MEMOREX

3672
Storage Control Unit
Theory of Operations Manual

3672.21-00
3672.21-0001

Publication Number 3672.21-00
and Publication Bulletin Number 3672.21-0001

Part Number [redacted] 30 4065

November 1975 edition

The information in this publication is subject to periodic revisions and to updates by Publications Bulletins. A bulletin is identified as Publication Number 3672.21-00XX, where "XX" equals "01" for the first bulletin released.

To request copies of this publication or any bulletin, send a Supplies Requisition (form number 7020) using the publication number to: Memorex Corporation, Marketing Distribution Center, San Tomas at Central Expressway, Santa Clara, California 95052.

A readers' comments form is provided at the back of this publication. If the form has been removed, please send comments to Memorex Corporation, Department 6940, M/S 14-07, San Tomas at Central Expressway, Santa Clara, California 95052.

PREFACE

This manual provides the Memorex Field Engineer (FE) with detailed operating theory about the 3672 Storage Control Unit, used in 3672-based disc storage subsystems with the 3673 Disc Drive Controller and 3670/3675 Disc Drive Modules. It is intended for use by the FE during training as essential course material, and during servicing as an aid in identifying equipment difficulties. The manual consists of six sections:

Section 1 — Describes briefly the overall functions performed by the 3672 Storage Control Unit and the 3672-based subsystem.

Section 2 — Describes commands executed by the 3672. Includes command format and summary information, and a detailed flow diagram of each command.

Section 3 — Presents details about the 3672/channel interface.

Section 4 — Contains a description of the microinstructions and microprograms used to execute commands.

Section 5 — Describes operation of the 650 Flexible Disc File, used to store operating and diagnostic microprograms executed by the 3672.

Section 6 — Describes the detailed operation of the 3672 logic.

Maximum benefit of this manual is achieved when used with the *3672 Storage Control Unit Logic Diagram Manual,* P/N 308312. The alphanumeric numbers which appear in each functional block of the block diagrams in Section 6 refer to corresponding pages of logic in the *Logic Diagram Manual.*

Other manuals that support the 3672 Storage Control Unit which may be of use to the FE are listed below:

3672.22-00 — *3672 Storage Control Unit Installation Manual*

3672.20-00 — *3672 Storage Control Unit Maintenance Manual*

3672.50-01 — *3672 Storage Control Unit Microdiagnostics Reference Manual* (two volumes)

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

LIST OF TABLES

SECTION 1. GENERAL DESCRIPTION

1.1 SCOPE

This manual contains operating principles for the MEMOREX 3672 Storage Control Unit (SCU), which functions as the primary controlling element in the MEMOREX 3672-based Disc Storage Subsystem. Since the SCU exercises primary control in the Subsystem and participates in all Subsystem operations, this section of the manual will describe pertinent characteristics of the complete 3672-based Subsystem, with particular details about the 3672 SCU in the following sections.

1.2 SUBSYSTEM CHARACTERISTICS

The MEMOREX 3672-based Disc Storage Subsystem (Figure 1-1) provides high performance, large capacity, direct access data storage. The 3672-based Subsystem is completely compatible with the IBM 3830-II/3333/3330 Disc Storage Subsystem and can interface with IBM System/360 and 370 which utilize a Block Multiplexer Channel.

The 3672-based Subsystem is composed of a 3672 SCU and up to four 3673 Disc Drive Controllers, each attached to four 3670 or 3675 Disc Drive Modules (DDM). The 3675 DDM is a double-density version of the 3670 DDM. Each DDM contains two drive spindles. The DDMs may be mixed in any combination (Figure 1-2).

The 3672 SCU attaches on the upstream side to an IBM System/360 or 370 through a Block Multiplexer Channel, or to a Selector Channel by means of the optional 2860 Selector Channel Attachment Feature. The SCU attaches on the downstream side to a maximum of four 3673 Controllers through a Controller Interface (CTL-I). Each Controller, in turn, may attach to as many as four DDMs. The SCU may be shared between two, three, or four Central Processing Units (CPUs) by means of an optional channel switch attachment in the SCU, and may connect to one to four 3673 Controllers through an optional string switch feature in the Controllers.

Storage media used in the 3672-based Subsystem is the MEMOREX Mark X, IBM 3336-I, or equivalent disc pack for subsystems utilizing 3670 DDMS; or the MEMOREX Mark X1, IBM 3336-II, or equivalent disc pack for subsystems equipped with 3675 DDMs.

Removable logical address plugs permit changing the logical device addresses of the drives within the 3672-based Subsystem. A service plug is provided with each Subsystem to enable offline or inline servicing by the Memorex Field Engineer (FE).

The 3672-based Subsystem uses IBM System/360 or 370 Channel Control Words (CCWs) to identify the data operation to be performed, and the addresses of the 3672 SCU, 3673 Controllers, and 367X DDMs that are to be controlled. The SCU transmits control signals to the selected controller and drive control circuitry in logical sequence and at the proper time to complete the desired operation.

1.3 STANDARD FEATURES

- Error Correction

  Error correction capability allows the detection and correction of data errors within the Subsystem.

- Error Recovery

  Command retry capability of the SCU allows recovery from subsystem errors and permits the using system to be available during retry attempts.

- Rotational Position Sensing

  Rotational position sensing allows the location of a record through its angular displacement relative to index and releases the channel during most of the record search time.

- Microprogramming

  Microprograms are stored in Writable Control Storage (WCS), which is loaded from a flexible disc in the SCU.

- Flexible Discs

  The flexible discs contain the operating microprogram, diagnostic programs, and other backup programs that may be loaded into the WCS for execution. The disc is small and easily changed; therefore, program updates can be made easily by changing the disc.

- Usage/Error Recording

  The SCU maintains a record of statistical data (number of seeks executed and number of bytes read) and error information for each drive.

- Record Overflow

  Provides a means of processing records which exceed track boundaries within a cylinder.

- Drive Addressing

  The logical address of any disc drive is easily changed by changing the unit select plug on the drive operator control panel.

- Program Compatibility

  The Subsystem receives, decodes, and interprets commands from the IBM System/360 (with Block Multiplexer) or 370 channel. It responds to the same set of (CCWs) used in IBM programs to operate the 3330 Facility.

- Data Protection

  To protect data, the operator can inhibit write commands by utilizing a READ-ONLY switch on each drive.

- Operator/Diagnostic Console

  To facilitate operator monitoring and FE maintenance, each drive within the module employs a separate Operator/Diagnostic console.

- System Disabling

  The operator can easily take the 3672-based Subsystem offline (for diagnostic testing or maintenance) by setting a switch on the 3672 SCU operator control panel and ensuring that the CPU enters the Wait State momentarily.

- Priority

  The Memorex FE can assign a priority to each 3672 SCU and its address.

1.4 OPTIONAL FEATURES

- Two-Channel Switch

  The two-channel switch provides the capability for the 3672 SCU to be shared by two IBM System/360 or 370 Block Multiplexer Channels. The two channels may be attached to either the same or different CPUs. Individual drives attached to the 3672 SCU may be reserved for the exclusive use of either of the channels.

- Three-Channel Switch

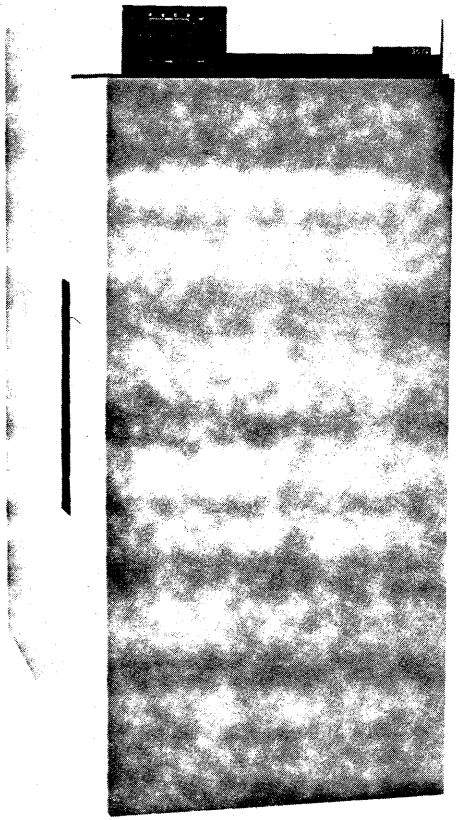
  The three-channel switch is identical to the two-channel switch in operation, except that three independent channels may be connected to the 3672 SCU. The three independent channels may be connected with one to three separate CPUs.

- Four-Channel Switch

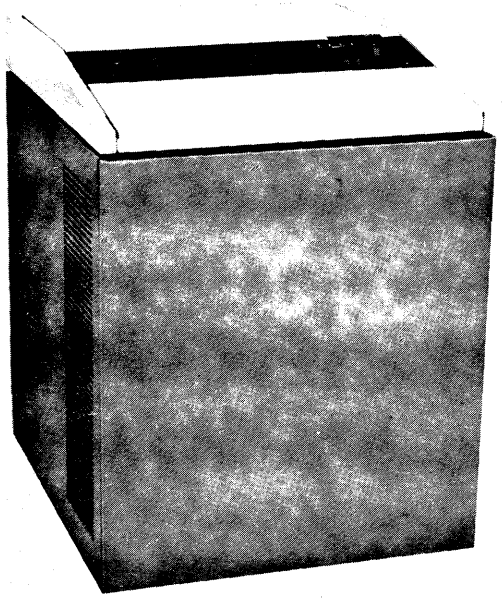
  The four-channel switch is identical to the two-channel switch in operation; however, four independent channels may be connected to the 3672 SCU. These four independent channels may be connected with one to four separate CPUs.

- Tag/Untag Switch

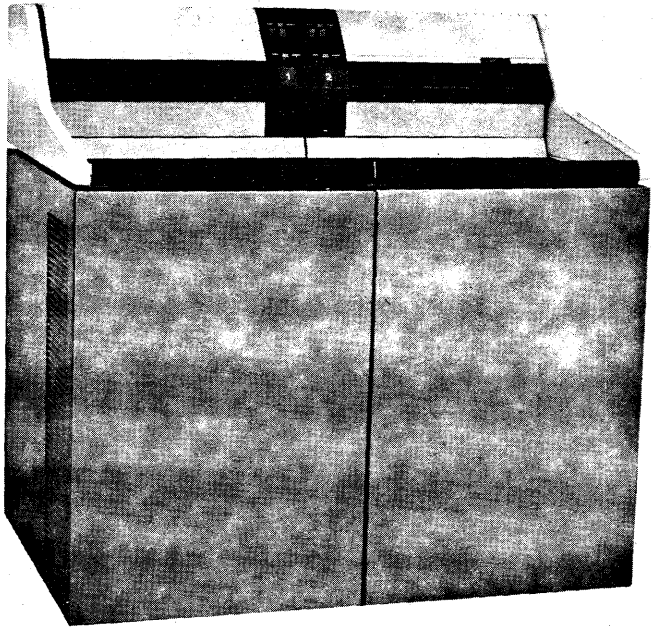
  This switch is utilized on SCUs equipped with one of the multiple channel switch features. When the switch is in the TAG position, each channel must accept the Device End signal resulting from a pack change or a unit plug change before the channel can



a. 3672 Storage Control Unit



b. 3673 Disc Drive Controller



c. 3670/3675 Disc Drive Module

Figure 1-1. 3672-Based Disc Storage Subsystem Components

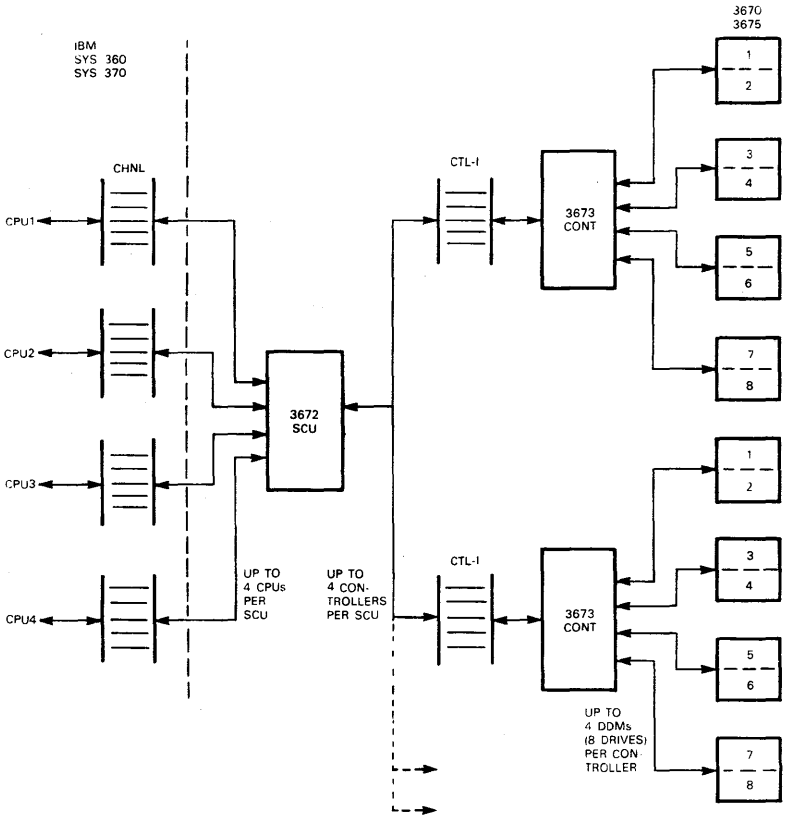


Figure 1-2. 3672-Based Subsystem Configurations

process data with that device. In the UNTAG position, acceptance of the Device End signal resulting from a pack change or a unit plug change by a channel will allow any channel to process data with that device.

- 32-Spindle Feature

  This feature allows for attaching up to 32 disc drive spindles (16 modules) to one 3672 SCU. The WCS Expansion is a prerequisite for this feature.

- WCS Expansion

  Expands WCS to support additional microprogramming. Required for 32-spindle addressing.

- Power Options

  A subsystem can be supplied to operate from a three-phase 208/230 Vac 60 Hz or 220/380 Vac 50 Hz power source.

- 2860 Selector Channel Attachment

  Permits the 3672-based Subsystem to be attached to a System/360 selector channel. Enables implementation of the software-provided Disconnected Command Chaining and Command Retry functions required for simulated block multiplexer operation with this attachment.

1.5 COMPATIBILITY

The 3672-based Subsystem attaches directly to the IBM System/370 Block Multiplexer Channels for Models 135 and up, and to System/360 Block Multiplexer Channels on Model 195. The Subsystem attaches to the selector channel, via the 2860 Selector Channel Attachment feature, of IBM System/360 Models 65, 67, 75, 91, and 95.

1.6 SPECIFICATIONS SUMMARY

Data Retrieval Times

Average Latency	8.33 msec
Track-to-Track Access Time	7 msec (maximum)
Maximum Access Time	50 msec
Average Access Time	27 msec
Data Transfer Rate	806,000 bytes/sec

Disc Pack Characteristics

No. of Recording Discs	10
No of Recording Surfaces	19
Tracks Per Surface	404 + 7 alternates (3670) 808 + 7 alternates (3675)
Track Capacity	13,030 bytes
Cylinder Capacity	247,570 bytes
Pack Capacity	100,018,280 bytes (3670) 200,036,560 bytes (3675)
Module Capacity	200,036,560 bytes (3670) 400,073,120 bytes (3675)

Data Recording Format

IBM 3330 Compatible

1.7 UNIT CHARACTERISTICS

In the discussion which follows, refer to the 3672-based Subsystem block diagram on Figure 1-3.

1.7.1 3672 Storage Control Unit

The 3672 SCU is a microprogrammed control unit with the microprogram stored in WCS. The WCS has its program loaded during the Power-On sequence from a flexible disc device (MRX 650) located in the SCU.

In general, the functions included in the SCU are as follows:

- Interfaces with the channel, receives commands from the channel, interprets and executes them.
- Interfaces with 3673 Controllers and 367X DDMs and controls the operation of each.
- Controls the transfer of information between the Controller and channel.
- Checks data and command for validity during all operations.
- Corrects data errors, or sends a correction algorithm to the channel for data correction, depending on where the error occured.

- Performs diagnostic evaluation of the Subsystem from the SCU maintenance panel.

In addition, the 3672 SCU performs the following miscellaneous functions:

- Overflow Record

  The SCU can operate on records that extend past the end of a track and continue on the next track. Overflow records are indicated as such in the flag byte, and when writing such records, the *Write Special Count, Key, and Data* command must be used.

- Multitrack Operation

  On Read or Search commands the operation will become a multitrack operation if bit 0 of the respective command is a 1. This allows an entire cylinder to be searched with one command.

- Channel Disconnect Operation

  When operating in conjunction with a block multiplex channel, the SCU can disconnect from the channel when a command requiring mechanical motion is being executed. This disconnect can occur even though chaining is indicated by the channel. The SCU retains all information required to control a disconnected CCW chain for each attached disc spindle.

- Error Correction

  The error correction function provided in the 3672-based Subsystem allows the SCU to detect errors and correct these errors through the use of Command Retry. The correction of a failing command through Command Retry is used for the following errors:

  1. If the data is determined to be uncorrectable (i.e., the error exceeds the correction code capability), the channel is signalled and the command may be retried under control of the channel.

  2. If the data error is correctable and it occurs in the data field of a record, the SCU passes on to the channel the information necessary to correct the data.

  3. If the data error is correctable and occurs in the Home Address, Count or Key fields, the SCU corrects the data which is being held in a buffer internally and then sends the corrected data to the channel. Upon reorientation to the field in error, the corrected data is transferred from the buffer to the channel. During the reorientation time, the SCU is disconnected from the channel.

  4. Miscellaneous Errors

     a. *Seek Errors* — If a Seek error is detected by the SCU, the Seek is retried before it is considered a malfunction.

     b. *Defective Track* — If a defective track is detected, the channel is notified and the user's control program may flag the defective track and reissue the command to cause the data to be written on an alternate track.

     c. *Command Overrun* — If during command chaining the channel fails to meet the real-time chaining constraint of the Subsystem, the SCU will initiate a retry of the failing command.

     d. *Data Overrun* — If during data transfer the channel fails to meet the real-time requirements of the Subsystem, the SCU will initiate a retry of the failing command.

- Usage and Error Logging

  The SCU contains counters which enable it to maintain a statistical log of the usage and error occurrences for each drive in the Subsystem. The usage counters count the total number of access motions and the total number of data bytes transferred to the channel. The error counter is an accumulation of the total number of correctable or uncorrectable Read errors and Seek errors. The counter data is periodically transferred to the channel for system logging.

- Diagnostics

  The control unit contains microprograms which are used for online, inline, and offline servicing of the SCU, Controllers, and DDMs. When the FE inserts the service module plug into one of the drives, the drive becomes offline to the operating system and online to the control unit. Using the proper Memorex diagnostic programs, diagnostics can then be run and error messages displayed via the using system. Diagnostics also can be run from the Field Engineer (FE) panel on the SCU. Error messages are displayed on the same FE panel.

The control of the 3673 Controller and 367X Disc Drives, and the data processing to provide the recording format on the disc track, is defined by the microprogram routines which are resident in the WCS portion of the Microprocessor. Logic signals to operate the disc drives and sequence the CPU channel are obtained from latches set by the microprogram in the Controller Interface and Channel Interface, respectively. The latches for a given function are grouped into a register, and binary data is loaded into these registers in timed sequences to stimulate the associated hardware. The microprogram can interrogate the status of the different groups of hardware by testing the state of the appropriate latches. These latches are also grouped by function into registers. Testing is accomplished by selecting the desired register and isolating the bit or bits to be investigated. The microprogram sequence is then alternated by branching if the state of the latch(es) corresponds to the condition for branching.

There is a prescribed interaction involving stimulus and response in which the microprogram controls both the timing and the sequence. The loading and accessing of registers is defined by the operand fields of the microinstruction word and the register's data is moved over internal logic buses.

1.7.2 3673 Disc Drive Controller

The 3673 Disc Drive Controller functions as an electronic interface unit that provides drive-related control functions to the DDMs. It accomplishes this by decoding and executing tag instructions from the Storage Control Unit. These tag instructions are divided into three groups: drive-related tags, controller-related tags, and drive/controller-related tags.

Drive-related tags are received over the CTL-I interface of the Controller and passed directly to the drive. The Controller does not operate on this group of tags and is completely transparent to their occurrence, except for certain data checking validity functions.

Controller-related tags are operated on by the Controller. These tags are not passed to the drive.

Controller/drive-related tags are operated on jointly by the Controller and drive. The read/write tags fall into this group as well as certain diagnostic tags used for diagnostic purposes

Functionally, the Controller is divided into three logical areas as shown in Figure 1-3: the Controller Interface logic, the Drive Interface logic, and the Read/Write logic. The interface groups provide signals between the Controller and the SCU and drives by means of line drivers and receivers. The Read/Write logic consists of two subgroups: the read/write data transfer and read/write control. The data transfer group consists of registers, multiplexers, and other logic required to move the data through the Controller to either the SCU, if reading data, or to the selected drive, if writing data. This group also contains the serializer/deserializer (serdes) used to assemble data into either serial form for writing onto the disc or into parallel form (deserialize) if reading from the disc. The control logic consists of latches, counters, and related logic that supervise the transfer of data under control of tag instructions.

1.7.3 3670 (3675) Disc Drive Module

The 3670 (3675) DDM consists of two independent disc storage drives. Each drive (spindle) stores and retrieves data on a MEMOREX Mark X, IBM 3336-I, or equivalent disc pack (MEMOREX Mark XI, IBM 3336-II, or equivalent for 3675 DDMs). Disc pack information written on one 3670 (3675) Drive may be retrieved or updated on any other 3670 (3675) Drive, IBM 3330 (3330-I), or equivalent.

Each drive is capable of seeking to any one of 404 (808) cylinders plus 7 alternates, selecting any one of 19 heads, and transferring information to the control unit at a nominal data rate of 806 kilobytes per second.

Up to four modules can be attached to the 3673 Controller. The primary functions performed by each of the disc drives are as follows:

- Position the access mechanism to a cylinder.
- Select a head.
- Read or write data.
- Respond to commands given by the Storage Control Unit.

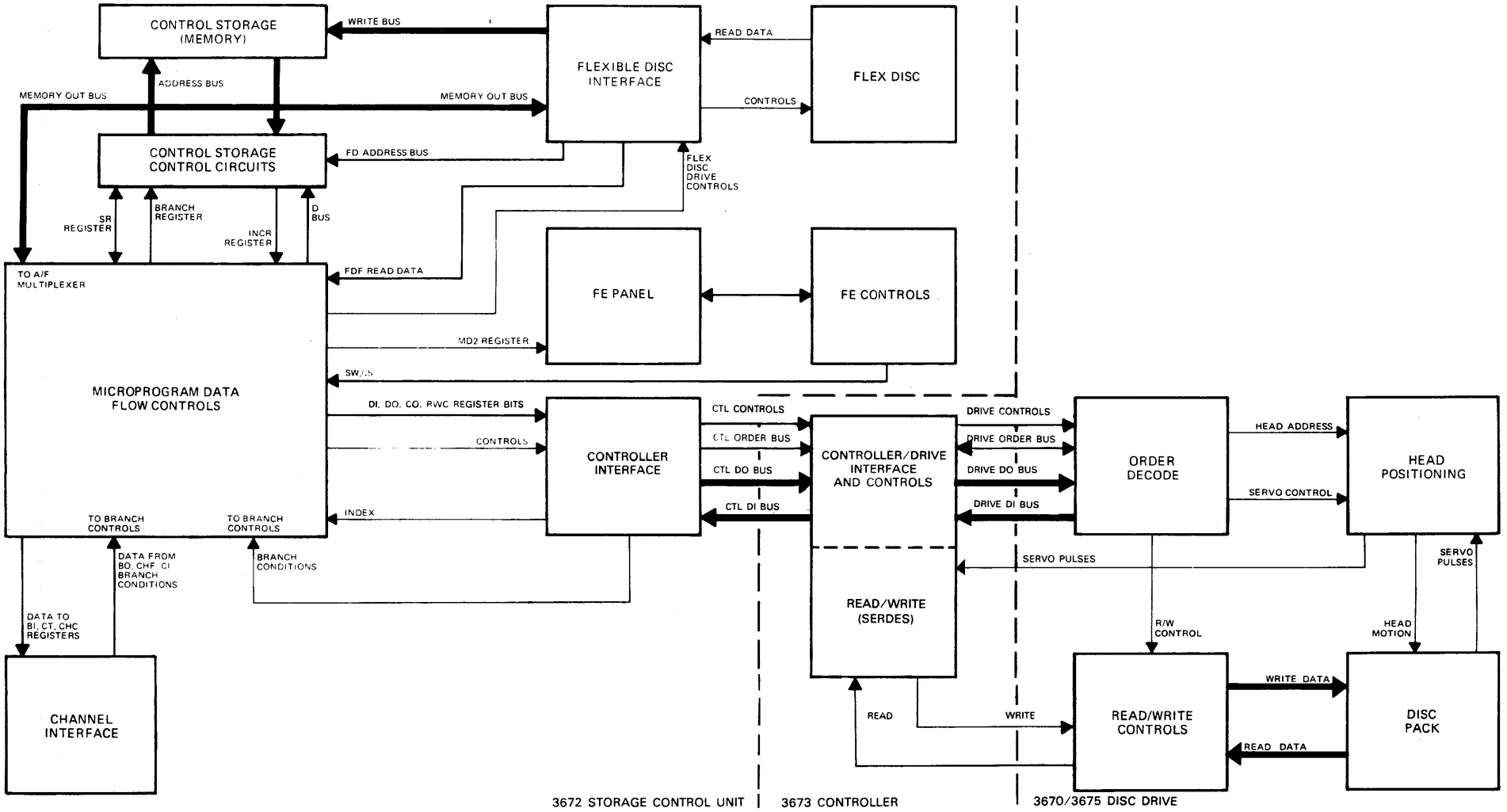


Figure 1-3. Disc Storage Subsystem Operational Block Diagram

The drive assembly consists of a motor and a spindle drive. The removable Mark X (Mark XI) pack is mounted on the spindle, which is driven at 3600 RPM. The motor cannot be started unless the pack access cover is closed. The access cover is locked automatically whenever the spindle motor is energized or a stop sequence is in progress.

The access mechanism consists of an electromagnetic voice coil motor and a carriage assembly on which the recording heads are mounted. Major assemblies of the access mechanism are:

- Actuator

  An electromagnetic actuator moves the carriage assembly to any of 411 (815) cylinder positions.

- Carriage Assembly

  The carriage assembly consists of 19 Read/Write heads and one servo head mounted on a carriage.

Access to any cylinder is achieved by a linear voice coil motor which drives a carriage on which the data heads and servo head are attached. The servo head is used to position and lock the carriage on the desired cylinder by servoing on information which was prerecorded on the disc pack at the factory. Once the desired cylinder has been reached and the servo system has locked on the servo track, any data head can be electronically selected and the data is recorded on, or retrieved from, the track.

1.8 DATA INTERFACE REQUIREMENTS

All interface cabling is provided with the 3672-based Subsystem. The channel interface circuitry in the Subsystem is compatible with the I/O interface requirements established by IBM.

The Memorex FE can establish priority to any 3672-based Subsystem. This capability enables users of multiple 3672-based Subsystems to reserve individual systems for selected applications. For example, high-activity applications can be reserved for a system assigned a normal priority, while low-activity applications can be reserved for a system assigned a low priority.

SECTION 2. COMMANDS

2.1 GENERAL

This section defines and discusses the commands performed by the 3672 SCU during execution of I/O instructions. The section is divided roughly into two halves. The first half contains instruction and command word formats, followed by a detailed description of each command in both narrative and flow diagram form. The second half discusses command-related hardware operations, such as command retry, rotational position sensing, and overflow records.

2.2 COMMAND SUMMARY

Table 2-1 summarizes the command groupings and coding information for the 3672-based subsystem commands. Detailed descriptions of individual commands are given on following pages.

2.3 INSTRUCTION AND WORD FORMAT

2.3.1 Introduction

The following material describes the formats for the following I/O instructions, channel words, and the Program Status Word:

- Start I/O
- Start I/O Fast Release
- Test I/O
- Halt I/O
- Halt Device
- Channel Address Word
- Channel Command Word
- Channel Status Word
- Program Status Word

2.3.2 I/O Instructions

The I/O instructions in the CPU program initiate I/O operations of the 3672-based subsystem. The operations are controlled by the channel using commands read from main storage. Arithmetic and logical (decision) operations are performed while the CPU is in the problem state: for I/O operations, the CPU must be in the supervisor state. The CPU is changed from the problem to the supervisor state when a Supervisor Call Routine is executed or when an I/O interrupt sequence occurs.

The status of the CPU existing at the time of the change is stored in the Program Status Word (PSW). After the specified instruction has been executed, the CPU can return to the problem state and continue the interrupted program by reloading the PSW originally stored when the program entered the supervisor state. In the supervisor state, the CPU can execute the following I/O instructions: *Start I/O, Start I/O Fast Release, Test I/O, Halt I/O,* and *Halt Device*.

START I/O

The *Start I/O* instruction initiates an I/O operation upon detection that the addressed channel, SCU, and drive are available.

START I/O FAST RELEASE

The *Start I/O Fast Release* instruction initiates an I/O operation upon detection that the addressed channel is available. The SCU and drive are assumed to be already available. If not, an I/O interrupt sequence occurs to indicate an unavailable condition.

TEST I/O

The *Test I/O* instruction sets the condition code in the PSW to indicate the status of the addressed channel, subchannel, SCU, and disc drive.

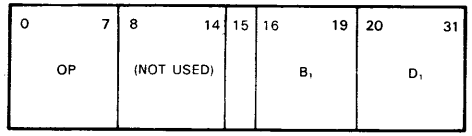
Halt I/O

The *Halt I/O* instruction terminates the operation in progress at the channel

HALT DEVICE

The *Halt Device* instruction terminates the operation in progress at the SCU without interfering with other I/O operations at the channel. This instruction is used instead of *Halt I/O* to terminate an operation on a device attached to multiplexer channels.

The I/O instruction format is as follows



0-7 OP (Operation Code)

Specifies the operation to be performed.

8-14 Not Used.

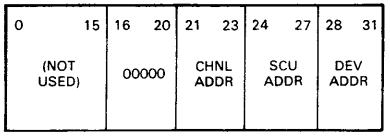
15 Set to 1 for *Start I/O Fast Release* and *Halt Device* instructions.

16-19 B_1 (Base Address Register Location)

Specifies the address of a general register in the CPU. The register is 32 bits long, but only the low-order 24 bits are used.

20-31 D_1 (Displacement)

An immediate field added to the contents of the register at B_1 to develop bits 16-31 of a 32-bit result. This result identifies the channel and device addressed by the instruction and has the following format:



0-15 Not Used

16-20 Must be zero.

21-23 CHNL ADDR (Channel Address)

Designates channel address.

24-27 SCU ADDR (SCU Address)

Designates SCU address.

29-31 DEV ADDR (Device Address)

Designates drive address.

2.3.3 Channel Address Word

The Channel Address Word (CAW) is read by the channel from main storage location 72 when a *Start I/O* or *Start I/O Fast Release* instruction is issued. Bits 0-3 of the CAW form the protection key for all commands associated with the I/O instruction. The protection key establishes the privilege of access to the particular main storage locations (for example, whether data can be stored or read). The command address in bits 8-31 designates the address of the first Channel Command Word (CCW). The three low-order bits of the command address must be zero to specify the CCW on double-word boundaries. CAWs are read by the channel hardware. The information must be set up in main storage location 72 prior to issuing the I/O instruction.

The Channel Address Word format is as follows:

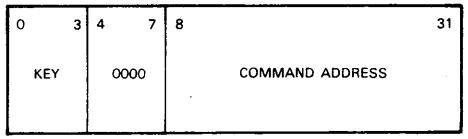


TABLE 2-1. COMMAND SUMMARY

COMMAND GROUPINGS	COMMAND NAME	HEX CODE: SINGLE TRACK	HEX CODE: MULTI-TRACK
CONTROL — **CONTROL COMMANDS** start operations not involving a transfer of data records between the SCU microprocessor and system main storage. These bytes enable the operation to take place and are parity checked during transfer. For most control functions, the entire operation to be started is specified by the command code. If not so specified, the additional information needed is fetched from main storage.	NO-OPeration	03	–
	SEEK	07	–
	SEEK CYLINDER	0B	–
	SEEK HEAD	1B	–
	SPACE COUNT	0F	–
	RECALIBRATE	13	–
	RESTORE	17	–
	SET FILE MASK	1F	–
	SET SECTOR	23	–
	DIAGNOSTIC LOAD	53	–
	DIAGNOSTIC WRITE	73	–
	SET RPS †	2F	–
SENSE — **SENSE COMMANDS** determine the status of the Subsystem and identify the specific nature of any errors or unusual conditions that have occurred.	SENSE I/O	04	–
	READ AND RESET BUFFERED LOG	A4	–
	READ DIAGNOSTIC STATUS 1	44	–
	DEVICE RESERVE	B4	–
	DEVICE RELEASE	94	–
READ — **READ COMMANDS** transfer information from a disc drive to the system CPU. All except Read IPL and Read Sector may operate on overflow records and in multitrack as well as single-track mode. On all read commands, the SCU examines correction code bytes to check the validity of each record area, and adds a parity bit to each byte.	READ DATA	06	86
	READ KEY AND DATA	0E	8E
	READ COUNT, KEY, AND DATA	1E	9E
	READ R0	16	96
	READ COUNT	12	92
	READ HOME ADDRESS	1A	9A
	READ IPL	02	–
	READ SECTOR	22	–
WRITE — **WRITE COMMANDS** transfer information from the system main storage to the SCU for recording on a disc pack. While writing a record on a disc track, the Controller appends the appropriate correction code bytes to each record area. **Update Write Commands (non-formatting)** operate on previously formatted tracks to update existing records. **Formatting Write Commands** initialize tracks and records, and establish the length of the areas within each record.	WRITE DATA	05	–
	WRITE KEY AND DATA	0D	–
	WRITE COUNT, KEY, AND DATA	1D	–
	WRITE SPECIAL COUNT, KEY, AND DATA	01	–
	WRITE R0	15	–
	WRITE HOME ADDRESS	19	–
	ERASE	11	–
SEARCH — **SEARCH COMMANDS** transfer a specific number of bytes from system main storage to the SCU. While executing a command, the channel operates in the Write mode while the disc storage operates in the Read mode. Incoming data from main storage is compared with outgoing data from disc storage. When search criteria specified in the command are satisfied, the status modifier bit is set. This bit causes the channel to skip the next CCW in the chain and fetch the next command from main storage. Each search command operates on one record at a time. To search another record, the command must be reissued.	SEARCH HOME ADDRESS EQUAL	39	B9
	SEARCH ID EQUAL	31	B1
	SEARCH ID HIGH	51	D1
	SEARCH ID EQUAL OR HIGH	71	F1
	SEARCH KEY EQUAL	29	A9
	SEARCH KEY HIGH	49	C9
	SEARCH KEY EQUAL OR HIGH	69	E9

SUMMARY INFORMATION

HEX CODES OF DRIVE ADDRESS BYTES

	0	1	2	3	4	5
BIN (PACK)	00	00				
CYLINDER			0000-019A		← 3670	
			0000-032E		← 3675	
HEAD					00	00-12

MASK BYTE BITS / **HEX CODES OF PERMIT COMMANDS**

0	1	3	4	WRITE 05	0D	1D	01	15	19	11	SEEK 07	0B	1B
0	0			√	√	√	√			√			
0	1			NONE PERMITTED									
1	0			√	√								
1	1			√	√	√	√	√	√	√			
		0	0								√	√	√
		0	1									√	√
		1	0										√
		1	1								NONE PERMITTED		

HEX CODE OF PREREQUISITE COMMAND (ANY ONE)

	15	19	29/A9*	31/B1*	39/B9*	1D
WRITE DATA			√	√		
WRITE KEY AND DATA				√		
WRITE COUNT, KEY, AND DATA	√		√	√		√
WRITE SPECIAL COUNT, KEY, AND DATA	√		√	√		√
WRITE R0		√			√	
WRITE HOME ADDRESS	NOT REQUIRED					
ERASE	√		√	√		√

BITS SET IF SEARCH CRITERIA ARE SATISFIED

STATUS BYTE	CSW BIT	NAME
1	33	STATUS MODIFIER
	36	CHANNEL END
	37	DEVICE END

† Set RPS command is valid only if the 2860 Attachment Feature is installed. See paragraph 3.12.

* Search must not be truncated.

0-3 KEY (Protection Key)

Identifies the storage protection key for all commands associated with *Start I/O*. This key must match the storage key.

4-7 Must be zero.

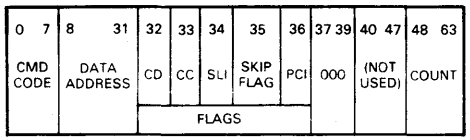
8-31 Command Address

Designates the location of the first CCW in main storage.

2.3.4 Channel Command Word

The Channel Command Word (CCW) is read by the channel from the address specified in the CAW. The first CCW specifies the operation to be performed, the main storage locations to be used, and the action to be taken when the operation is completed. If available when it receives the CCW, the channel attempts to select the I/O device specified in the I/O instruction by sending the address to all attached SCUs. If the addressed I/O device is attached to the channel and has power on, the command code portion of the CCW is sent to the SCU, which responds with an initial status byte to the channel. At this point, the *Start I/O* instruction is finished, releasing the CPU to perform the next instruction.

The results of the attempt to initiate execution of the command are indicated by the condition code in the Program Status Word (PSW). If the I/O operation was not started, new status information containing the reason for this condition is normally set in the Channel Status Word (CSW). The CCW format is as follows



0-7 CMD CODE (Command Code)

Specifies the operation to be performed. Either the two or four low-order bits of the command code identify the type of operation to the channel as follows:

Control	XXXX XX11
Sense	XXXX X100
Read	XXXX XX10
Write	XXXX XX01
Search	XXXX X001

The channel distinguishes Write, Control, Read, Sense, or Transfer-in-Channel (TIC) operations. All eight bits are transferred to the SCU when I/O operations are initiated.

8-31 Data Address

Specifies the location of a two-byte address in main storage. This is the address of the area associated with data transfer operations.

32 CD (Chain Data)

When set to 1, specifies chaining of data.

33 CC (Chain Command Flag)

When set to 1, and when the CD flag is zero, specifies chaining of commands. It causes the operation specified by the command code in the next CCW to be initiated upon normal completion of the current operation.

34 SLI (Suppress Length Indication)

When set to 1, suppresses an incorrect length condition except when the CCW count is not exhausted, in which case Channel End is present and data chaining is indicated.

35 Skip Flag

When set to 1, specifies suppression of a transfer of information to main storage during a Read or Sense operation. Checking takes place as though the information has been placed in main storage. When this bit is 0, normal transfer of data takes place.

36 PCI (Program Control Interruption)

When set to 1, causes the channel to generate an interrupt condition upon reading the CCW. When this bit is 0, normal operation takes place.

37-39 For every CCW other than one specifying TIC (Transfer in Channel), these bits contain zeros. Violation of this restriction generates a program-check condition.

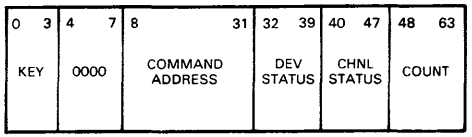
40-47 Not Used.

48-63 Count

Specifies the number of eight-bit byte locations in main storage area designated by the data address.

2.3.5 Channel Status Word

The Channel Status Word (CSW) informs the program of I/O device status or the conditions under which an I/O operation was terminated. It is stored at main storage location 64. The CSW is formed or changed during I/O interruptions and instruction execution. Status stored in the CSW remains unchanged until a subsequent interrupt occurs or a new I/O instruction is processed.



The Channel Status Word format is as follows:

0-3 KEY (Protection Key)

The storage protection key used in the chain of operations.

4-7 Must be zero.

8-31 Command Address

An address that is eight positions higher than the address of the last CCW used.

32 Attention—not used.

33 Status Modifier

Set for three conditions (1) Whenever a *Search ID/Key High, Search ID/Key Equal,* or *Search ID/Key Equal or High* command has been executed and the search criteria satisfied; (2) With Busy bit to indicate SCU Busy, or (3) With Unit Check and Channel End to indicate retry status.

34 SCU End

Set if an SCU Busy status has been generated previously and the busy condition has been terminated. SCU End may be given with any device address recognized by the SCU.

35 Busy

Indicates that the selected drive is busy. In conjunction with bit 33, indicates the SCU is busy.

36 Channel End

Set at the end of each channel command.

37 Device End

Indicates that an access mechanism is free to be used.

38 Unit Check

Set whenever an unusual or error condition is detected.

39 Unit Exception

Indicates an End-of-File has been detected during a *Read R0, Read IPL, Read Count, Key, and Data, Read Key and Data, Read Data, Write Key and Data,* or *Write Data* operation. It results from a data length of zero being detected in the count area of a record. When detected, no data is transferred from the data area. If the key length is not zero, the key area is transferred.

40-47 CHNL STATUS (Channel Status)

Indicate channel condition as follows:

Bit 40: Program-Controlled Interruption
Bit 41: Incorrect Length
Bit 42: Program Check
Bit 43: Protection Check
Bit 44: Channel Data Check
Bit 45: Channel Control Check
Bit 46: Interface Control Check
Bit 47: Chaining Check

48-63 Count

The residual byte count from the last CCW used.

2.3.5.1 STATUS PRESENTATION

Status is presented twice (initial status and ending status) for all commands except those commands that require access motion, and immediate commands not chained from write commands. Seek and Seek Cylinder commands present status three times: initial status, Channel End status (after transfer of the Seek Address), and Device End status (after the access is positioned).

2.3.5.2 INITIAL STATUS

The initial status byte is zero for *Test I/O* and all non-immediate commands unless one or more of the following conditions exist: SCU is busy, a status condition is pending, a Unit Check occurred, or initial status indicated Command Retry. Immediate commands (for example, commands not requiring data transfer) present Channel End and Device End in initial status.

2.3.5.3 ENDING STATUS

In most cases, Channel End and Device End are presented as the normal ending sequence for an operation. The exceptions are noted in the individual command descriptions which follow. If an error occurred during the operation, Unit Check will accompany the Channel End and Device End status.

2.3.5.4 PENDING STATUS

A pending status condition may exist for the SCU or disc drive. Status is pending for the SCU if one of the following conditions has ocurred:

- A disconnect was signalled after a command was issued, but before Channel End was accepted.
- Busy, Channel End, or Unit Check status was stacked by the channel.
- Zero status was stacked by the channel in response to a *Test I/O*.
- SCU Busy was presented to the channel
- Unit Check was detected for an operation after Device End had been cleared.
- Device End status for a Set Sector command was stacked by the channel.

Status pending for the SCU causes it to appear busy for all devices except the device for which the status condition exists. Unless the SCU is busy, it will request service to clear the pending status condition. Status is cleared when presented to, and accepted by, the channel. Status is pending for a drive if one of the following conditions has occurred:

- Channel End appears alone.
- Busy status is presented.
- The drive has gone from not-ready to ready.

Status pending for a disc drive causes the SCU to request service when both the SCU and disc drive are not busy. The status is cleared when presented to, and accepted by, the channel.

2.3.5.5 CONTINGENT CONNECTION

A contingent connection is established in the SCU after the channel accepts a status byte containing Unit Check. The connection lasts until one of the following conditions occur:

- A command (other than *Test I/O* or *No-Op*) receives an initial status byte of zero for the SCU and disc drive address which generated the Unit Check.
- A Selective or System Reset occurs.

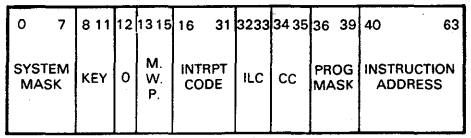
During the contingent connection state, the SCU appears busy to all SCU and device addresses other than the address for which the contingent connection was established.

2.3.6 Program Status Word

The Program Status Word (PSW) contains the status of the using system. Two PSWs are associated with subsystem interrupt conditions: an old PSW which contains the status information of the using system existing at the time of the interrupt, and a new PSW which is used to control instruction sequencing and hold the status of the using system relative to the program being executed. By storing the new PSW during an interruption, the CPU status is preserved for subsequent inspection by the program.

Loading a new PSW causes the state of the CPU to be initialized or changed to branch to a new instruction sequence. If, at the conclusion of an interrupt routine, an instruction is executed which restores the old PSW as the new PSW, the using system is restored to the state existing prior to the interruption, and the interrupted routine continues.

The PSW format is as follows:



0-7 System Mask

Designates the system mask as follows:

Bit 0: Channel 0 Mask
Bit 1: Channel 1 Mask
Bit 2: Channel 2 Mask
Bit 3: Channel 3 Mask
Bit 4: Channel 4 Mask
Bit 5: Channel 5 Mask
Bit 6: Channel 6 Mask
Bit 7: External Mask

8-11 KEY (Protection Key)

Identifies the storage protection key.

12 Must be zero.

13 M (Machine Check Mask)

14 W (Wait State)

15 P (Problem State)

16-31 INTRPT CODE (Interruption Code)

32-33 ILC (Instruction Length Code)

34-35 CC (Condition Code)

36-39 PROG MASK (Program Mask)

Designates the Program Mask as follows:

Bit 36: Fixed-Point Overflow Mask
Bit 37: Decimal Overflow Mask
Bit 38: Exponent Underflow Mask
Bit 39: Significance Mask

40-63 Instruction Address

2.4 CONTROL COMMANDS

2.4.1 Operation Block Diagram

The block diagram shown in Figure 2-1 and the text below provide an illustration of the System Control operation during the execution of a typical control command—that of a *Seek* command.

2.4.1.1 SYSTEM

The CPU executes a *Start I/O* instruction. Channel Address Word (CAW) specifies the main storage location of the first Channel Command Word (CCW). The SCU and drive to be used are specified by the *Start I/O* instruction.

2.4.1.2 SYSTEM INTERFACE

- Channel executes the CCW to transfer the *Seek* command and Seek address to the SCU
- Channel disconnects until the SCU signals with Device End status.
- Controls provide timing for data transfer and check parity of data transferred.

2.4.1.3 STORAGE CONTROL UNIT

- Channel Interface provides for communication between the SCU microprogram and the System Interface.
- Microprocessor contains the subsystem memory and all hardware for control of data flow between the Channel Interface and Disc Drive Module. Components are:
  a. Writable Control Storage that contains microprogram routines that define the data handling and drive control.
  b. Registers which decode and store the command code, and receive record-length control information
  c. Arithmetic Logic Unit which calculates the difference between present head location and new address in the stored command code.
- Controller Interface provides for communication of control, status, and address information between the SCU, Controller, and the Disc Drive Module.
- FE Interface provides controls and indicators for facility operation (at the Operator Panel) and maintenance (at the Interior Control Panel).

2.4.1.4 DRIVE CONTROLLER

The Drive Controller provides communication path for control, address, and status information between the SCU and the Disc Drive Module.

2.4.1.5 DISC DRIVE MODULE

- Head Positioning Controls determine the direction and amount of movement of the Head Access Mechanism. They maintain head position, and signal to the SCU when the desired new location is reached.
- Head Access Mechanism moves the head-arm assembly
- Disc Pack provides signal for each track passed from the servo surface in the pack.

2.4.2 Command Descriptions

Each control command is described in the following pages by means of a narrative description and a flow diagram. The number(s) adjacent to each flow diagram block refer to a routine in the 3672 μProgram Flowchart Manual. Numbers not in parentheses refer to mainline routines, while numbers within parentheses designate subroutines called by the mainline routines.

Many of the flow diagrams begin with an INITIAL SELECTION sequence. The flow diagram for this sequence is shown in Figure 2-1A.

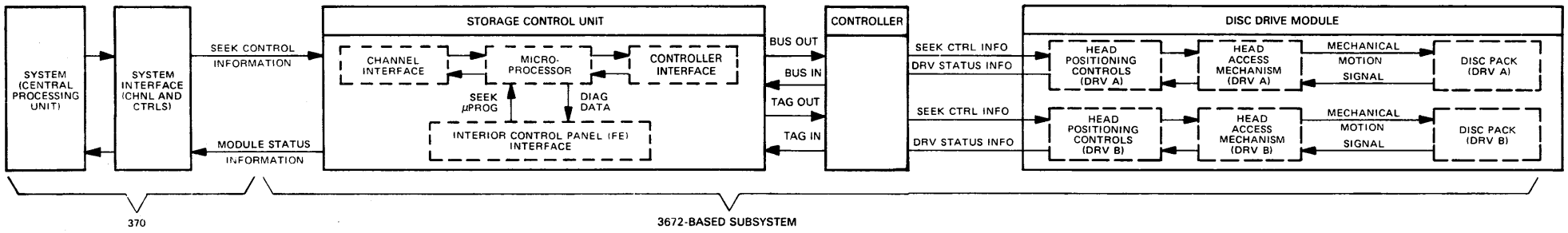


Figure 2-1. Seek Operation Block Diagram

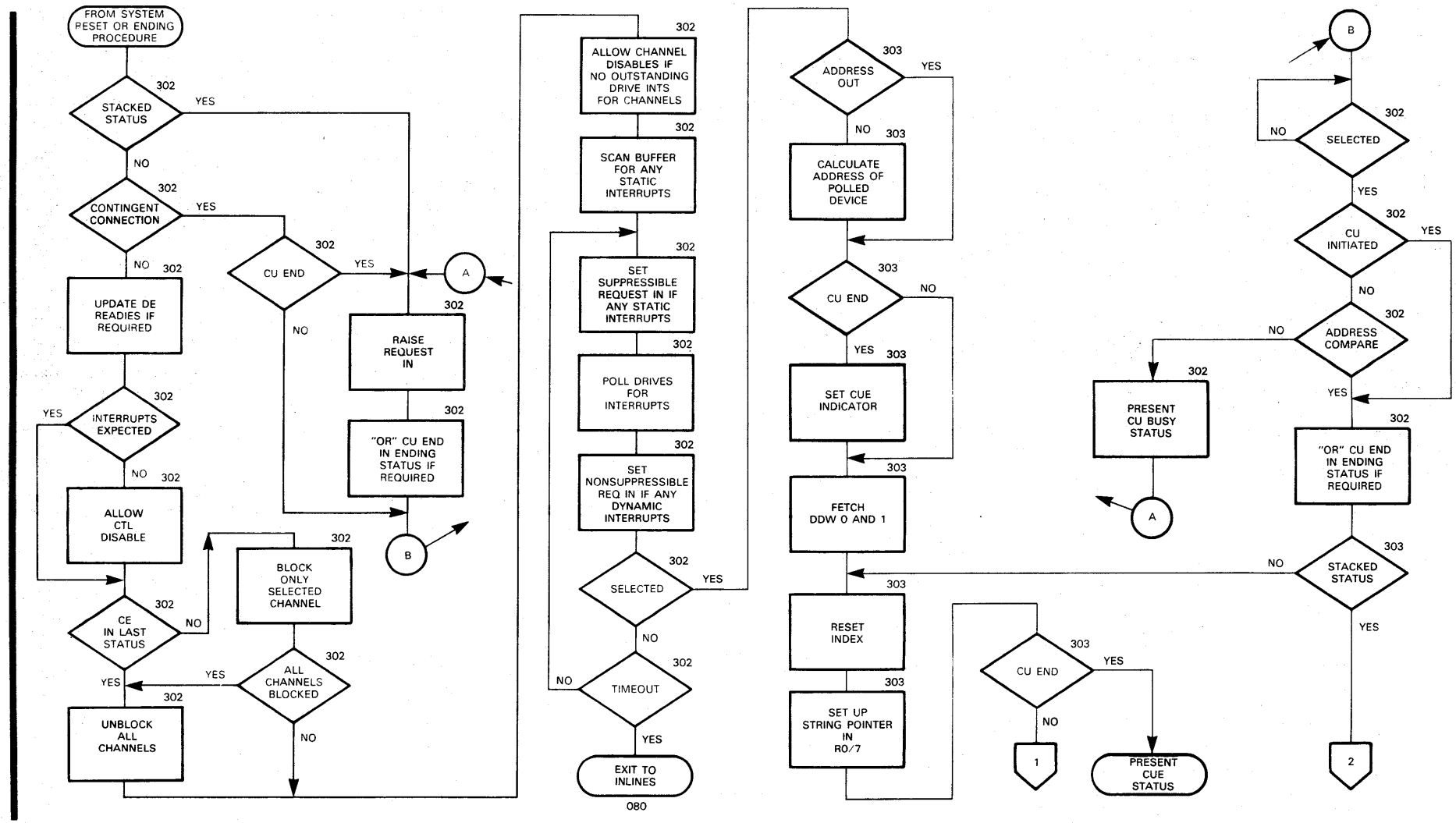


Figure 2-1A. Initial Selection Sequence, Block Diagram (1 of 6)

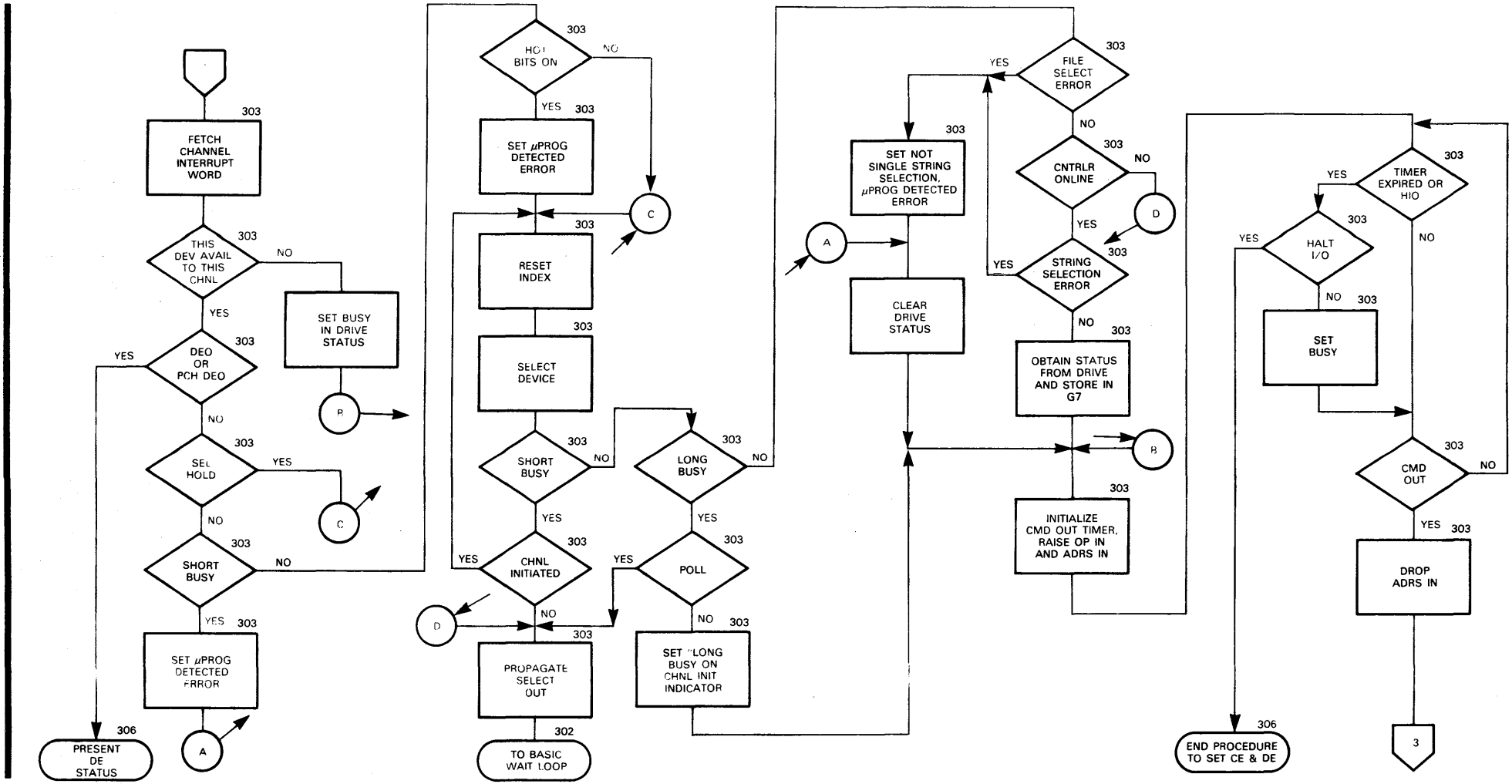


Figure 2-1A. Initial Selection Sequence, Block Diagram (2 of 6)

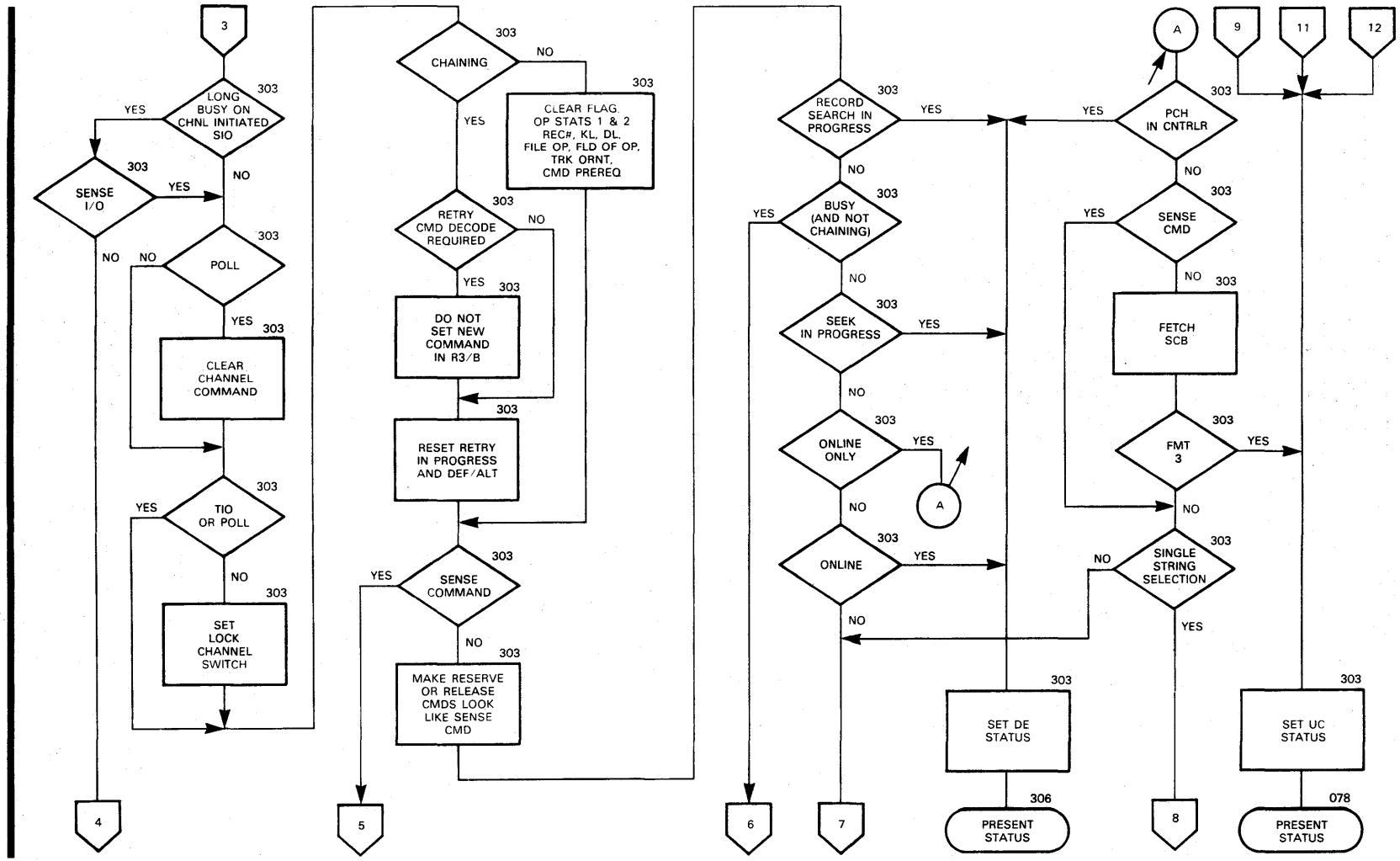


Figure 2-1A. Initial Selection Sequence, Block Diagram (3 of 6)

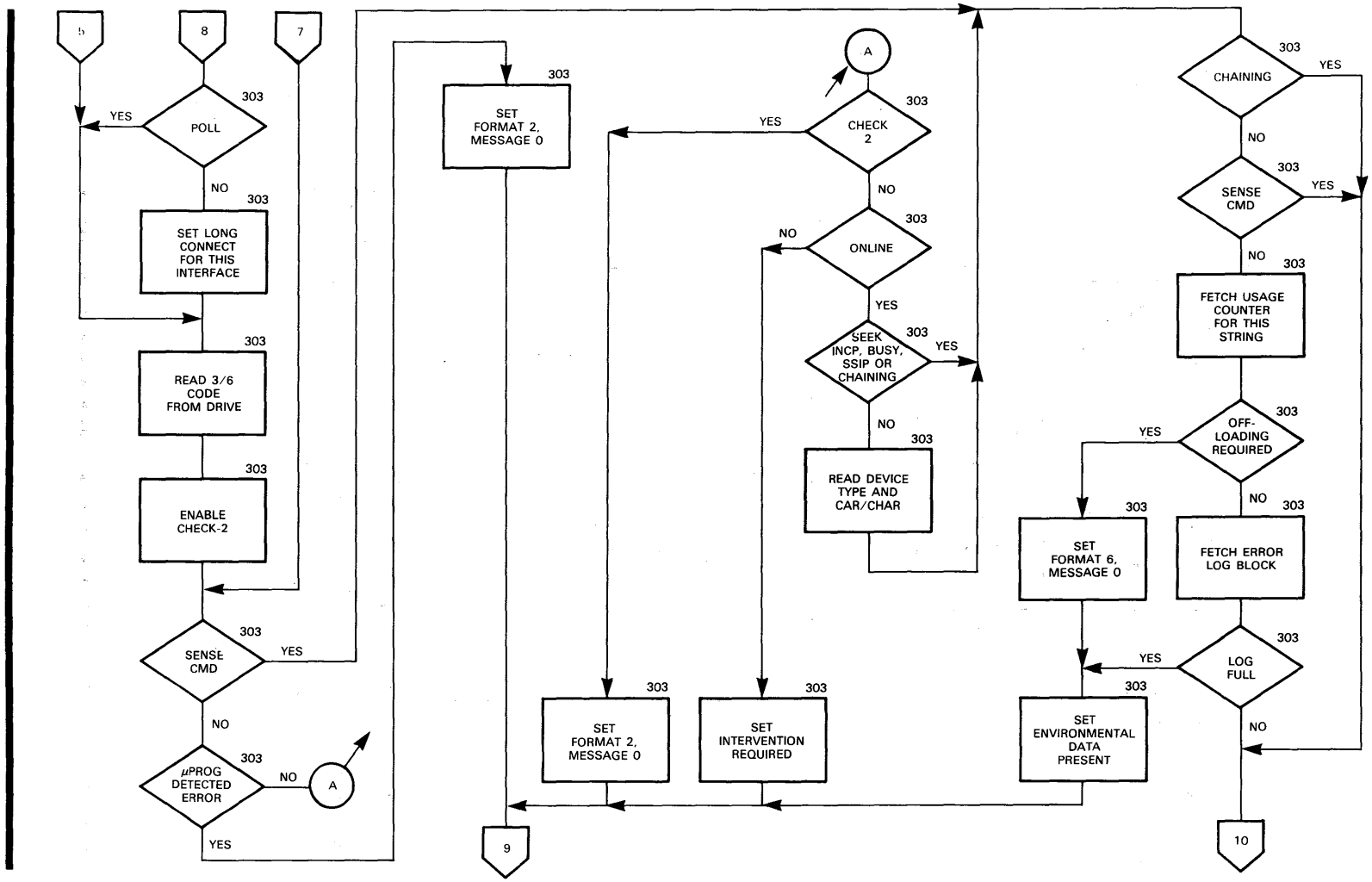


Figure 2-1A. Initial Selection Sequence, Block Diagram (4 of 6)

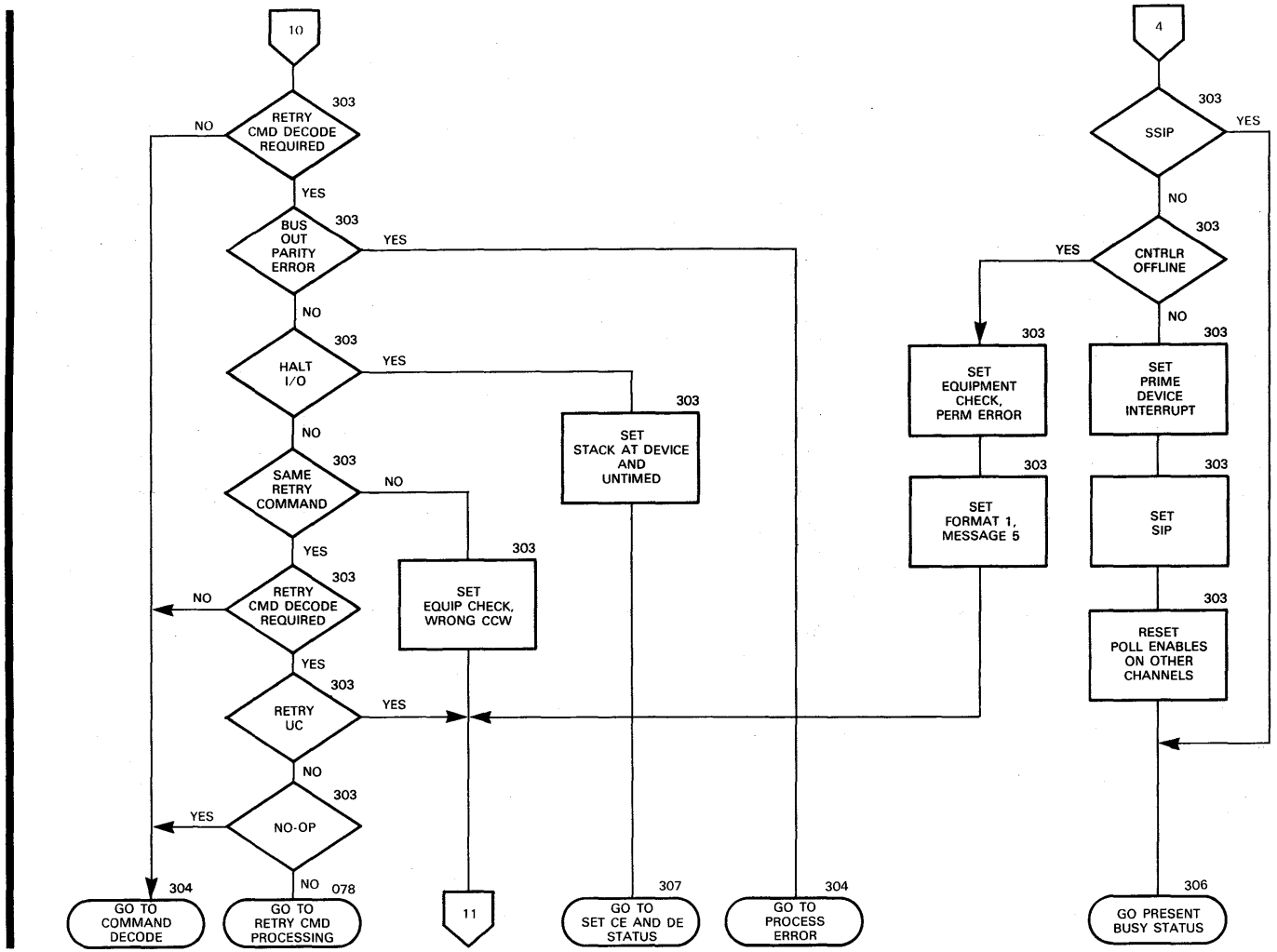


Figure 2-1A. Initial Selection Sequence, Block Diagram (5 of 6)

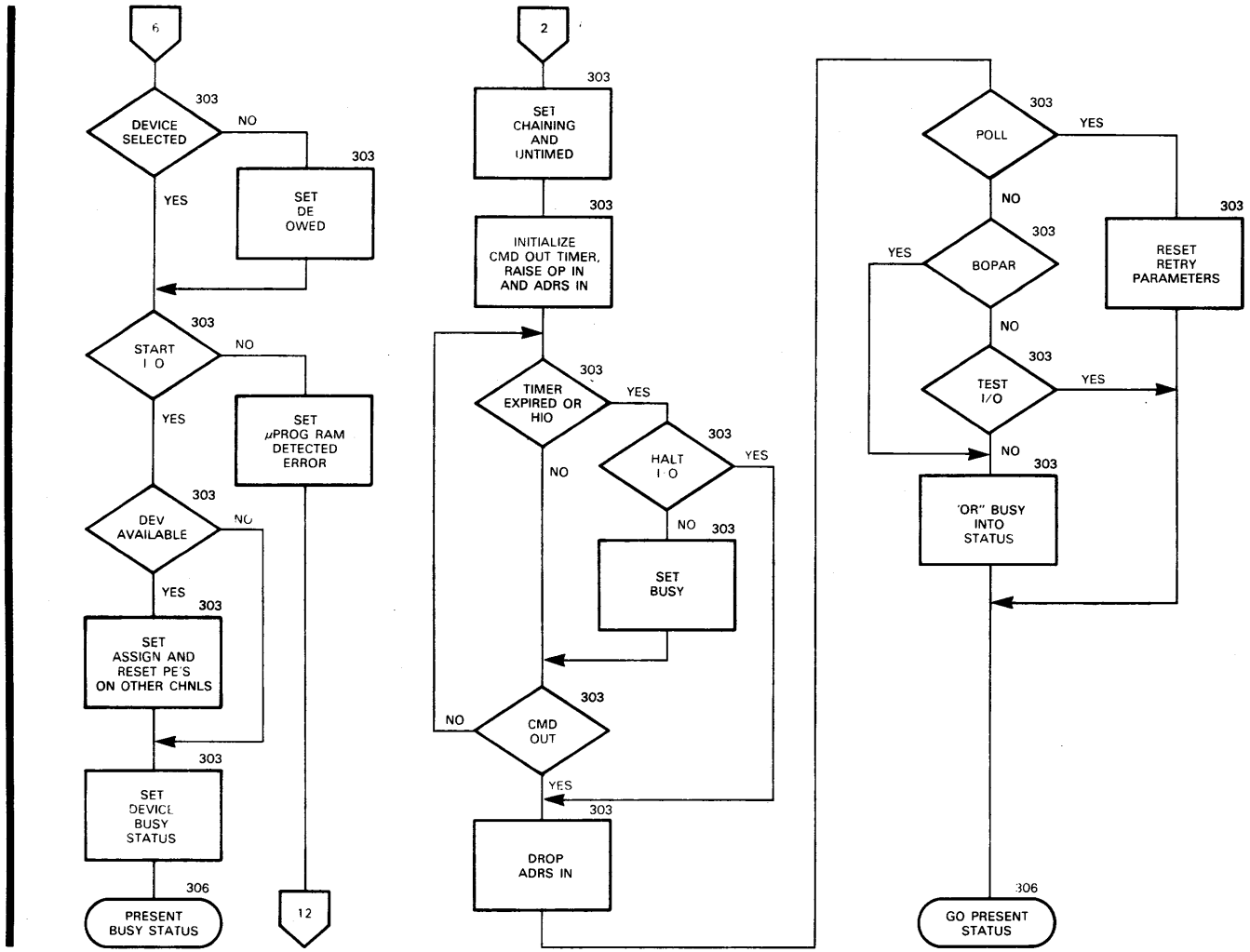


Figure 2-1A. Initial Selection Sequence, Block Diagram (6 of 6)

NO-OP (Figure 2-2)
Command Code 03 (hex)

An immediate command which causes no action at the addressed drive.

DATA ADDRESS—Not checked for validity, but should not exceed addressing capability.

FLAGS—SLI flag must be set to avoid an incorrect length indication.

COUNT—Must not be zero. A zero count sets the Program Check bit (bit 42) in the CSW.

INITIAL STATUS—Channel End and Device End presented

SPECIAL REQUIREMENTS—Indiscriminate usage must be avoided, since a *No-Op* resets orientation information causing all or part of record to be skipped. For example:

A *No-Op* inserted between *Read Count* and *Read Data* causes data of the following record to be skipped.

A *No-Op* inserted between a command which reads the data field of record n - 1 and a command which must process the Count field of record n may skip record n and process the Count field of record n + 1.

SEEK (Figure 2-3)
Command Code 07 (hex)

Transfers the six-byte seek address shown below to the SCU.

	ADDRESS BYTES (HEX)					
	BIN		CYLINDER		HEAD	
	BYTE 0	BYTE 1	BYTE 2	BYTE 3	BYTE 4	BYTE 5
3670	00 00	00 00	00 01	00 TO FF 00 TO 9A	00 00	00 TO 12 00 TO 12
3675	00 00 00 00	00 00 00 00	00 01 02 03	00 TO FF 00 TO FF 00 TO FF 00 TO 2E	00 00 00 00	00 TO 12 00 TO 12 00 TO 12 00 TO 12

The SCU selects the drive, moves access to the proper cylinder, and selects the proper head. Any access motion required is initiated after the seek address has been transferred. Limitations on address byte values are:

- Bytes 0, 1, and 4 must be zero.
- Bytes 2 and 3 together must not exceed 410 (dec) for 3670, or 814 (dec) for 3675.
- Byte 5 must not exceed 18 (dec).

Channel End is presented after the seek address has been transferred. Device End is presented with Channel End if no movement is required, or after access is positioned if movement is required.

DATA ADDRESS—Specifies main storage location of the seek address. Checked for both validity and correct parity by the SCU. If an invalid seek address or parity error is noted, the command is not executed; Unit Check, Channel End, and Device End are presented in ending status; and a subsequent sense command indicates either command reject (invalid condition) or bus-out parity error (parity error).

FLAGS—Can be used at discretion of the programmer.

COUNT—Must be six or greater. If less than six, command is not executed; Unit Check, Channel End, and Device End are presented in ending status; and a subsequent sense command indicates command reject.

INITIAL STATUS—Normally zero.

SPECIAL REQUIREMENTS—

- Command execution does not require preceding CCW.
- File mask must be set to allow seeks.

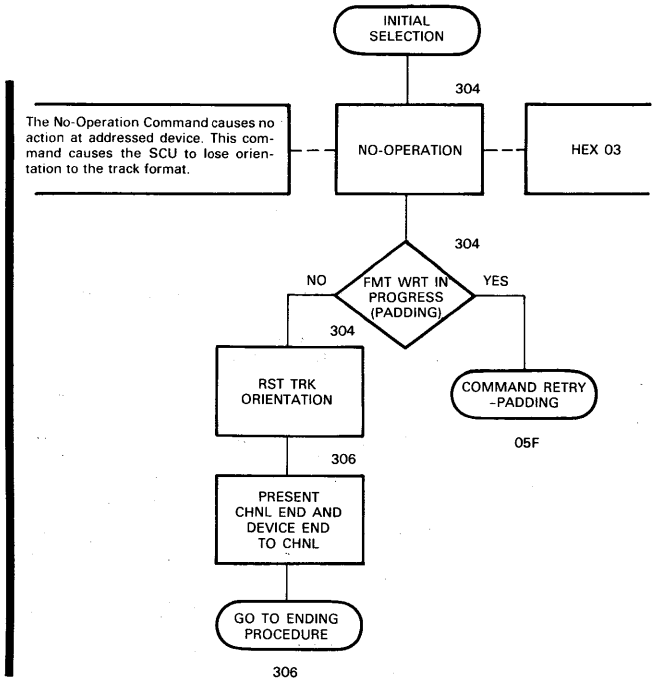


Figure 2-2. No-Op Flow Diagram

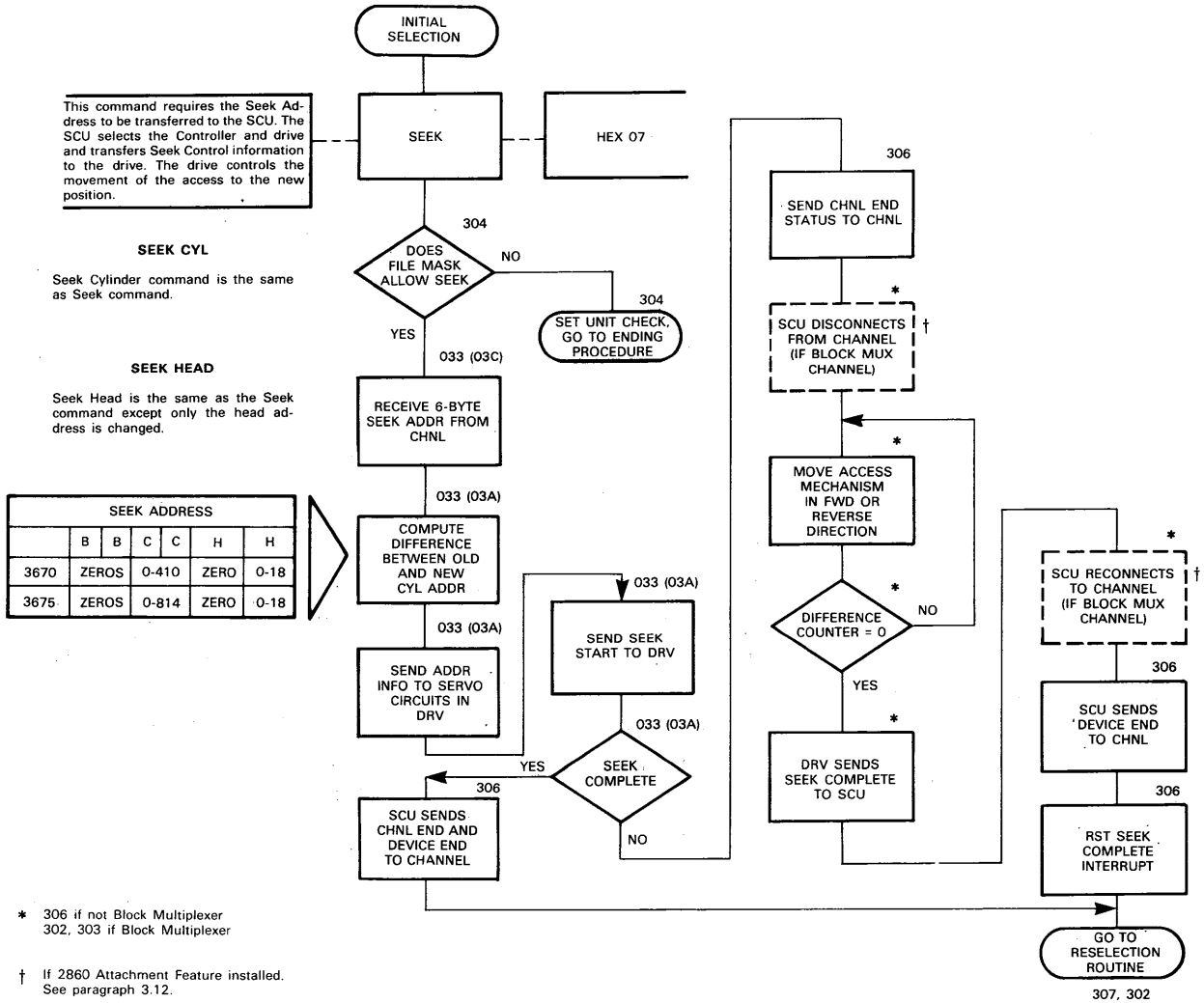


Figure 2-3. Seek Flow Diagram

SEEK CYLINDER (Figure 2-3)
Command Code 08 (hex)

Transfers the six-byte seek address shown below to the SCU.

	ADDRESS BYTES (HEX)					
	BIN		CYLINDER		HEAD	
	BYTE 0	BYTE 1	BYTE 2	BYTE 3	BYTE 4	BYTE 5
3670	00 00	00 00	00 01	00 TO FF 00 TO 9A	00 00	00 TO 12 00 TO 12
3675	00 00 00 00	00 00 00 00	00 01 02 03	00 TO FF 00 TO FF 00 TO FF 00 TO 2E	00 00 00 00	00 TO 12 00 TO 12 00 TO 12 00 TO 12

The SCU selects the drive, moves access to the proper cylinder, and selects the proper head. Any access motion required is initiated after the seek address has been transferred. Limitations on address byte values are:

- Bytes 0, 1, and 4 must be zero.
- Bytes 2 and 3 together must not exceed 410 (dec) for 3670 or 814 (dec) for 3675.
- Byte 5 must not exceed 18 (dec).

Channel End is presented after the seek address has been transferred. Device End is presented with Channel End if no movement is required, or after access is positioned if movement is required.

DATA ADDRESS—Specifies main storage location of the seek address. Checked for both validity and correct parity by the SCU. If an invalid seek address or parity error is noted, the command is not executed; Unit Check, Channel End, and Device End are presented in ending status; and a subsequent sense command indicates either command reject (invalid condition) or bus-out parity error (parity error).

FLAGS—Can be used at discretion of the programmer.

COUNT—Must be six or greater. If less than six, command is not executed; Unit Check, Channel End, and Device End are presented in ending status; and a subsequent sense command indicates command reject.

INITIAL STATUS—Normally zero.

SPECIAL REQUIREMENTS—

- Command execution does not require preceding CCW.
- File mask must be set to allow seeks.

SEEK HEAD (Figure 2-3)
Command Code 1B (hex)

Transfers the six-byte seek address shown below to the SCU.

	ADDRESS BYTES (HEX)					
	BIN		CYLINDER		HEAD	
	BYTE 0	BYTE 1	BYTE 2	BYTE 3	BYTE 4	BYTE 5
3670	00 00	00 00	00 01	00 TO FF 00 TO 9A	00 00	00 TO 12 00 TO 12
3675	00 00 00 00	00 00 00 00	00 01 02 03	00 TO FF 00 TO FF 00 TO FF 00 TO 2E	00 00 00 00	00 TO 12 00 TO 12 00 TO 12 00 TO 12

The SCU selects the drive and the proper head. Limitations on address byte values are:

- Bytes 0, 1, and 4 must be zero.
- Bytes 2 and 3 together must not exceed 410 (dec) for 3670 or 814 (dec) for 3675.
- Byte 5 must not exceed 18 (dec).

Channel End is presented after the seek address has been transferred. Device End is presented with Channel End if no movement is required, or after access is positioned if movement is required.

DATA ADDRESS—Specifies main storage location of the seek address. Checked for both validity and correct parity by the SCU. If an invalid seek address or parity error is noted, the command is not executed; Unit Check, Channel End, and Device End are presented in ending status; and a subsequent sense command indicates either command reject (invalid condition) or bus-out parity error (parity error).

FLAGS—Can be used at discretion of the programmer.

COUNT—Must be six or greater. If less than six, command is not executed; Unit Check, Channel End, and Device End are presented in ending status; and a subsequent sense command indicates command reject.

INITIAL STATUS—Normally zero.

SPECIAL REQUIREMENTS—

- Command execution does not require preceding CCW.
- File mask must be set to allow seeks.

SPACE COUNT (Figure 2-4)
Command Code 0F (hex)

Allows bypassing a defective count field on a track for recovering data in key and/or data fields following the defective field. Three bytes of information data are transferred from the channel. These bytes are interpreted by the SCU as the key length (first byte) and the data length (last two bytes) of the record to be recovered.

DATA ADDRESS—Specifies main storage location of the key and data lengths of record to be recovered.

FLAGS—Used at discretion of the programmer.

COUNT—Must be three or greater to transfer the requisite three bytes. If count is than three, the specified number of bytes is transferred and the value of the nontransferred bytes is assumed to be zero

CHAINING REQUIREMENTS—

- Cannot be chained from a Format Write or Erase.
- Must not be followed by a Write, Erase, or Set File Mask in the same chain.

Violation of these requirements will cause Channel End, Device End, and Unit Check to be presented to the channel.

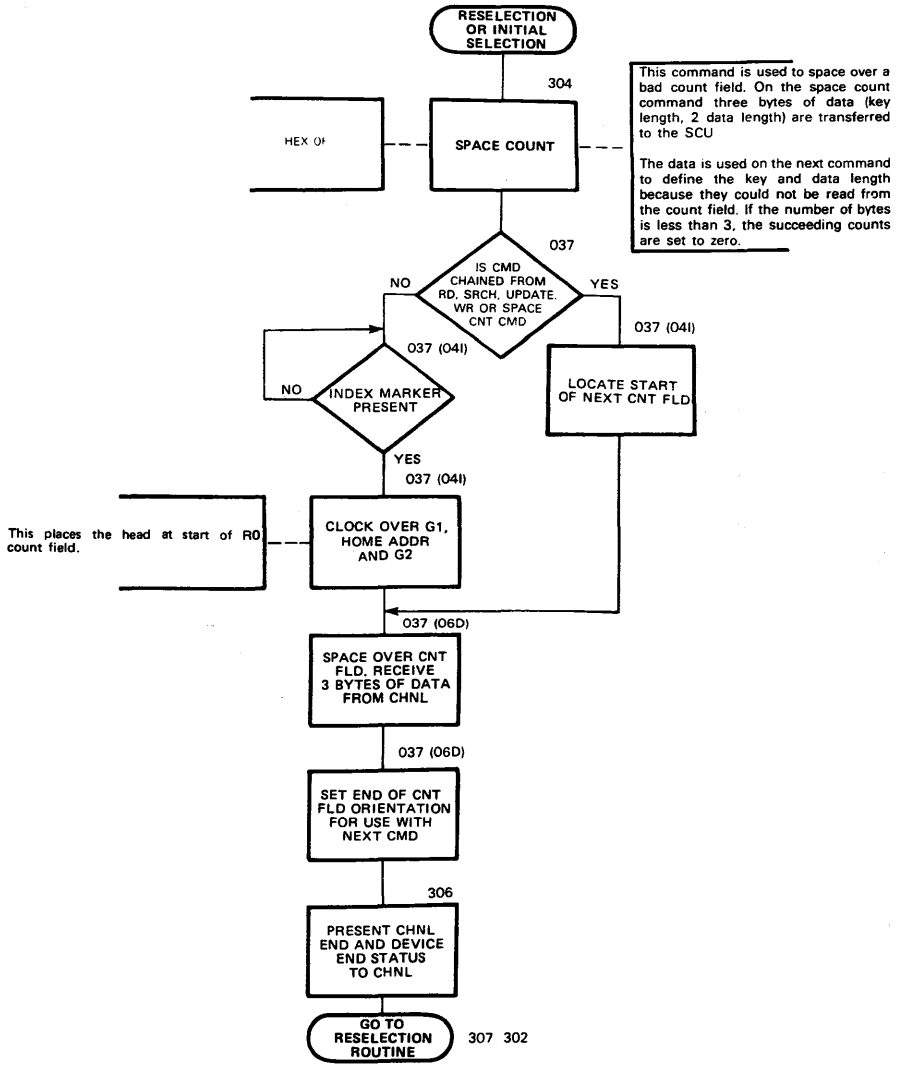


Figure 2-4. Space Count Flow Diagram

RECALIBRATE (Figure 2-5)
Command Code 13 (hex)

Causes addressed drive to seek to cylinder zero/head zero. Channel End presented in ending status. Device End presented when drive accesses to cylinder zero/head zero.

DATA ADDRESS—Not checked for validity, but should not exceed addressing capacity.

FLAGS—SLI flag must be set to avoid an incorrect length indication.

COUNT—Must not be zero. A zero count sets the Program Check bit (bit 42) in the CSW.

INITIAL STATUS—Normally zero. Not processed as an immediate command.

SPECIAL REQUIREMENTS—File mask must be set to allow seek commands.

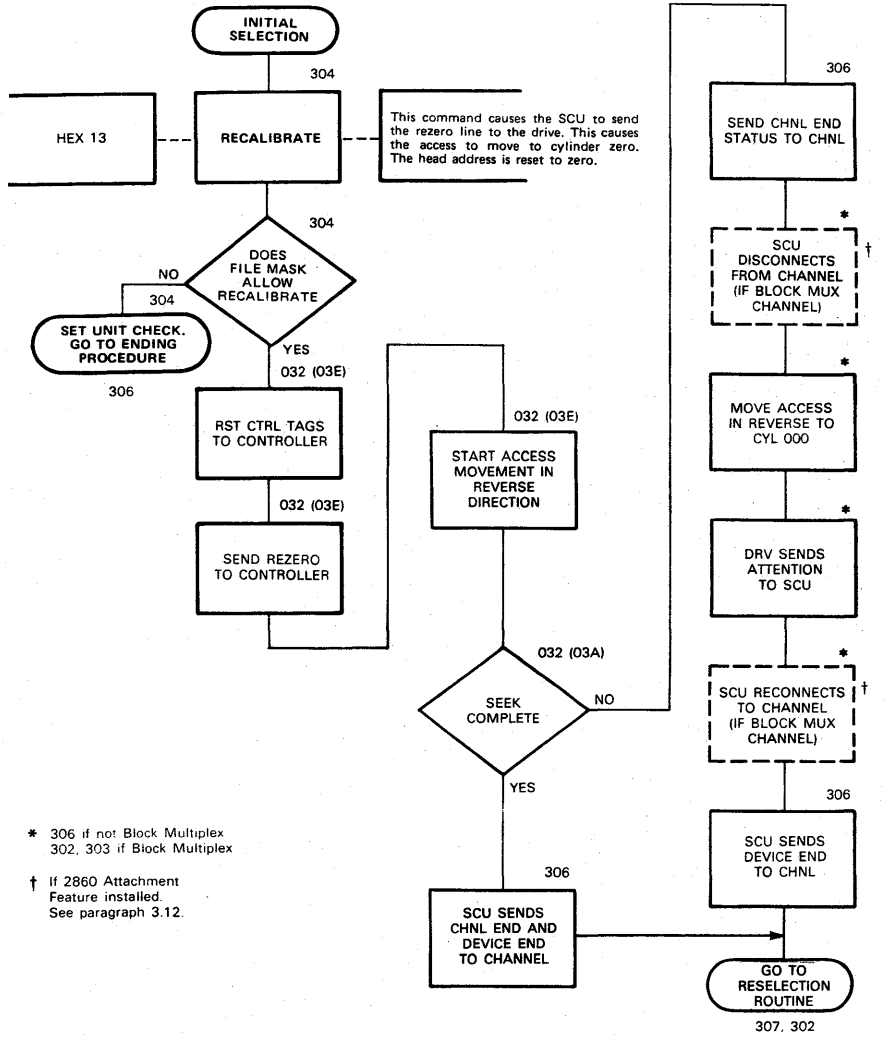


Figure 2-5. Recalibrate Flow Diagram

RESTORE (Figure 2-6)

Command Code 17 (hex)

Used primarily for compatibility with other direct-access storage devices. It causes no action to be performed but does result in loss of orientation. Channel End and Device End immediately follow initial status.

DATA ADDRESS—Not checked for validity, but must not exceed addressing capacity

FLAGS—SLI flag must be set to avoid an incorrect length indication

COUNT—Must not be zero. A zero count sets the Program Check bit (bit 42) in the CSW

INITIAL STATUS—Normally zero

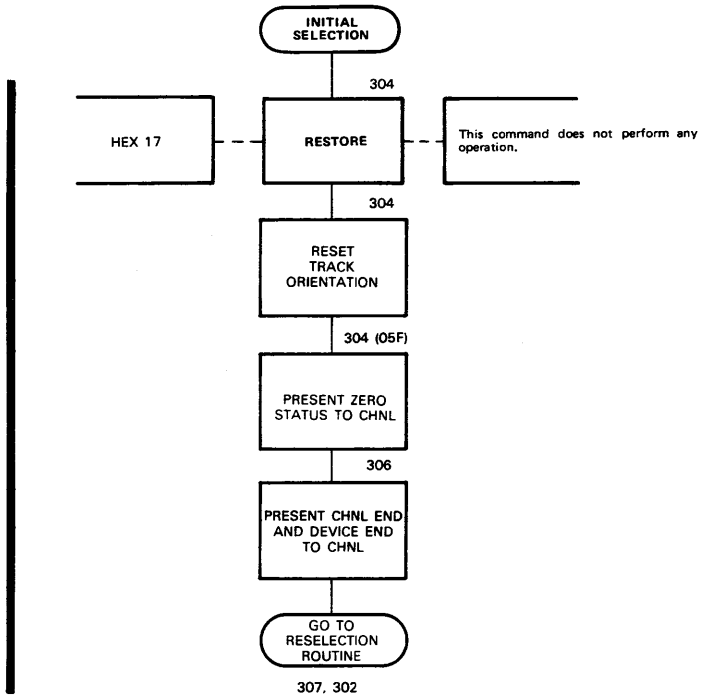


Figure 2-6. Restore Flow Diagram

SET FILE MASK (Figure 2-7)
Command Code 1F (hex)

Sets the write and seek masks which provide protection for 3672-based subsystem data and defines Command Retry–Program Control Interruption (PCI) interaction. Defined values of the mask byte and their function are listed below.

