UNIVERSAL DISC PROCESSOR

Model 8060

Technical Manual

November 1983

Publication Order Number: 303-003260-000

GOULD
Electronics

HISTORY

The Universal Disc Processor (Model 8060) Technical Manual, Publication Order Number **303-003260-000** was printed November, 1983.

The manual contains the following pages:

Title page
Copyright page
iii/iv through xii
1-1 through 1-8
2-1 through 2-3/2-4
3-1 through 3-23/3-24
4-1 through 4-110
5-1 through 5-2
6-1 through 6-4
A-1 through A-14
B-1 through B-4
C-1 through C-6
D-1 through D-2
E-1 through E-10
F-1 through F-44
G-1 through G-2

CONTENTS

Chapter

Page

Chapter Page

Appendixes

ILLUSTRATIONS

TABLES

CHAPTER 1

GENERAL DESCRIPTION

1.1 Introduction

This manual contains or references information pertaining to the maintenance and installation of the Universal Disc Processor Model 8060. The universal disc processor (UDP), as shown in Figure 1-1, is designed and manufactured by Gould Inc., Computer Systems Division, Fort Lauderdale, Florida.

The information in this manual is presented in the following order:

Chapter 1	General Description
Chapter 2	Operation
Chapter 3	Programming
Chapter 4	Theory of Operation
Chapter 5	Maintenance

The assemblies, circuit cards, logic drawings, and schematics are contained in the Universal Disc Processor Drawings Manual, publication number 304-0003260.

NOTE

The acronym UDP, as used throughout this manual, is synonymous with the Universal Disc Processor, Model 8060, unless otherwise noted.

1.2 Physical Description

The physical description is broken down into five sections: system organization, universal disc processor, disc drives, media specification, and interconnections.

1.2.1 System Organization

The universal disc processor (UDP) system is an extended I/O compatible disc system made up of the UDP and a collection of compatible disc drives (Figure 1-1). A distinction is made between the UDP system, which includes both the UDP and the disc drives, and the UDP, which includes the controller circuits but excludes the disc drives.

The UDP is an integrated channel controller capable of accommodating eight disc drives. Functionally, the channel is a block multiplexer channel capable of executing 16 I/O programs concurrently. The UDP accommodates fixed-head discs, moving-head discs, and winchester discs. Some of the compatible disc drives contain both fixed-head disc and moving-head disc storage on the same spindle.

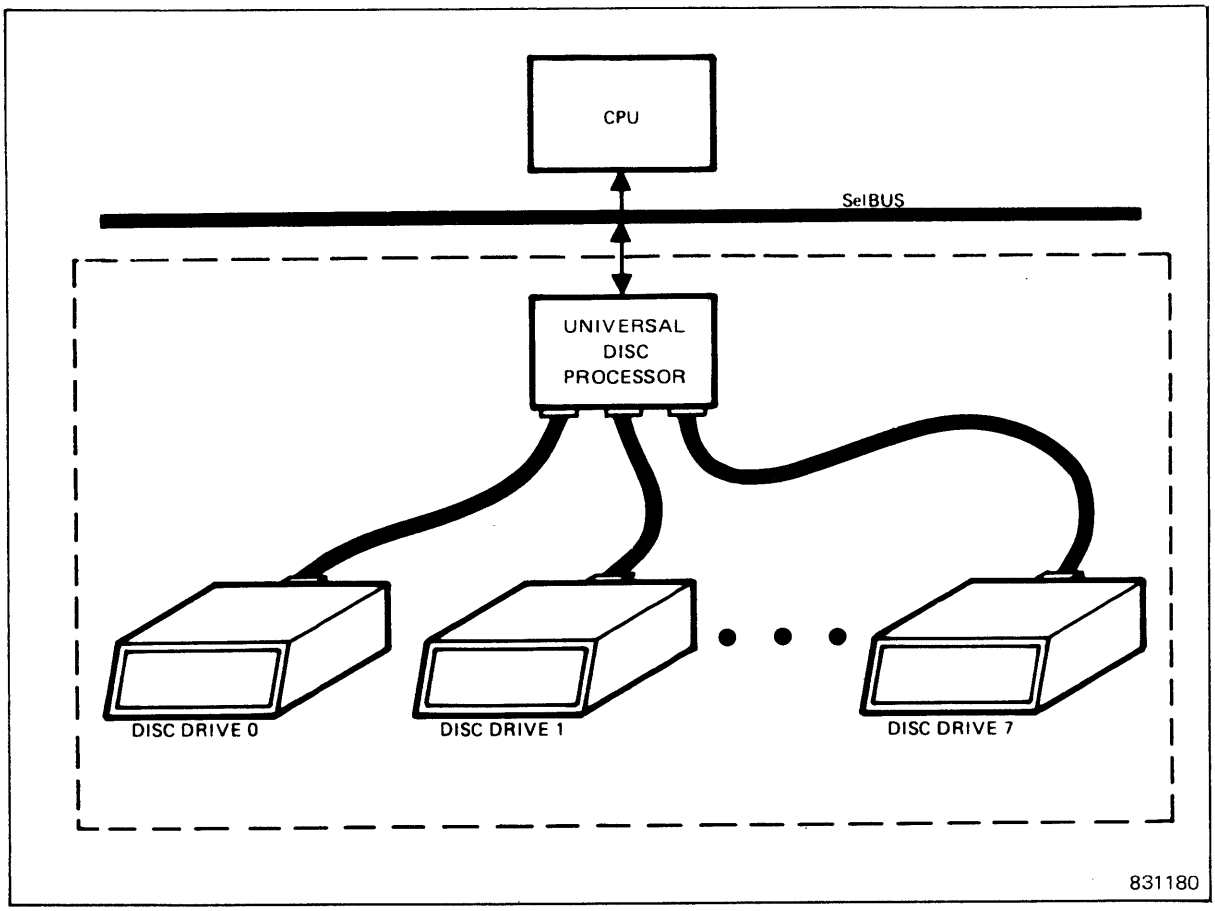


Figure 1-1. Universal Disc Processor System

1.2.2 Universal Disc Processor

The universal disc processor (UDP) consists of up to three circuit cards that act as an integrated channel and control up to eight disc drives. The set of three circuit cards (Figure 1-2) comprise the regional processing unit (RPU), a standard input/output system that has been customized by the addition of device interface logic and special firmware to control channel operation. The three circuit cards include the regional processor (RP), and one or two device interface adapter (DIA) cards. All cards plug into the standard SelBUS connector. The RP card plugs into the front of the SelBUS, while the two DIA cards plug into the back of the SelBUS.

The regional processor module is contained on a single 15-inch wide by 18-inch deep plug-in circuit card. A row of connector pins runs the full width of the card and provides the electrical interface to both the SelBUS and the device interface adapter. These pins are segmented into three groups: 184 pins in the middle (P1B) for the SelBUS and two adjacent groups of 50 pins each (P1A and P1C) for direct connection to the device interface adapter (DIA-1).

The device interface adapter (DIA) cards are contained on single 15-inch wide by 9.2-inch deep plug-in circuit cards. These cards have SelBUS connector pins that receive only power and ground from the 183-pin SelBUS portion of the connector. The DIA cards occupy adjacent card slots on the backplane side of the SelBUS. DIA-1 connects to the

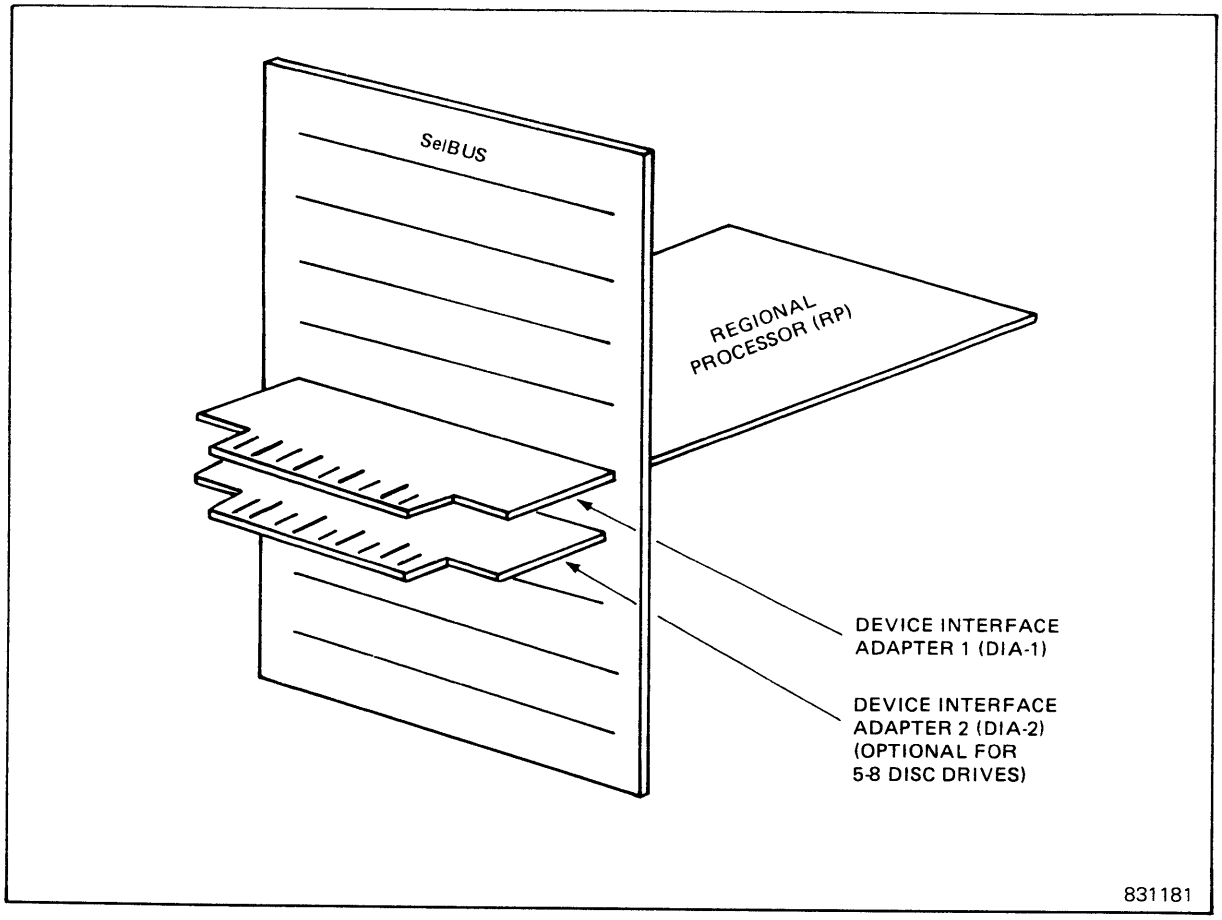


Figure 1-2. Universal Disc Processor

regional processor via backplane connectors J1A and J1C. DIA-2 mounts parallel to DIA-1 and connects to DIA-1 via a ribbon cable between connectors on the rear edge of the cards. DIA-1 plugs into the slot of the SelBUS just opposite the RP, while DIA-2 plugs into the slot of the SelBUS adjacent to the DIA-1 card.

Device interface adapter 1 (DIA-1) contains the primary device-dependent circuitry and provides direct interface with four disc drives.

Expansion from four drives to five or more drives (up to a maximum of eight) requires the installation of DIA-2. The DIA-2 card is also called the port expander, since it adds an additional four ports to the system. DIA-2 is not required if four or fewer disc drives are connected.

1.2.3 Disc Drives

The UDP system will accommodate fixed-head discs and moving-head discs. In addition to these drives, the system can be adapted to many other disc drives. Because no two disc drives are completely functionally identical, drive compatibility must be proven on a drive-by-drive basis. In most cases, a small amount of reengineering will be required to adapt disc drives.

1.2.4 Interconnection

Two types of cables, A cable and B cable, are used for connecting the universal disc processor and the disc drives. The A cable connects DIA-1 to all disc drives in a daisy-chain fashion. The B cable (or radial cable) fans out from either DIA-1 or DIA-2 to the appropriate drive. Figure 1-3 shows the system.

DIA-1 provides for connecting the 60-signal, daisy-chained A cable. This cable connects to all disc drives in the system. The order of connection is not significant; however, the last drive on the A cable bus must be equipped with a terminator, T. The disc drives also connect to the device interface adaptors via 26-signal, radial cable (B cable). B cables connect drives D0, D1, D2, and D3 to flat cable connectors on DIA-1; B cables connect drives D4, D5, D6, and D7 to flat cable connectors on DIA-2.

Each disc drive attached to the system is assigned a unique address in the range 0 to 7. The unit address is determined by a unit address plug installed in the drive. The disc drives must be attached to UDP ports according to their unit address. A port contains logic and a B cable connector. Drive 0 (D0) must connect to port 0 (P0), D1 to P1, D2 to P2, ..., and D7 to P7.

Additional cables connect DIA-1 to DIA-2.

1.3 Functional Description

The UDP is a block multiplexer channel containing up to eight disc drives. The system has the inherent capability of performing a maximum of 16 I/O programs simultaneously. These I/O programs are referred to as subchannels, while the UDP is referred to as a channel.

The channel (universal disc processor) can be considered a virtual channel, since it is implemented through the use of a microprocessor, a microprogram store, and a register store. Two kinds of register stores are used, RPU microprocessor registers and a subchannel status buffer located in main memory.

The UDP makes use of the inherent delays in any disc system (such as head-seek time and rotational delays) to multiplex data for the various subchannels in execution. An executive routine in the channel scans the operating status of the subchannel and makes scheduling decisions for executing the subchannel transfers. For example, if subchannel 0 is busy performing a head seek, the executive scheduler routine will advance to the next subchannel. Using a sector-targeting routine, the executive scheduler will check the current sector count against the sector target for subchannel 1. If the sector count is within the range of the target, the subchannel will execute; if not, the scheduler will advance to the next subchannel. In this manner, the scheduler steps through the subchannels, executing those that are ready and bypassing those that are waiting. The waiting subchannels are picked up on the next or subsequent cycles. Thus, the UDP attempts to maximize the data transfer.

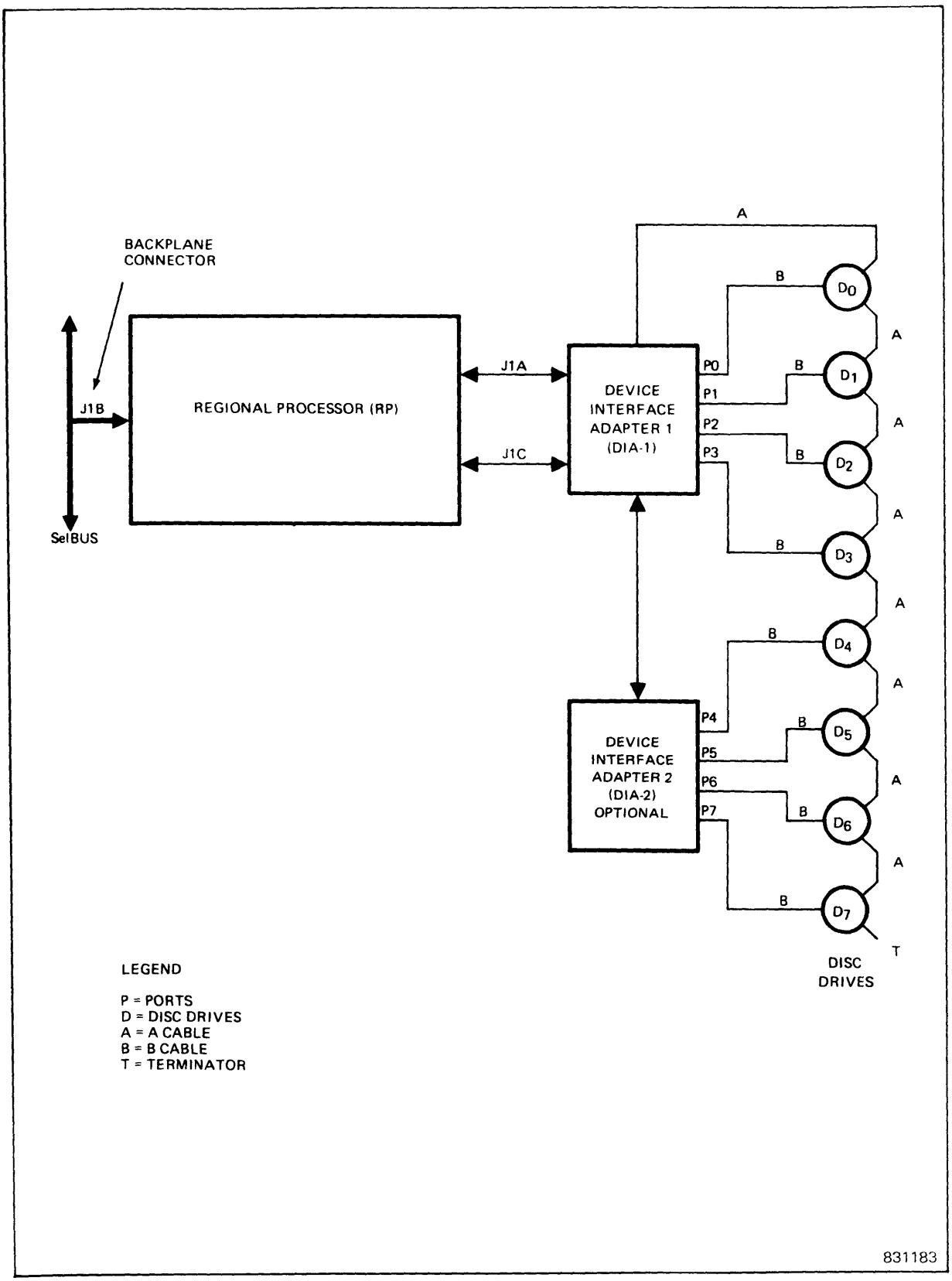


Figure 1-3. Universal Disc Processor Interconnection

Both a microprocessor and a microsequencer are used to implement the channel. The microprocessor, a standard component of the regional processor, controls basic channel operation. The microprocessor is controlled by a resident program (firmware) permanently stored in the regional processor. A microsequencer is located on DIA-1. The purpose of the microsequencer is to manage the actual data transfers to and from the disc. The microprocessor is similarly controlled by permanently stored firmware.

The ability to support both fixed-head disc (FHD) and moving-head disc (MHD) has been coupled with an ability to support a mixture of drive types concurrently. The result is a multipurpose disc system that provides effective support for several different media concepts in the same disc system. The media concepts are: FHD, removable media MHD, captive media FHD, and mixed media disc; mixed media discs are devices that contain combinations of the first three. The channel's block mulitplexing functional characteristics make these combinations feasible.

The channel's functional capabilities make full use of the programmed I/O concepts. These features are augmented with other functional capabilities, such as logical-end-of-disc markers, write protection, write lock, and embedded data buffers. Finally, several modes of error management are implemented or provided for. Implemented error management capabilities include

1. Facilities to support alternate track assignments
2. Facilities to support the use of error correcting codes (ECC)

1.4 Configurations

The UDP can be constructed in two configurations as described below:

Configuration	Equipment
1	RP, DIA-1, 1 to 4 disc drives
2	RP, DIA-1, DIA-2, 5 to 8 disc drives

Any combination of disc drives can be attached to DIA ports. The drive mix and assignment to port is completely arbitrary. However, the drive unit address must coincide with the port identification number.

1.5 Specifications

Specifications for the UDP are listed in Table 1-1.

Table 1-1
Specifications for the Universal Disc Processor (Sheet 1 of 2)

Characteristic	Specification
Regional Processor	
Physical dimensions	15 inches wide by 18 inches deep (38 cm by 46 cm)
Power requirements	Provided by CPU chassis
Environmental requirements	Same as CPU
Throughput rate	1.2 megabytes per second
Word size	
Data word	16/32 bits parallel structure
Microinstruction word	32 bits
Programmable read-only memory	
Capacity	8192 32-bit words
Cycle time	150 nanoseconds
General purpose registers	
Banks 0 and 1, A file	32 registers, 16 each bank
Banks 0 and 1, B file	32 registers, 16 each bank
Order outputs	48 (24 level and 24 pulsed)
Basic operations	Normal Wait on condition true/false Long branch on condition true/false Short branch on condition true/false Indirect long branch (conditional) Indirect short branch (conditional)

Table 1-1
Specifications for the Universal Disc Processor (Sheet 2 of 2)

Characteristic	Specification
ALU operations	Addition/subtraction OR/AND/exclusive OR Shift left/shift left 1 arithmetic/shift left 8 circular Complement Carry Add/subtract one to/from A
Device Interface Adapter 1 (DIA-1)	
Physical dimensions	15 inches wide by 9.2 inches deep
Power requirements	Provided by CPU chassis
Environmental requirements	Same as CPU
Orders	16 level and 8 pulsed from RP
Tests	32 (30-3F and 50-5F, inclusive) to RP
Data/address outputs to RP	16 (LEXT 00-15)
Data/address inputs from RP	32 (LAIN 00-15 and LBIN 00-15)
Microsequencer PROM	1024 20-bit words
Microsequencer orders	8 level and 24 pulsed
Microsequencer tests	16
Microsequencer cycle time	150 nanoseconds
FIFO RAM	256 8-bit bytes
Error-correcting code	30-bit polynomial
Disc ports	4 (discs 0-3)
Device Interface Adapter 2 (DIA-2)	
Physical dimensions	15 inches wide by 9.2 inches deep
Power requirements	Provided by CPU chassis
Environmental requirements	Same as CPU
Disc ports	4 (discs 4-7)

CHAPTER 2

OPERATION

2.1 Introduction

This chapter of the manual contains the operating instructions for the universal disc processor (UDP). Operation of the disc drives is described in the various disc manuals.

2.2 Controls and Indicators

2.2.1 Controls

2.2.1.1 OFFLINE Switch

The regional processor (RP) has an OFFLINE switch on the front edge of the circuit board. This switch must be placed in the ONLINE (right-hand) position for the RPU to communicate on the SelBUS.

2.2.1.2 Physical Address Selection

The RP module has jumpers to select the physical address (SelBUS address) of the RP. The address that is selected by these jumpers must correspond to the RP physical address associated with the I/O instruction address (command device address field) during the CPU initial program load (IPL) of the initial configuration list.

The physical address jumpers are shown on logic drawing 130-103623, Sheet 6, in the drawings manual. The physical address jumpers are referenced by a logic callout of D10-1 through D10-7 and must be set up to reflect the low true physical address of the RP. D10-1 selects the most significant address bit, and D10-7 selects the least significant address bit.

2.2.1.3 SelBUS Priority Recognition Selection

The RP module has a set of 21 priority recognition jumpers. These jumpers are used to provide recognition of system modules that have a SelBUS transfer priority higher than the RP. The priority recognition jumpers are shown on logic drawing 130-103623, Sheet 10, and are referenced by logic callouts A18-1 through A18-8, B18-1 through B18-8, and C18-1 through C18-5. To recognize higher priority transfer devices, the jumper corresponding to all higher priority levels must be jumpered in, and the priority level assigned to this RP and all lower priorities must be left out.

2.2.1.4 SelBUS Priority Generation Selection

The RP module has a set of 22 priority generation jumpers. These jumpers are used to assign the SelBUS transfer priority of this RP. The priority generation jumpers shown on logic drawing 130-103623, Sheet 10, and are referenced by logic callouts A20-1 through A20-8, B20-1 through B20-8, and C20-1 through C20-5. To assign a SelBUS transfer priority level to this RP, the jumper associated with the priority level must be left in, and all remaining jumpers must be removed. The priority level chosen by this procedure must correspond to the priority level used for this RP as set forth by the priority recognition jumpers.

2.2.1.5 SelBUS Priority Enable Selection

The CPU logic chassis backplane has a SelBUS terminator that contains a set of 22 priority enable jumpers. For priority levels assigned to modules on the SelBUS, the jumpers must be OPEN; the jumpers connected to unassigned priority levels must be CLOSED. The appropriate CPU technical manual provides a system-level description of all prerequisite jumpers that must be installed to communicate on the SelBUS.

2.2.1.6 Disc Drive Sector Count Selection

The two formats (determined by software) available with the UDP are Format 20 (20 sectors per track) and Format 16 (16 sectors per track). To set a specific drive to operate in one of these two formats, refer to the technical manual for that disc drive.

2.2.1.7 DIA-1 Configuration Jumpers

The DIA-1 card contains a set of eight configuration jumpers. These jumpers are shown on logic drawing 130-103282, Sheet 4, and identified by the callouts G02-1 through G02-8. These jumpers are used to establish the system configuration to the CPU. The configuration jumper conditions are shown in Table 2-1. The function is true when the jumper is connected.

Table 2-1
Configuration Jumper Definitions

Jumper	Function	Description
0	Not used	N/A
1	Number of drives	Jumper out 1-4 drives Jumper in 5 or more drives
2	Write lock SS1	When set all drives are write lock protected.
3	Write lock SS2	When set all drives are write lock protected programmatically via execute channel program.
4	Reserve track mode	When set a track is reserved via a special command set: Set reserve track (SRM) - enables the track allowing reads and writes. Reset reserve track (XRM) - disables the track from any reads or writes.
5	Not used	N/A
6	Not used	N/A
7	CPU time-out	Prevents overruns during strip label reads

2.2.2 Indicators

The universal disc processor does not have any indicators to monitor system operation. All indications are read from the CPU control panel or terminal.

2.3 Operating Procedures

There are no special operating procedures for the UDP. However, the technical manual for the I/O devices responding to the UDP should be consulted before commencing operation.

2.4 Power On/Off Procedures

DC power is supplied to the UDP through the logic chassis in which it is installed. The power on/off procedures for the logic chassis are described in the CPU technical manual. Refer to the disc drive technical manuals for the power on/off procedures for the disc drives.

CHAPTER 3

PROGRAMMING

3.1 Introduction

This section contains macroprogramming instructions for the universal disc processor (UDP). It also provides a brief description of the software instructions used to control and obtain status from the UDP. A more complete description of software I/O instructions is provided in the CPU reference manual.

Input/output operations are initiated and controlled by information with two types of formats: instructions and commands. Instructions are decoded by the CPU and are part of the CPU program. Commands are decoded and executed by the channel and the device.

Both instructions and commands are created by the software programmer, assembled, and loaded into memory. During execution, CPU firmware translates the macroinstructions and commands written by the programmer into a series of SelBUS transfers between the CPU, memory, and the channel. An example of a typical sequence for a start I/O instruction is shown in Figure 3-1.

3.2 I/O Instructions and Commands

3.2.1 Extended I/O Instructions

The UDP is controlled by the extended I/O set of software instructions. Figure 3-2 provides the basic format for these instructions and lists the suboperation codes of the instruction. The 10 codes shown (SIO, TIO, DCI, RSCHNL, HIO, STPIO, RSCTL, ECI, ACI, and DACI) are the only codes supported by the UDP firmware. Any other codes will generate an error response from the UDP.

For all extended I/O instructions, the constant in bits 16 through 31 of the instruction plus the contents of the general register indicated by bits 6 through 8 of the instruction (if bits 6 through 8 are nonzero) specify the logical channel and subaddress. The channel will ignore the subaddress for operations that pertain only to the channel. The extended I/O instructions can be executed only when the CPU is in privileged mode and operating with a mode setting of program status doubleword (PSD).

3.2.1.1 Start I/O (SIO)

The start I/O instruction initiates an I/O operation or returns the appropriate condition codes and status if an I/O execution cannot be started.

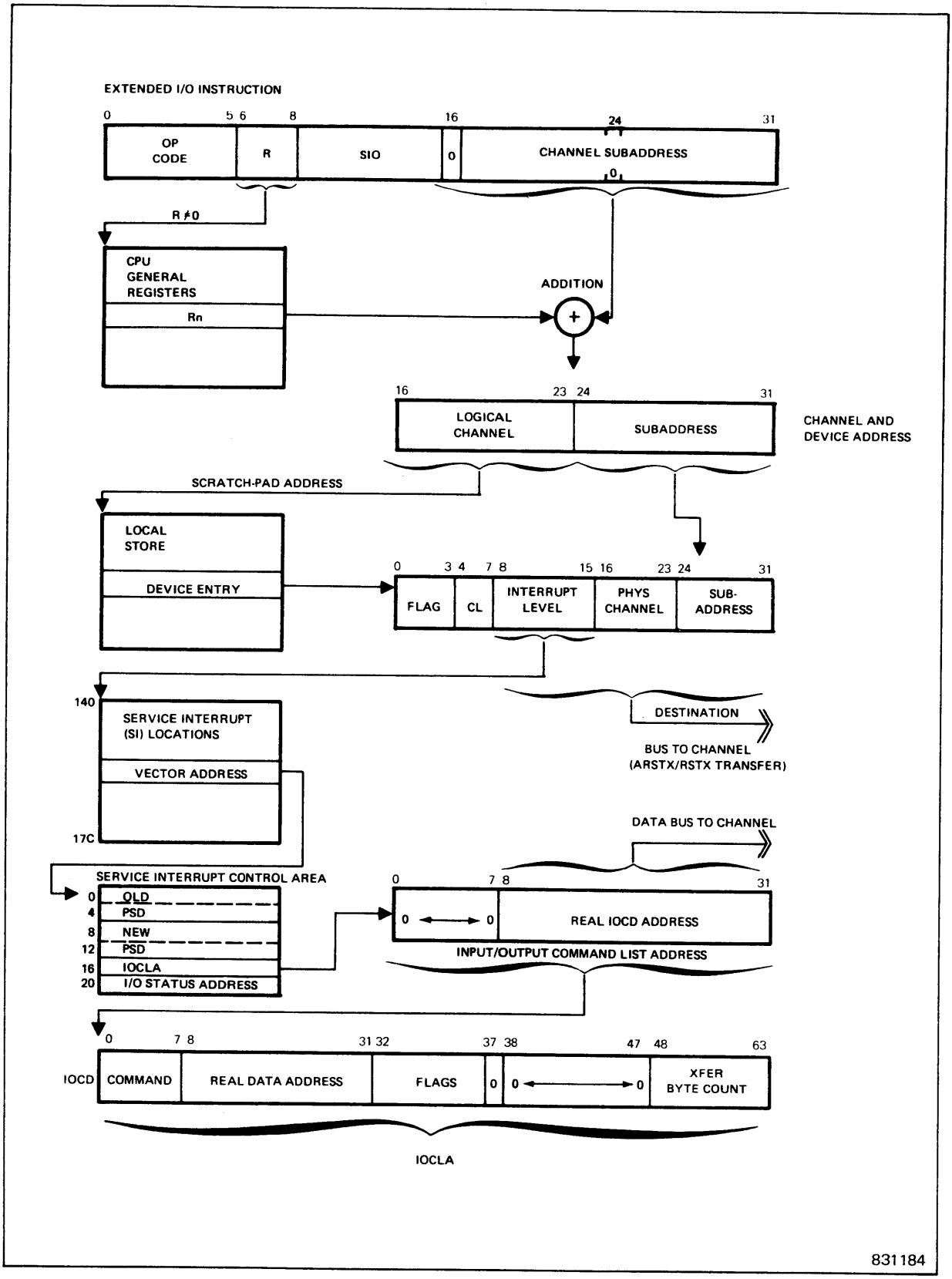
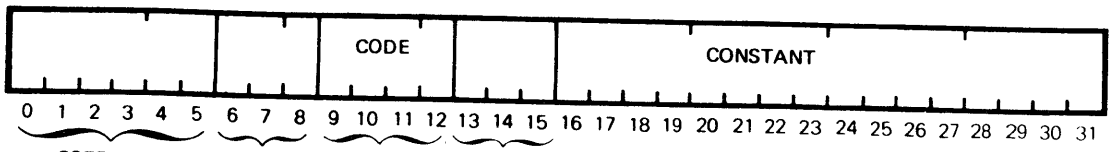


Figure 3-1. Instruction Execution Sequence, Start I/O



BITS	DEFINITION
0-5	SPECIFIES THE OPERATIONS CODE (111111)
6-8	WHEN NONZERO, THIS FIELD SPECIFIES THE GENERAL REGISTER WHOSE CONTENTS WILL BE ADDED TO THE CONSTANT FIELD (BITS 16-31) TO FORM THE LOGICAL CHANNEL AND SUBADDRESS.
9-12	SPECIFIES THE SUBOPERATION CODE:

0010	X'2'	START I/O	(SIO)
0011	X'3'	TEST I/O	(TIO)
0100	X'4'	STOP I/O	(STPIO)
0101	X'5'	RESET CHANNEL	(RSCHNL)
0110	X'6'	HALT I/O	(HIO)
0111	X'7'	GRAB CONTROLLER	(GRIO)*
1000	X'8'	RESET CONTROLLER	(RSCTL)
1001	X'9'	ENABLE WRITE CHANNEL WCS	(ECWCS)*
1011	X'B'	WRITE CHANNEL WCS	(WCWCS)*
1100	X'C'	ENABLE CHANNEL INTERRUPT	(ECI)
1101	X'D'	DISABLE CHANNEL INTERRUPT	(DCI)
1110	X'E'	ACTIVATE CHANNEL INTERRUPT	(ACI)
1111	X'F'	DEACTIVATE CHANNEL INTERRUPT	(DACI)

13-15	AUGMENT CODE (111)
16-31	CONSTANT FIELD RELATED TO ADDRESS

*NOT SUPPORTED BY THE UNIVERSAL DISC PROCESSOR

831186

Figure 3-2. Extended I/O Instruction Format

3.2.1.2 Test I/O (TIO)

The test I/O instruction interrogates the current state of the channel subchannel and may be used to clear pending interrupt conditions. The channel will return the appropriate condition codes and status reflecting the state of the channel and addressed subchannel.

3.2.1.3 Disable Channel Interrupt (DCI)

The disable channel interrupt instruction prevents the channel from requesting interrupts from the CPU.

3.2.1.4 Reset Channel (RSCHNL)

The reset channel instruction causes the channel to cease operation and reset all activity. The channel then returns to the idle state. The command also resets all

subchannels; however, the devices are not affected. Any requesting or active interrupt levels are also reset.

3.2.1.5 Halt I/O (HIO)

The halt I/O instruction causes an immediate but orderly termination in the controller. The device end status will notify software of actual termination in the channel.

3.2.1.6 Stop I/O (STPIO)

The stop I/O command performs an orderly termination of an input/output command doubleword (IOCD) list by resetting the command and data chain flags in the current IOCD. If there is pending status for any subchannel, this command will not be executed and the condition codes presented to the CPU will indicate status stored. The stop I/O command applies only to the addressed subchannel.

3.2.1.7 Reset Controller (RSCTL)

The reset controller command causes the address subchannel to terminate its I/O operation. If this subchannel is in a hung condition, the device will be reset so that normal I/O operation may resume. If the addressed subchannel is not currently being serviced by the disc microengine, then the I/O request will be cleared from the status control registers (SCRi). Any pending status for the addressed subchannel will be presented, but there will be no status generated as a result of the RSCTL command.

3.2.1.8 Enable Channel Interrupt (ECI)

The enable channel interrupt instruction allows the channel to request interrupts from the CPU.

3.2.1.9 Activate Channel Interrupt (ACI)

The activate channel interrupt instruction causes the channel to actively contend for interrupt priority; the channel never requests an interrupt. ACI has no effect on pending status conditions, which can only be cleared by a start I/O or test I/O.

3.2.1.10 Deactivate Channel Interrupt (DACI)

The deactivate channel interrupt instruction causes the channel to suspend contention for interrupt priority. If an interrupt request was queued, the channel may now request interrupt.

3.2.2 Input/Output Initiation

An I/O operation is initiated by a start I/O instruction. If the specified channel/subchannel is present and not busy, the SIO is accepted and the CPU continues to the next sequential instruction. The channel/controller asynchronously processes the I/O request specified by the instruction.

3.2.2.1 Input/Output Command List Address

The input/output command list address (IOCLA) specifies the real address of the first IOCD associated with a start I/O. The information is transferred to the channel, and the contents are not affected by an I/O operation.

Upon execution of the I/O instruction, the software is free to modify the IOCLA. A start I/O is the only instruction that can cause the IOCLA to be transferred to the channel.

Successful execution of the SIO causes the CPU to transmit the IOCLA to the channel. The IOCLA is located in main memory at locations specified by the service interrupt vector plus 16. Each of the 16 channels has a corrresponding service interrupt vector. The format for the IOCLA, indicated by the contents of the service interrupt vector plus 16, is shown in Figure 3-3. The real IOCLA is passed to the channel on the data bus.

3.2.2.2 Input/Output Command Doubleword (IOCD)

The IOCLA specifies the address of the input/output command list (IOCL) to be executed by the channel. An IOCL consists of one or more input/output command doublewords (IOCD). Each IOCD must be aligned on a fullword boundary and has the format shown in Figure 3-4.

The real data address specifies the starting address of the data area. The data address will be a byte address and the channel will internally align the information transferred to or from main memory. While any starting address is allowable, more efficient system operation will result if the software programmer aligns the data area to start at a fullword boundary (bits 30 and 31 of address being zero).

The byte count specifies the number of bytes that are to be transferred to or from main memory. Although the channel may transfer data to/from memory one, two or four bytes at a time, it will accommodate byte counts that are not multiples of its natural transfer width.

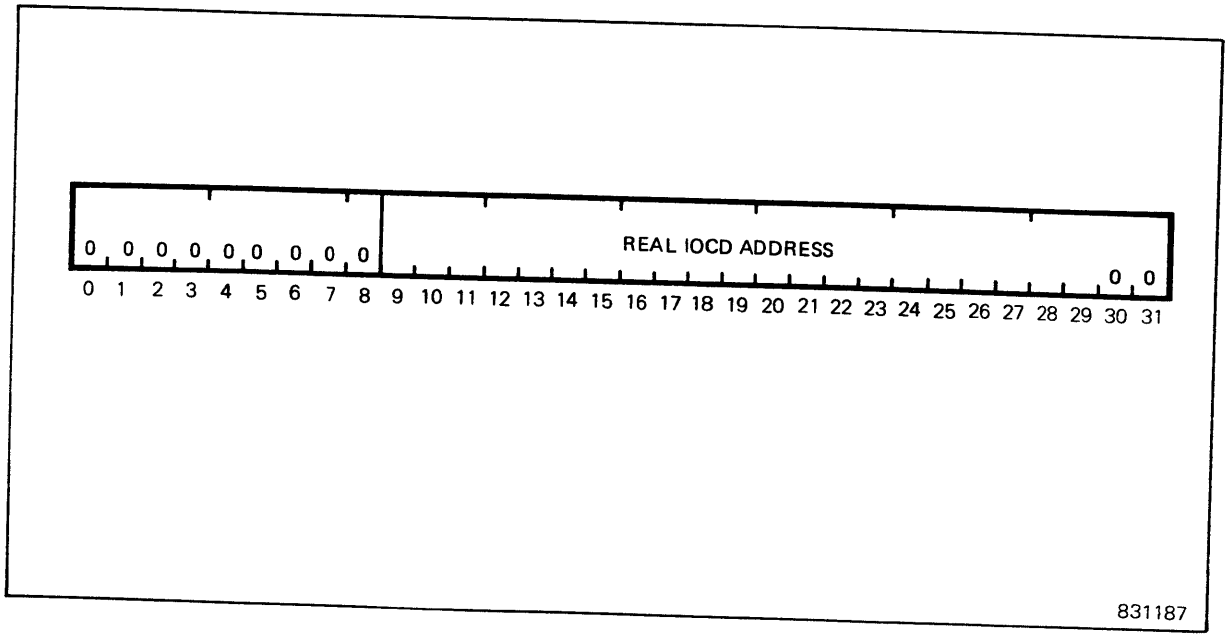
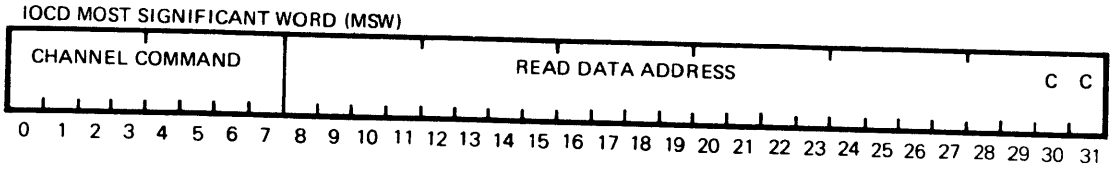
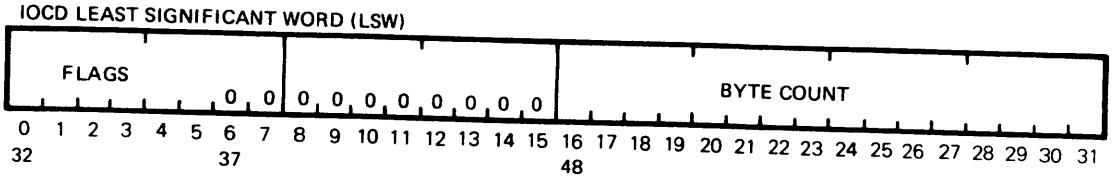


Figure 3-3. Input/Output Command List Address (IOCLA)





WORD	BITS	DEFINITION
MSW	0-7	COMMAND CODE. SPECIFIES CHANNEL COMMAND TO BE EXECUTED BY THE UNIVERSAL DISC PROCESSOR.
MSW	8-31	REAL DATA ADDRESS. THE ADDRESS OF THE LOCATION IN MAIN MEMORY WHERE DATA IS LOCATED OR TO WHICH DATA WILL BE TRANSFERRED.
LSW	0-7	FLAG BYTE. THE FOLLOWING FLAGS MODIFY COMMAND EXECUTION: 0 DATA CHAIN (HOLDS OFF TERMINATION WHEN XFER COUNT = 0) 1 COMMAND CHAIN 2 SUPPRESS INCORRECT LENGTH 3 SKIP 4 PROGRAM CONTROLLED INTERRUPT 5 N/A 6 ZERO 7 ZERO
LSW	8-15	NOT USED. ALWAYS ZERO.
LSW	16-31	BYTE COUNT. DESIGNATES THE NUMBER OF BYTES TO BE TRANSFERRED.

831189

Figure 3-4. Input/Output Command Doubleword (IOCD)

3.2.2.3 Channel Commands (Op Codes)

Channel commands (or op codes) are contained in the first eight bits of the most significant IOCD word. The channel commands can be divided into seven types:

Code	Description
XXXX0000	Channel control
MMMM0100	Sense
XXXX1000	Transfer in channel
MMMM1100	Read backward
MMMMMM01	Write
MMMMMM10	Read
MMMMMM11	Control

The read backward command has no significance for the UDP. The Xs in the codes above indicate the bits are not significant; the Ms indicate modifier bits that are interpreted uniquely for the UDP.

The current version of the UDP supports 24 channel commands, as shown in Table 3-1. For a complete description of channel commands, refer to Chapter 4.

3.3 Interrupt Control Instructions

The channel interrupt is controlled by the CPU instructions ECI, ACI, DACI, DCI, and BRI. Each of the 128 priority interrupts in the CPU has a dedicated memory location containing the address of a software program that should be programmed to handle the interrupt. This memory location is accessed whenever an interrupt occurs from a corresponding interrupt level. If the interrupt level is enabled and the level has the highest priority, the program addressed by the dedicated memory location is initiated. Table 3-2 lists the priority interrupt level available to the CPU and the dedicated memory location (SI address) for each level. Only levels 14H through 23H are dedicated to transfer interrupts.

The disable channel interrupt (DCI) instruction disables the channel interrupt. If there is status on queue, the channel will stop actively polling for a CPU interrupt. If the channel is in the active state, it will remain in this state.

The enable channel interrupt (ECI) instruction enables the channel interrupt. If interrupt status is pending in the channel when the interrupt is enabled, the channel will initiate interrupt polling at the first opportunity. This action can occur during the execution of the ECI instruction or later, depending upon the channel load conditions.

The activate channel interrupt (ACI) instruction activates the channel interrupt. The channel will activate the interrupt even though the channel interrupt is disabled; acceptance of the ACI does not imply that the channel interrupt is enabled nor does the acceptance cause the interrupt to become enabled. Successful execution of the ACI precludes further interrupts from the channel and all lower priority interrupt levels.

The deactivate channel interrupt (DACI) instruction deactivates the interrupt. If the interrupt is enabled, and if additional status is pending in the channel, interrupt polling will be initiated either during the DACI or the branch and reset interrupt (BRI) execution period or at a later time depending upon conditions in the channel.

Table 3-1
Channel Commands

Mnemonic	Code (Hex)	Description	Type
FNSK	OB	Format for no skip	Control
ICH	FF	Controller initialize	Control
IHA	47	Increment head address	Control
INCH	00	Initiate channel	Channel control
LMR	IF	Load mode register	Control
LPL	13	Lock protected label	Control
NOP	03	No operation	Control
POR	43	Priority override	Control
RAP	A2	Read angular positions	Read
RD	02	Read data	Read
REL	33	Release	Control
RES	23	Reserve	Control
RSL	32	Read sector label	Read
RTL	52	Read track label	Read
SCK	07	Seek cylinder, track, sector	Control
SNS	04	Sense	Sense
SRM	4F	Set reserve track mode	Control
TESS	AB	Test STAR (subchannel target address register)	Control
TIC	08	Transfer in channel	Transfer in channel
WD	01	Write data	Write
WSL	31	Write sector label	Write
WTL	51	Write track label	Write
XEZ	37	Rezero	Control
XRM	5F	Reset reserve track mode	Control

Table 3-2
Priority Interrupt Dedicated Memory Locations (Sheet 1 of 2)

Priority Level (H)	Memory Dedicated Address (H)	Function
00*	0F0	Power fail safe - Auto start interrupt
00*	0F4	Power fail safe - Auto start trap
01*	0F8	System override interrupt
01*	0FC	System override trap
02**	100	Input/output controller 1 transfer interrupt
03**	104	Input/output controller 2 transfer interrupt
04**	108	Input/output controller 3 transfer interrupt
05**	10C	Input/output controller 4 transfer interrupt
06**	110	Input/output controller 5 transfer interrupt
07**	114	Input/output controller 6 transfer interrupt
08**	118	Input/output controller 7 transfer interrupt
09**	11C	Input/output controller 8 transfer interrupt
0A**	120	Input/output controller 9 transfer interrupt
0B**	124	Input/output controller 10 transfer interrupt
0C**	128	Input/output controller 11 transfer interrupt
0D**	12C	Input/output controller 12 transfer interrupt
0E**	130	Input/output controller 13 transfer interrupt
0F**	134	Input/output controller 14 transfer interrupt
10**	138	Input/output controller 15 transfer interrupt
11**	13C	Input/output controller 16 transfer interrupt
12*	0E8	Memory parity trap
13*	0EC	Console interrupt (turnkey panel attention)
14	140	Input/output controller 0 service interrupt
15	144	Input/output controller 1 service interrupt
16	148	Input/output controller 2 service interrupt
17	14C	Input/output controller 3 service interrupt
18	150	Input/output controller 4 service interrupt
19	154	Input/output controller 5 service interrupt
1A	158	Input/output controller 6 service interrupt
1B	15C	Input/output controller 7 service interrupt
1C	160	Input/output controller 8 service interrupt
1D	164	Input/output controller 9 service interrupt
1E	168	Input/output controller 10 service interrupt
1F	16C	Input/output controller 11 service interrupt

* Present in first RTOM.

** These dedicated addresses are reserved for transfer control words (TCW) and cannot be used by priority interrupt software.

Table 3-2
Priority Interrupt Dedicated Memory Locations (Sheet 2 of 2)

Priority Level (H)	Memory Dedicated Address (H)	Function
20	170	Input/output controller 12 service interrupt
21	174	Input/output controller 13 service interrupt
22	178	Input/output controller 14 service interrupt
23	17C	Input/output controller 15 service interrupt
24*	190	Nonpresent memory trap
25*	194	Undefined instruction trap
26*	198	Privilege violation trap
27*	19C	Call monitor interrupt
28*	1A0	Real-time clock interrupt
29*	1A4	Arithmetic exception interrupt
2A*	1A8	External interrupt
2B*	1AC	External interrupt
2C*	1B0	External interrupt
2D*	1B4	External interrupt
2E*	1B8	External interrupt
2F*	1BC	External interrupt (last PI in standard RTOM)
30	1C0	External interrupt
↕	↕	
7F	2FC	External interrupt

* Present in first RTOM.

The execution of the channel portion of DACI may be delayed by periods up to 50 microseconds; these delays are infrequent and result from a need to deal with real-time character processing in the channel. Delays in excess of 50 microseconds are indicative of faulty channel operation.

The BRI instruction is also used by software to clear the active condition of the I/O controller interrupt level, although the BRI is not one of the interrupt control instructions. The format and the functions of the BRI instruction are described in the Gould 32 SERIES Computer Reference Manual.

3.4 CPU Initial Program Load

The initial program load (IPL) is used to initialize main memory with a user program. The IPL is dependent upon an operator stimulus (i.e., the panel IPL pushbutton). The presumed state of the CPU when IPL is initiated is as follows:

1. The CPU is halted.
2. The SYSTEM RESET pushbutton is pressed.

3. The operator specifies a physical IPL device address.
4. The IPL pushbutton is pressed.

The IPL will exclude the initial configuration load (ICL), which will become a software-initiated function upon completion of the IPL. The CPU firmware will generate for the extended I/O devices a single IOCD located in main memory locations 0 through 7. The contents of the IOCD will be as shown in Figure 3-5.

The IPL read will be issued to the appropriate channel after a load random access memory (RAM) is issued. The 120 bytes read into memory are in the following format:

0 NEW IPL PSD

8 IOCD1

16 IOCD2

The new program status doubleword (PSD) will be loaded upon receipt of the termination interrupt of the entire IPL operation. IOCD1 and IOCD2 contains the commands to sustain the program load. It should be noted that the initial IPL IOCD has specified command chaining; the next IOCD to be executed will be IOCD1. The IOCD list read from the device can be initialized to read as many records as required to satisfy programming requirements.

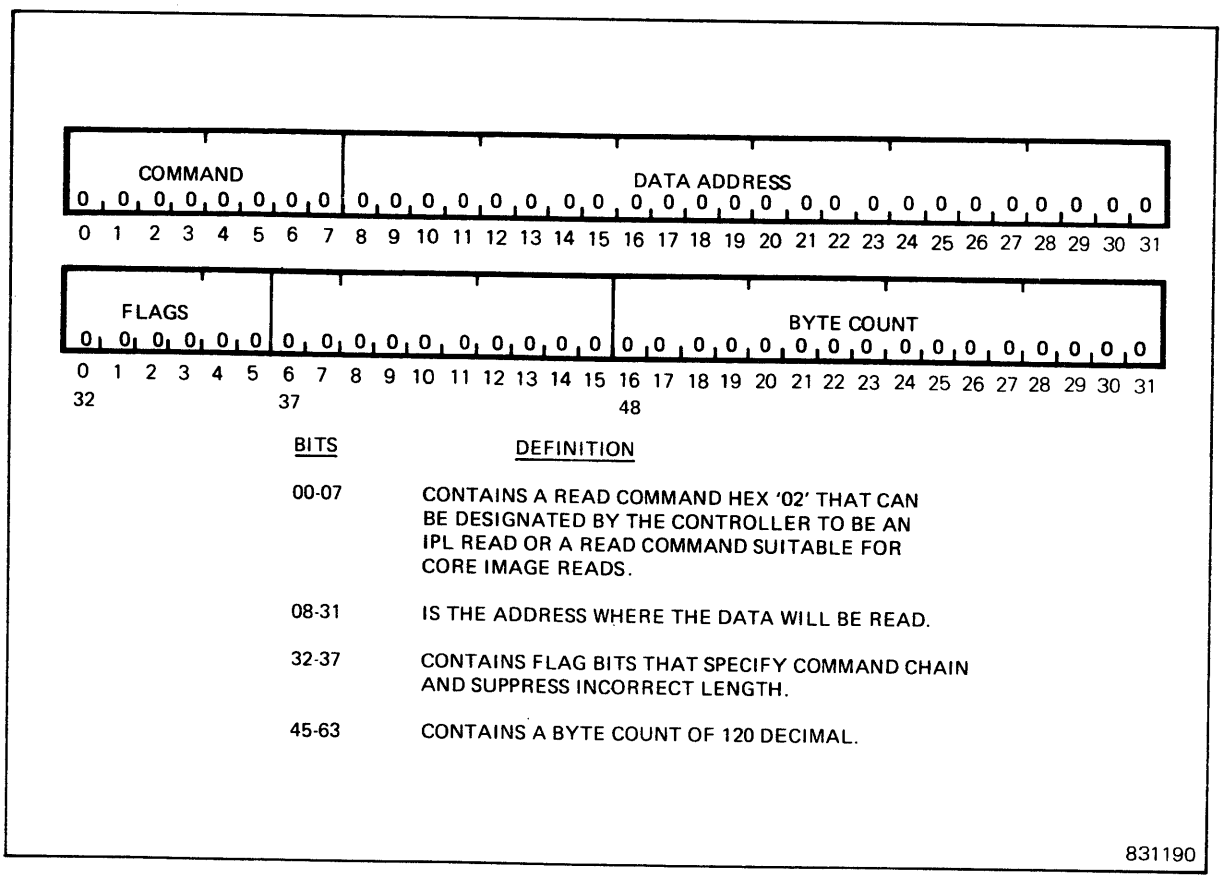


Figure 3-5. IOCD Generated by CPU for IPL

While the channel continues with the execution of the command list, the CPU will recursively scan byte locations 4 through 7 of main memory for a change of state. When byte locations 4 through 7 change from hexadecimal 60000 to some other value, this 60000078 condition indicates that the channel has stored the IPL new PSD. Now the CPU will issue an enable interrupt to the channel to indicate that the channel can store status at any time. Testing of locations 4 through 7 is required because the channel will store the final status for the IPL I/O operation into locations 0 through 7 when the CPU issues an enable interrupt to the initial program loading channel. When the I/O interrupt occurs, the CPU transfers control to the software indicated by the new PSD.

3.4.1 IPL Initiated from Cartridge Module Disc Drive

Whenever an IPL sequence is initiated from the cartridge module disc drive (CMD), the following events will occur:

1. A device rezero command is issued to the CMD. The rezero command causes the removable media to be selected by hardware default.

2. The UDP then reads a label (any label) from the CMD. This label indicates the number of heads configured and the number of sectors per track (16 or 20). Once the number of sectors per track is defined, the IOCD is accessed from main memory and control is then transferred to the op code decoder.

3. At this point, a normal data transfer process takes place. Control is dispatched to the read command processor, which will detect whether the drive is on target and compute a target value for cylinder track and sector. Since the IPL IOCD indicates a transfer count of 120 command chained bytes, these 120 bytes must contain input/output command doublewords (IOCD), which will bring in the system or, at least, enough of it to bring in the rest of the system.

4. If the user wishes to load the initial program from the captive media, the following microcode function will occur:

    a. Rezero disc and read a sector label, as previously defined.

    b. If the device is a cartridge module drive and an odd subchannel is selected (subchannel 1), the mode register is internally set to the captive media mode.

    c. All even subchannel I/O during IPL is taken from the removable media, and all odd subchannel I/O during the IPL sequence will be taken from the captive media. The system programmer should note that the odd subchannel will remain selected to the captive media until the user selects the removable media via a load mode register or reset channel.

3.5 Status Presentation

3.5.1 Condition Codes

Status returned from the channel is understood to be a subchannel status, associated with a specific subchannel. Subchannel status is routinely presented to the CPU. Status responses take two forms: a condition code setting and a status doubleword.

A condition code is always generated in response to a CPU instruction. The condition code may or may not be accompanied by a status doubleword. The channel response and condition codes are shown in Figure 3-6.

3.5.2 Status Doublewords

The channel maintains a 66-word status queue (STQ) located in the channel main memory buffer. This queue acts as a ring queue to maintain 33 doublewords of subchannel status. Status is placed on top of the stack by the channel and read from the bottom of the stack by the CPU. Thus, the queue is maintained on a first-in, first-out basis.

Status is formatted in the STQ in a doubleword format shown in Figure 3-7. Word 1 identifies the subchannel and contains the subchannel IOCD address contents that existed at I/O program termination. Word 2 presents 16 bits of status and the residual byte count. When a two-part final status presentation occurs, the two parts will differ only in the status bits presented.

When a status-generating event occurs during I/O program execution, the appropriate status bits are generated and placed in the subchannel image (SI) status register. If the status contains channel end (CE) or device end (DE), or both, a status doubleword is generated and placed on the STQ and interrupt polling is initiated. Once the status is queued, the subchannel is deactivated by storing the SI in subchannel storage (SST). The status is saved, since it was recorded in the SI. The subchannel remains busy until both CE status and DE status are queued and passed to the CPU.

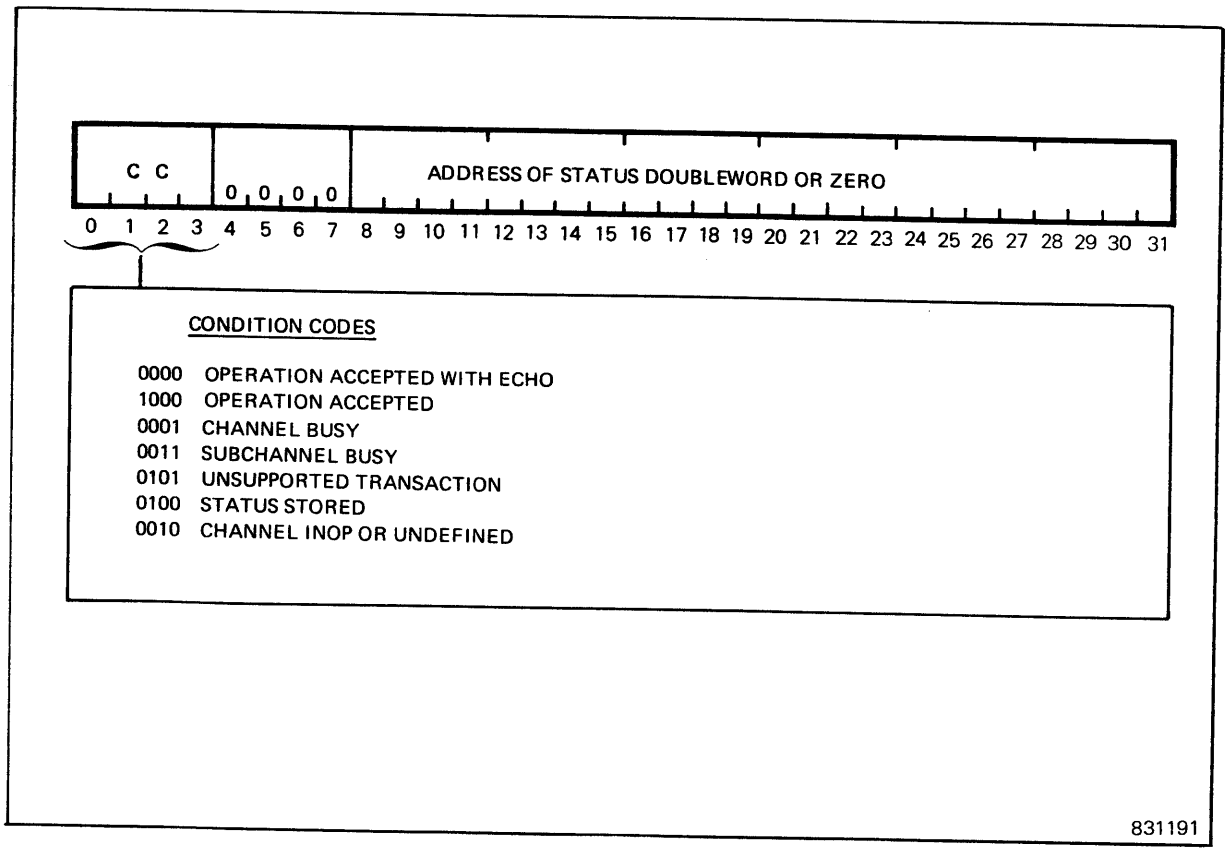
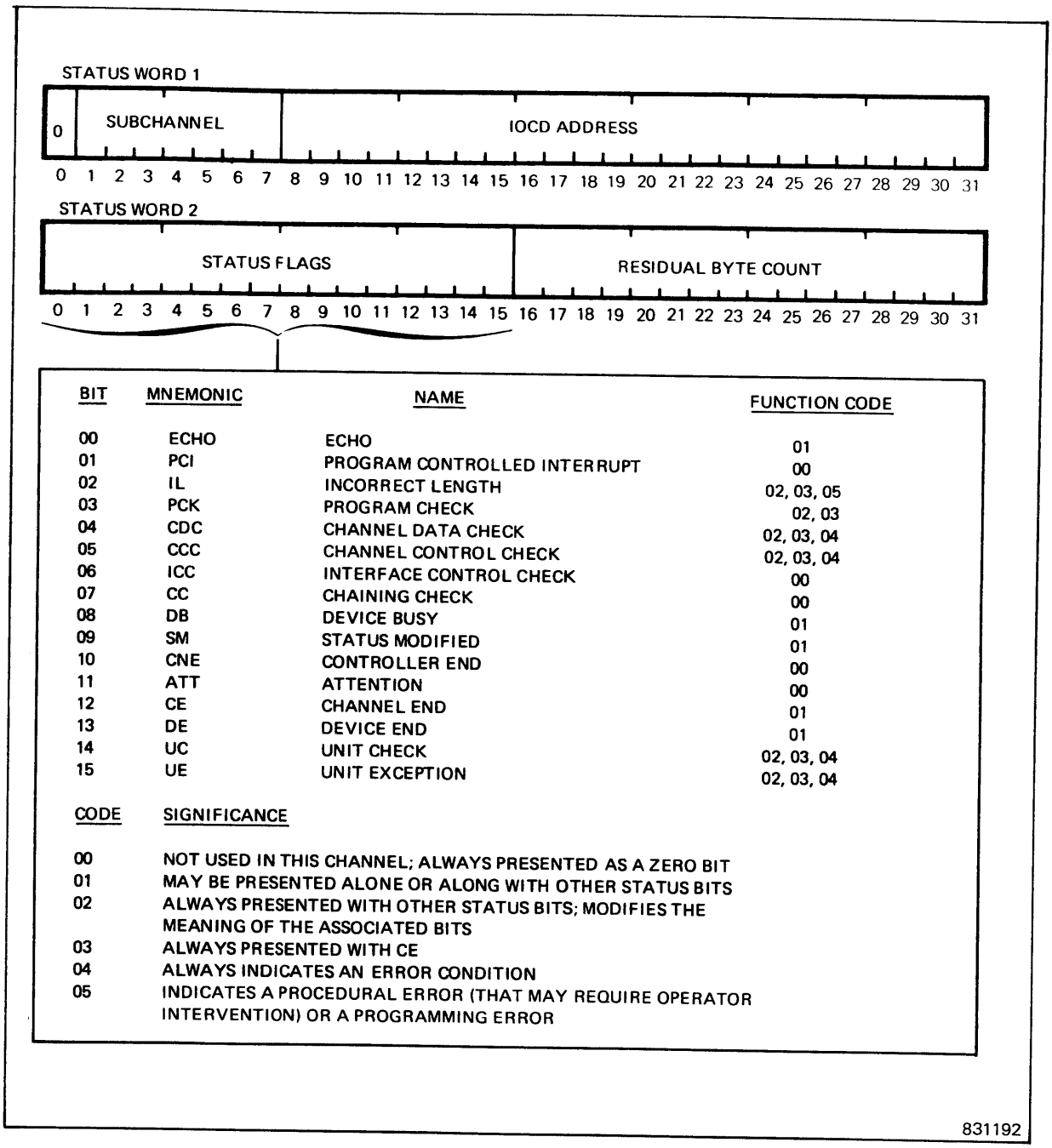


Figure 3-6. Channel Condition Code Response



BIT	MNEMONIC	NAME	FUNCTION CODE
00	ECHO	ECHO	01
01	PCI	PROGRAM CONTROLLED INTERRUPT	00
02	IL	INCORRECT LENGTH	02, 03, 05
03	PCK	PROGRAM CHECK	02, 03
04	CDC	CHANNEL DATA CHECK	02, 03, 04
05	CCC	CHANNEL CONTROL CHECK	02, 03, 04
06	ICC	INTERFACE CONTROL CHECK	00
07	CC	CHAINING CHECK	00
08	DB	DEVICE BUSY	01
09	SM	STATUS MODIFIED	01
10	CNE	CONTROLLER END	00
11	ATT	ATTENTION	00
12	CE	CHANNEL END	01
13	DE	DEVICE END	01
14	UC	UNIT CHECK	02, 03, 04
15	UE	UNIT EXCEPTION	02, 03, 04

CODE	SIGNIFICANCE
00	NOT USED IN THIS CHANNEL; ALWAYS PRESENTED AS A ZERO BIT
01	MAY BE PRESENTED ALONE OR ALONG WITH OTHER STATUS BITS
02	ALWAYS PRESENTED WITH OTHER STATUS BITS; MODIFIES THE MEANING OF THE ASSOCIATED BITS
03	ALWAYS PRESENTED WITH CE
04	ALWAYS INDICATES AN ERROR CONDITION
05	INDICATES A PROCEDURAL ERROR (THAT MAY REQUIRE OPERATOR INTERVENTION) OR A PROGRAMMING ERROR

Figure 3-7. Status Doubleword Format

3.5.2.1 Final Status Presentation

As indicated above, the termination of an I/O program requires the presentation of final status containing CE and DE status; the allowed presentations and their descriptions follow. The status may be presented via interrupts or in response to the CPU instructions SIO and TIO.

Regardless of the mode of presentation, all final status presentations will occur in either one or two transactions and consist of one or two of the defined status responses. When

the CPU executes an SIO or TIO instruction, that instruction will be directed to a particular subchannel. Should the channel have pending final status from that subchannel or any other subchannel, status will be presented even though it is not from the specified subchannel.

The presentation of SR1 status in response to SIO, TIO, or HIO, does not indicate that the status is not pending in the channel. Status may be available in the channel and load conditions may temporarily preclude its transfer to the CPU.

3.5.2.2 Program Controlled Interrupts (PCI)

The universal disc processor does not support program controlled interrupts. Setting the PCI bit in any IOCD causes I/O program termination with a program check when the PCI flag is encountered.

Although program controlled interrupts are not implemented, a similar capability exists. Whenever a subchannel executes a command or data chaining operation, the SST IOCD address field is updated to indicate the next IOCD address to be used. This update occurs independently of the SI filling and storing operations that occur during subchannel activation and deactivation. Since the SST is in a main memory allocation, the state of execution progress of any subchannel is always in the SST and is accurate to the execution span of one IOCD. The SST contains other information of value in tracking I/O program execution. However, this information is updated only when a subchannel is deactivated.

3.6 Programming Guidelines

The following paragraphs provide a variety of programming suggestions for software programmers. Specific examples are given to aid in understanding the programming.

3.6.1 Initialization

3.6.1.1 Memory Buffer Allocation

The channel requires a 224-word allocation in main memory to be used as auxiliary registers by the channel. This memory buffer consists of 128 words of subchannel storage (SST), 66 words of program status queue (PSQ), 26 words of scratchpad, and 4 words of label buffer register. The layout and format of the memory buffer allocation is described and illustrated in Chapter 4.

3.6.1.2 Channel Initialization

Prior to any I/O requests to a given disc drive, the channel must be properly initialized, which requires presenting to the channel a nine-word block of initialization data, as shown in Figure 3-8. The first entry in this block is the address of the first entry in the memory buffer allocation previously described. Following this address, there must be eight 32-bit words that describe the disc and controller types. These drive state registers define disc parameters to the channel.

The UDP also supports an alternate initialization sequence. This alternate sequence allows for a separate channel initialization and controller initialization. Refer to Figure 3-9.

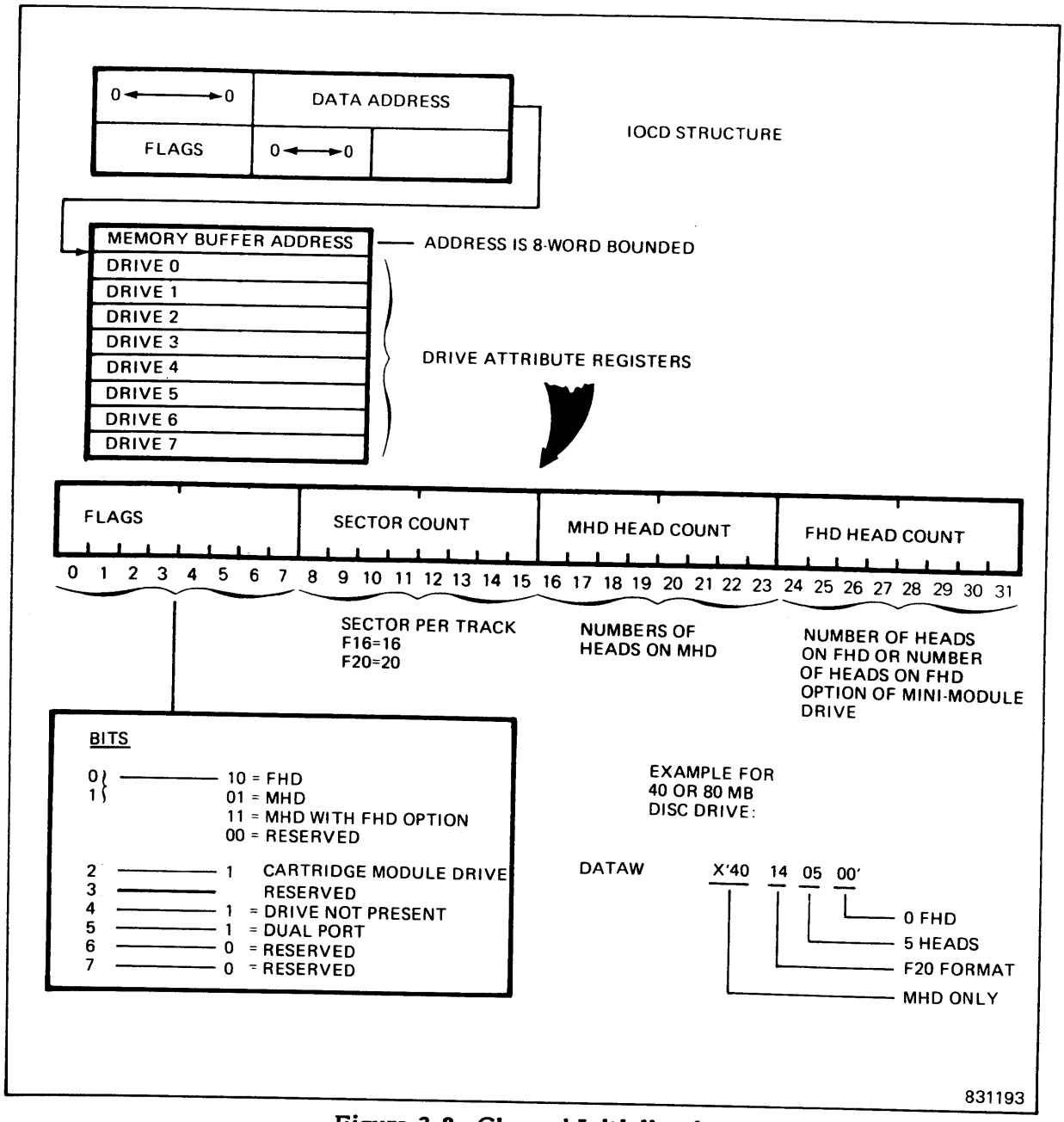


Figure 3-8. Channel Initialization

3.6.1.3 Disc Format Initialization

The UDP is capable of supporting two disc formats: F20 = 20 sectors, each 768 bytes in length, and F16 = 16 sectors, each 1024 bytes in length. Refer to Chapter 4 for a complete description of the disc formats. Figure 3-10 is an example of how to format an entire disc drive.

Note that it is unnecessary for the user to update the disc labels with the correct cylinder, track, and sector information or the drive attribute register during disc initialization. This information comes from the channel.

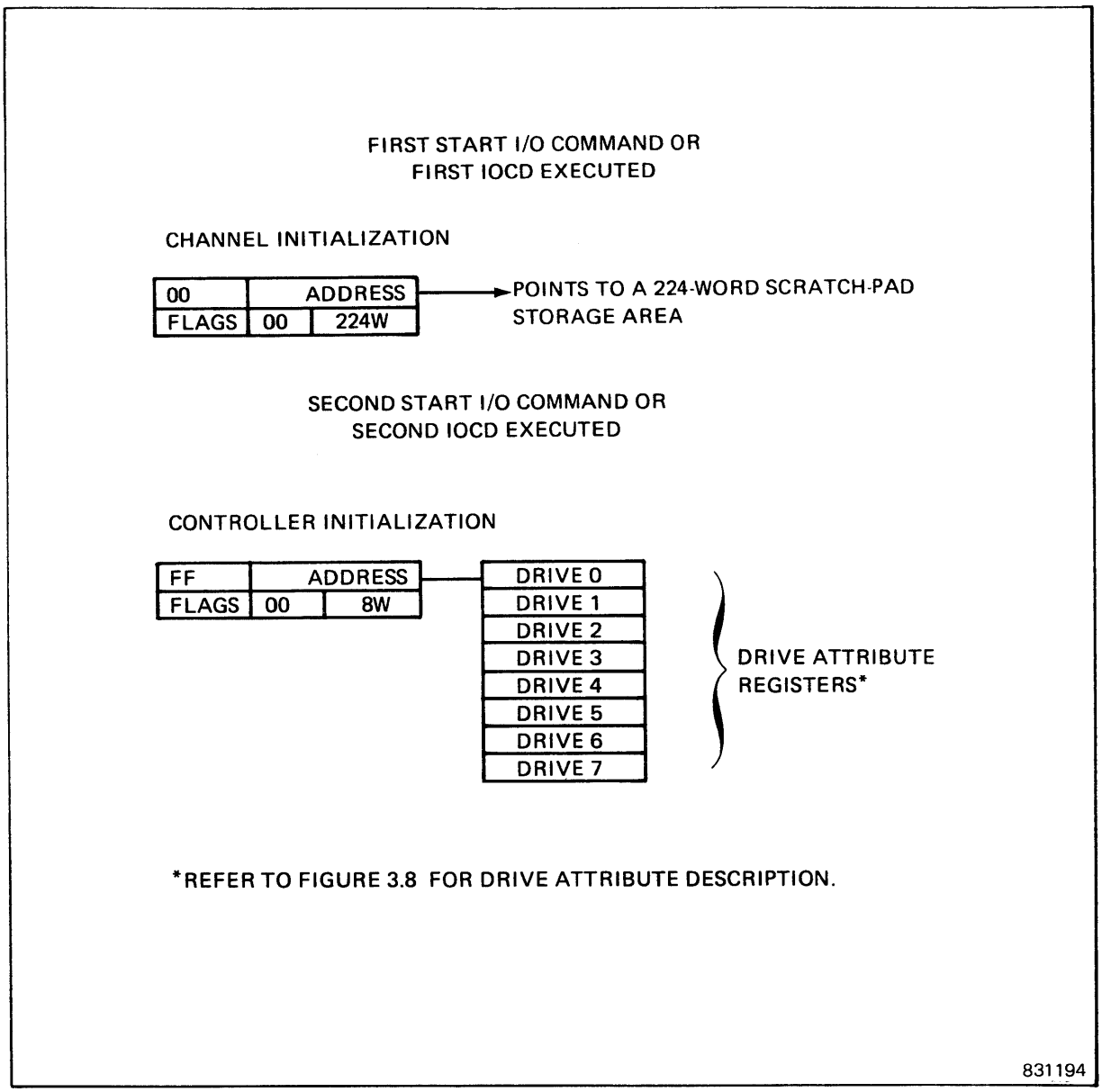


Figure 3-9. Alternate Initialization Sequence

3.6.1.4 Configuration Jumpers

The DIA-1 card contains a set of eight configuration jumpers. These jumpers are shown on logic drawing 130-103282, sheet 4, and identified by the callouts G02-1 through G02-8. These jumpers are used to establish the system configuration to the CPU. The configuration jumper conditions are shown in Table 2-1 and Figure 3-11. The function is true when the jumper is connected.

Jumper positions 0 and 1 are not used.

Jumper position 2, when in place, prevents any label that is write locked from being altered by the software. This rule applies to all drives configured on this controller.

