controller)

The CRTC is a LSI controller which is designed to provide an interface for microcomputers to raster scan type CRT displays. The CRTC belongs to the HMCS6800 LSI Family and has full compatibility with MPU in both data lines and control lines. Its primary function is to generate timing signal which is necessary for raster scan type CRT display according to the specification programmed by MPU. The CRTC is also designed as a programmable controller, so applicable to wide-range CRT display from small low-functioning character display up to raster type full graphic display as well as large high-functioning limited graphic display.

HD6845P, HD68A45P, HD68B45P

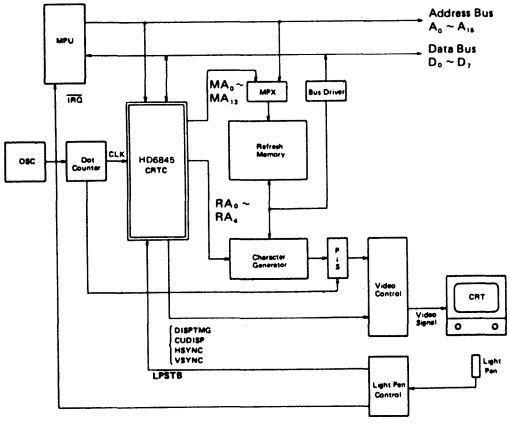


(DP-40)

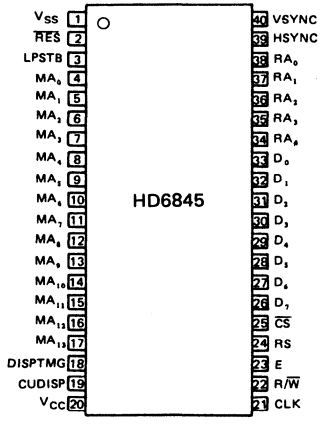
■ FEATURES

- Number of Displayed Characters on the Screen, Vertical Dot Format of One Character, Horizontal and Vertical Sync Signal, Display Timing Signal are Programmable
- Line Buffer-less Refreshing
- 14-bit Refresh Memory Address Output (16k Words max. Access)
- Programmable Interlace/Non-interlace Scan Mode
- Built-in Cursor Control Function
- Programmable Cursor Height and its Blink
- Built-in Light Pen Detection Function
- Paging and Scrolling Capability
- TTL Compatible
- Single +5V Power Supply
- Compatible with MC6845, MC68A45, MC68B45

■ SYSTEM BLOCK DIAGRAM



■ PIN ARRANGEMENT



(Top View)

■ ORDERING INFORMATION

CRTC	Bus Timing	CRT Display Timing
HD6845	1.0 MHz	3.0 MHz max.
HD68A45	1.5 MHz	
HD68B45	2.0 MHz	

■ ABSOLUTE MAXIMUM RATINGS

Item	Symbol	Value	Unit
Supply Voltage	V_{CC} *	-0.3 ~ +7.0	V
Input Voltage	V_{in} *	-0.3 ~ +7.0	V
Operating Temperature	T_{opr}	- 20 ~ + 75	°C
Storage Temperature	T_{stg}	- 55 ~ +150	°C

* With respect to V_{SS} (SYSTEM GND)

[NOTE] Permanent LSI damage may occur if maximum ratings are exceeded. Normal operation should be under recommended operating conditions. If these conditions are exceeded, it could affect reliability of LSI.

■ RECOMMENDED OPERATING CONDITIONS

Item	Symbol	min	typ	max	Unit
Supply Voltage	V_{CC} *	4.75	5.0	5.25	V
Input Voltage	V_{IL} *	-0.3	–	0.8	V
	V_{IH} *	2.0	–	V_{CC}	V
Operating Temperature	T_{opr}	- 20	25	75	°C

* With respect to V_{SS} (SYSTEM GND)

■ ELECTRICAL CHARACTERISTICS

• DC CHARACTERISTICS (V_{CC} = 5V ± 5%, V_{SS} = 0V, Ta = -20~+75°C, unless otherwise noted.)

Item	Symbol	Test Condition		min	typ*	max	Unit
Input "High" Voltage	V_{IH}			2.0	–	V_{CC}	V
Input "Low" Voltage	V_{IL}			-0.3	–	0.8	V
Input Leakage Current	I_{In}	V_{in} = 0 ~ 5.25V (Except D_0~D_7)		-2.5	–	2.5	μA
Three-State Input Current (off-state)	I_{TSI}	V_{in} = 0.4 ~ 2.4V V_{CC} = 5.25V (D_0~ D_7)		- 10	–	10	μA
Output "High" Voltage	V_{OH}	I_{LOAD} = -205 μA (D_0~D_7)		2.4	–	–	V
		I_{LOAD} = -100 μA (Other Outputs)					
Output "Low" Voltage	V_{OL}	I_{LOAD} = 1.6 mA		–	–	0.4	V
Input Capacitance	C_{In}	V_{in} = 0 Ta = 25°C f = 1.0 MHz	D_0 ~ D_7	–	–	12.5	pF
			Other Inputs	–	–	10.0	pF
Output Capacitance	C_{out}	V_{in} = 0V, Ta = 25°C, f = 1.0 MHz		–	–	10.0	pF
Power Dissipation	P_D			–	600	1000	mW

* Ta = 25°C, V_{CC} = 5.0V

- **AC CHARACTERISTICS (V_{CC} = 5V ±5%, V_{SS} = 0V, Ta = -20~+75°C, unless otherwise noted.)**

1. TIMING OF CRTC SIGNAL

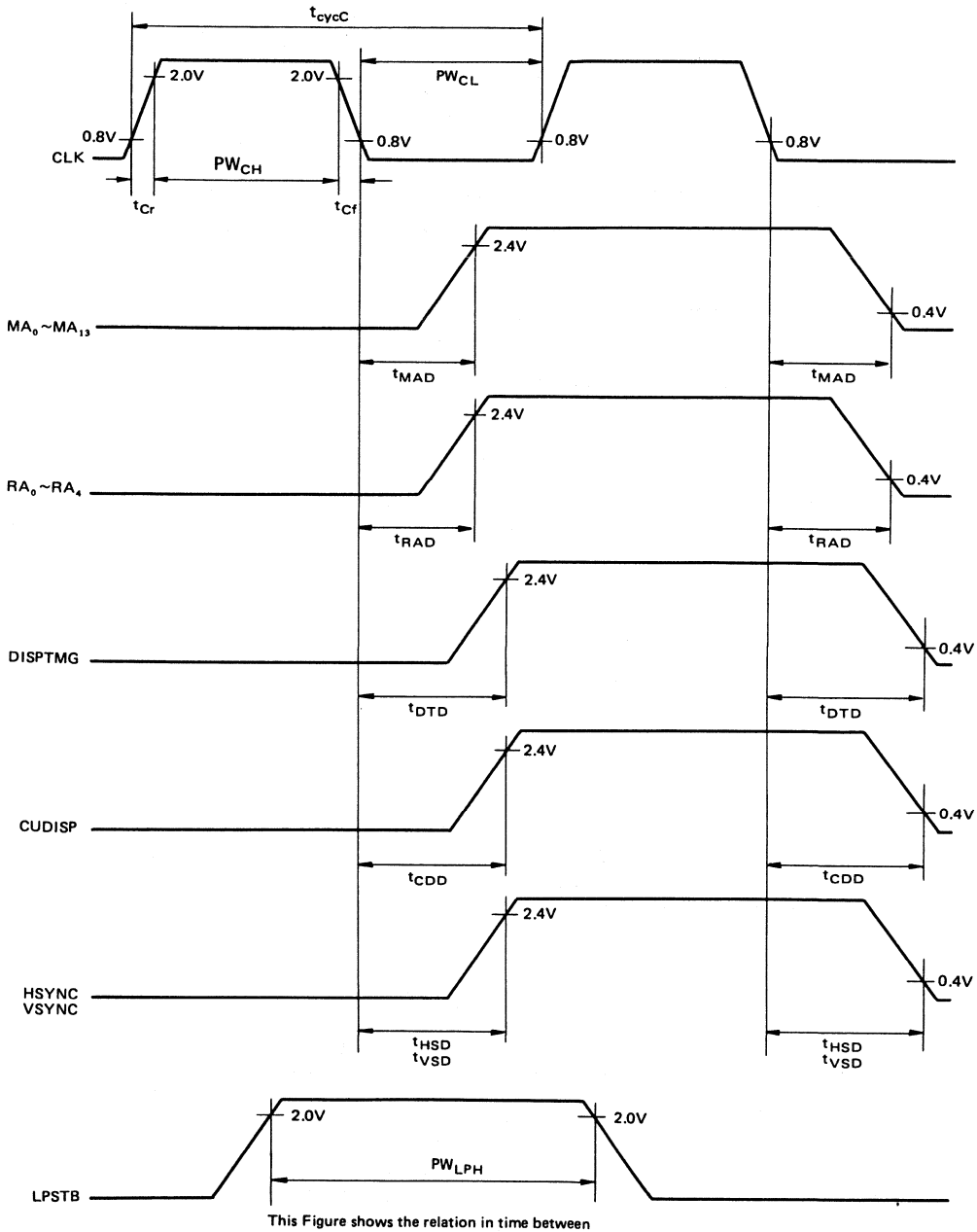
Item	Symbol	Test Condition	min	typ	max	Unit
Clock Cycle Time	t_{cycC}	Fig. 1	330	–	–	ns
Clock "High" Pulse Width	PW_{CH}		150	–	–	ns
Clock "Low" Pulse Width	PW_{CL}		150	–	–	ns
Rise andFall Time for Clock Input	t_{Cr}, t_{Cf}		–	–	15	ns
Memory Address Delay Time	t_{MAD}		–	–	160	ns
Raster Address Delay Time	t_{RAD}		–	–	160	ns
DISPTMG Delay Time	t_{DTD}		–	–	250	ns
CUDISP Delay Time	t_{CDD}		–	–	250	ns
Horizontal Sync Delay Time	t_{HSD}		–	–	250	ns
Vertical Sync Delay Time	t_{VSD}		–	–	250	ns
Light Pen Strobe Pulse Width	PW_{LPH}		80	–	–	ns
Light Pen Strobe	t_{LPD1}	Fig. 2	–	–	80	ns
Uncertain Time of Acceptance	t_{LPD2}		–	–	10	ns

2. MPU READ TIMING

Item	Symbol	Test Condition	HD6845			HD68A45			HD68B45			Unit
			min	typ	max	min	typ	max	min	typ	max	
Enable Cycle Time	t_{cycE}	Fig. 3	1.0	–	–	0.666	–	–	0.5	–	–	μs
Enable "High" Pulse Width	PW_{EH}		0.45	–	–	0.28	–	–	0.22	–	–	μs
Enable "Low" Pulse Width	PW_{EL}		0.40	–	–	0.28	–	–	0.21	–	–	μs
Enable Rise and Fall Time	t_{Er}, t_{Ef}		–	–	25	–	–	25	–	–	25	ns
Address Set Up Time	t_{AS}		140	–	–	140	–	–	70	–	–	ns
Data Delay Time	t_{DDR}		–	–	320	–	–	220	–	–	180	ns
Data Hold Time	t_{H}		10	–	–	10	–	–	10	–	–	ns
Address Hold Time	t_{AH}		10	–	–	10	–	–	10	–	–	ns
Data Access Time	t_{ACC}		–	–	460	–	–	360	–	–	250	ns

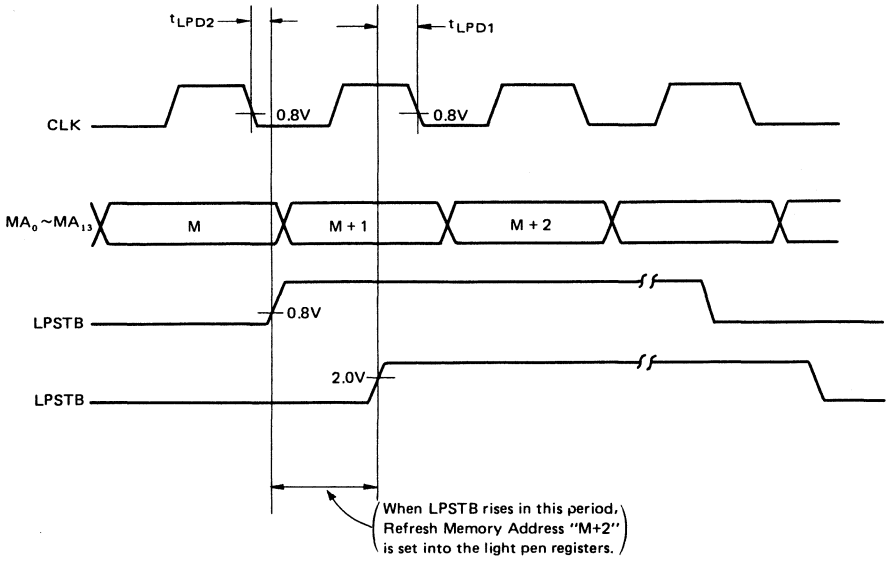
3. MPU WRITE TIMING

Item	Symbol	Test Condition	HD6845			HD68A45			HD68B45			Unit
			min	typ	max	min	typ	max	min	typ	max	
Enable Cycle Time	t_{cycE}	Fig. 4	1.0	–	–	0.666	–	–	0.5	–	–	μs
Enable "High" Pulse Width	PW_{EH}		0.45	–	–	0.28	–	–	0.22	–	–	μs
Enable "Low" Pulse Width	PW_{EL}		0.40	–	–	0.28	–	–	0.21	–	–	μs
Enable Rise and Fall Time	t_{Er}, t_{Ef}		–	–	25	–	–	25	–	–	25	ns
Address Set Up Time	t_{AS}		140	–	–	140	–	–	70	–	–	ns
Data Set Up Time	t_{DSW}		195	–	–	80	–	–	60	–	–	ns
Data Hold Time	t_{H}		10	–	–	10	–	–	10	–	–	ns
Address Hold Time	t_{AH}		10	–	–	10	–	–	10	–	–	ns



This Figure shows the relation in time between CLK signal and each output signals. Output sequence is shown in Figs. 10~15.

Figure 1 Time Chart of the CRTC



t_{LPD1}, t_{LPD2} : LPSTB's uncertain time of acceptance.

Figure 2 LPSTB Input Timing & Refresh Memory Address that is set into the light pen registers.

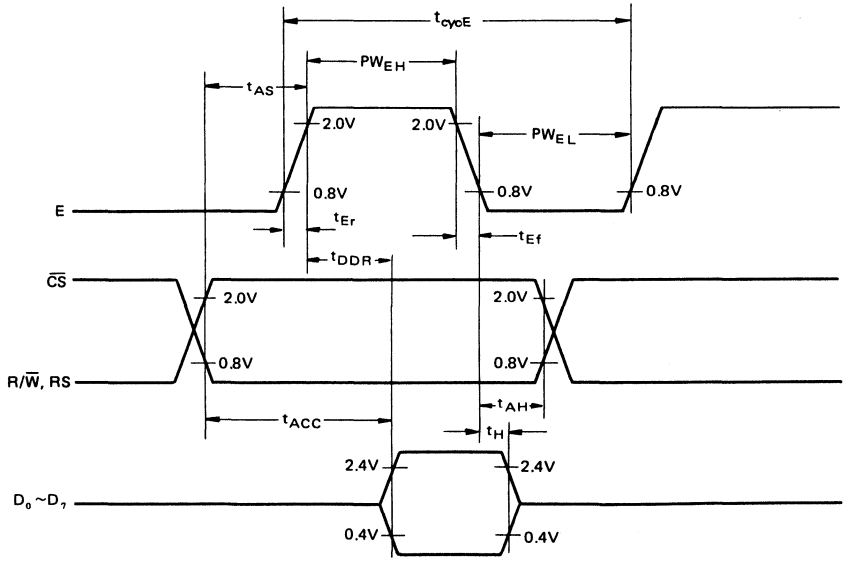


Figure 3 Read Sequence

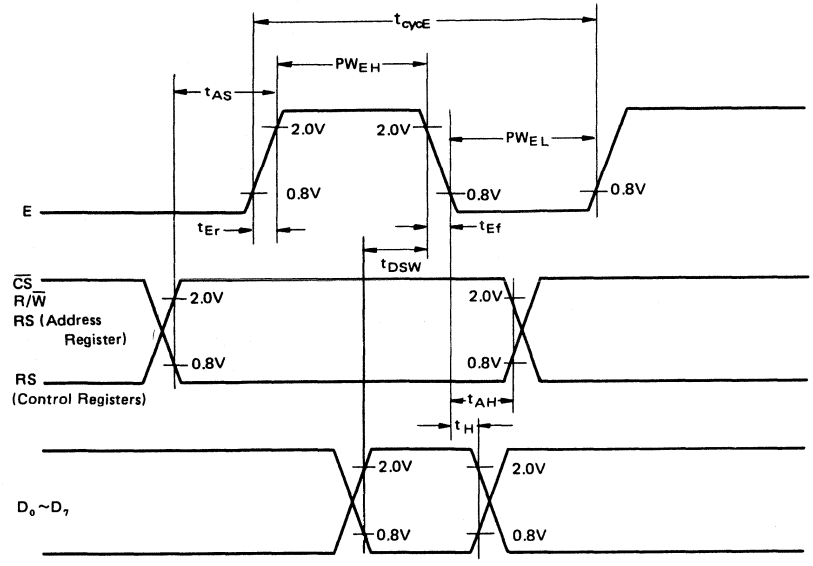
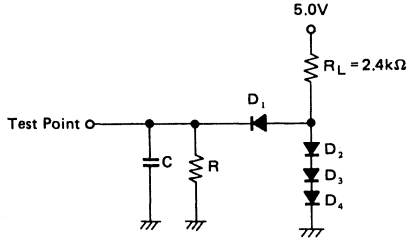


Figure 4 Write Sequence



C = 130pF (D_0~D_7)
= 30pF (Output signals except D_0~D_7)
R = 11kΩ (D_0~D_7)
= 24kΩ (Output signals except D_0~D_7)
D_1~D_4 are 1S2074Ⓗ or equivalent.

Figure 5 Test Loads

■ SYSTEM DESCRIPTION

The CRTC is a LSI which is connected with MPU and CRT display device to control CRT display. The CRTC consists of internal register group, horizontal and vertical timing circuits, linear address generator, cursor control circuit, and light pen detection circuit. Horizontal and vertical timing circuit generate RA_0~RA_4, DISPTMG, HSYNC, and VSYNC. RA_0~RA_4 are raster address signals and used as input signals for Character Generator. DISPTMG, HSYNC, and VSYNC signals are received by video control circuit. This horizontal and vertical timing circuit consists of internal counter and comparator circuit. Linear address generator generates refresh memory address MA_0 ~MA_{13} to be used for refreshing the screen. By these address signals, refresh memory is accessed periodically. As 14 refresh memory address signals are prepared, 16k words max are accessible. Moreover, the use of start address register enables paging and scrolling. Light pen detection circuit detects light pen position on the screen. When light pen strobe signal is received, light pen register memorizes linear address generated by linear address generator in order to memorize where light pen is on the screen. Cursor control circuit controls the position of cursor, its height, and its blink.

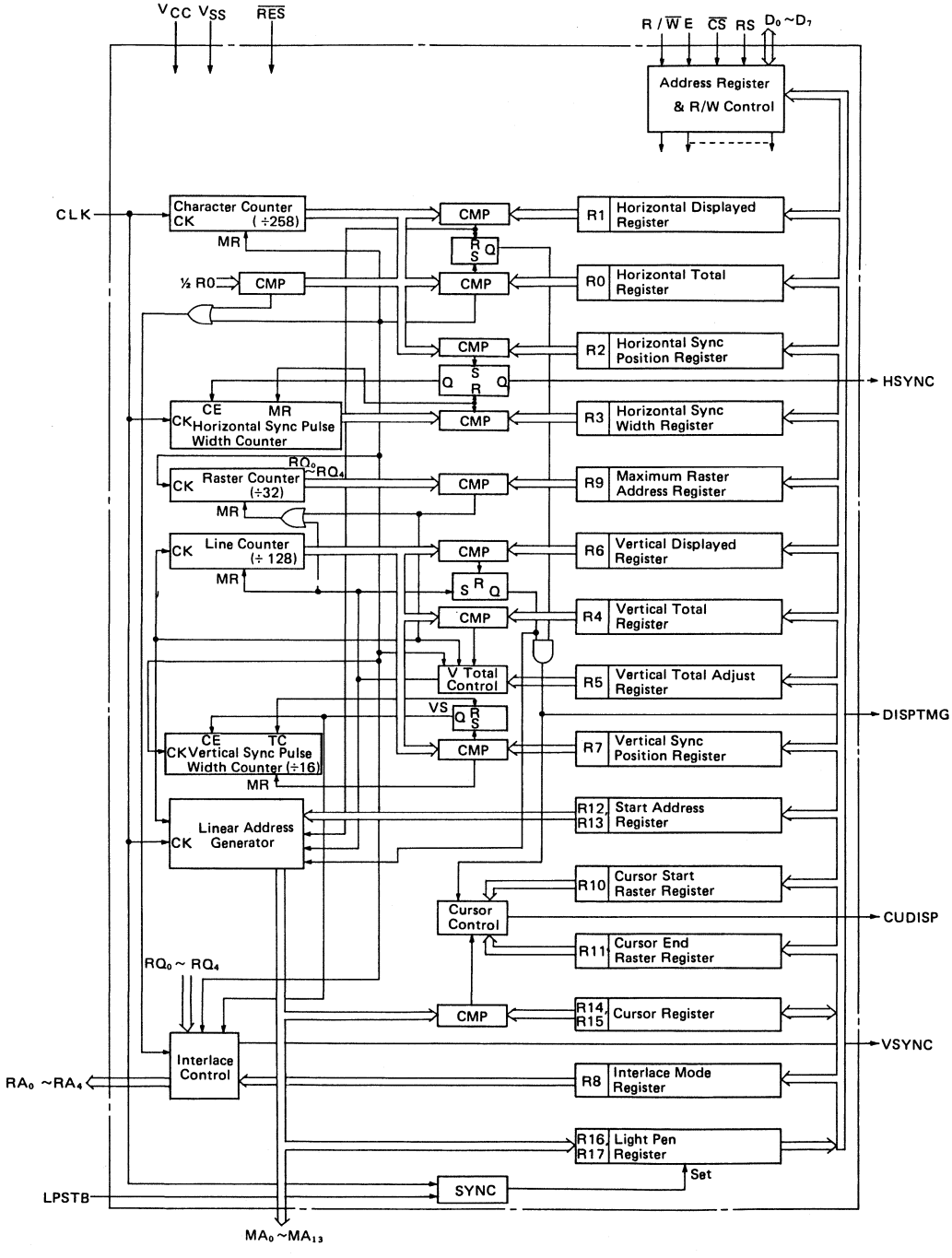


Figure 6 Internal Block Diagram of the CRTC

■ FUNCTION OF SIGNAL LINE

The CRTC provides 13 interface signals to MPU and 25 interface signals to CRT display.

• Interface Signals to MPU

Bi-directional Data Bus (D_0~D_7)

Bi-directional data bus(D_0~D_7) are used for data transfer between the CRTC and MPU. The data bus outputs are 3-state buffers and remain in the high-impedance state except when MPU performs a CRTC read operation.

Read/Write (R/$\overline{W}$)

Read/Write signal (R/$\overline{W}$) controls the direction of data transfer between the CRTC and MPU. When R/$\overline{W}$ is at "High" level, data of CRTC is transfered to MPU. When R/$\overline{W}$ is at "Low" level, data of MPU is transfered to CRTC.

Chip Select ($\overline{CS}$)

Chip Select signal ($\overline{CS}$) is used to address the CRTC. When $\overline{CS}$ is at "Low" level, it enables Read/Write operation to CRTC internal registers. Normally this signal is derived from decoded address signal of MPU under the condition that VMA of MPU is at "High" level.

Register Select (RS)

Register Select signal (RS) is used to select the address register and 18 control registers of the CRTC. When RS is at "Low" level, the address register is selected and when RS is at "High" level, control registers are selected. This signal is normally a derivative of the lowest bit (A0) of MPU address bus.

Enable(E)

Enable signal (E) is used as strobe signal in MPU Read/Write operation with the CRTC internal registers. This signal is normally a derivative of the HMCS6800 System ϕ_2 clock.

Reset ($\overline{RES}$)

Reset signal ($\overline{RES}$) is an input signal used to reset the CRTC. When $\overline{RES}$ is at "Low" level, it forces the CRTC into the following status.

1) All the counters in the CRTC are cleared and the device stops the display operation*.
2) All the outputs go down to "Low" level. (D_0~D_7 are not affected.)
3) Control registers in the CRTC are not affected and remain unchanged.

The CRTC internal registers can be accessed by MPU even in reset status.

This signal is different from other HMCS6800 family LSIs in the following functions and has restrictions for usage.

1) $\overline{RES}$ has capability of reset function only when LPSTB is at "Low" level.
2) The CRTC starts the display operation immediately after $\overline{RES}$ goes "High" level.
3) After $\overline{RES}$ has gone down to "Low" level, MAo ~ MA_{13} and RAo ~ RA_4, synchronizing with CLK "Low" level, go down to "Low" level.
   (At least 1 cycle CLK is necessary to reset.)
   *In the case "0" is set to Horizontal Sync Position Register (R2), CLK signal is output from HSYNC pin.

• Interface Signals to CRT Display Device

Character Clock (CLK)

CLK is a standard clock input signal which defines character timing for the CRTC display operation. CLK is normally derived from the external high-speed dot timing logic.

Horizontal Sync (HSYNC)

HSYNC is an active "High" level signal which provides horizontal synchronization for display device.

Vertical Sync (VSYNC)

VSYNC is an active "High" level signal which provides vertical synchronization for display device.

The pulse width is fixed at 16H.

Display Timing (DISPTMG)

DISPTMG is an active "High" level signal which defines the display period in horizontal and vertical raster scanning. It is necessary to enable video signal only when DISPTMG is at "High" level.

Refresh Memory Address (MA_0~MA_{13})

MA_0~MA_{13} are refresh memory address signals which are used to access to refresh memory in order to refresh the CRT screen periodically. These outputs enables 16k words max. refresh memory access. So, for instance, these are applicable up to 2000 characters/screen and 8-page system.

Raster Address (RA_0~RA_4)

RA_0~RA_4 are raster address signals which are used to select the raster of the character generator or graphic pattern generator etc.

Cursor Display (CUDISP)

CUDISP is an active "High" level video signal which is used to display the cursor on the CRT screen. This output is inhibited while DISPTMG is at "Low" level. Normally this output is mixed with video signal and provided to the CRT display device.

Light Pen Strobe (LPSTB)

LPSTB is an active "High" level input signal which accepts strobe pulse detected by the light pen and control circuit. When this signal is activated, the refresh memory address (MA_0~MA_{13}) which are shown in Fig. 2 are stored in the 14-bit light pen register. The stored refresh memory address need to be corrected in software, taking the delay time of the display device, light pen, and light pen control circuits into account.

■ REGISTER DESCRIPTION

Table 1 Internal Registers Assignment

$\overline{CS}$	RS	Address Register 4	3	2	1	0	Register #	Register Name	Program Unit	READ	WRITE	Data Bit 7	6	5	4	3	2	1	0
1	×	×	×	×	×	×			–	–	–								
0	0	×	×	×	×	×	AR	Address Register	–	×	O								
0	1	0	0	0	0	0	R0	Horizontal Total *	Character	×	O								
0	1	0	0	0	0	1	R1	Horizontal Displayed	Character	×	O								
0	1	0	0	0	1	0	R2	Horizontal Sync* Position	Character	×	O								
0	1	0	0	0	1	1	R3	Horizontal Sync Width	Character	×	O								
0	1	0	0	1	0	0	R4	Vertical Total *	Line	×	O								
0	1	0	0	1	0	1	R5	Vertical Total Adjust	Raster	×	O								
0	1	0	0	1	1	0	R6	Vertical Displayed	Line	×	O								
0	1	0	0	1	1	1	R7	Vertical Sync * Position	Line	×	O								
0	1	0	1	0	0	0	R8	Interlace mode	–	×	O							V	S
0	1	0	1	0	0	1	R9	Maximum Raster Address	Raster	×	O								
0	1	0	1	0	1	0	R10	Cursor Start Raster	Raster	×	O		B	P					
0	1	0	1	0	1	1	R11	Cursor End Raster	Raster	×	O								
0	1	0	1	1	0	0	R12	Start Address(H)	–	×	O								
0	1	0	1	1	0	1	R13	Start Address(L)	–	×	O								
0	1	0	1	1	1	0	R14	Cursor(H)	–	O	O								
0	1	0	1	1	1	1	R15	Cursor (L)	–	O	O								
0	1	1	0	0	0	0	R16	Light Pen(H)	–	O	×								
0	1	1	0	0	0	1	R17	Light Pen(L)	–	O	×								

(NOTE) 1. The register marked *: (Written Value) = (Specified Value) – 1
2. R6 is programmed to be N/2 (N: total number of lines) in interlace sync & video mode.
3. When S is "1", V specifies video mode. S specifies the interlace sync mode.
4. B specifies the cursor blink. P specifies the cursor blink period.
5. The cursor start and the cursor end raster registers must be both even or both odd in interlace sync & video mode.
6. R0 is ordinally programmed to be odd number in interlace mode.
7. R9 must be programmed to be odd number in interlace sync & video mode.
8. O; Yes, X; No

■ FUNCTION OF INTERNAL REGISTERS

• Address Register (AR)

This is a 5-bit register used to select 18 internal control registers (R0~R17). Its contents are the address of one of 18 internal control registers. Programming the data from 18 to 31 produces no results. Access to R0~R17 requires, first of all, to write the address of corresponding control register into this register. When RS and $\overline{CS}$ are at "Low" level, this register is selected.

• Horizontal Total Register (R0)

This is a register used to program total number of horizontal characters per line including the retrace period. The data is 8-bit and its value should be programmed according to the specification of the CRT. When M is total number of characters, M-1 shall be programmed to this register. When programming for interlace mode, M must be even.

• Horizontal Displayed Register (R1)

This is a register used to program the number of horizontal displayed characters per line. Data is 8-bit and any number that is smaller than that of horizontal total characters can be programmed.

• Horizontal Sync Position Register (R2)

This is a register used to program horizontal sync position as multiples of the character clock period. Data is 8-bit and any number that is under the following condition (horizontal sync position + horizontal pulse width < horizontal total characters) can be programmed. When H is character number of horizontal sync position, H-1 shall be programmed to this register. When programmed value of this register is increased, the display position on the CRT screen is shifted to the left. When programmed value is decreased, the position is shifted to the right. Therefore, the optimum horizontal position can be determined by this value.

• Horizontal Sync Width Register (R3)

This is a register used to program the horizontal sync pulse width. The horizontal sync pulse width is programmed in the lower 4-bit as multiples of the character clock period. "0" can't be programmed.

• Vertical Total Register (R4)

This is a register used to program total number of lines per frame including vertical retrace period. The data is within 7-bit and its value should be programmed according to the specification of the CRTC. When N is total number of lines, N-1 shall be programmed to this register.

• Vertical Total Adjust Register (R5)

This is a register used to program the optimum number to adjust total number of rasters per field. This register enables to decide the number of vertical deflection frequency more strictly.

• Vertical Displayed Register (R6)

This is a register used to program the number of displayed character rows on the CRT screen. Data is 7-bit and any number that is smaller than that of vertical total characters can be programmed.

Table 2 Pulse Width of Horizontal Sync Signal

HSW				Pulse Width
2^3	2^2	2^1	2^0	
0	0	0	0	– (Note)
0	0	0	1	1 CH
0	0	1	0	2
0	0	1	1	3
0	1	0	0	4
0	1	0	1	5
0	1	1	0	6
0	1	1	1	7
1	0	0	0	8
1	0	0	1	9
1	0	1	0	10
1	0	1	1	11
1	1	0	0	12
1	1	0	1	13
1	1	1	0	14
1	1	1	1	15

CH; Character clock period
(Note) HSW = "0" can't be used.

• Vertical Sync Position Register (R7)

This is a register used to program the vertical sync position on the screen as multiples of the horizontal character line period. Data is 7-bit and any number that is equal to or less than vertical total characters can be programmed. When V is character number of vertical sync position, V-1 shall be programmed to this register. When programmed value of this register is increased, the display position is shifted up. When programmed value is decreased, the position is shifted down. Therefore, the optimum vertical position may be determined by this value.

• Interlace mode Register (R8)

This is a register used to program raster scan mode.

Raster Scan Mode Program Bit (V, S)

Raster scan mode is programmed in the V, S bit.

Table 3 Raster Scan Mode (2^1, 2^0)

V	S	Raster Scan Mode
0	0	Non-interlace Mode
1	0	
0	1	Interlace Sync Mode
1	1	Interlace Sync & Video Mode

In the non-interlace mode, the rasters of even number field and old number field are scanned duplicatedly. In the interlace sync mode, the rasters of odd number field are scanned in the middle of even number field. Then it is controlled to display the same character pattern in two fields. In the interlace sync & video mode, the raster scan method is the same as the interlace sync mode, but it is controlled to display different character pattern in two field.

- **Maximum Raster Address Register (R9)**

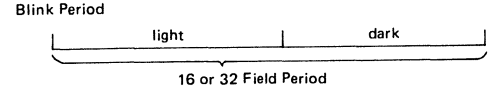
This is a register used to program maximum raster address within 5-bit. This register defines total number of rasters per character including line space. When total number of rasters is RN, RN-1 shall be programmed to this register. Moreover, RN must be even in interlace sync & video mode.

- **Cursor Start Raster Register (R10)**

This is a register used to program the cursor start raster address by lower 5-bit (2^0~2^4) and the cursor display mode by higher 2-bit (2^5, 2^6).

Table 4 Cursor Display Mode (2^6, 2^5)

B	P	Cursor Display Mode
0	0	Non-blink
0	1	Cursor Non-display
1	0	Blink 16 Field Period
1	1	Blink 32 Field Period



- **Cursor End Raster Register (R11)**

This is register used to program the cursor end raster address, both the cursor start raster register and the cursor end raster register must be even or odd in interlace sync & video mode.

- **Start Address Register (R12, R13)**

These are used to program the first address of refresh memory to read out.

Paging and scrolling is easily performed using this register.

- **Cursor Register (R14, R15)**

These two read/write registers stores the cursor location. The higher 2-bit (2^6, 2^7) of R14 are always "0".

- **Light Pen Register (R16, R17)**

These read only registers are used to catch the detection address of the light pen. The higher 2-bit (2^6, 2^7) of R16 are always "0". Its value needs to be corrected by software because there is time delay from address output of the CRTC to signal input LPSTB pin of the CRTC in the process that raster is lit after address output and light pen detects it. Moreover, delay time shown in Fig. 2 needs to be taken into account.

Restriction on Programming Internal Register

i) $0 < Nhd < Nht + 1 \leq 256$
ii) Nht : Odd Number (in the case of interlace sync mode or interlace sync & video mode)
iii) $0 < Nvd < Nvt + 1 \leq 128$
iv) $0 < Nhsp + Nhsw < Nht + 1$
v) $0 \leq Nvsp \leq Nvt$
vi) $0 \leq N_{CSTART} \leq N_{CEND} \leq Nr$
vii) In the case of interlace sync & video mode, either of the followings shall be selected.
  (1) N_{CSTART}, N_{CEND} : Even Number (The cursor is displayed on the even raster.)
  (2) N_{CSTART}, N_{CEND} : Odd Number (The cursor is displayed on the odd raster.)
viii) $0 \leq Nhd \leq Nhsp$

Notes for Use

(1) The method of directly using the value programmed in the internal registers of LSI for controlling the CRT is adopted. Consequently, the display may flicker on the screen when the contents of the registers are changed from bus side asynchronously with the display operation.

Cursor Register

Writing into this register at frequent intervals for moving the cursor should be performed during horizontal and vertical retrace period.

Start Address Register

Writing into the start address register at frequent intervals for scrolling and paging should be performed during horizontal and vertical display period.

It is desirable to avoid programming other registers during display operation.

(2) The $\overline{RES}$ assertion at power-on does not define the internal registers of the HD6845. For a proper operation based on the system specification, all the internal registers are requested to be programmed by users after power is supplied.

■ OPERATION OF THE CRTC

- **Time Chart of CRT Interface Signals**

The following example shows the display operation in which values of Table 5 are programmed to the CRTC internal registers. Fig. 7 shows the CRT screen format. Fig. 10 shows the time chart of signals output from the CRTC.

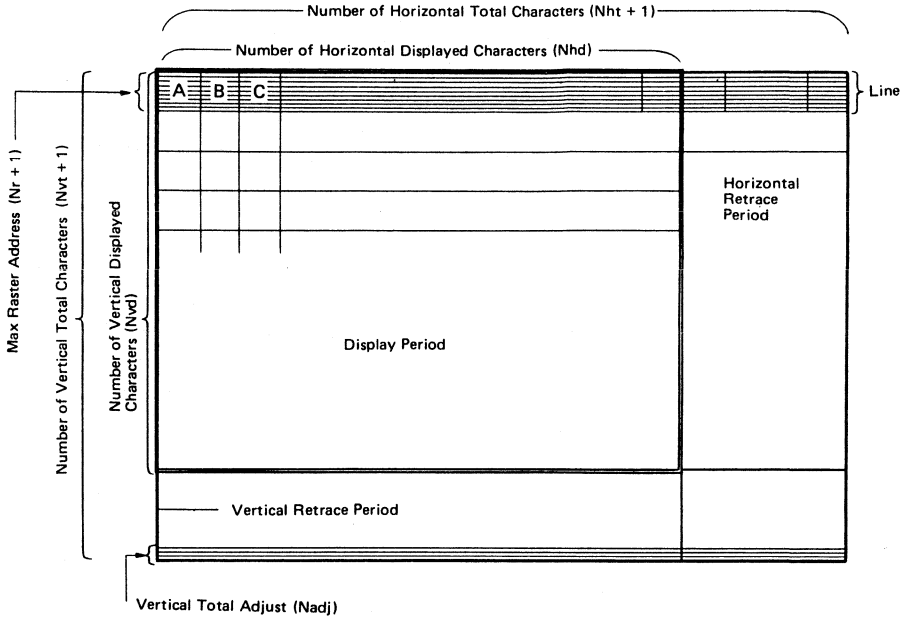


Figure 7 CRT Screen Format

Table 5 Programmed Values into the Registers

Register	Register Name	Value	Register	Register Name	Value
R0	Horizontal Total	Nht	R9	Max Raster Address	Nr
R1	Horizontal Displayed	Nhd	R10	Cursor Start Raster	
R2	Horizontal Sync Position	Nhsp	R11	Cursor End Raster	
R3	Horizontal Sync Width	Nhsw	R12	Start Address (H)	0
R4	Vertical Total	Nvt	R13	Start Address (L)	0
R5	Vertical Total Adjust	Nadj	R14	Cursor (H)	
R6	Vertical Displayed	Nvd	R15	Cursor (L)	
R7	Vertical Sync Position	Nvsp	R16	Light Pen (H)	
R8	Interlace mode		R17	Light Pen (L)	

[NOTE] Nhd<Nht, Nvd<Nvt

The relation between values of Refresh Memory Address ($MA_0 \sim MA_{13}$) and Raster Address ($RA_0 \sim RA_4$) and the display position on the screen is shown in Fig. 16. Fig. 16 shows the case where the value of Start Address is 0.

- **Interlace Control**

Fig. 8 shows an example where the same character is displayed in the non-interlace mode, interlace sync mode, and interlace sync & video mode.

Non-interlace Mode Control

In non-interlace mode, each field is scanned duplicatedly. The values of raster addresses ($RA_0 \sim RA_4$) are counted up one from 0.

Interlace Sync Mode Control

In the interlace sync mode, raster addressed in the even field and the odd field are the same as addressed in the non-interlace mode. One character pattern is displayed mutually and its displayed position in the odd field is set at 1/2 raster space down from that in the even field.

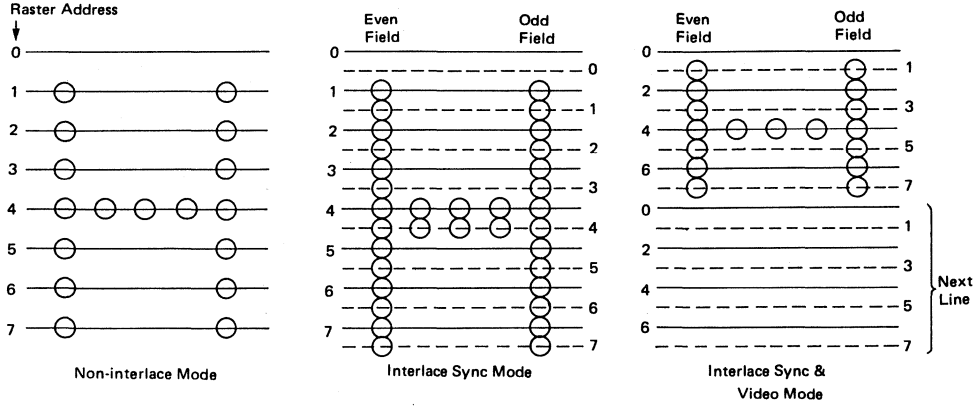


Figure 8 Example of Raster Scan Display

Interlace Sync & Video Mode Control

In the interlace sync & video mode, the character pattern of even number raster address is displayed in even field. Also odd number raster address is displayed in odd field.

- **Cursor Control**

Fig. 9 shows the display patterns where each value is programmed to the cursor start raster register and the cursor end raster register. Programmed values to the cursor start raster register and the cursor end raster register need to be under the following condition.

Cursor Start Raster Register ≦ Cursor End Raster Register ≦ Maximum Raster Address Register.

Time chart of CUDISP is shown in Fig. 14 and Fig. 15.

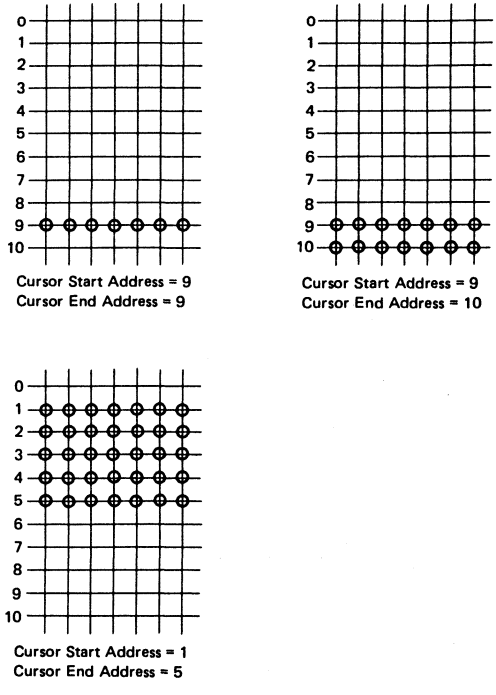


Figure 9 Cursor Control

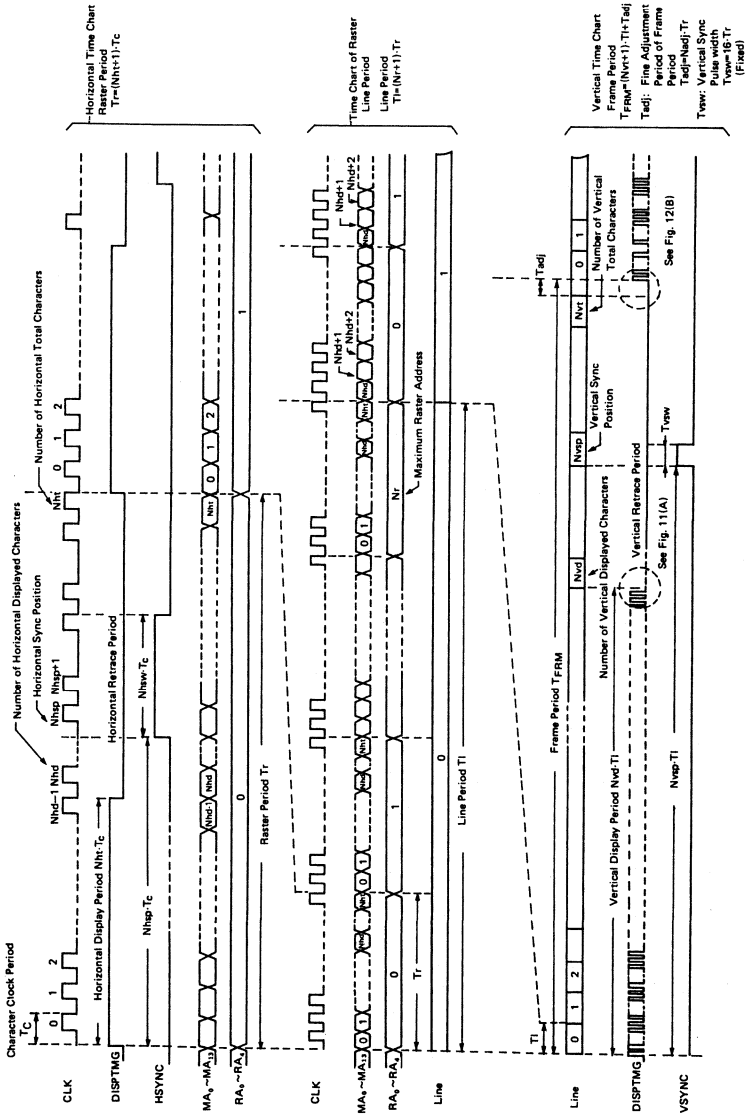


Figure 10 CRTC Time Chart

(Output waveform of horizontal & vertical display in the case where values shown in Table 5 are programmed to each register.)

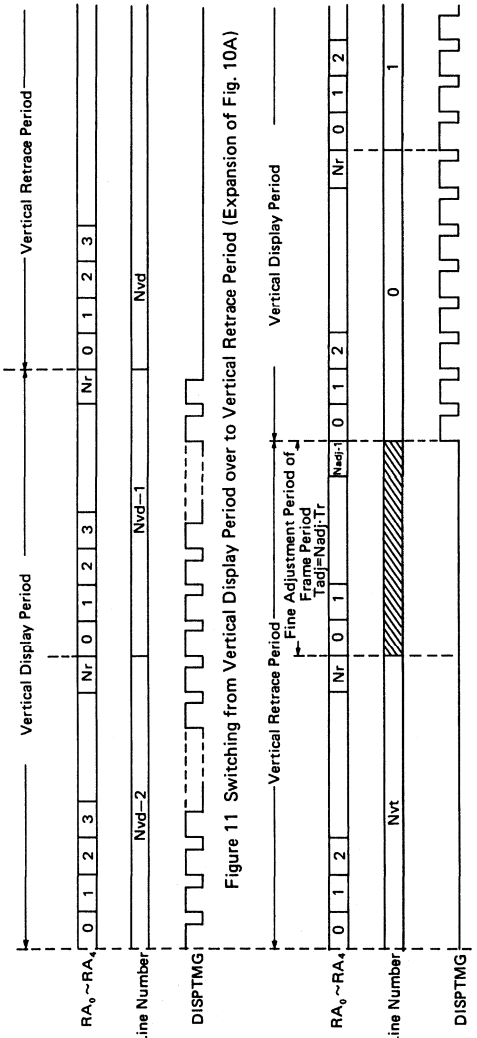
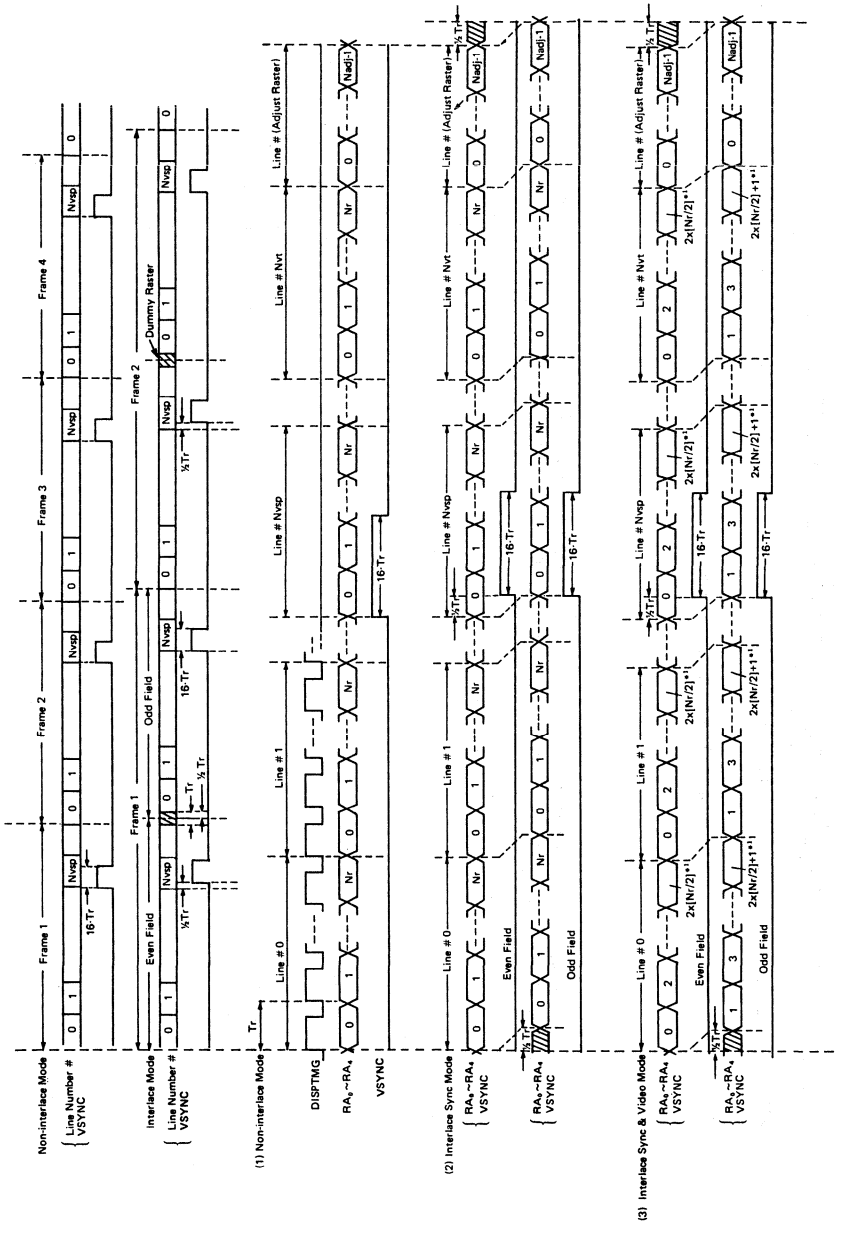


Figure 11 Switching from Vertical Display Period over to Vertical Retrace Period (Expansion of Fig. 10A)

Figure 12 Fine Adjustment Period of Frame in Vertical Display (Expansion of Fig. 10B)



(NOTE) 1. *1: Interlace sync & video mode, maximum raster address (Nr) shall be odd.
2. In interlace mode, Nht shall be odd.

Figure 13 Interlace Control

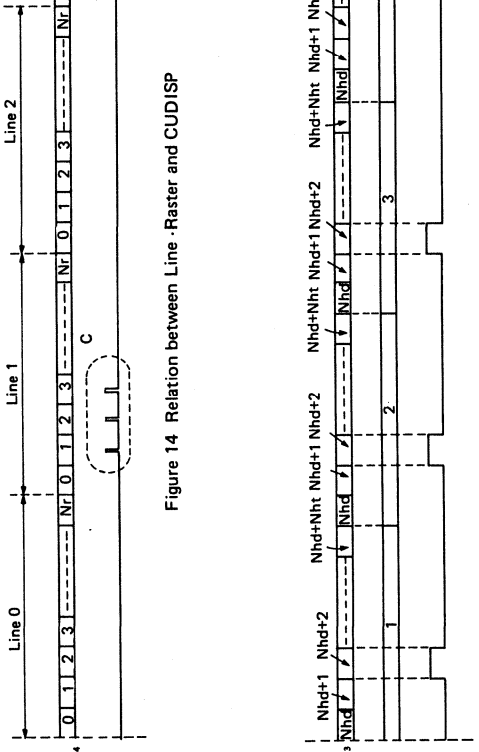


Figure 14 Relation between Line · Raster and CUDISP

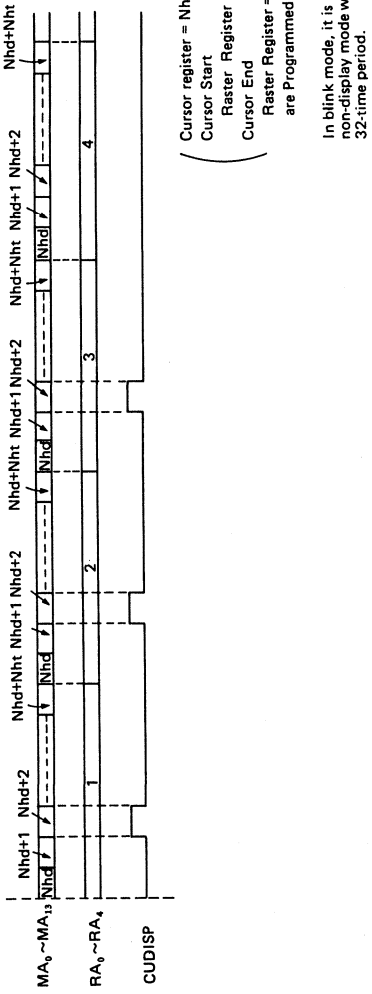


Figure 15 CUDISP Timing (Expansion of Fig. 14C)

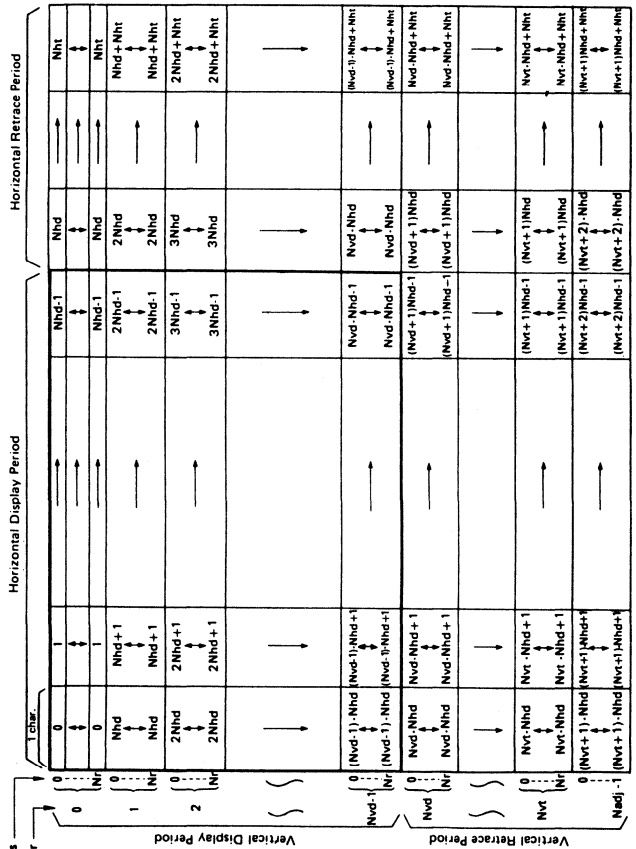


Figure 16 Refresh Memory Address (MA_0~MA_{13})

■ How to Use the CRTC

• Interface to MPU

As shown in Fig. 17, the CRTC is connected with the standard bus of MPU to control the data transfer between them. The CRTC address is determined by $\overline{CS}$ and RS, and the Read/Write operation is controlled by R/$\overline{W}$ and E. When $\overline{CS}$ is "Low" and RS is also "Low", the CRTC address register is selected. When $\overline{CS}$ is "Low" and RS is "High", one of 18 internal registers is selected.

$\overline{RES}$ is the system reset signal. When $\overline{RES}$ becomes "Low", the CRTC internal control logic is reset. But internal registers shown in Table 1 (R0~R17) are not affected by $\overline{RES}$ and remain unchanged.

The CRTC is designed so as to provide an interface to microcomputers, but adding some external circuits enables an interface to other data sources.

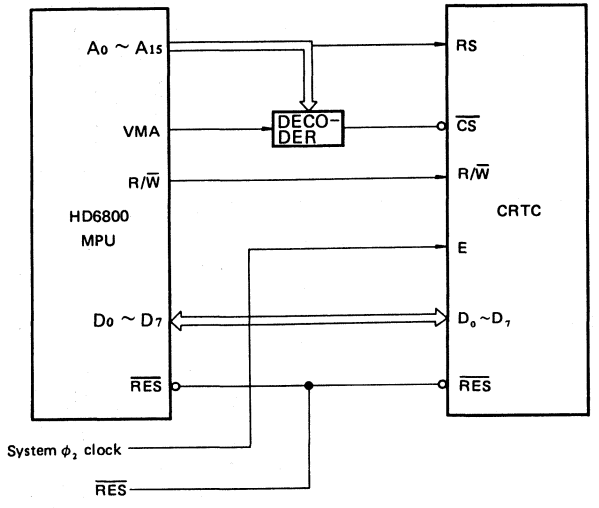


Figure 17 Interface to MPU

• Dot Timing Generating Circuit

CRTC's CLK input (21 pin) is provided with CLK which defines horizontal character time period from the outside. This CLK is generated by dot counter shown in Fig. 18. Fig. 18 shows a example of circuit where horizontal dot number of the character is "9". Fig. 19 shows the operation time chart of dot counter shown in Fig. 18. As this example shows explicitly, CLK is at "Low" level in the former half of horizontal character time and at "High" level in the latter half. It is necessary to be careful so as not to mistake this polarity.

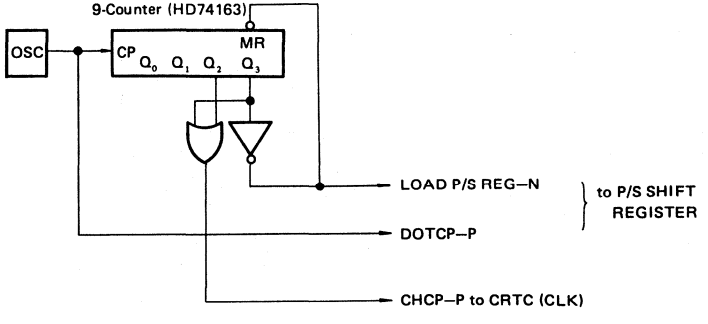


Figure 18 Example of Dot Counter

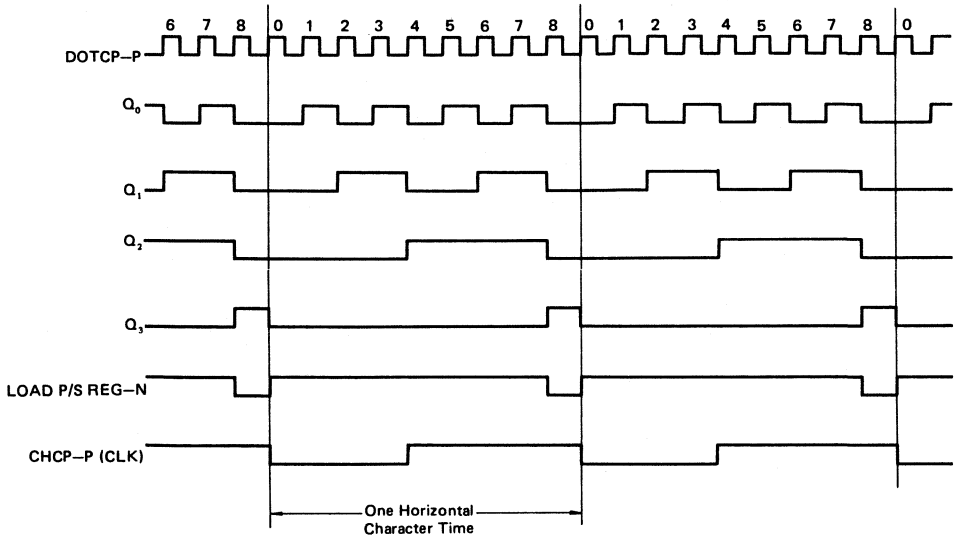


Figure 19 Time Chart of Dot Counter

■ INTERFACE TO DISPLAY CONTROL UNIT

Fig. 20 shows the interface between the CRTC and display control unit. Display control unit is mainly composed of Refresh Memory, Character Generator, and Video Control circuit. For refresh memory, 14 Memory Address line (0~16383) max are provided and for character generator, 5 Raster Address line (0~31) max are provided. For video control circuit, DISPTMG, CUDISP, HSYNC, and VSYNC are sent out. DISPTMG is used to control the blank period of video signal. CUDISP is used as video signal to display the cursor on the CRT screen. Moreover, HSYNC and VSYNC are used as drive signals respectively for CRT horizontal and vertical deflection circuits.

Outputs from video control circuit, (video signals and sync signals) are provided to CRT display unit to control the deflection and brightness of CRT, thus characters are displayed on the screen.

Fig. 21 shows more detailed block diagram of display control unit. This shows how to use DISPTMG and CUDISP signals. By delaying for one or two-character time. DISPTMG signal synchronizes with output timing of parallel-serial converter to control blanking of character video signal. By delaying for one or two-character time. CUDISP signal is mixed with character video signal. Whether delay time of DISPTMG and CUDISP signals should be one or two-character time, moreover, whether LATCH register is necessary for output from refresh memory or not are determined as shown in Fig. 21 by the relations among one horizontal character time, delay time of Memory Address, access time of refresh memory, and access time of character generator.

For reference, time chart from refresh memory address MA to video signal in the case of two-character time delay is shown in Fig. 22.

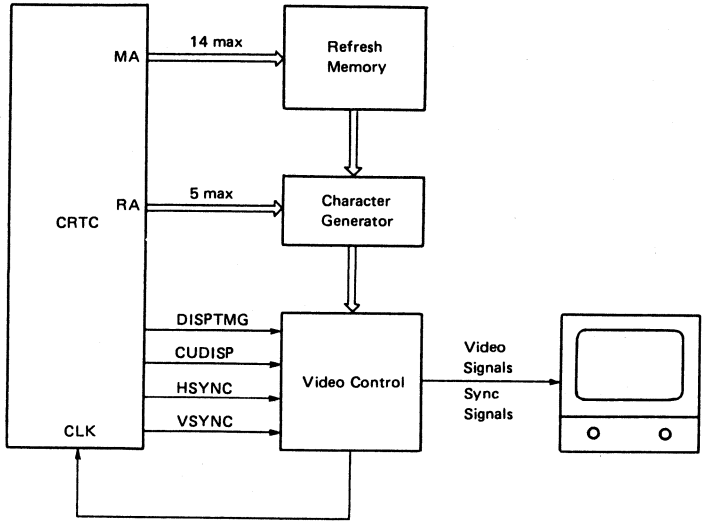
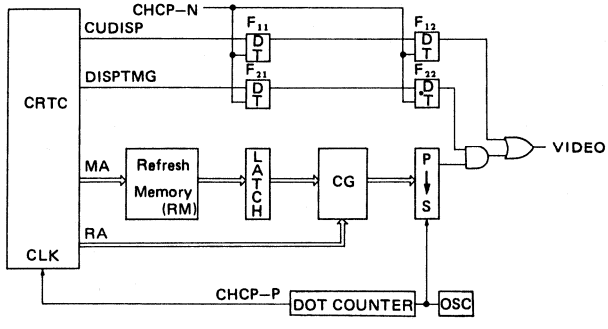


Figure 20 Interface to Display Control Unit



Case	Access Time of RM and CG	LATCH	F_{11}	F_{21}	F_{12}	F_{22}
1	RM Access + CG Access $> t_{CH}-t_{MAD}$	N	N	N	N	N
2	RM Access + CG Access $< t_{CH}-t_{MAD}$	UN	N	N	UN	UN

[NOTE] N : Necessary
UN : Unnecessary
t_{CH} : Cycle time of CHCP
t_{MAD} : Delay time of MA
RM : Refresh Memory

Figure 21 Display Control Unit

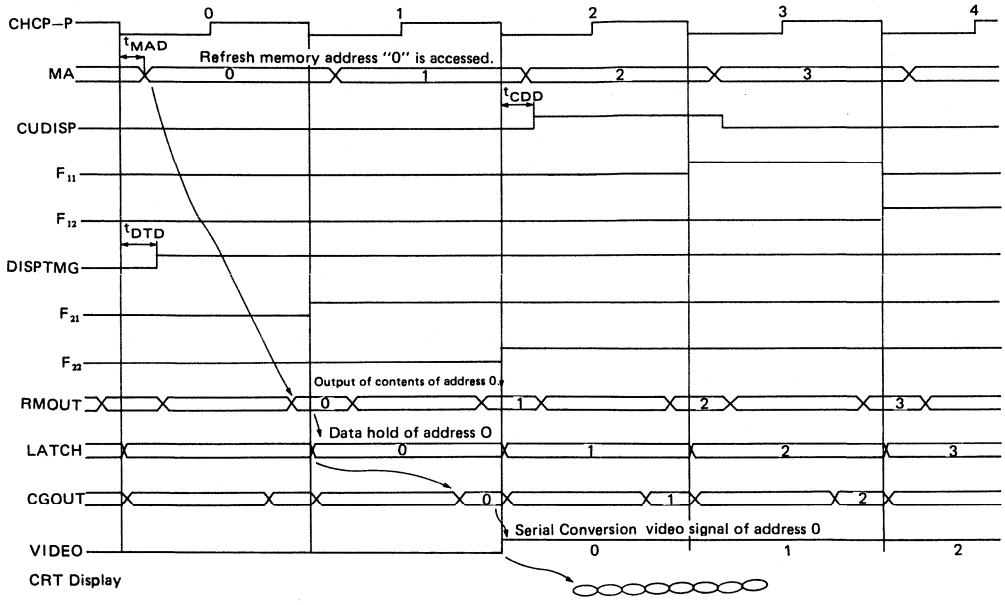


Figure 22 Time Chart of Display Control Unit

■ HOW TO DECIDE PARAMETERS SET ON THE CRTC

• How to Decide Parameters Based on Specification of CRT Display Unit (Monitor)

Number of Horizontal Total Characters

Horizontal deflection frequency f_h is given by specification of CRT display unit. Number of horizontal total characters is determined by the following equation.

$$f_h = \frac{1}{t_C\,(N_{ht} + 1)}$$

where,

t_C : Cycle Time of CLK (Character Clock)
N_{ht} : Programmed Value of Horizontal Total Register (R0)

Number of Vertical Total Characters

Vertical deflection frequency is given by specification of CRT display unit. Number of vertical Total characters is determined by the following equation.

1) Non-interlace Mode
$R_t = (N_{vt} + 1)(N_r + 1) + N_{adj}$

2) Interlace Mode
$R_t = (N_{vt} + 1)(N_r + 1) + N_{adj} + 0.5$

where,

R_t : Number of Total Rasters per frame (Including retrace period)
N_{vt} : Programmed Value of Vertical Total Register (R4)
N_r : Programmed Value of Maximum Raster Address Register (R9)
N_{adj} : Programmed Value of Vertical Total Adjust Register (R5)

Horizontal Sync Pulse Width

Horizontal sync pulse width is programmed to low order 4-bit of horizontal sync width register (R3) in unit of horizontal character time. Programmed value can be selected within from 1 to 15.

Horizontal Sync Position

As shown in Fig. 23, horizontal sync position is normally selected to be in the middle of horizontal retrace period. But there are some cases where is optimum sync position is not located in the middle of horizontal retrace period according to specification of CRT. Therefore, horizontal sync position should be determined by specification of CRT. Horizontal sync pulse position is programmed in unit of horizontal character time.

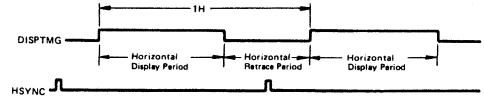


Figure 23 Time Chart of HSYNC

Vertical Sync Position

As shown in Fig. 24, vertical sync position is normally selected to be in the middle of vertical retrace period. But there are some cases where its optimum sync position is not located in the middle of vertical retrace period according to specification of CRT. Therefore, vertical sync position should be determined by specification of CRT. Vertical sync pulse position is programmed to vertical sync position register (R7) in unit of line period.

- **How to Decide Parameters Based on Screen Format**

Dot Number of Characters (Horizontal)

Dot number of characters (horizontal) is determined by character font and character space. An example is shown in Fig. 25. More strictly, dot number of characters (horizontal) N is determined by external N-counter. Character space is set by means shown in Fig. 26.

Dot Number of Characters (Vertical)

Dot number of characters (vertical) is determined by characters font and line space. An example is shown in Fig. 25. Dot number of characters (vertical) is programmed to maximum raster address (R9) of CRTC.

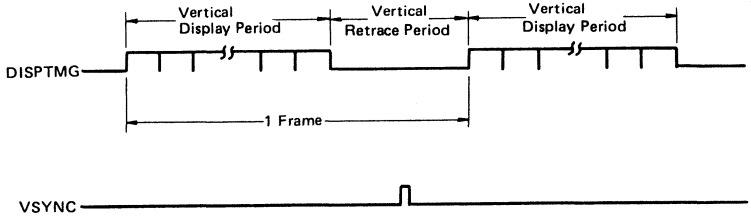


Figure 24 Time Chart of VSYNC

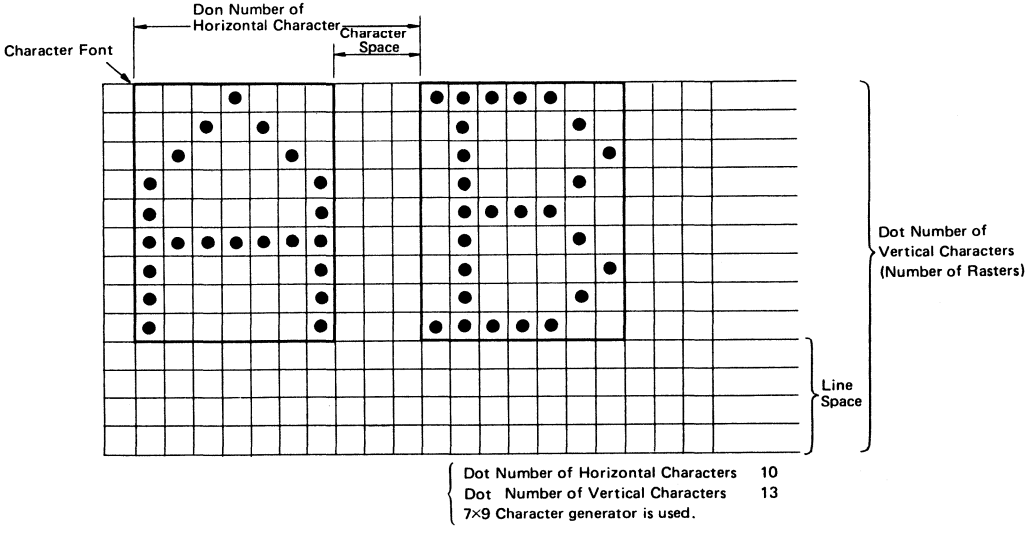


Figure 25 Dot Number of Horizontal and Vertical Characters

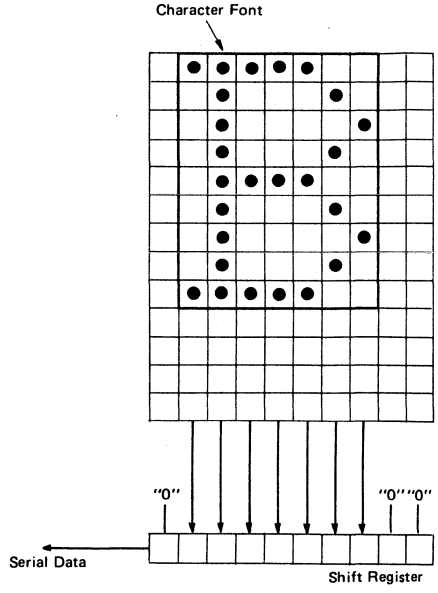


Figure 26 How to Make Character Space

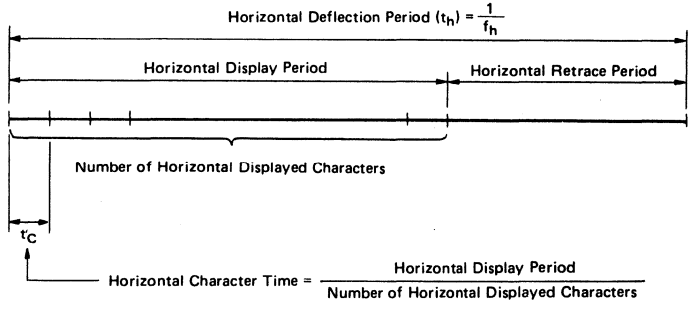


Figure 27 Number of Horizontal Displayed Characters

Number of Horizontal Displayed Characters

Number of horizontal displayed characters is programmed to horizontal displayed register (R1) of the CRTC. Programmed value is based on screen format. Horizontal display period, which is given by specification of horizontal deflection frequency and horizontal retrace period of CRT display unit, determines horizontal character time, being divided by number of horizontal displayed characters. Moreover, its cycle time and access time which are necessary for CRT display system are determined by horizontal character time.

Number of Vertical Displayed Characters

Number of vertical displayed characters is programmed to vertical displayed register (R6). Programmed value is based on screen format. As specification of vertical deflection frequency of CRT determines number of total rasters (Rt) including vertical retrace period and the relation between number of vertical displayed character and total number of rasters on a screen is as mentioned above, CRT which is suitable for desired screen format should be selected.

For optimum screen format, it is necessary to adjust number of rasters per line, number of vertical displayed characters, and total adjust raster (Nadj) within specification of vertical deflection frequency.

Scan Mode

The CRTC can program three-scan modes shown in Table 6 to interlace mode register (R8). An example of character display in each scan mode is shown in Fig. 8.

Table 6 Program of Scan Mode

V	S	Scan Mode	Main Usage
0	0	Non-interlace	Normal Display of Characters & Figures
1	0		
0	1	Interlace Sync	Fine Display of Characters & Figures
1	1	Interlace Sync & Video	Display of Many Characters & Figures Without Using High-resolution CRT

[NOTE] In the interlace mode, the number of times per sec. in raster scanning on one spot on the screen is half as many as that in non-interlace mode. Therefore, when persistence of luminescence is short, flickering may happen. It is necessary to select optimum scan mode for the system, taking characteristics of CRT, raster scan speed, and number of displayed characters and figures into account.

Cursor Display Method

Cursor start raster register and cursor end raster register (R10, R11) enable programming the display modes shown in Table 4 and display patterns shown in Fig. 9. Therefore, it is possible to change the method of cursor display dynamically according to the system conditions as well as to realize the cursor display that meets the system requirements.

Start Address

Start address resisters (R12, R13) give an offset to the address of refresh memory to read out. This enables paging and scrolling easily.

Cursor Register

Cursor registers (R14, R15) enable programming the cursor display position on the screen. As for cursor address, it is not X, Y address but linear address that is programmed.

■ Applications of the CRTC

● Monochrome Character Display

Fig. 28 shows a system of monochrome character display. Character clock signal (CLK) is provided to the CRTC through OSC and dot counter. It is used as basic clock which drives internal control circuits. MPU is connected with the CRTC by standard bus and controls the CRTC initialization and read/write of internal registers.

Refresh memory is composed of RAM which has capacity of one frame at least and the data to be displayed is coded and stored. The data to refresh memory is changed through MPU

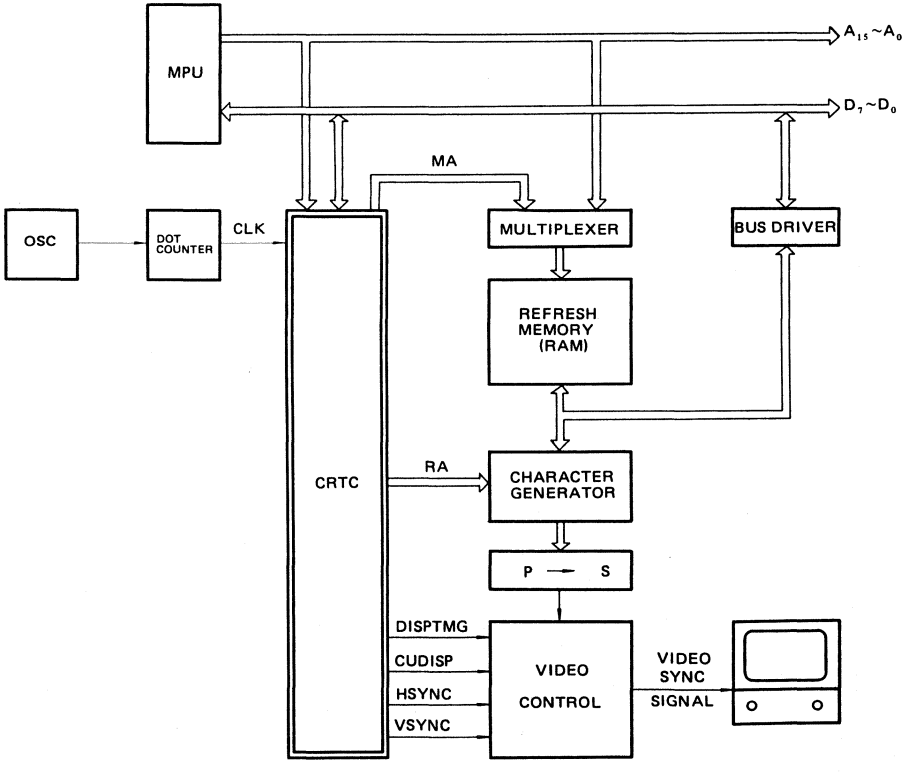


Figure 28 Monochrome Character Display

bus, while refresh memory is read out successively by the CRTC to display a static pattern on the screen. Refresh memory is accessed by both MPU and the CRTC, so it needs to change its address selectively by multiplexer. The CRTC has 14 MA (Memory Address output), but in fact some of them that are needed are used according to capacity of refresh memory.

Code output of refresh memory is provided to character generator. Character generator generates a dot pattern of a specified raster of a specified character in parallel according to code output from refresh memory and RA (Raster Address output) from the CRTC. Parallel-serial converter is normally composed of shift register to convert output of character generator into a serial dot pattern. Moreover, DISPTMG, CUDISP, HSYNC, and VSYNC are provided to video control circuit. It controls blanking for output of parallel-serial converter, mixes these signals with cursor video signal, and generates sync signals for an interface to monitor.

- **Color Character Display**

Fig. 29 shows a system of color character display. In this example, a 3-bit color control bit (R, G, B) is added to refresh memory in parallel with character code and provided to video control circuit. Video control circuit controls coloring as well as blanking and provides three primary color video signals (R, G, B signals) to CRT display device to display characters in seven kinds of color on the screen.

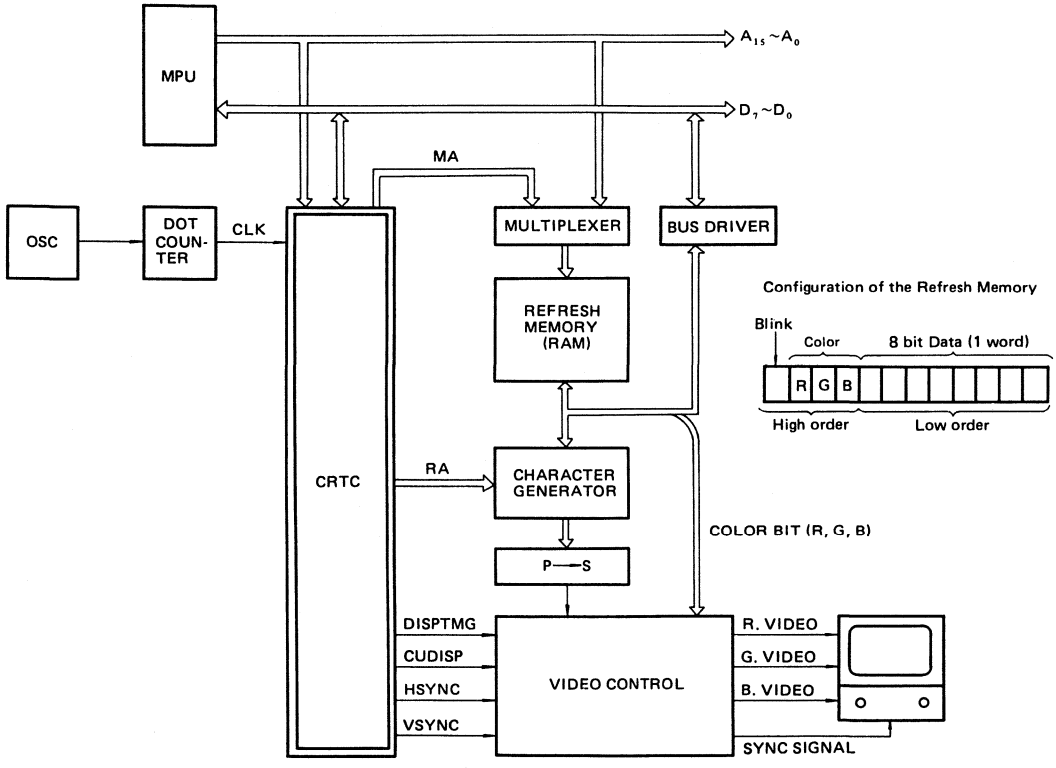


Figure 29 Color Character Display

- **Color limited Graphic Display**

Limited graphic display is to display simple figures as well as character display by combination of picture element which are defined in unit of one character.

As shown in Fig. 30, graphic pattern generator is set up in parallel with character generator and output of these generators are wire-ORed. Which generator is accessed depends on coded output of refresh memory.

In this example, graphic pattern generator adopts ROM, so only the combination of picture elements which are programmed to it is used for this graphic display system. Adopting RAM instead of ROM enables dynamically writable symbols in any combination on one display by changing the contents of them.

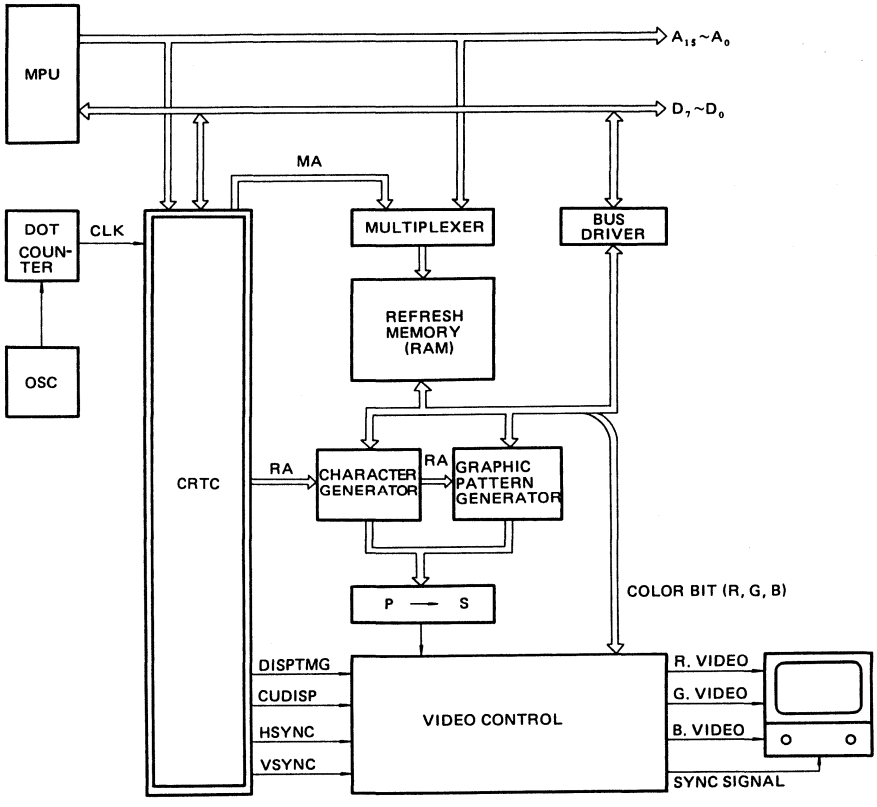


Figure 30 Color Limited Graphic Display

- **Monochrome Full Graphic Display**

Fig. 31 shows a system of monochrome full graphic display. While simple graphic display is figure display by combination of picture elements in unit of 1 picture elements, full graphic display is display of any figures in unit of 1 dot. In this case, refresh memory is dot memory that stores all the dot patterns, so its output is directly provided to parallel-serial converter to be displayed. Dot memory address to refresh the screen is set up by combination of MA and RA of CRTC.

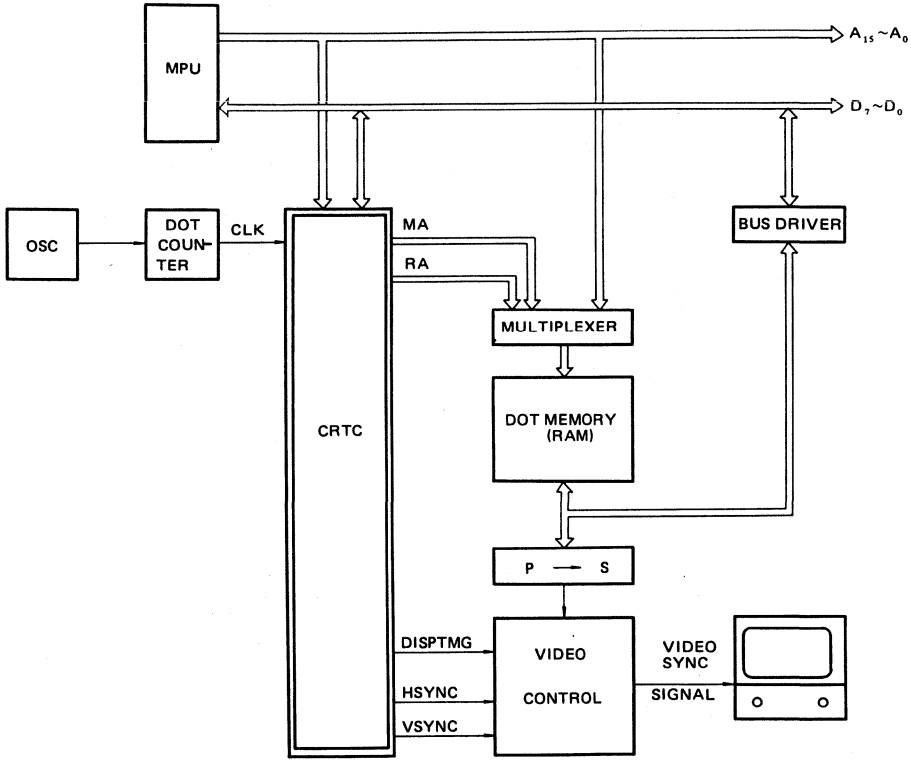


Figure 31 Monochrome Full Graphic Display

Fig. 32 shows an example of access to refresh memory by combination of MA and RA. Fig. 32 shows a refresh memory address method for full graphic display. Correspondence between dot on the CRT screen and refresh memory address is shown in Fig. 33.

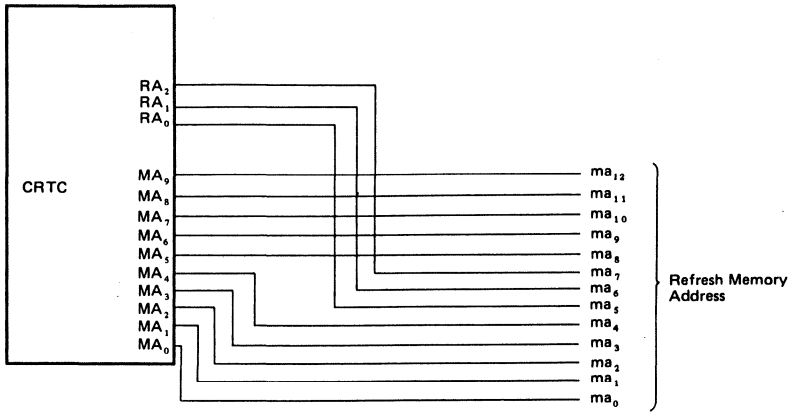


Figure 32 Refresh Memory Address Method for Full Graphic Display

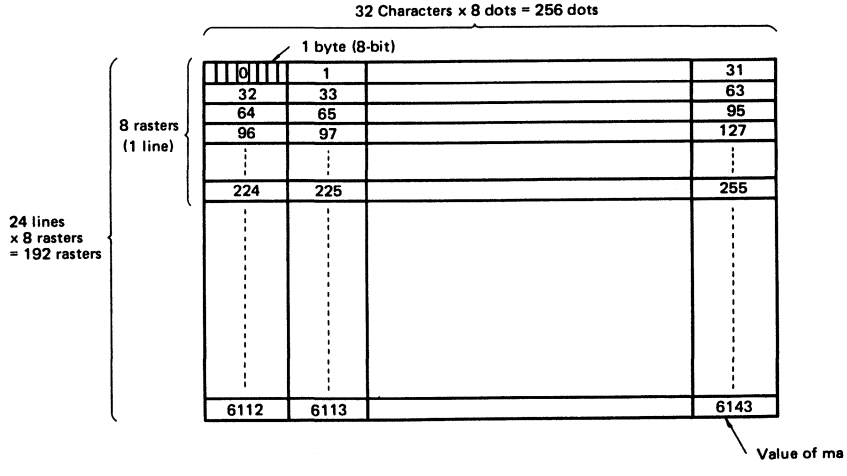


Figure 33 Memory Address and Dot Display Position on the Screen for Full Graphic Display

- **Color Full Graphic Display**

Fig. 34 shows a system of color full graphic display by 7-color display. Refresh memory is composed of three dot memories which are respectively used for red, green, and blue. These dot memories are read out in parallel at one time and their output is provided to three parallel-serial converters. Then video control circuit adds the blanking control to output of these converters and provides it to CRT display device as red, green, and blue video signals with sync signals.

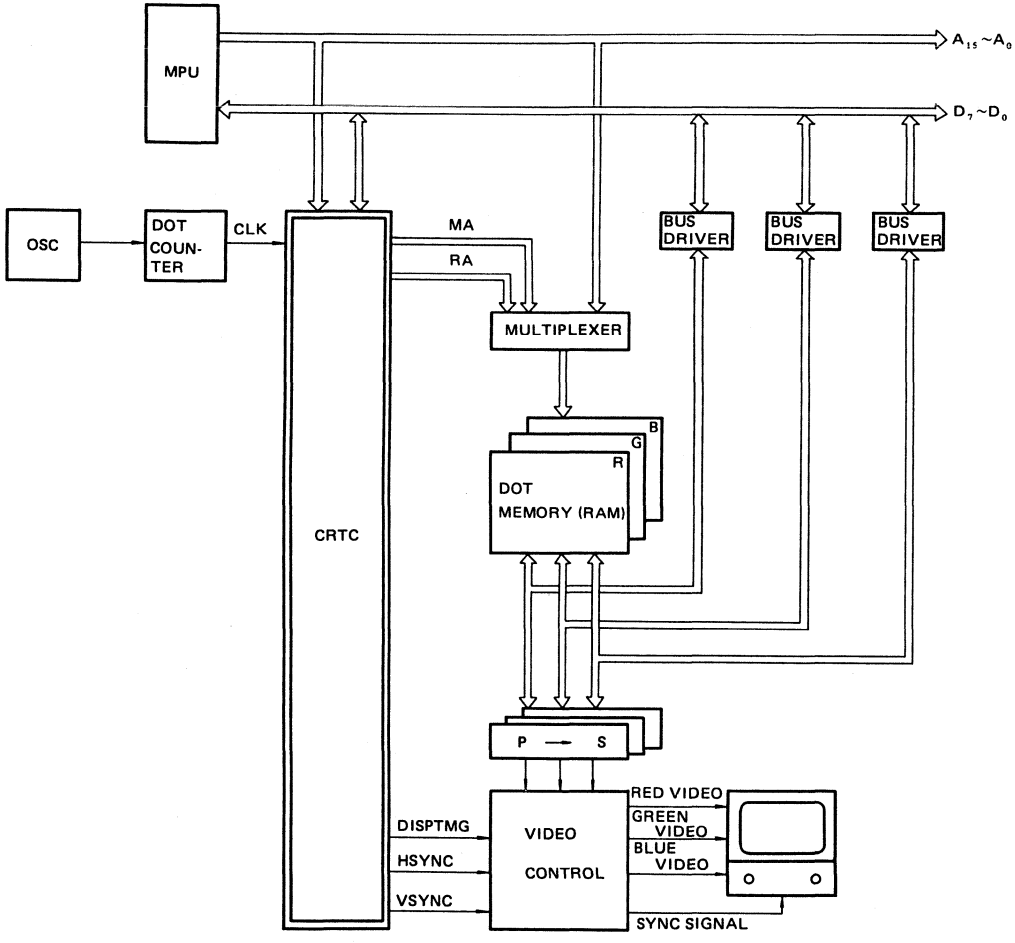


Figure 34 Color Full Graphic Display

- **Cluster Control of CRT Display**

The CRTC enables cluster control that is to control CRT display of plural devices by one CRTC. Fig. 35 shows a system of cluster control. Each display control unit has refresh memory, character generator, parallel-serial converter, and video control circuit separately, but these are controlled together by the CRTC.

In this system, it is possible for plural CRT display devices to have their own display separately.

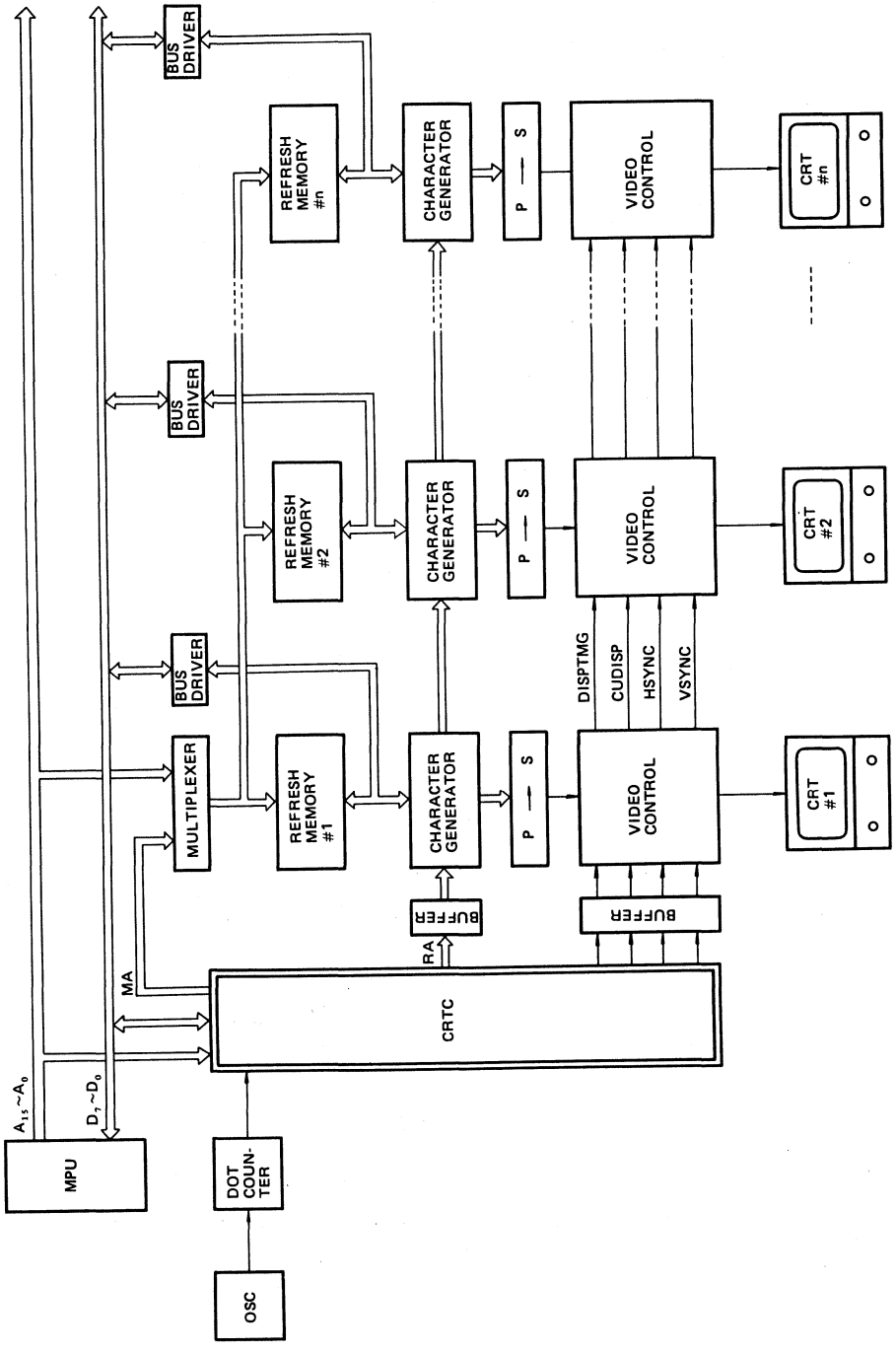


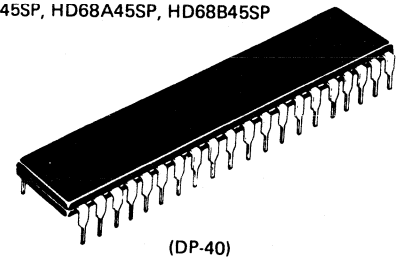
Figure 35 Cluster Control by the CRTC

HD6845S, HD68A45S, HD68B45S

CRTC (CRT Controller)

The CRTC is a LSI controller which is designed to provide an interface for microcomputers to raster scan type CRT displays. The CRTC belongs to the HMCS6800 LSI Family and has full compatibility with MPU in both data lines and control lines. Its primary function is to generate timing signal which is necessary for raster scan type CRT display according to the specification programmed by MPU. The CRTC is also designed as a programmable controller, so applicable to wide-range CRT display from small low-functioning character display up to raster type full graphic display as well as large high-functioning limited graphic display.

HD6845SP, HD68A45SP, HD68B45SP

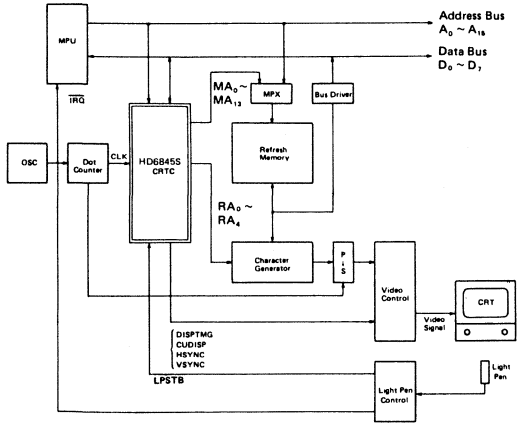


(DP-40)

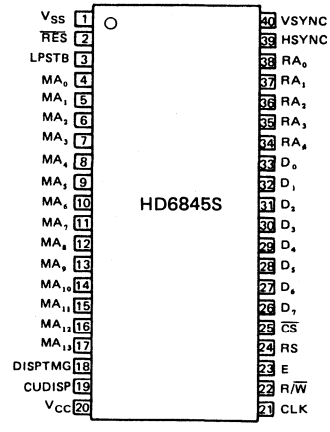
■ FEATURES

- Number of Displayed Characters on the Screen, Vertical Dot Format of One Character, Horizontal and Vertical Sync Signal, Display Timing Signal are Programmable
- 3.7 MHz High Speed Display Operation
- Line Buffer-less Refreshing
- 14-bit Refresh Memory Address Output (16k Words max. Access)
- Programmable Interlace/Non-interlace Scan Mode
- Built-in Cursor Control Function
- Programmable Cursor Height and its Blink
- Built-in Light Pen Detection Function
- Paging and Scrolling Capability
- TTL Compatible
- Single +5V Power Supply

■ SYSTEM BLOCK DIAGRAM



■ PIN ARRANGEMENT



(Top View)

■ ORDERING INFORMATION

CRTC	Bus Timing	CRT Display Timing
HD6845S	1.0 MHz	3.7 MHz max.
HD68A45S	1.5 MHz	
HD68B45S	2.0 MHz	

■ ABSOLUTE MAXIMUM RATINGS

Item	Symbol	Value	Unit
Supply Voltage	V_{CC} *	−0.3 ~ +7.0	V
Input Voltage	V_{in} *	−0.3 ~ +7.0	V
Operating Temperature	T_{opr}	− 20 ~ + 75	°C
Storage Temperature	T_{stg}	− 55 ~ +150	°C

* With respect to V_{SS} (SYSTEM GND)

[NOTE] Permanent LSI damage may occur if maximum ratings are exceeded. Normal operation should be under recommended operating conditions. If these conditions are exceeded, it could affect reliability of LSI.

■ RECOMMENDED OPERATING CONDITIONS

Item	Symbol	min	typ	max	Unit
Supply Voltage	V_{CC} *	4.75	5.0	5.25	V
Input Voltage	V_{IL} *	−0.3	−	0.8	V
	V_{IH} *	2.0	−	V_{CC}	V
Operating Temperature	T_{opr}	− 20	25	75	°C

* With respect to V_{SS} (SYSTEM GND)

■ ELECTRICAL CHARACTERISTICS

• DC CHARACTERISTICS (V_{CC} = 5V ± 5%, V_{SS} = 0V, Ta = −20~+75°C, unless otherwise noted.)

Item	Symbol	Test Condition		min	typ*	max	Unit
Input "High" Voltage	V_{IH}			2.0	−	V_{CC}	V
Input "Low" Voltage	V_{IL}			−0.3	−	0.8	V
Input Leakage Current	I_{in}	V_{in} = 0 ~ 5.25V (Except D_0~D_7)		−2.5	−	2.5	μA
Three-State Input Current (off-state)	I_{TSI}	V_{in} = 0.4 ~ 2.4V V_{CC} = 5.25V (D_0~ D_7)		− 10	−	10	μA
Output "High" Voltage	V_{OH}	I_{LOAD} = −205 μA (D_0~ D_7)		2.4	−	−	V
		I_{LOAD} = −100 μA (Other Outputs)					
Output "Low" Voltage	V_{OL}	I_{LOAD} = 1.6 mA		−	−	0.4	V
Input Capacitance	C_{in}	V_{in} = 0 Ta = 25°C f = 1.0 MHz	D_0 ~ D_7	−	−	12.5	pF
			Other Inputs	−	−	10.0	pF
Output Capacitance	C_{out}	V_{in} = 0V, Ta = 25°C, f = 1.0 MHz		−	−	10.0	pF
Power Dissipation	P_D			−	600	1000	mW

* Ta = 25°C, V_{CC} = 5.0V

● **AC CHARACTERISTICS (V_{CC} = 5V ±5%, V_{SS} = 0V, Ta = -20~+75°C, unless otherwise noted.)**

1. TIMING OF CRTC SIGNAL

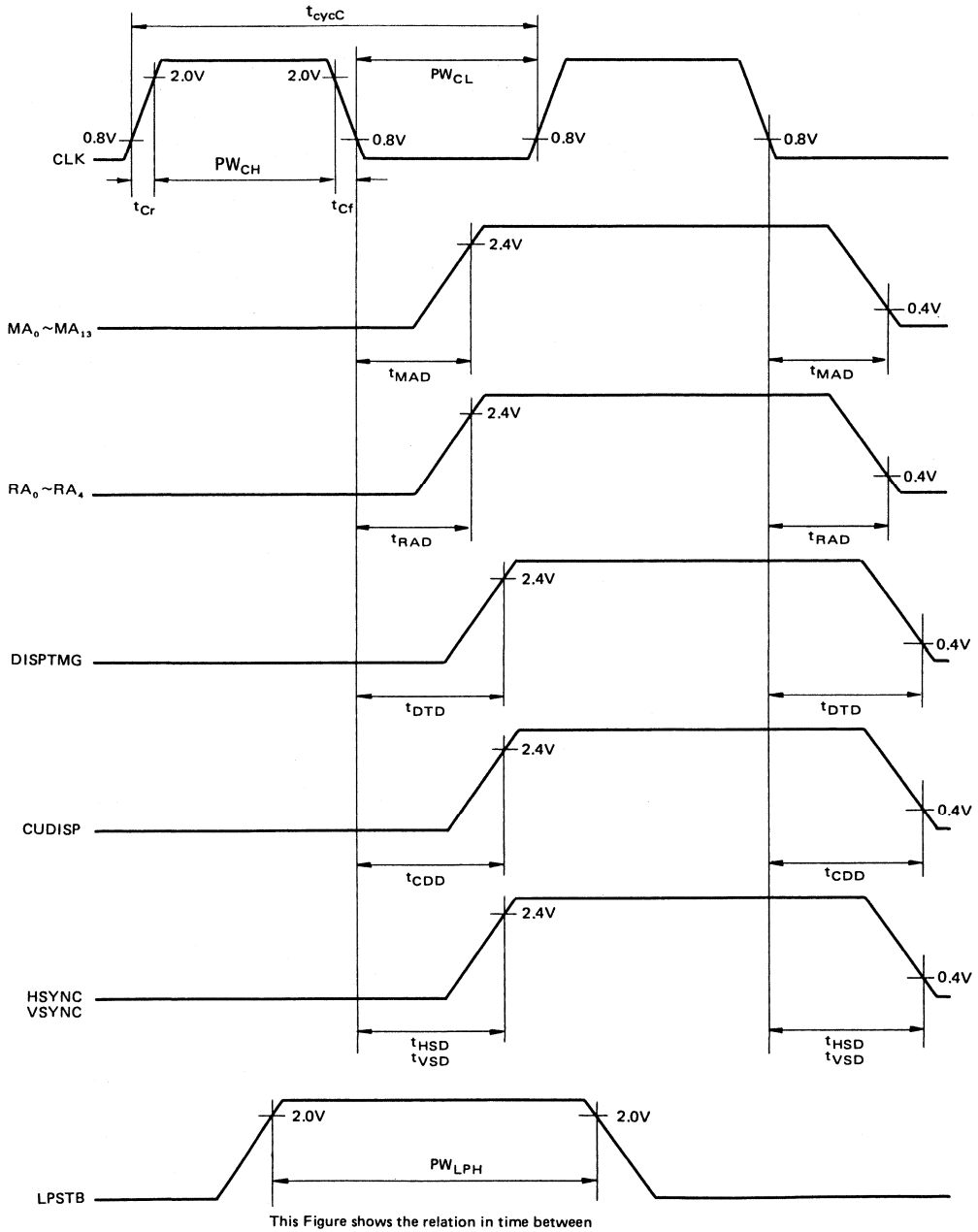
Item	Symbol	Test Condition	min	typ	max	Unit
Clock Cycle Time	t_{cycC}	Fig. 1	270	–	–	ns
Clock "High" Pulse Width	PW_{CH}		130	–	–	ns
Clock "Low" Pulse Width	PW_{CL}		130	–	–	ns
Rise and Fall Time for Clock Input	t_{Cr}, t_{Cf}		–	–	20	ns
Memory Address Delay Time	t_{MAD}		–	–	160	ns
Raster Address Delay Time	t_{RAD}		–	–	160	ns
DISPTMG Delay Time	t_{DTD}		–	–	250	ns
CUDISP Delay Time	t_{CDD}		–	–	250	ns
Horizontal Sync Delay Time	t_{HSD}		–	–	200	ns
Vertical Sync Delay Time	t_{VSD}		–	–	250	ns
Light Pen Strobe Pulse Width	PW_{LPH}		60	–	–	ns
Light Pen Strobe	t_{LPD1}	Fig. 2	–	–	70	ns
Uncertain Time of Acceptance	t_{LPD2}		–	–	0	ns

2. MPU READ TIMING

Item	Symbol	Test Condition	HD6845S			HD68A45S			HD68B45S			Unit
			min	typ	max	min	typ	max	min	typ	max	
Enable Cycle Time	t_{cycE}	Fig. 3	1.0	–	–	0.666	–	–	0.5	–	–	μs
Enable "High" Pulse Width	PW_{EH}		0.45	–	–	0.28	–	–	0.22	–	–	μs
Enable "Low" Pulse Width	PW_{EL}		0.40	–	–	0.28	–	–	0.21	–	–	μs
Enable Rise and Fall Time	t_{Er}, t_{Ef}		–	–	25	–	–	25	–	–	25	ns
Address Set Up Time	t_{AS}		140	–	–	140	–	–	70	–	–	ns
Data Delay Time	t_{DDR}		–	–	320	–	–	220	–	–	180	ns
Data Hold Time	t_{H}		10	–	–	10	–	–	10	–	–	ns
Address Hold Time	t_{AH}		10	–	–	10	–	–	10	–	–	ns
Data Access Time	t_{ACC}		–	–	460	–	–	360	–	–	250	ns

3. MPU WRITE TIMING

Item	Symbol	Test Condition	HD6845S			HD68A45S			HD68B45S			Unit
			min	typ	max	min	typ	max	min	typ	max	
Enable Cycle Time	t_{cycE}	Fig. 4	1.0	–	–	0.666	–	–	0.5	–	–	μs
Enable "High" Pulse Width	PW_{EH}		0.45	–	–	0.28	–	–	0.22	–	–	μs
Enable "Low" Pulse Width	PW_{EL}		0.40	–	–	0.28	–	–	0.21	–	–	μs
Enable Rise and Fall Time	t_{Er}, t_{Ef}		–	–	25	–	–	25	–	–	25	ns
Address Set Up Time	t_{AS}		140	–	–	140	–	–	70	–	–	ns
Data Set Up Time	t_{DSW}		195	–	–	80	–	–	60	–	–	ns
Data Hold Time	t_{H}		10	–	–	10	–	–	10	–	–	ns
Address Hold Time	t_{AH}		10	–	–	10	–	–	10	–	–	ns



This Figure shows the relation in time between CLK signal and each output signals. Output sequence is shown in Figs. 10~15.

Figure 1 Time Chart of the CRTC

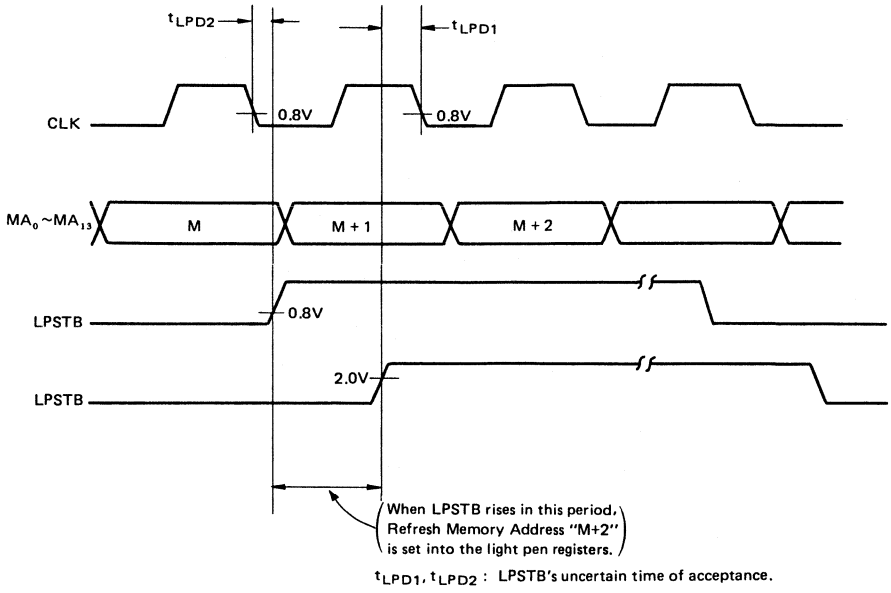


Figure 2 LPSTB Input Timing & Refresh Memory Address that is set into the light pen registers.

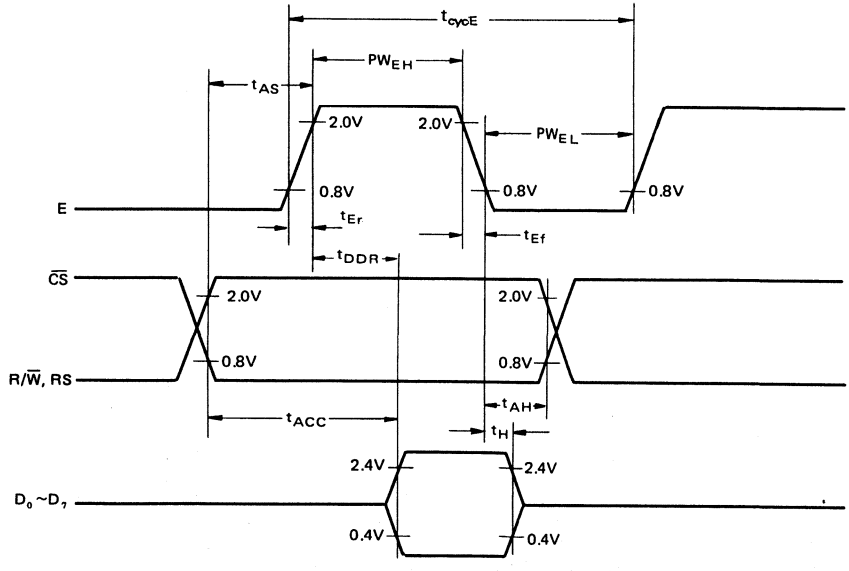


Figure 3 Read Sequence

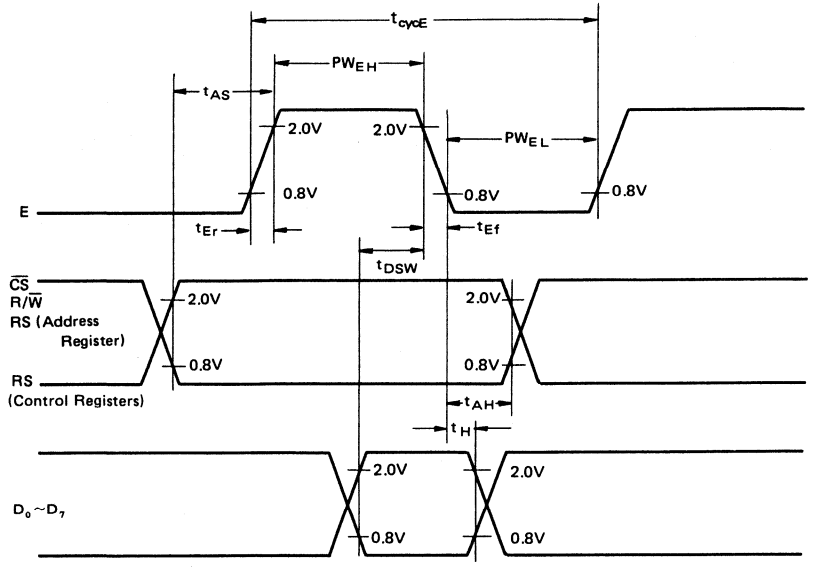
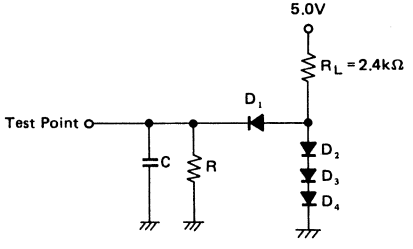


Figure 4 Write Sequence



C = 130pF (D_0~D_7)
= 30pF (Output signals except D_0~D_7)
R = 11kΩ (D_0~D_7)
= 24kΩ (Output signals except D_0~D_7)
D_1~D_4 are 1S2074Ⓗ or equivalent.

Figure 5 Test Loads

■ SYSTEM DESCRIPTION

The CRTC is a LSI which is connected with MPU and CRT display device to control CRT display. The CRTC consists of internal register group, horizontal and vertical timing circuits, linear address generator, cursor control circuit, and light pen detection circuit. Horizontal and vertical timing circuit generate RA_0~RA_4, DISPTMG, HSYNC, and VSYNC. RA_0~RA_4 are raster address signals and used as input signals for Character Generator. DISPTMG, HSYNC, and VSYNC signals are received by video control circuit. This horizontal and vertical timing circuit consists of internal counter and comparator circuit. Linear address generator generates refresh memory address MA_0 ~MA_{13} to be used for refreshing the screen. By these address signals, refresh memory is accessed periodically. As 14 refresh memory address signals are prepared, 16k words max are accessible. Moreover, the use of start address register enables paging and scrolling. Light pen detection circuit detects light pen position on the screen. When light pen strobe signal is received, light pen register memorizes linear address generated by linear address generator in order to memorize where light pen is on the screen. Cursor control circuit controls the position of cursor, its height, and its blink.

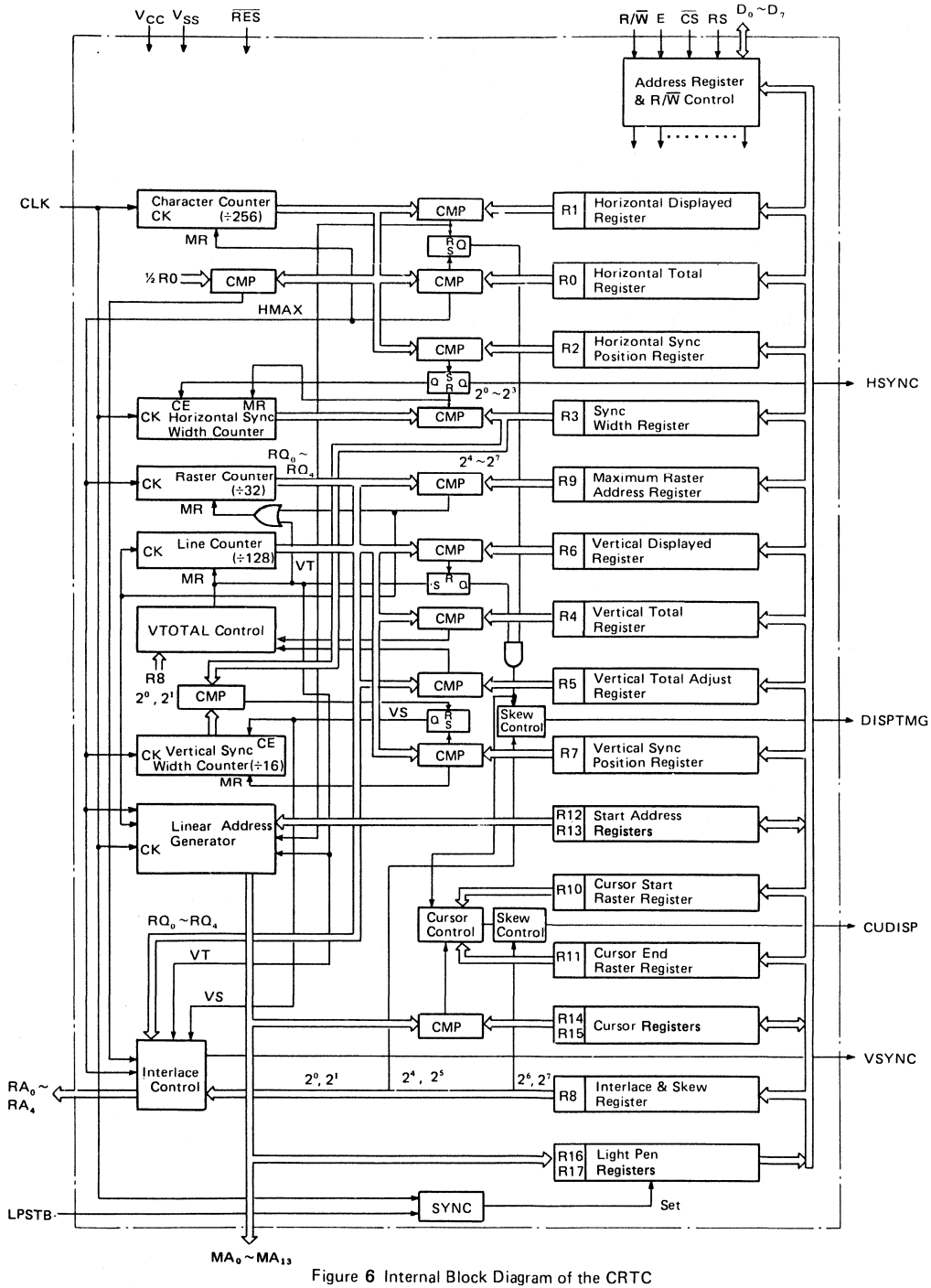


Figure 6 Internal Block Diagram of the CRTC

■ FUNCTION OF SIGNAL LINE

The CRTC provides 13 interface signals to MPU and 25 interface signals to CRT display.

• Interface Signals to MPU

Bi-directional Data Bus (D_0~D_7)

Bi-directional data bus(D_0~D_7) are used for data transfer between the CRTC and MPU. The data bus outputs are 3-state buffers and remain in the high-impedance state except when MPU performs a CRTC read operation.

Read/Write (R/$\overline{W}$)

Read/Write signal (R/$\overline{W}$) controls the direction of data transfer between the CRTC and MPU. When R/$\overline{W}$ is at "High" level, data of CRTC is transfered to MPU. When R/$\overline{W}$ is at "Low" level, data of MPU is transfered to CRTC.

Chip Select ($\overline{CS}$)

Chip Select signal ($\overline{CS}$) is used to address the CRTC. When $\overline{CS}$ is at "Low" level, it enables Read/Write operation to CRTC internal registers. Normally this signal is derived from decoded address signal of MPU under the condition that VMA of MPU is at "High" level.

Register Select (RS)

Register Select signal (RS) is used to select the address register and 18 control registers of the CRTC. When RS is at "Low" level, the address register is selected and when RS is at "High" level, control registers are selected. This signal is normally a derivative of the lowest bit (A0) of MPU address bus.

Enable(E)

Enable signal (E) is used as strobe signal in MPU Read/Write operation with the CRTC internal registers. This signal is normally a derivative of the HMCS6800 System ϕ_2 clock.

Reset ($\overline{RES}$)

Reset signal ($\overline{RES}$) is an input signal used to reset the CRTC. When $\overline{RES}$ is at "Low" level, it forces the CRTC into the following status.

1) All the counters in the CRTC are cleared and the device stops the display operation.
2) All the outputs go down to "Low" level.
3) Control registers in the CRTC are not affected and remain unchanged.

This signal is different from other HMCS6800 family LSIs in the following functions and has restrictions for usage.

1) $\overline{RES}$ has capability of reset function only when LPSTB is at "Low" level.
2) The CRTC starts the display operation immediately after $\overline{RES}$ goes "High" level.

• Interface Signals to CRT Display Device

Character Clock (CLK)

CLK is a standard clock input signal which defines character timing for the CRTC display operation. CLK is normally derived from the external high-speed dot timing logic.

Horizontal Sync (HSYNC)

HSYNC is an active "High" level signal which provides horizontal synchronization for display device.

Vertical Sync (VSYNC)

VSYNC is an active "High" level signal which provides vertical synchronization for display device.

Display Timing (DISPTMG)

DISPTMG is an active "High" level signal which defines the display period in horizontal and vertical raster scanning. It is necessary to enable video signal only when DISPTMG is at "High" level.

Refresh Memory Address (MA_0~MA_{13})

MA_0~MA_{13} are refresh memory address signals which are used to access to refresh memory in order to refresh the CRT screen periodically. These outputs enables 16k words max. refresh memory access. So, for instance, these are applicable up to 2000 characters/screen and 8-page system.

Raster Address (RA_0~RA_4)

RA_0~RA_4 are raster address signals which are used to select the raster of the character generator or graphic pattern generator etc.

Cursor Display (CUDISP)

CUDISP is an active "High" level video signal which is used to display the cursor on the CRT screen. This output is inhibited while DISPTMG is at "Low" level. Normally this output is mixed with video signal and provided to the CRT display device.

Light Pen Strobe (LPSTB)

LPSTB is an active "High" level input signal which accepts strobe pulse detected by the light pen and control circuit. When this signal is activated, the refresh memory address (MA_0~MA_{13}) which are shown in Fig. 2 are stored in the 14-bit light pen register. The stored refresh memory address need to be corrected in software, taking the delay time of the display device, light pen, and light pen control circuits into account.

■ REGISTER DESCRIPTION

Table 1 Internal Registers Assignment

$\overline{CS}$	RS	Address Register 4	3	2	1	0	Register #	Register Name	Program Unit	READ	WRITE	Data Bit 7	6	5	4	3	2	1	0
1	×	×	×	×	×	×			–	–	–								
0	0	×	×	×	×	×	AR	Address Register	–	×	O								
0	1	0	0	0	0	0	R0	Horizontal Total *	Character	×	O								
0	1	0	0	0	0	1	R1	Horizontal Displayed	Character	×	O								
0	1	0	0	0	1	0	R2	Horizontal Sync* Position	Character	×	O								
0	1	0	0	0	1	1	R3	Sync Width	Vertical-Raster, Horizontal-Character	×	O	wv3	wv2	wv1	wv0	wh3	wh2	wh1	wh0
0	1	0	0	1	0	0	R4	Vertical Total *	Line	×	O								
0	1	0	0	1	0	1	R5	Vertical Total Adjust	Raster	×	O								
0	1	0	0	1	1	0	R6	Vertical Displayed	Line	×	O								
0	1	0	0	1	1	1	R7	Vertical Sync * Position	Line	×	O								
0	1	0	1	0	0	0	R8	Interlace & Skew	–	×	O	C1	C0	D1	D0			V	S
0	1	0	1	0	0	1	R9	Maximum Raster Address	Raster	×	O								
0	1	0	1	0	1	0	R10	Cursor Start Raster	Raster	×	O		B	P					
0	1	0	1	0	1	1	R11	Cursor End Raster	Raster	×	O								
0	1	0	1	1	0	0	R12	Start Address(H)	–	O	O								
0	1	0	1	1	0	1	R13	Start Address(L)	–	O	O								
0	1	0	1	1	1	0	R14	Cursor(H)	–	O	O								
0	1	0	1	1	1	1	R15	Cursor (L)	–	O	O								
0	1	1	0	0	0	0	R16	Light Pen(H)	–	O	×								
0	1	1	0	0	0	1	R17	Light Pen(L)	–	O	×								

[NOTE] 1. The Registers marked *: (Written Value) = (Specified Value) – 1
2. Written Value of R9 is mentioned below.
   1) Non-interlace Mode, Interlace Sync Mode } (Written Value) = (Specified Value) - 1
   2) Interlace Sync & Video Mode
      (Written Value) = (Specified Value) -2
3. C0 and C1 specify skew of CUDISP.
   D0 and D1 specify skew of DISPTMG.
   When S is "1", V specifies video mode. S specifies the Interlace Sync Mode.
4. B specifies the cursor blink. P specifies the cursor blink period.
5. wv0~wv3 specify the pulse width of Vertical Sync Signal.
   wh0~wh3 specify the pulse width of Horizontal Sync Signal.
6. R0 is ordinally programmed to be odd number in interlace mode.
7. O; Yes, ×; No

■ FUNCTION OF INTERNAL REGISTERS

• Address Register (AR)

This is a 5-bit register used to select 18 internal control registers (R0~R17). Its contents are the address of one of 18 internal control registers. Programming the data from 18 to 31 produces no results. Access to R0~R17 requires, first of all, to write the address of corresponding control register into this register. When RS and $\overline{CS}$ are at "Low" level, this register is selected.

• Horizontal Total Register (R0)

This is a register used to program total number of horizontal characters per line including the retrace period. The data is 8-bit and its value should be programmed according to the specification of the CRT. When M is total number of characters, M-1 shall be programmed to this register. When programming for interlace mode, M must be even.

• Horizontal Displayed Register (R1)

This is a register used to program the number of horizontal displayed characters per line. Data is 8-bit and any number that is smaller than that of horizontal total characters can be programmed.

• Horizontal Sync Position Register (R2)

This is a register used to program horizontal sync position as multiples of the character clock period. Data is 8-bit and any number that is lower than the horizontal total number can be programmed. When H is character number of horizontal Sync Position, H-1 shall be programmed to this register. When programmed value of this register is increased, the display position on the CRT screen is shifted to the left. When programmed value is decreased, the position is shifted to the right. Therefore, the optimum horizontal position can be determined by this value.

• Sync Width Register (R3)

This is a register used to program the horizontal sync pulse width and the vertical sync pulse width. The horizontal sync pulse width is programmed in the lower 4-bit as multiples of the character clock period. "0" can't be programmed. The vertical sync pulse width is programmed in higher 4-bit as multiples of the raster period. When "0" is programmed in higher 4-bit, 16 raster period (16H) is specified.

• Vertical Total Register (R4)

This is a register used to program total number of lines per frame including vertical retrace period. The data is within 7-bit and its value should be programmed according to the specification of the CRTC. When N is total number of lines, N-1 shall be programmed to this register.

• Vertical Total Adjust Register (R5)

This is a register used to program the optimum number to adjust total number of rasters per field. This register enables to decide the number of vertical deflection frequency more strictly.

• Vertical Displayed Register (R6)

This is a register used to program the number of displayed character rows on the CRT screen. Data is 7-bit and any number that is smaller than that of vertical total characters can be programmed.

Table 2 Pulse Width of Vertical Sync Signal

VSW				Pulse Width
2^7	2^6	2^5	2^4	
0	0	0	0	16H
0	0	0	1	1
0	0	1	0	2
0	0	1	1	3
0	1	0	0	4
0	1	0	1	5
0	1	1	0	6
0	1	1	1	7
1	0	0	0	8
1	0	0	1	9
1	0	1	0	10
1	0	1	1	11
1	1	0	0	12
1	1	0	1	13
1	1	1	0	14
1	1	1	1	15

H; Raster period

Table 3 Pulse Width of Horizontal Sync Signal

HSW				Pulse Width
2^3	2^2	2^1	2^0	
0	0	0	0	– (Note)
0	0	0	1	1 CH
0	0	1	0	2
0	0	1	1	3
0	1	0	0	4
0	1	0	1	5
0	1	1	0	6
0	1	1	1	7
1	0	0	0	8
1	0	0	1	9
1	0	1	0	10
1	0	1	1	11
1	1	0	0	12
1	1	0	1	13
1	1	1	0	14
1	1	1	1	15

CH; Character clock period
(Note) HSW = "0" can't be used.

- **Vertical Sync Position Register (R7)**

This is a register used to program the vertical sync position on the screen as multiples of the horizontal character line period. Data is 7-bit and any number that is equal to or less than vertical total characters can be programmed. When V is character number of vertical sync position, V-1 shall be programmed to this register. When programmed value of this register is increased, the display position is shifted up. When programmed value is decreased, the position is shifted down. Therefore, the optimum vertical position may be determined by this value.

- **Interlace and Skew Register (R8)**

This is a register used to program raster scan mode and skew (delay) of CUDISP and DISPTMG.

Raster Scan Mode Program Bit (V, S)

Raster scan mode is programmed in the V, S bit.

Table 4 Raster Scan Mode (2^1, 2^0)

V	S	Raster Scan Mode
0	0	Non-interlace Mode
1	0	
0	1	Interlace Sync Mode
1	1	Interlace Sync & Video Mode

In the non-interlace mode, the rasters of even number field and odd number field are scanned duplicatedly. In the interlace sync mode, the rasters of odd number field are scanned in the middle of even number field. Then it is controlled to display the same character pattern in two fields. In the interlace sync & video mode, the raster scan method is the same as the interlace sync mode, but it is controlled to display different character pattern in two field.

Skew Program Bit (C1, C0, D1, D0)

These are used to program the skew (delay) of CUDISP and DISPTMG.

Skew of these two kinds of signals are programmed separately.

Table 5 DISPTMG Skew Bit (2^5, 2^4)

D1	D0	DISPTMG
0	0	Non-skew
0	1	One-character skew
1	0	Two-character skew
1	1	Non-output

Table 6 CUDISP Skew Bit (2^7, 2^6)

C1	C0	CUDISP
0	0	Non-skew
0	1	One-character skew
1	0	Two-character skew
1	1	Non-output

Skew function is used to delay the output timing of CUDISP and DISPTMG in LSI for the time to access refresh memory, character generator or pattern generator, and to make the same phase with serial video signal.

- **Maximum Raster Address Register (R9)**

This is a register used to program maximum raster address within 5-bit. This register defines total number of rasters per character including line space. This register is programmed as follows.

Non-interlace Mode, Interlace Sync Mode

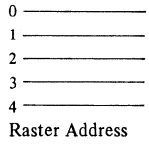
When total number of rasters is RN, RN-1 shall be programmed.

Interlace Sync & Video Mode

When total number of rasters is RN, RN-2 shall be programmed.

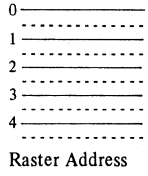
This manual defines total number of rasters in non-interlace mode, interlace sync mode and interlace sync & video mode as follows:

Non-interlace Mode



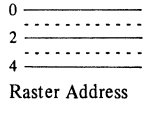
Total Number of Rasters:5
Programmed Value: Nr = 4
(The same as displayed total number of rasters)

Interlace Sync Mode



Total Number of Rasters:5
Programmed Value: Nr = 4
(In the interlace sync mode, total number of rasters in both the even and odd fields is ten. On programming, the half of it is defined as total number of rasters.)

Interlace Sync & Video Mode



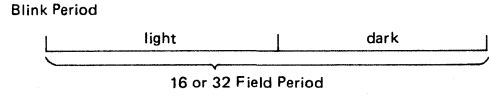
Total Number of Rasters:5
Programmed Value: Nr = 3
(Total number of rasters displayed in the even field and the odd field.)

- **Cursor Start Raster Register (R10)**

This is a register used to program the cursor start raster address by lower 5-bit (2^0~2^4) and the cursor display mode by higher 2-bit (2^5, 2^6).

Table 7 Cursor Display Mode (2^6, 2^5)

B	P	Cursor Display Mode
0	0	Non-blink
0	1	Cursor Non-display
1	0	Blink 16 Field Period
1	1	Blink 32 Field Period



- **Cursor End Raster Register (R11)**
  This is register used to program the cursor end raster address.

- **Start Address Register (R12, R13)**
  These are used to program the first address of refresh memory to read out.
  Paging and scrolling is easily performed using this register. This register can be read but the higher 2-bit (2^6, 2^7) of R12 are always "0".

- **Cursor Register (R14, R15)**
  These two read/write registers stores the cursor location. The higher 2-bit (2^6, 2^7) of R14 are always "0".

- **Light Pen Register (R16, R17)**
  These read only registers are used to catch the detection address of the light pen. The higher 2-bit (2^6, 2^7) of R16 are always "0". Its value needs to be corrected by software because there is time delay from address output of the CRTC to signal input LPSTB pin of the CRTC in the process that raster is lit after address output and light pen detects it. Moreover, delay time shown in Fig. 2 needs to be taken into account.

Restriction on Programming Internal Register

1) $0 < Nhd < Nht + 1 \leqq 256$
2) $0 < Nvd < Nvt + 1 \leqq 128$
3) $0 \leqq Nhsp \leqq Nht$
4) $0 \leqq Nvsp \leqq Nvt$*
5) $0 \leqq N_{CSTART} \leqq N_{CEND} \leqq Nr$ (Non-interlace, Interlace sync mode)
   $0 \leqq N_{CSTART} \leqq N_{CEND} \leqq Nr + 1$ (Interlace sync & video mode)
6) $2 \leqq Nr \leqq 30$ (Interlace Sync & Video mode)
7) $3 \leqq Nht$ (Except non-interlace mode)
   $5 \leqq Nht$ (Non-interlace mode only)

* In the interlace mode, pulse width is changed ±½ raster time when vertical sync signal extends over two fields.

Notes for Use

(1) The method of directly using the value programmed in the internal registers of LSI for controlling the CRT is adopted. Consequently, the display may flicker on the screen when the contents of the registers are changed from bus side asynchronously with the display operation.

Cursor Register

Writing into this register at frequent intervals for moving the cursor should be performed during horizontal and vertical retrace period.

Start Address Register

Writing into the start address register at frequent intervals for scrolling and paging should be performed during horizontal and vertical display period.

It is desirable to avoid programming other registers during display operation.

(2) The $\overline{RES}$ assertion at power-on does not define the internal registers of the HD6845S. For a proper operation based on the system specification, all the internal registers are requested to be programmed by users after power is supplied.

■ OPERATION OF THE CRTC

- **Time Chart of CRT Interface Signals**
  The following example shows the display operation in which values of Table 8 are programmed to the CRTC internal registers. Fig. 7 shows the CRT screen format. Fig. 10 shows the time chart of signals output from the CRTC.

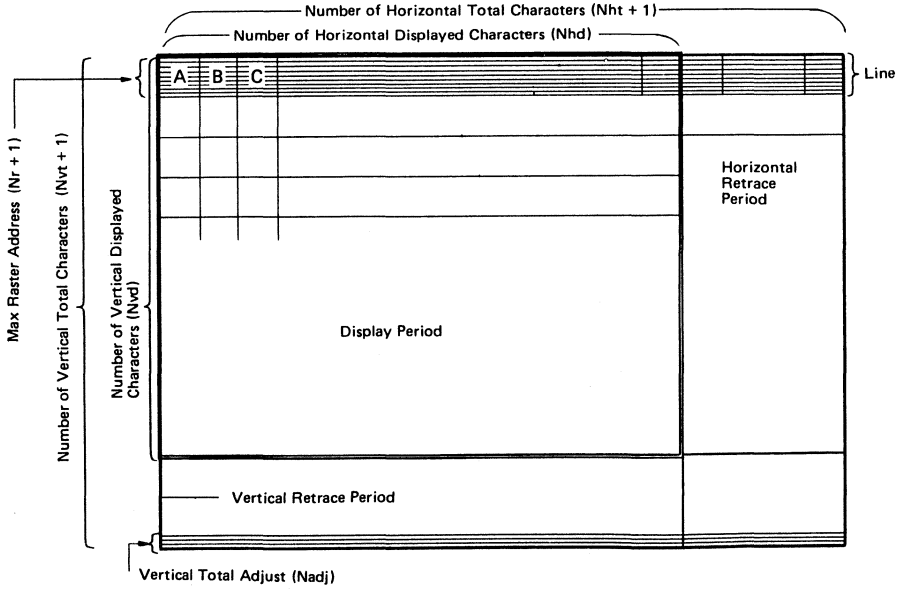


Figure 7 CRT Screen Format

Table 8 Programmed Values into the Registers

Register	Register Name	Value	Register	Register Name	Value
R0	Horizontal Total	Nht	R9	Max Raster Address	Nr
R1	Horizontal Displayed	Nhd	R10	Cursor Start Raster	
R2	Horizontal Sync Position	Nhsp	R11	Cursor End Raster	
R3	Sync Width	Nvsw, Nhsw	R12	Start Address (H)	0
R4	Vertical Total	Nvt	R13	Start Address (L)	0
R5	Vertical Total Adjust	Nadj	R14	Cursor (H)	
R6	Vertical Displayed	Nvd	R15	Cursor (L)	
R7	Vertical Sync Position	Nvsp	R16	Light Pen (H)	
R8	Interlace & Skew		R17	Light Pen (L)	

[NOTE] Nhd<Nht, Nvd<Nvt

The relation between values of Refresh Memory Address ($MA_0 \sim MA_{13}$) and Raster Address ($RA_0 \sim RA_4$) and the display position on the screen is shown in Fig. 16. Fig. 16 shows the case where the value of Start Address is 0.

- **Interlace Control**

Fig. 8 shows an example where the same character is displayed in the non-interlace mode, interlace sync mode, and interlace sync & video mode.

Non-interlace Mode Control

In non-interlace mode, each field is scanned duplicatedly. The values of raster addresses ($RA_0 \sim RA_4$) are counted up one from 0.

Interlace Sync Mode Control

In the interlace sync mode, raster addressed in the even field and the odd field are the same as addressed in the non-interlace mode. One character pattern is displayed mutually and its displayed position in the odd field is set at 1/2 raster space down from that in the even field.

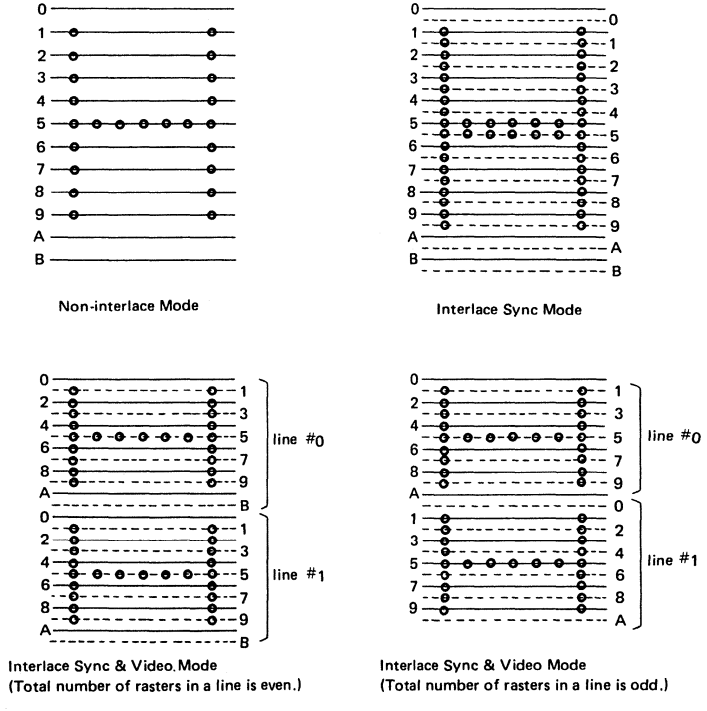


Figure 8 Example of Raster Scan Display

Interlace Sync & Video Mode Control

In interlace sync & video mode, the output raster address when the number of rasters is even is different from that when the number of rasters is odd.

Table 9 The Output of Raster Address in Interlace Sync & Video Mode

Total Number of Rasters in a Line \ Field		Even Field	Odd Field
Even		Even Address	Odd Address
Odd	Even Line*	Even Address	Odd Address
	Odd Line*	Odd Address	Even Address

* Internal line address begins from 0.

1) Total number of rasters in a line is even;

When number of rasters is programmed to be even, even raster address is output in the even field and odd raster address is output in the odd field.

2) Total number of rasters in a line is odd;

When total number of rasters is programmed to be odd, odd and even addresses are reversed according to the odd and even lines in each field. In this case, the difference in numbers of dots displayed between even field and odd field is usually smaller the case of 1). Then interlace can be displayed more stably.

[NOTE] The wide disparity of dots between number of dots between even field and odd field influences beam current of CRT. CRT, which has a stable high-voltage part, can make interlace display normal. On the contrary, CRT, which has unstable high-voltage part, moves deflection angle of beam current and also dots displayed in the even and odd fields may be shifted. Characters appears distroting on a border of the screen. So 2) programming has an effect to decrease such evil influences as mentioned above. Fig. 13 shows fine chart in each mode when interlace is performed.

- **Cursor Control**

Fig. 9 shows the display patterns where each value is programmed to the cursor start raster register and the cursor end raster register. Programmed values to the cursor start raster register and the cursor end raster register need to be under the following condition.

Cursor Start Raster Register ≦ Cursor End Raster Register ≦ Maximum Raster Address Register.

Time chart of CUDISP is shown in Fig. 14 and Fig. 15.

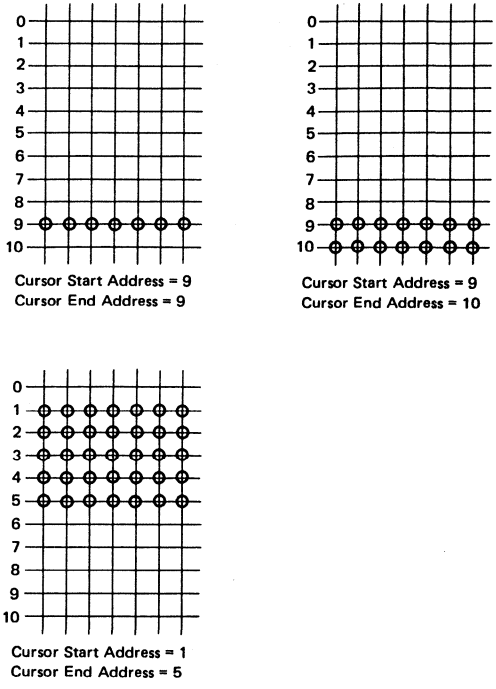


Figure 9 Cursor Control

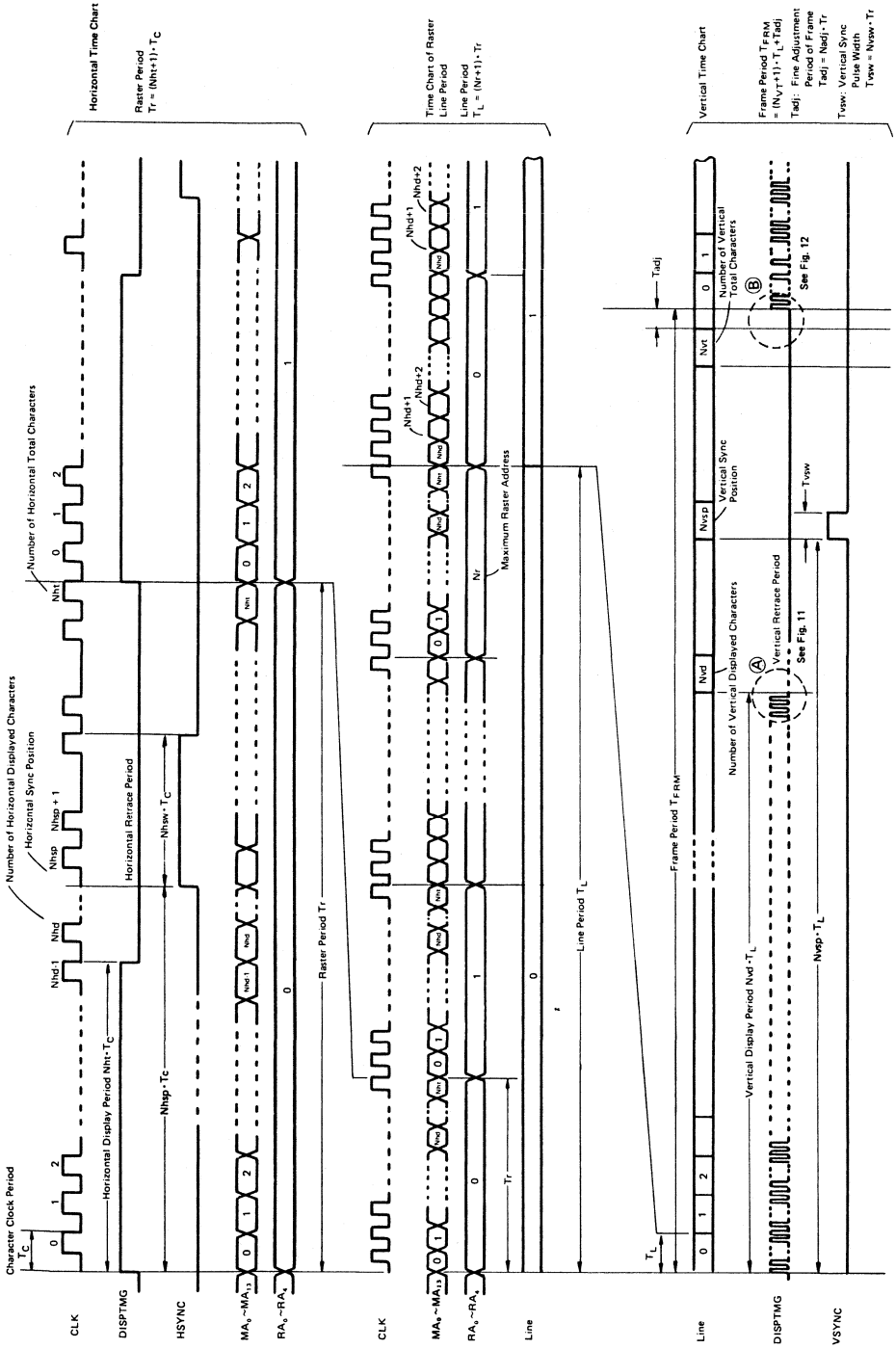


Figure 10 CRTC Time Chart

Output waveform of horizontal & vertical display in the case where values shown in Table 8 are Programmed to each register.

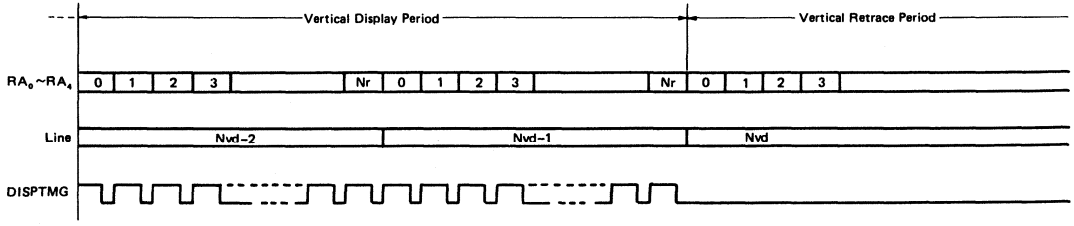


Figure 11 Switching from Vertical Display Period over to Vertical Retrace Period (Expansion of Fig. 10– Ⓐ)

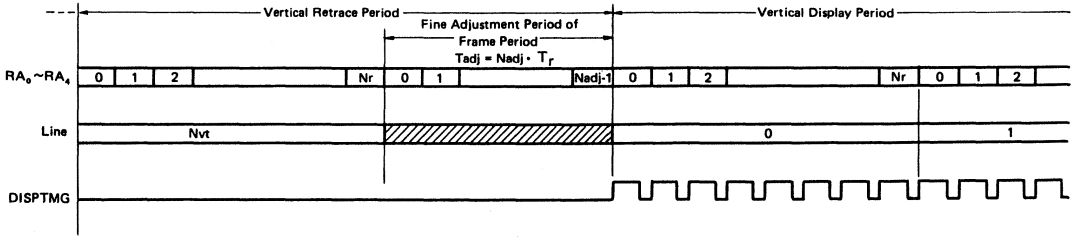


Figure 12 Fine Adjustment Period of Frame in Vertical Display (Expansion of Fig. 10– Ⓑ)

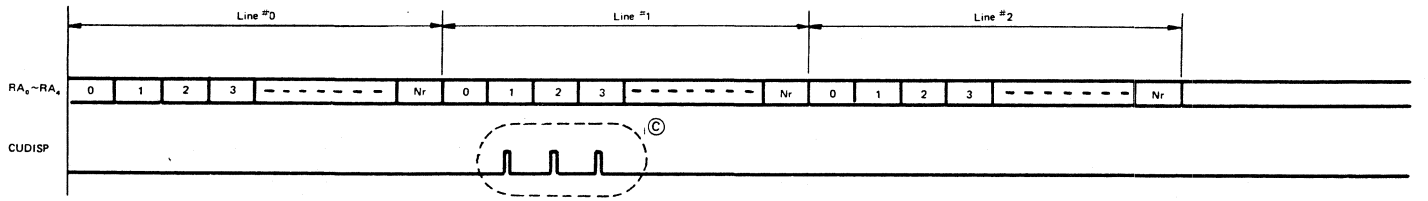
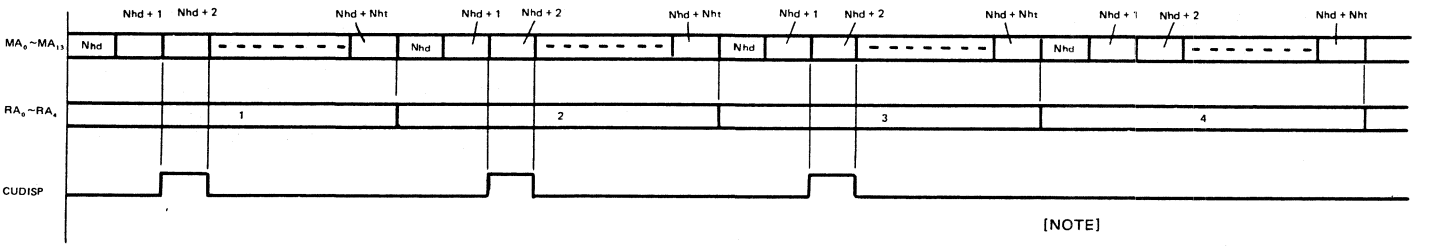


Figure 14 Relation between Line · Raster and CUDISP



[NOTE]

(Cursor register = Nhd+2
Cursor Start
Raster Register = 1
Cursor End
Raster Register = 3)
are Programmed in cursor display mode.

In blink mode, it is changed into display or non-display mode when field period is 16 or 32-time period.

Figure 15 CUDISP Timing (Expansion of Fig. 14. ©)

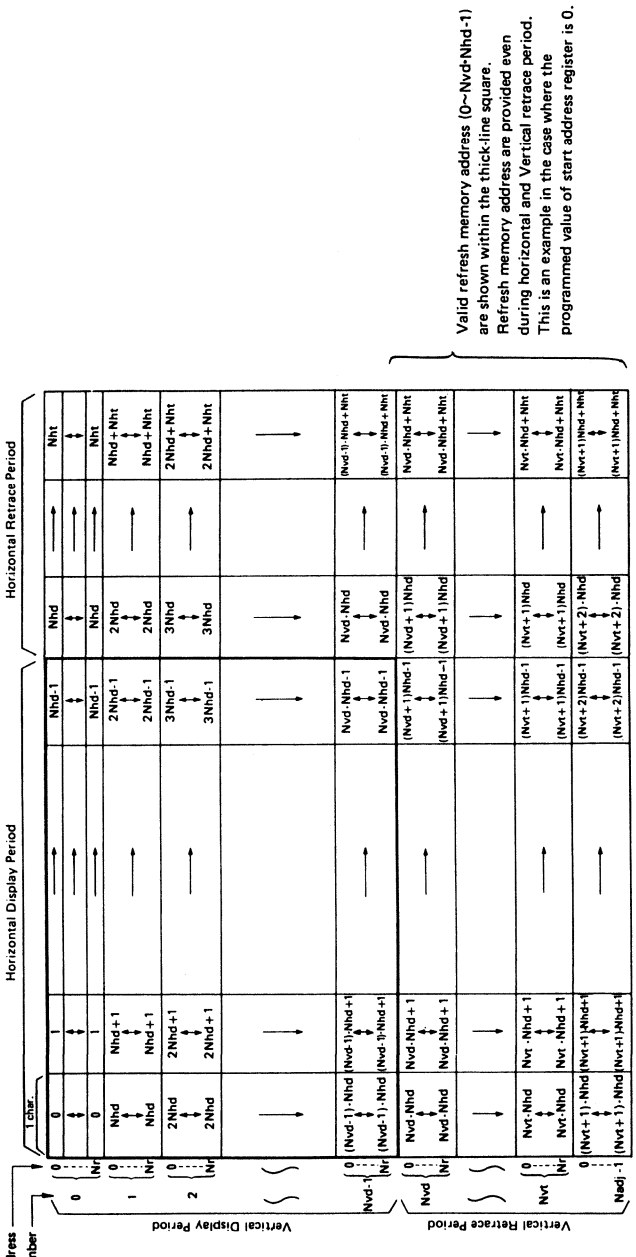


Figure 16 Refresh Memory Address (MA_0~MA_{13})

■ How to Use the CRTC

• Interface to MPU

As shown in Fig. 17, the CRTC is connected with the standard bus of MPU to control the data transfer between them. The CRTC address is determined by $\overline{CS}$ and RS, and the Read/Write operation is controlled by R/$\overline{W}$ and E. When $\overline{CS}$ is "Low" and RS is also "Low", the CRTC address register is selected. When $\overline{CS}$ is "Low" and RS is "High", one of 18 internal registers is selected.

$\overline{RES}$ is the system reset signal. When $\overline{RES}$ becomes "Low", the CRTC internal control logic is reset. But internal registers shown in Table 1 (R0~R17) are not affected by $\overline{RES}$ and remain unchanged.

The CRTC is designed so as to provide an interface to microcomputers, but adding some external circuits enables an interface to other data sources.

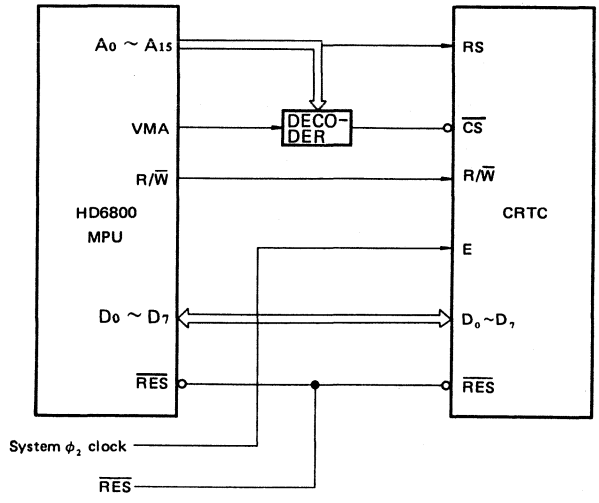


Figure 17 Interface to MPU

• Dot Timing Generating Circuit

CRTC's CLK input (21 pin) is provided with CLK which defines horizontal character time period from the outside. This CLK is generated by dot counter shown in Fig. 18. Fig. 18 shows a example of circuit where horizontal dot number of the character is "9". Fig. 19 shows the operation time chart of dot counter shown in Fig. 18. As this example shows explicitly, CLK is at "Low" level in the former half of horizontal character time and at "High" level in the latter half. It is necessary to be careful so as not to mistake this polarity.

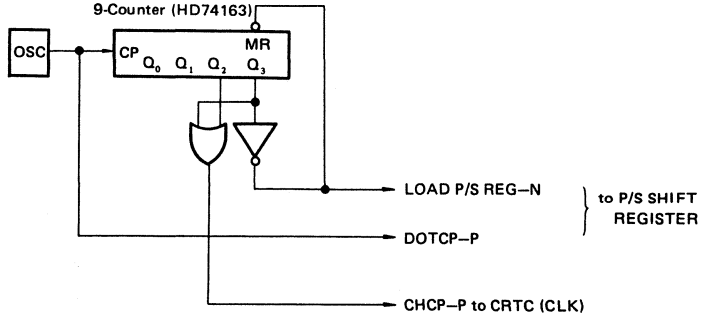


Figure 18 Example of Dot Counter

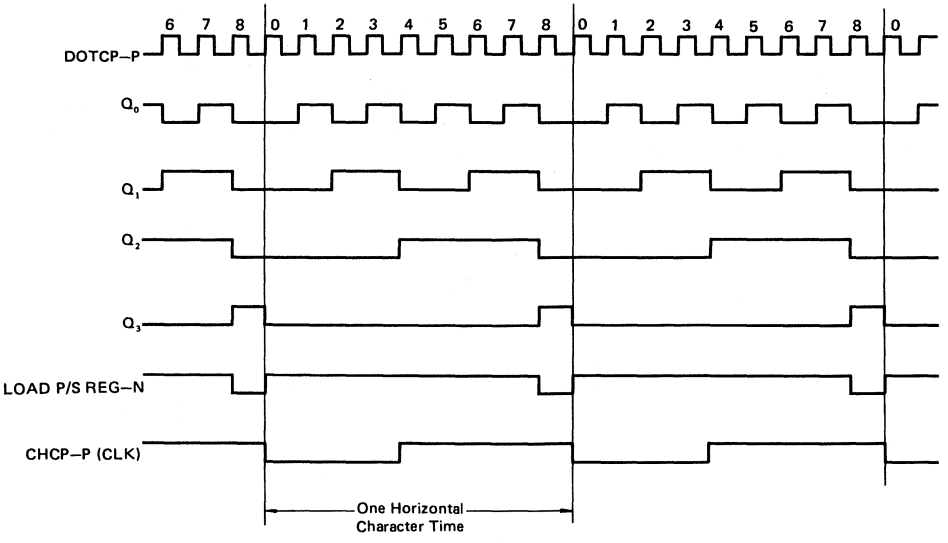


Figure 19 Time Chart of Dot Counter

■ INTERFACE TO DISPLAY CONTROL UNIT

Fig. 20 shows the interface between the CRTC and display control unit. Display control unit is mainly composed of Refresh Memory, Character Generator, and Video Control circuit. For refresh memory, 14 Memory Address line (0~16383) max are provided and for character generator, 5 Raster Address line (0~31) max are provided. For video control circuit, DISPTMG, CUDISP, HSYNC, and VSYNC are sent out. DISPTMG is used to control the blank period of video signal. CUDISP is used as video signal to display the cursor on the CRT screen. Moreover, HSYNC and VSYNC are used as drive signals respectively for CRT horizontal and vertical deflection circuits.

Outputs from video control circuit, (video signals and sync signals) are provided to CRT display unit to control the deflection and brightness of CRT, thus characters are displayed on the screen.

Fig. 21 shows detailed block diagram of display control unit. This shows how to use CUDISP and DISPTMG. CUDISP and DISPTMG should be used being latched at least one time at external flip-flop F1 and F2. Flip-flop F1 and F2 function to make one-character delay time so as to synchronize them with video signal from parallel-serial converter. High-speed D type flip-flop as TTL is used for this purpose. After being delayed at F1 and F2 DISPTMG is AND-ed with character video signal, and CUDISP is Or-ed with output from AND gate. By using this circuitry, blanking of horizontal and vertical retrace time is controlled. And cursor video is mixed with character video signal.

Fig. 21 shows the example in the case that both refresh memory and Character Generator can be accessed for horizontal one character time. Time chart for this case is shown in Fig. 24. This method is used when a few character needed to be displayed in horizontal direction on the screen.

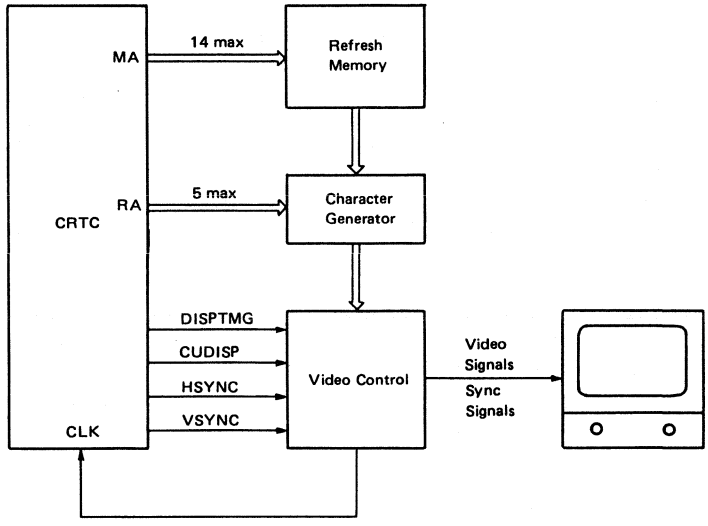


Figure 20 Interface to Display Control Unit

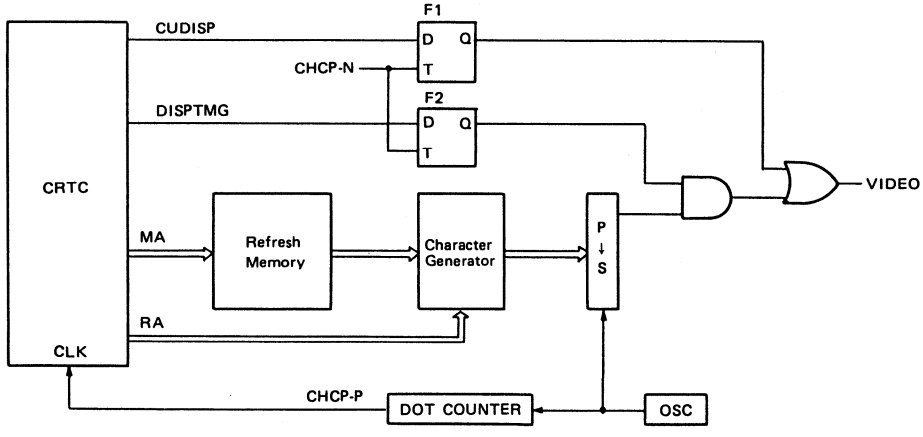


Figure 21 Display Control Unit (1)

When many characters are displayed in horizontal direction on the screen, and horizontal one-character time is so short that both refresh memory and Character Generator cannot be accessed, the circuitry shown in Fig. 22 should be used. In this case refresh memory output shall be latched and Character Generator shall be accessed at the next cycle. The time chart in this case is shown in Fig. 25. CUDISP and DISPTMG should be provided after being delayed by one-character time by using skew bit of interlace & skew register (R8). Moreover, when there are some troubles about delay time of MA during horizontal one-character time on high-speed display operation, system shown in Fig. 23 is adopted. The time chart in this case is shown in Fig. 26. Character video signal is delayed for two-character time because each MA outputs and refresh memory outputs are latched, and they are made to be in phase with CUDISP and DISPTMG by delaying for two-character time. Table 10 shows the circuitry selection standard of display units.

Table 10 Circuitry Standard of Display Control Unit

Case	Relation among t_{CH} Refresh Memory and Character Generator	Block Diagram	Interlace & Skew Register Bit Programming			
			C1	C0	D1	D0
1	$t_{CH} >$ RM Access + CG Access + t_{MAD}	Fig. 21	0	0	0	0
2	RM Access + CG Access + $t_{MAD} \geqq t_{CH} >$ RM Access + t_{MAD}	Fig. 22	0	1	0	1
3	RM Access + $t_{MAD} \geqq t_{CH} >$ RM Access	Fig. 23	1	0	1	0

t_{CH} : CHCP Period; t_{MAD} : MA Delay
RM: Refresh Memory CG: Character Generator

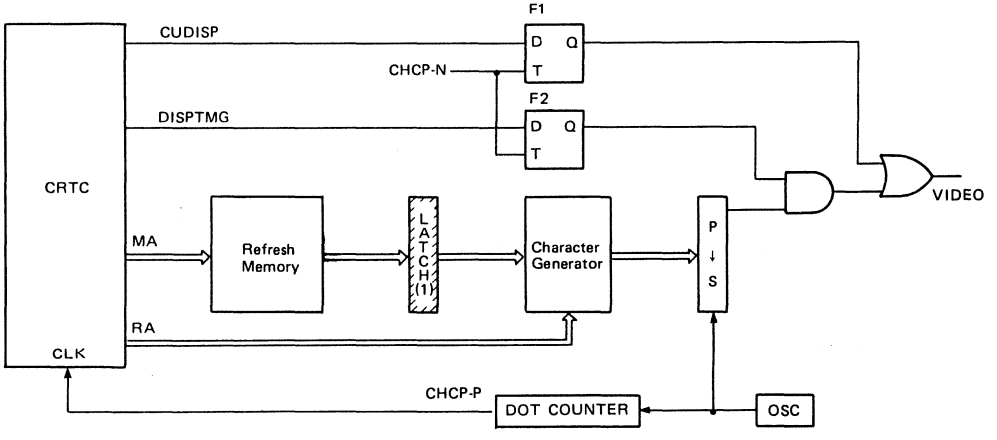


Figure 22 Display Control Unit (2)

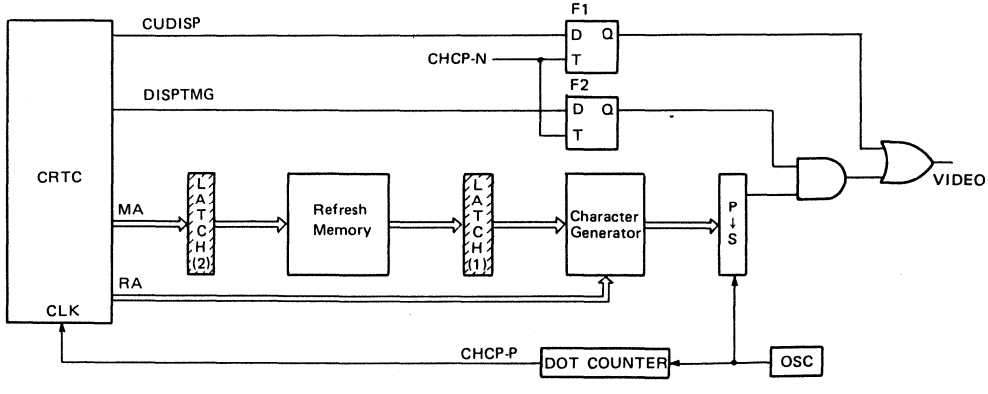


Figure 23 Display Control Unit (For high-speed display operation) (3)

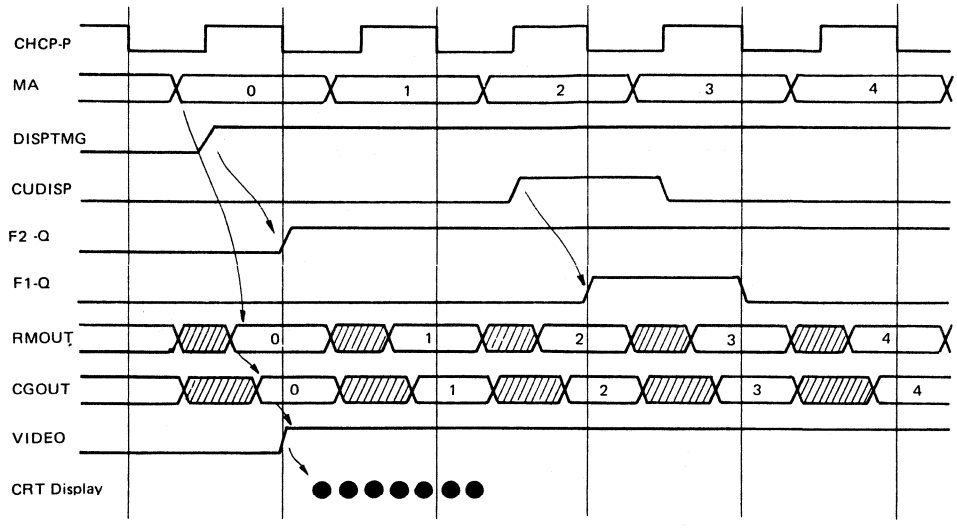


Figure 24 Time Chart of Display Control Unit (1)

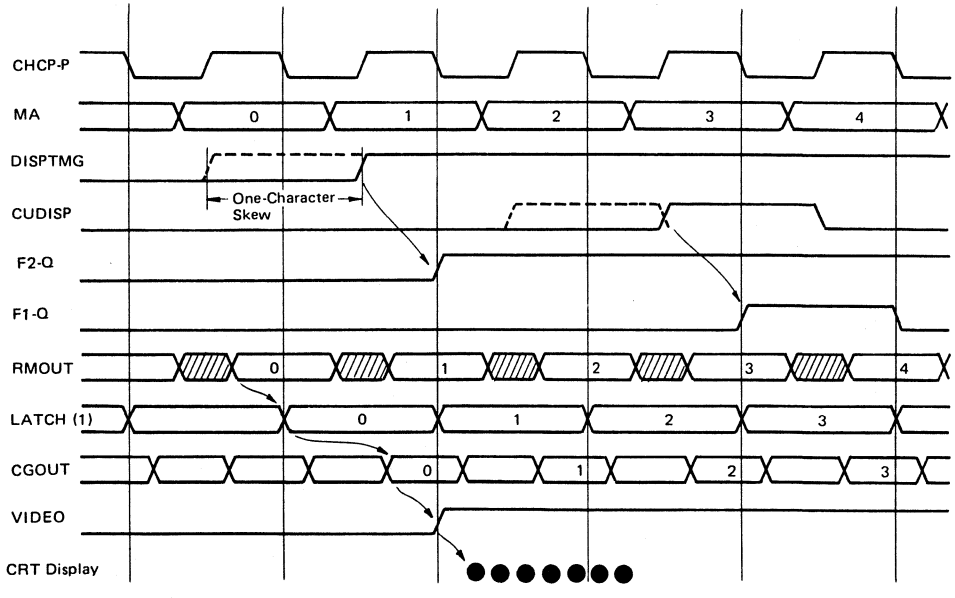


Figure 25 Time Chart of Display Control Unit (2)

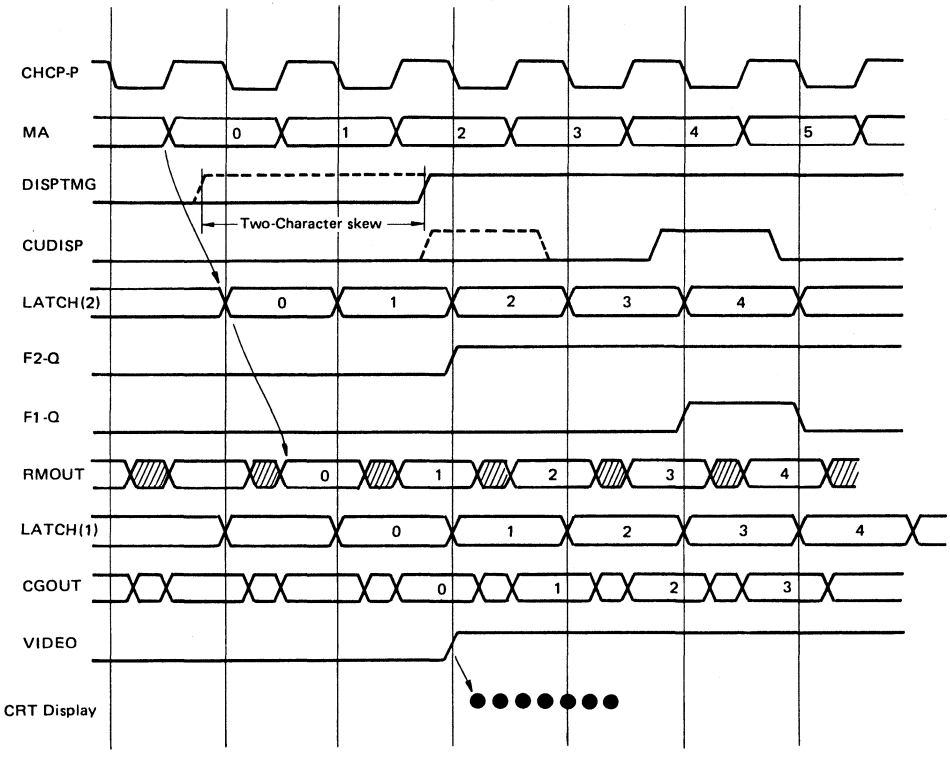


Figure 26 Time Chart of Display Unit (3)

■ HOW TO DECIDE PARAMETERS SET ON THE CRTC

• How to Decide Parameters Based on Specification of CRT Display Unit (Monitor)

Number of Horizontal Total Characters

Horizontal deflection frequency f_h is given by specification of CRT display unit. Number of horizontal total characters is determined by the following equation.

$$f_h = \frac{1}{t_C (N_{ht} + 1)}$$

where,

t_C : Cycle Time of CLK (Character Clock)

N_{ht} : Programmed Value of Horizontal Total Register (R0)

Number of Vertical Total Characters

Vertical deflection frequency is given by specification of CRT display unit. Number of vertical Total characters is determined by the following equation.

1) Non-interlace Mode

$Rt = (Nvt + 1)(Nr + 1) + Nadj$

2) Interlace Sync Mode

$Rt = (Nvt + 1)(Nr + 1) + Nadj + 0.5$

3) Interlace Sync & Video Mode

$$Rt = \frac{(Nvt + 1)(Nr + 2) + 2Nadj}{2} \quad \cdots\cdots\cdots\cdots \text{(a)}$$

$$Rt = \frac{(Nvt + 1)(Nr + 2) + 2Nadj + 1}{2} \quad \cdots\cdots\cdots\cdots \text{(b)}$$

(a) is applied when both total numbers of vertical characters (Nvt + 1) and that of rasters in a line (Nr + 2) are odd.

(b) is applied when total number of rasters (Nr + 2) is even, or when (Nr + 2) is odd and total number of vertical characters (Nvt + 1) is even.

where,

Rt : Number of Total Rasters per frame (Including retrace period)

Nvt : Programmed Value of Vertical Total Register (R4)

Nr : Programmed Value of Maximum Raster Address Register (R9)

Nadj : Programmed Value of Vertical Total Adjust Register (R5)

Horizontal Sync Pulse Width

Horizontal sync pulse width is programmed to low order 4-bit of horizontal sync width register (R3) in unit of horizontal character time. Programmed value can be selected within from 1 to 15.

Horizontal Sync Position

As shown in Fig. 27, horizontal sync position is normally selected to be in the middle of horizontal retrace period. But there are some cases where is optimum sync position is not located in the middle of horizontal retrace period according to specification of CRT. Therefore, horizontal sync position should be determined by specification of CRT. Horizontal sync pulse position is programmed in unit of horizontal character time.

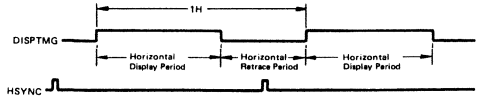


Figure 27 Time Chart of HSYNC

Vertical Sync Pulse Width

Vertical Sync Pulse Width is programmed to high order 4-bit of vertical sync pulse width register (R3) in unit of raster period. Programmed value can be selected within from 1 to 16.

Vertical Sync Position

As shown in Fig. 28, vertical sync position is normally selected to be in the middle of vertical retrace period. But there are some cases where its optimum sync position is not located in the middle of vertical retrace period according to specification of CRT. Therefore, vertical sync position should be determined by specification of CRT. Vertical sync pulse position is programmed to vertical sync position register (R7) in unit of line period.

- **How to Decide Parameters Based on Screen Format**

Dot Number of Characters (Horizontal)

Dot number of characters (horizontal) is determined by character font and character space. An example is shown in Fig. 29. More strictly, dot number of characters (horizontal) N is determined by external N-counter. Character space is set by means shown in Fig. 30.

Dot Number of Characters (Vertical)

Dot number of characters (vertical) is determined by characters font and line space. An example is shown in Fig. 29. Dot number of characters (vertical) is programmed to maximum raster address (R9) of CRTC.

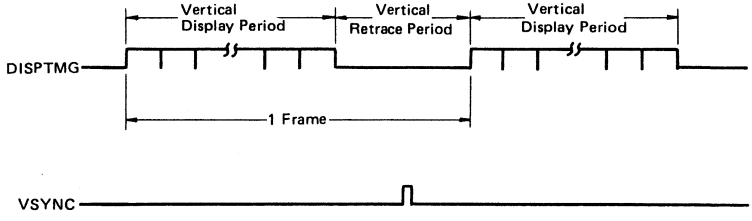


Figure 28 Time Chart of VSYNC

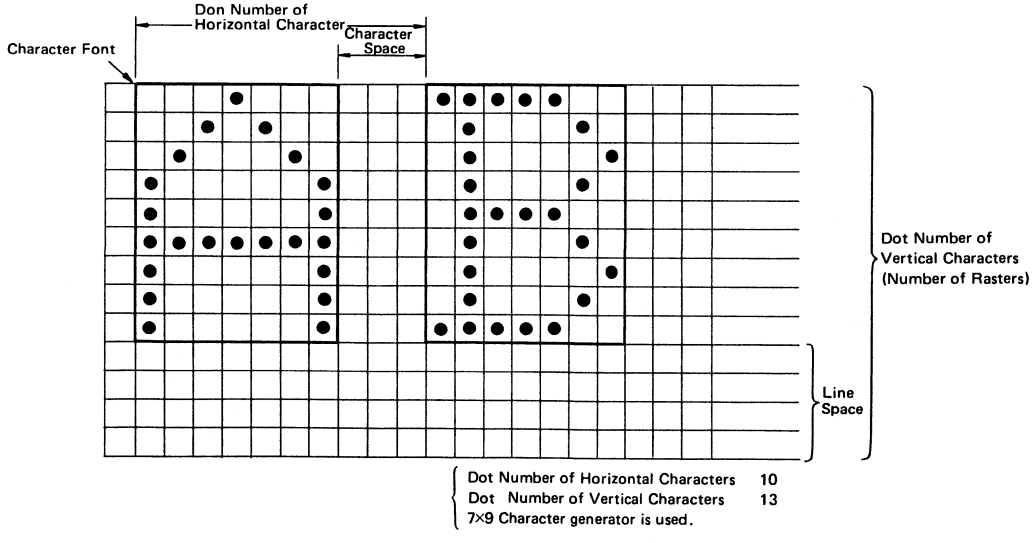


Figure 29 Dot Number of Horizontal and Vertical Characters

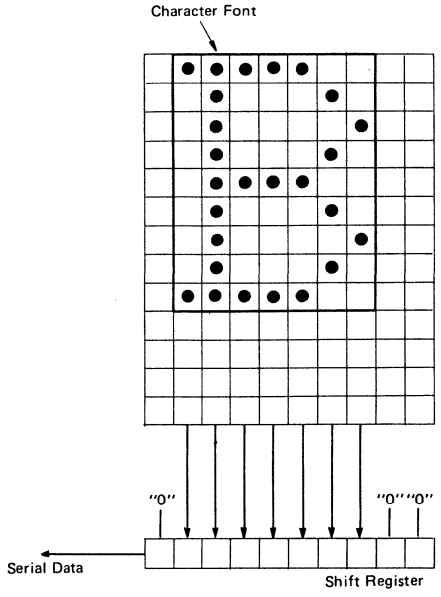


Figure 30 How to Make Character Space

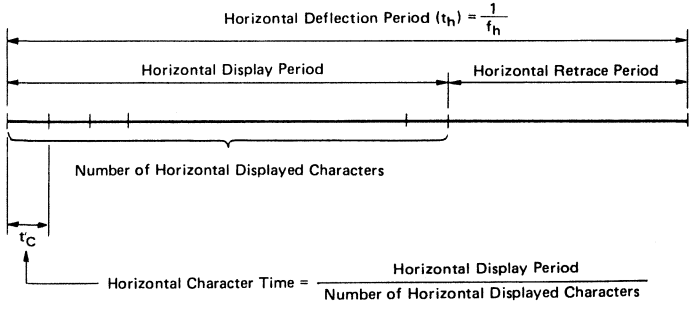


Figure 31 Number of Horizontal Displayed Characters

Number of Horizontal Displayed Characters

Number of horizontal displayed characters is programmed to horizontal displayed register (R1) of the CRTC. Programmed value is based on screen format. Horizontal display period, which is given by specification of horizontal deflection frequency and horizontal retrace period of CRT display unit, determines horizontal character time, being divided by number of horizontal displayed characters. Moreover, its cycle time and access time which are necessary for CRT display system are determined by horizontal character time.

