CONTROL DATA
CORPORATION

CONTROL DATA®
FA7A6/FA7A7 DISK CONTROLLER

HARDWARE REFERENCE MANUAL

TITLE: FA7A6/FA7A7 Disk Controller Hardware Reference Manual

CONTROL DATA CORPORATION
TECHNICAL PUBLICATIONS DEPARTMENT
7801 COMPUTER AVENUE
MINNEAPOLIS, MINNESOTA 55435

PUBLICATION NO: 70615200

REVISION: E

REASON FOR CHANGE: **DATE:** January 17, 1977

REASON	EFFECTIVITY	DESCRIPTION
1	All	Add functional Microprogram flowcharts.
2	All	Add page
3	All	Technical Correction

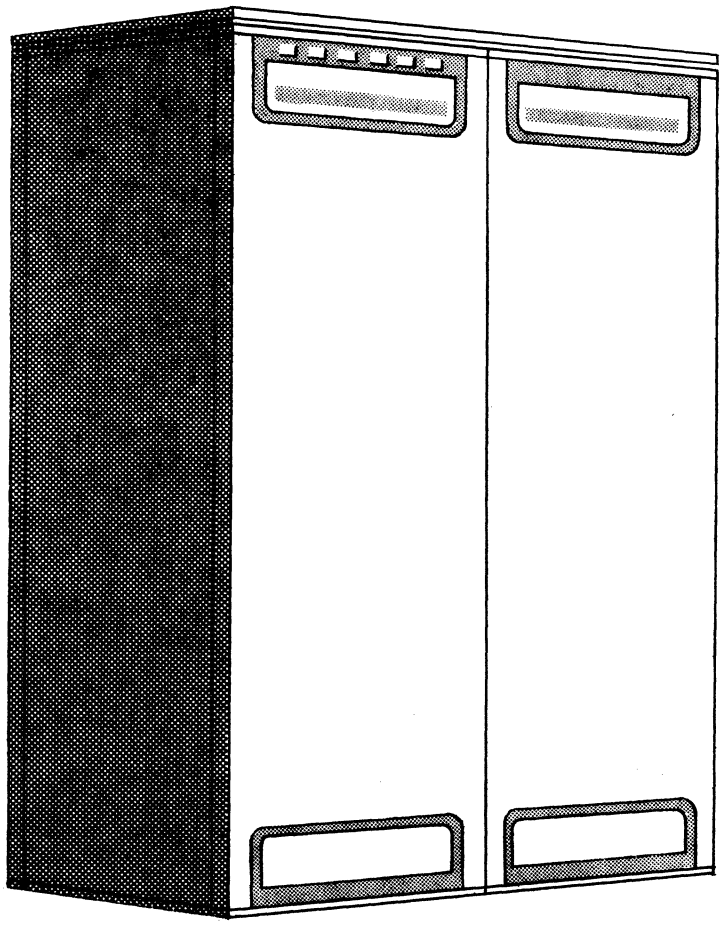
INSTRUCTIONS:

REPLACE, ADD, OR DELETE EXISTING PAGES ACCORDING TO THE FOLLOWING LIST:

PAGE NO.	REASON NO.	PAGE NO.	REASON NO.	PAGE NO.	REASON NO.	PAGE NO.	REASON NO.
ii	1						
iii	1						
iv	1						
ivA/ivB	1,2						
ix	1						
x	1						
xi/xii	2						
2-4	3						
3-10	3						
3-31	3						
Part 3 Divider	2						
3-113 thru 3-216	1,2						

CONTROL DATA
CORPORATION

CONTROL DATA®
FA7A6/FA7A7 DISK CONTROLLER



HARDWARE REFERENCE MANUAL

REVISION RECORD

REVISION	DESCRIPTION
A	Manual released.
(10-24-73)	
B	Manual revised to correct errors and add Theory of Operation - This edition obsoletes all previous
(6-30-74)	editions.
C	Manual revised to add technical and Editorial Changes. Changed Front Cover and Title Page.
(1-24-75)	Revised pages iii, iv, v/vi, vii, ix, x, 1-1, 1-2, 1-2A/1-2B (added), 1-3, 1-4, 1-6, 1-10, 1-17,
	1-18, 1-19, 1-21, 2-3, 2-5, 2-6, 2-7/2-8, 3-13, 3-14, 3-16, 3-50, 3-87.
D	Incorporate Engineering Change Orders PE32558 (O/C 149), PE40180 (O/C 148), Field
(9-26-75)	Change Order PE32558, and Technical Corrections affecting pages ii, iii, iv, 2-4,
	3-9, 3-10, 3-14, 3-22, 3-27, 3-31, 3-38, 3-39, 3-71, 3-72, 3-73, 3-74.
E	Manual revised to add functional microprogram flowcharts and technical corrections
(1-17-77)	affecting pages ii, iii, iv, ivA/ivB (added), ix, x, xi/xii (added), 2-4, 3-10, and
	3-31. Added section 3, Part 3 (pages 3-113 through 3-216).
Publication No. 70615200	

Printed in the United States of America

Address comments concerning this manual to:

Control Data Corporation
Technical Publications Department
7801 Computer Avenue
Minneapolis MN 55435

or use Comment Sheet in the back of this manual.

MANUAL STATUS CHECK SHEET

PUBLICATION NO. 70615200 REVISION E

PUBLICATION TITLE FA7A6/FA7A7 HARDWARE REFERENCE MANUAL

PAGE NUMBER	REV
Cover	-
Title Page	-
ii	E
iii	E
iv	E
ivA/ivB	E
v/vi	C
vii	C
viii	B
ix	E
x	E
xi/xii	E
Section 1	-
1-1	C
1-2	C
1-2A/1-2B	C
1-3	C
1-4	C
1-5	B
1-6	C
1-7	B
1-8	B
1-9	B

PAGE NUMBER	REV
1-10	C
1-11	B
1-12	B
1-13	B
1-14	B
1-15	B
1-16	B
1-17	C
1-18	C
1-19	C
1-20	B
1-21	C
1-22	B
1-23	B
1-24	B
1-25	B
1-26	B
Section 2	-
2-1	B
2-2	B
2-3	C
2-4	E
2-5	C

PAGE NUMBER	REV
2-6	C
2-7/2-8	C
Section 3	-
3-1	B
3-2	B
3-3	B
3-4	B
3-5	B
3-6	B
3-7	B
3-8	B
3-9	D
3-10	E
3-11	B
3-12	B
3-13	C
3-14	D
3-15	B
3-16	C
3-17	B
3-18	B
3-19	B
3-20	B

PAGE NUMBER	REV
3-21	B
3-22	D
3-23	B
3-24	B
3-25	B
3-26	B
3-27	D
3-28	B
3-29	B
3-30	B
3-31	E
3-32	B
3-33	B
3-34	B
3-35	B
3-36	B
3-37	B
3-38	D
3-39	D
3-40	B
3-41	B
3-42	B
3-43	B

MANUAL STATUS CHECK SHEET

PUBLICATION NO. 70615200 REVISION E

PUBLICATION TITLE FA7A6/FA7A7 HARDWARE REFERENCE MANUAL

PAGE NUMBER	REV
3-44	B
3-45	B
3-46	B
3-47	B
3-48	B
3-49	B
3-50	C
3-51	B
3-52	B
3-53	B
3-54	B
3-55	B
3-56	B
3-57	B
3-58	B
3-59	B
3-60	B
3-61	B
3-62	B
3-63	B
3-64	B
3-65	B
3-66	B

PAGE NUMBER	REV
3-67	B
3-68	B
3-69	B
3-70	B
3-71	D
3-72	D
3-73	D
3-74	D
3-75	B
3-76	B
3-77	B
3-78	B
3-79	B
3-80	B
3-81	B
3-82	B
3-83	B
3-84	B
3-85	B
3-86	B
3-87	C
3-88	B
3-89	B

PAGE NUMBER	REV
3-90	B
3-91	B
3-92	B
3-93	B
3-94	B
3-95	B
3-96	B
3-97	B
3-98	B
3-99	B
3-100	B
3-101	B
3-102	B
3-103	B
3-104	B
3-105	B
3-106	B
3-107	B
3-108	B
3-109	B
3-110	B
3-111/3-112	B
3-113	E

PAGE NUMBER	REV
3-114	E
3-115	E
3-116	E
3-117	E
3-118	E
3-119	E
3-120	E
3-121	E
3-122	E
3-123	E
3-124	E
3-125	E
3-126	E
3-127	E
3-128	E
3-129	E
3-130	E
3-131	E
3-132	E
3-133	E
3-134	E
3-135	E
3-136	E

MANUAL STATUS CHECK SHEET

PUBLICATION NO. 70615200 REVISION E

PUBLICATION TITLE FA7A6/FA7A7 HARDWARE REFERENCE MANUAL

PAGE NUMBER	REV
3-137	E
3-138	E
3-139	E
3-140	E
3-141	E
3-142	E
3-143	E
3-144	E
3-145	E
3-146	E
3-147	E
3-148	E
3-149	E
3-150	E
3-151	E
3-152	E
3-153	E
3-154	E
3-155	E
3-156	E
3-157	E
3-158	E
3-159	E

PAGE NUMBER	REV
3-160	E
3-161	E
3-162	E
3-163	E
3-164	E
3-165	E
3-166	E
3-167	E
3-168	E
3-169	E
3-170	E
3-171	E
3-172	E
3-173	E
3-174	E
3-175	E
3-176	E
3-177	E
3-178	E
3-179	E
3-180	E
3-181	E
3-182	E

PAGE NUMBER	REV
3-183	E
3-184	E
3-185	E
3-186	E
3-187	E
3-188	E
3-189	E
3-190	E
3-191	E
3-192	E
3-193	E
3-194	E
3-195	E
3-196	E
3-197	E
3-198	E
3-199	E
3-200	E
3-201	E
3-202	E
3-203	E
3-204	E
3-205	E

PAGE NUMBER	REV
3-206	E
3-207	E
3-208	E
3-209	E
3-210	E
3-211	E
3-212	E
3-213	E
3-214	E
3-215	E
3-216	E
Cmt Sheet	-
Rtn Env.	-
Back Cover	-

PREFACE

INTRODUCTION

This manual has been prepared for customer engineers and other technical personnel directly involved in maintaining the disk controller.

OTHER MANUALS

Other manuals applicable to the controller are:

Publication No.	Title
70615300	Maintenance Manual Vol. 1 (Installation and Checkout, Preventive Maintenance, and Corrective Maintenance)
83310400	Maintenance Manual Vol. 2 (Diagrams)
83303500	Maintenance Manual Vol. 3 (Wire Lists)
70619000	Subsystem Troubleshooting Manual
70625500	Parts Data Manual
70631200	Microprogram Manual

MACHINE CONFIGURATIONS

This manual provides reference information for all standard configurations of both the FA7A6 and FA7A7 controllers. The chart below specifies the differences between models.

Equipment Type No.	AC Power (Hz)	Channels	Density (Megabytes)
FA7A6 -A	60	2	100
-B	50	2	100
-C	60	1	100
-D	50	1	100
-E	60	2	200
-F	50	2	200
-G	60	1	200
-H	50	1	200
FA7A7 -A	60	2	100
-B	50	2	100
-C	60	1	100
-D	50	1	100
-E	60	2	200
-F	50	2	200
-G	60	1	200
-H	50	1	200

CONTENTS

FIGURES

TABLES

SECTION 1

GENERAL DESCRIPTION

INTRODUCTION

This section provides a basic description of the CONTROL DATA® FA7A6 and FA7A7 Disk Controllers. Brief explanations of the following concepts are discussed: controller organization and operation, track format, and channel commands.

SUBSYSTEM DESCRIPTION

The basic subsystem consists of one controller and from one to eight radially connected on-line drives (Figure 1-1). It is designed for use with one or two selector channels of the Siemens 4004/150 processor. Either channel may address the controller and its associated drives via a two channel switch installed within the controller.

The subsystem used the RCA Standard I/O Interface to identify the address of the drive to be controlled, the instruction to be executed, and the significance of data transferred. Data is transferred to or from one drive at a time, although off-line seek operations may occur simultaneously on the other connected drives.

The channel commands are interpreted to cause any of the following functions to be performed:

- Position the read/write heads to a specified physical location (Seek).
- Locate a record or portion of a record (Search).
- Read a record or portion of a record (Read).
- Write a record or portion of a record (Write).
- Supply channel with controller and drive status and error information (Sense).

The controller logical address is fixed at installation, while the logical address of each drive is controlled by removable address plugs.

Subsystem power is provided from the controller. All of the drives receive their ac power from the controller Power Control Unit (PCU). This permits subsystem power on/off control by using the EIN switch on the controller operator panel.

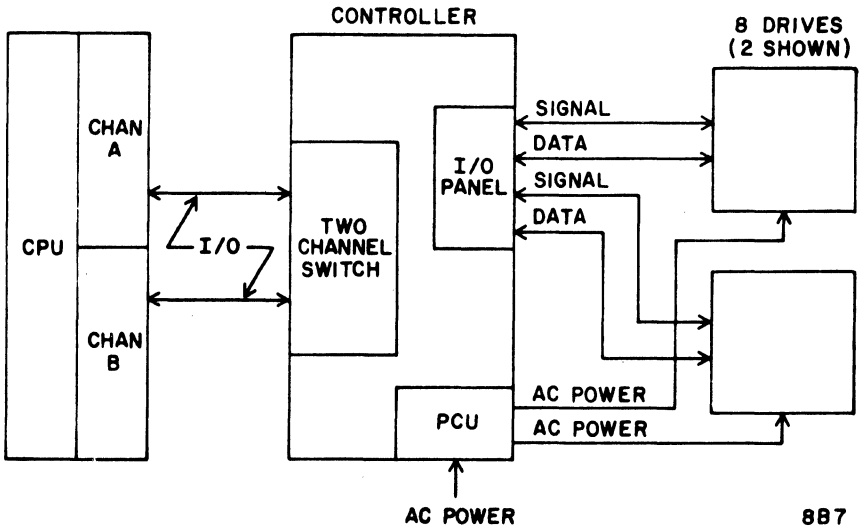


Figure 1-1. Basic System Configuration

Drives are powered on in groups of three with a 10 second delay between each group to decrease the total initial power surge. Each drive has its own independent dc power supplies.

Data is transferred between each drive and the controller by two I/O cables: the Signal Cable and the Data Cable. The Data Cable carries Read Data, Write Data, and Servo Clock Data. The remaining communication lines are in the Signal Cable. Since the I/O is in a radial configuration, each drive communicates with the controller with its own pair of I/O cables

TWO CHANNEL SWITCH

A two channel switch allows communication with two channels. When the controller is not busy, either channel has access to it. The first requesting channel reserves the controller until the command chain is complete, any pending interrupt is serviced, and all sense data that describes error that occurred. have been transmitted to that channel. If the second channel should attempt to communicate with the busy controller, the Standard Device Byte generated by the controller holds off the requesting channel. When the controller then becomes available, it signals an interrupt to the second channel.

TWO CONTROLLER FEATURE

The two controller (2 CU) feature permits two CPUs to communicate with the same drive. This feature is standard in all controllers and is optional in the drives used in the subsystem. Figure 1-1A illustrates the maximum subsystem configuration, that is, where all drives contain the option.

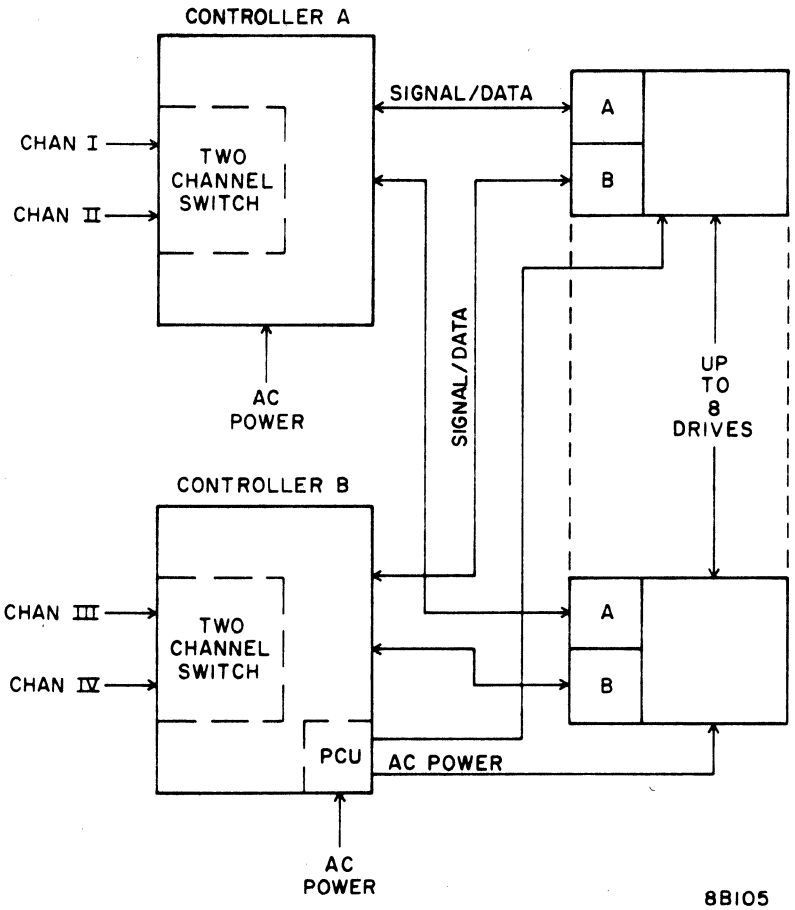


Figure 1-1A. Two Controller System Configuration

The first requesting channel reserves the drive. Two status conditions are applicable if another channel attempts to use the drive before the first channel releases it. If the first channel has not started a command chain, the second controller will provide sense information Device Reserved to Other Channel (0SB3, bit 2^3). On the other hand, if the chain is active, the second controller provides busy status. When the first channel has completed a chain (or job), it releases the drive. The drive may then be used by any channel.

SUBSYSTEM SPECIFICATIONS

Specifications for the controller and its associated drives are listed in Table 1-1.

TABLE 1-1. SUBSYSTEM SPECIFICATIONS

Characteristics	Controller	Drive
PHYSICAL SPECIFICATIONS		
SIZE		
Height	62 in.(157.8 cm)	38 in. max.(96 cm max.)
Width	49.5 in. (125.7 cm)	22 in. max. (56 cm max.)
Depth	24.7 in. (62.1 cm)	49 in. max.(124.3 cm max.)
Weight	691 lb. (314 kg)	670 lb. (305 kg)
ENVIRONMENTAL SPECIFICATIONS		
TEMPERATURE		
Operating	5°C to 50°C	16°C to 32°C
Max Temperature Gradient	.1°C/Min.	7°C/Hr.
Non-operating	-35°C to 65°C	-30°C to 68°C
RELATIVE HUMIDITY (without condensation)		
Operating	10% to 90%	20% to 80%
Non-operating	5% to 95%	5% to 95%

TABLE 1-1. SUBSYSTEM SPECIFICATIONS (Cont'd)

Characteristics	Controller	Drive
ALTITUDE		
Operating - min.	-300 meters (mean sea level)	mean sea level
Operating - max.	3000 meters (mean sea level)	3050 meters (mean sea level)
Non-operating - min.	-300 meters (mean sea level)	-305 meters (mean sea level)
Non-operating - max.	10,000 meters (mean sea level)	10,670 meters (mean sea level)
POWER SPECIFICATIONS		
INPUT AC VOLTAGE		
60 Hz (±0.6)	208v (±10%), 3Ø wye	208v (±10%), 3Ø* wye
50 Hz (±0.5)	220v (±10%), 3Ø delta	220v (±10%), 3Ø** delta
	NOTES: * Two phases are used per drive; drive power is connected such that each group of three drives/controller uses all three phases. Drive motors are single phase, connected phase-to-phase. ** One phase and neutral used per drive; drive power is connected such that each group of three drives/controller uses all three phases. Drive motors are single phase, connected phase-to-neutral.	
POWER CONSUMPTION		
Start-up		37 amp max. for 15 sec.max.
Standby		0.5 Kw (1600 BTU per hr)
One Drive Seeking	5.6 amp	1.3 Kw (4550 BTU per hr)
Eight Drives Seeking		10.4 Kw (36,400 BTU per hr)
DATA PROCESSING AND STORAGE SPECIFICATIONS		
DSU SEEK TIMES (milliseconds)		
Maximum		55
Average		30
Minimum		10
DISK PACK CONFIGURATION		
Recommended Disk Pack		CDC 879 (100 MB)* CDC 883 (200 MB)
Packs/Drive		1
Drives/System		8
Disks/Disk Pack		12
Recording Surfaces/ Disk Pack		19
Usable Tracks/ Recording Surface		404 (100 MB) 808 (200 MB)

* MB = Megabytes

TABLE 1-1. SUBSYSTEM SPECIFICATIONS (Cont'd)

Characteristics	Controller	Drive
DISK PACK CONFIGURATION (Cont'd)		
Spare Tracks/ Recording Surface		7 (100 MB) 15 (200 MB)
Tracks/Cylinder		19
Disk Pack Diameter		14 inches
Recording Diameter		
Outer (track 000)		12.743 inches
Inner (track 410)		8.480 inches
Tracks/Inch (nominal)		200 (100 MB) 400 (200 MB)
DISK PACK RECORDING		
Mode		Modified Frequency Modulation
Speed		3600 rpm ±2%
Bit Density		
Outer track		2659 bpi
Inner track		4040 bpi
Bit Rate		6.451 MHz
Data Transfer Rate (bytes/second)		806,375
DISK PACK MAXIMUM DATA CAPACITY (Data Field)*		
Bytes/Track		13,030
Bytes/Drive		100,018,280 (100 MB) 200,036,560 (200 MB)
Bytes/System		800,146,240 (100 MB) 1,600,292,480 (200 MB)

*NOTE: Capacity is reduced as the number of records per track increases, due to added gaps required to separate each record file. Capacity values given based on Home Address, Record Zero with no key field, and one data record per track.

CONTROLLER DESCRIPTION

GENERAL

The controller is a microprogrammed processor contained in one free standing cabinet. All logic circuits are ceramic high speed integrated circuits mounted on pluggable printed circuit boards.

The functions performed by the controller are:

1. Execute commands issued by the channel.
2. Control the channel and disk storage interfaces.
3. Control the transfer of read/write data between the channel and the disk pack.
4. Detect and correct errors in the count and key fields. Detect errors in the data field.
5. Provide the channel with subsystem status.
6. Run diagnostic tests on the subsystem.

Three modes of operation are available:

1. On line mode is the normal operating mode. All subsystem operations are under channel control.
2. In line mode is used to test or exercise the drive with the SP (spare) logical address plug installed. The controller time shares the selected diagnostic microprogram with the normal channel operations. The channel has priority.
3. Off line mode disconnects the controller from the channel interface. This mode is normally used for controller troubleshooting.

CONTROLLER ORGANIZATION

Memory

Basic internal controller operation is by means of a microprogram. The permanent storage of the microprogram is an easily changed magnetic tape cassette (see Figure 1-2).

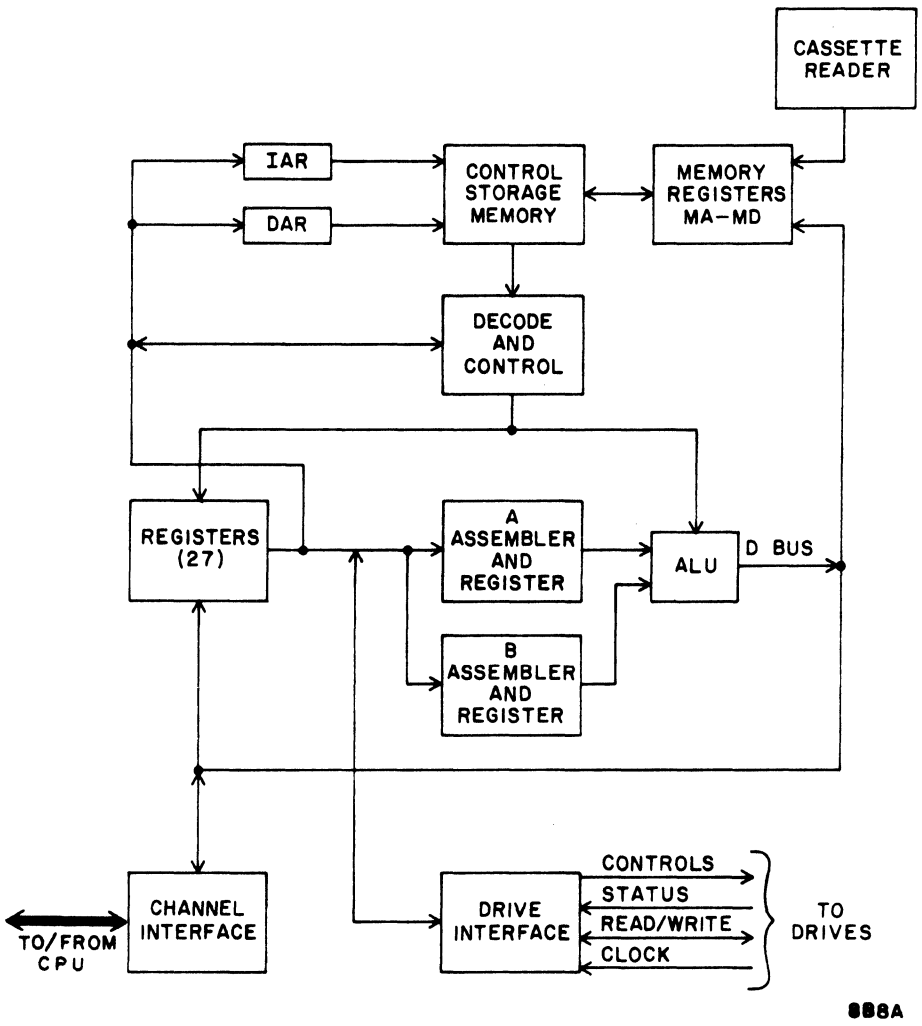


Figure 1-2. Controller Block Diagram

When the subsystem is powered up, the microprogram stored on the cassette is transferred from the cassette reader to a read/write memory. The control storage memory is 4K in size. Word length is 40 bits: 32 bits of insturctions or data and 8 bits of error correction data.

The primary use of the memory is for instruction residence; data storage is a secondary use.

One word of the microprogram is read out from the memory each 200 nsec. After decoding, the word is executed to accomplish the following functions:

1. Arithmetic or logical operations to be performed by the arithmetic-logical unit (ALU) are defined. The word also specifies the source and destination registers to be used in the operation.
2. Initiate, if necessary, a memory read or write.
3. Control I/O operations (channel and drive).
4. Select the next instruction to be executed. The next word address can be modified by the presence of another controller or interface condition; this is a branch condition test.

Access to the memory can be an instruction or a data cycle. For a data fetch or store operation, the Data Address Register (DAR) is gated to the memory. For an instruction cycle, the Instruction Address Register (IAR) is used.

The 8 bits of Error Correction Code (ECC) are generated and stored as data is fed from the cassette to memory. These bits are regenerated and compared to the stored ECC bits as each word is subsequently read out. Any one-bit error is corrected prior to command execution. Errors of two or more bits are detected, stopping the unit, but not corrected.

The cassette also stores diagnostic programs for subsystem checkout. With the controller in CE or in line mode, switches on the maintenance panel can be used to transfer a block of 256 words of diagnostic instructions from the cassette to control storage. These diagnostics exercise selected portions of the subsystem functions.

Registers

Several registers are provided to serve general or specific functions. Inputs to, and outputs from, these registers are controlled by certain fields of the microprogram word.

The registers are divided into the following categories:

- I (Interface)
- A (Arithmetic)
- S (Sink or External Input)
- E (External Output)
- M (Memory)

Arithmetic-Logical Unit

The ALU performs logic and arithmetic operations on two operands residing in the A and B registers. Under control of the OP field of the microprogram word, ALU performs the following functions: add, subtract, logical AND, logical OR, or logical Exclusive-OR. Two complete ALU's are used in parallel and their results are compared to ensure correct arithmetic operations. The output of ALU is placed on the D Bus for transfer to a general purpose register, an interface register, or a memory register.

Drive Interface

The Control Unit Drive Interface (CUDI) is used to control the transfer of control, status, or data information between the controller and its drives.

The Serializer/Deserializer (SERDES) provides the read/write interface. While reading, SERDES converts the serial (by bit) data read from the disk to parallel (by byte) data for transmission to the channel. While writing, SERDES converts parallel (byte) information from the channel into serial (bit) information for the disk.

The drive supplies an 806 kHz clock derived from the servo surface of the disk pack. This clock is used to synchronize the controller/drive interface during read or write operations.

Channel Interface

The channel interface attaches the controller to the channel. The interface maintains timing of I/O signals and data.

TRACK FORMAT

CYLINDER CONCEPT

Data is stored on the disk pack. Each pack has 12 disks mounted one-half inch apart vertically. These disks provide 19 surfaces for data recording. Data may be written on, or read from, the surfaces by read/write heads. The heads are numbered from 00 to decimal 18 (hexadecimal 12). The recording surface directly under a head is called a track.

The heads are positioned vertically with respect for each other and are all connected to one access mechanism. The access mechanism may be positioned horizontally to any one of 411 discrete positions. Any one head may be addressed at a time. Since any of the heads may be address without moving the access mechanism, the recording medium under the heads may be thought of as a cylinder. Cylinders are numbered from 000 (the track nearest the outside edge of the disk) to 410 (the innermost track).

Figure 1-3 illustrates the cylinder concept. The access mechanism is horizontally positioned so that head 00 is at track 15. Since all of the other heads are also positioned at their respective track 15, the access mechanism is considered as being at cylinder 15.

Any track may be addressed by specifying the drive module logical address, the track number and, finally, the head number.

INDEX

The logical beginning of each track is indicated by a signal called Index. It is generated by the drive at a specific circumferential location on the disk pack. All tracks share this signal. Its purpose is to serve as a common reference point to indicate the logical beginning of all of the disk pack tracks.

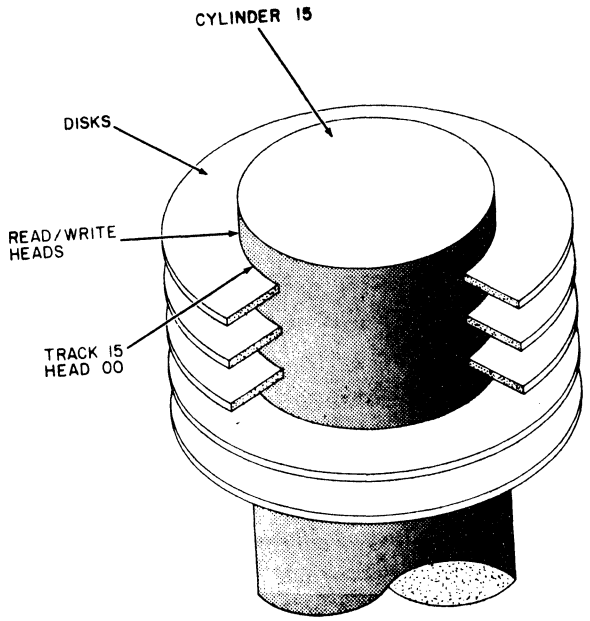


Figure 1-3. Cylinder Concept

RECORDS

The basic unit of information is the byte; each byte consists of 8 bits. Data is transferred between the channel and controller one byte at a time.

A group of related bytes is called a field. In turn, fields are grouped into logical units of information called records. Records normally consist of three fields: the count field, the key field (optional), and the data field. Gaps separate the fields.

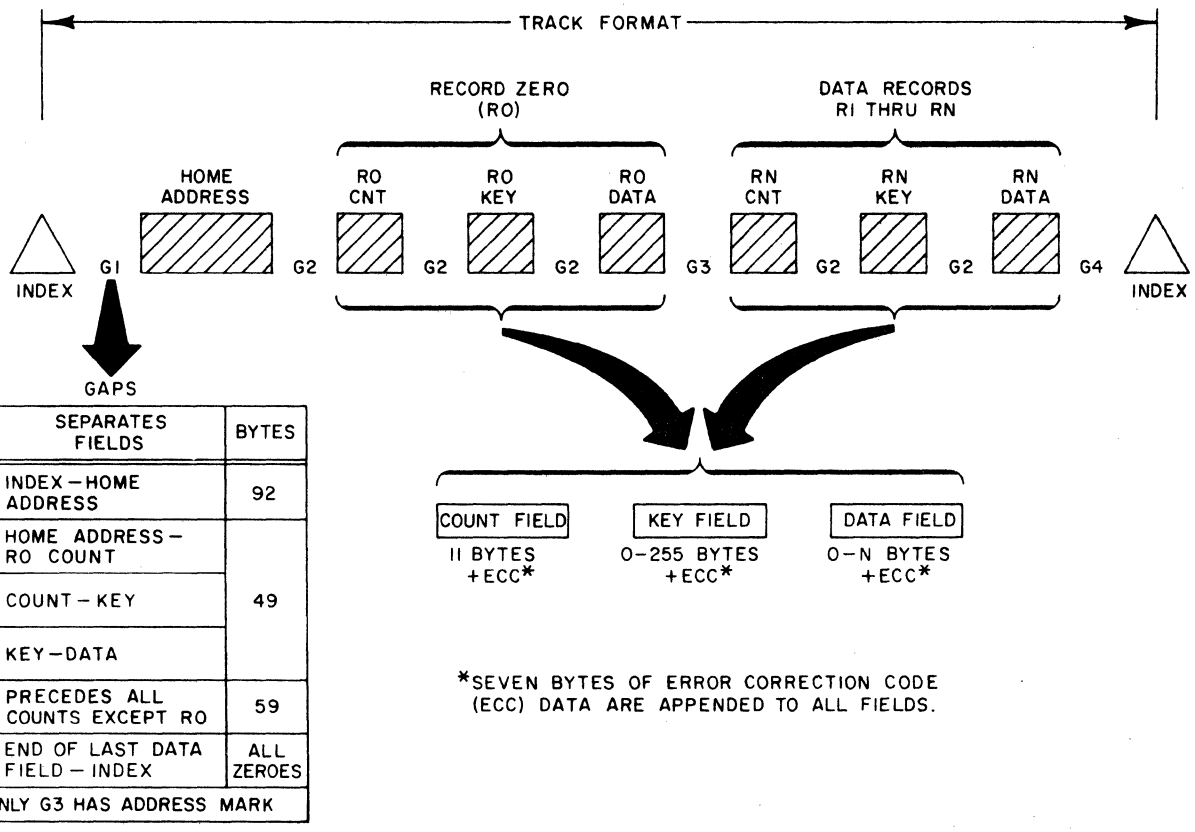
A series of similar records constitutes a logical file. The composition of records and logical files depends on the customer application.

DATA FORMAT

All tracks are formatted beginning with Index (Figure 1-4). Each track is formatted in an identical manner: Home Address, Record Zero (Track Descriptor record), and one or more data records (numbered from R1 to Rn). Record Zero and the data records normally have three fields: count field, key field, and data field.

Count Field

The count field (Table 1-2) defines the length of the key and data fields of that record. This field always



NUMBER	SEPARATES FIELDS	BYTES
G1	INDEX – HOME ADDRESS	92
G2	HOME ADDRESS – RO COUNT	49
	COUNT – KEY	
	KEY – DATA	
G3	PRECEDES ALL COUNTS EXCEPT RO	59
G4	END OF LAST DATA FIELD – INDEX	ALL ZEROES
NOTE: ONLY G3 HAS ADDRESS MARK		

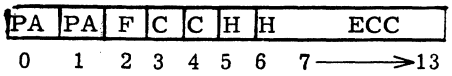
Figure 1-4. Track Format

TABLE 1-2. RECORD BYTES MEANINGS

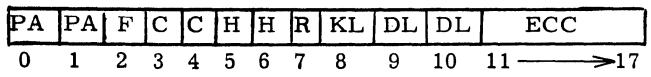
COUNT FIELD/HOME ADDRESS

Byte Configuration:

Home Address

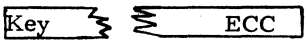
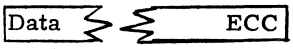


Count Field



Byte	Name	Function
0	Physical Address	Low order cylinder address.
1	Physical Address	Bit 0 = 0 Bit 1 is high order cylinder address Bit 2 = 0 Bits 3-7 are head address
2	Flag	Indicates track condition. Bits 0-3 = 0 Bit 4 = 0 except: if this is count field of an overflow data record (written by Write Special CKD), bit = 1 Bit 5 = 1 indicates this is CE pack Bit 6 = 0 indicates this is operative track Bit 6 = 1 indicates this is defective track Bit 7 = 0 indicates this is primary track Bit 7 = 1 indicates this is alternate track Record Zero is used to reposition defective tracks. If Flag Byte bits 6 and 7 = 10, CCHH bytes specify physical location of alternate track. If bits 6 and 7 = 01, CCHH bytes specify physical location of defective track. The flag byte for Home Address is provided by channel. On all other count fields, controller automatically writes flag byte by propagating it from Home Address or from another count field.
3	High Order Cylinder	Bits 0-6 = 0 Bit 7 on if cylinder is equal to, or greater than, decimal 256.
4	Low Order Cylinder	Cylinder address
5, 6	Head	Specifies the particular read/write head. Byte 5 is always zero. Byte 6 may be between decimal 0 and 18.

TABLE 1-2. RECORD BYTES MEANINGS (Cont'd)

Byte	Name	Function
7	Record Number	Designates the number of the record on the track. Record numbers may be sequential or random. The record number of the Track Descriptor Record is 0. Data records are 1 through the highest number on the track.
8	Key Length	Specifies the number of bytes (excluding ECC bytes) in the key field. Normally, Record Zero has no key, so this byte is zero.
9, 10	Data Length	Specifies the number of bytes (excluding ECC bytes) in the data field. Data length of zero is considered as End of File record.
ECC	Error Correction Code	Seven bytes of data used for error detection and correction. Bytes are generated by the controller without program intervention.
KEY FIELD		
Byte Configuration:		
DATA FIELD		
Byte Configuration:		

contains 11 data bytes plus 7 ECC bytes. The count field is generated when the record is written by a format write command.

The identifier (ID) portion of the count field includes the cylinder, head, and record number. The total ID is five bytes long.

Key Field

The key field usually contains unique information (up to 255 bytes) to identify the meaning of the subsequent data field. Use of the key field is optional; if not used, the field and its gap are omitted. The key field cannot be altered without also rewriting the data field.