Bits 2 and 6 of the mask byte must be zero. If these bits are not zero, the mask byte is considered to be invalid and a Unit Check condition (CSW bit 38) is generated. A subsequent sense command indicates Command Reject.

Write or seek commands that violate the file mask are not executed. In these cases, Unit Check is presented in initial status, and either Command Reject (if a write command) or File Protected (if a seek command) is indicated by a subsequent sense command. Multitrack or overflow operations that violate the file mask indicate Unit Check and File Protected.

A system or selective reset resets the file mask to zero. A *Start I/O* instruction issued after a reset without a *Set File Mask* command permits seek and write commands (except *Write Home Address* and *Write Record Zero*).

Channel End and Device End are presented to the channel after the mask byte is transferred.

DATA ADDRESS—Specifies main storage location of mask byte.

FLAGS—Used at discretion of the programmer.

COUNT—One.

INITIAL STATUS—Unit Check is presented if the file mask is violated or more than one *Set File Mask* appears in a chain.

CHAINING REQUIREMENTS—Only one *Set File Mask* is permitted in a CCW chain. An attempt to issue more than one *Set File Mask* in a chain causes Unit Check in initial reset status, Command Reject to be noted in a subsequent sense command, and reset of the file mask to zero at the end of the chain.

Bit 0	1	2	3	4	5	6	7	Function:
0	0	0	0	0	0	0	0	Inhibit write HA and write R0.
0	1	0	0	0	0	0	0	Inhibit all write commands.
1	0	0	0	0	0	0	0	Inhibit all format write commands.
1	1	0	0	0	0	0	0	Permit all write commands.
0	0	0	0	0	0	0	0	Permit all seek commands.
0	0	0	0	1	0	0	0	Permit seek cylinder and seek head.
0	0	0	1	0	0	0	0	Permit seek head.
0	0	0	1	1	0	0	0	Inhibit all seek commands and head switching.
0	0	0	0	0	0	0	0	Inhibit diagnostic write command.
0	0	0	0	0	1	0	0	Permit diagnostic write commands.
0	0	0	0	0	0	0	0	Not PCI fetch mode.
0	0	0	0	0	0	0	1	PCI fetch mode. (The storage control presents UNIT CHECK if command retry is used to recover from ECC uncorrectable data errors.)

*Set to zero.

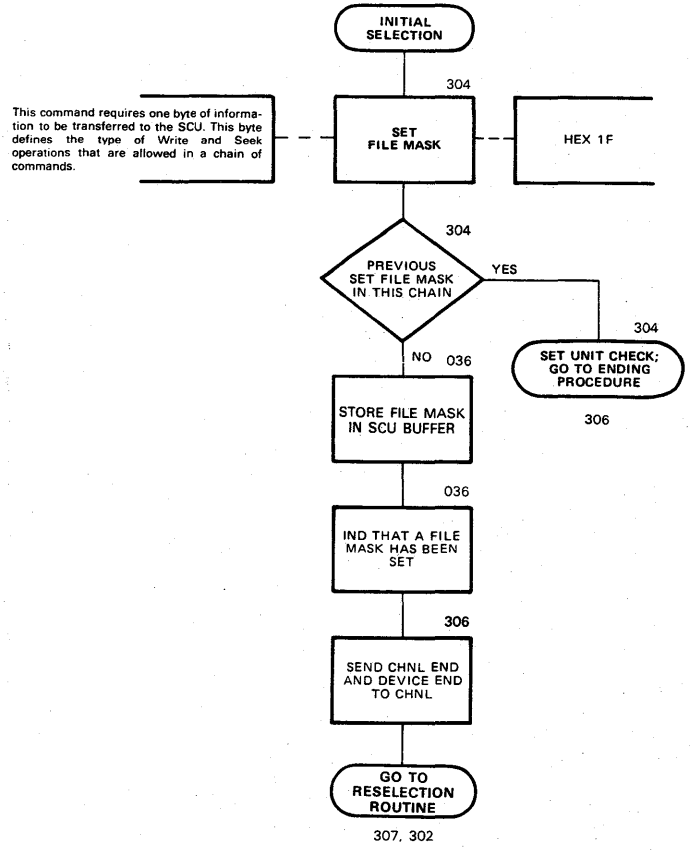


Figure 2-7. Set File Mask Flow Diagram

SET SECTOR (Figure 2-8)

Command Code 23 (hex)

Transfers a sector number (0-127 dec) from main storage to the SCU. This command is used on block multiplex channels to eliminate the need to maintain channel and SCU connection during drive rotational delays.

The sector number is checked for validity by the SCU. One of three validity conditions results: valid sector number (0-127), sector number > 127 and < 255, or sector number = 255.

Valid Sector Number (0-127)

SCU presents Channel End and disconnects. Device End is presented when sector number is reached and channel reconnects to continue chain. If reconnection does not occur, the SCU attempts to reconnect on subsequent revolutions. (All valid sector numbers are adjusted by the SCU to compensate for channel reselection delay.)

127 < Sector Number < 255

Channel, Device End, and Unit Check are presented in ending status. Command Reject indicated in a subsequent sense command.

Sector Number = 255

Command is treated as No-op. Channel End and Device End are presented in ending status. Track orientation is lost.

DATA ADDRESS—Specifies main storage location of desired sector number.

FLAGS—Used at discretion of the programmer.

COUNT—One.

SPECIAL REQUIREMENTS—

- *Set Sector* does not guarantee record orientation. Search commands must still be used for this function.
- Indiscriminate use of *Set Sector* with multitrack search may result in missing the desired record. A *Set Sector 0, Read Home Address, Search Multitrack* sequence will prevent this condition from occurring.

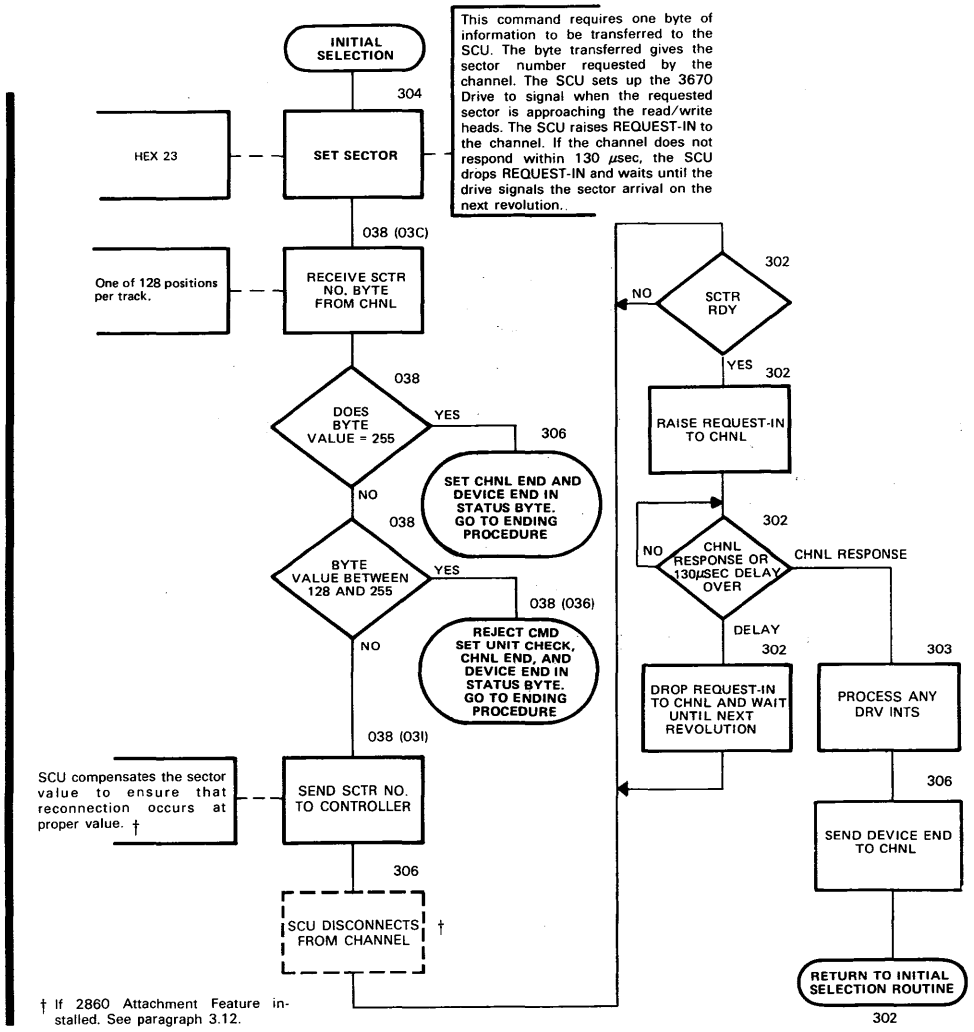


Figure 2-8. Set Sector Flow Diagram

DIAGNOSTIC LOAD (Figure 2-9)
Command Code 53 (hex)

Transfers a 668-byte block of data addressed by a control byte from the flexible disc file to writable control storage in the SCU. The data block transferred is a functional microprogram diagnostic test. During the transfer, the SCU is disconnected from the channel.

The control byte, specifying the diagnostic ID control number, is transferred from main storage to the SCU. Bits 0-5 of the control byte specify the flexible disc track address (0-49 dec) and bits 6 and 7 specify the sector pair (0-3).

The SCU checks the control byte for validity. A valid control byte presents Channel End in ending status. An invalid control byte sets Unit Check in the status byte and Command Reject in a subsequent sense byte.

Completion of the data transfer causes the SCU to request service and present Device End when polled.

DATA ADDRESS—Specifies main storage location of control byte.

FLAGS—Used at discretion of the programmer.

COUNT—One.

INITIAL STATUS—Normally zero.

SPECIAL REQUIREMENTS—

- Command execution allows any drive address to be used with the SCU address.
- *Read Diagnostic Status 1* command transfers the diagnostic test from writable control storage in the SCU to main storage.

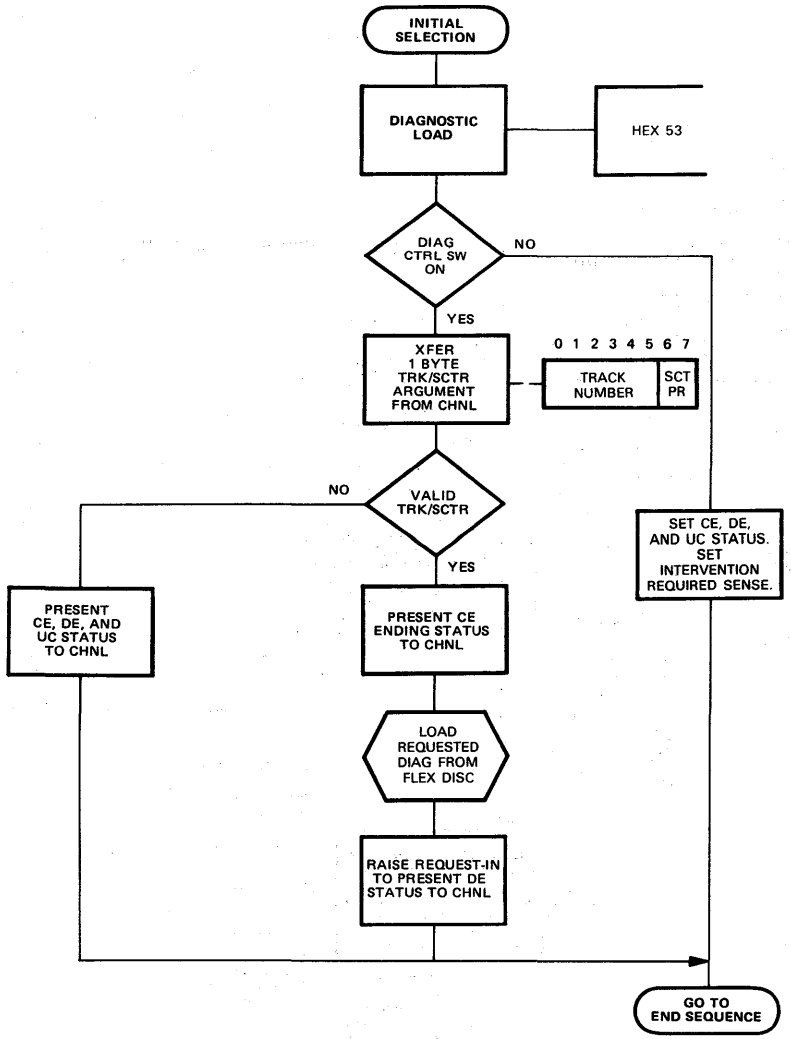


Figure 2-9. Diagnostic Load Flow Diagram

DIAGNOSTIC WRITE (Figure 2-10)
Command Code 73 (hex)

Transfers a 668-byte diagnostic test from main storage to writable control storage in the SCU. Upon completion of the data transfer, execution of the test begins. At completion of the test, a 16-byte error code message is stored in buffer storage of the SCU. The message is transferred from buffer storage to main storage by a subsequent *Read Diagnostic Status 1* command.

Channel End is presented after transfer of diagnostic test to the SCU. Device End is presented after the test is complete.

DATA ADDRESS—Specifies main storage location of the diagnostic test.

FLAGS—Used at discretion of the programmer

COUNT—Should be 668. If greater than 668, only 668 bytes are transferred. If less than 668, only the specified number of bytes are transferred; command is terminated; and Channel End, Device End, and Unit Check are presented in ending status.

INITIAL STATUS—Normally zero.

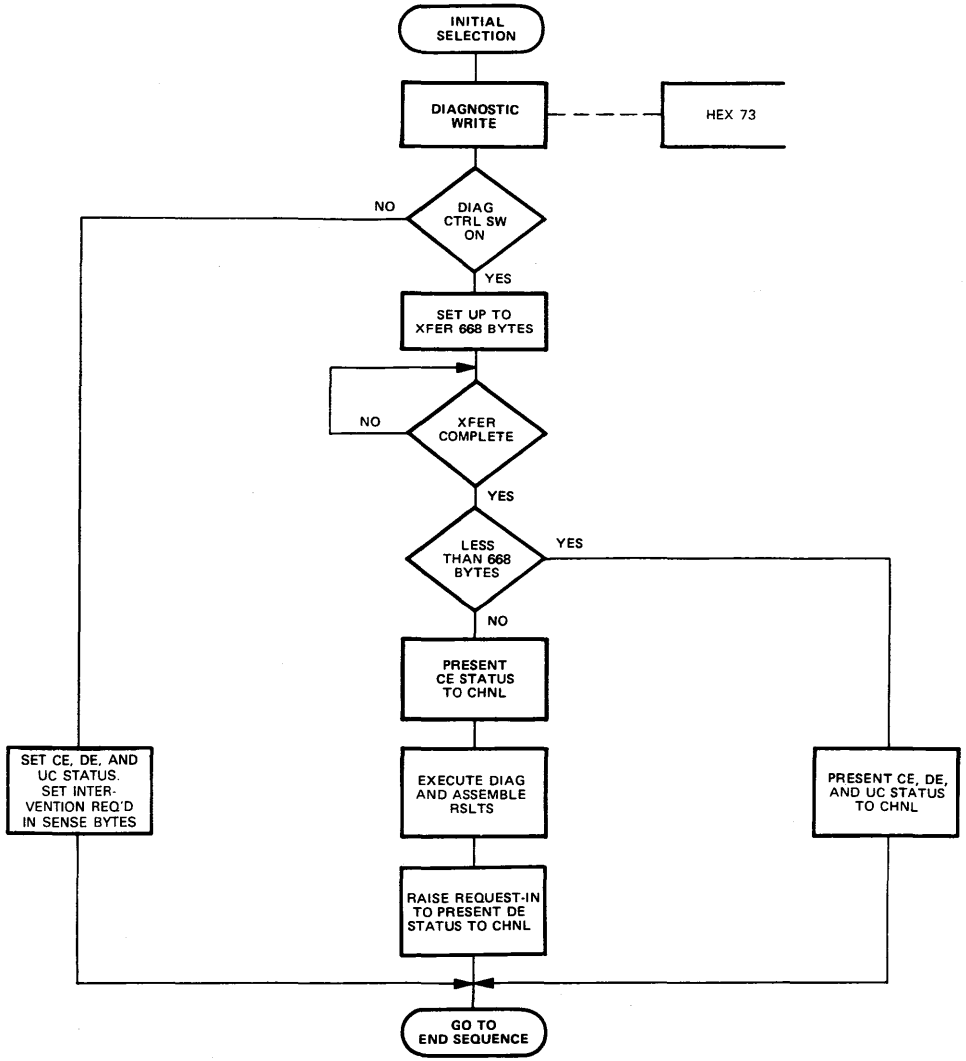


Figure 2-10. Diagnostic Write Flow Diagram

2.5 SENSE COMMANDS

2.5.1 Sense Byte Information

Conditions that occur during the execution of an instruction command sequence are reported to the using system program by the channel status word (CSW) and sense information. The sense information defines general and specific conditions in both the SCU, controller, and the drives that are not defined in the CSW. The sense information is contained in sense bytes which are transferred to the using system by the *Sense I/O* command. Upon receiving a *Sense I/O* command, the SCU transfers the 24 bytes of sense information to the using system. This command is normally executed following a status byte containing Unit Check (CSW bit 38).

Of the 24 bytes of sense information, the bit-by-bit meaning of messages reported in bytes 0 through 6 is preassigned. See Figure 2-11. The meaning of messages reported in bytes 8 through 23 depends on the format used for these bytes. This format is decoded in byte 7 (Bits 0 through 3).

The tables that follow this paragraph describe how sense bytes are assembled, the bit-by-bit meaning of each sense byte, and the formats and messages associated with sense bytes 8 through 23.

2.5.2 Sense Control Block

The sense control block (Table 2-2) is a set of two bytes which determine which bits are to be turned on in the sense assembly. The Sense Code byte sets bits in sense bytes 0, 1, and 2 as follows:

- Bit 0 (Write Inhibit) turns on bit 6 of sense byte 1.
- Bit 1 (Operation Incomplete) turns on bit 7 of sense byte 1 and defines a Restart command in sense byte 3.
- Bit 2 (Correctable) turns on bit 1 of sense byte 2.
- Bits 5, 6, and 7 specify other bits which may be turned on in addition to those described above. The byte in which a bit is turned on is specified by bits 3 and 4.

The Format/Message byte is formatted in the same manner as sense byte 7.

Upon entry, the sense bytes defined by the sense control block are stored by the subroutine at the starting location in G23.

2.5.3 Sense Byte Bit Definitions

Definitions of bits in all 24 sense bytes are listed in Tables 2-3, 2-4, and 2-5. Table 2-3 defines bits for sense bytes 0 through 6. Table 2-4 lists the definitions for format/message byte 7. Table 2-5 defines bits for bytes 8 through 23.

2.5.4 Assembling Sense Bytes 8 through 23

Sense bytes 8 through 23 are the second bytes to be assembled. These bytes are assembled according to the format specified in bits 0 through 3 of sense byte 7. The bit-by-bit meaning of sense bytes 8 through 23 for formats 0 through 6 are given in Table 2-5.

2.5.5 Command Descriptions

Each sense command is described on the following pages by means of a flow diagram and a narrative description.

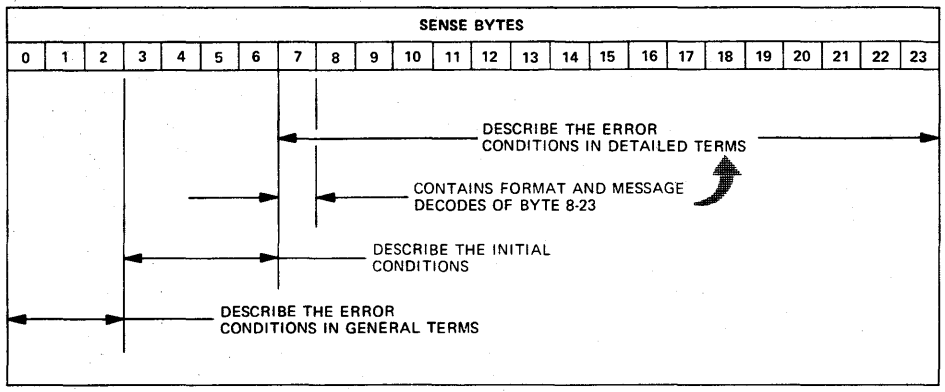


Figure 2-11. Sense Byte Definitions

TABLE 2-2. SENSE CONTROL BLOCK

BIT	SENSE CODE	FORMAT/MESSAGE
0	Write Inhibit	Bits 0 through 3 are decoded into formats 0 through 6. Formats 0 through 6 are defined as follows: 0 Message Only 1 Device Errors 2 SCU Errors 3 Selective Reset 4 ECC Uncorrectable 5 ECC Correctable 6 Usage/Error Count
1	Operation Incomplete	
2	Correctable	
3	Bit Arrangement: 01 10 11 00 — Byte Number: 0 1 2 0 and bit 0 of byte 1	
4		
5	Bit Position in Byte	Bits 4 through 7 are decoded into Message 0 through C. The meaning of the message depends upon the format decoded.
6		
7		

TABLE 2-5. SENSE BYTES 8-23 BIT DEFINITIONS (1 OF 2)

FORMAT 0—PROGRAMMING OR SYSTEM CHECK

SENSE BYTES 8-23
NOT USED - SET TO ZERO

FORMAT 1—DISC DRIVE EQUIPMENT CHECK

SENSE BYTE 8—MODULE STATUS

Bit 0	DI	Index error	ORDER 24 DO = /00
Bit 1		Offset active	
Bit 2		Seek incomplete	
Bit 3		Seek complete	
Bit 4		Online	
Bit 5		Attention	
Bit 6		Busy	
Bit 7		Record ready	

SENSE BYTE 9—MONITOR MODE

Bit 0	DI	Not used	ORDER 2C DO = /20
Bit 1		Diagnostic 4	
Bit 2		Diagnostic 2	
Bit 3		Diagnostic 1	
Bit 4		Not used	
Bit 5		Mode 4	
Bit 6		Mode 2	
Bit 7		Mode 1	

SENSE BYTE 10—MONITOR STATE

Bit 0	DI	Monitor state 8	ORDER 2C DO = /10
Bit 1		Monitor state 7	
Bit 2		Monitor state 6	
Bit 3		Monitor state 5	
Bit 4		Monitor state 4	
Bit 5		Monitor state 3	
Bit 6		Monitor state 2	
Bit 7		Monitor state 1	

SENSE BYTE 11—CHECK STATUS

Bit 0	DI	CE program stop	ORDER 2C DO = /08
Bits 1-3		Not used	
Bit 4		Interface check	
Bit 5		Monitor check	
Bit 6		Not used	
Bit 7		Drive command reject	

SENSE BYTE 12—SAFETY

Bit 0	DI	Select lock
Bit 1		Not used
Bit 2		Abnormal stop
Bit 3		Not used
Bit 4		Servo disable
Bit 5		Seek not enabled
Bit 6		Not servo data
Bit 7		Even cylinder

SENSE BYTE 13

Contents of DO Register

SENSE BYTE 14

Contents of DI Register

STATUS BYTE 15

Contents of CO Register

SENSE BYTE 16—CONTROLLER CHECK 16

Bit 0	Not used		ORDER 04 DO = /02
Bit 1	Write parity error		
Bit 2	Read parity error		
Bit 3	Bit ring error		
Bit 4	Write compensation error		
Bit 5	Data transfer control error		
Bit 6	Missing PLO pulses		
Bit 7	VFO phase error		

SENSE BYTE 17—CONTROLLER CHECK 17

Bit 0	ECC no input data		ORDER 04 DO = /01
Bit 1	ECC P0 or write error		
Bit 2	ECC P1 or P3 error		
Bit 3	ECC P2 error		
Bit 4	Sync out check		
Bit 5	PLO reorient counter check		
Bit 6	Gap counter check		
Bit 7	Gap control check		

SENSE BYTE 18—CONTROLLER CHECK 18

Bit 0	Controller check	MICRO-PROGRAM GENERATED
Bit 1	Select active check	
Bit 2	DI check	
Bit 3	Reserved	

Bits 4-7 define the following errors:

0000	Not used
0001	Tag valid missing (read/write)
0010	Normal end check end missing (control)
0011	Normal end missing (control)
0100	No index after 40 ms or solid index
0101	Unexpected status with check end
0110	Controller selection address check
0111	Preselection check
1000	Zero pattern alignment check
1001	Repetitive command overrun
1010	Drive interrupt during busy
1011	Drive status not as expected after Seek or Set Sector command
1100-1111	Not used

SENSE BYTE 19—CONTROLLER CHECK 19

Bit 0	Drive selection error		ORDER = 04 DO = /04
Bit 1	CTL-I tag bus check		
Bit 2	Device check		
Bit 3	CTL-I bus out check		
Bit 4	Write sense check		
Bit 5	Read or write valid check		
Bit 6	Device bus out check		
Bit 7	Controller bus in assembler check		

SENSE BYTES 20—DRIVE INOP 2

Bit 0	Write Overrun		ORDER = 2E DO = /12
Bit 1	AC Write		
Bit 2	Write Fail		
Bit 3	Subtractor Fail		
Bit 4	Multi Write		
Bit 5	Multi Head		
Bit 6	35V Reg. Fail		
Bit 7	Multi Write Data (3670) Pad Unsafe (3675)		

SENSE BYTE 21—DRIVE INOP 3

Bit 0	Write Offset		ORDER = 2E DO = /13
Bit 1	Write & No Fine Track		
Bit 2	Read & Write		
Bit 3	Short Write		
Bit 4	AC Write S.S.		
Bit 5	Add Mark S.S.		
Bit 6	Sink Fail		
Bit 7	Not used		

SENSE BYTE 22 AND 23—ERROR SYMPTOM CODE

FORMAT 2—SCU ERRORS

SENSE BYTE 8—CONTROL CHECK

Bit 0	ECR	Select Alert 1	MCM = /08
Bit 1		Select Active	
Bit 2		Sync In	
Bit 3		Unexpected End Check	
Bit 4		Normal End	
Bit 5		Check End	
Bit 6		Tag Valid	
Bit 7		Not Used	

SENSE BYTE 9—SERDES CHECK

Bit 0	ECR	DI Buffer Check	MCM = /18
Bit 1		Select Active Check	
Bit 2		Not Used	
Bit 3		CE Alert	
Bit 4		Channel Buffer Read Check	
Bit 5		Channel A/C Interface Check	
Bit 6		Channel B/D Interface Check	
Bit 7		Channel Transfer Error	

SENSE BYTE 10

Data Transfer Counter High Byte — MCM = /28

SENSE BYTE 11

Data Transfer Counter Low Byte — MCM = /38

SENSE BYTE 12
SET TO ZERO

SENSE BYTE 13

Contents of DO Register

SENSE BYTE 14

Contents of DI Register

SENSE BYTE 15

Contents of CO Register

SENSE BYTES 16-21
SET TO ZERO

SENSE BYTES 22-23—ERROR SYMPTOM CODE

FORMAT 3—SELECTIVE RESET

SENSE BYTE 8

Bit 0	MCK	Selective Reset	MCM = /03
Bit 1		Single Buffer Error	
Bit 2		Single WCS Error	
Bit 3		Failing Address Bit 0	
Bit 4		Failing Address Bit 1	
Bit 5		Failing Address Bit 2	
Bit 6		Failing Address Bit 3	
Bit 7		Failing Address Bit 4	

SENSE BYTE 9

Bit 0	MCK	Failing Address Bit 5	MCM = /04
Bit 1		Failing Address Bit 6	
Bit 2		Failing Address Bit 7	
Bit 3		Failing Address Bit 8	
Bit 4		Failing Address Bit 9	
Bit 5		Failing Address Bit 10	
Bit 6		Failing Address Bit 11	
Bit 7		Failing Address Bit 12	

SENSE BYTE 10

Bit 0	MCK	Not Used	MCM = /10
Bit 1		Multi-WCS Error	
Bit 2		A-Bus	
Bit 3		B-Bus	
Bit 4		T-Bus	
Bit 5		Branch	
Bit 6		IMPL	
Bit 7		ALU	

TABLE 2-5. SENSE BYTES 8-23 BIT DEFINITIONS (2 OF 2)

SENSE BYTE 11	FORMAT 4—DATA CHECKS NOT PROVIDING DISPLACEMENT INFORMATION		FORMAT 5—DATA CHECKS PROVIDING DISPLACEMENT INFORMATION	FORMAT 6—USAGE/ERROR COUNTS	
Bit 0 BR Multiplex Error Bit 1 PS Error Bit 2 OP Code Translation Error Bit 3 Subroutine Error Bit 4 WCS Error Bit 5 Multiplexer Buffer Error Bit 6 Buffer ECC Error Bit 7 Buffer Write T-Bus Error MCK MCM = /01	SENSE BYTE 8—CYLINDER (1) High-order cylinder byte of last seek address	BUFFER LOC 13B	SENSE BYTE 8—CYLINDER (1) High-order cylinder byte of last seek address	SENSE BYTES 8-11—BYTES READ These four bytes provide an accumulated count of the number of bytes processed by the controller in read operations.	DRIVE DATA WORD 3
	SENSE BYTE 9—CYLINDER (2) Low-order cylinder byte of last seek address		SENSE BYTE 9—CYLINDER (2) Low-order cylinder byte of last seek address	SENSE BYTES 12 AND 13—CORRECTABLE DATA CHECKS These two bytes provide an accumulated count of the number of ECC correctable data checks which have been recorded.	DRIVE DATA WORD 4
SENSE BYTE 12	SENSE BYTE 10—HEAD (1) High-order head byte of last seek address		SENSE BYTE 10—HEAD (1) High-order head byte of last seek address	SENSE BYTES 14 AND 15—UNCORRECTABLE DATA CHECKS These two bytes identify the number of uncorrectable data checks which have been recorded.	
Bit 0 Failing P1 Bit Bit 1 Failing P2 Bit Bit 2 Failing P3 Bit Bit 3 Failing P4 Bit Bit 4 Failing P5 Bit Bit 5 Failing P6 Bit Bit 6 Failing P7 Bit Bit 7 Failing P8 Bit MCK MCM = /02	SENSE BYTE 11—HEAD (2) Low-order head byte of last seek address		SENSE BYTE 11—HEAD (2) Low-order head byte of last seek address		
	SENSE BYTE 12—RECORD Record number of record in error	RO/C	SENSE BYTE 12—RECORD Record number of record in error	SENSE BYTES 16 AND 17—SEEKS These two bytes provide a count of the number of access motions initiated by the channel.	
SENSE BYTE 13	SENSE BYTE 13—SECTOR Sector number of record in error	BUFFER LOC 138 BYTES 0 AND 1	SENSE BYTE 13—SECTOR Sector number of record in error	SENSE BYTE 18 Bit 0 If zero, bytes 20-23 contain channel A and B information; if one, channel C and D information. Bits 1-7 Not Used	DRIVE DATA WORD 5
Contents of CHF Register					
SENSE BYTE 14	SENSE BYTE 14—OFFSET Amount of offset used to recover from error		SENSE BYTE 14—ACCESS OFFSET Amount of offset used to recover from error		
Conents of RWC Register	SENSE BYTE 15—RETRIES Number of retries required to recover from error	BUFFER LOC 138 BYTE 3			
SENSE BYTE 15				SENSE BYTE 19—SEEK ERRORS Identifies the total number of seek error which were successfully retried by the storage control.	
Bit 0 GA Error Bit 1 GB Error Bit 2 GC Error Bit 3 GD Error Bit 4 FD Read Error Bit 5 FD Seek Error Bit 6 FD Not Ready Bit 7 Write Bus Error MCK MCM = /05	SENSE BYTE 16—SOURCE DRIVE IDENTIFICATION Bits 0 and 1 Identifies the controller that was used to record the data in which the error occurred Bits 2-7 Identifies the disc drive that was used to record the data in which the error occurred.	BUFFER LOC 137	SENSE BYTES 15-17—RESTART DISPLACEMENT Specifies the number of bytes processed by the storage control to end of data field in error		
		BUFFER LOC 138 BYTE 2	SENSE BYTES 18 AND 19—ERROR DISPLACEMENT Displacement of first byte in error relative to end of the data field where error occurred	SENSE BYTE 20 AND 21—COMMAND AND DATA OVERRUN, CHANNEL A OR C Provides a count of the number of command overruns (byte 20) or data overruns (byte 21) which were retried by the storage control for channel A or C.	BUFFER LOC 127 AND 128
		BUFFER LOC 12D			
		BUFFER LOC 12E	SENSE BYTES 20-22—ERROR PATTERN Contain high, middle, and low error pattern bytes used for error correction function.	SENSE BYTE 22 AND 23—COMMAND AND DATA OVERRUN, CHANNEL B OR D Provides a count of the number of command overruns (byte 22) or data overruns (byte 23) which were retried by the storage control for channel B or D.	
SENSE BYTES 16-21 SET TO ZERO	SENSE BYTES 17-21 NOT USED—SET TO 0		SENSE BYTE 23 Bits 0-6 Not used—set to 0 Bit 7 Channel truncation		
SENSE BYTES 22 AND 23 ERROR SYMPTOM CODE	SENSE BYTES 22 AND 23 ERROR SYMPTOM CODE				

SENSE I/O (Figure 2-12)
Command Code 04 (hex)

Transfers 24 bytes of sense information from the SCU to the channel. This information describes unit check status, current status of the device that performed the operation, and system error recovery information.

Channel End and Device End are presented after sense bytes are transferred.

DATA ADDRESS—Specifies main storage location where sense bytes are to be transferred.

FLAGS—Used at discretion of the programmer

COUNT—24

INITIAL STATUS—Normally zero.

SPECIAL REQUIREMENTS—Unit Check should always be followed by a Sense command whether or not sense information is used; otherwise, expected future interrupts may not occur and some I/O access paths may be unavailable.

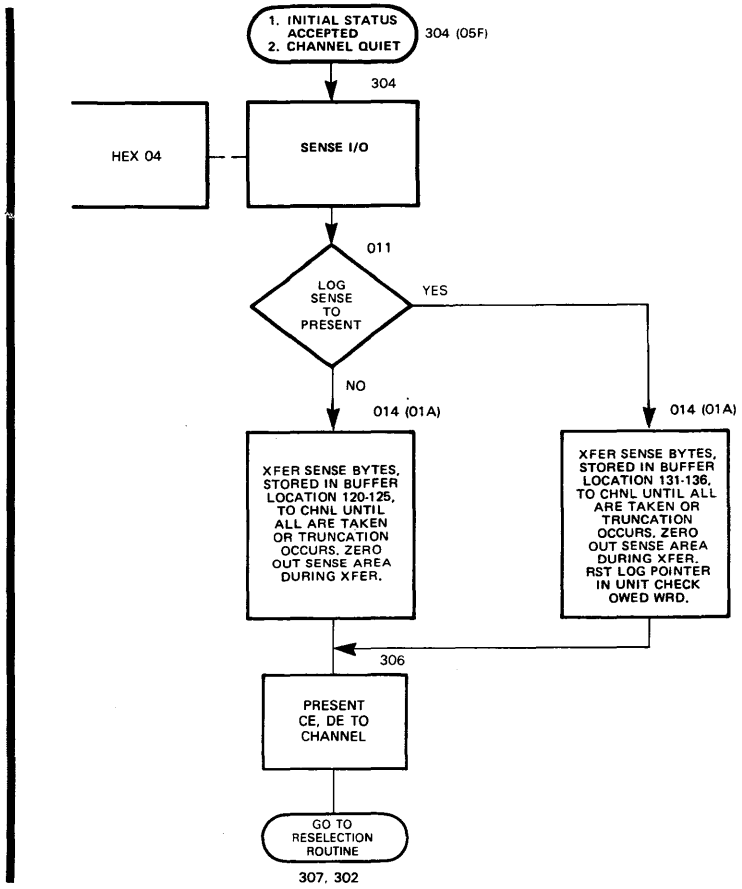


Figure 2-12. Sense I/O Flow Diagram

TEST I/O (Figure 2-13)
Command Code 00 (hex)

Transfers one status byte from the SCU to the channel. This byte is normally zero except when stacked or pending status is presented for the SCU or disc drive. Then, the status byte records status of the SCU and disc drive on the channel.

DATA ADDRESS—Specifies main storage location where status byte is to be transferred.

FLAGS—Not used.

COUNT —1.

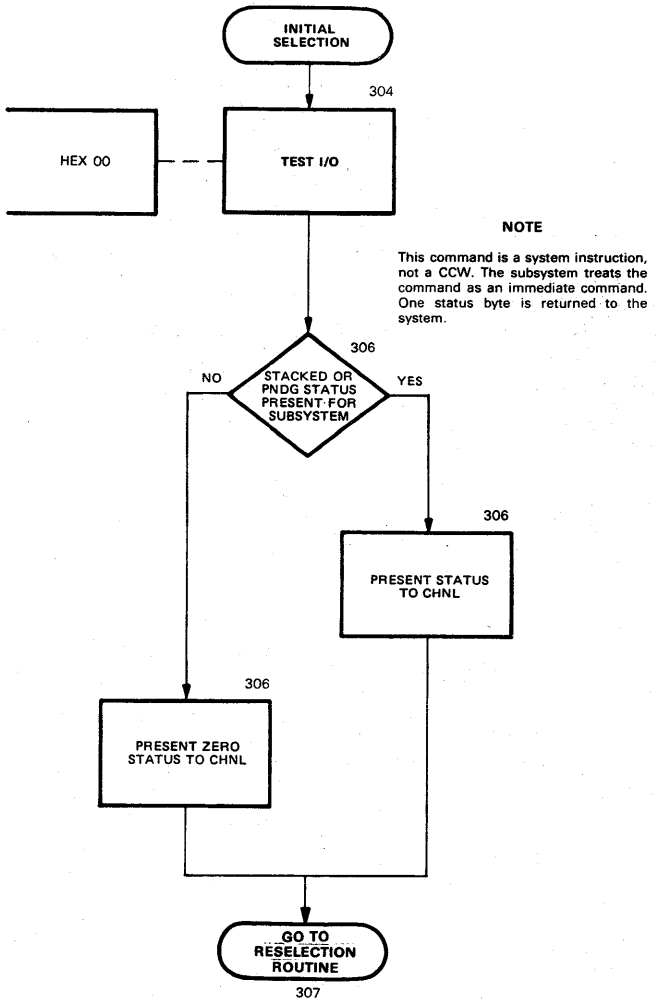


Figure 2-13. Test I/O Flow Diagram

READ AND RESET BUFFERED LOG (Figure 2-14)
Command Code A4 (hex)

Transfers 24 bytes of usage or error information from the SCU to the channel. This information, generated and available when their respective counters overflow, pertains to the SCU addressed by the *Start I/O* instruction and the drive identified in sense byte 4. The counters are reset after the data transfer.

Channel End and Device End are presented after the data transfer.

DATA ADDRESS—Specifies main storage location of first error byte or usage information.

FLAGS—Used at discretion of the programmer.

COUNT—24.

INITIAL STATUS—Normally zero.

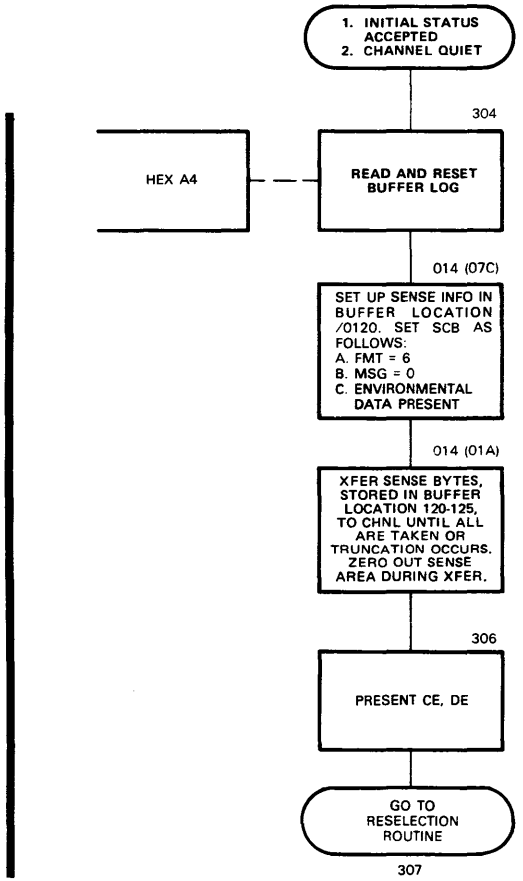


Figure 2-14. Read and Reset Buffer Log Flow Diagram

READ DIAGNOSTIC STATUS 1 (Figure 2-15)
Command Code 44 (hex)

This command may perform either of two functions, depending on whether it follows a *Diagnostic Write* or a *Diagnostic Read* command. If it follows a *Diagnostic Write,* the command transfers a 16-byte error code message from buffer storage in the SCU to main storage. For this type of transfer, the CCW Count field should specify 16. Channel End and Device End are presented after the transfer.

If it follows a *Diagnostic Load,* the command transfers a 668-byte diagnostic test from buffer storage in the SCU to main storage. For this type of transfer, the CCW Count field should specify 668. Channel End and Device End are presented after the transfer.

DATA ADDRESS—Specifies main storage location where data accumulated during prior *Diagnostic Load* or *Diagnostic Write* is to be stored.

FLAGS—Used at discretion of the programer.

COUNT—16 or 668, depending on type of transfer.

INITIAL STATUS—Normally zero.

SPECIAL REQUIREMENTS—*Diagnostic Load* or *Diagnostic Write* must precede *Read Diagnostic Status 1*; otherwise, 16 bytes of data are transferred from buffer storage in the SCU which normally contain the error message.

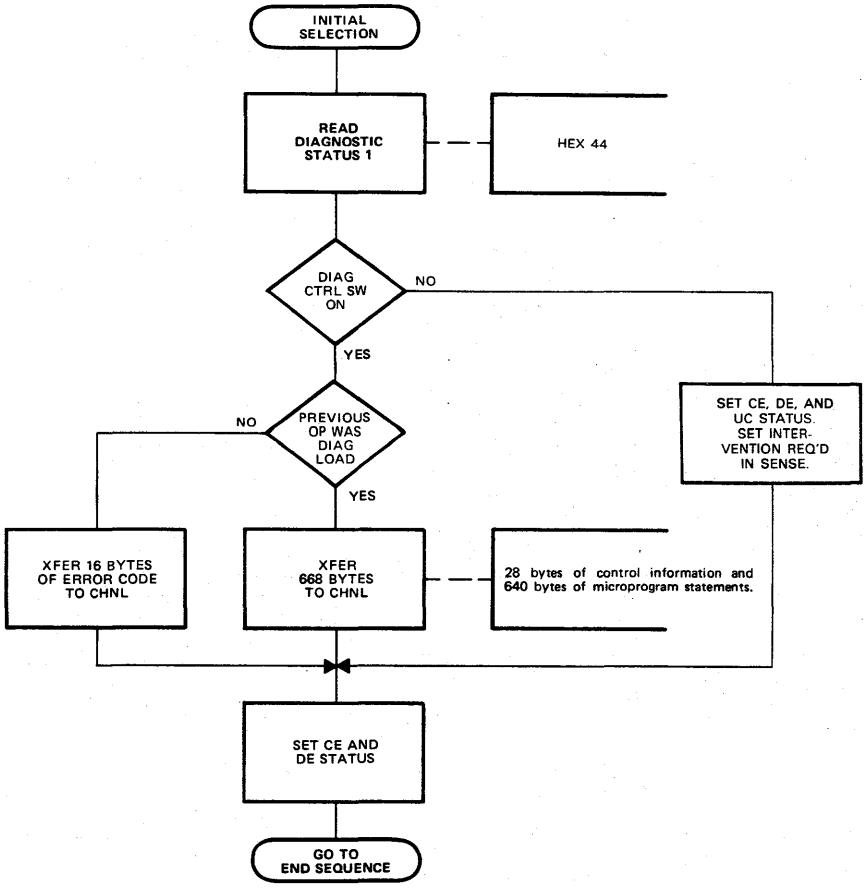


Figure 2-15. Read Diagnostic Status 1 Flow Diagram

DEVICE RESERVE (Figure 2-16)
Command Code B4 (hex)

Reserves the addressed drive to the channel issuing the command. Reservation is maintained until either a *Device Release* or a system reset is performed by the channel. In addition, 24 bytes of sense information are transfered to main storage across the channel.

Normal busy conditions cause a command reject and set the CSW Busy bit. Abnormal file status conditions (file unsafe, off-line, and so forth) do not halt command execution.

Channel End and Device End are presented after sense byte transfer.

DATA ADDRESS—Specifies main storage location where sense bytes are to be transferred.

FLAGS—Used at discretion of the programmer

COUNT—24.

INITIAL STATUS—Normally zero.

SPECIAL REQUIREMENTS—*Device Reserve* may not be executed if a *Set File Mask* precedes command in the same chain.

Violation of this requirement will cause command to be rejected, set CSW Unit Check bit, and Command Reject to be noted in a subsequent sense command.

DEVICE RELEASE (Figure 2-16)
Command Code 94 (hex)

Terminates reservation of the drive address reserved by *Device Reserve*. In addition, 24 bytes of sense information are transferred to main storage across the channel.

Normal busy conditions cause a command reject and set the CSW Busy bit. Abnormal file status conditions (file unsafe, off-line, and so forth) do not halt command execution.

Channel End and Device End are presented after sense byte transfer.

DATA ADDRESS—Specifies main storage location where sense bytes are to be transferred.

FLAGS—Used at discretion of the programer.

COUNT—24.

INITIAL STATUS—Normally zero.

SPECIAL REQUIREMENTS—*Device Release* may not be executed if a *Set File Mask* precedes command in the same chain.

Violation of this requirement will cause command to be rejected set CSW Unit Check bit, and Command Sense to be noted in a subsequent sense command.

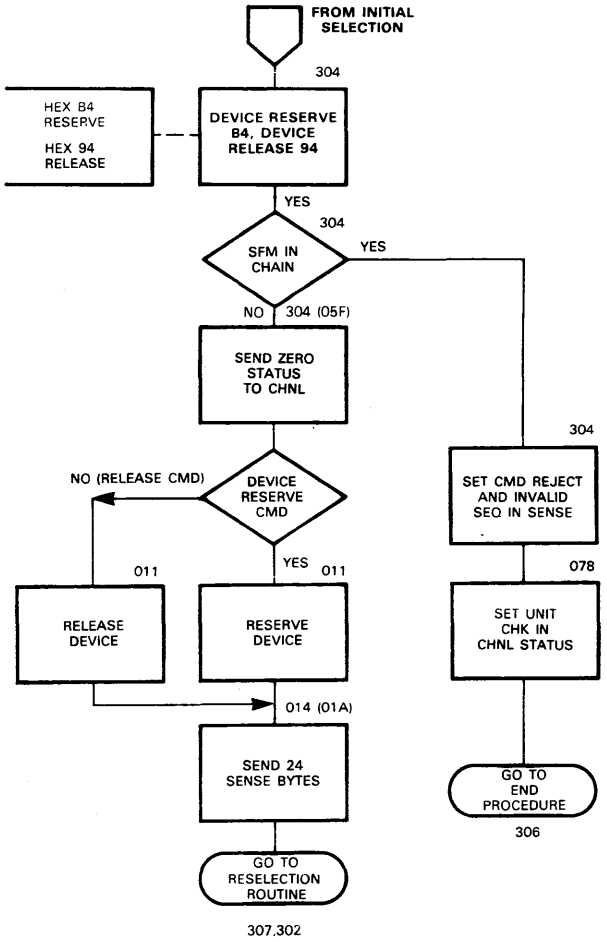


Figure 2-16. Device Reserve and Device Release Flow Diagrams

2.5.6 Transfer In-Line Diagnostics to SCU

2.5.6.1 INTRODUCTION

To provide maximum facility availability, the SCU can execute diagnostic tests on a drive concurrent with normal system operations on the remaining drives. The diagnostic tests are transferred from the system diagnostic library in main storage by a Transfer In-Line Diagnostics to SCU operation. This mode of operation allows the FSR to diagnose and repair most drive failures while the facility continues to operate other attached drives.

After the diagnostic is run, the error code is returned to the system. To permit temporary residence for a specific diagnostic test, the SCU provides a transient block of 128 words of control storage. The transient area is loaded under control of the On Line Test Executive Program (OLTEP). The *Diagnostic Write* command loads a selected test into control storage and instructs the SCU to execute the test. This loading and execution can also be initiated from the FE panel. After the test, error message information or test results are transferred from the SCU to main storage by a *Read Diagnostic Status 1* command. If the FE panel is used, the test results are displayed by the FE panel indicators.

The following text material is keyed to the block diagram of Figure 2-17.

2.5.6.2 SYSTEM

- Executes a *Start I/O* instruction which addresses a *Diagnostic Write* command.
- After receiving Device End status, executes a *Start I/O* instruction which addresses a *Read Diagnostic Status 1* command.

2.5.6.3 CHANNEL

- Transfers address and command information to the selected subsystem.
- Transfers status of subsystem to the system.
- Transfers diagnostic data to the subsystem.
- Transfers error code from subsystem to the system.

2.5.6.4 CHANNEL INTERFACE

- Controls timing of data to and from the channel.
- Checks parity of data transferred to the SCU.

2.5.6.5 STORAGE CONTROL UNIT (SCU)

- Decodes the *Diagnostic Write* command.
- Sets up and controls the transfer of data from Channel Interface to the Control Storage diagnostic buffer area.
- Sends Channel End to the channel.
- Runs in-line diagnostic.
- Stores 16-byte error code in the Control Storage buffer area if errors are found.
- Sends Device End status to the channel.
- Decodes *Read Diagnostic Status 1* command.
- Transfers error code from the control storage to the channel on *Read Diagnostic Status 1* command.
- Sends Channel End and Device End status to the channel.

2.5.6.6 CONTROL STORAGE

- Stores main microprogram to control *Diagnostic Write* and *Read Diagnostic Status 1* commands.
- Stores diagnostic data from system in a buffer area.
- Stores error code for any errors found.

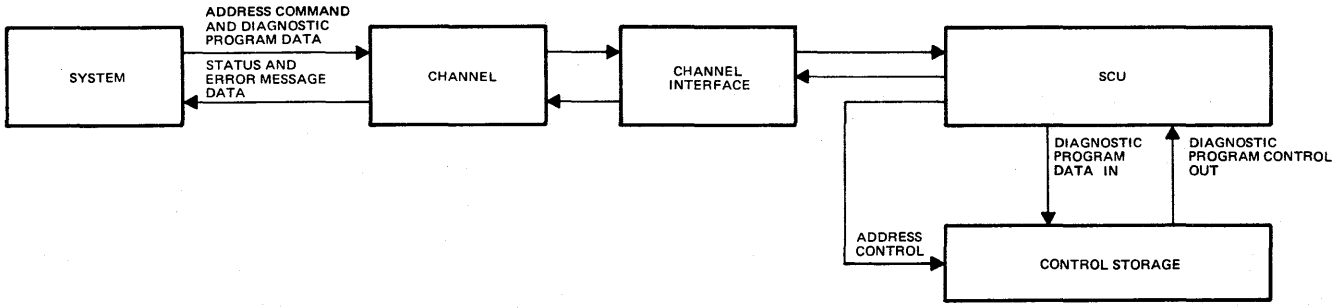


Figure 2-17. Transfer In-Line Diagnostics to SCU Block Diagram

2.6 READ COMMANDS

2.6.1 Operation Block Diagram

Read commands transfer data from the selected disc pack to the using system. These commands can operate in multitrack mode, indicated by setting bit 0 of the command code to one. Multi-track operations allow the SCU to automatically switch heads at index time without a *Seek Head* command.

The *Read Count, Key,* and *Data*; *Read Key* and *Data*; and *Read Data* commands can operate on overflow records. Overflow records allow data records of more than one track in length. Overflow operations are indicated by a bit in the flag byte.

Data checks may occur during execution of a read command. If such checks are detected, the SCU uses ECC to correct the error if correctable, or Command Retry if not correctable.

The following text material is keyed to the read operation block diagram shown in Figure 2-18.

2.6.1.1 SYSTEM

The system executes the *Start I/O* instruction which addresses a channel command word (CCW) containing a read command.

2.6.1.2 CHANNEL

- Executes the CCW to transfer the read command to the SCU.
- Transfers the address of the selected data to the SCU.
- Transfers data from the SCU to the using system.

2.6.1.3 CHANNEL INTERFACE

- Controls the timing of the transfer of data and control information between the channel and the SCU.
- Checks parity of data transferred to the SCU.

2.6.1.4 MICROPROCESSOR

- Decodes the read command.
- Selects the addressed drive head.
- Checks the status of the drive and transfers the status to the channel.
- Sets up the read tags to the controller.
- Controls the byte-by-byte transfer of read data from the controller.
- Monitors incoming tags for error indicators.
- Transfers ending status to the channel.
- Indicates retry status to the channel if error is a type that can be retried, and sets up to relocate record.
- Recognizes gap configurations.

2.6.1.5 CONTROLLER INTERFACE

- Transfers read control information to the selected drive.
- Transfers status of the selected drive to the Microprocessor.
- Controls the serial-by-byte transfer of read data from the serializer/deserializer (SERDES) to the controller interface.
- Receives the serial data from the drive.
- Recognizes gaps and synchronizes to the data.
- Changes serial-by-bit drive data to serial-by-byte data.
- Sends data to the ECC and the Microprocessor.
- Uses PLO pulses to maintain synchronization when not reading.

2.6.1.6 READ CIRCUIT

- Sets up the drive to read data from the disc.
- Changes currents from the read head into data pulses to SERDES.
- Provides the SCU with status of the drive.

2.6.1.7 READ/WRITE HEAD

- Changes magnetic flux from pack to read current signals.
- Selected head provides data pulses.
- PLO head provides PLO pulses.
- Servo head and circuits hold access at selected track.

2.6.1.8 PACK

The pack contains bits of information on the magnetic disc coating.

2.6.2 Command Descriptions

Each read command is described in the following pages by means of a narrative description and a flow diagram.

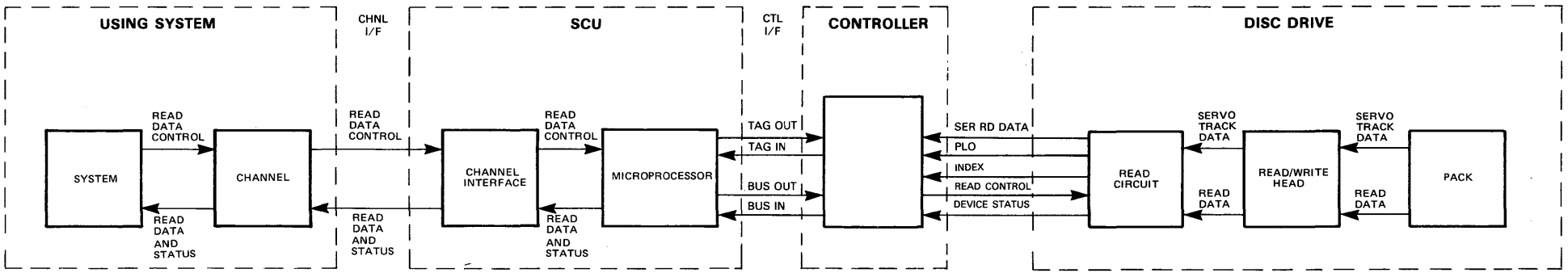


Figure 2-18. Read Operation Block Diagram

READ DATA (Figure 2-19)

Command Code: 06 (hex) single track
86 (hex) multi-track

Transfers the data area of a record from the disc to main storage. The data read can be one of the following:

- Data area of record read by *Search ID* or *Search Key* from which *Read Data* is chained.
- Data area of record read by *Read Count* from which *Read Data* is chained.
- Data area of record following next count area on the track (excluding R0).

Correction codes following each data area are used to check for data validity. A parity bit is added to each byte prior to its transfer to the channel.

Channel End and Device End are presented to the channel upon completion of the correction code check of the data area.

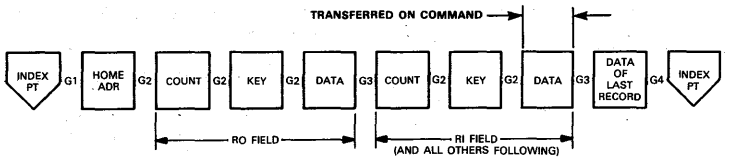
A data overrun or check condition, if detected, initiates an SCU recovery attempt by Command Retry. If Command Retry is not used or is unsuccessful when an overrun or check condition occurs, Channel End, Device End, and Unit Check are presented to the channel. (Command Retry is not used if a correctable data error; i.e., error burst of 11 bytes or less, is detected in the data area.)

DATA ADDRESS—Specifies main storage location where first data byte is to be transferred.

FLAGS—Used at discretion of the programmer.

COUNT—Specifies number of bytes to be read.

INITIAL STATUS—Normally zero.



G1, G2, AND G3: FIXED LENGTH GAPS
G4: VARIABLE LENGTH GAP

READ KEY AND DATA (Figure 2-19)

Command Code: 0E (hex) single track
8E (hex) multi-track

Transfers the key and data areas of a record from the disc to main storage. The key and data read can be one of the following:

- Key and data area of record read by *Search ID* from which *Read Key and Data* is chained.
- Key and data areas of record ready by *Read Count* from which *Read Key and Data* is chained.
- Key and data areas of record following next count area on the track (excluding R0).

Correction codes following each key and data area are used to check for data validity. A parity bit is added to each byte prior to its transfer to the channel.

Channel End and Device End are presented to the channel upon completion of the correction code check of the areas.

A data overrun or check condition, if detected, initiates an SCU recovery attempt by Command Retry. If Command Retry is not used or is unsuccessful when an overrun or check condition occurs, Channel End, Device End, and Unit Check are presented to the channel. (Command Retry is not used if a correctable data error; i.e., error burst of 11 bytes or less, is detected in the data area.)

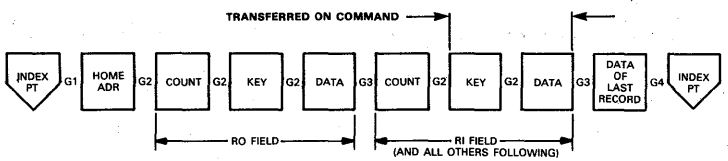
A key length of zero causes the command to operate as a *Read Data* command.

DATA ADDRESS—Specifies main storage location where first byte of key data is to be transferred.

FLAGS—Used at discretion of the programmer.

COUNT—Specifies the number of key and data area bytes to be read.

INITIAL STATUS—Normally zero.



G1, G2, AND G3: FIXED LENGTH GAPS
G4: VARIABLE LENGTH GAP

READ COUNT, KEY, AND DATA (Figure 2-19)
Command Code. 1E (hex) single track
9E (hex) multi-track

Transfers the next record encountered on the track from the disc to main storage (excluding Record Zero).

Correction codes following each area are used to check for data validity. A parity bit is added to each byte prior to its transfer to the channel.

Channel End and Device End are presented to the channel upon completion of the correction code of the areas.

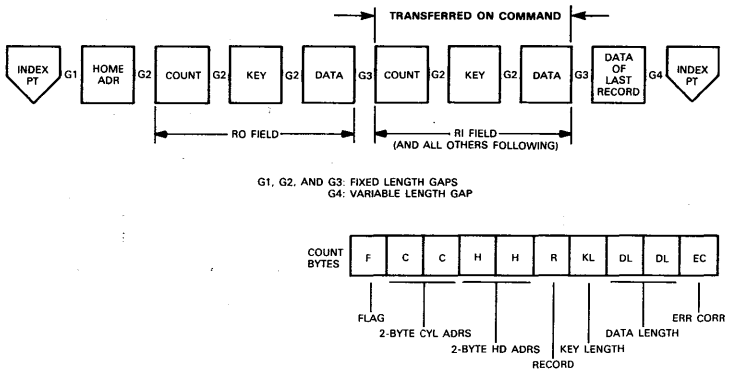
A data overrun or check condition, if detected, initiates an SCU recovery attempt by Command Retry. If Command Retry is not used or is unsuccessful when an overrun or check condition occurs, Channel End, Device End, and Unit Check are presented to the channel.

DATA ADDRESS—Specifies main storage location where first byte of count data is to be transferred.

FLAGS—Used at discretion of the programmer.

COUNT—Specifies the number of key and data area bytes to be read.

INITIAL STATUS—Normally zero.



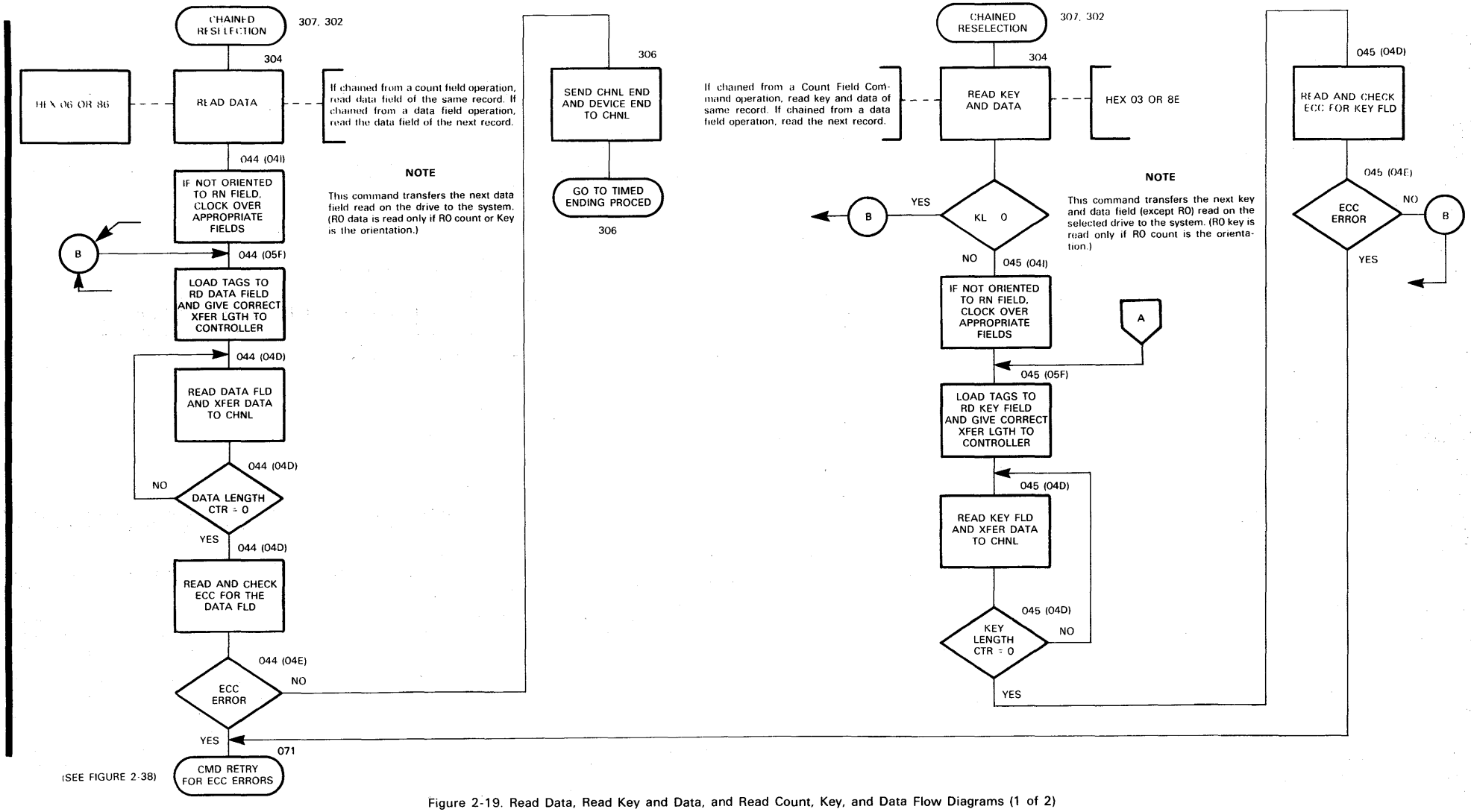


Figure 2-19. Read Data, Read Key and Data, and Read Count, Key, and Data Flow Diagrams (1 of 2)

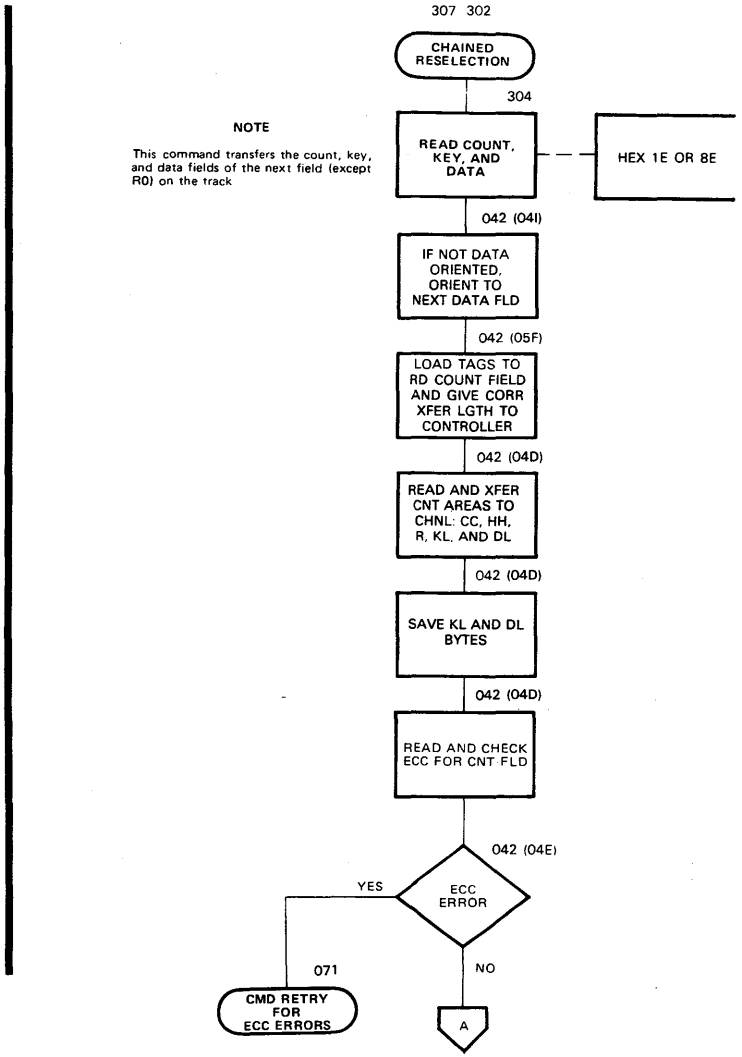


Figure 2-19. Read Data, Read Key and Data, and Read Count, Key, and Data Flow Diagrams (2 of 2)

READ RECORD ZERO (Figure 2-20)

Command Code: 16 (hex) single track
96 (hex) multi-track

Transfers count, key, and data areas of Record Zero (R0) from the disc to main storage. Data transfer of the R0 count area is initiated by the SCU, which searches for index and reads gap 1, home address, and gap 2.

Correction codes following each area are used to check for data validity. A parity check bit is added to each byte prior to its transfer to the channel.

Channel End and Device End are presented to the channel upon completion of the correction code check of the areas.

A data overrun or check condition, if detected, initiates an SCU recovery attempt by Command Retry. If Command Retry is not used or is unsuccessful, when an overrun or check condition occurs, Channel End, Device End, and Unit Check are presented to the channel. (Command Retry is not used if a correctable data error; i.e., error burst of 11 bytes or less, is detected in the data area.)

DATA ADDRESS—Specifies main storage location where first byte of R0 count data is to be transferred.

FLAGS—Used at discretion of the programmer.

COUNT—Specifies number of count, key, and data bytes to be read.

INITIAL STATUS—Normally zero.

SPECIAL CHARACTERISTICS—Command execution begins immediately if *Read R0* is chained from a *Search Home Address* or *Read Home Address*. In these cases, the SCU will not search for index.

READ COUNT (Figure 2-20)

Command Code: 12 (hex) single track
92 (hex) multi-track

Transfers the eight bytes (CC, HH, R, KL, and DL) of the next count area encountered on the track (excluding R0) from the disc drive to main storage.

Correction codes following each count area are used to check for data validity. A parity check bit is added to each byte prior to its transfer to the channel.

Channel End and Device End are presented to the channel upon completion of the correction code check of the areas.

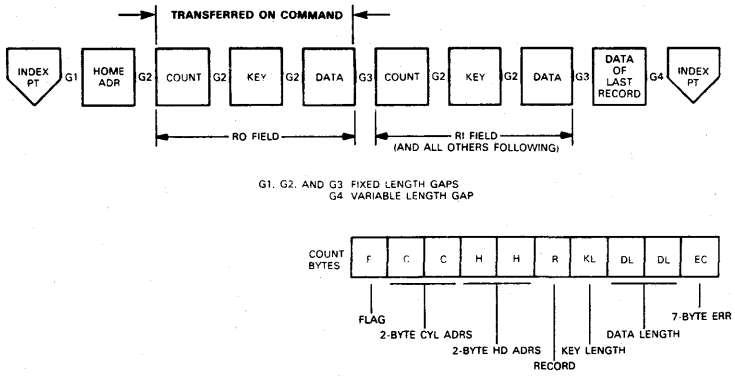
A data overrun or check condition, if detected, initiates an SCU recovery attempt by Command Retry. If Command Retry is not used or is unsuccessful when an overrun or check condition occurs, Channel End, Device End, and Unit Check are presented to the channel.

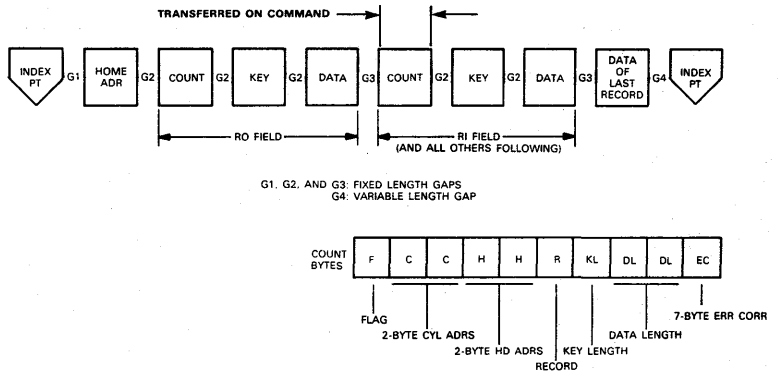
DATA ADDRESS—Specifies main storage location where first byte of count data is to be transferred.

FLAGS—Used at discretion of the programmer.

COUNT—Eight.

INITIAL STATUS—Normally zero.





READ HOME ADDRESS (Figure 2-20)
Command Code: 1A (hex) single track
9A (hex) multi-track

Transfers the F, CC, and HH bytes of the home address to main storage.

Correction codes following each home address area are used to check for data validity. A parity check bit is added to each byte prior to its transfer to the channel.

Channel End and Device End are presented to the channel upon completion of the correction code check of the areas.

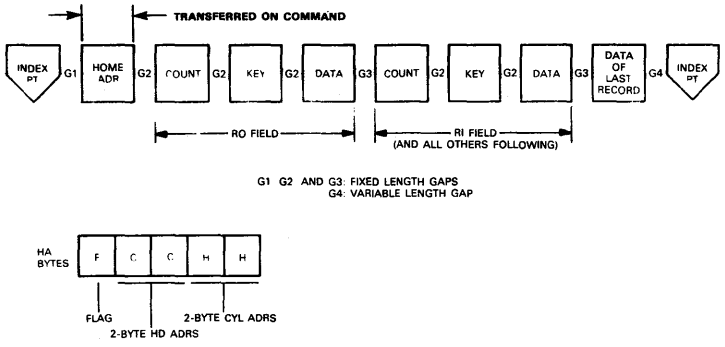
A data overrun or check condition, if detected, initiates an SCU recovery attempt by Command Retry. If Command Retry is not used or is unsuccessful when an overrun or check condition occurs, Channel End, Device End, and Unit Check are presented to the channel.

DATA ADDRESS—Specifies main storage location where first byte of home address is to be stored.

FLAGS—Used at discretion of the programmer.

COUNT—Five.

INITIAL STATUS—Normally zero.



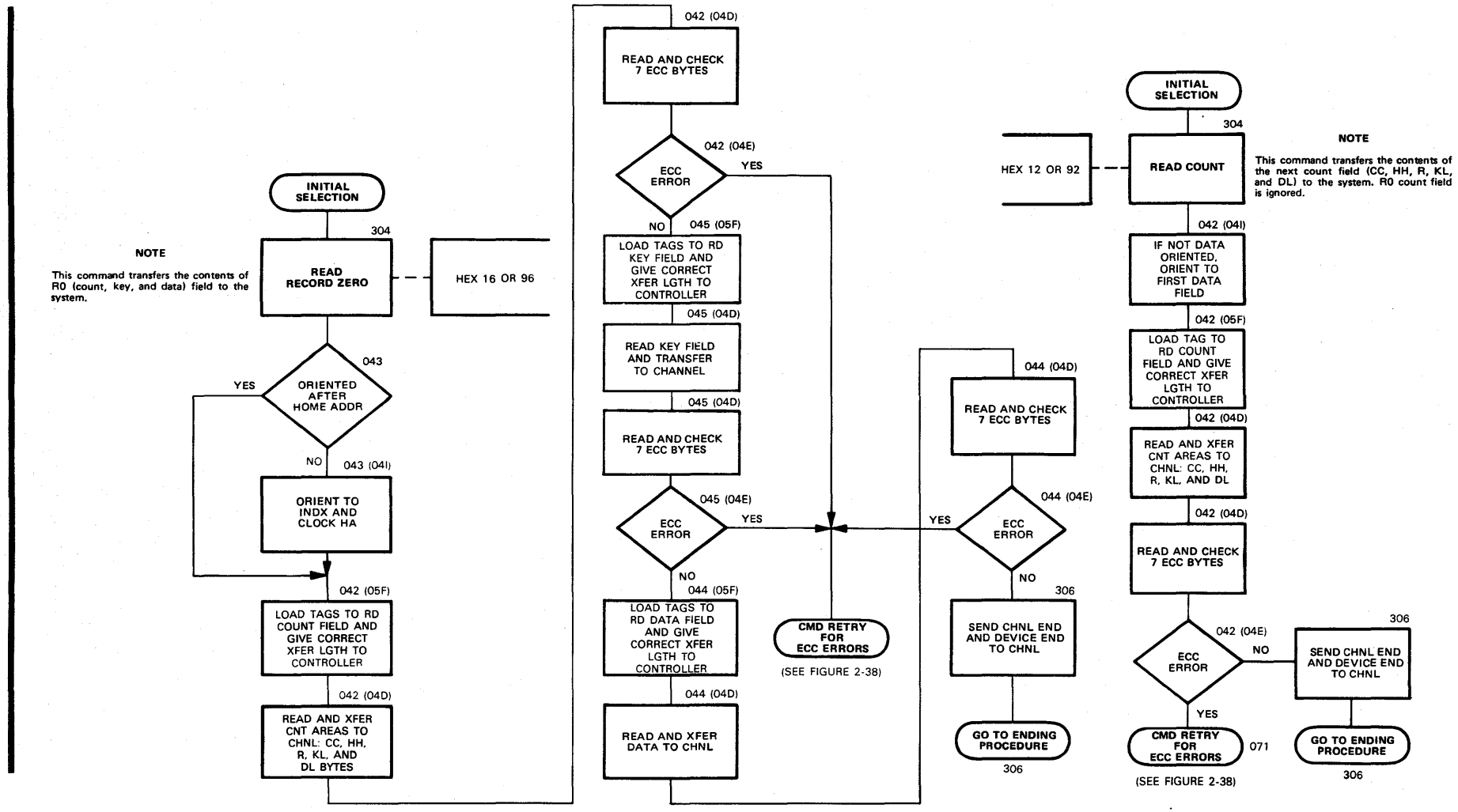


Figure 2-20. Read Record Zero, Read Count, and Read Home Address Flow Diagram (1 of 2)

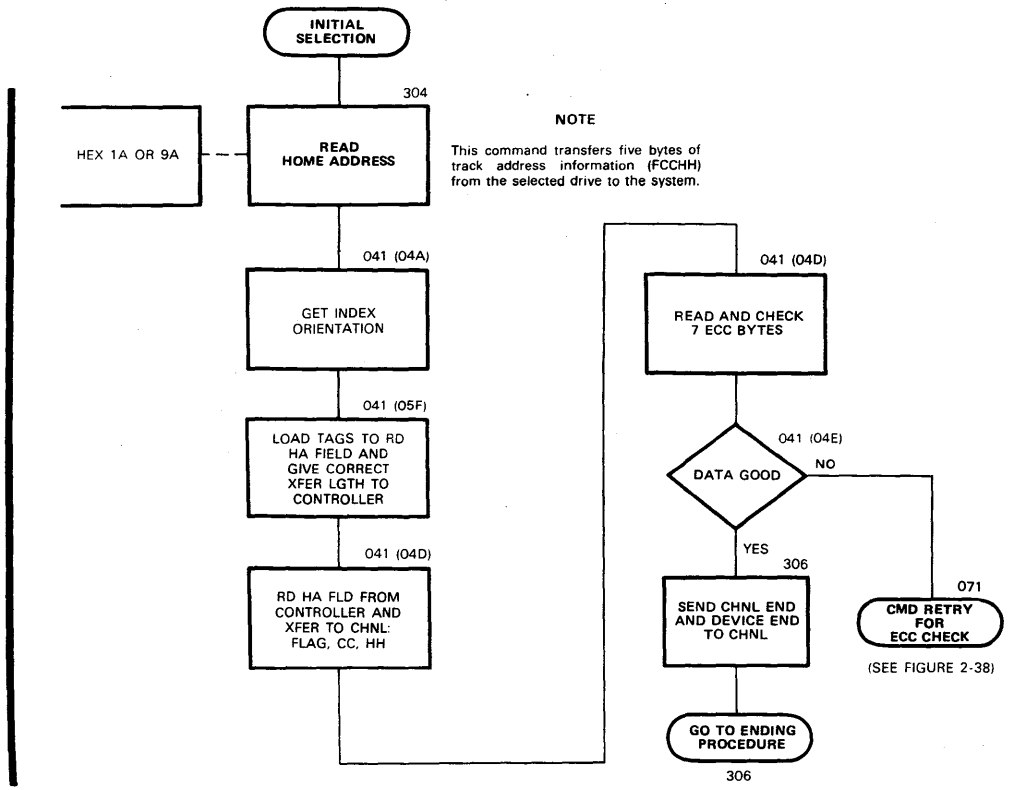


Figure 2-20. Read Record Zero, Read Count, and Read Home Address Flow Diagram (2 of 2)

READ IPL (Figure 2-21)
Command Code 02 (hex)

Performs an initial program load (IPL) by causing the SCU to seek to cylinder 0 and track 0 of a selected drive, and search for index. Upon detecting index, the R1 data area is read. This command is normally initiated by setting the storage device address in the LOAD ADDRESS switches on the FE panel and pressing the IMPL switch.

Correction codes following each data area are presented to the channel used to check for data validity. A parity check bit is added to each byte prior to its transfer to the channel.

Channel End and Device End are presented to the channel upon completion of the correction code check of the areas.

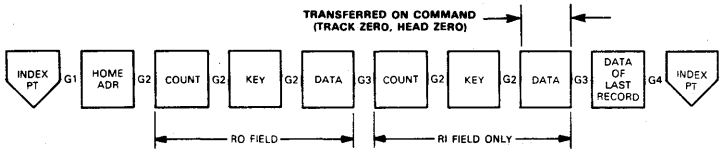
A data overrun or check condition, if detected, initiates an SCU recovery attempt by Command Retry. If Command Retry is not used or is unsuccessful when an overrun or check condition ocurs, Channel End, Device End, and Unit Check are presented to the channel. (Command Retry is not used if a correctable data error; i.e., error burst of 11 bytes or less, is detected in the area data area.)

DATA ADDRESS—Specifies main storage location where first byte of data is to be transferred.

FLAGS—Used at discretion of the programmer.

COUNT—Specifies number of bytes to be transferred.

CHAINING REQUIREMENT—Must not be preceded by a *Set File Mask* in the same chain.



G1, G2, AND G3 FIXED LENGTH GAPS
G4 VARIABLE LENGTH GAP

READ SECTOR (Figure 2-21)
Command Code 22 (hex)

Transfers one byte of data from the SCU to main storage. The byte transferred will normally contain the sector number (0-127 dec) required to access the last record processed. If a drive Power On sequence or system reset occurred after a record was processed, this byte will be zero. If the last record processed was an overflow record, this sector number is that of the last segment. Otherwise, the value will be that of the last record processed or the last set sector value loaded for that drive.

Execution of this command resets orientation information in the SCU.

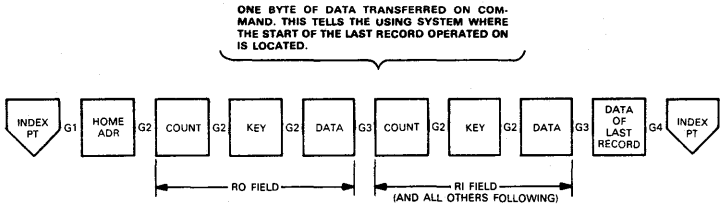
Channel End and Device End are presented to the channel after the sector number has been transferred.

DATA ADDRESS—Specifies the main storage location where the sector number is to be stored.

FLAGS—Used at discretion of the programmer.

COUNT—One.

INITIAL STATUS—Normally zero.



G1, G2, AND G3: FIXED LENGTH GAPS
G4: VARIABLE LENGTH GAP

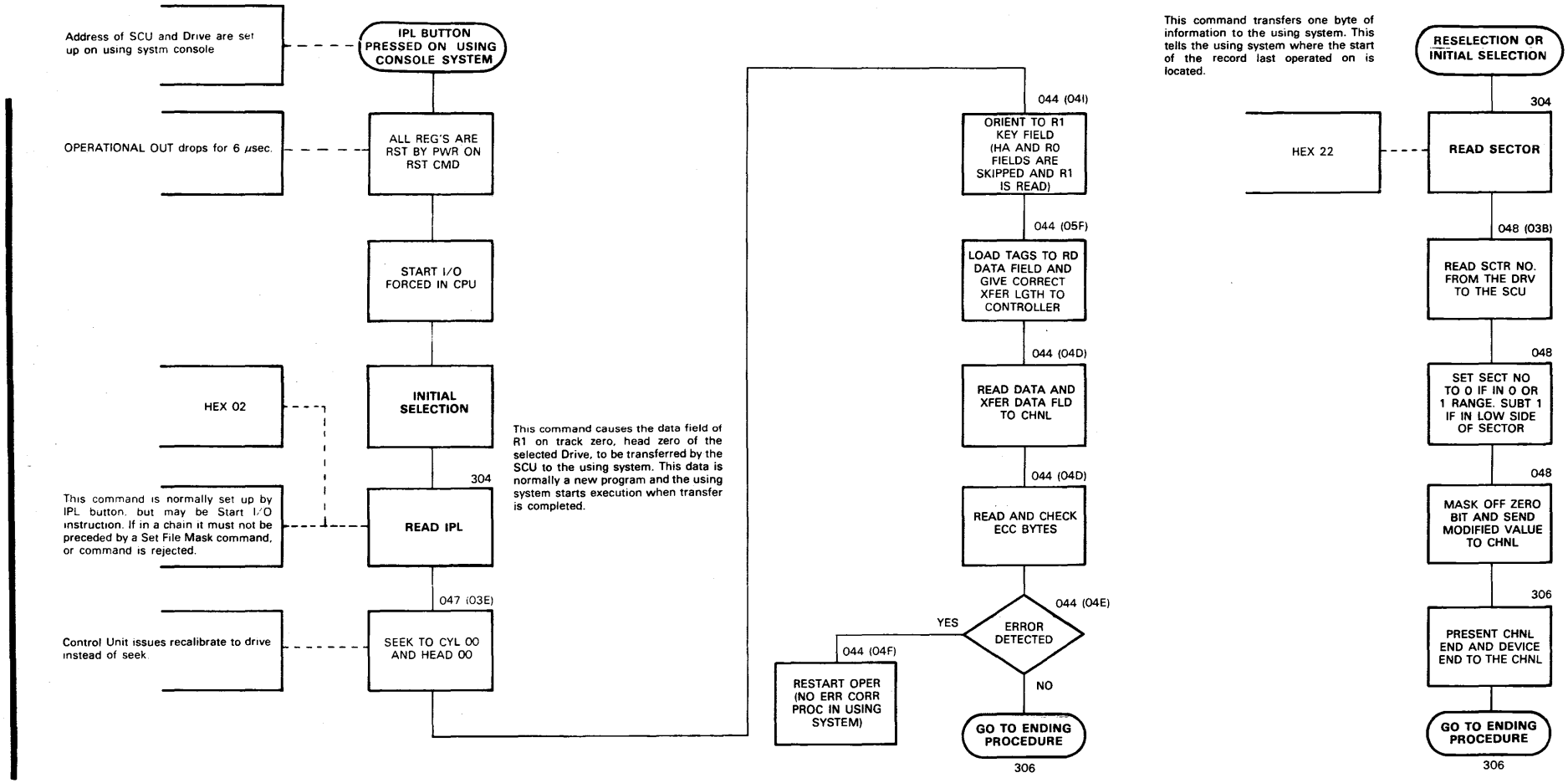


Figure 2-21 Read IPL and Read Sector Flow Diagrams

2.6.3 Read Data Transfer

The read data transfer function is described by means of the flow diagram and block diagram in Figure 2-22 and the microprogram subroutine in Figure 2-23. The flowchart is keyed to the block diagram and microprogram subroutine. Conventions for reading the microprogram subroutine are explained in Section 4.

2.6.4 Read Diagnostic to System

2.6.4.1 INTRODUCTION

Read diagnostics to the system are used to transfer diagnostic tests stored on the 650 Flexible Disc to the CPU of the using system. Any diagnostic test stored on the 650 Flexible Disc can be selected by the using system. To transfer the diagnostic test, the using system must execute a Diagnostic Load followed by a *Read Diagnostic Status 1* command. The *Diagnostic Load* command will transfer 668 bytes (containing the specific diagnostic) from the 650 Flexible Disc to control storage. The *Read Diagnostic Status 1* command then transfers the 668 bytes from control storage to the using system main storage.

Figure 2-24 illustrates a Read Diagnostic to System operation. The operation is described in the following paragraphs.

2.6.4.2 SYSTEM

- Executes a *Start I/O* instruction which addresses a diagnostic channel command.
- After receiving Device End status, executes a *Start I/O* instruction which address a *Read Diagnostic Status 1* command.

2.6.4.3 CHANNEL

- Transfers address and command information to the selected subsystem.
- Transfers status of subsystem to the system.
- Transfers diagnostic from subsystem to system storage.

2.6.4.4 CHANNEL CONTROLS

- Controls timing of data to and from the channel.
- Checks parity of data transferred to the SCU.

2.6.4.5 650 DRIVE

- Provides drive to disc.
- Provides for movement of head from track to track.
- Picks up data signals from disc via read head.

2.6.4.6 650 INTERFACE

- Starts 650 drive motor.
- Moves head to track selected by the number of Diagnostic Load commands.
- Reads 668 bytes of data from the disc.

2.6.4.7 CONTROL STORAGE

- Stores microprogram to control the operation of subsystem.
- Stores diagnostic data read from 650 Flexible Disc on diagnostic load operation for transfer on the *Read Diagnostic Status 1* command.
- Reads diagnostic data out to SCU via a Diagnostic Sense operation.

2.6.4.8 CONTROL UNIT

- Decodes the *Diagnostic Load* command.
- Sends Channel End status to the channel.
- Sends start, seek, and read control lines to the 650 interface according to the control byte sent with the *Diagnostic Load* command.
- Transfers diagnostic data from the 650 Flexible Disc to the control storage.
- Transfers Device End to the channel after data is read into control storage.
- Decodes *Read Diagnostic Status 1* command.
- Controls transfer of data from control storage to the channel.
- Transfers subsystem status to the channel.
- Controls transfer of data (668 bytes) from control storage to the channel.
- Sends disc drive and SCU status to the channel.