Number of Vertical Displayed Characters

Number of vertical displayed characters is programmed to vertical displayed register (R6). Programmed value is based on screen format. As specification of vertical deflection frequency of CRT determines number of total rasters (Rt) including vertical retrace period and the relation between number of vertical displayed character and total number of rasters on a screen is as mentioned above, CRT which is suitable for desired screen format should be selected.

For optimum screen format, it is necessary to adjust number of rasters per line, number of vertical displayed characters, and total adjust raster (Nadj) within specification of vertical deflection frequency.

Scan Mode

The CRTC can program three-scan modes shown in Table 11 to interlace mode register (R8). An example of character display in each scan mode is shown in Fig. 8.

Table 11 Program of Scan Mode

V	S	Scan Mode	Main Usage
0	0	Non-interlace	Normal Display of Characters & Figures
1	0		
0	1	Interlace Sync	Fine Display of Characters & Figures
1	1	Interlace Sync & Video	Display of Many Characters & Figures Without Using High-resolution CRT

[NOTE] In the interlace mode, the number of times per sec. in raster scanning on one spot on the screen is half as many as that in non-interlace mode. Therefore, when persistence of luminescence is short, flickering may happen. It is necessary to select optimum scan mode for the system, taking characteristics of CRT, raster scan speed, and number of displayed characters and figures into account.

Cursor Display Method

Cursor start raster register and cursor end raster register (R10, R11) enable programming the display modes shown in Table 7 and display patterns shown in Fig. 9. Therefore, it is possible to change the method of cursor display dynamically according to the system conditions as well as to realize the cursor display that meets the system requirements.

Start Address

Start address resisters (R12, R13) give an offset to the address of refresh memory to read out. This enables paging and scrolling easily.

Cursor Register

Cursor registers (R14, R15) enable programming the cursor display position on the screen. As for cursor address, it is not X, Y address but linear address that is programmed.

■ Applications of the CRTC

• Monochrome Character Display

Fig. 32 shows a system of monochrome character display. Character clock signal (CLK) is provided to the CRTC through OSC and dot counter. It is used as basic clock which drives internal control circuits. MPU is connected with the CRTC by standard bus and controls the CRTC initialization and read/write of internal registers.

Refresh memory is composed of RAM which has capacity of one frame at least and the data to be displayed is coded and stored. The data to refresh memory is changed through MPU

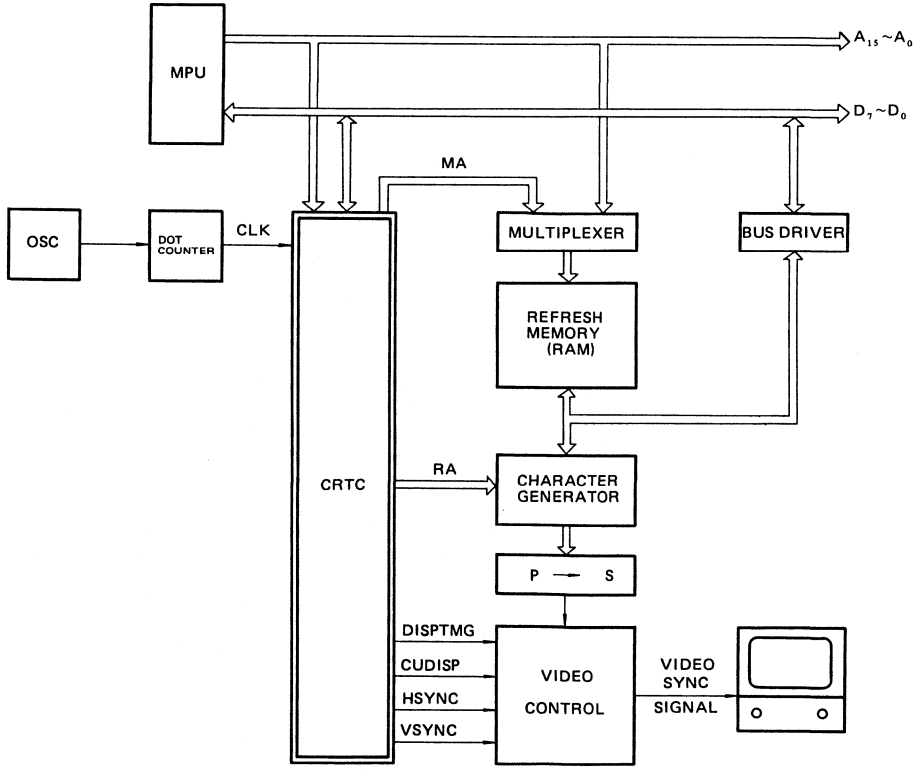


Figure 32 Monochrome Character Display

bus, while refresh memory is read out successively by the CRTC to display a static pattern on the screen. Refresh memory is accessed by both MPU and the CRTC, so it needs to change its address selectively by multiplexer. The CRTC has 14 MA (Memory Address output), but in fact some of them that are needed are used according to capacity of refresh memory.

Code output of refresh memory is provided to character generator. Character generator generates a dot pattern of a specified raster of a specified character in parallel according to code output from refresh memory and RA (Raster Address output) from the CRTC. Parallel-serial converter is normally composed of shift register to convert output of character generator into a serial dot pattern. Moreover, DISPTMG, CUDISP, HSYNC, and VSYNC are provided to video control circuit. It controls blanking for output of parallel-serial converter, mixes these signals with cursor video signal, and generates sync signals for an interface to monitor.

- **Color Character Display**

Fig. 33 shows a system of color character display. In this example, a 3-bit color control bit (R, G, B) is added to refresh memory in parallel with character code and provided to video control circuit. Video control circuit controls coloring as well as blanking and provides three primary color video signals (R, G, B signals) to CRT display device to display characters in seven kinds of color on the screen.

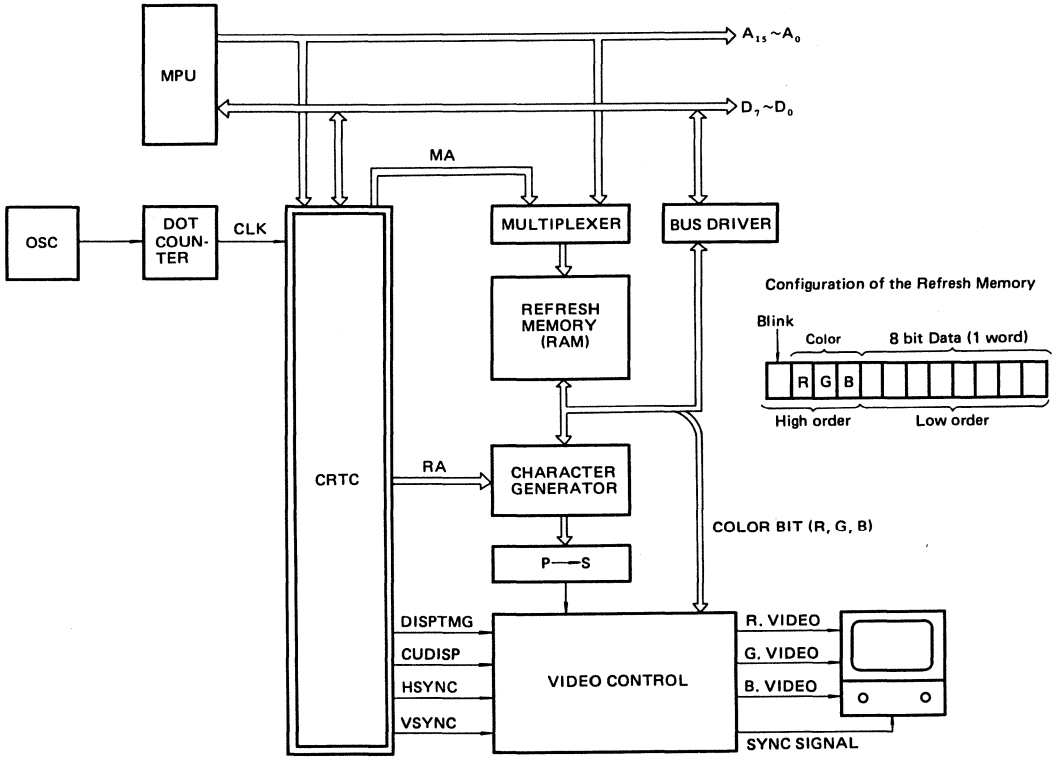


Figure 33 Color Character Display

- **Color limited Graphic Display**

Limited graphic display is to display simple figures as well as character display by combination of picture element which are defined in unit of one character.

As shown in Fig. 34, graphic pattern generator is set up in parallel with character generator and output of these generators are wire-ORed. Which generator is accessed depends on coded output of refresh memory.

In this example, graphic pattern generator adopts ROM, so only the combination of picture elements which are programmed to it is used for this graphic display system. Adopting RAM instead of ROM enables dynamically writable symbols in any combination on one display by changing the contents of them.

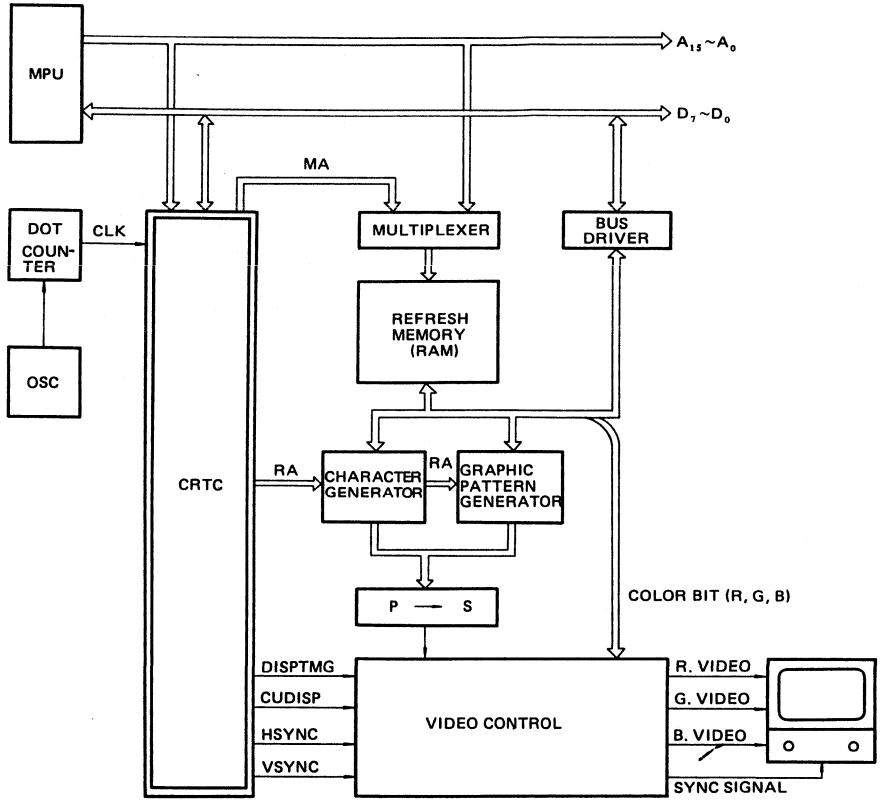


Figure 34 Color Limited Graphic Display

- **Monochrome Full Graphic Display**

Fig. 35 shows a system of monochrome full graphic display. While simple graphic display is figure display by combination of picture elements in unit of 1 picture elements, full graphic display is display of any figures in unit of 1 dot. In this case, refresh memory is dot memory that stores all the dot patterns, so its output is directly provided to parallel-serial converter to be displayed. Dot memory address to refresh the screen is set up by combination of MA and RA of CRTC.

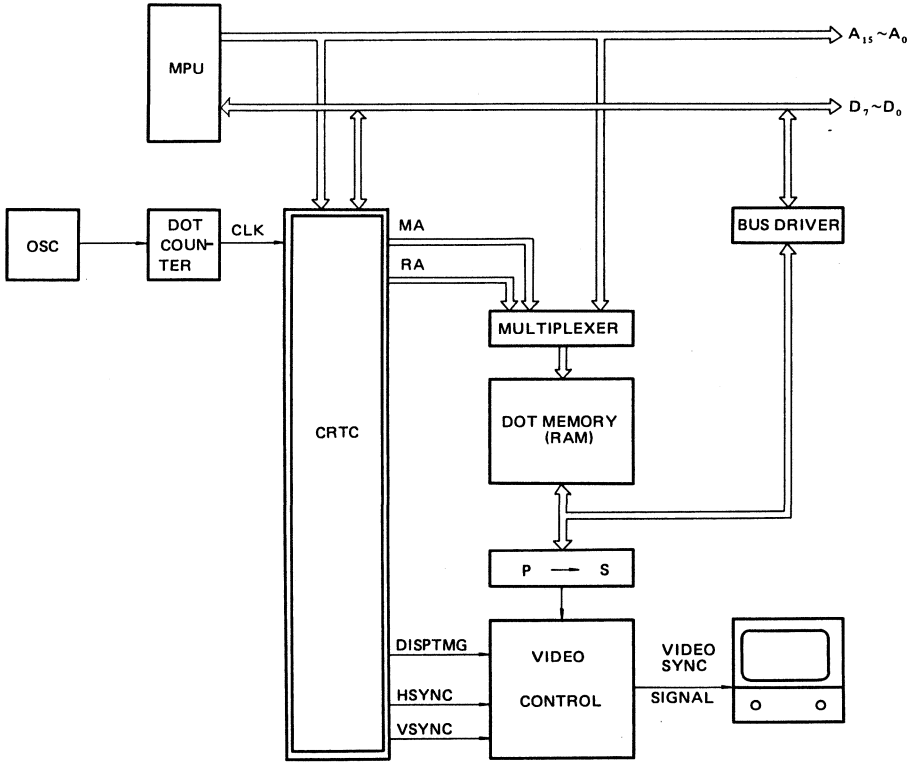


Figure 35 Monochrome Full Graphic Display

Fig. 36 shows an example of access to refresh memory by combination of MA and RA. Fig. 36 shows a refresh memory address method for full graphic display. Correspondence between dot on the CRT screen and refresh memory address is shown in Fig. 37.

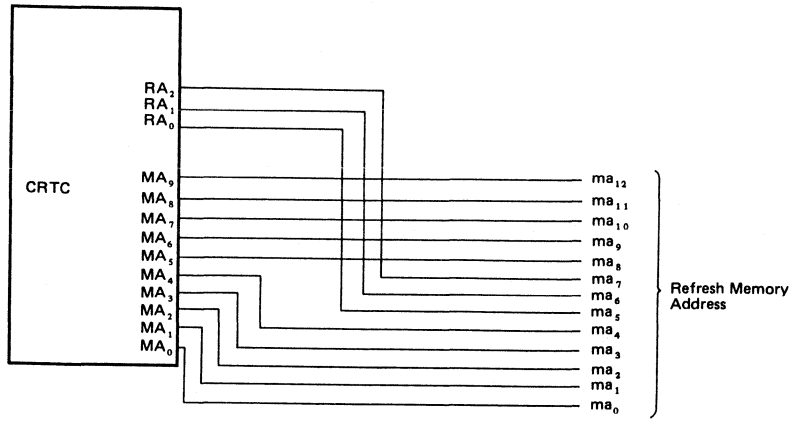


Figure 36 Refresh Memory Address Method for Full Graphic Display

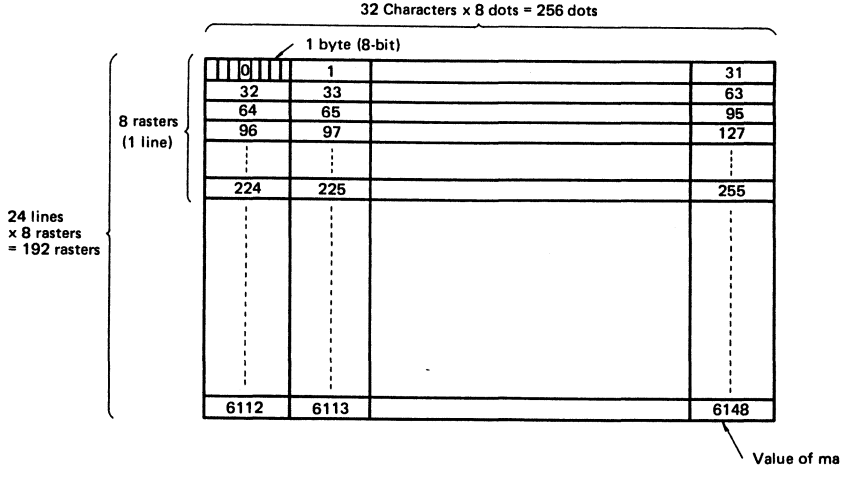


Figure 37 Memory Address and Dot Display Position on the Screen for Full Graphic Display

- **Color Full Graphic Display**

Fig. 38 shows a system of color full graphic display by 7-color display. Refresh memory is composed of three dot memories which are respectively used for red, green, and blue. These dot memories are read out in parallel at one time and their output is provided to three parallel-serial converters. Then video control circuit adds the blanking control to output of these converters and provides it to CRT display device as red, green, and blue video signals with sync signals.

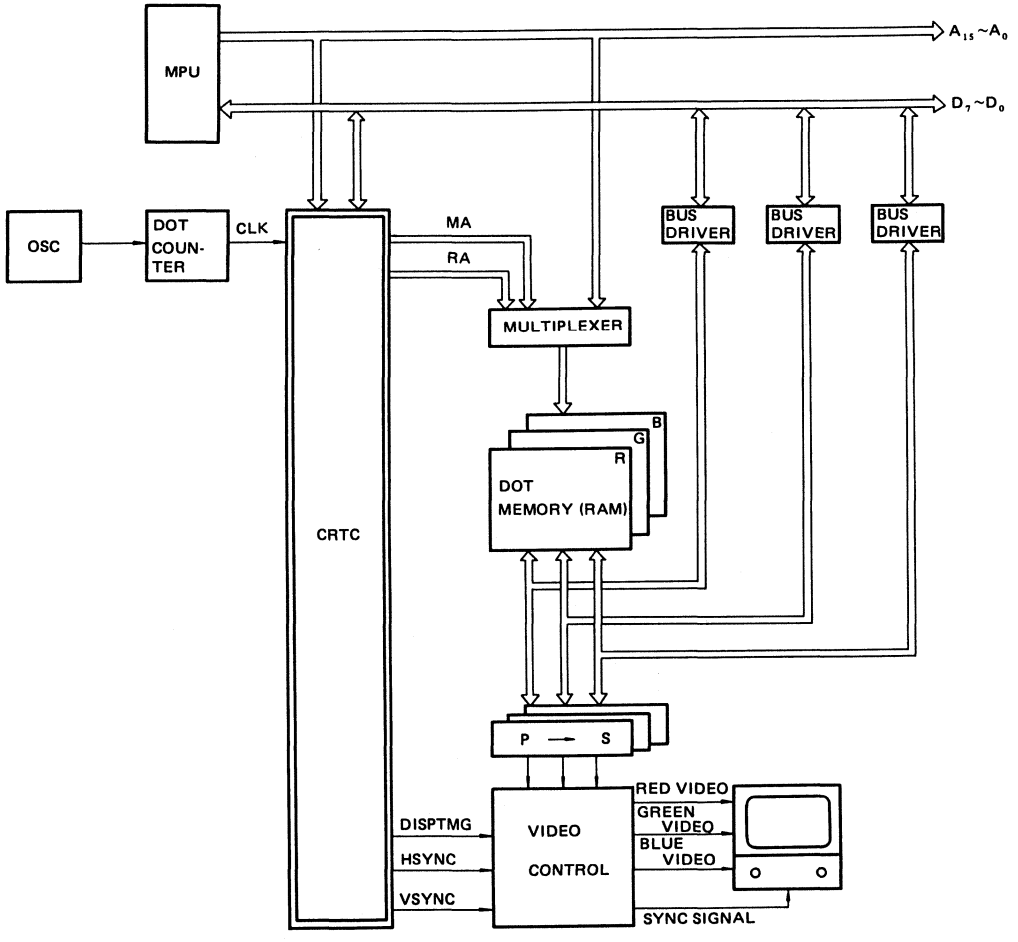


Figure 38 Color Full Graphic Display

- **Cluster Control of CRT Display**

The CRTC enables cluster control that is to control CRT display of plural devices by one CRTC. Fig. 39 shows a system of cluster control. Each display control unit has refresh memory, character generator, parallel-serial converter, and video control circuit separately, but these are controlled together by the CRTC.

In this system, it is possible for plural CRT display devices to have their own display separately.

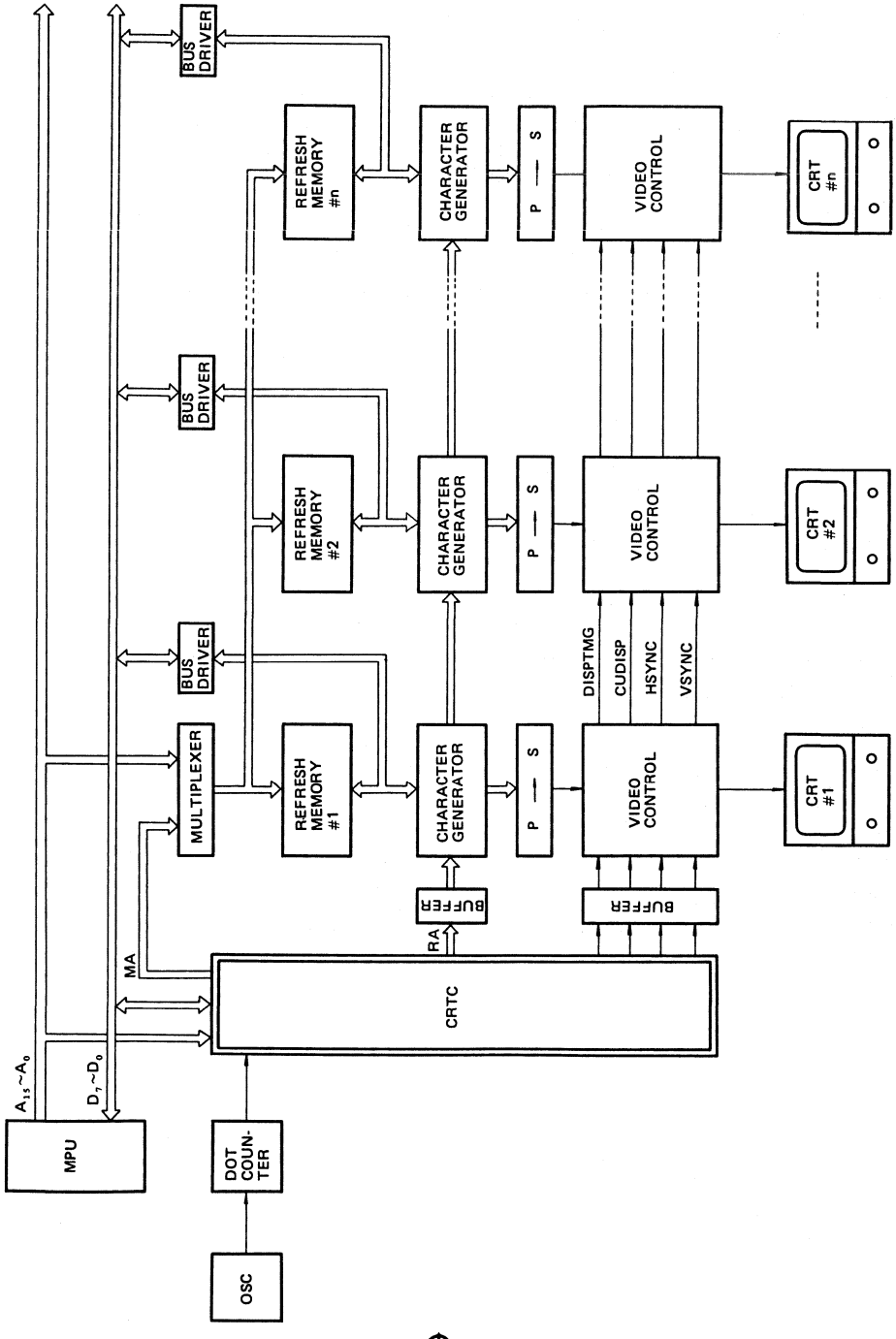


Figure 39 Cluster Control by the CRTC

■ EXAMPLES OF APPLIED CIRCUIT OF THE CRTC

Fig. 41 shows an example of application of the CRTC to monochrome character display. Its specification is shown in Table 12. Moreover, specification of CRT display unit is shown in Table 13 and initializing values for the CRTC are shown in Table 14.

Table 12 Specification of Applied Circuit

Item	Specification
Character Format	5 × 7 Dot
Character Space	Horizontal : 3 Dot Vertical : 5 Dot
One Character Time	1 μs
Number of Displayed Characters	40 characters × 16 lines = 640 characters
Access Method to Refresh Memory	Snychronous Method (DISPTMG Read)
Refresh Memory	640 B
Address Map	2^{15} 2^{14} 2^{13} 2^{12} 2^{11} 2^{10} 2^{9} 2^{8} 2^{7} 2^{6} 2^{5} 2^{4} 2^{3} 2^{2} 2^{1} 2^{0} Refresh Memory: 0 0 0 0 0 0 * * * * * * * * * * CRTC Address Register: 0 0 0 1 0 0 × × × × × × × × × 0 CRTC Control Register: 0 0 0 1 0 0 × × × × × × × × × 1 × · · · don't care, * · · · 0 or 1
Synchronization Method	HVSYNC Method

Table 13 Specification of Character Display

Item	Specification
Scan Mode	Non-interlace
Horizontal Deflection Frequency	15.625 kHz
Vertical Deflection Frequency	60.1 Hz
Dot Frequency	8 MHz
Character Dot (Horizontal × Vertical)	8 × 12 (Character Font 5 × 7)
Number of Displayed Characters (Row × Line)	40 × 16
HSYNC Width	4 μs
VSYNC Width	3 H
Cursor Display	Raster 9 ~ 10, Blink 16 Field Period
Paging, Scrolling	Not used

Table 14 Initializing Values for Character Display

Register	Name	Symbol	Initializing Value Hex (Decimal)
R0	Horizontal Total	Nht	3F (63)
R1	Horizontal Displayed	Nhd	28 (40)
R2	Horizontal Sync Position	Nhsp	34 (52)
R3	Sync Width	Nvsw, Nhsw	34
R4	Vertical Total	Nvt	14 (20)
R5	Vertical Total Adjust	Nadj	08 (8)
R6	Vertical Displayed	Nvd	10 (16)
R7	Vertical Sync Position	Nvsp	13 (19)
R8	Interlace & Skew		00
R9	Maximum Raster Address	Nr	0B (11)
R10	Cursor Start Raster	B, P, NCSTART	49
R11	Cursor End Raster	NCEND	0A (10)
R12	Start Address (H)		00 (0)
R13	Start Address (L)		00 (0)
R14	Cursor (H)		00 (0)
R15	Cursor (L)		00 (0)

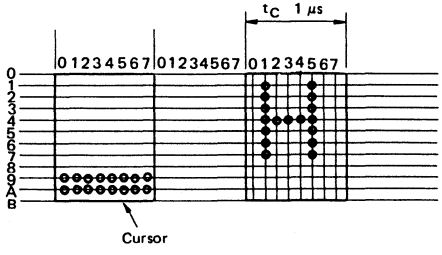


Figure 40 Non-interlace Display (Example)

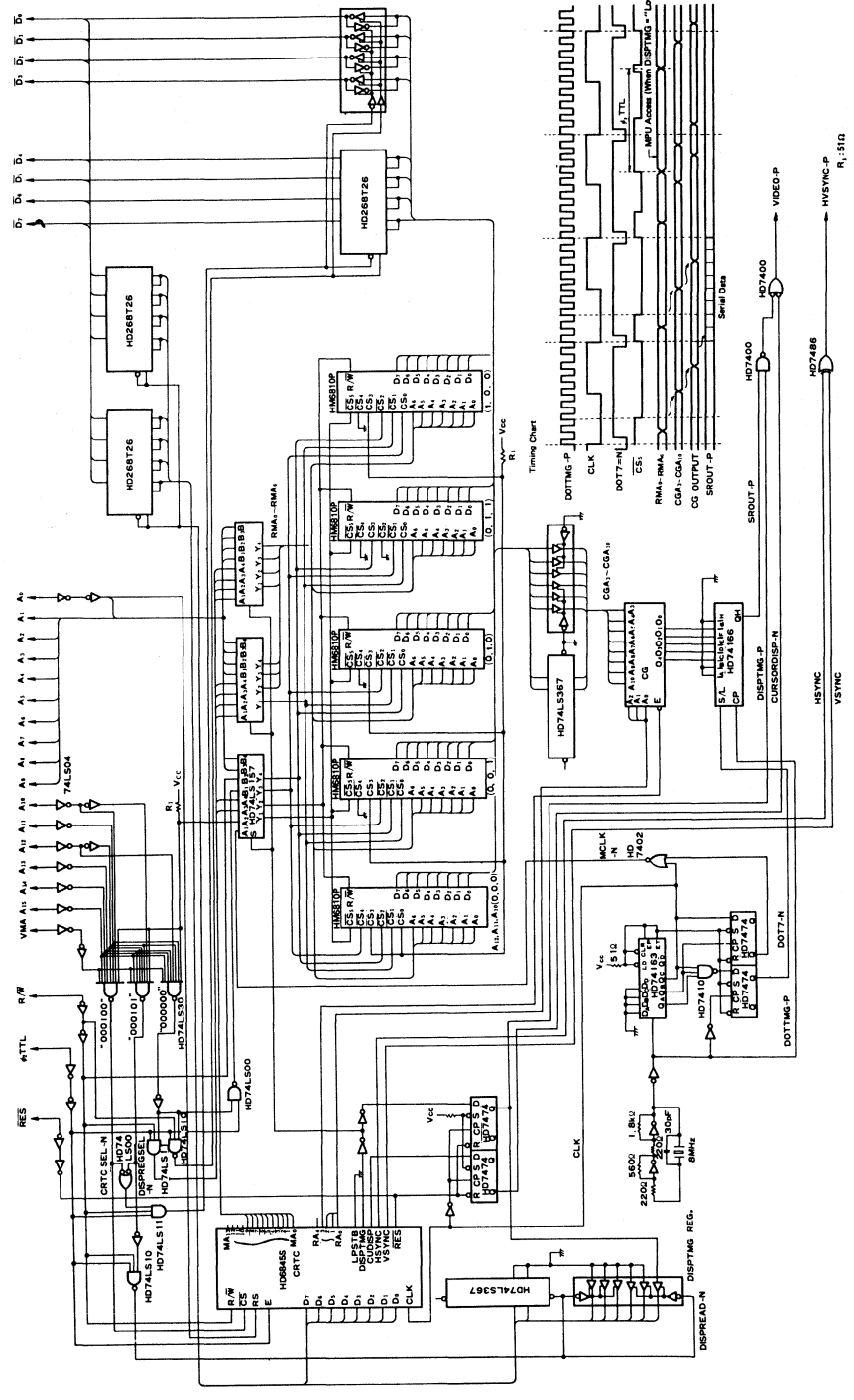


Figure 41 Example of Applied Circuit of the CRTC (Monochrome Character Display)

■ **DISPLAY SEQUENCE AFTER $\overline{RES}$ RELEASE OF HD6845S**

HD6845S starts the display operation immediately after the release of $\overline{RES}$. The operation at the first field is different from the normal subsequent display operation.

[Operation at the first field after the $\overline{RES}$ release]

(1) DISPTMG and CUDISP are not output. (They remain at "Low" level. The display is inhibited.)

(2) The data programmed in the start address register is not used. (MA and RA start at "0".)

(3) The sequences are shown in the following figures.

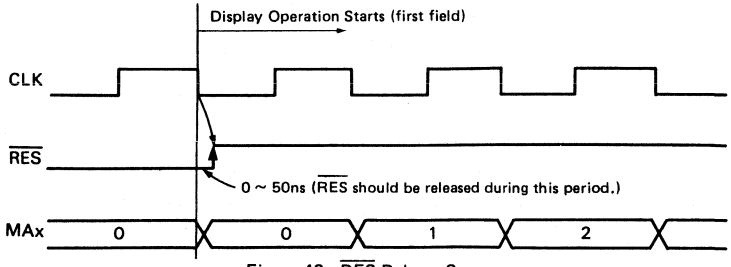


Figure 42 $\overline{RES}$ Release Sequence

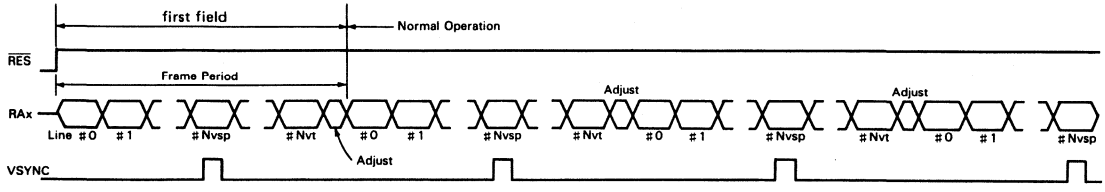
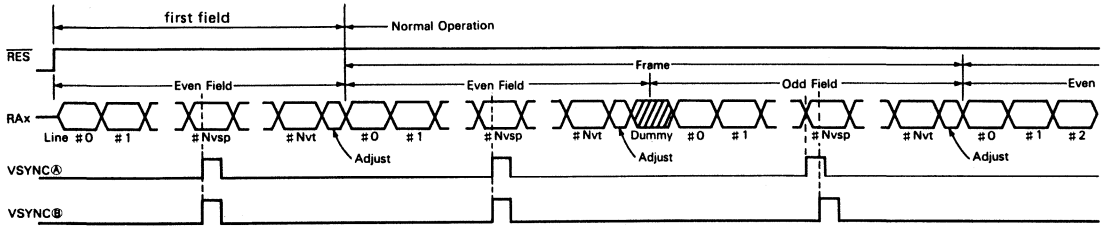
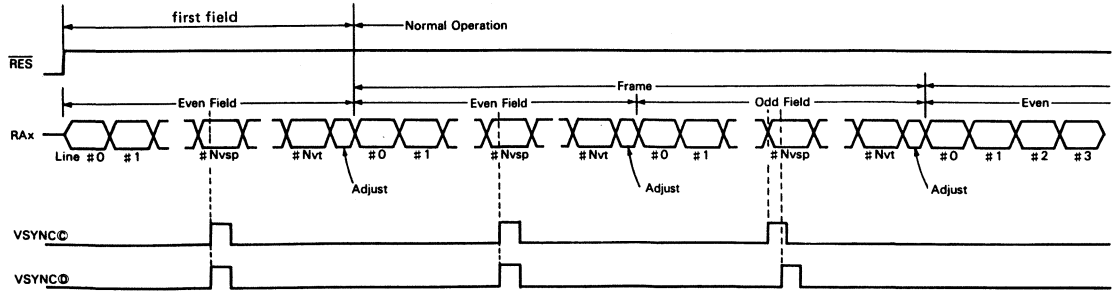


Figure 43 $\overline{RES}$ Release Sequence in The Non-interlace Mode



VSYNCⒶ : Interlace Sync Control
Interlace Sync & Video Control (Nr+2=Even)
Interlace Sync & Video Control (Nr+2=Odd, Nvt=Odd, Nvsp=Even)
VSYNCⒷ : Interlace Sync & Video Control (Nr+2=Odd, Nvt=Odd, Nvsp=Odd)

Figure 44 $\overline{RES}$ Release Sequence in The Interlace Mode (1)



VSYNCⒸ : Interlace Sync & Video Control (Nr+2=Odd, Nvt=Even, Nvsp=Even)
VSYNCⒹ : Interlace Sync & Video Control (Nr+2=Odd, Nvt=Even, Nvsp=Odd)

Figure 45 $\overline{RES}$ Release Sequence in The Interlace Mode (2)

■ ANOMALOUS OPERATIONS IN HD6845S CAUSED BY REWRITING REGISTERS DURING THE DISPLAY OPERATION*

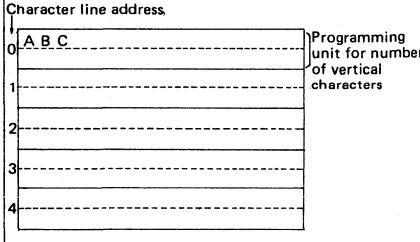
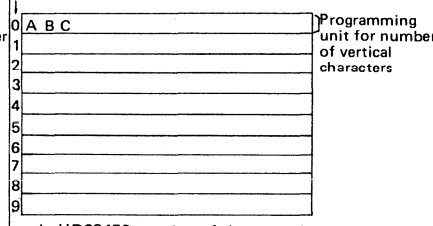
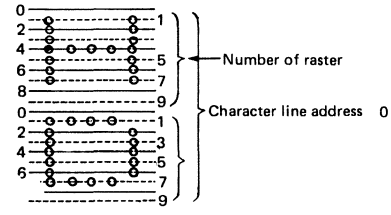
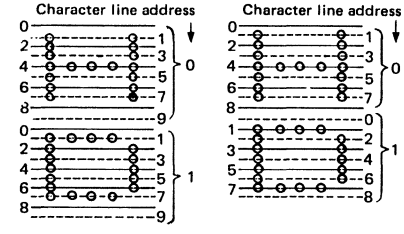
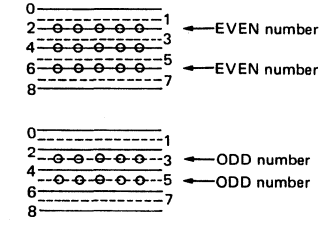
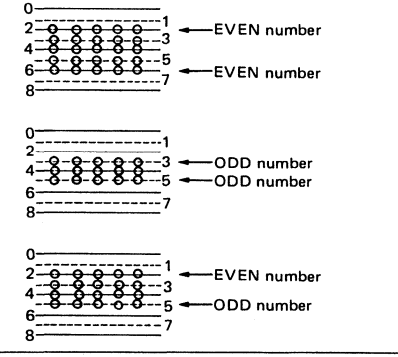
Register #	Register Name	Anomalous operations caused by rewriting registers & Conditions to avoid those operations	Rewriting** OK or NG
R0	Horizontal Total	The horizontal scan period is disturbed.	X
R1	Horizontal Displayed	There are some cases where the width of DISPTMG becomes shorter than the programmed value at the moment of a rewrite operation. An error operation occurs only during one raster period.	O
R2	Horizontal Sync Position	There are some cases where HSYNC is placed on the position different from the programmed value or the noise is output.	X
R3	Sync Width	When a rewrite operation is performed at a "High" level on HSYNC pulse or VSYNC pulse, there are some cases where the width pulse becomes shorter than the programmed value at the moment of a rewrite operation.	Δ
R4	Vertical Total	When a rewrite operation is performed during the last raster period in the line, there is a possibility that the disturbance occurs during the vertical scan period. There is no problem of a rewrite operation during raster period except this period.	Δ
R5	Vertical Total Adjust	When a rewrite operation is performed in the last character time of the raster period, there are some cases where the numbers of Adjust Raster, specified by program, are not added. (Only during the adjust raster period)	Δ
R6	Vertical Displayed	After the moment of a rewrite operation, there are some cases where the Display is inhibited. However, the display according to the programmed value is performed from the next field.	O
R7	Vertical Sync Position	There are some cases where VSYNC is placed on the position different from the programmed value or the noise is output.	X
R8	Interlace & Skew	Neither scan mode bit nor skew bit is rewritten dynamically. Dynamic Rewrite into scan mode bit and skew bit is prohibited.	X
R9	Maximum Raster Address	The internal operation will be disordered by a rewrite operation.	X
R10	Cursor Start Raster	When a rewrite operation is performed in the last character time of the raster period, there are some cases where the jitter occurs on the cursor raster or the cursor is not displayed correctly. There is also a possibility that the blink rate becomes temporally shorter than usual.	Δ
R11	Cursor End Raster	When a rewrite operation is performed in the last character time of the raster period, there are some cases where the jitter occurs on the cursor raster or the cursor is not displayed correctly. Moreover, there are also some cases where the blink rate becomes temporally shorter than normal operation.	Δ
R12	Start Address (H)	R12 and R13 are used in the last raster period of the field. A rewrite operation can be performed except during this period. However, when R12 and R13 are rewritten in each field separately, the display operation, whose start address is determined temporally by programming sequence, will be performed. A rewrite operation should be performed during the horizontal/vertical display period.	O
R13	Start Address (L)		O
R14	Cursor (H)	When a rewrite operation is performed during the display period, there are some cases where the cursor is temporally displayed at the address different from the programmed value. A rewrite operation should be performed during the horizontal/ vertical retrace period. Also, when R14 and R15 are rewritten in each field separately, the cursor is displayed temporally at the temporal address determined by programming sequence.	O
R15	Cursor (L)		O

* ····· means temporary abnormal operations in rewriting the internal register during the display operation. Normally, after a rewrite operation, the LSI performs the specified display operation from the next field.
(The operations in this table are outside our guarantee and are regarded as materials for reference.)

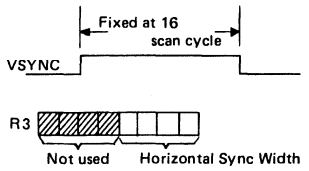
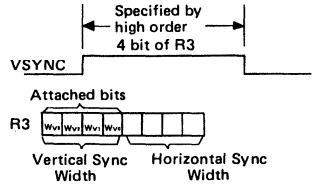
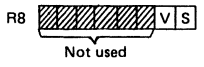
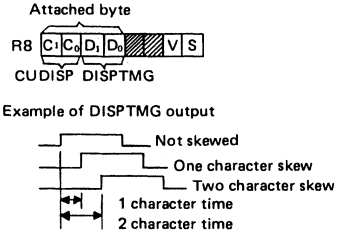
** O ····· A rewrite operation is possible without affecting the screen in the display so much.
** Δ ····· If conditions are satisfied, a rewrite operation is possible. If conditions are not satisfied, there are some cases where a flicker and so on occur temporally.
** X ····· When a rewrite operation is performed, there are some cases where a flicker and so on occur temporally.

■ COMPARISON BETWEEN HD6845S AND HD6845

● Comparison of function between HD6845S and HD6845

No.	Functional Difference		HD6845	HD6845S
1	Interlace Sync & Video Mode Display	Programming Method of Number of Vertical Characters	Character line address  In HD6845, number of characters is vertically programmed in unit of two lines, as illustrated above. (Number of vertical total characters, Number of vertical displayed characters, Vertical Sync Position) Example of above figure Programmed number into Vertical Displayed Register = 5	Character line address  In HD6845S, number of characters is vertically programmed in unit of one line, as illustrated above. (Number of vertical total characters, Number of vertical displayed characters, Vertical Sync Position) Example of above figure Programmed number into Vertical Displayed Register = 10
		Number of Rasters Per Character Line	Only even number can be specified.  Number of raster = 10 scan line (specified) However, number which is programmed into register is calculated at follows. Programmed number (Nr) = (Number specified) - 1	Both even number and odd number can be specified. Character line address Character line address  When number of raster per character line is EVEN. Number of raster = 10 scan line (specified) When number of raster per character line is ODD. Number of raster = 9 scan line (specified) However, number which is programmed into register is calculated as follows. Programmed number (Nr) = (Number specified) -2
		ırsor Display	Cursor is displayed in either EVEN field or ODD field.  	Cursor is displayed in both EVEN field and ODD field. 

(to be continued)

No.	Functional Difference	HD6845	HD6845S
2	Vertical Sync Pulse Width (VSYNC output)	Fixed at 16 raster scan cycle (16H)  	Programmable (1 ~ 16 raster scan cycle) 
3	SKEW Function	Not included  	SKEW function is newly included in DISPTMG, CUDISP signals. 
4	Start Address Register	Impossible to READ	Possible to READ
5	RESET Signal ($\overline{RES}$)	$MA_0 \sim MA_{13}$ Output, $RA^0 \sim RA_4$ Output ..... Synchronous reset Other Outputs ..... Asynchronous reset Output signals of $MA_0 \sim MA_{13}$, $RA_0 \sim RA_4$, synchronizing with DLK "Low" level, go to "Low" level, after $\overline{RES}$ has gone to "Low" Other outputs go to "Low" immediately after $\overline{RES}$ has gone to "Low" level	$MA_0 \sim MA_{13}$ Output, $RA_0 \sim RA_4$ Output, Other Outputs ..... Asynchronous reset Output signals of $MA_0 \sim MA_{13}$, $RA_0 \sim RA_4$ and others go to "Low" level immediately after $\overline{RES}$ has gone to "Low" level.

- **Comparison of Timing Signal between HD6845S and HD6845**

No.	Item	Symbol	HD6845			HD6845S			Unit
			min	typ	max	min	typ	max	
1	Clock Cycle Time	t_{cycC}	330	–	–	270	–	–	ns
2	Clock "High" Pulse Width	PW_{CH}	150	–	–	130	–	–	ns
3	Clock "Low" Pulse Width	PW_{CL}	150	–	–	130	–	–	ns
4	Rise and Fall Time for Clock Input	t_{Cr}, t_{Cf}	–	–	15	–	–	20	ns
5	Horizontal Sync Delay Time	t_{HSD}	–	–	250	–	–	200	ns
6	Light Pen Strobe Pulse Width	PW_{LPH}	80	–	–	60	–	–	ns
7	Light Pen Strobe Uncertain Time of Acceptance	t_{LPD1}	–	–	80	–	–	70	ns
		t_{LPD2}	–	–	10	–	–	0	ns

■ COMPATIBILITY OF HD6845S AND HD6845

Non-interlace mode control
Interlace sync mode control } ; Fully compatible with HD6845* HD6845 can be directly replaced by HD6845S in these modes.

Interlace sync & Video mode control :
Not compatible with HD6845 in regard to programming and data for vertical direction need to be changed.

* The functions added to HD6845S utilize undefined bits of the Control Register in HD6845. If "0" is programmed to the undefined bits in the initial set, it is possible to replace HD6845 with HD6845S without changing the parameters.

Note) The restriction on programming of HD6845S and HD6845 should be taken into consideration.

HD6350/HD6850

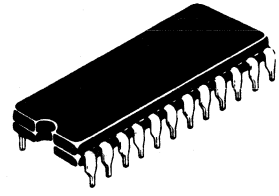
ACIA (Asynchronous Communications Interface Adapter)

The HD6350/HD6850 Asynchronous Communications Interface Adapter provides the data formatting and control to interface serial asynchronous data communications information to bus organized systems such as the HMCS6800 Micro-processing Unit.

The bus interface of the HD6350/HD6850 includes select, enable, read/write, interrupt and bus interface logic to allow data transfer over an 8-bit bi-directional data bus. The parallel data of the bus system is serially transmitted and received by the asynchronous data interface with proper formatting and error checking.

The functional configuration of the ACIA is programmed via the data bus during system initialization. A programmable Control Register provides variable word lengths, clock division ratios, transmit control, receive control, and interrupt control. For peripheral or modern operation three control lines are provided.

HD6350P, HD6850P



(DP-24)

FEATURES

- Serial/Parallel Conversion of Data
- Eight and Nine-bit Transmission
- Optional Even and Odd Parity
- Parity, Overrun and Framing Error Checking
- Peripheral/Modem Control Functions (Clear to Send $\overline{CTS}$, Request to Send $\overline{RTS}$, Data Carrier Detect $\overline{DCD}$)
- Optional ÷ 1, ÷ 16, and ÷ 64 Clock Modes
- One-or Two-Stop Bit Operation
- Double Buffered

– HD6350 –

- Low-Power, High-Speed, High-Density CMOS
- Compatible with NMOS ACIA (HD6850)
- Wide Range Operating Voltage (V_{CC} = 5V ±10%)
- Up to 1Mbps Transmission

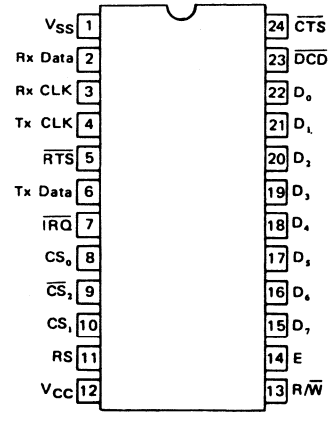
– HD6850 –

- Compatible with MC6850 and MC68A50
- Up to 500Kbps Transmission

TYPE OF PRODUCTS

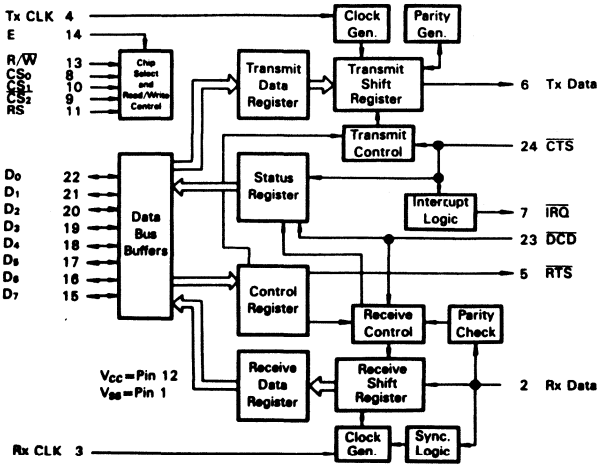
Type	Process	Clock Frequency	Package
HD6350	CMOS	1.0MHz	DP-24
HD63A50		1.5MHz	
HD63B50		2.0MHz	
HD6850	NMOS	1.0MHz	DP-24
HD68A50		1.5MHz	

■ PIN ARRANGEMENT



(Top View)

■ BLOCK DIAGRAM



■ ABSOLUTE MAXIMUM PATINGS

Item	Symbol	Value		Unit
		HD6350	HD6850	
Supply Voltage	V_{cc}*	−0.3 ~+7,0	−0.3 ~+7.0	V
Input Voltage	V_{in}*	−0.3 ~+7.0	−0.3 ~+7.0	V
Maximum Output Current**	$\|I_o\|$	10		mA
Operating Temperature	T_{opr}	−20 ~+75	−20 ~+75	°C
Storage Temperature	T_{stg}	−55 ~+150	−55 ~+150	°C

* With respect to V_{ss} (SYSTEM GND)

** Maximum output current is the maximum current which can flow out from one output terminal or I/O common terminal (D_0 ~D_7, RTS, Tx Data, IRQ).

(NOTE) Permanent LSI damage may occur if maximum ratings are exceeded. Normal operation should be under recommended operating conditions. If these conditions are exceeded, it could affect reliability of LSI.

■ RECOMMENDED OPERATING CONDITIONS

Item		Symbol	HD6350			HD6850			Unit
			min	typ	max	min	typ	max	
Supply Voltage		V_{cc}*	4.5	5.0	5.5	4.75	5.0	5.25	V
Input "Low" Voltage		V_{IL}*	0	–	0.8	−0.3	–	0.8	V
Input "High" Voltage	D_0~D_7, RS, Tx CLK, $\overline{DCD}$, $\overline{CTS}$ Rx Data	V_{IH}*	2.0	–	V_{cc}	2.0	–	V_{cc}	V
	CS_0, $\overline{CS_2}$, CS_1, R/$\overline{W}$, E, Rx CLK		2.2	–	V_{cc}				
Operating Temperature		T_{opr}	−20	25	75	−20	25	75	°C

* With respect to V_{ss} (SYSTEM GND)

■ ELECTRICAL CHARACTERISTICS

• DC CHARACTERISTICS (HD6350; V_{CC} = 5V ±10%, HD6850; V_{CC} = 5V ±5%, V_{SS} = 0V, Ta = −20 ~ +75°C, unless otherwise noted.)

Item		Symbol	HD6350				HD6850				Unit
			Test Condition	min	typ*	max	Test Condition	min	typ*	max	
Input "High" Voltage	D_0 ~ D_7, RS, Tx CLK, $\overline{DCD}$, $\overline{CTS}$, Rx Data	V_{IH}		2.0	–	V_{CC}		2.0	–	V_{CC}	V
	CS_0, $\overline{CS_2}$, CS_1, R/$\overline{W}$, E Rx CLK			2.2	–	V_{CC}					
Input "Low" Voltate	All Inputs	V_{IL}		−0.3	–	0.8		−0.3	–	0.8	V
Input Leakage Current	R/$\overline{W}$, CS_0, CS_1, CS_2, E	I_{in}	V_{in} = 0 ~ V_{CC}	−2.5	–	2.5	V_{in} = 0 ~ 5.25V	−2.5	–	2.5	μA
Three-State (Off State) Input Current	D_0 ~ D_7	I_{TSI}	V_{in} = 0.4 ~ V_{CC}	−10	–	10	V_{in} = 0.4 ~ 2.4V	−10	–	10	μA
Output "High" Voltage	D_0 ~ D_7	V_{OH}	I_{OH} = −400μA	4.1	–	–	I_{OH} = −205μA, Enable Pulse Width ≦ 25μs	2.4	–	–	V
			I_{OH} ≦ −10μA	V_{CC}−0.1	–	–					
	Tx Data, $\overline{RTS}$		I_{OH} = −400μA	4.1	–	–	I_{OH} = −100μA, Enable Pulse Width ≦ 25μs	2.4	–	–	V
			I_{OH} ≦ −10μA	V_{CC}−0.1	–	–					
Output "Low" Voltage	All Outputs	V_{OL}	I_{OH} = 1.6mA	–	–	0.4	I_{OL} = 1.6mA, Enable Pulse Width ≦ 25μs	–	–	0.4	V
Output Leakage Current (Off State)	$\overline{IRQ}$	I_{LOH}	V_{OH} = V_{CC}	–	–	10	V_{OH} = 2.4V	–	–	10	μA
Input Capacitance	D_0 ~ D_7	C_{in}	V_{in} = 0V, T_a = 25°C, f = 1.0 MHz	–	–	12.5	V_{in} = 0V, T_a = 25°C, f = 1.0 MHz	–	–	12.5	pF
	E, Tx CLK, Rx CLK, R/W, RS, RX Data, CS_0, CS_1, $\overline{CS_2}$, $\overline{CTS}$, $\overline{DCD}$			–	–	7.5		–	–	7.5	
Output Capacitance	$\overline{RTS}$, Tx Data	C_{out}	V_{in} = 0V, T_a = 25°C, f = 1.0 MHz	–	–	10	V_{in} = 0V, T_a = 25°C f = 1.0 MHz	–	–	10	pF
	$\overline{IRQ}$			–	–	5.0		–	–	5.0	
Supply Current	• Under transmitting and Receiving operation • 500 kbps • Data bus in R/W operation		E = 1.0 MHz	–	–	3					mA
			E = 1.5 MHz	–	–	4					
			E = 2.0 MHz	–	–	5					
	• Chip is not selected. • 500 kbps • Under non transmitting and receiving operation • Input level (Except E) V_{IH} min = V_{CC}−0.8V V_{IL} max = 0.8V.		E = 1.0 MHz	–	–	200					
			E = 1.5 MHz	–	–	250					
			E = 2.0 MHz	–	–	300					
Power Dissipation		P_D						–	300	525	mW

* T_a = 25°C, V_{CC} = 5.0V

- **AC CHARACTERISTICS (HD6350; V_{CC}= 5.0V ±10%, HD6850; V_{CC} = 5V ±5%, V_{SS} = 0V, T_a = −20~+75°C, unless otherwise noted.)**

1. TIMING OF DATA TRANSMISSION

Item		Symbol	Test Condition	HD6350		HD63A50		HD63B50		HD6850 /HD68A50		Unit
				min	max	min	max	min	max	min	max	
Minimum Clock Pulse Width	÷1 Mode	PW_{CL}	Fig. 1	900	–	650	–	500	–	900	–	ns
	÷16, ÷64 Modes			600	–	450	–	280	–	600	–	ns
	÷1 Mode	PW_{CH}	Fig. 2	900	–	650	–	500	–	900	–	ns
	÷16, ÷64 Modes			600	–	450	–	280	–	600	–	ns
Clock Frequency	÷1 Mode	f_C		–	500	–	750	–	1000	–	500	kHz
	÷16, ÷64 Modes			–	800	–	1000	–	1500	–	800	
Clock-to-Data Delay for Transmitter		t_{TOD}	Fig. 3	–	600	–	540	–	460	–	1000	ns
Receive Data Setup Time	÷1 Mode	t_{RDSU}	Fig. 4	250	–	100	–	30	–	500	–	ns
Receive Data Hold Time	÷1 Mode	t_{RDH}	Fig. 5	250	–	100	–	30	–	500	–	ns
$\overline{IRQ}$ Release Time		t_{IR}	Fig. 6	–	1200	–	900	–	700	–	1200	ns
$\overline{RTS}$ Delay Time		t_{RTS}	Fig. 6	–	560	–	480	–	400	–	1000	ns
Rise Time and Fall Time (or 10% of the pulse width if smaller)		t_r, t_f		–	1000	–	500	–	250	–	1000	ns

*

2. BUS TIMING CHARACTERISTICS

1) READ

Item	Symbol	Test Condition	HD6350		HD63A50		HD63B50		HD6850		HD68A50		Unit
			min	max	min	max	min	max	min	max	min	max	
Enable Cycle Time	t_{cycE}	Fig. 7	1000	–	666	–	500	–	1000	–	666	–	ns
Enable "High" Pulse Width	PW_{EH}	Fig. 7	450	–	280	–	220	–	450	25000	280	25000	ns
Enable "Low" Pulse Width	PW_{EL}	Fig. 7	430	–	280	–	210	–	430	–	280	–	ns
Setup Time, Address and R/$\overline{W}$ Valid to Enable Positive Transition	t_{AS}	Fig. 7	80	–	60	–	40	–	140	–	140	–	ns
Data Delay Time	t_{DDR}	Fig. 7	–	290	–	180	–	150	–	320	–	220	ns
Data Hold Time	t_H	Fig. 7	20	100	20	100	20	100	10	–	10	–	ns
Address Hold Time	t_{AH}	Fig. 7	10	–	10	–	10	–	10	–	10	–	ns
Rise and Fall Time for Enable Input	t_{Er}, t_{Ef}	Fig. 7	–	25	–	25	–	20	–	25	–	25	ns

2) WRITE

Item	Symbol	Test Condition	HD6350		HD63A50		HD63B50		HD6850		HD68A50		Unit
			min	max	min	max	min	max	min	max	min	max	
Enable Cycle Time	t_{cycE}	Fig. 8	1000	–	666	–	500	–	1000	–	666	–	ns
Enable "High" Pulse Width	PW_{EH}	Fig. 8	450	–	280	–	220	–	450	25000	280	25000	ns
Enable "Low" Pulse Width	PW_{EL}	Fig. 8	430	–	280	–	210	–	430	–	280	–	ns
Setup Time, Address and R/$\overline{W}$ Valid to Enable Positive Transition	t_{AS}	Fig. 8	80	–	60	–	40	–	140	–	140	–	ns
Data Setup Time	t_{DSW}	Fig. 8	165	–	80	–	60	–	195	–	80	–	ns
Data Hold Time	t_H	Fig. 8	10	–	10	–	10	–	10	–	10	–	ns
Address Hold Time	t_{AH}	Fig. 8	10	–	10	–	10	–	10	–	10	–	ns
Rise and Fall Time for Enable Input	t_{Er}, t_{Ef}	Fig. 8	–	25	–	25	–	20	–	25	–	25	ns

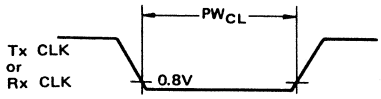


Figure 1 Clock Pulse Width, "Low" State

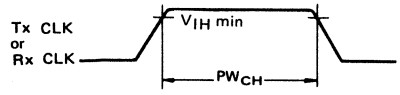


Figure 2 Clock Pulse Width, "High" State

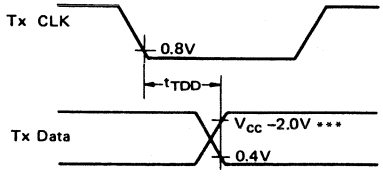


Figure 3 Transmit Data Output Delay

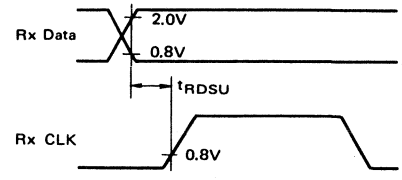


Figure 4 Receive Data Setup Time (÷1 Mode)

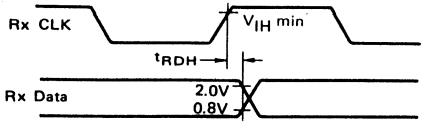
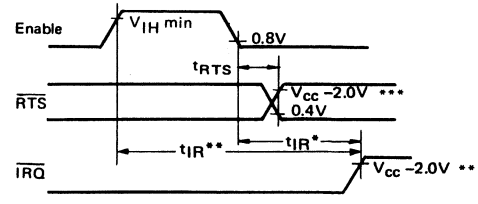


Figure 5 Receive Data Hold Time (÷1 Mode)



* (1) $\overline{IRQ}$ Release Time applied to Rx Data Register read operation.
(2) $\overline{IRQ}$ Release Time applied to Tx Data Register write operation.
(3) $\overline{IRQ}$ Release Time applied to control Register write TIE = 0, RIE = 0 operation.

** $\overline{IRQ}$ Release Time applied to Rx Data Register read operation right after read status register, when $\overline{IRQ}$ is asserted by $\overline{DCD}$ rising edge.

*** 2.4V for HD6850.

(Note) Note that followings take place when $\overline{IRQ}$ is asserted by the detection of transmit data register empty status. $\overline{IRQ}$ is released to "High" asynchronously with E signal when $\overline{CTS}$ goes "High". (Refer to Figure 14)

Figure 6 $\overline{RTS}$ Delay and $\overline{IRQ}$ Release Time

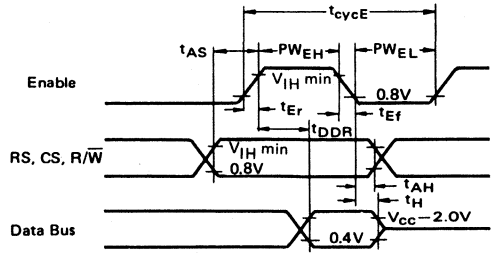


Figure 7 Bus Read Timing Characteristics
(Read information from ACIA)

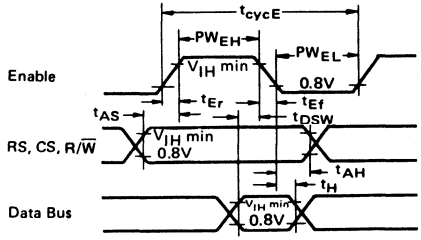
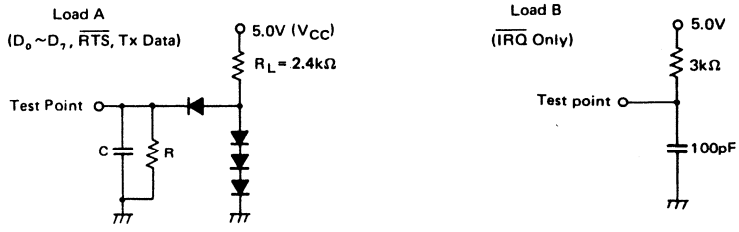


Figure 8 Bus Write Timing Characteristics
(Write information into ACIA)



C = 130pF for D_0~D_7
= 30pF for $\overline{RTS}$ and Tx Data
R*= 10kΩ for D_0 ~ D_7, $\overline{RTS}$ and Tx Data
All diodes are 1S2074Ⓗ or Equivalent.

* HD6840
R = 11kΩ for D_0 ~ D_7
= 24kΩ for RTS and Tx Data.

Figure 9 Bus Timing Test Loads

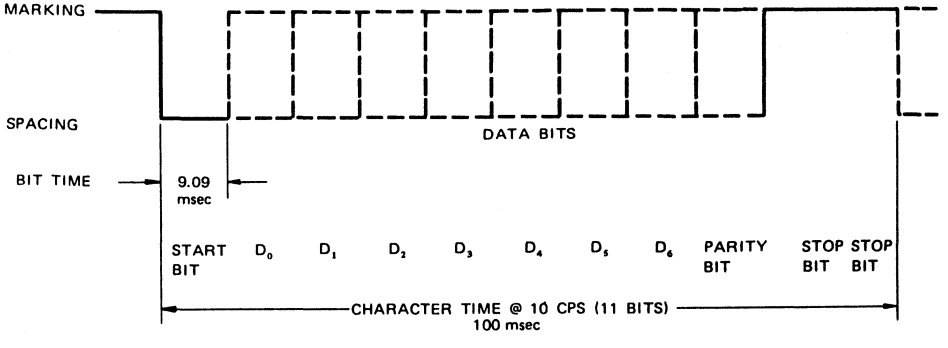
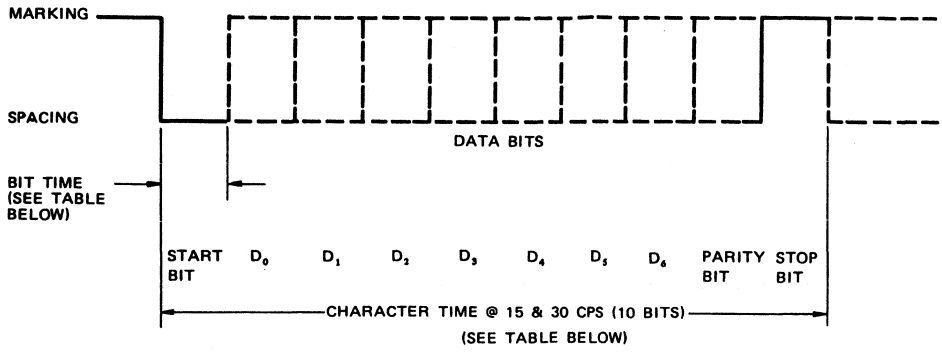


Figure 10 110 Baud Serial ASCII Data Timing



BAUD RATE	150	300
CHARACTERS/SEC	15	30
BIT TIME (msec)	6.67	3.33
CHARACTER TIME (msec)	66.7	33.3

$$\text{BIT TIME} = \frac{\text{SEC}}{\text{BAUD RATE}}$$

Figure 11 150 & 300 Baud Serial ASCII Data Timing

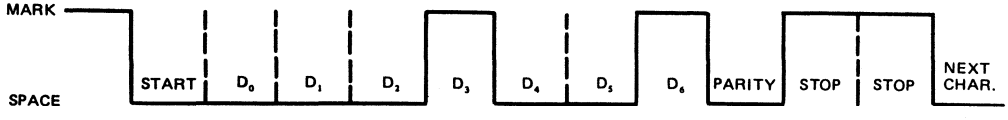


Figure 12 Send a 7 Bit ASCII Char. "H" Even Parity
– 2 Stop Bits H = 48_{16} = 1001000_2

■ DATA OF ACIA

ACIA is an interface adappter which controls transmission and reception of Asynchronous serial data. Some examples of serial data are shown in Figs. 10 ~ 12.

■ INTERNAL STRUCTURE OF ACIA

ACIA provides the following; 8-bit Bi-directional Data Buses (D_0 ~ D_7), Receive Data Input (Rx Data), Transmit Data Output (Tx Data), three Chip Selects (CS_0, CS_1, $\overline{CS_2}$), Register Select Input (RS), Two Control Input (Read/Write: R/$\overline{W}$, Enable: E), Interrupt Request Output($\overline{IRQ}$), Clear-to-Send ($\overline{CTS}$) to control the modem, Request-to-Send ($\overline{RTS}$), Data Carrier Detect($\overline{DCD}$) and Clock Inputs(Tx CLK, Rx CLK) used for synchronization of received and transmitted data. This ACIA also provides four registers; Status Register, Control Register, Receive Register and Transmit Register.

24-pin dual-in-line type package is used for the ACIA. Internal Structure of ACIA is illustrated in Fig. 13.

■ ACIA OPERATION

● Master Reset

ACIA has an internal master reset function controlled by software, since it has no hardware reset pin. Bit 0 and bit 1 of control register should be set to "11" to execute master reset, also bit 5 and bit 6 should be programmed to get predetermined $\overline{RTS}$ output accordingly. To release the master reset, the data other than "11" should be written into bit 0, bit 1 of the control register. When the master reset is released, the control register needs to be programmed to get predetermined options such as clock divider ratios, word length, one or two stop bits, parity(even, old, or none), etc.

It may happen that "Low" level output is provided in $\overline{IRQ}$ pin during the time after power-on till master reset. In the system using ACIA, interrupt mask bit of MPU should be released after the master reset of ACIA. (MPU interrupt should be prohibited until MPU program completes the master reset of ACIA.) Transmit Data Register (TDR) and Receive Data Register (RDR) can not be reset by master reset.

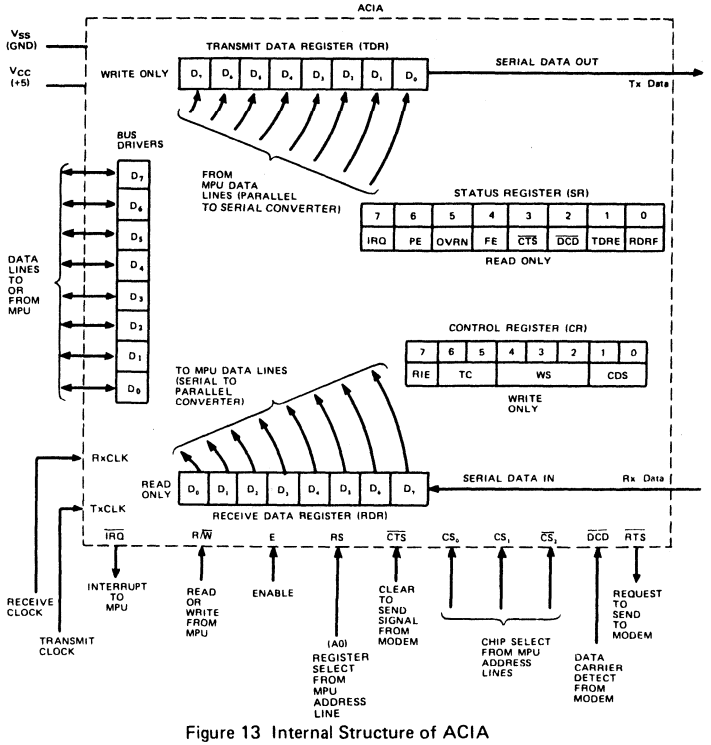


Figure 13 Internal Structure of ACIA

- **Transmit**

A typical transmitting sequence consists of reading the ACIA Status Register either as a result of an interrupt or in the ACIA's turn in a polling sequence. A character may be written into the Transmit Data Register if the status read operation has indicated that the Transmit Data Register is empty. This character is transferred to a Shift Register where it is serialized and transmitted from the Transmit Data output preceded by a start bit and followed by one or two stop bits. Internal parity (odd or even) can be optionally added to the character and will occur between the last data bit and the first stop bit. After the first character is written in the Data Register, the Status Register can be read again to check for a Transmit Data Register Empty condition and current peripheral status. If the register is empty, another character can be loaded for transmission even though the first character is in the process of being transmitted (because of double buffering). The second character will be automatically transferred into the Shift Register when the first character transmission is completed. This sequence continues until all the characters have been transmitted.

- **Receive**

Data is received from a peripheral by means of the Receive Data input. A divide-by-one clock ratio is provided for an externally synchronized clock (to its data) while the divide-by-16 and 64 ratios are provided for internal synchronization. Bit synchronization in the divide-by-16 and 64 modes is initiated by the detection of the leading mark-space transition of the start bit. False start bit delection capability insures that a full half bit of a start bit has been received before the internal clock is synchronized to the bit time. As a character is being received, parity (odd or even) will be checked and the error indication will be available in the Status Register along with framing error, overrun error, and Receive Data Register full. In a typical receiving sequence, the Status Register is read to determine if a character has been received from a peripheral. If the Receiver Data Register is full, the character is placed on the 8-bit ACIA bus when a Read Data command is received from the MPU. When parity has been selected for an 8-bit word (7 bits plus parity), the receiver strip the parity bit (D_7="0") so that data alone is transferred to the MPU. This feature reduces MPU programming. The Status Register can continue to be read again to determine when another character is available in the Receive Data Register. The receiver is also double buffered so that a character can be read from the data register as another character is being received in the Shift register. The above sequence continues until all characters have been received.

■ ACIA INTERNAL REGISTERS

The ACIA provides four registers; Transmit Data Register (TDR), Receive Data Register(RDR), Control Register(CR) and Status Register(SR). The content of each of the registers is summarized in Table 1.

Table 1 Definition of ACIA Register Contents

Buffer Address	RS=1 · R/$\overline{W}$=0 ****	RS=1 · R/$\overline{W}$=1	RS=0 · R/$\overline{W}$=0	RS=0 · R/$\overline{W}$=1
Data Bus	Transmit Data Register	Receiver Data Register	Control Register	Status Register
	(Write Only)	(Read Only)	(Write Only)	(Read Only)
0	Data Bit 0*	Data Bit 0	Counter Divide Select (CR0)	Rx Data Reg. Full (RDRF)
1	Data Bit 1	Data Bit 1	Counter Divide Select (CR1)	Tx Data Reg. Empty (TDRE)
2	Data Bit 2	Data Bit 2	Word Select 1 (CR2)	Data Carrier Detect ($\overline{DCD}$)
3	Data Bit 3	Data Bit 3	Word Select 2 (CR3)	Clear to Send ($\overline{CTS}$)
4	Data Bit 4	Data Bit 4	Word Select 3 (CR4)	Framing Error (FE)
5	Data Bit 5	Data Bit 5	Tx Control 1 (CR5)	Overrun (OVRN)
6	Data Bit 6	Data Bit 6	Tx Control 2 (CR6)	Parity Error (PE)
7	Data Bit 7***	Data Bit 7**	Rx Interrupt Enable (CR7)	Interrupt Request (IRQ)

* Leading bit = LSB = Bit 0
** Data bit will be zero in 7-bit plus parity modes.
*** Data bit is "don't care" in 7-bit plus parity modes.
**** 1 ··· "High" level, 0 ... "Low" level

- **Transmit Data Register (TDR)**

Data is written in the Transmit Data Register during the negative transition of the enable (E) when the ACIA has been addressed and RS · R/$\overline{W}$ is selected. Writing data into the register causes the Transmit Data Register Empty bit in the Status Register to go "0". Data can then be transmitted. If the transmitter is idling and no character is being transmitted, then the transfer will take place within 2 bit time + several E cycles of the trailing edge of the Write command. If a character is being transmitted, the new data character will commence as soon as the previous character is complete. The transfer of data causes the Transmit Data Register Empty (TDRE) bit to indicate empty.

- **Receive Data Register (RDR)**

Data is automatically transferred to the empty Receive Data Register (RDR) from the receiver deserializer (a shift register) upon receiving a complete character. This event causes the Receive Data Register Full bit (RDRF) on the status buffer to go "1" (full). Data may then be read through the bus by addressing the ACIA and R/$\overline{W}$ "High" when the ACIA is enabled. The non-destructive read cycle causes the RDRF bit to be cleared to empty although the data is retained in the RDR. The status is maintained by RDRF as to whether or not the data is current. When the Receive Data Register is full, the automatic transfer of data from the Receiver Shift Register to the Data Register is inhibited and the RDR contents remain valid with its current status stored in the Status Register.

- **Control Register**

The ACIA Control Register consists of eight bits of write-only buffer that are selected when RS and R/$\overline{W}$ are "Low". This register controls the function of the receiver, transmitter, interrupt enables, and the Request-to-Send ($\overline{RTS}$) peripheral/modem control output.

Counter Divide Select Bits (CR0 and CR1)

The Counter Divide Select Bits (CR0 and CR1) determine the divide ratios utilized in both the transmitter and receiver section of the ACIA. Additionally, these bits are used to provide a master reset for the ACIA which clears the Status Register (except for external conditions on $\overline{CTS}$ and $\overline{DCD}$) and initializes both the receiver and transmitter. Master reset does not affect other Control Register bits. Note that after power-on or a power fail/restart, these bits must be set "1" to reset the ACIA. After resetting, the clock divide ratio may be selected. These counter select bits provide for the following clock divide ratios:

Table 2 Function of Counter Devide Select Bit

CR1	CR0	Function
0	0	÷1
0	1	÷16
1	0	÷64
1	1	Master Reset

Word Select Bits (CR2, CR3, and CR4)

The Word Select bits are used to select word length, parity, and the number of stop bits. The encoding format is as follows:

Table 3 Function of Word Select Bit

CR4	CR3	CR2	Function
0	0	0	7 Bits + Even Parity + 2 Stop Bits
0	0	1	7 Bits + Odd Parity + 2 Stop Bits
0	1	0	7 Bits + Even Parity + 1 Stop Bit
0	1	1	7 Bits + Odd Parity + 1 Stop Bit
1	0	0	8 Bits + 2 Stop Bits
1	0	1	8 Bits + 1 Stop Bit
1	1	0	8 Bits + Even Parity + 1 Stop Bit
1	1	1	8 Bits + Odd Parity + 1 Stop Bit

Word length, Parity Select, and Stop Bit changes are not buffered and therefore become effective immediately.

Transmitter Control Bits (CR5 and CR6)

Two Transmitter Control bits provide for the control of the interrupt from the Transmit Data Register Empty condition, the Request-to-Send ($\overline{RTS}$) output, and the transmission of a Break level (space). The following encoding format is used:

Table 4 Function of Transmitter Control-Bit

CR6	CR5	Function
0	0	$\overline{RTS}$ = "Low", Transmitting Interrupt Disabled.
0	1	$\overline{RTS}$ = "Low", Transmitting Interrupt Enabled.
1	0	$\overline{RTS}$ = "High", Transmitting Interrupt Disabled.
1	1	$\overline{RTS}$ = "Low", Transmits a Break level on the Transmit Data Output. Transmitting Interrupt Disabled.

Receive Interrupt Enable Bit (CR7)

The following interrupts will be enabled by a "1" in bit position 7 of the Control Register (CR7): Receive Data Register Full, Overrun, or a "Low" to "High" transistion on the Data Carrier Detect ($\overline{DCD}$) signal line.

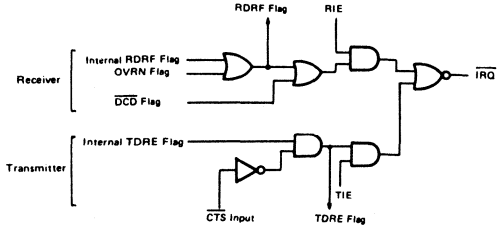


Figure 14 $\overline{IRQ}$ Internal Circuit

- **Status Register**

Information on the status of the ACIA is available to the MPU by reading the ACIA Status Register. This read-only register is selected when RS is "Low" and R/$\overline{W}$ is "High". Information stored in this register indicates the status of the Transmit Data Register, the Receive Data Register and error logic, and the peripheral/modem status inputs of the ACIA.

Receive Data Register Full (RDRF), Bit 0

RDRF indicates that received data has been transferred to the Receive Data Register. RDRF is cleared after an MPU read of the Receive Data Register or by a master reset. The cleared or empty state indicates that the contents of the Receive Data Register are not current. Data Carrier Detect ($\overline{DCD}$) being "High" also causes RDRF to indicate empty.

Transmit Data Register Empty (TDRE), Bit 1

The Transmit Data Register Empty bit being set "1" indicates that the Transmit Data Register contents have been transferred and that new data may be entered. The "0" state indicates that the register is full and that transmission of a new character has not begun since the last write data command.

Data Carrier Detect ($\overline{DCD}$), Bit 2

The $\overline{DCD}$ bit will be "1" when the $\overline{DCD}$ input from a modem has gone "High" to indicate that a carrier is not present. This bit going "1" causes an Interrupt Request to be generated when the Receive Interrupt Enable is set. It remains "1" after the $\overline{DCD}$ input is returned "Low" until cleared by first reading the Status Register and then the Data Register or until a master reset occurs. If the $\overline{DCD}$ input remains "High" after read status and read data or master reset has occurred, the interrupt is cleared, the $\overline{DCD}$ status bit remains "1" and will follow the $\overline{DCD}$ input.

Clear-to-Send ($\overline{CTS}$), Bit 3

The $\overline{CTS}$ bit indicates the state of the $\overline{CTS}$ input from a modem. A "Low" $\overline{CTS}$ input indicates that there is a $\overline{CTS}$ from the modem. In the "High" state, the Transmit Data Register Empty bit is inhibited and the $\overline{CTS}$ status bit will be "1". Master reset does not affect the Clear-to-Send Status bit.

Framing Error (FE), Bit 4

FE flag indicates that the received character is improperly framed by a start and a stop bit and is detected by the absence of the 1st stop bit. This error indicates a synchronization error, faulty transmission, or a break condition. The FE flag is set or reset during the receive data transfer time. Therefore, this error indicator is present throughout the time that the associated character is available.

Receiver Overrun (OVRN), Bit 5

Overrun is an error flag that indicates that one or more characters in the data stream were lost. That is, a character or a number of characters were received but not read from the Receive Data Register (RDR) prior to subsequent characters being received. The overrun condition begins at the midpoint of the last bit of the second character received in succession without a read of the RDR having occurred. The overrun does not occur in the Status Register until the valid character prior to Overrun has been read. The RDRF bit remains set until the Overrun is reset. Character synchronization is maintained during the Overrun condition. The Overrun indication is reset after the reading of data from the Receive Data Register or by a Master Reset.

Parity Error (PE), Bit 6

The PE flag indicates that the number of "1"s (highs) in the character does not agree with the preselected odd or even parity. Odd parity is defined to be when the total number of ones is odd. The parity error indication will be present as long as the data character is in the RDR. If no parity is selected, then both the transmitter parity generator output and the receiver parity check results are inhibited.

Interrupt Request (IRQ), Bit 7

The IRQ bit indicates the state of the $\overline{IRQ}$ output. Any interrupt condition with its applicable enable will be indicated in this status bit. Anytime the $\overline{IRQ}$ output is "Low" the IRQ bit will be "1" to indicate the interrupt or service request status. IRQ is cleared by a read operation to the Receive Data Register or a write operation to the Transmit Data Register. (Refer to Figure 14.)

■ SIGNAL FUNCTIONS

• Interface Signal for MPU

Bi-Directional Data Bus (D_0~D_7)

The bi-directional data bus (D_0~D_7) allow for data transfer between the ACIA and the MPU. The data bus output drivers are three-state devices that remain in the high impedance (off) state except when the MPU performs an ACIA read operation.

Enable (E)

The Enable signal, E, is a high impedance TTL compatible input that enables the bus input/output data buffers and clocks data to and from the ACIA. This signal will normally be a derivative of the HMCS6800 ϕ_2 Clock. The ACIA accepts both continuous pulse signal and strobe type signal as Enable input.

Read/Write (R/$\overline{W}$)

The R/$\overline{W}$ line is a high impedance input that is TTL compatible and is used to control the direction of data flow through the ACIA's input/output data bus interface. When R/$\overline{W}$ is "High" (MPU Read cycle), ACIA output drivers are turned on and a selected register is read. When it is "Low", the ACIA output drivers are turned off and the MPU writes into a selected register. Therefore, the R/$\overline{W}$ signal is used to select read-only or write-only registers within the ACIA.

Chip Select (CS_0, CS_1, $\overline{CS_2}$)

These three high impedance TTL compatible input lines are used to address the ACIA. The ACIA is selected when CS_0 and CS_1 are "High" and $\overline{CS_2}$ is "Low". Transfers of data to and from the ACIA are then performed under the control of the Enable signal, Read/Write, and Register Select.

Register Select (RS)

The RS line is a high impedance input that is TTL compatible. A "High" level is used to select the Transmit/Receive Data Registers and a "Low" level the Control/Status Registers. The R/$\overline{W}$ signal line is used in conjunction with Register Select to select the read-only or write-only register in each register pair.

Interrupt Request ($\overline{IRQ}$)

$\overline{IRQ}$ is a TTL compatible, open-drain (no internal pullup), active "Low" output that is used to interrupt the MPU. The $\overline{IRQ}$ output remains "Low" as long as the cause of the interrupt is present and the appropriate interrupt enable within the ACIA is set.

Clock Inputs

Separate high impedance TTL compatible inputs are provided for clocking of transmitted and received data. Clock frequencies of 1, 16 or 64 times the data rate may be selected.

Transmit Clock (Tx CLK)

The Tx CLK input is used for the clocking of transmitted data. The transmitter initiates data on the negative transition of the clock.

Receive Clock (Rx CLK)

The Rx CLK input is used for synchronization of received data. (In the ÷ 1 mode, the clock and data must be synchronized externally.) The receiver samples the data on the positive transition of the clock.

• Serial Input/Output Lines

Receive Data (Rx Data)

The Rx Data line is a high impedance TTL compatible input through which data is received in a serial format. Synchronization with a clock for detection of data is accomplished internally when clock rates of 16 or 64 times the bit rate are used. Data rates are in the range of 0 to 500 kbps when external synchronization is utilized.

Transmit Data (Tx Data)

The Tx Data output line transfers serial data to a modem or other peripheral. Data rates in the range of 0 to 500 kbps when external synchronization is utilized.

Modem Control

The ACIA includes several functions that permit limited control of a peripheral or modem. The functions included are $\overline{CTS}$, $\overline{RTS}$ and $\overline{DCD}$.

Clear-to-Send ($\overline{CTS}$)

This high impedance TTL compatible input provides automatic control of the transmitting end of a communications link via the modem $\overline{CTS}$ active "Low" output by inhibiting the Transmit Data Register Empty (TDRE) status bit. (Refer to Figure 15.)

Request-to-Send ($\overline{RTS}$)

The $\overline{RTS}$ output enables the MPU to control a peripheral or modem via the data bus. The $\overline{RTS}$ output corresponds to the state of the Control Register bits CR5 and CR6. When CR6=0 or both CR5 and CR6=1, the $\overline{RTS}$ output is "Low" (the active state). This output can also be used for Data Terminal Ready ($\overline{DTR}$). (Refer to Figure 15.)

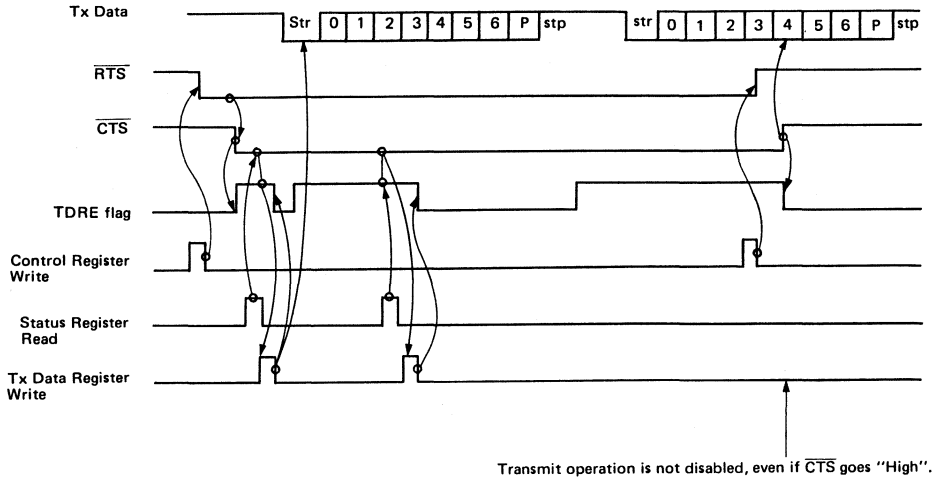


Figure 15 RTS and CTS Timing Chart (Example of 2 bytes transmission)

Data Carrier Detect (DCD)

DCD is the input signal corresponding to the "carrier detect" signal which shows carrier detect of modem.

DCD signal is used to control the receiving operation. When DCD input goes "High", ACIA stops all the receiving operation and sets receiving part in reset status. It means that receive shift register stops shifting, error detection circuit and synchro'zation circuit of receive clock are reset. When DCD is in "High" level, the receiving part of ACIA is kept in initial status and the operation in the receiving part is prohibited. When DCD goes "Low", the receiving part is allowed to receive data. In this case, the following process is needed to reset DCD flag and restarts the receive operation. (Refer to Figure 16.)

(1) Return DCD input from "High" to "Low".
(2) Read status register. (DCD flag = "1")
(3) Read receive data register (Uncertain data will be read.)

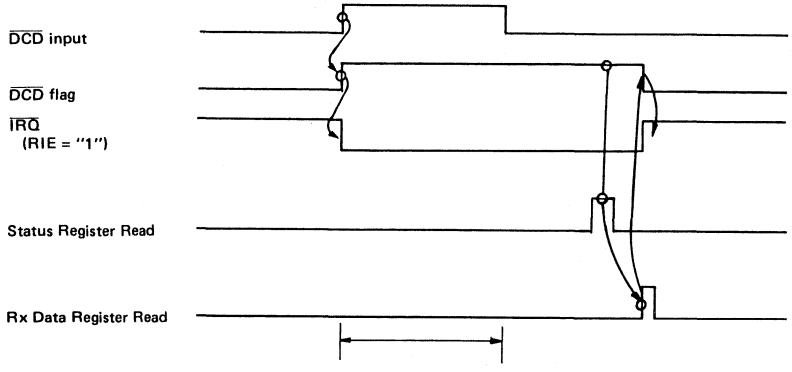


Figure 16 DCD Flag Timing Chart

■ **Note for Use** (HD6350 only)

Input Signal, which is not necessary for user's application, should be used fixed to "High" or "Low" level. This is applicable to the following signal pins.
Rx Data, Rx CLK, Tx CLK, CTS, DCD

HD6852, HD68A52

SSDA (Synchronous Serial Data Adapter)

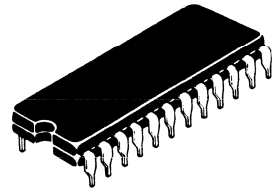
The HD6852 Synchronous Serial Data Adapter provides a bi-directional serial interface for synchronous data information interchange. It contains interface logic for simultaneously transmitting and receiving standard synchronous communications characters in bus organized systems such as the HMCS6800 Microprocessor systems.

The bus interface of the HD6852 includes select, enable, read/write, interrupt, and bus interface logic to allow data transfer over an 8-bit bi-directional data bus. The parallel data of the bus system is serially transmitted and received by the synchronous data interface with synchronization, fill character insertion/deletion, and error checking. The functional configuration of the SSDA is programmed via the data bus during system initialization.

Programmable control registers provide control for variable word length, transmit control, receive control, synchronization control and interrupt control. Status, timing and control lines provide peripheral or modem control.

Typical applications include data communications terminals, floppy disk controllers, cassette or cartridge tape controllers and numerical control systems.

HD6852P, HD68A52P

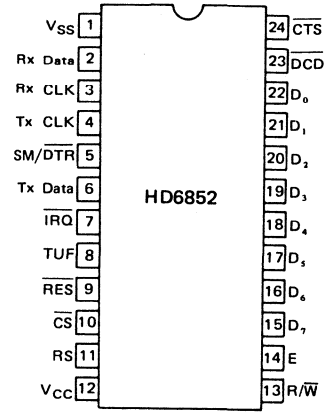


(DP-24)

■ FEATURES

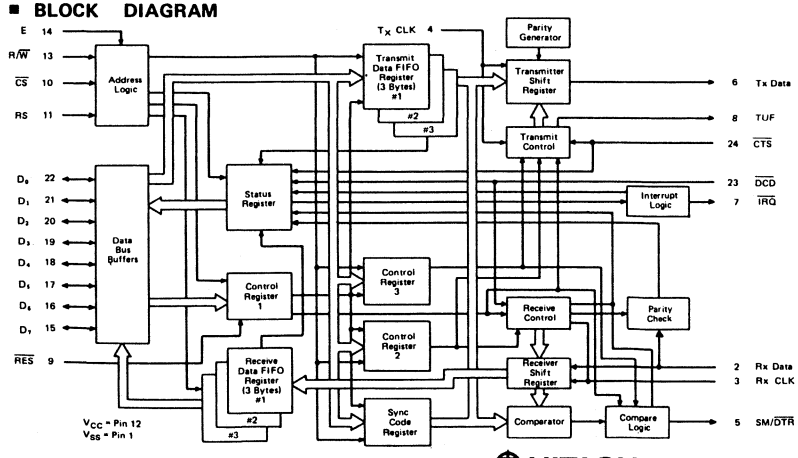
- Programmable Interrupts from Transmitter, Receiver, and Error Detection Logic
- Character Synchronization on One or Two Sync Codes
- External Synchronization Available for Parallel-Serial Operation
- Programmable Sync Code Register
- Up to 1Mkbps Transmitter
- Peripheral/Modem Control Functions
- Three Bytes of FIFO Buffering on Both Transmit and Receive
- 6, 7, or 8 Bit Data Transmission
- Optional Even and Odd Parity
- Parity, Overrun, and Underflow Status
- Compatible with MC6852 and MC68A52

■ PIN ARRANGEMENT



(Top View)

■ BLOCK DIAGRAM



HITACHI

■ ABSOLUTE MAXIMUM RATINGS

Item	Symbol	Value	Unit
Supply Voltage	V_{CC}*	−0.3 ~ +7.0	V
Input Voltage	V_{in}*	−0.3 ~ +7.0	V
Operating Temperature	T_{opr}	− 20 ~ + 75	°C
Storage Temperature	T_{stg}	−55 ~ +150	°C

* With respect to V_{SS} (SYSTEM GND)

(NOTE) Permanent LSI damage may occur if maximum ratings are exceeded. Normal operation should be under recommended operating conditions. If these conditions are exceeded, it could affect reliability of LSI.

■ RECOMMENDED OPERATING CONDITIONS

Item	Symbol	min	typ	max	Unit
Supply Voltage	V_{CC}*	4.75	5.0	5.25	V
Input Voltage	V_{IL}*	−0.3	–	0.8	V
	V_{IH}*	2.0	–	V_{CC}	V
Operating Temperature	T_{opr}	− 20	25	75	°C

* With respect to V_{SS} (SYSTEM GND)

■ ELECTRICAL CHARACTERISTICS

• DC CHARACTERISTICS (V_{CC} = 5V ± 5%, V_{SS} = 0V, Ta = −20~+75°C, unless otherwise noted.)

Item		Symbol	Test Condition	min	typ*	max	Unit
Input "High" Voltage	All Input	V_{IH}		2.0	–	–	V
Input "Low" Voltage	All Input	V_{IL}		−0.3	–	0.8	V
Output "High" Voltage	D_0~D_7	V_{OH}	I_{OH} = −205 μA, PW_{EH}, $PW_{EL} \leqq 25\mu s$	2.4	–	–	V
	Tx Data $\overline{DTR}$, TUF	V_{OH}	I_{OH} = −100 μA, PW_{EH}, $PW_{EL} \leqq 25\mu s$	2.4	–	–	V
Output "Low" Voltage	All Output	V_{OL}	I_{OL} = 1.6 mA, PW_{EH}, $PW_{EL} \leqq 25\mu s$	–	–	0.4	V
Input Leakage Current	TxCLK, RxCLK, Rx Data, E, $\overline{RES}$, RS, R/$\overline{W}$ $\overline{CS}$, $\overline{DCD}$, $\overline{CTS}$	I_{in}	V_{in} = 0 ~ 5.25V	−2.5	–	2.5	μA
Three-State Input Current (Off State)	D_0~ D_7	I_{TSI}	V_{in} = 0.4 ~ 2.4V, V_{CC} = 5.25V	−10	–	10	μA
Output Leakage Current (Off State)	$\overline{IRQ}$	I_{LOH}	V_{OH} = 2.4V	–	–	10	μA
Power Dissipation		P_D		–	300	525	mW
Input Capacitance	D_0~ D_7	C_{in}	V_{in} = 0V, Ta = 25°C, f = 1 MHz	–	–	12.5	pF
	RxData, RxCLK, TxCLK, $\overline{RES}$, $\overline{CS}$, RS, R/$\overline{W}$, E, $\overline{DCD}$, $\overline{CTS}$			–	–	7.5	
Output Capacitance	TxData, $\overline{DTR}$, TUF,	C_{out}	V_{in} = 0V, Ta = 25°C f = 1 MHz	–	–	10	pF
	$\overline{IRQ}$			–	–	5.0	

* Ta = 25°C, V_{CC} = 5V

- **AC CHARACTERISTICS** (V_{CC}=5V±5%, V_{SS}=0V, Ta=-20~+75°C, unless otherwise noted.)

1. TIMING OF THE DATA TRANSFER

Item	Symbol	Test Condition	HD6852			HD68A52			Unit
			min	typ	max	min	typ	max	
Clock "Low" Pulse Width	PW_{CL}	Fig. 1	700	–	–	400	–	–	ns
Clock "High" Pulse Width	PW_{CH}	Fig. 2	700	–	–	400	–	–	ns
Clock Frequency	f_C		–	–	600	–	–	1,000	kHz
Receive Data Setup Time	t_{RDSU}	Fig. 3	350	–	–	200	–	–	ns
Receive Data Hold Time	t_{RDH}	Fig. 3	350	–	–	200	–	–	ns
Sync Match Delay Time	t_{SM}	Fig. 3	–	–	1.0	–	–	0.666	μs
Clock-to-Data Delay for Transmitter	t_{TDD}	Fig. 4,6	–	–	1.0	–	–	0.666	μs
Transmitter Underflow	t_{TUF}	Fig. 4	–	–	1.0	–	–	0.666	μs
$\overline{DTR}$ Delay Time	t_{DTR}	Fig. 5	–	–	1.0	–	–	0.666	μs
$\overline{IRQ}$ Release Time	t_{IR}	Fig. 5	–	–	1.2	–	–	0.8	μs
$\overline{RES}$ Pulse Width	t_{RES}		1.0	–	–	0.666	–	–	μs
$\overline{CTS}$ Setup Time	t_{CTS}	Fig. 6	200	–	–	150	–	–	ns
$\overline{DCD}$ Setup Time	t_{DCD}	Fig. 7	500	–	–	350	–	–	ns
Input Rise and Fall Times(Except E)	t_r, t_f	0.8V to 2.0V	–	–	1.0*	–	–	1.0*	μs

* 1.0μ or 10% of the pulse width, whichever is smaller.

2. BUS TIMING

1) READ

Item	Symbol	Test Condition	HD6852		HD68A52		Unit
			min	max	min	max	
Enable Cycle Time	t_{cycE}	Fig. 8	1.0	–	0.666	–	μs
Enable "High" Pulse Width	PW_{EH}		0.45	25	0.28	25	μs
Enable "Low" Pulse Width	PW_{EL}		0.43	–	0.28	–	μs
Setup Time, Address and R/$\overline{W}$ valid to Enable positive transition	t_{AS}		140	–	140	–	ns
Data Delay Time	t_{DDR}		–	320	–	220	ns
Data Hold Time	t_H		10	–	10	–	ns
Address Hold Time	t_{AH}		10	80	10	80	ns
Rise and Fall Time for Enable input	t_{Er}, t_{Ef}		–	25	–	25	ns

2) WRITE

Item	Symbol	Test Condition	HD6852		HD68A52		Unit
			min	max	min	max	
Enable Cycle Time	t_{cycE}	Fig. 9	1.0	–	0.666	–	μs
Enable Pulse Width, "High"	PW_{EH}		0.45	25	0.28	25	μs
Enable Pulse Width, "Low"	PW_{EL}		0.43	–	0.28	–	μs
Setup Time, Address and R/$\overline{W}$ valid to Enable positive transition	t_{AS}		140	–	140	–	ns
Data Setup Time	t_{DSW}		195	–	80	–	ns
Data Hold Time	t_H		10	–	10	–	ns
Address Hold Time	t_{AH}		10	–	10	–	ns
Rise and Fall Time for Enable input	t_{Er}, t_{Ef}		–	25	–	25	ns

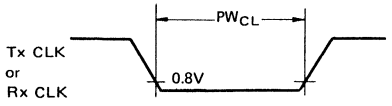


Figure 1 Clock Pulse Width ("Low" level)

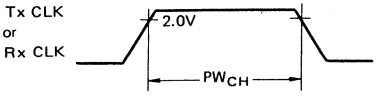


Figure 2 Clock Pulse Width ("High" level)

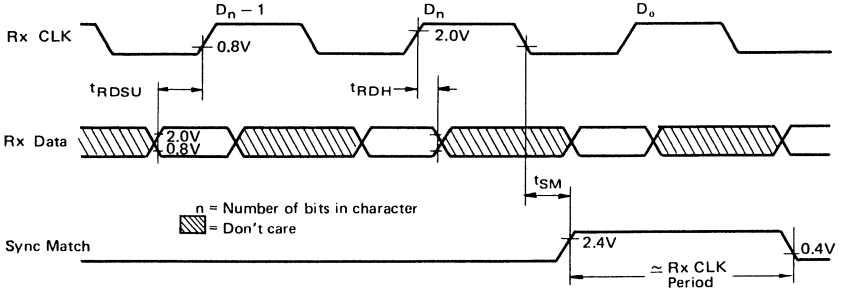


Figure 3 Receive Data Setup and Hold Times and Sync Match Delay Time

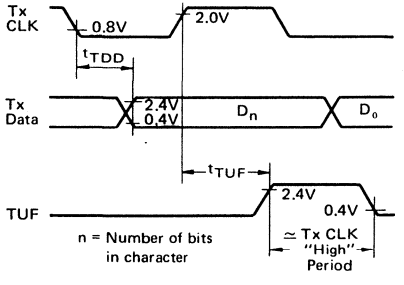
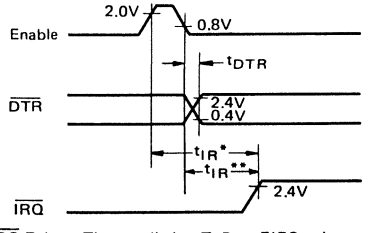


Figure 4 Transmit Data Output Delay and Transmitter Underflow Delay Time



* $\overline{IRQ}$ Release Time applied to TxData FIFO write operation and RxData FIFO read operation.

** $\overline{IRQ}$ Release Time applied to write "1" operation to RxRS, TxRS, CTUF, Clear $\overline{CTS}$ bits.

Figure 5 $\overline{DTR}$ and $\overline{IRQ}$ Release Time

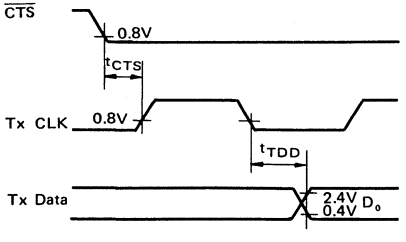
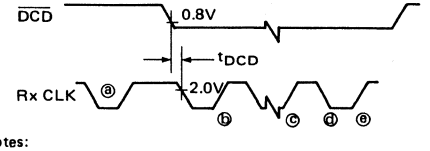


Figure 6 $\overline{CTS}$ Setup Time



Notes:
- ⓐ Must occur before $\overline{DCD}$ goes low.
- ⓑ First data bit placed in Rx shift register.
- ⓒ Last data bit of byte placed in Rx shift register.
- ⓓ Rx Data byte transferred from shift register to Rx FIFO.
- ⓔ Clock edge required for generation of $\overline{IRQ}$ by RDA status.

Note: Refer to Figure 3 for the Rx Data setup and hold times.

Figure 7 $\overline{DCD}$ Setup Time

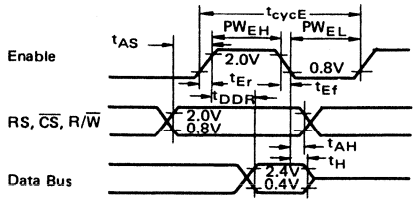


Figure 8 Bus Read Timing Characteristics (Read information from SSDA)

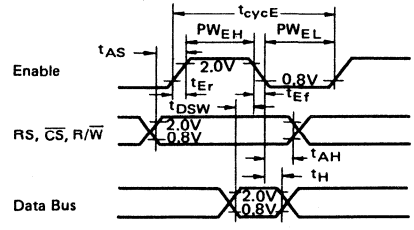


Figure 9 Bus Write Timing Characteristics (Write information into SSDA)

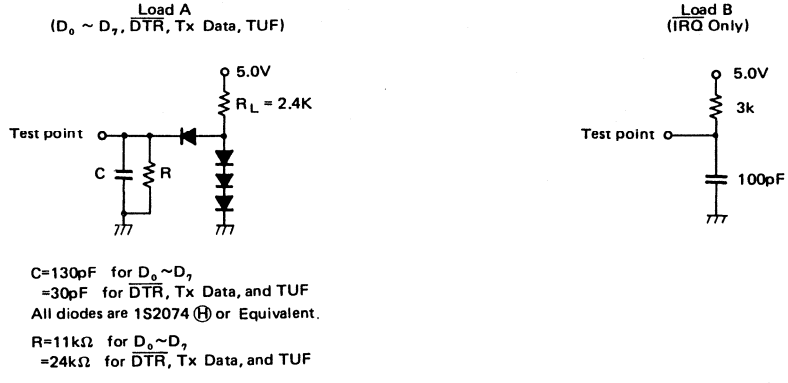


Figure 10 Test Loads

■ DEVICE OPERATION

At the bus interface, the SSDA appears as two addressable memory locations. Internally, there are seven registers: two read-only and five write-only registers. The read-only registers are Status and Receive Data; the write-only registers are Control 1, Control 2, Control 3, Sync Code and Transmit Data. The serial interface consists of serial input and output lines with independent clocks, and four peripheral/modem control lines.

Data to be transmitted is transferred directly into the 3-byte Transmit Data First-In First-Out (FIFO) Register from the data bus. Availability of the input to the FIFO is indicated by a bit in the Status Register; once data is entered, it moves through the FIFO to the last empty location. Data at the output of the FIFO is automatically transferred from the FIFO to the Transmitter Shift Register as the shift register becomes available to transmit the next character. If data is not available from the FIFO (underflow condition), the Transmitter Shift Register is automatically loaded with either a sync code or an all "1"s character. The transmit section may be programmed to append even, odd, or no parity to the transmitted word. An external control line ($\overline{CTS}$) is provided to inhibit the transmitter without clearing the FIFO.

Serial data is accumulated in the receiver based on the synchronization mode selected. In the external sync mode used for parallel-serial operation, the receiver is synchronized by the Data Carrier Detect ($\overline{DCD}$) input and transfers successive bytes of data to the input of the Receiver FIFO. The single-sync-character mode requires that a match occur between the Sync Code Register and one incoming character before data transfer to the FIFO begins. The two-sync-character mode requires that two sync codes be received in sequence to establish synchronization. Subsequent to synchronization in any mode, data is accumulated in the shift register, and parity is optionally checked. An indication of parity error is carried through the Receiver FIFO with each character to the last empty location. Availability of a word at the FIFO output is indicated by a bit in the Status Register, as is a parity error.

The SSDA and its internal registers are selected by the address bus, Read/Write (R/$\overline{W}$) and Enable control lines. To configure the SSDA, Control Registers are selected and the appropriate bits set. The Status Register is addressable for reading status.

Other I/O lines, in addition to Clear-to-Send ($\overline{CTS}$) and Data Carrier Detect ($\overline{DCD}$), include Sync Match/Data Terminal Ready (SM/$\overline{DTR}$) and Transmitter Underflow (TUF). The transmitter and receiver each have individual clock inputs allowing simultaneous operation under separate clock control. Signals to the microprocessor are the Data bus and Interrupt Request ($\overline{IRQ}$).

- **Initialization**

During a power-on sequence, the SSDA is reset via the $\overline{RES}$ input and internally latched in a reset condition to prevent erroneous output transmissions. The Sync Code Register, Control Register 2, and Control Register 3 should be programmed prior to the programmed release of the Transmitter and/or Receiver Reset bits; these bits in Control Register 1 should be cleared after the $\overline{RES}$ line has gone "High".

- **Transmitter Operation**

Data is transferred to the transmitter section in parallel form by means of the data bus and Transmit Data FIFO. The Transmit Data FIFO is a 3-byte register whose status is indicated by the Transmitter Data Register Available status bit (TDRA) and its associated interrupt enable bit. Data is transferred through the FIFO on negative edges of Enable (E) pulses. Two data transfer modes are provided in the SSDA. The 1-byte transfer mode provides for writing data to the transmitter section (and reading from the receiver section) one byte at a time. The 2-byte transfer mode provides for writing two data characters in succession.

Data will automatically transfer from the last register location in the Transmit Data FIFO (when it contains data) to the Transmitter Shift Register during the last half of the last bit of the previous character. A character is transferred into the Shift Register by the Transmitter Clock. Data is transmitted LSB first, and odd or even parity can be optionally appended. The unused bit positions in short word length characters from the data bus are "don't cares". (Note: The data bus inputs may be reversed for applications requiring the MSB to be transferred taken, e.g., IBM format for floppy disks; however, care must be taken to properly program the control registers – Table 1 will have its bit positions reversed.)

When the Shift Register becomes empty, and data is not available for transfer from the Transmit Data FIFO, an "underflow" occurs, and a character is inserted into the transmitter data stream to maintain character synchronization. The character transmitted on underflow will be either a "Mark" (all "1"s) or the contents of the Sync Code Register, depending upon the state of the Transmit Sync Code on Underflow control bit. The underflow condition is indicated by a pulse ($\approx$ Tx CLK "High" period) on the Underflow putput (when in Tx Sync on underflow mode). The Underflow output occurs coincident with the transfer of the last half of the last bit preceding the underflow character. The Underflow status bit is set until cleared by means of the Clear Underflow control bit. This output may be used in floppy disk systems to synchronize write operations and for appending CRCC.

Transmission is initiated by clearing the Transmitter Reset bit in Control Register 1. When the Transmitter Reset bit is cleared, the first full positive half-cycle of the Transmit Clock will initiate the transmit cycle, with the transmission of data or underflow characters beginning on the negative edge of the Transmit Clock pulse which started the cycle. If the Transmit Data FIFO was not loaded, an underflow character will be transmitted.

The Clear-to-Send ($\overline{CTS}$) input provides for automatic control of the transmitter by means of external system hardware; e.g., the modem $\overline{CTS}$ output provides the control in a data communications system. The $\overline{CTS}$ input resets and inhibits the transmitter section when "High", but does not reset the Transmit Data FIFO. The TDRA status bit is inhibited by $\overline{CTS}$ being "High" in either the one-sync character or two-sync-character mode of operation.

In the external sync mode, TDRA is unaffected by $\overline{CTS}$ in order to provide Transmit Data FIFO status for preloading and operating the transmitter under the control of the $\overline{CTS}$ input. When the Transmitter Reset bit (Tx Rs) is set, the Transmit Data FIFO is cleared and the TDRA status bit is cleared. After one E clock has occurred, the Transmit Data FIFO becomes available for new data with TDRA inhibited.

- **Receiver Operation**

Data and a presynchronized clock are provided to the SSDA receiver section by means of the Receive Data (Rx Data) and Receive Clock (Rx CLK) inputs. The data is a continuous stream of binary data bits without means for identifying character boundaries within the stream. It is, therefore, necessary to achieve character synchronization for the data at the beginning of the data block. Once synchronization is achieved, it is assumed to be retained for all successive characters within the block.

Data communications systems utilize the detection of sync codes during the initial portion of the preamble to establish character synchronization. This requires the detection of a single code or two successive sync codes. Floppy disk and cartridge tape units require sixteen bits of defined preamble and cassettes require eight bits of preamble to establish the reference for the start of record. All three are functionally equivalent to the detection of sync codes. Systems which do not utilize code detection techniques require custom logic external to the SSDA for character synchronization and use of the parallel-to-serial (external sync) mode.
(Note: The Receiver Shift Register is set to ones when reset)

- **Synchronization**

The SSDA provides three operating modes with respect to character synchronization: one-sync-character mode, two-sync-character mode, and external sync mode. The external sync mode requires synchronization and control of the receiving section through the Data Carrier Detect ($\overline{DCD}$) input. This external synchronization could consist of direct line control from the transmitting end of the serial data link or from external logic designed to detect the start of the message block. The one-sync-character mode searches on a bit-by-bit basis until a match is achieved between the data in the Shift Register and the Sync Code Register. The match indicates character synchronization is complete and will be retained for the message block. In the two-sync-character mode, the receiver searches for the first sync code match on a bit-by-bit basis and then looks for a second successive sync code character prior to establishing character synchronization. If the second sync code character is not received, the bit-by-bit search for the first sync code is resumed.

Sync codes received prior to the completion of synchronization (one or two character) are not transferred to the Receive Data FIFO. Redundant sync codes during the preamble or sync codes which occure as "fill characters" can automatically be stripped from the data, when the Strip Sync control bit is set, to minimize system loading. The character synchronization will be retained until cleared by means of the Clar Sync bit, which also inhibits synchronization search when set.

- **Receiving Data**

Once synchronization has been achieved, subsequent characters are automatically transferred into the Receive Data FIFO and clocked through the FIFO to the last empty location by E pulses (MPU System $\phi 2$). The Receiver Data Available status bit

(RDA) indicates when data is available to be read from the last FIFO location (#3) when in the 1-byte transfer mode. The 2-byte transfer mode causes the RDA status bit to indicate data is available when the last two FIFO register locations are full. Data being available in the Receive Data FIFO causes an interrupt request if the Receiver Interrupt Enable (RIE) bit is set. The MPU will then read the SSDA Status Register, which will indicate that data is available for the MPU read from the Receiver Data FIFO register. The $\overline{IRQ}$ and RDA status bits are reset by a read from the FIFO. If more than one character has been received and is resident in the Receive Data FIFO, subsequent E clocks will cause the FIFO to update and the RDA and $\overline{IRQ}$ status bits will again be set. The read data operation for the 2-byte transfer mode requires an intervening E clock between reads to allow the FIFO data to shift. Optional parity is automatically checked as data is received, and the parity status condition is maintained with each character until the data is read from the Receive Data FIFO. Parity errors will cause an interrupt request if the Error Interrupt Enable (EIE) has been set. The parity bit is not transferred to the data bus but must be checked in the Status Register. NOTE: In the 2-byte transfer mode, parity should be checked prior to reading the second byte, since a FIFO read clears the error bit.

Other status bits which pertain to the receiver section are Receiver Overrun and Data Carrier Detect ($\overline{DCD}$). The Overrun status bit is automatically set when a transfer of a character to the Receive Data FIFO occurs and the first register of the Receive Data FIFO is full. Overrun causes an interrupt if Error Interrupt Enable (EIE) has been set. The transfer of the overrunning character into the FIFO causes the previous character in the FIFO input register location to be lost. The Overrun status bit is cleared by reading the Status Register (when the overrun condition is present), followed by a Receive Data FIFO Register read. Overrun cannot occur and be cleared without providing an opportunity to detect its occurrence via the Status Register.

A positive transition on the $\overline{DCD}$ input causes an interrupt if the EIE control bit has been set. The interrupt caused by $\overline{DCD}$ is cleared by reading the Status Register when the $\overline{DCD}$ status bit is "1", followed by a Receive Data FIFO read. The $\overline{DCD}$ status bit will subsequently follow the state of the $\overline{DCD}$ input when it goes "Low".

■ SSDA REGISTERS

Seven registers in the SSDA can be accessed by means of the bus. The registers are defined as read-only or write-only according to the direction of information flow. The Register Select (RS) input selects two registers in each state, one being read-only and the other write-only. The Read/Write (R/$\overline{W}$) input defined which of the two selected registers will actually be accessed. Four registers (two read-only and two write-only) can be addressed via the bus at any particular time. These registers and the required addressing are defined in Table 1.

• Control Register 1 (C1)

Control Register 1 is an 8-bit wirte-only register that can be directly addressed from the data bus. Control Register 1 is addressed when RS = "Low" and R/$\overline{W}$ = "Low".

Receiver Reset (Rx Rs), C1 Bit 0

The Receiver Reset control bit provides both a reset and inhibit function to the receiver section. When Rx Rs is set, it clears the receiver control logic, error logic, Rx Data FIFO Control, Parity Error status bit, and $\overline{DCD}$ interrupt. The Receiver Shift Register is set ones. The Rx Rs bit must be cleared after the occurrence of a "Low" level on $\overline{RES}$ in order to enable the receiver section of the SSDA.

Transmitter Reset (Tx Rs), C1 Bit 1

The Transmitter Reset control bit provides both a reset and inhibit to the transmitter section. When Tx Rs is set, it clears the transmitter control section, Transmitter Shift Register, Tx Data FIFO Control (the Tx Data FIFO can be reloaded after one E clock pulse), the Transmitter Underflow status bit, and the $\overline{CTS}$ interrupt, and inhibits the TDRA status bit (in the one-sync-character and two-sync-character modes). The Tx Rs bit must be cleared after the occurrence of a "Low" level on $\overline{RES}$ in order to enable the transmitter section of the SSDA. If the Tx FIFO is not preloaded, it must be loaded immediately after the Tx Rs release to prevent a transmitter underflow condition.

Strip Synchronization Characters (Strip Sync), C1 Bit 2

If the Strip Sync bit is set, the SSDA will automatically strip all received characters which match the contents of the Sync Code Register. The characters used for synchronization (one or two characters of sync) are always stripped from the received data stream.

Clear Synchronization (Clear Sync), C1 Bit 3

The Clear Sync control bit provides the capability of dropping receiver character synchronization and inhibiting resynchronization. The Clear Sync bit is set to clear and inhibit receiver synchronization in all modes and is reset to zero to enable resynchronization.

Transmitter Interrupt Enable (TIE), C1 Bit 4

TIE enable both the Interrupt Request ($\overline{IRQ}$) output and Interrupt Request status bit to indicate a transmitter service request. When TIE is set and the TDRA status bit is "1", the $\overline{IRQ}$ output will go "Low" (the active state) and the $\overline{IRQ}$ status bit will go "1".

Receiver Interrupt Enable (RIE), C1 Bit 5

RIE enable both the Interrupt Request output ($\overline{IRQ}$) and the Interrupt Request status bit to indicate a receiver service request. When RIE is set and the RDA status bit is "1", the $\overline{IRQ}$ output will go "Low" (the active state) and the $\overline{IRQ}$ status bit will go "1".

Address Control 1 (AC1) and Address Control 2 (AC2), C1 Bits 6 and 7

AC1 and AC2 select one of the write-only registers – Control 2, Control 3, Sync Code, or Tx Data FIFO – as shown in Table 1, when RS = "High" and R/$\overline{W}$ = "Low".

• Control Register 2 (C2)

Control Register 2 is an 8-bit write-only register which can be programmed from the bus when the Address Control bits in Control Register 1 (AC1 and AC2) are reset, RS = "High" and R/$\overline{W}$ = "Low".

Peripheral Control 1 (PC1) and Peripheral Control 2 (PC2), C2 Bits 0 and 1

Two control bits, PC1 and PC2, determine the operating characteristics of the Sync Match/$\overline{DTR}$ output. PC1, when "High", selects the Sync Match mode. PC2 provides the inhibit/

enable control for the SM/$\overline{DRT}$ output in the Sync Match mode. A one-bit-wide pulse is generated at the output when PC2 is "0", and a match occurs between the contents of the Sync Code Register and the incoming data even if sync is inhibited (Clear Sync bit = "1"). The Sync Match pulse is referenced to the negative edge of Rx CLK pulse causing the match.

The Data Terminal Ready ($\overline{DTR}$) mode is selected when PC1 is "0". When PC2 = "1" the SM/$\overline{DTR}$ output = "Low" and vice versa. The operation of PC2 and PC1 is summarized in Table 4.

1-Byte/2-Byte Transfer (1-Byte/2-Byte), C2 Bit 2

When 1-Byte/2-Byte is set, the TDRA and RDA status bits will indicate the availability if their respective data FIFO registers for a single byte data transfer. Alternately, if 1-Byte/2-Byte is reset, the TDRA and RDA status bits indicate when two bytes of data can be moved without a second status read. An intervening Enable pulse must occur between data transfers.

Word Length Selects (WS1, WS2, WS3), C2 Bits 3, 4, 5

Word length Select bits WS1, WS2, and WS3 select word length of 7, 8, or 9 bits including parity as shown in Table 3.

Transmit Sync Code on Underflow (Tx Sync), C2 Bit 6

When Tx Sync is set, the transmitter will automatically send a sync character when data is not available for transmission. If Tx Sync is reset, the transmitter will transmit a Mark character (including the parity bit position) on underflow. When the underflow is detected, a pulse approximately a Tx CLK "High" period wide will occur on the underflow output if the Tx Sync bit is "1". Internal parity generation is inhibited during underflow except for sync code fill character transmission in 8 bit plus parity word lengths.

Error Interrupt Enable (EIE), C2 Bit 7

When EIE is set, the $\overline{IRQ}$ status bit will go "1" and the $\overline{IRQ}$ output will go "Low" if:

1) A receiver overrun occurs. The interrupt is cleared by reading the Status Register and reading the Rx Data FIFO.
2) $\overline{DCD}$ input has gone to a "High". The interrupt is cleared by reading the Status Register and reading the Rx Data FIFO.
3) A parity error exists for the character in the last location (#3) of the Rx Data FIFO. The interrupt is cleared by reading the Rx Data FIFO. The interrupt is cleared by reading the Rx Data FIFO.
4) The $\overline{CTS}$ input has gone to a "High". The interrupt is cleared by writing a "1" in the Clear $\overline{CTS}$ bit, C3 bit 2, or by a Tx Reset.
5) The transmitter has underflowed (in the Tx Sync on Underflow mode). The interrupt is cleared by writing a "1" into the Clear Underflow, C3 bit 3, or Tx Reset.

When EIE is a "0", the $\overline{IRQ}$ status bit and the $\overline{IRQ}$ output are disabled for the above error conditions. A "Low" level on the $\overline{RES}$ input resets EIE to "0".

- **Control Register 3 (C3)**

Control Register 3 is a 4-bit write-only register which can be programmed from the bus when RS = "High" and R/$\overline{W}$ = "Low" and Address Control bit AC1 = "1" and AC2 = "0".

External/Internal Sync Mode Control (E/I Sync), C3 Bit 0

When the E/I Sync Mode bit is "1", the SSDA is in the external sync mode and the receiver synchronization logic is disabled. Synchronization can be achieved by means of the $\overline{DCD}$ input or by starting Rx CLK at the midpoint of data bit "0" of a character with $\overline{DCD}$ "Low". Both the transmitter and receiver sections operate as parallel – serial converters in the External Sync mode. The Clear Sync bit in Control Register 1 acts as a receiver sync inhibit when "High" to provide a bus controllable inhibit. The Sync Code Register can serve as a transmitter fill character register and a receiver match register in this mode. A "Low" on the $\overline{RES}$ input resets the E/I Sync Mode bit placing the SSDA In the internal sync mode.

One-Sync-Character/Two-Sync-Character Mode, Control (1 Sync/2 Sync), C3 Bit 1

When the 1 Sync/2 Sync bit is set, the SSDA will synchronize on a single match between the received data and the contents of the Sync Code Register. When the 1 Sync/2 Sync bit is reset, two successive sync characters must be received prior to receiver synchronization. If the second sync character is not detected, the bit by bit search resumes from the first bit in the second character. See the description of the Sync Code Register for more details.

Clear $\overline{CTS}$ Status (Clear $\overline{CTS}$), C3 Bit 2

When a "1" is written into the Clear $\overline{CTS}$ bit, the stored status and interrupt are cleared. Subsequently, the $\overline{CTS}$ status bit reflects the state of the $\overline{CTS}$ input. The Clear $\overline{CTS}$ control bit does not affect the $\overline{CTS}$ input nor its inhibit of the transmitter secton. The Clear $\overline{CTS}$ command bit is self-clearing, and writing a "0" into this bit is a nonfunctional operation.

Clear Transmit Underflow Status (CTUF), C3 Bit 3

When a "1" is written into the CTUF status bit, the CTUF bit and its associated interrupt are reset. The CTUF command bit is self-clearing and writing a "0" into this bit is a nonfunctional operation.

- **Sync Code Register**

The Sync Code Register is an 8-bit register for storing the programmable sync code required for received data character synchronization in the one-sync-character and two-sync-character modes. The Sync Code Register also provides for stripping the sync/fill characters from the received data (a programmable option) as well as automatic insertion of fill characters in the transmitted data stream. The Sync Code Register is not utilized for teceiver character synchronization in the external sync mode; however, it provides storage of receiver match and transmit fill characters.

The Sync Code Register can be loaded when AC2 and AC1 are a "1" and "0" respectively, and R/$\overline{W}$ = "Low" and RS = "High".

The Sync Code Register may be changed after the detection of a match with the received data (the first sync code having been detected) to synchronize with a double-word sync pattern. (This sync code change must occur prior to the completion of the second character.) The sync match (SM) output can be used to interrupt the MPU system to indicate that the first eight bits have matched. The service routine would then change the sync match register to the second half of the pattern. Alternately, the one-sync-character mode can be used for sync codes for 16 or more bits by using software to check the second and subsequent bytes after reading them from the FIFO.

The detection of the sync code can be programmed to appear on the Sync Match/$\overline{DTR}$ output by writing a "1" in PC1 (C2 bit 0) and a "0" in PC2 (C2 bit 1). The Sync Match output will go "High" for one bit time beginning at the character interface between the sync code and the next character.

- **Parity for Sync Character**

Transmitter

Transmitter does not generate parity for the sync character except 9-bit mode.

9-bit (8-bit + parity) – 8-bit sync character + parity
8-bit (7-bit + parity) – 8-bit sync character (no parity)
7-bit (6-bit + parity) – 7-bit sync character (no parity)

Receiver

At Synchronization

Receiver automatically strips the sync character(s) (two sync characters if '2 sync' mode is selected) which is used to establish synchronization. And parity is not checked for these sync characters.

After Synchronization is Established

When 'strip sync' bit is selected, the sync characters (fill characters) are stripped and parity is not checked for the stripped sync (fill) characters. When 'strip sync' bit is not selected (0), the sync character is assumed to be normal data and it is transferred into FIFO after parity checking. (When non-parity format is selected, parity is not checked.)

Strip Sync (C1 Bit 2)	Data Format (C2 Bit 3-5)	Operation
1	x	No transfer of sync code. No parity check of sync code.
0	With Parity	*Transfer data and sync codes. Parity check.
0	Without Parity	*Transfer data and sync codes. No parity check.

* Subsequent to synchronization
x ..... don't care

It is necessary to pay attention to the selected sync character in the following cases.

1) Data format is (6 + parity), (7 + parity),
2) Strip sync is not selected ("0").
3) After synchronization when sync code is used as a fill character.

Transmitter sends sync character without parity, but receiver checks the parity as if it is normal data. Therefore, the sync character should be chosen to match the parity check selected for the receiver in this special case.

- **Receive Data First-In First-Out Register (Rx Data FIFO)**

The Receive Data FIFO Register consists of three 8-bit registers which are used for buffer storage of received data. Each 8-bit register has an internal status bit which monitors its full or empty condition. Data is always transferred from a full register to an adjacent empty register. The transfer from register to register occurs on E pulses. The RDA status bit will be "1" when data is available in the last location of the Rx Data FIFO.

In an Overrun condition, the overrunning character will be transferred into the full first stage of the FIFO register and will cause the loss of that data character. Successive overruns continue to overwite the first register of the FIFO. This destruction of data is indicated by means of the Overrun status bit. The Overrun bit will be set when the overrun occurs and remains set until the Status Register is read, followed by a read of the Rx Data FIFO.

Unused data bits for short word lengths (including the parity bit) will appear as "0"s on the data bus when the Rx Data FIFO is read.

- **Transmit Data First-In First-Out Register (Tx Data FIFO)**

The Transmit Data FIFO Register consists of three 8-bit registers which are used for buffer storage of data to be transmitted. Each 8-bit register has an internal status bit which monitors its full or empty condition. Data is always transferred from a full register to an adjacent empty register. The transfer is clocked by E pulses.

The TDRA status bit will be "High" if the Tx Data FIFO is available for data.

Unused data bits for short word lengths will be handled as "don't cares". The parity bit is not transferred over the data bus since the SSDA generates parity at transmission.

When an Underflow occurs, the Underflow character will be either the contents of the Sync Code Register or an all "1"s character. The underflow will be stored in the Status Register until cleared and will appear on the Underflow output as a pulse approximately a Tx CLK "High" period wide.

- **Status Register**

The Status Register is an 8-bit read-only register which provides the real-time status of the SSDA and the associated serial data channel. Reading the Status Register is a non-destructive process. The method of clearing status bits depends upon the function each bit represents and is discussed for each bit in the register.

Receiver Data Available (RDA), S Bit 0

The Receiver Data Available status bit indicates when receiver data can be read from the Rx Data FIFO. The receiver data being present in the last register (#3) of the FIFO causes RDA to be "1" for the 1-byte transfer mode. The RDA bit being "1" indicates that the last two registers (#2 and #3) are full when in the 2-byte transfer mode. The second character can be read without a second status rad (to determine that the character is available). And E pulse must occur between reads of the Rx Data FIFO to allow the FIFO to shift. Status must be read on a word-by-word basis if receiver data error checking is important. The RDA status bit is reset automatically when data is not available.

Transmitter Data Register Available (TDRA), S Bit 1

The TDRA status bit indicates that data can be loaded into the Tx Data FIFO Register. The first register (#1) of the Tx Data FIFO being empty will be indicated by a "1" in the TDRA status bit in the 1-byte transfer mode. The first two registers (#1 and #2) must be empty for TDRA to be "1" when in the 2-byte transfer mode. The Tx Data FIFO can be loaded with two bytes without an intervening status read; however, one E pulse must occur between loads. TDRA is inhibited by the Tx Reset or $\overline{RES}$. When Tx Reset is set, the Tx Data FIFO is cleared and then released on the next E clock pulse. The Tx Data FIFO can then be loaded with up to three characters of data, even though TDRA is inhibited. This feature allows preloading data prior to the release of Tx Reset. A "High" level on the $\overline{CTS}$ input inhibits the TDRA status bit in either sync mode of operation (one-sync-character or two-sync-character). $\overline{CTS}$ does not affect TDRA in the external sync mode. This

enables the SSDA to operate under the control of the $\overline{CTS}$ input with TDRA indicating the status of the Tx Data FIFO. The $\overline{CTS}$ input does not clear the Tx Data FIFO in any operating mode.

Data Carrier Detect ($\overline{DCD}$), S Bit 2

A positive transition on the $\overline{DCD}$ input is stored in the SSDA until cleared by reading both Status and Rx Data FIFO. A "1" written into Rx Rs also clears the stored $\overline{DCD}$ status. The $\overline{DCD}$ status bit, when set, indicates that the $\overline{DCD}$ input has gone "High", The reading of both Status and Receive Data FIFO allows Bit 2 of subsequent Status reads to indicate the state of the $\overline{DCD}$ input until the next positive transition.

Clear-to-Send ($\overline{CTS}$), S Bit 3

A positive transiton on the $\overline{CTS}$ input is stored in the SSDA until cleared by writing a "1" into the Clear $\overline{CTS}$ control bit or the Tx Rs bit. The $\overline{CTS}$ status bit, when set, indicates that the $\overline{CTS}$ input has gone "High". The Clear $\overline{CTS}$ command (a "1" into C3 Bit 2) allows Bit 3 of subsequent Status reads to indicate the state of the $\overline{CTS}$ input until the next positive transition.

Transmitter Underflow (TUF), S Bit 4

When data is not available for the transmitter, an underflow occurs and is so indicated in the Status Register (in the Tx Sync on underflow mode). The underflow status bit is cleared by writing a "1" into the Clear Underflow (CTUF) control bit or the Tx Rs bit. TUF indicates that a sync character will be transmitted as the next character. A TUF is indicated on the output only when the contents of the Sync Code Register is to be transferred (transmit sync code on underflow = "1").

Receiver Overrun (Rx Ovrn), S Bit 5

Overrun indicates data has been received when the Rx Data FIFO is full, resulting in data loss. The Rx Ovrn status bit is set when Overrun occurs. The Rx Ovrn status bit is cleared by reading Status followed by reading the Rx Data FIFO or by setting the Rx Rs control bit.

Receiver Parity Error (PE), S Bit 6

The parity error status bit indicates that parity for the character in the last register of the Rx Data FIFO did not agree with selected parity. The parity error is cleared when the character to which it pertains is read from the Rx Data FIFO or when Rx Rs occurs. The $\overline{DCD}$ input does not clear the Parity Error or Rx Data FIFO status bits.

Interrupt Request ($\overline{IRQ}$), S Bit 7

The Interrupt Request status bit indicates when the $\overline{IRQ}$ output is in the active state ($\overline{IRQ}$ output = "Low"). The IRQ status bit is subject to the same interrupt enables (RIE, TIE, and EIE) as the $\overline{IRQ}$ output. The IRQ status but simplifies status inquiries for polling systems by providing single bit indication of service requests.

Table 1 SSDA Programming Model

Register	Control* Inputs		Address Control		Register Content							
	RS	R/$\overline{W}$	AC2	AC1	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Status (S)	0	1	X	X	Interrupt Request ($\overline{IRQ}$)	Receiver Parity Error (PE)	Receiver Overrun (Rx Ovrn)	Transmitter Underflow (TUF)	Clear-to-Send ($\overline{CTS}$)	Data Carrier Detect ($\overline{DCD}$)	Transmitter Data Register Available (TDRA)	Receiver Data Available (RDA)
Control 1 (C1)	0	0	X	X	Address Control 2 (AC2)	Address Control 1 (AC1)	Receiver Interrupt Enable (RIE)	Transmitter Interrupt Enable (TIE)	Clear Sync	Strip Sync Characters (Strip Sync)	Transmitter Reset (Tx Rs)	Receiver Reset (Rx Rs)
*** Receive Data FIFO	1	1	X	X	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
Control 2 (C2)	1	0	0	0	Error Interrupt Enable (EIE)	Transmit Sync Code on Underflow (Tx Sync)	Word Length Select 3 (WS3)	Word Length Select 2 (WS2)	Word Length Select 1 (WS1)	1-Byte/2-Byte Transfer (1-Byte/2-Byte)	Peripheral Control 2 (PC2)	Peripheral Control 1 (PC1)
Control 3 (C3)	1	0	0	1	Not Used	Not Used	Not Used	Not Used	Clear Transmitter Underflow Status (CTUF)	Clear $\overline{CTS}$ Status (Clear $\overline{CTS}$)	One-Sync-Character/ Two-Sync Character Mode Control (1 Sync/ 2 Sync)	External/ Internal Sync Mode Control (E/I Sync)
** Sync Code	1	0	1	0	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
*** Transmit Data FIFO	1	0	1	1	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0

* 0 ; "Low" level, 1 ; "High" level

** "FF" should not be used as Sync Code.

*** When the SSDA is used in applications requiring the MSB of data to be receive and transmitted first, the data bus inputs to the SSDA may be reversed (D_0 to D_7, etc.). Caution must be used when this is done since the bit positions in this table will be reversed, and the parity should not be selected.

Table 2 Functions of SSDA Register

Register	Bit	Symbol	Function			
Status Register (S)	7	$\overline{IRQ}$	The $\overline{IRQ}$ flag is cleared when the source of the $\overline{IRQ}$ is cleared. The source is determined by the enables in the Control Registers: TIE, RIE, EIE.			
	6	PE	Conditions for Set	When parity error is detected in receive data.	Conditions for Reset	Read Rx Data FIFO, or a "1" into Rx Rs (C1 Bit 0).
	5	Rx Ovrn		When receive data FIFO overruns.		Read Status and then Rx Data FIFO, or a "1" into Rx Rs (C1 Bit 0).
	4	TUF		When under flow is occurred in the transmitter.		A "1" into CTUF (C3 Bit 3) or into Tx Rs (C1 Bit 1).
	3	$\overline{CTS}$		When $\overline{CTS}$ signal rises.		A "1" into Clear $\overline{CTS}$ (C3 Bit 2) or a "1" into Tx Rs (C1 Bit 1)
	2	$\overline{DCD}$		When $\overline{DCD}$ signal rises.		Read Status and then Rx Data FIFO or a "1" into Rx Rs (C1 Bit 0)
	1	TDRA		1 Byte Transfer Mode; when the transmit data FIFO (#1) is empty.		Write into Tx Data FIFO.
				2 Byte Transfer Mode; when the transmit data FIFO (#1, #2) is empty.		
	0	RDA		1 Byte Transfer Mode; when the data is received in the receive data FIFO (#3).		Read Rx Data FIFO.
				2 Byte Transfer Mode; when the data is received in the receive data FIFO (#2, #3).		
Control Register 1 (C1)	7 6	AC2 AC1	Used to access other registers, as shown Table 1.			
	5	RIE	When "1", enables interrupt on RDA (S Bit 0).			
	4	TIE	When "1", enables interrupt on TDRA (S Bit 1).			
	3	Clear Sync	When "1", clears receiver character synchronization.			
	2	Strip Sync	When "1", strips all sync codes from the received data stream.			
	1	Tx Rs	When "1", resets and inhibits the transmitter section.			
	0	Rx Rs	When "1", resets and inhibits the receiver section.			
Control Register 2 (C2)	7	EIE	When "1", enables the PE, Rx Ovrn, TUF, $\overline{CTS}$, and $\overline{DCD}$ interrupt flags (S Bits 6 through 2).			
	6	Tx Sync	When "1", allows sync code contents to be transferred on underflow, and enables the TUF Status bit and output. When "0", an all mark character is transmitted on underflow.			
	5 4 3	WS3 WS2 WS1	Word Length Select			
	2	1-Byte/2-Byte	When "1", enables the TDRA and RDA bits to indicate when a 1-byte transfer can occur; when "0", the TDRA and RDA bits indicate when a 2-byte transfer can occur.			
	1 0	PC2 PC1	SM/$\overline{DTR}$ Output Control			
Control Register 3 (C3)	3	CTUF	When "1", clears TUF (S Bit 4), and $\overline{IRQ}$ if enabled.			
	2	Clear $\overline{CTS}$	When "1", clears $\overline{CTS}$ (S Bit 3), and $\overline{IRQ}$ if enabled.			
	1	1-Sync/2-Sync	When "1", selects the one-sync-character mode; when "0", selects the two-sync-character mode.			
	0	E/I Sync	When "1", selects the external sync mode; when "0", selects the internal sync mode.			

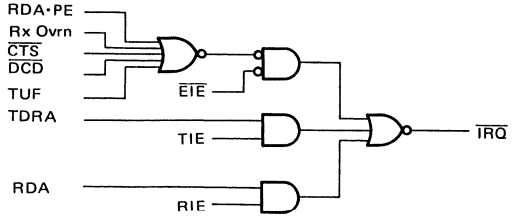
Table 3 Word Length

Bit 5 WS3	Bit 4 WS2	Bit 3 WS1	Word Length
0	0	0	6 Bits + Even Parity
0	0	1	6 Bits + Odd Parity
0	1	0	7 Bits
0	1	1	8 Bits
1	0	0	7 Bits + Even Parity
1	0	1	7 Bits + Odd Parity
1	1	0	8 Bits + Even Parity
1	1	1	8 Bits + Odd Parity

Table 4 SM/$\overline{DTR}$ Output Control

Bit 1 PC2	Bit 0 PC1	SM/$\overline{DTR}$ Output at Pin 5
0	0	"High" Level*
0	1	Pulse ⎍ 1-Bit Wide, on SM
1	0	"Low" Level*
1	1	SM Inhibited, "Low"*

* OUTPUT level is fixed by the data written into PC2, PC1.
** When "10" or "11", output is fixed at "Low".



■ INTERFACE SIGNALS FOR MPU

The SSDA interfaces to the HD6800 MPU with an 8-bit bi-directional data bus, a chip select line, a register select line, an interrupt request line, read/write line, an enable line, and a reset line. These signals, in conjunction with the HD6800 VMA output, permit the MPU to have complete control over the SSDA.

• Bi-Directional Data Bus (D_0~D_7)

The bi-directional data bus (D_0~D_7) allow for data transfer between the SSDA and the MPU. The data bus output drivers are three-state devices that remain in the high impedance (off) state except when the MPU performs an SSDA read operation.

• Enable (E)

The Enable signal, E, is a high impedance TTL compatible input that enables the bus input/output data buffers, clocks data to and from the SSDA, and moves data through the FIFO Registers. This signal is normally the continuous HMCS6800 System $\phi 2$ clock, so that incoming data characters are shifted through the FIFO.

• Read/Write (R/$\overline{W}$)

The Read/Write line is a high impedance input that is TTL compatible and is used to control the direction of data flow through the SSDA's input/output data bus interface. When Read/Write is "High" (MPU read cycle), SSDA output drivers are turned on if the chip is selected and a selected register is read. When it is "Low", the SSDA output drivers are turned off and the MPU writes into a selected register. The Read/Write signal is also used to select read-only or write-only registeres within the SSDA.

• Chip Select ($\overline{CS}$)

This high impedance TTL compatible input line is used to address the SSDA. The SSDA is selected when $\overline{CS}$ is "Low". VMA should be used in generating the $\overline{CS}$ input to insure that false selects will not occur. Transfers of data to and from the SSDA are then performed under the control of the Enable signal, Read/Write, and Register Select.

• Register Select (RS)

The Register Select line is a high impedance input that is TTL compatible. A "High" level is used to select Control Registers C2 and C3, the Sync Code Register, and the Transmit/Receive Data Registers. A "Low" level selects the Control 1 and Status Registers (see Table 1).

• Interrupt Request ($\overline{IRQ}$)

$\overline{IRQ}$ is a TTL compatible, open-drain (no internal pullup), active "Low" output that is used to interrupt the MPU. The $\overline{IRQ}$ remains "Low" until cleared by the MPU.

• Reset ($\overline{RES}$)

The $\overline{RES}$ input provides a means of resetting the SSDA from an external source. In the "Low" state, the $\overline{RES}$ input causes the following:

1) Receiver Reset (Rx Rs) and Transmitter Reset (Tx Rs) bits are set causing both the receiver and transmitter sections to be held in a reset condition.
2) Peripheral Control bits PC1 and PC2 are reset to zero, causing the SM/$\overline{DTR}$ output to be "High".
3) The Erfor Interrupt Enable (EIE) bit is reset.
4) An internal synchronization mode is selected.
5) The Transmitter Data Register Available (TDRA) status bit is cleared and inhibited.

When $\overline{RES}$ returns "High" (the inactive state), the transmitter and receiver sections will remain in the reset state until the Receiver Reset and Transmitter Reset bits are cleared via the bus under software control. The control Register bits affected by $\overline{RES}$ (Rx Rs, Tx Rs, PC1, PC2, EIE, and E/I Sync) cannot be changed when $\overline{RES}$ is "Low".

■ CLOCK INPUTS

Separate high impedance TTL compatible inputs are provided for clocking of transmitted and received data.

• Transmit Clock (Tx CLK)

The Transmit Clock input is used for the clocking of transmitted data. The transmitter shifts data on the negative transition of the clock.

• Receive Clock (Rx CLK)

The Receive Clock input is used for clocking in received data. The clock and data must be synchronized externally. The receiver samples the data on the positive transition of the clock.

■ SERIAL INPUT/OUTPUT LINES

• Receive Data (Rx Data)

The Receive Data line is a high impedance TTL compatible input through which data is received in a serial format. Data rates are from 0 to 600 kbps.

• Transmit Data (Tx Data)

The Transmit Data output line transfers serial data to a modem or other peripheral. Data rates are from 0 to 600 kbps.

■ PERIPHERAL/MODEM CONTROL

The SSDA includes several functions that permit limited control of a peripheral or modem. The functions included are $\overline{CTS}$, SM/$\overline{DTR}$, $\overline{DCD}$, and TUF.

• Clear-to-Send ($\overline{CTS}$)

The $\overline{CTS}$ input provides a real-time inhibit to the transmitter

section (the Tx Data FIFO is not disturbed). A positive $\overline{CTS}$ transition resets the Tx Shift Register and inhibits the TDRA status bit and its associated interrupt in both the one-sync-character and two-sync-character modes of operation. TDRA is not affected by the $\overline{CTS}$ input in the external sync mode.

The positive transition of $\overline{CTS}$ is stored within the SSDA to insure that its occurrence will be acknowledged by the system. The stored $\overline{CTS}$ information and its associated $\overline{IRQ}$ (if enabled) are cleared by writing a "1" in the Clear $\overline{CTS}$ bit. The $\overline{CTS}$ status bit subsequently follows the $\overline{CTS}$ input when it goes "Low".

The $\overline{CTS}$ input provides character timing for transmitter data when in the external sync mode. Transmission is initiated on the negative transition of the first full positive clock pulse of the transmitter clock (Tx CLK) after the release of $\overline{CTS}$ (see Figure 6).

- **Data Carrier Detect ($\overline{DCD}$)**

The $\overline{DCD}$ input provides a real-time inhibit to the receiver section (the Rx FIFO is not disturbed). A positive $\overline{DCD}$ transition resets and inhibits the receiver section except for the Receive FIFO and the RDRA status bit and its associated $\overline{IRQ}$.

The positive transition of $\overline{DCD}$ is stored within the SSDA to insure that its occurrence will be acknowledged by the system. The stored $\overline{DCD}$ information and its associated $\overline{IRQ}$ (if enabled) are cleared by reading the Status Register and then the Receiver FIFO, or by writing a "1" into the Receiver Reset bit. The $\overline{DCD}$ status bit subsequently follows the $\overline{DCD}$ input when it goes "Low". The $\overline{DCD}$ input provides character synchronization timing for the receiver during the external sync mode of operation. The receiver will be initialized and data will be sampled on the positive transition of the first full Receive Clock cycle after release of $\overline{DCD}$ (see Figure 7).

- **Sync Mach/Data Terminal Ready (SM/$\overline{DTR}$)**

The SM/$\overline{DTR}$ output provides four functions (see Table 4) depending on the state of the PC1 and PC2 control bits. When the Sync Match mode is selected (PC1 = "1", PC2 = "0"), the output provides a one-bit-wide pulse when a sync code is detected. This pulse occurs for each sync code match even if the receiver has already attained synchronization. The SM output is inhibited when PC2 = "1". The $\overline{DTR}$ mode (PC1 = "0") provides an output level corresponding to the complement of PC2 ($\overline{DTR}$ = "0" when PC2 = "1".) (see Table 4.)

- **Transmitter Underflow (TUF)**

The Underflow output indicates the occurrence of a transfer of a "fill character" to the Transmitter Shift Register when the last location (#3) in the Transmit Data FIFO is empty. The Underflow output pulse is approximately a Tx CLK "High" period wide and occurs during the last half of the last bit of the character preceding the "Underflow" (see Figure 4). The Underflow output pulse does not occur when the Tx Sync bit is in the reset state.

- **NOTE FOR USAGE**

If the hold time of $\overline{CS}$ signal and R/$\overline{W}$ signal is within 50~230 ns, there is a case that Transmit Data FIFO is not cleared and TDRA flag is not set when software reset using TxRS (TxRS=1) is executed. Usual program for data transmission will start to send the data as shown in Fig. 11 and Fig. 12.

In this case, the data of the first three bytes are not preset and unexpected data which is remaining in Transmit Data FIFO are sent in the first two bytes.

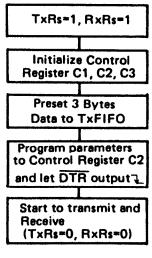


Figure 11 Normal Flow of Starting the Transmission and Reception

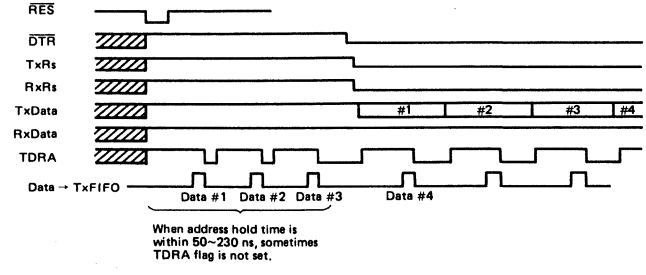


Figure 12 Transmission Start Sequence

In case of SSDA, Address Hold Time should be from 20 to 50 ns or over 230 ns.

- **$\overline{DCD}$ Input in External Synchronization Mode**

In case of receiving data in External Synchronization Mode, Receive data is put off by one bit at times, when $\overline{DCD}$ is drived like ⎍ in RxCLK cycle in which RDA flag is set.

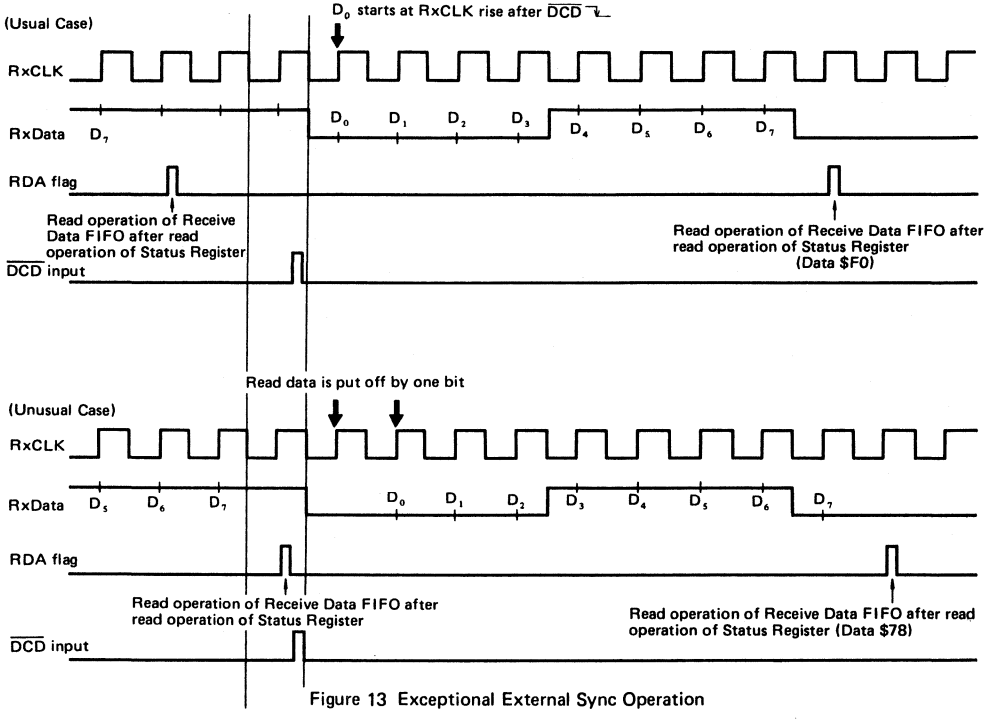


Figure 13 Exceptional External Sync Operation

To avoid this case, use SSDA in the following method.

(1) $\overline{DCD}$ ↓ and RxCLK ↓ should meet the relation shown in Fig. 14.

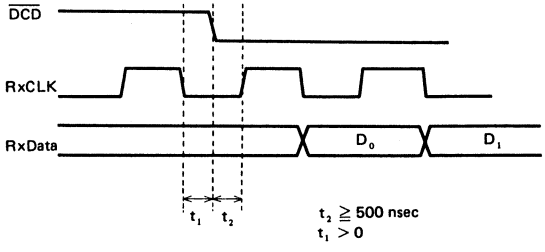


Figure 14 $\overline{DCD}$ Input Timing in External Sync Mode

(2) RxData should be input regarding the second RxCLK rise as D_0 bit, after $\overline{DCD}$ ↓.

HD46508, HD46508-1, HD46508A, HD46508A-1

ADU (Analog Data Acquisition Unit)

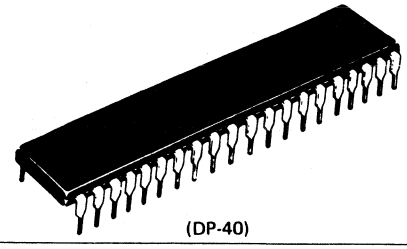
The HD46508 is a monolithic NMOS device with a 10-bit analog-to-digital converter, a programmable voltage comparator, a 16-channel analog multiplexer and HMCS6800 microprocessor family compatible interface.

Each of 16 analog inputs is either converted to a digital data by the analog-to-digital converter or compared with the specified value by the programmable comparator. The analog-to-digital converter uses successive approximation method as the conversion technique. It's intrinsic resolution is 10 bits but it can be 8 bits if the programmer so desires. The programmable voltage comparator compares the input voltage with the value specified by the programmer. The result (greater than, or smaller than) is reflected to the flag in the status register.

The device can expand its capability by controlling the external circuits such as sample holder, pre-amplifier and external multiplexer.

With these features, this device is ideally suited to applications such as process control, machine control and vehicle control.

HD46508P, HD46508P-1, HD46508PA, HD46508PA-1

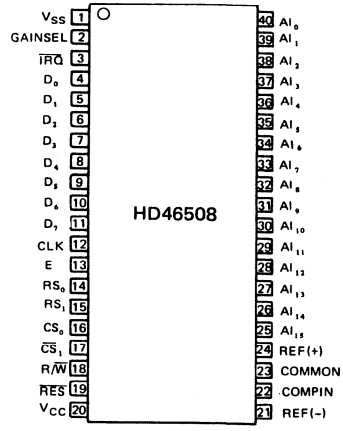


(DP-40)

■ FEATURES

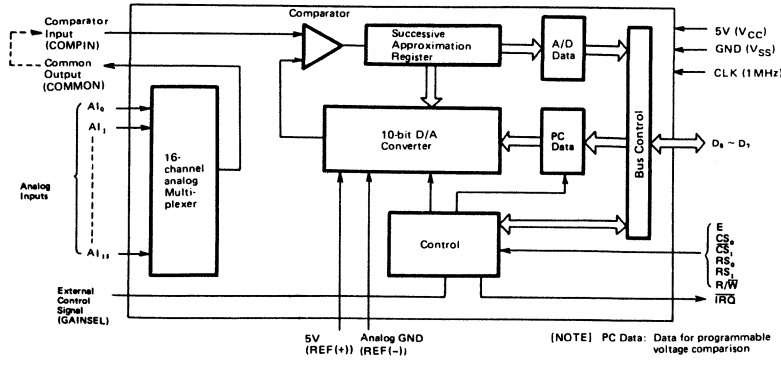
- 16-channel Analog multiplexer
- Programmable A/D Converter resolution (10-bit or 8-bit)
- Programmable Voltage comparison (PC)
- Conversion Time 100μs (A/D), 13μs(PC)
- External Sample and Hold Circuit Control
- Auto Range-switching Control of External Amplifier
- Waiting Function for the Settling Time of External Amplifier
- Interrupt Control (Only for A/D conversion)
- Single +5V Power Supply
- Compatible with HMCS6800 Bus (The connection with other Asynchronous Buses possible)

■ PIN ARRANGEMENT



(Top View)

■ BLOCK DIAGRAM



[NOTE] PC Data: Data for programmable voltage comparison

■ ORDERING INFORMATION

ADU	Bus Timing	Non Linearity*
HD46508A	1 MHz	±1 LSB
HD46508A-1	1.5 MHz	
HD46508	1 MHz	±3 LSB
HD46508-1	1.5 MHz	

* Specification for 10 bit A/D conversion

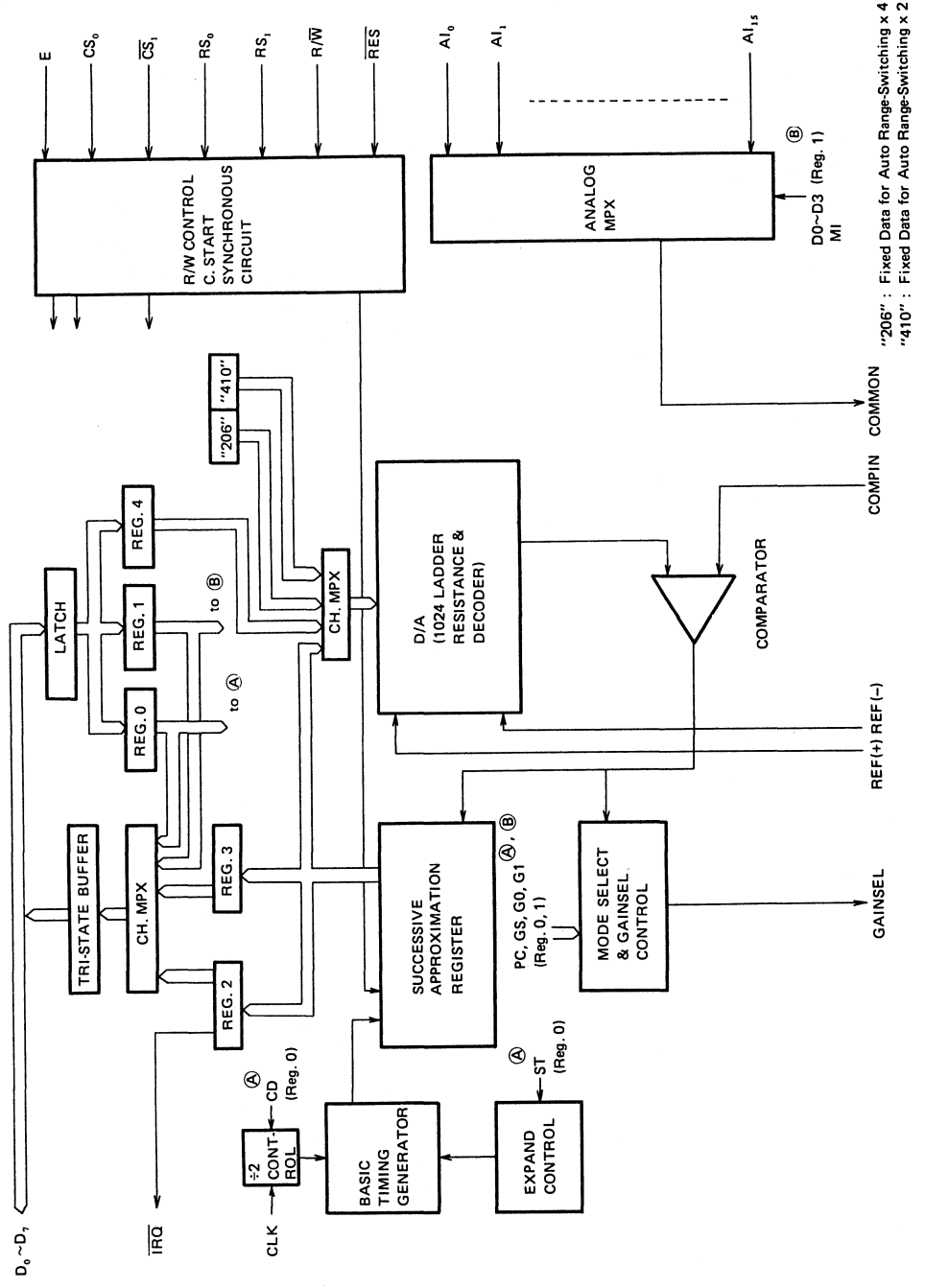


Figure 1 Internal Block Diagram

■ ABSOLUTE MAXIMUM RATINGS

Item	Symbol	Value	Unit
Supply Voltage	V_{CC}*	−0.3 ~ +7.0	V
Input Voltage	V_{in}*	−0.3 ~ +7.0	V
Analog Input Voltage	V_{Ain}*	−0.3 ~ +7.0	V
Operating Temperature	T_{opr}	− 20 ~ + 75	°C
Storage Temperature	T_{stg}	− 55 ~ +150	°C

* With respect to V_{SS} (SYSTEM GND)

[NOTE] Permanent LSI damage may occur if maximum ratings are exceeded. Normal operation should be under recommended operating conditions. If these conditions are exceeded, it could affect reliability of LSI.

■ RECOMMENDED OPERATING CONDITIONS

Item	Symbol	min	typ	max	Unit
Supply Voltage	V_{CC}*	4.75	5.0	5.25	V
Input "High" Voltage	V_{IH}*	2.0	–	V_{CC}	V
Input "Low" Voltage	V_{IL}*	−0.3	–	0.8	V
Analog Input Voltage	V_{Ain}*	0	–	$V_{REF(+)}$	V
Reference Voltage	$V_{REF(+)}$*	–	V_{CC}	V_{CC}+0.25	V
	$V_{REF(-)}$*	−0.1	0	–	
Voltage Center of Ladder	$\frac{V_{REF(+)} + V_{REF(-)}}{2}$*	–	$\frac{V_{CC}}{2}$	$\frac{V_{CC}}{2}$+0.25	V
Operating Temperature	T_{opr}	− 20	25	75	°C

*With respect to V_{SS} (SYSTEM GND)

■ ELECTRICAL CHARACTERISTICS

● DC CHARACTERISTICS <1> (V_{CC} = 5V ± 5%, V_{SS} = 0V, Ta = −20 ~ +75°C, unless otherwise noted.)

Item		Symbol	Test Condition	min	typ	max	Unit
Input "High" Voltage		V_{IH}		2.0	–	V_{CC}	V
Input "Low" Voltage		V_{IL}		−0.3	–	0.8	V
Output "High" Voltage	$D_0 \sim D_7$	V_{OH}	I_{OH} = −205μA	2.4	–	–	V
	GAINSEL		I_{OH} = −200μA	2.4	–	–	
			I_{OH} = −10μA	V_{CC}−1.0	–	–	
Output "Low" Voltage	$D_0 \sim D_7$, GAINSEL	V_{OL}	I_{OL} = 1.6 mA	–	–	0.4	V
	$\overline{IRQ}$		I_{OL} = 3.2 mA	–	–	0.4	
Input Leakage Current	E, CLK, R/$\overline{W}$, $\overline{RES}$, RS_0, RS_1, CS_0, $\overline{CS_1}$	I_{in}	V_{in} = 0 ~ 5.25V	−2.5	–	2.5	μA
Three-State (off state) Input Current	$D_0 \sim D_7$	I_{TSI}	V_{in} = 0.4 ~ 2.4V	−10	–	10	μA
Output Leakage Current	$\overline{IRQ}$	I_{LOH}	V_{OH} = 2.4V	–	–	10	μA
Power Dissipation		P_D		–	–	500	mW
Input Capacitance	$D_0 \sim D_7$	C_{in}	V_{in} = 0V, Ta = 25°C f = 1 MHz	–	–	12.5	pF
	E, CLK, R/$\overline{W}$, $\overline{RES}$, RS_0, RS_1, CS_0, $\overline{CS_1}$			–	–	10.0	pF
Output Capacitance	$\overline{IRQ}$, GAINSEL	C_{out}	V_{in} = 0V, Ta = 25°C f = 1 MHz	–	–	10.0	pF

- **DC CHARACTERISTICS <2> (V_{CC} = 5V ± 5%, V_{SS} = 0V, Ta = -20~+75°C, unless otherwise noted.)**

Item	Test Condition	min	typ	max	Unit
Analog Multiplexer ON Resistance	V_{Ain} = 5.0V, V_{CC} = 4.75V, Ta = 25°C	–	–	1	kΩ
OFF Channel Leakage Current	V_{Ain} = 5.0V, V_{CC} = 4.75V, Ta = 25°C, COMMON = 0V	–	10	100	nA
	V_{Ain} = 0V, Ta = 25°C, V_{CC} = 4.75V, COMMON = 5V	-100	-10	–	nA
Analog Multiplexer Input Capacitance		–	–	7.5	pF
Ladder Resistance (from REF(+) to REF(–))	$V_{REF\ (+)}$ = 5.0V, $V_{REF\ (-)}$ = 0V, Ta = 25°C	10	–	40	kΩ

- **CONVERTER SECTION (Ta = 25°C, V_{CC} = $V_{REF(+)}$ = 5.0V, t_{cycC} = 1μs, unless otherwise noted.)**

1. 10-BIT A/D CONVERSION

Item	HD46508A, HD46508A-1			HD46508, HD46508-1			Unit
	min	typ	max	min	typ	max	
Resolution	–	10	–	–	10	–	bits
Non-linearity Error *	–	±1/2	±1	–	±1	±3	LSB
Zero-Error	–	±1/2	±3/4	–	±1/2	±1	LSB
Full-Scall Error	–	±1/4	±1/2	–	±1/2	±1	LSB
Quantization Error	–	–	±1/2	–	–	±1/2	LSB
Absolute Accuracy *	–	±1	±3/2	–	±2	±4	LSB

2. 8-BIT A/D CONVERSION

Item	HD46508A, HD46508A-1			HD46508, HD46508-1			Unit
	min	typ	max	min	typ	max	
Resolution	–	8	–	–	8	–	bits
Non-linearity Error *	–	±1/8	±1/4	–	±1/4	±3/4	LSB
Zero-Error	–	±1/4	±3/8	–	±3/8	±1/2	LSB
Full-Scall Error	–	±1/4	±3/8	–	±3/8	±1/2	LSB
Quantization Error	–	–	±1/2	–	–	±1/2	LSB
Absolute Accuracy *	–	±5/8	±3/4	–	±3/4	±5/4	LSB

3. PROGRAMMABLE VOLTAGE COMPARISON (PC)

Item	HD46508A, HD46508A-1			HD46508, HD46508-1			Unit
	min	typ	max	min	typ	max	
Resolution	–	8	–	–	8	–	bits
Non-linearity Error *	–	±1/8	±1/4	–	±1/4	±3/4	LSB
Zero-Error	–	±1/4	±3/8	–	±3/8	±1/2	LSB
Full-Scall Error	–	±1/4	±3/8	–	±3/8	±1/2	LSB
Absolute Accuracy *	–	±3/8	±5/8	–	±1/2	±1	LSB

*Temperature Coefficient; 25 ppm of FSR/°C (max)

- **AC CHARACTERISTICS (V_{CC} = 5.0V ± 5%, V_{SS} = 0V, Ta = −20 ~ +75°C, unless otherwise noted.)**

1. CLOCK WAVEFORM

Item	Symbol	Test Conditions	CD* = 0			CD* = 1			Unit
			min	typ	max	min	typ	max	
CLK Cycle Time	t_{cycC}	Fig. 2	1.0	–	10	0.5	–	5	μs
CLK "High" Pulse Width	PW_{CH}		0.45	–	4.5	0.22	–	2.2	μs
CLK "Low" Pulse Width	PW_{CL}		0.40	–	4.0	0.21	–	2.1	μs
Rise and Fall Time of CLK	t_{Cr}, t_{Cf}		–	–	25	–	–	25	ns

* CD : CLK Divider bit

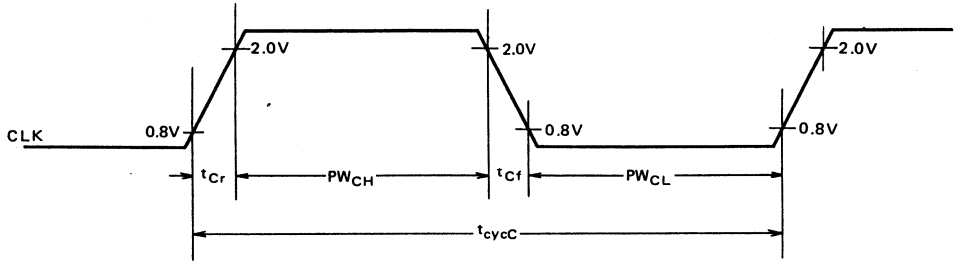


Figure 2 CLK Waveform

2. $\overline{IRQ}$, GAINSEL OUTPUT

Item	Symbol	Test condition	min	typ	max	Unit
$\overline{IRQ}$ Release Time	t_{IR}	Fig. 3	–	–	750	ns
GAINSEL Delay Time	t_{GSD1}	Fig. 4	–	–	750	ns
	t_{GSD2}		–	–	750	ns

t_{GSD1} : TTL Load
t_{GSD2} : CMOS Load

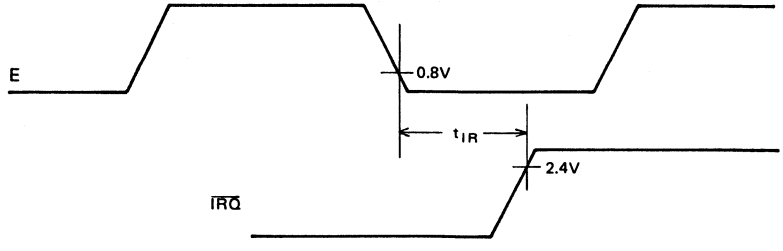
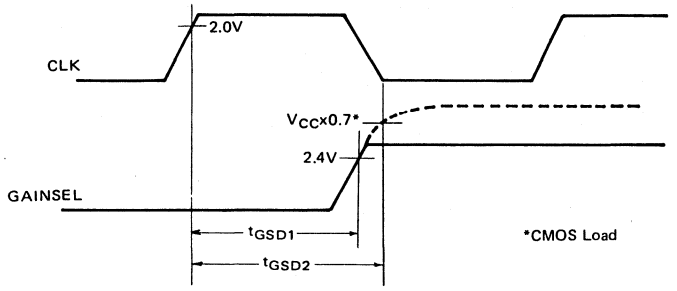


Figure 3 $\overline{IRQ}$ Release Time

(1) Sample & Hold



(2) x2, x4 Auto Range-Switching, Programmable Gain

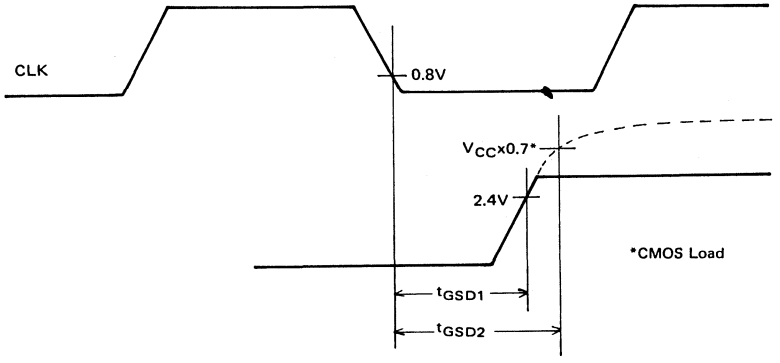


Figure 4 GAINSEL Delay Time

3. BUS TIMING CHARACTERISTICS

READ OPERATION SEQUENCE

Item	Symbol	Test Condition	HD46508 HD46508A			HD46508-1 HD46508A-1			Unit
			min	typ	max	min	typ	max	
Enable Cycle Time	t_{cycE}	Fig. 5	1.0	–	–	0.666	–	–	μs
Enable "High" Pulse Width	PW_{EH}		0.45	–	–	0.28	–	–	μs
Enable "Low" Pulse Width	PW_{EL}		0.40	–	–	0.28	–	–	μs
Rise and Fall Time of Enable	t_{Er}, t_{Ef}		–	–	25	–	–	25	ns
Address Set Up Time	t_{AS}		140	–	–	140	–	–	ns
Data Delay Time	t_{DDR}		–	–	320	–	–	220	ns
Data Access Time	t_{ACC}		–	–	460	–	–	360	ns
Data Hold Time	t_H		10	–	–	10	–	–	ns
Address Hold Time	t_{AH}		10	–	–	10	–	–	ns

WRITE OPERATION SEQUENCE

Item	Symbol	Test Condition	HD46508 HD46508A			HD46508-1 HD46508A-1			Unit
			min	typ	max	min	typ	max	
Enable Cycle Time	t_{cycE}	Fig. 6	1.0	–	–	0.666	–	–	μs
Enable "High" Pulse Width	PW_{EH}		0.45	–	–	0.280	–	–	μs
Enable "Low" Pulse Width	PW_{EL}		0.40	–	–	0.280	–	–	μs
Rise and Fall Time of Enable	t_{Er}, t_{Ef}		–	–	25	–	–	25	ns
Address Set Up Time	t_{AS}		140	–	–	140	–	–	ns
Data Set Up Time	t_{DSW}		195	–	–	80	–	–	ns
Data Hold Time	t_H		10	–	–	10	–	–	ns
Address Hold Time	t_{AH}		10	–	–	10	–	–	ns

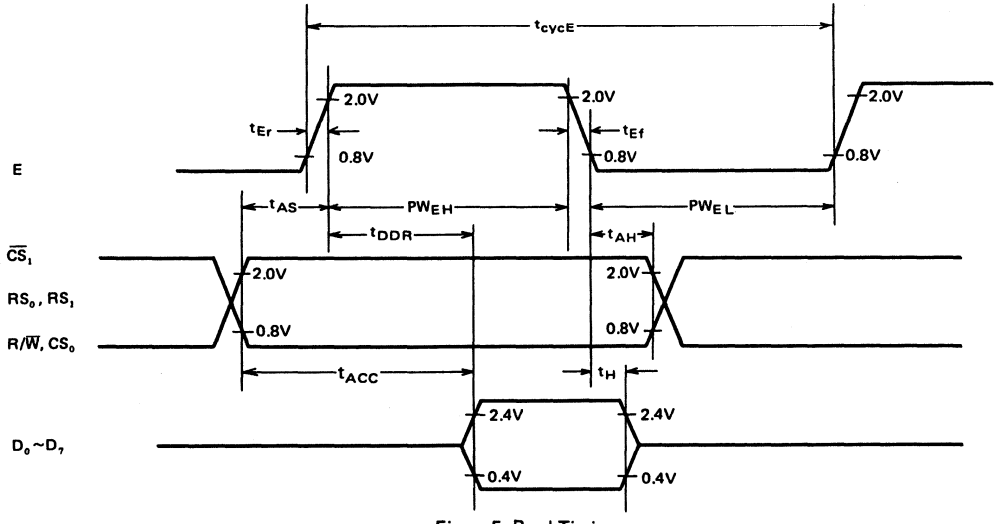


Figure 5 Read Timing

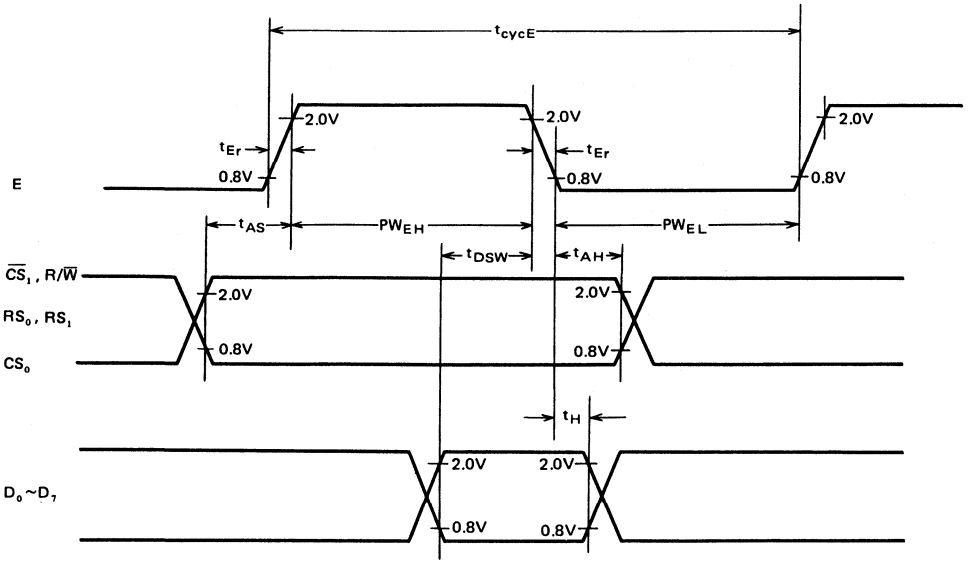


Figure 6 Write Timing

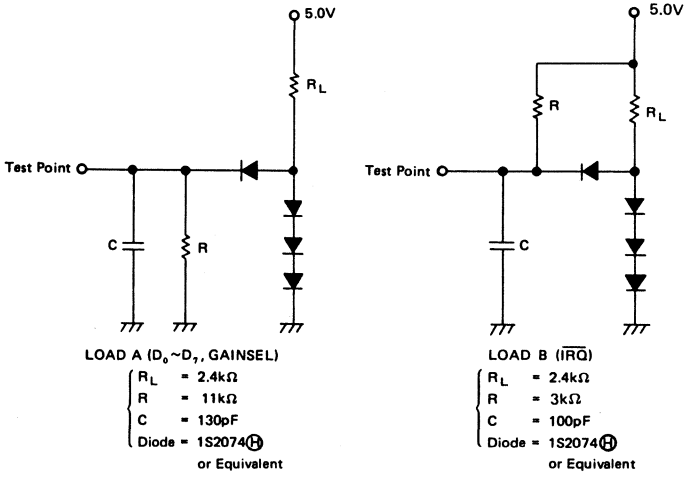


Figure 7 Test Load

■ SIGNAL DESCRIPTION

• Processor Interface

Data Bus (D_0~D_7)

The Bi-directional data lines (D_0~D_7) allow data transfer between the ADU and MPU. Data bus output drivers are three state buffers that remain in the high-impedance state except when MPU performs a ADU read operation.

Enable (E)

The Enable signal (E) is used as strobe signal in MPU R/W operation with the ADU internal registers. This signal is normally derived from the HMCS6800 system clock (ϕ_2).

Chip Select (CS_0, $\overline{CS_1}$)

The Chip Select lines (CS_0, $\overline{CS_1}$) are used to address the ADU. The ADU is selected when CS_0 is at "High" and $\overline{CS_1}$ is at "Low" level.

Read/Write (R/$\overline{W}$)

The R/$\overline{W}$ line controls the direction of data transfer between the ADU and MPU. When R/$\overline{W}$ is at "High" level, data of ADU is transferred to MPU. When R/$\overline{W}$ is at "Low" level, data of MPU is transferred to ADU.

Register Select (RS_0, RS_1)

The Register Select line (RS_0, RS_1) are used to select one of the 4 ADU internal registers. Table 1 shows the relation between (RS_0, RS_1) address and the selected register. The lowest 2 address lines of MPU are usually used for these signals.

Reset ($\overline{RES}$)

This input is used to reset the ADU. An input "Low" level on $\overline{RES}$ line forces the ADU into following status.

1) All the shift-registers in ADU are cleared and the conversion operation is stopped.
2) The GAINSEL output goes down to "Low" level. The $\overline{IRQ}$ output is made "Off" state and the D_0~D_7 are made high impedance state.

Interrupt Request ($\overline{IRQ}$) (Open Drain Output)

This output line is used to inform the A/D conversion end signal to the MPU. This signal becomes active "Low" level when IE bit in the control register 1 is "1" and IRQ bit in the control register 2 goes "1" at the end of conversion. And this signal returns to "High" right after The MPU reads the A/D Data Register (R3). Programmable voltage comparison does not affect this signal.

• Analog Data Interface

Analog Input (AI_0~AI_{15})

The Input Analog Data to be measured is applied to these Analog Input (AI_0~AI_{15}). These are multiplexed by internal 16 channel multiplexer and output to COMMOM pin. A particular input channel is selected when the multiplexer channel address is programmed into the control Register 1 (R1).

Multiplexer Common Output (COMMON)

This signal is the output of the 16 channel analog multiplexer, and may be connected to the input of pre-amplifier or sample/hold circuit according to user's purposes. When no external circuit needed, this output should be connected to the COMPIN input.

Comparator Input (COMPIN)

This is a high impedance input line that is used to transmit selected analog data to comparator. The COMMON line is usually connected to this input. When external Pre-amplifier or Sample/hold circuit is used, output of these circuits may be connected to this input.

Reference Voltage (+) (REF (+))

This line is used to apply the standard voltage to the internal ladder resistors.

Reference Voltage (–) (REF (–))

This line is connected to the analog ground.

• ADU Control

Conversion Clock (CLK)

The CLK is a standard clock input signals which defines internal timing for A/D conversion and PC operation.

Gain Select (GAINSEL) (CMOS Compatible Output)

This output is used to control the external circuit. The function of this signal is programmable and it is specified by (G1, G0) bits in Control Register 0. By using this output, user can control the auto-range-switching of external pre-amplifier, also control external sample & hold circuit, etc. as well.