Data Field

The data field contains the information identified by the count and key fields of the same record. This field can be altered without affecting any other field as long as its length is not changed; if so, the record must be reformatted.

The length of the data field is specified in the two KL bytes in the count field. A data field length of zero is considered an End of File record.

Gaps

Each record is separated from the other records by gaps (Figure 1-5). All gaps except G4 (Figure 1-4) are MFM-recorded "0" bits; G4 consists of "0" bits.

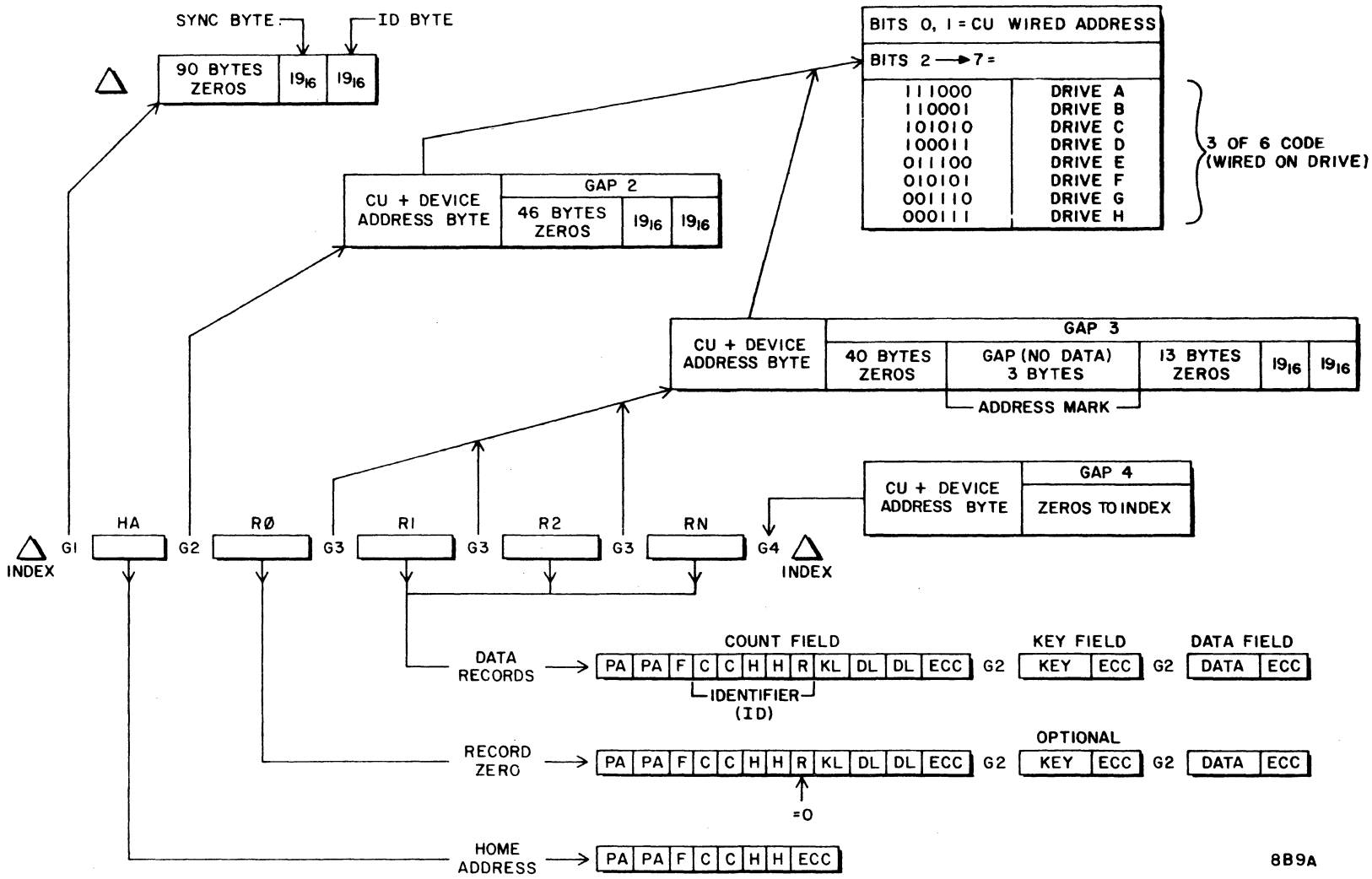


Figure 1-5. Records/Gaps Format

In addition, gaps separate the fields within records. The gaps allow time for operating mode changes between fields. For example, the gaps permit the channel to drop one command (such as a Read operation) and to raise another command (such as a Write operation) between fields of the same record.

All gaps end with two bytes of hexadecimal 19. These bytes indicate the beginning of valid data to SERDES and to the microprogram.

Gap G3 is a special gap that precedes all count fields except the count field of RO. This gap contains three bytes of unrecorded clock or data pulses, that is, there are no flux transitions recorded on the disk. This area is called the address mark (AM) to indicate the beginning of a data record.

HOME ADDRESS

The first record following Index is the Home Address (HA) record. One HA is written per track. The record contains five bytes that define physical track location and conditions.

Because of the special nature of this record, it may be operated upon only by the following commands:

- Write Home Address
- Read Home Address
- Search Home Address Equal

Home address records are normally written only during pack initialization.

TRACK DESCRIPTION RECORD (RO)

The Track Descriptor Record (Record Zero) follows the home address record. This record is normally written by utility programs concurrent with writing home address. Special read and write commands are provided that operate only on RO.

If a defective track is involved with RO, the CCHH bytes of RO specify the physical address of the alternate track. If this is an alternate track, the CCHH bytes specify the address of the defective track. The controller uses this information for internal error recovery procedures. For instance, if a defective track is encountered, the controller will issue a seek to the alternate track.

DATA RECORDS

One or more data records follow the track descriptor record. Data records are used to record customer generated information.

Depending on file organization, data records may be written with or without key fields.

DATA CHECKING

CPU PARITY

The accuracy of data transferred between the CPU and the controller is checked by associating a parity bit with each byte. Odd parity is maintained, that is, the parity bit is set to "0" or "1" so that the number of "1" bits in the byte are always odd.

ERROR CORRECTION

Memory Correction

During an IMPL operation, data from the cassette tape is checked for horizontal parity. As each microinstruction word is assembled and loaded into memory, an eight bit check byte is computed and added to each word. The check byte should agree with the byte computed as each word is read out of memory. If the bytes do not agree, a memory error is indicated. A one bit error is automatically corrected; no error is indicated internally or externally to the logic. Bit errors of two or more cause the controller to stop.

Data Correction

As data is written on the disk, seven bytes of serially calculated error correction (ECC) data are appended at the end of all fields. During subsequent read or search operations, the error check bytes are recalculated. An error has occurred if the recalculation does not match the written ECC bytes. If the error is correctable (error burst of 11 bits or less) and is in the HA, Count, or Key fields, the controller: corrects the data, requests Command Retry, reorients to the record with the error, and substitutes the corrected data. If the correctable error is in the Data field, the controller only calculates the error displacement; the processor corrects the erroneous data. The operation is retried if the error is uncorrectable.

CHANNEL COMMANDS

COMMAND SUMMARY

A condensed list of channel commands is provided in Table 1-3.

CONTROL COMMANDS

Control commands cause operations that do not transmit read/write data between the channel and the controller. These operations include positioning the access mechanism and selecting the head.

TABLE 1-3. COMMAND SUMMARY

Command Name	Hex Code		Function
	Single Track	Multi-Track	
CONTROL COMMANDS			
Seek Seek Cylinder	07 0B	-- --	Move access to selected cylinder and select specified head.
Seek Head	47	--	Select specified head.
Enable Retry	57	--	Releases retry reserved device that stopped execution because of errors and has requested previous command be retried. The retry operation starts.
Set File Mask	67	--	Defines permitted: writes, seeks, and retry/error corrections.
Space Count	97	--	Allow bypassing of defective count field for recovery of data in key/data fields.
Device Release	A7	--	Terminate reservation status of drive.
Device Reserve	B7	--	Reserve selected drive.
Restore	C7	--	Move access to cylinder zero and select head zero.
SENSE COMMANDS			
Sense 1	01	--	Cause controller to return Operational Status Bytes 1, 2, 3.
Read Error Log	41	--	Cause controller to transfer data, error, and error retry logs.
Sense 2	81	--	Cause controller to return Diagnostic Status Bytes 4 through 24.

TABLE 1-3. COMMAND SUMMARY (Cont'd)

Command Name	Hex Code: Single Track	Hex Code: Multi-Track	Function
			SEARCH COMMANDS
Search Home Address Equal	33	3B	Locate Home Address field equal to argument specified by system.
Search ID Equal	53	5B	Locate Count field (CCHHR bytes) equal to argument specified by system.
Search ID High	73	7B	Locate Count field higher than argument specified by system.
Search ID Equal or High	93	9B	Locate Count field equal to, or higher than, argument specified by system.
Search Key Equal	B3	BB	Locate Key field equal to argument specified by system.
Search Key High	D3	DB	Locate Key field higher than argument specified by system.
Search Key Equal or High	F3	FB	Locate Key field equal to or higher than, argument specified by system.
			READ COMMANDS
Read Initial Program Load	05	--	Seek drive to cylinder 0 and head 0; wait for Index; read R1 data field.
Read Home Address	25	2D	Read five bytes (FCCHH) of Home Address field.
Read Record Zero	45	4D	Read Count, Key, and Data fields of Record Zero.
Read Key & Data	65	6D	Read Key and Data fields of record: a. If chained, read area immediately following count area of same record. b. If not chained, read area after next address mark (gap 3).
Read Data	A5	AD	Read Data field (chaining same as Read Key & Data).
Read Count, Key & Data	85	8D	Read next record (excluding HA or RØ).
Read Track	CD	--	Read all fields on one track or, if chained from another command, transfers data from next field to end of track.
Read Count	E5	ED	Read Count field following next address mark.

TABLE 1-3. COMMAND SUMMARY (Cont'd)

Command Name	Hex Code		Function
	Single Track	Multi-Track	
WRITE COMMANDS			
Write Special Count Key & Data	03	--	Write segment of overflow record.
Write Home Address	23	--	Write five bytes (FCCHH) of Home Address record.
Write Record Zero	43	--	Write all fields of Record Zero.
Write Key & Data	63	--	Write Key and Data fields of any record other than HA or RØ.
Write Count, Key & Data	83	--	Write all fields of any record other than HA or RØ.
Write Data	A3	--	Write Data field of any record other than RØ.
Write Count	E3	--	Special maintenance write command to check ECC circuits: write Count field or record R1 through Rn by transferring all 18 bytes from PA through ECC) of field.
Erase	E7	--	Write zeros from next gap 3 until Index.

SENSE COMMANDS

Sense commands are used to report subsystem status. Twenty four sense bytes are available. The first three sense bytes are called Operational Status Bytes (OSB1 through OSB3) and are reported in response to a Sense 1 command. The remaining 21 bytes are the Diagnostic Status Bytes (DSB4 through DSB24); they are reported in response to a Sense 2 command. A brief summary of byte/bit meanings is contained in Figure 1-6.

SEARCH COMMANDS

Search commands cause the controller to compare the information coming from the channel with information read from the drive. Data coming from the channel (which is operating in write mode) is called the search argument. This data is compared with data read from the drive. Status returned to the channel indicates whether or not the search criteria are met. If they are (for example, if the compare was equal), the Status Modifier bit of the Standard Device Byte is on. This causes the channel to skip to the next CCW in the chain to perform the next operation.

The search comparison is performed one record at a time. An unsuccessful search will result in the Status Modifier bit remaining off. Normally, the Transfer-in-Channel (TIC) command is used to automatically reissue the command in a chain as follows:

Search Key Equal
TIC
Write Data

Status Modifier, when set by the controller in conjuntion with End, causes the TIC to be skipped so that the Write Data command can be executed. The controller remains oriented with respect to track location to permit searching each record on a track sequentially. This process can continue until the search is satisfied or an error status is returned because the entire track has been search unsuccessfully.

Search commands may operate either on one track or on all tracks within a cylinder. This function is controlled by the 3 bit of the command: if it is set to "1", incremental head switching at Index is automatic. Automatic head switching is not executed outside of a command chain. Searching continues until the search is successful or all tracks have been searched.

READ COMMANDS

Read commands transfer data read from the drive to the channel. A parity bit is added to the byte to supply odd parity.

Read commands may operate either on one track or on all tracks within a cylinder. This function is controlled by the 3 bit on all read commands except Read track.

WRITE COMMANDS

Write commands transfer data from the channel to the drive. After each field is written, the controller automatically adds seven ECC bytes to the field. These bytes are used during subsequent read commands to ensure accuracy. Write commands cannot be issued or executed with the associated drive offset from the theoretical track. All write commands require prior file mask enabling.

Write commands can format the write records too large for a single track; these are called overflow records.

There are two types of write commands: format and update. Format write commands are used to initialize tracks and records; they also establish the length of the key/data fields of each record. All write commands except Write Count, Write Data, and Write Key and Data are format write commands.

After a format write has completed writing the data associated with the command, it writes valid zeros in the remainder of the track to Index which defers execution of subsequent commands until after Index. The only exception is if the write is chained to another format write.

The three update write commands are Write Count, Write Data and Write Key & Data. The first command is used to diagnose hardware malfunctions, the last two to update records that have already been written. Update writes must be chained from a valid search command.

ASSEMBLY LOCATIONS

Figure 1-7 illustrates the major assemblies within the controller. Detailed information on the functions of these assemblies is provided in Section 3 of this manual.

Each of the major electrical assemblies has been assigned a reference designation. These reference designations are as follows:

- A1 - Top Backpanel
- A2 - Middle Backpanel
- A3 - Bottom Backpanel
- A4 - Maintenance Panel
- A5 - Logic Chassis Miscellaneous
- A6 - I/O Panel
- A7 - Control Panel
- A8 - Cassette Deck
- A9 - Power Control Unit

Seven dc power supplies have received reference designations of PS1 through PS7.

OPERATIONAL STATUS BYTES			
BIT	1	2	3
2^7	Read or Write Error	Not Used	Environmental Data Present
2^6	Service Request Not Honored	Overflow Incomplete	Equipment Check
2^5	Seek Check	Missing AM	Not Used
2^4	Transmission Error	File Protected	Not Used
2^3	Track Check	Not Found	Device Reserved to Other Channel
2^2	Automatic Head Switching Error	Invalid Sequence	Correctable Data Field Error
2^1	End of File	End of Cylinder	Additional Error Information Indicator
2^0	Command Code Reject	Track End	Permanent Read/Write Error

Figure 1-6. Status Bytes (Sheet 1 of 7)

DIAGNOSTIC STATUS BYTES

4	5	6	7	8
PHYSICAL ID	**HI ORDER BITS**	**CYLINDER**	**HEAD**	**FORMAT/MESSAGE**
Bits 2^7, 2^6 = CU address wired in at installation Bits 2^5 thru 2^0 = Drive ID (3 of 6 Code) Code Drive 111 000 A 110 001 B 101 010 C 100 011 D 011 100 E 010 101 F 001 110 G 000 111 H	2^7 = Reverse 2^5 = CAR 512 2^4 = CAR 256 2^2 = DIFF 512 2^1 = DIFF 256	Low order bits of last seek address received from channel	*2^4 thru 2^0 = Head Address of last seek *Excluding retry seeks	Bits 2^7 thru 2^4 = Format of bytes 9 thru 24 0000 = Format 0 = Programming or system check 0001 = Format 1 = Equipment check 0011 = Format 3 = Hard check errors 0100 = Format 4 = Uncorrectable data check 0101 = Format 5 = Correctable data check 0110 = Format 6 = Usage/Format statistics Refer to remaining sheets of this figure for format byte/bit meanings. Format 0 does not use bytes 9 thru 24. Format 2 is not used. Bits 2^3 thru 2^0 are decoded to form message. Meanings of messages depend upon format decoded.

Figure 1-6. Status Bytes (Sheet 2 of 7)

DSB8 MESSAGES

MESSAGE DECODE	FORMAT 0	FORMAT 1	FORMAT 3	FORMAT 4	FORMAT 5	FORMAT 6
0	No Message	No Message	No Message	HA Field ECC uncorrectable	HA Field correctable	No Message
1	Invalid command	Sector Wraparound	Unused	Count field ECC uncorrectable	Count field correctable	Unused
2	Unused	Unused		Key field ECC uncorrectable	Key field correctable	
3	CCW count truncated	No write gate at drive		Data field ECC uncorrectable	Data field correctable	
4	CCW data incorrect	No write current sense		HA field no sync byte found	Unused	
5	CCW count too large	Cylinder Wraparound		Count field no sync byte found		
6	Unused	High cylinder/high difference Wrap-around		Key field no sync byte found		
7	Retry command incorrect	Head Wraparound		Data field no sync byte found		
8	Read track-sub-quent record	Difference Wraparound		Unused		
9	Hard command not as expected	File status incorrect		AM detection failure on retry		
A	Invalid track format	Seek error		① Restart command equals read data or overflow incomplete.		
B	Improper alternate track pointer	Physical address verification seek check		② Restart command equals write data on overflow incomplete.		
C	SERDES malfunction -no ST4	No interrupt from drive		Unused	Unused	
D	Permanent error during retry	Permanent error during retry	Permanent error during retry	Permanent error during retry	Permanent error during retry	Permanent error during retry
E	Unused	ECC-P1 or P3 error	Unused		Unused	Unused
F	Sector counter/ orientation counter disagree	ECC-P2 error		Retry inhibited		

Figure 1-6. Status Bytes (Sheet 3 of 7)

FORMAT 1 DISK DRIVE EQUIPMENT CHECK

BIT	BYTE 9	BYTE 10	BYTE 11	BYTE 12	BYTE 13	BYTE 14	BYTE 15	BYTE 16
	MODULE STATUS	MONITOR MODE	MONITOR STATE	CHECK STATUS	FAULT	FILE BUS OUT	FILE BUS IN	TAG BUS (IB REGISTER)
2^7	Index Error	Not Used	8	CE Program	Data Fault	If message decode is 9, expected data is posted here. If message decode is not 9, contents of EC register is posted.	Contents of ID Register	Contents of IB Register
2^6	Offset Active	Diagnostic 4	7	Speed	Servo Fault			
2^5	Seek Incomplete	Diagnostic 2	6	Motor On	Temp Fault			
2^4	Seek Complete	Diagnostic 1	5	Local	Neg Voltage Fault			
2^3	On Line	Not Used	4	CUDI Bus Out Parity	Pos. Voltage Fault			
2^2	Attention	Mode 4	3	Monitor Check	Air Flow Fault			
2^1	Busy	Mode 2	2	Pack On	Not Heads Loaded			
2^0	Record Ready-Search in Progress	Mode 1	1	Command Reject Drive	Even Cylinder			

BYTES 17, 18	BYTE 19	BYTE 20	BYTE 21	BYTE 22	BYTES 23, 24
NOT USED	CONTROL CHECK 1	SERDES CHECK (CONTROL CHECK 2)	ECC CHECK (CONTROL CHECK 3)	CUDI CHECK (CONTROL CHECK 4)	ERROR SYMPTOM CODE
	Buffer Input Parity Check	Not Used	No Input Data Received	Drive Selection Failure	Error Symptom Code
	Data Transfer Check	Write Parity Check	PØ or Write Error	Tag Invalid	
	Buffer Address Check	Read Parity Check	P1 or P3 Error	Device Check	
	Buffer Output Parity Check	Bit Ring Check	P2 Error	Not Used	
	Interface Check A	Write Compensation Check	Zero	CUDI Bus In Check	
	Interface Check B	PLO Input Check (Missing Servo)	Zero	Sector Count Check	
	Not Used	VFO Input Check	Zero	Not Used	
	Not Used	VFO Phase	Zero	Not Used	

Figure 1-6. Status Bytes (Sheet 4 of 7)

FORMAT 3 HARD CHECK ERRORS

BIT	BYTE 9	BYTE 10	BYTE 11	BYTE 12	BYTES 13 THRU 22	BYTES 23, 24
2^7	High order address byte of controller word (FAR) when error was detected	Low order address byte of controller word (FAR) when error was detected	Memory Parity Error Byte 0	A Register Check	Not Used	Error Symptom Code
2^6			Memory Parity Error Byte 1	B Register Check		
2^5			Memory Parity Error Byte 2	Not Used		
2^4			Memory Parity Error Byte 3			
2^3			Clock Check			
2^2			DAR Check			
2^1			IAR Check			
2^0			ALU Check			

Figure 1-6. Status Bytes (Sheet 5 of 7)

FORMAT 4 DATA CHECKS NOT PROVIDING DISPLACEMENT INFORMATION

BYTE 9	BYTE 10	BYTE 11	BYTE 12	BYTE 13	BYTE 14	BYTE 15	BYTE 16
CYLINDER HI	CYLINDER LO	HEAD HI	HEAD LO	RECORD	SECTOR	OFFSET	RETRIES
High order cylinder byte of last seek address	Low order cylinder byte of last seek address	High order head byte of last seek address	Low order head byte of last seek address	Record number of record in error	Sector number of sector in error	Amount of offset required to recover from error	Number of retries required to recover from error

BYTE 17	BYTES 18 THRU 22	BYTES 23,24
SOURCE DRIVE INFORMATION		ERROR SYMPTOM CODE
Read from ID byte written before each field Bits 2^7, 2^6=CU address wired in at installation Bits 2^5 thru 2^0= Drive ID (3 of 6 code) Code — Drive 111 000 — A 110 001 — B 101 010 — C 100 011 — D 011 100 — E 010 101 — F 001 110 — G 000 111 — H	Not Used	Error Symptom Code

Figure 1-6. Status Bytes (Sheet 6 of 7)

FORMAT 5 DATA CHECKS PROVIDING DISPLACEMENT INFORMATION

BYTES 9 THRU 15	BYTES 16, 17, 18	BYTES 19, 20	BYTES 21, 22, 23	BYTE 24
	RESTART DISPLACEMENT	ERROR DISPLACEMENT	ERROR PATTERN	TRUNCATION
Same as Format 4	Number of bytes processed to end of data field in error	Error displacement of first byte in error relative to end of data field in error	Error pattern used for error correction	Bits 2^7 thru 2^1= Not Used Bit 2^0=Channel Truncation

FORMAT 6 ERROR/USAGE STATISTICS

BYTES 9 THRU 12	BYTES 13, 14	BYTES 15, 16	BYTES 17, 18	BYTE 19	BYTE 20	BYTE 21	BYTE 22
BYTES READ	CORRECTABLE DATA CHECKS	RETRY DATA CHECKS	SEEKS		SEEK ERRORS	COMMAND OVERRUN A	DATA OVERRUN A
Number of key/data bytes processed in read or search operations. Does not include retry operations	Number of ECC correctable data checks	Number of ECC uncorrectable data checks successfully retried	Number of access motions initiated by channel	Not Used	Number of seek errors unsuccessfully retried	Number of command overruns retried for Channel A	Number of data overruns retried for Channel A

BYTE 23	BYTE 24
COMMAND OVERRUN B	DATA OVERRUN B
Number of command overruns retried for Channel B	Number of data overruns retried for Channel B

Figure 1-6. Status Bytes (Sheet 7 of 7)

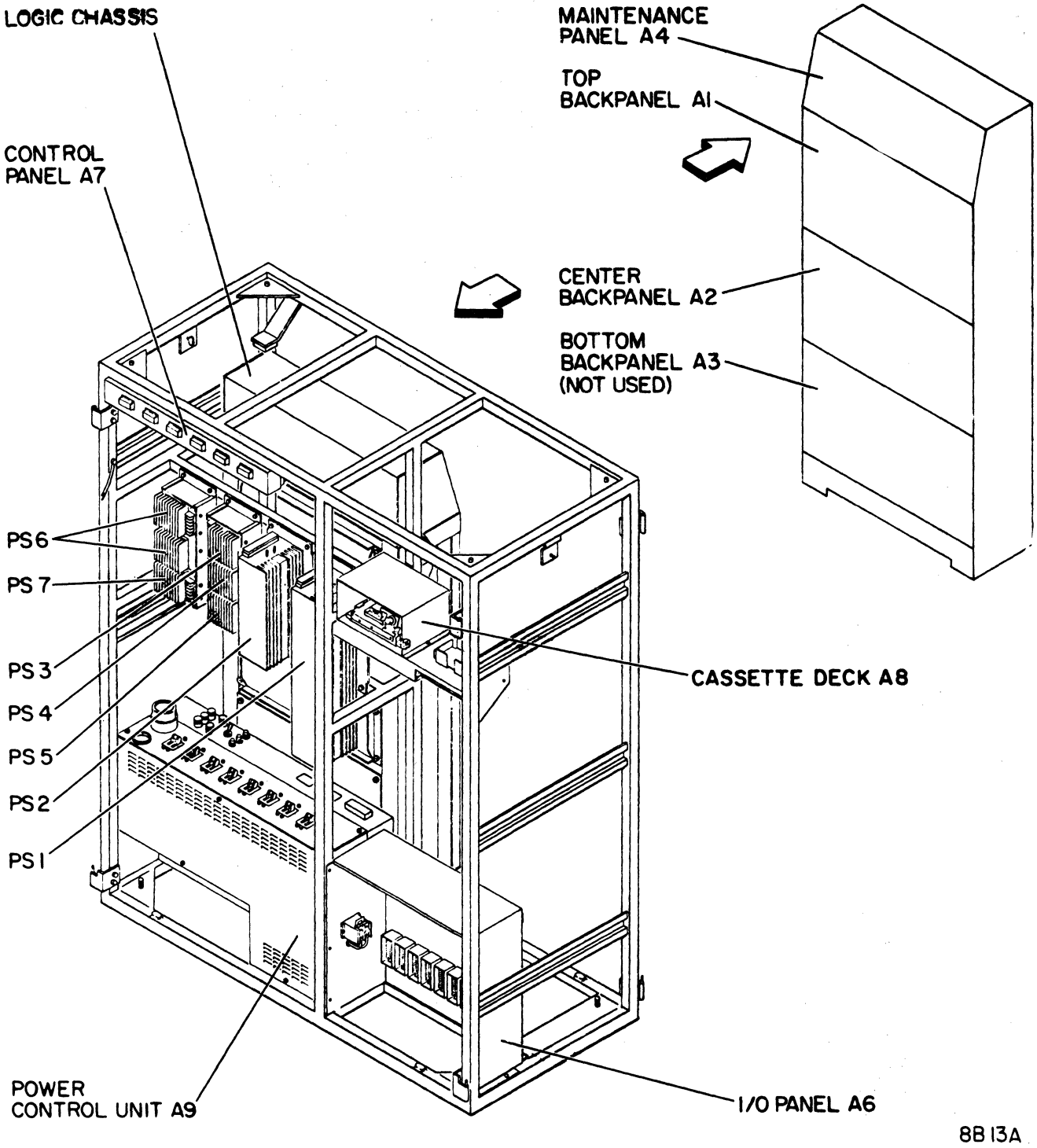


Figure 1-7. Controller Assemblies Locator

LOGIC CHASSIS

The logic chassis serves as the mounting point for the main complement of the logic cards. The chassis is hinge-mounted for easy access to the backpanel terminals. The backpanel terminals provide ready access for monitoring of all signals entering and leaving each card.

Two backpanels are used: top backpanel A1 and middle backpanel A2. Bottom backpanel A3 is not used. Each backpanel provides space for 24 printed circuit cards. Cards mounted in A1 serve as the basic arithmetic, control, and control storage memory functions of the controller. Cards mounted in A2 are primarily used in the channel and drive interfaces; also, cards needed to process read/write data and to control the cassette reader are located in A2. Twenty-four auxiliary connectors are positioned around the perimeter of each backpanel. These connectors are used to route signals between the backpanels and other electrical components.

The Key to Logic portion (Cross Reference Numbers in the 0000 class) of the logic diagrams fully explains the logic chassis numbering scheme.

Two cooling fans are located at the bottom of the logic chassis.

The maintenance panel (A4) contains switches needed to maintain and troubleshoot the controller. A printed circuit card mounted within the panel contains several light-emitting diodes (LEDs) to display controller and subsystem status. Refer to Section 2 of this manual for an explanation of the switches and indicators.

I/O PANEL

The I/O panel (A6) contains connectors for cables to the channel and to the drives. Connector J1 connects the controller to channel A; J10 to channel B.

Two I/O cables are needed for each drive. Most of the controller/drive signals are via the signal cables (J11 through J18). The data cables (J2 through J9) have coax conductors for three signals: Read Data, Write Data, and Servo Data.

CONTROL PANEL

The control panel (A7) mounts switches required by the system operator.

CASSETTE DECK

The cassette deck (A8) contains the tape cassette reader and its associated electronics. An easily changed tape cassette contains the microprogram instructions required for controller operation. The cassette drive is energized during IMPL operations to transfer the microprogram from the cassette to the control storage located at A1BB through A1BL. The cassette is also used to load in microprogram diagnostics under control of the maintenance panel switches.

POWER CONTROL UNIT

The power control unit (A9), or PCU, functions as the ac power switching and distribution unit.

Three-phase ac power is applied to terminal board TB01. This power is distributed to the drives through TB04 and TB05; ac distribution to the dc power supplies and to the logic chassis fans is via TB03.

Other control functions provided by the PCU are:

1. Subsystem power on/off control (via switches on the maintenance and control panels).
2. Drive spindle motor on/off control. When the subsystem is first powered up, the drive spindle motors are turned on in groups of three, each group power-on enable has a 10-second delay.
3. Power On Reset (POR) to the controller logic.
4. COP/DIP power of -24v from the PCU internal dc supply.
5. Error detection of ±5v and ±24v power from the external dc supplies.
6. A 3 vac signal used as an IMPL delay clock and as an inline control clock.

DC POWER SUPPLIES

Seven dc power supplies (PS1 through PS7) supply all dc voltages used within the controller. Their ac inputs are controlled by the PCU.

PS1 is rated +5v at 100 amperes. It supplies power to the A1 backpanel.

PS2 is rated +5v at 50 amperes. It supplies power to: A2 backpanel and cassette reader, and drive to the control panel switches.

PS3 is rated -5v at 5 amperes. It supplies power to the cards at A2AA through A2AH.

PS4 is rated -24v at 1.5 amperes. It supplies power to the channel receivers/transmitters at A2BL and A2BM.

PS5 is rated +24v at 1.5 amperes. It also supplies power to the channel receivers/transmitters at A2BL and A2BM.

PS6 is rated -24v at 4.5 amperes. It supplies +24v driver power to the control panel lamps and -24v power to the cassette reader.

PS7 is rated +24v at 4.5 amperes. It supplies power to the cassette reader.

SECTION 2

OPERATION

GENERAL

This section provides descriptions of the Maintenance and Control panels. It also details various procedures used in the maintenance and troubleshooting of the subsystem. Procedures included cover: Subsystem power on/off, microprogram loading, removing controller form channel interface, the CE function, lamp test, register loading, fetching and altering of memory, and the selecting of stop/sync points. For procedures used in running the hardcore and inline diagnostics, refer to the Subsystem Troubleshooting manual.

MAINTENANCE PANEL

The Maintenance panel, illustrated in Figure 2-1, permits intervention in normal controller and drive operations. Operation may be either on line, in line, or off line. Operation is controlled by the Maintenance Panel FUNCTION switch and the Operators Panel KANAL A GESP and KANAL B GESP switches. In all three FUNCTION switch positions (CE/NORMAL/IN LINE) the controller is available to the channel. In order to break the channel/controller interface the KANAL A and B GESP switches must be set to the disable position. In order to protect customer data, the CE and IN LINE FUNCTION switch positions should not be used while customer disk packs are on line. Refer to Table 2-1 for an explanation of switch and indicator functions.

TABLE 2-1. MAINTENANCE PANEL SWITCH AND INDICATOR FUNCTIONS

Switch/Indicator	Enable In Function	Description
FUNCTION switch	All	Controls machine CE, NORMAL, and IN LINE functions.
CE position		All panel switches are active.
NORMAL position		CHECK RESET/LAMP TEST is only active control.
IN LINE position		CHECK RESET/LAMP TEST, EXECUTE, and CONTROL STORAGE ADDRESS switches are active.
CONTROL STORAGE ADDRESS SWITCHES	CE and IN LINE	Controls access to an Instruction or Data Address register. Function of individual switches depends on settings of FUNCTION and MODE SELECT switches.
IN LINE FUNCTION		Controls loading of In Line routine information.
DATA/ROUTINE/ PARAMETER		Two switches used together to load hexadecimal coded data byte. Used in both CE and IN LINE functions.
OPERATION switch	CE	Controls operation of machine in relation to errors.
NORMAL position		Machine stops only on Hard Check errors.
RUN ON ERROR position		Machine does not stop on any error.
STOP/SINGLE STEP position		Machine performs one microblock for every actuation of EXEC switch.
STOP ON ERROR position		Machine stops on all errors, both Hard Check and Control Check.

TABLE 2-1. MAINTENANCE PANEL SWITCH AND INDICATOR FUNCTIONS (Cont'd)

Switch/Indicator	Enable In Function	Description
MODE SELECT switch	CE	Selects one of nine modes to be performed. All modes, except RUN, RECY, and LOAD COMP must be performed with OPERATION switch in STOP/SINGLE STEP position.
RECY position		Allows the program to cycle between two preselected addresses.
RUN position		Allows the program to return to a normal run mode.
LOAD DAR position		Used to load the Data Address register.
LOAD COMP position		Used to load the Compare register.
LOAD DATA position		Used to load information into selected registers.
STR MEM position		Used to store information which has previously been loaded in memory registers MA through MD.
FETCH MEM position		Used to retrieve content of a Data Address register.
IMPL position		Used to initiate the microprogram loading.
REG SELECT switch	CE	Selects one of 32 registers. Each switch position has two registers specified. Position of INNER/OUTER switch determines which of the two register is selected.
DISPLAY BUS SELECT switch	CE	Selects one of eight sources to be indicated on the Display Bus.
RESET switch	CE	Returns microprogram to address 0000, and clears IAR, BAR, and DAR.
CHECK RESET/LAMP TEST switch	All	In CHECK RESET position error latches are cleared and ERROR DISPLAY and CMPR indicators are extinguished. In LAMP TEST position all panel lamps are turned on.
INNER/OUTER switch	CE	Selects between inner and outer ranks of REG SELECT switch positions.
DISPLAY BUS indicators	All	Sixteen indicators which display contents of register or memory location selected by DISPLAY BUS SELECT switch. With switch set to RTN/ERR position routine code is displayed on eight left hand indicators and error code is displayed on eight right hand indicators. With switch in A/B REG position A bus content is displayed on left hand indicators and B bus content on right hand indicators.
REGISTER DISPLAY indicators	CE and IN LINE	Nine indicators which display content of register selected by REG SELECT switch.

TABLE 2-1. MAINTENANCE PANEL SWITCH AND INDICATOR FUNCTIONS (Cont'd)

Switch/Indicator	Enable In Function	Description
ERROR DISPLAY indicators	All	Eleven indicators display occurrence of various error conditions.
EXEC DELAYED indicator	CE and IN LINE	Indicates that microprogram is currently unable to service operation commanded. When program is free to branch to commanded operation, indicator will go out.
IMPL indicator	CE	Indicates that microprogram is being loaded into memory from cassette. Lamp goes out when bootstrap is loaded.
CMPR 1 indicator	CE	Lights each time address in IAR compares with address in CONTROL STORAGE ADDRESS switches. It resets with CHECK RESET/LAMP TEST switch.
STOP/SYNC 1 switch	CE	In STOP position, machine halts when IAR address compares with address in CONTROL STORAGE ADDRESS switches. In SYNC position machine does not stop, but a pulse is available at test point each time compare occurs.
STOP indicator	All	Lights whenever machine stops, whether due to a malfunction or an address stop.
CMPR 2 indicator	CE	Lights each time address in IAR compares with address in Compare register, is reset with CHECK RESET/LAMP TEST switch.
STOP/SYNC 2 switch	CE	Same as STOP/SYNC 1, except that comparison is with Compare register.
CHANNEL TAGS indicators	All	Indicates status of channel interface. Indicators light when command is active.
POWER ON/OFF switch	CE	Turns controller power on and off. In power on sequence, IMPL command is issued.

POWER ON/OFF PROCEDURE

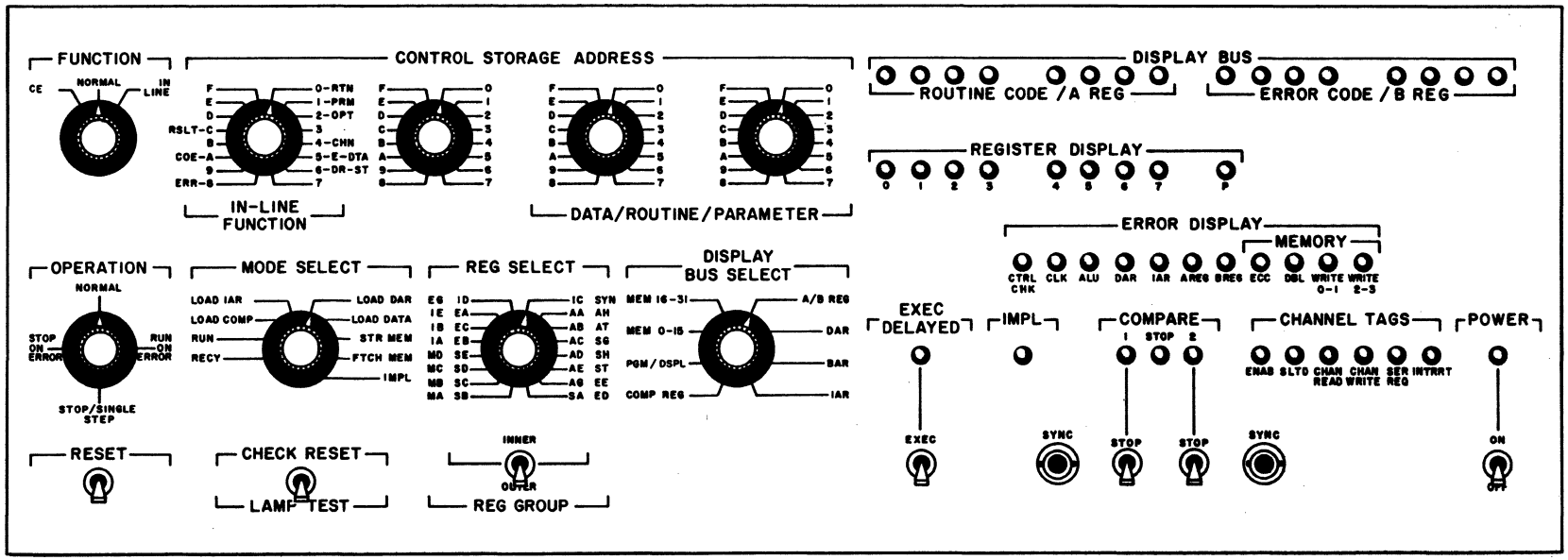
Subsystem power is controlled at the controller.