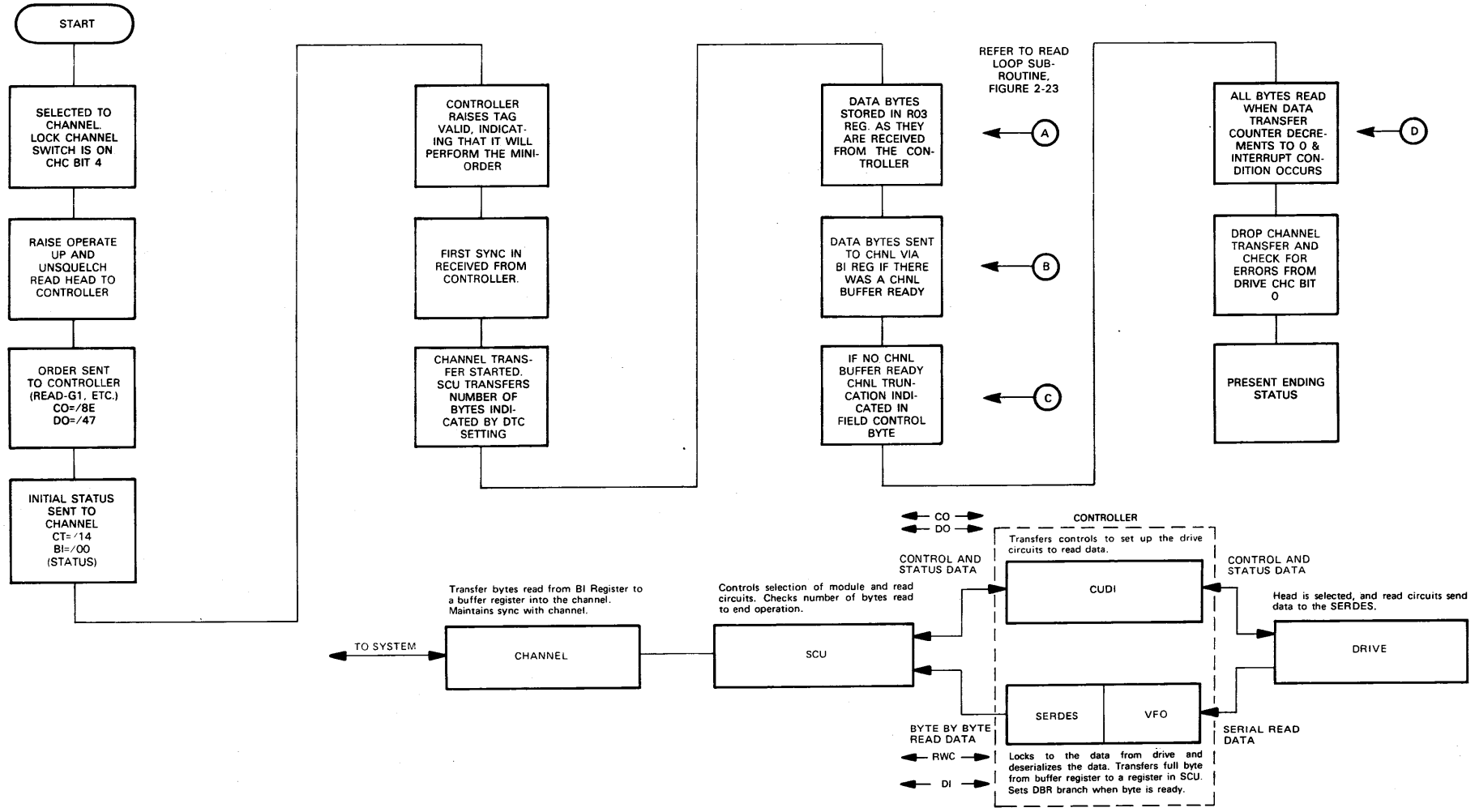


Figure 2-22. Read Data Transfer Flow and Block Diagram

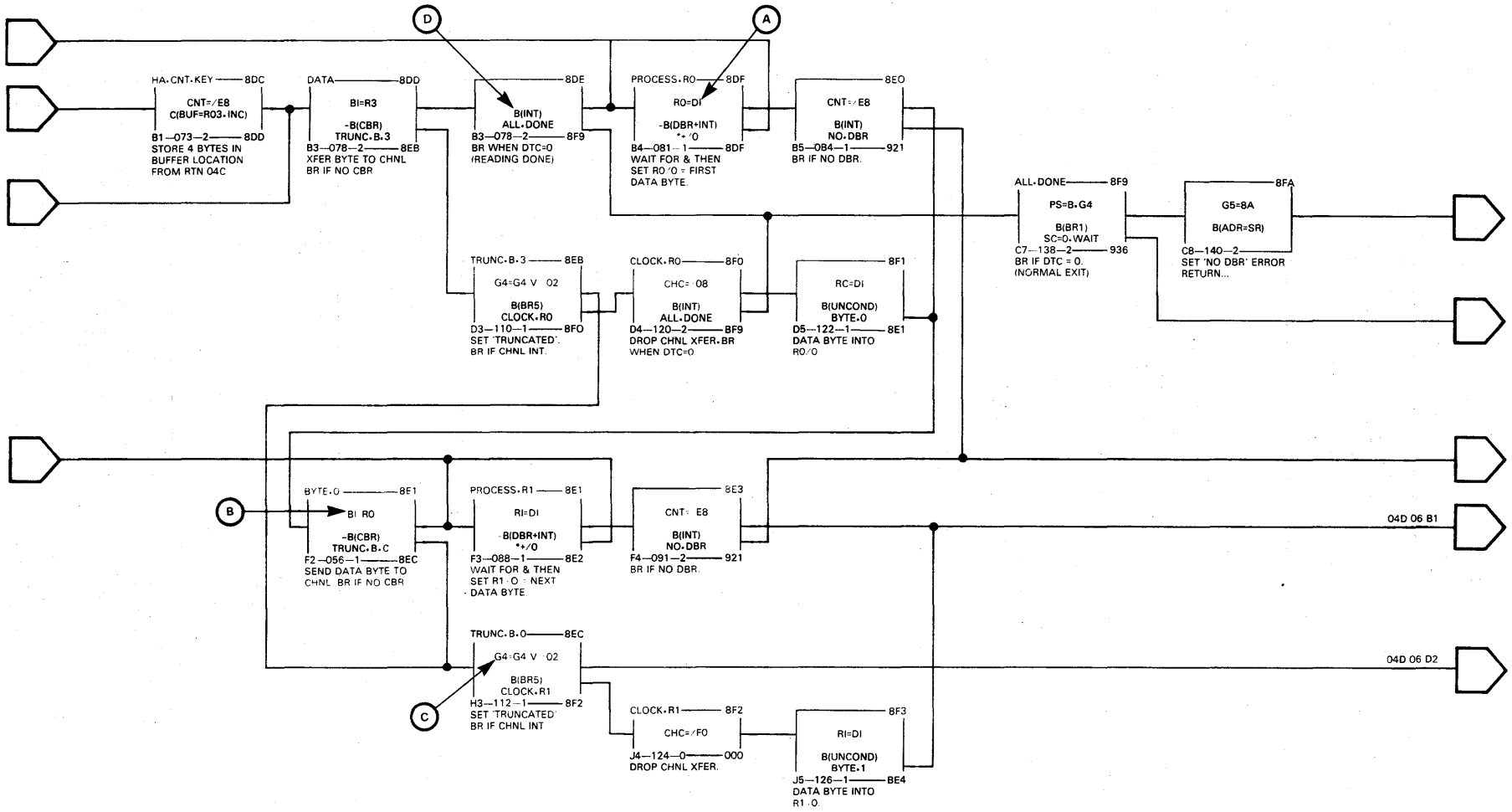


Figure 2-23. Read Loop Microprogram Subroutine

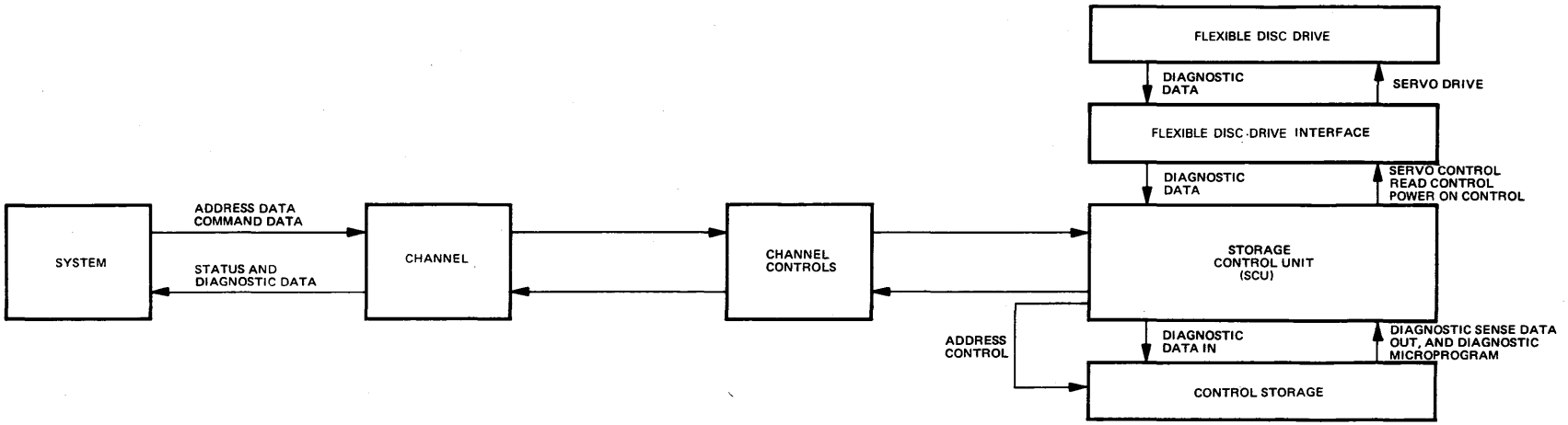


Figure 2-24 Read Diagnostic to System Operation

2.7 WRITE COMMANDS

2.7.1 Operation Block Diagram

Write commands store data from the using system on the disc pack of a selected drive. Format write commands *Write Home Address, Write R0; Write Count, Key, and Data;* and *Erase* cause the balance of the track to be changed. The Write *Key Data* and *Write Data* commands change only the key and/or data fields of a record to be changed (field length stays the same).

The following text is keyed to the Write Operation block diagram shown in Figure 2-25.

2.7.1.1 SYSTEM

The system executes the *Start I/O* instruction which addresses a channel command word (CCW) containing a write command.

2.7.1.2 CHANNEL

The channel executes the CCW to transfer the write command to the SCU.

2.7.1.3 CHANNEL INTERFACE

- Controls timing of transfer of data between the channel and the SCU.
- Checks parity of data transferred to the SCU.
- Transfers the address where data is to be stored to the SCU.
- Transfers data from the using system to the SCU.

2.7.1.4 MICROPROCESSOR

- Decodes the write command.
- Selects the addressed drive and head.
- Checks the status of the drive and transfers the status to the channel.
- Sets up the write controls in the controller.
- Controls the byte-by-byte transfer of write data from the channel interface to the controller.
- Checks subsystem for errors.
- Monitors incoming tags for error indicators.
- Transfers ending status to the channel.
- Indicates retry status to the channel if the error is a type that can be retried, and sets up to relocate the record.

2.7.1.5 CONTROLLER

- Transfers write control information to the selected drive.
- Transfers the status of the selected drive to the Microprocessor.
- Controls gap configurations.
- Receives data serial-by-byte from the Microprocessor.
- Changes data byte to serial-by-bit data.
- Sends serial data to ECC.
- Sends serial data to the MFM for precompensation.
- Uses VFO trigger pulses to control write timing.

2.7.1.6 WRITE CIRCUITS

- Sets up the drive to write on the disc pack.
- Changes data pulses from SERDES to currents for the write head.
- Provides the SCU with status of the drive.
- Index pulse generated from servo data and sent to SCU.

2.7.1.7 READ/WRITE HEAD

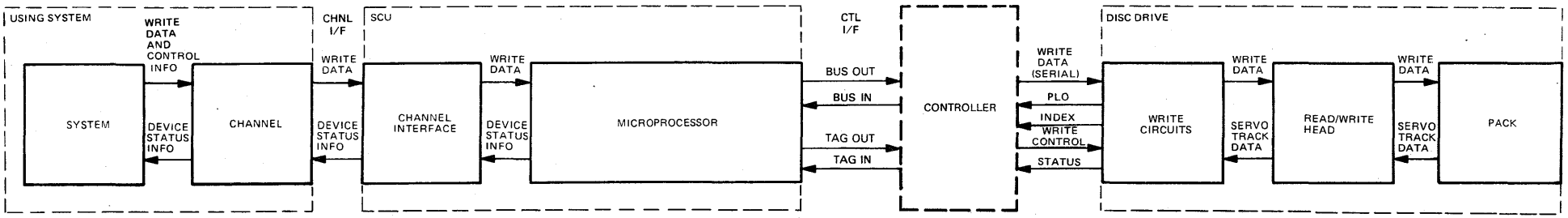
- Changes currents in write windings to magnetic flux.
- Selected head writes data.
- PLO head provides PLO pulses.
- Servo head and circuits hold access at selected track.

2.7.1.8 PACK

The pack receives and stores bits of information on the magnetic coating of the disc.

2.7.2 Command Descriptions

Each write command is described in the following pages by means of a flow diagram and a narrative description.



WRITE DATA (Figure 2-26)
Command Code 05 (hex)

Performs normal record updating after track formatting. Execution of command causes specified data in main storage to be written in data area of selected record. Number of bytes written is specified in the CCW count field. It may be less than the data length (DL) specified in the formatted record.

Channel End and Device End are presented after the ECC bytes have been written for the data area.

DATA ADDRESS—Specifies main storage location of data used to update record.

FLAGS—Used at discretion of programmer.

COUNT—Specifies number of data bytes to be written. If CCW Count field is less than the formatted record DL, the SCU writes zeros in the remaining data area, followed by writing ECC bytes, and presents Channel End and Device End to the channel. If CCW count field is greater than the formatted record DL, the SCU writes only the number of bytes indicated in the DL and then writes ECC bytes.

CHAINING REQUIREMENTS—Must be chained from *Search ID* or *Search Key Equal.* (The *Search* command must compare equal on all bytes of the searched field.) If chaining requirement is not met Unit Check is presented in initial status.

WRITE KEY AND DATA (Figure 2-26)
Command Code 0D (hex)

Performs record updating after track formatting. Execution of command causes specified data in main storage to be written in key and data areas of selected record. Number of bytes written is specified in the CCW count field. It may be less than the key length (KL) and data length (DL) specified in the formatted record.

Channel End and Device End are presented after the ECC bytes have been written for the data area.

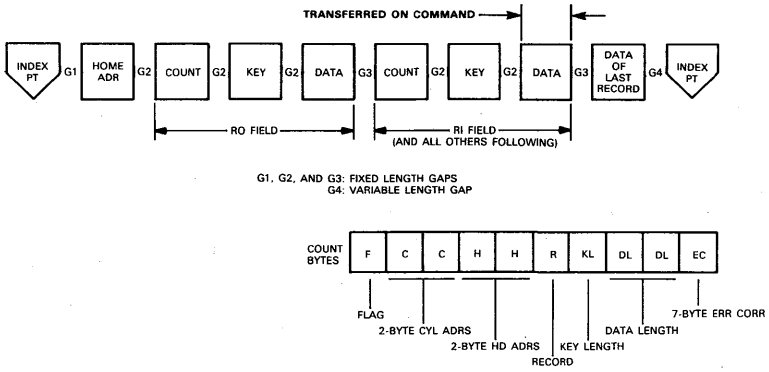
DATA ADDRESS—Specifies main storage location of data used to update record.

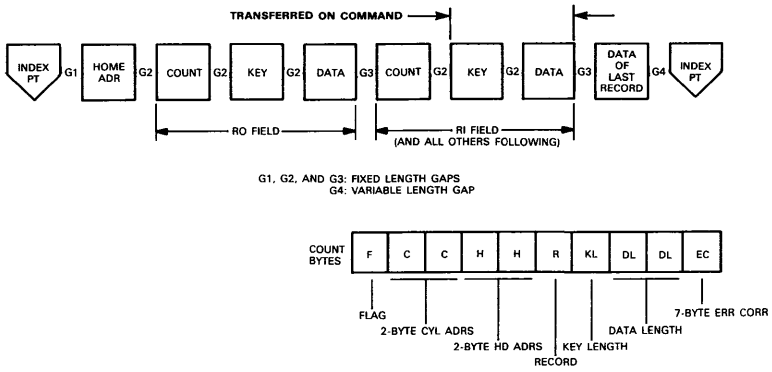
FLAGS—Used at discretion of programmer.

COUNT—Specifies number of key and data bytes to be written. If CCW count is less than the formatted record KL/DL, the SCU writes zeros in the remaining areas, followed by writing ECC bytes, and presents Channel End and Device End to the channel. If CCW count is greater than the formatted record KL/DL, the SCU writes only the number of bytes indicated in the KL/DL and then writes ECC bytes.

INITIAL STATUS—Normally zero.

CHAINING REQUIREMENT—Must be chained from a *Search ID* Equal command. (The *Search ID* command must compare equal on all bytes of the searched field.) If chaining requirement is not met, Unit Check is presented in initial status.





WRITE COUNT, KEY, AND DATA (Figure 2-26)
Command Code 1D (hex)

Causes the count, key, and data area of a record in main storage to be written on a selected drive. The count area is made up of the first eight bytes from main storage. The flag byte is generated by the SCU; the remaining data is written in the key and data areas as specified by the KL and DL bytes in the count area.

Channel End and Device End are presented to the channel after correction code bytes are written for the data area.

DATA ADDRESS—Specifies main storage location where count, key, and data bytes of record are located.

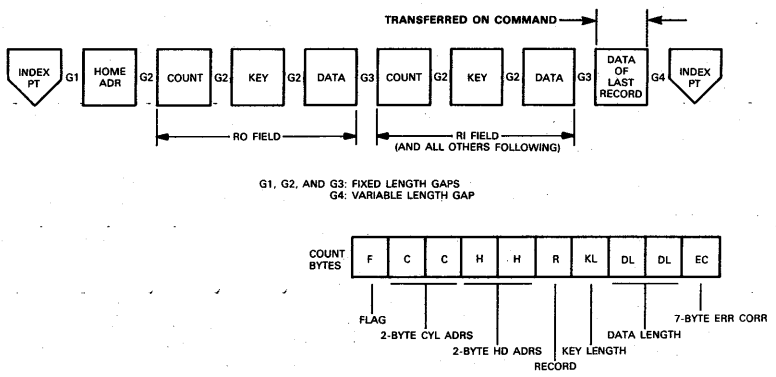
FLAGS—Used at discretion of the programmer.

COUNT—Specifies number of count, key, and data bytes (8 + KL + DL) to be written. If CCW count area is less than 8 + KL + DL, the SCU writes zeros in the remainder of the record.

INITIAL STATUS—Normally zero.

CHAINING REQUIREMENTS—Must be chained from *Write Record Zero; Write Count, Key, and Data; Search ID Equal,* or *Search Key Equal* CCW. If chaining requirement is not met, Unit Check is presented in initial status.

A *Read Data* or *Read Key and Data* command may be inserted between *Search* command and *Write Count, Key, and Data* command.



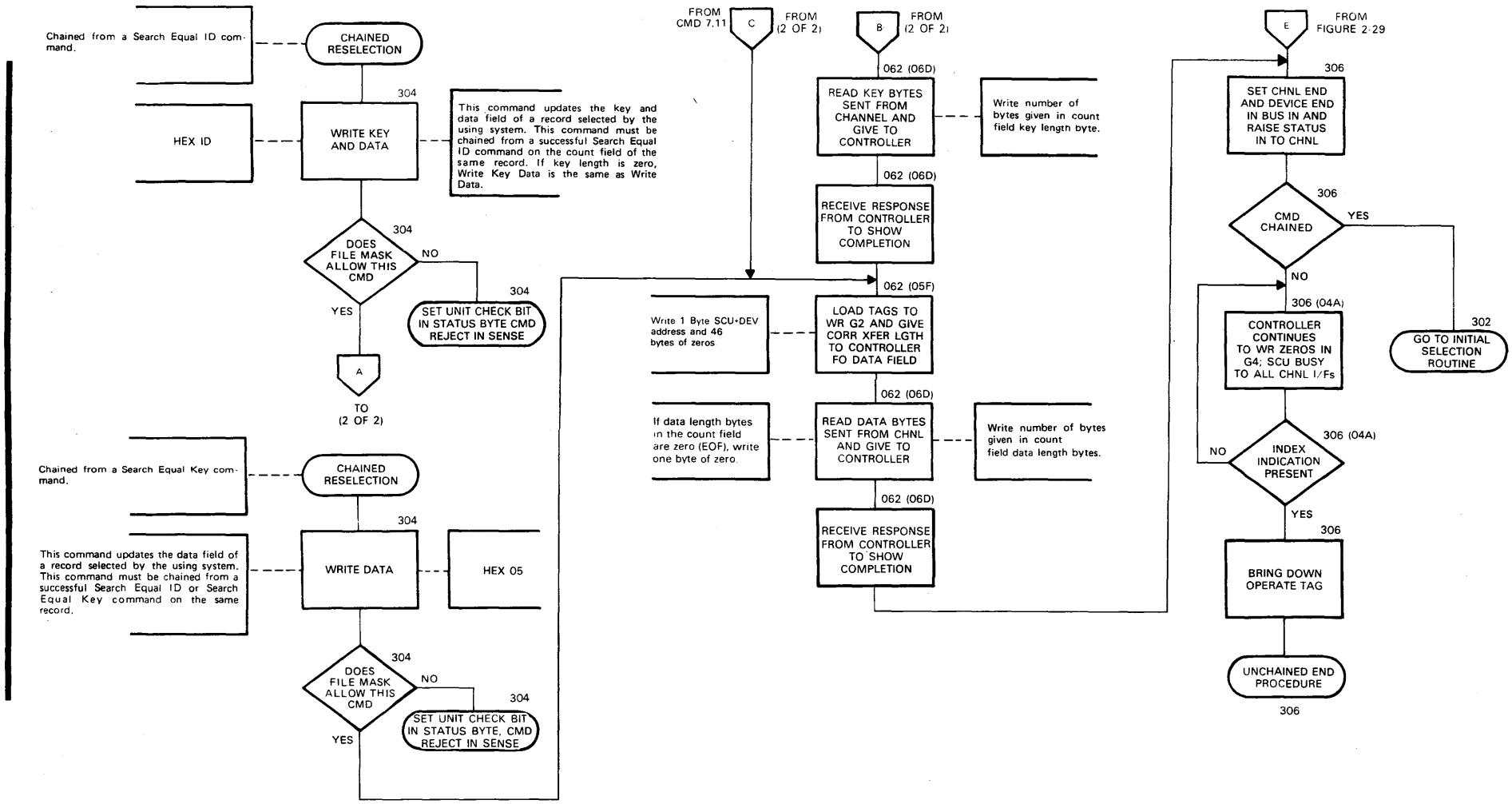


Figure 2-26. Write Count, Key, and Data; Write Key and Data; and Write Data Flow Diagrams (1 of 2)

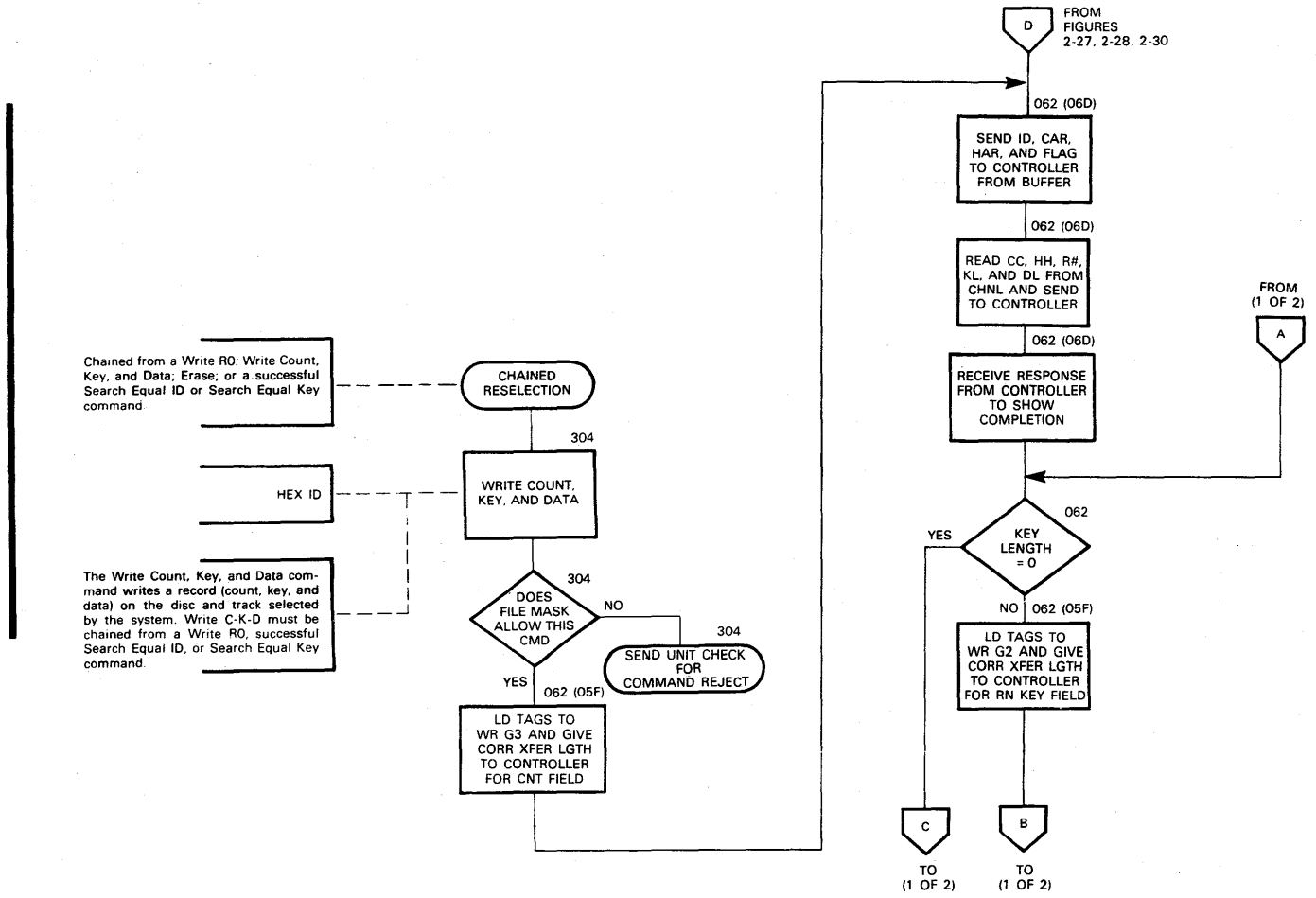


Figure 2-26. Write Count, Key, and Data; Write Key and Data; and Write Data Flow Diagrams (2 of 2)

WRITE SPECIAL COUNT, KEY, AND DATA (Figure 2-27)

Command Code 01 (hex)

Formats a segment of an overflow record, the last segment of which is written by a normal *Write Count, Key, and Data* command. The count area of the record is made up of the first eight bytes from main storage. The flag byte contains a one written in bit position 4. This bit, written by the SCU, indicates that another part of the next record is located on the next track. Correction code bytes are written by the SCU at the end of each record area.

Channel End and Device End are presented to the channel after correction bytes have been written for the data area.

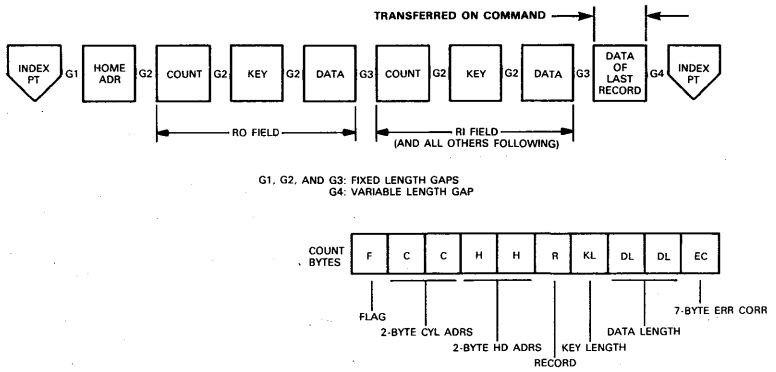
DATA ADDRESS—Specifies the main storage location where the count, key, and data areas of the record are located.

FLAGS—Used at discretion of the programmer.

COUNT—Specifies number of bytes in the count, key, and data areas bytes (8 + KL + DL) to be transferred. If the CCW count is less than 8 + KL + DL, the SCU writes zeros in the remainder of the record.

INITIAL STATUS—Normally zero.

CHAINING REQUIREMENTS—Must be chained from a *Write Count, Key, and Data; Search ID Equal;* or *Search Key Equal* command. (The search commands must compare equal on all bytes of the searched field.) A *Read Data* or *Read Key and Data* command may be inserted between the Search command and *Write Special Count, Key, and Data* command. If chaining requirements are not met, Unit Check is presented in initial status. An overflow record must be the last record on the track. A subsequent overflow record must be the only record on the track (excluding R0).



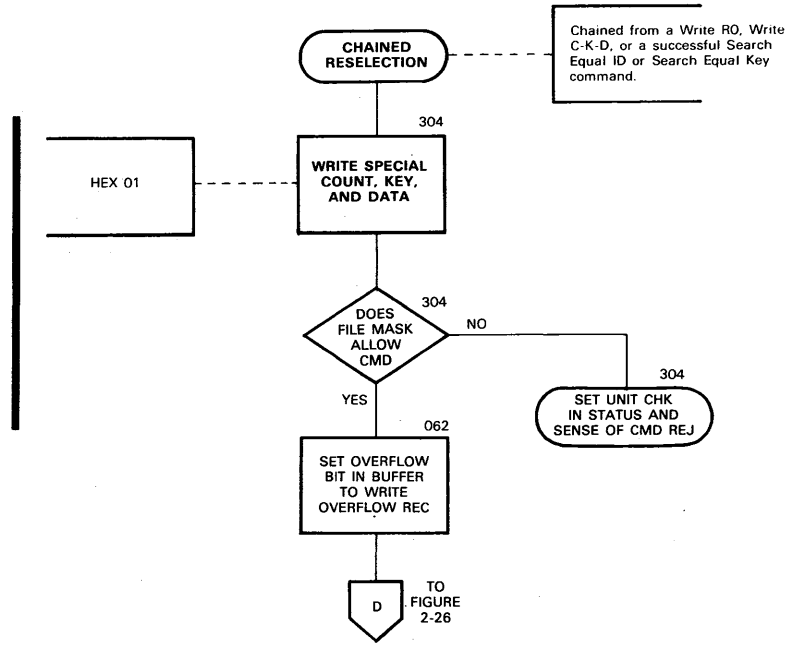


Figure 2-27. Write Special Count, Key, and Data Flow Diagram

WRITE RECORD ZERO (Figure 2-28)
Command Code 15 (hex)

Causes specified data in record zero (R0) count, key, and data bytes in main storage to be written on selected drive. The count area is made up of the first eight bytes from main storage. The flag is generated by the SCU; the remaining data is written in the key and data areas specified by the KL and DL bytes in the count area.

Channel End and Device End are presented to the channel after correction code bytes are written for the data area.

Proper operation with Operation System (OS) requires an eight-byte data field in R0.

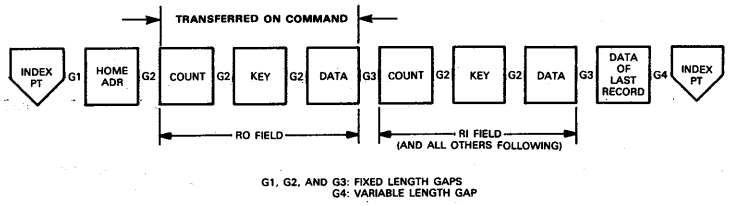
DATA ADDRESS—Specifies main storage location of R0 count, key, and data bytes.

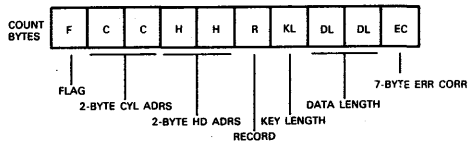
FLAGS—Used at discretion of the programmer.

COUNT—Specifies number of count, key, and data bytes (8 + KL + DL) to be written. If CCW count area is less than 8 + KL + DL, the SCU writes zeros in the remainder of the record.

INITIAL STATUS—Normally zero.

CHAINING REQUIREMENT—Must be chained from a successful *Write Home Address,* or *Search Home Address Equal* command.





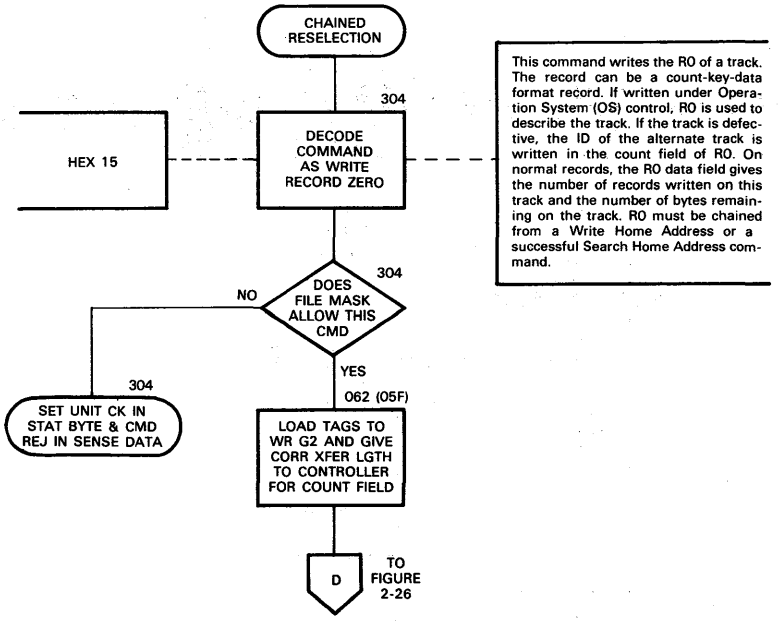


Figure 2-28. Write Record Zero Flow Diagram

WRITE HOME ADDRESS (Figure 2-29)
Command Code 19 (hex)

Establishes track identity, a prerequisite for data operations on that track. The SCU orients on index; writes gap 1, home address, ECC bytes, source ID; and then writes gap 2 or gap 4. Bit 5 of the home address flag byte (CE pack) must be zero.

Channel End and Device End are presented after the ECC bytes have been written for the data area.

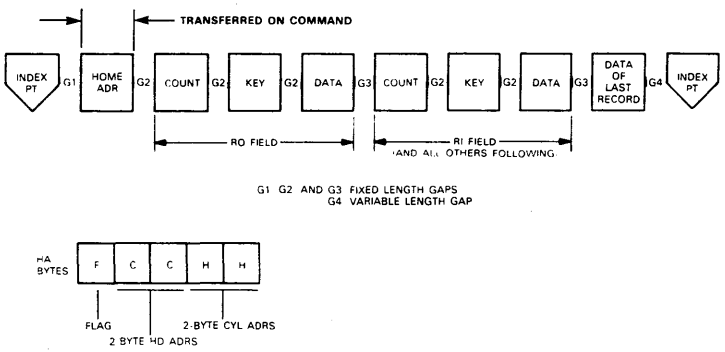
DATA ADDRESS—Specifies main storage location of the home address bytes (F, CC, and HH).

FLAGS—Used at discretion of the programmer

COUNT—Should be five. If less than five, the SCU records zeros until five bytes have been written. If count is greater than five, only the first five bytes at the main storage location are written.

INITIAL STATUS—Normally zero.

CHAINING REQUIREMENTS—Must be preceded by a *Set File Mask* permitting *Write Home Address* commands. If requirement is not met, Unit Check is presented in initial status.



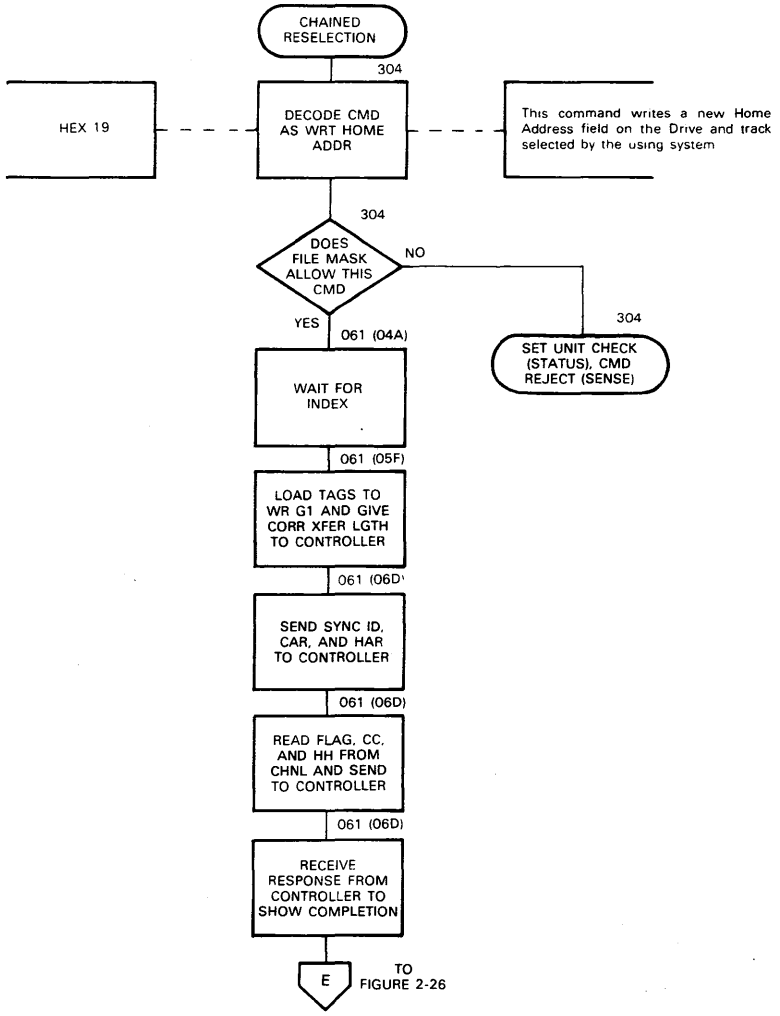


Figure 2-29. Write Home Address Flow Diagram

ERASE (Figure 2-29)
Command Code 11 (hex)

Writes zeros in count, key, and data areas on a selected drive. The address mark is not written with this command. The erased record and all records that follow on the track are unrecoverable.

Channel End and Device End are presented at the end of the data area. Remainder of the track is padded with zeros.

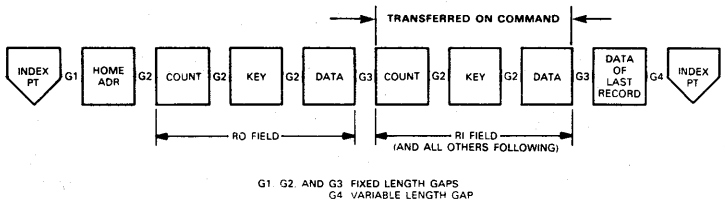
DATA ADDRESS—Specifies main storage location where count, key, and data areas of the record are located.

FLAGS—Used at discretion of the programmer.

COUNT—Specifies number of bytes in count, key, and data areas of the record.

CHAINING REQUIREMENTS--

- Must be chained from a *Write Record Zero; Write Count, Key and Data* command; *Search ID Equal;* or *Search Key Equal* command. (The Search commands must compare equal on all bytes of the searched field.)
- Must not be chained from an *Erase* command.



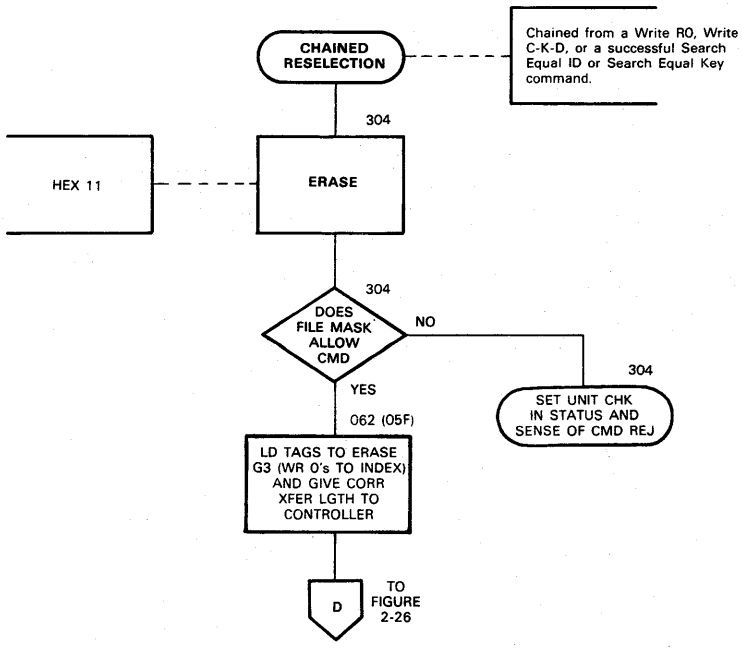


Figure 2-30. Erase Flow Diagram

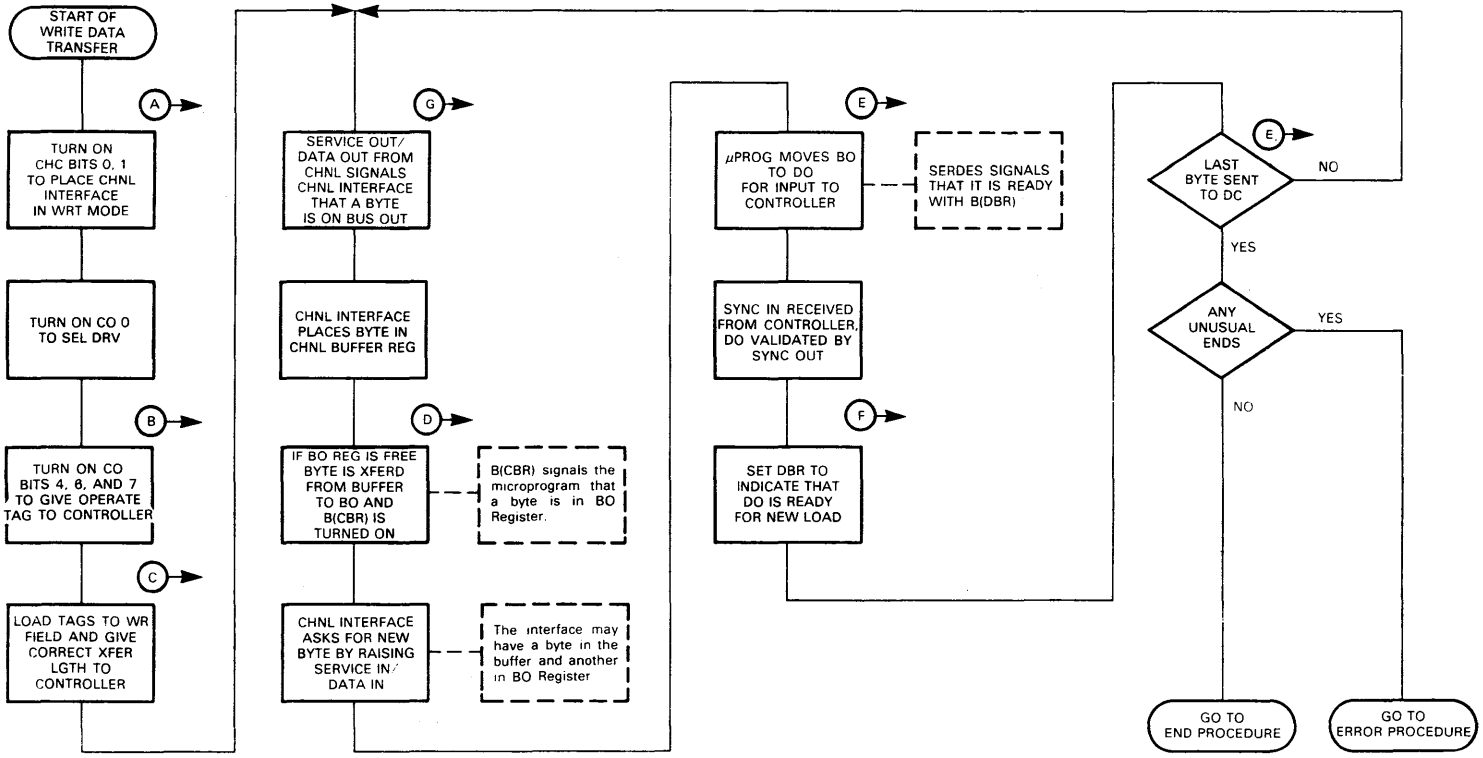


Figure 2-31. Write Data Transfer (1 of 2)

2.7.3 Write Data Transfer

The write data function is described by means of the flow diagram and block diagram in Figure 2-31, and the microprogram subroutine in Figure 2-32. The flowchart is keyed to block diagram and microprogram subroutine. Conventions for reading the microprogram subroutine are explained in Section 4.

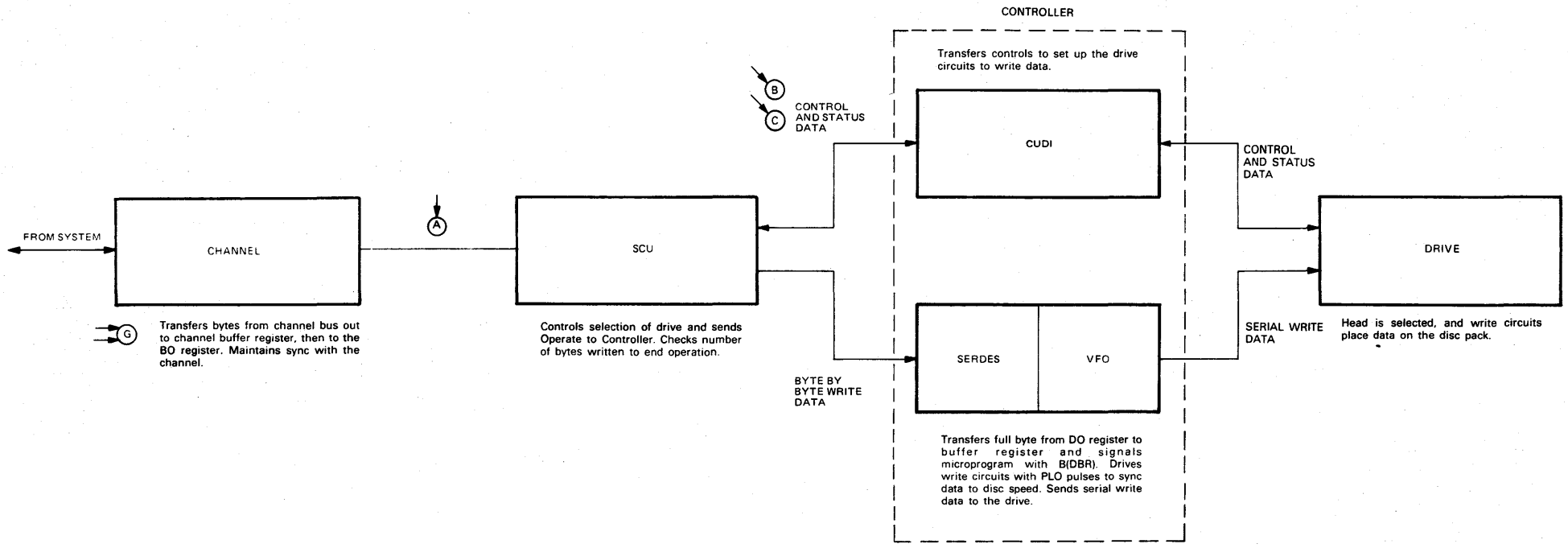


Figure 2-31. Write Data Transfer (2 of 2)

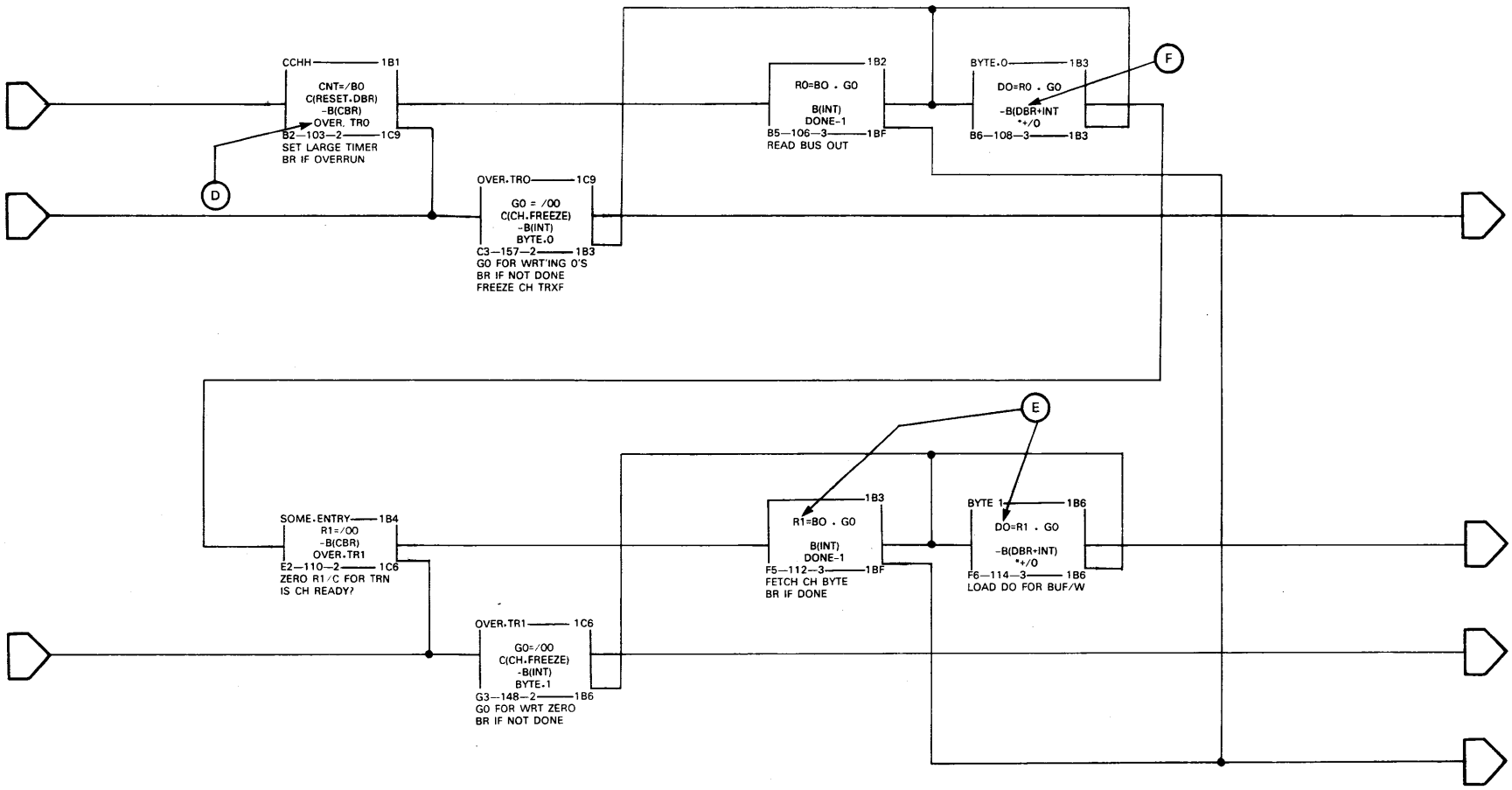


Figure 2-32. Write Loop Microprogram Subroutine

2.8 SEARCH COMMANDS

Search commands encompass *Search Home Address, Search ID Equal, Search ID High, Search ID Equal or High, Search Key Equal, Search Key High,* and *Search Key Equal or High.* Descriptions and flow diagrams for each of these commands are presented in the following pages.

Operation with the *Search ID Equal, Search ID High,* and *Search ID Equal or High* commands is as follows:

- These commands compare search data from the using system with data read from the drive.
- Only one record can be searched for at a time.
- When a search is successful, Status Modifier (bit 1) is turned on in the status byte with Channel End and Device End.
- If Multi-track (bit 0 of the command byte) is on, automatic head switching will take place when index is passed
- Multi-track (MT) operation will not cross cylinder boundary
- When a search is unsuccessful, the Search command must be reissued by the channel to continue the search.
- Transfer In Channel (TIC) command must follow a Search command to allow continued search.
- Passing the index point twice on the same track or the detection of end of cylinder discontinues Search operation. Channel End, Device End, and Unit Check are presented in the ending status byte. No Record Found is set in sense byte 1 for single track searches or End of Cylinder for multi-track operations.

SEARCH HOME ADDRESS EQUAL (Figure 2-33)

Command Code 39 (hex) single track
B9 (hex) multi-track

Causes SCU to compare home address (HA) on designated track with HA read from main storage. Execution begins with a search for Index. If the single track command code is designated, the search takes place on the present track; if the multi-track code is indicated, the search starts on the next track. When Index is detected, the cylinder and head numbers from main storage are compared with those in the track HA. If the comparison is equal, Channel End, Device End, and Status Modifier are presented to the channel. If the comparison is unequal, Channel End and Device End only are presented to the channel.

The Flag byte is not transferred to the channel or compared during command execution. If a Bus-Out parity error is detected, Channel End, Device End, and Unit Check Status are presented in ending status.

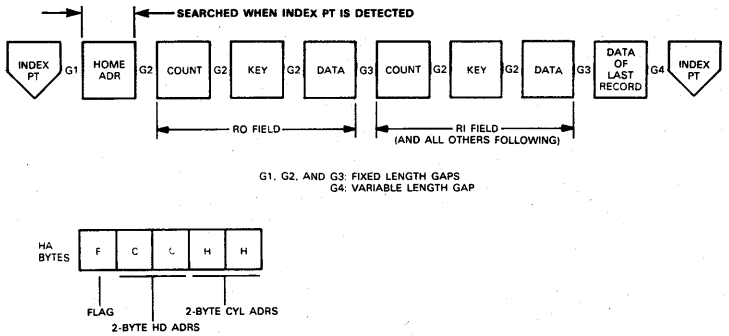
DATA ADDRESS—Specifies main storage location of a cylinder number (CC) and head number (HH).

FLAGS—Used at discretion of the programmer.

COUNT—Should be four. If count is greater than four bytes, the search is completed when four bytes are received by the SCU and the command is terminated with Channel End and Device End (and Status Modifier if the comparison was equal.)

If the count is less than four bytes, a comparison of the storage and track data continues until the CCW count is decremented to zero. Channel End and Device End are presented to the channel when the home address and correction code bytes are read and checked. Status Modifier is presented with Channel End and Device End if the search was satisfied on short field. Chained write is not allowed by the Controller.

INITIAL STATUS—Normally zero.



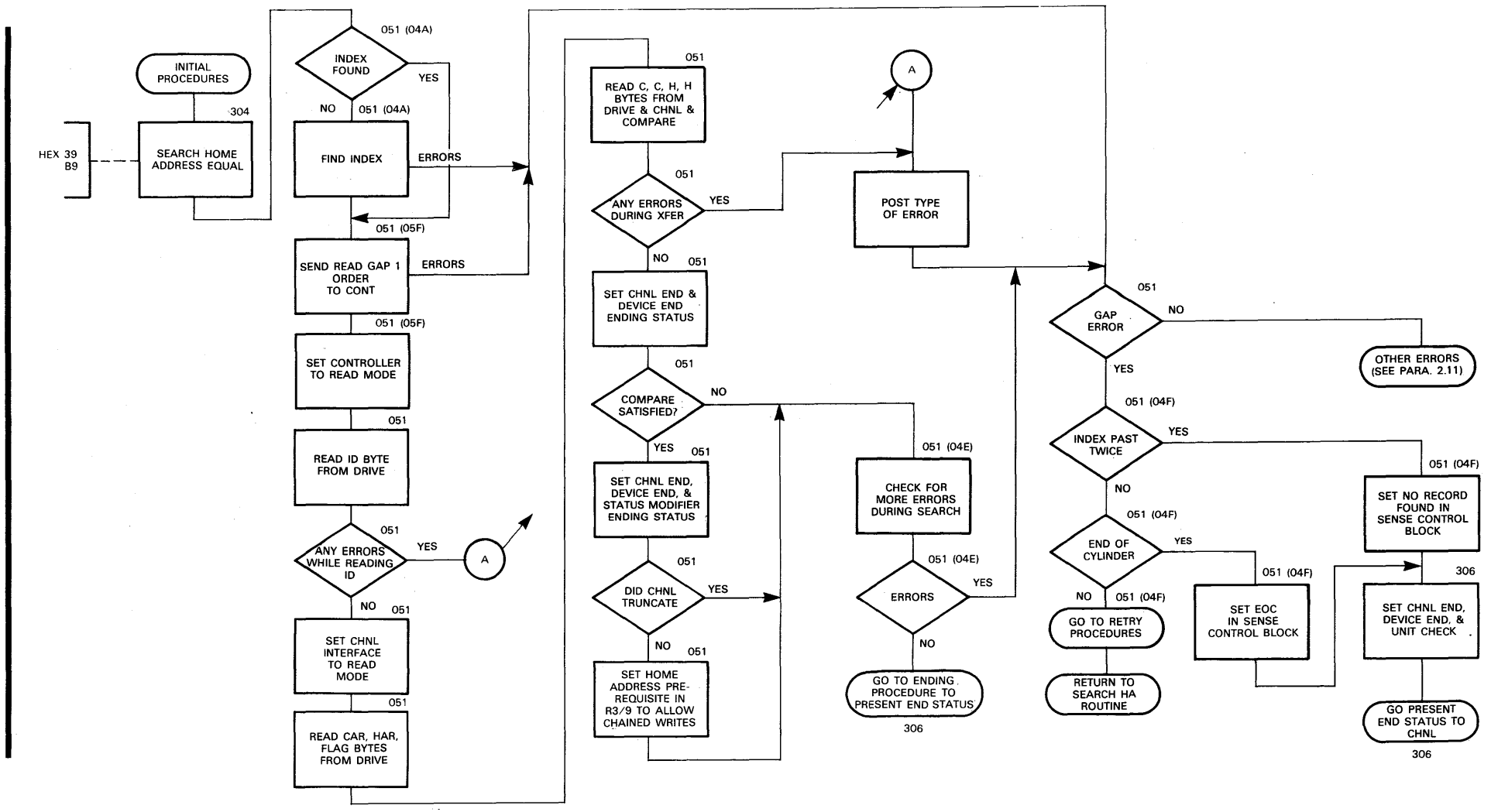


Figure 2-33. Search Home Address Equal Flow Diagram

SEARCH ID EQUAL (Figure 2-34)

Command Code: 31 (hex) single track
B1 (hex) multi-track

Compares an ID in main storage with the next count ID area on the track (including R0). If an equal comparison results, Channel End, Device End, and status modifier are presented to the channel. If an unequal comparison results, Channel End and Device End only are presented to the channel.

If a single-track search is specified, the search is confined to one track. The search is repeated until the search condition is satisfied or until two index points are detected. Upon detection of the second index, Channel End, Device End, and Unit Check are presented to the channel; and No Record Found is set in sense byte 1.

If a multi-track search is indicated, the search continues from track to track as long as the command is presented to the channel. The head number is automatically incremented at index until the search condition is satisfied, or until end of cylinder is reached. Upon detection of end of cylinder, Channel End, Device End, and Unit Check are presented to the channel; and End of Cylinder is set in sense byte 1.

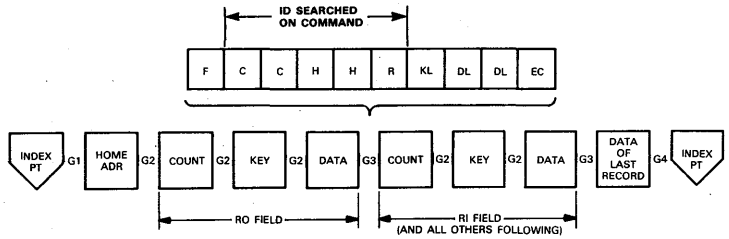
DATA ADDRESS—Specifies main storage address of a five-byte portion of a count area containing CC, HH, and R.

FLAGS—Used at discretion of the programmer.

COUNT—Should be five. If count is greater than five, only the first five bytes from main storage are compared. Channel End and Device End are presented to terminate the command, and status modifier is presented if the comparison was equal.

If count is less than five, a comparison of main storage and track data continues until the CCW count is zero. Channel End and Device End are presented to the channel when the ID and correction code bytes are read and checked. Status modifier is presented if the search on the short field is satisfied.

INITIAL STATUS—Normally zero.



G1, G2, AND G3: FIXED LENGTH GAPS
G4: VARIABLE LENGTH GAP

SEARCH ID HIGH (Figure 2-34)

Command Code: 51 (hex) single track
D1 (hex) multi-track

Compares an ID in main storage with the next count ID area on the track (including R0). If a high comparison results, the track ID is greater than the main storage ID and Channel End, Device End, and status modifier are presented to the channel. If a high comparison does not result, the track ID is not greater than the main storage ID and Channel End and Device End only are presented to the channel.

If a single-track search is specified, the search is confined to one track. The search is repeated until the search condition is satisfied or until two index points are detected. Upon detection of the second index, Channel End, Device End, and Unit Check are presented to the channel; and No Record Found is set in sense byte 1.

If a multi-track search is indicated, the search continues from track to track as long as the command is presented to the channel. The head number is automatically incremented at index until the search condition is satisfied, or until end of cylinder is reached. Upon detection of end of cylinder, Channel End, Device End, and Unit Check are presented to the channel; and End of Cylinder is set in sense byte 1.

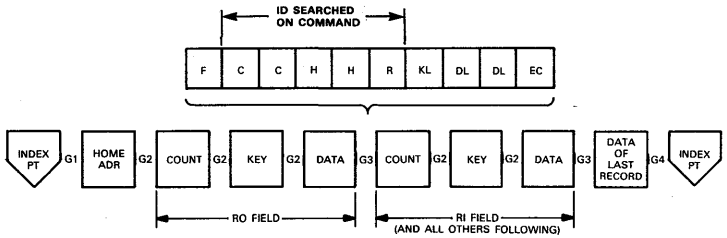
DATA ADDRESS—Specifies main storage address of a five-byte portion of a count area containing CC, HH, and R.

FLAGS—Used at discretion of the programmer.

COUNT—Should be five. If count is greater than five, only the first five bytes from main storage are compared. Channel End and Device End are presented to terminate the command, and status modifier is presented if the track ID was high.

If count is less than five, a comparison of main storage and track data continues until the CCW count is zero. Channel End and Device End are presented to the channel when the ID and correction code bytes are read and checked. Status modifier is presented if the search on the short field is satisfied.

INITIAL STATUS—Normally zero.



G1, G2, AND G3: FIXED LENGTH GAPS
G4: VARIABLE LENGTH GAP

SEARCH ID EQUAL OR HIGH (Figure 2-34)

Command Code: 71 (hex) single track
F1 (hex) multi-track

Compares an ID in main storage with the next count ID area on the track (including R0). If an equal or high comparison results, the track ID is equal to or greater than the main storage and Channel End, Device End, and status modifier are presented to the channel. If an equal or high comparison does not result, the track ID is less than the main storage ID and Channel End and Device End only are presented to the channel.

If a single-track search is specified, the search is confined to one track. The search is repeated until the search condition is satisfied or until two index points are detected. Upon detection of the second index, Channel End, Device End, and Unit Check are presented to the channel; and No Record Found is set on sense byte 1.

If a multi-track search is indicated, the search continues from track to track as long as the command is presented to the channel. The head number is automatically incremented at index until the search condition is satisfied, or until end of cylinder is reached. Upon detection of end of cylinder, Channel End, Device End, and Unit Check are presented to the channel; and End of Cylinder is set in sense byte 1

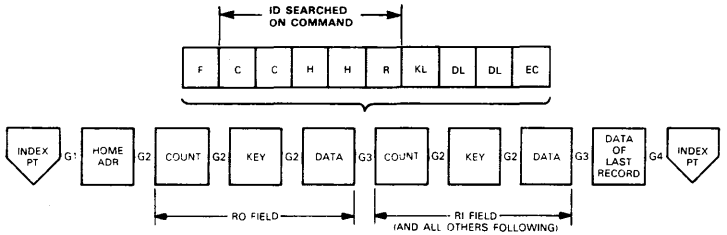
DATA ADDRESS—Specifies main storage address of a five-byte portion of a count area containing CC, HH, and R.

FLAGS—Used at discretion of the programmer.

COUNT—Should be five. If count is greater than five, only the first five bytes from main storage are compared. Channel End and Device End are presented to terminate the command, and status modifier is presented if the comparison was equal or high.

If count is less than five, a comparison of main storage and track data continues until the CCW count is zero. Channel End and Device End are presented to the channel when the ID and correction code bytes are read and checked. Status modifier is presented if the search on the short field is satisfied.

INITIAL STATUS—Normally zero.



G1 G2 AND G3 FIXED LENGTH GAPS
G4 VARIABLE LENGTH GAP

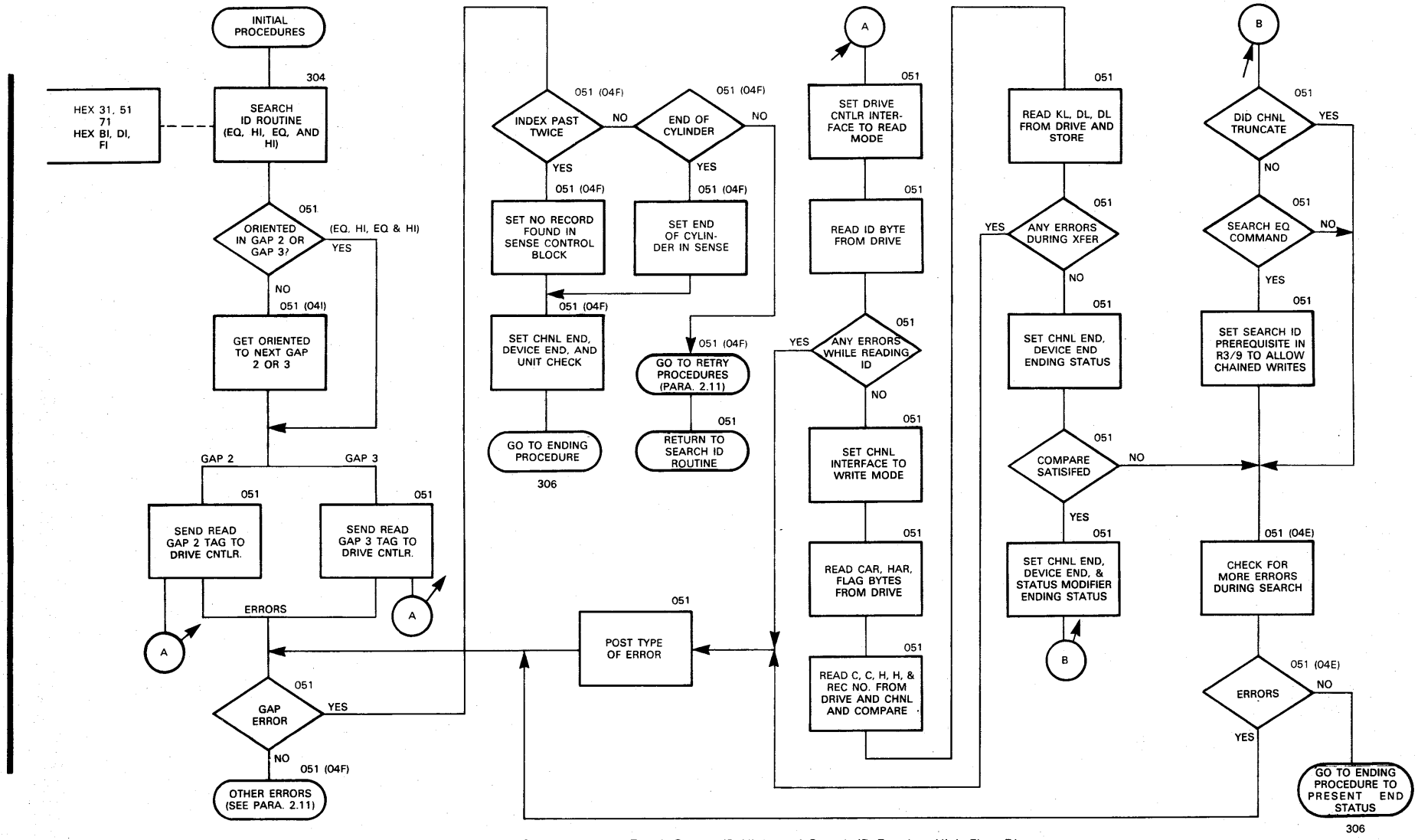


Figure 2-34. Search ID Equal, Search ID High, and Search ID Equal or High Flow Diagram

SEARCH KEY EQUAL (Figure 2-35)

Command Code: 29 (hex) single track
A9 (hex) multi-track

Compares a key area in main storage with the next key area on the track (excluding R0). If an equal comparison results, Channel End, Device End, and status modifier are presented to the channel. If an unequal comparison results, Channel End and Device End only are presented to the channel.

If a single-track search is specified, the search is confined to one track. The search is repeated until the search condition is satisfied or until two index points are detected. Upon detection of the second index, Channel End, Device End, and Unit Check are presented to the channel; and No Record Found is set in sense byte 1.

If a multi-track search is indicated, the search continues from track to track as long as the command is presented to the channel. The head number is automatically incremented at index until the search condition is satisfied, or until end of cylinder is reached. Upon detection of end of cylinder, Channel End, Device End, and Unit Check are presented to the channel; and End of Cylinder is set in sense byte 1

Execution of this command with a key length (KL) of zero does not set a status modifier. If followed by a chained *Read Data* command, the data area read is that of the next record.

DATA ADDRESS—Specifies main storage location where key to be compared is located.

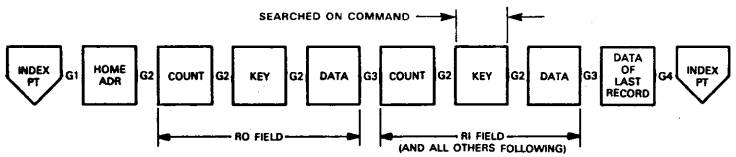
FLAGS—Used at discretion of the programmer

COUNT—Should be equal to KL of record containing key to be compared. If count is greater than KL, the search operation is completed when the key area is read. Channel End and Device End are presented to the channel, terminating the command. Status modifier is presented if the comparison was valid.

If count is less than KL, the track and main storage comparison continues until the CCW count is zero. Channel End and Device End are presented after the key area and subsequent correction code bytes are read and checked. Status modifier is presented if the search on the short field was satisfied.

INITIAL STATUS—Normally zero.

SPECIAL NOTE—When command is chained from *Search ID* or *Read Count,* the key compared is in the same record as the ID or count. *Search Key Equal* bypasses R0 key field unless chained from *Search ID* command, which search R0.



G1, G2, AND G3: FIXED LENGTH GAPS
G4: VARIABLE LENGTH GAP

SEARCH KEY HIGH (Figure 2-35)

Command Code: 49 (hex) single track

Compares a key area in main storage with the next key area on the track (excluding R0). If a high comparison results, the track key is greater than the main storage key and Channel End, Device End, and status modifier are presented to the channel. If an equal or high comparison does not result, the track key is not greater than the main storage key and Channel End and Device End only are presented to the channel.

If a single-track search is specified, the search is confined to one track. The search is repeated until the search condition is satisfied or until two index points are detected. Upon detection of the second index, Channel End, Device End, and Unit Check are presented to the channel; and No Record Found is set in sense byte 1.

If a multi-track search is indicated, the search continues from track to track as long as the command is presented to the channel. The head number is automatically incremented at index until the search condition is satisfied, or until end of cylinder is reached. Upon detection of end of cylinder, Channel End, Device End, and Unit Check are presented to the channel; and End of Cylinder is set in sense byte 1.

Execution of this command with a key length (KL) of zero does not set a status modifier. If followed by a chained *Read Data* command, the data area read is that of the next record.

DATA ADDRESS—Specifies main storage location where key to be compared is located.

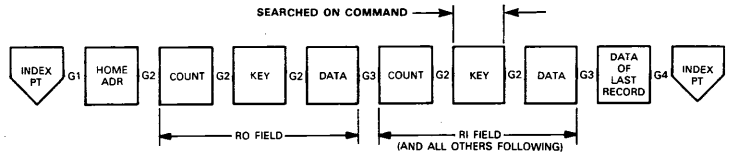
FLAGS—Used at discretion of the programmer.

COUNT—Should be equal to KL of record containing key to be compared. If count is greater than KL, the search operation is completed when the key area is read. Channel End and Device End are presented to the channel, terminating the command. Status modifier is presented if the track key was high.

If count is less than KL, the track and main storage comparison continues until the CCW count is zero. Channel End and Device End are presented after the key area and subsequent correction code bytes are read and checked. Status modifier is presented if the search on the short field was satisfied.

INITIAL STATUS—Normally zero.

SPECIAL NOTE—When command is chained from *Search ID* or *Read Count,* the key compared is in the same record as the ID or count. *Search Key Equal* bypasses R0 key field unless chained from *Search ID* command, which searched R0.



G1, G2, AND G3: FIXED LENGTH GAPS
G4: VARIABLE LENGTH GAP

SEARCH KEY EQUAL OR HIGH (Figure 2-35)

Command Code: 29 (hex) single track
A9 (hex) multi-track

Compares a key area in main storage with the next key area on the track (excluding R0). If a high comparison results, the track key is greater than the main storage key and Channel End, Device End, and status modifier are presented to the channel. If an equal or high comparison does not result, the track key is not greater than the main storage key and Channel End and Device End only are presented to the channel.

If a single-track search is specified, the search is confined to one track. The search is repeated until the search condition is satisfied or until two index points are detected. Upon detection of the second index, Channel End, Device End, and Unit Check are presented to the channel; and No Record Found is set in sense byte 1.

If a multi-track search is indicated, the search continues from track to track as long as the command is presented to the channel. The head number is automatically incremented at index until the search condition is satisfied, or until end of cylinder is reached. Upon detection of end of cylinder, Channel End, Device End, and Unit Check are presented to the channel; and End of Cylinder is set in sense byte 1.

Execution of this command with a key length (KL) of zero does not set a status modifier. If followed by a chained *Read Data* command, the data area read is that of the next record.

DATA ADDRESS—Specifies main storage location where key to be compared is located.

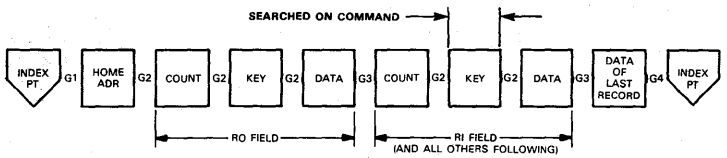
FLAGS—Used at discretion of the programmer.

COUNT—Should be equal to KL of record containing key to be compared. If count is greater than KL, the search operation is completed when the key area is read. Channel End and Device End are presented to the channel, terminating the command. Status modifier is presented if the comparison was equal or high.

If count is less than KL, the track and main storage comparison continues until the CCW count is zero. Channel End and Device End are presented after the key area and subsequent correction code bytes are read and checked. Status modifier is presented if the search on the short field was satisfied.

INITIAL STATUS—Normally zero.

SPECIAL NOTE—When command is chained from *Search ID* or *Read Count,* the key compared is in the same record as the ID or count. *Search Key Equal* bypasses R0 key field unless chained from *Search ID* command, which searched R0.



G1, G2, AND G3: FIXED LENGTH GAPS
G4: VARIABLE LENGTH GAP

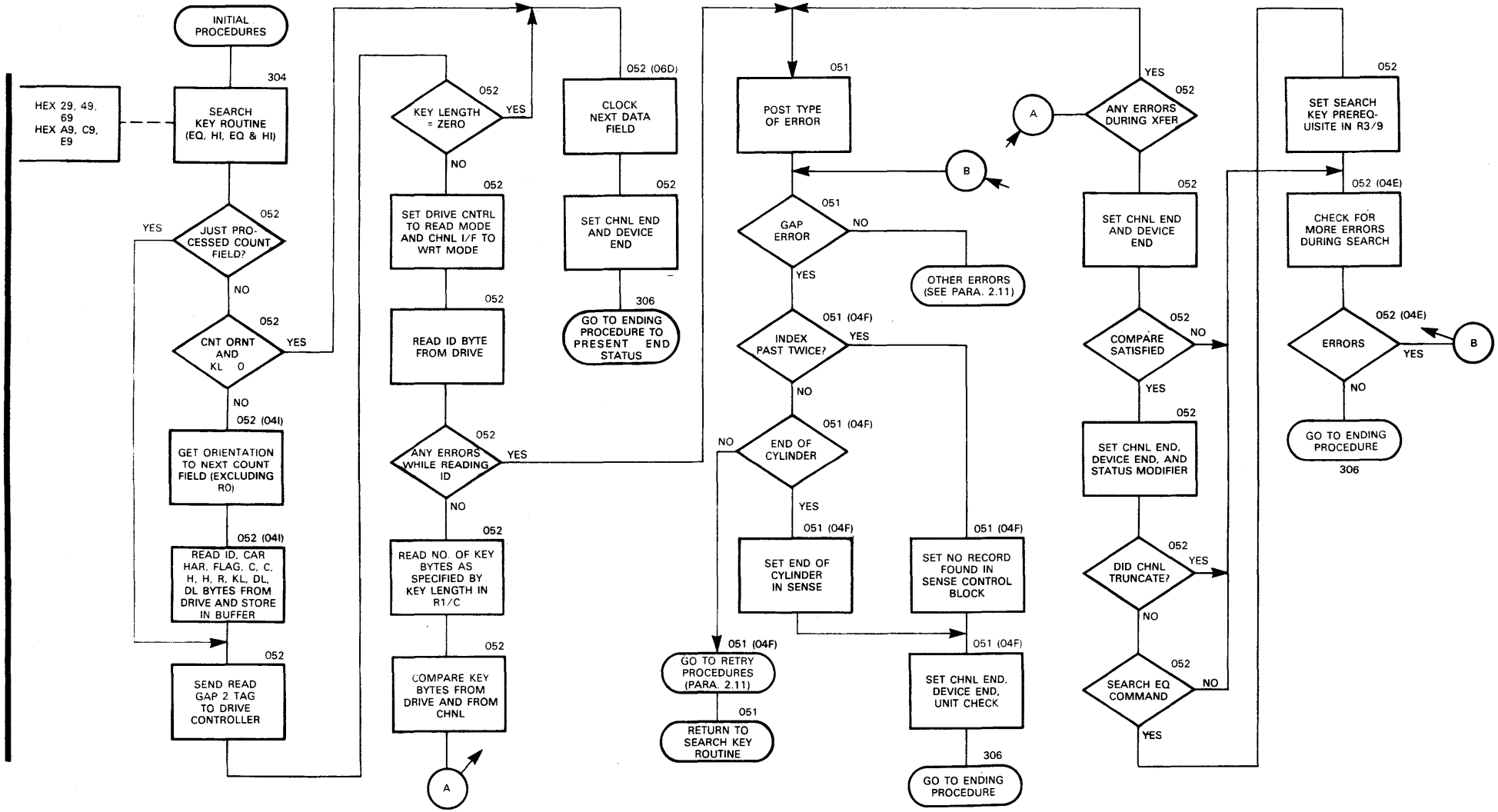


Figure 2-35. Search Key Equal, Search Key High, and Search Key Equal or High Flow Diagram

2.9 SELECTIVE RESET

The Selective Reset function is described by means of the flowchart in Figure 2-36. Refer to format 3 portion of Table 2-5 for bit definitions of the Selection Reset sense bytes.

Machine Check Multiplex (MCM) is set to obtain data from the machine. Check 1 conditions are multiplexed through the Machine Check Register (MCK). When assembling of sense bytes is completed, Check 1 and 2 conditions are reset.

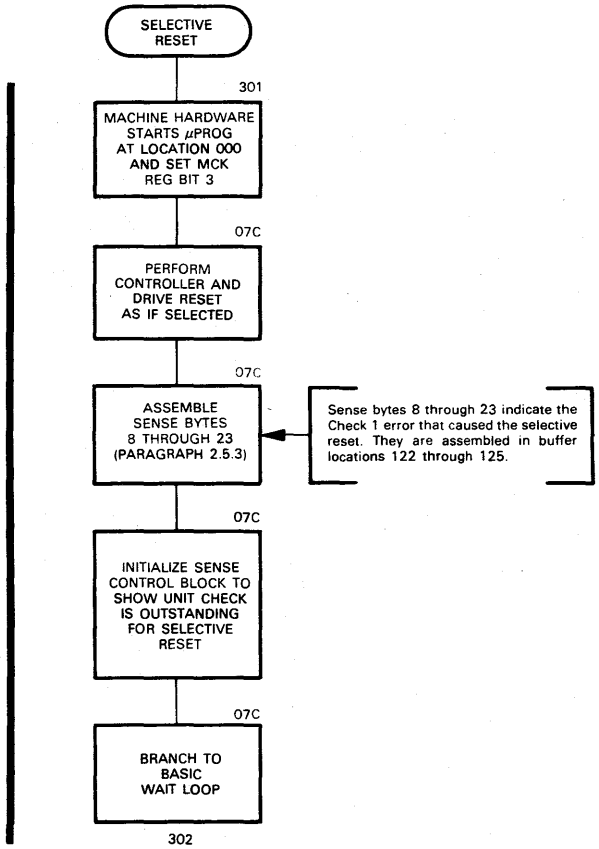


Figure 2-36. Selective Reset Flow Diagram

2.10 SYSTEM RESET

Figure 2-37 illustrates the sequence of operations during a System Reset. In addition to the conditions described in paragraph 3.5.17, System Reset also occurs as follows

- When the power for the system is turned on.
- When the RESET/LT switch is pressed.
- When the channel is off-line to the interface.
- As part of the initial program loading procedure.

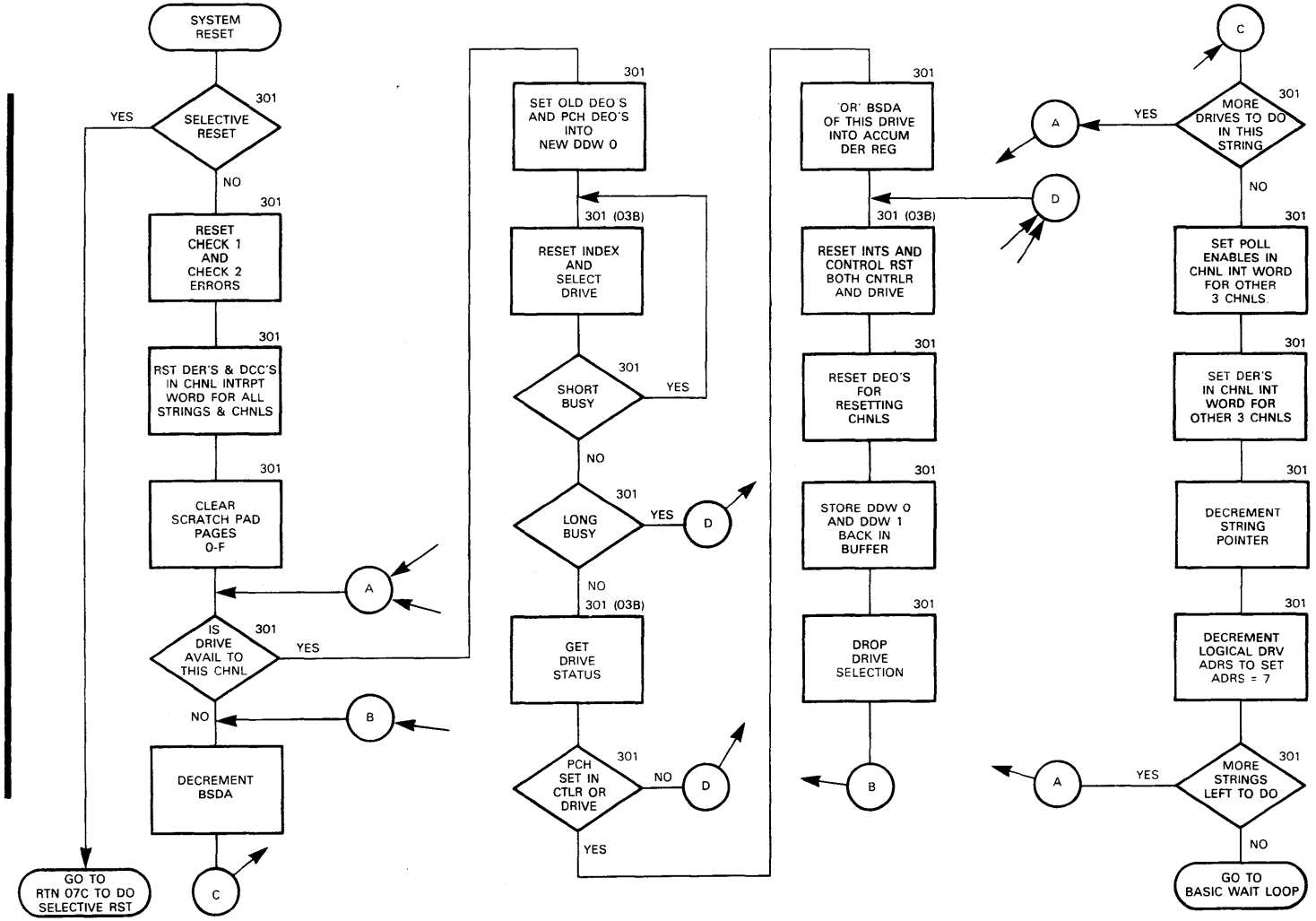


Figure 2-37. System Reset Flow Diagram

2.11 COMMAND RETRY

Command Retry is a channel/SCU procedure that causes automatic retry of an improperly executed command in a channel program. The retry does not cause an I/O interrupt, and programmed error recovery procedures are not required.

2.11.1 Check and Error-Initiated Retry

2.11.1.1 DATA CHECKS

If a correctable error occurs in the Key field, Count field, or Home Address field and the error is not in a subsequent overflow segment, the command is retried. The Key, Count, or Home Address read from the disc is placed in a buffer in the SCU. When the correctable error occurs, the SCU corrects the data in the buffer and requests the channel to reissue the command which originally caused the error. During reorientation to the record, the SCU disconnects and frees the channel. When the failing command is reexecuted, the corrected data in the buffer is used instead of the actual data from the track.

If a correctable error occurs in the Data field and the error is not in a subsequent overflow segment, the error is corrected by the system Error Recovery Procedure (ERP) using information sent to the channel in the sense data.

If, during a Read operation, an uncorrectable error occurs in the Key field, Count field, or Home Address field and the error is not in a subsequent overflow segment, the Read command is retried a maximum of 28 times. If the retry is not successful, the error becomes a permanent error. If this error occurs during a Write operation, the Write command is retried a maximum of 15 times.

If a correctable error occurs in a subsequent overflow segment in the Data field, Key field, Count field, or Home Address field, the error is corrected by the system ERP using information sent to the channel in the sense data.

2.11.1.2 SYNC CHECK

A Sync Check is executed when a sync byte is missed during a read operation. The Uncorrectable Flag is set in the microprogram to signify that the error is uncorrectable. The sequence then is the same as for a Data Check.

2.11.1.3 AM CHECK

An AM Check is executed when the Address Mark is missed on retry. The Address Mark error is uncorrectable. The sequence of the AM Check is the same as that given for an uncorrectable error of a Read operation that does not occur in a subsequent overflow segment.

2.11.1.4 PHYSICAL ID ERROR - SEEK CHECK

Physical ID Errors are detected during read-end processing of the Count Buffer against the Seek CAR-HAR in the Drive Data Buffer. If the Physical ID Error is not in an overflow segment, the Seek is retried a maximum of ten times. If the retry is not successful, the error becomes a permanent error. If the Physical ID Error occurs in an overflow segment, Operation Incomplete sense information is presented to the channel in addition to status of Channel End, Device End, and Unit Check.

2.11.1.5 SEEK INCOMPLETE ERROR

After initial status has been presented, Seek Incomplete errors are detected for all untimed commands, except Sense commands. Seek Incomplete errors are retried and, if interrupted or incomplete on retry, cause Equipment Check sense information to be presented to the channel for use in the Retry Command Check.

2.11.2 Defective or Alternate Track

When an alternate or defective track condition is detected before data transfer begins, the SCU determines the location of the alternate or defective track (from R0 on the track), initiates a seek to this track, orients on index, and reissues the original command.

2.11.3 Command and Data Overrun

When a command overrun or late command chaining condition occurs because of interference from another channel or the CPU, the SCU initiates a retry of the command that was late.

When a data overrun occurs, a retry of the command is initiated, except when the data overrun occurs during one of the following operations:

- During a record overflow operation in the subsequent overflow segments.
- During a format write operation.

2.11.4 Padding

Padding occurs when the Write gate is on after initial status presentation and the command is not a Format Write. The padding sequence tests to see if the drive has the disconnected auto padding feature present. If the feature is active, the SCU presents retry status and lets the drive disconnect from the SCU to finish padding and reconnect as if it were a disconnected command chaining sequence. If this feature is not installed in the drive, padding will cause the command to be retried and the SCU to wait for index before reconnecting to the command.

2.11.5 Write Offset

Write Offset occurs when the Write command detects that the access is offset. Write Offset causes the command to be retried after offset is reset.

2.11.6 Invalid Count Field Sync Detected

Invalid Count field sync is detected when the index occurs prior to the third ECC byte of the Count field and after the Lock VFO to Data Point in the gap preceding the Count field. Invalid Count field sync causes the command to be retried.

2.11.7 Index Detected in Data Field

Index is detected in the Data field when the index occurs between the Lock to VFO Point and the Source ID Byte of the Data field. Index detected in the Data field causes the command to be retried.

2.11.8 Command Retry Flowcharts

Figures 2-38 through 2-45 which follow contain flowcharts for the Command Retry procedures discussed in the preceding paragraphs. The flowcharts are presented in the following order:

- Data, Sync, and AM Checks (Figure 2-38)
- Physical ID and Seek Errors (Figure 2-39)
- Defective Track (Figure 2-40)
- Alternate Track (Figure 2-41)
- Data and Command Overruns (Figure 2-42)
- Padding (Figure 2-43)
- Write Offset (Figure 2-44)
- Invalid Count Field Sync Detected and Index Detected in Data Field (Figure 2-45)

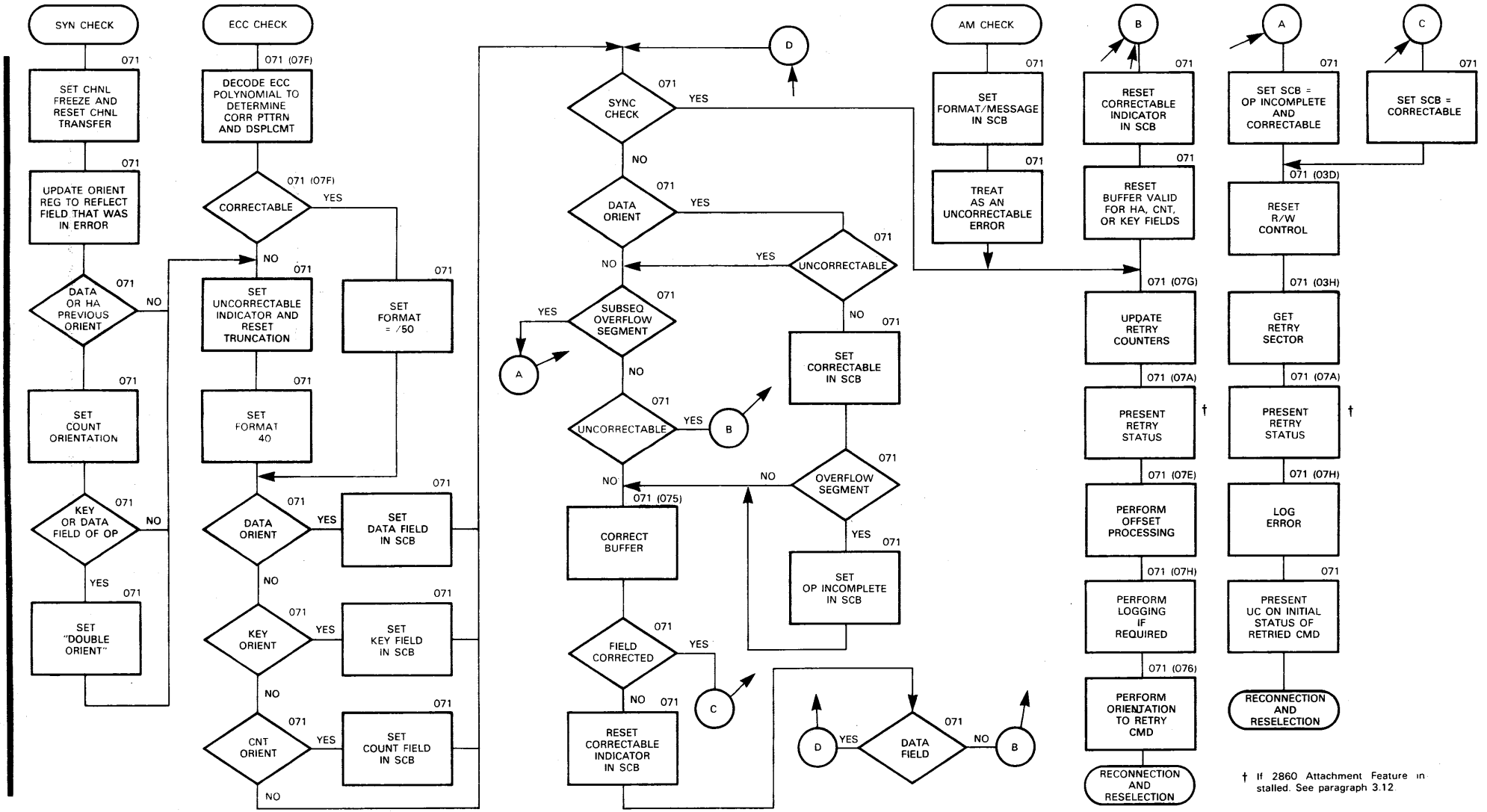


Figure 2-38. Command Retry, Data, Sync, and AM Checks

† If 2860 Attachment Feature installed. See paragraph 3.12.