[NOTE] This LSI is different from other HMCS6800 family LSIs in following function
- $\overline{RES}$ doesn't affect IE bit of R0

■ FUNCTION OF INTERNAL REGISTERS

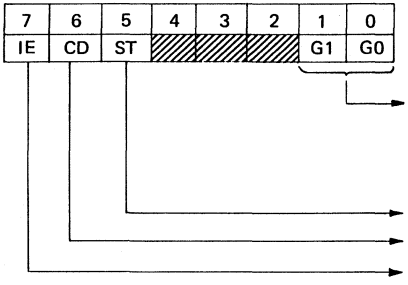
• Structure

Table 1 Internal Registers of the ADU

$\overline{CS_1}$	CS_0	RS_1	RS_0	Reg. #	Register Name	Read	Write	Data Bit 7	6	5	4	3	2	1	0
0	1	0	0	R0	Control Register 0	O	O	IE	CD	ST				G1	G0
0	1	0	1	R1	Control Register 1	O	O	SC	GS	PC	MI	D3	D2	D1	D0
0	1	1	0	R2	Status & A/D Data Register (H)	O	x	IRQ	BSY	PCO		OV	DW	C9	C8
0	1	1	1	R3	A/D Data Register (L)	O	x	C7	C6	C5	C4	C3	C2	C1	C0
0	1	1	1	R4	PC Data Register	x	O	B7	B6	B5	B4	B3	B2	B1	B0

(Note) O --- YES
x --- NO

Control Register 0 (R0)

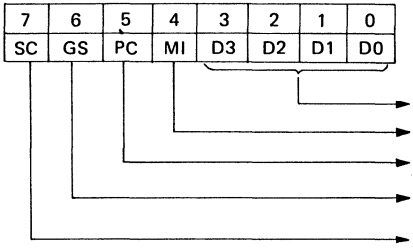


	"1"	"0"
Mode Select	See Table 2	
Not Used		
Not Used		
Not Used		
Settling Time	Available	Not Available
CLK Divider	CLK/2	CLK
Interrupt Enable*	Enable IRQ	Mask IRQ

Figure 8 Control Register 0

*$\overline{RES}$ doesn't affect IE bit.

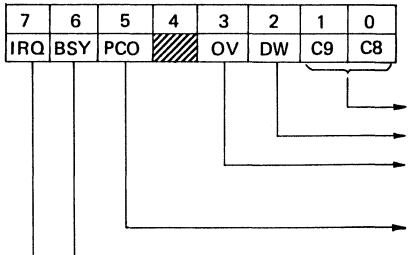
Control Register 1 (R1)



	"1"	"0"
MPX Channel Address	See Table 3	
MPX Inhibit	Inhibited	Not Inhibited
Prog. Comparator Select	Prog. Comparator mode	A/D Converter mode
GAINSEL Enable	GAINSEL Enable	GAINSEL Disable
Short-cycle Conversion	8-bit Length	10-bit Length

Figure 9 Control Register 1

Status & A/D Data Register (H)



	"1"	"0"
Upper bit (10 bit data)		
Data Weight	See Table 4.	
Data Over Scale flag	Data is over scale	Within the scale
Not Used		
Programmable Comparator Output	$V_{Ain} > V_p$	$V_{Ain} < V_p$
Busy flag	Under Conversion	Conversion Completed
IRQ flag	Requested	Not Requested

V_{Ain} : Unknown Input Voltage
V_p : Programmed Voltage by R4
C9, C8 bits are cleared when 8 bit A/D conversion is performed.

Figure 10 Status & A/D Data Register (H)

A/D Data Register (L)

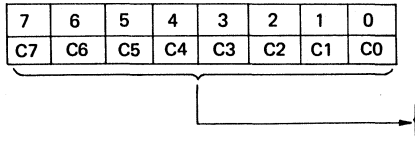


Figure 11 A/D Data Register (L)

PC Data Register

7	6	5	4	3	2	1	0
B7	B6	B5	B4	B3	B2	B1	B0

8 bit Data for Programmable Voltage Comparison

Figure 12 PC Data Register

- **Description for the Internal Registers**

Control Register 0 (R0)

This Register is a 5-bit read/write register that is used to specify Interrupt Enable (IE), CLK Divider (CD), Settling Time (ST) and Mode Select (G0, G1). This Register should be written before writing R1.

IE bit: (Interrupt Enable)
- IE = "1", Interrupt is requested through the $\overline{IRQ}$ output.
- IE = "0", Interrupt is masked.

CD bit: (Clock Divider)
- CD = "1", CLK ÷ 2 is used as internal clock.
- CD = "0", CLK is used directly.

ST bit: (Settling Time)
- ST = "1", First comparison is executed after 1 expanded cycle in order to compensate external amplifiers settling delay.
- ST = "0", Cycle is not delayed.

G0, G1 bit; (Mode select)
These bits are used to specify the function of GAINSEL signal when GS bit is "1".

Table 2 Function of G0, G1

G1	G0	Mode Select
0	0	Sample & Hold
0	1	Auto Range-Switching x 2
1	0	Auto Range-Switching x 4
1	1	Programmable Gain Control

Control Register 1 (R1)

This register is an 8-bit read/write register that is used to store the command for A/D conversion mode and programmable comparison mode. This register includes MPX channel address (D_0~D_3), MPX inhibit (MI), programmable comparator select (PC), GAINSEL enable (GS) and short-cycle conversion (SC) bits. When this register (R1) is programmed, each conversion mode starts.

SC bit (Short-cycle)
- SC = "1", Short-cycle conversion (8 bit length)
- SC = "0", Normal conversion (10 bit length)

GS bit (GAINSEL Enable)
- GS = "1", GAINSEL signal is enabled. The function of GAINSEL is specified by (G0, G1) bits.
- GS = "0", GAINSEL signal is disabled. ("Low" level)

PC bit (Program comparator)
- PC = "1", Programmable voltage comparator mode
- PC = "0", A/D conversion mode

MI bit (MPX Inhibit)
- MI = "1", Internal MPX channel is inhibited in order to use external MPX channel.
- MI = "0", Internal MPX channel is used.

D0~D3 (MPX channel)
These bits are used to select the particular MPX channel.

Table 3 MPX Channel Addressing

Channel #1	D3	D2	D1	D0	Analog Input
0	0	0	0	0	AI_0
1	0	0	0	1	AI_1
2	0	0	1	0	AI_2
3	0	0	1	1	AI_3
4	0	1	0	0	AI_4
5	0	1	0	1	AI_5
6	0	1	1	0	AI_6
7	0	1	1	1	AI_7
8	1	0	0	0	AI_8
9	1	0	0	1	AI_9
10	1	0	1	0	AI_{10}
11	1	0	1	1	AI_{11}
12	1	1	0	0	AI_{12}
13	1	1	0	1	AI_{13}
14	1	1	1	0	AI_{14}
15	1	1	1	1	AI_{15}

Table 4 Function Select

PC	SC	Function	GS	(G0, G1)
0	0	10 bit AD CONV.	0	DISABLE
			1	ENABLE*
	1	8 bit AD CONV.	0	DISABLE
			1	ENABLE*
1	x	PROG. COMP (8 bit)	x	DISABLE

x = Do not care
* = See Table 6
[NOTE] CD bit and ST bit are effective in every case.

Status & A/D Data Register (H) (R2)

This register is a 7-bit read only register that is used to store the upper 2-bit data (C8, C9), data weight (DW), data overscale (OV), programmable comparator output (PCO), busy (BSY) and interrupt request(IRQ).

(C8, C9) (Upper bit data) : These bits store upper 2-bit data measured by 10 bit length conversion.

DW bit (Data weight) : This bit indicates data weight when Auto range-switching mode is selected. This bit is set or reset when the conversion has completed. The conditions are shown in following Table.
In this mode GAINSEL output also goes "High" or "Low" on the same condition shown in Table 5.
Other status of DW bit is shown in Table 6.

OV bit (Over scale) : This bit is set when analog data is greater than or equal to reference Voltage ($V_{REF(+)}$).

PCO bit (Programmable comparator Output) : This bit indicates the result of programmable voltage comparison.
"1" → PCO $V_{Ain} > Vp$
"0" → PCO $V_{Ain} < Vp$
V_{Ain} : Analog Input Voltage to be compared
Vp : Programmed Voltage (R4)

BSY bit (Busy) : This bit indicates that the ADU is now under conversion.

IRQ bit (Interrupt Request) : This bit is set when the A/D conversion has completed and cleared by reading the R3.

A/D Data Register (L) (R3)

This register is an 8-bit read-only register that is used to store the lower 8 bits data of 10-bit conversion or full 8 bits data of the 8-bit conversion.

PC Data Register (R4)

This register is an 8-bit write-only register prepared for Programmable Voltage comparison. Stored data is converted to digital voltage, and compared with analog input to be measured. The result of comparison is set into PCO bit.

Table 5 Data Weight (DW) Set or Reset Condition

Mode \ Condition	Set ("1")	Reset ("0")
Auto Range-Switching (x2)	$V_{Ain} < \frac{410}{1024} \cdot V_{REF(+)}$	$V_{Ain} > \frac{410}{1024} \cdot V_{REF(+)}$
Auto Range-Switching (x4)	$V_{Ain} < \frac{206}{1024} \cdot V_{REF(+)}$	$V_{Ain} > \frac{206}{1024} \cdot V_{REF(+)}$

V_{Ain} : Analog Input Voltage to be measured
$V_{REF(+)}$: Voltage Applied to REF(+)

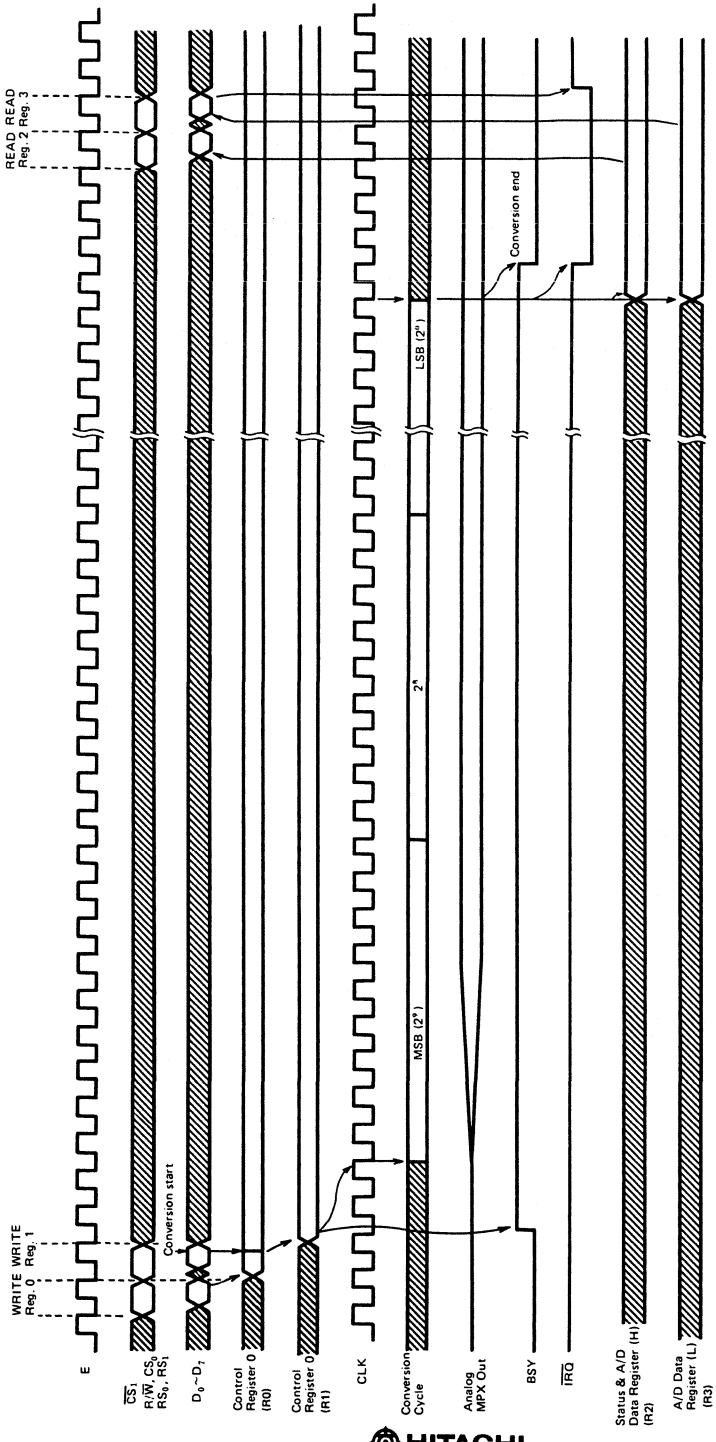
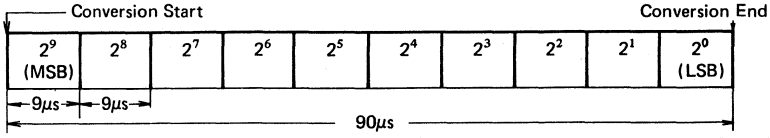


Figure 13 A/D Conversion Timing Chart (Basic Sequence)

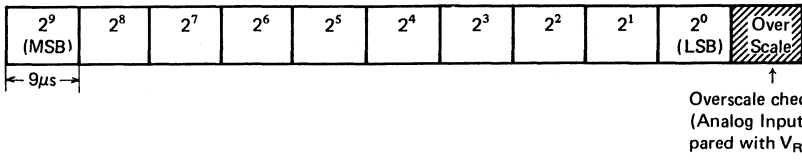
- **A/D Conversion and PC sequence** (t_{cyc}=1μs)

10 bits A/D Conversion

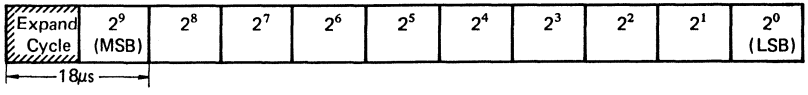
1) Basic Sequence
(SC = "0", ST = "0", GS = "0")



2) Basic Sequence (When overscale is detected)

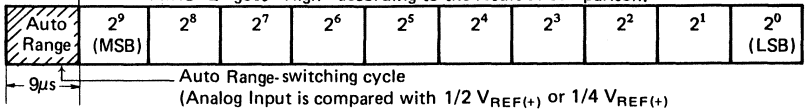


3) Expanded Sequence
(SC = "0", ST = "1", GS = "0")



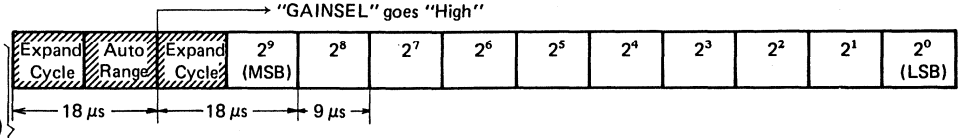
MSB cycle is expanded to compensate external amplifier's settling delay.

4) Auto Range-Switching Control Sequence
(SC = "0", ST = "0", GS = "1", G0 = "0", G1 = "1") or (G0 = "1", G1 = "0")

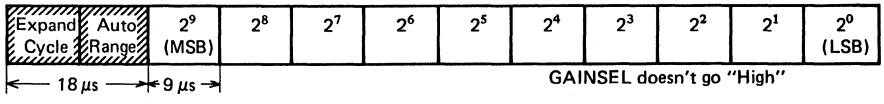


5) Auto Range-Switching & Expansion Control Sequence
(SC = "0", ST = "1", GS = "1", G0 = "0", G1 = "1") or (G0 = "1", G1 = "0")

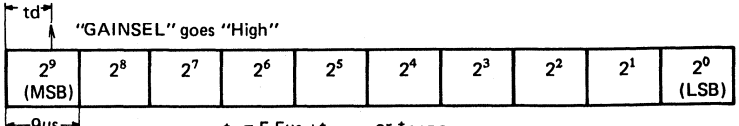
a) Analog Input < 1/2 $V_{REF(+)}$ or 1/4 $V_{REF(+)}$



b) Analog Input > 1/2$V_{REF(+)}$ or 1/4$V_{REF(+)}$

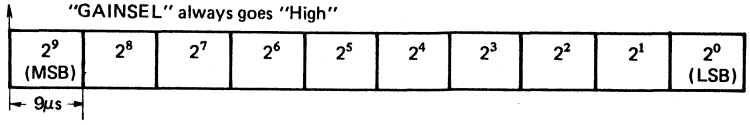


6) Sample & Hold Control Sequence
(SC = "0", ST = "0", GS = "1", G0 = "0", G1 = "0")



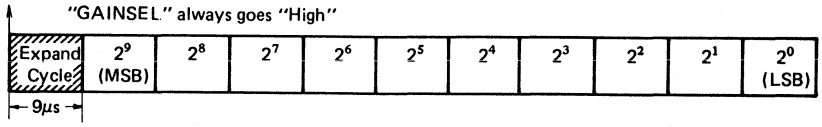
7) Programmable Gain Control Sequence

(SC = "0", ST = "0", GS = "1", G0 = "1", G1 = "1")



8) Programmable Gain & Expansion Control Sequence

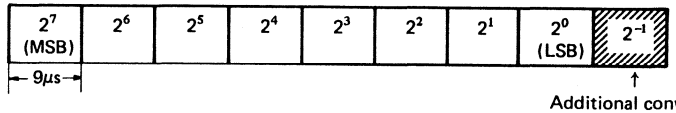
(SC = "0", ST = "1", GS = "1", G0 = "1", G1 = "1")



8 Bit A/D Conversion

1) Basic Sequence

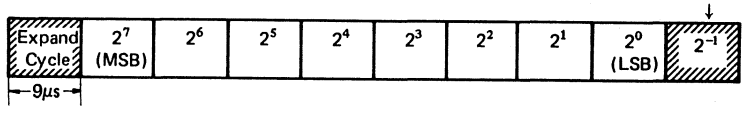
(SC = "1", ST = "0", GS = "0")



Additional conversion cycle for rounding the LSB – 1 Bit.

2) Expanded Sequence

(SC = "1", ST = "1", GS = "0")



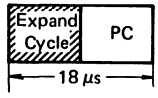
Programmable Voltage Comparison

1) Basic Sequence

(PC = "1", ST = "0")

2) Expanded Sequence

(PC = "1", ST = "1")



■ HOW TO USE THE ADU

● Functions of GAINSEL

The ADU is equipped with programmable GAINSEL output signal. By using GAINSEL output and external circuit, the ADU is able to implement following control.

1) Auto Range-Switching (Auto Gain) Control
2) Programmable Gain control
3) Sample & Hold control

GAINSEL output is controlled by Mode Select bit (G0, G1) when GAINSEL enable bit (GS) is "1".

Table 6 GAINSEL Control

GS	G1	G0	GAINSEL	Control Mode	DW
0	x	x	"Low"	Normal Use (GAINSEL is not used)	0
1	0	0	"High"	Sample & Hold control	0
1	0	1	*	Auto Range Switching x 2 control	**
1	1	0	*	Auto Range Switching x 4 control	**
1	1	1	"High"	Programmable Gain control	1

* GAINSEL goes "High" or "Low" according to the condition shown in Table 5.
** See, Table 5.

How to Control External Circuit

(1) Sample & Hold Control (G1=0, G0=0)

An example of Sample & Hold circuit is shown in Fig. 14. When ADU is set in Sample & Hold Control Mode, GAINSEL becomes "High" level on conversion and controls the data holding.

(2) Automatic Range Switching Control (G1=0, G0=1 or G1=1, G0=0)

The GAINSEL signal controls the external amplifier which can change the ratio of voltage amplification. (GAIN: 1 → 2 times or 1 → 4 times). Fig. 15 shows Automatic Range Switching Control. In this case, when the input voltage is lower than 206/1024 $V_{REF(+)}$, GAINSEL becomes "High" level. This makes the GAIN of the amplifier change from 1 to 4 times, and 4 times value of the input voltage is A/D converted. Using this function even if an input signal is small, it is possible to execute A/D conversion in nearly full scale. In this mode, when GAINSEL signal becomes "High", DW bit becomes "1" to show the range switching is in a progress.

(3) Programmable GAIN Control (G1=1, G0=1)

The GAINSEL signal is used for controlling the external amplifier of any GAIN which is fit to the system.

In this mode, GAINSEL always becomes "High" at the beginning of A/D conversion, so the change of range is controlled by GS bit. Converted data need to be corrected in software in accordance with GAIN of the amplifier.

This mode is effective in the case of converting very small input voltage.

(Note) Refer to "ADU Function Sequence" (A/D Conversion and PC Sequence) for the timing in which GAINSEL signal becomes "High". GAINSEL signal becomes "Low" in accordance with "1" → "0" change of BSY bit. Refer to Fig. 13.

x1 Sample & Hold

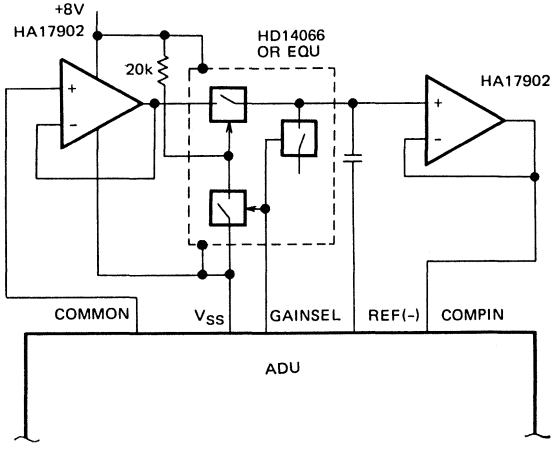


Figure 14 Sample & Hold Circuit

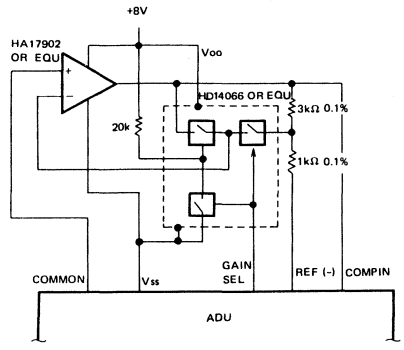
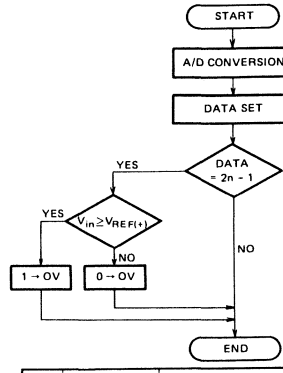


Figure 15 Pre-amplifier Circuit
(x1, x4 Auto-Range Switching)

- **Overscale Check**

ADU is equipped with hardware overscale detection function. The overscale detection is performed automatically when the result of A/D conversion is 2^n-1 (all bits = 1). When analog input V_{Ain} is higher than $V_{REF(+)}$, overscale bit (OV) is set to "1". The definition of the overscale is illustrated in Fig. 17. And the flow of overscale check is shown in Fig. 16.



OV	DATA	NOTE
0	11 1	NOT OVERSCALE
1	11 1	OVERSCALE

Figure 16 Overscale Check Flow

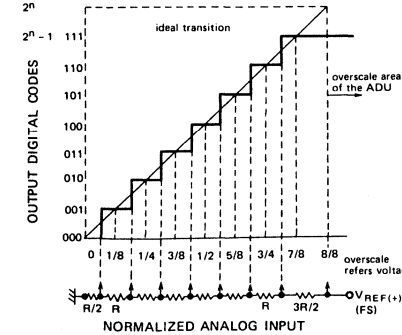


Figure 17 Definition ADU's Overscale

- **Usage of the PC**

The ADU has a programmable threshold voltage comparator (PC) function. The threshold voltage is pre-setable from 0V to 5V range with 8 bit resolution. The comparator's output is stored into PCO bit at the end of comarison.

The programmable voltage comparison time is so short that the interrupt is not requested at this mode. The end of comparison needs to be confirmed by reading the 1→0 transition of the BSY bit in R2.

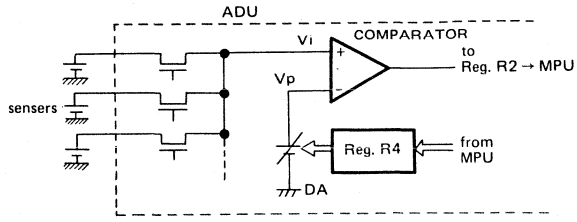


Figure 18 Function Diagram of the PC

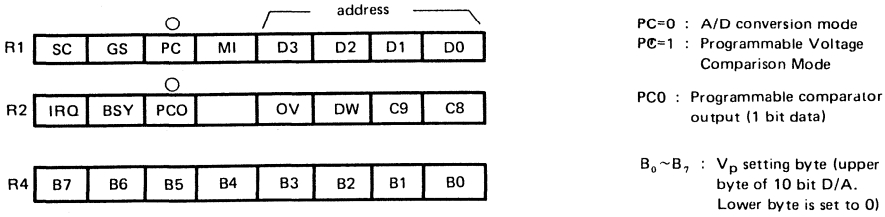


Figure 19 Registers of the PC Mode

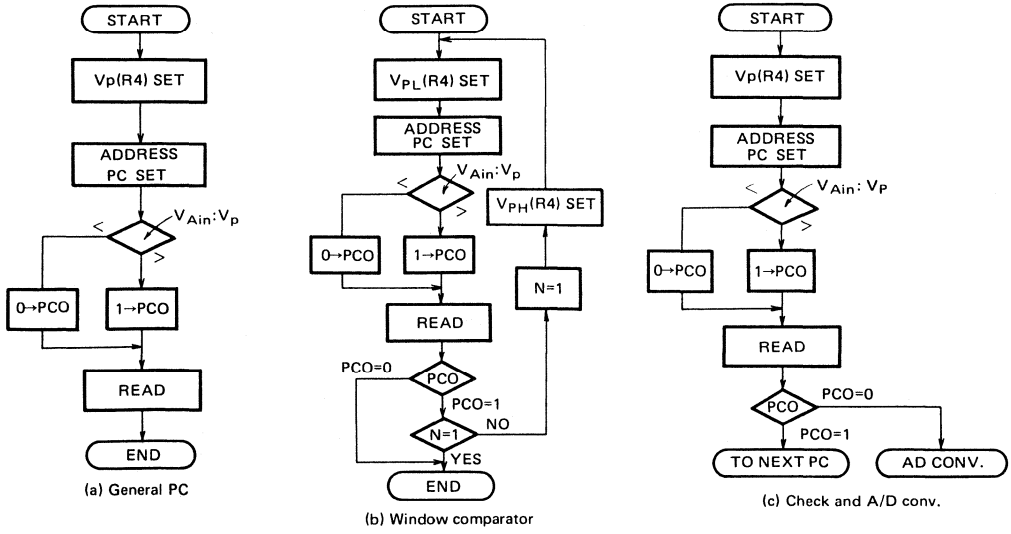
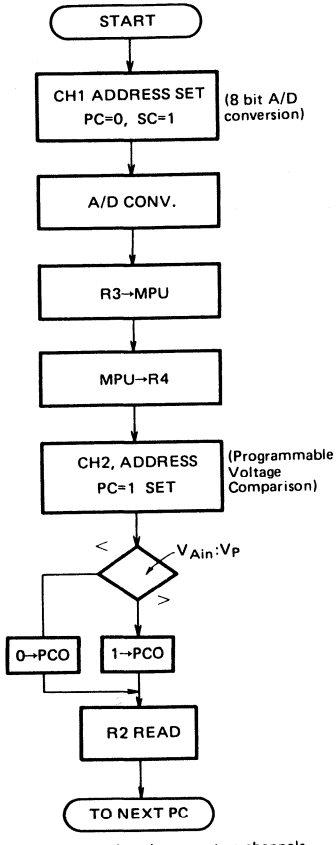


Figure 20 PC Application Flow Chart Examples



(d) Voltage Comparison between two channels.

Figure 20 PC Application Flow Chart Examples (continued)

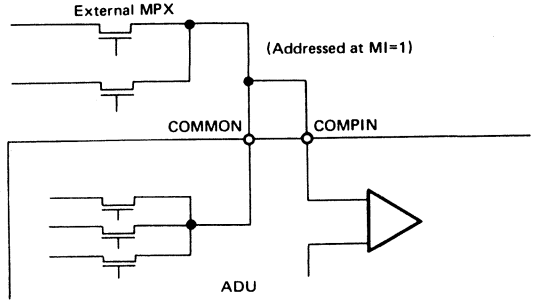
- **How to use MI bit**

MI bit (R1) functions as follows.

MI = 1: Internal MPX channel is inhibited in order to use attached external MPX channel.

MI = 0: Internal MPX channel is enabled.

MI bit used to select either of External MPX and Internal MPX. External MPX is connected as follows.



[NOTE] When external MPX is used as the way figure 20, 1 dammy AD conversion or PC at MI=1 should be performed.

Figure 21 How to use External MPX

■ EXAMPLE OF APPLIED CIRCUIT OF THE ADU

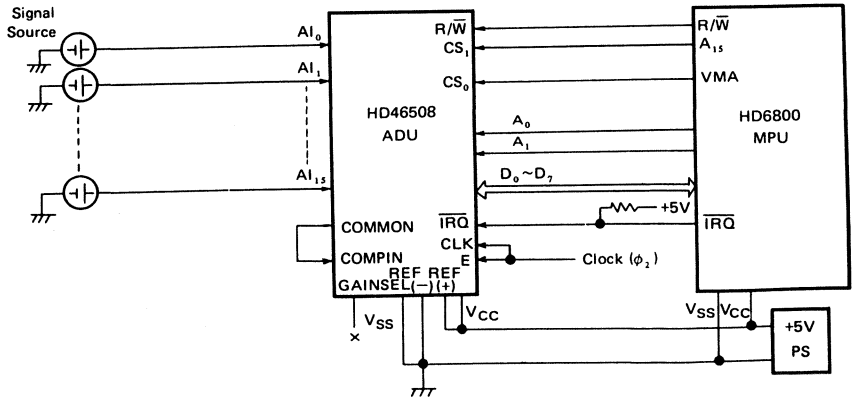


Figure 22 Single ADU System

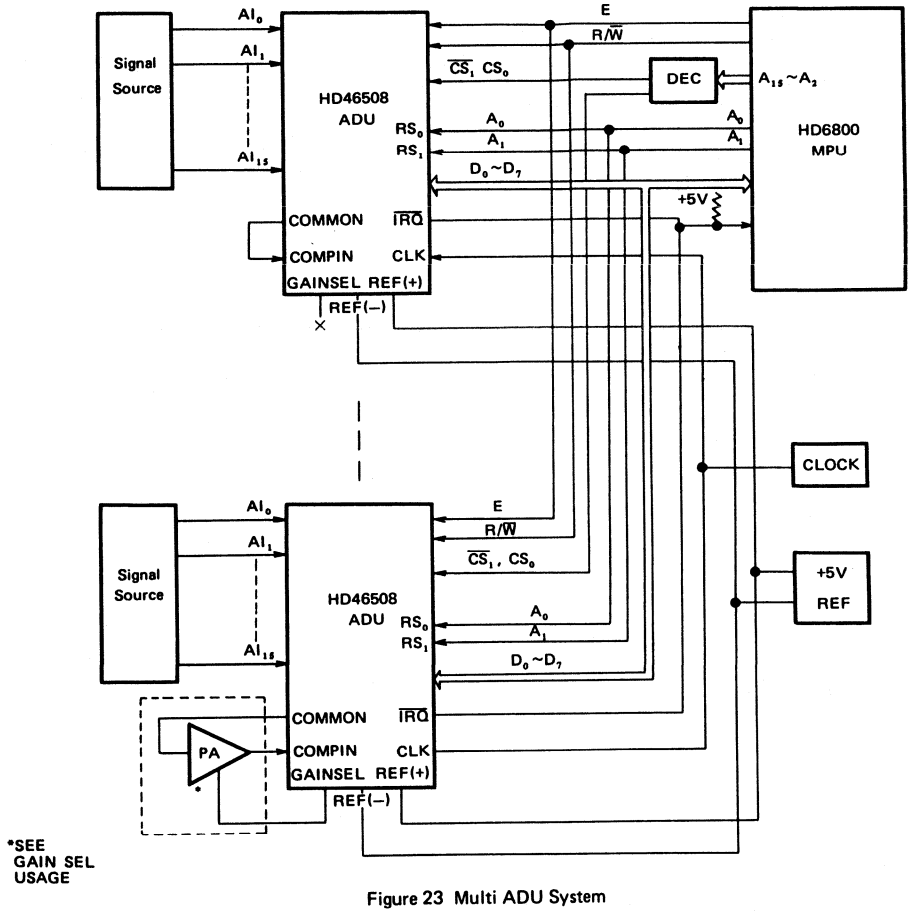


Figure 23 Multi ADU System

■ DEFINITIONS OF ACCURACY

Definitions of accuracy applyed to HD46508 are as follows.

(1) Resolution . . . The number of output binary digit.

(2) Offset Error . . . The difference between actual input voltage and ideal input voltage for the first transition. (when digital output code is changed from 000 . . . 00 to 000 . . . 01.)

(3) Full Scale Error . . . The difference between actual input voltage and ideal input voltage for the final transition. (when digital output code is changed from 111 . . . 10 to 111 . . . 11.)

(4) Quantizing Error . . . Error equipped in A/D converter inherently. Always ±½ LSB is applied.

(5) Non-linearity Error . . . The maximum deviation of the actual transfer line from an ideal straight line. This error doesn't include Quantizing Error, Offset, or Full Scale Errors.

(6) Absolute Accuracy . . . The deviation of the digital output code from an analog input voltage. Absolute accuracy includes all of (2), (3), (4), (5).

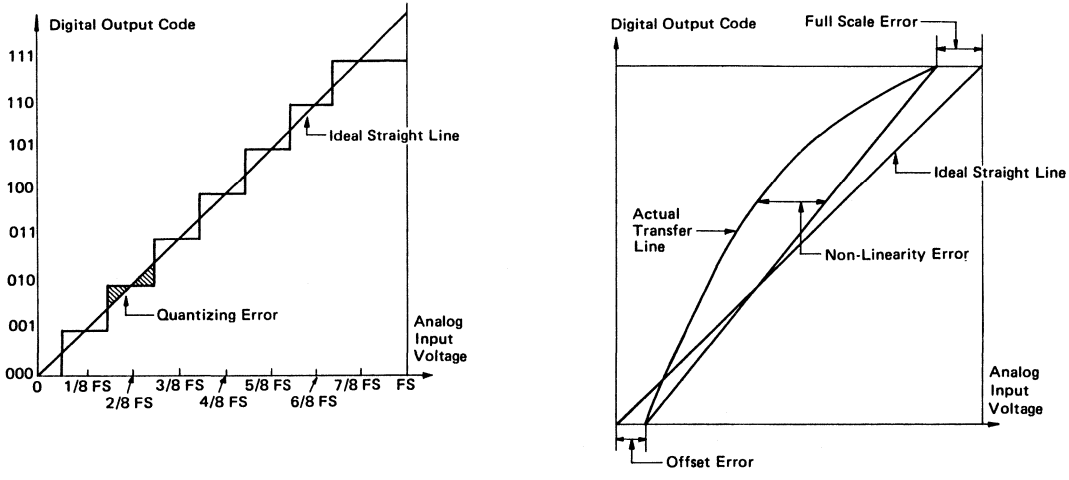


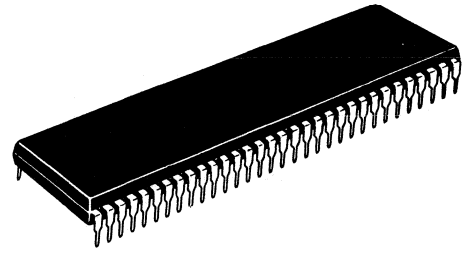
Figure 24 Definition of Accuracy

HD63084
DIPP (Document Image Pre-Processor)

—PRELIMINARY—

The HD63084 (DIPP) is an document image processor used as a peripheral of a microcomputer. It reads analog image signals that have been photoelectrically converted by CCD line sensors or other optical devices, corrects the shading distortion of the signals, then converts the signals to digital form.

HD63084

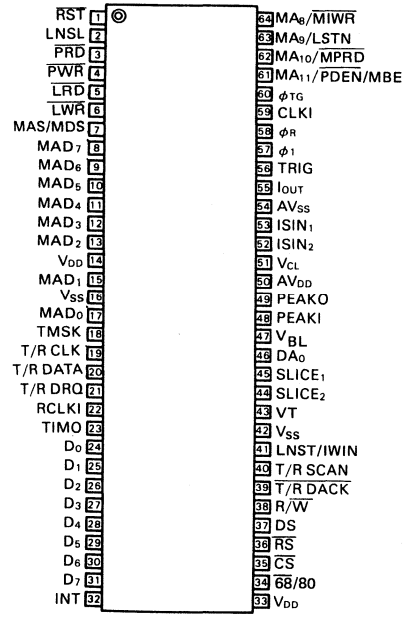


(DP-64S)

■ FEATURES

- High speed reading of image signals
  - 5M pixel/sec (at input clock frequency of 10 MHz)
- Highly accurate processing of image signals
  - Peak level of image signals, 0.1V ~ 2.0V
  - Built-in 8-bit peak value detection circuit
  - Built-in 7-bit successive approximation pixel A/D and D/A converter
  - Built-in 4-bit flash-type A/D converter
- Various output modes
  - Binary data output mode
  - Dithered data output mode (Programmable dithered pattern of 16 pixel x 16 pixel)
  - 4-bit coded data mode (16 gradations)
- Automatic judgement of horizontal and vertical resolutions
- Interfaceable with either Motorola type or Intel type MPU
- Programmable magnification and reduction rates
  - Reduction of read image signal
    : 0.125 ~ 1 times (about 1000 gradations)
  - Magnification of image signal to be recorded
    : 1 ~ 8 times (about 1000 gradations)
- Implements the following functions on a single chip
  - (Built-in) sample and hold circuit
  - (Built-in) shading distortion correction RAM
  - (Built-in) sensor interface
  - Parallel to serial conversion of the image signal to be recorded.
- 2.5 μm CMOS process technology
- Single 5V supply

■ PIN ARRANGEMENT



(Top View)

HD63085

DICEP (Document Image Compression and Expansion Processor)

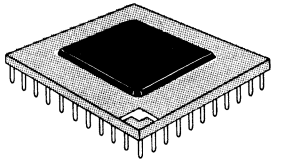
— PRELIMINARY —

The HD63085 (DICEP) is a LSI that compresses (or encodes) and expands (or decodes) the digital data representing a document image. The DICEP uses Modified Huffman (MH) coding scheme, Modified Relative Element Address Designate (MR) coding scheme and Modified MR (M^2R) coding scheme which are compatible with the CCITT (Comité Consultatif International Télégraphique et Téléphonique) recommendations for Group 3 and Group 4 facsimile apparatus.

As the DICEP stores coding and decoding algorithms in the microprogram ROM as firmare, a single MPU command allows this LSI to encode or decode a scan line of digital data.

This LSI is suitable for Group 3 and Group 4 facsimile apparatus, file serves, intelligent copies, terminals, word processors, laser beam printers and other office automation systems.

HD63085Y

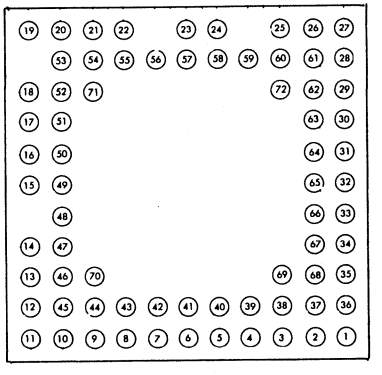


(PC-72)

■ FEATURES

- Various coding schemes ..... MH, MR, M^2R and Run Length coding
- Compatible with the CCITT recommendation for Group 3 and Group 4 facsimile apparatus
- Interfaceable with either Motorola type MPU or Intel type MPU
- DMA capability through the document image bus
  4M Byte/sec (at input clock frequency of 32 MHz)
- A variety of programmable parameters
  - The length of a scan line : 0 ~ 65535 bits
  - The number of RTC or EOL code words: 0 ~ 65535
  - Programmable starting address
  - Coding and decoding of a desired part of a document
- Selectable document image bus size
  Document image bus : 8 bits or 16 bits
  System bus : 8 bits
- 64K Bytes of document image memory is available independently of the MPU
- 2 μm CMOS process technology
- Single 5V supply

● PIN ARRANGEMENT



Pin No.	Function	Pin No.	Function	Pin No.	Function
1	MA/MD_0	25	$\overline{68}/80$	49	$\overline{DMA}$
2	MA/MD_1	26	NC	50	V_{SS}
3	MA/MD_2	27	$\overline{RESET}$	51	low
4	MA/MD_3	28	V_{DD}	52	$\overline{DACK0}$
5	MA/MD_4	29	D_0	53	$\overline{DRQT1}$
6	MA/MD_{10}	30	D_3	54	NC
7	MA/MD_{12}	31	D_4	55	$\overline{CS}$
8	MA/MD_{14}	32	D_5	56	DRQT0
9	V_{SS}	33	V_{DD}	57	BRQT
10	TESTO	34	READY	58	A2
11	4CLK	35	MAS	59	A0
12	TDATA	36	MA/MD1	60	IRQT
13	VDS	37	MA/MD3	61	V_{DD}
14	$\overline{MW}$	38	MA/MD5	62	D_1
15	SET	39	MA/MD7	63	D_2
16	V_{SS}	40	MA/MD9	64	D_4
17	$\overline{IOR}$	41	MA/MD11	65	D_7
18	DTC	42	MA/MD13	66	V_{DD}
19	R/$\overline{W}$	43	MA/MD15	67	NC
20	AGE	44	V_{SS}	68	$\overline{MAEN}$
21	$\overline{DS}$	45	TEST1	69	NC
22	$\overline{DACK1}$	46	$\overline{LDS}$	70	NC
23	$\overline{BACK}$	47	$\overline{MDEN}$	71	NC
24	A1	48	$\overline{MR}$	72	NC

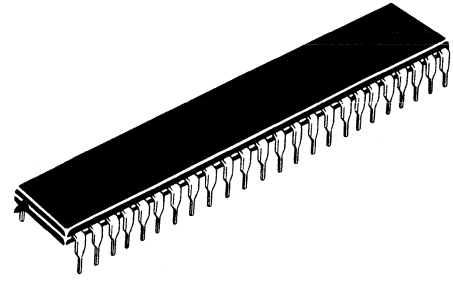
HD68230

PI/T (Parallel Interface Timer)

—PRELIMINARY—

The HD68230 Parallel Interface/Timer provides versatile double buffered parallel interfaces and an operating system oriented timer to HD68000 systems. The parallel interfaces operate in unidirectional or bidirectional modes, either 8 or 16 bits wide. In the unidirectional modes, an associated data direction register determines whether the port pins are inputs or outputs. In the bidirectional modes the data direction registers are ignored and the direction is determined dynamically by the state of four handshake pins. These programmable handshake pins provide an interface flexible enough for connection to a wide variety of low, medium, or high speed peripherals or other computer systems. The PI/T ports allow use of vectored or autovectored interrupts, and also provide a DMA Request pin for connection to the HD68450 Direct Memory Access Controller or a similar circuit. The PI/T timer contains a 24-bit wide counter and a 5-bit prescaler. The timer may be clocked by the system clock (PI/T CLK pin) or by an external clock (TIN pin), and a 5-bit prescaler can be used. It can generate periodic interrupts, a square wave, or a single interrupt after a programmed time period. Also it can be used for elapsed time measurement or as a device watchdog.

HD68230P8 HD68230P10

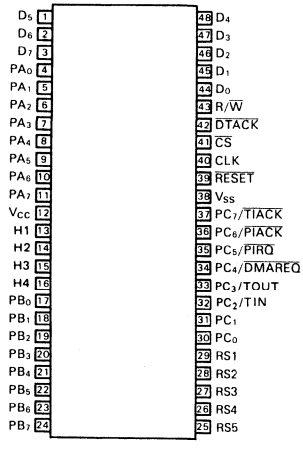


(DP-48)

FEATURES

- HMCS68000 Bus Compatible
- Port Modes Include:
  - Bit I/O
  - Unidirectional 8-Bit and 16-Bit
  - Bidirectional 8-Bit and 16-Bit
- Selectable Handshaking Options
- 24-Bit Programmable Timer
- Software Programmable Timer Modes
- Contains Interrupt Vector Generation Logic
- Separate Port and Timer Interrupt Service Requests
- Registers are Read/Write and Directly Addressable
- Registers are Addressed for MOVEP (Move Peripheral) and DMAC Compatibility

PIN ARRANGMENT

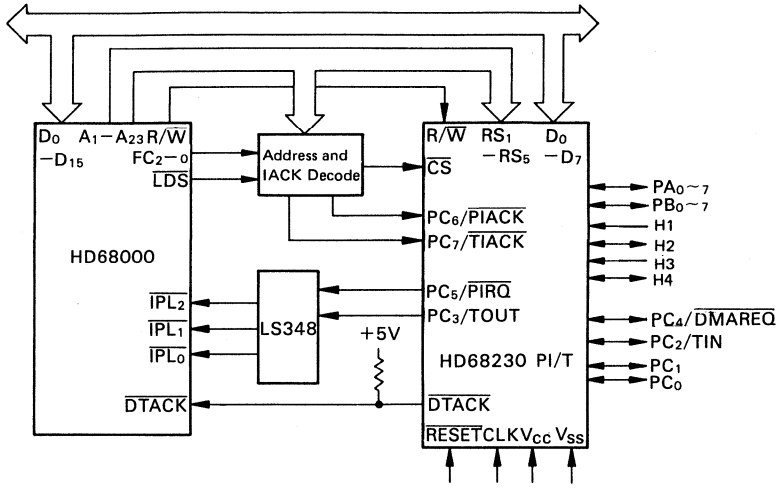


(Top View)

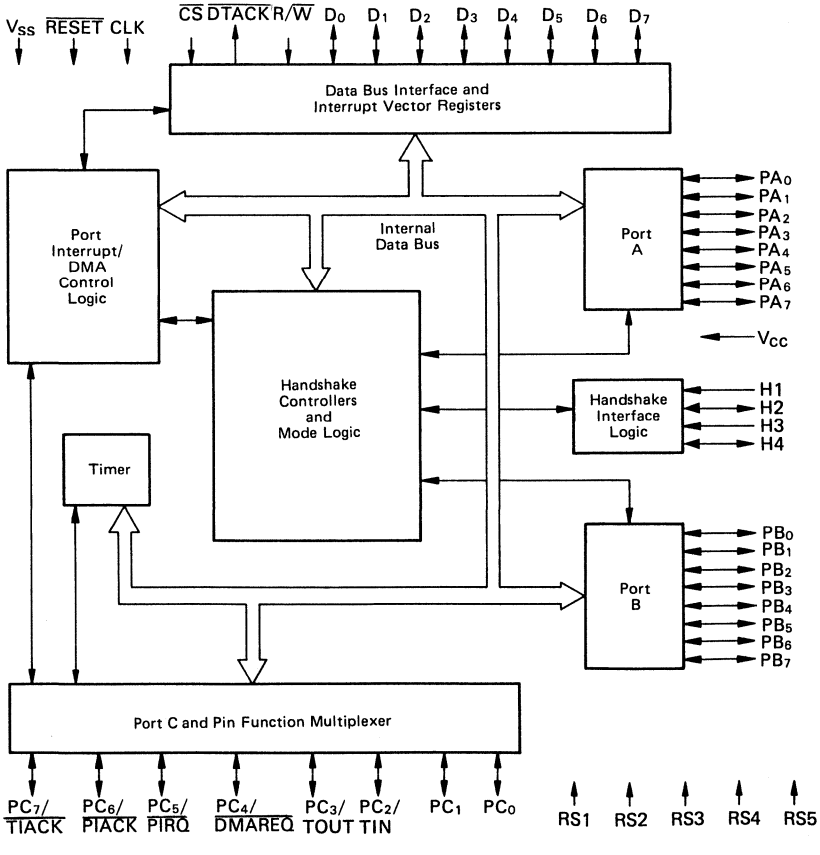
TYPE OF PRODUCTS

Type No.	Bus Timing
HD68230P8	8 MHz
HD68230P10	10 MHz

PI/T SYSTEM BLOCK DIAGRAM



BLOCK DIAGRAM



■ ABSOLUTE MAXIMUM RATINGS

Item	Symbol	Value	Unit
Supply Voltage	V_{CC}*	-0.3~+7.0	V
Input Voltage	V_{in}*	-0.3~+7.0	V
Operating Temperature Range	T_{opr}	0~+70	°C
Storage Temperature	T_{stg}	-55~+150	°C

* With respect to V_{SS} (SYSTEM GND)

(NOTE) This device contains circuitry to protect the inputs against damage due to high static voltages or electric fields; however, it is advised that normal precautions be taken to avoid application of any voltage higher than maximum-rated voltages to this high impedance circuit. Rehability of operation is enhanced if unused inputs are tied to an appropriate logic voltage level (e.g., either V_{SS} or V_{CC}).

■ RECOMMENDED OPERATING CONDITIONS

Item	Symbol	min	typ	max	Unit
Supply Voltage	V_{CC}*	4.75	5.0	5.25	V
Input Voltage	V_{IH}*	V_{SS} + 2.0	–	V_{CC}	V
	V_{IL}*	V_{SS} - 0.3	–	V_{SS} + 0.8	V
Operating Temperature	T_{opr}	0	25	70	°C

* With respect to V_{SS} (SYSTEM GND)

■ ELECTRICAL CHARACTERISTICS

• DC CHARACTERISTICS (V_{CC} = 5V ±5%, V_{SS} = 0V, Ta = 0 ~ +70°C unless otherwise noted.)

Item		Symbol	Test Condition	min	typ	max	Unit
Input "High" Voltage		V_{IH}		V_{SS}+2.0	–	V_{CC}	V
Input "Low" Voltage		V_{IL}		V_{SS}-0.3	–	V_{SS}+0.8	V
Input Leakage Current	H1, H3, R/$\overline{W}$, $\overline{RESET}$ CLK, RS1-RS5, $\overline{CS}$	I_{in}	Vin = 0 ~ 5.25V	–	–	10.0	μA
Three-State (Off State) Input Current	$\overline{DTACK}$, PC_0-PC_7, D_0-D_7	I_{TSI}	Vin = 0.4 ~ 2.4V	–		20	μA
	H2, H4, PA_0-PA_7, PB_0-PB_7			-0.1		-1.0	mA
Output "High" Voltage	D_0-D_7, $\overline{DTACK}$ PA_0-PA_7, PB_0-PB_7, H2, H4 PC_0-PC_7	V_{OH}	I_{OH} = -400μA I_{OH} = -150μA I_{OH} = -100μA	V_{SS}+2.4	–	–	V
Output "Low" Voltage	PC_3/TOUT, PC_5/$\overline{PIRQ}$ D_0-D_7, $\overline{DTACK}$ PA_0-PA_7, PB_0-PB_7, H2, H4 PC_0-PC_2, PC_4, PC_6, PC_7	V_{OL}	I_{OL} = 8.8mA I_{OL} = 5.3mA I_{OL} = 2.4mA I_{OL} = 2.4mA	–	–	0.5	V
Power Dissipation		P_{INT}	TA = 0°C	–	–	500	mW
Capacitance (Package Type Dependent)		C_{in}	V_{in} = 0V, Ta = 25°C	–	–	15	pF

• CLOCK TIMING

Characteristic	Symbol	8 MHz HD68230P8		10 MHz HD68230P10		Unit
		min	max	min	max	
Frequency of Operation	f	2.0	8.0	2.0	10.0	MHz
Cycle Time	t_{CYC}	125	500	100	500	ns
Clock Pulse Width	t_{CL}	55	250	45	250	ns
	t_{CH}	55	250	45	250	
Clock Rise and Fall Times	t_{Cr}	–	10	–	10	ns
	t_{Cf}	–	10	–	10	

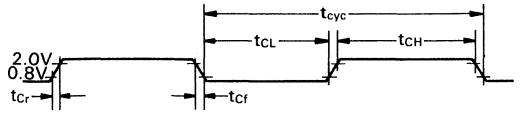


Figure 1 Input Clock Waveform

- **AC CHARACTERISTICS (V_{CC} = 5V ±5%, V_{SS} = 0V, Ta = 0 ~ +70°C unless otherwise noted.)**

No.	Characteristic	8 MHz HD68230P8		10 MHz HD68230P10		Unit
		min	max	min	max	
1	R/$\overline{W}$ RS1-RS5 Valid to $\overline{CS}$ Low (Setup Time)	0	–	0	–	ns
2(*11)	$\overline{CS}$ Low to R/$\overline{W}$ and RS1-RS5 Invalid (Hold Time)	100	–	65	–	ns
3(*1)	$\overline{CS}$ Low to CLK Low (Setup Time)	30	–	20	–	ns
4(*2)	$\overline{CS}$ Low to Data Out Valid (Delay)	–	75	–	60	ns
5	RS1-RS5 Valid to Data Out Valid (Delay)	–	140	–	100	ns
6	CLK Low to DTACK Low (Read/Write Cycle) (Delay)	0	70	0	60	ns
7(*3)	$\overline{DTACK}$ Low to $\overline{CS}$ High (Hold Time)	0	–	0	–	ns
8	$\overline{CS}$ or $\overline{PIACK}$ or $\overline{TIACK}$ High to Data Out Invalid (Hold Time)	0	–	0	–	ns
9	$\overline{CS}$ or $\overline{PIACK}$ or $\overline{TIACK}$ High to D_0-D_7 High Impedance (Delay)	–	50	–	45	ns
10	$\overline{CS}$ or $\overline{PIACK}$ or $\overline{TIACK}$ High to $\overline{DTACK}$ High (Delay)	–	50	–	30	ns
11	$\overline{CS}$ or $\overline{PIACK}$ or $\overline{TIACK}$ High to $\overline{DTACK}$ High Impedance (Delay)	–	100	–	55	ns
12	Data Invalid to $\overline{CS}$ Low (Setup Time)	0	–	0	–	ns
13	$\overline{CS}$ Low to Data in Invalid (Hold Time)	100	–	65	–	ns
14	Input Data Valid to H1(H3) Asserted (Setup Time)	100	–	60	–	ns
15	H1(H3) Asserted to Input Data Invalid (Hold Time)	20	–	20	–	ns
16	Handshake Input H1(H4) Pulse Width Asserted	40	–	40	–	ns
17	Handshake Input (H1-H4) Pulse Width Negated	40	–	40	–	ns
18	H1(H3) Asserted to H2(H4) Negated (Delay)	–	150	–	120	ns
19	CLK Low to H2(H4) Asserted (Delay)	–	100	–	100	ns
20(*4)	H2(H4) Asserted to H1(H3) Asserted	0	–	0	–	ns
21(*5)	CLK Low to H2(H4) Pulse Negated (Delay)	–	125	–	125	ns
22(*9,*12)	Synchronized H1(H3) to CLK Low on which $\overline{DMAREQ}$ is Asserted (See Figures 18 and 19)	2.5	3.5	2.5	3.5	CLK Per
23	CLK Low $\overline{DMAREQ}$ is Asserted to CLK Low on which $\overline{DMAREQ}$ is Negated	3	3	3	3	CLK Per
24	CLK Low to Output Data Valid (Delay) (Modes 0, 1)	–	150	–	120	ns
25(*9,*12)	Synchronized H1(H3) to Output Data Invalid (Modes 0, 1)	1.5	2.5	1.5	2.5	CLK Per
26	H1 Negated to Output Data Valid (Modes 2, 3)	–	70	–	50	ns
27	H1 Asserted to Output Data High Impedance (Modes 2, 3)	0	70	0	70	ns
28	Read Data Valid to $\overline{DTACK}$ Low (Setup Time)	0	–	0	–	ns
29	CLK Low to Data Output Valid (Interrupt Acknowledge Cycle)	–	120	–	100	ns
30(*7)	H1(H3) Asserted to CLK High (Setup Time)	50	–	40	–	ns
31	$\overline{PIACK}$ or $\overline{TIACK}$ Low to CLK Low (Setup Time)	50	–	40	–	ns
32(*12)	Synchronized $\overline{CS}$ to CLK Low on which $\overline{DMAREQ}$ is Asserted (See Figures 18 and 19)	3	3	3	3	CLK Per
33(*9,*12)	Synchronized H1(H3) to CLK Low on which H2(H4) is Asserted	3.5	4.5	3.5	4.5	CLK Per
34	CLK Low to $\overline{DTACK}$ Low (Interrupt Acknowledge Cycle) (Delay)	–	75	–	60	ns
35	CLK Low to $\overline{DMAREQ}$ Low (Delay)	0	120	0	100	ns
36	CLK Low to $\overline{DMAREQ}$ High (Delay)	0	120	0	100	ns
–	CLK Low to $\overline{PIRQ}$ Low or High Impedance	–	200	–	150	ns
–(*8)	TIN Frequency (External Clock) – Prescaler Used	0	1	0	1	Fclk(Hz)(6)
–	TIN Frequency (External Clock) – Prescaler Not used	0	1/32	0	1/32	Fclk(Hz)(6)
–	TIN Pulse Width High or Low (External Clock)	55	–	45	–	ns
–	TIN Pulse Width Low (Run/Halt Control)	1	–	1	–	CLK
–	CLK Low to TOUT High, Low, or High Impedance	0	200	0	150	ns
–	$\overline{CS}$, $\overline{PIACK}$, or $\overline{TIACK}$ High to $\overline{CS}$, $\overline{PIACK}$, or $\overline{TIACK}$ Low	50	–	30	–	ns

(NOTES)(*1) This specification only applies if the PI/T had completed all operations initiated by the previous bus cycle when $\overline{CS}$ was asserted. Following a normal read or write bus cycle, all operations are complete within three CLKs after the falling edge of the CLK pin on which $\overline{DTACK}$ was asserted. If $\overline{CS}$ is asserted prior to completion of these operations, the new bus cycle, and hence, $\overline{DTACK}$ is postponed.
If all operations of the previous bus cycle were complete when $\overline{CS}$ was asserted, this specification is made only to insure that $\overline{DTACK}$ is asserted with respect to the falling edge of the CLK pin as shown in the timing diagram, not to guarantee operation of the part. If the $\overline{CS}$ setup time is violated, $\overline{DTACK}$ may be asserted as shown, or may be asserted one clock cycle later.

(*2) Assuming the RS1-RS5 to Data Valid Time has also expired.

(*3) This specification imposes a lower bound on $\overline{CS}$ low time, guaranteeing that $\overline{CS}$ will be low for at least 1 CLK period.

(*4) This specification assures recognition of the asserted edge of H1(H3).

(*5) This specification applies only when a pulsed handshake option is chosen and the pulse is not shortened due to an early asserted edge of H1 (H3).

(*6) CLK refers to the actual frequency of the CLK pin, not the maximum allowable CLK frequency.

(*7) If timing number 30 is violated, H1(H3) will be recognized no later than the next rising edge of the clock.

(*8) This limit applies to the frequency of the signal at TIN compared to the frequency of the CLK signal during each clock cycle. If any period of the waveform at TIN is smaller than the period of the CLK signal at that instant, then it is likely that the timer circuit will completely ignore one cycle of the TIN signal. Since the frequency measured by a frequency counter is the average frequency of a signal over a specific length of time, the actual frequency at any one time will vary above and below the average. These variations occur in both the TIN and CLK signals.
If these two signals are derived from different sources they will have different instantaneous frequency variations. In this case the frequency applied to the TIN pin must be distinctly less than the frequency at the CLK pin to avoid lost cycles of the TIN signal. Measurements have shown that with signals derived from different crystal oscillators applied to the TIN and CLK pins with fast rise and fall times. The TIN frequency can approach 80 to 90% of the frequency of the CLK signal without a loss of a cycle of the TIN signal.
If these two signals are derived from the same frequency source then the frequency of the signal applied to TIN can be 100% of the frequency at the CLK pin. They may be generated by different buffers from the same signal or one may be an inverted version of the other.
The TIN signal may be generated by an 'AND' function of the clock and a control signal.

(*9) This limit applies in every case. There are no exceptions to this specification.

(*10) If a bus access and peripheral access occur at the same time, add one clock to specifications 22 and 33.

(*11) See BUS INTERFACE CONNECTION section for exception.

(*12) This Limit specifies the nominal outputing in PI/T clock cycles. To obtain the output timing in nanoseconds, add or subtract the appropriate setup time and/or propagation time from the signals to the respective clock edges.

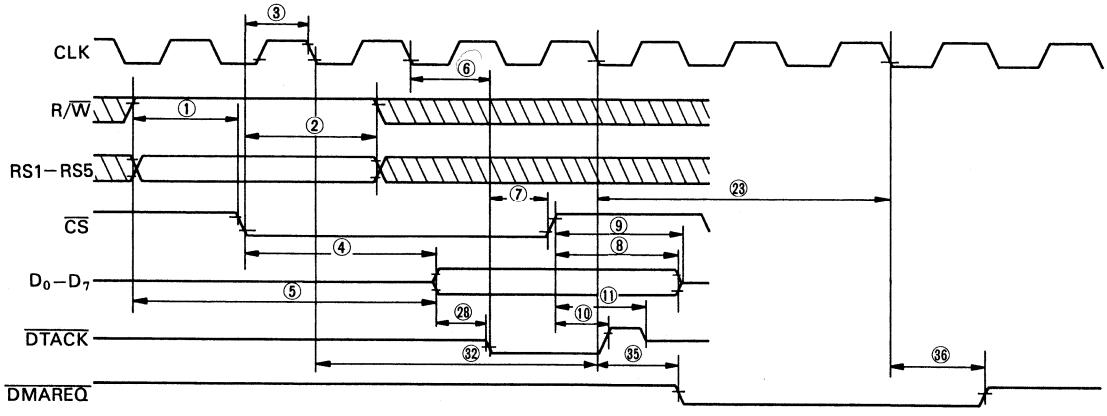


Figure 2 Bus Read Cycle Timing

NOTE: Timing measurements are referenced to and from a low voltage of 0.8 volts and a high voltage of 2.0 volts, unless otherwise noted.

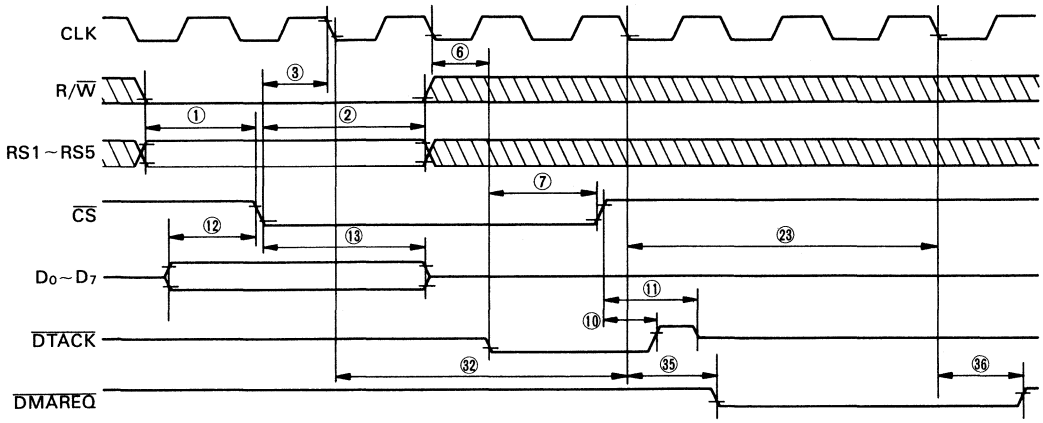


Figure 3 Bus Write Cycle Timing

NOTE: Timing measurements are referenced to and from a low voltage of 0.8 volts and a high voltage of 2.0 volts, unless otherwise noted.

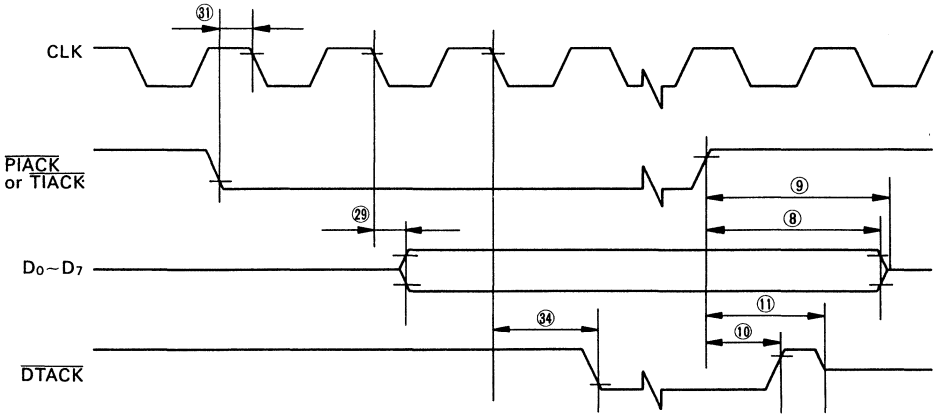


Figure 4 Interrupt Acknowledge Functional Timing Diagram

NOTE: Timing measurements are referenced to and from a low voltage of 0.8 volts and a high voltage of 2.0 volts, unless other wise noted.

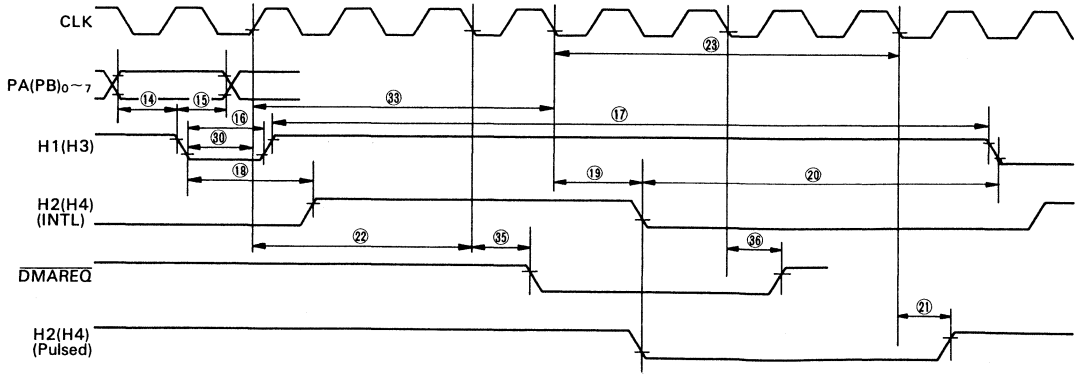


Figure 5 Peripheral Interface Input Timing

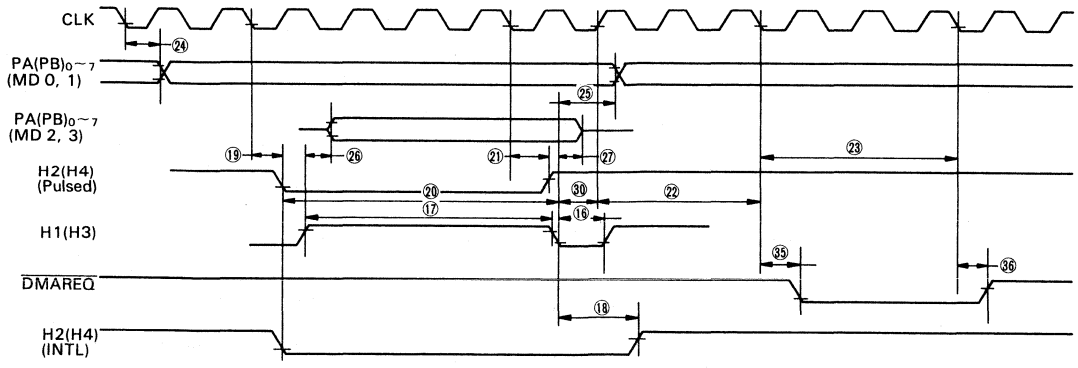


Figure 6 Peripheral Interface Output Timing

■ GENERAL DESCRIPTION

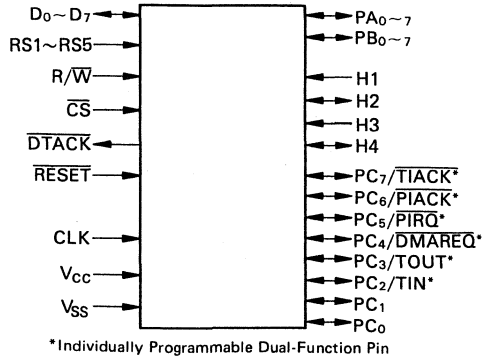
The PI/T consists of two logically independent sections: the ports and the timer. The port section consists of Port A (PA_{0-7}), Port B (PB_{0-7}), four handshake pins (H1, H2, H3, and H4), two general I/O pins, and six dual-function pins. The dual-function pins can individually operate as a third port (Port C) or an alternate function related to either Ports A and B, or the timer. The four programmable handshake pins, depending on the mode, can control data transfer to and from the ports, or can be used as interrupt generating inputs, or I/O pins.

The timer consists of a 24-bit counter, optionally clocked by a 5-bit prescaler. Three pins provide complete timer I/O: PC_2/TIN, PC_3/TOUT, and $PC_7/\overline{TIACK}$. Of course, only the ones needed for the given configuration perform the timer function, while the others remain Port C I/O.

The system bus interface provides for asynchronous transfer of data from the PI/T to a bus master over the data bus (D_0-D_7). Data transfer acknowledge ($\overline{DTACK}$), register selects (RS1-RS5), chip select, the read/write line (R/$\overline{W}$), and Port Interrupt Acknowledge ($\overline{PIACK}$) or Timer Interrupt Acknowledge ($\overline{TIACK}$) control data transfer between the PI/T and the HD68000.

■ PIN DESCRIPTION

Throughout the data sheet, signals are presented using the terms active and inactive or asserted and negated independent of whether the signal is active in the high-voltage state or low-voltage state. (The active state of each logic pin is given below.) Active low signals are denoted by a superscript bar. R/$\overline{W}$ indicates a write is active low and a read active high.



*Individually Programmable Dual-Function Pin

Figure 7 Input and Output Signals

Bidirectional Data Bus – (D_0-D_7)

The data bus pins D_0-D_7 form an 8-bit bidirectional data bus to/from the HD68000 or other bus master. These pins are active high.

Register Selects – (RS1-RS5)

RS1-RS5 are active high high-impedance inputs that determine which of the 25 possible registers is being addressed. They are provided by the HD68000 or other bus master.

Read/Write Input – (R/$\overline{W}$)

R/$\overline{W}$ is the high-impedance Read/Write signal from the HD68000 or bus master, indicating whether the current bus cycle is a read (high) or write (low) cycle.

Chip Select Input – ($\overline{CS}$)

$\overline{CS}$ is a high-impedance input that selects the PI/T registers for the current bus cycle. Address strobe and the data strobe (upper or lower) of the bus master, along with the appropriate address bits, must be included in the chip select equation. A low level corresponds to an asserted chip select.

Data Transfer Acknowledge Output – ($\overline{DTACK}$)

$\overline{DTACK}$ is an active low output that signals the completion of the bus cycle. During read or interrupt acknowledge cycles, $\overline{DTACK}$ is asserted by the HD68230 after data has been provided on the data bus; during write cycles it is asserted after data has been accepted at the data bus. Data transfer acknowledge is compatible with the HD68000 and with other Hitachi bus masters such as the HD68450 DMA controller. A holding resistor is required to maintain $\overline{DTACK}$ high between bus cycles.

Reset Input – ($\overline{RESET}$)

$\overline{RESET}$ is a high-impedance input used to initialize all PI/T functions. All control and data direction registers are cleared and most internal operations are disabled by the assertion of $\overline{RESET}$ (low).

Clock Input – (CLK)

The clock pin is a high-impedance TTL-compatible signal with the same specifications as the HD68000. The PI/T contains dynamic logic throughout, and hence this clock must not be gated off at any time. It is not necessary that this clock maintain any particular phase relationship with the HD68000 clock. It may be connected to an independent frequency source (faster or slower) as long as all bus specifications are met.

Port A and Port B – (PA_0-PA_7 and PB_0-PB_7)

Ports A and B are 8-bit ports that may be concatenated to form a 16-bit port in certain modes. The ports may be controlled in conjunction with the handshake pins H1-H4. For stabilization during system power-up, Ports A and B have internal pullup resisters to V_{CC}. All port pins are active high.

Handshake pins (I/O depending on the Mode and Submode) – (H1-H4)

Handshake pins H1-H4 are multi-purpose pins that (depending on the operational model) may provide an interlocked handshake, a pulsed handshake, an interrupt input (independent of data transfers), or simple I/O pins. For stabilization during system power-up, H2 and H4 have internal pullup resisters to V_{CC}. Their sense (active high or low) may be programmed in the Port General Control Register bits 3-0. Independent of the mode, the instantaneous level of the handshake pins can be read from the Port Status Register.

(PC_0-PC_7 /Alternate function) – (Port C)

This port can be used as eight general purpose I/O pins (PC_0-PC_7) or any combination of six special function pins and two general purpose I/O pins (PC_0-PC_1). (Each dual function pin can be standard I/O or a special function independent of the other port C pins.) The dual function pins are defined in the following paragraphs. When used as port C pin, these pins are active high. They may be individually programmed as inputs or outputs by the Port C Data Direction Register.

The alternate functions (TIN, TOUT, and $\overline{\text{TIACK}}$) are timer I/O pins. TIN may be used as a rising-edge triggered external clock input or an external run/halt control pin (the timer is in the run state if run/halt is high and in the halt state if run/halt is low). TOUT may provided an active low timer interrupt request output or a general-purpose square-wave output, initially high. $\overline{\text{TIACK}}$ is an active low high-impedance input used for timer interrupt acknowledge.

Port A and B functions have an independent pair of active low interrupt request ($\overline{\text{PIRQ}}$) and interrupt acknowledge ($\overline{\text{PIACK}}$) pins.

The $\overline{\text{DMAREQ}}$ (Direct Memory Access Request) pin provides an active low Direct Memory Access Controller (DMAC) request pulse of 3 clock cycles, completely compatible with the HD68450 DMAC.

■ REGISTER MODEL

A register model that includes the corresponding Register Selects is shown in Table 1.

Table 1 Register Model

Register Select Bits 5	4	3	2	1	7	6	5	4	3	2	1	0	
0	0	0	0	0	Port Mode Control		H34 Enable	H12 Enable	H4 Sense	H3 Sense	H2 Sense	H1 Sense	Port General Control Register
0	0	0	0	1	*	SVCRQ Select		Interrupt PFS		Port Interrupt Priority Control			Port Service Request Register
0	0	0	1	0	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Port A Data Direction Register
0	0	0	1	1	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Port B Data Direction Register
0	0	1	0	0	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Port C Data Direction Register
0	0	1	0	1	Interrupt Vector Number						*	*	Port Interrupt Vector Register
0	0	1	1	0	Port A Submode		H2 Control			H2 Int Enable	H1 SVCRQ Enable	H1 Stat Ctrl	Port A Control Register
0	0	1	1	1	Port B Submode		H4 Control			H4 Int Enable	H3 SVCRQ Enable	H3 Stat Ctrl	Port B Control Register
0	1	0	0	0	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Port A Data Register
0	1	0	0	1	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Port B Data Register
0	1	0	1	0	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Port A Alternate Register
0	1	0	1	1	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Port B Alternate Register
0	1	1	0	0	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Port C Data Register
0	1	1	0	1	H4 Level	H3 Level	H2 Level	H1 Level	H4S	H3S	H2S	H1S	Port Status Register
0	1	1	1	0	*	*	*	*	*	*	*	*	(null)
0	1	1	1	1	*	*	*	*	*	*	*	*	(null)
1	0	0	0	0	TOUT/$\overline{\text{TIACK}}$ Control			ZD Ctrl	*	Clock Control		Timer Enable	Timer Control Register
1	0	0	0	1	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Timer Interrpt Vector Register
1	0	0	1	0	*	*	*	*	*	*	*	*	(null)
1	0	0	1	1	Bit 23	Bit 22	Bit 21	Bit 20	Bit 19	Bit 18	Bit 17	Bit 16	Counter Preload Register (High)
1	0	1	0	0	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	(Mid)
1	0	1	0	1	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	(Low)
1	0	1	1	0	*	*	*	*	*	*	*	*	(null)
1	0	1	1	1	Bit 23	Bit 22	Bit 21	Bit 20	Bit 19	Bit 18	Bit 17	Bit 16	Count Register (High)
1	1	0	0	0	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	(Mid)
1	1	0	0	1	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	(Low)
1	1	0	1	0	*	*	*	*	*	*	*	ZDS	Timer Status Register
1	1	0	1	1	*	*	*	*	*	*	*	*	(null)
1	1	1	0	0	*	*	*	*	*	*	*	*	(null)
1	1	1	0	1	*	*	*	*	*	*	*	*	(null)
1	1	1	1	0	*	*	*	*	*	*	*	*	(null)
1	1	1	1	1	*	*	*	*	*	*	*	*	(null)

*Unused, read as zero.

■ PROGRAMMER'S MODEL

Table 2 PI/T Register Addressing Assignments

Register	Register Select Bits					Accessible	Affected by Reset	Affected by Read Cycle
	5	4	3	2	1			
Port General Control Register (PGCR)	0	0	0	0	0	R W	Yes	No
Port Service Request Register (PSRR)	0	0	0	0	1	R W	Yes	No
Port A Data Direction Register (PADDR)	0	0	0	1	0	R W	Yes	No
Port B Data Direction Register (PBDDR)	0	0	0	1	1	R W	Yes	No
Port C Dara Direction Register (PCDDR)	0	0	1	0	0	R W	Yes	No
Port Interrupt Vector Register (PIVR)	0	0	1	0	1	R W	Yes	No
Port A Control Register (PACR)	0	0	1	1	0	R W	Yes	No
Port B Control Register (PBCR)	0	0	1	1	1	R W	Yes	No
Port A Data Register (PADR)	0	1	0	0	0	R W	No	**
Port B Data Register (PBDR)	0	1	0	0	1	R W	No	**
Port A Alternate Register (PAAR)	0	1	0	1	0	R	No	No
Port B Alternate Register (PBAR)	0	1	0	1	1	R	No	No
Port C Data Register (PCDR)	0	1	1	0	0	R W	No	No
Port Status Register (PSR)	0	1	1	0	1	R W*	Yes	No
Timer Control Register (TCR)	1	0	0	0	0	R W	Yes	No
Timer Interrupt Vector Register (TIVR)	1	0	0	0	1	R W	Yes	No
Counter Preload Register High (CPRH)	1	0	0	1	1	R W	No	No
Counter Preload Register Middle (CPRM)	1	0	1	0	0	R W	No	No
Counter Preload Register Low (CPRL)	1	0	1	0	1	R W	No	No
Count Register High (CNTRH)	1	0	1	1	1	R	No	No
Count Register Middle (CNTRM)	1	1	0	0	0	R	No	No
Count Register Low (CNTRL)	1	1	0	0	1	R	No	No
Timer Status Register (TSR)	1	1	0	1	0	RW*	Yes	No

* A write to this register may perform a special status resetting operation.
** Mode dependent.

R = Read
W = Write

The internal accessible register organization is represented in Table 2. Address space within the address map is reserved for future expansion. Throughout the PI/T data sheet the following conventions are maintained:

(1) A read from a reserved location in the map results in a read from the "null register." The null register returns all zeros for data and results in a normal bus cycle. A write to one of these locations results in a normal bus cycle but no write occurs.
(2) Unused bits of a defined register are denoted by "•" and are read as zeroes.
(3) Bits that are unused in the chosen mode/submode but are used in others, are denoted by "X", and are readable and writeable. Their content, however, is ignored in the chosen mode/submode.
(4) All registers are addressable as 8-bit quantities. To facilitate operation with the MOVEP instruction and the DMAC, addresses are ordered such that certain sets of registers may also be accessed as words (2 bytes) or long words (4 bytes).

- **Port General Control Register (PGCR)**

Port General Control Register (PGCR)

7	6	5	4	3	2	1	0
Port Mode Control		H34 Enable	H12 Enable	H4 Sense	H3 Sense	H2 Sense	H1 Sense

The Port General Control Register (PGCR) controls may of the functions that are common to the overall operation of the ports. The PGCR is composed of three major fields; bits 7 and 6 define the operational mode of Ports A and B and affect operation of the handshake pins and status bits; bits 5 and 4 allow a software controlled disabling of particular hardware associated with the handshake pins of each port; and bits 3-0 define the sense of the handshake pins. The PGCR is always readable and writeable.

All bits are reset to 0 when the $\overline{\text{RESET}}$ pin is asserted.

The Port Mode Control field should be altered only when the H12 Enable and H34 Enable bits are 0. Except when Mode 0 is desired, the Port General Control register must be written once to establish the mode, and again to enable the respective

operation(s).

PGCR

7	6	Port Mode Control
0	0	Mode 0 (Unidirectional 8-Bit Mode)
0	1	Mode 1 (Unidirectional 16-Bit Mode)
1	0	Mode 2 (Bidirectional 8-Bit Mode)
1	1	Mode 3 (Bidirectional 16-Bit Mode)

PGCR

5	H34 Enable
0	Disabled
1	Enabled

PGCR

4	H12 Enable
0	Disabled
1	Enabled

PGCR

3-0	Handshake Pin Sense
0	The associated pin is at the high-voltage level when negated and at the low-voltage level when asserted.
1	The associated pin is at the low-voltage level when negated and at the high voltage level when asserted.

- **Port Service Request Register (PSRR)**

Port Service Request Register (PSRR)

7	6	5	4	3	2	1	0
*	SVCRQ Select		Interrupt PFS		Port Interrupt Priority Control		

The Port Service Request Register (PSRR) controls other functions that are common to the overall operation to the ports. It is composed of four major fields; bit 7 is unused and is always read as 0; bits 6 and 5 define whether interrupt or DMA requests are generated from activity on the H1 and H3 handshake pins; bit 4 and 3 determine whether two dual function pins operate as Port C or port interrupt request/acknowledge pins; and bits 2, 1, and 0 control the priority among all port interrupt sources. Since bits 2, 1, and 0 affect interrupt operation, it is recommended that they be changed only when the affected interrupt(s) is (are) disabled or known to remain inactive. The PSRR is always readable and writeable.

All bits are reset to 0 when the $\overline{\text{RESET}}$ pin is asserted.

PSRR

6	5	SVCRQ Select
0	X	The $PC_4/\overline{DMAREQ}$ pin carries the PC_4 function; DMA is not used.
1	0	The $PC_4/\overline{DMAREQ}$ pin carries the $\overline{DMAREQ}$ function and is associated with duouble-buffered transfers controlled by H1. H1 is removed from the PI/T's interrupt structure, and thus, does not cause interrupt requests to be generated. To obtain $\overline{DMAREQ}$ pulses, Port A Control Register bit 1 (H1 SVCRQ Enable) must be a 1.
1	1	The $PC_4/\overline{DMAREQ}$ pin carries the $\overline{DMAREQ}$ function and is associated with double-buffered transfers controlled by H3. H3 is removed from the PI/T's interrupt structure, and thus, does not cause interrupt requests to be generated. To obtain $\overline{DMAREQ}$ pulses, Port B Control Register bit 1 (H3 SVCRQ Enable) must be a 1.

PSRR

4	3	Interrupt Pin Function Select
0	0	The $PC_5/\overline{PIRQ}$ pin carries the PC5 function. The $PC_6/\overline{PIACK}$ pin carries the PC6 function.
0	1	The $PC_5/\overline{PIRQ}$ pin carries the $\overline{PIRQ}$ function. The $PC_6/\overline{PIACK}$ pin carries the PC6 function.
1	0	The $PC_5/\overline{PIRQ}$ pin carries the PC5 function. The $PC_6/\overline{PIACK}$ pin carries the $\overline{PIACK}$ function.
1	1	The $PC_5/\overline{PIRQ}$ pin carries the $\overline{PIRQ}$ function. The $PC_6/\overline{PIACK}$ pin carries the $\overline{PIACK}$ function.

Bits 2, 1, and 0 determine port interrupt priority. The priority is shown in descending order left to right.

PSRR 2	1	0	Port Interrupt Priority Control Highest ..... Lowest			
0	0	0	H1S	H2S	H3S	H4S
0	0	1	H2S	H1S	H3S	H4S
0	1	0	H1S	H2S	H4S	H3S
0	1	1	H2S	H1S	H4S	H3S
1	0	0	H3S	H4S	H1S	H2S
1	0	1	H3S	H4S	H2S	H1S
1	1	0	H4S	H3S	H1S	H2S
1	1	1	H4S	H3S	H2S	H1S

- **Port A Data Direction Register (PADDR)**

The Port A Data Direction Register (PADDR) determines the direction and buffering characteristics of each of the Port A pins. One bit in the PADDR is assigned to each pin. A 0 indicates that the pin is used as an input, while a 1 indicates it is used as an output. The PADDR is always readable and writeable. This register is ignored in Mode 3.

All bits are reset to the 0 (input) state when the $\overline{\text{RESET}}$ pin is asserted.

- **Port B Data Direction Register (PBDDR)**

The PBDR is identical to the PADDR for the Port B pins and the Port B Data Register, except that this register is ignored in Modes 2 and 3.

- **Port C Data Direction Register (PCDDR)**

The Port C Data Direction Register (PCDDR) specifies whether each dual-function pin that is chosen for Port C operation is an input (0) or an output (1) pin. The PCDDR, along with bits that determine the respective pin's function, also specify the exact hardware to be accessed at the Port C Data Register address. (See the Port C Data Register description for more details.) The PCDDR is an 8-bit register that is readable and writeable at all times. Its operation is independent of the chosen PI/T mode.

These bits are cleared to 0 when the $\overline{\text{RESET}}$ pin is asserted.

- **Port Interrupt Vector Register (PIVR)**

Port Interrupt Vector Register (PIVR)

7	6	5	4	3	2	1	0
Interrupt Vector Number						*	*

The Port Interrupt Vector Register (PIVR) contains the upper order six bits of the four port interrupt vectors. The

contents of this register may be read two ways; by an ordinary read cycle, or by a port interrupt acknowledge bus cycle. The exact data read depends on how the cycle was initiated and other factors. Behavior during a port interrupt acknowledge cycle is summarized above in Table 5.

From a normal read cycle ($\overline{CS}$), there is never a consequence to reading this register. Following negation of the $\overline{RESET}$ pin, but prior to writing to the PIVR, a $OF will be read. After writing to the register, the upper 6 bits may be read and the lower 2 bits are forced to 0. No prioritization computation is performed.

- **Port A Control Register (PACR)**

Port A Control Register (PACR)

7	6	5	4	3	2	1	0
Port A Submode		H2 Control			H2 Int. Enable	H1 SVCRQ Enable	H1 Stat. Ctrl.

The Port A Control Register (PACR) in conjunction with the programmed mode and the Port B submode, control the operation of Port A and the handshake pins H1 and H2. The Port A Control Register contains five fields; bits 7 and 6 specify the Port A submode; bits 5, 4, and 3 control the operation of the H2 handshake pin and H2S atatus bit; bit 2 determines whether an interrupt will be generated when the H2S status bit goes to 1; bit 1 determines whether a service request (interrupt request or DMA request) will occur; bit 0 controls the operation of the H1S status bit. The PACR is always readable and writeable.

All bits are cleared to 0 when the $\overline{RESET}$ pin is asserted.

When the Port A submode field is relevant in a mode/submode definition, it must not be altered unless the H12 Enable bit in the Port General Control Register is 0. (See Table 4.)

The operation of H1 and H2 and their related status bits is given below, for each of the modes specified by Port General Control Register bits 7 and 6. This description is organized such that for each mode/submode all programmable options of each pin and status bit are given.

Bits 2 and 1 carry the same meaning in each mode/submode, and thus are specified only one.

PACR

2	H2 Interrupt Enable
0	The H2 interrupt is disabled.
1	The H2 interrupt is enabled.

PACR

1	H1 SVCRQ Enable
0	The H1 interrupt and DMA request are disabled.
1	The H1 interrupt and DMA request are enabled.

(1) PACR Mode 0 Port A Submode 00

PACR

5	4	3	H2 Control
0	X	X	Input pin – status only.
1	0	0	Output pin – always negated.
1	0	1	Output pin – always asserted.
1	1	0	Output pin – interlocked input handshake protocol.
1	1	1	Output pin – pulsed input handshake protocol.

PACR

0	H1 Status Control
X	Not Used.

(2) PACR Mode 0 Port A Submode 01

PACR

5	4	3	H2 Control
0	X	X	Input pin – status only.
1	0	0	Output pin – always negated.
1	0	1	Output pin – always asserted.
1	1	0	Output pin – interlocked output handshake protocol.
1	1	1	Output pin – pulsed output handshake protocol.

PACR

0	H1 Status Control
0	The H1S status bit is 1 when either the Port A initial or final output latch can accept new data. It is 0 when both latches are full and cannot accept new data.
1	The H1S status bit is 1 when both of the Port A output latches are empty. It is 0 when at least one latch is full.

(3) PACR Mode 0 Port A Submode 1X

PACR

5	4	3	H2 Control
0	X	X	Input pin – status only.
1	X	0	Output pin – always negated.
1	X	1	Output pin – always asserted.

PACR

0	H1 Status Control
X	Not used.

(4) PACR Mode 1 Port A Submode XX Port B Submode X0

PACR

5	4	3	H2 Control
0	X	X	Input pin – status only.
1	X	0	Output pin – always negated.
1	X	1	Output pin – always asserted.

PACR

0	H1 Status Control
X	Not used.

(5) PACR Mode 1 Port A Submode XX Port B Submode X1

PACR

5	4	3	H2 Control
0	X	X	Input pin – status only.
1	X	0	Output pin – always negated.
1	X	1	Output pin – always asserted.

PACR

0	H1 Status Control
X	Not used.

(6) PACR Mode 2

PACR

5	4	3	H2 Control
X	X	0	Output pin – interlocked output handshake protocol.
X	X	1	Output pin – pulsed output handshake protocol.

PACR

0	H1 Status Control
0	The H1S status bit is 1 when either the Port B initial or final output latch can accept new data. It is 0 when both latches are full and cannot accept new data.
1	The H1S status bit is 1 when both of the Port B output latches are empty. It is 0 when at least one latch is full.

(7) PACR Mode 3

PACR

5 4 3	H2 Control
X X 0	Output pin – interlocked output handshake protocol.
X X 1	Output pin – pulsed output handshake protocol.

PACR

0	H1 Status Control
0	The H1S status bit is 1 when either the initial or final output latch of Port A and B can accept new data. It is 0 when both latches are full and cannot accept new data.
1	The H1S status bit is 1 when both the initial and final output latches of Ports A and B are empty. It is 0 when either the initial or final latch of Ports A and B is full.

- **Port B Control Register (PBCR)**

Port B Control Register (PBCR)

7	6	5	4	3	2	1	0
Port B Submode		H4 Control			H4 Int. Enable	H3 SVCRQ Enable	H3 Stat. Ctrl.

The Port B Control Register (PBCR) specifies the operation of Port B and the handshake pins H3 and H4. The Port B control register contains five fields; bits 7 and 6 specify the Port B submode; bis 5, 4, and 3 control the operation of the H4 handshake pin and H4S status bit; bit 2 determines whether an interrupt will be generated when the H4S status bit goes to 1; bit 1 determines whether a service request (interrupt request or DMA request) will occur; bit 0 controls the operation of the H3S status bit. The PACR is always readable and writeable. There is never a consequence to reading the register.

All bits are cleared to 0 when the $\overline{\text{RESET}}$ pin is asserted.

When the Port B submode field is relevant in a mode/submode definition, it must not be altered unless the H34 Enable bit in the Port General Control Register is 0. (See Table 4.)

The operation of H3 and H4 and their related status bits is given below, for each of the modes specified by Port General Control Register bis 7 and 6. This description is organized such that for each mode/submode all programmable options of each pin and status bit are given.

Bits 2 and 1 carry the same meaning in each mode/submode, and thus are specified only once.

PBCR

2	H4 Interrupt Enable
0	The H4 interrupt is disabled.
1	The H4 interrupt is enabled.

PBCR

1	H3 SVCRQ Enable
0	The H3 interrupt and DMA request are disabled.
1	The H3 interrupt and DMA request are enabled.

(1) PBCR Mode 0 Port B Submode 00

PBCR

5 4 3	H4 Control
0 X X	Input pin – status only.
1 0 0	Output pin – always negated.
1 0 1	Output pin – always asserted.
1 1 0	Output pin – interlocked input handshake protocol.
1 1 1	Output pin – pulse input handshake protocol.

PBCR

0	H3 Status Control
X	Not used.

(2) PBCR Mode 0 Port B Submode 01

PBCR

5 4 3	H4 Control
0 X X	Input pin – status only.
1 0 0	Output pin – always negated.
1 0 1	Output pin – always asserted.
1 1 0	Output pin – interlocked output handshake protocol.
1 1 1	Output pin – pulse output handshake protocol.

PBCR

0	H3 Status Control
0	The H3S status bit is 1 when either the Port B initial or final output latch can accept new data. It is 0 when both latches are full and cannot accept new data.
1	The H3S status bit is 1 when both of the Port B output latches are empty. It is 0 when at least one latch is full.

(3) PBCR Mode 0 Port B Submode 1X

PBCR

5 4 3	H4 Control
0 X X	Input pin – status only.
1 X 0	Output pin – always negated.
1 X 1	Output pin – always asserted.

PBCR

0	H3 Status Control
X	Not used .

(4) PBCR Mode 1 Port B Submode X0

PBCR

5 4 3	H4 Control
0 X X	Input pin – status only.
1 0 0	Output pin – always negated.
1 0 1	Output pin – always asserted.
1 1 0	Output pin – interlocked input handshake protocol.
1 1 1	Output pin – pulse input handshake protocol.

PBCR

0	H3 Status Control
X	Not used.

(5) PBCR Mode 1 Port B Submode X1

PBCR 5	4	3	H4 Control
0	X	X	Input pin – status only.
1	0	0	Output pin – always negated.
1	0	1	Output pin – always asserted.
1	1	0	Output pin – interlocked output handshake protocol.
1	1	1	Output pin – pulsed output handshake protocol.

PBCR 0	H3 Status Control
0	The H3S status bit is 1 when either the initial or final output latch of Port A and B can accept new data. It is 0 when both latches are full and cannot accept new data.
1	The H3S status bit is 1 when both the initial and final output latches of Port A and B are empty. It is 0 when neither the initial or final latch of Port A and B is full.

(6) PBCR Mode 2

PBCR 5	4	3	H4 Control
X	X	0	Output pin – interlocked input handshake protocol.
X	X	1	Output pin – pulsed input handshake protocol.

PBCR 0	H3 Status Control
X	Not used.

(7) PBCR Mode 3

PBCR 5	4	3	H4 Control
X	X	0	Output pin – interlocked input handshake protocol.
X	X	1	Output pin – pulsed input handshake protocol.

PBCR 0	H3 Status Control
X	Not used.

- **Port A Data Register (PADR)**

The Port A Data Register (PADR) is an address for moving data to and from the Port A pins. The Port A Data Direction Register determines whether each pin is an input (0) or an output (1), and is used in configuring the actual data paths. This is mode dependent and is described with the modes above.

This register is readable and writeable at all times. Depending on the chosen mode/submode, reading or writing may affect the double-buffered handshake mechanism. The Port A Data Register is not affected by the assertion of the $\overline{\text{RESET}}$ pin.

- **Port B Data Register (PBDR)**

The Port B Data Register (PBDR) is an address for moving data to and from the Port B pins. The Port B Data Direction Register determines whether each pin is an input (0) or an output (1), and is used in configuring the actual data paths. This is mode dependent and is described with the modes, above.

This register is readable and writeable at all times. Depending on the chosen mode/submode, reading or writing may affect the double-buffered handshake mechanism. The Port B Data Register is not affected by the assertion of the $\overline{\text{RESET}}$ pin.

- **Port A Alternate Register (PAAR)**

The Port A Alternate Register (PAAR) is an alternate address for reading the Port A pins. It is a read-only address and no other PI/T condition is affected. In all modes and the instantaneous pin level is read and no input latching is performed except at the data bus interface (see Bus Interface Connection.) Writes to this address are answered with $\overline{\text{DTACK}}$, but the data is ignored.

- **Port B Alternate Register (PBAR)**

The Port B Alternate Register (PBAR) is an alternate address for reading the Port B pins. It is a read-only address and no other PI/T condition is affected. In all modes the instantaneous pin level is read and no input laching is performed except at the data bus interface (see Bus Interface Connection). Writes to this address are answered with $\overline{\text{DTACK}}$, but the data is ignored.

- **Port C Data Register (PCDR)**

The Port C Data Register (PCDR) is an address for moving data to and from each of the eight Port C/alternate-function pins. The exact hardware accessed is determined by the type of bus cycle (read or write) and individual conditions affecting each pin. These conditions are (1) whether the pin is used for the Port C or alternate function, and (2) whether the Port C Data Direction Register indicates the input or output direction. The Port C Data Register is single buffered for output pins and not buffered for input pins. These conditions are summarized in Table 3.

The Port C Data Register is not affected by the assertion of the $\overline{\text{RESET}}$ pin.

The operation of the PCDR is independent of the chosen PI/T mode.

Table 3 PCDR Hardware Accesses

Read Port C Data Register			
Port C function PCDDR = 0	Port C function PCDDR = 1	Alternate function PCDDR = 0	Alternate function PCDDR = 1
pin	Port C output register	pin	Port C output register
Write Port C Data Register			
Port C function PCDDR = 0	Port C function PCDDR = 1	Alternate function PCDDR = 0	Alternate function PCDDR = 1
Port C output register, buffer disabled	Port C output register, buffer enabled	Port C output register	Port C output register

Note that two additional useful benefits result from this structure. First, it is possible to directly read the state of a dual-function pin while used for the non-Port C function. Second, it is possible to generate program controlled transitions on alternate-function pins by switching back to the Port C function, and writing to the PCDR.

This register is readable and writeable at all times.

- **Port Status Register (PSR)**

Port Status Register (PSR)

7	6	5	4	3	2	1	0
H4 Level	H3 Level	H2 Level	H1 Level	H4S	H3S	H2S	H1S

The Port Status Register (PSR) contains information about handshake pin activity. Bits 7-4 show the instantaneous level of the respective handshake pin, and is independent of the handshake pin sense bits in the Port General Control Register. Bit 3-0 are the respective, status bits referred to throughout this data sheet. Their interpretation depends on the programmed mode/submode of the PI/T. For Bits 3-0 a 1 is the active or asserted state.

- **Timer Control Register (TCR)**

Timer Control Register (TCR)

7	6	5	4	3	2	1	0
TOUT/$\overline{\text{TIACK}}$ Control			Z.D. Ctrl.	*	Clock Control		Timer Enable

The Timer Control Register (TCR) determines all operations of the timer. Bits 7-5 configure the PC_3/TOUT and $PC_7/\overline{TIACK}$ pins for Port C, square wave, vectored interrupt, or auto-vectored interrupt operation; bit 4 specifies whether the counter receives data from the Counter Preload Register or continues counting when zero detect is reached; bit 3 is unused and is read as 0; bits 2 and 1 configure the path from the CLK and TIN pins to the counter controller; bit 0 enables the timer. This register is readable and writeable at all times.

All bits are cleared to 0 when the $\overline{\text{RESET}}$ pin is asserted.

TCR 7 6 5 — TOUT:$\overline{\text{TIACK}}$ Control

- 0 X X The dual-function pins PC_3/TOUT and $PC_7/\overline{TIACK}$ carry the Port C function.
- 0 1 X The dual-function pin PC_3/TOUT cames the TOUT function. In the run state it is used as a square wave output and is toggled on zero detect. The TOUT pin is high while in the halt state. The dual-function pin $PC_7/\overline{TIACK}$ carries the PC_7 function.
- 1 0 0 The dual-function pin PC_3/TOUT carries the TOUT function. In the run or halt state it is used as a timer interrupt request output. The timer interrupt is disabled; thus, the pin is always three-stated. The dual-function pin $PC_7/\overline{TIACK}$ carries the $\overline{TIACK}$ function; however, since interrupt request is negated, the PI/T produces no response, i.e., no data or $\overline{DTACK}$, to an asserted $\overline{TIACK}$. Refer to Time Interrupt Cycle section for details. This combination and the 101 state below support vectored timer interrupts.
- 1 0 1 The dual-function pin PC_3/TOUT carries the TOUT function and is used as a timer interrupt request output. The timer interrupt is enabled; thus, the pin is low when the timer ZDS status bit is 1. The dual function pin $PC_7/\overline{TIACK}$ carries the $\overline{TIACK}$ function and is used as a timer interrupt acknowledge input. Refer to the Timer Interrupt Acknowledge Cycle section for details. This combination and the 100 state above support vectored timer interrupts.
- 1 1 0 The dual-function pin PC_3/TOUT carries the TOUT function. In the run or halt state it is used as a timer interrupt request output. The timer interrupt is disabled; thus, the pin is always three-stated. The dual-function pin $PC_7/\overline{TIACK}$ carries the PC_7 function.
- 1 1 1 The dual-function pin PC_3/TOUT carries the TOUT function and is used as a timer interrupt request output. The timer interrupt is enabled; thus, then pin is low when the timer ZDS status bits is 1. The dual-function pin $PC_7/\overline{TIACK}$ carries the PC_7 function and autovectored interrupts are supported.

TCR 4 — Zero Detect Control

- 0 The counter is loaded from the Counter Preload Register on the first clock to the 24-bit counter after zero detect, and resumes counting.
- 1 The counter rolls over on zero detect, then continues counting.

Bit 3 is unused and is always read as 0.

TCR 2 1 — Clock Control

- 0 0 The PC2/TIN input pin carries the Port C function and the CLK pin and prescaler are used. The prescaler is decremented on the falling transition of the CLK pin; the 24-bit counter is decremented or loaded from the Counter Preload Register when the prescaler rolls over from $00 to $1F. The Timer Enable bit determines whether the timer is in the run or halt state.
- 0 1 The PC_2/TIN pin serves as a timer input and the CLK pin and prescaler are used. The prescaler is decremented on the falling transition of the CLK pin; the 24-bit counter is decremented or loaded from the Counter Preload Registers when the prescaler rolls over from $00 to $1F. The timer is in the run state when the Timer Enable bit is 1 and the TIN pin is high; otherwise the timer is in the halt state.
- 1 0 The PC_2/TIN pin serves as a timer input and the prescaler is used. The prescaler is decremented following the rising transition of the TIN pin after syncing with the internal clock. The 24-bit counter is decremented or loaded from the counter preload registers when the prescaler rolls over from $00 to $1F. The Timer Enable bit determines whether the timer is in the run or halt state.
- 1 1 The PC_2/TIN pin serves as a timer input and the prescaler is unused. The 24-bit counter is decremented or loaded from the Counter Preload Registers following the rising edge of the TIN pin after syncing with the internal clock. The Timer Enable bit determines whether the timer is in the run or halt state.

TCR 0 — Timer Enable

- 0 Disabled.
- 1 Enabled.

- **Timer Interrupt Vector Register (TIVR)**

The timer interrupt vector register contains the 8-bit vector supplied when the timer interrupt acknowledge pin $\overline{TIACK}$ is asserted. The register is readable and writeable at all times, and the same value is always obtained from a normal read cycle and a timer interrupt acknowledge bus cycle ($\overline{TIACK}$). When the $\overline{\text{RESET}}$ pin is asserted the value of $0F is automatically loaded into the register. Refer to Timer Interrupt Acknowledge Cycle section for more details.

- **Counter Preload Register H,M,L (CPRH–L)**

Counter Preload Register H,M,L (CPRH-L)

7	6	5	4	3	2	1	0	
Bit 23	Bit 22	Bit 21	Bit 20	Bit 19	Bit 18	Bit 17	Bit 16	CPRH
Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	CPRM
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	CPRL

The Counter Preload Registers (CPRH-L) are a group of three 8-bit registers used for storing data to be transferred to the counter. Each of the registers is individually addressable, or the group may be accessed with the MOVEP. L or the MOVEP, W instructions. The address one less than the address of CPRH is the null register, and is reserved so that zeros are read in the upper 8 bits of the destination data register when a MOVEP.L is used. Data written to this address is ignored.

The registers are readable and writeable at all times. A read cycle proceeds independently of any transfer to the counter, which may be occuring simultaneously.

To insure proper operation of the PI/T Timer, a value of $000000 may not be stored in the Counter Preload Registers for use with the counter.

The $\overline{\text{RESET}}$ pin does not affect the contents of these registers.

- **Count Register H,M,L (CNTRH–L)**

Count Register H,M,L (CNTRH-L)

7	6	5	4	3	2	1	0	
Bit 23	Bit 22	Bit 21	Bit 20	Bit 19	Bit 18	Bit 17	Bit 16	CNTRH
Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	CNTRM
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	CNTRL

The count registers (CNTRH-L) are a group of three 8-bit addresses at which the counter can be read. The contents of the counter are not latched during a read bus cycle; thus, the data read at these addresses is not guaranteed if the timer is in the run state. (Bits 2, 1, and 0 of the Timer Control Register specify the state.) Write operations to these addresses result in a normal bus cycle but the data is ignored.

Each of the registers is individually addressable, or the group may be accessed with the MOVEP.L or the MOVEP.W instructions. The address one less than the address of CNTRH is the null register, and is reserved so that zeros are read in the upper 8 bits of the destination data register when a MOVEP.L is used. Data written to this address is ignored.

- **Timer Status Register (TSR)**

Timer Status Register (TSR)

7	6	5	4	3	2	1	0
*	*	*	*	*	*	*	ZDS

The Timer Status Register (TSR) contains one bit from which the zero detect status can be determined. The ZDS status bit (bit 0) is an edge-sensitive flip-flop that is set to 1 when the 24-bit counter decrements from $000001 to $000000. The ZDS status bit is cleared to 0 following the direct clear operation (similar to that of the ports), or when the timer is halted. Note also that when the $\overline{\text{RESET}}$ pin is asserted the timer is disabled, and thus enters the halt state.

This register is always readable without consequence. A write access performs a direct clear operation if bit 0 in the written data is 1. Following that, the ZDS bit is 0.

This register is constructed with a reset dominant S-R flip-flop so that all cleaning conditions prevail over the possible zero detect condition.

Bits 7-1 are unused and are read as 0.

■ PORT CONTROL STRUCTURE

The primary focus of most applications will be on Ports A and B, the handshake pins, the port interrupt pins, and the DMA request pin. They are controlled in the following way: the Port General Control Register contains a 2-bit field that specifies a set of four operation modes. These govern the overall operation of the ports and determine their interrelationships. Some modes require additional information from each port's control register to further define its operation. In each port control register, there is a 2-bit submode field that serves this purpose. Each port mode/submode combination specifies a set of programmable characteristics that fully define the behavior of that port and two of the handshake pins. This structure is summarized in Table 4 and Figure 8.

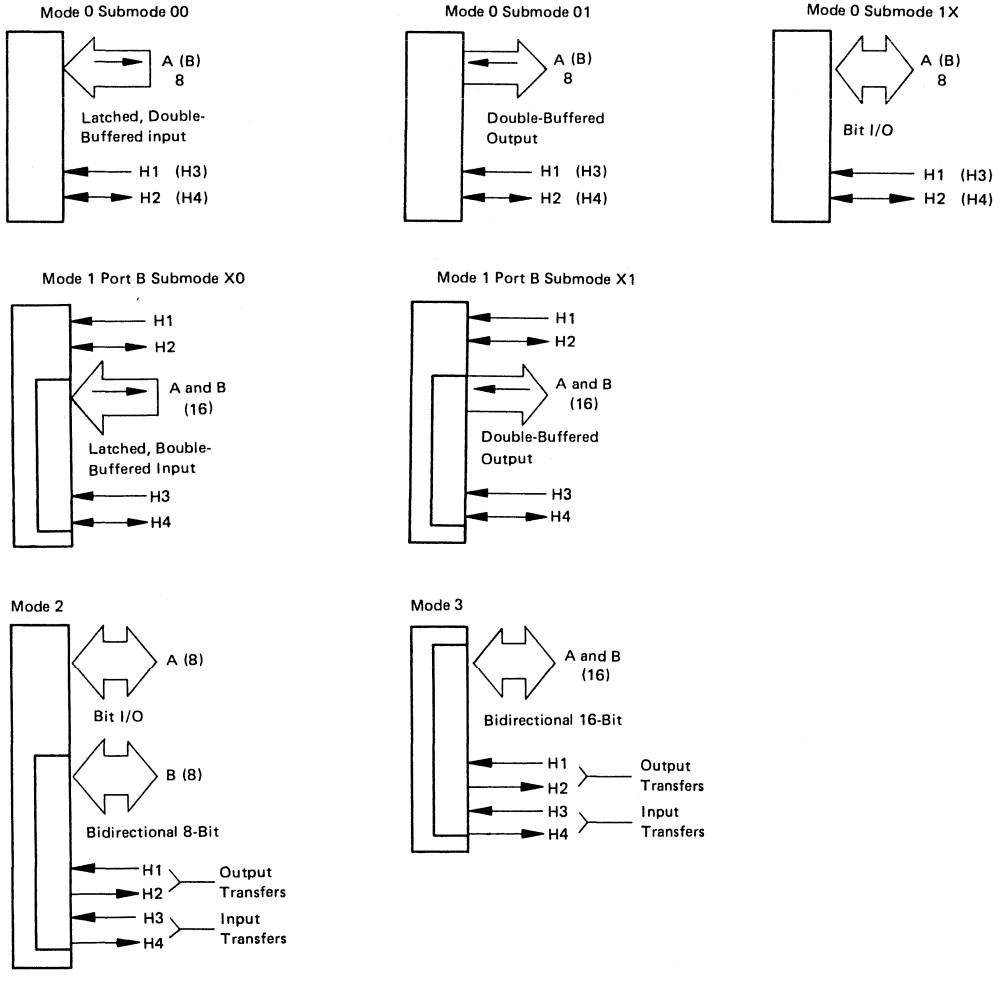


Figure 8 Port Mode Layout

PORT GENERAL INFORMATION AND CONVENTIONS

The following paragraphs introduce concepts that are generally applicable to the PI/T ports independent of the chosen mode and submode. For this reason, no particular port or handshake pins are mentioned; the notation H1 (H3) indicates that, depending on the chosen mode and sub-mode, the statement given may be true for either the H1 or H3 handshake pin.

• Unidirectional vs Bidirectional

Figure 8 shows the configuration of Ports A and B and each of the handshake pins in each port mode and submode. In Modes 0 and 1, a data direction register is associated with each of the ports. These registers contain one bit for each port pin to determine whether that pin is an input or an output. Modes 0 and 1 are, thus, called unidirectional modes because each pin assumes a constant direction, changeable only by a reset condition or a programming change. These modes allow double-buffered data transfers in one direction. This direction, determined by the mode and submode definition, is known as the primary direction. Data transfers in the primary direction are controlled by the handshake pins. Data transfers not in the primary direction are generally unrelated, and single or unbuffered data paths exist.

In Modes 2 and 3 there is no concept of primary direction as in Modes 0 and 1. Except for Port A in Mode 2 (Bit I/O), the data direction registers have no effect. These modes are bidirectional, in that the direction of each transfer (always 8 or 16 bits, double-buffered) is determined dynamically by the state of the handshake pins. Thus, for example, data may be transferred out of the ports, followed very shortly by a transfer into the same port pins. Transfers to and from the ports are independent and may occur in any sequence. Since the instantaneous direction is always determined by the external system, a small amount of arbitration logic may be required.

• Control of Double-Buffered Data Paths

Generally speaking, the PI/T is a double-buffered device. In the primary direction, double-buffering allows orderly transfers by using the handshake pins in any of several programmable protocols. (When Bit I/O is used, double-buffering is not available and the handshake pins are used as outputs or status/interrupt inputs.)

Use of double-buffering is most beneficial in situations where a peripheral device and the computer system are capable of transferring data at roughly the same speed. Double-buffering allows the fetch operation of the data transmitter to be overlapped with the store operation of the data receiver. Thus, throughput measured in bytes or words-per-second may be greatly enhanced. If there is a large mismatch in transfer capability between the computer and the peripheral, little or no benefit is obtained. In these cases there is no penalty in using double-buffering.

• Double-Buffered Input Transfers

In all modes, the PI/T supports double-buffered input transfers. Data that meets the port setup and hold times is latched on the asserted edge of H1(H3). H1(H3) is edge-sensitive, and may assume any duty-cycle as long as both high and low minimum times are observed. The PI/T contains a Port Status Register whose H1S(H3S) status bit is set anytime any input data is present in the double-buffered latches that has not been read by the bus master. The action of H2(H4) is programmable; it may indicate whether there is room for more data in the PI/T latches or it may serve other purposes. The

Table 4 Port Mode Control Summary

Mode 0 (Unidirectional 8-Bit mode)	Mode 1 (Unidirectional 16-Bit mode)	Mode 2 (Bidirectional 8-Bit mode)	Mode 3 (Bidirectional 16-Bit mode)
Port A	Port A – Double-Buffered Data (Most significant)	Port A – Bit I/O (with no handshaking pins)	Port A – Double-Buffered Data (Most significant)
Submode 00 – Double-Buffered Input H1 – Latches input data H2 – Status/interrupt generating input, general-purpose output, or operation with H1 in the interlocked or pulsed input handshake protocols Submode 01 – Double-Buffered Output H1 – Indicates data received by peripheral H2 – Status/interrupt generating input, general-purpose output, or operation with H1 in the interlocked or pulsed output handshake protocols. Submode 1X – Bit I/O H1 – Status/interrupt generating input H2 – Status/interrupt generating input or general-purpose output	Submode XX (not used) H1 – Status/interrupt generating input H2 – Status/interrupt generating input or general-purpose output	Submode XX (not used)	Submode XX (not used)
Port B	Port B – Double-Buffered Data (Least significant)	Port B – Bidirectional 8-Bit Data (Double-Buffered)	Port B – Double-Buffered Data (Least significant)
H3 and H4 – Identical to Port A, H1 and H2	Submode X0 – Unidirectional 16-Bit Input H3 – Latches input data H4 – Status/interrupt generating input, general-purpose output, or operation with H3 in the interlocked or pulsed input handshake protocols Submode X1 – Unidirectional 16-Bit Output H3 – Indicates data received by peripheral H4 – Status/interrupt generating input, general-purpose output, or operation with H3 in the interlocked or pulsed output handshake protocols	Submode XX (not used) H1 – indicates output data received by peripheral H2 – Operation with H1 in the interlocked or pulsed output handshake protocols H3 – Latches input data H4 – Operation with H3 in the interlocked or pulsed input handshake protocols	Submode XX (not used) H1 – Indicates output data received by peripheral H2 – Operation with H1 in the interlocked or pulsed output handshake protocols H3 – Latches input data H4 – Operation with H3 in the interlocked or pulsed input handshake protocols

following options are available, depending on the mode.

(1) H2(H4) may be an edge-sensitive input that is independent of H1(H3) and the transfer of port data. On the asserted edge of H2(H4), the H2S(H4S) status bit is set. It is cleared by the direct method (refer to Direct Method of Resetting Status), the $\overline{\text{RESET}}$ pin being asserted, or when the H12 Enable (H34 Enable) bit of the Port General Control Register is 0.

(2) H2(H4) may be a general purpose output pin that is always negated. The H2S(H4S) status bit is always 0.

(3) H2(H4) may be a general purpose output pin that is always asserted. The H2S(H4S) status bit is always 0.

(4) H2(H4) may be an output pin in the interlocked input handshake protocol. It is asserted when the port input latches are ready to accept new data. It is negated asynchronously following the asserted edge of the H1(H3) input.

As soon as the input latches become ready, H2(H4) is again asserted. When the input double-buffered latches are full, H2(H4) remains negated until data is removed. Thus, anytime the H2(H4) output is asserted, new input data may be entered by asserting H1(H3). At other times transitions on H1(H3) are ignored. The H2S(H4S) status bit is always 0. When H12 Enable (H34 Enable) is 0, H2(H4) is held negated.

(5) H2(H4) may be an output pin in the pulsed input handshake protocol. It is asserted exactly as in the interlocked input protocol, but never remains asserted longer than 4 clock cycles. Typically, a four clock cycle pulse is generated. But in the case that a subsequent H1(H3) asserted edge occurs before termination of the pulse, H2(H4) is negated asynchronously. Thus, anytime after the leading edge of the H2(H4) pulse, new data may be entered in the PI/T double-buffered input latches. The H2S(H4S) status bit is always 0. When H12 Enable (H34 Enable) is 0, H2(H4) is held negated.

A sample timing diagram is shown in Figure 9. The H2(H4) interlocked and pulsed input handshake protocols are shown. The $\overline{\text{DMAREQ}}$ pin is also shown assuming it is enabled. All handshake pin sense bits are assumed to be 0 (refer to Port General Control Register); thus, the pins are in the low state when asserted. Due to the great similarity between modes, this timing diagram is applicable to all double-buffered input transfers.

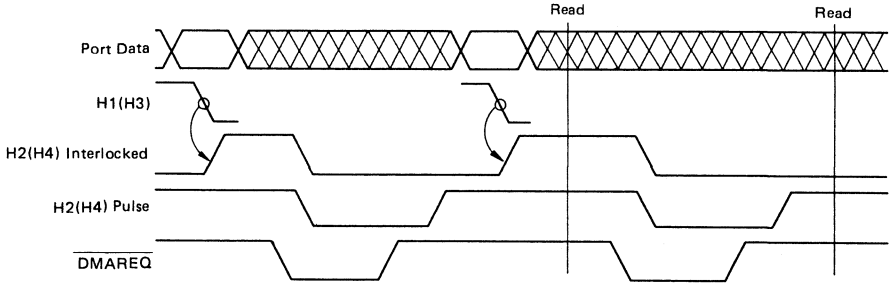


Figure 9 Double-buffered Input Transfers

- **Double-Buffered Output Transfers**

The PI/T supports double-buffered output transfers in all modes. Data, written by the bus master to the PI/T, is stored in the port's output latch. The peripheral accepts the data by asserting H1(H3), which causes the next data to be moved to the port's output latch as soon as it is available. The function of H2(H4) is programmable; it may indicate whether new data has been moved to the output latch or it may serve other purposes. The H1S(H3S) status bit may be programmed for two interpretations. Normally the status bit is a 1 when there is at least one latch in the double-buffered data path that can accept new data. After writing one byte/word of data to the ports, an interrupt service routine could check this bit to determine if it could store another byte/word; thus, filling both latches. When the bus master is finished, it is often useful to be able to check whether all of the data has been transferred to the peripheral. The H1S(H3S) Status Control bit of the Port A and B Control Registers provide this flexibility. The programmable options of the H2(H4) pin are given below, depending on the mode.

(1) H2(H4) may be an edge-sensitive input pin independent of H1(H3) and the transfer of port data. On the asserted edge of H2(H4), the H2S(H4S) status bit is set. It is reset by the direct method (refer to Direct Method of Resetting Status), the $\overline{\text{RESET}}$ pin being asserted, or when the H12 Enable (H34 Enable) bit of the Port General Control Register is 0.

(2) H2(H4) may be a general-purpose output pin that is always negated. The H2S(H4S) status bit is always 0.

(3) H2(H4) may be a general-purpose output pin that is always asserted. The H2S(H4S) status bit is always 0.

(4) H2(H4) may be an output pin in the interlocked output handshake protocol. H2(H4) is asserted two clock cycles after data is transferred to the double-buffered output latches. The data remains stable and H2(H4) remains asserted until the next asserted edge of the H1(H3) input. At that time, H2(H4) is asynchronously negated. As soon as the next data is available, it is transferred to the output latches. When H2(H4) is negated, asserted transitions on H1(H3) have no effect on the data paths. As is explained later, however, in Modes 2 and 3 they do control the three-state output buffers of the bidirectional port(s). The H2S(H4S) status bit is always 0. When H12 Enable (H34 Enable) is 0, H2(H4) is held negated.

(5) H2(H4) may be an output pin in the pulsed output handshake protocol. It is asserted exactly as in the interlocked output protocol above, but never remains asserted longer than four clock cycles. Typically, a four clock pulse is

generated. But in the case that a subsequent H1(H3) asserted edge occurs before termination of the pulse, H2(H4) is negated asynchronously shortening the pulse. The H2S(H4S) status bit is always 0. When H12 Enable (H34 Enable) is 0 H2(H4) is held negated.

A sample timing diagram is shown in Figure 10. The H2(H4) interlocked and pulsed output handshake protocols are shown. The $\overline{\text{DMAREQ}}$ pin is also shown assuming it is enabled. All handshake pin sense bits are assumed to be 0; thus, the pins are in the low state when asserted. Due to the great similarity between modes, this timing diagram is applicable to all double-buffered output transfer.

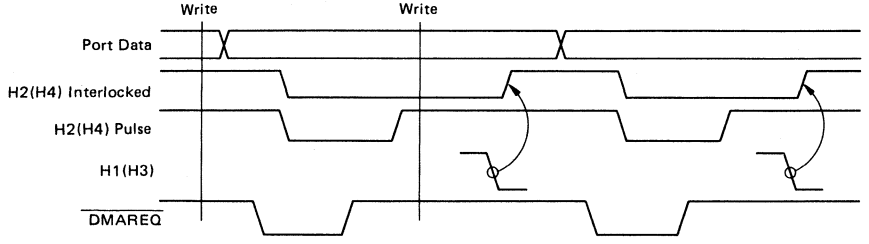


Figure 10 Double-buffered Output Transfers

- **Requesting Bus Master Service**

The PI/T has several means of indicating a need for service by a bus master. First, the processor may poll the Port Status Register. It contains a status bit for each handshake pin, plus a level bit that always reflects the instantaneous state of that handshake pin. A status bit is 1 when the PI/T needs servicing, i.e., generally when the bus master needs to read or write data to the ports, or when a handshake pin used as a simple status input has been asserted. The interpretation of these bits is dependent on the chosen mode and submode.

Second, the PI/T may be placed in the processor's interrupt structure. As mentioned previously, the PI/T contains Port A and B Control Registers that configure the handshake pins. Other bits in these registers enable an interrupt associated with each handshake pin. This interrupt is made available through the PC5/$\overline{\text{PIRQ}}$ pin, if the $\overline{\text{PIRQ}}$ function is selected. Three additional conditions are required for $\overline{\text{PIRQ}}$ to be asserted: (1) the handshake pin status bit set, (2) the corresponding interrupt (service request) enable bit is set, (3) and DMA requests are not associated with that data transfer (H1 and H3 only). The conditions from each of the four handshake pins and corresponding status bits are ORed to determine $\overline{\text{PIRQ}}$.

The third method of requesting service is via the PC4/$\overline{\text{DMAREQ}}$ pin. This pin can be associated with double-buffered transfers in each mode. If it is used as a DMA controller request, it can initiate requests to keep the PI/T's input/output double-buffering empty/full as much as possible. It will not overrun the DMA controller. The pin is compatible with the HD68450 Direct Memory Access Controller (DMAC).

- **Vectored, Prioritized Port Interrupts**

Use of HD68000-compatible vectored interrupts with the PI/T requires the $\overline{\text{PIRQ}}$ and $\overline{\text{PIACK}}$ pins. When $\overline{\text{PIACK}}$ is asserted, the PI/T places an 8-bit vector on the data pins D_0-D_7. Under normal conditions, this vector corresponds to highest priority, enabled, active port interrupt source with which the $\overline{\text{DMAREQ}}$ pin is not currently associated. The most-significant six bits are provided by the Port Interrupt Vector Register (PIVR), with the lower two bits supplied by prioritization logic according to conditions present when $\overline{\text{PIACK}}$ is asserted. It is important to note that the only affect on the PI/T caused by interrupt acknowledge cycles is that the vector is placed on the data bus. Specifically, no registers, data, status, or other internal states of the PI/T are affected by the cycle.

Several conditions may be present when the $\overline{\text{PIACK}}$ input is asserted to the PI/T. These conditions affect the PI/T's response and the termination of the bus cycle. If the PI/T has no interrupt function selected, or is not asserting $\overline{\text{PIRQ}}$, the PI/T will make no response to $\overline{\text{PIACK}}$ ($\overline{\text{DTACK}}$ will not be asserted). If the PI/T is asserting $\overline{\text{PIRQ}}$ when $\overline{\text{PIACK}}$ is received, the PI/T will output the contents of the Port Interrupt Vector Register and the prioritization bits. If the PIVR has not been initialized, $0F will be read from this register. These conditions are summarized in Table 5.

Table 5 Response to Port Interrupt Acknowledge

Conditions	$\overline{\text{PIRQ}}$ negated OR interrupt request function not selected	$\overline{\text{PIRQ}}$ asserted
PIVR has not been initialized since $\overline{\text{RESET}}$	No response from PI/T. No $\overline{\text{DTACK}}$.	PI/T provides $0F, the Uninitialized Vector.*
PIVR has been initialized since $\overline{\text{RESET}}$	No response from PI/T. No $\overline{\text{DTACK}}$.	PI/T provides PIVR contents with prioritization bits.

*The uninitialized vector is the value returned from an interrupt vector register before it has been initialized.

The vector table entries for the PI/T appear as a contiguous block of four vector numbers whose common upper six bits are programmed in the PIVR. The following table pairs each interrupt source with the 2-bit value provided by the prioriza-tion logic, when interrupt acknowledge is asserted.

H1 source – 00
H2 source – 01
H3 source – 10
H4 source – 11

- **Autovectored Port Interrupts**

Autovectored interrupts use only the $\overline{\text{PIRQ}}$ pin. The operation of the PI/T with vectored and autovectored interrupts is identical except that no vectors are supplied and the PC6/$\overline{\text{PIACK}}$ pin can be used as a Port C pin.

- **Direct Method of Resetting Status**

In certain modes one or more handshake pins can be used as edge-sensitive inputs for sole purpose of setting bits in the Port Status Register. These bits consist of simple flip-flops. They are set (to 1) by the occurrence of the asserted edge of the handshake pin input. Resetting a handshake status bit can be done by writing an 8-bit mask to the Port Status Register. This is called the direct method of resetting. To reset a status bit that is resettable by the direct method, the mask must contain a 1 in the bit position of the Port Status Register corresponding to the desired bit. Other positions must contain 0's. For status bits that are not resettable by the direct method in the chosen mode, the data written to the port status register has no effect. For status bits that are resettable by the direct method in the chosen mode, a 0 in the mask has no effect.

- **Handshake Pin Sense Control**

The PI/T contains exclusive-OR gates to control the sense of each of the handshake pins, whether used as inputs or outputs. Four bits in the Port General Control Register may be programmed to determine whether the pins are asserted in the low or high voltage state. As with other control registers, these bits are reset to 0 when the $\overline{\text{RESET}}$ pin is asserted, defaulting the asserted level to be low.

- **Enabling Ports A and B**

Certain functions involved with double-buffered data transfers, the handshake pins, and the status bits, may be disabled by the external system or by the programmer during initialization. The Port General Control Register contains two bits, H12 Enable and H34 Enable, which control these functions. These bits are cleared to the 0 state when the $\overline{\text{RESET}}$ pin is asserted, and the functions are disabled. The functions are the following.

(1) Independent of other actions by the bus master or peripheral (via the handshake pins), the PI/T's disabled handshake controller is held to the "empty" state, i.e., no data is present in the double-buffered data path.
(2) When any handshake pin is used to set a simple status flip-flop, unrelated to double-buffered transfers, these flip-flops are held reset to 0. (See Table 4.)
(3) When H2(H4) is used in an interlocked or pulsed handshake with H1(H3), H2(H4) is held negated, regardless of the chosen mode, submode, and primary direction. Thus, for double-buffered input transfers, the programmer may signal a peripheral when the PI/T is ready to begin transfers by setting the associated handshake enable bit to 1.

- **The Port A and B Alternate Registers**

In addition to the Port A and B Data Registers, the PI/T contains Port A and B Alternate Registers. These registers are read-only, and simply provide the instantaneous level of each port pin. They have no effect on the operation of the handshake pins, double-buffered transfers, status bits, or any other aspect of the PI/T, and they are mode/submode independent.

■ PORT MODES

This section contains information that distinguishes the various port modes and submodes.

- **Mode 0 – Unidirectional 8-Bit Mode**

In Mode 0, Ports A and B operate independently. Each may be configured in any of its three possible submodes:

Submode 00 – Double-Buffered Input
Submode 01 – Double-Buffered Output
Submode 1X – Bit I/O

Handshake pins H1 and H2 are associated with Port A and configured by programming the Port A Control Register. (The H12 Enable bit of the Port General Control Register, enables Port A transfers.) Handshake pins H3 and H4 are associated with Port B and configured by programming the Port B Control Register. (The H34 Enable bit of the Port General Control Register enables Port B transfers.) The Port A and B Data Direction Registers operate in all three submodes. Along with the submode, they affect the data read and written at the associated data register according to Table 6. They also enable the output buffer associated with each port pin. The $\overline{\text{DMAREQ}}$ pin may be associated with either (not both) Port A or Port B, but does not function if the Bit I/O submode is programmed for the chosen port.

Table 6 Mode 0 Port Data Paths

Mode	Read Port A/B Data Register		Write Port A/B Data Register	
	DDR = 0	DDR = 1	DDR = X	
0 Submode 00	FIL. D.B.	FOL Note 3	FOL. S.B.	Note 1
0 Submode 01	Pin	FOL Note 3	IOL/FOL. D.B.	Note 2
0 Submode 1X	Pin	FOL Note 3	FOL. S.B.	Note 1

Abbreviations:
IOL – Initial Output Latch
FOL – Final Output Latch
FIL – Final Input Latch
S.B. – Single Buffered
D.B. – Double Buffered
DDR – Data Direction Register

Note 1: Data is latched in the output data registers (final output latch) and will be single buffered at the pin if the DDR is 1. The output buffers will be turned off if the DDR is 0.
Note 2: Data is latched in the double-buffered output data registers. The data in the final output latch will appear on the port pin if the DDR is a 1.
Note 3: The output drivers that connect the final output latch to the pins are turned on.

(1) Port A or B Submode 00 (8-Bit Double-Buffered Input)

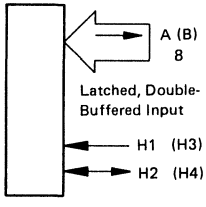


Figure 11 Mode 0 Submode 00

In Mode 0, double-buffered input transfers of up to 8-bits are available by programming Submode 00 in the desired port's control register. The operation of H2 and H4 may be selected by programming the Port A and Port B Control Registers, respectively. All five double-buffered input handshake options, previously mentioned in the Port General Information and Conventions section, are available.

For pins used as outputs, the data path consists of single latch driving the output buffer. Data written to the port's data register does not affect the operation of any handshake pin, status bit, or any other aspect of the PI/T. Output pins may be used independently of the input transfer. However, read bus cycles to the data register do remove data from the port. Therefore, care should be taken to avoid processor instructions that perform unwanted read cycles.

Refer to PARALLEL PORTS Double-Buffered Input Transfers for a sample timing diagram. (Figure 9)

(2) Port A or B Submode 01 (8-Bit Double-Buffered Output)

In Mode 0, double-buffered output transfers of up to 8 bits are available by programming submode 01 in the desired port's control register. The operation of H2 and H4 may be selected by programming the Port A and Port B Control Registers, respectively. All five double-buffered output handshake options, previously mentioned in the Port General Information and Conventions section, are available.

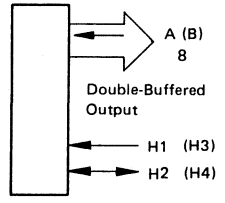


Figure 12 Mode 0 Submode 01

For pins used as inputs, data written to the associated data register is double-buffered and passed to the initial or final output latch, as usual, but the output buffer is disabled.

Refer to PARALLEL PORTS Double-Buffered Output Transfers for a sample timing diagram (Figure 10)

(3) Port A or B Submode 1X (Bit I/O)

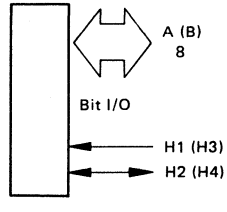


Figure 13 Mode 0 Submode 1X

In Mode 0, simple Bit I/O is available by programming Submode 1X in the desired port's control register. This submode is intended for applications in which several independent devices must be controlled or monitored. Data written to the

associated data register is single-buffered. If the data direction register bit for that pin is a 1 (output), the output buffer is enabled. If it is 0 (input), data written is still latched, but is not available at the pin. Data read from the data register is the instantaneous value of the pin or what was written to the data register, depending on the contents of the data direction register. H1(H3) is an edge-sensitive status input pin only and it controls no data-related function. The H1S(H3S) status bit is set following the asserted edge of the input waveform. It is reset by the direct method, the $\overline{\text{RESET}}$ pin being asserted, or when the H12 Enable (H34 Enable) bit is 0.

H2(H4) can be programmed as a simple status input (identical to H1(H3)), or as an asserted or negated output. The interlocked or pulsed handshake configurations are not available.

- **Mode 1 – Unidirectional 16-Bit Mode**

In Mode 1, Ports A and B are concatenated to form a single 16-bit port. The Port B Submode field controls the configuration of both ports. The possible submodes are:

Port B Submode X0 – Double-Buffered Input
Port B Submode X1 – Double-Buffered Output

Handshake pins H3 and H4, configured by programming the Port B Control Register, are associated with the 16-bit double-buffered transfer. These 16-bit transfers, are enabled by the H34 Enable bit of the Port General Control Register. Handshake pins H1 and H2 may be used as simple status inputs not related to the 16-bit data transfer or H2 may be an output. Enabling of the H1 and H2 handshake pins is done by the H12 Enable bit of the Port General Control Register. The Port A and B Data Direction Registers operate in each sub-mode. Along with the submode, they affect the data read and written at the data register according to Table 7. They also enable the output buffer associated with each port pin. The $\overline{\text{DMAREQ}}$ pin may be associated only with H3.

Mode 1 can provide convenient, high-speed 16-bit transfers. The Port A and B data registers are addressed for compatibility with the HD68000 Move Peripheral (MOVEP) instruction and with the HD68450 DMAC. To take advantage of this, Port A should contain the most-significant byte of data and always be read or written by the bus master first. The interlocked and pulsed handshake protocols are keyed to accesses to the Port B Data Register in Mode 1. If it is accessed last, the 16-bit double-buffered transfers proceed smoothly.

Table 7 Mode 1 Port Data Paths

Mode	Read Port A/B Register		Write Port A/B Register	
	DDR = 0	DDR = 1	DDR = 0	DDR = 1
1, Port B Submode X0	FIL. D.B.	FOL Note 3	FOL, S.B. Note 2	FOL, S.B. Note 2
1, Port B Submode X1	Pin	FOL Note 3	IOL/FOL. D.B., Note 1	IOL/FOL. D.B., Note 1

Note 1: Data written to Port A goes to a temporary latch. When the Port B data register is later written, Port A data is transferred to IOL/FOL.
Note 2: Data is latched in the output data registers (final output latch) and will be single buffered at the pin if the DDR is 1. The output buffers will be turned off if the DDR is 0.
Note 3: The output drivers that connect the final output latch to the pins are turned on.

Abbreviations:
IOL – Initial Output Latch
FOL – Final Output Latch
FIL – Final Input Latch
S.B. – Single Buffered
D.B. – Double Buffered
DDR – Data Direction Register

(1) Port B Submode X0 (16-Bit Double-Buffered Input)

In Mode 1 Port B Submode X0, double-buffered input transfers of up to 16 bits may be obtained. The level of all 16 pins is asynchronously latched with the asserted edge of H3. The processor may check H3S status bit to determine if new data is present. The $\overline{\text{DMAREQ}}$ pin may be used to signal a DMA controller to empty the input buffers. Regardless of the bus master, Port A data should be read first. (Actually, Port A data need not be read at all.) Port B data should be read last. The operation of the internal handshake controller, the H3S bit, and $\overline{\text{DMAREQ}}$ are keyed to the reading of the Port B data register. (The HD68450 DMAC can be programmed to perform the exact transfers needed for compatibility with the PI/T.) H4 may be programmed for all five of the handshake options mentioned in the Port General Information and Conventions section.

For pins used as outputs, the data path consists of a single latch driving the output buffer. Data written to the port's data register does not affect the operation of any handshake pin, status bit, or any other aspect of the PI/T. Thus, output pins may be used independently of the input transfer. However, read bus cycles to the Port B Data Register do remove

data, so care should be taken to avoid unwanted read cycles.

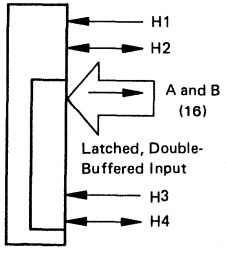


Figure 14 Mode 1 Port B Submode X0

(2) Port B Submode X1 (16-Bit Double-Buffered Output)

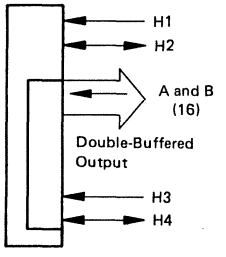


Figure 15 Mode 1 Port B Submode X1

Refer to PARALLEL PORTS Double-Buffered Input Transfers for a sample timing diagram (Figure 9).
In Mode 1 Port B Submode X1, double-buffered output transfers of up to 16 bits may be obtained. Data is written by the bus master (processor or DMA controller) in two bytes. The first byte (most-significant) is written to the Port A Data Register. It is stored in a temporary latch until the next byte is written to the Port B Data Register. Then all 16 bits are transferred to the final output latches of Ports A and B. Both options for interpretation of the H3S status bit, mentioned in Port General Information and Comments section, are available and apply to the 16-bit port as a whole. The $\overline{\text{DMAREQ}}$ pin may be used to signal a DMA controller to transfer another word to the port output latches. (The HD68450 DMAC can be programmed to perform the exact transfers needed for compatibility with the PI/T.) H4 may be programmed for all five of the handshake options mentioned in Port General Information and Comments section.

For pins used as inputs, data written to either data register is double-buffered and passed to the initial or final output latch, as usual, but the output buffer is disabled.
Refer to PARALLEL PORTS Double-Buffered Input/Output Transfer for a sample timing diagram (Figure 10).

- **Mode 2 — Bidirectional 8-Bit Mode**

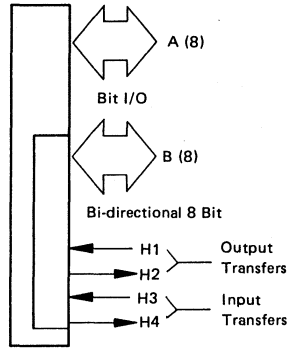


Figure 16 Mode 2

In Mode 2, Port A is used for simple bit I/O with no associated handshake pins. Port B is used for bidirectional 8-bit double-buffered transfers. H1 and H2, enabled by the H12 Enable bit in the Port General Control Register, control output transfers, while H3 and H4, enabled by the Port General Control Register bit H34 Enable, control input transfers. The instantaneous direction of the data is determined by the H1 handshake pin. The Port B Data Direction Register is not used. The Port A and Port B submode fields do not affect PI/T operation in Mode 2.
(1) Double-Buffered I/O (Port B)
The only aspect of bidirectional double-buffered transfers that differs from the uni-directional modes lies in controlling the Port B output buffers. They are controlled by the level of H1. When H1 is negated, the Port B output buffers (all 8) are enabled and the pins drive the bidirectional bus. Generally, H1 is negated in response to an asserted H2, which indicates that new output data is present in the double-buffered latches. Following acceptance of the data, the peripheral asserts H1, disabling the Port B output buffers. Other than controlling the output buffer, H1 is edge-sensitive as in other modes. Input transfers proceed identically to the double-buffered input protocol described in the Port General Information and Conventions Section. In Mode 2, only the interlocked and pulsed

Table 8 Mode 2 Port B Data Paths

Mode	Read Port B Data Register	Write Port B Data Register
2	FIL, D.B.	IOL/FOL, D.B.

Abbreviations:
IOL — Initial Output Latch
FOL — Final Output Latch
D.B. — Double Buffered
FIL — Final Input Latch

handshake pin options are available on H2 and H4. The $\overline{\text{DMAREQ}}$ pin may be associated with either input transfers (H3) or output transfers (H1), but not both. Refer to Table 8 for a summary of the Port B Data Register responses in Mode 2.

(2) Bit I/O (Port A)

Mode 2, Port A performs simple bit I/O with no associated handshake pins. This configuration is intended for applications in which several independent devices must be controlled or monitored. Data written to the Port A data register is single-buffered. If the Port A Data Direction Register bit for that pin is 1 (output), the output buffer is enabled. If it is 0, data written is still latched but not available at the pin. Data read from the data register is either the instantaneous value of the pin or what was written to the data register, depending on the contents of the Port A Data Direction Register. This is summarized in Table 9.

Table 9 Mode 2 Port A Data Paths

Mode	Read Port A Data Register		Write Port A Data Register	
	DDR = 0	DDR = 1	DDR = 0	DDR = 1
2	Pin	FOL	FOL	FOL, S.B.

Abbreviations:

S.B. – Single Buffered
FOL – Final Output Latch
DDR – Data Direction Register

- **Mode 3 – Bidirectional 16-Bit Double Buffered I/O**

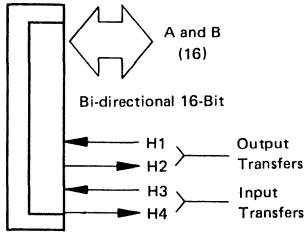


Figure 17 Mode 3

In Mode 3, Ports A and B are used for bidirectional 16-bit double-buffered transfers. H1 and H2 control output transfers, while H3 and H4 control input transfers. (H1 and H2 are enabled by the H12 Enable bit while H3 and H4 are enabled by the H34 Enable bit of the Port General Control Register.) The instantaneous direction of the data is determined by the H1 handshake pin, and thus, the data direction registers are not used. The Port A and Port B submode fields do not affect PI/T operation in Mode 3.

The only aspect of bidirectional double-buffered transfers that differs from the unidirectional modes lies in controlling the Port A and B output buffers. They are controlled by the level of H1. When H1 is negated, the output buffers (all 16) are enabled and the pins drive the bidirectional bus. Generally, H1 is negated in response to an asserted H2, which indicates that new output data is present in the double-buffered latches. Following acceptance of the data, the peripheral asserts H1, disabling the output buffers. Other than controlling the output buffers, H1 is edge-sensitive as in other modes. Input transfers proceed identically to the double-buffered input protocol described in the Port General Information and Conventions section. Port A and B data is latched with the asserted edge of H3. In Mode 3, only the interlocked and pulsed handshake pin options are available to H2 and H4. The $\overline{\text{DMAREQ}}$ pin may be associated with either input transfers (H3) or output transfers (H1), but not both. H2 indicates when new data is available in the Port B (and implicitly Port A) output latches, but unless the buffer is enabled by H1, the data is not driving the pins.

Mode 3 can provide convenient high-speed 16-bit transfers. The Port A and B Data Registers are addressed for compatibility with the HD68000's Move Peripheral (MOVEP) instruction and with the HD68450 DMAC. To take advantage of this, Port A should contain the most-significant data and always be read or written by the bus master first. The interlocked and pulsed handshake protocols are keyed to accesses to the Port B Data Register in Mode 3. If it is accessed last, the 16-bit double-buffered transfer proceed smoothly. Refer to Table 10 for a summary of the Port A and B data paths in Mode 3.

■ DMA REQUEST OPERATION

The Direct Memory Access Request ($\overline{\text{DMAREQ}}$) pulse can be associated with output or input transfers to keep the initial and final output latches full or initial and final input latches empty respectively. Figure 18 and 19 show all the possible paths in generating DMA requests.

Table 10 Mode 3 Port A and B Data Paths

Mode	Read Port A and B Data Register	Write Port A and B Data Register
3	FIL, D.B.	IOL/FOL. D.B., Note 1

Note 1: Data written to Port A goes to a temporary latch. When the Port B data register is later written. Port A data is transferred to IOL/FOL.

Abbreviations:

IOL – Initial Output Latch
FOL – Final Output Latch
FIL – Final Input Latch
S.B. – Single Buffered
D.B. – Double Buffered

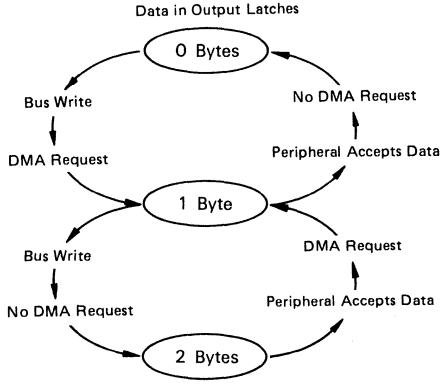


Figure 18 $\overline{\text{DMAREQ}}$ Associated with Output Transfers

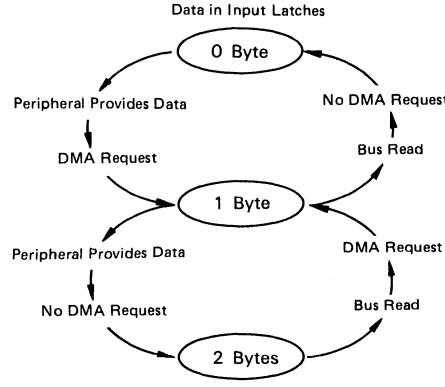


Figure 19 $\overline{\text{DMAREQ}}$ Associated with Input Transfers

■ TIMER

The HD68230 timer can provide several facilities needed by HD68000 operating systems. It can generate periodic interrupts, a square wave, or a single interrupt after a programmed time period. Also, it can be used for elapsed time measurement or as a device watchdog. This section describes the programmable options available, capabilities, and restrictions that apply to the timer.

The PI/T timer contains a 24-bit synchronous down counter that is loaded from three 8-bit Counter Preload Registers. The 24-bit counter may be clocked by the output of a 5-bit (divide-by-32) prescaler or by an external timer input TIN. If the prescaler is used, it may be clocked by the system clock (CLK pin) or by the TIN external input. The counter signals the occurrence of an event primarily through zero detection. (A zero is when the counter of the 24-bit timer is equal to zero.) This sets the zero detect status (ZDS) bit in the Timer Status Register. It may be checked by the processor or may be used to generate a timer interrupt. The ZDS bit is reset by writing a 1 to the Timer Status Register in that bit position.

The general operation of the timer is flexible and easily programmable. The timer is fully configured and controlled by programming the 8-bit Timer Control Register. It controls: (1) the choice between the Port C operation and the timer operation of three timer pins,
(2) whether the counter is loaded from the Counter Preload Register or rolls over when zero detect is reach,
(3) the clock input,
(4) whether the prescaler is used, and
(5) whether the timer is enabled.

• RUN/HALT Definition

The overall operation of the timer is described in terms of the run or halt states. The control of the current state is determined by programming the Timer Control Register. When in the halt state, all of the following occur.
(1) The prior contents of the counter is not altered and is reliably readable via the Count Registers.
(2) The prescaler is forced to $1F whether or not it is used.
(3) The ZDS status bit is forced to 0, regardless of the possible zero contents of the 24-bit counter.

The run state is characterized by:
(1) The counter is clocked by the source programmed in the Timer Control Register.
(2) The counter is not reliably readable.
(3) The prescaler is allowed to decrement if programmed for use.
(4) The ZDS status bit is set when the 24-bit counter transitions from $000001 to $000000.

- **Timer Rules**

This section provides a set of rules that allow easy application of the timer.

(1) When the $\overline{\text{RESET}}$ pin is asserted, all bits of the Timer Control Register go to 0, configuring the dual function pins as Port C inputs.
(2) The contents of the Counter Preload Registers and counter are not affected by the $\overline{\text{RESET}}$ pin.
(3) The Count Registers provide a direct read data path from each portion of the 24-bit counter, but data written to their addresses is ignored. (This results in a normal bus cycle.) These registers are readable at any time, but their contents are never latched. Unreliable data may be read when the timer is in the run state.
(4) The Counter Preload Registers are readable and writable at any time and this occurs independently of any timer operation. No protection mechanisms are provided against ill-timed writes.
(5) The input frequency to the 24-bit counter from the TIN pin or prescaler output, must be between 0 and the input frequency at CLK pin divided by 32 regardless of the configuration chosen.
(6) For configurations in which the prescaler is used (with the CLK pin or TIN pin as an input), the contents of the Counter Preload Register (CPR) is transferred to the counter the first time that the prescaler passes from $00 to $1F (rolls over) after entering the run state. Thereafter, the counter decrements or is loaded from the Counter Preload Register when the prescaler rolls over.
(7) For configurations in which the prescaler is not used, the contents of the Counter Preload Registers are transferred to the counter on the first asserted edge of the TIN input after entering the run state. On subsequent asserted edges the counter decrements or is loaded from the Counter Preload Registers.
(8) The lowest value allowed in the Counter Preload Register for use with the counter is $000001.

- **Timer Interrupt Acknowledge Cycles**

Several conditions may be present when the timer interrupt acknowledge pin ($\overline{\text{TIACK}}$) is asserted. These conditions affect the PI/T's response and the termination of the bus cycle. (see Table 11)

Table 11 Response to Timer Interrupt Acknowledge

PC3/TOUT Function	Response to Asserted $\overline{\text{TIACK}}$
PC3 – Port C Pin	No response. No $\overline{\text{DTACK}}$.
TOUT – Square Wave	No response. No $\overline{\text{DTACK}}$.
TOUT – Negated Timer Interrupt Request	No response. No $\overline{\text{DTACK}}$.
TOUT – Asserted Timer Interrupt Request	Timer Interrupt Vector Contents. $\overline{\text{DTACK}}$ Asserted.

- **TIMER APPLICATIONS SUMMARY**

This section outlines programming of the Timer Control Register for several typical examples.

(1) Periodic Interrupt Generator

7	6	5	4	3	2	1	0
TOUT/$\overline{\text{TIACK}}$ Control			Z.D. Ctrl.	*	Clock Control		Timer Enable
1	X	1	0	0	00 or 1X		changed

In this configuration the timer generates a periodic interrupt. The TOUT pin is connected to the system's interrupt request circuitry and the $\overline{\text{TIACK}}$ pin may be used as an interrupt acknowledge input to the timer. The TIN pin may be used as a clock input.

The processor loads the Counter Preload Registers and Timer Control Register, and then enables the timer. When the 24-bit counter passes from $000001 to $000000 the ZDS status bit is set and the TOUT (interrupt request) pin is asserted. At the next clock to the 24-bit counter it is again loaded with the contents of the CPR's, and thereafter decrements. In normal operation, the processor must direct clear the status bit to negate the interrupt request. (Figure 20)

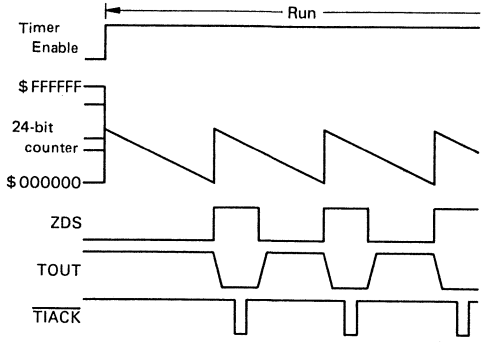


Figure 20 Periodic Interrupt Generator

(2) Square Wave Generator

Square Wave Generator

7	6	5	4	3	2	1	0
TOUT/$\overline{\text{TIACK}}$ Control			Z.D. Ctrl.	*	Clock Control		Timer Enable
0	1	X	0	0	00 or 1X		changed

In this configuration the timer produces a square wave at the TOUT pin. The TOUT pin is connected to the user's circuitry and the $\overline{\text{TIACK}}$ pin is not used. The TIN pin may be used as a clock input.

The processor loads the Counter Preload Registers and Timer Control Register, and then enables the timer. When the 24-bit counter passes from $000001 to $000000 the ZDS

status bit is set and the TOUT (square wave output) pin is toggled. At the next clock to the 24-bit counter it is again loaded with the contents of the CPRs, and thereafter decrements. In this application there is no need for the processor to direct clear the ZDS status bit; however, it is possible for the processor to sync itself with the square wave by clearing the ZDS status bit, then polling it. The processor may also read the TOUT level at the Port C address.

Note that the $\overline{PC_3}$/TOUT pin functions as PC3 following the negation of $\overline{RESET}$. If used in the square wave configuration a pullup resistor may be required to keep a known level prior to programming. Prior to enabling the timer, TOUT is high. (Figure 21)

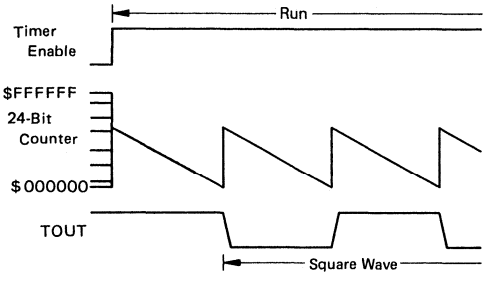


Figure 21 Square Wave Generator

(3) Interrupt After Timeout

Interrupt After Timeout

7	6	5	4	3	2	1	0
TOUT/$\overline{TIACK}$ Control			Z.D. Ctrl.	*	Clock Control		Timer Enable
1	X	1	1	0	00 or 1X		changed

In this configuration the timer generates an interrupt after a programmed time period has expired. The TOUT pin is connected to the system's interrupt request circuitry and the $\overline{TIACK}$ pin may be an interrupt acknowledge input to the timer. The TIN pin may be used as a clock input.

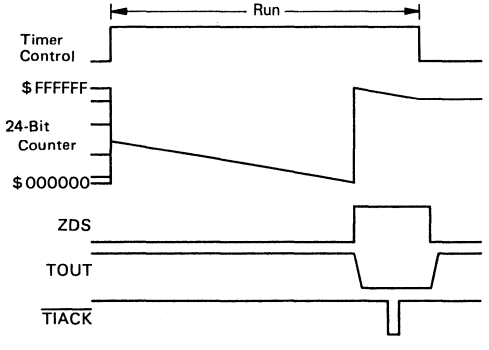


Figure 22 Interrupt After Timeout

This configuration is similar to the periodic interrupt generator except that the Zero Detect Control bit is set. This forces the counter roll over after Zero Detect is reached, rather than reloading from the CPRs. When the processor takes the interrupt it can halt the timer and read the counter. This allows the processor to measure the delay time from Zero Detect (interrupt request) to entering the service routine. Accurate knowledge of the interrupt latency may be useful in some applications. (Figure 22)

- **Elapsed Time Measurement**

Elapsed time measurement takes several forms; two are described below.

(1) System Clock

System Clock

7	6	5	4	3	2	1	0
TOUT/$\overline{TIACK}$ Control			Z.D. Ctrl.	*	Clock Control		Timer Enable
0	0	X	1	0	0	0	changed

This configuration allows time interval measurement by software. No timer pins are used.

The processor loads the Counter Preload Registers (generally with all 1s) and Timer Control Register, and then enables the timer. The counter decrements until the ending event takes place. When it is desired to read the time interval, the processor must halt the timer, then read the counter.

For applications in which the interval could have exceeded that programmable in this timer, interrupts can be counted to provide the equivalent of additional timer bits. At the end, the timer can be halted and read.

(2) External Clock

External Clock

7	6	5	4	3	2	1	0
TOUT/$\overline{TIACK}$ Control			Z.D. Ctrl.	*	Clock Control		Timer Enable
0	0	X	1	0	1	X	changed

This configuration allows measurement (counting) of the number of input pulses occurring in an interval in which the counter is enabled. The TIN input pin provides the input pulses. Generally the TOUT and $\overline{TIACK}$ pins are not used.

This configuration is identical to the Elapsed Time Measurement/System Clock configuration except that the TIN pin is used to provide the input frequency. It can be connected to a simple oscillator, and the same methods could be used. Alternately, it could be gated off and on externally and the number of cycles occurring while in the run state can be counted. However, minimum pulse width high and low specifications must be met. (Figure 23)

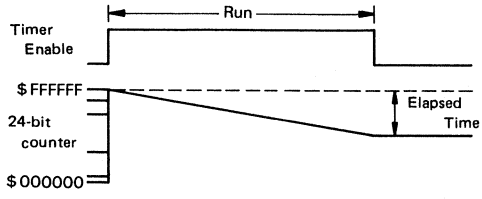


Figure 23 Elapsed Time Measurement

- **Device Watchdog**

Device Watching

7	6	5	4	3	2	1	0
TOUT/$\overline{\text{TIACK}}$ Control			Z.D. Ctrl.	*	Clock Control		Timer Enable
1	X	1	1	0	0	1	changed

This configuration provides the watchdog function needed in many systems. The TIN pin is the timer input whose period at the high (1) level is to be checked. Once allowed by the processor, the TIN input pin controls the run/halt mode. The TOUT pin is connected to external circuitry requiring notification when the TIN pin has been asserted longer than the programmed time. The $\overline{\text{TIACK}}$ pin (interrupt acknowledge) is only needed if the TOUT pin is connected to interrupt circuitry.

The processor loads the Counter Preload Register and Timer Control Register, and then enables the timer. When the TIN input is asserted (1, high) the timer transfers the contents of the Counter Preload Register to the counter and begins counting. If the TIN input is negated before Zero Detect is reached, the TOUT output and the ZDS status bit remain negated. If Zero Detect is reached while the TIN input is still asserted the ZDS status bit is set and the TOUT output is asserted. (The counter rolls over and keeps on counting.)

In either case, when the TIN input is negated the ZDS status bit is 0, the TOUT output is negated, the counting stops, and the prescaler is forced to all 1s. (Figure 24)

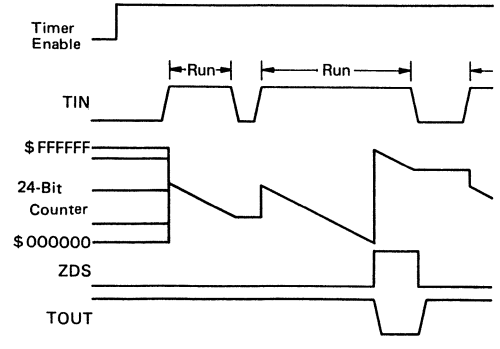


Figure 24 Device Watchdog

■ BUS INTERFACE CONNECTION

The PI/T has an asynchronous bus interface, primarily designed for use with the HD68000 microprocessor. With care, however, it can be connected to synchronous microprocessor buses. This section completely describes the PI/T's bus interface, and is intended for the asynchronous bus designer unless otherwise mentioned.

In an asynchronous system the PI/T CLK may operate at a significantly different frequency, either higher or lower, than the bus master and other system components, as long as all bus specifications are met. The HD68230 CLK pin has the same specifications as the HD68000 CLK, and must not be gated off at any time.

The following signals generate normal read and write cycles to the PI/T: $\overline{\text{CS}}$ (Chip Select), R/$\overline{\text{W}}$ (Read/Write). RS1-RS5 (five Register Select bits), D_0-D_7 (the 8-bit bidirectional data bus), and $\overline{\text{DTACK}}$ (Data Transfer Acknowledge). To generate interrupt acknowledge cycles $PC_5/\overline{\text{PIACK}}$ or $PC_7/\overline{\text{TIACK}}$ is used instead of CS, and the Register Select pins are ignored. No combination of the following pins may be asserted simultaneously: $\overline{\text{CS}}$, $\overline{\text{PIACK}}$, or $\overline{\text{TIACK}}$.

- **Read Cycles Via Chip Select**

This category includes all register reads, except port or timer interrupt acknowledge cycles. When $\overline{\text{CS}}$ is asserted, the Register Select and R/$\overline{\text{W}}$ inputs are latched internally. They must meet small setup and hold time requirements with respect to the asserted edge of $\overline{\text{CS}}$. (See the AC ELECTRICAL CHARACTERISTICS table.) The PI/T is not protected against aborted (shortened) bus cycles generated by an Address Error or Bus Error exception in which it is addressed.

Certain operations triggered by normal read (or write) bus cycles are not complete within the time allotted to the bus cycle. On example is transfers to/from the double-buffered latches that occur as a result of the bus cycle. If the bus master's CLK is significantly faster than the PI/T's the possibility exists that, following the bus cycle, $\overline{\text{CS}}$ can be negated then re-asserted before completion of these internal operations. In this situation the PI/T does not recognize the re-assertion of $\overline{\text{CS}}$ until these operations are complete. Only at that time does it begin the internal sequencing necessary to react to the asserted $\overline{\text{CS}}$. Since $\overline{\text{CS}}$ also controls the $\overline{\text{DTACK}}$ response, this "bus cycle recovery time" can be related to the CLK edge on which $\overline{\text{DTACK}}$ is asserted for that cycle. The PI/T will recognize the subsequent assertion of $\overline{\text{CS}}$ three (3) CLK periods after the CLK edge on which $\overline{\text{DTACK}}$ was previously asserted.

The Register Select and R/$\overline{\text{W}}$ inputs pass through an internal latch that is transparent when the PI/T can recognize a new $\overline{\text{CS}}$ pulse (see above paragraph). Since the internal data bus of the PI/T is continuously enabled for read transfers, the read access time (to the data bus buffers) begins when the Register Selects are stabilized internally. Also, when the PI/T is ready to begin a new bus cycle, the assertion of $\overline{\text{CS}}$ enables the data bus buffers within a short propagation delay. This does not contribute to the overall read access time unless $\overline{\text{CS}}$ is asserted significantly after the Register Select and R/$\overline{\text{W}}$ inputs are stabilized (as may occur with synchronous bus microprocessors).

In addition to Chip Select's previously mentioned duties, it controls the assertion of $\overline{\text{DTACK}}$ and latching of read data at the data bus interface. Except for controlling input latches and enabling the data bus buffers, all of these functions occur only after $\overline{\text{CS}}$ has been recognized internally and synchronized with the internal clock. Chip Select is recognized on the falling edge of the CLK if the setup time is met, $\overline{\text{DTACK}}$ is asserted

(low) on the next falling edge of the CLK. Read data is latched at the PI/T's data bus interface at the same time $\overline{DTACK}$ is asserted. It is stable as long as Chip Select remains asserted independent of other external conditions.

From the above discussion it is clear that if the $\overline{CS}$ setup time prior to the falling edge of the CLK is met, the PI/T can consistently respond to a new read or write bus cycle every four (4) CLK cycles. This fact is especially useful in designing the PI/T's clock in synchronous bus systems not using $\overline{DTACK}$. (An extra CLK period is required in interrupt acknowledge cycles, see Read Cycles via Interrupt Acknowledge.)

In asynchronous bus systems in which the PI/T's CLK differs from that of the bus master, generally there is no way to guarantee that the $\overline{CS}$ setup time with respect to the PI/T CLK is met. Thus, the only way to determine that the PI/T recognized the assertion of $\overline{CS}$ is to wait for the assertion of $\overline{DTACK}$. In this situation, all latched bus inputs to the PI/T must be held stable until $\overline{DTACK}$ is asserted. These include Register Select, R/$\overline{W}$, and write data inputs (see below).

System specifications impose a maximum delay from the trailing (negated) edge of Chip Select to the negated edge of $\overline{DTACK}$. As system speeds increase this becomes more difficult to meet with a simple pullup resistor tied to the $\overline{DTACK}$ line. Therefore, the PI/T provides an internal active pullup device to reduce the rise time, and a level-sensitive circuit that later turns this device off. $\overline{DTACK}$ is negated asynchronously as fast as possible following the rising edge of Chip Select, then three-stated to avoid interference with the next bus cycle.

The system designer must take care that $\overline{DTACK}$ is negated and three-stated quickly enough after each bus cycle to avoid interference with the next one. With the HD68000 this necessitates a relatively fast external path from the data strobe to $\overline{CS}$ going negated.

- **Write Cycles**

In many ways write cycles are similar to normal read cycles (see above). On write cycles, data at the D_0-D_7 pins must meet the same setup specifications as the Register Select and R/$\overline{W}$ lines. Like these signals, write data is latched on the asserted edge of $\overline{CS}$, and must meet small setup and hold time requirements with respect to that edge. The same bus cycle recovery conditions exist as for normal read cycles. No other differences exist.

- **Read Cycles Via Interrupt Acknowledge**

Special internal operations take place on PI/T interrupt acknowledge cycles. The Port Interrupt Vector Register or the Timer Interrupt Vector Register are implicitly addressed by the assertion of $PC_6/\overline{PIACK}$ or $PC_7/\overline{TIACK}$, respectively. The signals are first synchronized with the falling edge of the CLK. One clock period after they are recognized the data bus buffers are enabled and the vector is driven onto the bus. $\overline{DTACK}$ is asserted after another clock period to allow the vector some setup time prior to $\overline{DTACK}$.$\overline{DTACK}$ is negated, then three-stated as with normal read or write cycle, when $\overline{PIACK}$ or $\overline{TIACK}$ is negated.

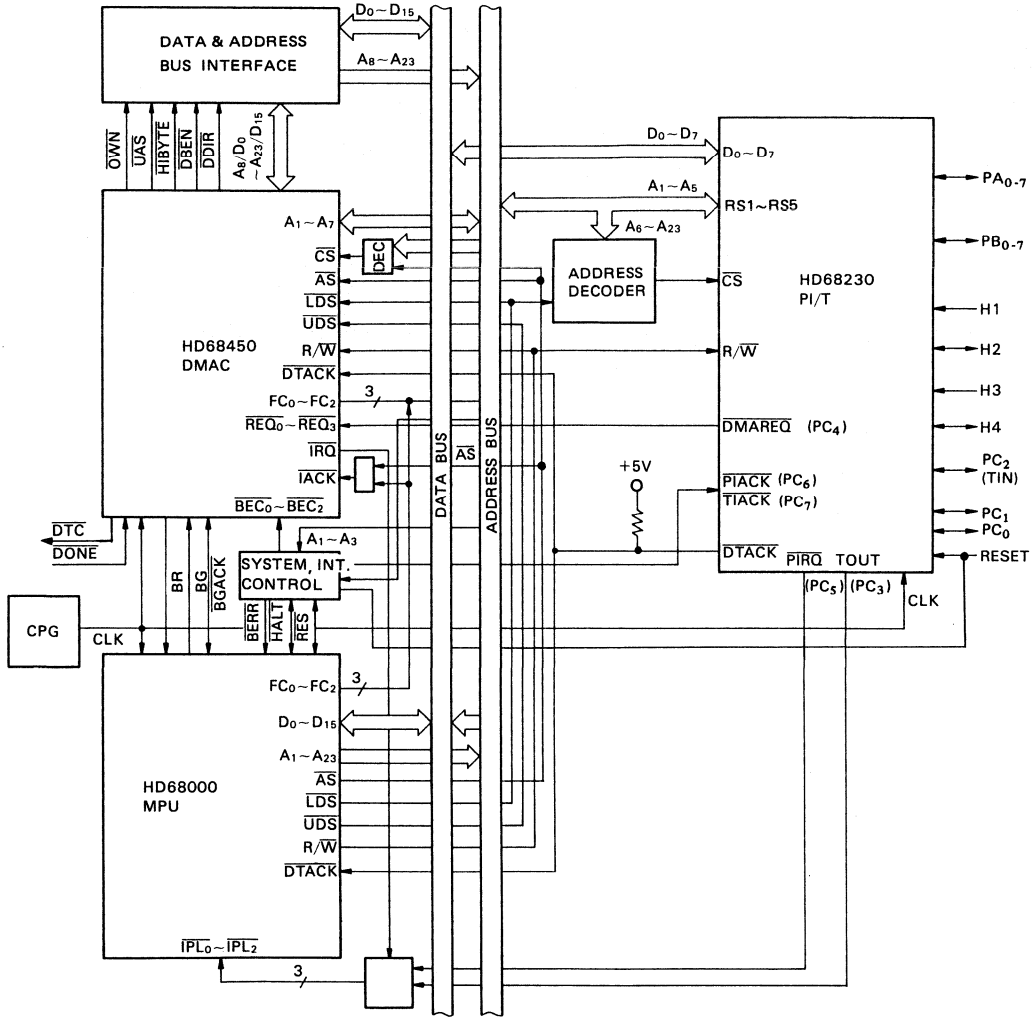


Figure 25 HMCS68000 Interface-Example

■ RESTRICTION ON HD68230 USAGE

The parallel interface/timer (PI/T) exhibits an anomaly during interrupt acknowledge ($\overline{IACK}$) cycles for certain configurations of the part. If the PI/T is configured to have only one interrupt source (either the port or the timer), and pins 36 ($PC_6/\overline{PIACK}$) and 37 ($PC_6/\overline{TIACK}$) are both low during $\overline{IACK}$ cycles, an incorrect vector number will be placed on the data bus and the interrupt vector register corresponding to the IACK cycles will be changed.

Specifically, if:

the PI/T is programmed to generate a vectored timer interrupt (i.e., pin 33 is programmed as TOUT and pin 37 as $\overline{TIACK}$), and pin 36 is programmed to be a general-purpose input or output that is low during $\overline{TIACK}$ cycles.

or if

the PI/T is programmed to generate a vectored port interrupt (i.e., pin 35 is programmed as $\overline{PIRQ}$ and pin 36 as PIACK), and pin 37 is programmed to be a general-purpose input or output that is low during $\overline{PIACK}$ cycles,

then,

during $\overline{IACK}$ cycles, the PI/T will misinterpret the low signals present on pins 36 and 37 as simultaneously asserted $\overline{PIACK}$ and $\overline{TIACK}$ signals, which is an illegal condition.

There is both a hardware solution and a software solution for this anomaly:

Hardware: Insure that whichever of the two pins not programmed as an $\overline{IACK}$ input will be high during $\overline{IACK}$ cycles. For example, if pin 37 is used as $\overline{TIACK}$ and pin 36 is programmed as a PC_6 input, force pin 36 high whenever pin 37 is low. This can be accomplished with either a pullup resistor or external logic.

Software: If only timer interrupts are to be used, initialize the PIVR with $FC and select a vector number for the TIVR that has the two least significant bits clear (i.e., binary xxxxxx00). If only port interrupts are to be used, initialize the TIVR with $FF and select any vector number for the PIVR.

Note that this anomaly will not arise if the PI/T interrupts are autovectored (since no $\overline{IACK}$ signal will be required) or if the PI/T is programmed to accept both port and timer interrupt acknowledges (since external $\overline{IACK}$ logic will insure that pins 36 and 37 are never low simultaneously).

HD63310
S-DPRAM (Smart Dual Port RAM)

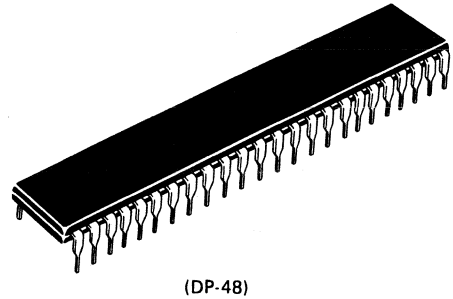
—ADVANCE INFORMATION—

The HD63310 (S-DPRAM) is a high intelligent DPRAM, which provide a communication path between multiprocessor systems.

The HD63310 has 1024 x 8 bit RAM, 62 x 8 bit registers and individual dual I/O ports. The dual ports perform read/write operations independently and simultaneously.

User can select one of the two mode (DPRAM or FIFO mode) by the program. This architecture makes it possible to communicate efficiently according to applications.

HD63310P-15, HD63310P-20



(DP-48)

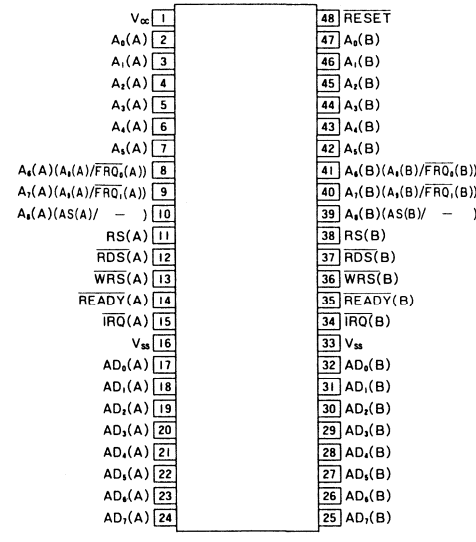
■ FEATURES

- 2 independent asynchronous bus operation Address/Data bus configurable as multiplexed or non-multiplexed bus.
- Dual port large scale data buffer space
  - Dual port RAM mode: 1024 byte
  - FIFO mode: 2 FIFOs for 1024 byte
- 62 internal registers
  - Semaphore registers which support multi-processing (8 bit)
  - 32 registers which user can use freely
- Access Time
  150 ns/200 ns
- Low power consumption
  2 μm full CMOS circuit

■ TYPE OF PRODUCTS

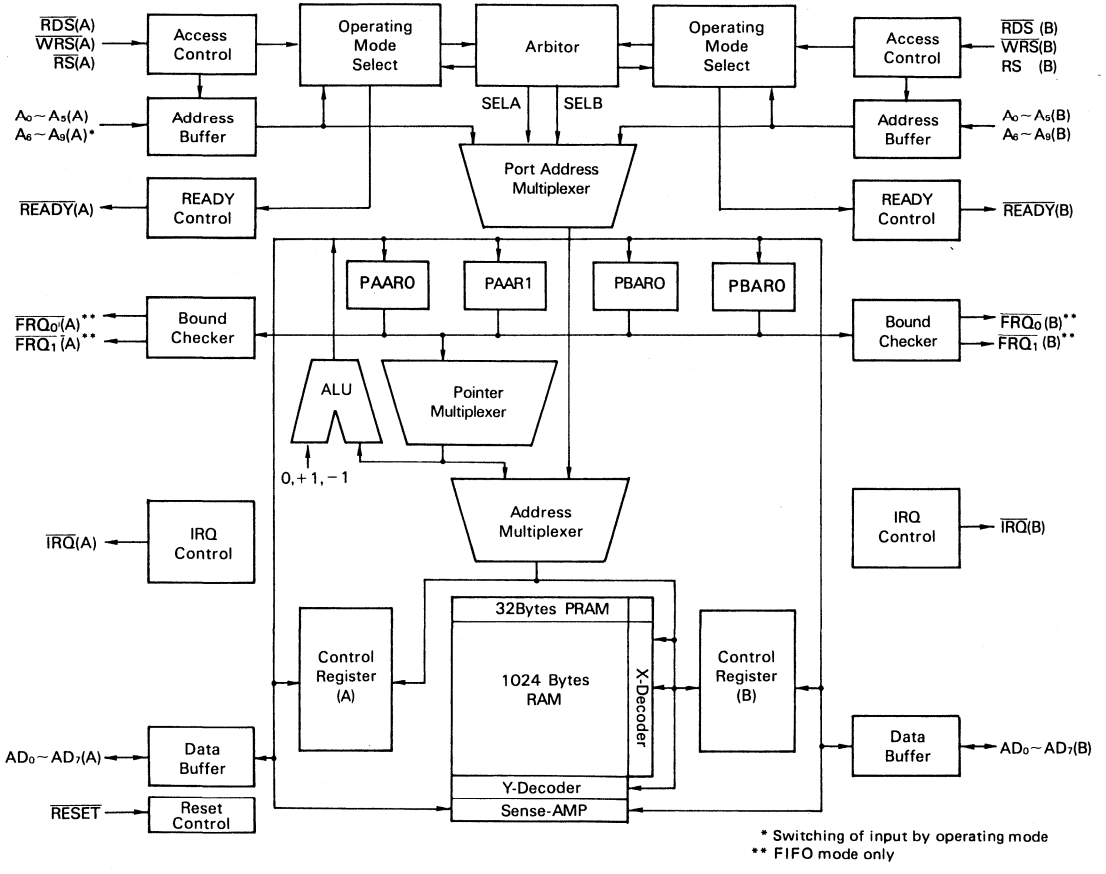
Type No.	Access Time
HD63310P-15	150 ns
HD63310P-20	200 ns

■ PIN ARRANGEMENT

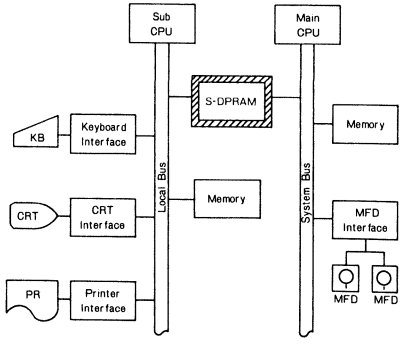


(Top View)

■ BLOCK DIAGRAM



■ SYSTEM BLOCK DIAGRAM



HD63450,HD63450Y,HD63450P, HD63450PS,HD63450CP

DMAC (Direct Memory Access Controller)

—PRELIMINARY—

The HD63450 is the Direct Memory Access Controller (DMAC) realized by the CMOS technology. It is upward compatible with the NMOS DMAC HD68450.

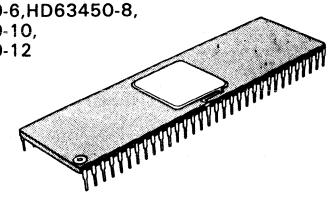
The DMAC performs one or several blocks of data transfer (Operand ; byte, word, or long word) between memory and peripheral device at high speed. The block transfer restart operation is provided (Multi-Block Transfer with $\overline{\text{DONE}}$ Mode). The number of operands in a block is determined by a transfer count.

The power dissipation is lowered by adopting the CMOS process.

■ FEATURES

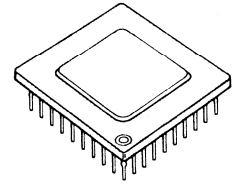
- HMCS68000 Bus Compatible
- 4 independent DMA Channels with Programmable Priority
- Memory-to-Memory, Memory-to-Device, Device-to-Memory Transfers
- Programmable 8-Bit or 16-Bit I/O Device Types
- Auto-Request and External-Request Transfer Modes
- Interface Lines for Requesting, Acknowledging, and Incidental Control of the Peripheral Devices
- Block Transfer Operation
  - In Single-Block : ● Unchaining Transfer
  - In Multi-Block : ● Continue Mode Transfer
    - ● Array-Chaining and Linked-Array-Chaining Transfers
    - ● Multi-Block Transfer with $\overline{\text{DONE}}$
- 68000 Bus Exception Processing Support
- 2 Vectored Interrupts for each Channel
- Variable System Bus Bandwidth Rate Utilization
- Fast Transfer Rates: Up to 6.25 Mbytes/sec. at 12.5MHz
- CMOS +5 Volts Operation

HD63450-6,HD63450-8,
HD63450-10,
HD63450-12



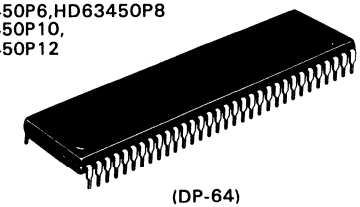
(DC-64)

HD63450Y6,HD63450Y8,
HD63450Y10,
HD63450Y12



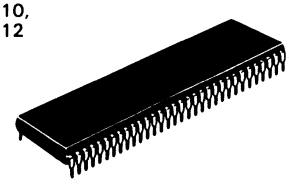
(PGA-68)

HD63450P6,HD63450P8
HD63450P10,
HD63450P12



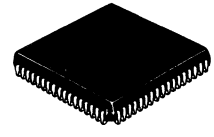
(DP-64)

HD63450PS6,HD63450PS8
HD63450PS10,
HD63450PS12



(DP-64S)

HD63450CP6,HD63450CP8
HD63450CP10,
HD63450CP12



(CP-68)

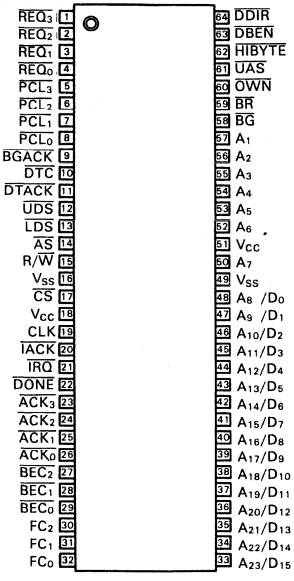
■ TYPE OF PRODUCTS

Type No.	Bus Timing	Packaging
HD63450-6	6MHz	DC-64
HD63450-8	8MHz	
HD63450-10	10MHz	
HD63450-12	12.5MHz	
HD63450Y6	6MHz	PGA-68
HD63450Y8	8MHz	
HD63450Y10	10MHz	
HD63450Y12	12.5MHz	
HD63450P6*	6MHz	DP-64
HD63450P8*	8MHz	
HD63450P10*	10MHz	
HD63450P12*	12.5MHz	
HD63450PS6*	6MHz	DP-64S
HD63450PS8*	8MHz	
HD63450PS10*	10MHz	
HD63450PS12*	12.5MHz	
HD63450CP6*	6MHz	CP-68
HD63450CP8*	8MHz	
HD63450CP10*	10MHz	
HD63450CP12*	12.5MHz	

*Under development

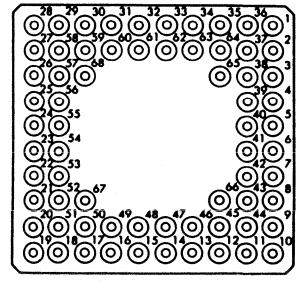
■ PIN ARRANGEMENT

● HD63450, HD63450P, HD63450PS



(Top View)

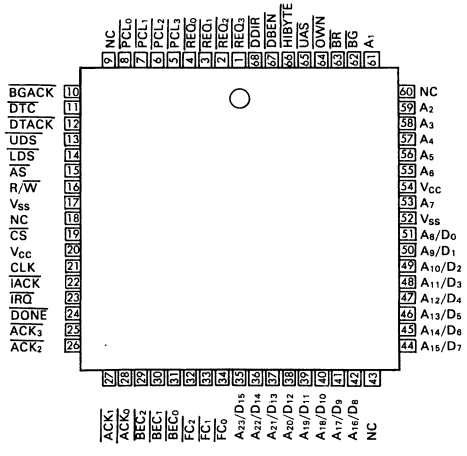
● HD63450Y



(Bottom View)

Pin No.	Function	Pin No.	Function	Pin No.	Function	Pin No.	Function
1	N/C	18	$\overline{PCL_1}$	35	A_{19}/D_{11}	52	$\overline{BGACK}$
2	A_{13}/D_5	19	$\overline{DTACK}$	36	A_{17}/D_9	53	$\overline{LDS}$
3	A_{11}/D_3	20	$\overline{UDS}$	37	A_{15}/D_7	54	V_{SS}
4	A_{10}/D_2	21	$\overline{AS}$	38	A_{12}/D_4	55	V_{CC}
5	A_8/D_0	22	R/$\overline{W}$	39	A_9/D_1	56	$\overline{DONE}$
6	A_7	23	N/C	40	V_{SS}	57	$\overline{IRQ}$
7	A_6	24	$\overline{CS}$	41	V_{CC}	58	$\overline{ACK_2}$
8	A_5	25	CLK	42	A_4	59	$\overline{BEC_2}$
9	A_3	26	$\overline{IACK}$	43	A_2	60	$\overline{BEC_0}$
10	N/C	27	$\overline{ACK_3}$	44	$\overline{BG}$	61	FC_0
11	$\overline{BR}$	28	$\overline{ACK_0}$	45	$\overline{OWN}$	62	A_{21}/D_{13}
12	$\overline{UAS}$	29	$\overline{BEC_1}$	46	$\overline{HIBYTE}$	63	A_{18}/D_{10}
13	$\overline{DBEN}$	30	FC_2	47	$\overline{DDIR}$	64	A_{16}/D_8
14	$\overline{REQ_3}$	31	FC_1	48	$\overline{REQ_1}$	65	A_{14}/D_6
15	$\overline{REQ_2}$	32	A_{23}/D_{15}	49	$\overline{PCL_2}$	66	A_1
16	$\overline{REQ_0}$	33	A_{22}/D_{14}	50	$\overline{PCL_0}$	67	$\overline{DTC}$
17	$\overline{PCL_3}$	34	A_{20}/D_{12}	51	N/C	68	$\overline{ACK_1}$

● HD63450CP



(Top View)

■ ABSOLUTE MAXIMUM RATINGS

Item	Symbol	Value	Unit
Supply Voltage	V_{CC}*	−0.3～+7.0	V
Input Voltage	V_{in}*	−0.3～V_{CC}+0.3	V
Operating Temperature Range	T_{opr}	0～+70	°C
Storage Temperature	T_{stg}	−55～+150	°C

*With respect to V_{SS} (SYSTEM GND)

(NOTE) Permanent LSI damage may occur if maximum ratings are exceeded. Normal operation should be under recommended operating conditions. If these conditions are exceeded, it could affect reliability of LSI.