CAUTION

Turning power on or off at the controller causes noise generation on the Channel Interface lines. In order to prevent the noise generation, the following procedures must be observed.

Power On

1. Set the Operator's panel KANAL A GESP, KANAL B GESP, and STOP switches to disable the channels if circuit breakers are on (switches light when set to disable position).
2. Ensure that all circuit breakers are set to on.
3. Set FUNCTION switch to CE.



8BIOC

Figure 2-1. Maintenance Panel

4. Set POWER switch to ON. The POWER ON indicator shall light.
5. Set FUNCTION switch to NORMAL.
6. If controller/channel communication is desired, set KANAL A GESP, KANAL B GESP and STOP switches to enable channels.

Power Off

1. Press STOP switch and wait 10 seconds before proceeding.
2. Press KANAL A GESP and KANAL B GESP switches to disable channels.
3. Press UNABH switch to put controller in local mode (switch lighted).
4. Set FUNCTION switch to CE.
5. Set POWER switch to OFF.

MICROPROGRAM LOADING

The microprogram is automatically loaded each time a power on sequence is executed. The first record loaded is a bootstrap loader that controls the remainder of the sequence. Next, five hardcore diagnostics are loaded. These check the register, ALU, and Memory functions. Assuming that the hardcores are successfully loaded, the functional microprogram is then loaded. The microprogram may also be loaded on command from the Maintenance Panel, without going through the power off/power on sequence. Manually load the microprogram as follows:

1. Set FUNCTION switch to CE.
2. Set MODE SELECT switch to IMPL.
3. Set DISPLAY BUS SELECT switch to DAR.
4. Actuate EXEC switch. The IMPL indicator will light and the first three hardcores and bootstrap loader will be loaded. When loading is completed, the IMPL indicator goes out and the DISPLAY BUS indicates 03FC.
5. Set MODE SELECT switch to RUN.
6. Actuate EXEC switch. The hardcore diagnostics already loaded are executed and remainder of hardcores are loaded and executed. As the ECC tests are executed, DISPLAY BUS appears as though it contains first 037C and then 0EBC. Program stops with the DISPLAY BUS indicating 09EO.
8. Actuate EXEC switch. The functional microprogram will be loaded. Assuming that the program is loaded successfully, it will go into the basic wait loop. This will be indicated by the DISPLAY BUS appearing as though it contained 13BC (program is actually cycling through the addresses in the wait loop).

REMOVING CONTROLLER FROM INTERFACE

The controller may be removed from the channel interface lines for either CE or IN LINE functions. This is accomplished by pressing the Operators panel KANAL A GESP and KANAL B GESP switches, then STOP switch, wait 10 seconds and press UNABH switch.

CE FUNCTION

CAUTION

Do not use the CE function if customer disk packs are on line. Use only the CE disk packs or customer data may be destroyed.

Select CE function by setting FUNCTION switch to CE. This function is normally used for controller checkout, although the In Line diagnostics may be run in this function to checkout a drive.

LAMP TEST

Check the Maintenance Panel lamps before performing any maintenance. Set the CHECK RESET/LAMP TEST switch to LAMP TEST position to verify proper lamp operation. All panel lamps shall light.

MANUAL REGISTER LOADING

All controller registers except ID may be manually loaded from the Maintenance Panel. Load registers as follows:

1. Set FUNCTION switch to CE.
2. Set OPERATION switch to STOP/SINGLE STEP.
3. Set MODE SELECT switch to LOAD DATA.
4. Set REG SELECT and INNER/OUTER switches to desired register.
5. Set two DATA/ROUTINE/PARAMETER switches to the value to be loaded.
6. Actuate EXEC switch.
7. Verify correct loading by observing REGISTER DISPLAY. Register contents, including parity bit (which is automatically loaded) are displayed on the REGISTER DISPLAY. If loading takes place correctly, the REGISTER DISPLAY will change to the value being loaded, when the EXEC switch is actuated. ST Register does not have parity.

FETCHING CONTROL STORAGE MEMORY

Any control storage address in the microprogram may be accessed and displayed. Fetch addresses as follows:

1. Set FUNCTION switch to CE.
2. Set MODE SELECT switch to LOAD DAR.
3. Set CONTROL STORAGE ADDRESS switches to desired memory address.
4. Actuate EXEC switch.
5. Verify correct accessing by setting DISPLAY BUS SELECT switch to DAR. If accessing was correct, the DISPLAY BUS will display the selected address.
6. Set MODE SELECT to FETCH MEM (memory).
7. Actuate EXEC switch.
8. Set DISPLAY BUS SELECT switch to MEM 0-15, then to MEM 16-31. In each position the corresponding memory bits are displayed on the DISPLAY BUS.

ALTERING CONTROL STORAGE MEMORY

The data in any control storage memory address may be changed. Change memory locations as follows:

1. Perform memory fetching procedure previously described.
2. Using REG SELECT and INNER/OUTER switches, select MA register.
3. Set the two DATA/ROUTINE/PARAMETER switches to desired value to be loaded.
4. Actuate EXEC switch. This loades the value set into the DATA/ROUTINE/PARAMETER switches into the register selected in step 2.
5. Repeat steps 2, 3, and 4 for the MB, MC, and MD registers respectively.
6. Set MODE SELECT switch to STR MEM (store memory).
7. Actuate EXEC switch. This stores the data loaded in the MA, MB, MC, and MD registers at the address selected in step 1.
8. Verify correct memory loading by setting MODE SELECT switch to FETCH MEM, actuating EXEC switch, and displaying memory bits 0-15 and 16-31 respectively.

SELECTING STOP/SYNC POINTS

The Maintenance Panel provides two STOP/SYNC controls. These controls are used to stop the microprogram when a selected address is loaded into IAR or to cause it to cycle between two addresses. The switches allow the selection of either a stop or a sync mode of operation. In the SYNC position, a 40 nsec wide, +3v pulse is available at the SYNC test point when the microprogram reaches the selected address; in addition, the CMPR lamp lights when the address is reached. One STOP/SYNC control may be used to either stop the microprogram or signal that an address has been reached (by lighting the

CMPR lamp and outputting a pulse). Using both controls in the SYNC position and the RECY position of MODE SELECT switch allows cycling through selected segments of the program. STOP/SYNC 1 triggers on the address set in the CONTROL STORAGE ADDRESS switches. STOP/SYNC 2 triggers on the address stored in the Compare register. It should be noted that the STOP/SYNC address is not actually executed. Triggering takes place when the address is loaded into IAR.

When using both STOP/SYNC controls, load STOP/SYNC 2 before loading STOP/SYNC 1.

To setup STOP/SYNC 2:

1. Set FUNCTION switch to CE.
2. Set MODE SELECT switch to LOAD COMP.
3. Set CONTROL STORAGE ADDRESS switches to desired address.
4. Actuate EXEC switch. This loads the selected address into the Compare register.
5. Set STOP/SYNC 2 switch to desired position.

To setup STOP/SYNC 1:

1. Set FUNCTION switch to CE.
2. Set CONTROL STORAGE ADDRESS switches to desired address.
3. Set STOP/SYNC 1 switch to desired position.
4. Set OPERATION and MODE SELECT switches according to desired mode of running routine.
5. Actuate EXEC switch. Microprogram will start at address in CONTROL STORAGE ADDRESS switches and, depending on what is selected, either run to or recycle on address in Compare register.

CONTROL PANEL

All controls needed for operator control are located on the Control Panel (Figure 2-2).

EIN (Power On/Off)

When controller is in NORMAL mode this switch/indicator turns on ac/dc power within the controller. It also applies sequence power to the attached drives.

KANAL A/B GESP. (Channel Disconnect)

Each switch (KANAL A and KANAL B) independently controls I/O communications between the controller and the two CPU channels. Pressing (to light) the switch starts the disabling process. Disabling of a channel is accomplished by inhibiting the applicable transmitters.

STOP

Pressing (to light) the switch stops subsystem operation. Any sequences in operation come to an orderly halt. Controller then goes inoperable, except that a Sense can still be executed if there are any secondary conditions at the time of termination.

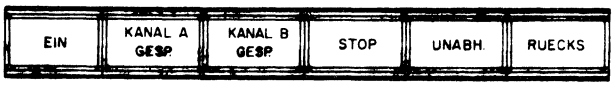
If STOP is used, wait at least 10 seconds before placing the subsystem in local or power off status. This gives the channel time to process all outstanding interrupts.

UNABH. (Local/Remote)

Pressing (to light) this switch places the subsystem in Local mode. The transmitters are inhibited and the RUECKS switch is enabled.

RUECKS (General Reset)

This is a momentary switch that commands a general reset within the controller. All device reservations are released and the subsystem enters the quiescent state. The switch is effective only when the UNABH switch is set to local control.



885

Figure 2-2. Control Panel

SECTION 3

THEORY OF OPERATION

PART 1

CIRCUIT THEORY

INTRODUCTION

Circuit theory explains the basics of the hardware, and how it relates to the microprogram. The emphasis of this section is on the hardware and how it operates. For information on the microprogram, refer to Part 2 of this section.

Circuit theory is divided into eight areas as follows:

- Controller Power – Explains the basic functioning of the controller power system. It provides information on the Power Control Unit, DC Power Supplies, Power On Sequencing, and Power Off.

- Memory – Explains the controller memory functions and capabilities. It provides information on Control Storage, the Instruction Address Register, Backup Address Register, Failing Address Register, Data Address Register, Memory Registers, the Check Bit Generator, and Syndrome Generator.

- Microprogram Load – Provides information on the loading of the microprogram from the cassette tape. It provides information on the Data Format, IMPL Function, and MPL Function.

- Arithmetic – Explains the arithmetic processing done by the controller. It provides information on the A Bus Assembler, A Register, B Bus Assembler, B Register, Arithmetic/Logical Unit, and the D Bus.

- CUDI – Explains the basics of the Control Unit Device Interface. It provides information on Signals To the Drive, Signals From the Drive, the ID Assembler, and the Orientation Counter.

- SERDES – Explains the Serialization and Deserialization of data. It covers the SERDES Clock, SERDES Write, and SERDES Read functions.

- Read/Write ECC – Explains the Read/Write Error Correction Code. It provides a Simplified Example, and information on ECC Write, ECC Read, and ECC Control.

- Machine Timing – Explains the controller timing. It provides information on the Machine Clock, Introduction Timing, and Machine Stops.

CONTROLLER POWER

INTRODUCTION

The controller receives three-phase power from the facility main power source. Input voltage may be either 208v (phase-to-phase), 60 Hz, or 220v (phase-to-neutral), 50 Hz. This power, under control of the Power Control Unit (PCU), supplies ac power to the drives and to the dc power supplies inside the controller. Seven dc power supplies provide ±5v and ±24v outputs.

Subsystem power is controlled solely by the controller without the need of sequence control by the channel or CPU.

POWER CONTROL UNIT

Drive Power

The PCU (A9) supplies the relay functions needed to control ac power distribution to the controller and to the drives. It also monitors the outputs of the dc power supplies to ensure that their outputs are within their required tolerances.

Each drive independently receives ac power from its power cable connected to PCU terminal board TB04 or TB05. Power is supplied to the terminal boards such that each 60 Hz drive receives two-phase power (ØA-ØB, ØA-ØC, or ØB-ØC), or each 50 Hz drive receives one of the three phases. This method of power connection rotates the phases so that each group of three drives presents a balanced load to the facility power. With circuit breaker CB01 (Figure 3-1) closed, ac power is continuously supplied to all drives. The drives may, therefore, be turned on for maintenance purposes without powering up the entire subsystem.

Controller Power

Closing circuit breaker CB02 (Figure 3-1) has the following effects:

1. Controller fans are energized. Two fans are located at the underside of the logic chassis to cool the logic modules.
2. A 3 vac clock signal is applied to the logic. During operation, this clock signal is used by the IMPL logic in its two-second forward delay function and also by CUDI to increment the block inline counter.
3. An internal +24v power supply is energized. Emergency Power Off (EPO) relay K1 picks to supply power-up sequence power to the subsystem.
4. Another internal power supply provides +5v and +4.75v reference signals for the PCU voltage monitor circuits.
5. If circuit breaker CB03 is also closed, single-phase power is supplied to the controller convenience outlet.

Relay Functions

The PCU contains 14 relays used for power control (Figure 3-2). Refer to Table 3-1 for a list of relay functions.

Voltage Monitor

The voltage monitor circuits (Figure 3-3) determine if the controller dc voltages are within their proper limits.

During the power on sequence, the outputs of the 5v power supplies are checked to make sure that they are at least coarsely operational. They must attain an output of at least 4v within three seconds after power initiation. If so, relay K5 picks to permit continuation of the power on sequence; if not, relay K7 picks before K5 to open K14: ac power to the dc power supplies is interrupted and the power on sequence stops.

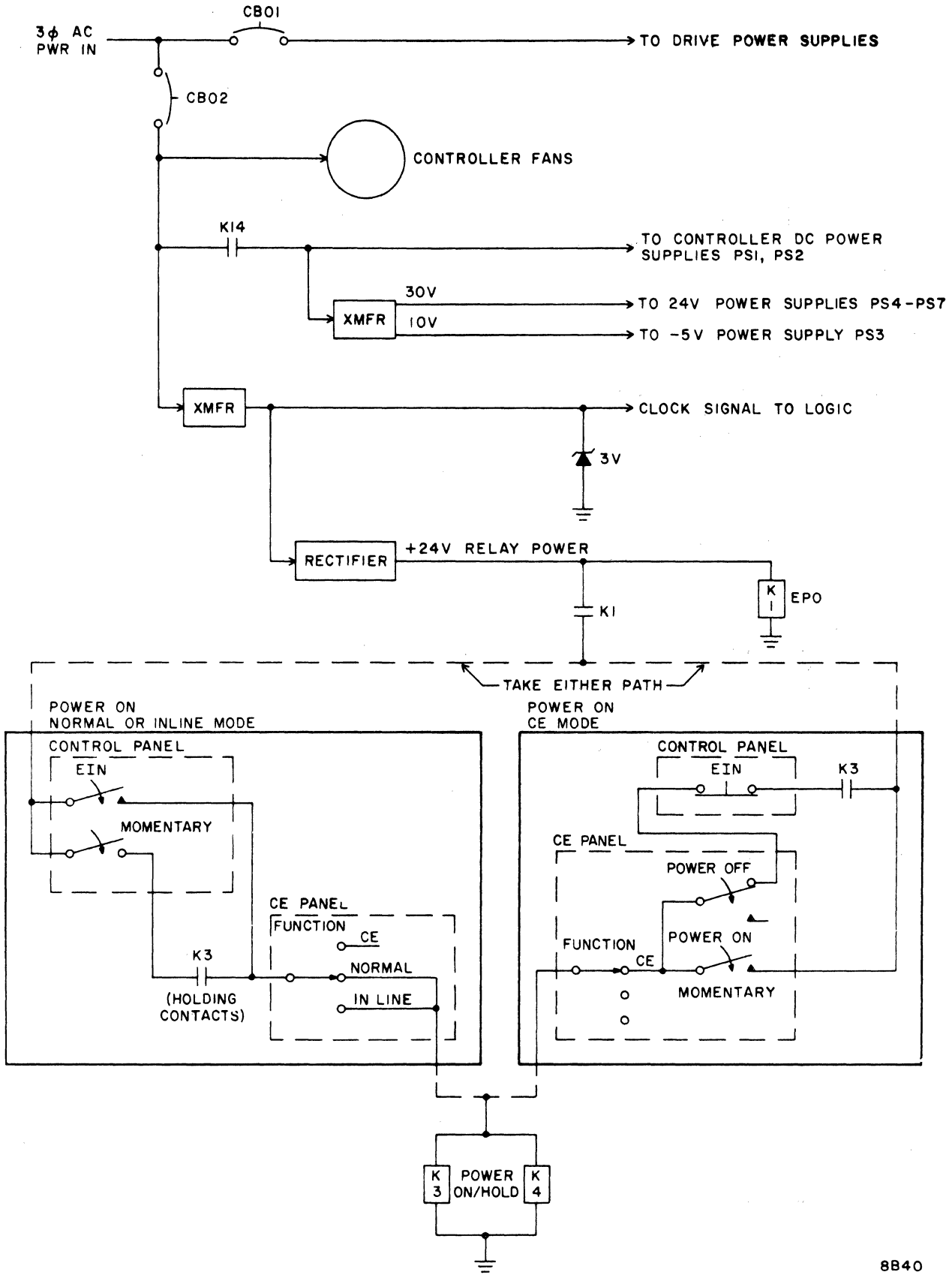


Figure 3-1. AC Power Distribution and Power On

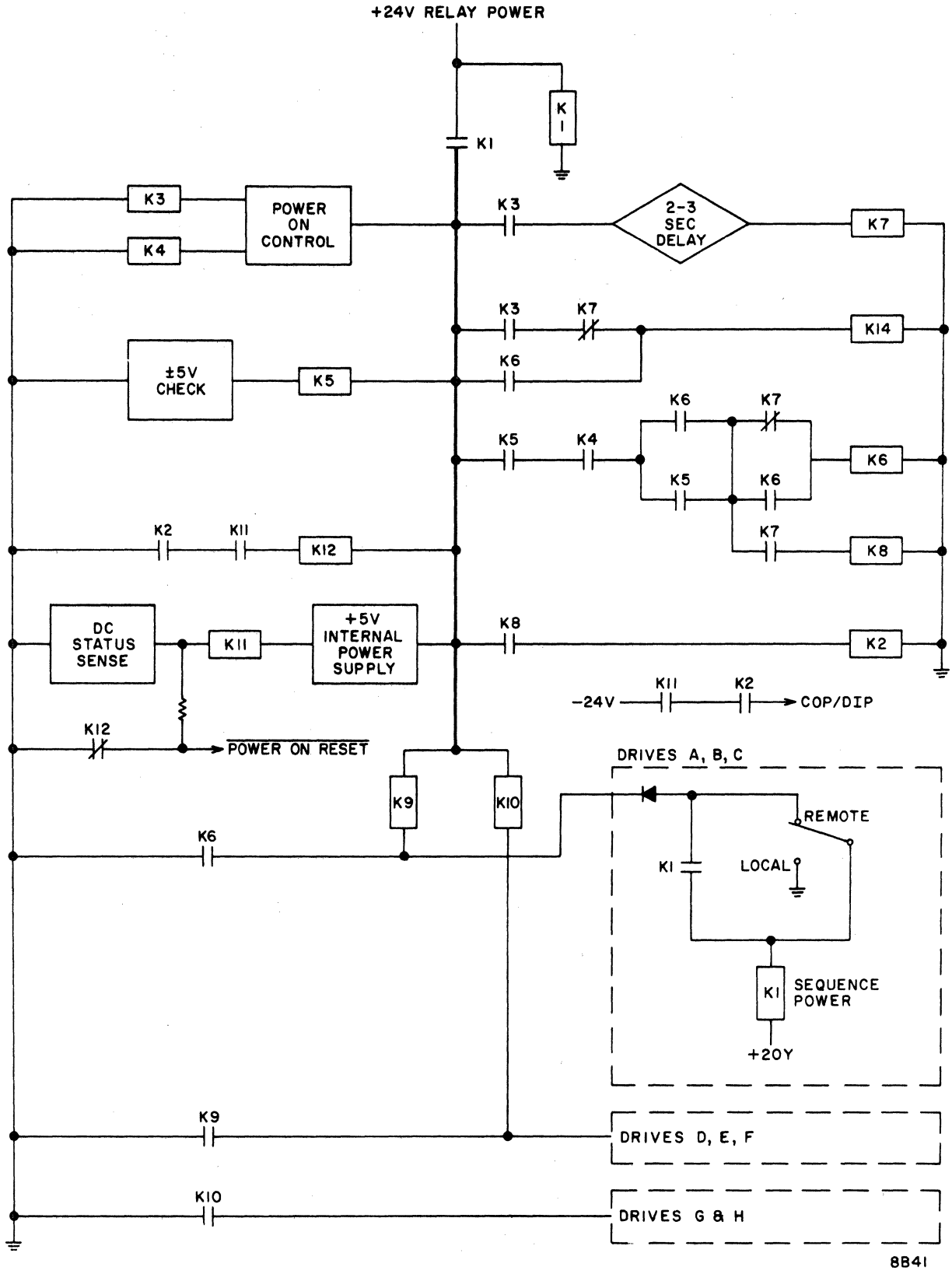


Figure 3-2. PCU Relays

TABLE 3-1. PCU RELAY FUNCTIONS

Relay	Name	Coil Enable	Function
NOTE Relays are listed in same order as energized during power up.			
K1	EPO	+24v Relay Power	Supplies +24v relay power to power switches and other relays.
K3, K4	Power On/Hold	Power On Switches	Energize when EIN switch (Normal or Inline modes) on control panel or POWER ON switch (CE mode) on maintenance panel is pressed. Relay K3 then provides holding contacts until power is turned off.
K14	DC Pwr Supply Source	K3 or K6	Provides ac power to controller dc power supplies.
K5	Pwr Supply Sense	Voltage Monitor	Energized when ±5v power supplies have attained at least 4v.
K6	Start ABC	K5 before K7	Applies ground to drives A, B, and C to enable their spindle motors (provided that drive interlocks are closed). Enables K9.
K7	3 Sec Timeout	K3 plus 3 seconds	Energized 3 seconds after K3 picks. Inhibits further power on if K5 is not already picked.
K8	Power On OK	K7	Energized if K5 picks before K7. Enables K2, ±5v sense, and ±24v sense.
K2	Power On	K8	Partially enables CPO/DIP.
K11	DC Voltages Normal	±5v, ±24v Monitors	Energized if dc power supplies are within normal operating tolerances. Provides COP to drives and DIP to channel. Enables K12.
K12	Power On Reset	K11	Removes Power On Reset (ground) applied to controller logic. Initial Microprogram Load (IMPL) automatically starts.
K9	Start DEF	K6	Ten seconds after K6 picks, applies spindle motor enable ground to drives D, E, and F.
K10	Start GH	K9	Ten seconds after K9 picks, applies spindle motor enable ground to drives G and H.
K13	Margin Enable	ERROR DISABLE Switch	Permits checking voltage margins by disabling ±5v and ±24v sense circuits.

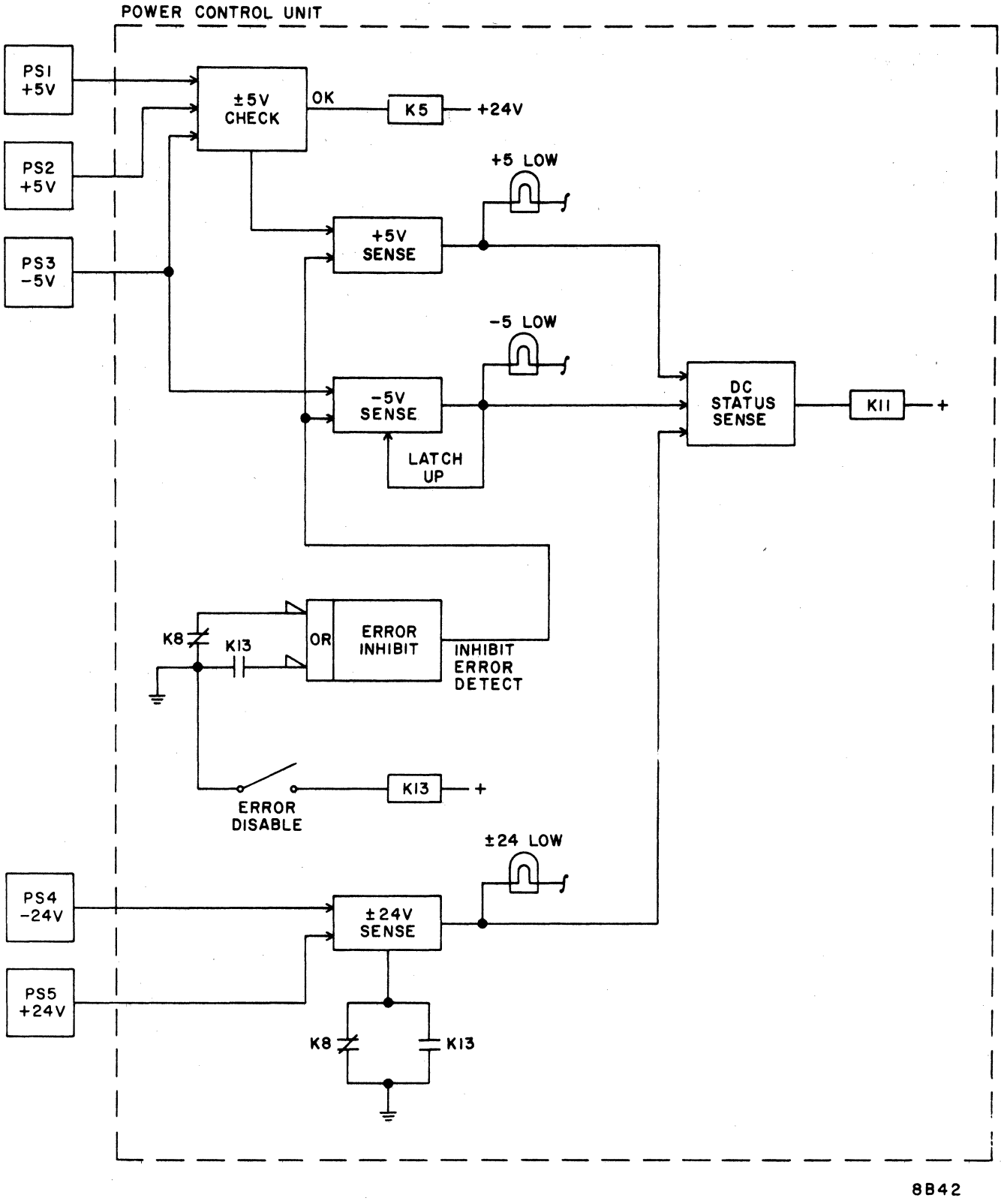


Figure 3-3. Voltage Monitor Circuits

After relay K8 picks, the outputs of power supplies PS1 through PS5 must be within their normal operational tolerances. Three sense circuits independently monitor +5v, -5v, and the ±24v power generated by PS4 and PS5. (PS4 and PS5 provide power to the transmitters and receivers. The lamp drive and cassette reader 24v power generated by PS6 and PS7 is not monitored.) If all monitored voltages are normal, K11 picks to permit continuation of the power-on sequence. If not:

- K12 remains de-energized to inhibit COP/DIP and to maintain a Power On Reset condition.
- The applicable error lamp on the PCU lights.

The voltages are continuously monitored during controller operations. If any of the voltages stray from their required tolerances (4.75v for the 5v supplies, or 16 to 22 volts for the 24v supplies), K11 drops to initiate another Power-On Reset. Loss of -5v latches the error to prevent returning to normal operation without a complete power down.

Voltage margin tests run during preventive maintenance require that the voltages be reduced below their normal values. Setting the ERROR DISABLE switch on the PCU to its ON position inhibits the voltage monitors.

DC POWER SUPPLIES

Seven dc power supplies (PS1 through PS7) supply all dc voltages used within the controller. They receive their ac inputs via relay K14 in the PCU. Power supply functions are listed in Table 3-2.

POWER ON SEQUENCE

The subsystem power on sequence is illustrated in Figure 3-4.

POWER OFF

Turning off power de-energizes relays K3 and K4. All power to the controller and its drives is immediately dropped nonsequentially. AC power remains available to the drives via circuit breaker CB01 to permit their operation for maintenance purposes provided that the applicable drive's LOCAL/REMOTE switch is set to LOCAL.

TABLE 3-2. DC POWER SUPPLY FUNCTIONS

Power Supply	Output Voltage	Output Use
PS1	+5v	A1 backpanel
PS2	+5v	A2 backpanel Drive for controller panel switches Cassette reader
PS3	-5v	A2AA through A2AH
PS4	-24v	Channel rcvr/xmtr (A2BL, A2BM) COP and DIP
PS5	+24v	Channel rcvr/xmtr (A2BL, A2BM)
PS6	-24v	Control panel lamps Cassette reader
PS7	+24v	Cassette reader

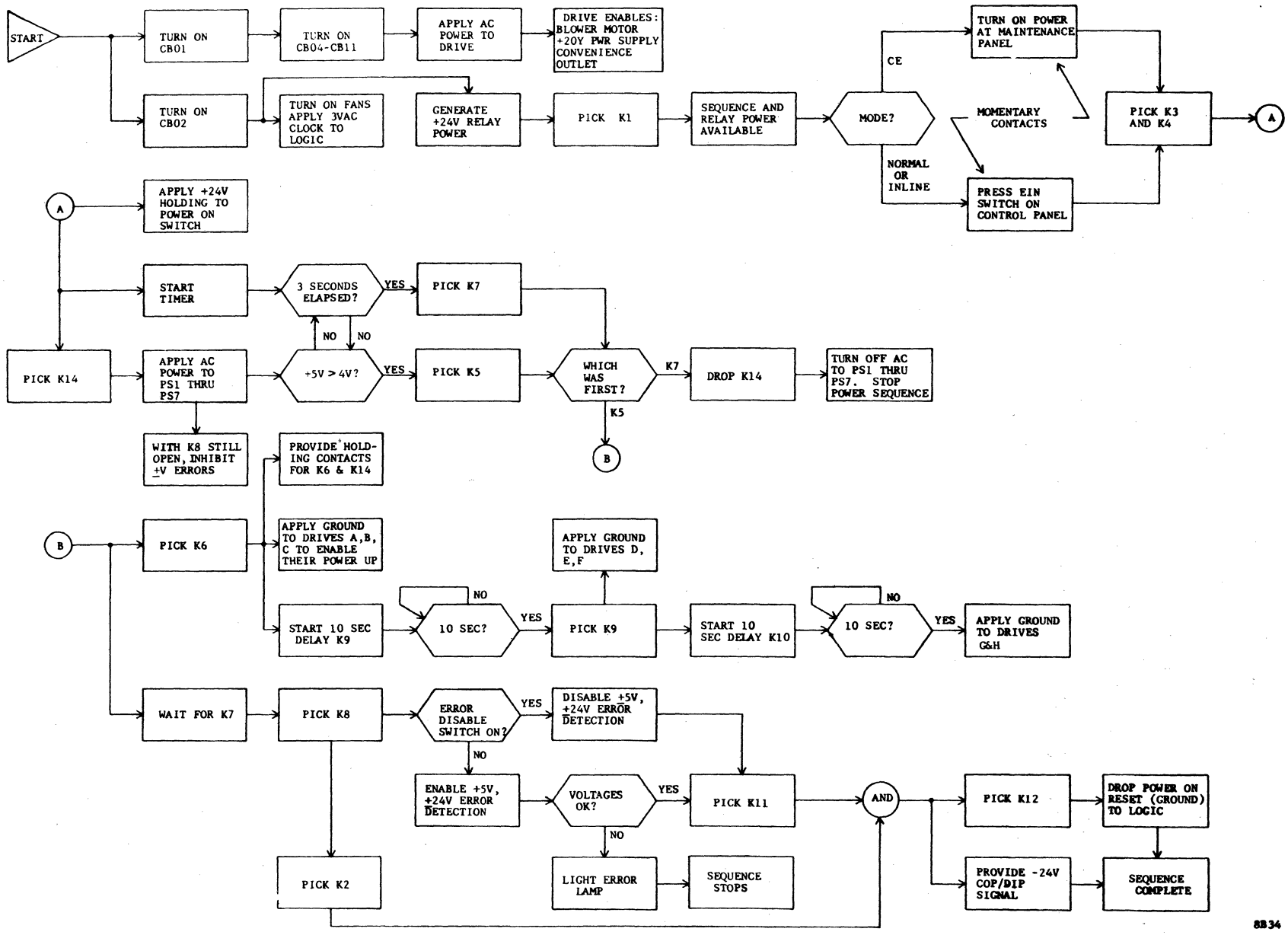


Figure 3-4. Subsystem Power-On Flow Chart

MEMORY

INTRODUCTION

The memory function (Figure 3-5) consists of control storage and the elements allied with it. Major elements of the memory function are:

1. Control storage, used to store the microprogram.
2. Instruction Address Register (IAR), used to address control storage during an instruction cycle.
3. Backup Address Register (BAR), used for maintenance purposes.
4. Failing Address Register (FAR), used for maintenance purposes.
5. Data Address Register (DAR), used to address control storage during a store or fetch cycle.
6. Memory registers MA, MB, MC, and MD, used to load data into and out of control storage.

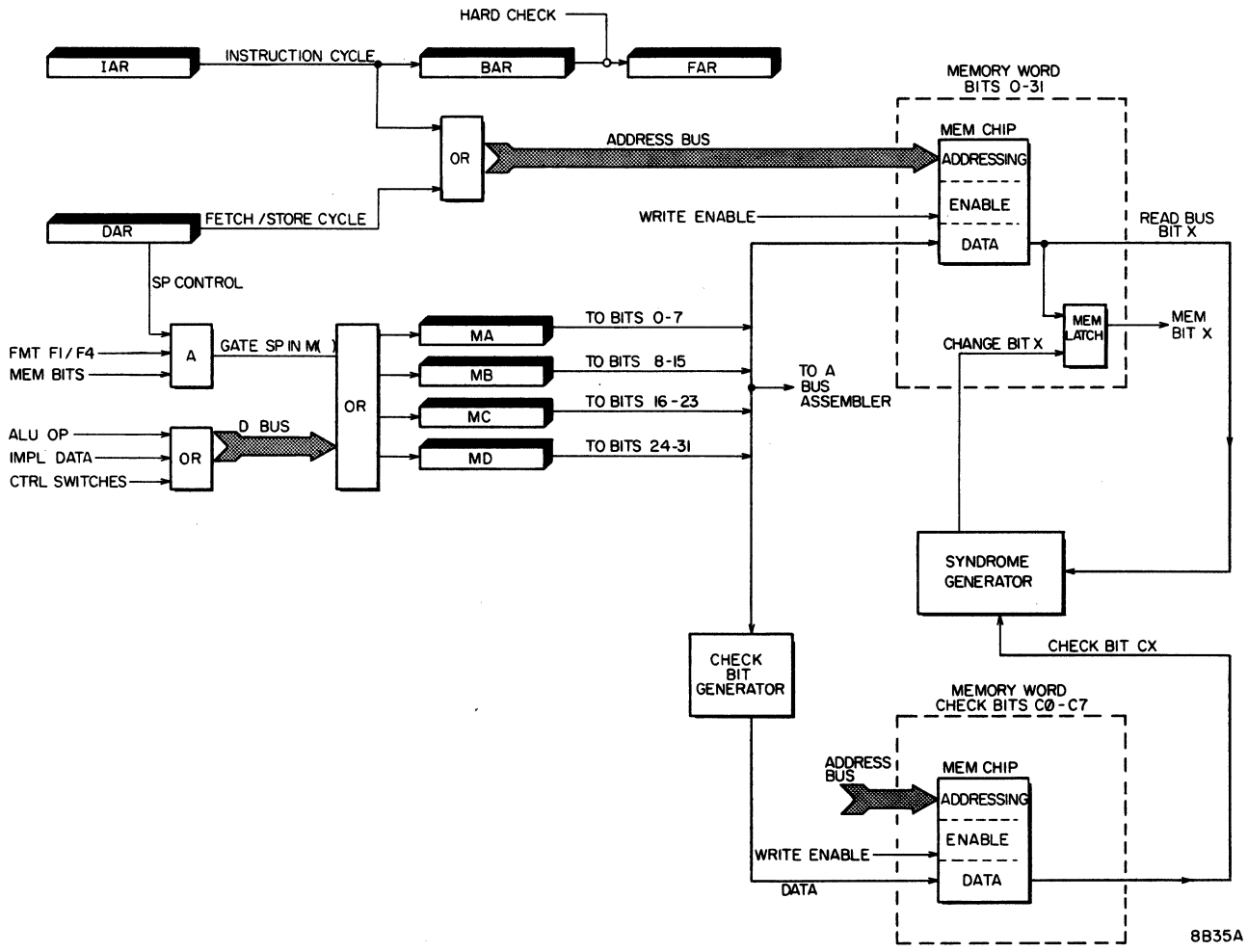


Figure 3-5. Memory Function

7. Check Bit Generator, used to generate a check byte as data is loaded into control storage.

8. Syndrome Generator, used to generate a syndrome byte as data is read from control storage. The syndrome byte is compared with the check byte to ensure readout accuracy.

CONTROL STORAGE

Function

The control storage stores the microinstructions used to control and test the subsystem. It is also used to store error usage log information and miscellaneous control data.

The memory is 4096 words (4K x 40) with each word consisting of 32 data bits and 8 bits of error correction code (ECC) used for error detection and correction.

The full 4K x 40 memory is packaged over 10 circuit boards. The 10 boards have monolithic integrated circuits. Each of the boards contains 64 memory chips. In turn, each chip stores one bit for 256 of the 4096 addresses. Therefore, each board contains 4 bits of the 32-bit instruction word or 4 bits of the 8-bit check word. The 32 bits, 0 through 31, comprising each instruction word are located on the boards at A1BB through A1BJ. The eight bits, C0 through C7, comprising the check byte for each word are located on the boards at A1BK and A1BL. As any word is read into or out of memory, all 10 boards are active in constructing the full 40-bit microword.

In addition to the memory chips, each board also contains the memory latches; one for each instruction bit. However, only the memory latches on the eight instruction word boards are used. The output of these latches is the 32-bit instruction word which is decoded by machine logic to control subsystem operations. These latches can be corrected by the syndrome generator to correct any one-bit error in the instruction word. The check byte boards physically contain the latches, but they are not used.

The memory chips have three basic inputs: Address Bus, Write Enable, and Data In.