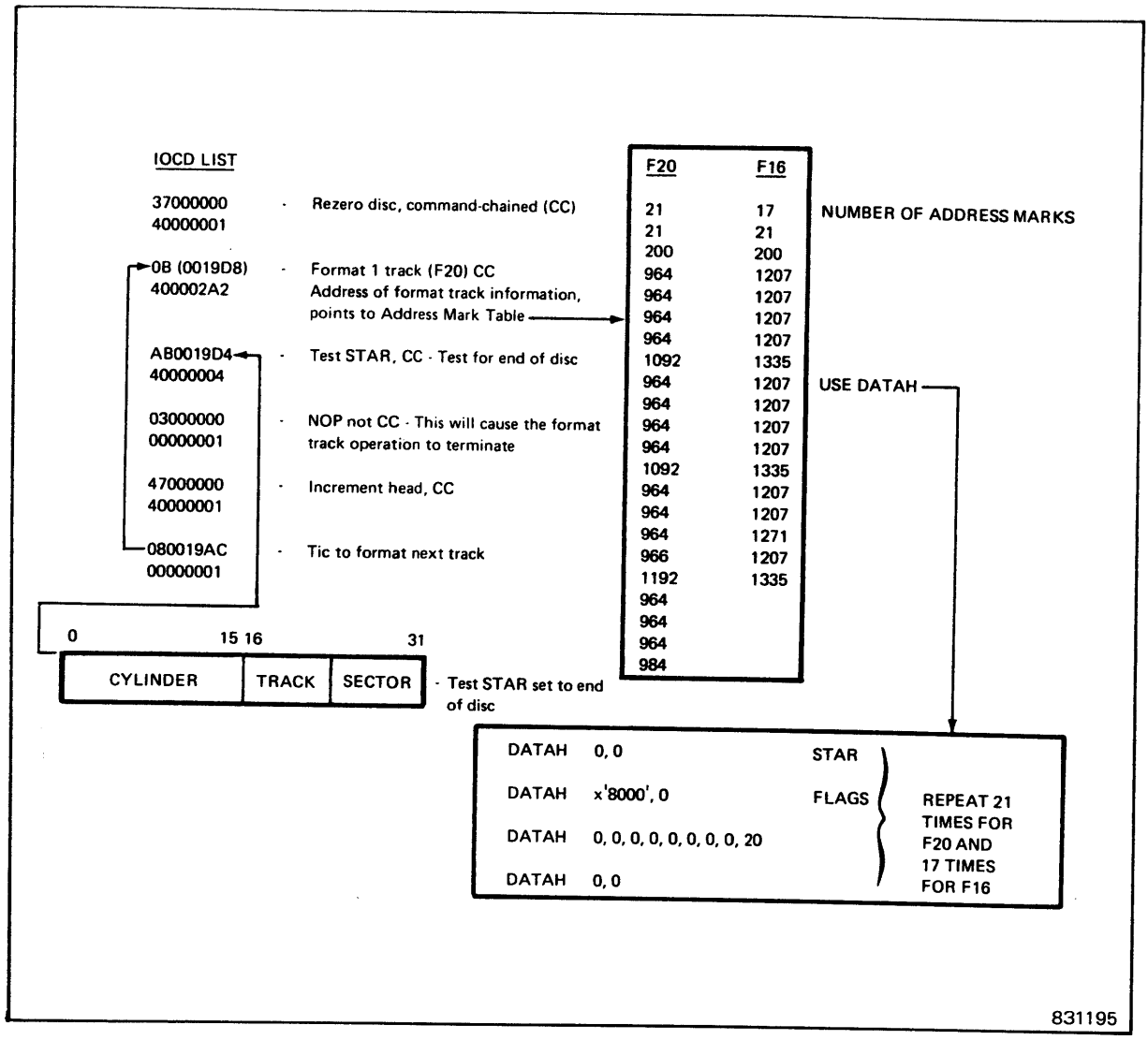


Figure 3-10. Sample IOCD List to Format a Complete Disc

Jumper position 3, when in place, allows the system programmer the option of programmatically write locking any or all disc drives on the disc system. When the system is booted in (after a system reset), all labels may be read or written regardless of the state of the write lock flags within the label. Once a lock protected label (LPL) command is issued, all labels for that drive that specify locked will be protected by the disc firmware. If all drives are to be protected, the LPL command must be issued to each subchannel.

Jumper position 4, when in place, allows any user in the reserve track mode (as defined by the mode register) to access any area of the disc that is defined by a sector label as being a reserved track. If this jumper is not in place, access of a reserve track is not permitted.

Jumper positions 5 and 6 are not used.

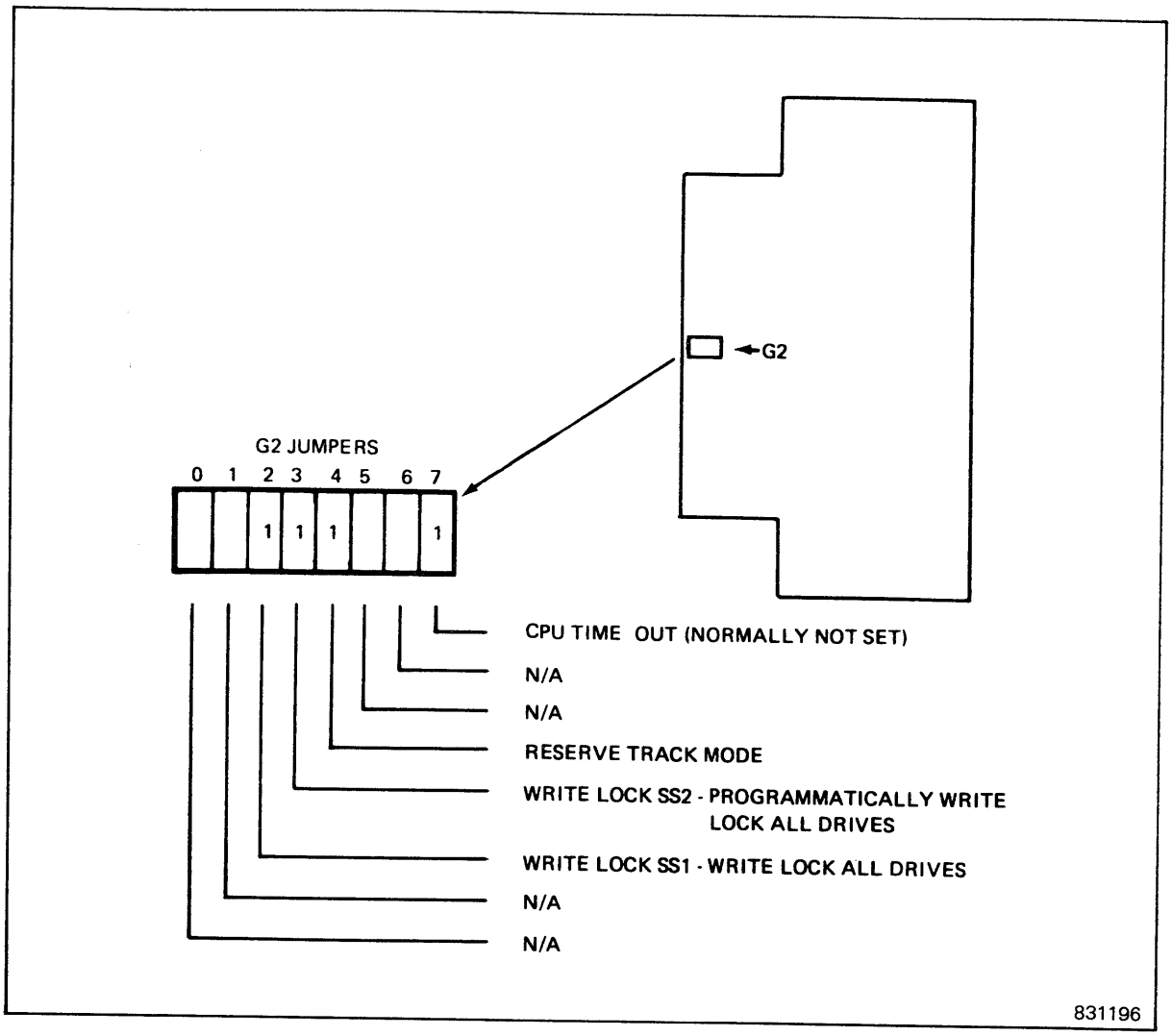


Figure 3-11. Configuration Jumpers

Jumper position 7 is a CPU timer disable switch, which is used to prevent overrun during strip label reads. Note that normal I/O operation is such that this jumper should not be set. Setting jumper position 7 will allow overrun conditions during strip label operation in a multidisc environment. Setting of this jumper ensures that the UDP will never lock out the CPU for more than 38.4 microseconds. With the jumper reset in the multidisc environment, the CPU may be locked out for a maximum of 60.3 microseconds. If the CPU firmware is not at a level of C75R1 or greater, then the user should insert jumper 7.

3.6.2 Routine Operations

3.6.2.1 Disc Seek Operation

The normal seek operation selects the disc drive. In a dual-ported disc system, when the drive is currently attached to the opposite channel, the seek operation will be suspended until the drive becomes available, at which time the seek will take place.

The normal seek is as follows:

IOCD 07001CD4 Seek - Byte count = 4, CC

400000004

CYLINDER	TRACK	SECTOR

If the user specifies a byte count greater than four and sets suppress incorrect length (SIL), the seek will take place normally. If the byte count is equal to three, only cylinder and track will be selected (SIL must be set). If the byte count is equal to two, only seek cylinder will take place (SIL must be set). If the byte count is equal to one, then a device rezero and clear fault will be executed (SIL must be set). Any byte count less than one will result in a program check error.

3.6.2.2 Write Data Operation

The normal write data command list is usually constructed with the following commands:

IOCD - Seek, command chained
IOCD - Write data

The seek command will perform disc drive selection, seek cylinder, and head selection. Once the drive comes on cylinder, a sector target is computed based on the requested sector number and whether the command is a read or write. After a target has been computed, all control is returned to the microcode executive processor, which will dispatch control to the proper command processor for whichever drive comes up on target first.

The write data command reads the preceding label and determines if the drive is on target, not write protected, etc., and begins the data transfer process. This process will continue across head and cylinder boundaries until the byte count goes to zero. If the byte count size is not a multiple of an even sector transfer, then the remainder of the last sector is padded with zeros. Once the byte count has gone to zero, a test for data chaining is performed. If data chaining is specified, then the next IOCD is acquired from memory and the data transfer process continues normally.

It should be noted that in all data chaining sequences for both read and write, the programmers need not concern themselves as to where the data chaining points are on a disc sector. Also, the user may read or write as little as one byte of data to the disc.

IOCD Structure

07001CD4 Seek, CC, byte count = 4
40000004
01002308 Write data, DC, byte count = x'1F4'
800001F4
010024FC Write data, byte count = x'10C' (not command chained)
0000010C

3.6.2.3 Read Data Operation

The normal read data command list is constructed with the following commands:

IOCD - Seek, command chained

IOCD - Read data

This seek command performs disc drive selection, seek cylinder, and head selection.

Once the drive is on cylinder, a sector target is computed, based on the read command and the sector number requested. Once the drive target has been computed, control is returned to the disc controller executive where control is then dispatched to the proper command processor for the drive, which is on target.

Once the read command processor begins execution, a test is made to determine if any data is to be skipped (is the skip flag in IOCD set?). If the skip flag is set and data is to be skipped, a test is made to determine if data chaining is set. Once the specified number of bytes has been skipped, a new IOCD will be acquired from memory and the reading of data to memory begins. The user should note that more than a full sector may be skipped, which would require reading the sector being skipped. In most cases, it would be considered a good programming habit to skip only partial sectors of data.

Read data chained is considered an extension of skip read, because both types of reads must be data chained. When all of the requested data has been transferred to memory, channel end (CE) and device end (DE) states are presented to the user.

IOCD Structure

07001CD4	Seek
40000004	
02000000	Read, skip, DC
9000010C	Byte count = x'10C'
02002308	Read, byte count = x'1F4'
000001F4	

3.6.2.4 Label Operation

The labels on the disc contain the control information required to operate the disc system. Within the label are flag bytes that allow the protection of data on the disc. Refer to Chapter 4 for a description of the labels. There is another bit that prevents the labels from being changed. Since it is always necessary to read the previous label on a label write operation, we find that the labels protect the data. A previous label protects the current label, and the track label, in turn, protects all the labels of the track.

If the user wants to protect his data, he need only read the sector label for a given track, set the write protect bit, and then write those labels to the disc. Any attempt to write on this sector of the disc results in a program violation, and the data remains unchanged.

If the user wants to ensure that no one will change the labels that would then allow the data to be altered, the user must also set the lock protected label flag in the label and perform a similar function to the track label, in that order. If the proper jumper switches are set and a user tries to alter the data or labels, the I/O operation will be

terminated with a program violation. If the proper jumpers were not set, any user could alter the labels and then alter the data.

The proper procedure for unlocking a track of data follows:

1. Reset proper jumpers on disc controller board.
2. Reboot system.
3. Read track label and reset write locked flags, etc.
4. Write track label to the disc.
5. Read all sector labels and reset proper bits in flag register.
6. Write sector labels to the disc.

The user should note that all label boundaries must be on a halfword boundary and data chaining is not supported.

3.6.2.5 Track Relocation

It is suggested that when a disc is formatted a few spare cylinders be left unused to accommodate defective tracks as they might occur. Once a track is defined as being defective and no longer usable to the system, it may be flagged as defective and a spare track may be assigned as an alternate track for the defective one.

The procedure for assigning alternate tracks follows:

1. Generate N label prototypes when N = 20 or 16 F20/F16 format.

2. Update the label with the alternate cylinder and track number. Write these labels to the disc.

3. Perform the same function with the track label.

4. Seek to the alternate track and write labels that indicate alternate track; in the alternate track portion of the label, write the cylinder and track address of the defective track. The user should note that the defective track points to the alternate track, and the alternate track points to the defective track.

3.6.2.6 Error Recovery

When the user initiates an I/O operation, the channel responds with some type of status as a result. If this status contains channel end and device end status only, then the user can assume that the I/O took place without errors. If an error is indicated, the user should initiate a sense command as the next command to determine the exact error type. The following is a brief list of the device status and the action that should be taken:

x'100C' PGM-CK, DE, CE

1. Invalid OP CODE. Command reject. Correct user program OP CODE.
2. Invalid cylinder address. Seek out of bounds; improper head selected.
3. Address boundary error (label not halfword-bounded, etc.)
4. Nonpresent memory error. Correct user program.
5. Write protect error; don't try to write to protected areas.

6. Mode register check (CK). The user attempted to write to a reserved mode track and mode register not properly set.

7. Chain error. The user has specified data chaining for a command that does not support data chaining, or the user should have set data chaining based on IOCD flags. User program must correct chaining problem.

x'300C' PGM=CK, ICL, DE, CE

This error indicates that the user has specified an incorrect byte count. Example: labels are not 30 bytes in length, seek command is not 4 bytes, etc.

x'000E' DE, CE, UC

1. Seek error, device error. The drive executed a seek to the incorrect cylinder, and the label verification indicated seek error. The user should rezero the disc. If the error continues, the drive may have problems.

2. Disc overrun. An overrun condition may occur during times of heavy activity and extensive data chaining. The user should restart the I/O operation.

3. Drive lost. This error indicates that the drive is dual-ported and is unexpectedly on the opposite CPU. The user should restart the I/O operation.

4. Drive lost. When the drive is not dual-ported, the drive may be powered-down (intervention required). Power-up drive and restart the I/O operation.

5. Release fault. When a drive is dual-ported and a release command is executed to a device that is not reserved, the device will be reserved when the opposite channel releases the drive. The proper recovery for this condition is to execute a reserve command followed by a release command.

x'000C' CE, DE, UE

1. This error indicates that a disc format error (incorrect labels, drive not properly initialized, F16 disc pack on an F20 initialized drive, etc.).

2. This error also indicates that a user has attempted to perform an I/O operation to a drive that is not configured in the disc system as defined by the initialization label.

x'040C' DE, CE, channel data check

This error indicates a memory parity error.

x'000F' CE, DE, UC, UE

This error indicates that an ECC error has been detected, and the sense data indicates where the error occurred: track label, sector label, or data. If there are ECC errors in the labels, then the labels must be recreated.

x'000C' GOOD STATUS, continue

NOTE

In all error conditions, the user should read the sense data to determine the specific type.

CHAPTER 4

THEORY OF OPERATION

4.1 Introduction

The universal disc processor (UDP) is shown in the overall block diagram Figure 4-1. The channel (UDP), indicated by the dashed lines in the figure, consists of three circuit cards, the regional processor (RP), and two device interface adapter (DIA) cards. All three cards are installed in the logic chassis. The RP card is plugged into the front side of the SelBUS, while the other two cards are plugged into the back. DIA-1 is required in all system configurations; DIA-2 is an optional card that is required when five or more (up to eight) drives are installed.

4.2 General Theory

4.2.1 Purpose

The purpose of the UDP is to provide high-capacity, random-access storage utilizing a variety of fixed- and moving-head disc drives, intermixed in a single system. Operation is initiated under software control from the CPU. Once operation is initiated by the CPU, the UDP takes over control and executes the data transfers to or from the disc drives without further intervention by the CPU. Up to 16 separate programs can be executed simultaneously by the channel. Each of these separate programs is called a subchannel. At the completion of a task, the subchannel sends an interrupt to the CPU and presents status. The channel is then ready to accept a new task.

4.2.2 Basic Organization

4.2.2.1 System Components

The UDP consists of four types of components: three circuit cards and a disc drive. The three circuit cards are the regional processor, the device interface adapter 1 (DIA-1), and the device interface adapter 2 (DIA-2). All three cards are plugged into the main logic chassis.

The regional processor is a general purpose microprogrammable input/output controller. By the addition of device interface (DI) hardware and a permanent, internal control program (firmware), the regional processor can be made to control the disc drives without routine intervention by the CPU.

The DIA-1 card is utilized to connect from one to four discs to the RP. The DIA-1 contains the primary device-dependent logic, including a 256-byte RAM for temporary storage of read-write data, input and output registers, and a microsequencer to control data transfer. DIA-1 plugs into the SelBUS backplane directly opposite the RP. The card is directly connected to the RP via connectors J1A and J1C. In addition, a daisy-chained A cable and a set of radial B cables connect DIA-1 to the supported disc drives.

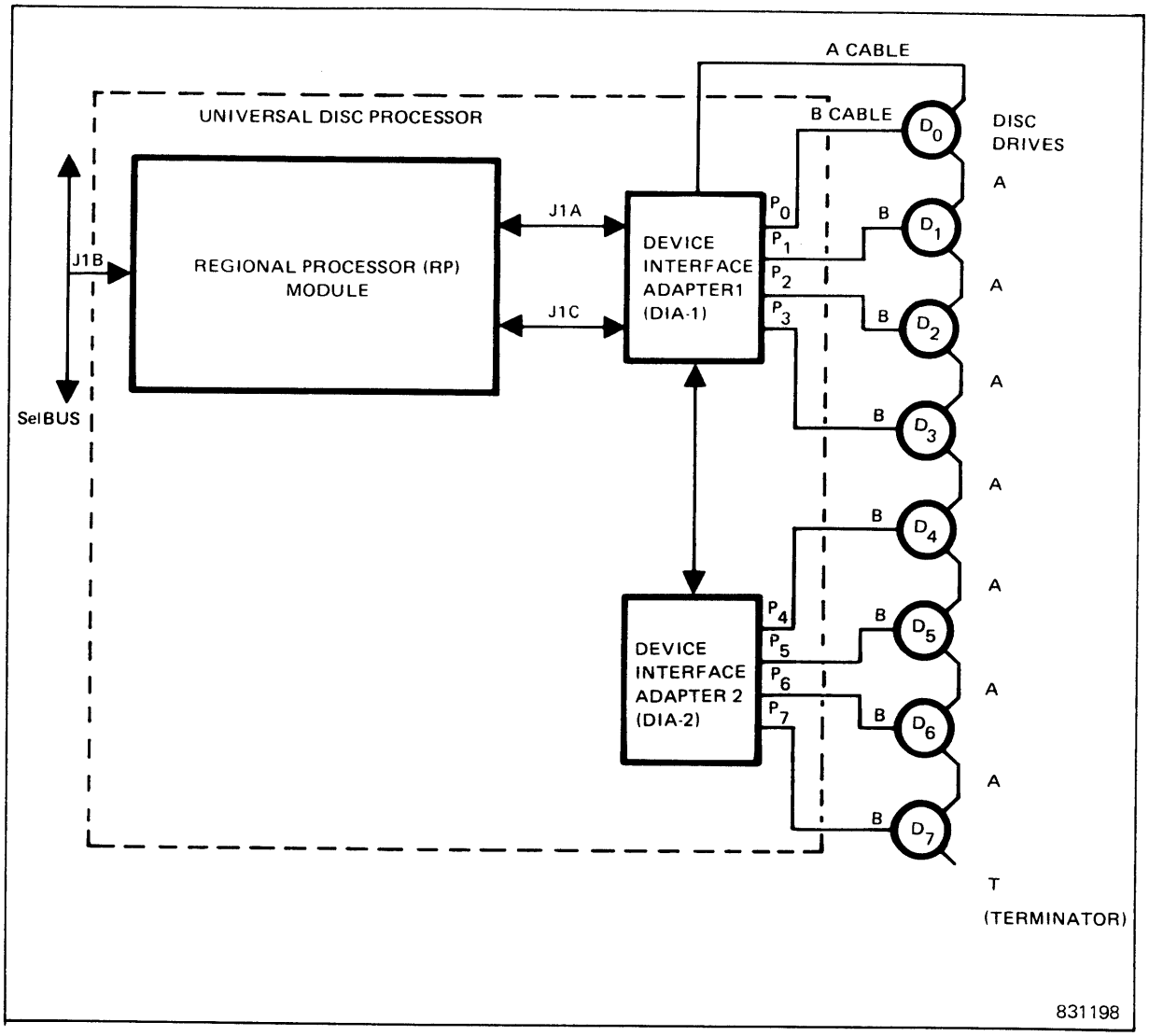


Figure 4-1. Universal Disc Processor Block Diagram

DIA-2 is an optional card used to expand from five to eight the number of disc drives that can be connected to the disc processor. This card is essentially a port expander, adding four more disc ports to the system. The card primarily contains cable drivers and receivers associated with a particular port. The card is connected to the disc drives by the radial B cables and to the DIA-1 card by another pair of cables.

4.2.2.2 System Firmware

In addition to the hardware components described above, the UDP contains two sets of firmware to control system operation. One set of firmware resides in the ROM storage in the RP. This firmware controls the overall operation of the RP's microprocessor. Specifically, this firmware controls the receipt and processing of CPU commands, processing I/O commands, transfer of data to and from memory, and transfer of data to and from the DIA-1 card.

The second set of firmware resides in the DIA-1 PROM storage. This set of firmware controls the operation of a microsequencer located in the DIA-1 card. The purpose of

the microsequencer is to control data transfer between the RP and the disc drive. The microsequencer firmware is considerably smaller and simpler than the RP firmware, since its function is more limited.

4.2.2.3 Overall Operation

In overall operation, the UDP is a block multiplexer channel capable of executing 16 I/O programs concurrently. That is, the channel can handle two programs each on the eight disc drives. The UDP attempts to maximize the data transfer rate by utilizing head seeks and rotational delays to process other discs. Data can only be read from or written to one disc drive at a time. However, the relatively large rotational delays inherent in a disc system provide ample time for the execution of other programs while waiting for disc rotation or head seek. The UDP contains a complex scheduling system that multiplexes the various disc transfers to take advantage of time normally lost during disc rotation.

In operation, the CPU first initiates the channel, causing the UDP to build a memory buffer in main memory. This buffer is used to store both channel and subchannel status. An individual subchannel can be initiated with a start I/O command addressed to the subchannel. Up to two subchannels can be initiated for each disc drive connected to the system.

The UDP firmware contains an executive routine that scans each active subchannel and makes scheduling decisions to maximize data transfer. The basic criteria for scheduling is a sector targeting scheme within the UDP. In this scheme, the device-dependent logic for each disc drive keeps a running tally of the sector count. Periodically, the UDP firmware scans the sector counters and compares the values of the sector count to the targeted sector of each subchannel. The executive routine selects the subchannel whose target value is closest to the sector counter for that disc. Upon completion of the data transfer I/O commands, the executive routine again scans the subchannels and selects the next subchannel for execution.

Data transfer is executed through the cooperative efforts of the RP microprocessor and the DIA-1 microsequencer. The RP microprocessor decodes and interprets CPU commands, handles all SelBUS transfers, and provides instructions to the DIA-1 microsequencer. Data from memory to be written on the disc are transferred by the RP in 32-bit word format. The RP breaks this word into two halfwords (16 bits) for transfer to DIA-1. DIA-1 further breaks the halfwords into bytes for storage in a FIFO queue. Then the bytes are converted to a serial bit stream and transferred to the disc.

Data read from the disc operates in reverse. The serial bit stream is converted to bytes by the DIA-1 microsequencer. The bytes are assembled into halfwords and passed to the RP. The RP then assembles the halfwords into 32-bit words and transfers them to memory.

The DIA-1 circuit card contains a 256-byte FIFO RAM, which is used as a data buffer. The RAM is a circular queue used to store, read, or write data temporarily. The purpose of the FIFO is to prevent data overruns or underruns that could result from delays inherent in the normal operation of the SelBUS. During a read operation, data is stored, one byte at a time, in the buffer. The data is removed from the buffer in a first-in, first-out basis.

The main memory buffer previously mentioned contains a 128-word allocation for subchannel status (SST). The SST contains 16 eight-word entries, one entry for each subchannel. Each SST entry contains all the necessary information to allow the RP

firmware to operate the subchannel. When actual processing is to occur for a given subchannel, the SST entry for that subchannel is fetched and loaded in temporary registers in the RP. At the end of data transfer, the SST entry is updated and returned to its SST address in main memory for future reference.

The RP executive constantly scans the subchannels and selects one for processing. Upon completion of that portion, the SST entry is updated and the executive routine selects the next subchannel for data transfer. In this manner, the executive routine cycles through all the active subchannels.

4.3 Detailed Theory

The detailed theory description is divided into four topics: the regional processor, device interface adapter 1, device interface adapter 2, and UDP firmware. A description of the UDP functional characteristics follows this chapter.

The following description provides logic drawing references to locate the logic components on a specific logic drawing sheet. The logic drawing references are intended to orient the user to the overall organization of the logic drawings. Three sets of drawings are necessary to cover the complete UDP. Drawing 130-103623 covers the RPU, drawing 130-103282 covers the DIA-1, and drawing 130-103291 covers DIA-2.

The 130-103282 and 130-103291 drawings are found in the drawings manual. The conventions required to read the drawings are provided in the circuit registration manual, which also provides a complete description of all intergrated circuits used in the system.

4.3.1 Regional Processor

The regional processor (RP) is thoroughly covered in the Regional Processing Unit User Reference and Design Manual, publication 310-000430, which is included in the peripheral device documentation package. The following paragraphs are only a general description of the RP to help orient the reader. Figure 4-2 shows the regional processor as a part of the overall UDP system. Figure 4-3 is a block diagram of the RP itself.

The regional processor (RP) contains a SelBUS interface logic, a microprogrammable processor, and an internal storage capacity for as many as 8192 32-bit words of microprogrammable read-only memory (ROM). The RP directly interfaces to the SelBUS and the device interface adapter modules. The primary function of the regional processor is to control the execution of input and/or output data transfers (writes and reads) from an external peripheral device after the operation has been directed by the CPU software. Once the CPU has initiated the operation, the RP executes the data transfers between memory and the addressed peripheral device, independent of CPU operations. Prior to the initiation or termination of any input/output operation, the RP determines its status and that of the addressed peripheral device; upon receipt of a status request from the controlling CPU, the RP responds with the assembled status information, as requested.

4.3.1.1 SelBUS Interface Logic

The SelBUS interface logic is controlled by the microprogram (firmware), which initiates activities of the regional processor by directing the functional logic. The functional logic generates the required control signals to the SelBUS interface, as commanded

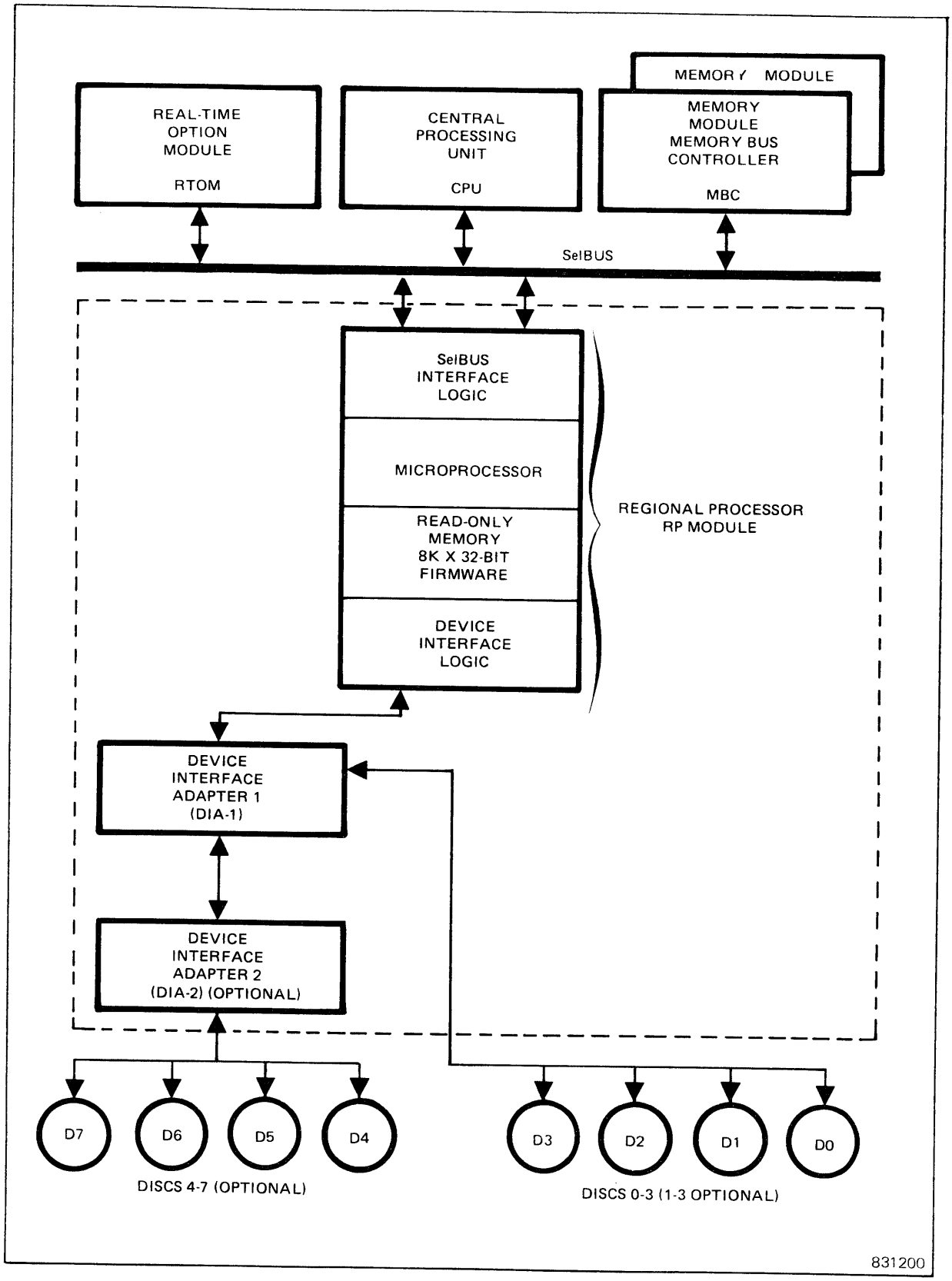


Figure 4-2. Regional Processor Functional Diagram

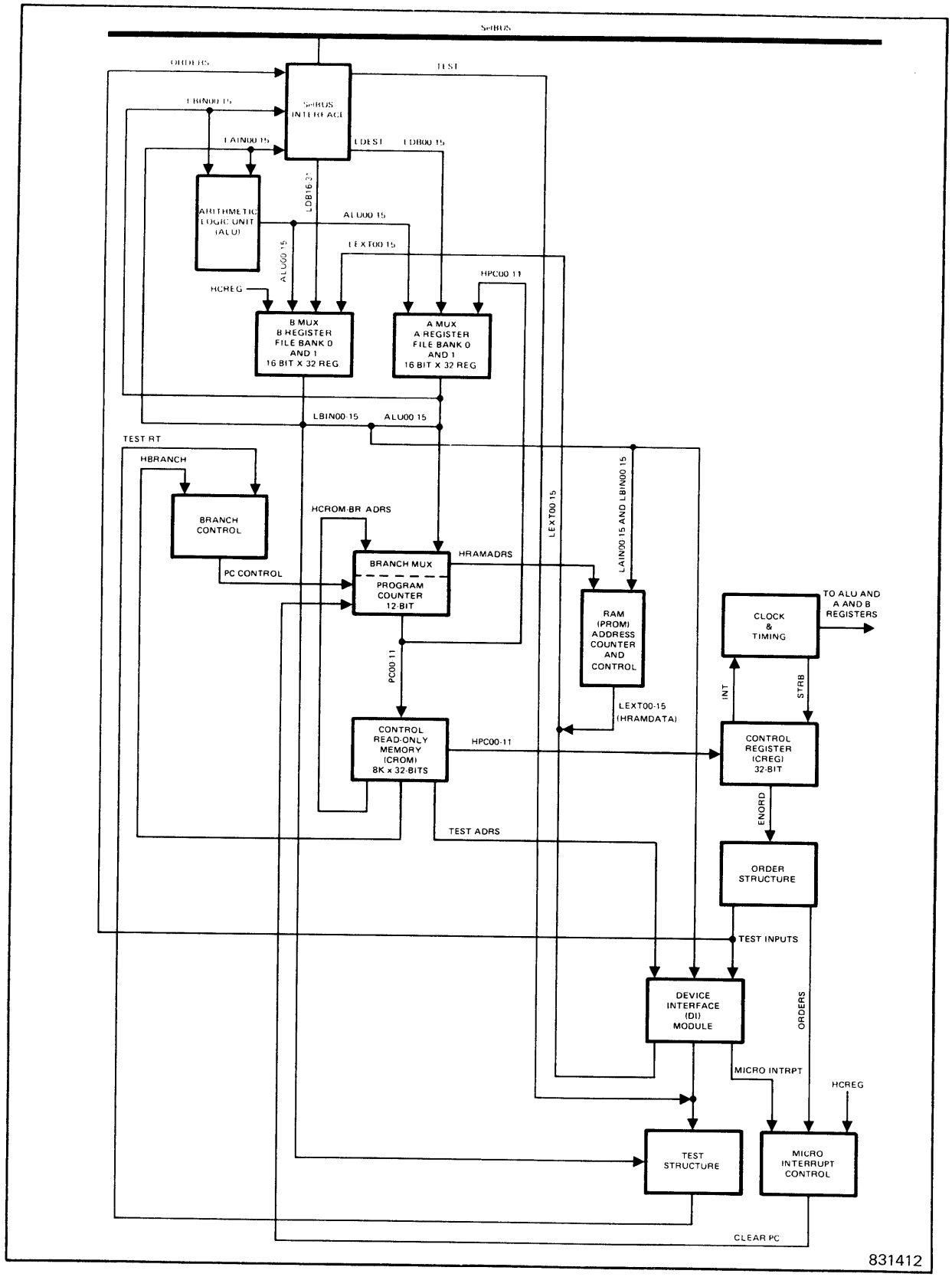


Figure 4-3. Regional Processor Block Diagram

previously by the firmware instructions. For example, the CPU is initiated with an instruction by CPU software. This instruction is decoded and the CPU firmware converts the instruction into a series of SelBUS transfers. The transfers are forwarded to the RP and the peripheral device addressed by the software instruction. The SelBUS interface of the regional processor detects this transfer, and if the correct address is present, the SelBUS interface stores the contents of the CPU SelBUS transfer for scrutiny by the microprocessor. After receipt and storage of the SelBUS transfer, the SelBUS interface generates a microprogram interrupt to the microprocessor. This microinterrupt then causes the microprogram to branch to a subroutine, which will process the stored transfer. The microprogram recognizes and identifies the specific type of SelBUS transfer and initiates the actions required to execute the operation.

Upon completion of the foregoing, the regional processor signals the CPU that it is ready to transfer its status. The CPU informs the regional processor to forward the status. The firmware responds to this CPU status request by generating a SelBUS data return transfer (DRT), which transfers the assembled status from the regional processor to the CPU, completing this portion of the implementing operation.

4.3.1.2 Microprogrammable Processor

The RP microprogrammable processor contains the following:

1. A 32-bit parallel-operated microinstruction control register that operates at a 150-nanosecond cycle time.
2. A 16- by 32-bit parallel data structure.
3. A 16-bit arithmetic logic unit.
4. Sixty-four 16-bit general purpose registers.
5. Input/output 16-bit data interfaces.
6. A programmable RP interrupt.
7. An order structure consisting of 48 programmable level and pulse interface lines used for tag and activation signals.
8. A test structure consisting of 64 inputs to the processor control circuits; 32 inputs are connected internally and 32 are available for device interface.

4.3.1.3 Read-only Memory

The read-only memory (ROM) portion of the UDP consists of as many as 8192 program control word locations. Each control word consists of 32 bits of instruction, which is decoded and executed by the microprocessor. The control program (firmware) is permanently loaded into the ROM.

4.3.1.4 Device Interface Logic

The device interface logic consists of components to interface with the DIA-1 circuit card through connectors J1A and J1C on the SelBUS backplane.

4.3.2 Device Interface Adapter 1 (DIA-1)

4.3.2.1 DIA Logic Organization

Figure 4-4 is a simplified block diagram showing the DIA logic organization and its relationship to the balance of the UDP. The diagram shows both DIA-1 and DIA-2. Most of the logic is located in DIA-1. Only the port multiplexer logic for disc drives 4 through 7 is located in DIA-2.

The DIA-1 control logic contains a microsequencer controlled by a resident microprogram stored in ROM (firmware). The microsequencer controls the drive interface signals, such as read and write, and the protocol with the disc drives. The microsequencer also regulates data transmission on a byte basis between the serial data logic and the FIFO queue or the RP, as appropriate. In general, the microsequencer controls and synchronizes data operations in the DIA.

The port logic consists of the port multiplexer and eight ports, P0 through P7. The ports P0, P1, P2, and P3 reside in DIA-1. The ports P4, P5, P6, and P7 reside in DIA-2. The port multiplexer is distributed on the same basis. All other logic resides in DIA-1.

The port logic also contains a six-bit counter for each port, which responds to index and sector pulses received from the associated drive. The counter is reset at index pulse time (beginning of track) and is incremented by one whenever a sector pulse is received. The contents of the sector pulse counters can be gated into an RP register. The RP uses this counter as a basis for angular position targeting. The collection of eight sector counters permits the RP to determine the angular position of any one of the eight disc drives to within one-half of a data sector by reading the sector counters.

The port multiplexer selects clock and serial data signals from one of the eight attached disc drives and gates these signals to the serial data logic and the microsequencer. The port multiplexer also gates the selected disc's index pulse to the microsequencer. In general, the port multiplexer connects the signal set available in a particular B cable to the balance of the DIA logic.

The serial data logic performs a serial-to-parallel and parallel-to-serial data conversion. It also accomplishes pilot bit recognition during read synchronization. The serial data logic contains two serial shift registers, two byte registers, and two byte counters.

The error detection logic provides the logic functions required to support error detection and correction. The logic consists of a 30-bit error code generating and checking mechanism. The results of error detection logic can be read into the RP in two parts via the 16-bit data bus.

The A bus interface logic provides the facilities to issue commands to the disc drive under control of the RP. The microsequencer uses some A bus interface facilities to control read and write timing and to control address mark generation and detection. The A bus interface also contains the line driver and receiver circuits needed to terminate the A cable bus (A Bus).

The FIFO queue is a 256-byte ring queue used to buffer all data to be transmitted between the CPU main memory and disc storage. This buffer is provided to compensate for main memory accessing delays in the SelBUS protocol. The FIFO queue is prefilled during disc write operations and serves as a memory data buffer during disc read operations. FIFO queue transactions are byte-oriented.

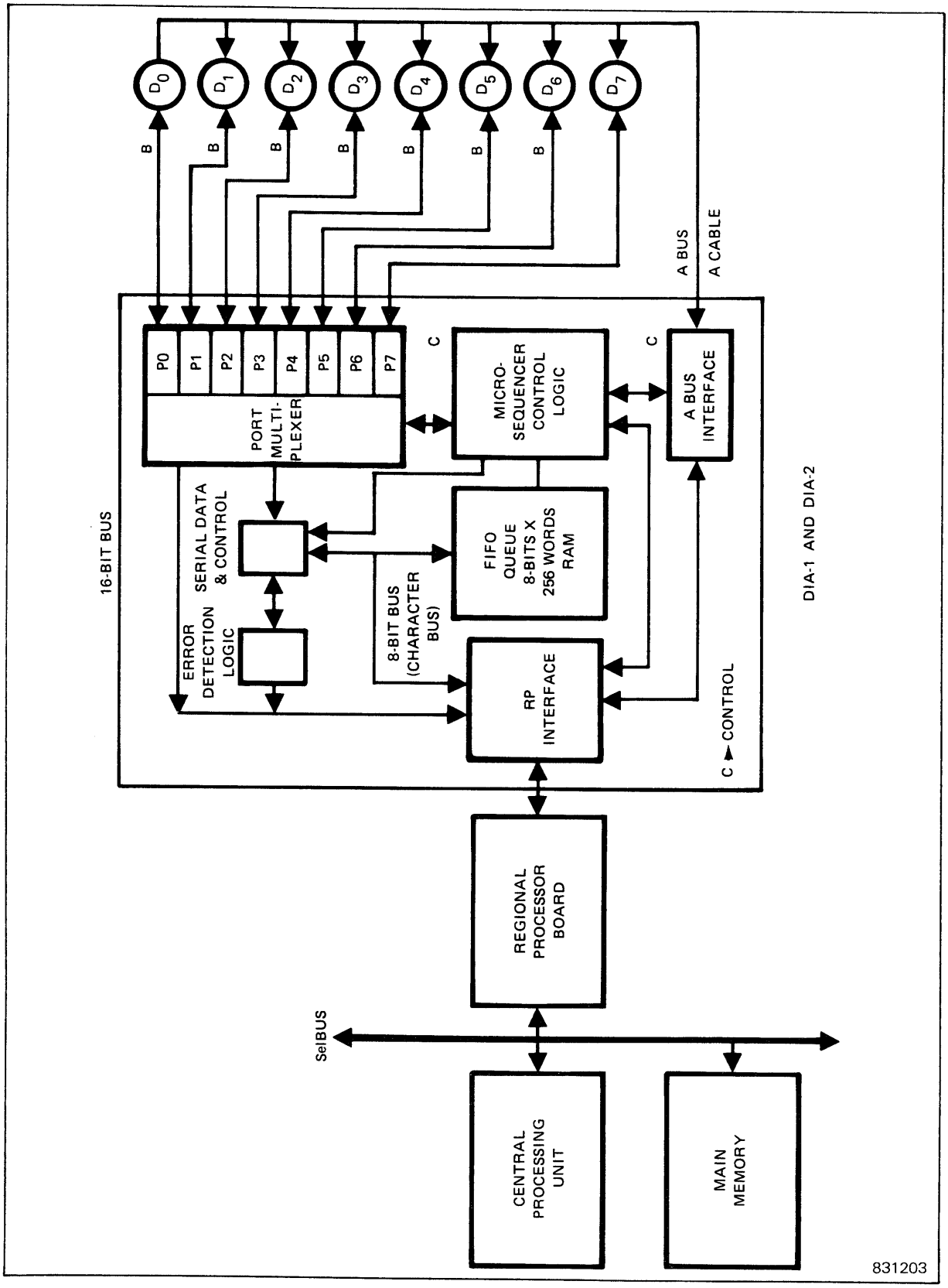


Figure 4-4. Device Dependent Logic Organization

The RP interface logic provides the facilities required to interface the RP to the DIA. This logic includes two 16-bit data registers, pulse order logic, level order logic, test order logic, and RP interrupt logic. The 16-bit data registers provide for filling or unloading the FIFO queue in support of disc write or read operations, respectively. Data transmission between the RP and RP interface logic are normally halfword in size; however, provision has been made for the occasional transfer of a single byte in order to provide for odd-byte data address boundaries.

4.3.2.2 RP Input/Output Signals

Input/output signals to and from the regional processor (RP) travel over connectors P1A and P1C of the device interface adapter 1 (DIA-1). Tables 4-1 and 4-2 list the signal definitions for the two connectors. Note that P1A and P1C occupy the outer parts of the DIA-1 circuit card edge connector, while P1B occupies the center part, as shown in Figure 4-5. P1B is the SelBUS connection described in the RPU reference manual. Since the J1A and J1B connectors are connected straight through the backplane of the logic chassis, the regional processor and the DIA-1 boards must be plugged in directly opposite each other.

4.3.2.2.1 Inputs from RP

CLOCK SIGNALS

The RP provides three clock signals for use by the DIA-1 circuit card. The L2CLK4, L3CLK3, and L2CLKFREERUN signals (sheet 3, drawing 130-103282) provide a negative-going edge that marks the start/end of a microinstruction or bus transfer cycle. The clock signals are 150-nanosecond signals that correspond to the microinstruction cycle time for the regional processor.

RESET SIGNALS

The RP provides two reset signals for use by the DIA-1 circuit card. The LRESET4 signal occurs as a negative pulse to reset various registers in the DIA board. The HRESET2 signal occurs as a positive pulse and is used to reset the disc drives. Both signals are generated whenever a system power-on condition occurs or whenever the SYSTEM RESET switch on the CPU is pressed.

CREG INPUT SIGNALS

The RP provides four bits from the RP control register (CREG) to the DIA-1. These four bits, HCREG12 through 15, are decoded into pulse and level orders for controlling the DIA-1 circuit card. The input signals and the decoding logic are shown on sheet 3, drawing 130-103282.

ORDER ENABLE SIGNALS

The external orders that are decoded from the CREG bits described above must be enabled by four signals. The HORDADR00 and LENPORD signals (sheet 3) enable a bank of pulse orders. The LENLVLORD01 and LENLVLORD02 signals (sheet 3) enable the first and second band of level orders respectively.

Table 4-1
SelBUS Connector Assignments, P1A

Pin	Mnemonic	130-103282 Logic Drawing Sheet Number	Signal
1			(Not used)
2	LENVECTINT	3	Enable vectored interrupt (not used)
3	LENPORD	3	Enable pulse order
4	HRDADR00	3	Read address
5			(Not used)
6	LENLVLORD1	3	Enable level order bit 1
7	LENLVLORD2	3	Enable level order bit 2
8	HCREG12	3	Control register bit 12
9	HCREG13	3	Control register bit 13
10	HCREG14	3	Control register bit 14
11	HCREG15	3	Control register bit 15
12			(Not used)
13			(Not used)
14			(Not used)
15	LAIN00	9	A file input from RP, bit 00
16	LAIN01	9	A file input from RP, bit 01
17	LAIN02	9	A file input from RP, bit 02
18	LAIN03	9	A file input from RP, bit 03
19	LAIN04	9	A file input from RP, bit 04
20	LAIN05	9	A file input from RP, bit 05
21	GND	9	Ground
22	LAIN06	9	A file input from RP, bit 06
23	LAIN07	9	A file input from RP, bit 07
24	LAIN08	9	A file input from RP, bit 08
25	LAIN09	9	A file input from RP, bit 09
26	LAIN10	9	A file input from RP, bit 10
27	LAIN11	9	A file input from RP, bit 11
28	LAIN12	9	A file input from RP, bit 12
29	GND	9	Ground
30	LAIN13	9	A file input from RP, bit 13
31	LAIN14	9	A file input from RP, bit 14
32	LAIN15	9	A file input from RP, bit 15
33	LEXT00	6	External bus to RP B file, bit 00
34	LEXT01	6	External bus to RP B file, bit 01
35	LEXT02	6	External bus to RP B file, bit 02
36	LEXT03	6	External bus to RP B file, bit 03
37	GND	6	Ground
38	LEXT04	6	External bus to RP B file, bit 04
39	LEXT05	6	External bus to RP B file, bit 05
40	LEXT06	6	External bus to RP B file, bit 06
41	LEXT07	6	External bus to RP B file, bit 07
42			(Not used)
43	LERRORIN	5	Error in tag
44			(Not used)
45			(Not used)
46	LNONPRSMEM	5	Nonpresent memory tag
47	LCPUDATAHERE	5	CPU requesting tag
48	L2CLKFREERUN	3	Free-running system clock
49	LRESET4	3	Reset signal
50	L3CLK3	3	System clock

Table 4-2
SelBUS Connector Assignments, P1C

Pin	Mnemonic	130-103282 Logic Drawing Sheet Number	Signal
1	L2CLK4	3	System clock
2	HRESET2	23	Reset signal
3	LHALTIO	5	Halt input/output tag
4			(Not used)
5			(Not used)
6			(Not used)
7	HDATAINT	5	Data interrupt tag
8	LEXT08	6	External bus to RP B file, bit 08
9	LEXT09	6	External bus to RP B file, bit 09
10	LEXT10	6	External bus to RP B file, bit 10
11	LEXT11	6	External bus to RP B file, bit 11
12	LEXT12	6	External bus to RP B file, bit 12
13	GND	6	Ground
14	LEXT13	6	External bus to RP B file, bit 13
15	LEXT14	6	External bus to RP B file, bit 14
16	LEXT15	6	External bus to RP B file, bit 15
17			(Not used)
18			(Not used)
19			(Not used)
20			(Not used)
21	GND	22	Ground
22			(Not used)
23			(Not used)
24	LBIN06	22	B file input from RP, bit 06
25	LBIN07	22	B file input from RP, bit 07
26	LBIN08	22	B file input from RP, bit 08
27	LBIN09	22	B file input from RP, bit 09
28	LBIN10	22	B file input from RP, bit 10
29	GND	22	Ground
30	LBIN11	22	B file input from RP, bit 11
31	LBIN12	22	B file input from RP, bit 12
32	LBIN13	22	B file input from RP, bit 13
33	LBIN14	22	B file input from RP, bit 14
34	LBIN15	22	B file input from RP, bit 15
35			(Not used)
36			(Not used)
37			(Not used)
38	LBUFTSTAD5	4	Buffered test address
39	LBUFTSTAD6	4	Buffered test address
40	LBUFTSTAD7	4	Buffered test address
41	HTESTB2	4	Test output to B group, bit 2
42	LTESTB2	4	Test output to B group, bit 2
43	HTESTB3	4	Test output to B group, bit 3
44	LTESTB3	4	Test output to B group, bit 3
45			(Not used)
46	HTESTA2	4	Test output to A group, bit 2
47	LTESTA2	4	Test output to A group, bit 2
48			(Not used)
49			(Not used)
50			(Not used)

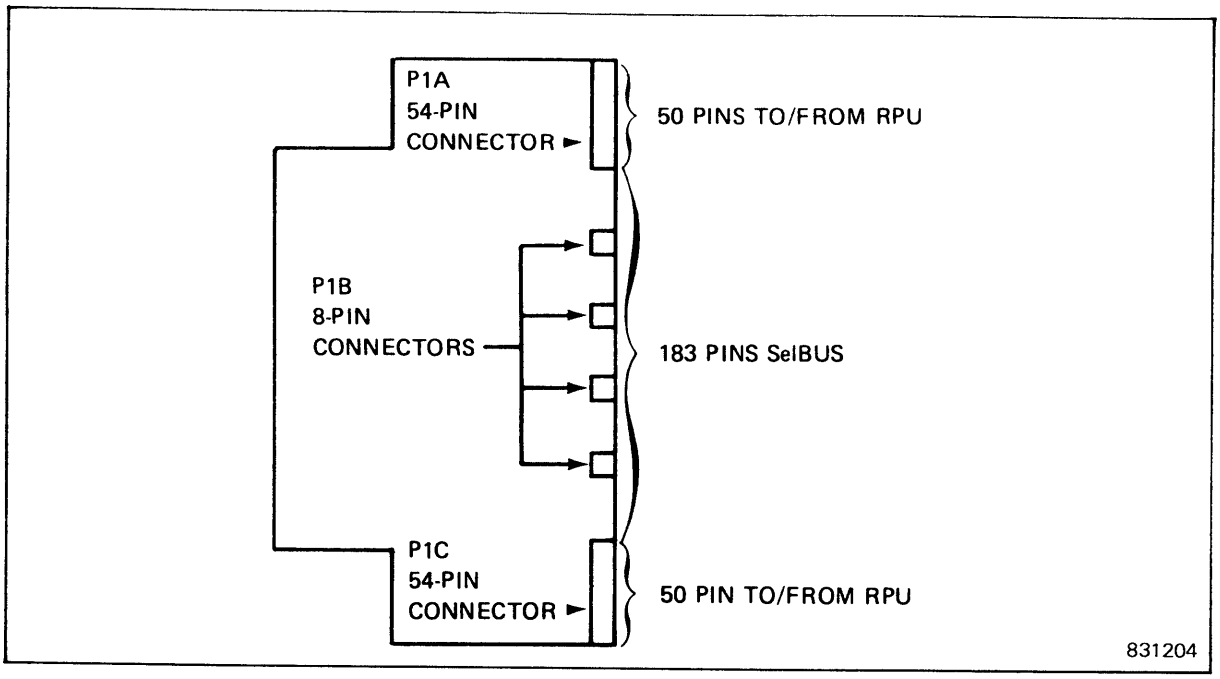


Figure 4-5. DIA-1 Edge Connector

TAG LINE INPUTS

The RP generates a set of tag signals produced by decoding the SelBUS tag bus signal. These tags are used in the interrupt logic shown on sheet 5. The LHALTIO signal indicates a halt I/O command has been generated by the CPU. The signal LCPUDATAHERE indicates that a command for action by the CPU has been received by the RP. The LERRORIN indicates a parity error has occurred on a main memory read transfer. The LNORPRSMEM signal indicates that a nonexistent memory location has been addressed. All of these tag inputs are inserted into the test structure and can be tested by the microprogram.

A FILE INPUTS

The RP has the capability of inputting 16 bits of information into the DIA-1 circuit card from any addressed register in the RP A file. The 16 input bits, LAIN00 through 15, are shown on sheet 9, entering the two output registers for data from the RP. This bus is used to transfer data in halfword format that will be written onto the disc. Note that bits 12 through 14 (LAIN12 through 14) also appear on sheet 5. These three bits are decoded into read select pulses to control how the input data are to be used by the DIA-1.

B FILE INPUTS

The RP has the capability of inputting ten bits of information into the DIA card from any addressed register in the RP B file. The ten input bits (LBIN06 through 15, sheets 22 and 23) are the least significant ten bits of the addressed register. This data consists of control and parameter information that will then be transmitted to the addressed disc drive.

BUFFERED TEST ADDRESS INPUTS

These signals, LBUFTSTAD5 through 7, are the buffered test address for the RP test structure. Since some of the test encoding is done in the DIA-1 logic, these address signals select which signal is to be tested. Sheet 4 shows the test address signals applied to a series of three multiplexers, selecting the addressed signal for use by the RP's test structure.

4.3.2.2.2 Outputs to RP

EXTERNAL BUS

The DIA-1 has the capability of outputting 16 bits of data to any addressed register in the regional processor's B file. These bits, LEXT00 through 15 (sheet 7), comprise a 16-bit external bus to the RP. The bus can be used for a variety of information including data read from the disc, sector count, and ECC data.

TEST STRUCTURE OUTPUT

The DIA-1 can output three pairs of test structure output signals to the RP. These signals are HTESTA2, LTESTA2, HTESTB2, LTESTB2, HTESTB3, and LTESTB3. The signals are input directly into the RP test structure. Both polarities (high true and low true are included so that the tests can be defined either way. One pair of test signals (HTESTA2, LTESTA2) is applied to test group A in the RP, while the other two pairs are applied to test group B in the RP.

DATA INTERRUPT

The data interrupt (HDATAINT) signal (sheet 5) is the primary interrupt sent from the DIA-1 to the RP. This signal results from a number of conditions, including attention flip-flop set, CPU requesting, and a halt I/O received.

VECTORED INTERRUPT

The LENVECTINT signal (sheet 3) is not used.

4.3.2.3 SelBUS Connections

Although the DIA-1 is not functionally connected directly to the SelBUS, the DIA-1 does receive power and ground signals from connector P1B. Power and ground connections are shown on sheet 1, drawing 130-103282.

4.3.2.4 Disc Drive Input/Output Signals

4.3.2.4.1 A and B Cables

Interconnection between DIA-1 and the disc drives is via two cables, a daisy-chained A cable and a set of radial B cables. Pin and signal identifications of the A and B cable signals are given in Tables 4-3 and 4-4. In general, the A cable carries control and status

Table 4-3
A Cable Connector Assignments (J1-J3) (Sheet 1 of 2)

Pin				130-103282 Logic Drawing	
J1	J2	J3	Mnemonic	Sheet Number	Signal
1	1	-	LTAG1	23	Cylinder address tag
2	2	-	HTAG1	23	Cylinder address tag
3	3	-	LTAG2	23	Head select tag
4	4	-	HTAG2	23	Head select tag
5	5	-	LTAG3	23	Control select tag
6	6	-	HTAG3	23	Control select tag
7	7	-	HBUSBIT0	23	Control signal bus bit 0
8	8	-	LBUSBIT0	23	Control signal bus bit 0
9	9	-	HBUSBIT1	23	Control signal bus bit 1
10	10	-	LBUSBIT1	23	Control signal but bit 1
11	11	-	LBUSBIT2	22	Control signal bus bit 2
12	12	-	HBUSBIT2	22	Control signal bus bit 2
13	13	-	LBUSBIT3	22	Control signal bus bit 3
14	14	-	HBUSBIT3	22	Control signal bus bit 3
15	15	-	LBUSBIT4	22	Control signal bus bit 4
16	16	-	HBUSBIT4	22	Control signal bus bit 4
17	17	-	LBUSBIT5	22	Control signal bus bit 5
18	18	-	HBUSBIT5	22	Control signal bus bit 5
19	19	-	LBUSBIT6	22	Control signal bus bit 6
20	20	-	HBUSBIT6	22	Control signal bus bit 6
21	21	-	LBUSBIT7	22	Control signal bus bit 7
22	22	-	HBUSBIT7	22	Control signal bus bit 7
23	23	-	LBUSBIT8	22	Control signal bus bit 8
24	24	-	HBUSBIT8	22	Control signal bus bit 8
25	25	-	LBUSBIT9	22	Control signal bus bit 9
26	26	-	HBUSBIT9	22	Control signal bus bit 9
27	27	-	HOPENCBLE	23	Open cable detector signal
28	28	-	LOPENCBLE	23	Open cable detector signal
29	29	-	LFAULT	24	Fault signal
30	30	-	HFAULT	24	Fault signal
31	31	-	LSEEKERR	24	Seek error signal
32	32	-	HSEEKERR	24	Seek error signal
33	33	-	LONCYL	24	On cylinder signal
34	34	-	HONCYL	24	On cylinder signal
35	-	-			(Not used)
36	-	-			(Not used)
37	-	3	LUNITRDY	24	Drive unit ready signal
38	-	4	HUNITRDY	24	Drive unit ready signal
39	-	5	LAMF	24	Address mask found
40	-	6	HAMF	24	Address mask found
41	-	7	LBUSY	24	Drive busy signal
42	-	8	HBUSY	24	Drive busy signal
43	-	9	LUSTAG	23	Unit select tag
44	-	10	HUSTAG	23	Unit select tag
45	-	21	LWRTPROT	24	File is write protected
46	-	22	HWRTPROT	24	File is write protected
47	-	-			(Not used)
48	-	-			(Not used)

Table 4-3
A Cable Connector Assignments (J1-J3) (Sheet 2 of 2)

Pin J1	Pin J2	Pin J3	Mnemonic	130-103282 Logic Drawing Sheet Number	Signal
49	-	-			(Not used)
50	-	-			(Not used)
-	-	1			(Not used)
-	-	2			(Not used)
-	-	11	HUNITSEL0	23	Unit select bit 0
-	-	12	LUNITSEL0	23	Unit select bit 0
-	-	13	HUNITSEL1	23	Unit select bit 1
-	-	14	LUNITSEL1	23	Unit select bit 1
-	-	15			(Not used)
-	-	16			(Not used)
-	-	17	HUNITSEL2	23	Unit select bit 2
-	-	18	LUNITSEL2	23	Unit select bit 2
-	-	19	LUNITSEL3	23	Unit select bit 3
-	-	20	HUNITSEL3	23	Unit select bit 3
-	-	23			(Not used)
-	-	24			(Not used)
-	-	25			(Not used)
-	-	26			(Not used)

between the addressed disc drive and the DIA-1 card. The B cable, on the other hand, is used primarily for read/write data and the associated protocol with each individual disc drive.

The A cable utilizes three connectors, J1, J2, and J3. These connectors are designed to allow the system to connect to either a 50-pin or 60-pin cable interface. Currently, only the 60-pin flat cable interface is used. This interface connects to J2 and J3.

The B cable consists of a 26-pin flat cable interface. Four connectors, J6, J7, J8, and J9, are provided at the rear of the DIA-1 circuit card for connection of the B cables.

4.3.2.4.2 DIA-1/Disc Drive Interface Signals

Figure 4-6 is a diagram of the interface connections between the disc drives and the UDP. Table 4-5 provides a complete description of each drive interface signal.

4.3.2.5 Signals To/From DIA-2

Device interface adapter 2 is connected to DIA-1 via two cables and to disc drives by up to four B cables (radial cables). B cable signals are identical to the B cable connections for DIA-1, as described in Figure 4-6 and Tables 4-4 and 4-5.

A 26-flat cable connects J4 of DIA-2 to J4 of DIA-1. The signals on this cable are shown in Table 4-6. This cable consists of external bus bits 0 through 15 (LEXT00 through 15). The cable carries the sector count for addressed disc drive over this cable.

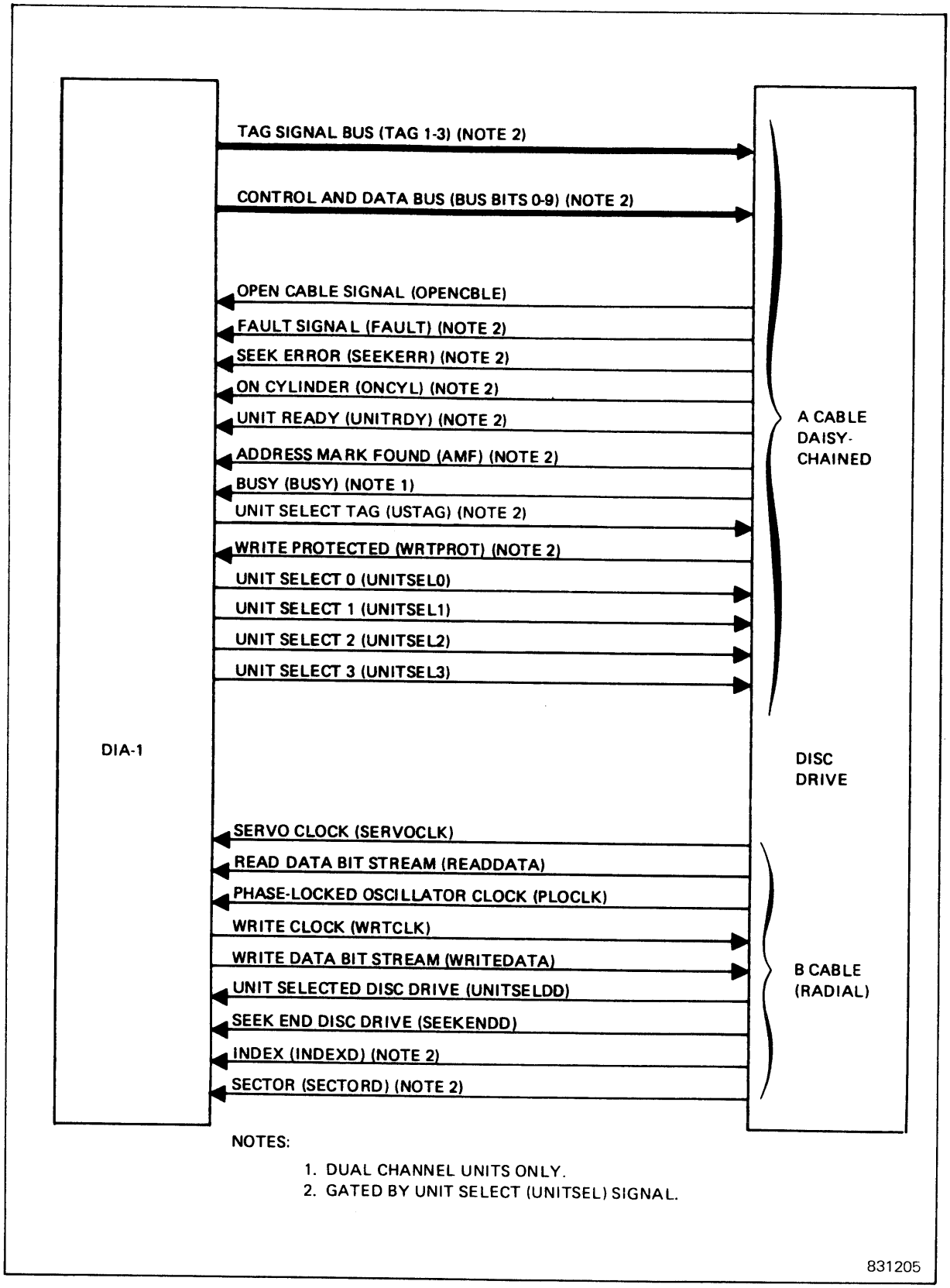


Figure 4-6. DIA-1/Disc Drive Interconnection (Sheet 1 of 2)

		LEVEL ORDERS			
		HCYCLINDSEL	HHEADSEL	HCONTROLSEL	HUNITSEL
		TAG 1 IN	TAG 2 IN	TAG 3 IN	UNIT SELECT2
RPU B FILE LBIN	BUS BITS	CYLINDER ADDRESS	HEAD SELECT	CONTROL SELECT	
15	0	0	0	WRITE GATE	UNIT SEL 2^0
14	1	1	1	READ GATE	UNIT SEL 2^2
13	2	2	2	SERVO OFFSET PLUS	UNIT SEL 2^2
12	3	3		SERVO OFFSET MINUS	
11	4	4		FAULT CLEAR	
10	5	5		ADDRESS MARK ENABLE	
9	6	6		RETURN TO ZERO (RTZ)	
8	7	7		DATA STROBE EARLY	
7	8	8		DATA STROBE LATE	
6	9	9		RELEASE	PRIORITY SELECT

831206

Figure 4-6. DIA-1/Disc Drive Interconnection (Sheet 2 of 2)

Another 26-pin flat cable connects J5 of DIA-2 to J5 of DIA-1. The signals on this cable are shown in Table 4-7. The HSERVOCLK, LINDEX, HPLOCLK, and HREADDATA signals are the same as previously described in Table 4-5. The HWRITEDATAOUT is the serial data to be written on the disc. The LREADCONT45 and LREADCONT67 signals provide an enable to allow the sector counters for disc drives 4/5 or 6/7 to be read.

4.3.2.6 DIA-1 Interface Logic

The DIA-1 interface logic is the device-dependent logic that tailors the general purpose regional processor to operate with moving-head discs. Figure 4-4 and the associated description given previously in this chapter provide an overview of the organization and function of the DIA-1 interface logic. Figure 4-7 is a more detailed block diagram of the DIA-1 interface logic. The principal elements of the logic shown on this diagram are the

Table 4-4
B Cable Connector Assignments (J6-J9)

Pin	Mnemonic	130-103282 Logic Drawing Sheet Number	Signal
1	GND	25	Ground
2	HSERVOCLK0-3	25	Servo clock
3	LSERVOCLK0-3	25	Servo clock
4	GND	25	Ground
5	LREADDATA0-3	25	Read data (serial bit stream from disc)
6	HREADDATA0-3	25	Read data (serial bit stream from disc)
7	GND	25	Ground
8	HPLOCLK0-3	25	Phase-locked oscillator clock
9	LPLOCLK0-3	25	Phase-locked oscillator clock
10	GND	24	Ground
11	LWRITECLK0-3	24	Write clock
12	HWRITECLK0-3	24	Write clock
13	GND	24	Ground
14	HWRITEDATA0-3	24	Write data (serial bit stream to disc)
15	LWRITEDATA0-3	24	Write data (serial bit stream to disc)
16	GND	25	Ground
17	HUNITSELDD0-3	26	Unit select disc drive
18	LUNITSELDD0-3	26	Unit select disc drive
19	LSEEKENDD0-3	26	Seek end
20	HSEEKENDD0-3	26	Seek end
21	GND	25	Ground
22	HINDEXD0-3	26	Index pulse
23	LINDEXD0-3	26	Index pulse
24	GND	25	Ground
25	LSECTORD0-3	26	Sector pulse
26	HSECTORD0-3	26	Sector pulse

Note: Disc 0 = J6, disc 1 = J7, disc 2 = J8, and disc 3 = J9

input and output registers, the serial-to-parallel and parallel-to-serial output registers, the FIFO RAM, the byte counter, the microsequencer, the error correcting code logic, the test and order structure, and the A and B cable interface logic. These elements of the logic circuitry are described in the following paragraphs. Details of these circuits are shown on logic drawing 130-103282 (sheets 1 through 26) in the drawings manual. Sheet references in the following text and shown in Figure 4-7 refer to logic drawing 130-103282.

4.3.2.6.1 General Description

Figure 4-7 shows both DIA-1 and DIA-2. DIA-2 is indicated by the dashed lines. The DIA-2 card is also called the port expander, since it contains ports for the connection of four additional drives for a total of eight drives in the system. As seen in the figure, control of the discs is exercised by sending control commands from the RP's B file (LBIN00 through 15) to an addressed disc via the A cable. Write data is transferred in

Table 4-5
Controller/Disc Interface Signal Definitions (Sheet 1 of 4)

Mnemonic	Signal	Definition
A CABLE		
TAG1-3	Tag bus	These three tag lines identify what type of information is on the 10-bit data bus to the disc drive. If tag one is true, the cylinder address is contained on the data bus. If tag two is true, the head select address is contained on the data bus. If tag three is true, control select functions are on the data bus.
BUSBIT0-9	Control and data bus	This 10-bit bus carries control and addressing information to the disc drives. The data on the bus are interpreted as follows, depending on which tag lines are true:
	Tag 1	Cylinder address
	Tag 2	Head select
	Tag 3	Control select
		Bit 0: Write gate. Enables write driver.
		Bit 1: Read gate. Enables digital read data on transmission lines.
		Bit 2: Servo offset plus. When this signal is true, the actuator is offset from the nominal on cylinder position away from the spindle.
		Bit 3: Servo offset minus. When this signal is true, the actuator is offset from the nominal on cylinder position away from the spindle.
		Bit 4: Fault clear. This bit clears the fault flip-flop in the disc drive if the fault condition no longer exists.
		Bit 5: Address mark enable. Allows the writing or recovering of address marks in conjuction with write gate or read gate. When address mark enable is true, the writer stops toggling and erases the data, creating an address mark. Write fault detection is inhibited by this signal.

Table 4-5
Controller/Disc Interface Signal Definitions (Sheet 2 of 4)

Mnemonic	Signal	Definition
		Bit 6: RTZ (return to zero). This bit will cause the actuator to seek track zero, reset the head register and clear the seek error flip-flop in the disc drive. Since this seek is significantly longer than a normal seek to track zero, it is only used to reset the disc drive.
		Bit 7: Data strobe early. When true, this bit will cause the disc phase-locked oscillator data separator to strobe the data earlier than normal. Normal strobe timing will be returned when the line is false. The signal can be used as an aid to recovering marginal data on the disc.
		Bit 8: Data strobe late. Same as bit 7, except later than normal.
	Unit select	Bit 9: Release. (For dual-ported disc drives only.) Enabling this line will release the channel reserve and permit the other channel to access the disc drive.
OPENCBLE	Open cable	The open cable detect circuit disables the interface in the event that the A cable is disconnected or controller power is lost.
FAULT	Fault	When true, this line indicates a fault in the disc drive. The following types of faults may be detected: dc voltage fault, head select fault, write fault, write or read while off cylinder, and write gate during a read operation. A fault condition will immediately inhibit the writer to prevent data destruction. The dc voltage fault indicates a below-normal voltage from positive or negative power supplies. The head select fault indicates that more than one head is selected. The write fault indicates a low (or absent) write current, as well as the absence of write data. This line may be cleared by setting bit 4 of a control select command, by fault clear on the operator panel or master fault clear on the fault card.
SEEKERR	Seek error	When true, a seek error has occurred in the disc drive. The error may only be cleared by executing an RTZ (bit 6) with the control select command. The signal indicates that the moving-head carriage has moved to a position outside the recording field.

Table 4-5
Controller/Disc Interface Signal Definitions (Sheet 3 of 4)

Mnemonic	Signal	Definition
ONCYL	On cylinder	This signal is a status indicator generated when the servo has positioned the heads over a track. The signal is cleared by any seek command or an RTZ.
UNITRDY	Unit ready	When this signal is true and the unit is selected, it indicates that the unit is up to speed, the head is in position over the recording tracks, and the no fault condition exists in the drive.
AMF	Address mark found	This signal is generated when the disc identifies the address mark. Upon receipt of this signal, the disc processor drops the address mark enable line (bit 5). Then, valid data will be presented on the input/output lines and the address mark found pulse will be reset.
BUSY	Busy	This signal is used with dual channel disc drives. If the drive is already reserved or selected, a busy signal will be issued to the A cable and a unit selected will be issued to the B cable.
USTAG	Unit select tag	This signal gates the disc address (UNITSEL0-3) into the address compare circuit.
WRTPROT	Write protected	This signal informs the controller that the disc is write protected and the writer is inhibited under all conditions. Attempting to write while protected will cause a fault to be issued.
UNITSEL0-3	Unit select	These four lines are binary coded to select 1 of 16 disc drives. In the disc processor, the valid range of addresses is 0 through 7.
B CABLE		
SERVOCLK	Servo clock	The servo clock is a phase-locked 9.677 MHz clock generated from the disc's servo track. The servo clock is used to generate write bits and is available at all times.
READDATA	Read data	This line contains the serial bit stream that is read from the disc.
PLOCKL	Phase-locked oscillator clock	This clock signal defines the beginning of a data cell and is synchronous with the detected data. The signal is used by the controller as a read clock.

Table 4-5
Controller/Disc Interface Signal Definitions (Sheet 4 of 4)

Mnemonic	Signal	Definition
WRTCLK	Write clock	This signal is generated by the UDP to provide a write clock to write data onto the disc. The write clock is the servo clock retransmitted to the disc drive during a write operation.
WRITEDATA	Write data	This line contains the serial bit stream of data that is to be written onto the disc.
UNITSELDD	Unit selected	When the four unit select bit lines compare with the settings of the unit select switches in the disc drive and when the unit select tag is received, the unit selected line on the B cable becomes true. This signal identifies to the controller which disc drive has been selected.
SEEKEND	Seek end	The seek end signal is the combination of on-cylinder and seek errors indicating that a seek operation has terminated.
INDEXD	Index pulse	The index pulse is used to identify the starting point of a track. This signal occurs once per revolution, and its leading edge is considered the leading edge of sector zero.
SECTORD	Sector pulse	The sector pulses identify the sector areas on the track. The sector is derived from the servo track. The number of sector pulses per revolution is switch selectable and is determined by counting servo clocks. There are two pulses per sector area. For the UDP, two switch settings are used: one for a 16-sector disc and one for a 20-sector disc. Refer to Chapter 2 for switch settings.

halfwords to the DIA-1 from the RP's A file (LAIN00 through 15). The write data is separated into two bytes that are converted into a serial bit stream. The serial bit stream is routed through the error correcting code logic and transmitted to the addressed disc via the B cable.

A read operation works in the reverse direction. Serial information from the disk is transmitted via the B cable to the DIA-1. This data is processed by the error correcting code logic and converted from serial data into two bytes. The bytes are assembled into a halfword and transmitted to the RP, which assembles the data into a word and transmits it to memory via the SelBUS.