■ RECOMMENDED OPERATING CONDITIONS

Item	Symbol	min	typ	max	Unit
Supply Voltage	V_{CC}*	4.75	5.0	5.25	V
Input Voltage	V_{IH}*	2.0		V_{CC}	V
	V_{IL}*	−0.3		0.8	V
Operating Temperature	T_{opr}	0	25	70	°C

*With respect to V_{SS} (SYSTEM GND)

■ ELECTRICAL CHARACTERISTICS

● DC CHARACTERISTICS (V_{CC}=5V±5%, V_{SS}=0V, T_a=0～+70°C, unless otherwise noted.)

Item		Symbol	Test Condition	min	typ	max	Unit
Input "High" Voltage		V_{IH}		2.0		V_{CC}	V
Input "Low" Voltage		V_{IL}		V_{SS}−0.3		0.8	V
Input Leakage Current	$\overline{CS}$, $\overline{IACK}$, $\overline{BG}$, CLK, $\overline{BEC_0}$～$\overline{BEC_2}$ $\overline{REQ_0}$～$\overline{REQ_3}$	I_{in}				10	μA
Three-State (Off State) Input Current	A_1～A_7, D_0～D_{15}/A_8～A_{23}, $\overline{AS}$, $\overline{UDS}$, $\overline{LDS}$, R/$\overline{W}$, $\overline{UAS}$, $\overline{DTACK}$, $\overline{BGACK}$, $\overline{OWN}$, $\overline{DTC}$, $\overline{HIBYTE}$, $\overline{DDIR}$, $\overline{DBEN}$, FC_0～FC_2, $\overline{PCL_0}$～$\overline{PCL_3}$	I_{TSI}				10	μA
Open Drain (Off State) Input Current	$\overline{IRQ}$, $\overline{DONE}$	I_{ODI}				20	μA
Output "High" Voltage	A_1～A_7, D_0～D_{15}/A_8～A_{23}, $\overline{AS}$, $\overline{UDS}$, $\overline{LDS}$, R/$\overline{W}$, $\overline{UAS}$, $\overline{DTACK}$, $\overline{BGACK}$, $\overline{BR}$, $\overline{OWN}$, $\overline{DTC}$, $\overline{HIBYTE}$, $\overline{DDIR}$, $\overline{DBEN}$, $\overline{ACK_0}$～$\overline{ACK_3}$, $\overline{PCL_0}$～$\overline{PCL_3}$, FC_0～FC_2	V_{OH}	I_{OH}=−400μA	V_{CC}−2.0			V
Output "Low" Voltage	A_1～A_7, FC_0～FC_2	V_{OL}	I_{OL}=3.2mA			0.5	V
	D_0～D_{15}/A_8～A_{23}, $\overline{AS}$, $\overline{UDS}$, $\overline{LDS}$, R/$\overline{W}$, $\overline{DTACK}$, $\overline{BR}$, $\overline{OWN}$, $\overline{DTC}$, $\overline{HIBYTE}$, $\overline{DDIR}$, $\overline{DBEN}$, $\overline{ACK_0}$～$\overline{ACK_3}$, $\overline{UAS}$, $\overline{PCL_0}$～$\overline{PCL_3}$, $\overline{BGACK}$	V_{OL}	I_{OL}=5.3mA			0.5	
	$\overline{IRQ}$, $\overline{DONE}$	V_{OL}	I_{OL}=8.9mA			0.5	
Power Dissipation		P_D	f=8MHz, V_{CC}=5.0V T_a=25°C		0.5	1.0	W
Capacitance		C_{in}	V_{in}=0V T_a=25°C, f=1MHz			15	pF

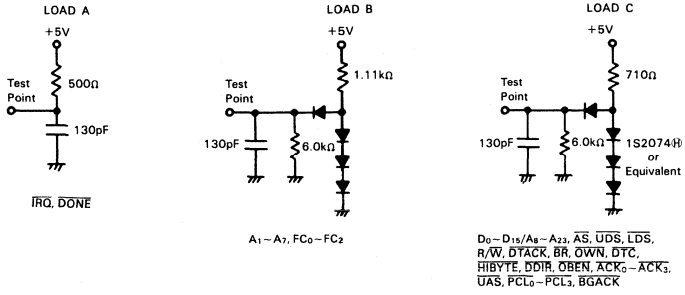


Figure 1 Test Loads

● AC ELECTRICAL SPECIFICATIONS (V_{CC}=5V±5%, V_{SS}=0V, T_a=0~+70°C, unless otherwise noted.)

No.	Item	Symbol	Test Condition	6MHz Version		8MHz Version		10MHz Version		12.5MHz Version		Unit
				min	max	min	max	min	max	min	max	
	Frequency of Operation	f		4.0	6.0	4.0	8.0	4.0	10.0	4.0	12.5	MHz
1	Clock Period	t_{cyc}		167	500	125	500	100	500	80	250	ns
2	Clock Width Low	t_{CL}		75	250	55	250	45	250	35	125	ns
3	Clock Width High	t_{CH}		75	250	55	250	45	250	35	125	ns
4	Clock Fall Time	t_{Cf}			10		10		10		5	ns
5	Clock Rise Time	t_{Cr}			10		10		10		5	ns
6	Asynchronous Input Setup Time	t_{ASI}		25		20		15		15		ns
7	Data in to $\overline{DBEN}$ Low	t_{DIDBL}		0		0		0		0		ns
8	$\overline{DTACK}$ Low to Data Invalid	t_{DTLDI}		0		0		0		0	-	ns
9	Address in to $\overline{AS}$ in Low	t_{AIASL}		0		0		0		0		ns
10	$\overline{AS}$, $\overline{DS}$ in High to Address in Invalid	t_{SIHAIV}		0		0		0		0		ns
10A	$\overline{DS}$ in High to $\overline{CS}$ High	t_{DSHCSH}			1.0		1.0		1.0	clk.per.	1.0	clk.per.
11	Clock High to $\overline{DDIR}$ Low	t_{CHDRL}			80		70		60		55	ns
12	Clock High to $\overline{DDIR}$ High	t_{CHDRH}			80		70		60		55	ns
13	$\overline{DS}$ in High to $\overline{DDIR}$ High Impedance	t_{DSHDRZ}	Figure1		140		120		110		110	ns
14	Clock Low to $\overline{DBEN}$ Low	t_{CLDBL}	~		80		70		60		55	ns
15	Clock Low to $\overline{DBEN}$ High	t_{CLDBH}			80		70		60		55	ns
16	$\overline{DS}$ in High to $\overline{DBEN}$ High Impedance	t_{DSHDBZ}			140		120		110		110	ns
17	Clock High to Data Out Valid (MPU read)	t_{CHDVM}	Figure8		230		180		160		140	ns
18	$\overline{DS}$ in High to Data Out Invalid	t_{DSHDZn}		0		0		0		0		ns
19	$\overline{DS}$ in High to Data High Impedance	t_{DSHDZ}			140		120		110		110	ns
20	Clock Low to $\overline{DTACK}$ Low	t_{CLDTL}			80		70		60		55	ns
21	$\overline{DS}$ in High to $\overline{DTACK}$ High	t_{DSHDTH}			130		110		110		110	ns
22	$\overline{DTACK}$ Width High	t_{DTH}		10		10		10		10		ns
23	$\overline{DS}$ in High to $\overline{DTACK}$ High Impedance	t_{DSHDTZ}			200		180		160		160	ns
24	$\overline{DTACK}$ Low to $\overline{DS}$ in High	t_{DTLDSH}		0		0		0		0		ns
25	$\overline{REQ}$ Width Low	t_{REQL}		2.0		2.0		2.0		2.0		clk.per.
26	$\overline{REQ}$ Low to $\overline{BR}$ Low	t_{RELBRL}		334		250		200		160		ns
27	Clock High to $\overline{BR}$ Low	t_{CHBRL}			80		70		60		55	ns
28	Clock High to $\overline{BR}$ High	t_{CHBRH}			80		70		60		55	ns
29	$\overline{BG}$ Low to $\overline{BGACK}$ Low	t_{BGLBL}		4.5		4.5		4.5		4.5		clk.per.
30	$\overline{BR}$ Low to MPU Cycle End ($\overline{AS}$ in High)	t_{BRLASH}		0		0		0		0		ns

(to be continued)

No.	Item	Symbol	Test Condition	6MHz Version		8MHz Version		10MHz Version		12.5MHz Version		Unit
				min	max	min	max	min	max	min	max	
31	MPU Cycle End ($\overline{AS}$ in High) to $\overline{BGACK}$ Low	t_{ASHBL}		4.5	5.5	4.5	5.5	4.5	5.5	4.5	5.5	clk.per.
32	$\overline{REQ}$ Low to $\overline{BGACK}$ Low	t_{REQLBL}		6.5		6.5		6.5		6.5		clk.per.
33	Clock High to $\overline{BGACK}$ Low	t_{CHBL}			80		70		60		55	ns
34	Clock High to $\overline{BGACK}$ High	t_{CHBH}			80		70		60		55	ns
35	Clock Low to $\overline{BGACK}$ High Impedance	t_{CLBZ}			100		80		70		65	ns
36	Clock High to FC Valid	t_{CHFCV}			120		100		90		80	ns
37	Clock High to Address Valid	t_{CHAV}			140		120		110		100	ns
38	Clock High to Address/FC/Data High Impedance	t_{CHAZx}			120		100		100		100	ns
39	Clock High to Address/FC/Data Invalid	t_{CHAZn}		0		0		0		0		ns
40	Clock Low to Address High Impedance	t_{CLAZ}			120		100		90		80	ns
41	Clock High to $\overline{UAS}$ Low	t_{CHUL}			80		70		60		55	ns
42	Clock High to $\overline{UAS}$ High	t_{CHUH}			80		70		60		55	ns
43	Clock Low to $\overline{UAS}$ High Impedance	t_{CLUZ}			100		80		70		65	ns
44	$\overline{UAS}$ High to Address Invalid	t_{UHAI}		40		30		20		15		ns
45	Clock High to $\overline{AS}$, $\overline{DS}$ Low	t_{CHSL}			70		60		55		55	ns
46	Clock Low to $\overline{DS}$ Low (write)	t_{CLDSL}			70		60		55		55	ns
47	Clock Low to $\overline{AS}$, $\overline{DS}$ High	t_{CLSH}			80		70		60		60	ns
48	Clock Low to $\overline{AS}$, $\overline{DS}$ High Impedance	t_{CLSZ}			100		80		70		65	ns
49	$\overline{AS}$ Width Low	t_{ASL}		350		255		195		160		ns
50	$\overline{DS}$ Width Low	t_{DSL}		265		190		145		120		ns
51	$\overline{AS}$, $\overline{DS}$ Width High	t_{SH}	Figure1	180		150		105		65		ns
52	Address/FC Valid to $\overline{AS}$, $\overline{DS}$ Low	t_{AVSL}		40		30		20		0		ns
53	$\overline{AS}$, $\overline{DS}$ High to Address/FC/Data Invalid	t_{SHAZ}	~	40		30		20		15		ns
54	Clock High to R/$\overline{W}$ Low	t_{CHRL}			80		70		60		55	ns
55	Clock High to R/$\overline{W}$ High	t_{CHRH}	Figure8		80		70		60		55	ns
56	Clock Low to R/$\overline{W}$ High Impedance	t_{CLRZ}			100		80		70		65	ns
57	Address/FC Valid to R/$\overline{W}$ Low	t_{AVRL}		40		20		10		0		ns
58	R/$\overline{W}$ Low to $\overline{DS}$ Low (write)	t_{RLSL}		170		120		90		70		ns
59	$\overline{DS}$ High to R/$\overline{W}$ High	t_{SHRH}		50		40		20		15		ns
60	Clock Low to $\overline{OWN}$ Low	t_{CLOL}			80		70		60		55	ns
61	Clock Low to $\overline{OWN}$ High	t_{CLOH}			80		70		60		55	ns
62	Clock High to $\overline{OWN}$ High Inpedance	t_{CHOZ}			100		80		70		65	ns
63	$\overline{OWN}$ Low to $\overline{BGACK}$ Low	t_{OLBL}		40		30		20		15		ns
64	$\overline{BGACK}$ High to $\overline{OWN}$ High	t_{BHOH}		40		30		20		15		ns
65	$\overline{OWN}$ Low to $\overline{UAS}$ Low	t_{OLUL}		40		30		20		15		ns
66	Clock High to $\overline{ACK}$ Low	t_{CHACL}			80		70		60		55	ns
67	Clock Low to $\overline{ACK}$ Low	t_{CLACL}			80		70		60		55	ns
68	Clock High to $\overline{ACK}$ High	t_{CHACH}			80		70		60		55	ns
69	$\overline{ACK}$ Low to $\overline{DS}$ Low	t_{ACLDSL}		140		100		80		60		ns
70	$\overline{DS}$ High to $\overline{ACK}$ High	t_{DSHACH}		40		30		20		15		ns
71	Clock High to $\overline{HIBYTE}$ Low	t_{CHHIL}			80		70		60		55	ns
72	Clock Low to $\overline{HIBYTE}$ Low	t_{CLHIL}			80		70		60		55	ns
73	Clock High to $\overline{HIBYTE}$ High	t_{CHHIH}			80		70		60		55	ns
74	Clock Low to $\overline{HIBYTE}$ High Impedance	t_{CLHIZ}			100		80		70		65	ns
75	Clock High to $\overline{DTC}$ Low	t_{CHDTL}			80		70		60		55	ns
76	Clock High to $\overline{DTC}$ High	t_{CHDTH}			80		70		60		55	ns

(to be continued)

No.	Item	Symbol	Test Condition	6MHz Version		8MHz Version		10MHz Version		12.5MHz Version		Unit
				min	max	min	max	min	max	min	max	
77	Clock Low to $\overline{DTC}$ High Impedance	t_{CLDTZ}			100		80		70		65	ns
78	$\overline{DTC}$ Width Low	t_{DTCL}		147		105		80		60		ns
79	$\overline{DTC}$ Low to $\overline{DS}$ High	t_{DTLDH}		50		30		20		15		ns
80	Clock High to $\overline{DONE}$ Low	t_{CHDOL}			80		70		60		55	ns
81	Clock Low to $\overline{DONE}$ Low	t_{CLDOL}			80		70		60		55	ns
82	Clock High to $\overline{DONE}$ High	t_{CHDOH}			140		130		120		120	ns
83	Clock Low to $\overline{DDIR}$ High Impedance	t_{CLDRZ}			100		80		70		65	ns
84	Clock Low to $\overline{DBEN}$ High Impedance	t_{CLDBZ}			100		80		70		65	ns
85	$\overline{DDIR}$ Low to $\overline{DBEN}$ Low	t_{DRLDBL}		40		30		20		15		ns
86	$\overline{DBEN}$ High to $\overline{DDIR}$ High	t_{DBHDRH}		40		30		20		15		ns
87	$\overline{DBEN}$ Low to Address/Data High Impedance	t_{DBLAZ}			17		17		17		17	ns
88	Clock Low to $\overline{PCL}$ Low (1/8 clock)	t_{CLPL}			80		70		60		55	ns
89	Clock Low to $\overline{PCL}$ High (1/8 clock)	t_{CLPH}	Figure1		80		70		60		55	ns
90	$\overline{PCL}$ Width Low (1/8 clock)	t_{PCLL}		4.0		4.0		4.0		4.0		clk.per.
91	$\overline{DTACK}$ Low to Data In (setup time)	t_{DALDI}	~		200		150		115		85	ns
92	$\overline{DS}$ High to Data Invalid (hold time)	t_{SHDI}		0		0		0		0		ns
93	$\overline{DS}$ High to $\overline{DTACK}$ High	t_{SHDAH}	Figure8	0	160	0	120	0	90	0	70	ns
94	Data Out Valid to $\overline{DS}$ Low	t_{DOSL}		0		0		0		0		ns
95	Data In to Clock Low (setup time)	t_{DICL}		25		15		15		15		ns
96	$\overline{BEC}$ Low to $\overline{DTACK}$ Low	t_{BECDAL}		50		50		50		50		ns
97	$\overline{BEC}$ Width Low	t_{BECL}		2.0		2.0		2.0		2.0		clk.per.
98	Clock High to $\overline{IRQ}$ Low	t_{CHIRL}			80		70		60		55	ns
99	Clock High to $\overline{IRQ}$ High	t_{CHIRH}			140		130		120		120	ns
100	$\overline{READY}$ In to $\overline{DTC}$ Low (Read)	t_{RALDTL}		180		145		120		100		ns
101	$\overline{READY}$ In to $\overline{DS}$ Low (Write)	t_{RALDSL}		240		205		170		140		ns
102	$\overline{DS}$ High to $\overline{READY}$ High	t_{DSHRAH}		0	160	0	120	0	90	0	70	ns
103	$\overline{DONE}$ In Low to $\overline{DTACK}$ Low	t_{DOLDAL}		50		50		50		50		ns
104	$\overline{DS}$ High to $\overline{DONE}$ In High	t_{DSHDOH}		0	160	0	120	0	90	0	70	ns
105	Asynchronous Input Hold Time	t_{ASIH}		15		15		15		15		ns

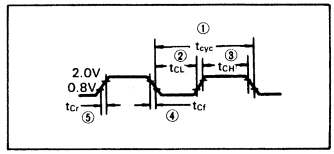


Figure 2 Input Clock Waveform

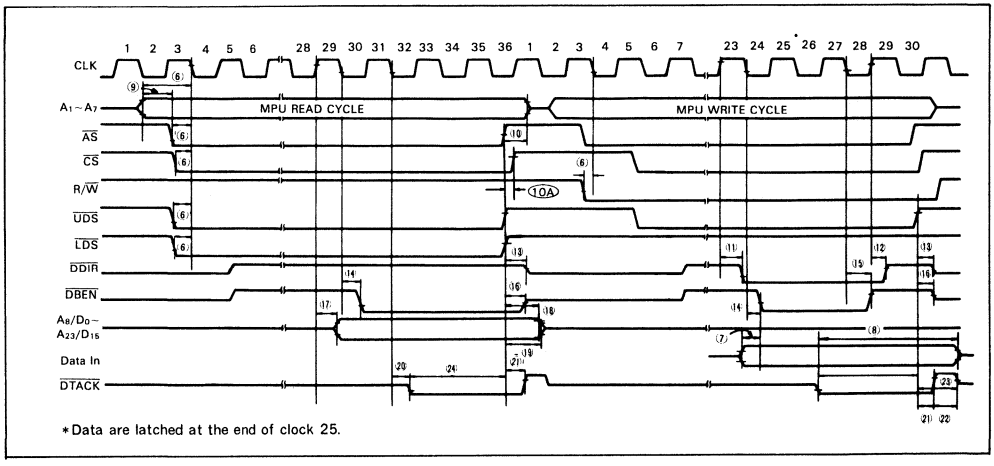


Figure 3 AC Electrical Waveforms-MPU Read/Write

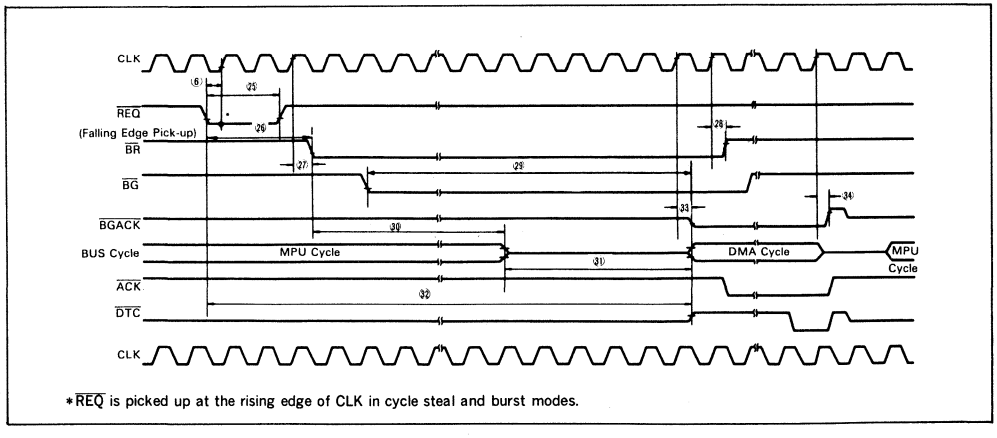
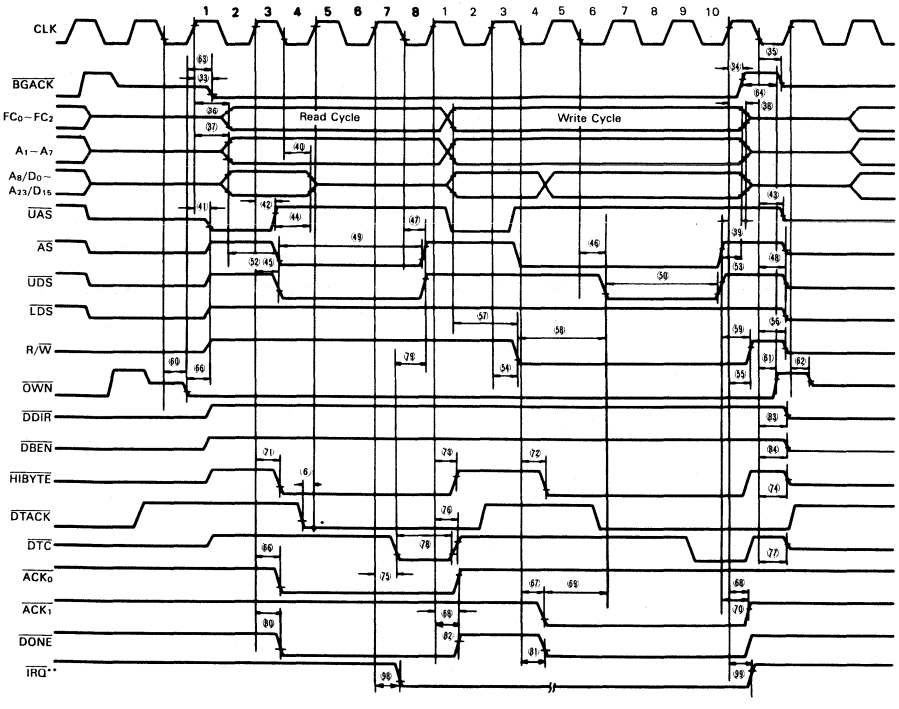
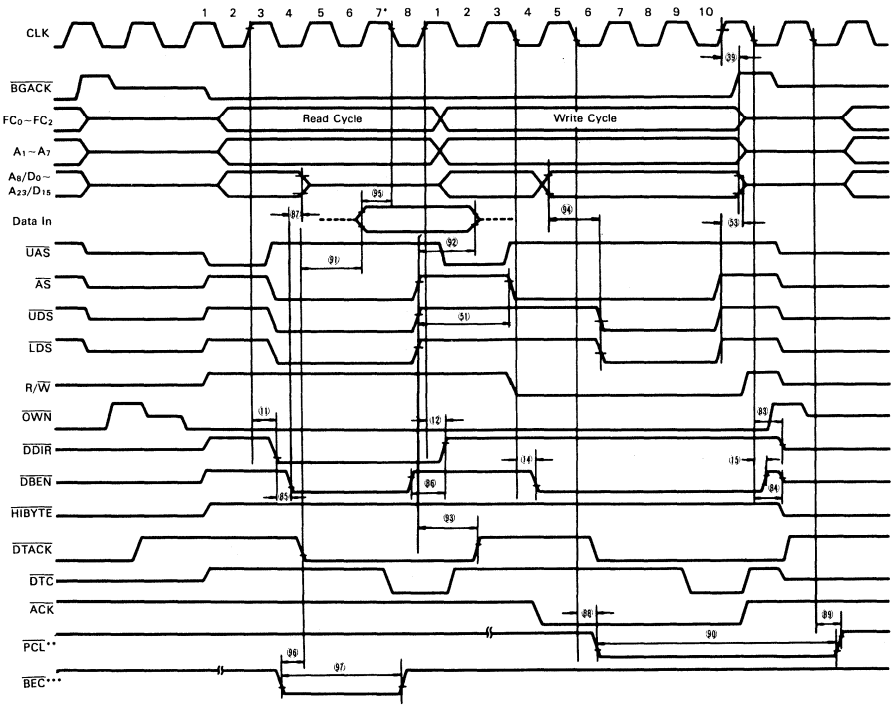


Figure 4 AC Electrical Waveforms-Bus Arbitration



*$\overline{\text{DTACK}}$ is picked up at the rising edge of CLK. This is different from HD68000.
**This timing is not related to DMA Read/Write (Single Cycle) sequence.

Figure 5 AC Electrical Waveforms-DMA Read/Write (Single Cycle)



*Data are latched at the end of clock 7. This timing is the same as HD68000.
**This timing is not related to DMA Read/Write (Dual Cycle) sequence. This timing is only applicable when 1/8 clock pulse mode is selected.
***This timing is applicable when a bus exception occurs.

Figure 6 AC Electrical Waveforms-DMA Read/Write (Dual Cycle)

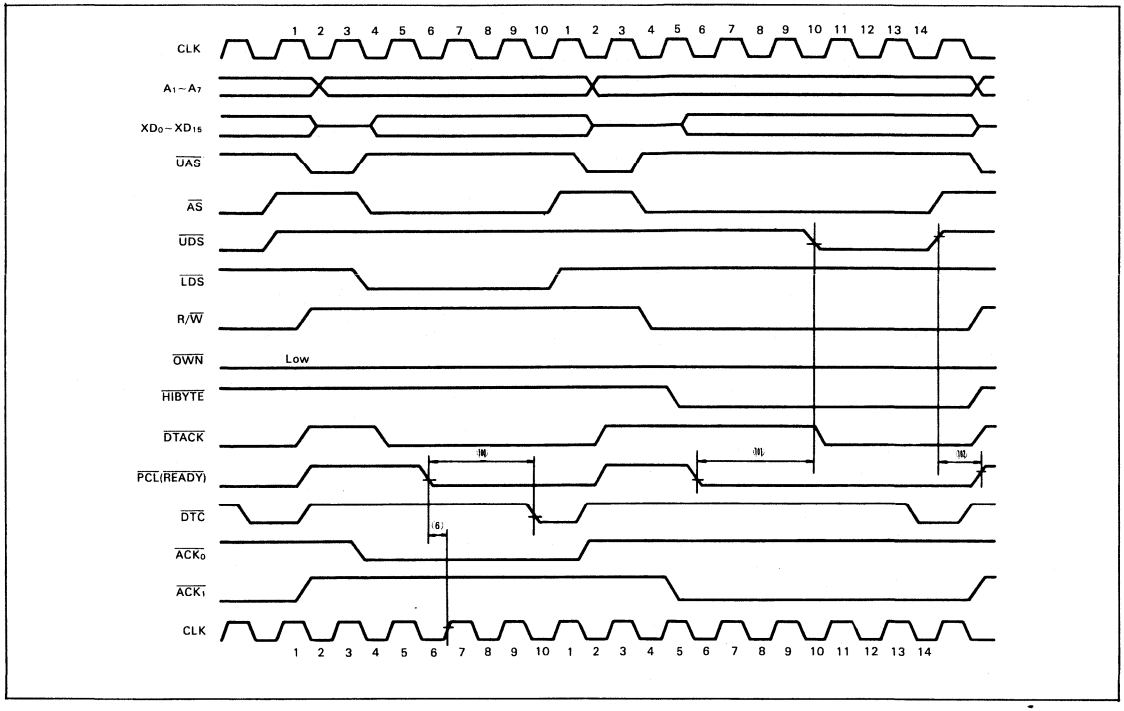


Figure 7 AC Electrical Waveforms-DMA Read/Write (Single Cycle with $\overline{ACK}$ and $\overline{READY}$)

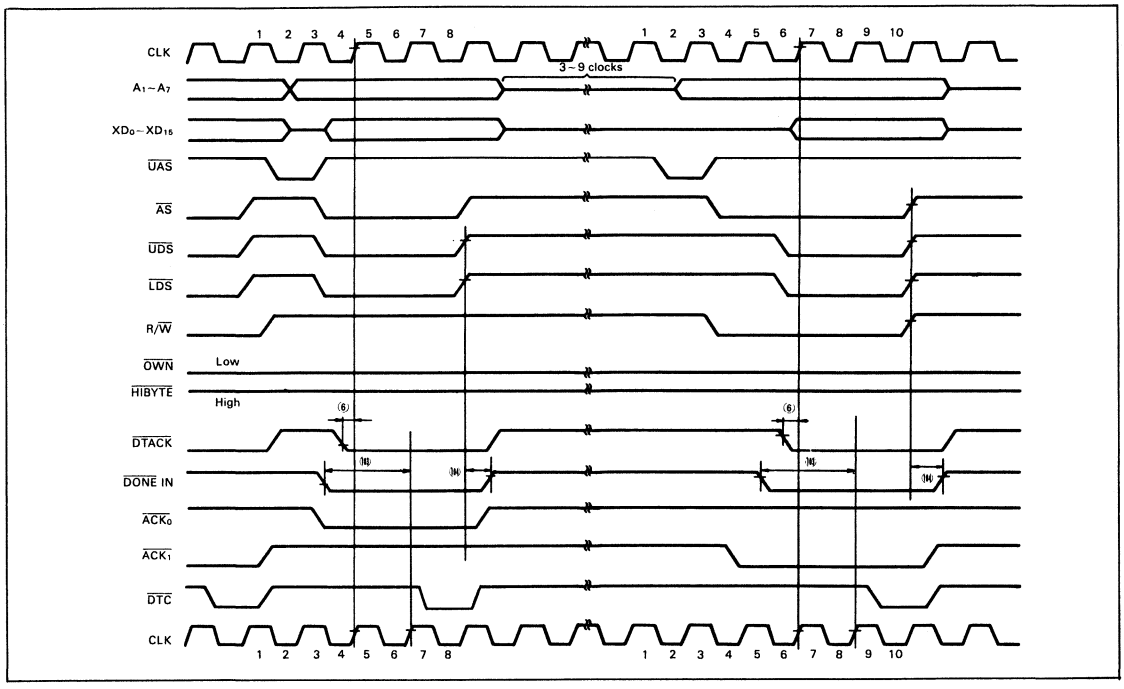


Figure 8 AC Electrical Waveforms-$\overline{\text{DONE}}$ Input

(NOTES for Figure 3 through 8)

1) Setup time for the asynchronous inputs $\overline{\text{BG}}$, $\overline{\text{BGACK}}$, $\overline{\text{CS}}$, $\overline{\text{IACK}}$, $\overline{\text{AS}}$, $\overline{\text{UDS}}$, $\overline{\text{LDS}}$, and R/$\overline{\text{W}}$ guarantees their recognition at the next falling edge of the clock. Setup time for $\overline{\text{BEC0}}$~$\overline{\text{BEC2}}$, $\overline{\text{REQ0}}$~$\overline{\text{REQ3}}$, $\overline{\text{PCL0}}$~$\overline{\text{PCL3}}$, $\overline{\text{DTACK}}$, and $\overline{\text{DONE}}$ guarantees their recognition at the next rising edge of the clock.
2) Timing measurements are referred to and from a low voltage of 0.8 volts and a high voltage of 2.0 volts.
3) These waveforms should only be referred in regard to the edge-to-edge measurement of the timing specifications. They are not intended as a functional description of the input and output signals. Refer to other functional descriptions and their related diagrams for device operation.

■ INTRODUCTION

The main purpose of a direct memory access (DMA) controller is to perform memory-to-memory, device-to-memory, and memory-to-device data transfers at high speed. The DMAC is required in any system including the device for data input/output like a floppy disk, a hard disk, a display terminal etc..

Figure 9 illustrates a typical system configuration using the DMAC. In this figure, the DMAC transfers blocks of data between the HDC and memory in a quick and efficient manner. Memory-to-memory data transfer is also provided by using data registers in the DMAC. Both 8-bit and 16-bit I/O devices are supportable. 8-/16-/32-bit data can be accessed in the DMA data transfer.

■ OPERATION MODES

The HD63450 DMAC operates through the MPU's writing operation into the internal control registers, then the DMAC will be in one of three operating modes :

1) MPU mode
   This is the state that the DMAC is chip-selected by another bus master in the system (MPU etc.), or that it is asserting the vector number during the interrupt acknowledge cycle.
2) DMA mode
   This is the state that the DMAC is acting as a bus master to perform an operand transfer. The DMA bus cycle refers to the bus cycle that is executed by the DMAC in the DMA mode.
3) IDLE mode
   This is the state that the DMAC is reset by an external device. The DMAC is waiting for an access by MPU or an operand transfer request from a peripheral. Many of the bus control signals are three-stated.

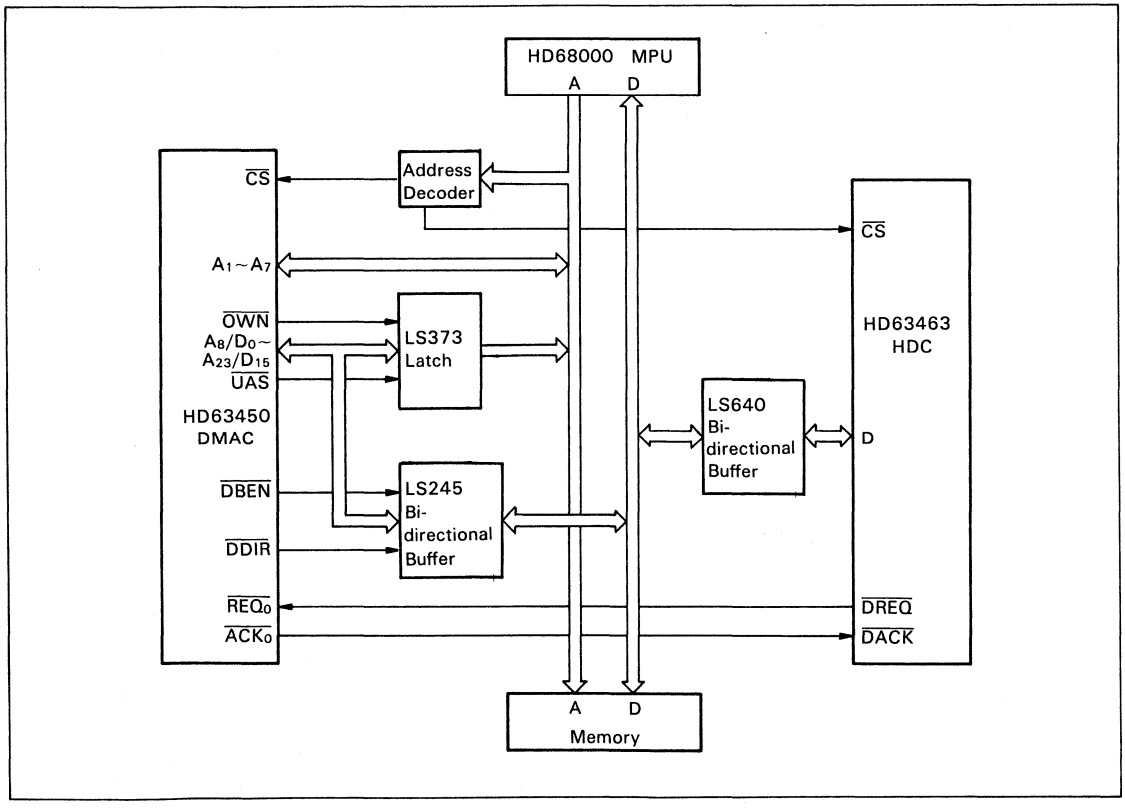


Figure 9 Typical System Configuration

■ SIGNAL DESCRIPTION

In this data sheet, the state of the signals is described with "active/inactive" or "assert/negate".
Figure 10 illustrates the input and output signals. Each function is described in the following.

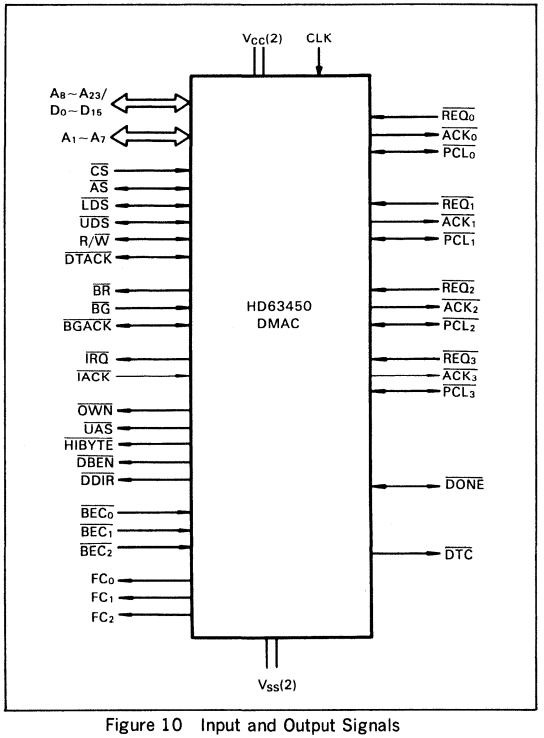


Figure 10 Input and Output Signals

● Address/Data Bus (A_8/D_0 through A_{23}/D_{15})

Input/Output	Three-statable

These lines are time multiplexed for address and data bus. The lines $\overline{DDIR}$, $\overline{DBEN}$, $\overline{UAS}$ and $\overline{OWN}$ are used to control the demultiplexing of the data and address lines externally. Demultiplexing is explained in the later section. The bi-directional data bus is used to transfer data between DMAC, MPU, memory and I/O devices.

Address lines are outputs to address memory and I/O devices.

● Address Bus (A_1 through A_7)

Input/Output	Three-statable

In the MPU mode, the DMAC internal registers are accessed with these lines and $\overline{LDS}$, $\overline{UDS}$. The address map for these registers is shown in Table 1. During a DMA bus cycle, A_1-A_7 are outputs containing the low order address bits of the location being accessed.

● Function Code (FC_0 through FC_2)

Output	Three-statable

These output signals provide the function codes during DMA bus cycles. They are three-stated except in the DMA bus cycles. They are used to control the HMCS68000 memories.

● Clock (CLK)

Input

This is the input clock to the HD63450, and should never be terminated at any time. This clock can be different from the MPU clock since HD63450 operates completely asynchronously.

● Chip Select ($\overline{CS}$)

Input

This input signal is used to chip select the DMAC in "MPU" mode. If the $\overline{CS}$ input is asserted during a bus cycle which is generated by the DMAC, the DMAC internally terminates the bus cycle and signals an address error. This function protects the DMAC from accessing its own register.

● Address Strobe ($\overline{AS}$)

Input/Output	Three-statable

In the "MPU mode", this line is an input indicating valid address input, and during the DMA bus cycle it is an output indicating valid address output from the DMAC on the address bus.

The DMAC monitors these input lines during bus arbitration to determine the completion of the bus cycle by the MPU or other bus masters.

● Upper Address Strobe ($\overline{UAS}$)

Output	Three-statable

This line is an output to latch the upper address lines on the multiplexed data/address lines. It is three-stated except in the "DMA mode".

● Own ($\overline{OWN}$)

Output	Three-statable

This line is asserted by the DMAC during DMA mode, and is used to control the output of the address line latch. This line may also be used to control the direction of bi-directional buffers when loads on $\overline{AS}$, $\overline{LDS}$, $\overline{UDS}$, R/$\overline{W}$ and other signals exceed the drive capability. It is three-stated in the "MPU mode" and the "IDLE mode".

● Data Direction ($\overline{DDIR}$)

Output	Three-statable

This line controls the direction of data through the bi-directional buffer which is used to demultiplex the data/address lines. It is three-stated during the "IDLE mode".

● Data Bus Enable ($\overline{DBEN}$)

Output	Three-statable

This line controls the output enable line of bi-directional buffers on the multiplexed data/address lines. It is three-stated during the "IDLE mode".

● High Byte ($\overline{HIBYTE}$)

Output	Three-statable

This line is used when the operand size is byte in the single addressing mode. It is asserted when data is present on the upper eight bits of the data bus. It is used to control the output of bidirectional buffers which connects the upper eight bits of the data bus with the lower eight bits. It is three-stated during the "MPU mode" and the "IDLE mode".

● **Read/Write (R/$\overline{W}$)**

Input/Output	Three-statable

This line is an input during the "MPU mode" and an output during the "DMA mode". It is three-stated during the "IDLE mode". It is used to control the direction of data flow.

● **Upper Data Strobe ($\overline{UDS}$), Lower Data Strobe ($\overline{LDS}$)**

Input/Output	Three-statable

These lines are extensions of the address lines indicating which byte or bytes of data of the addressed word are being addressed. These lines combined corresponds to address line A_0 in table 1.

● **Data Transfer Acknowledge ($\overline{DTACK}$)**

Input/Output	Three-statable

In the "MPU mode", this line is an output indicating the completion of Read/Write bus cycle by the MPU.

In the "DMA mode", the DMAC monitors this line to determine when a data transfer has completed. In the event that a bus exception is requested, except for $\overline{HALT}$, prior to or concurrent with $\overline{DTACK}$, the $\overline{DTACK}$ response is ignored and the bus exception is honored. In the "IDLE mode", this signal is three-stated.

● **Bus Exception Controls ($\overline{BEC_0}$ through $\overline{BEC_2}$)**

Input

These lines provide an encoded signal input indicating an exceptional condition in the DMA bus cycle. See bus exception section for details.

● **Bus Request ($\overline{BR}$)**

Output

This output line is used to request ownership of the bus by the DMAC.

● **Bus Grant ($\overline{BG}$)**

Input

This line is used to indicate to the DMAC that it is to be the next bus master. The DMAC cannot assume bus ownership until both $\overline{AS}$ and $\overline{BGACK}$ become inactive. Once the DMAC acquires the bus, it dose not continue to monitor the $\overline{BG}$ input.

● **Bus Grant Acknowledge ($\overline{BGACK}$)**

Input/Output	Three-statable

Bus Grant Acknowledge ($\overline{BGACK}$) is a bi-directional control line. As an output, it is generated by the DMAC to indicate that it is the bus master.

As an input, $\overline{BGACK}$ is monitored by the DMAC, in limited rate auto-request mode, to determine whether or not the current bus master is a DMA device or not. $\overline{BGACK}$ is also monitored during bus arbitration in order to assume bus ownership.

● **Interrupt Request ($\overline{IRQ}$)**

Output	Open drain

This line is used to request an interrupt to the MPU.

● **Interrupt Acknowledge ($\overline{IACK}$)**

Input

This line is an input to the DMAC indicating that the current bus cycle is an interrupt acknowledge cycle by the MPU. The DMAC responds the interrupt vector of the channel with the highest priority requesting an interrupt. There are two kinds of the interrupt vectors for each channel: normal (NIV) or error (EIV). $\overline{IACK}$ is not serviced if the DMAC has not generated $\overline{IRQ}$.

● **Channel Request ($\overline{REQ_0}$ through $\overline{REQ_3}$)**

Input

These lines are the DMA transfer request inputs from the peripheral devices.

These lines are falling edge sensitive inputs when the request mode is cycle steal. They are low-level sensitive when the request mode is burst.

● **Channel Acknowledge ($\overline{ACK_0}$ through $\overline{ACK_3}$)**

Output

These lines indicate to the I/O device requesting a transfer that the request is acknowledged and the transfer is to be performed. These lines may be used as a part of the enable circuit for bus interface to the peripheral.

● **Peripheral Control Line ($\overline{PCL_0}$ through $\overline{PCL_3}$)**

Input/Output	Three-statable

The four lines ($\overline{PCL_0}$~$\overline{PCL_3}$) are multi-purpose lines which may be individually programmed to be a START output, an Enable Clock input, a READY input, an ABORT input, a STATUS input, or an INTERRUPT input.

$\overline{REQ_x}$, $\overline{ACK_x}$, and $\overline{PCL_x}$ are provided for each channel.

● **Done ($\overline{DONE}$)**

Input/Output	Open Drain

As an output, this line is asserted concurrently with the $\overline{ACK_x}$ timing to indicate the last data transfer to the peripheral device. As an input, it allows the peripheral device to request a normal termination of the DMA transfer.

● **Device Transfer Complete ($\overline{DTC}$)**

Output	Three-statable

This line is asserted when the DMA bus cycle has terminated normally with no exceptions. It may be used to supply the data latch timing to the peripheral device. In this case, data is valid at the falling edge of $\overline{DTC}$.

■ INTERNAL ORGANIZATION

The DMAC has four independent DMA channels. Each channel has its own set of channel registers. These registers define and control the activity of the DMAC in processing a channel operation.

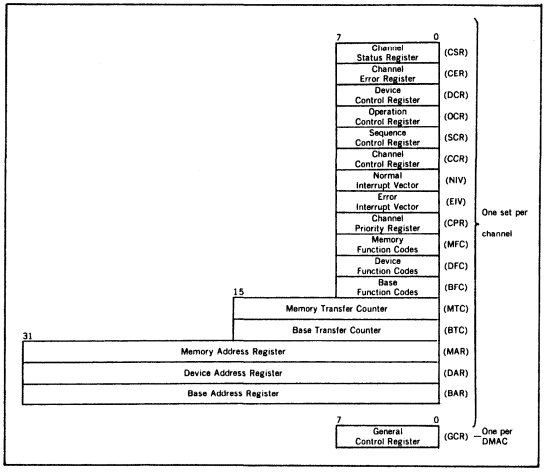


Figure 11 Internal Registers

● Register Organization

The internal register addresses are represented in Table 1. Address space not used within the address map is reserved for future expansion. A read from an unused location in the map results in a normal bus cycle with all ones for data. A write to one of these locations results in a normal bus cycle but no write occurs.

Unused bits of the defined registers in Table 1 are read as zeros.

Table 1 Internal Register Addressing Assignments

Register	Address Bits 7 6 5 4 3 2 1 0	Mode
Channel Status Register	c c 0 0 0 0 0 0	R/W*
Channel Error Register	c c 0 0 0 0 0 1	R
Device Control Register	c c 0 0 0 1 0 0	R/W
Operation Control Register	c c 0 0 0 1 0 1	R/W
Sequence Control Register	c c 0 0 0 1 1 0	R/W
Channel Control Register	c c 0 0 0 1 1 1	R/W
Memory Transfer Counter	c c 0 0 1 0 1 b	R/W
Memory Address Register	c c 0 0 1 1 s s	R/W
Device Address Register	c c 0 1 0 1 s s	R/W
Base Transfer Counter	c c 0 1 1 0 1 b	R/W
Base Address Register	c c 0 1 1 1 s s	R/W
Normal Interrupt Vector	c c 1 0 0 1 0 1	R/W
Error Interrupt Vector	c c 1 0 0 1 1 1	R/W
Channel Priority Register	c c 1 0 1 1 0 1	R/W
Memory Function Codes	c c 1 0 1 0 0 1	R/W
Device Function Codes	c c 1 1 0 0 0 1	R/W
Base Function Codes	c c 1 1 1 0 0 1	R/W
General Control Register	1 1 1 1 1 1 1 1	R/W

cc:
- 00-Channel #0
- 01-Channel #1
- 10-Channel #2
- 11-Channel #3

ss:
- 00-High-order
- 01-Upper middle
- 10-Lower middle
- 11-Low-order

b:
- 0-High-order
- 1-Low-order

*see Channel Status Register section in page 17

● Device Control Register (DCR)

The DCR is a device oriented control register. The XRM bits specify whether the channel is in burst or cycle steal request mode. The DTYP bits define what type of device is on the channel. If the DTYP bits are programmed to be a HMCS6800 device, the PCL definition is ignored and the $\overline{PCL}$ line is an Enable clock input. If the DTYP bits are programmed to be a device with $\overline{READY}$, the PCL definition is ignored and the $\overline{PCL}$ line is a $\overline{READY}$ input. The DPS bit defines the port size (eight or sixteen bits) of the peripheral device. (A port size is the largest data which the peripheral device can transfer during a DMA bus cycle.) The PCL bits define the function of the $\overline{PCL}$ line. If the DTYP bits are programmed to be HMCS6800 device, or Device with $\overline{ACK}$ and $\overline{READY}$, these definitions are ignored. The XRM bits are ignored if an auto-request mode (REQG=00 or 01 in Operation Control Register) is selected.

7	6	5	4	3	2	1	0
XRM		DTYP		DPS	0	PCL	

XRM (EXTERNAL REQUEST MODE)
- 00 Burst Transfer Mode
- 01 (undefined, reserved)
- 10 Cycle Steal Mode without Hold
- 11 Cycle Steal Mode with Hold

DTYP (DEVICE TYPE)
- 00 HD68000 compatible device, explicitly addressed (dual addressing mode)
- 01 HD6800 compatible device, explicitly addressed (dual addressing mode)
- 10 Device with $\overline{ACK}$, implicitly addressed (single addressing mode)
- 11 Device with $\overline{ACK}$ and $\overline{READY}$, implicitly addressed (single addressing mode)

DPS (DEVICE PORT SIZE)
- 0 8 bit port
- 1 16 bit port

PCL (PERIPHERAL CONTROL LINE)
- 00 Status Input
- 01 Status Input with Interrupt
- 10 1/8 Start Pulse
- 11 Abort Input

Bit 2 Not Used

● Operation Control Register (OCR)

The OCR is an operation control register. The DIR bit defines the direction of the transfer. The BTD bit defines the execution of the multi-block transfer with $\overline{DONE}$. The SIZE bits define the size of the operand. The CHAIN bits define the type of the CHAIN mode. The REQG bits define how requests for transfers are generated.

7	6	5	4	3	2	1	0
DIR	BTD	SIZE		CHAIN		REQG	

DIR (DIRECTION)
- 0 Transfer from memory to device (transfer from MAR address to DAR address)
- 1 Transfer from device to memory (transfer from DAR address to MAR address)

BTD (MULTI BLOCK TRANSFER WITH $\overline{\text{DONE}}$ MODE)
- 0 Terminates channel operation after the current DMA bus cycle completion
- 1 Restarts next block transfer after the current DMA bus cycle completion

(Note 2)

SIZE (OPERAND SIZE)
- 00 Byte (8 bits)
- 01 Word (16 bits)
- 10 Long Word (32 bits)
- 11 Byte Transfer without Packing (Port size : 8 bits)

CHAIN (CHAINING OPERATION)
- 00 Chain operation is disabled
- 01 (undefined, reserved)
- 10 Array Chaining
- 11 Linked Array Chaining

REQG (DMA REQUEST GENERATION METHOD)
- 00 Auto-request at transfer rate limited by General Control Register (Limited Rate Auto-Request)
- 01 Auto-request at maximum rate
- 10 $\overline{\text{REQ}}$ line requests an operand transfer
- 11 Auto-request the first operand, external request for subsequent operands

(Note 2) See Page 41 for details

● Sequence Control Register (SCR)

The SCR is used to define the sequencing of memory and device addresses.

7	6	5	4	3	2	1	0
0	0	0	0	MAC		DAC	

MAC (MEMORY ADDRESS COUNT)
- 00 Memory address register does not count
- 01 Memory address register counts up
- 10 Memory address register counts down
- 11 (undefined, reserved)

DAC (DEVICE ADDRESS COUNT)
- 00 Device address register does not count
- 01 Device address register counts up
- 10 Device address register counts down
- 11 (undefined, reserved)

Bits 7, 6, 5, 4 Not Used

● Channel Control Register (CCR)

The CCR is used to start or terminate the operation of a channel. This register also determines if an interrupt request is to be generated. Setting the STR bit causes immediate activation of the channel; the channel will be ready to accept request immediately. The STR and CNT bits of the register cannot be reset by a write to the register. The SAB bit is used to terminate the operation forcedly. Setting the SAB bit will reset STR and CNT. Setting the HLT bit will halt the channel operation, and clearing the HLT bit will resume the operation. Setting start bit must be done by byte access. Otherwise, timing error occurs.

7	6	5	4	3	2	1	0
STR	CNT	HLT	SAB	INT	0	0	0

STR (START OPERATION)
- 0 No operation is pending
- 1 Start operation

CNT (CONTINUE OPERATION)
- 0 No continuation is pending
- 1 Continue operation

HLT (HALT OPERATION)
- 0 Operation not halted
- 1 Operation halted

SAB (SOFTWARE ABORT)
- 0 Channel operation not aborted
- 1 Abort channel operation

INT (INTERRUPT ENABLE)
- 0 No interrupts enabled
- 1 Interrupts enabled

Bits 2, 1, 0 Not Used

● Channel Status Register (CSR)

The CSR is a register containing the status of the channel.

7	6	5	4	3	2	1	0
COC	BTC	NDT	ERR	ACT	DIT	PCT	PCS

COC (CHANNEL OPERATION COMPLETE)
- 0 Channel operation incomplete
- 1 Channel operation complete

BTC (BLOCK TRANSFER COMPLETE)
- 0 Block transfer incomplete
- 1 Block transfer complete

NDT (NORMAL DEVICE TERMINATION)
- 0 No normal device termination by $\overline{\text{DONE}}$ input
- 1 Device terminated operation normally by $\overline{\text{DONE}}$ input

ERR (ERROR BIT)
- 0 No errors
- 1 Error as coded in CER

ACT (CHANNEL ACTIVE)
- 0 Channel not active
- 1 Channel active

DIT ($\overline{\text{DONE}}$ INPUT TRANSITION)
- 0 No $\overline{\text{DONE}}$ input transition occurred
- 1 $\overline{\text{DONE}}$ input transition occurred when BTD bit is set

PCT ($\overline{\text{PCL}}$ TRANSITION)
- 0 No $\overline{\text{PCL}}$ transition occurred
- 1 $\overline{\text{PCL}}$ transition occurred

PCS (THE STATE OF THE $\overline{\text{PCL}}$ INPUT LINE)
- 0 $\overline{\text{PCL}}$ "Low"
- 1 $\overline{\text{PCL}}$ "High"

● Channel Error Register (CER)
The CER is an error condition status register. The ERR bit of CSR indicates if there is an error or not. Bits 0-4 indicate what type of error has occurred.

7	6	5	4	3	2	1	0
0	0	0	ERROR CODE				

Error Code

00000	No error
00001	Configuration error
00010	Operation timing error
00101	Address error in MAR
00110	Address error in DAR
00111	Address error in BAR
01001	Bus error in MAR
01010	Bus error in DAR
01011	Bus error in BAR
01101	Count error in MTC
01111	Count error in BTC
10000	External abort
10001	Software abort

Bits 7, 6, 5 Not Used

● Channel Priority Register (CPR)
The CPR is used to define the priority level of the channel. Priority level 0 is the highest and priority level 3 is the lowest priority.

7	6	5	4	3	2	1	0
0	0	0	0	0	0	CP	

CP (CHANNEL PRIORITY)

00	Priority level 0
01	Priority level 1
10	Priority level 2
11	Priority level 3

Bit 7 through 2 Not Used

● General Control Register (GCR)
The GCR is used to define what portion of the bus cycles is available to the DMAC for limited rate auto-request generation. GCR is also used to specify the hold time for cycle steal mode with hold.

7	6	5	4	3	2	1	0
0	0	0	0	BT		BR	

BT (BURST TIME)

The number of DMA clock cycles per burst that the DMAC allows in the auto-request at a limited rate of transfer is controlled by these two bits. The number is $2^{(BT+4)}$ (two to the BT +4 power).

BT	Clock Cycle
00	16 Clocks
01	32 Clocks
10	64 Clocks
11	128 Clocks

BR (BANDWIDTH RATIO)
The amount of the bandwidth utilized by the auto-request at a limited rate transfer is controlled by these two bits. The ratio is $2^{(BR+1)}$ (two to the BR+1 power).

BR	Bandwidth Ratio
00	50.00%
01	25.00%
10	12.50%
11	6.25%

The hold time for cycle steal mode with hold is defined to be minimum of 1 sample interval and maximum of 2 sample intervals. A sample interval is defined to be $2^{(BT+BR+5)}$ (two to the BT+BR+5 power) clock cycles.

Bit 7 through 4 Not Used

● Address Registers (MAR, DAR, BAR)
Three 32-bit registers are utilized to implement the Memory Address Register, Device Address Register, and the Base Address Register. Only the least significant twenty-four bits are connected to the address output pins. The content of the MAR is outputted when the memory is accessed in single or dual addressing mode. The content of the DAR is outputted when the peripheral device is accessed. The contents of the BAR is outputted when reading chain information from memory in the Array Chaining Mode or the Linked Array Chaining Mode. It is also used to set the top address of the next block transfer in Continue mode.

● Function Code Registers (MFC, DFC, BFC)
The DMAC has three function code register per channel: the Memory Function Code Register (MFC), Device Function Code Register (DFC), and the Base Function Code Register (BFC). The contents of these registers are outputted from FC_0 through FC_2 lines when an address is outputted from MAR. DAR, or BAR, respectively. The BFC is also used to set the MFC for the transfer of the next data block in the Continue mode.

7	6	5	4	3	2	1	0
0	0	0	0	0	FC2	FC1	FC0

Bit 3 through 7 Not Used

● Transfer Count Registers (MTC, BTC)
Each channel has two 16-bit counters: the Memory Transfer Counter (MTC) and the Base Transfer Counter (BTC). The MTC counts the number of transfer words in one block, and is decreased by one for every operand transfer.

The BTC is used to count the number of data blocks in the Array Chaining Mode. BTC is also used to set the number of operands to transfer for the next data block in the Continue Mode.

The specifiable number is up to "2^{16}-1".

● Interrupt Vector Registers (NIV, EIV)
Each channel has a Normal Interrupt Vector register and an Error Interrupt Vector register.

When an interrupt acknowledge cycle occurs, an interrupt vector is outputted from one of those registers. If the error bit (CSR) is set for the channel with interrupt pending, then content of EIV is outputted, otherwise content of NIV is outputted.

■ OPERATION DESCRIPTION

A DMAC channel operation proceeds in three principal phases. During the initialization phase, the MPU sets the channel control registers, supplies the initial address and the number of transfer words, and starts the channel. During the tranfer phase, the DMAC accepts requests for data operand transfers, and provides addressing and bus controls for the trasnfers. The termination phase occurs after the operation is completed.

This section describes DMAC operations. A description of the MPU/DMAC communication is given first. Next, the transfer phase is covered, including how the DMAC recognizes requests and how the DMAC arranges for data transfer. Following this, the initialization phase is described. The termination phase is covered, introducing chaining, error signaling, and bus exceptions. A description of the channel priority scheme rounds out the section.

■READ/WRITE OF THE DMAC REGISTERS BY MPU

The MPU reads and writes the DMAC internal registers and controls the DMA transfer.

Figure 12 indicates the timing diagram when the MPU reads the contents of the DMAC register. The MPU outputs A_1-A_{23}, FC_0-FC_2, $\overline{AS}$, R/$\overline{W}$, $\overline{UDS}$, and $\overline{LDS}$, and accesses the DMAC internal register. The specific internal register is selected by A_1-A_7, $\overline{LDS}$ and $\overline{UDS}$. The $\overline{CS}$ and $\overline{IACK}$ lines are generated by the external circuit with A_8-A_{23} and FC_0-FC_2. The DMAC outputs data on the data bus, together with $\overline{DDIR}$, $\overline{DBEN}$ and $\overline{DTACK}$. The $\overline{DDIR}$ and $\overline{DBEN}$ control the bi-directional buffer on the bus and the $\overline{DTACK}$ indicates that the data has been sent or received by the DMAC. Read Cycle is eighteen CLKs.

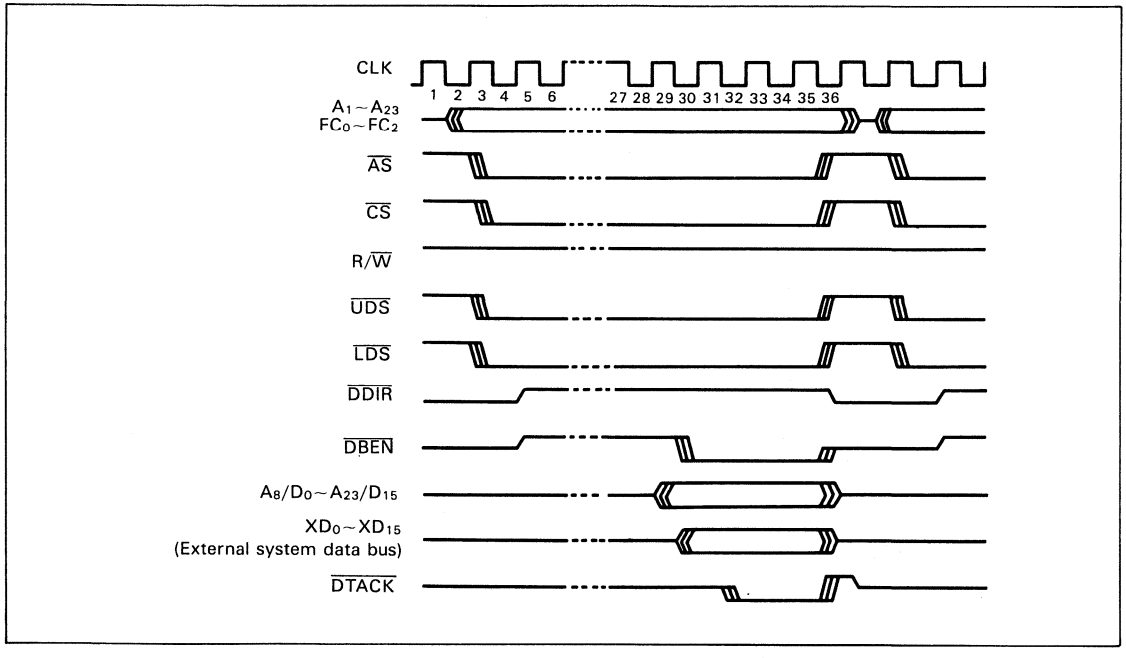


Figure 12 MPU Read from DMAC-Word

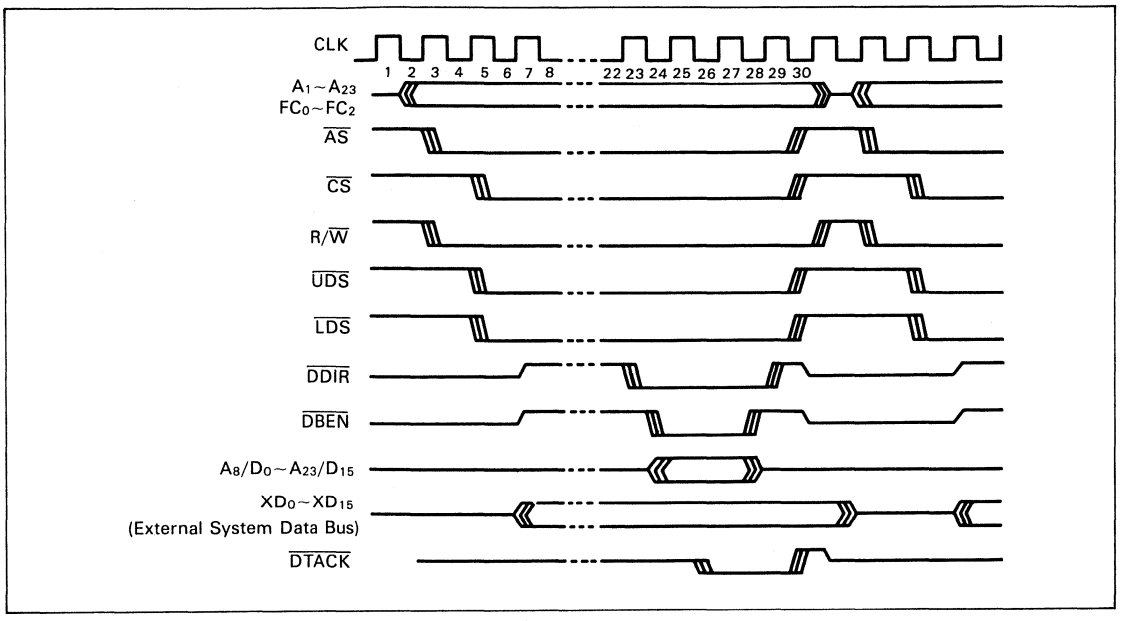


Figure 13 MPU Write to DMAC-Word

Figure 13 shows the MPU write cycle. Write cycle is fifteen CLKs.

Note the following points.

(1) The clock reference shown in this figure is the DMAC input clock.
(2) The $\overline{DDIR}$ and the $\overline{DBEN}$ are three-stated at the beginning which detects $\overline{CS}$ and the ending of the cycle.
(3) During the MPU read cycle, the $\overline{DTACK}$ is asserted after the data is valid on the system bus.
(4) During the MPU write cycle, the $\overline{DDIR}$ line will be driven low to direct the data buffers toward to DMAC before the buffers are enabled.
(5) During the MPU write cycle, the DMAC will latch the data before asserting $\overline{DTACK}$. Then it will negate $\overline{DBEN}$ and $\overline{DDIR}$ in the proper order.
(6) After the MPU cycle and the $\overline{LDS}$ and the $\overline{UDS}$ are negated by the MPU, the DMAC will put $\overline{DBEN}$, $\overline{DDIR}$ and the address data lines to a high impedance state.
(7) $\overline{DTACK}$ will once go "High" and then to a high impedance state after negating $\overline{LDS}$ and $\overline{UDS}$.

■BUS ARBITRATION

The followings are the description of the bus arbitration. The DMAC must obtain the ownership of the bus in order to transfer data. Figure 14 indicates the DMAC bus arbitration timing. It is completely compatible with that of HD68000 MPU. The DMAC asserts the Bus Grant ($\overline{BG}$) to request the bus mastership. The

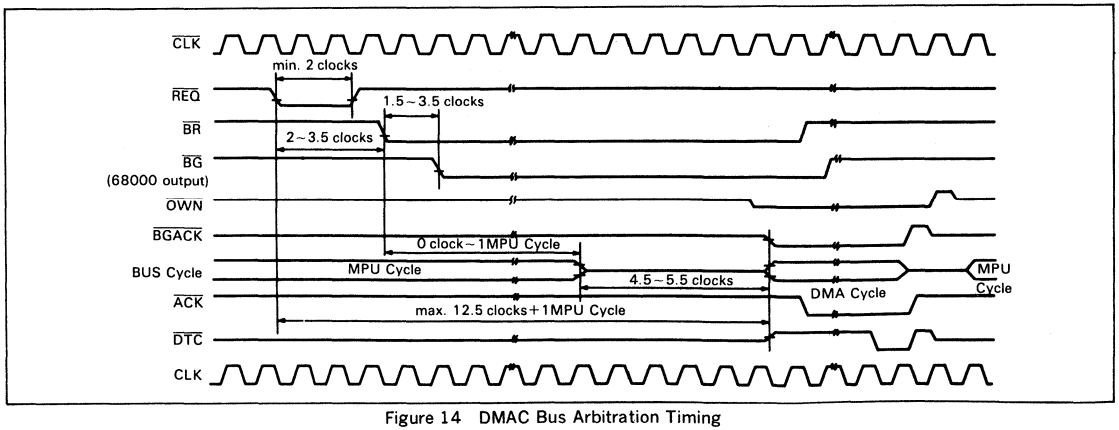


Figure 14 DMAC Bus Arbitration Timing

MPU recognizes the request and asserts $\overline{BG}$, then it grants the ownership in the next bus cycle. After the end of the current cycle ($\overline{AS}$ is negated), the MPU relinquishes the bus to the DMAC. The DMAC asserts the bus grant acknowledge ($\overline{BGACK}$) to indicate that it has the bus ownership. A half clock before $\overline{BGACK}$ is asserted, the DMAC asserts $\overline{OWN}$. $\overline{OWN}$ is kept asserted for a half clock after $\overline{BGACK}$ is negated at the end of the DMA cycle. $\overline{BR}$ is negated one clock after $\overline{BGACK}$ is asserted.

■DEVICE/DMAC COMMUNICATION

Communication between peripheral devices and the DMAC is accomodated by five signal lines. Each channel has $\overline{REQ}$, $\overline{ACK}$ and $\overline{PCL}$, and the last two lines, the $\overline{DONE}$ and $\overline{DTC}$ lines, are shared among the four channels.

● Request ($\overline{REQ}$)

The peripheral devices assert $\overline{REQ}$ to request data transfers. See the "Requests" section for details.

● Acknowledge ($\overline{ACK}$)

This line is used to implicitly address the device which is transferring the data (This device is not selected by address lines). It is also asserted when the content of DAR is outputted during memory-to-memory transfer except for the auto-request mode at a limited rate or at the maximum rate.

● Peripheral Control Line ($\overline{PCL}$)

The function of this line is quite flexible and is determined by the DCR (Device Control Register).

The DTYP bits of the DCR define what type of device is on the channel. If the DTYP bits are programmed to be a HMCS6800 device, the PCL difinition is ignord and the $\overline{PCL}$ line is an Enable clock (E clock) input. If the DTYP bits are programmed to be a device with $\overline{READY}$, the PCL definition is ignored and the $\overline{PCL}$ line is a ready input.

(1) $\overline{PCL}$ as a Status Input

The $\overline{PCL}$ line may be programmed as a status input. The status level of this line can be determined by the PCS bit in the CSR, regardless of the PCL function determined by the DCR. If a negative transition occurs and remains stable for a minimum of two clocks, the PCT bit of the CSR is set. This PCT bit is cleared by resetting the DMAC or writing "1" to the PCT bit.

(2) $\overline{PCL}$ as an Interrupt

The $\overline{PCL}$ line may be programmed to generate an interrupt on a negative transition. This enables an interrupt which is requested if the PCT bit of the CSR is set. When using this function, it is necessary to reset the PCT bit in the CSR before the PCL bit in the DCR is set to interrupt, in order to avoid assertion of IRQ line at this time.

(3) $\overline{PCL}$ as a 1/8 Starting Pulse

The $\overline{PCL}$ line may be programmed to output a 1/8 starting pulse. This active low starting pulse is outputted when a channel is activated, and is "Low" for a period of four clock cycles.

(4) $\overline{PCL}$ as an Abort Input

The $\overline{PCL}$ line may be programmed to be a negative transition abort input which terminates an operation by setting the external abort error in CER. It is necessary to reset the PCT bit in the CSR before activating the channel (Setting the ACT bit of CCR) so that the channel operation is not immediately aborted.

(5) $\overline{PCL}$ as an Enable Clock (E Clock) Input

If the DTYP bits are programmed to be a HMCS6800 device, the PCL definition is ignored and the $\overline{PCL}$ line is an Enable Clock input. The Enable clock downtime must be as long as five clock cycles, and must be high for a minimum of three DMAC clock cycles, but need not be synchronous with the DMAC's clock.

(6) $\overline{PCL}$ as a $\overline{READY}$ Input

If the DTYP bits are programmed to be a device with $\overline{READY}$, the PCL definition is ignored and the $\overline{PCL}$ line is a $\overline{READY}$ input. The $\overline{READY}$ is an active low input.

● DONE ($\overline{DONE}$)

This line is an active low Input/Output signal with an open drain. It is asserted when the memory transfer count is exhausted in a single block transfer. In the chaining operation, $\overline{DONE}$ is asserted only at the last transfer to the peripheral device of the last data block. In the continue mode, $\overline{DONE}$ is asserted for each data block. It is asserted and negated in coincident with the $\overline{ACK}$ line for the last data transfer to the peripheral device. It is also outputted in coincident with the $\overline{ACK}$ line of the last bus cycle, in which the address is outputted from the DAR, in the memory-to-memory transfer (dual addressing mode) that uses the $\overline{ACK}$ line.

The DMAC also monitors the state of the $\overline{DONE}$ line during the DMA bus cycle. If the device asserts $\overline{DONE}$ during $\overline{ACK}$ active, the DMAC will terminate the operation after the transfer of the current operand. If $\overline{DONE}$ is asserted on the first byte of 2-byte operation or the first word of long word operation, the DMAC does not terminate the operation before the whole operand transfer is completed. If $\overline{DONE}$ is asserted, then the DMAC terminates the operation by clearing the ACT bit of the CSR, and setting the COC and NDT bits of the CSR. If both the DMAC and the device assert $\overline{DONE}$, the device termination is not recognized, but the channel operation does terminate. $\overline{DONE}$ is outputted again for the retry exceptions bus cycles.

The case that the multi-block transfer with $\overline{DONE}$ mode is set is described later.

● Data Transfer Complete ($\overline{DTC}$)

$\overline{DTC}$ is an active low signal which is asserted when the actual data transfer is accomplished. It is also asserted in the bus cycle which read a chain information from memory in the Chaining mode. However, if exceptions are generated and the DMA bus cycle terminates, $\overline{DTC}$ is not asserted. $\overline{DTC}$ is asserted one half clock before $\overline{LDS}$ and $\overline{UDS}$ are negated, and negated one half clock after $\overline{LDS}$ and $\overline{UDS}$ are negated.

■REQUESTS

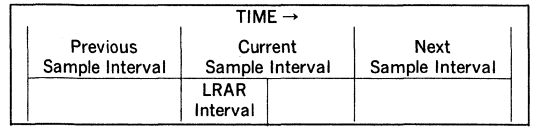
Requests may be externally generated by circuitry in the peripheral device, or internally generated by the auto-request mechanism. The REQG bits of the OCR determine these modes. The DMAC also supports an operation in which the DMAC auto-requests the first transfer and then waits for the peripheral device to request the following transfers.

● Auto-request Transfers

The auto-request mechanism provides generation of requests within the DMAC. These requests can be generated at either of two rates: maximum-rate and limited-rate. In the former case, the channel always has a request pending.

The limited rate auto-request functions by monitoring the bus utilization.

(1) Limited-rate Auto-request



In the limited-rate auto-request, the DMAC devides time into equal length sample intervals by counting clock cycles. The end of one sample interval makes the beginning of the next. During a sample interval, the DMAC monitors, by means of $\overline{BGACK}$ pin, the system bus activity of the DMAC and other bus master devices. At the end of the sample interval, decision is made whether or not to perform the channel's data transfer during the next sample interval. Namely, based on the activity of the DMAC or other bus master devices during the current sample interval, the DMAC allows limited-rate auto-requests for some initial portion of the next sample interval.

The length of the sample interval, and the lenght of the limited-rate auto-request generation period (the limited-rate auto-request interval) are controlled by the BT and BR bits in the GCR. The length in clock cycles of the limited-rate auto-request interval is $2^{(BT+4)}$ (2 raised to the BT+4 power). For example, if BT equals 2 and the DMA utilization of the bus was low during the previous sample interval, then the DMAC generates the auto-request transfers during the first 64 clock cycles.

The ratio of the length of the sample interval to the length of the limited-rate auto-request interval is controlled by the BR bits. The ratio of the system bus utilization of the MPU to other bus master devices including the DMAC is $2^{(BR+1)}$ (2 raised to the BR +1 power). If the fraction of DMA clock cycles during the sample interval exceeds the programmed utilization level, the DMAC will not allow limited-rate auto-requests during the next sample interval.

For example, if BR equals 3, then at most one out of 16 clock cycles during a sample interval can be used by the DMAC and other bus master devices, and still the DMAC would allow limited rate auto-request during the next sample interval. Therefore, from the viewpoint of long period, the ratio of the system bus utilization of the MPU to I/O devices including the DMAC is about 16 : 1. The sample interval length is not a direct parameter, but it is equal to $2^{(BT+BR+5)}$ clock cycles. Thus, the sample interval can be programmed between 32 and 2048 clock cycles.

The DMAC uses the $\overline{BGACK}$ to differentiate between the MPU bus cycle and DMAC or other bus master devices. If $\overline{BGACK}$ is active, then the DMAC assumes that the bus is used by a DMAC or other bus master devices. If it is inactive, then the DMAC assumes that it is used by the MPU.

(2) Maximum-rate Auto-request

If the REQG bits in the OCR indicate auto-request at the maximum rate, the DMAC acquires the bus after the start bit is set and keeps it until the data transfer is completed.

If a request is made by another channel of higher priority, the DMAC services that channel and then resumes the auto-request sequence. If two or more channels are set to equal priority level and maximum rate auto-request, then the channels will rotate in a "round robbin" fashion.

If the HMCS68000 compatible device is connected to a channel, the $\overline{ACK}$ line is held inactive during an auto-request operation. Consequently, any channel may be used for the memory-to-memory transfer with the auto-request function in addition to the operation of data transfer between memory and peripheral device with using the $\overline{REQ}$ pin. Refer to Figure 15 for the timing of the memory-to-memory transfer. In this mode, the $\overline{ACK}$, $\overline{HIBYTE}$ and $\overline{DONE}$ outputs are always inactive.

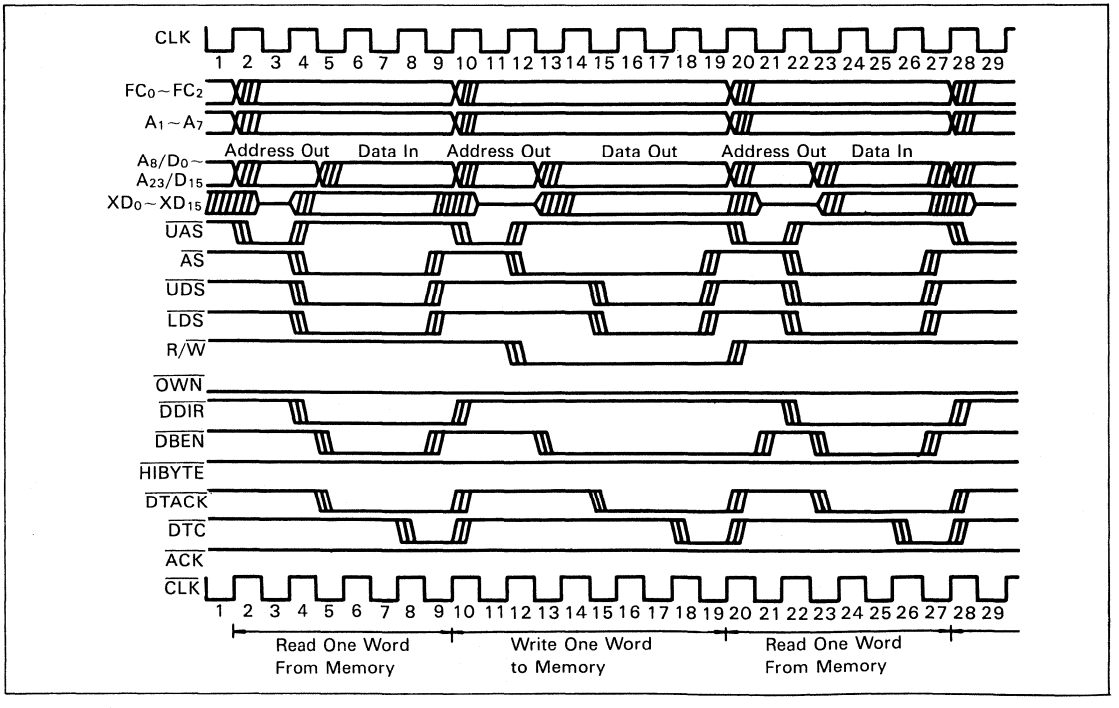


Figure 15 Memory-to-Memory Transfer Read-Write-Read Cycles

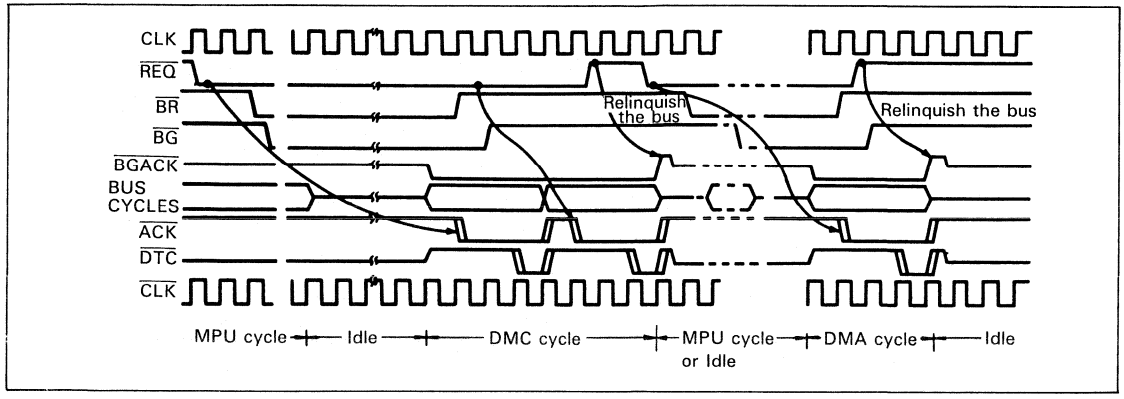


Figure 16 Burst Mode Request Timing (Only one channel is active)

● External Requests

If the REQG bits of the OCR indicate that the $\overline{REQ}$ line generates requests, the transfer requests are generated externally. The request line associated with each channel allows the device to externally generate requests for DMA transfers. When the device wants an operand transferred, it makes a request by asserting the request line. The external request mode is determined by the XRM bits of the DCR, which allows both burst and cycle steal request modes. The burst request mode allows a channel to request the transfer of multiple operands using consecutive bus cycles. The cycle steal request mode allows a channel to request the transfer of a single operand. The followings are the description of the burst and the cycle steal modes.

(1) Burst Request Recognition

In the burst request mode, the $\overline{REQ}$ line is an active low input. The level sampled at the rising edge of the clock. Once the burst request is asserted, it needs to be held low until the first DMA bus cycle starts in order to insure at least one data transfer operation. In order to stop the burst mode transfer after the current bus cycle, the $\overline{REQ}$ line has to be negated one clock before the $\overline{DTC}$ output clock of this cycle. Refer to Figure 16 or the burst mode timing.

(2) Cycle Steal Request Recognition

In the cycle steal request mode, the peripheral device requests the DMA transfer by generating an falling edge at the $\overline{REQ}$ line. The $\overline{REQ}$ line needs to be held "low" for at least 2 clock cycles. In the cycle steal mode, if the $\overline{REQ}$ line changes from "High" to "Low" between $\overline{ACK}$ output and one clock before the clock that outputs $\overline{DTC}$, then the next DMA transfer is performed without relinquishing the bus. If the bus is not relinquished, then maximum of 5 idle clocks is inserted between bus cycles.

Refer to Figure 17 for the request timing of the cycle steal mode. If the XRM bits specify cycle steal without hold, the DMAC will relinquish the bus. If the XRM bits specify cycle steal with hold, the DMAC will hold the bus and wait for the next $\overline{REQ}$ input for at least 1 sample interval after the current bus cycle completion. The allowable hold time is up to 2 sample intervals.

Figure 18 shows the request timing in the cycle steal bus hold. If the $\overline{REQ}$ is inputted during the hold time, the $\overline{ACK}$ is outputted after a maximum of 7.5 clock cycles from the picked-up clock.On the cycle steal with hold mode, the DMAC will hold the bus even when the transfer count is exhausted and the last data has been transferred. If DMA transfer is requested from other channel during this period, they are executed normally.

(3) Request Recognition in Dual-address Transfers

In a following section, dual-address transfers are defined. Dual address transfer is an exception to the request recognition rules

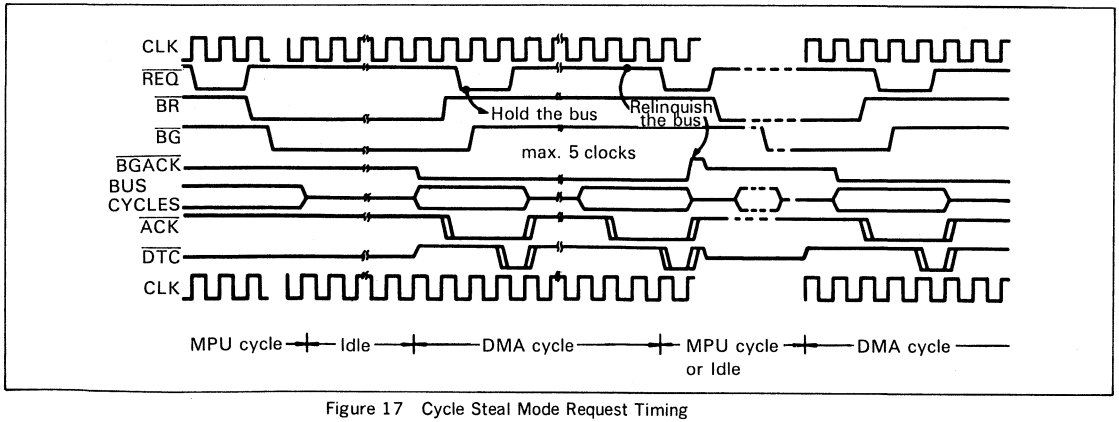


Figure 17 Cycle Steal Mode Request Timing

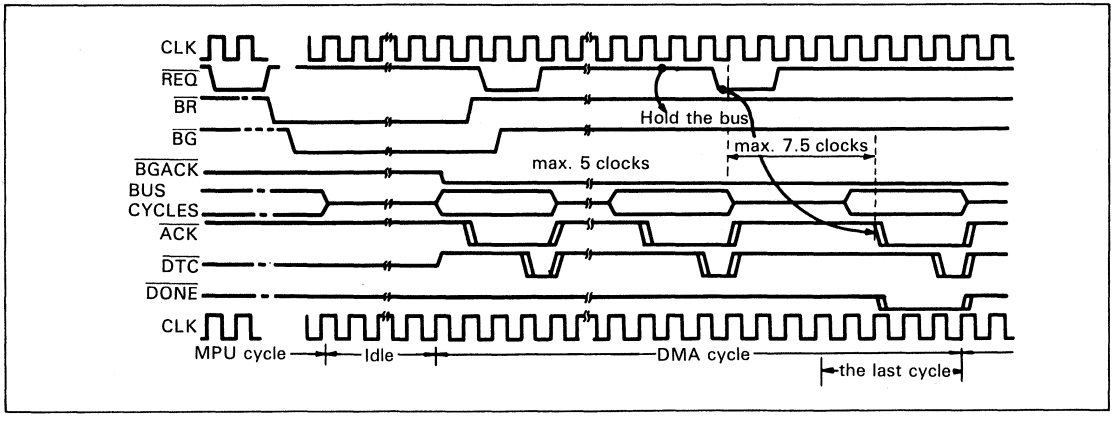


Figure 18 Cycle Steal Bus Hold Mode Request Timing

in the previous paragraphs. In the cycle steal request mode, when there are two or more transfers between the DMAC and the peripheral device during one operand transfer, the request is not recognized until the last transfer between the DMAC and the I/O device starts.

● Mixed Request Generation

A single channel can mix two request generation methods. By programming the REQG bits of the OCR to "11", when the channel is started, the DMAC auto-requests the first transfer. Subsequent requests are then generated externally by the device. The $\overline{ACK}$ and $\overline{PCL}$ lines perform their normal functions in this operation.

■DATA TRANSFERS

All DMAC data trasfers are assumed to be between memory and the peripheral device. The word "memory" means a 16-bit HMCS68000 bus compatible device. By programming the DCR, the characteristics of the peripheral device may be assigned. Each channel can communicate using any of the following protocols.

DTYP	Device Type	
00	HMCS68000 compatible device	Dual Addressing
01	HMCS6800 compatible device	
10	Device with $\overline{ACK}$	Single Addressing
11	Device with $\overline{ACK}$ and $\overline{READY}$	

● Dual Addressing

HMCS68000 and HMCS6800 compatible devices may be explicity addressed. This means that before the peripheral transfers data, a data register within the device must be addressed. Because the address bus is used to address the peripheral, the data cannot be directly transferred to/from the memory because the memory also requires addressing. Instead, the data is transferred from the source to the DMAC and held in an internal DMAC holding register. A second bus transfer between the DMAC and the destination is then required to complete the operation. Because both the source and destination of the transfer are explicitey addressed, this protocol is called dual addressing.

(1) HMCS68000 Compatible Device Transfers

In this operation, when a request is received, the bus is obtained and the transfer is completed using the protocol as shown in Figures 19 and 20. Figures 21 through 24 show the transfer timings. Figures 21 and 24 show the operation when the memory is the source and the peripheral device is the destination. Figures 22 and 23 show the transfer in the opposite direction. The peripheral device is a 16-bit device in Figures 21 and 22, and a 8-bit device in Figures 23 and 24.

(2) HMCS6800 Compatible Device Transfers

When a channel is programmed to perform HMCS6800 compatible transfers, the $\overline{PCL}$ line for that channel is defined as an Enable clock input. The DMAC performs data transfers between itself and the peripheral device using the HMCS6800 bus protocol, with the $\overline{ACK}$ output providing the $\overline{VMA}$ (valid memory address) signal. Figure 25 illustrates this protocol. Refer to Figure 26 for the read cycle timing and Figure 27 for the write cycle timing. In Figure 26, the DMAC latches the data at the falling edge of clock 19, so a latch to hold the data is necessary as shown in the figure.

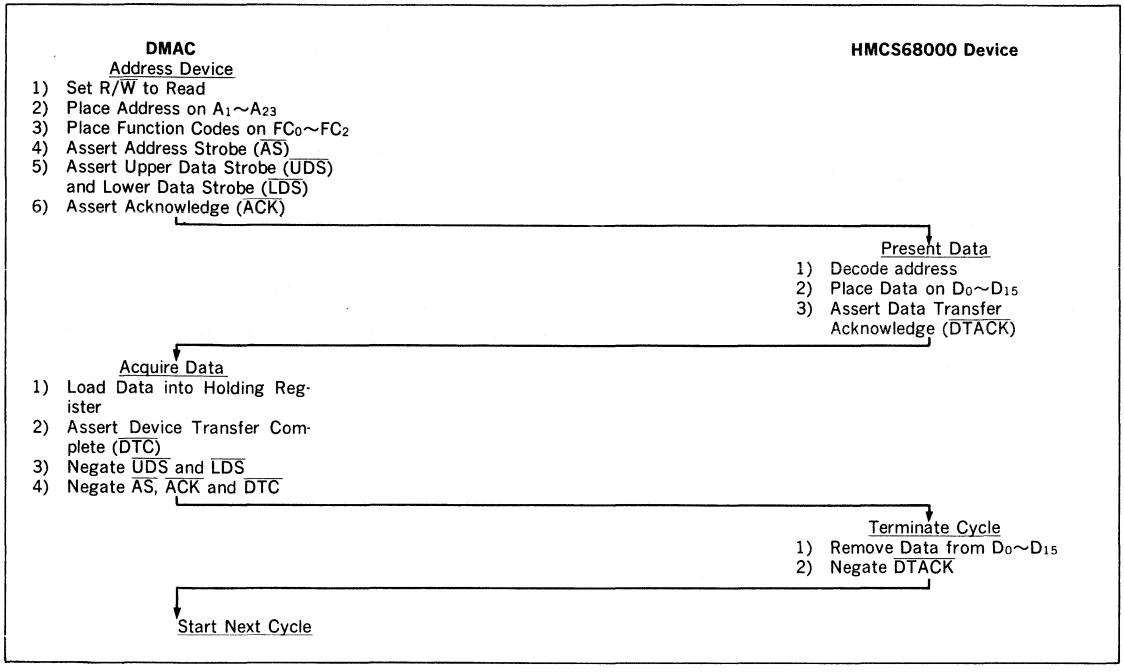


Figure 19 Word Read Cycle Flowchart HMCS68000 Type Device

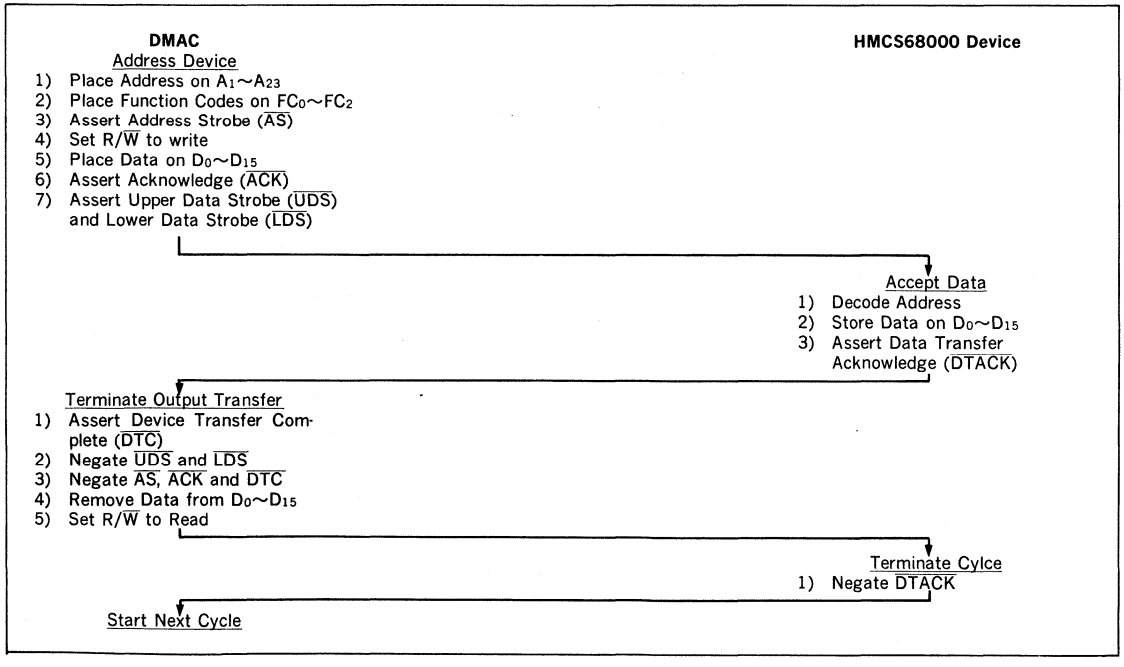


Figure 20 Word Write Cycle Flowchart HMCS68000 Type Device

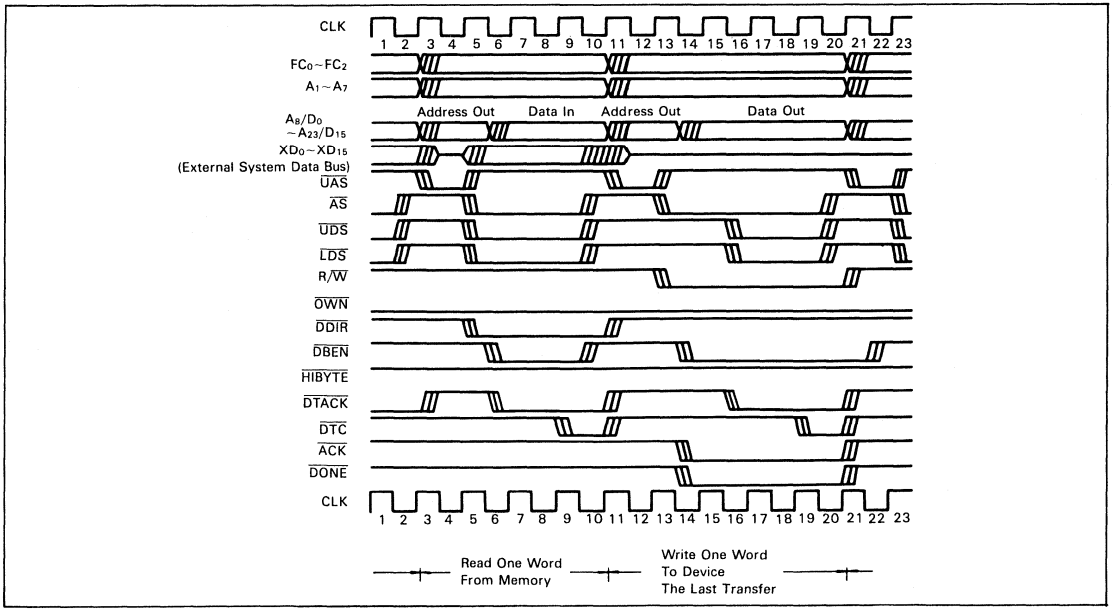


Figure 21 Dual Addressing Mode, Read/Write Cycle, Destination=16-bit Device, Word Operand

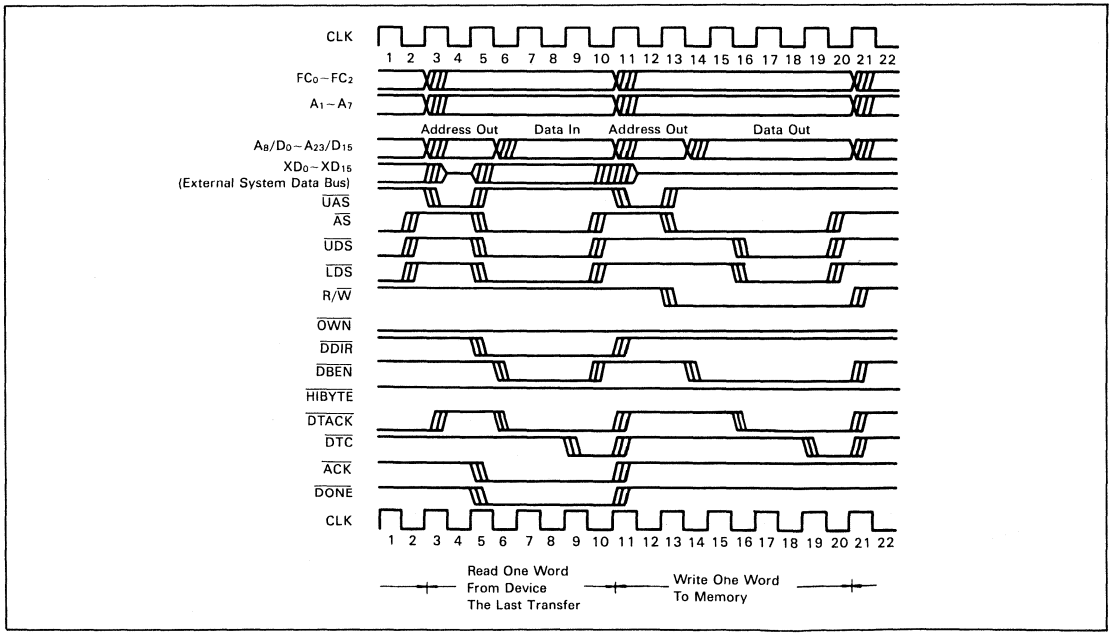


Figure 22 Dual Addressing Mode, Read/Write Cycle, Source=16-bit Device, Word Operand

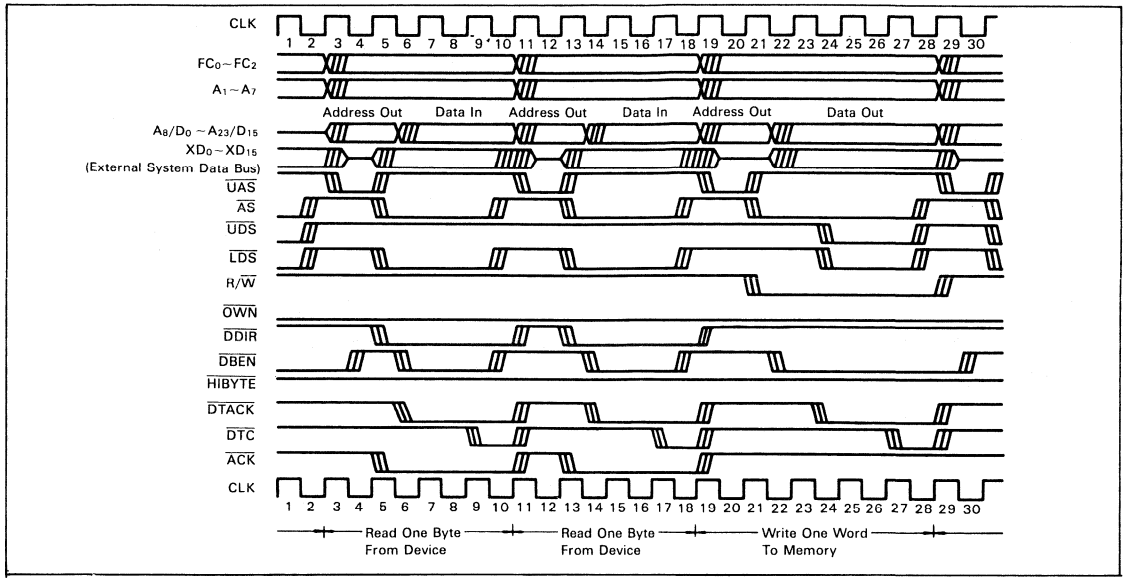


Figure 23 Dual Addressing Mode, Read/Write Cycle, Source=8-bit Device, Word Operand

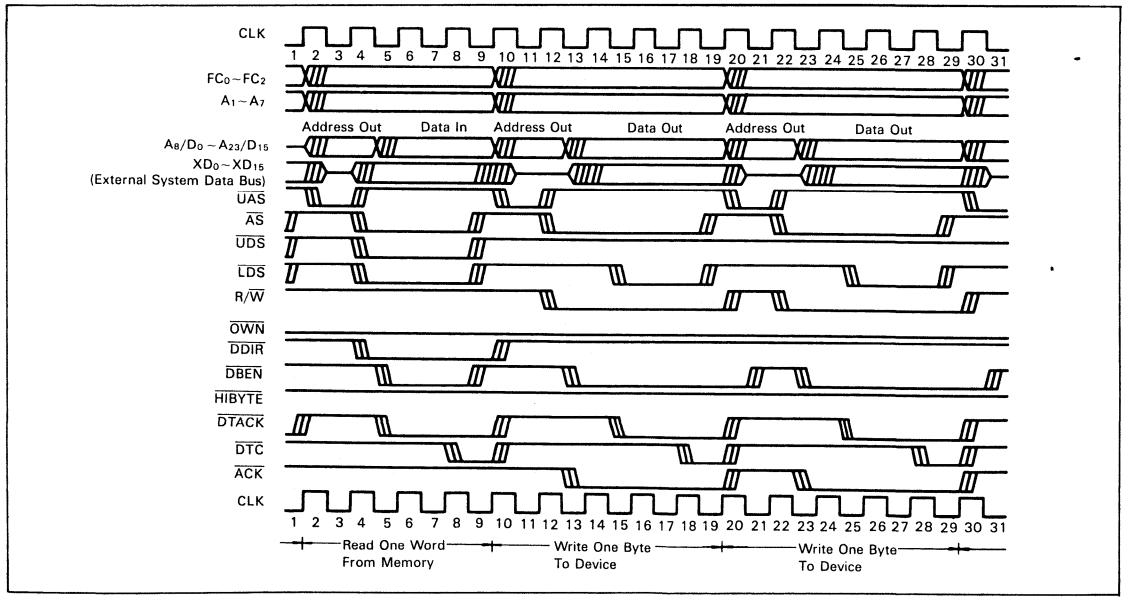


Figure 24 Dual Addressing Mode, Read/Write Cycle, Destination=8-bit Device, Word Operand

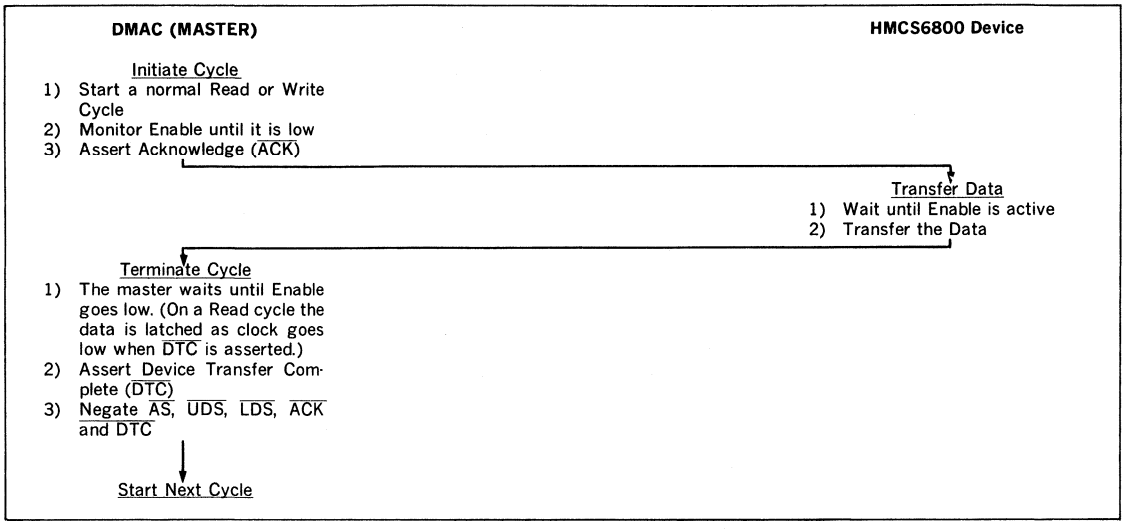


Figure 25 HMCS6800 Cycle Flowchart

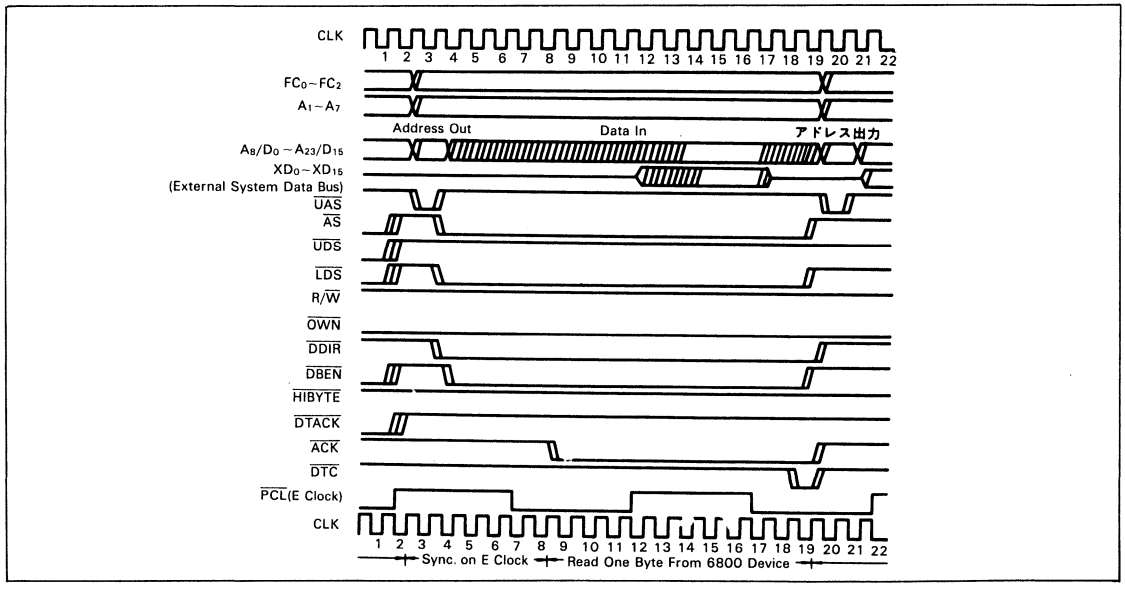


Figure 26 Dual Addressing Mode, HMCS6800 Compatible Device, Read Cycle

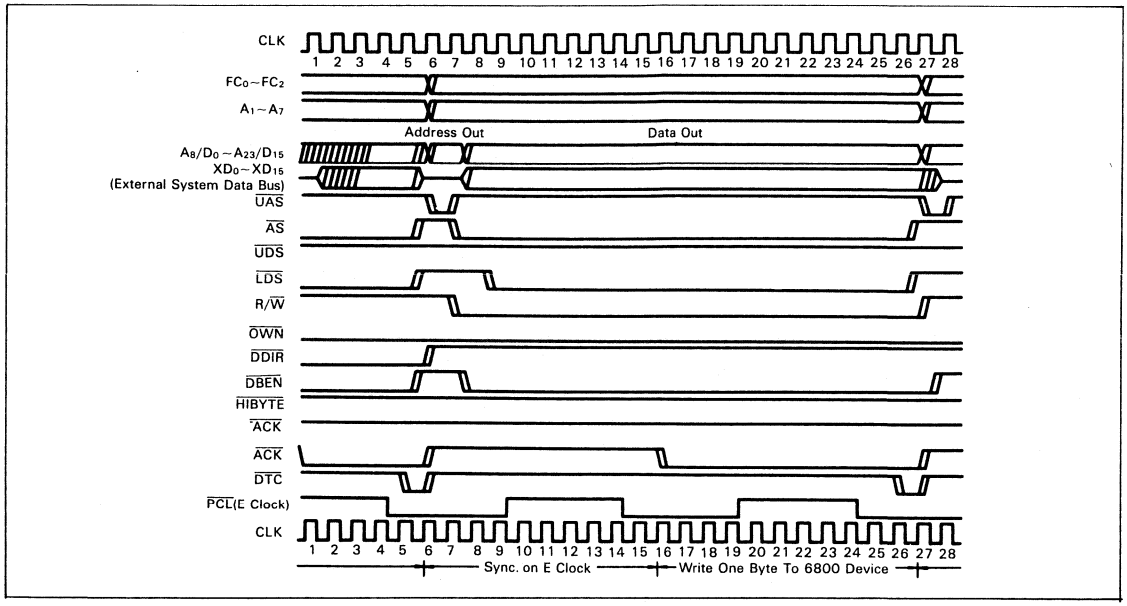


Figure 27 Dual Addressing Mode, HMCS6800 Compatible Device, Write Cycle

(3) An Example of a Dual Address Transfer

This scetion contains an example of a dual address transfer using Table 2 of Dual-Address Sequencing. The transfer mode of this example is the following.

1. Device Port size=8 bits
2. Operand size=Long Word (32 bits)
3. Memory to Device Transfer
4. Source (Memory) Counts up, Destination (Device) Counts Down

In this mode, a data transfer from the source (memory) is done according to the 6th row of Table 2, since the port size of the memory is always 16 bits. A data transfer to the destination (device) is done according to the 3rd row of Table 2. Table 3 shows the data transfer sequence.

The port size in Table 2 is not related to the DPS bit of the DCR. The DPS defines the port size of the device only. The DPS is set to "0" in this example as the device port size is 8 bits.

The memory map of this example is shown in Table 4. The operand consists of BYTE A through BYTE D in memory of Table 4. Prior to the transfer, MAR and DAR are set to 00000012 and 00000108 respectively. The operand is transferred to the 8-bit port device according to the order of transfer number in Table 3.

Table 2 Dual-Address Sequencing

Row No.	Port Size	Operand Size	Operand Part Size	Operand Part Addresses	Address Increment		
					+	=	−
1	8	BYTE	BYTE	A	+2	0	−2
2	8	WORD	BYTE	A, A+2	+4	0	−4
③	8	LONG	BYTE *4	A, A+2, A+4, A+6 *3 *5 *7 *8	+8	0	−8 *10
4	16	BYTE	PACK (BYTE or WORD)**	A	+P	0	−P
5	16	WORD	WORD	A	+2	0	−2
⑥	16	LONG	WORD *2	A, A+2 *1 *6	+4 *9	0	−4

*Numbers in Table 2 correspond to ones in Tables 3 and 4.
**Refer to Address Sequencing on Operand Part Size and PACK.

Table 3 An Example of a Data Transfer for One Operand

SRC: Source (Memory), DST Destination (Device), HR: Holding Register (DMAC Internal Reg.)

Transfer No.	Data Transfer	Address Output	Data Size on Bus	DMAC Registers after Transfer		Comment
				MAR	DAR	
0	—	—	—	00000012	00000108	Initial Register Setting
1	SRC→HR	00000012 *1	WORD *2	00000014	00000108	Higher order 16 bits of operand is fetched.
2	HR →DST	00000108 *3	BYTE *4	00000014	0000010A	Higher order 16 bits of operand is transferred.
3	HR →DST	0000010A *5	BYTE *4	00000014	0000010C *10	
4	SRC→HR	00000014 *6	WORD *2	00000016 *9	0000010C	Lower order 16 bits of operand is fetched.
5	HR →DST	0000010C *7	BYTE *4	00000016	0000010E	Lower order 16 bits of operand is transferred.
6	HR →DST	0000010E *8	BYTE *4	00000016	00000110 *10	
6′	—	—	—	00000016	00000110	MAR, DAR are pointing the next operand addresses when the transfer is complete.

Mode: Port size=8, operand size=Long Word, Memory to Device, Source (Memory) Counts Up, Destination (Device) Counts Down

Table 4 Memory Map for the Example of the Data Transfer

ADDRESS			ADDRESS
00000010			00000011
00000012	BYTE A *1	BYTE B *1	00000013
00000014	BYTE C *6	BYTE D *6	00000015
00000016			00000017

Source (Memory)

ADDRESS			
00000106			00000107
00000108	BYTE A *3		00000109
0000010A	BYTE B *5		0000010B
0000010C	BYTE C *7		0000010D
0000010E	BYTE D *8		0000010F
00000110			00000111

Destination (Device)

● Single Addressing Mode

Implicitly addressed devices are peripheral devices selected not by address but by $\overline{ACK}$. They do not require addressing of data register during data transfer. Transfers between memory and these devices are controlled by the request/acknowledge protocol. Such peripherals require only one bus cycle to transfer data, and the DMAC internal holding register is not used. Because only the memory is addressed during a data transfer and a transfer is done in only one bus cycle, this protocol is called single-addressing.

(1) Device with $\overline{ACK}$ Transfers

Under this protocol, the communication between peripheral device and the DMAC is performed with a two signal $\overline{REQ}/\overline{ACK}$ handshake. When a request is generated using the request method programmed in the DMAC's internal control registers, the DMAC obtains the bus and responds with $\overline{ACK}$. The DMAC asserts all the bus control signals required for the memory access. Refer to Figure 28 for the flowchart of the data transfer from memory to the device with $\overline{ACK}$. Figure 29 shows the flowchart of the data transfer from the device with $\overline{ACK}$ to memory. Receiving the transfer request, the DMAC obtains the bus. Then the DMAC outputs the memory address and asserts $\overline{ACK}$ to inform the I/O device that the transfer request has been acknowledged. When the DMAC accepts $\overline{DTACK}$ from memory, it asserts $\overline{DTC}$ and informs the peripheral device of the transfer termination.

Figures 30 and 31 show the transfer timings of the device with $\overline{ACK}$: the port size for the former figure is 8-bit and the latter is 16-bit respectively.

When the transfer is from memory to a device, data is valid when $\overline{DTACK}$ is asserted and remains valid until the data strobes are negated. The assertion of $\overline{DTC}$ from the DMAC may be used to latch the data as data strobes are not negated half clock period after the assertion of $\overline{DTC}$.

When the transfer is from device to memory, data must be valid on the HMCS68000 bus before the DMAC asserts the data strobes. The data strobes are asserted one clock period after $\overline{ACK}$ is asserted. When the DMAC obtains the bus and starts a DMA cycle, the three-state of the $\overline{OWN}$ line is cancelled a half clock earlier than other control lines. If the DMA Cycle terminates and the DMAC relinquishes the bus, all the control signals get three-stated a half clock before $\overline{OWN}$. The $\overline{DDIR}$ and

$\overline{DBEN}$ lines are not asserted in the single addressing mode. Four-clock cycle is the smallest bus cycle for the transfer from memory to device. Five-clock cycle is the smallest bus cycle for the transfer from device to memory. If the device port size is 8bits, either $\overline{LDS}$ or $\overline{UDS}$ is asserted. In the single addressing mode, $A_8/D_0 \sim A_{23}/D_{15}$ are outputted for only one and a half clock from the beginning of the DMA bus cycle. Therefore, A_8/D_0 through A_{23}/D_{15} need to be latched externally just like in the dual addressing mode.

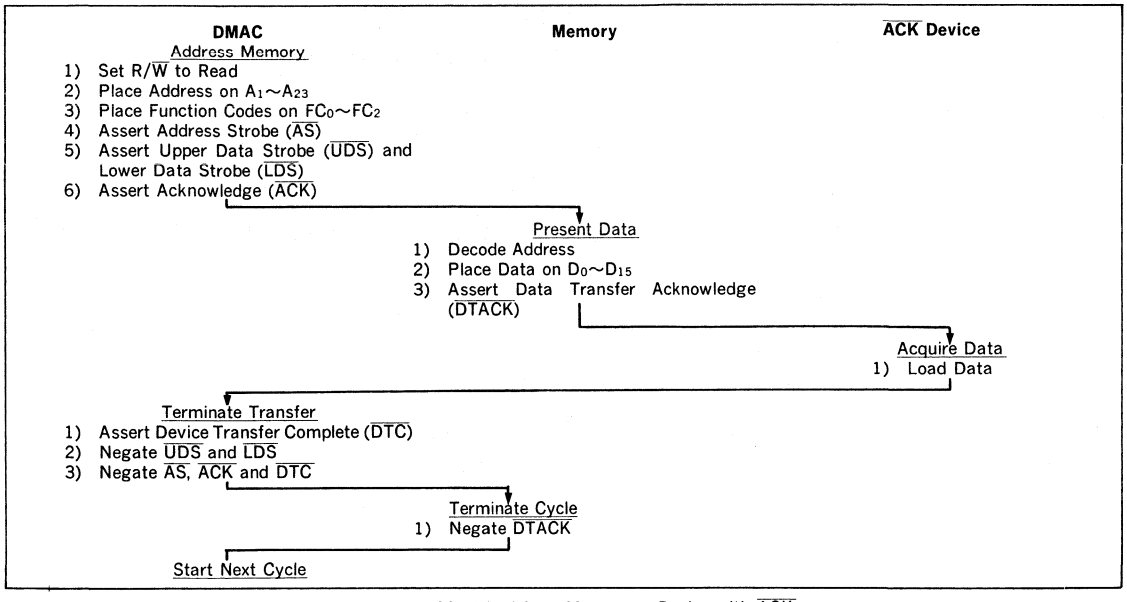


Figure 28 Word from Memory to Device with $\overline{ACK}$

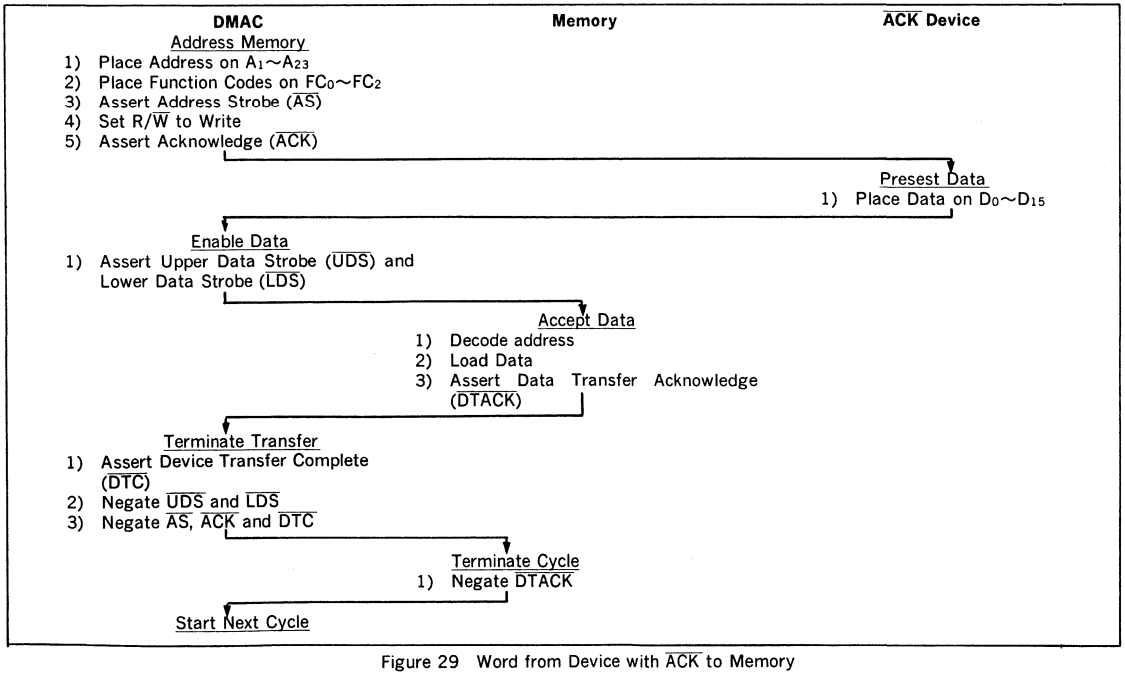


Figure 29 Word from Device with $\overline{ACK}$ to Memory

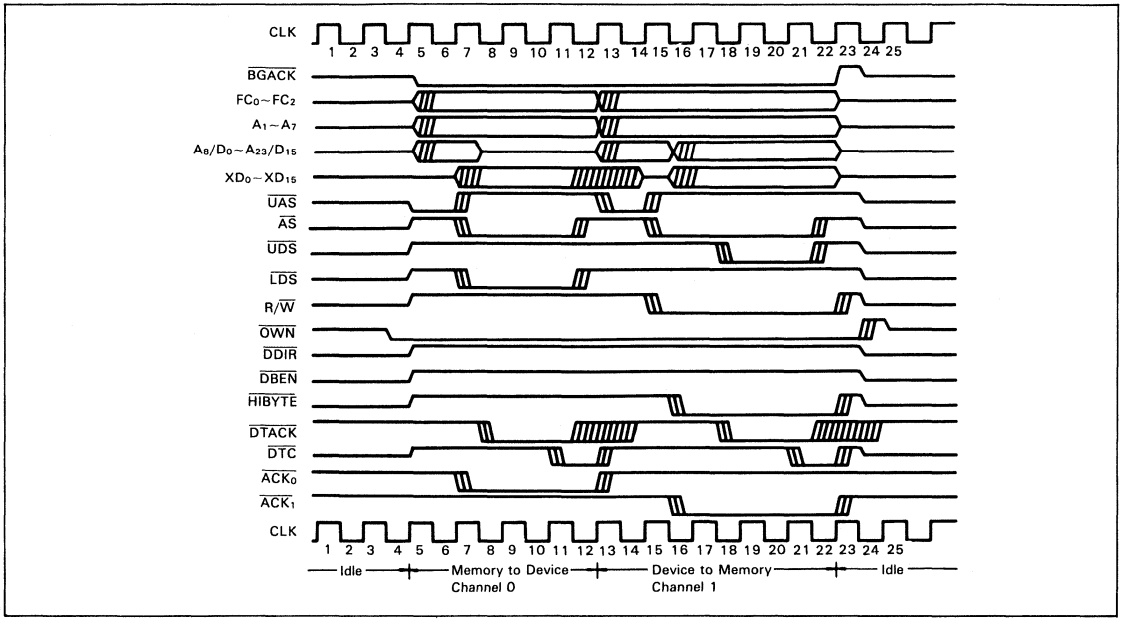


Figure 30 Single Addressing Mode with 8-Bit Devices as Source and Destination (Read-Write Cycles)

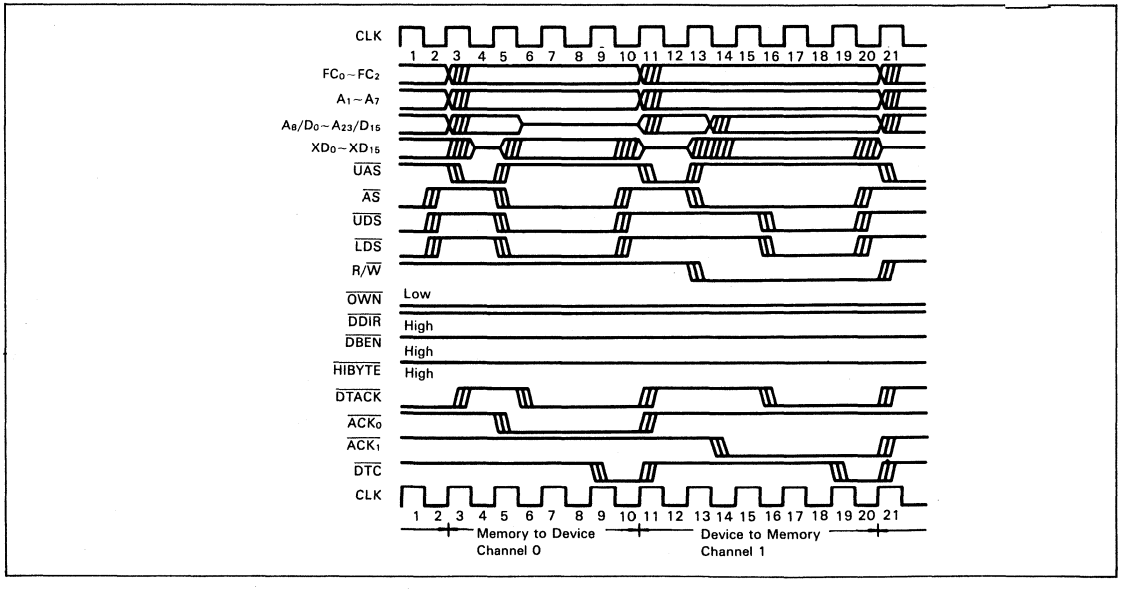


Figure 31 Single Addressing Mode with 16-bit Devices as Source and Destination (Read-Write Cycles)

(2) Device with $\overline{ACK}$ and $\overline{READY}$ Transfers

Under this protocol, the communication between peripheral device and the DMAC is performed using a three signal $\overline{REQ}$/$\overline{ACK}$/$\overline{READY}$ handshake. The $\overline{READY}$ input to the DMAC is provided by the $\overline{PCL}$ line. The $\overline{READY}$ line is active low. When a request is generated using the request method programmed in the control registers, the DMAC obtains the bus and asserts $\overline{ACK}$ to notify the device that the transfer is to take place. The DMAC waits for $\overline{READY}$ ($\overline{PCL}$ input), which is a response from the device, in addition to $\overline{DTACK}$ which is a response from memory.

When the DMAC accepts both signals, it terminates the transfer. Refer to Figures 34 and 35 for the flowcharts of the data transfer between memory and the device with $\overline{ACK}$ and $\overline{READY}$. Refer to Figure 36 for the transfer timing of the 8-bit device. When the data transfer is from memory to a device, data is valid from the assertion of $\overline{DTACK}$ to the negation of $\overline{LDS}$ and $\overline{UDS}$. $\overline{DTC}$ is asserted a half clock before $\overline{LDS}$ and $\overline{UDS}$ are negated, so this line may be used for latching the data by the peripheral device. In this case, $\overline{READY}$ ($\overline{PCL}$ input) indicates that the device has received the data. Both $\overline{DTACK}$ and $\overline{READY}$ ($\overline{PCL}$ input) signals are needed for terminating the DMA cycle.

When the data transfer is from the device to memory, data must be valid on the bus before the DMAC asserts $\overline{LDS}$ and $\overline{UDS}$. Therefore, $\overline{READY}$ ($\overline{PCL}$ input) is used as the signal to indicate that the peripheral device has outputted the data on the bus. When the DMAC detects PCL ($\overline{READY}$ input), then it asserts $\overline{LDS}$ and $\overline{UDS}$. After asserting $\overline{LDS}$ and $\overline{UDS}$, the DMAC terminates the cycle when $\overline{DTACK}$ signal from the memory is detected.

As mentioned above, the I/O device and the DMAC communicate each other through three handshake signals in Figure 32.

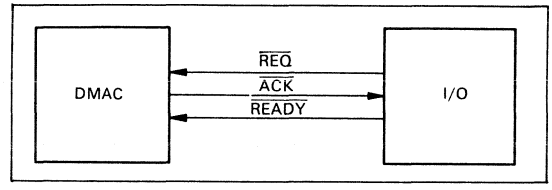


Figure 32 Device with $\overline{ACK}$ and $\overline{READY}$, and DMAC transfer protocol

Figure 33 shows the timing of transfers between the memory and the device with $\overline{ACK}$ and $\overline{READY}$. The detail is as Follows.

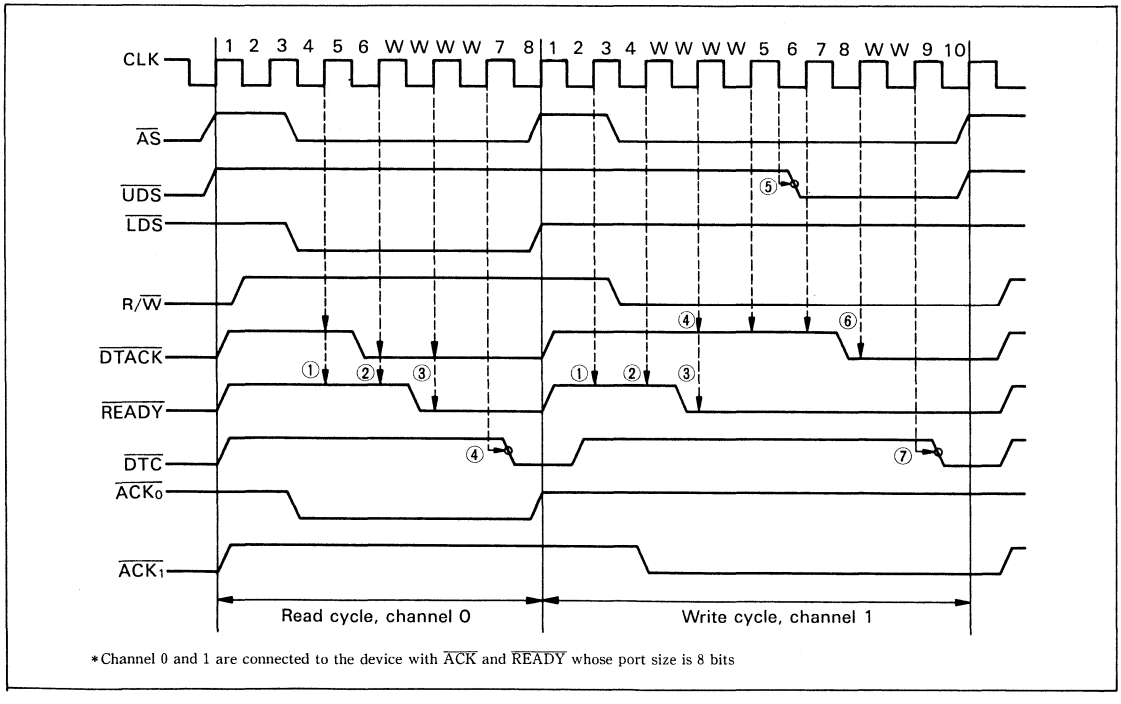


Figure 33 Device with $\overline{ACK}$ and $\overline{READY}$ Mode Timing

(i) Data transfer from the memory to the I/O device (read cycle)

The DMAC samples both $\overline{DTACK}$ and $\overline{READY}$ signals at the rising edge of CLK 5 (Figure 33 ①).

Until both signals are asserted, the DMAC repeats wait cycles and samples those signals at each rising edge of the clock : the DMAC does not proceed to CLK 7 and 8 (Figure 33 ②).

When both $\overline{DTACK}$ and $\overline{READY}$ signals are asserted (Figure 33 ③), the DMAC proceeds to CLK 7 and 8, asserts $\overline{DTC}$ at the rising edge of CLK 7 (Figure 33 ④), and terminates the bus cycle. The bus cycle is 4-clock long when there is no wait cycles.

(ii) Data transfer from the I/O device to the memory (write cycle)

The DMAC samples $\overline{READY}$ signal at the rising edge of CLK 3 (Figure 33 ①). Until $\overline{READY}$ signal is asserted, the DMAC repeats wait cycles and samples the signal at each rising edge of the clock : the DMAC does not proceed to CLK 5 and 6 (Figure 33 ②).

When $\overline{READY}$ signal is asserted (Figure 33 ③), the DMAC proceeds to CLK 5~8, and asserts $\overline{DS}$ at the falling edge of CLK 5 (Figure 33 ⑤).

Table 5 indicates the combinations of port size and operand size of the peripheral devices supported by the DMAC in the single and dual addressing modes. In the single addressing mode, port size and operand size must be the same. In the dual addressing mode, byte operand cannot be used when the port size is sixteen and the REQG bit is 10 or 11.

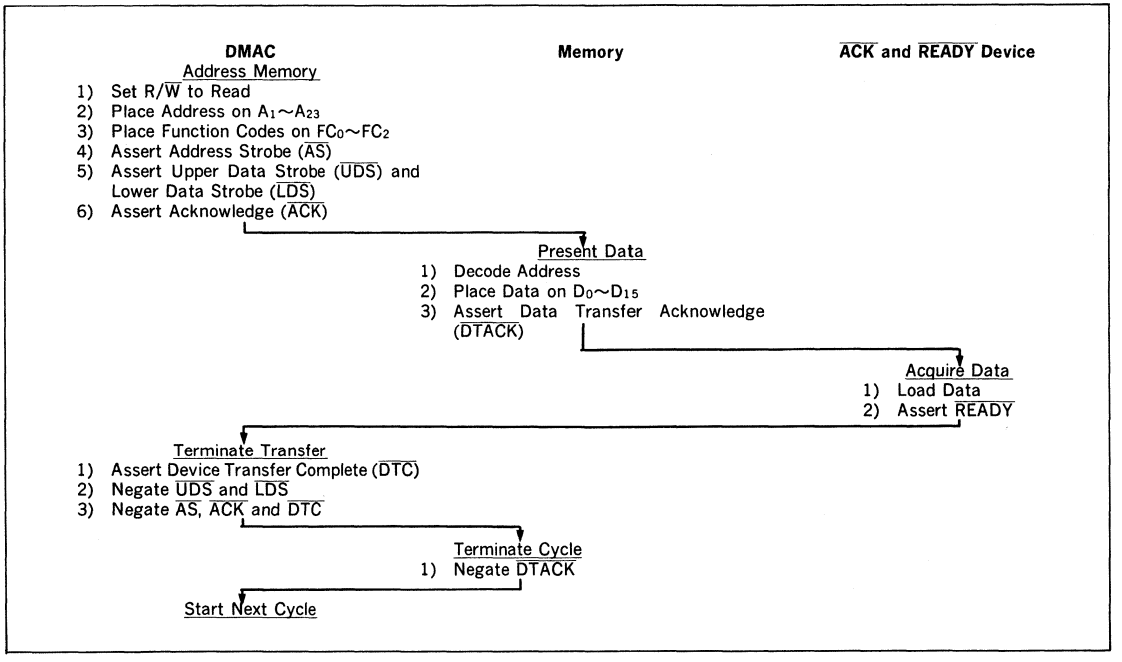


Figure 34 Word from Memory to Device with $\overline{ACK}$ and $\overline{READY}$

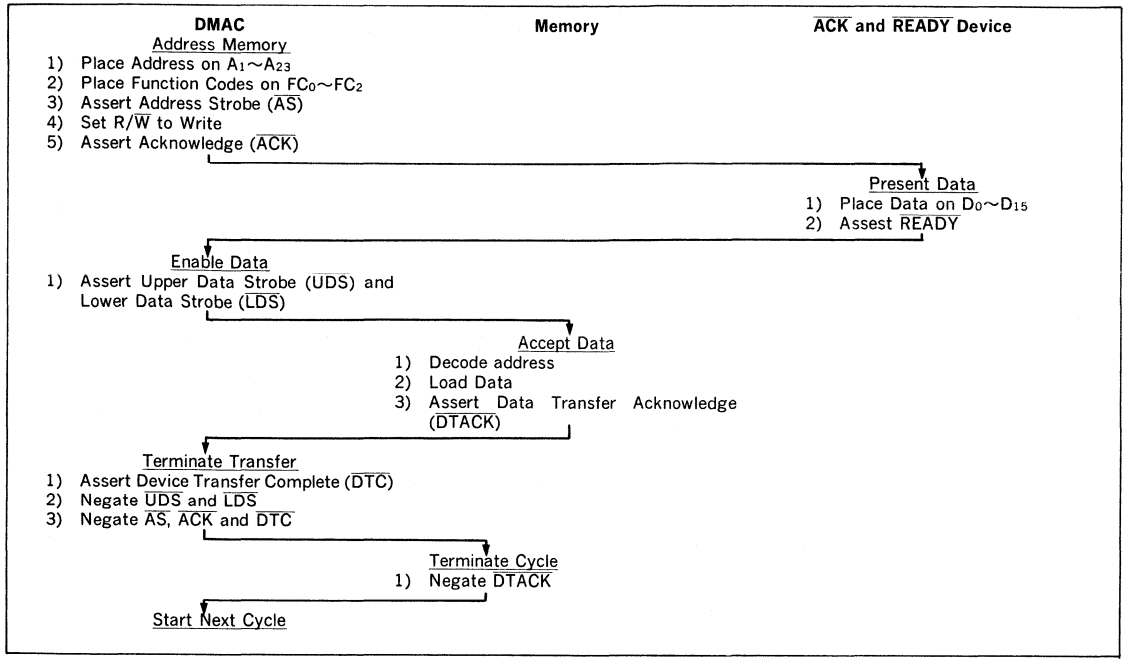


Figure 35 Word from Device with ACK and READY to Memory

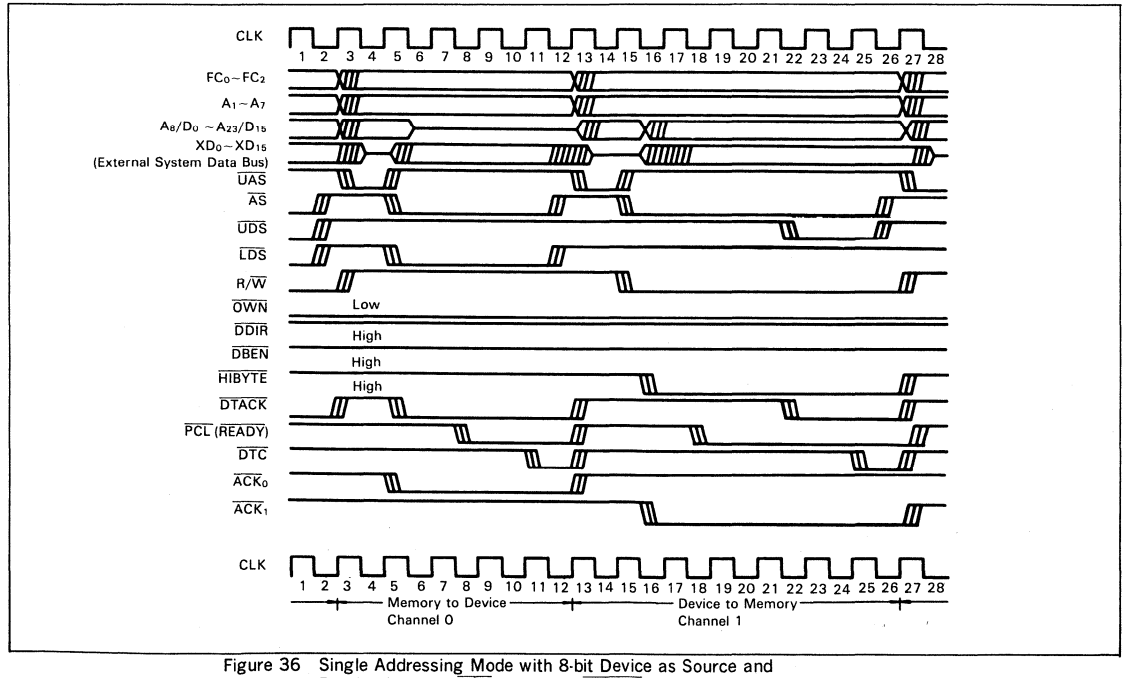


Figure 36 Single Addressing Mode with 8-bit Device as Source and Destination with PCL Used as a READY input (Read-Write Cycles)

Table 5 Operation Combinations

Addressing	Device Type	Port	Operand			REQG bits of OCR
			Byte	Word	Long Word	
Dual	68000, 6800	8	○	○	○	00, 01, 10, 11
Dual	68000, 6800	16	○	○	○	00, 01
Dual	68000, 6800	16	×	○	○	10, 11
Single	with $\overline{ACK}$ or $\overline{ACK}$ & $\overline{READY}$	8	○	×	×	00, 01, 10, 11
		16	×	○	×	00, 01, 10, 11

○ : supported × : not supported

REQG
- 00: Auto request at a limited rate
- 01: Auto request at a maximum rate
- 10: $\overline{REQ}$ line requests the operand transfer
- 11: First operand is auto requested, and subsequent operands are externally requested.

● Address Sequencing

The sequence of addresses generated depends upon the port size, whether the addresses are to count up, down or not change and whether the transfer is executed in the single addressing mode or the dual addressing mode. The memory address count method and the peripheral device address count method is programmed using the Memory address count (MAC) bit and the Device address count (DAC) bit in the Sequence Control Register (SCR).

(i) Single addressing mode
In the single addressing mode, memory address sequencing is shown in Table 6. If the operand size is byte, the memory address increment is one (1). If the operand size is word, the memory address increment is two (2). If the memory address register does not count, the memory address is unchanged after the transfer.

(ii) Dual addressing mode
In the dual addressing mode, the operand size need not match the port size. Thus the transfer of an operand may require several DMA bus cycles. Each DMA bus cycle, between memory and DMAC and between DMAC and the device, is called the operand part and transfers a portion or all of the operand. The addresses of the operand parts are in a linear increasing sequence. The step between the addresses of the operand is two. The size of the operand part is the minimum of the port size and the operand size. The number of the operand part is the operand size divided by the port size.

In the dual addressing mode, memory is regarded as a device whose port size is 16 bits and the operand size is a byte or a word. When the operand is transferred to the memory from the I/O device whose port size is 8 bits and the operand size is byte, the DMAC reads 2-byte operand one byte at a time from the I/O device and writes 2 bytes at the same time to the memory, or reads one byte from the I/O device and writes one byte to the memory. Thus, when the port size is 8 bits and the operand size is byte, two-operand transfer which is performed at the same time is called PACK. Utilizing the PACK, the DMAC may improve the DMA bus efficiency. However, packing is not performed if the address does not count. When the port size is 8 bits and the operand size is byte (port size : 8 bits, without PACK) with the DMAC in the dual addressing mode, the DMAC repeats the following cycles :

① READ BYTE (reads data from the I/O device or the memory)
② WRITE BYTE (writes data to the I/O device or the memory)

Table 7 shows the dual addressing sequencing

Table 6 Single Address Sequencing

Port Size	Operand Size	Memory Address Increment		
		+(increment)	=(unchanged)	−(decrement)
8	Byte	+1	0	−1
16	Word	+2	0	−2

Table 7 Dual Address Sequencing

Port Size	Operand Size	Part Size	Operand Part Address	Address Increment		
				+	=	−
8	Byte	Byte	A	+2	0	−2
8	Word	Byte	A, A+2	+4	0	−4
8	Long	Byte	A, A+2, A+4, A+6	+8	0	−8
16	Byte	Pack	A	+P	0	−P
16	Word	Word	A	+2	0	−2
16	Long	Word	A, A+2	+4	0	−4

P =1 if packing is not done
=2 if packing is done

Pack =byte if packing is not done
=word if packing is done

■INITIATION AND CONTROL OF CHANNEL OPERATION

● Operation Initiation

To initiate the operation of a channel, the STR bit of the CCR is set to start the operation. Setting the STR bit causes the immediate activation of the channel, the channel will be ready to accept requests immediately. The channel initiates the operation by resetting the STR bit and setting the channel active bit in the CSR. Any pending requests are cleared, and the channel is then ready to receive requests for the new operation. If the channel is configured for an illegal operation, the configuration error is signaled, and no channel operation is run. The illegal operations include the selection of any of the options marked "(undefined, reserved)". If the MTC is set to zero in any mode other than the chaining mode, or BTC is set to zero in the array chaining mode, then the count error is signaled and the channel is not activated. The channel cannot be started if any of the ACT, COC, BTC, NDT or ERR bit is set in the CSR. In this case, the channel signals the operation timing error.

● Operation Continuation (Continue Mode)

When the STR bit or the ACT bit in the CSR is set, setting the CNT (Continue) bit in the CCR allows multiple blocks to be transferred as in the chaining modes. The CNT bit is set in order to continue the current channel operation. To set the CNT bit, the initial address of the next block to be transferred, the corresponding function code, and the number of words to be transferred must be previously set to the BAR, BFC and BTC. If the CNT bit is set when either the STR or the ACT bit is not set, the operation timing error is signaled. The configuration error is signaled when the CNT bit is set in the chaining modes.

● Operation Halting (Halt)

The CCR has a halt bit which allows suspension of the operation of the channel. If this bit is set, a request may still be generated and recognized, but the DMAC does not attempt to acquire the bus or to make transfers for the halted channel. When this bit is reset, the channel resumes operation and services any request that may have been received while the channel was halted. However, in the burst request mode, the transfer request should be kept asserted until the initiation of the first transfer after clearing the halt bit.

● Operation Abort by Software (Software Abort)

Setting the software abort bit (SAB) in the CCR allows the current operation of the channel to be aborted. In this case, the ERR bit and the COC bit in the CSR are set and the ACT bit is reset. The error code for the software abort is set in the CER. The SAB bit is designed to be reset if the ERR bit is reset. When the CCR is read, the SAB always reads as zero(0).

■CHANNEL OPERATION TERMINATION

As part of the transfer of an operand, the DMAC decrements the memory transfer counter(MTC). If the chaining mode is not used and the CNT bit is not set or the last block is transferred in the chaining mode, the operation of the channel is complete when the last operand transfer is completed and the MTC is zero. The DMAC notifies the peripheral device of the channel completion via the $\overline{\text{DONE}}$ output.

However, in the continue mode, $\overline{\text{DONE}}$ is outputted at the termination of every data block transfer. When the channel operation has been completed, the ACT bit of the CSR is cleared, and the COC bit of the CSR is set.

The occurrence of errors, such as the bus error, during the DMA bus cycle also terminates the channel operation. In this case, the ACT bit in the CSR is cleared, the ERR and the COC bits are set, and at the same time the code corresponding to the error that occurred is set in the CER.

● Channel Status Register (CSR)

The channel status register contains the status of the channel at the channel operation termination. The register is cleared by writing a one (1) into each bit of the register to be cleared.

COC

The channel operation complete (COC) bit is set if the channel operation has completed. The COC bit must be cleared in order to start another channel operation. The COC bit is cleared only by writing a one to this bit or resetting the DMAC.

PCS

The peripheral status (PCS) bit reflects the level of the $\overline{\text{PCL}}$ line regardless of its programmed function. If $\overline{\text{PCL}}$ is at "High" level, the PCS bit reads as one. If $\overline{\text{PCL}}$ is at "Low" level, the PCS bit reads as zero. The PCS bit is unaffected by writing to the CSR.

PCT

The peripheral control transition (PCT) bit is set, if a falling edge transition has occurred on the $\overline{\text{PCL}}$ line. (The $\overline{\text{PCL}}$ line must remain at "low" level for at least two clock cycles.) The PCT bit is cleared by writing a one to this bit or resetting the DMAC.

BTC

Block transfer complete (BTC) bit is set when the continue (CNT) bit of CCR is set and the memory transfer counter (MTC) is exhausted. The BTC bit must be cleared before the another continuation is attempted (namely, setting the CNT bit again), otherwise an operation timing error occurs. The BTC bit is cleared by writing a one to this bit or resetting the DMAC.

NDT

Normal device termination (NDT) bit is set when the peripheral device terminates the channel operation by asserting the $\overline{\text{DONE}}$ line while the peripheral device was being acknowledged. The NDT bit is cleared by writing a one to this bit or resetting the DMAC.

ERR

Error (ERR) bit is set if any errors have been signaled, When the ERR bit is set, the code corresponding to the kind of the error that occured is set in the CER. The ERR bit is clered by writing a one to this bit or resetting the DMAC.

ACT

The active (ACT) bit is asserted after the STR bit has been set and the channel operation has started. This bit remains set until the channel operation is terminated. The ACT bit is unaffected by write operations. This bit is cleared by the termination of the channel or resetting the DMAC.

DIT

Done input transition (DIT) bit is set if the $\overline{\text{DONE}}$ input is generated while the multiple block transfer mode with $\overline{\text{DONE}}$ is being set. The DIT bit is cleared by writing a one to this bit or resetting the DMAC.

● Interrupts

The DMAC can signal the termination of the channel operation by generating an interrupt request. The interrupt request is generated by the following condition.

① INT=1
and
② COC=1 or BTC=1 or ERR=1 or NDT=1 or PCT=1
(the $\overline{PCL}$ line is an interrupt input)
This may be represented as
$\overline{IRQ}=INT \cdot \overline{(COC+BTC+ERR+NDT+PCT^*)}$
(*PCL line is programmed as an interrupt input.)

When the $\overline{IRQ}$ line is asserted, changing the INT bit from one to zero to one will cause the $\overline{IRQ}$ output to change from "low" to "high" to "low" again. The $\overline{IRQ}$ should be negated by clearing the COC, the BTC, the ERR, the NDT and the PCT bits.

If the DMAC receives $\overline{IACK}$ from the MPU during asserting the $\overline{IRQ}$, the DMAC provides an interrupt vector. If multiple channels have interrupt requests, the determination of which channel presents its interrupt vector is made using the same priority scheme defined for the channel operations.

The bus cycle in which the DMAC provides the interrupt vector when receiving an $\overline{IACK}$ from the MPU is called the interrupt acknowledge cycle. The interrupt vector returned to the MPU comes from either the nomal or the error interrupt vector register. The normal interrupt register is used unless the ERR bit of CSR is set, in which case the error interrupt vector register is used. The content of the interrupt vector register is placed on $A_8/D_0 \sim A_{15}/D_7$, and $\overline{DTACK}$ is asserted to indicate that the vector is on the data bus. If a reset occurs, all interrupt vector registers are set to $(00001111)_2$, the value of the uninitialized interrupt vector of the HD68000 MPU. The timing of the interrupt acknowledge cycle is shown in Figure 36. The HD68000 MPU outputs the interrupt level into A_1-A_3 and "1" into A_4-A_7 during the interrupt acknowledge cycle, but the HD63450 DMAC ignores these signals.

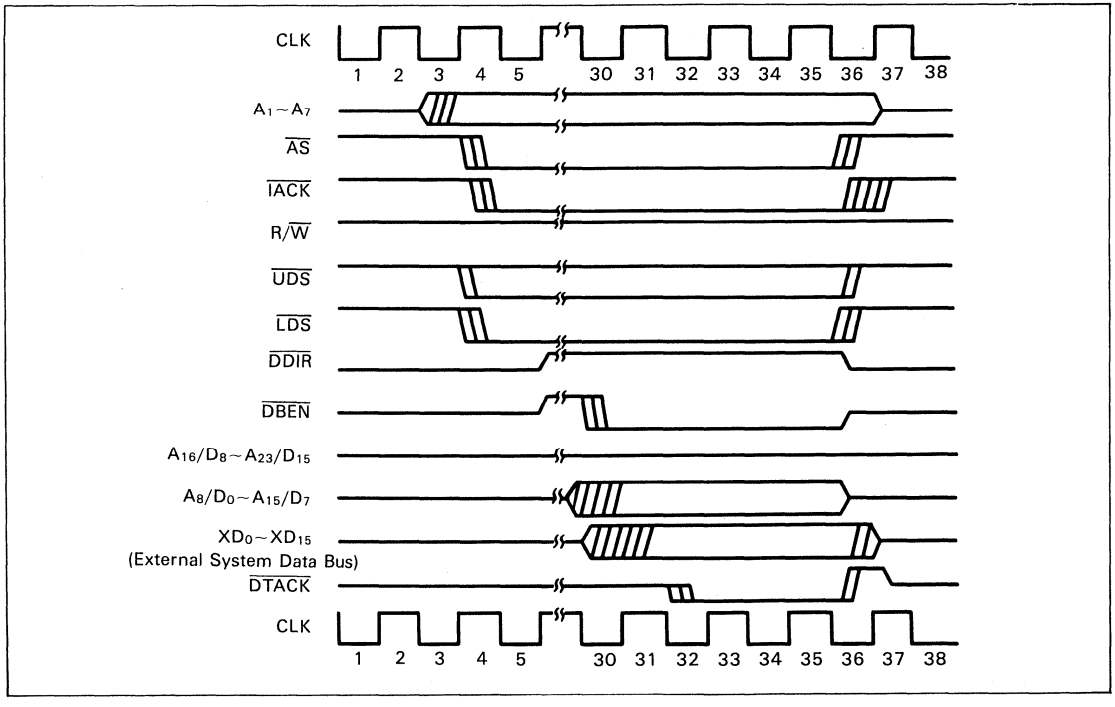


Figure 37 MPU $\overline{IACK}$ Cycle to DMAC

● Multiple Data Block Transfer Operation

When the memory transfer counter (MTC) is exhausted, the channel operation still continues if the channel is set to the array chaining mode or the linked array chaining mode, and the chain is not exhausted. The channel operation also continues if the continue bit (CNT) of the CCR is set. The DMAC provides the initialization of the memory address register and the memory transfer counter in these cases so that the DMAC can transfer the multiple blocks.

Continued Operation

The continued operation is described in the Initiation and the Control of the Channel Operation section.

Array Chaining

This type of chaining uses an array in memory consisting of memory addresses and transfer counts. Each entry in the array is six bytes long and, consists of four bytes of address followed by two bytes of transfer count. The beginning address of this array is in the base address register, and the number of entries in the array is in the base transfer counter. Before starting any block transfers, the DMAC fetches the entry currently pointed to by the base address register. The address information is placed in the memory address register, and the count information is placed in the memory transfer counter. As each chaining entry is fetched, the base transfer counter is decremented by one. After the chaining entry is fetched, the base address register is incremented

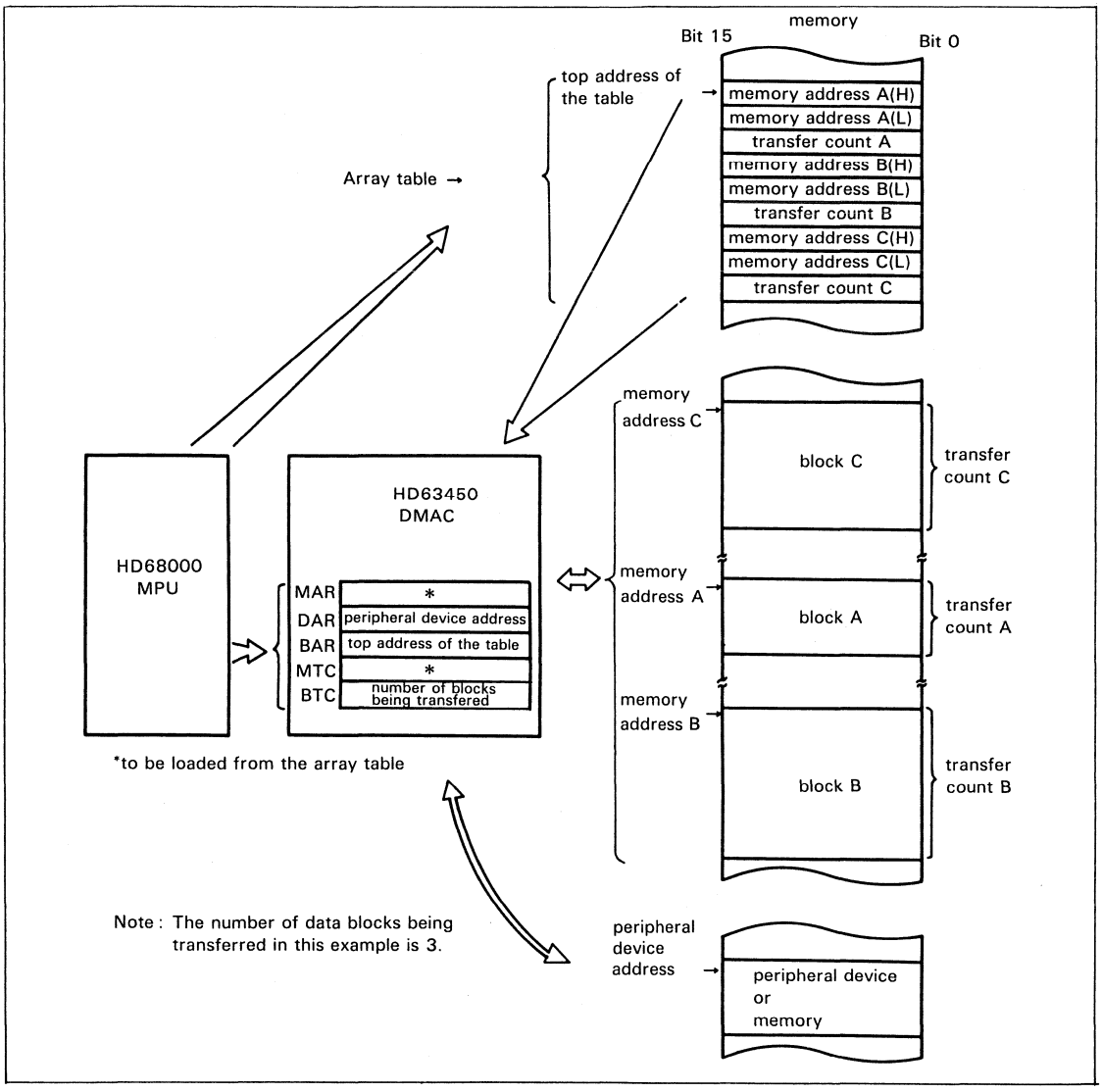


Figure 38 Transfer Example of the Array Chaining Mode

to point the next entry. When the base transfer counter reaches a terminal count of zero, the chain is exhausted, and the entry just fetched determines the last block of the channel operation.

An example of the array chaining mode operation and the memory format for supporting for array chaining is shown in Figure 38. The array must start at an even address, or the entry fetch results in an address error. If a terminal count is loaded into the memory transfer counter or the base transfer counter, the count error is signaled.

Linked Array Chaining

This type of chaining uses a list in memory consisting of memory address, transfer counts, and link addresses. Each entry in the chain list is ten bytes long, and consists of four bytes of memory address, two bytes of transfer count and four bytes of link address. The address of the first entry in the list is in the base address register, and the base transfer counter is unused. Before starting any block transfers, the DMAC fetches the entry currently pointed to by the base address register. The address information is placed in the memory address register, the count information is placed in the memory transfer counter, and the link address replaces the current contents of the base address

register. The channel then begins a new block transfer. As each chaining entry is fetched, the update base address register is examined for the terminal link which has all 32 bits equal to zero. When the new base address is the terminal address, the chain is exhausted, and the entry just fetched determines the last block of the channel operation.

An example of the linked array chaining mode operation and the memory format for supporting it is shown is Figure 39.

In Figure 39, the DMAC transfers data blocks in the order of Block A, Block B, and Block C. In the linked array chaining mode, the BTC is not used. When the DMAC refers to the linked array table, the value of the BFC is outputted as the function code. The values of the function code registers are unchanged by the linked array chaining operation.

This type of chaining allows entries to be easily removed or inserted without having to reorganize data within the chain. Since the end of the chain is indicated by a terminal link, the number of entries in the array need not be specified to the DMAC.

The linked array table must start at an even address in the linked array chaining mode. Starting the table at an odd address results in an address error. If "0" is initially loaded to the MTC, the count error is signaled. Because the MPU can read all of the DMAC registers, all necessary error recovery information is available to the operating system.
The comparison of both chaining modes is shown in Table 8.

Table 8 Chaining Mode Address/Count Information

Chaining Mode	Base Address Register	Base Transfer Counter	Completed When
Array Chaining	address of the array table	number of data blocks being transferred	Base Transfer Count=0
Linked Array Chaining	address of the linked array table	(not used)	Linked Address=0

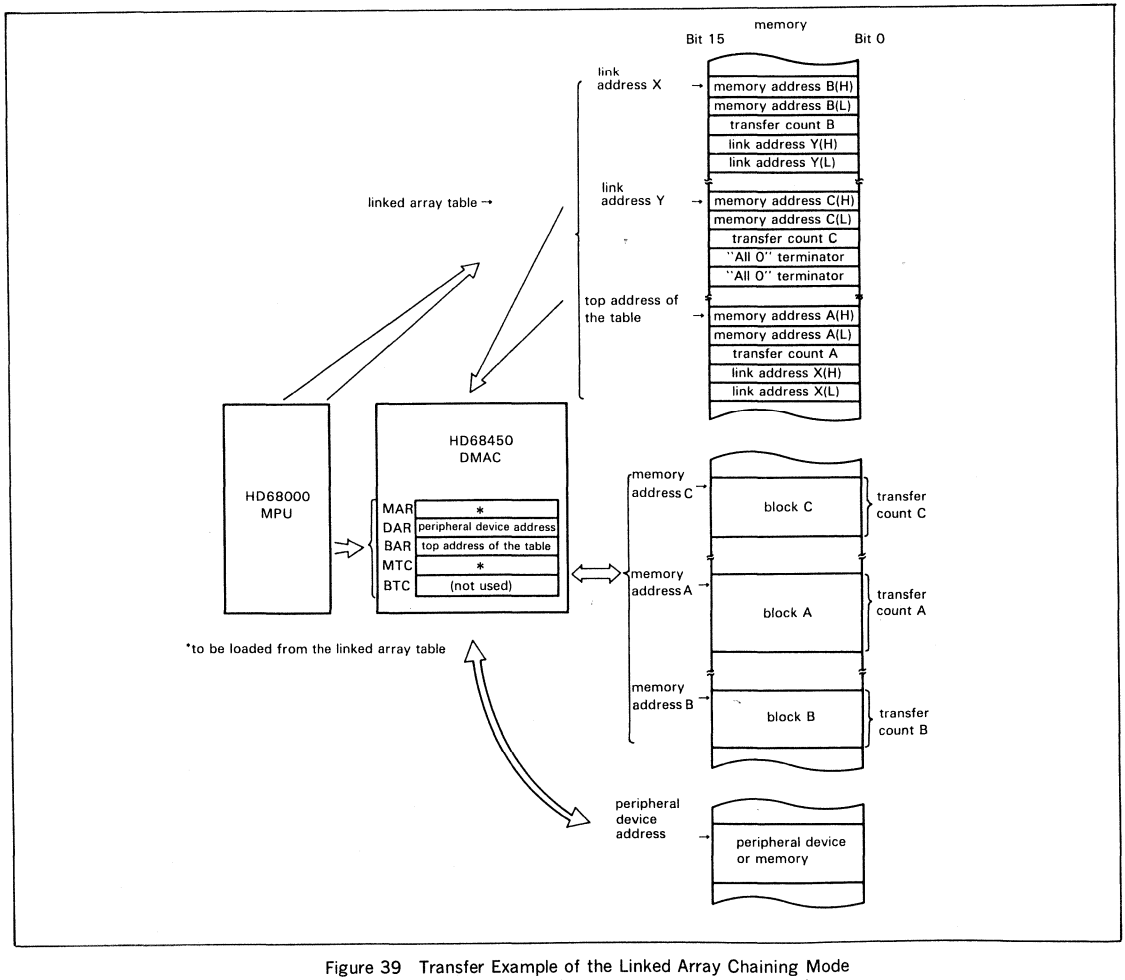


Figure 39 Transfer Example of the Linked Array Chaining Mode

Multi-Block Transfer with $\overline{\text{DONE}}$ Mode

The multi-block transfer with $\overline{\text{DONE}}$ mode is set by setting BTD bit of the OCR. In this mode, data block transfer continues even if the $\overline{\text{DONE}}$ signal is inputted during the DMA bus cycle. If $\overline{\text{DONE}}$ is inputted during the DMA bus cycle when the multi-block transfer is not performed, the DMAC resets ACT bit of the CSR, sets NDT and COC bits, and terminates the DMA operation.

When $\overline{\text{DONE}}$ is inputted from the I/O device during the DMA bus cycle in which $\overline{\text{ACK}}$ is outputted, the DMAC terminates the operand transfer and then the current block transfer. Then, maintaining the bus, the DMAC sets DIT bit of the CSR and reads the data block transfer information from the memory. After that, the DMAC transfers the next block as required.

In the continue mode, if $\overline{\text{DONE}}$ is inputted from the I/O device during the DMA bus cycle in which $\overline{\text{ACK}}$ is outputted, the DMAC terminates the operand transfer and terminates the current block transfer. Then the DMAC shifts the data in BAR, BFC and BTC to MAR, MFC and MTC, waits for the transfer request, and transfers the next block. If the value of BAR, BFC and BTC is the same as that of MAR, MFC and MTC, the DMAC repeats transferring the same block.

As stated above, the multi-block transfer with $\overline{\text{DONE}}$ mode realizes termination (stops the current block transfer) and restart (starts transferring the next block) of the multi-block transfer in the high-speed data transfer system without MPU interposition.

● Bus Exception Conditions

The DMAC has three lines for inputting bus exception conditions called $\overline{\text{BEC}_0}$, $\overline{\text{BEC}_1}$, and $\overline{\text{BEC}_2}$. The priority encoder can be used to generate these signals externally. These lines are encoded as shown in Table 9.

Table 9 BEC Bus Exception Condition

$\overline{\text{BEC}_2}$	$\overline{\text{BEC}_1}$	$\overline{\text{BEC}_0}$	Exception Condition
1	1	1	No exception condition
1	1	0	Halt
1	0	1	Bus error
1	0	0	Retry
0	1	1	Relinquish bus and retry
0	1	0	(undefined, reserved)
0	0	1	(undefined, reserved)
0	0	0	Reset

In order to guarantee reliable decoding, the DMAC verifies that the incoming code has been stable for two DMAC clock cycles before acting on it. The DMAC picks up $\overline{\text{BEC}_0}$-$\overline{\text{BEC}_2}$ at the rising edge of the clock. If $\overline{\text{BEC}_0}$-$\overline{\text{BEC}_2}$ is asserted to the undefined code, the operation of the DMAC does not proceed. For example, when the DMAC is waiting for $\overline{\text{DTACK}}$, inputting $\overline{\text{DTACK}}$ does not result in the termination of the cycle if $\overline{\text{BEC}_0}$-$\overline{\text{BEC}_2}$ is asserted to the undefined code. In addition, when the transfer request is received, $\overline{\text{BR}}$ is not output if the $\overline{\text{BEC}_0}$-$\overline{\text{BEC}_2}$ is not set to code (111).

If exception condition, except for HALT, is inputted during the DMA bus cycle prior to, or in coincidence with $\overline{\text{DTACK}}$, the DMAC terminates the current channel operation immediately. Here coincident means meeting the same set up requirements for the same sampling edge of the clock. $\overline{\text{BEC}_0}$~$\overline{\text{BEC}_2}$ is ignored in the current DMA bus cycle if it is input after $\overline{\text{DTACK}}$. If a bus exception condition exists, the DMAC does not generate any bus cycles until it is removed. However, the DMAC still recognizes requests.

Halt

The timing diagram of halt is shown in Figure 40. This diagram shows halt being generated during a read cycle from the 68000 compatible device in the dual addressing mode. If the halt exception is asserted during a DMA bus cycle, the DMAC does not terminate the bus cycle immediately. The DMAC waits for the assertion of $\overline{\text{DTACK}}$ before terminating the bus cycle so that the bus cycle is completed normally. In the halted state, the DMAC puts all the control signals to high impedance and relinquishes the bus to the MPU. The DMAC does not output the $\overline{\text{BR}}$ until halt exception is negated. When halt exception is negated, the DMAC acquires the bus again and proceeds the DMA operation. In order to insure a halt exception operation, the $\overline{\text{BEC}}$ lines must be set to halt at least until the assertion of $\overline{\text{DTC}}$.

If the halt is asserted when the DMAC has the bus but is not executing any bus cycle, the DMAC relinquishes the bus as soon as halt exception is asserted.

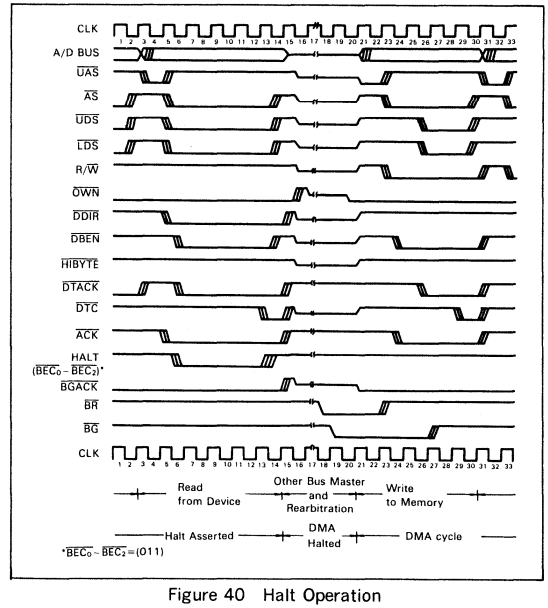


Figure 40 Halt Operation

Bus Error

The bus error exception is generated by external circuitry to indicate the current transfer cannot be successfully completed and is to be aborted. As soon as the DMAC recognizes the bus error exception, the DMAC immediately terminates the bus cycle and proceeds to the error recovery cycle. In this cycle, the DMAC adjusts the values of the MAR, the DAR, the MTC and the BTC to the values when the bus error exception occurred. 24 clocks are required for the error recovery cycle in the single addressing mode and in the read cycle of the dual addressing mode. 28 clocks are required in the write cycle of the dual addressing mode. If the DMAC does not have any transfer request in the other channels after the error recovery cycle, the DMAC relinquishes the bus.

The diagram of the bus error timing is shown in Figure 41.

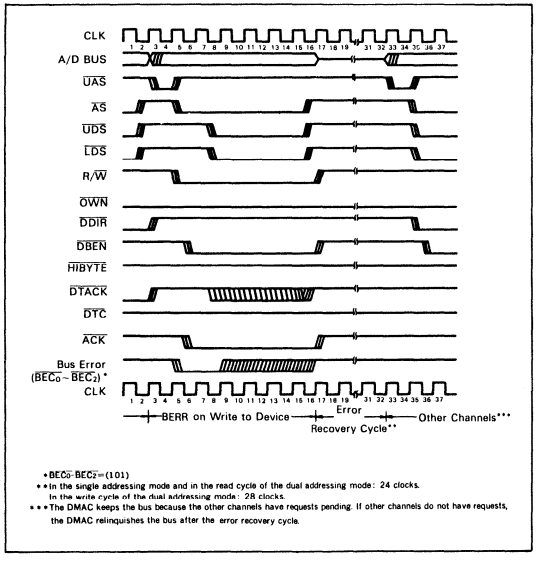


Figure 41 Bus Error Operation

Retry

The retry exception causes the DMAC to terminate the present operation and retry that operation when retry is removed, and thus will not honor any requests until it is removed. However, the DMAC still recognizes requests. The retry timing is shown in Figure 42.

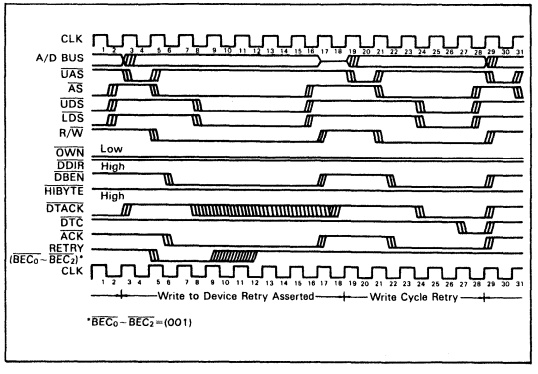


Figure 42 Retry Operation

Relinquish and Retry (R & R)

The relinquish and retry exception causes the DMAC to relinquish the bus and three-state all bus master controls and when the exception is removed, rearbitrate for the bus to retry the previous operation.

The diagram of the relinquish and retry timing is shown in Figure 43.

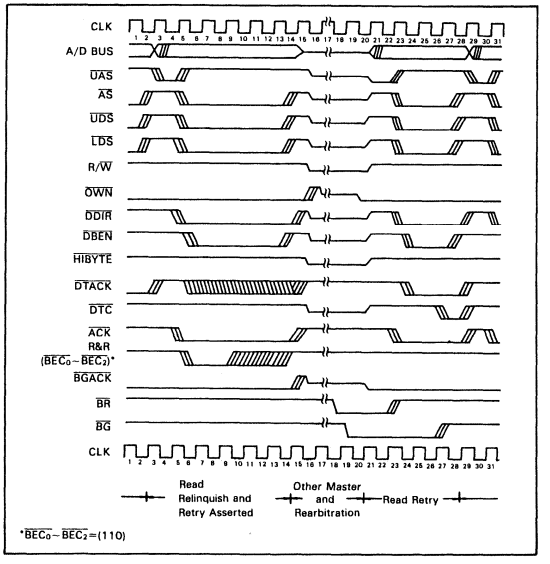


Figure 43 Relinquish and Retry Operation

Reset

The reset provides a means of resetting and initializing the DMAC. If the DMAC is bus master when the reset is asserted, the DMAC relinquishes the bus. Reset clears GCR, DCR, OCR, SCR, CCR, CSR, CPR, and CER for all channels. The NIV and the EIV are all set to $(OF)_{16}$, which is the uninitialized interrupt vector number for the HD68000 MPU. MTC, MAR, DAR, BTC, BAR, MFC, DFC, and BFC are not affected (see Table 10).

In order to insure a reset, $\overline{BEC_0}$~$\overline{BEC_2}$ must be kept at "Low" level for at least ten clocks.

Table 10 The Value after Resetting

Register	Value	Comments
MAR	×××××××	
DAR	×××××××	
BAR	×××××××	
MFC	×	
DFC	×	
BFC	×	
MTC	××××	
BTC	××××	
NIV	0 F	uninitialized vector
EIV	0 F	uninitialized vector
CPR	0 0	
DCR	0 0	
OCR	0 0	
SCR	0 0	
CCR	0 0	
CSR	0 0 or 0 1	depending on $\overline{PCL}$
CER	0 0	
GCR	0 0	

×—indefinite value, or the value before resetting

● Error Conditions

When an error is signaled on a channel, all activity on that channel is stopped. The ACT bit of the CSR is cleared, and the COC bit is set. The ERR bit of the CSR is set, and the error code is indicated in the CER. All pending operations are cleared, so that both the STR and CNT bits of CCR are cleared.

Enumerated below are the error signals and their sources.

(a) Configuration Error—This error occurs if the STR bit is set in the following cases.
  (i) The CNT bit is set at the same time STR bit in the chaining mode.
  (ii) DTYP specifies a single addressing mode, and the device port size is not the same as the operand size.
  (iii) DTYP specifies a dual addressing mode, DPS is 16 bits, SIZE is 8 bits and REQG is "10" or "11".
  (iv) An undefined configuration is set in the registers. The undefined configuraions are: XRM=01, MAC=11, DAC=11, CHAIN=01, and SIZE=11.
  When the port size is 8 bits, SIZE=11 is not an error in the dual addressing mode.

(b) Operation Timing Error—An operation timing error occurs in the following cases:
  (i) When the CNT bit is set after the ACT bit has been set by the DMAC in the chaining mode, or when the STR and the ACT bits are not set.
  (ii) The STR bit is set when ACT, COC, BTC, NDT or ERR is set.
  (iii) An attempt to write to the DCR, OCR, SCR, MAR, DAR, MTC, MFC, or DFC is made when the STR bit or the ACT bit is set.
  (iv) An attempt to set the CNT bit is made when the BTC and the ACT bits are set.

(c) Address Error—An address error occurs in the following cases:
  (i) An odd address is set for word or long word operands.
  (ii) $\overline{CS}$ or $\overline{IACK}$ is asserted during the DMA bus cycle.

(d) Bus Error—Bus error occurs when a bus error exception is signaled during a DMA bus cycle.

(e) Count Error—A count error occurs in the following cases:
  (i) The STR bit is set when zero is set in the MTC and the chaining mode is not used.
  (ii) The STR bit is set when zero is set in BTC for the array chaining mode.
  (iii) Zero is loaded from memory or the BTC to the MTC in the chaining modes or the continue mode.

(f) External Abort—External abort occurs if an abort is asserted by the external circuitry when the $\overline{PCL}$ line is configured as an abort input and the STR or the ACT bit is set.

(g) Software Abort—Software abort occurs if the SAB bit is set when the STR or the ACT bit is set.

Error Recovery Procedures

If an error occurs during a DMA transfer, appropriate information is available to the operating system (OS) to allow a software failure recovery operation. The operating system must be able to determine how much data was transferred, where the data was transferred to, and what type of error occurred.

The information available to the operating system consists of the present value of the Memory Address, Device Address and Base Address Register, the Memory Transfer and Base Transfer Counters, the channel status register, the channel error register. After the successful completion of any transfer, the memory and device address registers point to the location of the next operand to be transferred and the memory transfer counter contains the number of operands yet to be transferred. If an error occurs during a transfer, that transfer has not completed and the registers contain the values they had before the transfer was attempted. If the channel operation uses chaining, the Base Address Register points to the next chain entry to be serviced, unless the termination occurred while attempting to fetch an entry in the chain. In that case, the Base Address Register points to the entry being fetched. However, in the case of external abort, there are cases in which the previous values are not recovered.

Bus Exception Operating Flow

The bus exception operating flow in the case of multiple exception conditions occurring continuously in sequence is shown in Figure 44. Note that the DMAC can receive and execute the next exception condition before completing the current exception operation. For example, if the retry exception occurs, and next the relinquish and retry exception occurs while the DMAC is waiting for the retry condition to be cleared, the DMAC relinquishes the bus and waits for the exception condition to be cleared. If a bus error occurs during this period, the DMAC executes the bus error exception operation.

The flow diagram of the normal operation without exception operation or errors is shown in Figure 45.