Address Bus

The Address Bus selects any one address for either a memory read or a memory write operation. The bus source is the IAR for instruction cycles and is DAR for fetch/store cycles. Although IAR and DAR are 16-bit registers, only their bits 2 through 13 are used for the address bus. This means that the highest accessible address is hex 3FFC. Also, since bits 14 and 15 of the address are always zero, the least significant address byte must be 0, 4, 8, or C.

Except when data is being loaded into control storage by a memory write operation, the memory chips continuously supply the contents of the selected address to the read bus. Read bus data is the complement of the stored data.

Write Enable

Write Enable is required to store data into control storage. DAR is the address bus source. Write Enable is functional from Clock D to Clock G time under any one of the following conditions:

1. An S4 instruction.
2. The IMPL function counter indicates that a 4-byte word has been loaded into memory registers MA through MD for storage.
3. A CE store function (STR MEM) has been selected on the maintenance panel.

Data In

Data is loaded during a Write Enable. Signal sources are:

Bits	Source
0-7	MA Reg
8-15	MB Reg
16-23	MC Reg
24-31	MD Reg
C0-C7	Check Bit Generator

During a subsequent read cycle, data placed on the read bus is the complement of the stored data.

INSTRUCTION ADDRESS REGISTER

Function

The Instruction Address Register (IAR), Figure 3-6, is a 16-bit register used to specify the address of the control storage word to be accessed. Bits 0 and

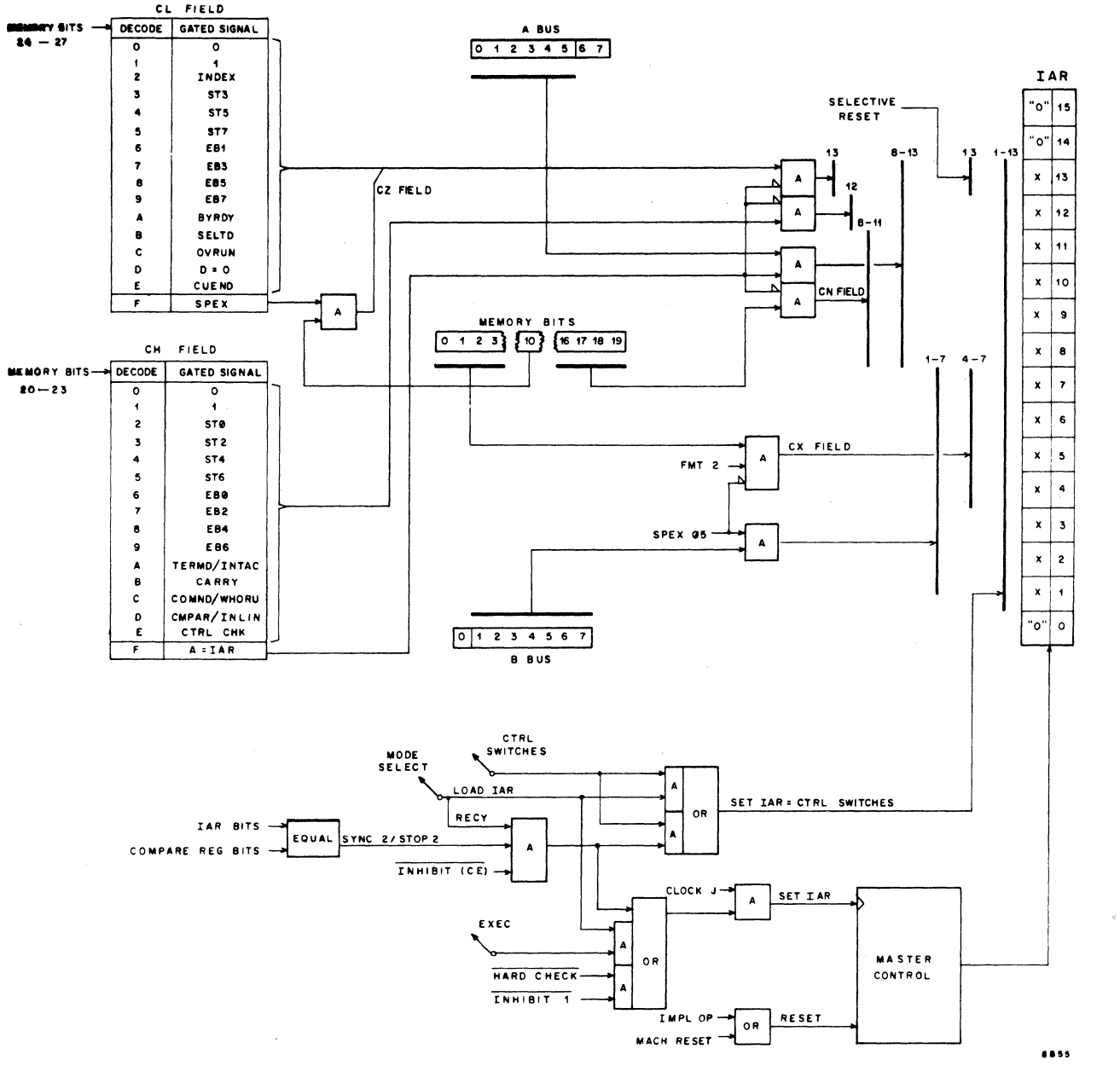


Figure 3-6. Instruction Address Register Inputs

1 are always "0" since there are only 4096 words; bits 14 and 15 are always "0" since each instruction is on a four-byte boundary.

In formats 1 through 5, only IAR is gated into memory. In formats F1, F4, and S4, IAR is gated into memory for an instruction cycle followed by DAR for a data cycle.

IAR Loading

Data loaded into IAR varies with the format selected by the microword. A list of bits gated into IAR is provided by Table 3-3.

TABLE 3-3. IAR INPUTS

Formats	Conditions	IAR Bits											
		2	3	4	5	6	7	8	9	10	11	12	13
--	IMPL	Reset to Zero											
--	Selective Reset	0	0	0	0	0	0	0	0	0	0	0	1
--	CE Load or Recycle	Control Storage Address Switches											
1, 3, 4, 5 F1, F4, S4	$\overline{A=IAR}$	NC						CN				CH	CL
1, 3, 4, 5 F1, F4, S4	A=IAR	NC						A Bus 0-5					
2	$\overline{A=IAR}$	NC			CX			CN				CH	CL
2	A=IAR	NC			CX			A Bus 0-5					
1, 3, 4	$\frac{SPEX\cdot}{\overline{SPEX\ 05}}$	NC						CN				CH	CZ
2	$\frac{SPEX\cdot}{\overline{SPEX\ 05}}$	NC			CX			CN				CH	CZ
1, 3, 4	SPEX 05	B Bus 2-7						CN				CH	CZ

Abbreviations:
NC=no change
CN=memory bits 16-19
CH=CH field branch test
CL=CL field branch test
CX=memory bits 0-3
CZ=memory bit 10

NOTE: SPEX and A=IAR are mutually exclusive except for SPEX 05 + A=IAR.

Branching

The CH and CL fields permit conditional tests of external conditions. These fields control IAR bits 12 and 13, as permitted by certain formats, to establish a potential four-way branching scheme. Each field selects the status of one of fifteen conditions; the selected status bit is transferred to the associated bit of IAR. The CH field is bits 24 through 27 of the memory word. The CL field is bits 20 through 23.

Branching conditions are shown in Figure 3-6. One of these conditions is true (setting the associated IAR bit to "1") if the field decode calls for a condition test and if the test result is true. If the test result is false, the IAR bit is turned off. The meanings of these tests are as follows:

1. The ST branches are self-explanatory. These check previously stored ST register states to allow status to control the sequence of microword execution. The ST bits are normally turned on or off by a previous CS field command. Two exceptions are ST3 and ST4. Not only can ST3 be turned on or off by the CS field control, but it is also controlled by an ALU operation specifying a carry out store (for example, A+B=DC). ST4 is turned on only by SERDES when a byte has been deserialized (while reading) or serialized (while writing); it is turned off by a CS decode of 0=ST4.

2. The EB branches examine their corresponding bits of the EB register. This function is used primarily for command decode.

3. A CH/CL decode of 0 or 1 forces the corresponding IAR bit to 0 or 1.

4. INDEX indicates that the selected drive generated an Index pulse following a CS decode of 1=ST1.

5. CARRY and D=0 are related to ALU operations. They examine the results of the last instruction executed by ALU. There can be both a CARRY and a D=0 branch in the same microblock. For example:

   A Reg = 1100 1011
   B Reg = 0011 0101
   D Bus = 0000 0000 with a carry out

6. The remaining branch tests are related to channel or drive interface operations.

Error Detection

IAR is checked against its fields to determine if there is an error; if so, IAR Error is generated. In turn, this generates a Hard Check. The following checks are made:

1. If A=IAR is used, A register bits 0-5 are compared with IAR bits 8-13.

2. If A=IAR is not used, IAR bits 8-11 are compared with the CN field (memory bits 16-19).

3. In format 2, IAR bits 4-7 are compared with the CX field (memory bits 0-3).

4. In SPEX 05 operation, IAR bits 2-7 are compared with B Bus bits 2-7.

BACKUP ADDRESS REGISTER

The Backup Address Register (BAR) is used as a maintenance aid to assist in microprogram troubleshooting. BAR saves the previous instruction address from IAR. Therefore, IAR contains the address of the next word to be executed while BAR contains the address of the word being executed. BAR is not used by the microprogram; it is for display only.

FAILING ADDRESS REGISTER

The Failing Address Register (FAR) preserves the address being executed whenever a Hard Check

occurs. Contents of BAR are loaded into FAR at the leading edge of the Hard Check. This address remains valid until another Hard Check occurs or until the unit is powered down.

FAR can be manually displayed by displaying the ID Assembler as follows:

1. Load hex 00 into IB register.
2. Display ID. FAR bits 2 through 7 are displayed on REGISTER DISPLAY (bits 0 and 1 are off).
3. Load hex 01 into IB register.
4. Display ID. FAR bits 8 through 13 are displayed on REGISTER DISPLAY (bits 0 through 5. Display bits 6 and 7 are off.

DATA ADDRESS REGISTER

Function

The Data Address Register (DAR) is a 16-bit register used to address a data word being fetched or stored. However, since there are only 4096 words in memory, bits 0 and 1 are not used. DAR is the input to the address bus during a fetch (F1/F4) or store (S4) instruction. It is also used to address memory during an IMPL or MPL operation, or during a CE fetch/store (STR, MEM, FTCH MEM). In Formats F1, F4, and S4, IAR is gated into memory for an instruction cycle followed by DAR for a data cycle.

DAR bits 14 and 15 have a special function in F1 and F4 formats. Refer to SP Gating in Memory Registers discussion.

DAR Loading

Data loaded into DAR varies with the format selected by the microword. Table 3-4 lists bits gated into DAR while Figure 3-7 illustrates DAR gating.

TABLE 3-4. DAR INPUTS

Formats	Conditions	DAR Bits														
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
--	IMPL	0	0	0	0	0	IMPL Address bits 0-7								0	0
--	Start MPL	0	0	0			NC									
--	CE Load	Control Storage Address Switches														
F1	--	NC			CW				CV				CY			
F4, S4	--	NC			CW		A Bus 0-7									
1, 2, 3, 4	SPEX 04	B Bus 5-7					NC									

Abbreviations: NC=no change
CV=memory bits 4-7
CW=memory bits 8-11
CY=memory bits 12-15

NOTE: SPEX is invalid in formats F1/F4/S4 and 5.

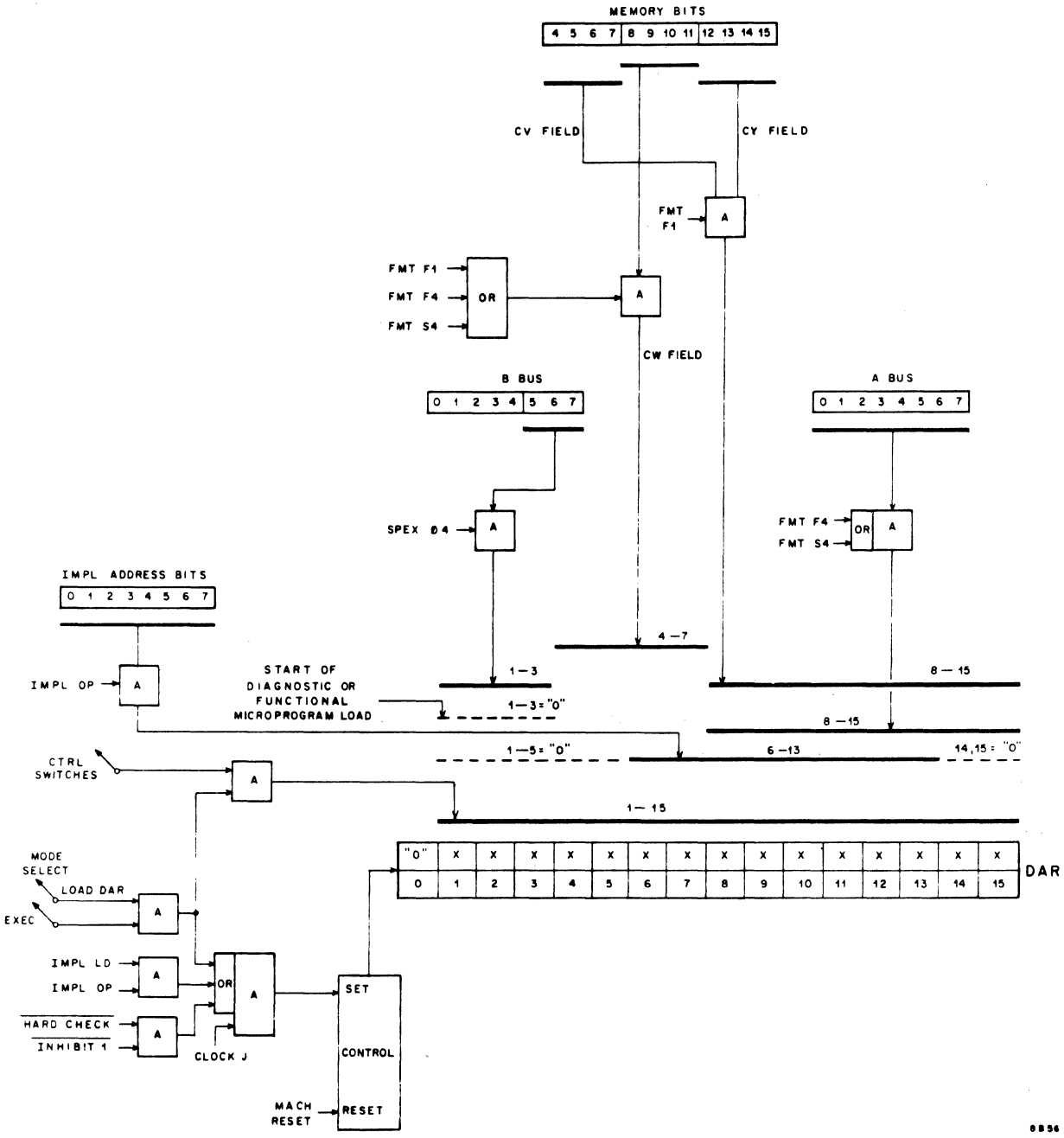


Figure 3-7. Data Address Register Inputs

Error Detections

DAR is checked against its fields to determine if there is an error; if so, DAR Error is generated. In turn, this generates a Hard Check. The following checks are made:

1. In formats F1, F4, and S4, DAR bits 4-7 are compared with the CW field (memory bits 8-11).
2. In format F1, DAR bits 8-11 are compared with CV (memory bits 4-7). DAR bits 12-15 are compared with the CY (memory bits 12-15) field.
3. In formats F4 and S4, DAR bits 8-13 are compared with A register bits 0-5.
4. In SPEX 04 operations, DAR bits 2 and 3 are compared with B register bits 6 and 7.

MEMORY REGISTERS

Function

The Memory registers are labeled MA, MB, MC, and MD. They store data to be transferred into, or out of, control storage. These registers have five possible inputs:

1. During normal ALU operations, they can be loaded by a normal CD decode of MA, MB, MC, or MD.
2. During IMPL operations, they receive IMPL read data. An IMPL counter controls the destination register (MA through MD) for each byte read from the cassette tape.
3. During MPL operations, they receive IMPL read data. However, the destination register for each byte is controlled by the microprogram.
4. In fetch (F1 or F4) operations, they receive specified bytes from memory. Refer to SP Gating.
5. In CE mode, they may be loaded from the maintenance panel.

The outputs of these registers are applied to the A bus assembler, control storage, and to the check bit generator.

SP Gating

SP Gating is used during fetch (F1 and F4) operations. D bus inputs to MA through MD are blocked and specific memory bytes are gated into the memory registers. As Figure 3-8 shows, DAR bits 14 and 15 control this gating.

In format F1, only one byte is fetched from the memory location specified by DAR bits 14 and 15. This byte is then loaded into the MA register. Memory registers MB, MC, and MD remain unaltered.

In format F4, four bytes are fetched from the memory location specified by DAR bits 14 and 15. Byte 0 or 1 or 2 or 3 is loaded into MA. Regardless of the DAR status, bytes 1, 2, and 3 are loaded into MB, MC, and MD, respectively. One special variation occurs during a SPEX 25 word. If a SPEX 25 precedes an F4 fetch, only MA is loaded. The contents of MB, MC, and MD remain unchanged.

Store 4 Instruction

During a format S4 instruction, the memory bytes are loaded into the control storage address specified by DAR bits 1 through 15. Byte assignments are:

Byte	Reg	Memory Bits
0	MA	0-7
1	MB	8-15
2	MC	16-23
3	MD	24-31

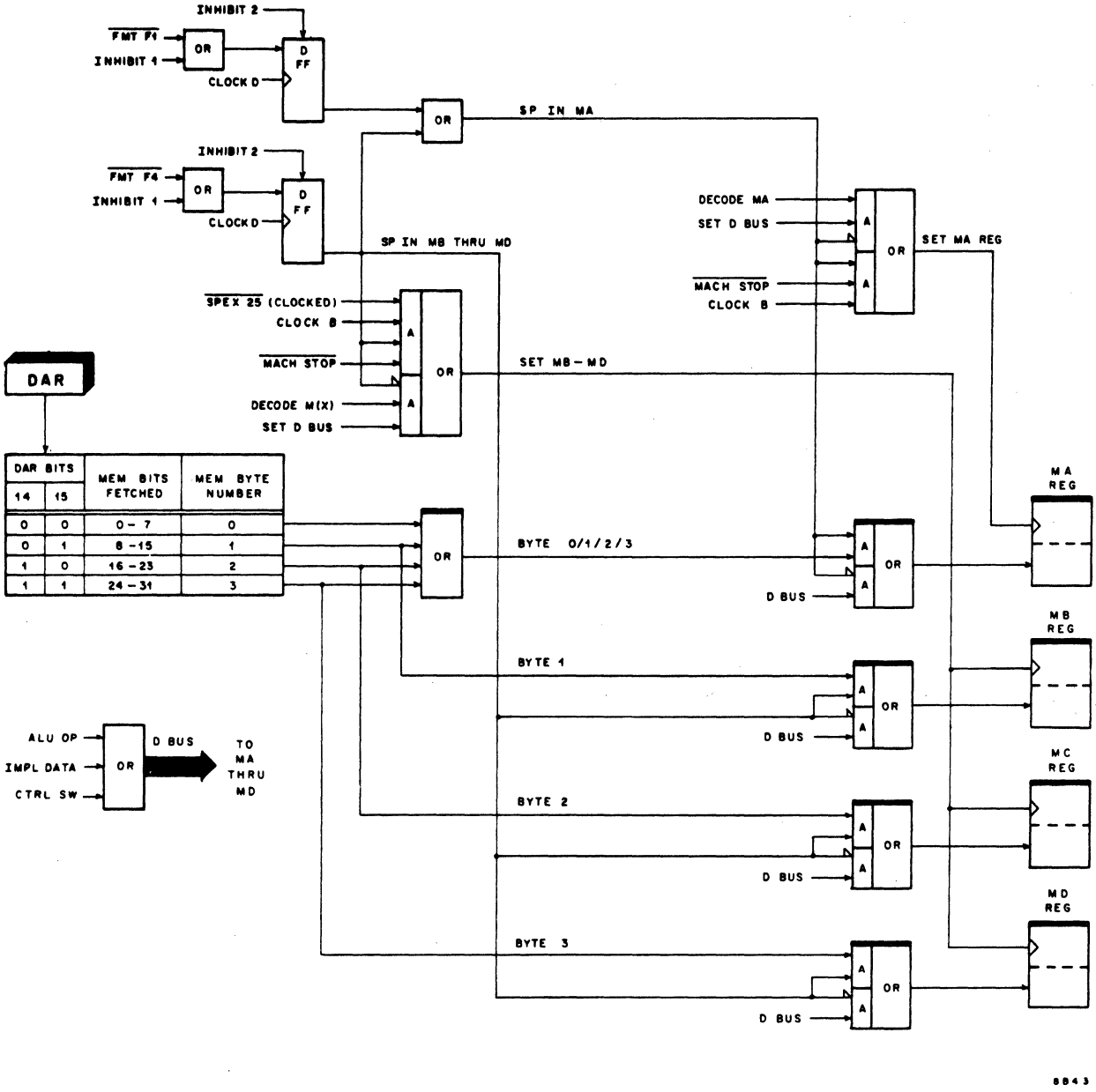


Figure 3-8. SP Gating into Memory Registers

Error Detection

During a write cycle, parity is checked on all four memory bytes. Memory byte 0 is the contents of the MA register plus its parity bit; byte 1 is MB; byte 2 is MC; byte 3 is MD. Any parity error will generate a Hard Check.

CHECK BIT GENERATOR

Function

The Check Bit Generator generates the check byte (bits C0 through C7) forming the last eight bits of each microword. The check byte is hardware-generated and loaded into control storage concurrent with the loading of the 32-bit instruction portion of each word. Figure 3-9 shows the hardware used to generate check byte bit C0. Check byte generation is always enabled following power up or any other machine reset condition. SPEX 17 is used to block writing check bytes for diagnostic purposes. After completion of the test, SPEX 03 returns the check bit generator to normal operation.

Each of the 32 instruction bits is examined by hardware controlling three bits of the check byte. A unique combination of these three bits exists for each memory bit. This combination is shown in Figure 3-10. Note that each bit of the check byte examines 12 memory bits. Each bit of the check byte is generated by forming odd parity with the 12 memory bits.

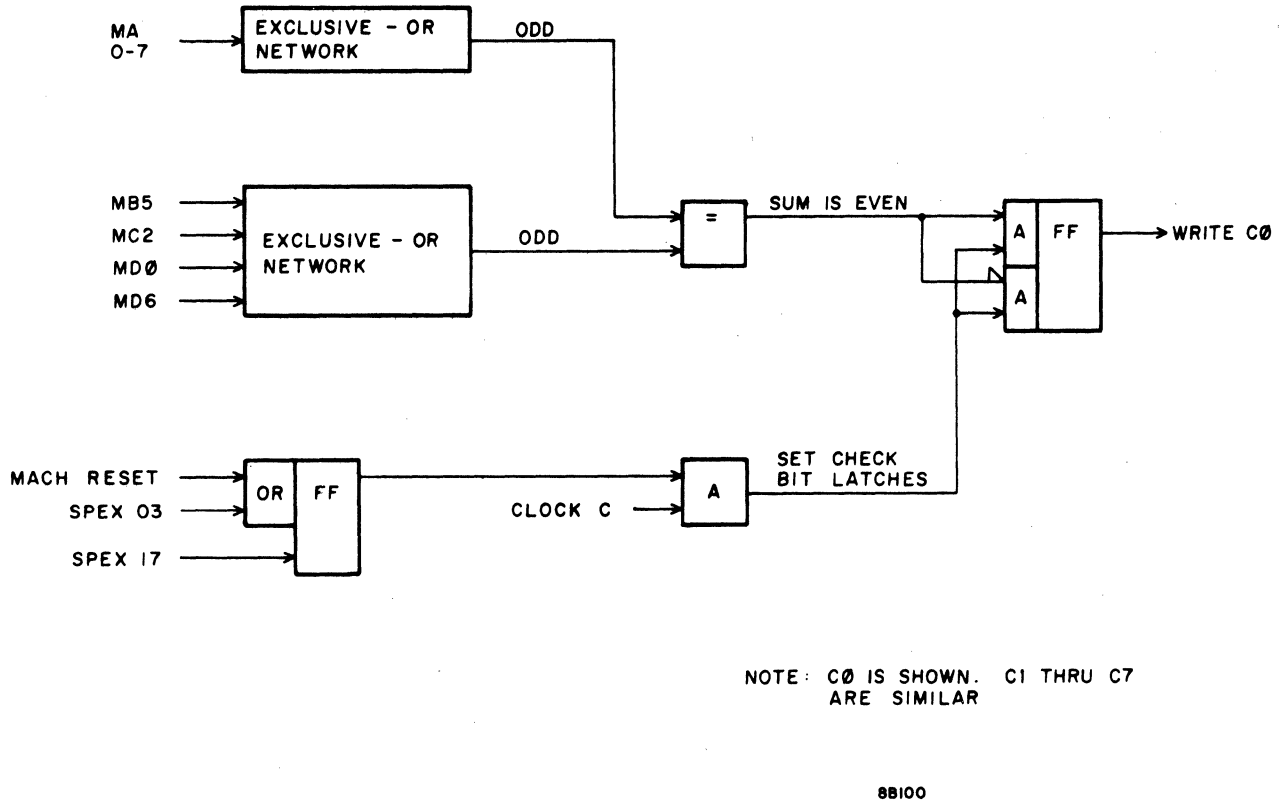


Figure 3-9. Check Bit Generator Logic

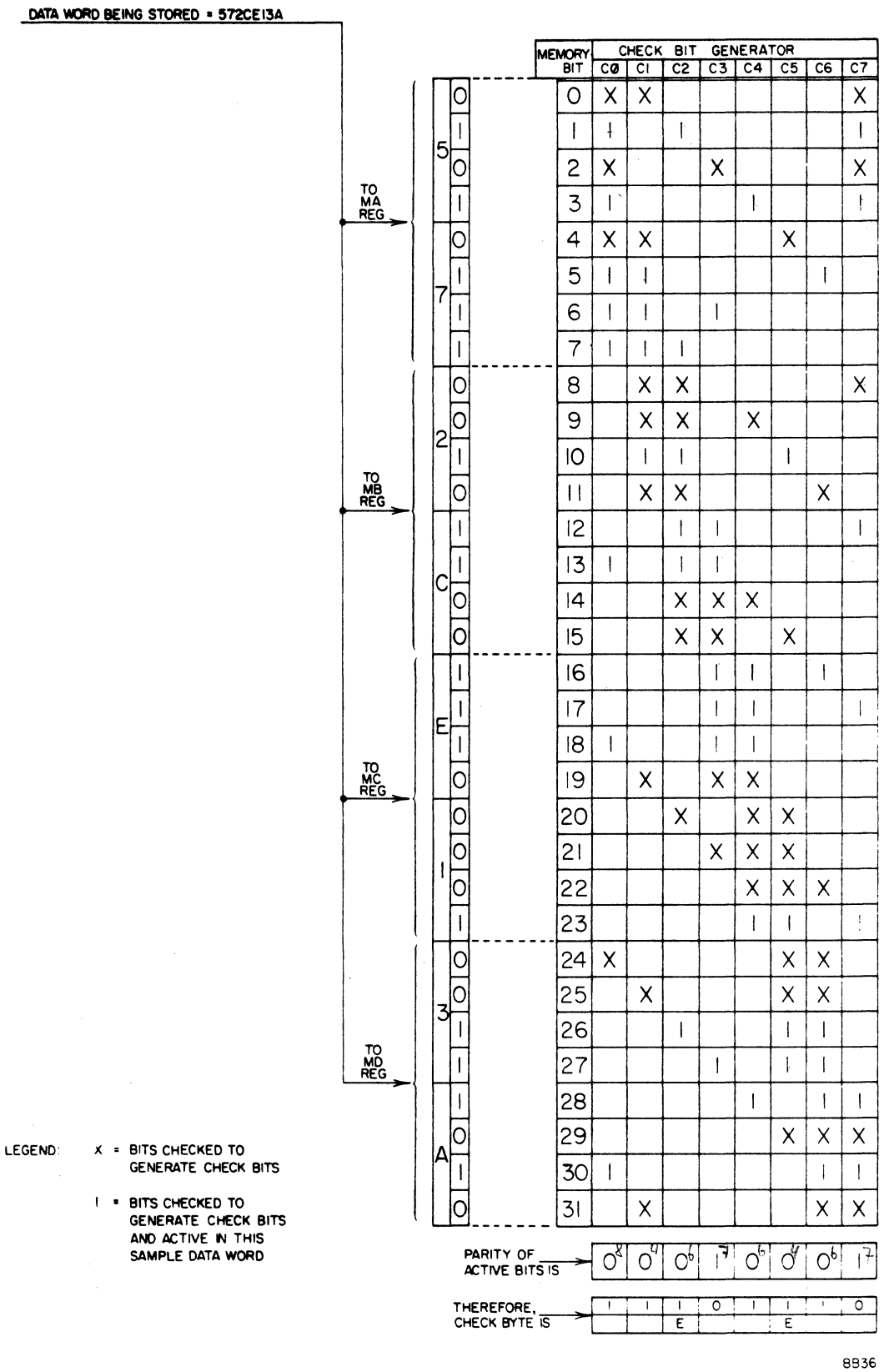


Figure 3-10. Check Bit Generation Example

Generation Example

Assume the bytes of the word 572CE13A are being written into control storage (Figure 3-10). Bit 0 of MA will become bit 0 of memory: this bit is checked by logic controlling C0, C1, and C7. No other memory bit uses this unique combination. Bit 1 of MA will become bit 1 of memory: this bit is checked by logic controlling C0, C2, and C7. This combination is also unique. All of the MA through MD inputs are checked in a similar fashion. Note that bit C0 examines memory bits 0 through 7, 13, 18, 24, and 30. Thus, 12 memory bits are checked to form C0. Since eight of the bits checked by C0 are on, parity is even. Bit C0 is turned on to make odd parity.

The remaining check bits are generated in an identical manner, although the memory bits examined differ uniquely in each case. The final word stored in memory is 572CE13AEE, where EE is the check byte.

SYNDROME GENERATOR

Function

The Syndrome Generator examines the 32-bit instruction portion of the microword and compares it with the 8-bit check byte during control storage read operations. A bit pattern identical to that examined by the check bit generator is used. This comparison is made as data is read out of control storage, but prior to memory word decoding. A memory correction byte, syndromes S0 through S7, is generated. If all of the syndromes are zero, there is no error. With three of the eight syndrome bits set to one, a correctable (one memory bit) error exists. If a single syndrome bit is set to one, the check byte is in error but the information data is correct. Any other combination of ones is detectable but incorrectable.

Correction Example

Assume that the same microword (572CE13AEE) explained in the Check Bit Generator discussion is being read out of memory. Figure 3-11 shows the bit pattern examined by the syndrome generator. For purposes of this explanation, memory bit 3 is a "0" rather than the recorded "1".

Figure 3-12 illustrates the logic that finds the error during this example. A non-error condition exists if the sum of the read bus bits being checked by any syndrome along with the complement of the associated check bit is even. Consider the read bus bits and check bit C0 examined by the syndrome zero (S0) circuit. Read bus bits 13, 18, 24, and 30 are compared with the complement of check bit C0: the result is odd. Then this result is compared with the sum of read bus bits 0 through 7. But, since bit 3 is off rather than on, that sum is even. The total result is odd (even + odd = odd), so an error exists – syndome bit S0 is on.

With bit 3 in error, S4 and S7 are also on. The unique combination of S0, S4, and S7 generates a Change Bit 3 signal. Referring back to Figure 3-10, note that this is the same combination used to examine bit 3 when the check byte was generated. The Change Bit 3 signal enables the memory latch to be turned on to form a corrected Memory Bit 3 output from the memory card. The bit 3 read bus signal is not changed; it remains in error for this microword. If bit 3 failed because of a hardware failure, the bit continues to be corrected with every word until another bit fails. (Remember, failures of two or more bits are uncorrectable.)

All of the other memory latches are set or cleared in accordance with their original read bus inputs.

ECC Error Determination

Two types of errors are detected by circuitry monitoring the syndrome generator: Double Error and ECC Error. Either error causes a Hard Check and lights the applicable ERROR DISPLAY indicator on the maintenance panel. In addition, this status is available to bits 0 and 1 of the SG register. The syndrome pattern can be displayed by setting the REGISTER SELECT switch to SYN.

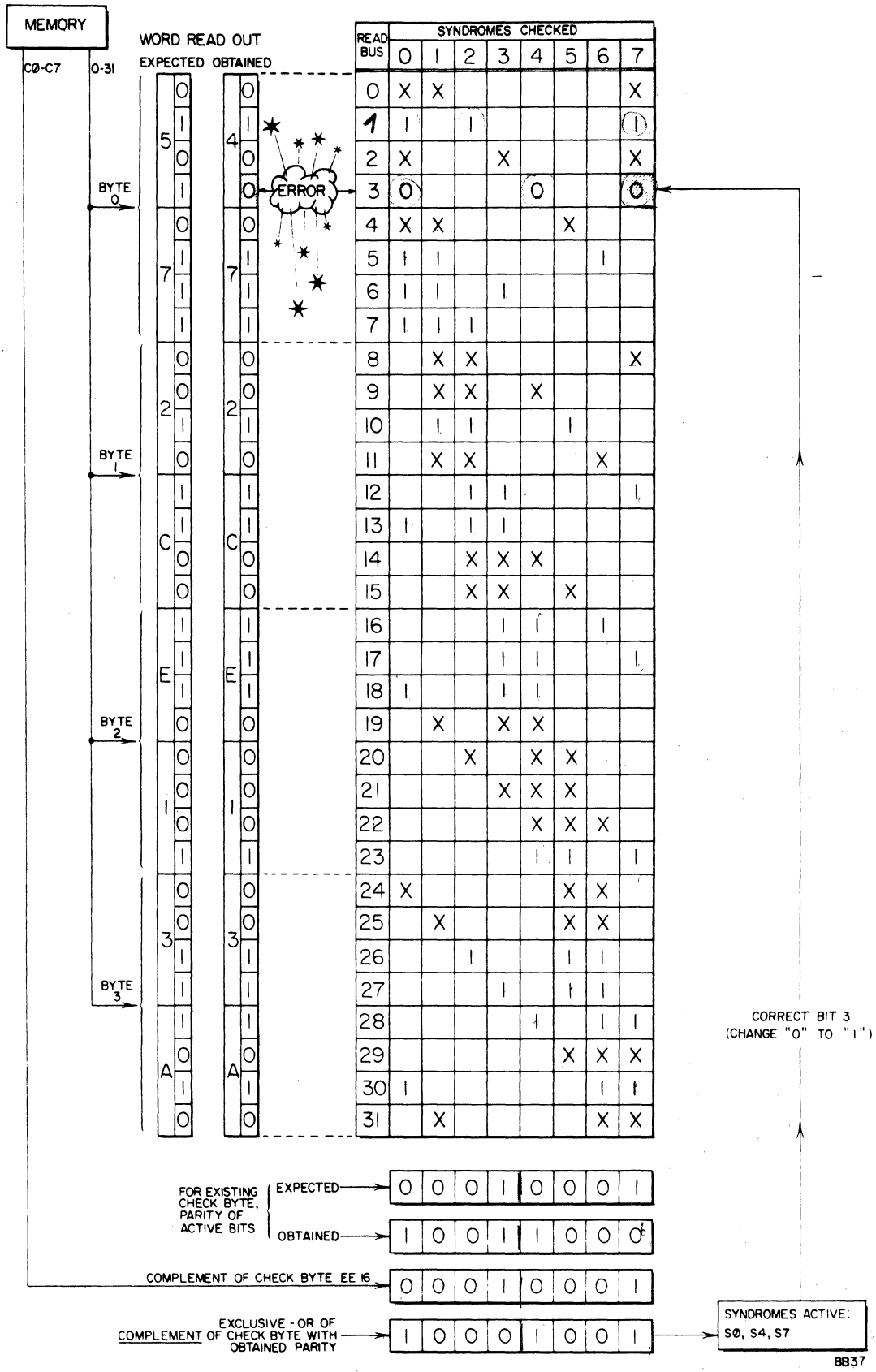


Figure 3-11. Memory Bit Correction Example

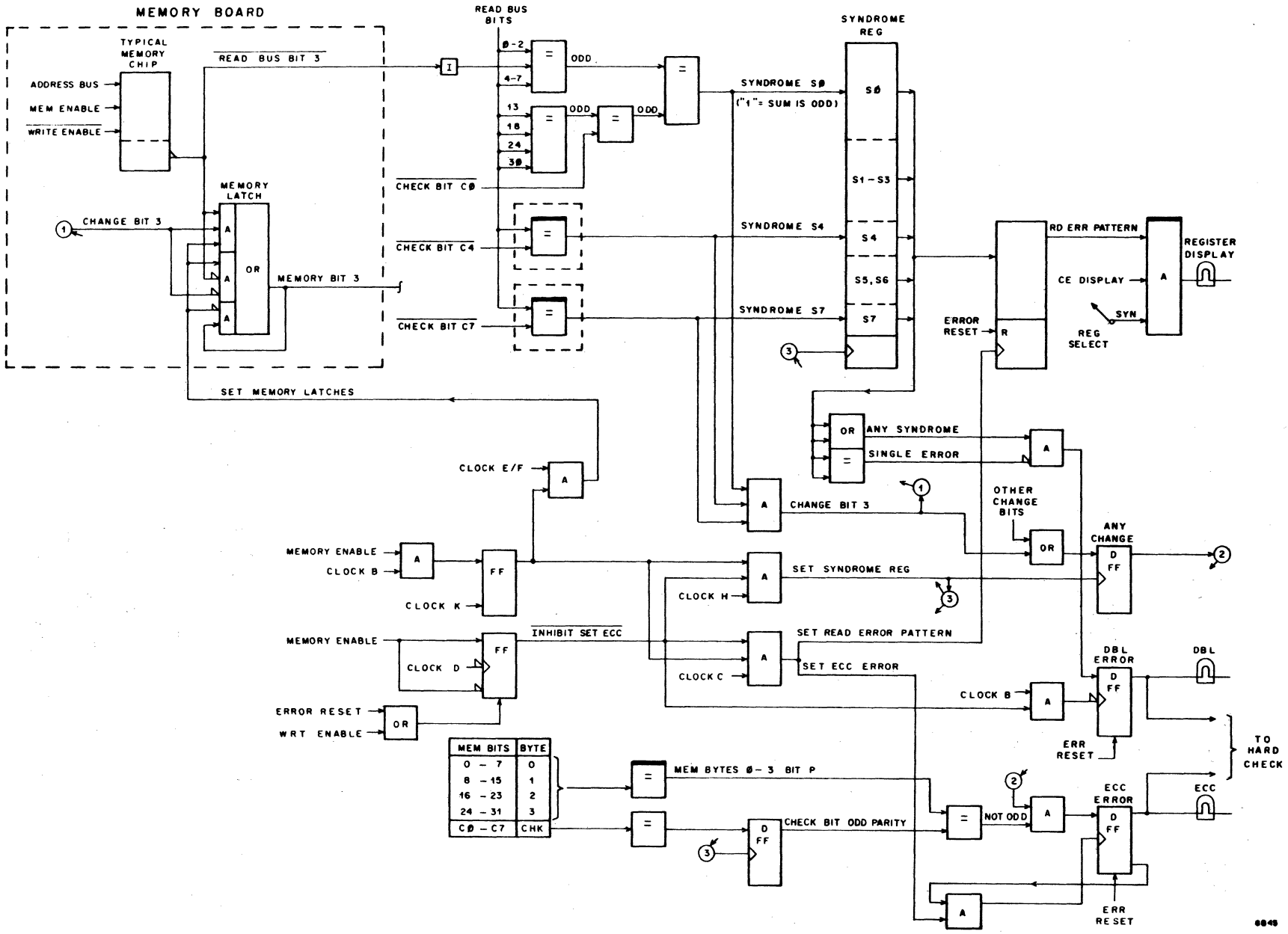


Figure 3-12. Syndrome Generator

Double Error (Figure 3-12) indicates that more than one bit of the 32-bit instruction read bus has an error. This is an uncorrectable error. Referring back to Figure 3-11, assume that both bits 3 and 4 are in error: bit 4 is a "1" rather than a "0". Syndrome S0 is now off because the bit 3 and 4 errors cancel each other. Syndromes S1 and S5 are turned on, however. As a result, syndromes S1, S4, S5, and S7 are on. Each syndrome correction byte can have only three bits on to provide a correctable error.