Table 4-6
DIA-2 (Port Expander) Connector Assignments (J4)

Pin	Mnemonic	130-103282 Logic Drawing Sheet Number	Signal
1	GND		Ground
2	LEXT00		External bus bit 00
3	LEXT01		External bus bit 01
4	LEXT02		External bus bit 02
5	LEXT03		External bus bit 03
6	GND		Ground
7	LEXT04		External bus bit 04
8	LEXT05		External bus bit 05
9	LEXT06		External bus bit 06
10	LEXT07		External bus bit 07
11	GND		Ground
12	LEXT08		External bus bit 08
13	LEXT09		External bus bit 09
14	LEXT10		External bus bit 10
15	LEXT11		External bus bit 11
16	GND		Ground
17	LEXT12		External bus bit 12
18	LEXT13		External bus bit 13
19	LEXT14		External bus bit 14
20	LEXT15		External bus bit 15
21	GND		Ground
22			(Not used)
23			(Not used)
24			(Not used)
25			(Not used)
26			(Not used)

Internal control of the DIA-1 logic is maintained by a microsequencer operating under control of a stored program (firmware). The basic function of the microsequencer is to coordinate the sequence of events involved in a data transfer. All the buffers, registers, byte count registers, and sector count registers operate under the control of the microsequencer.

4.3.2.6.2 RP Input Registers

The regional processor input registers (R10 and R11) consist of two eight-bit registers, as shown on sheet 7. The purpose of these registers is to accept bytes of information from the DIA-1, assemble them into halfwords, and transfer the halfwords to the regional processor. Inputs to the register are from the eight-bit-wide character bus (LCBUS0

Table 4-7
DIA-2 (Port Expander) Connector Assignments, J5

Pin	Mnemonic	130-103282 Logic Drawing Sheet Number	Signal
1			(Not used)
2			(Not used)
3			(Not used)
4	HSERVOCLK	25	Servo clock pulse
5	LINDEX	25	Disc index pulse
6	HPLOCLK	25	Phase-locked oscillator clock signal
7			(Not used)
8	HREADDATA	25	Serial data stream for a read operation
9			(Not used)
10	HWRITEDATAOUT	24	Serial data stream for a write operation
11	GND	24	Ground
12	H2WRITECLK1	24	Write clock
13	GND	24	Ground
14	LREADCNT45	5	Read sector counter enable, drives 4 & 5
15			(Not used)
16	LREADCNT67	5	Read sector counter enable, drives 6 & 7
17			(Not used)
18			(Not used)
19			(Not used)
20			(Not used)
21			(Not used)
22			(Not used)
23			(Not used)
24			(Not used)
25			(Not used)
26			(Not used)

through 7) that permits byte transfers from a variety of locations. The registers are loaded by application of two load clock signals, LLOARDRI0 and LLOADRI1. Both clock signals are generated by the microsequencer. The registers can be read out by the application of the LREADI signal, a pseudo order generated by the regional processor via the read control logic described below.

4.3.2.6.3 RP Output Registers

The regional processor output registers (R00 and R01) are similar to the input registers described above. They consist of two eight-bit registers, shown on sheet 9. The purpose of the registers is to accept halfwords of information from the regional processor, then break the halfwords down into bytes. Inputs consist of 16 bits from the RP's A file (LAIN00 through 15), while outputs from the registers consist of the eight-bit character bus (LCBUS0 through 7). The registers are loaded by the HLOADR0 signal clocked by the LZCLKFREERUN signal. The registers are read out by the application of either LENABER00 or LENABLER01 signals.

4.3.2.6.4 FIFO RAM

The first-in, first-out (FIFO) RAM (sheet 9) acts as a data cushion to prevent data overruns or underruns during read or write operations. The RAM is eight bits wide and holds a maximum of 256 bytes. The RAM is connected to the character bus such that it can read from or write to the character bus (LCBUS0 through 7). Addressing is controlled by an address counter and control logic, as described below.

4.3.2.6.5 FIFO Address Counters

Sheet 11 shows the FIFO address counters. The purpose of this logic is to keep track of the address locations used by the RAM and control the read and write addresses to the RAM. The logic consists of two eight-bit counters and a pair of 2:1 data selectors. The left counter on sheet 11 (consisting of B01 and C01) contains the write address. The right counter on sheet 11 (consisting of B03 and C03) contains the read address of the FIFO.

In operation, the FIFO acts as a ring queue. Initially, both counters are reset to zero by the LDIARESET2 signals. When data is to be written into the FIFO, the HENWRTFIFO signal is raised and the data is clocked in by H3CLK4. Each time a byte is loaded into the FIFO, the counter increments by one. The FIFO can be loaded with a maximum of 256 bytes. After some number of bytes has been loaded, the HENRDFIFO signal is raised to read the data out of the FIFO. Initially, the read counter is reset to zero, so the byte with the zero address is read first. Reading continues with the read counter incrementing by one each time a byte is read. This process continues with the two counters maintaining the addresses of the first and last byte in the FIFO. When either counter reaches 255, it automatically returns to zero to complete the cycle.

The output of the counters is selected by the LENFIFORAM signal. Raising the signal selects the write counter, and lowering the signal selects the read counter. The count from the selected counter is gated through the data selectors to become the address bits (HFIFOADDR0 through 7) for the FIFO RAM.

4.3.2.6.6 FIFO Control Logic

The FIFO control logic (sheet 12) keeps track of how many bytes are in the FIFO. The eight-bit counter (D21, D23) is enabled by either a read or write and counts up during a write or down during a read. Thus, the counter always contains the number of bytes in the FIFO.

Data written into the FIFO originates from four locations, and data read out of the FIFO goes to four locations. These transfers are controlled by the microsequencer order structure. Table 4-8 lists the signals that control transfers to and from the FIFO. Any of these signals can increment or decrement the FIFO counter. If the counter becomes full, the rate error flip-flop (A23-2) is set, a condition that can be tested by the regional processor. In addition, further transfers from the FIFO are inhibited by the LFIFOEMPTY signal. Note that DIARESET clears both counters and the error flip-flop.

4.3.2.6.7 RP External Orders

The RP external order logic is shown on sheet 3. The external orders are encoded in the RP microinstruction in control register bits 12 through 15 (HCREG12 through 15). These

Table 4-8
FIFO Transfer Signals

Mnemonic	Definition
LLOADFQRI-IB	Write FIFO queue and RI register from input buffer
LLOADFQ-IB	Write FIFO queue from input buffer
LLOADFQ-R01	Write FIFO queue from output register 1
LLOADFQ-R00	Write FIFO queue from output register 0
LLOADOB-FQ	Read FIFO queue to output buffer
LLOADBC-FQ	Read FIFO queue to byte count register
LLOADRI1-FQ	Read FIFO queue to input register 1
LLOADRI0-FQ	Read FIFO queue to input register 0

bits are decoded by the DIA-1 logic. The external orders consist of eight pulse orders and 16 level orders, although not all of the orders are used. Pulse orders are generated by a 3:8 decoder (A20) when enabled by LENPORD and addressed by HORDADR00. Level orders are generated by a pair of eight-bit addressable latches (B20, C20). Each bank of level orders is enabled by either LENLVLORD1 or LENLVLORD2.

4.3.2.6.8 System Clock and Reset Signals

Three system clocks (sheet 3) are received from the RP: L2CLK4, L2CLKFREERUN, and L3CLK3. The L2CLK4 is expanded to provide an additional three clocks for a variety of purposes.

Reset results from either of two conditions: receipt of LRESET4 or a microcode instruction in CREG bits 12 through 15. LRESET4 is generated by the RP if a power-up occurs or if the RESET switch on the CPU is pressed. The microcode instruction causes pulse order 78, LDIACLEAR. Either reset source will produce the same result: clearing all registers and counters in the DIA-1.

4.3.2.6.9 RP External Test Structure and Configuration Jumpers

Sheet 4 shows the RP external test structure. This logic consists of three 8:1 data selectors. The signals to be tested are applied to the data selectors and addressed by the buffered test address lines (LBUFTSTAD5 through 7) from the regional processor. The selected test line is gated to pin Y of the data selector, while the inverted signal appears on pin W. The resulting six test outputs (HTESTA2, LTESTA2, HTESTB2, LTESTB2, HTESTB3, and LTESTB3) are connected as inputs to the regional processor test structure.

One of the test structure data selectors is connected to a set of eight jumpers. These jumpers are connected to identify the system configuration to the RP firmware. Only jumpers 0 through 4 are used. These jumpers enable the firmware to test the configuration of the system and respond accordingly. Refer to Chapter 2 and sheet 4 of logic drawing 130-103282 for connection instructions and definitions of the jumper positions.

4.3.2.6.10 DIA Interrupt Logic and Read Select Logic

DIA interrupt and read select logic (sheet 5) consists of several related functions: condition flip-flops, interrupt logic, read select logic, and test signal synchronization.

Three condition flip-flops are shown on sheet 5. The address mark flip-flop (A19-1) is toggled by the LAMFOUND signal, which is generated by the disc drive when the address mark is encountered. The flip-flop is cleared by level order 36 (HENAMF-I), which is generated by a decode of the microinstruction. When set, the address mark flip-flop supplies a signal (HAMFF), which is testable by the microsequencer.

The end flip-flop (B19-2) marks the end of a particular routine. This flip-flop is set by the microsequencer and reset by an order from the RP. When set, the flip-flop marks the end of a particular command function (DIAC) that has been completed by the microsequencer.

The attention flip-flop (B19-1) is used by the microsequencer to request attention from the RP. When set, the HATTENTION signal is applied to the RP test structure.

The interrupt logic also uses the HATTENTION signal along with two other signals to generate an interrupt (HDATAINT) to the regional processor. The HATTENTION signal must be enabled by a HENATTN-I signal from the RP external order logic. Two other signals also generate an interrupt to the RP: LCPUDATAHERE and LHALTIO. LCPUDATAHERE is generated when the SelBUS tags indicate the CPU has transmitted command for processing. LHALTIO is generated when a reset channel instruction from the CPU is pending.

The read select logic consists of a 3:8 decoder (A09), shown on sheet 5. This logic decodes bits 12, 13, and 14 of the A file input (LAIN12 through 14) from the RP. The purpose of the logic is to generate a set of select pulse orders to control reading of data. The result is a set of pulse orders controlled by the RP and totally separate from the RP external order structure or from the microsequencer order structure. LREADCNT01 enables the output of the sector counters for drives 0 and 1. Similarly, LREADCNT23, 45, and 67 enable the output of the sector counters for the remainder of the eight drives. The LREADRI signal enables the output of the DIAR input data register. The LREADECC signal enables the output of the error correction code, while the LSELECCODD signal selects the inputs to the ECC data selector.

The two registers on the right of sheet 5 (D25, D26) simply synchronize the test functions with the system clock. The basic tests are applied, and the registers are clocked by H3CLK4. The resulting signals are consequently synchronized with the clock signal. The signal mnemonics reflect the synchronization by the addition of an "S" at the end of the mnemonic.

4.3.2.6.11 Error Correcting Code Logic

The error correcting code (ECC) logic is shown on sheets 6, 15, 16, and 17. The purpose of this logic is to generate an error correcting code that is appended to each sector of data written onto the disc. During a read operation, the sector data is routed through the ECC logic, and an ECC code is generated. This code is compared to the original code written on the disc. A difference in the two codes indicates a transmission error. The RP firmware can use the result of the ECC logic to compute if the error is correctable or uncorrectable. If the error is correctable, the firmware provides the mask and position information for correcting the data.

Sheets 15 and 16 contain polynomial generators to produce the error correcting codes. Sheet 6 contains a data selector for reading the ECC register into the RP. Sheet 17 contains the control and data select logic.

The error correction facilities employ a generator polynomial of degree 30, which is factored into three relatively prime polynomials whose product is the generator polynomial. These three polynomials are referred to as P0, P1, and P2 in the following test. The generator polynomial, g(X) can be described as

$$g(X) = P0*P1*P2$$

Note: * implies module 2 multiplication.

1. P0, P1, and P2 are all irreducible and relatively prime to each other.

2. The polynomials have a repeat period that is small, which implies a short microcode execution time when generating error correction data. This repeat will be described further in the text.

3. The order of magnitude for g(X) is 30, requiring 32 bits of track storage per ECC.

4. The polynomials will have the ability to correct any one nine-bit error and detect an error up to 29 bits in length in any sector.

The following are the polynomials used in the ECC and some of their properties:

$$P0 = X^{11} + X^9 + X^7 + X^6 + X^5 + X + 1$$

$$P1 = X^{10} + X^9 + X^8 + X^7 + X^6 + X^5 + X^4 + X^3 + X^2 + X + 1$$

$$P2 = X^9 + X^7 + X^4 + X^3 + 1$$

$$G(X) = X^{30} + X^{29} + X^{28} + X^{27} + X^{26} + X^{25} + X^{20} + X^{16} + X^{15} + X^{14} + X^{13} + X^{12} + X^{10} + X^5 + X^4 + X^3 + 1$$

Polynomial	Period	Constant
P0	23	3796
P1	11	
P2	73	14674
M	18469	

When information is transferred to or from the disc, it is shifted bit-serially through a 30-bit exclusive OR register (the ECC generator) (sheets 15 and 16, logic drawing 130-103282). When writing to the disc, the contents of this register are appended to the sector data. When data are read from the disc to memory, the 1024/768 bytes of data and the ECC are again shifted through the ECC generator and will cause the hardware check code computation to go to zero if there are no errors. Figure 4-8 is a diagram of the encoder logic and its relation to the ECC polynomials.

4.3.2.6.12 Input/Output Register Control Logic

Input/output register control logic (sheet 10) controls the RP input registers (RI0 and RI1) and the RP output registers (R00 and R01). The purpose of the circuit is the loading

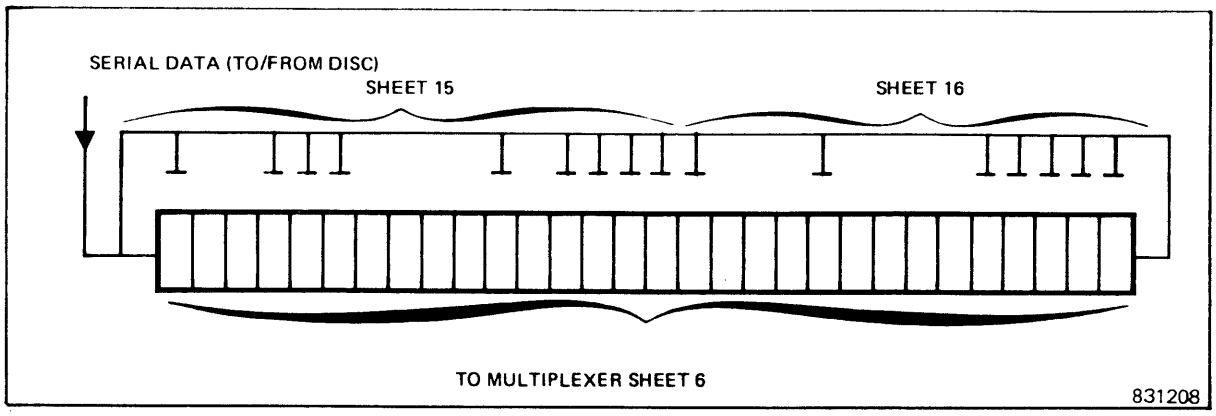


Figure 4-8. Encoder (ECC Bits Generator)

and unloading of the four registers. Four flip-flops comprise the heart of the circuit. These flip-flops (A17-1, A17-2, A15-1, and A15-2) are set or reset, as required, to indicate when any of the four registers are full (loaded). Logic gates establish the restrictive conditions under which transfers to or from the registers can be made.

All four flip-flops are reset to zero by LDIARESET1, while L3CLK3 provides the clock signal. The transfer signal (LXFR) generated by the microsequencer controls the timing of all movements to and from the registers. All of the load and enable signals for the registers are timed with the LXFR signal.

The following example shows how control of the RP output register is exercised when transferring a byte from R00 to the FIFO queue. First, the R00 full flip-flop is reset by LDIARESET1. Then the RP issues a LLOADR0 pulse order to load the R00 and R01 registers, thus setting both the R00 and R01 flip-flops to indicate the registers are full. If the R00 register is full, if the RP has generated a write operation order (HWRITEOPER), and if the LXFR strobe is received from the microsequencer, then a LLOADFQ-R00 and a LENABLER00 are generated, which strobes the byte from R00 into the FIFO queue. At the same time, the R00 full flip-flop is reset to the empty condition. Also, since the R00 data can be loaded into the byte count register, a microsequencer pulse order (LLOADBCH-R0) will enable the output of R00 and reset the flip-flop.

The remainder of the logic operates in similar fashion. One condition requires additional explanation. The LBYTEXFR pulse order from the RP signals the transfer of a single byte rather than a halfword from the RP. When this signal is low, the R01 flip-flop is inhibited from being set. The single byte is subsequently transferred into R00 and into the FIFO queue.

4.3.2.6.13 Serial-In and Serial-Out Shift Registers

Since data transfers to and from the disc must be in serial form, the circuit must make provisions for converting serial data to parallel form and parallel data to serial form. The logic for this purpose (sheet 13) consists of four registers. Data that has been read from the disc (HREADDATA) enters an eight-bit serial-in, parallel-out shift register. The data is clocked in by the HPLOCK signal from the disc drive. When eight bits have been loaded into the register, the input buffer register is strobed with the LLOADIB signal. The byte that has been read from the disc can now be output onto the character bus (LCBUS0 through 7) with the LENABLEIB signal. The character bus transfers the byte into the FIFO queue.

Write data is converted from parallel to serial in the reverse manner. A byte from the FIFO queue is transferred over the character bus (LCBUS0 through 7) to the output buffer. The output data is loaded into the output buffer by the LLOADOB signal and is applied to an eight-bit parallel-in, serial-out shift register. The LLOADSOREG signal loads the parallel data into the registers. When this signal goes high, the data is shifted out in serial form to the disc drive to be written on the disc. The servo clock signal (HSERVOCLK) obtained from the disc is used to clock the serial data out.

4.3.2.6.14 Serial Write and Read Control Logic

The serial-in and serial-out shift sequences described above are controlled by the logic shown on sheet 14. This logic consists of a number of condition flip-flops and two counters. The counters count the number of bits in the serial stream and provide a strobe pulse every eight bits. One counter (A14) controls the serial read data while the other (B10) controls the serial write data. The eight-count outputs (HLOADIBEN and HLLOADSOREG) control the shifting of data from serial to parallel and parallel to serial.

The write gate flip-flop (A23-1) is set by the LSWRITE signal and clocked with the index pulse (LINDEX). The LXWRITE or DIARSET2 signals reset the flip-flop. When the flip-flop is set, the write gate in the disc drive is enabled by the LWRITEGATEFF signal.

The sync search logic is used when reading a sector to identify the beginning of data on the sector. The sync character X'80' identifies the beginning of the sector data. The microsequences issue a pulse order, HSYNSRCH, to search for the sync code on the disc, thus setting the sync search flip-flop (A12-1), which will, in turn, set the sync search sync flip-flop (A13-1). The output of the sync search flip-flop is applied to a gate (C07) along with the L1READCLK2 signal. When the sync signal (1000 0000) is received from the disc on LSIN0 through 7, the gate is enabled, allowing data to be clocked in from the disc through the ECC logic. Simultaneously, the sync search flip-flop is reset.

The count enable flip-flop (A13-2) enables the read counter (A13). The input buffer full flip-flop (A12-7) indicates when the input buffer has been loaded with a full eight bits. Similarly, the output buffer empty flip-flop indicates when the output buffer is empty and ready for another byte. The write sync flip-flop enables the serial write counter.

4.3.2.6.15 Microsequencer PROM and Control Register

Sheet 18 shows the microsequencer PROM and control register (CREG). The PROM contains the permanently stored microsequencer firmware and consists of 1024 instruction words, each 20 bits wide. When an address is applied to the PROM, the contents of that location are applied to the control register. Note that the most significant 11 bits of the microinstruction are used for sequencer control, while the least significant nine bits are used for addressing. The CREG is cleared by LDIARESET3, thus the microprogram always returns to the zero address whenever the DIA-1 is reset. The least significant bit of the address can be modified by either the LTEST8-F or the LTEST10-17. Application of either signal during an even-numbered instruction cycle can increment the address by one. This feature is used to perform a test and branch instruction.

4.3.2.6.16 Microsequencer Test and Order Structure

The microsequencer test and order structure (sheet 19) is similar to the RP test and order structure. Tests allow the microsequencer to test various conditions within the logic. Orders let the microsequencer exercise direct control over operation.

The microsequencer test structure consists of two 8:1 data selectors that are addressed by decoding three PROM bits (HSPROM05 through 07). The results of the test provide two outputs (LTEST8-F and LTEST10-17) to the CREG. These outputs are used to modify the least bit of the microsequencer address (see sheet 18). As a result of this logic, tests can only be made to even-numbered instruction locations.

The microsequencer order structure consists of three 3:8 decoder/demultiplexers and an eight-bit addressable latch. The decoder/demultiplexers produce pulse orders, while the eight-bit addressable latch produces level orders. Level orders are decoded from CREG bits 0 through 2 and 8 through 10 (HSCREG00 through 02, HSCREG08 through 10).

4.3.2.6.17 Byte Counter

The purpose of the byte counter (sheet 20) is to count the number of bytes of information to be transferred. The byte counter can be used for many purposes, including a read transfer count, a write transfer count, delay counts, and others. The byte counter consists of three, four-bit synchronous binary counters divided into a low and a high range. The byte count can be loaded from either the character bus (LCBUS0 through 7) or from the SCREG. Normally, the byte count is loaded from the R0 registers via the character bus. The CREG is used to load values for microsequencer timing and delays. In operation, the byte counter is loaded with the complement of the byte count, then the counter is toggled each time a byte is transferred. When the count reaches zero, the count is complete. Two indications of a completed byte count are used. The HBCSMLZERO signal is generated when the least eight bits of the byte counter are at zero. The HBCBIGZERO signal is generated when all 12 bits of the byte counter are at zero.

4.3.2.6.18 Sector Counters

The sector counters (sheet 21) count the sector pulses obtained from the disc drive. DIA-1 contains four counters, one each for disc drives 0 through 3. DIA-2 contains another four sector counters for disc drives 4 through 7. Only six bits of the counters are used, since the maximum number of sector pulses to be counted is 40. Each counter is reset to zero by the appropriate index pulse from the disc.

4.3.2.6.19 Gating for Unit Selected and Sector Counters

Sheet 8 shows the gating used to select the desired unit selected and sector count. The six-bit sector count (H0SECCNT through H3SECCNT), the HSEEKEND0 through HSEEKEND3 signals, and the UNITSELD0 through UNITSELD3 signals are applied to four buffer driver/receivers. The outputs are enabled onto the RP B file input (LEXT00 through 15) by either the LREADCNT01 or the LREADCNT23 signals. LREADCNT01 loads both disc zero and disc one into the B file. LREADCNT23 loads both discs two and three into the B file. The firmware must separate the desired disc's data by selecting either the most or the least significant eight bits for processing.

4.3.2.6.20 A Cable Registers, Drivers, and Receivers

The A cable registers, drivers, and receivers are shown on sheets 22, 23, and 24. A set of registers (sheet 22) is used to store the ten bits of data bus information (BUSTIT0 through BUSTIT9) to the disc drive. This data, which consists of commands to the disc drive, are output from the RP B file (LBIN06 through 15). The outputs are converted to standard differential signals by line drivers. Sheet 23 shows additional drivers for the unit select, open cable, and tag signals. Refer to Table 4-5 for definitions of these signals.

Sheet 24 shows the input signals from the disc. The differential input signals are converted to standard logic levels by the line receivers. Refer to Table 4-5 for definitions of these signals.

4.3.2.6.21 B Cable Drivers and Receivers

Sheet 24 shows the B cable output drivers. This data consists of the write data and the write clock. Four separate sets of signals are produced, one for each disc drive, 0 through 3. The DIA-2 circuit card contains a similar set of drivers for disc drives 4 through 7. Sheets 25 and 26 show the B cable input receivers. Sheet 25 contains the logic for read data and the associated clocks. Unit select signals (LUNITSELD0 through LUNITSELD3) select which of the four sets of inputs are to be processed by the DIA-1. Sheet 26 shows another set of inputs, which includes the HSEEKEND, LUNITSELD, HINDEX, and the LSECTORP signals for disc drives 0 through 3. Refer to Table 4-5 for definitions of these signals.

4.3.3 Device Interface Adapter 2 (DIA-2)

4.3.3.1 Input/Output Signals

The DIA-2 card contains six connectors, J4 through J9. Four of these connectors, J6 through J9, are for B cable connection to disc drives 4 through 7. The input/output signals for the B cable are covered in Table 4-4, Table 4-5, and Figure 4-6. The remaining two connectors, J4 and J5, are fully described in 4.3.2.5. In addition to these connectors, DIA-2 is also plugged into the SelBUS and receives power and ground signals from the SelBUS. Refer to drawing 130-103291 in the Drawings Manual for details of DIA-2.

4.3.3.2 DIA-2 Interface Logic

The DIA-2 interface logic is identical to parts of the DIA-1 circuit card logic. Refer to the similar logic descriptions for the DIA-1 circuit card.

4.3.4 Firmware Description

Two sets of firmware, or control programs, are used in the UDP. One program is stored in the regional processor and controls the overall operation of the system. The other is stored in the DIA-1 circuit card and controls the routine of data transfer to and from disc.

4.3.4.1 Regional Processor Firmware

4.3.4.1.1 General

The universal disc processor RP firmware is a control program that is permanently stored in PROMs mounted on the RP circuit card. The firmware consists of a set of addressble 32-bit control words stored in the PROMs. When a particular instruction is addressed, the instruction is read from the PROMs into a register, the bits are decoded by hardware, and the instruction is executed.

The firmware is written and assembled using the file definitions for an assembly language predefined by Gould. This assembler language is referred to as RPUDEF. A complete description of this language may be found in the RPU User Reference and Design manual.

The firmware listing provides both the source statements and the machine code microinstructions generated by the source statement. In addition, numerous comment lines, notes, and cross-reference information are included in the listing. To read the listing for testing or trouble shooting purposes, it is necessary to read and understand the description of RPUDEF provided in the RPU User Reference and Design manual.

4.3.4.1.2 Firmware Mnemonics

A variety of mnemonics is used in the firmware program listing. An alphabetical glossary of these mnemonics may be found in Appendix A at the back of this volume. Several types of mnemonics are listed. Pulse order or level order mnemonics are used by the assembler to call up various pulse and level orders. Mnemonics are also assigned to tests that are used by the assembler to identify the tests and set the test bits into the microinstruction. Some mnemonics are assigned to fixed values for programmer convenience. These mnemonics, which refer to a fixed value, are variously called constants or literal equates. In the cross-reference listing at the end of the listing, they are referred to as variables, since their value can be assigned by the microprogrammer. Once values are assigned, however, they become fixed and are therefore constants. Mnemonics are also assigned to registers. Finally, and perhaps most importantly, mnemonics are assigned as subroutine names. The programmer uses the names, or labels, to address a subroutine for branching. The assembler converts the label mnemonic to a microprogram sequential address.

4.3.4.1.3 Microinstruction Definitions

The 32-bit micorword used by UDP firmware is divided into a variety of fields for different purposes. These fields are decoded by hardware in the regional processor and executed, using the test and order structure. A full description of the microinstruction is found in the RPU User Reference and Design manual. Table 4-9 shows the general layout of the instruction word for reference. Refer to the RPU manual for the details of each field.

4.3.4.1.4 Program Flow

A set of generalized flow diagrams is given in Appendix F. These diagrams show the major subroutines handled by the firmware. In the interest of clarity, the flow diagrams have been simplified. For precise details of the program, refer to the program listing.

Table 4-9
Microinstruction Bit Definitions

	Normal			Wait	Branches	
Fields	Bit	ALU Function and Order Execution	Literal Load	Wait for Test Condition Met or Wait Enable Order	Long or Short Direct Branch	Long or Short Indirect Branch Order Two Microinstructions. Required First Order - Second Standard Branch
F0	00 01	Instruction sequence control (normal = 00)	Instruction sequence control (normal = 00)	Instruction sequence control (wait or wait enable order = 01)	Instruction sequence control (short = 10 and long = 11)	Instruction sequence control (normal = 00)
	02	Enable order control (bit 02=1)	Not used (zero)	Enable order control (wait = 0; wait enable order = 1)	Not used (zero)	Enable order control (bit 02 = 1)
	03	Select control ROM/RAM (bit 03 = 0, CROM) (bit 03 = 1, CRAM)	Select control ROM/RAM (bit 03 = 0, CROM) (bit 03 = 1, CRAM)	Select control ROM/RAM (must be same as previous instruction)	Select control ROM/RAM (bit 03 = 0, CROM) (bit 03 = 1, CRAM)	Select control ROM/RAM (bit 03 = 0, CROM) • (bit 03 = 1, CRAM)
F1	04 05 06 07	Register B address	Register B address	*(Register B address)	*(Register B address)	*(Register B address)
F2	08 09 10 11	Register A address	Literal value (bits 00-11) complemented (to be stored in addressed B register)	*(Register A address)	Long branch address (bits 00-03) *(register A address if short BR)	Register A address
F3	12	Set/reset level order		Order address for wait enable order	Long/short branch address (bits 04-11)	Order address = $1A_{16}$
	13 14 15	Order address (levels and pulses)				
F4	16 17 18 19					
F5	20 21	A/B mux control	A/B mux control (bit 20=1, bit 21=0)	*(A & B reg input control)	*(A & B reg input control) (bit 22 = 1 if long branch)	*(A/B mux control)
	22	Inhibit RA write (bit 22 = 1)	Inhibit RA write (bit 22 = 1)			Inhibit RA write (bit 22 = 1)
	23	Inhibit RB write (bit 23 = 1)	Inhibit RB write (bit 23 = 0)			*(Inhibit RB write)
F6	24 25 26 27	ALU function and carry source	Not used (zeros)	High/low test condition (high, bit 24 = 1)	High/low test condition (high, bit 24 = 1)	*(ALU function and carry source)
F7	28 29 30		Literal value (bits 12-15) complemented (to be stored in addressed B register)	Test structure mux address	Test structure mux address (unconditional = CE) (If bit 25 = 0 test group A) (If bit 26 = 0 test group B)	
	31	Save ALU status (bit 31 = 1)				

* Optional use for secondary functions.

Note: An indirect branch requires two microinstructions. The order microinstruction is shown in the column at the far right and is followed by a standard branch microinstruction, as shown to the immediate left.

4.3.4.1.5 Test and Order Structure

Refer to Appendix A for definitions of test and order mnemonics.

4.3.4.1.6 Register Definitions

Refer to 4.4.1.4 for register definitions.

4.3.4.2 Microsequencer Firmware

4.3.4.2.1 General

The microsequencer firmware is a control program permanently stored in PROMs mounted on the DIA-1 card. The firmware consists of a set of addressable 20-bit control words stored in the PROMs. When a particular instruction is addressed, the instruction is read from the PROMs into a register, the bits are decoded by hardware, and the instruction is executed.

The firmware is written and assembled using the file definitions for an assembly language predefined by Gould. This assembly language is referred to as SQDEF and is unique to the universal disc processor.

4.3.4.2.2 Assembly Language Description

The language SQDEF is similar in many respects to RPUDEF but with a number of major differences. The principal differences are that the SQDEF

1. Uses a 20-bit control word instead of a 32-bit control word
2. Uses a different set of directives
3. Contains the branch address in the control word
4. Executes a test and branch by modifying the least significant bit of the control word
5. Is much simpler in scope that RPUDEF

4.3.4.2.3 SQDEF Directives

The SQDEF language uses four primary directives, T(...,...), B(...), LL(...,...), and ORDER. These directives are defined and examples are given in Table 4-10.

4.3.4.2.4 Firmware Mnemonics

A glossary of firmware mnemonics is found in Appendix B at the back of this volume.

Table 4-10
SQDEF Directive Statement Definitions

Directive	Description	Example and Explanation
T(...,...)	Test and branch to indicated address if test is not true. Branch to address +1 if branch is true	T(HBCSZERO,DO). Test for HBSCZERO: if false, branch to location indicated by label 'DO'; if true, branch to 'DO +1. T(LENDOFF,$). Test for LENDOFF: if false, branch to current address ($); if true, branch to current address +1. This type of instruction is equivalent to a wait, since a loop results until the test is true, then the program advances to the next step.
B(...)	Unconditional branch to address indicated.	B(READ.L). Branch unconditionally to the address associated with READ.L
LL(...,...)	Load literal into the indicated register.	LL(OB,0). Load the literal zero into the output buffer (OB). LL(BC,13). Load the value 13 into the byte count register (BC).
ORDER	Issue order.	XFR. Transfer data enable. SWRITE. Set the writer flip-flop.

4.3.4.2.5 Microinstruction Definitions

The 20-bit microword used by the microsequencer firmware is divided into a variety of fields for different purposes. These fields are decoded by hardware in the microsequencer and executed, using the test and order structure. Figure 4-9 shows the general layout of the microinstruction.

4.3.4.2.6 Program Flow

A set of generalized flow diagrams is given in Appendix E. These diagrams show the major subroutines handled by the firmware. The flow diagrams have been simplified for clarity and do not show precise details. To examine the firmware in detail, always refer to the firmware listing.

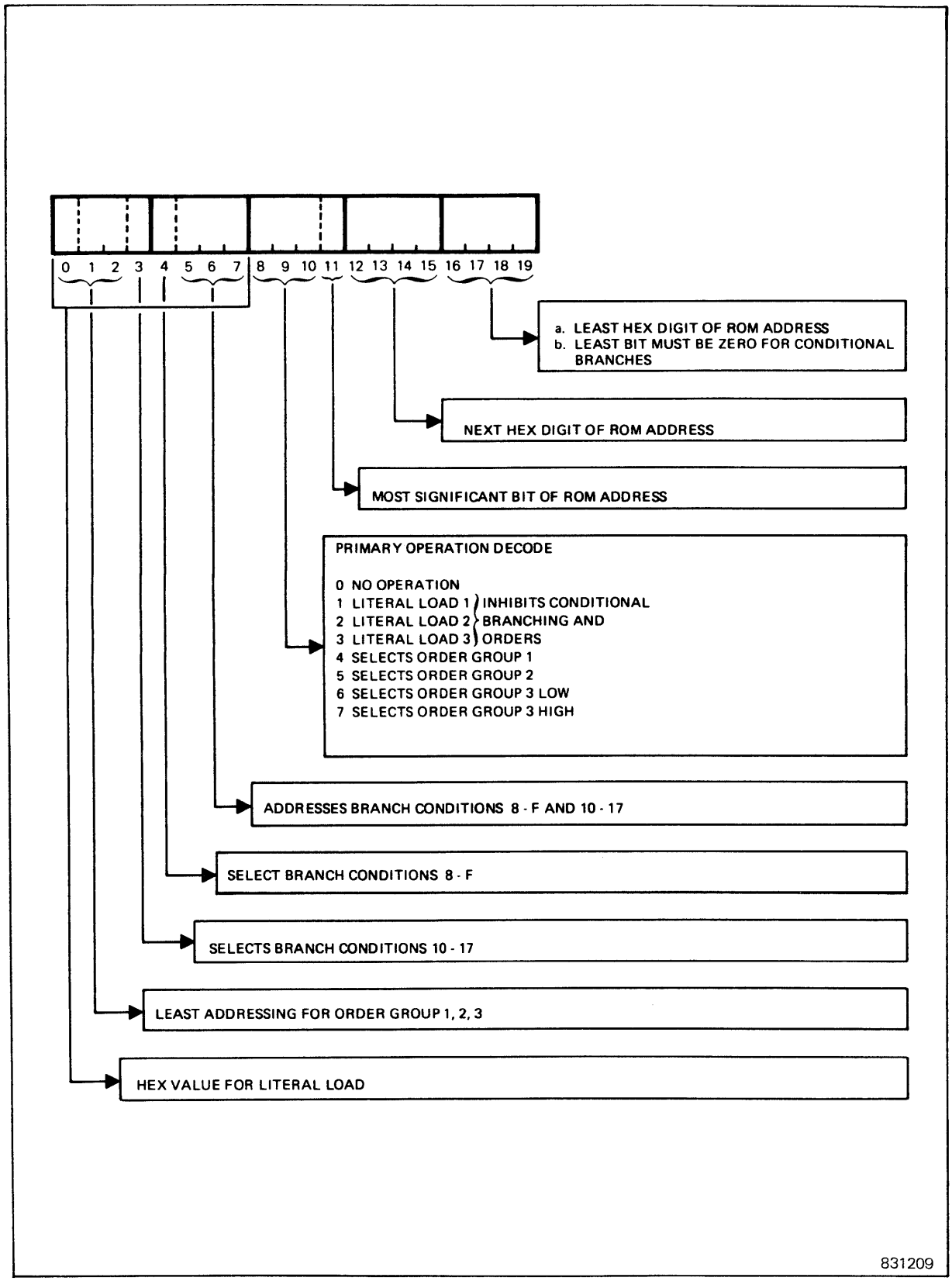


Figure 4-9. Microsequencer Microinstruction Bit Definitions

4.3.4.2.7 Test and Order Structure

Refer to Appendix B for definitions of tests and orders.

4.4 Universal Disc Processor Functional Characteristics

4.4.1 Universal Disc Processor Functional Organization

4.4.1.1 Functional Components

Figure 4-10 is a block diagram of the functional organization of the universal disc processor. The UDP (channel) portion of the system connects to the SelBUS and from one to eight disc drives that must be single-ported.

The channel is a block multiplexer channel. The structure in Figure 4-10 is hierarchial in that it is organized into levels of control from left to right. The channel is the highest level of control. The channel performs all channel scheduling functions. In addition, it manages or controls all interactions with the CPU or memory via the SelBUS.

The channel is connected to 16 subchannels identified as subchannel 0/1, 2/3, . . . 14/15, which correspond to drives 0, 1, 2, . . . 7. Each drive is connected to an odd and an even subchannel. Each subchannel has the facility to execute one I/O program. The assignments are permanent and defined by channel design. The subchannels may be considered to be I/O program contexts that are associated with a physical device on a permanent basis. All subchannels may execute programs concurrently. The channel activates a subchannel context, using the channel scheduling functions described below. Once the immediate processing requirements of the subchannel are complete, the context is stored in a memory buffer (subchannel storage) until further processing is required and the channel scheduler selects the subchannel. When no subchannel is active, the channel scans the subchannels to identify any that require service.

The subchannel controllers (SC) are the portions of a subchannel that are not stored in subchannel storage SST; they are the permanently resident portion of the context. The subchannel controllers consist of a small, one byte per subchannel, RP register allocation for the byte and a supporting set of microprogram routines. The microprogram routines are shared by the subchannels. The subchannel controllers serve two primary purposes: one is to eliminate a need to scan the SST (in main memory) and thus reduce memory accessing time. The second purpose is to reduce the time required to make a scheduling decision within the channel.

The permanently resident status byte of the subchannel controller describes the subchannel's needs for processing. For example, a given subchannel may be busy executing a program and, at the same time, waiting for a head seek to be completed in the associated drive. Based on this information, the channel scheduler can do something else until the seek completes. Other bits in the status byte have other similar meanings to the channel.

The device controllers contain the channel logic dedicated to a particular physical disc drive. This logic includes DIA port logic, shared DIA hardware, RP register space, and supporting microprogram routines. The port logic includes a sector pulse counter and facilities for multiplexing the drives' radial cable signals into the shared hardware. The RP register space describes the driver's operational state to the supporting shared microprograms.

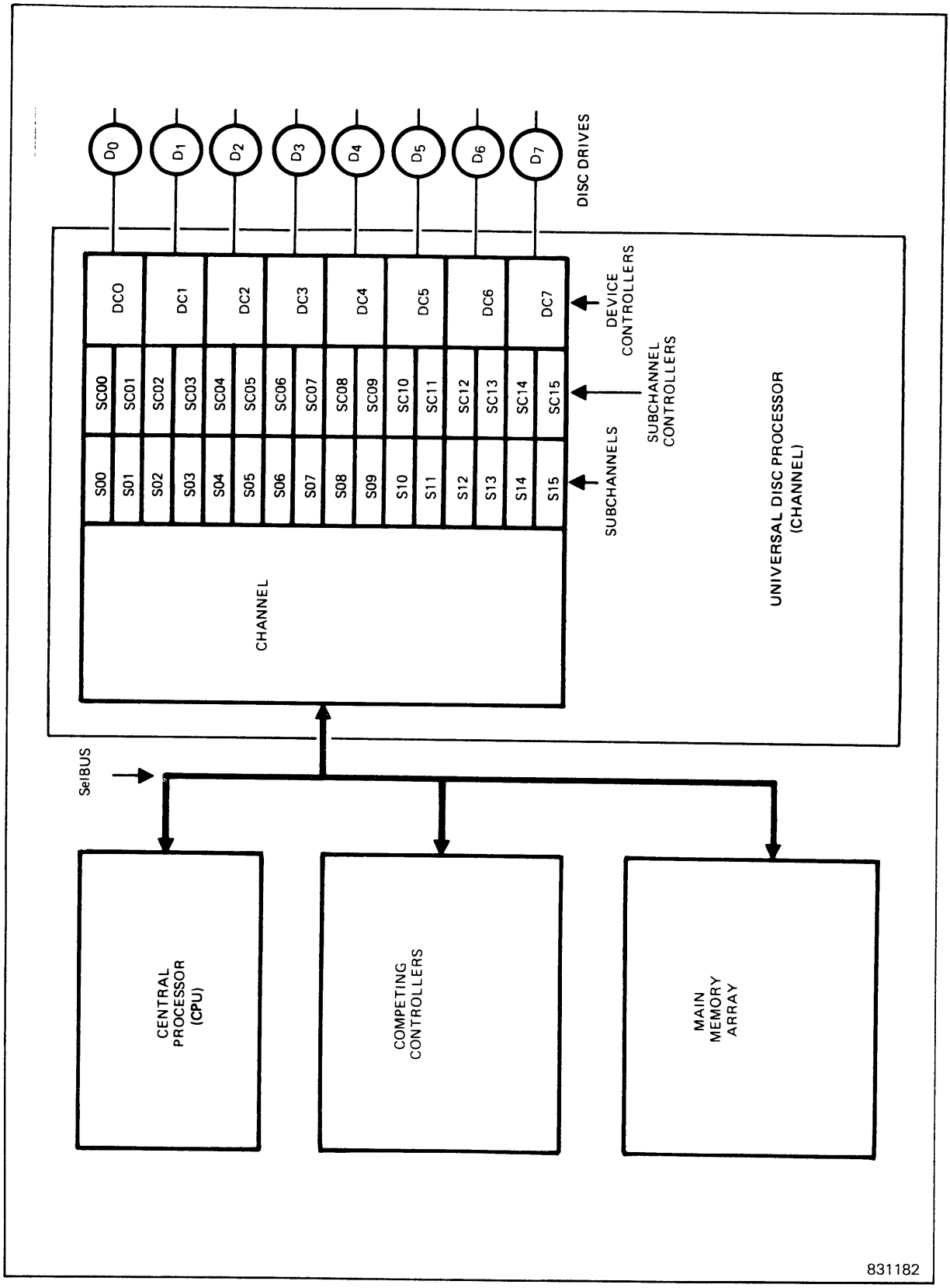


Figure 4-10. Universal Disc Processor Functional Organization

4.4.1.2 Functional Description

The functional characteristics of the device controllers are device-dependent. Three kinds of disc drives can be connected:

1. Single spindle disc drives that contain moving-head disc (MHD) only.

2. Single spindle disc drives that contain fixed-head disc (FHD) only.

3. Disc drives that contain both FHD and MHD (FMHD) on the same spindle.

Configuration data contained in disc tracks and the SST define which drive type is attached to the associated port.

During I/O program initiation, the subchannel/subchannel controller combination functions in a manner that is dependent upon the responses received from the device controllers.

1. An I/O program can be initiated in a subchannel whenever the subchannel is not already busy. The CPU instruction SIO is used to start the program. In general, an I/O program will execute until a conflict with the rules of channel protocol occur or until the I/O program reaches an end point.

2. When the disc is an MHD, both of the associated subchannels can be in execution concurrently. However, only one subchannel can be active at any point in time, and the associated device controllers and discs are dedicated to the first subchannel to be started. The device controllers and discs remain dedicated to the subchannel until it completes execution or releases the device controller and disc programmatically. This release occurs automatically when the I/O program terminates execution and attempts to present final status to the CPU when the subchannel having control executed a release (REL) command. The MHD cannot be shared on a concurrent basis, because head motion is generally required before a data access can occur.

3. When the disc is an FHD, both associated subchannels can be in execution concurrently. However, only one can be active at any given point in time. The device controller and discs are assigned to the active subchannel. The active subchannel always prefaces a data access with a cylinder and track selection, which is feasible since FHD seek times are, for practical purposes, zero.

4. When the disc is an FMHD, both subchannels can be in execution concurrently, but only one can be active at any given point in time. From a conceptual point of view, the FMHD is two devices: an FHD device and an MHD device. These two devices share a common interface. Consequently, it is not possible to perform data accesses to more than one of these two devices concurrently.

4.4.1.3 Main Memory Buffer

The UDP creates a main memory buffer during initialization, which must be local to the CPU and cannot be located in remote memories that might have long or unpredictable access times. A map describing this buffer is shown in Figure 4-11. The buffer consists of 224 words divided into four parts, as described in the following paragraphs.

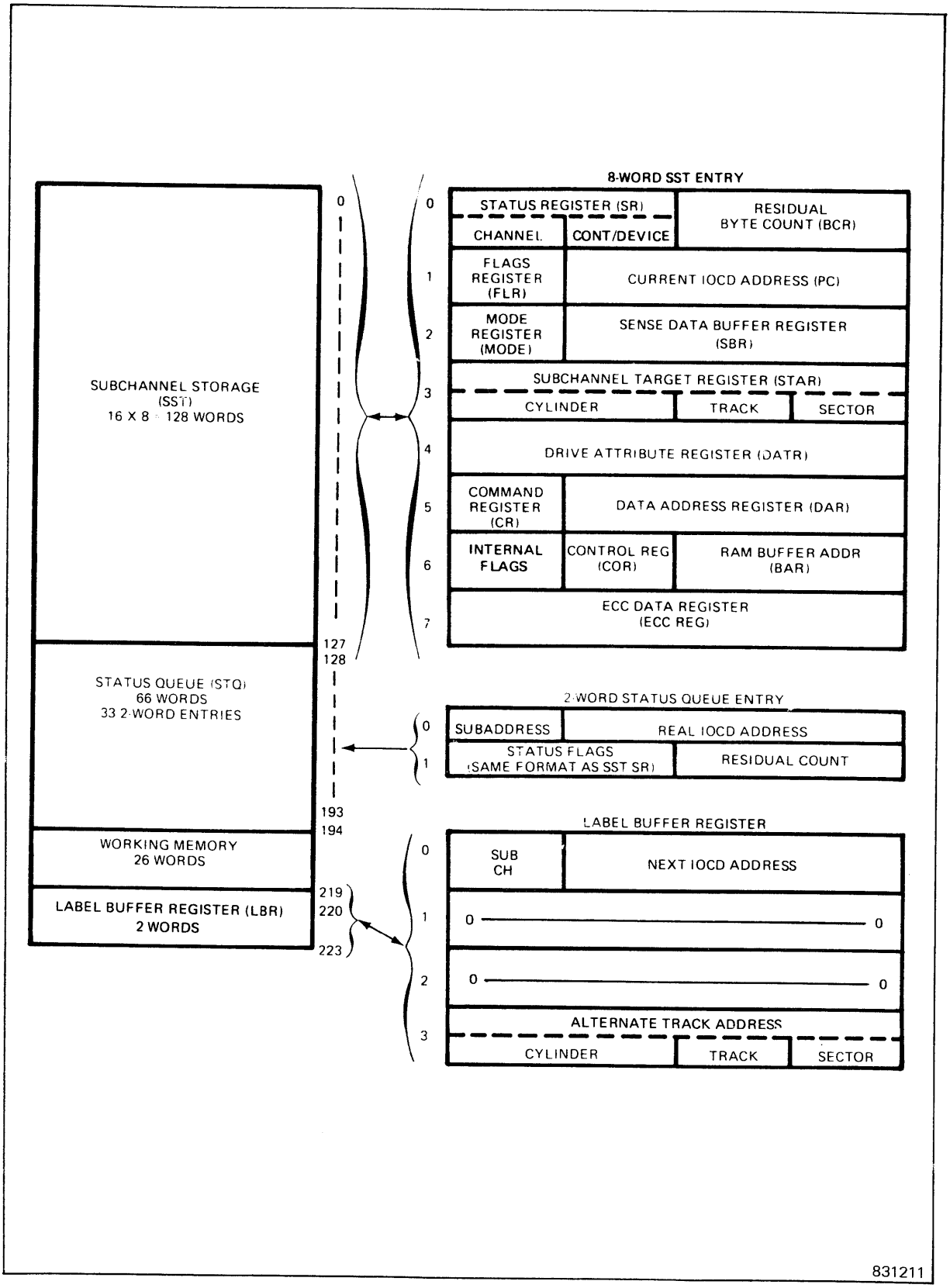


Figure 4-11. Main Memory Buffer Map

4.4.1.3.1 Subchannel Storage (SST)

This portion of the buffer contains the first 128 words. The SST consists of 16 sets of eight-word entries. Each entry provides storage for one subchannel. Figure 4-12 shows a breakdown of the various registers within the eight-word entry. Table 4-11 describes each register. During subchannel execution, the SST for the active subchannel is fetched and stored in the RP's subchannel image registers. At the end of execution, the subchannel image is updated and returned to the SST.

4.4.1.3.2 Status Ring Queue (STQ)

The next 66 words in the memory buffers are used as 33 doubleword storage locations for status. This allocation is used both as a status ring queue (STQ) and as a combination buffer for channel-to-CPU status transmission.

4.4.1.3.3 Scratchpad

The next 30 words of memory buffer are available to the UDP firmware for channel internal functions.

4.4.1.3.4 Label Buffer Register

The label buffer register consists of two words of label buffer data used in alternate track addressing and as a supplement to PCI.

4.4.1.4 Channel and Subchannel Registers

The RP register space consists of 64 half-word registers arranged into 4 groups containing 16 halfwords each. These registers are for practical purposes the total set of writable memory elements available within the channel. Most of this register space has been assigned to functions identifiable at the virtual machines level; the balance is used as working storage or to record channel operational states at the microevent level. The virtual registers fall into two groups, those that contain permanently resident information and those that are paged from subchannel storage (SST) during context switching. The latter is called the subchannel image (SI). The SI is filled from the SST when a subchannel is activated and saved in the SST when the subchannel is deactivated. The SI and each eight-word entry in the SST contain the register allocations shown in Figure 4-12 and described in Table 4-11. Table 4-12 shows the RP register layout. Register definitions are given in Table 4-13. Figure 4-13 provides the bit layouts of certain key registers.

4.4.2 Track Format and Timing

The UDP employs two track formats. One format, designated F16, provides 16 data sectors, where each data sector contains storage for 1024 data bytes. The second format, designated F20, provides 20 data sectors. where each data sector contains 768 data bytes. The F16 format is provided for long-term use and should be used by any software development not already committed to the F20 format. The F20 format is provided only to satisfy the sector size requirements of existing software.

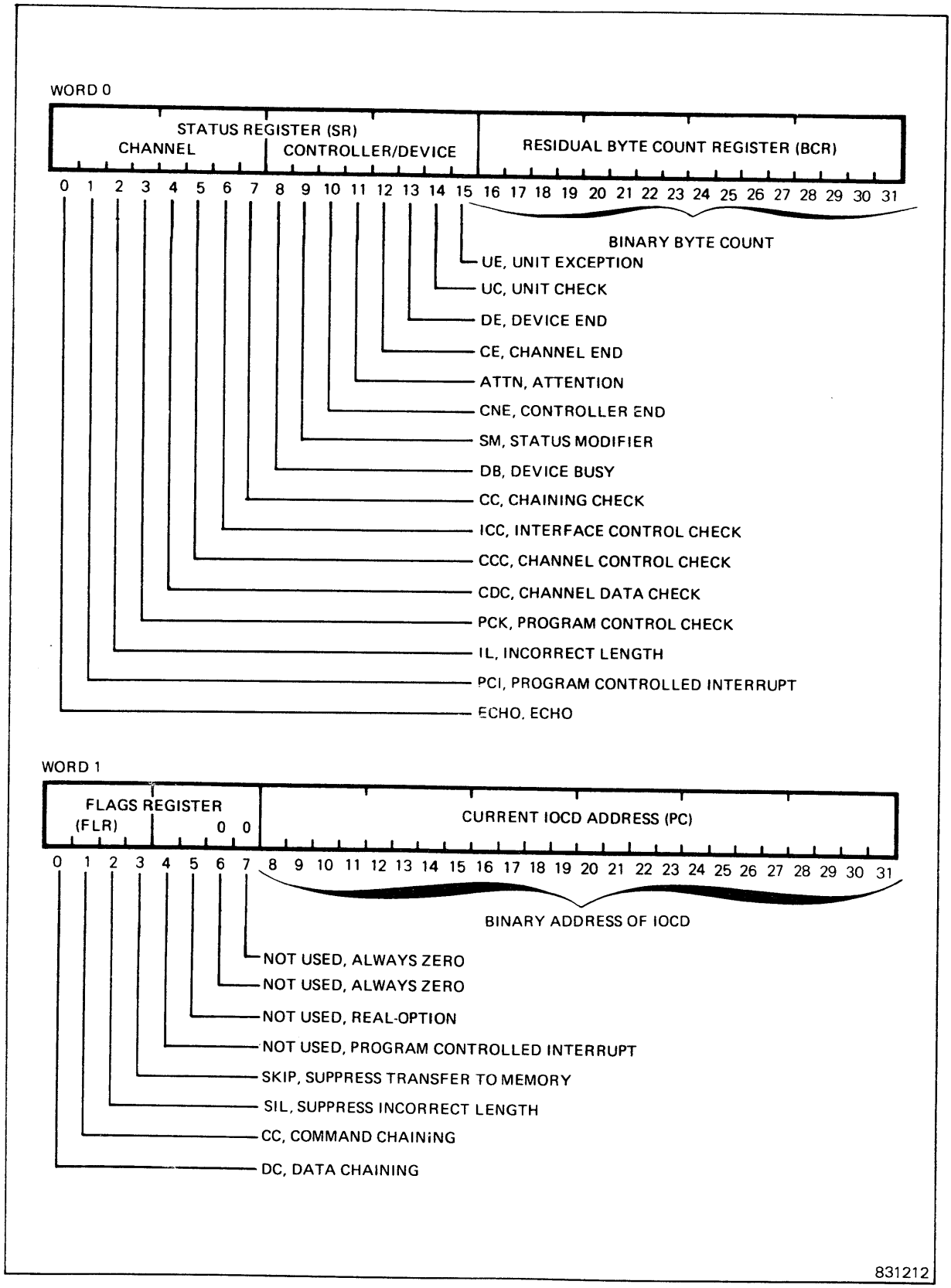
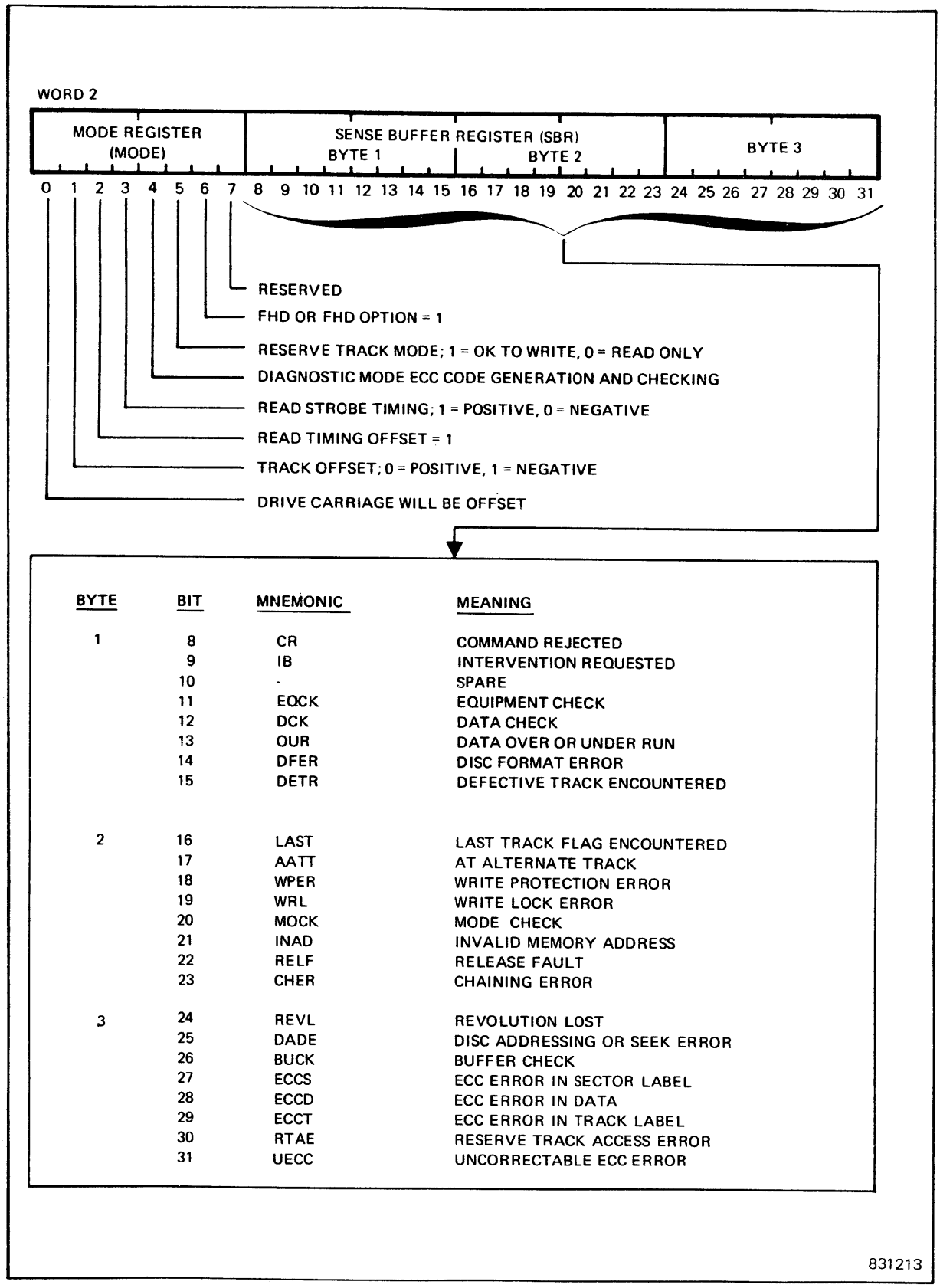


Figure 4-12. Register Layouts, Subchannel Storage and Subchannel Image (Sheet 1 of 4)



BYTE	BIT	MNEMONIC	MEANING
1	8	CR	COMMAND REJECTED
	9	IB	INTERVENTION REQUESTED
	10	-	SPARE
	11	EQCK	EQUIPMENT CHECK
	12	DCK	DATA CHECK
	13	OUR	DATA OVER OR UNDER RUN
	14	DFER	DISC FORMAT ERROR
	15	DETR	DEFECTIVE TRACK ENCOUNTERED
2	16	LAST	LAST TRACK FLAG ENCOUNTERED
	17	AATT	AT ALTERNATE TRACK
	18	WPER	WRITE PROTECTION ERROR
	19	WRL	WRITE LOCK ERROR
	20	MOCK	MODE CHECK
	21	INAD	INVALID MEMORY ADDRESS
	22	RELF	RELEASE FAULT
	23	CHER	CHAINING ERROR
3	24	REVL	REVOLUTION LOST
	25	DADE	DISC ADDRESSING OR SEEK ERROR
	26	BUCK	BUFFER CHECK
	27	ECCS	ECC ERROR IN SECTOR LABEL
	28	ECCD	ECC ERROR IN DATA
	29	ECCT	ECC ERROR IN TRACK LABEL
	30	RTAE	RESERVE TRACK ACCESS ERROR
	31	UECC	UNCORRECTABLE ECC ERROR

831213

Figure 4-12. Register Layouts, Subchannel Storage and Subchannel Image (Sheet 2 of 4)

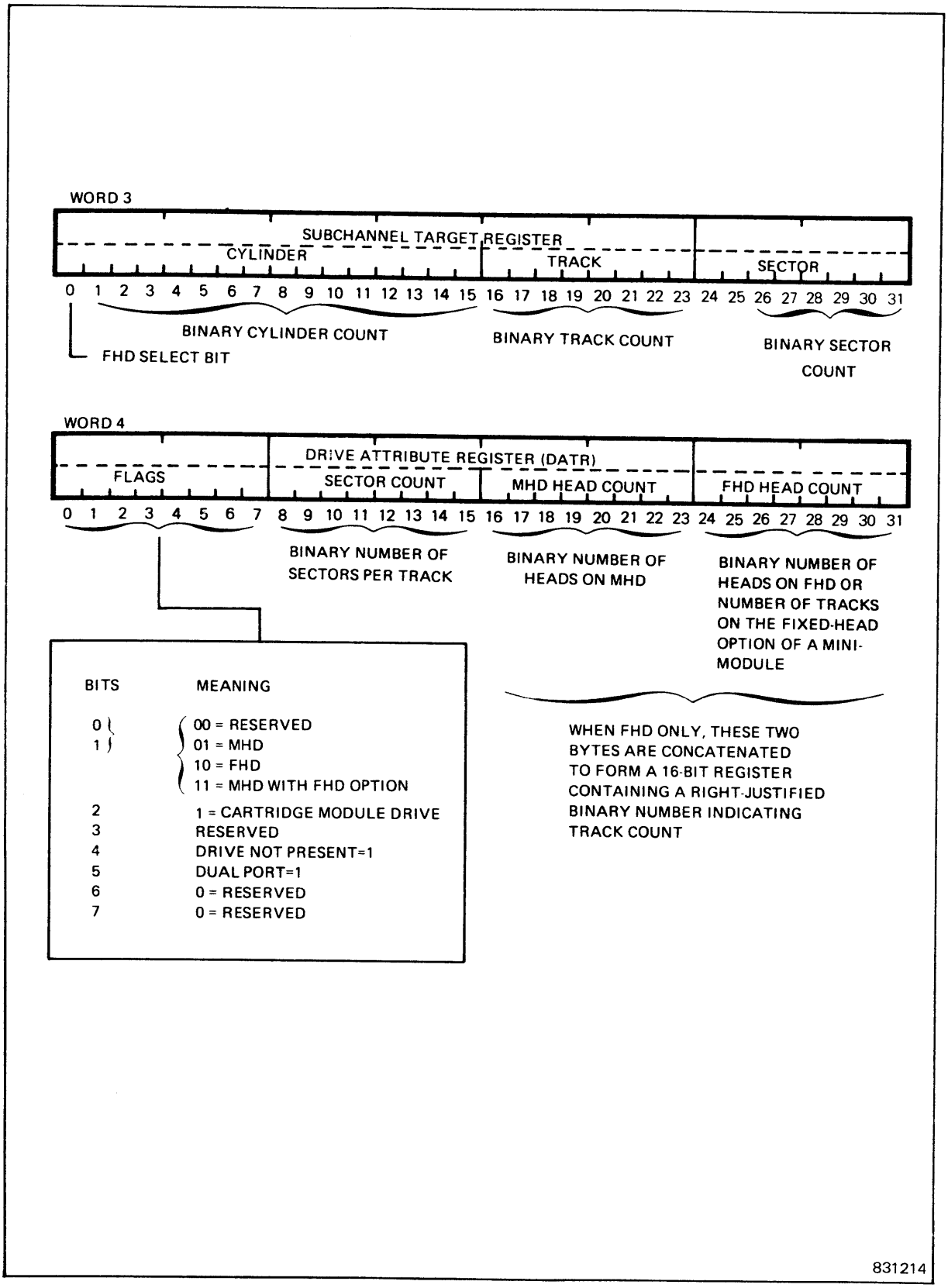


Figure 4-12. Register Layouts, Subchannel Storage and Subchannel Image (Sheet 3 of 4)

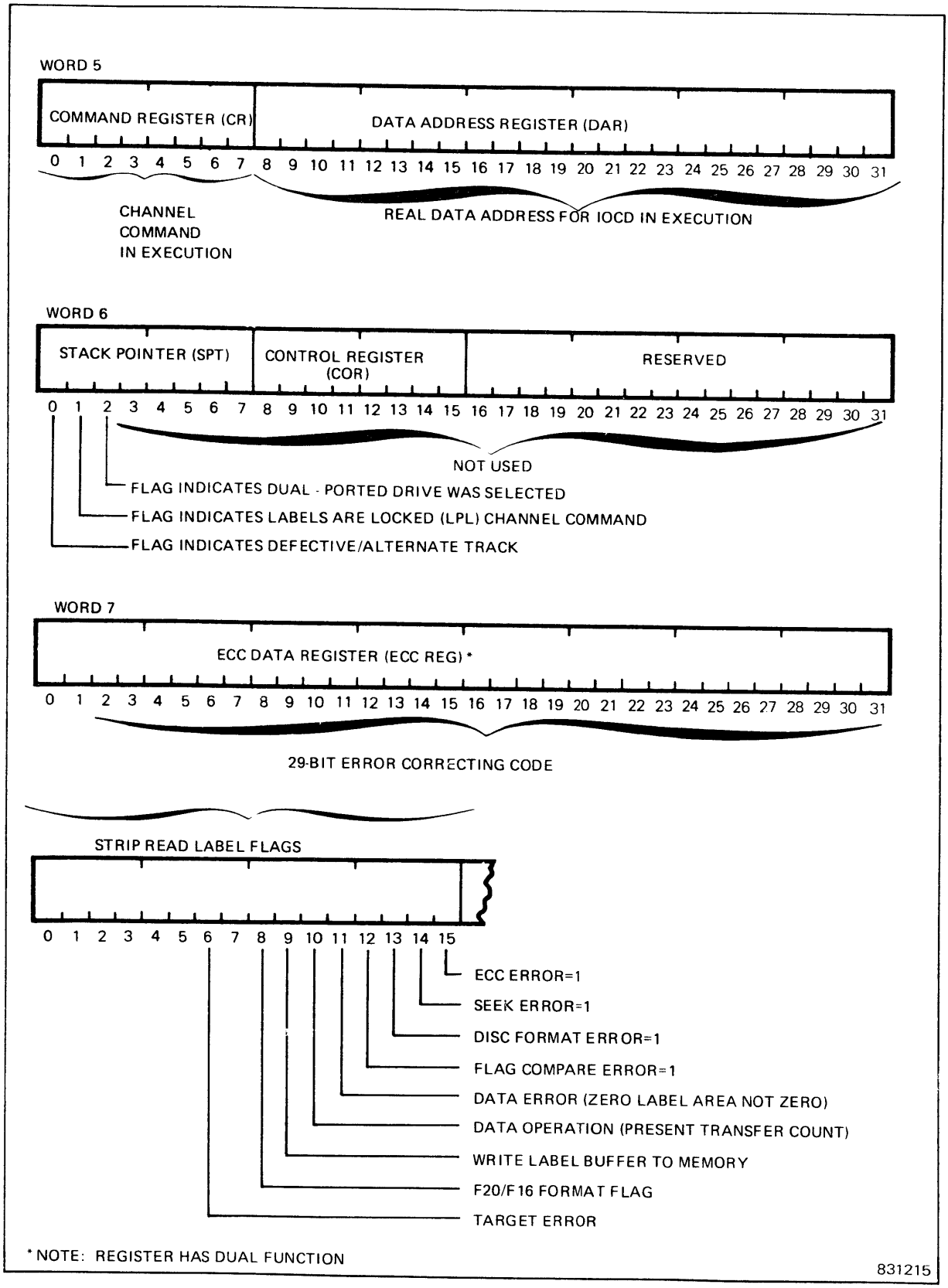


Figure 4-12. Register Layouts, Subchannel Storage and Subchannel Image (Sheet 4 of 4)

Table 4-11
Subchannel Storage (SST) and Subchannel Image (SI) Register Definitions

Name	Description
SR	Status Register. This register contains the two-byte status for the active channel.
BCR	Residual Byte Count Register. This register contains the residual byte count for the IOCD in execution.
FLR	Flags Register. This register is a one-byte register containing the flags taken for the IOCD in execution.
PC	Current IOCD Address. This register contains the address of the next IOCD to be acquired from main memory.
MODE	Mode Register. This register contains the mode bits for the subchannel in execution.
SBR	Sense Buffer Register. This three-byte register contains the sense data associated with the subchannel.
STAR	Subchannel Target Register. This 32-bit register contains the target for the subchannel in execution. The data consists of the cylinder, track, and sector address.
DATR	Drive Attribute Register. This register defines or notes the attributes of the associated disc drive and is filled during execution of the initiate channel (INCH) command.
CR	Command Register. This register is designed to hold the channel command in execution.
DAR	Data Address Register. This register contains the real data address of the IOCD in execution.
SPT	SPT. This register contains flags critical to the operation of the subchannel.
COR	Control Register. This register is a state of operation control byte used by the channel control logic to note subchannel operational states.
BAR	Buffer Address Register. This register is not used in the disc processor.
ECC.REG	ECC Register. This register contains error correcting code data from the preceding operations.

Table 4-12
RPU Register Layout (Sheet 1 of 2)

Register Number		RA-Name	Definition	RB-Name	Definition
High Bank					
0	*	IN.RET	Interrupt return	WORK0	Scratchpad
1	*	WORK1	Scratchpad	WORK1	Scratchpad
2		WORK2	Scratchpad	WORK2	Scratchpad
3		WORK3	Scratchpad	WORK3	Scratchpad
4		SCR0.1	Subchannel register	SCR2.3	Subchannel register
5		SCR4.5	Subchannel register	SCR6.7	Subchannel register
6		SCR8.9	Subchannel register	SCR10.11	Subchannel register
7		SCR12.13	Subchannel register	SCR14.15	Subchannel register
Subchannel Image					
8		SR	Status register	BCR	Byte count register
9		FLR	Flags register	IOCD	IOCD address
10		MODE	Mode register	SBR	Sense buffer register
11		STAR	Subchannel target register	STAR	Subchannel target register
12		DATR	Drive attribute register	DATR	Drive attribute register
13		CR	Command register	DAR	Data address register
14		SPT.COR	Stack pointer and control	BAR	Buffer address register
15		ECC.REG	Error correction code	ECC.REG	Error correction code

* Used for interrupts; not to be used for any other purposes.

Table 4-12
RPU Register Layout (Sheet 2 of 2)

Register Number	RA-Name	Definition	RB-Name	Definition
Low Bank				
0 *	IN.RET	Interrupt return	WORK0	Scratchpad
1 *	WORK1	Scratchpad	WORK1	Scratchpad
2 *	WORK2	Scratchpad	WORK2	Scratchpad
3 *	WORK3	Scratchpad	WORK3	Scratchpad
4	WORK4	Scratchpad	WORK4	Scratchpad
5	WORK5	Scratchpad	WORK5	Scratchpad
6	WORK6	Scratchpad	WORK6	Scratchpad
7	DSR	Drive state register	DSR	Drive state register
8	SBCH	Subchannel in execution	ICR	Interrupt control register
9 *	MX.DATA	Input data from CPU	MX.DATA	Input data from CPU
10	DUMMY.SEC	Dummy sector	DEV.ADR	Device address register
11	WORK7	Scratchpad	WORK7	Scratchpad
12	LINK	Link register	PIL	Priority interrupt level
13	RTN	Return address register	IOCR	I/O control register
14	RTN2	Return 2 address register	STATUS.ADR. CT	Status address count
15	SST.ADDR	Subchannel status ADDR	SST.ADDR	Subchannel status ADDR

* Used for interrupts; not to be used for any other purposes.

Table 4-13
RPU Register Definitions (Sheet 1 of 2)

Number	Name	Description
High Bank		
RA(0)	IN.RET	Interrupt Return. This register contains the address of the last microinstruction to be executed prior to an interrupt by the CPU. When the CPU interrupt is complete, this address is fetched, incremented by one, and loaded into the microinstruction addressing structure.
RAB(0-3)	WORK0-3	Work. These are general-purpose scratchpad registers.
RAB(4-7)	SCR	Subchannel Registers. This set of eight registers contains the 16 status bytes for subchannel 0 through 15. Even subchannel numbers are located in the most significant eight bits (left) and odd subchannel numbers are located in the least significant eight bits (right). The subchannel status bytes enable the channel to scan subchannel status quickly to make scheduling decisions without having to fetch the subchannel storage (SST) entries from main memory.
RAB(8-15)	Various	Subchannel Image. This set of registers is identical to the set of registers in the main memory subchannel status (SST). Subchannel image registers contain data for the current subchannel in execution. When the subchannel is selected, the subchannel image is filled from the SST memory location for that subchannel. At the completion of subchannel processing, the subchannel image is updated to show subchannel status at the end of processing. Then the subchannel image is stored in the SST for future use.
Low Bank		
RA(0)	IN.RET	Interrupt Return. This register contains the address of the last microinstruction to be executed prior to an interrupt by the CPU. When the CPU interrupt is complete, this address is fetched, incremented by one, and loaded into the microinstruction addressing structure.
RAB(0-6)	WORK0-6	Work. These are general-purpose scratchpad registers.
RAB(11)	WORK7	
RAB(7)	DSR	Drive State Register. This register allocates four bits of the register to each of the eight disc drives. The purpose of the register is to permit dual-port and dual subchannel I/O operations.
RB(8)	SBCH	Subchannel in Execution. This register contains the subaddress of the subchannel currently in execution.

Table 4-13
RPU Register Definitions (Sheet 2 of 2)

Number	Name	Description
RB(8)	ICR	Interrupt Control Register. This is a 16-bit register used to keep track of the current interrupt environment and other related functions within the microprocessor.
RAB(9)	MX.DATA	Multiplex Data. This is a 32-bit register used to hold a data word to be transferred to or from the SelBUS interface.
RA(10)	DUMMY.SEC	Dummy Sector. This register contains the next head and sector for writing labels and formatting disc.
RB(10)	DEV.ADR	Device Address Register. This register contains the channel address used during channel interrupt polling.
RA(12)	LINK	Link Register. This register is a return address register dedicated to handling defect skip.
RA(12)	PIL	Priority Interrupt Level. This register contains the priority interrupt level of the disc processor as assigned by the CPU's load RAM command.
RA(13)	RTN	Return Register. This register contains an RP microinstruction return address for use by the UDP firmware. The firmware will store an address in this register, branch to a subroutine (with a *LINK instruction), then return and fetch the address previously loaded into the RTM register. The address is then incremented by one and loaded into the program counter.
RB(13)	IOCR	Input/Output Control Register. This is a 16-bit register used to keep track of all noninterrupt related functions within the microprocessor.
RA(14)	RTN2	Return 2 Register. This register is the same as the RTN register.
RB(14)	STATUS ADDR.CT	Status Address Count. This register contains, in the right eight bits, the number of status queue entries. The left-most eight bits contain an index into the status queue. The status queue begins at the SST.ADDR plus 128 words. It should be noted that the status for subchannels on a given disc controller is stored in a circular status queue in order of occurrence. The status is presented to the CPU on a first-in, first-out basis.
RAB(15)	SST.ADDR	Status Stack Address. The SST.ADDR register is the memory address of the subchannel storage used for multi-disc operation and status queues. This address always points to an eight-word boundary, which is the first SST entry.