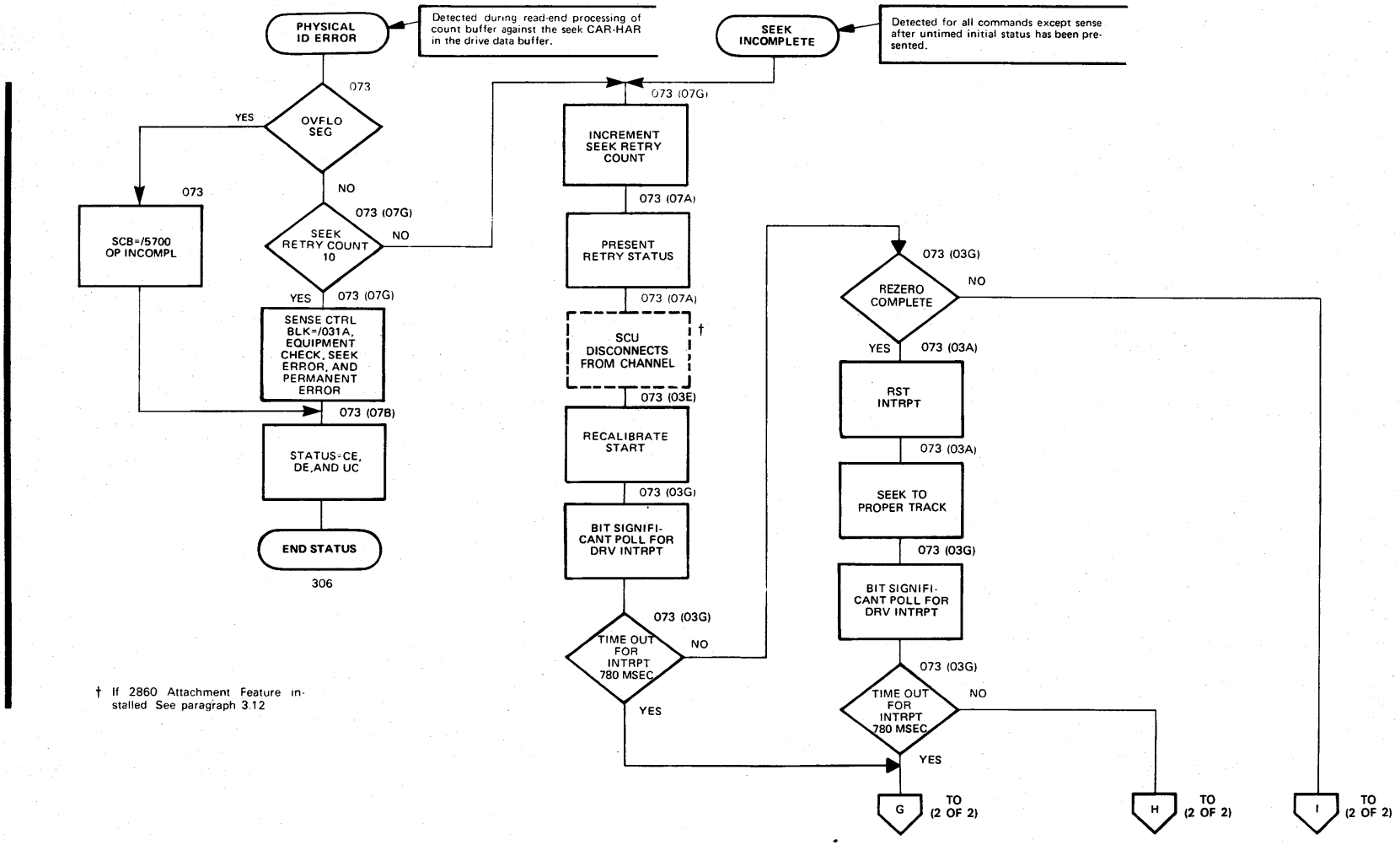
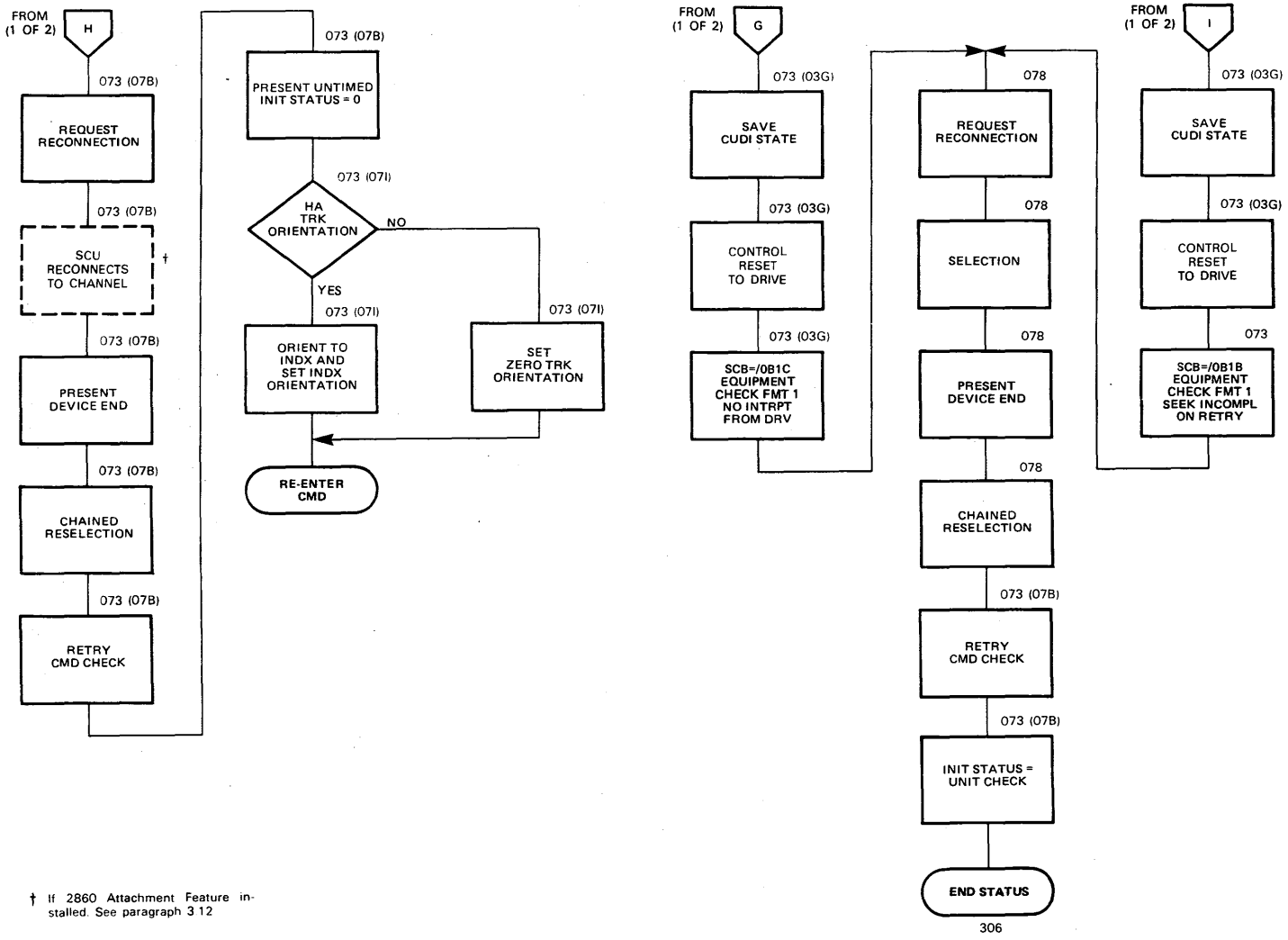
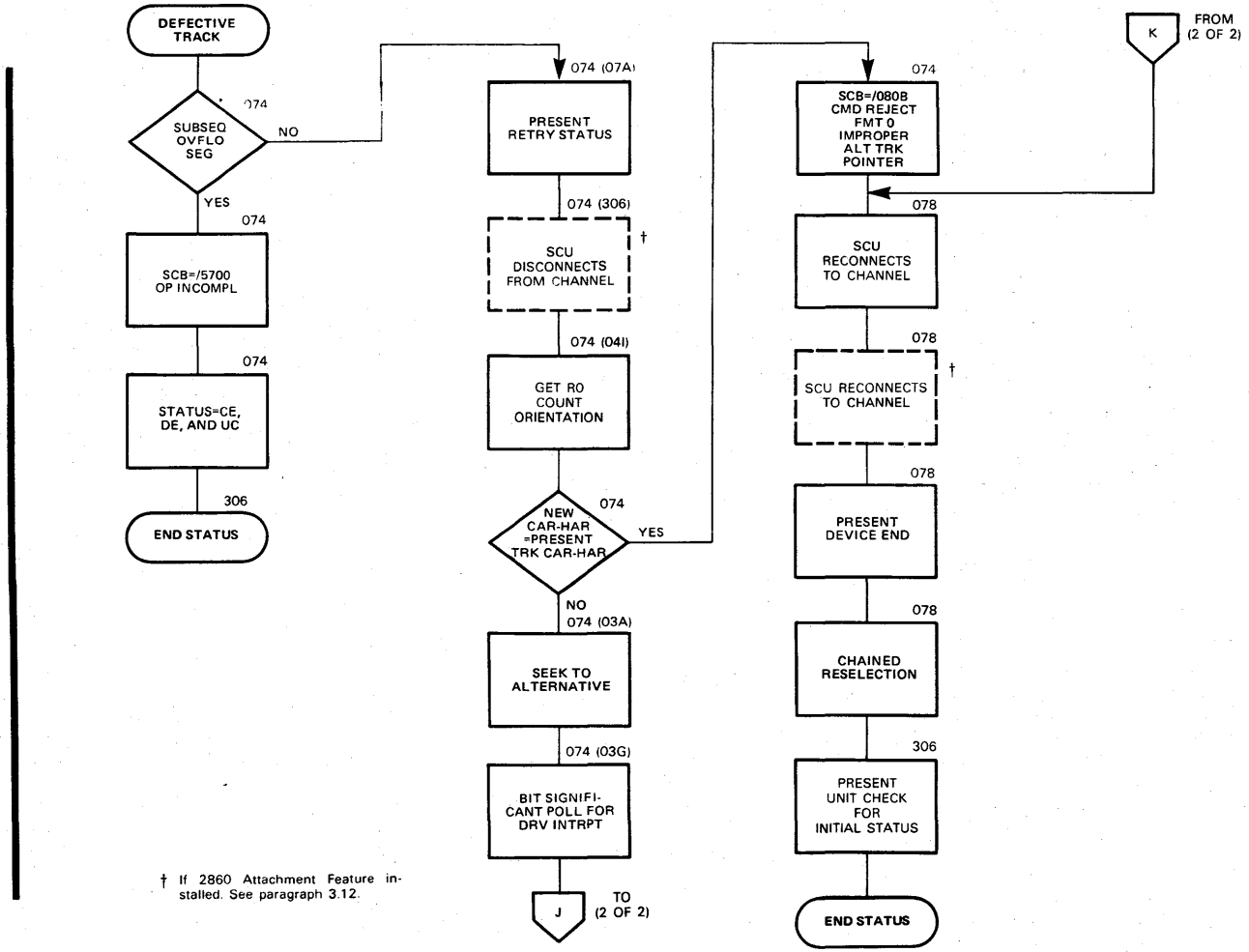


Figure 2-39. Command Retry Physical ID and Seek Errors (1 of 2)



† If 2860 Attachment Feature installed. See paragraph 3.12

Figure 2-39. Command Retry, Physical ID and Seek Errors (2 of 2)



† If 2860 Attachment Feature installed. See paragraph 3.12.

Figure 2-40. Command Retry, Defective Track (1 of 2)

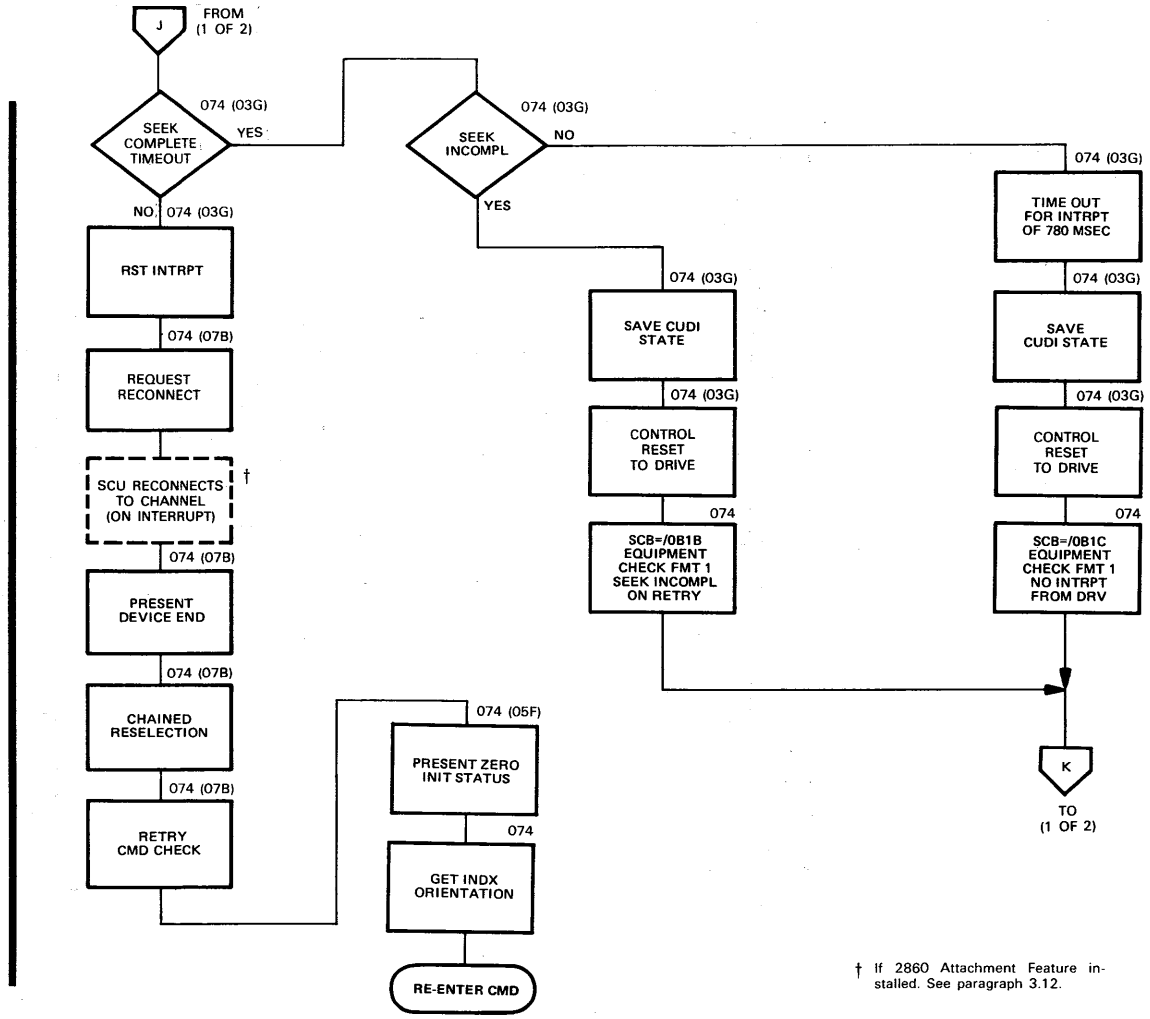
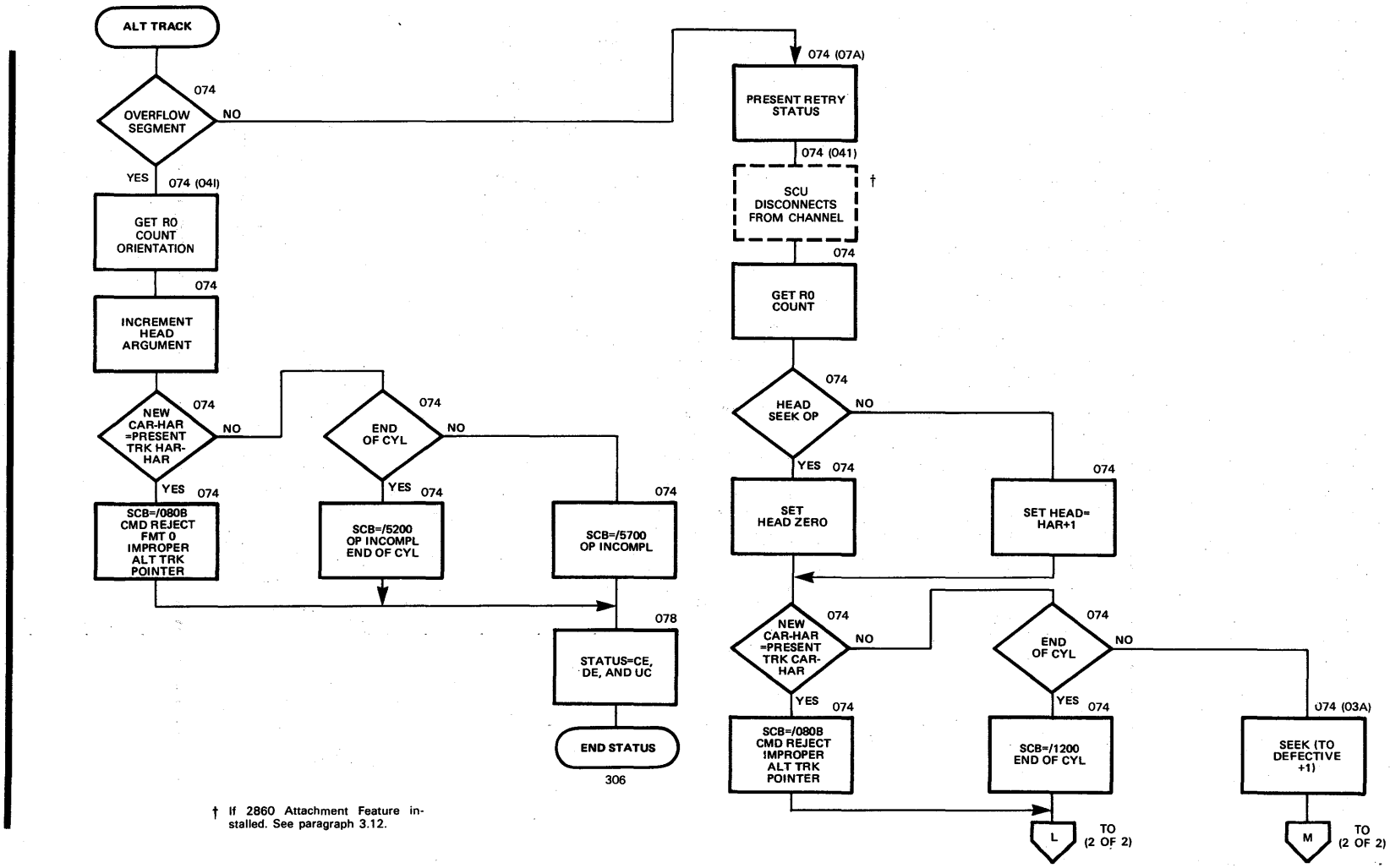
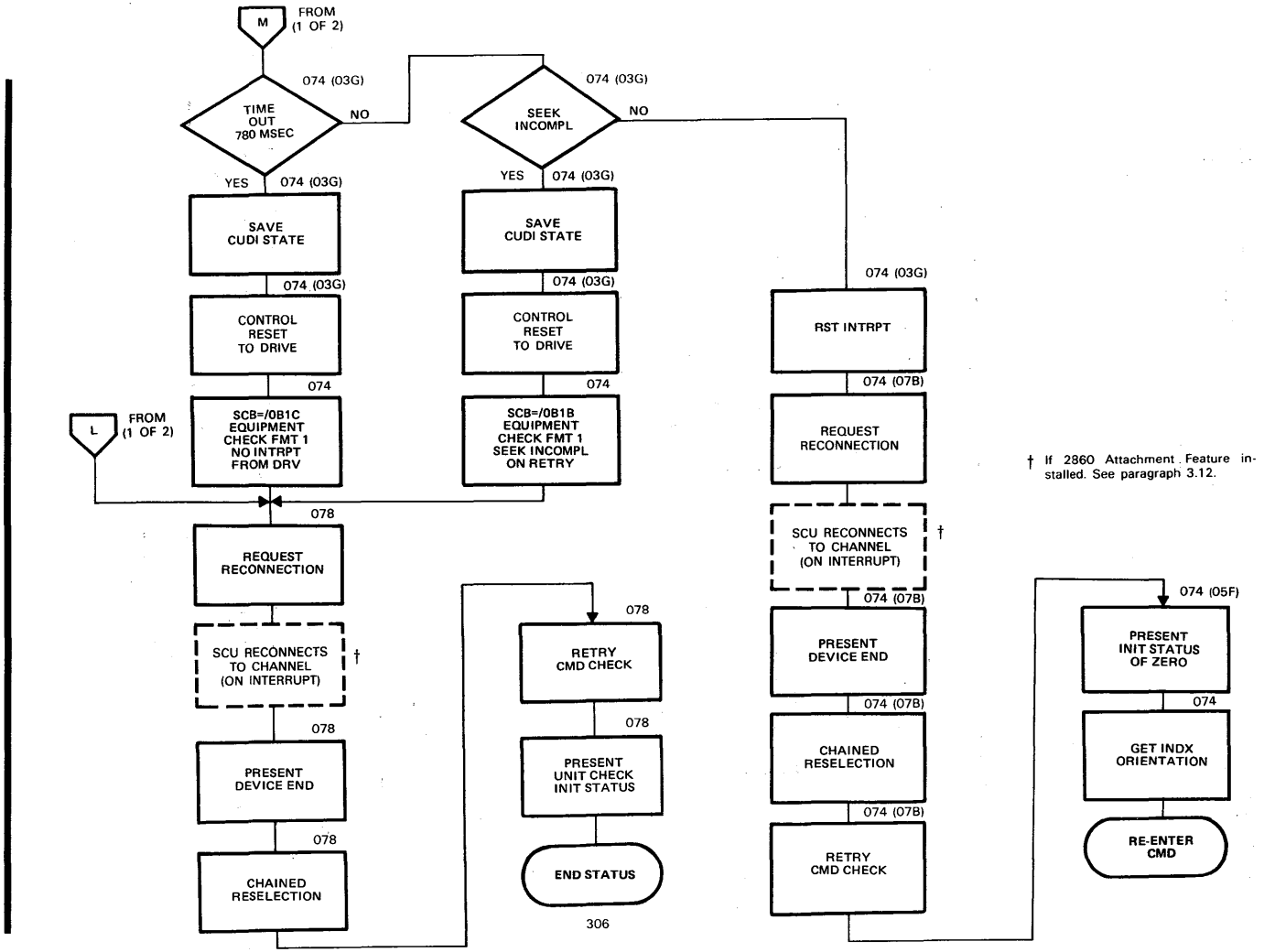


Figure 2-40. Command Retry, Defective Track (2 of 2)



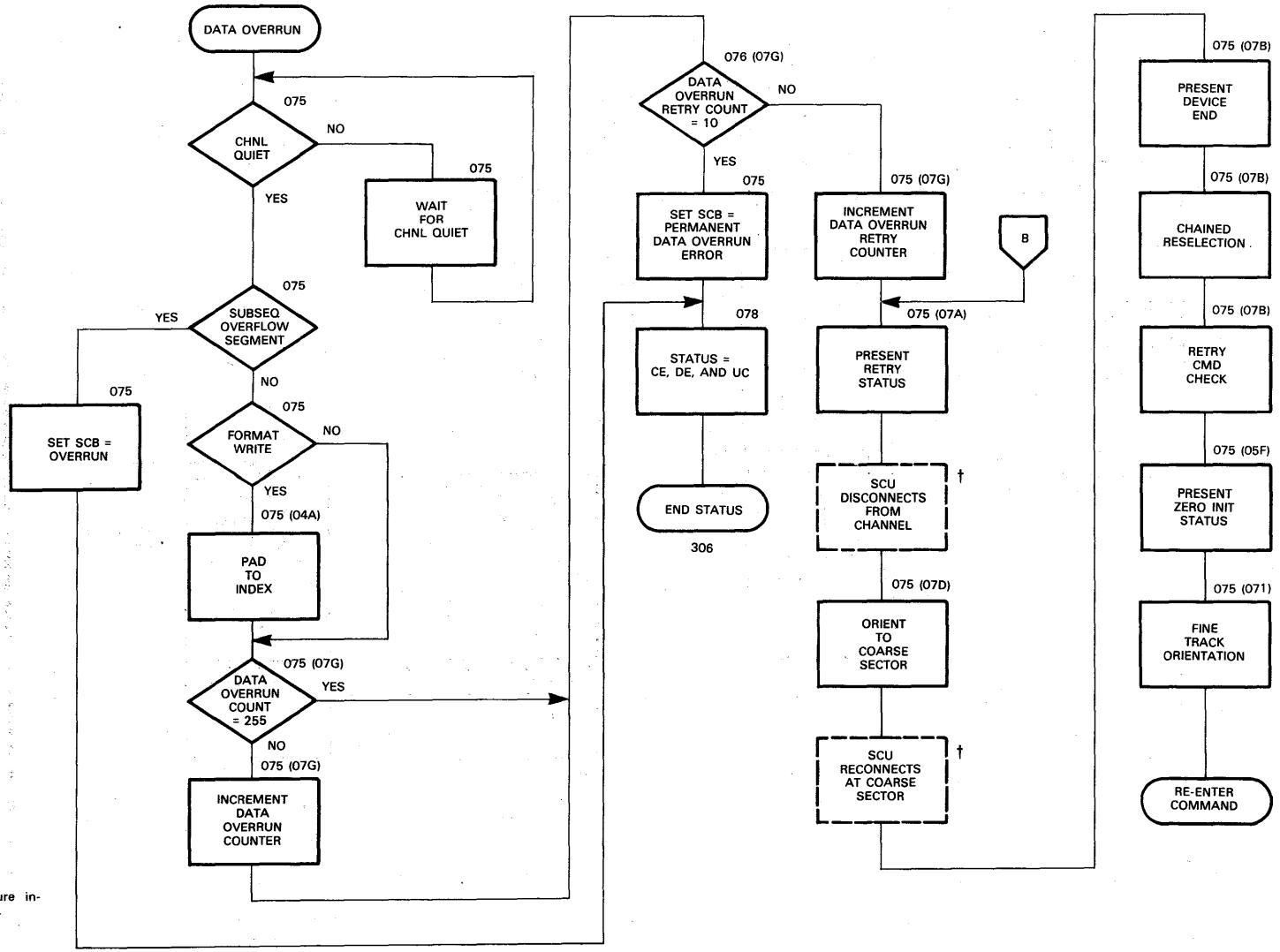
† If 2860 Attachment Feature installed. See paragraph 3.12.

Figure 2-41. Command Retry, Alternate Track (1 of 2)



† If 2860 Attachment Feature installed. See paragraph 3.12.

Figure 2-41. Command Retry, Alternate Track (2 of 2)



† If 2860 Attachment Feature installed. See paragraph 3.12.

Figure 2-42. Command Retry, Data and Command Overruns (1 of 2)

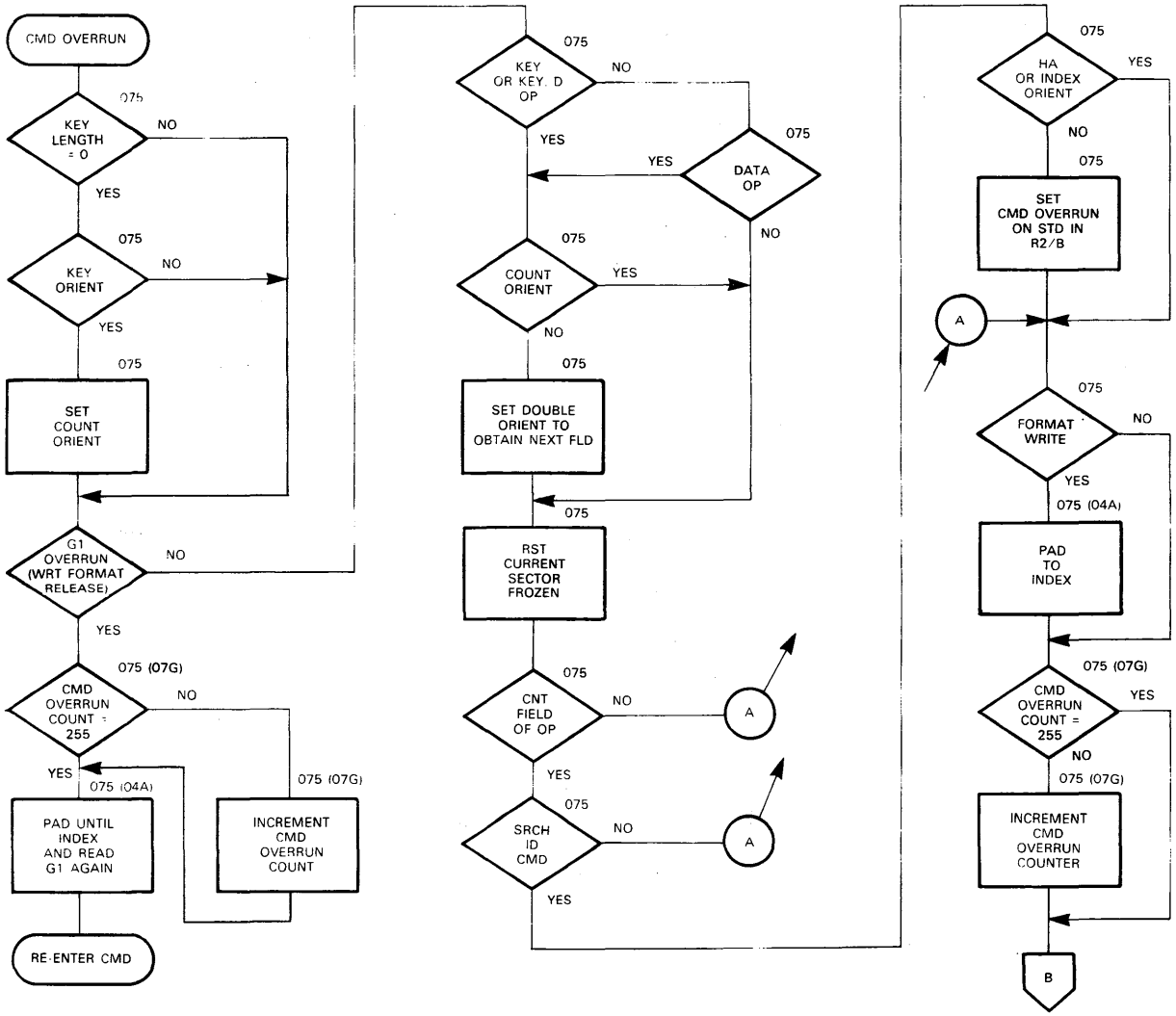


Figure 2-42. Command Retry, Data and Command Overruns (2 of 2)

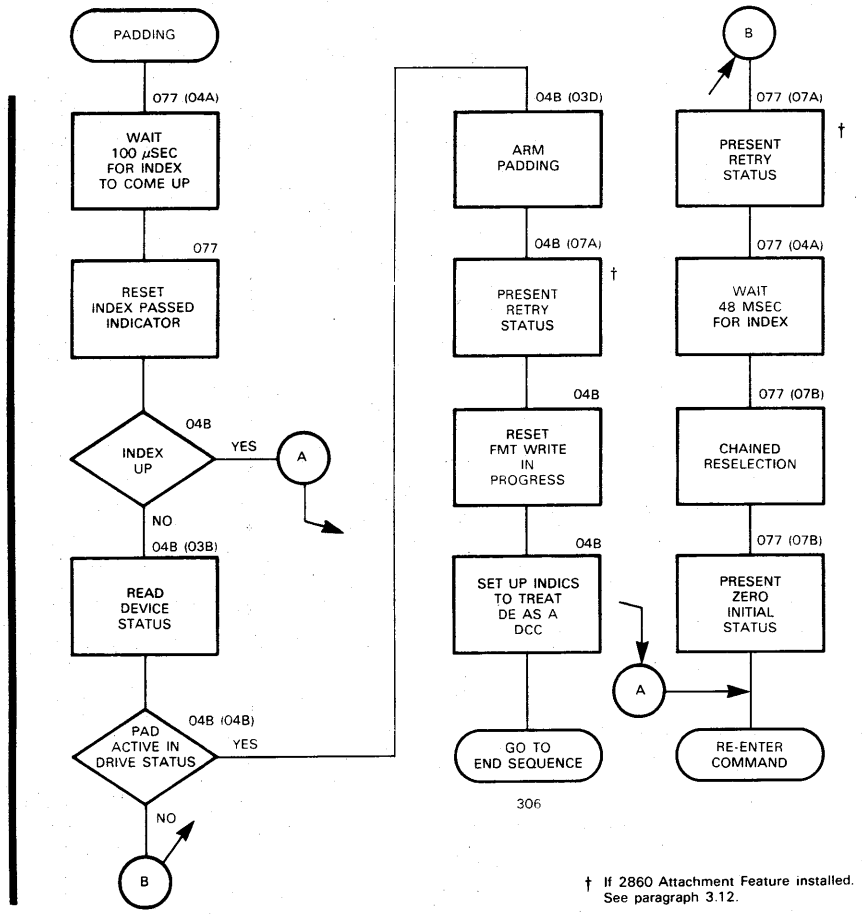


Figure 2-43. Command Retry, Padding

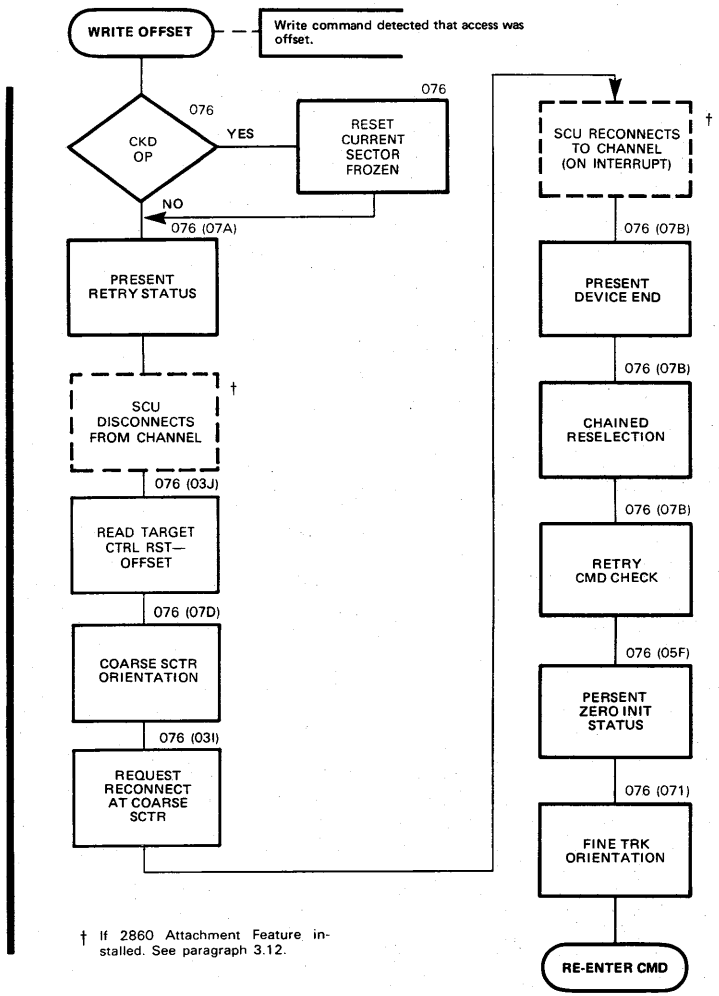


Figure 2-44. Command Retry, Write Offset

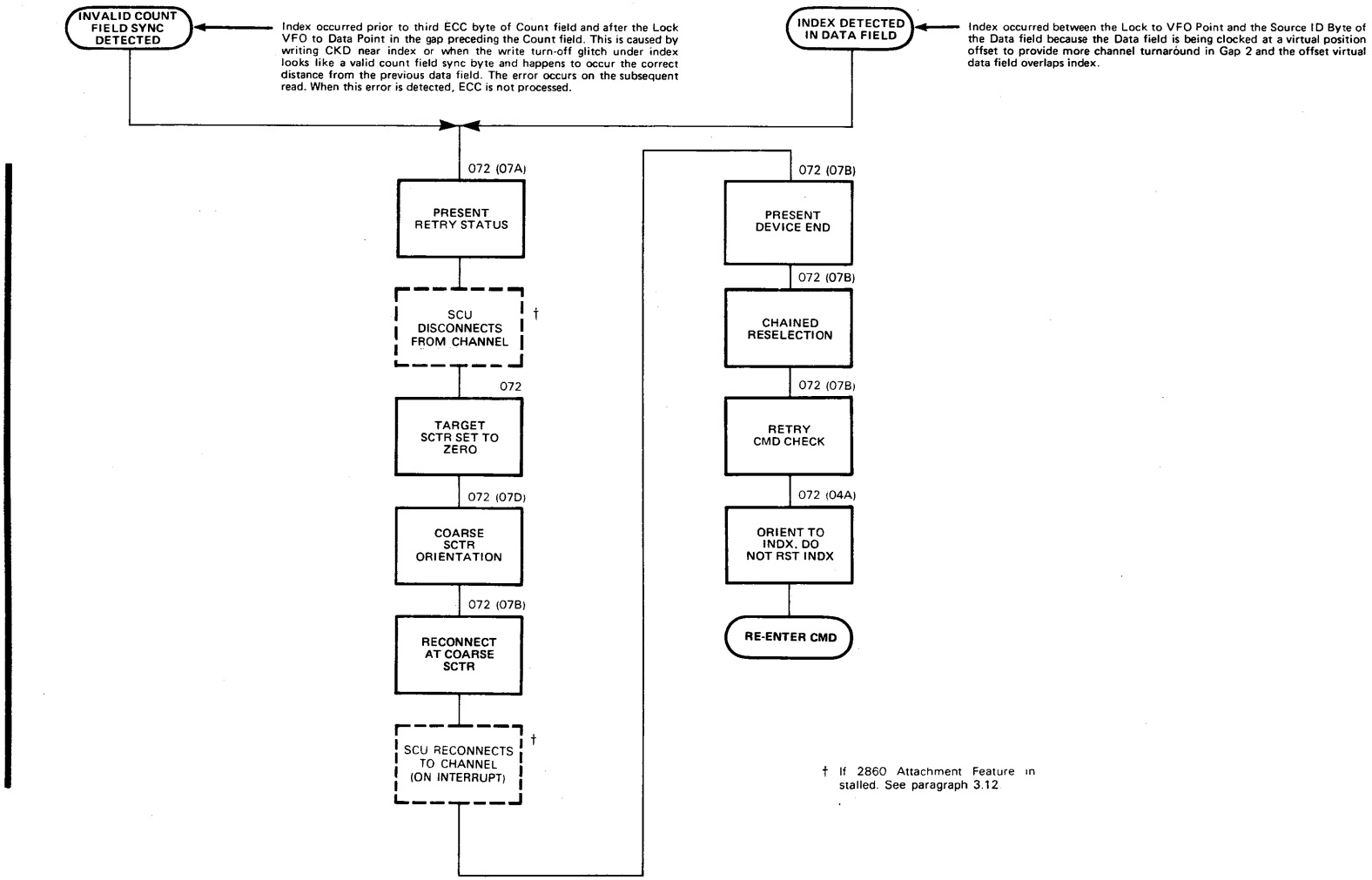


Figure 2-45. Command Retry, Invalid Count Field Sync Detected and Index Detected in Data Field

This page intentionally left blank.

2.12 ROTATIONAL POSITION SENSING

2.12.1 Introduction

Rotational position sensing (RPS) is a subsystem feature that allows the channel and the SCU to be released during most of a record search time, thereby increasing their availability for other operations. Two commands, *Read Sector* and *Set Sector,* are associated with RPS operation. These commands are described in the following paragraphs.

2.12.2 Read Sector

The *Read Sector* command transfers one byte of information from the SCU to main storage (Figure 2-46). This byte provides the sector number (angular displacement from index) of the last record processed. If the last record processed was an overflow record, the sector number returned is that of the first segment processed in the current command chain.

Execution of a *Read Sector* command causes loss of record orientation previously established. Therefore, if further processing of records of the same track (or cylinder) is desired, a search for the desired record must be made.

2.12.3 Set Sector

Refer to Figure 2-46. The *Set Sector* command transfers one byte of data from main storage to the SCU. This byte specifies one of 128 possible sector numbers per track. The byte value is checked for validity by the SCU. If the value is proper, the SCU generates Channel End and allows the channel to disconnect.

When the desired sector (which has an adjusted reselection delay factor) is reached, the SCU signals Device End. The channel can then connect to continue the command chain. If the channel does not respond within 130 microseconds, connection is tried on subsequent revolutions.

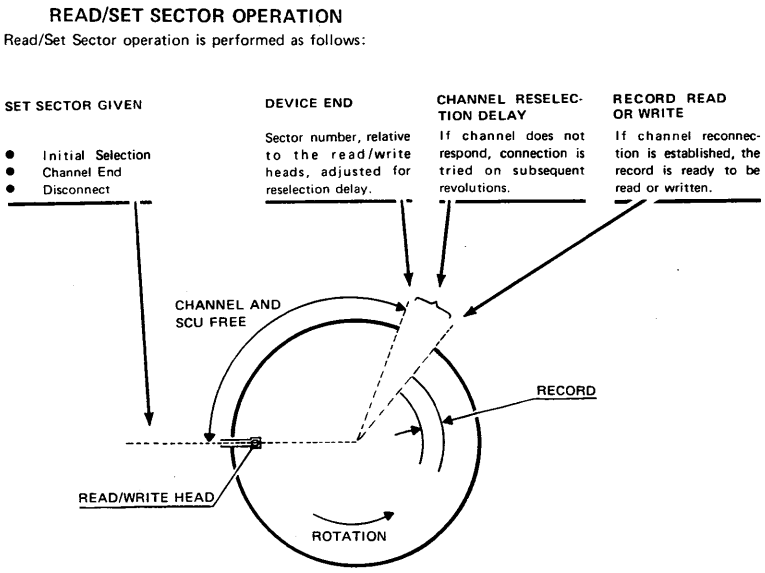


Figure 2-46 Read/Set Sector Operation

2.12.4 Disc and Track Layout

The track layout on individual recording surfaces in the disc pack is shown in Figure 2-47. A recording surface is divided into 128 sectors (0-127). The time for pack rotation through one sector is approximately 130 microseconds.

When a record is written, the sector number may be stored in the using system by a *Read Sector* command. For example, assume Record 3 is written. Referring to Figure 2-48, Sector 9 is stored by using the *Read Sector* command. This sector number can be calculated by the using system if the record lengths are fixed.

The sector number is set in the Target Register of the selected drive. The sector that is set (target sector) is always two less than the desired sector. When the sector counter and Target Register compare, SCU raises Request In to the channel. Then the channel can command a Search ID for the desired record with minimum loss of system/CPU time.

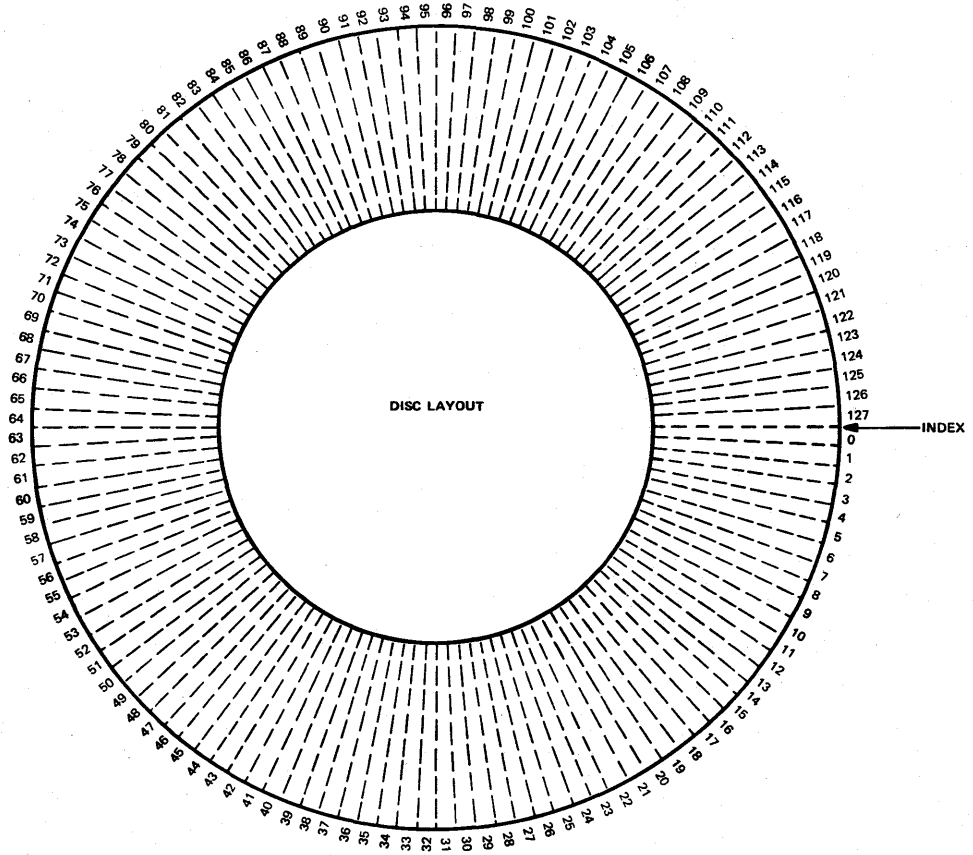


Figure 2-47. Track Sector Layout

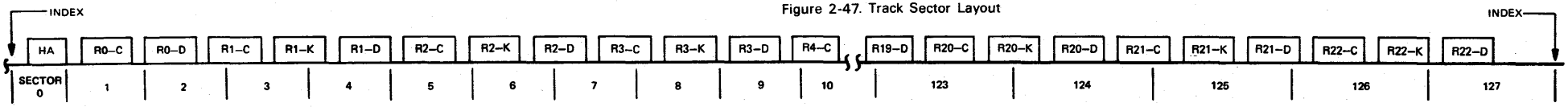


Figure 2-48. Typical Track Field Layout

2.13 MULTITRACK OPERATION

The flowchart in Figure 2-49 illustrates the sequence for a multitrack Read or Search operation. A multitrack coded Search or Read command must be executed to perform multitrack operation. During multitrack operations, the drive Head Address Register (HAR) is automatically incremented, thus eliminating the need for *Seek Head* commands in a chained Read or Search command.

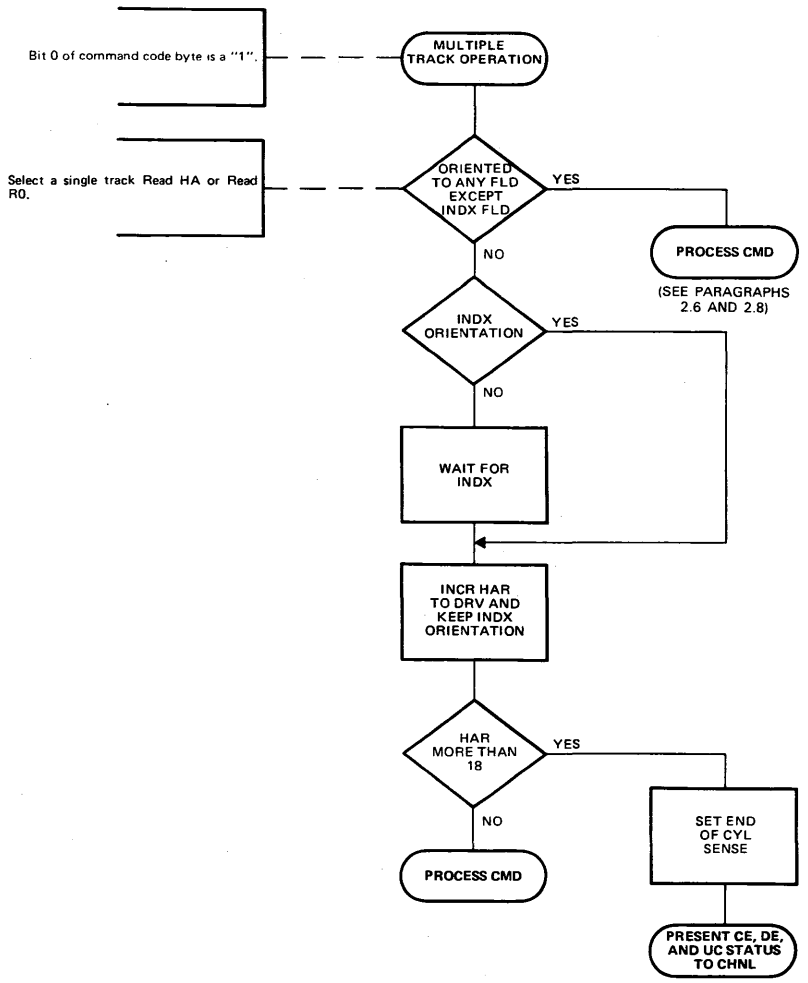


Figure 2-49. Multitrack Operation

2.14 OVERFLOW RECORD

2.14.1 Introduction

Overflow Records allow logical data records longer than one track to be read or written on the disc. Format writing of an overflow record is controlled by the using system. Read and nonformatting write operations are controlled by the SCU.

Formatting and processing of Overflow Records are described in Figure 2-50 and the following paragraphs.

2.14.2 Formatting

These segments are written using *Write Special Count, Key, and Data* commands. The sequence is as follows (letters refer to corresponding tracks of the record):

(A) Search ID (R1)
Write Special CKD (1st segment)
Seek Head (next track)

(B) Search ID (R0)
Write Special CKD (2nd segment)
Seek Head (next track)

(C) Search ID (R0)
Write Special CKD (3rd segment)
Seek Head (next track)

(D) Search ID (R0)
Write CKD (4th segment)

2.14.3 Processing

The SCU switches to the next head on all overflow records (flag byte bit 4 = 1) for read and nonformatting write commands. The command sequence consists of a Search ID (R2) succeeded by one of the following: Read Key and Data, Read Data, Write Key and Data, or Write Data (read or write segments 1, 2, 3, and 4 with auto head switching at index, controlled by the SCU).

Errors on the first segment (excluding correctable data field data checks) are retried by the SCU.

An error on the second or subsequent segments (data check, defective track, etc.) sets sense byte 1, bit 7 (Operation Incomplete). A restart CCW is provided to the using system error recovery procedure (ERP) in sense byte 3. After the ERP has corrected the problem, the restart CCW is sent to the SCU to resume the operation.

00000110 (06) A read command was in progress.

00000101 (05) A write command was in progress.

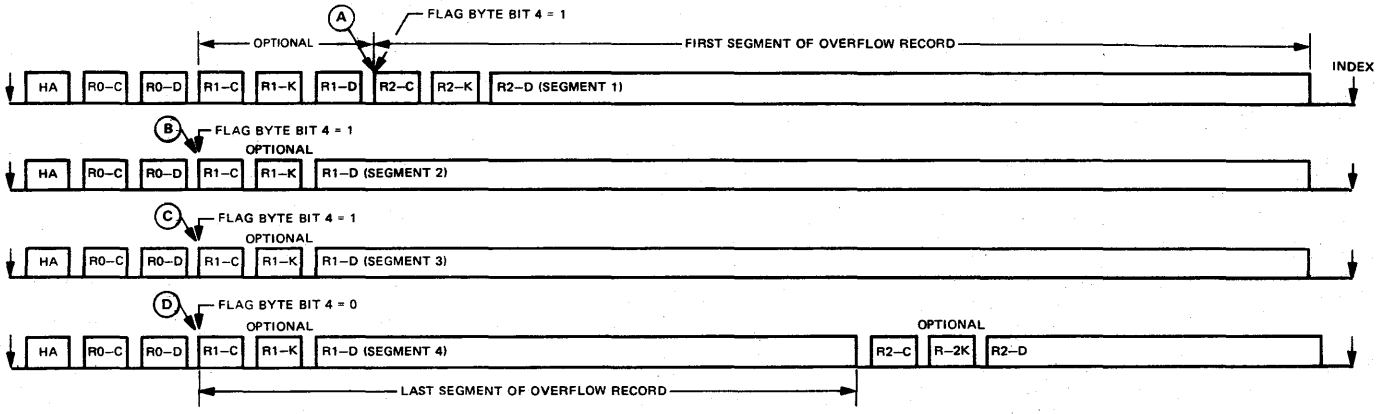


Figure 2-50. Overflow Record Operation

2.15 MULTICHANNEL OPERATION

Multichannel operation permits the SCU to communicate with up to four CPUs over any of four channels (maximum). Such operation allows any of the eight disc drives controlled by the SCU through the Controller to be reserved exclusively for any of the channels so enabled. A channel is enabled for multichannel operation by means of a corresponding CHANNEL switch on the SCU Operator Control Console. There are four switches, one for each channel. Setting the switch to the enable position merely enables the channel for operation, actual channel switching and device reservation is controlled by the channel program.

Reservation of a device is accomplished by executing a *Device Reserve* command. A device address reserved to one channel, or being used by one channel, presents a Busy status to any other channel if selection is attempted. Release of a device previously reserved is accomplished by executing a *Device Release* command. In addition, 24 bytes of sense information are transferred to the channel when either the *Device Reserve* or *Device Release* command is executed. (Refer to paragraph 3.11 for a detailed description of multichannel switch operation.)

The block diagram in Figure 2-51 illustrates multichannel operation considering two channels. An associated table for the *Device Reserve* and *Device Release* commands are presented in Table 2-6. Refer to Figure 2-16 for the corresponding flowchart of these commands.

TABLE 2-6. DEVICE RESERVE/RELEASE COMMAND FUNCTIONS

COMMAND	HEX CODE	FUNCTION	ERROR TYPE
Device Reserve	B4	1 Set Bit 0 of R0, page 9 of Register File in micro processor. 2. Transfer 24 bytes of sense data to channel. 3. Send ending status to channel.	If command is preceded by a Set File Mask in the chain, Command Reject will be set in sense data and Unit Check will be initial status.
Device Release	94	1. Reset Bit 0 of R0, page 9 of Register File in microprocessor. 2. Transfer 24 bytes of sense data to the channel. 3. Send ending status to channel.	

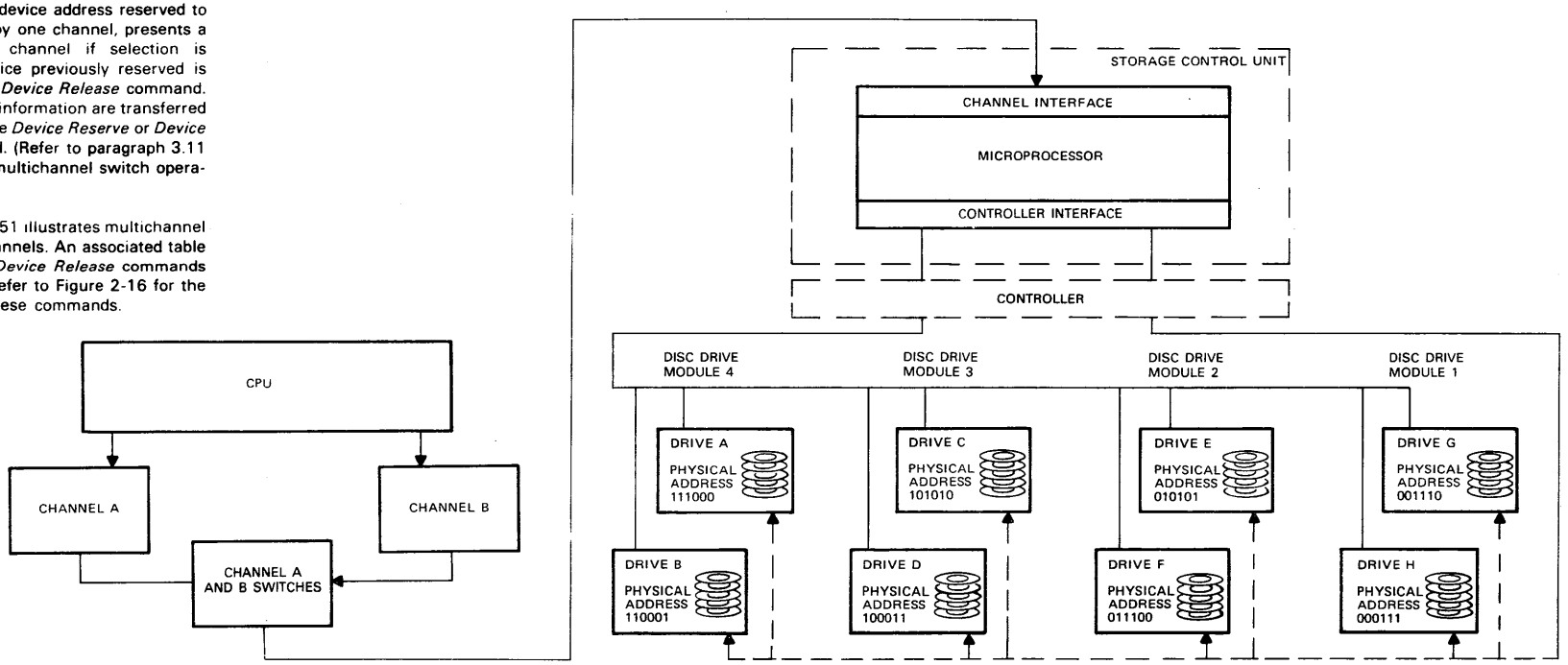


Figure 2-51. Multichannel Operation Block Diagram

SECTION 3. CHANNEL INTERFACE DESCRIPTION

3.1 GENERAL

This section describes the I/O interface between the System/370 channel and the 3672 Storage Control Unit (SCU). In general, this interface is characterized by the following features:

- Provides for common connection and communication between an IBM System/370 block multiplex channel and SCU.
- Provides signal sequence and information flow common to all SCUs.
- Is defined by IBM, and amended by Memorex for the 3672 SCU in terms of interface line groupings and additional special controls.
- Logically connects only one SCU to the channel at a time.

The standard channel interface is composed of 37 lines which are divided into five groups: bus lines, tag lines, scan controls, interlock lines, and special controls. These groups, made up of both inbound (to the CPU) and outbound (to the SCU) lines, are shown in Figure 3-1.

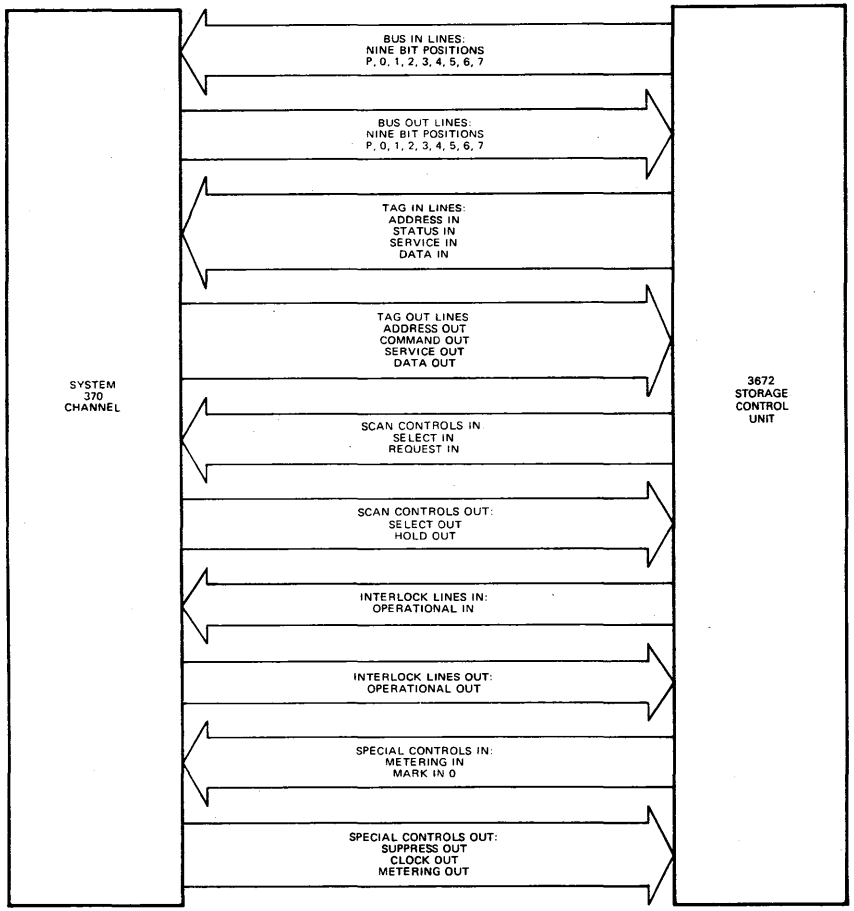


Figure 3-1. Channel Interface Signal Groups

Bus lines transfer information between the SCU and the channel. Each bus is a set of nine lines consisting of one parity line and eight information lines. Information on a bus is arranged so that bit position 7 carries the least significant bit within an eight-bit byte. The most significant bit is in position 0 and the intervening bits are in ascending order from position 6 to 1.

Tag lines are used for special sequences and for interlocking and controlling information on the buses.

Scan controls enable an SCU to request service from the channel on a priority basis. These controls also allow the channel to contact the SCU.

Interlock lines lock out nonselected SCUs, permitting only one SCU to be serviced at a time. The interlock lines also gate the tag lines, and when down, initiate reset of the interface operations.

Special controls are used to provide special functions: conditioning of usage meters, CPU interlock control, and a marker tag line for command retry.

3.2 CHANNEL INTERFACE LINES DESCRIPTION

3.2.1 Inbound Lines

Bus In

The Bus In lines transfer data, address, status, and sense information to the channel. The inbound tag lines indicate the type of information transferred over the Bus In lines. Either the tag and interlock lines, or scan controls determine the period during which signals are valid. The duration of this period is

- when OPERATIONAL IN is up,
- within 100 nanoseconds after the rise of an identifying inbound tag to the responding outbound tag, or
- until SELECT OUT drops in an SCU busy sequence.

Address In

Signal ADDRESS IN informs the channel when the address of the currently selected SCU and disc drive has been placed on the Bus In lines. The channel responds by raising COMMAND OUT. The ADDRESS IN signal remains up until COMMAND OUT is generated and must drop so that COMMAND OUT may drop. It cannot be up concurrently with any other inbound tag.

Status In

Signal STATUS IN informs the channel when the selected SCU has placed status information on the Bus In lines. The channel responds by raising SERVICE OUT (accept) or COMMAND OUT (stack). STATUS IN remains up until an outbound tag is generated, or until SELECT OUT drops in an SCU busy sequence. STATUS IN must drop so that the responding outbound tag may drop. It cannot be up concurrently with any other inbound tag.

Service In

Signal SERVICE IN informs the channel when the selected SCU is ready to transmit or receive a byte of information. It remains up until the channel responds by raising SERVICE OUT or COMMAND OUT or, during an interface disconnect sequence, by ADDRESS OUT. The conditions for generating SERVICE IN are as follows:

- during read and sense operations when information is available on the Bus In lines.
- during write and control operations when information is required on the Bus Out lines.

SERVICE IN cannot be up concurrently with any other inbound tag. When the channel does not respond in time to the preceding SERVICE IN, an overrun condition occurs. SERVICE IN does not drop if an outbound tag has not risen and does not rise if SERVICE OUT has not dropped.

Data In

Signal DATA IN informs the channel when the selected SCU is ready to transmit or receive a byte of information. It remains up until the channel responds by raising DATA OUT or COMMAND OUT or, during an interface disconnect sequence, by ADDRESS OUT. The conditions for generating DATA IN are as follows:

- during read and sense operations when information is available on the Bus In lines.
- during write and control operations when information is required on the Bus Out lines.

DATA IN cannot be up concurrently with any other inbound tag. When the channel does not respond in time to the preceding DATA IN, an overrun condition occurs. DATA IN does not drop if an outbound tag has not risen and does not rise if DATA OUT has not dropped.

Select In

Signal SELECT IN extends the SELECT OUT signal from the jumper in the terminator block to the channel. It provides a return path to the channel for the SELECT OUT signal.

Request In

Signal REQUEST IN indicates that the SCU is requesting a selection sequence to present status information. When OPERATIONAL IN rises, REQUEST IN is dropped unless

- additional selection sequences are required,
- when the SCU is no longer ready to present the status information, or
- when the selection requirements are met by another channel of a multi-channel SCU.

If the sequence meets the service requirements, REQUEST IN falls not more than 250 nanoseconds after OPERATIONAL IN goes low. If the request for status presentation is suppressible, REQUEST IN does not remain up when SUPPRESS OUT is up. During a suppressible operation, REQUEST IN falls at the SCU within 1.5 microseconds after the rise of SUPPRESS OUT at the SCU.

Operational In

Signal OPERATIONAL IN informs the channel that an SCU has been selected, except during an SCU busy sequence. OPERATIONAL IN remains up for the duration of the selection. The address byte transmitted over the Bus In lines identifies the selected SCU. OPERATIONAL IN rises only when the inbound SELECT OUT to the SCU is up and the outbound SELECT OUT is down. The selected SCU blocks the transfer of SELECT OUT to the next SCU. OPERATIONAL IN drops only after SELECT OUT drops.

When OPERATIONAL IN is raised, it remains up until all required information is transmitted between the channel and the SCU. If SELECT OUT is down, OPERATIONAL IN drops after the rise of the outbound tag associated with the transfer of the last byte of information. With the exception of REQUEST IN or METERING IN, all inbound signals are reset within 1.5 microseconds of the fall of OPERATIONAL IN at the SCU.

Metering In

Signal METERING IN is transmitted over a line from all attached SCUs and is used to condition the CPU meter for operation. METERING IN rises and falls concurrently with OPERATIONAL IN for any interface signaling sequence. METERING IN is not raised during the following conditions:

- between generation and acceptance of Device End,
- between generation of Device End and acceptance of the next command during chaining,
- while a device is waiting for initiation of an automatic start.

Mark In 0

Signal MARK IN 0 is used as a marker tag to indicate that the SCU is requesting command retry. Associated with MARK IN 0 is the retry status. MARK IN 0 is up when OPERATIONAL IN is up. Retry status is associated with MARK IN 0.

3.2.2 Outbound Lines

Bus Out

The Bus Out lines transfer data, address, and commands to the SCUs. The outbound tag lines indicate the type of information transferred over the Bus Out lines. Either the tag and interlock lines, or scan controls determine the period during which signals are valid. The duration of this period is

- from the rise of ADDRESS OUT to the rise of OPERATIONAL IN or SELECT IN during transmission of the SCU and disc drive address signal,
- until STATUS IN drops in an SCU busy sequence, or
- from the rise of the identifying outbound tag to the fall of the responding inbound tag during the transmission of any other type of information.

Address Out

Signal ADDRESS OUT informs the SCU to decode the SCU and disc drive address on the Bus Out lines. Except in an SCU busy sequence, the SCU responds when recognizing the address by raising OPERATIONAL IN when SELECT OUT rises with ADDRESS OUT still up. ADDRESS OUT rises not less than 250 nanoseconds after the SCU and disc drive address have been placed on the Bus Out lines. ADDRESS OUT must be down for not less than 250 nanoseconds before being raised for disc drive selection. Selection of the SCU is cancelled if ADDRESS OUT drops before SELECT OUT rises.

Except for a disconnect sequence, ADDRESS OUT must rise only when SELECT OUT, HOLD OUT, SELECT IN, STATUS IN, and OPERATIONAL IN are down. If HOLD OUT is down and ADDRESS OUT rises, or ADDRESS OUT is up and HOLD OUT drops, the SCU will drop its OPERATIONAL IN and thereby disconnect from the interface. ADDRESS OUT must remain up until

OPERATIONAL IN drops. This occurs within 6 microseconds after the disconnect indication (this sequence occurs during a *Halt I/O* instruction). Mechanical motion in process continues to a normal stopping point. Status information is generated and presented to the channel when appropriate; in this case, ADDRESS OUT may be up concurrently with another outbound tag.

Command Out

Signal COMMAND OUT informs the selected SCU in response to a signal on ADDRESS IN, SERVICE IN, or STATUS IN. During initial selection sequence, raising COMMAND OUT in response to ADDRESS IN indicates to the selected SCU that the channel has placed a command byte on the Bus Out lines (the command byte has a fixed format). The rise of COMMAND OUT indicates that information on the Bus In lines is no longer required to be valid. COMMAND OUT stays up until the fall of the associated ADDRESS IN, STATUS IN, or SERVICE IN tag.

COMMAND OUT cannot be up concurrently with any other outbound tag line, except possibly during an interface disconnect sequence For this case, ADDRESS OUT may be up. During a channel-initiated selection sequence, COMMAND OUT indicates that BUS OUT defines the operational command to be performed. During an SCU-initiated sequence, COMMAND OUT is interpreted as follows:

- Proceed with sequence if in response to ADDRESS IN.
- Stop sequence if in response to SERVICE IN or DATA IN.
- Stack sequence if in response to STATUS IN.

Service Out

Signal SERVICE OUT is issued to the selected SCU in response to a SERVICE IN or STATUS IN signal from the SCU. SERVICE OUT indicates to the selected SCU that the channel has accepted the information on the Bus In lines or has provided the data requested by SERVICE IN on the Bus Out lines.

When SERVICE OUT is transmitted in response to SERVICE IN during a Read or Sense operation, or to STATUS IN, SERVICE OUT must rise after the channel accepts the information on the Bus In lines. During these operations, SERVICE OUT indicates that the information on the Bus In lines is no longer required to be valid and is not associated with any information on the Bus Out lines.

When SERVICE OUT is transmitted in response to SERVICE IN during a Write or Control operation, SERVICE OUT indicates that the channel has provided the requested information on the Bus Out lines. In this operation, SERVICE OUT rises after the information has been placed on the Bus Out lines. SERVICE OUT remains up until the fall of SERVICE IN. SERVICE OUT cannot be up concurrently with any other outbound tag, except during an interface disconnect sequence (when ADDRESS OUT may be up).

When SERVICE OUT is transmitted to the SCU in response to STATUS IN while SUPPRESS OUT is up, SERVICE OUT indicates to the SCU that the operation is being chained and that this status is accepted by the channel SERVICE OUT remains up until the fall of STATUS IN

Data Out

Signal DATA OUT, transmitted over a tag line from the channel to all attached SCUs, is issued in response to DATA IN from the selected SCU. DATA OUT indicates that the channel has accepted the information on BUS IN or has provided the data requested by DATA IN on Bus Out lines.

Select Out

Signal SELECT OUT is sent on an outbound tag line connected serially through each SCU and is used to select or poll an SCU. An SCU is selected by raising SELECT OUT and ADDRESS OUT; polling an SCU is done by raising SELECT OUT without ADDRESS OUT. SELECT OUT and SELECT IN form a loop from the channel through each SCU to the cable terminator block (SELECT OUT), and through each SCU back to the channel (SELECT IN). SCU selection circuitry can be attached to either SELECT OUT or SELECT IN.

The logical operation of the SCU ensures that the process of logically bypassing SELECT OUT before power changes does not interfere with the propagation of SELECT OUT. By using a special latch circuit, the SCU ensures that SELECT OUT discontinuities that may occur when the SCU is powered up do not affect propagation of SELECT OUT. If the SCU is powered up, the latch is turned on by the ANDing of SELECT OUT and HOLD OUT and is reset by HOLD OUT going low. The latch circuit is in series with the remaining selection circuitry in the SCU and provides a constant SELECT OUT within the SCU and to the other SCUs, regardless of variations in the incoming SELECT OUT signal.

The following description pertains to SCU operation with a properly operating latch circuit. The SCU raises its OPERATIONAL IN only when its incoming SELECT OUT goes high. If the SCU does not require selection, it propagates the signal to the next SCU. When the SCU propagates SELECT OUT, it does not raise OPERATIONAL IN or respond with an SCU busy sequence until the next rise of SELECT OUT. When an operation is being initiated by the channel, SELECT OUT rises a minimum of 400 nanoseconds after ADDRESS OUT goes high (the address of the SCU and disc drive being selected). The channel maintains SELECT OUT up until either SELECT IN, ADDRESS IN and OPERATIONAL IN, or STATUS IN rises

The SCU becomes selected when it raises OPERATIONAL IN. SELECT OUT drops to allow OPERATIONAL IN to drop, but after SELECT OUT is dropped the SCU maintains OPERATIONAL IN up until the associated signal sequence is completed. If the SCU raises OPERATIONAL IN, it prevents SELECT OUT from being propagated to the next SCU. If the SCU is not selected, it propagates SELECT OUT within 1.8 microseconds to the next SCU.

When STATUS IN rises in response to SELECT OUT during a selection sequence (indicating that the SCU is busy), SELECT OUT drops and does not rise until ADDRESS OUT has been reset.

Hold Out

Signal HOLD OUT is used in conjunction with SELECT OUT to provide synchronization of SCU selection. HOLD OUT is also used to minimize propagation effects when SELECT OUT goes low by purging the SELECT OUT signal from the SELECT OUT signal path. When HOLD OUT drops, it does not rise again until at least 4 microseconds. The minimum down time of HOLD OUT is 2 microseconds under any other condition.

Operational Out

Signal OPERATIONAL OUT connects the channel to all SCUs and is used for interlocking purposes. All lines from the channel are significant only when OPERATIONAL OUT is up, except for SUPPRESS OUT. When OPERATIONAL OUT is down, all inbound lines from the SCU must drop and any operation currently in process over the interface must be reset. Under these conditions, all SCU-generated interface signals go down within 1.5 microseconds after the fall of OPERATIONAL OUT at the SCU.

Suppress Out

Signal SUPPRESS OUT is transmitted over a line from the channel to all attached SCUs. It is used both alone and in conjunction with the outbound tag lines to provide the following special functions: suppress data, suppress status, command chaining, and selective reset. (Additional information about the use of SUPPRESS OUT during status and data suppression is contained in paragraphs 3.5.6 and 3.5.12.)

Clock Out

Signal CLOCK OUT is sent out from the channel to all attached SCUs and is used to provide the CPU interlock control necessary for changing the enable-disable states of SCUs. The CLOCK OUT signal must be down to permit changing enable-disable states. The minimum duration of the CLOCK OUT down state is 1 microsecond.

Metering Out

Signal METERING OUT is transmitted over a line from the channel to all attached SCUs. It is used to condition all other meters in the SCUs and disc drive modules. It is raised whenever the CPU customer meter is recording time. (The 3672 SCU does not use METERING OUT.)

3.3 INTERFACE SEQUENCES

3.3.1 Overall Descriptions

3.3.1.1 INITIAL SELECTION SEQUENCE

The channel places the address of the desired SCU, controller, and the disc drive on the Bus Out lines and raises ADDRESS OUT (Figure 3-2). Each SCU connected to the channel attempts to decode the SCU address on the bus and see if it compares with the unique address assigned each SCU. To be accepted by the addressed SCU, the address must have correct parity.

The channel then issues SELECT OUT. The SCU blocks propagation of SELECT OUT and raises OPERATIONAL IN. When OPERATIONAL IN rises, the channel responds by dropping ADDRESS OUT. After ADDRESS OUT falls and the SCU, controller, and disc drive address is on the Bus In lines, ADDRESS IN may rise. After the channel confirms the address, it responds by placing the command on the Bus Out lines and raising COMMAND OUT. The selected SCU processes the command and drops ADDRESS IN, which allows COMMAND OUT to fall. After COMMAND OUT drops, the SCU places the status information on BUS IN and raises STATUS IN.

If the channel accepts this status condition, it responds with SERVICE OUT. SERVICE OUT allows STATUS IN to fall, completing the initial selection sequence.

If the command is rejected by the SCU, the SCU presents the Unit Check status condition. No operation is initiated at the SCU and no ending status is generated.

3.3.1.2 DATA TRANSMISSION SEQUENCE

Data transfers may be initiated by the selected SCU after the selection sequence. To transmit data to the channel, the SCU places a data byte on the Bus In lines and raises DATA IN (Figure 3-2). The tag and validity of the Bus In lines must be maintained until the outbound tag is raised in response.

To request data from the channel, DATA IN is raised, and the channel places the data on the Bus Out lines and signals with DATA OUT. The channel maintains validity of the Bus Out lines until DATA IN falls. After DATA IN falls, the channel responds by dropping DATA OUT.

During Sense and Read operations, DATA IN rises when data is available on the Bus In lines. During Control and Write and Control operations, DATA IN rises when data is required on the Bus Out lines. When DATA IN is alternated with SERVICE IN, DATA IN may rise when SERVICE OUT is raised in response to SERVICE IN. However, DATA IN is not considered valid until SERVICE IN is dropped. Similarly, SERVICE IN may rise when DATA OUT is raised in response to DATA IN. However, SERVICE IN is not considered valid until DATA IN is dropped.

When DATA OUT is sent in response to DATA IN during a Sense or Read operation, the DATA OUT signal must rise after the channel accepts the information on the Bus In lines. In these cases, DATA OUT indicates that the information is no longer required to be valid on the Bus In lines and is not associated with any information on the Bus Out lines. When DATA OUT is sent in response to DATA IN or a Control or Write operation, DATA OUT indicates that the channel has provided the requested information on the Bus Out lines. DATA OUT must remain up until DATA IN goes low.

Detailed flow diagram descriptions of the read and write data transfers are presented in paragraphs 3.8 and 3.9, respectively.

3.3.1.3 ENDING SEQUENCE

The ending sequence may be initiated by either the channel or SCU. If initiated by the channel, the disc drive may still require time to reach the point where the proper status information is available, in which case a second signal sequence is necessary to complete the ending sequence. If by the SCU, the ending sequence is completed in one signal sequence, assuming that both the Channel End (CE) and Device End (DE) status conditions occur together (Figure 3-2).

Upon initiation of the ending sequence, one of three situations may exist (assume selection is already obtained):

1. The channel recognizes the end of an operation before the disc storage subsystem reaches its ending point. For this case, the SCU raises SERVICE IN whenever the SCU requires service again. The channel responds with COMMAND OUT, indicating stop. The SCU drops SERVICE IN and proceeds to its normal ending point without requesting further service. When the subsystem reaches the point at which it would normally send CE, the SCU places the ending status on the Bus In lines and raises STATUS IN. The channel responds with SERVICE OUT, unless it is necessary to stack the status condition, in which case the channel responds with COMMAND OUT.

2. The channel and subsystem recognize the end of an operation simultaneously. Status information is available at the SCU. The SCU places the ending status on the Bus In lines and raises STATUS IN.

3. The subsystem recognizes the end of an operation before the channel reaches the end of an operation. For this case also, status information is available at the SCU. The SCU places the ending status on the Bus In lines and raises STATUS IN.

If DE does not occur with CE, DE is presented when it is available and an additional status sequence is required. Additional information pertaining to the stop condition is presented in paragraph 3.5.11.

3.3.1.4 SCU BUSY RESPONSE

The SCU busy response begins when the channel places the I/O address on the Bus Out lines and raises ADDRESS OUT. SELECT OUT is then raised. Each SCU attempts to decode the address on the Bus Out lines. When SELECT OUT rises at the addressed SCU, the SCU blocks propagation of SELECT OUT, places the Busy status byte on BUS IN, and raises STATUS IN. OPERATIONAL IN is not raised.

After accepting the status byte, the channel drops SELECT OUT. The SCU responds by dropping STATUS IN and disconnecting from the interface. The channel must keep ADDRESS OUT up until STATUS IN drops, thereby completing the SCU busy response.

The SCU busy sequence is not used in response to an initial selection sequence addressed to a disc drive for which chaining has just been indicated.

3.3.1.5 SCU-INITIATED SEQUENCE

An SCU requiring service initiates a service sequence by raising REQUEST IN to the channel. The next time SELECT OUT rises at any SCU requiring service and no I/O selecting sequence is being attempted by the channel (ADDRESS OUT down), the SCU places the address of the SCU and disc drive on the Bus In lines and raises both ADDRESS IN and OPERATIONAL IN. When the channel recognizes the address, COMMAND OUT is sent to the SCU, indicating Proceed. After ADDRESS IN drops, the channel responds by dropping COMMAND OUT.

3.3.1.6 IMMEDIATE COMMAND SEQUENCE

An immediate command is one whose execution meets the following requirements:

- Execution requires no more information than that in the command byte; that is, no data or information bytes are transferred.

- Channel end time coincides with initial status time and, on a normal operation, at least Channel End instead of Zero status will be in the initial status byte.

A channel response of COMMAND OUT to STATUS IN cannot prevent the execution of an immediate command.

3.3.2 Initial Selection Sequence

The following two paragraphs describe the initial selection of an SCU, controller, and disc drive in terms of the operations performed jointly by the channel and SCU to effect the selection. Paragraph 3.3.2.1 is a simplified description of the sequence, discussing operation of the channel and SCU only. Paragraph 3.3.2.2 discusses the initial selection in greater detail by including channel interface and microprogram operations.

3.3.2.1 SIMPLIFIED SEQUENCE

The channel sends SCU and the address of the requested disc drive to all attached SCUs. In order of priority, each SCU compares its address to the address sent by the channel. If the addresses compare (are equal), the addressed SCU sends its address and the address of the requested disc drive to the channel.

The channel then checks the address received with the address sent. If the addresses do not compare, the channel turns on Interface Control Check. If the addresses compare, the channel sends a command to the addressed SCU.

The SCU accepts the command and sends its status to the channel. If the status is to be accepted, the channel responds with SERVICE OUT which tells the SCU it is

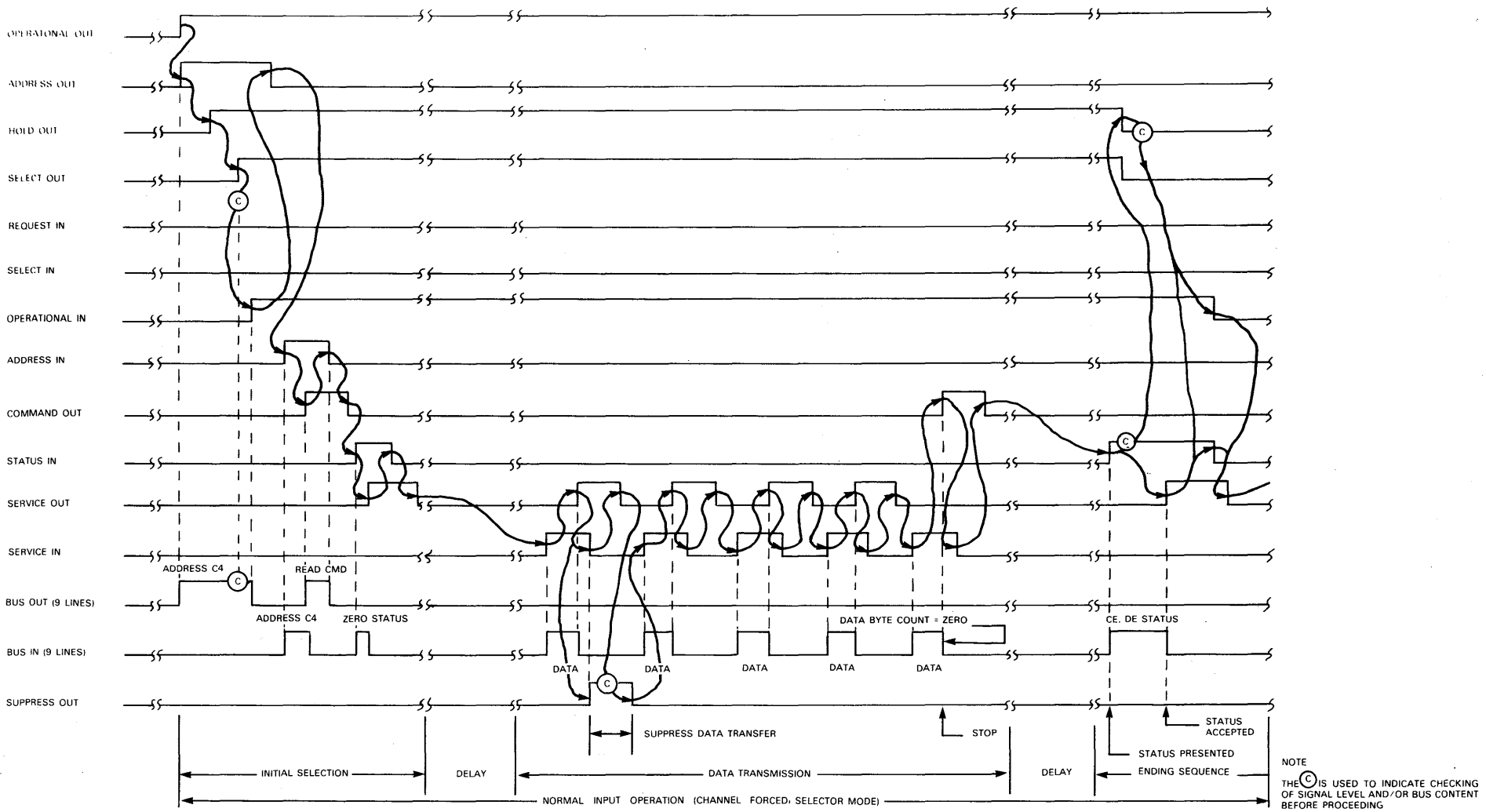


Figure 3-2. Selector Channel Operation Data Transfer Sequences

ready to begin the data transfer. If status is non-zero, the channel drops SELECT OUT and HOLD OUT and the sequence is terminated.

The above sequence is implemented by channel- and SCU-initiated events as listed below, where **(C)** indicates channel.

(C) Raises OPERATIONAL OUT, ADDRESS OUT, HOLD OUT, and SELECT OUT. Places address of requested disc drive on BUS OUT.

(SCU) Compares address on BUS OUT with address of SCU.

If address does not compare: sends SELECT OUT/SELECT IN to next lower priority SCU.

If address compares, but SCU is busy: raises STATUS IN.

If address compares, and SCU is not busy: raises OPERATIONAL IN.

(C) Recognizes OPERATIONAL IN and drops ADDRESS OUT.

(SCU) Recognizes fall of ADDRESS OUT. Places SCU, controller, and disc drive address on BUS IN. Raises ADDRESS IN.

(C) Compares address on BUS IN with requested address sent to SCU.

If address does not compare: sets Interface Control Check.

If address compares: places command on BUS OUT and raises COMMAND OUT.

(SCU) Places disc drive status on BUS IN. Raises STATUS IN.

(C) Checks for zero status.

If non-zero: channel drops SELECT OUT and HOLD OUT and the sequence is terminated.

If zero: responds with SERVICE OUT to indicate to selected SCU that channel has accepted the status.

3.3.2.2 DETAILED SEQUENCE

The following list of events provides a detailed description of the initial selection sequence. Abbreviations are as follows: channel **(C)**, channel interface **(CI)**, and microprogram **(M)**. Registers listed are described in detail in Section 6.

(C) Raise OPERATIONAL OUT, ADDRESS OUT, HOLD OUT, and SELECT OUT. Place requested SCU address on BUS OUT.

(CI) Compare SCU address to address on BUS OUT. If equal, set SCU Selected Latch. Set Switched to A, B, C, or D Latch. If SCU is busy, SCU responds to channel with STATUS IN. If SCU is not busy,* set Interrupt (INT) Register bit 5 (Channel Interrupt). Set Channel Flags (CHF) Register bit 3 (Address Out). Bus Out (BO) Register contains address.

* When SCU goes not-busy, SCU initiates a Polling Interrupt. This places SCU End in the Status Byte.

(M) Microprogram recognizes INT 5 branch condition, sets bit 3 (Operational In) and bit 4 (Address In) in Channel Tag (CT) Register, and places SCU, controller, and disc drive address in Bus In (BI) Register. Microprogram records address from BO Register for later use.

(CI) Set bit 3 (Operational In) and bit 4 (Address In) of CT Register. BI Register contains SCU, controller, and disc drive address. Raise OPERATIONAL IN.

(C) Channel recognizes OPERATIONAL IN and drops ADDRESS OUT.

(CI) Reset CHF Register bit 3 (Address Out). Gate SCU, controller, and disc drive address from BI Register to BUS IN and raise ADDRESS IN.

(C) Channel compares address on BUS IN with address sent to SCU to ensure that they are the same. Place Command (Read, Write, etc.,) on BUS OUT and raise COMMAND OUT.

(CI) Generate Set BO which causes the command on BUS OUT to be loaded into BO Register. COMMAND OUT from channel sets bit 4 (Command Out) in CHF Register. BO Register contains command.

(M) Microprogram recognizes setting of Command Out bit in CHF Register and resets CT Register bit 4 (Address In).

(CI) Reset CT Register bit 4 (Address In). Drop ADDRESS IN to channel.

(C) Channel recognizes the fall of ADDRESS IN and drops COMMAND OUT.

(CI) Reset CHF Register bit 4 (Command Out).

(M) Microprogram recognizes resetting of Command Out bit in CHF Register, determines SCU status, places status in BI Register and sets CT Register bit 5 (Status In).

(CI) Set CT Register bit 5 (Status In). BI Register contains SCU Status.

(CI) Gate contents of BI Register to BUS IN. Raise STATUS IN to channel with SCU status on BUS IN.

(C) Channel checks for zero status. If status is non-zero, channel drops SELECT OUT and HOLD OUT. If status is zero, channel raises SERVICE OUT.

(CI) Set CHF Register bit 7 (Service Out).

(M) Microprogram recognizes setting of Service Out bit and drops CT Register bit 5 (Status In).

(CI) Drop CT Register bit 5 (Status In).

(C) Channel drops SERVICE OUT.

(CI) Reset CHF Register bit 7 (Service Out).

(M) Microprogram recognizes resetting of Service Out bit. Microprogram decodes and executes command and if command requires it, prepares for data transfer.

3.3.3 Ending Sequence

The following two paragraphs describe the ending sequence of an SCU, controller, and disc drive in terms of the operations performed jointly by the channel and SCU to effect the ending sequence. Paragraph 3.3.3.1 is a simplified description of the sequence, discussing operation of the channel and SCU only. Paragraph 3.3.3.2 discusses the ending sequence in greater detail by including channel interface and microprogram operations.

3.3.3.1 SIMPLIFIED SEQUENCE

(C) Channel word count equals zero. Channel responds to DATA IN or SERVICE IN with COMMAND OUT.

—OR—*

(SCU) SCU recognizes COMMAND OUT to stop data transfer. When word count from track field equals 0, SCU places status on BUS IN and raises STATUS IN.

(C) Channel drops SELECT OUT, HOLD OUT, and raises SERVICE OUT.

(SCU) SCU recognizes SERVICE OUT and drops STATUS IN and OPERATIONAL IN.

(C) Channel drops SERVICE OUT (chaining is indicated by SUPPRESS OUT being up).

(SCU) Recognizes fall of SERVICE OUT and drops OPERATIONAL IN. If SCU has additional status to present, it is presented via polling sequence.

* This sequence begins with either the (C) or (SCU) operation.

3.3.3.2 DETAILED SEQUENCE

The following list of events provides a detailed description of the ending sequence. Abbreviations are as follows: channel **(C)**, channel interface **(CI)**, and microprogram **(M)**. Registers listed are described in detail in Section 6.

(C) Channel word count equals zero. Channel responds to DATA IN or SERVICE IN with COMMAND OUT.

(CI) Truncation will set B(INT) branch condition in microprocessor.

(M) Microprogram recognizes setting of bit 5 of B(INT) branch condition and stops read or write data transfer.

(CI) Reset Channel Control (CHC) Register bit 0 (Channel Transfer).

(CI) Drop DATA IN or SERVICE IN.

(C) Drop COMMAND OUT.

(M) When word count from count field of the record equals zero, microprogram sets SCU status into Bus In (BI) Register and sets Channel Tag (CT) Register bit 5 (Status In).

(CI) Set CT Register bit 5 (Status In).

(CI) Gate BI Register onto BUS IN, and raise STATUS IN to channel with status on BUS IN.

(C) Channel stores status in channel. Channel drops SELECT OUT, HOLD OUT, and raises SERVICE OUT

(CI) Set CHF Register bit 7 (Service Out).

(M) Microprogram recognizes setting of Service Out bit and resets CT Register bit 5 (Status In)

(CI) Reset CT Register bit 5 (Status In).

(CI) Drop STATUS IN to channel.

(C) Channel drops SERVICE OUT (chaining is indicated by SUPPRESS OUT).

(CI) Reset CHF Register bit 7 (Service Out).

(M) Microprogram resets CT Register bit 3 (Operational In).

(CI) Reset CT Register bit 3 (Operational In).

(CI) Drop OPERATIONAL IN to channel.

(C) OPERATIONAL IN to channel is dropped.

(M) If SCU has additional status to present, it is presented via polling sequence.

3.3.4 Polling Sequence and Status Presentation

The following two paragraphs describe the SCU-initiated polling sequence and status presentation of a subsystem in terms of the operations performed jointly by the channel and SCU to effect the sequence. Paragraph 3.3.4.1 is a simplified description of the sequence, discussing operation of the channel and SCU only. Paragraph 3.3.4.2 discusses the polling sequence in greater detail by including the channel interface and microprogram operations.

3.3.4.1 SIMPLIFIED SEQUENCE

(SCU) SCU has status to present to channel and raises REQUEST IN.

(C) Channel raises SELECT OUT and HOLD OUT.

(SCU) SCU prevents propagation of SELECT OUT; drops REQUEST IN; raises OPERATIONAL IN and ADDRESS IN; and places SCU, controller, and disc drive address on BUS IN.

(C) Channel saves SCU address and raises COMMAND OUT.

(SCU) SCU drops ADDRESS IN.

(C) Channel drops COMMAND OUT

(SCU) SCU places SCU, controller, or disc drive status on BUS IN and raises STATUS IN.

(C) Channel stores status, raises SERVICE OUT, and drops SELECT OUT and HOLD OUT.

(SCU) SCU drops STATUS IN and OPERATIONAL IN.