ECC Error is checked after the memory latches have been set. If the sum of memory bits and check bits is not odd, the correction was in error. Memory data is, therefore, unreliable.

MICROPROGRAM LOAD

INTRODUCTION

The microprogram load function provides a means of loading the microprogram into control storage. The functional components are the cassette deck assembly, a tape cassette, and one logic card at location A2BB.

Two independent functions are provided: initial microprogram load (IMPL) and microprogram load (MPL).

IMPL is required since the control storage is volatile, that is, memory is lost when power is removed. Immediately after a subsystem power on, or if commanded by enabling switches on the maintenance panel, the controller enters IMPL mode. The cassette rewinds to the beginning-of-tape (BOT), then moves forward while reading the tape. Under hardware control, the bootstrap loader and an initial hardcore diagnostic microprogram are loaded into memory. Motion stops after this function has been completed. The initial hardware diagnostics in this first record check the IAR and IAR gating of CX, CN, and CB fields; ALU test for all ALU operations; and the registers needed by the bootstrap to read in the rest of the microcode. Further reading to complete the loading of the remaining diagnostics and the functional microprogram is under microprogram control.

MPL is used to load the next four diagnostic records. These records contain diagnostics that check the remaining untested registers, branching, and CS decode. If the optional tests have not been requested, the controller overlays the hardcores with the functional microcode and begins execution.

MPL is also used to load any of the inline diagnostics. The desired diagnostic is called by selecting its number with switches on the maintenance panel. First, the microprogram computes the location of the microdiagnostic on the tape. The cassette rewinds to BOT, skips over the required number of preceding records, and then reads in a 256 word microdiagnostic into control storage. Only one microdiagnostic may be loaded at one time.

CASSETTE DECK

The cassette deck (assembly A8) contains the tape cassette reader and its associated electronics. The cassette drive mechanism contains the electromechanical components needed to accurately position the cassette with respect to a read head, to provide forward or reverse tape motion, and to sense BOT or end-of-tape (EOT). Two printed circuit cards contain electronics to control motion, process the analog read data read from the tape to digital data, and to interface the deck with the main logic.

Signals provided to the deck from the logic are:

1. Forward – to command forward tape motion. The signal is active low. Tape speed is approximately 12 inches per second.

2. Rewind – to cause reverse motion to BOT. The signal is active low. Total rewind time is about 15 seconds.

3. Operating voltages of +5v and ±24v.

Signals supplied by the deck to the logic are:

1. Cassette Sense – a ground to indicate that the tape is properly loaded and that the latch mechanism is locked.
2. Read Data (active low) – to indicate the detection of a "1" on the tape. Data format is Return to Zero (RZ).
3. Read Clock (active high) – to signify a bit cell time. The relationship of clock and data is shown in Figure 3-13.
4. Leader (active low) – to indicate BOT/EOT. This is a clear section of tape sensed by a phototransistor.

DATA FORMAT

IMPL and MPL data are recorded serially on a single channel. The factory prerecords and verifies the tape; field change is not possible. Cassettes are mechanically interchangeable between controllers.

CAUTION

Although tapes are mechanically interchangeable, data recorded on tapes may differ with controller logic configurations and/or special applications. Refer to the Microprogram Manual for instructions on cassette usability.

Data is recorded in a series of records (Figure 3-14). Each record consists of 1K bytes (diagnostics) or 16K bytes (functional) microprogram data; since four bytes are required to construct a microword, this is equivalent to 256 or 4096 words of control storage. Each byte consists of eight data bits plus one bit to make odd parity (total number of "1" bits per byte, including parity bit, is odd). A gap separates each record from the next record.

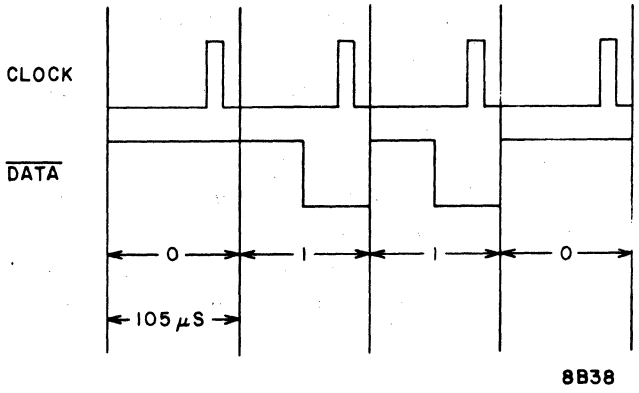


Figure 3-13. IMPL Clock/Data Timing

Table 3-5 lists the function of each record and its control storage addresses. Information on routine numbers versus the function accomplished is contained in the Subsystem Troubleshooting Manual.

The block of 5 hardcore records, 27 inline records, and 1 functional record is written twice on one side of the tape. If the data in the first block becomes unreliable, the second block may be used. The cassette may be flipped over, where the two blocks are recorded again. Both the IMPL circuit function and the MPL function attempt to read the first block before trying the second block.

Reading is possible only while tape is moving forward.

IMPL FUNCTION

The IMPL function is used to read in the first record of data. This function is completely hardware controlled without microprogram intervention. Figure 3-15 shows the applicable logic while Figure 3-16 illustrates the IMPL sequence.

The function starts under either of the following conditions:

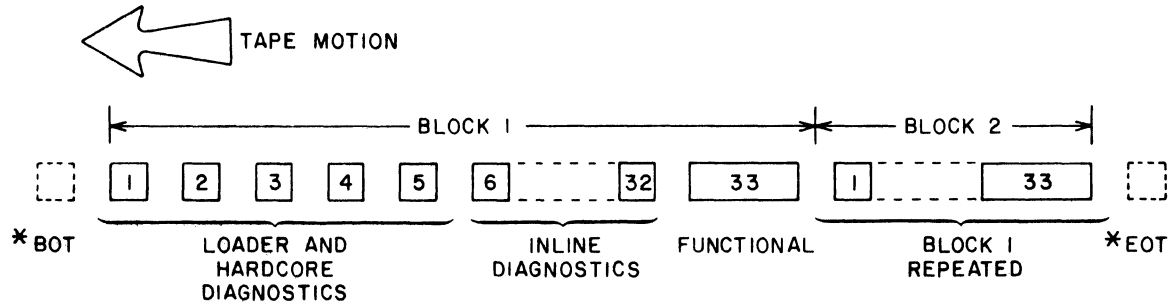


Figure 3-14. Tape Format

TABLE 3-5. CASSETTE TAPE ORGANIZATION

Record	Address (Start/Stop)		Microdiagnostics
	Real*	Pseudo**	
1	0000/03FC	8000/83FC	X020, X022, X027, X029
2	0400/07FC	8400/87FC	X040, X220, X080
3	0800/0BFC	8800/8BFC	X060, X064, X065, X067
4	0C00/0FFC	8C00/8FFC	X108, X116, X900
5	1000/13FC	9000/93FC	X200, X230
6	0400/07FC	9400/97FC	Y260 (Routine 94)
7	▲	9800/9BFC	Y100 (Routine 98)
8		9C00/9FFC	Y140 (Routine 9C)
9		A000/A3FC	Y180 (Routine A0)
10		A400/A7FC	Y160 (Routine A4)
11		A800/ABFC	Y280 (Routine A8)
12		AC00/AFFC	Y120 (Routine A8)
13		B000/B3FC	Y300 (Routine B0)
14		B400/B7FC	Y320 (Routine B4)
15		B800/BBFC	Y340 (Routine B8)
16		BC00/BFFC	Y360 (Routine BC)
17		C000/C3FC	Y380 (Routine C0)
18		C400/C7FC	Y3A0 (Routine C4)
19		C800/CBFC	Y3C0 (Routine C8)
20		CC00/CFFC	Y3E0 (Routine CC)
21		D000/D3FC	Y400 (Routine D0)
22		D400/D7FC	Y420 (Routine D4)
23		D800/DBFC	Y440 (Routine D8)
24		DC00/DFFC	Y460 (Routine DC)
25		E000/E3FC	Y500 (Routine E0)
26		E400/E7FC	Y480 (Routine E4)
27		E800/EBFC	Not used
28		EC00/EFFC	Not used
29		F000/F3FC	Y200 (Routine F0)
30		F400/F7FC	Y240 (Routine F4)
31	▼	F800/FBFC	Y000 (Routine F8)
32	0400/07FC	FC00/FFFC	Y220 (Routine FC)
33	0000/3FFC	0000/3FFC	Functional (Records 1-5 destroyed)

*Real address indicates location data is stored in control storage.

**Pseudo address is arbitrarily assigned in order to distinguish the diagnostic programs from the functional programs which are overlayed into the same address. The pseudo address is a simple addition of 8000_{16} to the real address. All inlines are actually stored in addresses 0400 through 07FC in control storage.

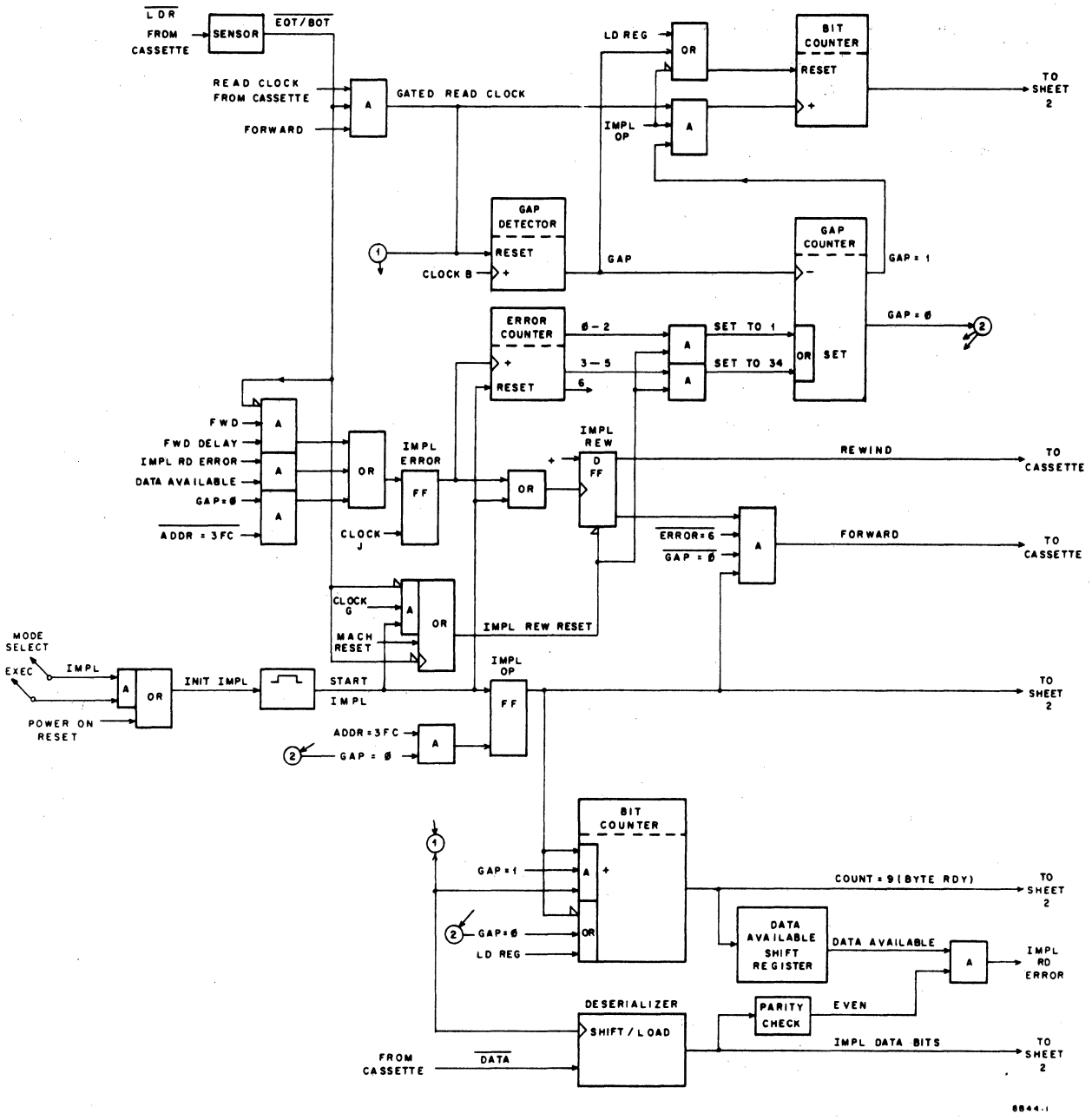


Figure 3-15. IMPL Function Control Logic (Sheet 1 of 2)

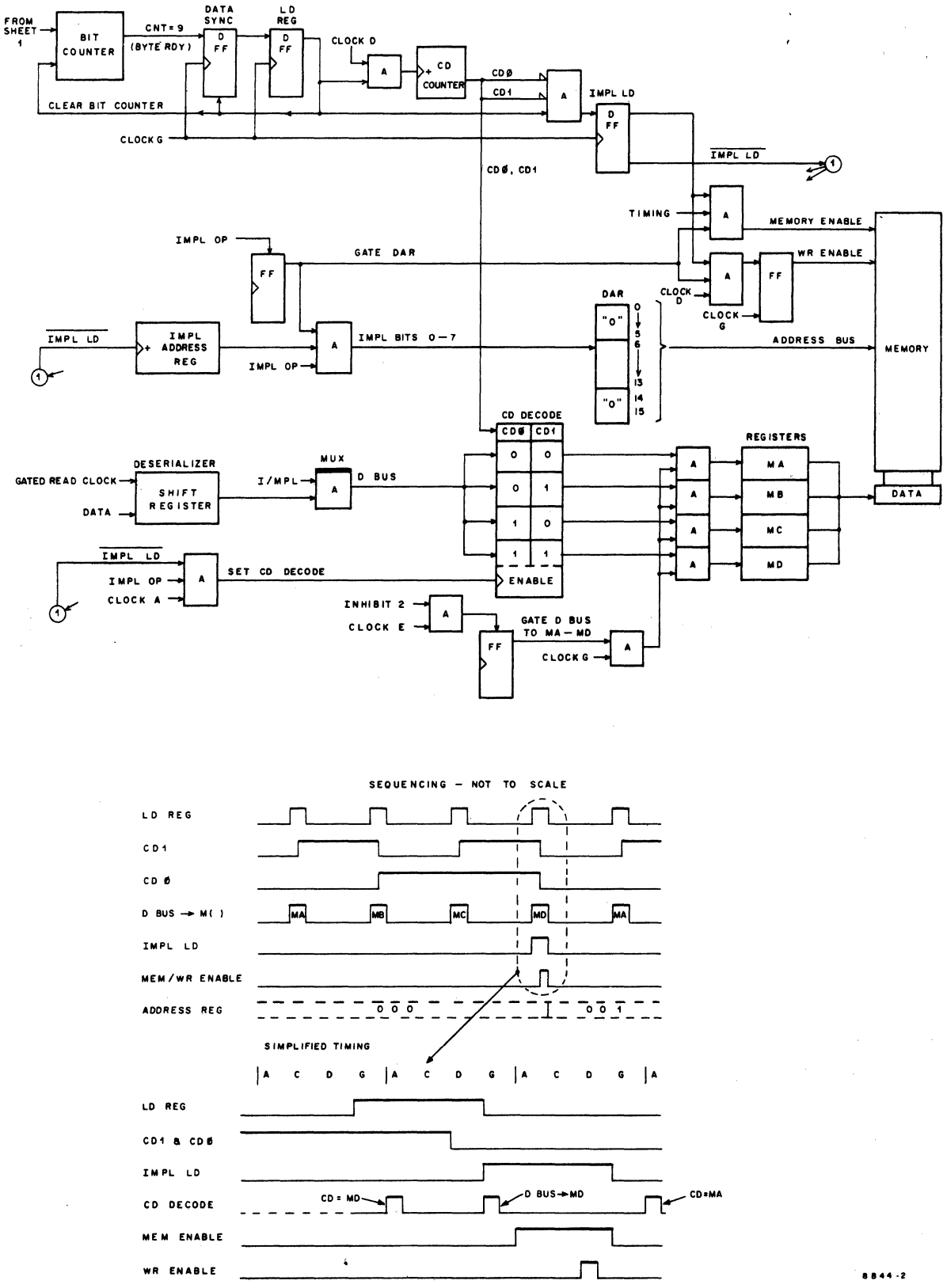


Figure 3-15. IMPL Function Control Logic (Sheet 2 of 2)

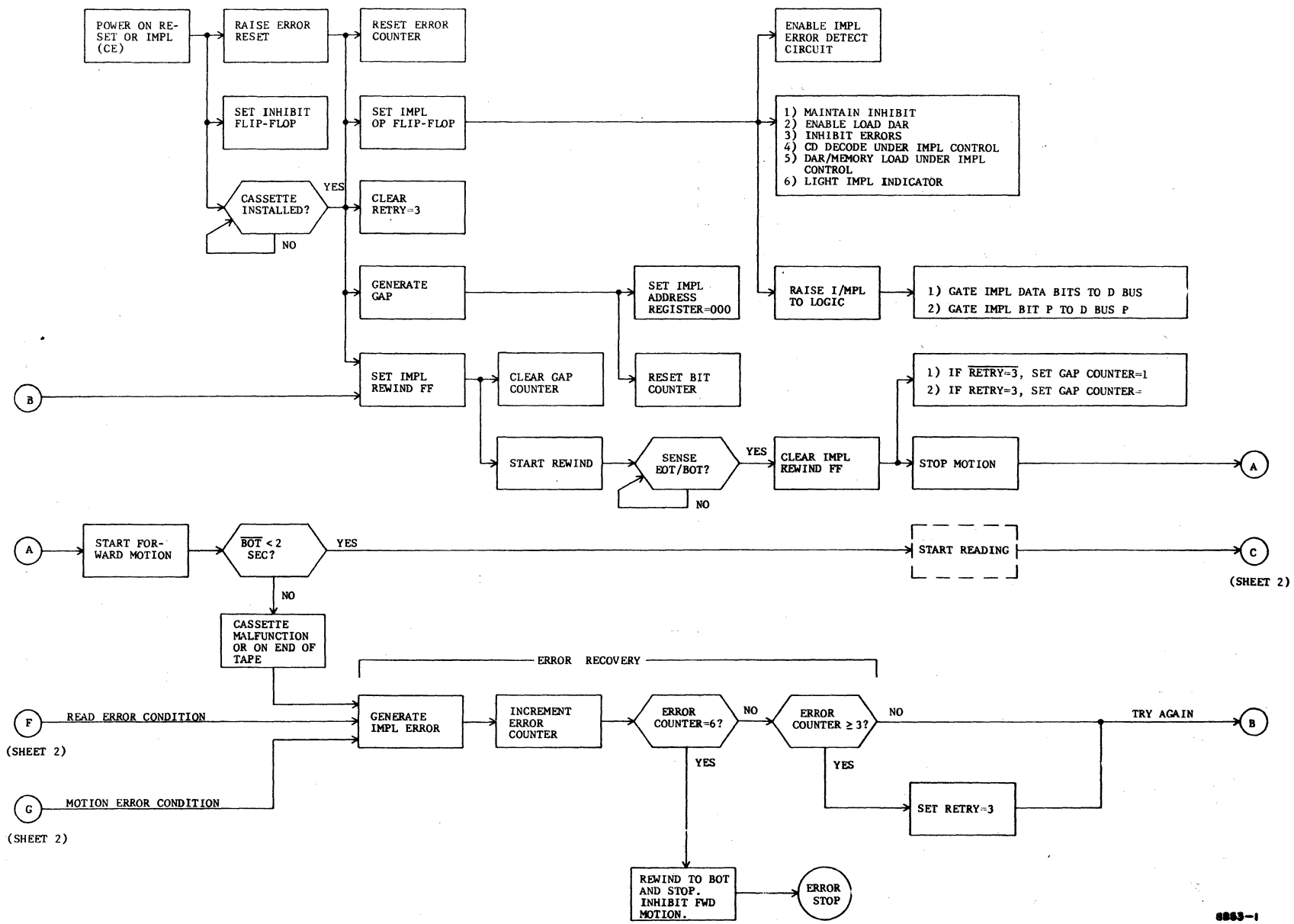


Figure 3-16. IMPL Operation (Sheet 1 of 2)

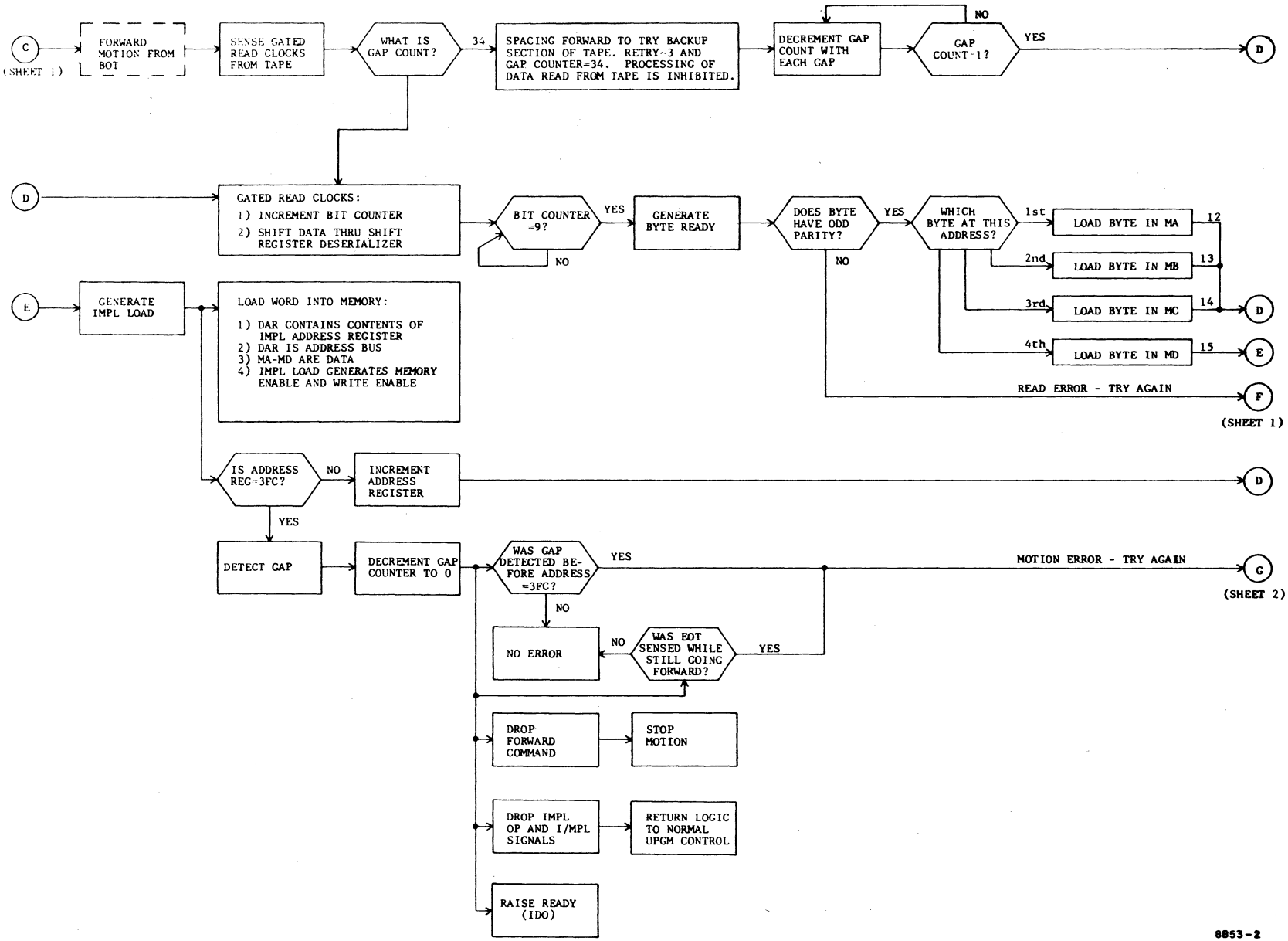


Figure 3-16. IMPL Operation (Sheet 2 of 2)

1. Power On Reset.

2. The EXEC switch on the maintenance panel is actuated while the MODE SELECT switch is in the IMPL position.

The controller then enters the IMPL mode. The Inhibit flip-flops set, to stop all normal machine operations; the process can be stopped only by a Machine Reset. A constant Error Reset signal inhibits all errors except for those generated by the IMPL logic itself.

IMPL Circuit Elements

The following major circuit elements are involved:

1. A shift register used as a deserializer (serial data converted to parallel data). Read Data enters the register, then is shifted down one stage by each Read Clock.

2. The bit counter, which counts the number of Read Clocks sensed. Every group of nine bits indicates that an 8-bit byte has been deserialized by the shift register. (The ninth bit is the parity bit.) A count of nine indicates that the byte may be transferred to the enable destination register.

3. The IMPL Address register specifies where data is to be loaded into control storage. The register output is forced to 000 at the start of the IMPL; its highest count is hex 3FC. The register output is applied to DAR.

4. DAR acts as the source for the address bus to the control storage elements. DAR bits 0 through 5, and 14 and 15 are forced to zero.

5. A 2-bit CD counter counts the number of bytes processed to determine if each assembled byte will be loaded into MA, MB, MC, or MD.

6. A gap detector senses a gap if no Read Clocks are sensed in less than 400 microseconds while moving tape forward.

7. The gap counter counts the number of detected gaps. It is decremented with each gap. This counter must be equal to one to enable reading.

Loading Memory

The Tape rewinds to BOT and starts forward. Each memory word of 32 bits is assembled in groups of four bytes per group. Assume that reading has just started with the Address register at 000 and the CD counter at 0, 0:

1. The first byte is being deserialized. During this time, the CD count of 0, 0 provides a CD (destination) decode of MA register. The contents of the shift register are loaded into MA with each Clock G. At the count of nine, MA is loaded with its final (and now valid) data. The CD counter increments to 0, 1.

2. With a CD count of 0, 1, the CD decode is the MB register. The second byte loads MB until the next count of nine; the CD counter increments to 1, 0.

3. With a CD count of 1, 0, MC loads. At the next count of nine, the CD count increments to 1, 1.

4. With a CD count of 1, 1, the CD decode is the MD register. At the count of 9, the CD counter returns to its 0, 0 state. This sets the IMPL Load FF for one machine cycle; the four bytes in MA through MD can be loaded into control storage.

5. DAR is the Address Bus source. In turn, DAR is fed by the IMPL Address register (still at 000).

6. Write Enable loads the contents of MA through MD into address 000. At the same time, check bits C0 through C7 (computed by the check bit generator) are loaded into memory to complete the 40-bit microword.

7. The IMPL Address register increments to 001.

This process repeats for the remainder of the first record. When the IMPL Address register reaches a count of 3FC, that word is also assembled and loaded. It is the last word of the first record. A gap is detected to set the gap counter to zero. Operation stops. The Inhibit flip-flops are freed and the remaining records of the hardcore diagnostics are loaded in under normal microprogram control (refer to MPL Function). Normally, after executing the hardcores, the microprogram skips over the microdiagnostics and reads in the functional microprogram record. This function overlays all memory locations used by the hardcores.

IMPL Error Recovery

Any error occurring during an IMPL operation will cause the tape to be rewound and the operation to be retried. The retry initiates immediately upon detection of the error. Errors are:

1. If the leader is detected more than two seconds after starting forward motion, either a cassette drive problem exists or the EOT has been sensed.

2. If read parity is not odd at the bit count of nine, the read data is unreliable.

3. If a gap is detected before the IMPL Address register reaches 3FC, the operation is incomplete.

If three errors occur while attempting an IMPL, the first block of data is considered to be unreliable. After the third error, the tape rewinds again to BOT. The gap counter is set to 34_{10} to attempt reading the second block. Setting the counter to 34 has two effects: first, deserializing reading is inhibited because the gap count does not equal one; second, the 33 records comprising the first block are skipped. The counter decrements at each gap. Finally, with the counter at one (at the start of the first record of the second block), read operation continues normally. Any further errors will cause a full rewind to BOT, skip 33 records, then read the 34th. An error count of six (three on the first block, three on the second) will terminate retries. The tape should be removed, turned over, reloaded, and the operation started again.

MPL FUNCTION

The MPL function is used to read in records two through five of the hardcores following an IMPL. It is also used to load in any one of the inline microdiagnostics and the functional microprogram. All operations are under microprogram control.

Figure 3-17 shows the applicable logic while Figure 3-18 illustrates a typical MPL sequence.

MPL Circuit Elements

The bit counter, shift register deserializer, gap detector, and gap counter function as they do in an IMPL operation.

The IMPL Address register, the CD counter, and the IMPL error detection circuits are not used. These functions are under microprogram control. Status and error conditions are applied to the ID assembler for monitoring by the microprogram. DAR and CD (destination) control are also microprogram controlled.

MPL Initiation

An MPL operation can be started only by the microprogram. It computes the difference between the present tape location and the number of gaps forward or reverse to find the desired record. If the record

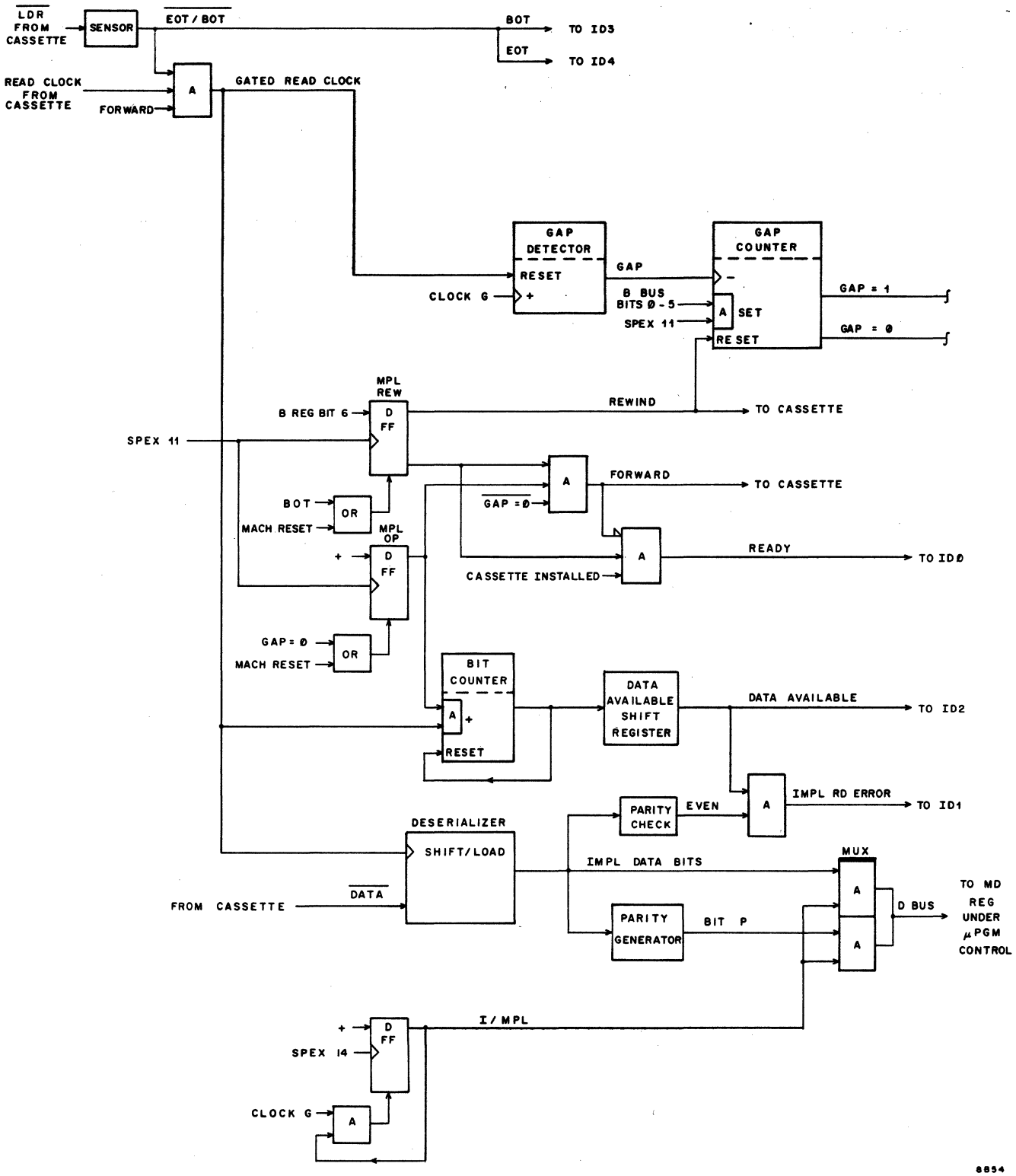


Figure 3-17. MPL Function Control Logic

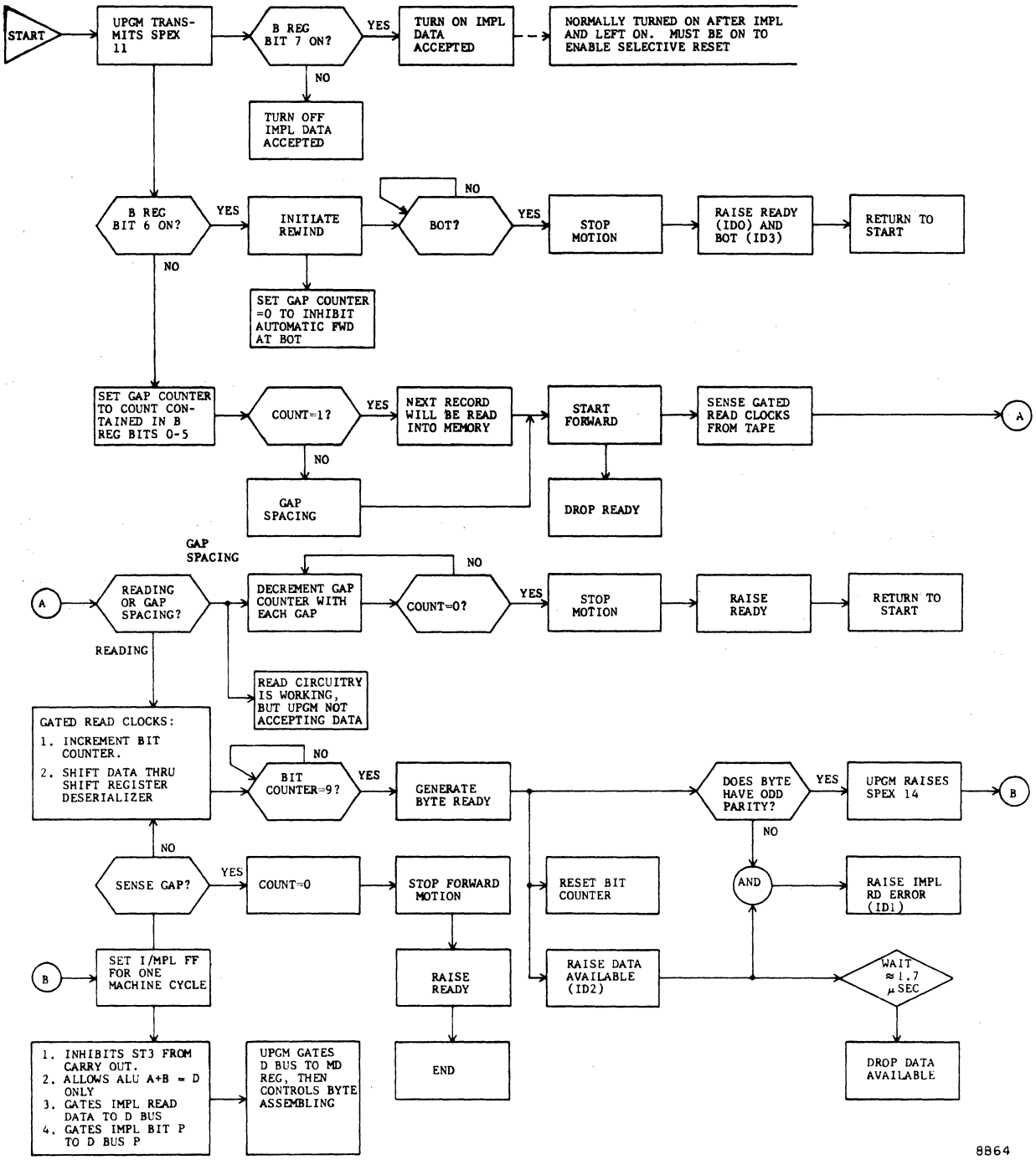


Figure 3-18. Microprogram Load Flow Chart

is forward of the present location, an immediate forward command can be executed. On the other hand, if the record is closer to BOT than the present location, a rewind must first be executed since reverse reading and/or gap counting is unobtainable.