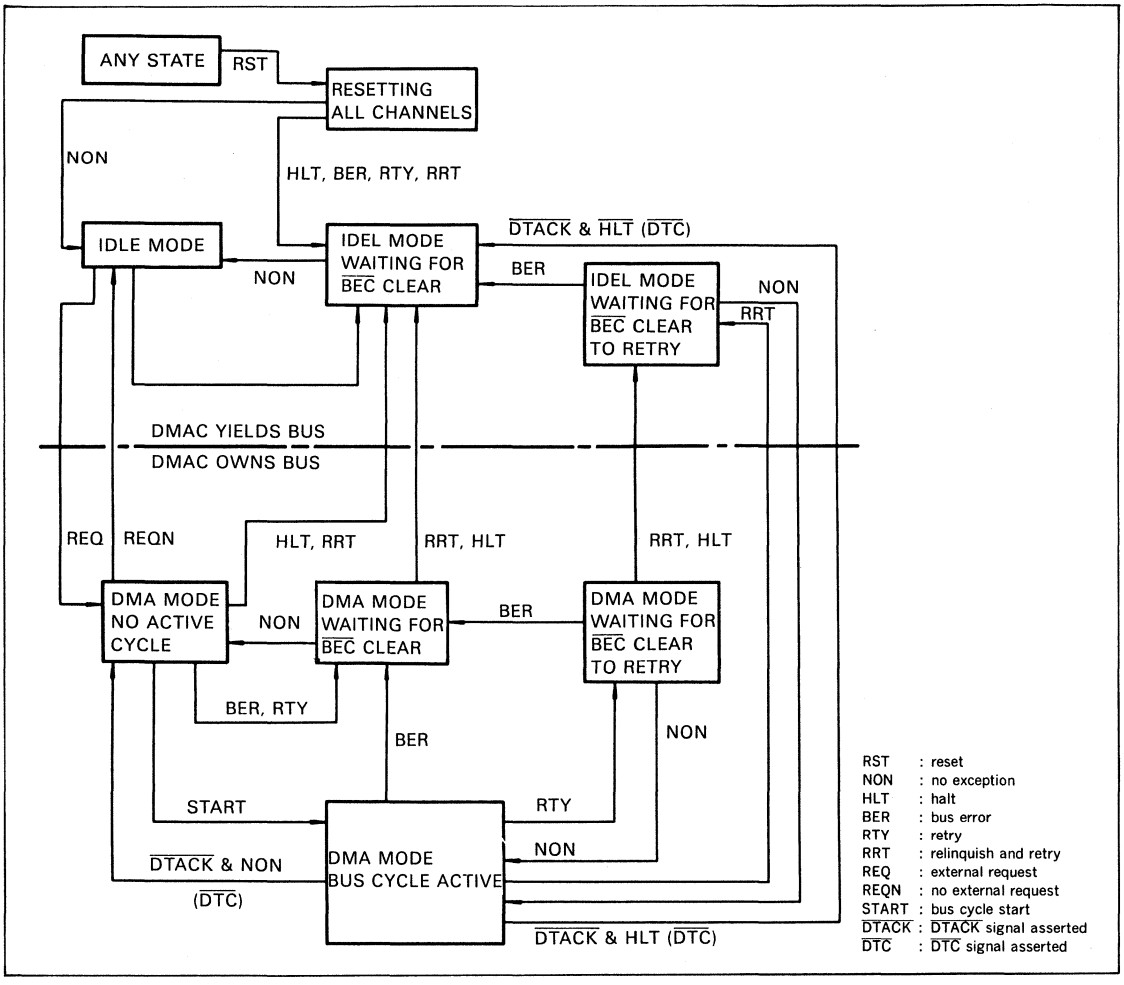


Figure 44 Bus Exception Flow Diagram

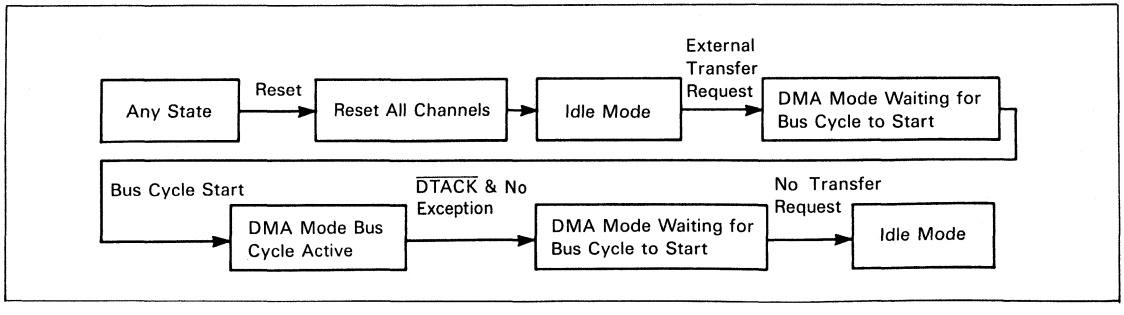


Figure 45 Flow of Normal Operation Without Exception or Error Condition

■CHANNEL PRIORITIES

Each channel has a priority level, which is determined by the contents of the Channel Priority Register (CPR). The priority of a channel is set by writing one of values $(00)_2$ through $(11)_2$, to CPR, $(00)_2$ being the highest priority level. When multiple requests are pending at the DMAC, the channel with the highest priority receives first service. The priority of a channel is independent of the device protocol or the request mechanism for that channel. If there are several requesting channels at the same priority level, a round-robin resolution is used, that is the DMAC does operand transfers in rotation starting form the channel of the lowest address.

Resetting the DMAC sets the priority level of all channels to "$(00)_2$", the highest priority level.

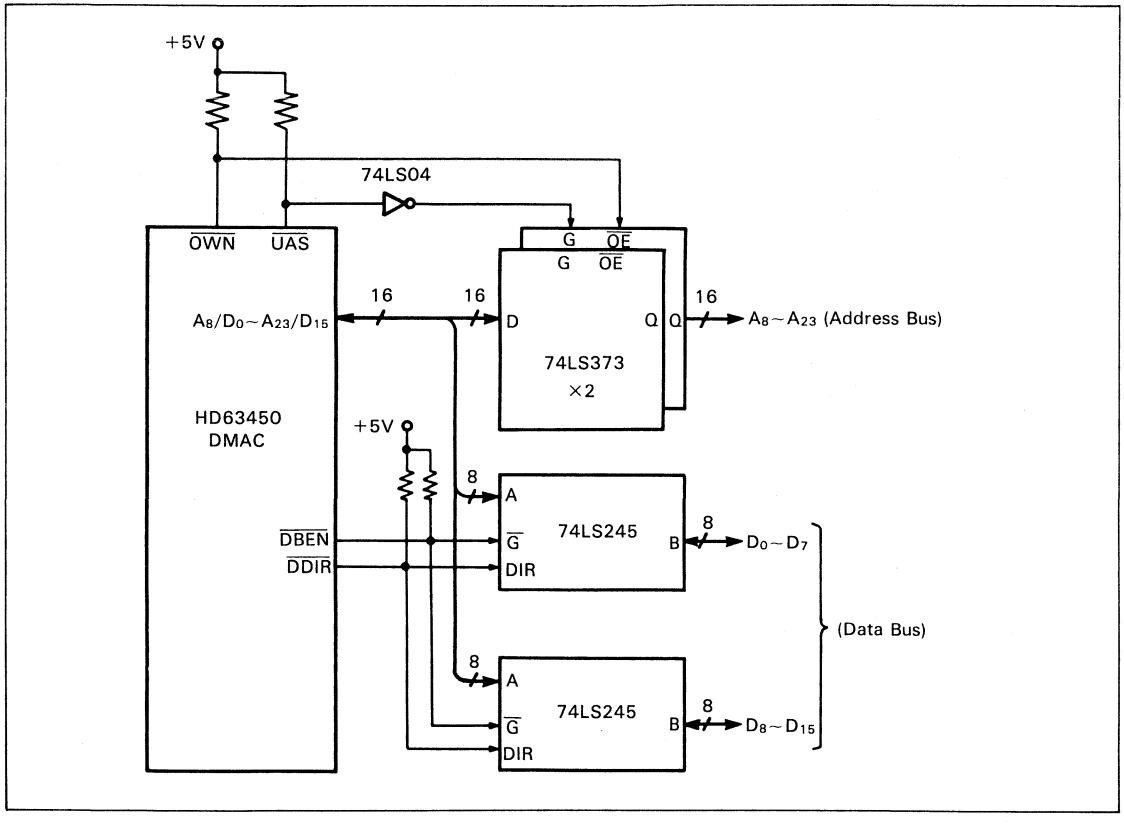


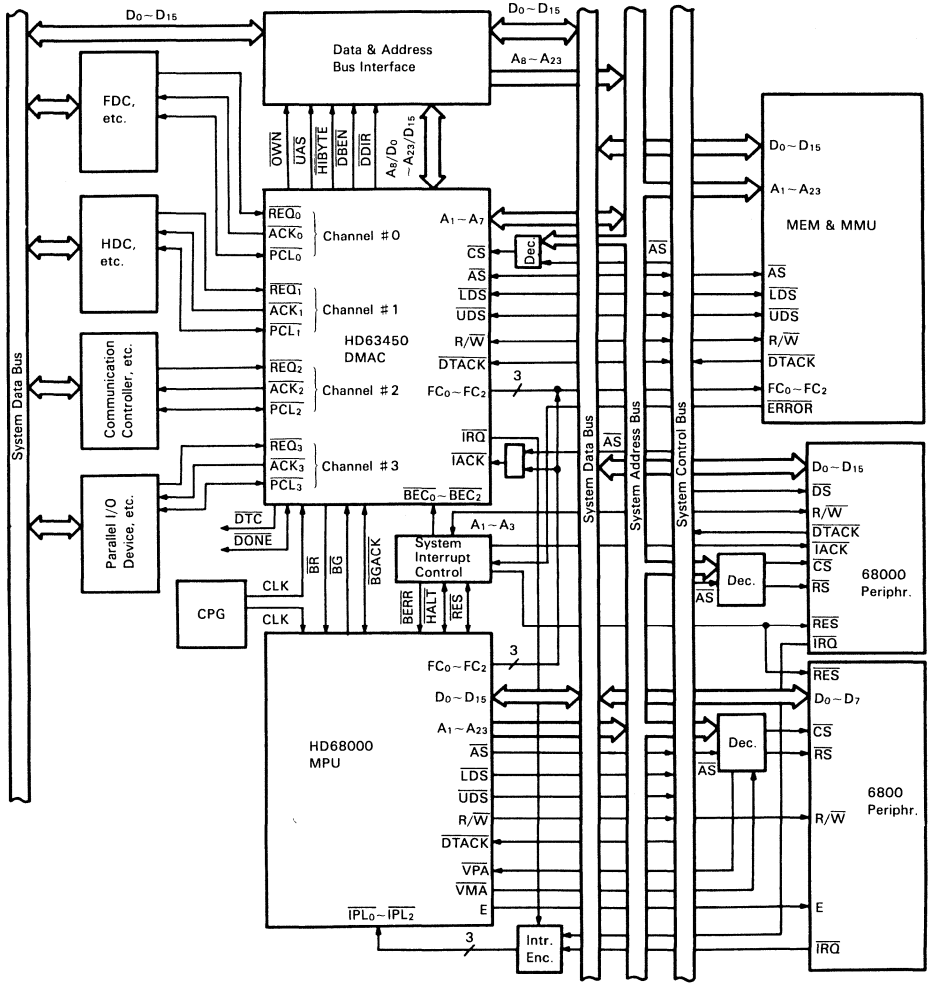
Figure 46 An Example of the Demultiplexed Address Data Bus

■ APPLICATIONS INFORMATION

Examples of how to interface HD63450 to an HD68000 based system are shown in Figures 46 and 47.

Figure 46 shows an example of how to demultiplex the address/data bus. $\overline{OWN}$ and $\overline{UAS}$ are used to control 74LS373 for latching the address. $\overline{DBEN}$ and $\overline{DDIR}$ are used to control the bi-directional buffer 74LS245. These signals are three-stated at active low, which requires pull-up resistors.

Figure 47 shows an example of inter-device connection in the HMCS68000 system. $\overline{REQ}$, $\overline{ACK}$, $\overline{PCL}$, $\overline{DTC}$, and $\overline{DONE}$ are used to control I/O devices.



The address bus and the system control bus in each device are omitted in this Figure

Figure 47 An Example of Inter-Device Connnection in the HMCS68000 System

■ NOTES FOR USING HD63450

(1) I/O device connection in 6800 mode

When the DMAC is reading data from the 6800 type peripheral device, the data is to be latched not at the falling edge of E clock but at that of CLK of $\overline{DTC}$ assertion. As shown in the figure below, the 74LS373 is externally required to keep this data on the bus of the DMAC.

(2) External abort during the $\overline{DONE}$ input cycle

In case of I/O device-to-Memory transfer under the dual addressing mode, the $\overline{DONE}$ input occurs during the read cycle of I/O device-to-DMAC registers.

The external abort ($\overline{PCL}$ is configured as an external abort input.) will be ignored during the write cycle which subsequently starts after the DMAC enters the $\overline{DONE}$ input cycle.

In this case, the registers CSR and $\overline{CER}$ indicate the normal transfer termination informed by the $\overline{DONE}$ input. When PCT = "1", ERR = "0", and NDT = "1" are set in CSR, an external abort has occurred.

(3) Multiple errors in one channel

In case of the sequential multiple errors in one channel, the DMAC indicates only the first one. The error code once set in CER is reserved until the ERR bit is reset.

(4) Attention for mounting the DMAC

The thick wiring is recommended to be used to connect the V_{SS} pin of the DMAC to the ground of the circuit board.

When using a socket, note that the V_{SS} pin should make a good contact with the socket.

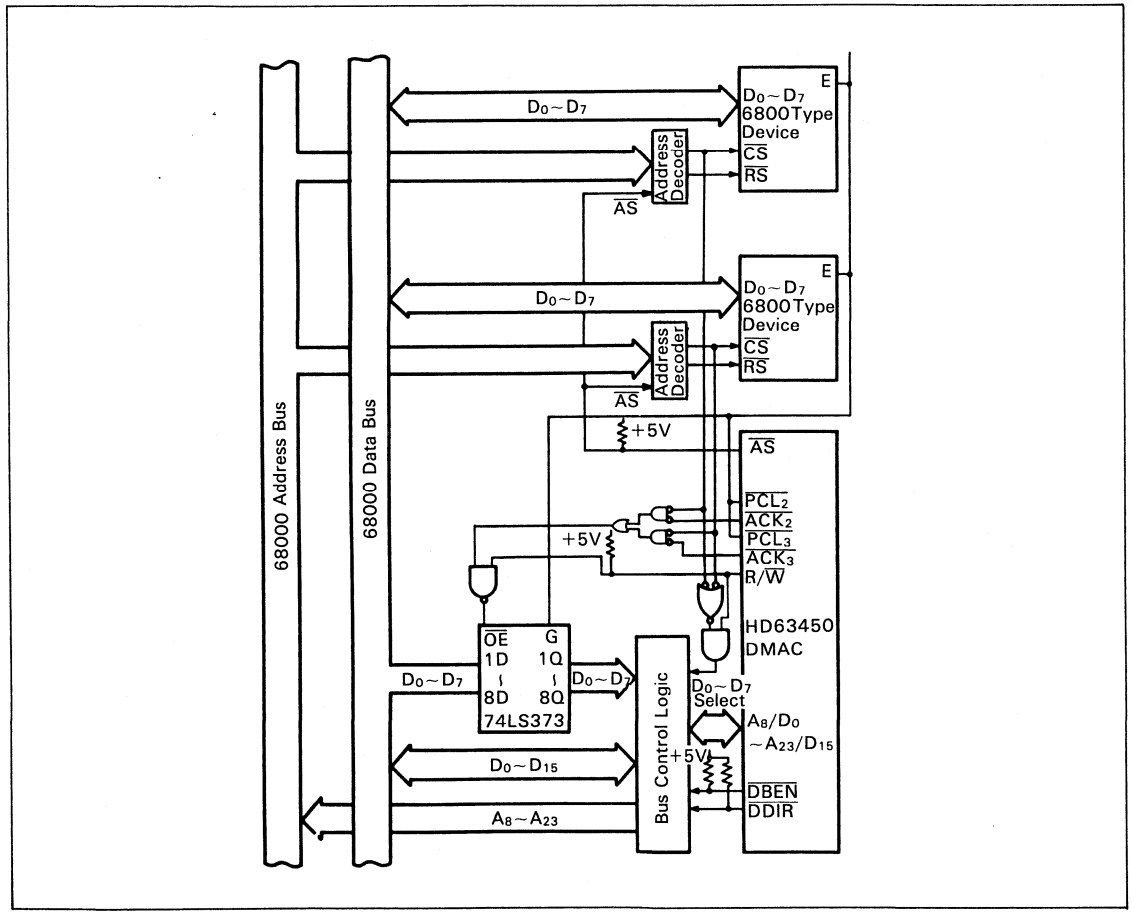


Figure 48 An Example of Connection with 6800 type Peripheral Devices (channel 2 and 3 are used)

HD68450, HD68450Y

DMAC (Direct Memory Access Controller)

Microprocessor implemented systems are becoming increasingly complex, particularly with the advent of high-performance 16-bit MPU devices with large memory addressing capability. In order to maintain high throughput, large blocks of data must be moved within these systems in a quick, efficient manner with minimum intervention by the MPU itself.

The HD68450 Direct Memory Access Controller (DMAC) is designed specifically to complement the performance and architectural capabilities of the HD68000 MPU by providing the following features:

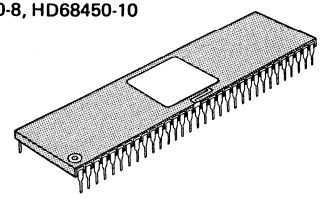
- HMCS68000 Bus Compatible
- 4 independent DMA Channels
- Memory-to-Memory, Memory-to-Device, Device-to-Memory Transfers
- MMU Compatible
- Array-Chained and Linked-Array-Chained Operations
- On-Chip Registers that allow Complete Software Control by the System MPU
- Interface Lines for Requesting, Acknowledging, and Incidental Control of the Peripheral Devices
- Variable System Bus Bandwidth Utilization
- Programmable Channel Prioritization
- 2 Vectored interrupts for each Channel
- Auto-Request and External-Request Transfer Modes
- +5 Volt Operation

The DMAC functions by transferring a series of operands (data) between memory and peripheral device; operand sizes can be byte, word, or long word. A block is a sequence of operations; the number of operands in a block is determined by a transfer count. A single-channel operation may involve the transfer of several blocks of data between memory and device.

■ TYPE OF PRODUCTS

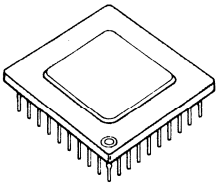
Type No.	Bus Timing	Packaging
HD68450-4	4MHz	DC-64
HD68450-6	6MHz	
HD68450-8	8MHz	
HD68450-10	10MHz	
HD68450Y4	4MHz	PGA-68
HD68450Y6	6MHz	
HD68450Y8	8MHz	
HD68450Y10	10MHz	

HD68450-4, HD68450-6,
HD68450-8, HD68450-10



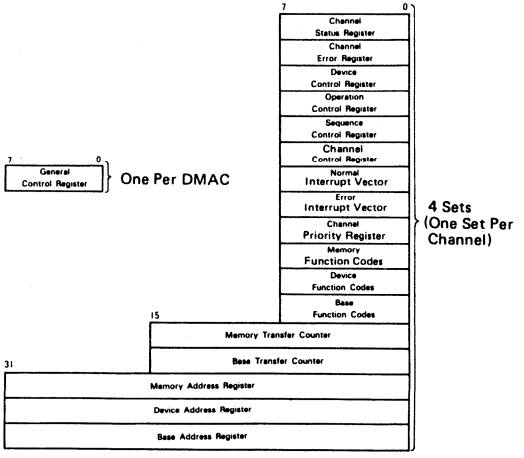
(DC-64)

HD68450Y4, HD68450Y6,
HD68450Y8,
HD68450Y10



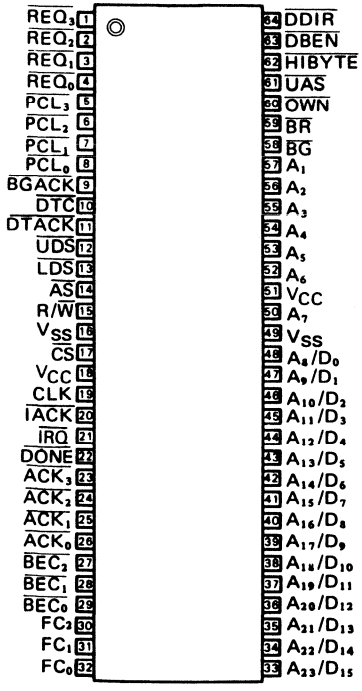
(PGA-68)

■ PROGRAMMING MODEL



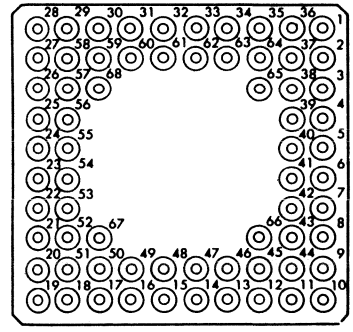
■ PIN ARRANGEMENT

● HD68450



(Top View)

● HD68450Y



(Bottom View)

Pin No.	Function	Pin No.	Function	Pin No.	Function	Pin No.	Function
1	N/C	18	$\overline{PCL_1}$	35	A_{19}/D_{11}	52	$\overline{BGACK}$
2	A_{13}/D_5	19	$\overline{DTACK}$	36	A_{17}/D_9	53	$\overline{LDS}$
3	A_{11}/D_3	20	$\overline{UDS}$	37	A_{15}/D_7	54	Vss
4	A_{10}/D_2	21	$\overline{AS}$	38	A_{12}/D_4	55	Vcc
5	A_8/D_0	22	R/$\overline{W}$	39	A_9/D_1	56	$\overline{DONE}$
6	A_7	23	N/C	40	V_{SS}	57	$\overline{IRQ}$
7	A_6	24	$\overline{CS}$	41	V_{CC}	58	$\overline{ACK_2}$
8	A_5	25	CLK	42	A_4	59	$\overline{BEC_2}$
9	A_3	26	$\overline{IACK}$	43	A_2	60	$\overline{BEC_0}$
10	N/C	27	$\overline{ACK_3}$	44	$\overline{BG}$	61	FC_0
11	$\overline{BR}$	28	$\overline{ACK_0}$	45	$\overline{OWN}$	62	A_{21}/D_{13}
12	$\overline{UAS}$	29	$\overline{BEC_1}$	46	$\overline{HIBYTE}$	63	A_{18}/D_{10}
13	$\overline{DBEN}$	30	FC_2	47	$\overline{DDIR}$	64	A_{16}/D_8
14	$\overline{REQ_3}$	31	FC_1	48	$\overline{REQ_1}$	65	A_{14}/D_6
15	$\overline{REQ_2}$	32	A_{23}/D_{15}	49	$\overline{PCL_2}$	66	A_1
16	$\overline{REQ_0}$	33	A_{22}/D_{14}	50	$\overline{PCL_0}$	67	$\overline{DTC}$
17	$\overline{PCL_3}$	34	A_{20}/D_{12}	51	N/C	68	$\overline{ACK_1}$

■ ABSOLUTE MAXIMUM RATINGS

Item	Symbol	Value	Unit
Supply Voltage	V_{CC}*	-0.3 ~ +7.0	V
Input Voltage	V_{in}*	-0.3 ~ +7.0	V
Operating Temperature Range	T_{opr}	0 ~ +70	°C
Storage Temperature	T_{stg}	-55 ~ +150	°C

* With respect to V_{SS} (SYSTEM GND)

(NOTE) Permanent LSI damage may occur if maximum ratings are exceeded. Normal operation should be under recommended operating conditions. If these conditions are exceeded, it could affect reliability of LSI.

■ RECOMMENDED OPERATING CONDITIONS

Item	Symbol	min	typ	max	Unit
Supply Voltage	V_{CC}*	4.75	5.0	5.25	V
Input Voltage	V_{IH}*	2.0	–	V_{CC}	V
	V_{IL}*	-0.3	–	0.8	V
Operating Temperature	T_{opr}	0	25	70	°C

* With respect to V_{SS} (SYSTEM GND)

■ ELECTRICAL CHARACTERISTICS

● DC CHARACTERISTICS (V_{CC} = 5V ±5%, V_{SS} = 0V, Ta = 0 ~ +70°C, unless otherwise noted.)

Item		Symbol	Test Condition	min	typ	max	Unit
Input "High" Voltage		V_{IH}		2.0	–	V_{CC}	V
Input "Low" Voltage		V_{IL}		V_{SS} -0.3	–	0.8	V
Input Leakage Current	$\overline{CS}$, $\overline{IACK}$, $\overline{BG}$, CLK, $\overline{BEC_0}$ ~ $\overline{BEC_2}$, $\overline{REQ_0}$ ~ $\overline{REQ_3}$	I_{in}		–	–	10	μA
Three-State (Off State) Input Current	A_1 ~ A_7, D_0 ~ D_{15}/A_8 ~ A_{23}, $\overline{AS}$, $\overline{UDS}$, $\overline{LDS}$, R/$\overline{W}$, $\overline{UAS}$, $\overline{DTACK}$, $\overline{BGACK}$, $\overline{OWN}$, DTC, $\overline{HIBYTE}$, $\overline{DDIR}$, $\overline{DBEN}$, FC_0 ~ FC_2	I_{TSI}		–	–	10	μA
Open Drain (Off State) Input Current	$\overline{IRQ}$, $\overline{DONE}$	I_{ODI}		–	–	20	μA
Output "High" Voltage	A_1 ~ A_7, D_0 ~ D_{15}/A_8 ~ A_{23}, $\overline{AS}$, $\overline{UDS}$, $\overline{LDS}$, R/$\overline{W}$, $\overline{UAS}$, $\overline{DTACK}$, $\overline{BGACK}$, $\overline{BR}$, $\overline{OWN}$, $\overline{DTC}$, $\overline{HIBYTE}$, $\overline{DDIR}$, $\overline{DBEN}$, $\overline{ACK_0}$ ~ $\overline{ACK_3}$, $\overline{PCL_0}$ ~ $\overline{PCL_3}$, FC_0 ~ FC_2	V_{OH}	I_{OH} = -400 μA	2.4	–	–	V
Output "Low" Voltage	A_1 ~ A_7, FC_0 ~ FC_2	V_{OL}	I_{OL} = 3.2 mA	–	–	0.5	V
	D_0 ~ D_{15}/A_8 ~ A_{23}, $\overline{AS}$, $\overline{UDS}$, $\overline{LDS}$, R/$\overline{W}$, $\overline{DTACK}$, $\overline{BR}$, $\overline{OWN}$, $\overline{DTC}$, $\overline{HIBYTE}$, $\overline{DDIR}$, $\overline{DBEN}$, $\overline{ACK_0}$ ~ $\overline{ACK_3}$, $\overline{UAS}$, $\overline{PCL_0}$ ~ $\overline{PCL_3}$, $\overline{BGACK}$	V_{OL}	I_{OL} = 5.3 mA	–	–	0.5	
	$\overline{IRQ}$, $\overline{DONE}$	V_{OL}	I_{OL} = 8.9 mA	–	–	0.5	
Power Dissipation		P_D	f = 8 MHz, V_{CC} = 5.0 V, Ta = 25°C	–	1.4	2.0	W
Capacitance		C_{in}	V_{in} = 0V, Ta = 25°C, f = 1 MHz	–	–	15	pF

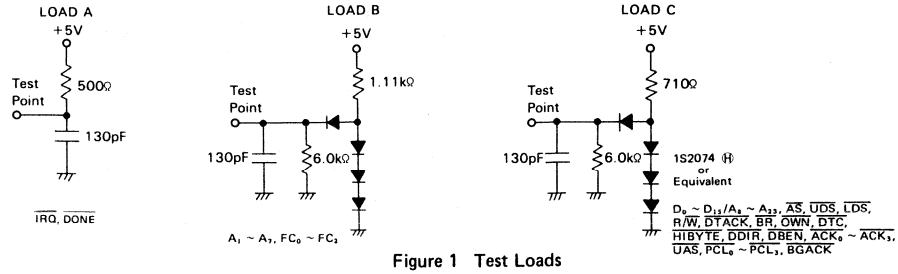


Figure 1 Test Loads

• AC ELECTRICAL SPECIFICATIONS (V_{CC} = 5V ±5%, V_{SS} = 0V, Ta = 0~+70°C)

No.	Item	Symbol	Test Condition	4MHz HD68450-4 HD68450Y4		6MHz HD68450-6 HD68450Y6		8MHz HD68450-8 HD68450Y8		10MHz HD68450-10 HD68450Y10		Unit
				min	max	min	max	min	max	min	max	
	Frequency of Operation	f		2	4	2	6	2	8	2	10	MHz
1	Clock Period	t_{cyc}		250	500	167	500	126	500	100	500	ns
2	Clock Width Low	t_{CL}		115	250	75	250	55	250	45	250	ns
3	Clock Width High	t_{CH}		115	250	75	250	55	250	45	250	ns
4	Clock Fall Time	t_{Cf}		–	10	–	10	–	10	–	10	ns
5	Clock Rise Time	t_{Cr}		–	10	–	10	–	10	–	10	ns
6	Asynchronous Input Setup Time	t_{ASI}		30	–	25	–	20	–	15	–	ns
7	Data in to $\overline{DBEN}$ Low	t_{DIDBL}		0	–	0	–	0	–	0	–	ns
8	$\overline{DTACK}$ Low to Data Invalid	t_{DTLDI}		0	–	0	–	0	–	0	–	ns
9	Address in to $\overline{AS}$ in Low	t_{AIASL}		0	–	0	–	0	–	0	–	ns
10	$\overline{AS}$, $\overline{DS}$ in High to Address in Invalid	t_{SIHAIV}		0	–	0	–	0	–	0	–	ns
10A	$\overline{DS}$ in High to $\overline{CS}$ High	t_{DSHCSH}		–	1.0	–	1.0	–	1.0	–	1.0	clk. per.
11	Clock High to $\overline{DDIR}$ Low	t_{CHDRL}		–	90	–	80	–	70	–	60	ns
12	Clock High to $\overline{DDIR}$ High	t_{CHDRH}		–	90	–	80	–	70	–	60	ns
13	$\overline{DS}$ in High to $\overline{DDIR}$ High Impedance	t_{DSHDRZ}		–	160	–	140	–	120	–	110	ns
14	Clock Low to $\overline{DBEN}$ Low	t_{CLDBL}		–	90	–	80	–	70	–	60	ns
15	Clock Low to $\overline{DBEN}$ High	t_{CLDBH}		–	90	–	80	–	70	–	60	ns
16	$\overline{DS}$ in High to $\overline{DBEN}$ High Impedance	t_{DSHDBZ}		–	160	–	140	–	120	–	110	ns
17	Clock High to Data Out Valid (MPU read)	t_{CHDVM}		–	290	–	230	–	180	–	160	ns
18	$\overline{DS}$ in High to Data Out Invalid	t_{DSHDZn}		0	–	0	–	0	–	0	–	ns
19	$\overline{DS}$ in High to Data High Impedance	t_{DSHDZ}		–	160	–	140	–	120	–	110	ns
20	Clock Low to $\overline{DTACK}$ Low	t_{CLDTL}		–	90	–	80	–	70	–	60	ns
21	$\overline{DS}$ in High to $\overline{DTACK}$ High	t_{DSHDTH}		–	160	–	130	–	110	–	110	ns
22	$\overline{DTACK}$ Width High	t_{DTH}		10	–	10	–	10	–	10	–	ns
23	$\overline{DS}$ in High to $\overline{DTACK}$ High Impedance	t_{DSHDTZ}		–	220	–	200	–	180	–	160	ns
24	$\overline{DTACK}$ Low to $\overline{DS}$ in High	t_{DTLDSH}		0	–	0	–	0	–	0	–	ns
25	$\overline{REQ}$ Width Low	t_{REQL}		2.0	–	2.0	–	2.0	–	2.0	–	clk. per.
26	$\overline{REQ}$ Low to $\overline{BR}$ Low	t_{RELBRL}		500	–	334	–	250	–	200	–	ns
27	Clock High to $\overline{BR}$ Low	t_{CHBRL}	Fig. 1 ~ Fig. 8	–	90	–	80	–	70	–	60	ns
28	Clock High to $\overline{BR}$ High	t_{CHBRH}		–	90	–	80	–	70	–	60	ns
29	$\overline{BG}$ Low to $\overline{BGACK}$ Low	t_{BGLBL}		4.5	–	4.5	–	4.5	–	4.5	–	clk. per.
30	$\overline{BR}$ Low to MPU Cycle End ($\overline{AS}$ in High)	t_{BRLASH}		0	–	0	–	0	–	0	–	ns
31	MPU Cycle End ($\overline{AS}$ in High) to $\overline{BGACK}$ Low	t_{ASHBL}		4.5	5.5	4.5	5.5	4.5	5.5	4.5	5.5	clk. per.
32	$\overline{REQ}$ Low to $\overline{BGACK}$ Low	t_{REQLBL}		12.0	–	12.0	–	12.0	–	12.0	–	clk. per.
33	Clock High to $\overline{BGACK}$ High	t_{CHBL}		–	90	–	80	–	70	–	60	ns
34	Clock High to $\overline{BGACK}$ High	t_{CHBH}		–	90	–	80	–	70	–	60	ns
35	Clock Low to $\overline{BGACK}$ High Impedance	t_{CLBZ}		–	120	–	100	–	80	–	70	ns
36	Clock High to FC Valid	t_{CHFCV}		–	140	–	120	–	100	–	90	ns
37*	Clock High to Address Valid	t_{CHAV}		–	160	–	140	–	120	–	110	ns
38	Clock High to Address/FC/Data High Impedance	t_{CHAZx}		–	140	–	120	–	100	–	100	ns
39	Clock High to Address/FC/Data Invalid	t_{CHAZn}		0	–	0	–	0	–	0	–	ns
40	Clock Low to Address High Impedance	t_{CLAZ}		–	140	–	120	–	100	–	90	ns
41	Clock High to $\overline{UAS}$ Low	t_{CHUL}		–	90	–	80	–	70	–	60	ns
42	Clock High to $\overline{UAS}$ High	t_{CHUH}		–	90	–	80	–	70	–	60	ns
43	Clock Low to $\overline{UAS}$ High Impedance	t_{CLUZ}		–	120	–	100	–	80	–	70	ns
44	$\overline{UAS}$ High to Address Invalid	t_{UHAI}		50	–	40	–	30	–	20	–	ns
45	Clock High to $\overline{AS}$, $\overline{DS}$ Low	t_{CHSL}		–	80	–	70	–	60	–	55	ns
46	Clock Low to $\overline{DS}$ Low (write)	t_{CLDSL}		–	80	–	70	–	60	–	55	ns
47	Clock Low to $\overline{AS}$, $\overline{DS}$ High	t_{CLSH}		–	90	–	80	–	70	–	60	ns
48	Clock Low to $\overline{AS}$, $\overline{DS}$ High Impedance	t_{CLSZ}		–	120	–	100	–	80	–	70	ns
49	$\overline{AS}$ Width Low	t_{ASL}		545	–	350	–	255	–	195	–	ns
50	$\overline{DS}$ Width Low	t_{DSL}		420	–	265	–	190	–	145	–	ns
51	$\overline{AS}$, $\overline{DS}$ Width High	t_{SH}		285	–	180	–	150	–	105	–	ns
52	Address/FC Valid to $\overline{AS}$, $\overline{DS}$ Low	t_{AVSL}		50	–	40	–	30	–	20	–	ns
53	$\overline{AS}$, $\overline{DS}$ High to Address/FC/Data Invalid	t_{SHAZ}		50	–	40	–	30	–	20	–	ns
54	Clock High to R/$\overline{W}$ Low	t_{CHRL}		–	90	–	80	–	70	–	60	ns
55	Clock High to R/$\overline{W}$ High	t_{CHRH}		–	90	–	80	–	70	–	60	ns

(to be continued)

No.	Item	Symbol	Test Condition	4MHz HD68450-4 HD68450Y4		6MHz HD68450-6 HD68450Y6		8MHz HD68450-8 HD68450Y8		10MHz HD68450-10 HD68450Y10		Unit
				min	max	min	max	min	max	min	max	
56	Clock Low to R/$\overline{W}$ High Impedance	t_{CLRZ}	Fig. 1 ~ Fig. 8	–	120	–	100	–	80	–	70	ns
57	Address/FC Valid to R/$\overline{W}$ Low	t_{AVRL}		100	–	40	–	20	–	10	–	ns
58	R/$\overline{W}$ Low to $\overline{DS}$ Low (write)	t_{RLSL}		285	–	170	–	120	–	90	–	ns
59	$\overline{DS}$ High to R/$\overline{W}$ High	t_{SHRH}		60	–	50	–	40	–	20	–	ns
60	Clock Low to $\overline{OWN}$ Low	t_{CLOL}		–	90	–	80	–	70	–	60	ns
61	Clock Low to $\overline{OWN}$ High	t_{CLOH}		–	90	–	80	–	70	–	60	ns
62	Clock High to $\overline{OWN}$ High Impedance	t_{CHOZ}		–	120	–	100	–	80	–	70	ns
63	$\overline{OWN}$ Low to $\overline{BGACK}$ Low	t_{OLBL}		50	–	40	–	30	–	20	–	ns
64	$\overline{BGACK}$ High to $\overline{OWN}$ High	t_{BHOH}		50	–	40	–	30	–	20	–	ns
65	$\overline{OWN}$ Low to $\overline{UAS}$ Low	t_{OLUL}		50	–	40	–	30	–	20	–	ns
66	Clock High to $\overline{ACK}$ Low	t_{CHACL}		–	90	–	80	–	70	–	60	ns
67	Clock Low to $\overline{ACK}$ Low	t_{CLACL}		–	90	–	80	–	70	–	60	ns
68	Clock High to $\overline{ACK}$ High	t_{CHACH}		–	90	–	80	–	70	–	60	ns
69	$\overline{ACK}$ Low to $\overline{DS}$ Low	t_{ACLDSL}		230	–	140	–	100	–	80	–	ns
70	$\overline{DS}$ High to $\overline{ACK}$ High	t_{DSHACH}		50	–	40	–	30	–	20	–	ns
71	Clock High to $\overline{HIBYTE}$ Low	t_{CHHIL}		–	90	–	80	–	70	–	60	ns
72	Clock Low to $\overline{HIBYTE}$ Low	t_{CLHIL}		–	90	–	80	–	70	–	60	ns
73	Clock High to $\overline{HIBYTE}$ High	t_{CHHIH}		–	90	–	80	–	70	–	60	ns
74	Clock Low to $\overline{HIBYTE}$ High Impedance	t_{CLHIZ}		–	120	–	100	–	80	–	70	ns
75	Clock High to $\overline{DTC}$ Low	t_{CHDTL}		–	90	–	80	–	70	–	60	ns
76	Clock High to $\overline{DTC}$ High	t_{CHDTH}		–	90	–	80	–	70	–	60	ns
77	Clock Low to $\overline{DTC}$ High Impedance	t_{CLDTZ}		–	120	–	100	–	80	–	70	ns
78	$\overline{DTC}$ Width Low	t_{DTCL}		230	–	147	–	105	–	80	–	ns
79	$\overline{DTC}$ Low to $\overline{DS}$ High	t_{DTLDH}		95	–	50	–	30	–	20	–	ns
80	Clock High to $\overline{DONE}$ Low	t_{CHDOL}		–	90	–	80	–	70	–	60	ns
81	Clock Low to $\overline{DONE}$ Low	t_{CLDOL}		–	90	–	80	–	70	–	60	ns
82	Clock High to $\overline{DONE}$ High	t_{CHDOH}		–	150	–	140	–	130	–	120	ns
83	Clock Low to $\overline{DDIR}$ High Impedance	t_{CLDRZ}		–	120	–	100	–	80	–	70	ns
84	Clock Low to $\overline{DBEN}$ High Impedance	t_{CLDBZ}		–	120	–	100	–	80	–	70	ns
85	$\overline{DDIR}$ Low to $\overline{DBEN}$ Low	t_{DRLDBL}		50	–	40	–	30	–	20	–	ns
86	$\overline{DBEN}$ High to $\overline{DDIR}$ High	t_{DBHDRH}		50	–	40	–	30	–	20	–	ns
87	$\overline{DBEN}$ Low to Address/Data High Impedance	t_{DBLAZ}		–	17	–	17	–	17	–	17	ns
88	Clock Low to $\overline{PCL}$ Low (1/8 clock)	t_{CLPL}		–	90	–	80	–	70	–	60	ns
89	Clock Low to $\overline{PCL}$ High (1/8 clock)	t_{CLPH}		–	90	–	80	–	70	–	60	ns
90	$\overline{PCL}$ Width Low (1/8 clock)	t_{PCLL}		4.0	–	4.0	–	4.0	–	4.0	–	clk. per.
91	$\overline{DTACK}$ Low to Data In (setup time)	t_{DALDI}		–	320	–	200	–	150	–	115	ns
92	$\overline{DS}$ High to Data Invalid (hold time)	t_{SHDI}		0	–	0	–	0	–	0	–	ns
93	$\overline{DS}$ High to $\overline{DTACK}$ High	t_{SHDAH}		0	240	0	160	0	120	0	90	ns
94	Data Out Valid to $\overline{DS}$ Low	t_{DOSL}		0	–	0	–	0	–	0	–	ns
95	Data In to Clock Low (setup time)	t_{DICL}		30	–	25	–	15	–	15	–	ns
96	$\overline{BEC}$ Low to $\overline{DTACK}$ Low	t_{BECDAL}		50	–	50	–	50	–	50	–	ns
97	$\overline{BEC}$ Width Low	t_{BECL}		2.0	–	2.0	–	2.0	–	2.0	–	clk. per.
98	Clock High to $\overline{IRQ}$ Low	t_{CHIRL}		–	90	–	80	–	70	–	60	ns
99	Clock High to $\overline{IRQ}$ High	t_{CHIRH}		–	150	–	140	–	130	–	120	ns
100	$\overline{READY}$ In to $\overline{DTC}$ Low (Read)	t_{RALDTL}		270	–	180	–	145	–	120	–	ns
101	$\overline{READY}$ In to $\overline{DS}$ Low (Write)	t_{RALDSL}		395	–	240	–	205	–	170	–	ns
102	$\overline{DS}$ High to $\overline{READY}$ High	t_{DSHRAH}		0	240	0	160	0	120	0	90	ns
103	$\overline{DONE}$ In Low to $\overline{DTACK}$ Low	t_{DOLDAL}		50	–	50	–	50	–	50	–	ns
104	$\overline{DS}$ High to $\overline{DONE}$ In High	t_{DSHDOH}		0	240	0	160	0	120	0	90	ns
105	Asynchronous Input Hold Time	t_{ASIH}		15	–	15	–	15	–	15	–	ns

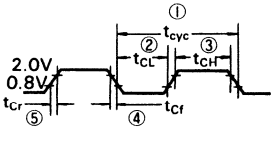
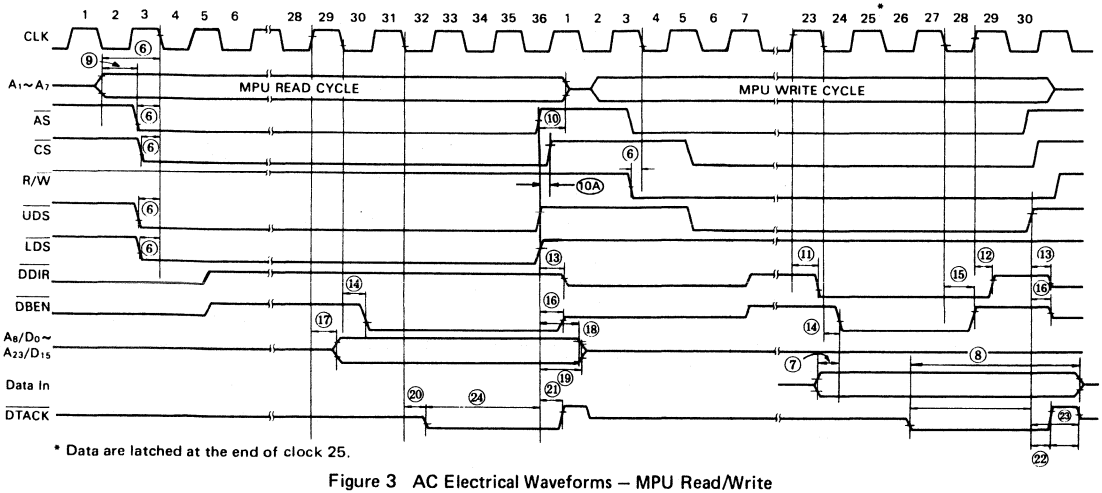
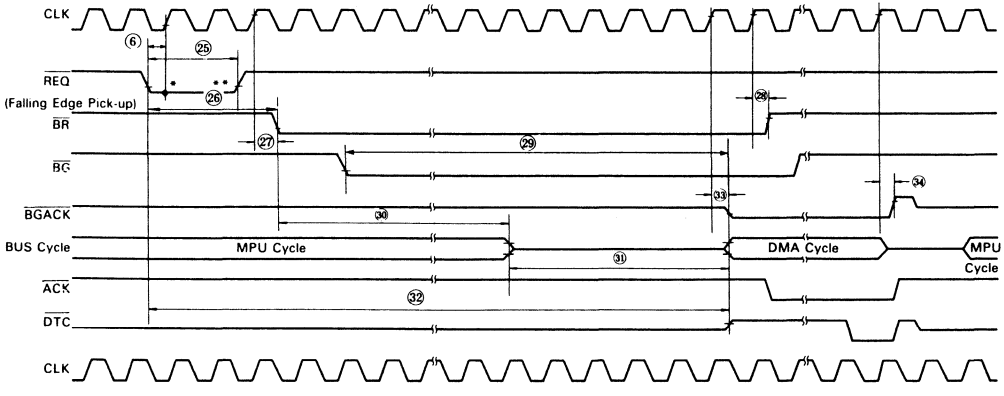


Figure 2 Input Clock Waveform



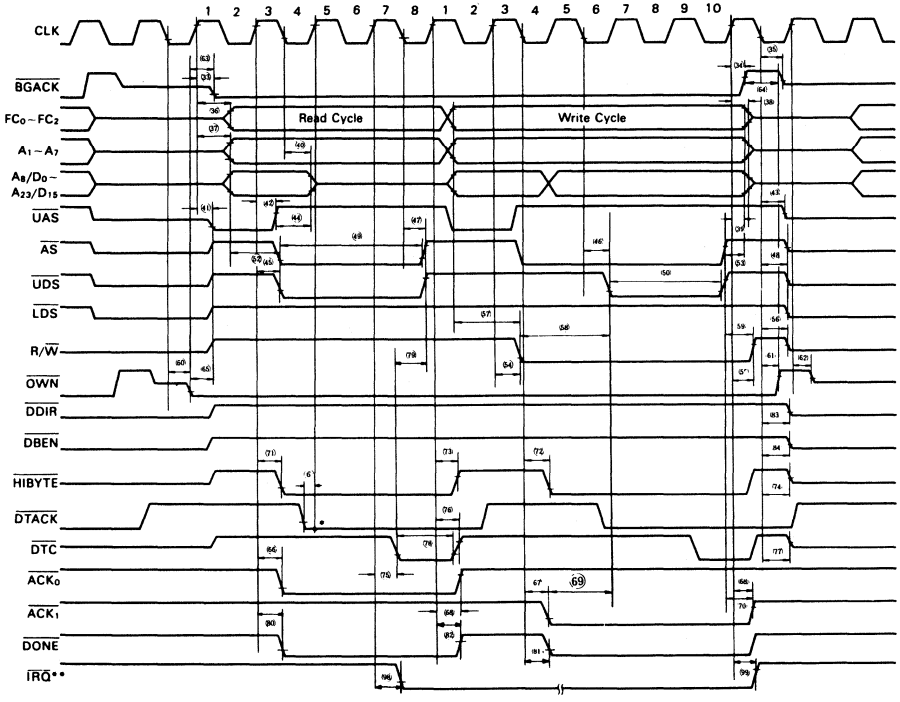
* Data are latched at the end of clock 25.

Figure 3 AC Electrical Waveforms – MPU Read/Write



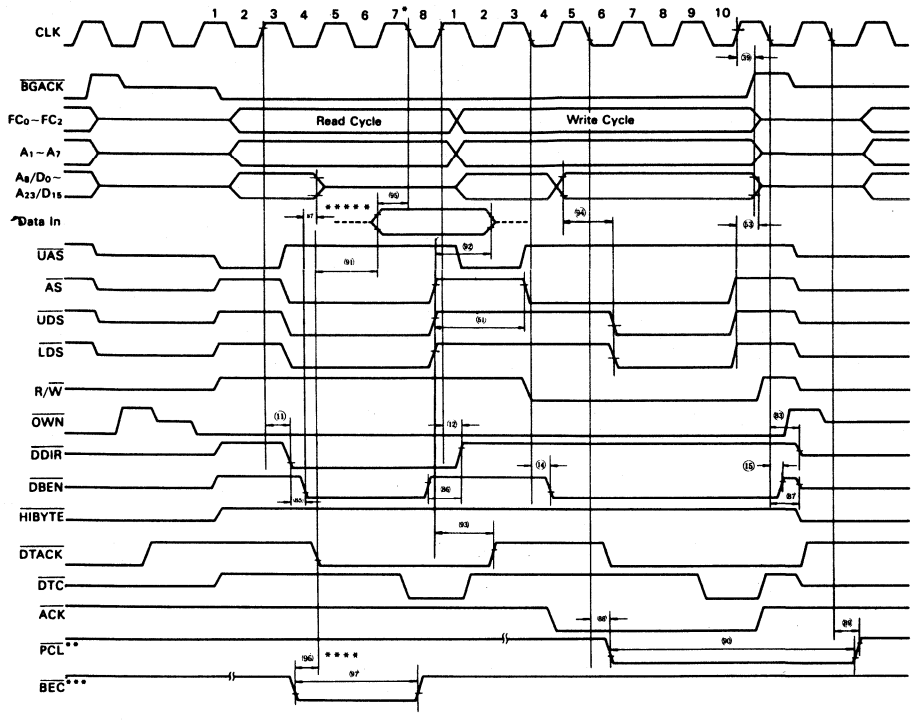
* REQ is picked up at the rising edge of CLK in cycle steal and Burst modes.
** BR isn't asserted while some BEC exception condition exists or DMAC is accessed by MPU.

Figure 4 AC Electrical Waveforms – Bus Arbitration



* $\overline{\text{DTACK}}$ is picked up at the rising edge of CLK. This is different from HD68000.
** This timing is not related to DMA Read/Write (Single Cycle) sequence.

Figure 5 AC Electrical Waveforms — DMA Read/Write (Single Cycle)



* Data are latched at the end of clock 7. This timing is the same as HD68000.
** This timing is not related to DMA Read/Write (Dual Cycle) sequence. This timing is only applicable when 1/8 clock pulse mode is selected.
*** This timing is applicable when a bus exception occures.
**** If #6 is satisfied for both $\overline{\text{DTACK}}$ and $\overline{\text{BEC}}$, #96 may be Ons.
***** If the propagation delay of the external bidirectional buffer LS245 is less than 17nsec, the conflict may occur between the address output of the DMAC and the system data bus. In this case, the output of $\overline{\text{DBEN}}$ must be delayed externally.

Figure 6 AC Electrical Waveforms — DMA Read/Write (Dual Cycle)

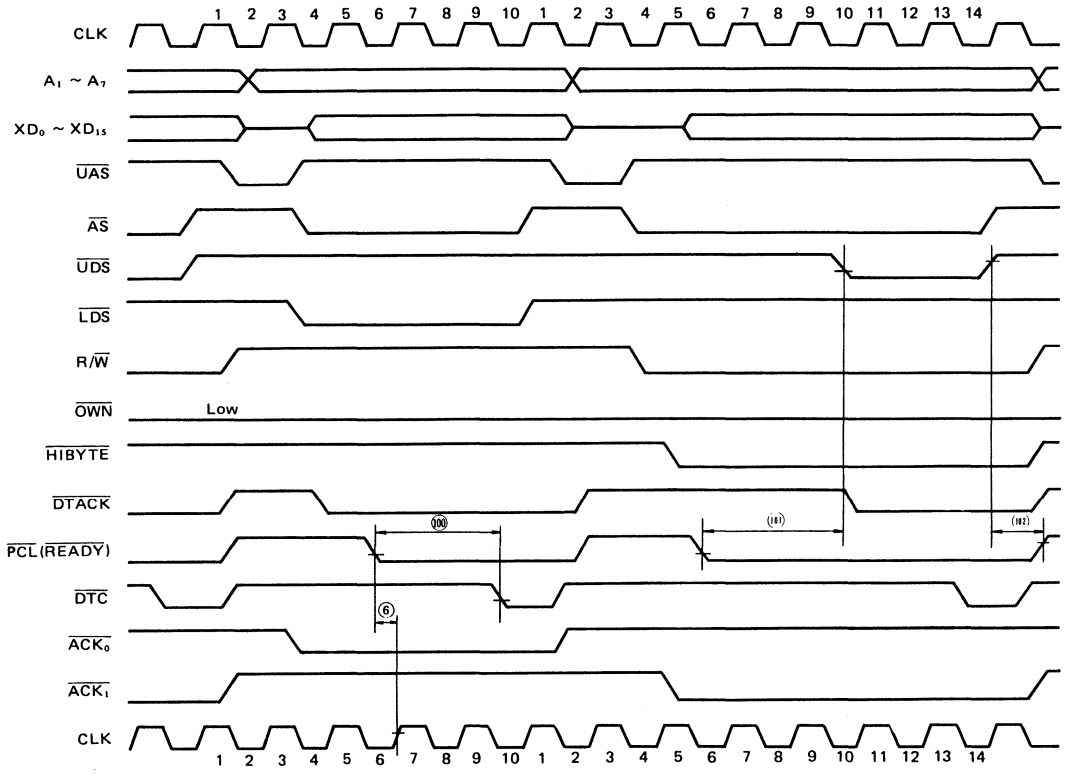
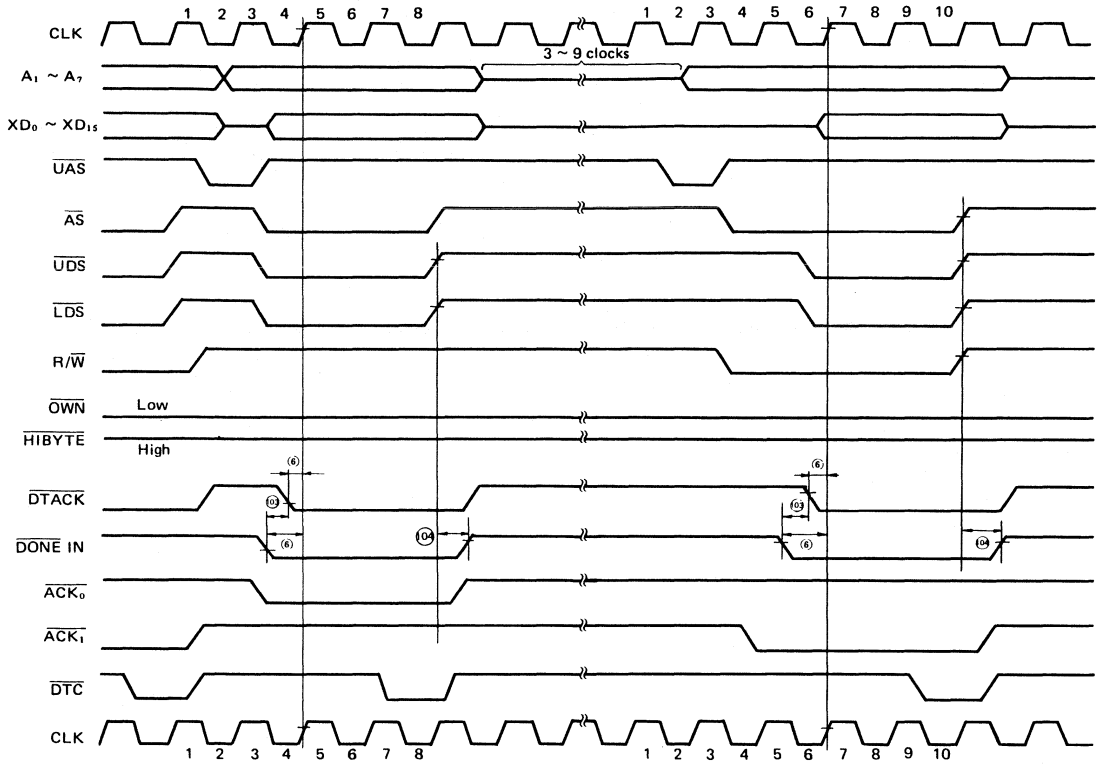


Figure 7 AC Electrical Waveforms — DMA Read/Write (Single Cycle with $\overline{PCL}$)



* If #6 is satisfied for both $\overline{\text{DTACK}}$ and $\overline{\text{DONE}}$, #103 may be 0ns.

Figure 8 AC Electrical Waveforms – $\overline{\text{DONE}}$ Input

(NOTES for Figure 3 through 8)

1) Setup time for the asynchronous inputs $\overline{\text{BG}}$, $\overline{\text{BGACK}}$, $\overline{\text{CS}}$, $\overline{\text{IACK}}$, $\overline{\text{AS}}$, $\overline{\text{UDS}}$, $\overline{\text{LDS}}$, and R/$\overline{\text{W}}$ guarantees their recognition at the next falling edge of the clock. Setup time for $\overline{\text{BEC}}_0 \sim \overline{\text{BEC}}_2$, $\overline{\text{REQ}}_0 \sim \overline{\text{REQ}}_3$, $\overline{\text{PCL}}_0 \sim \overline{\text{PCL}}_3$, $\overline{\text{DTACK}}$, and $\overline{\text{DONE}}$ guarantees their recognition at the next rising edge of the clock.
2) Timing measurements are referenced to and from a low voltage of 0.8 volts and a high voltage of 2.0 volts.
3) These waveforms should only be referenced in regard to the edge-to-edge measurement of the timing specifications. They are not intended as a functional description of the input and output signals. Refer to other functional descriptions and their related diagrams for device operation.

■ SIGNAL DESCRIPTION

The following section identifies the signals used in the DMAC. In the definitions, "MPU mode" refers to the state when the DMAC is chip selected by MPU. The term "DMA mode" refers to the state when the DMAC assumes ownership of the bus. The DMAC is in the "IDLE mode" at all other times. Moreover, the DMA bus cycle refers to the bus cycle that is executed by the DMAC in the "DMA mode".

NOTE) In this data sheet, the state of the signals is described with these words: active or assert, inactive or negate.

This is done to avoid confusion when dealing with a mixture of "active-low" and "active-high" signals. The term assert or assertion is used to indicate that a signal is active or true independent of whether that voltage is low or high. The term negate or negation is used to indicate that a signal is inactive or false.

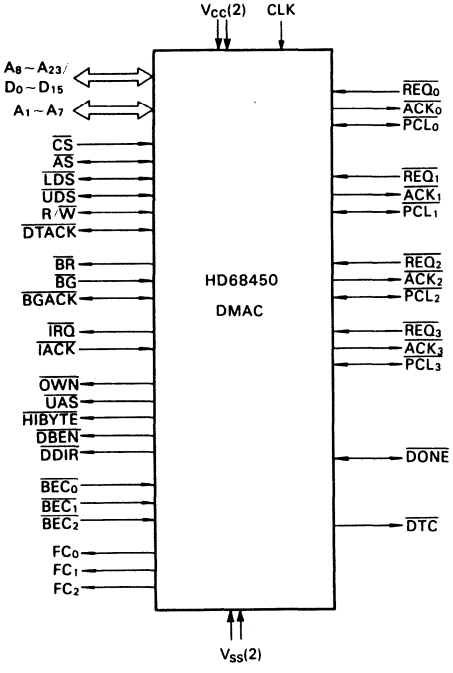


Figure 9 Input and Output Signals

- **Address/Data Bus (A_8/D_0 through A_{23}/D_{15})**

Input/Output Active-high	Three-statable

These lines are time multiplexed for address and data bus. The lines $\overline{DDIR}$, $\overline{DBEN}$, $\overline{UAS}$ and $\overline{OWN}$ are used to control the demultiplexing of the data and address lines externally. Demultiplexing is explained in the later section. The bi-directional data bus is used to transfer data between DMAC, MPU, memory and I/O devices.

Address lines are outputs to address memory and I/O devices.

- **Address Bus (A_1 through A_7)**

Input/Output Active-high	Three-statable

In the MPU mode, the DMAC internal registers are accessed with these lines and $\overline{LDS}$, $\overline{UDS}$. The address map for these registers is shown in Table 1. During a DMA bus cycle, A_1-A_7 are outputs containing the low order address bits of the location being accessed.

- **Function Code (FC_0 through FC_2)**

Output Active-high	Three-statable

These output signals provide the function codes during DMA bus cycles. They are three-stated except in the DMA bus cycles. They are used to control the HMCS68000 memories.

- **Clock (CLK)**

Input

This is the input clock to the HD68450, and should never be terminated at any time. This clock can be different from the MPU clock since HD68450 operates completely asynchronously.

- **Chip Select ($\overline{CS}$)**

Input Active low

This input signal is used to chip select the DMAC in "MPU" mode. If the $\overline{CS}$ input is asserted during a bus cycle which is generated by the DMAC, the DMAC internally terminates the bus cycle and signals an address error. This function protects the DMAC from accessing its own register.

- **Address Strobe ($\overline{AS}$)**

Input/Output Active low	Three-statable

In the "MPU mode", this line is an input indicating valid address input, and during the DMA bus cycle it is an output indicating valid the address output from the DMAC on the address bus.

The DMAC monitors these input lines during bus arbitration to determine the completion of the bus cycle by the MPU or other bus masters.

- **Upper Address Strobe ($\overline{UAS}$)**

Output Active low	Three-statable

This line is an output to latch the upper address lines on the multiplexed data/address lines. It is three-stated except in the "DMA mode".

- **Own ($\overline{OWN}$)**

Output Active low	Three-statable

This line is asserted by the DMAC during DMA mode, and is used to control the output of the address line latch. This line may also be used to control the direction of bi-directional buffers when loads on $\overline{AS}$, $\overline{LDS}$, $\overline{UDS}$, R/$\overline{W}$ and other signals exceed the drive capability. It is three-stated in the "MPU mode" and the "IDLE mode"

- **Data Direction ($\overline{DDIR}$)**

Outputs	Three-statable
Active low	(when data direction is input to the DMAC)
Active high	(when the data direction is output from the DMAC)

This line controls the direction of data through the bidirectional buffer which used to demultiplex the data/address lines. It is three-stated during the "IDLE mode"

- **Data Bus Enable ($\overline{DBEN}$)**

Output	Three-statable
Active low	

This line controls the output enable line of bidirectional buffers on the multiplexed data/address lines. It is a three-stated during the "IDLE mode".

- **High Byte ($\overline{HIBYTE}$)**

Output	Three-statable
Active low	

This line is used when the operand size is byte in the single addressing mode. It is asserted when data is present on the upper eight bits of the data bus. It is used to control the output of bidirectional buffers which connects the upper eight bits of the data bus with the lower eight bits. It is three-stated during the "MPU mode" and the "IDLE mode"

- **Read/Write (R/$\overline{W}$)**

Input/Output	Three-statable
Active low (write)	
Active high (read)	

This line is an input in the "MPU mode" and an output during the "DMA mode". It is three-stated during the "IDLE mode". It is used to control the direction of data flow.

- **Upper Data Strobe ($\overline{UDS}$), Lower Data Strobe ($\overline{LDS}$)**

Input/Output	Three-statable
Active low	

These lines are extensions of the address lines indicating which byte or bytes of data of the addressed word are being addressed. These lines combined corresponds to address line A_0 in table 1.

- **Data Transfer Acknowledge ($\overline{DTACK}$)**

Input/Output	Three-statable
Active low	

In the "MPU mode", this line is an output indicating the completion of Read/Write bus cycle by the MPU.

In the "DMA mode", the DMAC monitors this line to determine when a data transfer has completed. In the event that a bus exception is requested, except for $\overline{HALT}$, prior to or concurrent with $\overline{DTACK}$, the $\overline{DTACK}$ response is ignored and the bus exception is honored. In the "IDLE mode", this signal is three-stated.

- **Bus Exception Controls ($\overline{BEC_0}$ through $\overline{BEC_2}$)**

Input	
Active low	

These lines provide an encoded signal input indicating an exceptional condition in the DMA bus cycle. See bus exception section for details.

- **Bus Request ($\overline{BR}$)**

Output	
Active low	

This output line is used to request ownership of the bus by the DMAC.

- **Bus Grant ($\overline{BG}$)**

Input	
Active low	

This line is used to indicate to the DMAC that it is to be the next bus master. The DMAC cannot assume bus ownership until both $\overline{AS}$ and $\overline{BGACK}$ becomes inactive. Once the DMAC acquires the bus, it does not continue to monitor the $\overline{BG}$ input.

- **Bus Grant Acknowledge ($\overline{BGACK}$)**

Input/Output	Three-statable
Active low	

Bus Grant Acknowledge ($\overline{BGACK}$) is a bidirectional control line. As an output, it is generated by the DMAC to indicate that it is the bus master.

As an input, $\overline{BGACK}$ is monitored by the DMAC, in limited rate auto-request mode, to determine whether or not the current bus master is a DMA device or not. $\overline{BGACK}$ is also monitored during bus arbitration in order to assume bus ownership.

- **Interrupt Request ($\overline{IRQ}$)**

Output	Open drain
Active low	

This line is used to request an interrupt to the MPU

- **Interrupt Acknowledge ($\overline{IACK}$)**

Input	
Active low	

This line is an input to the DMAC indicating that the current bus cycle is an interrupt acknowledge cycle by the MPU. The

DMAC responds the interrupt vector of the channel with the highest priority requesting an interrupt. There are two kinds of the interrupt vectors for each channel: normal (NIV) or error (EIV). $\overline{IACK}$ is not serviced if the DMAC has not generated $\overline{IRQ}$.

- **Channel Request ($\overline{REQ_0}$ through $\overline{REQ_3}$)**

Input	
Active low or falling edge	

These lines are the DMA transfer request inputs from the peripheral devices.

These lines are falling edge sensitive inputs when the request mode is cycle steal. They are low-level sensitive when the request mode is burst.

- **Channel Acknowledge ($\overline{ACK_0}$ through $\overline{ACK_3}$)**

Output	
Active low	

These lines indicate to the I/O device requesting a transfer that the request is acknowledged and the transfer is to be performed. These lines may be used as a part of the enable circuit for bus interface to the peripheral.

- **Peripheral Control Line ($\overline{PCL_0}$ through $\overline{PCL_3}$)**

Input/Output	Three-statable
Active low	

The four lines ($\overline{PCL_0} \sim \overline{PCL_3}$) are multi-purpose lines which may be individually programmed to be a START output, an Enable Clock input, a READY input, an ABORT input, a STATUS input, or an INTERRUPT input.

- **Done ($\overline{DONE}$)**

Input/Output	Open Drain
Active low	

As an output, this line is asserted concurrently with the $\overline{ACK_X}$ timing to indicate the last data transfer to the peripheral device. As an input, it allows the peripheral device to request a normal termination of the DMA transfer.

- **Device Transfer Complete ($\overline{DTC}$)**

Output	Three-statable
Active low	

This line is asserted when the DMA bus cycle has terminated normally with no exceptions. It may be used to supply the data latch timing to the peripheral device. In this case, data is valid at the falling edge of $\overline{DTC}$.

■ INTERNAL ORGANIZATION

The DMAC has four independent DMA channels. Each channel has its own set of channel registers. These registers define and control the activity of the DMAC in processing a channel operation.

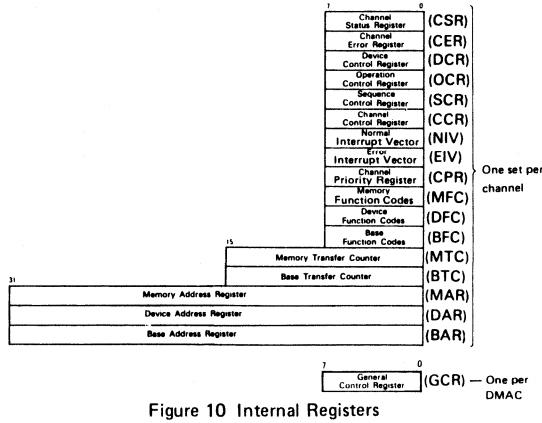


Figure 10 Internal Registers

- **Register Organization**

The internal register addresses are represented in Table 1. Address space not used within the address map is reserved for future expansion. A read from an unused location in the map results in a normal bus cycle with all ones for data. A write to one of these locations results in a normal bus cycle but no write occurs.

Unused bits of the defined registers in Table 1 read as zeros.

Table 1 Internal Register Addressing Assignments

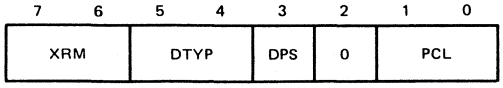
	Address Bits								
Register	7	6	5	4	3	2	1	0	Mode
Channel Status Register	c	c	0	0	0	0	0	0	R/W*
Channel Error Register	c	c	0	0	0	0	0	1	R
Device Control Register	c	c	0	0	0	1	0	0	R/W
Operation Control Register	c	c	0	0	0	1	0	1	R/W
Sequence Control Register	c	c	0	0	0	1	1	0	R/W
Channel Control Register	c	c	0	0	0	1	1	1	R/W
Memory Transfer Counter	c	c	0	0	1	0	1	b	R/W
Memory Address Register	c	c	0	0	1	1	s	s	R/W
Device Address Register	c	c	0	1	0	1	s	s	R/W
Base Transfer Counter	c	c	0	1	1	0	1	b	R/W
Base Address Register	c	c	0	1	1	1	s	s	R/W
Normal Interrupt Vector	c	c	1	0	0	1	0	1	R/W
Error Interrupt Vector	c	c	1	0	0	1	1	1	R/W
Channel Priority Register	c	c	1	0	1	1	0	1	R/W
Memory Function Codes	c	c	1	0	1	0	0	1	R/W
Device Function Codes	c	c	1	1	0	0	0	1	R/W
Base Function Codes	c	c	1	1	1	0	0	1	R/W
General Control Register	1	1	1	1	1	1	1	1	R/W

cc:00-Channel #0,01-Channel #1,
10-Channel #2,11-Channel #3,
ss:00-high-order, 01-upper middle,
10-lower middle,11-low-order
b: 0-high-order, 1-low-order
* see Channel Status Register Section

- **Device Control Register (DCR)**

The DCR is a device oriented control register. The XRM bits specifies whether the channel is in burst or cycle steal request mode. The DTYP bits define what type of device is on the channel. If the DTYP bits are programmed to be a HMCS6800 device, the PCL definition is ignored and the $\overline{PCL}$ line is an Enable clock input. If the DTYP bits are programmed to be a device with $\overline{READY}$, the PCL definition is ignored and the $\overline{PCL}$ line is a $\overline{READY}$ input. The DPS bit defines the port size (eight or sixteen bits) of the peripheral device. (A port size is the largest data which the peripheral device can transfer during a DMA bus cycle.) The PCL bits define the function of the $\overline{PCL}$ line. If the DTYP bits are programmed to be HMCS6800 device, or Device with $\overline{ACK}$ and $\overline{READY}$, these definitions are ignored. The XRM

bits are ignored if an auto-request mode (REQG = 00 or 01 in Operation Control Register) is selected.



XRM (EXTERNAL REQUEST MODE)
- 00 Burst Transfer Mode
- 01 (undefined, reserved)
- 10 Cycle Steal Mode without Hold
- 11 Cycle Steal Mode with Hold

DTYP (DEVICE TYPE)
- 00 HD68000 compatible device, explicitly addressed (dual addressing mode)
- 01 HD6800 compatible device, explicitly addressed (dual addressing mode)
- 10 Device with $\overline{ACK}$, implicitly addressed (single addressing mode)
- 11 Device with $\overline{ACK}$ and $\overline{READY}$, implicitly addressed (single addressing mode)

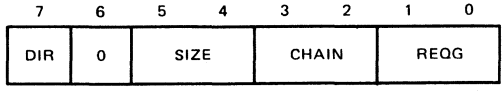
DPS (DEVICE PORT SIZE)
- 0 8 bit port
- 1 16 bit port

PCL (PERIPHERAL CONTROL LINE)
- 00 Status Input
- 01 Status Input with Interrupt
- 10 Start Pulse
- 11 Abort Input

Bit 2 Not Used

- **Operation Control Register (OCR)**

The OCR is an operation control register. The DIR bit defines the direction of the transfer. The SIZE bits define the size of the operand. The CHAIN bits define the type of the CHAIN mode. The REQG bits define how requests for transfers are generated.



DIR (DIRECTION)
- 0 Transfer from memory to device (transfer from MAR address to DAR address)
- 1 Transfer from device to memory (transfer from DAR address to MAR address)

SIZE (OPERAND SIZE)
- 00 Byte (8 bits)
- 01 Word (16 bits)
- 10 Long Word (32 bits)
- 11 (undefined, reserved)

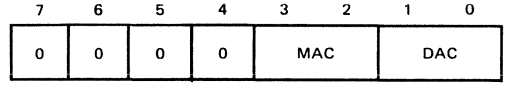
CHAIN (CHAINING OPERATION)
- 00 Chain operation is disabled
- 01 (undefined, reserved)
- 10 Array Chaining
- 11 Linked Array Chaining

REQG (DMA REQUEST GENERATION METHOD)
- 00 Auto-request at transfer rate limited by General Control Register (Limited Rate Auto-Request)
- 01 Auto-request at maximum rate
- 10 $\overline{REQ}$ line requests an operand transfer
- 11 Auto-request the first operand, external request for subsequent operands

Bit 6 Not Used

- **Sequence Control Register (SCR)**

The SCR is used to define the sequencing of memory and device addresses.



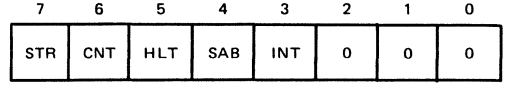
MAC (MEMORY ADDRESS COUNT)
- 00 Memory address register does not count
- 01 Memory address register counts up
- 10 Memory address register counts down
- 11 (undefined, reserved)

DAC (DEVICE ADDRESS COUNT)
- 00 Device address register does not count
- 01 Device address register counts up
- 10 Device address register counts down
- 11 (undefined, reserved)

Bits 7, 6, 5, 4 Not Used

- **Channel Control Register (CCR)**

The CCR is used to start or terminate the operation of a channel. This register also determines if an interrupt request is to be generated. Setting the STR bit causes immediate activation of the channel; the channel will be ready to accept request immediately. The STR and CNT bits of the register cannot be reset by a write to the register. The SAB bit is used to terminate the operation forcedly. Setting the SAB bit will reset STR and CNT. Setting the HLT bit will halt the channel operation, and clearing the HLT bit will resume the operation. Setting start bit must be done by byte access. Otherwise, timing error occurs.



STR (START OPERATION)
- 0 No operation is pending
- 1 Start operation

CNT (CONTINUE OPERATION)
- 0 No continuation is pending
- 1 Continue operation

HLT (HALT OPERATION)
- 0 Operation not halted
- 1 Operation halted

SAB (SOFTWARE ABORT)
- 0 Channel operation not aborted
- 1 Abort channel operation

INT (INTERRUPT ENABLE)
- 0 No interrupts enabled
- 1 Interrupts enabled

Bits 2, 1, 0 Not Used

- **Channel Status Register (CSR)**

The CSR is a register containing the status of the channel.

7	6	5	4	3	2	1	0
COC	BTC	NDT	ERR	ACT	0	PCT	PCS

COC (CHANNEL OPERATION COMPLETE)
0 Channel operation incomplete
1 Channel operation complete
BTC (BLOCK TRANSFER COMPLETE)
0 Block transfer incomplete
1 Block transfer complete
NDT (NORMAL DEVICE TERMINATION)
0 No normal device termination by $\overline{\text{DONE}}$ input
1 Device terminated operation normally by $\overline{\text{DONE}}$ input
ERR (ERROR BIT)
0 No errors
1 Error as coded in CER
ACT (CHANNEL ACTIVE)
0 Channel not active
1 Channel active
PCT ($\overline{\text{PCL}}$ TRANSITION)
0 No $\overline{\text{PCL}}$ transition occurred
1 $\overline{\text{PCL}}$ transition occurred
PCS (THE STATE OF THE $\overline{\text{PCL}}$ INPUT LINE)
0 $\overline{\text{PCL}}$ "Low"
1 $\overline{\text{PCL}}$ "High"
Bit 2 Not Used

- **Channel Error Register (CER)**

The CER is an error condition status register. The ERR bit of CSR indicates if there is an error or not. Bits 0-4 indicate what type of error occurred.

7	6	5	4	3	2	1	0
0	0	0	ERROR CODE				

Error Code
00000 No error
00001 Configuration error
00010 Operation timing error
00101 Address error in MAR
00110 Address error in DAR
00111 Address error in BAR
01001 Bus error in MAR
01010 Bus error in DAR
01011 Bus error in BAR
01101 Count error in MTC
01111 Count error in BTC
10000 External abort
10001 Software abort
Bits 7, 6, 5 Not Used

- **Channel Priority Register (CPR)**

The CPR is used to define the priority level of the channel. Priority level 0 is the highest and priority level 3 is the lowest priority.

7	6	5	4	3	2	1	0
0	0	0	0	0	0	CP	

CP (CHANNEL PRIORITY)
00 Priority level 0
01 Priority level 1
10 Priority level 2
11 Priority level 3
Bit 7 through 2 Not Used

- **General Control Register (GCR)**

The GCR is used to define what portion of the bus cycles is available to the DMAC for limited rate auto-request generation. GCR is also used to specify the hold time for cycle steal mode with hold.

7	6	5	4	3	2	1	0
0	0	0	0	BT		BR	

BT (BURST TIME)
The number of DMA clock cycles per burst that the DMAC allows in the auto-request at a limited rate of transfer is controlled by these two bits. The number is 2(BT+4) (two to the BT+4 power).
BR (BANDWIDTH RATIO)
The amount of the bandwidth utilized by the auto-request at a limited rate transfer is controlled by these two bits. The ratio is 2(BR+1) (two to the BR+1 power).
The hold time for cycle steal mode with hold is defined to be minimum of 1 sample interval and maximum of 2 sample intervals. A sample interval is defined to be 2(BT+BR+5) (two to the BT+BR+5 power) clock cycles.

Bits 7 through 4 Not Used

- **Address Registers (MAR, DAR, BAR)**

Three 32-bit registers are utilized to implement the Memory Address Register, Device Address Register, and the Base Address Register. Only the least significant twenty-four bits are connected to the address output pins. The content of the MAR is outputted when the memory is accessed in single or dual adressing mode. The content of the DAR is outputted when the peripheral device is accessed. The contents of the BAR is outputted when reading chain information from memory in the Array Chaining Mode or the Linked Array Chaining Mode. It is also used to set the top address of the next block transfer in Continue mode.

- **Function Code Registers (MFC, DFC, BFC)**

The DMAC has three function code registers per channel: the Memory Function Code Register (MFC), Device Function Code Register (DFC), and the Base Function Code Register (BFC). The contents of these registers are outputted from FC_0 through FC_2 lines when an address is outputted from MAR, DAR, or BAR, respectively. The BFC is also used to set the MFC for the transfer of the next data block in the Continue mode.

7	6	5	4	3	2	1	0
0	0	0	0	0	FC2	FC1	FC0

Bits 3 through 7 Not Used

- **Transfer Count Registers (MTC, BTC)**

Each channel has two 16-bit counters: the Memory Transfer Counter (MTC) and the Base Transfer Counter (BTC). The MTC

counts the number of transfer words in one block, and is decreased by one for every operand transfer.

The BTC is used to count the number of data blocks in the Array Chaining Mode. BTC is also used to set the number of operands to transfer for the next data block in the Continue Mode.

- **Interrupt Vector Registers (NIV, EIV)**

Each channel has a Normal Interrupt Vector register and an Error Interrupt Vector register.

When an interrupt acknowledge cycle occurs, an interrupt vector is outputted from one of those registers. If the error bit (CSR) is set for the channel with interrupt pending, then content of EIV is outputted, otherwise content of NIV is outputted.

■ OPERATION DESCRIPTION

A DMAC channel operation proceeds in three principal phases. During the initialization phase, the MPU sets the channel control registers, supply the initial address and the number of transfer words, and starts the channel. During the transfer phase, the DMAC accepts requests for data operand transfers, and provides addressing and bus controls for the transfers. The termination phase occurs after the operation is completed.

This section describes DMAC operations. A description of the MPU/DMAC communication is given first. Next, the transfer phase is covered, including how the DMAC recognizes requests and how the DMAC arranges for data transfer. Following this, the initialization phase is described. The termination phase is covered, introducing chaining, error signaling, and bus exceptions. A description of the channel priority scheme rounds out the section.

- **Read/Write of the DMAC Registers by MPU**

The MPU reads and writes the DMAC internal registers and controls the DMA transfer. Figure 11 indicates the timing diagram when the MPU reads the contents of the DMAC register. The MPU outputs A_1-A_{23}, FC_0-FC_2, $\overline{AS}$, R/$\overline{W}$, $\overline{UDS}$, and $\overline{LDS}$, and accesses the DMAC internal register. The specific internal register is selected by A_1-A_7, $\overline{LDS}$ and $\overline{UDS}$. The $\overline{CS}$ and $\overline{IACK}$ lines are generated by the external circuit with A_8-A_{23} and FC_0-FC_2. The DMAC outputs data on the data bus, together with $\overline{DDIR}$, $\overline{DBEN}$ and $\overline{DTACK}$. The $\overline{DDIR}$ and $\overline{DBEN}$ control the bidirectional buffer on the bus and the $\overline{DTACK}$ indicates that the data has been sent or received by the DMAC. Read Cycle is eighteen CLKs. Figure 12 shows the MPU write cycle. Write cycle is fifteen CLKs.

Note the following points.

(1) The clock reference shown in this figure is the DMAC input clock.
(2) The $\overline{DDIR}$ and the $\overline{DBEN}$ are three-stated at the beginning which detects $\overline{CS}$ and the ending of the cycle.
(3) During the MPU read cycle, the $\overline{DTACK}$ is asserted after the data is valid on the system bus.
(4) During the MPU write cycle, the $\overline{DDIR}$ line will be driven low to direct the data buffers toward to DMAC before the buffers are enabled.
(5) During the MPU write cycle, the DMAC will latch the data before asserting $\overline{DTACK}$. Then it will negate $\overline{DBEN}$ and $\overline{DDIR}$ in the proper order.
(6) After the MPU cycle and the $\overline{LDS}$ and the $\overline{UDS}$ are negated by the MPU, the DMAC will put $\overline{DBEN}$, $\overline{DDIR}$ and the address data lines to a high impedance state.
(7) $\overline{DTACK}$ will once go "High" and then to a high impedance state after negating $\overline{LDS}$ and $\overline{UDS}$.

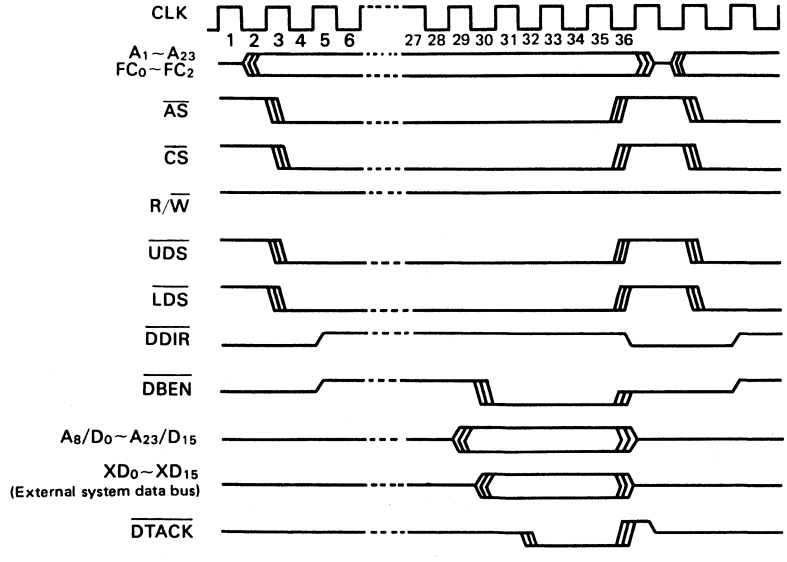


Figure 11 MPU Read from DMAC – Word

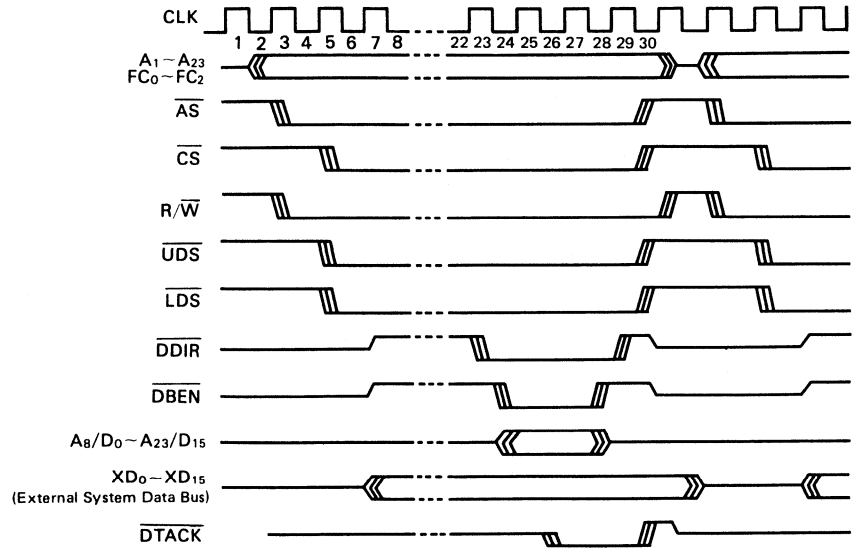
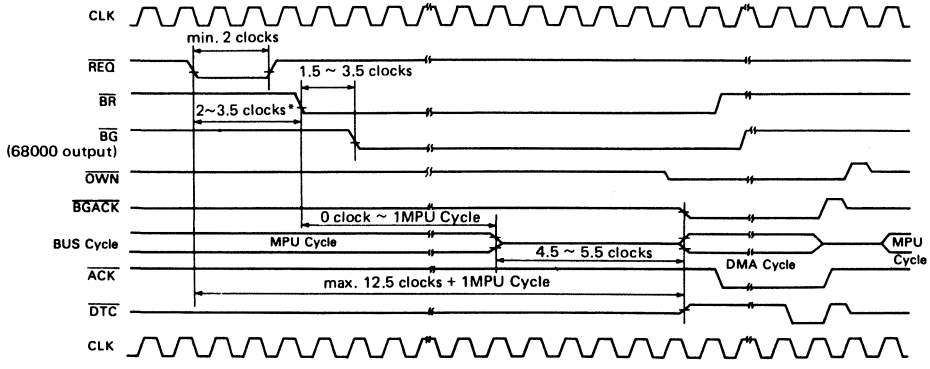


Figure 12 MPU Write to DMAC – Word

- **Bus Arbitration**

The followings are the description of the bus arbitration. The DMAC must obtain the ownership of the bus in order to transfer data. Figure 13 indicates the DMAC bus arbitration timing. It is completely compatible with that of HD68000 MPU. The DMAC asserts the Bus Grant ($\overline{BG}$) to request the bus mastership. The MPU recognizes the request and asserts $\overline{BG}$, then it grants the ownership in the next bus cycle. After the end of the current cycle ($\overline{AS}$ is negated), the MPU relinquishes the bus to the DMAC. The DMAC asserts the bus grant acknowledge ($\overline{BGACK}$) to indicate that it has the bus ownership. A half clock before $\overline{BGACK}$ is asserted, the DMAC asserts $\overline{OWN}$. $\overline{OWN}$ is kept asserted for a half clock after $\overline{BGACK}$ is negated at the end of the DMA cycle. $\overline{BR}$ is negated one clock after $\overline{BGACK}$ is asserted.



* This case assumes that no exception condition exists and DMAC isn't accessed by MPU.

Figure 13 DMAC Bus Arbitration Timing

- **Device/DMAC Communication**

Communication between peripheral devices and the DMAC is accomodated by five signal lines. Each channel has $\overline{REQ}$, $\overline{ACK}$ and $\overline{PCL}$, and the last two lines the $\overline{DONE}$ and $\overline{DTC}$ lines, are shared among the four channels.

(1) Request ($\overline{REQ}$)

The peripheral devices assert $\overline{REQ}$ to request data transfers. See the "Requests" section for details.

(2) Acknowledge ($\overline{ACK}$)

This line is used to implicitly address the device which is transferring the data (This device is not selected by address lines.) It is also asserted when the content of DAR is outputted during memory-to-memory transfer except for the auto-request mode at a limited rate or at the maximum rate.

(3) Peripheral Control Line ($\overline{PCL}$)

The function of this line is quite flexible and is determined by the DCR (Device Control Register).

The DTYP bits of the DCR define what type of device is on the channel. If the DTYP bits are programmed to be a HMCS6800 device, the PCL definition is ignored and the $\overline{PCL}$ line is an Enable clock (E clock) input. If the DTYP bits are programmed to be a device with $\overline{READY}$, the PCL definition as ignored and the $\overline{PCL}$ line is a ready input.

$\overline{PCL}$ As a Status Input

The $\overline{PCL}$ line may be programmed as a status input. The status level of this line can be determined by the PCS bit in the CSR, regardless of the PCL function determined by the DCR. If a negative transition occurs and remains stable for a minimum of two clocks, the PCT bit of the CSR is set. This PCT bit is cleared by resetting the DMAC or the writing "1" to the PCT bit.

$\overline{PCL}$ As an Interrupt

The $\overline{PCL}$ line may be programmed to generate an interrupt on a negative transition. This enables an interrupt which is requested if the PCT bit of the CSR is set. When using this function, it is necessary to reset the PCT bit in the CSR before the PCL bit in the DCR is set to interrupt, in order to avoid assertion of IRQ line at this time.

$\overline{PCL}$ As a Starting Pulse

The $\overline{PCL}$ line may be programmed to output a starting pulse. This active low starting pulse is outputted when a channel is activated, and is "Low" for a period of four clock cycles.

$\overline{PCL}$ As an Abort Input

The $\overline{PCL}$ line may be programmed to be a negative transition abort input which terminates an operation by setting the external abort error in CER. It is necessary to reset the PCT bit in the CSR before activating the channel (Setting the ACT bit of CCR) so that the channel operation is not immediately aborted.

$\overline{PCL}$ As an Enable Clock (E Clock) Input

If the DTYP bits are programmed to be a HMCS6800 device, the PCL definition is ignored and the $\overline{PCL}$ line is an Enable Clock input. The Enable clock downtime must be as long as five clock cycles, and must be high for a minimum of three DMAC clock cycles, but need not be synchronous with the DMAC's clock.

$\overline{PCL}$ As a $\overline{READY}$ Input

If the DTYP bits are programmed to be a device with $\overline{READY}$, the PCL definition is ignored and the $\overline{PCL}$ line is a $\overline{READY}$ input. The $\overline{READY}$ is an active low input.

(4) DONE ($\overline{DONE}$)

This line is an active low Input/Output signal with an open drain. It is asserted when the memory transfer count is exhausted in a single block transfer. In the chaining operation, $\overline{DONE}$ is asserted only at the last transfer to the peripheral device of the last data block. In the continue mode, $\overline{DONE}$ is asserted for each data block. It is asserted and negated in coincident with the $\overline{ACK}$ line for the last data transfer to the peripheral device. It is also outputted in coincident with the $\overline{ACK}$ line of the last bus cycle, in which the address is outputted from the DAR, in the memory-to-memory transfer (dual addressing mode) that uses the $\overline{ACK}$ line.

The DMAC also monitors the state of the $\overline{DONE}$ line during the DMA bus cycle. If the device asserts $\overline{DONE}$ during $\overline{ACK}$ active, the DMAC will terminate the operation after the transfer of the current operand. If $\overline{DONE}$ is asserted on the first byte of 2 byte operation or the first word of long word operation, the DMAC does not terminate the operation before the whole operand transfer is completed. If $\overline{DONE}$ is asserted, then the DMAC terminates the operation by clearing the ACT bit of the CSR, and setting the COC and NDT bits of the CSR. If both the DMAC and the device assert $\overline{DONE}$, the device termination is not recognized, but the channel operation does terminate. $\overline{DONE}$ is outputted again for the retry exceptions bus cycles.

(5) Data Transfer Complete ($\overline{DTC}$)

$\overline{DTC}$ is an active low signal which is asserted when the actual data transfer is accomplished. It is also asserted in the bus cycle which read a chain information from memory in the Chaining mode. However, if exceptions are generated and the DMA bus cycle terminates, $\overline{DTC}$ is not asserted. $\overline{DTC}$ is asserted one half clock before $\overline{LDS}$ and $\overline{UDS}$ are negated, and negated one half clock after $\overline{LDS}$ and $\overline{UDS}$ are negated.

- **Requests**

Requests may be externally generated by circuitry in the peripheral device, or internally generated by the auto-request mechanism. The REQG bits of the OCR determine these modes. The DMAC also supports an operation in which the DMAC auto-requests the first transfer and then wait for the peripheral device to request the following transfers.

(1) Auto-request Transfers

The auto-request mechanism provides generation of requests within the DMAC. These requests can be generated at either of two rates: maximum-rate and limited-rate. In the former case, the channel always has a request pending.

The limited rate auto-request functions by monitoring the bus utilization.

Limited-rate Auto-request

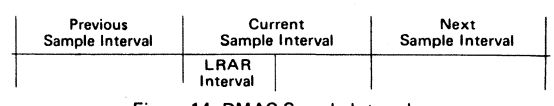


Figure 14 DMAC Sample Intervals

In the limited-rate auto-request the DMAC devides time into equal length sample intervals by counting clock cycles. The end of one sample interval makes the beginning of the next. During a sample interval, the DMAC monitors by means of $\overline{BGACK}$ pin the system bus activity of the DMAC and other bus master devices. At the end of the sample interval, decision is made whether or not to perform the channel's data transfer during the next sample interval. Namely, based on the activity of the DMAC or other bus master devices during the current sample interval, the DMAC allows limited-rate auto-requests for some initial portion of the next sample interval.

The length of the sample interval, and the portion of the sample interval during which limited-rate auto-requests can be

made (the limited-rate auto-request interval) are controlled by the BT and BR bits in the GCR. The length in clock cycles of the limited-rate auto-request interval is $2^{(BT+4)}$ (2 raised to the BT+4 power). For example, if BT equals 2 and the DMA utilization of the bus was low during the previous sample interval, then the DMAC generates the auto-request transfers during the first 64 clock cycles.

The ratio of the length of the sample interval to the length of the limited-rate auto-request interval is controlled by the BR bits. The ratio of the system bus utilization of the MPU to other bus master devices including he DMAC is $2^{(BR+1)}$ (2 raised to the BR+1 power). If the fraction of DMA clock cycles during the sample interval exceeds the programmed utilization level, the DMAC will not allow limited-rate auto-requests during the next sample interval.

For example, if BR equals 3, then at most one out of 16 clock cycles during a sample interval can be used by the DMAC and other bus master devices, and still the DMAC would allow limited rate auto-request during the next sample interval. Therefore, from the viewpoint of long period, the ratio of the system bus utilization of the MPU to I/O devices including the DMAC is about 16:1. The sample interval length is not a direct parameter, but it is equal to $2^{(BT+BR+5)}$ clock cycles. Thus, the sample interval can be programmed between 32 and 2048 clock cycles.

The DMAC uses the $\overline{\text{BGACK}}$ to differentiate between the MPU bus cycle and DMAC or other bus master devices. If $\overline{\text{BGACK}}$ is active, then the DMAC assumes that the bus is used by a DMAC or other bus master devices. If it is inactive, then the DMAC assumes that it is used by the MPU.

Maximum-rate Auto-request

If the REQG bits in the OCR indicate auto-request at the maximum rate, the DMAC acquires the bus after the start bit is set and keeps it until the data transfer is completed.

If a request is made by another channel of higher priority, the DMAC services that channel and then resumes the auto-request sequence. If two or more channels are set to equal priority level and maximum rate auto-request, then the channels will rotate in a "round robbin" fashion.

If the HMCS68000 compatible device is connected to a channel, the $\overline{\text{ACK}}$ line is held inactive during an auto-request operation. Consequently, any channel may be used for the memory-to-memory transfer with the auto-request function in addition to the operation of data transfer between memory and peripheral device with using the $\overline{\text{REQ}}$ pin. Refer to Figure 15 for the timing of the memory-to-memory transfer. In this mode, the $\overline{\text{ACK}}$, $\overline{\text{HIBYTE}}$ and $\overline{\text{DONE}}$ outputs are always inactive.

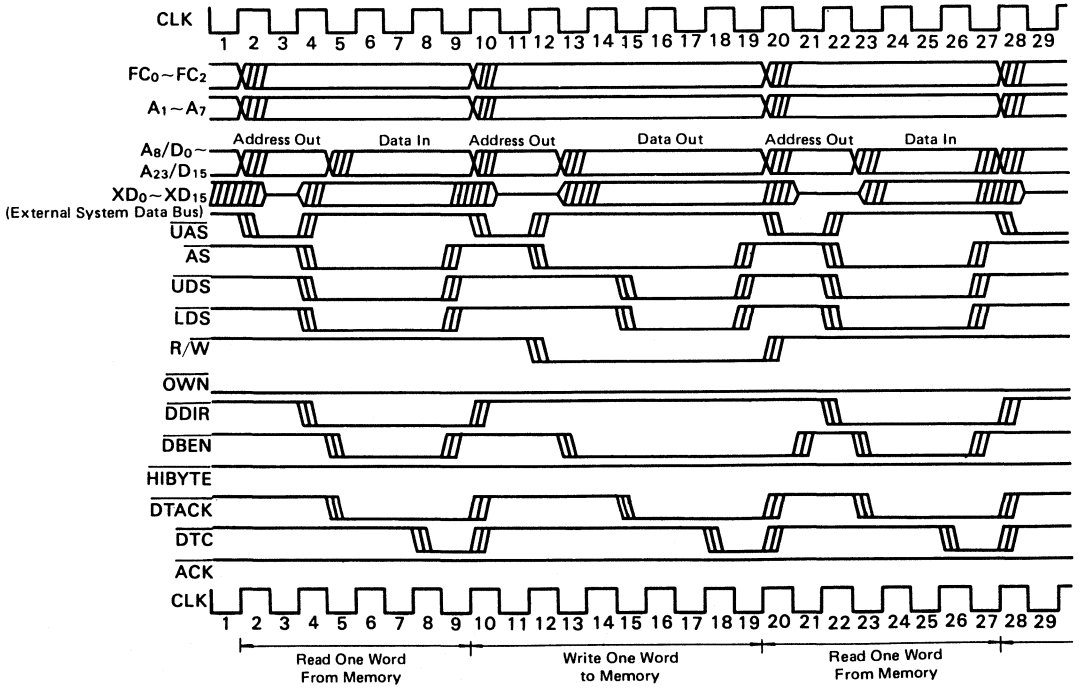


Figure 15 Memory-to-Memory Transfer Read-Write-Read Cycles

(2) External Requests

If the REQG bits of the OCR indicate that the $\overline{REQ}$ line generates requests, the transfer requests are generated externally. The request line associated with each channel allows the device to externally generate requests for DMA transfers. When the device wants an operand transferred, it makes a request by asserting the request line. The external request mode is determined by the XRM bits of the DCR, which allows both burst and cycle steal request modes. The burst request mode allows a channel to request the transfer of multiple operands using consecutive bus cycles. The cycle steal request mode allows a channel to request the transfer of a single operand. The followings are the description of the burst and the cycle steal modes.

Burst Request Recognition

In the burst request mode, the $\overline{REQ}$ line is an active low input. The level sampled at the rising edge of the clock. Once the burst request is asserted, it needs to be held low until the first DMA bus cycle starts in order to insure at least one data transfer operation. In order to stop the burst mode transfer after the current bus cycle, the $\overline{REQ}$ line has to be negated one clock before the $\overline{DTC}$ output clock of this cycle. Refer to Figure 16 or the burst mode timing.

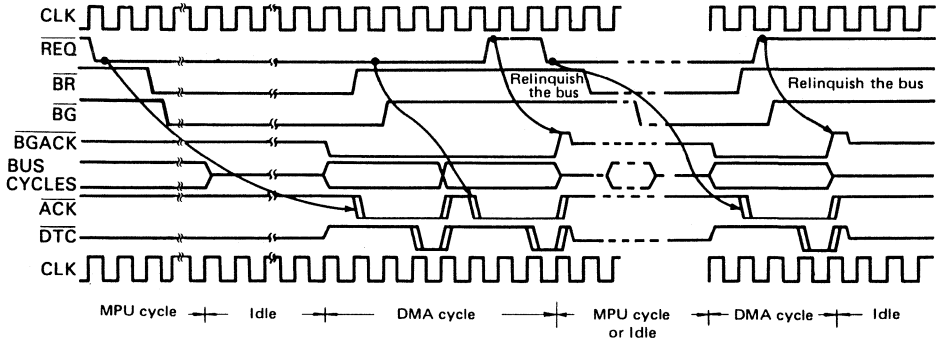


Figure 16 Burst Mode Request Timing
(Only one channel is active)

Cycle Steal Request Recognition

In the cycle steal request mode, the peripheral device requests the DMA transfer by generating an falling edge at the $\overline{REQ}$ line. The $\overline{REQ}$ line needs to be held "low" for at least 2 clock cycles. In the cycle steal mode, if the $\overline{REQ}$ line changes from "High" to "Low" between $\overline{ACK}$ output and one clock before the clock that outputs $\overline{DTC}$, then the next DMA transfer is performed without relinquishing the bus. If the bus is not relinquished, then maximum of 5 idle clocks is inserted between bus cycles. Refer to Figure 17 for the request timing of the cycle steal mode. If the XRM bits specify cycle steal without hold, the DMAC will relinquish the bus. If the XRM bits specify cycle steal with hold, the DMAC will retain ownership. The bus is not given up for arbitration until the channel operation terminates or until the device pauses. The device is determined to have paused if it does not make any requests during the next full sample interval. The sample interval counter is free running and is not reset or modified by this mode of operation. The sample interval counter is the same counter that is used for Limited Rate Auto Request and is programmed via the GCR. Figure 18 shows the request timing in the cycle steal bus hold. If the $\overline{REQ}$ is inputted during the hold time, the $\overline{ACK}$ is outputted after a maximum of 7.5 clock cycles from the picked-up clock. On the cycle steal with hold mode, the DMAC will hold the bus even when the transfer count is exhausted and the last data has been transferred. If DMA transfer is requested from other channels during this period, they are executed normally.

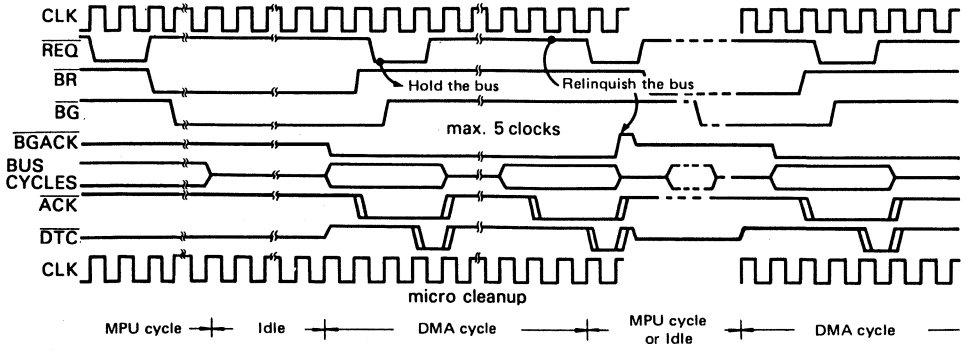


Figure 17 Cycle Steal Mode Request Timing

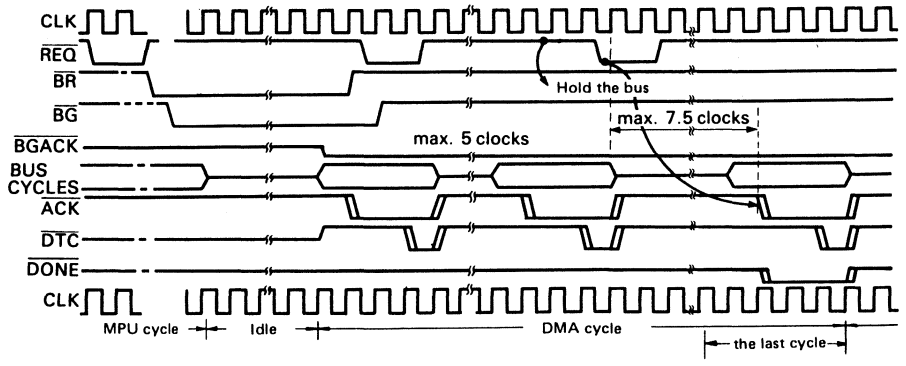


Figure 18 Cycle Steal Bus Hold Mode Request Timing

Request Recognition in Dual-address Transfers

In a following section dual-address transfers are defined. Dual address transfer is an exception to the request recognition rules in the previous paragraphs. In the cycle steal request mode, when there are two or more than transfers between the DMAC and the peripheral device during one operand transfer, the request is not recognized until the last transfer between the DMAC and the I/O device starts.

(3) Mixed Request Generation

A single channel can mix the two request generation methods. By programming the REQG bits of the OCR to "11", when the channel is started, the DMAC auto-requests the first transfer. Subsequent requests are then generated externally by the device. The $\overline{ACK}$ and $\overline{PCL}$ lines perform their normal functions in this operation.

- **Data Transfers**

All DMAC data transfers are assumed to be between memory and the peripheral device. The word "memory" means a 16-bit HMCS68000 bus compatible device. By programming the DCR, the characteristics of the peripheral device may be assigned. Each channel can communicate using any of the following protocols.

DTYP	Device Type	
00	HMCS68000 compatible device	Dual Addressing
01	HMCS6800 compatible device	
10	Device with $\overline{ACK}$	Single Addressing
11	Device with $\overline{ACK}$ and $\overline{READY}$	

(1) Dual Addressing

HMCS68000 and HMCS6800 compatible devices may be explicitly addressed. This means that before the peripheral transfers data, a data register within the device must be addressed. Because the address bus is used to address the peripheral, the data cannot be directly transferred to/from the memory because the memory also requires addressing. Instead, the data is transferred from the source to the DMAC and held in an internal DMAC holding register. A second bus transfer between the DMAC and the destination is then required to complete the operation. Because both the source and destination of the transfer are explicitly addressed, this protocol is called dual-addressed.

HMCS68000 Compatible Device Transfers

In this operation, when a request is received, the bus is obtained and the transfer is completed using the protocol as shown in Figures 19 and 20. Figures 21 through 24 show the transfer timings. Figure 21 and 24 show the operation when the memory is the source and the peripheral device is the destination. Figures 22 and 23 show the transfer in the opposite direction. The peripheral device is a 16-bit device in Figures 21 and 22, and a 8-bit device in Figures 23 and 24.

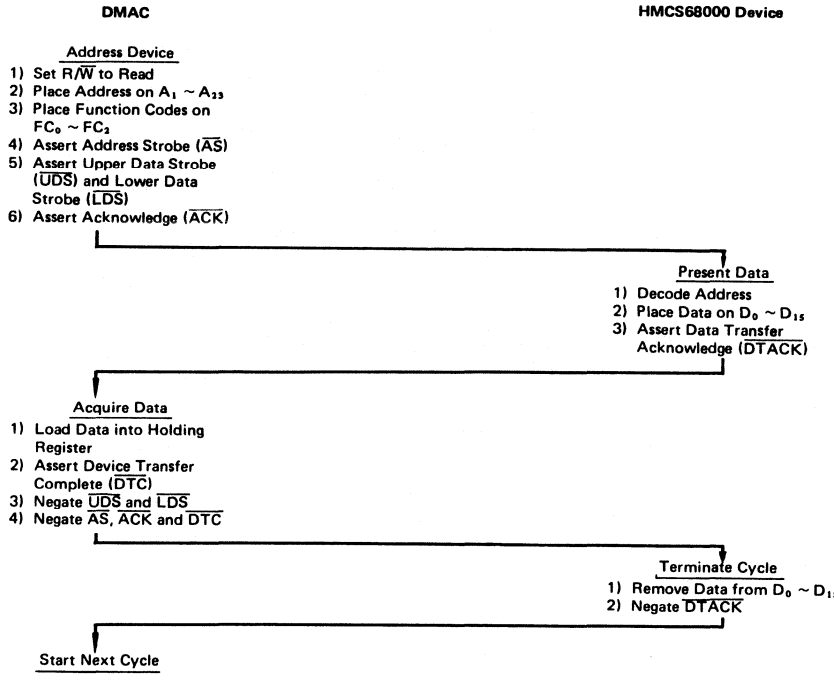


Figure 19 Word Read Cycle Flowchart HMCS68000 Type Device

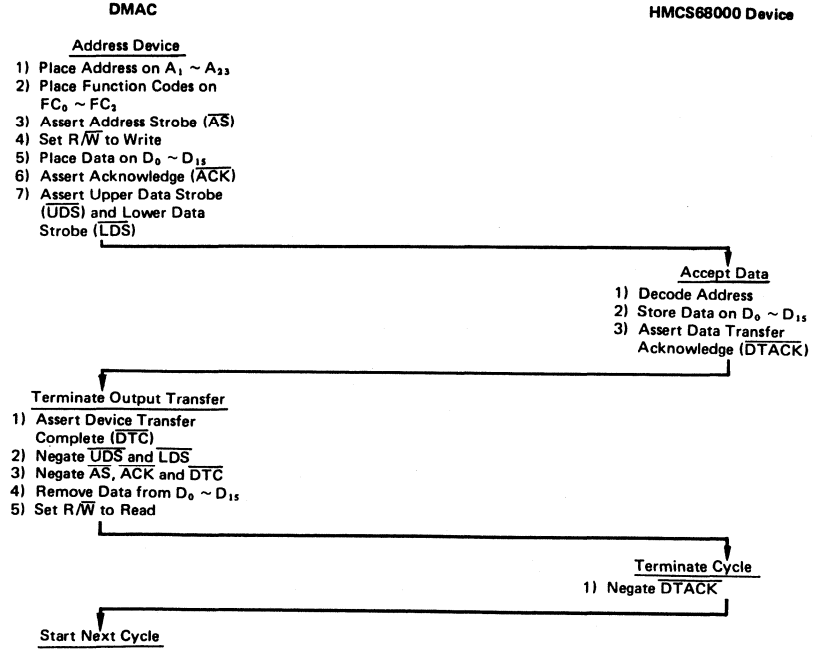


Figure 20 Word Write Cycle Flowchart HMCS68000 Type Device

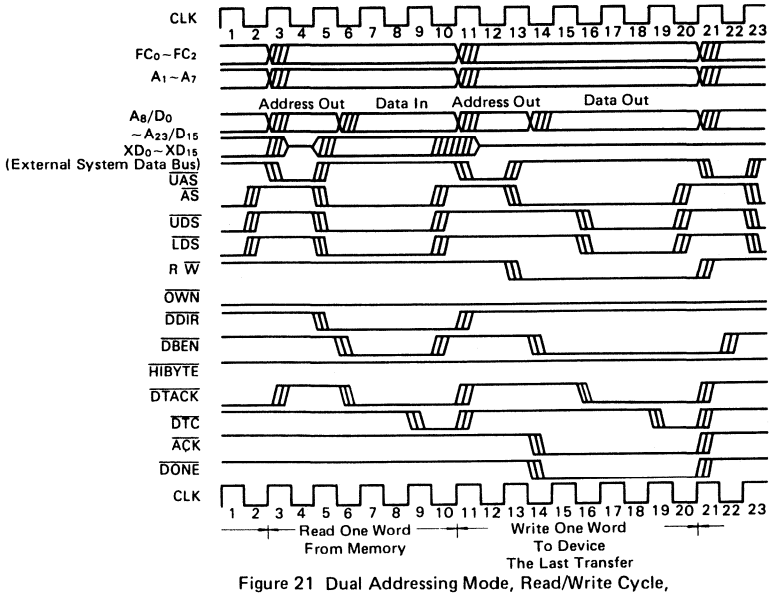


Figure 21 Dual Addressing Mode, Read/Write Cycle, Destination = 16-bit Device, Word Operand

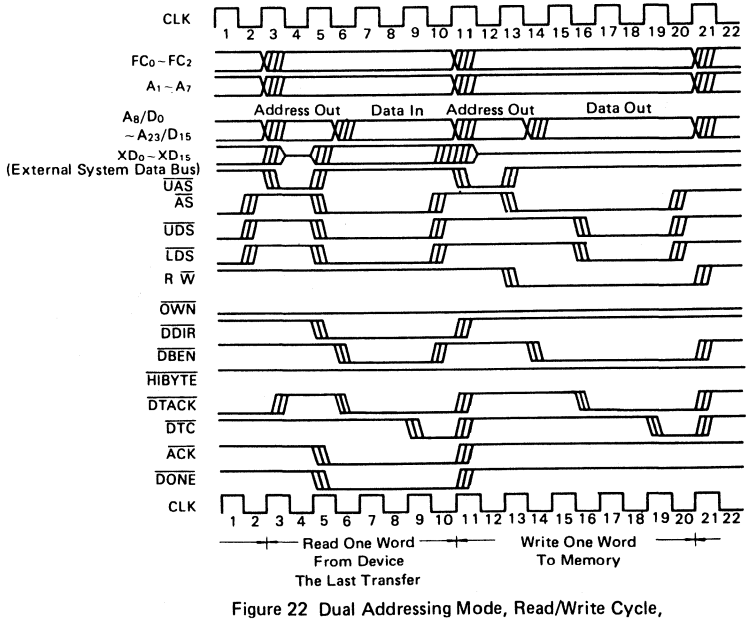


Figure 22 Dual Addressing Mode, Read/Write Cycle, Source = 16-bit Device, Word Operand

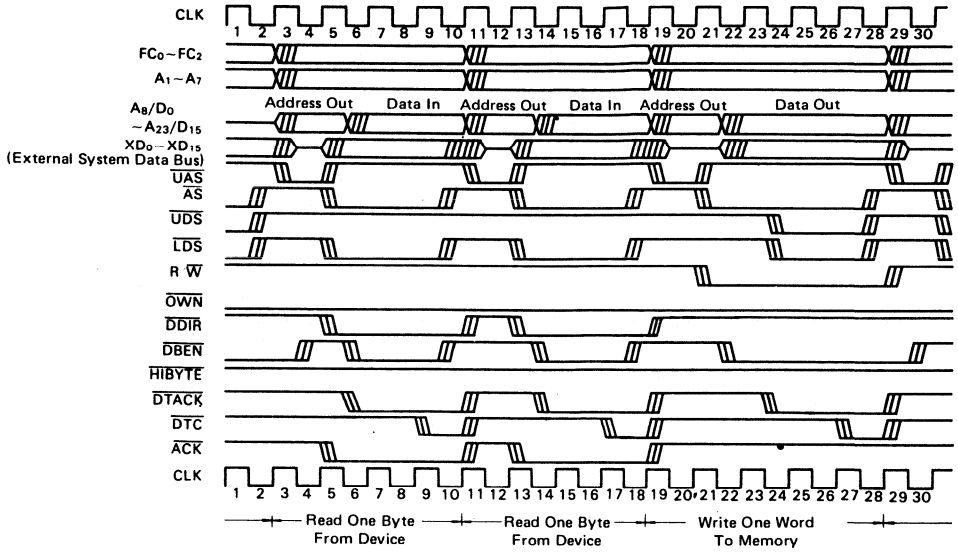


Figure 23 Dual Addressing Mode, Read/Write Cycle
Source = 8-bit Device, Word Operand

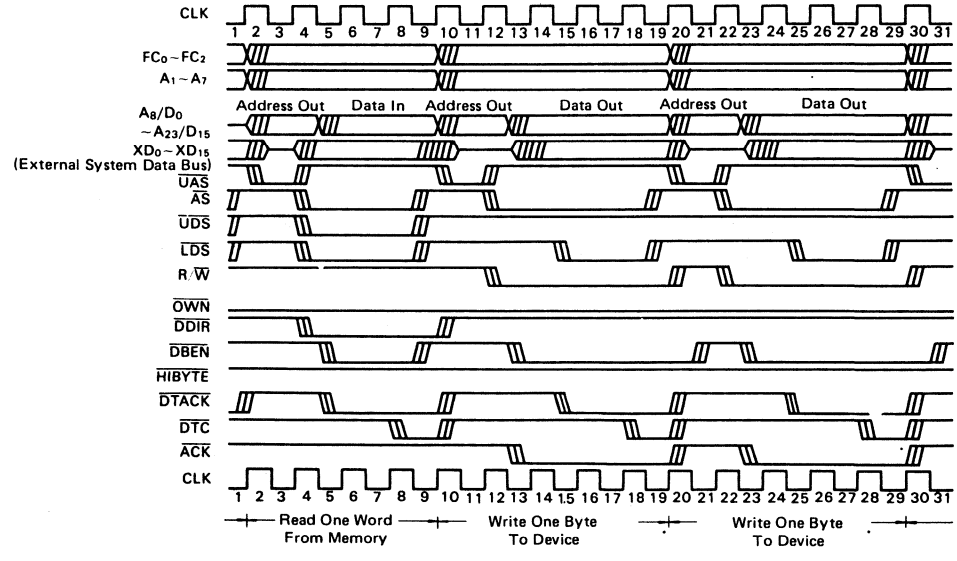


Figure 24 Dual Addressing Mode, Read/Write Cycle,
Destination = 8-bit Device, Word Operand

HMCS6800 Compatible Device Transfers

When a channel is programmed to perform HMCS6800 compatible transfers, the $\overline{PCL}$ line for that channel is defined as an Enable clock input. The DMAC performs data transfers between itself and the peripheral device using the HMCS6800 bus protocol, with the $\overline{ACK}$ output providing the $\overline{VMA}$ (valid memory address) signal. Figure 25 illustrates this protocol. Refer to Figure 26 for the read cycle timing and Figure 27 for the write cycle timing. In Figure 26, the DMAC latches the data at the falling edge of clock 19, so a latch to hold the data is necessary as shown in Figure 47.

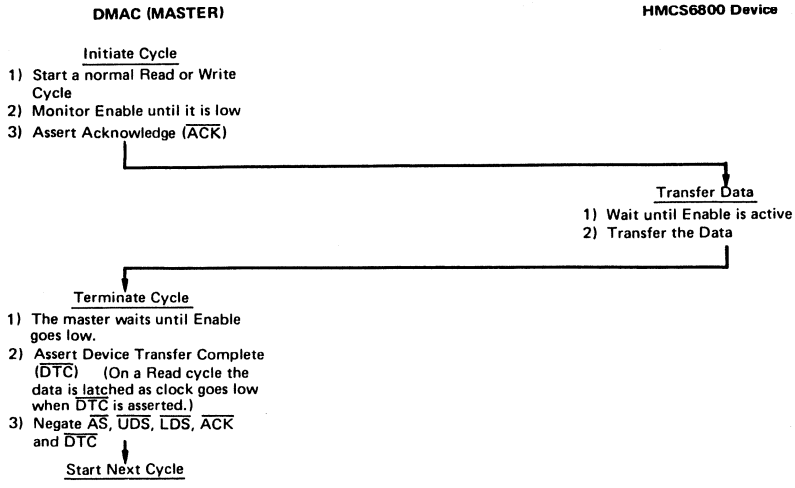


Figure 25 HMCS6800 Cycle Flowchart

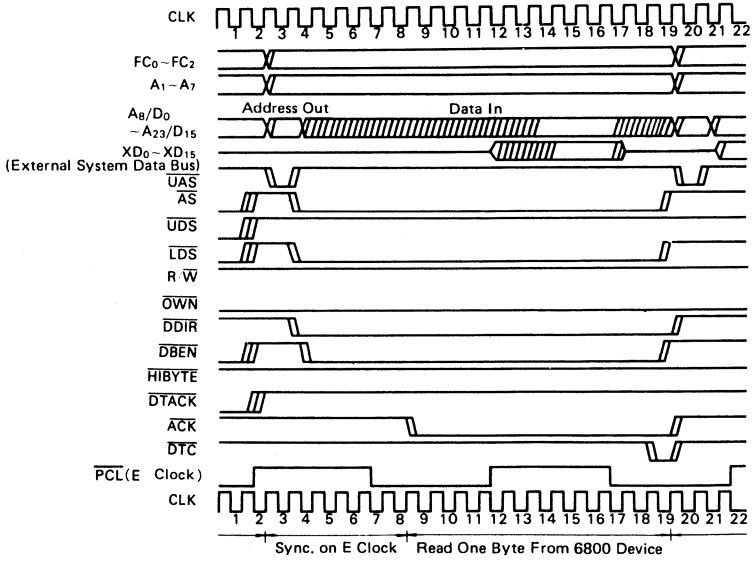


Figure 26 Dual Addressing Mode, HMCS6800 Compatible Device, Read Cycle

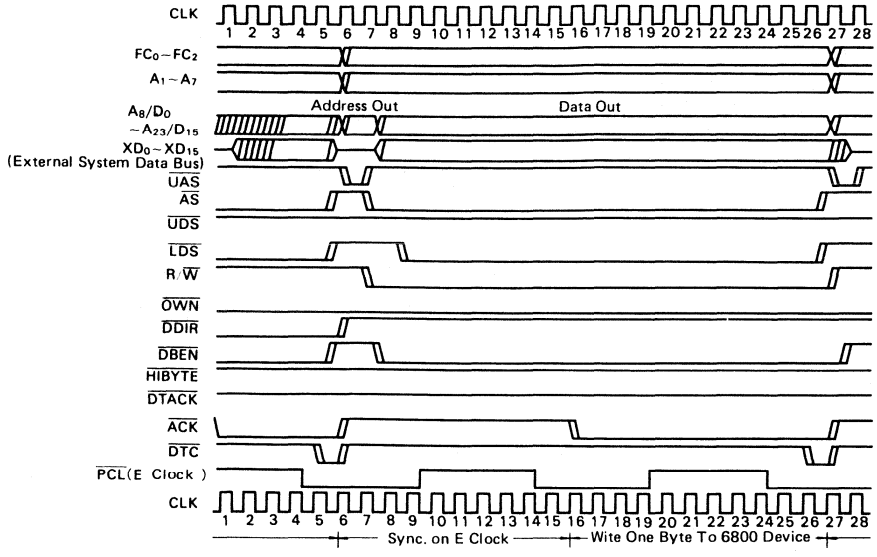


Figure 27 Dual Addressing Mode, HMCS6800 Compatible Device, Write Cycle

(2) Single Addressing Mode

Implicitly addressed devices are peripheral devices selected not by address but by $\overline{ACK}$. They do not require addressing of data register during data transfer. Transfers between memory and these devices are controlled by the request/acknowledge protocol. Such peripherals require only one bus cycle to transfer data, and the DMAC internal holding register is not used. Because only the memory is addressed during a data transfer and a transfer done in only on bus cycle, this protocol is called single-address.

Device with $\overline{ACK}$ Transfers

Under this protocol, the communication between peripheral device and the DMAC is performed with a two signal $\overline{REQ}/\overline{ACK}$ handshake. When a request is generated using the request method programmed in the DMAC's internal control registers, the DMAC obtains the bus and responds with $\overline{ACK}$. The DMAC asserts all the bus control signals required for the memory access. Refer to Figure 28 for the flowchart of the data transfer from memory to the device with $\overline{ACK}$. Figure 29 shows the flowchart of the data transfer from the device with $\overline{ACK}$ to memory. When a request is generated using the request method programmed in the control registers, the DMAC obtains the bus and responds with acknowledge. The DMAC asserts all HMCS68000 bus control signals needed for the transfer. When the DMAC accepts $\overline{DTACK}$ from memory, it asserts $\overline{DTC}$ and informs the peripheral device of the transfer termination. Figure 30 and 31 show the transfer timings of the device with $\overline{ACK}$: the port size for the former figure is 8-bit and the latter is 16-bit respectively.

When the transfer is from memory to a device, data is valid when $\overline{DTACK}$ is asserted and remains valid until the data strobes are negated. The assertion of $\overline{DTC}$ from the DMAC may be used to latch the data.

When the transfer is from device to memory, data must be valid on the HMCS68000 bus before the DMAC asserts the data strobes. The data strobes are asserted one clock period after $\overline{ACK}$ is asserted. When the DMAC obtains the bus and starts a DMA cycle, the tri-state of the $\overline{OWN}$ line is cancelled a half clock earlier than other control lines. If the DMA Cycle terminates and the DMAC relinquishes the bus, all the control signals get tri-stated a half clock before $\overline{OWN}$. The $\overline{DDIR}$ and $\overline{DBEN}$ lines are not asserted in the single addressing mode. Four clocks cycle is the smallest bus cycle for the transfer from memory to device. Five clocks cycle is the smallest bus cycle for the transfer from device to memory. If the device port size is 8-bit, either $\overline{LDS}$ or $\overline{UDS}$ is asserted. In the single adressing mode, A_8-A_{23} are outputted for only one and a half clock from the beginning of the DMA bus cycle. Therefore, A_8 through A_{23} needs to be latched externally just like in the dual addressing mode.

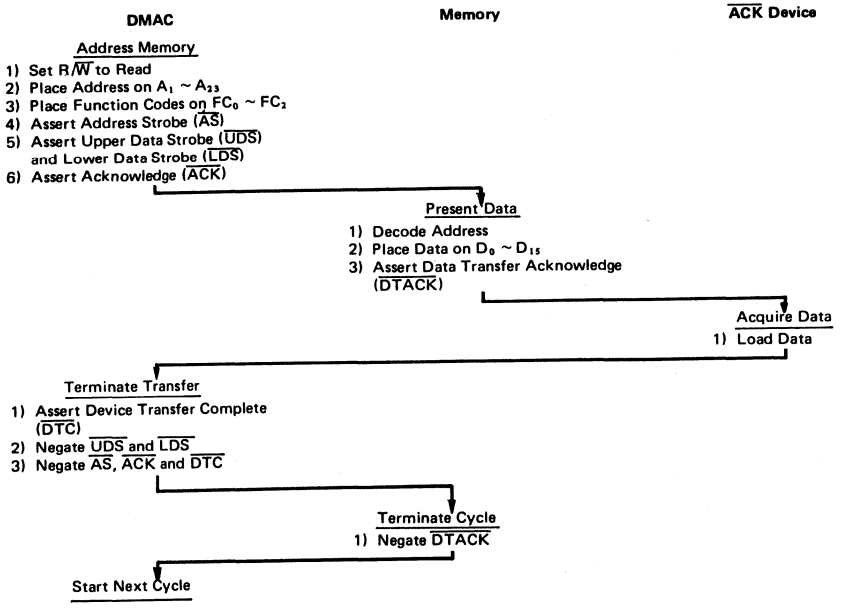


Figure 28 Word from Memory to Device with ACK

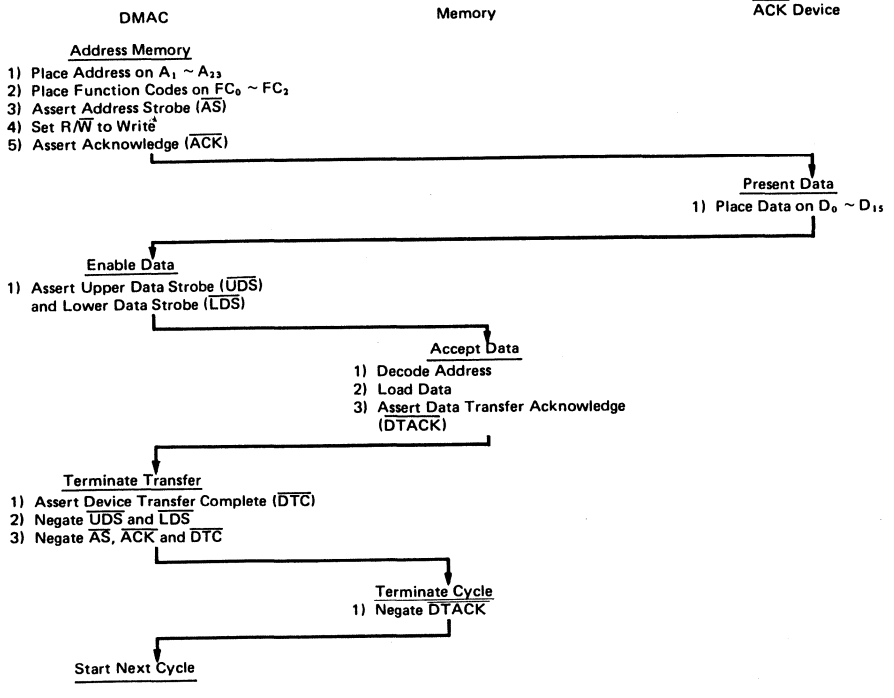


Figure 29 Word from Device with ACK to Memory

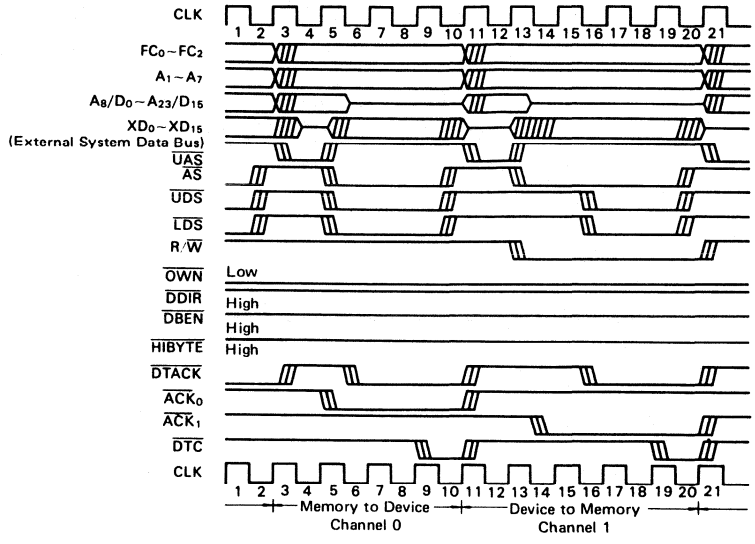


Figure 30 Single Addressing Mode with 16-Bit Devices as Source and Destination (Read-Write Cycles)

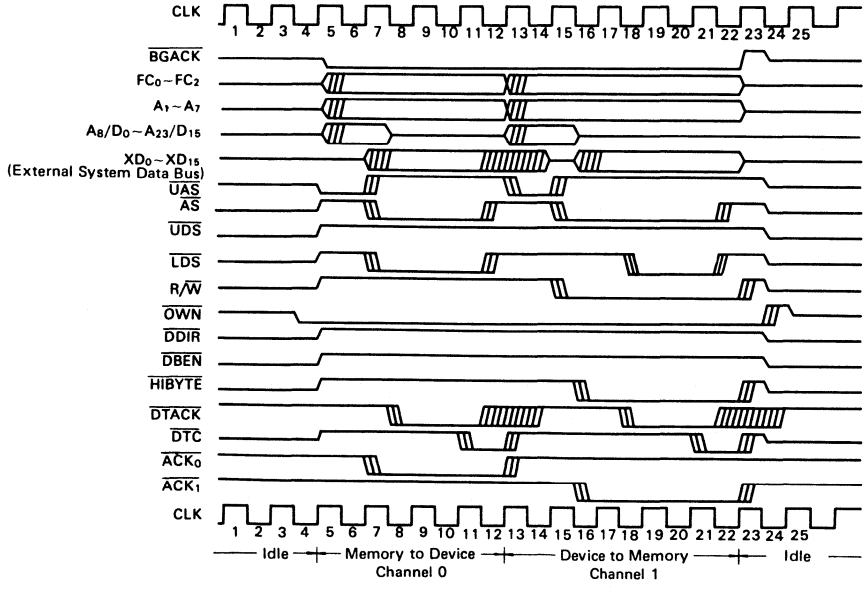


Figure 31 Single Addressing Mode with 8-Bit Device as Source and Destination (Read-Write Cycles)

Device with $\overline{ACK}$ and $\overline{READY}$ Transfers

Under this protocol, the communication between peripheral device and the DMAC is performed using a three signal $\overline{REQ}/\overline{ACK}/\overline{READY}$ handshake. The $\overline{READY}$ input to the DMAC is provided by the $\overline{PCL}$ line. The $\overline{READY}$ line is active low. When a request is generated using the request method programmed in the control registers, the DMAC obtains the bus and asserts $\overline{ACK}$ to notify the device that the transfer is to take place. The DMAC waits for $\overline{READY}$ ($\overline{PCL}$ input), which is a response from the device, in addition to $\overline{DTACK}$ which is a response from memory.

When the DMAC accepts both signals, it terminates the transfer. Refer to Figures 33 and 34 for the flowcharts of the data transfer between memory and the device with $\overline{ACK}$ and $\overline{READY}$. Refer to Figure 35 for the transfer timing of the 8-bit device. When the data transfer is from memory to a device, data is valid from the assertion of $\overline{DTACK}$ to the negation of $\overline{LDS}$ and $\overline{UDS}$. $\overline{DTC}$ is asserted a half clock before $\overline{LDS}$ and $\overline{UDS}$ are negated, so this line may be used for latching the data by the peripheral device. In this case, $\overline{READY}$ ($\overline{PCL}$ input) indicates that the device has received the data. Both $\overline{DTACK}$ and $\overline{READY}$ ($\overline{PCL}$ input) signals are needed for terminating the DMA cycle.

When the data transfer is from the device to memory, data must be valid on the bus before the DMAC asserts $\overline{LDS}$ and $\overline{UDS}$. Therefore, $\overline{READY}$ ($\overline{PCL}$ input) is used as the signal to indicate that the peripheral device has outputted the data on the bus. When the DMAC detects PCL ($\overline{READY}$ input), then it asserts $\overline{LDS}$ and $\overline{UDS}$. After asserting $\overline{LDS}$ and $\overline{UDS}$, the DMAC terminates the cycle when $\overline{DTACK}$ signal from the memory is detected.

When Array Chain or Link Array Chain is set in Device with $\overline{ACK}$ and $\overline{READY}$ Transfer mode, $\overline{READY}$ input is also necessary during DMA bus cycles for reading the chain information from memory. The circuit as shown in Figure 32 may be used in order to generate $\overline{READY}$ input when reading the chain information from memory.

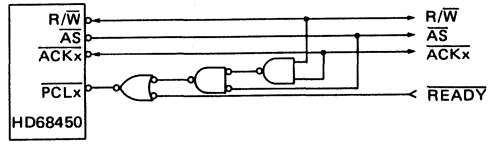


Figure 32 $\overline{READY}$ Circuit When Array or Link Array Chain is set for Device with $\overline{ACK}$ and $\overline{READY}$

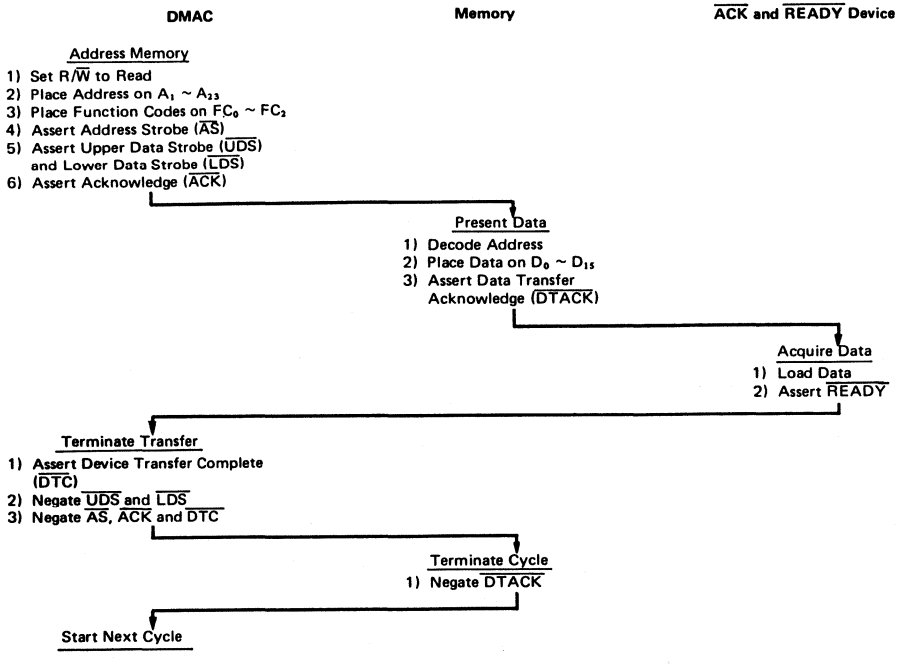


Figure 33 Word from Memory to Device with $\overline{ACK}$ and $\overline{READY}$

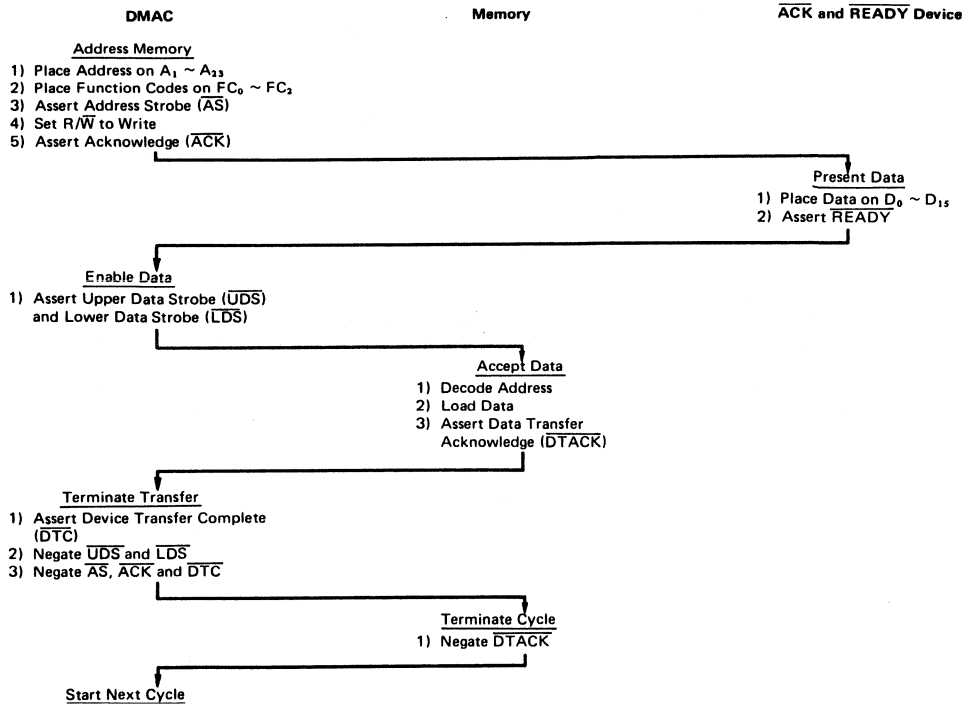


Figure 34 Word from Device with ACK and READY to Memory

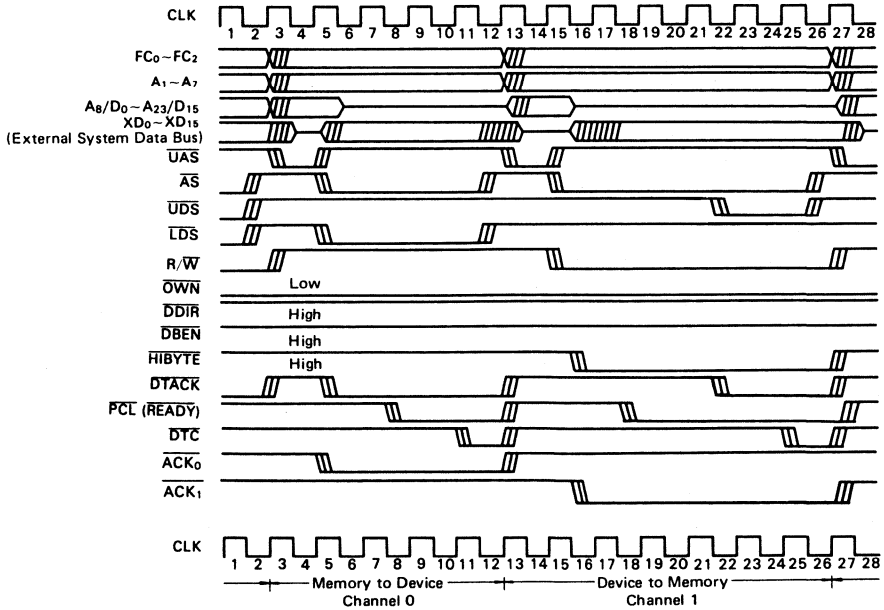


Figure 35 Single Addressing Mode with 8-Bit Devices as Source and Destination with PCL Used as a READY Input (Read-Write Cycles)

Operands and Addressing

Three factors enter into how the actual data is handled: port size, operand size and address sequencing.

Port Size

The DCR is used to program the device port size.

DPS Device Port Size
- 0 8 bit port
- 1 16 bit port

The port size is the number of bits of data which the device can transfer in a single bus cycle. During a DMAC bus cycle, a 16-bit port transfers 16 bits of data on $D_0 \sim D_{15}$, while an 8-bit port transfers 8 bits of data, either on $D_0 \sim D_7$ or on $D_8 \sim D_{15}$. The memory is always assumed to have a port size of 16.

Operand Size

OCR is used to program the operand size.

SIZE Operand Size
- 00 Byte
- 01 Word
- 10 Long word
- 11 (undefined, reserved)

The operand size is the number of bits of data to be transferred to honor a single request. Multiple bus cycles may be required to transfer the operand through the device port. A byte operand consists of 8 bits of data, a word operand consists of 16 bits of data, a long word operand consists of 32 bits of data. The transfer counter counts the number of operands transferred.

Table 2 indicates the combinations supported by the DMAC about the peripheral devices with different port size and operand sizes in the single and dual addressing mode. In the single addressing mode, port size and operand size must be the same. In the dual addressing mode, byte operand cannot be used when the port size is sixteen and the REQG bit is 10 or 11.

Table 2 Operation Combinations

Addressing	Device Type	Port	Operand			REQG bits of OCR
			Byte	Word	Long Word	
Dual	68000, 6800	8	O	O	O	00, 01, 10, 11
Dual	68000, 6800	16	O	O	O	00, 01
Dual	68000, 6800	16	X	O	O	10, 11
Single	with $\overline{ACK}$ or	8	O	X	X	00, 01, 10, 11
	$\overline{ACK}$ & $\overline{READY}$	16	X	O	X	00, 01, 10, 11

O ; enable X ; disable

(3) Address Sequencing

The sequence of addresses generated depends upon the port size, operand size, whether the addresses are to count up, down or not change and whether the transfer is executed in the single addressing mode or the dual addressing mode. The memory address count method and the peripheral device address count method is programmed using the Memory address count (MAC) bit and the Device address count (DAC) bit in the Sequence Control Register (SCR).

(i) Single addressing mode

In the single addressing mode, memory address sequencing is shown in Table 3. If the operand size is byte, the memory address increment is one (1). If the operand size is word, the memory address increment is two (2). If the memory address register does not count, the memory address is unchanged after the transfer.

If the memory address counts up, the increment is added to the memory address; if the memory address counts down, the increment is subtracted from the memory address. The memory address is changed after the operand is transferred.

Table 3 Single Address Sequencing

Port Size	Operand Size	Memory Address Increment		
		+ (increment)	= (unchanged)	- (decrement)
8	Byte	+1	0	-1
16	Word	+2	0	-2

(ii) Dual addressing mode

In the dual addressing mode, the operand size need not match the port size. Thus the transfer of an operand may require several DMA bus cycles. Each DMA bus cycle, between memory and DMAC and between DMAC and the device, is called the operand part and transfers a portion or all of the operand. The addresses of the operand parts are in a linear increasing sequence. The step between the addresses of the operand is two. The size of the operand parts is the minimum of the port size and the operand size. The number of the operand part is the operand size divided by the port size. In the dual addressing mode, memory is regarded as a device whose port size is 16-bits.

If the port size is 16 bits, the operand size is byte, and the request generation method is auto request or auto request at a limited rate, the DMAC packs consecutive transfers. This means that word transfers are made from the associated address with an address increment of two (2). If the initial source address location contains a single byte, the first transfer is a byte transfer to the internal DMAC holding register, and subsequent transfers from the source are word transfers. If the initial destination location contains a single byte, the first transfer is a byte transfer from the internal DMAC holding register, and any remaining byte remains in the holding register. Likewise, if either the final source or destination location contains a single byte, only a byte transfer is done. Packing is not performed if the address does not count; each byte is transferred by a separate access to the same location. The dual address sequencing is shown in Table 4.

Table 4 Dual Address Sequencing

Port Size	Operand Size	Part Size	Operand Part Address	Address Increment		
				+	=	-
8	Byte	Byte	A	+2	0	-2
8	Word	Byte	A, A+2	+4	0	-4
8	Long	Byte	A, A+2, A+4, A+6	+8	0	-8
16	Byte	Pack	A	+P	0	-P
16	Word	Word	A	+2	0	-2
16	Long	Word	A, A+2	+4	0	-4

P = 1 if packing is not done
= 2 if packing is done

Pack = byte if packing is not done
= word if packing is done

An Example of a Dual Address Transfer

This section contains an example of a dual address transfer using Table 4 of Dual-Address Sequencing. The table is reproduced here as Table 5. The transfer mode of this example is the following:

1. Device Port size = 8 bits
2. Operand size = Long Word (32 bits)
3. Memory to Device Transfer
4. Source (Memory) Counts up, Destination (Device) Counts Down
5. Memory Transfer Counter = 2

In this mode, a data transfer from the source (memory) is done according to the 6th row of Table 5, since the port size of the memory is always 16 bits. A data transfer to the destination (device) is done according to the 3rd row of Table 5. Table 6 shows the data transfer sequence.

The memory map of this example is shown in Table 7. The operand consists of BYTE A through BYTE D in memory of Table 7. Prior to the transfer, MAR and DAR are set to 00000012 and 00000108 respectively. The operand is transferred to the 8 bit port device according to the order of transfer number in Table 6.

Table 5 Dual-Address Sequencing (Table 4)

Row No.	Port Size	Operand Size	Operand Part Size	Operand Part Addresses	Address Increment		
					+	=	-
1	8	BYTE	BYTE	A	+2	0	-2
2	8	WORD	BYTE	A, A+2	+4	0	-4
③	8	LONG	BYTE *4	A, A+2, A+4, A+6 *3 *5 *7 *8	+8	0	-8 *10
4	16	BYTE	PACK (BYTE or WORD)**	A	+P	0	-P
5	16	WORD	WORD	A	+2	0	-2
⑥	16	LONG	WORD *2	A, A+2 *1 *6	+4 *9	0	-4

* Numbers in Table 5 correspond to ones in Table 6 and 7.
** Refer to Address Sequencing on Operand Part Size and PACK.

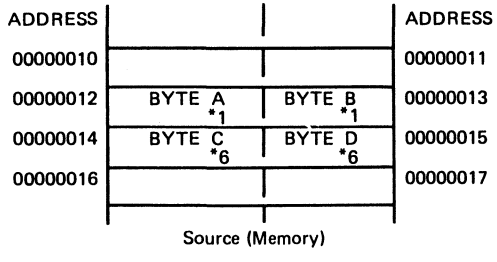
Table 6 An Example of a Data Transfer for One Operand

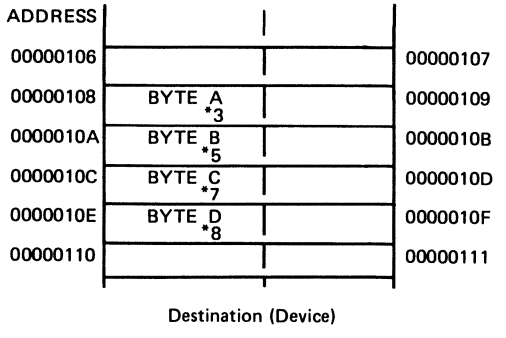
SRC: Source (Memory), DST Destination (Device), HR: Holding Register (DMAC Internal Reg.)

Transfer No.	Data Transfer	Address Output	Data Size on Bus	DMAC Registers after Transfer		Comment
				MAR	DAR	
0	–	–	–	00000012	00000108	Initial Register Setting
1	SRC → HR	00000012 *1	WORD *2	00000014	00000108	Higher order 16 bits of operand is fetched.
2	HR → DST	00000108 *3	BYTE *4	00000014	0000010A	Higher order 16 bits of operand is transferred.
3	HR → DST	0000010A *5	BYTE *4	00000014	0000010C *10	
4	SRC → HR	00000014 *6	WORD *2	00000016 *9	0000010C	Lower order 16 bits of operand is fetched
5	HR → DST	0000010C *7	BYTE *4	00000016	0000010E	Lower order 16 bits of operand is transferred.
6	HR → DST	0000010E *8	BYTE *4	00000016	00000110 *10	
6'	–	–	–	00000016	00000110	MAR, DAR are pointing the next operand addresses when the transfer is complete.

Mode: Port size = 8, Operand size = Long Word, Memory to Device, Source (Memory) Counts Up, Destination (Device) Counts Down

Table 7 Memory Map for the Example of the Data Transfer





• Initiation and Control of Channel Operation

(1) Operation Initiation

To initiate the operation of a channel the STR bit of the CCR is set to start the operation. Setting the STR bit causes the immediate activation of the channel, the channel will be ready to accept requests immediately. The channel initiates the operation by resetting the STR bit and setting the channel active bit in the CSR. Any pending requests are cleared, and the channel is then ready to receive requests for the new operation. If the channel is configured for an illegal operation, the configuration error is signaled, and no channel operation is run. The illegal operations include the selection of any of the options marked "(undefined, reserved)". If the MTC is set to zero in any operation or BTC is set to zero in the array chaining mode, then the count error is signaled and the channel is not activated. The channel cannot be started if any of the ACT, COC, BTC, NDT or ERR bits is set in the CSR. In this case, the channel signals the operation timing error.

(2) Operation Continuation (Continue Mode)

The continue bit (CNT) allows multiple blocks to be transferred in unchained operations. The CNT bit is set in order to continue the current channel operation. If an attempt is made to continue a chained operation, a configuration error is signaled. The base address register and base transfer counter should have been previously initialized.

The continue bit may be set as the channel is started or while the channel is still active. The operation timing error bit is signaled if a continuation is otherwise attempted.

When the memory transfer counter is exhausted and the continue bit of the CCR is set, the DMAC performs a continuation of the channel operation. The base address, base function code, and base transfer count registers are copied into the memory address, memory function code, and memory transfer count registers. The block transfer complete (BTC) bit of the CSR is set, the continue bit is reset, and the channel begins a new block transfer. If the memory transfer counter is loaded with a terminal count, the count error is signaled.

(3) Operation Halting (Halt)

The CCR has a halt bit which allows suspension of the operation of the channel. If this bit is set, a request may still be generated and recognized, but the DMAC does not attempt to acquire the bus or to make transfers for the halted channel. When this bit is reset, the channel resumes operation and services any request that may have been received while the channel was halted. However, in the burst request mode, the transfer request should be kept asserted until the initiation of the first transfer after clearing the halt bit.

(4) Operation Abort by Software (Software Abort)

Setting the software abort bit (SAB) in the CCR allows the current operation of the channel to be aborted. In this case, the ERR bit and the COC bit in the CSR are set and the ACT bit is reset. The error code for the software abort is set in the CER. The SAB bit is designed to be reset if the ERR bit is reset. When the CCR is read, the SAB always reads as zero(0).

(5) Interrupt Enable

The CCR has an interrupt enable bit (INT) which allows the channel to request interrupts. If INT is set, the channel can request interrupts. If it is clear, the channel will not request interrupts.

- **Channel Operation Termination**

As part of the transfer of an operand, the DMAC decrements the memory transfer counter (MTC). If the chaining mode is not used and the CNT bit is not set or the last block is transferred in the chaining mode, the operation of the channel is complete when the last operand transfer is completed and the MTC is zero. The DMAC notifies the peripheral device of the channel completion via the $\overline{\text{DONE}}$ output.

However, in the continue mode, $\overline{\text{DONE}}$ is outputted at the termination of every data block transfer. When the channel operation has been completed, the ACT bit of the CSR is cleared, and the COC bit of the CSR is set.

The occurrence of errors, such as the bus error, during the DMA bus cycle also terminates the channel operation. In this case, the ACT bit in the CSR is cleared, the ERR and the COC bits are set, and at the same time the code corresponding to the error that occurred is set in the CER.

(1) Channel Status Register (CSR)

The channel status register contains the status of the channel. The register, except for ACT and PCS bits, is cleared by writing a one (1) into each bit of the register to be cleared. Those bits positions which contain a zero (0) in the write data remain unaffected. ACT and PCS bits are unaffected by the write operation.

COC

The channel operation complete (COC) bit is set if the channel operation has completed. The COC bit must be cleared in order to start another channel operation. The COC bit is cleared only by writing a one to this bit or resetting the DMAC.

PCS

The peripheral status (PCS) bit reflects the level of the $\overline{\text{PCL}}$ line regardless of its programmed function. If $\overline{\text{PCL}}$ is at "High" level, the PCB bit reads as one. If $\overline{\text{PCL}}$ is at "Low" level, the PCS bit reads as zero. The PCS bit is unaffected by writing to the CSR.

PCT

The peripheral control transition (PCT) bit is set, if a falling edge transition has occurred on the $\overline{\text{PCL}}$ line. (The $\overline{\text{PCL}}$ line must remain at "low" level for at least two clock cycles.) The PCT bit is cleared by writing a one to this bit or resetting the DMAC.

BTC

Block transfer complete (BTC) bit is set when the continue (CNT) bit of CCR is set and the memory transfer counter (MTC) is exhausted. The BTC bit must be cleared before the another continuation is attempted (namely, setting the CNT bit again), otherwise an operation timing error occurs. The BTC bit is cleared by writing a one to this bit or resetting the DMAC.

NDT

Normal device termination (NDT) bit is set when the peripheral device terminates the channel operation by asserting the $\overline{\text{DONE}}$ line while the peripheral device was being acknowledged. The NDT bit is cleared by writing a one to this bit or resetting the DMAC.

ERR

Error (ERR) bit is set if any errors have been signaled. When the ERR bit is set, the code corresponding to the kind of the error that occurred is set in the CER. The ERR bit is cleared by writing a one to this bit or resetting the DMAC.

ACT

The active (ACT) bit is asserted after the STR bit has been set and the channel operation has started. This bit is remains set until the channel operation is terminated. The ACT bit is unaffected by write operations. This bit is cleared by the termination of the channel or resetting the DMAC.

(2) Interrupts

The DMAC can signal the termination of the channel operation by generating an interrupt request. The INT bit of the CCR determines if an interrupt can be generated. The interrupt request is generated by the following condition.

① INT = 1
and
② COC = 1 or BTC = 1 or ERR = 1 or NDT = 1 or PCT = 1
(the $\overline{\text{PCL}}$ line is an interrupt input)

This may be represented as

$$\overline{\text{IRQ}} = \overline{\text{INT}} \cdot (\text{COC} + \text{BTC} + \text{ERR} + \text{NDT} + \text{PCT}^*)$$

($^*\overline{\text{PCL}}$ line is programmed as an interrupt input.)

When the $\overline{\text{IRQ}}$ line is asserted, changing the INT bit from one to zero to one will cause the $\overline{\text{IRQ}}$ output to change from "low" to "high" to "low" again. The $\overline{\text{IRQ}}$ should be negated by clearing the COC, the BTC, the ERR, the NDT and the PCT bits.

If the DMAC receives $\overline{\text{IACK}}$ from the MPU during asserting the $\overline{\text{IRQ}}$, the DMAC provides an interrupt vector. If multiple channels have interrupt requests, the determination of which channel presents its interrupt vector is made using the same priority scheme defined for the channel operations.

The bus cycle in which the DMAC provides the interrupt vector when receiving an $\overline{\text{IACK}}$ from the MPU is called the interrupt acknowledge cycle. The interrupt vector returned to the MPU comes from either the normal or the error interrupt vector register. The normal interrupt register is used unless the ERR bit of CSR is set, in which case the error interrupt vector register is used. The content of the interrupt vector register is placed on $D_0 \sim D_7$, and $\overline{\text{DTACK}}$ is asserted to indicate that the vector is on the data bus. If a reset occurs, all interrupt vector registers are set to $0F (binary 00001111), the value of the uninitialized interrupt vector. The timing of the interrupt acknowledge cycle is shown in Figure 36. The HD68000 MPU outputs the interrupt level into A_1-A_3 and A_4-A_7 is held "high" during the interrupt acknowledge cycle, but the HD68450 DMAC ignores these signals.

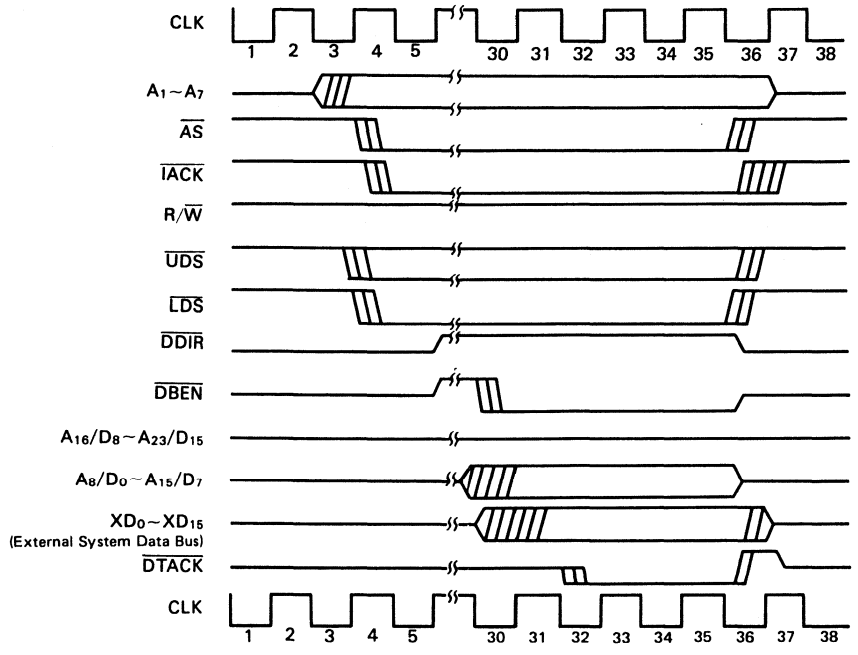


Figure 36 MPU $\overline{IACK}$ Cycle to DMAC

(3) Multiple Data Block Transfer Operation

When the memory transfer counter (MTC) is exhausted, the channel operation still continues if the channel is set to the array chaining mode or the linked array chaining mode and the chain is not exhausted. The channel operation also containes if the continue bit (CNT) of the CCR is set. The DMAC provides the initialization of the memory address register and the memory transfer counter in these cases so that the DMAC can transfer the multiple blocks.

Continued Operation

The continued operation is described in the Initiation and the Control of the Channel Operation section.

Array Chaining

This type of chaining uses an array in memory consisting of memory addresses and transfer counts. Each entry in the array is six bytes long and, consists of four bytes of address followed by two bytes of transfer count. The beginning address of this array is in the base address register, and the number of entries in the array is in the base transfer counter. Before starting any block transfers, the DMAC fetches the entry currently pointed to by the base address register. The address information is placed in the memory address register, and the count information is placed in the memory transfer counter. As each chaining entry is fetched, the base transfer counter is decremented by one. After the chaining entry is fetched, the base address register is incremented to point the next entry. When the base transfer counter reaches a terminal count of zero, the chain is exhausted, and the entry just fetched determines the last block of the channel operation.

An example of the array chaining mode operation and the memory format for supporting for array chaining is shown in Figure 37. The array must start at an even address, or the entry fetch results is an address error. If a terminal count is loaded into the memory transfer counter or the base transfer counter, the count error is signaled. Since the base registers may be read by the MPU, appropriate error recovery information is available should the DMAC encounter an error anywhere in the chain. Contents of the BFC is outputted as the function code when the DMAC is accessing the memory using the base address register. The value of the function code registers are unchanged in the array chaining operation.

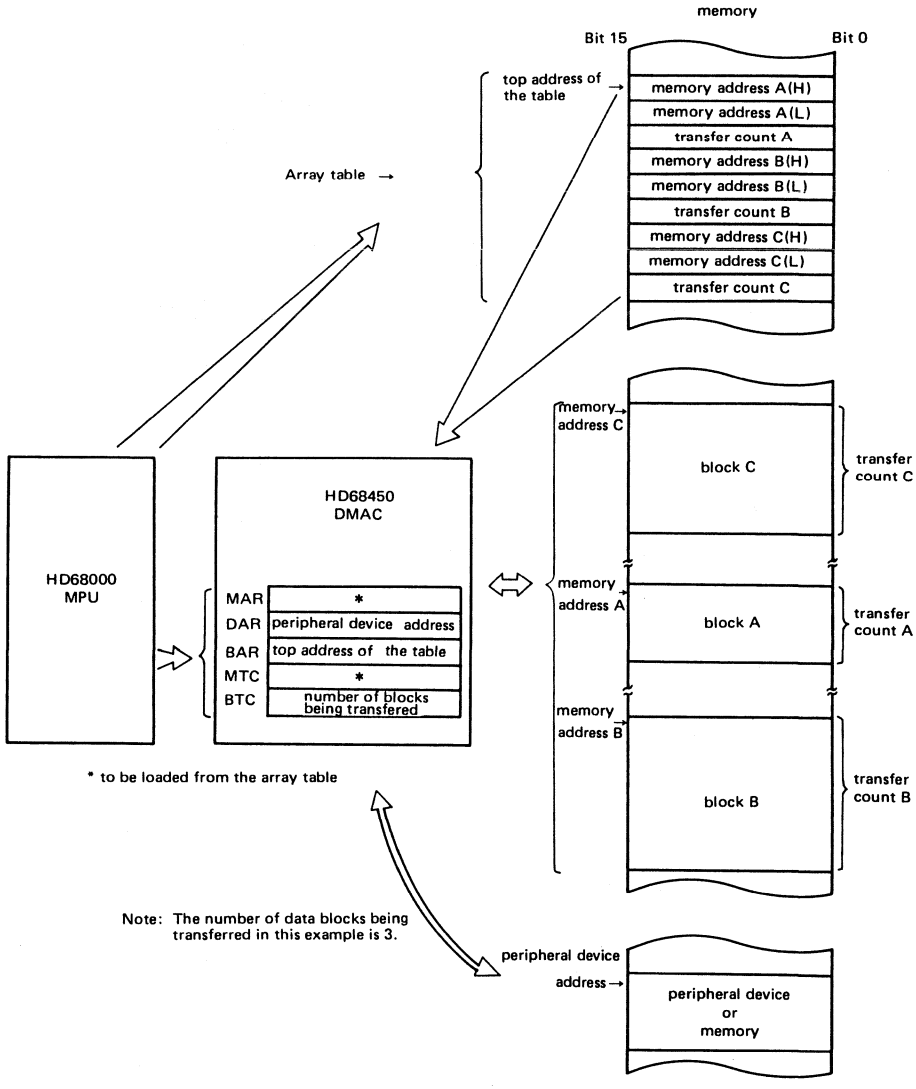


Figure 37 Transfer Example of the Array Chaining Mode

Linked Array Chaining

This type of chaining uses a list in memory consisting of memory address, transfer counts, and link addresses. Each entry in the chain list is ten bytes long, and consists of four bytes of memory address, two bytes of transfer count and four bytes of link address. The address of the first entry in the list is in the base address register, and the base transfer counter is unused. Before starting any block transfers, the DMAC fetches the entry currently pointed to by the base address register. The address information is placed in the memory address register, the count information is placed in the memory transfer counter, and the link address replaces the current contents of the base address register. The channel then begins a new block transfer. As each chaining entry is fetched, the update base address register is examined for the terminal link which has all 32 bits equal to zero. When the new base address is the terminal address, the chain is exhausted, and the entry just fetched determines the last block of the channel operation.

An example of the linked array chaining mode operation and the memory format for supporting it is shown is Figure 38.

In Figure 38, the DMAC transfers data blocks in the order of Block A, Block B, and Block C. In the linked array chaining

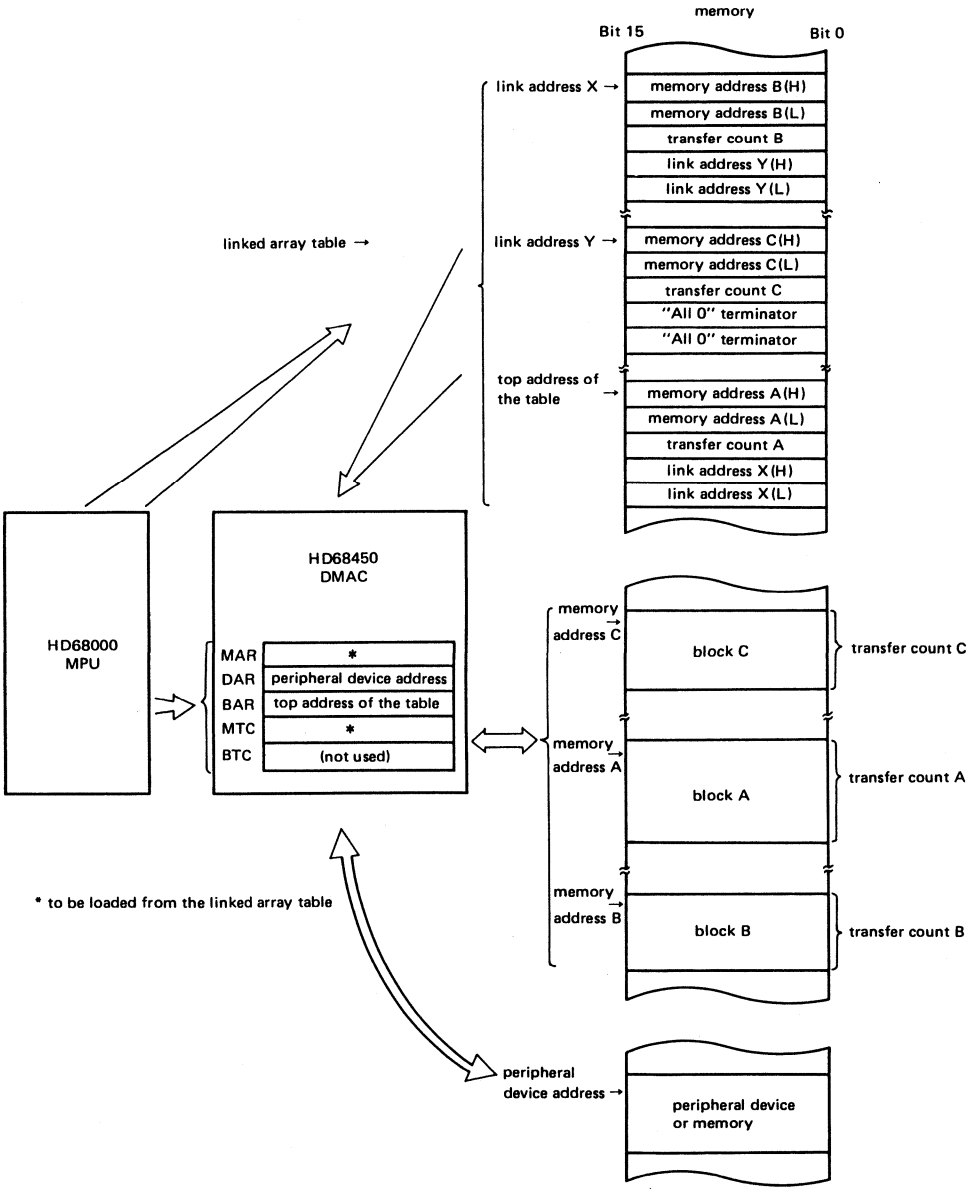


Figure 38 Transfer Example of the Linked Array Chaining Mode

mode, the BTC is not used. When the DMAC refers to the linked array table, the value of the BFC is outputted as the function code. The values of the function code registers are unchanged by the linked array chaining operation.

This type of chaining allows entries to be easily removed or inserted without having to reorganize data within the chain. Since the end of the chain is indicated by a terminal link, the number of entries in the array need not be specified to the DMAC.

The linked array table must start at an even address in the linked array chaining mode. Starting the table at an odd address results in an address error. If "0" is initially loaded to the MTC, the count error is signaled. Because the MPU can read all of the DMAC registers, all necessary error recovery information is available to the operating system.

The comparision of both chaining modes is shown in Table 8.

Table 8 Chaining Mode Address/Count Information

Chaining Mode	Base Address Register	Base Transfer Counter	Completed When
Array Chaining	address of the array table	number of data blocks being transferred	Base Transfer Count = 0
Linked Array Chaining	address of the linked array table	(unused)	Linked Address = 0

(4) Bus Exception Conditions

The DMAC has three lines for inputting bus exception conditions called $\overline{BEC_0}$, $\overline{BEC_1}$, and $\overline{BEC_2}$. The priority encoder can be used to generate these signals externally. These lines are encoded as shown in Table 9.

Table 9

$\overline{BEC_2}$	$\overline{BEC_1}$	$\overline{BEC_0}$	Exception Condition
1	1	1	No exception condition
1	1	0	Halt
1	0	1	Bus error
1	0	0	Retry
0	1	1	Relinquish bus and retry
0	1	0	(undefined, reserved)
0	0	1	(undefined, reserved)
0	0	0	Reset

In order to guarantee, reliable decoding, the DMAC verifies that the incoming code has been statable for two DMAC clock cycles before acting on it. The DMAC picks up $\overline{BEC_0}$-$\overline{BEC_2}$ at the rising edge of the clock. If $\overline{BEC_0}$-$\overline{BEC_2}$ is asserted to the undefined code, the operation of the DMAC does not proceed. For example, when the DMAC is waiting for $\overline{DTACK}$, inputting $\overline{DTACK}$ does not result in the termination of the cycle if $\overline{BEC_0}$-$\overline{BEC_2}$ is asserted to the undefined code. In addition, when the transfer request is received, $\overline{BR}$ is not asserted if the $\overline{BEC_0}$-$\overline{BEC_2}$ is not set to no exception condition.

If exception condition, except for HALT, is inputted during the DMA bus cycle prior to, or in coincidence with $\overline{DTACK}$, the DMAC terminates the current channel operation immediately. Here coincident means meeting the same set up requirements for the same sampling edge of the clock. If a bus exception condition exists, the DMAC does not generate any bus cycles until it is removed. However, the DMAC still recognizes requests.

Halt

The timing diagram of halt is shown in Figure 39. This diagram shows halt being generated during a read cycle from the 68000 compatible device in the dual addressing mode. If the halt exception is asserted during a DMA bus cycle, the DMAC does not terminate the bus cycle immediately. The DMAC waits for the assertion of $\overline{DTACK}$ before terminating the bus cycle so that the bus cycle is completed normally. In the halted state, the DMAC puts all the control signals to high impedance and relinquishes the bus to the MPU. The DMAC does not output the $\overline{BR}$ until halt exception is negated. When halt exception is negated, the DMAC acquires the bus again and proceeds the DMA operation. In order to insure a halt exception operation, the $\overline{BEC}$ lines must be set to halt at least until the assertion of $\overline{DTC}$.

If the DMAC has the bus, but is not executing any bus cycle, the DMAC relinquishes the bus as soon as halt exception is asserted.

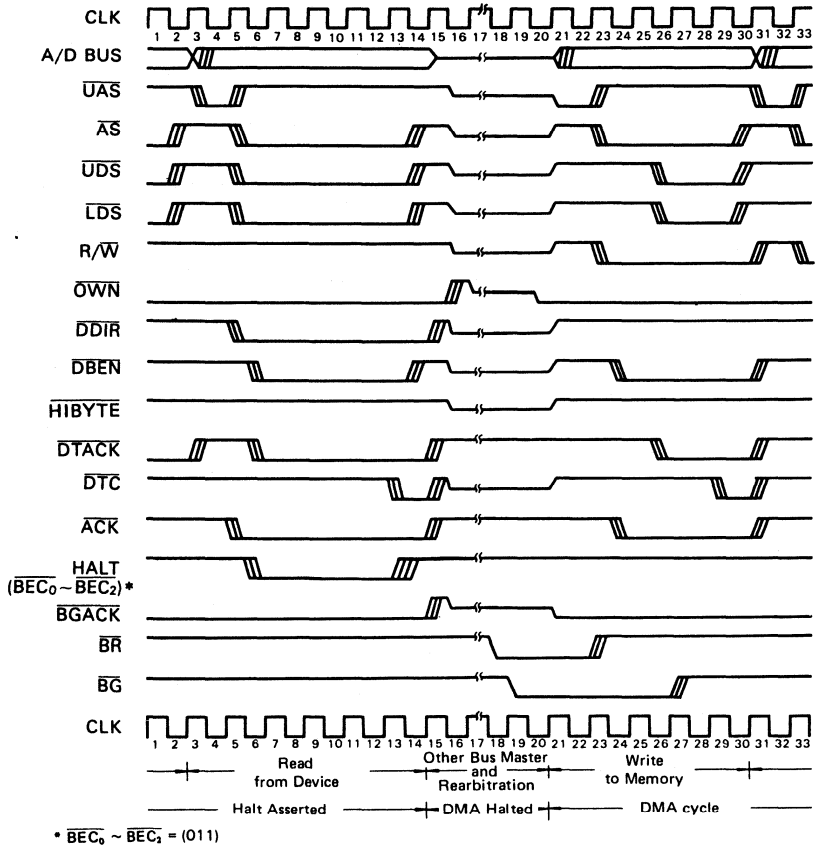
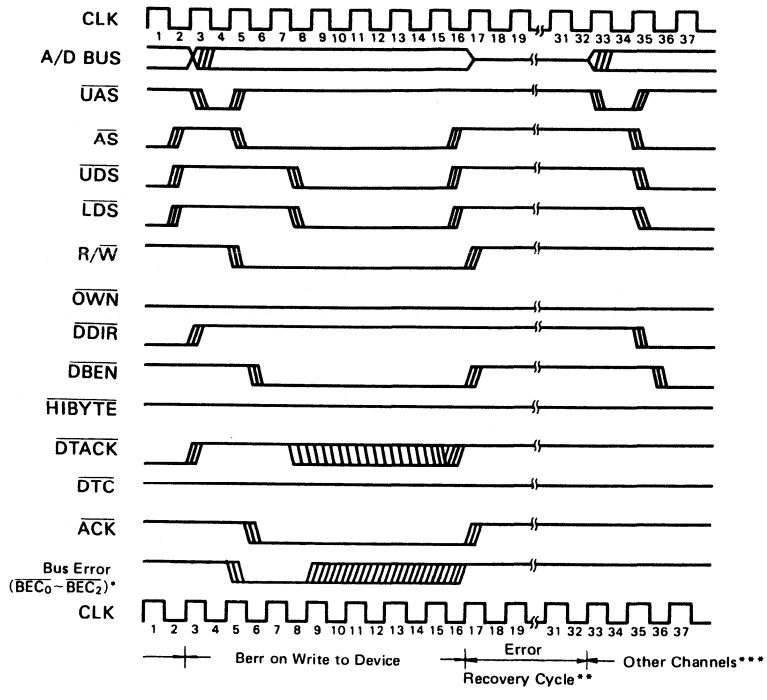


Figure 39 Halt Operation

Bus Error

The bus error exception is generated by external circuitry to indicate the current transfer cannot be successfully completed and is to be aborted. The recognition of this exception during a DMAC bus cycle signals the internal bus error condition for the channel for which the current bus cycle is being run. As soon as the DMAC recognizes the bus error exception, the DMAC immediately terminates the bus cycle and proceeds to the error recovery cycle. In this cycle, the DMAC adjusts the values of the MAR, the DAR, the MTC and the BTC to the values when the bus error exception occurred. 25 clocks are required for the error recovery cycle in the single addressing mode and in the read cycle of the dual addressing mode. 29 clocks are required in the write cycle of the dual addressing mode. If the DMAC does not have any transfer request in the other channels after the error recovery cycle, the DMAC relinquishes the bus.

The diagram of the bus error timing is shown in Figure 40.



* $\overline{BEC_0}$·$\overline{BEC_2}$ = (101)
** In the single addressing mode and in the read cycle of the dual addressing mode: 25 clocks
In the write cycle of the dual addressing mode: 29 clocks
*** The DMAC keeps the bus because the other channels have requests pending. If other channels do not have requests, the DMAC relinquishes the bus after the error recovery cycle.

Figure 40 Bus Error Operation

Retry

The retry exception causes the DMAC to terminate the present operation and retry that operation when retry is removed, and thus will not honor any requests until it is removed. However, the DMAC still recognizes requests. The retry timing is shown in Figure 41.

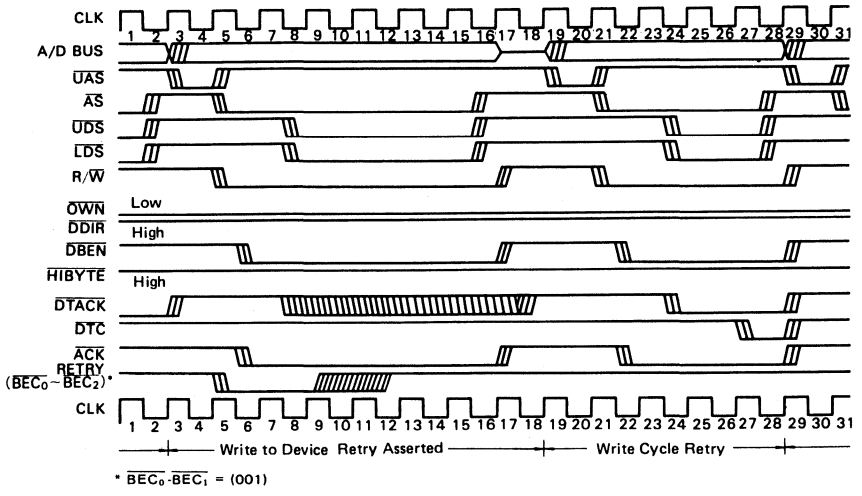


Figure 41 Retry Operation

Relinquish and Retry (R&R)

The relinquish and retry exception causes the DMAC to relinquish the bus and three-state all bus master controls and when the exception is removed, rearbitrate for the bus to retry the previous operation.

The diagram of the relinquish and retry timing is shown in Figure 42.

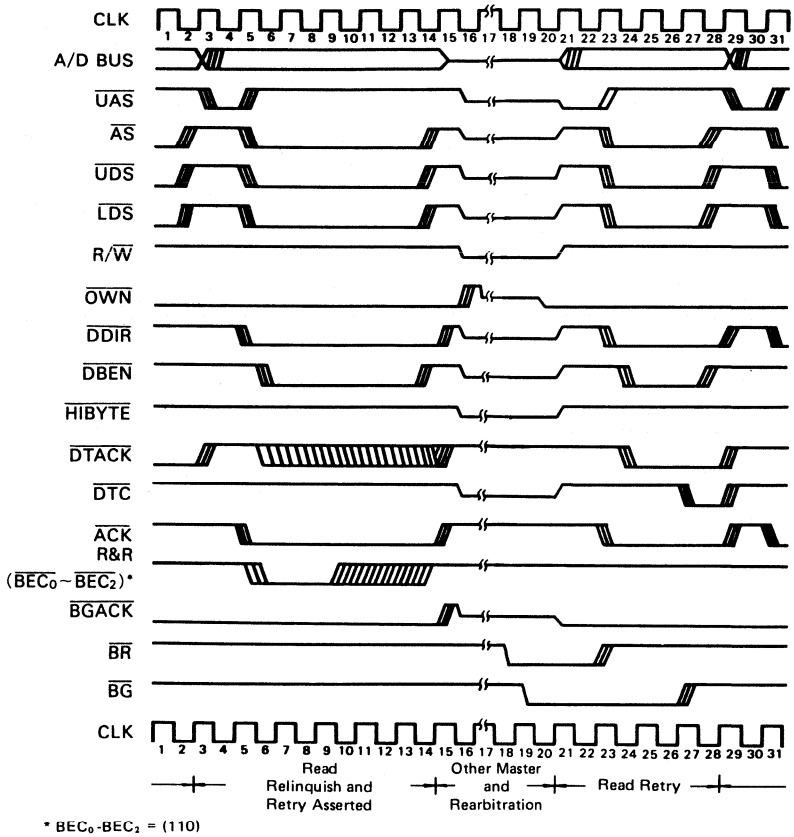


Figure 42 Relinquish and Retry Operation

Reset

The reset provides a means of resetting and initializing the DMAC. If the DMAC is bus master when the reset is asserted, the DMAC relinquishes the bus. Reset clears GCR, DCR, OCR, SCR, CCR, CSR, CPR, and CER for all channels. The NIV and the EIV are all set to $(0F)_{16}$, which is the uninitialized interrupt vector number for the HD68000 MPU. MTC, MAR, DAR, BTC, BAR, MFC, DFC, and BFC are not affected. In order to insure a reset, $\overline{BEC_0} \sim \overline{BEC_2}$ must be kept at "Low" level for at least ten clocks.

(5) Error Conditions

When an error is signaled on a channel, all activity on that channel is stopped. The ACT bit of the CSR is cleared, and the COC bit is set. The ERR bit of the CSR is set, and the error code is indicated in the CER. All pending operations are cleared, so that both the STR and CNT bits of CCR are cleared.

Enumerated below are the error signals and their sources.

(a) Configuration Error – This error occurs if the STR bit is set in the following cases.
  (i) the CNT bit is set at the same time STR bit in the chaining mode.
  (ii) DTYP' specifies a single addressing mode, and the device port size is not the same as the operand size.
  (iii) DTYP specifies a dual addressing mode, DPS is 16 bits, SIZE is 8 bits and REQG is "10" or "11".
  (iv) an undefined configuration is set in the registers. The undefined configurations are: XRM = 01, MAC = 11, DAC = 11, CHAIN = 01, and SIZE = 11.

(b) Operation Timing Error – An operation timing error occurs in the following cases:
  (i) when the CNT bit is set after the ACT bit has been set by the DMAC in the chaining mode, or when the STR and the ACT bits are not set.
  (ii) the STR bit is set when ACT, COC, BTC, NDT or ERR is set.
  (iii) an attempt to write to the DCR, OCR, SCR, MAR, DAR, MTC, MFC, or DFC is made when the STR bit or the ACT bit is set.
  (iv) an attempt to set the CNT bit is made when the BTC and the ACT bits are set.

(c) Address Error – An address error occurs in the following cases:
  (i) an odd address is set for word or long word operands.
  (ii) $\overline{CS}$ or $\overline{IACK}$ is asserted during the DMA bus cycle.