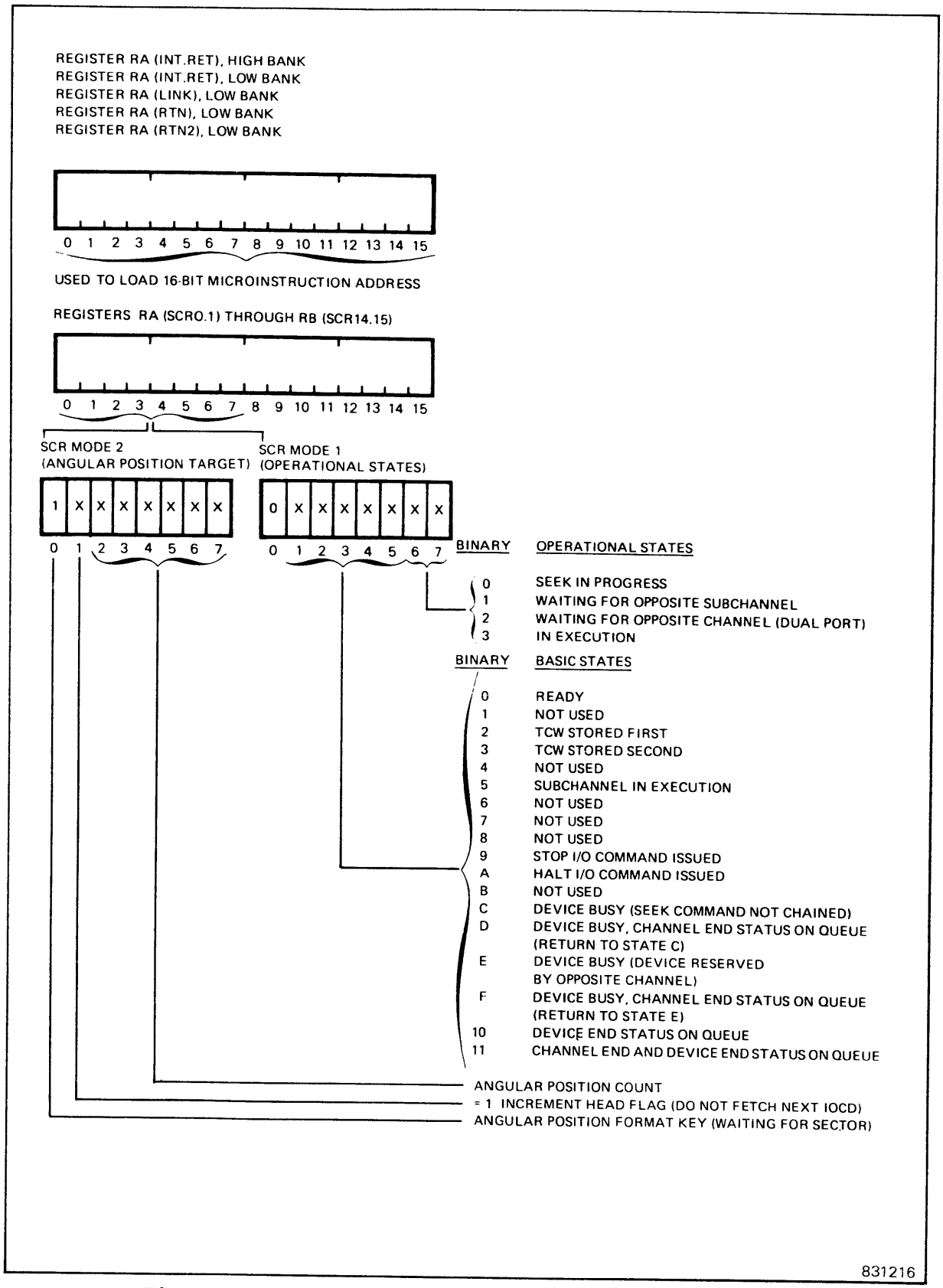


Figure 4-13. Regional Processor Register Layouts (Sheet 1 of 4)

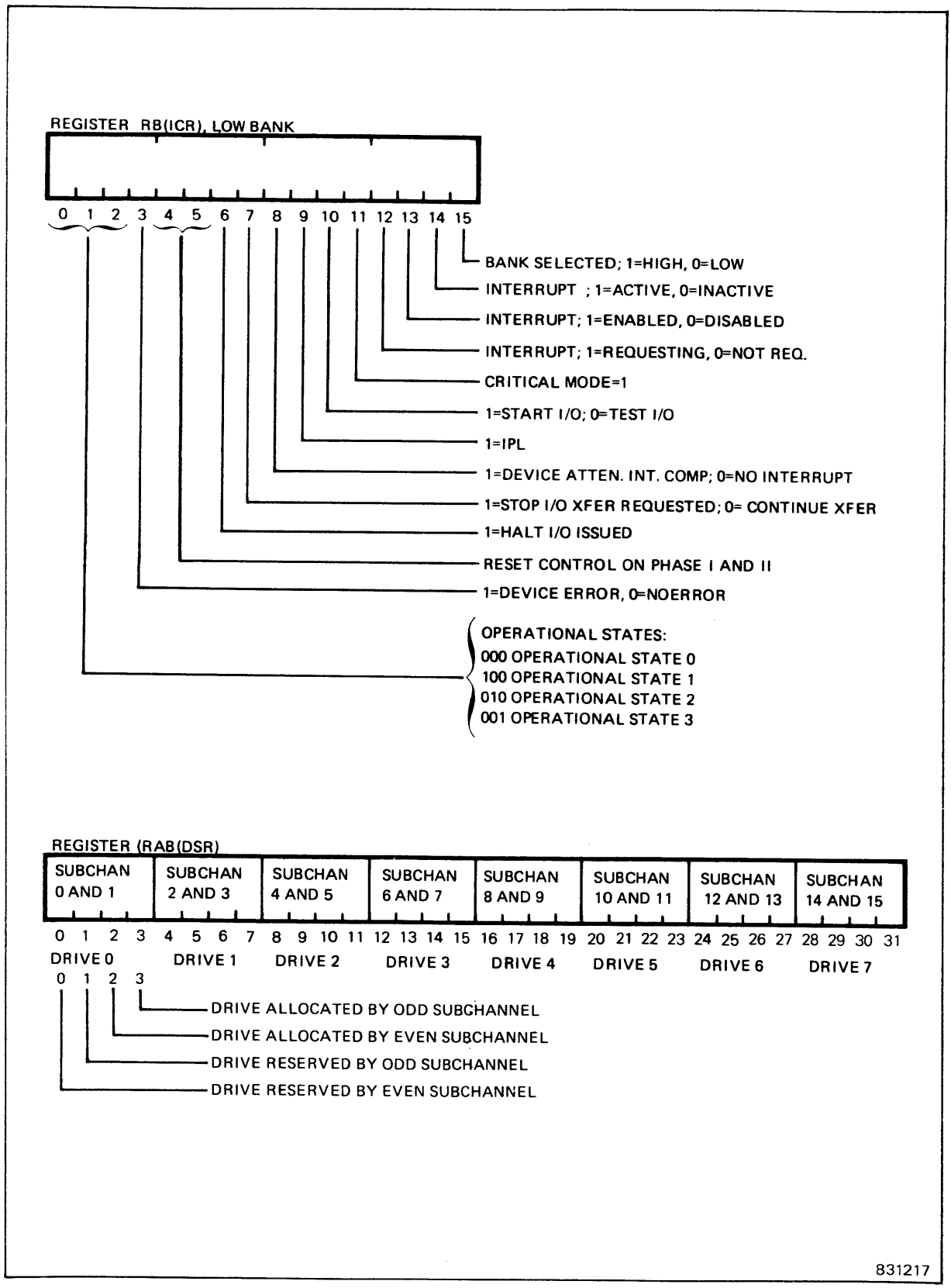


Figure 4-13. Regional Processor Register Layouts (Sheet 2 of 4)

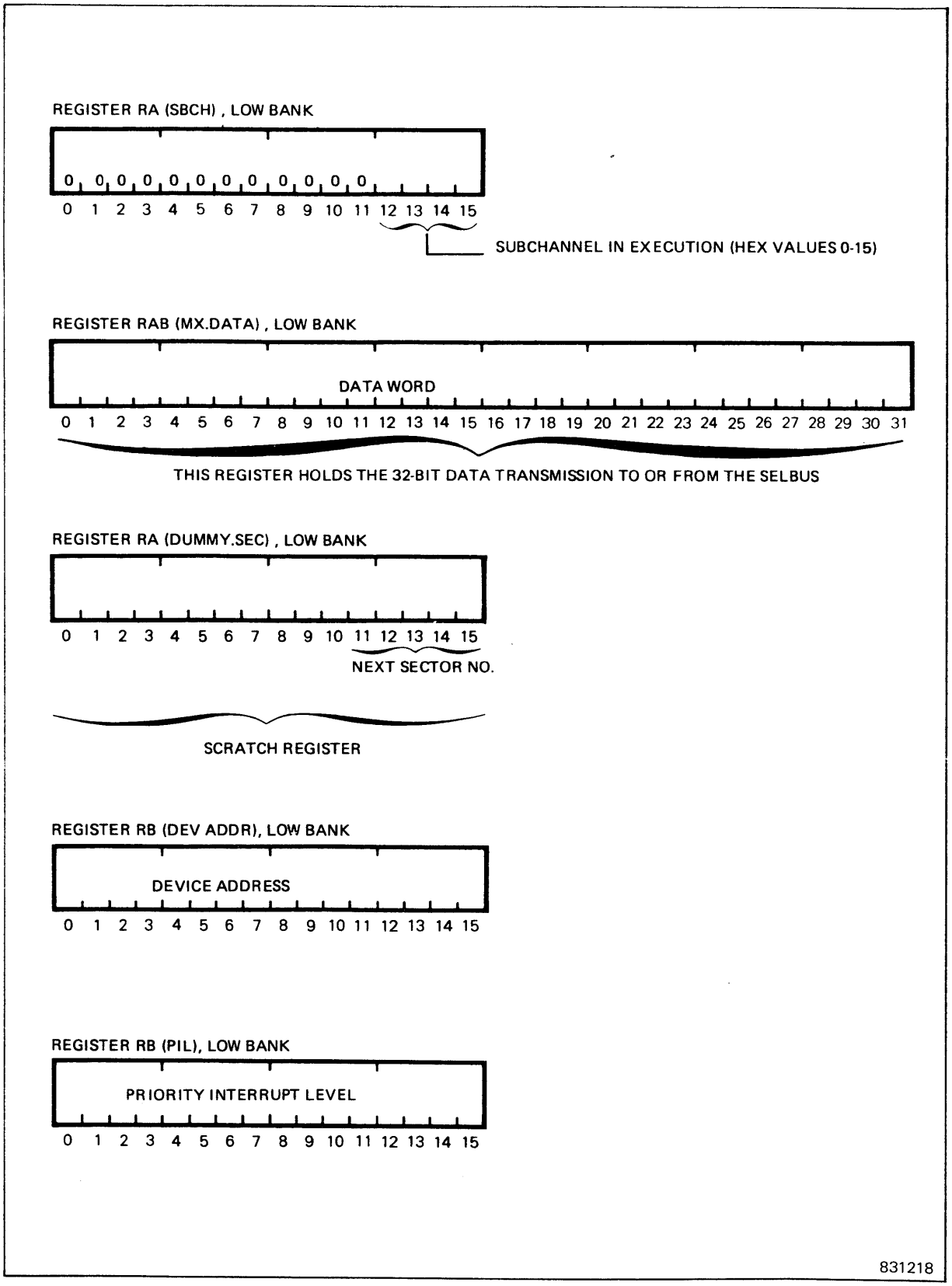


Figure 4-13. Regional Processor Register Layouts (Sheet 3 of 4)

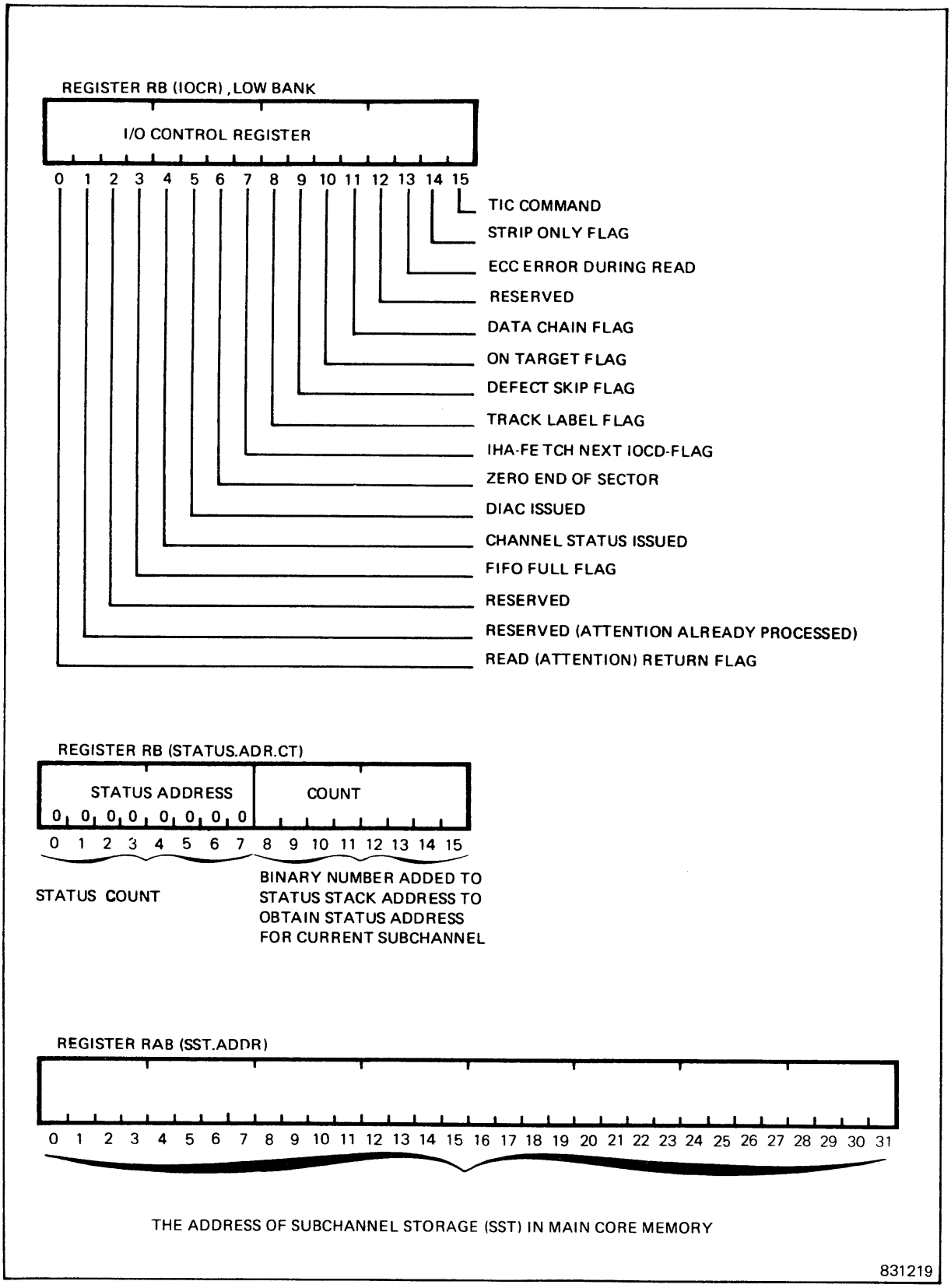


Figure 4-13. Regional Processor Register Layouts (Sheet 4 of 4)

Both F16 and F20 use a fixed sector track format, where each sector provides storage for a sector label and a data string as separately writable entities. Both track formats also include a track label. Two kinds of sectoring are used concurrently: hardsectoring and softsectoring.

4.4.2.1 F16 Format

4.4.2.1.1 F16 Hardsector Format

All disc drives attached to the UDP must be equipped with a facility to hardsector a disc revolution into 32 equal time periods. The beginning of each time period is signalled by the issuance of a short sector pulse from the disc drive via the drive B cable interface. The initial sector pulse is the drive index pulse. The issuance of these pulses may be keyed to the angular position of the disc so that variations in disc speed are eliminated.

The sector pulses will normally be generated in one of two ways: by detecting detents in an electronics cam attached to the drive media drive shaft or by servo clock pulses derived from a drive servo surface. The cam is normally based upon the optical or magnetic detection of cam lobes. Most UDP drives will be servo surface drives employing servo clock counters to generate the sector pulses.

Regardless of the implementation, the sector pulses partition the track into 32 segments, each approximately 630 bytes long (each track is intended to contain 20,160 bytes). Assuming that a disc speed of 3600 revolutions per minute applies, each period will have a duration of 520.833 microseconds, a subsector time. Two such periods, 1041.66 microseconds, are a sector time. The hardsector track format is shown in Figure 4-14.

Each of 16 sectors, labeled hexadecimally 0 through F, is partitioned into two subsectors. The first subsector in a sector is called the star subsector and denoted by an asterisk (*). The second subsector is called the prime subsector and is denoted by a prime mark (').

The port logic associated with each disc drive contains a six-bit sector counter that counts the sector pulses; the counter is reset by the index pulse at the beginning of each track. DIA logic makes the sector count available to the RP on demand; thus, the channel can determine the angular position of any attached disc drive by reading the associated sector counter. This capability forms the basis for angular position targeting (APT), which is described later.

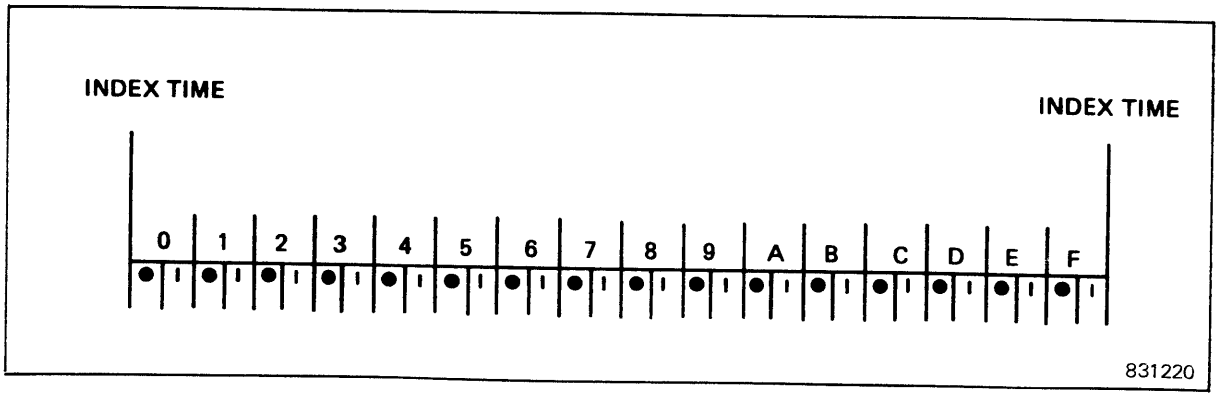


Figure 4-14. F16 Hardsector Track Format

4.4.2.1.2 F16 Softsector Format

All disc drives to be attached to the UDP must be equipped with a facility to generate and detect address marks (AMs); AMs are timing marks that are embedded in a data track.

The AM facilities are used to create the softsector format described in Table 4-14. The table has four columns. Column 1 is an item number. Column 2 identifies the function and includes the location of the first byte of a three-byte AM as a number in parentheses whenever the item contains an AM. Column 3 presents the track space in bytes used by the item. Column 4 presents the time required to traverse the item, assuming 800 nanoseconds per byte. The AM positions presented in column 2 are the byte position relative to index mark for the AM, assuming the bytes are labeled beginning with 1 (one) and ending with 20,160, the track capacity.

The timing is based upon the data rate that occurs when the disc drives are rotating at 1.03 x 3600 revolutions per minute, 3 percent fast. All time-sensitive channel logic functions are based upon this timing.

Table 4-14 lists 30 items. Item 1, the index mark, is provided only for reference, since only one edge of the associated pulse is used, i.e., zero track space is a valid assumption. Item 2 provides for head scatter, which is the variation in the angular position of data heads with respect to the servo head. This provision is required to achieve support for removable media-disc drives. Item 3, the track label, is described below. Item 4 is a timing pad, which allows time for controller functions, such as chaining and interaction with CPU functions. Timing pads are provided after each data sector as well. Item 5, data sector 0, is described below. Items 15, 26, and 37 are space allocations for defect skipping and are described further below. Item 39 is the end of the track timing pad, which provides for end of track chaining functions and for small differences in track capacity that might result from variations in drive characteristics.

The space allocation for defect skipping consists of three 128-byte track space allocations used to mask media flaws. This masking process requires the relocation of these skipzones to other positions in the track. In the UDP, the skip zones are placed as needed to mask flaws after the track label or any data sector. Future versions of the disc system will provide for placing the skip zones within the data field of a data sector. The skip zone positions shown in Table 4-14 will be used in tracks that have no maskable flaws. However, they can be moved to any position between items in the table, with the exception of items 1, 2, and 3, which will not be separated.

4.4.2.1.3 F16 Track Label Format

Table 4-15 presents the track label format (refer to item 3 in Table 4-14). The top part of the table contains three columns, item, function, and space allocation in bytes, respectively. Items 1 and 2 are presented for reference; the track label proper consists of items 3 through 24. Item 3 is the address mark and item 4 is a write splice zone that provides for write amplifier turn on. Item 5 is the read oscillator lockup zone; the read clock oscillator becomes synchronized during this time zone. Item 6 is the sync byte, which contains a single one bit, the pilot bit, used to locate a byte boundary. Items 7 and 8 are the track identifiers, which are right-justified binary numbers. Item 9 identifies the label as being a track label when it contains FF. Item 12 is four bytes of zero data that are reserved for future use. Items 13 through 18 are provided for the future implementation of defect skipping. Items 19 and 20 locate the alternate track when the accessed track is defective. Item 21 indicates the number of sectors in the track, 16 for

Table 4-14
F16 Softsector Track Format

Item	Function	Space Allocation (Bytes)	Timing (usec)
1	Index mark	0	0
2	Head scatter	20	16.0
3	Track label (21)	72	57.6
4	Timing pad	84	67.2
5	Sector 0 (177)	1123	898.4
6	Timing pad	84	67.2
7	Sector 1 (1384)	1123	898.4
8	Timing pad	84	67.2
9	Sector 2 (2591)	1123	898.4
10	Timing pad	84	67.2
11	Sector 3 (3798)	1123	898.4
12	Timing pad	84	67.2
13	Sector 4 (5005)	1123	898.4
14	Timing pad	84	67.2
15	Skip defect allocation	128	102.4
16	Sector 5 (6340)	1123	898.4
17	Timing pad	84	67.2
18	Sector 6 (7547)	1123	898.4
19	Timing pad	84	67.2
20	Sector 7 (8754)	1123	898.4
21	Timing pad	84	67.2
22	Sector 8 (9961)	1123	898.4
23	Timing pad	84	67.2
24	Sector 9 (11168)	1123	898.4
25	Timing pad	84	67.2
26	Skip defect allocation	128	102.4
27	Sector A (12503)	1123	898.4
28	Timing pad	84	67.2
29	Sector B (13710)	1123	898.4
30	Timing pad	84	67.2
31	Sector C (14917)	1123	898.4
32	Timing pad	84	67.2
33	Sector D (16124)	1123	898.4
34	Timing pad	84	67.2
35	Sector E (17331)	1123	898.4
36	Timing pad	84	67.2
37	Skip defect allocation	128	102.4
38	Sector F (18666)	1123	898.4
39	Timing pad	372	297.6
		Total 20,160	16.13 ms

F16. Item 22 describes the associated disc drive and configuration data. These four bytes are discussed in more detail below. Item 23 is a four-byte error correcting code. Item 24 is a write pad that causes a delay in turning off the write signal to the disc drive until all bits have been encoded and written. Item 25 is another write splice zone.

Table 4-15
F16 Track Label Format

Item	Function	Space Allocation (Bytes)
1	Index pulse	0
2	Head scatter	20
3	Address mark	3
4	Write splice zone, contains write splatter	12
5	Read oscillator lockup zone, bytes of all zero bits	16
6	Sync byte '80'; the first bit is the pilot bit	1
7	Cylinder number	2
8	Head or track number	1
9	Track label identifier, FF_{16} means track label	1
10	Flag byte 1	1
11	Flag byte 2	1
12	Spare, must be zero	4
13	Defect number 1 sector and byte position	2
14	Defect number 2 sector and byte position	2
15	Defect number 3 sector and byte position	2
16	Defect number 1 absolute position	2
17	Defect number 2 absolute position	2
18	Defect number 3 absolute position	2
19	Alternate cylinder number or return cylinder number	2
20	Alternate track number or return cylinder number	1
21	Data sector count	1
22	Configuration data	4
23	Error correction code	1
24	Write pad byte	1
25	Write splice zone; contains write splatter	5
	Total	92

Flat Byte 1 Format		Flag Byte 2 Format	
0	Good track	0	Write lock
1	Alternate track	1	Write protect
2	Spare track	2	Zero
3	Reserved track	3	Zero
4	Defective track	4	Zero
5	Last track	5	Zero
6	Zero	6	Zero
7	Zero	7	Zero

Items 10 and 11 are flag byte 1 and flag byte 2, respectively. Flag byte 1 is the track condition byte, and flag byte 2 is the access control byte. The bit utilization for these bytes is presented at the bottom of Table 4-15. These flags are described further in another section.

The track label uses a total of 92 bytes. The write splice zones have been made larger than necessary to provide for variation in address mark length, cable delays, logic delays, and write electronics performance.

4.4.2.1.4 F16 Sector Format

Table 4-16 presents the sector format that applies to all 16-track sectors. A sector has two separately writable parts, the sector label and the data. Each part is terminated with an error code. Items 1 through 20 form the sector label. Item 21 separates the label from the data. Items 22 through 27 make up the data field.

Items 1 through 4 are similar to corresponding track label items. Item 5 identifies the cylinder address as a right-justified binary number. The most significant bit of the cylinder address is set to one to identify FHD tracks. Item 6 is the head address, and item 7 is the sector number; both are right-justified binary numbers. The sector number always has values in the range 00 to 0F (hexadecimal). The head number has a range that is drive-dependent. Items 8 and 9 are flag bytes 1 and 2, respectively. The flag byte formats are defined at the bottom of the table. Flag byte 1 is the track condition byte, and flag byte 2 is the access control byte. Item 10 is a group of four spare bytes provided for future use.

Items 11, 12, and 13 provide for the future addition of defect skipping. Item 14 provides six additional spare bytes reserved as a contingency for the implementation of defect skipping. Items 15 and 16 locate the alternate cylinder and track when the track is flagged as defective or the return track when the track is labeled an alternate. Item 17 is the data sector count, 16 for F16. Item 18 describes the configuration for the associated disc drive. Item 19 is a four-byte error correction code. Item 20 is a write pad provided to delay turning off the drive write signal until the last ECC byte has been written. Item 21 is a write splice zone in which the write amplifiers are turned on or off, depending upon the operation.

Items 22 through 27 make up the data field, which may be written separately from the sector label. The error correction code is designed to cover a data length of at least 1024 bytes.

4.4.2.1.5 Configuration Data

All track and sector labels contain a four-byte configuration code that describes the disc drive containing the configuration bytes to the channel. These bytes are item 22 in Table 4-15 and item 18 in Table 4-16. The bytes are the same in all labels in a given disc drive.

Media certification software places the bytes in the labels during track test and formatting operations. The bytes are provided exclusively as a means of describing a disc drive to a subchannel. The configuration bytes are loaded into SST DATRs during channel initialization by the channel INCH command. The bytes in the SST must match the bytes in a disc track and sector labels when reading or writing formatted disc tracks.

The drive attribute register (DATR) and the drive attribute fields in a track or sector label have the same format and must have the same content after system initialization; this format is presented in Figure 4-12. DATR byte 1 defines the sector count as a right-justified binary number. Byte 2 defines the number of heads per MHD cylinder. Byte 3 defines the number of tracks in the FHD media. Bytes 2 and 3 are both right-justified binary numbers. When the drive is an FHD-only device, bytes 2 and 3 are concatenated to form a single 16-bit track count. For most drives, this will cause byte 2 to be zero; however, some drive manufacturers are planning to use track counts in excess of 256.

Table 4-16
F16 Sector Format

Item	Function	Space Allocation (Bytes)
1	Address mark	3
2	Write splice, contains write splatter	12
3	Read oscillator lockup bytes, bytes of zero	16
4	Sync byte '80'	1
5	Cylinder number	2
6	Head number	1
7	Sector, 00 to $0F_{16}$	1
8	Flag byte 1	1
9	Flag byte 2	1
10	Spare, all zeros	4
11	Defect sector and byte position	2
12	Defect sector and byte position	2
13	Defect sector and byte position	2
14	Spare, zero bytes	6
15	Alternate cylinder number or return cylinder number	2
16	Alternate head number or return head number	1
17	Data sector count	1
18	Configuration data	4
19	Error correcting code	4
20	Write pad	1
21	Write splice zone	5
22	Read oscillator lockup bytes	16
23	Sync byte '80'	1
24	Data bytes	1024
25	Error correction code	4
26	Write pad	1
27	Write splice zone, write splatter	5
	Total	1123

	Flag Byte 1		Flag Byte 2
0	Good track	0	Write lock
1	Alternate track	1	Write protect
2	Spare track	2	Zero
3	Reserved track	3	Zero
4	Defective track	4	Zero
5	Last track	5	Zero
6	Zero	6	Zero
7	Zero	7	Zero

4.4.2.2 F20 Format

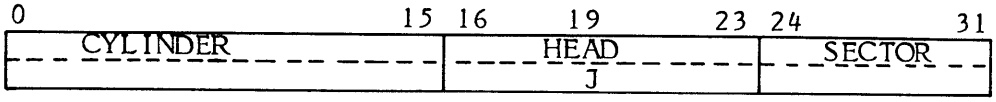
F20 track format is described in Figure 4-15 and Tables 4-17, 4-18, and 4-19. The track allocations are similar to those of F16. Notable differences are that there are 20 hard and soft sectors that each provide 768 bytes of storage. The F20 track format is less efficient than the F16 format.

4.4.3 Disc Storage Access Control

4.4.3.1 Sector Target Registration Operations

The material presented in this section deals with channel operation on the subchannel STAR register defined below. All descriptions are based upon the F16 track format. Operations on the F20 format or other formats are similar.

Each SST entry contains a four-byte sector target register (STAR) used to select the cylinder, track, and sector to be addressed in the associated disc drive. A STAR has the format shown below:



Bits 0 through 15 of the STAR are reserved for the cylinder address. The cylinder address is a right-justified, 10-bit binary number in the range of 0 to 3FF (hexadecimal). However, if the address is invalid, the drive will respond with a seek error.

Bit 19 specifies the lower volume of a cartridge module drive and is set whenever the fixed-head bit in the mode register is set, indicating the lower module.

The track number (TK) is also a right-justified binary number. The channel validates all head addresses before issuing head selects to a drive. Configuration information contained in the various track labels and the SST indicates the track count per cylinder (TC). The TK must be a binary number in the range 0 to TC-1. The sector number (SEC) is a right-justified binary number having a range of 0 to F or 0 to 13 (hexadecimal).

It should be noted that the J bit is only set in the sector target address register (STAR) and is totally transparent to software I/O operations.

The STAR points to the next cylinder, track, and sector to be accessed by a disc read or write operation. System reset functions reset the STAR to the all-zero state, selecting MHD cylinder, track, and sector zero. The disc seek command fills the entire STAR or selected portions of the STAR. The contents are then used by disc read and write commands to select the indicated disc sector.

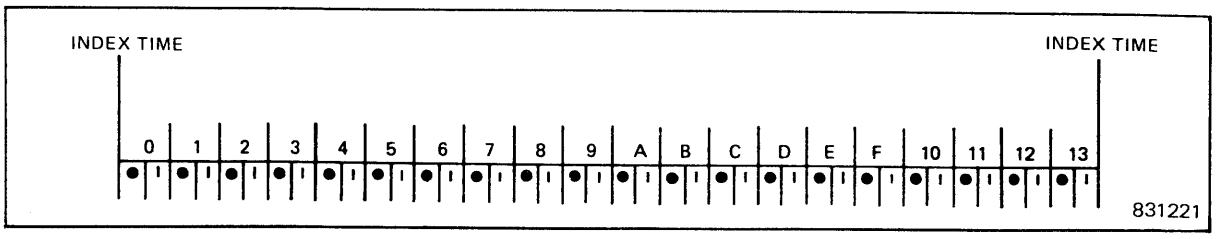


Figure 4-15. F20 Hardsector Format

Table 4-17
F20 Softsector Track Format

Item	Function	Space Allocation (Bytes)	Timing (usec)
1	Index mark	0	0
2	Head scatter	20	16.0
3	Track label (21)	72	57.6
4	Timing pad	97	77.6
5	Sector 0 (190)	867	693.6
6		97	77.6
7	Sector 1 (1154)	867	693.6
8		97	77.6
9	Sector 2 (2118)	867	693.6
10		97	77.6
11	Sector 3 (3082)	867	693.6
12		97	77.6
13	Sector 4 (4046)	867	693.6
14	Defect skip allocation	128	122.4
15		97	77.6
16	Sector 5 (5138)	867	693.6
17		97	77.6
18	Sector 6 (6102)	867	693.6
19		97	77.6
20	Sector 7 (7066)	867	693.6
21		97	77.6
22	Sector 8 (8030)	867	693.6
23		97	77.6
24	Sector 9 (8994)	867	693.6
25	Defect skip allocation	128	102.4
26		97	77.6
27	Sector 10 (10086)	867	693.6
28		97	77.6
29	Sector 11 (11050)	867	693.6
30		97	77.6
31	Sector 12 (12014)	867	693.6
32		97	77.6
33	Sector 13 (12978)	867	693.6
34		97	77.6
35	Sector 14 (13942)	867	693.6
36	Defect skip allocation	128	102.4
37		97	77.6
38	Sector 15 (15034)	867	693.6
39		97	77.6
40	Sector 16 (15998)	867	693.6
41		97	77.6
42	Sector 17 (16962)	867	693.6
43		97	77.6
44	Sector 18 (17926)	867	693.6
45		97	77.6
46	Sector 19 (18890)	867	693.6
47		404	323.2
	Total	20,160	16.126 ms

Table 4-18
F20 Track Label Format

Item	Function	Space Allocation (Bytes)
1	Index pulse	0
2	Head scatter	20
3	Address mark	3
4	Write splice zone, contains write splatter	12
5	Read oscillator lockup zone, bytes of all zero bits	16
6	Sync byte, '80', the first bit is the pilot bit	1
7	Cylinder number	2
8	Head or track number	1
9	Track label identifier, FF_{16} means track label	1
10	Flag byte 1	1
11	Flag byte 2	1
12	Spare, must be zero	4
13	Defect number 1 sector and byte position	2
14	Defect number 2 sector and byte position	2
15	Defect number 3 sector and byte position	2
16	Defect number 1 absolute position	2
17	Defect number 2 absolute position	2
18	Defect number 3 absolute position	2
19	Alternate cylinder number or return cylinder number	2
20	Alternate track number or return cylinder number	1
21	Data sector count	1
22	Configuration data	4
23	Error correction code	4
24	Write pad byte	1
25	Write splice zone, contains write splatter	5
	Total	92

Flag Byte 1 Format		Flag Byte 2 Format	
0	Good track	0	Write lock
1	Alternate track	1	Write protect
2	Spare track	2	Zero
3	Reserved track	3	Zero
4	Defective track	4	Zero
5	Last track	5	Zero
6	Zero	6	Zero
7	Zero	7	Zero

Table 4-19
F20 Sector Format

Item	Function	Space Allocation (Bytes)
1	Address mark	3
2	Write splice, contains write splatter	12
3	Read oscillator lockup bytes, bytes of zero	16
4	Sync byte '80'	1
5	Cylinder number	2
6	Head number	1
7	Sector, 00 to 13_{16}	1
8	Flag byte 1	1
9	Flag byte 2	1
10	Spare, all zeros	4
11	Defect sector and byte position	2
12	Defect sector and byte position	2
13	Defect sector and byte position	2
14	Spare, zero bytes	6
15	Alternate cylinder number or return cylinder number	2
16	Alternate head number or return head number	1
17	Data sector count	1
18	Configuration data	4
19	Error correction code	4
20	Write pad	1
21	Write splice zone	5
22	Oscillator lockup bytes	16
23	Sync byte '80'	1
24	Data bytes	768
25	Error correction code	4
26	Write pad	1
27	Write splice zone, write splatter	5
	Total	867

	Flag Byte 1		Flag Byte 2
0	Good track	0	Write lock
1	Alternate track	1	Write protect
2	Spare track	2	Zero
3	Reserved track	3	Zero
4	Defective track	4	Zero
5	Last track	5	Zero
6	Zero	6	Zero
7	Zero	7	Zero

As indicated in the sections that deal with track formatting, each track contains 16 sectors and a track label. Track label operations use only the CYL and TK fields in the STAR. When addressing sectors, the STAR SEC field must be loaded with a value in the range 0 to F or 0 to 13 (hexadecimal).

A single disc read or write command execution may encompass more than one sector and may extend over all sectors in the drive by using data-chaining and the TIC command.

The channel modifies the STAR at the end of each sector to reflect the address of the next sector to be accessed. When the operation is continuous across track boundaries, SEC is reset and TK is incremented at the end of the last sector in the cylinder. In this case, a seek to the next higher cylinder is initiated automatically. Most high-performance disc drives will complete a one-cylinder seek within the period defined by one disc revolution. The auto seek then causes the loss of one revolution.

When a disc operation ends at track end, i.e., when the byte count goes to zero at the last byte in the last sector of a track, the SEC is incremented at sector end, but the TK is not incremented. The data operation did not require track switching, so track switching was not executed. If the subchannel command chains to another read or write command, the operation will terminate, due to end-of-track conditions.

When a data operation ends at the end of cylinder, the SEC is incremented and the CYL and TK are not changed. In view of this and the operation described in the preceding paragraph, the SEC can be incremented to the value 10 (hexadecimal), which is an invalid sector number. This value is used and stored in the STAR to indicate that the last sector accessed was sector F (hexadecimal). This SEC value resolves an ambiguity that would occur if the SEC were to be reset at track end before operations continued. For example, if a multitrack operation continued to cylinder end and SEC was then reset, it would not be apparent from the STAR contents that the operation covered the last track in the cylinder. Yet, incrementing the track and cylinder address would be premature, since the need to move the heads to the next cylinder had not been established. SEC is rounded to zero only when the operation continues.

Track switching can be accomplished explicitly through the use of the increment head address (IHA) channel command. This command causes the TK field to be incremented and the next sequential track to be selected. If TK points to the last track in a cylinder when the IHA command is executed, the TK is reset, CYL is incremented, and a seek to the next cylinder is initiated. IHA also resets the SEC field, i.e., sets it to zero.

4.4.3.2 Track Label Flag Bytes

Each track and sector label contains two flag bytes, the track condition byte and the access control byte.

4.4.3.2.1 Flag Byte 1

Six of the eight bits in this byte have been assigned a function. The remaining two are required to be zero. The functions are assigned so that only one of bits 0 to 4 can be on at any given time. An attempt to write an invalid flag byte or read an invalid byte will cause a PCK and I/O program termination; the subchannel SBR disc format error bit (DFER) will also be set.

A good track bit indicates that the track is usable and is not dedicated to a special function. Most tracks will be flagged in this manner. The alternate cylinder and alternate track fields of good track must be zero filled. Refer to Table 4-26 for a description of disc space allocation.

A defective track is one that has been designated unusable by disc certification software. When an access to a defective track occurs, the alternate cylinder and track fields will normally be nonzero and point to an alternate track. If this is the case, a seek to the alternate track occurs automatically. The track thus located must be an alternate

for the defective track, or the I/O program will be terminated with a PCK and a sense data DADE indication. In addition, if the defective track contains a zero-alternate cylinder and track field, the I/O program will be terminated with a PCK and a disc format error (DFER).

When a defective track bit is accompanied by the last track bit, an alternate track seek is not initiated. Instead, the I/O operator is terminated with unit exception (UE) status and an SBR indication defective track (DETR) and last track (LAST). In this case, the alternate cylinder and track fields in the sector and track labels may be zero. These labels can be read by using the various label read commands. The last track flag, when used in this way, makes automatic seek to alternate tracks a user option.

A spare track is a track that has been set aside as unused. An attempt to perform data field read or write operations on a track flagged as spare will result in I/O program termination with PCK and a SBR DADE indication. A similar result will occur when sector label write operations are attempted; sector label read operations will not produce a PCK. A sector label write operation is possible only when the track label spare track flag has been reset, and one of good track, reserved track, alternate track, or defective track is set. When the track label is set to this condition, sector label 0 may be changed to a valid flag setting and, subsequently, sectors 1,2, and 3, etc. This conversion can be accomplished by a single multisector write label command.

The reserved track flag identifies reserved tracks; reserved tracks are tracks that contain information of a nonoperation type. These tracks will normally contain media defect information, disc utility programs, and, possibly, diagnostic programs for other system modules. The reserved tracks can be accessed only when a mechanical jumper, the reserved track jumper, is set to the on state and the subchannel mode reserved track mode bit is set to the one state. The jumper is part of the DIA configuration jumpers (see Figure 3-10). The mode bit can be set only when the switch is set. A reserved track is accessed controlled in the same manner as spare tracks, described in the preceding paragraph, i.e., data writes cannot occur unless reserved track mode is set. The track label must be changed before sector labels can be changed.

The alternate track flag is used to identify a track substituting for a defective alternate track. The alternate track field of the alternate-track track and sector labels must contain a nonzero value and point to the defective track for which it is being substituted. When a read or write operation is continuous across track end, an automatic seek is initiated to the track following the defective track, unless the last track flag is set in the alternate track. In this case, command execution is terminated, as described below in the section concerning the last track flag. However, the status will contain a UE bit, the sense data will contain the at alternate track (AATT) indication, and the STAR will point to the alternate track.

The last track byte indicates that multitrack operations are not to continue beyond the flagged track. When this flag byte is encountered, it will be accompanied by any of the other five defined flags. A last track operation will continue to the end of the track; however, track switching is suppressed. The STAR will contain the cylinder, track, and last sector processed indication. If F16 sector F (hexadecimal) was accessed, SEC will indicate 10 (hexadecimal).

The contents of flag byte 1 must be the same in all labels in a track. When the flag byte 1 in the track label is found to be different from a sector label flag byte 1 or any two sector labels are found to have differing flag byte 1 contents, a UC is generated with the SBR DFER indication and the I/O program is terminated.

4.4.3.2.2 Flag Byte 2

Flag byte 2 is the access control byte. This byte contains the write lock bit and the write protect bit; the remaining six bits must be zero.

The sector label write protect bit, when set to one, write protects the associated data in the same sector.

The write lock bit in one label write protects the succeeding label in the track. This bit write locks all labels in a track. Protection extends from the track label to sector 0 label, sector 0 to sector 1, etc., and from the last sector to the track label. When a write lock bit is set in a label and an effort is made to write the following label, the I/O program is terminated with a PCK and write protect error (WPER) SBR indication; a similar check occurs when an attempt is made to write protected data.

When write lock is operative in all labels in a track, it is not possible to change a label in the track. The data portions of the track are then protected, as indicated by the write protect bits in individual sector labels.

Write lock is made operative in one of several ways. Two configuration switches establish the write lock mode:

1. Write lock mode 0: disabled.
2. Write lock mode 1: permanently invoked.
3. Write lock mode 2: programmatically invoked after reset.

Write lock mode 0 (WRLM0) is used to disable the write lock checking. The write lock bits can be written and read; however, no write lock violation can occur when writing labels.

WRLM1 is used to permanently invoke the write lock function. In this case, operator intervention would be necessary to provide the channel with an ability to change write locked labels. This mode could be used in dual-ported disc systems to preclude a slave CPU from changing some portions of a disc file structure while giving the master CPU complete access to the same file structure. This would be accomplished by placing the master channel in WRL0 and the slave channel in WRL1.

In WRLM2, write lock is disabled after a power on reset or HCHNL and is subsequently invoked on a subchannel by subchannel basis when the command lock protected labels is executed in the subchannel.

WRLM2 can be used to selectively disable write accessibility for individual disc drives. In addition, when the drives are dual-ported, two or more drives can be write locked on a different basis, depending on the requirements of that system.

WRLM1 can be used to create a zone of read-only disc. This zone might contain the operating system or other programs and data that are not to be changed without operator intervention. This mode could be used to preclude clobbering the operating system when checking out programs that have direct access to the disc system.

4.4.4 Channel Modes and the Mode Register

Each subchannel contains an eight-bit mode register (MODE) that is used to define the operational mode for the associated subchannel. A list of the MODE bits and their functions appears on page 4-72.

Bit	Function
0	A 1 implies that the drive carriage will be offset.
1	This bit is effective only when bit 0 is in the 1 state; a 0 implies a positive track offset, and a 1 implies a negative track offset; a positive offset is an offset toward the next higher cylinder number.
2	A 1 implies a read-timing offset.
3	This bit is effective only when bit 2 is in the 1 state; a 0 implies that a positive-read strobe-timing adjustment will be used; a 1 implies that a negative-read strobe-timing adjustment will be used.
4	A 1 implies diagnostic mode ECC code generation and checking.
5	A 1 implies that reserved tracks can be accessed without causing an error; a zero implies that reserved track data cannot be written.
6	A 1 implies that the associated subchannel controllers will access the FHD or the FHD portion of an FMHD.
7	A 1 implies that the channel functions will use the RAM buffer for data operations, i.e., buffer mode is invoked.

When all MODE bits are set to the zero state, data operations occur between main memory and MHD; this setting might be considered to be the normal mode. An HCHNL places all channels in this mode. An HIO does not change the selected subchannel's mode.

4.4.4.1 Offset Modes and Their Uses

Some disc drives are equipped with features that allow I/O programs to adjust their read-strobe time and the radial position of the data heads. The features will normally be available in removable media drives and normally not be available in captive media drives; they are provided primarily to compensate for small differences in read-timing or head-position adjustment in drives that exchange media with other drives.

The normal setting for MODE bits 0 to 3 is 0; the bits must be 0 when a disc write operation is specified. Failure to observe this restriction will cause a PCK I/O program termination and a mode check (MOCK) SBR indication.

The offset bits provide a normal setting (all 0s) and eight offset positions. When attempting to recover data from a removable media drive, it is sometimes necessary to use the offset functions to obtain an error-free disc read operation. The procedure often used is the attempt to read three or more times at the normal position, then three or more times at both plus and minus track offset positions, then three or more times at both plus and minus read-time offset positions, and finally, three or more times at each of the four double offset positions. This process uses the four offset bits to select points in an eight-point rectangle surrounding the normal read position; head position offset is most likely to be successful, so it is used first. Since some errors are likely to be soft (not repeatable), three or more tries should be made at each position.

4.4.4.2 Reserved Track Mode

When a subchannel is operated in this mode, tracks flagged as reserved tracks may be accessed as if they were good tracks. A MODE reserved track mode bit can be set only when a DIA configuration switch is set to the reserved track mode enabled position. Reserved tracks are reserved for nonoperational or stand-alone maintenance functions.

4.4.4.3 Fixed-Head Disc Mode (FHDM) and Moving-Head Disc Mode (MHDM)

A subchannel operates in FHDM when the MODE FHDM bit is set to 1 and in MHDM when the FHDM bit is set to 0.

The FHDM is used as the basis for selecting the FHD position of a FMHD drive. When this bit is set to the 1 state, the channel interacts with the drive so the FHD portion of the drive is selected. During the data operations that follow, the STAR contents and the track labels read in the disc must be consistent with FHDM in that they must be flagged as being FHD parameters.

The channel scheduler is sensitive to the subchannel mode and characteristics of the attached disc drive. The subchannel mode, as described by the MODE contants, and the drive characteristics, as described by the DATR contents, form one basis for channel scheduling decisions. When a drive is an FHD-only drive, it is shared by a subchannel pair on an as-needed basis; an FHD drive cannot be reserved to a single subchannel or a subchannel pair. Access to the FHD is granted on a first-opportunity basis (first-opportunity refers to required-sector and disc angular position).

When a drive is an MHD-only drive, the subchannel and scheduler dedicate the disc to one of the subchannels in a subchannel pair. Processing is possible in the subchannel that does not possess the drive. However, in this case, it cannot be the processing that requires interaction with the drive.

The FHDM bit can be thought of in a different way than as previously described. For example, the UDP supports the cartridge module disc drive. This product appears to the software user as two separate devices, but it is really two moving-head discs, an upper module (removable media) and a lower module (captive media). Whenever the fixed-head bit is set to the zero state, the removable media will be selected, and whenever the FHD bit is set to a one, the lower module (captive media) is selected.

4.4.5 Channel Initialization

A UDP is always in one of four operational states (OSi) with respect to initialization; the states are

1. State 0: reset, waiting for a load RAM operation.
2. State 1: operational and waiting for an SST allocation.
3. State 2: operational, as a block multiplexer channel.
4. State 3: fault inoperational.

4.4.5.1 Master Reset and Operational State 0 (OS0)

A master reset causes the termination of any operation in progress and entry to OS0. The HCHNL instruction initiates a master reset. If any reset occurs while a disc write

operation is in progress, the DIA board reset is delayed until sector end is reached and a valid ECC code is written.

The master reset operation does not extend to the attached disc drives. Drive initialization is software controlled (the channel command XEZ rezero is used to initialize the disc drive attached to the executing subchannel). The master reset operation consists of a hardware reset, followed by the execution of a microprogram that initializes channel registers in preparation for channel startup. The channel interrupt is deactivated and disabled by the reset.

The channel remains in OS0 until a load RAM operation places the channel in OS1 or until an operational error causes entry to OS3. The channel will respond to all CPU generated service requests while in OS0. Only two service requests are valid when the channel is in OS0: load RAM and HCHNL. The HCHNL operation simply causes the channel reset procedure to be repeated with subsequent reentry to OS0. The load RAM operation causes the channel to enter OS1. Invalid AICT service requests are responded to in a manner that causes a CPU timeout and a system trap. All invalid service requests cause the channel to enter OS3. Invalid ARSTX sequences are responded to with an indication that the instruction is not supported.

4.4.5.2 Load RAM

The load RAM operation loads the channel IPL register with the interrupt level that is used in interrupt polling operations initiated subsequent to the load RAM operation. Execution of a load RAM operation in OS0 causes entry to state OS1.

A load RAM operation is valid when the channel is in OS2; however, its execution while in this state does not affect the channel's operational state. When the load RAM operation is executed in OS1 or OS2, the IPL register is filled without affecting any interrupt polling in progress when the load RAM was received. The channel design assumes that the load RAM operation will not be issued while an interrupt polling operation is in progress or while the interrupt is active. However, should this occur, the load RAM operation will occur normally and subsequent interrupt polling operations, other than the one in progress, will use the new priority level.

4.4.5.3 Operational State 1 (OS1)

OS1 provides for disc IPL read operations and completion of channel start-up. While in OS1, the channel functions as a selector channel, i.e., it is capable of executing a single I/O program. The channel cannot support multiple I/O program execution and context switching until a main memory space has been allocated to the channel for use as the SST.

OS1 imposes two restrictions on the execution of I/O programs. One is that the I/O programs must begin with either an IPL read command or with an INCH command that completes the channel initialization; the other is that the I/O program execution reaches the point that both CE and DE status must be available before status is presented to the CPU. The requirement that the I/O program begin with an IPL-read or INCH is enforced by causing an immediate I/O program termination when the requirement is not met. The I/O program termination is accompanied by entry into OS3. No attempt to present status is made in this case.

The channel is responsive to AICT service requests while it is in OS1. The responses and execution of AICT functions are the same as those that occur in normal operation. Since the load RAM function does not enable the channel interrupt, the channel will not attempt to interrupt the CPU while in OS1 unless the I/O program initiation was preceded by an enable interrupt operation. The I/O program was initiated as part on an IPL read operation and the interrupt will be enabled subsequent to the issuance of the IPL read command. In view of this, final status associated with the IPL read I/O program will be collected by an AICT sequence. The channel interrupt can be enabled prior to an I/O program that begins with an INCH command. When this occurs, the channel will attempt to present status using the channel interrupt.

I/O programs executed in OS1 must begin as described above with either an IPL read command or an INCH command. Either type of first command IOCD can use command chaining to attach I/O programs that have an arbitrary makeup. Data chaining cannot be used in the IPL read IOCD, specifying data chaining in the IPL read causes the IOCD to be treated as if it were invalid. Data chaining cannot be used in the INCH IOCD.

Execution of an INCH command completes channel initialization and causes the channel to enter OS2 during the execution period of the command. The INCH command is invalid when the channel is in OS2. The command assigns and initializes the SST. It enables the execution of multiple I/O programs. If a fault or programming error is detected during INCH command execution, the channel will enter OS3 without a status presentation. OS2 is reached only when INCH command execution is completely successful.

INCH command execution is described in detail in 4.4.10.1. This command transfers data from memory to the channel. The data includes the subchannel storage address and configuration data for each subchannel entry.

4.4.5.4 Operational State 2 (OS2)

Once the channel enters OS2 it is capable of context switching and multiple I/O program execution. It remains in OS2 until reset or until a fault causes entry to OS3. The INCH command is invalid in OS2; however, its use will not cause channel shutdown.

4.4.5.5 Operational State 3 (OS3)

OS3 is entered from any other OSi when the channel detects an error or fault that implies a system reset is in order. Typical of such faults are an SIO ARSTX service request while the channel is in OS0 and the occurrence of a memory read parity error while accessing the SST in OS2.

If an I/O program is in execution when the fault occurs, the I/O program is terminated without a status presentation. The fault does not change the state of operation for the channel interrupt level. This means that the interrupt will remain active if it was active prior to the fault and that the fault will not cause interrupt polling. However, the channel will respond to a request to deactivate the channel interrupt while in OS3. Should interrupt polling be in progress when the fault occurs, the polling is ended.

Once in OS3, the channel repeatedly presents the unsupported transaction status response to all ARSTX service requests except RSCHNL. RSCHNL or a power-on reset causes channel initialization and a return to OS0.

4.4.6 I/O Program Execution

4.4.6.1 Starting I/O Programs

The CPU SIO instruction is used to start all I/O programs; there are three parts to this start-up procedure:

1. Status presentation
2. TCW acceptance
3. TCW processing

4.4.6.2 Status Presentation and I/O Command List Address (IOCLA) Acceptance

Status presentation occurs during the SIO instruction's period of execution. The status response indicates that the I/O program will be started. Any other status response precludes the I/O program start.

The status response indicates that the status queue (STQ) did not contain pending final status for some other subchannel and that the selected subchannel has completed all previously initiated processing functions. This means that CE and DE status is not pending in the channel and that the IOCLA presentation will result in a successful I/O program start.

The IOCLA presentation occurs during the SIO period of execution; its acceptance completes the CPU channel interaction. Upon receiving the IOCLA, the channel notes the change in subchannel state in the appropriate SCR and stores the IOCD address in the associated IOCD register in SST. The IOCD address is not validated during IOCLA presentation. The SCR state indicator is scanned by the channel scheduler so that the act of noting the change of state in the SCR is equivalent to scheduling IOCLA processing; this processing can be delayed for long periods, perhaps several minutes in rare cases, until channel load conditions permit it to be processed. This scheduling of IOCLA processing notes the order of arrival for IOCLAs within a subchannel pair. Within a subchannel pair, the IOCLAs are processed in the order they were received.

The CPU channel interactions that occur during SIO instruction execution may, and often will, interrupt some other channel activity, such as data shovel; the SIO is a real-time processing task, as seen in the channel. In view of this, this task is dealt with in the most expeditious manner possible and the interrupted routine is reentered. The acceptance of IOCLA is given high priority by the channel scheduling mechanism; the processing of IOCLAs is given lower priority. In addition, the channel design, which includes the track format, provides for the acceptance of IOCLAs even during periods of essentially continuous data transfers that occur during multitrack disc read or write operations.

During periods of continuous channel operation, execution of an SIO will follow from the completion of an I/O operation and final status presentation. Normally, this status will be presented via interrupt and interrupt acknowledgement. Assuming that these interrupts are not allowed to stack up in the channel, the interval between SIO instructions will be governed by the natural processing characteristics of the disc system. This system consists of a single data path shared by as many as 16 I/O programs. Sharing of the data path tends to distribute I/O process-completions over time, with the average interval between completions being greater than the minimum dedicated data path time; this time is less than two sector times. A sector time is about 1.0 millisecond.

A data transfer between main memory and disc may have a length that is shorter than a sector time. Although this is the case, the channel must be dedicated to supporting the disc for the entire sector time. This rule applies when writing a sector, because the channel must provide zero bytes to pad out the sector when data is not to be provided from main memory. When reading the disc, the channel must delay to the end of the sector to acquire and validate the ECC. When reading or writing data, the channel must set up for a read or write and, after the operation, shut down the subchannel. Assuming that the read or write operation is confined to a single sector implies a minimum dedicated channel time that is about 1.75 sector times. When labels are to be written, the minimum time increases to 2.75 sector times.

The rate at which sector-sized data transactions may occur is one significant measure of disc system performance. The maximum rate for such transactions is limited by the minimum dedicated data path time, which is 1.75 sector times for this disc system. The inverse of this period is a theoretical maximum transaction rate for the system; this rate is about 550 transactions per second, assuming the F16 track format. Other factors, including the probability of finding a disc that is at the correct angular position and the probability of finding a disc on cylinder and scheduled for I/O, cause the realizable rate to be lower than this maximum. The UDP will support realizable transaction rates in excess of 350 transactions per second when properly configured and supported with software capable of driving it at this rate. Each transaction implies that the CPU must process one interrupt and issue one SIO to the channel. The accesses may be random or ordered accesses into the disc space.

The foregoing description was predicated on the assumption that interrupts would not be allowed to stack up in the channel. If this situation occurs, the processing backlog in the channel will diminish, since no subchannel will accept a new processing task until all status pending in the subchannel is passed to the CPU. Rebuilding the backlog will require the issuance of a number of IOCLAs, perhaps as many as 16, in a short period of time. This load building process can occur at CPU speed when the channel is not supporting data transfers to a disc surface. When the channel is supporting data transfers between main memory and a disc surface, the channel will limit the acceptance rate for IOCLAs to a rate that is related to its success at maintaining the FQ. This, in turn, will be related to main memory performance as seen by the channel. The channel will reject SIOs in these circumstances by issuing channel busy when it cannot accept additional IOCLAs. The normal in-data shovel acceptance rate for IOCLAs will exceed two per sector time, to be determined, even when competition for main memory access is great.

4.4.6.3 IOCLA Processing

IOCLA processing is scheduled by the channel scheduler. This channel processor acquires the IOCLA from the SST and validates the address for being on a word boundary. Should the IOCLA address point to a byte or halfword boundary, the I/O program is terminated with a program check (PCK) and SBR invalid address (INAD) indication. The status subsequently presented to the CPU will contain the invalid IOCLA address in the IOCD address field of status word 1. This error will cause the prior contents of the associated SBR to be lost; the only SST entry information retained from the preceding I/O operations will be the STAR and MODE contents.

4.4.6.4 IOCD Processing and Execution

I/O program execution follows from IOCLA processing. Starting an I/O program is accomplished by building an SST entry in the subchannel image (SI) registers; the following occurs during this IOCD processing:

1. The SI is cleared and the IOCD register is loaded with the IOCD address contained in the IOCLA.

2. The SI STAR and MODE are filled from the appropriate SST, STAR, and MODE registers, respectively.

3. An IOCD is acquired from memory in two successive memory read operations; the IOCD contents are placed in the appropriate SI registers.

4. The command acquired in the IOCD is tested for being a TIC or NOP; if it is either, the I/O program is terminated immediately with a PCK and an SBR command rejected (CR) indication. The first IOCD cannot contain a TIC or NOP.

5. The command acquired in the IOCD is tested for being a sense command; if it is a sense command, the associated SST SBR entry is loaded into the SI. This saves the sense data from the preceding operation. When the first command is not a sense command, the SBR contents are lost.

6. IOCD execution is initiated.

The flag bits 04, 05, 06, and 07 of the IOCD must be zero. Flag bits 00 through 04 have the following significance:

00 DC, data chain

01 CC, command chain

02 SIL, suppress incorrect length indication

03 SKIP, skip read data

04 PCI, program controller interrupt

The command field of the IOCD defines the operation to be performed during command execution. Upon acquiring IOCD word 1, the channel places the command byte in the CR and the absolute data address in the 24-bit DAR. The data address must be a 24-bit address.

Upon acquiring IOCD word 2, the channel places the byte count in the BCR. The five flags are set into the FLR and are used to modify command execution and to control IOCD sequencing, as indicated below.

The requirements for IOCD validity apply to all IOCDs regardless of their intended function, i.e., regardless of the contents of the command field. In some cases, the microprogram logic may not enforce the requirements rigidly. However, the I/O programs should observe these requirements, since future versions of the microprogram or channel may enforce the requirements.

The requirements are

1. IOCD word 2, bits 05 through 15, must be zero.
2. The PCI bit must be zero.
3. The byte count must be nonzero.
4. TIC branch addresses must be 24-bit addresses that address word boundaries.

The acquisition and validation of the first IOCD completes the SI buildup. The SKIP flag causes the suppression of data transfer from disc to memory.

IOCD processing and sequencing consists of IOCD validation, command execution, data chaining, and command chaining. These activities require that the channel access memory for a variety of reasons. Memory read references occur when writing data or when storing the SI in the SST during subchannel deactivation. Any memory reference can produce a nonexistent memory error. In addition, memory read access can produce a memory parity error. All such memory errors cause a channel control check (CCC) and I/O program termination. Modifying SBR indications are not generated.

When the memory access channel condition code (CCC) occurs during a data operation or during IOCD acquisition, the active subchannel is terminated and no other action is taken. This causes a CCC modified CE and DE status response. When the memory access CCC occurs during SST operations, all I/O programs in execution are terminated and the channel enters OS3. No additional memory references occur subsequent to the SST access error; the SST is presumed to be unusable. Entry into OS3 causes the channel to become inoperable until it is reset.

4.4.6.5 Command Execution

Command execution is initiated upon completing IOCLA processing and subsequent to each command chaining point in an I/O program. Data chaining occurs during the execution period of a single channel command and serves only to extend the size of byte counts or to link scattered main memory data buffers that are to be operated upon by a single command. Once command execution is initiated, it continues until all aspects of its execution are complete (a normal end) or until the requirements for its execution come in conflict with channel protocol or the availability of channel resources (an abnormal end). The abnormal end always generates status bits that reflect its cause and that preclude command chaining. Command execution continues in a list of command and data chained IOCDs until equipment-imposed delays prevent further processing.

When a command requires interaction with a device, conditions at the device may cause command processing delays (here, device is used to mean an FHD, an MHD, the FHD portion of an FMHD, or the MHD portion of an FMHD or a cartridge module drive).

Some of these equipment-caused delays will cause subchannel deactivation and others, I/O program termination. The causative factors are grouped in the following manner:

1. Delays that result from the selection of a nonoperational device.
2. Delays that result from an operational device being unavailable to the selecting subchannels.
3. Delays that result from latency at an available device.

A nonoperational device causes I/O program termination with a CE and DE status response modified by a UE status bit and a sense data bit intervention required (IR). This condition will normally require operator intervention; the response is indicative of an attempt to select a nonexistent disc drive or an attempt to select a disc drive that is present but in an open interlock or powered down condition. The open interlock condition may result from the removal of a disc pack at a removable media drive.

When a device is immediately available to the subchannel, it is operational, and equipment-caused delays will be the result of latency at the device. This latency takes two forms: seek time in MHD devices and rotational delays in both FHD and MHD devices. When the command in execution is a seek command that is initiating head movement in an MHD device, the head movement is initiated, the state of operation is noted in the associated subchannel register, and the subchannel is deactivated. The channel scheduler monitors the SCR and will reactivate the subchannel when the disc drive reports seek complete. The MHD is dedicated to the subchannel throughout the device dependent portion of I/O program execution.

Rotational latency may cause subchannel command processing delays when the command in execution requires that data be transferred to or from disc media. In this case, the device will be on cylinder due to a previous cylinder seek operation or because the device is an FHD. The subchannel requests access to the sector that is pointed to be the STAR resident in the SI. The channel compares the STAR SEC value with the priority interrupt SPC contents to determine the time availability of the required disc sector. If the disc angular position is such that a delay of more than one sector time will be required to reach the required disc angular position, the fact that the disc in on-cylinder and in the required angular position is noted in the SCR; the subchannel is then deactivated.

The channel scheduler monitors both the SCR and disc drive angular positions for all SCR and discs and will reactivate the associated subchannel when any disc reaches an angular position that is within one sector time of the required sector time (first opportunity scheduling). This scheduling mechanism permits large transaction rates when several discs are on cylinder and the transactions are small, 1 to 4 sectors. Note that FHDs are always on cylinder and that the use of parallel seeks in MHD systems can keep several MHDs on cylinder. When successful, this scheduling mechanism interlaces several data transfers in a single disc revolution (sector interlacing).

In some cases, subchannel processing delays will result from conflicts for access to a device. Such conflicts can result from two-channel conflicts for dual-ported disc drives; refer to the section titled "Dual-Port Operation" for a description of the dual-port reserve and release functions. All commands that interact with a device will automatically reserve a disc during device selection (auto reserve). A reserve operation can be called for explicitly by executing a reserve command (RES). The conflict for access to a drive can also result when both subchannels of a subchannel pair require access to a device concurrently; sector interlacing may resolve such conflicts when one or both of the subchannels is accessing FHD storage. When both subchannels require access to an MHD, priority scheduling is used to resolve the conflict.

An operational device will always be in one of seven functional states, as seen by a given subchannel attempting to access the device. These states are

1. FS0: unavailable and uncommitted; reserved to the opposite channel and not committed to a competing subchannel.

2. FS1: unavailable and committed; reserved to the opposite channel and committed to the competing subchannel.

3. FS2: unavailable and acquired; reserved to the opposite channel and committed to the requesting subchannel.

4. FS3: idle; not in use and not needed by any channel or subchannel.

5. FS4: reserved and idle; unavailable to the opposite channel, available to this channel and not committed to either competing subchannel in this channel.

6. FS5: reserved and committed; unavailable to the opposite channel and committed to the competing subchannel.

7. FS6: reserved and acquired; reserved to this channel and possessed by the requesting subchannel.

FS0, FS1, and FS2 apply only to dual-ported disc drives, while FS3 through FS6 apply to both single- and dual-ported discs. FS5 and FS6 apply to MHD devices only. This is the case because FHD is always committed to a subchannel only while the subchannel is active, i.e., an FHD is always in one of the states FS0, FS1, FS2, FS3, and FS4 while both of the associated subchannels are inactive.

Again, the FSi applies to a device that may be the FHD or MHD portion of a FMHD. When a channel reserves a device, it automatically reserves the entire device. Consequently, some state transitions are linked and occur automatically.

An initial attempt to access a device will occur when an I/O program passes from a non-device-dependent execution phase to a phase that requires the use of a device. During this initial access phase, the requirements of the subchannel are established in terms of command requirements, device state (FSi), and device type (FHD or MHD); the events that follow are dependent on these factors.

The reserve command (RES) operation is different from all other commands that access a device. When an RES command accesses a device and the device is in one of the states FS0 or FS1, the I/O program is terminated with a CE status presentation. The RES command is stacked in the subchannel SCR and the device is placed in FS2 as seen by this subchannel. The subchannel is then deactivated. The channel scheduler periodically attempts to reserve the disc drive; when it is successful, the associated SI is activated, the device is placed in FS4, a DE status presentation is prepared, and status presentation is initiated. The subchannel is then deactivated and no further subchannel reactivation is scheduled or required for this subchannel. The device is then available to either subchannel in a subchannel pair for at least 500 milliseconds. The HIO CPU instruction terminates a stacked RES command and restores the device to FS0 or FS1.

Commands other than RES do not cause I/O program termination when the selected device is in state FS0 or FS1. When the device is in FS0, the channel places it in FS2 for the requesting subchannel and deactivates the subchannel. The scheduler subsequently attempts to reserve the disc on a periodic basis. When the disc is acquired, the sought-after device is placed in FS6 for the requesting subchannel and subchannel activation occurs. If the acquired disc is a two-device disc drive, the unrequested portion of the disc is placed in FS4.

When the sought-after device is in FS1, the channel stacks a request in the subchannel's SCR and the subchannel becomes a stacked subchannel. A stacked request requires no further scheduler action until the subchannel that possesses the device acquires it, uses it, and releases it. When this release occurs, the device is placed in FS6 for the stacked subchannel; the stacked subchannel is activated automatically and immediately when the

subchannel that originally possessed the device is deactivated. Because of this coupling between the subchannel in a subchannel pair, a device can be kept in an essentially continuous state of operation. The device stays in operation in association with one subchannel of a subchannel pair, while status is presented on the opposite subchannel of a subchannel pair (interrupt overlap).

The state transitions from FS3 to FS4, FS5, or FS6 occurs as follows. FS3 to FS4 is accomplished exclusively by an RES command. The transition from FS3 to FS5 or FS6 may be accomplished by any command that both selects and uses the device. However, an active suchannel will leave a device in FS5 as seen by its competing subchannel when the device is an MHD; this preserves the cylinder position for the controlling subchannel. As was the case above, when a subchannel finds a device in FS5, it becomes a stacked subchannel and the scheduler assigns the device to the stacked subchannel when it becomes available.

The transition from FS4, FS5, and FS6 to FS3 occurs automatically when the device is a single-ported drive and neither subchannel of a pair requires the disc. When the disc is dual-ported, these transitions occur only when a release command is executed or when an unscheduled release occurs. Subsequent transition to FS0 occurs when the opposite channel reserves the disc. This transition is not noted in this channel until a subchannel attempts to acquire the device.

Unscheduled releases are caused by release timer timeouts or by a priority override request being issued by the competing channel. The timer is drive-resident logic that causes the device to release after 500 milliseconds of nonuse. The activation of the timer is optional. When an unscheduled release occurs, it is indicative of faulty equipment operation or faulty operational procedure. The UDP concept assumes that in normal operation all releases of dual-port equipment will be scheduled and will be caused by the execution of the release (REL) command.

4.4.6.6 Data Chaining and Data Management

Data chaining is called for when the data chain (DC) flag is set in an IOCD. During a data transfer, data chaining is executed when the byte count contained in the BCR is reduced to zero and the additional data is available for transfer. Data chaining requires that the following occur:

1. An IOCD is acquired from memory and the PC is incremented by 8.
2. Data address in IOCD word 1 is placed in the DAR.
3. The byte count contained in word 2 of the IOCD is placed in the BCR.
4. The remaining IOCD flags are loaded into the FLR.

Data transmission continues normally.

IOCDs containing TIC commands can be used to link data-chained IOCDs; the TIC IOCD execution is transparent, except for its effect on chaining time, i.e., it locates the next IOCD without affecting the mode of operation in the channel. The TIC IOCD flags are ignored.

Data chaining can be used in conjunction with all read and write data commands. It is not valid when applied to selected read or write commands, the sense command, and most control commands.

See Table 4-20 for an indication of which commands invalidate the use of data chaining. When data chaining is specified in conjunction with a command for which it is invalid, the I/O program is terminated with a final status presentation of CE and DE modified by a PCK status bit and a chaining error (CHER) SBR indication.

Data chaining can be used to perform gather read or scatter write data operations within limitations imposed by channel execution times and channel data buffering times. The channel execution times are affected by main memory access times and, consequently, by competition for access to main memory. In view of this, the limitations on the use of data chaining are total computer system configuration dependent.

The read and write commands that operate on track or sector labels must use addresses that fall on halfword boundaries. Data addresses may be on byte boundaries. When performing multisector data operations, data chaining may be used to move data from or into buffers that begin or end on odd byte boundaries.

It is possible to perform several data chaining operations per sector time, provided that restrictions on minimum byte count are observed. These restrictions are described below and are different for read and write commands. However, regardless of the command type, the byte count must have a value that is 30 or larger. Again, chaining is not valid within a track or sector label transmission, and if chaining is specified within these fields, the I/O program is terminated with a PCK-modified CE and DE status response and a chaining error (CHER) SBR indication. Note that data chaining is affected by the speed of the memories and the SAE of the memory modules.

4.4.6.7 Angular Position Targeting (APT)

The execution of an I/O program usually includes the execution of one or more disc read or write commands. These commands use APT as the basis for subchannel activation and deactivation. All read or write commands begin execution by initiating a seek to cylinder using the cylinder address presented in the STAR. In most cases, this will result in an immediate response of on cylinder, since a seek command will have positioned the heads over the selected cylinder. However, because the disc drives are dual-ported, it is possible that the heads have been moved since the STAR in the subchannel in question was last used; the implicit seek to cylinder provides for this contingency. Head motion in these circumstances is considered to be normal and results in subchannel deactivation and restart once the disc is on cylinder.

When the disc is on cylinder, the read or write command issues a select using the STAR TRK field to select the head. Head switching causes no significant delay. Once head switching is complete, the read or write command reads the disc angular position; the angular position is maintained in one of the eight sector counters. This value is used to establish the time availability of the required disc sector; if the required disc sector is more than one full sector time away, the required angular position is noted in the SCR and the subchannel is deactivated.

The sector target recorded in the SCR is derived from the STAR SEC field. All read write commands target as described in Table 4-20 and Table 4-22. All commands that write labels target on the sector immediately prior to the sector to be written, since the channel always reads a label prior to writing a label. The sector counters are double frequency counters so that the target count is computed as (2x-1) mod(2n) where n equals the number of sectors in the track, (-1) mod(2n) equals 2n-1, (-3) mod(2n) equals 2n-3, and X is the target. All read commands and the write commands set X=SECi. All commands that write labels set X=(SEC-1) mode(n) where (-1) mod(n) equals n-1.

Table 4-20
Channel Commands

Hexadecimal Code	Binary Code	Description	Mnemonic	Data Chaining
Channel Control				
00	0000 0000	Initiate channel	INCH	N
Sense				
04	0000 0100	Sense	SNS	N
Transfer in Channel				
08	0000 1000	Transfer in channel	TIC	Y
Write Commands				
01	0000 0001	Write data	WD	Y
31	0011 0001	Write sector label	WSL	N
51	0101 0001	Write track label	WTL	N
Read Commands				
02	0000 0010	Read data	RD	Y
32	0011 0010	Read sector label	RSL	N
52	0101 0010	Read track label	RTL	N
A2	1010 0010	Read angular position	RAP	N
Control Commands				
03	0000 0011	No operation	NOP	N
13	0001 0011	Lock protected label	LPL	N
23	0010 0011	Reserve	RES	N
33	0011 0011	Release	RL	N
07	0000 0111	Seek cyl, trk, sec	SCK	N
37	0010 0111	Rezero	XEZ	N
47	0100 0111	Increment head address	IHA	N
0B	0000 1011	Format for no skip	FNSK	N
AB	1010 1011	Test STAR	TESS	N
1F	0001 1111	Load mode register	LMR	N
43	0100 0011	Priority override	POR	N
4F	0100 1111	Set reserve track mode	SRM	N
5F	0101 1111	Reset reserve track mode	XRM	N
FF	1111 1111	Initialize controller	ICH	N

Table 4-21
F16 Targeting

Sector Count	Byte Count	Sector	Address Mark Position	Target Count
0	0	0	177	31
1	630			
2	1260	1	1384	1
3	1890			
4	2520	2	2591	3
5	3150			
6	3780	3	3798	5
7	4410			
8	5040	4	5005	7
9	5670			
10	6300	5	6340	9
11	6930			
12	7560	6	7547	11
13	8190			
14	8820	7	8754	13
15	9450			
16	10080	8	9961	15
17	10710			
18	11340	9	11168	17
19	11970			
20	12600	10	12503	19
21	13230			
22	13860	11	13710	21
23	14490			
24	15120	12	14917	23
25	15750			
26	16380	13	16124	25
27	17010			
28	17640	14	17331	27
29	18270			
30	18900	15	18666	29
31	19530			
32	20160	0	177	31

Once the SCR is loaded with the target sector number, the subchannel is deactivated and the scheduler is reentered. The scheduler sequentially tests each SCR in an effort to identify a need to activate some subchannel. The scheduler has an execution time sufficiently short to test each SCR several times per sector time. In view of this, the deactivated subchannel will normally be activated in the first part of the targeted subsector.

The targeting mechanism will occasionally error when the channel has been processing in a subchannel, deactivating that subchannel and returning to the scheduler. The error will occur if the channel detects a bit late in the targeted count; however, the probability rate is low, and the scheduler has logic to detect and recover from this error with small effect on total system performance.

Table 4-22
F20 Targeting

SP(i)	Byte Count	Sector	AM Position	Target Count
0	0	0	190	39
1	504			
2	1008	1	1154	1
3	1512			
4	2016	2	2118	3
5	2520			
6	3024	3	3082	5
7	3528			
8	4032	4	4046	7
9	4536			
10	5040	5	5138	9
11	5544			
12	6048	6	6102	11
13	6552			
14	7056	7	7066	13
15	7560			
16	8064	8	8030	15
17	8568			
18	9072	9	8994	17
19	9576			
20	10080	10	10086	19
21	10584			
22	11088	11	11050	21
23	11592			
24	12096	12	12014	23
25	12600			
26	13104	13	12978	25
27	13608			
28	14112	14	13942	27
29	14616			
30	15120	15	15034	29
31	15624			
32	16128	16	15998	31
33	16632			
34	17136	17	16962	33
35	17640			
36	18144	18	17926	35
37	18648			
38	19152	19	18890	37
39	19656			
40	20160	0	190	39

4.4.6.8 Disc Write Operation

When writing disc, the subchannel is activated 1-1/2 sector times prior to reaching the sector to be written. Upon activation, the command reads and verifies a label. The verification includes confirmation of cylinder, track, and sector position values. It also establishes that the label flags are in a proper condition to allow the write to continue.

Successful label verification indicates that command execution may proceed. This is accomplished by initiating the DIA's FIFO queue (FQ) prefilling operation. The following description applies to a write data operation.

The FQ prefilling operation will begin after proper label verification has taken place on the previous sector label and the strip label and write data command has been issued to the disc controller by the disc firmware.