(C) Channel drops SERVICE OUT.

3.3.4.2 DETAILED SEQUENCE

The following list of events provides a detailed description of the polling sequence and status presentation. Abbreviations are as follows: channel **(C)**, channel interface **(CI)**, and microprogram **(M)**. Registers listed are described in Section 6.

(M) SCU, controller, or disc drive has status to present to channel. Microprogram sets Channel Tag (CT) Register bit 0 (Request In).

(CI) Set CT Register bit 0 (Request In). Raise REQUEST IN to channel.

(C) Channel recognizes REQUEST IN and raises SELECT OUT and HOLD OUT.

(CI) Prevent propagation of SELECT OUT. Set SELECT OUT latch and B(INT) branch condition.

(M) Microprogram recognizes B(INT) branch condition, resets CT Register bit 0 (Request In), and sets CT Register bit 4 (Address In) and bit 3 (Operational In). Place appropriate device address in Bus In (BI) Register

(CI) Set CT Register bit 4 (Address In) and bit 3 (Operational In). Reset CT Register bit 0 (Request In).

(CI) Drop REQUEST IN and raise OPERATIONAL IN and ADDRESS IN. Place SCU, controller, and disc drive address from BI Register on BUS IN

(C) Channel saves SCU, controller, or disc drive address and raises COMMAND OUT.

(CI) Set Channel Flags (CHF) Register bit 4 (Command Out).

(M) Microprogram recognizes setting of Command Out bit and resets CT Register bit 4 (Address In).

(CI) Reset CT Register bit 4 (Address In). Drop ADDRESS IN.

(C) Channel drops COMMAND OUT.

(CI) Reset CHF Register bit 4 (Command Out).

(M) Microprogram recognizes resetting of Command Out bit, places SCU, controller, or disc drive status in BI register, and sets CT Register bit 5 (Status In).

(CI) Set CT Register bit 5 (Status In).

(CI) Gate BI Register to BUS IN, and raise STATUS IN with SCU, controller, or disc drive status on BUS IN.

(C) Channel stores SCU, controller, or disc drive status and responds with SERVICE OUT. Channel drops SELECT OUT and HOLD OUT.

(CI) Set CHF Register bit 7 (Service Out).

(M) Microprogram recognizes setting of Service Ott bit and resets CT Register bit 5 (Status In).

(CI) Reset CT Register bit 5 (Status In)

(CI) Drop STATUS IN.

(C) Channel drops SERVICE OUT.

(M) Microprogram drops OPERATIONAL IN.

(CI) Reset CT Register bit 3 (Operational In). Drop OPERATIONAL IN.

3.4 CHANNEL OPERATION

The SCU is capable of communicating in two different channel modes: selector mode and block multiplexer mode. Operating characteristics of the SCU when communicating in either of these modes are discussed below.

3.4.1 Selector Mode Operation

The normal mode of SCU operation is in a channel-forced burst mode. The channel is a block multiplexer channel operating in a selector mode. The mode is implemented by setting a channel mode bit in a CPU control register. This bit is set to enable the selector mode at IPL or upon system reset, and can be altered by programming at any time. The burst mode indicates that the complete block of data called for by the CCW from one device is transferred as a complete entity, with no interleaving of data called for by other CCWs associated with other devices. The burst mode is implemented by holding up OPERATIONAL IN for the duration of the data transfer.

Individual sequences of the selector mode manner of operation have been described in paragraphs 3.3.1 and 3.3.2. They are shown in terms of the interface signals raised and lowered in Figures 3-2 and 3-3.

3.4.2 Block Multiplexer Operation

3.4.2.1 INTRODUCTION

The block multiplexer channel enables data to be transferred between the CPU and storage devices at higher effective rates of speed than is possible with the selector channel. The higher effective data rates allow attachment of high-speed I/O devices such as the 3672-based subsystem.

A block multiplexer channel performs differently from a selector channel in the way that command-chained channel programs are handled. While executing such programs, a selector channel or a block multiplexer channel operating in a selector mode, is busy during the entire time the channel program is in operation, whether data transfer is occurring or not. A block multiplexer channel executing a command-chained channel program has the ability to disconnect from the operational channel program during certain non-data transfer operations. Such operations as disc drive seeking are channel nonproductive activities. A block multiplexer channel can be freed during a nonproductive activity to allow more data to be transferred per unit of channel busy time.

A single block multiplexer channel can support interleaved concurrent execution of multiple channel programs. It has multiple subchannels, each of which has an associated unit control word (UCW). UCWs are stored in normally unaddressable auxiliary storage of 16K bytes, referred to as "bump" storage and can support one I/O operation.

To facilitate channel scheduling, an interrupt condition called Channel Available has been defined for internal use on block multiplexer channels. At disconnect time for a channel program, the channel is available for the resumption of an uncompleted channel program previously started, or another channel program can be initiated. A Channel Available interrupt occurs at disconnect time to indicate channel availability if a *Start I/O, Test I/O, Test Channel,* or *Halt I/O* instruction was issued previously while the block multiplexer channel was busy.

The block multiplexer mode is enhanced through the use of two standard features of the 3672-based subsystem: rotational position sensing (RPS) and disconnected command chaining. These two features, together with block multiplexing, increase system throughput by increasing channel throughput.

The use of RPS frees the channel more often during drive operations. For example, during the time required to position a track to a specific record, channel programs are permitted to be initiated sooner on block multiplexer channels than is possible with selector channels.

Disconnected command chaining is implemented in the SCU to enable it to handle concurrent execution of multiple channel programs. For example, the SCU can simultaneously control many channel programs, one for each of its drives.

3.4.2.2 ROTATIONAL POSITION SENSING

Use of RPS reduces the time the channel is busy searching for a disc record. It permits a Search command to be initiated just before the desired record is to come under the Read/Write head; that is, when the desired rotational position is reached. To accomplish this, the tracks in each cylinder of a drive are considered to consist of equally spaced sectors. A sector is defined as the length of track arc that passes under the Read/Write heads in approximately 130 microseconds. Track formatting is unchanged but each record has a sector number as well as a record address. There are 128 sectors per track on the 367X drive, with each sector allocated a unique sector number (00-7F hexadecimal). See Figure 3-4.

The SCU can determine the sector currently under the Read/Write heads of each of its drives. A sector counter is contained in each drive. The counter is incremented once every sector time period (approximately 130 microseconds) and set to zero each time the index marker passes under the heads. The sector in which a record falls is a function of the length of all records that precede it and of its sequential position on the track. Therefore, the sector location can be calculated for fixed-length records and a sector number allocated.

Two disc commands are provided for use with rotational position sensing: *Set Sector* and *Read Sector*. *Set Sector* used in conjunction with the block multiplexer channel permits a single command-chained channel program to be initiated for each disc operation that frees the channel and SCU during rotational positioning operations. If the sector address of a record is known or can be calculated, a *Set Sector* command can be included in the disc channel program to cause the SCU to look for the designated sector. Once the SCU accepts the sector number, both the SCU and the channel disconnect are available for another I/O operation.

The *Read Sector* command is useful for sequential disc processing and for write verification. When chained from a Read, Write, or Search command, *Read Sector* provides the sector number required to access the record processed by the previous CCW. This sector number can be used to reposition the track to verify the record just written or to read or write the next sequential record.

3.4.2.3 DISCONNECTED COMMAND CHAINING

Disconnected Command Chaining (also referred to as multiple requesting) is used for the following purposes:

- Allows the SCU to disconnect from the channel on commands which require long delays due to mechanical motion or searches. The channel and SCU are free during the delay period.
- Enables multiple requesting, which allows separate command chains to be active in the subsystem, one for each disc drive.

A block diagram of Disconnected Command Chaining is shown in Figure 3-5.

Disconnected Command Chaining enables the SCU to disconnect from the channel after an operation such as Seek or Set Sector has started even though chaining is indicated. Since the burst mode is not forced during the execution of *Seek* or *Set Sector* commands, the SCU can disconnect between Channel End and Device End. The disconnect function reduces the CPU interrupts needed to overlap channel data transfers with mechanical motion of the drives. The SCU retains the information necessary to control a disconnected CCW chain for each drive in the subsystem. In effect, the SCU is capable of executing many CCW chains (one per drive) simultaneously.

During a Seek operation, the SCU attempts to reconnect after drive mechanical motion is completed. During a Set Sector operation, the SCU attempts to reconnect when the desired sector number is detected. In either case, the channel is made available during access motion and rotational delay periods. Disconnect is also allowed on Command Retry procedures.

Disconnected Command Chaining can involve two channels, or one channel can have several CCW chains going.

Two examples of Disconnected Command Chaining are shown in Figure 3-5.

3.4.2.4 MULTIPLEXER CHANNEL OPERATION

This paragraph contains an explanation of block multiplexing operations. References are made to Figure 3-6, which illustrates the detailed signal sequences for an SCU-forced, burst mode, multiplexer channel operation. This is the normal output operation.

Assume a block multiplexer channel is executing a channel program consisting of multiple command-chained CCWs. When Channel End is presented without a corresponding Device End, the channel disconnects from the current channel program and becomes available for an I/O operation via another device. The disconnection occurs even though the current channel program is not complete. At disconnect time, the information necessary to restart the disconnected channel program is stored in the UCW assigned to the active subchannel and SCU.

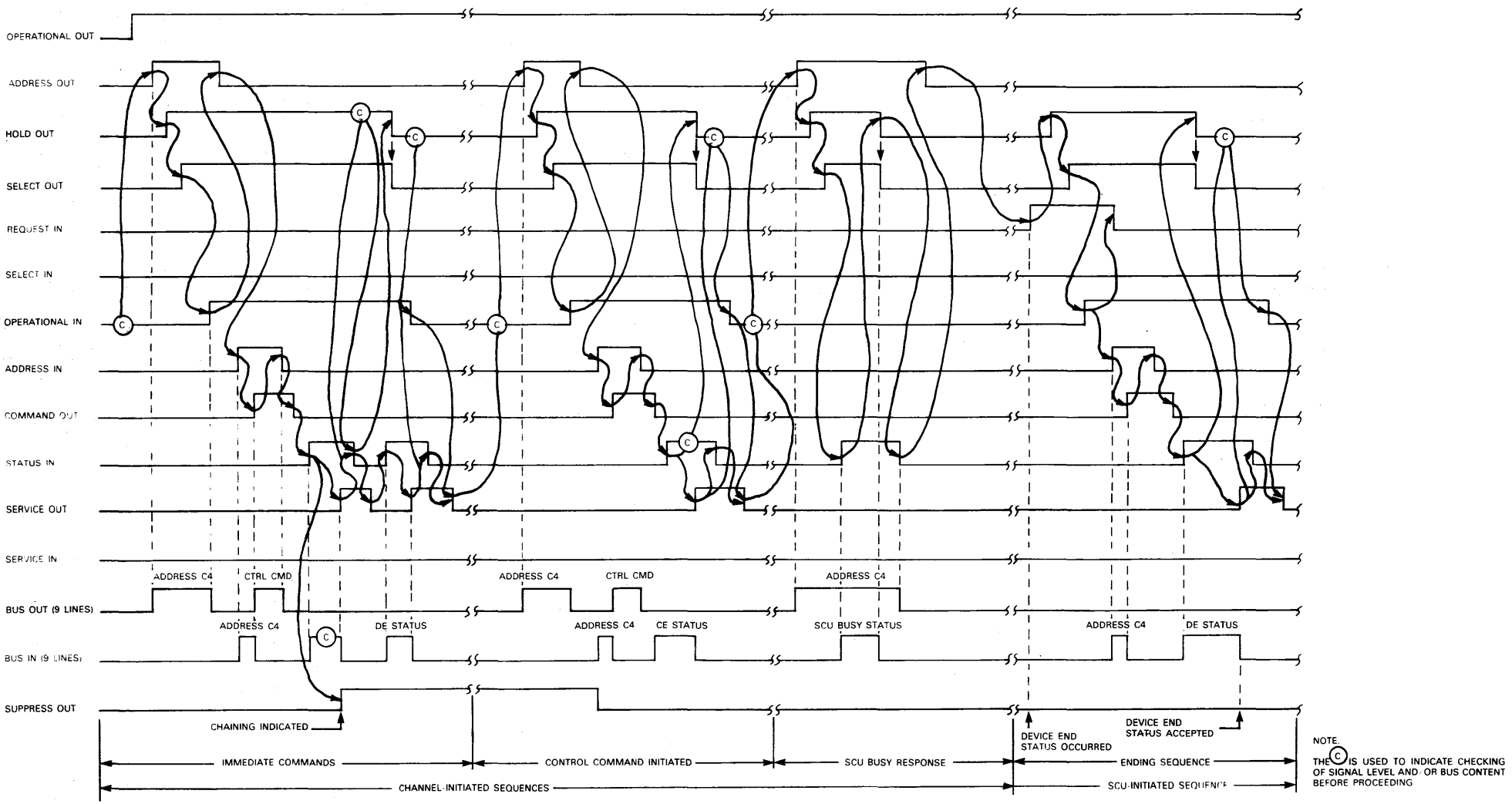


Figure 3-3 Selector Mode Operation, Miscellaneous Sequences

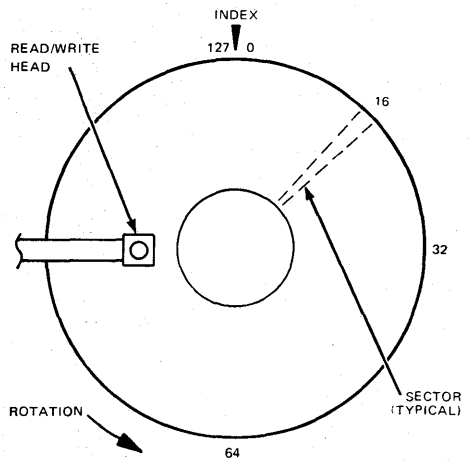


Figure 3-4. Disc Track Sector Format

When Device End is presented to signal that the drive is again ready for the channel, the SCU attempts to regain use of the channel by signalling REQUEST IN. If the channel is free, the active channel registers are reloaded with the information previously saved in the UCW subchannel assigned to the drive, and the disconnected channel program is resumed at the appropriate CCW.

If the channel is busy when reconnection is requested, the SCU must wait until the channel becomes available. Once multiple channel programs have been initiated on one channel, the interleaving of data transfer operations is controlled by block-multiplexer-channel hardware and the SCUs of disc drives operating in the block multiplexing mode.

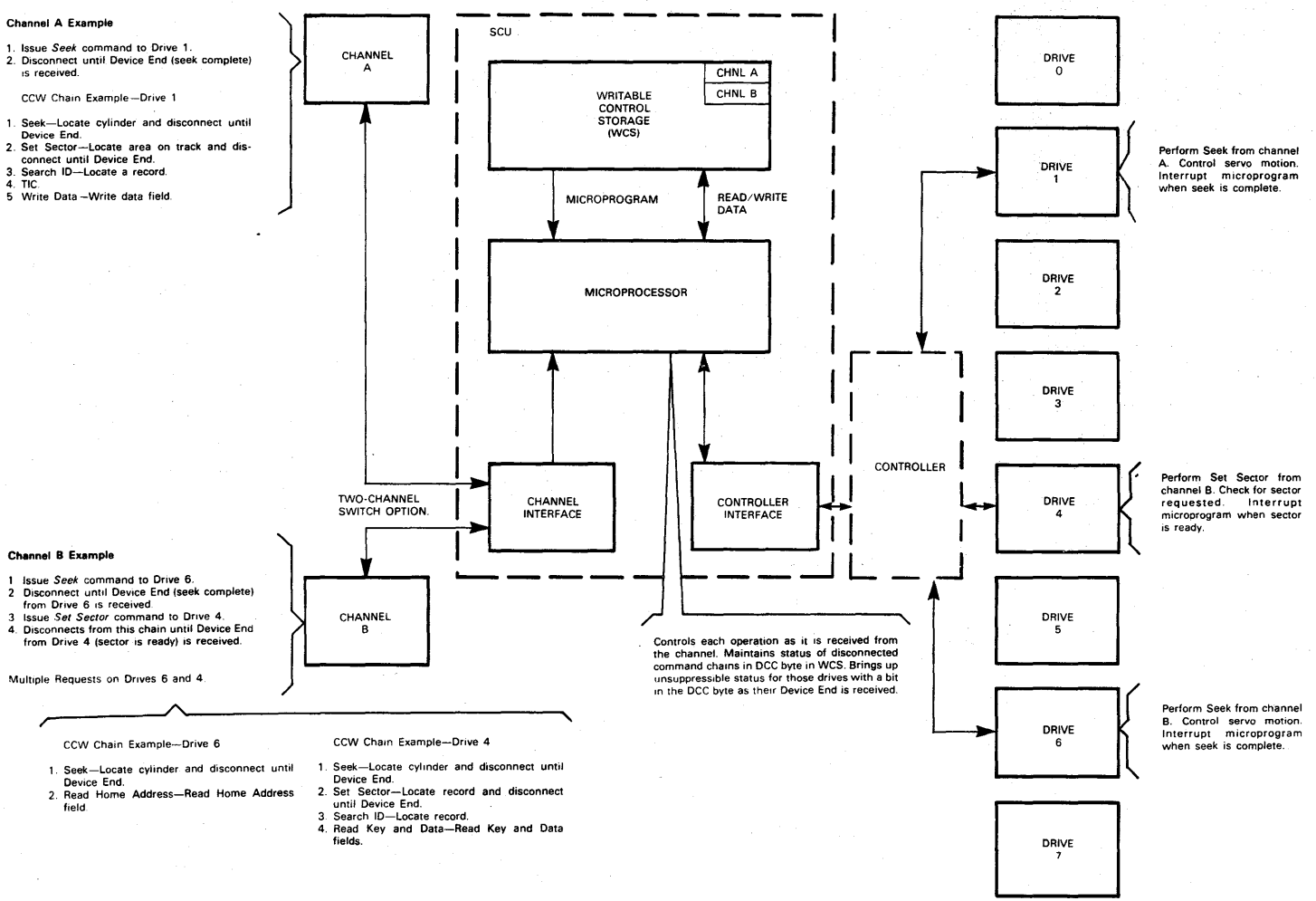


Figure 3-5. Disconnect Command Chaining Block Diagram and Examples

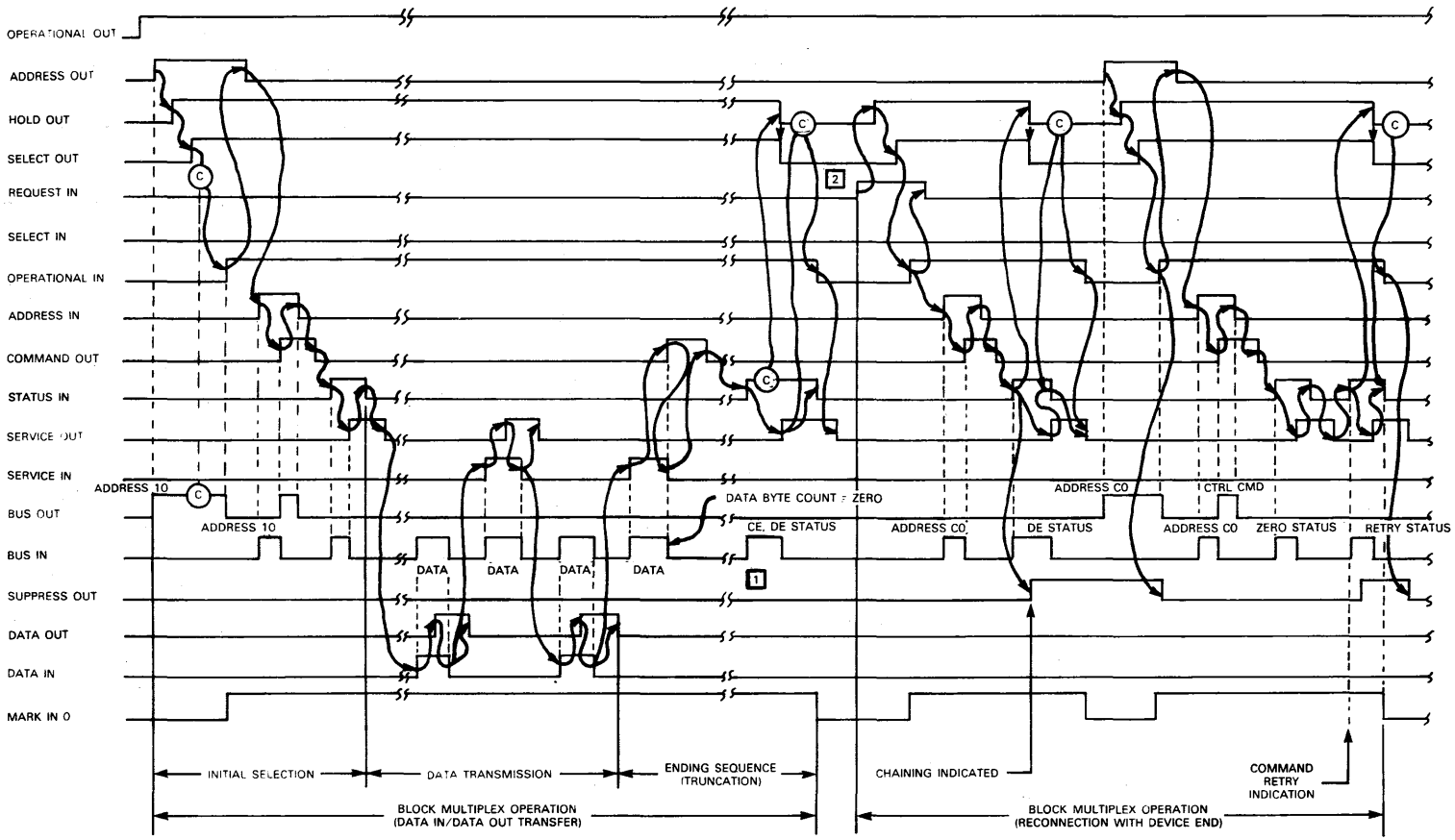


Figure 3-6. Block Multiplexer Channel Operation (1 of 2)

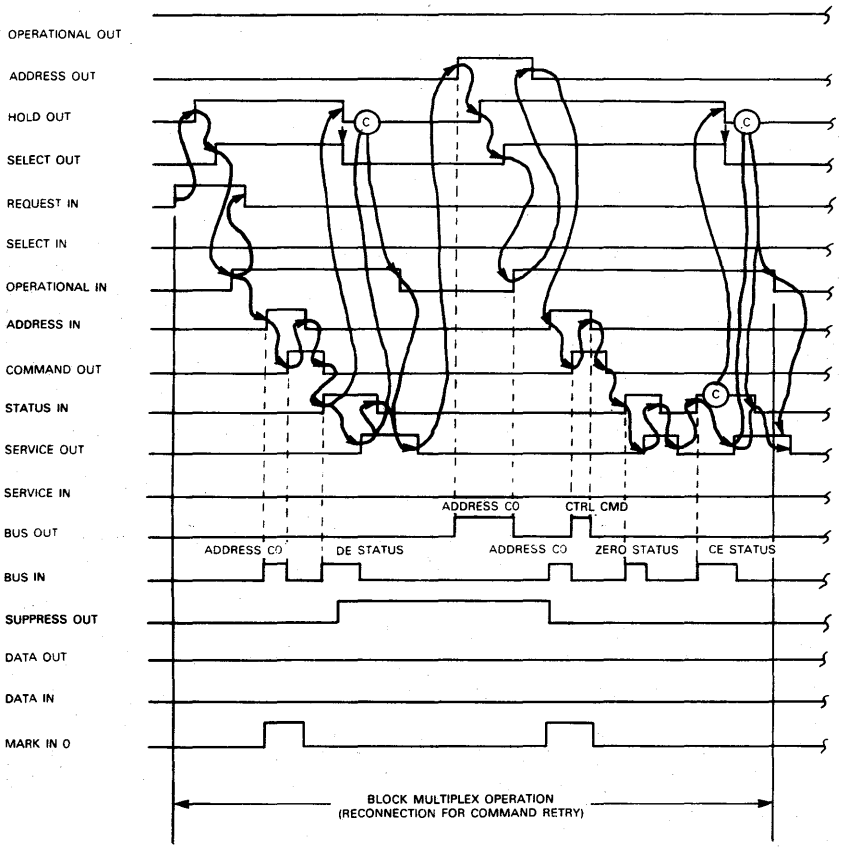


Figure 3-6. Block Multiplexer Channel Operation (2 of 2)

When multiple channel programs are operating concurrently in the block multiplexing mode, a device can regain control of the channel only when the channel is not busy. Therefore, only cyclic devices, such as the 367X Drive with RPS, can disconnect during execution of a command-chained channel program on a block multiplexer channel and resume operation later.

Data transfer for concurrently operating devices on a block multiplexer channel are interleaved on a first-come, first-served basis as the desired records become available. Therefore, devices are serviced in the order in which their records become available, not necessarily in the order in which their channel programs are initiated.

3.5 SEQUENCE CONTROLS

3.5.1 Command Chaining

The 3672-based subsystem permits command chaining, which is the ability to execute a series of channel commands as a result of a single I/O instruction. Command chaining is specified when bit 33 (CC) in the CCW is set to 1. Upon completing the current CCW with CC on, the channel fetches a new CCW that specifies the next I/O operation. This operation is automatically executed when the subsystem completes the current operation and Device End status is accepted by the channel. Upon completing the current CCW, an I/O interrupt does not occur, and the count, indicating the amount of transferred data, is not available to the program. All subsystem programs normally use command chaining. Time is available in the gap between record areas to execute command chaining functions. See Section 2 for restrictions on the command sequence within a chain.

Command chaining is indicated if SUPPRESS OUT is up when SERVICE OUT is raised in response to STATUS IN. To ensure recognition of command chaining by the SCU, SUPPRESS OUT must be up at least 250 nanoseconds before SERVICE OUT rises in response to STATUS IN and must not fall before STATUS IN.

Command chaining is suppressed if an operation is terminated with either Unit Check or Unit Exception status conditions noted.

3.5.2 Data Chaining

Data transferred between CPU main storage and the subsystem may be chained. This data chaining permits blocks of data to be transferred to or from noncontiguous areas of main storage. Data chaining is specified by turning on bit 32 (CD) of the CCW. The new CCW command code is ignored unless the command code (CCW bits 0-7) specifies transfer-in-channel (TIC).

Data chaining occurs immediately after the last byte of data designated by the current CCW has been transferred to main storage or accepted by the subsystem.

Data chaining takes precedence and command chaining is ignored if both data chaining and command chaining are indicated in the CCW

NOTE

Data chaining capabilities are dependent on several variable factors, such as system type, I/O configuration, and channel loading. Because of these dependencies, Read or Write data chaining within record areas may cause unpredictable overruns or chaining checks. Refer to the FIPs volume if these conditions are encountered or suspected.

3.5.3 Branching in Channel Programs

Branching in channel programs allows the next CCW to be read from a storage address other than the address normally accessed, which is 8 positions higher than the current CCW location. This branching is accomplished either of two ways:

1. If command chaining is specified in a Search command, and execution of the command results in a Status Modifier indication (search satisfied), the channel fetches the next CCW from a main storage location, sixteen positions higher than the current CCW.

2. *TIC* command may be used to modify the sequence of a chain of commands. The data address portion of the TIC CCW specifies the main storage location of the next CCW. Therefore, the next CCW may be fetched from any valid main storage location.

3.5.4 Unit Selection and Device Addressing

The I/O addresses of the 3672 SCU, 3673 Controller, and 367X Drives are designated by an eight-bit binary number in an I/O instruction. See paragraph 2.3.2. These addresses consist of three parts:

1. SCU address in the four most significant bits (three if a 32-spindle configuration).

2. Controller address in the next least significant bit(s).

3. Drive address in the three least significant bits.

The SCU address is determined by the customer when the subsystem is initially installed. A drive address is determined by logical assignments specified by the drive-inserted Logical Address Plug. The SCU accepts any controller/drive address from 0000 to 1111 (00000 to 11111 for 32-spindle configuration). An operation is terminated with Unit Check status if multiple responses to an address occur due to hardware failures or duplicate (modulo-8) Logical Address Plugs on the same controller.

3.5.5 Stack Status

Stack Status is indicated by a COMMAND OUT response from the channel to STATUS IN (paragraph 3.3.1.3). The Stack Status indication causes status information to be retained at the SCU, controller, or disc drive until that status is accepted on a subsequent status cycle with SERVICE OUT. When Stack Status occurs, the SCU is disconnected from the channel interface after SELECT OUT goes down, and COMMAND OUT remains up until OPERATIONAL IN falls. An attempt by the SCU to initiate a selection sequence to present the stacked status is under control of SUPPRESS OUT.

3.5.6 Suppress Status

Whenever the channel is unable to process an immediate request resulting in status conditions, SUPPRESS OUT may be raised. With this line up, the SCU may not attempt to initiate a selection sequence to present suppressible-type status information. Stacked Status is suppressible. Status that contains Channel End is not suppressible until stacked, except when an Interface Disconnect is received for that address. Status that contains Device End is not suppressible when chaining has been indicated, until it is stacked. Other asynchronous status conditions may be suppressible at the option of the SCU without being stacked.

SUPPRESS OUT must be up at least 250 nanoseconds before SELECT OUT rises at the SCU if suppression of status is to be ensured. If SUPPRESS OUT rises after a status sequence has been started, the status sequence proceeds normally.

3.5.7 Disconnect In

The DISCONNECT IN signal enables the SCU to alert the system of a malfunction that is preventing the SCU from signalling properly over the I/O interface. In response to DISCONNECT IN, the channel performs a Selective Reset.

3.5.8 Interface Disconnect

The SCU recognizes Interface Disconnect when ADDRESS OUT is up, and SELECT OUT and HOLD OUT are down at least 250 nanoseconds before the completion of any signal sequence. In this case, ADDRESS OUT may be up concurrently with another outbound tag. When OPERATIONAL IN drops, the channel may drop ADDRESS OUT to complete the interface disconnect sequence. ADDRESS OUT must be down at least 250 nanoseconds before a new channel-initiated selection sequence may be initiated.

Except for REQUEST IN, the SCU responds to the Interface Disconnect by removing all signals from the interface. The SCU remains busy after it receives an Interface Disconnect, while performing an operation, until the Device End status is received by the channel.

3.5.9 Selective Reset

The drive presently operating over the channel interface may be put into a reset condition, termed Selective Reset. This condition is indicated whenever SUPPRESS OUT is up and OPERATIONAL OUT drops. As a result, OPERATIONAL IN goes low, and the disc drive operating over the interface and its status are reset. The operation in process proceeds to a normal stopping point, if applicable, with no further data transfer. Only the disc drive operating over the interface is reset. The disc drive path is in the busy state throughout the Selective Reset period.

The ready or not-ready state of the SCU is generally not changed by a Selective Reset. However, when the SCU's Enable/Disable or Online/Offline switch was changed before the reset but had not become effective due to the required inhibiting conditions, the ready or not-ready state may change if the reset clears those inhibiting conditions.

Selective Reset is issued only as a result of a timeout by the channel or a malfunction detected at the channel.

3.5.10 Command Retry

When the command being executed encounters a condition requiring retry, the SCU indicates this requirement by raising STATUS IN while presenting Unit Check, Channel End, and Status Modifier (MARK IN 0 being up for retry status)

The channel acknowledges the occurrence of command retry by accepting the status byte containing retry status and indicating chaining. When Device End is presented to the channel, it is accepted with chaining indicated and a normal reselection occurs to reissue the previous command. A channel indicates refusal to perform a Command Retry by accepting the status byte containing retry status without indicating chaining or by stacking the status byte.

Additional information about Command Retry is contained in Paragraph 2.11.

3.5.11 Proceed

At any time other than during a channel-initiated selection sequence, proceed is indicated whenever COMMAND OUT responds to ADDRESS IN. It causes the SCU to continue the normal servicing sequences on the interface.

3.5.12 Stop (or Truncation)

Stop is indicated by COMMAND OUT in response to SERVICE IN, DATA IN, or as a result of an Interface Disconnect which occurs before the normal Channel End for the operation in process. It is used to signal the SCU that the channel is ending the current operation. On receipt of the stop indication, the SCU must proceed to its normal ending point without sending any further SERVICE IN signals to the channel. The SCU remains busy until the ending status is presented and accepted by the channel.

For I/O operations that have relatively long times between stop and ending status and no time-dependent chaining requirements, OPERATIONAL IN is dropped on receipt of the stop indication.

3.5.13 Suppress Data

Input/Output operations whose rate of data transfer can be adjusted without overrunning are subject to suppression of data transfer by SUPPRESS OUT. A suppression of data can occur as follows:

1 Unless the data transfer is contiguous with initial selection, SUPPRESS OUT is ignored for the first data byte of any selection sequence. This means no deselection and reselection occurs between initial selection and data transfer.

2 To ensure suppression of subsequent data, SUPPRESS OUT must be up either 250 nanoseconds before the rise of SERVICE OUT or DATA OUT, or at least 250 nanoseconds before SERVICE OUT or DATA OUT falls.

3. When SUPPRESS OUT is up at the SCU, the SCU must not raise SERVICE IN or DATA IN for subsequent suppressible data.

3.5.14 Data Acceptance

During a Read or Sense operation, information placed on the Bus In lines which has been accepted by the channel is indicated by raising SERVICE OUT in response to SERVICE IN, or DATA OUT in response to DATA IN.

3.5.15 Data Ready

During a Write or Control operation, the requested information which has been placed on the Bus Out lines and is ready for acceptance by the SCU is indicated by raising SERVICE OUT in response to SERVICE IN, or DATA OUT in response to DATA IN.

3.5.16 Status Acceptance

Status placed on the Bus In lines which has been accepted by the channel is indicated by raising SERVICE OUT in response to STATUS IN.

3.5.17 System Reset

A System Reset condition is indicated whenever OPERATIONAL OUT and SUPPRESS OUT are down concurrently and the disc drive is in the on-line mode. This condition causes OPERATIONAL IN to fall, and all SCUs and their attached controllers and disc drives, along with their status, to be reset. The SCUs remain in a busy state for the duration of their reset procedure. The ready or not-ready state of an SCU is not changed by a System Reset.

To ensure a proper reset, OPERATIONAL OUT and SUPPRESS OUT must be down concurrently for at least 6 microseconds.

3.6 STATUS CONDITIONS

3.6.1 Overview

When requested by the channel, status information about the subsystem is transferred to the channel by means of the *Test I/O* command. This command is generated by the channel as a result of the *Test I/O* instruction, or as an internal channel function when the channel requests status.

The *Test I/O* command transfers one status byte to the channel. This byte is placed on the Bus In lines and is defined when STATUS IN is high. The status byte reports status conditions about the SCU, controller, and disc drive whose addresses appeared on the Bus In lines (with ADDRESS IN up) during the polling or selection portion of the sequence. In the case of the SCU busy sequence, when no ADDRESS IN occurs, it is assumed that the conditions pertain to the SCU.

NOTE

Unless otherwise stated, information on this page pertains to SCUs attached to only one channel interface.

Transfer of the status byte occurs in seven situations:

- during the initial selection sequence;
- to present Channel End at the end of data transfer;
- to present Device End and any associated conditions to the channel;
- to present Control Unit End or Device End to signal that the SCU, controller, or disc drive previously busy, and then interrogated, is now free;
- to present any previously stacked status when allowed to do so;
- to present any externally initiated status to the channel because of not-ready to ready transition; and
- Conclusion of a *Test I/O* command

Designations of the status conditions, and their positions within the status byte and CSW, are as follows:

DEVICE STATUS BYTE, BIT POSITION	CSW BIT	DESIGNATION
P		PARITY
0	32	ATTENTION (NOT USED)
1	33	STATUS MODIFIER
2	34	CONTROL UNIT END
3	35	BUSY
4	36	CHANNEL END
5	37	DEVICE END
6	38	UNIT CHECK
7	39	UNIT EXCEPTION

3.6.2 Description of Status Conditions

The status conditions that are detected by the disc drive, controller, or SCU and indicated to the channel over the I/O interface by the presentation of the status byte, are described below. Once the conditions are accepted by the channel, the status byte is not presented again.

3.6.2.1 STATUS MODIFIER

Status Modifier is used in three situations:

- When presented with Busy, to differentiate a busy SCU from a busy controller/disc drive during the initial selection sequence.
- When presented with Device End, to recognize special ending conditions, such as Search Equal, when the special condition occurs. Recognition of special ending (or synchronizing) conditions indicates that the normal sequence of commands must be modified.
- When presented with Unit Check to indicate that an unusual condition, calling for a retry of the last channel command, has occurred.

A busy SCU but not-busy controller/disc drive occurs when the SCU is required to perform a function that does not involve the I/O interface, or the SCU has status pending for a disc drive other than the one addressed.

3.6.2.2 CONTROL UNIT END

The SCU provides Control Unit End if the SCU was previously addressed while in the busy state. The busy state is defined in the next paragraph.

3.6.2.3 BUSY

Busy can occur only during a channel-initiated-selection sequence. It indicates that the disc drive, controller, or SCU cannot execute the command because of one of the following reasons:

- A previously initiated operation is being executed. An operation is being executed from the time initial status is accepted until Device End is accepted.
- Status conditions exist. The conditions accompany the Busy indication. If the Busy indication pertains to an SCU function, it is accompanied by Status Modifier.
- Unavailable path exists on a string-switch controller.

Busy is indicated to the *Test I/O* command only if a previously initiated operation is still being executed and no end status is available. It causes command chaining to be suppressed.

3.6.2.4 CHANNEL END

Channel End indicates the portion of any I/O operation involving the transfer of control information or data has been completed. Each I/O operation can cause only one Channel End signal to be generated. The signal is not generated unless the command is accepted. Acceptance is indicated by the status byte containing either all zeros, or Channel End and not Busy. The exact time when Channel End is generated depends on the operation, as follows:

- For operations such as reading, Channel End is generated when the block has been read.
- During normal control operations, Channel End is usually generated after the control information is transferred to the SCU. Channel End may be delayed until the completion of short control operations.
- For operations that do not cause data to be transferred, Channel End may be provided during the initial selection sequence.

3.6.2.5 DEVICE END

Device End is caused by completing an I/O operation at the disc drive or controller, or by manually changing the disc drive from the not-ready to ready state. ("Not ready" means that the disc drive or controller requires operator intervention to become operational.) It normally indicates that the disc drive or controller has completed the current operation. Each I/O operation can generate only one Device End signal. The signal is not generated unless the command is accepted.

Device End is generated either with Channel End or later. For data transfer operations, the disc drive terminates the operation when Channel End is generated, and both Channel End and Device End occur together. For control operations, Device End is generated upon completing the operation at the disc drive. The operation may be completed either at the time Channel End is generated or later. Generation of Device End in special cases is described as follows:

- When command chaining occurs, only the Device End of the last operation of the chain is normally made available to the program.
- When Device End is received in the absence of any unusual conditions while chaining, it causes the channel to initiate a new I/O operation. If an unusual condition is detected during the initiation of a chained command, the chain is terminated without Device End. Sense information is available defining the unusual condition.
- An I/O device shared between more than one channel path and which generates Device End because of the disc drive going from a not-ready to ready state must present Device End to all attached channels if the Tag switch is in the TAG position. Device End will be accepted by only one channel if the switch is in the UNTAG position.
- If a disc drive is addressed while busy, Device End must be signalled to the path that initiates the command when the disc drive becomes not busy.

3.6.2.6 UNIT CHECK

Unit Check status indicates that the SCU, controller, or disc drive has detected an unusual condition that is described by the information available to a Sense command. For example:

- Indication that an equipment or programming error has occurred.
- The not-ready state of the disc drive or controller has affected the execution of the command.
- Exceptional conditions other than the one identified by Unit Exception have occurred.

An error condition causes Unit Check only when it occurs during execution of a command or some activity associated with an I/O operation. Unless the error condition pertains to the activity initiated by a command and is of significance to the program, the condition does not cause the program to be alerted until after Device End has been cleared. A malfunction may, however, cause the disc drive to become not ready.

NOTE

The ending interruption condition can be cleared by *Test I/O* without generation of Unit Check if a disc drive becomes not ready on completion of a command.

Unit Check status is presented when the not-ready state interferes with proper execution of the command, or when the command, by its nature, tests the state of the disc drive as follows:

- Termination of an operation with Unit Check status causes command chaining to be suppressed.
- If the condition that precludes proper execution of the operation occurs after execution has started, Unit Check accompanies Channel End, Control Unit End, or Device End, depending upon when the condition was detected. Errors detected after Device End is cleared may be indicated by Unit Check with Control Unit End.
- When Unit Check appears with Channel End and without Device End, a shared SCU must preserve the sense data and an available disc drive path until after Device End is accepted.
- Errors, such as invalid command code parity or invalid command code, do not cause Unit Check when the disc drive is working or contains a pending interruption condition at the time of selection. In these situations, the SCU responds by providing Busy and the pending disc drive interruption condition, if any. The command code validity is not indicated.
- Selection of a disc drive in the not-ready state does not cause a Unit Check indication when a Sense command is issued, and whenever an interruption condition is pending for the addressed disc drive at the SCU.
- Unless a command is designed to cause Unit Check, Unit Check is not indicated if the command is properly executed, even though the disc drive has become not ready either during, or as a result of, the operation. Similarly, Unit Check is not indicated if the command can be executed with the disc drive not ready.
- If, during the initial selection sequence, the SCU detects that the command cannot be executed, Unit Check status is presented to the channel. It appears without Channel End, Control Unit End, or Device End. This status indicates that no action has been taken at the disc drive in response to the command.

3.6.2.7 UNIT EXCEPTION

Unit Exception indicates that the SCU detected an unusual condition such as end of file. It has only one meaning for any particular command. A sense operation is not required as a response to the acceptance of a Unit Exception status condition.

Unit Exception conditions can be generated only when the disc drive is involved with some activity associated with an I/O operation and the condition is of immediate significance.

If a condition that precludes normal execution of an operation occurs after an operation is started, Unit Exception is accompanied by Channel End, Control Unit End, or Device End, depending on when the condition is detected. Unit Exception suppresses command chaining.

3.7 SENSE CONDITIONS

3.7.1 Overview

Conditions that occur during the execution of an instruction command sequence are reported as channel status and as sense information. Sense information defines conditions that occurred in the SCU, controller, and disc drive that are not defined in the status. This sense information is obtained by executing a Sense command.

Information concerning the actual state of the SCU, controller, and disc drive; and unusual conditions detected in the last operation, are provided by data transfer during a sense operation. It may describe reasons for the Unit Check indication of a status byte and, for example, may also contain an indication that the susystem is in the not-ready state.

Under certain conditions, the sense information that pertains to the last I/O operation or other unit action may be reset by the next command addressed to the SCU. These conditions are that the Busy bit is not included in the initial-selection status byte, except where the command is a *Sense I/O* or a *Test I/O* instruction addressed to the SCU.

The sense information may be changed when not-ready to ready Device End status is generated, or as a result of other such asynchronous actions.

A total of 24 bytes of sense information is transferred to the system by the SCU when it receives a *Sense I/O* command. The first two bytes normally provide all sense information significant to the use of the subsystem. The bit-by-bit meaning of messages reported in bytes 0 through 6 is preassigned. The meaning of messages reported in bytes 8 through 23 depends on the format used for these bytes. This format is decoded in byte 7, bits 0 to 3.

3.7.2 Conditions Indicated by Bits of Sense Byte 0

The bits making up sense byte 0 are independent of each other and are designated as follows:

BIT	DESIGNATION
0	Command Reject
1	Intervention Required
2	Bus-Out Parity
3	Equipment Check
4	Data Check
5	Overrun
6	(not used)
7	(not used)

Command Reject—Bit 0

This bit indicates that the command cannot be performed for any of several reasons. Typical examples are listed below:

- Command op code, or its information, is invalid. For example, a Read command was received which the subsystem is not designed to execute (Read Backward issued to a drive).
- Sequence of commands or the protected conditions has been violated. For example, a Write command was received which the subsystem cannot execute because the command violates the file mask.
- Write portion of the file mask has been violated. For example, an invalid sequence of commands was recognized (a Write command is issued to a drive without previously performing Search).

An additional sense bit may define the reason for the rejection.

Intervention Required—Bit 1

This bit indicates that the subsystem is either not connected or for some reason is not ready.

Intervention Required is generated when the subsystem is in the test mode or the not-ready state. It is also generated when the last operation could not be executed because of a condition that requires intervention at the subsystem. For example:

- The addressed subsystem was in the not-ready state.
- A *Diagnostic Write* or *Diagnostic Load* command was issued, but the microdiagnostic is not resident in the SCU.

Bus-Out Parity—Bit 2

Bus-Out Parity is generated when the SCU detects invalid parity in a data or command byte received from the channel. It indicates that incorrect data may have been recorded during a write operation. However, writing the incorrect data does not cause the write operation to be terminated prematurely, unless an error prevents meaningful continuation of the operation.

Equipment Check—Bit 3

This bit indicates that an unusual hardware condition, originated in the channel, SCU, controller, or disc drive has been detected. The condition is defined by the bits in sense bytes 7 through 23.

During output operations, the unusual condition or malfunction may have caused invalid data to be recorded. When the malfunction prevents any meaningful continuation of the operation, Equipment Check is generated to stop data transmission and terminate the operation prematurely.

Data Check—Bit 4

This bit indicates that one of the following errors has been detected:

- A correctable data error has been detected in the information received from a disc drive (byte 2, bit 1 on; correction information is provided in sense bytes 15 through 19).
- An uncorrectable data error has been detected in the information received from a disc drive (condition further defined in sense byte 7).

The bit is set for an error in the count field with a secondary indication but is not set for an error in the key field during a read data operation.

The SCU forces correct parity on data sent to the channel. Data errors on reading and writing cause the operation to be terminated prematurely only when the errors prevent meaningful continuation of the operation.

Overrun—Bit 5

This bit indicates that one of the following conditions has occurred:

1. The SCU received a byte from a disc drive before the last byte read was accepted by the channel.
2. During a write operation, a data byte was received too late from the channel.
3. When command chaining is indicated, the channel does not respond with subsequent commands in a timely fashion.
4. A channel-discontinued retry is signalled on a channel which has the 2860 Attachment Feature installed, and the *RPS* command state is 0. See paragraph 3.12.

Overrun can occur when the total activity of the program exceeds the capability of the channel. During output operations, the Overrun condition indicates that data recorded at the subsystem may be invalid. In these cases, data transfer is stopped and the operation is terminated.

3.7.3 Conditions Indicated by Bits of Sense Bytes 1 through 23

For designations of bytes 1 through 23, and conditions indicated by their individual bits, see paragraph 2.5.

3.8 READ DATA TRANSFER

3.8.1 Description

This paragraph provides a simplified block diagram description of the Read Data Transfer operation. Paragraph 3.8.2 provides a flowchart of this operation. The alphabetical keys adjacent to lines of the block diagram in Figure 3-7 are keyed to the text material below.

[A] NUMBER OF BYTES TO BE TRANSFERRED

1. Before entering the Read Data Transfer sequence, the Channel Buffer Ready (CBR) and the Queue-Empty (Q.EMPTY) branch conditions in the microprogram are set by Channel Transfer bit 0 in the CHC Register being down or reset (paragraph 6.2.2.5).

2. Microprogram loads the CC Register with the number of bytes to be transferred (paragraph 6.2.2.5).

3. Loading the CC Register causes the CC = 0 latch to be reset in the microprocessor.

[B] CHANNEL TRANSFER BIT AND CHANNEL WRITE BIT

1. Microprogram initiates data transfers by setting Channel Transfer bit 0 in the CHC Register. This gates the two-byte buffer into the data transfer path.

2. At the same time, the microprogram establishes a read operation by resetting Channel Write bit 1 in the CHC Register.

[C] CHANNEL BUFFER READY AND QUEUE EMPTY BRANCH CONDITIONS

When the CBR condition is set, it indicates that one or both buffers have not been loaded by the microprogram. When the Q.EMPTY condition is set, it indicates that both buffers are empty.

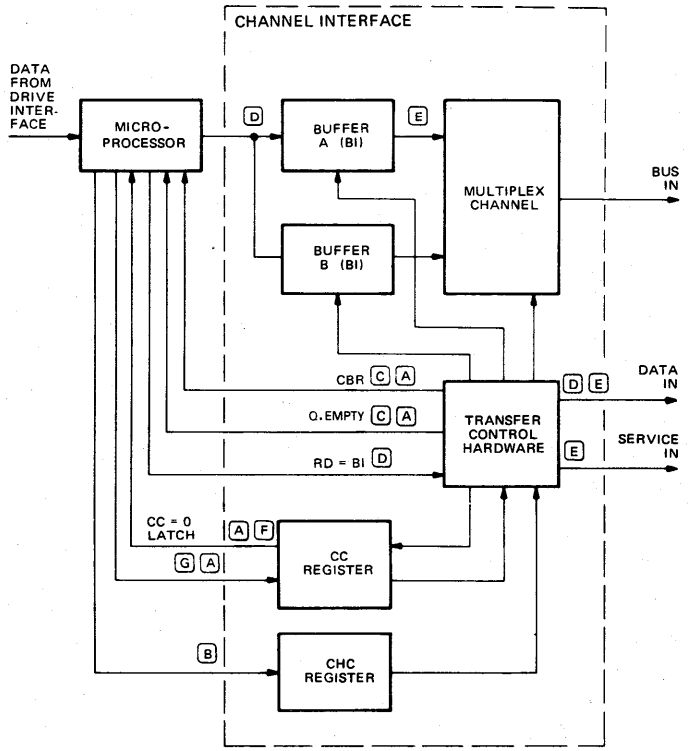


Figure 3-7. Read Data Transfer Block Diagram

[D] MICROPROGRAM RESPONSE TO CBR

1. After checking for CBR, the microprogram executes an RD = BI statement and loads a byte into Buffer A. The Transfer Control Hardware places this byte on the Bus In lines to the channel and raises DATA IN. Because one buffer now contains a byte of data, the Q.EMPTY branch condition is reset in the microprocessor.

2. Because there is still one available buffer for more data, the CBR branch condition remains set.

3. At this point in the control sequence, one of the following occurs:

   a. Channel responds to DATA IN with DATA OUT or COMMAND OUT. DATA OUT indicates that the byte has been accepted. COMMAND OUT indicates truncation and that the byte has been rejected by channel. A response of DATA OUT causes the Q.EMPTY branch condition to be set. If the CC Register is decremented to zero, DATA IN and SERVICE IN are inhibited from being raised until the CC Register is reloaded.

   b. Microprogram loads another byte into Buffer B. This causes the CBR branch condition to be reset in the microprocessor. Both buffers then contain data that has not been accepted by the channel. The microprogram cannot load another byte of data into the buffers until the channel accepts at least one of the buffered bytes.

[E] DATA TRANSFER

Data is transferred to the channel by the Transfer Control Hardware alternately raising DATA IN and SERVICE IN to transfer successive bytes. Each time a byte is transferred, the CC Register is decremented to show how many bytes remain to be transferred to the channel.

F END OF DATA TRANSFER

Transfer of data to channel is ended by one of the following conditions:

1. When the CC Register is decremented to zero, the CC = 0 latch is set in the microprocessor, and the Transfer Control Hardware is inhibited from raising SERVICE IN or DATA IN. This stops transfer of data.

2. Channel truncates data transfer by raising COMMAND OUT in response to SERVICE IN or DATA IN. When this occurs, the CC Register is not decremented, the Q.EMPTY condition is not set, and the CBR condition can be set or reset. When the microprogram detects truncation, it resets bit 0 in the CHC Register. This stops transfer of data.

3. The channel issues a *Halt I/O* or *Halt Device* command to the SCU by raising ADDRESS OUT. This causes the microprogram to reset Channel Transfer bit 0 in the CHC Register. This stops the transfer of data.

G SENSE COMMANDS

For sense commands where data transfer is a read operation over channel and data need not be transferred regularly, the microprogram indicates Suppressible Data Transfer by setting bit 5 in the CHC Register. Transfer operation is identical to transfer described in paragraphs A through F, except that the Transfer Control Hardware is inhibited from raising DATA IN or SERVICE IN to transfer a new byte while SUPPRESS OUT is raised by the channel. When SUPPRESS OUT is down, the Transfer Control Hardware is not inhibited.

3.8.2 Flow Diagram

A flow diagram of the Read Data Transfer operation, listing events that occur in the channel, channel interface, and microprogram, is shown in Figure 3-8. Each event listed is preceded by a mnemonic that indicates where the event occurs during the operation: channel (C), channel interface (CI), or microprogram (M).

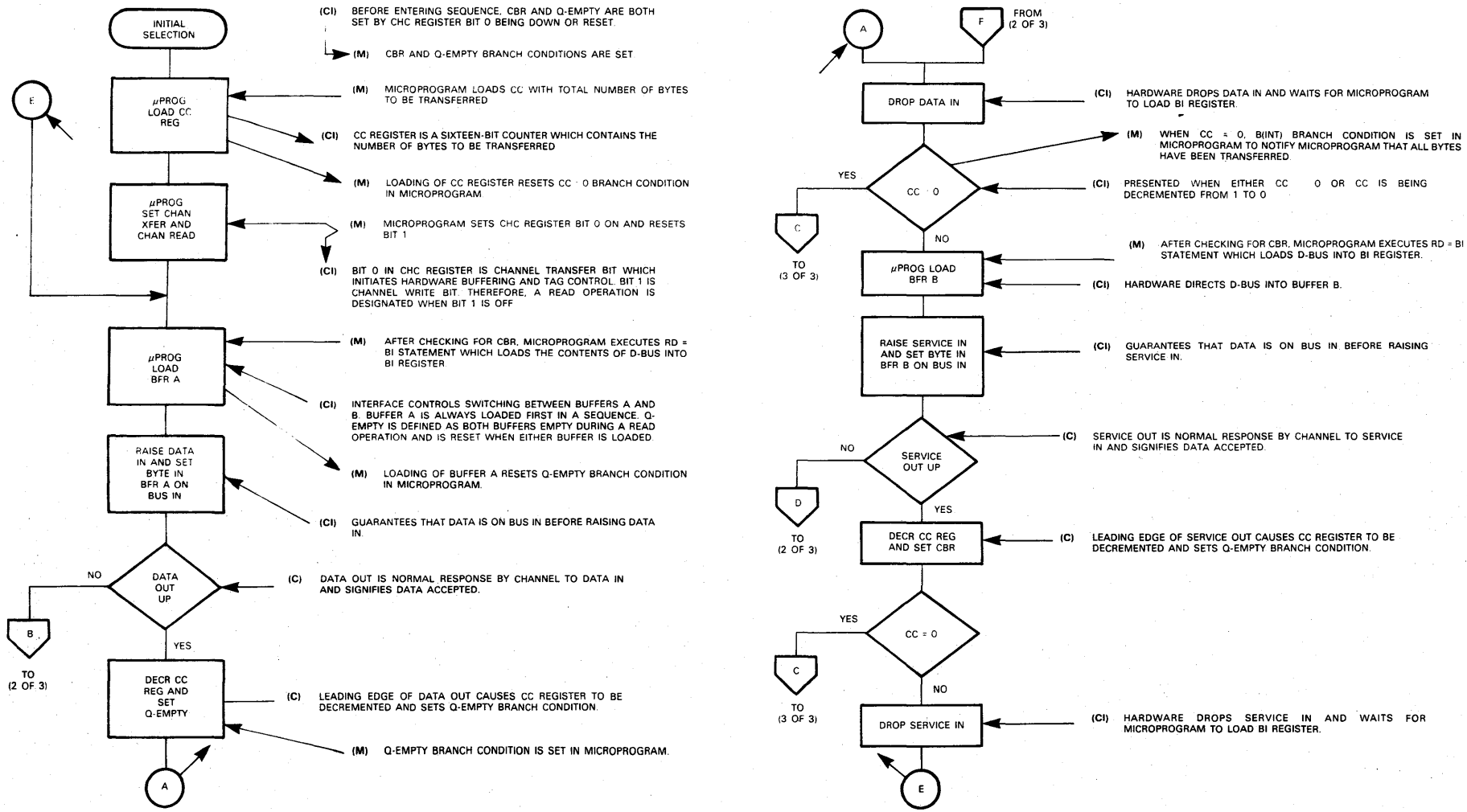


Figure 3-8. Read Data Transfer Flow Diagram (1 of 3)

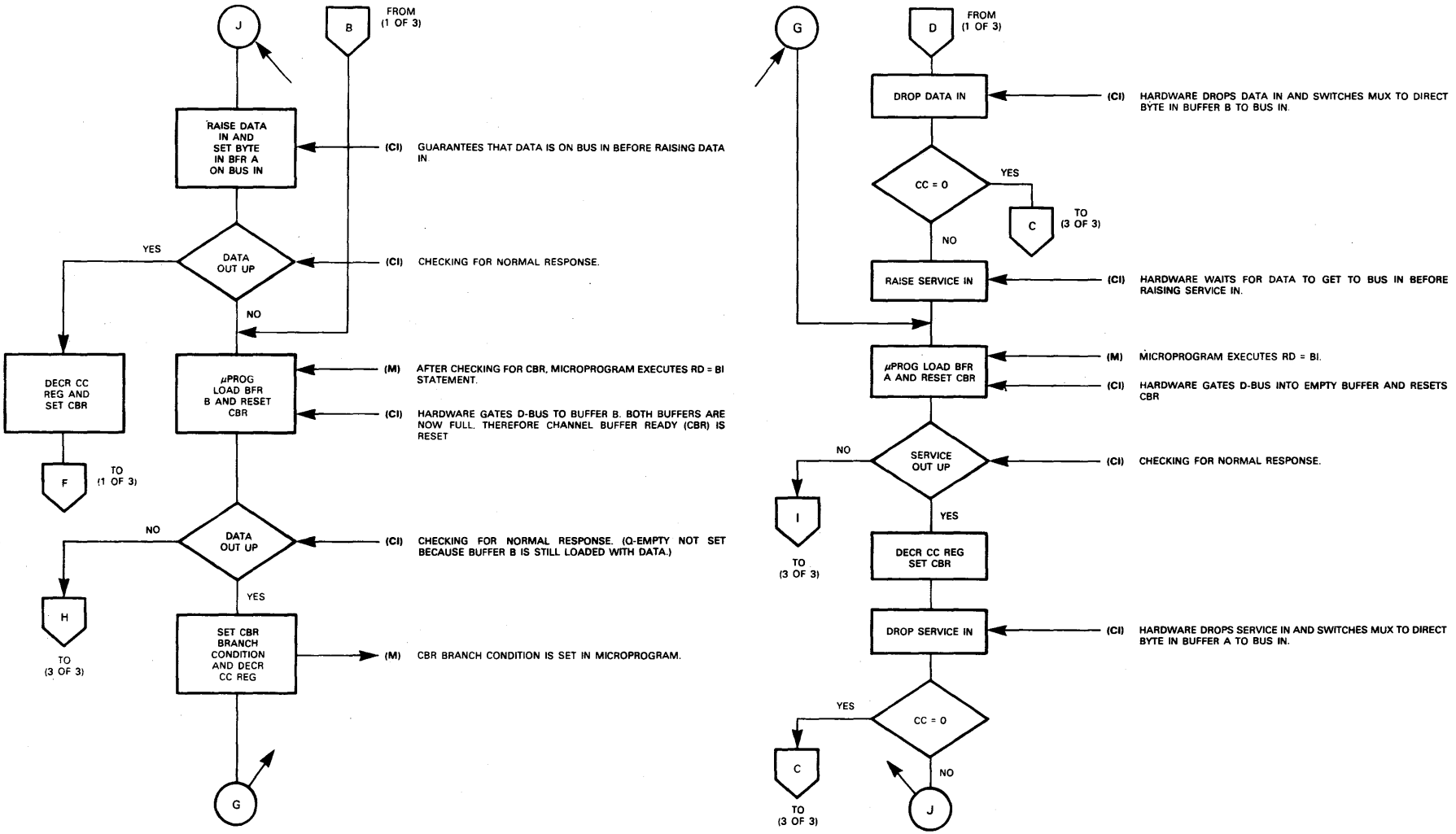


Figure 3-8. Read Data Transfer Flow Diagram (2 of 3)

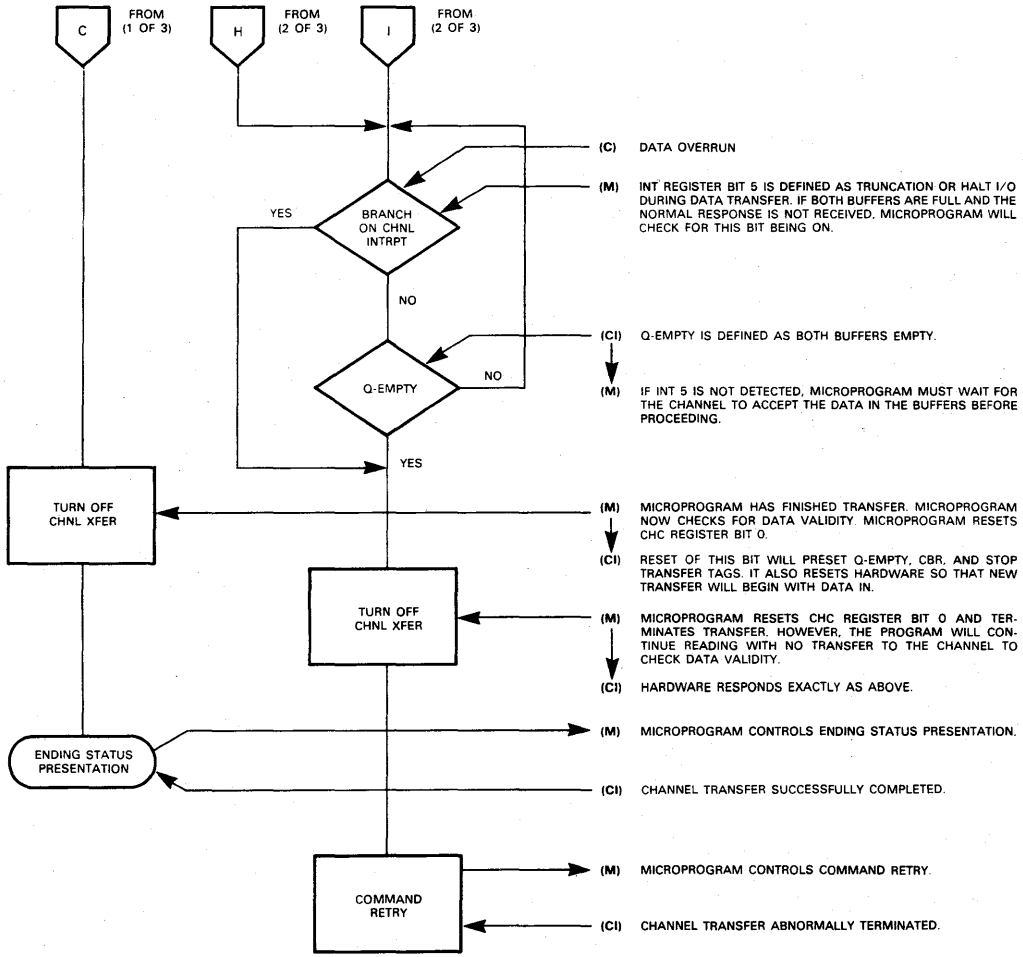


Figure 3-8. Read Data Transfer Flow Diagram (3 of 3)

3.9 WRITE DATA TRANSFER

3.9.1 Description

This paragraph provides a simplified block diagram description of the Write Data Transfer operation. Paragraph 3.9.2 provides a flowchart of this operation. The alphabetical keys adjacent to lines of the block diagram in Figure 3-9 are keyed to the text material below.

[A] NUMBER OF BYTES TO BE TRANSFERRED

1. Microprogram loads the CC Register with number of bytes to be transferred (paragraph 6.2.2.5).

2. Loading the CC Register causes the CC = 0 latch to be reset in the microprocessor.

[B] CHANNEL TRANSFER BIT AND CHANNEL WRITE BIT

1. Microprogram initiates data transfers from the channel to the channel interface by setting Channel Transfer bit 0 in the CHC Register. This gates the two-byte buffer into the data transfer path.

2. At the same time, the microprogram establishes a write operation by setting Channel Write bit 1 in the CHC Register.

[C] CHANNEL BUFFER READY BRANCH CONDITION

1. One instruction cycle after Channel Write bit 1 has been set, the Transfer Control Hardware resets the CBR branch condition in the microprocessor.

2. CBR reset condition indicates to the microprogram that data is not yet available from the channel.

[D] REQUESTING FIRST DATA BYTE FROM CHANNEL

1. Resetting the CBR condition, after setting Channel Transfer bit 0 and Channel Write bit 1 with CC = 0 latch reset, causes the Transfer Control Hardware to raise DATA IN. Raising DATA IN requests the first data byte from channel.

2. CBR remains reset until the first byte is transferred from the channel.

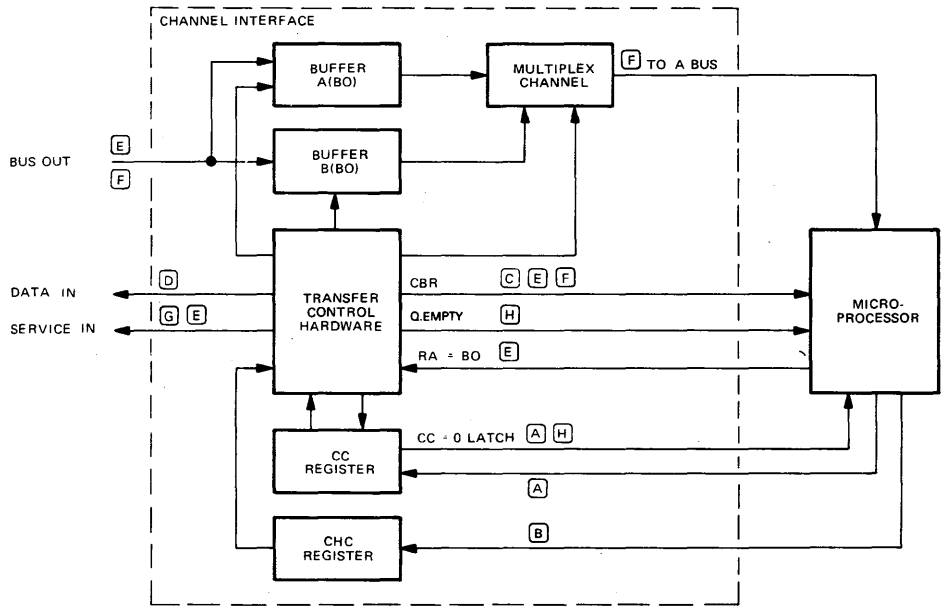


Figure 3-9. Write Data Transfer Block Diagram

[E] TRANSFER OF FIRST DATA BYTE

1. When the channel transfers the first data byte, the Transfer Control Hardware gates the byte into one of the buffers.

2. Transfer Control Hardware then sets the CBR branch condition to indicate to the microprogram that data is available in one of the buffers.

3. Because one buffer is still available for data, Transfer Control Hardware immediately raises SERVICE IN to request another data byte.

[F] CONTROL SEQUENCE ALTERNATIVES

At this point in the control sequence, one of the following occurs:

1. Microprogram responds to the CBR branch condition by causing the Transfer Control Hardware to gate the first data byte to Bus A. This causes the CBR branch condition to be reset.

2. Channel transfers another data byte to channel interface (since one buffer remains empty). This causes the CBR branch condition to remain set. The Transfer Control Hardware cannot request another byte from the channel until the microprogram accepts at least one of the data bytes. Each transfer of a data byte from the channel causes the CC Register to be decremented. If the CC Register is decremented to zero, DATA IN and SERVICE IN are inhibited from being raised until the CC Register is reloaded.

[G] TRANSFER OF REMAINING DATA BYTES

Data is transferred from the channel by the Transfer Control Hardware alternately raising

DATA IN and SERVICE IN to transfer successive bytes. Each time a byte is transferred, the CC Register is decremented to show how many bytes remain to be transferred from channel.

[H] END OF DATA TRANSFER

Transfer of data from channel is ended by one of the following conditions:

1. When the CC Register is decremented to zero (data transfer completed), the microprogram resets Channel Transfer bit 0 and Channel Write bit 1 in the CHC Register. This stops transfer of data.

2. Channel truncates data transfer by raising COMMAND OUT in response to SERVICE IN or DATA IN. When this occurs, the CC Register is not decremented and the CBR condition is not set (if it was reset). When the microprogram detects truncation, it executes a Channel Freeze control statement. This Channel Freeze control statement essentially has no effect, but is executed because of timing requirements for overrun conditions.

3. When the channel overruns the disc drive data transfer rate by not responding to SERVICE IN or DATA OUT before a data byte is required by the SCU, the microprogram executes a Channel Freeze control statement. Executing this control statement causes the present SERVICE IN or DATA IN transfer sequence to be completed, and inhibits initiation of any new transfer sequences. If SERVICE IN and DATA IN are both reset when Channel Freeze is executed, the data transfer is halted immediately. After Channel Freeze control is executed on data overruns, the Q.EMPTY branch condition in microprocessor is set by the Transfer Control Hardware when the present transfer sequence has been completed and no new sequences are to be initiated. When the Q.EMPTY branch condition is set, the microprogram then resets Channel Transfer bit 0 and Channel Write bit 1 in the CHC Register.

4. Channel issues a *Halt I/O* or *Halt Device* command to the SCU by raising ADDRESS OUT. This causes the microprogram to reset Channel Transfer bit 0 and Channel Write bit 1 in the CHC Register. This stops transfer of data.

[I] SUPPRESSIBLE DATA TRANSFER

For seek commands where data transfer is a write operation over channel and data need not be transferred regularly, the microprogram indicates Suppressible Data transfer to the Transfer Control Hardware by setting bit 5 in the CHC Register. This transfer operates identically to that described in [A] through [H], except that the Transfer Control Hardware is inhibited from setting DATA IN or SERVICE IN to transfer a new byte while SUPPRESS OUT is set at the channel. When SUPPRESS OUT is down, the Transfer Control Hardware is not inhibited.

3.9.2 Flow Diagram

A flow diagram of the Write Data Transfer operation, listing events that occur in the channel, channel interface, and microprogram, is shown in Figure 3-10. Each event is preceded by a mnemonic that indicates where the event occurs during the operation: channel (C), channel interface (CI), or microprogram (M).

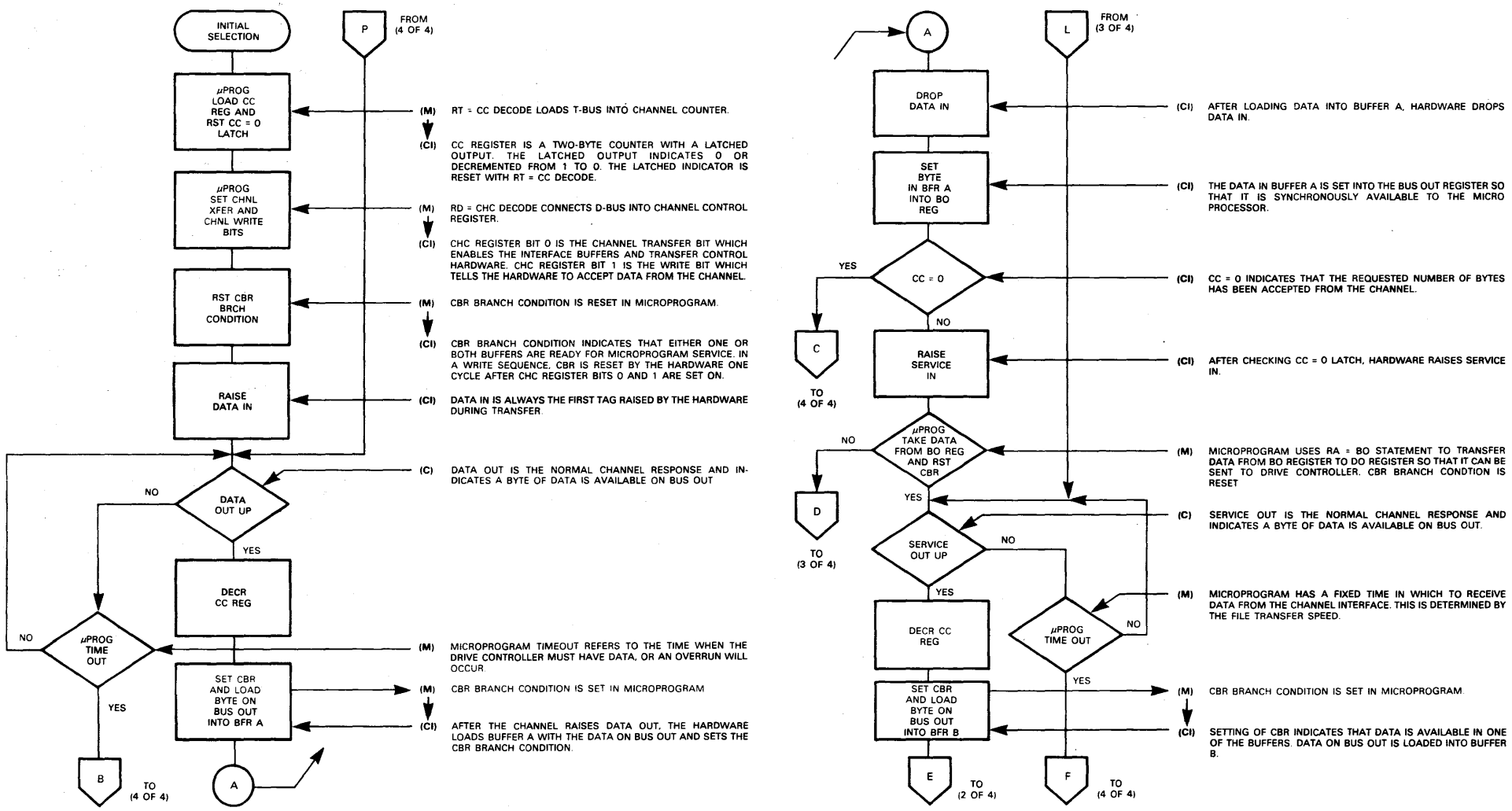


Figure 3-10. Write Data Transfer Flow Diagram (1 of 4)

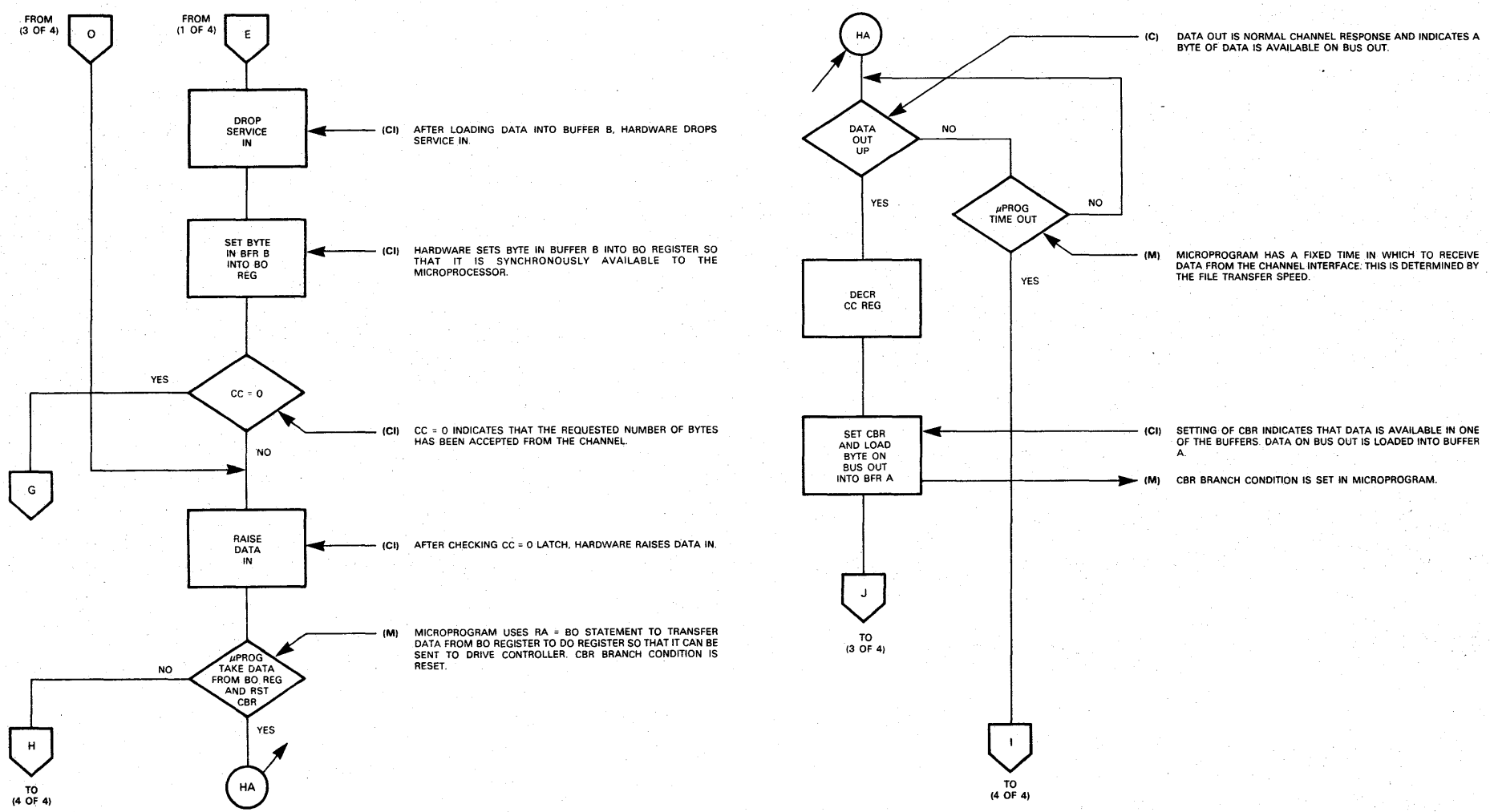


Figure 3-10. Write Data Transfer Flow Diagram (2 of 4)

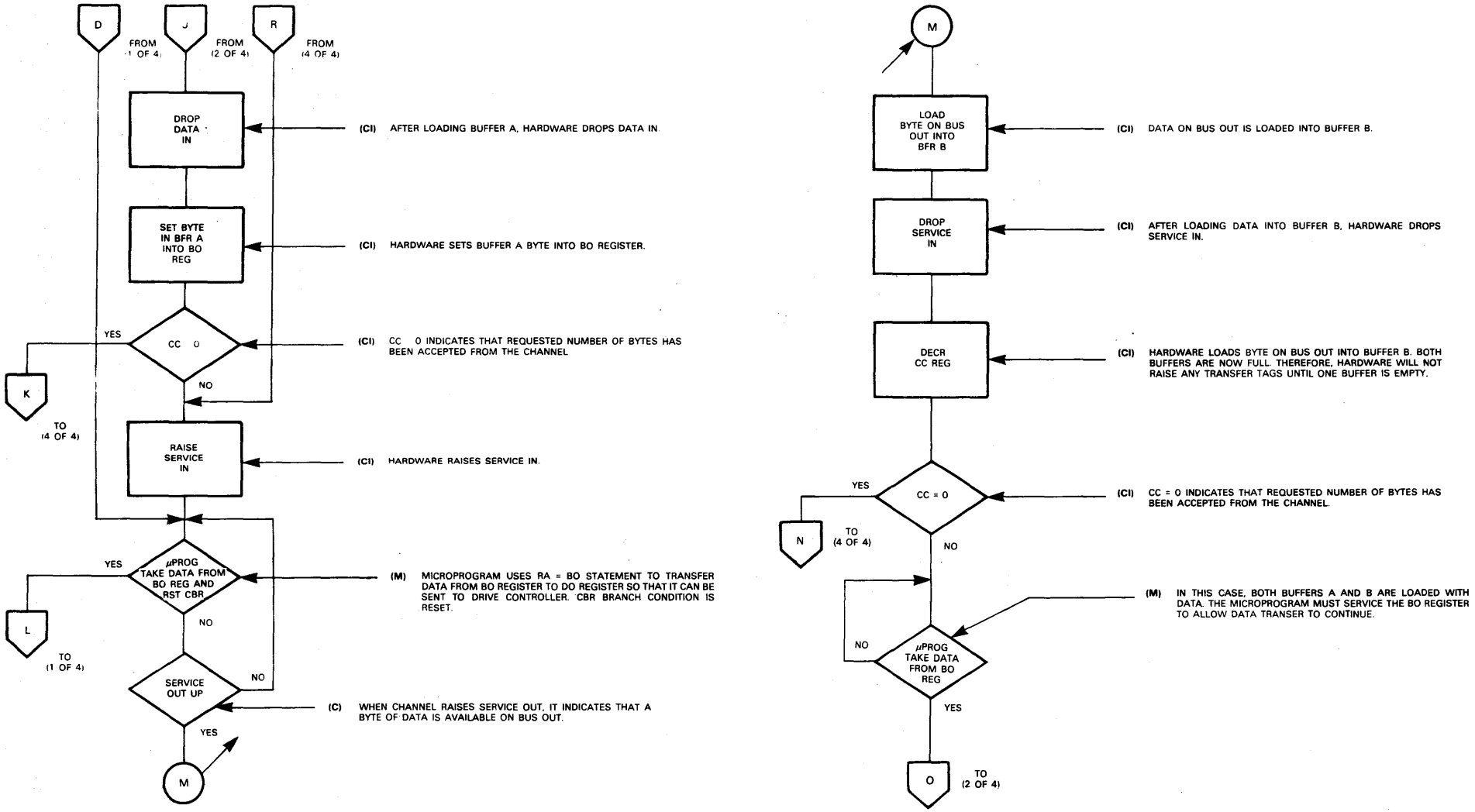


Figure 3-10. Write Data Transfer Flow Diagram (3 of 4)

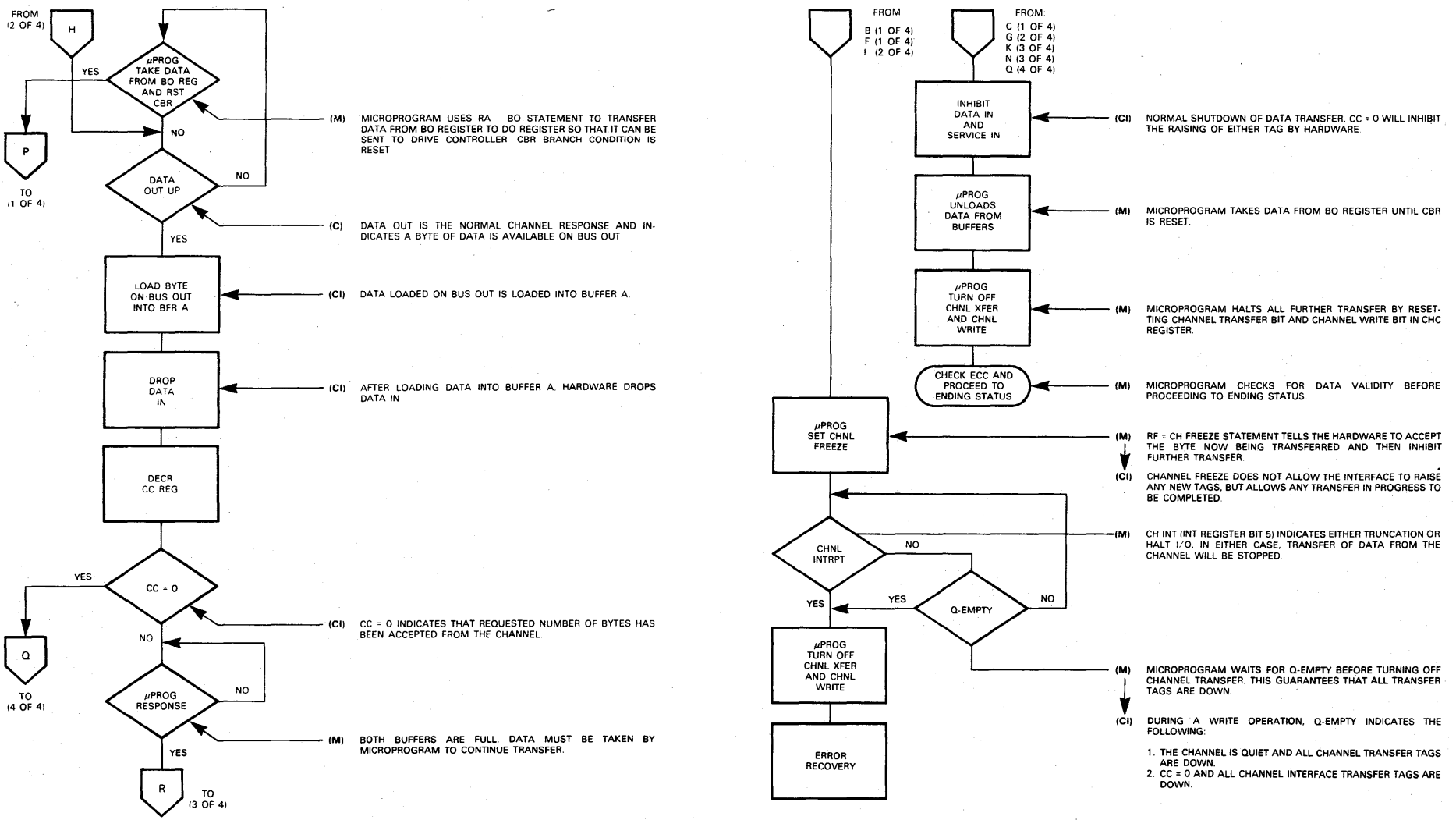
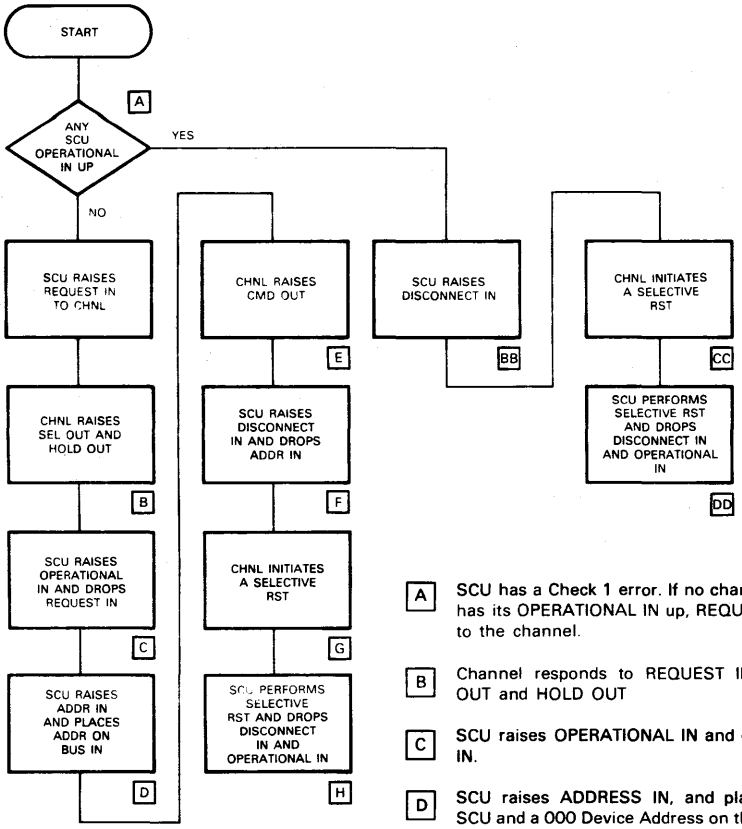
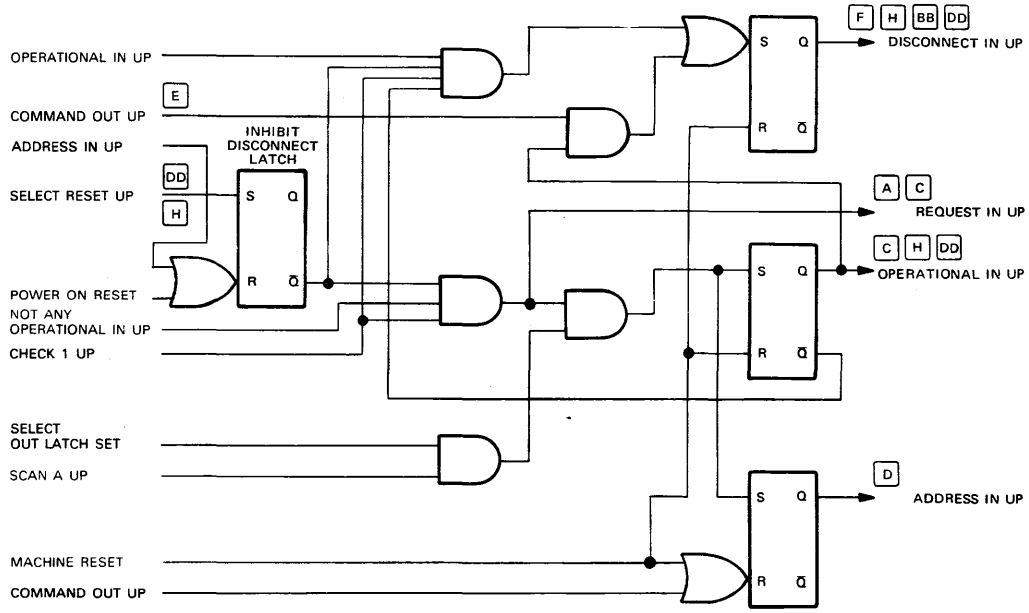


Figure 3-10. Write Data Transfer Flow Diagram (4 of 4)

3.10 SCU-INITIATED CHECK 1 ERROR CONTROL SEQUENCE (POLLING), SIMPLIFIED

The flowchart and simplified logic diagram shown in Figure 3-11 illustrate an SCU-Initiated Check 1 Error Control Sequence. The flowchart and logic diagram are keyed to the text material below.





A SCU has a Check 1 error. If no channel in this SCU has its OPERATIONAL IN up, REQUEST IN is raised to the channel.

B Channel responds to REQUEST IN with SELECT OUT and HOLD OUT

C SCU raises OPERATIONAL IN and drops REQUEST IN.

D SCU raises ADDRESS IN, and places address of SCU and a 000 Device Address on the Bus In lines.

E Channel stores address and responds with COMMAND OUT.

F SCU raises DISCONNECT IN and drops ADDRESS IN.

G Channel initiates a selective reset by raising SUPPRESS OUT and dropping OPERATIONAL OUT.

H SCU performs selective reset and drops DISCONNECT IN and OPERATIONAL IN.

AA If the SCU with the Check 1 error is already selected (its OPERATIONAL IN is up) and its address has already been given to the channel, the control sequence proceeds as follows:

BB SCU raises DISCONNECT IN.

CC Channel initiates a selective reset by raising SUPPRESS OUT and dropping OPERATIONAL OUT.

DD SCU performs a selective reset and drops DISCONNECT IN and OPERATIONAL IN

Figure 3-11. Check 1 Error Controlling Sequence Flowchart and Logic Diagram

3.11 MULTI-CHANNEL SWITCH OPERATION

The Multi-Channel Switch (MCS) feature permits the SCU to be shared by up to four selector or block multiplexer channels in any combination; i.e., four selector channels, four block multiplexer channels, or any combination thereof. The channels may be attached to the same or different central processing units. Any one of the channels may be reserved exclusively by any of the drives attached to the SCU. Channel switching and device reservation are controlled by the channel program. Two special commands are invoked for MCS operation: *Device Reserve* and *Device Release*. Refer to the MCS logic shown in Figure 3-12 throughout the discussion that follows.

3.11.1 Channel Selection Switch

Once the SCU has been selected by any channel, it is reserved by that channel until Ending Status is presented. The SCU can then be selected by any channel unless:

- The last status byte was part of a channel-initiated signal sequence and was stacked by the channel.
- A contingent connection is established.
- Chaining is indicated and a format write operation is in progress.
- Chaining is indicated without Device End in the status byte, the channel disconnects, and the SCU becomes busy to allow:
  a. Completion of a format write operation.
  b. Execution of a *Diagnostic Load* or *Diagnostic Write* command.
  c. Execution of an SCU error recovery procedure.
- Chaining is indicated without Device End in the status byte, and the channel does not disconnect.
- Chaining is indicated and Device End is included in the status byte.
- Ending status associated with an interface disconnect has not been accepted by the channel.

When a channel connection is maintained as a result of a contingent connection, the SCU will not respond to polling by the channel, except to present Control Unit End. When the contingent connection is terminated, the SCU can be selected by any channel.