(d) Bus Error – Bus error occurs when a bus error excep-

tion is signaled during a DMA bus cycle.

(e) Count Error – A count error occurs in the following cases:
  (i) The STR bit is set when zero is set in the MTC and the chaining mode is not used.
  (ii) The STR bit is set when zero is set in BTC for the array chaining mode.
  (iii) Zero is loaded from memory or the BTC to the MTC in the chaining modes or the continue mode.

(f) External Abort – External abort occurs if an abort is asserted by the external circuitry when the $\overline{PCL}$ line is configured as an abort input and the STR or the ACT bit is set.

(g) Software abort – Software abort occurs if the SAB bit is set when the STR or the ACT bit is set.

Error Recovery Procedures

If an error occurs during a DMA transfer, appropriate information is available to the operating system (OS) to allow a software failure recovery operation. The operating system must be able to determine how much data was transferred, where the data was transferred to, and what type of error ocurred.

The information available to the operating system consists of the present value of the Memory Address, Device Address and Base Address Registers, the Memory Transfer and Base Transfer Counters, the channel status register, the channel error register. After the successful completion of any transfer, the memory and device address registers points to the location of the next operand to be transferred and the memory transfer counter contains the number of operands yet to be transferred. If an error occurs during a transfer, that transfer has not completed and the registers contain the values they had before the transfer was attempted. If the channel operation uses chaining, the Base Address Register points to the next chain entry to be serviced, unless the termination occurred while attempting to fetch an entry in the chain. In that case, the Base Address Register points to the entry being fetched. However, in the case of external abort, there are cases in which the previous values are not recovered.

Bus Exception Operating Flow

The bus exception operating flow in the case of multiple exception conditions occurring continuously in sequence is shown in Figure 43. Note that the DMAC can receive and execute the next exception condition. For example, if the retry exception occurs, and next the relinquish and retry exception occurs while the DMAC is waiting for the retry condition to be cleared, the DMAC relinquishes the bus and waits for the exception condition to be cleared. If a bus error occurs during this period, the DMAC executes the bus error exception operation.

The flow diagram of the normal operation without exception operation or errors is shown in Figure 44.

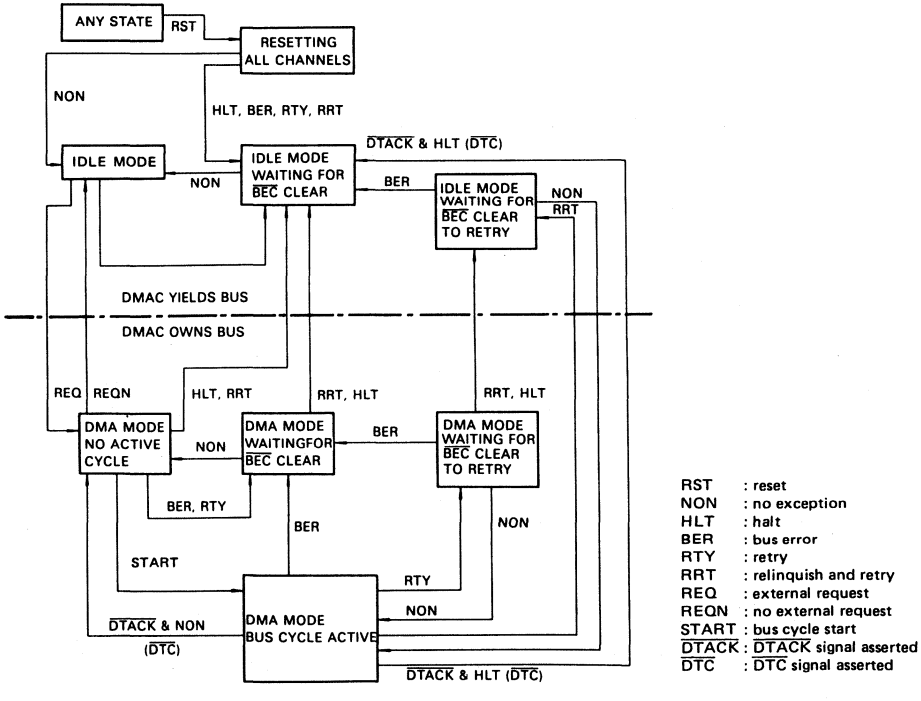


Figure 43 Bus Exception Flow Diagram

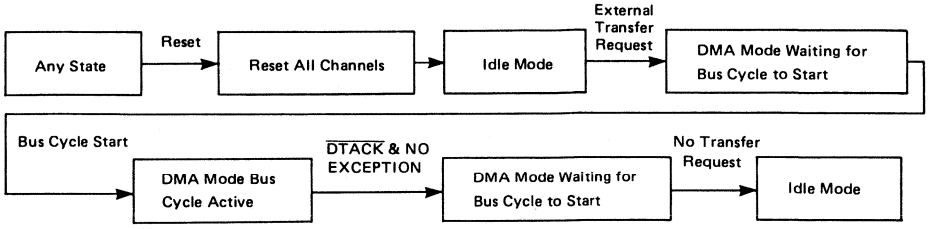


Figure 44 Flow of Normal Operation Without Exception or Error Condition

- **Channel Priorities**

Each channel has a priority level, which is determined by the contents of the Channel Priority Register (CPR). The priority of a channel is a number from 0 to 3, with 0 being the highest priority level. When multiple requests are pending at the DMAC, the channel with the highest priority receives first service. The priority of a channel is independent of the device protocol or the request mechanism for that channel. If there are several requesting channels at the highest priority level, a round-robin resolution is used, that is, as long as these channels continue to have requests, the DMAC does operand transfers in rotation.

Resetting the DMAC puts the priority level of all channels to "0", the highest priority level.

■ APPLICATIONS INFORMATION

Examples of how to interface HD68450 to an HD68000 based system are shown in Figure 45 and Figure 46.

Figure 45 shows an example of how to demultiplex the address/data bus. $\overline{OWN}$ and $\overline{UAS}$ are used to control 74LS373 for latching the address. $\overline{DBEN}$ and $\overline{DDIR}$ are used to control the bi-directional buffer 74LS245.

Figure 46 shows an example of inter-device connection in the HMCS68000 system.

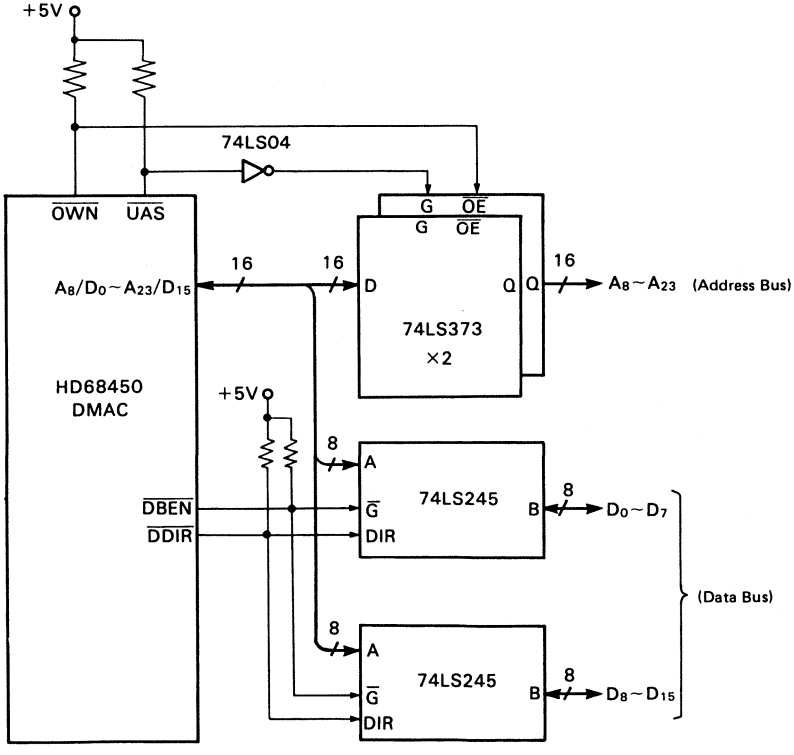
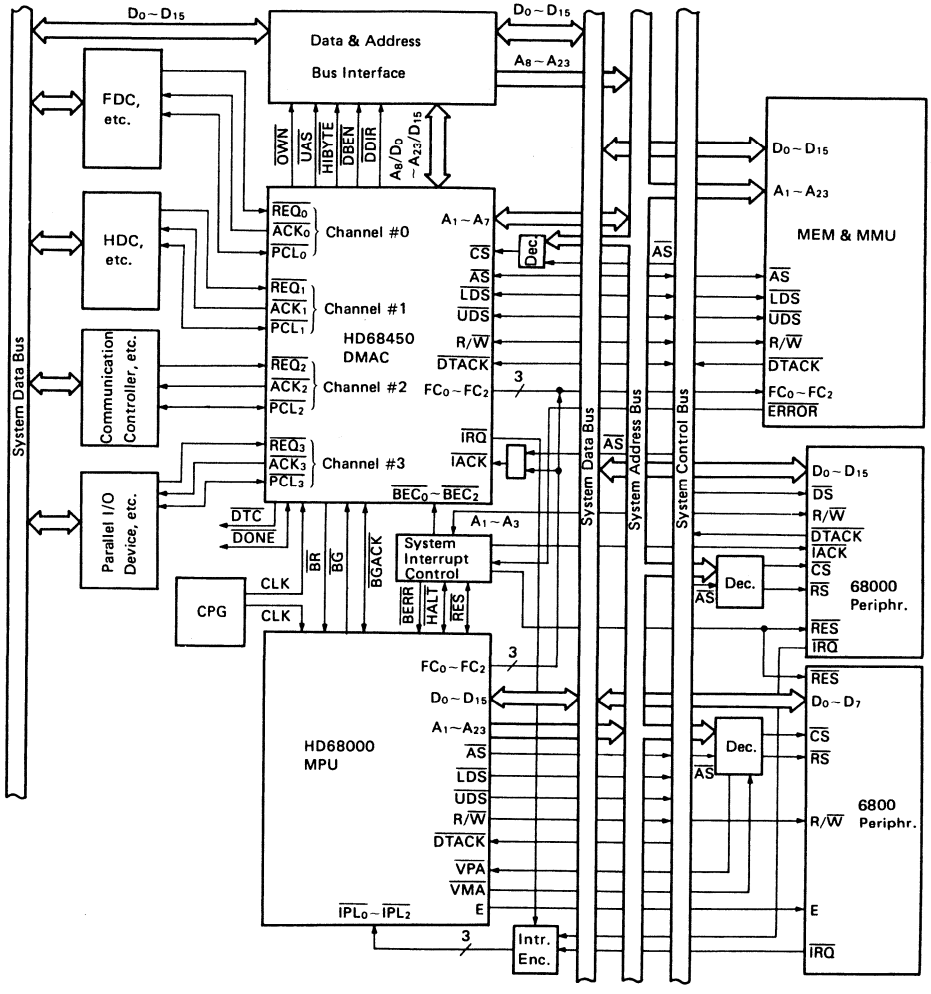


Figure 45 An Example of the Demultiplexed Address Data Bus



The address bus and the system control bus in each device are omitted in this Figure.

Figure 46 An Example of Inter-device Connection in the HMCS68000 System

■ ATTENTION ON USAGE

(1) How to interface various 6800 type peripheral devices to the DMAC based system.

When the DMAC is reading data from the 6800 device, the DMAC latches the data when $\overline{DTC}$ is asserted and not at the falling edge of E clock. The 74LS373 needs to be provided externally as shown in Figure 47 so that the data from the 6800 device can be held on the bus for a large period of time until the DMAC can latch the correct data.

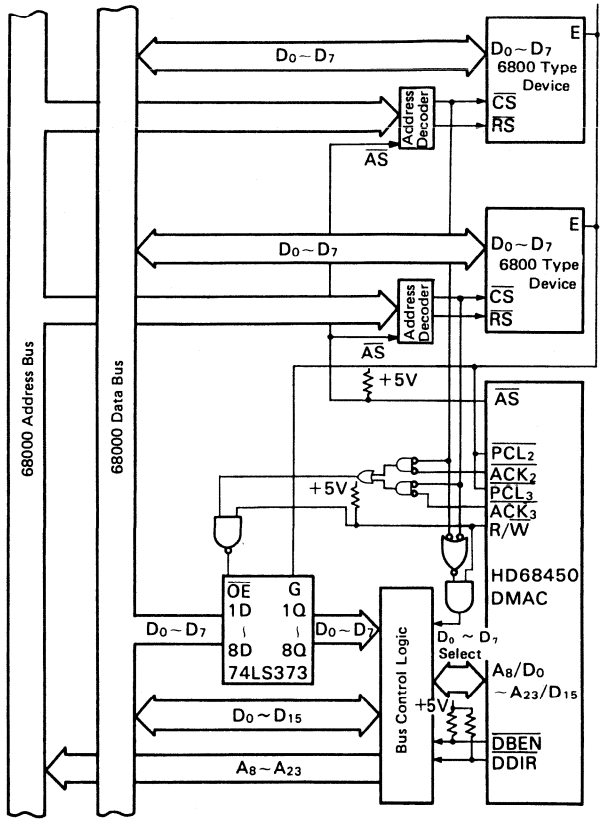


Figure 47 An Example of Connection with 6800 type Peripheral Devices (channel 2 and 3 are used)

(2) When "external abort" is inputted during the $\overline{DONE}$ input cycle

When the transfer direction is from the peripheral device to memory and $\overline{PCL}$ signal is set to the external abort input mode in the dual addressing mode, the external abort will be ignored during the subsequent write cycle from the DMAC's internal holding register to memory if $\overline{DONE}$ is inputted during the read cycle from the peripheral device to the DMAC's internal holding register.

In this case, the channel status register (CSR) and the channel error register (CER) show the normal termination caused by $\overline{DONE}$ Input. The user is able to examine the PCT bit and the ERR bit in order to detect the external abort inputted at the timing described above. If PCT = 1, ERR = 0, and NDT = 1, then an external abort has occurred.

(3) Multiple Errors

The DMAC will log the first error encountered in the channel error register. If an error is pending in the error register and another error is encountered the second error will not be logged. Even though the second error is not logged in the CER, it will still be recognized internally and the channel will not start.

(4) Relinquish & Retry Exception During Dual Address Mode Operation

When the following two conditions occur simultaneously, incorrect data is outputted by the DMAC at the write cycle immediately following the negation of the relinquish & retry (R&R) exception.

(1) R&R is asserted at the write cycle in the dual address mode.

(2) MPU access to the DMAC's internal register is done after the DMAC relinquishes the bus due to R&R exception.

When the R&R exception occurs during the write cycle of

the dual addressing mode, and the MPU accesses the DMAC's register, then the DMAC's proper operation sequence should be the following (refer to the Fig. 48):

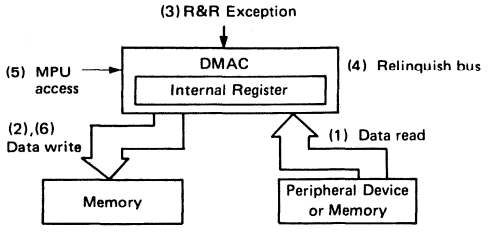


Fig. 48

(1) Data is read by the DMAC during the read cycle
(2) Data read at (1) is outputted at the write cycle
(3) R&R exception is asserted during the write cycle
(4) DMAC relinquishes the bus
(5) MPU accesses the DMAC and it is completed normally
(6) When R&R exception is negated, the DMAC obtains the bus and write cycle is done to output the data read at (1).

But instead, incorrect data is outputted at (6). This is because the data read at (1), which is held internally by the DMAC, is destroyed at (5) when the MPU accesses the DMAC. Avoid occurance of the above condition. For example:

(1) Assert R&R exception only during the read cycle when using dual addressing mode.
(2) If the R&R exception occurred at the write cycle of the dual addressing mode, avoid accessing the DMAC's registers.
(3) Use HALT exception instead of R&R exception to access the DMAC's internal registers.

(5) $\overline{CS}$ Negation Timing

When the $\overline{LDS}$, $\overline{UDS}$ high to $\overline{CS}$ high timing (chip select negation delay) is over 1 clock and the MPU access is long word (32-bit data), then the data stored in the lower word of the register accessed is destroyed and becomes all zeros. In other words, the data in the lower word of the read access is lost and cleared to all zeros. This does not always occur, but on occasional basis due to the asynchronous input timing of the CS signal.

Please observe the timing specification shown in Fig. 49.

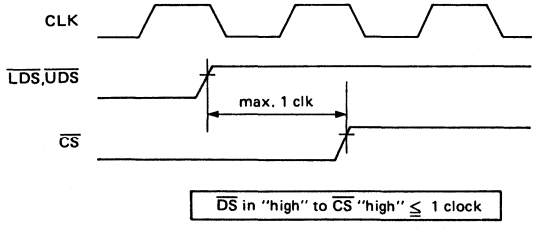


Fig. 49

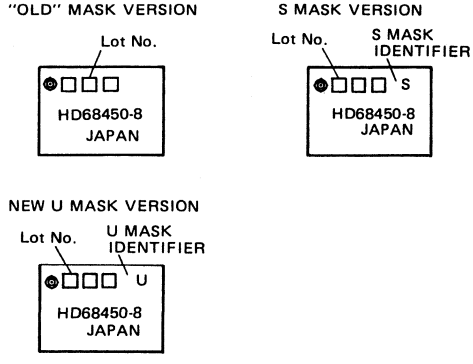
(6) Unused Function Code (FC_0–FC_2) Lines

When the FC_0, FC_1, and FC_2 lines are not used, please keep these lines "high" by using a pull-up resistor. If these lines are left unconnected, the HD68450 DMAC may not operate properly.

(7) The use of thick wiring is recommended between Vss of the HD68450 and the ground of the circuit board. When a socket is used to install the DMAC on the board, please make sure that the contact of the Vss pins are made well.

■ DIFFERENCES AND COMPATIBILITY AMONG THE MASK VERSIONS

• Marking



• The "OLD" Mask

For the "Old" mask version of the HD68450 DMAC, please refer to the following material, which may be obtained from the sales office:

Microcomputer Device/System Technical Information
No. T-026
"Anomalies of the HD68450 DMAC" Aug. 26, 1983

• The "S" Mask

With the S mask version of the HD68450 DMAC, the anomalies listed in the "Microcomputer Device/System Technical Information, No. T-026 – Anomalies of the HD68450 DMAC" (dated Aug. 26, 1983) is fixed with the exception of item #3 – "Extra data transfer in the burst mode." For the description of the S mask's anomaly, please contact the sales office.

The other remaining anomaly in the "S" mask version – one byte of transfer data is left in the DMAC.

When the DMAC is set to dual addressing mode, port size 8 bits, external request mode, and data transfer from peripheral device to memory, the last byte of the transfer may be left inside the DMAC's internal register without being transferred to memory if the transfer is stopped before the transfer count is exhausted. The last byte that is left inside the DMAC becomes unaccessable by the MPU.

In this mode, the DMAC transfers data repeating the following bus cycles:

(1) READ BYTE
(Byte is read from the peripheral device to DMAC)

(2) READ BYTE
(Byte is read from the peripheral device to DMAC)

(3) WRITE WORD
(Word is written to memory from DMAC)

If the transfer is terminated after (1) READ BYTE (see NOTE*), then the byte data that was ready by (1) READ BYTE bus cycle is not written to memory and is left inside the DMAC's internal holding register. The DMAC's internal holding register cannot be accessed by the MPU, so that it becomes "lost".

This will not occur when single addressing mode is used. So, please use the single addressing mode when the transfer needs to be terminated before the transfer is exhausted.

NOTE*: The methods to terminate the transfer operation before the transfer counter becomes zero are (1) assert external abort using the $\overline{PCL}$. (2) set the SAB bit to cause software abort.

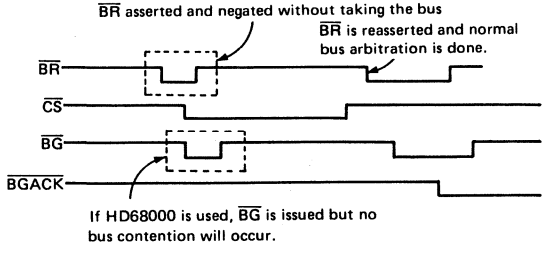
- **The New "U" Mask**

The remaining anomaly of the S mask – "Extra data transfer in the burst mode" is fixed in the U mask. In order to correct the remaining anomaly of the S mask-one byte of transfer data is left in the DMAC, the U mask has a new mode operation that repeats READ BYTE – WRITE BYTE operation in the dual addressing mode, port size 8 bits, external request mode, and data transfer from peripheral device to memory. The new mode uses the SIZE = 11 in the operation control register. In the "OLD" mask and the S mask version, SIZE = 11 causes an configuration error.

- **Bus Arbitration Problem in the S and U mask version**

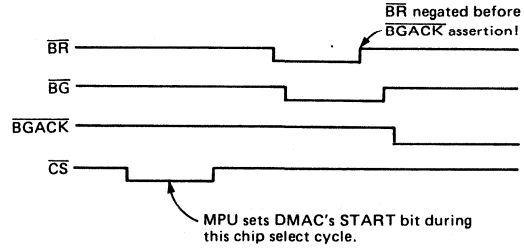
$\overline{BR}$ Negation on MPU's $\overline{CS}$

If the MPU asserts DMAC's $\overline{CS}$ when the DMAC has its $\overline{BR}$ asserted, then the DMAC negates its $\overline{BR}$. This is shown as follows:



$\overline{BR}$ negation before $\overline{BGACK}$ assertion

When the MPU sets the START bit of the DMAC, and a different channel is already active in the limited-rate auto-request mode, then the following bus arbitration timing may occur.



In this timing, the $\overline{BR}$ is negated too early, e.g. before the $\overline{BGACK}$ is asserted. This will cause bus contention between the DMAC and the MPU. For the description of these problems, please contact the sales office.

HD63463

HDC (Hard Disk Controller)

DESCRIPTION

The HD63463 (HDC: Hard Disk Controller) is a CMOS device developed for use as a peripheral LSI for the 16-bit microprocessor HD68000 (MPU: Microprocessing Unit). The HDC connects the host system and the Winchester type hard disk device with or without HD68450 (DMAC: Direct Memory Access Controller).

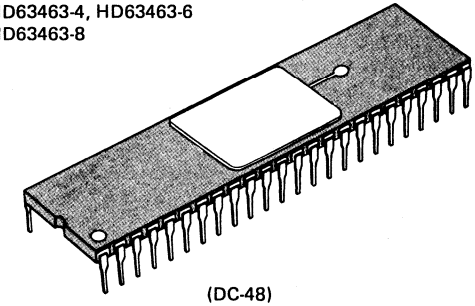
FEATURES

- Two types of disk interface, ST506/ST412/ST412HP and SMD
- Choice of data bus width — 16 or 8 bits
- Serial data transfer rate — max 20 Mbit/s [NRZ], max 10 Mbit/s [MFM]
- Step rate — max 1.1 MHz
- Internal data buffer (256 bytes x 2)
- Zero-sector interleave access
- Programmed I/O access
- Automatic error correction
- Zero pattern detector/generator, address mark detector/generator, write precompensation logic, etc internally provided
- 22 high-level function commands
- 25 error codes
- External circuit diagnosis command
- Low power dissipation — typ 250 mW

TYPE OF PRODUCTS

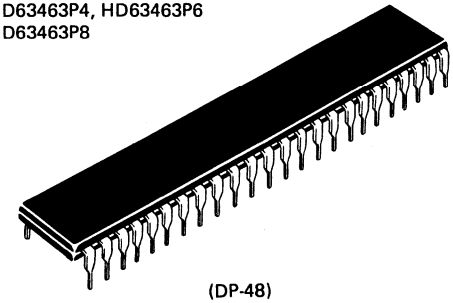
Type No.	Clock Frequency (MHz)		Package
	Host	Drive	
HD63463-4	4.0	10.0	DC-48
HD63463-6	6.0	15.0	
HD63463-8	8.0	20.0	
HD63463P4	4.0	10.0	DP-48
HD63463P6	6.0	15.0	
HD63463P8	8.0	20.0	
HD63463CP4	4.0	10.0	CP-52
HD63463CP6	6.0	15.0	
HD63463CP8	8.0	20.0	

HD63463-4, HD63463-6
HD63463-8



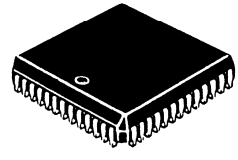
(DC-48)

HD63463P4, HD63463P6
HD63463P8



(DP-48)

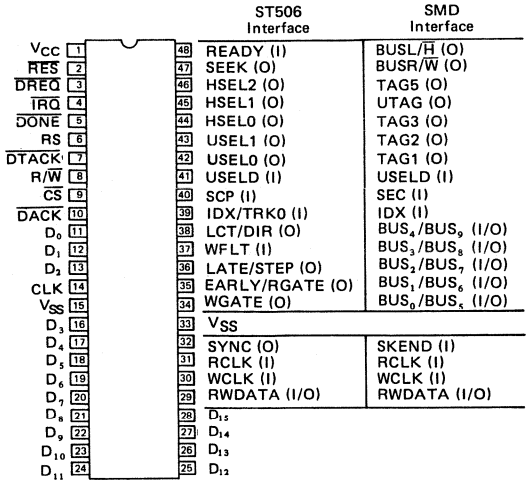
HD63463CP4, HD63463CP6
HD63463CP8



(CP-52)

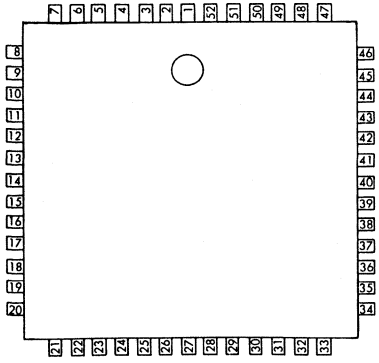
PIN ARRANGEMENT

DC-48, DP-48



(Top View)

CP-52



(Top View)

Pin No.	ST506 Interface	SMD Interface	Pin No.	ST506 Interface	SMD Interface
1	V_{CC}		27	V_{CC}	
2	V_{CC}		28	D_{12} (I/O)	
3	$\overline{RES}$ (I)		29	D_{13} (I/O)	
4	$\overline{DREQ}$ (O)		30	D_{14} (I/O)	
5	$\overline{IRQ}$ (O)		31	D_{15} (I/O)	
6	$\overline{DONE}$ (I)		32	RWDATA (I/O)	RWDATA (I/O)
7	RS (I)		33	WCLK (I)	WCLK (I)
8	$\overline{DTACK}$ (O)		34	RCLK (I)	RCLK (I)
9	R/$\overline{W}$ (I)		35	SYNC (O)	SKEND (I)
10	CS (I)		36	V_{SS}	
11	$\overline{DACK}$ (I)		37	V_{SS}	
12	D_0 (I/O)		38	WGATE (O)	BUS_0/BUS_5 (I/O)
13	D_1 (I/O)		39	EARLY/RGATE (O)	BUS_1/BUS_6 (I/O)
14	D_2 (I/O)		40	LATE/STEP (O)	BUS_2/BUS_7 (I/O)
15	CLK (I)		41	WFLT (I)	BUS_3/BUS_8 (I/O)
16	V_{SS}		42	LCT/DIR (O)	BUS_4/BUS_9 (I/O)
17	V_{SS}		43	IDX/TRKO (I)	IDX (I)
18	D_3 (I/O)		44	SCP (I)	SEC (I)
19	D_4 (I/O)		45	USELD (I)	USELD (I)
20	D_5 (I/O)		46	USEL0 (O)	TAG1 (O)
21	D_6 (I/O)		47	USEL1 (O)	TAG2 (O)
22	D_7 (I/O)		48	HSEL0 (O)	TAG3 (O)
23	D_8 (I/O)		49	HSEL1 (O)	UTAG (O)
24	D_9 (I/O)		50	HSEL2 (O)	TAG5 (O)
25	D_{10} (I/O)		51	SEEK (O)	BUSR/$\overline{W}$ (O)
26	D_{11} (I/O)		52	READY (I)	BUSL/$\overline{H}$ (O)

SIGNAL LINES

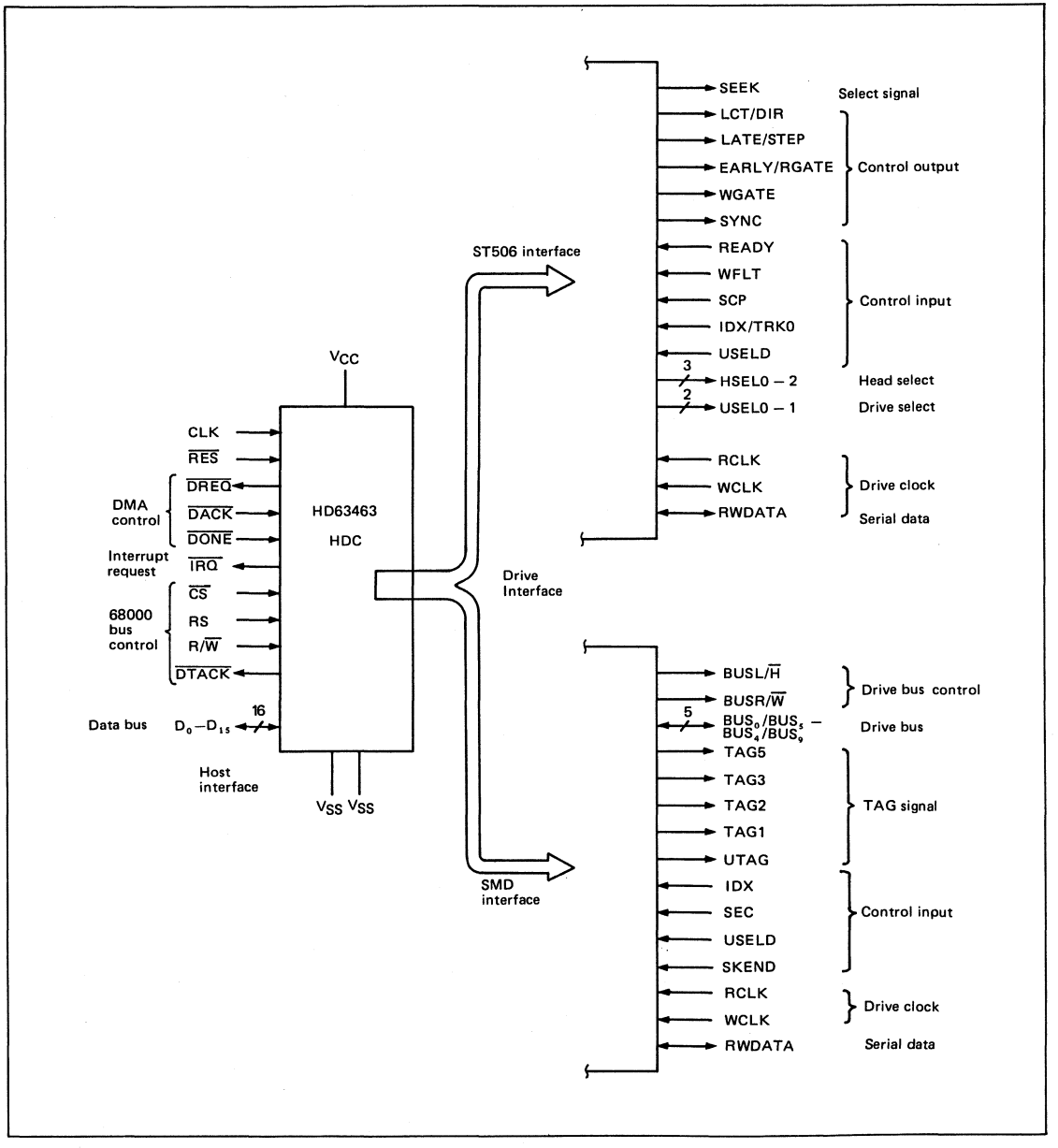


Figure 1 Input/Output Signals

MAJOR FUNCTIONS OF HDC

Drive Interface	ST506*	Number of Heads	Max 8
		Number of Drives	Max 4
		Write Precompensation	Controlled
		High-Speed Seek	Step rate min 875 ns (CLK: 8 MHz)
	Storage Module Drive (SMD)	Number of Heads	Max 32
		Number of Drives	Max 8
		Track Offset	Controlled
		Data Strobe	Controlled
	Serial Data Format	NRZ or MFM (selectable)	
	Serial Data Transfer Rate	Max 20 Mbps (NRZ), max 10 Mbps (MFM)	
	Multiple Sector Access	Multiple sector, multiple track	
	Parallel Seek	Possible	
	Diagnosis	External circuit diagnosis	
Disk Format	Data Length	256, 512, 1024, 2048, or 4096 bytes (selectable)	
	Number of Sectors	Max 255 (hard sector), max 128 (soft sector)	
	Number of Cylinders	Max 1024	
	Sector Format	Hard sector or soft sector (selectable)	
Error Processing	Processing Code	16-bit CRC (error detection), 2 types	
		32-bit ECC (error detection or correction)	
	Error Correction Capability	Automatic correction of single burst error (up to 11 bits)	
On-Chip Data Buffer	Capacity	256 bytes x 2	
	Data Transfer	Simultaneous transfer on host side and drive side	
	Addressing	Stack type (pointer can be updated)	
Host Interface	Operation Cycle Time	Max 8 MHz	
	Data Bus Width	16 or 8 bits (selectable)	
	DMA Transfer	Burst or cycle steal mode (selectable)	
	Programmed I/O	Possible	
Commands	Seek	2 commands	
	Disk Read/Write	8 commands	
	Data Buffer	4 commands	
	Drive Check	2 commands	
	Others	6 commands	

*ST506/ST412/ST412HP is referred to as "ST506" in this document.

HARDWARE DESCRIPTION

SYSTEM CONFIGURATION

An HDC-based system configuration is illustrated in figure 2. The HDC is used to connect more than one hard disk drives (also called "drive" in this manual) to the host system. The host system consists of the MPU (Microprocessing Unit), the main memory and the DMAC (Direct Memory Access Controller). The HDC requires minimum number of external circuits that are connected to the host system including the HD68000 (MPU) and the HD68450 (DMAC).

If the host system does not require a high throughput, then the DMAC is not necessary. A few additional external circuits allow the HDC to be connected to another host system other than the HD68000.

The HDC may control either up to four ST506 drives or up to eight SMD drives.

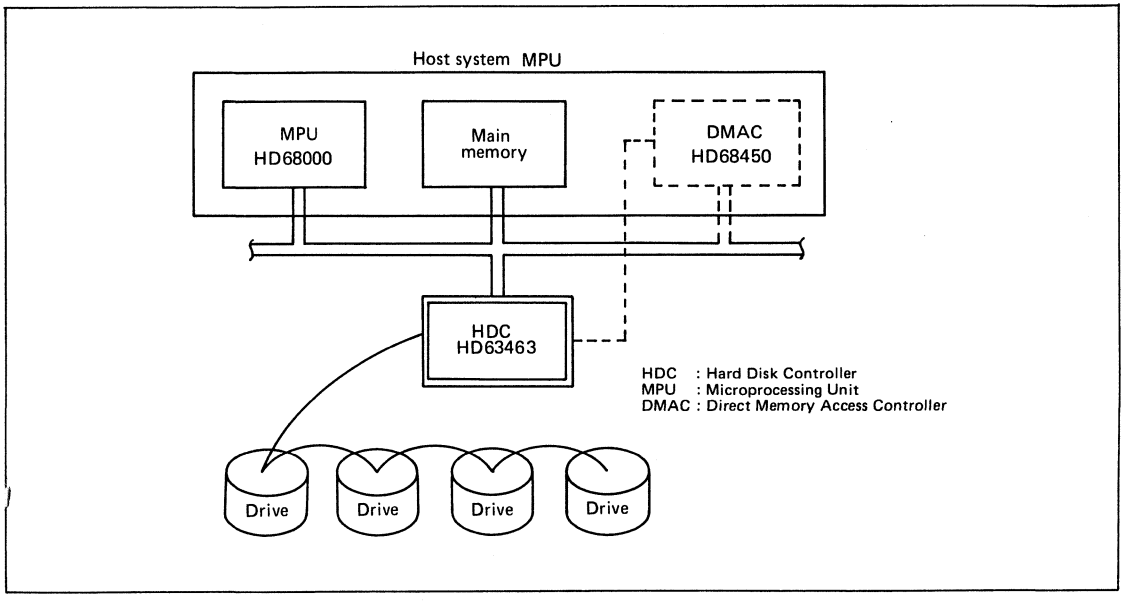


Figure 2 System Configuration Example

SIGNAL LINES

The input/output signal lines of the HDC are shown in figure 1 which covers both ST506 and SMD drive interface.

Each signal is described in the following tables and descriptions.

Host Interface

Category	Signal	I/O
Clock, Reset	CLK	I
	$\overline{RES}$	I
DMA Control (HD68450 Compatible)	$\overline{DREQ}$	O
	$\overline{DACK}$	I
	$\overline{DONE}$	I
Interrupt Request	$\overline{IRQ}$	O
Bus Control (HD68000 Compatible)	$\overline{CS}$	I
	RS	I
	R/$\overline{W}$	I
	$\overline{DTACK}$	O
Data Bus (HD68000 Compatible)	$D_0 - D_{15}$	I/O

CLK – Clock signal from the host system.

$\overline{RES}$ – Reset signal from the host system.

$\overline{DREQ}$ – Asserted when a DMA data transfer is requested to the DMAC. Receiving this signal, the DMAC obtains the bus mastership from the MPU.

$\overline{DACK}$ – Transfer acknowledge signal from the DMAC. Receiving this signal, the HDC transfers data through the host bus in the DMA mode. When $\overline{RES}$ signal is at "0", HDC data bus configuration change from 16 bits to 8 bits when $\overline{DACK}$ = 0.

$\overline{DONE}$ – Receiving this signal from the DMAC, the DMA data transfer through the host bus stops immediately.

$\overline{IRQ}$ – Interrupt request signal sent to the MPU (open drain output).

$\overline{CS}$ – Chip select signal generated by decoding address, address strobe, and data strobe of the host.

RS – Register select signal of the HDC. It selects registers when set to "0", data buffers when set to "1".

R/$\overline{W}$ – Signal that indicates the data transfer direction.

$\overline{DTACK}$ – Acknowledge signal that indicates the end of data transfer in the host bus. The host waits until the HDC asserts this signal to terminate the transfer.

D_0–D_{15} – 16-bit bi-directional data bus. When used as an 8-bit data bus, D_8–D_{15} must be open.

ST506 Interface

Category	Signal	I/O
Select Signal	SEEK	O
Control Output	LCT/DIR	O
	LATE/STEP	O
	EARLY/RGATE	O
	WGATE	O
	SYNC	O
	HSEL0–HSEL2	O
	USEL0–USEL1	O
Control Input	READY	I
	WFLT	I
	SCP	I
	IDX/TRK0	I
	USELD	I
Drive Clock	RCLK	I
	WCLK	I
Drive Data	RWDATA	I/O

SEEK – Used to decode multiplexed signals. SEEK signal remains high during the head positioning operation of the HDC.

LCT/DIR – When SEEK is at "1", this signal indicates the head direction: the head moves toward the spindle if the LCT/DIR signal is at "1". When SEEK is at "0", write current is reduced if this signal is at "1".

LATE/STEP – When SEEK is at "1", this signal serves as step pulses output for head positioning. When SEEK is at "0", this signal requests write precompensation (LATE).

EARLY/RGATE – When the WGATE signal is at "1", this signal requires write precompensation (EARLY). Otherwise, it serves as a read gate signal that requests reading data from the drive.

WGATE – Write gate signal that requests writing data to the drive.

SYNC – Signal to select the loop gain of the external data separator. SYNC is at "1" when RGATE is at "1" and one byte of $00 is detected in the disk data. SYNC is at "0" when RGATE is at "0".

HSEL0–HSEL2 – 3-bit signal that selects up to eight heads in the drive.

USEL0- USEL1 – 2-bit signal that selects one of four drives.

READY – Signal indicating that the selected drive is ready.

WFLT – Signal indicating that the drive has detected a fault which may cause a write error.

SCP – Signal indicating that head positioning (seek operation) has been completed in the drive.

IDX/TRK0 – When SEEK is at "1", this signal indicates that the head in the drive is positioned at the outermost track (track 0). When SEEK is at "0", this signal serves as an index signal that indicates the beginning of a track.

USELD – Response signal from the drive indicating that the drive is selected.

RCLK – Disk read clock from the external data separator. Its frequency is twice the serial transfer rate.

WCLK – Disk write clock from the external oscillator. Its frequency is twice the serial transfer rate.

RWDATA – Disk read/write serial data is input/output as modified frequency modulation code. This pin is normally in read state.

SMD Interface

Category	Signal	I/O
Drive Bus Control	BUSL/$\overline{H}$	O
	BUSR/$\overline{W}$	O
Drive Bus	BUS_0/BUS_5 – BUS_4/BUS_9	I/O
Tag	TAG5	O
	TAG3	O
	TAG2	O
	TAG1	O
	UTAG	O
Control Input	IDX	I
	SEC	I
	USELD	I
	SKEND	I
Drive Clock	RCLK	I
	WCLK	I
Drive Data	RWDATA	I/O

BUSL/$\overline{H}$ – When this signal is at high, lower 5 bits of 10-bit drive bus are being transferred.

BUSR/$\overline{W}$ – When this signal is at high, BUS_0/BUS_5 – BUS_4/BUS_9 are in input state.

BUS_0/BUS_5 – BUS_4/BUS_9 – 5-bit bi-directional bus that is used to control the drive. When the data is output from 10-bit drive bus, the high-order 5 bits are output first and then the low-order 5 bits. When the data is input to the bus, higher 3 bits of the 8-bit status are fetched first, then the lower 5 bits.

TAG5 – Together with TAG2, used as a status input select signal.

TAG3 – Tag signal indicating that the drive bus contains information such as disk read/write instruction.

TAG2 – Tag signal indicating that the drive bus contains the address for head selection. When drive status is read from the drive bus, TAG2 and TAG5 are decoded to select the status.

TAG1 – Tag signal indicating that the drive bus contains the destination cylinder address.

UTAG – Tag signal indicating that the drive bus contains a 3-bit drive number.

IDX – Index signal indicating the beginning of a track.

SEC – Signal indicating the beginning of a sector.

USELD – Response signal from the drive indicating that the drive is selected.

SKEND – Signal that is generated when the servo circuit in the disk drive terminates seek operation and the head is placed on the track.

RCLK – Disk read clock from the drive. Its frequency is the same as the serial data transfer rate.

WCLK – Disk write clock from the drive. Its frequency is the same as the serial data transfer rate.

RWDATA – Disk read/write serial data that is input/output as Non-Return to Zero code.

HOST INTERFACE

The HDC can be directly connected to the HD68000 (MPU) and the HD68450 (DMAC), so that the HDC data is transferred asynchronously with the host system. The HDC data bus is 16-bit wide, in addition, an 8-bit configuration is available by asserting the $\overline{DACK}$ signal externally during reset. The minimum access time that is required for communication between the HDC and the MPU, or between the HDC and the DMAC is listed in Table 1 (wait cycles are not included).

Table 1 Host Interface Minimum Access Time

Transfer Direction \ Data Bus	8-Bit Data Bus (cycles)	16-Bit Data Bus (cycles)
MPU read (HDC to MPU)	3	4
MPU write (MPU to HDC)	3	3
DMA write (memory to HDC)	3	3
DMA read (HDC to memory)	3	4

DRIVE INTERFACE

The HDC can interface with two different types of drives. DIF mode of the Operation Mode register 0 (OM0) selects the drive interface. ST506 interface is selected when DIF is at "0", and SMD interface when DIF is at "1". The external circuits required for HDC drive interface vary according to which interface is used. Figures 3 and 4 show the interface circuits which connect ST506 device and SMD device respectively.

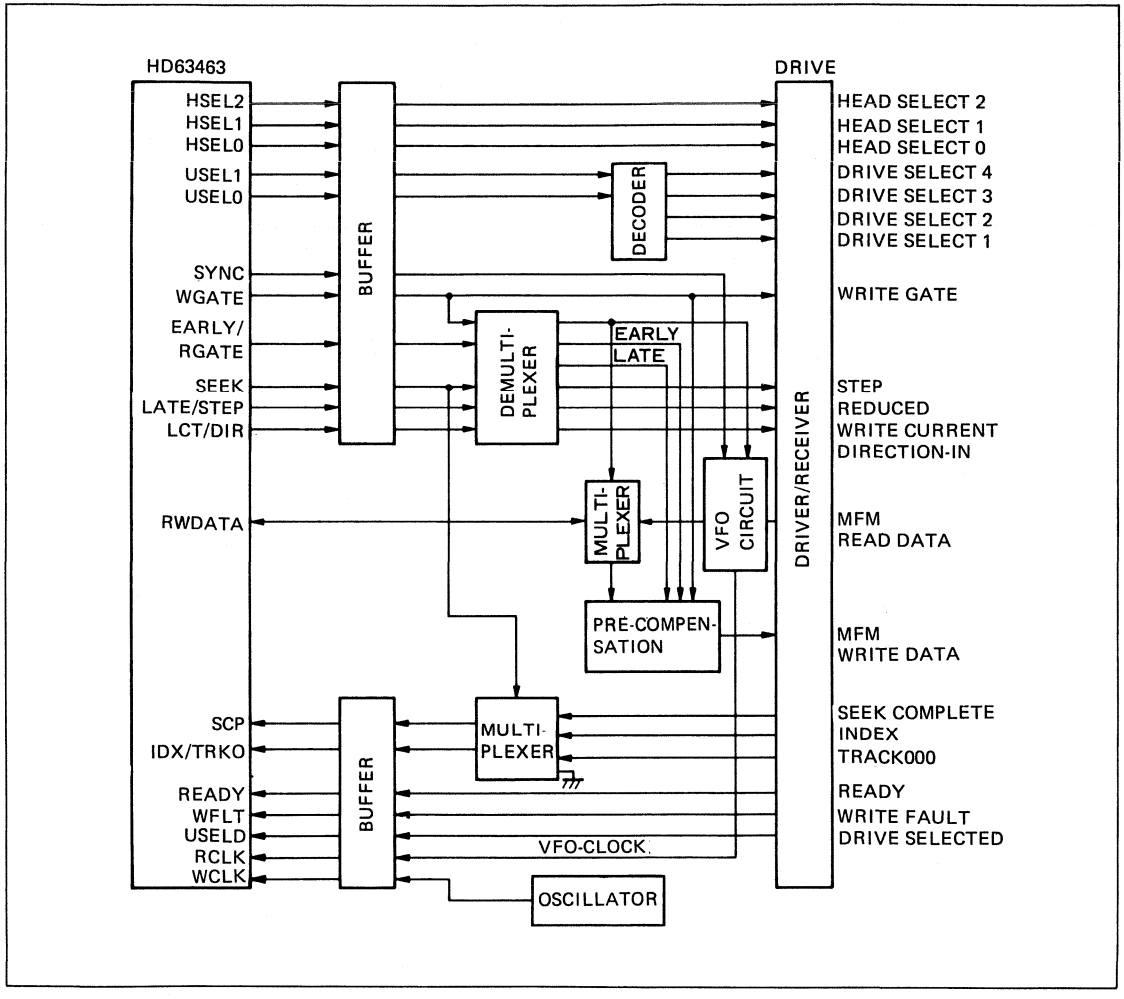


Figure 3 ST506 Interface Circuit

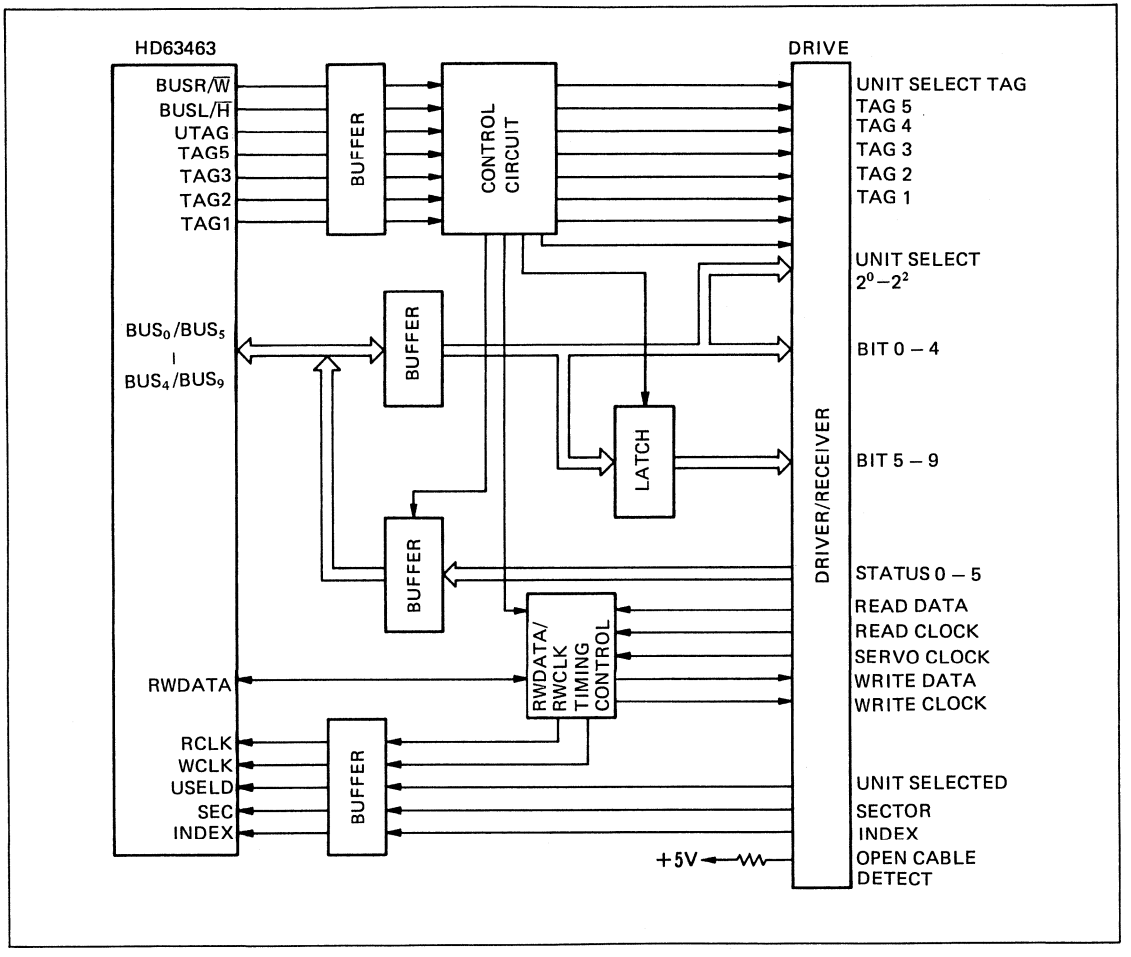


Figure 4 SMD Interface Circuit

SOFTWARE INTERFACE

The HDC is furnished with 22 commands, which are classified into 6 categories; specification, head positioning, disk access, data transfer, drive check, and others.

Table 2 List of Commands

Category	Command	Mnemonic
Specifications	Specify	SPC
Head Positiong	Recalibrate	RCLB
	Seek	SEK
Disk Access	Read Data	RD
	Read Erroneous Data	RED
	Read ID	RID
	Read ID Skew (Note 1)	RIS
	Find ID	FID
	Check Data	CKD
	Compare Data	CMPD
	Write Data	WD
	Write Format	WFM
	Write Format Skew (Note 1)	WFS
Data Transfer	Memory to Buffer	MTB
	Buffer to Memory	BTM
	Open Buffer Write	OPBW
	Open Buffer Read	OPBR
Drive Check	Polling	POL
	Check Drive	CKV
Others	Abort	ABT
	Check ECC	CKE
	Polling Disable	POD
	Recall	RCAL
	Test	TST

(Note 1) Valid only for the hard sector. Disabled for the soft sector.

Specify – Specifies the HDC's operation mode, data transfer mode, etc.

Recalibrate – Moves the drive head to the outermost track (track 0).

Seek – Moves the drive head onto a track specified by the Next Cylinder Address.

Read Data – Reads the data of specified sectors and stores it in the data buffer.

Read Erroneous Data – Reads disk data and stores it in the data buffer no matter a CRC error occurs or not in ID area.

Read ID – Reads ID areas from specified number of sectors.

Read ID Skew – Reads ID area of a sector formatted by Write Format Skew command.

Find ID – Reads ID areas and stores the data in data buffers. ID area containing a CRC error is skipped reading, and the subsequent ID area without any CRC error is searched for.

Check Data – Checks if there is any ECC or CRC error in DATA area of specified sectors. No data is transferred to data buffers or to the main memory while checking.

Compare Data – Compares the data in data buffers and the data read from specified sectors.

Write Data – Writes the data stored in data buffers into DATA area of specified sectors.

Write Format – For the hard sector, formats a specified number of sectors starting with a specified physical sector address. For the soft sector, formats a track.

Write Format Skew – Formats a sector specified by a physical sector address by skewing ID area by 64 bytes.

Memory to Buffer – Transfers data from the main memory to the data buffer by DMA transfer.

Buffer to Memory – Transfers data stored in the data buffer to the main memory by DMA transfer.

Open Buffer Write – Provides initialization of pointer to write data into the data buffer starting from an address specified by Pointer Offset. The data buffer is written in the PIO mode after the command execution ends.

Open Buffer Read – Provides initialization to read the data stored in the data buffer from an address specified by Pointer Offset. The data buffer is read in the PIO mode after the command execution ends.

Polling – Monitors drive status including seek end.

Check Drive – Sets result parameters indicating status of a specified drive to Parameter Block.

Abort – Stops all operations being executed by the HDC.

Check ECC – For ECC errors occurred during RD and RED command execution, reports result parameters indicating addresses and patterns of erroneous data.

Polling Disable – Stops Polling command execution.

Recall – Clears all bits of the status register and sets a buffer pointer to the start address of parameter block so that command parameters can be accepted.

Test – Makes the output pins of the HDC's drive interface three-stated.

Table 3 Command Code, Interrupt Request, and DMA Data Transfer

Command Name	Command Code		Interrupt Factor Bits			DMA Transfer	
	Binary	HEX	CED	SED	DER	DTM = 0	DTM = 1
Specify	1110 1000	E8					
Recalibrate	1100 1000	C8	o	o	o		
Seek	1100 0000	C0	o	o	o		
Read Data	0100 0000	40	o	o	o		o
Read Erroneous Data	0111 0000	70	o	o	o		o
Read ID	0110 0000	60	o	o	o		o
Read ID Skew	0110 1000	68					
Find ID	0110 0001	61	o	o	o		o
Check Data	0100 1000	48	o	o	o		
Compare Data	1000 1000	88	o	o	o		o
Write Data	1000 0111	87	o	o	o		o
Write Format	1010 0011	A3	o	o	o		o
Write Format Skew	1010 1011	AB					
Memory to Buffer	1001 0000	90	o			o	o
Buffer to Memory	0101 0000	50	o			o	o
Open Buffer Read	0011 0000	30					
Open Buffer Write	0011 1000	38					
Polling	0001 0000	10		o	o		
Check Drive	0010 1000	28			o		
Abort	1111 ****	F0-FF	o				
Check ECC	0010 0000	20					
Test	1110 0000	E0					
Polling Disable	0001 1000	18		o	o		
Recall	0000 1000	08					
Inhibited	1101 ****	D0-DF					

o: Set at the end of command execution.

o: DMA transfer is performed.

DMA: Direct Memory Access
DTM: Data Transfer Mode
*: don't care

PROGRAMMING MODEL

Figure 5 shows the internal configuration of the HDC and a programming model. The HDC internally provides an 8-bit STR (Status register), an 8-bit CMR (Command register), a 16-byte PB (Parameter Block) and two 256-byte data buffers (DBUF0 and DBUF1). The data can be written to/read from PB, DBUF0 and DBUF1 by externally accessing DTR (Data Transfer register). Their address are specified by a pointer, and is incremented for each access. Internal processors set pointer value, or select one of PB, DBUF0 and DBUF1 to be connected to DTR. These internal processors are initialized by writing a command from the MPU to CMR. The result of command execution is reflected in the STR.

At the beginning of the command execution, internal processors read command parameters in PB written by the MPU and determine the command operation mode. At the end of the command execution, the MPU reads result parameters in PB written by internal processors.

When the HDC writes data to a disk, the host system sends disk write data to DBUF, then internal processors write the contents of DBUF into the disk. When the HDC reads data from a disk, internal processors read the contents of the disk and stores them into DBUF, then the host system reads data from DBUF. The MPU can access STR, CMR, and DTR while the DMAC can access only DTR in the single addressing mode.

When used in 8-bit mode, the HDC requires 2-byte address space. One byte (RS (Resister Select signal) = 0) is an 8-bit read only STR or an 8-bit write only CMR. Another byte (RS = 1) is an 8-bit DTR.

When used in a 16-bit mode, the HDC requires 2-word address space. One word (RS = 0) consists of an 8-bit CMR and an 8-bit read only STR. Another word (RS = 1) serves as a 16-bit DTR.

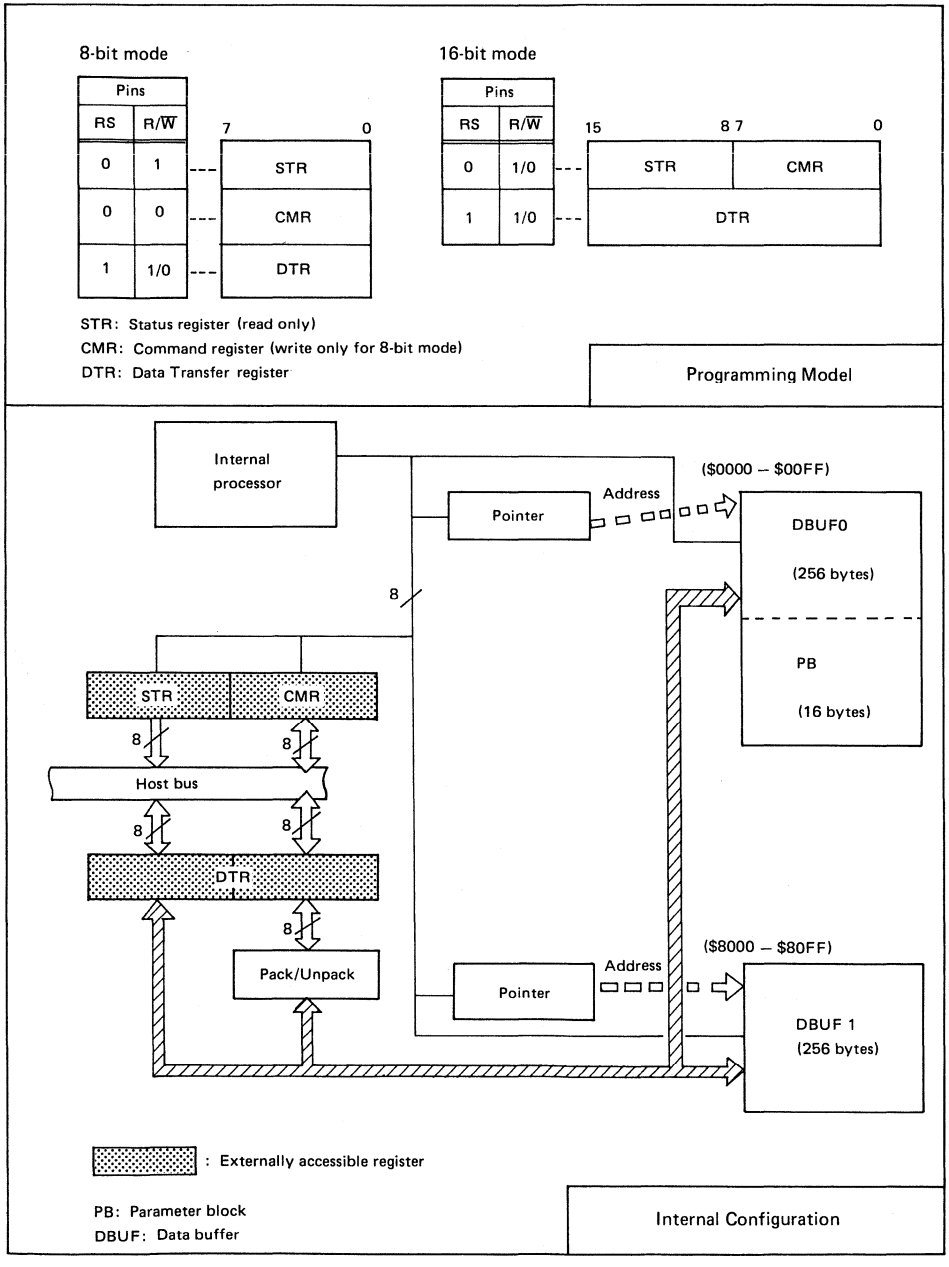


Figure 5 Internal Configuration and Programming Model

When the HDC is either in idle state or command execution end state, the MPU can access PB by accessing DTR. While the HDC is executing a command (during data transfer), DBUF0 or DBUF1 can be accessed by accessing DTR. Before accessing DBUF0 or DBUF1, the MPU must issue a command to open the buffer.

STATUS REGISTER

The bit configuration of the Status register is shown in the figure below, and each bit is described in Table 4.

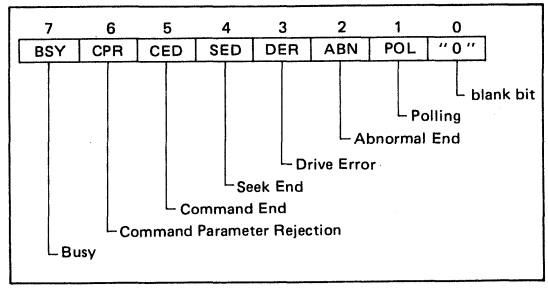


Table 4 Status Register Bit Description

Bit	Abbr.	Bit Name	Set "1" Condition	Reset "0" Condition	Interrupt Source Bit	Description
7	BSY	Busy	Command acknowledged	Command execution end (except for POL command)		While HDC is executing a command, BSY bit is set to "1".
6	CPR	Command Parameter Rejection	Command execution	RCAL command received		With this bit reset to "0", command parameters can be written.
5	CED	Command End	Command execution end	RCAL command received	○	Set to "1" when command shown in table 3 ends. When this bit is set, HDC asserts $\overline{IRQ}$ signal.
4	SED	Seek End	Drive seek end detected	RCAL command received	○	If detecting drive seek end during execution of command shown in table 3, HDC sets "1" in this bit at the end of command execution.
3	DER	Drive Error	Drive error detected	RCAL command received	○	If detecting drive error during execution of command shown in table 3, HDC sets "1" in this bit at the end of command execution.
2	ABN	Abnormal End	Error detected	RCAL command received		If acknowledging errors such as illegal command, drive faults, data over/under run, the HDC sets "1" to ABN at the end of command execution. The SSB contains the error code.
1	POL	Polling	POL command received	POD command received, seek operation end detected, drive error detected		Set to "1" during execution of POL command.
0						Blank bit, always set to "0".

○: An interrupt is generated when this bit is set.

HDC CONTROL PROCEDURE

The MPU's procedure to control the HDC is shown in figure 6. To control the HDC, the MPU must read STR of the HDC. The MPU may issue command parameters to PB in the HDC only when both BSY and CPR bits are cleared.

This enables the HDC to change its status from idle state to command wait state. Although a data buffer pointer does not indicate the start address of PB under command wait state, STR has the same contents as in the idle state. It is impossible to distinguish these two internal HDC states externally.

After this, the MPU issues a command. The HDC executes a command after setting BSY bit to "1". At the end of command execution, result parameters are stored in PB, the CPR bit is set to "1", and BSY bit is cleared. Under this condition, the MPU reads result parameters from PB. Issuing Recall command after this enables the HDC state to change from command execution end state to idle state.

Some commands do not require either or both of the command parameter and result parameter. For some commands, the CED bit, SED bit, or DER bit is set to "1", which enables the HDC to generate an interrupt request to the MPU.

In the DMA (direct memory access) mode, data transfer takes place between the main memory and the HDC when a disk access command is received. For this reason, the MPU must initialize the DMAC before issuing a disk access command to the HDC. Data such as system memory address and number of transfer words are written into the DMAC register.

When the MPU writes data into internal control register in the DMAC, the DMAC state changes from the idle state to transfer request wait state. Upon reception of a transfer request signal ($\overline{DREQ}$) from the HDC, the DMAC transfers data to/from the HDC and system memory until specified number of words are transferred. The DMAC may generate an interrupt request to the MPU when the transfer is completed.

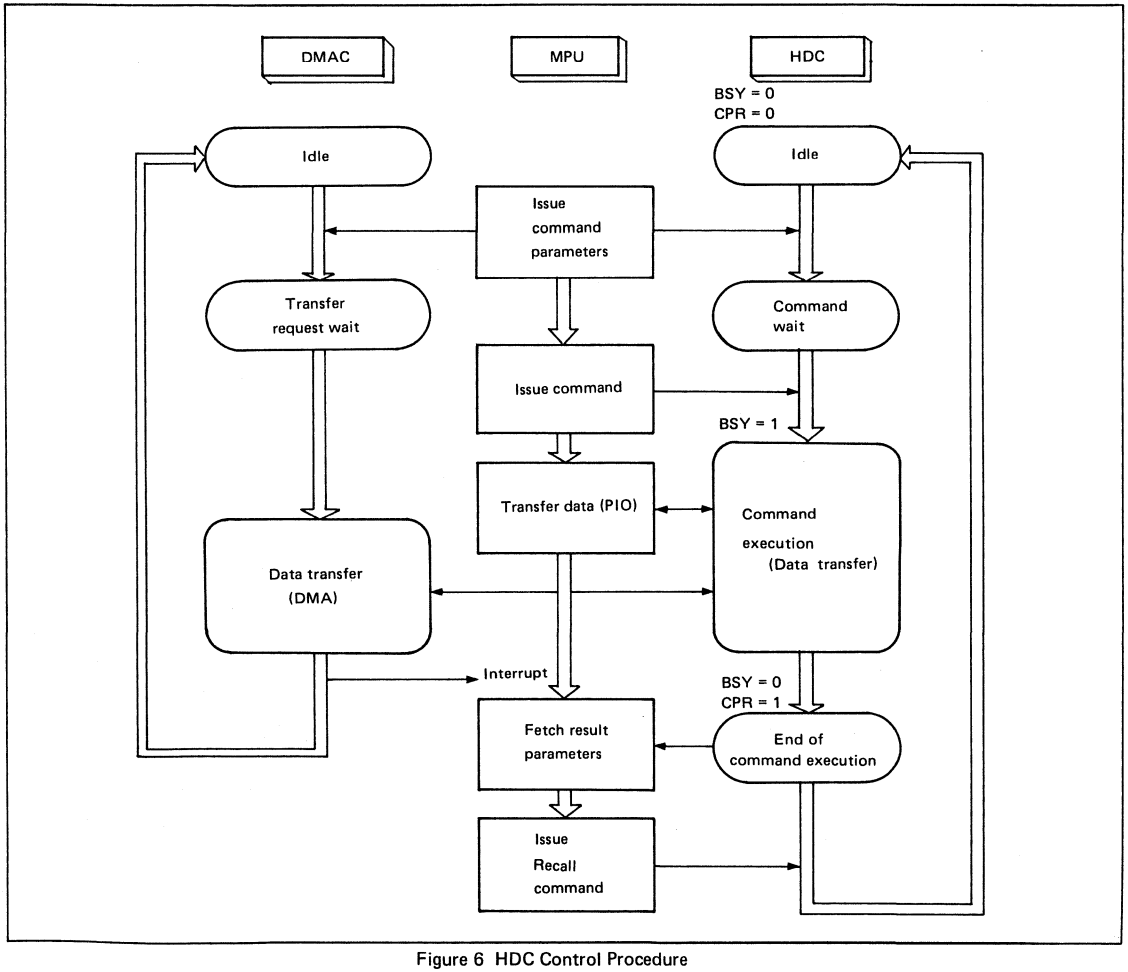


Figure 6 HDC Control Procedure

COMMAND PARAMETERS

Command parameters are listed in Table 5. Command parameters used by the HDC to control ST506 type hard disk drive are listed in the upper row of each command, and those for the SMD type in the lower row.

Table 5 Command Parameter (byte organization)

Commands	Parameters (Upper row: ST506 / Lower row: SMD)
Specify	OM0 OM1 OM2 CUL TO/NCH NCL NH NS SH/RL GPL1 GPL2 GPL3 LCCH LCCL PCCH PCCL OM0 OM1 OM2 CUL TO/NCH NCL NH NS SH/RL GPL1 GPL2 RGTLT
Recalibrate	US $00 US $00
Seek	US $00 NCAH NCAL US $00 NCAH NCAL
Read Data	US PHA LCAH LCAL LHA LSA SCNTH SCNTL US PHA $00 FLAG LCAH LCAL LHA LSA SCNTH SCNTL
Read Erroneous Data	US PHA SCNTH SCNTL US PHA $00 PSA SCNTH SCNTL
Read ID Read ID Skew (Note 1)	US PHA $00 OFFSET $00 SCNTL US PHA $00 PSA $00 SCNTL
Find ID	US PHA $00 OFFSET $00 SCNTL US PHA $00 PSA $00 SCNTL
Check Data	US PHA LCAH LCAL LHA LSA SCNTH SCNTL US PHA $00 FLAG LCAH LCAL LHA LSA SCNTH SCNTL
Compare Data	US PHA LCAH LCAL LHA LSA SCNTH SCNTL US PHA $00 FLAG LCAH LCAL LHA LSA SCNTH SCNTL
Write Data	US PHA LCAH LCAL LHA LSA SCNTH SCNTL US PHA $00 FLAG LCAH LCAL LHA LSA SCNTH SCNTL
Write Format Write Format Skew (Note 1)	US PHA SCNTH SCNTL US PHA $00 PSA SCNTH SCNTL
Memory to Buffer	POFFH POFFL POFFH POFFL
Buffer to Memory	POFFH POFFL POFFH POFFL
Open Buffer Read	POFFH POFFL POFFH POFFL
Open Buffer Write	POFFH POFFL POFFH POFFL
Polling	None
Check Drive	US $00 US $00
Abort	None
Check ECC	None
Test	None
Polling Disable	None
Recall	None

(Note 1) Read ID Skew and Write Format Skew are valid only for SMD interface.

SPECIFY COMMAND PARAMETERS

Parameters of SPC command may be used to specify the HDC operation mode and the disk format. Parameters are listed in table 6. Contents of these parameters are different according to which type of drive is used: either ST506 or SMD. Each parameter is described in tables 7 and 8. Table 7 lists parameters which are specified by a bit, and table 8 lists parameters which are specified by 3–16 bits.

The HDC supports soft sector format (in ST506 interface) and hard sector format (in SMD interface) which are shown in figure 7.

Table 6 Specify Command Parameter Organization

	Bit 7	6	5	4	3	2	1	0
OM0	SECT	MOD	DIF	PADP	ECD	CRCP	CRCI	ACOR
OM1	DTM	BRST	CEDM	SEDM	DERM	0	AMEX	PSK
OM2	SL (ST506)							
	CUL							
	TO						NCH	
	NCL							
	NH							
	NS							
	SH					RL		
	GPL1							
	GPL2							
	GPL3							
	LCC High (LCCH)							
	LCC Low (LCCL)							
	PCC High (PCCH)							
	PCC Low (PCCL)							
(SMD)								
OM2	0	0	0	0	SOFM	SOFP	STBL	STBE

Table 7 Specify Command Parameters (a)

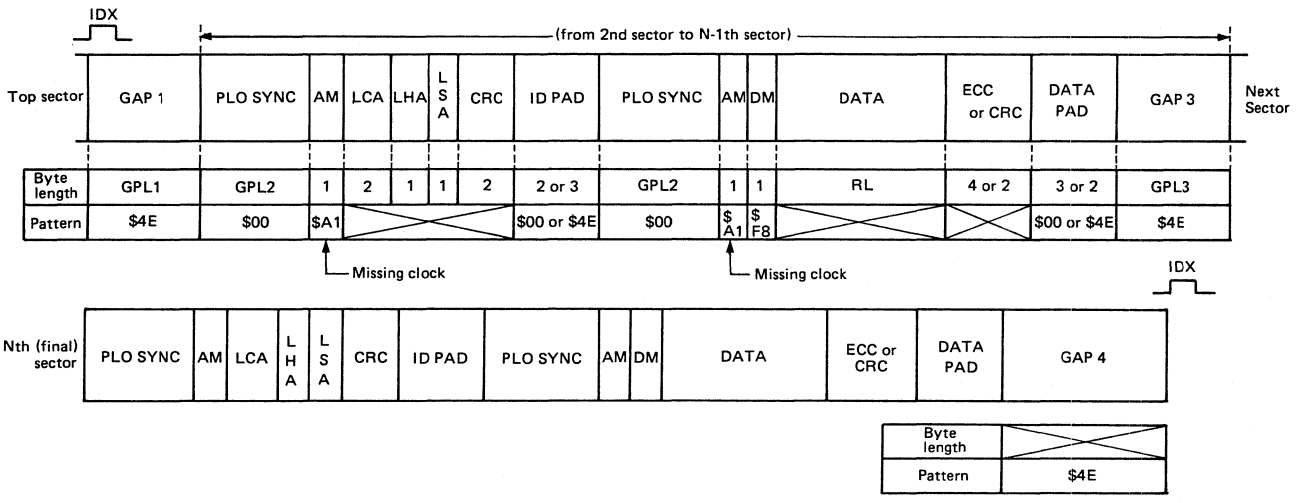
Abbreviation	Name	0	1	ST506	SMD
SECT	Sector Organization	Soft Sector	Hard Sector	0	1
MOD	Data Modulation	MFM	NRZ	0	1
DIF	Drive Interface	ST506	SMD	0	1
PADP	PAD Pattern	$00	$4E	*	0
ECD	Error Check Code	CRC	ECC	*	*
CRCP	CRC Polynomial	$X^{16} + 1$	$X^{16} + X^{12} + X^5 + 1$	*	*
CRCI	CRC Initial Value	$0000	$FFFF	*	*
ACOR	Automatic Correction	Disabled	Enabled	*	*
DTM	Data Transfer Mode	PIO	DMA	*	*
BRST	DMA Burst Mode	Cycle Steal	Burst	*	*
CEDM	Command End Mask	Unmasked	Masked	*	*
SEDM	Seek End Mask	Unmasked	Masked	*	*
DERM	Drive Error Mask	Unmasked	Masked	*	*
AMEX	Address Mark Exclude	Included	Not Included	*	*
PSK	Parallel Seek	Normal	Parallel	*	1
SOFM	Servo Offset Minus	Normal	Minus	None	*
SOFP	Servo Offset Plus	Normal	Plus	None	*
STBL	Strobe Late	Normal	Late	None	*
STBE	Strobe Early	Normal	Early	None	*

* Either 0 or 1

Table 8 Specify Command Parameters (b)

Abbreviation	Name	ST506	SMD
SL	Step Pulse Low	8 bits	None
CUL	Connecting Unit List	4 bits	8 bits
TO	Read/Write Time-over	6 bits	6 bits
NC	Number of Cylinders	10 bits	10 bits
NH	Number of Heads	3 bits	5 bits
NS	Number of Sectors	8 bits	8 bits
SH	Step Pulse High	5 bits	None
RL	Record Length	3 bits	3 bits
GPL1	Gap Length 1 (8 bits)	GAP1	HEAD SCAT
GPL2	Gap Length 2 (8 bits)	PLO SYNC	PLO SYNC
GPL3	Gap Length 3 (8 bits)	GAP3	RGATE Latency
LCC	Low Current Cylinder	16 bits	None
PCC	Precompensation Cylinder	16 bits	None

Soft Sector Format Supported by HDC



Hard Sector Format Supported by HDC

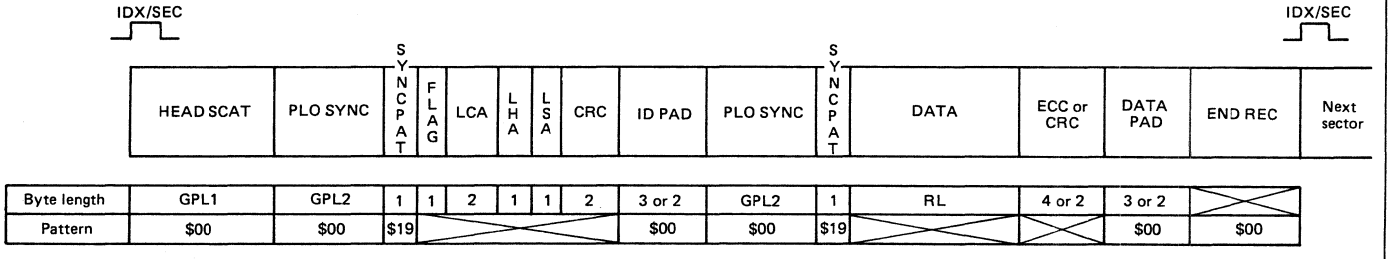


Figure 7 Disk Format

OM0 (Operation Mode 0)

(1) SECT (Sector Format) bit

This bit specifies the format of the drive to be connected to the HDC. There are two drive formats available: hard sector and soft sector.

SECT = 1: Hard Sector Format
SECT = 0: Soft Sector Format

(2) MOD (Modulation) bit

This bit specifies modulation mode for data written to/read from the drive.

MOD = 1: NRZ (Non Return to Zero)
MOD = 0: MEM (Modified FM)

(3) DIF (Drive Interface) bit

This bit specifies the type of drive interface: either ST506 or SMD. The pin function of the HDC changes according to the interface type.

DIF = 1: SMD Interface
DIF = 0: ST506 Interface

In SMD interface, the HDC performs seek instruction, head specification, drive status check, etc to the drive through 5-bit bi-directional buffer BUS_0/BUS_5–BUS_4/BUS_9.

In ST506 interface, the HDC makes the drive perform seek operation by issuing step pulses.

(4) PADP (PAD Pattern) bit

This bit specifies the data pattern of PAD area that follows ID and DATA areas. The value of the PADP bit gives the data pattern of PAD area that is written into the drive by WFM or WD command execution.

	Hard Sector	Soft Sector
PADP = 1	Prohibited	$4E
PADP = 0	$00	$00

(5) ECD (Error Check Code) bit

This bit specifies the error check code which is added to the end of the DATA area. CRC code is always specified for the ID area regardless of ECD bit.

ECD = 1: ECC (Error Correction Code)
ECD = 0: CRC (Cyclic Redundancy Check Code)

ECC enables error detection and correction. A 4-byte ECC code is added to the end of DATA area. A generation polynominal G(x) of ECC is as follows (the initial value is fixed to "00").

$$G(x) = (x^{21} + 1)(x^{11} + x^2 + 1)$$
$$= x^{32} + x^{23} + x^{21} + x^{11} + x^2 + 1$$

CRC enables error detection, but not error correction. A 2-byte CRC code is added to the end of DATA area. A generation polynominal G(x) of CRC is specified by the CRCP bit.

(6) CRCP (CRC Polynominal) bit

This bit determines the polynominal G(x) that generates CRC of ID and DATA areas when ECD = 0.

CRCP = 1: $G(x) = x^{16} + x^{12} + x^5 + 1$
CRCP = 0: $G(x) = x^{16} + 1$

(7) CRCI (CRC Initial) bit

This bit sets the initial value of CRC.

CRCI = 1: Initial value = $FFFF
CRCI = 0: Initial value = $0000

(8) ACOR (Automatic Correction) bit

This bit selects whether or not the HDC will automatically correct an error detected in DATA area during RD command execution.

ACOR = 1: Automatic correction is performed.
ACOR = 0: Automatic correction is not performed.

Automatic correction mode is valid when ECC is specified as the error check code of DATA area and the sector length is 256 bytes (RL = $01, ECD = 1). For any other cases specify "0" to the ACOR bit.

OM1 (Operation Mode 1)

(1) DTM (Data Transfer Mode) bit

This bit is used to specify data transfer operation between the HDC and the main memory during the execution of the following commands:

RD	RED	RID
RIS	FID	WD
CMPD	WFM	WFS

These commands normally perform transfer between drive and memory via HDC data buffers. However, it is possible to cease transfer between the HDC and memory during these command execution by utilizing DTM bit.

DTM = 1: DMA mode
DTM = 0: PIO mode

In DMA mode, the HDC performs transfer between drive and memory via HDC data buffer. In this case, DMA transfer is performed between the HDC and memory, and transfer mode is specified by BRST bit.

In PIO mode, the HDC performs transfer between drive and the HDC. In this case, transfer between the HDC and main

memory must be supported by the host system using one of four buffer access commands of the HDC: MTB, BTM, OPBR, and OPBW.

(2) BRST (DMA Burst) bit
This bit specifies DMA transfer mode of buffer access commands and drive access commands which perform transfer between the HDC and the main memory.

BRST = 1: Burst mode
BRST = 0: Cycle Steal mode

(3) CEDM, SEDM, DERM bits
CEDM (Command End Mask), SEDM (Seek End Mask), and DERM (Drive Error Mask) bits specify whether the $\overline{IRQ}$ signal is to be asserted or not. CEDM, SEDM, and DERM correspond to CED (Command End), SED (Seek End), and DER (Drive Error) bits in STR (Status register).

1: $\overline{IRQ}$ is masked (not asserted).
0: $\overline{IRQ}$ is not masked (asserted).

Mask bit	Corresponding bits in STR	
CEDM	CED	bit 5
SEDM	SED	bit 4
DERM	DER	bit 3

(4) AMEX (Address Mark Excluded) bit
This bit specifies whether or not the byte-synchronization pattern marking the beginning of ID area or DATA area (AM in soft sector, SYNCPAT in hard sector) is to be included in the CRC or ECC error detection span. The AMEX bit affects the byte length of ID PAD area and DATA PAD area.

	AM or SYNCPAT	PAD Length
AMEX = 1	Excluded	2 bytes
AMEX = 0	Included	3 bytes

(5) PSK (Parallel Seek) bit
This bit specifies seek operation mode, and is valid only in ST506 interface (in SMD interface, PSK must be fixed to "1"). The HDC specifies step pulse issue timing by utilizing the value of OM2, SH, and PSK bits.

PSK = 0: Normal Seek mode
PSK = 1: Parallel Seek mode

In Normal Seek mode, the HDC issues step pulses in long cycle (0.1–32 ms). SEK and RCLB command execution ends when the HDC issues step pulses and then detects seek end.

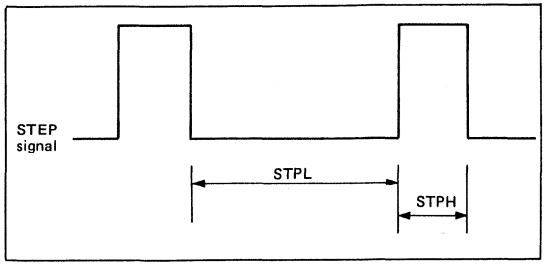
In Parallel Seek mode, the HDC issues step pulses in short cycle (0.5–115 μs). SEK and RCLB command execution ends when the HDC issues step pulses. Since the HDC does not check the seek end, parallel seek operation in multiple drives is realized by issuing SEK or RCLB command to these drives. Seek end is to be checked by using POL command.

OM2 (Operation Mode 2)
This 8-bit register specifies step pulse low width in ST506 interface, and specifies drive control output signal during disk read command execution in SMD interface.

In ST506 interface, OM2 indicates SL which specifies step pulse low width (STPL: Step Pulse Low). Low-speed seek mode (Normal Seek mode) is selected when PSK = 0, and high-speed seek mode (Parallel Seek mode) is selected when PSK = 1. Highest seek speed is realized when PSK = 1 and SL = $FF. The relation between step pulse low width and SL is shown in the following table (see Step Pulse High/Record Length register to specify step pulse high width).

7	6	5	4	3	2	1	0
Step Pulse Low (SL)							

Seek Mode	SL	STPL (step pulse low)
Normal seek (PSK = 0)	SL = $00	STPL = 988 CLK
	$01 ≦ SL ≦ $FE (1) (254)	STPL = (SL − 1) x 1280 + 2364 CLK (2364 CLK ≦ STPL ≦ 326204 CLK)
	SL = $FF	Disabled
Parallel seek (PSK = 1)	SL = $00	STPL = 27CLK
	$01 ≦ SL ≦ $FE (1) (254)	STPL = SL x 6 + 28 CLK (34CLK ≦ STPL ≦ 1152 CLK)
	SL = $FF	STPL = 5CLK



In SMD interface, OM2 specifies drive control output signal from the HDC during data read command execution. In data write command (all the commands that assert WGATE signal) and SEK, RCLB command execution, outputs of drive control signals are low regardless of the contents of OM2. Bits 4–7 in OM2 are to be set to "0".

SOFM bit: With SOFM = 1, the drive head is offset from the normal position away from the spindle.
SOFP bit: With SOFP = 1, the drive head is offset from the normal position towards the spindle.
STBL bit: With STBL = 1, the data from the drive PLO data separator is strobed later than usual.
STBE bit: With STBE = 1, the data from the drive PLO data separator is strobed earlier than usual.

7	6	5	4	3	2	1	0
0	0	0	0	SOFM	SOFP	STBL	STBE

Strobe Early
Strobe Late
Servo Offset Plus
Servo Offset Minus

CUL (Connecting Unit List)
This register stores bit-map information indicating which drive is connected to the HDC.

In ST506 interface, bits 0–3 correspond to drives 0–3 respectively. To connect a drive, write "1" into the corresponding bit (up to 4 drives can be connected).

7	6	5	4	3	2	1	0
0	0	0	0				

Drive Unit 0
Drive Unit 1
Drive Unit 2
Drive Unit 3

In SMD interface, bits 0–7 correspond to drives 0–7 respectively. To connect a drive, write "1" into the corresponding bit (up to 8 drives can be connected).

7	6	5	4	3	2	1	0

Drive Unit 0
Drive Unit 1
Drive Unit 2
Drive Unit 3
Drive Unit 4
Drive Unit 5
Drive Unit 6
Drive Unit 7

TO/NCH, NCL (Read/Write Time Over, Number of Cylinders High/Low)
NCH and NCL registers specify the number of cylinders in disk drive, and time-over during disk access command execution.

(1) TO (Read/Write Time Over)
The high-order 6 bits in TO/NCH are used to assign the ID search time: time-over (t_{over}). According to the value of TO, the HDC sets time-over period as follows.
$\$01 \leqq TO \leqq \$3F$ (TO = $00 is prohibited.)
$TO \times 8 \times 10^4$ CLK $\leqq t_{over} \leqq (TO + 1) \times 8 \times 10^4$ CLK

(2) NC (Number of Cylinders)
The low-order 2 bits of TO/NCH and 8 bits of NCL specify the number of cylinders (NC). NC is 1023 at a maximum. Its value is number of cylinders minus 1.

The HDC uses NC to issue NC + 10 step pulses during RCLB command execution (ST506 interface), or to check whether or not the command parameter NCA (Next Cylinder Address) exceeds NC during SEK command execution.

15	14	13	12	11	10	9	8
Read/Write Time Over						NCH	

Number of Cylinders High

7	6	5	4	3	2	1	0
Number of Cylinders Low							

NH (Number of Heads)
This register indicates the number of heads. Its value is to be number of drive heads minus 1.

In disk access command execution, the HDC checks whether or not PHA (Physical Head Address) specified by command parameters exceeds NH. When PHA exceeds NH, the HDC sets IPH (Invalid Physical Head Address) to result parameter SSB (Sense Status Byte) and abnormally terminates the execution.

To select a head during multiple track operation in disk access command execution, the HDC checks whether PHA exceeds NH or not. When PHA exceeds NH, the HDC sets IPH to SSB and abnormally terminates the execution.

7	6	5	4	3	2	1	0
Number of Heads							

In ST506 interface, up to 8 heads can be selected ($00 ≦ NH ≦ $07). Bits 3–7 must be fixed to "0".

7	6	5	4	3	2	1	0
0	0	0	0	0	2^2	2^1	2^0

In SMD interface, up to 32 heads can be selected ($00 ≦ NH ≦ $1F). Bits 5–7 must be fixed to "0".

7	6	5	4	3	2	1	0
0	0	0	2^4	2^3	2^2	2^1	2^0

NS (Number of Sectors)
This register indicates the number of sectors. Its value is number of sectors/track minus 1 ($00 ≦ NS ≦ $FE).

In data read/write command execution, the HDC checks whether LSA (Logical Sector Address) exceeds NS or not.

In multiple sector operation in data read/write command execution, the HDC checks whether LSA exceeds NS or not each time LSA is incremented after one sector operation. If LSA exceeds NS, the HDC sets "0" to LSA, increments LHA and PHA, and compares NH and PHA. If NH exceeds PHA, the HDC executes multiple track operation. If PHA exceeds NH, the HDC sets IPH (Invalid Physical Head Address) to result parameter SSB and abnormally terminates the execution.

7	6	5	4	3	2	1	0
Number of Sectors							

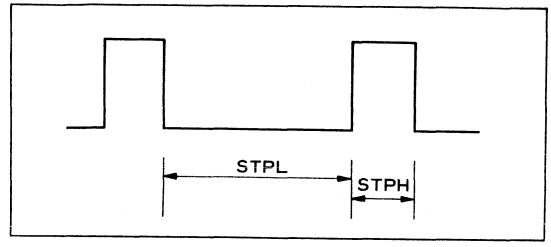
SH/RL (Step Pulse High/Record Length)

7	6	5	4	3	2	1	0
Step Pulse High (SH)					Record Length (RL)		

(1) SH (Step Pulse High)
The high-order 5-bit SH/RL indicates step pulse high width (STPH: Step Pulse High) in ST506 interface.

STPH is fixed to 2CLK in highest speed seek mode (PSK = 1 and SL = $FF). Otherwise, SH sets STPH regardless of the PSK bit. The relation between STPH and SH is shown in the following table. SH is ignored when maximum speed seek mode is selected in ST506 interface.

In SMD interface, the high-order 5 bits are to be fixed to "0".



SH	STPH (step pulse high)
SH = * * : don't care	STPH = 2 CLK (Note 1)
SH = $00	STPH = 3 CLK
$01 ≦ SH ≦ $1F (1) (31)	STPH = SH x 3 + 1 CLK (4 CLK ≦ STPH ≦ 94 CLK)

(Note 1) Highest-speed seek mode (PSK = 1, SL = $FF)

(2) RL (Record Length)
The low-order 3-bit SH/RL indicates record length per sector.

RL			Record Length
Bit 2	Bit 1	Bit 0	
0	0	0	Inhibited
0	0	1	256 bytes
0	1	0	512 bytes
0	1	1	1024 bytes
1	0	0	2048 bytes
1	0	1	4096 bytes
1	1	0	Inhibited
1	1	1	Inhibited

GPL1, 2 (Gap Length 1, 2)
These registers specify the length of gap and SYNC area in the sector during WD and WFM command execution.

GPL1 specifies the length of gap areas (GAP1 in soft sector, HEAD SCAT in hard sector) that follow an index or a sector pulse by byte. It is used for WFM command execution. These areas are formatted 6 bytes longer than the value set to GPL1 during the command execution. [$00 ≦ GPL1 ≦ $FF]

7	6	5	4	3	2	1	0
Gap Length 1							

GPL2 specifies the length of SYNC area located at the beginning of ID and DATA areas by byte. It is used for WD and WFM commands. This area is formatted 3 bytes longer than the value set to GPL2 during the command execution. [$08 ≦ GPL2 ≦ $FF]

7	6	5	4	3	2	1	0
Gap Length 2							

GPL3/RGTLT (Gap Length 3, Read Gate Latency)
GPL3/RGTLT specifies the length of GAP3 for the soft sector and read gate latency delay for the hard sector by byte.

GPL3 specifies the length of GAP3 located at the end of a sector in soft sector format by byte. It is used for WFM command. This area is formatted 3 bytes longer than the value set to GPL3. [$09 ≦ GPL3 ≦ $FF]

7	6	5	4	3	2	1	0
Gap Length 3							

RGTLT specifies the time period between the detection of an index or a sector pulse and the assertion of RGATE in unit of byte. It is used for disk read commands. RGATE is asserted 5 bytes later than the value set to RGTLT. RGTLT must be set to assert RGATE at the beginning of or before PLO SYNC area in the ID field. During RIS command execution, 64 bytes are added to the amount of latency automatically. [$00 ≦ RGTLT ≦ $FF]

7	6	5	4	3	2	1	0
RGATE Latency							

LCCH, LCCL (Low Current Cylinder High/Low)
For the inner cylinders of the drive, it is necessary to reduce write current during WFM and WD command execution. These registers specify address of the outermost cylinder from where write current is reduced. This is valid only in ST506 interface. When a disk write command is executed to any cylinder whose address is equal to or greater than LCC, the HDC asserts the LCT pin to high. [$0000 ≦ LCC ≦ NC (Number of Cylinders)]

15	14	13	12	11	10	9	8
Low Current Cylinder High							

7	6	5	4	3	2	1	0
Low Current Cylinder Low							

PCC (Precompensation Cylinder)
This register specifies the address of the outermost cylinder from where compensation of the bit data timing is required. This is valid only in ST506 interface. When a disk write command is executed to any cylinder whose address is equal to or greater than PCC, either EARLY or LATE signal is generated in accordance with the bit data timing. [$0000 ≦ PCC ≦ NC]

15	14	13	12	11	10	9	8
Pre-compensation Cylinder High							

7	6	5	4	3	2	1	0
Pre-compensation Cylinder Low							

OTHER COMMAND PARAMETERS

The following describes command parameters other than SPC command parameters in alphabetical order.

(1) FLAG
This parameter is used to specify the FLAG byte of ID area of a hard sector that the HDC will access. If the FLAG given by the command parameter does not match the FLAG read from ID area of the disk, the HDC will not access the sector.

(2) LCA (Logical Cylinder Address)
This parameter is used to specify the logical cylinder address of ID area (16 bits: the high-order 8 bits for LCAH and the low-order 8 bits for LCAL) of a sector that the HDC will access. If LCA given by the command parameter does not match LCA read from ID area, the HDC will not access the sector. In soft sector format, it is prohibited to specify $F8 to the high-order 8 bits (LCAH).

(3) LHA (Logical Head Address)
LHA is used to specify the logical head address of ID area of a sector that the HDC will access. If LHA given by the command parameter does not match the LHA read from ID area, the HDC will not access that sector. In multiple track operation, LHA is automatically incremented by one. Since LHA is logical, it may exceed the number specified by the parameter NH (number of heads).

(4) LSA (Logical Sector Address)
LSA is used to specify the logical sector address of ID area that the HDC will access. If LSA given by the command parameter does not match LSA read from ID area, the HDC will not access the sector. In multiple sector operation, LSA is automatically incremented by one. Since LSA is compared with the control register NS (number of sectors) for switching of heads, the LSA must not exceed the NS.

(5) NCA (Next Cylinder Address)
This parameter is used to specify the physical address (10 bits) of a cylinder to which the head will move when a SEK command is issued. The outermost cylinder address is $0000. The high-order 8 bits of NCA are used for NCAH and the low-order 8 bits for NCAL. The high-order 6 bits of NCAH must be fixed to "0". If NCA exceeds NC, com-

mand execution abnormally terminates.

(6) OFFSET
OFFSET specifies how many sectors to be skipped reading after an index pulse. Then the HDC reads ID fields, and stores them into the data buffer.

(7) PHA (Physical Head Address)
The MPU specifies PHA when issuing a disk access command. Unlike LHA, PHA is physical, and the bit status of PHA is directly output as HSEL signals. The high-order 5 bits of PHA must be fixed to "0" for ST506 interface, and the high-order 3 bits for SMD interface. In multiple track operation, PHA is automatically incremented by one within the HDC if another head switching is required. If PHA exceeds the value given by NH, the command execution will abnormally terminate.

(8) POFF (Pointer Offset)
The MPU specifies a transfer start address (16 bits) when issuing a command for accessing the data buffer. The MSB of the high-order 8 bits (POFFH) selects one of two data buffers, and the remaining 7 bits must be fixed to "0". The low-order 8 bits (POFFL) specifies a transfer start address of the selected data buffer (256 bytes). For the 16-bit data bus, POFF is limited to an even address only.

(9) PSA (Physical Sector Address)
This parameter is used to specify a physical address of a hard sector at which the execution of RID, RIS, FID, WFM, or WFS starts. If PSA is $00, the sector following an index pulse is specified.

(10) SCNT (Sector Count)
This parameter is used to specify the number of sectors (16 bits) that the HDC will continuously access in a disk access command execution. Upper 8 bits are SCNTH, and lower 8 bits are SCNTL. Up to 1024 sectors are specified in ST506 interface (128 sectors x 8 heads), and up to 8160 sectors are specifiable in SMD interface (255 sectors x 32 heads). For commands relating to the ID (RID and WFM), they perform no multiple track operation. In addition, maximum number of sectors that can be formatted at a time by WFM command is 128 for soft sector, and 102 for hard sector.

(11) US (Unit Select)
The MPU specifies the address (8 bits) of a target drive when issuing a head positioning, disk access or drive check command. The contents of US are directly output from USEL signals. The high-order 6 bits of US must be fixed to "0" for ST506 interface, and the high-order 5 bits for SMD interface. US is the high-order 8 bits of a 16-bit word, and the low-order 8 bits of the word are PHA or $00. It is not necessary to issue $00 to the low-order 8 bits when CKV or RCLB command is issued.

RESULT PARAMETERS

Result parameters are listed in Table 9. In this table, result parameters used by the HDC to control ST506 type hard disk drive are found in the upper row of each command, and those for SMD type in the lower row.

Table 9 Result Parameters (Byte-organized)

Command	Parameter (Upper row: ST506 / Lower row: SMD)
Recalibrate	$00 SSB US VUL $00 SSB US VUL
Specify	($00 SSB) ($00 SSB)
Seek	$00 SSB US VUL $00 SSB US VUL
Read Data	$00 SSB US PHA LCAH LCAL LHA LSA SCNTH SCNTL $00 SSB US PHA $00 FLAG LCAH LCAL LHA LSA SCNTH SCNTL
Read Erroneous Data	$00 SSB US PHA SCNTH SCNTL $00 SSB US PHA $00 PSA SCNTH SCNTL
Read ID Read ID Skew (Note 1)	$00 SSB US PHA $00 SCNTL $00 SSB US PHA $00 PSA $00 SCNTL
Find ID	$00 SSB US PHA $00 SCNTL $00 SSB US PHA $00 PSA $00 SCNTL
Check Data	$00 SSB US PHA LCAH LCAL LHA LSA SCNTH SCNTL $00 SSB US PHA $00 FLAG LCAH LCAL LHA LSA SCNTH SCNTL
Compare Data	$00 SSB US PHA LCAH LCAL LHA LSA SCNTH SCNTL $00 SSB US PHA $00 FLAG LCAH LCAL LHA LSA SCNTH SCNTL
Write Data	$00 SSB US PHA LCAH LCAL LHA LSA SCNTH SCNTL $00 SSB US PHA $00 FLAG LCAH LCAL LHA LSA SCNTH SCNTL
Write Format Write Format Skew (Note 1)	$00 SSB US PHA SCNTH SCNTL $00 SSB US PHA $00 PSA SCNTH SCNTL
Memory to Buffer	($00 SSB) ($00 SSB)
Buffer to Memory	($00 SSB) ($00 SSB)
Open Buffer Read	($00 SSB) ($00 SSB)
Open Buffer Write	($00 SSB) ($00 SSB)
Polling	$00 SSB US VUL $00 SSB US VUL
Check Drive	$00 SSB US $00 DST0 $00 $00 SSB US $00 DST0 DST1 DST2 DST3
Abort	$00 SSB $00 SSB
Check ECC	$00 SSB EA0 EA1 EP0 EP1 EP2 $00 $00 SSB EA0 EA1 EP0 EP1 EP2 $00
Test	($00 SSB) ($00 SSB)
Polling Disable	($00 SSB US VUL) ($00 SSB US VUL)
Recall	($00 SSB US VUL) ($00 SSB US VUL)

(Note 1) Read ID Skew and Write Format Skew are valid only for SMD.
(Note 2) Parenthesized parameters are reported when a command is issued under the illegal condition.

DST (Drive Status)

Table 10 Drive List (DST)

Bit		7	6	5	4	3	2	1	0
ST506	DST0	READY	SCP	TRK0	0	WFLT	0	0	0
SMD	DST0	*	*	*	WPRT	FLT	SKERR	OCYL	URDY
	DST1	*	*	SELER	WERR3	WERR2	WERR1	SERR2	SERR1
	DST2	*	*	SAD32	SAD16	SAD8	SAD 4	SAD 2	SAD 1
	DST3	*	*	DTP32	DTP16	DTP8	DTP4	DTP2	DTP1

*: Depends on the state of BUS_1/BUS_6 – BUS_4/BUS_9.

Each bit of DST indicates drive interface input signal level. DST0 (1 byte) is reported in ST506 interface, and DST0–3 (4 bytes) are reported in SMD interface.

In ST506 interface, the WFLT bit in DST0 indicates the signal level of WFLT pin. Even if WFLT signal is momentarily asserted, the internal latch memorizes this and reflects its state on WFLT bit. When WFLT signal is negated, the latch read operation by the HDC clears this latch.

In SMD interface, the HDC reads 8 bits by 4 words of the drive status signal which is selected by signals TAG2 and TAG5.

TAG2	TAG5	Status
0	0	DST0
0	1	DST1
1	0	DST2
1	1	DST3

Following (1) through (5) are read during CKV command execution and checked during disk access command execution. DST1 through DST3 are referred to during CKV command execution.

Status that each bit in DST1–3 indicates may vary according to the drive connected to the HDC. Following descriptions indicate DST bit function when the HDC is connected to a Hitachi 8″ disk drive, DK-812S.

(1) URDY (Unit Ready)
This bit indicates that a selected drive is in ready state.

(2) OCYL (On Cylinder)
This bit indicates that a head is positioned correctly on a track.

(3) SKERR (Seek Error)
This bit indicates that errors have been detected in a drive during seek operation.

(4) FLT (Fault)
This bit indicates that errors relevant to disk access have been detected in the drive. Result parameter DST1 indicates what type of error has occurred. See (6) through (11) for DST bits indicating error status.

(5) WPRT (Write Protected)
This bit indicates that a selected drive is write protected.

(6) SERR1 (Status Error 1)

(7) SERR2 (Status Error 2)

(8) WERR1 (Write Error 1)

(9) WERR2 (Write Error 2)

(10) WERR3 (Write Error 3)

(11) SELER (Select Error)

(12) SAD1–32 (Sector Address)

(13) DTP1–32 (Drive Type)

EA (Error Address)
Reports the start address (16 bits) of a byte from where the burst error that is detected during CKE command execution exists. The high-order 8 bits are EA0, and the low-order 8 bits are EA1. As the HDC corrects up to 11 bits of burst error, the MPU corrects contiguous 3 bytes starting from a byte specified by EA. EA = $0000 indicates a start address of sector data.

EP (Error Pattern)
Reports EP0, EP1, and EP2 as the pattern required for error correction as a result of CKE command execution. The MPU exclusive-OR 3-byte data containing errors with EP0, EP1, and EP2.

FLAG (Flag)
Reports the same value as FLAG specified by command parameters.

LCA (Logical Cylinder Address)
Reports the same value as LCA specified by command parameters.

LHA (Logical Head Address)
Bit organization of LHA is the same as that of the command parameter LHA. In multiple track operation, LHA is incremented by one each time access to one track ends.

LSA (Logical Sector Address)
Bit organization of LSA is the same as that of the command parameter LSA. In multiple sector operation, LSA is incremented by one each time access to one sector ends.

PHA (Physical Head Address)
Bit organization of PHA is the same as that of the command parameter PHA. In multiple track operation, PHA is incremented by one each time access to one track ends.

PSA (Physical Sector Address)
Bit organization of PSA is the same as that of the command parameter PSA. In multiple sector operation, PSA is incremented by one each time access to one sector ends.

SCNT (Sector Count)
Bit organization of SCNT is the same as that of the command parameter SCNT. In multiple sector operation, SCNT is decremented by one each time access to one sector ends. When SCNT reaches "0", command execution normally terminates.

SSB (Sense Status Byte)
This is the area where 8-bit error code is set ($00 is set for normal termination). There are 25 error codes which notify what kind of error has occurred during the command execution (see tables 11–14).

US (Unit Select)
Reports the same value as US specified by command parameters.

VUL (Valid Unit List)
VUL is a bit-mapped list which gives the address of a drive that is ready to accept a head positioning or a disk access command. Its bit organization is the same as that of CUL. This bit is set when seek operation in the drive is terminated, or when SPC command is executed. This bit is reset when seek operation starts, when drive error occurs, or when the bit in CUL corresponding to the drive is not set to "1".

Table 11 Error Codes Relevant to Host Interface

Mnemonic	Name	Error Code	Contents
ABT	Command Aborted	04	ABT command has been accepted.
IVC	Invalid Command	08	An invalid command has been accepted.
PER	Parameter Error	0C	The command parameter has not been stored in an appropriate area in PB.
NIN	Not Initialized	10	Head positioning, disk access, and drive check commands have been accepted, before SPC command is executed.
RTS	Rejected Test	14	The TST command is received after SPC command has been received.

Table 12 Error Codes Relevant to Drives

Mnemonic	Name	Error Code	Contents
NUS	No USELD	18	USELD signal for a selected drive has not been returned.
WFL	Write Fault	1C	WFLT signal (ST506 interface) or FLT signal (SMD interface) has been detected.
NRY	Not Ready	20	READY signal has been negated.

Table 13 Error Codes Relevant to Head Positioning Commands

Mnemonic	Name	Error Code	Contents
NSC	No SCP	24	SCP signal (ST506 interface) or the SKEND signal (SMD interface) has not been returned in a certain period.
ISE	In Seek	28	SEK, or a disk access command has been issued for a drive in seek operation.
INC	Invalid NCA	2C	NCA (Next Cylinder Address) greater than NC (number of cylinders) has been specified.
ISR	Invalid Step Rate	30	The highest-speed seek has been specified in the normal seek mode.
SKE	Seek Error	34	SEK or a disk access command has been issued to a drive which is in seek error state (SMD only).

Table 14 Error Codes Relevant to Disk Read/Write Commands

Mnemonic	Name	Error Code	Contents
OVR	Over Run	38	A transfer between the main memory and data buffers has not caught up with a transfer between a drive and data buffers.
IPH	Invalid PHA	3C	PHA (Physical Head Address) greater than NH (Number of Heads) has been specified.
DEE	DATA Field ECC Error	40	A data error has been detected by ECC (Error Correction Code).
DCE	DATA Field CRC Error	44	A CRC (Cyclic Redundancy Check Code) error has occurred in DATA area.
ECR	Error Corrected	48	An ECC error detected in DATA area has been automatically corrected.
DFE	DATA Field Fatal Error	4C	A fatal ECC error has occurred in DATA area.
NDA	No DATA AM	60	The address mark in DATA area has not been detected.
NHT	Not Hit	50	In CMPD command execution, data from the host and disk data have not coincided with each other.
ICE	ID Field CRC Error	54	A CRC error in ID area has been detected in RID command execution in SMD interface.
TOV	Time Over	58	ID has not been found in the period specified by TO (Time Over).
NIA	No ID AM	5C	The ID area that begins with improper address mark has been detected.
NWR	Not Writable	64	WD command has been issued to a drive where the write protect signal is asserted (SMD interface)

CONTROL SEQUENCE FOR ST506 TYPE DRIVE

DISK READ

The ID search timing for RD or WD command execution is shown in figure 27. (a) is the timing where AM is found. SYNC signal is negated 4-bit period (on the basis of disk data) after CRC pattern of ID area is completed. After reading CRC, the HDC switches the clock in the satellite processor from RCLK to WCLK and then negates SYNC signal.

RCLK and WCLK clock signals are independent each other. To remove the hazard during switching, the switching signal and the clock signal must be synchronized. Therefore, SYNC is normally negated 4-bit period after reading CRC to switch the clock in the satellite processor from RCLK to WCLK.

RGATE signal is usually negated 4-bit period after SYNC signal is negated, and then asserted again one to two-byte period later to read SYNC area preceding DATA area. RGATE is asserted for 1-byte period when the AMEX bit is set to "1" during SPC command execution, and for 2-byte period when set to "0".

When AM is not found as shown in (b), the HDC searches for AM again by negating both SYNC and RGATE signals. RGATE signal is usually negated 4-bit period after SYNC signal is negated, and is negated for 2-byte period.

As the MFM is specified during SPC command execution, the frequency of clock signals synchronizing with disk data (such as RCLK, WCLK) is doubled compared to the data transfer rate. Therefore, "4-bit period" or "1-byte period" description is formally expressed as "8 WCLK cycles" or "16 WCLK cycles" respectively.

After reading DATA area and ECC or CRC area by RD command execution, RGATE and SYNC signals are negated in the same timing as they are negated after reading ID CRC area (see figure 27 (a)).

DISK WRITE

The timing of WGATE signal for WD command execution is shown in figure 28. The HDC negates RGATE signal after reading ID area. Then WGATE signal is asserted 1 or 2-byte period after RGATE is negated (2-byte when AMEX specified by SPC command is at "0", 1-byte when AMEX is at "1"). WGATE signal is negated immediately after PAD of DATA field ends.

The relation between IDX signal and WGATE signal during WFM command execution is shown in figure 29. WGATE signal rises almost simultaneously with IDX signal (refer to electrical characteristics), and falls 3-byte or 4-byte period after IDX signal rises. The condition of EARLY and LATE signal generation is shown in figure 8.

HEAD POSITIONING

The relation between DIR, STEP, and SEEK signals is shown in figure 30. The unit is the number of CLK cycles (t_{cyc}).

When 8 MHz clock signal is supplied to CLK, STEP signal is output at least 33.8 μs after DIR signal becomes valid.

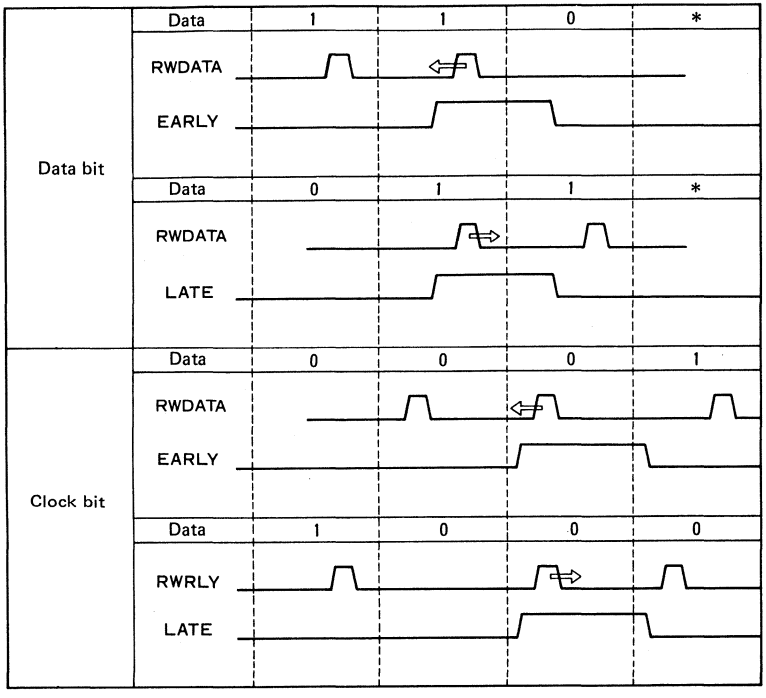


Figure 8 Write Precompensation

CONTROL SEQUENCE FOR SMD TYPE DRIVE

DRIVE SELECTION, STATUS CHECK, FAULT CLEAR

The HDC performs drive selection, status check, and Fault Clear, for all disk control command execution. Figure 35 shows the drive selection timing. The drive receives $BUS_0/BUS_5 - BUS_4/BUS_9$ from the HDC and latches them using UTAG signal. If the drive judges that its own drive number is specified, it returns USELD signal within 9 CLK after the detection of UTAG signal. Figure 36 shows the drive status check sequence. The HDC changes the direction of $BUS_0/BUS_5 - BUS_4/BUS_9$ before reading the status. Then the HDC sets BUSL/$\overline{H}$ signal to high, therefore, the external circuit of the HDC can supply drive status signal to the HDC during this period. At this time, BUSL/$\overline{H}$ is in high, and lower 5 bits are input to the HDC.

The HDC resets FLT after reading the drive status which includes Fault, and the timing is shown in figure 37. To set the high-order 5 bits of BUS outputs to low, the HDC sets $BUS_0/BUS_5 - BUS_4/BUS_9$ to low, and asserts TAG3 signal with maintaining BUSL/$\overline{H}$ in low. In external circuits of the HDC, $BUS_0/BUS_5 - BUS_4/BUS_9$ is latched by using TAG3 signal as a strobe signal. If BUSL/$\overline{H}$ is in low, external circuits must be used not to supply TAG3 signal of the HDC to the drive. Then, the HDC outputs the low-order 5-bit information to $BUS_0/BUS_5 - BUS_4/BUS_9$, sets BUSL/$\overline{H}$ signal to high, and asserts TAG3 signal. The external circuits provide the drive with above mentioned 5-bit latch outputs, the low-order 5-bit outputs on $BUS_0/BUS_5 - BUS_4/BUS_9$, and TAG3. In Fault Clear, BUSL/$\overline{H}$ signal is in high, and only BUS_4/BUS_9 is in high.

Head selection is performed by disk access commands. Figure 40 shows the head selection timing. Upper 5 bits are in low, and the head address is supplied from the low-order 5 bits of BUS. At this time, TAG2 signal is used as a strobe signal.

Further, in the execution of disk access command group, the HDC asserts TAG3 and generates RGATE or WGATE signal. Figures 41, 42 and 43 show the total sequence. When TAG3 is asserted, the high-order 5 bits of BUS output Strobe Early/Late and RTZ (it is not used for disk access). Then, the HDC outputs the low-order 5 bits. Servo offset is specified in figure 42. When TAG3 is asserted along with BUS_2/BUS_7 or BUS_3/BUS_8, servo offset is performed in the drive. Therefore, the head is moved and the HDC waits until SKEND signal is returned from the drive. Then, the HDC asserts BUS_1/BUS_6 and supplies RGATE signal to the drive to perform the read operation.

Figure 43 shows assert/negate timing of RGATE and WGATE signals which correspond to the disk format. During disk access command or ID read command execution, RGATE signal is

asserted after a time period corresponding to byte count of disk data has elapsed since IDX or SEC signal is detected. After reading CRC, the HDC itself switches the clock to WCLK signal, then negates RGATE signal with a typical delay time equivalent to 8 bits of disk data.

To read the data field, the HDC asserts RGATE again with a dwell time of 1–2 bytes period. RGATE is negated for 1-byte period when AMEX bit is set to "1" during SPC command execution, and 2-byte period when AMEX bit is set to "0". The HDC negates RGATE signal 8-bit period after reading CRC or ECC in data field.

To write the DATA field, the HDC reads ID area and asserts WGATE signal 1–2 bytes period after RGATE signal is negated. WGATE signal is to be negated when DATA PAD ends. During WFM command execution, WGATE signal rises in the same timing as IDX signal, and is negated 3–4 bytes period after IDX or SEC signal is detected.

Figure 38 shows the execution timing of RCLB command. RTZ instruction is supplied to the drive through BUS_1/BUS_6 and TAG3. Figure 39 shows the execution timing of SEK command. 10-bit cylinder address is output through $BUS_0/BUS_5-BUS_4/BUS_9$, 5 bits at a time. Strobe signal is TAG1 at this time.

Figures 44, 45, 46 and 47 shows the execution timing of CKV command. The HDC reads 8-bit drive status through $BUS_0/BUS_5-BUS_4/BUS_9$, 3 bits first and then 5 bits. Then the HDC reads status four times by switching TAG2 and TAG5 signals. Switching order is: [TAG2 = 0, TAG5 = 1], [TAG2 = 0, TAG5 = 1], [TAG2 = 1, TAG5 = 0], [TAG2 = 1, TAG5 = 1].

COMMON CONTROL

The control which is common to both ST506 and SMD interfaces is described in this section.

AUTOMATIC CORRECTION

In SPC command execution, the HDC operates in automatic correction mode if the host sets both ECD and ACOR bits in OM0 to "1". If a sector with correctable errors is detected, the erroneous data is corrected in data buffers. Automatic correction is normally performed only when the sector length is 256 bytes. Specification of the automatic correction mode is prohibited when the sector length is longer than 512 bytes because it causes the erroneous operation of the HDC.

If DTM (Data Transfer Mode) bit in OM0 is set to "1" as well as ECD and ACOR bits during SPC command execution, the HDC transfers the corrected data to the main memory in DMA mode after the automatic correction. Then, the command execution abnormally terminates. When DTM = 0, the command execution abnormally terminates after the automatic correction and data transfer is not performed. If the HDC finds the data uncorrectable, the command execution abnormally terminates without data transfer regardless of the state of DTM bit.

CORRECTION BY HOST

The host corrects erroneous data detected during RD command execution when it sets ECD and ACOR bits in OM0 to "1" and "0" respectively. When ECD bit is set to "1", the host corrects erroneous data detected during RED command execution if necessary, regardless of the status of ACOR bit. The command execution abnormally terminates when erroneous data is detected during any command execution. Then $40 is set to SSB when an error is correctable, $4C when not correctable.

When DTM = 1, the HDC sends 1-sector data including an error during RD or RED command execution and then abnormally terminates the command execution. After confirming that $40 is set to SSB, the host issues CKE command to the HDC.

When DTM = 0, the HDC abnormally terminates the command execution after 1-sector data including an error is stored in the data buffer during RD or RED command execution. After confirming that $40 is set to SSB, the host issues BTM command to the HDC after DMAC's initialization, or, issues OPBR command and reads data buffers using move and load instructions and then stores buffer data to the main memory. Then, the host issues CKE command to the HDC.

Receiving CKE command, the HDC calculates an error address (EA) and an error pattern (EP), and then sends result parameters to the host. The erroneous data exists in three contiguous bytes of the corresponding sector in main memory. EA indicates start address of this 3-byte area ($0000 indicates a start address of sector data). The host exclusive-OR the 3-byte area and EP to correct erroneous data.

The host judges CKE command execution end from status change of BSY bit from "1" to "0". When SSB is $4C, CKE command issue is prohibited as it causes the erroneous operation of the HDC. Even if errors are reported to be correctable (SSB = $40) during RD or RED command execution and the host issues CKE command, these errors may turn out to be not correctable after CKE command execution.

DMA DATA TRANSFER

In DMA data transfer, the HDC selects one of two modes: Burst mode and Cycle Steal mode. DMA transfer mode is specified by BRST (burst) bit which is a command parameter of SPC command.

In Burst mode (BRST = 1), the HDC maintains $\overline{DREQ}$ signal at low level until data transfer is terminated. $\overline{DREQ}$ signal is negated when $\overline{DONE}$ signal is input synchronously with $\overline{DACK}$ signal, or when DMA transfer of data in 256-byte internal data buffer is finished.

In Cycle Steal mode (BRST = 0), the HDC asserts $\overline{DREQ}$ signal until $\overline{DACK}$ signal is asserted. The HDC negates $\overline{DREQ}$ signal when $\overline{DACK}$ signal assertion is detected. $\overline{DREQ}$ signal will be reasserted if $\overline{DACK}$ is negated and the HDC has the transfer request. The DMA transfer conditions of Cycle Steal mode are the same as those of Burst mode.

DATA TRANSFER IN HOST INTERFACE

DMA Data Transfer in Disk Access

DMA data transfer mode, either Burst mode or Cycle Steal mode, is selected by BRST bit specified by SPC command. Figure 9 shows DMA data transfer sequence during RD and RED command execution. For 256-byte sector, (a) shows the case that the host computer reads the disk data from data buffers (DBUF0 and DBUF1) by DMA at a high speed. The capacity of two data buffers is 256 bytes for each.

If data buffers are not provided, most of high-speed host bus is occupied by transferring drive data during low-speed reading of drive data.

Internal data buffers can separate high-speed host bus timing and low-speed drive data timing, which effectively shorten drive data transfer time in host bus. Therefore, host system throughput is notably improved. The data transfer from data buffers to the main memory is performed only when the HDC is reading ID area or DATA area of the disk, but is exceptional when data in the last sector is transferred to the main memory during the multiple sector operation.

Figure 9 (b) shows the low-speed DMA transfer of the host in 256-byte sector organization. According to the figure, DBUF0 receives data of the sector 1, and DBUF1 receives that of the sector i+1. While DBUF1 is receiving the data of sector i+1, the data transfer from DBUF0 to the main memory cannot be finished because the host cannot operate promptly. Therefore, being unable to receive the data of i+2, DBUF0 waits until the disk makes one rotation (all data of DBUF0 is sent to the main manner, no data overrun occurs even if operation speed of the host is slow. Therefore, an interleave format is not required.

Figure 9 (c) indicates the sequence in 512-byte sector organization. Capacity of buffers is 256 bytes for each, and each buffer stores disk data and transfers the data to the main memory alternately. Therefore, buffers effectively operates in toggle fashion even if sector length exceeds 256 bytes, and host system throughput is improved. If operation speed of the host is too slow, data overrun may occur because data buffer cannot be emptied to receive disk data. The data transfer from data buffer to main memory must be terminated while the HDC is accessing disk data in ID or DATA area.

If the HDC receives $\overline{\text{DONE}}$ signal from the DMAC during RD or RED command execution, the HDC immediately terminates DMA data transfer. Disk reading operation continues until the HDC finishes reading the sector which has been read when $\overline{\text{DONE}}$ signal is applied.

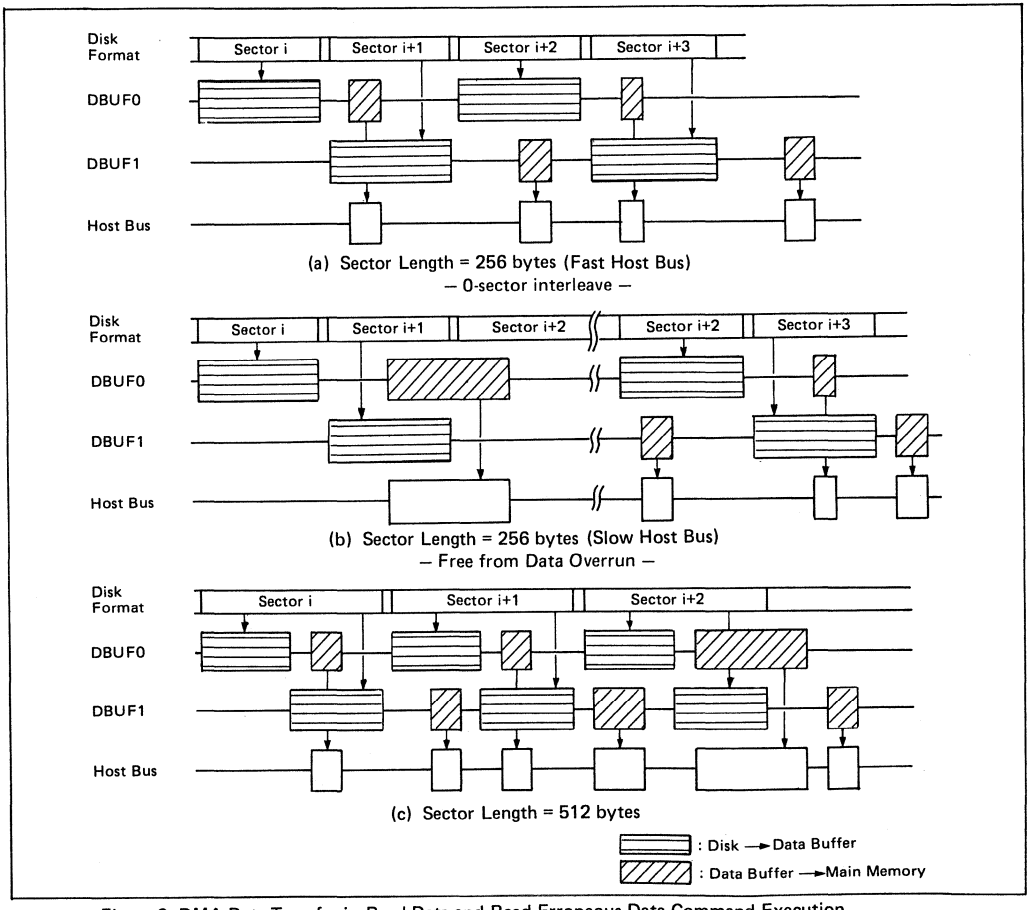


Figure 9 DMA Data Transfer in Read Data and Read Erroneous Data Command Execution

DMA data transfer during WD and CMPD command execution is shown in figure 10. This figure differs from figure 9 (a) in data transfer direction, DMA transfer order, and access method of first and last sectors. In figure 10, the host is fast and sector length is 256 bytes. When the host is slow and sector length is 256 bytes, or when sector length is 512 bytes or more, data transfer sequence is the same as figure 9.

When the HDC receives $\overline{\text{DONE}}$ signal from the DMAC during WD command execution, the HDC immediately terminates DMA data transfer. Then all data stored in DBUF0 and DBUF1 are written to DATA area of the sector. If there is any room in DATA area after writing buffer data, the old data in buffers is written to DATA area. After DATA area is filled with buffer data, command execution terminates.

When the HDC receives $\overline{\text{DONE}}$ signal from the DMAC during CMPD command execution, the HDC immediately terminates DMA data transfer. However, the data of the sector which has been accessed when $\overline{\text{DONE}}$ is received, is used for comparison with disk data, further, the reading operation continues until reading of the sector is finished. Therefore, $50 (Not Hit) is set to SSB in most cases when reading of the sector is finished, and then the command execution abnormally terminates.

DMA data transfer during RID or FID command execution is performed as follows (see figure 11). When DBUF0 is filled with ID information, DBUF1 receives the succeeding ID information. DMA data transfer is not performed unless all the ID information is stored in data buffer. After DBUF0 with/without DBUF1 finishes accepting ID information from specified number of sectors, data are sent to the main memory by DMA. DMA data transfer continues until 512-byte transfer is finished.

DMA data transfer can be stopped when the DMAC sends $\overline{\text{DONE}}$ to the HDC. Therefore, the host can store only the necessary ID information to the main memory by setting the number of ID areas to be read, to the DMAC.

DMA data transfer during WFM command execution is shown in figure 12. The HDC starts formatting operation when both DBUF0 and DBUF1 are filled with ID information from the host. The HDC writes data buffer data to the ID area of a specified sector, and writes fixed pattern to other area such as DATA area. When DBUF0 becomes empty during formatting operation, DBUF1 becomes the source in turn. When the DMAC issues $\overline{\text{DONE}}$ signal to the HDC, the HDC stops data transfer to data buffers and starts formatting. Therefore, the host can write only the necessary ID information to data buffer by setting data of how many ID areas to be transferred, to the DMAC.

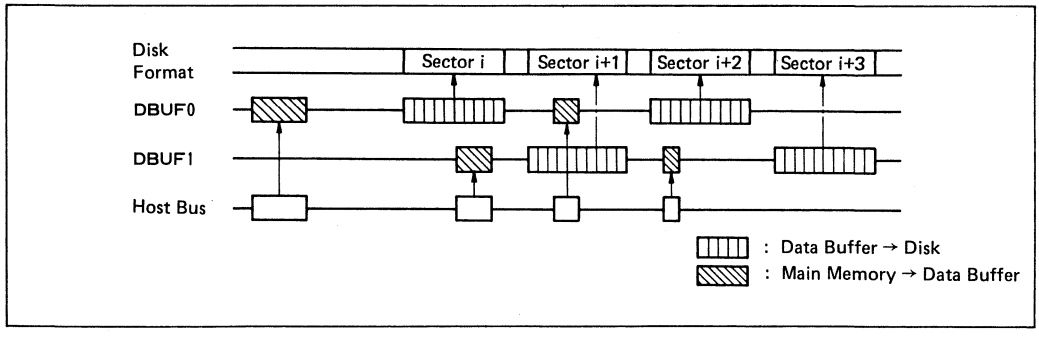


Figure 10 DMA Data Transfer in Write Data, Compare Data Command Execution (Sector Length = 256 bytes)

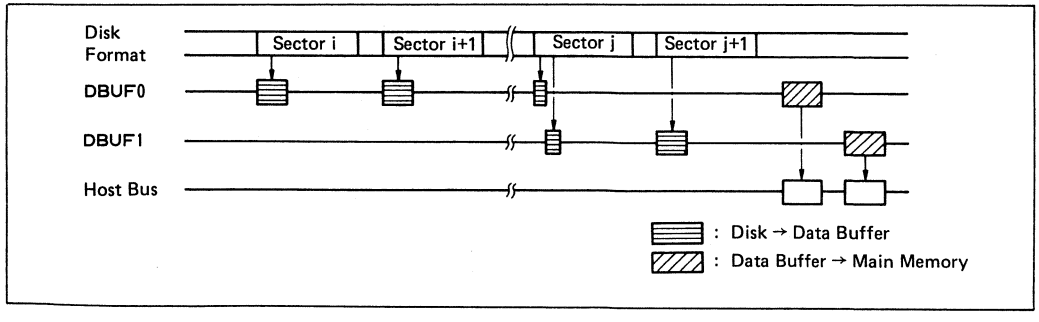


Figure 11 DMA Transfer in Read ID, Find ID Command Execution

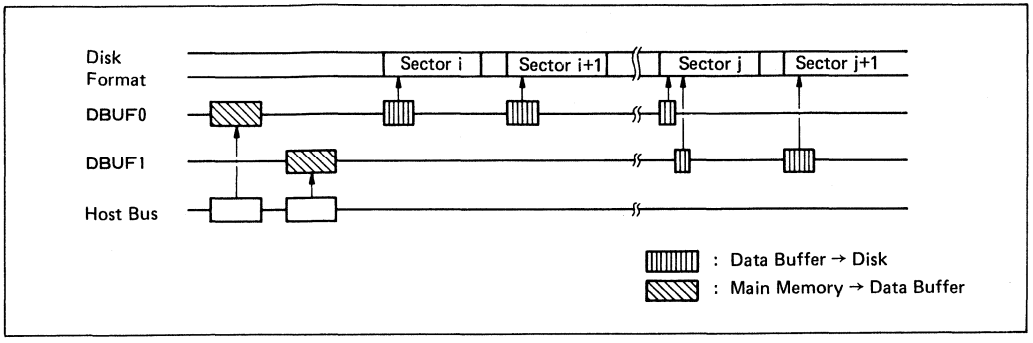


Figure 12 DMA Data Transfer in Write Format Command Execution

DMA Data Transfer by Data Transfer Commands

Accepting data transfer commands, the HDC can perform DMA data transfer between the main memory and data buffers without disk access. Either Burst mode or Cycle Steal mode is selected by the BRST bit of SPC command parameter. There are two commands available, BTM and MTB.

Selection of either DBUF0 or DBUF1, and access start address of each buffer (address 0 to 250 are specifiable), are specified by POFF (Pointer Offset). POFF is set to the data buffer pointer by command execution.

Data buffer pointer is incremented each time the DMAC accesses data buffer (+2 for 16-bit bus mode, +1 for 8-bit bus mode). Data buffer data is not guaranteed if data buffer pointer exceeds address 255. Receiving $\overline{\text{DONE}}$ signal from the DMAC, the command execution immediately terminates normally, and the CED bit in STR is set to "1". Receiving above data transfer commands, the HDC requests transfer by issuing $\overline{\text{DREQ}}$ signal to the DMAC within 150 CLK cycles.

Data Buffer Access by PIO

The host MPU can access data buffers by move or load/store instruction (Programmed I/O). In this case, the host issues OPBR or OPBW command prior to PIO access. Receiving these commands, the HDC sets BSY bit in STR to "1". Then the HDC sets data buffer pointer, sets BSY bit to "0", and terminates the command execution. It takes up to 100 CLK cycles from the command reception to BSY bit clear. Before issuing above commands, POFF must be specified by command parameters. POFF specifies either DBUF0 or DBUF1 to be accessed, and access start address (specifiable range is address 0 to 255).

The host issues move instruction to the HDC after confirming that BSY = 0. Data buffer pointer is incremented each time the HDC receives move instruction (+2 for 16-bit bus mode, +1 for 8-bit bus mode). If the pointer exceeds address 255 during data transfer in PIO mode, the buffer data is not guaranteed. The host must issue RCAL command to close the buffer.

Notes on Data Buffer

Item \ Sector Length (Byte)	256	512	1024	2048	4096
High Speed DMA (0-sector interleave)	○	○	○	○	○
PIO	○	○			
Low Speed DMA (free from overrun)	○				
Automatic Correction	○				

○ : Provided

Others

Recall command – After the host issues RCAL command, the HDC finishes the operation within up to 40 CLK cycles and clears all bits in STR.

Test command – After the host issues TST command, the HDC makes drive output pins floated within 60 CLK cycles. To cancel this state, $\overline{\text{RES}}$ signal is to be externally supplied.

Specify command – After the host issues SPC command, the HDC makes the drive interface pins fitted for either ST506 or SMD, and clears BSY bit within 250 cycles.

Abort command – After the host issues ABT command, the HDC negates all drive output pins within up to 180 CLK cycles, terminates all operation within 300 CLK cycles, and clears BSY bit. Receiving this command, the HDC clears all internal flip-flops, but the value of the control register which has been set by SPC command still remains.

Reset – Reset is not a command, but after receiving $\overline{\text{RES}}$ signal, the HDC terminates initialization within up to 150 CLK cycles and clears BSY bit.

ABSOLUTE MAXIMUM RATINGS

Item	Symbol	Value	Unit
Supply Voltage	V_{CC} *1	−0.3 − +7.0	V
Input Voltage	V_{in} *1	−0.3 − V_{CC} + 0.3	V
Output Current per Pin	$\|Io\|$ *2	5	mA
Total Output Current	$\|\Sigma Io\|$ *3	80	mA
Operating Temperature	T_{opr}	0 − +70	°C
Storage Temperature	T_{stg}	−55 − +150	°C

*1 This value is in reference to V_{SS} = 0V.

*2 The allowable output current is the maximum current that may be drawn from, or flow out to, one output terminal or one input/output common terminal.

*3 The total allowable output current is the total sum of currents that may be drawn from, or flow out to, output terminals or input/output common terminals.

(Note) Using an LSI beyond its maximum ratings may result in its permanent destruction. LSI's should usually be used under recommended operating conditions. Exceeding any of these conditions may adversely affect its reliability.

RECOMMENDED OPERATING CONDITIONS

Item	Symbol	Min	Typ	Max	Unit
Supply Voltage	V_{CC} *	4.75	5.0	5.25	V
Input Low Level Voltage	V_{IL} *	0	–	0.8	V
Input High Level Voltage	V_{IH} *	2.2	–	V_{CC}	V
Operating Temperature	T_{opr}	0	25	70	°C

* This value is in reference to V_{SS} = 0V.

ELECTRICAL CHARACTERISTICS

DC CHARACTERISTICS (V_{CC} = 5.0V ± 5%, V_{SS} = 0V, Ta = 0 – 70°C, unless otherwise noted)

Item			Symbol	4 MHz Version HD63463-4		6 MHz Version HD63463-6		8 MHz Version HD63463-8		Unit	Test Condition
				min	max	min	max	min	max		
Input High Level Voltage	All Inputs		V_{IH}	2.2	V_{CC}	2.2	V_{CC}	2.2	V_{CC}	V	
Input Low Level Voltage	All Inputs		V_{IL}	–0.3	0.8	–0.3	0.8	–0.3	0.8	V	
Input Leak Current	R/$\overline{W}$, $\overline{CS}$, RS, $\overline{RES}$ $\overline{DACK}$, CLK $\overline{DONE}$		I_{IN}	–2.5	2.5	–2.5	2.5	–2.5	2.5	μA	V_{in} = 0–V_{CC}
	ST506	SMD									
	IDX/ TRKO	IDX									
	SCP	SEC									
	USELD	USELD									
	WCLK	WCLK									
	RCLK	RCLK									
Three State (Off State) Input Current	RWDATA	RWDATA	I_{TSI}	–10	10	–10	10	–10	10	μA	V_{in} = 0.4–V_{CC}
	SYNC*1	SKEND									
	WGATE*2	BUS_0/BUS_5									
	EARLY/ RGATE*1	BUS_1/BUS_6									
	LATE/ STEP*1	BUS_2/BUS_7									
	WFLT	BUS_3/BUS_8									
	LCT/DIR*1	BUS_4/BUS_9									
	READY	BUSL/$\overline{H}$*1									
	D_0 – D_{15}										
Output High Level Voltage	All Outputs		V_{OH}	V_{CC}–1.0	–	V_{CC}–1.0	–	V_{CC}–1.0	–	V	I_{OH} = –400 μA
Output Low Level Voltage			V_{OL}	–	0.5	–	0.5	–	0.5	V	I_{OL} = 2.2 mA
Output Leak Cu ent (Off State)	$\overline{IRQ}$		I_{LOH}	–	10	–	10	–	10	μA	V_{OH} = V_{CC}

(to be continued)

*1 These signals are defined when HDC is in Test mode or when drive interface is not specified. Otherwise, these signals are not defined since these are switched to output signals.

*2 This signal is defined when HDC is in test mode, otherwise this signal is not defined.

Item			Symbol	4 MHz Version HD63463-4 min	4 MHz Version HD63463-4 max	6 MHz Version HD63463-6 min	6 MHz Version HD63463-6 max	8 MHz Version HD63463-8 min	8 MHz Version HD63463-8 max	Unit	Test Condition
Signal Capacity	$\overline{RES}$ $\overline{DONE}$ RS R/$\overline{W}$ $\overline{CS}$ $\overline{DACK}$ $D_0 - D_{15}$ CLK $\overline{IRQ}$		C_{pin}	–	17	–	17	–	17	pF	V_{in} = 0 V Ta = 25°C f = 1.0 MHz
	ST506	SMD									
	RWDATA	RWDATA									
	WCLK	WCLK									
	RCLK	RCLK									
	–	SKEND									
	–	BUS_0/BUS_5									
	–	BUS_1/BUS_6									
	–	BUS_2/BUS_7									
	WFLT	BUS_3/BUS_8									
	–	BUS_4/BUS_9									
	IDX/TRK0	IDX									
	SCP	SEC									
	USELD	USELD									
	READY	–									
Current Consumption			I_{CC}	–	50	–	65	–	80	mA	• Data bus in read/write operation • Command execution in progress

AC CHARACTERISTICS (V_{CC} = 5V ± 5%, V_{SS} = 0V, Ta = 0 – 70°C, unless otherwise noted.)

Clock Timing

No.	Item	Symbol	4MHz Version HD63463-4		6MHz Version HD63463-6		8MHz Version HD63463-8		Unit	Test Condition
			min	max	min	max	min	max		
1	Clock Cycle time	tCYC	250	500	167	500	125	500	ns	See Figure 15
2	Clock Low Level Pulse Width	tPWCL	115	250	75	250	55	250	ns	
3	Clock High Level Pulse Width	tPWCH	115	250	75	250	55	250	ns	
4	Clock Rise Time	tCR	—	10	—	10	—	10	ns	
5	Clock Fall Time	tCF	—	10	—	10	—	10	ns	
6	Write Clock Cycle time	tWCYC	100	250	62.5	250	50	250	ns	
7	Write Clock Low Level Pulse Width	tPWCL	45	115	25	115	20	115	ns	
8	Write Clock High Level Pulse Width	tWCH	45	115	25	115	20	115	ns	
9	Write Clock Rise Time	tWCR	—	10	—	10	—	10	ns	
10	Write Clock Fall Time	tWCF	—	10	—	10	—	10	ns	
11	Read Clock Cycle Time	tRCYC	100	250	62.5	250	50	250	ns	
12	Read Clock Low Level Pulse Width	tRCL	45	115	25	115	20	115	ns	
13	Read Clock High Level Pulse Width	tRCH	45	115	25	115	20	115	ns	
14	Read Clock Rise Time	tRCR	—	10	—	10	—	10	ns	
15	Read Clock Fall Time	tRCF	—	10	—	10	—	10	ns	

Data Bus Configuration and $\overline{IRQ}$

No.	Item	Symbol	4MHz Version HD63463-4		6MHz Version HD63463-6		8MHz Version HD63463-8		Unit	Test Condition
			min	max	min	max	min	max		
21	$\overline{RES}$ Input Pulse Width	tRES	10	—	10	—	10	—	tcyc	See Figure 16
22	$\overline{DACK}$ Setup Time for $\overline{RES}$	tDACKSR	100	—	100	—	100	—	ns	
23	$\overline{DACK}$ Hold Time For $\overline{RES}$	tDACKHR	0	120	0	80	0	50	ns	
24	$\overline{IRQ}$ Delay Time 1	tIRD1	—	250	—	200	—	150	ns	See Figure 17
26	$\overline{RES}$ Rise Time	tRESR	—	10	—	10	—	10	μs	See Figure 16

MPU Interface

No.	Item	Symbol	4MHz Version HD63463-4		6MHz Version HD63463-6		8MHz Version HD63463-8		Unit	Test Condition
			min	max	min	max	min	max		
31	R/$\overline{W}$ Setup Time for $\overline{CS}$ Assert	tRWS	70	—	60	—	50	—	ns	See Figures 18 and 19
32	R/$\overline{W}$ Hold Time	tRWH	0	—	0	—	0	—	ns	
33	RS Setup Time for $\overline{CS}$ Assert	tRSS	70	—	60	—	50	—	ns	
34	RS Hold Time	tRSH	0	—	0	—	0	—	ns	
35	$\overline{CS}$ Setup Time	tCSS	50	—	40	—	40	—	ns	
36	$\overline{CS}$ Negate Hold Time	tCSNH	40	—	40	—	40	—	ns	
37	$\overline{CS}$ Negate Width	tCSNW	90	—	80	—	80	—	ns	
38	Write Data Setup Time	tWDS	80	—	60	—	40	—	ns	
39	Write Data Hold Time	tWDH	10	—	10	—	10	—	ns	
40	$\overline{DTACK}$ Delay Time	tDTKZL	—	90	—	85	—	80	ns	
41	$\overline{DTACK}$ Hold Time	tDTKLH	—	60	—	60	—	60	ns	
43	Data Bus 3 State Recovery Time	tDBR	0	—	0	—	0	—	ns	
44	Read Data Access Time	tRDAC	—	90	—	80	—	70	ns	
45	Read Data Hold Time	tRDH	10	—	10	—	10	—	ns	
46	$\overline{CS}$ Fall Time	tCSF	—	1	—	1	—	1	tcyc	
47	$\overline{CS}$ Rise Time	tCSR	—	1	—	1	—	1	tcyc	

DMA Interface

No.	Item	Symbol	4MHz Version HD63463-4		6MHz Version HD63463-6		8MHz Version HD63463-8		Unit	Test Condition
			min	max	min	max	min	max		
50	$\overline{\text{DREQ}}$ Assert Delay Time 1	tDRAD1	—	90	—	80	—	80	ns	See Figures 20 and 21
51	$\overline{\text{DREQ}}$ Negate Delay Time 1 (Cycle Steal Mode)	tDRND1	—	90	—	80	—	80	ns	
52	$\overline{\text{DREQ}}$ Assert Delay Time 2 (Cycle Steal Mode)	tDRAD2	—	90	—	80	—	80	ns	
53	$\overline{\text{DREQ}}$ Negate Delay Time 2	tDRND2	—	90	—	80	—	80	ns	
54	$\overline{\text{DREQ}}$ Negate Delay Time 3 ($\overline{\text{DONE}}$ Assert)	tDRND3	—	90	—	80	—	80	ns	
55	DMA R/$\overline{\text{W}}$ Setup Time	tDRWS	70	—	60	—	50	—	ns	
56	DMA R/$\overline{\text{W}}$ Hold Time	tDRWH	0	—	0	—	0	—	ns	
57	$\overline{\text{DACK}}$ Setup Time	tDACKS	50	—	40	—	40	—	ns	
58	$\overline{\text{DACK}}$ Negate Hold Time	tDACKHN	40	—	40	—	40	—	ns	
59	$\overline{\text{DACK}}$ Negate Width	tDAKNW	90	—	80	—	80	—	ns	
60	DMA Write Data Setup Time	tDWDS	80	—	60	—	40	—	ns	
61	DMA Write Data Hold Time	tDWDH	10	—	10	—	10	—	ns	
62	DMA $\overline{\text{DTACK}}$ Delay Time	tDDTZL	—	90	—	85	—	80	ns	
63	DMA $\overline{\text{DTACK}}$ Hold Time	tDDTLH	—	60	—	60	—	60	ns	

(to be continued)

No.	Item	Symbol	4MHz Version HD63463-4		6MHz Version HD63463-6		8MHz Version HD63463-8		Unit	Test Condition
			min	max	min	max	min	max		
65	$\overline{\text{DONE}}$ Input Pulse Width	tPWDN	1.5	—	1.5	—	1.5		tcyc	See Figures 20 and 21
66	DMA Data Bus 3 State Recovery Time	tDDBR	0	—	0	—	0	—	ns	
67	DMA Read Data Access Time	tDRDAC	—	90	—	80	—	70	ns	
68	DMA Read Data Hold Time	tDRDH	10	—	10	—	10	—	ns	
69	$\overline{\text{DACK}}$ Fall Time	tDACKF	—	1	—	1	—	1	tcyc	
70	$\overline{\text{DACK}}$ Rise Time	tDACKR	—	1	—	1	—	1	tcyc	

ST506 Interface ce

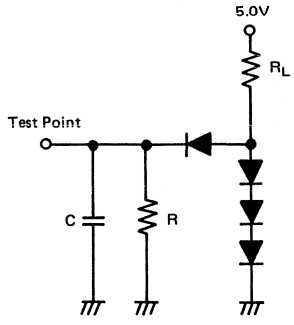
No.	Item	Symbol	4MHz Version HD63463-4		6MHz Version HD63463-6		8MHz Version HD63463-8		Unit	Test Condition
			min	max	min	max	min	max		
110	USELD Setup Time (for USEL)	tUSLDS	—	5	—	5	—	5	tcyc	See Figure 22
111	WGATE Delay Time for Index	tWGTIDX	—	200	—	150	—	100	ns	See Figure 23
112	WFLT Pulse Width	tWFLT	2	—	2	—	2	—	tcyc	
113	Index Pulse Width	tIDXW	8tcyc 24twcyc	— —	8tcyc 24twcyc	— —	8tcyc 24twcyc	— —	—	(Note 1) See Figure 23
114	WGATE Delay Time	tWGTD	—	135	—	130	—	125	ns	See Figure 24
116	Write Data Delay Time	tWDD	—	150	—	130	—	125	ns	
118	LATE/EARLY Delay Time	tELD	—	150	—	130	—	125	ns	
120	LATE/EARLY Setup Time (for Write Data)	tWDS	0	—	0	—	0	—	ns	
121	RGATE Delay Time	tRGTD	—	135	—	130	—	125	ns	See Figure 25
123	SYNC Delay Time	tSYND	—	135	—	130	—	125	ns	
125	Read Data Setup Time	tRDS	20	—	20	—	15	—	ns	
126	Read Data Hold Time	tRDH	20	—	20	—	15	—	ns	
127	USELD-DIR Time	tUS-DIR	70	—	70	—	70	—	tcyc	See Figure 26
128	STEP-USEL Time	tSTPUS	80	—	80	—	80	—	tcyc	
129	DIR-STEP Time	tDIRSTP	270	—	270	—	270	—	tcyc	
130	STP-DIR Time	tSTPDIR	80	—	80	—	80	—	tcyc	
131	SEEK-USEL Time	tSEKUS	1	—	1	—	1	—	tcyc	
132	SCP Wait Time	tSCP	—	1.0×10^7	—	1.0×10^7	—	1.0×10^7	tcyc	

(Note 1) The index pulse width must satisfy min 8 tcyc and min 24 twcyc.

SMD Interface

No.	Item	Symbol	4MHz Version HD63463-4		6MHz Version HD63463-6		8MHz Version HD63463-8		Unit	Test Condition
			min	max	min	max	min	max		
150	Index Pulse Width	tIDXW	8tcyc 12twcyc	50tcyc	8tcyc 12twcyc	50tcyc	8tcyc 12twcyc	50tcyc	—	(Note 2) See Figure 32
151	USELD Setup Time	tUSLDS	—	7	—	7	—	7	tcyc	See Figure 31
152	Sector Pulse Width	tPWSET	8tcyc 12twcyc	50tcyc	8tcyc 12twcyc	50tcyc	8tcyc 12twcyc	50tcyc	—	(Note 2) See Figure 32
153	WGATE Delay Time for SEC/IDX	tWGTSEC	—	200	—	150	—	125	ns	See Figure 32
156	REGATE Delay Time	tRGTD	—	135	—	130	—	125	ns	See Figure 33
158	Read Data Setup Time	tRDS	20	—	20	—	15	—	ns	
159	Read Data Hold Time	tRDH	20	—	20	—	15	—	ns	
165	WGATE Delay Time	tWGTD	—	135	—	130	—	125	ns	See Figure 34
169	Write Data Delay Time	tWDD	—	135	—	130	—	125	ns	

(Note 2) The index sector pulse width must satisfy min 8 tcyc and min 12 twcyc.



Signal		Load Condition
$D_0 - D_{15}$ $\overline{DTACK}$ $\overline{DREQ}$ RWDATA		RL = 1.8 kΩ C = 40 pF
ST506	SMD	
SYNC	–	R = 10 kΩ
WGATE	BUS_0/BUS_5	
EARLY/ RGATE	BUS_1/BUS_6	
LATE/ STEP	BUS_2/BUS_7	All diodes are
–	BUS_3/BUS_8	1S2074 Ⓗ's or
LCT/DIR	BUS_4/BUS_9	the equivalent.
USEL0	TAG1	
USEL1	TAG2	
HSEL0	TAG3	
HSEL1	UTAG	
HSEL2	TAG5	
SEEK	BUSR/$\overline{W}$	
–	BUSL/$\overline{H}$	

Figure 13 Test Load Circuit A

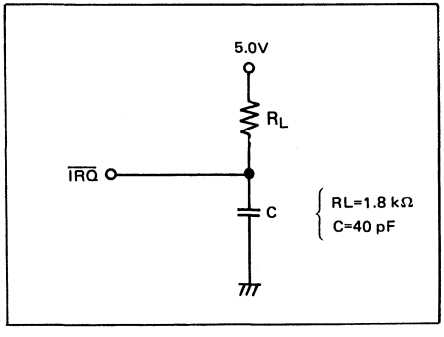


Figure 14 Test Load Circuit B

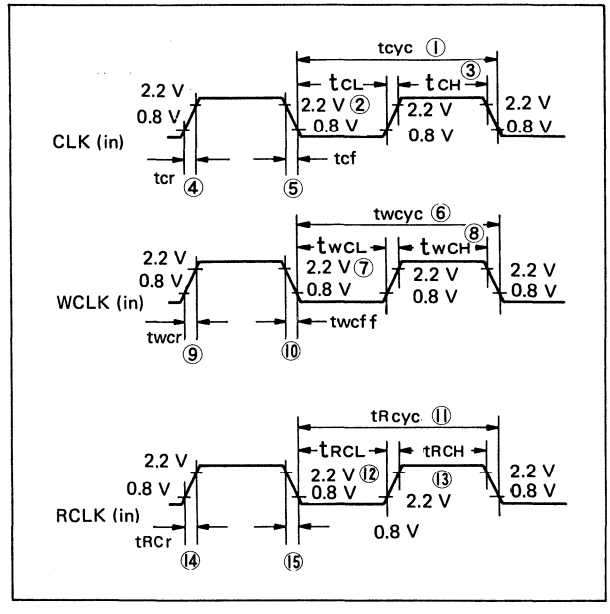


Figure 15 CLOCK

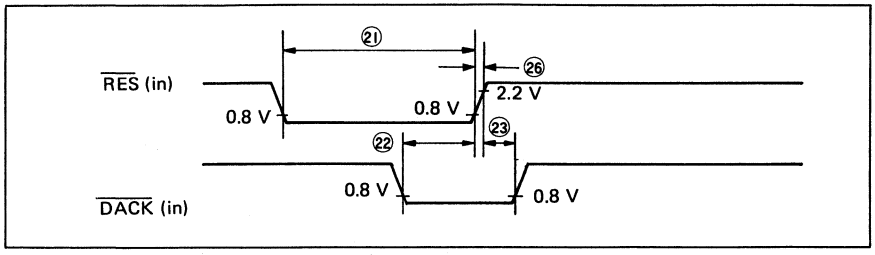


Figure 16 RES – $\overline{DACK}$ Input Timing (Data Bus Width Selection)

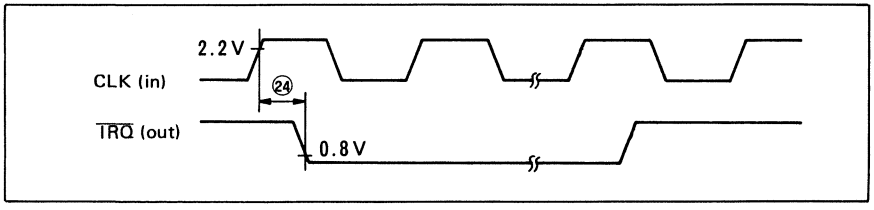
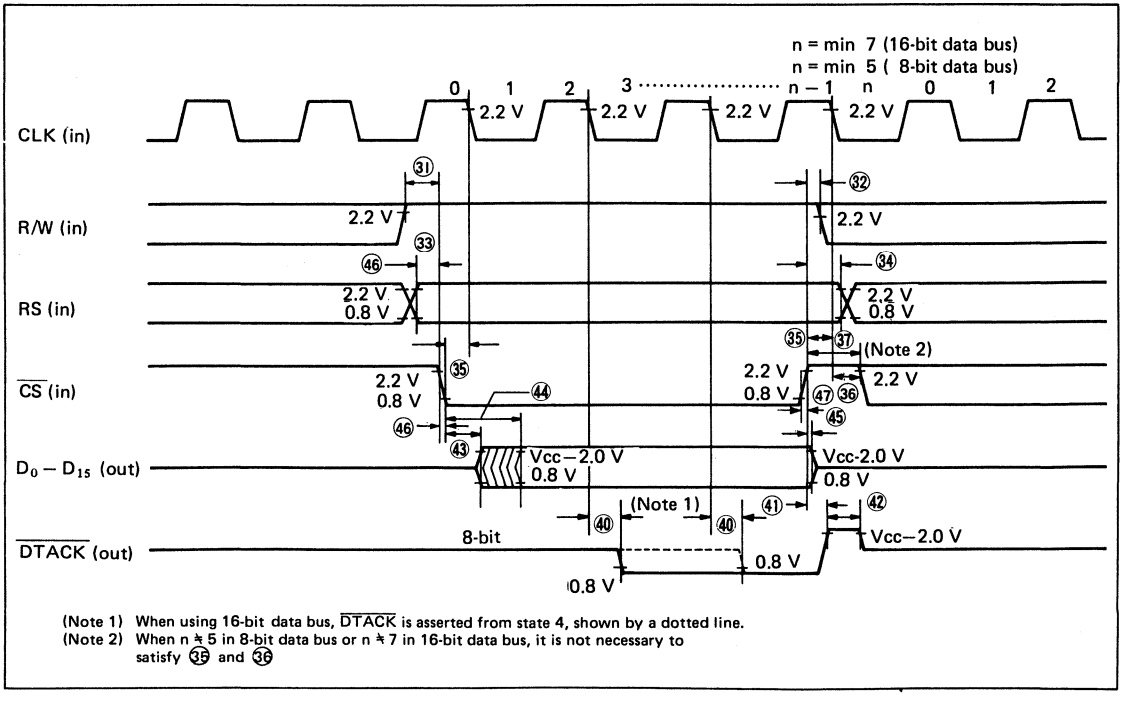


Figure 17 $\overline{IRQ}$ Output Timing



(Note 1) When using 16-bit data bus, $\overline{DTACK}$ is asserted from state 4, shown by a dotted line.
(Note 2) When n ≠ 5 in 8-bit data bus or n ≠ 7 in 16-bit data bus, it is not necessary to satisfy ㉟ and ㊱

Figure 18 MPU Read Cycle
HDC → MPU

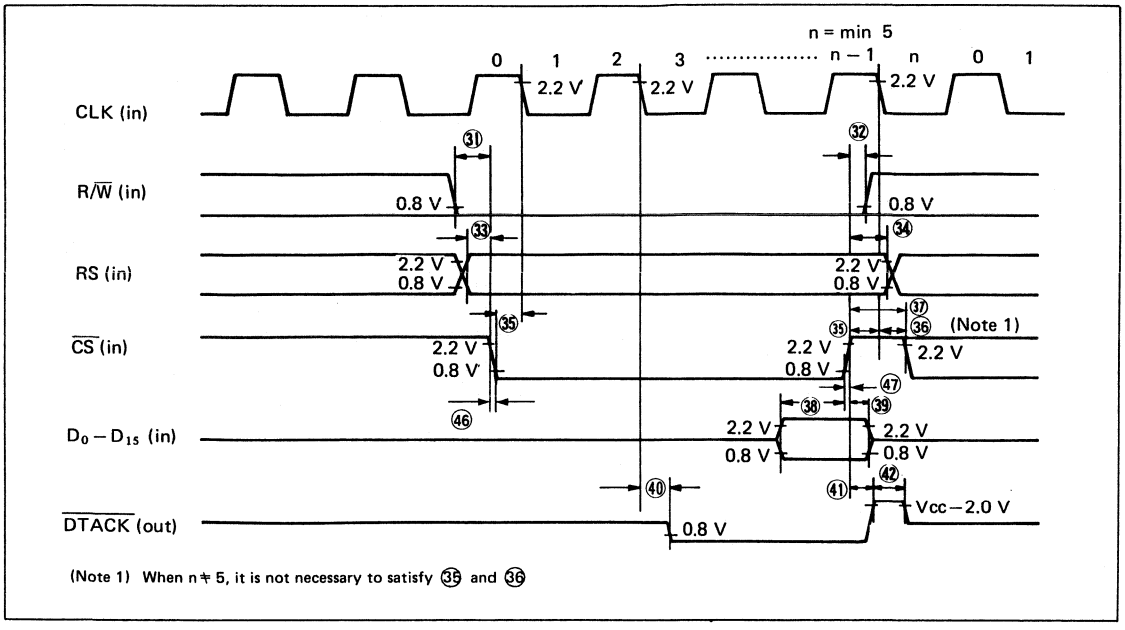


Figure 19 MPU Write Cycle
MPU → HDC
(8 or 16-bit data bus)

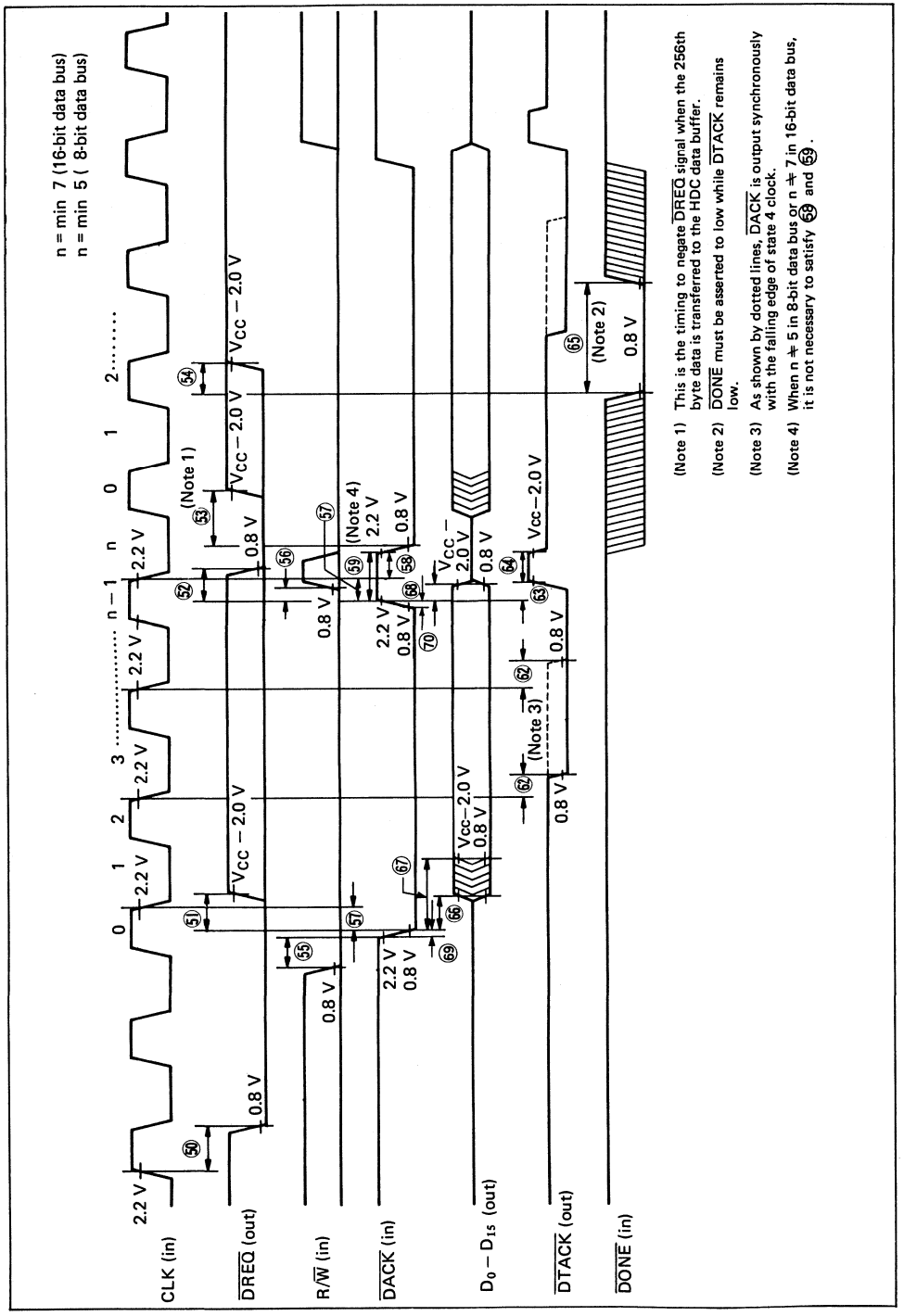
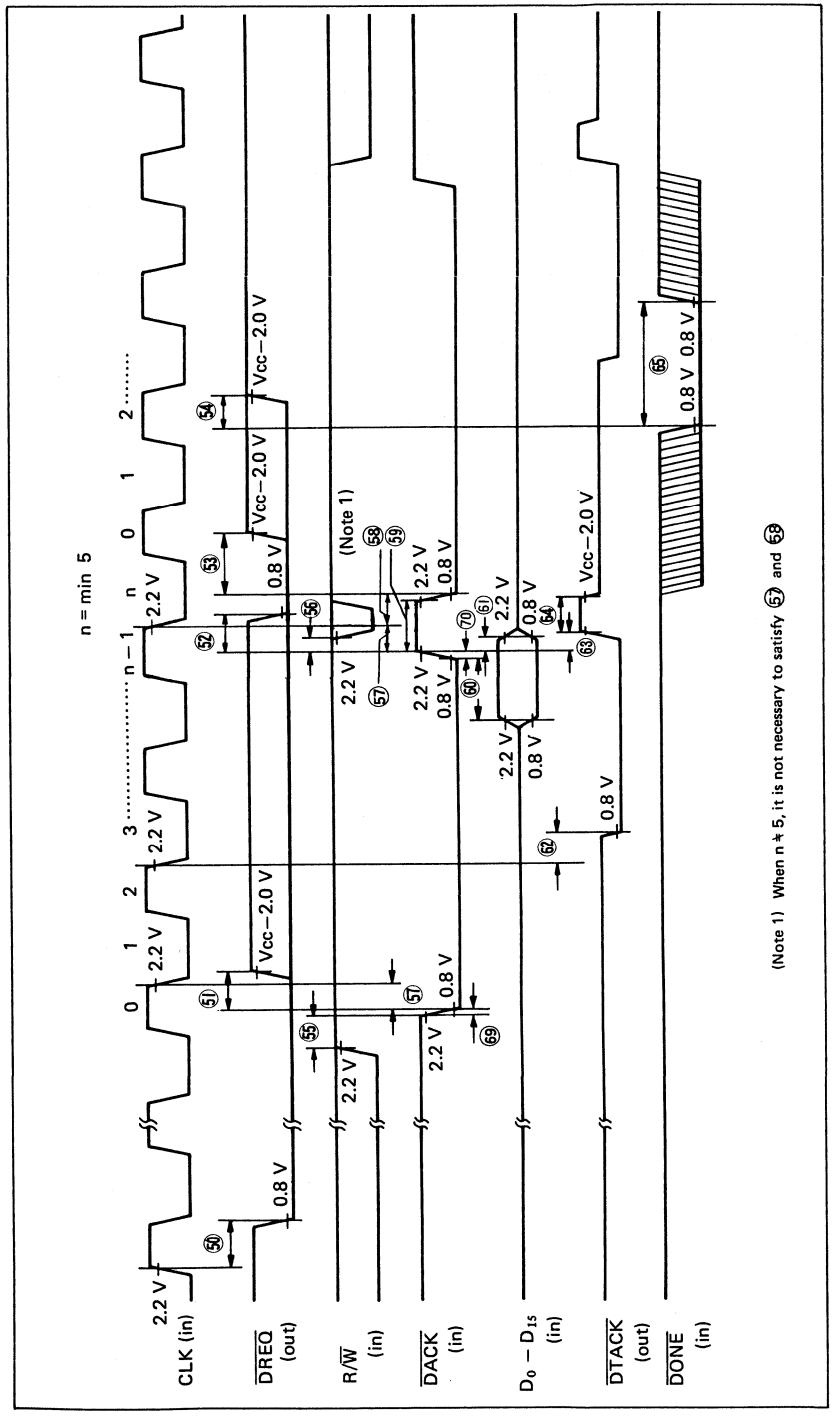


Figure 20 DMA Read Cycle
HDC→ Memory



(Note 1) When n ≠ 5, it is not necessary to satisfy 57 and 69

Figure 21 DMA Write Cycle
Memory → HDC

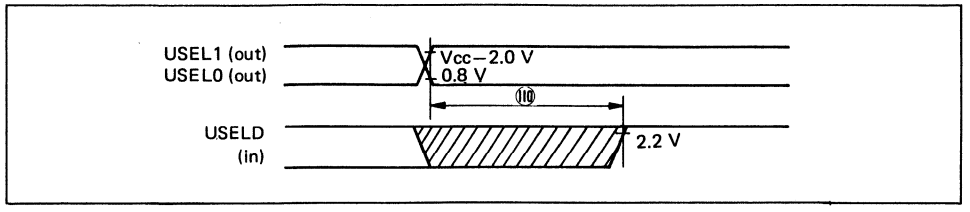
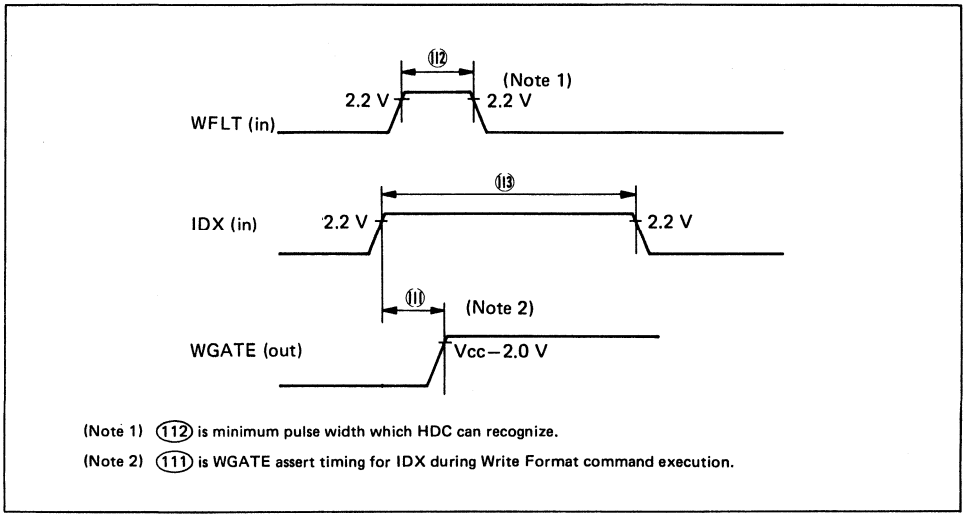


Figure 22 USEL, USELD Timing (ST506)



(Note 1) (112) is minimum pulse width which HDC can recognize.

(Note 2) (111) is WGATE assert timing for IDX during Write Format command execution.