The FQ prefilling operation will continue until the FQ is full or until a data chaining point is encountered. If a chaining point is encountered before the FQ is full, the operation is executed and the FQ filling is continued. Once the FQ is full, the channel enters a wait loop to wait for the arrival of the next sector label. When the sector label information becomes available, the contents of the label are verified. If the label is valid, the data write operation is initiated in the DIA logic and FQ prefilling is reinitiated in the channel. The DIA logic manages the transfer of data from the FQ to disc without support from the channel. The channel attempts to maintain the FQ in a full condition. If this requires data chaining, the chaining is executed as a single uninterrupted event that relies on the contents of the FQ to provide sufficient buffer time to complete the data chaining operation.

The FQ prefilling operation will continue to sector end and possibly beyond. Once a sufficient number of bytes has been transferred to the FQ to complete the data sector being written, prefilling will continue across a data chaining point even though the label in the next sector has not been read. If the sector being written is in the last sector in a track, FQ prefilling beyond the end of the sector being written will not occur under any circumstances.

Note that FQ prefilling will not continue across a track boundary. Because of a possibility that the next sequential track may be defective, the prefilling operation will not occur. Chaining is reenabled once the track label in the sequential track is verified and it is certain that head motion is not required. In view of this, data chaining points should be located so that some 32 or more bytes are residual in the FQ at track end.

4.4.6.9 Disc Read Operation

Disc read operations are initiated by activating a subchannel in the half sector time immediately prior to the availability of the read data. No useful subchannel processing can occur until the sector to be read is under the disc head. When reading disc, all information that is to be transferred to memory is buffered in the FIFO queue (FQ). Labels and checksums (ECCs) are stored in the FQ only when they are to be transferred to the main memory array. Both the label and ECC are verified in real time, i.e., as they are acquired from disc.

Read data is transferred from the FQ to memory as the FQ is filled. Data chaining occurs as required, and its execution relies on the FQ to provide the buffer time needed for chaining to occur. The restrictions on chaining are less severe for read operations than for writes because the disc read has occurred prior to the need to chain and because memory write access delays are significantly shorter than are read access delays.

When the disc reaches the end of the data portion of a sector, ECC verification occurs in real time. Normally, at least one and perhaps as many as 50 to 100 data bytes may be residing in the FQ at this point. The larger numbers will apply when CPU demands have interrupted data transmission. After the ECC has been verified, the channel returns to data shovel. The data shovel continues until the FQ is empty or until the next label verification is required.

The microprograms that verify the ECC at sector end also provide for reading the next sector, when appropriate. The preparations include the generation of a sector number compound and the initiation of the next sector read in the disc interface logic. Once these activities are accomplished, the effort to empty the FQ is resumed. It may be that the FQ will not be empty before the next label verification is required. When this occurs, data from the preceding sector is carried over into the next sector while the new label is verified and reading the next data field is initiated. In view of this, data from two consecutive sectors may coexist in the FQ concurrently. Once the label verification is complete, FQ emptying is resumed. When all data from the preceding sector is successfully transferred to main memory, the STAR is updated and data operations for the new sector are initiated.

The carry-over from sector to sector does not occur at track end. If the FQ does not reach an empty state before index time, the track label read will be missed and the I/O program will be terminated with a UE modified CE and DE status response and a lost revolution (LREV) SBR indication. This occurs because the ECC verification routine will not initiate track switching and reading the next sector when the disc is at track end. This occurs as a follow-on to emptying the FQ. If the FQ does not reach an empty condition in time, the track label will be missed. This is detected when some other label is read and fails to validate as a valid track label.

4.4.6.10 Command Chaining

Command chaining is called for when an IOCD command chain (CC) flag is set to the one state. Command chaining is executed when the functions specified by an IOCD are completely executed, no exceptional conditions were detected during IOCD execution, and the FLR CC bit is in the one state. Most command chaining operations occur in the following manner:

1. The SR SM bit is tested for being in the one state. If the SM bit is one, the PC is incremented by eight and the SM bit is reset. Otherwise, the next step is executed.

2. The PC is used to acquire an IOCD; the PC is incremented by eight during this process.

3. The IOCD is processed in the following manner: if the command is a TIC, the TIC logic entered and the TIC is executed. This implies that the SI IOCD register is reloaded and another IOCD is acquired. A TIC does not change the contents of the FLR.

4. The command is placed in the SI registers.

5. The byte count is loaded into the BCR.

6. Command execution is initiated.

The SR status modifier (SM) bit provides the command execution logic with an ability to precipitate a program skip based on tests conducted during command execution. The test STAR (TESS) command makes use of this feature to control command (IOCD) sequencing based on comparisons made between memory data and data in the STAR register. The command precipitates the program skip simply by setting the SM bit in the SR. The command chaining logic accomplishes the skip when chaining is initiated.

Program sequencing may also be altered through the use of the transfer-in-channel (TIC) command. The TIC command simply loads the SI IOCD registers with a new IOCD address, whereupon the next IOCD is acquired from memory and executed. This capability, when coupled with the SR SM bit functions, makes it possible to format an entire disc completely under control of the channel.

As indicated above, command chaining is initiated only when no exceptional conditions arise during command execution. Here command execution is extended to encompass data chaining. All such conditions will be reflected in SR bit settings. In view of this, the command chaining logic determines whether or not chaining is in order by testing the state of SR flags. When the flags preclude chaining, program termination is initiated. The following SR bits may be set during command execution to cause I/O program termination: UC, UE, IL, PCK, and CDC. The conditions causing these flags to be set are described in other sections.

The IL bit causes program termination only when the FLR SIL bit is in the zero state. The SIL bit is set in an IOCD to indicate a mismatch between the byte count and the data length is expected. The SIL flag can be used to enable command chaining when the residual byte count is nonzero; it cannot be used to enable data chaining when the data length is incorrect.

Command chaining is initiated only when all aspects of command execution are complete. Thus, disc write operations terminate in the gap between sectors or in the gap between the track label and the sector 0 label. Disc read operations terminate after the disc head has traversed the last sector to be read; however, emptying the FQ may cause command execution to approach or extend into the next sector time. In view of this, command chaining after a sector read that includes reading the data field precludes operating on the next sector without the loss of a disc revolution. Command chaining to a write operation always causes the loss of at least one sector time except at track end.

The end-of-track timing pads provide sufficient time to execute at least two command chaining operations. Chaining operations are executed in the interval between the end of F16 sector F (hexadecimal) and the beginning of the track label. The end-of-track command chaining operations provide for read-to-write switching, write-to-read switching, and program controlled track switching without the loss of a disc revolution. Program controlled track switching may be accomplished through the use of either the increment head address command or the seek command.

Disc read and write commands contain an implied seek to cylinder, track, and sector. The contents of the STAR point to the sought after sector. When read or write command execution is initiated, the channel reads the appropriate sector counters to determine the angular position of the disc. If the disc angular position matches the requirements for execution, command execution proceeds; the angular positioning requirements are command dependent, as described in 4.4.6.4.

4.4.7 Interrupt Operation

The load-RAM SelBUS operation assigns the channel interrupt level. This operation causes the channel PIL register to be filled with the priority interrupt level to be used during interrupt polling and, subsequently, the suppression of lower priority interrupts when the channel interrupt is active. A load-RAM operation is issued during the execution of the first CPU instruction to the disc controller. A load-RAM operation will be accepted at other times as well; however, its acceptance causes the channel PIL register to be filled and will not change interrupt polling in progress when the load-RAM

operation occurs. The channel will not be responsive to I/O requests or other service requests that require or might require the use of interrupt bus polling until a load-RAM operation has initialized the interrupt logic.

The channel interrupt has four states:

1. Disabled and inactive
2. Disabled and active
3. Enabled and inactive
4. Enabled and active

The disabled and inactive state is the reset state for the interrupt. The HCHNL instruction causes the interrupt to enter this state. The disabled and active state is entered when the CPU and channel execute an ACI instruction while the interrupt is in the disable state. The interrupt is enabled depending upon conditions in the channel when the ECI was received. If the interrupt state was disabled and active, the new state after ECI execution will be enabled and active. Similarly, the enabled and inactive state will result if the interrupt was inactive when the ECI was received.

The enabled and active state is entered in one of two ways. It is entered from the enabled and inactive state during the execution of an ACI instruction. It is also entered when the CPU issues and acknowledges and activates response to a channel interrupt.

The channel prepares status during I/O program termination. When a status doubleword is generated, it is placed in the status queue (STQ). There are two STQ entries for each subchannel and one STQ entry for the CPU. If the STQ is empty when status is queued and if the channel interrupt is enabled, the channel will initiate interrupt polling subsequent to queuing the status. When the status is on queue, the channel will not accept an SIO request to initiate the execution of an I/O program. Instead, the channel uses such attempts as an opportunity to present status.

The status queuing mechanism uses the STQ as a 33-entry ring queue. The queue is implemented with the assumption that the CPU will not modify the contents of the queue without first resetting the channel. Changing the contents of STQ will not cause a breakdown in channel function; however, it might cause the presentation of erroneous status. When status is generated it is placed at the bottom of the queue. When status is collected, it is taken from the top of the queue, a first-in, first-out queue. Since all I/O programs terminate with either one or two status presentations, the 16 subchannels cannot occupy more than 32 STQ entries. In some cases, a subchannel may add DE status to a CE already on queue. In other cases, a subchannel will place two separate entries on queue even though an earlier CE status from the same subchannel is still on queue.

The channel passes status to the CPU by transmitting an address pointer to the CPU. This pointer locates the first word of the status doubleword at the top of the queue. The status doubleword thus indicated becomes allocated to the CPU until the next CPU channel interaction occurs. The channel will not modify a status doubleword that is allocated to the CPU. The period of allocation begins when the address is passed and ends when the CPU responds to another interrupt or when any CPU I/O instruction causes the CPU to issue an ARSTX sequence to the channel.

The channel may place as many as 34 status doublewords on queue if all I/O programs terminate during a span of time when the CPU is not responsive to channel interrupts. Although this condition is improbable, it is possible and is provided for in the channel design. Since the STQ is operated as a FIFO queue, status is always presented in the order of occurrence even though the presentation is delayed for a sustained period.

When the CPU issues a deactivate interrupt while status is on queue, the interrupt is deactivated and polling is reinitiated immediately. In view of this, it is possible that the CPU will be interrupted immediately following the deactivate instruction.

The channel will respond correctly to both the acknowledge with and without activate interrupt acknowledge sequence.

4.4.8 Dual-Ported Operations

The UDP firmware is designed to operate dual-ported I/O such that greater system performance with the least software intervention may be realized. When the disc processor is in operation to a device that is defined as dual-ported, I/O may be initialized in one of two ways. They are as follows:

1. Implied reserve and release. In this mode of operation, there is no software distinction between single- and dual-ported I/O. The UDP knows that the addressed device is dual-ported and will issue the appropriate device reserve command (implied reserve). Once the I/O is complete the UDP knows that the device was reserved by the implied reserve and will automatically issue a device release.

2. Absolute reserve and release. In this mode of operation, the user software will issue a device reserve command, process its I/O (which may be multiple I/O commands), and at the end, the user will issue a device release command.

The UDP has built-in intelligence such that it can operate in the dual-subchannel mode of operation, operate in the implied reserve mode on one subchannel and in the absolute mode on the opposite subchannel, and still maintain disc credibility in terms of not prematurely issuing a device release for one or the other subchannels. This type of disc protection is necessary in a system which supports multiple users and multiple directories on dual-ported disc.

The software implication for this type of functionality is that when the implied reserve technique is used, it is no longer necessary for the software to issue a device release command. This may mean the association of a firmware release that is tied to a software release.

Each dual-ported disc drive is equipped with a release timer. This timer is reset when the drive is selected, and it is held reset as long as the drive is selected on the A bus or A cable. Once the drive is deselected and a no-seek in-progress status is realized, the timer runs. At the end of 500 milliseconds, the drive is released automatically. This should not occur in normal operation. The timer timeout function can be disabled by operating a switch in the drive.

4.4.9 Media Defect - Error Recovery

A significant number of associated disc problems are media problems that can be avoided by properly formatting the disc, which requires interaction between the current operating system (MPX), the diagnostic, and the capabilities designed into the UDP.

The UDP contains a special logic for the detection of errors resulting from the read/write operation. This logic consists of a code generator that produces a 30-bit error correcting code (ECC) that is appended to the end of each sector of data on the disc.

During a write operation, the ECC is generated by the hardware logic within the DIA-1 and written onto the disc at the end of each sector. During a read operation, the read data is passed through the ECC logic again and, in similar fashion, an error code is generated. The generated error code is then compared to the original code written at the end of the sector. If the two codes are equal, there is no error. If the two codes are different, an error has occurred and an error flag is set.

The error detection logic employs a polynomial generator that offers the capability of not only detecting errors in the sector, but also of correcting errors.

4.4.9.1 Definition of Errors

Read errors normally occur if a bit is absent (reduced in amplitude) or shifted significantly from its normal position. The UDP is capable of correcting read errors of this type up to nine bits in length and with not more than one error in a sector.

4.4.9.1.1 Multiple Burst Errors

Multiple errors are possible within a sector. Whenever these errors are separated by more than nine bits, they are no longer correctable. This type of error may occur on a random basis; one time the data may be read without error and the next time, an error will appear. This situation is due to minor media defects and signal degradation.

4.4.9.1.2 False Address Marks

Media flaws, called false address marks, can occur. An address mark is a specially written identifier that has a special purpose and is used by the disc drive itself. Whenever there is a media flaw that looks like a zone of erasure at approximately 2.5 to 3 bytes in length, then the entire track is totally unusable. It would be impossible to consistently read all of the sector labels on a track with false address marks.

4.4.9.1.3 Soft Errors

Soft errors are minor blemishes in the media that, from time to time, cause read errors. This type of error usually disappears if the data is reread. However, retry logic should include a maximum of three retries.

4.4.9.2 Error Recovery

There are three options available as a means of recovering from media flaws or data that cannot be read for various reasons.

4.4.9.2.1 Read ECC

The quickest means of recovering from an ECC error in data is to execute the read ECC command. This command will invoke special firmware that will take the incorrectly generated ECC data and manipulate it such that an exclusive OR mask will be generated with a bit displacement. If the media defect is greater in length than nine bits, status will be generated indicating an uncorrectable ECC error. If channel end (CE) and device

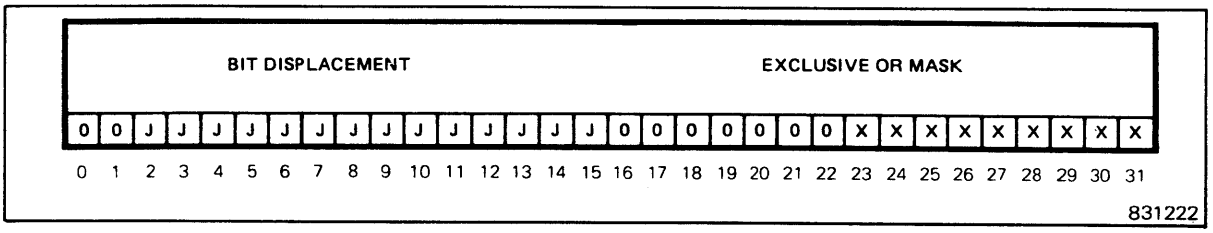
end (DE) status are generated, the error is correctable and the exclusive OR mask must be applied at the proper location, as specified by the bit displacement.

The read ECC command provides a means of recovering from media flaws on a disc pack and random read errors. Since it is quicker to try and correct an error than it is to reread it, error correction should be used first.

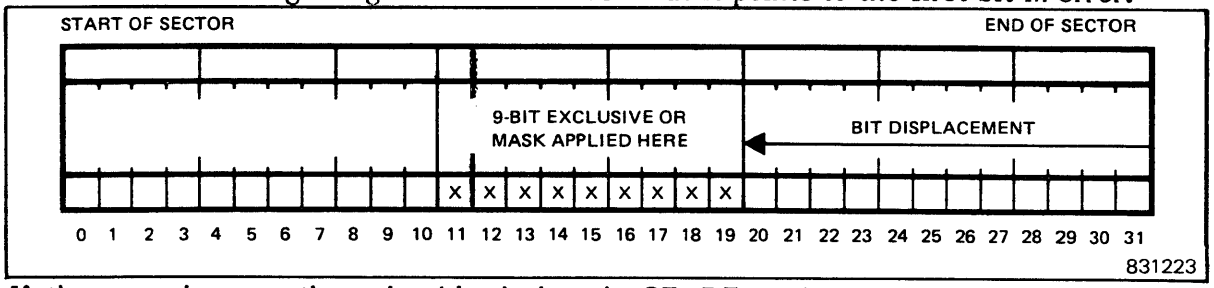
When a read error occurs on the disc and the sense buffer indicates the error is contained in the data field, i.e., not a label, then error correction may be attempted. If a user is attempting a multisector transfer, then he should determine where his buffer started and ended. The user must also take into consideration that he may be working in a mapped environment.

The read ECC IOCD should specify a byte count of four and a byte address that is halfword-bounded. If the error burst is less than or equal to nine bits in length, data will be transferred to memory and channel end (CE) and device end (DE) will be generated.

The data sent to memory are a bit displacement and an exclusive OR mask. It is the user's (or system handler's) responsibility to compute the location in memory where the exclusive OR mask is to be applied. The data format is as follows:



where J is the maximum number of bits for the bit displacement, right-justified and X is the nine-bit exclusive OR bit pattern. The bit displacement is from the end of the data sector toward the beginning of the sector such that it points to the first bit in error.



If the error is more than nine bits in length, CE, DE, and UC status will be generated with a sense buffer indication of an uncorrectable ECC error. There is a remote possibility that if there are more than 30 bits of error, the read ECC command processor will produce a mask and displacement that is incorrect. However, the displacement would fall outside the range of the sector. If the displacement is a negative number, the error occurred in the ECC data itself.

4.4.9.2.2 Reread Data

If an error is uncorrectable, it should be standard practice to try and reread the data at least three times. Error correction may also be tried on each of the three retry reads. It is possible that on a reread operation the data will come in without any errors. In a situation where multiple errors occur in a sector, one of the errors may reread alright and the other can then be corrected.

4.4.9.2.3 Read Offset Modes and Their Uses

Some disc drives are equipped with features that allow I/O programs to adjust their read-strobe time and the radial position of the data heads. This feature is normally available in removable media drives and unavailable in captive media drives; it is provided primarily to compensate for small differences in read timing or head position adjustment in drives that exchange media with other drives.

The normal setting for mode register bits 0 to 3 is zero; they must be zero when a disc write operation is specified. Failure to observe this restriction will cause a program check I/O program termination and a mode check (MOCK) sense buffer indication.

The offset bits provide for a normal setting (all zeros) and eight offset positions. When attempting to recover data from a removable media drive, it is sometimes necessary to use the offset functions to obtain an error free disc read operation. The procedure often used is to attempt to read three or more times at the normal position, then three or more times at both plus and minus track offset positions, then three or more times at both plus and minus read time offset positions, and finally, three or more times at each of the four double offset positions. This process uses the four offset bits to select points in an eight-point rectangle surrounding the normal read position. Since some errors are likely to be soft (not repeatable), three or more tries should be made at each position.

4.4.9.3 Defect Skip

The UDP has the capability of avoiding media flaws, even when they reside within the data portion of a sector. The UPD handles as many as three of these defects in a track. There is no loss of performance or disc storage when this function is effected. Currently, when the track is formatted there are three defect skip zones already imbedded in the track, at uniformly spaced positions on the disc. When it becomes necessary to effect this function these skip zones may be repositioned over the media flow.

Defect skipping is implemented by first formatting the disc such that the address modes are positioned to allow an extra 128 bytes. Data is then recorded in the labels via a write sector label and write track label command, indicating the sector that has the defect and the byte position within the sector. Each label has the capacity to hold three entries of these defective sectors and the byte within the sector. These three entries in the label should be thought of as a ring queue because the UDP only looks at the first entry. It is the responsibility of the program formatting the disc to order this information in the labels.

The UDP generates a skip zone equal to 128 bytes in length. In order to preserve a safe margin for error, the defect that is to be skipped may be in the range of 1 bit to 72 bytes in length.

Example: Byte 200 of sector 5 has a media flaw in it. There may be additional flaws in other sectors on this track.

NNNN	NJJJ	JJJJ	JJJJ
XXXX	XKKK	KKKK	KKKK
YYYY	YLLL	LLLL	LLLL

where NNNN = 5 (sector number)
J...J = 200-64=136 (start of skip zone)

where 200 = location of flaw
64 = back up in front of flaw 64 bytes and start skip of 128 bytes

The order of N, X, Y must be altered in the labels throughout the track such that XXXXX must be the first entry for sector X in sector X label. These three entries would then be altered to accommodate the next media flaw.

4.4.10 Channel Commands

There are seven I/O command classifications. The commands have been assigned standard command byte bit configurations; some provide several modifier bits (M bits) that modify their meaning. The following are the command classifications and their bit patterns:

Channel control	X	X	X	X	0	0	0	0
Sense	M	M	M	M	0	1	0	0
Transfer in channel	X	X	X	X	1	0	0	0
Read backward	M	M	M	M	1	1	0	0
Write	M	M	M	M	M	M	0	1
Read	M	M	M	M	M	M	1	0
Control	M	M	M	M	M	M	1	1

The M bits are used to define controller level command functional variations.

The UDP uses one or more command variations from each of the classifications listed, except read backward; all variations of read backward are invalid. All valid commands are listed in Table 4-20.

4.4.10.1 Initiate Channel (INCH)

The INCH command is valid only when the channel is in operational state 1 (OS1). The command fills the SST address register (SSAR) with a 24-bit address that locates the UDP main memory buffer and initializes the buffer. The INCH command is invalid when the channel is in operational state 2 (OS2).

There are two steps in INCH command execution:

1. SST.ADDR filling
2. SST initialization

The successful completion of these steps causes the channel to enter OS2.

The INCH data address must fall on a word boundary, and the byte count myst be 36 or greater; the byte count provides for transferring nine words of information from memory to the channel. The first word is loaded into the SST.ADDR register. This word is assumed to contain a 24-bit address that points to the first word in the channel's

224-word allocation. The address placed in the SST.ADDR register must fall on an eight word boundary. Failure to observe this restriction causes a program check (PCK) and an I/O program termination. The next eight words are drive configuration data that are loaded into the SST subchannel pairs during SST initialization. Each word is loaded into an odd-even subchannel pair as the SST is initialized. Refer to Figure 3-8 for a detailed illustration of the initialization data.

4.4.10.2 Initialize Controller (ICH)

An alternate method of initializing the channel and controller is provided such that the channel can be initialized as a separate command and the controller can be initialized separately. The alternate channel/controller initialization sequence requires that the initialize channel command IOCD specifies a byte count of 896. The data address provided in the IOCD is placed in the SST.ADDR register. The initialize controller, op code FF (hexadecimal), causes the device donfiguration data to be transferred to the controller portion of the channel.

This alternate initialization sequence provides greater compatibility with disc products that are configurated or an integrated channel controller.

4.4.10.3 Sense (SNS)

The SNS command transfers 12 or 14 bytes of information from the controller to the designated memory locations. The bytes have the following significance:

1. Bytes 0, 1, 2, and 3 are the contents of the sector target register (STAR); bytes 0 and 1 are the CYL field, byte 2 is the TRK field, and byte 3 is the SEC field.

2. Byte 4 is the content of the mode register (MODE).

3. Bytes 5, 6, and 7 are the contents of sense buffer register (SBR), bytes 0, 1, and 2, respectively.

4. Bytes 8, 9, 10, and 11 are the contents of drive attribute register (DATR), bytes 0, 1, 2, and 3, respectively.

5. Bytes 12 and 13 contain drive-related status, as described below.

The IOCD byte count determines the number of bytes of sense data moved to main memory. Executing an SNS command causes the SBR to be cleared. When the command causes the SBR contents to be transferred to memory, the SBR is cleared as the SBR contents are transferred, i.e., before the DATR is transferred.

When an error occurs during SNS execution, the error causes the SBR to be cleared and the sense data associated with the SNS command execution to be placed in the SBR.

Figure 4-12 describes the three SBR bytes. Each bit in the three bytes is assigned a function or designated as a spare. The contents of Figure 4-12 assigns a mnemonic and presents a brief description of the significance of the bit.

Sense bytes 8, 9, 10, and 11 are a simple copy of the contents of the drive attribute register (DATR). Although this information could be acquired directly from the SST by software, it should not be, since future versions of the channel may not use main memory to store the SST.

Sense bytes 12 and 13 are described in Table 4-23. Byte 12 contains two drive status signals and the contents of the sector pulse counter associated with the drive. Byte 13 contains six status bits presented by the drive. These bits include both normal status and error bits.

Sense bytes 0 through 12 may be collected without interacting with the drive. In fact, all 13 bytes may be collected, even though a drive is not attached to the channel. Collecting byte 13 requires that the drive be selected. This selection will cause the drive to be reserved to the channel when it is dual-ported. The channel does not release the drive automatically at the completion of the execution of the SNS command. In view of this, it is the responsibility of the system programmer to affect a release, if this is in order.

The contents of sense byte 13 reflects the operational state of the drive. If the drive is not available, i.e., if it is working with a competing channel, all bits in sense byte 13 will be zero except bit 5, BUSY, which will be a one. This busy condition is accepted as normal and will not cause I/O program termination or suspension of execution in the I/O program. Instead, command execution and command chaining follow from SNS command execution regardless of the response generated by the drive. In view of this, it is possible to command chain a REL command onto the SNS command as a means for automatically releasing the drive.

When the SNS byte count is excessive, IL is set in the status and I/O program termination will occur unless the SIL flag is set in the SNS IOCD.

4.4.10.4 Transfer in Channel (TIC)

Transfer in channel causes the IOCD address of the subchannel IOCD register and FLR register to be replaced with bits 08 to 31 of the TIC IOCD word 1. This address must fall on a word address boundary, i.e., IOCD word 1, bits 30 and 31, must be zero. IOCD word

Table 4-23
Drive Status Bit Assignments

Byte	Bit	Mnemonic	Meaning
12	0	SEND	Seek end
	1	USEL	Unit selected
	2	SPCi 0	Sector pulse counter bit 0
	3	SPCi 1	Sector pulse counter bit 1
	4	SPCi 2	Sector pulse counter bit 2
	5	SPCi 3	Sector pulse counter bit 3
	6	SPCi 4	Sector pulse counter bit 4
	7	SPCi 5	Sector pulse counter bit 5
13	0	FLT	Disc drive detected a fault
	1	SKER	Seek error
	2	ONC	On cylinder
	3	UNR	Unit ready
	4	WRP	Write protected
	5	BUSY	Drive is busy
	6		Spare
	7		Spare

2 is not used during TIC IOCD execution; however, it must conform to conventional IOCD requirements, i.e., a nonzero byte count must be provided and IOCD word 2, bits 08 to 15, must be zero. Although IOCD word 2 is not actually read and verified in the UDP at present, future revisions to the channel microcode may implement IOCD word 2 validation functions.

TIC IOCD execution does not change subchannel flags; the FLR contents are not affected by TIC execution. TIC IOCDs may appear in both data-chained and command-chained IOCD lists.

A TIC command cannot point to another TIC command and a TIC cannot be the first command in an IOCD list. Either occurrence causes I/O program termination with a PCK status bit and a CR SBR indication.

4.4.10.5 Read/Write Commands

Read/write command execution is described in Tables 4-24 and 4-25. Table 4-24 contains four columns and lists all read/write commands. Table 4-25 defines the microcode functions (MF) listed in column 4 of Table 4-24. The material in the tables is expanded upon for selected commands or command groups in the following subsections.

4.4.10.6 Track Read/Write Commands (RSL, RD, RTL, WSL, WD, WTL)

The following commands read information from a formatted disc track: RSL, RD, and RTL. All apply to a single unit of information or multiple units of information (track labels, sector labels, or data fields). All are terminated immediately when an ECC error is detected; the error causes the ECC codes to be stored in the associated SST entry and the I/O program to be terminated with a final CE and DE status presentation modified by

Table 4-24
Read/Write Command Functions

Command	Mnemonic	Op Code	Applicable Microcode Functions
Read sector label	RSL	12	00,03,04,08,11,20,23
Read data	RD	02	00,01,02,03,04,05,08,09*,10,12, 17,20,23,26
Read track label	RTL	32	03,04,08,11,20,24
Read angular position	RAP	A2	20,24,25
Write sector label	WSL	31	00,03,04,06,14,19,21,23
Write data	WD	01	00,01,02,04,05,06,08,09*,16,18, 19,22,23,26
Write track label	WTL	51	03,04,07,15,19,21,24

* Applies when a multitrack operation seeks to the next track

Table 4-25
Microcode Functions (Sheet 1 of 2)

Microcode Function	Description
MF00	Automatic STARi SEC incrementation occurs at sector end when the byte count exceeds the availability of information bytes in label fields, data fields, or a combination of label and data fields. This implies that a multisector operation may extend over an entire track.
MF01	Automatic STARi TRK incrementation occurs at track end when the byte count exceeds the availability of information bytes in label fields, data fields, or a combination of label and data fields. This implies that a multitrack operation may extend over an entire cylinder.
MF02	Automatic seek to head zero of the next cylinder is initiated when the byte operation is continuous across cylinder end. All MF02 seeks include seek target verification. This may be an implicit operation when the command calls for an MF06 or MF07 function.
MF03	Automatic command termination occurs at the end of the track for label and data operations, at the end of the last track label for sector label and flag operation, and at the end of the track label for track label operations.
MF04	When a seek locates a defective track, an automatic seek to the alternate track is initiated. Failure to locate the appropriate track at first try causes a seek error and I/O program termination.
MF05	When sequential track operations cause MF02 to be invoked at the end of an alternate track, an automatic seek to the specified sequential cylinder and track is initiated. This track may be defective, which implies that MF02 will be invoked a second time; however, since the second seek is from a defective track, a seek error will occur unless the required alternate track is located.
MF06	The sector label or track label preceding the target sector label is read and verified before performing the indicated write operation. The write lock flag is invoked when write lock is operative. Error checking is applied to this read label operation.
MF07	The sector label preceding the track label is read and verified before performing the write track label operation. The write lock flag is invoked when write lock is operative. Error checking is applied to this read label operation.
MF08	The label in the target sector is read and verified before or during the indicated data operation. This read includes error checking of the label.
MF09	The track label is read and verified during the indicated data operation. This requires that the track label have proper format and that the label have a zero checksum residual.

Table 4-25
Microcode Functions (Sheet 2 of 2)

Microcode Function	Description
MF10	Error checking is performed at the end of the data field. A nonzero checksum causes termination of command execution.
MF11	The channel transfers the entire label read from disc to the specified destination.
MF12	The channel transfers the data read from disc to the specified destination.
MF13	The channel causes a label to be written in the designated sector. The flag byte 2 is acquired from the specified source. The balance of the information is obtained in the MF06 operation, the propagated information.
MF14	The channel causes a label to be written in the designated sector. All label bytes are acquired from the specified source.
MF15	The channel causes a track label to be written in the designated track. All bytes written are acquired from the specified source.
MF16	The channel causes a data field to be written after the label in target sector. The data is acquired from the designated source.
MF17	The information destination may be either CPU main memory or a RAM buffer, as specified by the subchannel mode register buffer mode bit.
MF18	The data source may be either CPU main memory or a RAM buffer, as specified by the subchannel mode register buffer mode bit.
MF19	Underflow can occur because of memory read access delays or non-existent memory.
MF20	Overflow can occur because of memory write access delays or non-existent memory.
MF21	The write label operation buffers the entire label in the FQ befre the write is initiated. In view of this, it is not possible to write improper labels, and underflow cannot cause the writing of a defective label.
MF22	Underflow can cause a data field to be only partially written, but it cannot cause the data field to have an incorrect checksum.
MF23	The STAR is incremented on a sector by sector basis.
MF24	The STAR is not affected by this command.
MF25	A disc drive is not required by this command.
MF26	Automatic command termination occurs at track end when the track is flagged as a last track.

a UC bit and data check (DCK) SBR indication. The STAR will point to the sector causing the check when the error occurs in a sector. An error in a track label causes the invalid STAR SEC F16 value of 10 (hexadecimal) to be presented in the sense data, i.e., end of a track.

The commands WSL, WD, and WTL transfer information to a disc track. Again, label write operations are initiated only after all required label bytes are buffered in the channel's FIFO queue (FQ); a faulty label write can then occur only when the equipment fails. Track write operations cause label read operations to occur; in view of this, it is possible to incur an ECC read error while writing disc. Such an error is reported as described above for a read label operation.

The command WSL writes sector labels; WTL writes a track label. These commands read the label immediately preceding the label to be written before writing the specified label. The information acquired by this label read operation is used to validate the label information to be written. The information to be written must be consistent with that read except in two cases: the sector label write protect bit can be set to any state desired provided that write lock protection is not violated.

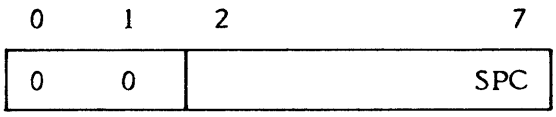
Label write operations are single track operations in that they will not initiate track switching automatically. WTL writes a track label and terminates immediately after the track label ECC is written. In view of this, approximately a full disc revolution must occur before another track label can be written. Multitrack WTL operation can be programmed by command chaining WTL, IHA, and TESS commands to write the labels in a specified set of tracks. When the subchannel executing the I/O program is not competing with other disc operations, the I/O program will not lose disc revolutions. When reading and writing track labels, the STAR will always point to sector zero at the completion of the IOCD.

WSL writes one or more sector labels. Sector labels are written beginning with the sector label specified in the STAR. Label writing is terminated when the byte count is insufficient to permit writing another label or immediately after writing the last label in the track. When the label count is terminated at track end, the termination occurs while the disc heads are approximately at the beginning of the data field in the last sector of the track. In view of this, approximately 1-1/2 sector times must pass before another sector label can be written. Multitrack write sector label opeations can be created by command chaining WTL, IHA, and TESS commands to write the labels in a specified set of tracks. However, all such operations cause the loss of one disc revolution per track written.

The WD command will automatically switch tracks at track end when there is a residual byte count, and a single command when followed by a data chained TIC loop can write the entire disc. Finally, a subchannel pair may be programmed to operate on a single drive in a multi-track operation that occurs without the loss of a disc revolution. This approach provides an interrupt each time an I/O program terminates and the operation switches to the alternate subchannel.

4.4.10.7 Read Angular Position (RAP)

This command reads the sector pulse counter (SPC) from the associated disc port. The type acquired is stored in the main memory location specified in IOCD. The byte has the following format:



Bits 0 and 1 will always be 0. Bits 2 to 7 specify the disc angular position to within a half sector.

4.4.10.8 No Operation (NOP)

This command executes without selecting the associated disc drives.

4.4.10.9 Lock Protect Labels (LPL)

This command invokes write lock when the write lock switches indicate write lock mode 2. If write lock had been invoked earlier, the I/O program is terminated with a PCK status and a WRL SBR indication. The error is also produced when LPL is executed while the write lock mode is 0 to 1.

4.4.10.10 Reserve (RES)

The RES command causes the channel to initiate a device select sequence to the associated disc drive. If the drive responds selected and ready, command execution continues to a normal conclusion. If command chaining is specified, normal conclusion permits immediate command chaining and the execution of subsequent IOCDs in the IOCD list. If command chaining is not specified, normal conclusion implies I/O program termination with an unmodified CE and DE status response.

A dual-ported disc drive will respond busy to an attempt to select it when the drive is reserved to the opposite channel. In this case, I/O program termination occurs immediately with a DB modified CE status response, RES command execution is stacked in the subchannel register, and the subchannel is deactivated. The channel scheduler periodically reinitiates the effort to reserve the disc drive. When this effort is successful, RES command execution is terminated with a DE status response. The effort to reserve the disc drive occurs out of the channel scheduler without subchannel reactivation. When the disc drive responds with any response other than busy, the subchannel is reactivated and RES command execution is completed.

When a disc drive port response indicates that the drive is not present or is nonoperational, UE status will accompany any other outstanding status response. The UE status can accompany the CE and DE status response when a disc drive undergoes a change in operational mode subsequent to stacking the RES command.

The RES command may be issued at any time to reset a disc drive release timer, even though the drive is already reserved by the reserving channel.

When the RES command is executed in one subchannel or a subchannel pair and the disc drive is in use by (seek in progress) or reserved to the opposite subchannel of the pair, the drive is selected to reset the drive release timer, the reserving subchannel's SCR is set to indicate that the subchannel needs the drive, and RES command execution is terminated normally. This permits command chaining and the execution of subsequent IOCDs in the reserving subchannel. This stacking will occur only when the device being sought after is an MHD.

In view of the preceding comments, the RES command is successful only when the drive can be successfully selected during RES command execution. The stacked reserve status is set in the reserving subchannel's SCR only to resolve scheduling ambiguities that might result when both subchannels of a subchannel pair require the use of the same MHD device. The second subchannel bows to the prior rights of the first subchannel to select the device.

4.4.10.11 Release (REL)

The REL command is used to terminate a disc drive's reserved state. Once released, the drive becomes available for use by the opposite channel.

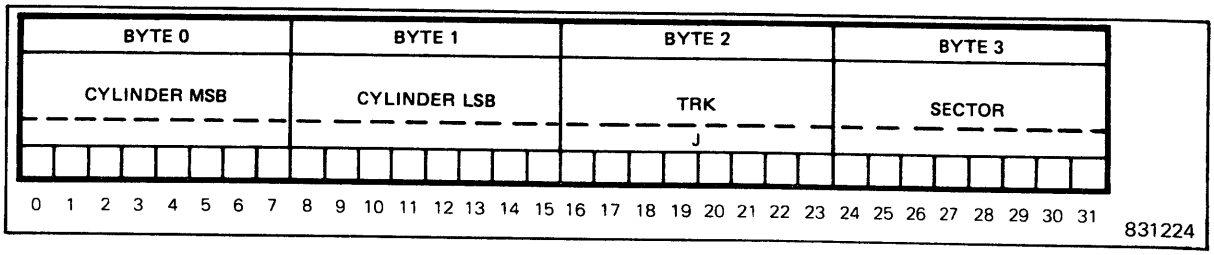
The REL command may not be the final command in an IOCD list; when the command chaining is specified and REL command execution is successful, I/O program execution continues normally.

The REL command terminates execution normally only when it successfully selects an operational disc drive. The drive must respond not busy and selected for REL execution to be successful. Unusual status conditions are reported in a manner similar to that described for RES command execution. However, REL will not stack in a subchannel. Any response other than a normal release will cause the I/O program to be terminated with some form of abnormal status.

4.4.10.12 Seek to Cylinder, Track, and Sector (SKC)

The SKC command loads the STAR in the associated subchannel and initiates a seek to cylinder disc drive when the subchannel is in MHD mode. The seek is not issued to the drive when the subchannel is in FHD mode. Command execution is then terminated in either case.

The SKC command reads four bytes of information from the memory location specified by the IOCD data address. These bytes have the following format:



where J is set by the UDP whenever the mode register bit 6 is set to a 1 and the disc drive is a cartridge module device. When set, the J bit causes the lower volume to be selected. This bit is only maintained in the STAR.

The first two bytes are loaded into the STAR CYL field. Bytes 2 and 3 are used to load the STAR TK and SEC fields, respectively. The SKC command does not validate any information placed in the STAR. However, loading invalid information will cause a disc addressing error (DADE) if the invalid STAR contents are used by a disc read or write operation.

A byte count of four causes all four STAR bytes to be filled from memory. A byte count of one causes all four STAR bytes to be set to 0, a rezero operation.

The byte count accompanying the SKC command should be four or larger. When the byte count is greater than four, command execution occurs normally. However, the command terminates with a nonzero residual byte count and an IL status indication. This precludes data chaining unless the SIL flag is set.

When the byte count is equal to two, only the cylinder address is updated. A byte count of one will affect a device rezero. Any negative byte count will cause program termination with program check. If a byte count of 1, 2, or 3 is used, then SIL must also be set.

The IOCD data address must point to a halfword boundary. Failure to observe this restriction causes a PCK I/O program termination and a CR SBR indication. This error precludes changing any part of the STAR.

When the subchannel is in MHD mode, an effort is made to initiate a seek to cylinder in the associated disc drive. If this effort is frustrated because the MHD is reserved to another channel or to the opposite subchannel of a subchannel pair, the seek command execution is suspended and the subchannel is deactivated until the device becomes available. When the device is available, head motion is initiated and the subchannel is deactivated until seek command execution is terminated by seek completion. Such a termination implies that command chaining is possible, provided that the appropriate flags are set. Again, SIL must be set when the nonzero byte is in some value other than four.

When the subchannel is in the FHD mode, no effort is made to initiate a seek in the associated disc drive. Instead, command execution is terminated normally after the STAR is filled. Again, command chaining is possible, if flags and SKC final status permit. FHD seeks occur during read/write command execution and require only several microseconds to complete, so that it is possible to execute them immediately prior to the beginning of the data transfer.

The attempt to initiate a seek in an MHD drive may select a nonexistent or nonoperational device. In this case, I/O program is terminated with a UE-modified CE and DE status response and an EQCK SBR indication. The error occurs after the STAR is filled and during SKC execution.

The STAR is used to access a disc drive during the execution of any of the track read or write disc commands. Track read/write commands always include a label read operation that serves to verify the proper cylinder and track is being accessed. This verification procedure serves to identify seek errors or other operational faults that could result in read/write errors if they went undetected. Most seek errors will be revealed by this mechanism. Some seek errors will cause a seek out of bounds, i.e., outside the range of valid cylinder addresses, which are reported by drive status signals accompanying the drive seek complete signals.

Regardless of their nature, all seek errors are reported only when a track read/write command attempts to access a disc and finds the disc to be improperly addressed. Seek errors are indicated by a UC-modified CE and DE status presentation accomplished by a DADE SBR indication. These comments apply to both the FHD and MHD discs. Note that a seek error can occur during a read command after several tracks or cylinders have been read successfully, due to faulty head switching or faulty cylinder-to-cylinder head motion.

When an SKC command is presented in an IOCD that does not call for command chaining, the STAR is filled, required MHD head motion is initiated, and the I/O program is

terminated with one of two possible status responses: CE or CE and DE. CE is issued only when MHD heads must be moved, and DE status is presented when head motion completes. When the SKC occurs in FHD mode or when the MHD responds with on-cylinder immediately, CE and DE status is presented.

The UDP formatted capacity and the validity of a cylinder address (CYL value) are drive-dependent.

All disc read and write commands contain an implicit seek each time the command is initiated or reactivated during subchannel reactivation. The implicit seek includes cylinder and track selection. When the subchannel is in MHD mode, the issuance of the cylinder select is normally a precaution that serves to confirm proper disc head position. This precaution is not useful when track read/write commands that perform label verifications are in execution. When the test track or format commands are being used, the implicit seek is needed to reveal unplanned head motion in dual-ported disc drives.

4.4.10.13 Rezero (XEZ)

The rezero command causes the STAR in the associated subchannel to be zero-filled and, subsequently, a disc drive rezero operation to be initiated. This XEZ drive rezero operation is not equivalent to the SKC seek-to-cylinder zero, regardless of the manner in which SKC is used. The XEZ command employs drive interface signals, control signals, that effectively reset the drive's seek logic and cause the drive to locate cylinder and track 0. The XEZ command should be used to recover from a drive error, i.e., one or more XEZ commands should be used subsequent to a seek error and prior to any effort to use any of the command SKC.

An XEZ command also issues a clear fault operation of the drive while initiating the drive rezero operation. The clear fault operation clears any drive-detected fault conditions. In normal operation, the fault will have been reported in some previous final status presentation with the associated sense data. XEZ provides the only means for clearing the fault under program control.

All SKC MHD seeks cause the disc drive to move its head incrementally relative to its current position. When an MHD seek error occurs, the final physical position of the heads does not correspond to the contents of the disc drive's internal cylinder address register. In view of this, the subsequent execution of SKC will continually produce seek error until the disc drive is recalibrated or rezeroed. The XEZ command initiates the recalibration procedure. This procedure causes an MHD drive to locate its read/write heads over disc cylinder zero, to select track zero in that cylinder, and to reset to zero the drive's cylinder and track registers.

XEZ command execution is not terminated until all head motion in an MHD disc drive is successfully completed. FHD operation is similar; however, head motion is not required.

4.4.10.14 Increment Head Address (IHA)

The IHA command modifies the STAR to select sector zero in the next sequential track in the associated disc drive. If head motion is required, it is initiated during IHA command execution and command execution is terminated immediately.

IHA command execution is similar to SKC execution in that it is a seek-to-cylinder; however, the IHA seeks-to-cylinder only when addressing the next sequential track requires head motion. IHA command execution requires that the TK field of the STAR be valid when IHA command execution is initiated. This means that the TK value in the STAR must be less than or equal to the cylinder size attributed to the associated FHD or MHD device. When the STAR TK field is not valid, the I/O program is terminated with a UC-modified CE and DE status response and an INAD SBR indication. The error precludes modifying the STAR.

When the STAR TK field is valid, TK is incremented by one and compared to the cylinder size. If the resulting TK is larger than the cylinder size, SEC is set to zero, TK is set to zero, and the 15-bit CYL field is incremented by one. When the subchannel is in MHD mode and the CYL is incremented, a seek-to-cylinder is issued to the MHD device. When the subchannel is in FHD mode, incrementing the CYL field does not cause a seek-to-cylinder to be issued to the device. In view of this, the IHA command interacts with a device only when MHD head motion is required.

It is possible that an IHA command will select a nonexistent or inoperable device when attempting to initiate disc head motion. When this occurs, the I/O program is terminated with a UE-modified CE and DE status response and an IR SBR indication.

4.4.10.15 Format for No Skip (FNSK)

The FNSK command formats a disc track in two disc revolutions. The track to be formatted is located by issuing an SKC command that fills the STAR. In multiple track-test and format operations, the IHA command may be used to select the next sequential track. FNSK command execution is confined to a single track, the track selected by the STAR, and always requires two revolutions to complete when the format operation is successful.

The first disc revolution is used to write the address marks (AMs) that define the beginning of the information fields in the track. The information fields are the track label and sectors described in 4.4.2.1.3. The second disc revolution is used to fill the information fields with valid label and data information. Here, validity implies that label information is correct and ECC checksums are in place in both label and data fields.

The data address in the FNSK IOCD is used to acquire a string of halfwords that govern FNSK execution and the resulting track format (see Figure 3-9). The IOCD data address must fall on a halfword boundary. Failure to observe this restriction causes I/O program termination. This termination causes a PCK-modified CE and DE status response and CR SBR indication.

The string of halfwords (Figure 3-9) contains four groups of information. The first group contains a single halfword. This halfword contains a right-justified binary count (n) that indicates the number of AMs to be written in the track and the size of the second information group. This count may have any value in the range 001 to 00FF (hexadecimal); however, values inconsistent with planned track formats should not be used. Values larger than 00FF (hexadecimal) cause a PCK, I/O program termination, and a CR SBR indication.

The second information group consists of n count values specifing that byte position of AMs be written into the track. One count value is presented as a right-justified binary number in each halfword. There is one such count value for each AM to be written. The first word count specifies the position of the first AM with respect to the track index

pulse. This AM locates the beginning of the track label. The second count defines the distance in bytes from the beginning of the first AM to the beginning of the second AM. The second count is the track label track space allocation. The third count defines the distance in bytes from the beginning of the second AM to the beginning of the third AM. The third count defines the sector zero track space allocation for each of the succeeding sectors. In all cases, the first count will be exactly consistent with the track formats defined in 4.4.2. Counts 2 through n will normally be consistent with these track formats.

The count values are not checked or validated individually. However, if the sum of the counts exceeds 20,160, the FNSK command execution is terminated prematurely, with a PCK-modified CE and DE status response and a DFER SBR indication. The residual byte count will reflect the number of halfwords processed.

The count n and the n count values are used during the first FNSK disc revolution to control the process of writing AMs in the track. The third and fourth information groups acquired from main memory are used to fill the track during the second FNSK revolution. The third group contains the 30 bytes or 15 halfwords used to initialize the track label (refer to items 7 through 22 in Table 4-15). The first and second halfwords in this third group are not actually used; however, memory space must be allocated for them. The cylinder number, item 7 in Table 4-15, and the track number are filled from the STAR. Item 9 in Table 4-15, the track label identifier, is provided by the channel. All other track label bytes are written as acquired from memory (items 10 through 22 in Table 4-15).

The fourth group of halfwords contains the information used to write the sector labels. Since n-1 sector labels are to be written, the fourth group of FNSK control bytes must contain 30 (n-1) bytes or 15 (n-1) halfwords. Each set of 15 halfwords corresponds to items 5 through 18 in Table 4-16. As was the case for the track label, items 4 and 5 are filled from the STAR by the channel. The sector data fields are zero-filled during this process.

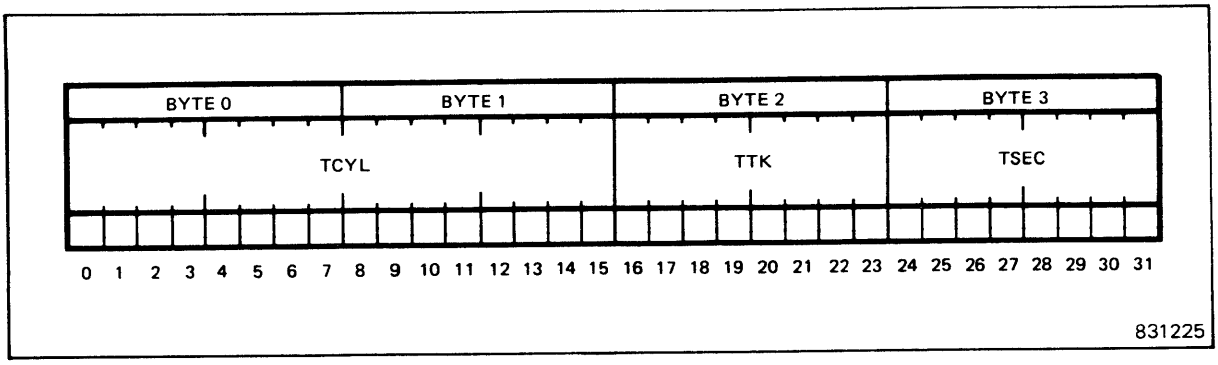
The data field filling includes the generation of valid ECCs. The byte count provided in the FNSK IOCD must be exactly sufficient to write all AMs and all labels in the track. The track data fields are not included in the byte count. The number of sector labels written must correspond to the value n-1 provided in the first halfword read from memory. If the byte-count count-zero condition is not coincident with the last sector label byte written, the I/O program is terminated with a PCK-modified CE and DE status response and a DFER SBR indication. This error indicates that partially formatted track may have been created.

The sector label sector-number, item 7 in Table 4-16, is filled from the STAR. The FNSK command operates on the STAR SEC field during the second FNSK disc revolution. The SEC count is reset at the beginning of the track and incremented as each sector write is completed. Thus, the SEC value that exists at the normal completion of FNSK execution is one greater than the maximum sector number, a 10 (hexadecimal) for the 16-sector track format.

The FNSK command formats exactly one disc track. However, command chaining and IOCD TIC loops can be used to format multiple tracks using the same FNSK control string. The control strings and FNSK execution are track-independent, except where some track requires special action. All cylinder, track, and sector numbers are generated from within the channel; one FNSK control string can serve as a prototype that applies to multiple tracks. The IHA command provides a means for selecting the next sequential track. The TESS provides a means for defining a stop-track.

4.4.10.16 Test STAR (TESS)

The TESS command reads two, three, or four bytes of information from main memory and compares them with the STAR. The bytes read from main memory are assumed to have the following format:



The data and STAR contents are compared, and if the disc address indicated in the data is equal to or greater than the STAR address, the subchannel SM status bit is set. This will cause an I/O program skip when command chaining is specified. The IOCD byte count must be two or greater if command execution is to be successful. A count of one causes a PCK and I/O program termination. A count of two or three truncates the comparison to CYL or CYL and TK, respectively.

The byte count of two comparisons causes the SM bit to be set when TCYL CYL. The byte count of three comparisons causes the SM bit to be set when

$TCYL > CYL$

or when

$TCYL = CYL$ and $TTK \geq TK$

The byte count of four comparisons causes the SM bit to be set when

$TYCL > CYL$

or when

$TCYL = CYL$ and $TTK > TK$

or when

$TCYL = CYL$, $TTK = TK$, and $TSEC \geq SEC$

A byte count greater than four allows normal command execution; however, command chaining will be suppressed by the IL indication unless the SIL flag is set.

4.4.10.17 Load Mode Register (LMR)

The LMR command loads the subchannel MODE register with a mode byte read from main memory location specified by the IOCD data address. A byte count other than one causes an IL indication; SIL is operative.

The LMR can be used to set any MODE register mode bit combination of mode bits 0 to 4 and 6. Mode bit 7, the buffer mode bit, can be set only when the RAM buffer is installed and made operational. The reserved track mode bit, bit 5, can be set only when reserved track mode is switch-enabled. An attempt to set an invalid mode bit causes I/O program termination with a PCK-modified CE and DE status response and a MOCK SBR indication. The MODE register is not changed with the new contents of the MODE register.

4.4.10.18 Priority Override (POR)

The POR command provides a mechanism for overriding and disabling the normal dual-ported disc drive reserve release function. The POR command issues a priority select control sequence to the associated disc drive. This control sequence overrides any normally reserved state existing at the drive and POR reserves the drive to the channel executing the POR command. The priority select sequence will interrupt any normal I/O operation in progress on the opposite channel. Consequently, it is possible to create partially written records when the POR is used.

A drive that is selected using the POR command is absolutely reserved to the channel gaining control over the drive until such time that the channel releases the drive; the drive's internal timer has no effect on the POR reserved state. Once one channel has POR reserved a drive, the drive will not accept a POR reserve from the opposite channel, i.e., a successful POR reserve is absolute.

The POR command can be used in two ways. One application consists of using the POR command to acquire control over a drive that is normally reserved to a faulty channel or system. The POR select prevents the opposite system, which is presumed to be broken, from acquiring the drive under any circumstances.

A second application for the POR command is as a modifier to a normal reserve operation. In this application a POR command is command chained onto a RES command. Should the RES be successful in acquiring the drive, the POR command is executed absolutely reserving the drive. The POR command was used only to ensure that ongoing I/O activities in the opposite channel are not disrupted. The POR command then serves to make the reserved state absolute. This use of the POR command permits creation of master-slave software I/O protocol. In addition, it provides for a software disabling of the drive's release timer.

POR command execution is identical to that of RES command execution, with one exception: the drive is selected using the priority select interface control signal. If the opposite channel has already selected the drive using a POR select sequence, the drive will respond busy and the I/O program will be terminated with a UC, CE, and DE status presentation and a release fault (RELF) sense buffer indication.

4.4.10.19 Set Reserve Track Mode (SRM)

The set reserve track mode (SRM) command causes the UDP firmware to "OR" in the reserve track mode bit in the MODE register. This will only occur when the reserve track mode jumper is set on the DIA-1. Once a subchannel is in the reserve track mode, then all data areas designated as reserve tracks may be read or written. If a user attempts to set a subchannel to the reserve track mode and the jumper is not set on DIA-1, then the command is not executed and program check is set, modified by mode check status response.

4.4.10.20 Reset Reserve Track Mode (XRM)

The reset reserve track mode (XRM) command will reset the reserve track mode bit in the MODE register for the specified subchannel. This command will not alter any other bits that may be set in the mode register. The SRM and XRM commands will not generate an IL indication.

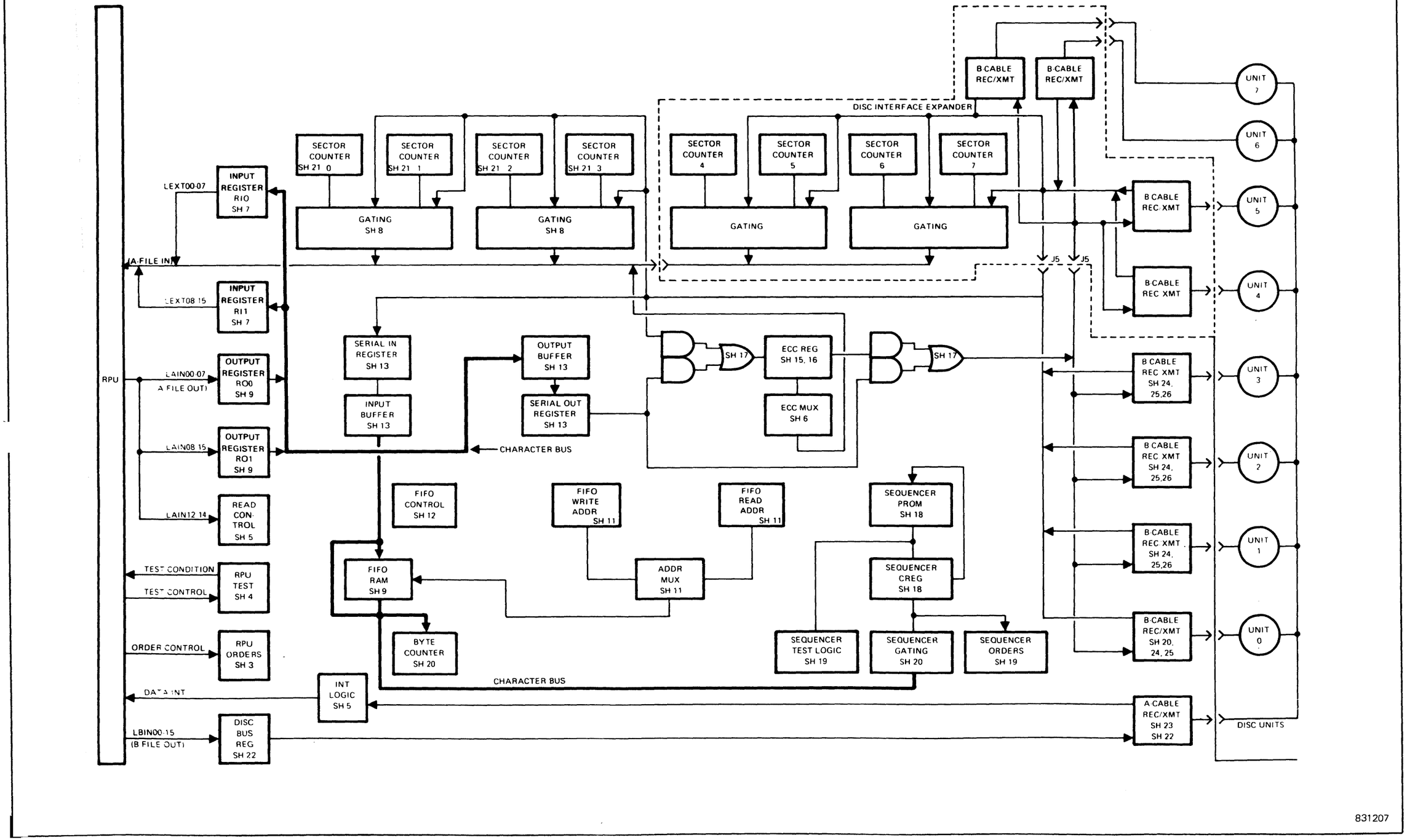


Figure 4-7. Device Interface Adapter-1 and -2 Functional Block Diagram

CHAPTER 5

MAINTENANCE

5.1 Scope

This section provides general maintenance instructions for the universal disc processor. Included are: general maintenance procedures, testing, fault isolation, and corrective maintenance.

5.2 References

The principal reference publication for circuit card maintenance is the Circuit Registration Manual, publication 313-000670. This publication describes specific recommended procedures for maintenance of all components of the system. Refer to the Regional Processor Unit (RPU) User Reference and Design Manual, publication 310-000430, for additional information.

5.3 Maintenance Philosophy

It is recommended that maintenance be confined to the testing, troubleshooting, and replacement of complete circuit card assemblies. Defective circuit cards should be returned to Gould for repair. The repairs will be made and the circuit card will be returned to the user by the most expeditious method. Gould has a complete set of testing and troubleshooting equipment. In addition, manufacturing diagnostic routines are available to check out a circuit card. These diagnostics, supported by specially designed test stands, enable technicians to completely check a given circuit card and demonstrate total reliability.

In the event the user chooses to perform component level maintenance, components can be easily replaced on the circuit card. Replacement parts are available from a wide variety of commercial sources. Many different types of test equipment are also commercially available for testing and troubleshooting. Refer to the circuit registration manual for failure analysis and repair suggestions.

5.4 General Maintenance Procedures

Refer to the Circuit Registration Manual, publication 313-000670, for detailed circuit card maintenance procedures.

5.5 Testing

If a specific malfunction is suspected, an operational test can be devised to exercise the area of suspected malfunction. Using software instructions, a short program that focuses on the area of interest can be prepared. Use of standard instructions gives a technician the ability to initiate specific actions to determine status immediately.

A wide variety of such operational tests can be devised. Each test must be tailored to suit the particular suspected malfunction. Refer to Chapter 3, Programming, and to Chapter 4 Theory of Operation, for detailed descriptions of circuits and routines.

CHAPTER 6

HARDWARE SYSTEM CONFIGURATIONS

When MOS memories are part of the total system configuration, memory interleaving must be implemented. This is due to the large MOS memory modules and normal CPU interference during command and data chaining. For further documentation on memory interleaving, refer to the technical manual for the specific memory board that's being used.

When the UDP is operating in a shared memory environment, the user should restrict the use of shared memory to buffers only. It has been determined that, under certain load conditions, excessive overruns may occur during data chained operation. If data chaining is a requirement in shared memory, the following guidelines should be enforced:

1. All data chained I/O to shared memory should be word-bounded.
2. Shared memory should be used for buffer storage only.

NOTE

If shared memory is used for both executable code and buffer storage, overruns may occur during normal I/O transfers but are recoverable by a retry procedure. However, in this mode of operation data chaining should not be attempted.

Tables 6-1 through 6-3 show how well the UDP performs under the different memory configurations over various MBA links.

In Tables 6-1 through 6-3, the following definitions are applicable:

Yes — indicates that the device will operate error free when transferring into and out of the memory, as configured.

No — indicates that the device will not operate and can lose data when transferring into and out of memory, as configured.

Marginal — indicates that the device will generally run but will experience some number of underflow/overflow, which will be recoverable by software reread or rewrite retries.

Note — All data chaining I/O should be word-bounded.

Table 6-1
MBA-MBA < 30 Feet Memory Type

Discs Over MBA-MBA Into Various Memory	30 Feet	600-ns Core	600-ns MOS Interleaved	900-ns Core	900-ns Interleaved	600-ns MOS Noninterleaved	900-ns MOS Noninterleaved
One disc accessing at a time	Yes Note	Yes Note	No Note	No	No Note	No Note	
Two discs accessing at same time or just one disc and one tape accessing at same time		Marginal	Marginal	No	No	No	No
Other than the two cases shown above		No	No	No	No	No	No

Table 6-2
Memory Configuration

SelBUS Device	Direct MBA	Connected via: MBA to MBC	MBA to MBA 30 Feet	MBA to MBA 80 Feet
Memory Type - 600-ns Core				
XIO	Yes	Yes	Refer to Table 6-1	No
Memory Type - 600-ns MOS (Noninterleaved)				
XIO	No Note	No Note	Refer to Table 6-1	No
Memory Type - 600-ns Semi (Interleaved)				
XIO	Yes	Yes	Refer to Table 6-1	No
Memory Type - 900-ns Core				
XIO	Yes	Yes	Refer to Table 6-1	No
Memory Type - 900-ns MOS (Interleaved)				
XIO	Yes	Yes	No	No
Memory Type - 900-ns MOS (Noninterleaved)				
XIO	No	No	No	No

Table 6-3
Memory Configuration, Low-End Memory, One Board, Noninterleaved

SelBUS Device	Direct SelBUS	MBA to MBC	MBA to MBA 30 Feet	MBA to MBA 80 Feet
XIO	Yes	Not shared/not allocated		

APPENDIX A

GLOSSARY OF UNIVERSAL DISC PROCESSOR FIRMWARE TERMS

Mnemonic	Description	Function
A.B.ERROR	Address boundary error	Label
ABE	Address boundary error	Label
ACTIVE	Causes service interrupt level to be active	Level order 04
ADD.250	Add 250 to final value	Label
ADDRESS.MARK	Address mark	Label
AI	Activate interrupt	Label
AICT	Advance interrupt control transfer	Label
AICT1	Advance interrupt control transfer 1	Label
AMF-F	Address mark flip-flop	Test A34
ANYERROR	Any error	Test B55
ARST2	Advance request for status transfer	Label
ARSTX.OR.AICT	Advance request for status transfer or advance interrupt control transfer	Label
ARSTX1	Advance request for status transfer X1	Label
ARSTX2	Advance request for status transfer X2	Label
ATTENTION	Attention signal to RPU	Test B54
BANK	Selects upper or lower register bank	Level order 74
BANKSELO	Checks for lower bank selected	Test A2D
BAR	Buffer address register	Register
BCR	Byte count register	Register
BUFFER.CMD	Buffer command	Label
BYTE	Byte	Label
BYTE.CNT.OK	Byte count is OK	Label
BYTE.CNT.OK.1	Byte count is OK 1	Label
BYTE.TO.MEM.XFR	Transfer one byte to main memory	Label
BYTE.XFR	Byte transfer	Label
BYTE3	Byte 3	Label
BYTEXFR	Byte transfer	Level order 16
CE	Chaining error	Label
CHAIN.ERROR	Chaining error	Label
CHANNEL.CMD	Channel command	Label
CHK.HW.BYT3	Check halfword for byte 3	Label
CHK.OFFSET	Check offset	Label
CHK.OFFSET.1	Check offset 1	Label
CHK.OFFSET.2	Check offset 2	Label
CHK.OFFSET.3	Check offset 3	Label
CHK.OFFSET.4	Check offset 4	Label

Mnemonic	Description	Function
CL.ER.FLAGS	Clear the error flags	Label
CLR.HALTIO	Clear halt I/O	Pulse order 19
CLRINT	Clear interrupt	Pulse order 49
CNT0IN	SelBUS control 0 tag signal	Test A2B
CNT1IN	SelBUS control 1 tag signal	Test A2C
COMP.AP.1	Compare angular position 1	Label
COMP.AP.2	Compare angular position 2	Label
COMP.AP.3	Compare angular position 3	Label
COMP.AP.4	Compare angular position 4	Label
COMP.AP.5	Compare angular position 5	Label
COMP.AP.TARG	Compare angular position of the target	Label
COMP.DEFECT.ADR	Compare defective address	Label
COMPUTE.SECTOR. XFER.CNT	Compute sector transfer count	Label
CONTROL.CMD	Control command	Label
CONTROLSEL	Control select	Level order 65
CP.DRT	Central processing unit data return transfer	Constant
CPU.WINDOW	Central processing unit window (for interrupt)	Label
CPU.WINDOW.EXIT	Exit from CPU window	Label
CPU.WINDOW.EXIT1	Exit from CPU window	Label
CPU.WND	CPU window	Label
CPUDATAHERE	CPU data present at SelBUS interface	Test
CPUREQUEST	Central processing unit requesting	Test A2E
CR	Command register	Register
CYLINDSEL	Cylinder select	Level order 45
DAR	Data address register	Register
DATAOPER	Data operation	Level order 26
DATR	Drive attribute register	Register
DBUSY-S	Drive busy - synchronized	Test A31
DC.CONT	Data chaining continued	Label
DC.CONT1	Data chaining continued 1	Label
DC.CONT2	Data chaining continued 2	Label
DC.EXIT	Exit from data chaining	Label
DCRMT.BCR	Decrement byte count register	Label
DDX	Disc data transfer	Label
DDX10	Disc data transfer 10	Label
DDX11	Disc data transfer 11	Label
DDX12	Disc data transfer 12	Label
DDX13	Dsic data transfer 13	Label
DDX14	Disc data transfer 14	Label
DDX15	Disc data transfer 15	Label
DDX20	Disc data transfer 20	Label
DDX21	Disc data transfer 21	Label
DDX22	Disc data transfer 22	Label
DDX24	Disc data transfer 24	Label
DDX3	Disc data transfer 3	Label
DDX30	Disc data transfer 30	Label
DDX31	Disc data transfer 31	Label
DDX32	Disc data transfer 32	Label

Mnemonic	Description	Function
DDX4	Disc data transfer 4	Label
DDX40	Disc data transfer 40	Label
DDX41	Disc data transfer 41	Label
DDX6	Disc data transfer 6	Label
DDX9	Disc data transfer 9	Label
DECODE.TBL	Decode table	Constant
DEFECT	Defect	Level order 6
DELAY.30	Delay 30 microseconds	Label
DELAY.30.MICRO1	Delay 30 microseconds	Label
DELAY7	Delay 7 cycles	Label
DEV.ADDR	Device address register	Register
DEV.ERROR	Device error	Label
DEV.ERROR.2	Device error 2	Label
DI	Disable interrupt	Label
DIA.INT	Device interface adapter interrupt	Label
DIACLEAR	Device interface adapter clear	Pulse order 78
DIAGMODE	Diagnostic mode	Level order 15
DISC.FMAT.ERR	Disc format error	Label
DR.ST.1	Drive state 1	Label
DR.ST.2	Drive state 2	Label
DR.ST.3	Drive state 3	Label
DR.ST.4	Drive state 4	Label
DR.ST.5	Drive state 5	Label
DRIVE.LOST	Drive lost	Label
DRIVE.NOT.SELECTED	Drive not selected	Label
DRIVE.OFF.LINE	Drive offline	Label
DRIVE.SEL	Drive select	Label
DSR	Drive state register	Register
DUMMY.READ	Dummy read	Label
DUMMY.READ.01	Dummy read 1	Label
DUMMY.SEC	Dummy sector register	Register
ECC.ERROR.DATA	Error correcting code error data	Label
ECC.ERROR. SEC.LABEL	Error correcting code error sector label	Label
ECC.ERROR. TRK.LABEL	Error correcting code error track label	Label
ECC.READ	Read error correcting code	Label
ECC.REG	Error correcting code register	Register
ECC.RST	Error correcting code reset	Label
ECC2.ERR	Error correcting code error	Label
ECCRESET	Error correcting code reset	Pulse order 68
EETL	ECC error track label	Label
EI	Enable interrupt	Label
EI.1	Enable interrupt 1	Label
EI.2	Enable interrupt 2	Label
ENAMF-1	Enable address mark found	Level order 36
ENATTN-I	Enable attention	Level order 46
ENDFF	Enable end flip-flop	Test A35
ENMICROINT	Enable microinterrupt	Level order
ER.X1	Error condition 1	Label
ER.X2	Error condition 2	Label
ER.X3	Error condition 3	Label
ER.X4	Error condition 4	Label

Mnemonic	Description	Function
ER.XX	Error condition X	Label
ERR	Error, unsupported transaction	Label
ERR1	Nonpresent memory error	Label
ERR1X	Error 1X	Label
ERR2X	Error 2X	Label
ERROR	Error processor, invalid op code	Label
ERROR2	Error 2	Label
ERRORIN	Error in	Test 2F
EVEN.SKIP	Even skip read	Label
EVEN.SKIP.RD	Even skip read	Label
EXEC	Executive scheduler	Label
EXEC10	Executive routine 10	Label
EXEC3	Executive routine 3	Label
EXEC4	Executive routine 4	Label
EXEC8	Executive routine 8	Label
EXEC9	Executive routine 9	Label
F.STATUS.01	Final status 1	Label
F.STATUS.02	Final status 2	Label
F.STATUS.03	Final status 3	Label
F.STATUS.04	Final status 4	Label
F.STATUS.05	Final status 5	Label
F.STATUS.06	Final status 6	Label
F.STATUS.09	Final status 9	Label
F.STATUS.10	Final status 10	Label
F.STATUS.11	Final status 11	Label
F.STATUS.12	Final status 12	Label
F.STATUS.25	Final status 25	Label
F.STATUS.30	Final status 30	Label
F20.CHK	Check for F20 format	Label
F20.FORMAT	F20 format processing	Label
F20.FORMAT.01	F20 format processing	Label
FAULT-S	Fault, synchronized	Test A30
FDSK	Format for defect skip	Label
FIFOFULL	FIFO (first-in, first-out) register full	Test B57
FINAL.ST	Final status processor	Label
FIRST.LENGTH	First length	Label
FLAG.PROCESSOR	Flag processor	Label
FLR	Flags register	Register
FMAT.OR.BUFF	Format or buffer	Label
FNSK	Processor format for no skip	Label
FP		Label
FP01		Label
FRMT.TRK	Format track	Label
GATE.RETRY	Gate microretry	Label
GATE.RETRY.01	Gate microretry 1	Label
GATE.RETRY.02	Gate mircobusy 2	Label
GATE.RETRY.03	Gate mircobusy 3	Label
GEN.01	General processor	Label
GEN.PROCESSOR	General processor	Label
GET.01	Get SST words	Label
GET.02	Get SST words	Label
GET.AP.CTR	Get angular position counter value	Label

Mnemonic	Description	Function
GET.DSR	Get drive state register value	Label
GET.IOCD	Get input/output command doubleword	Label
GET.IOCD.1	Get input/output command doubleword 1	Label
GET.IOCD.2	Get input/output command doubleword 2	Label
GET.MAXIMUM.SECTOR	Get maximum sector	Label
GET.MAXIMUM. SECTOR.01	Get maximum sector 01	Label
GET.MEM.ADDR.	Get memory data address	Label
GET.SCR	Get the current subchannel register data	Label
GET.SCR.TBL	Get subchannel vector table	Label
GET.SI	Read designated subchannel data (subchannel image) from subchannel storage stack	Label
GO.CRITICAL	Go critical (lock out the CPU)	Label
GP01	General processor 1	Label
GP02	General processor 2	Label
GP03	General processor 3	Label
GP04	General processor 4	Label
GP05	General processor 5	Label
GP06	General processor 6	Label
HALF.WD.L	Halfwords left	Label
HALFWORD	Halfword	Label
HALT.IO	Halt input/output	Label
HALT.SBCH	Halt subchannel	Label
HALTIO	Halt input/output	Test 4C
HEADSEL	Head select	Level order 55
HIO.01	Halt input/output 1	Label
HIO.02	Halt input/output 2	Label
HIO.STOP	Halt input/output stop	Label
HSDT	High-speed data transfer	Label
HSDT.1	High-speed data transfer 1	Label
HSDT.EXIT	High-speed data transfer exit	Label
ICR	Interrupt control register	Register
ICT	Interrupt control transfer	Label
ICT10	Interrupt control transfer 10	Label
ICT11	Interrupt control transfer 11	Label
ICT12	Interrupt control transfer 12	Label
ICT4	Interrupt control transfer 4	Label
ICT6	Interrupt control transfer 6	Label
ICT7	Interrupt control transfer 7	Label
ICT8	Interrupt control transfer 8	Label
ID.TYP	Identification type	Constant
ID1	Identify input/output protocol 1	Label
ID2	Identify input/output protocol 2	Label
ID3	Identify input/output protocol 3	Label
ID4	Identify input/output protocol 4	Label
ID5	Identify input/output protocol 5	Label
ID6	Identify input/output protocol 6	Label
IDENTIFY.IO.PRO	Identify input/output protocol	Label

Mnemonic	Description	Function
IHA	Increment head address	Label
IHA.01	Increment head address 1	Label
IHA.02	Increment head address 2	Label
IHA.03	Increment head address 3	Label
IHA.04	Increment head address 4	Label
IL	Incorrect length	Label
IN.RET	Interrupt return register	Register
INC.HEAD	Subroutine to increment to the next head	Label
INCH1	Initiate Channel 1	Label
INCOR.LENGTH	Incorrect lenth	Label
INCREMENT HEAD	Increment head	Label
IOCD	Input/output command doubleword register	Register
IOCD.TIC	Input/output command doubleword transfer in channel	Label
IOCR	Input/output control register	Register
IPL	Initial program load	Label
IPL.01	Initial program load 1	Label
IPL.MEM.READ	Initial program load memory read	Constant
IPL.RD	Initial program load read	Label
IPL.RD.01	Initial program load read 1	Label
ISSUE.DIAC	Issue write label DIAC	Label
ISSUE.DIAC.1	Issue DIAC 1	Label
ISSUE.DIAC.2	Issue DIAC 2	Label
JUMP1	Jumper 1, not used	Test B59
JUMP2	Jumper 2, write lock SW1; write lock all drives	Test B5A
JUMP3	Jumper 3, write lock SW2; programmatically write lock drive	Test B5B
JUMP4	Jumper 4, reserve track mode	Test B5C
JUMP5	Jumper 5, not used	Test B5D
JUMP6	Jumper 6, not used	Test B5E
JUMP7	Jumper 7, not used	Test B5F
JUMP0	Jumper 0, not used	Test B58
KMODULUS	Literal value = 81469	Constant
KP0	Literal value = 3796	Constant
KP1	Literal value = 14674	Constant
LABEL.TO.FQ	Read label and write to FIFO register	Label
LBR.INDEX	Library index; literal = SST size and status queue and working memory = X'2EC'	Constant
LESS.THAN.16	Less than 16 bytes	Label
LINK	Link register	Register
LMR	Load mode register	Label
LMR.01	Load mode register 1	Label
LMR.02	Load mode register 2	Label
LOAD.FIFO.VALUES	Load first-in, first-out register values	Label
LOAD.RAM	Load FIFO RAM	Label
LOADBUS	Load bus	Pulse order 58
LOADRO	Load output register	Pulse order 8