The length of time the SCU is connected to a channel determines how the channel selection switch responds to a channel-initiated sequence from the other channels. The channel/SCU connection is classified as "instantaneous" or "long."

NOTE

In the following descriptions, the conditions of any of the channels may be interchanged.

3.11.2 Instantaneous Connection

At the moment the channel selection switch connects the SCU to a channel, the connection is considered to be instantaneous until otherwise determined by the SCU. The state of a channel connection changes from instantaneous to long if:

- Channel is processing a *Start I/O* instruction, or
- Channel indicates chaining on Device End during a polling sequence.

If a channel/SCU connection is caused by a condition other than the above, the state of the connection remains instantaneous until the operation is complete.

During an instantaneous connection on channel A, the selection logic does not respond to a channel-initiated sequence on any other channel until the instantaneous connection is terminated. When the instantaneous connection terminates, the selection logic either remains connected to channel A (if the connection state changed from instantaneous to long), or the SCU becomes available to all channels (neutral). If the switch returns to neutral from channel A during a channel-initiated selection sequence on any other channel, the logic immediately connects to the applicable channel. If the connection state changed from instantaneous to long on channel A, the response to the selection logic is as defined in paragraph 3.11.3.

3.11.3 Long Connection

During a long connection on channel A, the SCU responds with a short Control Unit Busy Sequence (busy and status modifier) to a channel-initiated selection sequence on any other channel. Whenever the short Control Unit Busy Sequence occurs, the microprogram attempts to present Control Unit End status to the appropriate channel once the selection logic returns to neutral. The address byte associated with this status is the address of the lowest numerically addressed device which is not implicitly connected to another interface. The pending Control Unit End status does not cause the SCU to appear busy to channel A as long as the selection switch is not actually connected to any one of the other channels.

3.11.4 Device Status

Device End status resulting from a not-ready-to-ready transition will be presented under control of the TAG/UNTAG switch. A not-ready-to-ready sequence Device End occurs after a disc pack change is completed and the affected module is returned to an on-line condition.

When the switch is in the TAG position, Device End for the not-ready-to-ready sequence is presented to all channels. After Device End is accepted by a particular channel, the module can be addressed for command execution by that channel. Before any other channel can use the module, it must also receive the not-ready-to-ready sequence Device End.

When the switch is in the UNTAG position, Device End for the not-ready-to-ready sequence is also presented to all channels. However, this Device End is cleared as soon as it is accepted by any channel, and no further attempt is made to present Device End to any other channel.

When a device is busy for any reason (including reservation to channel A), any command from another channel addressed to that device will be rejected with a Busy status. This, in turn, causes the SCU to attempt sending a status byte containing Device End to the appropriate channel after the busy condition has been terminated. The address byte associated with this status byte will be the same as that associated with the busy status byte.

Device End status resulting from any channel command will be presented to the channel that issued the command.

3.11.5 Addressing

The base address (four high-order bits for 16-spindle configuration, three for 32-spindle configuration) of the storage control on one channel is independent of the base addresses on the other channels. However, the three low-order address bits for any attached device must be the same on all channels.

3.11.6 Resets

System Reset clears all reservations and status conditions stored in the SCU for the resetting channel, terminates all block multiplex command chains in progress on the resetting channel, and resets all device interrupts not associated with the other channels. The reset may be initiated by any channel at any time. Reservations, status, and device interrupts for the other channels, as well as block multiplex chains in progress on the other channels, are not affected. If a channel initiates a System Reset while the selection logic is connected to another channel, a Machine Reset is performed when the selection logic goes to neutral. A Selective Reset has no effect on device reservations or status.

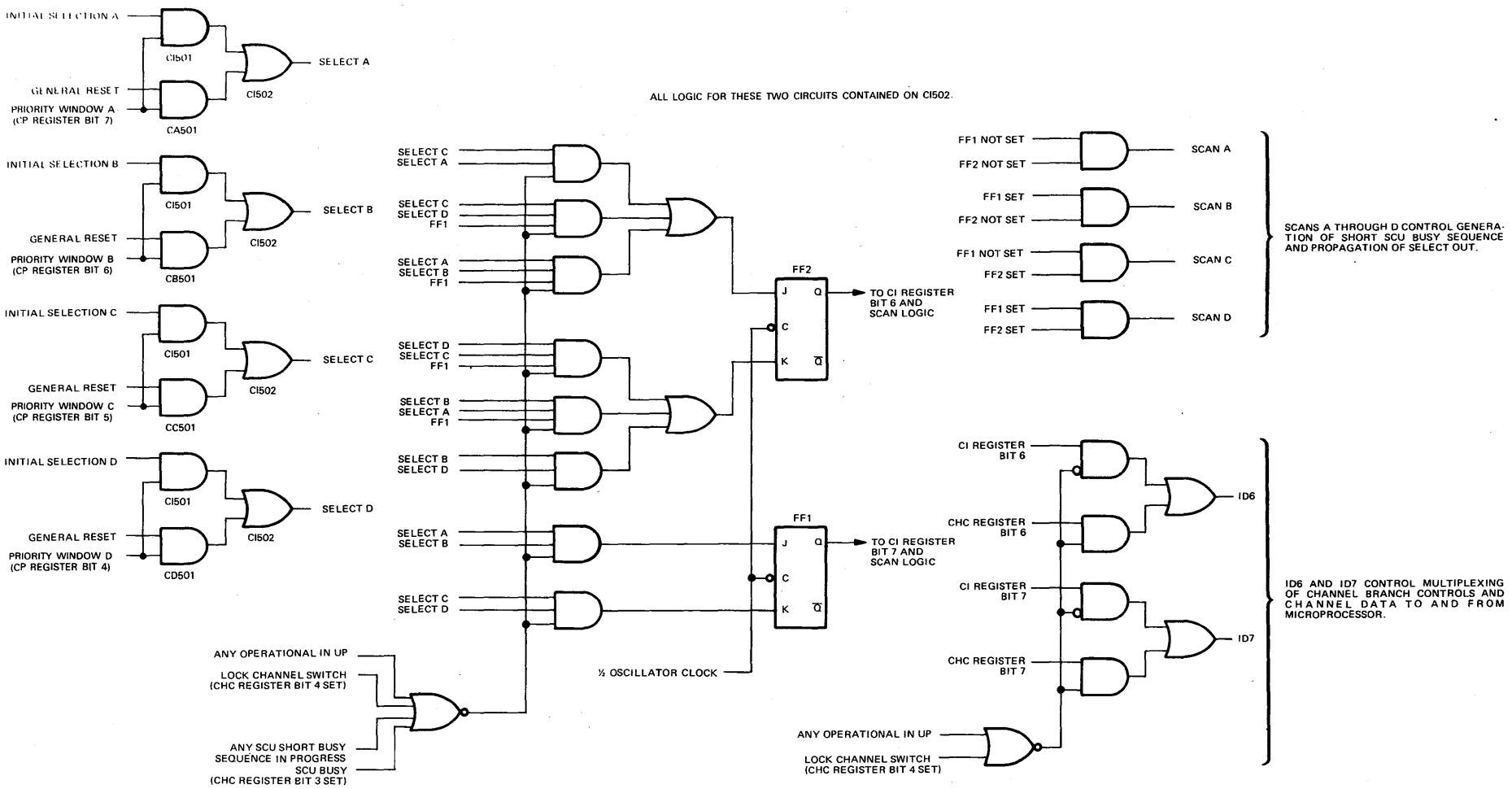


Figure 3-12. Multi-Channel Switching Logic

3.12 2860 SELECTOR CHANNEL ATTACHMENT FEATURE

3.12.1 Description

The 3672 2860 Attachment Feature provides the capability for the Memorex 3672 Storage Control Unit (SCU) to operate in a simulated block multiplex mode when connected to an IBM 2860 Selector Channel. The objective of simulating block multiplex operations requires that channel command chaining be broken to effect a channel disconnection (to provide channel and SCU availability) and that, upon the SCU signalling the channel with Retry status, the last command be retransmitted to the SCU to provide error recovery procedures. On actual block multiplex channels, Disconnected Command Chaining (DCC) and Command Retry (CR) are performed by interaction between the SCU and the channel without CPU interaction. However, the 2860 Selector Channel does not support DCC or CR, and if a SCU without the 3672 2860 Feature attempts either of these sequences on a 2860 channel the following occur, respectively:

a. The 2860 maintains connection to the SCU upon receipt of Channel End (CE) only status.

b. The 2860 inhibits chaining upon receipt of Retry status, that is, Status Modifier (SM), Channel End (CE), and Unit Check (UC). This causes the SCU to discontinue the retry sequence and indicates Equipment Check in a subsequent *Sense* command.

In order to simulate block multiplex operation, the software in the using System/360 must provide the necessary interaction with the SCU to perform DCC and CR as opposed to the channel/CU interaction that occurs on actual block multiplex channels. As an example, typical channel programs for the 3670/5 disc on block multiplex channels (IBM-OS release 21.6) are of the following form:

Seek
Set File Mask
Set Sector
Search
Write
Set Sector
Search
Read

To support the SCU on the 2860 Selector Channel, the using software system must include a new command, *Set RPS,* in every channel program for the 3670/5. This command causes the SCU to enter the "RPS Mode" and unique status is presented to force a channel disconnection or command retry sequence on the 2860 channel. A channel program for the 3670/5 then takes on the following form:

Set RPS
Seek
Set File Mask
Set Sector
Search
Write
Set Sector
Search
Read

The *Start I/O* instruction is executed by the CPU and receipt of Condition Code 0 in the Program Status Word (PSW) indicates that the channel is executing the channel program. The SCU executes the channel commands beginning with the *Set RPS* command. Upon receipt of the *Seek* command, the SCU presents disconnect status to the channel if mechanical motion is required. The channel disconnects and the SCU "marks" the seeking device as being in the DCC state, even though the channel program was broken when the channel did not indicate command chaining when it accepted the disconnect status. The SCU is now available for operations with other connected devices which are not busy or offline.

The SCU tests for the seek complete condition at the devices when it is in the idle state and initiates a selection to the channel when the seek complete condition is detected. The channel selects the SCU and stacks the seek complete status, Device End (DE). The SCU is available after DE is stacked for *Start I/O, Test I/O,* or *Halt I/O* on other devices. The SCU signals a request for reconnection with DE status, and the channel interrupts the CPU and generates a *Test I/O* to obtain device status from the SCU. DE status is placed in the Channel Status Word (CSW) and the CSW is stored in main storage during the I/O interrupt sequence.

When DE ending status to the *Test I/O* instruction is accepted,the channel does not indicate chaining. Instead, the SCU maintains a contingent connection and waits for the restart command. A selection of another device will not be accepted and UC will be presented in initial status.

The SCU reconnection on an actual block multiplex channel is similar to the above except that the channel indicates chaining to DE status and the chained reselection loop is entered to wait for the new command with the Suppress Out tag (chaining indication) active until the channel reselects the SCU for the next command. The Suppress Out tag will not be active on the 2860 selector channel because the channel processing an interrupt to the CPU rather than a channel hardware reconnection.

To effect reconnection, the using software system continues the original command chain with the *Set RPS* command prefixed to it. The *Start I/O* instruction required to continue the command chain then points to the following command sequence:

Set RPS
Set File Mask
Set Sector
Search
Write
Set Sector
Search
Read

The receipt of Condition Code 0 indicates that the channel program has been restarted. The channel program continues until the next disconnect occurs. For the example being analyzed, two more disconnect/reconnect sequences would take place (in the *Set Sector* commands) identical to that described above for the *Seek* command.

Up to four channels in any combination of block multiplexer and 2860 may be attached to the SCU. Support of the *Reserve* and *Release* commands associated with the SCU Multichannel Switch Feature is provided for the single channel and multichannel versions of the SCU when the 3672/2860 Attachment Feature is present.

When channel programs are not prefixed with the new *Set RPS* command on a 2860 channel, the SCU does not break the channel program for CE only disconnection and some types of CR disconnection; instead, channel programs are executed to the end of the chain. No seek or RPS overlap is available in this mode and the CR procedure for error recovery is not attempted.

3.12.2 Functional Characteristics

3.12.2.1 GENERAL

The functional characteristics of the SCU with the 2860 Attachment Feature are identical to the basic SCU characteristics except as follows:

1. The SCU will accept and execute the new *Set RPS* command on all 2860 interfaces.

2. If the *Set RPS* command has been issued to the SCU, disconnect status of CE, UC, and UE is presented on 2860 interfaces instead of CE only. In addition, the SCU will maintain DCC and CR on 2860 interfaces even though chaining is not indicated by the channel(s).

The following sections describe in detail these differences for 2860 interfaces only.

3.12.2.2 CHANNEL COMMANDS

CONTROL COMMANDS

SET RPS Figure 3-13

Command Code 2F (hex)

Changes the state of the SCU to RPS mode, and coordinates the simulated Disconnected Command Chaining and Command Retry between the SCU and the CPU

The *Set RPS* command is valid only on 2860 Selector channels attached to the SCU. If this command is received from a block multiplex channel attached to the SCU, Unit Check will be presented in initial status and Command Reject/Invalid Command will be indicated in a subsequent *Sense* command. If the *Set RPS* command is truncated, the SCU will give Unit Check and the subsequent sense will indicate Command Reject/CCW Count Less Than Required.

Execution of the *Set RPS* command causes a state code byte to be transferred to the SCU. Bits 6 and 7 of the state code define the operation to be performed by the command. Bits 0-5 must be zero.

Bits 6, 7	Operation
00	Not valid, ending status of CE, DE, UC (command reject), data value not as required
01	Set RPS mode
10	Reconnect after disconnection (CE only)
11	Reconnect after command retry disconnection

Upon receipt of the *Set RPS* command, the SCU checks the state of the 3670/75 and the state code received in the one-byte argument. When the state code does not correspond to the 3670/75 state, the SCU posts ending status of (CE, DE, UC) and a subsequent Sense will indicate Command Reject/Invalid Sequence.

3670 State	Definition
—	No such state
1	Initialize RPS mode
2	Disconnected command chaining
3	Command retry disconnected

When the 3670/75 is in state 2 or 3, the receipt of any command, except *Set RPS* or *Test I/O*, will cause UC in initial status and Command Reject/Invalid Sequence in a subsequent Sense.

DATA ADDRESS—Specifies main storage location of the state code byte

FLAGS—Used at discretion of the programmer

COUNT—One

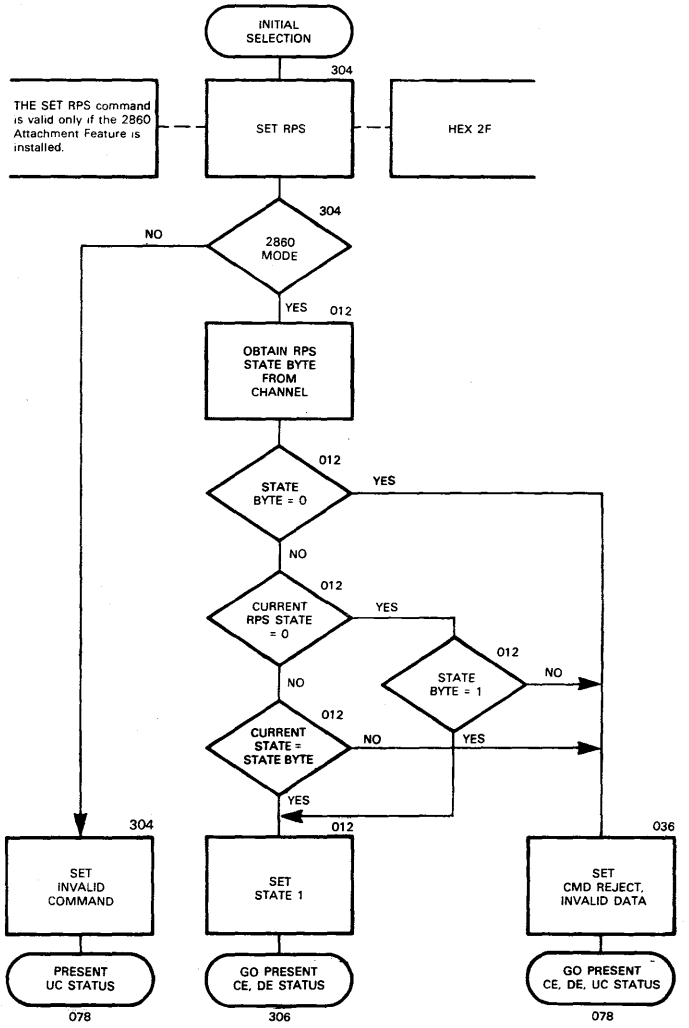


Figure 3-13. Set RPS Flow Diagram

Seek, Seek Cylinder, Recalibrate

These commands present disconnect ending status (CE, UC, UE) when access motion is required. The command operation is unchanged but the channel sequence is modified when the SCU is in RPS mode. The completion of the access motion results in an SCU-initiated selection and a subsequent CPU interrupt when DE is posted. The DE interrupt is the signal that the device is no longer busy and the channel program may be restarted. Reconnection of the channel program requires prefixing a *Set RPS* command with a state code byte of state 2 to the channel program. The restart address in the channel program is specified by the CCW address in the CSW which was stored when the disconnect interrupt occurred.

Set Sector

The *Set Sector* command presents disconnect ending status (CE, UC, UE) when rotational positioning is required. Rotational positioning is not required when the current disc's rotational position is equal to the target sector minus seven; in this case, the ending status is CE and DE.

The SCU will raise the Request In tag to initiate reconnection when the SCU is in the wait loop and the drive interrupt is active. The drive interrupt is active for one sector period (130 μsec) on each disc revolution. The sector corresponding to the drive interrupt period is the sector specified by the *Set Sector* command minus 7. The reconnect request on the 2860 is therefore 5 sectors earlier than the corresponding request for a block multiplex channel. This allows for an additional delay to accomplish the software analysis of the Device End status (I/O interrupt) and execution of the reconnect command *Set RPS*.

The SCU will raise the channel Request In tag to initiate reconnection when any one of the attached disc drives' rotational position corresponds to the *Set Sector* argument minus 7.

Diagnostic Load, Diagnostic Write

The operation of these commands is not affected by the RPS mode of the SCU. They do not present disconnect status.

Sense

All sense command operations are unchanged.

Read

All read command operations are unchanged.

Write

All write command operations are unchanged.

Search

All search command operations are unchanged.

3.12.2.3 DISCONNECTED COMMAND CHAINING

The 2860 Attachment Feature allows the software in the CPU to simulate disconnected command chaining (DCC). The SCU operation is modified when in RPS mode as follows.

Disconnect Status

Disconnect status is changed from CE only to (CE, UC, UE). (The 2860 maintains connection to the SCU when CE only status is presented.) When the new status is presented, the 2860 disconnects from the SCU and posts the status to the CPU and a I/O interrupt.

Chaining While Disconnected

The SCU receives the not chaining indication from the 2860 when disconnect status of (CE, UC, UE) is presented, but the "disconnected and chaining" housekeeping is performed to simulate the disconnected command chaining (DCC) state. This is done to maintain the current drive status data so that the command chaining may be restarted when the mechanical motion has completed. When reconnection is initiated and the *Start I/O* is issued, the SCU will not treat the command as the start of a new chain, but will treat it as if it were chained to the previous command (which caused the disconnect).

Reconnect Status

The DE status posted by the SCU (connected to a 2860) to reconnect to the channel is unchanged. The 2860 does not indicate chained reconnection by indicating chaining when status is presented and the SCU must maintain a contingent connection on the device until the CPU software provides the simulated reconnection.

Reconnect Sequence

Operations that occur during a reconnect sequence are as follows:

a. SCU completes a channel sequence and enters the wait loop (idle state).

b. SCU detects a drive interrupt.

c. SCU raises Request In.

d. 2860 polls (sequences channel tags by raising SELECT OUT while ADDRESS OUT is down).

e. SCU detects poll and performs selection sequence with channel using the device address of the lowest logical numbered drive in the DCC state.

f. SCU presents DE status to channel and channel indicates stack status to SCU.

g. SCU disconnects from the channel and maintains the DCC state of the drive. The stacked status state is not entered because the "Device End pending" from the drive interrupt is unchanged.

h. SCU enters the wait loop and raises Request In when a drive interrupt is present at one or more of the drives.

j. 2860 attempts to obtain an I/O interrupt for the drive whose status was stacked.

k. 2860 obtains interrupt from CPU.

l. 2860 performs *Test I/O* (channel hardware generated) on device whose status was previously stacked and completes the CPU interrupt sequence.

m. SCU posts DE status and resets interrupt at drive. Status is accepted but chaining is not indicated by channel.

n. SCU enters contingent connection wait loop. All Request In tags are down so no poll selection will occur. Channel-initiated selection of any other device will be rejected with UC status and any selection from the other channels will get a control unit busy.

p. 2860 receives *Start I/O* from CPU for the reconnect channel program on the device which presented the reconnect interrupt.

q. SCU processes the *Set RPS* command whose state code byte indicates state 2 (reconnect) and presents ending status of CE, DE.

r. 2860 accepts status and indicates chaining.

s. 2860 continues channel program.

t. SCU continues channel program.

3.12.2.4 Error Recovery

Error recovery procedures are changed to simulate command retry disconnect and reconnect sequences for 2860 channels only.

Command Retry Status

The command retry status of (SM, CE, UC) is unchanged. The 2860 does not support command retry and breaks the command chain when this status is presented by the SCU. The 2860 posts this status to the CPU with a I/O interrupt.

Chaining While Disconnected For Command Retry

The SCU receives the not chaining indication from the 2860 when the command retry disconnect status of (SM, CE, UC) is presented, but error recovery processing is performed with the chaining state simulated. When the reconnection is initiated and the *Set RPS* command is processed, the command chained to the *Set RPS* command will be treated as the retry command in the same manner as would occur for normal retry command processing on a block multiplex channel.

Command Retry Reconnect Status

The status of DE posted by the SCU to reconnect to the channel is unchanged. Although the 2860 does not indicate chained reconnection by indicating chaining when status is presented, the SCU maintains the pseudo-chaining state of the device until the CPU software provides the simulated reconnection.

Reconnect Sequence

The reconnect sequence for a command retry disconnect is similar to that of a normal disconnect. After the *Set RPS* command with state code byte indicating state 3 is processed, the error recovery sequence is continued as below:

a. DE status is accepted by the 2860

b. SCU enters a pseudo reconnect wait loop. This is equivalent to the UC contingent connection wait loop.

c. SCU receives the *Set RPS* command for command retry reconnect and presents CE, DE ending status.

d. SCU receives chaining indication from the 2860 and enters the chained reselection loop.

e. SCU receives the retry command and enters the command retry procedure to perform the retry of the command.

f. SCU continues the channel program.

g. 2860 continues the channel program.

3.12.2.5 NON-RPS MODE OPERATION

Block multiplex channels always operate in the "not RPS mode" and there is no effect upon these channels when the 2860 Attachment Feature is present. The attached 2860 channels are presented with different status for error recovery operations in the SCU to provide compatibility with software which cannot accept the command retry status on an I/O interrupt. The SCU does not attempt to provide disconnected command chaining for overlap of mechanical motion or for error recovery operations.

Disconnect Status

The SCU posts CE-only status for disconnect, but the 2860 does not disconnect. The result is that the channel remains connected to the SCU until mechanical motion is completed. No RPS or seek overlap is available except for the case where chaining is not indicated when CE status is posted. In this case, the 2860 presents CE to the CPU with an I/O interrupt. After the motion has completed, the SCU presents DE to the channel and another I/O interrupt is taken.

Command Retry Status

The SCU posts (CE, DE, UC) when command retry would normally be posted. A subsequent sense command to the SCU sends sense bytes 0, 1, and 2 of hex 04 00 00 (Overrun), to the host system, plus the format/message and data bytes which correspond to the specific error condition. The normal recovery for Overrun is to retry the entire channel program 10 times. Sense byte responses to specific error conditions are listed below:

a. Data Check Uncorrectable recovery causes Overrun and Format 4 to be posted in the sense data. The data check may be due to a missing Address Mark or a sync check.

b. Data Check Correctable recovery causes Overrun and Format 5 to be posted in the sense data.

c. Seek Check recovery causes an Overrun and Format 1, Message A to be posted in the sense data.

Alternate/defective track recovery is modified to bypass the presentation of the command retry disconnect status and the reconnect status. The recovery action takes place during the command execution with no indication to the channel.

Write offset recovery is not required as no offset for data checks occurs in the not-RPS mode and offset is reset when the seek at the start of a channel program is processed.

Index continue recovery is modified to bypass the presentation of command retry disconnect status and reconnect status. The recovery action takes place during the command execution with no indication to the channel.

Track padding disconnect for a format write command is modified to not issue command retry disconnect status and reconnect status. The padding takes place during the execution of the command with no indication to the channel.

Overrun error recovery causes Overrun and Format 0, Message 0 to be posted in the sense data.

3.12.3 I/O Programming

3.12.3.1 CHANNEL PROGRAMS

The SCU will execute channel programs which run on an IBM 3330 drive. Timing differences may occur due to the simulation of disconnected command chaining.

Command retry requires that the CPU software locate the CCW which contains the failing command, using the CCW address supplied by the CSW when the I/O interrupt occurs. The failing command is at the CSW-indicated address minus eight, except when data chaining. Data chaining may cause the indicated CCW to be a data address pointer rather than the command CCW.

3.12.3.2 STATUS CONDITION EXCEPTIONS

Status conditions added or changed as a result of the 2860 Attachment Feature are listed below.

Control Unit End (CUE)

a. CUE will be included with stacked status (except during reconnect sequences).

b. CUE will be included with DE for unchained *Seek* or *Set Sector* commands which previously presented CE only (when not in RPS mode only).

c. CUE will not be presented in conjunction with disconnect status (CE, UC, UE) or command retry disconnect status (SM, CE, UC).

Channel End (CE)

CE only status is presented to initiate a channel disconnect on block multiplex channels for *Seek* and *Set sector* commands which require mechanical motion. This status is also given on 2860 channels when the SCU is not in RPS mode, but no channel disconnect occurs and the channel remains connected to the SCU until the mechanical motion is completed and DE status presented. This 2860 selector channel sequence also occurs on a 370 channel when the channel program is executed using a shared unit control word (UCW) in the channel rather than the normal unshared UCW.

When the SCU is in RPS mode, the CE-only status is modified to force a disconnect. The modified status is CE, UC, UE.

Command retry status (SM, CE, UC) is a special case of disconnect status.

Unit Check (UC)

UC status is presented when the SCU cannot continue a command chain due to an error condition. The data in a subsequent sense indicates the nature of the error. The exception to this usage occurs for disconnect status (CE, UC, UE) on a 2860 channel when the SCU is in RPS mode and for command retry status (SM, CE, UC). In these cases, no sense data is required to define the error.

When disconnect status (CE, UC, UE) or command retry status (SM, CE, UC) is presented, the SCU will present DE in a subsequent status to signal the request for reconnection.

Unit Exception (UE)

Disconnect status on a 2860 channel is (CE, UC, UE) when the SCU is operating in the RPS mode. The normal disconnect status of CE-only is modified to force disconnection on a selector channel.

3.12.3.3 I/O INTERRUPTS

The channel performs an I/O interrupt to send device status to the CPU when a channel program ends or as a result of a channel poll sequence which obtained status from a device. If the CPU has channel interrupt disabled, the channel enters the "interrupt in channel" state for ending status obtained when a channel program ends. The channel CSW must be cleared before other operations can be performed by the channel. This can be done by a *Test I/O* instruction to the device address for which the CSW applies, or by enabling the channel interrupt.

An "interrupt in channel" condition does not occur for an interrupt which is pending for status which is obtained during a channel poll sequence. The channel will interrupt the CPU and form the CSW by reselecting the device and accepting the device status using a channel-generated *Test I/O* if channel interrupt is enabled. If *Start I/O* or *Halt I/O* is issued before the interrupt is taken, however, the status will remain at the device and the *Start I/O* or *Halt I/O* will be performed.

SECTION 4. MICROPROGRAM

4.1 MICROINSTRUCTION

4.1.1 General

The microprogram controls the operation of the 3672-based subsystem and is permanently stored on a flexible disc. Immediately after power has been applied to the subsystem, the microprogram is read from the flexible file, and stored in writable control storage (WCS).

The following paragraph contains a description and format of a microinstruction, and a description of decoding and translation of a field within a microinstruction. Also presented is a layout and definition of a microinstruction word in the microprogram, and a microprogram subroutine example. In addition, a chart depicting the various routines within the microprogram is included.

4.1.2 Microinstruction Word Description

The microprogram controls the hardware of the 3672 SCU through the sequencing of microinstructions. These microinstructions are divided into fields. Each field controls a specific function in the hardware. In the SCU, the data for these microinstructions is stored in WCS.

There are four microinstruction word formats. Each format, and the fields contained therein, activates specific signal lines which control the logic circuits.

Because of the large number of controls needed within the SCU, the 34 bits of the microinstruction word are translated into 57 usable bits. This translation concept is used to allow a smaller memory word size.

4.1.3 Microinstruction Format

4.1.3.1 FORMAT 0 MICROINSTRUCTIONS

This format provides the ability to transfer data into two registers with one instruction. It is shown in Figure 4-1, part a.

One of the registers is specified by the D field. It is loaded with the result of an ALU operation on the data from the register specified by the A field and the constant Z (immediate operand Z). The ALU function is specified by the F field D = F(A,Z) .

The other register is specified by the C field. It is loaded with the constant (immediate operand) contained in the X field (C = X).

The loading of the destination registers occurs at register time of the present cycle. Specifying R03 as a register designator causes operation upon register R0, R1, R2, and R3 in parallel. When the destination register of one bus is designated R03 and that of another bus is R0, R1, R2, or R3, the bus operation on the single register will have its data loaded and the other three registers will be loaded from the other bus.

4.1.3.2 FORMAT 1 MICROINSTRUCTIONS

This format, shown in Figure 4-1, part b, provides the ability to set data into a register and also branch on a machine condition.

The BA field is used to form the low-order 6 bits of the WCS address register, if the condition specified by the branch condition field BC is satisfied. The high-order bits of the address register are unchanged when a branch is made, which limits Format 1 branches to the present 64-word storage block (64-word page). The word at the next sequential storage address is executed if the branch condition is not satisfied. Since the instruction read and execute operations are overlapped, the branch decision is made on the value of the machine conditions at the end of the previous microinstruction cycle.

The register specified by the D field is loaded at register time with the result of an ALU operation on the data from the register specified by the A field and the constant Z (immediate operand Z). The ALU function is specified by the F field D = F(A,Z) .

4.1.3.3 FORMAT 2 MICROINSTRUCTIONS

This format (Figure 4-1, part c) provides the ability to perform a control operation, set data into a register, and also branch on a machine condition. The BA and BA2 fields are used to form the low-order 9 bits of the WCS address register if the condition specified by the branch condition field BC is satisfied. The high-order bits of the address register are unchanged when a branch is taken, which limits Format 2 branches to the present 512-word storage block (512-word page). The word at the next sequential storage address is executed if the branch condition is not satisfied. The branch decision is made on the value of the machine conditions at the end of the previous microinstruction.

The constant Z (immediate operand Z) is loaded at register time into the register specified by the D field (D = Z).

The control field CL is decoded into one of 32 control functions. Some control functions set and reset latches in the hardware, and others provide register-to-register data transfers. The destination registers are loaded at register time. Specifying R0, R1, R2 or R3 in the D field when the register-to-register transfer specifies register R03 will cause the D Bus data to be used in place of the register-to-register data for the D field register specified and the other three registers of the group R03 (R0, R1, R2, R3) will receive their data from the source register of the specified register-to-register transfer.

4.1.3.4 FORMAT 3 MICROINSTRUCTIONS

This format (Figure 4-1, part d) provides the ability to perform an ALU operation on two registers, set data into a third register, and also branch on a machine condition.

The BA and BA3 fields are used to form the low-order 12 bits of the WCS address register if the condition specified by the branch condition field BC is satisfied. The word at the next sequential storage address is executed if the branch condition is not satisfied. The branch decision is made on the value of the machine conditions at the end of the previous Microinstruction cycle.

The register specified by the D field is loaded with the result of an ALU operation on the data from the register specified by the A field and the data from the register specified by the B field. The ALU function is specified by the F field [D = F(A,B)].

4.1.4 Microinstruction Field Definitions

Table 4-1, when used with the word formats explained on the previous page, shows which controls will be decoded, dependent upon the contents of the microprogram word fields.

4.1.5 Field Translation

The control of the hardware by the microinstruction fields is accomplished through groups of bits which control specific hardware functions. Because the SCU requires 57 hardware controls, but only 34 bits are available in the microinstruction, the microinstruction word is translated (expanded). During each machine cycle, the microinstruction format code is used to translate the microprogram fields (explicitly and implicitly) into the proper hardware controls.

The relationship of the hardware controls described below to the microprogram instruction, is shown in Figure 4-2.

RA Control (5-Bit Group)

This group specifies the register to be multiplexed to the A Bus.

RB Control (3-Bit Group)

This group specifies the register to be multiplexed to the B Bus.

RC Control (3-Bit Group)

This group specifies function to be performed by the ALU.

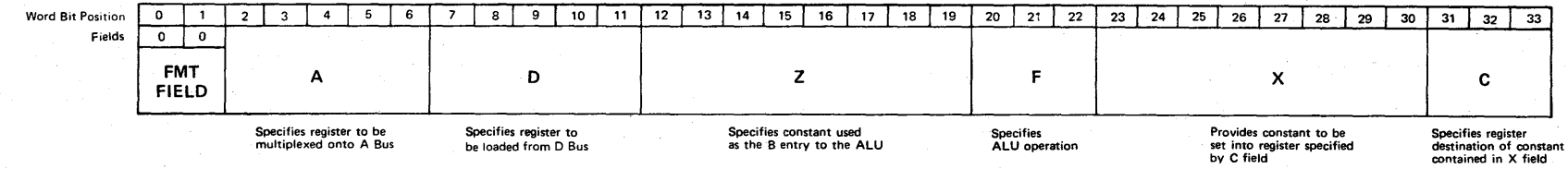
RD Control (5-Bit Group)

This group is decoded to send a set pulse to the proper machine register to transfer the D Bus contents into that particular register.

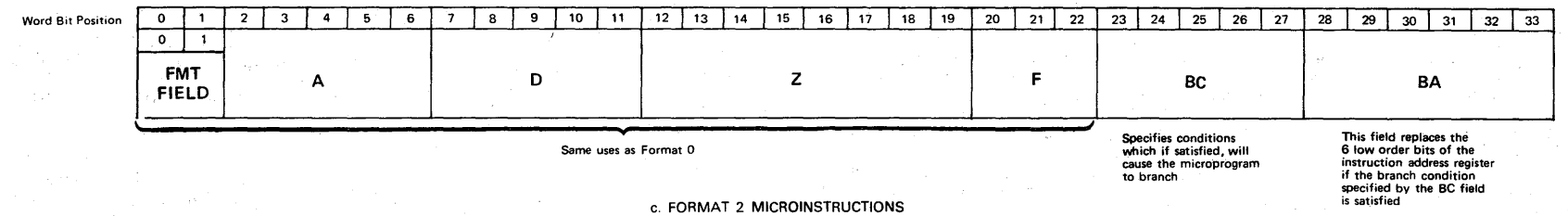
RF Control (4-Bit Group)

This group specifies the register to be multiplexed to the T Bus.

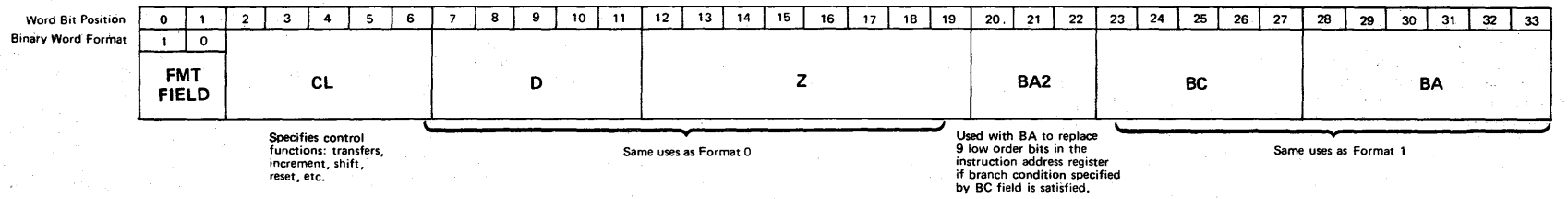
a. FORMAT 0 MICROINSTRUCTIONS



b. FORMAT 1 MICROINSTRUCTIONS



c. FORMAT 2 MICROINSTRUCTIONS



d. FORMAT 3 MICROINSTRUCTIONS

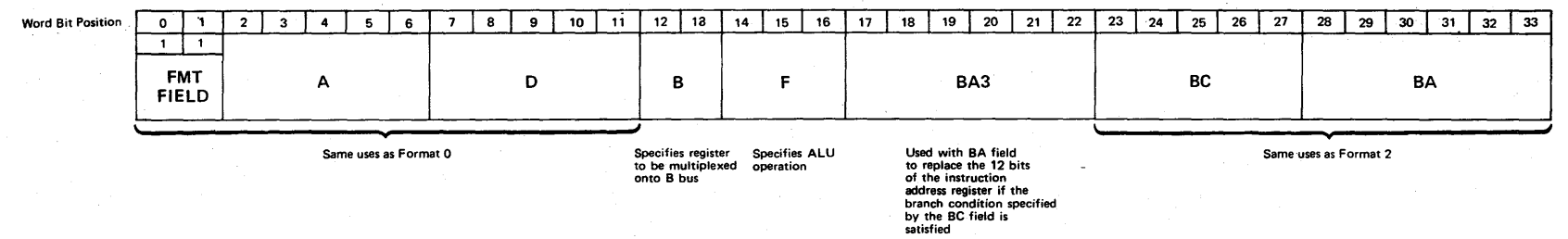


Figure 4-2. Microinstruction Formats

TABLE 4-1 MICROINSTRUCTION FIELD DEFINITIONS

FIELD CONTENTS (HEX)	FIELD NAME								
	A	B	F	D	C	BC		CL	
00	0	G0	+	D	G03	B (BR0)	Branch if BR0 is set.	C (G03 = R03)	Transfer contents of R03 into G03.
01	CI	G1	+C	CT	R1	B (BR1)	Branch if BR1 is set.	C (BUF = R03.INC)	Write data from R03 into buffer at address specified by BAR and increment the BAR.
02	CHF	G2	+1	CHC	R2	B (BR2)	Branch if BR2 is set.	C (CC = R23)	Transfer contents of R2 and R3 into CC register. Reset CC = 0 latch.
03	BO	G3	.	BI	CO	B (BR3)	Branch if BR3 is set.	C (DTC = R23)	Transfer contents of R2 and R3 into DTC (Data Transfer Counter).
04	G5		V	G5	DO	B (BR4)	Branch if BR4 is set.	C (SR = R23)	Transfer contents of R23 into SR Register.
05	G6		*	G6	CT	B (BR5)	Branch if BR5 is set.	C (BUF = R02.LD)	Write data from R03 into buffer at address specified by BAR prior to its being loaded from G3. High BAR is not modified.
06	SW		K	MD2	PS	B (BR6)	Branch if BR6 is set.	C (BUF = R03)	Write data from R03 into buffer at address specified by BAR. BAR is not modified.
07	CS		L		CNT	B (BR7)	Branch if BR7 is set.	C(NONE)	No Operation.
08	CCK		↑	MCM		B (D = 0)	Branch if D Bus = zero.	C (R03 = BUF)	Transfer BUF (buffer data register) to R03 Buffer storage is not addressed or read.
09	SFD		SEE ALU	FDC		-B (D = 0)	No branch if D Bus = zero.	C (R03 = BUF.INC)	Transfer BUF to R03 and increment BAR.
0A	ECR		FUNCTION	$$$		B (D = FF)	Branch if D Bus = FF (hex).	C (R03 = BUF.LD)	Transfer BUF to R03 and then transfer G3 to BAR. High BAR is not modified.
0B	RWC		CODES	RWC		-B (D = FF)	No branch if D bus = FF (hex).	C (RESET.INDX)	Reset INT Register bit 4 (index latch).
0C	DI			SP		B (CARRY)	Branch if Carry latch is set.	C (CH.FREEZE)	Complete channel transfer sequence in progress, then inhibit further transfers.
0D	CT			FDD		-B (CARRY)	No branch if Carry latch is set.	C (RESET.K)	Reset compare latches
0E	CO			CO		-B (COMPARE.EQ)	Branch if A ≠ B latch is set.	C (RESET.CKS)	Reset Check 1 and Check 2 latches.
0F	DO			DO		B (COMPARE.HI)	Branch if A > B latch is set.	C (READ.CS)	Load word from WCS into data register in FD interface
10	R0	G4 00		R0		B (INT)	Branch on selected interrupt true. Controlled by IMK Register.	C (RESET.DBR)	Reset DBR latch
11	R1	R1 00		R1		-B (INT)	Branch on selected interrupt false. Controlled by IMK Register.	C (D.EVEN)	Generate even parity for the D bus for the present microinstruction.
12	R2	RW 00		R2		B (CNT = 0)	Branch if counter carry latch is set.	C (LD.SR.*+1)	Load next sequential microinstruction address into SR.
13	R3	R3 00		R3		-B (CNT = 0)	Branch if counter carry latch is false.	C (BAR + 1)	Increment BAR and begin buffer read cycle.
14	INT	INT 00		IMK		B (LD.SR. * +1) LOW	Load next sequential microinstruction address into SR. Branch to a microinstruction in the lower portion of WCS. The branch address is determined from the branch address fields of the microinstruction.	C (BAR = G3)	Transfer G3 to BAR (8 bits).
15	G4	R0 00		G4				C (CLOCK.DTC)	Check the DTC. The DTC counts up if RWC bit 5 is on, and down if RWC bit 5 is off.
16	G7	G7 00		G7		B (DBR)	Branch if DBR (data buffer ready) is true.	C (BAR = G23)	Transfer G2 and G3 to BAR (9 bits)
17	PS	D 00		PS		NOT USED		C (G23 = BAR)	Transfer BAR to G2 and G3. Registers G0 and G1 are unchanged.
18	G0	CHF 00	SEL BY PS BITS ON BR BUS	G0		B (ILACT)	Branch if inline is active.		
19	G2			G2		-B (DBR + INT)	No branch if DBR or the selected interrupt is true.		NOT USED
1A				CNT		B (CBR)	No branch if channel buffer ready (CBR)	C (R03 = G03)	Transfer G03 to R03
1B	TR			R03		B (Q.EMPTY)	Branch when channel buffers A and B are available for new data		NOT USED
1C	G1	DI 00		G1		B (NO.BRANCH)	No branch	C (R23 = SR)	Transfer SR to R2 and R3. Register R0 and R1 are unchanged.
1D	G3			G3		B (LD.SR. * +1) HI	Load next sequential microinstruction address into SR. Branch to a microinstruction in the upper portion of WCS. The branch address is determined from the branch address fields of the microinstruction.	C (R23 = CC)	Transfer CC (channel counter) to R2 and R3
1E	CNT			TR		B (UNCOND)	Unconditional branch to microinstruction address developed by branch address fields of this microinstruction	C (WRITE.CS)	Transfer contents of data register in FD interface to WCS at address specified by the address register
1F	CP	NBR 00		CP		B (D.BUS)	Branch to microinstruction address developed by logical OR of D bus with 8 low order bits developed by branch address fields of this microinstruction.		NOT USED
						B (ADR = SR)	Branch to microinstruction address obtained from SR Register.	C (SPECIAL)	Expands number of control statements by using three bits on D bus. Following functions are performed via this control statement:

D BUS 567
- 000 - Not used.
- 001 - Not used.
- 010 - ILACT (set ILACT).
- 011 - ILACT.OFF (reset ILACT).
- 100 - Not used.
- 101 - Propagate Select Out
- 110 - Sets 13th address bit for Read/Write Control Storage.
- 111 - Resets 13th address bit for Read/Write Control Storage.

ALU FUNCTION CODES:

+ = ADD
+C = ADD WITH CARRY DETERMINED BY CARRY LATCH
+1 = ADD WITH CARRY
. = LOGICAL AND
V = LOGICAL OR
* = EXCLUSIVE OR
K = COMPARE
L = SHIFT "A" LEFT

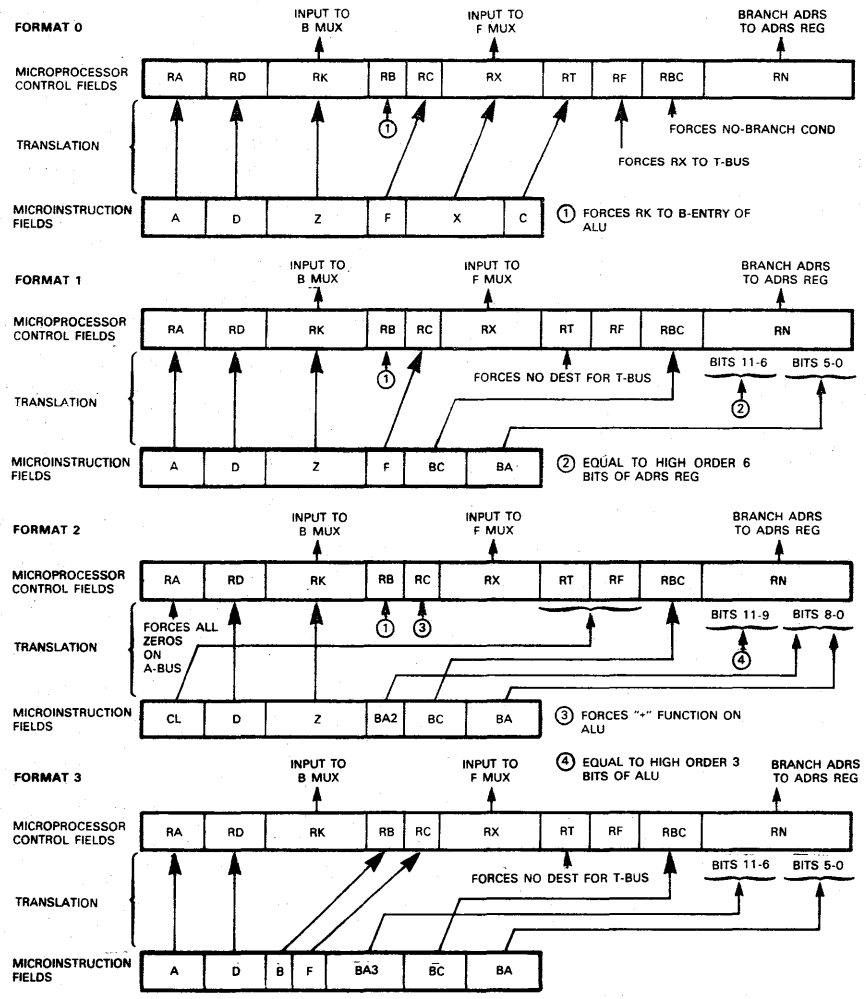


Figure 4-2. Microinstruction Expansion

RT Control (4-Bit Group)

This group is decoded to send a set pulse to the proper machine register to transfer the T Bus contents into that particular register.

RBC Control (5-Bit Group)

This group specifies which branch condition will control the next address executed by the Control Unit.

RN Control (12-Bit Group)

This group forms the branch address for the WCS. It is developed by the BA, BA2, and BA3 fields; and certain bits of the Address Register.

RK Control (8-Bit Group)

This group consists of the microinstruction Z field gated by the microinstruction format code. RK is gated to the B Bus with certain formats.

RX Control (8-Bit Group)

This group consists of the X field of the microinstruction format code. RX is gated to the T Bus by the format 0 microinstruction.

4.2 MICROPROGRAM

4.2.1 Microprogram Block Format

The microinstructions listed on a microprogram page are formatted into blocks. The format of each block depends on the format of the microinstruction. The different formats (0, 1, 2 and 3) plus the subroutine block format are shown in Figure 4-3. This figure depicts the difference between each format, and defines the entries contained in each microinstruction block.

4.2.2 Example Microprogram

The example shown in Figure 4-4 is the actual Read Home Address routine portion of the microprogram. The example depicts a sequence of microinstructions and an explanation of the entries within a microinstruction.

4.2.3 Microprogram Routine Organization

The microprogram is divided into a number of routines. Each routine performs a basic function, but may accomplish several different operations, depending on how it is used.

The microprogram routine organization is shown in Figure 4-5. A brief explanation of each block is also contained in the following paragraphs.

Reset Procedure

The machine hardware is initially reset to the power-on reset state. The microprogram is then loaded (IMPL) starting at the IMPL address, or restarted (Selective Reset and System Reset) at location 000. IMPL starts a new microprogram with the Writable Control Storage (WCS) and buffer storage in the reset state. General Reset only resets the devices allocated to the channel initiating the General Reset command. Selective Reset only resets the particular operation in progess.

Basic Wait Loop

The Basic Wait Loop responds to any of the following sequences:

1. Initial Channel Selection
2. Control Unit Initial Selection
3. Starting the diagnostic monitor when sequence 1 and 2 have not been in progress for 150 milliseconds.

File Status Analysis

File Status Analysis sets up the current device parameters in scratch pages 7-B, when selection occurs. File Status Analysis also receives the command from the channel. The parameters and the channel registers are examined to determine what status should be presented in ending status. If no ending status exists, the decode of the Channel Command is entered.

Command Decode

Command Decode can be entered from File Status Analysis, or from Chained End Procedure. When entering from Chained End Procedure, the time through command decode is taken up in the gaps. Command Decode checks the validity of the command, checks that the proper sequencing has occurred prior to execution of the command, and sets up the command indicators required to process the command.

Initial Status Presentation

Initial Status Presentation presents zero status, for the SCU commands, to the channel. If orientation exists with the track, the routine will wait for Command Out signal to drop, raise Status In, and then exit. If orientation does not exist, the sequence is completed by waiting for Service Out in response to Status In.

Ending Status Presentation

Ending Status Presentation presents ending status to the channel. The section then waits for channel acceptance, and determines if the channel wants to continue the chain, discontinue the chain, or block multiplex.

Chained End Procedure

Chained End Procedure is used when the channel chains one command to another. This section insures that the device interrupt is received and reset when the channel is acting like a selector channel, and that the new command is received from the channel.

Unchained End Procedure

Unchained End Procedure is used when the channel or SCU ends a command chain, and the Controller and drive must be deselected. When Unit Check is presented in the preceding status, the section must assemble the sense bytes in Buffer Storage location 120 (hex). The drive parameters are also stored in the buffer. If In lines are active, the program loops back to the Diagnostic Monitor section, otherwise the program goes back to the Basic Wait Loop section.

Sense Commands

Four Sense Commands accept sense information from various locations in the Buffer Storage and send the information to the channel for analysis. A hardware channel transfer is used to switch DATA IN and SERVICE IN when transferring the sense bytes.

Control Commands

The Control Command section is not directly involved with transferrring data from the 3670 file. This section moves the access position, reads and sets the sector values, receives the file mask from the channel, and spaces across bad count fields.

Read Commands

The Read Command section transfers data directly from the Controller, using hardware channel transfer. Correctable errors occurring in the HA, Count, or Key Fields are corrected in Buffer Storage. The corrected data is transferred from the Buffer Storage using hardware channel transfer.

Search Commands

Search Commands section accepts bytes from the Controller and from the channel. The section compares the bytes fo: n equal, high, or high equal state. The section then transfers the status of the comparison to the channel.

Write Commands

The Write Command section accepts bytes from the channel, and using hardware channel transfer, sends the bytes to the Controller to be written. The ECC is appended to the data train to be written by the Controller.

Diagnostic Commands

The Diagnostic Commands section controls the in-line microdiagnostics from the CPU. The section also enables the 367X Fault Isolation Detection System to evaluate the test results.

Error Processing

The Error Processing section handles all retryable errors and error analysis. The retryable errors are: (1) ECC errors, (2) Seek Incomplete, (3) Defective and Alternate track errors, (4) Retry pading, (5) Sync check, (6) AM check, (7) Index Continue, (8) Offset before Write, (9) Command Overrun, and (10) Data Overrun.

Diagnostic Monitor

The utilization of all microdiagnostic tests is accomplished through the resident SCU Diagnostic Monitor section of the Microprogram. The Diagnostic Monitor interfaces with the functional Microprogram and with the microdiagnostic required to perform a specific test.

Microdiagnostic

The Microdiagnostic routines are loaded into the CPU from the Flexible Disc, and then transferred to the WCS. Execution of a microdiagnostic is initiated by a Diagnostic Write Command. The microdiagnostics test the SCU, Controller, and the Disc Drive sequences; and monitors any errors which may occur. The errors are stored in the sense bytes for analysis by the Error Recovery Procedure (EREP) or the Field Engineer (FE).

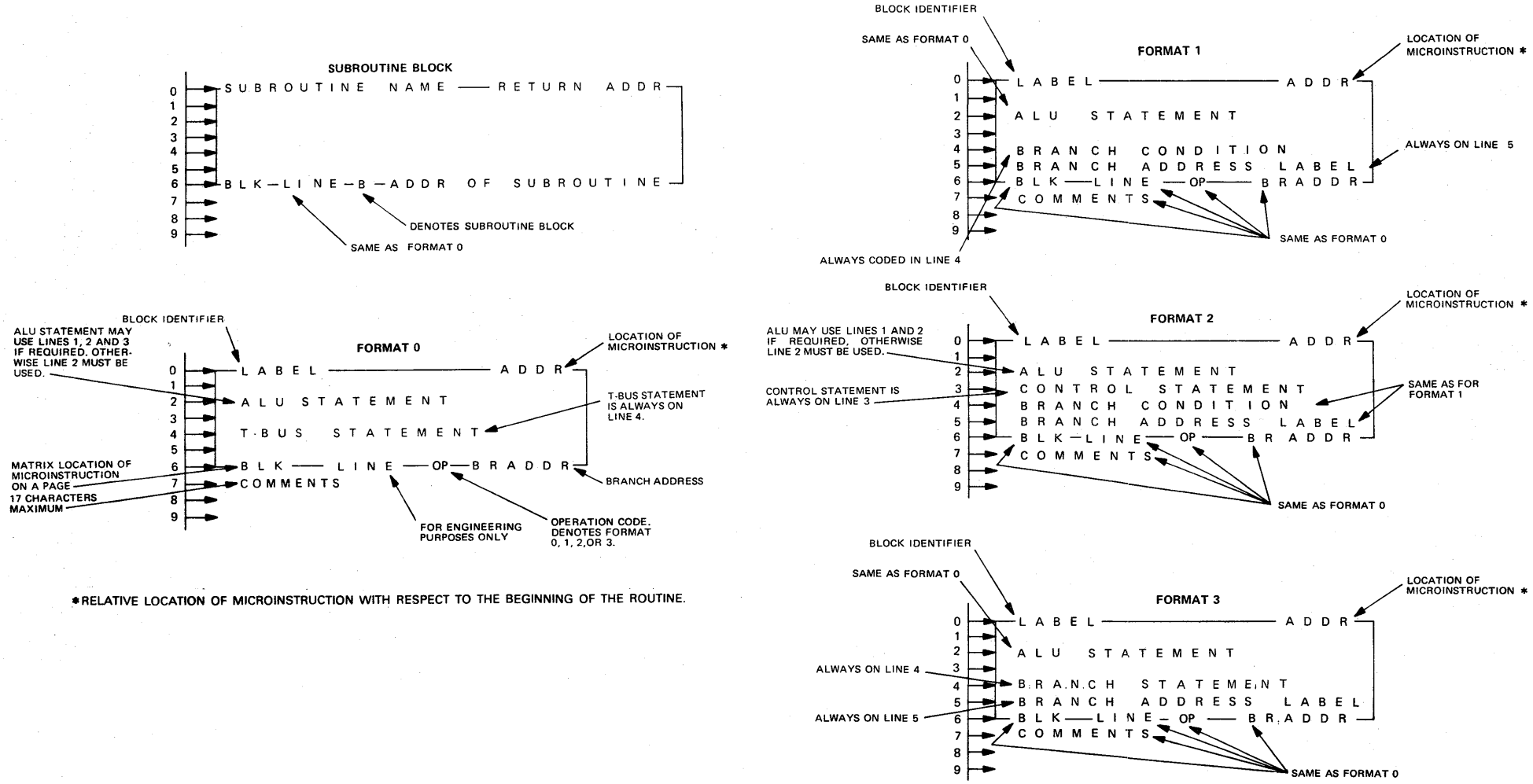


Figure 4-3. Microprogram Block Formats

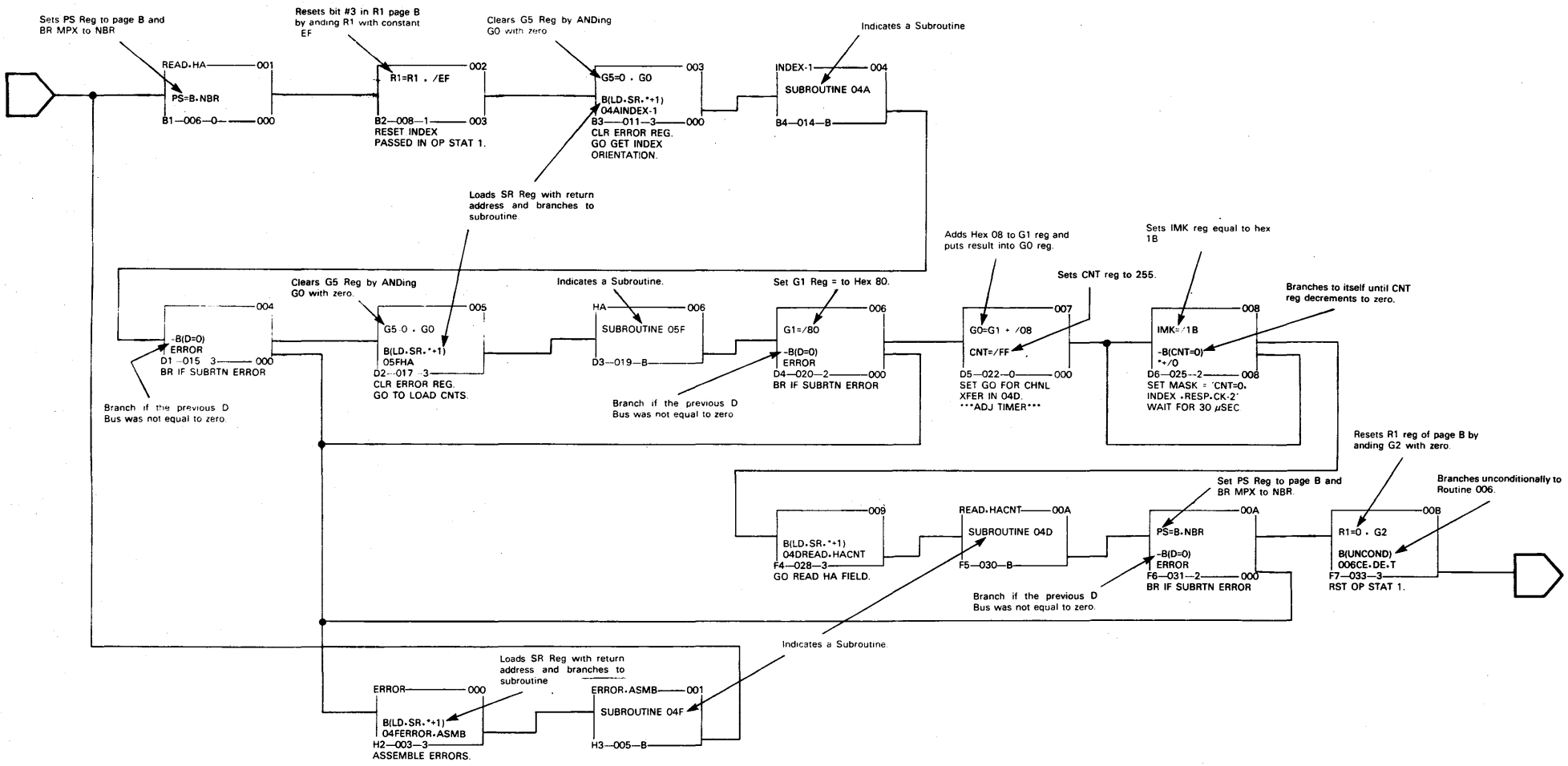


Figure 4-4. Microprogram Format Example

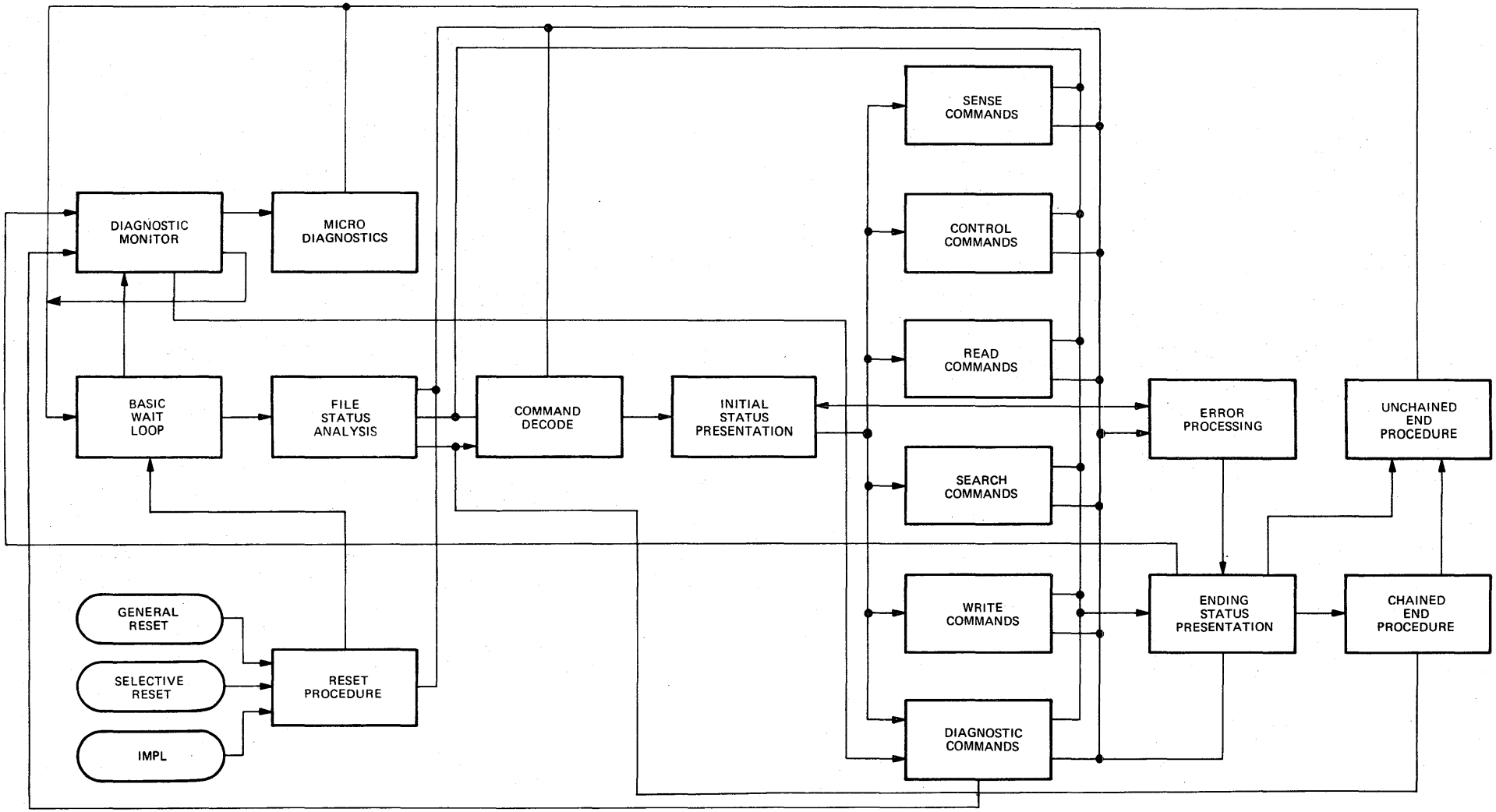


Figure 4-5. Microprogram Routine Organization

SECTION 5. FLEXIBLE DISC FILE

5.1 GENERAL DESCRIPTION

The Memorex 650 Flexible Disc File (Figure 5-1) is a compact, direct access, removable disc unit intended to simplify the distribution, processing, and storage of digital information. The 650 stores the subsystem microprogram and the microdiagnostic program library.

The 650 is composed of a drive mechanism, read head, head actuator, and associated electronics. Connections are made to the SCU Microprocessor Flexible Disc Interface by signal and power cables, which supply addressing, function requests, data formatting, and power.

The FD/III Flexible Disc Cartridge (Figure 5-2) used in the 650 is a flat disc composed of a Mylar ® substrate coated with a magnetic oxide. For protective purposes during handling, operation, and storage, the disc is permanently encased in a flexible plastic jacket, 8 inches square by 1/16 inch thick.

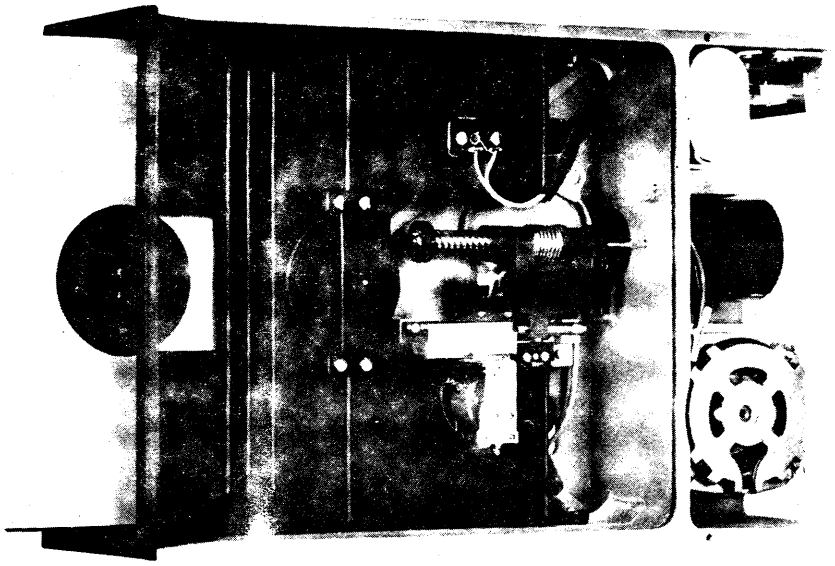


Figure 1. Memorex 650 Flexible Disc File

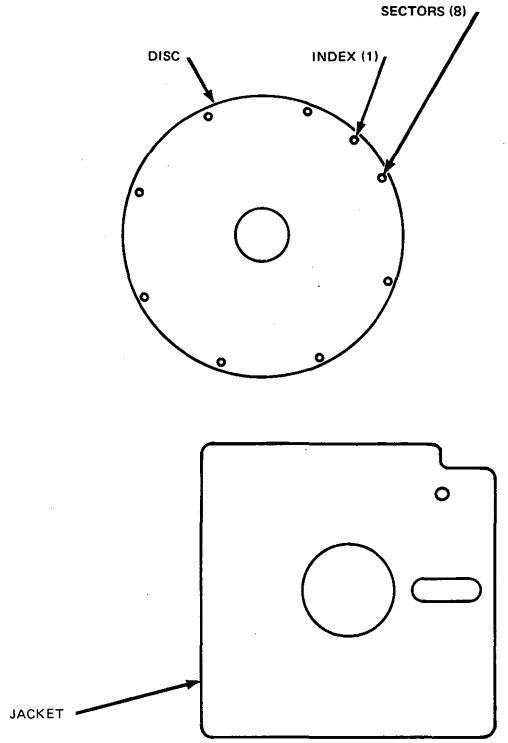


Figure 5-2. Flexible Disc Cartridge

5.2 SPECIFICATIONS

5.2.1 Machine Characteristics

Machine characteristics include data retrieval time, disc characteristics, and data recording format. These items are described as follows.

Data Retrieval Times

• Rotational Speed	375 rpm
• Single Track Access Time	20 msec
• Data Transfer Rate	200 kilobits/sec

Disc Characteristics

• Number of Tracks	50
• Recording Density	2400 bits per inch (inside track)
• Record Length Sectorized (8 per track)	3.5 kilobits
• Record Length Indexed (1 per track)	30 kilobits
• Disc Capacity Sectorized	1.4 megabits
• Disc Capacity Indexed	1.5 megabits

Data Recording Format

• Recording Mode	Frequency modulation
• Sectors per Track	8
• Index per Track	1

5.2.2 Operating Capabilities

Operation of the 650 in the SCU is fully automated, requiring no operator intervention during normal operation. All maintenance and diagnostic procedures are predicated on the proper application of power from the SCU flexible disc interface.

All 650's are capable of being operated with 50 or 60 Hz power. For 60 Hz, the disc drive motor pulley is mounted on the shaft with the large end toward the motor. The belt is driven by the small diameter pulley. For 50 Hz, the pulley is reversed on the shaft and the belt is driven by the large diameter pulley.

5.2.3 Power Requirements

All power required by the 650 is supplied by the SCU flexible disc interface.

AC Power

110 ±10% volts
50/60 Hz, single phase
0.75 amperes

DC Power

+5 ±0.10 volts @ 0.6 amps, 50 mV ripple
-15 ±0.30 volts @ 0.12 amps, 50 mV ripple
*-12 ±0.25 volts @ 0.12 amps, 50 mV ripple
+24 ±1 volts @ 2.0 amps, 100 mV ripple

* May be used in lieu of -15 volts with no modifications to the file or cables.

5.3 SPECIAL PRECAUTIONS

The 650 can be damaged by improper servicing, handling, or operating techniques. The following procedures should be observed to properly operate and maintain the 650.

5.3.1 Cartridge Loading

The cartridge consists of the flexible disc encased in a plastic jacket. Wipe cushions are bonded to the inside of the jacket. The disc is housed and rotates between these cushions during normal operation. Figure 5-3 shows how the cartridge is loaded in the cartridge guide. To load the cartridge open the door, insert the cartridge into the cartridge guide, and close the door.

5.3.2 Disc Interchangeability

To ensure interchangeability, store discs in a location that is within ±5°F of the using system ambient temperature and within ±10% of the using system humidity. Discs stored outside the recommended ranges must be placed in the using system environment at least 20 minutes before use.

5.3.3 Physical Damage

When removed from the 650, the disc cartridge is stored in a plastic-coated paper envelope (Part No. 204268). To protect the cartridge, the same care and handling procedures specified for computer magnetic tapes apply. Additional precautionary procedures are as follows:

1. Return the cartridge to its storage envelope whenever it is removed from file.
2. Store cartridges vertically.
3. Keep cartridges away from magnetic fields and from ferromagnetic materials which might become magnetized. Strong magnetic fields greater that 50 oersteds can distort recorded data on the disc.
4. Replace storage envelopes when they become worn, cracked, or distorted. Envelopes are designed to protect the disc.
5. Do not write on the plastic cartridge. Writing pressure may damage the disc.
6. Do not smoke while handling cartridges. Heat and contamination from a carelessly dropped ash can damage the disc.
7. Do not expose cartridges to heat or sunlight. The Read/Write head on the 650 cannot properly track a warped disc.
8. Do not touch or attempt to clean the disc surface. Abrasions may cause loss of stored data.

5.3.4 Safety

AC and dc power are controlled by the SCU. Before working on the file, verify that all power is removed from the 650.

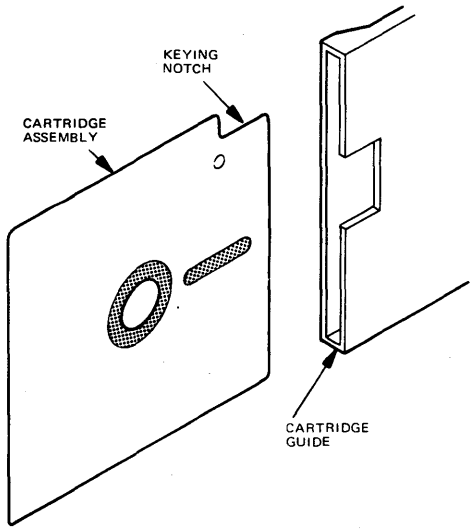


Figure 5-3. Cartridge Loading

5.4 PRINCIPLES OF OPERATION

5.4.1 General Operation

The 650 consists of control and read electronics, disc drive motor, read head, head position actuator, and removable disc (Figure 5-4). The primary functions performed by the 650 are:

- Receive and generate control signals
- Generate status signals
- Access the appropriate track
- Read data upon command

The Control and Read Logic interface directly with the flexible disc interface electronics in the SCU microprocessor, described in paragraph 6.3.6. The Head Positioning Actuator positions the read head at the desired track on the disc and the Head Load Actuator loads the disc against the head, so that data may then be read from the disc.

The electronics are packaged on one PCB. The PCB contains:

- Sector/Index Detector Circuits
- Track Position Actuator Driver
- Head Load Actuator Driver
- Read Amplifier and Transition Detector
- Data/Clock Separation Circuits

An electrical stepping motor (Head Position Actuator) and lead screw position the read head. Step in and step out pulses from the flexible disc interface cause the stepping motor to rotate the lead screw clockwise or counterclockwise in 15-degree increments. A 15-degree rotation of the lead screw moves the read head one track position.

The Disc Drive Motor rotates the spindle at 375 rpm through a belt-drive system. Either 50 or 60 Hz power is accommodated by means of a stepped pulley. A registration hub, centered on the face of the spindle, positions the disc. A clamp (that moves in conjunction with the door) fixes the disc to the registration hub. The disc is held vertically, and rotates in a plastic jacket, which protects and cleans the recording surface during operation.

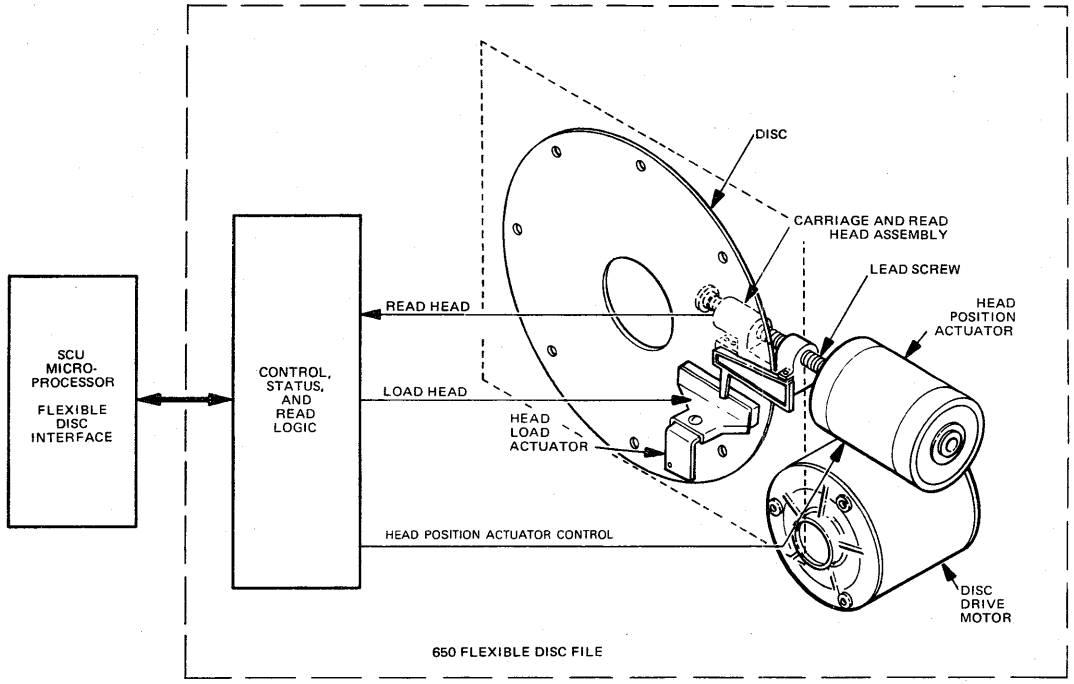


Figure 5-4. Flexible Disc File/Microprocessor Interface

5.4.2 Functional Assemblies

The 650 is divided into three functional logic blocks and electromechanical transducer as shown in Figure 5-5.

The read head is mounted on a carriage that is moved by the lead screw. Head loading is achieved when the disc is lightly loaded against the rigidly mounted head by moving a load pad against the disc with a solenoid actuated bail. Head to disc compliance is achieved by restraining the disc between the head the load pad. Figure 5-6 schematically illustrates this action.

The disc, 7 5 inches in diameter, has eight holes spaced around the periphery for sector definition There is also one additional hole for indexing. The disc rotates inside the plastic envelope during normal operation Figure 5-7 illustrates the disc and disc envelope

The Light Emitting Diode (LED) and Detector generate disc index and sector pulses used by the Microprocessor to format and orient data written on the disc. As shown in Figure 5-7, eight holes (0.10 inches diameter) divide the disc into equal sectors. The ninth hole (same diameter), spaced midway between two sector holes.

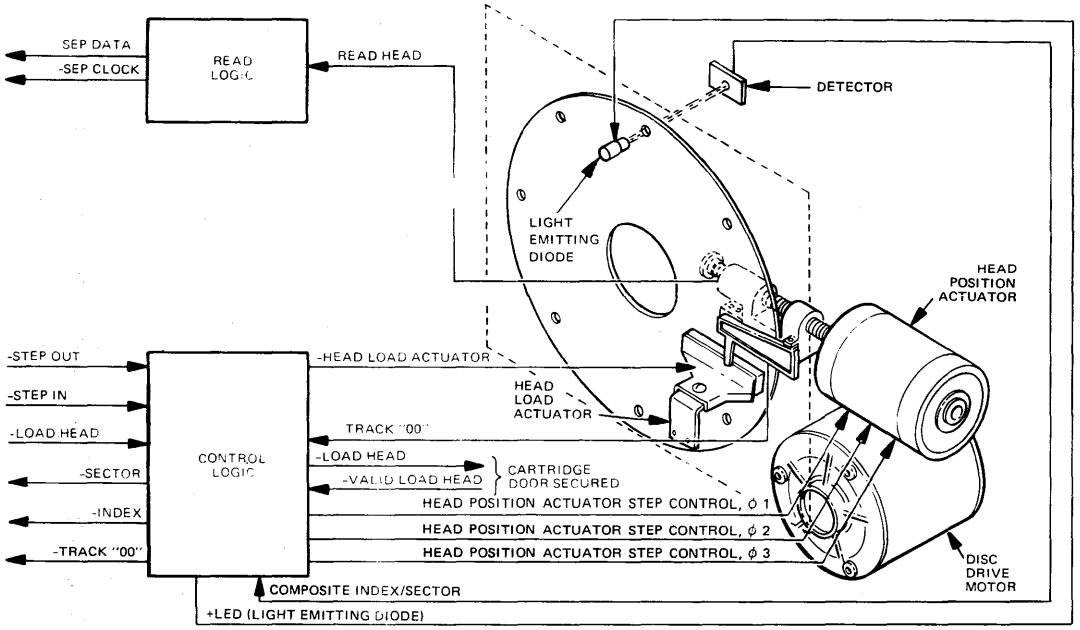


Figure 5-5. Flexible Disc File Functional Block Diagram

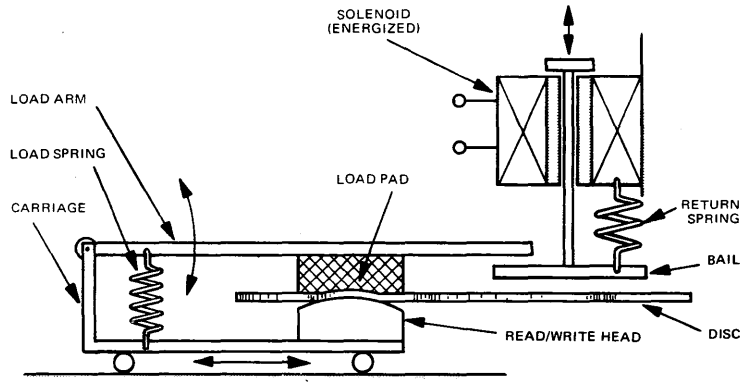


Figure 5-6. Head Loading Mechanical Assembly

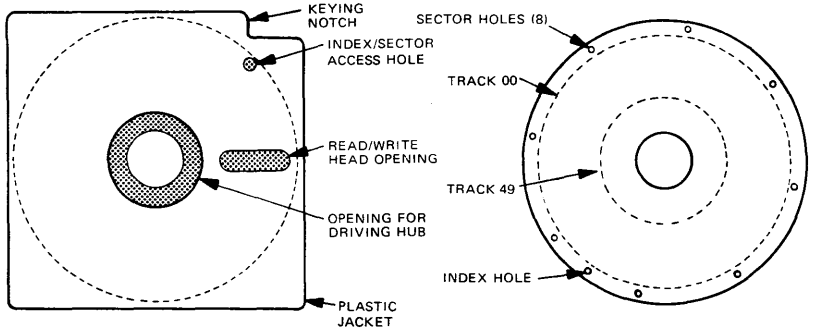


Figure 5-7. Disc Cartridge and Disc Configuration

indicates one disc revolution. The LED and Detector (photo transistor) are placed on opposite sides of the disc. As the disc revolves, the holes pass between the LED and Detector, exposing the Detector to infrared light, which turns on the the Detector. The Detector output is shaped by a threshold detector, and an output pulse is obtained. Output is approximately at 2.5 volts with a transition to 5.0 volts for the pulse.

5.4.3 Logic And Analog Functional Descriptions

These paragraphs discuss each logic block and signal name. The descriptions are divided into two logic blocks: Control and Read. All input interface functions are generated by the SCU. All output interface functions are generated by the 650.

5.4.3.1 CONTROL AND STATUS LOGIC

The basic functions of the Control and Status Logic are to place the read head on the proper track, hold the disc against the head for read operations, and indicate disc rotational position. The SCU interface functions are as follows:

STEP OUT	Increments the Head Position Actuator. Each pulse moves the head outward one track away from the center of the disc.
STEP IN	Increments the Head Position Actuator. Each pulse moves the head one track inward toward the center of the disc.
LOAD HEAD	Loads the disc to the read head.
SECTOR and INDEX	Provides disc sector and index position information. Each signal is a separate output. These pulses are generated from holes located around the periphery of the disc and are used for the accessing of data on the disc.
TRACK 00	Indicates when the read head is located at Track 00.

Transducer signals between the 650 Control Logic and electromechanical assemblies are as follows:

HEAD LOAD	Energizes the solenoid as shown in the head-loading block diagram. It is a function of LOAD HEAD.
TRACK 00	Switch closure indicating that the read head is located at Track 00.
VALID LOAD HEAD	Switch closure interlock indicating that the cartridge door is secured. If this condition is not satisfied, the read head cannot be loaded.
LED	Provides power to the Light Emitting diode (LED).
COMPOSITE INDEX/SECTOR	Detector signal input to the Control Logic providing disc sector and index information. Within the Control Logic, the INDEX and SECTOR signals are separated, and are transmitted by separate interface lines to the SCU.

5.4.3.2 READ LOGIC

The basic function of the Read Logic is to receive analog signals from the disc and convert this composite signal into separate clock and data pulses. Data and clock come from the Read Logic block when the disc is loaded onto the head. SCU interface functions are as follows:

SEP DATA	Digital data bits read from the disc.
SEP CLOCK	Digital clock bits read from the disc.

Transducer signal communication between the 650 Read Logic and the read head is as follows:

READ HEAD	An analog representation of the recorded information consisting of composite data and clock signals, differentiated by alternating polarity. Those signals are amplified, differentiated, limited, and then shaped. The shaped data is applied to a data separator circuit which separates the data bits from the clock bits.

5.4.4 Functional Operation

The 650 functional sequences are divided into the following three phases of operation: initialization, track access, and read. The initialization phase is used when the power is turned on. During normal operation, the head is positioned at the addressed track during the track access phase, and the read phase is performed.

5.4.4.1 INITIALIZATION PHASE

Whenever the 3671 Storage Control Unit applies power to the 650, a Power On sequence occurs automatically to prepare the disc file for operation. The events and timing for this sequence are shown in Figure 5-8.

The SCU flexible disc interface applies primary and secondary power to the 650. After a two-second delay, -STEP OUT is pulsed until the head is positioned at Track 00. This operation is performed to ensure that the head is properly oriented before a read operation begins. When the head is positioned at Track 00, the TRACK 00 signal becomes true. It changes to false when the head leaves that track.

The -LOAD HEAD signal can be applied any time after the power has been turned on. When the -LOAD HEAD line is false, the head is unloaded from the disc. The disc must be loaded on the head before a read operation can begin.

5.4.4.2 TRACK ACCESS PHASE

The -STEP OUT and -STEP IN functions are used for positioning the head to the desired track. The only restriction placed on these signals is that each pulse must be spaced by at least 20 msec. The -STEP IN pulse width is the same as the pulse width for -STEP OUT.

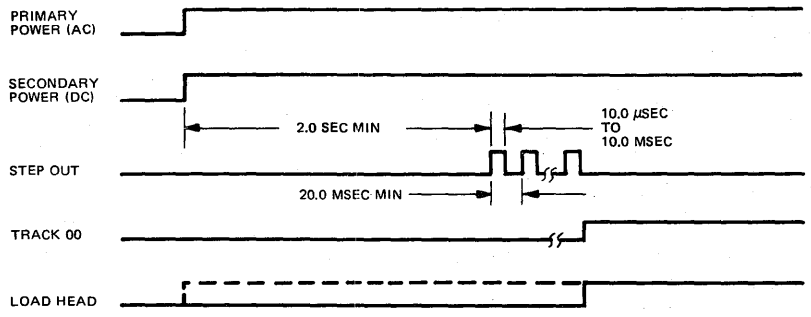


Figure 5-8. Initialization Sequence

5.4.4.3 READ PHASE

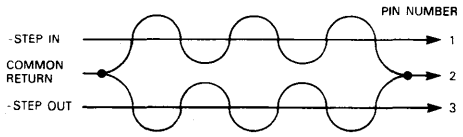
A read operation comprises a track address and index-/sector seek sequence followed by a read data sequence. Waveforms for a read sequence are shown in Figure 5-9.

The Read logic is shown in the Logic Manual on page FD310. The Read head reads the combined clock and data pulses recorded on the disc. The read logic amplifies these signals and separates them into two outputs: separated clock signals (SEP CLOCK) and separated data signals (SEP DATA). These waveforms are shown in Figure 5-10.

5.4.5 Interface Signal Description

The interface signals can be divided into three categories: control, data, and power. The following paragraphs describe the signals with relationship to connector, polarity, logic level, and pulse width. The wiring diagram of the 650 with signal names, pin numbers, and connectors is shown in Figure 5-11

The 650 requires only two cables, power and control/data. The twisted-pair lines are physically four wires, and are wired as illustrated below.



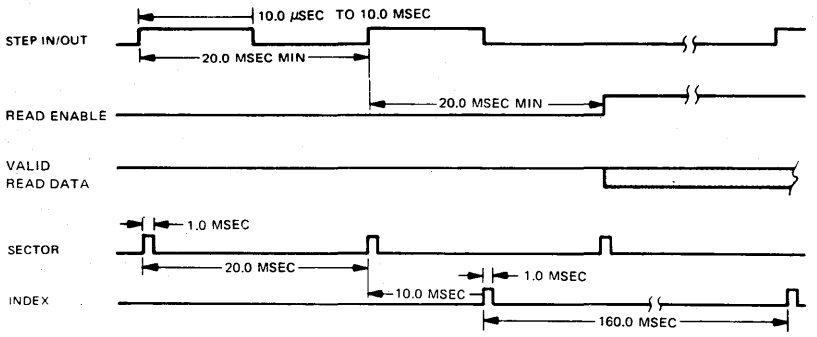


Figure 5-9. Read Sequence

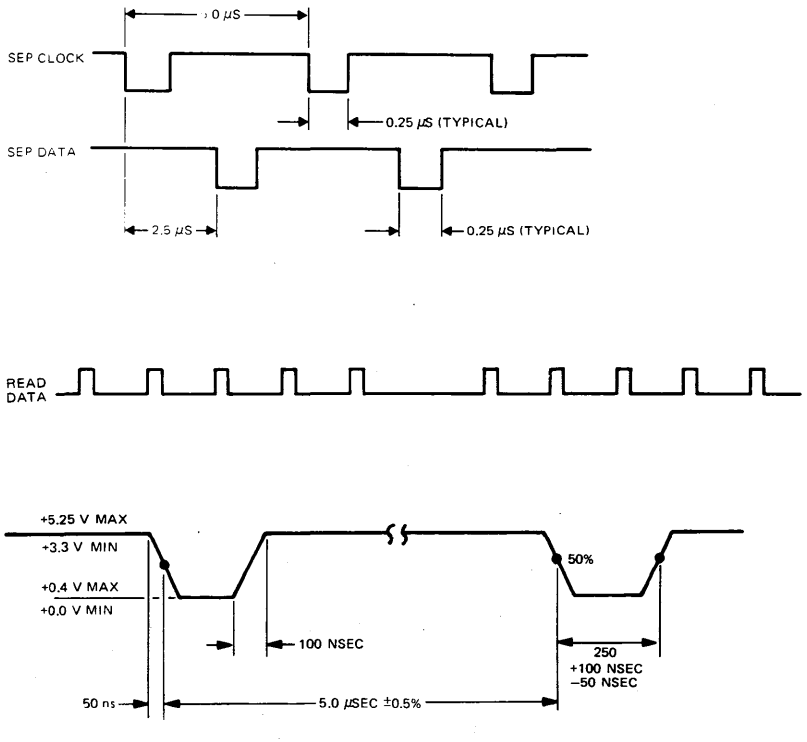


Figure 5-10. Read Data Waveforms

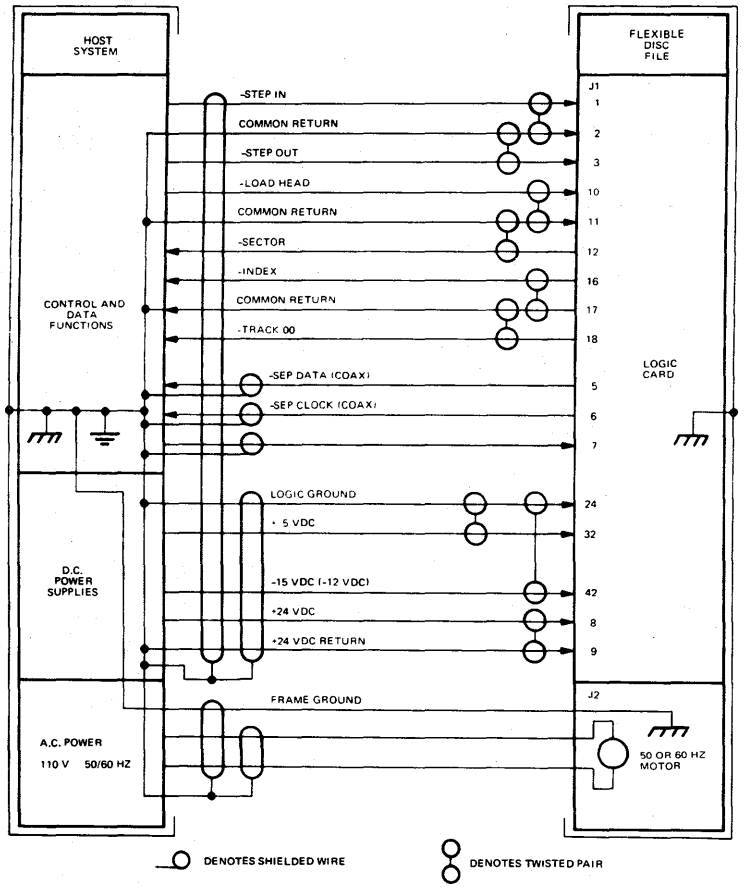


Figure 5-11. Flexible Disc File Interface Diagram

5.4.5.1 CONTROL

The interface control signals are divided into two types: input and output. The input signals are provided by the SCU flexible disc interface, and output signals are from the 650. Paragraph 5.4.3.1 previously described these signals. Logic levels vary, depending on the particular signal requirements. Signal levels are indicated on the logic diagrams in the Logic Manual. Negative logic, however, is used throughout the system. Logical 1 is true, and is a low level signal, indicated by a minus sign. Logical 0 is false, and is a high level signal, indicated by a plus sign.

Input Signals

Table 5-1 lists characteristics of the input signals.

Output Signals

Table 5-2 lists characteristics of the output signals.