Figure 23 WFLT, IDX, WGATE Timing (ST506)

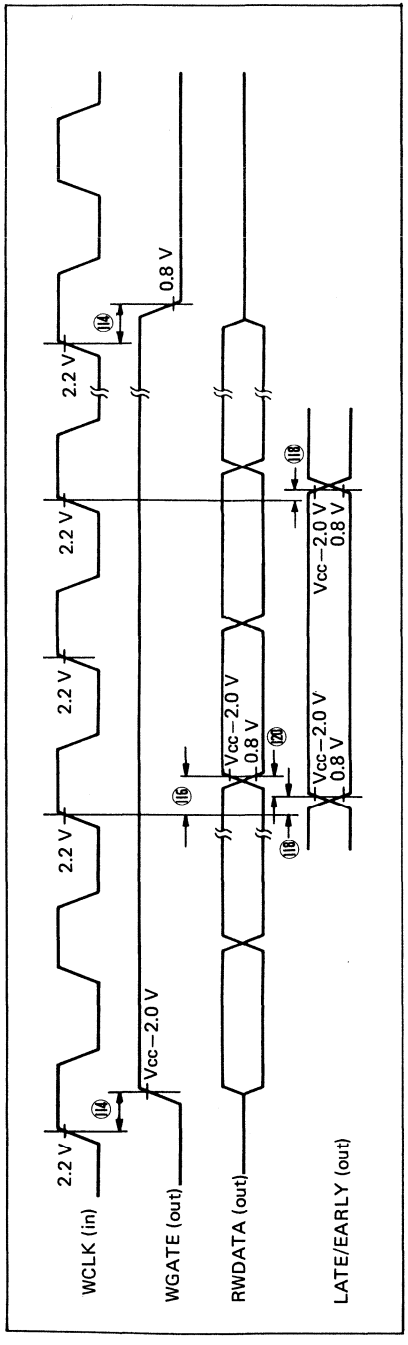


Figure 24 Write Operation (ST506)

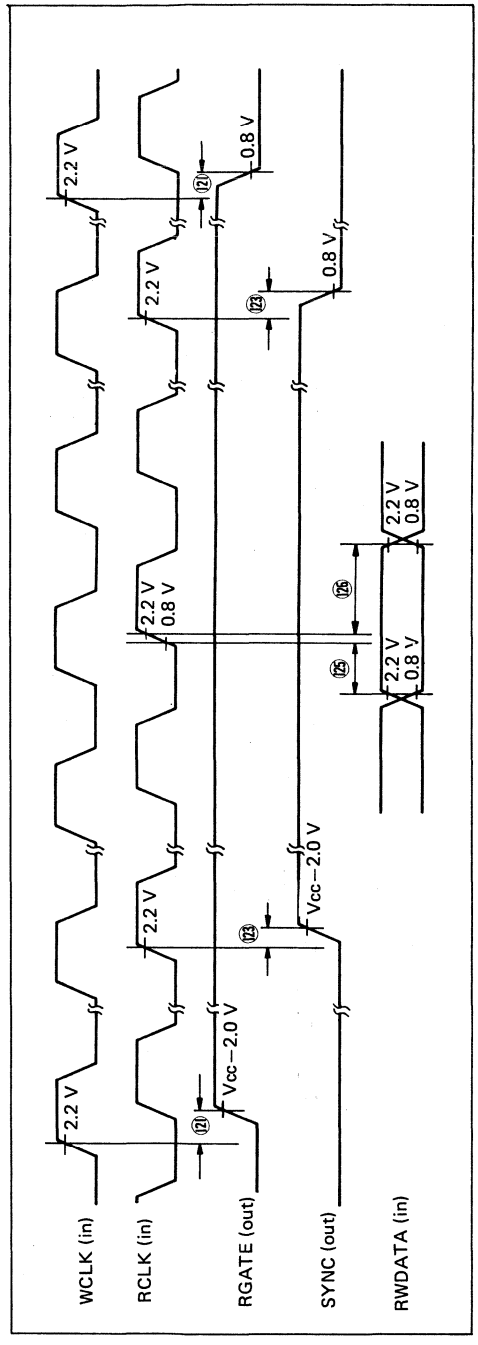


Figure 25 Read Operation (ST506)

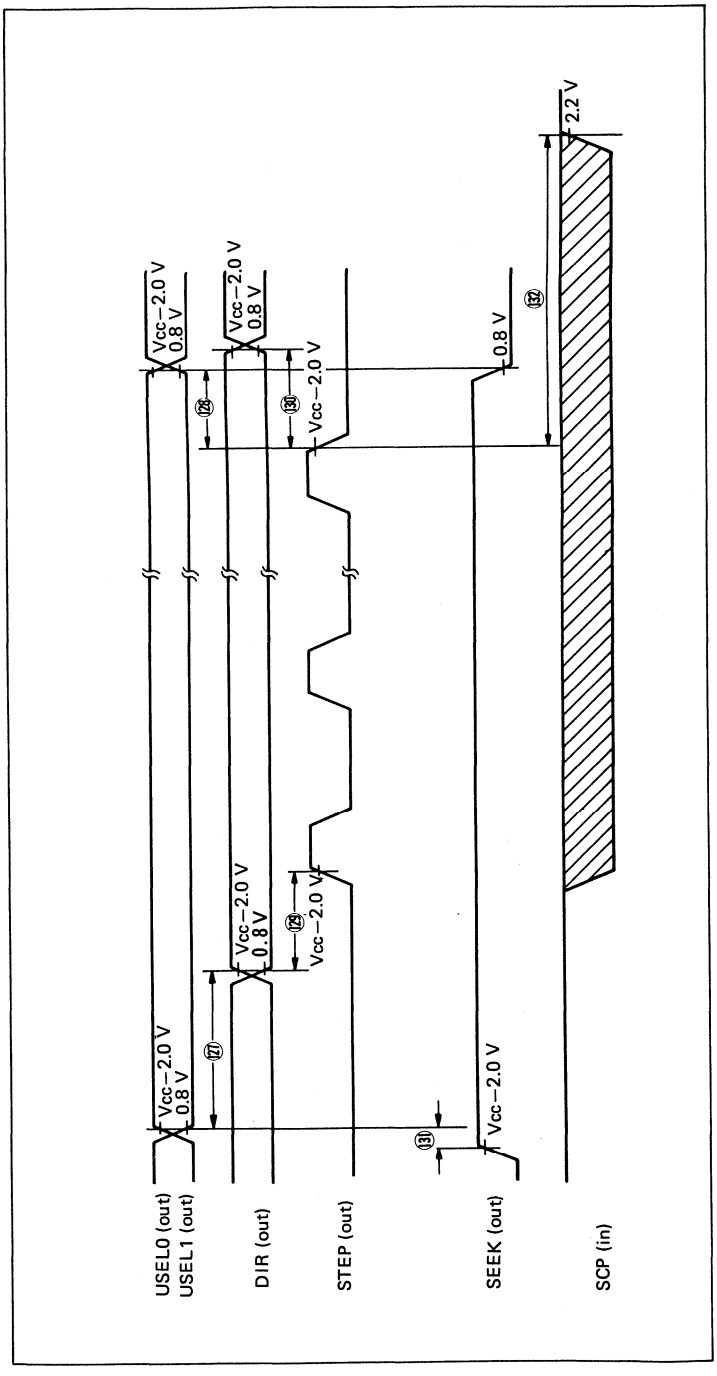


Figure 26 Head Positioning Operation (ST506)

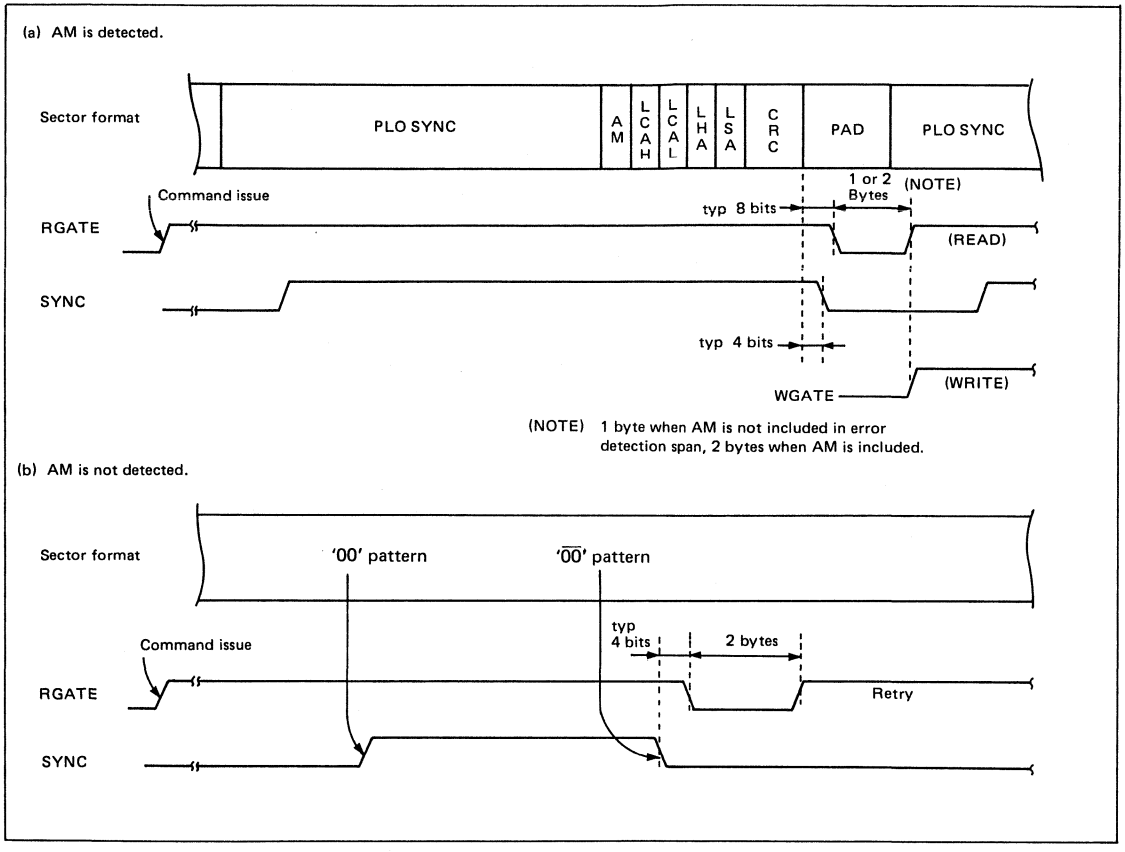


Figure 27 ID Search Timing Chart (ST506)

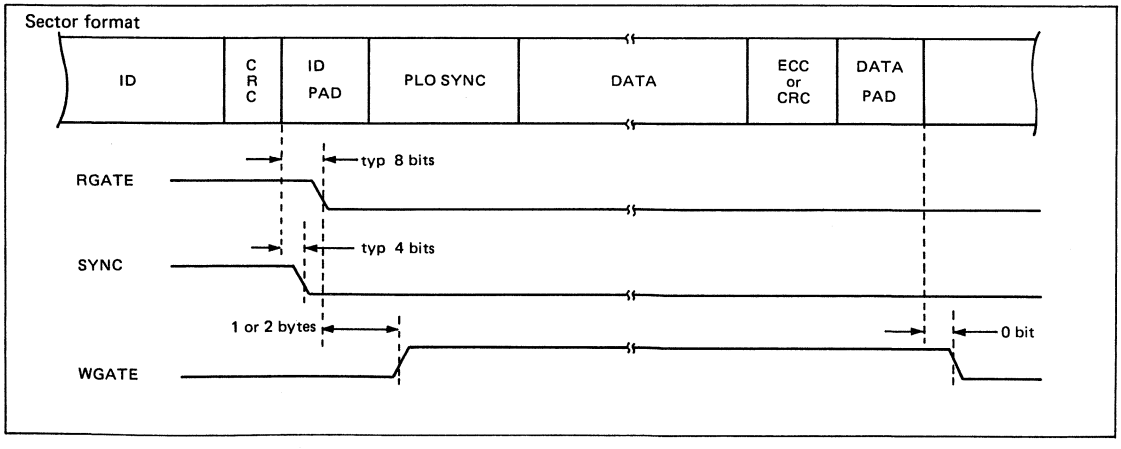


Figure 28 Write Data Timing (ST506)

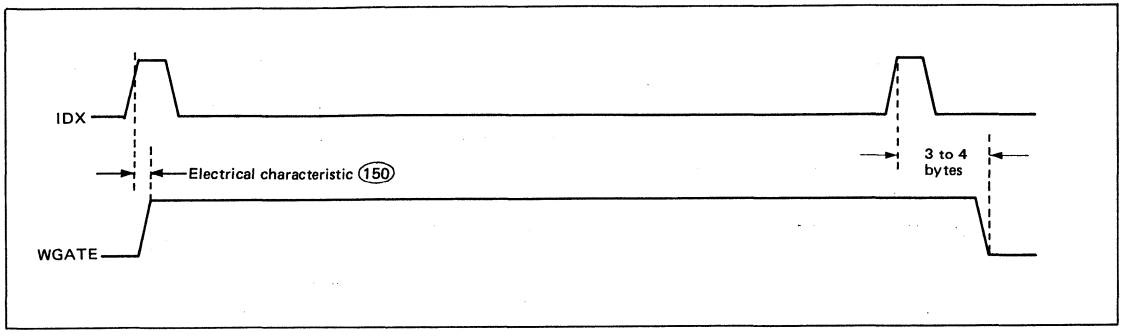


Figure 29 Write Format Timing (ST506)

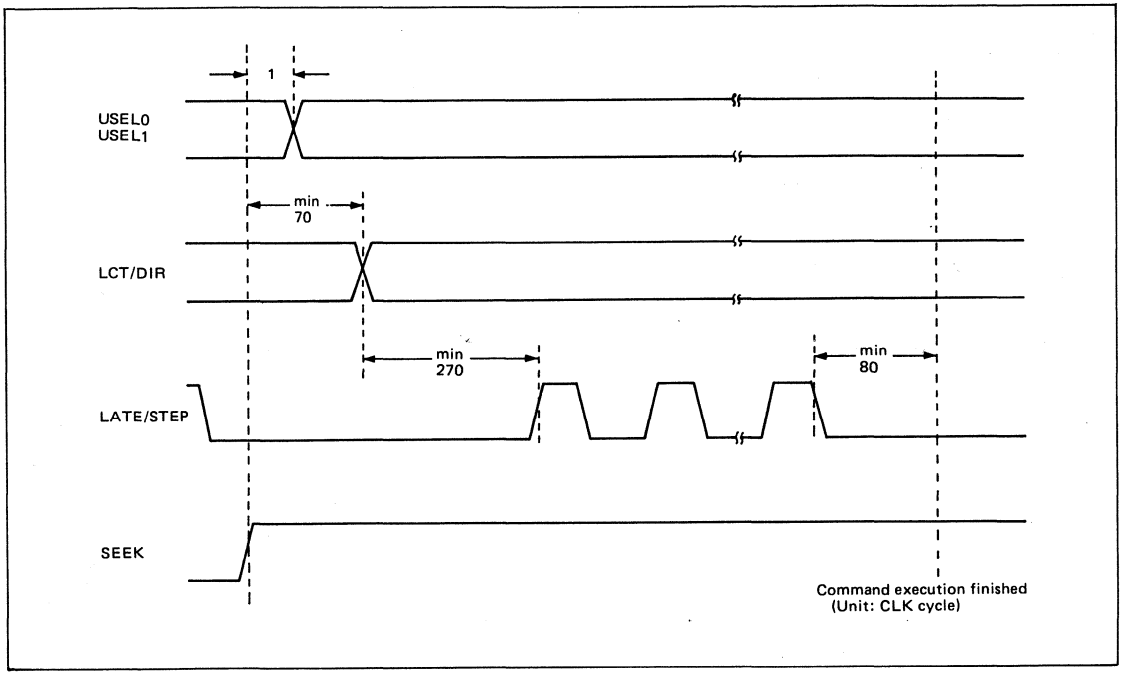


Figure 30 Seek, Recalibrate Command Execution Timing (ST506)

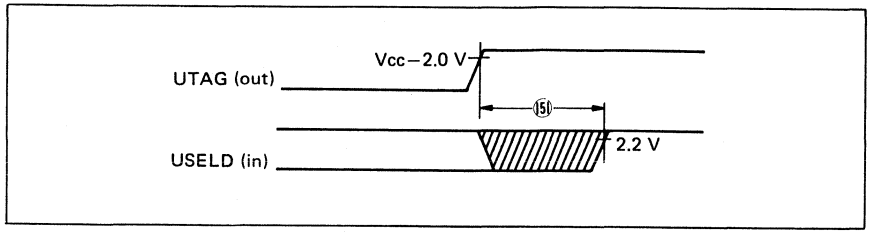


Figure 31 UTAG, USELD Timing (SMD)

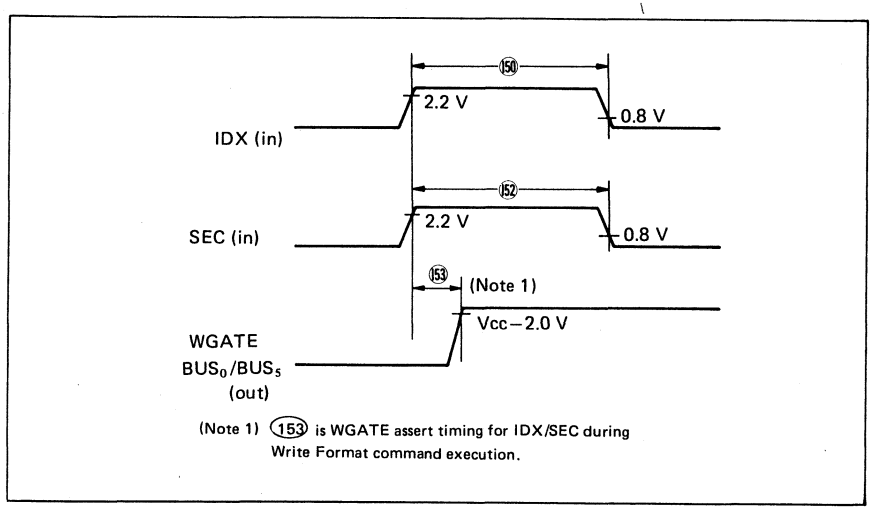


Figure 32 IDX/SEC, WGATE Timing (SMD)

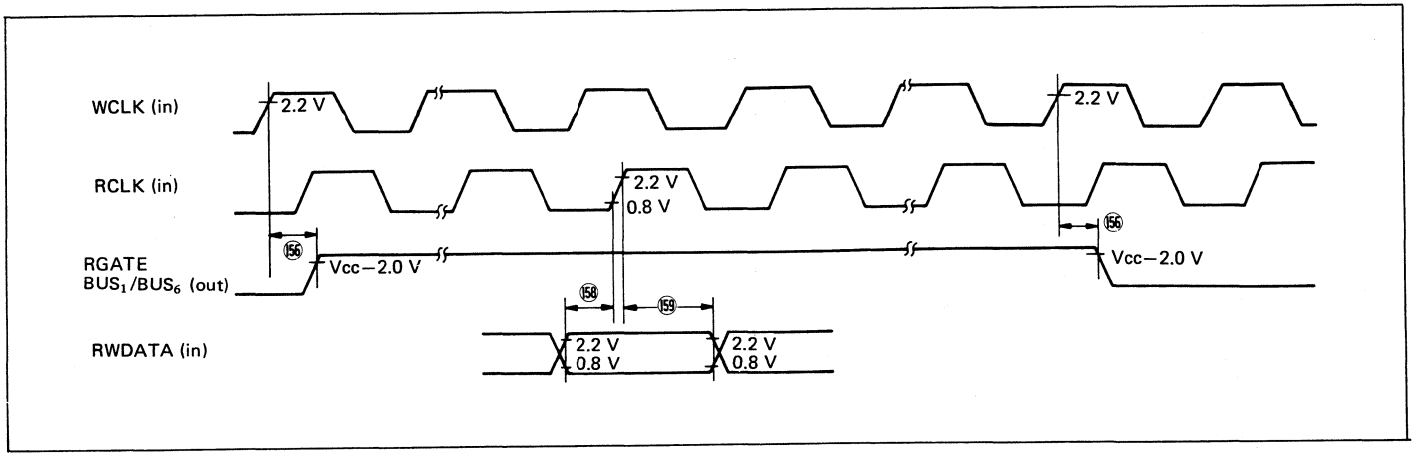


Figure 33 Read Operation (SMD)

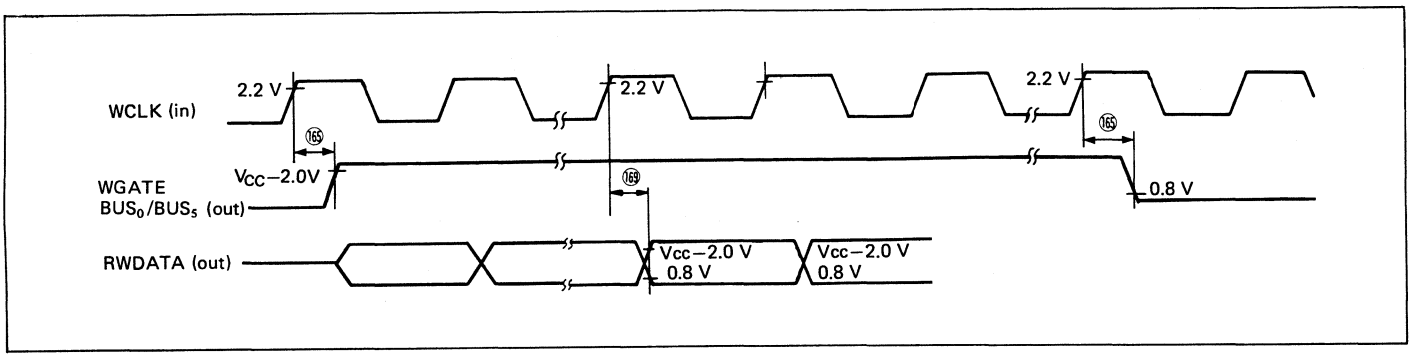
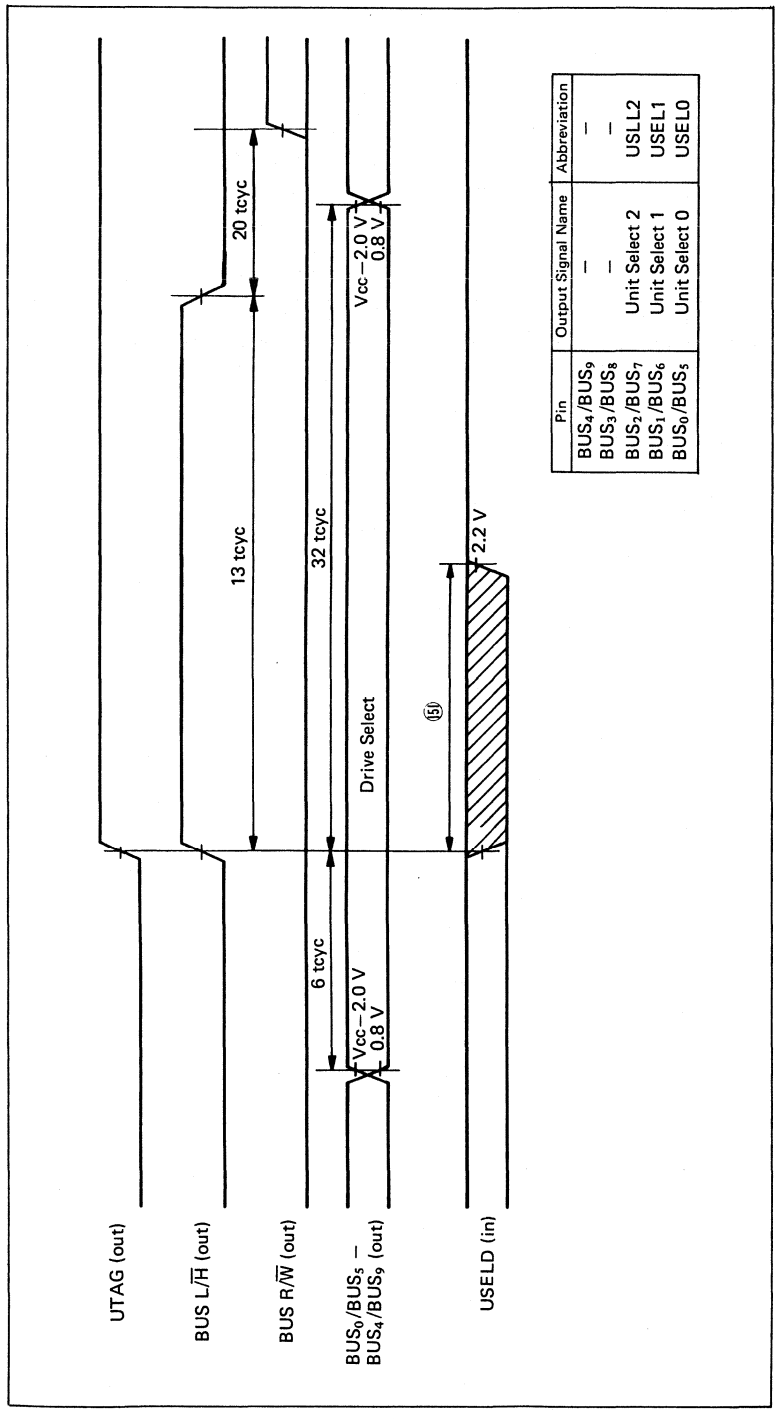
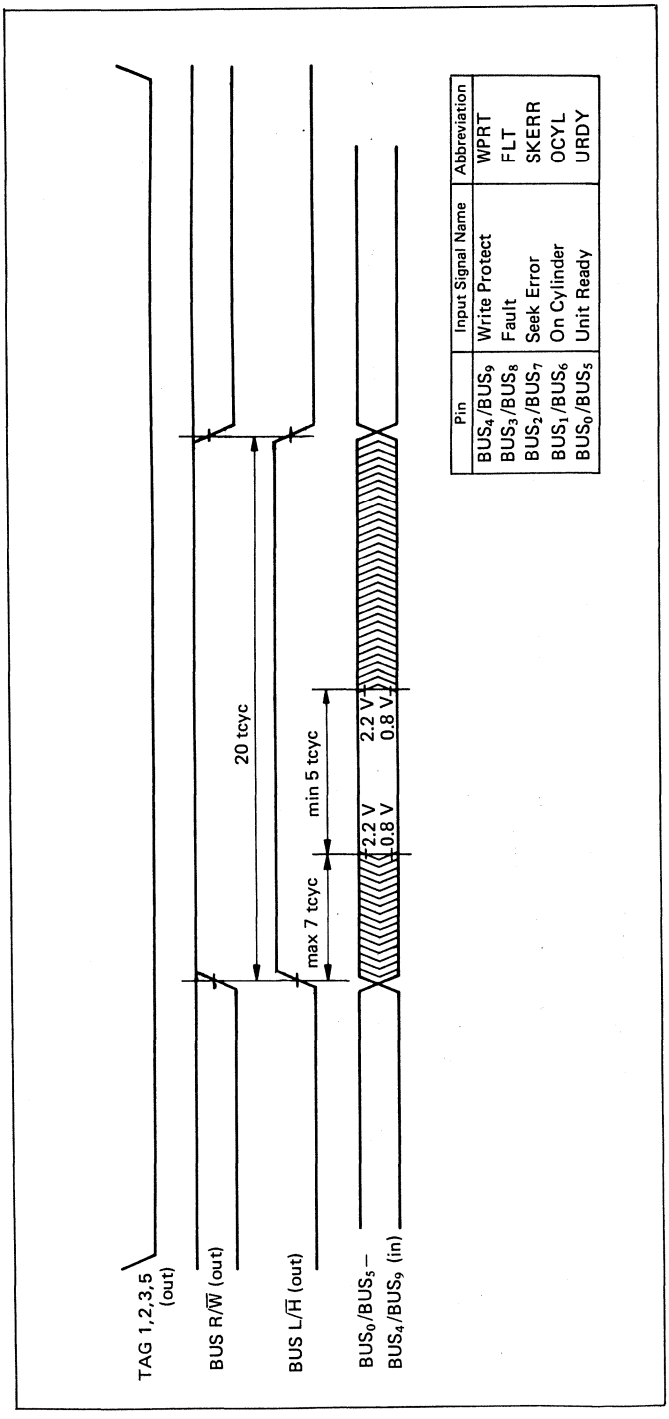


Figure 34 Write Operation (SMD)



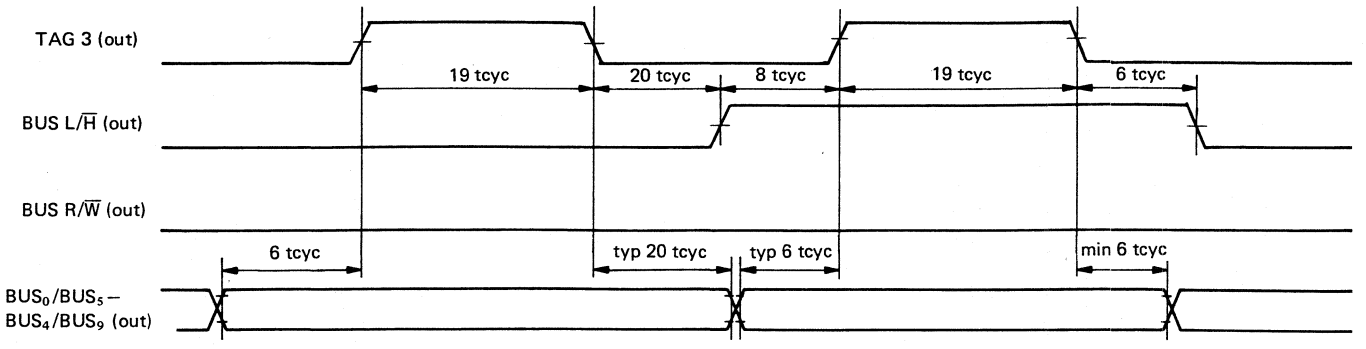
Pin	Output Signal Name	Abbreviation
BUS_4/BUS_9	–	–
BUS_3/BUS_8	–	–
BUS_2/BUS_7	Unit Select 2	USLL2
BUS_1/BUS_6	Unit Select 1	USEL1
BUS_0/BUS_5	Unit Select 0	USEL0

Figure 35 Drive Select Operation (SMD)



Pin	Input Signal Name	Abbreviation
BUS$_4$/BUS$_9$	Write Protect	WPRT
BUS$_3$/BUS$_8$	Fault	FLT
BUS$_2$/BUS$_7$	Seek Error	SKERR
BUS$_1$/BUS$_6$	On Cylinder	OCYL
BUS$_0$/BUS$_5$	Unit Ready	URDY

Figure 36 Drive Check Operation (SMD)



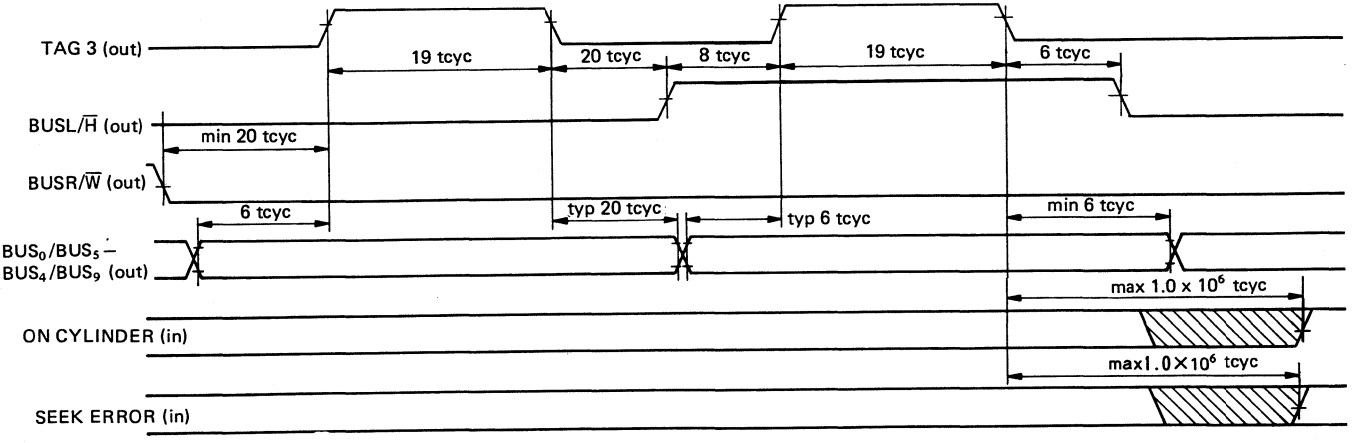
BUSL/H = low

Pin	Output Signal Name	Abbreviation	Level
BUS_4/BUS_9	(Reserve)	0	low
BUS_3/BUS_8	Strobe Late	STBL	low
BUS_2/BUS_7	Strobe Early	STBE	low
BUS_1/BUS_6	Return to Zero	RTZ	low
BUS_0/BUS_5	Reserve	0	low

BUSL/H = high

Pin	Output Signal Name	Abbreviation	Level
BUS_4/BUS_9	Fault Clear	FCLR	high
BUS_3/BUS_8	Servo Offset Minus	SOFM	low
BUS_2/BUS_7	Servo Offset Plus	SOFP	low
BUS_1/BUS_6	Read Gate	RGATE	low
BUS_0/BUS_5	Write Gate	WGATE	low

Figure 37 Fault Clear Operation (SMD)



BUSL/$\overline{H}$ = low

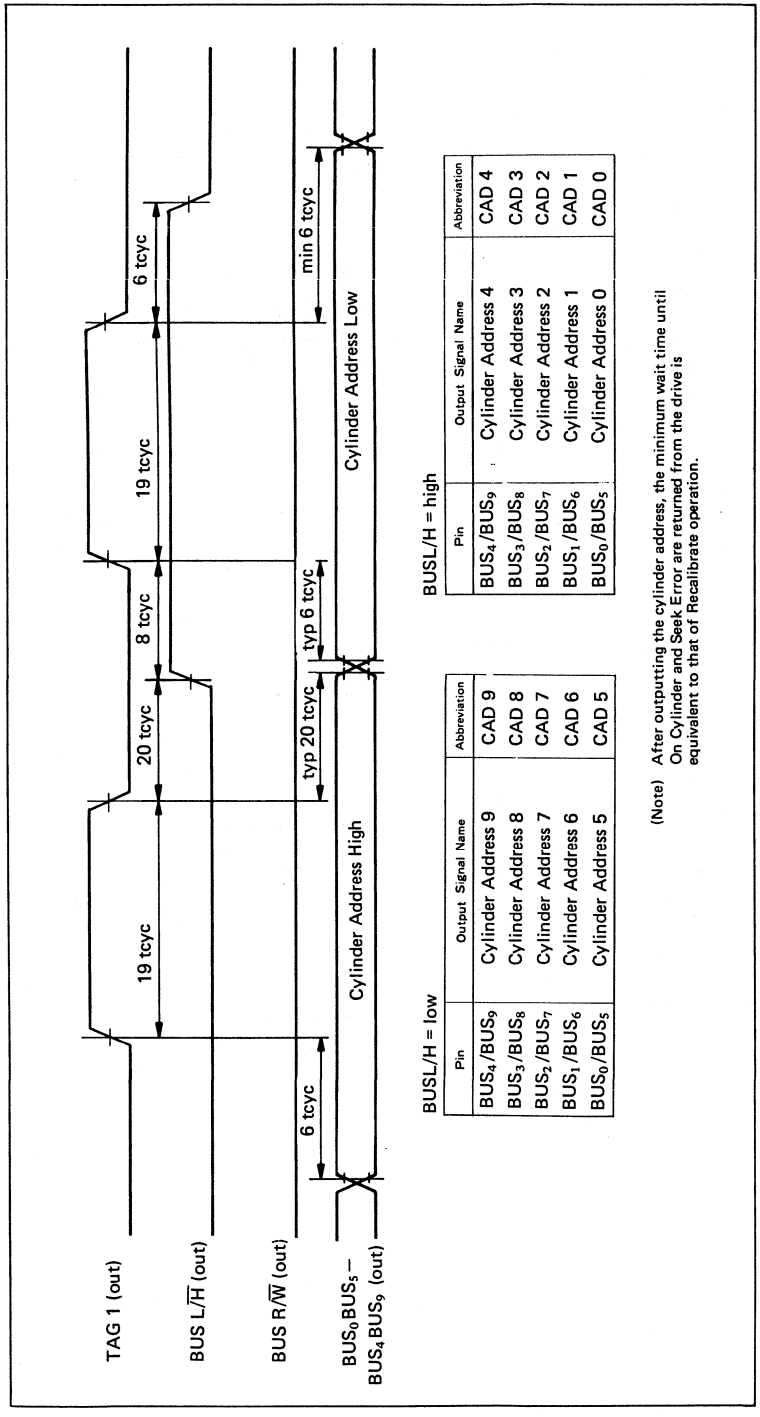
Pin	Output Signal Name	Abbreviation	Level
BUS_4/BUS_9	Reserve	O	low
BUS_3/BUS_8	Strobe Late	STBL	low
BUS_2/BUS_7	Strobe Early	STBE	low
BUS_1/BUS_6	Return to Zero	RTZ	high
BUS_0/BUS_5	Reserve	O	low

BUSL/$\overline{H}$ = high

Pin	Output Signal Name	Abbreviation	Level
BUS_4/BUS_9	Fault Clear	FCLR	low
BUS_3/BUS_8	Servo Offset Minus	SOFM	low
BUS_2/BUS_7	Servo Offset Plus	SOFP	low
BUS_1/BUS_6	Read Gate	RGATE	low
BUS_0/BUS_5	Write Gate	WGATE	low

(Note) On Cylinder and Seek Error are checked by Polling command.

Figure 38 Recalibrate Command Execution Timing (SMD)



BUSL/H = low

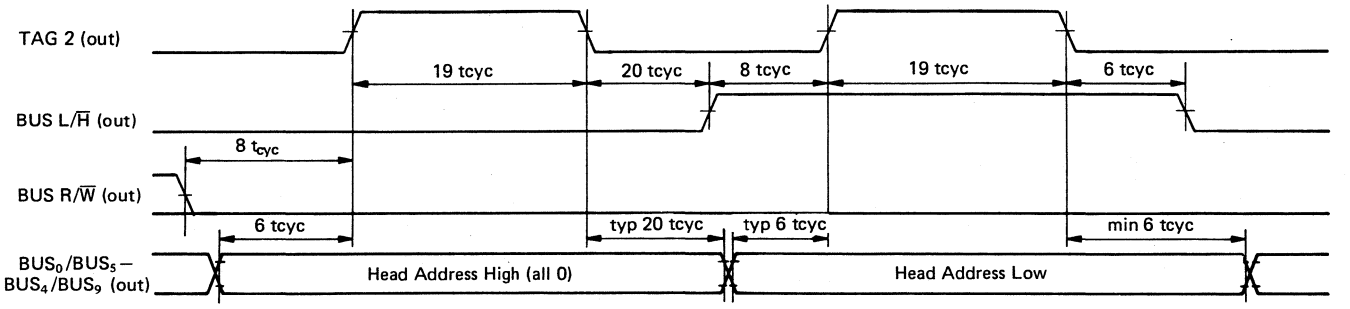
Pin	Output Signal Name	Abbreviation
BUS_4/BUS_9	Cylinder Address 9	CAD 9
BUS_3/BUS_8	Cylinder Address 8	CAD 8
BUS_2/BUS_7	Cylinder Address 7	CAD 7
BUS_1/BUS_6	Cylinder Address 6	CAD 6
BUS_0/BUS_5	Cylinder Address 5	CAD 5

BUSL/H = high

Pin	Output Signal Name	Abbreviation
BUS_4/BUS_9	Cylinder Address 4	CAD 4
BUS_3/BUS_8	Cylinder Address 3	CAD 3
BUS_2/BUS_7	Cylinder Address 2	CAD 2
BUS_1/BUS_6	Cylinder Address 1	CAD 1
BUS_0/BUS_5	Cylinder Address 0	CAD 0

(Note) After outputting the cylinder address, the minimum wait time until On Cylinder and Seek Error are returned from the drive is equivalent to that of Recalibrate operation.

Figure 39 Seek Command Execution Timing (SMD)



BUSL/$\overline{H}$ = low

Pin	Output Signal Name	Abbreviation
BUS_4/BUS_9	Head Address 9	HAD 9
BUS_3/BUS_8	Head Address 8	HAD 8
BUS_2/BUS_7	Head Address 7	HAD 7
BUS_1/BUS_6	Head Address 6	HAD 6
BUS_0/BUS_5	Head Address 5	HAD 5

BUSL/$\overline{H}$ = high

Pin	Output Signal Name	Abbreviation
BUS_4/BUS_9	Head Address 4	HAD 4
BUS_3/BUS_8	Head Address 3	HAD 3
BUS_2/BUS_7	Head Address 2	HAD 2
BUS_1/BUS_6	Head Address 1	HAD 1
BUS_0/BUS_5	Head Address 0	HAD 0

(Note) Head Address 5 – 9 are fixed at low level.

Figure 40 Head Selection Operation (SMD)

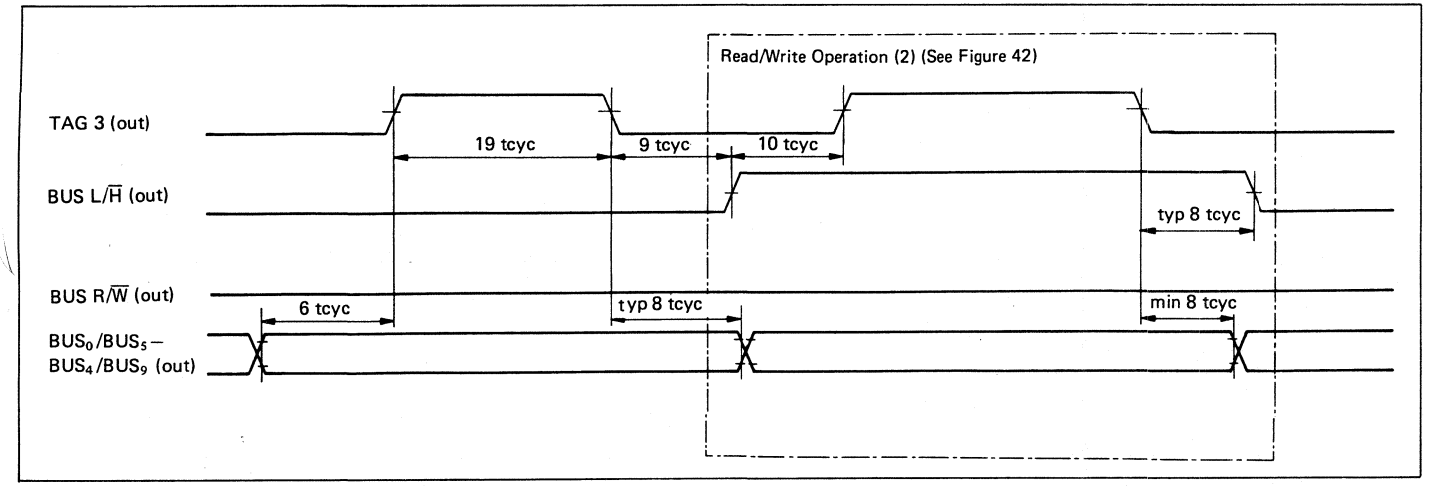


Figure 41 Read/Write Operation (1) (SMD)

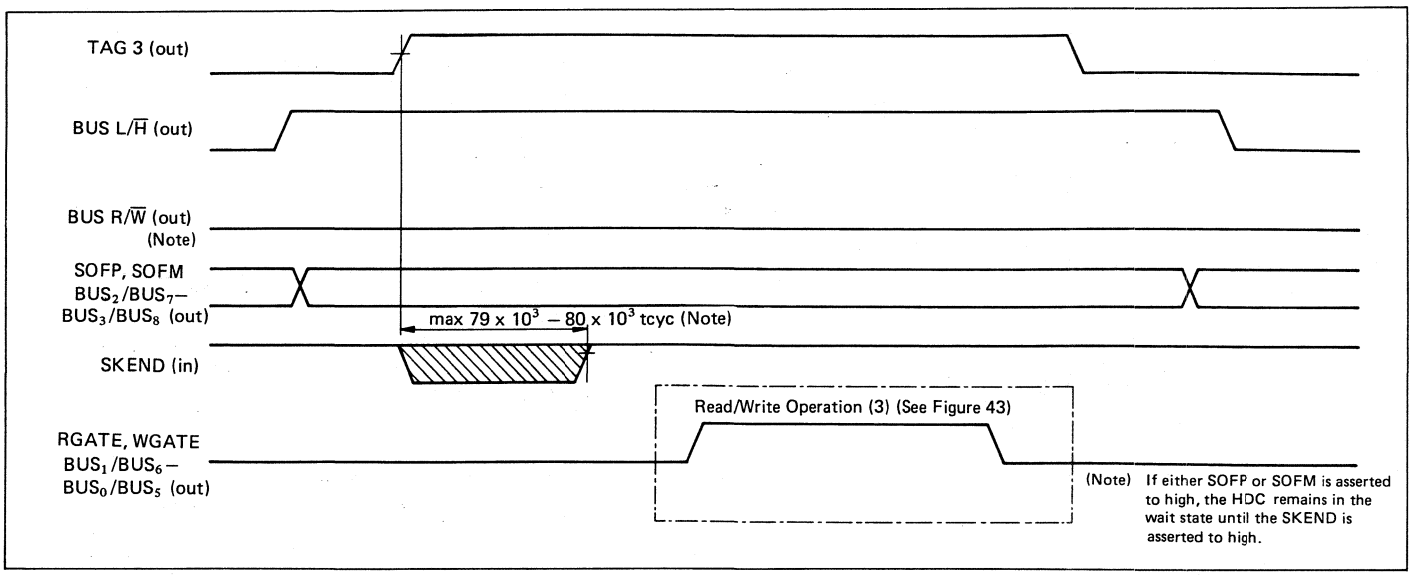


Figure 42 Read/Write Operation (2) (SMD)

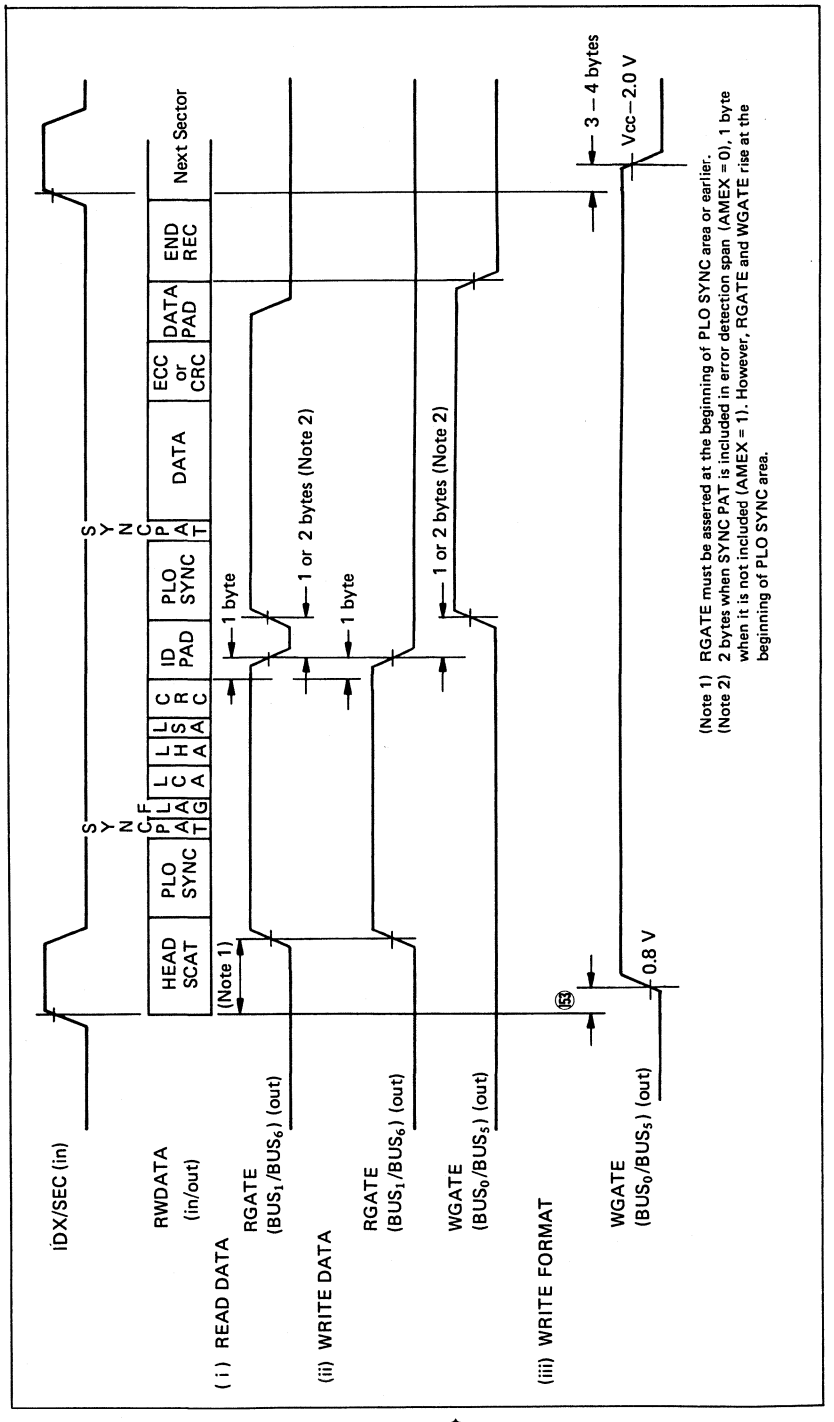
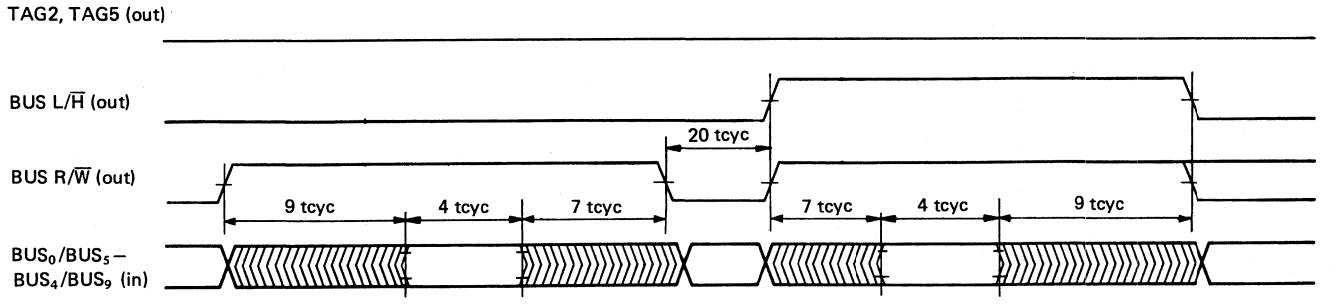


Figure 43 Read/Write Operation (3) (SMD)



BUSL/H̄ = low

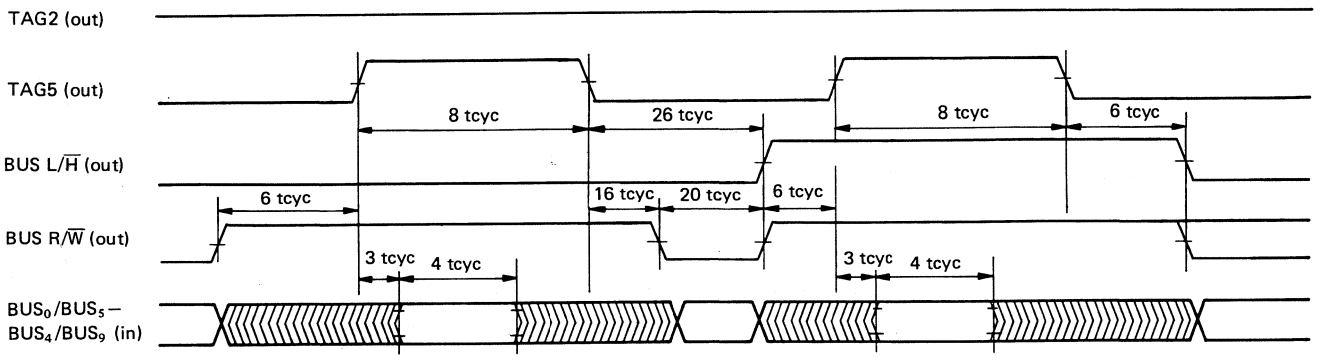
Pin	Input Signal Name	Abbreviation
BUS_4/BUS_9	—	—
BUS_3/BUS_8	—	—
BUS_2/BUS_7	*	*
BUS_1/BUS_6	*	*
BUS_0/BUS_5	*	*

BUSL/H̄ = high

Pin	Input Signal Name	Abbreviation
BUS_4/BUS_9	Write Protected	WPRT
BUS_3/BUS_8	Fault	FLT
BUS_2/BUS_7	Seek Error	SKERR
BUS_1/BUS_6	On Cylinder	OCYL
BUS_0/BUS_5	Unit Ready	URDY

(Note 1) Names of signals which are read from a drive when the HDC is connected to the DK812S (Hitachi).
(Note 2) User Area

Figure 44 Sense Drive Operation (1) (SMD)



BUSL/$\overline{H}$ = low

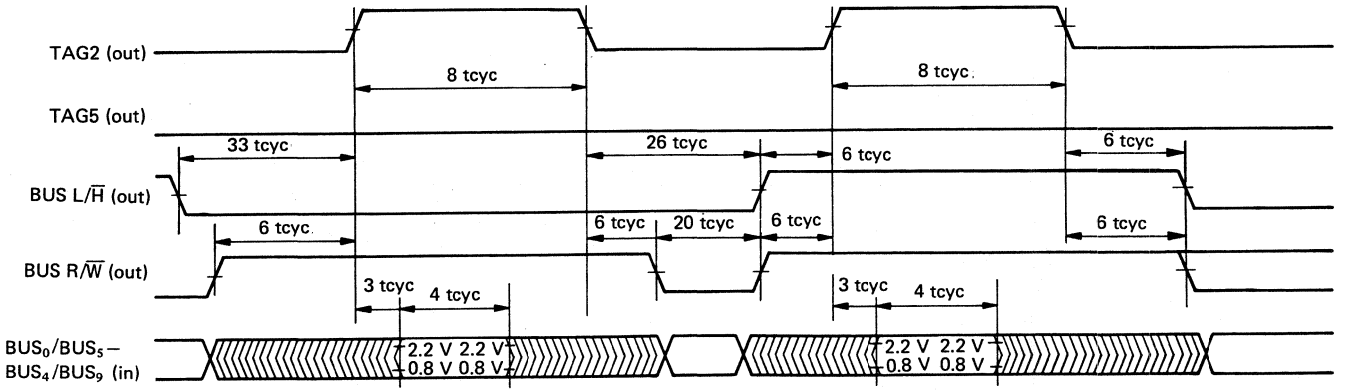
Pin	Input Signal Name (Note 1)	Abbreviation
BUS_4/BUS_9	–	–
BUS_3/BUS_8	–	–
BUS_2/BUS_7	* (Note 2)	*
BUS_1/BUS_6	*	*
BUS_0/BUS_5	SELECT ERROR	SELER

BUSL/$\overline{H}$ = high

Pin	Input Signal Name (Note 1)	Abbreviation
BUS_4/BUS_9	Write Error 3	WERR3
BUS_3/BUS_8	Write Error 2	WERR2
BUS_2/BUS_7	Write Error 1	WERR1
BUS_1/BUS_6	Status Error 2	SERR2
BUS_0/BUS_5	Status Error 1	SERR1

(Note 1) Names of signals which are read from a drive when the HDC is connected to the DK812S (Hitachi).
(Note 2) User Area

Figure 45 Check Drive Status (2) (SMD)



BUSL/$\overline{H}$ = low

Pin	Input Signal Name (Note 1)	Abbreviation
BUS_4/BUS_9	–	–
BUS_3/BUS_8	–	–
BUS_2/BUS_7	* (Note 2)	*
BUS_1/BUS_6	* (Note 2)	*
BUS_0/BUS_5	Sector Address 32	SAD32

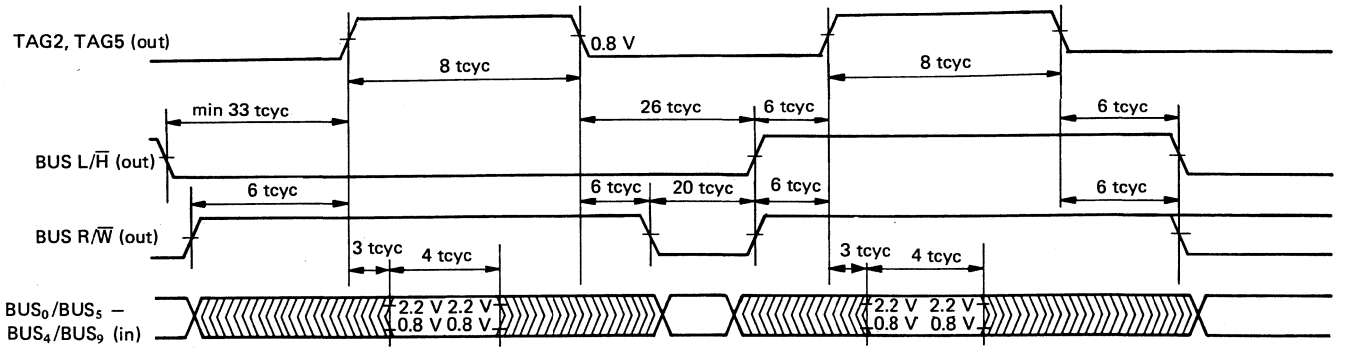
BUSL/$\overline{H}$ = high

Pin	Input Signal Name (Note 1)	Abbreviation
BUS_4/BUS_9	Sector Address 16	SAD16
BUS_3/BUS_8	Sector Address 8	SAD8
BUS_2/BUS_7	Sector Address 4	SAD4
BUS_1/BUS_6	Sector Address 2	SAD2
BUS_0/BUS_5	Sector Address 1	SAD1

(Note 1) Names of signals which are read from a drive when the HDC is connected to the DK812S (Hitachi).

(Note 2) User area

Figure 46 Check Drive Status (3) (SMD)



BUSL/H̄ = low

Pin	Input Signal Name (Note 1)	Abbreviation
BUS_4/BUS_9	–	–
BUS_3/BUS_8	–	–
BUS_2/BUS_7	* (Note 2)	*
BUS_1/BUS_6	* (Note 2)	*
BUS_0/BUS_5	Device Type 32	DTP32

BUSL/H̄ = high

Pin	Input Signal Name (Note 1)	Abbreviation
BUS_4/BUS_9	Device Type 16	DTP16
BUS_3/BUS_8	Device Type 8	DTP8
BUS_2/BUS_7	Device Type 4	DTP4
BUS_1/BUS_6	Device Type 2	DTP2
BUS_0/BUS_5	Device Type 1	DTP1

(Note 1) Names of signals which are read from a drive when the HDC is connected to the DK812S (Hitachi).

(Note 2) User Area

Figure 47 Check Drive Status (4) (SMD)

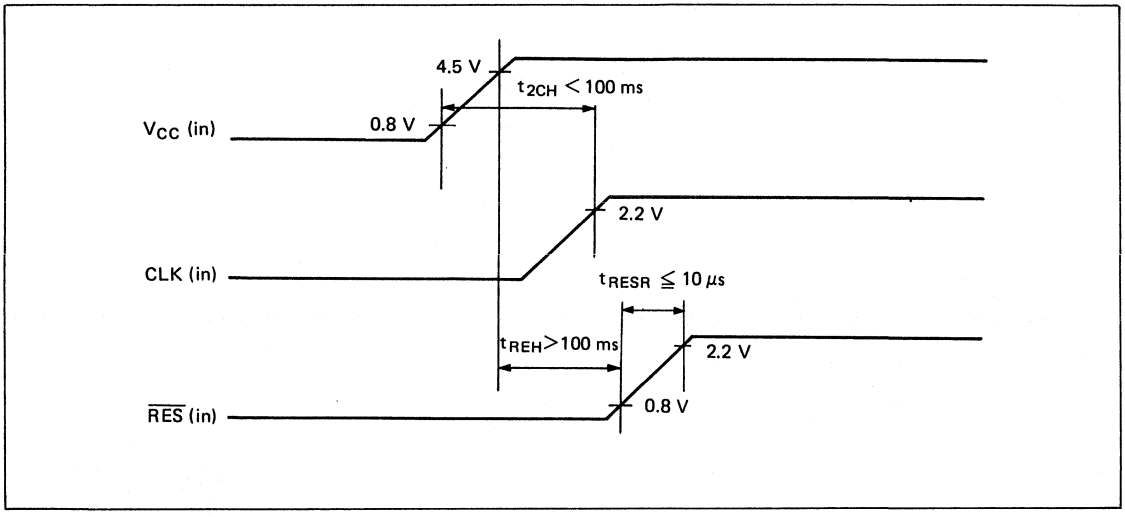


Figure 48 Timing Requirement during Power-On.

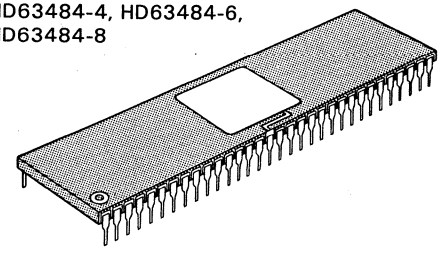
HD63484
ACRTC (Advanced CRT Controller)

The Advanced CRT Controller (ACRTC) is a CMOS VLSI microcomputer peripheral device capable of controlling raster-scan CRTs to display both graphics and characters. The ACRTC is a new generation CRT controller that is based on a bit-mapped technology. It executes high-level drawing commands, like Line, Ellipse, Paint, Pattern, and Copy, issued by the MPU in X-Y coordinates, and performs address translation to draw into frame memory. It can draw in up to 64K colors, on 3 split screens and an independent window, and perform area clipping and hitting. It has more display control functions than an HD6845S (CRTC).

The ACRTC controls a CRTC in one of three modes; character only, graphic only and multiplexed character/graphic modes. Therefore, the ACRTC has many applications, from character only display devices to large full-graphic systems.

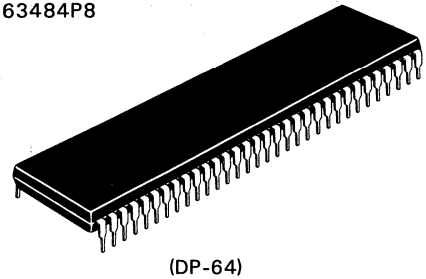
The ACRTC can reduce CPU software overhead and enhance system throughput.

HD63484-4, HD63484-6, HD63484-8



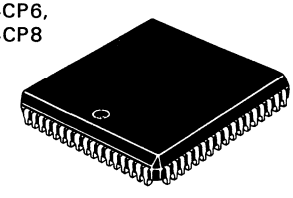
(DC-64)

HD63484P4, HD63484P6, HD63484P8



(DP-64)

HD63484CP4, HD63484CP6, HD63484CP8

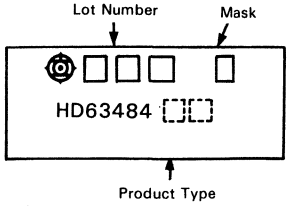


(CP-68) — Under development —

■ FEATURES

- High-speed graphics
  - Drawing rate: Maximum 500 ns/pixel (Color drawing)
  - Commands: Dot, Line, Rectangle, Poly-line, Polygon, Circle, Ellipse, Paint, Copy, etc.
  - Colors: 16 bits/word
    1, 2, 4, 8, 16 bits/pixel (5 types) monochrome to max. 64k colors.
  - Pattern RAM (32 bytes)
  - Conversion of logical X-Y coordinates into physical address.
  - Color operation and conditional drawing
  - Drawing area control for hardware clipping and hitting
- Large frame-memory space
  - Maximum 2M bytes graphic memory and 128k bytes character memory separated from the MPU memory
  - Maximum, resolution 4096 x 4096 (1 bit/pixel mode)
- CRT display controls
  - Split screens (3 displays and 1 window)
  - Zooming up (1 to 16 times)
  - Scroll (Vertical and horizontal)
- Interleaved Access mode for flashless display and superimposition
- External synchronization between ACRTCs or between the ACRTC and external device (ex. TV system or other controller)
- DMA interface
- Two programmable cursors
- Three scan modes
  - Non-interlaced
  - Interlace sync.
  - Interlace sync. and video
- Interrupt request to MPU
- 256 characters/line, 32 rasters/line, 4096 rasters/screen
- Maximum clock frequency 8 MHz
- CMOS, +5V single power supply

■ PRODUCTS MARK



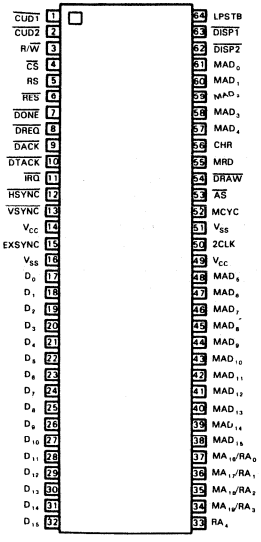
■ TYPE OF PRODUCTS

Product type	Clock frequency (2CLK)	Package	Available mask
HD63484-4 HD63484-6 HD63484-8	4 MHz 6 MHz 8 MHz	DC-64 64 pin ceramic DIP	R mask, S mask
HD63484P4 HD63484P6 HD63484P8	4 MHz 6 MHz 8 MHz	DP-64 64 pin plastic DIP	S mask
HD63484CP4* HD63484CP6* HD63484CP8*	4 MHz 6 MHz 8 MHz	CP-68 68 pin PLCC	S mask

* indicates the product is under development

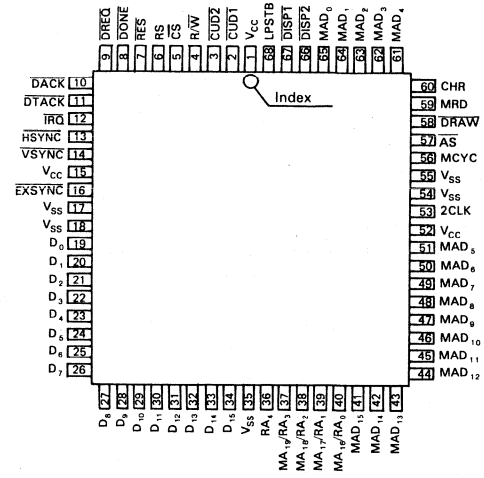
■ PIN ARRANGEMENT (Top View)

- **HD63484, HD63484P**



(Top View)

- **HD63484CP**



(Top View)

■ ABSOLUTE MAXIMUM RATINGS

Item	Symbol	Rating	Unit
Supply Voltage	V_{CC}*	−0.3 to +7.0	V
Input Voltage	V_{in}*	−0.3 to V_{CC}+0.3	V
Allowable Output Current	$\|I_O\|$**	5	mA
Total Allowable Output Current	$\|\Sigma I_O\|$***	120	mA
Operating Temperature	T_{opr}	0 to +70	°C
Storage Temperature	T_{stg}	−55 to +150	°C

* This value is in reference to V_{SS} = 0V.

** The allowable output current is the maximum current that may be drawn from, or flow out to, one output terminal or one input/output common terminal.

*** The total allowable output current is the total sum of currents that may be drawn from, or flow out to, output terminals or input/output common terminals.

(Note) Using an LSI beyond its maximum ratings may result in its permanent destruction. LSI's should usually be used under recommended operating conditions. Exceeding any of these conditions may adversely affect its reliability.

■ RECOMMENDED OPERATING CONDITIONS

Item	Symbol	Min	Typ	Max	Unit
Supply Voltage	V_{CC}*	4.75	5.0	5.25	V
Input "Low" Level Voltage	V_{IL}*	0	–	0.7	V
Input "High" Level Voltage	V_{IH}*	2.2	–	V_{CC}	V
Operating Temperature	T_{opr}	0	25	70	°C

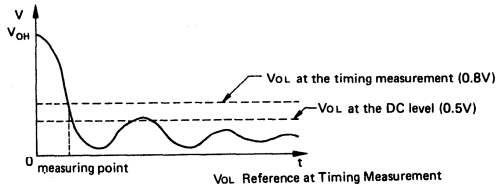
* This value is in reference to V_{SS} = 0V.

■ TIMING MEASUREMENT

The timing measurement point for the output "low" level is defined at 0.8V throughout this specification.

The output "low" level at stable condition (DC characteristics) is defined at 0.5V.

The output "high" level is defined at V_{CC}−2.0V.



■ **ELECTRICAL CHARACTERISTICS**

• **DC CHARACTERISTICS (V_{CC} = 5.0V ± 5%, V_{SS} = 0V, Ta = 0 to +70°C unless otherwise noted)**

Item		Symbol	Measuring Condition	4 MHz Version		6 MHz Version		8 MHz Version		Unit
				Min	Max	Min	Max	Min	Max	
Input "High" Level Voltage	All Inputs	V_{IH}		2.2	V_{CC}	2.2	V_{CC}	2.2	V_{CC}	V
Input "Low" Level Voltage	All Inputs	V_{IL}		−0.3	0.7	−0.3	0.7	−0.3	0.7	V
Input Leak Current	R/$\overline{W}$, $\overline{CS}$, RS, $\overline{RES}$, $\overline{DACK}$, 2CLK, LPSTB	I_{in}	V_{in}=0 to V_{CC}	−2.5	2.5	−2.5	2.5	−2.5	2.5	μA
Three State (Off State) Input Current	D_0−D_{15}, $\overline{EXSYNC}$, MAD_0−MAD_{15}	I_{TSI}	V_{in}=0.4 to V_{CC}	−10	10	−10	10	−10	10	μA
Output "High" Level Voltage	D_0−D_{15}, MAD_0−MAD_{15}, $\overline{CUD1}$, $\overline{CUD2}$, $\overline{DREQ}$, $\overline{DTACK}$, $\overline{HSYNC}$, $\overline{VSYNC}$, $\overline{EXSYNC}$,	V_{OH}	I_{OH}=−400 μA	V_{CC} −1.0	−	V_{CC} −1.0	−	V_{CC} −1.0	−	V
Output "Low" Level Voltage	$\overline{DISP1}$, $\overline{DISP2}$, CHR, MRD, $\overline{DRAW}$, $\overline{AS}$, MCYC, RA_4, MA_{16}/RA_0−MA_{19}/RA_3	V_{OL}	I_{OL}=2.2 mA	−	0.5	−	0.5	−	0.5	V
	$\overline{IRQ}$, $\overline{DONE}$	V_{OL}	I_{OL}=2.5 mA	−	0.5	−	0.5	−	0.5	V
Output Leak Current (Off State)	$\overline{IRQ}$, $\overline{DONE}$	I_{LOD}	V_{OH}=V_{CC}	−	10	−	10	−	10	μA
Input Capacity	D_0−D_{15}, $\overline{EXSYNC}$, MAD_0−MAD_{15}	C_{in}	V_{in}=0 V, Ta=25 °C, f=1.0 MHz	−	17	−	17	−	17	pF
	R/$\overline{W}$, $\overline{CS}$, $\overline{RS}$, $\overline{RES}$, $\overline{DACK}$, 2CLK, LPSTB		V_{in}=0 V, Ta=25 °C, f=1.0 MHz	−	17	−	17	−	17	
Output Capacity	$\overline{IRQ}$, $\overline{DONE}$	C_{out}	V_{in}=0V, Ta=25°C, f=1.0MHz	−	15	−	15	−	15	pF
Current Consumption		I_{CC}	• Chip not selected • Display in progress	−	60	−	80	−	100	mA
			• Data bus in read/write operation • Display in progress • Command execution in progress	−	60	−	80	−	100	

- **AC CHARACTERISTICS (V_{CC} = 5.0V ± 5%, V_{SS} = 0V, Ta = 0 to +70°C unless otherwise noted)**

Clock

No.	Item	Symbol	Reference figure	4 MHz Version		6 MHz Version		8 MHz Version		Unit
				Min	Max	Min	Max	Min	Max	
	Operatin Frequency of 2CLK	f		1	4	1	6	1	8	MHz
1	Clock Cycle Time	t_{cyc}	Fig. 1	250	1000	167	1000	125	1000	ns
2	Clock "High" Level Pulse Width	t_{PWCH}		115	500	75	500	55	500	ns
3	Clock "Low" Level Pulse Width	t_{PWCL}		115	500	75	500	55	500	ns
4	Clock Rise Time	t_{Cr}		–	10	–	10	–	10	ns
5	Clock Fall Time	t_{Cf}		–	10	–	10	–	10	ns

MPU Read/Write Cycle

No.	Item	Symbol	Reference figure	4 MHz Version		6 MHz Version		8 MHz Version		Unit
				Min	Max	Min	Max	Min	Max	
6	R/$\overline{W}$ Setup Time	t_{RWS}	Fig. 2– Fig. 4	70	–	60	–	50	–	ns
7	R/$\overline{W}$ Hold Time	t_{RWH}		0	–	0	–	0	–	ns
8	RS Setup Time	t_{RSS}		70	–	60	–	50	–	ns
9	RS Hold Time	t_{RSH}		0	–	0	–	0	–	ns
10	$\overline{CS}$ Setup Time	t_{CSS}		50	–	40	–	40	–	ns
11	$\overline{CS}$ "High" Level Width	t_{WCSH}	Figs. 2&3	80	–	70	–	60	–	ns
12										
13	Read Wait Time	t_{RWAI}	Fig. 2, Fig. 4	0	–	0	–	0	–	ns
14	Read Data Access Time	t_{RDAC}		–	120	–	100	–	80	ns
15	Read Data Hold Time	t_{RDH}		10	–	10	–	10	–	ns
16	Read Data Turn Off Time	t_{RDZ}		–	60	–	60	–	60	ns
17	$\overline{DTACK}$ Delay Time (Z to L)	t_{DTKZL}	Fig. 2– Fig. 4	–	90	–	80	–	70	ns
18	$\overline{DTACK}$ Hold Time (D to L)	t_{DTKDL}	Figs. 2&4	0	–	0	–	0	–	ns
19	$\overline{DTACK}$ Release Time (L to H)	t_{DTKLH}	Fig. 2– Fig.4	–	100	–	90	–	80	ns
20	$\overline{DTACK}$ Turn Off Time (H to Z)	t_{DTKZ}		–	100	–	100	–	100	ns
21	Data Bus 3 State Recovery Time 1	t_{DBRT1}	Fig. 2, Fig. 4	0	–	0	–	0	–	ns
22	Write Wait Time	t_{WWAI}	Fig. 3, Fig. 4	0	–	0	–	0	–	ns
23	Write Data Setup Time	t_{WDS}		80	–	60	–	40	–	ns
24	Write Data Hold Time	t_{WDH}		10	–	10	–	10	–	ns

DMA Read/Write Cycle

No.	Item	Symbol	Reference figure	4 MHz Version		6 MHz Version		8 MHz Version		Unit
				Min	Max	Min	Max	Min	Max	
25	$\overline{\text{DREQ}}$ Delay Time1	t_{DRQD1}	Fig. 5 – Fig. 8	–	150	–	130	–	110	ns
26	$\overline{\text{DREQ}}$ Delay Time2	t_{DRQD2}		–	90	–	80	–	70	ns
27	DMA R/$\overline{\text{W}}$ Setup Time	t_{DRWS}		70	–	60	–	50	–	ns
28	DMA R/$\overline{\text{W}}$ Hold Time	t_{DRWH}		0	–	0	–	0	–	ns
29	$\overline{\text{DACK}}$ Setup Time	t_{DAKS}		50	–	40	–	40	–	ns
30	$\overline{\text{DACK}}$ "High" Level Width	t_{DAKH}		80	–	70	–	60	–	ns
31										
32	DMA Read Wait Time	t_{DRW}	Fig. 5, Fig. 6	0	–	0	–	0	–	ns
33	DMA Read Data Access Time	t_{DRDAC}		–	120	–	100	–	80	ns
34	DMA Read Data Hold Time	t_{DRDH}		10	–	10	–	10	–	ns
35	DMA Read Data Turn Off Time	t_{DRDZ}		–	60	–	60	–	60	ns
36	DMA $\overline{\text{DTACK}}$ Delay Time (Z to L)	t_{DDTZL}	Fig. 5 – Fig. 8	–	90	–	80	–	70	ns
37	DMA $\overline{\text{DTACK}}$ Delay Time (D to L)	t_{DDTDL}	Fig. 5, Fig. 6	0	–	0	–	0	–	ns
38	DMA $\overline{\text{DTACK}}$ Release Time (L to H)	t_{DDTLH}	Fig. 5 – Fig. 8	–	100	–	90	–	80	ns
39	DMA $\overline{\text{DTACK}}$ Turn Off Time (H to Z)	t_{DDTHZ}		–	100	–	100	–	100	ns
40	$\overline{\text{DONE}}$ Output Delay Time	t_{DND}		–	90	–	80	–	70	ns
41	$\overline{\text{DONE}}$ Output Turn Off Time (L to Z)	t_{DNLZ}		–	100	–	90	–	80	ns
42	Data Bus 3 State Recovery Time 2	t_{DBRT2}	Fig. 5, Fig. 6	0	–	0	–	0	–	ns
43	$\overline{\text{DONE}}$ Input Pulse Width	t_{DNPW}	Fig. 5 – Fig. 8	2	–	2	–	2	–	t_{cyc}
44	DMA Write Wait Time	t_{DWW}	Fig. 7, Fig. 8	0	–	0	–	0	–	ns
45	DMA Write Data Setup Time	t_{DWDS}		80	–	60	–	40	–	ns
46	DMA Write Data Hold Time	t_{DWDH}		10	–	10	–	10	–	ns
47										

Frame Memory Read/Write Cycle

No.	Item	Symbol	Reference figure	4 MHz Version		6 MHz Version		8 MHz Version		Unit
				Min	Max	Min	Max	Min	Max	
48	$\overline{AS}$ "Low" Level Pulse Width	t_{PWASL}	Fig. 9 – Fig. 12	80	–	40	–	25	–	ns
49	Memory Address Hold Time 2	t_{MAH2}		10	–	10	–	10	–	ns
50	$\overline{AS}$ Delay Time 1	t_{ASD1}		–	90	–	75	–	60	ns
51	AS Delay Time 2	t_{ASD2}		5	90	5	75	5	65	ns
52	Memory Address Delay Time	t_{MAD}		15	95	15	80	15	70	ns
53	Memory Address Hold Time 1	t_{MAH1}		10	–	10	–	10	–	ns
54	Memory Address Turn Off Time (A to Z)	t_{MAAZ}	Figs. 9, 10, and 12	–	50	–	50	–	50	ns
55	Memory Address Data Setup Time	t_{MRDS}	Fig. 10	60	–	50	–	40	–	ns
56	Memory Read Data Hold Time	t_{MRDH}		10	–	10	–	10	–	ns
57	MA/RA Delay Time	t_{MARDE}	Fig. 6 – Fig. 12	–	100	–	90	–	80	ns
58	MA/RA Hold Time	t_{MARAH}	Fig. 9 – Fig. 11	10	–	10	–	10	–	ns
59	MCYC Delay Time	t_{MCYCD}	Fig. 9 – Fig. 13	–	60	–	50	–	50	ns
60	MRD Delay Time	t_{MRDD}	Fig. 9 – Fig. 12	–	90	–	80	–	70	ns
61	MRD Hold Time	t_{MRDH}		10	–	10	–	10	–	ns
62	$\overline{DRAW}$ Delay Time	t_{DRWD}		–	90	–	80	–	70	ns
63	$\overline{DRAW}$ Hold Time	t_{DRWH}		10	–	10	–	10	–	ns
64	Memory Write Data Delay Time	t_{MWDD}	Fig. 11	–	90	–	80	–	70	ns
65	Memory Write Data Hold Time	t_{MWDH}		10	–	10	–	10	–	ns
110	Memory Address Setup Time	t_{MAS}	Fig. 9 – Fig. 12	10	–	10	–	10	–	ns

Note 1) Characteristic of No.52 is defined independently of the clock frequency (2CLK) and those of No.51 and No.110.
Note 2) New characteristics of No.50 and No.52 shown above are applicable only to lot numbers 5M*, 6**, 7**, and greaters (*: don't care).
For the other lot numbers, applicable characteristics are as follows:

No.	Symbol	4 MHz Version		6 MHz Version		8 MHz Version		Unit
		Min	Max	Min	Max	Min	Max	
50	t_{ASD1}	–	90	–	75	–	65	ns
52	t_{MAD}	–	95	–	80	–	70	ns

Display Control Signal Output Timing

No.	Item	Symbol	Reference figure	4 MHz Version		6 MHz Version		8 MHz Version		Unit
				Min	Max	Min	Max	Min	Max	
67	$\overline{HSYNC}$ Delay Time	t_{HSD}	Fig. 12 – Fig. 14	–	90	–	80	–	70	ns
68	$\overline{VSYNC}$ Delay Time	t_{VSD}	Fig. 13	–	90	–	80	–	70	ns
69	$\overline{DISP1}$, $\overline{DISP2}$ Delay Time	t_{DSPD}		–	90	–	80	–	70	ns
70	$\overline{CUD1}$, $\overline{CUD2}$ Delay Time	t_{CUDD}		–	90	–	80	–	70	ns
71	$\overline{EXSYNC}$ Output Delay Time	t_{EXD}		20	90	20	80	20	70	ns
72	CHR Delay Time	t_{CHD}		–	90	–	80	–	70	ns
73										
74										

EXSYNC Input Timing

No.	Item	Symbol	Reference figure	4 MHz Version		6 MHz Version		8 MHz Version		Unit
				Min	Max	Min	Max	Min	Max	
75	$\overline{\text{EXSYNC}}$ Input Pulse Width	t_{EXSW}	Fig. 14	3	–	3	–	3	–	t_{cyc}
76	$\overline{\text{EXSYNC}}$ Input Setup Time	t_{EXS}		60	–	60	–	50	–	ns
77	$\overline{\text{EXSYNC}}$ Input Hold Time	t_{EXH}		30	–	30	–	30	–	ns

Input LPSTB Timing

No.	Item	Symbol	Reference figure	4 MHz Version		6 MHz Version		8 MHz Version		Unit
				Min	Max	Min	Max	Min	Max	
78	LPSTB Uncertain Time 1	t_{LPD1}	Fig. 15, Fig. 16	70	–	70	–	70	–	ns
79	LPSTB Uncertain Time 2	t_{LPD2}		10	–	10	–	10	–	ns
80	LPSTB Input Hold Time	t_{LPH}		10	–	10	–	10	–	ns
81	LPSTB Input Inhibit Time	t_{LPI}		4	–	4	–	4	–	t_{cyc}

RES Input and DACK Input Timing

No.	Item	Symbol	Reference figure	4 MHz Version		6 MHz Version		8 MHz Version		Unit
				Min	Max	Min	Max	Min	Max	
82	$\overline{\text{DACK}}$ Setup Time for $\overline{\text{RES}}$	t_{DAKSR}	Fig. 17	100	–	100	–	100	–	ns
83	$\overline{\text{DACK}}$ Hold Time for $\overline{\text{RES}}$	t_{DAKHR}		0	–	0	–	0	–	ns
84	$\overline{\text{RES}}$ Input Pulse Width	t_{RES}		10	–	10	–	10	–	t_{cyc}

IRQ Output, Video Attributes Output Cycle Timing

No.	Item	Symbol	Reference figure	4 MHz Version		6 MHz Version		8 MHz Version		Unit
				Min	Max	Min	Max	Min	Max	
85	$\overline{\text{IRQ}}$ Delay Time 1	t_{IRQ1}	Fig. 18, Fig. 19	–	250	–	200	–	150	ns
86	$\overline{\text{IRQ}}$ Delay Time 2	t_{IRQ2}		–	500	–	500	–	500	ns
87	ATR Delay Time 1	t_{ATRD1}	Fig. 12	–	100	–	90	–	80	ns
88	ATR Hold Time 1	t_{ATRH1}		10	–	10	–	10	–	ns
89										
90	ATR Delay Time 2	t_{ATRD2}	Fig. 12	–	100	–	90	–	80	ns
91	ATR Hold Time 2	t_{ATRH2}		10	–	10	–	10	–	ns

MPU Read/Write Cycle Timing (synchronous bus), DMA Read/Write Cycle Timing (synchronous bus)

No.	Item	Symbol	Reference figure	4 MHz Version		6 MHz Version		8 MHz Version		Unit
				Min	Max	Min	Max	Min	Max	
100	$\overline{CS}$ Cycle Time	t_{CSC}	Fig. 2, Fig. 3	4	–	4	–	4	–	t_{cyc}
101	$\overline{CS}$ "Low" Level Width	t_{WCSL}		2	–	2	–	2	–	t_{cyc}
102	$\overline{CS}$ "High" Level Width	t_{WCSH}		2	–	2	–	2	–	t_{cyc}
104	$\overline{DACK}$ Cycle Time	t_{DACKC}	Fig. 6, Fig. 8	4	–	4	–	4	–	t_{cyc}
105	$\overline{DACK}$ "Low" Level Width	t_{WDACKL}		2	–	2	–	2	–	t_{cyc}
106	$\overline{DACK}$ "High" Level Width	t_{WDACKH}		2	–	2	–	2	–	t_{cyc}

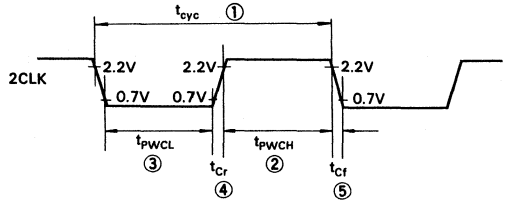
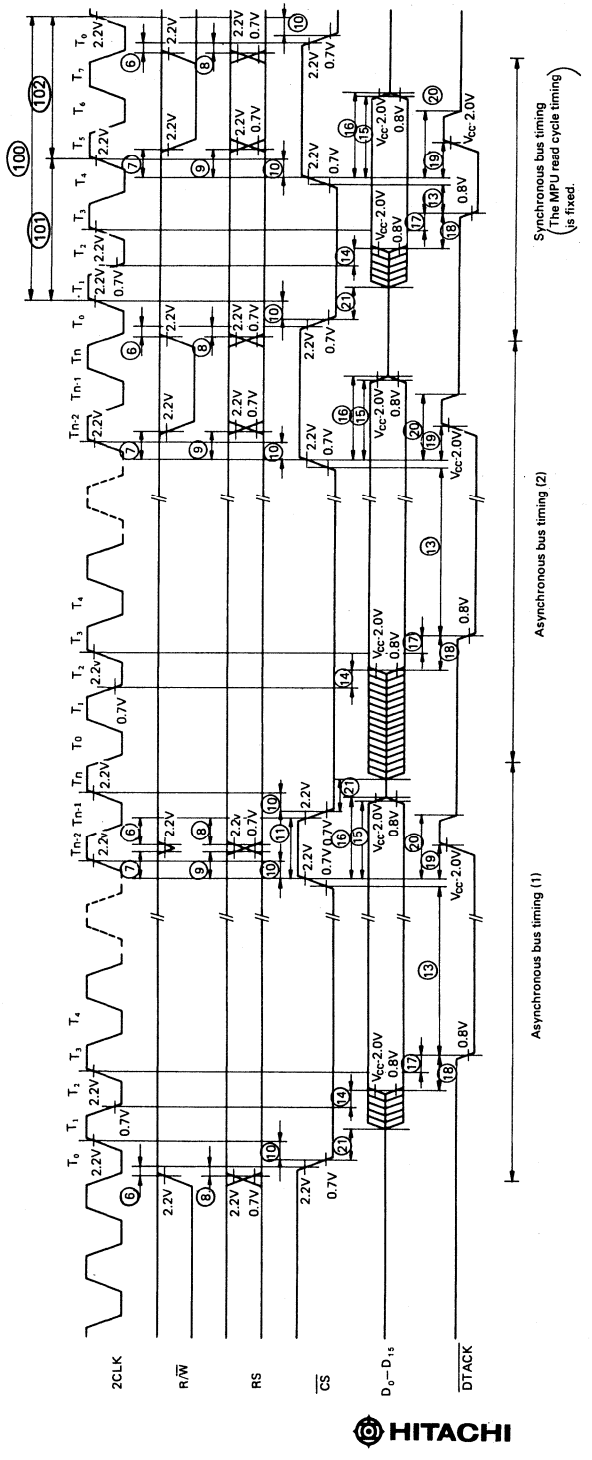


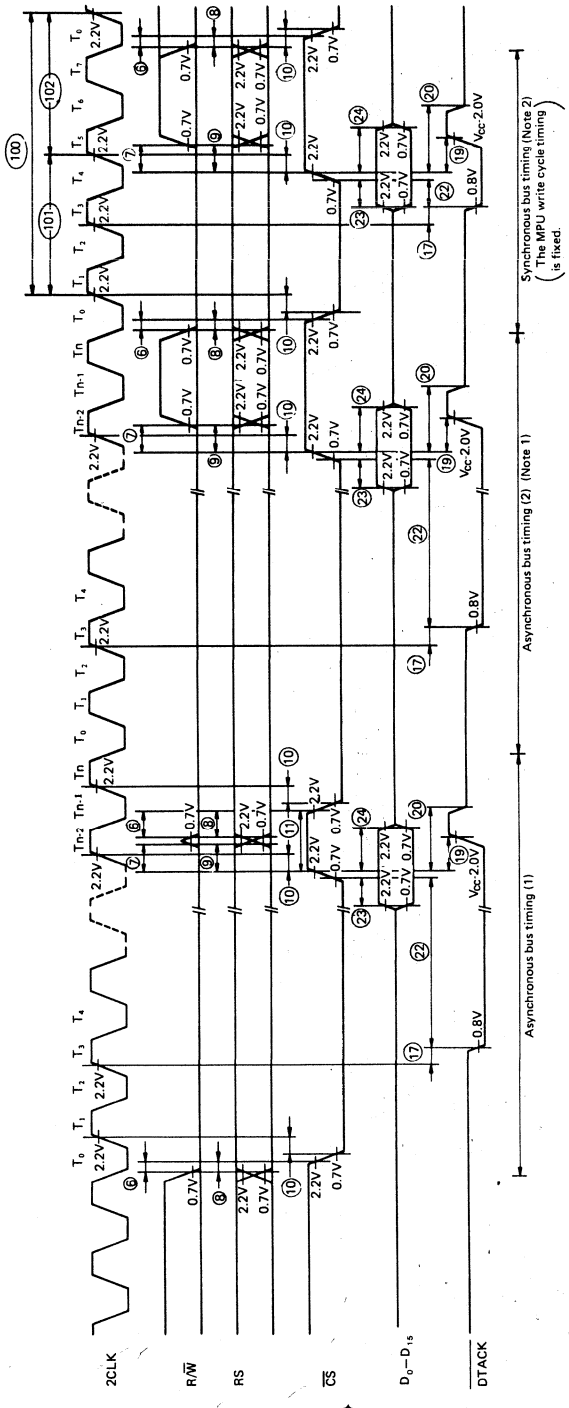
Figure 1 2CLK Waveform



Note 1) $\overline{CS}$ "high" level width must satisfy specification (1).
Unless satisfying the spec (102) DTACK and read data responses to the succeeding cycle are delayed.

Note 2) (100) (101) and (102) are synchronous bus timing specifications.

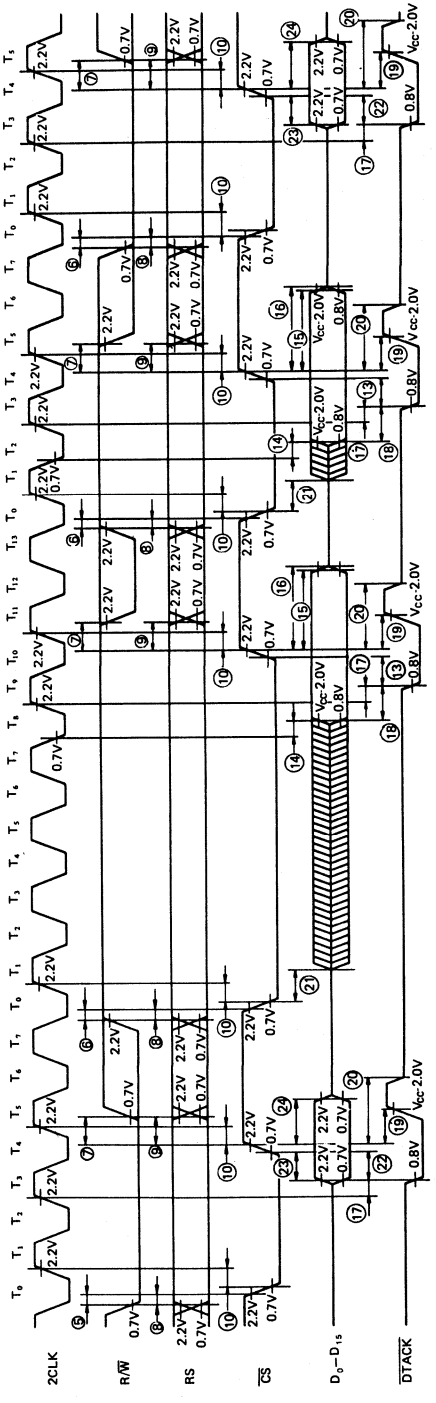
Figure 2 MPU Read Cycle Timing (MPU ← ACRTC)



(Note 1) $\overline{CS}$ "high" width must satisfy the specification (1).
Unless satisfying the spec. (102), DTACK response to the succeeding cycle is delayed.

(Note 2) When the ACRTC is used with the synchronous bus timing, the specifications (100), (101) and (102) must be satisfied.

Figure 3 MPU Write Cycle Timing (MPU → ACRTC)



(Note) When the MPU read cycle immediately follows the MPU write cycle execution, DTACK and the read data responses are delayed (by 3 cycles of 2CLK) even though spec. (102) is satisfied.

Figure 4 MPU Read/Write Cycle Timing (MPU → ACRTC)

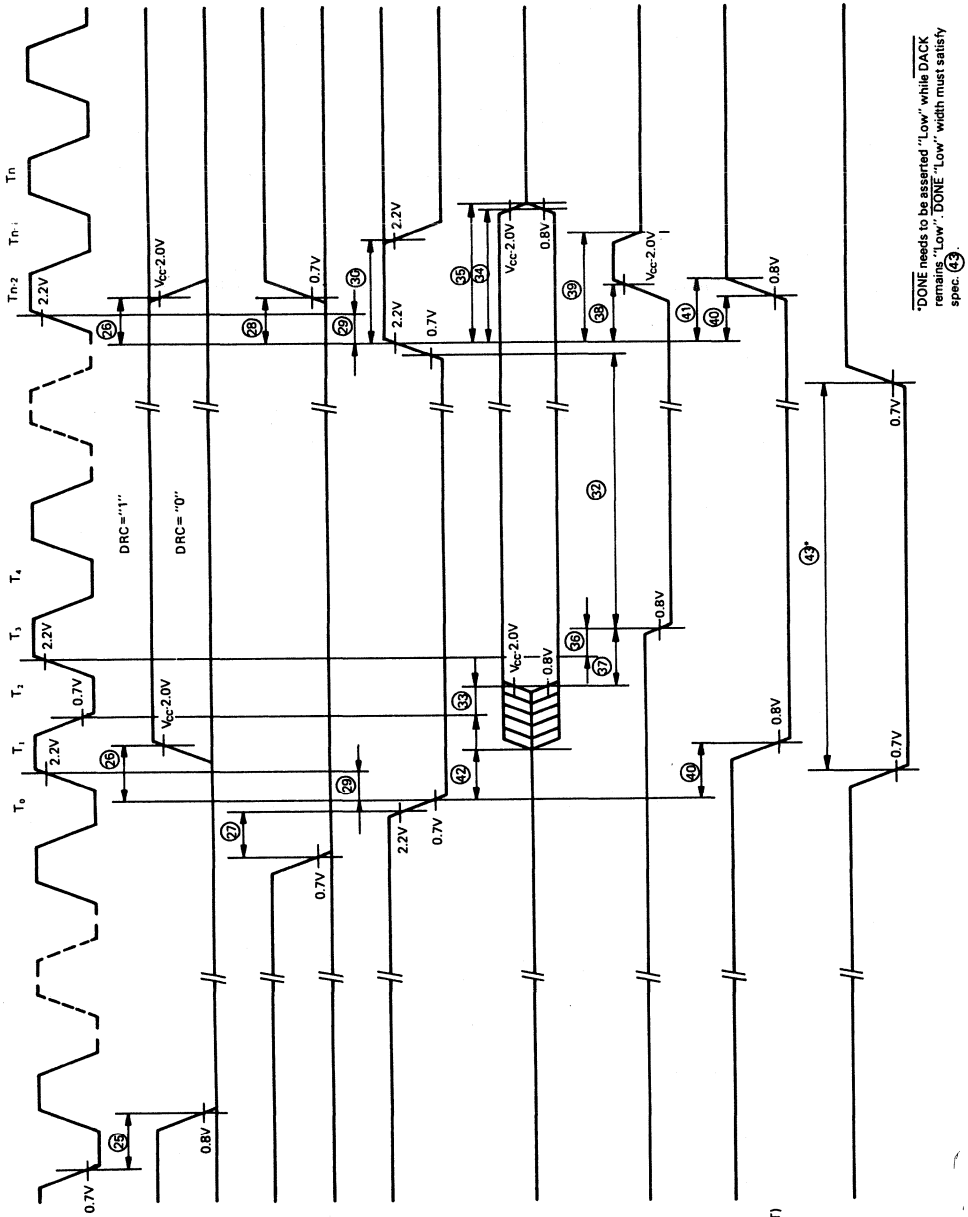
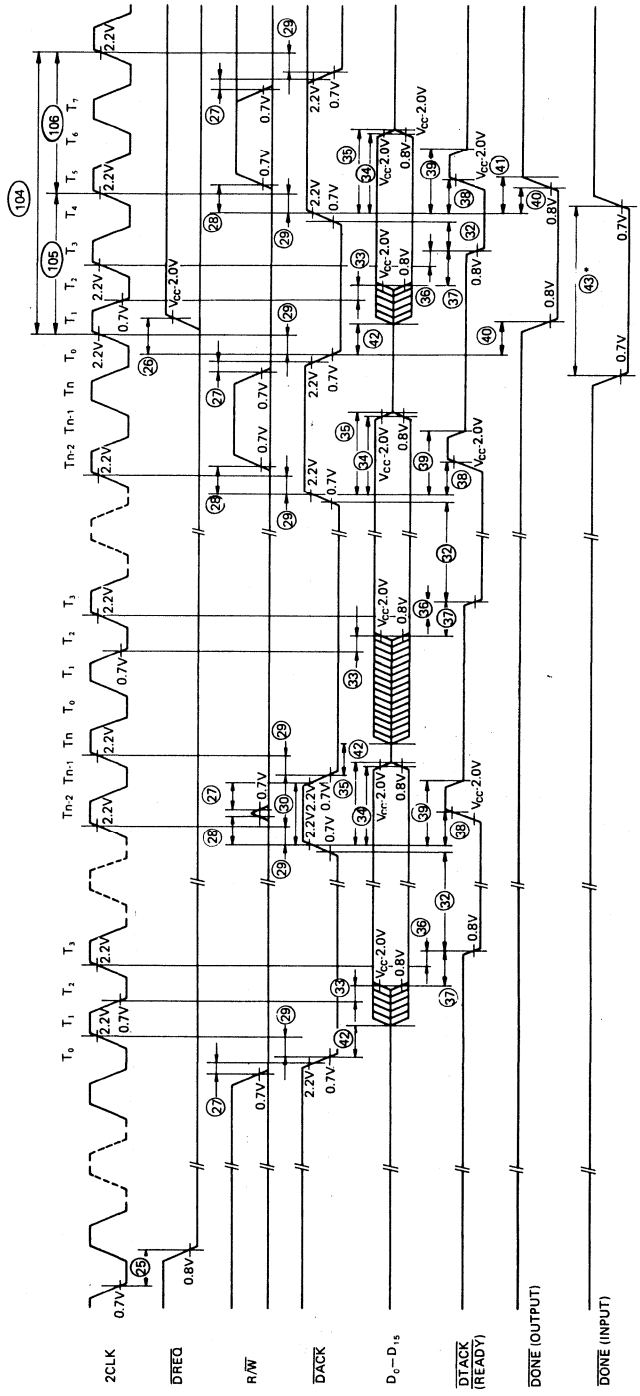


Figure 5 DMA Read Cycle Timing (Memory ← ACRTC)



*DONE needs to be asserted "Low" while DACK remains "Low". DONE "Low" width must satisfy spec. (43).

(Note) DACK "high" width must satisfy spec (39). Unless satisfying the spec. (106), DTACK and the read data responses to the succeeding cycle are delayed. When the ACRTC is used with the synchronous bus timing, the specifications (104), (105) and (106) must be satisfied.

Figure 6 DMA Read Cycle Timing (Memory ← ACRTC): Burst Mode

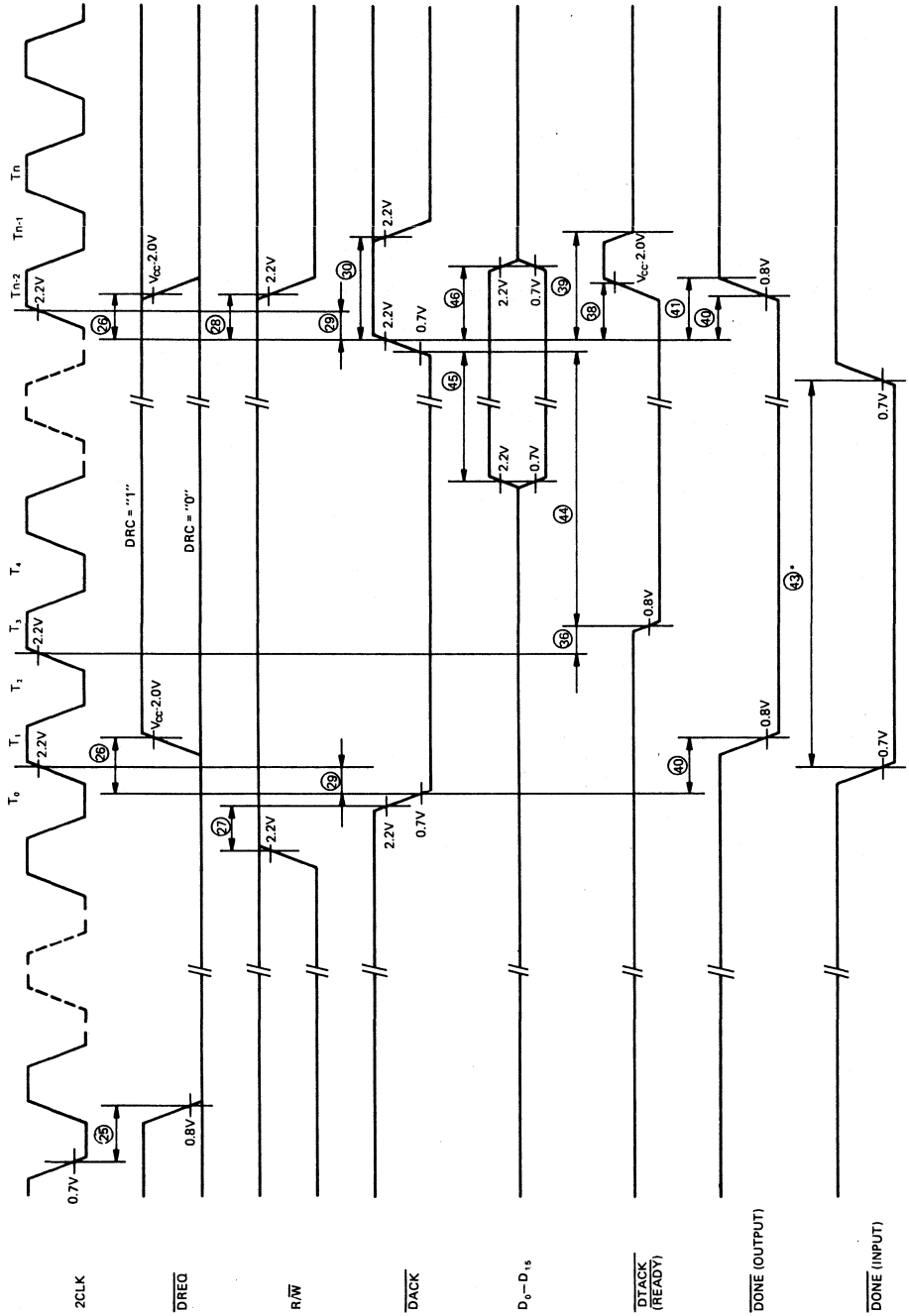
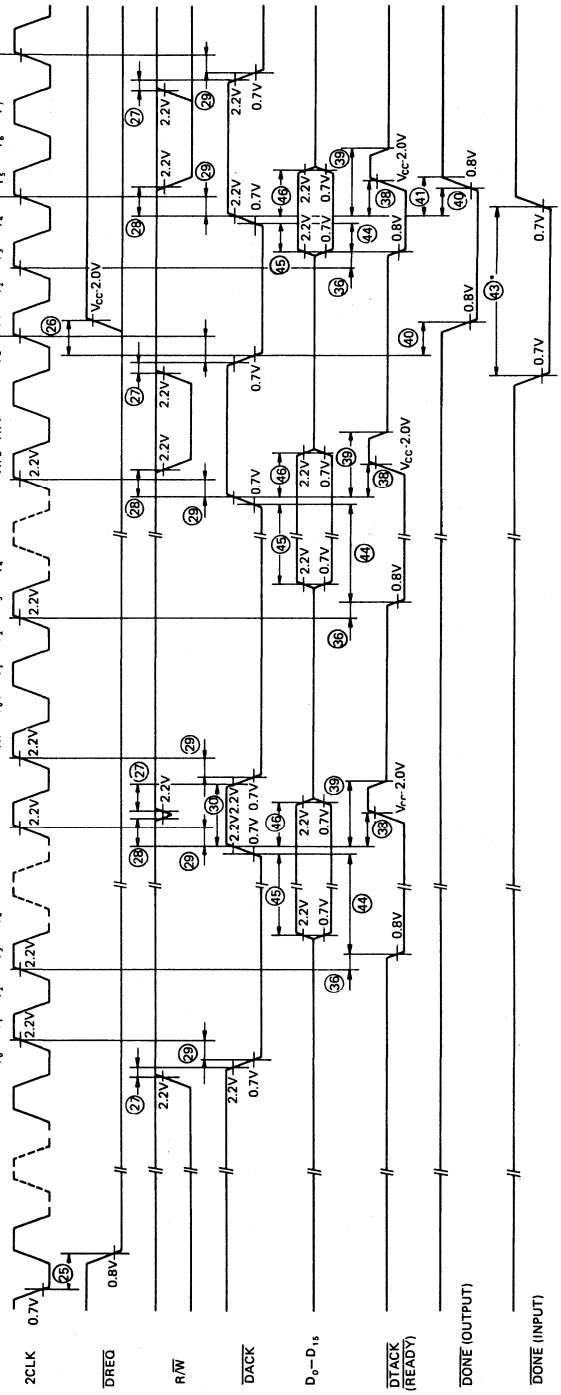


Figure 7 DMA Write Cycle Timing (Memory — ACRTC)



•DONE needs to be asserted "Low" while DACK remains "low".
DONE "low" width must satisfy the spec (43).

(Note) DACK "high" width must satisfy the spec. (30). unless satisfying the spec. (106).
DTACK response to the succeeding cycle is delayed.
When the ACRTC is used with the synchronous bus timing, the specifications (104), (105), and (106) must be satisfied.

Figure 8 DMA Write Cycle Timing (Memory → ACRTC): Burst Mode

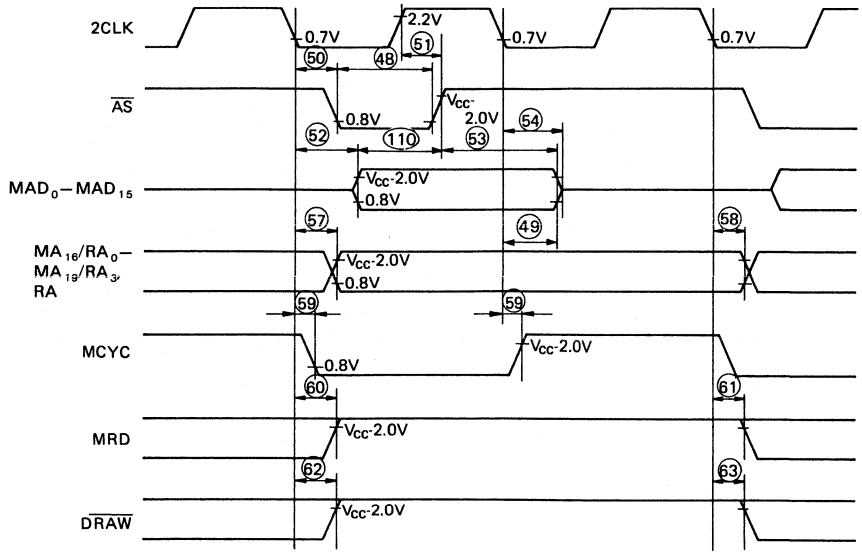


Figure 9 Display Cycle Timing

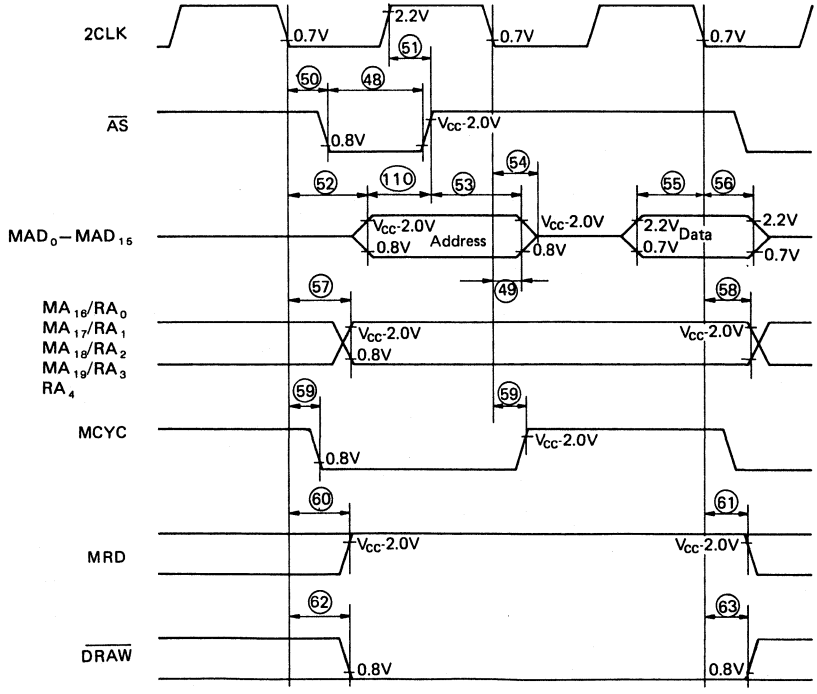


Figure 10 Frame Memory Read Cycle Timing
(ACRTC ← Frame Memory)

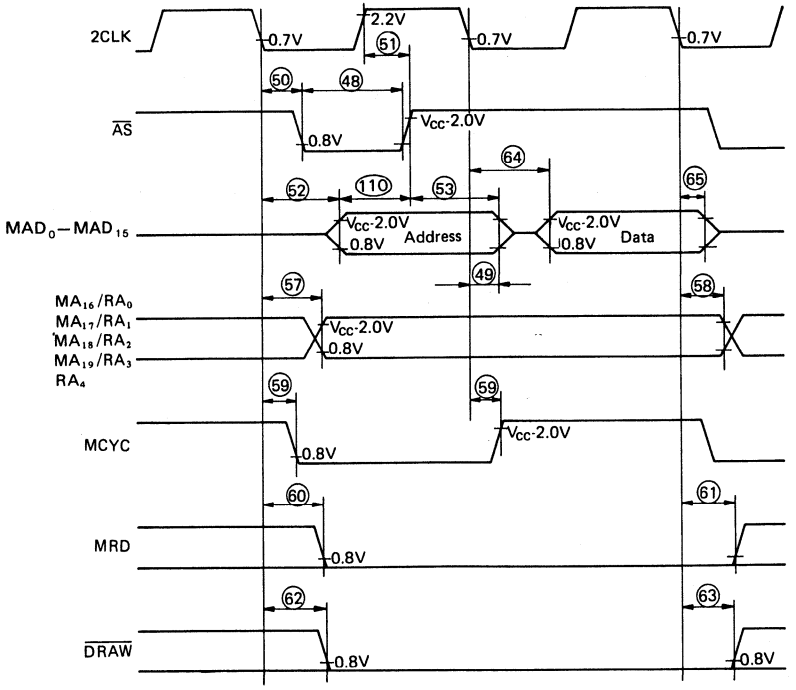


Figure 11 Frame Memory Write Cycle Timing
(ACRTC → Frame Memory)

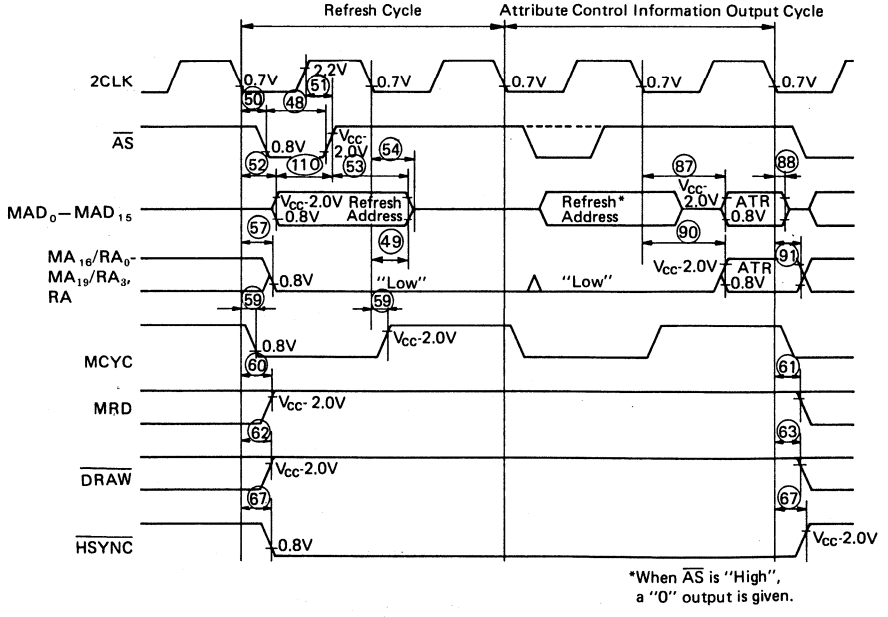


Figure 12 Frame Memory Refresh/Video Attributes Output Cycle Timing

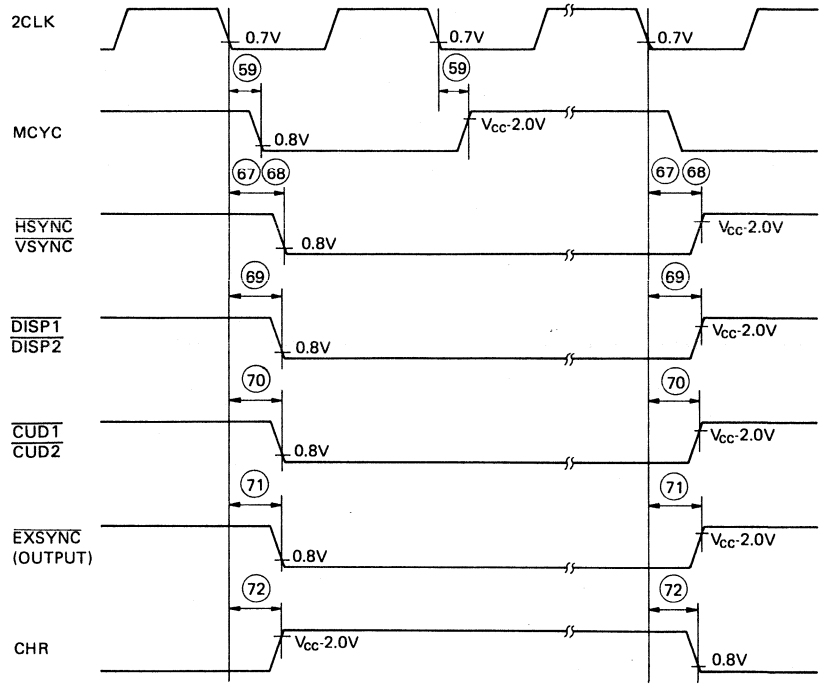
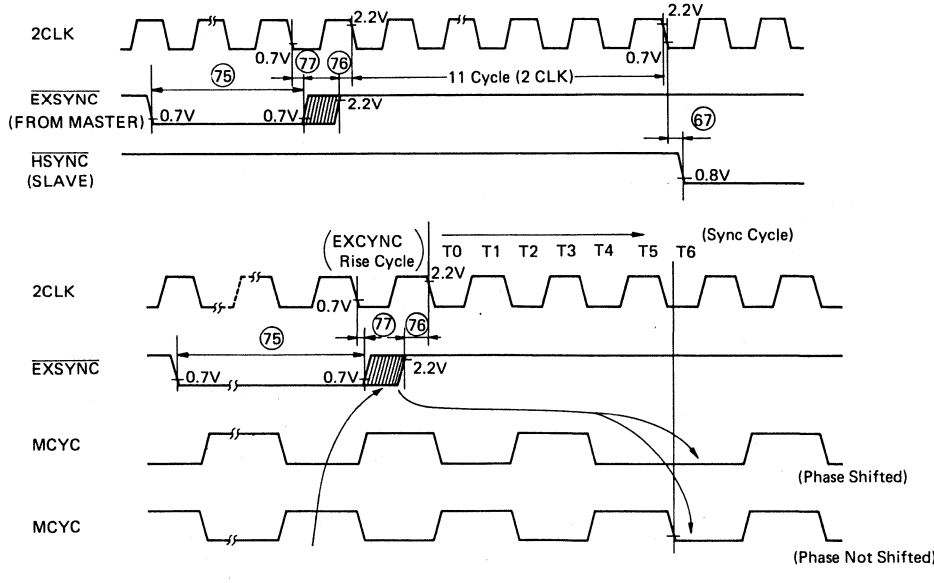


Figure 13 Display Control Signal Output Timing



(When the leading edge of $\overline{\text{EXSYNC}}$ enters this period, ACRTC shifts the internal phase as shown.)

Figure 14 $\overline{\text{EXSYNC}}$ Input Timing

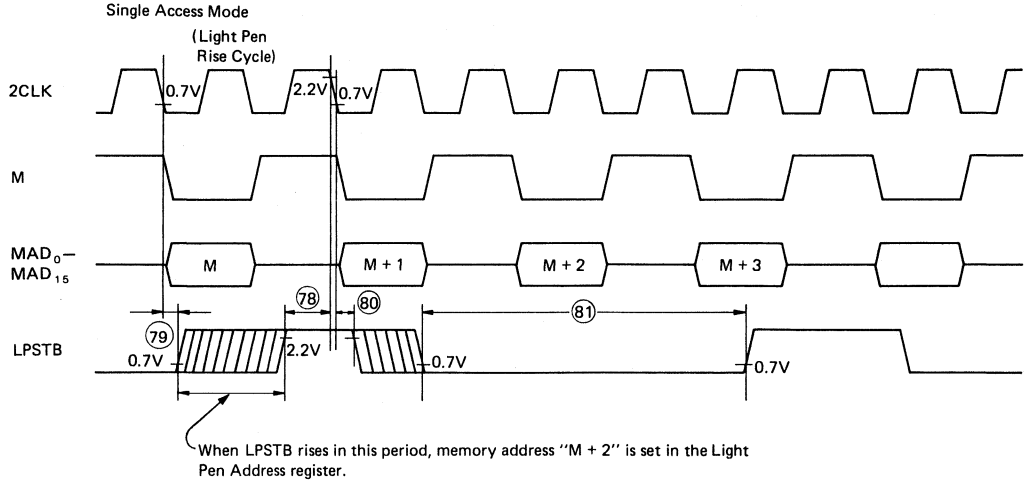
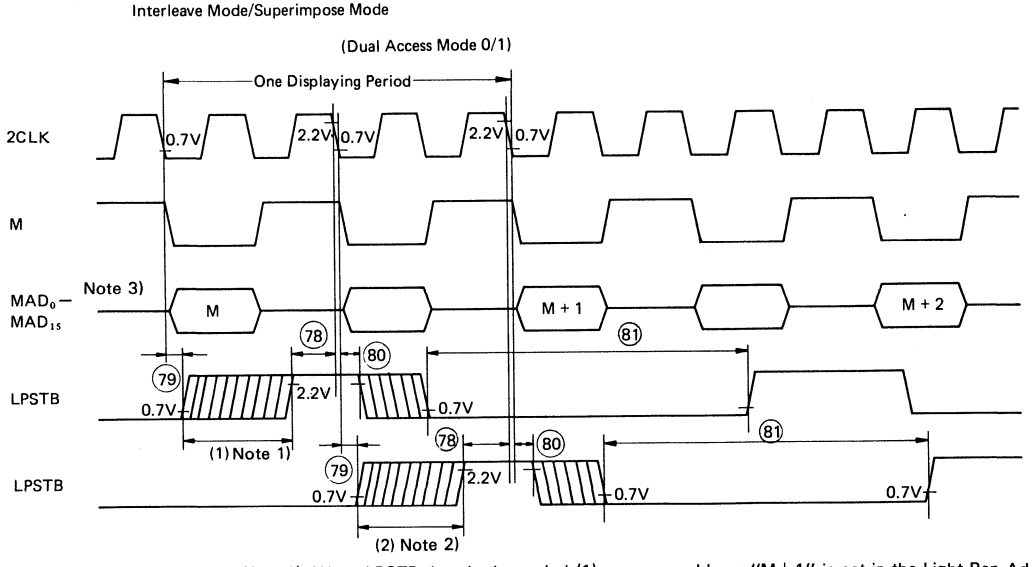


Figure 15 LPSTB Input Timing (Single Access Mode)



Note 1) When LPSTB rises in the period (1), memory address "M+1" is set in the Light Pen Address register.

Note 2) When LPSTB rises in the period (2), memory address "M+2" is set in the Light Pen Address register.

Note 3) In the Interleave Mode, memory address "M", "M+1", "M+2" denote the display address. In the Superimpose Mode, memory address "M", "M+1", "M+2" denote the display address of the background screen.

Figure 16 LPSTB Input Timing (Dual Access Mode)

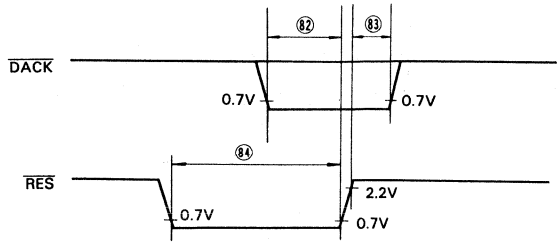


Figure 17 $\overline{RES}$ Input and $\overline{DACK}$ Input Timing
(System Reset and 16-bit/8-bit Selection)

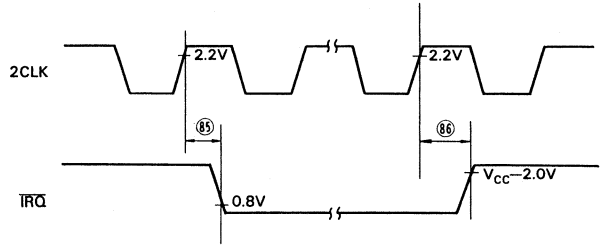
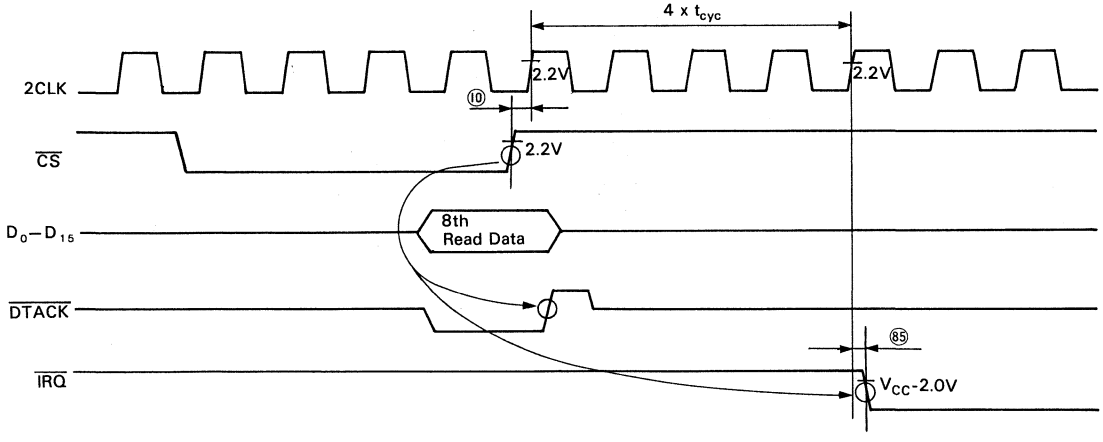
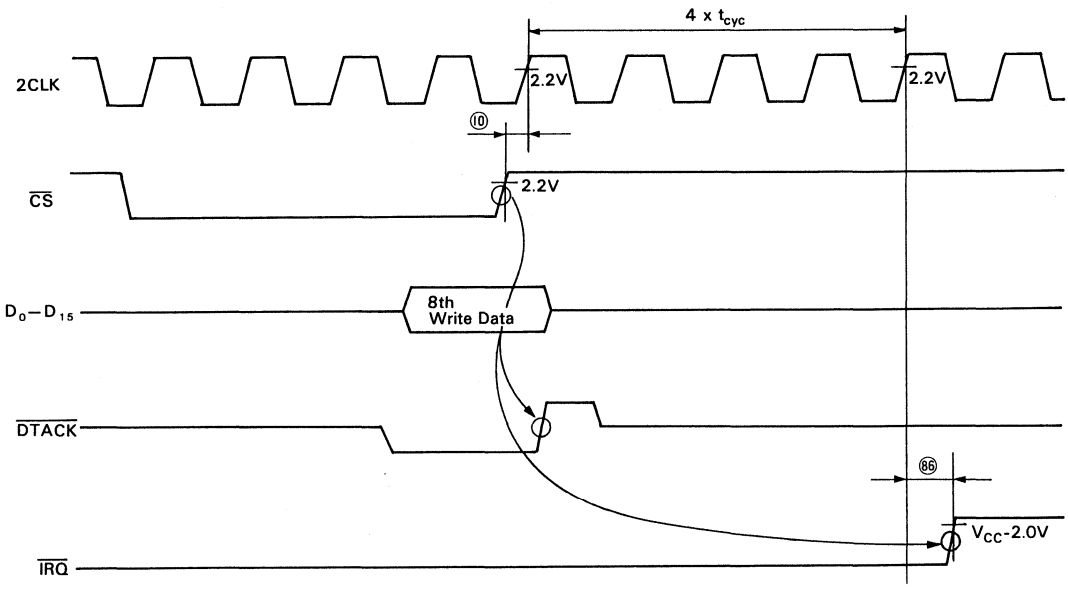


Figure 18 $\overline{IRQ}$ Output Timing



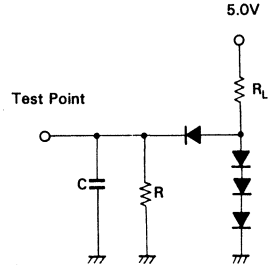
Note) This timing is applicable to WEE, WRE, and RRE.

Figure 19(a) $\overline{IRQ}$ Generate Timing (Example: Read FIFO Full Interrupt Enable)



Note) This timing is applicable to WEE, RRE, and RFE.

Figure 19(b) $\overline{IRQ}$ Negate Timing (Example: Write FIFO Ready Interrupt Enable)



Signal	Load Condition
D_0–D_{15} $\overline{DTACK}$ $\overline{DREQ}$ MAD_0–MAD_{15} MA_{16}/RA_0–MA_{19}/RA_3 RA_4 $\overline{VSYNC}$, $\overline{HSYNC}$ $\overline{EXSYNC}$ MCYC, $\overline{AS}$, MRD $\overline{DRAW}$, CHR $\overline{DISP1}$, $\overline{DISP2}$ $\overline{CUD1}$, $\overline{CUD2}$	R_L = 1.8kΩ C = 40pF R = 10kΩ All diodes are 1S2074Ⓗ's or the equivalent.

Figure 20 Test Load Circuit A

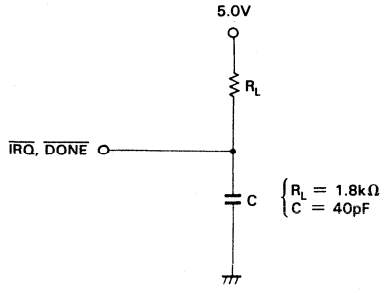


Figure 21 Test Load Circuit B

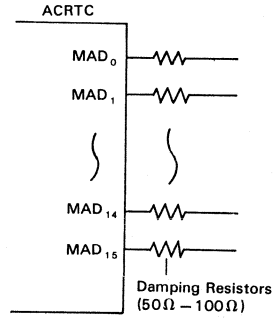


Figure 24 Damping Resistors

Note For Use

(1) Power-on Sequence
The following condition needs to be satisfied at power-on.

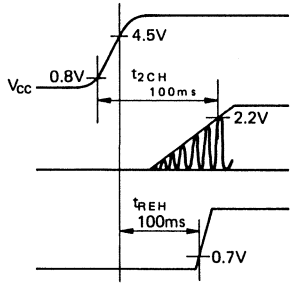
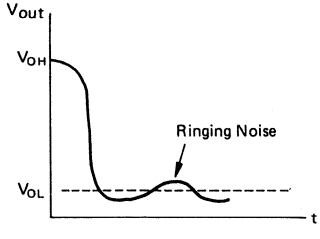


Figure 22 Power-on Sequence

(2) Output Waveform
If excessive ringing noise occurs on CRT data buses (MAD_0–MAD_{15}, MA_{16}/RA_0–MA_{19}/RA_3, RA_4), damping resistors may be required for data buses as shown in the figures 23 and 24.



.Figure 23 Ringing Noise

Note: The ringing level depends on the load capacity, and it can be V_{OL} + 0.1V.

(3) Power Supply Circuit
When designing V_{CC} and V_{SS} pattern of the circuit board, locate capacitors nearest to each power supply pin (V_{CC} and V_{SS}) as shown in the figure 25.

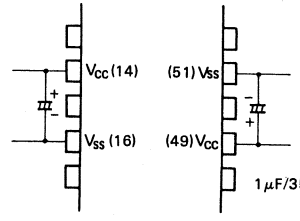


Figure 25(a) Power Supply Circuit Example (64 pin DIP)

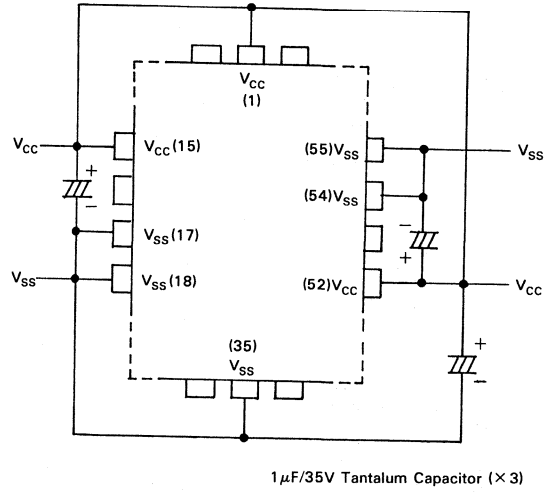


Figure 25(b) Power Supply Circuit Example (68 pin PLCC)

■ SYSTEM CONFIGURATION

Existing CRTCs provide a single bus interface to the frame buffer which must be shared with the host MPU. However, the refresh of large frame buffers and the requirement to access the frame buffer for drawing operations can quickly saturate this shared bus bandwidth.

As shown in figure 26, the ACRTC uses separate host MPU and frame buffer bus interfaces. This allows the ACRTC full access to the frame buffer for display refresh, DRAM refresh, and drawing operations while minimizing the ACRTC's usage of the MPU system bus. Thus, overall system performance is maximized. A related benefit is that a large frame buffer (2M byte for each ACRTC) is useable even if the host MPU has a smaller address space or segment size restriction.

The ACRTC can utilize an external DMA controller. This increases system throughput when large amounts of command, parameter, and data information must be transferred to the ACRTC. Also, advanced DMAC features, such as the HD68450 DMAC's 'chaining' modes, can be used to develop powerful graphics system architectures.

However, more cost sensitive or less performance sensitive applications do not require a DMAC. The interface to the ACRTC can be handled completely under MPU software control.

While both ACRTC bus interfaces (Host MPU and Frame Buffer) are 16-bit data paths for maximum performance, the ACRTC also offers an 8-bit MPU mode for easy connection to popular 8-bit bus structures.

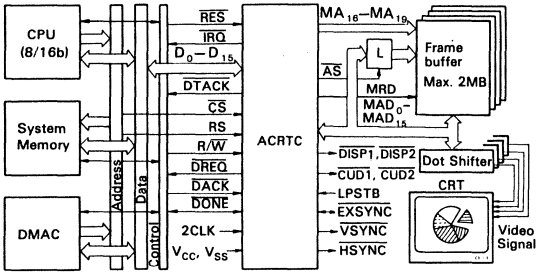


Figure 26 System Configuration

■ INTERNAL FUNCTIONS

• BLOCK DIAGRAM

The ACRTC consists of five major functional blocks. These functional blocks operate in parallel to achieve maximum performance. Two of the blocks perform the external bus interface for the host MPU and CRT respectively.

- MPU Interface
  Manages the asynchronous host MPU interface including the programmable interrupt control unit and DMA handshaking control unit.
- CRT Interface
  Manages the frame buffer bus and CRT timing input and output control signals. Also, either display refresh address or drawing address outputs is selected.
  The other three blocks are separately microprogrammed processors which operate in parallel to perform the major functions of drawing, display control and timing.
- Drawing Processor
  Interprets commands and command parameters issued by the host bus (MPU and/or DMAC) and performs the drawing operations on the frame buffer memory. This processor is responsible for the execution of ACRTC drawing algorithms and conversion of logical pixel X-Y addresses to physical frame buffer addresses.
  Communication with the host bus is via separate 16 byte read and write FIFOs.
- Display Processor
  Manages frame buffer refresh addressing based on the user programmed specification of display screen organization. Combines and displays as many as 4 independent screen segments (3 horizontal splits and 1 window) using an internal high speed address calculation unit. Controls display refresh address outputs based on Graphic (physical frame buffer address) or Character (physical frame buffer address + row address) display modes.
- Timing Processor
  Generates the CRT synchronization signals and other timing signals used internally by the ACRTC.
  The ACRTC's software visible registers are similarly partitioned and reside in the appropriate internal processor depending on function. The registers in the Display and Timing processors are loaded with basic display parameters during system initialization. During operation, the host primarily communicates with the ACRTCs Drawing processor via the on-chip FIFOs.

• SIGNAL DESCRIPTION

Following is a brief description of the ACRTC pin functions organized as MPU Interface, DMAC Interface, CRT Interface and Power Supply.

MPU Interface

$\overline{RES}$ (Input)
Hardware reset input to the ACRTC.

D_0-D_{15} (Input/Output)
The bidirectional data bus for communication with the host MPU or DMAC. In 8 bit data bus mode, D_0-D_7 are used.

R/$\overline{W}$ (Input)
Controls the direction of host ↔ ACRTC transfers.

$\overline{CS}$ (Input)
Enables data transfers between the host and the ACRTC.

RS (Input)
Selects the ACRTC register to be accessed and is normally connected to the least significant bit of the host address bus.

$\overline{DTACK}$ (Output)
Provides asynchronous bus cycle timing and is compatible with the HD68000 MPU $\overline{DTACK}$ input.

$\overline{IRQ}$ (Output)
Generates interrupt service requests to the host MPU.

DMAC Interface

$\overline{DREQ}$ (Output)
Generates DMA service requests to the host DMAC.

$\overline{DACK}$ (Input)
Receives DMA acknowledge timing from the host DMAC.

$\overline{DONE}$ (Input/Output)
Terminates DMA transfer and is compatible with the HD68450 DMAC $\overline{DONE}$ signal.

CRT Interface

2CLK (Input)
Basic ACRTC operating clock derived from the dot clock.

MAD_0-MAD_{15} (Input/Output)
Multiplexed frame buffer address/data bus.

$\overline{AS}$ (Output)
Address strobe for demultiplexing the frame buffer address/data bus (MAD_0-MAD_{15}).

$MA_{16}/RA_0-MA_{19}/RA_3$ (Output)
The high order address bits for graphic screens and the raster address outputs for character screens.

RA_4 (Output)
Provides the high order raster address bit (up to 32 rasters) for character screens.

CHR (Output)
Indicates whether a graphic or character screen is being accessed.

MCYC (Output)
Frame buffer memory access timing – one half the frequency of 2CLK.

MRD (Output)
Frame Buffer data bus direction control.

$\overline{DRAW}$ (Output)
Differentiates between drawing cycles and CRT display refresh cycles.

$\overline{DISP1}$, $\overline{DISP2}$ (Output)
Programmable display enable timing used to selectively enable, disable and blank logical screens.

$\overline{CUD1}$, $\overline{CUD2}$ (Output)
Provides cursor timing determined by ACRTC programmed parameters such as cursor definition, cursor mode, cursor address, etc.

$\overline{VSYNC}$ (Output)
CRT device vertical synchronization pulse.

$\overline{HSYNC}$ (Output)
CRT device horizontal synchronization pulse.

$\overline{EXSYNC}$ (Input/Output)
For synchronization between multiple ACRTCs and other video signal generating devices.

LPSTB (Input)
Connection to an external light pen.

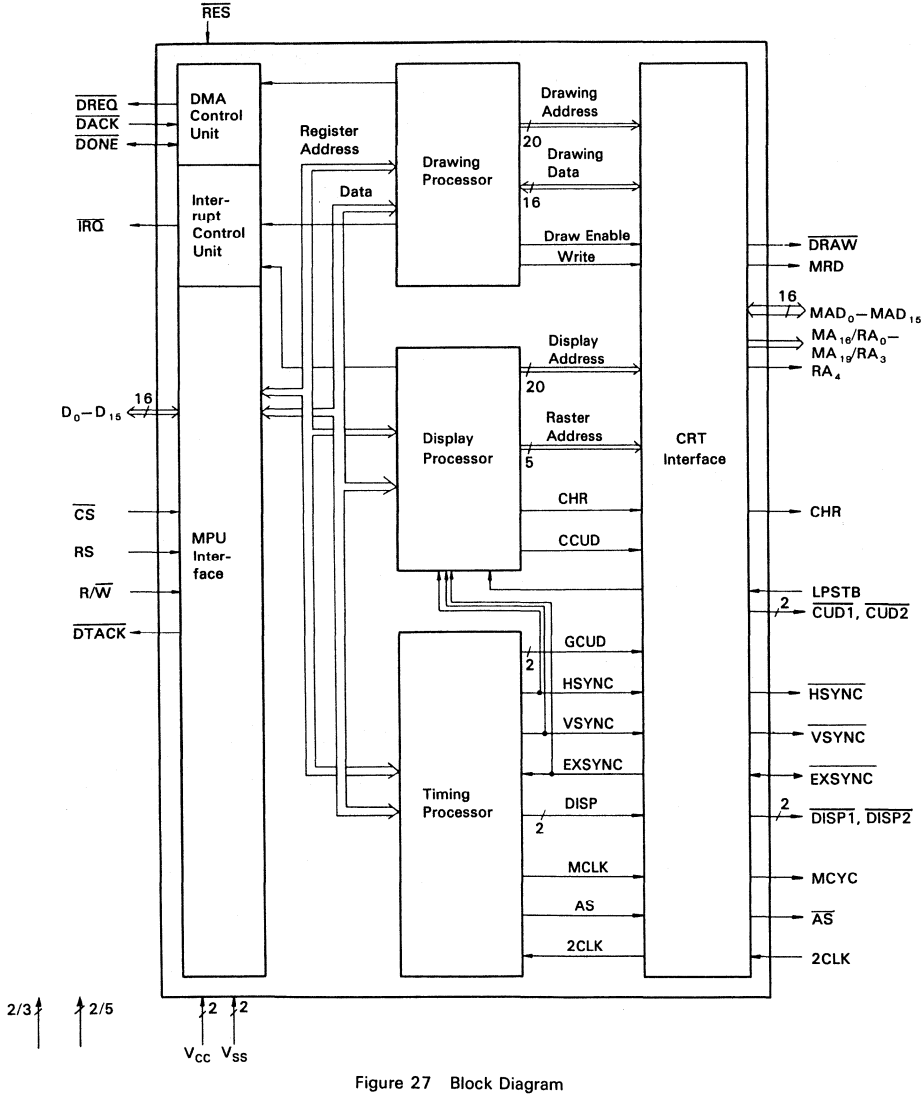


Figure 27 Block Diagram

Video Attributes

The ACRTC outputs 20 bits of video attributes (figure 28) on MAD_0-MAD_{15} and $MA_{16}/RA_0-MA_{19}/RA_3$. These attributes are output at the last cycle prior to the rising edge of $\overline{HSYNC}$ and should be latched externally. Thus, video attributes can be set on a raster by raster basis.

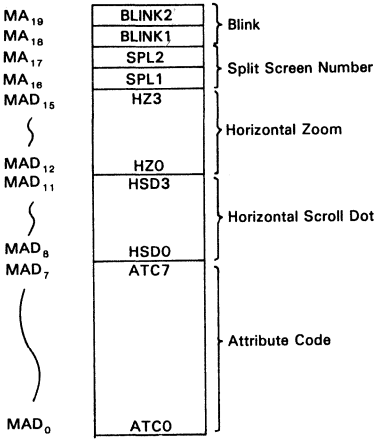


Figure 28 Video Attributes

Attribute Code (ATC0–ATC7: MAD_0-MAD_7)

These are user-defined attributes. The programmed contents of the Attribute Control bits (ATR) of the Display Control Register (DCR) are output on these lines.

Note) The data written into ATR can be externally used after the completion of current raster scanning.

Attribute Code (ATC0–ATC7) Application

The following shows some application examples.

(1) Amount of horizontal dot shift for window smooth scroll.
(2) Horizontal width of crosshair cursor and the amount of horizontal dot shift (including Block cursor).
(3) Frame buffer specification in blocks (used for the base register).
(4) Back ground screen color or character color code.
(5) Display screen selection during screen blink (used with SPL).
(6) Interrupt vector address storage.
(7) Polarity selection of horizontal/vertical synchronization signal.
(8) Blinking signal for indicator lights.
(9) Code storage (max. 8 bit), selection signal, etc.

Horizontal Scroll Dot (HSD0–HSD3: MAD_8-MAD_{11})

These are used in conjunction with external circuitry to implement smooth horizontal scroll. These lines contain the encoded start dot address which is used to control the external shift register load timing and data. HSD usually corresponds to the start dot address of the background screens. However, if the window smooth scroll (WSS) bit of OMR (Operation Mode Register) is set to 1, HSD outputs the start dot address of the window screen segment.

Note) HSD outputs the valid value only within the specified raster area. Changing the register contents during the scanning does not cause any external effects, because the value loaded at the beginning of the area is reserved.

Horizontal Zoom Factor (HZ0–HZ3: $MAD_{12}-MAD_{15}$)

These lines output the encoded (1-16) horizontal zoom factor as stored in the Zoom Factor Register (ZFR). Horizontal zoom is accomplished by the ACRTC repeating a single display address and using the HZ outputs to control the external shift register clock. Horizontal zoom can only be applied to the Base screen.

Split Screen Code (SPL1–SPL2: $MA_{16}-MA_{17}$)

These lines present the encoded information showing the split screen currently being displayed by the ACRTC.

SPL2	SPL1	
0	0	Out of background screen
0	1	Base Screen
1	0	Upper Screen
1	1	Lower Screen

Even if the split screen display is prohibited, SPL is output if the area is specified.

Blink (BLINK1–BLINK2: $MA_{18}-MA_{19}$)

The lines alternate from high to low periodically as defined in the Blink Control Register (BCR). the blink frequency is specified in units of 4 field times. A field is defined as the period between successive VSYNC pulses. These lines are used to implement character and screen blinking.

• ADDRESS SPACE

The ACRTC allows the host to issue commands using logical X-Y coordinate addressing. The ACRTC converts these to physical linear word addresses with bit field offsets in the frame buffer.

Figure 26 shows the relationship between a logical X-Y screen address and the frame buffer memory, organized as sequential 16 bit words. The host may specify that a logical pixel consists of 1, 2, 4, 8 or 16 physical bits in the frame buffer. In the example, 4 bits per logical pixel is used allowing 16 colors or tones to be selected.

Up to four logical screens (Upper, Base, Lower, and Window) are mapped into the ACRTC physical address space. The host specifies a logical screen physical start address, logical screen physical memory width (number of memory words per raster), logical pixel physical memory width (number of bits per pixel) and the logical origin physical address. Then, logical pixel X-Y addresses issued by the host or by the ACRTC Drawing processor are converted to physical frame buffer addresses. The ACRTC also performs bit extraction and masking to map logical pixel operations (in the example, 4 bits) to 16-bit word frame buffer accesses.

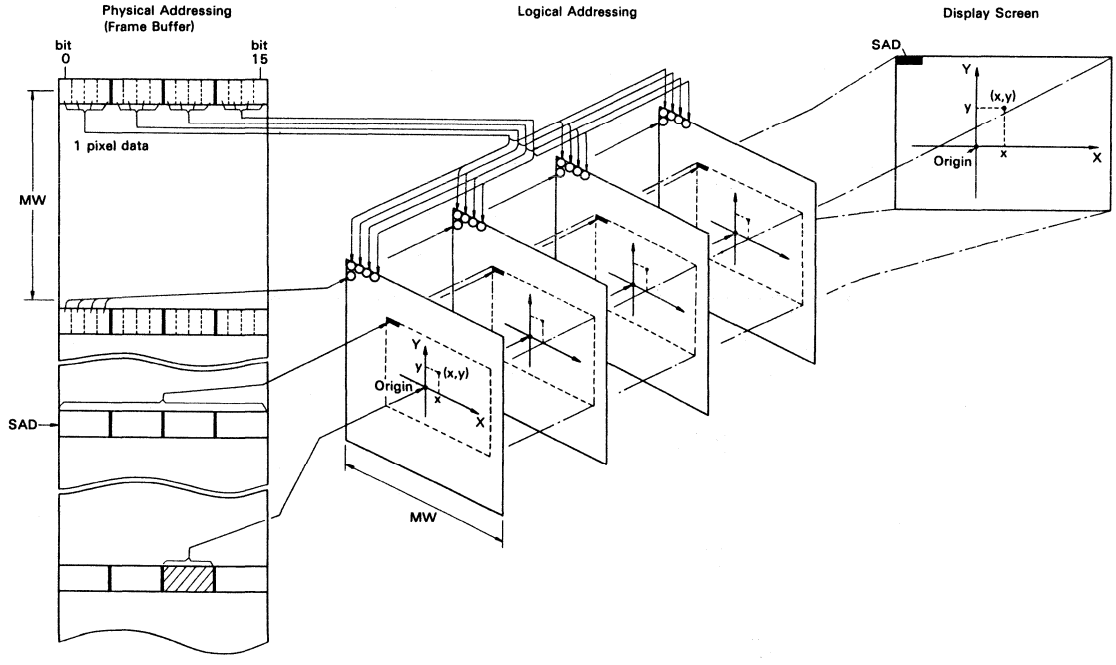


Figure 29 Logical/Physical Addressing

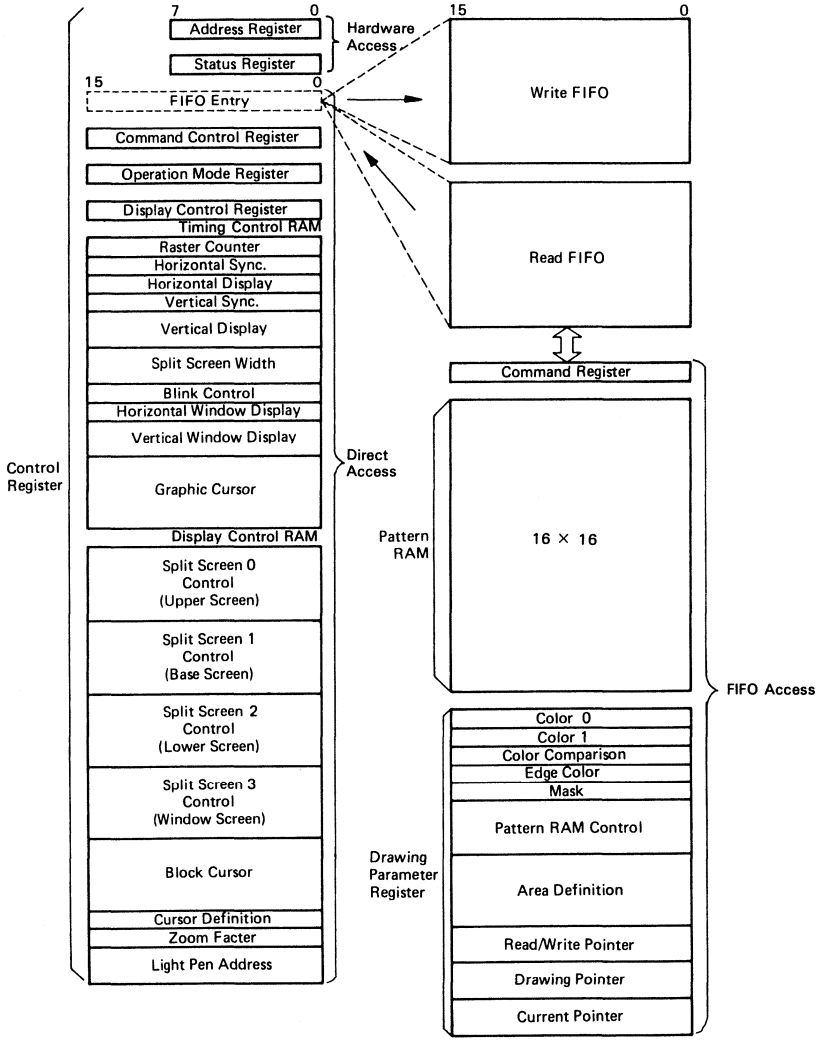


Figure 30 Programming Model

Table 1 Programming Model (Hardware Access, Direct Access Registers)

CS	RS	R/W	Reg. No.	Register Name		Abbre.	DATA (H) 15	14	13	12	11	10	9	8	DATA (L) 7	6	5	4	3	2	1	0
1	-	-	-	—		—	.....								.....							
0	0	0	AR	Address Register		AR	.....								Address							
0	0	1	SR	Status Register		SR	.....								CER	ARD	CED	LPD	RFF	RFR	WFR	WFE
		1/0	r00	FIFO Entry		FE	FE															
		1/0	r02	Command Control		CCR	ART	PSE	DDM	CDM	DRC	GBM			CRF	ARF	CEF	LPE	RFE	RRE	WRE	WEE
		1/0	r04	Operation Mode		OMR	M/S	STR	ACP	WSS	CSK		DSK		RAM	GAI			ACM		RSM	
		1/0	r06	Display Control		DCR	DSP	SE1	SE0		SE2		SE3		ATR							
		-	r08 ↓ r7E	(undefined)		—	.....															
		1	r80	Raster Count		RCR	.....				RC											
		1/0	r82	Horizontal Sync.		HSR	HC								.....				HSW			
			r84	Horizontal Display		HDR	HDS								HDW							
		1/0	r86	Vertical Sync.		VSR	.....				VC											
		1/0	r88	Vertical Display		VDR	VDS								.....				VSW			
		1/0	r8A	Split Screen Width		SSW	.....				SP1											
		1/0	r8C				.....				SP0											
		1/0	r8E				.....				SP2											
		1/0	r90	Blink Control		BCR	BON1				BOFF1				BON2				BOFF2			
		1/0	r92	Horizontal Window Display		HWR	HWS								HWW							
		1/0	r94	Vertical Window Display		VWR	.....				VWS											
		1/0	r96				.....				VWW											
		1/0	r98	Graphic Cursor		GCR	CXE								CXS							
		1/0	r9A				.....				CYS											
		1/0	r9C				.....				CYE											
		-	r9E																			
		-	rA0 ↓ rBE	(undefined)		—	.....															
0	1	1/0	rC0	Upper Screen	Raster Addr.0	RAR0	.....				LRA0				.....				FRA0			
		1/0	rC2		Memory Width 0	MWR0	CHR	.....			MW0											
		1/0	rC4		Start Addr.0	SAR0	.....				SDA0				.....				SA0H/SRA0			
		1/0	rC6				SA0L															
		1/0	rC8	Base Screen	Raster Addr. 1	RAR1	.....				LRA1				.....				FRA1			
		1/0	rCA		Memory Width 1	MWR1	CHR	.....			MW1											
		1/0	rCC		Start Addr. 1	SAR1	.....				SDA1				.....				SA1H/SRA1			
		1/0	rCE				SA1L															
		1/0	rD0	Lower Screen	Raster Addr.2	RAR2	.....				LRA2				.....				FRA2			
		1/0	rD2		Memory Width 2	MWR2	CHR	.....			MW2											
		1/0	rD4		Start Addr.2	SAR2	.....				SDA2				.....				SA2H/SRA2			
		1/0	rD6				SA2L															
		1/0	rD8	Window Screen	Raster Addr. 3	RAR3	.....				LRA3				.....				FRA3			
		1/0	rDA		Memory Width 3	MWR3	CHR	.....			MW3											
		1/0	rDC		Start Addr.3	SAR3	.....				SDA3				.....				SA3H/SRA3			
		1/0	rDE				SA3L															
		1/0	rE0	Block Cursor 1		BCUR1	BCW1				BCSR1				.....				BCER1			
		1/0	rE2				BCA1															
		1/0	rE4	Block Cursor 2		BCUR2	BCW2				BCSR2				.....				BCER2			
		1/0	rE6				BCA2															
		1/0	rE8	Cursor Definition		CDR	CM		CON1			COFF1			.....		CON2			COFF2		
		1/0	rEA	Zoom Factor		ZFR	HZF				VZF				.....							
		1	rEC	Light Pen Address		LPAR	.....								CHR	.....			LPAH			
		1	rEE				LPAL															
		-	rF0 ↓ rFE	(undefined)		—	.....															

Note { 1 ·····"High" level
0 ·····"Low" level

ABT : Abort
ACM : Access Mode
ACP : Access Priority
Address : Control register number
ARD : Area Detect
ARE : Area Detect Interrupt Enable
ATR : Attribute Control
CDM : Command DMA Mode
CED : Command End
CEE : Commad End Interrupt Enable
CER : Command Error
CRE : Command Error Interrupt Enable
CSK : Cursor Display Skew
DDM : Data DMA Mode
DRC : DMA Request Control
DSK : $\overline{\text{DISP}}$ Skew
DSP : $\overline{\text{DISP}}$ Signal Control
FE : FIFO Entry
GAI : Graphic Address Increment Mode
GBM : Graphic Bit Mode
HC : Horizontal Cycle
HDS : Horizontal Display Start
HDW : Horizontal Display Width
HSW : Horizontal Sync. Width
LPD : Light Pen Strobe Detect
LPE : Light Pen Strobe Interrupt Enable
M/S : Master/Slave
PSE : Pause
RAM : RAM Mode
RC : Raster Count
RFE : Read FIFO Full Interrupt Enable
RFF : Read FIFO Full
RFR : Read FIFO Ready
RRE : Read FIFO Ready Interrupt Enable
RSM : Raster Scan Mode
SE0 : Split Enable 0
SE1 : Split Enable 1
SE2 : Split Enable 2
SE3 : Split Enable 3
STR : Start
VC : Vertical Cycle
VDS : Vertical Display Start
VSW : Vertical Sync. Width
WEE : Write FIFO Empty Interrupt Enable
WFE : Write FIFO Empty
WFR : Write FIFO Ready
WRE : Write FIFO Ready Interrupt Enable
WSS : Window Smooth Scroll

SP0, SP1, SP2 : Split Screen 0 Width, Split Screen 1 Width, Split Screen 2 Width
BON1, BON2 : Blink ON 1, Blink ON 2
BOFF1, BOFF2 : Blink OFF 1, Blink OFF 2
HWS : Horizontal Window Start
HWW : Horizontal Window Width
VWS : Vertical Window Start
VWW : Vertical Window Width
CXS, CYS : Cursor X Start, Cursor Y Start
CXE, CYE : Cursor X End, Cursor Y End
FRA : First Raster Address
LRA : Last Raster Address
CHR : Character
MW : Memory Width
SDA : Start Dot Address
SAH/SRA : Start Address "High"/Start Raster Address
SAL : Start Address "Low"
BCW1, BCW2 : Block Cursor Width 1, Block Cursor Width 2
BCSR1, BCSR2 : Block Cursor Start Raster 1, Block Cursor Start Raster 2
BCER1, BCER2 : Block Cursor End Raster 1, Block Cursor End Raster 2
BCA1, BCA2 : Block Cursor Address 1, Block Cursor Address 2
CM : Cursor Mode
CON1, CON2 : Cursor ON 1, Cursor ON 2
COFF1, COFF2 : Cursor OFF 1, Cursor OFF 2
HZF, VZF : Horizontal Zoom Factor, Vertical Zoom Factor
LPAH : Light Pen Address "High"
LPAL : Light Pen Address "Low"

Table 1 (cont.) Programming Model (Drawing Parameter Registers)

Register No.	Read/ Write	Name of Register	Abbr.	Data (H)								Data (L)							
				15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Pr00	R/W	Color 0	CL0	CL0															
Pr01	R/W	Color 1	CL1	CL1															
Pr02	R/W	Color Comparison	CCMP	CCMP															
Pr03	R/W	Edge Color	EDG	EDG															
Pr04	R/W	Mask	MASK	MASK															
Pr05	R/W	Pattern RAM Control	PRC	PPY				PZCY				PPX				PZCX			
↓				PSY				▨				PSX				▨			
Pr07				PEY				PZY				PEX				PZX			
Pr08	R/W	Area Definition **	ADR	XMIN															
↓				YMIN															
				XMAX															
Pr0B				YMAX															
Pr0C	R/W	Read Write Pointer	RWP	DN			▨					RWPH							
Pr0D				RWPL												▨			
Pr0E	—	.....	—	—															
Pr0F				—															
Pr10	R	Drawing Pointer	DP	DN			▨					DPAH							
Pr11				DPAL												DPD			
Pr12	R	Current Pointer **	CP	X															
Pr13				Y															
Pr14	—	.....	—	—															
Pr15				—															

▨ . . . . Always set to "0"
** . . . . Set two's complements for negative values of X and Y axis.

DRAWING PARAMETER REGISTER

R	: Register which can be read by Read Parameter Register Command (RPR)
W	: Register which can be written into by Write Parameter Register Command (WPR)
—	: Access is not allowed
CL0	: Defines the color data used for the drawing when logical drawing data=0
CL1	: Defines the color data used for the drawing when logical drawing data=1
CCMP	: Defines the comparative color of the drawing operation
EDG	: Defines the edge color
MASK	: Defines the bit pattern used to mask bits upon which data transfer should not be performed
PSX, PSY	: Pattern Start Point
PEX, PEY	: Pattern End Point
PPX, PPY	: Pattern Scan Start Point
PZX, PZY	: Pattern Zoom
PZCX, PZCY	: Pattern Zoom Count
XMIN, YMIN	: Start point of Area definition
XMAX, YMAX	: End point of Area definition
DN	: Screen Number
RWPH	: High-order 8 bit of Read Write Pointer Address
RWPL	: Low-order 12 bit of Read Write Pointer Address
DPAH	: High-order 8 bit of Drawing Pointer Address
DPAL	: Low-order 12 bit of Drawing Pointer Address
DPD	: Drawing Pointer Dot Address
X, Y	: Position indicated by Current Pointer on X-Y coordinate

• REGISTERS

The ACRTC has over two hundred bytes of accessible registers. These are organized as Hardware, Directly and FIFO accessible.

○ Hardware Accessible

The ACRTC is connected to the host MPU as a standard peripheral which occupies two word locations of the host address space. The RS (Register Select) pin selects one of these two locations. When RS is low, reads access the Status Register and writes access the Address Register.

The Status Register summarizes the ACRTC state and is used by the MPU to monitor the overall operation of the ACRTC.

The Address Register is used to program the ACRTC with the address of the specific directly accessible register which the MPU wishes to access.

○ Directly Accessible

These registers are accessed by prior loading of the Address Register with the chosen register address. Then, when the MPU accesses the ACRTC with RS=1, the chosen register is accessed.

The FIFO entry enables access to FIFO accessible registers using the ACRTC read and write FIFOs.

The Command Control Register controls overall ACRTC operation such as aborting or pausing commands, defining DMA protocols, enabling/disabling interrupt sources, etc.

The Operation Mode Register defines basic parameters of ACRTC operation such as frame buffer access mode, display or drawing priority, cursor and display timing skew factors, raster scan mode, etc.

The Display Control Register independently enables and disables each of the four ACRTC logical display screens (Base, Upper, Lower and Window). Also, this register contains the 8 bits of user defineable video attributes.

The Timing Control RAM contains registers which define ACRTC timing. This includes timing specification for CRT control signals (e.g. $\overline{\text{HSYNC}}$, $\overline{\text{VSYNC}}$), logical display screen size and display period, blink timing, etc.

The Display Control RAM contains registers which define logical screen display parameters such as start addresses, raster addresses and memory width. Also included are the cursor(s) definition, zoom factor and light pen registers.

○ FIFO Accessible

For high performance drawing, key Drawing Processor registers are coupled to the host via the ACRTCs separate 16 byte read and write FIFOs.

ACRTC commands are sent from the MPU via the write FIFO to the Command register. As the ACRTC completes command execution, the next command is automatically fetched from the FIFO into the Command register.

The Pattern RAM is used to define drawing and painting 'patterns'. The Pattern RAM is accessed using the ACRTCs Read Pattern RAM (RPTN) and Write Pattern RAM (WPTN) register access commands.

The Drawing Parameter Registers define detailed parameters of the drawing process, such as color control, area control (hitting/clipping), and Pattern RAM pointers. The Drawing Parameter Registers are accessed using the ACRTCs Read Parameter Register (RPR) and Write Parameter Register (WPR) register access commands.

■ COMMANDS

The ACRTC has 38 commands classified into three groups — REGISTER ACCESS, DATA TRANSFER, and GRAPHIC DRAWING.

Five REGISTER ACCESS commands allow access to Drawing processor Drawing Parameter Registers and the Pattern RAM.

Ten DATA TRANSFER commands are used to move data between the host system memory and the frame buffer, or within the frame buffer.

Twenty three GRAPHIC DRAWING commands cause the ACRTC to perform drawing operations. Parameters for these commands are specified using logical X-Y addressing.

All the above commands, parameters and data are transferred via the ACRTC read and write FIFOs.

Assuming the ACRTC has been properly initialized, the MPU must perform two steps to cause graphic drawing.

First, the MPU must specify certain drawing parameters which define a number of details associated with the drawing process. For example, to draw a figure or paint an area, the MPU must specify the drawing or painting 'pattern' by initializing the ACRTC Pattern RAM and related pointers. Also, if clipping and hitting control are desired, the MPU specifies the 'area' to be monitored during drawing by initializing area definition registers. Other drawing parameters include color, edge definition, etc.

After the drawing parameters have been specified, the MPU issues a graphic drawing command and any required command parameters, such as the CRCL (Circle) command with a radius parameter. The ACRTC then performs the specified drawing operation by reading, modifying, and rewriting the contents of the frame buffer.

• COMMAND FORMAT

ACRTC commands consist of a 16-bit operation code, optionally followed by 1 or more 16-bit parameters. When 8 bit MPU mode is used, commands, parameters and data are sent to and from the ACRTC in the order of high byte, low byte.

(a) 16 bit interface

For a 16-bit interface, first move the 16-bit operation code and then move the necessary 16-bit parameters one by one.

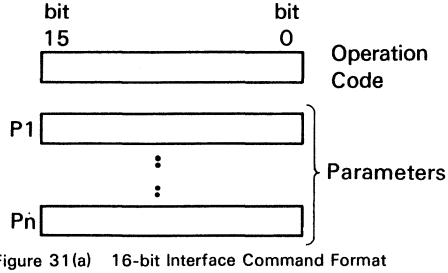


Figure 31(a) 16-bit Interface Command Format

(b) 8 bit interface

For a 8-bit interface, first move the operation code's high byte followed by low byte. Then move those of parameters in the same order.

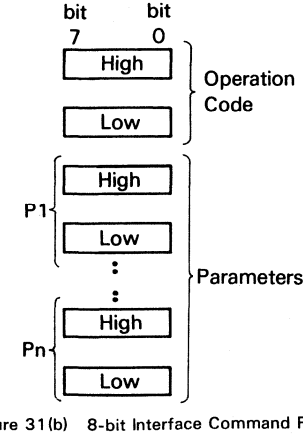


Figure 31(b) 8-bit Interface Command Format

Program Transfer

Program transfer occurs when the MPU specifies the FIFO entry address and then writes commands/parameters to the write FIFO under program control (RS = high, R/W, CS = low). The MPU writes are normally synchronized with ACRTC FIFO status by software polling or interrupts.

- Software Polling (WFR, WFE interrupts disabled)
  a) MPU program checks the SR (Status Register) for Write FIFO Ready (WFR) flag = 1, and the writes 1-word op-code/parameters.
  b) MPU program checks the SR (Status Register) for Write FIFO Empty (WFE) flag = 1, and then writes 1- to 8-word op-code/parameters.
- Interrupt Driven (WFR, WFE interrupts enabled)
  a) MPU WFR interrupt service routine writes 1-word op-code/parameters.
  b) MPU WFE interrupt service routine writes 1- to 8-word op-code/parameters.

  In the specific case of Register Access Commands and an initially empty write FIFO, MPU writes need not be synchronized to the write FIFO status. The ACRTC can fetch and execute these commands faster than the MPU can issue them.

Command DMA Transfer

Commands and parameters can be transferred from MPU system memory using an external DMAC. The MPU initiates and terminates Command DMA Transfer mode under software control (CDM bit of CCR). Command DMA can also be terminated by assertion of the ACRTC $\overline{\text{DONE}}$ signal. $\overline{\text{DONE}}$ is treated as an input in Command DMA Transfer Mode.

Using Command DMA Transfer, the ACRTC will issue cycle stealing DMA requests to the DMAC when the write FIFO is empty. The DMA data is automatically sent from system memory to the ACRTC write FIFO regardless of the contents of the Address Register.

Note 1) Make sure that the write FIFO is empty and all the commands are terminated before starting the Command DMA Transfer.

Note 2) The Data DMA Command cannot be executed in the Command DMA Transfer Mode.

Note 3) In the R mask and S mask versions, the Command DMA Transfer is not in use.

Table 2-1 ACRTC Command Table

TYPE	MNEMONIC	COMMAND NAME	OPERATION CODE	PARAMETER	# (words)	OPERATION CYCLES *1)
Register Access Command	ORG	Origin	0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0	DPH DPL	3	8
	WPR	Write Parameter Register	0 0 0 0 1 0 0 0 0 0 0 RN	D	2	6
	RPR	Read Parameter Register	0 0 0 0 1 1 0 0 0 0 0 RN		1	6
	WPTN	Write Pattern RAM	0 0 0 1 1 0 0 0 0 0 0 PRA	n $D_1, \ldots, D_n$	n+2	4n+8
	RPTN	Read Pattern RAM	0 0 0 1 1 1 0 0 0 0 0 PRA	n	2	4n+10
Data Transfer Command	DRD	DMA Read	0 0 1 0 0 1 0 0 0 0 0 0 0 0 0 0	AX AY	3	(4x+8)y+12[x·y/8↑]+(62~68)
	DWT	DMA Write	0 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0	AX AY	3	(4x+8)y+16[x·y/8↑]+34
	DMOD	DMA Modify	0 0 1 0 1 1 0 0 0 0 0 0 0 0 MM	AX AY	3	(4x+8)y+16[x·y/8↑]+34
	RD	Read	0 1 0 0 0 1 0 0 0 0 0 0 0 0 0 0		1	12
	WT	Write	0 1 0 0 1 0 0 0 0 0 0 0 0 0 0 0	D	2	8
	MOD	Modify	0 1 0 0 1 1 0 0 0 0 0 0 0 0 MM	D	2	8
	CLR	Clear	0 1 0 1 1 0 0 0 0 0 0 0 0 0 0 0	D AX AY	4	(2x+8)y+12
	SCLR	Selective Clear	0 1 0 1 1 1 0 0 0 0 0 0 0 0 MM	D AX AY	4	(4x+6)y+12
	CPY	Copy	0 1 1 0 S DSD 0 0 0 0 0 0 0 0 0 0	SAH SAL AX AY	5	(6x+10)y+12
	SCPY	Selective Copy	0 1 1 1 S DSD 0 0 0 0 0 0 0 0 MM	SAH SAL AX AY	5	(6x+10)y+12
Graphic Command	AMOVE	Absolute Move	1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	X Y	3	56
	RMOVE	Relative Move	1 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0	dX dY	3	56
	ALINE	Absolute Line	1 0 0 0 1 0 0 0 AREA COL OPM	X Y	3	P·L+18
	RLINE	Relative Line	1 0 0 0 1 1 0 0 AREA COL OPM	dX dY	3	P·L+18
	ARCT	Absolute Rectangle	1 0 0 1 0 0 0 0 AREA COL OPM	X Y	3	2P(A+B)+54
	RRCT	Relative Rectangle	1 0 0 1 0 1 0 0 AREA COL OPM	dX dY	3	2P(A+B)+54
	APLL	Absolute Polyline	1 0 0 1 1 0 0 0 AREA COL OPM	n X1, Y1, . . Xn, Yn	2n+2	Σ[P·L+16]+8
	RPLL	Relative Polyline	1 0 0 1 1 1 0 0 AREA COL OPM	n dX1, dY1, . dXn, dYn	2n+2	Σ[P·L+16]+8
	APLG	Absolute Polygon	1 0 1 0 0 0 0 0 AREA COL OPM	n X1, Y1, . . Xn, Yn	2n+2	Σ[P·L+16]+P·Lo+20
	RPLC	Relative Polygon	1 0 1 0 0 1 0 0 AREA COL OPM	n dX1, dY1,...dXn, dYn	2n+2	Σ[P·L+16]+P·Lo+20
	CRCL	Circle	1 0 1 0 1 0 0 C AREA COL OPM	r	2	8d+66
	ELPS	Ellipse	1 0 1 0 1 1 0 C AREA COL OPM	a b dX	4	10d+90
	AARC	Absolute Arc	1 0 1 1 0 0 0 C AREA COL OPM	Xc Yc Xe Ye	5	8d+18
	RARC	Relative Arc	1 0 1 1 0 1 0 C AREA COL OPM	dXc dYc dXe dYe	5	8d+18
	AEARC	Absolute Ellipse Arc	1 0 1 1 1 0 0 C AREA COL OPM	a b Xc Yc Xe Ye	7	10d+96
	REARC	Relative Ellipse Arc	1 0 1 1 1 1 0 C AREA COL OPM	a b dXc dYc dXe dYe	7	10d+96
	AFRCT	Absolute Filled Rectangle	1 1 0 0 0 0 0 0 AREA COL OPM	X Y	3	(P·A+B)B+18
	RFRCT	Relative Filled Rectangle	1 1 0 0 0 1 0 0 AREA COL OPM	dX dY	3	(P·A+B)B+18
	PAINT	Paint	1 1 0 0 1 0 0 E AREA 0 0 0 0 0		1	(18A+102)B-58 *2)
	DOT	Dot	1 1 0 0 1 1 0 0 AREA COL OPM		1	8
	PTN	Pattern	1 1 0 1 SL SD AREA COL OPM	SZ	2	(P·A+10)B+20
	AGCPY	Absolute Graphic Copy	1 1 1 0 S DSD AREA 0 0 OPM	Xs Ys DX DY	5	((P+2)A+10)B+70
	RGCPY	Relative Graphic Copy	1 1 1 1 S DSD AREA 0 0 OPM	dXs dYs DX DY	5	((P+2)A+10)B+70

Note *1) Unit of operation cycle is 2 clock cycle (2CLK)

Note *2) Operation cycle of PAINT is variable. The operation cycle on table 2 is applicable to rectangular figures.

COMMAND ABBREVIATIONS

(a) Register Access Command

RN : Register number of the drawing parameter register ($0-$13)
PRA : Pattern RAM address at which Read/Write operation starts ($0-$F)
DPH : Drawing pointer register High word (figure 32(a))
DPL : Drawing pointer register Low word (figure 32(a))
DPAH : Higher 8 bits of Drawing Pointer address
DPAL : Lower 12 bits of Drawing Pointer address
DPD : Dot position in the memory address

D, D_1,·····, Dn : Write data
n : Number of Read/Write data

[↑] : Round up

(b) Data Transfer Command

MM : Modify mode (Description on page 36)
S : Source scan direction (Table 2-3)
DSD : Destination scan direction (Table 2-4)

AX : Number of word in X-axis direction − 1
AY : Number of word in Y-axis direction − 1
D : Write data
SAH : Source Start Address High word (figure 32(b))
SAL : Source Start Address Low word (figure 32b))

x : Number of word in X-axis direction
y : Number of word in Y-axis direction
↑ : Rounding up

(c) Graphic Drawing Command

AREA : Area mode (Description on page 38)
COL : Color mode (Description on page 38)
OPM : Operation mode (Description on page 36)
C : Circling direction (Table 2-5)
E : Definition of edge color (Table 2-6)
SL : Slant (Table 2-7)
SD : Scan direction (Table 2-7)
S : Source scan direction (Table 2-8)
DSD : Destination scan direction (Table 2-9)

X, X1, . . . , Xn : Absolute X-address from the origin point
Y, Y1, . . . , Yn : Absolute Y-address from the origin point
dX : Relative X-address from the current pointer
dY : Relative Y-address from the current pointer
n : Number of nodes
dX1, . . . , dXn : Relative X-address from each node
dY1, . . . , dYn : Relative Y-address from each node
r : Dot number on radius
a, b : Ratio of squared dX and dY of ellips a:b=(dX)²:(dY)²
DX : X-direction dot number
DY : Y-direction dot number
Xc : Absolute X-address of the center point of arc/ellipse
Yc : Absolute Y-address of the center point of arc/ellipse
dXc : Relative X-address from the current pointer to the center point of arc/ellipse
dYc : Relative Y-address from the current pointer to the center point of arc/ellipse
Xe : Absolute X-address of the end point of arc/ellipse
Ye : Absolute Y-address of the end point of arc/ellipse
dXe : Relative X-address from the current pointer to the end point of arc/ellipse
dYe : Relative Y-address from the current pointer to the end point of arc/ellipse
Xs : Absolute X-address of the start dot position in source area
Ys : Absolute Y-address of the start dot position in source area
dXs : Relative X-address from the current pointer to the start dot position in source area
dYs : Relative Y-address from the current pointer to the start dot position in source area
P : Operation cycles p=4 cycles at OPM=000−011, p=6 cycles at OPM=100−111
L, Lo : Dot number on straight line
d : Total dot number
A : Scan main direction dot number
B : Scan sub direction dot number
SZ : Pattern Size : SZy, SZx

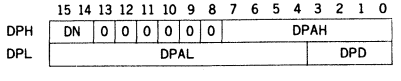


Figure 32(a) Drawing Pointer

Table 2-2 DN : Screen Number

DN	Screen No.
00	Upper Screen
01	Base Screen
10	Lower Screen
11	Window Screen

Table 2-3 S : Source Scan Direction

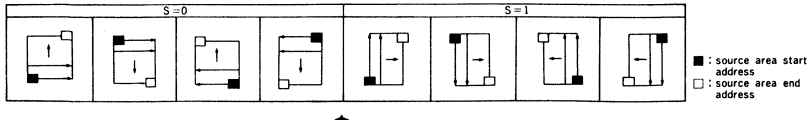
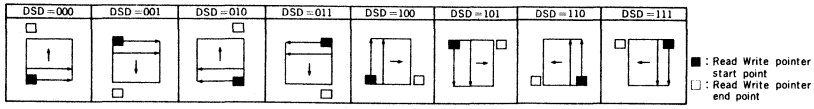


Table 2-4 DSD : Destination Scan Direction



SAH and SAL

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SAH	0	0	0	0	0	0	0	0	(SAH)							
SAL	(SAL)												0	0	0	0

Figure 32(b) Source Start Address

Table 2-5 C : Circling Direction

C	Direction
0	Counterclockwise
1	Clockwise

Table 2-6 E : Definition of Edge Color

E	Definition
0	Edge color is defined by the data in the edge color register.
1	Edge color is defined by the data excluding the above.

Table 2-7 SL : Slant, SD : Scan Direction

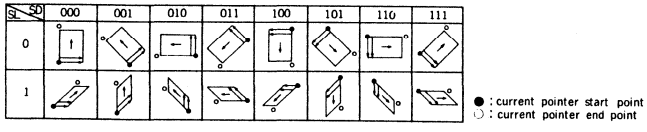


Table 2-8 S : Source Scan Direction

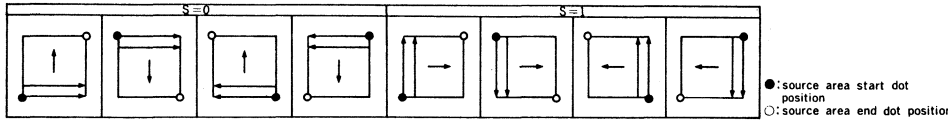
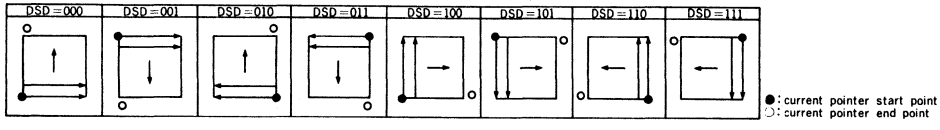


Table 2-9 DSD : Destination Scan Direction



- **REGISTER ACCESS COMMANDS**

Registers associated with the Drawing processor (Pattern RAM and Drawing Parameter Registers) are accessed through the read and write FIFOs using the Register Access Commands.

- **DATA TRANSFER COMMANDS**

Data Transfer Commands are used to move blocks of data between the MPU system memory and the ACRTC frame buffer or within the frame buffer itself. Before issuing these commands, a physical 20 bit frame buffer address must be specified in the RWP (Read Write Pointer) Drawing Parameter Register.

Table 3-1 Register Access Commands

Command	Function
ORG	Initialize the relation between the origin point in the X-Y coordinates and the physical address.
WPR	Write into the parameter register
RPR	Read the parameter register
WPTN	Write into the pattern RAM
RPTN	Read the pattern RAM

Table 4-1 Data Transfer Commands

Command	Function
DRD	Transfer data, by DMA transfer, from the frame buffer to the MPU system memory.
DWT	Transfer data, by DMA transfer, from the MPU system memory to the frame buffer.
DMOD	Transfer data, by DMA transfer, from the MPU system to the frame buffer subject to logical modification (bit maskable).
RD	Read one word of data from the frame buffer specified by the read/write pointer (RWP), and load the word into Read FIFO.
WT	Write one word of data to the frame buffer specified by the read/write pointer (RWP).
MOD	Perform logical operation on one word in the frame buffer specified by the read/write pointer (RWP) (bit maskable).
CLR	Clear a rectangular area of the frame buffer with a data in the command parameter.
SCLR	Initialize a rectangular area of the frame buffer with 1-word data subject to logical operation (bit maskable).
CPY	Copy frame buffer data from one area (source area) to another area (destination area) specified by the read/write pointer (RWP).
SCPY	Copy frame buffer data from one area (source area) to another area (destination area) subject to logical modification by word. The source and destination areas must reside on the same screen (bit maskable).

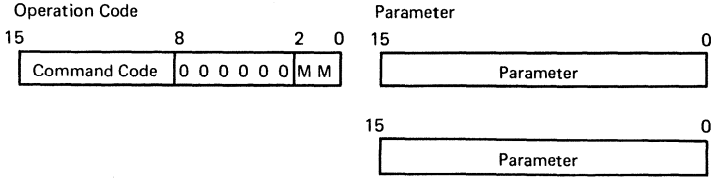


Figure 33 Data Transfer Command Format

Modify Mode

The DMOD, MOD, SCLR and SCPY commands allow 4 types of bit level logical operations to be applied to frame buffer data. The modify mode is encoded in the lower two bits (MM) of these op-codes. The bit positions within each frame buffer word to be modified are selectable using the mask register (MASK). Bits set to 1 are modifiable, ones to 0 are masked and not modifiable.

Table 4-2 Modify Mode

MM		Modify Mode
0	0	REPLACE frame buffer data with command parameter data.
0	1	OR frame buffer data with command parameter data and rewirte to the frame buffer.
1	0	AND frame buffer data with command parameter data and rewrite to the frame buffer.
1	1	EOR frame buffer data with command parameter data and rewrite to the frame buffer.

• GRAPHIC DRAWING COMMANDS

The ACRTC has 23 graphic drawing commands (table 5-1). Graphic drawing is performed by modifying the contents of the frame buffer based upon microcoded drawing algorithms in the ACRTC drawing processor.

Most coordinate parameters for graphic drawing commands are specified using logical pixel X-Y addressing. The complex task of translating a logical pixel address to a linear frame buffer word address, and further selecting the appropriate sub-field of the word (for example, a given logical pixel in 4 bits per logical pixel mode might reside in bits 8-11 of a frame buffer word) is performed at high speed by ACRTC hardware.

Many instructions allow specification of X-Y coordinates with either absolute or relative X-Y coordinates (e.g. ALINE and RLINE). In both cases, two's complement numbers are used to represent positive and negative values.

(a) Absolute Coordinate Specification
The screen address (X, Y) is specified in units of logical pixels relative to an origin point defined with the ORG command (figure 34).

(b) Relative Coordinate Specification
The screen address (dX,dY) is specified in units of logical pixels relative to the current drawing pointer (CP) position (figure 35).

A graphic drawing command consists of a 16-bit op-code and optionally 0 to 64k 16-bit parameters.

The 16-bit op-code consists of an 8-bit command code, an AREA Mode specifier (3 bits), a Color Mode specifier (2 bits) and an Operation Mode specifier (3 bits).

The Area Mode allows versatile clipping and hitting detection. A drawing area can be defined, and should drawing operations attempt to enter or leave that area, a number of programmable actions can be taken by the ACRTC.

The Color Mode determines whether the Pattern RAM is used indirectly to select Color Registers or is directly used as the color information.

The Operation Mode defines one of eight logical operations to be performed between the frame buffer read data and the color data in the Pattern RAM to determine the drawing data to be rewritten into the frame buffer.

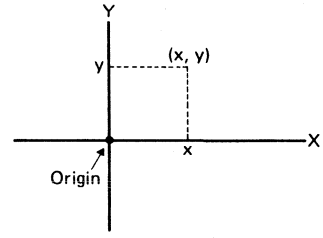


Figure 34 Absolute Coordinate Specification

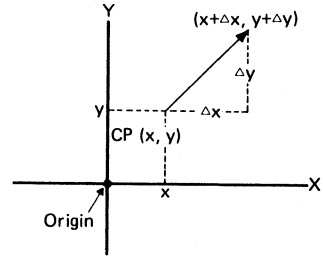


Figure 35 Relative Coordinate Specification

Table 5-1 Graphic Drawing Commands

Command	Function
AMOVE	Move the Current Pointer (CP) to an absolute logical pixel X-Y address.
RMOVE	Move the Current Pointer (CP) to a relative logical pixel X-Y address.
ALINE	Draw a straight line from the Current Pointer (CP) to a command specified end point of the absolute coordinates.
RLINE	Draw a straight line from the Current Pointer (CP) to a command specified end point of the relative coordinates.
ARCT	Draw a rectangle defined by the Current Pointer (CP) and a command specified diagonal point of the absolute coordinates.
RRCT	Draw a rectangle defined by the Current Pointer (CP) and a command specified diagonal point of the relative coordinates.
APLL	Draw a polyline (multiple contiguous segments) from the Current Pointer (CP) through command specified points of the absolute coordinates.
RPLL	Draw a polyline (multiple contiguous segments) from the Current Pointer (CP) through command specified points of the relative coordinates.
APLG	Draw a polygon which connects the start pointer (CP) and command specified points of the absolute coordinates.
RPLG	Draw a polygon which connects the start pointer (CP) and command specified points of the relative coordinates.
CRCL	Draw a circle of the radius R placing the Current Pointer (CP) at the center.
ELPS	Draw a ellipse whose shape is specified by command parameters, placing the Current Pointer (CP) at the center.
AARC	Draw an arc by using the Current Pointer (CP) as a start point with an end point and a center point of the absolute coordinates.
RARC	Draw an arc by using the Current Pointer (CP) as a start point with an end point and a center point of the relative coordinates.
AEARC	Draw an ellipse arc by using the Current Pointer (CP) as a start point with an end point and a center point of the absolute coordinates.
REARC	Draw an ellipse arc by using the Current Pointer (CP) as a start point with an end point and a center point of the relative coordinates.
AFRCT	Paint a rectangular area specified by the Current Pointer (CP) and command parameters (absolute coordinates) according to a figure pattern stored in the Pattern RAM (Tiling).
RFRCT	Paint a rectangular area specified by the Current Point (CP) and command parameters (relative coordinates) according to a figure pattern stored in the Pattern RAM (Tiling).
PAINT	Paint a closed area surrounded by edge color using a figure pattern stored in the Pattern RAM (Tiling).
DOT	Mark a dot on the coordinates where the Current Point (CP) indicates.
PTN	Draw a graphic pattern defined in the Pattern RAM onto a rectangular area specified by the Current Pointer (CP) and by the pattern size (rotation angle: 45°).
AGCPY	Copy a rectangular area specified by the absolute coordinates to the address specified by the Current Pointer (CP) (rotation angle: 90°/mirror turnover).
RGCPY	Copy a rectangular area specified by the relative coordinates to the address specified by the Current Pointer (CP) (rotation angle: 90°/mirror turnover).

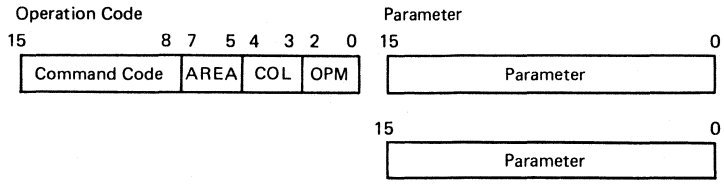


Figure 36 Graphic Drawing Command Format

Operation Mode

The Operation Mode (OPM bits) of the Graphic Drawing Command specify the logical drawing condition.

Figure 37 shows examples of a drawing pattern applied with various OPM modes.

Table 5-2 Operation Mode

OPM			Operation Mode
0	0	0	REPLACE: Replaces the frame buffer data with the color data.
0	0	1	OR: ORs the frame buffer data with the color data. The result is rewritten to the frame buffer.
0	1	0	AND: ANDs the frame buffer data with the color data. The result is rewritten to the frame buffer.
0	1	1	EOR: EORs the frame buffer data with the color data. The result is rewritten to the frame buffer.
1	0	0	CONDITIONAL REPLACE (Read Data=CCMP): When the frame buffer data at the drawing position is equal to the comparison color (CCMP), the frame buffer data is replaced with the color data.
1	0	1	CONDITIONAL REPLACE (Read Data≠CCMP): When the frame buffer data at the drawing position is not equal to the comparison color (CCMP), the frame buffer data is replaced with the color data.
1	1	0	CONDITIONAL REPLACE (Read Data<CL): When the frame buffer data at the drawing position is less than the color register data (CL), the frame buffer data is replaced with the color data.
1	1	1	CONDITIONAL REPLACE (Read Data≧CL): When the frame buffer data at the drawing position is greater than or equal to the color register data (CL), the frame buffer data is replaced with the color data. Note) In case that the read data = CL, and CL is used for the color data to be drawn, replacement cannot be identified, because replaced data is the same color as the read data.

Note 1) The color data is generally the color register data (CL) which is either CL0 or CL1 selected by the pattern pointer. But in case of COL=11, pattern RAM data directly becomes the color data, and in graphic copy commands (AGCPY and RGCPY), it is the source area data.

Note 2) The same color should be set to both CL0 and CL1 (CL0=CL1) when using a graphic copy command at OPM=110 or 111, or when using COL=11 at OPM=110 or 111.

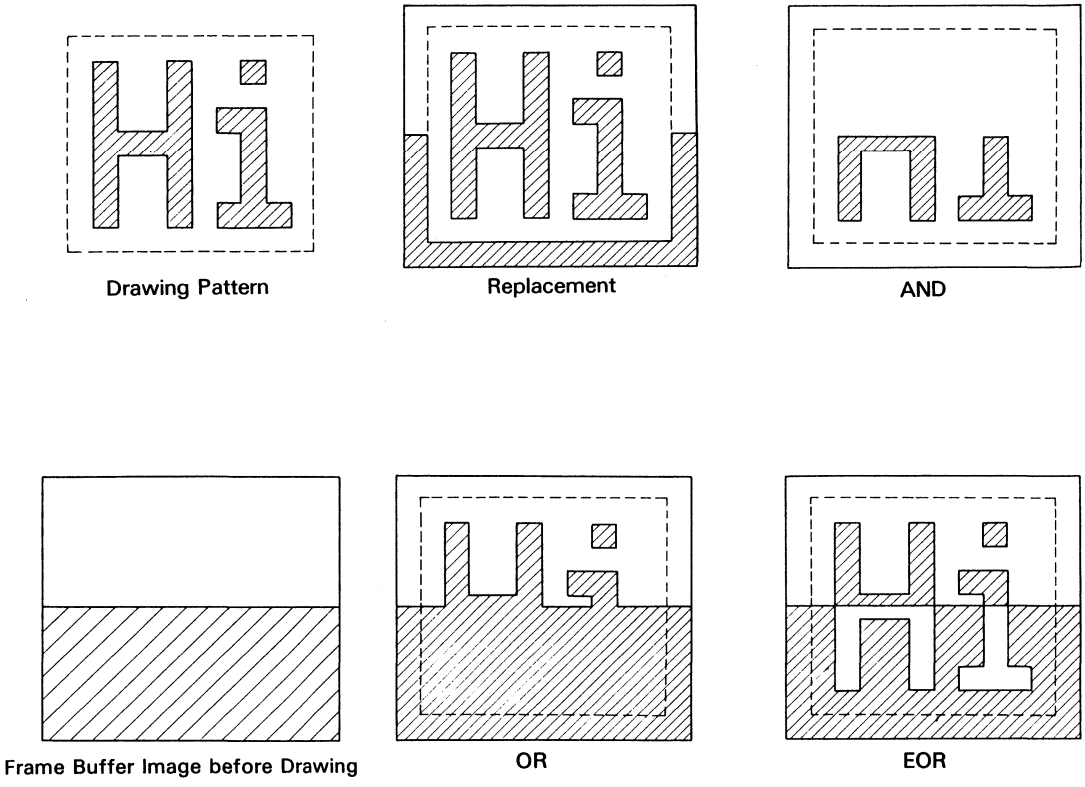


Figure 37 Operation Mode Examples

Color Mode

The Color Mode (COL bits) specifies the source of the drawing color data as directly or indirectly (using the Color Registers) determined by the contents of the Pattern RAM.

Table 5-3 Color Mode

COL		Color Mode
0	0	When Pattern RAM data = 0, Color Register 0 is used. When Pattern RAM data = 1, Color Register 1 is used.
0	1	When Pattern RAM data = 0, drawing is suppressed. When Pattern RAM data = 1, Color Register 1 is used.
1	0	When Pattern RAM data = 0, Color Register 0 is used. When Pattern RAM data = 1, drawing is suppressed.
1	1	Pattern RAM contents are directly used as color data.

The Color Mode chooses the source for color information based on the contents (0 or 1) of a particular bit in the 16 bit by 16 bit (32 byte) Pattern RAM. A sub-pattern is specified by programming the Pattern RAM Control Register (PRC) with the

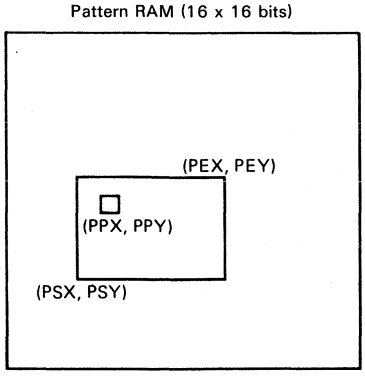


Figure 38 Pattern RAM

start (PSX, PSY) and end (PEX, PEY) points which define the diagonal of the sub-pattern. Furthermore, a specific starting point for Pattern RAM scanning is specified by PPX and PPY.

Normally, the color register (CL0 or CL1) should be loaded with one color data based on the number of bits per pixel. For example, if 4 bits/pixel are used, the 4 bit color pattern (e.g. 0001) should be repeated four times in the color register, i.e.