The operation starts by the microprogram placing data on the B Bus and raising SPEX 11. Bit assignments are:

0-5 - number of gaps to be skipped
6 - rewind to BOT
7 - turn on IMPL Data Accepted

Commanding a rewind (bit 6 true) will cause immediate rewind. Another SPEX 11 is required to set the gap counter again and to initiate forward motion. Bit 7 is turned on when the microprogram has completed loading in the functional microprogram. This prevents Selective Reset from having an effect on controller operations until the microprogram can cope with it. Bit 7 is never intentionally turned off during subsequent operations.

When the gap counter counts down to zero, motion stops and Ready is applied to bit 0 of the ID assembler. Another SPEX 11 is issued (B Bus=0000101) and forward motion is initiated. When a byte is available in the deserializer, Data Available is raised for 1.6 to 1.8 microseconds to ID2. The microprogram then issues a SPEX 14 followed by a 0=MD microinstruction. The following events occur:

1. The D bus does not receive its normal output from ALU. Instead, the IMPL Read Data residing in the deserializer is applied to the D Bus.
2. D Bus data is loaded into the MD register.
3. As the deserializer assembles the next byte, the microprogram transfer the byte in MD into MA, MB, or MC as required to assemble a full instruction word. A Store 4 procedure loads the word into control storage. At the same time, the check bit generator generates the check byte to complete the 40-bit microinstruction.

This procedure continues until a gap is detected. At that time the full record has been assembled and loaded. The gap counter decrements to zero; motion stops.

MPL Error Detection

Unlike IMPL operations, hardware does not initiate recovery procedures if an error occurs. The microprogram must monitor the ID assembler for current status and recover from the error. Applicable ID status bits are:

ID0 - Ready
ID1 - Read Error
ID2 - Data Available
ID3 - Beginning-of-tape
ID4 - End-of-tape

ARITHMETIC

INTRODUCTION

The arithmetic function performs the numerical and logical operations on data passing through the controller. This function is controlled primarily by the microprogram. Principal elements of the arithmetic function are:

1. The A Bus Assembler, which selects any one of 31 possible inputs to be applied to the A register or to be displayed.
2. The A register, an 8-bit register serving as one of the ALU inputs.
3. The B Bus, which selects one of the inputs to be applied to the B register.
4. The B register, an 8-bit register which serves as the second ALU input.
5. The Arithmetic/Logical Unit (ALU), which performs the actual arithmetic or logical operation on data.
6. The D Bus, which serves as the logical data source for information to be stored in one of the general purpose registers.

A BUS ASSEMBLER

Function

The A Bus Assembler (Figure 3-19) selects one of the inputs to be placed on the A Bus. Outputs of the A Bus are applied to the following:

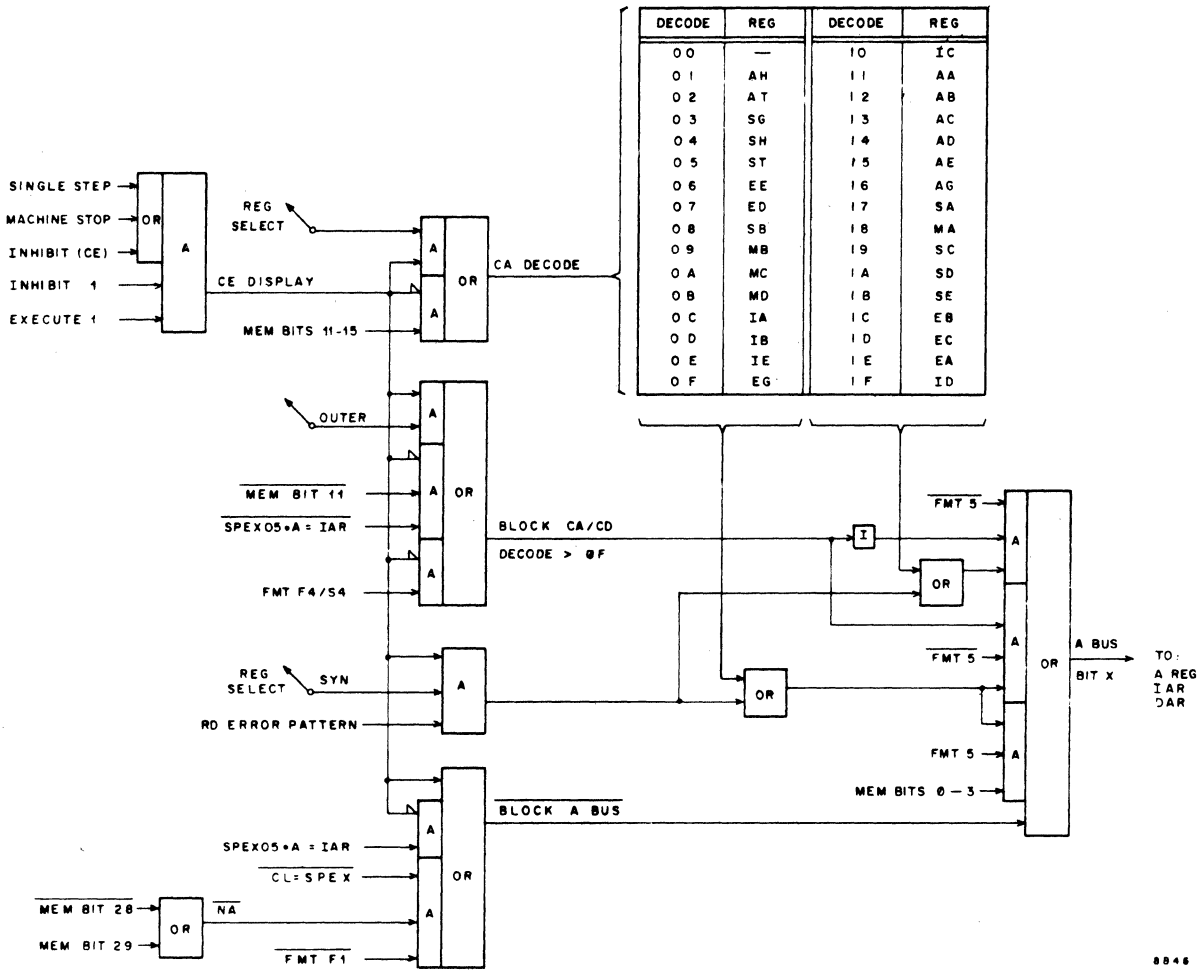


Figure 3-19. A Bus Assembler

1. A register, if permitted by the format decode.
2. IAR in an A-IAR operation.
3. DAR in a format F4 or S4 microinstruction.
4. REGISTER DISPLAY indicators on the maintenance panel.

During regular machine operations, data placed on the A Bus is controlled by the CA/CD or CA field decode.

The A Bus also transmits the parity bit from the selected register for error checking by ALU.

CA Field Decode

The CA or CA/CD field of the instruction word controls the source of data placed on the A Bus. In formats 1, 2, 3, and 4, memory bits 11 through 15 control both the CA and CD fields: the source register placed onto the A Bus is the only register that can receive the data placed on the D bus after an ALU operation. In formats F4 and S4, memory bit 12 through 15 control the CA and CD fields. In format 5, the CA field is controlled independently by memory bits 0 through 3, the CD (destination) field is under control of memory bits 11 through 15. A listing of the effect of the CA and CA/CD fields is provided in Table 3-6.

In CE mode, actuating the EXEC switch on the maintenance panel will gate the register selected by the REG SELECT switch onto the A Bus. The contents of the selected register are displayed by the REGISTER DISPLAY indicators. This data does not enter the A register. If, at the same time, the MODE SELECT switch is in the LOAD DATA position, and OPERATION is in STOP/SINGLE STEP, the contents of the DATA/ROUTINE/PARAMETER switches are loaded into the selected register.

A REGISTER

Function

The sole purpose of the A register is to serve as one of the inputs to ALU. The contents of the A bus are gated into the A register at clock J time if the machine is not stopped. The A register is cleared by a Machine Reset.

When in CE mode with the DISPLAY BUS SELECT switch set to A/B REG, actuating the EXEC switch will cause the contents of the A register to be indicated on the ROUTINE CODE/A REG section of the DISPLAY BUS indicators. Displayed data indicates the last A Bus input to the A register before the machine was stopped. The A register cannot be loaded manually.

Error Detection

To ensure data integrity, the eight A register data bits are compared with the register's parity bit. If the sum is not odd, A Reg Error is generated to produce a Hard Check. The one exception is when the CA decode calls for ST register as the source; A Reg Error is blocked since the ST register does not have a parity bit.

B BUS ASSEMBLER

Function

The B Bus Assembler (Figure 3-20) selects one of the inputs to be placed on the B Bus. Outputs of the B Bus are applied to the following:

1. B register, if permitted by the format decode.
2. IAR in a SPEX 05 operation.
3. DAR in a SPEX 04 operation.

The B Bus also transmits the parity bit from the selected register for error checking by ALU.

TABLE 3-6. CA-CA/CD FIELD EFFECTS

Format	Memory Bits 28(NA)	Memory Bits 29(ND)	A=IAR	SPEX	Gated to A Bus	Gated from D Bus to Destination Reg	Comments
1-4	1	1					Invalid
1-4	0	1	No	No	Any register as as specified by mem bits 11-15	None	Mem bit 11 selects reg with decode greater than 0F
1-4	1	0	No	No	Forced to zero	Any register as specified by mem bits 11-15	
1-4	0	0	No	No	Any register specified by mem bits 11-15	Same register specified as A Bus source	
5	1*	1*	No	No	Any register with decode of 00-0F only	Any register as specified by mem bits 11-15	CA decode is controlled by mem bits 0-3
F1	1*	1*	No	No	Forced to zero	None	No ALU op
F4, S4	1*	1*	No	No	Any register specified by mem bits 12-15 before ALU op applied to DAR 8-15	Any register specified by mem bits 12-15	Only ALU op is A+B=D. CA/CD Regs cannot exceed 0F
					Special Cases		
CE Load	x	x	x	x	Any register	Any register; loaded by LOAD DATA	Register selected by REG SELECT and EXEC
1-4	0*	1*	No	Yes	Forced to zero	None	Mem bits 11-15 specify SPEX number
1,3,4	0*	1*	Yes	05 Only	AE register bits 0-5 before ALU op are applied to DAR 8-13	None	Only ALU op is A+B=D. Permits address branch to any location. AE register is the only A Bus source
1-4	#	#	Yes	No	Any register bits 0-5 specified by mem bits 11-15 before ALU op are applied to DAR 8-13	Any register specified by mem bits 11-15 if ND=0	Permits 64-way address branch

NOTES: *Indicates these bits are forced by format requirement; otherwise, NA/ND bits can be on or off as required by word being executed.

#Indicates NA/ND bits can be on or off as required by word being executed.

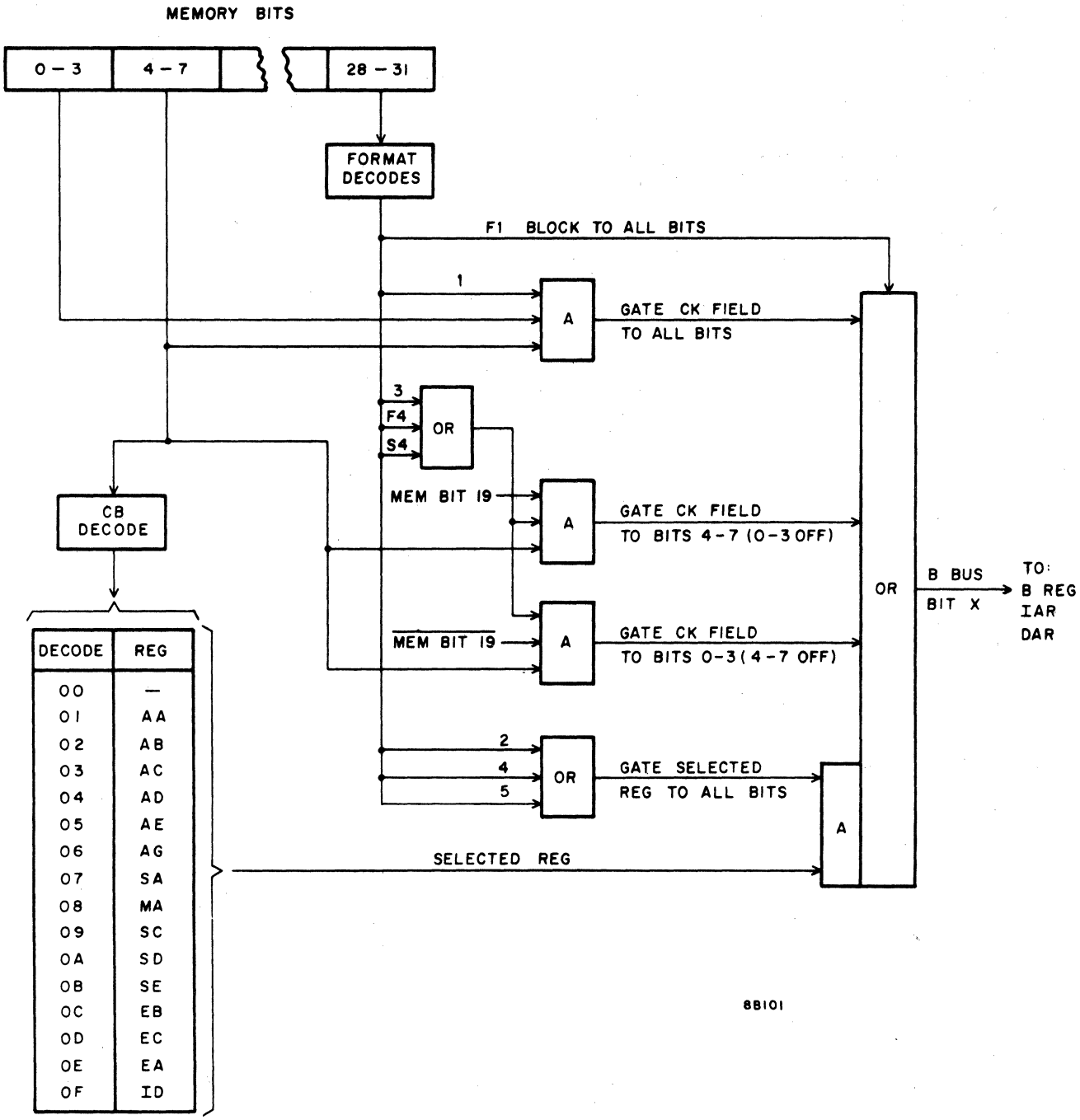


Figure 3-20. B Bus Assembler

B Bus Gating

Depending upon the format selected, B Bus entries may either be data from one of 15 registers or a constant provided by the microwords CK field. B Bus gating and inputs are listed in Table 3-7.

B REGISTER

Function

The B register serves as the second data source input to ALU. The contents of the B Bus are gated into the B register at clock J time if the machine is not stopped. The B register is cleared by a Machine Reset.

The B register has two secondary functions. One is to provide an input to the IMPL gap counter to set the count of the number of gaps to be skipped before reading in data. This function is enabled during a SPEX 11 operation. The second function, enabled during SPEX 22, sets the block inline counter.

When in CE mode with the DISPLAY BUS SELECT switch set to A/B REG, actuating the EXEC switch causes the contents of the B register to be indicated on the ERROR CODE/B REG section of the DISPLAY BUS indicators. Displayed data indicates the last B Bus input to the B register before the machine was stopped. The B register cannot be loaded manually.

Error Detection

To ensure data integrity, the eight B register data bits are compared with the register's parity bit. If the sum is not odd, B Reg Error is generated to produce a Hard Check. The B Bus parity bit is a function of one of the following conditions:

1. If in format 1, it is parity for memory byte 0 (parity bit for memory bits 0 through 7).
2. In formats 3, F4, and S4, it is parity for memory bits 4 through 7.
3. If the CB field is used, it is parity associated with the selected register.

TABLE 3-7. B BUS CONTROL

Format	Mem Bit 19	SPEX	Gated to B Bus	Data Destination
1	x	No	CK Field (mem 0-7)	B Register
2, 4, 5	x	No	Selected register	B Register
3, F4, S4	0	No	Mem bits 4-7	B Register bits 0-3 (4-7 off)
3, F4, S4	1	No	Mem bits 4-7	B Register bits 4-7 (0-3 off)
F1	x	No	Forced to zero	None
1-4	x	04	Depends on format	B Bus bits 5-7 to DAR 1-3
1-4	x	05	Depends on format	IAR bits 0-7

ARITHMETIC/LOGICAL UNIT

Function

The ALU (Figure 3-21) performs either numerical or logical operations on eight bits of data. The numerical operations are addition and subtraction by one's complement addition. The logical operations are AND, OR, and exclusive-OR.

Inputs to ALU are:

- A Register
- B Register
- Carry In

Carry In may be conditional or forced. It is conditional in ALU statements of the general form A+B+C=DC. The carry in is applied to the least significant bit of ALU only if ST3 was set prior to execution of this microword; ST3 off means that there will be no carry in. Carry in is forced, without regard to ST3 status, if the ALU statement is A-B+1=D.

ALU outputs are applied to the D Bus.

A second, or backup ALU, operates in parallel with the primary ALU. Their two results are compared: if not equal, an ALU Error generates a Hard Check. No attempt is made to recover from the error or to correct the answer. The output of the backup ALU is not used for any other function.

OP Field

The type of operation to be performed is specified by the OP field of the microword. The OP field is bits 8, 9, and 10 in most formats. In formats F1, F4, and S4, there is no OP field. This forces an add-only function (A+B=D).

A summary of OP field decodes versus operations performed is provided by Table 3-8.

Carry

Carry is a signal generated whenever the output of ALU overflows (result greater than 255) in an add operation. The Carry FF sets. This flip-flop stores the carry for one instruction cycle. It may be tested by the microprogram by utilizing the CARRY decode in the CH branch. Longer storage of this carry requires an OP code that generates a Gate Carry Out enable. Then, if there is a carry out, ST3 sets; otherwise, it clears.

D=O

The D=0 signal is the logical OR of the eight low order bits of the ALU output. If the result is zero, the D=0 FF sets for one instruction cycle. This may be tested by the microprogram using the D=0 decode in the CL branch. Longer storage of this status uses ST2. To be effective, the CS field of a previous format 3, 4, F1, F4, or S4 microword must perform a 0=ST2. Then, if the current microword has a CS decode of DNST21 (D Not Zero, set ST2 to one), and if the result of this instruction is any value other than zero, ST2 sets. ST2 is not turned off if D equals zero; ST2 is turned off only by a 0=ST2 CS decode.

D=0 does not examine the carry output of ALU. As a result, any given execution can result in both a carry and D=0 status.

D BUS

Function

The D Bus is a three-way selector that serves as the input to all registers enabled by the CA/CD or CD field decode. Its inputs are any one of the following:

- ALU
- Control panel switches
- IMPL circuit

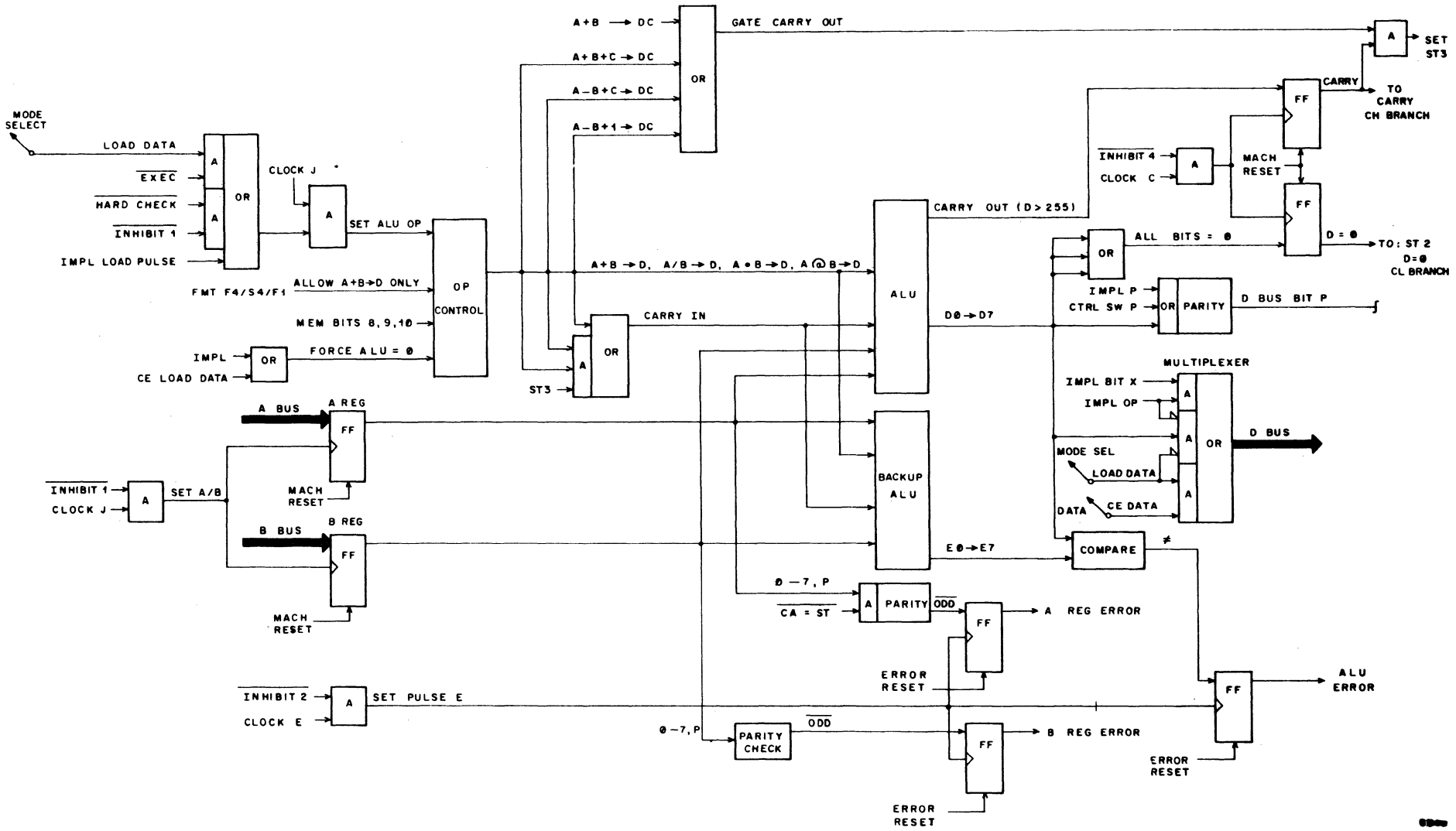


Figure 3-21. Arithmetic/Logical Unit

TABLE 3-8. ALU OPERATIONS SUMMARY

Memory Bits 8	9	10	Sample Microprogram Statement	Function	Operation Example
0	0	0	AE+AD=D	A+B=D (Add)	AE=1100 1010 (202_{10}) AD=0001 1001 (025_{10}) D=1110 0011 (227_{10})
0	0	1	AE+AD=DC	A+B=DC (Add. Set ST3 if carry out. Clear ST3 if no carry out.)	AE=1100 1010 AD=0001 1001 D=1110 0011 (No carry out; clear ST3)
0	1	0	AE+AD+C=DC	A+B+C=DC (Add. If ST3=1, force a carry in. If ST3=0, no carry in. Set ST3 if carry out; clear ST3 if no carry out	AE=1100 1010 AD=1001 1001 Carry in = 0 D=0110 0011 with carry (ST3 was off: no carry in. Carry out: set ST3.)
0	1	1	AE-AD+C=DC	A-B+C=DC (Add one's complement of B Reg to A Reg. If ST3=1, force carry in; if ST3=0, no carry in. Set ST3 if carry out; clear ST3 if no carry out.	Assume AD=1100 1001 (201_{10}) AE=1100 1110 (206_{10}) AD=0011 0110 (054_{10}) Carry in = 1 D=0000 0101 (5_{10}) with carry (ST3 was on: carry in. Carry was generated: set ST3.) NOTE Since subtraction is one's complement, result is mathematically true only if there is a carry in. If no carry in, result is off by one.
1	0	0	AE-AD+1=D	A-B+1=D (Add one's complement of B Reg to A Reg. Force a carry in. ST3 not involved.)	Same as A-B+C=DC above, except ST3 is unaffected.
1	0	1	AE/AD=D	A/B=D (Logical OR)	AE=1100 1010 AD=0001 1001 D=1101 1011
1	1	0	AE·AD=D	A·D=D (Logical AND)	AE=1100 1010 AD=0001 1001 D=0000 1000
1	1	1	AE@AD=D	A@D=D (Logical exclusive-OR)	AE=1100 1010 AD=0001 1001 D=1101 0011

The D Bus generates a parity bit that accompanies each data byte placed on the D Bus.

CD Field Decode

During normal machine operations, the D Bus receives its input from ALU. The destination of the D Bus output is controlled by the CD field. See Figure 3-22. Normally, the CD field is controlled concurrently with the CA field. For further information, refer to CA Field Decode and to Table 3-6.

During SPEX and format F1 operations, the memory word bits normally used to specify destination registers are used for other purposes; therefore, there is no CD control and registers cannot be loaded.

Manual Loading

While in CE mode, data can be manually loaded into a register selected by the REG SELECT switch. With the MODE SELECT switch in the LOAD DATA position, actuating EXEC will load data from the DATA/ROUTINE/PARAMETER switches into the selected register. Note that ALU is bypassed in this function.

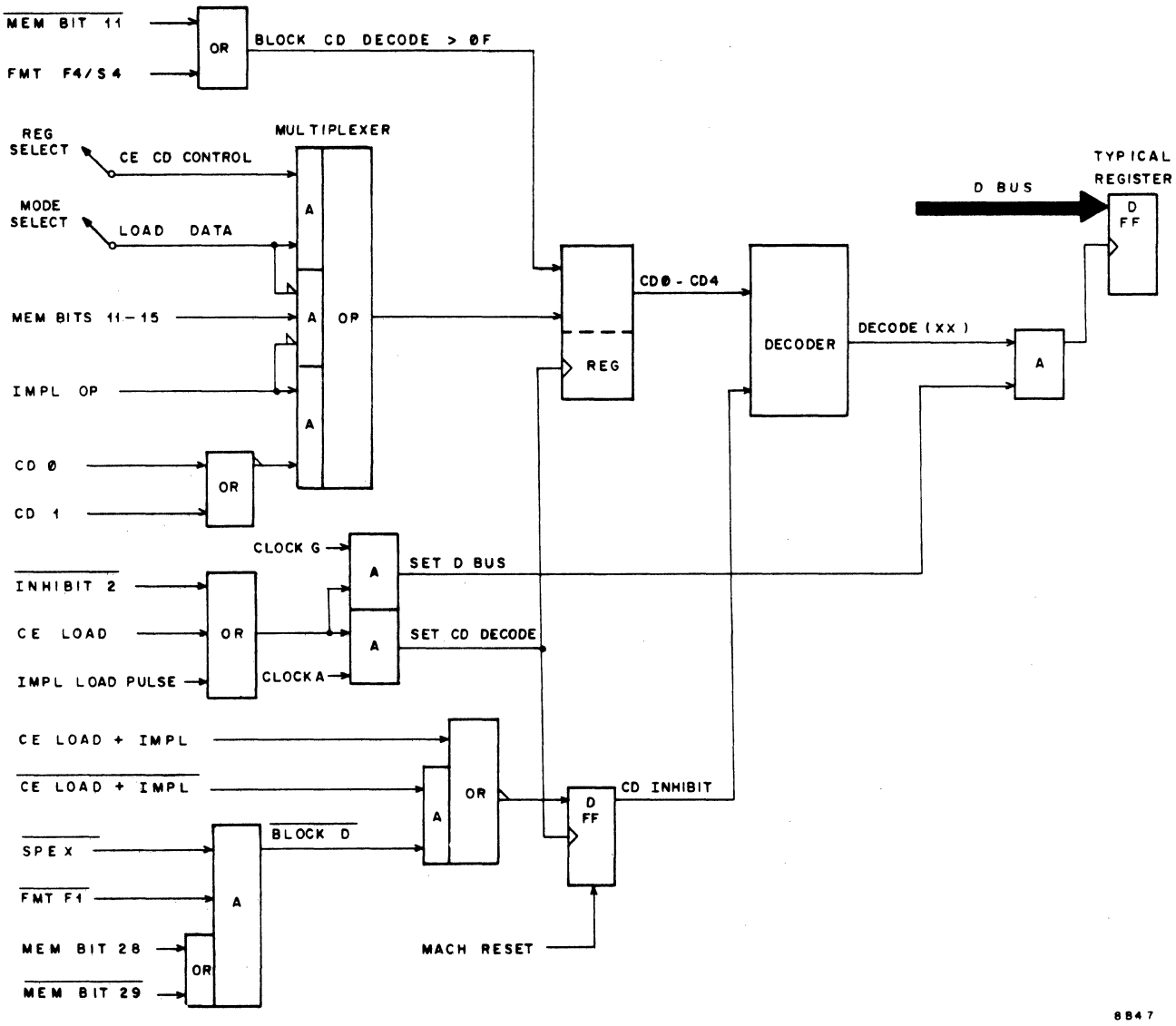


Figure 3-22. D Bus Destination Control

IMPL Loading

During IMPL operations, IMPL Read Data deserialized from the cassette tape are placed on the D Bus for loading into one of the Memory registers (MA-MD). In the IMPL (but not MPL) functions, destination control is provided by a two-bit counter in the IMPL logic. This counter forces a CD decode of MA, MB, MC, or MD. In MPL operations, the microprogram controls the CD decode.

CUDI

INTRODUCTION

The Control Unit-Device Interface (CUDI) logic serves as the interface between the controller and its attached drives. Figure 3-23 is a simplified diagram of signals transmitted between units.

The controller/drive interface is connected in a star configuration, that is, a separate set of cables is connected to each drive. Three cables are in each of these independent connections. One cable supplies ac power to the drive power supplies and convenience outlet. The second cable, the Data cable, contains coaxial conductors for three signals: Read Data, Write Data, and Servo Data. The third cable, the Signal cable, is used for the remaining I/O signals.

SIGNALS TO DRIVE

Before any drive can accept signals from the controller, the controller PCU must supply the -24v Controller Out Power (COP) signal. Refer to the Power Supplies discussion earlier in this section for information on COP generation.

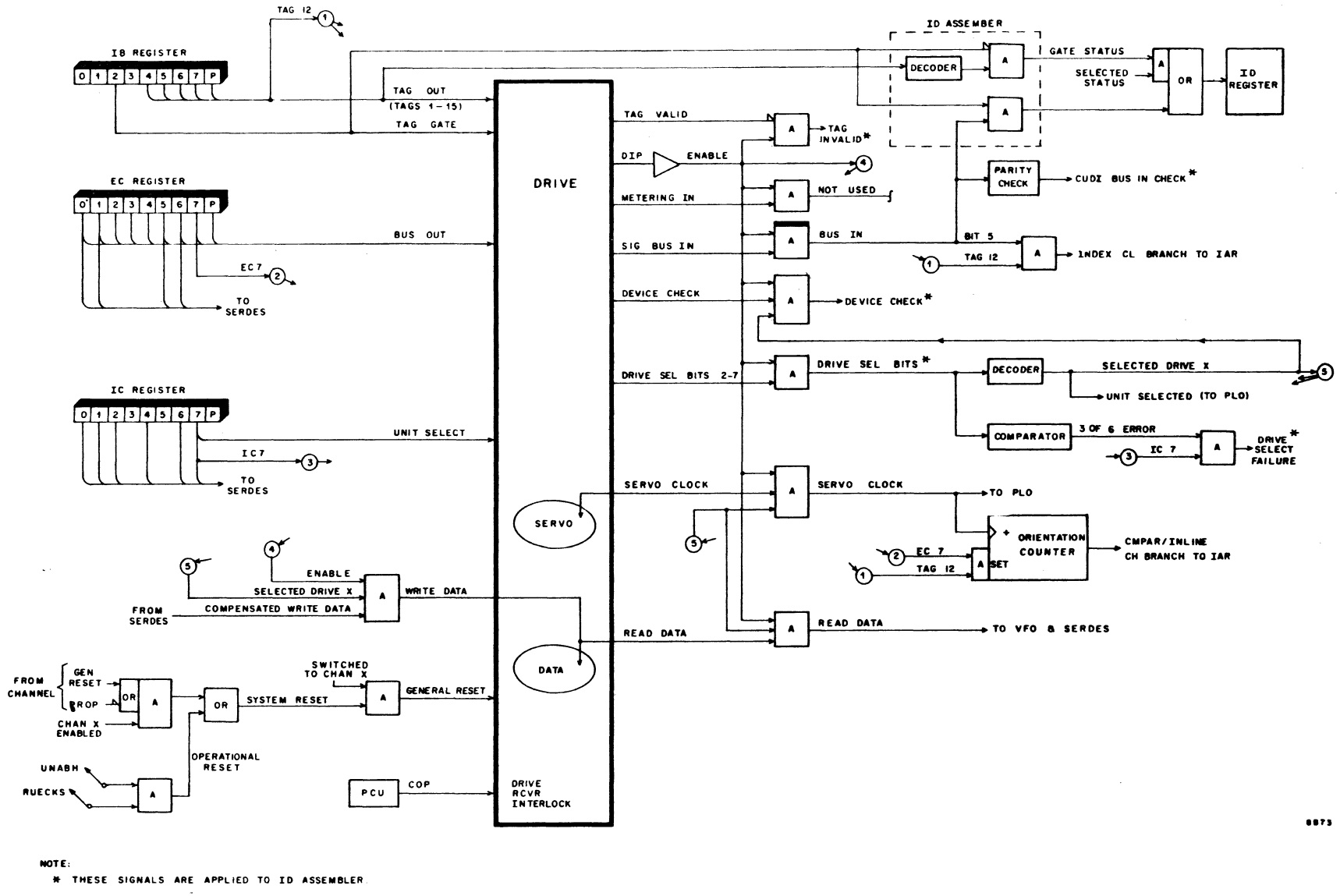
Commands are transmitted to the drive by using a combination of Tag Out and Bus Out lines. Tags are generated by bits 4 through 7 of the IB register. Bus Out, which is the contents of the EC register, has different meanings depending upon which tag signal is active. Bit 7 of the IC register (Unit Select) goes high when the bus/tag lines are stable. The drive decodes one of the 15 tags when the Tag Gate (IB2) is up. All tags except Tag 2 (Poll Devices) and Tag 3 (Transmit Module Address) require previous drive selection.

Table 3-9 lists the bit significance of the IB, IC, and EC registers.

SIGNALS FROM DRIVE

Drive inputs to the controller are the following:

1. Device In Power (DIP), indicating that the unit is powered up and on line.
2. Tag Valid, indicating that the Tag Out parity is odd.
3. Signal Bus In, which are eight lines plus parity that supply information; bit meanings are a function of Tag Out.
4. Device Check, indicating a drive fault or error condition.
5. Metering In A and Metering In B, which are not used.
6. Drive Select bits 2 through 7, indicating the drive physical address (A through H). This address is hard-wired at installation and is known as the "3-of-6" code. Do not confuse this address with the logical address assigned by the drive logical address plug. The logical address is provided on Bus In, in response to Tag 3 (Transmit Module Address).
7. Servo Clock, derived from the servo surface of the disk pack. One pulse is provided once each cell time (1.24 microseconds).
8. Read Data, a pulse indicating that a flux transition has been detected by the selected read head.



NOTE:
* THESE SIGNALS ARE APPLIED TO ID ASSEMBLER

Figure 3-23. CUDI Diagram

TABLE 3-9. CUDI I/O REGISTER FUNCTIONS

Bits	Name	Conditions	Function
		EC REGISTER – BUS OUT CONTROL	
0-7	Bus Out	Tag Out (IB2) On	Places control/data function on Bus Out lines. Function to be accomplished is defined by Tag Out bits (see IB Register). Refer to drive Reference Manual for definition of Tag Out/Bus Out functions.
		NOTE Remaining EC register bits description refers only to Bus Out conditions that also affect controller logic.	
0	AM Operation	Tag 12 On IC4 Off	Enables drive and SERDES logic to write Address Mark (suppressed clock/data).
1, 6	AM Search	Tag 12 On	Enables drive and SERDES logic to search for Address Mark while reading.
5	Write Gate	Tag 12 On	Enables SERDES Write function.
7	Save Sector	Tag 12 On	In drive: sets Sector register to current sector address. In controller: Initializes orientation counter
		IB REGISTER – ID REGISTER INPUT CONTROL	
0, 1	CMPAR Branch Enable (CH Field)	0 0 1 0 1 1	In Line (Repeat or Execute FF) Orientation Counter Rnage Orientation Counter Hit
2	Tag Gate	Off	1. Resets Enable CUDI Checks. 2. Gates status bits listed below to ID register (refer to ID register for bit definitions) in accordance with IB bits 4 thru 7:
		IB-7=0 =1 =2 =3 =4 =5 =6 =7 =8 =9 =A =B =C =D =E =F	FAR bits 2 thru 7 FAR bits 8 thru 13 CU/drive physical address Not used Control Check 1 (Chan interface errors) Control Check 2 (Read/Write errors) Control Check 3 (ECC errors) CUDI checks Hard Checks (SG0-SG6, SH2) Hard Checks (A/B Reg errors in SH) CU Addr, Chan enable Inline Control Sw (bits 0-3) Inline Data Sw (bits 8-15) Channel pointer (Sw to B) Not used Cassette status
		On	1. Raises Tag Gate to drive. 2. Gates CUDI Bus In from drive to ID register.