Mnemonic	Description	Function
LPL	Lock protected label	Label
LPL.WCS	Lock protected label - writable control store (PROM card)	Label
LSBA	Load subchannel buffer address	Label
LTF.1	Label to FIFO queue 1	Label
LTF.2	Label to FIFO queue 2	Label
MEM.READ	Read from memory	Constant
MEM.READ.BYTE	Read byte from memory	Constant
MEM.WRITE	Write in memory	Constant
MEM.WRITE.BYTE	Write byte in memory	Constant
MEMDATAHERE	Memory data here	Test A29
MEMORY.WRITE	Memory write	Label
MICRO.TRANS.01	Microtransmit 01	Label
MICROACK	Clears SelBUS interface busy condition	Pulse order 59
MICROBUSY	Enables generation of channel bus on the SelBUS	Level order 14
MICRODATALD	Loads the A and B file output into SelBUS data bus register	Pulse order 5A
MICRODESTLD	Loads the A and B file output into SelBUS destination bus register	Pulse order 6A
MICROINPUT	SelBUS interface logic has entered output transfer sequence	Test B4B
MICROREADY	Generates ready signal on SelBUS	Pulse order 7A
MICRORETRY	Enables the generation of retry on the SelBUS	Level order 24
MICROTRANS	Initiates SelBUS output transfer sequence	Pulse order 69
MODE	Mode register	Register
MODE.CK	Mode check	Label
MUX.DATA	Enables loading data input from SelBUS interface	-
MUX.DEST	Enables loading address from SelBUS interface	-
MX.DATA	Data word storage register	Register
NEGATIVE	ALU test for negative value	Test
NEGATIVE.1	Negative value subroutine 1	Label
NEGATIVE.2	Negative value subroutine 2	Label
NEGATIVE.3	Negative value subroutine 3	Label
NEXT.SECTOR.CHECK	Next sector check	Label
NO.DIAC.ISSUED	No DIAC issued	Label
NONPRSMEM	Nonpresent memory	Test B49
NPM	Nonpresent memory	Test B49
ONCYLIND	On cylinder	Test A33
OP.CODE.1	Operational code 1	Label
OP.CODE.DECD	Operational code decoder	Label
OS.3	Operational state 3	Label
OUTPUT.250	Output 250 bytes	Label
OUTPUT.FIFO	Output to first-in, first-out register	Label
OUTPUT.VALUE.SVE. AND.ZERO		Label
OVERRUN	FIFO overrun	Label

Mnemonic	Description	Function
OVERRUN1	FIFO overrun 1	Label
PARITY.ERR	Parity error	Label
PC	Program counter	Label
PC	Current instruction address	Variable
PC.01	Program check 1	Label
PGM.CK.01	Program check 1	Label
PGM.CK.02	Program check 2	Label
PIL	Priority interrupt level register	Register
POR	Priority override	Label
POSTERROR	Post the error	Label
PRESENT.XFER.CNT	Present transfer count	Label
PRESENT.XFR.CNT	Present transfer count	Label
R.B.C	Residual byte count	Label
R.S.C	Residual sector count	Label
R0STAT	R0 status, start I/O command accepted	Constant
R1STAT	R1 status, command accepted with echo	Constant
R2STAT	R2 status, channel busy	Constant
R3STAT	R3 status, subchannel busy	Constant
R4STAT	R4 status, unsupported transaction	Constant
R5STAT	R5 status, status stored	Constant
R6STAT	R6 status, channel inoperable or undefined	Constant
RAP.1	Read angular position	Label
RAP.1.WCS	Read angular position (writable control store - PROM card)	Label
RATEERROR-S	Rate error - synchronized	Test B51
RB00-15	Register B bits 00-15	Test B40-B47, A20-A27
RBM	Read buffer memory	Label
RD	Read FIFO	Label
RD.01-RD.40	Read routines	Label
RD.CONTINUE	Continue read	Label
RD.DC.17	Read routine	Label
RDC	Read and compare	Label
RDIN	SelBUS read tag signal	Test A2A
READ.01	Read 01	Label
READ.DATA	Read data from disc	Label
READ.LOOP	Read data loop	Label
READ.MEM	Read memory subroutine	Label
READ.MEM.01	Read memory 1	Label
READ.MEM.02	Read memory 2	Label
READ.MEMORY. HALFWORD.ADDRESS	Read memory halfword address	Label
READ.OR.CONTROL	Read or control commands	Label
READDIA	Read device interface adapter	Pulse order 18
READOPER	Read operation	Level order 35
READY-S	Ready - synchronized	Test B53
REC	Read error code	Label
REL	Release command processing	Label
REL.WCS	Release command processing (writable control store)	Label

Mnemonic	Description	Function
REQINT	Request interrupt	Pulse order
RES	Reserve command processing	Label
RES.WCS	Reserve command processing (writable control store)	Label
RESET.CONT	Reset continued; unsupported transaction return	Label
RETURN	Return from subroutine	Label
RETURN1	Return from secondary subroutine	Label
REZ	Rezero disc	Label
RIFULL	Input register full	Test A36
RLD	Read track label and data command	Label
RLD.OR.RTL	Read track label and data command or read track label	Label
ROEMPTY	Output register empty	Test B56
RPUDEF	Regional processor unit assembler file definitions	File
RSC.GT.BC	Residual sector count greater than byte count	Label
RSC.LE.15	Residual sector count less than or equal to 15	Label
RSC.LT.BC	Residual sector count less than byte count	Label
RSETAM	Reset address mark flip-flop	Pulse order 48
RSETATTN	Reset attention flip-flop	Pulse order 38
RSETEND	Reset end flip-flop	Level order 5
RSL	Read sector label	Label
RSL.01-09	Read sector label 01-09	Label
RSL.TEST.BCR	Read sector label - test byte count register	Label
RSL.WCS	Read sector label (writable control store)	Label
RST.FLAG.TRK	Reset flag track	Label
RTL	Read track label	Label
RTL.01-04	Read track label	Label
RTL.WCS	Reat track label	Label
RTN	Return register	Register
RTN2	Return 2 register	Register
RTT	Read test track	Label
S.DSR.1-16	Set drive state register steps 1-16	Label
SBCH	Current subchannel in execution register	Register
SBM	Set buffer mode	Label
SBR	Sense buffer register	Register
SBXF	Single byte transfer	Label
SCK	Seek cylinder head section	Label
SCR.TBL	Subchannel register vector table	Constant
SCR.X1	Right-justify subchannel	Label
SCR0.1	Subchannel register, subchannel 0 and 1	Register
SCR10.11	Subchannel register, subchannel 10 and 11	Register

Mnemonic	Description	Function
SCR12.13	Subchannel register, subchannel 12 and 13	Register
SCR14.15	Subchannel register, subchannel 14 and 15	Register
SCR2.3	Subchannel register, subchannel 2 and 3	Register
SCR4.5	Subchannel register, subchannel 4 and 5	Register
SCR6.7	Subchannel register, subchannel 6 and 7	Register
SCR8.9	Subchannel register, subchannel 8 and 9	Register
SE	Seek error	Label
SECTOR.N	Sector N	Label
SEEK.COMPARE	Seek compare end routine	Label
SEEK.COMPARE.1	Seek and compare 1	Label
SEEK.COMPARE.2	Seek and compare 2	Label
SEEK.COMPARE.3	Seek and compare 3	Label
SEEK.COMPARE.4	Seek and compare 4	Label
SEEK.CYL	Subroutine to seek cylinder	Label
SEEK.ER	Seek error	Label
SEEK.OR.SENSE	Seek or sense decode routine	Label
SEEKERROR-S	Seek error - synchronized	Test B50
SENSE	Sense command processor	Label
SENSE.2	Sense processor 2	Label
SENSE.3	Sense processor 3	Label
SENSE.4	Sense processor 4	Label
SET.ATT.TCZ	Set attention, transfer count = zero flag	Label
SETUP.STRIP. LABEL.RD	Setup to strip and read the label	Label
SIO.01	Start I/O 1	Label
SIO.02	Start I/O 2	Label
SIO.MEM.READ	Start I/O memory read	Constant
SIO10-21	Start I/O 10-21	Label
SK1-16	Seek cylinder, head, sector 1-16	Label
SKIP.RD	Skip read	Label
SKIP.RD.17	Skip read 17	Label
SKIP.RD.CONTINUE	Skip read continue	Label
SKIP.RD.CONTINUE.1	Skip read continue 1	Lbel
SL.COMPLETE	Select complete	Label
SL.COMPLETE.01	Select complete 1	Label
SL.COMPLETE.02	Select complete 2	Label
SLC01	Select complete 1	Label
SLL	Subroutine to sheet left logical	Label
SPT.COR	Stack pointer and control register	Register
SR	Status register	Register
SR99	Subroutine to clear TIC flag	Label
SRL.02	Strip, read, and verify label 02	Label
SRL.03	Strip, read, and verify label 03	Label
SRL.04	Strip, read, and verify label 04	Label
SRM	Set reserve track mode	Label

Mnemonic	Description	Function
SRVL.01-07	Strip, read, and verify label 01-07	Label
SST.ADDR	Status stack address register	Register
SST.SIZE	Status stack size	Constant
ST01-08	Start 01-08	Label
ST81.2	No status stack address	Label
STAR	Subchannel target register	Register
START	Program start	Label
START.IO	Start input/output	Label
STATUS.ADR.CT	Status address count register	Register
STATUS.QUE	Status queue size	Constant
STORE.2	Store SCR	Label
STORE.DSR	Subroutine to store drive state register	Label
STORE.SCR	This subroutine will update the current SCR state	Label
STORE.SI	Write subchannel image to memory	Label
STORE.TBL	Table to store SCR states	Constant
STORE.TBL1	Table to store SCR states	Constant
STRIP.COMP	Compute sector transfer count	Label
STRIP.RD.VERIFY. LABEL	Strip, read, and verify label subroutine	Label
STRIP.RD.VERIFY. LABEL.01	Strip, read, and verify label 01	Label
SVE.253	Save 253 bytes	Label
SWAPBANK.A	Select bank A register	Pulse order 0A
SWAPBANK.B	Select bank B register	Pulse order 09
SYNC	Synchronizing signal	Test B4D
SYSTEM.RST	System reset interrupt processor	Label
TB.OK	Test flag byte OK	Label
TB1-4	Test flag byte error	Label
TCW.ST.1ST	Transfer control word stored first	Label
TESS	Test star (subchannel status register)	Label
TESS.01-05	Test star (subchannel status register)	Label
TEST.AP	Test angular position	Label
TEST.CYL	Test cylinder	Label
TEST.CYL.2	Test cylinder 2	Label
TEST.CYL.3	Test cylinder 3	Label
TEST.CYL.4	Test cylinder 4	Label
TEST.CYL.5	Test cylinder 5	Label
TEST.DEF.ALT.TRK	Test defective alternate track	Label
TEST.DEV.AVAILABLE	Test for device available	Label
TEST.FLAG.BYTE	Test flag byte	Label
TEST.FOR.DC	Test for data chaining	Label
TEST.FOR.ERROR	Test for strip label errors	Label
TEST.FOR.HIO	Test for halt input/output	Label
TEST.FOR.WRITE. LOCKED	Test for device write locked	Label
TEST.FOR.WRITE. LOCKED.01	Test for device write locked 1	Label
TEST.IO	Test input/output	Label

Mnemonic	Description	Function
TEST.LAST.TRACK	Test for last track	Label
TEST.SECTOR	Test sector	Label
TEST.TRACK	Test track	Label
TEST03	Test to compute current status address	Label
TEST04	Test 4	
TFE.01	Test for target error	Label
TFWL.01	Test for write locked	Label
TRACK.LABEL.FF	Track label flip-flop	Label
TRACK.LABEL.TO.FQ	Track label to FIFO queue	Label
TRACK.LABEL.TO.FQ1	Track label to FIFO queue 1	Label
TRANSFERS.COMP	Transfers complete	Label
TRK.SEL	Subroutine to select the proper track	Label
TRKL.FQ	Track label transfers to FIFO queue	Label
TTT	Test test track	Label
TWO	Read most significant halfword to be transferred from the FIFO	Label
TWPE	Test for write protected error	Label
UNITSEL	Unit select	Level order 75
UPDATE.STAR.1	Update subchannel status register 1	Label
VERIFY	Subroutine to strip and verify label	Label
VERIFY.01	Subroutine to strip and verify label	Label
W.FLAGS	Write flags	Label
WBM	Write buffer memory	Label
WBM.OR.WTT	Write buffer memory of write test track	Label
WD.02-36	Write data processor 02-36	Label
WFD	Write flags and data	Label
WK.MEM	Working memory size	Constant
WLD	Write label and data	Label
WLD.OR.WTL	Write label and data or write test track	Label
WLE	Write lock error	Label
WM.01	Write memory 1	Label
WNSL	Write next sector label	Label
WORD	Read word from FIFO	Label
WORK0	Work 0 register	Register
WORK1	Work 1 register	Register
WORK2	Work 2 register	Register
WORK3	Work 3 register	Register
WORK4	Work 4 register	Register
WORK5	Work 5 register	Register
WORK6	Work 6 register	Register
WORK7	Work 7 register	Register
WPE	Write protect error	Label
WRD	Read most significant halfword of a word to be transferred	Label
WRITE.DATA	Write data command processor	Label

Mnemonic	Description	Function
WRITE.LOCK.ERROR	Attempt to write in a protected area	Label
WRITE.PROTECT.ERROR		Label
WRITE.SEC.LB.DIAC	Subroutine to issue write label DIAC	Label
WRITEOPER	Write operation enable	Level order 25
WRITEPROT	Write protected	Test B52
WRLOOP	Write loop	Label
WSL	Write sector label	Label
WSL.02	Write sector label 02	Label
WSL.04	Write sector label 04	Label
WSL.05	Write sector label 05	Label
WSL.06	Write sector label 06	Label
WSL.07	Write sector label 07	Label
WSL.08	Write sector label 08	Label
WSL.WCS	Write sector label - writable control store	Label
WTL.WCS	Write track label - writable control store	Label
WTT	Write test track	Label
XBM	Reset buffer mode	Label
XEZ.1	Rezero disc	Label
XEZ.03	Rezero disc	Label
XEZ.2	Rezero disc	Label
XFER.CNT	Transfer count	Label
XFER.BYTE.TO.MEM	Transfer byte to memory	Label
XFR.LAST.BYTE	Transfer last byte	Label
XFR.LAST.THREE. WORDS	Transfer last three words	Label
XFER.LAST.WRD.HW. BYTE	Transfer last word/halfword/byte	Label
XRM	Reset reserve track mode	Label
XT.INPUT	Enables loading external input	-
Z132	Increment memory address by one full word	Label
ZERO	ALU test for zero value	Test
ZERO.3	Zero 3	Label
ZERO.FILL.FQ	Fill the FIFO queue with all zeros	Label
ZERO.FILL.FQ1	Fill the FIFO queue with all zeros 1	Label
ZFFQ1	Zero fill the FIFO queue	Label

APPENDIX B

GLOSSARY OF DIA SEQUENCER FIRMWARE TERMS

Mnemonic	Description	Function
B( )	Unconditional branch to address in parentheses	Directive
BC	Identifies byte count register	-
CK14-18	Check for a read or write sector	Label
D0-7	Branch instructions for DIAC decode	Label
DECBC	Decrement byte counter	Pulse order 22
ECCTEST	Go to ECC test	Label
ENECCTIME.R	Disable ECC write	Level order 3L4
ENECCTIME.S	Enable ECC time	Level order 3H4
HAMENABLE	Address mark enable	Level order 3H7
HAMFF	Address mark flip-flop	Test A
HBCBZERO	Byte count big zero (12 bits = 0)	Test 14
HBCSZERO	Byte count small zero (8 least bits=0)	Test 13
HDATAOPER	Data operation flag	Test C
HDECC	Disable read data to ECC	Level order 3H5
HDEFECT	Defect skip flag	Test D
HDIAGMODE	Diagnostic mode flag	Test 9
HENECCWCK	Enable ECC write	Level order 3H6
HIBFULL	Input buffer full	Test 11
HOBEMTY	Output buffer empty	Test 10
HRDOPER	Read operation flag	Test 16
HREADGATE	Read gate true	Level order 3H3
HSETEND	Set end flip-flop	Level order 3H0
HWRCONT	Enables write counter	Level order 3H2
HWRGATE	Write gate flip-flop	Test 12
HWROPER	Write operation flag	Test 17
LAMENABLE	No action	Level order 3L7
LATTENTION	Attention flip-flop	Test 8
LDBCFQ	Load byte count register from FIFO queue	Pulse order 17
LDBCHRO	Load byte count register high from output register	Order
LDBCLRO	Load byte count register least from output register	Pulse order 23
LDECC	No action	Level order 3L5
LDFQIB	Load FIFO queue from the input buffer	Pulse order 14
LDFQRIIB	Load FIFO queue and input register from input buffer	Pulse order 13
LDOBFQ	Load output buffer from FIFO queue	Pulse order 16
LDRIIB	Load RPU input register from input buffer	Pulse order 12
LENDFF	End flip-flop high reset	Test 15
LENECCWCK	Disable ECC write	Level order 3L6
LINDEX	Index pulse from disc	Test B

Mnemonic	Description	Function
LL( )	Literal load the value in parentheses	Directive
LREADGATE	Read gate false	Level order 3L3
LSETEND	No action	Level order 3L0
LWRCONT	Write control - disables write control	Level order 3L2
NOP	No operation	Directive
OB	Identifies output buffer	-
READ.L	Strip and read label	Label
RL1-20	Read label, steps 1-23	Label
RTTRACK	Read test track	Label
SERIAL	Go to serial test	Label
SERROR	Set rate error flip-flop	Pulse order 26
SETATTN	Set attention flip-flop	Pulse order 15
SL1-20	Strip label, steps 1-20	Label
SQDEF	Sequencer file definitions	File
START	Program start	Label
STRIP	Strip label	Label
SWRITE	Set write gate flip-flop	Pulse order 20
SYNSRCH	Set search for sync byte	Pulse order 27
T( )	Test and branch to address in parentheses if false; branch to address +1 if true	Directive
U.ECC1-4	Upper PROM bank, error correcting codes, steps 1-4	Label
U.ECCTST	Upper PROM bank, ECC test	Label
U.R.DATA	Upper PROM bank, read data sector	Label
U.RD1-10	Upper PROM bank, read data sector, steps 1-10	Label
U.RDD1-12	Upper PROM bank, read defect data sector, steps 1-12	Label
U.RSEC	Upper PROM bank, read normal sector	Label
URSEC1-7	Upper PROM bank, read normal sector, steps 1-7	Label
U.RTT	Upper PROM bank, read test track	Label
U.RTT1-18	Upper PROM bank, read test track, steps 1-18	Label
U.SERTST	Upper PROM bank, serial test	Label
U.U.ECC3	Upper PROM bank, error correcting code test 3	Label
U.WAITOB	Upper PROM bank, wait output buffer empty	Label
U.WD	Upper PROM bank, write data	Label
U.WD1-3	Upper PROM bank, write data	Label
U.WDATA	Upper PROM bank, write data	Label
U.WDEF	Upper PROM bank, write defect skip sector	Label
U.WDEF1-13	Upper PROM bank, write defect skip sector	Label
U.WSEC	Upper PROM bank, write normal sector	Label
U.WSEC1-10	Upper PROM bank, write normal sector, steps 1-10	Label
U.ZCK	Upper PROM bank, test big zero load RI	Label
UNLDIB	Unload input buffer	Pulse order 25
UPPERHALFR	Upper PROM bank of sequencer microcode set	Level order

Mnemonic	Description	Function
UPPERHALFS	Upper PROM bank of sequencer microcode reset	Level order
WAM	Wait one byte time	Label
WAM1-9	Wait for address mark, steps 1-9	Label
WAMA-F	Write address mark, steps A-F	Label
WERROR	Error, no write clocks from disc	Label
WL1-22	Write label, steps 1-22	Label
WLABEL	Write label	Label
WRITEAM	Write address marks	Label
WSTOP	Stop writing set error	Label
WTT	Write test track	Label
WTT1-23	Write test track, steps 1-23	Label
XFR	Transfer FIFO to/from RPU	Pulse order 11
XWRITE	Reset write gate flip-flop	Pulse order 21

APPENDIX C

GLOSSARY OF DIA-1 LOGIC DIAGRAM TERMS

Mnemonic	Logic Drawing 130-103282 Sheet Number	Description
HADDRMRKEN	19	Address mark enable
LAIN00-15	9	A file input bits 0-15 from regional processor
HAMF	24	Address mark found signal from disc
LAMF	24	Address mark found signal from disc
HAMFF	5	Address mark flip-flop
LAMFOUND	24	Address mark found
LANYERROR	5	Any error test (test 55)
HATTENTION	5	Attention flag
LATTENTION	5	Attention flag
HBCBIGZERO	20	Byte count register = 0 flag
HBCSMLZERO	20	Least 8 bits of byte count register = 0 flag
LBIN06-15	22	B file input bits 0-15 from regional processor
LBUFTSTAD5-7	4	Buffered test structure address bits 5-7
HBUSBIT0-1	23	Disc control bus bits 0-1
LBUSBIT0-1	23	Disc control bus bits 0-1
HBUSBIT2-9	22	Disc control bus bits 2-9
LBUSBIT2-9	22	Disc control bus bits 2-9
LBUSREG0-2	22	Disc unit select bits from bus register
HBUSY	24	Disc drive busy signal
LBUSY	24	Disc drive busy signal
LBYTEXFR	3	Byte transfer
LCBUS0-7	9	Character bus bits 0-7
L2CLK4	3	System clock 2
L3CLK3	3	System clock 3
H3CLK4	3	System clock 3 inverted
L3CLK4	3	System clock 3
H3CLK5	3	System clock 3 inverted
L2CLKFREERUN	3	Free running clock
HCONTROLSEL	3	Control select
LCPUDATAHERE	5	CPU data is present at SelBUS interface
HCREG12-15	3	Control register bits 12-15
HCYLINDSEL	3	Cylinder select
HDATAINT	5	Data interrupt signal to regional processor
HDATAOPER	3	Data operation enabled
LDBUSY	24	Disc busy signal
LDBUSY-S	5	Disc busy signal, synchronized
LDECBC	19	Decrement byte count pulse
HDEFECT	3	Defect flag (level order 06)
LDIACLEAR	3	Device interface adapter clear pulse
HDIAGMODE	3	Diagnostic mode
HDIARESET-4	3	Device interface adapter reset
HDISABLEECC	19	Disable error correcting code

Mnemonic	Logic Drawing 130-103282 Sheet Number	Description
LDISCFAULT	24	Disc fault
HECCCLOCLK	17	Error correcting code clock
HECCDATAIN	17	Error correcting code data in
HECCFEEDBACK	15	Error correcting code feedback
LECCRESET	3	Error correcting code reset
HECCTIME	17	Error correcting code time
LECCTIME	17	Error correcting code time
LENABLEIB	14	Enable input buffer
LENABLER00-1	10	Enable output registers R00 and R01
HENAMF-I	3	Enable address mark found (level order 36)
HENATTN-I	3	Enable attention flip-flop flag (level order 46)
HENDFF	5	Sequencer end of operation flip-flop
LENDFF	5	Sequencer end of operation flip-flop
HENECCTIME	19	Enable error correcting code time
HENECCWCLK	19	Enable error correcting code write clock
LENFIFORAM	12	Enable FIFO RAM output
LENLVLORD1-2	3	Enable level orders 1-2
LENPORD	3	Enable pulse orders
HENRDFIFO	12	Enable read FIFO
LENVECTINT	3	Enable vectored interrupts
HENWRFIFO	12	Enable write FIFO
LERRORIN	5	Error in
LEXT00-15	6	External bus to RPU B file input, bits 0-15
HFAULT	24	Fault signal from drive
LFAULT	24	Fault signal from drive
LFAULT-S	5	Disc fault signal - synchronized
HFIFOADDR0-7	11	FIFO RAM address bits 0-7
LFIFOEMPTY	12	FIFO empty
LFIFOFULL	12	FIFO full
LFIFOFULL-16	12	FIFO full minus 16
LHALTIO	3	Halt input/output
HHEADSEL	3	Head select
HIBFULL	14	Input buffer full
LINDEX	25	Index pulse from disc drive head
HINDEX0-3	26	Index pulses from disc drives 0 through 3
HINDEXD0-3		Index pulses from disc drives 0 through 3
LINDEXD0-3		Index pulses from disc drives 0 through 3
LLOADBC-CREG	19	Load byte counter from control register
LLOADBC-FQ	19	Load byte counter from FIFO queue
LLOADBCH-RO	19	Load high range of byte counter from output register
LLOADBCL-RO	19	Load low range of byte counter from output register
LLOADBUS	3	Load data onto command bus to disc
LLOADFQ-IB	19	Load FIFO queue from input buffer
LLOADFQ-R00	10	Load FIFO queue from output register 0
LLOADFQ-R01	10	Load FIFO queue from output register 1
LLOADFQRI-IB	19	Load FIFO queue and input register from input buffer
LLOADIB	14	Load input buffer
HLOADIBEN	14	Load input buffer enable

Mnemonic	Logic Drawing 130-103282 Sheet Number	Description
LLOADOB	14	Load output buffer
LLOADOB-CREG	19	Load output buffer from control register
LLOADOB-FQ	19	Load output buffer from FIFO queue
LLOADRI	10	Load input register
LLOADRI-IB	19	Load input register from input buffer
LLOADRI1	10	Load input register 1
LLOADRI1-FQ	10	Load input register 1 from FIFO queue
LLOADRI0	10	Load input register 0
LLOADRI0-FQ	10	Load input register 0 from FIFO queue
HLOADRO	10	Load output register
LLOADRO	3	Load output register
LLOADSOREG	14	Load serial-out register
LNONPRSEM	5	Nonpresent memory tag
HOBEMTY	14	Output buffer empty
HONCYL	24	On cylinder
LONCYL	24	On cylinder
HONCYLIND	24	On cylinder
HONCYLIND-S	5	On cylinder - synchronized
HOPENCBLE	23	Open cable
LOPENCBLE	23	Open cable
HPLOCLK	25	Phase locked oscillator clock from disc drive
HPLOCLK0-3	25	Phase locked oscillator clock from disc drive 0-3
LPLOCLK0-3	25	Phase locked oscillator clock from disc drive 0-3
HSCREG00-10	18	Sequencer control register bits 00-10
H0SECCNT2-7	21	Sector count, drive 0, bits 2-7
H1SECCNT2-7	21	Sector count, drive 1, bits 2-7
H2SECCNT2-7	21	Sector count, drive 2, bits 2-7
H3SECCNT2-7	21	Sector count, drive 2, bits 2-7
HSECTORD0-3	26	Sector pulses drives 0-3
LSECTORD0-3	26	Sector pulses drives 0-3
LSECTORP0-3	26	Sector pulses drives 0-3
HSEEKEND0-3	26	Seek end 0-3
HSEEKENDD0-3	26	Seek end disc 0-3
LSEEKENDD0-3	26	Seek end disc 0-3
HSEEKERR	24	Seek error
LSEEKERR	24	Seek error
LSEEKERROR	24	Seek error
LSEEKERROR-S	5	Seek error - synchronized
HSERVOCLK	25	Servo clock
HSERVOCLK0-3	25	Servo clock from disc 0-3
LSERVOCLK0-3	25	Servo clock from disc 0-3
LSETATTN	19	Set attention flip-flop
HSETEND	19	Set end flip-flop
LSETERROR	19	Set rate error flip-flop
LSIN0-7	13	Byte output of serial-in register (bits 0-7)
HSPROM03-08	18	Sequencer PROM bits 3-8
LSTEST8-F	19	Sequencer test bits 8-F
LSTEST10-17	19	Sequencer test bits 10-17
LSWRITE	19	Preset write flip-flop

Mnemonic	Logic Drawing 130-103282 Sheet Number	Description
HSYNCDETECT	14	Sync detect pulse
HSYNSRCH	19	Sync search pulse
LRATEERROR	12	Data rate error (FIFO queue overflow or underflow)
LRATEERROR-S	5	Data rate error - synchronized
L1READCLK1	13	Read clock (from servo clock)
H2READCLK1	13	Read clock (from servo clock)
LREADCNT01	5	Read sector count for drives 0 and 1
LREADCNT23	5	Read sector count for drives 2 and 3
LREADCNT45	5	Read sector count for drives 4 and 5
LREADCNT67	5	Read sector count for drives 6 and 7
HREADDATA	25	Read data (serial bit stream from disc read)
LREADDATA	13	Read data (serial bit stream from disc read)
HREADDATA0-3	25	Read data inputs from disc drives 0-3
LREADDATA0-3	25	Read data inputs from disc drives 0-3
LREADDIA	3	Read device interface adapter; enable for read select
LREADECC	5	Read error correcting code
HREADGATE	19	Read gate signal to disc drive
HREADOPER	3	Read-operation order from regional processor
HREADOPER-S	5	Read operation test - synchronized
HREADRI	7	Read input register
LREADRI	5	Read input register
HREADY-S	5	Disc drive ready - synchronized
HRESET2	23	Open cable reset signal to regional processor
LROEMPTY	10	Output register empty
LRIFULL	10	Input register full
LRSETATTN	3	Reset attention flip-flop
HRSETEND	3	Reset end flip-flop
HTAG1-3	23	Disc tag bits 1-3
LTAG1-3	23	Disc tag bits 1-3
HTESTA2	4	Test signal to test group A of RPU
LTESTA2	4	Test signal to test group A of RPU
HTESTB2	4	Test signal to test group B of RPU
LTESTB2	4	Test signal to test group B of RPU
HTESTB3	4	Test signal to test group B of RPU
LTESTB3	4	Test signal to test group B of RPU
HUNITRDY	24	Unit ready
LUNITRDY	24	Unit ready
HUNITREADY	24	Unit ready
HUNITSEL	3	Unit select
HUNITSEL0-3	23	Unit select, drives 0-3
LUNITSEL0-3	23	Unit select, drives 0-3
LUNITSELD0-3	26	Unit select disc 0-3
HUNITSELDD0-3	26	Unit select disc 0-3
LUNITSELDD0-3	26	Unit select disc 0-3
LUNLOADIB	19	Unload input buffer
HUPPERHALF	19	Select upper half of microsequencer PROM
HUSTAG	23	Unit select tag
LUSTAG	23	Unit select tag
HWRITECLK0-3	24	Write clock for drives 0-3

Mnemonic	Logic Drawing 130-103282 Sheet Number	Description
LWRITECLK0-3	24	Write clock for drives 0-3
L1WRITECLK1	13	Write clock
H2WRITECLK1	13	Write clock
HWRITECONTROL	19	Write control order
HWRITEDATA	13	Write data (serial bit stream to disc)
LWRITEDATA	13	Write data (serial bit stream to disc)
HWRITEDATA0-3	24	Write data for drives 0-3
LWRITEDATA0-3	24	Write data for drives 0-3
HWRITEDATAOUT	24	Write data signal to DIA-2
HWRITEDATOUT	17	Output from write data flip-flop
LWRITEFIFO	12	Low write, high read signal to FIFO RAM
HWRITEGATEFF	14	Write gate flip-flop
LWRITEGATEFF	14	Write gate flip-flop
HWRITEOPER	3	Write operation
HWRITEPROT	24	File is write-protected
HWRITEPROT-S	5	File is write-protected - synchronized
HWRTPRO	24	File is write-protected
LWRTPROT	24	File is write-protected
LXFR	19	Microsequencer transfer order (pulse order 11)
LXWRITE	19	Clear write flip-flop
HX00-15	15	Error correction code register bits 00-15
HX16-29	16	Error correction code register bits 16-29
HX29DLY2	16	Error correction code register bit 29 two clocks delayed

APPENDIX D

GLOSSARY OF DIA-2 LOGIC DIAGRAM TERMS

Mnemonic	Logic Drawing 130-103291 Sheet Number	Description
LEXT00-15	2	External bus to RPU B file input, bits 0-15
LINDEX	5	Index pulse from disc drive head
HINDEX4-7	5	Index pulses from disc drives 4-7
HINDEXD4-7	5	Index pulses from disc drives 4-7
LINDEXD4-7	5	Index pulses from disc drives 4-7
HPLOCLK	5	Phase locked oscillator clock from disc drive
HPLOCLK4-7	5	Phase locked oscillator clock from disc drive 4-7
LPLOCLK4-7	5	Phase locked oscillator clock from disc drive 4-7
LREADCNT45	1	Read sector count for drives 4 and 5
LREADCNT67	1	Read sector count for drives 6 and 7
HREADDATA	5	Read data (serial bit stream from disc read)
HREADDATA4-7	5	Read data inputs from disc drives 4-7
LREADDATA4-7	5	Read data inputs from disc drives 4-7
H4SECCNT2-7	3	Sector count, drive 4, bits 2-7
H5SECCNT2-7	3	Sector count, drive 5, bits 2-7
H6SECCNT2-7	3	Sector count, drive 6, bits 2-7
H7SECCNT2-7	3	Sector count, drive 7, bits 2-7
HSECTORD4-7	6	Sector pulses drives 4-7
LSECTORD4-7	6	Sector pulses drives 4-7
LSECTORP4-7	6	Sector pulses drives 4-7
HSEEKEND4-7	6	Seek end 4-7
HSEEKENDD4-7	6	Seek end disc 4-7
LSEEKENDD4-7	6	Seek end disc 4-7
HSERVOCLK	5	Servo clock
HSERVOCLK4-7	5	Servo clock from disc 4-7
LSERVOCLK4-7	5	Servo clock from disc 4-7
HUNITSELD4-7	6	Unit select disc 4-7
HUNITSELDD4-7	6	Unit select disc 4-7
LUNITSELDD4-7	6	Unit select disc 4-7
HWRITECLK4-7	4	Write clock for drives 4-7
LWRITECLK4-7	4	Write clock for drives 4-7
H2WRITECLK4-7	4	Write clock
HWRITEDATA	4	Write data (serial bit stream to disc)
HWRITEDATA4-7	4	Write data for drives 4-7
LWRITEDATA4-7	4	Write data for drives 4-7

APPENDIX E

MICROSEQUENCER FIRMWARE FLOW DIAGRAMS

This appendix provides generalized flow diagrams for the microsequencer used in the DIA-1. These diagrams, when used in conjunction with the firmware listing, will aid in understanding the sequencer operation. Figure E-1 identifies the conventions used in the flow diagrams.

The flow diagrams show the generalized program flow. For purposes of explanation, they have been simplified by collecting many individual steps into a functional statement. To determine the exact program steps, refer to the program listing. The flow diagrams can be cross-referenced to the program listing by using the subroutine labels shown on the flow diagrams. The program listing contains an alphabetical listing of subroutine labels, which is located at the rear of the program listing.

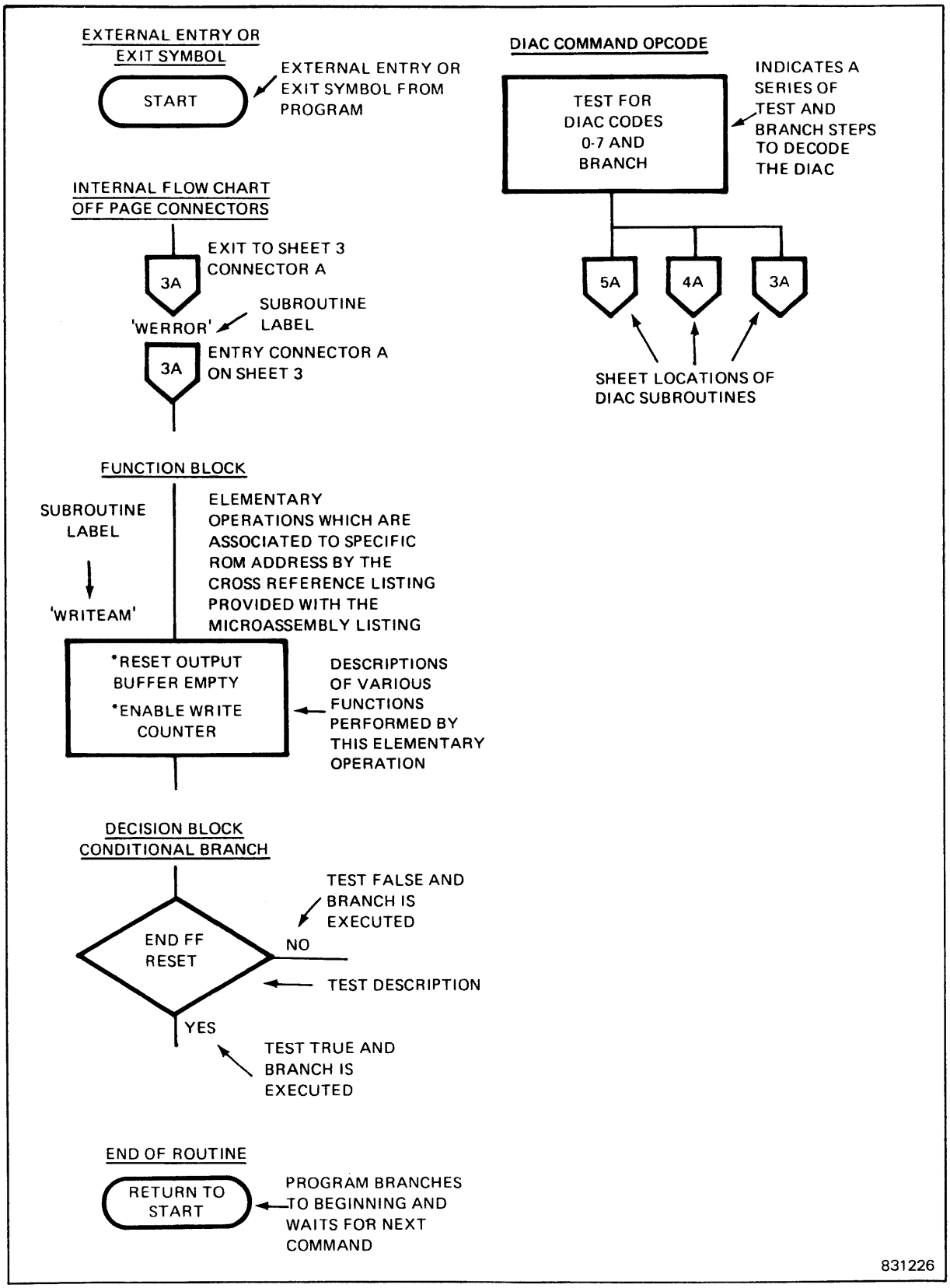


Figure E-1. Flow Chart - Symbol Identification and Convention

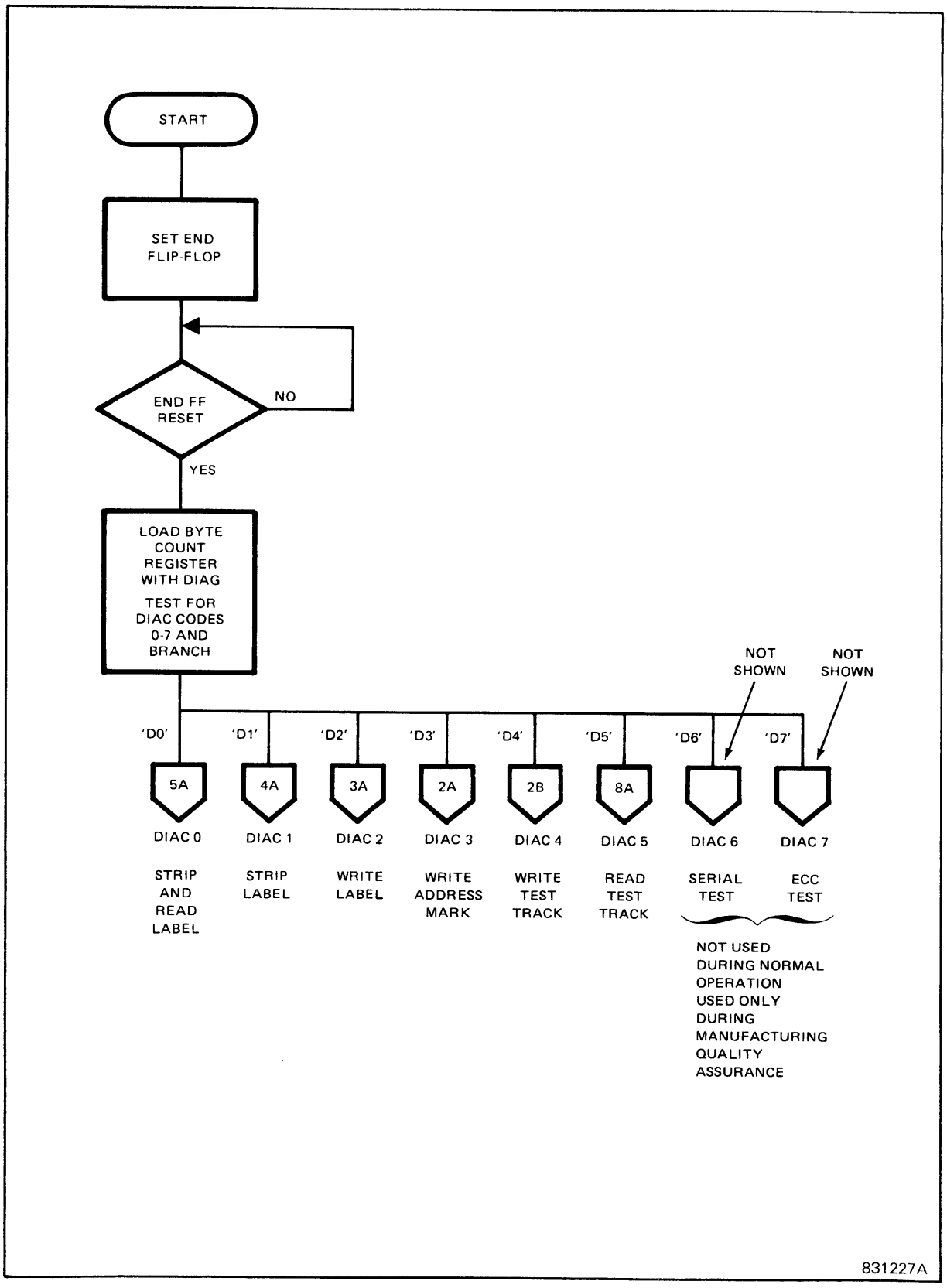


Figure E-2. DIAC Decode (Sheet 1 of 8)

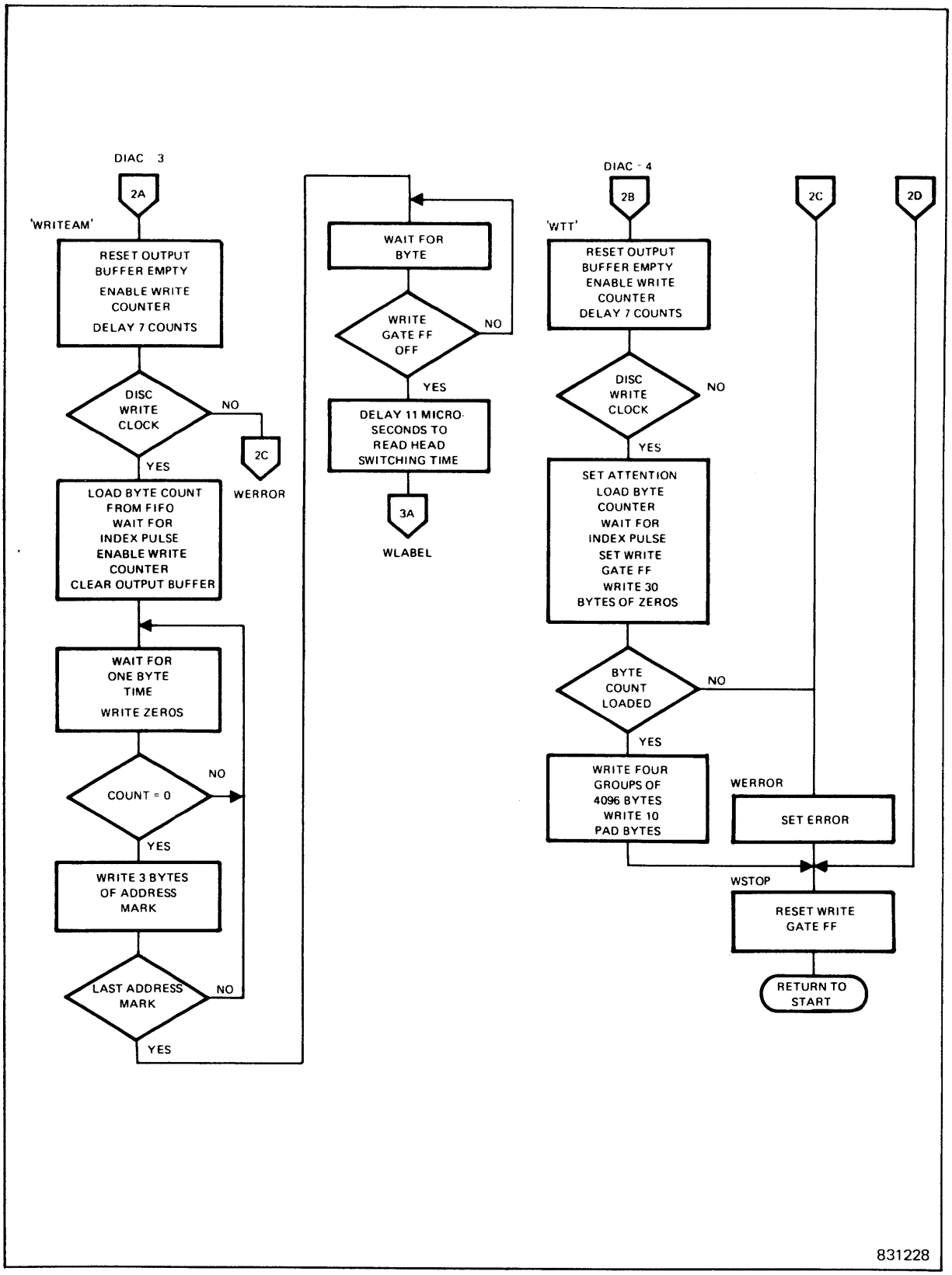


Figure E-2. Write Address Mark (DIAC 3) and Write Test Track (DIAC 4) (Sheet 2 of 8)

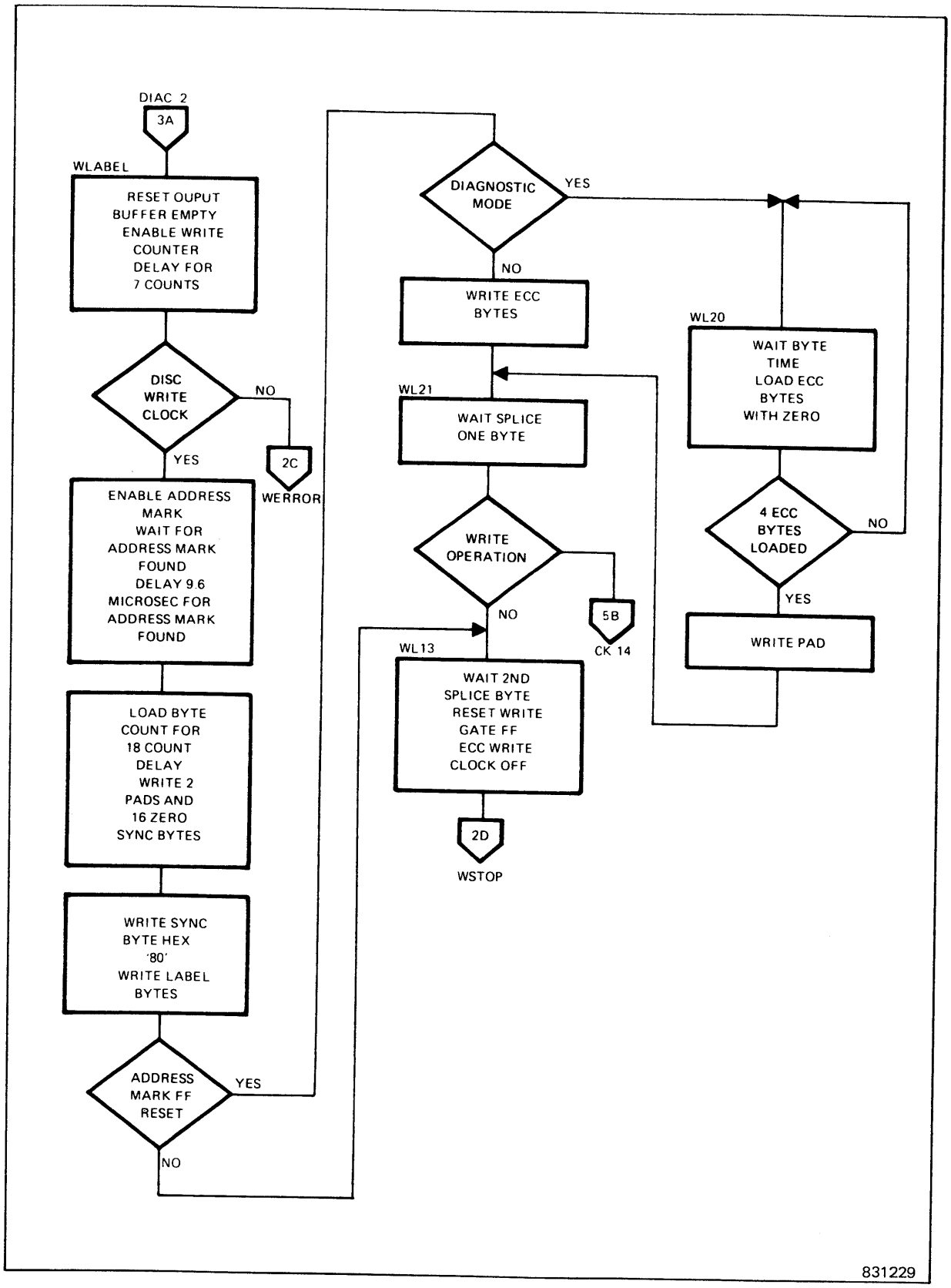


Figure E-2. Write Label (DIAC 2) (Sheet 3 of 8)

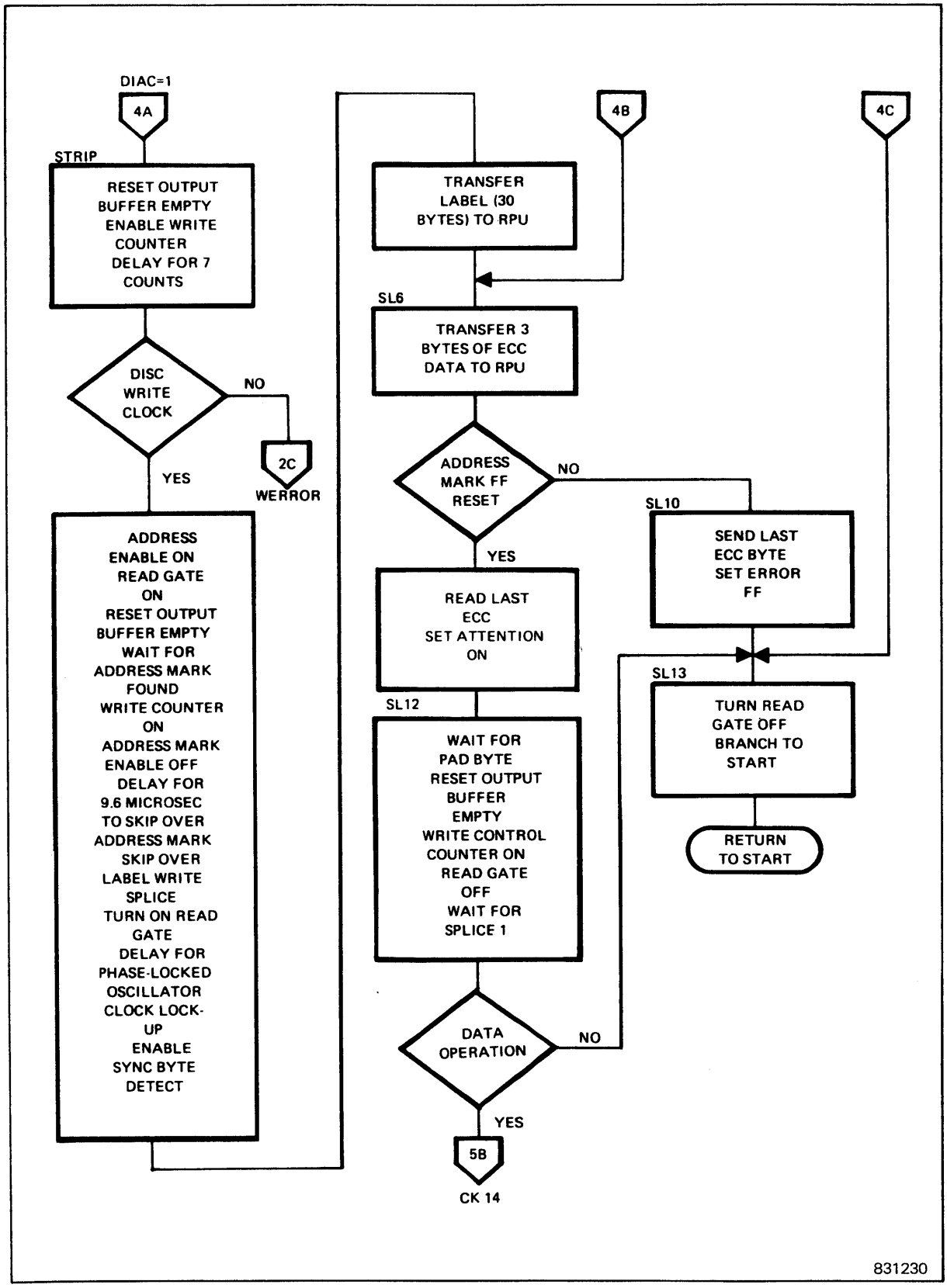


Figure E-2. Strip Label (DIAC 1) (Sheet 4 of 8)

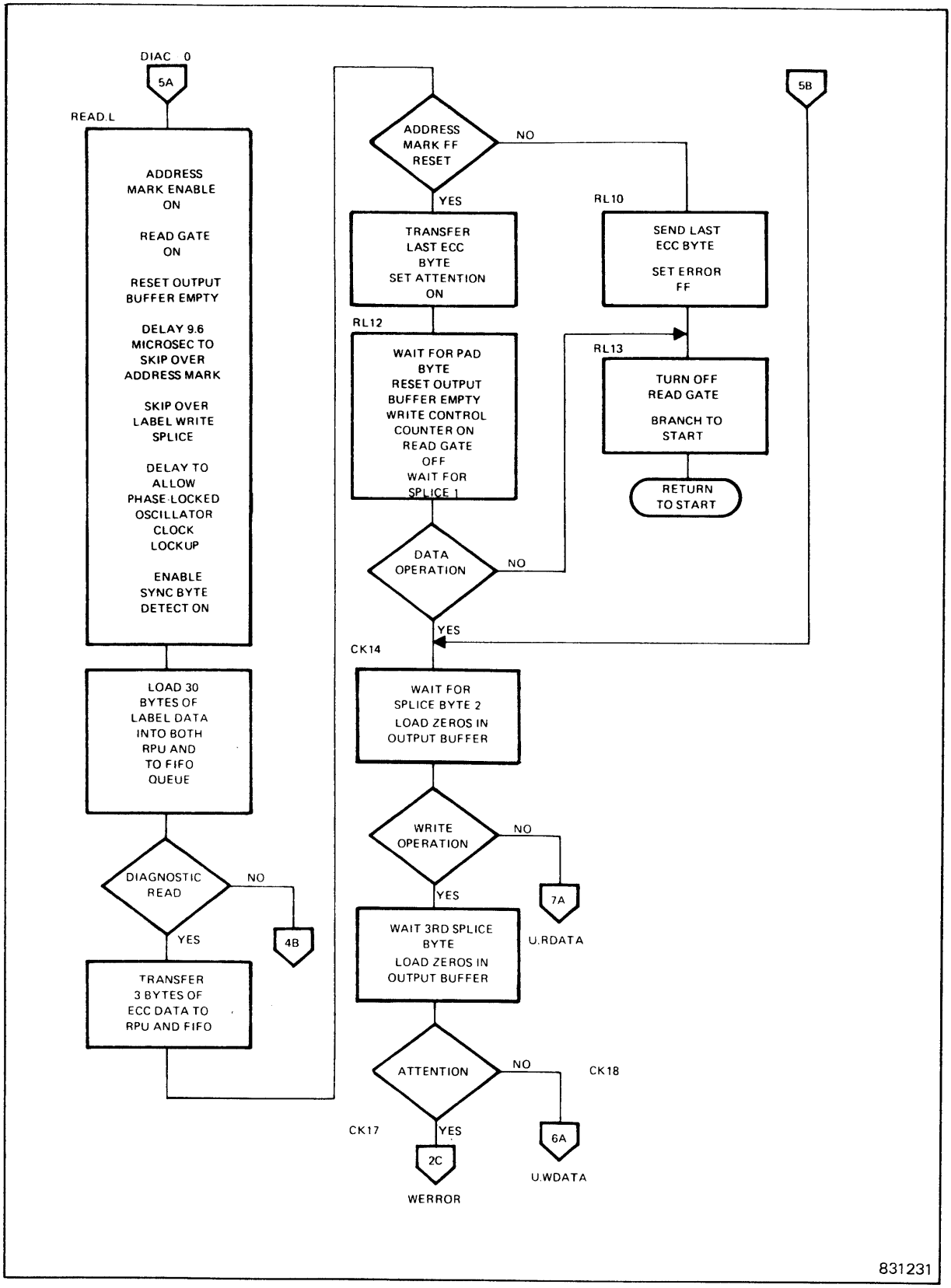


Figure E-2. Read Label (DIAC 0) (Sheet 5 of 8)

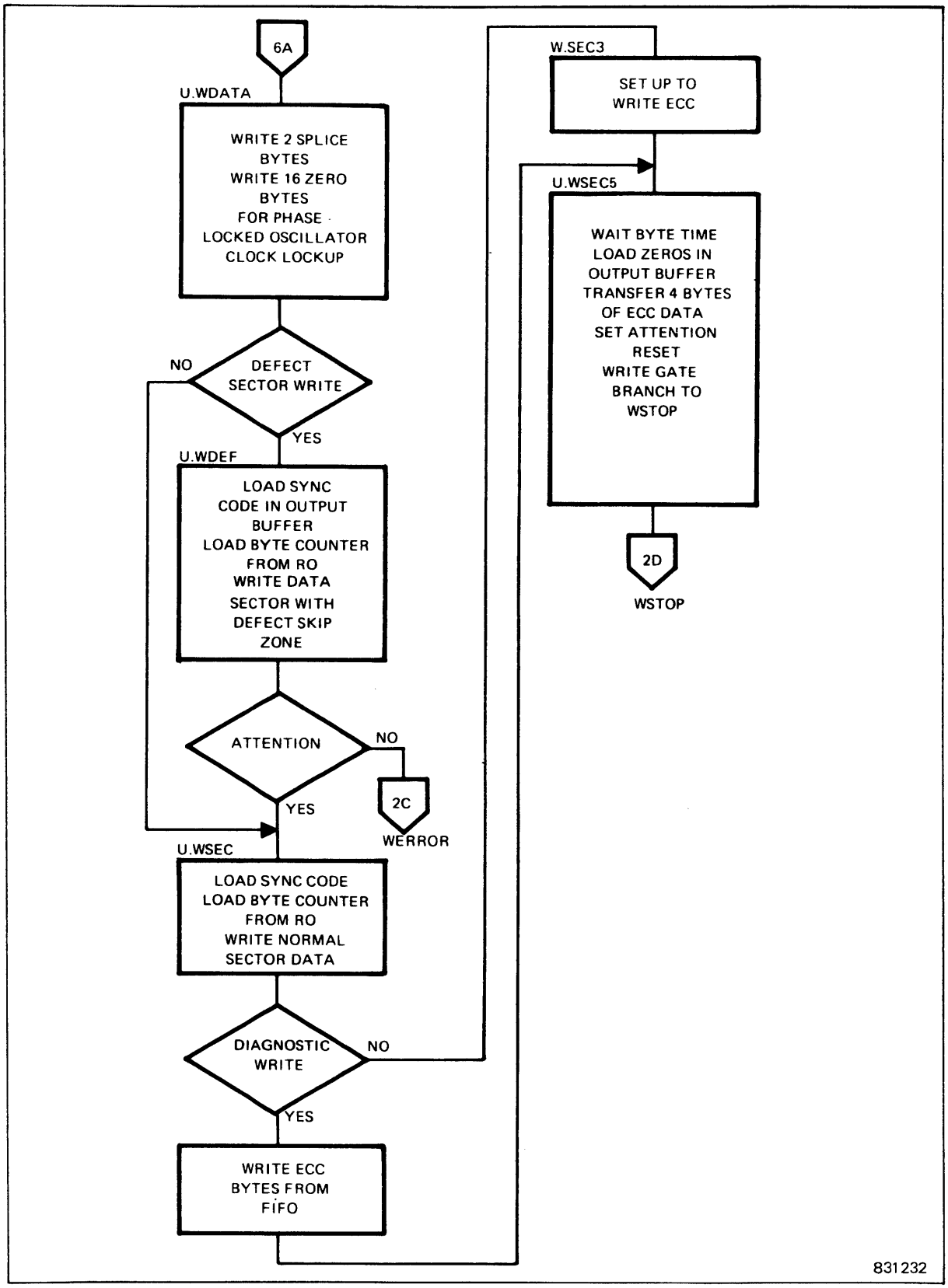


Figure E-2. Write Data (Sheet 6 of 8)

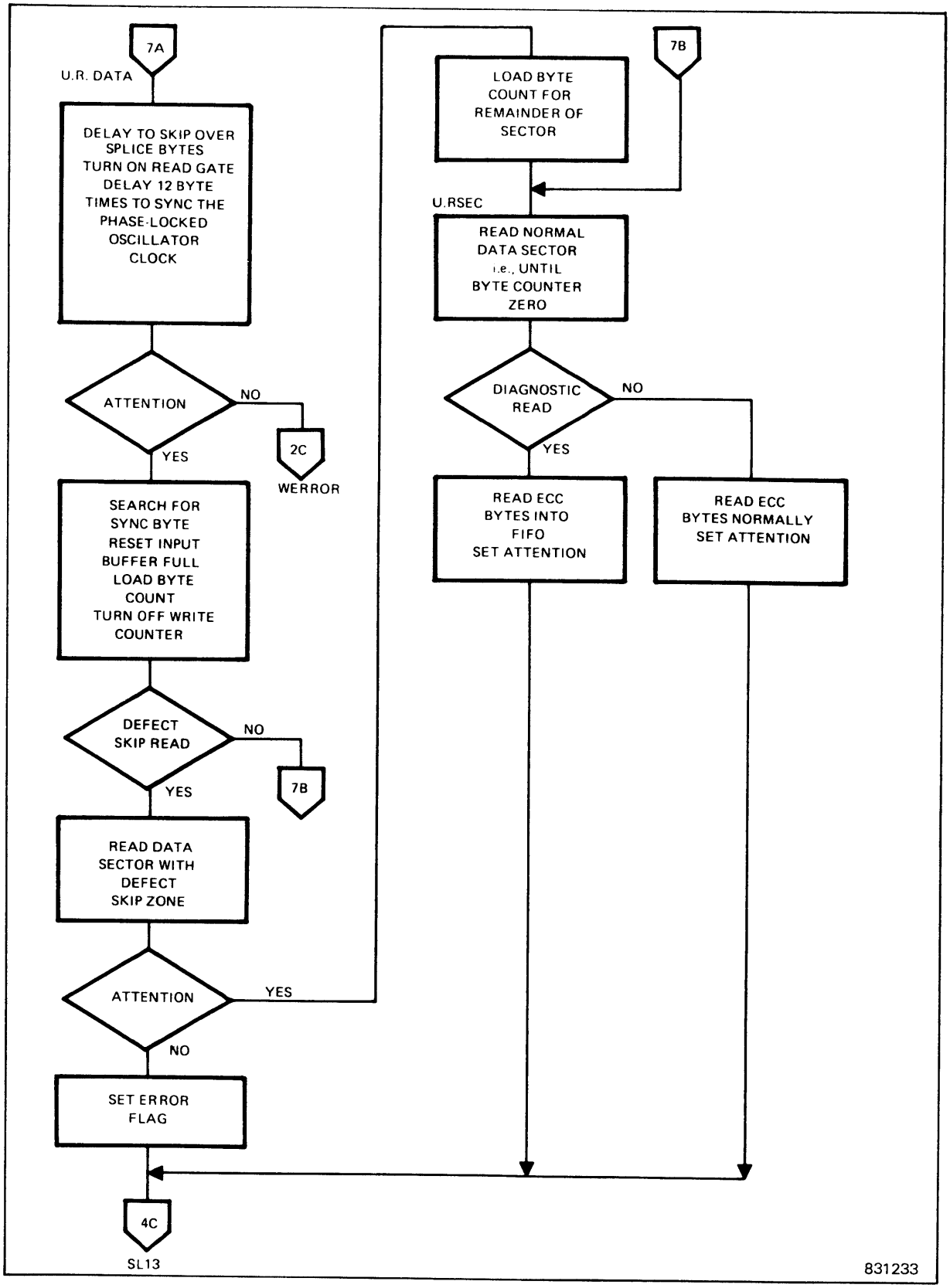


Figure E-2. Read Data (Sheet 7 of 8)

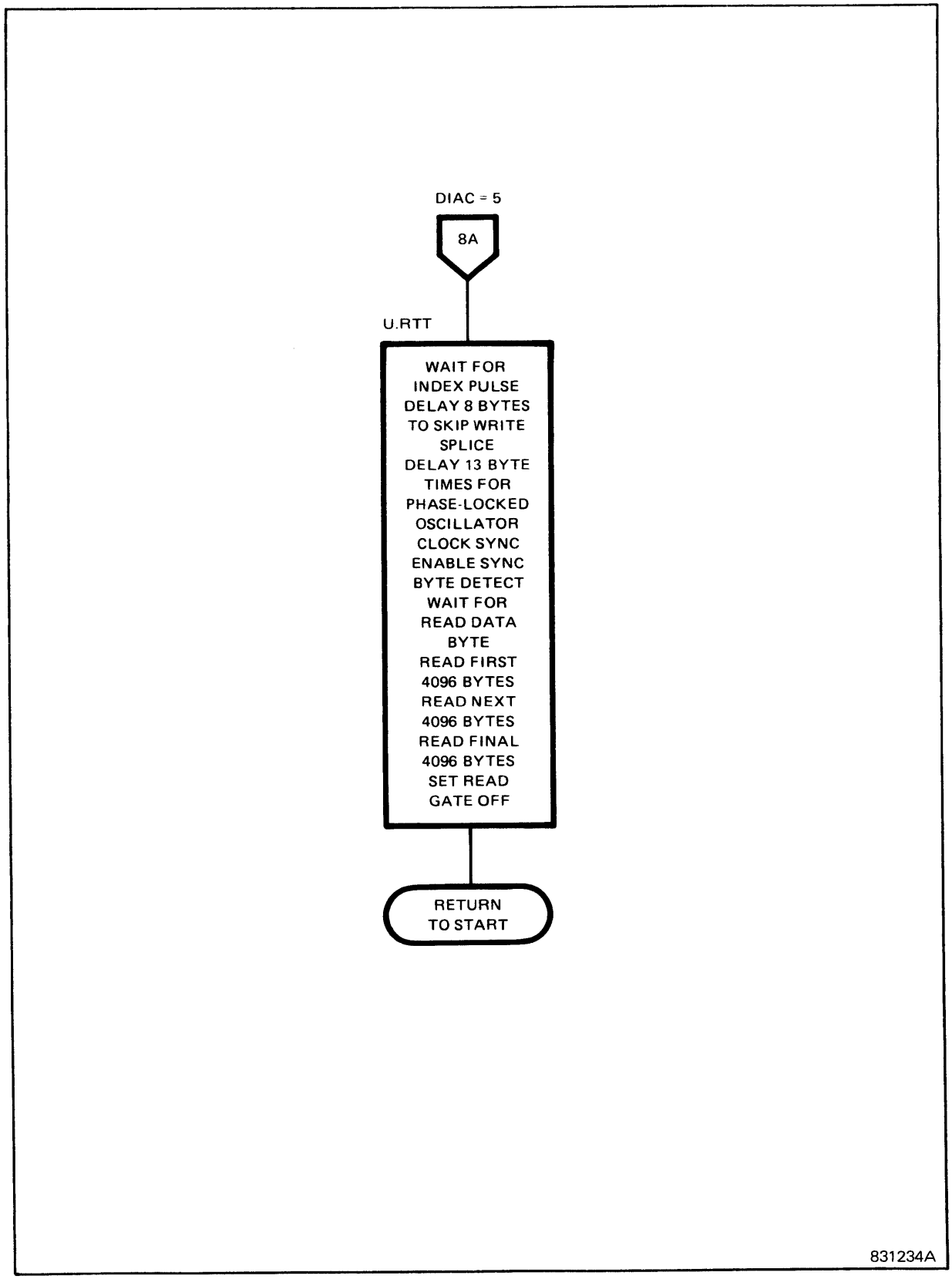


Figure E-2. Read Test Track (DIAC 5) (Sheet 8 of 8)

APPENDIX F

UNIVERSAL DISC PROCESSOR FIRMWARE FLOW DIAGRAMS

This appendix provides generalized flow diagrams for the UDP firmware. These diagrams will aid in understanding the UDP microprogram operation. Figure F-1 identifies the symbols and conventions used in the flow diagrams.

The flow diagrams are general in nature. For purposes of clarity, many individual steps and branches have been omitted. In addition, a number of minor subroutines have been omitted. However, the flow diagrams show the major routines, commands, and op codes. No attempt has been made to show all subroutine interconnections, since the purpose of the flow diagrams is visualization of program flow.

To determine exact program steps, refer to the program listing. The flow diagrams can be cross-referenced to the program listing by using the subroutine labels. Subroutines labels are marked upper left of each symbol. An alphabetical listing of the subroutine labels is also found at the end of the program listing. By looking up the subroutine label in this list, the program listing line(s) where the label appears can be determined.

Flow diagrams are arranged in the same sequence as the subroutines in the firmware listing.