Color Register = | 0 0 0 1 | 0 0 0 1 | 0 0 0 1 | 0 0 0 1 |

In this way, color changes due to changing dot address are avoided.

Table 5-4 Drawing Area Mode

AREA			Drawing Area Mode
0	0	0	Drawing is executed without Area checking.
0	0	1	When attempting to exit the Area, drawing is stopped after setting ABT (Abort Bit).
0	1	0	Drawing suppressed outside the Area (Drawing operation continues and the ARD flag is not set outside the Area).
0	1	1	Drawing suppressed outside the Area (Drawing operation continues and the ARD flag is set at every drawing operation outside the Area).
1	0	0	Same as AREA = 0 0 0.
1	0	1	When attempting to enter the Area, drawing is stopped after setting ABT (Abort Bit).
1	1	0	Drawing suppressed inside the Area (Drawing operation continues and the ARD flag is not set inside the Area).
1	1	1	Drawing suppressed inside the Area (Drawing operation continues and the ARD flag is set at every drawing operation inside the Area).

Area Mode

Prior to drawing, a drawing area may be defined (Area Definition Register). Then, during Graphics Drawing operation the ACRTC will check if the drawing point is attempting to enter or exit the defined drawing area. Based on eight Area Modes, the ACRTC will take appropriate action for clipping or hitting.

■ SYSTEM INTERFACE

• BASIC CLOCK

The ACRTC basic clock is 2CLK. 2CLK controls all primary ACRTC display and logic timing parameters.

2CLK, along with the specification of number of bits per logical pixel, the Graphic Address Increment mode, and the Display Access mode, also determines the video data rate.

The basic clock must be input with its cycle, max. and min. of "High" and "Low" level width as shown in the AC characteristics.

In any case, be careful not to stop the basic clock, fixing it at "High", "Low", or open, which can destroy the LSI.

• CRT INTERFACE

Frame Buffer Access

(1) Access Modes

The three ACRTC display memory access modes are Single, Interleaved and Superimposed.

(a) Single Access Mode

A display (or drawing) cycle is defined as two cycles of 2CLK. During the first 2CLK cycle, the frame buffer display or drawing address is output. During the second 2CLK cycle, the frame buffer data is read (display cycles and/or drawing cycles) or written (drawing cycles).

In this mode, display and drawing cycles contend for access to the frame buffer. The ACRTC allows the priority to be defined as display priority or drawing priority. If display has priority, drawing cycles are only allowed to occur during horizontal/vertical flyback period. So, a 'flashless' display is obtained at the expense of slower drawing. If drawing has priority, drawing may occur during display, so high speed drawing is obtained. However the display may flash.

(b) Interleaved Access Mode (Dual Access Mode 0)

In this mode, display cycles and drawing cycles are interleaved. A display/drawing cycle is defined as four cycles of 2CLK. During the first 2CLK cycle, the frame buffer display address is output. During the second 2CLK cycle, the display data is read from the frame buffer. During the third 2CLK cycle, the frame buffer drawing address is output. During the fourth 2CLK cycle, the drawing data is read or written.

Since there is no contention between display and drawing cycles, a 'flashless' display is obtained while maintaining full drawing speed. However, for a given configuration, frame buffer memory access time must be twice as fast as an equivalent Single Access Mode configuration.

(c) Superimposed Access Mode (Dual Access Mode 1)

In this mode, two separate logical screens are accessed during each display cycle. The display cycle is defined as four 2CLK cycles. During the first 2CLK cycle, the Background (Upper, Base, or Lower) screen frame buffer address is output. During the second 2CLK cycle, the Background screen display data is read. During the third 2CLK cycle, the window screen frame buffer address or the drawing frame buffer address is output. During the fourth 2CLK cycle, the window screen display or drawing data is read (display or drawing) or written (drawing). Note that the third and fourth cycles can be used for drawing (similar to Interleaved mode) when these cycles are not used for Window display.

SA (SINGLE ACCESS MODE)

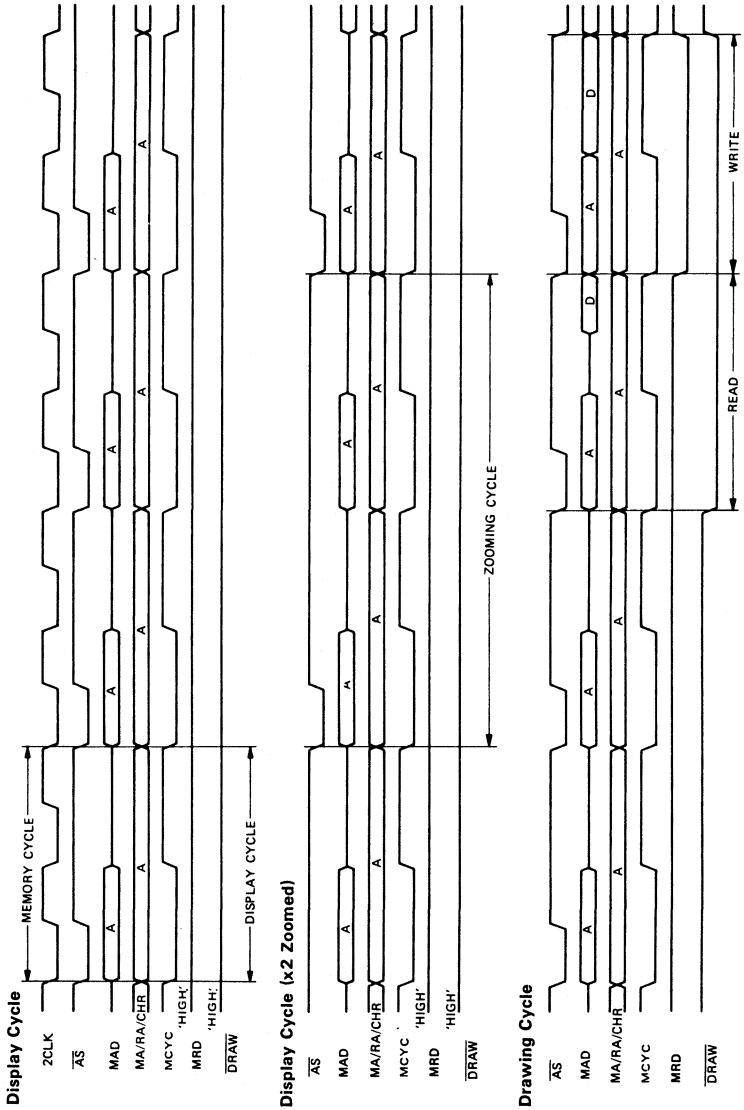


Figure 39 Single Access Mode Timing

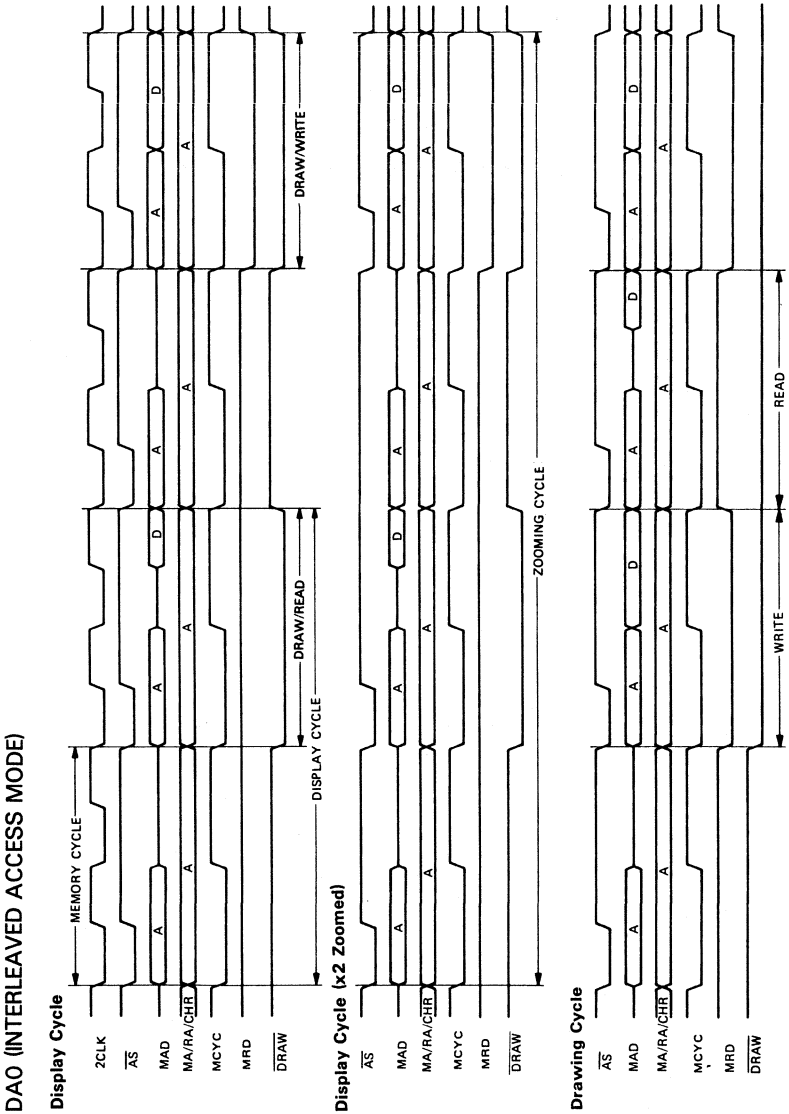


Figure 40 Interleaved Access Mode Timing

DA1 (SUPERIMPOSED ACCESS MODE)

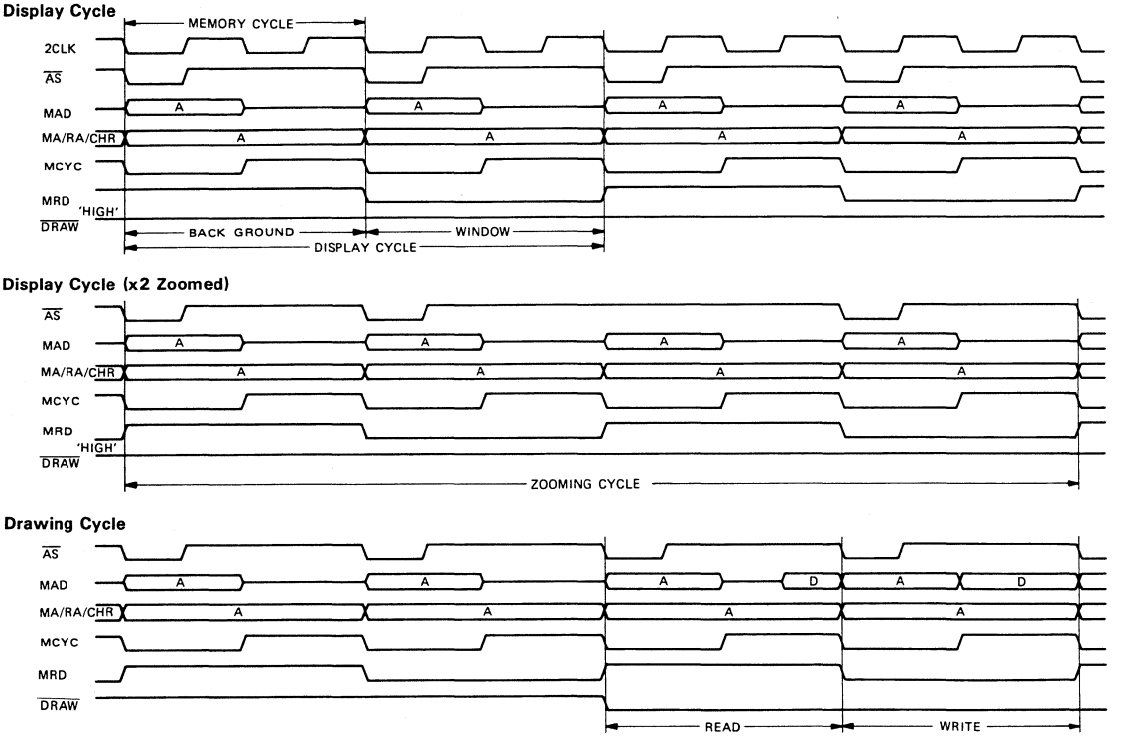


Figure 41 Superimposed Access Mode Timing

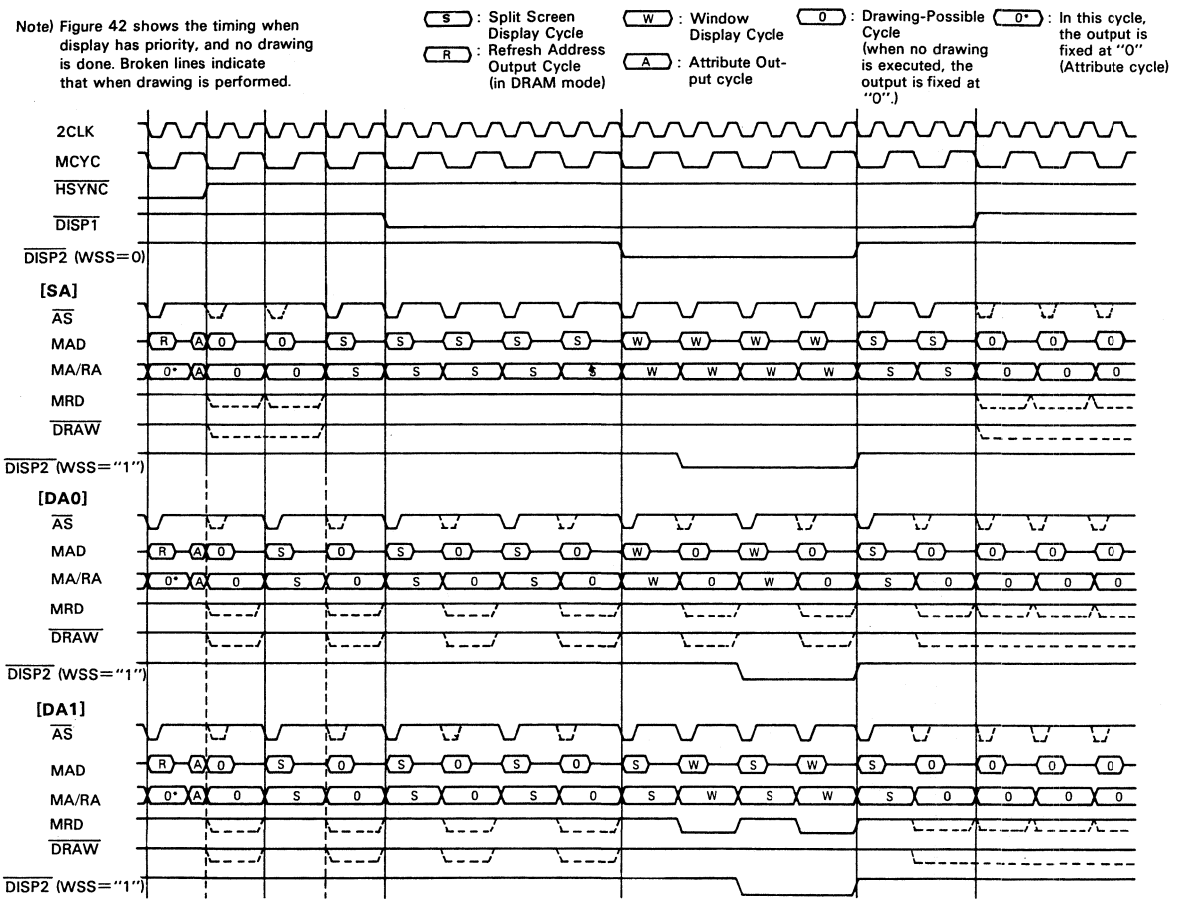


Figure 42 Horizontal Scan Sequence

(2) Graphic Address Increment Mode (GAI)

During display operation, the ACRTC can be programmed to control the graphic display address in seven ways including increment by 1, 2, 4, 8 and 16* words, 1 word every two display cycles and no increment.

Setting GAI to increment by 2, 4, 8 or 16* words per display cycle achieves linear increases in the video data rate i.e. for a given configuration setting GAI to 2, 4, 8 or 16* words will achieve 2, 4, 8 or 16* times the video data rate corresponding to GAI=1. This allows increasing the number of bits/logical pixel and logical pixel resolution while meeting the 2CLK maximum frequency constraint.

Table 6 shows the summary relationship between 2CLK, Display Access Mode, Graphic Address Increment, number of bits/logical pixel, memory access time and video data rate. The frame buffer cycle frequency (Fc) is shown by the following equation where:

Fv = Dot Clock
N = Number of bits/logical pixel
D = Display Access Mode
1 for Single Access Mode
2 for Interleaved and Superimposed Access Modes
A = Graphic Address Increment (1/2, 1, 2, 4, 8, 16*)
Fc = $(Fv \times N \times D)/(A \times 16)$

Table 6 Graphic Address Increment Modes

Dot Rate		16 MHz		32 MHz		64 MHz		128 MHz	
Number of Bits/Pixel	Access Mode / Memory Cycle	SA	DA	SA	DA	SA	DA	SA	DA
1	250 ns	–	+1/2	+1/2	+1	+1	+2	+2	+4
	500 ns	+1/2	+1	+1	+2	+2	+4	+4	+8
2	250 ns	+1/2	+1	+1	+2	+2	+4	+4	+8
	500 ns	+1	+2	+2	+4	+4	+8	+8	+16*
4	250 ns	+1	+2	+2	+4	+4	+8	+8	+16*
	500 ns	+2	+4	+4	+8	+8	+16*	+16*	–
8	250 ns	+2	+4	+4	+8	+8	+16*	+16*	–
	500 ns	+4	+8	+8	+16*	+16*	–	–	–
16	250 ns	+4	+8	+8	+16*	+16*	–	–	–
	500 ns	+8	+16*	+16*	–	–	–	–	–

Note) * R mask version does not +16 increment mode.

Dynamic RAM Refresh

When dynamic RAMs (DRAMs) are used for the frame buffer memory, the ACRTC can automatically provide DRAM refresh addressing.

The ACRTC maintains an 8 bit DRAM refresh counter which is decremented on each frame buffer access. During $\overline{\text{HSYNC}}$ low, the ACRTC will output the sequential refresh addresses on MAD. The refresh address assignment depends on Graphic Address Increment (GAI) mode as shown in Table 7.

The ACRTC provides "0" output on the remaining address line of MAD and MA/RA.

DRAM refresh cycle timing must be factored into the determination of $\overline{\text{HSYNC}}$ low pulse width (HSW – specified in units of frame buffer memory cycles).

If the horizontal scan rate is Fh (kHz), number of DRAM refresh cycles is N and the DRAM refresh cycle time is Tr (msec) then horizontal sync width (HSW) is specified by the following equation:

$HSW \geqq N / (Tr \times Fh)$

For example, if the scan rate is 15.75 kHz and the DRAMS have 128 refresh cycles of 2 ms, HSW must be greater than or equal to 5.

$HSW \geqq 128 / (2 \times 15.75) = 4.06$

Table 7 GAI and DRAM Refresh Addressing

Graphic Address Increment Mode	Refresh Address Output Terminal
+0 (GAI=101)	MAD_0~MAD_7
+1 (GAI=000)	MAD_0~MAD_7
+2 (GAI=001)	MAD_1~MAD_8
+4 (GAI=010)	MAD_2~MAD_9
+8 (GAI=011)	MAD_3~MAD_{10}
+16 (GAI=100)*	MAD_4~MAD_{11}
+1/2 (GAI=$^{111}_{110}$)	MAD_0~MAD_7

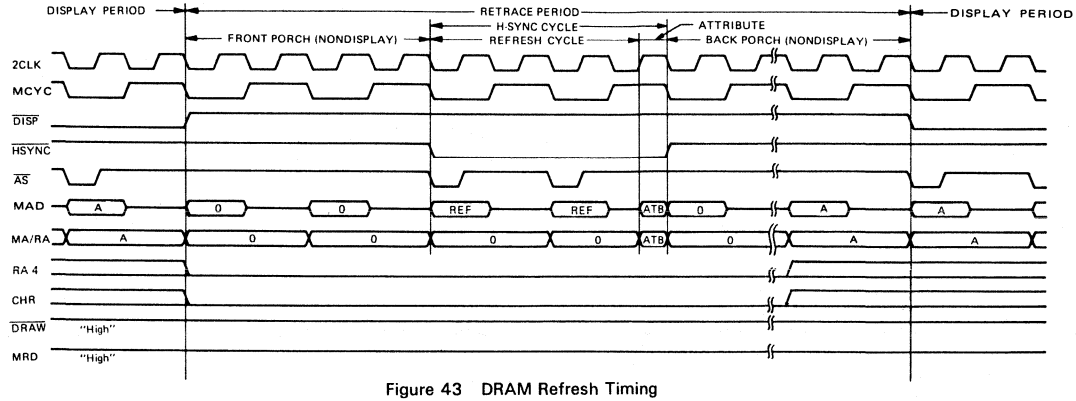


Figure 43 DRAM Refresh Timing

External Synchronization

The ACRTC $\overline{\text{EXSYNC}}$ pin allows synchronization of multiple ACRTCs or other video signal generators. The ACRTC may be programmed as a single Master device, or as one of a number of Slave devices.

To synchronize multiple ACRTCs, simply connect all the $\overline{\text{EXSYNC}}$ pins together.

For synchronizing to other video signals, the connection scheme depends on the raster scan mode. In Non-Interlace mode, $\overline{\text{EXSYNC}}$ corresponds to $\overline{\text{VSYNC}}$. In Interlace modes, $\overline{\text{EXSYNC}}$ corresponds to $\overline{\text{VSYNC}}$ of the odd field.

Note 1) The ACRTC performs the synchronization every time it accepts the pulse input from $\overline{\text{EXSYNC}}$ in the slave mode.
It is recommended that the synchronous pulse should be input from $\overline{\text{EXSYNC}}$ only when the synchronization gap between the synchronous signal of the master device and that of ACRTC in the slave mode ($\overline{\text{HSYNC}}$ and $\overline{\text{VSYNC}}$ are output also in the slave mode.).

Note 2) The ACRTC needs to be controlled not to draw during $\overline{\text{EXSYNC}}$ input.

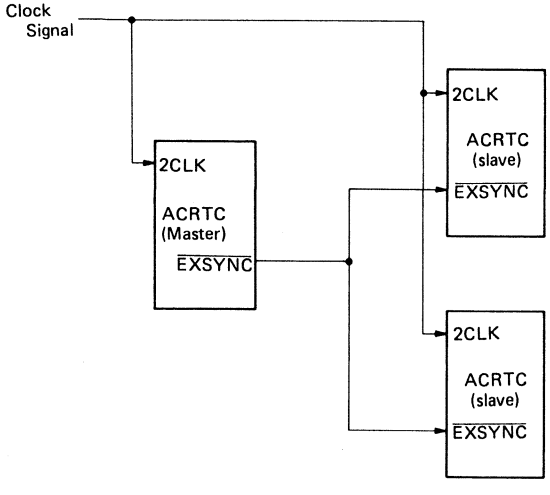


Figure 44 External Synchronization

• MPU INTERFACE

MPU Bus Cycle

The ACRTC interfaces to the MPU as an 8 or 16 bit peripheral as configured during $\overline{\text{RES}}$.

An MPU bus cycle is initiated when $\overline{\text{CS}}$ is asserted (following the assertion of RS and R/$\overline{\text{W}}$). The ACRTC responds to $\overline{\text{CS}}$ low by asserting $\overline{\text{DTACK}}$ low to complete the data transfer. $\overline{\text{DTACK}}$ will be returned to the MPU in between 1 and 1.5 2CLK cycles.

MPU WAIT states will be added in the following two cases.

(a) If the ACRTC 2CLK input is much slower than the MPU clock, continuous ACRTC accesses may be delayed due to internal processing of the previous bus cycle.
Note) $\overline{\text{CS}}$ "High" width must not be less than two 2CLK cycles.

(b) If a read cycle immediately follows a write cycle, a WAIT state may occur due to ACRTC preparation for bus 'turn-around'. However, MPUs (for example 68000) normally have no instruction which lets a read cycle follow a write cycle immediately.

For connection to synchronous bus interface MPUs, $\overline{\text{DTACK}}$ can simply be left open assuming the system design guarantees that WAIT states cannot occur as described above. If WAIT states may occur, $\overline{\text{DTACK}}$ can be used with external logic to synthesize a READY signal.

DMA Transfer

The ACRTC can interface with an external DMA controller using three handshake signals, DMA Request ($\overline{\text{DREQ}}$), DMA Acknowledge ($\overline{\text{DACK}}$) and DMA Done ($\overline{\text{DONE}}$).

The ACRTC uses the external DMAC for two types of transfers, Command/Parameter DMA and Data DMA. For both types, DMA transfers use the ACRTC read and write FIFOs.

(1) Command/Parameter DMA

The MPU initiates this mode by setting bit 12 (CDM) in the ACRTC Command Control Register to 1. Then, the ACRTC will automatically request DMA transfer for commands and their associated parameters as long the write FIFO has space. Only cycle steal request mode ($\overline{\text{DREQ}}$ pulses low for each data transfer) can be used. Command/Parameter DMA is terminated when the MPU resets bit 12 in CCR to 0 or the external $\overline{\text{DONE}}$ input is asserted.

Note) The R mask version and the S mask version can't perform Command/Parameter DMA transfer. So CDM (bit 12) should be set to 0.

(2) Data DMA

Data DMA is used to move data between the MPU system

memory and the ACRTC frame buffer.

The MPU sets-up the transfer by specifying the frame buffer transfer address (and other parameters of the transfer, such as 'on-the fly' logical operations) to the ACRTC. Next, when the MPU issues a Data Transfer Command to the ACRTC, the ACRTC will request DMA transfer to and from system memory. The ACRTC will request DMA, automatically monitoring FIFO status, until the DMA Transfer Command is completed.

Data DMA request mode can be cycle steal (as in Command/Parameter DMA) or burst mode in which $\overline{DREQ}$ is a low level control output to the DMAC which allows multiple data transfers during each acquisition of the MPU bus.

Interrupts

The ACRTC recognizes eight separate conditions which can generate an interrupt including command error detection, command end, drawing edge detection, light pen strobe and four FIFO status conditions. Each condition has an associated mask bit for enabling/disabling the associated interrupt. The ACRTC removes the interrupt request when the MPU performs appropriate interrupt service by reading or writing to the ACRTC.

■ DISPLAY FUNCTION

• SCREEN DISPLAY CONTROL

Logical Display Screens

The ACRTC allows division of the frame buffer into four separate logical screens.

Screen Number	Screen Name	Screen Group Name
0	Upper Screen	Background Screens
1	Base Screen	
2	Lower Screen	
3	Window Screen	

In the simplest case, only the Base screen parameters must be defined. Other screens may be selectively enabled, disabled and blanked under software control.

The Background (Upper, Base and Lower) screens partition the display into three horizontal splits whose positions are fully programmable. A typical application might use the Base screen for the bulk of user interaction, using the Lower screen for a 'status line(s)' and the Upper screen for 'pull-down menu(s)'.

The Window screen is unique, since the ACRTC usually gives the Window screen higher priority than Background screens. Thus, when the Window, whose size and position is fully programmable, overlaps a Background screen, the Window screen is displayed. The exception is the ACRTC Superimposed Access Mode, in which the Window has the same display priority as Background screens. In this case, the Window and Background screen are 'superimposed' on the display.

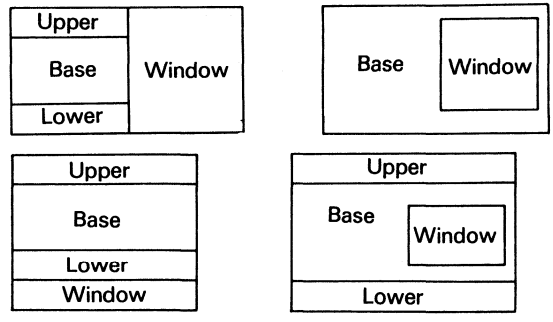


Figure 45 Screen Combination Examples

Frame Memory Setup

The ACRTC can have two types of independent frame memories, 2M byte frame buffer and 128k byte refresh memory, and CHR signal controls which memory to be accessed.

For the frame memory, memory width is defined by setting up Memory Width Register (MWR), and horizontal display width is independently defined by Horizontal Display Register (HDR). Therefore the frame memory area can be specified bigger than display area as shown in figure 46.

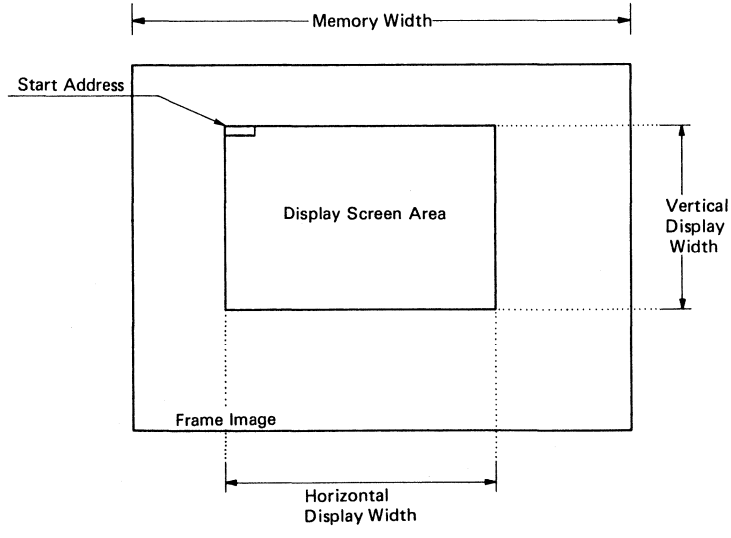


Figure 46 Frame Memory and Display Screen Area

Display Control

Figure 47 shows the relation between the frame memory and the display screens.

Each screen has its own memory width, start address, vertical display width, and attribution of frame memory (character/graphic), and those are specified by the control registers.

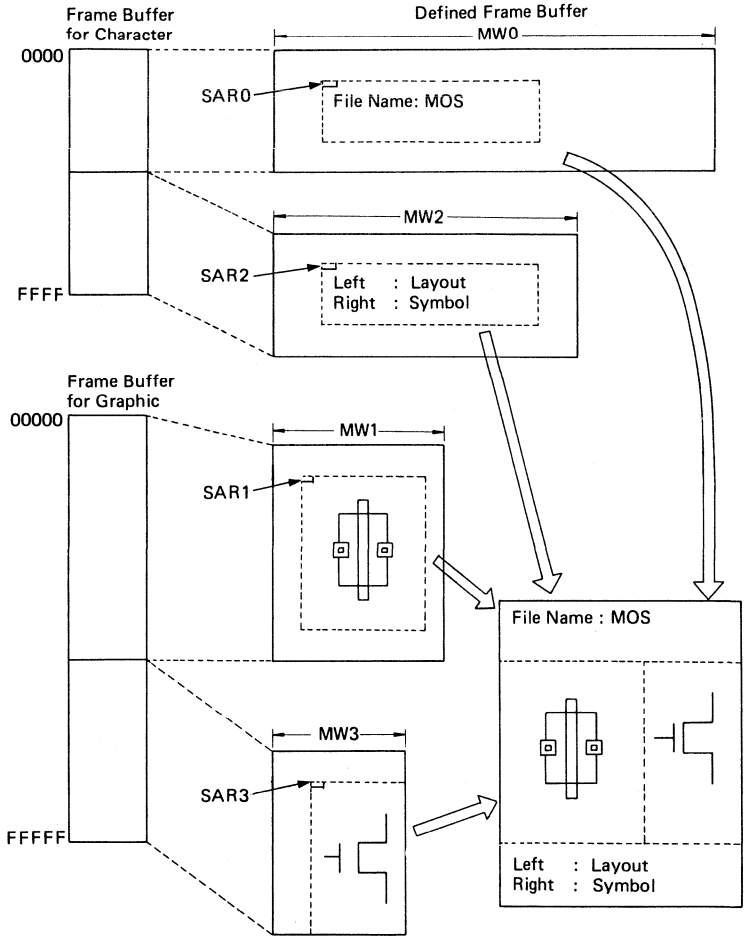


Figure 47 Frame Memory and Display Screens

Figure 48 shows the relation between the control registers and the display screens. Registers for horizontal display control are set in units of memory cycles, and registers for vertical display control are set in units of rasters.

Note a display width specified by * marked register is:

(Display width) = (Setup value) + 1 memory cycles

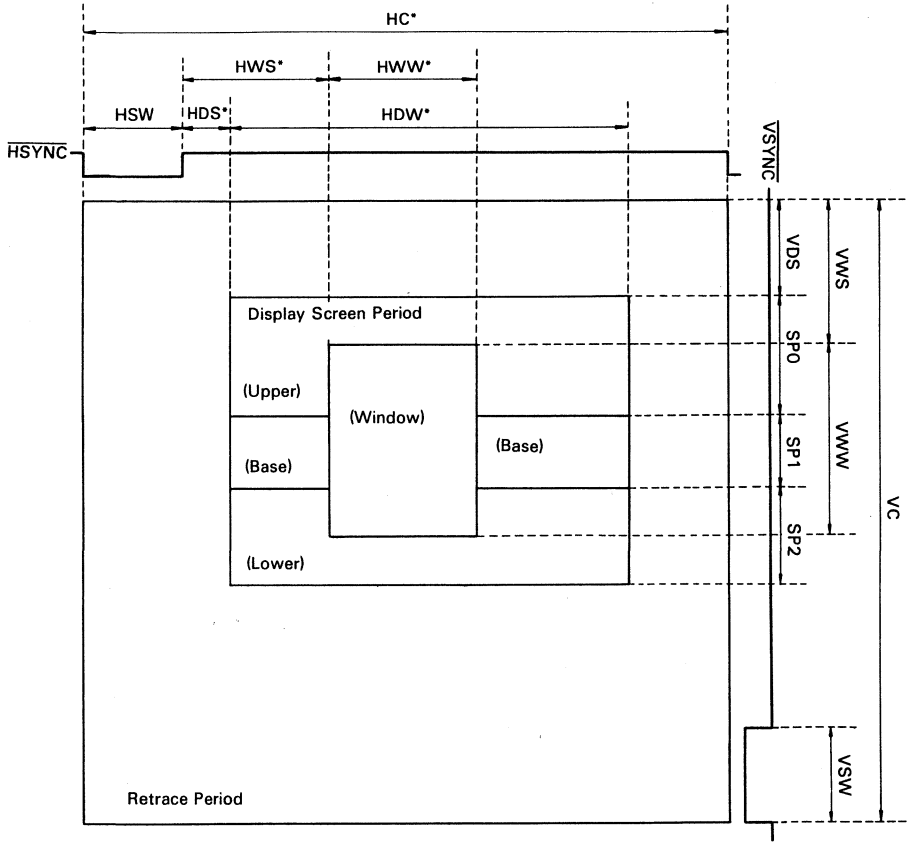


Figure 48 Display Screen Specification

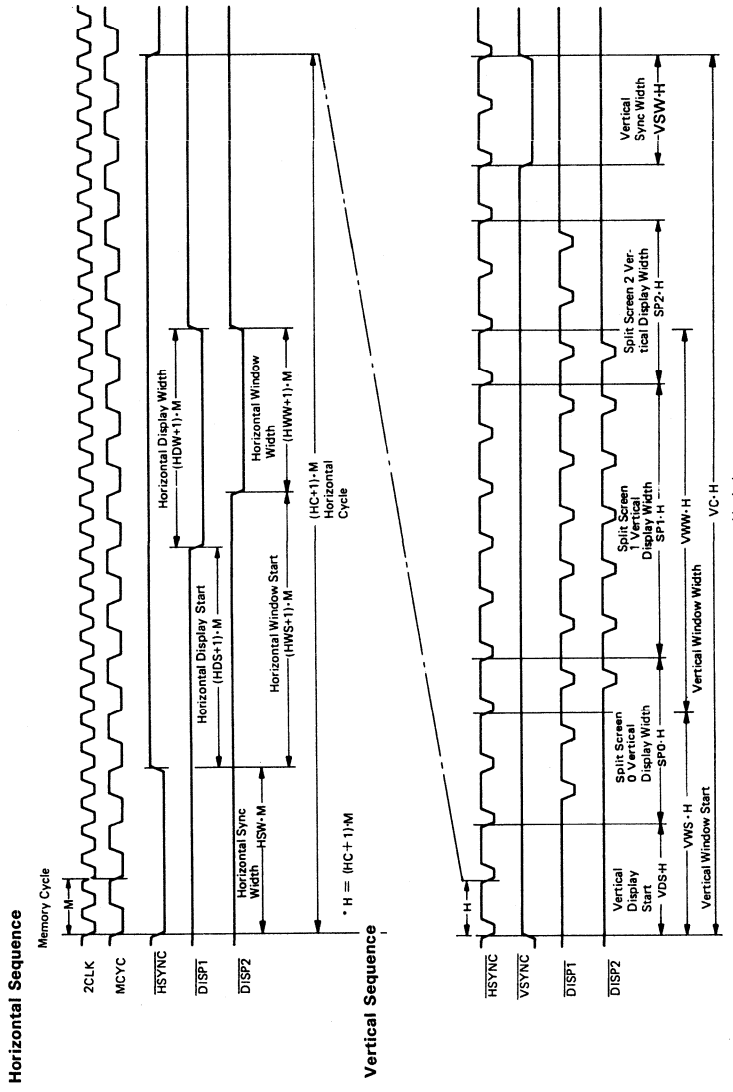


Figure 49 Display Scan Sequence (Horizontal and Vertical)

Graphic/Character Address Spaces

The ACRTC controls two separate logical address spaces. The CHR pin allows external decoding if physically separate frame memories are desired.

Each of the four logical screens (Upper, Base, Lower and Window) is programmed as residing in the Graphics address space or the Character address space.

ACRTC accesses to Graphics screens are treated as bit mapped using a 20 bit frame buffer address, with an address space of one megaword (1M by 16 bit).

ACRTC accesses to Character screens are treated as character generator mapped. In this case, a 64k word address space is used and 5 bits of raster address are output to an external character generator.

Multiple logical screens defined as Character can be externally decoded to use separate character generators or different addresses within a combined character generator. Also, each Character screen may be defined with separate line spacing, separate cursors, etc.

- **CURSOR CONTROL**

The ACRTC has two Block Cursor Registers and a Graphic Cursor Register.

A Block cursor is used with Character screens. The cursor start and ending raster addresses are fully programmable. Also, the cursor width can be defined as one to eight memory cycles.

A Graphic cursor is defined by specifying the start/end memory cycle in the X dimension and the start/end raster in the Y dimension.

The Graphic cursor can be output on character and Graphic screens.

The ACRTC provides two separate cursor outputs, $\overline{\text{CUD1}}$ and $\overline{\text{CUD2}}$. These are combined with two character cursor registers and a graphic cursor register to provide three cursor modes.

(1) Block Mode
Two Block cursors are output on $\overline{\text{CUD1}}$ and $\overline{\text{CUD2}}$ respectively.

(2) Graphic Mode
The Graphic cursor is output on $\overline{\text{CUD1}}$. Using an external cursor pattern memory allows a graphic cursor of various shapes. Two Block cursors are multiplexed on $\overline{\text{CUD2}}$.

(3) Crosshair Mode
The horizontal and vertical components of the Graphic cursor are output on $\overline{\text{CUD1}}$ and $\overline{\text{CUD2}}$ respectively. This allows simple generation of a crosshair cursor control signal.

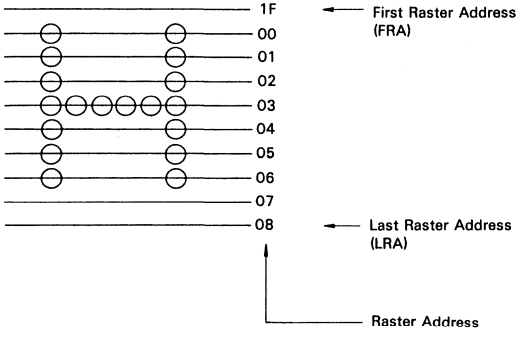


Figure 50 Character Screen Raster Addressing

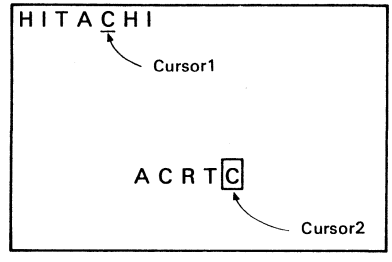


Figure 51 Two Separate Block Cursors

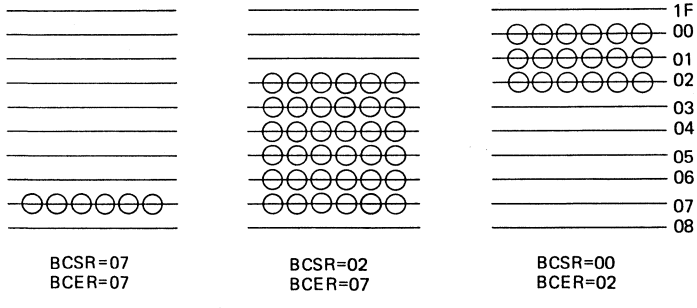


Figure 52 Block Cursor Examples

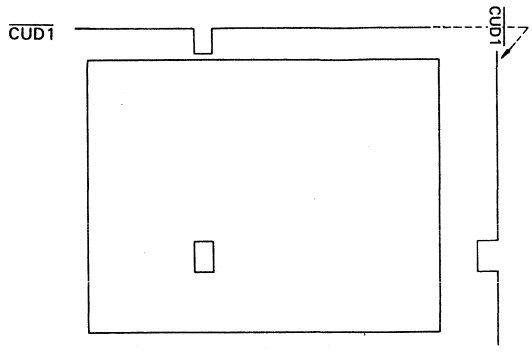


Figure 53 Graphic Cursor

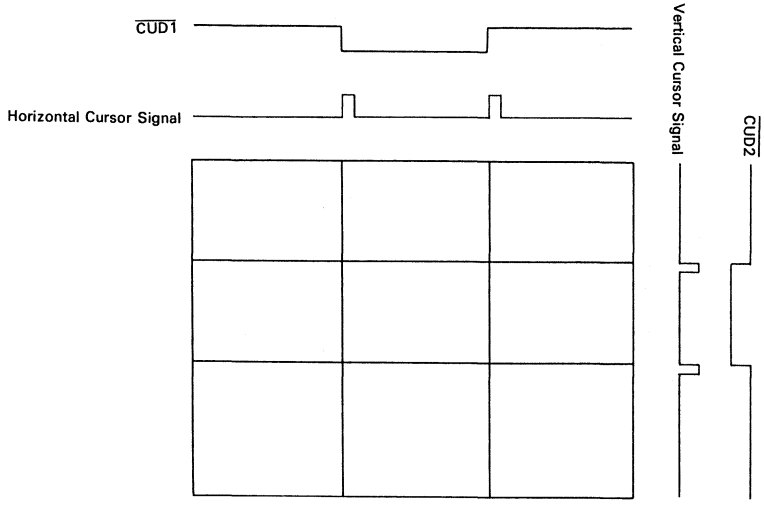


Figure 54 Crosshair Cursor

• SCROLLING

Vertical Scroll

Each logical screen performs independent vertical scroll. On Character Screens, vertical smooth scroll is accomplished using the programmable Start Raster Address (SRA). Line by line scroll is accomplished by increasing or decreasing the screen start address by one unit of horizontal memory width.

On Graphics screens, vertical smooth scroll is accomplished by increasing or decreasing the screen start address by one unit of horizontal memory width.

Horizontal Scroll

Horizontal scroll can be performed in units of characters for Character screens and units of words (multi logical pixels) for Graphic screens by increasing or decreasing the screen start address by 1.

For smooth horizontal scroll, the ACRTC has dot shift video attributes which can be used with an external circuit which conditions shift register load/clocking.

Since this dot shift information is output each raster, horizontal smooth scroll is limited to either the Background screens or the Window screen at any given time. However, horizontal smooth scroll is independent for each of the Background screens (Upper, Base, Lower).

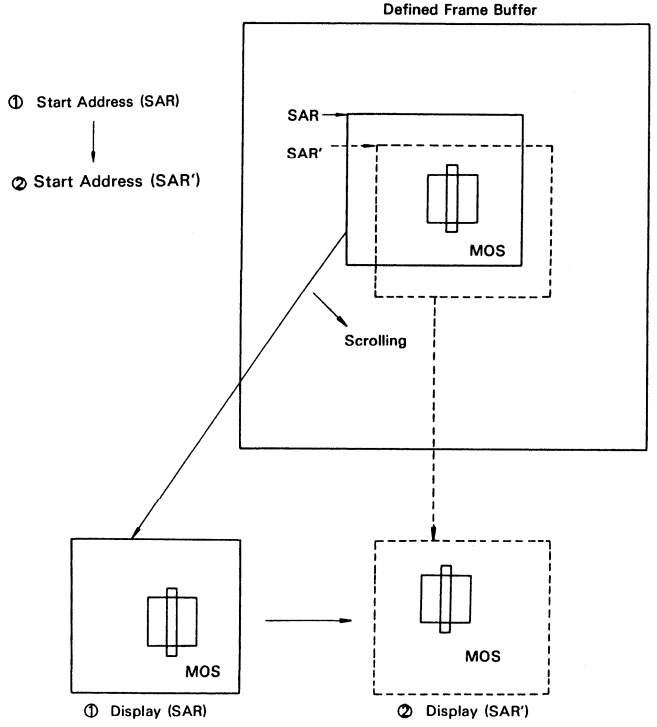


Figure 55 Scrolling by SAR (Start Address Register) Rewrite

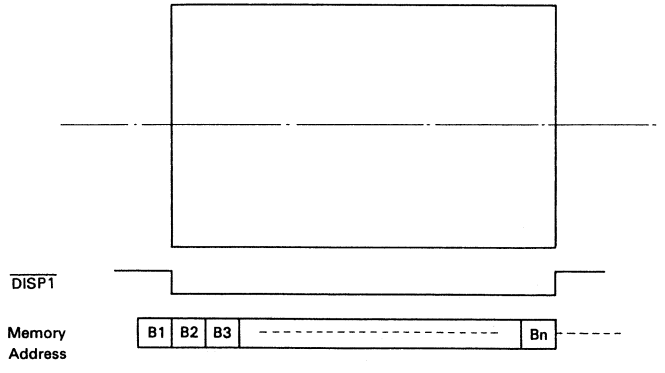


Figure 56 Horizontal Smooth Scroll — Background Screen

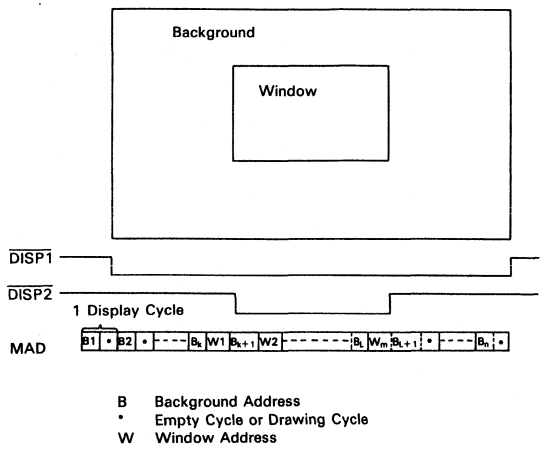


Figure 57 Horizontal Smooth Scroll — Window Screen (WSS=1)

• RASTER SCAN MODES

The ACRTC has three software selectable raster scan modes — Non-Interlace, Interlace Sync and Interlace Sync & Video. In Non-Interlace mode a frame consists of one field. In the Interlace modes, a frame consists of two fields, the even and odd fields.

The Interlace modes allow increasing screen resolution while avoiding limits imposed by the CRT display device, such as maximum horizontal scan frequency or maximum video dot rate.

Interlace Sync mode simply repeats each raster address for both the even and odd fields. This is useful for increasing the quality of a displayed figure when using an interlaced CRT device such as a Television Set with RF modulator.

Interlace Sync & Video mode displays alternate even and odd rasters on alternate even and odd fields. For a given number of rasters/character, this mode allows twice as many characters to be displayed in the vertical direction as Non-Interlace mode.

Note that for Interlace modes, the refresh frequency for a given dot on the screen is one-half that of the Non-Interlace mode. Interlace modes normally require the use of a CRT with a more persistent phosphor to avoid a flickering display.

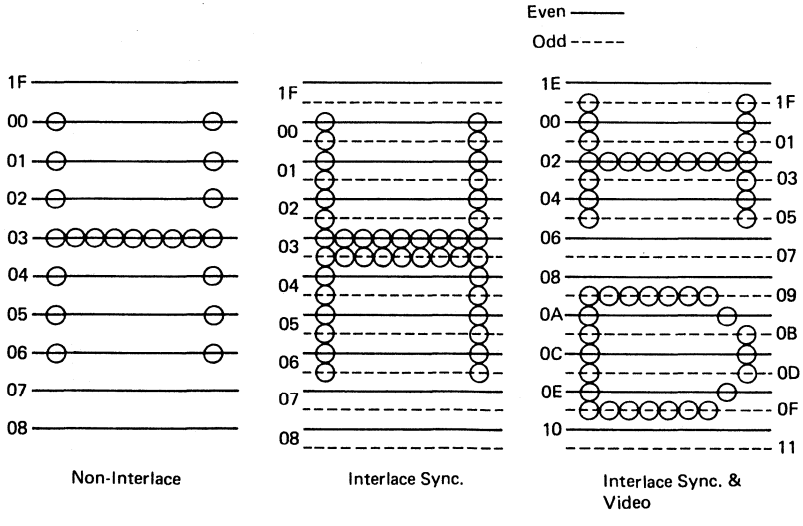


Figure 58 Raster Scan Modes

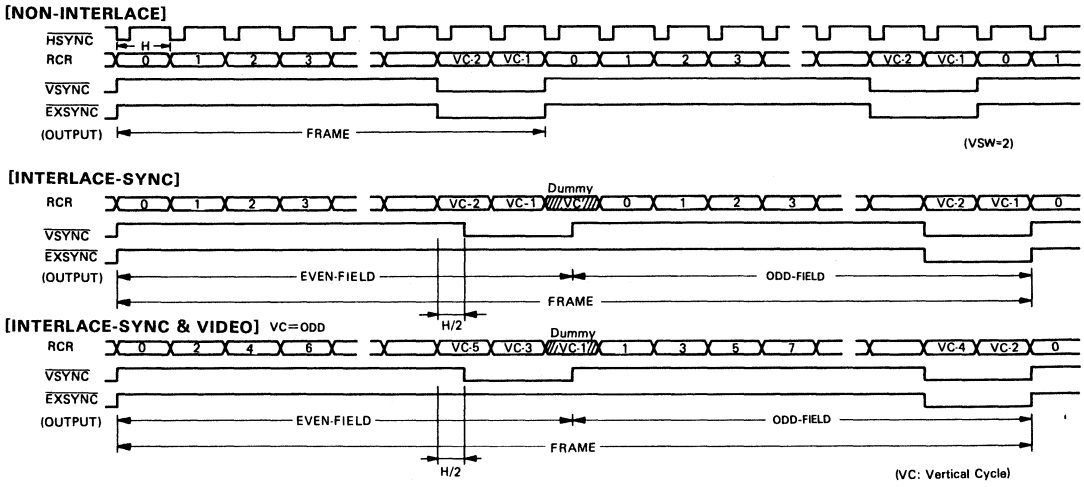


Figure 59 Raster Scan Timing

• ZOOMING

The ACRTC supports a zoom function for the Base screen (Screen 1). Note that ACRTC zooming is performed by controlling the CRT timing signals. The contents of the frame buffer area being zoomed are not changed.

The ACRTC allows specification of zoom factors (1 to 16) independently in the X and Y directions.

For horizontal zoom, the programmed zoom factor is output as video attributes. An external circuit uses this factor to condition the external shift register clock to accomplish horizontal zooming.

For vertical zoom, no external circuit is required. The ACRTC will scan a single raster multiple times to accomplish vertical zooming.

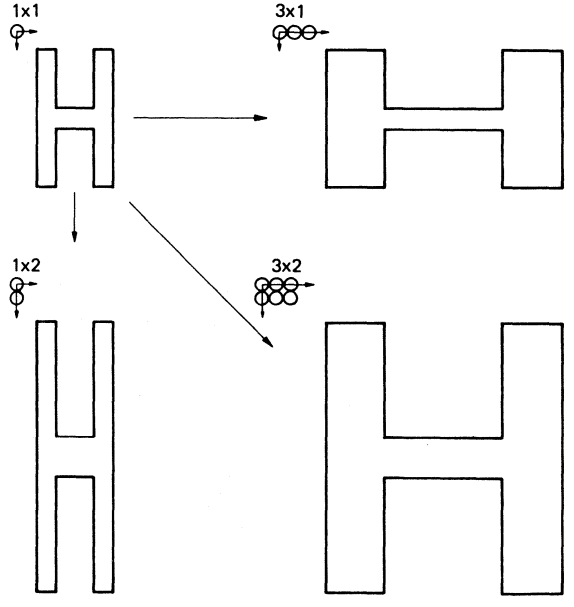


Figure 60 Zooming

• LIGHT PEN

The ACRTC provides a 20 bit Light Pen Address Register and a Light Pen Strobe (LPSTB) input pin for connection with a light pen.

A light pen strobe pulse will occur when the CRT electron beam passes under the light pen during display refresh. When this pulse occurs, the contents of the ACRTC display refresh address counter will be latched into the Light Pen Address Register along with a logical screen (Character or Graphic screen) designator. Also, an ACRTC status flag indicating light pen activity is set, generating an optional (maskable) MPU interrupt. Note that for Superimposed access mode, when the light pen strobe occurs in an area in which the Window overlaps a Background screen, the Background screen address will be latched. And even for all access mode, the Drawing address will never be latch.

Various system and ACRTC delays will cause the latched address to differ slightly from the actual light pen position. The light pen address can be corrected using software, based upon system specific delays. Or, if the application does not require the highest light pen pointing resolution, software can 'bound' the light pen address by specifying a range of values associated with a given area of the screen.

■ NOTES ON SYSTEM DESIGNING

• GND IMPEDANCE

Problem Description

In case that load capacitance on frame memory address/data bus (MAD) and GND impedance are large, noise may occur on output signals especially at the following timings:

1) Starting of address output on MAD_0-MAD_{15} (plus $MA_{16}/RA_0-MA_{19}/RA_3$)
2) MAD_0-MAD_{15} turning high impedance after address output
3) Starting and finishing of drawing data output on MAD_0-MAD_{15}
4) Starting and finishing of video attributes output on MAD_0-MAD_{15} (plus $MA_{16}/RA_0-MA_{19}/RA_3$)

The noise generally occurs sharply on "Low" leveled output signals as shown in figure 61 and figure 64. This problem is caused by GND impedance, as MAD bus and output signal lines are sharing GND line in side the LSI as shown in figure 62. Depending on impedance of the GND line, the GND level is influenced by excessive discharge on internal GND line which occurs when MAD state changes from high impedance to "Low" level, from "High" level to "Low" level, or from "High" level to high impedance.

In case that load capacitance on bi-directional host system data bus and GND impedance are large, noise may occur on output signals at data turnings. However, the level of this noise is much lower than that of the former case, because the GNDs for the data bus and output signal lines are almost separated to minimize the noise.

Note the level of the noise increases depending on the load capacitance. So the load capacitance should be less than the condition in fiure 20.

Problem Analysis

Figure 63 shows the relations between the noise voltage and parameters: V_{CC}, GND impedance, load capacitance on MAD, and number of MAD lines which cause discharge. As level of noise depends on these parameters and output signal pins, it is required to check the condition of the system. Following output signal pins tend to have remarkable noise:

$\overline{MA_{16}}/RA_0-MA_{19}/RA_3$, RA_4, CHR, MRD, $\overline{DRAW}$, $\overline{DISP1}$, $\overline{DISP2}$, and others

A typical waveform of the noise is shown in figure 64, and the noise voltage in the worst case is shown in figure 65.

Countermeasures

In case of having the noise, please take measures for designing the system as follows:

1) Latch $MA_{16}/RA_0-MA_{19}/RA_3$ and RA_4 by $\overline{AS}$ as shown in figure 66.
2) Latch output signals other than MA/RA by 2CLK rising or falling, or by other timings as shown in figure 67. Latch timing should be determined in consideration of the noise timing and clock timing/other timings.
3) Insert Schmitt trigger circuit in output signal lines which have little noise as shown in figure 68. For taking this measure instead of former measures, please check the condition of each output signal line, such as noise level and pulse width.

Following are measures to reduce the discharge into the internal GND line and the noise level.

1) Reduce the load capacitance on MAD bus. If it is large, bus buffer is necessary.
2) Keep supply voltage (V_{CC}) as low as possible.
3) Insert damping resistors into MAD_0-MAD_{15} as shown in figure 24. The damping resistors are 50-100Ω, and should be located as near pins as possible. Figure 69 shows the effect of the damping resistors. This is also an measure against ringing problem (figure 23).

Relational Notes

Rush current occurs at switching of output signals.

Level of this transient depends on load capacitance of external circuit, clock frequency, impedance of V_{CC} and GND, etc. It is required to design the system in consideration of measures against noise caused by the transient.

(1) Notes on power supply circuit
Keep impedance of V_{CC} and GND as little as possible to minimize voltage change as follows.
   1) Use a multilayer board (more than 4 layers with internal V_{CC} and GND).
   2) Keep V_{CC} and GND lines wide for a doublelayer board.
   3) Insert bypass capacitors as shown in figure 25.
   4) Keep no voltage difference between each V_{CC} pin, and between each GND pin. (Remark GND voltage for a doublelayer board)

(2) Notes on buses
Transient occurs at switching of MAD bus and/or host system data bus (about hundreds mA). Followings are measures to reduce the transient.
   1) Minimize the load capacitance on MAD_0-MAD_{15} and D_0-D_{15} by reducing fan out, inserting bus buffers, or shortening the buses.
   2) Insert damping resistors (50-100Ω) in buses near LSI pins as shown in figure 24.

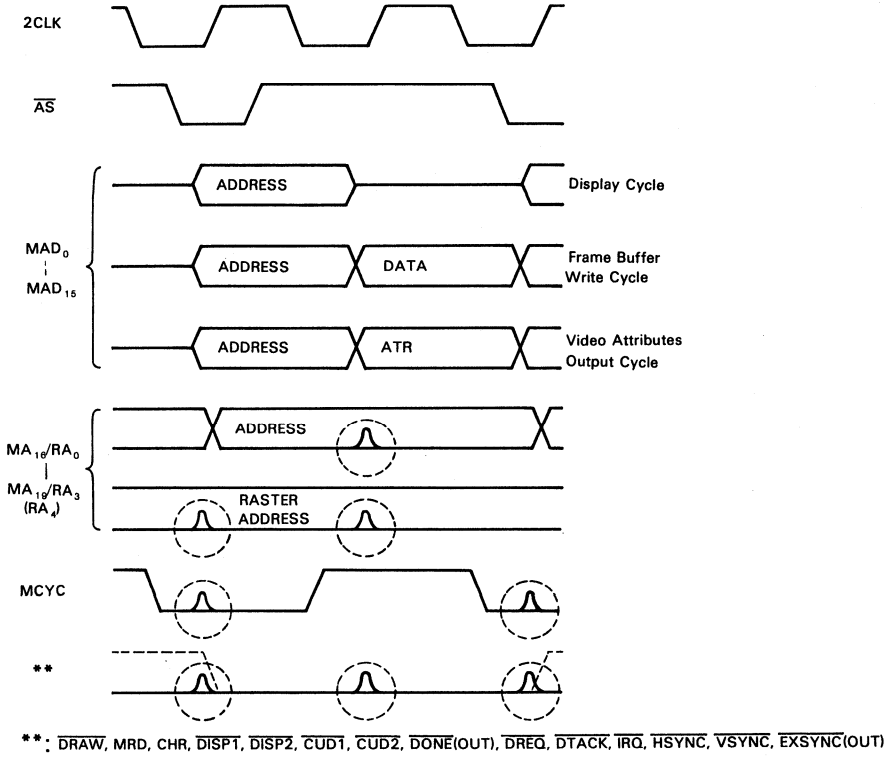


Figure 61 Noise on the "Low" Leveled Output Signals Caused by MAD Change

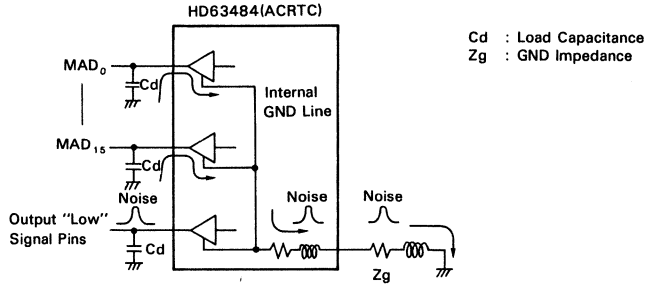


Figure 62 Internal Circuit

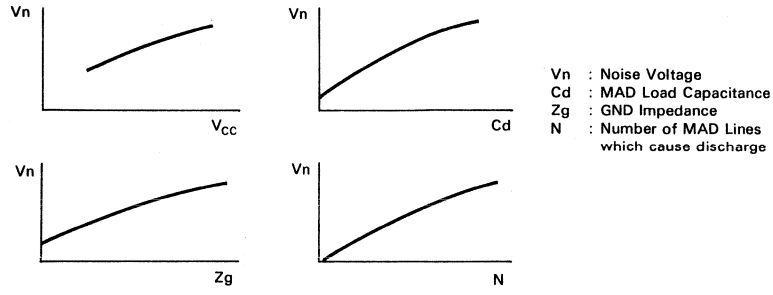


Figure 63 Relations Between the Noise Voltage and Parameter

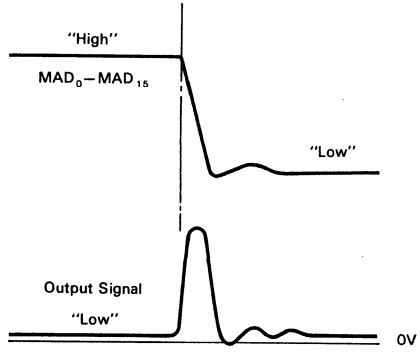


Figure 64 Typical Waveform of the Noise

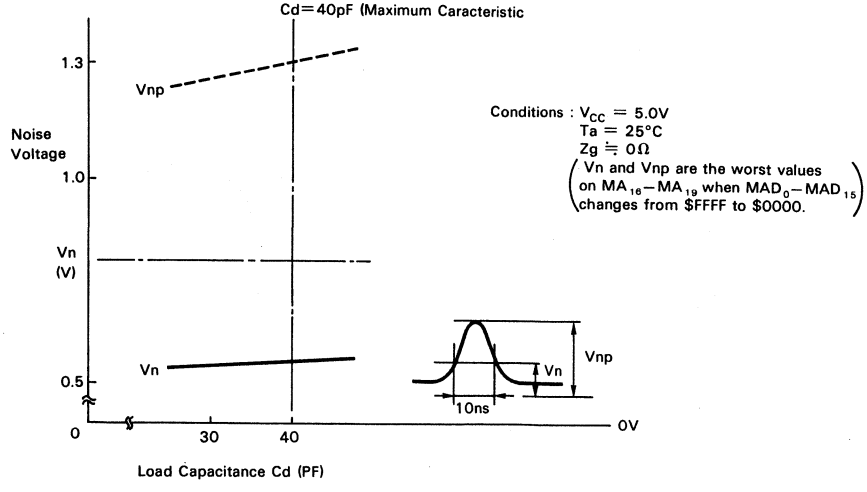


Figure 65 Relation Between the Noise and MAD Load Capacitance

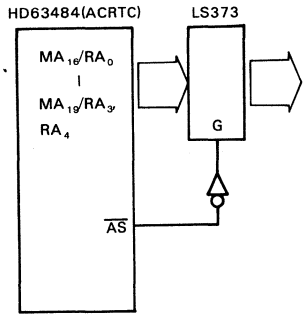


Figure 66 Countermeasure ①

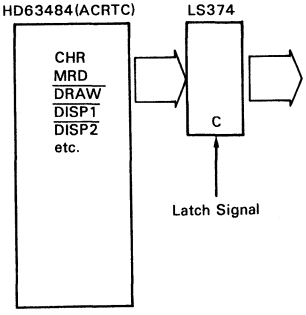


Figure 67 Countermeasure ②

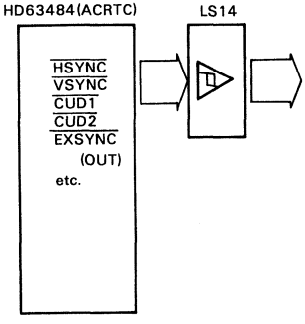


Figure 68 Countermeasure ③

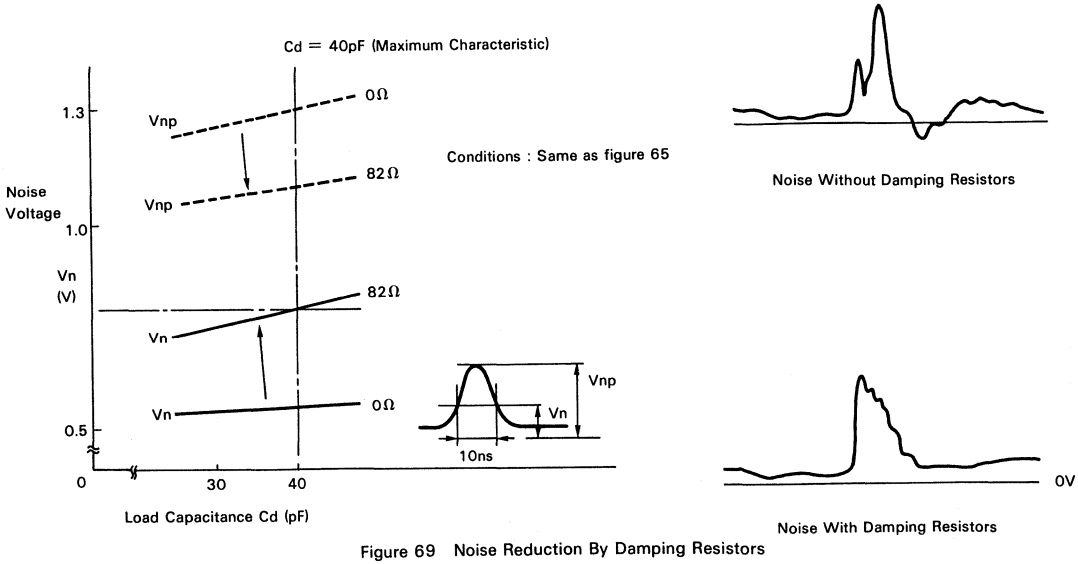


Figure 69 Noise Reduction By Damping Resistors

HD63485

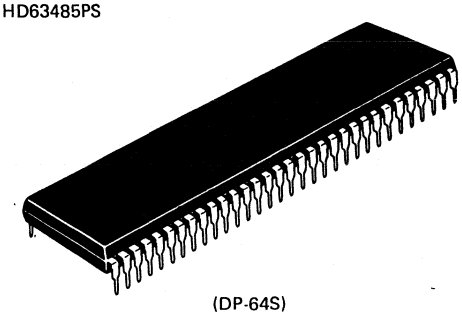
GMIC (Graphic Memory Interface Controller)

—ADVANCE INFORMATION—

The HD63485 (GMIC) is a peripheral LSI in ACRTC (Advanced CRT Controller: HD63484) family, and controls memory access for graphic display and drawing. The GMIC has built-in bus drivers and DRAM interface circuit, and can directly interface between ACRTC and DRAM without requiring additional external circuits. Main functions of the GMIC are address latching, row/column address multiplexing, generations of $\overline{RAS}$, $\overline{CAS}$, $\overline{WE}$ and 2CLK.

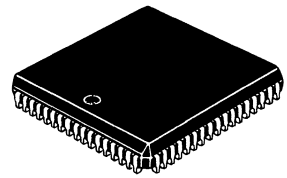
Hi-Bi CMOS process realizes the high speed memory access with low power dissipation.

HD63485PS



(DP-64S)

HD63485CP



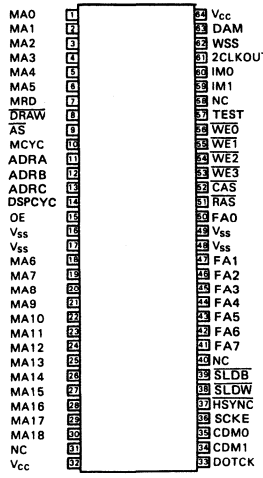
(CP-68)

■ FEATURES

- Capable of direct driving frame buffers (I_{OL} = 24mA max.)
- Generation of DRAM interface signals (Row/column address, $\overline{RAS}$, $\overline{CAS}$, $\overline{WE}$ etc.)
- Direct interface with the ACRTC
- Generation of shift clock for horizontal zooming
- Generation of load signal for horizontal smooth scrolling
- Programmable address increment mode
- Generation of 2CLK signal for ACRTC
- TTL compatible input/output
- +5V single power supply

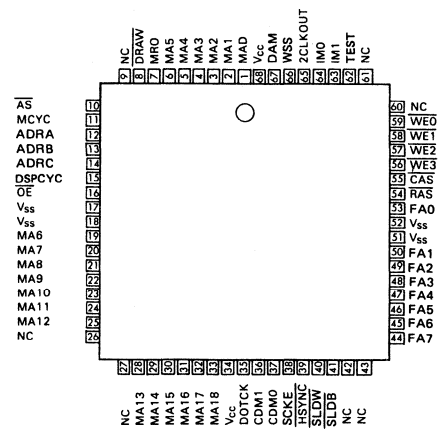
■ PIN ARRANGEMENT

- HD63485PS



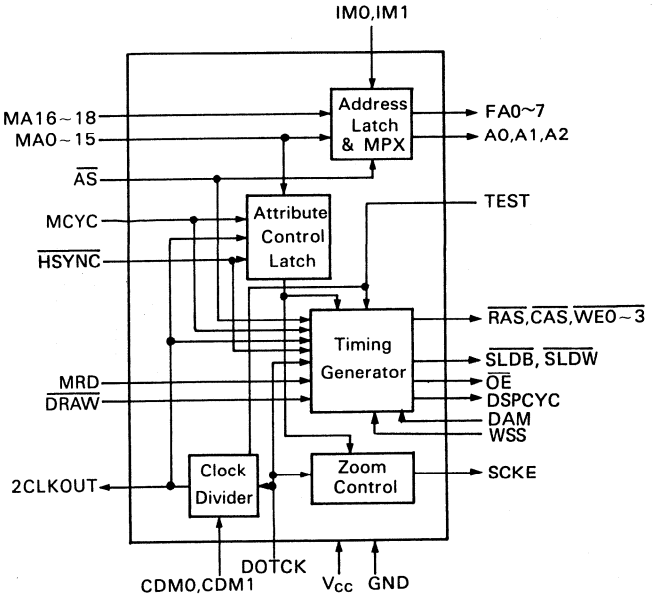
(Top View)

- HD63485CP

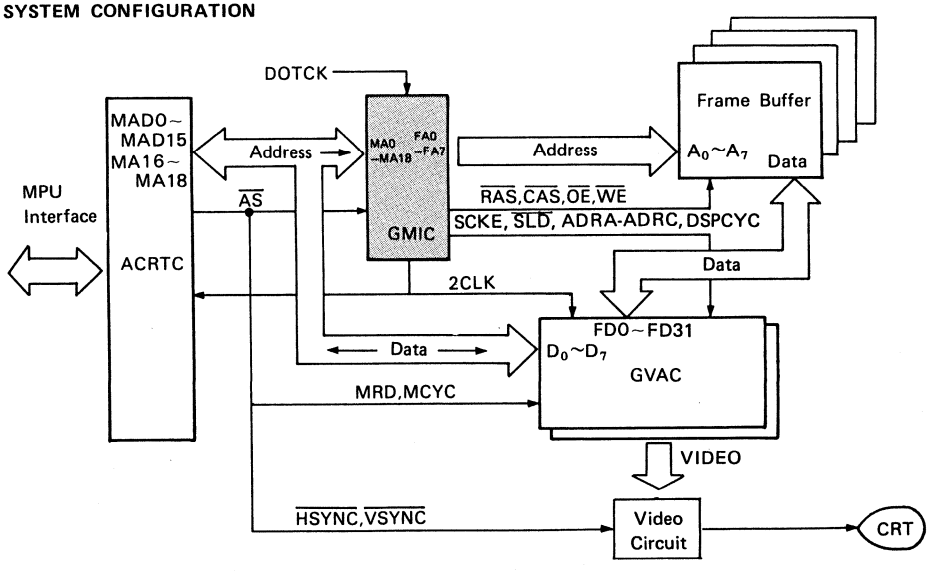


(Top View)

■ BLOCK DIAGRAM



■ SYSTEM CONFIGURATION



HD63486

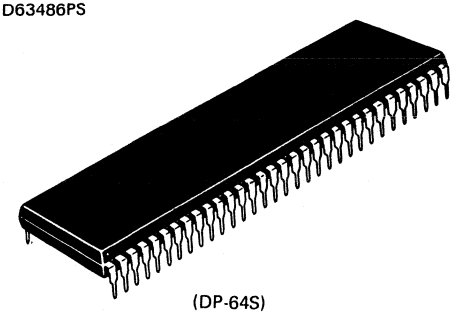
GVAC (Graphic Video Attribute Controller)

—ADVANCE INFORMATION—

The HD63486 (GVAC) is a peripheral LSI in ACRTC (Advanced CRT Controller: HD63484) family and converts the data from the frame buffer to serial video signals. Built-in shift register and the built-in peripheral video controlling circuits required for graphic display can generate high-speed video signals. The bit length of the shift register is programmable for color graphics and gray level graphic applications. The applications coverage can be extended by driving plural GVACs in parallel. The GVAC supports horizontal smooth scroll and horizontal zoom under control signals from the HD63485 (GMIC).

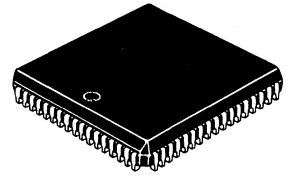
Hi-Bi CMOS process realizes high speed video signal generation with low power dissipation.

HD63486PS



(DP-64S)

HD63486CP



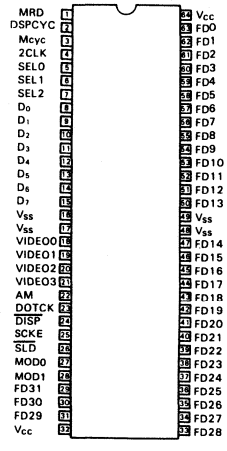
(CP-68)

■ FEATURES

- Built-in shift registers for video signals (Programmable bit length)
  Selectable: Four- 8-bit shift registers / Two 16-bit shift registers / One 32-bit shift register
- High speed video signal generation (dot-rate: 64 MHz max)
- Capable of driving plural GVACs in parallel
- Built-in bidirectional data bus buffer for interfacing with the frame buffer
- Zoom and horizontal smooth scroll functions (Under the control signal from GMIC or equivalent)
- Direct interface with the ACRTC
- TTL compatible input/output
- +5V single power supply

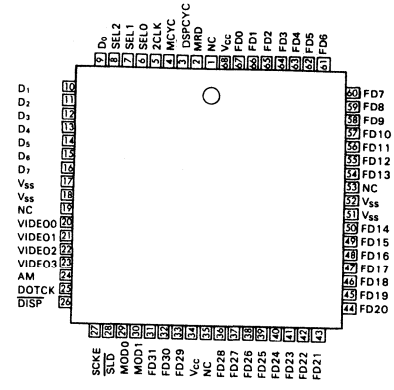
■ PIN ARRANGEMENT

● HD63486PS



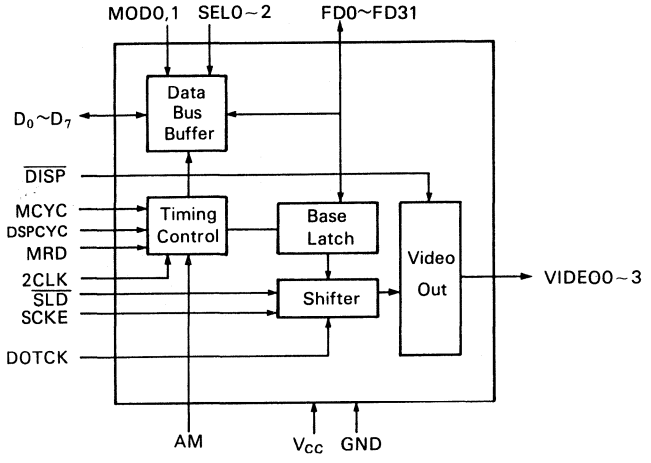
(Top View)

● HD63486CP

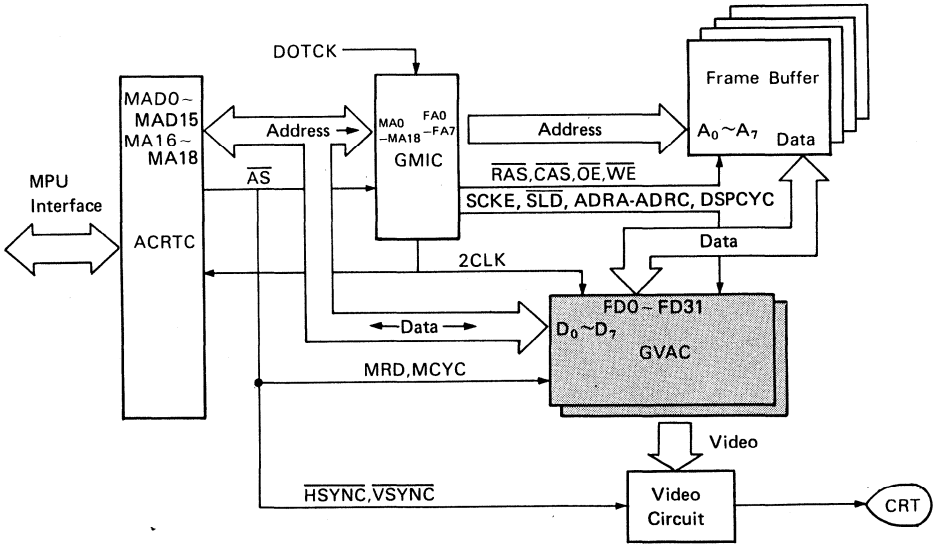


(Top View)

■ BLOCK DIAGRAM



■ SYSTEM CONFIGURATION



HD68562

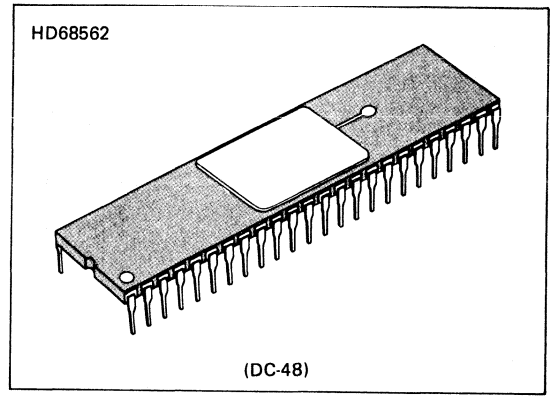
DUSCC (Dual Universal Serial Communications Controller)

—ADVANCE INFORMATION—

The HD68562 Dual Universal Serial Communications Controller (DUSCC) transforms parallel data which is transferred from central processing unit into serial data. It is a single chip MOS-LSI communications device designed to be a foundation of universal high-performance data-communication subsystems, particularly for the 68000 family microprocessors.

The DUSCC provides two independent, multi-protocol, full duplex receiver/transmitter channels in a single package. Since the DUSCC supports a wide range of protocols, it handles data communications with the minimum intervention, usually just a few commands from its host processor. The controller's data rate is maximum 4M bits/s which meets the requirement of the presently most powerful systems. A high-speed, high-performance communication system is realized with minimum external logic at low cost through a variety of functions provided by the chip: 16-bit multi-function counter/timer, a digital phase locked loop (DPLL), a parity/CRC generator and checker, and baud rate generator.

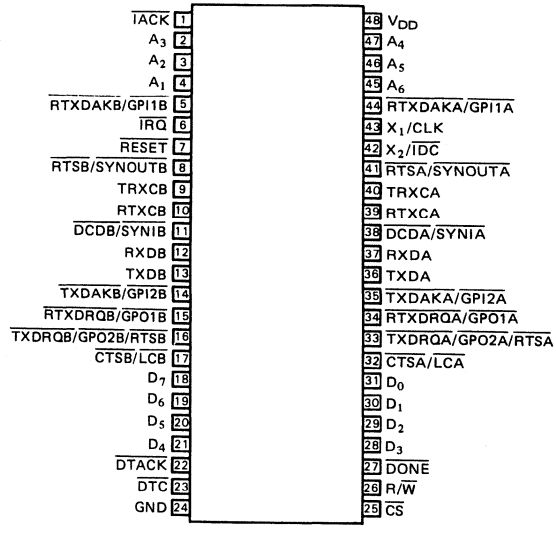
The DUSCC is useful for communication between host computer and terminals, electric mail, VIDEOTEX, local area network (LAN), communications network among personal computers, etc.



■ FEATURES

- Channel data rate: 4M bps maximum
- Receiver/Transmitter: Two channels, dual full-duplex synchronous/asynchronous
- Multi-protocol operation: BOP (Bit Oriented Protocol) BCP (Byte Controlled Protocol) COP (Character Oriented Protocol) ASYNC (Asynchronous)
- High data transfer efficiency: Four-character receiver/transmitter FIFOs
- Parity and FCS (Frame Check Sequence): VRC, LRC-8, CRC-16, CRC-CCITT
- Programmable data encoding/decoding: NRZ, NRZI, FM0, FM1, Manchester
- DMA interface: Compatible with Hitachi HD68450/HD-63450 DMAC and other DMA controllers
- Multi-function programmable 16-bit C/T: Baud rate generator, etc.
- Clock oscillator: On-chip oscillator for crystal
- Power supply: Single +5V

■ PIN ARRANGEMENT



■ MAJOR FUNCTIONS OF DUSCC

Item	FUNCTION
Maximum operating frequency	4 MHz
Maximum data transfer rate	4 Mbits/s
Data length	5-8 bits
Bus interface	Compatible with HD68000 (8 bits bus)
FIFO	4 bytes for each receiver/transmitter
Number of channels	2 channels
Error check	Parity, framing, over run, under run, FCS
Channel mode	Half-duplex, full-duplex, auto-echo, local loopback
Data transfer mode	Polled, interrupt, DMA, wait
Protocol operation	ASYNC : 5-8 bits plus optional parity COP : BISYNC, X.21 BCP : DDCMP BOP : HDLC/ADCCP, SDLC, SDLC Loop, Link Level, X.75 Link Level
Baud rate generator	Built-in
Selection of baud rate	(1) 16 fixed rates: 50 to 38.4K baud. (2) Optional baud rate by timer.
Encoding/Decoding	NRZ, NRZI, FM0, FM1, Manchester
Digital phase locked loop	Built-in
DMA interface	Compatible with HD68450/HD63450 Half or full duplex operation Single or dual address data transfer
Interrupt capabilities	(1) Daisy chain option (2) Vector output (fixed or modified by status) (3) Maskable interrupt conditions (4) Programmable internal priorities
Model control	RTS, CTS, DCD Four general purpose I/O pins per channel
16-bit counter timer	Built-in
Oscillator	Built-in
Package	Ceramic DIP 48-pin
Power supply	5V ± 10% Ta = 0 to 70°C
Power dissipation	Typ. 1 W

■ INTERNAL BLOCK DIAGRAM

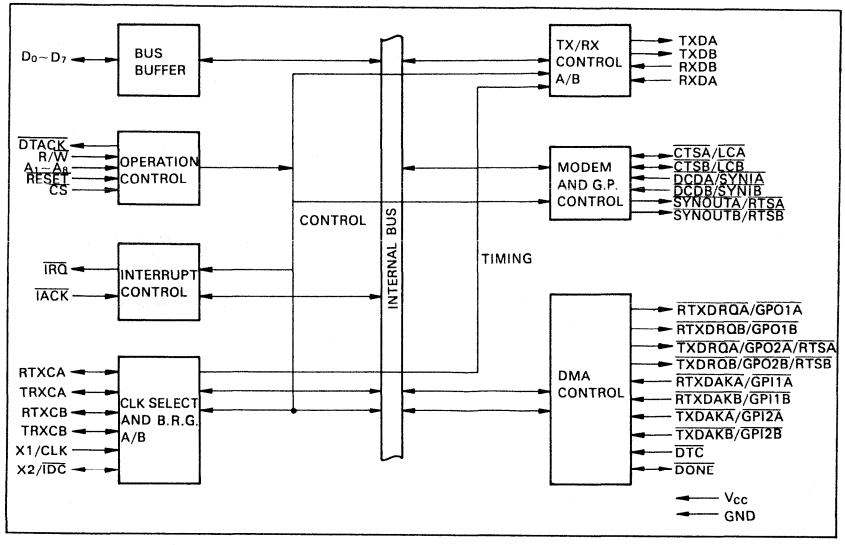


Figure 1 DUSCC Block Diagram

■ SYSTEM CONFIGURATION

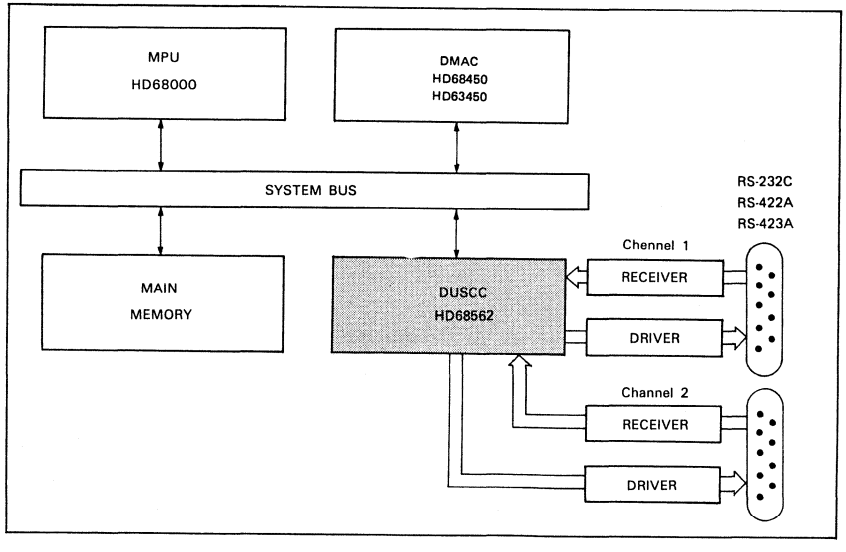


Figure 2 System Configuration Example

HD63645F

LCTC (Liquid Crystal Display Timing Controller)

–ADVANCE INFORMATION–

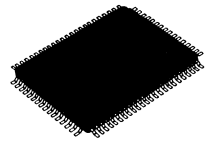
The HD 63645F (LCTC) is a controller for the large size dot matrix liquid crystal graphic display. It controls the screen of max 640 x 512 dots. Either Single or Dual screen configuration is selectable. Up to 1/512 display duty ratio can be specified for the screen.

The programming method of internal registers and display buffer is based on the HD6845 (CRTC), which makes possible LCD display system that is software-compatible with HD6845 (CRTC) system.

The bit-mapped display or the character display is selectable.

CPU's direct access to the display memory enables high-speed drawing.

HD63645F

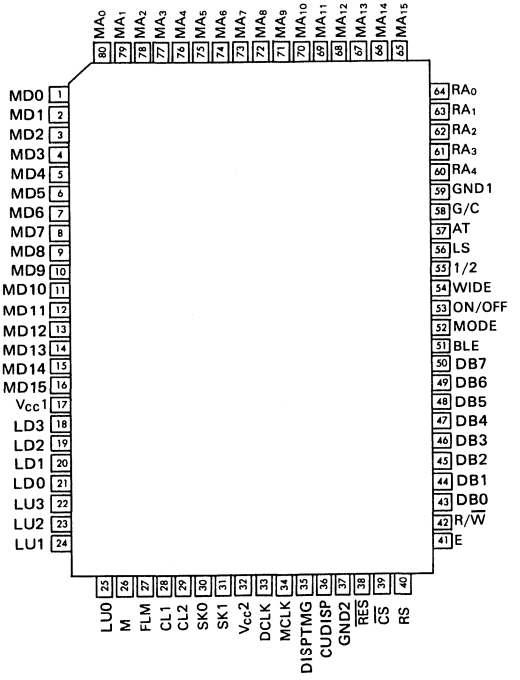


(FP-80)

■ FEATURES

- Software compatible with CRTC (HD6845)
- Programmable screen size: Max. 640 x 512 dots
- Single/Dual selectable screen configuration
- Programmable display duty ratio: Max. 1/512 duty
- Programmable character font: 1 ~ 32[H] x 8[W] dots
- Attribute functions:
  (Applicable only to the character display mode. Unit: 1 character) reverse, blink, all Black/White
  (Underline and ruled line is also possible)
- Cursor: ON/OFF, blink, programmable cursor height
- Cursor: Vertical smooth scroll,
  horizontal scroll (by characters)
- Display memory: 64 bytes x 2 (8- or 16-bit access).
- Refresh address output for dynamic RAM.
- LCD driver interface: 4 bits of 8 bits Parallel
- LCD drive LSI: HD61104/HD61105
- CPU Interface: 6800 family
- CMOS Process
- Power supply: single +5V ± 10%
- 80-pin Flat Plastic Package

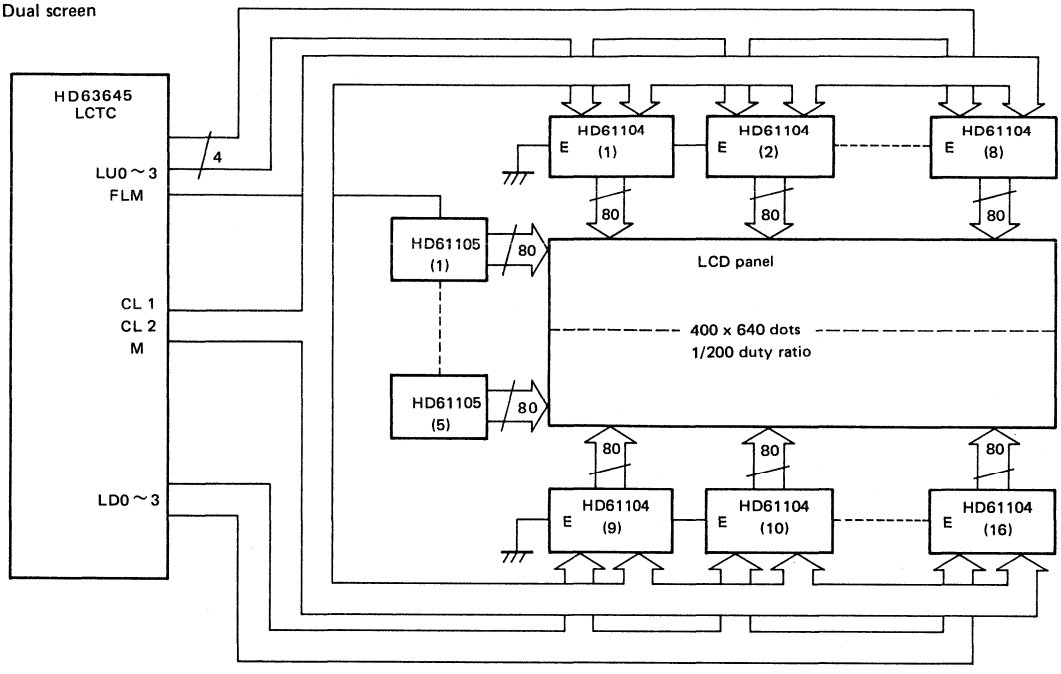
■ PIN ARRANGEMENT



(Top View)

■ SYSTEM CONFIGURATION

● Dual screen



● Single screen

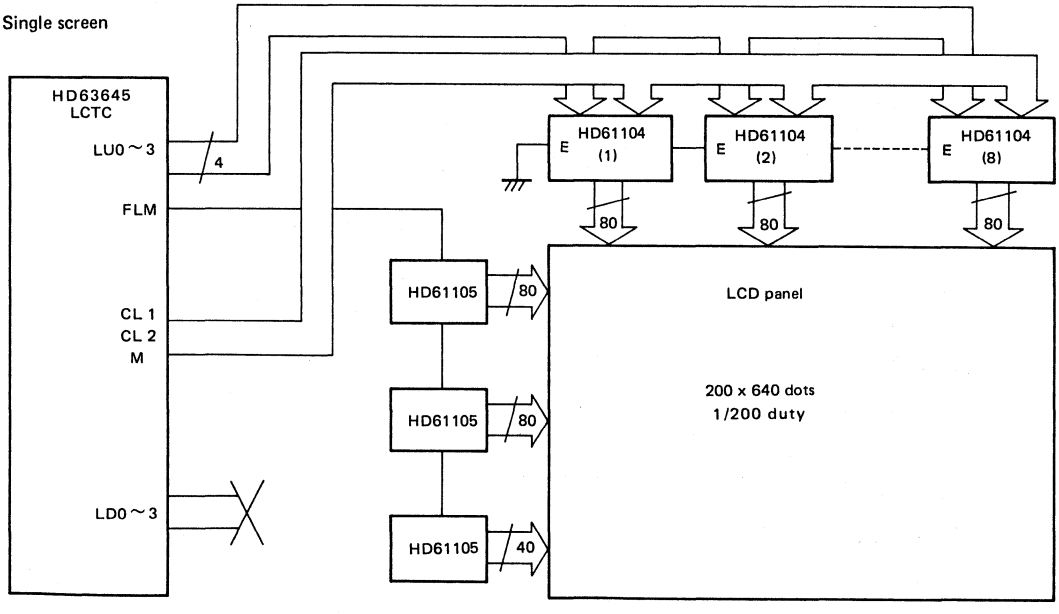


Figure 1 Example of Connection to LCD Drives

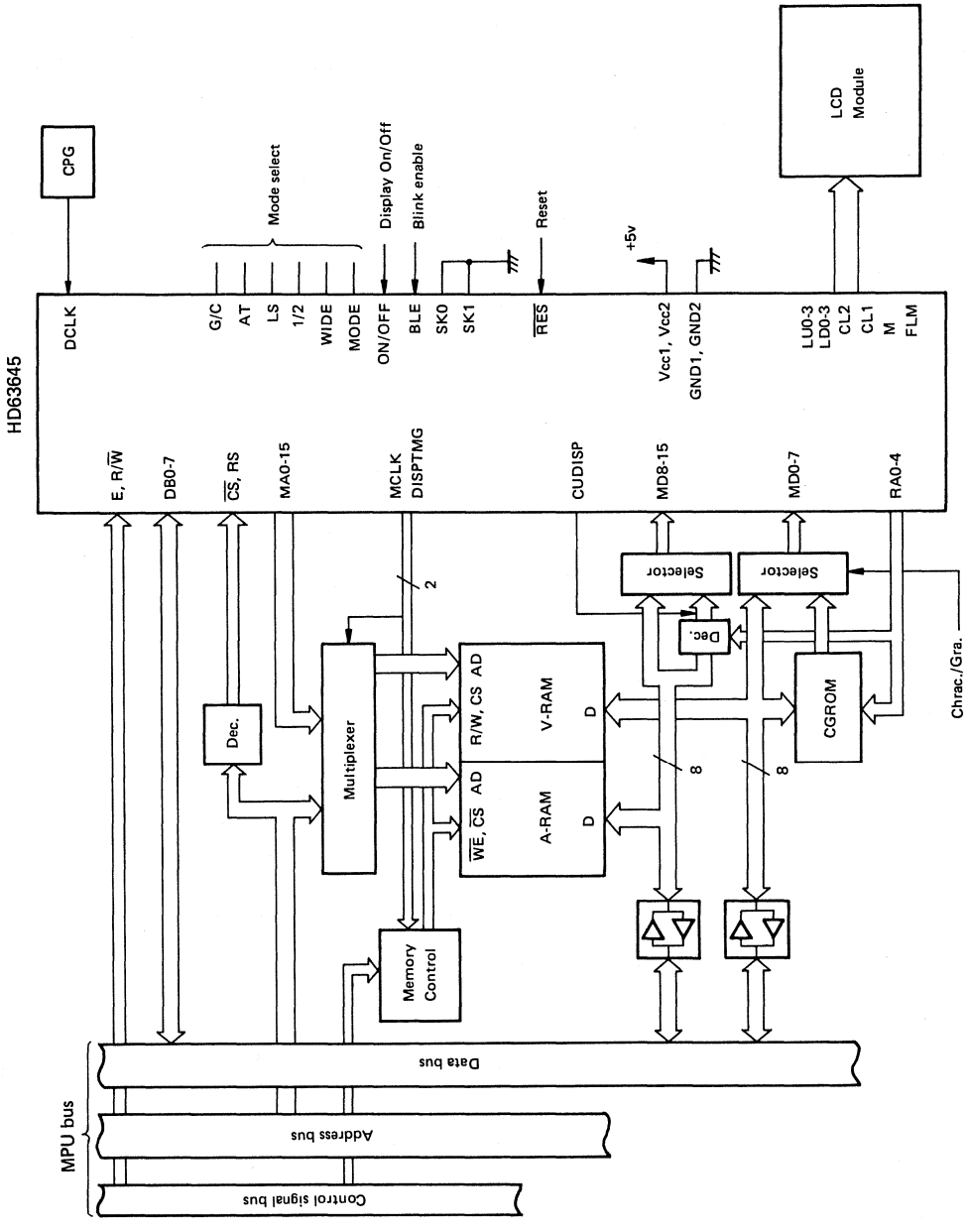


Figure 2 Example of Character/Graphic Display System

HD64941

ACI (Asynchronous Communications Interface)

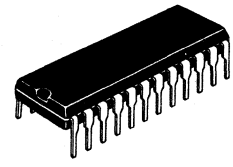
–ADVANCE INFORMATION–

The HD64941 (ACI) is a universal asynchronous data communications controller chip that interfaces directly to most 8-bit microprocessors and may be used in a polled or interrupt-driven system environment. The HD64941 accepts programmed instructions from the microprocessor while supporting asynchronous serial data communications in full- or half-duplex mode.

The HD64941 serializes parallel data characters received from the microprocessor for transmission. Simultaneously, it can receive serial data and convert it into parallel data characters for input to the microcomputer.

The HD64941 contains a baud rate generator which can be programmed to either accept an external clock or to generate internal transmit or receive clocks. Sixteen different baud rates can be selected under program control when operating in the internal clock mode.

HD64941

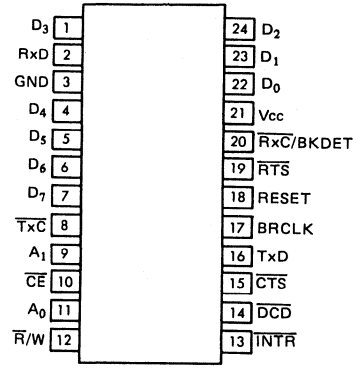


(DP-24N)

■ **FEATURES**

- 5- to 8-bit characters plus parity
- 1, 1½ or 2 stop bits transmitted
- Odd, even or no parity
- Parity, overrun and framing error delection
- Line break detection and generation
- False start bit detection
- Automatic serial echo mode (echoplex)
- Local or remote maintenance loopback mode
- Baud rate:
  - DC to 1M bps (1 x clock)
  - DC to 62.5k bps (16 x clock)
  - DC to 15.625k bps (64 x clock)
- Internal or external baud rate clock
- 16 internal rates
- Double-buffered transmitter and receiver
- Single +5V power supply

■ **PIN ARRANGEMENT**



(Top View)

■ BLOCK DIAGRAM

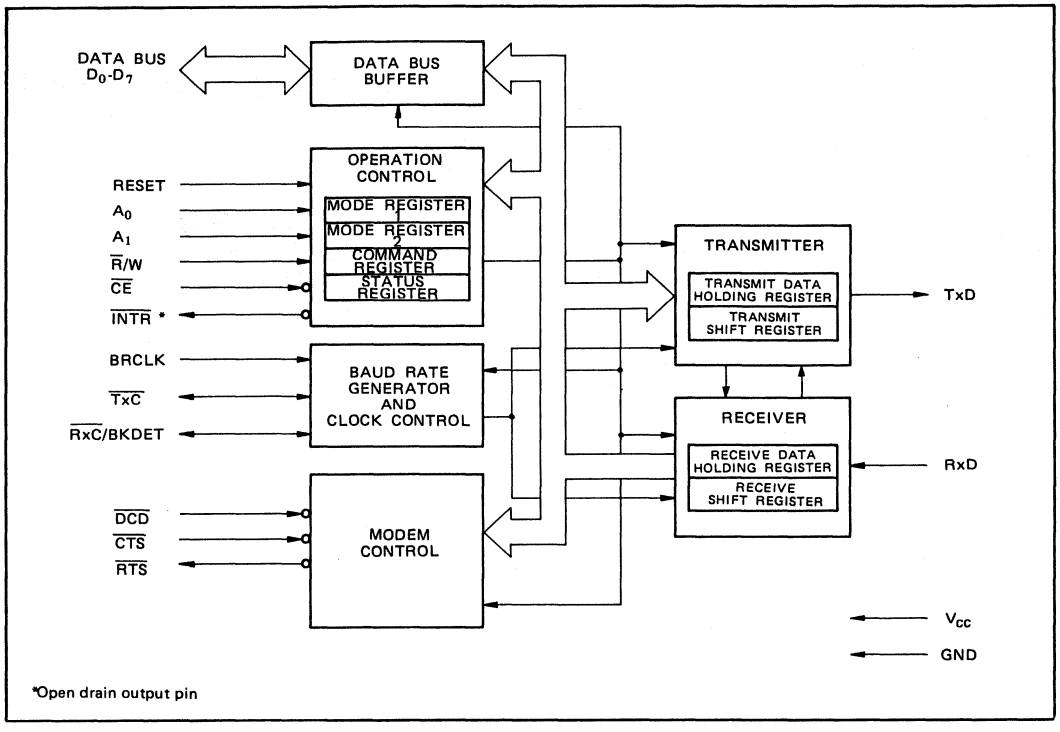


Figure 1 ACI Block Diagram

■ APPLICATIONS

- Intelligent terminals
- Network processors
- Front-end processors
- Remote data concentrators
- Serial peripherals

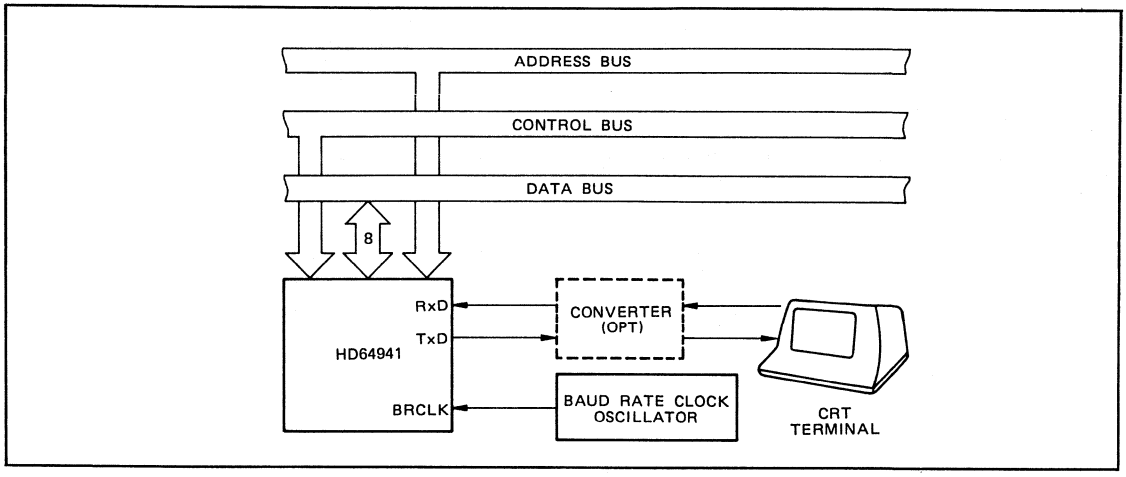


Figure 2 Asynchronous Interface to CRT Terminal

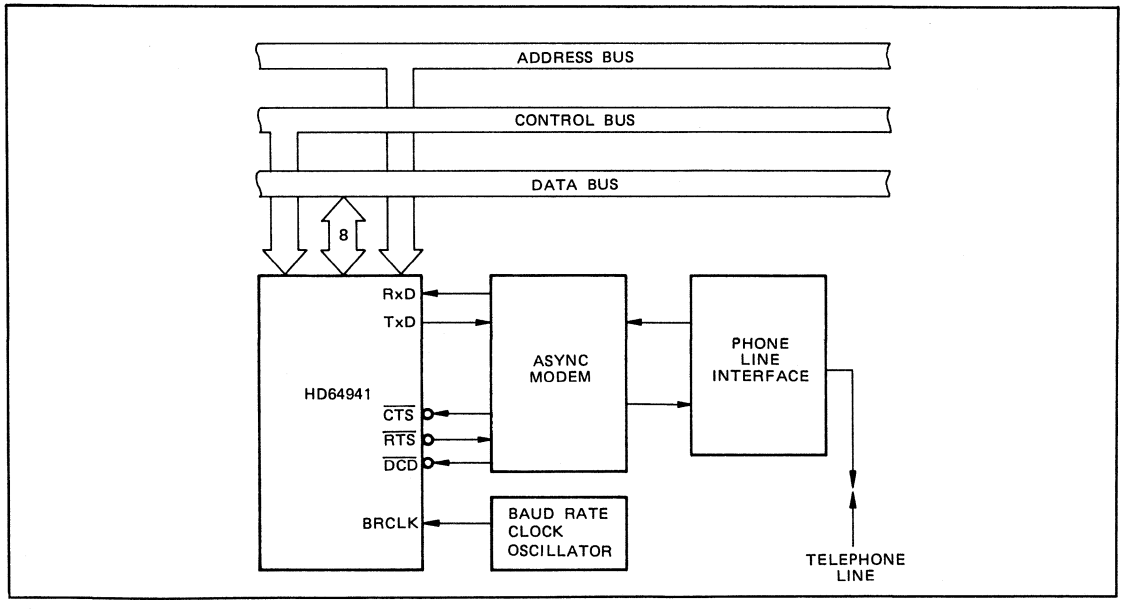


Figure 3 Asynchronous Interface to Telephone Lines

HD146818

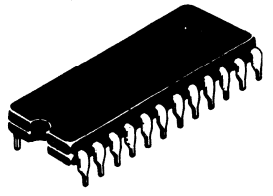
RTC (Real Time Clock Plus RAM)

The HD146818 is a HMCS6800 peripheral CMOS device which combines three unique features: a complete time-of-day clock with alarm and one hundred calendar, a programmable periodic interrupt and square-wave generator, and 50 bytes of Low-power static RAM.

This device includes HD6801, HD6301 multiplexed bus interface circuit and 8085's multiplexed bus interface as well, so it can be directly connected to HD6801, HD6301 and 8085.

The Real-Time Clock plus RAM has two distinct uses. First, it is designed as battery powered CMOS part including all the common battery backed-up functions such as RAM, time, and calender. Secondly, the HD146818 may be used with a CMOS microprocessor to relieve the software of timekeeping workload and to extend the available RAM of an MPU such as the HD6301.

HD146818P



(DP-24)

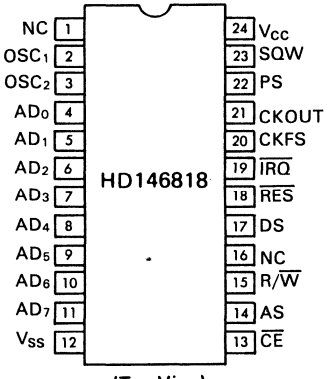
HD146818FP

(FP-24)

■ FEATURES

- Time-of-Day Clock and Calendar
  - Counts Seconds, Minutes, and Hours of the Day
  - Counts Days of Week, Date, Month, and Year
- Binary or BCD Representation of Time, Calendar, and Alarm
- 12- or 24 Hour Clock with AM and PM in 12-Hour Mode
- Automatic End of Month Recognition
- Automatic Leap Year Compensation
- Interfaced with Software as 64 RAM Locations
  - 14 Bytes of Clock and Control Register
  - 50 Bytes of General Purpose RAM
- Three Interrupt are Separately Software Maskable and Testable
  - Time-of-Day Alarm, Once-per-Second to Once-per-Day
  - Periodic Rates from 30.5μs to 500ms
  - End-of-Clock Update Cycle
- Programmable Square-Wave Output Signal
- Three Time Base Input Options
  - 4.194304 MHz
  - 1.048576 MHz
  - 32.768 kHz
- Clock Output May be used as Microprocessor Clock Input
  - At Time Base Frequency ÷4 or ÷1
- Multiplexed Bus Interface Circuit of HD6801, HD6301 and 8085
- Low-Power, High-Speed, High-Density CMOS
- Battery Backed-up Operation
- Motorola MC146818 Compatible

■ PIN ARRANGEMENT



(Top View)

■ ABSOLUTE MAXIMUM RATINGS

Item	Symbol	Value	Unit
Supply Voltage	V_{CC} *	-0.3 ~ +7.0	V
Input Voltage	V_{in} *	-0.3 ~ +7.0	V
Operating Temperature	T_{opr}	0 ~ +70	°C
Storage Temperature	T_{stg}	-55 ~ +150	°C

* With respect to V_{SS} (SYSTEM GND)

(NOTE) Permanent LSI damage may occur if maximum rating are exceeded. Normal operation should be under recomended operating condition. If these conditions are exceeded, it could affect reliability of LSI.

■ RECOMENDED OPERATING CONDITIONS

Item	Symbol	min	typ	max	Unit
Supply Voltage	V_{CC} *	4.5	5.0	5.25	V
Input Voltage	V_{IL} *	-0.3	–	0.7	V
	V_{IH} *	V_{CC}-1.0	–	V_{CC}	V
Operating Temperature	T_{opr}	0	25	70	°C

* With respect to V_{SS} (SYSTEM GND)
(NOTE) Refer to Battery Backed-up Electrical characteristics.

■ ELECTRICAL CHARACTERISTICS

● DC CHARACTERISTICS (V_{CC} = 4.5 ~ 5.25V, V_{SS} = 0V, Ta = 0 ~ +70°C, unless otherwise noted.)

Item		Symbol	Test Condition	min	typ	max	Unit
Input "High" Voltage	AD_0~AD_7, $\overline{CE}$, AS, R/$\overline{W}$, DS, CKFS, PS	V_{IH}		V_{CC}-2.0	–	V_{CC}	V
	$\overline{RES}$			V_{CC}-1.0	–	V_{CC}	
	OSC_1			V_{CC}-1.0	–	V_{CC}	
Input "Low" Voltage	AD_0~AD_7, $\overline{CE}$, AS, R/$\overline{W}$, DS, CKFS, PS	V_{IL}		-0.3	–	0.7	V
	$\overline{RES}$			-0.3	–	0.8	
	OSC_1			-0.3	–	0.8	
Input Leakage Current	OSC_1, $\overline{CE}$, AS, R/$\overline{W}$, DS, $\overline{RES}$, CKFS, PS	$\lvert I_{in}\rvert$		–	–	2.5	μA
Three-state (off state) Input Current	AD_0~AD_7	$\lvert I_{TSI}\rvert$		–	–	10	μA
Output Leakage Current	$\overline{IRQ}$	I_{LOH}		–	–	10	μA
Output "High" Voltage	AD_0~AD_7	V_{OH}	I_{OH} = -1.6 mA	4.1	–	–	V
	SQW, CKOUT						
	AD_0~AD_7		I_{OH} <-10 μA	V_{CC}-0.1	–	–	V
	SQW, CKOUT						
Output "Low" Voltage	AD_0~AD_7	V_{OL}	I_{OL} = 1.6 mA	–	–	0.5	V
	CKOUT		I_{OL} = 1.6 mA				
	$\overline{IRQ}$, SQW		I_{OL} = 1.6 mA				
Input Capacitance	AD_0~AD_7	C_{in}	V_{in} = 0V, Ta = 25°C, f = 1 MHz	–	–	12.5	pF
	All inputs except AD_0~AD_7			–	–	12.5	pF
Output Capacitance	SQW, CKOUT, $\overline{IRQ}$	C_{out}		–	–	12.5	pF
Supply Current (MPU Read/Write operating) Crystal Oscillation	f_{osc} = 4 MHz	I_{CC} *	V_{CC} = 5.0V, SQW: disable, CKOUT = f_{osc} (No Load), t_{cyc} = 1 μs, Circuit: Fig. 11, Parameter: Table 1	–	–	10	mA
	f_{osc} = 1 MHz			–	–	7	
	f_{osc} = 32 kHz			–	–	5	
Supply Current** (MPU not operating) Crystal Oscillation	f_{osc} = 4 MHz			–	–	5	mA
	f_{osc} = 1 MHz			–	–	2	
	f_{osc} = 32 kHz			–	300	500	μA
Supply Current (MPU Read/Write operating) External Clock	f_{osc} = 4 MHz	I_{CC} *	V_{CC} = 5.0V, SQW: disable, CKOUT = f_{osc} (No Load), OSC_2: open, t_{cyc} = 1 μs, Circuit: Fig. 17	–	–	10	mA
	f_{osc} = 1 MHz			–	–	7	
	f_{osc} = 32 kHz			–	–	5	
Supply Current** (MPU not operating) External Clock	f_{osc} = 4 MHz			–	–	4	mA
	f_{osc} = 1 MHz			–	–	1	
	f_{osc} = 32 kHz			–	60	100	μA

* Supply current of HD146818 is defined as the value when the time-base frequency to be used is programmed into Register A. When power is turned on, these bits are unfixed, so there is a case that current more than the above specification may flow. Please never fail to set the time-base frequency after turning on power supply.

** $V_{IH\ min} = V_{CC}$-0.2V
$V_{IL\ max} = V_{SS}$+0.2V

- **AC CHARACTERISTICS (V_{CC} = 4.5 ~ 5.25V, V_{SS} = 0V, Ta = 0 ~ +70°C, unless otherwise noted.)**

BUS TIMING

Item	Symbol	min	typ	max	Unit
Cycle Time	t_{cyc}	953	–	–	ns
Pulse Width, AS/ALE "High"	PW_{ASH}	100	–	–	ns
AS Rise Time	t_{ASr}	–	–	30	ns
AS Fall Time	t_{ASf}	–	–	30	ns
Delay Time DS/E to AS/ALE Rise	t_{ASD}	40	–	–	ns
DS Rise Time	t_{DSr}	–	–	30	ns
DS Fall Time	t_{DSf}	–	–	30	ns
Pulse Width, DS/E Low or $\overline{RD}/\overline{WR}$ "High"	PW_{DSH}	325	–	–	ns
Pulse Width, DS/E High or $\overline{RD}/\overline{WR}$ "Low"	PW_{DSL}	300	–	–	ns
Delay Time, AS/ALE to DS/E Rise	t_{ASDS}	90	–	–	ns
Address Setup Time (R/$\overline{W}$)	t_{AS1}	15	–	–	ns
Address Setup Time ($\overline{CE}$)	t_{AS2}	55	–	–	ns
Address Hold Time (R/$\overline{W}$, $\overline{CE}$)	t_{AH}	10	–	–	ns
Muxed Address Valid Time to AS/ALE Fall	t_{ASL}	50	–	–	ns
Muxed Address Hold Time	t_{AHL}	20	–	–	ns
Peripheral Data Setup Time	t_{DSW}	195	–	–	ns
Write Data Hold Time	t_{DHW}	0	–	–	ns
Peripheral Output Data Delay Time From DS/E or $\overline{RD}$	t_{DDR}	–	–	220	ns
Read Data Hold Time	t_{DHR}	10	–	–	ns
Input Rise and Fall Time	t_r, t_f	–	–	30	ns

CONTROL SIGNAL TIMING

Item		Symbol	min	typ	max	Unit
Oscillator Startup	1 MHz, 4 MHz	t_{RC}	–	–	100	ms
	32 kHz		–	–	1000	
Reset Pulse Width		t_{RWL}	5.0	–	–	μs
Reset Delay Time		t_{RLH}	5.0	–	–	μs
Power Sense Pulse Width		t_{PWL}	5.0	–	–	μs
Power Sense Delay Time		t_{PLH}	5.0	–	–	μs
$\overline{IRQ}$ Release from DS		t_{IRDS}	–	–	2.0	μs
$\overline{IRQ}$ Release from $\overline{RES}$		t_{IRR}	–	–	2.0	μs
VRT Bit Delay		t_{VRTD}	–	–	2.0	μs

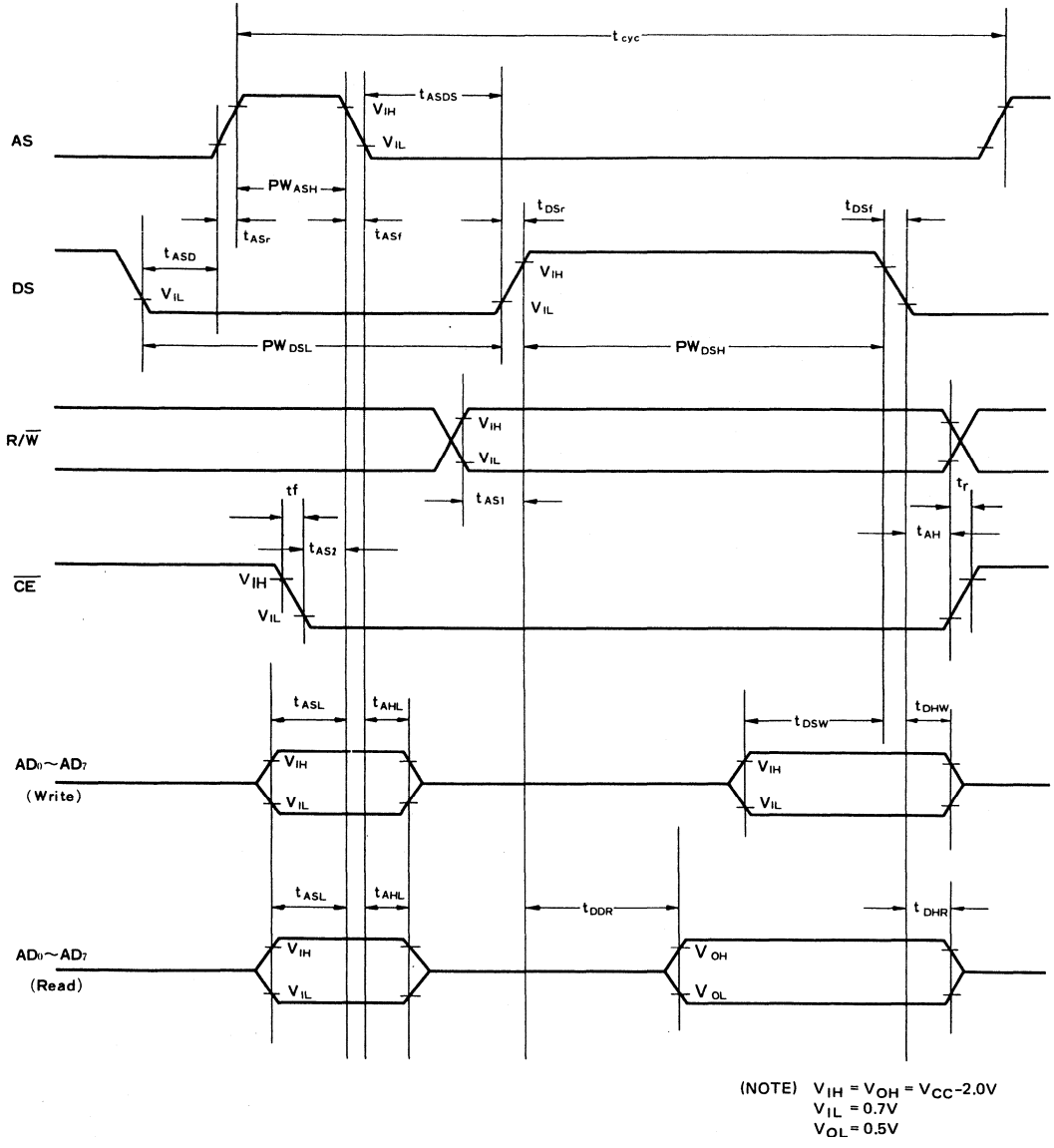
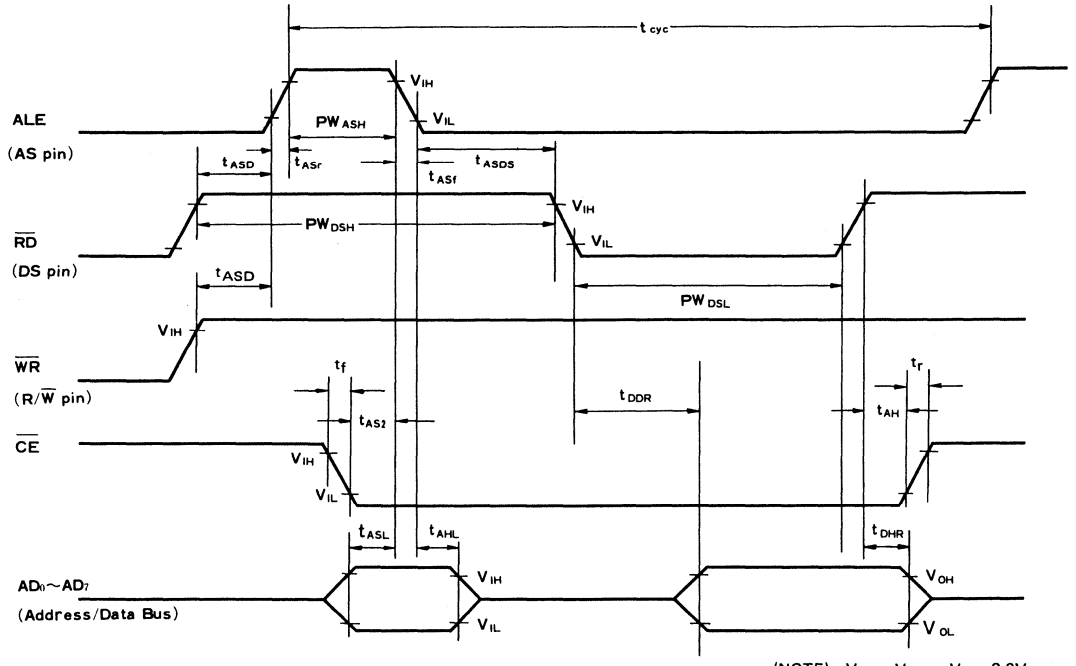
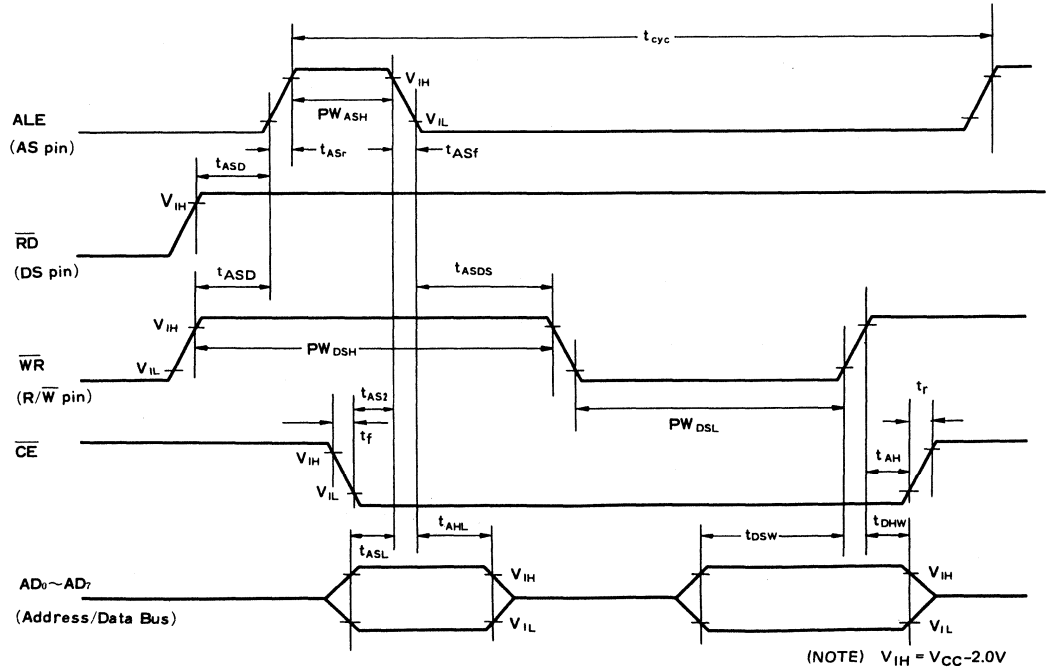


Figure 1 Bus Read, Write Timing (6801 Family)



(NOTE) $V_{IH} = V_{OH} = V_{CC}-2.0V$
$V_{IL} = 0.7V, V_{OL} = 0.5V$

Figure 2 Read Timing (8085 Family)



(NOTE) $V_{IH} = V_{CC}-2.0V$
$V_{IL} = 0.7V$

Figure 3 Write Timing (8085 Family)

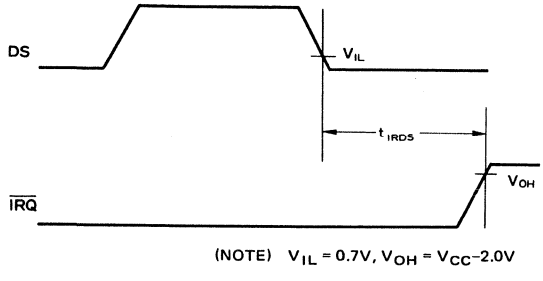


Figure 4 $\overline{IRQ}$ Release Delay (from DS)

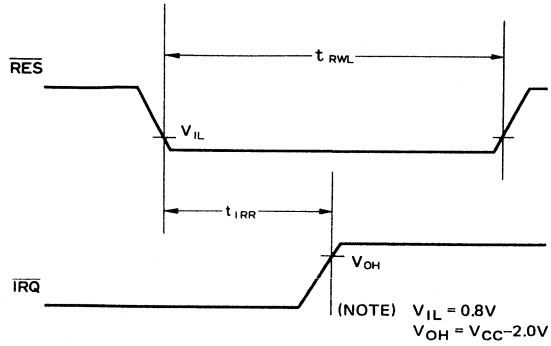
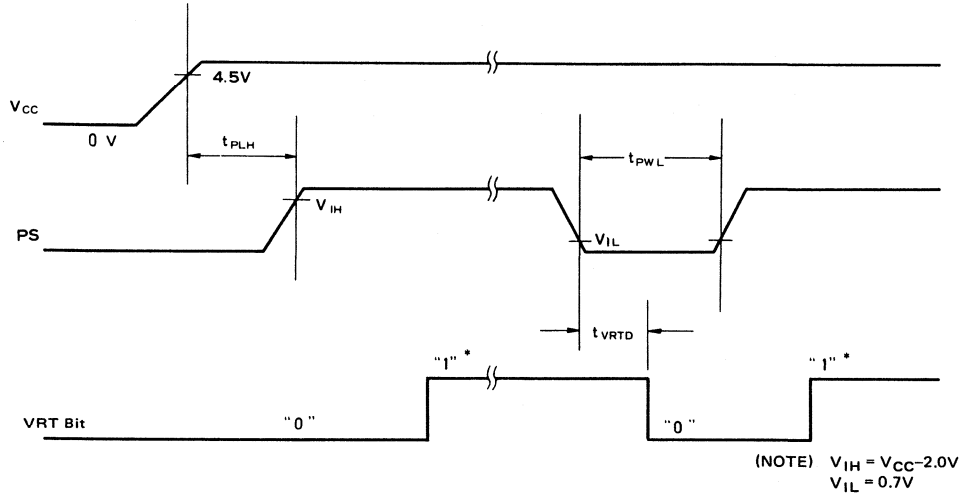


Figure 5 $\overline{IRQ}$ Release Delay (from $\overline{RES}$)



* The VRT bit is set to a "1" by reading control register #D. There is no additional way to clear the VRT bit.

Figure 6 VRT Bit Clear Timing

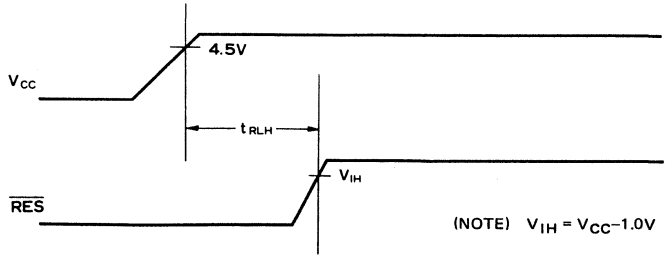


Figure 7 $\overline{RES}$ Release Delay

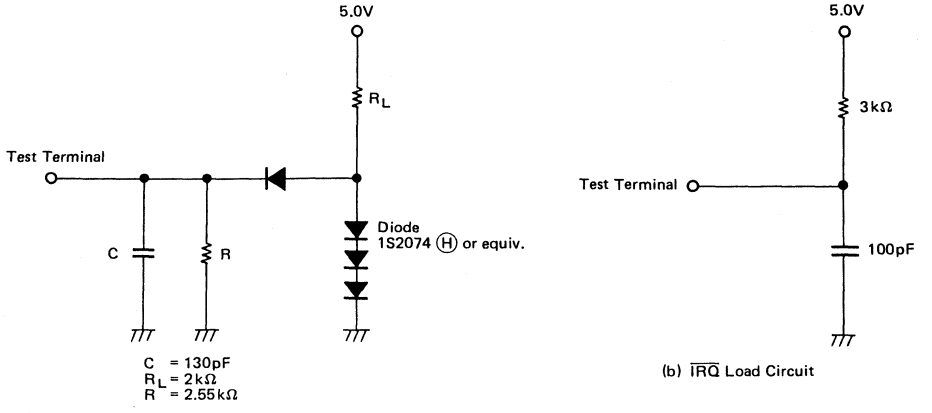


Figure 8 Test Load

■ BATTERY BACKED-UP OPERATION

• DEFINITION OF BATTERY BACKED-UP OPERATION

Active functions
(1) Clock function
(2) Retention of RAM data
(3) RES, IRQ, CKFS, CKOUT, PS, SQW functions

Inactive functions
(1) Data bus read/write operation

• BATTERY BACKED-UP ELECTRICAL CHARACTERISTICS (V_{SS} = 0V, T_a = 0 ~ +70°C, unless otherwise noted.)

Item		Symbol	Test Condition		min	typ	max	Unit
Supply Voltage		V_{CCL}			2.7	–	4.5	V
Supply Current	Crystal Oscillation	I_{CCL}*	V_{CCL} = 3.0V, SQW : disable, CKOUT: fosc (No load)	4MHz	–	–	600	μA
				1MHz	–	–	350	μA
				32kHz	–	50	100	μA
	External Clock		V_{CCL} = 3.0V, SQW : disable, CKOUT: fosc (No load)	4MHz	–	–	500	μA
				1MHz	–	–	150	μA
				32kHz	–	30	70	μA
Battery Backed-up Transit Setup Time		t_{CE}	Fig. 9		0	–	–	ns
Operation Recovery Time		t_R			t_{cyc}	–	–	ns
Supply Voltage Fall Time		t_{Pf}			300	–	–	μs
Supply Voltage Rise Time		t_{Pr}			300	–	–	μs
Input "High" Voltage		V_{IHL}	V_{CCL} = 2.7V ~ 3.5V	$\overline{CE}$, PS, CKFS	0.7×V_{CCL}	–	V_{CCL}	V
			V_{CCL} = 3.5V ~ 4.5V	$\overline{CE}$, PS, CKFS	2.5	–	V_{CCL}	V
				$\overline{RES}$	0.8×V_{CCL}	–	V_{CCL}	V
				OSC_1	0.8×V_{CCL}	–	V_{CCL}	V
Input "Low" Voltage		V_{ILL}		CKFS, PS	-0.3	–	0.5	V
				$\overline{RES}$	-0.3	–	0.5	V
				OSC_1	-0.3	–	0.5	V
Output "High" Voltage		V_{OHL}	I_{OH} = -800μA	SQW, CKOUT	0.8×V_{CCL}	–	–	V
Output "Low" Voltage		V_{OLL}	I_{OL} = 800μA	SQW, CKOUT	–	–	0.5	V
				$\overline{IRQ}$	–	–	0.5	V

* The time-base frequency to be used needs to be chosen in Register A.

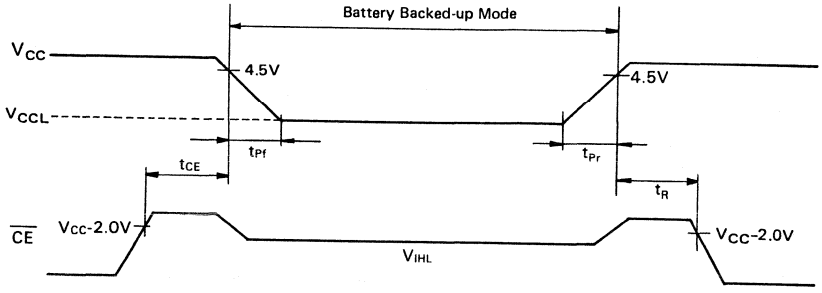


Figure 9 Battery Backed-up Timing

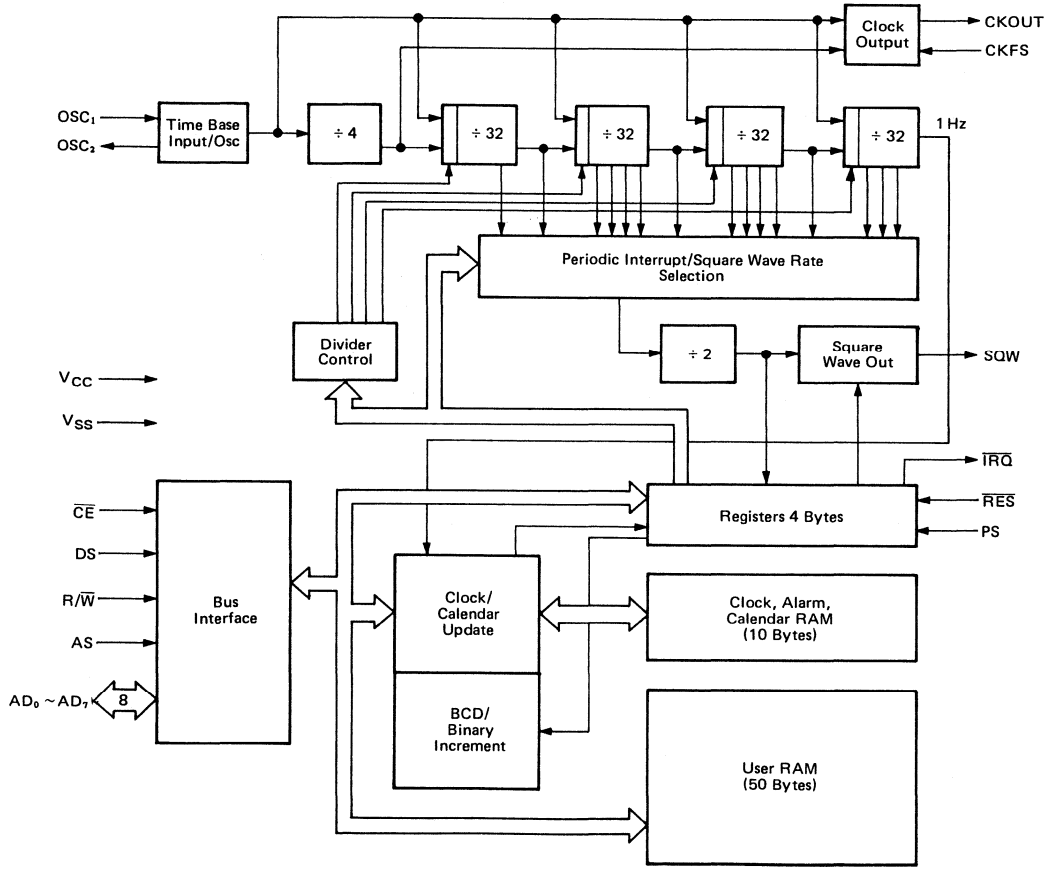


Figure 10 Block Diagram

■ CRYSTAL OSCILLATION CIRCUIT

The on-chip oscillator is designed for a parallel resonant crystal at 4.194304 MHz or 1.048576 MHz or 32.768 kHz frequencies. The crystal connections are shown in Figure 11.

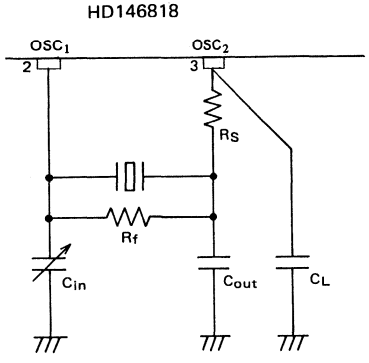


Figure 11 Crystal Oscillator Connection

Table 1 Oscillator Circuit Parameters

Parameter \ fosc	4.194304 MHz	1.048576 MHz	32.768 kHz
R_S	–	–	150 kΩ
R_f	150 kΩ	150 kΩ	5.6 MΩ
C_{in}	22 pF	33 pF	15 pF
C_{out}	22 pF	33 pF	33 pF
C_L	–	–	33 pF
CI	80 Ω (max)	700 Ω (max)	40 kΩ (max)

(NOTE) 1. R_S, C_L are used for 32.768 kHz only.
2. Capacitance (C_{in}) should be adjusted to accurate frequency. Parameters listed above are applied to the supply current measurement (See table of DC CHARACTERISTICS).
3. CI: Crystal Impedance

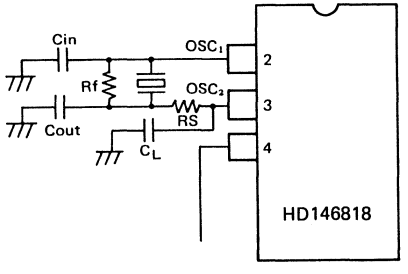
■ NOTE FOR BOARD DESIGN OF THE OSCILLATION CIRCUIT

In designing the board, the following notes should be taken when the crystal oscillator is used.

(1) Crystal oscillator, load capacity C_{in}, C_{out}, C_L and R_f, R_S must be placed near the LSI as much as possible.
[Normal oscillation may be disturbed when external noise is induced to pin 2 and 3.]

(2) Pin 3 signal line should be wired apart from pin 4 signal line as much as possible. Don't wire them in parallel, or normal oscillation may be disturbed when this signal is feedbacked to OSC_1.

(3) A signal line or a power source line must not cross or go near the oscillation circuit line as shown in the right figure to prevent the induction from these lines and perform the correct oscillation. The resistance among OSC_1, OSC_2 and other pins should be over 10MΩ.



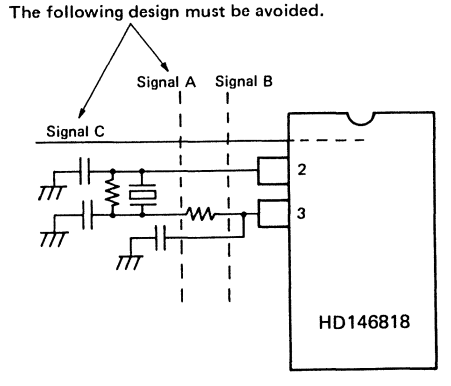


Figure 12 Note for Board Design of the Oscillation Circuit

■ INTERFACE CIRCUIT FOR HD6801, HD6301 AND 8085 PROCESSOR

HD146818 has a new interface circuit which permits the HD146818 to be directly interfaced with many type of multiplexed bus microprocessor such as HD6801, HD6301 and 8085 etc.

Figure 13 shows the bus control circuit. This circuit automatically selects the processor type by using AS/ASE to latch the state of DS/$\overline{RD}$ pin. Since DS is always "Low" and $\overline{RD}$ is always "High during AS/ALE, the latch automatically indicates which processor type is connected.

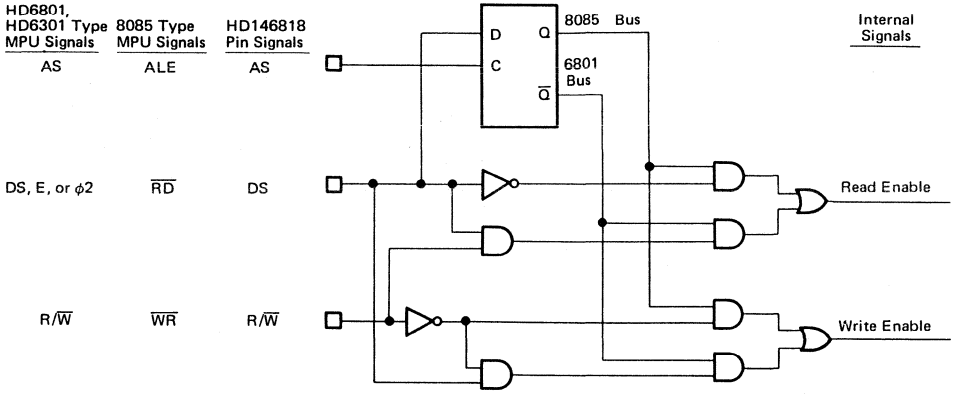


Figure 13 Functional Diagram of the Bus Control Circuit

■ ADDRESS MAP

Figure 14 shows the address map of the HD146818. The memory consists of 50 general purpose RAM bytes, 10 RAM bytes which normally contain the time, calendar, and alarm data, and four control and status bytes. All 64 bytes are directly readable and writable by the processor program except Registers C and D which are read only. Bit 7 of Register A and the seconds byte are also read only. Bit 7, of the second byte, always reads "0". The contents of the four control and status registers are described in the Register section.

• Time, Calendar, and Alarm Locations

The processor program obtains time and calendar information by reading the appropriate locations. The program may initialize the time, calendar, and alarm by writing to these RAM locations. The contents of the 10 time, calendar, and alarm byte may be either binary or binary-coded decimal (BCD).

Before initializing the internal registers, the SET bit in Register B should be set to a "1" to prevent time/calendar updates from occurring. The program initializes the 10 locations in the selected format (binary or BCD), then indicates the format in the data mode (DM) bit of Register B. All 10 time, calendar, and alarm bytes must use the same data mode, either binary or BCD. The SET bit may now be cleared to allow updates. Once initialized the real-time clock makes all updates in the selected data mode. The data mode cannot be changed without reinitializing the 10 data bytes.

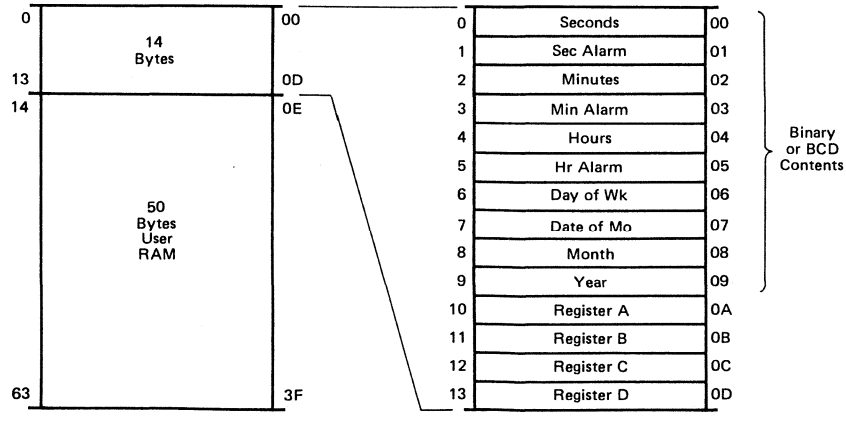


Figure 14 Address Map

Table 2 shows the binary and BCD formats of the 10 time, calendar, and alarm locations. The 24/12 bit in Register B establishes whether the hour locations represent 1-to-12 or 0-to-23. The 24/12 bit cannot be changed without reinitializing the hour locations. When the 12-hour format is selected the high-order bit of the hours byte represents PM when it is a "1".

The time, calendar, and alarm bytes are not always accessable by the processor program. Once-per-second the 10 bytes are switched to the update logic to be advanced by one second and to check for an alarm condition. If any of the 10 bytes are read at this time, the data outputs are undefined. The update lockout time is 248 μs at the 4.194304 MHz and 1.048567 MHz time bases and 1948 μs for the 32.768 kHz time base. The Update Cycle section shows how to accommodate the update cycle in the processor program.

The three alarm bytes may be used in two ways. When the program inserts an alarm time in the appropriate hours, minutes, and seconds alarm locations, the alarm interrupt is initiated at the specified time each day if the alarm enable bit is "1". The alternate usage is to insert a "don't care" state in one or more of three alarm bytes. The "don't care" code is any hexadecimal byte from C0 to FF. That is, the two most-significant bits of each byte, when set to "1", create a "don't care" situation. An alarm interrupt each hour is created with a "don't care" code in the hours alarm location. Similarly, an alarm is generated every minute with "don't care" codes in the hours and minutes alarm bytes. The "don't care" codes in all three alarm bytes create an interrupt every second.

Table 2 Time, Calendar, and Alarm Data Modes

Address Location	Function	Decimal Range	Range		Example*	
			Binary Data Mode	BCD Data Mode	Binary Data Mode	BCD Data Mode
0	Seconds	0~59	$00~$3B	$00~$59	15	21
1	Seconds Alarm	0~59	$00~$3B	$00~$59	15	21
2	Minutes	0~59	$00~$3B	$00~$59	3A	58
3	Minutes Alarm	0~59	$00~$3B	$00~$59	3A	58
4	Hours (12 Hour Mode)	1~12	$01~$0C (AM) and $81~$8C (PM)	$01~$12 (AM) and $81~$92 (PM)	05	05
	Hours (24 Hour Mode)	0~23	$00~$17	$00~$23	05	05
5	Hours Alarm (12 Hour Mode)	1~12	$01~$0C (AM) and $81~$8C (PM)	$01~$12 (AM) and $81~$92 (PM)	05	05
	Hours Alarm (24 Hour Mode)	0~23	$00~$17	$00~$23	05	05
6	Day of the Week Sunday = 1	1~7	$01~$07	$01~$07	05	05
7	Day of the Month	1~31	$01~$1F	$01~$31	0F	15
8	Month	1~12	$01~$0C	$01~$12	02	02
9	Year	0~99**	$00~$63	$00~$99	4F	79

* Example: 5:58:21 Thursday 15th February 1979

** Set the lower two digits of year in AD. If this number is multiple of 4, update applied to leap year is excuted.

• Static CMOS RAM

The 50 general purpose RAM bytes are not dedicated within the HD146818. They can be used by the processor program, and are fully available during the update cycle.

When time and calendar information must use battery back-up, very frequently there is other non-volatile data that must be retained when main power is removed. The 50 user RAM bytes serve the need for low-power CMOS battery-backed storage, and extend the RAM available to the program.

When further CMOS RAM is needed, additional HD146818s may be included in the system. The time/calendar functions may be disabled by holding the dividers, in Register A, in the reset state by setting the SET bit in Register B or by removing the oscillator. Holding the dividers in reset prevents interrupts or SQW output from operating while setting the SET bit allows these functions to occur. With the dividers clear, the available user RAM is extended to 59 bytes. Bit 7 of Register A, Registers C and D, and the high-order Bit of the seconds byte cannot effectively be used as general purpose RAM.

■ INTERRUPTS

The RTC plus RAM includes three separate fully automatic sources of interrupts to the processor. The alarm interrupt may be programmed to occur at rates from once-per-second to one-a-day. The periodic interrupt may be selected for rates from half-a-second to 30.517 μs. The update-ended interrupt may be used to indicate to the program that an up-date cycle is completed. Each of these independent interrupt conditions are described in greater detail in other sections.

The processor program selects which interrupts, if any, it wishes to receive. Three bits in Register B enable the three interrupts. Writing a "1" to a interrupt-enable bit permits that interrupt to be initiated when the event occurs. A "0" in the interrupt-enable bit prohibits the $\overline{IRQ}$ pin from being asserted due to the interrupt cause.

If an interrupt flag is already set when the interrupt becomes enabled, the $\overline{IRQ}$ pin is immediately activated, though the interrupt initiating the event may have occurred much earlier. Thus, there are cases where the program should clear such

earlier initiated interrupts before first enabling new interrupts.

When an interrupt event occurs a flag bit is set to a "1" in Register C. Each of the three interrupt sources have separate flag bits in Register C, which are set independent of the state of the corresponding enable bits in Register B. The flag bit may be used with or without enabling the corresponding enable bits.

In the software scanned case, the program does not enable the interrupt. The "interrupt" flag bit becomes a status bit, which the software interrogates, when it wishes. When the software detects that the flag is set, it is an indication to software that the "interrupt" event occurred since the bit was last read.

However, there is one precaution. The flag bits in Register C are cleared (record of the interrupt event is erased) when Register C is read. Double latching is included with Register C so the bits which are set are stable throughout the read cycle. All bits which are high when read by the program are cleared, and new interrupts (on any bits) are held until after the read cycle. One, two, or three flag bits may be found to be set when Register C is read. The program should inspect all utilized flag bits every time Register C is read to insure that no interrupts are lost.

The second flag bit usage method is with fully enabled interrupts. When an interrupt-flag bit is set and the corresponding interrupt-enable bit is also set, the $\overline{IRQ}$ pin is asserted "Low". $\overline{IRQ}$ is asserted as long as at least one of the three interrupt sources has its flag and enable bits both set. The IRQF bit in Register C is a "1" whenever the $\overline{IRQ}$ pin is being driven "Low".

The processor program can determine that the RTC initiated the interrupt by reading Register C. A "1" in bit 7 (IRQF bit) indicates that one of more interrupts have been initiated by the part. The act of reading Register C clears all the then-active flag bits, plus the IRQF bit. When the program finds IRQF set, it should look at each of the individual flag bits in the same byte which have the corresponding interrupt-mask bits set and service each interrupt which is set. Again, more than one interrupt-flag bit may be set.

■ DIVIDER STAGES

The HD146818 has 22 binary-divider stages following the time base as shown in Figure 10. The output of the dividers is a 1 Hz signal to the update-cycle logic. The dividers are controller by three divider bus (DV2, DV1, and DV0) in Register A.

• Divider Control

The divider-control bits have three uses, as shown in Table 3. Three usable operating time bases may be selected (4.194304 MHz, 1.048576 MHz, or 32.768 kHz). The divider chain may be held reset, which allows precision setting of the time. When the divider is changed from reset to an operating time base, the first update cycle is one second later. The divider-control bits are also used to facilitate testing the HD146818.

Table 3 Divider Configurations

Time-Base Frequency	Divider Bits Register A			Operation Mode	Divider Reset	Bypass First N-Divider Bits
	DV2	DV1	DV0			
4.194304 MHz	0	0	0	Yes	–	N = 0
1.048576 MHz	0	0	1	Yes	–	N = 2
32.768 kHz	0	1	0	Yes	–	N = 7
Any	1	1	0	No	Yes	–
Any	1	1	1	No	Yes	–

(NOTE) Other combinations of divider bits are used for test purposes only.

• Square-Wave Output Selection

Fifteen of the 22 divider taps are made available to a 1-of-15 selector as shown in Figure 10. The first purpose of selecting a divider tap is to generate a square-wave output signal in the SQW pin. Four bits in Register A establish the square-wave frequency as listed in Table 4. The SQW frequency selection shares the 1-of-15 selector with periodic interrupts.

Once the frequency is selected, the output of the SQW pin may be turned on and off under program control with the square-wave enable (SQWE) bit in Register B. Altering the divider, square-wave output selection bits, or the SQW output-enable bit may generate an asymetrical waveform at the time of execution. The square-wave output pin has a number of potential uses. For example, it can serve as a frequency standard for external use, a frequency synthesizer, or could be used to generate one or more audio tones under program control.

• Periodic Interrupt Selection

The periodic interrupt allows the $\overline{IRQ}$ pin to be triggered from once every 500 ms to once every 30.517 μs. The periodic interrupt is separate from the alarm interrupt which may be output from once-per-second to once-per-day.

Table 4 shows that the periodic interrupt rate is selected with the same Register A bits which select the square-wave frequency. Changing one also changes the other. But each function may be separately enabled so that a program could switch between the two features or use both. The SQW pin is enabled by the SQWE bit. Similarly the periodic interrupt is enabled by the PIE bit in Register B.

Periodic interrupt is usable by practically all real-time systems. It can be used to scan for all forms of input from contact closures to serial receive bits or tyes. It can be used in multiplexing displays or with software counters to measure inputs, create output intervals, or await the next needed software function.

Table 4 Periodic Interrupt Rate and Square Wave Output Frequency

Rate Select Control Register 1				4.194304 or 1.048576 MHz Time Base		32.768 kHz Time Base	
RS3	RS2	RS1	RS0	Periodic Interrupt Rate t_{PI}	SQW Output Frequency	Periodic Interrupt Rate t_{PI}	SQW Output Frequency
0	0	0	0	None	None	None	None
0	0	0	1	30.517 μs	32.768 kHz	3.90625 ms	256 Hz
0	0	1	0	61.035 μs	16.384 kHz	7.8125 ms	128 Hz
0	0	1	1	122.070 μs	8.192 kHz	122.070 μs	8.192 kHz
0	1	0	0	244.141 μs	4.096 kHz	244.141 μs	4.096 kHz
0	1	0	1	488.281 μs	2.048 kHz	488.281 μs	2.048 kHz
0	1	1	0	976.562 μs	1.024 kHz	976.562 μs	1.024 kHz
0	1	1	1	1.953125 ms	512 Hz	1.953125 ms	512 Hz
1	0	0	0	3.90625 ms	256 Hz	3.90625 ms	256 Hz
1	0	0	1	7.8125 ms	128 Hz	7.8125 ms	128 Hz
1	0	1	0	15.625 ms	64 Hz	15.625 ms	64 Hz
1	0	1	1	31.25 ms	32 Hz	31.25 ms	32 Hz
1	1	0	0	62.5 ms	16 Hz	62.5 ms	16 Hz
1	1	0	1	125 ms	8 Hz	125 ms	8 Hz
1	1	1	0	250 ms	4 Hz	250 ms	4 Hz
1	1	1	1	500 ms	2 Hz	500 ms	2 Hz

- **Initialization of the Time and the Start Sequence**

The first update of the time occurs about 500ms later after the SET bit of control register B is reset. So keep followings in mind when initializing and adjusting the time.

Procedure of time initialization

(1) Set the SET bit of control register B. (SET = "1")
(2) Set "1" into all the DV0, 1, 2 bits of control register A. (DV0 = DV1 = DV2 = "1")
(3) Set the time and calendar to each RAM.
(4) Set the frequency in use into DV0, 1 and DV2.
(5) Reset the SET bit. (SET = "0")

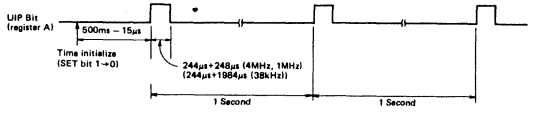


Figure 15 Time Initialization and the First Update

Restriction on Time-of-day and Calendar Initialization

There is a case in HD146818 (RTC) that update is not executed correctly if time of day and calendar shown below are initialized. Therefore, initialize the RTC without using time of day shown below.

Calendar, Time of day & Status after Update	Examples
If 29th 23:59:59 in all the months is initialized, update to 1st in the next month is executed. (Jan. – Dec. However except for Feb. 29th in leap year)	Mar. 29th →Apr. 1st
If 30th 23:59:59 in Apr., June, Sept., and Nov. is initialized, update to 31st in each month is executed.	Apr. 30th →Apr. 31st
If Feb. 28th 23:59:59 (not in leap year) is initialized, update to Feb. 29th is executed.	Feb. 28th,1983 →Feb. 29th,1983
If Feb. 28th 23:59:58 (in leap year) is initialized, update to Mar. 1st is executed.	Feb. 28th,1984 →Mar. 1st,1984

■ UPDATE CYCLE

The HD146818 executes an update cycle once-per-second, assuming one of the proper time bases is in place, the divider is not clear, and the SET bit in Register B is clear. The SET bit in the "1" state permits the program to initialize the time and calendar bytes by stopping an existing update and preventing a new one from occurring.

The primary function of the update cycle is to increment the seconds byte, check for overflow, increment the minutes byte when appropriate and so forth through to the year of the century byte. The update cycle also compares each alarm byte with the corresponding time byte and issues an alarm if a match or if a "don't care" code (11XXXXXX) is present in all three positions.

With a 4.194304 MHz or 1.048576 MHz time base the up-

date cycle takes 248 μs while a 32.768 kHz time base update cycle takes 1984 μs. During the update cycle, the time, calendar, and alarm bytes are not accessable by the processor program. The HD146818 protects the program from reading transitional data. This protection is provided by switching the time, calendar, and alarm portion of the RAM off the microprocessor bus during the entire update cycle. If the processor reads these RAM locations before the update is complete the output will be undefined. The update in progress (UIP) status bit is set during the interval.

A program which randomly accesses the time and date information finds data unavailable statistically once every 4032 attempts. Three methods of accommodating nonavailability during update are usable by the program. In discussing the three methods it is assumed that at random points user programs are able to call a subroutine to obtain the time of day.

The first method of avoiding the update cycle uses the update-ended interrupt. If enabled, an interrupt occurs after every update cycle which indicates that over 999 ms are available to read valid time and date information. During this time a display could be updated or the information could be transfered to continuously available RAM. Before leaving the interrupt service routine, the IRQF bit in Register C should be cleared.

The second method uses the update-in-progress bit (UIP) in Register A to determine if the update cycle is in progress or not. The UIP bit will pulse once-per-second. Statistically, the UIP bit will indicate that time and date information is unavailable once every 2032 attempts. After the UIP bit goes "1", the update cycle begins 244 μs later. Therefore, if a "0" is read on the UIP bit, the user has at least 244 μs before the time/calendar data will be changed. If a "1" is read in the UIP bit, the time/calendar data may not be valid. The user should avoid interrupt service routines that would cause the time needed to read valid time/calendar data to exceed 244 μs.

The third method uses a periodic interrupt to determine if an update cycle is in progress. The UIP bit in Register A is set "1" between the setting of the PF bit in Register C (see Figure 16) Periodic interrupts that occur at a rate of greater than $t_{BUC} + t_{UC}$ allow valid time and date information to be read at each occurrence of the periodic interrupt. The reads should be completed within $(t_{PI} \div 2) + t_{BUC}$ to insure that data is not read during the update cycle.

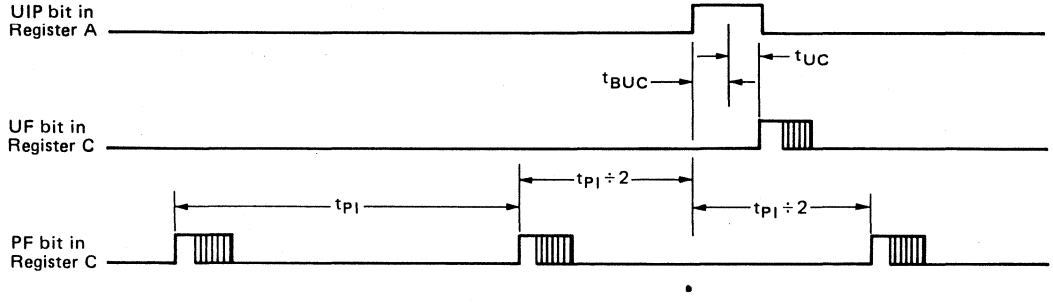
■ POWER-DOWN CONSIDERATIONS

In most systems, the HD146818 must continue to keep time when system power is removed. In such systems, a conversion from system power to an alternate power supply, usually a battery, must be made. During the transition from system to battery power, the designer of a battery backed-up RTC system must protect data integrity, minimize power consumption, and ensure hardware reliability according to the specification described in the section regarding Battery Backed-up operation.

The chip enable ($\overline{CE}$) pin controls all bus inputs (R/$\overline{W}$, DS, AS, $AD_0 \sim AD_7$). $\overline{CE}$, when negated, disallows any unintended modification of the RTC data by the bus. $\overline{CE}$ also reduces power consumption by reducing the number of transitions seen internally.

Power consumption may be further reduced by removing resistive and capacitive loads from the clock out (CKOUT) pin and the squarewave (SQW) pin.

During and after the power source conversion, the V_{IN} maximum specification must never be exceeded. Failure to meet the V_{IN} maximum specification can cause a virtual SCR to appear which may result in excessive current drain and destruction of the part.



t_{PI} = Periodic Interrupt Time Interval (500 ms, 250 ms, 125 ms, 62.5 ms, etc.)
t_{UC} = Update Cycle Time (248 μs or 1984 μs)
t_{BUC} = Delay Time Before Update Cycle (244 μs)

Figure 16 Update-Ended and Periodic Interrupt Relationship

■ SIGNAL DESCRIPTIONS

The block diagram in Figure 10, shows the pin connection with the major internal functions of the HD146818 Real-Time Clock plus RAM. The following paragraphs describe the function of each pin.

• V_{CC}, V_{SS}

DC power is provided to the part on these two pins, V_{CC} being the most positive voltage. The minimum and maximum voltages are listed in the Electrical Characteristics tables.

• OSC_1, OSC_2 – Time Base

Input	Pin No. 2, 3

The time base for the time functions may be an external signal or the crystal oscillator. External square waves at 4.194304 MHz, 1.048576 MHz, or 32.768 kHz may be connected to OSC_1 as shown in Figure 17 The time-base frequency to be used is chosen in Register A.

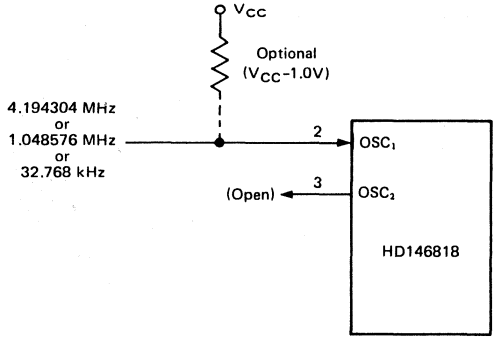


Figure 17 External Time-Base Connection

The on-chip oscillator is designed for a parallel resonant crystal at 4.194304 MHz or 1.048576 MHz or 32.768 kHz frequencies. The crystal connections are shown in Figure 11.

• CKOUT – Clock Out

Output	Pin No. 21

The CKOUT pin is an output at the time-base frequency divided by 1 or 4. A major use for CKOUT is as the input clock to the microprocessor; thereby saving the cost of a second crystal. The frequency of CKOUT depends upon the time-base frequency and the state of the CKFS pin as shown in Table 5.

• CKFS – Clock Out Frequency Select

Input	Pin No. 20

The CKOUT pin is an output at the time-base frequency divided by 1 or 4. CKFS tied to V_{CC} causes CKOUT to be the same frequency as the time base at the OSC_1 pin. When CKFS is at V_{SS}, CKOUT is the OSC_1 time-base frequency divided by four. Table 5 summarizes the effect of CKFS.

Table 5 Clock Output Frequencies

Time Base (OSC_1) Frequency	Clock Frequency Select Pin (CKFS)	Clock Frequency Output Pin (CKOUT)
4.194304 MHz	"High"	4.194304 MHz
4.194304 MHz	"Low"	1.048576 MHz
1.048576 MHz	"High"	1.048576 MHz
1.048576 MHz	"Low"	262.144 kHz
32.768 kHz	"High"	32.768 kHz
32.768 kHz	"Low"	8.192 kHz

• SQW – Square Wave

Output	Pin No. 23

The SQW pin can output a signal one of 15 of the 22 internal-divider stages. The frequency and output enable of the SQW may be altered by programming Register A, as shown in Table 4. The SQW signal may be turned on and off using a bit in Register B.

• $AD_0 \sim AD_7$ – Multiplexed Bidirectional Address/Data Bus

Input/Output	Pin No. 4 ~ 11

Multiplexed bus processors save pins by presenting the address during the first portion of the bus cycle and using the same pins during the second portion for data. Address-then-data multiplexing does not slow the access time of the HD146818 since the bus reversal from address to data is occurring during the internal RAM access time.

The address must be valid just prior to the fall of AS/ALE at which time the HD146818 latches the address from AD_0 to AD_5. Valid write data must be presented and held stable during the latter portion of the DS or $\overline{WR}$ pulses. In a read cycle, the HD146818 outputs 8 bits of data during the latter portion of the DS or $\overline{RD}$ pulses, then ceases driving the bus (returns the output drivers to three-state) when DS falls in the HD6801, HD6301 case or $\overline{RD}$ rises in the other case.

• AS – Multiplexed Address Strobe

Input	Pin No. 14

A positive going multiplexed address strobe pulse serves to demultiplex the bus. The falling edge of AS or ALE causes the address to be latched within the HD146818. The bus control circuit in the HD146818 also latches the state of the DS pin with the falling edge of AS or ALE.

• DS – Data Strobe or Read

Input	Pin No. 17

The DS pin has two interpretations via the bus control circuit. When emanating from 6801 family type processor,

DS is a positive pulse during the latter portion of the bus cycle, and is variously called DS (data strobe), E (enable), and ϕ_2 (ϕ_2 clock). During read cycles, DS signifies the time that the RTC is to drive the bidirectional bus. In write cycles, the trailing edge of DS causes the Real-Time Clock plus RAM to latch the written data.

The second interpretation of DS is that of $\overline{RD}$, $\overline{MEMR}$, or $\overline{I/OR}$ emanating from the 8085 type processor. In this case, DS identifies the time period when the real-time clock plus RAM drives the bus with read data. This interpretation of DS is also the same as an output-enable signal on a typical memory.

The bus control circuit, within the HD146818, latches the state of the DS pin on the falling edge of AS/ALE. When 6801 mode, DS must be "Low" during AS/ALE, which is the case with 6801 family multiplexed bus processors. To insure the 8085 mode of this circuit the DS pin must remain "High" during the time AS/ALE is "High".

- **R/$\overline{W}$ – Read/Write**

Input	Pin No. 15

The bus control circuit treats the R/$\overline{W}$ pin in one of two ways. When 6801 family type processor is connected, R/$\overline{W}$ is a level which indicates whether the current cycle is a read or write. A read cycle is indicated with a "High" level on R/$\overline{W}$ while DS is "High", whereas a write cycle is a "Low" on R/$\overline{W}$ during DS

The second interpretation of R/$\overline{W}$ is as a negative write pulse, $\overline{WR}$, $\overline{MEMW}$, and $\overline{I/OW}$ from 8085 type processors. This circuit in this mode gives R/$\overline{W}$ pin the same meaning as the write ($\overline{W}$) pulse on many generic RAMs.

- **$\overline{CE}$ – Chip Enable**

Input	Pin No. 13

The chip-enable ($\overline{CE}$) signal must be asserted ("Low") for a bus cycle in which the HD146818 is to be accessed. $\overline{CE}$ is not latched and must be stable during DS and AS (in the 6801 case) and during $\overline{RD}$ and $\overline{WR}$ (in the 8085 case). Bus cycles which take place without asserting $\overline{CE}$ cause no actions to take place within the HD146818. When $\overline{CE}$ is "High", the multiplexed bus output is in a high-impedance state.

When $\overline{CE}$ is "High", all address, data, DS, and R/$\overline{W}$ inputs from the processor are disconnected within the HD146818. This permits the HD146818 to be isolated from a powered-down processor. When $\overline{CE}$ is held "High", an unpowered device cannot receive power through the input pins from the real-time clock power source. Battery power consumption can thus be reduced by using a pullup resistor or active clamp on $\overline{CE}$ when the main power is off.

- **$\overline{IRQ}$ – Interrupt Request**

Output	Pin No. 19

The $\overline{IRQ}$ pin is an active "Low" output of the HD146818 that may be used as an interrupt input to a processor. The $\overline{IRQ}$ output remains "Low" as long as the status bit causing the interrupt is present and the corresponding interrupt-enable bit is set. To clear the $\overline{IRQ}$ pin, the processor program normally reads Register C. The $\overline{RES}$ pin also clears pending interrupts.

When no interrupt conditions are present, the $\overline{IRQ}$ level is in the high-impedance state. Multiple interrupting devices may thus be connected to an $\overline{IRQ}$ bus with one pullup at the processor.

- **$\overline{RES}$ – Reset**

Input	Pin No. 18

The $\overline{RES}$ pin does not affect the clock, calendar, or RAM functions. On powerup, the $\overline{RES}$ pin must be held "Low" for the specified time, t_{RLH}, in order to allow the power supply to stabilize, Figure 18 shows a typical representation of the $\overline{RES}$ pin circuit.

When $\overline{RES}$ is "Low" the following occurs:

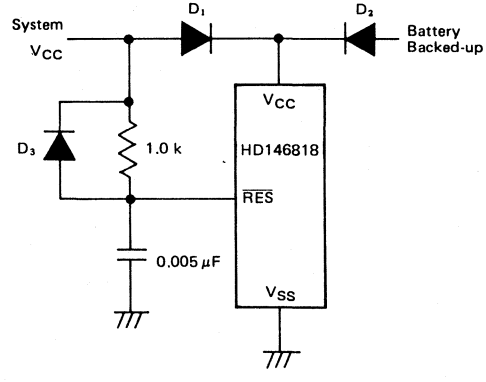
a) Periodic Interrupt Enable (PIE) bit is cleared to "0".
b) Alarm Interrupt Enable (AIE) bit is cleared to "0".
c) Update ended interrupt Enable (UIE) bit is cleared to "0".
d) Update ended Interrupt Flag (UF) bit is cleared to "0".
e) Interrupt Request status Flag (IRQF) bit is cleared to "0".
f) Periodic Interrupt Flag (PF) bit is cleared to "0".
g) Alarm Interrupt Flag (AF) bit is cleared to "0".
h) $\overline{IRQ}$ pin is in high-impedance state, and
i) Square Wave output Enable (SQWE) bit is cleared to "0".

- **PS – Power Sense**

Input	Pin No. 22

The power-sense pin is used in the control of the valid RAM and time (VRT) bit in Register C. When the PS pin is "Low" the VRT bit is cleared to "0".

During powerup, the PS pin must be externally held "Low" for the specified time, t_{PLH}. As power is applied the VRT bit remain "Low" indicating that the contents of the RAM, time registers, and calendar are not guaranteed. When normal opera-



(NOTE) If the RTC is isolated from the MPU or MCU power by a diode drop, care must be taken to meet V_{in} requirements.

Figure 18 Typical Powerup Delay Circuit for $\overline{RES}$

tion commences PS should be permitted to go "High". Output signal from external power sence circuit will be connected to this input.

■ REGISTERS

The HD146818 has four registers which are accessible to the processor program. The four registers are also fully accessible during the update cycle.

• Register A ($0A)

MSB							LSB	
b7	b6	b5	b4	b3	b2	b1	b0	Read/Write Register except UIP
UIP	DV2	DV1	DV0	RS3	RS2	RS1	RS0	

UIP – The update in progress (UIP) bit is a status flag that may be monitored by the program. When UIP is a "1" the update cycle is in progress or will soon begin. When UIP is a "0" the update cycle is not in progress and will not be for at least 244 μs (for all time bases). This is detailed in Table 6. The time, calendar, and alarm information in RAM is fully available to the program when the UIP bit is zero – it is not in transition. The UIP bit is a read-only bit, and is not affected by Reset. Writing the SET bit in Register B to a "1" inhibit any update cycle and then clear the UIP status bit.

Table 6 Update Cycle Times

UIP Bit	Time Base (OSC_1)	Update Cycle Time (t_{UC})	Minimum Time Before Update Cycle (t_{BUC})
1	4.194304 MHz	248 μs	–
1	1.048576 MHz	248 μs	–
1	32.768 kHz	1984 μs	–
0	4.194304 MHz	–	244 μs
0	1.048576 MHz	–	244 μs
0	32.768 kHz	–	244 μs

DV2, DV1, DV0 – Three bits are used to permit the program to select various conditions of the 22-stage divider chain. The divider selection bits identify which of the three time-base frequencies is in use. Table 3 shows that time bases of 4.194304 MHz, 1.048576 MHz, and 32.768 kHz may be used. The divider selection bits are also used to reset the divider chain. When the time/calendar is first initialized, the program may start the divider at the precise time stored in the RAM. When the divider reset is removed the first update cycle begins half a second later. These three read/write bits are never modified by the RTC and are not affected by $\overline{RES}$.

RS3, RS2, RS1, RS0 – The four rate selection bits select one of 15 taps on the 22-stage divider, or disable the divider output. The tap selected may be used to generate an output square wave (SQW pin) and/or a periodic interrupt. The program may do one of the following: 1) enable the interrupt with the PIE bit, 2) enable the SQW output pin with the SQWE bit, 3) enable both at the same time at the same rate, or 4) enable neither. Table 4 lists the periodic interrupt rates and the square-wave frequencies that may be chosen with the RS bits. These four bits are read/write bits which are not affected by $\overline{RES}$ and are never changed by the RTC.

• Register B ($0B)

MSB							LSB	
b7	b6	b5	b4	b3	b2	b1	b0	Read/Write Register
SET	PIE	AIE	UIE	SQWE	DM	24/12	DSE	

SET – When the SET bit is a "0", the update cycle functions normally by advancing the counts once-per-second. When the SET bit is written to a "1", any update cycle in progress is aborted and the program may initialize the time and calendar bytes without an update occurring in the midst of initializing. SET is a read/write bit which is not modified by $\overline{RES}$ or internal functions of the HD146818.

PIE – The periodic interrupt enable (PIE) bit is a read/write bit which allows the periodic-interrupt flag (PF) bit to cause the $\overline{IRQ}$ pin to be driven "Low". A program writes a "1" to the PIE bit in order to receive periodic interrupts at the rate specified by the RS3, RS2, RS1, and RS0 bits in Control Register A. A "0" in PIE blocks $\overline{IRQ}$ from being initiated by a periodic interrupt, but the periodic flag (PF) bit is still at the periodic rate. PIE is not modified by any internal HD146818 functions, but is cleared to "0" by a $\overline{RES}$.

AIE – The alarm interrupt enable (AIE) bit is a read/write bit which when set to a "1" permits the alarm flag (AF) to assert $\overline{IRQ}$. An alarm interrupt occurs for each second that the three time bytes equal the three alarm bytes (including a "don't care" alarm code of binary 11XXXXXX). When the AIE bit is a "0", the AF bit does not initiate an $\overline{IRQ}$ signal. The $\overline{RES}$ pin clears AIE to "0". The internal functions do not affect the AIE bit.

UIE – The UIE (update-ended interrupt enable) bit is a read/write bit which enables the update-end flage (UF) bit to assert $\overline{IRQ}$. The $\overline{RES}$ pin going "Low" or the SET bit going "1" clears the UIE bit.

SQWE – When the square-wave enable (SQWE) bit is set to a "1" by the program, a square-wave signal at the frequency specified in the rate selection bits (RS3 to RS0) appears on the SQW pin. When the SQWE bit is set to a "0" the SQW pin is held "Low". The state of SQWE is cleared by the $\overline{RES}$ pin. SQWE is a read/write bit.

DM – The data mode (DM) bit indicates whether time and calendar updates are to use binary or BCD formats. The DM bit is written by the processor program and may be read by the program, but is not modified by any internal functions or $\overline{RES}$. A "1" in DM signifies binary data, while a "0" in DM specified binary-coded-decimal (BCD) data.

24/12 – The 24/12 control bit establishes the format of the hours bytes as either the 24-hour mode (a "1") or the 12-hour mode (a "0"). This is a read/write bit, which is affected only by the software.

DSE – The daylight savings enable (DSE) bit is a read/write bit which allows the program to enable two special updates (when DSE is a "1"). On the last Sunday in April the time increments from 1:59:59 AM to 3:00:00 AM. On the last Sunday in October when the time first reaches 1:59:59 AM it changes to 1:00:00 AM. These special updates do not occur when the DSE bit is a "0". DSE is not changed by any internal operations or reset.

- **Register C ($0C)**

MSB							LSB	
b7	b6	b5	b4	b3	b2	b1	b0	Read-Only Register
IRQF	PF	AF	UF	0	0	0	0	

IRQF – The interrupt request flag (IRQF) is set to a "1" when one or more of the following are true:

PF = PIE = "1"
AF = AIE = "1"
UF = UIE = "1"

i.e., IRQF = PF · PIE + AF · AIE + UF · UIE

Any time the IRQF bit is a "1", the $\overline{IRQ}$ pin is driven "Low". All flag bits are cleared after Register C is read by the program or when the $\overline{RES}$ pin is low. A program write to Register C does not modify any of the flag bits.

PF – The periodic interrupt flag (PF) is a read-only bit which is set to a "1" when a particular edge is detected on the selected tap of the divider chain. The RS3 to RS0 bits establish the periodic rate. PF is set to a "1" independent of the state of the PIE bit. PF being a "1" initiates an $\overline{IRQ}$ signal and the IRQF bit when PIE is also a "1". The PF bit is cleared by a $\overline{RES}$ or a software read of Register C.

AF – A "1" in the AF (alarm interrupt flag) bit indicates that the current time has matched the alarm time. A "1" in the AF causes the $\overline{IRQ}$ pin to go "Low", and a "1" to appear in the IRQF bit, when the AIE bit also is a "1". A $\overline{RES}$ or a read of Register C clears AF.

UF – The update-ended interrupt flag (UF) bit is set after each update cycle. When the UIE bit is a "1", the "1" in UF causes the IRQF bit to be a "1", asserting $\overline{IRQ}$. UF is cleared by a Register C read or a $\overline{RES}$.

b3 to b0 – The unused bits of Status Register C are read as "0's". They can not be written.

- **Register D ($0D)**

MSB							LSB	
b7	b6	b5	b4	b3	b2	b1	b0	Read Only Register
VRT	0	0	0	0	0	0	0	

VRT – The valid RAM and time (VRT) bit indicates the condition of the contents of the RAM, provided the power sense (PS) pin is satisfactorily connected. A "0" appears in the VRT bit when the power-sense pin is "Low". The processor program can set the VRT bit when the time and calendar are initialized to indicate that the RAM and time are valid. The VRT is a read/only bit which is not modified by the $\overline{RES}$ pin. The VRT bit can only be set by reading the Register D. For setting this bit, PS signal needs to be "High" level.

b6 to b0 – The remaining bits of Register D are unused. They cannot be written, but are always read as "0's".

■ NOTE FOR USE

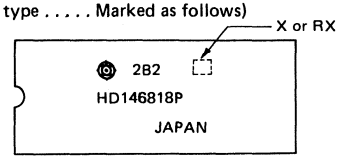
Input Signal, which is not necessary for user's application, should be used fixed to "High" or "Low" level. This is applicable to the following signal pins.

CKFS, PS

RESTRICTION ON HD146818 USAGE (1)

The daylight saving function can not be performed on the HD146818P (X type). So do not use this function for the system design.

< Type number > HD146818P (X type . . . . . Marked as follows)



< Restriction on usage >

Please set "0" to DSE bit (Daylight Saving Enable bit) on initializing the control register B.
DSE = "1" is prohibited.

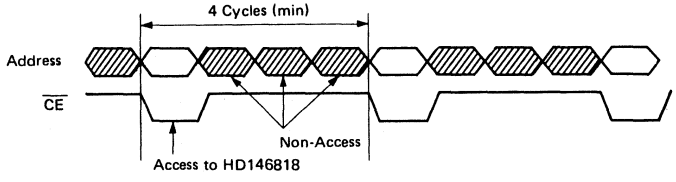
RESTRICTION ON HD146818 USAGE (2)

Access to HD146818 needs to be performed under following conditions.

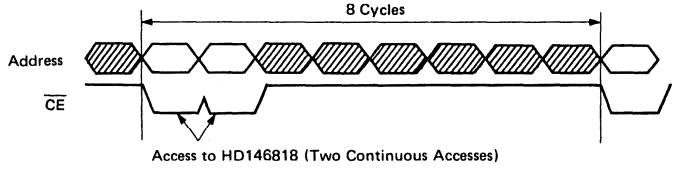
(i) Chip-enable ($\overline{CE}$) must be asserted to active "Low" level only when MPU performs read/write operation from/into internal RAM (Time and Calendar RAM, Control register, User RAM).

(ii) User RAM and control register must be accessed in less than 1/4 frequency shown below.
(Example: After one access, non-access cycles more than three cycles are necessary to be inserted.)

[Example 1]



[Example 2]



As shown in the above [example 2], when HD146818 is accessed continuously, continuous access must not be executed over fifty times.

(iii) The application that User RAM is used for program area should be avoided. (Inhibit continuous access.)

(iv) Minimize the noise by inserting noise bypass condenser between power supply and ground pin (V_{CC}-V_{SS}).
(Insert noise bypass condenser as near HD146818 as possible.)

RESTRICTION ON HD146818 USAGE (3)

Chip-enable ($\overline{CE}$) must be stable between falling edge of DS and rising edge of AS shown below. (Address decoder hazard needs to be externally suppressed in this period.)