TABLE 3-9. CUDI I/O REGISTER FUNCTIONS (Cont'd)

Bits	Name	Conditions	Function
		IB REGISTER (Cont'd)	
3	Not used		
4-7	Tag Out	IB2 On	Four control lines to drive that define meaning of control or data on Bus Out lines. Refer to drive manual for complete listing of all Tag Out/Bus Out bit meanings. Tag names are listed below:
		IB4-7=0	Not used
		=1	Transmit Sector
		=2	Poll Devices
		=3	Transmit Module Address
		=4	Request Status
		=5	Request Address
		=6	Transmit Cylinder Address
		=7	Transmit High Cylinder/Difference
		=8	Transmit Head Address
		=9	Transmit Difference or Transmit Offset
		=A	Transmit Control 1
		=B	Transmit Control 2
		=C	Operate
		=D	Request Diagnostic Sense
		=E	Mode & Diagnostic Control
		=F	Wrap
		IB2 Off	Gates status bits to ID register. Refer to IB2 for further information.
		IC REGISTER – FILE INTERFACE CONTROL	
0	Gate PLO		Gates 6.44 MHz output of PLO card to input of VFO card.
1	Gate Data		Gates Read Data signal from drive to input of VFO card.
2	Special Sync		Allows any Read Data bit to set Data Good FF in SERDES. Used for diagnostic purposes: normally, sync byte (hex 19) must be detected to set Data Good.
3	Not Used		
4	Erase	Write Operations	Causes only logical "0's" to be written on disk pack. Clock pulses are not suppressed
5	Not Used		
6	Sync PLO		Reverses effects of IC0 and IC1: degates PLO or Read Data input to VFO card.
7	Unit Select		Provides Unit Select signal to drive.

ID ASSEMBLER

The ID Assembler provides the means for the microprogram to monitor subsystem, controller, and drive status. With IB2 (Tag Gate) on, Bus In data received from the drive is gated to the ID register via the assembler. With IB2 off, the remaining IB register bits control the status bits to be loaded into ID. The ID register is loaded with ID Assembler data each Clock D time; ID cannot be loaded from the D Bus. Table 3-10 lists all inputs to the ID Assembler.

ORIENTATION COUNTER

The Orientation Counter (Figure 3-24) is used to maintain track orientation while the controller is disconnected from the channel during a retry operation. This feature permits the microprogram to: determine that a read error occurred; signal retry; and, at the convenience of the channel, relocate the faulty record.

TABLE 3-10. ID REGISTER BITS

IB Reg Bits 4-7 Decode	ID Reg Bit	Name	Meaning	Source Card
NOTE The following bit assignments are true if IB2=0. If IB2=1, ID register inputs are from CUDI Bus In.				
0	0, 1	Not used		
	2	FAR bit 2	BAR address at Hard Check	A1AM
	3	FAR bit 3		
	4	FAR bit 4		
	5	FAR bit 5		
	6	FAR bit 6		
	7	FAR bit 7		
1	0	FAR bit 8	BAR address at Hard Check	A1AM
	1	FAR bit 9		
	2	FAR bit 10		
	3	FAR bit 11		
	4	FAR bit 12		
	5	FAR bit 13		
	6, 7	Not used		
2	0	Not used		
	1	Physical Address	Controller Physical Address	A2AM
	2	Drive Sel bit 2	3-of-6 code	A2AM
	3	Drive Sel bit 3		
	4	Drive Sel bit 4		
	5	Drive Sel bit 5		
	6	Drive Sel bit 6		
	7	Drive Sel bit 7		
3	-	Not used		

TABLE 3-10. ID REGISTER BITS (Cont'd)

IB Reg Bits 4-7 Decode	ID Reg Bit	Name	Meaning	Source Card
4	0	Buf In Par Check	In Buffer parity was even during buffer write operation	A2BE
	1	Not used		
	2	Buf Addr Check	Buffer address (0-4, P) was even	A2BF
	3	Buf Out Par Check	Out Buffer parity was even during buffer read operation	A2BE
	4	Interface Check A	Faulty Controller/Channel A timing occurred during read/write operation	A2BH
	5	Interface Check B	Faulty Controller/Channel B timing occurred during read/write operation	A2BG
	6, 7	Not used		
5	0	Not used		
	1	Write Parity Check	Parity of byte written did not agree with parity bit in SD register	A2AJ
	2	Read Parity Error	Parity bit generated for deserialized byte did not form odd parity	
	3	Bit Ring Check	Number of bits set in SERDES Bit Ring Counter was even	
	4	Write Compensation Check	Both early and late compensation were applied to clock or data bit written on disk	
	5	PLO Input Check	More than 14 continuous Servo Clock pulses were missing after drive was selected	A2AH
	6	VFO Input Check	Frequency of 2F Clock is not same as 6.44 MHz PLO Out signal	
	7	VFO Phase Error	Rising edge of Not PLO Out signal was not coincident with Not 1F Clock	
6	0	Error—No Input Data	ECC Hardware failed to detect any data transfer during read or write	A2AK
	1	ECC Error, P0 or Write	Reading: P0-P3 parity error Writing: parity error detected in ECC registers in series	A2AL
	2	ECC Error, P1 or P3	P1 or P3 parity error during read	
	3	ECC Error, P2	P2 parity error during read	
	4-7	Not used		
7	0	Drive Sel Fail	3-of-6 Code returned by Drive Sel bits was incorrect	A2BA
	1	Tag Invalid	Drive returned Tag Invalid	A2AA,A2AD
	2	Device Check	Drive returned Unit Check	A2AC,A2AF
	3	Not used		
	4	CUDI Bus In Check	During Tag Out 2, 3, 4, 9, 10, 11, or 12, Unit Bus In bits did not have odd parity	A2BA
	5-7	Not used		

TABLE 3-10. ID REGISTER BITS (Cont'd)

IB Reg bits 4-7 Decode	ID Reg Bit	Name	Meaning	Source Card
8	0	SG Reg bit 0	ECC Error	A1BA
	1	SG Reg bit 1	Double Error	A1BA
	2	SG Reg bit 2	Write Error, Byte 0-1	A1AL
	3	SG Reg bit 3	Write Error, Byte 2-3	A1AL
	4	SG Reg bit 4	Clock Error	A1AK
	5	SG Reg bit 6	DAR Error	A1AJ
	6	SG Reg bit 5	IAR Error	A1AJ
	7	SH Reg bit 2	ALU Error	A1AC
9	0	SH Reg bit 0	A Register Error	A1AC
	1	SH Reg bit 1	B Register Error	A1AC
	2-7	Not used		
A	0	Log Address 0	These two bits (combined) for controller addresses of 0, 4, 8, or C	A2BC
	1	Log Address 1		
	2-4	Not used		
	5	Chan Disabled	Channel selected by IE5 (0=A, 1=B) has its KANAL GESP switch active	A2BE
	6,7	Not used		
B	0	Control Sw bit 0	IN LINE FUNCTION switches	Maint. Pnl.
	1	Control Sw bit 1		
	2	Control Sw bit 2		
	3	Control Sw bit 3		
	4	(Not) 1 CU	2 CU feature installed	A2BC
	5	(Not)Multichannel Switch Active	Either KANAL GESP switch is active or controller is single-channel	A2BF
	6	Exec/Rpt	Diagnostic mode (Execute or Repeat FF)	A1BM
	7	CE Function	0=FUNCTION switch in CE position 1=FUNCTION switch in NORMAL or INLINE position	Maint. Pnl.
C	0	Control Sw bit 8	DATA/ROUTINE/PARAMETER switches	Maint. Pnl.
	1	Control Sw bit 9		
	2	Control Sw bit 10		
	3	Control Sw bit 11		
	4	Control Sw bit 12		
	5	Control Sw bit 13		
	6	Control Sw bit 14		
	7	Control Sw bit 15		
D	0-4	Not used		
	5	Switched to B	Channel logic switched to Chan B	A2BE
	6,7	Not used		
E	0-7	Not used		

TABLE 3-10. ID REGISTER BITS (Cont'd)

IB Reg bits 4-7 Decode	ID Reg Bit	Name	Meaning	Source Card
F	0	Ready	Cassette is installed and tape is not moving	A2BB
	1	IMPL Read Error	IMPL Read Data did not form odd parity	
	2	Data Available	IMPL data byte is available on D Bus for microprogram processing	
	3	BOT	Cassette tape is located at beginning-of-tape leader	
	4	EOT	Cassette tape is located at end-of-tape leader	
	5-7	Not used		

Figure 3-24 illustrates the interaction of the logic and microprogram if an ECC error is detected while reading the count field of data record R9 which, in this example, is located in sector 65. Sequencing is as follows:

1. While the read head is located in the interrecord gap, the microprogram raises the AM Search command.

2. When SERDES detects the three bytes comprising the address mark (AM), the microprogram locks the Read Data signal to the PLO and waits for Data Good (sync byte detected).

3. When the sync byte is detected, the microprogram turns on EC7 (Bus Out bit 7).
   - The drive sector count, which is continuously updated by the drive, is transferred to the drive Sector register. In this example, the register is set to decimal 65.
   - The orientation counter is preset to hexadecimal 10 (decimal 16). This allows a subsequent AM search window to open about 32 bytes before the R9 sync byte in the next revolution.

4. If a read error is detected after reading the ECC bytes, the microprogram:
   - Initiates the controller/channel retry operation.
   - Reads the drive Sector register value, subtracts one (set to 64), and reloads the register by a Transmit Sector command.

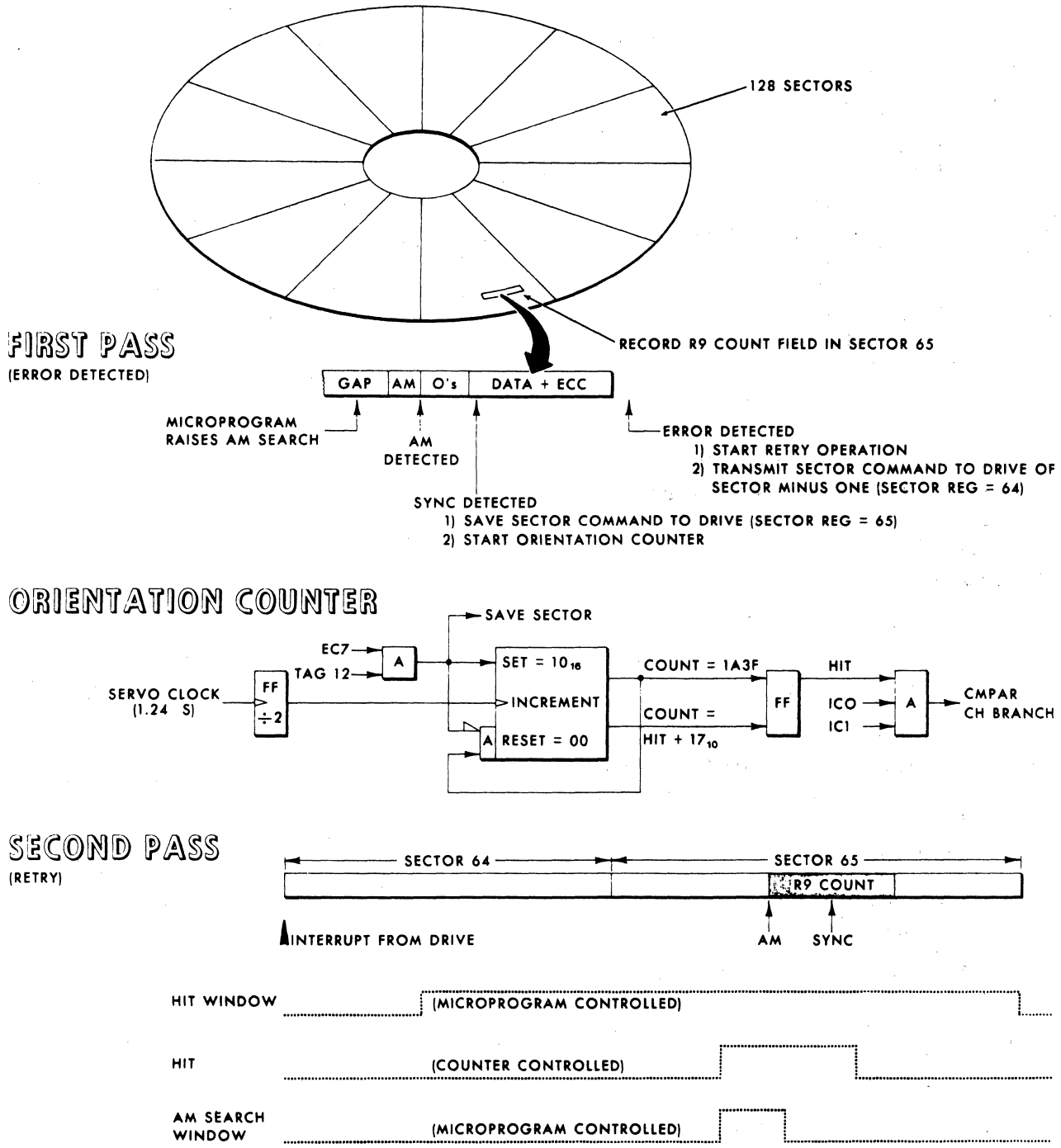
5. The channel can disconnect while waiting for the disk to rotate back to the defective record. During this time, the orientation counter increments once each two bytes.

6. When the drive reaches sector 64, it signals the controller with a Record Ready Interrupt. The microprogram then opens the Hit Window by turning on IC0 and IC1. This enables the counter to provide a CMPAR branch to the CH field of the microword.

7. When the counter reaches 1A3F (decimal 6719), Hit is generated to provide CMPAR. The microprogram initiates an AM search. Also, the orientation counter is reset to zero.

8. When the AM is found, reading procedes normally.

If the channel is not ready to respond in time to process the data after the first revolution, another Hit is generated once each revolution. This process continues as long as necessary.



REFERENCE CONSTANTS (NOMINAL)

DISK PERIOD	: 16.67 MS/REVOLUTION
SECTOR PERIOD	: 130 μS/SECTOR
BYTE PERIOD	: 1.24 μS/BYTE
TRACK CAPACITY	: 13,440 BYTES/TRACK
SECTOR CAPACITY	: 105 BYTES/SECTOR

8B74

Figure 3-24. Orientation Counter

SERDES

INTRODUCTION

The Serializer/Deserializer (SERDES) is used during read or write operations to control transfer of data to or from the disk pack. During read operations, SERDES receives serial (bit-by-bit) data from the drive and converts the data into 8-bit parallel bytes. This is the deserializer function. During write operations, SERDES converts the parallel (byte) data into serial format for transmission to the drive. This is the serializer function. The major components used by SERDES (Figure 3-25) for both read and write operations are:

- Bit Ring Counter – a modulo eight counter used as the timing control. When enabled, it increments once each cell time.
- Shift Register – an 8-bit register used to analyze the serial data pattern. During read operations, the register detects the sync byte (hex 19) at the beginning of each field; while writing, the pattern in the register controls the write compensation network.
- FDR Register – an 8-bit register that buffers incoming and outgoing data: the serializer/deserializer function.

Two other components used in conjunction with SERDES are the SD register and bit 4 of the ST register. The SD register serves as the interface between the microprogram and SERDES: it is the primary data source to SERDES for write operations and receives parallel data from SERDES while reading. Transfer of data between the SD register and SERDES is controlled by SERDES hardware independently of other machine functions. Transfer of data between the SD register and the remainder of the controller is microprogram-controlled. Bit 4 of the ST register is turned on by SERDES to indicate to the microprogram that a byte has been processed– if reading, a byte has been placed in SD for microprocessing; if writing, the microprogram may load a new byte into SD for serializing. ST4 can be turned off only by the microprogram.

Information on data recording formats, record gaps, and address marks are contained in Section 1 of this manual. Refer to the drive Reference Manual for information on MFM recording techniques.

SERDES CLOCK

Introduction

SERDES contains its own timing control to maintain synchronism of data transfer between the controller and drive. This timing is largely independent of the machine clock.

Phase Lock Loop

The heart of the timing is the phase lock loop (Figure 3-26). This type of circuit is used in several places to permit locking or synchronizing an oscillator frequency with an incoming signal.

The phase lock loop is a feedback system comprised of a phase comparator, a low pass filter, an error amplifier, and a voltage-controlled oscillator (VCO). The loop is considered as locked when the output of the VCO is identical in frequency and phase with the external input. When locked, the phase comparator (a flip-flop) is in the set state 50% of the time and in the reset state 50% of the time. A phase change, indicating that the incoming frequency is changing, causes an asymetrical comparator output. The resulting error voltage is proportional to the phase and frequency difference between the two signals applied to the comparator. This error voltage is then filtered and amplified to serve as the control voltage to the VCO. The control voltage causes the VCO frequency to vary in the direction needed to eliminate the frequency difference.

The terms "phase lock loop" and "phase locked oscillator" are interchangeable.

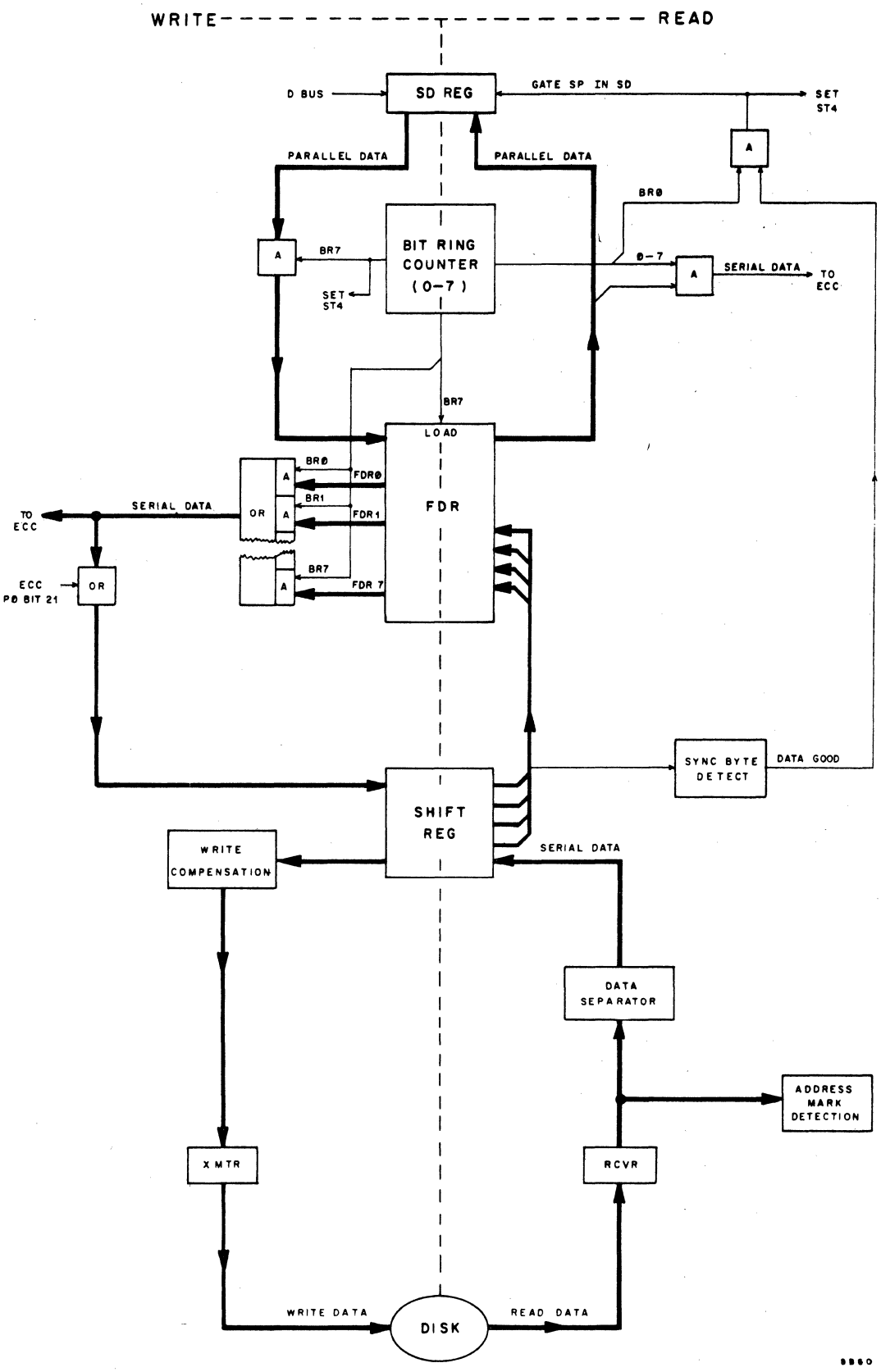


Figure 3-25. SERDES Block Diagram

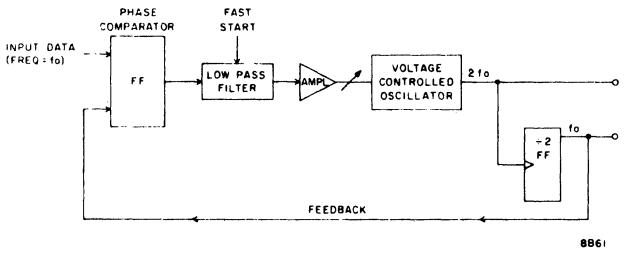


Figure 3-26. Phase Lock Loop

PLO Synchronization

When a drive is not selected, a crystal oscillator circuit in the PLO card (Figure 3-27) acts as the data input to the comparator. This artificial read data signal maintains the PLO timing at a frequency close to the drive's data rate to minimize lock-on time.

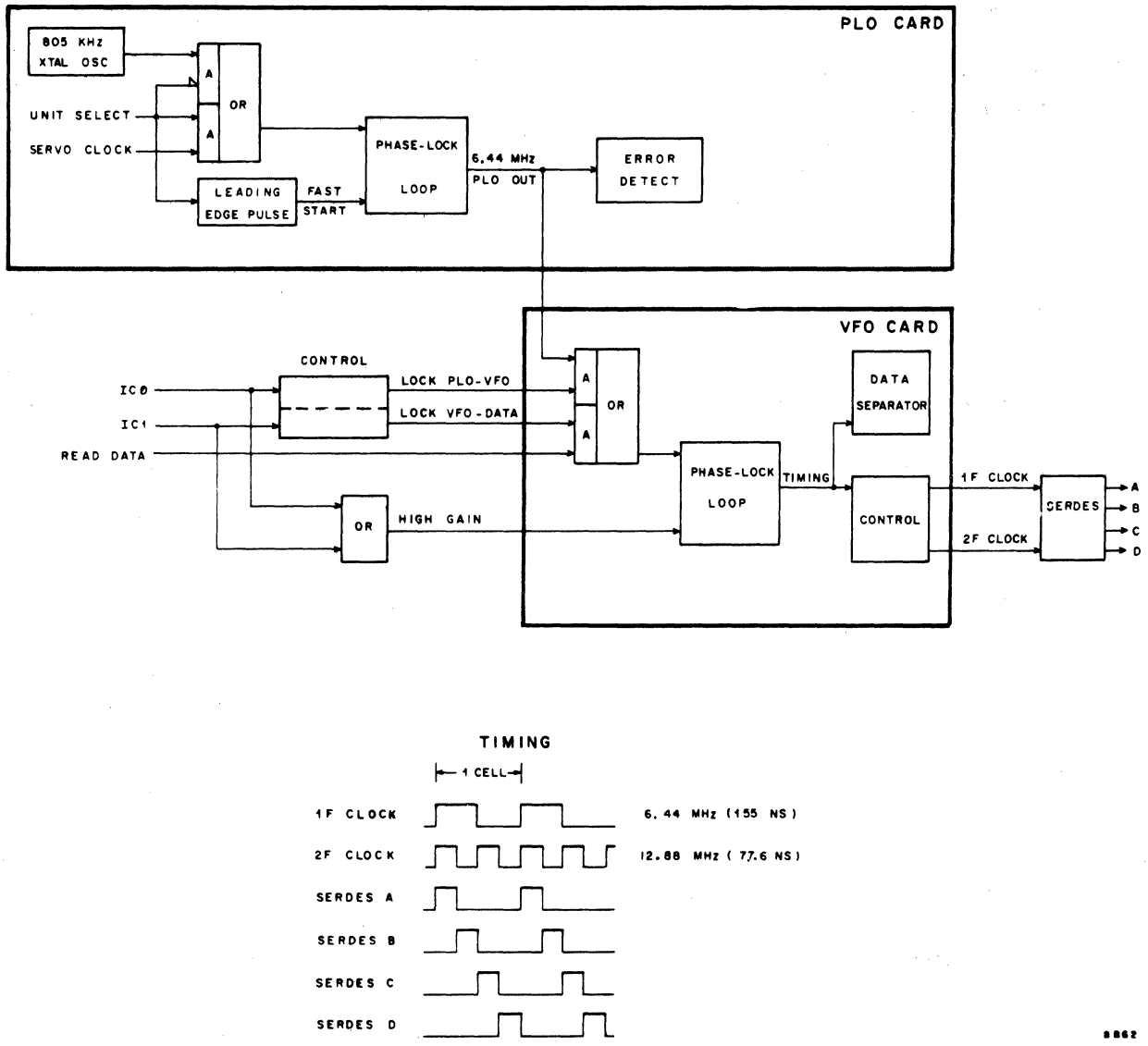


Figure 3-27. SERDES Clocks

Selecting the drive transfers the PLO input from the crystal oscillator to the servo clock signal derived from the disk pack servo dibits. A 50-microsecond fast start pulse increases loop gain to assist in rapid frequency synchronization. After the fast start timeout, the PLO is synchronized with the disk pack servo frequency.

VFO Synchronization

The variable frequency oscillator (VFO) card contains a phase lock loop to provide the SERDES clock source. During write operations or in address mark searches while reading, the VFO is slaved to the Servo Clock. This synchronizes the circuitry with the disk pack rotational speed, which can vary slightly between drives. Then, when real data is to be read, Read Data acts as the data source to the VFO – the VFO phase lock loop can then lock in on the data phasing. Turning on IC0 or IC1 provides a 20-microsecond fast start synchronizing time.

The 1F and 2F clock signals are used to derive the four SERDES clocks: A, B, C, and D.

Since read operations are always initiated while in the inter-field gap, and because the gaps always contain logical o's, the VFO phase lock loop output is phase-locked accurately enough to define the difference between an MFM clock pulse and an MFM data "1" pulse. The data separator circuit uses this information to separate the clock/data pulses and provide SERDES with defined logical 1's and 0's. Loop gain (rate of output frequency change) is low enough to prevent the oscillator frequency from drifting off of the data frequency.

Timing Errors

Error detection circuits in the PLO card check for proper synchronization. Any errors that could degrade accuracy generate a Control Check (File Interface Check). These errors, and their inputs to the file ID assembler are:

1. PLO Input Check (ID5) indicates that more than 14 continuous Servo Clocks were missing after the unit is selected.

2. VFO Input Check (ID6) indicates that the frequency of the 2F clock is not the same as the 6.44 MHz PLO Out signal.

3. VFO Phase Check (ID7) indicates that the rising edge of the Not PLO Out signal is not coincident with Not 1F Clock.

SERDES WRITE

When the microprogram raises tag 12 (Operate) and EC Register bit 5, writing is enabled. Data is transferred from the SD register, serialized, then transmitted to the drive one bit at a time. Circuit elements used in writing are illustrated in Figure 3-28.

Typical Write Sequence

1. Prior to Unit Select (IC7), the Bit Ring Counter is reset and the VFO runs freely.

2. Selecting the drive allows the PLO to lock in on servo pulses from the drive as shown in Figure 3-27. The Bit Ring Counter (Figure 3-28) starts counting.

3. Raising IC0 (Lock PLO to VFO) syncs the VFO to the drive servo pulses.

4. After the microprogram turns on Write Gate (EC5) and, at the trailing edge of the next Bit Ring count of seven (BR7), data is transferred from the SD register to the FDR register.

5. A series of gates now acts as a serializer to convert parallel data in FDR into the serial data to be written on the disk. As the Bit Ring Counter advances, it samples FDR, one bit at a time, to generate Serial Data. If an FDR bit is set while the corresponding

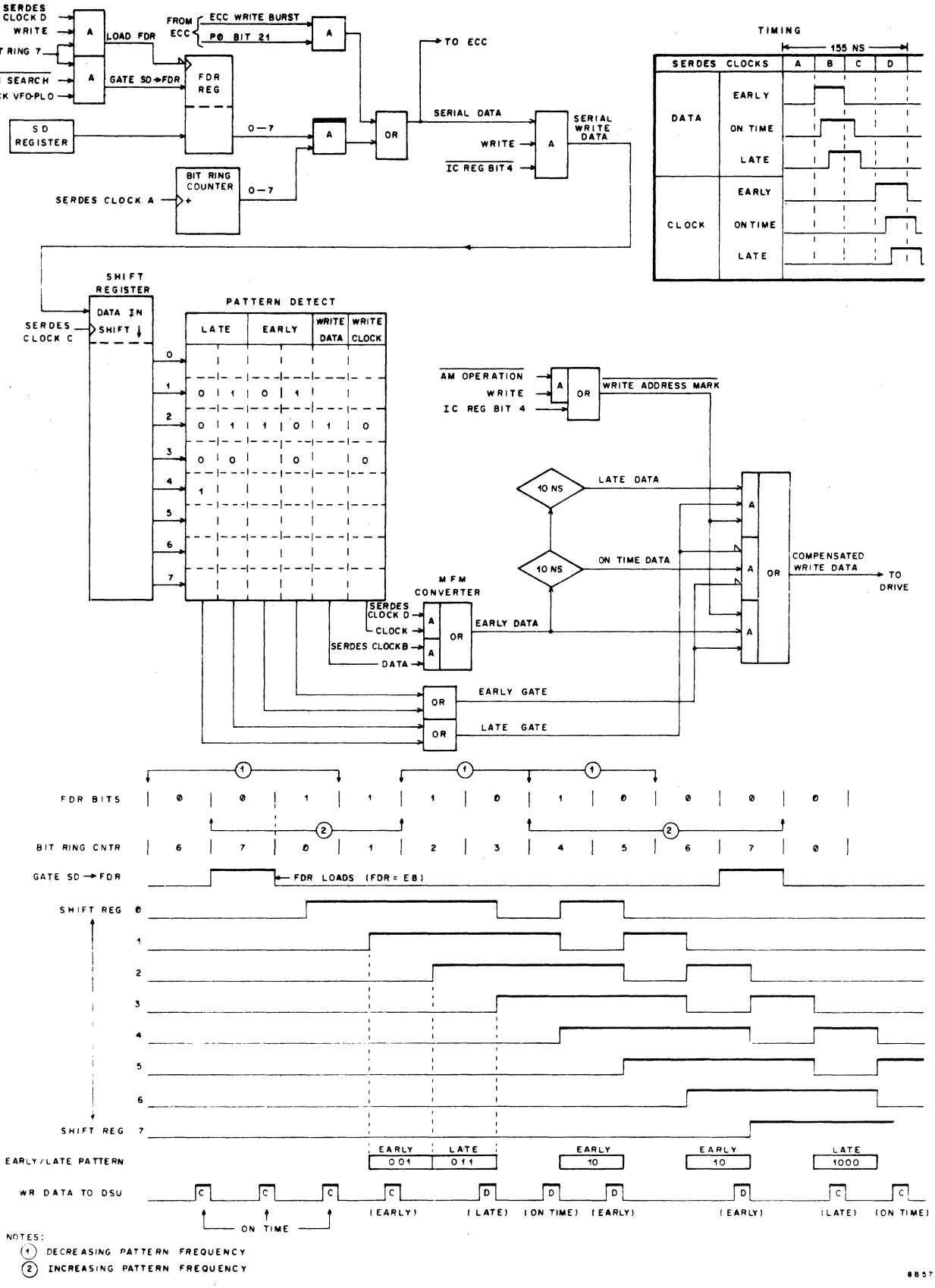


Figure 3-28. SERDES Write Circuit

bit ring count is up, a "1" will be written; if the FDR bit is off, the data is "0".

6. At the next BR7 count, ST4 is set to inform the microprogram that SERDES has finished serializing the byte. The microprogram must then load another byte into SD in time for it to be serialized.

Write Compensation

The write compensation circuit converts the shift register data into MFM data while compensating for a read condition known as peak shift. Peak shift is an effect that degrades read accuracy by distorting the waveform.

With modern high frequency recording techniques, adjacent clock/data pulses are close enough to interact with each other. Peak shift is the result of the interaction of the pulses. Because two pulses tend to have a portion of their individual signals superimpose themselves on each other, the voltage induced in the read/write head during reading is the algebraic summation of the pulses.

When all "1's" or all "0's" are being recorded, the data frequency is constant: pulses are placed apart by one cell (155 nanoseconds). As a result, the pulse spacing causes the overlap errors to be equal and opposite – they cancel out each other. The total net peak shift is, therefore, zero. This is the "zero peak shift" condition of the "...111..." pattern shown in Figure 3-29.

Peak shift occurs when there is a change in frequency increase since there is a delay of about 1.5 cell between the "01" and only 1.0 cell between the "11". As a result, the squeezing of the cell causes the mathematical average (the actual readback voltage) to shift the apparent peak to the left. This is early peak shift.

On the other hand, a "10" pattern represents a frequency decrease since a pulse is not written at all in the second cell. In addition, a "001" pattern is also a frequency decrease since there is a 1.0 cell interval between the first two bits and 1.5 cell between the last two bits.

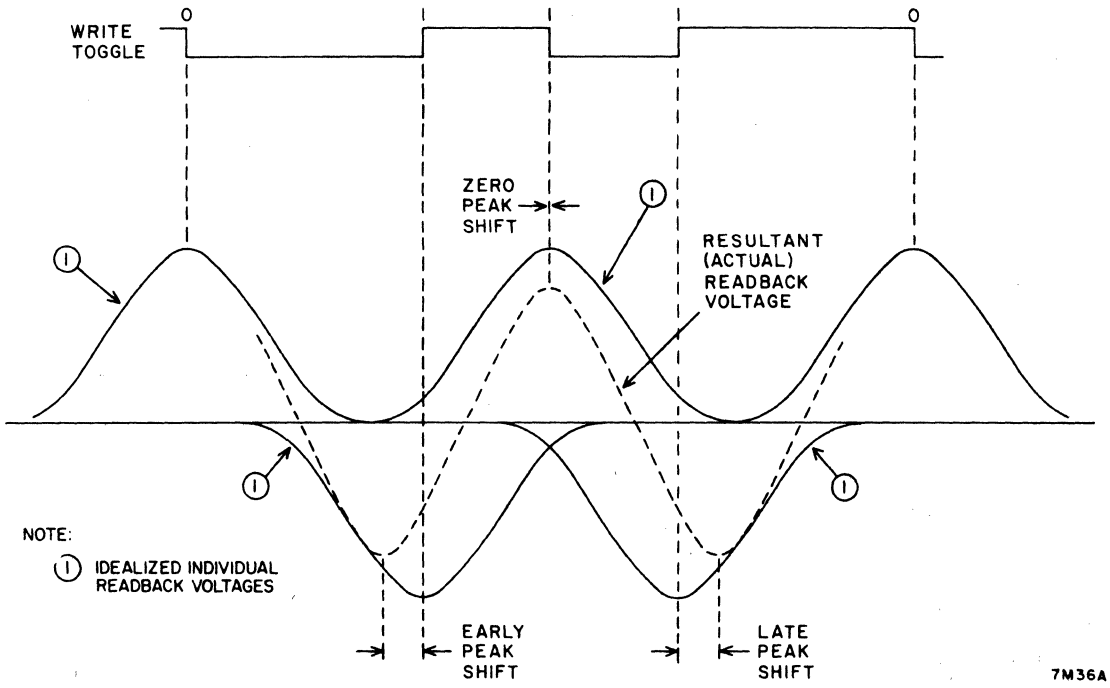
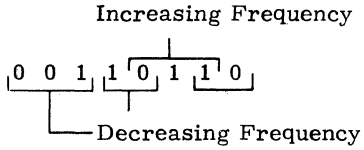


Figure 3-29. Peak Shift

The examples listed above examined only two or three bits without regard to the preceding or subsequent data pattern. The actual combinations are somewhat more complex. The write compensation logic examines and defines the following patterns:

Pattern	Frequency Change
011	Increasing
1000	Increasing
10	Decreasing
001	Decreasing

Any data pattern will have considerable overlapping of the data pattern frequency changes. Consider the overlap of these eight bits:



Any of these peak shift conditions can cause errors during subsequent read operations. The write circuit compensates for these predictable errors by intentionally writing a pulse earlier or later than nominal.

As shown in Figure 3-28, data is serialized by comparing the state of each FDR bit with the corresponding bit ring count; the result is Serial Data. Serial Data is continuously loaded into, and shifted through, the shift register. As the data shifts through the register, the register contents are examined by a series of gates to analyze the bit pattern. These gates determine if the incoming data frequency is constant (00000 or 11111), increasing (011 or 1000), or decreasing (10 or 001). The timing of the Compensated Write Data pulses are adjusted to compensate for the frequency shift as follows:

1. If frequency is constant, there will be no peak shift. An MFM Clock pulse is written during SERDES Clock D. An MFM Data pulse is written during SERDES Clock B. The pulse, clock or data, is intentionally delayed by 10 nanoseconds before being transferred to the drive.

2. If frequency is decreasing, the apparent readback peak (Figure 3-29) would occur later than normal. To compensate for this, the data is written earlier than nominal. Early Gate is enabled. This causes clock/data to bypass the 10-nanosecond delay.

3. If frequency is decreasing, the apparent readback peak would occur earlier than normal. Therefore, data is intentionally written later than nominal. Late Gate is Enabled. This causes the write data to be written 20 nanoseconds after the beginning of the cell.

As Figure 3-28 indicates, MFM Clock is written whenever there is a decode of 00 along with SERDES Clock D. The clock is advanced or retarded if there is a simultaneous early or late decode (1000, etc.). The MFM Data signal is written whenever there is a decode of "1" along with SERDES Clock B. Data may also be written on time, advanced, or retarded.

The change from writing data to writing the ECC bytes is controlled by the ECC logic as directed by the microprogram. After the final bit of data has been written, the SERDES serializer is degated from the write circuitry and the ECC bytes are shifted, one bit at a time, to the write circuitry.

Writing Address Marks

Address marks (three bytes of missing clock and data) are written when the microprogram raises Tag 12 with Bus Out Bit 0 (Write AM) on. SERDES continues to function as it does in any other write operation, except that the Compensated Write Data output to the transmitter is inhibited.

If IC register bit 4 (Erase) is set, no data bits are recorded. An all zeros pattern (MFM Clock pulses only) is written on the disk.