TABLE 5-1. INPUT CONTROL SIGNALS

INPUT SIGNAL DESIGNATION	CONNECTOR AND PIN	ACTIVATION POLARITY	PULSE WIDTH	COMMENTS
-STEP IN	J1-1	Negative	10 μsec to 10 msec	Track Positioning
COMMON RETURN	J1-2	Negative Return		
-STEP OUT	J1-3	Negative	10 μsec to 10 msec	
-LOAD HEAD	J1-10	Negative	Level	Enables head load solenoid
COMMON RETURN	J1-11	Negative Return		

TABLE 5-2. OUTPUT CONTROL SIGNALS

OUTPUT SIGNAL DESIGNATION	CONNECTOR AND PIN	ACTIVATION POLARITY	PULSE WIDTH	COMMENTS
-SECTOR	J1-12	Negative	1.0 msec	Indicates location on disc
COMMON RETURN	J1-11	Negative Return		
-INDEX	J1-16	Negative	1.0 msec	Indicates location on disc
COMMON RETURN	J1-17	Negative Return		
-TRACK 00	J1-18	Negative	Level	Indicates when head is positioned on Track 00
COMMON RETURN	J1-17	Negative Return		

5.4.5.2 DATA

Data input and output signals characteristics are listed in Table 5-3.

5.4.5.3 POWER

Table 5-4 lists the power requirements

TABLE 5-3. DATA SIGNALS

DATA SIGNAL DESIGNATION	CONNECTOR AND PIN	ACTIVATION POLARITY	PULSE WIDTH	COMMENTS
-SEP DATA	J1-5	Negative	0.25 μsec	Output data from disc
-SEP CLOCK	J1-6	Negative	0.25 μsec	Output clock from disc

TABLE 5-4. POWER REQUIREMENTS

POWER SIGNAL DESIGNATION	CONNECTOR AND PIN	ACTIVATION POLARITY	DRIVE CHARACTERISTIC	PULSE WIDTH	COMMENTS
LOGIC GROUND	J1-24	Logic ground	Logic ground	Logic ground	DC power supply ground
+5VDC	J1-32	Positive	+5 ± 0.10VDC @ 0.6A 50 mv ripple	Power level	Logic Power supply
-15VDC	J1-42	Negative	-15 ± 0.30VDC @ 0.12A 50 mv ripple	Power level	DC power supply for read/write amplifiers
(-12VDC)*			-12 ± 0.25VDC @ 0.12A 50 mv ripple		-12VDC can be used in lieu of -15VDC
+24VDC	J1-8	Positive	+22 ± 1VDC @ 2.0A 100 mv ripple	Power level	DC power supply for head positioning motor and head load solenoid
+24VDC RETURN	J1-9	DC Power ground	DC power ground	DC power ground	+24VDC power ground
110 VAC 50/60Hz	J2 Three terminal socket	Line AC	110 ± 10% VAC @ 0.75A 50/60 ± 0.5Hz single phase	Line AC	Must be provided from a branch circuit protected at no more than 20 amperes.
FRAME GROUND	J2 Center Socket	Frame ground	Frame ground	Frame ground	Center socket of 3-wire AC socket

5.5 CONTROL CIRCUITRY DESCRIPTION

5.5.1 Functional Description

The electrical and electronic circuitry consists of control and status circuits composed primarily of solid-state integrated circuit components mounted on a single printed circuit board (PCB). The block diagram of Figure 5-12 shows the primary motor, actuators, switches, circuit components and circuit proups, and the associated control and status signals. The circuitry involved in the generation and transfer of the control and status signals is shown on the schematic and logic diagrams in the Logic Manual.

5.5.2 Printed Circuit Board

The PCB (Figure 5-13) is mounted at the rear of the 650 enclosure. It is secured to the baseplate by four screws, one in each corner. A connector plug on one side connects with connector PC1. The PCB drawing shows the component side of the board, and indicates the locations of the test points. Refer to the Logic Manual, pages FD300 through FD320, for test point circuit locations and normal waveforms and test values.

5.5.3 Harness Assembly

The harness assembly comprises connector block J1, PC board connector housing PC1, and the interconnecting cable consisting of the primary 650 input/output leads.

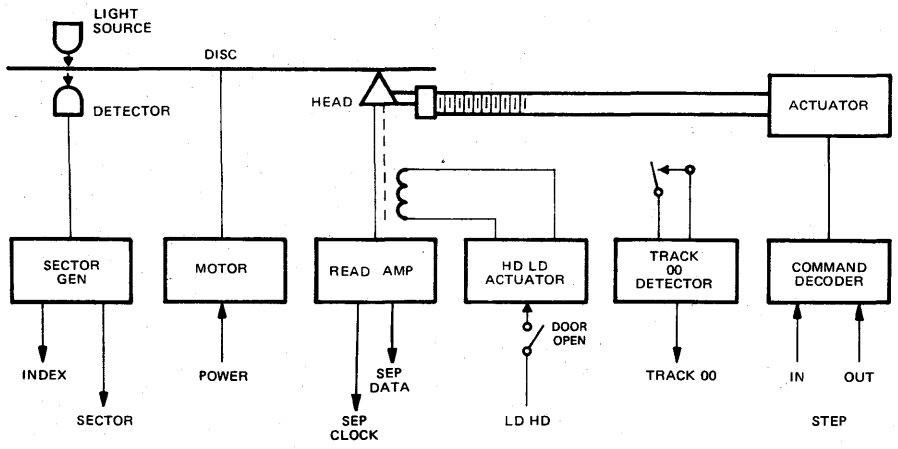


Figure 5-12. Control Circuitry Block Diagram

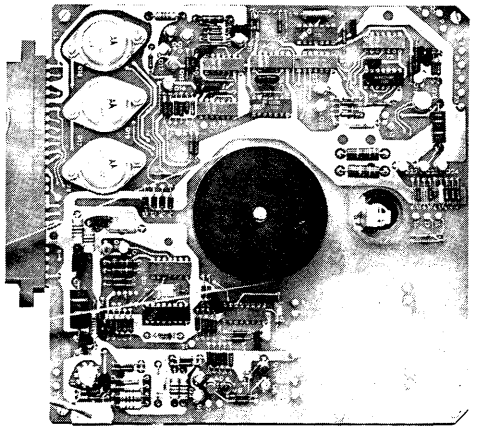


Figure 5-13. Flexible Disc File PCB

SECTION 6. STORAGE CONTROL UNIT DESCRIPTION

6.1 GENERAL DESCRIPTION

The 3672 Storage Control Unit (SCU) interfaces between the using IBM System, the 3673 Controller, and the Disc Drive Modules. The SCU contains the hardware used to control operation of the 3672-based Subsystem. The SCU is organized into four major hardware areas:

- Channel Interface — Communications link between the SCU and the using IBM system.
- Microprocessor — Contains the microprogram, ALU, special purpose registers, error checking circuits, Flexible Disc Interface, and data busses needed to control the Subsystem.
- Controller Interface— Communications link between the SCU, the Controller, and the DDM's.
- FE Interface — Contains the logic required to manually operate and control the FE panel for diagnostic control and maintenance.

The relationship of these areas to each other is shown in Figure 6-1.

Operation of the SCU is directly controlled by a microprogram stored in the Writable Control Storage (WCS) portion of the Microprocessor.

The Flexible Disc File provides permanent storage of the microprogram as well as offline and inline diagnostic routines. During an initial Power On sequence of the Subsystem, the microprogram is read from the Flexible Disc and stored in the WCS. The Flexible Disc is automatically powered down after the microprogram is read, and remains off until a microprogram reload is required or a diagnostic routine is requested. At this time, the SCU Microprogram is in a "wait loop," until commands selecting the SCU are received via the users channel. The commands are received by the Channel Interface portion of the SCU, and passed on to the microprogram for interpretation.

The microprogram generates control signals for the SCU during command execution. Microprogram subroutines and branching conditions allow the SCU to perform multiple operations, such as reading from one disc file while completing a Seek operation on another disc file. The control signals generated by the microprogram cause the Controller Interface to send orders to the DDM's, via the controller, enabling data to be stored or retrieved.

Controller and disc drive status is monitored by the microprogram for examination by the using channel, and stored in the Buffer Storage section of the Microprocessor. Data integrity is constantly checked by the SCU through the use of parity checking and Error Correction Codes.

The FE Interface section works in conjunction with the FE Panel to provide a means for the FE to manually control the Subsystem for maintenance and troubleshooting. The capability for running inline and offline diagnostic routines is also contained on the FE Panel.

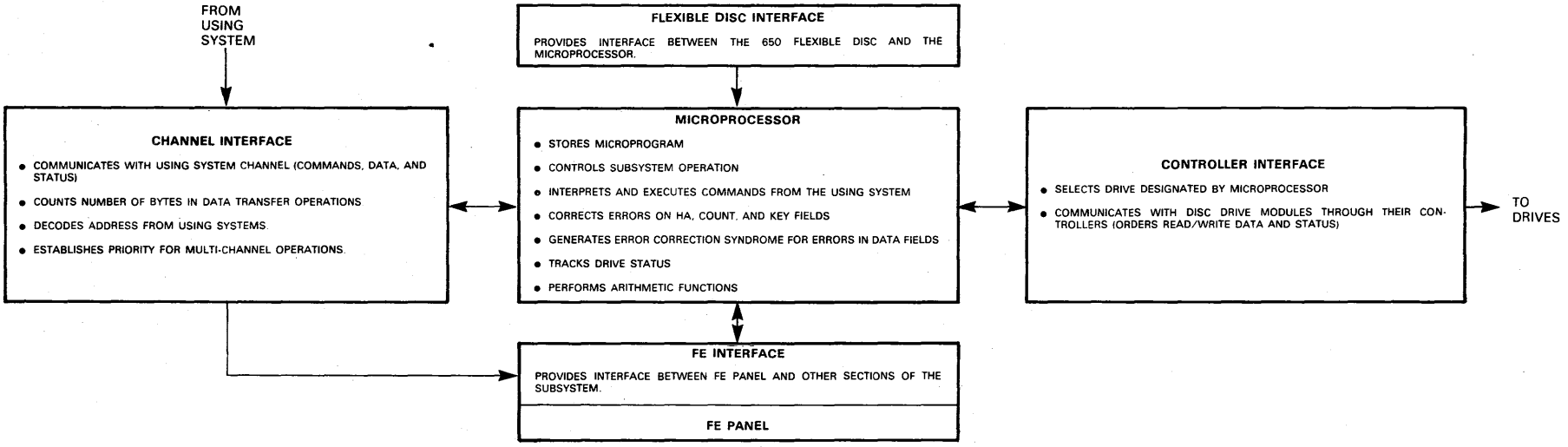


Figure 6-1. 3672 SCU Block Diagram

6.2 CHANNEL INTERFACE DESCRIPTION

6.2.1 General

The SCU Channel Interface is the asynchronous link between the IBM Channel and the SCU Microprocessor. The Channel Interface hardware consists of Bus In lines and registers, Bus Out lines and registers, address compare logic, select logic, priority logic, tag lines and registers, data transfer logic, control registers, and the multiplexers necessary to handle data to and from multiple channels. Refer to the block diagram in Figure 6-2.

NOTE

Refer to Section 3 of this manual for descriptions and definitions of interface lines, commands, and selection sequence.

6.2.2 Logic Description

6.2.2.1 ADDRESS COMPARE LOGIC

The address compare circuit compares the hardwired (jumpered) address of the SCU with the address sent from the channel, via the Bus Out lines and identified by the Address Out tag. If the addresses compare equal, the selection logic will be enabled and Select Out will not be propagated to the next control unit.

6.2.2.2 SELECT LOGIC

The select logic enables the initial selection sequence to continue by setting bit 2 of the CHF Register via INITIAL SELECT. In addition, INT Register bit 5 (paragraph 6.3.4.2) is set, which allows the microprogram to recognize that the SCU is being selected.

6.2.2.3 PRIORITY LOGIC

The priority logic is used with a multi-channel SCU. This logic serves two basic purposes.

- Allows only one channel to select the SCU at any one time, and serves as a tie breaker if two channels attempt to select the SCU at the same time. Priority is determined by CP Register bits. When these bits are set, no new selection can occur.
- Prevents any channel from having two successive selection sequences when another channel is attempting to select the SCU.

6.2.2.4 MULTIPLEXER/DEMULTIPLEXER CIRCUITS

These circuits either take multiple signal inputs and gate them onto a single bus with appropriate timing, or they gate a single bus input into multiple outputs.

6.2.2.5 REGISTERS

The sequencing of the Channel Interface lines in the SCU to communicate with the channel is controlled by six special purpose registers.

1. Channel Interrupt (CI) Register
2. Channel Flags (CHF) Register
3. Channel Control (CHC) Register
4. Channel Tags (CT) Register
5. Channel Priority (CP) Register
6. Transmit Request (TR) Register

These registers provide the SCU with the capability of testing channel outbound tags, setting or resetting channel inbound tags, and controlling the special interface sequences.

Channel Interrupt (CI) Register.

The CI Register can be multiplexed onto the A bus.

CI REGISTER BIT ASSIGNMENTS

0	1	2	3	4	5	6	7
SCU ADDR 8	SCU ADDR 4	SCU ADDR 2	SCU ADDR 1	CU BUSY IN PROG	TAG	ID CODE BIT 2	ID CODE BIT 1
FOR SELECTING CHANNEL						FOR SELECTING CHANNEL	

CI Register Bits 0-3; SCU Address. This is the SCU address corresponding to the selecting channel. These bits are hardwired by the FE at installation time to the control unit address desired by the customer.

CI Register Bit 4; CU Busy in Progress. This bit is set when at least one channel is in the process of receiving a Short Control Unit Busy sequence. The bit is reset when there is no Short Control Unit Busy sequence in progress.

CI Register Bit 5; Tag. This bit is set when the TAG/UNTAG switch is in the TAG position.

CI Register Bits 6-7; Channel Identification Code. This is the two-bit output of the channel scanner and tells the microprogram which channel has been hardware-selected when the Multiple Channel Switch feature is installed.

Channel Flags (CHF) Register

The Channel Flags (CHF) Register is an assembly of bits used as branch conditions for the microprogram, gated to the BR Register by Enable CHF. This register contains indications for: Bus Out Parity Error, CU End Owed, Initial Select, Address Out, Command Out, Chaining, Data Out, and Service Out.

CHF REGISTER BIT ASSIGNMENT

0	1	2	3	4	5	6	7
BUS OUT PARITY ERROR	CU END OWED	IN-ITIAL SELECT	ADDR OUT	CMD OUT	CHAIN-ING	DATA OUT	SER-VICE OUT

When the Multiple Channel Switch feature is installed there will be one CHF Register for each channel. In these cases, the CHF Registers are multiplexed by bits 6 and 7 of the CI Register.

CHF Register Bit 0; Bus Out Parity Error. Set by Channel Interface parity checking logic if incorrect parity is transmitted with any byte of data across the channel with CHC 2 reset.

CHF Register Bit 1; Control Unit End Owed. Set by Channel Interface hardware, indicating that a "short SCU busy" has been returned to the respective channel. This bit is reset when the channel accepts Control Unit End status.

CHF Register Bit 2; Initital Select. Set when the interface has started the selection sequence and reset when Hold Out falls. This bit is also active for the duration of a short/busy sequence.

CHF Register Bits 3-7; Out Tags. Set by the respective Out tags from each channel. Of these, Data Out/Service Out (CHF 6-7) are used by the microprogram for microprogram-controlled data transfer when Channel Transfer (CHC 0) is reset. For buffered channel data transfer (CHC 0 set), these bits are used by the interface hardware to sequence the data transfer.

Channel Control (CHC) Register

The Channel Control (CHC) Register is a special-purpose register with its bits defined as control functions. Loaded from D Bus in the microprocessor, it controls functions of Channel Transfer, Channel Write, Lock Channel Switch, Reset Bus Out Parity Error, Control Unit Busy, Suppressible Data Transfer, and a two-bit Channel ID Code.

CHC REGISTER BIT ASSIGNMENT

0	1	2	3	4	5	6	7
CHNL TRANS-FER	CHNL WRITE	RESET BUS OUT PARITY CHECK	CON-TROL UNIT BUSY	LOCK CHNL SW	SUP-PRESS-SIBLE DATA TRANS-	ID CODE BIT 2	ID CODE BIT 1
						FOR CHNL MUX	

CHC Register Bit 0; Channel Transfer. Set by the microprogram to initiate buffered data transfer.

CHC Register Bit 1; Channel Write. Set by the microprogram to indicate a Write operation to the buffered data transfer hardware; reset to indicate a Read operation.

CHC Register 2; Reset Bus Out Parity Check. Set by the microprogram to inhibit checking the Bus Out lines for correct parity. If bit 0 of CHF (Bus Out Parity Error) is set, setting this bit will reset the error indication.

CHC Register Bit 3; Control Unit Busy. Set by the microprogram to indicate to the interface hardware that the control unit is busy. Causes a "short SCU busy" to be returned to all channels requesting service.

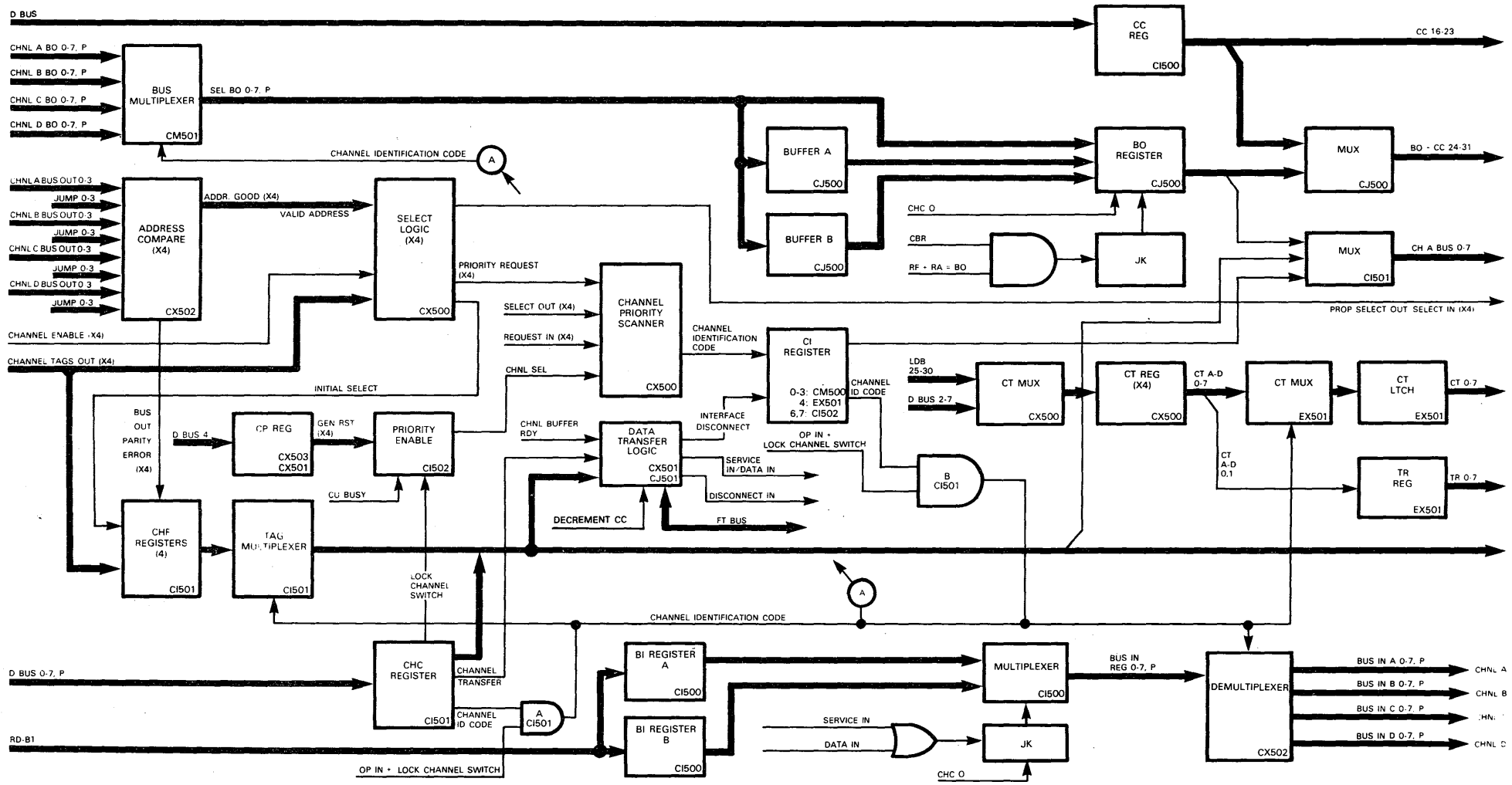


Figure 6-2. Channel Interface Block Diagram

CHC Register Bit 4; Lock Channel Switch. Set by the microprogram to disable the channel priority scanner to inhibit selection by channels other than the one already being serviced. Causes a "short SCU busy" to be returned to all other channels.

CHC Reg. Bit 5; Suppressible Data Transfer.

CHC Register Bits 6,7; Channel Identification Code. Set by microprogram or channel interface hardware to select one of four pairs of CHF and CT Registers when the Multiple Channel Switch feature is installed. If either Operational In (CT 3) or Lock Channel Switch (CHC 4) is set, the Channel ID (CI 6,7) is gated into CHC 6,7). If neither Operational In nor Lock Channel Switch is set, the microprogram has control of CHC 6,7.

CT REGISTER BIT ASSIGNMENTS							
0	1	2	3	4	5	6	7
REQUEST IN QUEUE STATUS	REQUEST IN STACKED STATUS	ALLOW DISABLE OF THIS CHNL	OPERATIONAL IN	ADDR IN	STATUS IN	DATA IN	SERVICE IN

Channel Tag (CT) Register

The CT Register can be loaded by specifying CT in the D field or C field (in Format 0) of a microinstruction. When the Multiple Channel Switch feature is installed there will be one CT Register for each channel. In these cases, one CT Register is selected by bits 6 and 7 of the CHC Register.

CT Register Bit 0; Request In, Queued Status. Set by the microprogram to raise Request In on the channel interface. This bit is ORed with the appropriate bit in the TR Register to raise Request In on the channel interface.

CT Register Bit 1; Request In, Stacked Status. Set by the microprogram to raise Request In on the channel interface if Suppress Out is down. This bit is ORed with Request In, Queued Status (CT 0), and the appropriate bits in the TR Register to return Request In to the selected channel.

CT Register Bit 2; Allow Disable. Set by the microprogram to allow the Channel Disable switch to affect the Channel Disable latch. This ensures that the microprogram finishes its operation on the channel before the channel is disabled.

CT Register Bit 3; Operational In (Controlled by Microprogram). Reset by interface hardware upon detecting a channel interface disconnect sequence.

CT Register Bit 4; Address In. Set by the microprogram to notify the channel that the SCU address is on the Data In bus.

CT Register Bit 5; Status In. Set by the microprogram to notify the channel that the SCU status is on the Data In bus.

CT Register Bit 6; Data In. Set by the microprogram to notify the channel that data is present on the Data In bus.

CT Register Bit 7; Service In. Set by the microprogram to notify the channel that data is present on the Data In bus.

Channel Priority (CP) Register

The CP Register can be loaded from the D Bus and multiplexed onto the A Bus.

CP REGISTER BIT ASSIGNMENTS							
0	1	2	3	4	5	6	7
		CHANNEL ERROR POINTER	OVERRUN ON CHANNEL WRITE	PRIORITY WINDOWS CH D	PRIORITY WINDOWS CH C	PRIORITY WINDOWS CH B	PRIORITY WINDOWS CH A

CP Register Bit 0-1; Unused. Set to zero.

CP Register Bit 2; Channel Error Pointer. Indicates channel-detected errors on Channels C or D.

CP Register Bit 3; Overrun on Channel Write. Indicates channel freeze under hardware data transfer. This bit is a flag only. It cannot be set or reset by a D Bus statement to this register.

CP Register Bit 4-7; Priority Windows, Channels A-D. When set by multichannel processor, blocks microprogram detection of INT bits (5) and CHF 2. If the channel attempts selection with this bit set, Select Out is blocked and no further channel action takes place until either: 1) CU Busy (CHC 3) and/or Lock Channel Switch (CHC 4) is set, in which case a short busy sequence will begin; 2) Priority Window (CT 7) is reset, in which case selection may proceed normally; or if the control unit is already selected, a short busy sequence will begin.

Transmit Request (TR) Register

The TR register can be loaded from the D Bus and multiplexed onto the A Bus.

TR REGISTER BIT ASSIGNMENTS							
0	1	2	3	4	5	6	7
REQUEST IN QUEUED STATUS CH D	REQUEST IN QUEUED STATUS CH C	REQUEST IN QUEUED STATUS CH B	REQUEST IN QUEUED STATUS CH A	REQUEST IN STACKED STATUS CH D	REQUEST IN STACKED STATUS CH C	REQUEST IN STACKED STATUS CH B	REQUEST IN STACKED STATUS CH A

TR Register Bits 0-3; Request In Queued Status. Set by the microprogram to raise Request In on the desired channel interface. Any combination of channels may have Request In active concurrently.

TR Register Bits 4-7; Request In Stacked Status. Identical to TR Register bits 0-3 except that the channel may suppress the request by raising Suppress Out.

TR Register Bits 0-7. When displayed on A Bus, represent the state of the channel tag, which may have been set by the CT Register. Either the CT or TR Register may reset Request In regardless of which register activated this tag.

Bus In (BI) Register

The BI Register is the interface with the Microprocessor and the Bus In lines on the channel. Normally the register is loaded from the D Bus as if it were an eight-bit register, into BI Register A only. In the buffered data transfer mode, however, the D Bus lines are loaded into both BI Registers A and B to permit buffered operation. These registers are multiplexed and transferred by the channel hardware to the Bus In lines of the channel. When CHC bit 0 (Channel Transfer) is set without bit 1 (Channel Write), BI interfaces directly with the two-byte buffer in the Channel Interface hardware, which then interfaces with the Bus In lines on the channel hardware. When Channel Transfer is reset, BI interfaces directly with the Bus In lines. BI is loaded by the microprogram via the D Bus. While Channel Transfer is set, BI is loaded at least one instruction after the microprogram has verified that the CBR condition is set. While Channel Transfer is reset, BI is loaded at least one instruction prior to the associated inbound tag being set in the CT Register.

Bus Out (BO) Register

The BO Register is the interface with the Microprocessor and the Bus Out lines on the channel. This register is the data communication link between the channel and the Microprocessor. The register is made up of three registers: two used as buffers which are loaded asynchronously by channel tags, and a third which is synchronous with the Microprocessor. The input to the synchronous register is multiplexed and controlled by the processor decode of RA = BO.

When in the nondata transfer mode, information is passed directly to the multiplexer and is latched in the synchronous register. The other two registers are inactive at this time. When CHC bit 0 (Channel Transfer) and bit 1 (Channel Write) are set, the Bus Out lines interface directly with the two-byte buffer in the channel hardware, which then interface with the BO entry on the A Bus in the Microprocessor. While Channel Transfer is reset, the Bus Out lines interface directly with the BO entry to the A Bus. During buffered data transfers for write operations, the data in BO may be transferred in the same microinstruction in which the CBR condition was tested.

Channel Byte Count (CC) Register

The CC Register is a sixteen-bit preloadable counter which controls the number of bytes transferred during hardware data transfers with the channel. While CHC bit 0 (Channel Transfer) is set, the CC Register is decremented each time the channel responds to Data In or Service In with Data Out or Service Out, respectively. Since the decrementing of the CC Register (when Channel Transfer is set) is asynchronous to the Microprocessor cycle, the CC Register can be loaded to or from R23 only when the Channel Transfer bit is reset.

6.2.3 Data Transfer Organization

6.2.3.1 GENERAL

The SCU Channel Interface provides the capability for buffered data transfer between the user's system channel and the SCU Microprocessor. Buffered data transfers to the channel are initiated through the CHC and CC Registers.

Data transfers between the channel and SCU fall into two categories:

- Read Operations—Cause data to be transferred from the SCU to the channel via the Bus In lines.
- Write Operations—Cause data to be transferred from the channel to the SCU via the Bus Out lines.

The microprogram initiates buffered data transfers by setting Channel Transfer bit 0 in the CHC Register. At the same time, the microprogram establishes whether the transfer will be a Write or a Read by respectively setting or resetting Channel Write bit 1 in CHC.

Once the microprogram has initiated a buffered data transfer, the channel hardware sequences Service In and Data In to effect the transfer of data between the one-byte channel bus and the two-byte buffer within the Channel Interface hardware. Each time a byte is transferred, the CC Register is decremented to show how many bytes are remaining to be transferred between the channel and the SCU. The decrement condition is logically Service Out AND Service In, or Data Out AND Data In. When the CC Register is decremented to zero, a latch (CC = 0) is set in the INT Register of the Microprocessor, and the Channel Interface hardware is inhibited from raising Service In or Data In, thus stopping the transfer of data. For example, if a count of five is placed in the CC Register, five bytes will be transferred between the SCU and channel on both Read and Write operations. The decrementing of CC is asynchronous to the Microprocessor; therefore, the Channel Transfer bit in CHC must be reset to transfer CC to or from R23.

The set condition for the CC = 0 latch is logically Service In AND Service Out, or Data In AND Data Out, when the value of the channel counter is equal to one. The CC = 0 latch is reset when the microprogram reloads the CC Register. The CC = 0 latch, which is INT Register bit 0, is synchronous with the Microprocessor cycle so that the microprogram may branch on the condition at any time.

The interface between the channel busses and the Microprocessor are the two registers, Bus In (BI) and Bus Out (BO). While the Channel Transfer bit in CHC is reset, these two registers directly reflect the Bus In and Bus Out lines of the channel. When Channel Transfer is set in CHC for Read operations, BI Register interfaces with the Bus In lines on the channel via the two-byte buffer. For Write Operations, BO Register interfaces with the Bus Out lines on the channel via the two-byte buffer.

The detailed operation of the data transfer hardware for Read and Write operations is described below.

6.2.3.2 READ OPERATIONS

A simplified block diagram of the Channel Interface circuits involved during a Read operation is shown in Figure 6-3. During Read operations, the Channel Interface hardware accepts data from the microprogram via the BI Register and transfers this data asynchronously to the channel over the Bus In lines. The BI Register is structured as two eight-bit buffers between the D Bus and the Bus In lines. Since the data from the disc drive is processed regularly by the microprogram (average of 1 byte every 1.24 microseconds), these two buffers allow the time between byte transfers to the channel to vary considerably.

To initiate buffered data transfers from the SCU to the channel, the microprogram loads the CC Register with the number of bytes to be transferred and sets the Channel Transfer in the CHC Register. Loading CC causes the CC = 0 latch to be reset if it was on, and setting Channel Transfer gates the two-byte BI buffer into the data transfer path. After the correct count has been loaded in CC and Channel Transfer has been set, the Channel Interface hardware sets the Channel Buffer Ready (CBR) and Queue Empty (QE) branch conditions in the Microprocessor. The CBR conditions indicate that one or both buffers have not been loaded by the microprogram, and the QE condition indicates that both buffers are empty.

When the microprogram responds to the CBR condition by loading a byte into BI, the Channel Interface hardware places this byte on Bus In to the channel and raises Data In. The microprogram loading BI causes the QE indication to be reset since one buffer now contains a byte of data.

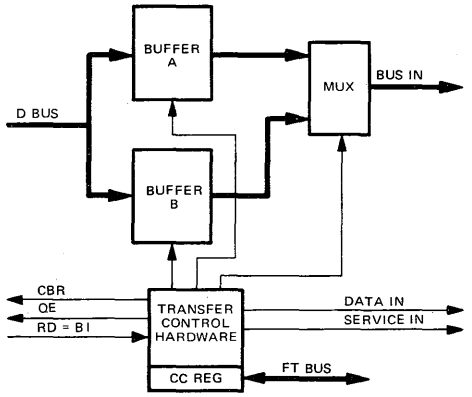


Figure 6-3. Read Data Transfer Block Diagram

CBR, however, remains set since there is still one available buffer for more data. Loading BI initiates the buffered data transfer.

At this point, the channel may respond to Data In with Data Out (or Command Out indicating truncation) indicating the byte has been accepted (or rejected in the case of Command Out) by the channel, and thereby cause the QE condition to be set again. Alternately, the microprogram may load another byte into BI (since the CBR condition is still set) and thereby cause the CBR condition to be reset. If the former condition occurs, the CC Register is decremented. If decremented to zero, Data In and Service In are inhibited from being set again until the CC Register is reloaded. On the other hand, if the latter condition occurs, both buffers will contain data which had not been accepted by the channel, and the microprogram may not load another byte of data into BI until the channel accepts at least one of the buffered bytes.

The data transfer to the channel continues with the Channel Interface hardware alternately raising Data In and Service In to transfer successive bytes until one of the following conditions occurs:

- The CC Register is decremented to zero, in which case the data transfer has been completed.
- The channel truncated the data transfer by raising Command Out in response to Service In or Data In, in which case the CC Register is not decremented, the QE condition is not set, and the CBR condition can be either on or off. When the microprogram detects truncation through INT bit 5 (Channel Interrupt) or CHF bit 4 (Command Out). It will reset the Channel Transfer bit in CHC, ending the buffered data transfer.
- The channel issues a *Halt I/O* or *Halt Device* instruction to the SCU by raising Address Out, in which case the microprogram will reset the Channel Transfer bit in CHC.

For Sense commands, where the data transfer is a Read operation over the channel and the data need not be transferred regularly, the microprogram indicates suppressible data transfer to the channel hardware by setting bit 5 in CHC. The transfer operates identically to that described above, except that the Channel Interface hardware is inhibited from setting Data In or Service In to transfer a new byte while Suppress Out is set at the channel. When Suppress Out is off, however, the data transfer hardware continues unaffected.

6.2.3.3 WRITE OPERATIONS

A simplified block diagram of the Channel Interface circuits involved during a Write operation is shown in Figure 6-4. During Write operations, the Channel Interface hardware accepts data asynchronously from the channel over the Bus Out lines and transfers this data to the microprogram via the BO Register. The channel hardware contains a two-byte buffer between the Bus Out lines and the BO Register. Since data is required for writing by the controller and disc drive regularly (average of 1 byte every 1.24 microseconds), these two buffers allow the time between byte transfers from the channel to vary considerably.

To initiate buffered data transfers from the channel to the SCU, the microprogram loads the CC Register with the number of bytes to be transferred and sets Channel Transfer and Channel Write in the CHC Register. Loading CC causes the CC = 0 condition to be reset if it was set prior to the loading. Setting the Channel Transfer and Channel Write bits in CHC gates the two-byte BO buffer into the data transfer path. CBR is reset by the hardware one instruction cycle after Channel Transfer and Channel

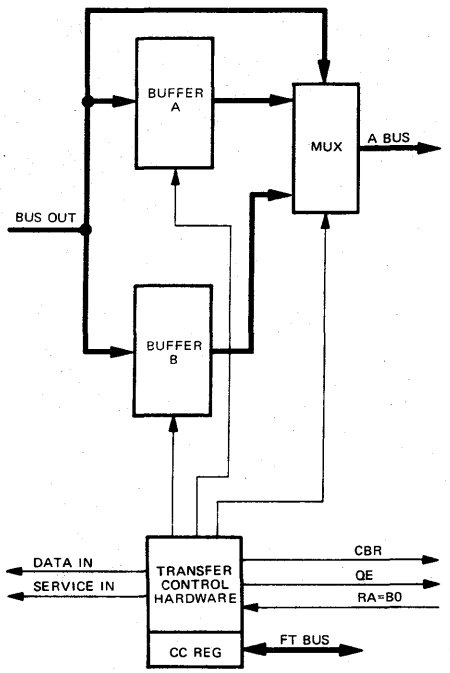


Figure 6-4. Write Data Transfer Block Diagram

Write are set. When the microprogram sets the Channel Transfer and Channel Write bits with the CC = 0 condition reset, the channel hardware raises Data In, requesting the first byte of data from the channel. Until the first byte of data is transferred by the channel, the CBR condition remains reset, indicating to the microprogram that data is not yet available from the channel. As soon as the channel transfers the first byte of data, the Channel Interface hardware gates the byte to the BO Register. The CBR condition is then set to indicate to the microprogram that data is available in the BO Register from the channel. Also, since there is still one buffer available for data, the channel hardware immediately raises Service In to request another byte of data to be loaded into the second buffer.

At this point the microprogram may respond to the CBR condition by gating the BO Register to the A Bus, thereby causing the CBR condition to be reset. Alternately, the channel could transfer another byte to the SCU (since one buffer is still available for new data), thereby causing the CBR condition to remain set. In either case, the CC Register is decremented with each byte transferred by the channel, and if decremented to zero, Data In and Service In are inhibited from being set again until the CC Register is reloaded. If the latter case occurs, however, both buffers will contain data which had not been accepted by the microprogram; consequently, the Channel Interface hardware cannot request another byte from the channel until the microprogram has accepted at least one of the buffered bytes.

The data transfer from the channel continues with the Channel Interface hardware alternately raising Data In and Service In to request successive bytes until one of the following conditions occurs:

- The CC Register is decremented to zero, in which case the data transfer has been completed. When the microprogram detects this through INT bit 0, it resets the Channel Transfer and Channel Write bits in CHC.
- The channel truncates the data transfer by raising Command Out in response to Service In or Data In, in which case the CC Register is not decremented and the CBR condition is not set if it was reset. The microprogram detects truncation through INT bit 5 or CHF bit 4.
- The channel overran the data transfer rate of the disc drive by not responding to Service In or Data In with Service Out or Data Out before a byte of data was required. When the microprogram detects overrun, it will execute a Channel Freeze control statement and set CP Register bit 3 (Overrun on Channel Write). Executing this control statement causes the present Service In or Data In transfer sequence to be completed, and inhibits the initiation of any new transfer sequences. If Service In and Data In are both reset when the Channel Freeze control is executed, data transfer is immediately halted. After the Channel Freeze control is executed on data overruns, the Queue Empty condition is set when the present transfer sequence has been completed and no new sequences are to be initiated. When QE is set, the microprogram will then reset Channel Transfer and Channel Write.
- The channel issues a *Halt I/O* or *Halt Device* instruction to the SCU by raising Address Out, in which case the microprogram will reset the Channel Transfer and Channel Write bits in CHC.

For Seek commands where the data transfer is a Write operation over the channel and the data need not be transferred regularly, the microprogram indicates Suppressible Data transfer to the channel hardware by setting bit 5 in CHC. The transfer operates identically to that described above, except that the Channel Interface hardware is inhibited from setting Data In or Service In to transfer a new byte while Suppress Out is set at the channel. When Suppress Out is off, however, the data transfer hardware continues unaffected.

6.2.4 Channel Interface Branch Conditions

6.2.4.1 CHANNEL BUFFER READY (CBR)

Channel Read

Indicates that one or both channel buffers are empty.

Channel Write

Indicates that one or both channel buffers are full. The buffer may be unloaded in the same instruction as the -B (CBR) statement.

6.2.4.2 QUEUE EMPTY

Indicates that both channel buffers are empty during a Read operation or the CC Register contents are zero.

6.2.4.3 CHANNEL INTERRUPT (INT 5)

Channel Interrupt is active under one of the following conditions:

1. Initial Selection AND Not Short Busy Sequence AND Not Operational In.
2. Operational In AND Halt I/O + Truncation.

6.3 MICROPROCESSOR DESCRIPTION

6.3.1 Microprocessor General Description

The Microprocessor controls overall operation of the 3672-based system via the microprogram Writable Control Storage (WCS). The Microprocessor contains the registers, data busses, and storage necessary to interpret and execute subsystem commands. Commands are executed by translating the microinstruction fields into hardware control signals which perform the actions required to execute microinstructions.

The Microprocessor is organized into the following functional areas (Figure 6-5):

- Writable Control Storage—Provides storage for the microprogram and diagnostic routines.
- Data Paths—Composed of the ALU, registers, and busses necessary to operate the Microprocessor.

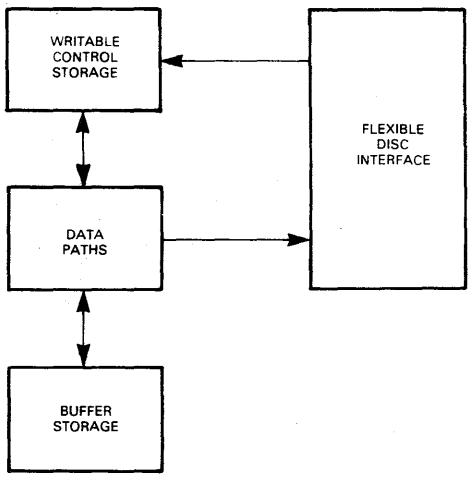


Figure 6-5. Microprocessor Block Diagram

- Buffer Storage—Provides temporary storage for Drive status, a working area for some types of error correction, and information related to drive orders.
- Flexible Disc Interface—Provides control and data interface for the Flexible Disc Unit.

6.3.2 Microinstructions

6.3.2.1 DESCRIPTIONS AND FORMATS

The microprogram controls the SCU hardware through the sequencing of microinstructions. These microinstructions are divided down into fields whereby each field controls a specific function in the hardware. In the SCU the data for these microinstructions is stored in WCS.

There are four microinstruction word formats. Each format, and the fields contained therein, raises specific controls which operate upon the logic circuits.

Because of the large number of controls needed within the SCU, the 34 usable bits of the microinstruction word are translated into 57 usable bits. This translation concept is used to allow a smaller memory word size.

Specific details about the four microinstruction formats and the interpretation of each field are contained in paragraph 4.1 of Section 4. For convenience in the discussion of microinstructions in this section, the four microinstruction formats and their expansion from a 34-bit word stored in WCS to a 57-bit word used for hardware control, Figure 4-3 is repeated in this section as Figure 6-6.

6.3.2.2 MICROINSTRUCTION FIELD TRANSLATION AND BRANCHING

Translation

The microinstruction translation hardware translates the 34-bit microinstruction word into 57 hardware control lines. This translation occurs after the microinstruction has passed through the ECC circuits and the data bits are latched at the output of the ECC circuits (Figure 6-10).

The translation during each machine cycle is controlled by the microinstruction format code. The 57 hardware control bits are gated through the translation matrix by

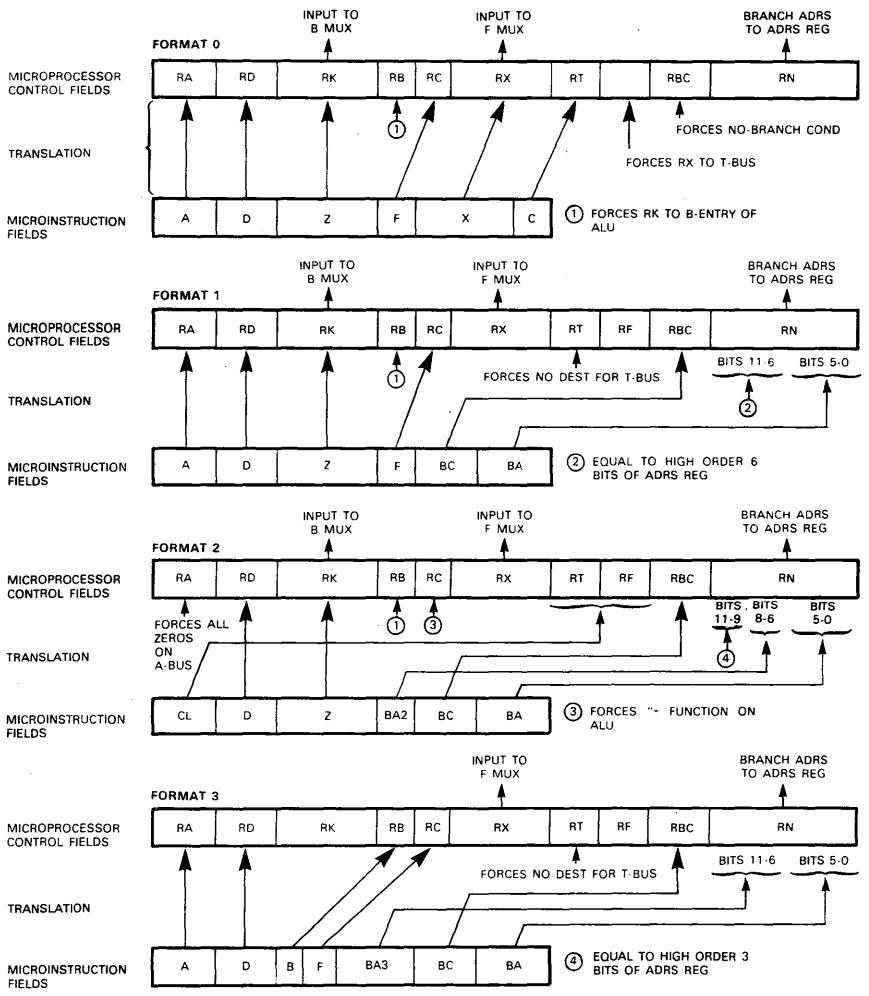


Figure 6-6. Microinstruction Expansion

the format code as shown in Figure 6-7. The outputs of the translation matrix control the Microprocessor hardware while performing ALU operations, register-to-register transfers, and so forth.

Branching

The branch multiplexing circuits are shown in Figure 6-8. These circuits multiplex inputs from several different sources under control of the Page Select (PS) Register. This result is multiplexed with a branch condition or the Interrupt (INT) Register contents, under control of the microinstruction BC field, to produce a final branch address.

The first stage of multiplexing is accomplished by four circuits: the BRA multiplexer, the BR multiplexer, and gates A and B. All four circuits are enabled by PS 4-7. The BRA multiplexer gates inputs from R0, R1, R2, R3, G4, G7, D, and eight interrupt conditions via the four PS bits through the PSA latch. The BR multiplexer gates the output of the BRA multiplexer and DI Register via ENABLE DI. This enable is also developed from PS 4-7 through an encoder. Gates A and B enable the output from the BR multiplexer or CHF via ENABLE CHF. This enable is also developed from PS 4-7. The table in paragraph 6.3.4.1 gives a complete listing of the encoded combinations of PS 4-7 required to gate each input to the multiplexers.

The result of this first stage of multiplexing is called BR + CHF. The BR + CHF lines are multiplexed with the branch conditions, or the INT Register, by the BC field to determine the branch condition specified in the Microinstruction word. If the branch condition is met, the BRANCH ADDRESS SET signal is generated to load the specified branch address.

6.3.2.3 MICROINSTRUCTION EXECUTION EXAMPLE

An example of how microinstructions are executed from the information contained in their fields is shown in Figure 6-9. The microinstruction field information can be summarized as below.

Execution of this microinstruction is carried out in the following sequence. First, the microinstruction is read from WCS location 00E1. Next, the ALU operation specified in the F field is performed on the registers designated by the A and B fields; namely, the contents of the BO Register are added to those of the GI Register and

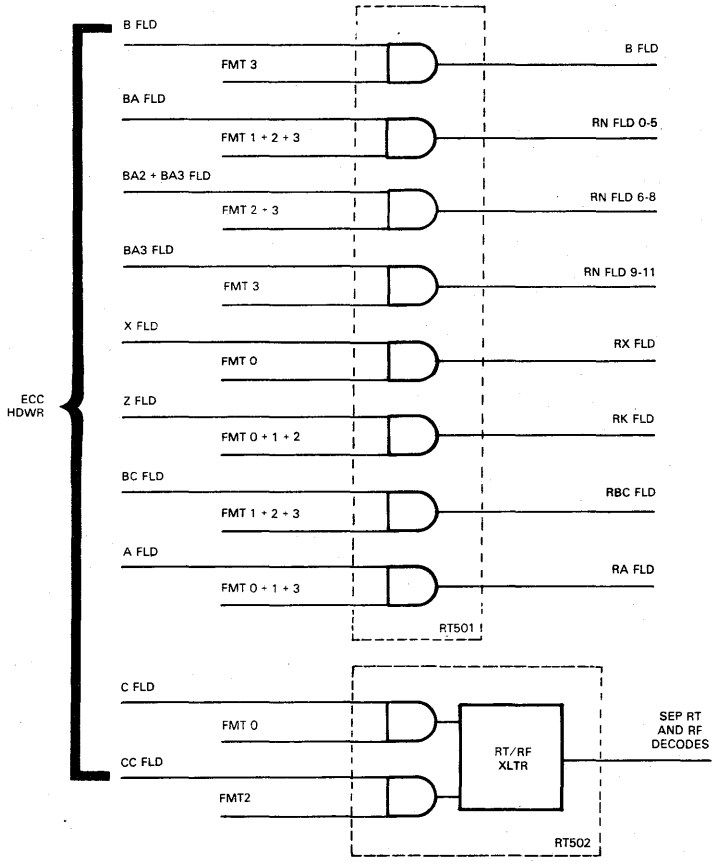


Figure 6-7. Microinstruction Translation

the result stored in the CHC Register. Then the branch specified by the BC field is performed. Since the branch is unconditional, formation of the branch address takes place immediately in the Address Register. This amounts to gating the contents of the BA3 and BA fields to form the high-order and low-order parts of a 12-bit branch address. (Bits 11-0 of this register run from left to right, opposite from all other SCU Registers.) Expressed in hexadecimal form, this branch address is 06CA, which is where the program jumps to read the next microinstruction in the program.

6.3.3 Writable Control Storage Description

6.3.3.1 GENERAL

The Writable Control Storage (WCS) is a volatile semiconductor storage system organized on seven printed circuit boards. Each of the seven boards contains ninety-six 256 × 1 RAM chips. The chips are arranged on each board in a 6 × 16 matrix so that each board stores 6 bits of 4096 words. The seven boards together provide 4,096 words of 42 bits. The 42-bit word is subdivided to provide 34 data bits (normal microinstruction or microdiagnostic word as it resides in storage) plus 8 bits used for error correction. A WCS extension feature is available to increase the size of WCS from the nominal 4,096 words.

6.3.3.2 READING FROM WCS

Refer to the WCS block diagram in Figure 6-10.

The read access of WCS words takes place whenever a valid address exists at the input of the memory board. This address is provided from the RAM Address Register which may be conditioned by the following:

- Branch on D Bus (D BUS)

  The low-order 8 address bits to WCS are determined by the information resulting from an ALU operation logically "ORed" with the Branch Address field of WCS. The remaining bits are determined in the normal way, depending on which format is being executed.

- Initial Microprogram Load (IMPL)

  The WCS address during IMPL is determined by a loader source register. The entire address field

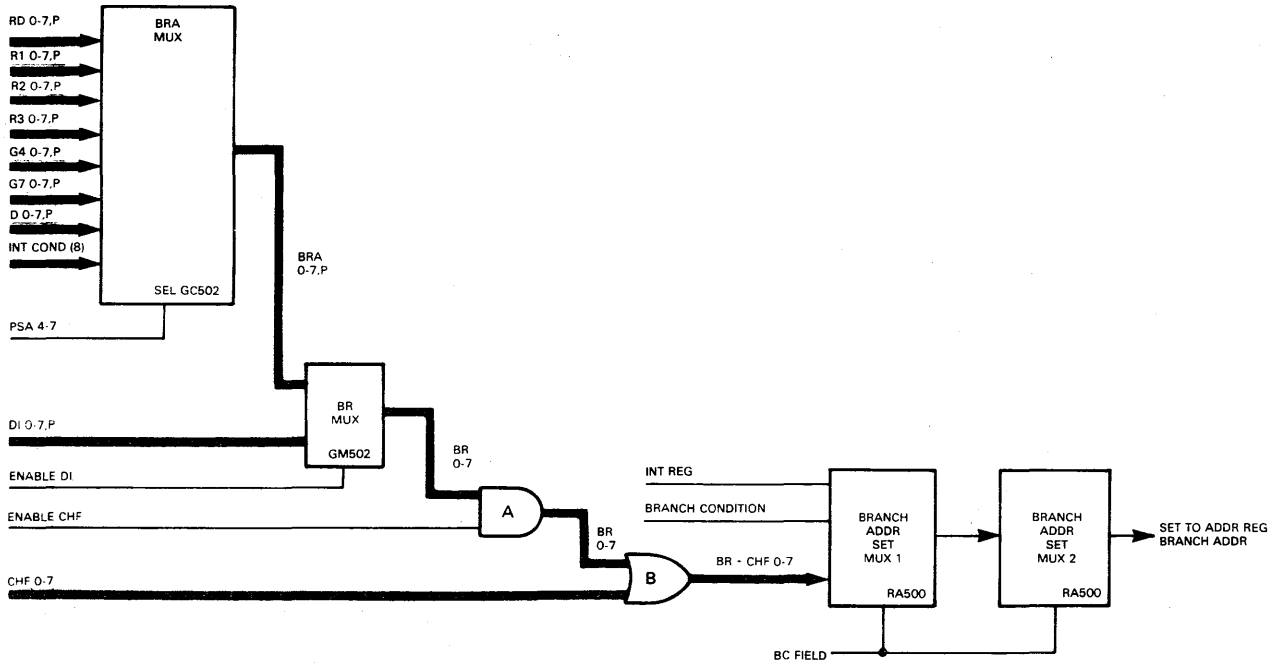


Figure 6-8. Branch Multiplexing Circuits

is determined by the contents of this register. Other address sources are inhibited from loading the Address Register during this mode.

- Start Address (START ADRS)

  The Start Address is loaded from the FE panel control and selection. The logic on this board accepts loading of the start address mode.

- Subroutine Register Address Source (SR EXT)

  With the B(ADDR = SR) statement, the address is derived externally from this card by means of the SR Register. All other normal operational means of address determination are inhibited.

- Address Incrementer (INCR)

  The address incrementer is a parallel load counter loaded by the present address initiated. The counter then is incremented by 1 to determine the next address during Format 0 execution or when branch conditions are not satisfied. During the memory scan mode, the incrementer is a constant source for addressing.

- Branch Address (BRCH)

  The branch address is determined by whether the list of branch conditions shown in Figure 6-9 are satisfied and which microinstruction format mode is being operated on.

Once the address is received at the input of the RAM boards, address bits 8, 9, 10, and 11 are decoded to provide a select for one column of six RAM chips in the 6 × 16 (per board) memory matrix (Figure 6-11). The other 8 address bits (0-7) are applied at the chip level to select the data bit stored at one of 256 addresses. This decoded select and chip address provides 6 bits read-out of each of the seven memory boards for a total of 42 bits. The output of the RAM boards is set into the RAM Data Register to be acted upon while the next address is being conditioned into the Address Register.

A read access from WCS requires 110 nanoseconds after the address is presented to the RAM board for data to be good at the output of the card. For the relationship in time of addressing, refer to paragraph 6.3.7.

6.3.3.3 WRITING INTO WCS MEMORY

During a Write into the WCS, the address input to the RAM board is decoded in the same fashion as during the Read to select the chips in which data is to be written. The following input relationship is necessary to accomplish a Write function.

A RAM word is written when the WCS signal is sent to the RAM PCB.

The data written is the data contained in a 42-bit Write Data Buffer located in the FD interface. A Write function takes place in two different ways:

1. During IMPL the bootstrap is assembled from the FD unit and stored into RAM under hardware control.

2. When the microinstruction being executed contains the WCS control statement.

6.3.3.4 ERROR CORRECTION CODE (ECC)

The microinstruction and microdiagnostic words read from the Flexible Disc are accompanied by an ECC read from the Flexible Disc. As a word is passed through the Flexible Disc Interface, a new ECC pattern is generated and compared with the ECC pattern read from the Flexible Disc. A miscompare results in an FD Read Error, and a Check-1 condition is set if doing an IMPL (paragraph 6.6). If not doing IMPL, the error causes a Check-2 condition.

ECC Generation

Due to the complexity of ECC generation, it will be explained by example.

The ECC is generated using the matrix shown in Table 6-1

An example data word applied to the ECC matrix is shown in Table 6-2. Each 1 bit of the data word is applied to three rows of the matrix. Each 1 bit in a horizontal row (A through H) is counted and the ECC bit (P_1 through P_8) for that row is set (or reset) to make the bit count for a given row even. As an example, bit position 2 of the example data word is a 1 and is applied to rows A, G and H of the ECC matrix. Bit position 23 of the example data word is a 1 and is applied to rows A, C and D of the ECC matrix, etc.

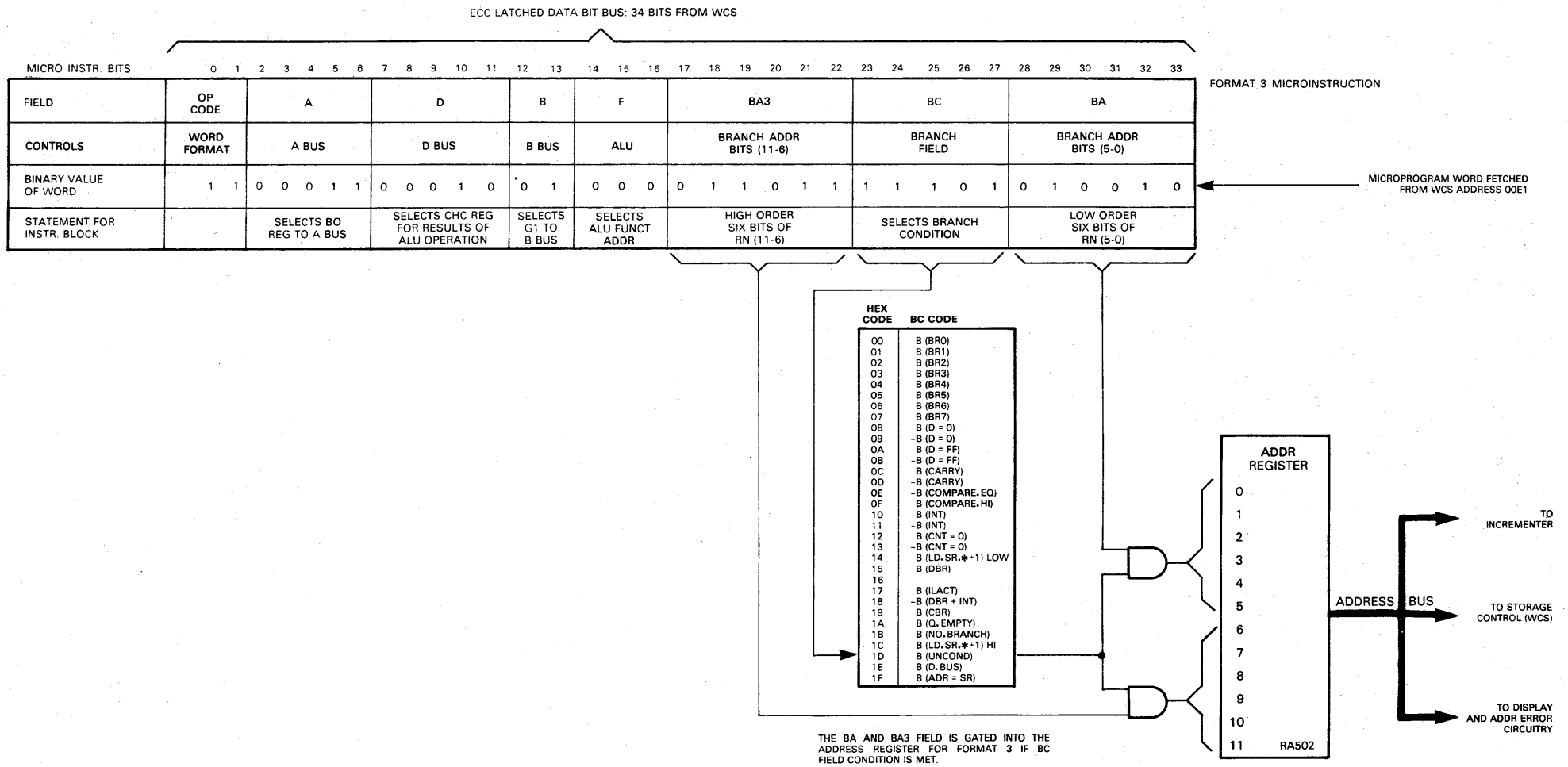


Figure 6-9. Microinstruction Execution Example

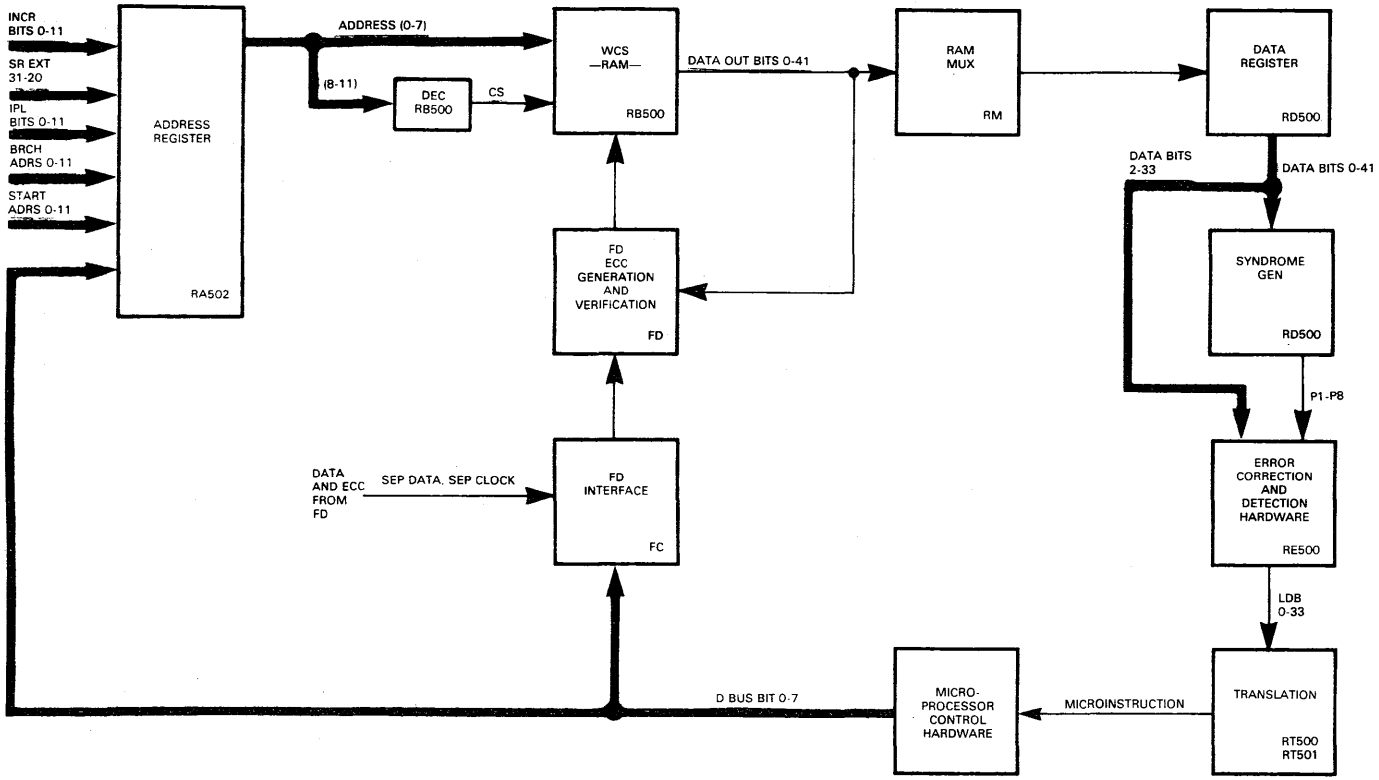


Figure 6-10. WCS Block Diagram

Row A of the example has bits 2 and 23 noted, giving an even count for that row; therefore P_1 is a zero. However, row H has bit position 2, 5, and 6 noted, giving that row an odd count. Therefore, P_8 is turned on to make the count for row H even.

Other bits of the data word are applied to the matrix in a similar manner. The generated ECC is shown at the right of the example.

ECC Checking

The ECC used with WCS allows detection and correction of single bit errors (one bit dropped or picked up). Multiple bit errors can be detected but not corrected.

After the data from the WCS is set in the Data Register, the data and its ECC bits are passed through a syndrome generating circuit. The generated syndrome is applied to the correction circuits. If the syndrome is all zero's, no error exists. If the syndrome has 3 and only 3 bits present, a correctable error exists and is therefore corrected. If any combination of syndrome bits are present, except 3, an uncorrectable error exists and the SCU will stop.

ECC Checking From the FE Panel

When necessary, the contents of a WCS word, after error correction, can be displayed on the FE panel (Display Roller at position C). The syndrome bits (P1-P8) used in error correction may also be displayed (Display Roller at position H).

Also note that the SCU can be forced to stop on single (correctable) ECC errors when in FE mode with the CHK2 STOP switch in the on (up) position. This enables the FE to detect single bit errors during scheduled maintenance.

6.3.3.5 WCS EXPANSION

The WSC expansion feature enlarges the size of WCS by providing increased storage to support additional features. These features are implemented through the use of additional RAM boards. As shown in Figure 6-12, each RAM board can store 512 42-bit words. These boards are addressed by ADDR BIT 0-12. Bits 0-7 are used to select one of 256 words on a board. Bit 8 selects either the upper or lower 256-word portion of a board. Bits 9-12 are used to select one of the RAM boards. (The detailed selection scheme is shown in the logics for the RX board.)

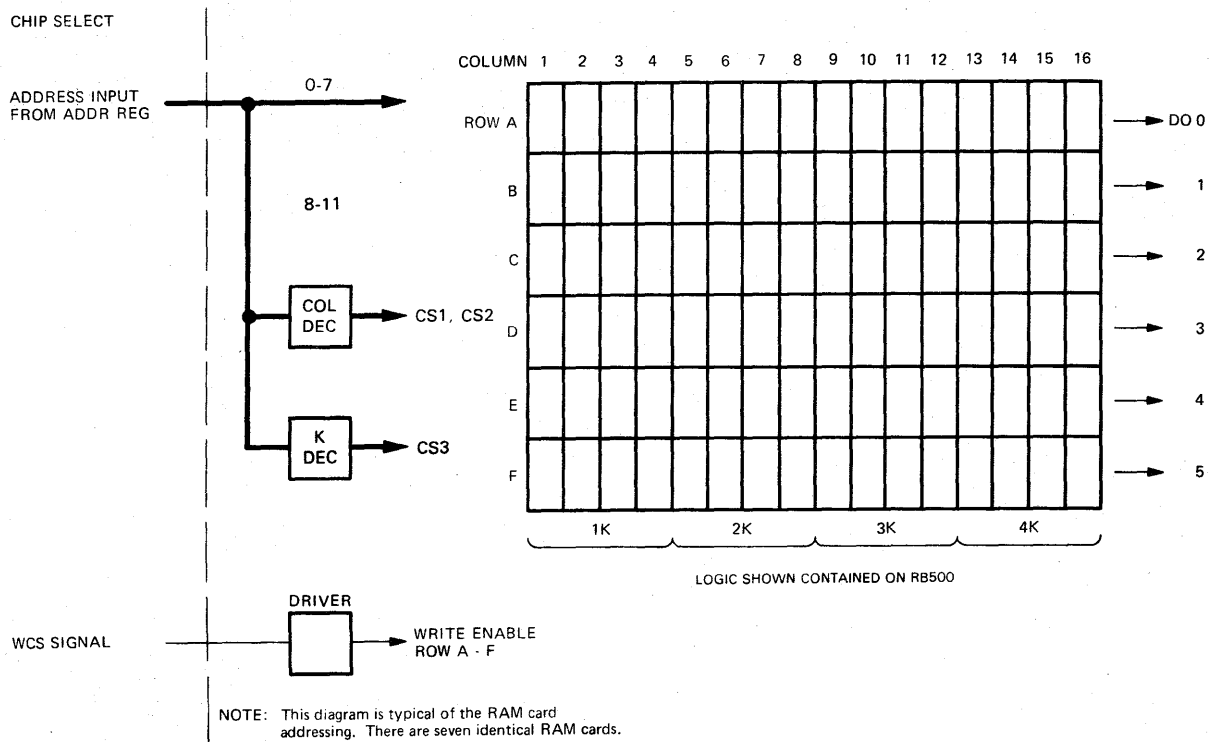


Figure 6-11. RAM Board Column Select

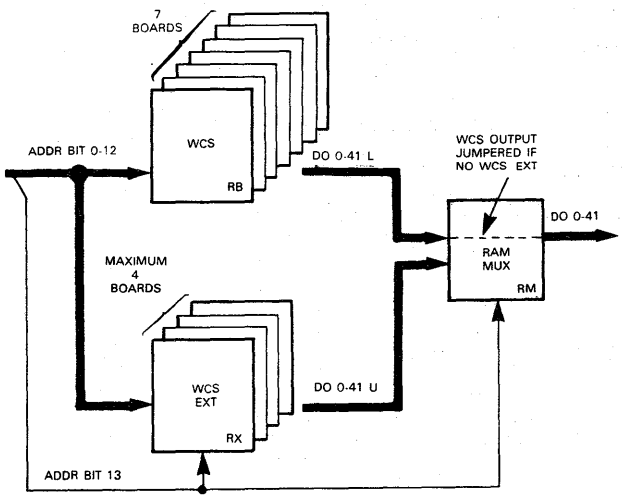


Figure 6-12. WCS Extension Block Diagram

The selected word from the WCS extension is sent to the RAM data multiplexer. The multiplexer is also fed with the output from the 4K WCD, and is used to select words from either the WCS or the WCS extension. Selection is accomplished via address bit 12, which selects WCS if a zero or WCS extension is a one. If the WCS extension feature is not present in the SCU, jumpers are connected on the multiplexer board back panel to bypass the multiplexer.

6.3.4 Data Paths

6.3.4.1 BUSES

The Microprocessor has five major buses which allow the transfer of data throughout the SCU. These buses consist of the A Bus, B Bus, D Bus, T Bus and BR Bus. The T Bus is 36 bits wide to allow the transfer of 32 data bits and 4 parity bits. The A, B, D and BR buses are each 8 bits wide. Figure 6-13 illustrates the functional arrangement of the buses.

A Bus

This bus transfers data from a register selected by the RA control bits (as specified by the A field of the microinstruction word) to the A entry of the ALU in Format 0, 1 and 3 microinstructions. In Format 2 microinstructions, the RA field is forced to zero to place all zeros on the A entry to the ALU.

B Bus

This bus transfers the RK control bits (specified by the Z field of the microinstruction) to the B entry of the ALU in Format 0, 1 and 2 microinstructions. In Format 3 microinstructions, the data from a register selected by the RB control bits (as specified by the B field of the microinstruction word) is transferred to the B entry of the ALU.

D Bus

This bus transfers data from the output of the ALU to a register determined by the RD control bits (as specified by the D field of the microinstruction word).

T Bus

This bus transfers data from the output of the F multiplexer to a register determined by the RT control bits. In Format 0 microinstructions, the X field is placed on the T bus and the RT control bits are specified by the C field of the microinstruction. In Format 2 microinstructions, the RT control bits are decoded from the CL field of the microinstruction to effect the necessary register-to-register transfer.

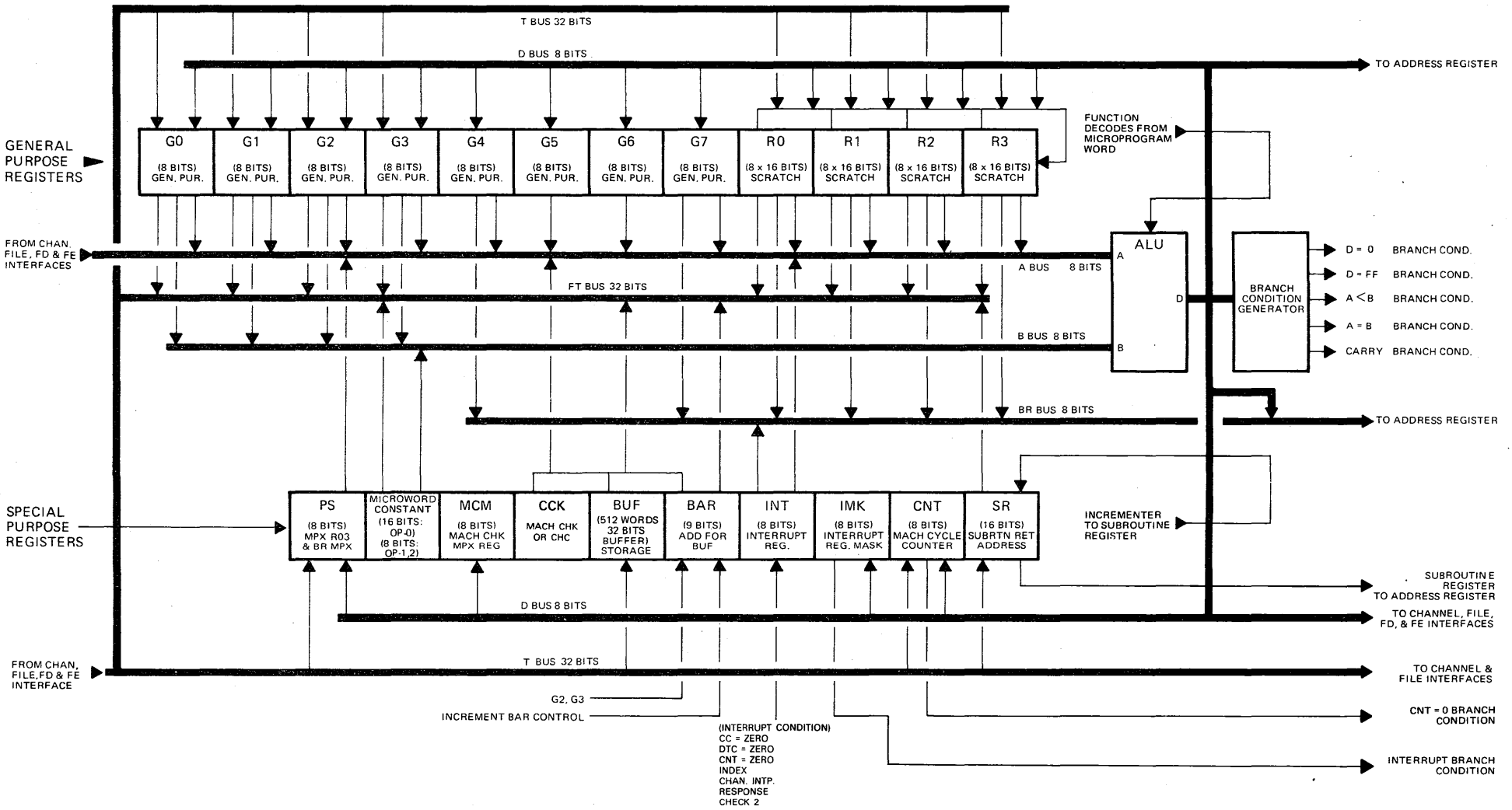


Figure 6-13. Microprocessor Buses and Registers

TABLE 6-1. WCS ECC MATRIX

ROW	BIT POSITION 0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	P_1 34	P_2 35	P_3 36	P_4 37	P_5 38	P_6 39	P_7 40	P_8 41
A	1	1	1					1	1							1	1							1	1	1			1	1	1				1							
B	1	1						1		1	1							1	1								1	1	1			1	1	1		1						
C				1						1		1	1							1	1			1	1		1	1		1		1						1				
D	1				1						1		1							1		1	1	1		1	1				1		1					1			
E		1				1			1		1				1	1		1			1	1			1	1		1					1						1			
F							1						1	1	1		1		1				1						1	1	1	1	1	1					1		
G			1	1	1	1	1	1	1	1	1	1	1	1	1																									1	
H			1	1	1	1	1									1	1	1	1	1	1	1	1																			1

TABLE 6-2. EXAMPLE DATA WORD APPLIED TO WCS ECC MATRIX

BIT POSITION	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	P_1 34	P_2 35	P_3 36	P_4 37	P_5 38	P_6 39	P_7 40	P_8 41
EXAMPLE DATA WORD	0	0	1	0	0	1	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0								
A			1																					1											0							
B																																				0						
C													1											1													0					
D																								1														1				
E						1																																	1			
F							1						1																											0		
G			1			1	1						1																												0	
H			1			1	1																																			1

BR Bus

This bus transfers data from one of the eight-bit registers to the branching multiplexer. The register to be multiplexed onto the BR bus is determined by bits 4, 5, 6 and 7 of the PS Register. The registers available to the BR multiplexer and the associated hexadecimal code required in the PS Register are shown below.

HEXADECIMAL CODE IN PS REGISTER BITS 4, 5, 6 AND 7	REGISTER GATED TO BR BUS
0	G4
1	R1
2	R2
3	R3
4	INT
5	R0
6	G7
7	D
8	CHF
9	
A	
B	
C	DI
E	
F	

6.3.4.2 REGISTERS

The storing of data and the controlling of interface hardware in the SCU is accomplished with general purpose and special purpose registers. The general purpose registers are not identified with any special hardware function of the machine. At one point in the microprogram, a general purpose register may be used to store the count of time duration; however, at another point, that same register may be used to store data which has been obtained from the CPU for a compare operation. Special purpose registers, however, are identified with special hardware functions in the machine and, in fact, may be directly attached to signal lines on an external interface on the machine. A functional block diagram of the general purpose registers and special purpose registers is shown in Figure 6-13.

General Purpose Registers

Register File R03. Register File R03 (R0, R1, R2, and R3) is a storage array of 16 four-byte words which are paged by bits 0-3 of the PS Register. The addressed word of the

register file can be referenced a byte at a time by separately specifying R0, R1, R2, or R3.

When R0, R1, R2 or R3 are specified, the corresponding eight bits are gated to the A Bus or loaded from the D Bus. The word operated upon in a microinstruction is determined by the contents of the PS Register at the end of the execution of the previous microinstruction.

Besides the ALU path, several special transfers can be made to and from R0, R1, R2, and R3 over the T Bus. When one of these special transfers takes place, with the exception of loading R1 and R2 from the T Bus, up to four bytes can be placed on the T Bus or loaded from the T Bus. These special transfers are as follows:

- Transferring data between the Buffer Data register (BUF) and R03.
- Transferring data between the Channel Counter (CC) and R23.
- Transferring data between the Data Transfer Counter (DTC) and R23.
- Transferring data between the four 1-byte registers G0, G1, G2 and G3 (G03) and R03.
- Transferring data between the Subroutine Register and R23.

G03 Register. The G03 Register is composed of the four one-byte registers G0, G1, G2 and G3. Each of these registers can be accessed individually through the A or B Bus and loaded individually from the D Bus. Additionally, data may be transferred to or from these registers simultaneously, utilizing the T Bus. These special transfers include the following:

- Transferring data between the Buffer Address Register (BAR) and G23.
- Transferring data between R03 and G03.

G4 Register. The G4 Register is a one-byte register which can be accessed through the A or BR Bus and can be loaded from the D Bus.

G5 and G6 Registers. The G5 and G6 Registers are one-byte registers which can be accessed through the A Bus and loaded from the D Bus.

G7 Register. The G7 Register is a one-byte register which can be accessed through the A or BR Bus and can be loaded from the D Bus.

Special Purpose Registers

INT Register. The INT Register is one byte wide and can be accessed through the A or BR Bus.

INT REGISTER BIT ASSIGNMENTS							
0	1	2	3	4	5	6	7
CHNL BYTE CNT ZERO	DATA XFER CNT ZERO	2860 MODE	CNT = 0	IDX	CHNL INT	RESP	CHK 2

Bit 0, Channel Byte Count Zero. This latch is set when the SCU Channel Interface has finished its data transfer. The Channel Byte Counter is decremented after each byte is transferred, either to or from the channel.

A byte at count one is transferred. When this transfer has been made, INT Register bit 0 is set and no further transfers are initiated until the latch is reset. The latch is reset each time the Channel Byte Counter Register is loaded.

Bit 1, Data Transfer Count Zero. This latch is set when the Data Transfer Counter is decremented from one to zero. The latch is reset each time the Data Transfer Counter is loaded.

Bit 2, 2860 Mode. This bit is set when OPERATIONAL IN is raised if the channel PCB selected (i.e., CA, CB, CC, or CD) has the 2860 Attachment Feature installed.

Bit 3, CNT =0. This bit is identical to the B(CNT = 0) branch condition. It is set when the counter incremented from FF to 00. It is reset when the counter is loaded (at Register Time).

Bit 4, Index. This bit is set at register time of an instruction if Select Alert 2 is active on the controller interface. This bit is reset by the control statement C(RESET. INDX) at Register Time.

Bit 5, Channel Interrupt. When the Operational In inbound channel tag is down, the state of the selection latch is gated to Bit 5 at Register Time. The selection latch is set during the initial-selection channel-sequence, when the control unit hardware detects that it is being selected. The selection latch state is not gated to bit 5 when Operational In is set.

When Operational In inbound channel tag is up, bit 5 is loaded at Register Time by the logical OR of the following conditions.

- Interface Disconnect. This occurs when the channel tag Address Out is set and the channel tag Select Out is reset.
- Truncation. This occurs when the channel responds to a Service In or Data In tag with the Command Out tag.

Bit 6, Response. This bit is set at register time of an instruction if Tag Valid, Normal End, or Check End are active on the controller interface. The bit is reset at register time of an instruction if Tag Valid, Normal End, and Check End are no longer active.

Bit 7, Check 2. This bit is set when a CK 2 condition occurs. Bit 7 is loaded at each register time from the CK2 latches. CK 2 latch is reset via the C(RESET.CKS) control statement at Register Time.

IMK Register. This IMK Register is one byte wide and can be loaded from the D Bus. The B(INT) branch condition is formed by the logical AND of the INT bits and corresponding bits set in the IMK Register.

CNT Register. The CNT Register is a one-byte register which is incremented at Register Time, and which can be accessed through the A Bus and loaded from the D Bus or T Bus. On any cycle which loads the CNT Register, incrementing the CNT Register is inhibited and the Counter Carry latch is reset.

The B(CNT = 0) branch condition is true if the Counter Carry latch is set. The Counter Carry latch is set at Register Time, if bits 0-7 of the CNT register are all 1s. The selected interrupt conditon of CNT = 0 operates identically to the B(CNT = 0) branch condition.

Page Select Register. The Page Select (PS) Register is one byte wide, and can be accessed through the A Bus and loaded from the D or T Bus.

PS REGISTER BIT ASSIGNMENTS							
0	1	2	3	4	5	6	7
PAGE SEL 8	PAGE SEL 4	PAGE SEL 2	PAGE SEL 1	BR MUX SEL 8	BR MUX SEL 4	BR MUX SEL 2	BR MUX SEL 1

Bits 0-3, Page Select. Bits 0-3 are used to address the Register File. The Register File word operated upon during a microinstruction's execution is determined by the value of the PS Register at the end of the previous microinstruction's execution. The PS Register is loaded at Register Time.

Bits 4-7, Branch Multiplex Select. Bits 4-7 are used to select the register to be multiplexed onto the BR Bus. The data on the BR Bus is loaded into a buffer at Register Time of a microinstruction's execution.

CCK Register. The CCK Register is a multiplex of two independent registers — the CHC Register and the MCK Register. MCM Register bit 4 controls which of these two registers is multiplexed onto the A Bus as CCK. When MCM bit 4 is a set, the CHC Register is multiplexed onto the A Bus and the MCK Register is degated.

MCK Register. The MCK Register is the collection of 47 latches which are set by specific error conditions (paragraph 6.6.1). The contents of these latches can be accessed eight bits at a time through the A Bus. The eight bits available at any point in time are controlled by the MCM Register. The error conditions associated with each latch are shown in Table 6-3. This table presents error conditions, along with the necessary states of the MCM Register, to access the conditions through the A Bus.

MCM Register. The MCM Register is one byte wide and can be loaded from the D Bus. The function of MCM is to control the error conditions available to the A Bus in the MCK Register. This register also controls the multiplexing of ECR pages to the A Bus, as well as the selection of A Bus entries from the CHC, SP, and MCK/FDF Registers. Specific assignments for bits 4-7 of MCM are also shown in Table 6-3.

MCM REGISTER BIT ASSIGNMENTS							
0	1	2	3	4	5	6	7
EN CK 2	NOT USED BIT	ECR MUX BIT 2	ECR MUX MUX 1	CCK/ SFD BIT	MCK MUX BIT 4	MCK MUX BIT 2	MCK MUX BIT 1

Subroutine (SR) Register. The SR Register is designed to be loaded with the present microinstruction address plus one when microinstructions containing the B(LD. SR.*+1) branch condition or containing the control statement C(LD.SR.* +1) are executed. Also the SR Register can be loaded directly into the microinstruction address register by specifying the branch condition B(ADR = SR). In order to provide nesting of subroutines the SR Register can be transferred to or from R2 and R3 of the selected scratchpad page via the T Bus.

NOTES

A description of the CC, CI, CHF, CT, CP, TR, BO and BI Registers will be found in paragraph 6.2.

A description of the DTC CO, DI, DO, and RWC Registers will be found in paragraph 6.4.

6.3.4.3 ARITHMETIC LOGIC UNIT

Operations

The arithmetic logic unit (ALU) is capable of performing eight arithmetic or logical functions on two eight-bit data bytes on the A and B Buses. Figure 6-14 shows a block diagram of the ALU. The result of the ALU operation is placed onto the D Bus, and for arithmetic operation, the carry-out is saved in the carry latch. The state saved in the carry latch can be tested with a branch allowing the sequence of microinstructions to be altered by the result of an arithmetic operation. The state of the carry latch is determined by the carry-out from bit 0 of the ALU. The D = 0 latch which is utilized by the branching circuitry is set only when all eight bits of the D Bus are zero. The D = FF latch, which is also utilized by the branching circuitry, is set only when all eight bits of the D Bus are all 1s.

TABLE 6-3. CHECK 1 MULTIPLEX ERROR CONDITIONS

MCM REGISTER BIT POSITION 4 5 6 7	MCK 0	MCK 1	MCK2	MCK 3	MCK 4	MCK 5	MCK 6	MCK 7
0 0 0 0	CHC PARITY ERROR	MULTIPLE WCS	A BUS	B BUS	T BUS	BRANCH	IMPL	ALU
0 0 0 1	BR	PS	TRANS-LATION	SUB-ROUTINE	WCS ECC	MULTIPLE BUFF	BUFF ECC	T-BUFFER ERROR
0 0 1 0	P1	P2	P3	P4	P5	P6	P7	P8
0 0 1 1	SEL RESET	SINGLE BUFF	SINGLE WCS	FA 12	FA 11	FA 10	FA 9	FA 8
0 1 0 0	FA7	FA 6	FA 5	FA 4	FA 3	FA 2	FA 1	FA 0
0 1 0 1	GA	GB	GC	GD	FD READ	FD SEEK	FD NOT READY	WRITE BUS

The ALU is actually two ALUs operating in parallel. The outputs of the two ALUs are compared and, if unequal, a machine check error condition is set.

Functions

Add (+). Arithmetic sum with carry-in of zero is performed on operands A and B. The carry latch is loaded when this function is executed, and is available for branching on the next machine cycle.

Add With A Carry (+1). Arithmetic sum with carry-in of one is performed on operands A and B. The carry latch is loaded when this function is executed and is available for branching on the next machine cycle.

Add With a Carry Determined by the Carry Latch (+C). Arithmetic sum with carry-in equal to the value of the carry latch is performed on operands A and B. The carry latch is loaded when this function is executed and is available for branching on the next machine cycle.

Shift A Left (L). A shift left operation is performed on operand A with the carry-in latch being set with the carry-out from the ALU from the previous instruction's execution. This function provides Rotate Left One Bit Position once the carry latch is loaded or Multiply-by-2 if the carry latch is zero.

Compare (K). Arithmetic sum with carry-in of 1 is performed upon operand A and the 1s complement of operand B. This function is a true subraction of operand B from operand A.

Carry latch is always loaded with the carry-out from bit 0 of the ALU when this function is executed. The result of the comparison (subtraction) is stored in two latches, which determine if A ≠ B and/or A < B. When this function is performed, if D Bus does not equal all zeros, the A ≠ B latch is set and the A < B latch is set with the complement of the carry-out from bit 0 of the ALU. Once the A ≠ B latch is set, the state of both comparison latches remain the same until reset with the RESET. K control command C(RESET. K). The comparison latches can be tested with the branch condition of -COMPARE. EQ and COMPARE. HI. The -COMPARE. EQ condition is the compare-unequal latch. The COMPARE. HI is the output of the A < B latch.

Exclusive-OR (*). Logical Exclusive-OR is performed upon operands A and B when this function is executed.

Logical OR (V). Logical OR is performed upon operands A and B when this function is executed. The carry latch is reset when this function is executed.

Logical And (.). Logical AND is performed upon operands A and B when this function is executed. The carry latch is reset when this function is executed.

6.3.5 Buffer Storage

6.3.5.1 GENERAL

The buffer is a 512-word semiconductor storage unit completely separate from the WCS with a word size of 39 bits. Thirty-two bits are used for data with the remaining seven bits being used for ECC checking and correction. The buffer continuously cycles in a Read operation, which is interrupted only when a write command is being executed. Minimum buffer-cycle time for a Read or Write operation is 480 nanoseconds.

The buffer is used to store the following types of information:

- Operation sequence and status for each drive.
- Usage and error log counters.
- Home Address, Count, and Key records from Read or Search operations.
- Corrected Home Address, Count, and Key records erroneously read during a Read or Search operation.
- Seek, File Mask, and Set Sector arguments for disconnected chains when the SCU is operating in a Multiple Request mode.
- Running status for inline diagnostic routines.

6.3.5.2 DESCRIPTION

A block diagram of the buffer storage is shown in Figure 6-15.

T Bus Register

This register is four bytes wide and, during operations involving the buffer, holds the data which is read on a Read cycle or written on a Write cycle.