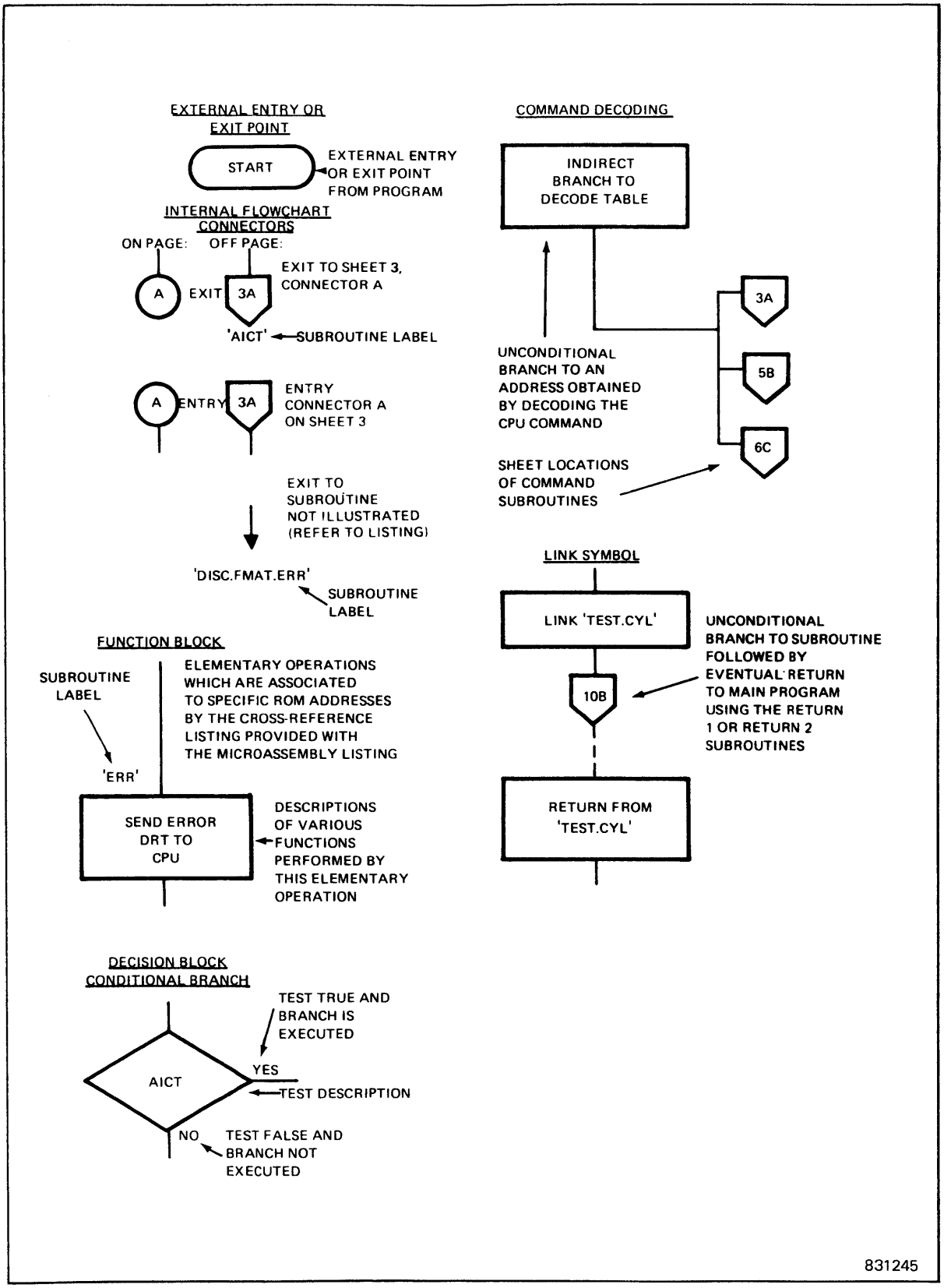


Figure F-1. Flow Chart - Symbol Identification and Conventions

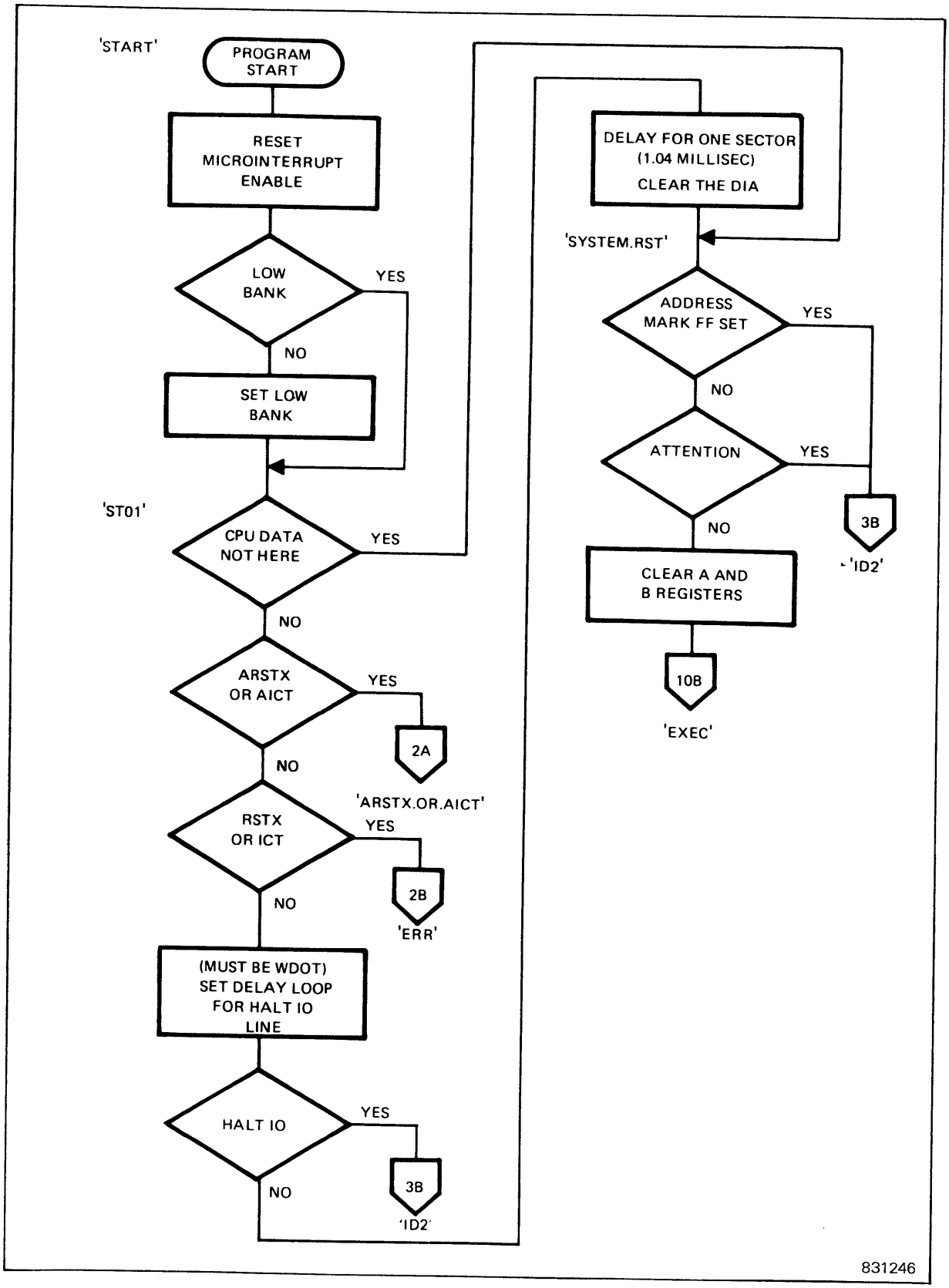


Figure F-2. Program Start (Sheet 1 of 41)

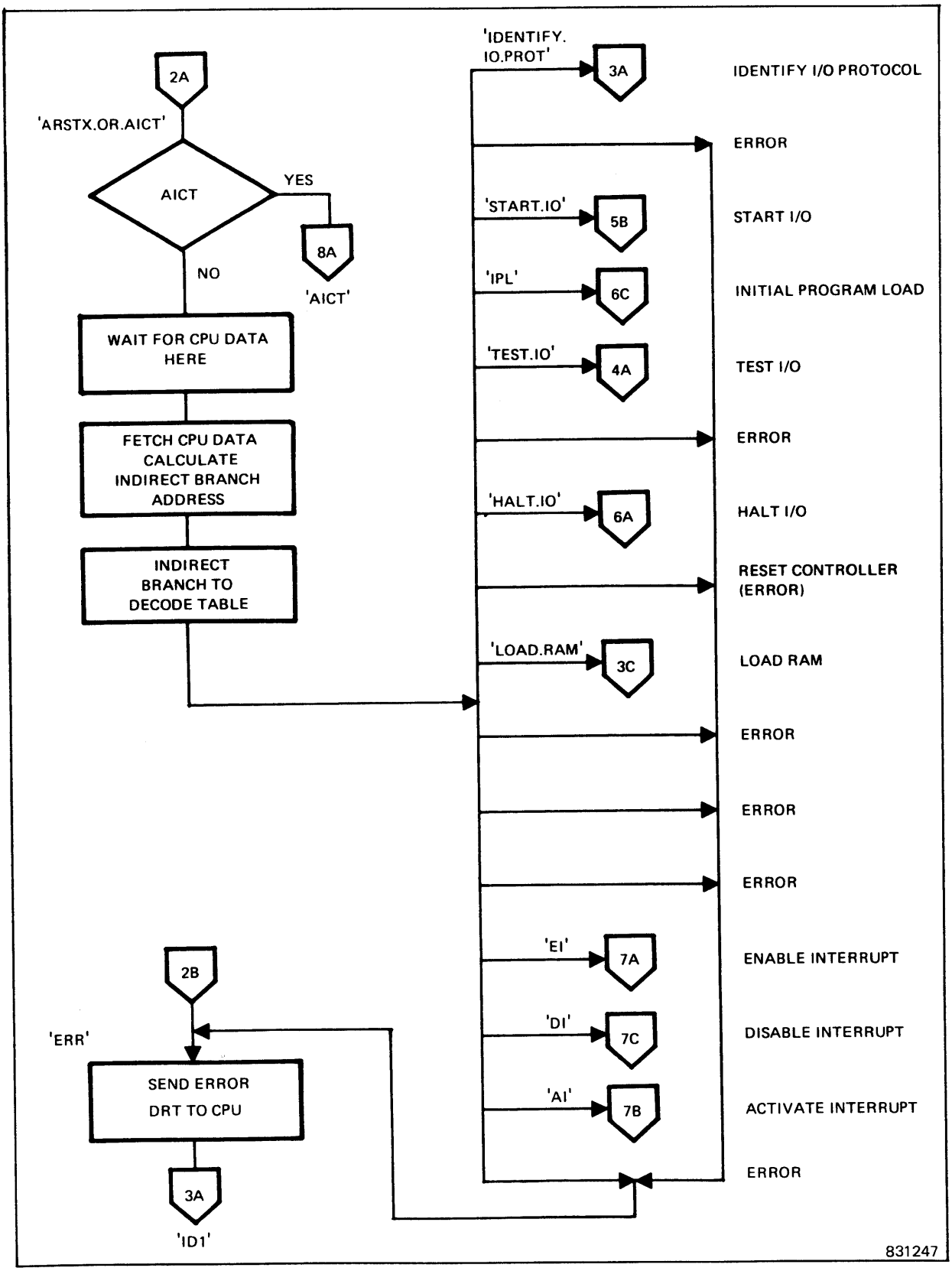


Figure F-2. Command Decoding (Sheet 2 of 41)

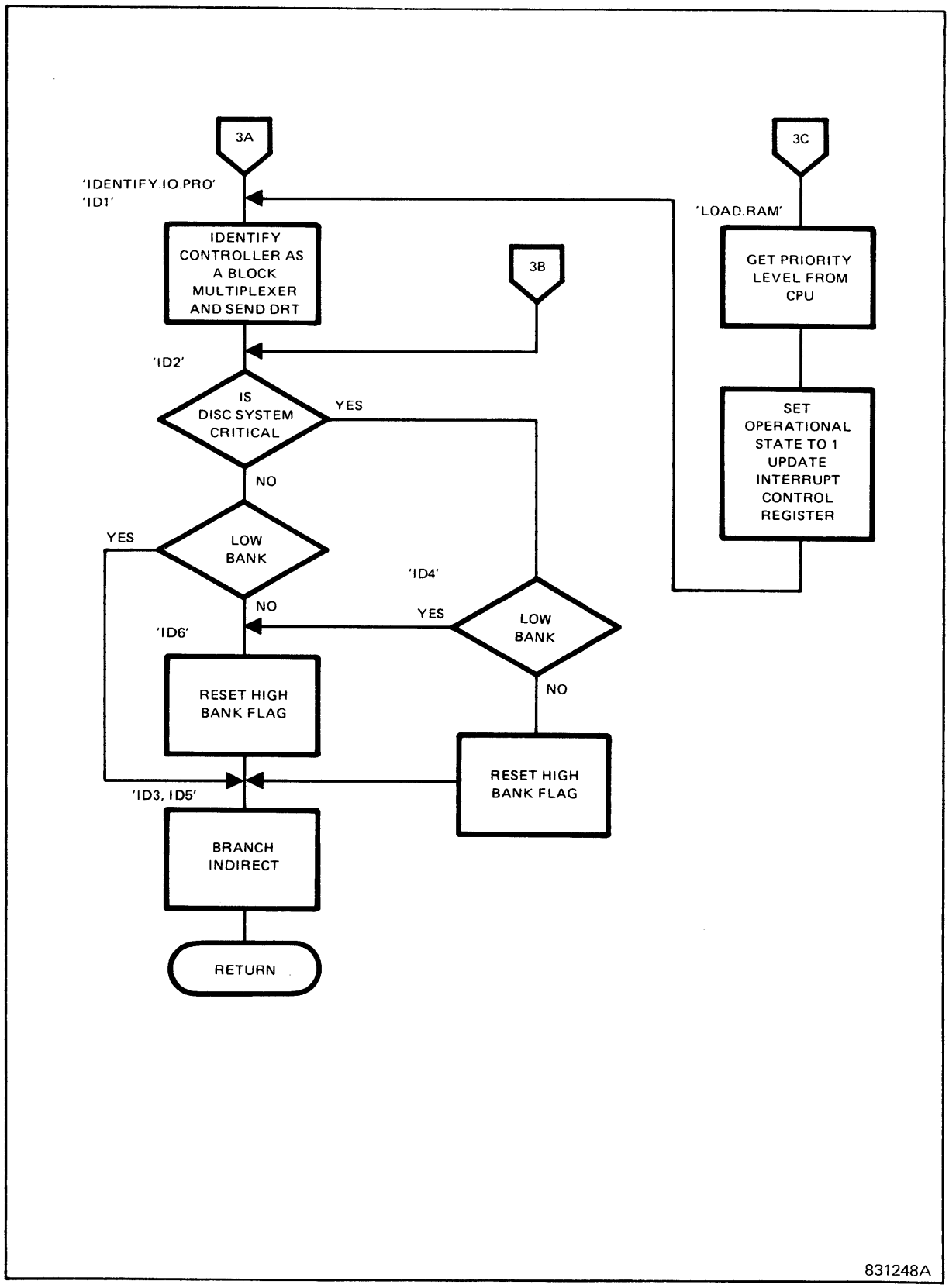


Figure F-2. Identify I/O Protocol; Load RAM (Sheet 3 of 41)

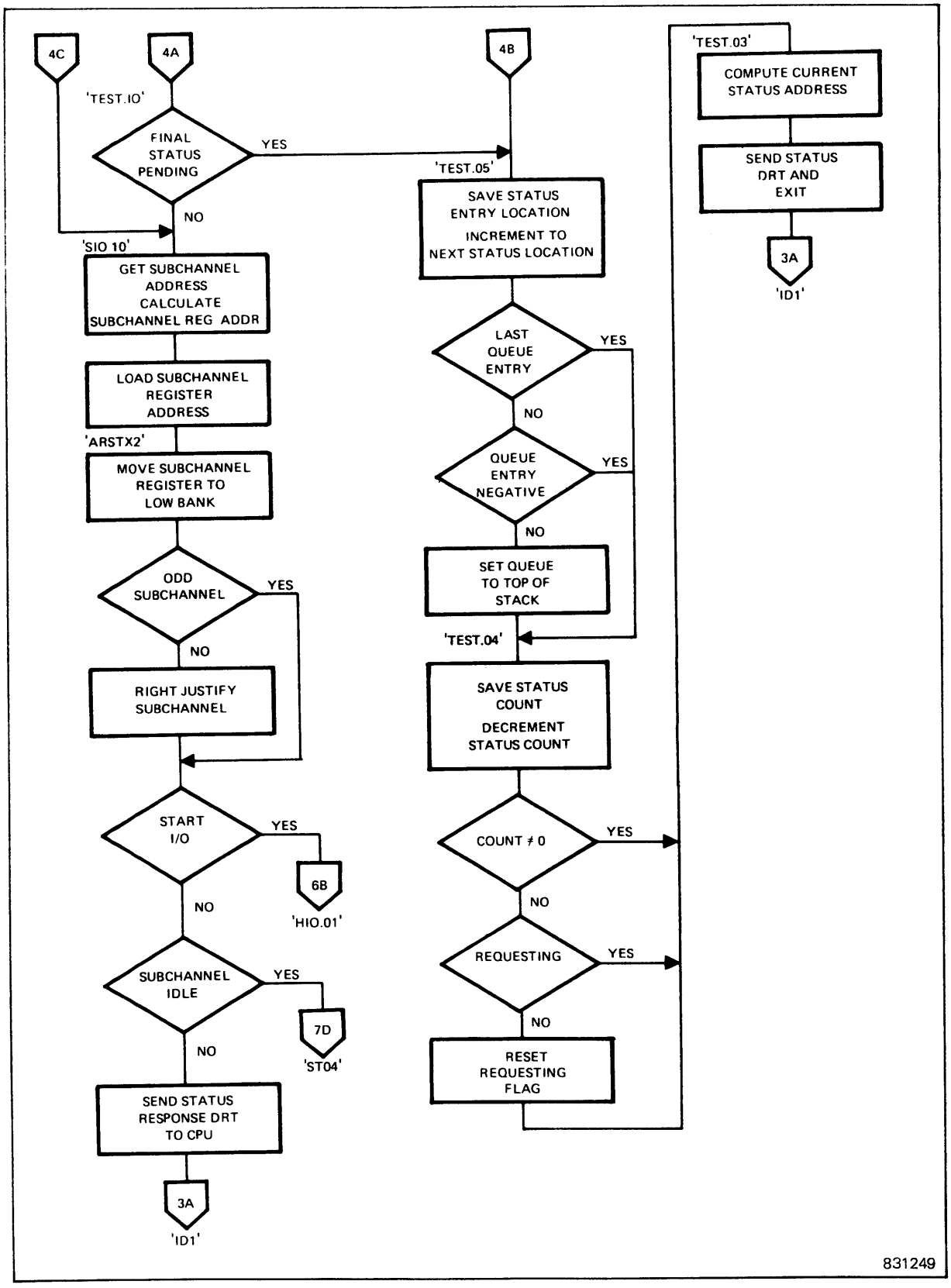


Figure F-2. Test I/O (Sheet 4 of 41)

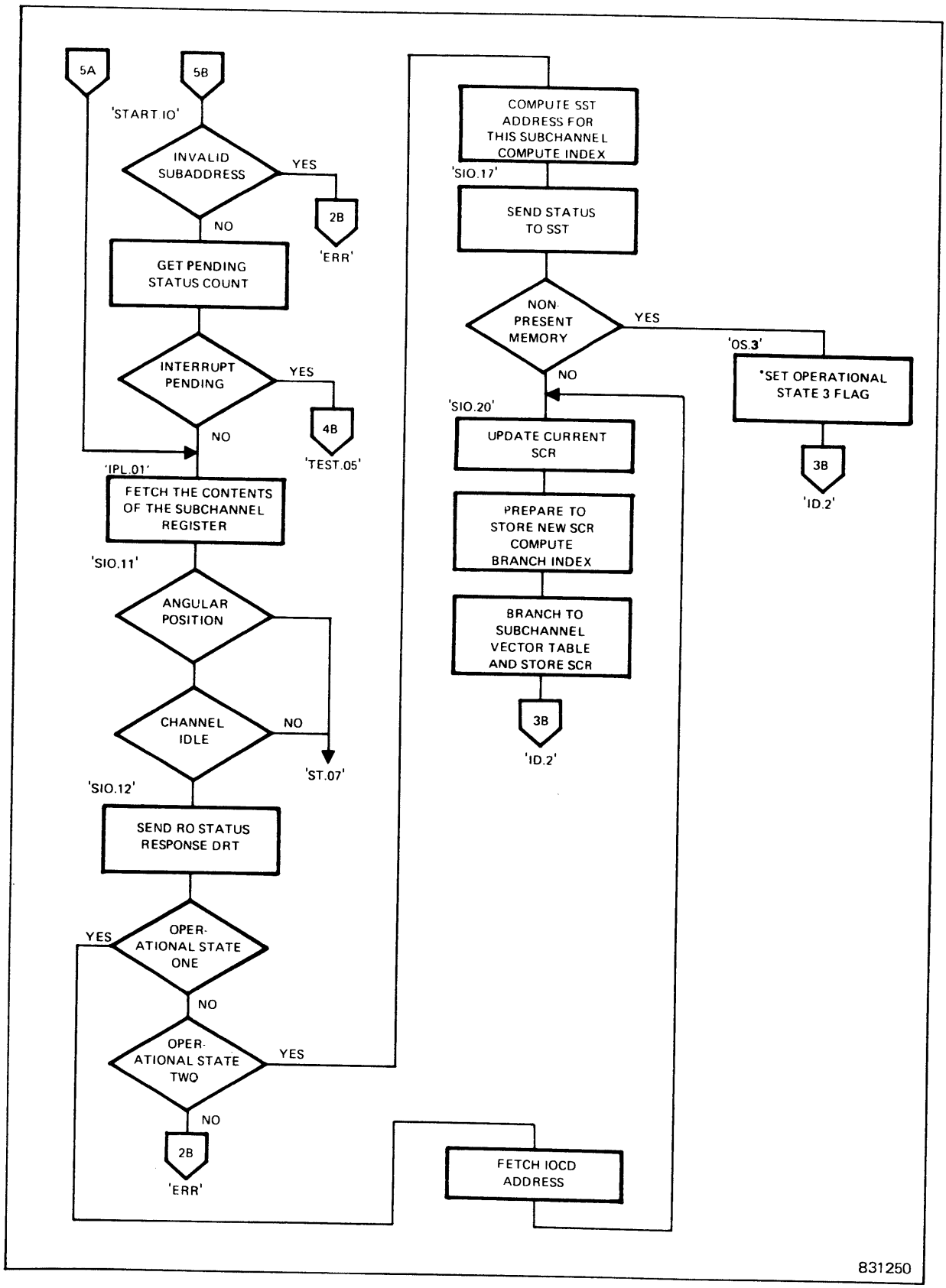
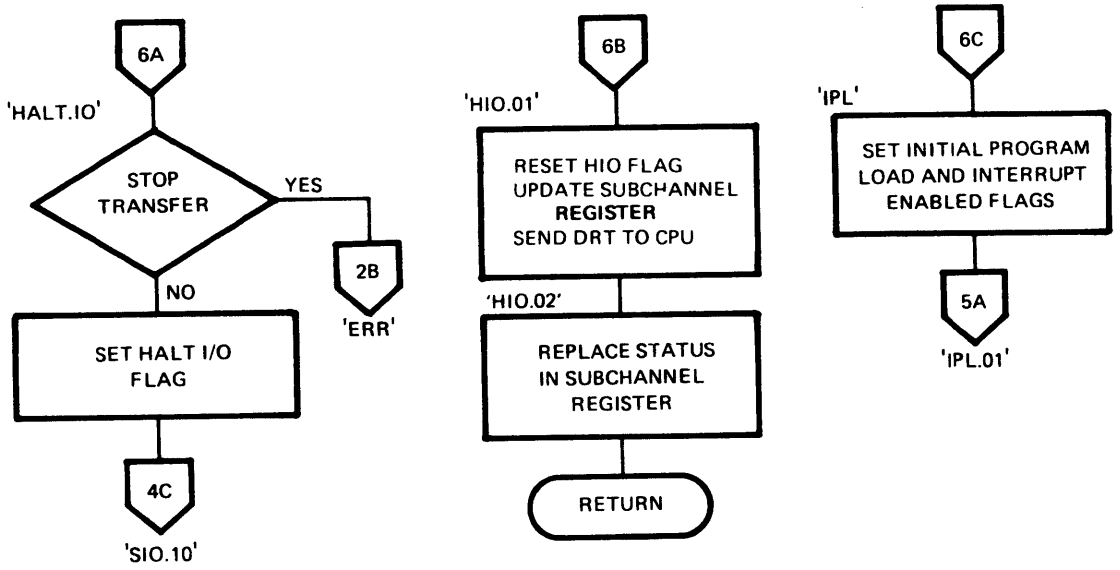


Figure F-2. Start I/O (Sheet 5 of 41)



831251

Figure F-2. Halt I/O; Initial Program Load (Sheet 6 of 41)

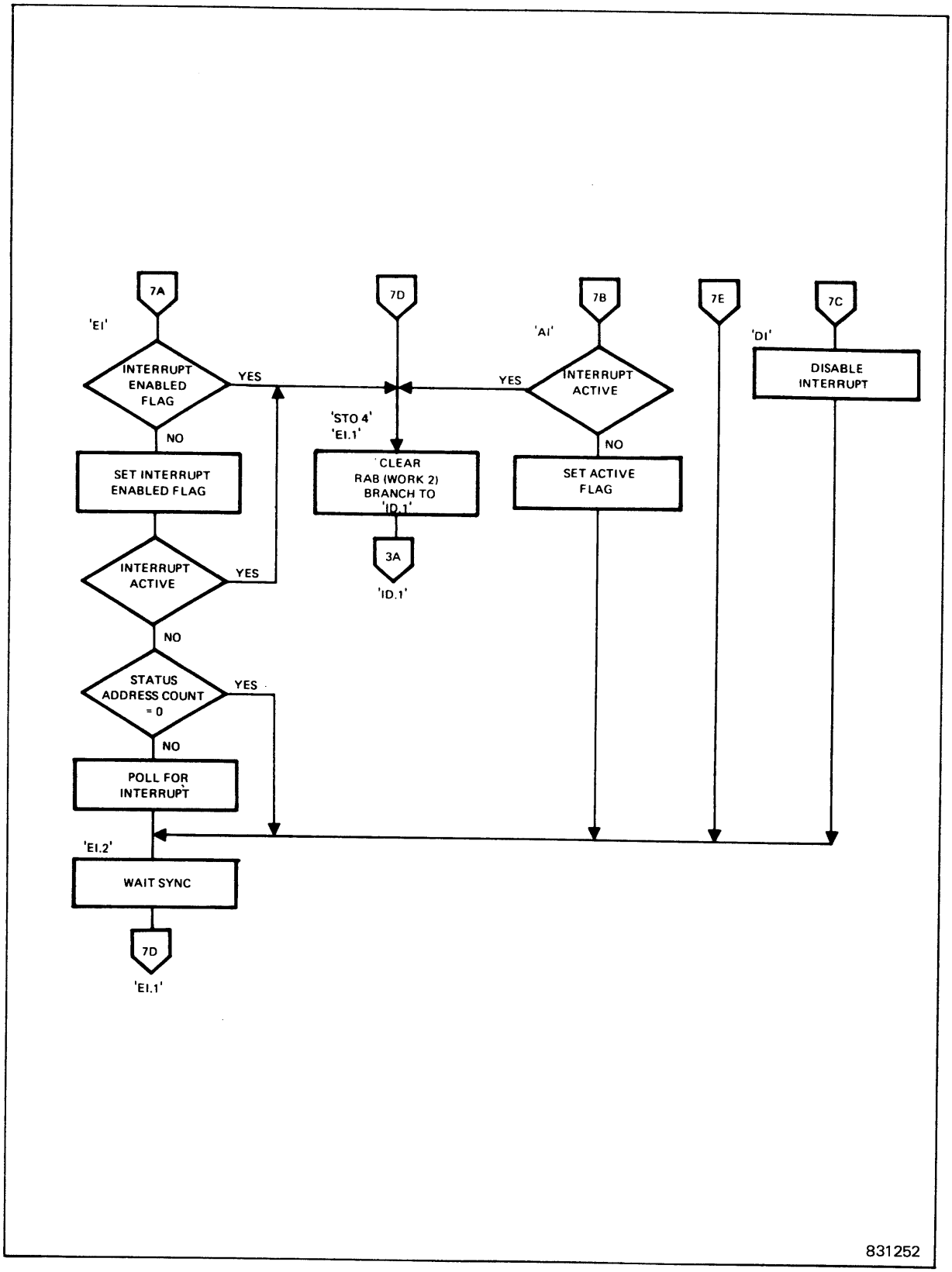


Figure F-2. Enable Interrupt; Activate Interrupt; Deactivate Channel Interrupt (Sheet 7 of 41)

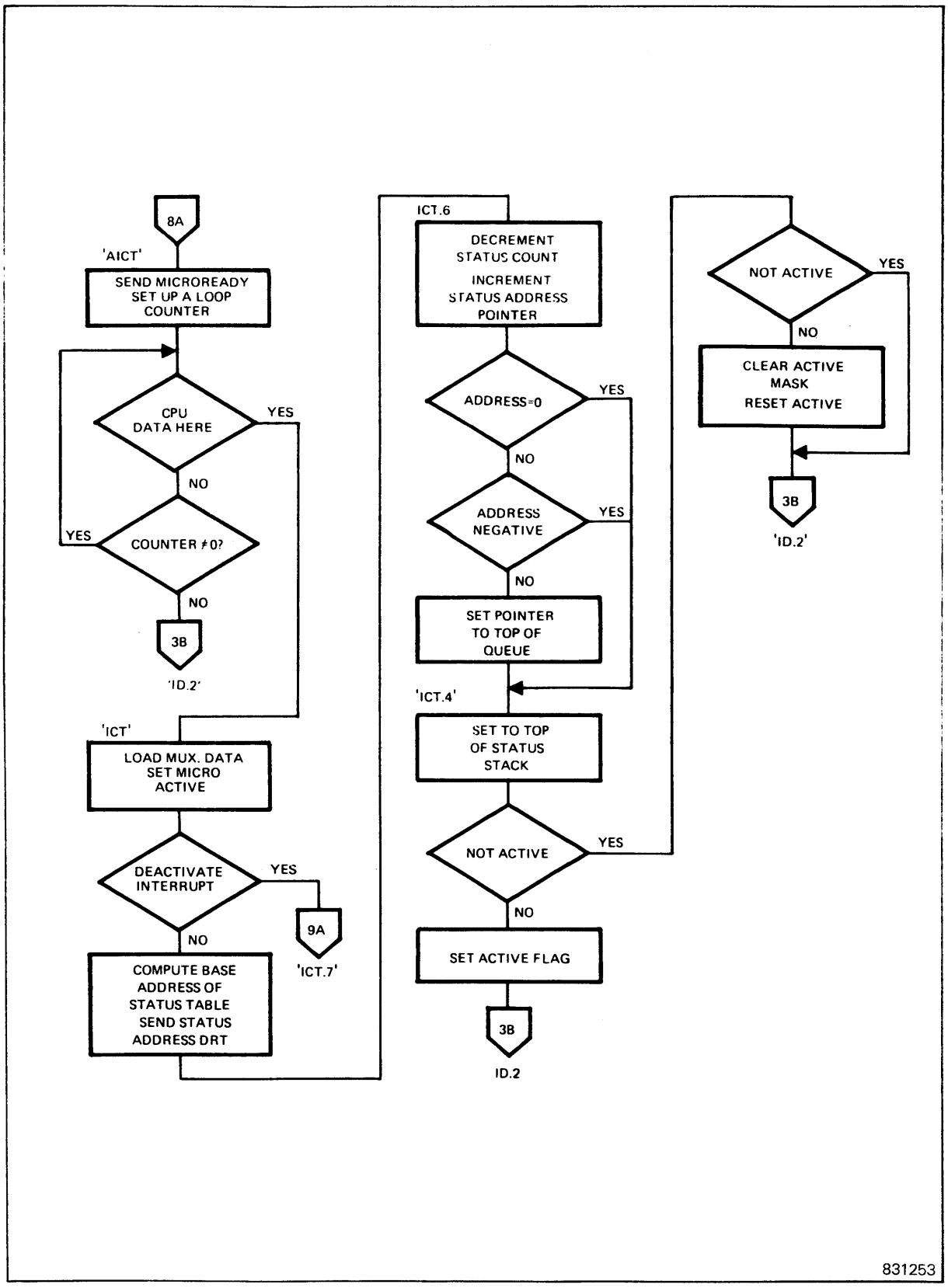


Figure F-2. AICT/ICT (Sheet 8 of 41)

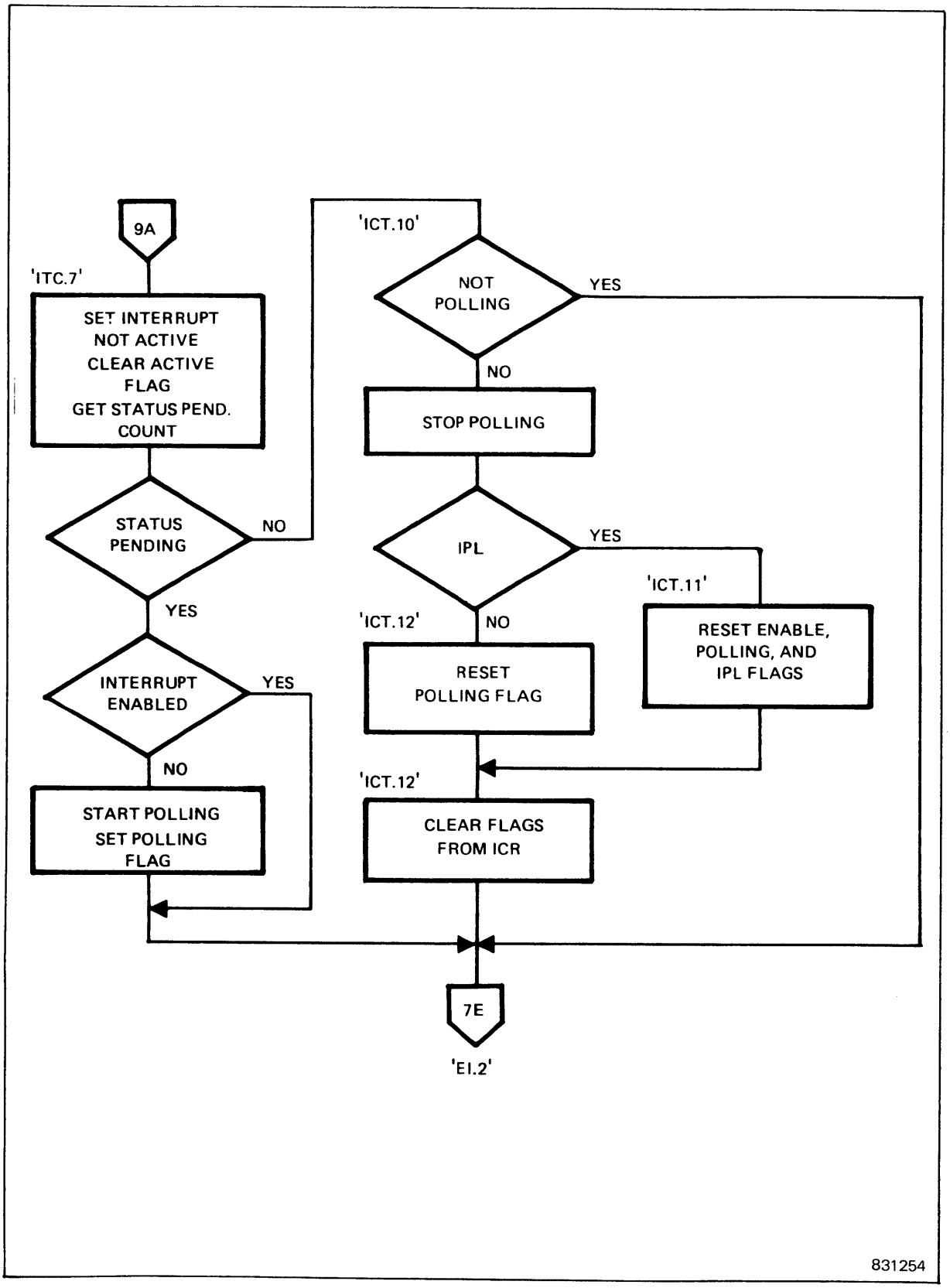


Figure F-2. ITC 7 (Sheet 9 of 41)

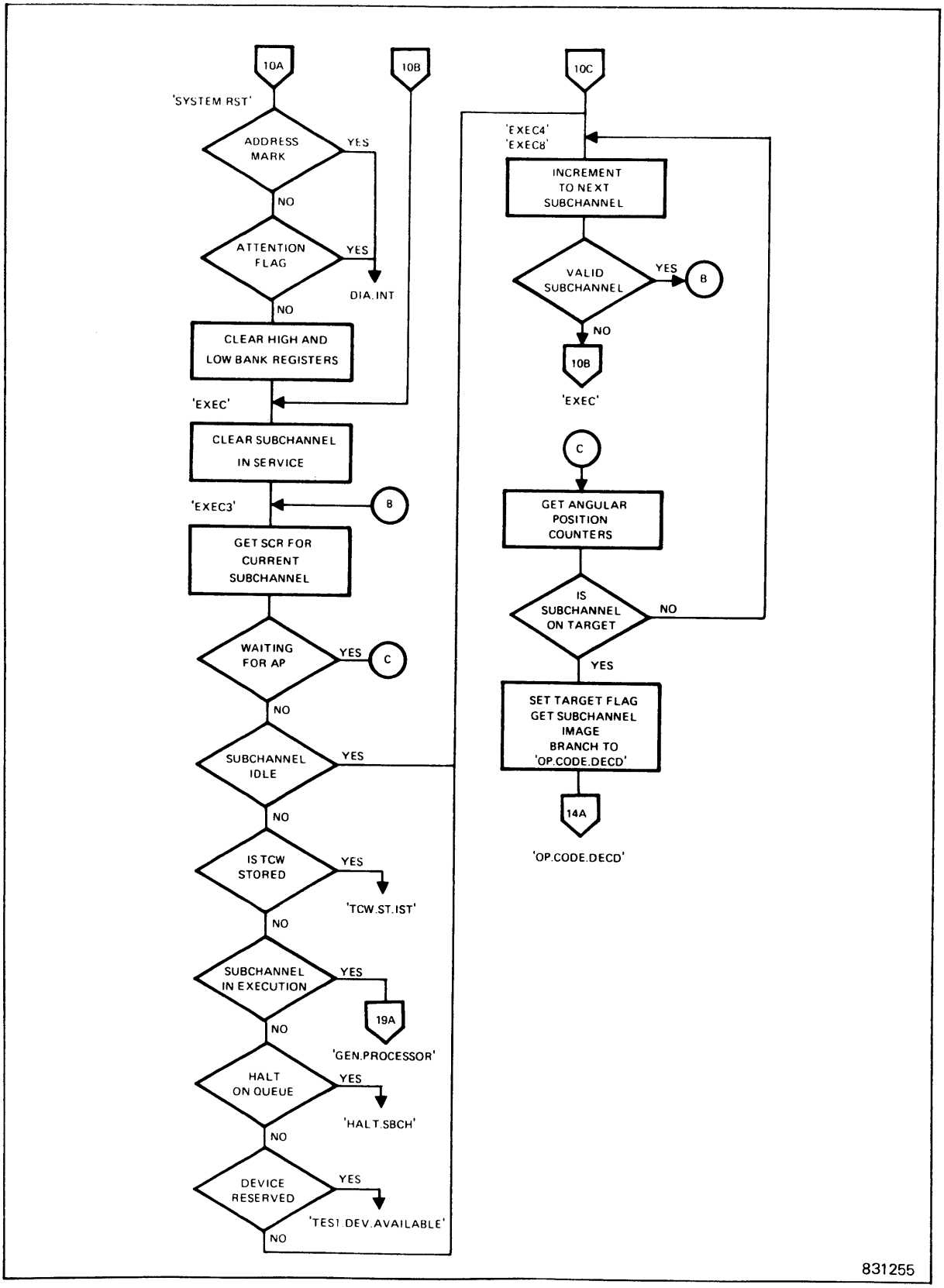


Figure F-2. System Reset and Executive Schedules (Sheet 10 of 41)

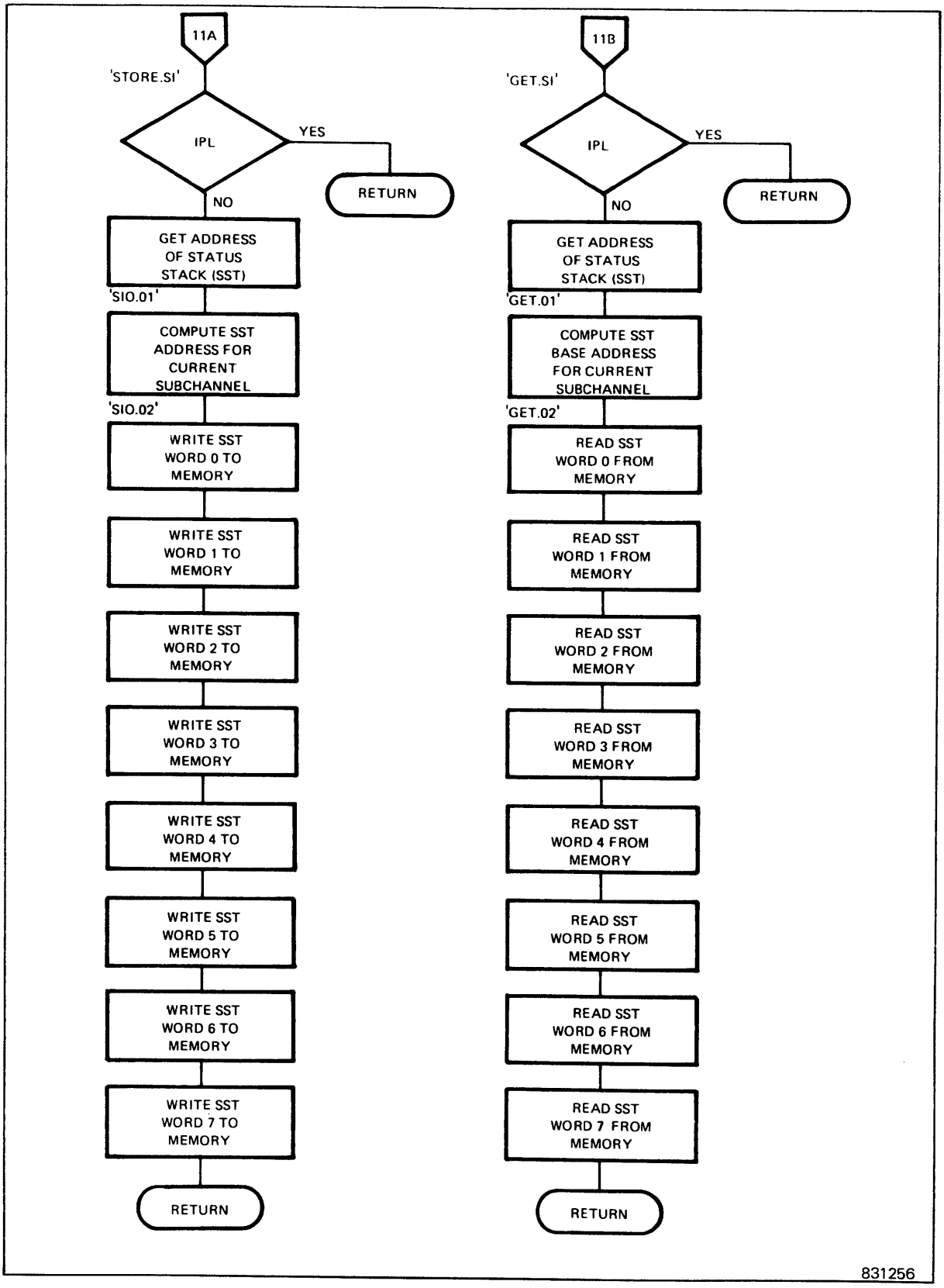


Figure F-2. Store Subchannel Image; Get Subchannel Image (Sheet 11 of 41)

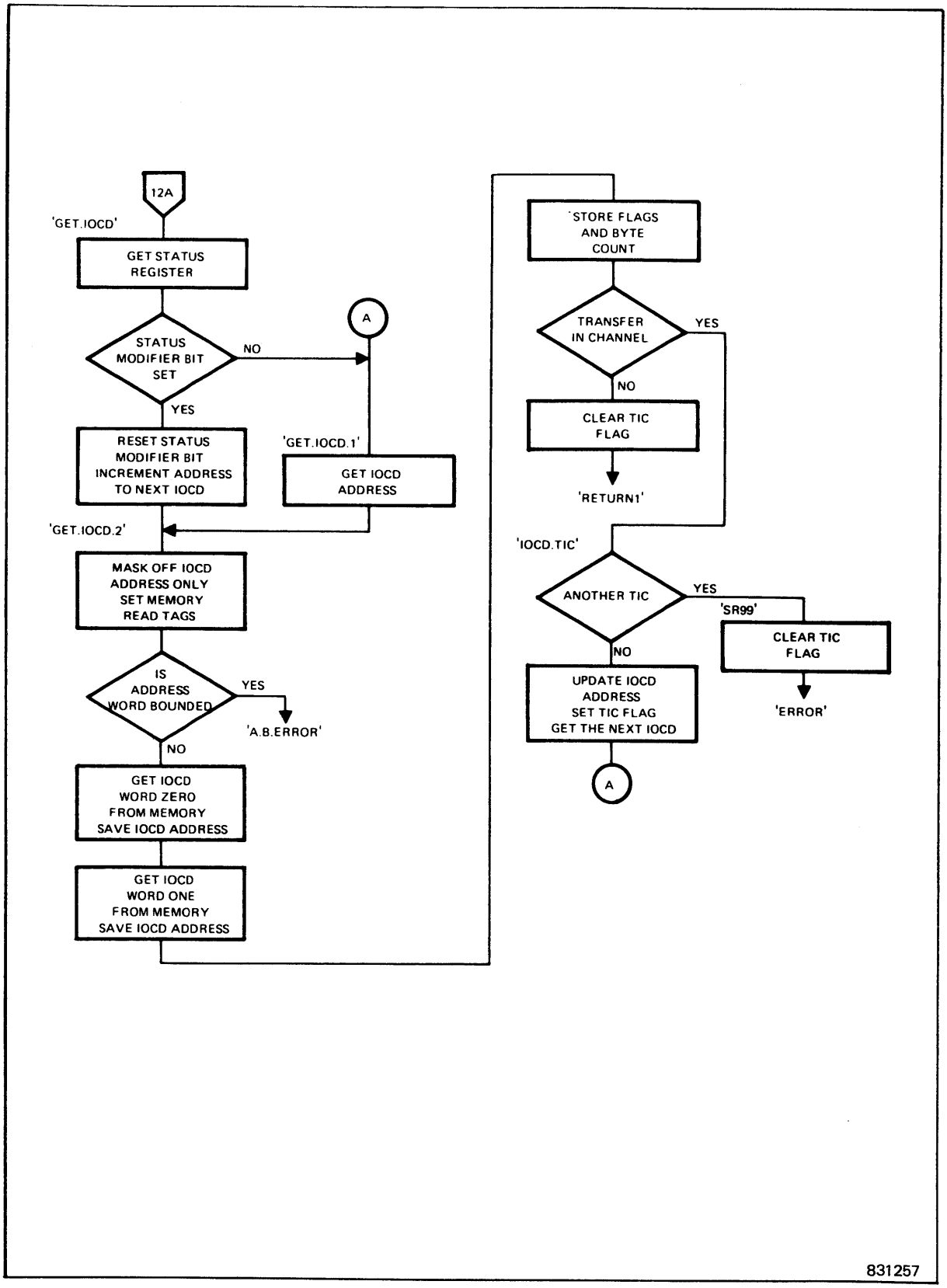


Figure F-2. Get IOCD (Sheet 12 of 41)

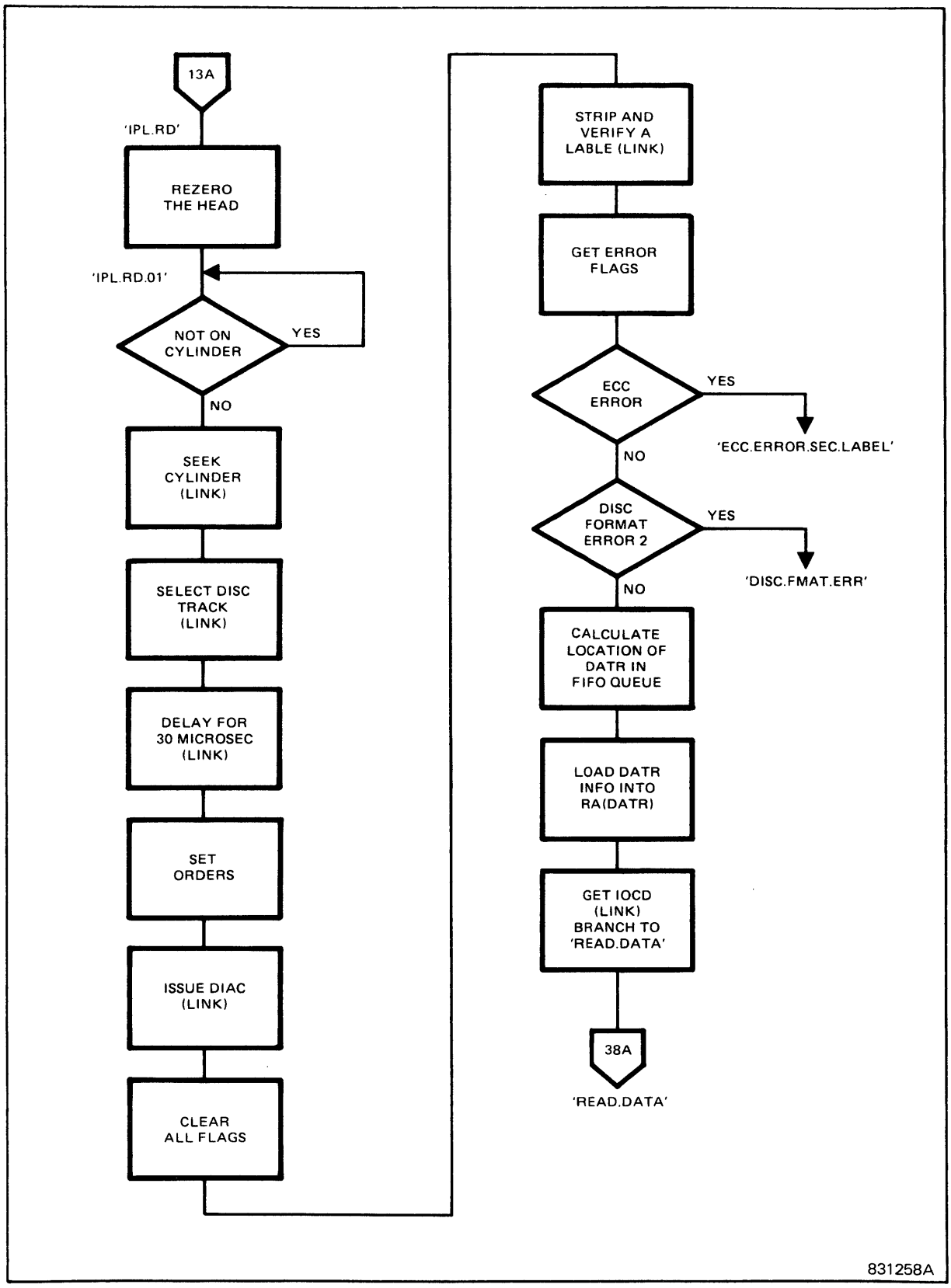


Figure F-2. IPL Read (Sheet 13 of 41)

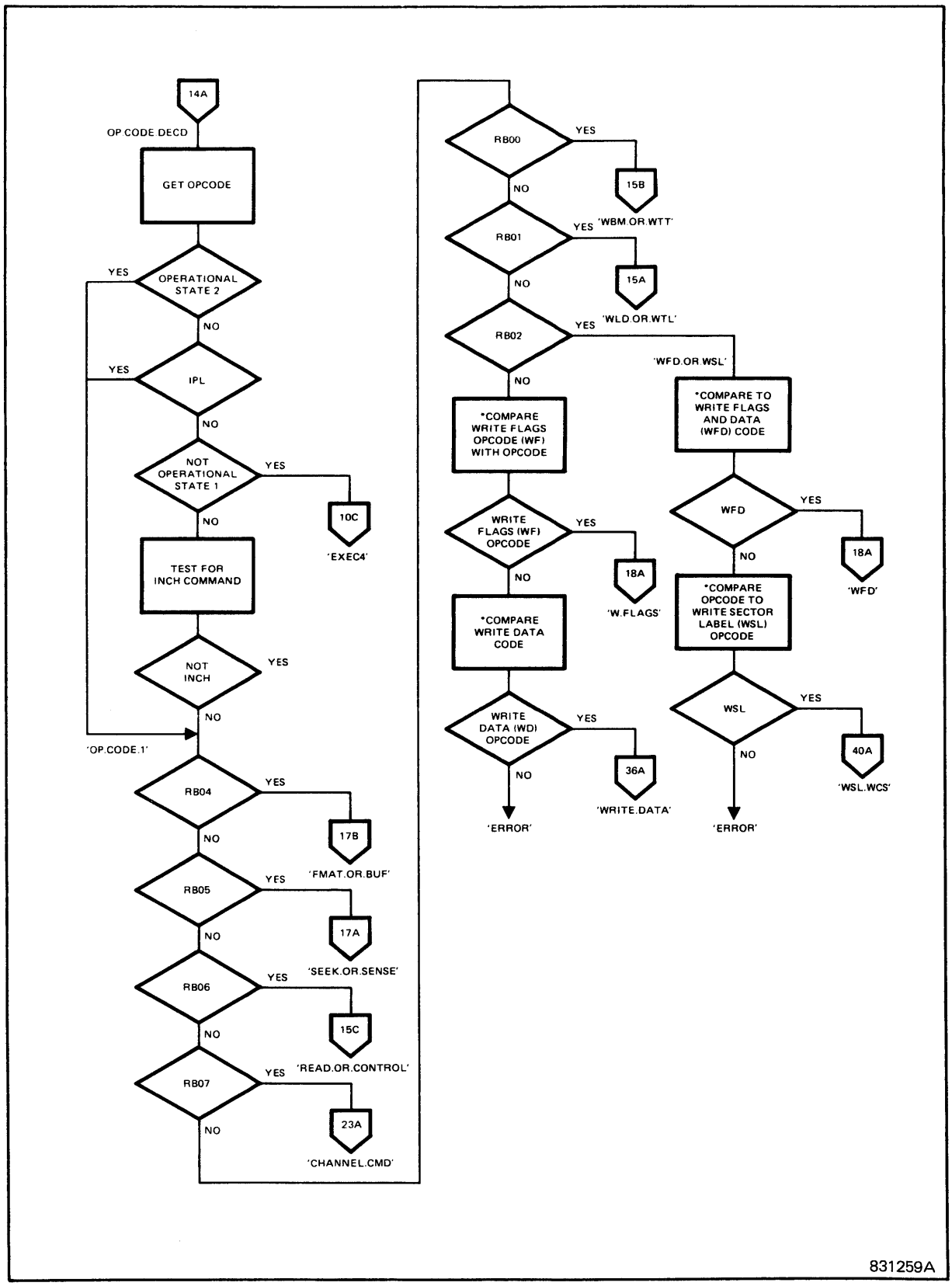


Figure F-2. Op Code Decode (Sheet 14 of 41)

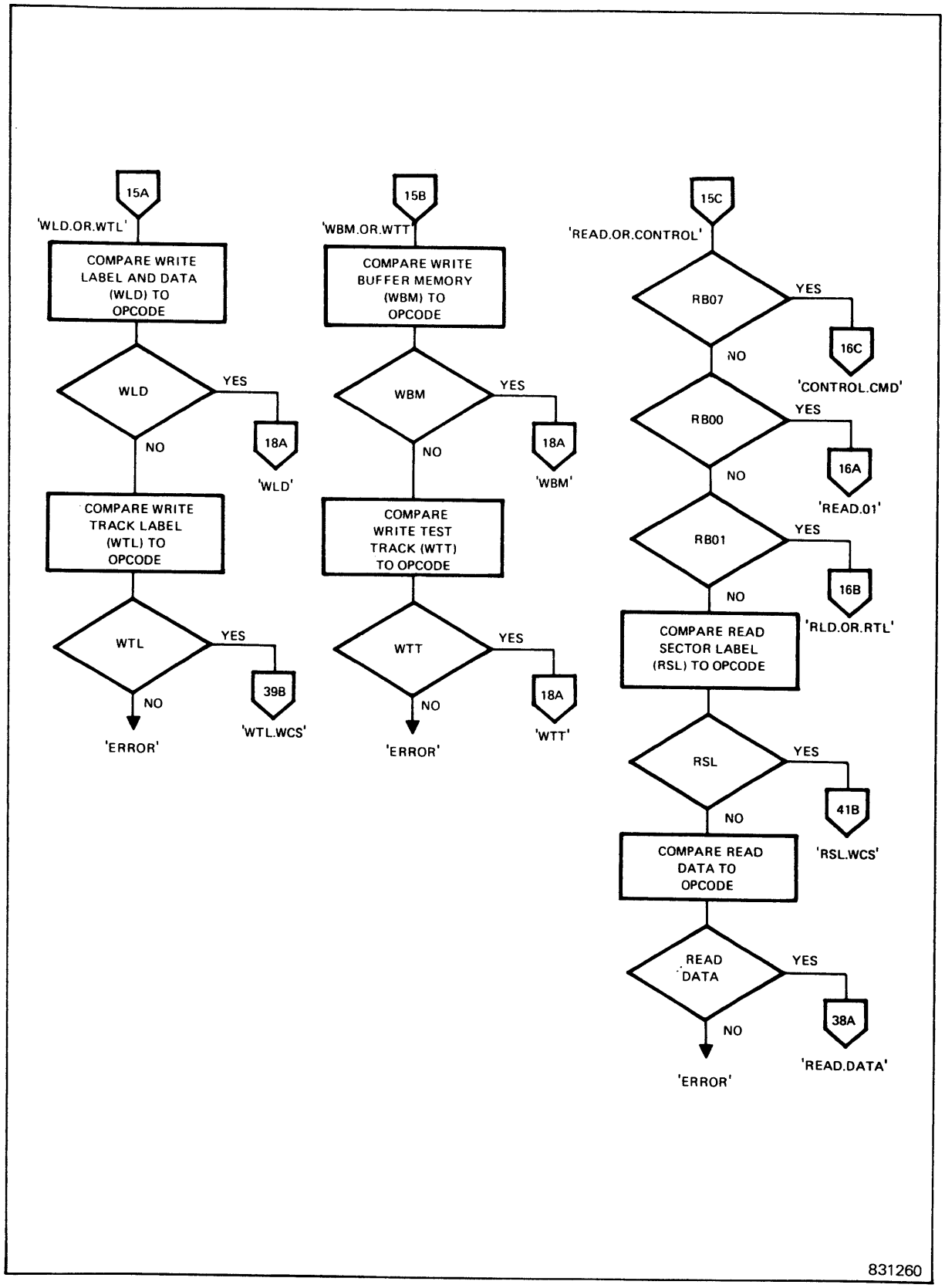


Figure F-2. Op Code Decode (Sheet 15 of 41)

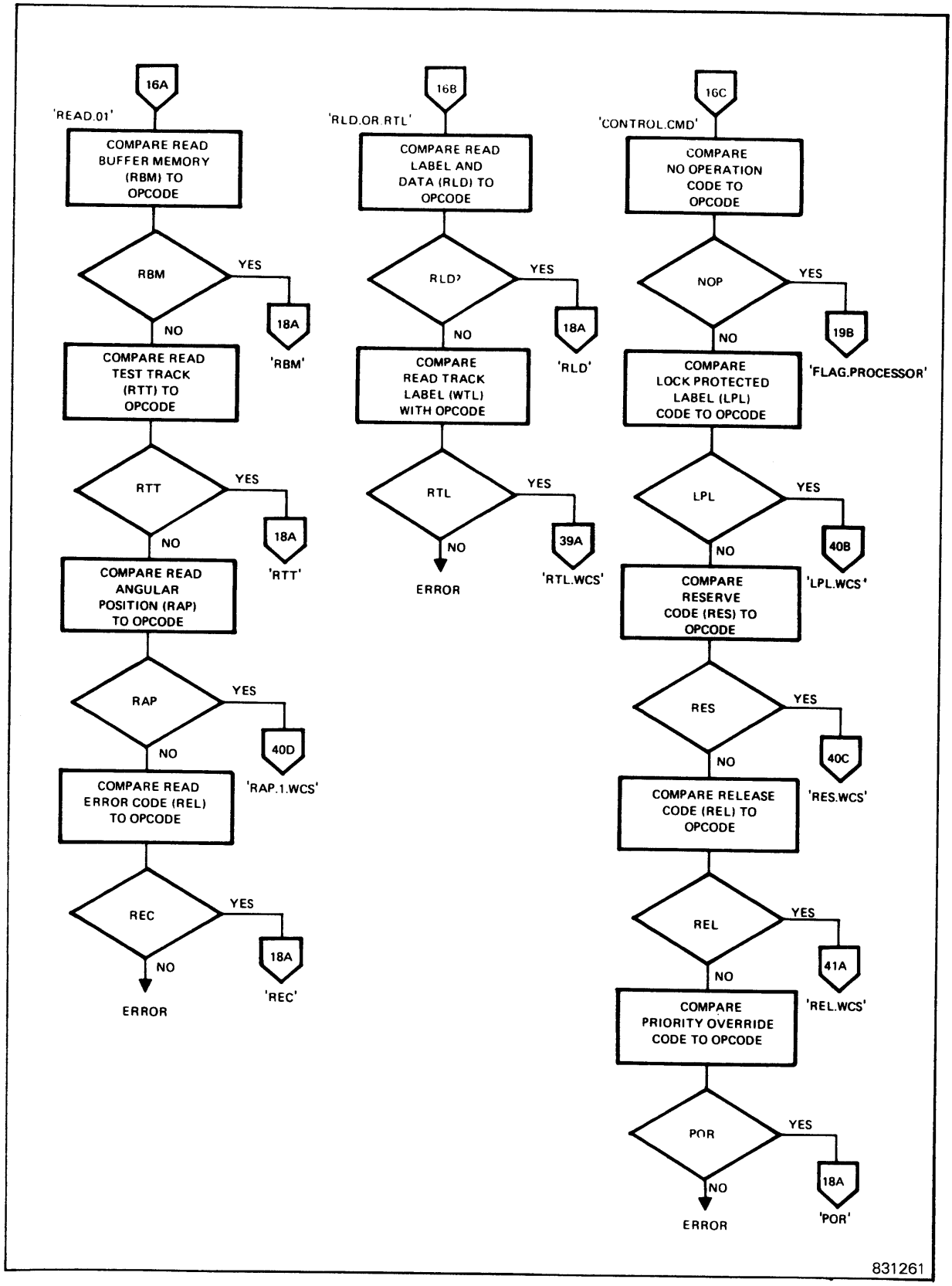


Figure F-2. Op Code Decode (Sheet 16 of 41)

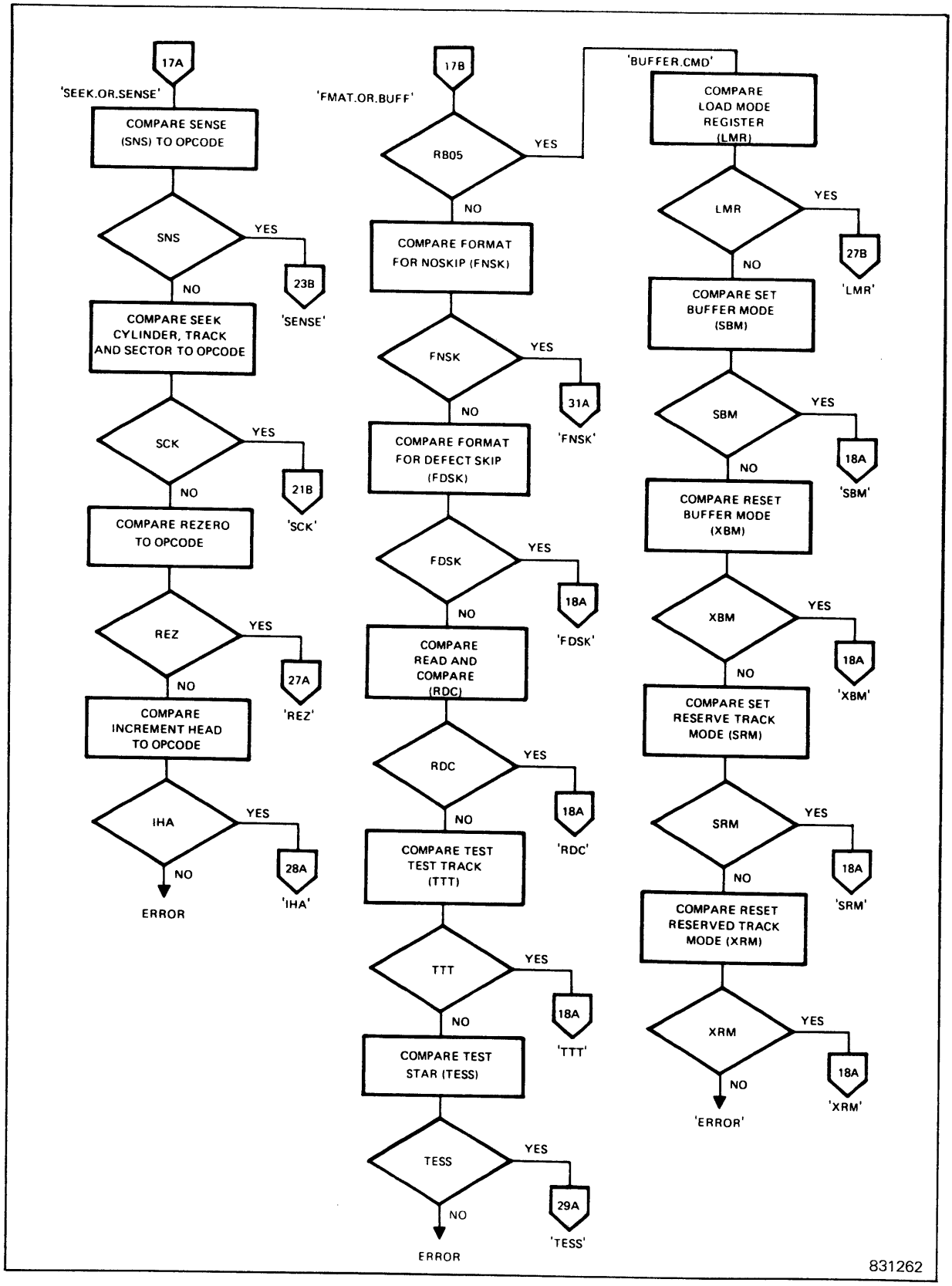


Figure F-2. Op Code Decode (Sheet 17 of 41)

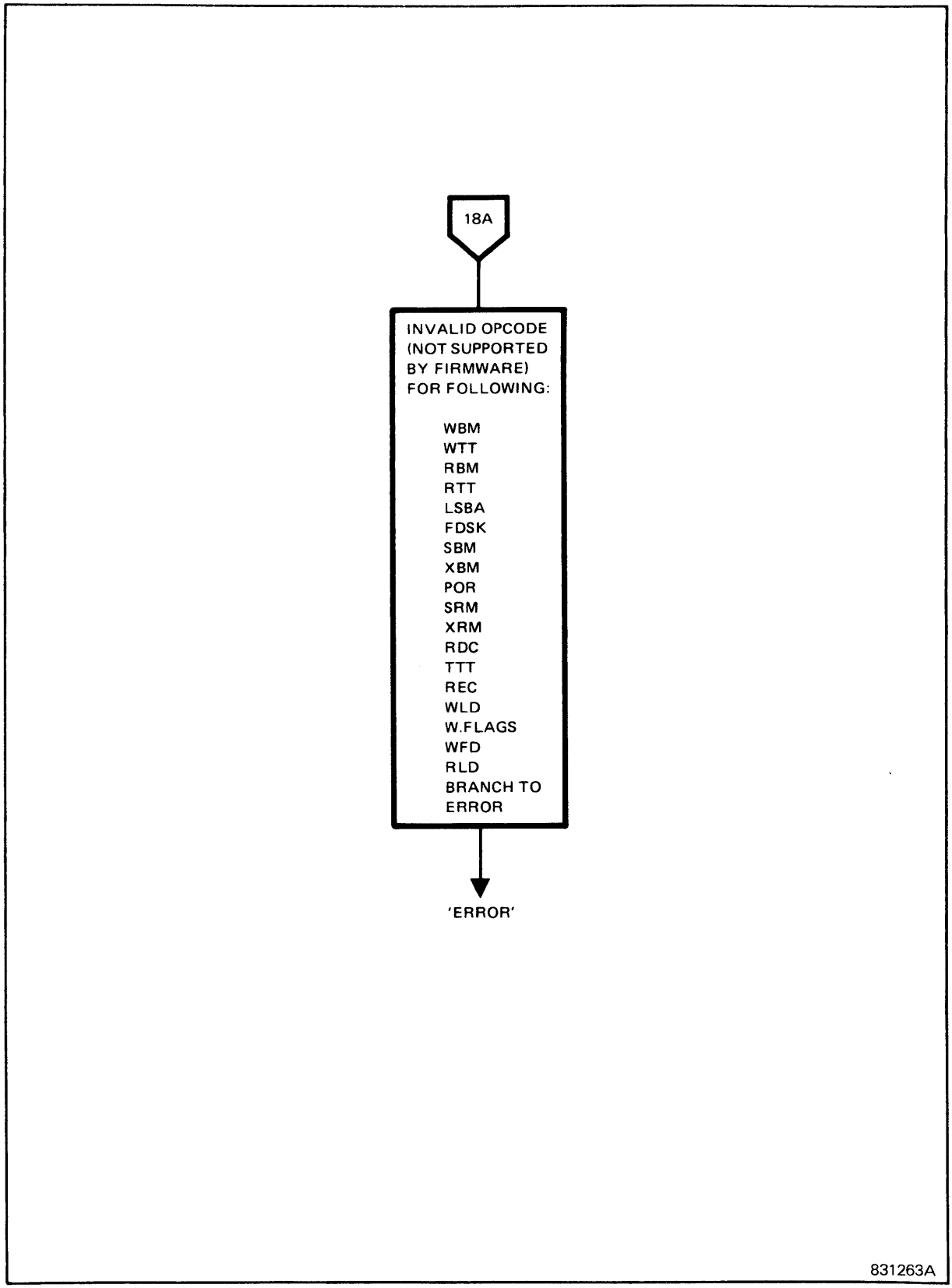


Figure F-2. Op Code Decode (Sheet 18 of 41)

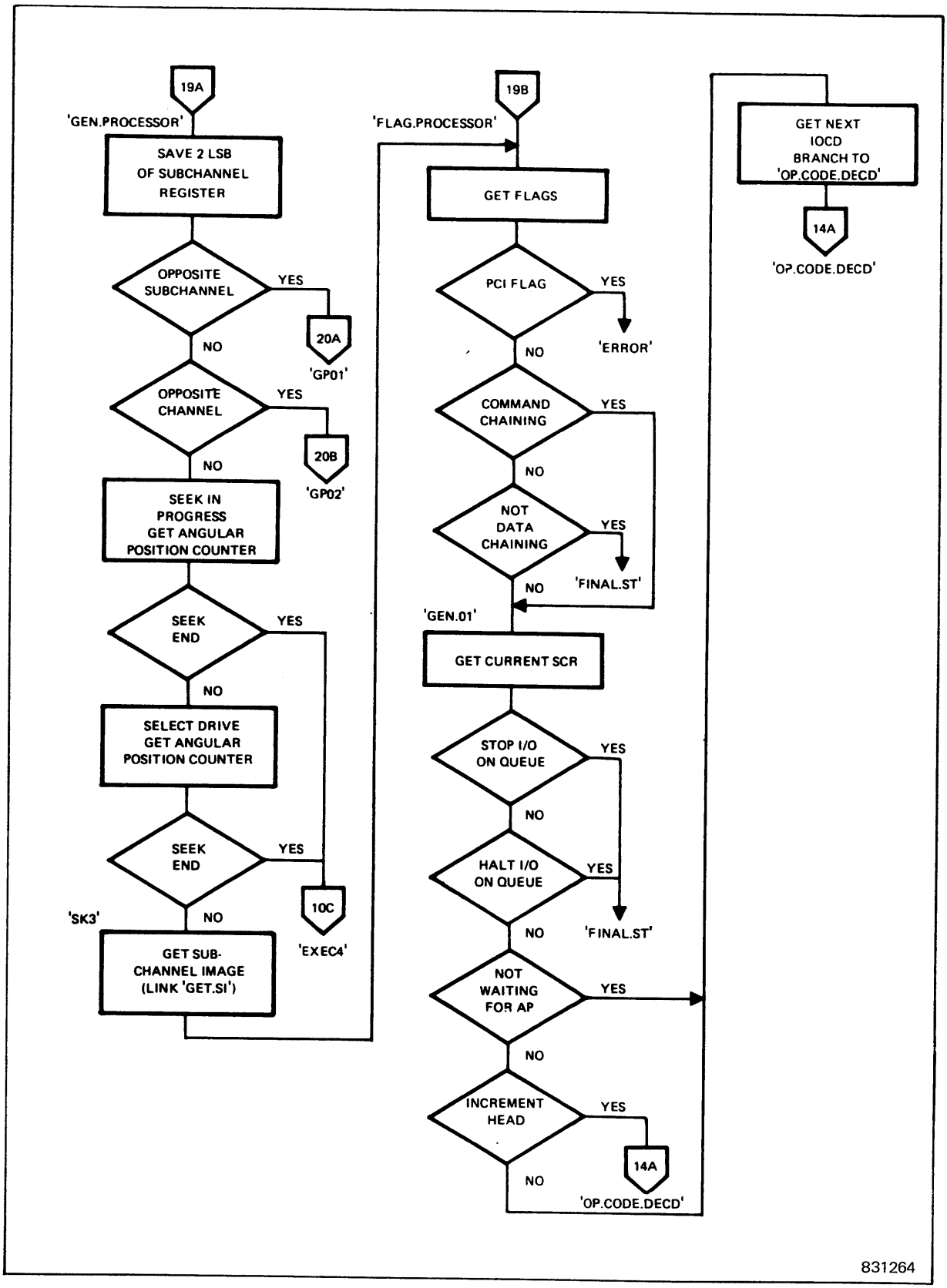


Figure F-2. General Processor; Flag Processor (Sheet 19 of 41)

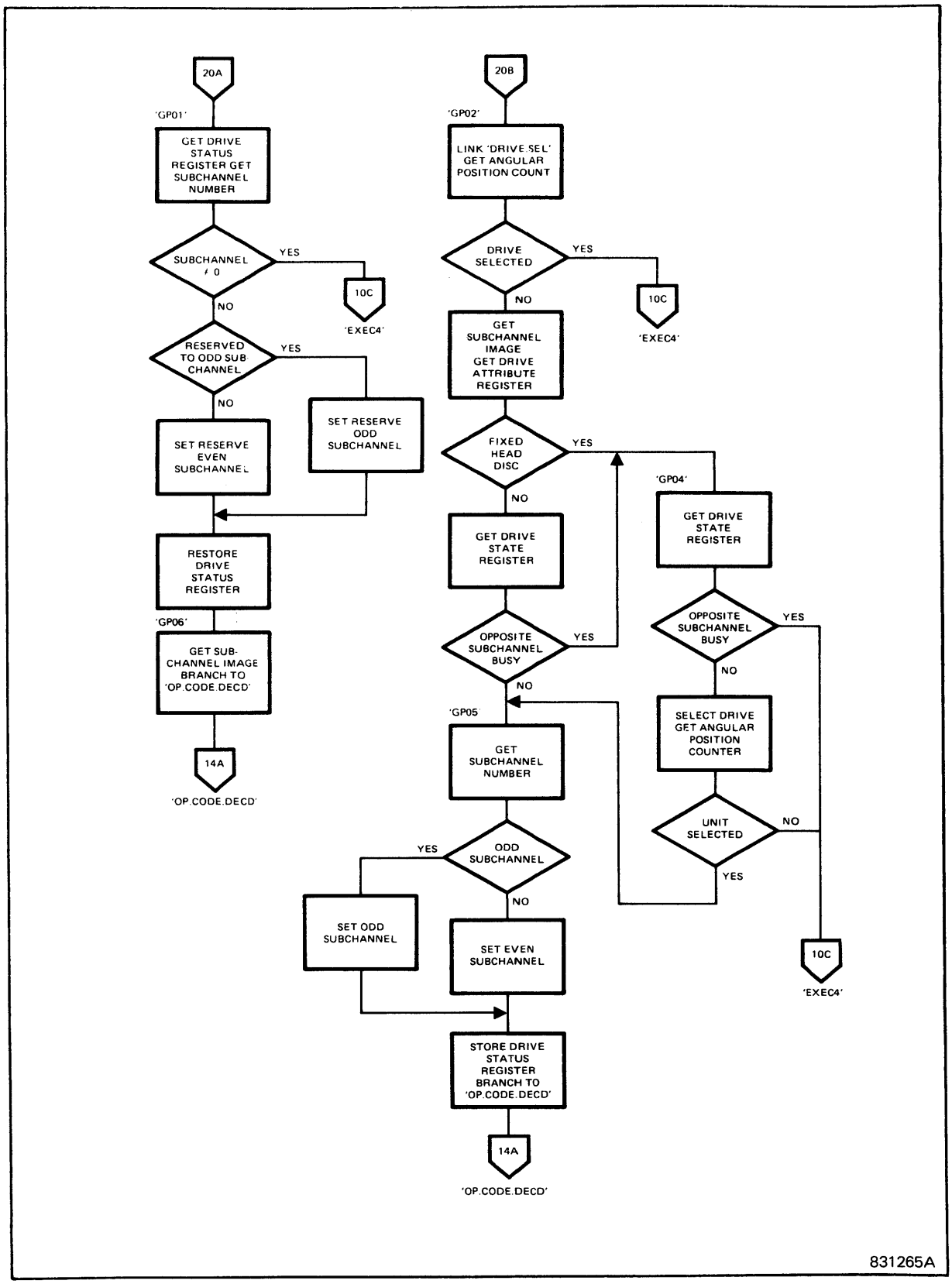


Figure F-2. General Processor (Sheet 20 of 41)

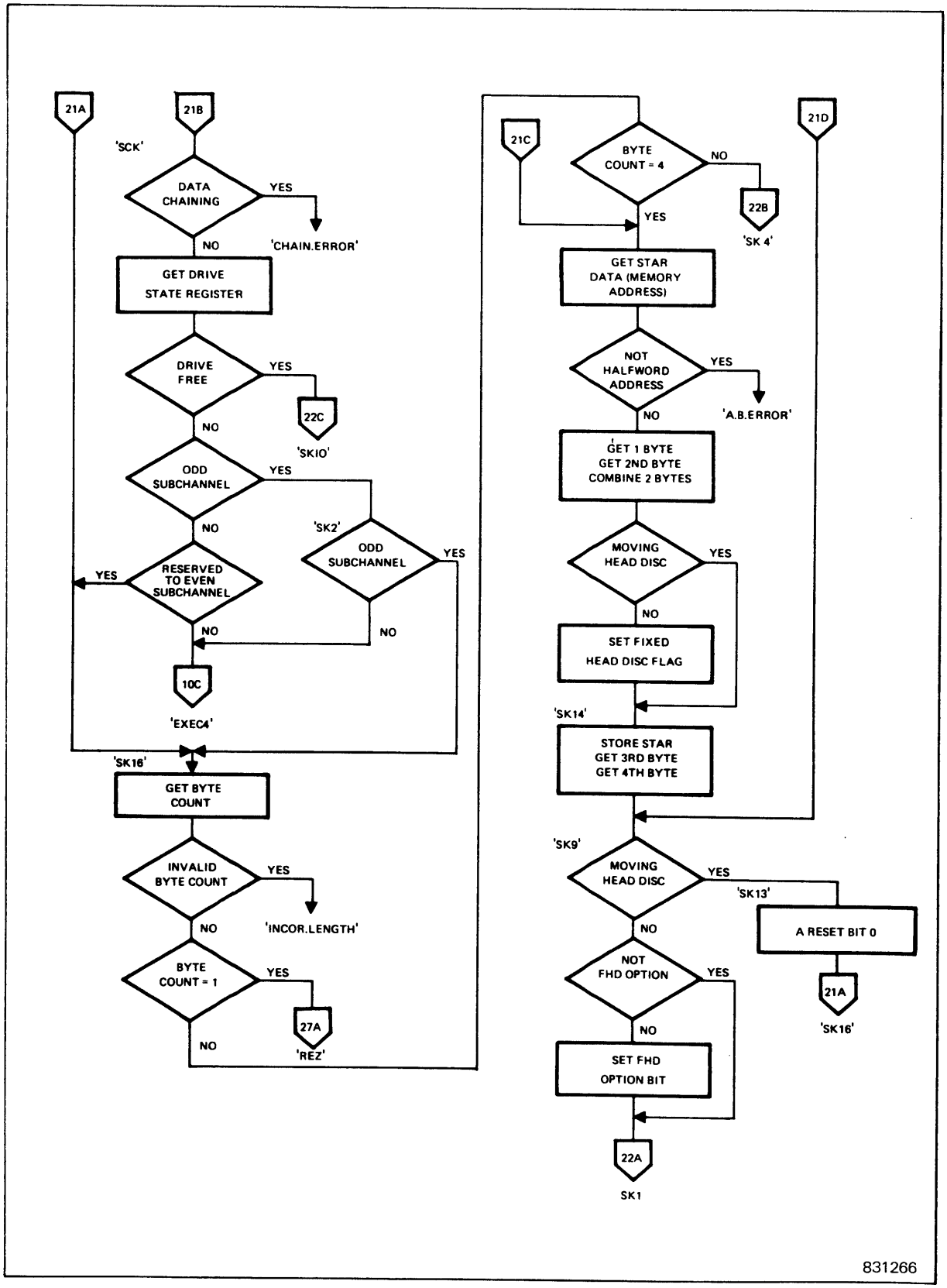


Figure F-2. Seek Cylinder, Head, and Sector (Sheet 21 of 41)

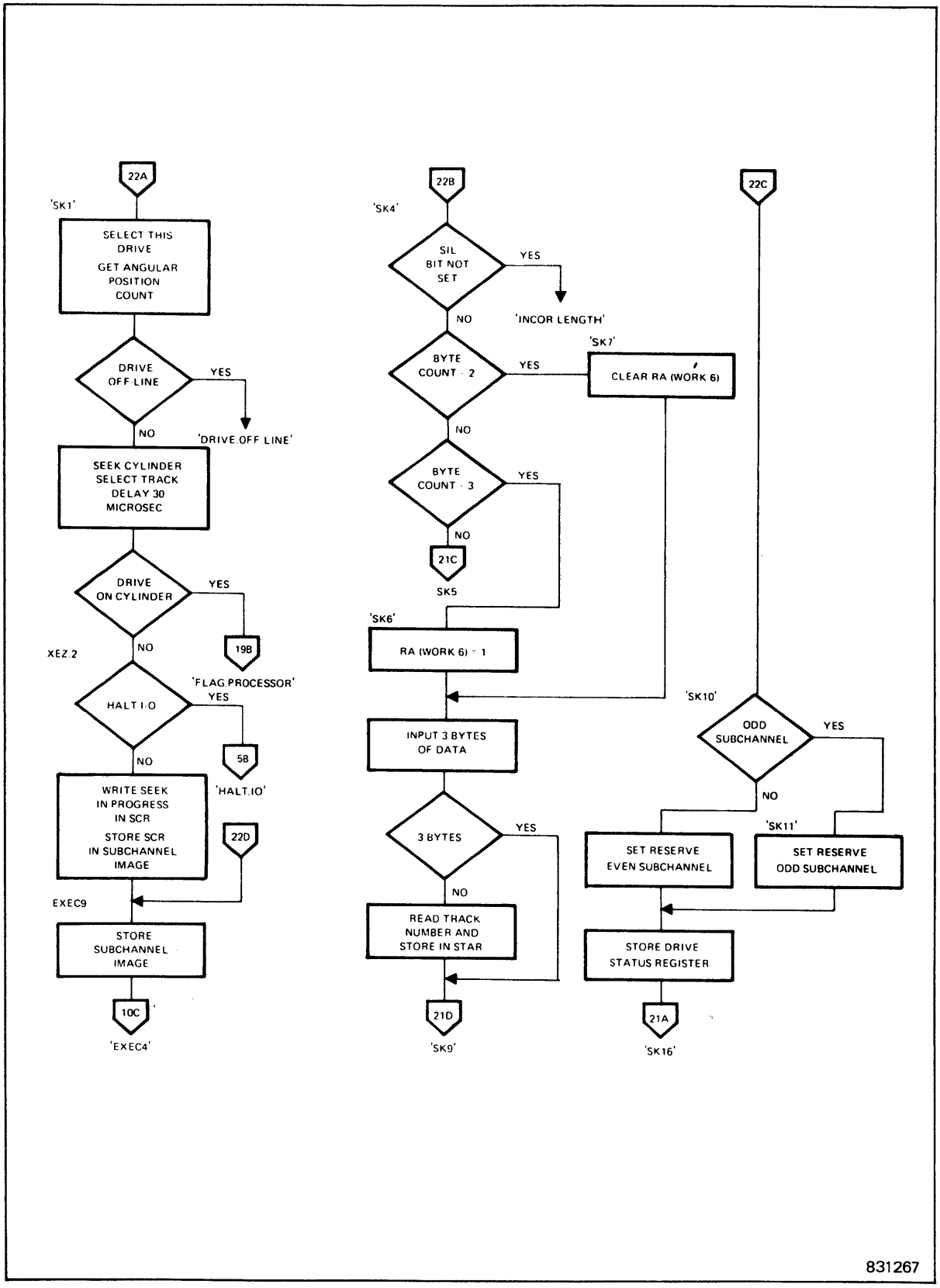


Figure F-2. Seek Cylinder, Head, and Sector (Sheet 22 of 41)

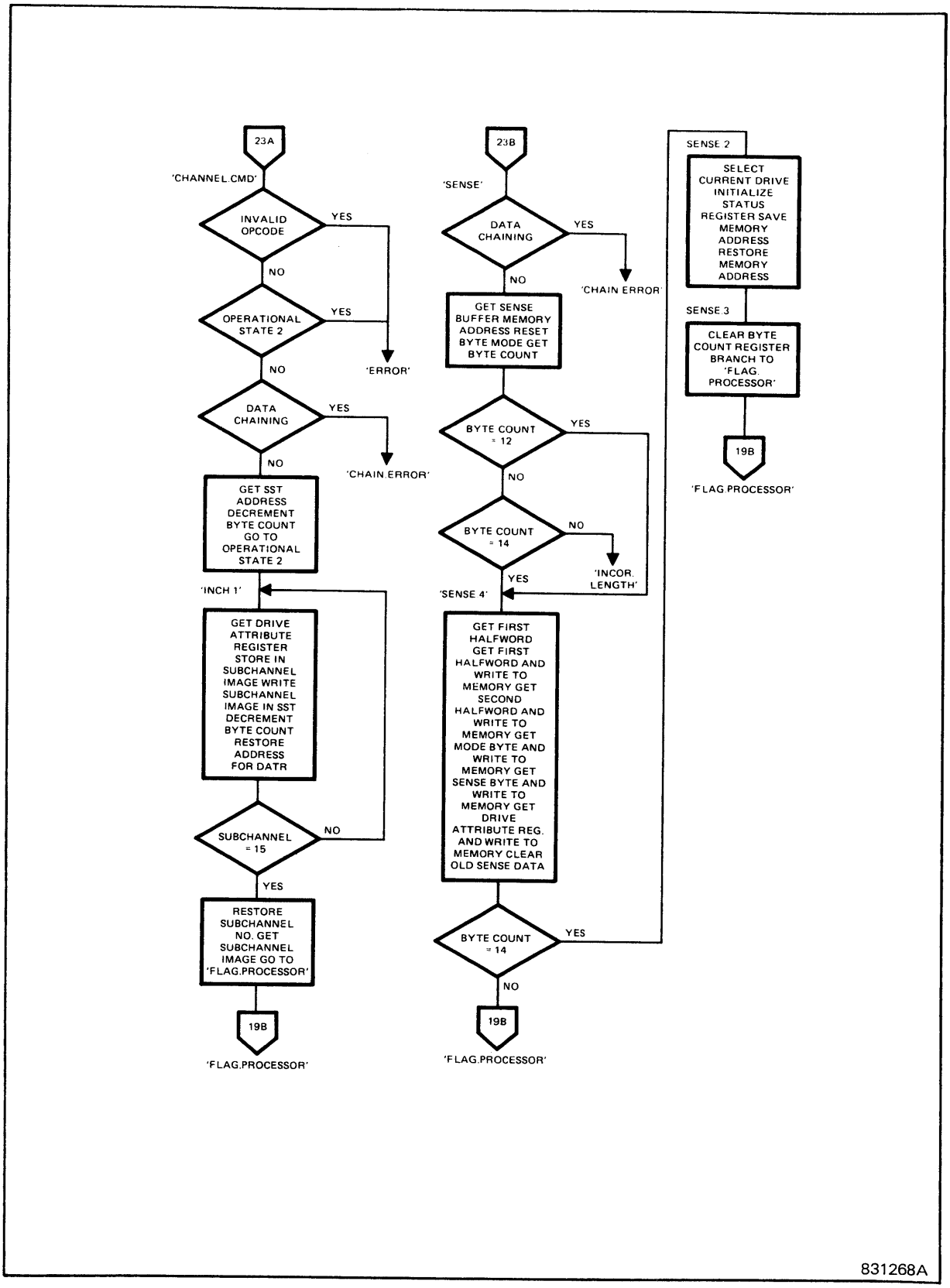


Figure F-2. Channel Command; Sense (Sheet 23 of 41)

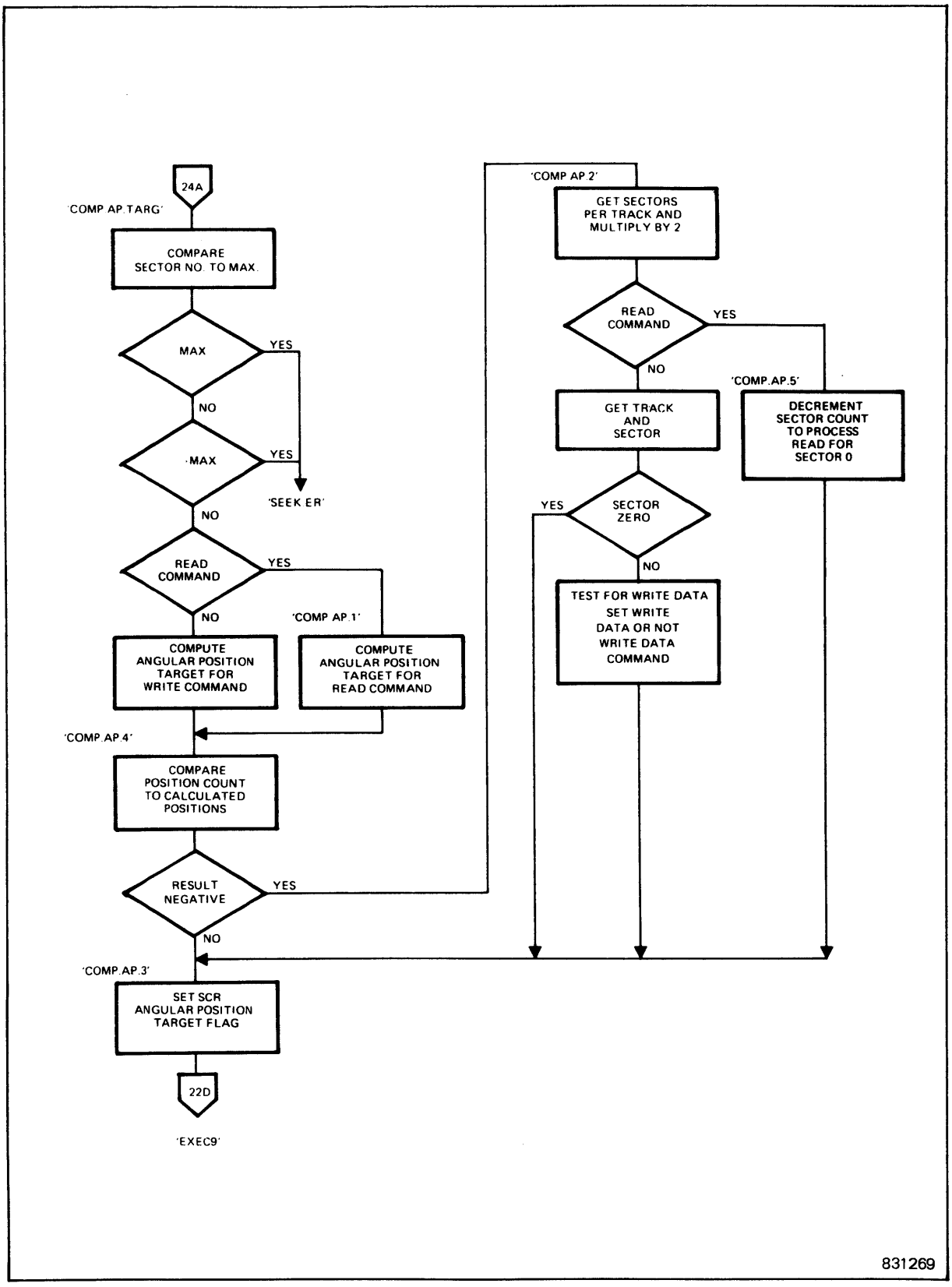


Figure F-2. Compare Angular Position Target (Sheet 24 of 41)

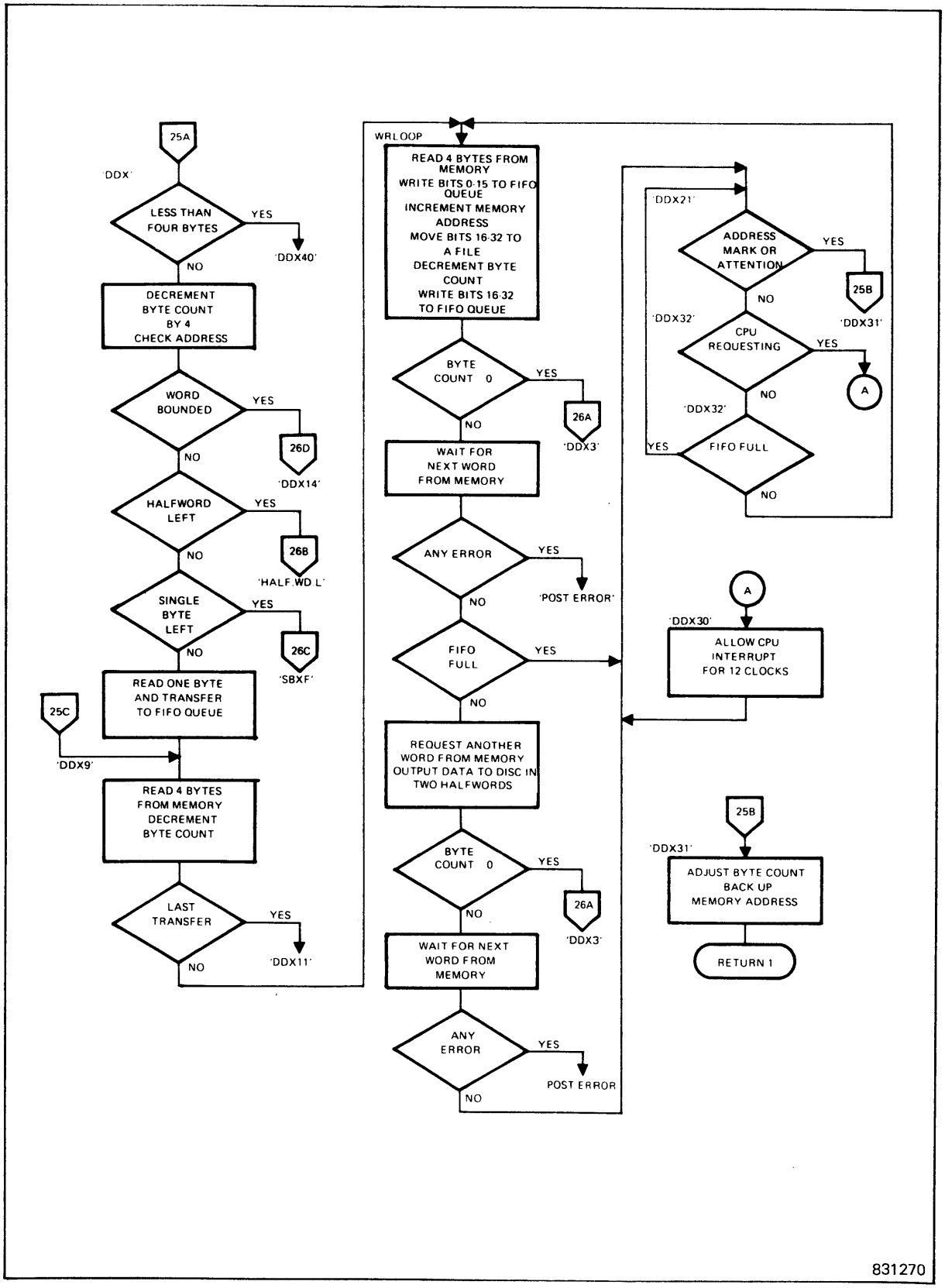


Figure F-2. Disc Data Transfer (Sheet 25 of 41)

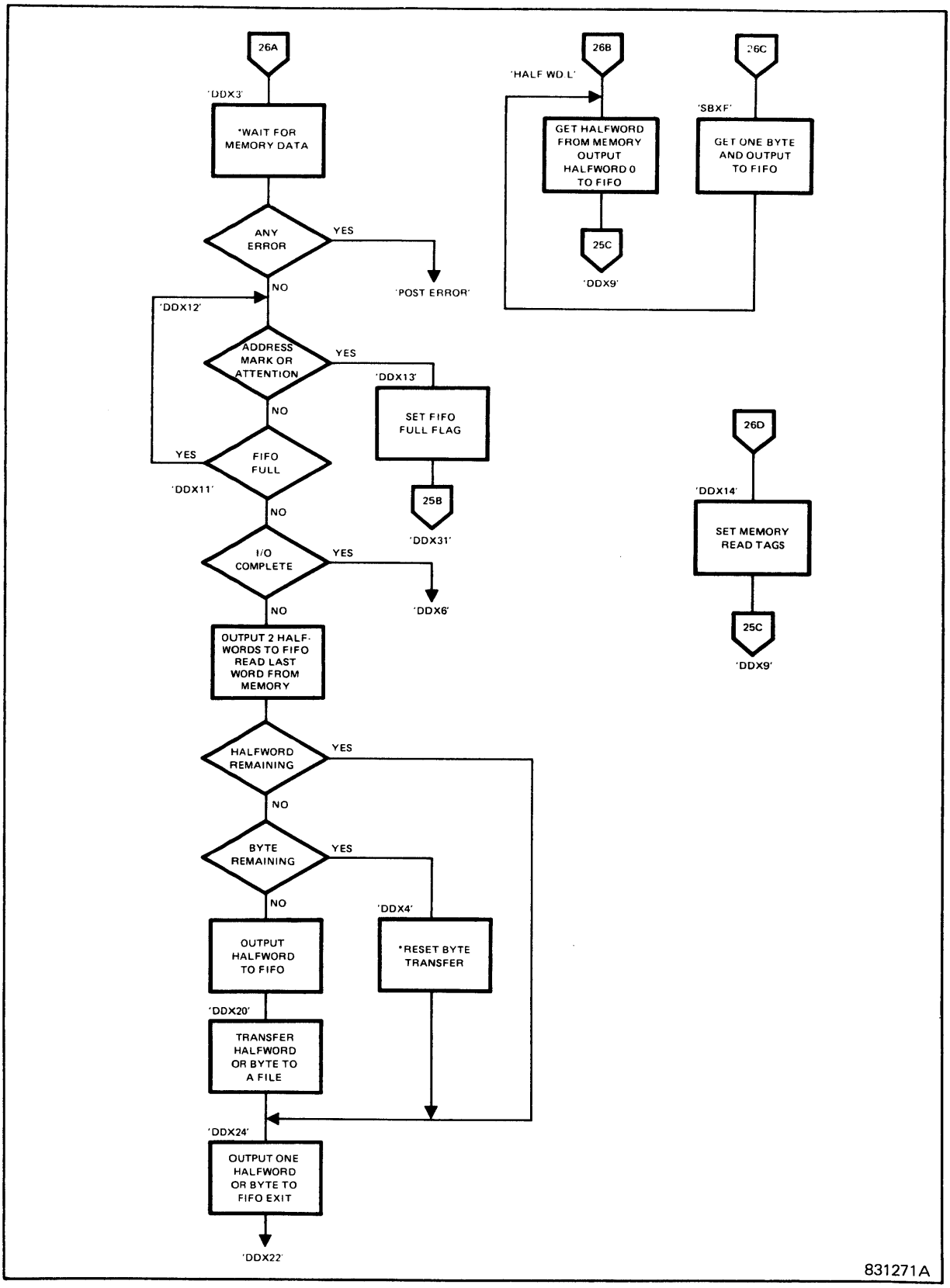


Figure F-2. Disc Data Transfer (Sheet 26 of 41)

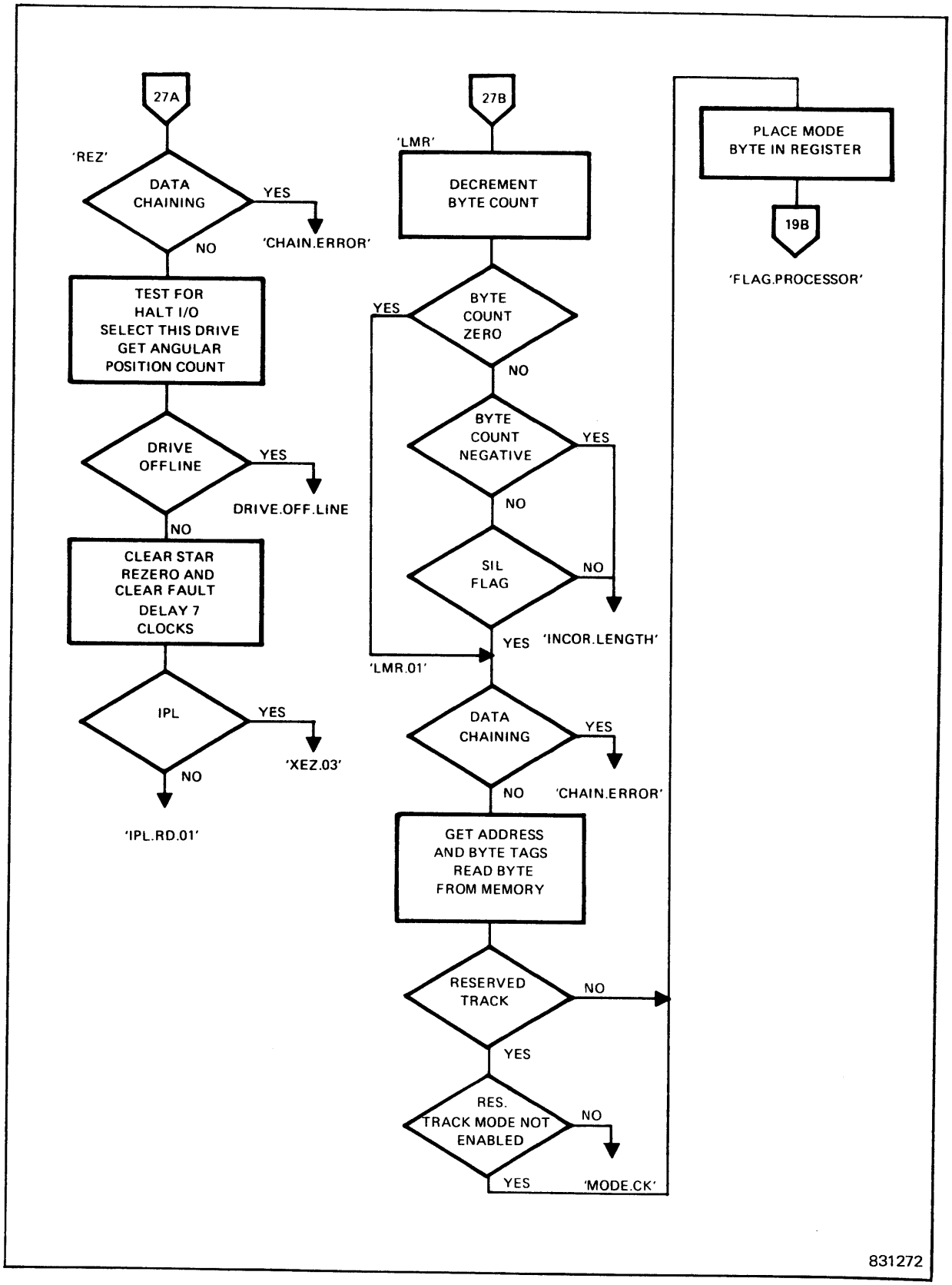


Figure F-2. Rezero Head; Load Mode Register (Sheet 27 of 41)

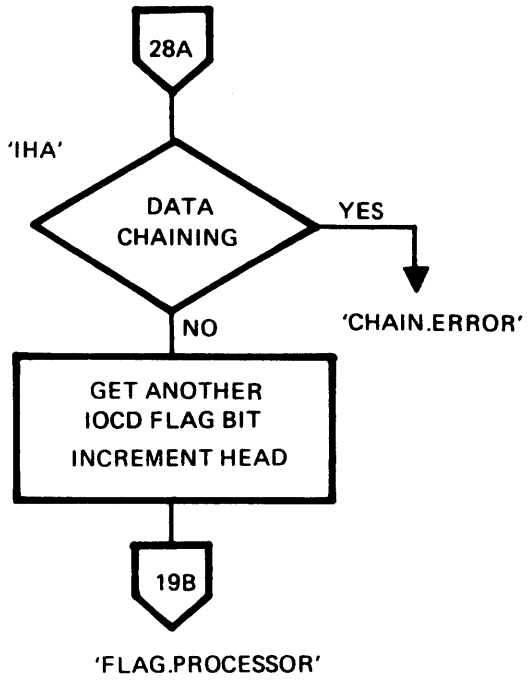


Figure F-2. Increment Head (Sheet 28 of 41)

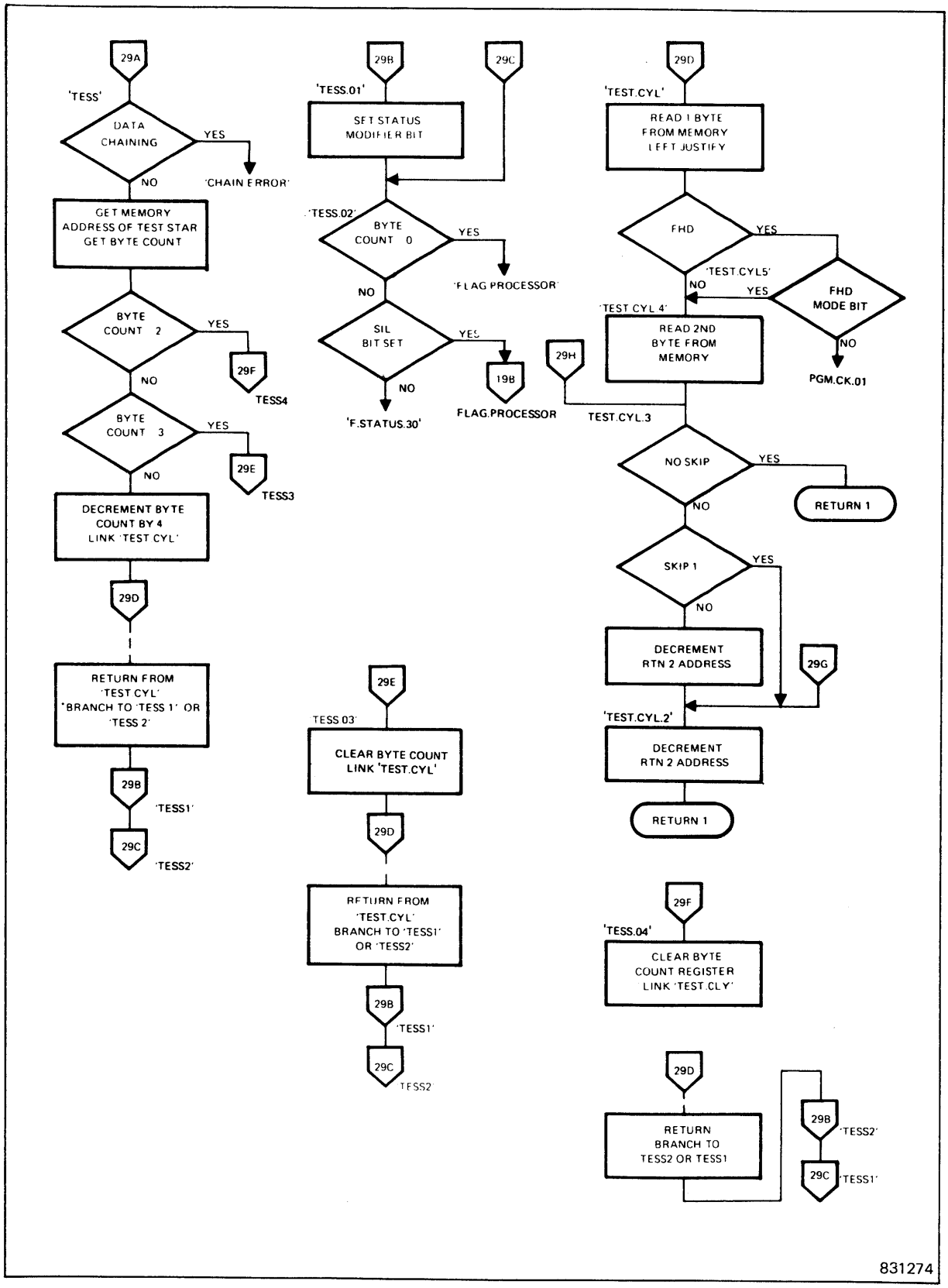


Figure F-2. Test Star (Sheet 29 of 41)

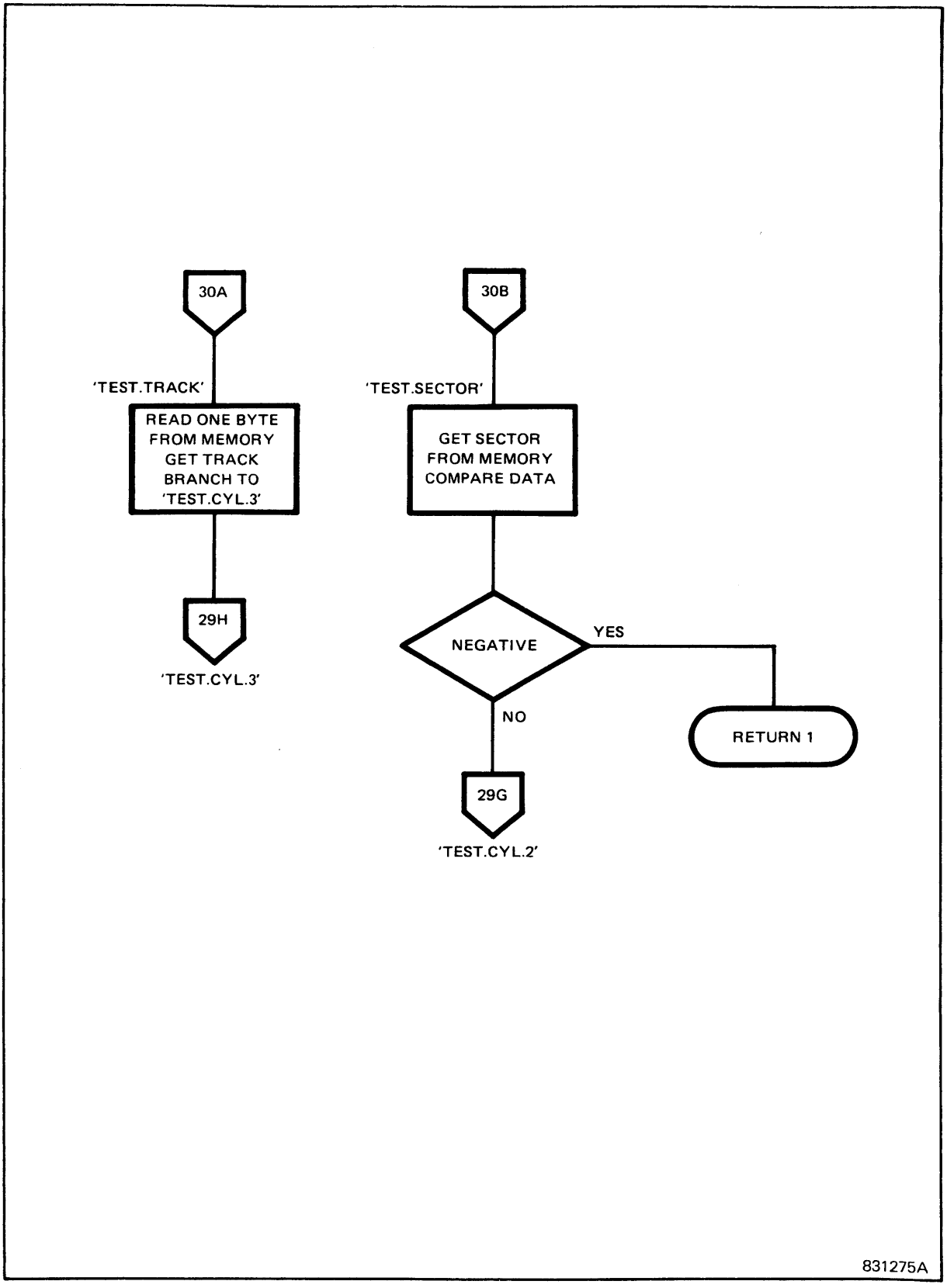


Figure F-2. Test Star (Sheet 30 of 41)

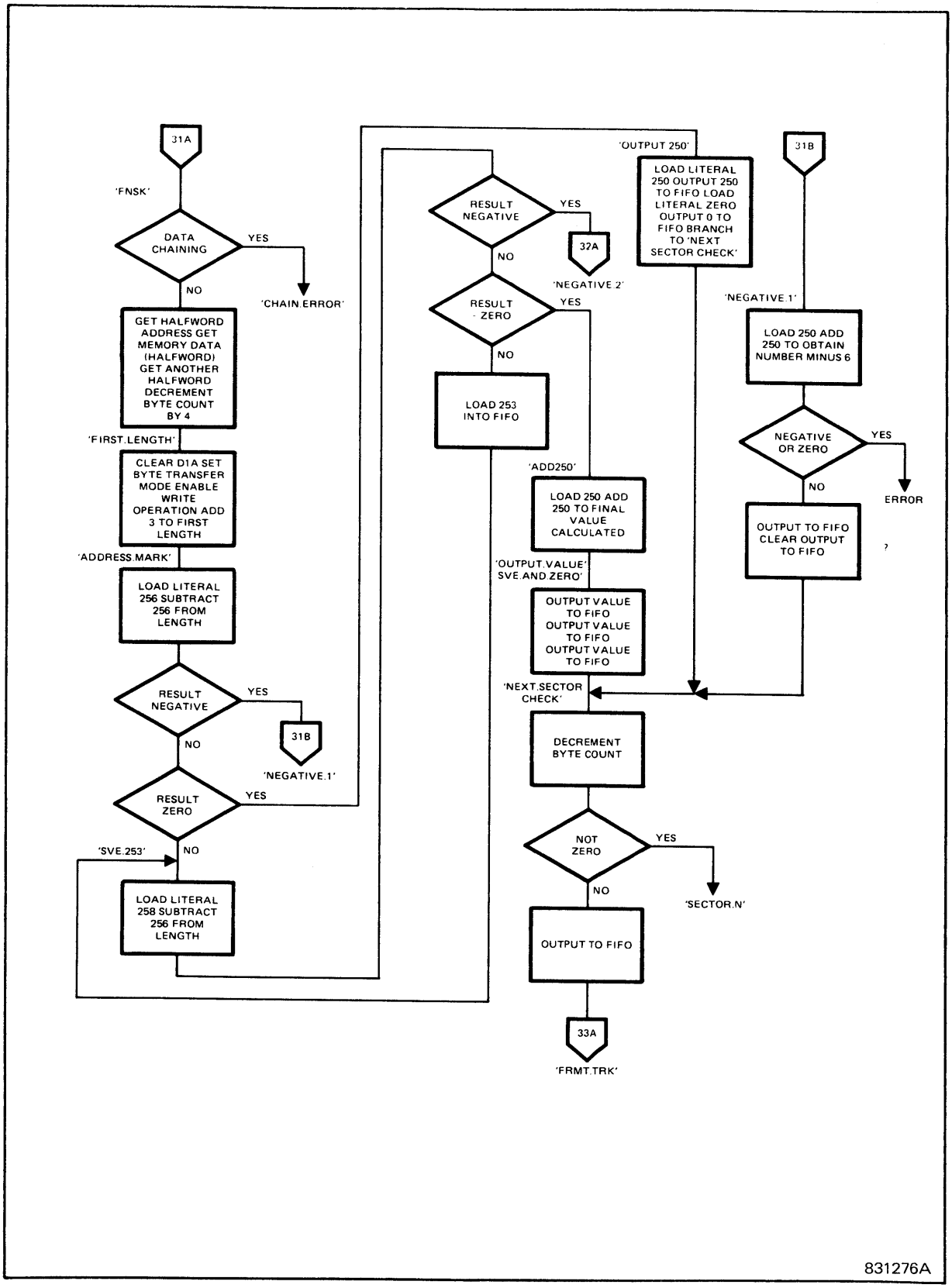


Figure F-2. Format for No Skip (Sheet 31 of 41)

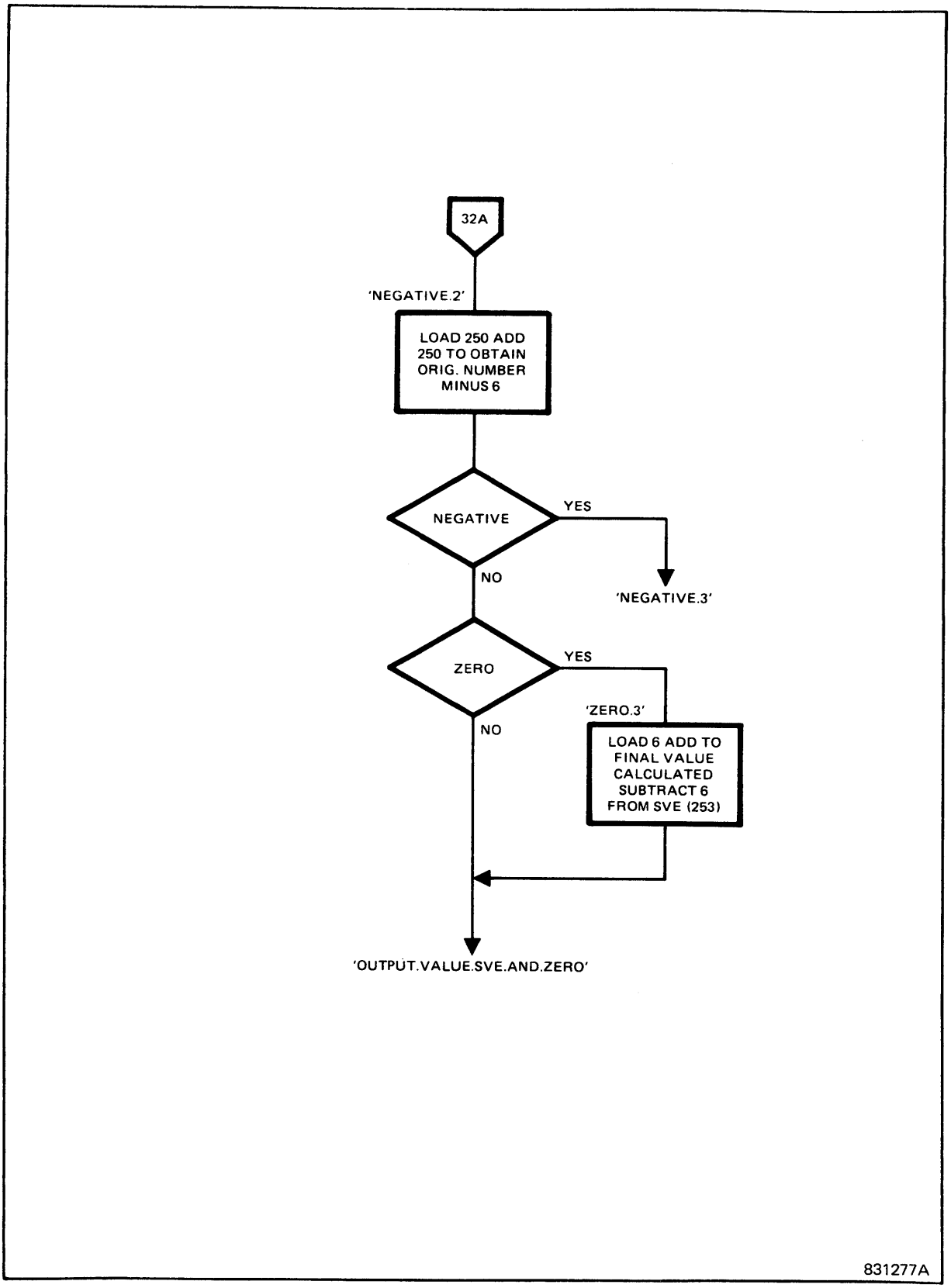


Figure F-2. Format for No Skip (Sheet 32 of 41)

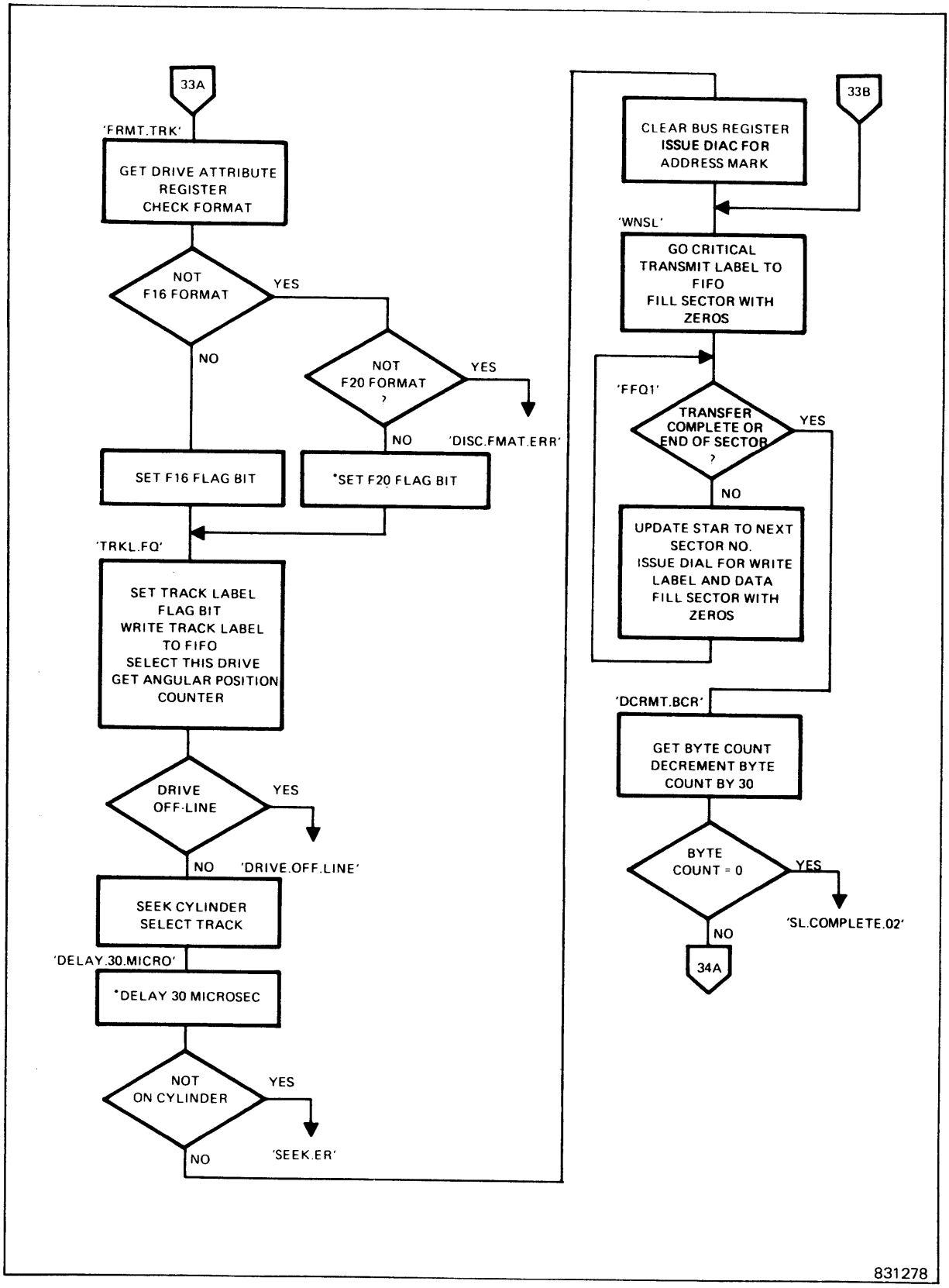


Figure F-2. Format Track (Sheet 33 of 41)

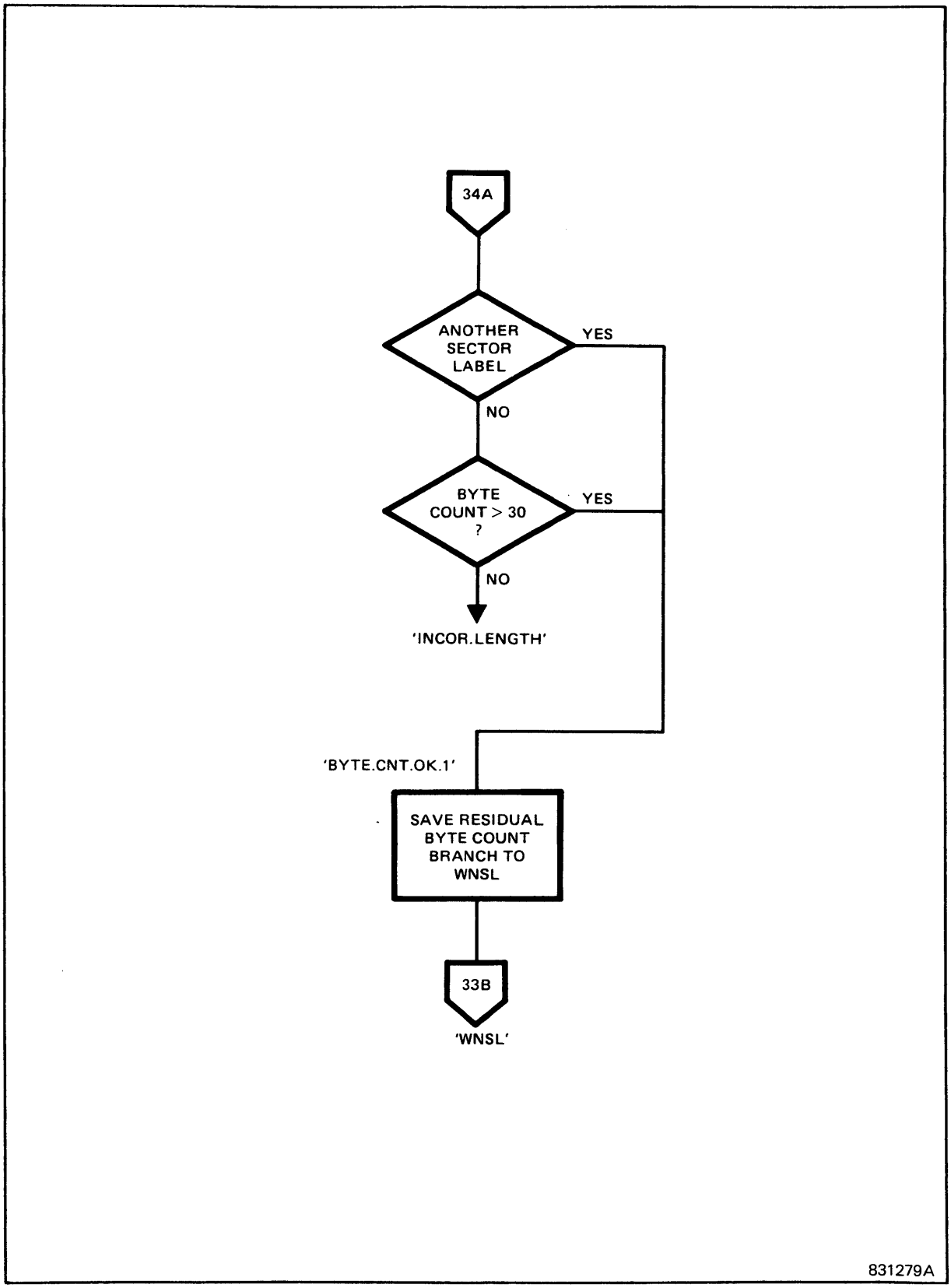


Figure F-2. Format Track (Sheet 34 of 41)

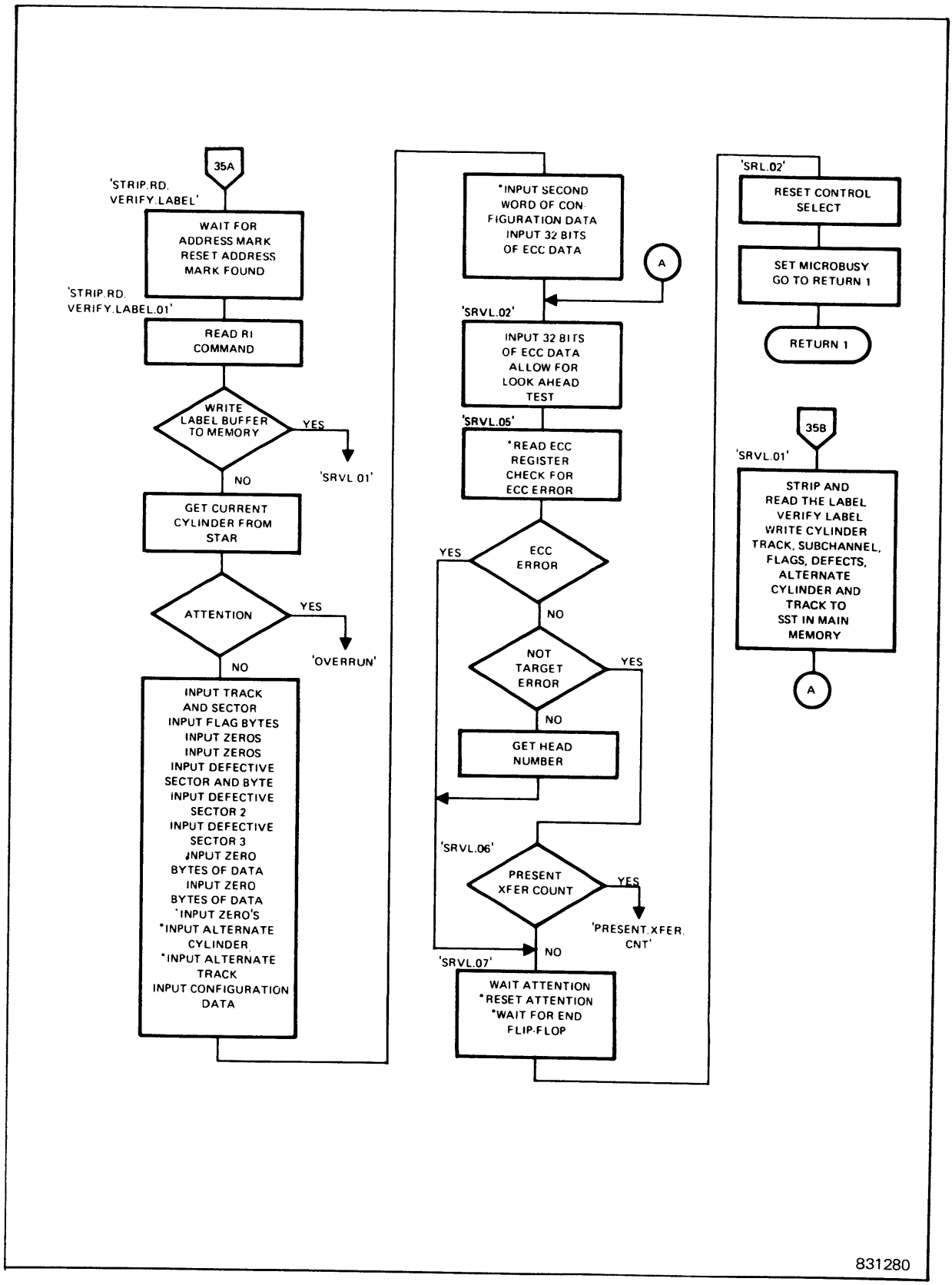


Figure F-2. Strip Read and Verify Label (Sheet 35 of 41)

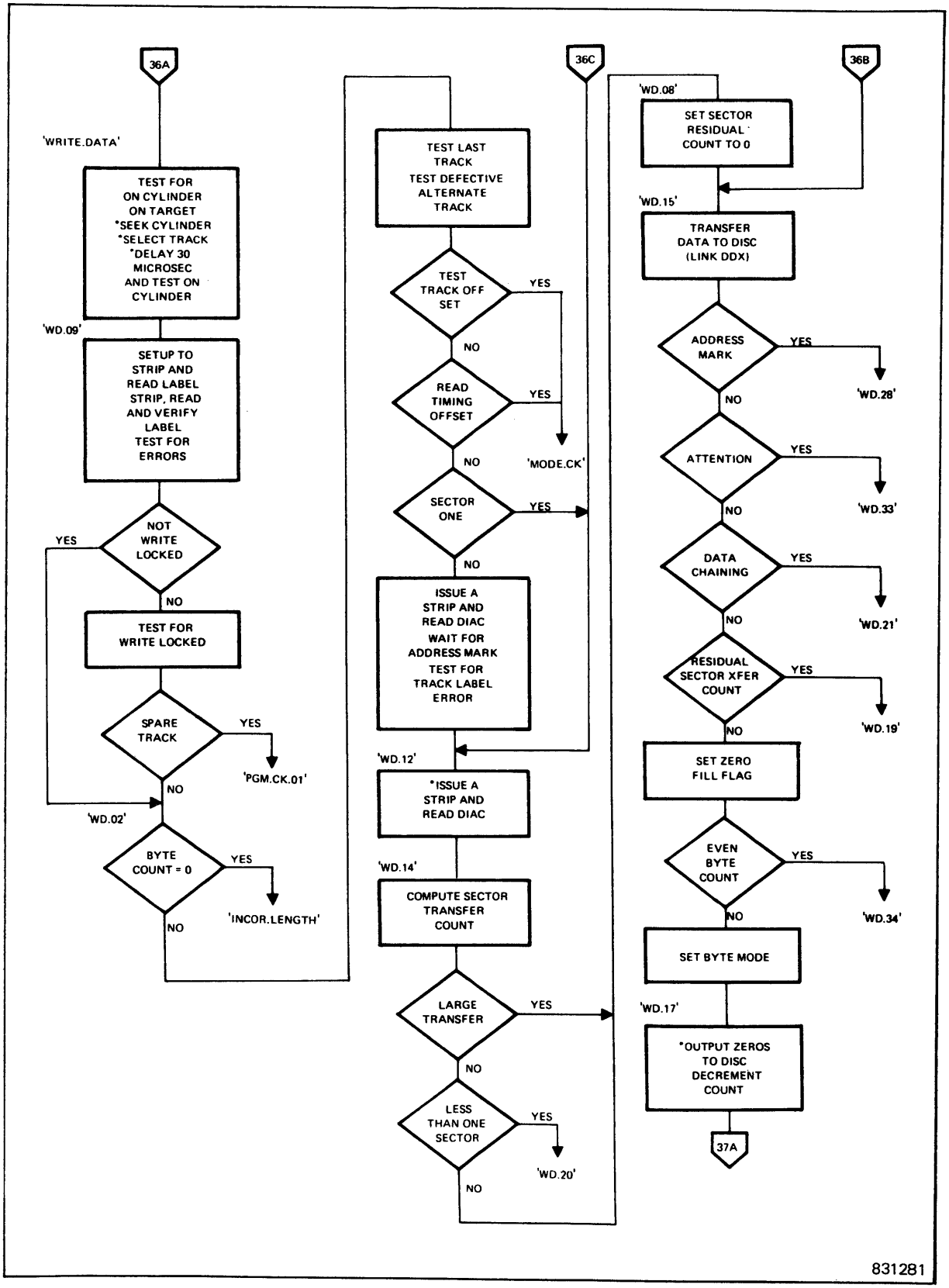


Figure F-2. Write Data (Sheet 36 of 41)

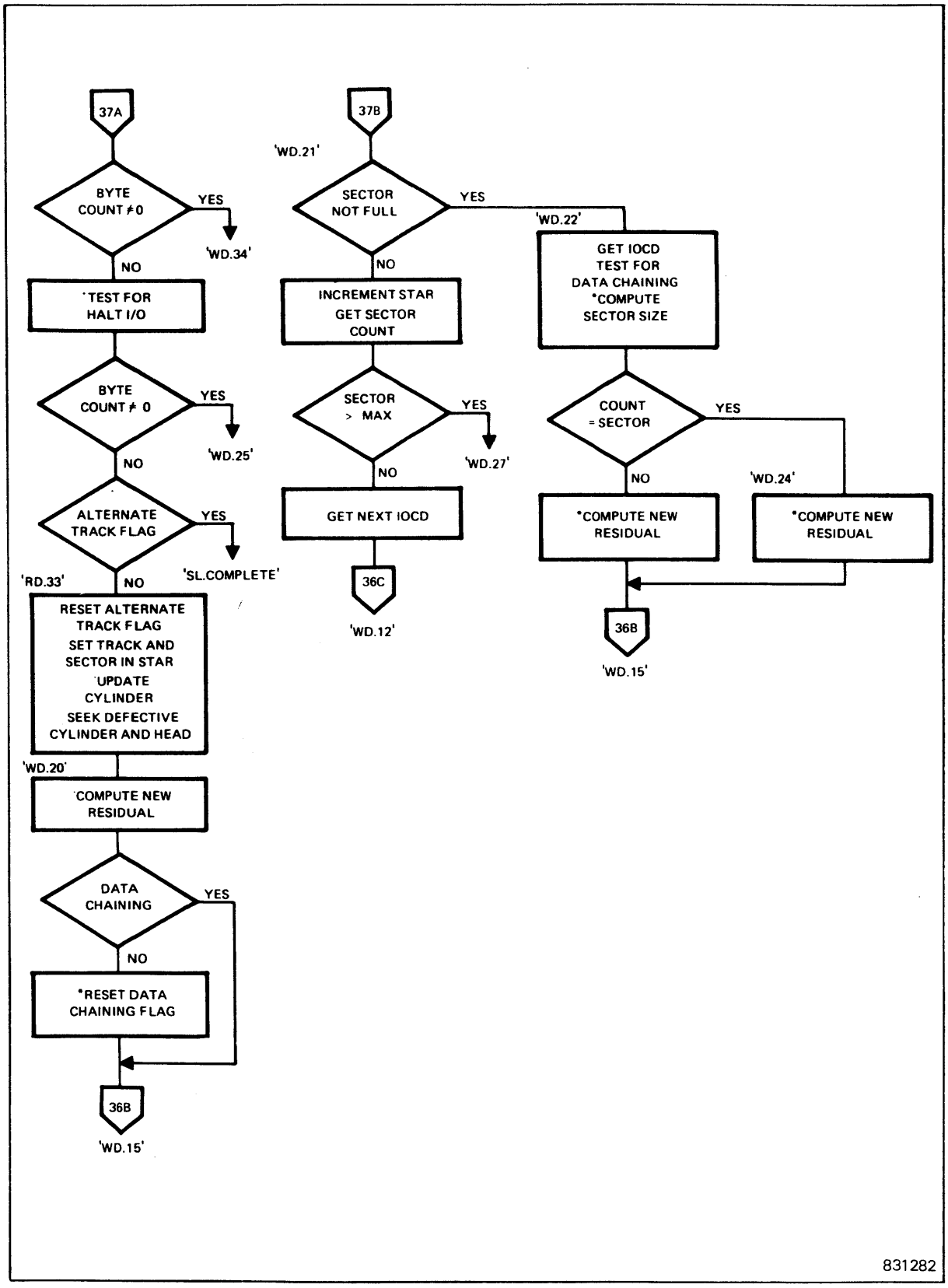


Figure F-2. Write Data (Sheet 37 of 41)

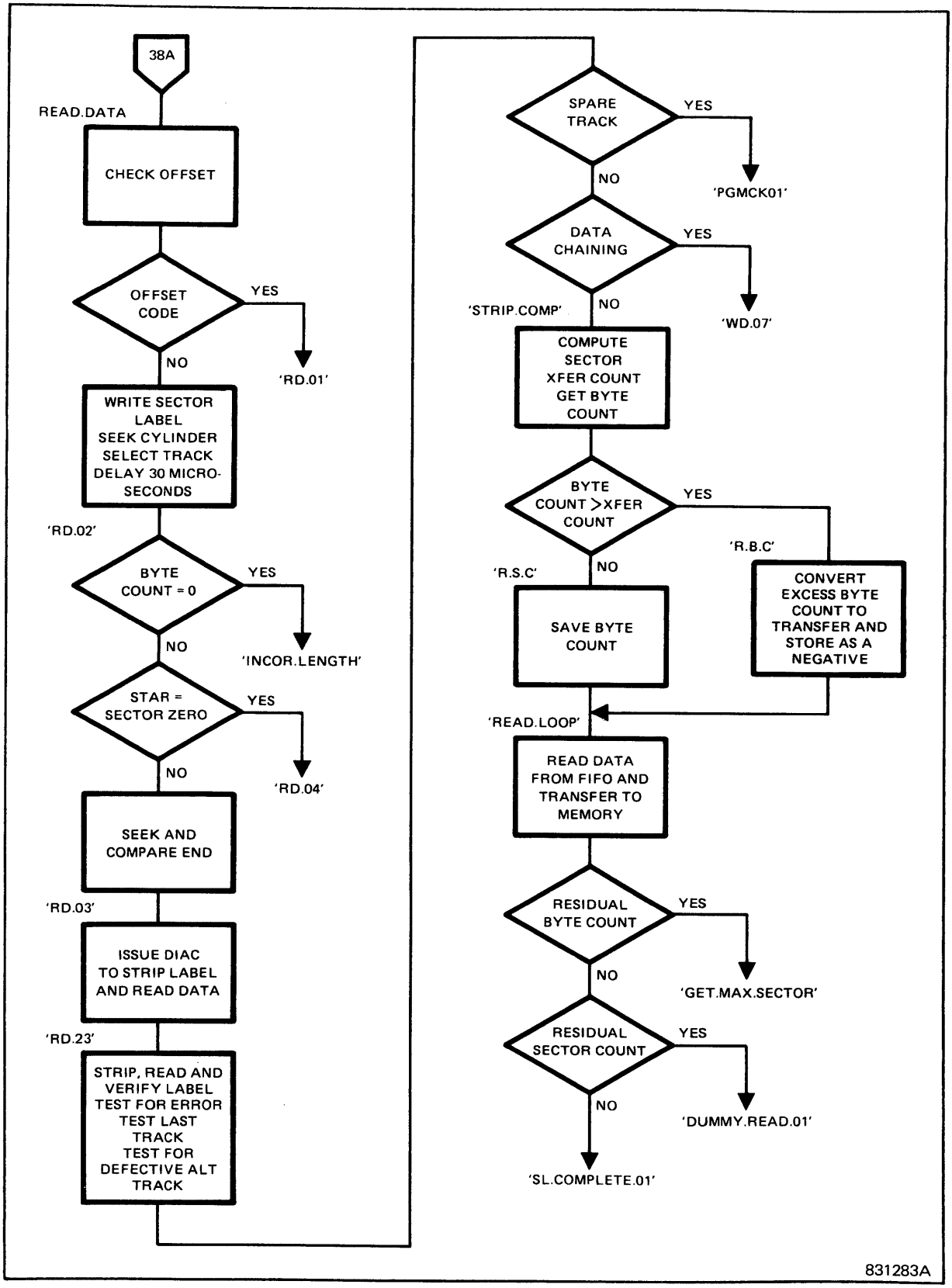


Figure F-2. Read Data (Sheet 38 of 41)

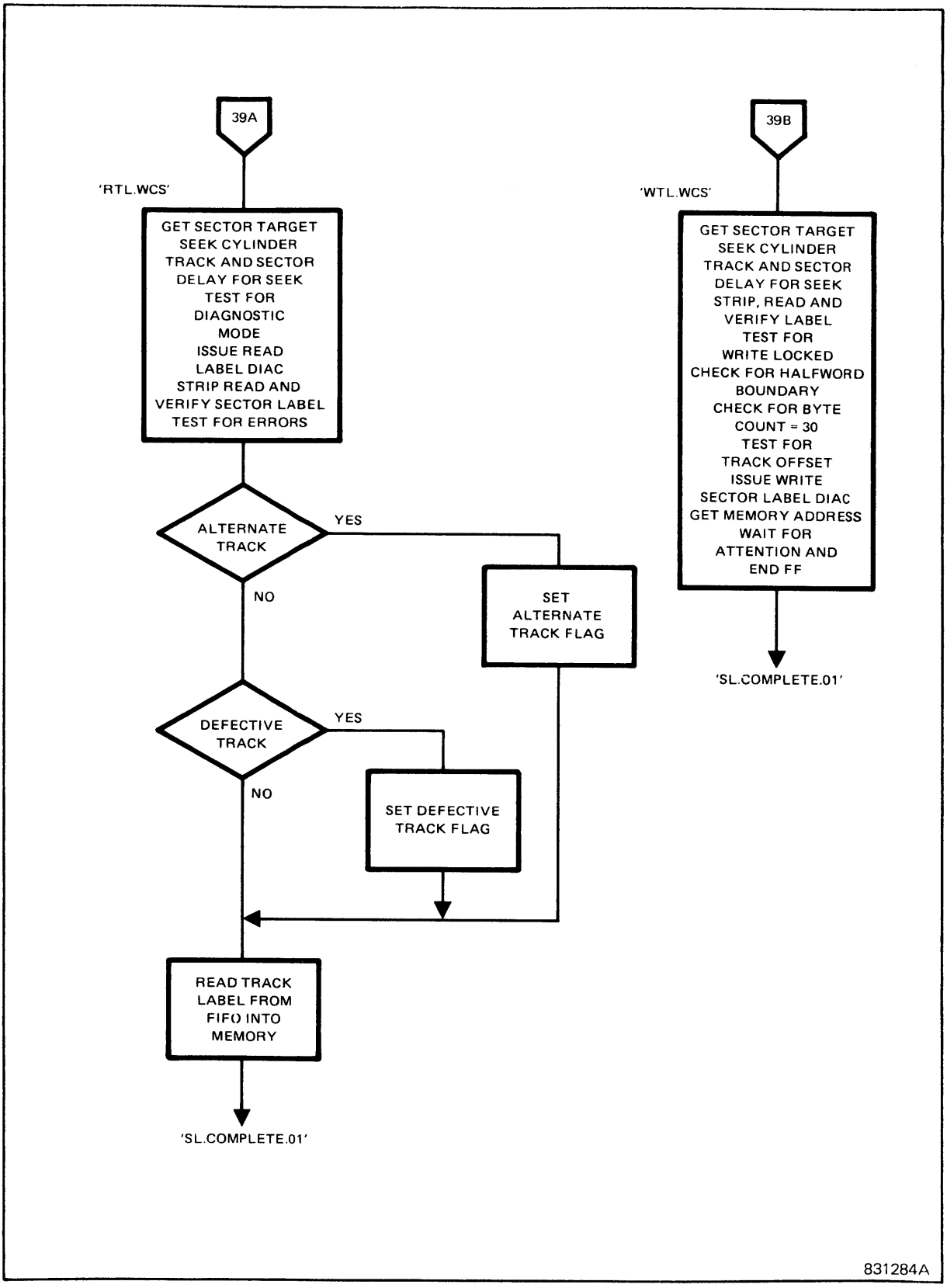


Figure F-2. Read Track Label; Write Track Label (Sheet 39 of 41)

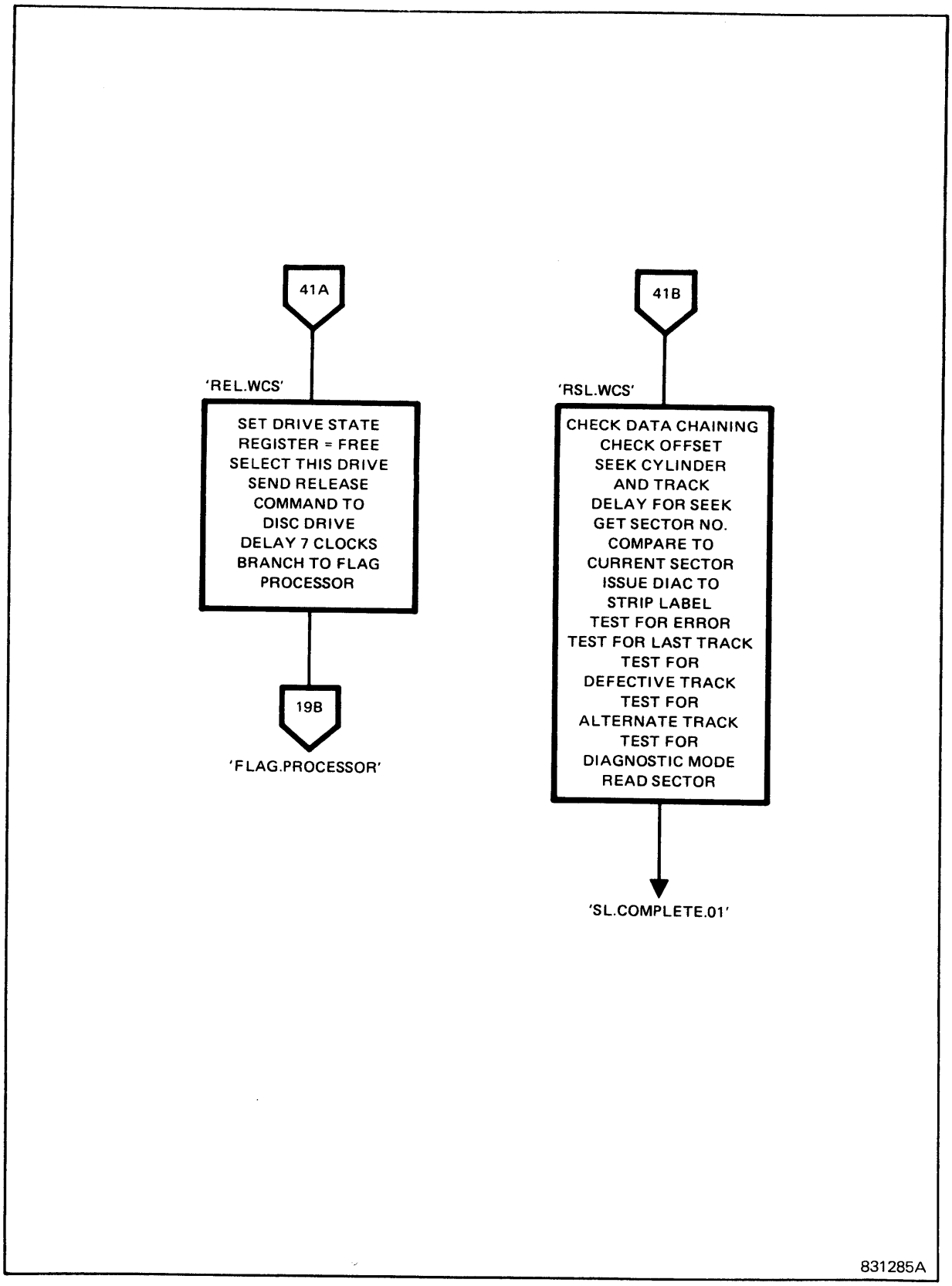


Figure F-2. Write Sector Label; Lock Protected Labels; Reserve (Sheet 40 of 41)

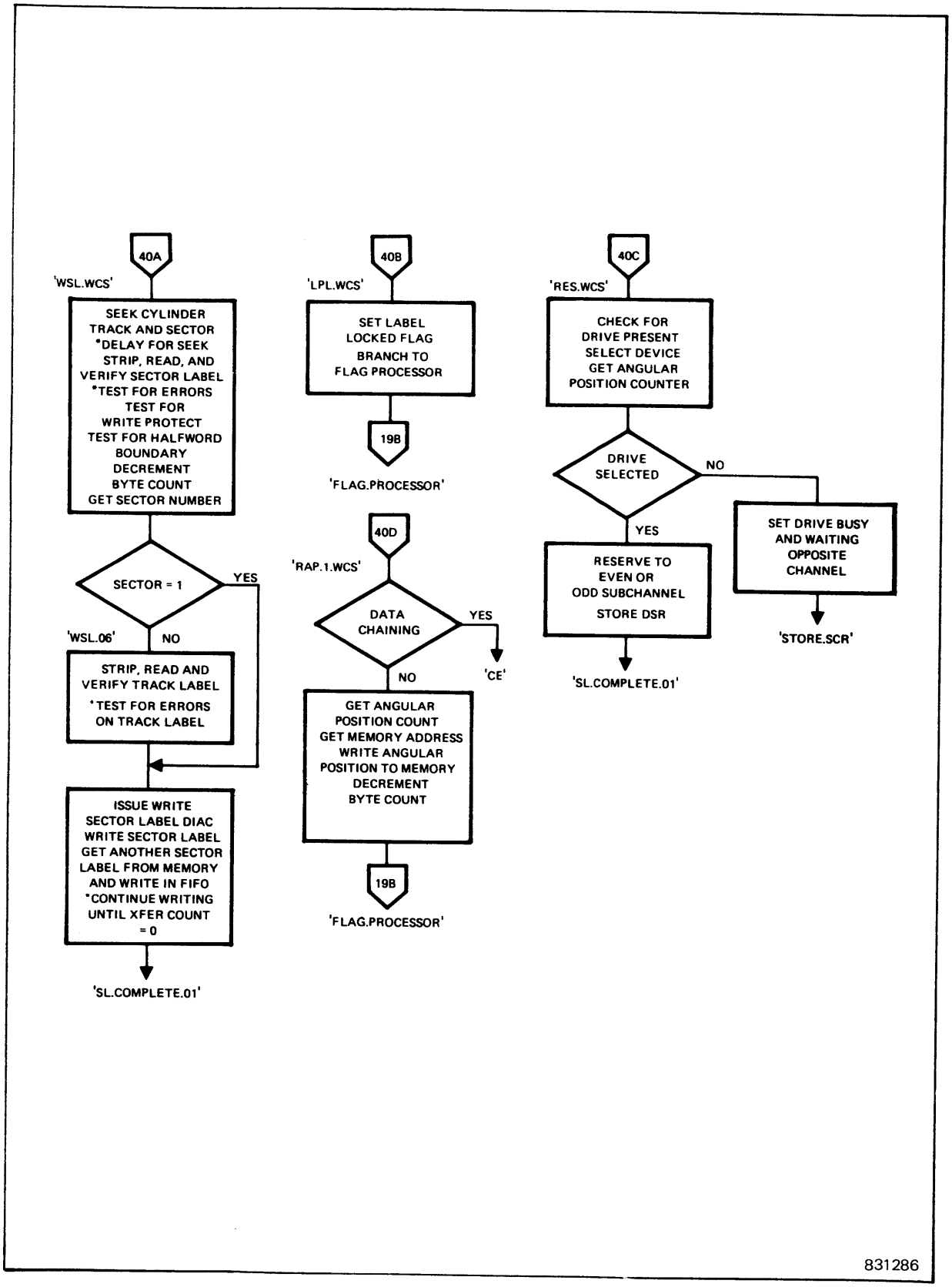


Figure F-2. Release; Read Sector Label (Sheet 41 of 41)

APPENDIX G

GLOSSARY OF GENERAL DISC TERMS

Address Mark	A zone of erasure on the disc, approximately three bytes in length. The address mark is used by disc hardware to find data on the disc and to trigger internal circuitry.
Certify Disc	A procedure used to format a track, find media flaws, and possibly, reformat the track to avoid media flaws. This procedure may include defect skipping, alternate track addressing, and radially displacing labels to avoid ECC errors in labels.
Cylinder	A cylinder of data that refers to all of the sectors of data for all of the heads without moving the heads.
Defect Skip	A means by which data can be written to the disc and recorded in a segmented manner. There can be only one skip zone in a single sector.
False Address Mark	A media defect that looks like an address mark. If there is a false address mark, then that track cannot be used.
Format Track	A command issued to the disc processor that first causes an address mark to be written to the disc and then causes the sector, track labels, and data to be written to that track. This procedure must be done to every new disc.
Head	A reference to the physical read/write heads on a disc drive. One head is capable of accessing one full track of information without being moved.
Sector	A reference to 1 of 16 or 20 sectors of data on a specified track.
STAR	A sector target address register. An internal register containing the cylinder, track, and sector that the drive is to seek to. STAR will contain any volume change/select information as required by specific disc drives.
Track	A track that refers to all of the sectors of data for a given head on a given cylinder.

INDEX