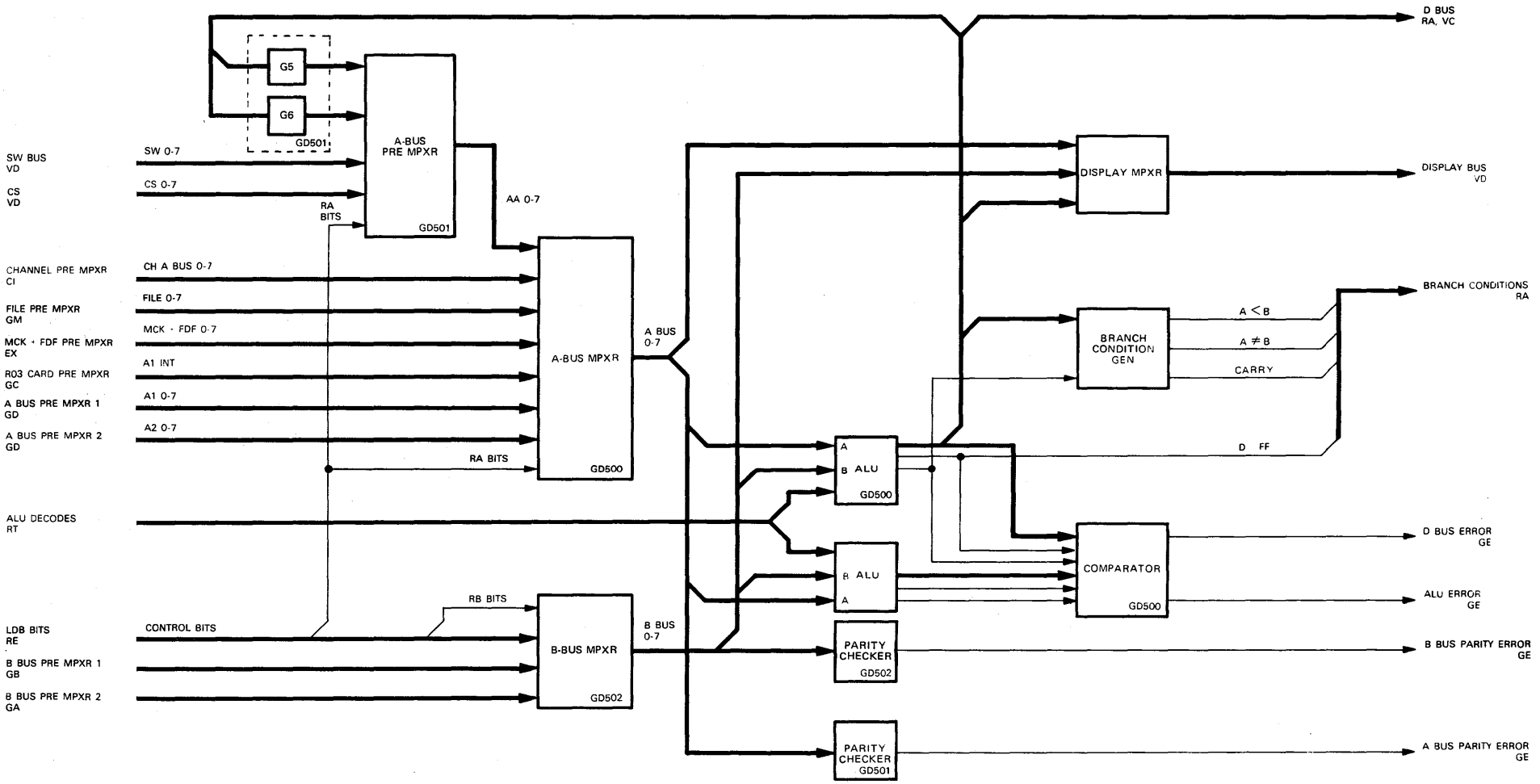


Figure 6-14. Buses and ALU Block Diagram

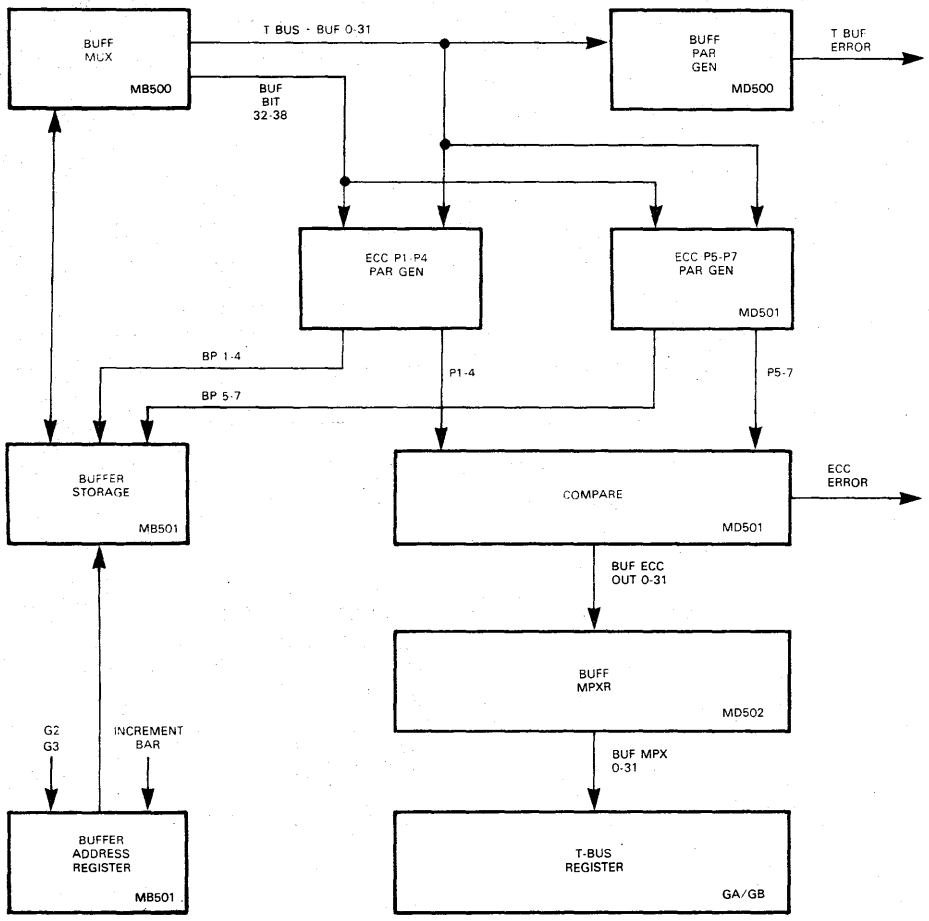


Figure 6-15. Buffer Storage Block Diagram

Buffer Address Register

This register is nine bits wide and holds the buffer address. Loading of this register or incrementing of this register will initiate a Read cycle on the buffer.

Error Correction

NOTE

The buffer storage ECC circuits are entirely separate from other WCS ECC circuits in the SCU.

All single bit data errors are detected and corrected by the ECC circuits. Multiple bit errors are detected but not corrected. Whenever a correctable data error is detected, the SCU microprogram is notified by the setting of a Check-2 error condition (MCK Register bit 1 with multiplex select of 3). Uncorrectable errors will cause a Check-1 error condition (MCK Register bit 5 with multiplex select of 1).

When data is written into the buffer, the ECC is generated and written into bit positions 32 through 38 of the storage location. When the data is read from the buffer, the data is again passed through an ECC syndrome generating circuit. If a correctable error is detected, three of the seven lines, P1 through P7 will be high and cause the error to be corrected.

6.3.5.3 BUFFER CONTROL COMMANDS

C(R03 = BUF) Command

Four bytes are transferred from the buffer (BUF) to registers R03. The output of the buffer is latched in the T Bus Register which in turn is gated into R03.

C(BUF = R03) Command

Four bytes of data in R03 are written into the Buffer Storage array. The T Bus Register holds the data while the Write cycle is performed. The location in the buffer is defined by the contents of the Buffer Address Register (BAR). The BAR is not modified. If this command is given before a previous Read cycle is completed, the cycle in progress is aborted and a new Write cycle is initiated.

C(BAR = G3) Command

Transfers the contents of the G3 Register to the BAR. The most significant bit of the BAR remains unchanged.

C(G23 = BAR) Command

Transfers the contents of the BAR to general purpose registers G2 and G3. Registers G0 and G1 remain unchanged.

C(BAR = G23) Command

Two bytes from general purpose registers G2 and G3 are transferred to the BAR and a Read cycle is initiated on the buffer. During the Read cycle, the data from the buffer storage location specified by the BAR is transferred to the buffer ECC. If this command is given and the buffer had not completed a previous cycle, the previous cycle is aborted and the execution of this command initiated. This is a Register Transfer operation and a Storage-Cycle-Read initiation command.

C(BAR + 1) Command

The BAR is incremented and a Read cycle is initiated on the BUF. This command has the characteristics of the BAR = G23 command.

C(R03 = BUF, LD) Command

This is a command which first executes the R03 = BUF sequence followed by a BAR = G3 sequence. The BAR = G3 sequence only loads the least significant eight bits of the BAR.

C(R03 = BUF, INC) Command

This command causes the register-to-register transfer R03 = BUF, a BAR increment (BAR = BAR +1), and a Read cycle initiation. Registers R03 are loaded with the contents of BUF before BUF is loaded with new data.

C(BUF = R03, LD) Command

This command first executes the BUF = R03 sequence followed by the BAR = G3 sequence.

C(BUF = R03, INC) Command

This command first executes the BUF = R03 sequence followed by the BAR = BAR +1 sequence.

6.3.5.4 ERROR CORRECTION CODE

The buffer storage ECC is generated using the matrix shown in Table 6-4. Bits 0 through 31 of the data word are examined in each horizontal row of the matrix. Each 1 bit of the data word is examined three times in the vertical rows of the matrix. If an even number of 1s is present, one of the P1 through P7 bits is turned on to make the count for that row odd. This scheme of bit examination along with comparison of the P1 through P7 bits written into the buffer allows a single data bit error (bit dropped or picked up) to be detected and corrected.

Example of ECC Generation

Assume the data word to be written into the buffer as shown in Table 6-5. Each data bit which is a 1 is applied to the ECC matrix. For instance, bit 0 which is a 1 in the example data word is noted in horizontal rows A, B, and C while bit 6 is noted in rows A, C, and F, and so forth.

In each horizontal row 1 bits are counted and if the count is even the ECC bit for that row is set. For instance, bits 0, 1, and 6 in row A give an even count so that ECC bit P1 is not set. In row C bits 0, 6, 9, and 19 are an even count; thus ECC bit P3 is set and so forth.

Example of ECC Correction

Using the matrix in the previous example, assume bit 6 is missing (dropped). As data is read from the buffer it is again examined and compared to the ECC code from the buffer. Any single bit error will cause three of the lines, P1 through P7, to be true. In the case of this example, P1, P3, and P6 will be true, which indicates an error associated with rows A, C, and F of the matrix. Examination of the ECC matrix shows that the only bit appearing in rows A, C, and F is bit 6. Therefore bit 6 was in error. At the proper time, the ANDed outputs of P1, P3, and P6 will be gated into the T Bus Register and the error will be corrected.

Parity

Parity for the data read from the buffer is generated by the ECC circuits and gated into the T Bus Register. Odd parity is used with a parity bit for each byte of the data word.

6.3.5.5 BUFFER STORAGE ALLOCATION

The tables shown in Figures 6-16 and 6-16A depict the Buffer Storage allocation for a typical subsystem.

TABLE 6-4. BUFFER STORAGE ECC MATRIX

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	P_1	P_2	P_3	P_4	P_5	P_6	P_7
A	1	1	1	1	1	1	1	1	1								1								1	1	1						1						
B	1	1	1	1					1	1	1	1	1	1	1	1		1										1	1					1					
C	1				1	1	1			1	1						1	1	1	1	1	1	1	1				1							1				
D		1			1							1	1						1	1	1				1	1	1	1	1	1	1	1				1			
E			1			1		1		1		1		1	1				1			1	1		1					1	1						1		
F				1			1	1			1			1		1				1		1		1		1			1	1		1					1	
G									1				1		1	1	1	1			1		1	1			1				1	1						1

TABLE 6-5. EXAMPLE DATA WORD APPLIED TO BUFFER STORAGE ECC MATRIX

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38
DATA WORD	1	1	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	P1	P2	P3	P4	P5	P6	P7
A	1	1					1																																
B	1	1								1																													
C	1						1			1										1															1				
D		1																		1																			
E										1																													
F							1													1																	1		
G																																							

Figure 6-16 shows the Buffer Storage allocation itself while Figure 6-16A defines the meaning of bytes and bits within selected words listed in Figure 6-16. Refer to the Glossary for definitions of abbreviations.

6.3.6 Flexible Disc Interface

6.3.6.1 GENERAL

The Flexible Disc Interface controls the data flow between the Flexible Disc and the Microprocessor. The Flexible Disc stores the microprogram and the diagnostic programs. The Flexible Disc Interface automatically generates Initial Program Load control signals to the Microprocessor and to the disc file whenever power is applied to the SCU.

A block diagram of the Flexible Disc Interface is shown in Figure 6-17. Two special-purpose eight-bit registers, Flexible Disc Data (FDD) and Flexible Disc Control (FDC), are used in the Flexible Disc Interface. Each register is loaded from the D Bus. The FDC Register defines the data contained in the FDD Register.

The SERIAL DATA from the Flexible Disc goes to the FD Shift Register where it is assembled into words, is applied to the Write Data Buffer, and then strobed out to the WCS and to the FDF MUX.

The Flexible Disc Interface also contains the logic to determine the status of the Flexible Disc. The status is applied to the FDF MUX, which channels the status via the A Bus to the Microprocessor.

Error Correction Code (ECC) circuits in the Flexible Disc Interface are used to check the ECC from the Flexible Disc and to generate an ECC for the WCS. The ECC circuits in the Flexible Disc Interface are used for all Read and Write operations associated with WCS. Details of ECC generation and checking may be found in paragraph 6.3.3.4.

6.3.6.2 FUNCTIONAL DESCRIPTION

The function description of the Flexible Disc Interface includes a brief description of each functional block, followed by an IPL (sequence of events) flow chart.

HEXA-DECIMAL ADDRESS	BUFFER STORAGE ALLOCATION
000	String 0 Drive 0 Data Word 0*
007	String 0 Drive 7 Data Word 0*
008	String 1 Drive 0 Data Word 0*
00F	String 1 Drive 7 Data Word 0*
010	String 2 Drive 0 Data Word 0*
017	String 2 Drive 7 Data Word 0**
018	String 3 Drive 0 Data Word 0*
01F	String 3 Drive 7 Data Word 0*
020	String 0 Drive 0 Data Word 1*
03F	String 3 Drive 7 Data Word 1*
040	String 0 Drive 0 Data Word 2*
05F	String 3 Drive 7 Data Word 2*

HEXA-DECIMAL ADDRESS	BUFFER STORAGE ALLOCATION
060	String 0 Drive 0 Data Word 3*
07F	String 3 Drive 7 Data Word 3*
080	String 0 Drive 0 Data Word 4*
09F	String 3 Drive 7 Data Word 4*
0A0	String 0 Drive 0 Data Word 5*
0BF	String 3 Drive 7 Data Word 5*
0C0	String 0 Drive 0 Data Word 6*
0DF	String 3 Drive 7 Data Word 6*
0E0	String 0 Drive 0 Data Word 7*
0FF	String 3 Drive 7 Data Word 7*

HEXA-DECIMAL ADDRESS	BUFFER STORAGE ALLOCATION
100	Not Used
10F	
110	String 0 Channel A Interrupt Word*
113	String 0 Channel D Interrupt Word*
114	String 1 Channel A Interrupt Word*
117	String 1 Channel D Interrupt Word*
118	String 2 Channel A Interrupt Word*
11B	String 2 Channel D Interrupt Word*
11C	String 3 Channel A Interrupt Word*
11F	String 3 Channel D Interrupt Word*
120	Sense Bytes 0-23
125	

HEXA-DECIMAL ADDRESS	BUFFER STORAGE ALLOCATION
126	Counter Unit Check Owed Word*
127	Log Control Block*
128 129	Channel Overrun Counters A and B Channel Overrun Counters C and D
12A 12B	Home Address Buffer
12C	Byte Displacement Accumulated
12D	ECC Pattern
12E 130	Not Used
131 136	Log Sense Bytes 0-23

HEXA-DECIMAL ADDRESS	BUFFER STORAGE ALLOCATION
137	Restart Displacement
138 139	Retry Parameters*
13A 13B 13C	Count Buffer
13D 17E	Key Buffer
17F	Data Buffer
180 18F	Diagnostic Status and Control Block
190	Basic Wait Loop Timer
191 IFF	Diagnostic Scratch

*See Figure 6-16A for definitions of bytes within word.

FIGURE 6-16. BUFFER STORAGE ALLOCATION AND CONTENTS

HEX ADDRESS RANGE	BYTE 0								BYTE 1								BYTE 2								BYTE 3							
	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7
000-01F	LOGICAL DRIVE ADDRESS								LAST PHYSICAL ADDRESS								DE OWED				PACK CHANGE DE OWED				FILE MASK							
				String 2	String 1	Device 4	Device 2	Device 1	CU 0	CU 1	0	1	2	3	4	5	D	C	B	A	D	C	B	A	Write Mask 1	Write Mask 2	SFM in Chain	Seek Mask 1	Seek Mask 2	Diag. Write Mask	Alt. Auto Pad	PCI Fetch Mask
020-03F	CHANNEL DEVICE STATUS								DRIVE STATUS 1								BIT SIGNIFICANT DEVICE ADDRESS								CHANNEL STATUS							
	Reserved	Chaining	Unit Check	SIP or SSIP	RPS 2	RPS 1	Chnl 2	ID 1	Index Error	Offset Active	Seek Incomplete	Seek Complete	Online	Attention	Busy	Sector Ready	0	1	2	3	4	5	6	7	Stacked	SM	CUE	Busy	CE	DE	UC	UE
040-05F	HIGH CYLINDER								LOW CYLINDER								HIGH CYLINDER			HEAD ADDRESS					DEVICE TYPE							
							512	256	128	64	32	16	8	4	2	1		512 DD 256 SD	256 DD	16	8	4	2	1							Double Density	
060-07F	BYTES READ								BYTES READ								BYTES READ								BYTES READ							
	2,147,483,648	1,073,741,824	536,870,912	268,435,456	134,217,728	67,108 864	33,554,432	16,777,216	8,388,608	4,194,304	2,097,152	1,048 576	524,288	262,144	131,072	65,536	32,768	16,384	8,192	4,096	2,048	1,024	512	256	128	64	32	16	8	4	2	1
080-09F	CORRECTABLE DATA CHECKS								CORRECTABLE DATA CHECKS								RETRY DATA CHECKS								RETRY DATA CHECKS							
	32 768	16.384	8.192	4.096	2.048	1.024	512	256	128	64	32	16	8	4	2	1	32,768	16,384	8,192	4.096	2,048	1.034	512	256	128	64	32	16	8	4	2	1
0A0-0BF	SEEKS								SEEKS								(NOT USED)								SEEK ERRORS							
	32,768	16,384	8,192	4,096	2,048	1,024	512	256	128	64	32	16	8	4	2	1									128	64	32	16	8	4	2	1
0C0-0DF	(NOT USED)								(NOT USED)								(NOT USED)								(NOT USED)							
0E0-0FF	(NOT USED)								(NOT USED)								(NOT USED)								(NOT USED)							
110-113	POLL ENABLE, STRING 0 [1]								DISCONNECT COMMAND CHAINING, STRING 0 [2]								(NOT USED)								DEVICE END READY, STRING 0 [3]							
	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7									0	1	2	3	4	5	6	7
114-117	POLL ENABLE, STRING 1 [1]								DISCONNECT COMMAND CHAINING, STRING 1 [2]								(NOT USED)								DEVICE END READY, STRING 1 [3]							
	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7									0	1	2	3	4	5	6	7
118-11B	POLL ENABLE, STRING 2 [1]								DISCONNECT COMMAND CHAINING, STRING 2 [2]								(NOT USED)								DEVICE END READY, STRING 2 [3]							
	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7									0	1	2	3	4	5	6	7
11C-11F	POLL ENABLE, STRING 3 [1]								DISCONNECT COMMAND CHAINING, STRING 3 [2]								(NOT USED)								DEVICE END READY, STRING 3 [3]							
	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7									0	1	2	3	4	5	6	7
126	UNIT CHECK OWED, STRING 0 [4]								UNIT CHECK OWED, STRING 2 [4]								UNIT CHECK OWED, STRING 2 [4]								UNIT CHECK OWED, STRING 3 [4]							
	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7

FIGURE 6-16A. BUFFER STORAGE BYTE DEFINITIONS (1 of 2)

HEX ADDRESS RANGE	BYTE 0								BYTE 1								BYTE 2								BYTE 3							
	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7
127	LOG CONTROL BLOCK								LOG CONTROL BLOCK								LOG CONTROL BLOCK								LOG CONTROL BLOCK							
	UC. SIO Log	Done Log [5]	String 4	String 2	String 1	Drive 4	Drive 2	Drive 1	Format 8	Format 4	Format 2	Format 1	Mssg 8	Mssg 4	Mssg 2	Mssg 1								Log Sense							Log Count 2	Log Count 1
138	RETRY SECTOR VALUE								CURRENT OFFSET								(NOT USED)								DATA CHECK RETRY COUNT							
	128	64	32	16	8	4	2	1	128	64	32	16	8	4	2	1									128	64	32	16	8	4	2	1
139	COMMAND OVERRUN COUNTER								(NOT USED)								DATA OVERRUN COUNTER								SEEK RETRY COUNTER							
	128	64	32	16	8	4	2	1									128	64	32	16	8	4	2	1	128	64	32	16	8	4	2	1

[1] When bit is set to a 1, numbered device is available to this channel.

[2] When bit is set to a 1, numbered device has Disconnect Command Chaining in progress for this channel.

[3] When bit is set to a 1, Device End is owed to this channel due to a previously presented Device Busy status (generated by the SCU) or due to a Pack Change Interrupt (multi-tag mode).

[4] When bit is set to a 1 Unit Check is owed for this device.

[5] Bit set to a 1 when count equals 4, reset to a 0 when logging is initiated.

FIGURE 6-16A. BUFFER STORAGE BYTE DEFINITIONS (2 of 2)

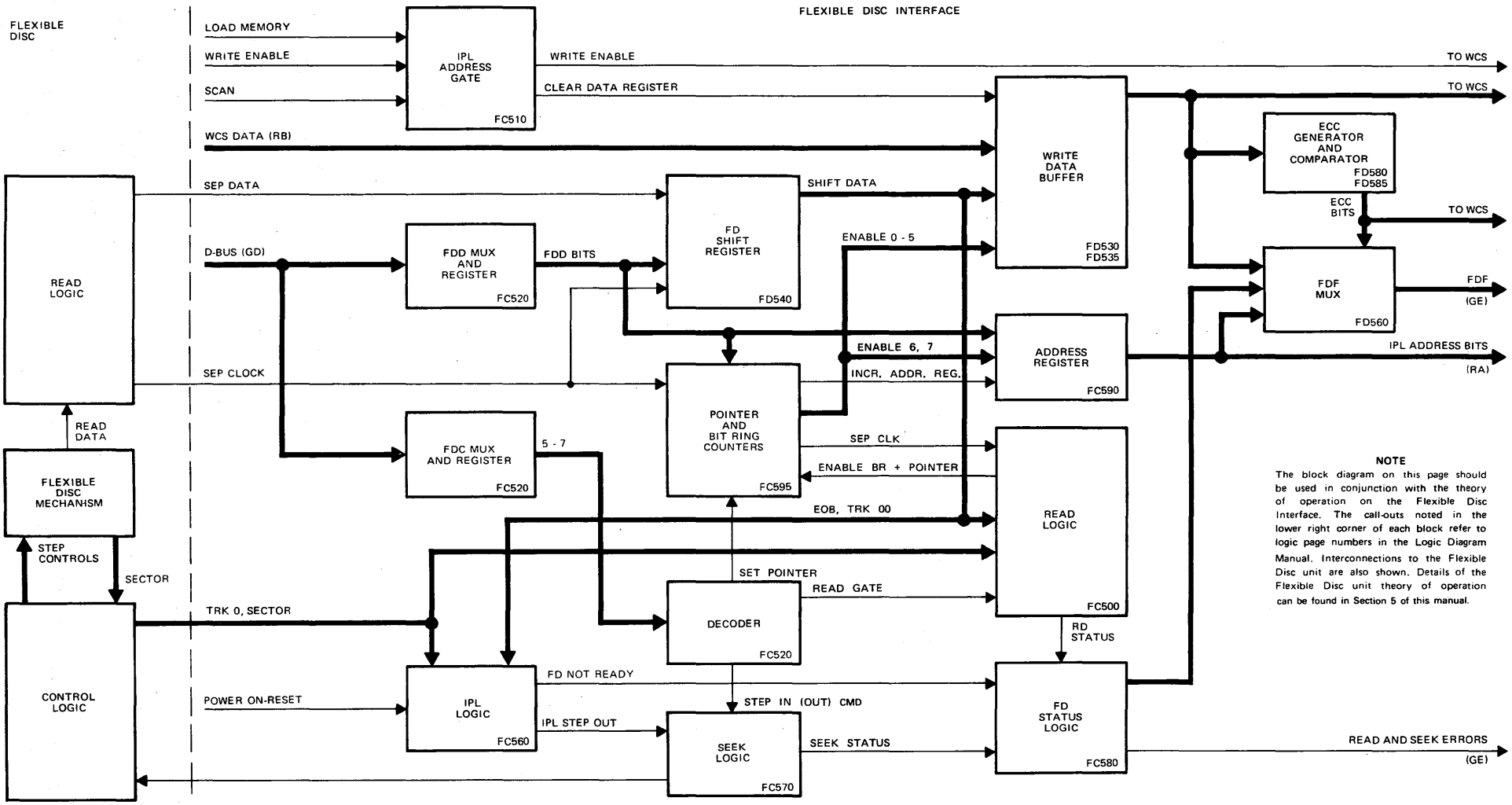


Figure 6-17. Flexible Disc Interface Block Diagram

FDD Register

The FDD Register is loaded from the D Bus. The output of the register is routed to the FD Shift Register, the FD Pointer, or the Address Register. The destination of the FDD Register output is controlled by the FDC Register.

FDC Register

The FDC Register is loaded from the D Bus by raising FD = FDC. The output of the FDC controls the FDF MUX, the Seek Logic, the Read Logic, the Address Register and the FD Pointer which in turn controls the data strobed through the Write Data Buffer.

FDC REGISTER BITS							
0	1	2	3	4	5	6	7
FDF MUX 0	FDF MUX 1	FDF MUX 2	POWER ON TO FD	ECC BIT SELEC-TION	FDC MUX 0	FDC MUX 1	FDC MUX 2

FDC Reg. bits 0-2; FDF MUX. These three bits enable reports through the FDF multiplexer. Loaded from the D Bus by raising RD = FDC.

FDC Reg. bit 3; Power On to FD. When set to 0, this bit enables the FD to be powered on. When set to 1, this bit enables the FD to be powered off. Loaded from the D Bus by raising RD = FDC.

FDC Reg. bit 4; ECC Bit Selection. When set to 0, enables ECC bits generated by the FD interface logic to be loaded into WCS. This bit is set to 0 during normal operation. It is set to 1 only during error diagnostics.

FDC Reg. bits 5-7; Decodes. These bits go to a BCD-to-decimal decoder from the D Bus which generates eight decodes. The eight decodes are defined as follows:

- FDCMUX = 0

  No operation.

- FDCMUX = 1 — Set Pointer

  Data bits 5, 6, 7 of FDD will be transferred into the pointer (FDP). The pointer values correspond to the following:

  FDP = 0: bits 32-39 of Write Data Buffer
  FDP = 1: bits 40, 41 of Write Data Buffer
  FDP = 2: bits 0-7 of Write Data Buffer
  FDP = 3: bits 8-15 of Write Data Buffer
  FDP = 4: bits 16-23 of Write Data Buffer
  FDP = 5: bits 24-31 of Write Data Buffer
  FDP = 6: bits 0-7 of Address Register
  FDP = 7: bits 8-11 of Address Register

- FDCMUX = 2 — Set Data

  The data in FDD will be loaded into the area corresponding to the pointer value.

- FDCMUX = 3 — Start Micro Read

  The Read logic will be activated and data will be transferred from the FD to the Write Data Buffer until EOB is received. This condition is latched up and is reset with either END-OF-BLOCK or RELEASE.

- FDCMUX = 4 — Step In

  The FD will move the access mechanism one track away from track 0. This condition is latched up and is reset with SEEK COMPLETE.

- FDCMUX = 5 — Step Out

  The FD will move the access mechanism one track toward track 0. This condition is latched up and is reset with SEEK COMPLETE.

- FDCMUX = 6 — Reset Word Ready

  De-activate the word ready indicator. Failure to execute this instruction within 38 microseconds after Word Ready indicator is activated will result in an overrun condition.

- FDCMUX = 7 — Release

  The FD interface is available to the second processor of a dual processor machine, This does not cause the FD to power down.

FD Read Logic

The Read logic is the hardware control for loading the bootstrap section of IPL. This logic also controls the loading of the rest of IPL and diagnostics by way of control signals from the Microprogram.

FD Seek Logic

The Seek Logic causes the FD to automatically seek to track 000 during bootstrap and also to step in or out to any track called out by the microprogram.

IPL Logic

The IPL (Initial Program Load) Logic and read Gate contains the logic required for initial program loading.

An IMPL error is detected in the IMPL Error Logic and is an indication of the following:

- Read Overrun as described in FDF bit 5.
- Seek Time Out — Seek not completed within one second.
- Read Time Out as described in FDF bit 5.
- Sixteen occurrences of either or combinations of the following:
  a. IPL Seek Error — no track = 0 decode after two sync bytes during IPL.
  b. Data Bus Parity — parity error detected while reading from the FD.

FD Shift Register

The FD Shift Register takes serial data from the flexible disc and assembles it into bytes to be loaded into the Write Data Buffer a byte at a time. The Shift Register can also accept a parallel load of one byte of data from the FDD register.

Pointer And Bit Ring

The pointer points to a particular section of the Write Data Buffer (enable 0-5) into which a single byte of data will be loaded. During bootstrap IPL, the bit ring counts bits of data off the Flexible Disc and increments the pointer for each byte of data to sequentially load, a byte at a time, a complete data word (42 bits) into the Write Data Buffer. The output, Address Bits 0-11, are routed to the WCS Address Register and to the FDF MUX.

Address Register

The Address Register contains the address where data, from the FD, will be loaded into memory. During IPL, the Address Register is incremented for each data word, starting at address 000. For other operations the address comes from the FDD Register.

Write Data Buffer

The Write Data Buffer temporarily stores, and makes available to WCS, the data to be written into memory. It also stores data out of memory during a read control store. The output is also available to the A Bus through the FDF MUX.

FD Status Logic

The FD Status and IMPL Error Logic accepts signals from throughout the Flexible Disc Interface logic circuits, and generates ready or error signals which are sent out to the Microprocessor via the FDF MUX.

ECC Circuit

The ECC generator accepts data bits during a read operation and generates an EC code. The code is compared with the ECC initially written into the disc during a write operation. If the ECC compares, the code is written into memory either from the ECC generator or from the Flexible Disc, depending on the state of bit FDC 4. A noncomparison results in a Read Data Check signal to the FD status logic.

FDF MUX

The FDF MUX makes data available to the A-Bus. This data can be either Write Control Store data, IPL Address Bits, or Status Bits denoting the status of the FD interface. The data is multiplexed through a byte at a time by FDC bits 0, 1, and 2. A chart depicting the FDF MUX output bytes is shown in Table 6-6. A description of the FD status bits (FDC 0, 1, 2 = 0) is provided in the following text.[300]

FDF MUX = 0, FDF Bit 0 (Overrun). Word ready was not reset before new data was loaded into the Write Data Buffer.

TABLE 6-6. FDF MUX OUTPUT BITS

FDF MUX OUTPUT BITS	0	1	2	3	4	5	6	7
FDC 0, 1, 2 = 0	OVERRUN	SEEK COMPL	BUSY	WORD READY	FD SEEK READY	FD READ ERROR	FD NOT READY	EOB
FDC 0, 1, 2 = 1	$\overline{A}_7$	$\overline{A}_6$	$\overline{A}_5$	$\overline{A}_4$	$\overline{A}_3$	$\overline{A}_2$	$\overline{A}_1$	$\overline{A}_0$
FDC 0, 1, 2 = 2	D_{40}	D_{41}	(Sctr 2 only) D_{42}	$\overline{A}_{12}$	$\overline{A}_{11}$	$\overline{A}_{10}$	$\overline{A}_9$	$\overline{A}_8$
FDC 0, 1, 2 = 3	D_{32}	D_{33}	D_{34}	D_{35}	D_{36}	D_{37}	D_{38}	D_{39}
FDC 0, 1, 2 = 4	D_{24}	D_{25}	D_{26}	D_{27}	D_{28}	D_{29}	D_{30}	D_{31}
FDC 0, 1, 2 = 5	D_{16}	D_{17}	D_{18}	D_{19}	D_{20}	D_{21}	D_{22}	D_{23}
FDC 0, 1, 2 = 6	D_8	D_9	D_{10}	D_{11}	D_{12}	D_{13}	D_{14}	D_{15}
FDC 0, 1, 2 = 7	D_0	D_1	D_2	D_3	D_4	D_5	D_6	D_7

FDF MUX = 0, FDF Bit 1 (Seek Complete). SEEK COMPLETE is normally up unless a Seek is initiated by raising STEP IN or STEP OUT. SEEK COMPLETE will come back up, allowing another Seek to be initiated, within a minimum of 20 milliseconds and a maximum of 40 milliseconds after a Seek is initiated.

FDF MUX = 0, FDF Bit 2 (Busy). After executing an FDC = 00 the busy bit will be on if the FD Interface is connected to the other processor of a dual processor system.

FDF MUX = 0, FDF Bit 3 (Word Ready). WORD READY is active when the data on the write bus is stable and ready for loading into the Writable Control Store.

FDF MUX = 0, FDF Bit 4 (FD Seek Error). An FD Seek Error is an indication of the following:

- A Track = 0 Decode was not found after two successive Sync Byte decodes during IMPL.
- A Seek was initiated and was not completed within one second.

FDF MUX = 0, FDF Bit 5 (FD Read Error). An FD Read Error is an indication of one of the following conditions:

- A Data Bus Parity error was detected during a read from the FD.
- An overrun condition occurred; i,e., WORD READY is still up when the next byte of data from FD is ready to be loaded into the Write Bus Data Buffer during a read.
- A read from the FD was initiated and a sync byte (32, 32) was not detected in the read data within one second.

FDF MUX =, FDF Bit 6 (FD Not Ready). Power on signal is activated to PDU but FD is not operational. This line is up normally for 5 seconds.

FDF MUX = 0, FDF Bit 7 (EOB). The following conditions will reset EOB (End of Block):

- A machine reset
- Initiating a Microprogram read
- Release

FDF MUX = 1 through 7. These bits control the following:

- A0 through A11: Address of Writable Control Storage into which write bus data will be written.
- DO through D41: Write bus data.

Track Format

The FD track format, depicting Sync Bytes, Record ID Bytes and Data Bytes is shown in Figure 8-16.

6.3.6.3 IMPL FLOW CHART

The IMPL flow chart of Figure 6-19 depicts the sequence of events which occur in the Flexible Disc Interface logic. The flow chart also shows timing (i.e., leading edge of Sector Pulse), and references the appropriate logic diagram page number.

6.3.7 Microprocessor Timing

Microprocessor timing for the SCU is generated by a 25-MHz crystal controlled oscillator. The output of the oscillator is fed to a six-stage ring counter whose outputs, A through F, furnish the basic timing pulses for the SCU. See Figure 6-20.

The nominal timing shown in Figure 6-23 and detailed timing shown in Figures 6-22 and 6-21 show the relationship of basic machine timing to major timing signals. Each major timing pulse is 40 nanoseconds wide resulting in a complete timing cycle of 240.

Except where noted on the timing diagrams, the trailing edge of the clock pulses are used to set the logic.

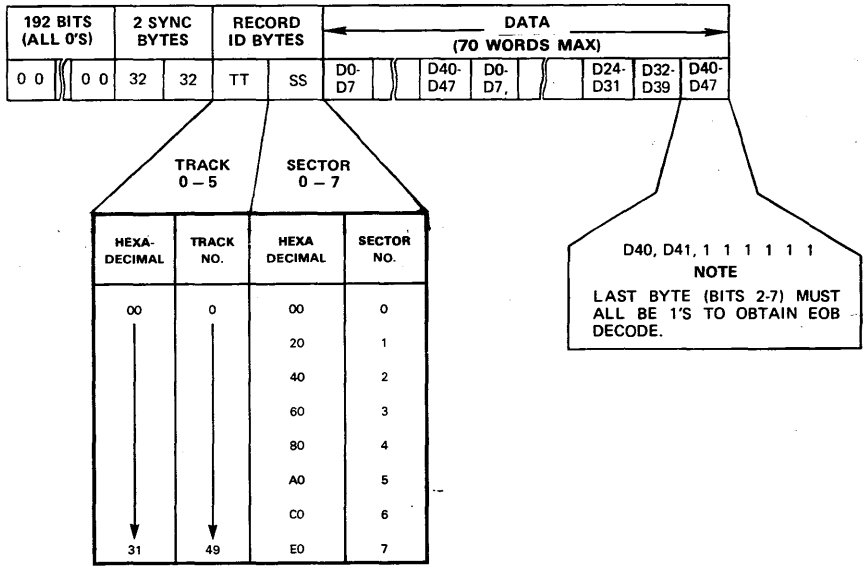


Figure 6-18. FD Track Format

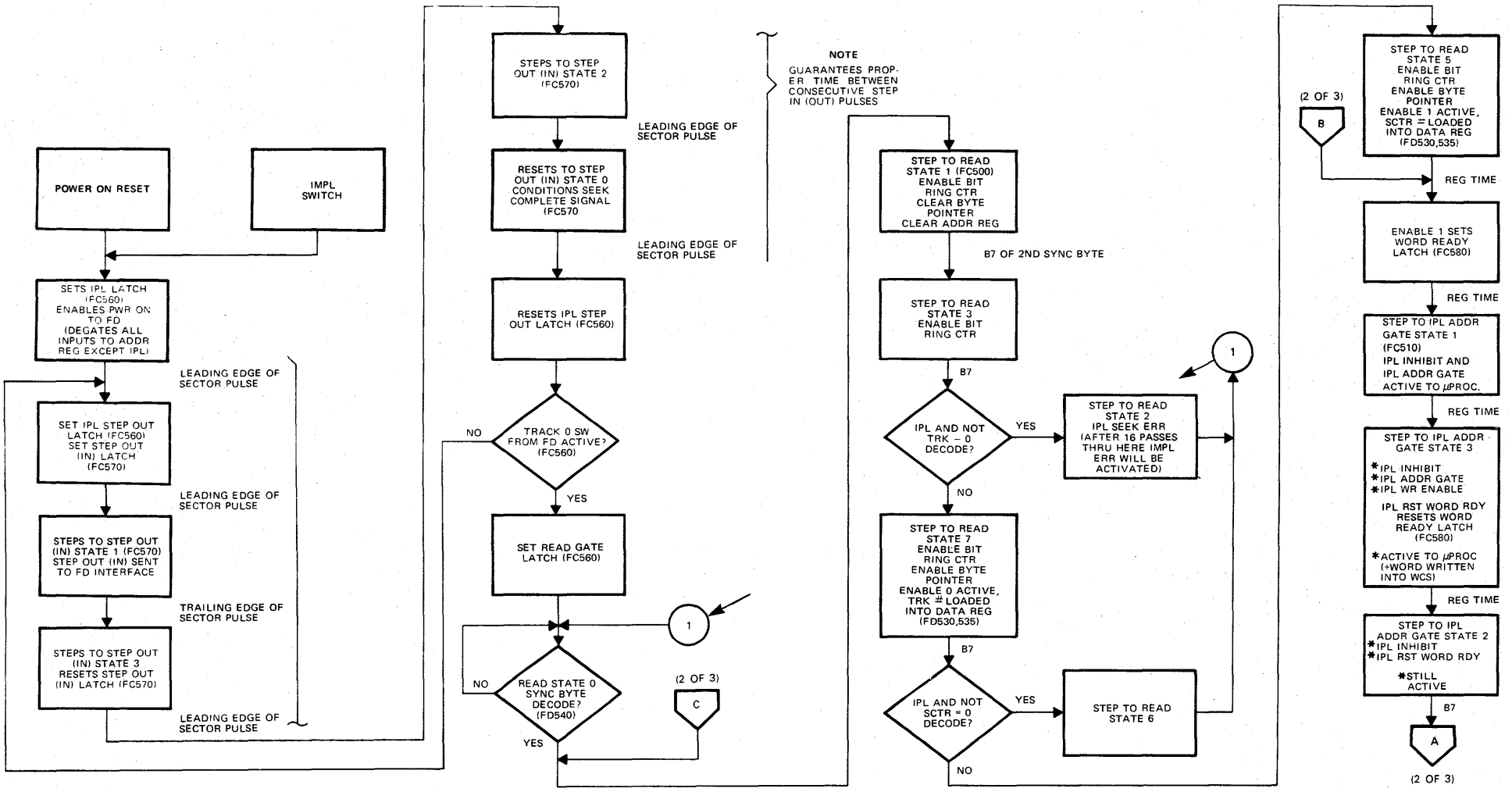


Figure 6-19. IPL Flow Chart (1 of 3)

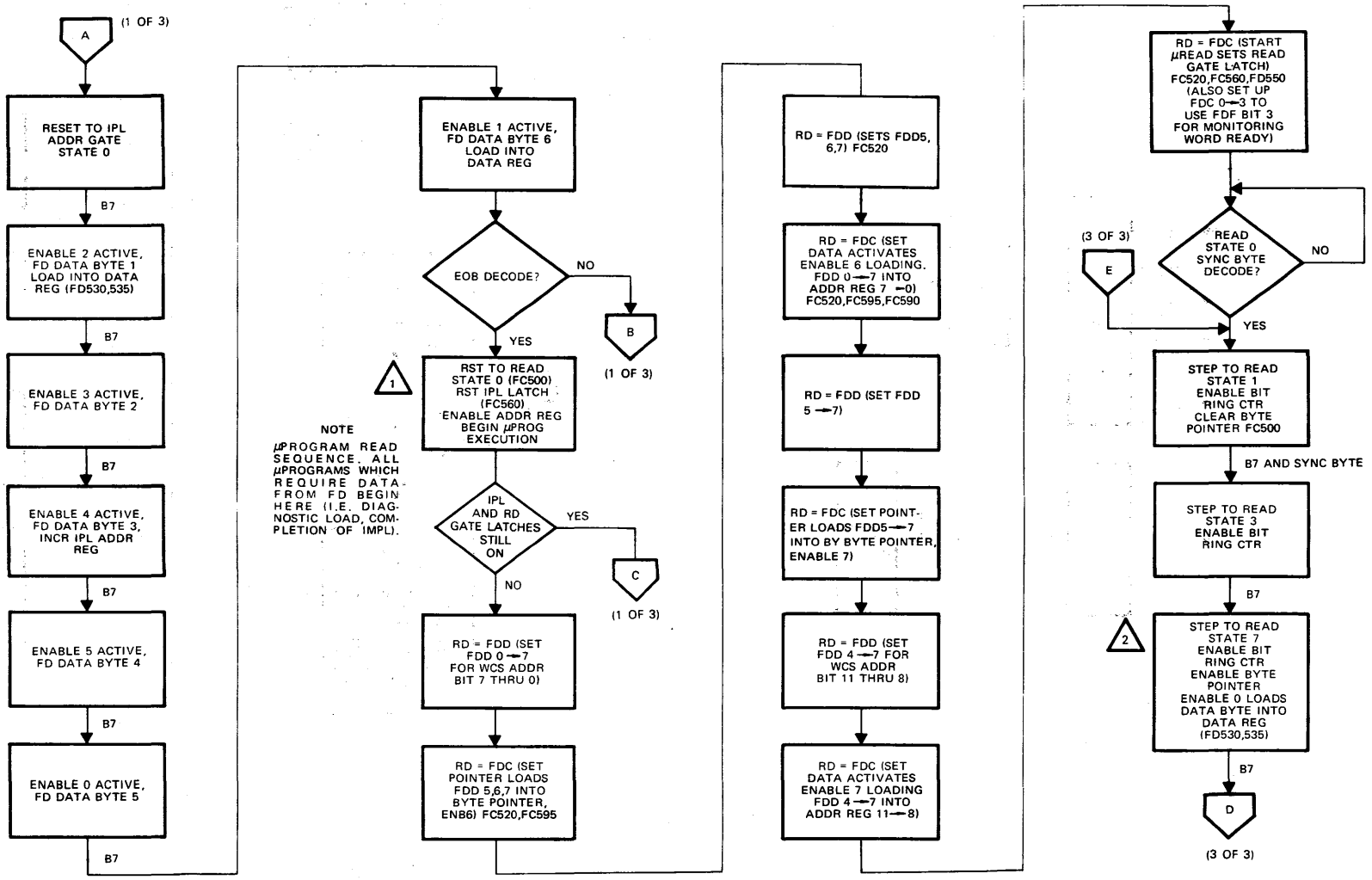


Figure 6-19. IPL Flow Chart (2 of 3)

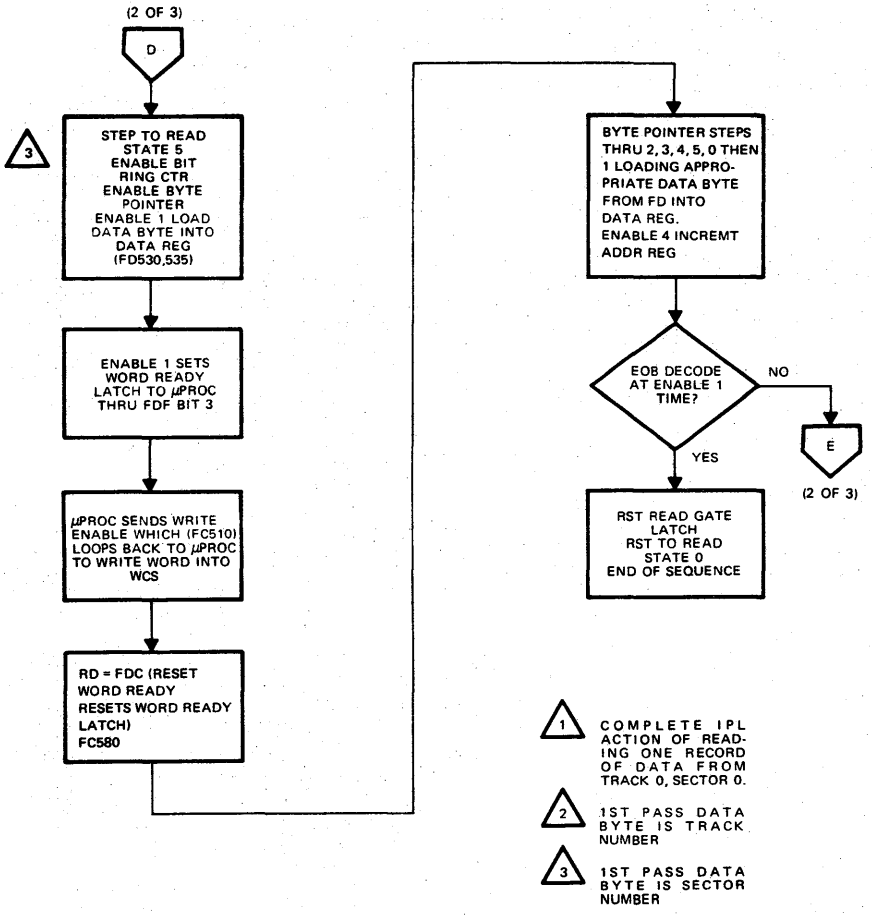


Figure 6-19. IPL Flow Chart (3 of 3)

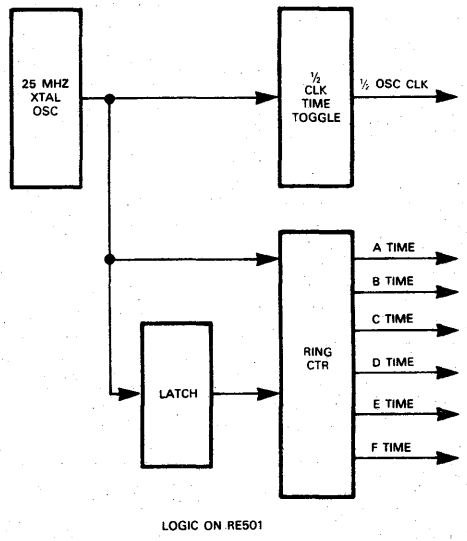


Figure 6-20. Microprocessor Timing Logic

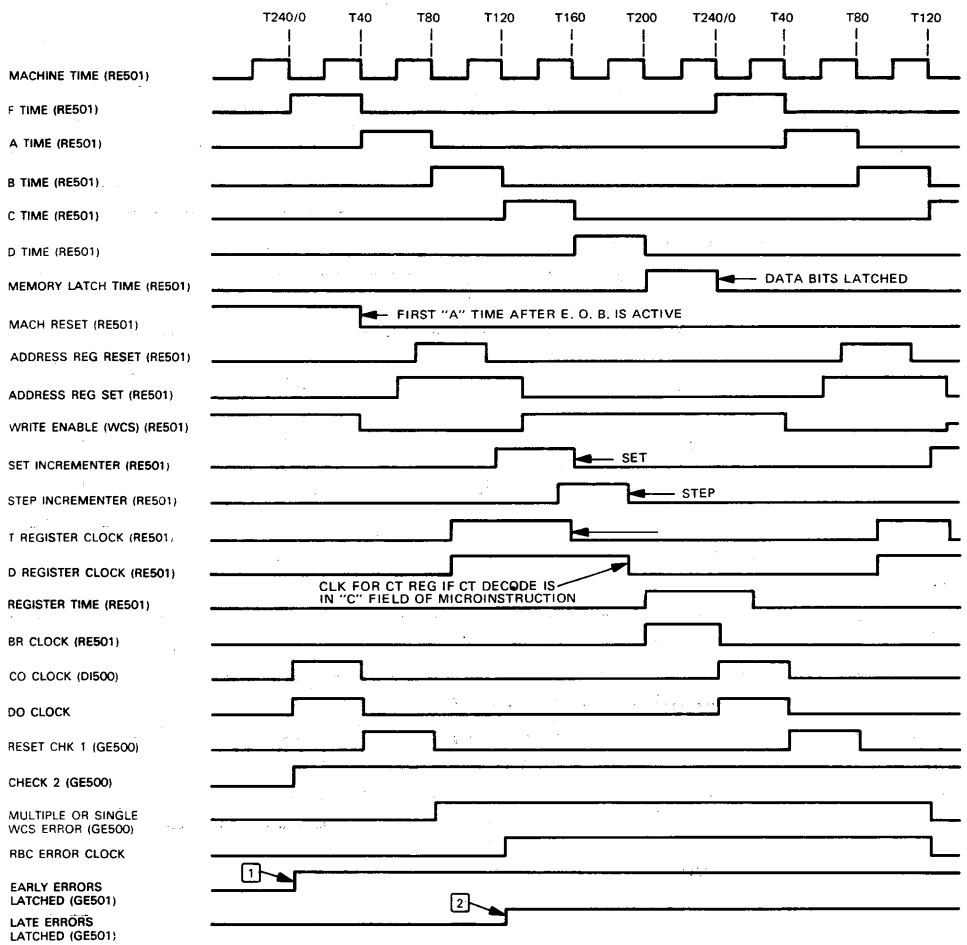


Figure 6-21. Nominal Microprocessor Timing

NOTES

1. Shaded areas on each side of the waveforms indicate allowable timing variations under worse case conditions.
2. Reset always follows set on leading edge by ≤ 10 nsec and falls ≤ 10 nsec before set falls.

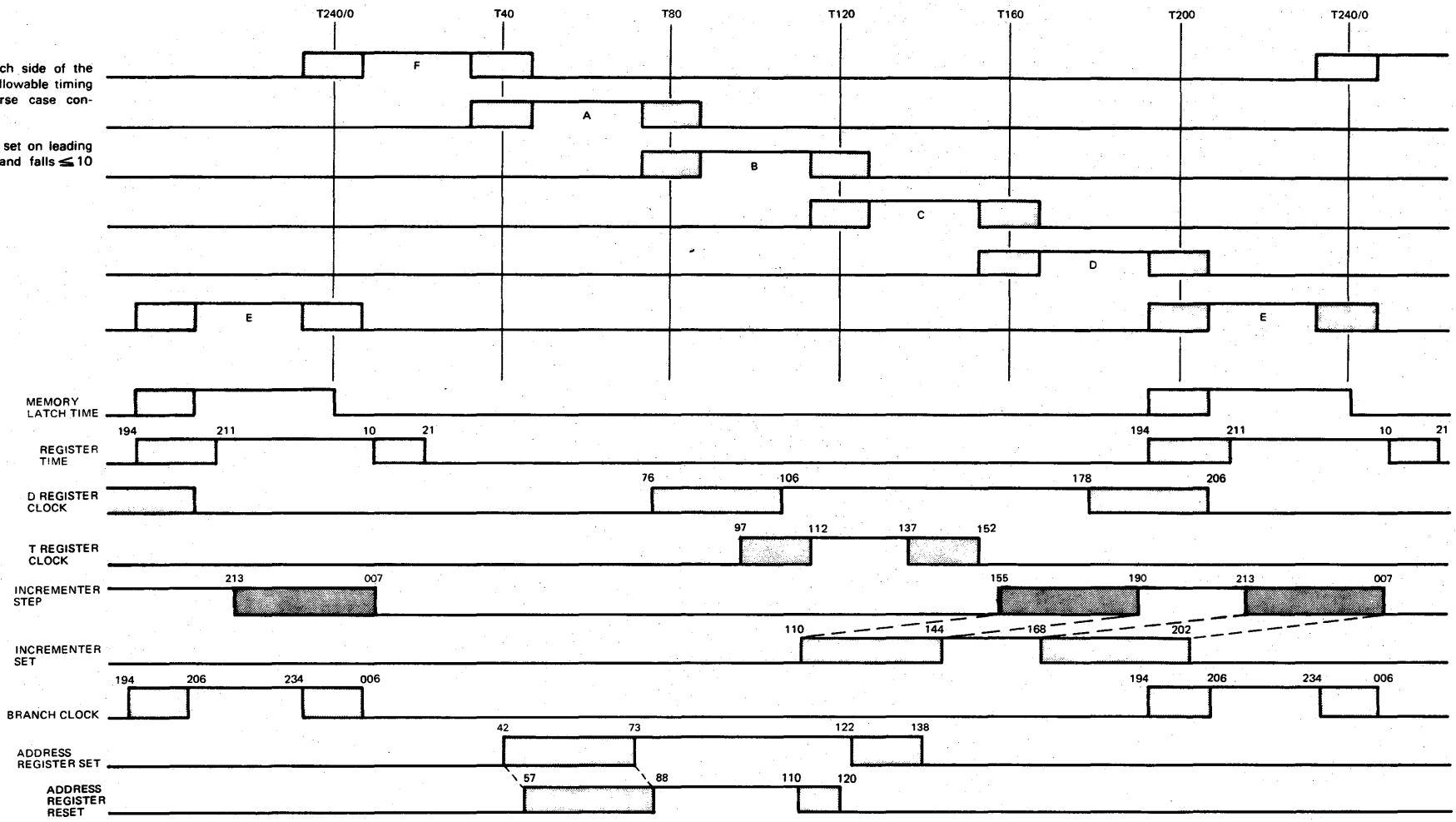


Figure 6-22. Detailed Microprocessor Clock Timing

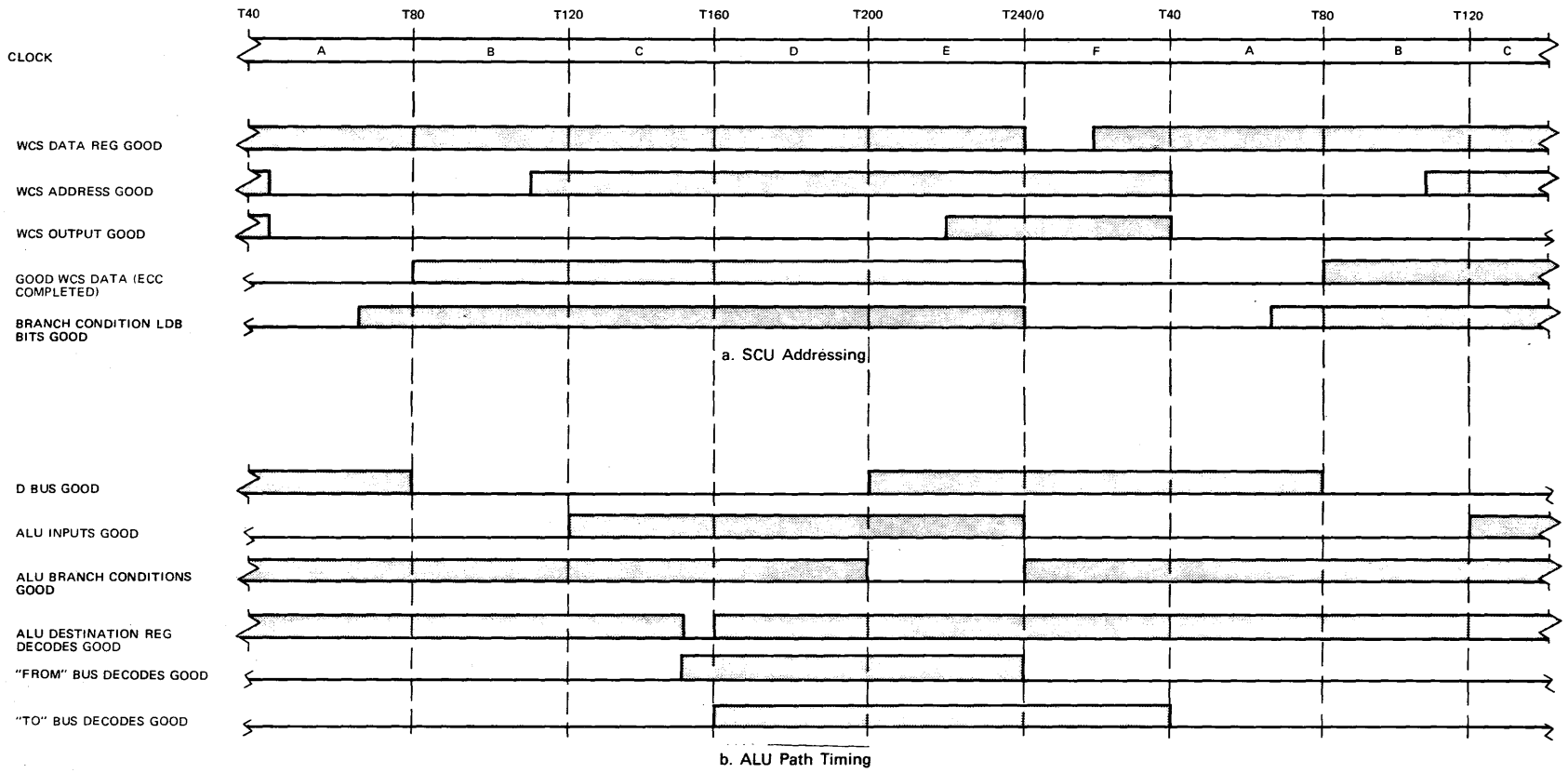


Figure 6-23. Detailed Microprocessor SCU Addressing and ALU Path Timing

6.4 CONTROLLER INTERFACE

6.4.1 General

The Controller interface (CTL-I) consists of special-purpose registers, counters and support hardware organized to perform the sequencing and data transfer functions between the SCU and the Controller. A block diagram of these elements is shown in Figure 6-24. The CTL-I is under direct control of the SCU Control Program. Data transfer operations within the SCU are initiated when the CPU transfers a command to the SCU requiring such a transfer. Data transfers between the Controller and the SCU are initiated by read and write commands:

a. Read Commands

Read commands are those commands which cause data to be transferred from the Controller to the SCU.

b. Write Commands

Write commands are those commands which cause data to be transferred from the SCU to the Controller.

During read and search operations, read data bits are transferred from the selected Disc Drive through the Controller to the SCU. The data bits are serially transferred from the Disc Drive to the Controller. The Controller transforms the serial data into parallel by bit/serial by byte data and transferred to the SCU for subsequent transfer to the System/370 Channel or for comparison with data from the System/370 Channel. The Microprocessor has access to the assembled data bytes when the control program addresses the Data Buffer Register.

Logic for the CTL-I is located on two PCBs. The DC board contains the major portion of the CTL-I circuitry such as registers, error checking logic, and line drivers and receivers. The GM board contains the general multiplexing logic to route data from a selected CTL-I register to the ALU via the A bus. This board also contains the Data Transfer Counter.

6.4.2 Interface Signals

Data and control signals transmitted between the SCU and Controller are buffered in the SCU by means of the line drives and receivers shown in Figure 6-25.

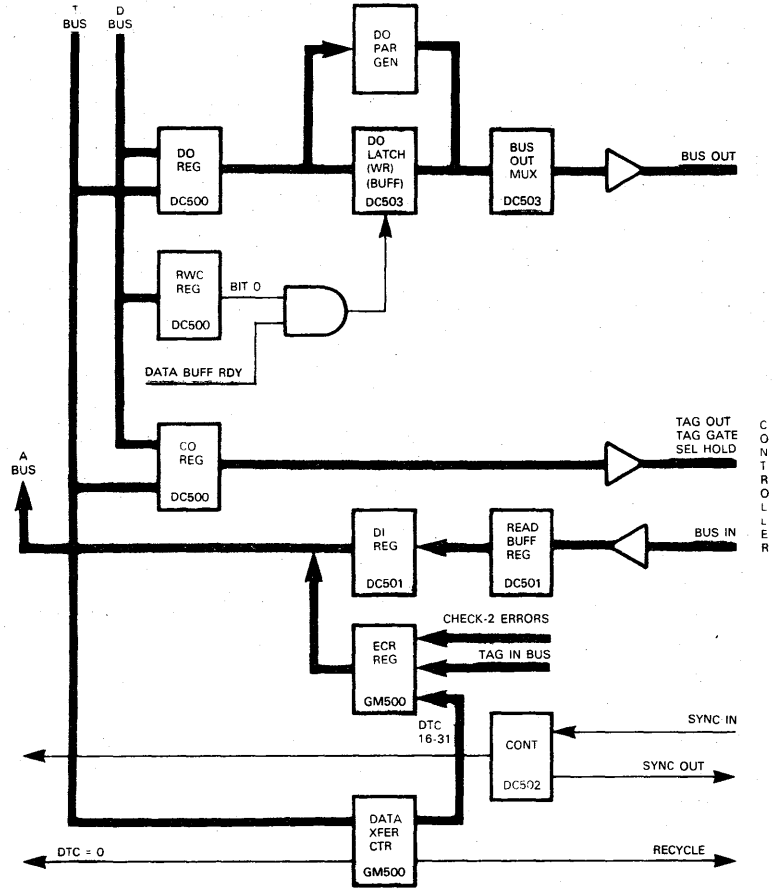


Figure 6-24. Controller Interface Block Diagram

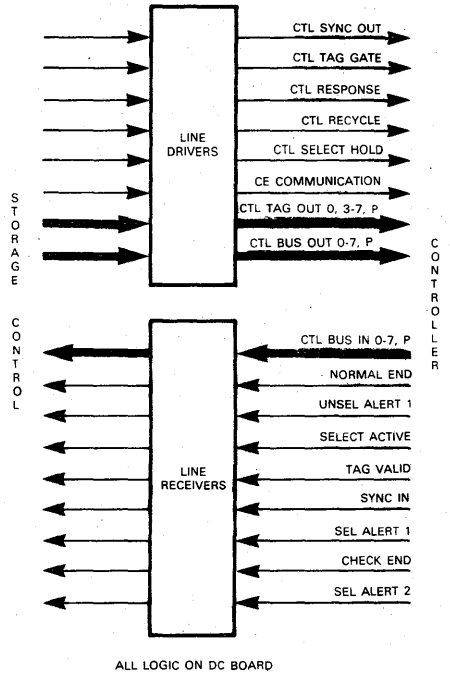


Figure 6-25. Storage Control Interface Line Drivers and Receivers

6.4.2.1 SCU TO CONTROLLER SIGNALS

Bus Out

The Bus Out lines are used for two purposes. The lines carry tag modifier, control, or address information when deskewed and validated by TAG GATE. The lines transmit data from the SCU to the Controller when deskewed and validated by SYNC OUT. The Bus Out lines consist of eight data lines plus parity.

Tag Out Bus

The Tag Out bus sends the five-bit tag instruction to the Controller to identify the operation to be performed. TAG GATE validates the instruction on the Tag Out bus. The bus consists of six bits plus parity; however, bit 3 of the bus is not used by the SCU or Controller.

Odd parity on the Tag Out bus is required. Even parity inhibits the TAG VALID signal, thereby preventing the Storage Control from communicating with the Controller and the Drives.

Tag Gate

The TAG GATE control line is used to deskew the Bus Out and Tag Out lines to the Controller and Drive. TAG GATE is also used to establish the time at which the Tag Out and the Bus Out lines are valid. The line remains valid until acknowledged by the Controller with TAG VALID (see Figure 6-26 for timing relationship).

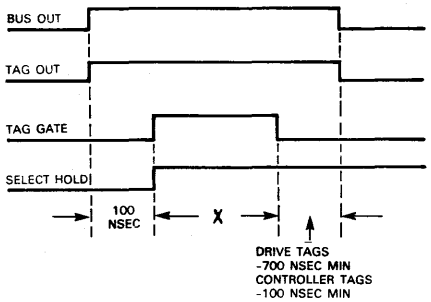


Figure 6-26. SCU to Controller Interface Timing

Select Hold

The SELECT HOLD line rises during any select tag and remains up to maintain selection of a Controller and/or Drive. The line stays up until the end signal of the last operation to be performed on the Controller and/or Drive is received and acknowledged (see Figure 6-26 for timing relationship).

Sync Out

The SYNC OUT line is used to validate and deskew the Bus Out bits during data transfers from SCU to Controller. It is also used during data transfers from the Controller to SCU to check the data count (see BUS OUT/SYNC OUT waveforms of Figure 6-27 for timing relationship).

Response

This line indicates acknowledgement of a Normal End or Check End condition for extended (Read/Write) operations.

Recycle

The RECYCLE signal is used during a (read or write) data transfer to keep the four-bit data transfer counter in the Controller counting when more than 16 bytes of data are to be transferred. The signal is controlled by the data transfer counter in the Storage Control. RECYCLE prevents the Controller from setting End of Data Transfer until the SCU drops RECYCLE.

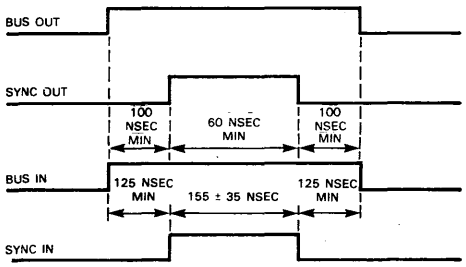


Figure 6-27. Controller to SCU Interface Timing (Data Transfers)

CE Communication

The CE Communication (CE COMM) line is used for diagnostic purposes only.

6.4.2.2 CONTROLLER TO SCU SIGNALS

Select Active

The SELECT ACTIVE line becomes active as a result of a selection sequence. The line remains active to indicate proper selection as long as SELECT HOLD is active and selection of the Drive is correctly maintained by the Controller.

Sync In

During data transfers from Controller to SCU, SYNC IN validates and times the Bus In data. It rises after Bus In data is valid. Bus In data remains valid until after SYNC IN ends. During data transfers from the Storage Control to the Controller, SYNC IN provides timing for data transferred (refer to BUS IN/SYNC IN waveforms of Figure 6-27 for timing relationship).

Normal End

NORMAL END indicates that the normal ending of an operation occurred with the expected results obtained. Ending information on Bus In is validated by the rise of NORMAL END. For immediate tags, NORMAL END is generated by TAG GATE, TAG VALID, and NOT DATA TRANSFER. NORMAL END drops when TAG GATE goes inactive.

For extended instructions, NORMAL END is not presented until after the read or write operation is completed which is some undetermined time after TAG GATE is reset. NORMAL END is reset by RESPONSE after an extended operation.

Check End

CHECK END indicates that an abnormal ending condition exists. The abnormal condition is presented on Bus In along with proper parity during the duration of CHECK END. For Read or Write operations, CHECK END stays on and BUS IN maintains proper parity until the Storage Control acknowledges the receipt of the abnormal end status information with the RESPONSE line. CHECK END is used for extended operations only.

Tag Valid

The TAG VALID line indicates that the Controller or Drive has validated and accepted the tag instruction sent from the SCU.

Selected Alert 1

This line indicates an unusual condition (Equipment Check) in the selected Controller or Drive. This line is Reset by a Check Reset or Controller Reset operation performed by the SCU.

Selected Alert 2

The SELECTED ALERT 2 line indicates the detection of Index in the selected Drive. This line will not be active unless a Drive is presently in operation (i.e., an *Operate Up* tag (decode 8B) has been issued after selection).

Unselected Alert 1

One line used for diagnostic purposes only.

Bus In

Data from the Controller or Drive during a Read operation, as well as ECC conditions or error information, are transmitted to the SCU by means of the Bus In lines. This bus consists of nine lines: eight bits plus parity.

The SCU has the responsibility of deskewing Bus In information except during read data transfers.

6.4.3 Register Definitions

CO Register

The CO Register generates the tag out bits for the Controller along with SELECT HOLD and CTL TAG GATE for control of the file. The register can be loaded from either the D Bus or the T Bus.

CO REGISTER BIT ASSIGNMENTS							
CO 0	CO 1	CO 2	CO 3	CO 4	CO 5	CO 6	CO 7
SELECT OUT	SE-QUENCE OUT	TAG BUS 0	TAG BUS 3	TAG BUS 4	TAG BUS 5	TAG BUS 6	TAG BUS 7

The tag out bits (0, 3-7) have odd parity generated to be sent with the tag bits to the Controller.

DO Register

The DO Register is used when transferring data to the file in the write mode, or used to define the command to the Controller in conjunction with the tag out bits defined by the CO Register. The register can be loaded from either the D Bus or the T Bus.

DO REGISTER BIT ASSIGNMENTS							
DO 0	DO 1	DO 2	DO 3	DO 4	DO 5	DO 6	DO 7
BUS OUT 0	BUS OUT 1	BUS OUT 2	BUS OUT 3	BUS OUT 4	BUS OUT 5	BUS OUT 6	BUS OUT 7

During write operations, the DO Register is loaded by the control program one byte at a time. The load of DO takes place whenever the control program specifies the DO as a destination on either the T or D Bus. There is a two-buffer sequence for all write operations in the SCU. It is initiated by loading DO with the physical ID, and then raising the Load Write Buffer (BUF/W) signal (RWC bit 0) to gate DO to BUF/W. Then DO is loaded with the next byte. Hence, the SCU has two bytes ready to transfer to the Controller upon receiving SYNC IN.

Data Transfer Counter

The Data Transfer Counter (DTC) Register is a sixteen-bit special-purpose, up/down counter register. The DTC Register is loaded as a two-byte register from R2 and R3 on T Bus via the control statement, C(DTC = R23).

The DTC Register is loaded by the control program to specify the number of data bytes to be transferred between the SCU and the Controller. The DTC can also be used as a special-purpose counter.

1. Read Operations

   During the time that a read mode is indicated (RWC bit 6 is set), the DTC Register is decremented with SYNC IN. Since the DTC Register is synchronized with the machine clock, the DTC Register contents in the ECR Register may be sampled at any time by use of MCM bits 2 and 3.

2. Write Operations

   During the time that a write mode is indicated (RWC bit 7 is set), the DTC Register is decremented with each SYNC IN tag from the Controller. When the counter reaches zero, the DTC = 0 branch condition is set on.

Read Buffer Register

The Read Buffer (DI) Register is a one-byte special-purpose register used to synchronize and buffer the read data between the SCU Microprocessor and the Controller. Transfers of data between the DI Register and Controller are done one byte at a time with SYNC IN/SYNC OUT control tags.

DI REGISTER BIT ASSIGNMENTS							
DI 0	DI 1	DI 2	DI 3	DI 4	DI 5	DI 6	DI 7
BUS IN 0	BUS IN 1	BUS IN 2	BUS IN 3	BUS IN 4	BUS IN 5	BUS IN 6	BUS IN 7

During read operations the DI Register is loaded one byte at a time at a nominal rate of 1.24 microseconds per byte. The actual transfer of the assembled byte takes place with SYNC IN synchronized by a clock signal. When DI is loaded, the DBR branch condition is raised to the microprocessor. When DBR is raised, the control program must initiate a read of DI within approximately 1000 nanoseconds or an overlay of the byte may take place (430 nanoseconds is the worst case delay between the load and rise of DBR plus control program branch uncertainty). Thus, not counting the instruction to branch on DBR, the control program has two instructions to read DI. The read of DI can occur at the same time that the CBR condition is set. Initially, data bytes are placed in DI starting with the physical ID of the field, following initiation of the read mode in the RWC register. Data bytes are placed in DI as long as RWC bit remains set and SYNC IN is latched. After RWC bit 6 is reset, no more bytes are placed into DI by SYNC IN until the read mode is again initiated.

RWC Register

The RWC register is used by the control program to control the transfer functions of the CTL-I.

RWC REGISTER BIT ASSIGNMENTS							
0	1	2	3	4	5	6	7
GATE DO TO WRITE BUFFER	CTL RE-CYCLE	GATE BUS IN	ALLOW RE-SPONSE		CLOCK DTC UP	CTL-I READ MODE	CTL-I WRITE MODE

RWC Register Bit 0 Gate DO Reg to Write Buffer. This bit loads the Write Buffer Register with the contents of the DO Register. The contents are either a command to the Controller or data to be written into the files by a write command.

RWC Register Bit 1 Enable Recycle for Data Transfer. This bit is used to enable the RECYCLE signal to the Controller. RECYCLE is used to inform the Controller that it should recycle its byte counter when it reaches zero because there are more data bytes to be transferred during read or write operations. The recycle function is active as long as the DTC Register is greater than 15 during read or write mode.

RWC Register Bit 2 Gate Bus In. This bit is used to gate data from the Read Buffer Register into the DI Register at Register Time during a Read Mode or command sequence. The data transfer takes place with or without TAG VALID, NORMAL END, or CHECK END.

RWC Register Bit 3 Allow Response. This bit enables the generation of the RESPONSE signal to the Controller whenever NORMAL END or CHECK END is received from the Controller in an extended operation.

RWC Register Bit 4 Not Used.

RWC Register Bit 5 Clock DTC Up. This bit controls the incrementing or decrementing state of the DTC Register. When bit 4 is high, the DTC Register will count up with each clock pulse. When bit 4 is low, the DTC Register will count down with each clock pulse.

RWC Register Bit 6 Read Mode. This bit on indicates that a Read operation is in progress, and enables control and timing pulses for any Read operation sequence to the Controller.

RWC Register Bit 7 Write Mode. This bit on indicates that a Write operation is in progress, and enables control and timing pulses for any Write operation sequence to the Controller.

ECR Register

This register (Table 6-7) is pageable by bits 1, 2, and 3 of the MCM register. The ECR register contains information on the inbound tags from the Controller Check-2 error conditions and the 16 bits of the DTC.

MCM Register

The MCM Register is a four-bit register used to multiplex the ECR Register contents to the Microprocessor. MCM bit 4 with Latched Data Bit 6 (LDB 6) is used to define the entries into the MCK + FDF multiplexor which is an input to the A Bus of the ALU. The MCM Register is loaded with data from D Bus bits 0-4.

The D Bus bit 0 entry to the MCM Register is defined as the "Enable CK2" signal, used to look for Check 2 errors.

TABLE 6-7. ECR BITS

MCM 1 2 3	0	1	2	3	4	5	6	7
0 0 0	SELECT ALERT 1	SELECT ACTIVE CHECK	SYNC IN	UNEXP. END	NORMAL END	CHECK END	TAG VALID	(NOT USED)
0 0 1	DI BUFFER CHECK	SELECT ACTIVE CHECK	(NOT USED)	CE ALERT	CHANNEL INTERFACE CHECKS			
0 1 0	DTC 0	DTC 1	DTC 2	DTC 3	DTC 4	DTC 5	DTC 6	DTC 7
0 1 1	DTC 8	DTC 9	DTC 10	DTC 11	DTC 12	DTC 13	DTC 14	DTC 15

6.5 FE INTERFACE

6.5.1 General

The FE (Field Engineer) interface contains the logic required to manually address, load, start and stop the SCU during maintenance procedures. The logic required to control and display diagnostic routines and errors, while in the In-line or Off-line mode, is also contained in the FE interface.

6.5.2 Functional Description

The functional description of the FE Interface includes a brief description of each functional block in the block diagram of Figure 6-28.

The Toggle Switch Logic contains the latches and gating for the MACHINE CONTROL toggle switches on the FE panel.

The Error Light Driver buffers and drives the MACHINE CHECK and DIAGNOSTIC CONTROL signals from the various printed circuit boards (i.e., GE, GD, CM, etc.) to be displayed on the FE panel.

The Diagnostic Indicator Driver and MD2 Register logic buffers, drives and gates the Diagnostic Control Indicator signals from the Microprocessor, via the D Bus, to the DIAGNOSTIC CONTROL indicator on the FE panel.

The Display Selection Logic gates microprocessor timing pulses and uncoded control panel commands to generate data and parity display strobes for the Display Register. The logic also generates select enable signals to the various input boards and to the Address Register Control logic.

The Switch Drivers drive command signals from the Roller Bar switch to the ALU.

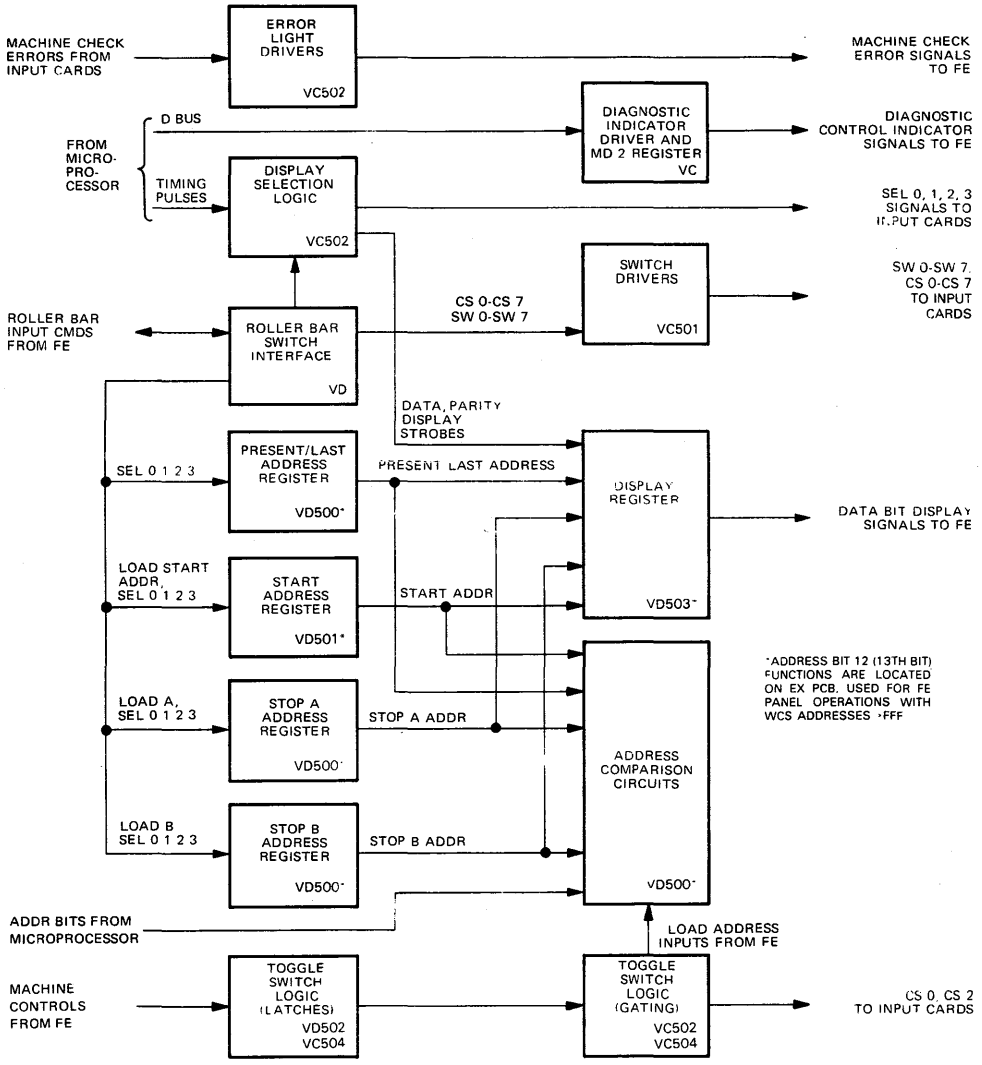


Figure 6-28 FE Interface Block Diagram

The Roller Bar Switch Interface gates and drives MACHINE CONTROL commands to the input cards and to the Address Register Control logic and comparison circuits.

The Present/Last Address Register buffers the address bits from the Microprocessor and gates them with the uncoded Control Panel Display Commands for display on the FE panel.

The Start Address register stores and gates the LOAD ADDRESS input commands from the FE panel, and is strobed by a LOAD START ADDR command.

The Stop A Address Register stores and gates the LOAD ADDRESS input commands from the FE panel, and is strobed by a LOAD A command.

The Stop B Address Register stores and gates the LOAD ADDRESS input commands from the FE panel and is strobed by a LOAD B command.

The Data Display Register stores up to 36 bits of data which are displayed on the FE panel. The register is strobed by a Data Set Strobe pulse, or by a Parity Set Strobe pulse.

The Address Comparison Circuits input address bits from the Start, Stop A and Stop B Address Registers and compare the bits with the address bits from the Microprocessor. Each Address Register control circuit generates a pulse (SYNC EQUAL, STOP A EQUAL, STOP B EQUAL) whenever comparison is attained.

For FE panel operations requiring access to the WCS expansion feature, a thirteenth address bit (bit 12) is provided to access locations greater than 4095 (0FFF hex). The switch and present/last indicators associated with this address bit are located on the EX board, along with the sync/stop A/stop B equal logic.

6.6 CHECK-1 ERRORS

6.6.1 Error Detection and Sequences

The 3672 Storage Control Unit includes circuitry which detects the presence of internal Microprocessor errors (called Check-1 errors). These error conditions are catastrophic in the sense that the SCU functional microprogram cannot recover from them with any assurance of proper microprogram orientation or customer data integrity. (Check-1 errors are in contrast to Check-2 errors, which are recoverable.) For this reason, the detection of a Check-1 error forces a program halt and the initiation of a channel sequence which, in turn, forces the channel to issue a Selective Reset. The Selective Reset restarts the microprogram. The microprogram then resets its operational status and assembles the error information which was stored in the Machine Check (MCK) Register upon error detection. This error information is sent to the channel upon a *Sense I/O* command following the next *Test I/O* command.

The channel sequence following Check-1 error detection falls into two categories, depending on the state of the OPERATIONAL IN signal. If OPERATIONAL IN is up when the error is detected, the SCU raises DISCONNECT IN and the channel issues a Selective Reset (raises SUPPRESS OUT and drops OPERATIONAL OUT). If OPERATIONAL IN is down when the error is detected, the SCU raises REQUEST IN. Then in response to COMMAND OUT in the SCU-initiated sequence, the SCU raises DISCONNECT IN and the channel follows with a Selective Reset.

A DISCONNECT IN interlock in the SCU channel interface prohibits the raising of DISCONNECT IN after a Selective Reset until the point where ADDRESS IN is raised in the next channel sequence. If a Check-1 error occurs while DISCONNECT IN interlock is active, the microprogram stops and the TAGS IN lines are frozen. At this point, the channel can ignore the SCU or, after a time out, issue a reset. The DISCONNECT IN interlock circuitry insures that the channel does not get into a DISCONNECT IN Selective Reset loop.

6.6.2 Machine Check (MCK) Register

The MCK Register consists of 48 bits which contain information concerning the error status of the machine. These bits are multiplexed to the A Bus by MCM bits 5, 6 and 7 as shown in Table 6-3 of paragraph 6.3.4.2. Error bits 0 through 15 are the actual Check-1 error bits. If all these bits are false, the SCU Microprocessor is functioning properly. When any one of these error bits is set, the status of all 16 bits (0-15) is frozen as well as the failing address (27-39) associated with the error(s). If the active error bit is MULTIPLE WCS ERROR (1), the failing WCS error-correcting parity pattern is also frozen (16-23). If the active error bit is A Bus, B Bus or T Bus parity error (2-4), then MCK bits 40 through 43 point to the PCB (GA, GB, GC or GD) which contains the source register associated with that particular parity error. MCK bits 25, 26 and 44 through 47 are Check-2 errors.

6.6.3 Check-1 Error Display

When a Check-1 error is detected, one or more discrete LED indicators on the SCU FE panel illuminate, defining the particular error detected. The state of these indicators is derived directly from MCK bits 1 through 15 and 40 through 43. For example, the BRNCH error indicator is illuminated if MCK bit 5, 8 or 11 is set (Branch Error, BR Multiplexer Error, or Sub-routine Error). The indicator associated with each MCK bit is listed with the MCK bit definitions in Table 6-8. If there is a Check-1 error indicated by the LED matrix, then the address displayed in roller position E of the FE panel is the failing address associated with that error.

6.6.4 Reset of Check-1 Error Conditions

Although the SCU Check-1 error circuitry will stop the clock each time an error is detected, the state of the MCK Check-1 error bits is frozen once any Check-1 bit is set (MCK bits 0-15). To arm these MCK bits for further error latch-up, these bits must be reset. The microprogram can issue a C(RESET.CKS) control statement to perform this function. Under normal system operation, the microprogram will reset the Check-1 bits after it has assembled the error status into the sense bytes following a Selective Reset. In addition, the FE can reset the Check-1 bits and failing address using the FE RESET switch. As a further control option, the FE can allow the microprogram to continue to run when a Check-1 error occurs through use of the CHECK-1 OVERRIDE switch.

When the SCU is in the Scan mode, all Check-1 error latches are held reset, except MULTIPLE WCS ERROR and WCS ECC ERROR. If a noncorrectable WCS error occurs, the clock stops with the appropriate error indicator (RAM or ECC) illuminated and the failing address displayable at roller position E. The FE RESET switch resets these error bits if the SCAN switch is deactivated. The CHECK-1 OVERRIDE switch can also be utilized while in Scan mode. The only Check-2 error enabled to stop the clock while in Scan mode is SINGLE RAM ERROR. No indicator will be displayed except STOP CLOCK. This error can also be degated.

Table 6-8. MCK REGISTER DEFINITION

BIT	DEFINITION	ERROR CLASS	DISCRETE DISPLAY
0	CHC PARITY ERROR CHC Register contains wrong parity error.	CK-1	None
1	MULTIPLE WCS ERROR A multiple (noncorrectable) WCS accessing error.	CK-1	RAM Indicator
2	A BUS PARITY ERROR A miscompare of the parity generated for a register on A Bus and the parity bit previously stored for that register.	CK-1	A Bus Indicator
3	B BUS PARITY ERROR A miscompare of the parity generated for a register on B Bus and the parity bit previously stored for that register.	CK-1	B Bus Indicator
4	T BUS PARITY ERROR A miscompare of the parity generated for a register on T Bus and the parity bit(s) previously stored for that register.	CK-1	T Bus Indicator
5	BRANCH ERROR a. Miscompare of address parity and incrementer parity if increment cycle. b. Miscompare of address parity and SR parity if B (ADR = SR) cycle. c. Detection of simultaneous Branch Set and Increment Set signals.	CK-1	BRNCH Indicator
6	IMPL ERROR During hardware IMPL, one of the following conditions was detected: a. Single Track Seek was not completed within one second. b. A sync byte was not found in the FD read data within one second. c. Read overrun. d. Sixteen occurrences of: 1. No TRACK = 0 decode (after TRACK 0 line from FD valid), or 2. Parity error detected on FD Read Data.	CK-1	IMPL Indicator
7	ALU ERROR a. Miscompare of duplicate ALU outputs including D = FF and carry conditions. b. Miscompare of latched D Bus with the ALU output.	CK-1	ALU Indicator
8	BR MPXR ERROR Miscompare of parity generated for a register at the output of the BR Multiplexer with the parity bit previously stored for that register.	CK-1	BRNCH Indicator
9	PS ERROR Miscompare of duplicate PS registers.	CK-1	PAGE Indicator
10	OP CODE TRANSLATION ERROR Invalid control field status for the current OP code.	CK-1	TRANS Indicator
11	SUBROUTINE ERROR Miscompare of SR parity and address parity parity for C(LD.SR.*+1) or B(LD.SF.*+1).	CK-1	BRNCH Indicator
12	WCS ECC ERROR Invalid combination of overall WCS access parity, corrected word parity, and any P.	CK-1	ECC Indicator
13	MULTIPLE BUFFER ERROR A multiple (noncorrectable) buffer accessing error.	CK-1	BUFF Indicator
14	BUFFER ECC ERROR Invalid combination of buffer access parity, corrected word parity, and any P.	CK-1	BUFF Indicator
15	BUFFER WRITE T BUS ERROR Miscompare of T Bus parity with parity generated at the input of the buffer prior to a write, C(BUF = R03).	CK-1	BUFF Indicator
16,23	FAILING P1-P8 The 8-bit error correcting parity pattern associated with a multiple or single WCS error (MCK 1 OR MCK 26).	CK-1 or CK-2	
24	SELECTIVE RESET Indicates that channel has issued a Selective Reset (Microprogram checks this bit in the reset procedure to distinguish between General and Selective Reset).		
25	SINGLE BUFFER ERROR Single (correctable) buffer accessing error.	CK-2 (for diagnostic purposes only)	
26	SINGLE WCS ERROR Single (correctable) WCS accessing error.	CK-2 (in SCAN only)	
27 39	FAILING ADDRESS 12-0 Address to be associated with the Check-1 error(s) latched in MCK bits 0 through 15.	CK-1	Roller position E
40	GA PCB The PCB containing the source register for an A Bus, B Bus, or T Bus parity error is the GA PCB.	CK-1	GA Indicator
41	GB PCB The PCB containing the source register for an A Bus parity error is the GD PCB.	CK-1	GB Indicator
42	GC PCB The PCB containing the source register for an A Bus or T Bus parity error is the GC PCB.	CK-1	GC Indicator
43	GD PCB The PCB containing the source register for an A Bus parity error is the GD PCB.	CK-1	GD Indicator
44	FD READ ERROR a. Parity error was detected on the one second, or b. Read overrun, or c. A sync byte was not detected in the FD read data within one second.	CK-2 or CK-1 (if IMPL)	
45	FD SEEK ERROR a. A Seek was not completed within one second, or b. A TRACK = 0 Read decode was not found within two sectors after the TRACK 0 line from the FD went active.	CK-2 or CK-1 (if IMPL)	
46	FD NOT READY A time delay signal which should be 5-7 seconds in duration after FD power is initiated.		
47	FD WRITE BUS ERROR A parity error was detected at the FDD or FDC registers.	CK-2	

NOTE:

All MCK bits can be displayed through SET DISPLAY of MCK register (microprogram display).

GLOSSARY OF TERMS

ADDR, ADRS	Address
ALU	Arithmetic Logic Unit
AM	Address Mark
B	Branch
BAR	Buffer Address Register
BI	Bus In
BO	Bus Out
B_1	I/O Instruction Base Address Register Location
C	Count
CAR	Cylinder Address Register
CAW	Channel Address Word
CBR	Channel Buffer Ready
CC	Chain Command, Channel Byte Count, Condition Code
C, C	Cylinder Address (two bytes)
CD	Chain Data
CCW	Channel Command Word
CE	Channel End
CHC	Channel Control
CHF	Channel Flags
CHNL	Channel
CI	Channel Interrupt
CK	Check
CKD	Count/Key/Data
CLK	Clock
CMD	Command
CP	Channel Priority
CPU	Central Processing Unit
CT	Channel Tag
CTL	Controller
CTL-I	Controller Interface
CUDI	Control Unit Device Interface
D	Data
DC	Data Counter
DCC	Disconnected Command Chaining
DDM	Disc Drive Module
DE	Device End
DEV	Device
DI	Device Interrupt
DL	Data Length
DO	Data Out
DTC	Data Transfer Counter
D_1	I/O Instruction Displacement
ECC	Error Correction Code
EOB	End of Block
EREP	Error Recovery Procedure
ERP	Error Report
FD	Flexible Disc
FDC	Flexible Disc Control
FDD	Flexible Disc Drive
FE	Field Engineer
FIDS	Fault Isolation Diagnostic System
FLD	Field
FSR	Field Service Representative
Gn	Gap n
HA	Home Address
HAR	Home Address Register
H, H	Head Address (two bytes)
ID	Identifier
ILC	Instruction Length Code
INCR	Increment
I(M)PL	Initial (Micro) Program Load
K	Key (or Constant = 1024)
KL	Key Length
M	Machine Check Mask
MCK	Machine Check
MPXR	Multiplexer
MUX	Multiplexer
NO-OP	No Operation
OLTEP	On Line Test Executive Program
OP	Operation
P	Problem State
PCI	Program Control Interrupt
PLO	Phase Locked Oscillator
PS	Page Select
QE	Queue Empty
SCTR	Sector
SCU	Storage Control Unit
SEQ	Sequence
SERDES	Serializer/Deserializer
SLI	Suppress Length Indicator
SR	Subroutine
SW	Switch
R	Record
RN	Record n
RAM	Random Access Memory
RST	Reset
R/W	Read/Write
RWC	Read/Write Control
R03	Registers 0 through 3
TIC	Transfer In Channel
TR	Transmit Request
TRK	Track
UC	Unit Check
UE	Unit Exception
VFO	Variable Frequency Oscillator
W	Wait State
WCS	Writable Control Storage
μ	Micro (used as prefix, as in μprogram)

COMMENTS FORM

3672 Storage Control Unit Theory of Operations — 3672.21-00/0001

Please send us your comments; they will help us produce better publications. Whenever possible, please use specific page and paragraph references.

• Is the material:	Yes	No
Easy to understand?	☐	☐
Conveniently organized?	☐	☐
Complete?	☐	☐
Well illustrated?	☐	☐
Accurate?	☐	☐
Suitable for its intended audience?	☐	☐

- For what purpose did you use this publication (reference, general interest, etc.)?

_____

- Please state your department's function: _____

_____

Please use the space below for specific comments which you feel would improve any aspect of this publication. All comments become the property of Memorex Corporation.

_____

_____

_____

_____

_____

_____

_____

MEMOREX

First Class

Permit No. 250
Santa Clara
California
95050

Business Reply Mail

No Postage Necessary if Mailed in the United States

Postage Will Be Paid By

Memorex Corporation

Publications Department
Mail Stop 14-07
San Tomas at Central Expressway
Santa Clara, California 95052

Thank you for your information.....

Our goal is to provide better, more useful manuals, and your comments will help us to do so.

.....Memorex Publications