Write Errors

Any errors detected during SERDES write operations are reported to the microprogram as Control Checks. The errors are gated to the ID Assembler when IB2=0 and IB4-7=5. The following bits apply to write operations:

- ID1 (Write Parity Check) – the parity of the byte written on the disk did not agree with the parity bit transferred with the byte from the SD register.
- ID3 (Bit Ring Check) – while SERDES was active, the number of bits set in the Bit Ring Counter was even (0, 2, 4, 6).
- ID4 (Write Compensation Check) – both early and late compensation were applied to a clock or data bit written on the disk.

SERDES READ

Reading is enabled when the controller raises Tag 12 along with Bus Out bits 2 (Not Squelch), 3 (Head Select), and 6 (Read). Data read from the disk is monitored, but not transferred to microprogram control, until the sync byte of hexadecimal 19 (Figure 3-30) indicates the start of a new field. Data Good FF sets; all further incoming data is deserialized and transferred to the SD register.

Data Separator

The data separator (Figure 3-31) separates the data "1's" from the clock pulses.

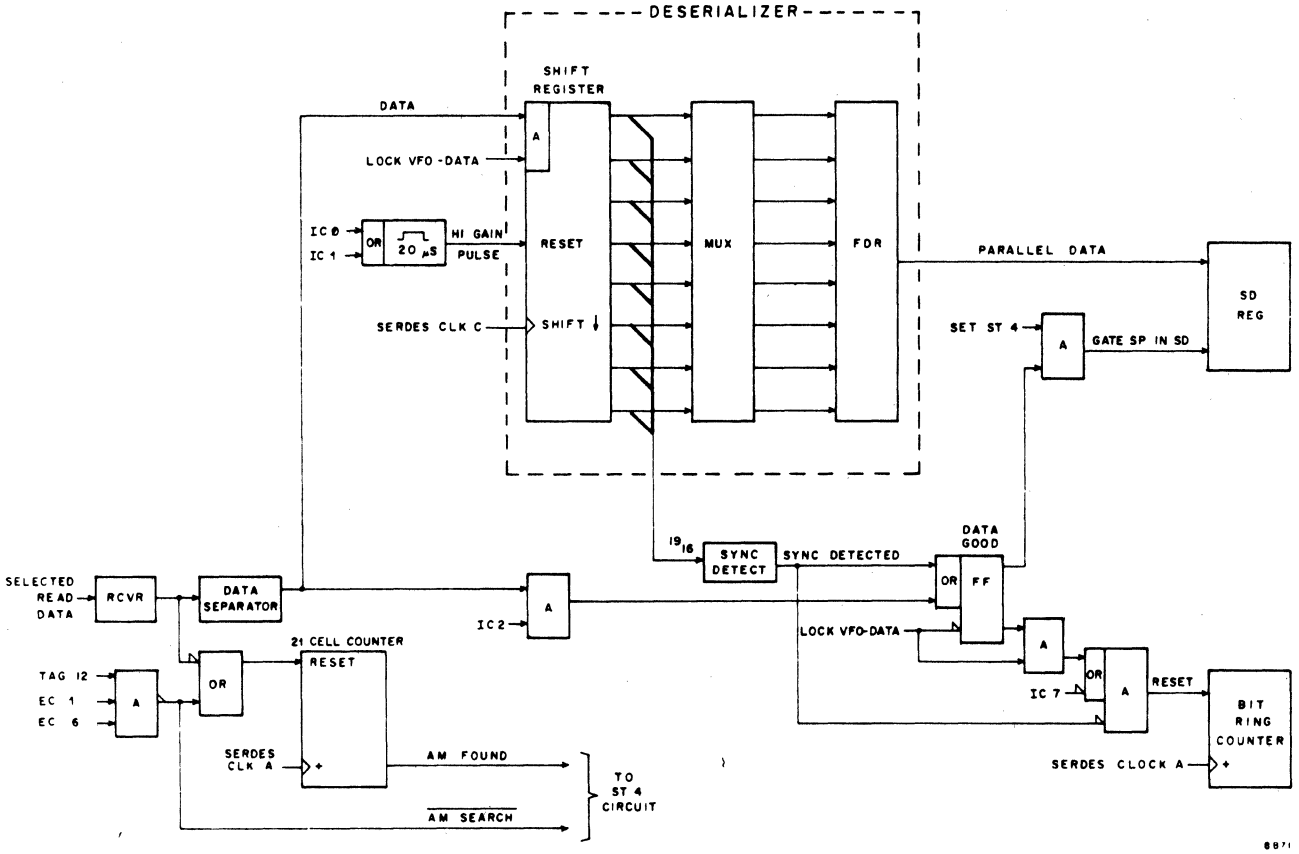


Figure 3-30. SERDES Read

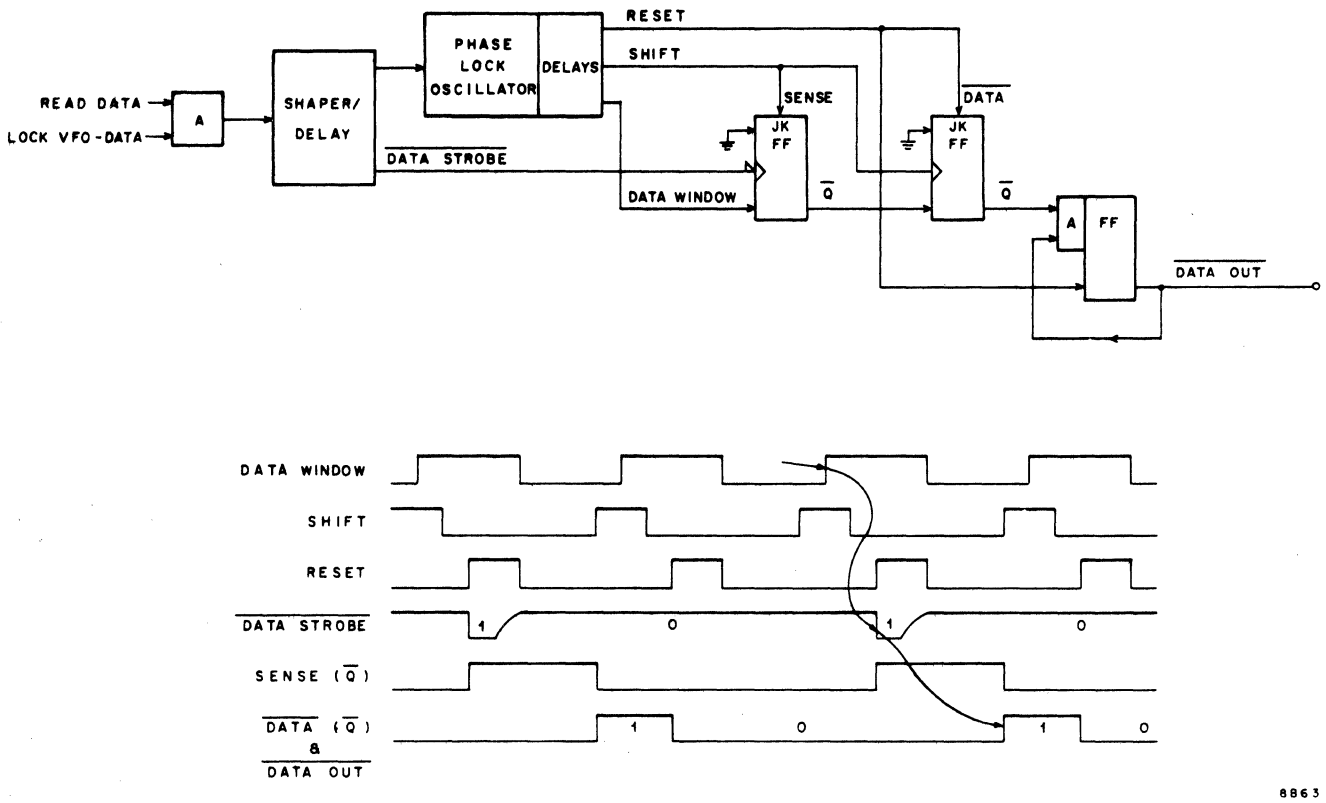


Figure 3-31. Data Separator

Prior to the read operation, the data separator phase lock oscillator was maintained in frequency synchronization by having servo clock signals gated to it (Lock VFO to PLO). When the microprogram raises ICO (Lock VFO to Data), the following events occur:

1. Data received from the drive is gated to the oscillator.
2. A 20-microsecond high gain pulse is applied to the oscillator to assist in rapid phase synchronization. During this time, the oscillator resyncs its phasing from servo clocks to the MFM "0" clock pulses in the inter-field gap. This high gain pulse also holds the SERDES shift register in a reset condition.

After synchronization, the data separator can differentiate between clock pulses and data "1" pulses. Because the circuit is now synchronized in both phase and frequency, any Data Strobe pulse occurring in the middle of Data Window may be defined as a data "1"; a Data Strobe occurring during Not Data Window is a "0" clock pulse.

Address Mark Detection

Address marks are three bytes of missing clock and data pulses preceding a count field. The AM search is initiated by turning on bits 1 and 6 of the EC register (Figure 3-30). SERDES searches for 21 consecutive empty data cells. When such an area is found, ST4 is set to notify the microprogram. No read data is otherwise processed during an AM operation.

Sync Byte Detection

A sync byte (0001 1001) precedes all record fields. This byte is detected by analyzing the bit pattern as it shifts through the shift register (Figure 3-30). When this byte is found, SERDES sets the Data Good FF with the following results:

1. The Bit Ring Counter is reset to zero. It is then allowed to run continuously from 0 to 7 for data deserialization.

2. ST4 sets to signal the microprogram that valid read data is being processed.

The sync byte itself is not transferred to microprogram control. The first byte to be loaded into the SD register for I/O operations is the ID byte, which is also hexadecimal 19.

A special synchronization function is allowed for diagnostic purposes. With bits 2 and 7 of the IC register set, SERDES allows any data "1" bit to set the Data Good FF. This allows the bit ring counter to run and set ST4. The presence of any data on the track can thus be detected by setting IC2 and IC7 and monitoring ST4.

Deserializer

The deserializer (Figure 3-32) converts the serial separated "1" data bits into parallel, or byte, information.

All data is loaded into, and shifted through, the shift register. At the start of the read operation, however, no significant processing occurs. Since Data Good (Figure 3-30) has not set yet, the Bit Ring Counter is held in a reset (no count) state. Once the sync byte has been detected, the counter is released to begin counting continuously from 0 to 7.

Any further bytes being processed through the shift register are gated into FDR at the trailing edge of each Bit Ring count of 7. That byte remains frozen in FDR for one entire byte time. Shortly afterward the byte is automatically transferred by hardware from FDR to the SD register and ST4 is set to signal byte availability. The microprogram must transfer this byte from SD to another register before the next FDR-to-SD transfer or the byte is overwritten by a new byte.

Delayed data is examined to supply an odd parity bit to SD concurrent with the eight data bits in the byte. Parity bits are hardware-generated since they are not written on the track.

Read Errors

Any errors detected during SERDES read operations are reported to the microprogram as Control Checks The errors are gated to the ID Assembler when IB2=0 and IB4-7=5. The following bits apply to read operations:

- ID2 (Read Parity Check) – the parity bit generated for the byte as the byte was deserialized did not result in odd parity for the byte transferred to the SD register.

- ID3 (Bit Ring Check) – while SERDES was active, the number of bits set in the Bit Ring Counter was even (0, 2, 4, 6).

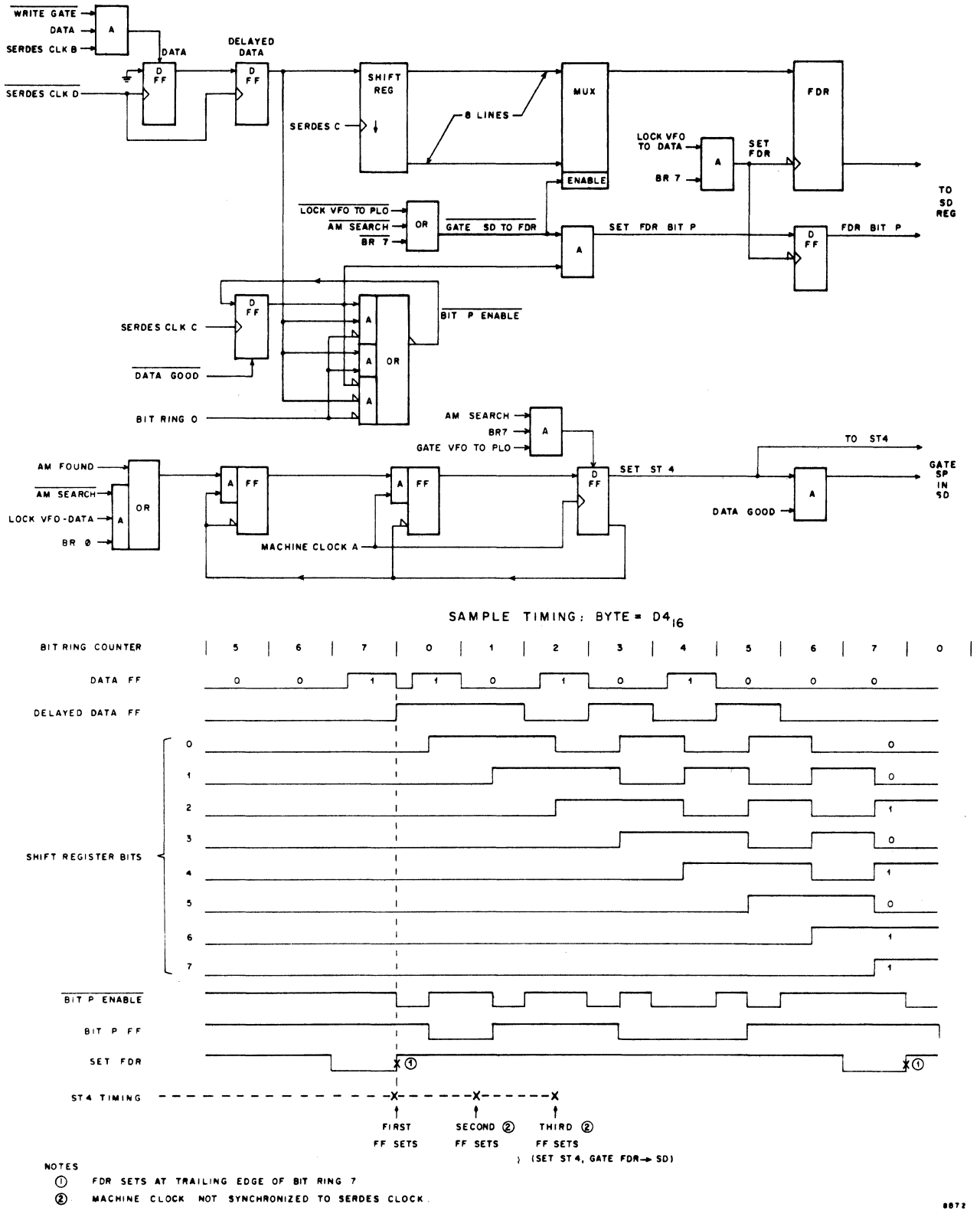


Figure 3-32. Deserializer

READ/WRITE ECC

INTRODUCTION

Read/write error detection and correction are accomplished by use of an error correction code (ECC). This code consists of seven bytes of ECC data appended to the end of every field (Home Address, Count, Key, and Data) during a write operation. During a read operation, the ECC enables detection and correction of errors in the data.

The ECC is a hardware-generated code that detects most errors; it corrects any single error burst 11 bits or less. ECC operations are under microprogram control.

SIMPLIFIED EXAMPLE

ECC uses a 56-bit polynominal generator. In order to understand principles of ECC operation, a simplified 7-bit system is explained. It is for example only. This example uses a field consisting of five data bits and seven ECC bits.

Simplified ECC Write

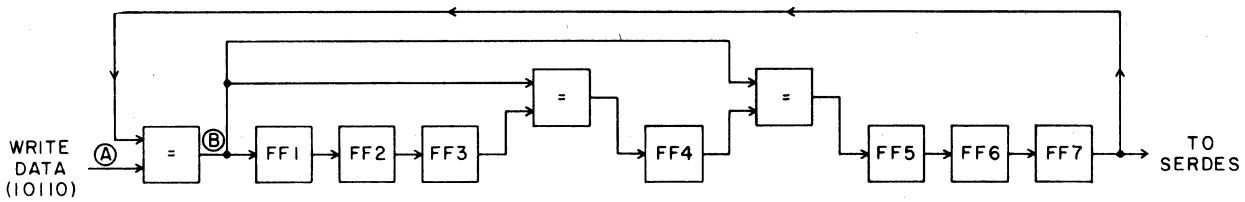
The ECC pattern is generated by a circuit consisting of a seven bit shift register with feedback (Figure 3-33). Prior to the write operation, all flip-flops are reset to zero.

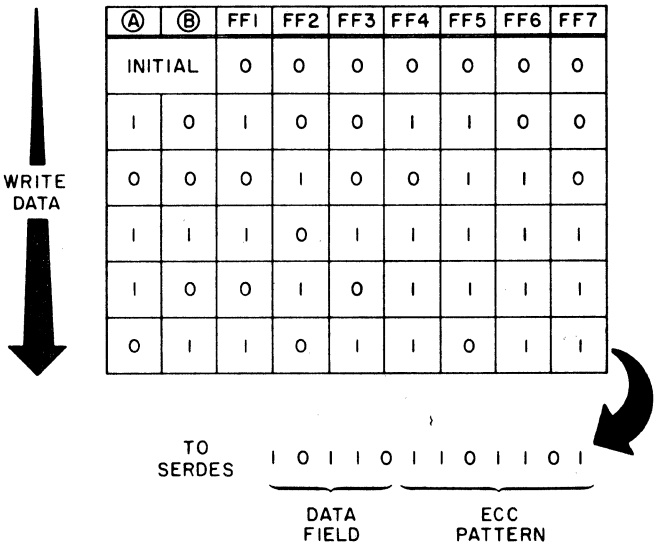
As the data is being written, it is exclusive-ORed with the contents of FF7; the resultant is placed in FF1. A similar operation is performed for data entering FF4 and FF5.

After the data field has been written, the seven flip-flops are connected serially with all feedback and exclusive-ORs removed. The ECC pattern is shifted out of FF7, one bit at a time, and added to the five-bit data field.

Simplified ECC Read

ECC operation during a subsequent read operation uses the same seven flip-flops, except that they are now connected as a three position displacement





Ⓐ	Ⓑ	FF1	FF2	FF3	FF4	FF5	FF6	FF7
INITIAL		0	0	0	0	0	0	0
1	0	1	0	0	1	1	0	0
0	0	0	1	0	0	1	1	0
1	1	1	0	1	1	1	1	1
1	0	0	1	0	1	1	1	1
0	1	1	0	1	1	0	1	1

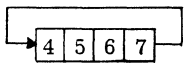
Figure 3-33. Simplified ECC Write

register (with feedback) and a four position pattern register (with feedback). See Figure 3-34.

All 12 of the read bits (5 data plus 7 ECC) are shifted through the flip-flops. If both registers contain zero after the operation, there is no error.

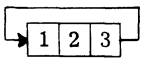
Figure 3-35 illustrates the same circuit with a two-bit read error: data is 10000 rather than 10110. In this case, the error is correctable because both registers are nonzero. If one register is nonzero while the other is zero, the error is uncorrectable. The error is corrected as follows:

1. Shift Pattern register, with feedback, until error bits are in low-order positions



1 1 0 0 Shifts required=0

2. Shift Displacement register, with feedback, until its pattern matches Pattern register



Initial: 1 0 1
Shift: 1 1 0 Shifts required=1

3. Compute Displacement (D)

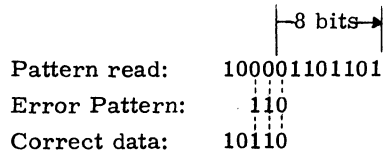
D=P -(4Y+9Z)

P=Maximum correction data length (12)
Y=Number of Displacement register shifts
Z=Number of Pattern register shifts

If (4Y + 9Z) is greater than 12, divide by 12 and use remainder.

D=12- [(4x1) + (9x0)]
=12 - 4
=8 bits

4. Counting from end of ECC pattern, find first erroneous bit by counting off number of computed displacement bits. Invert data bits corresponding to "1" bits in error pattern.



ECC WRITE

The controller generates the 56 bit (7 byte) ECC code with four polynominal generators: P1, P2, and P3. See Figure 3-36. The generators are connected in series with feedback to form one large polynominal generator. As serial data leaves SERDES to be written on a disk pack, the data is shifted through the generator to form an ECC checkword. At the end of the data write operation, this ECC checkword (seven bytes long) is serially shifted out of the generator without feedback. The seven ECC bytes are appended to the end of the data with PO bit 21 being the first bit written.

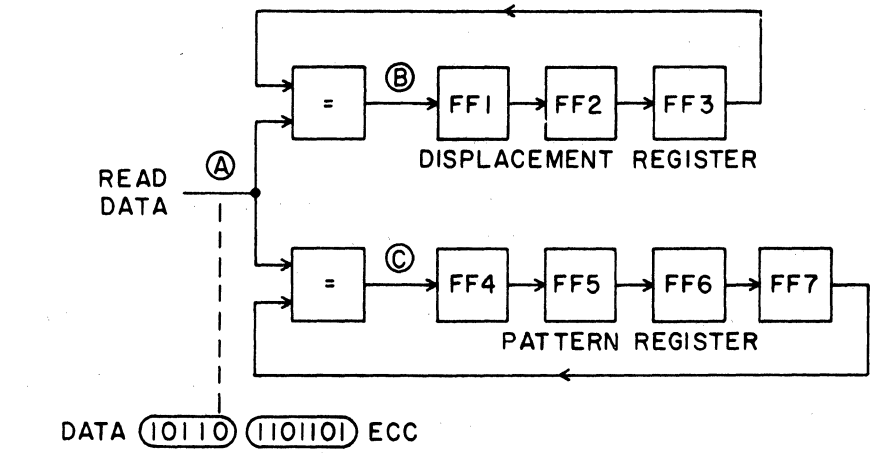
ECC READ

The four polynominal generators are connected in parallel during a read operation as shown in Figure 3-37. Starting from the sync byte, all of the data and ECC bytes are shifted through the generators with feedback. If all generators are equal to zero at the end of the operation, there were no errors.

On the other hand, any nonzero pattern indicates an error. The microprogram recovers from the error by using P0 as the pattern register (as shown in Figure 3-34) with P1, P2, and P3 serving as the displacement register. ECC status is monitored by checking the byte configurations of the contents of the SB, SC, and ID registers as listed in Table 3-11.

Error Correction

To be correctable, the error pattern in P0 must be a maximum of 11 bits. In addition, all four generators must contain bits and these bits shall have a configuration such that they can be shifted to match the P0 pattern. The microprogram error correction sequence is as follows:



PATTERN CHECK

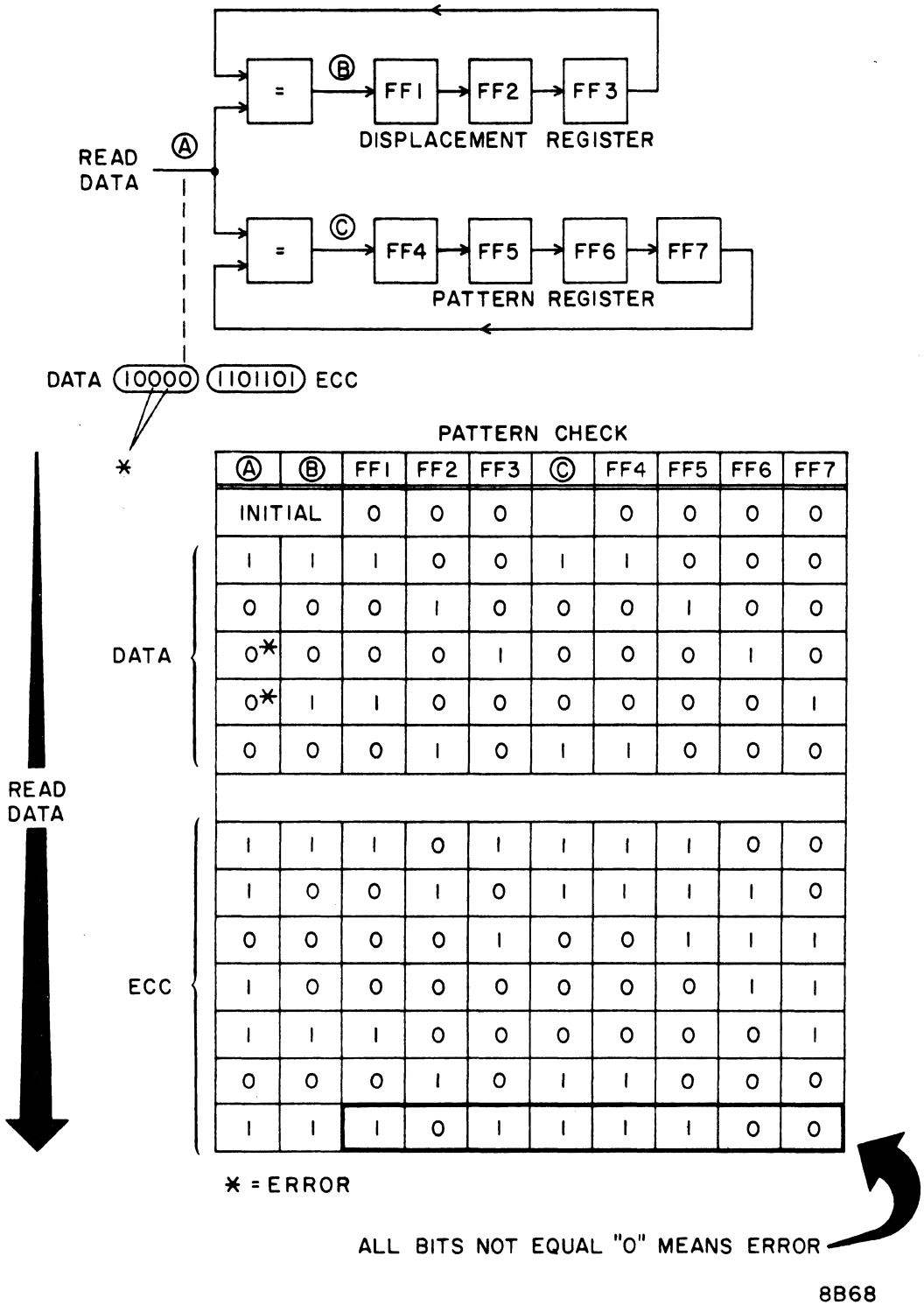
Ⓐ	Ⓑ	FF1	FF2	FF3	Ⓒ	FF4	FF5	FF6	FF7	
INITIAL		0	0	0		0	0	0	0	
1	1	1	0	0	1	1	0	0	0	DATA
0	0	0	1	0	0	0	1	0	0	
1	1	1	0	1	1	1	0	1	0	
1	0	0	1	0	1	1	1	0	1	
0	0	0	0	1	1	1	1	1	0	
1	0	0	0	0	1	1	1	1	1	ECC
1	1	1	0	0	0	0	1	1	1	
0	0	0	1	0	1	1	0	1	1	
1	1	1	0	1	0	0	1	0	1	
1	0	0	1	0	0	0	0	1	0	
0	0	0	0	1	0	0	0	0	1	
1	0	0	0	0	0	0	0	0	0	

READ DATA

ALL "0" MEANS NO ERROR

8B67

Figure 3-34. Simplified ECC Read (No Error)



Ⓐ	Ⓑ	FF1	FF2	FF3	Ⓒ	FF4	FF5	FF6	FF7
INITIAL		0	0	0		0	0	0	0
1	1	1	0	0	1	1	0	0	0
0	0	0	1	0	0	0	1	0	0
0*	0	0	0	1	0	0	0	1	0
0*	1	1	0	0	0	0	0	0	1
0	0	0	1	0	1	1	0	0	0
1	1	1	0	1	1	1	1	0	0
1	0	0	1	0	1	1	1	1	0
0	0	0	0	1	0	0	1	1	1
1	0	0	0	0	0	0	0	1	1
1	1	1	0	0	0	0	0	0	1
0	0	0	1	0	1	1	0	0	0
1	1	1	0	1	1	1	1	0	0

Figure 3-35. Simplified ECC Read (Error)

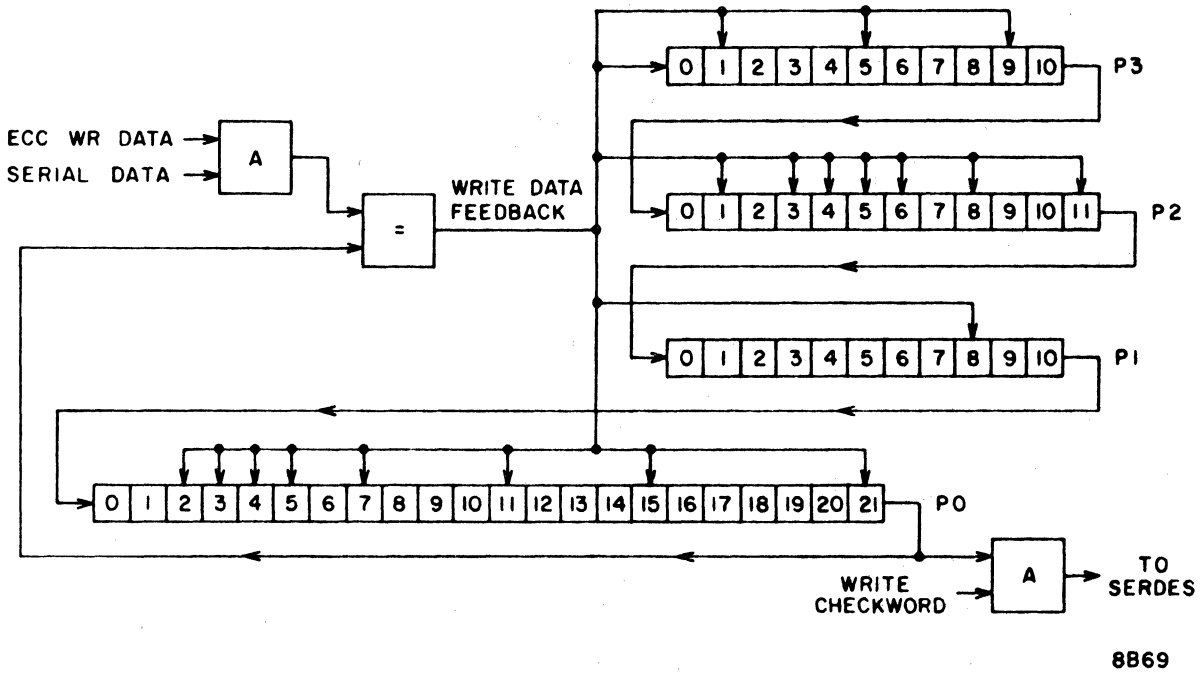


Figure 3-36. ECC Write

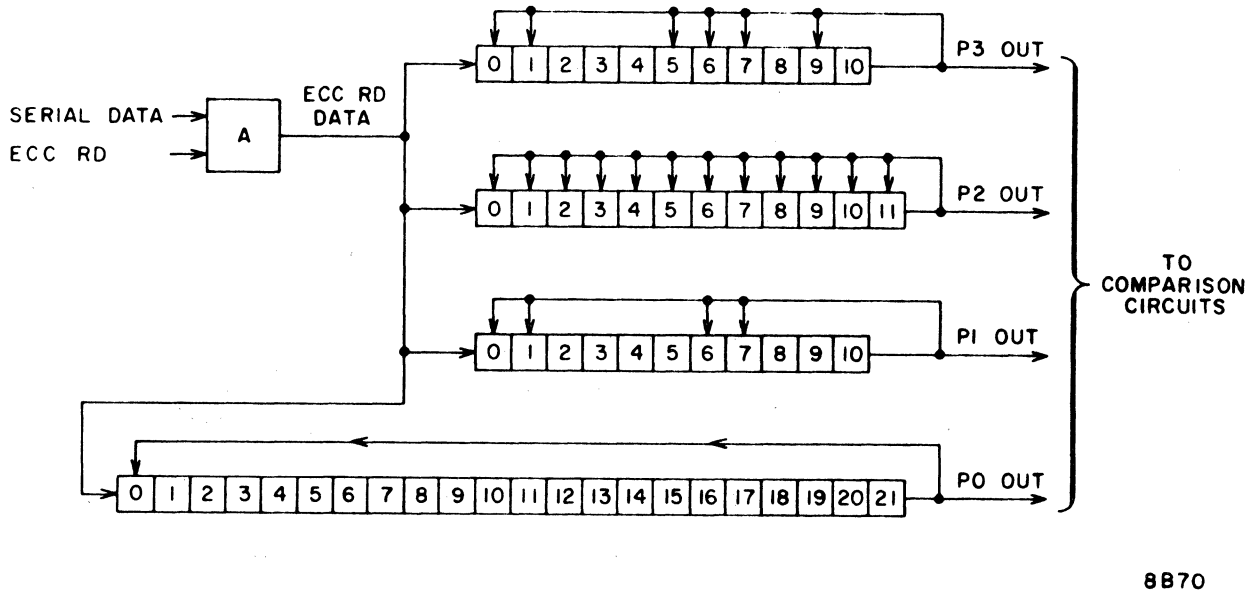


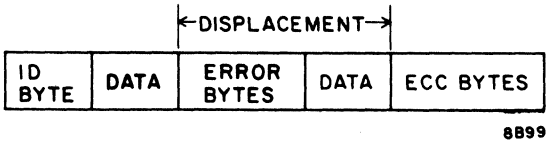
Figure 3-37. ECC Read

TABLE 3-11. ECC STATUS

Register	Bit	Function
SB	0	P0 low not equal to P3 or P0 parity not equal to P3 parity
	1	P0 low not equal to P3
	2	P0 low not equal to P2 or P0 parity not equal to P2 parity
	3	P0 low not equal to P2
	4	P0 low not equal to P1 or P0 parity not equal to P1 parity
	5	P0 low not equal to P1
	6	P0 low not equal to zero
	7	P0 high not equal to zero
SC	0	P0 bit 18
	1	P0 bit 17
	2	P0 bit 16
	3	P0 bit 15
	4	P0 bit 14
	5	P0 bit 13
	6	P0 bit 12
	7	P0 bit 11
ID	0	No Input Data Received. ECC hardware failed to detect any data transfer during read or write operation.
	1	P0 or Write. During read operation, P0 parity error sets this bit; P1, P2, and P3 errors set this bit in conjunction with bits 2 and 3. During write operation, parity error was detected in one of the ECC registers connected in series.
	2	P1 or P3. P1 or P3 parity error during read.
	3	P2. P2 parity error during read.

NOTES:
1. Data is gated to SB and SC during Allow Decode Read or Allow Decode Write operations.
2. Errors gated to ID Assembler are Control Checks reported when IB2=0 and IB4-7=0110.

1. Using SPEX 23, P0 is shifted, one bit at a time while counting shifts, until all error bits are contained within P0 bits 0 through 10. This is the error pattern, where each "1" indicates a read data bit error.
2. SPEX 24 is raised to enable shift of P1, P2, and P3. They are simultaneously shifted, one bit at a time while counting shifts, until any one of them contains a pattern that matches the P0 pattern. The microprogram stores the number of shifts required.
3. Step 2 is repeated to store the shifts required for the other two registers to also match the P0 pattern.
4. The microprogram calculates the displacement in bytes from the end of the data bytes (not the ECC bytes) to the first byte in error.



5. During a sense operation, the microprogram supplies the following data to the channel:
   - Correctable ECC Error
   - Syndrome bits (P0)
   - Displacement in bytes
6. The error pattern is displaced under CPU program control to match the read data and is exclusive-ORed to invert the erroneous bits.

If the displacement is zero, the error is in the ECC bytes. A displacement of one or two indicates that the error may be partially in the ECC bytes.

Uncorrectable Errors

The following errors are uncorrectable:

1. If any generator is zero while any other generator is nonzero.
2. P0 is shifted until all error bits are in the leftmost position (bit 21 feeds back to bit 0). If all of the error bits cannot be contained within the 11 leftmost bit positions, the error is uncorrectable.
3. If P1, P2, or P3 cannot be shifted to match the P0 error pattern.
4. If the displacement is larger than the data field; for example, if the data field is 90 bytes (720 bits) while the displacement is 100 bytes (800 bits).
5. If more than 89 shifts are required in any register.

ECC CONTROL

ECC hardware is controlled by bits 0, 1, and 2 of the IE register. The control configurations are listed in Table 3-12.

TABLE 3-12. ECC CONTROL

IE Bits			Name	Function
0	1	2		
0	0	0	Reset	Clears all ECC registers.
0	0	1	Read	Enables read feedback. Gates serial read data to all registers.
0	1	0	Write	Connects ECC registers in series and enables write feedback. Gates serial write data to lowest order ECC bit.
0	1	1	Write Checkword	Connects registers in series and enables feedback to lowest order bit. Appends the generated checkword to the field just written.
1	0	1	Allow Decode Read	Enables read feedback to all ECC registers. Allows P0 to be shifted one bit by SPEX 23. Allows P1, P2, and P3 to be simultaneously shifted one bit by SPEX 24. Allows the SB and SC registers to be loaded by the ECC hardware.
1	1	1	Allow Decode	Connects registers in series and enables feedback to lowest order bit. Allows SPEX 23 to shift all registers one bit. Allows SPEX 24 to shift P1, P2, and P3 one bit. Allows the SB and SC registers to be loaded by the ECC hardware.

MACHINE TIMING

CLOCK

The clock circuit consists of a 25 MHz crystal oscillator driving a 10-latch ring counter. The counter period, which is also one machine cycle, is 200 nanoseconds.

The counter has 10 clock outputs, clocks A through K (clock I does not exist). Each clock is 40 nanoseconds wide and overlaps the previous pulse and following pulse by 20 nanoseconds. For example, clock B occurs starting 20 nanoseconds from the rising edge of clock A and falls 20 nanoseconds after the rise of clock C.

If a Machine Stop condition occurs, the counter stops with clocks J and K both active.

INTRODUCTION TIMING

Figure 3-38 illustrates machine timing while executing five microwords. Figure 3-39 shows the major logic elements involved in a normal execution sequence. The microwords selected by this example are used to load the parameter byte from the maintenance panel switches into the in-line diagnostic routine.

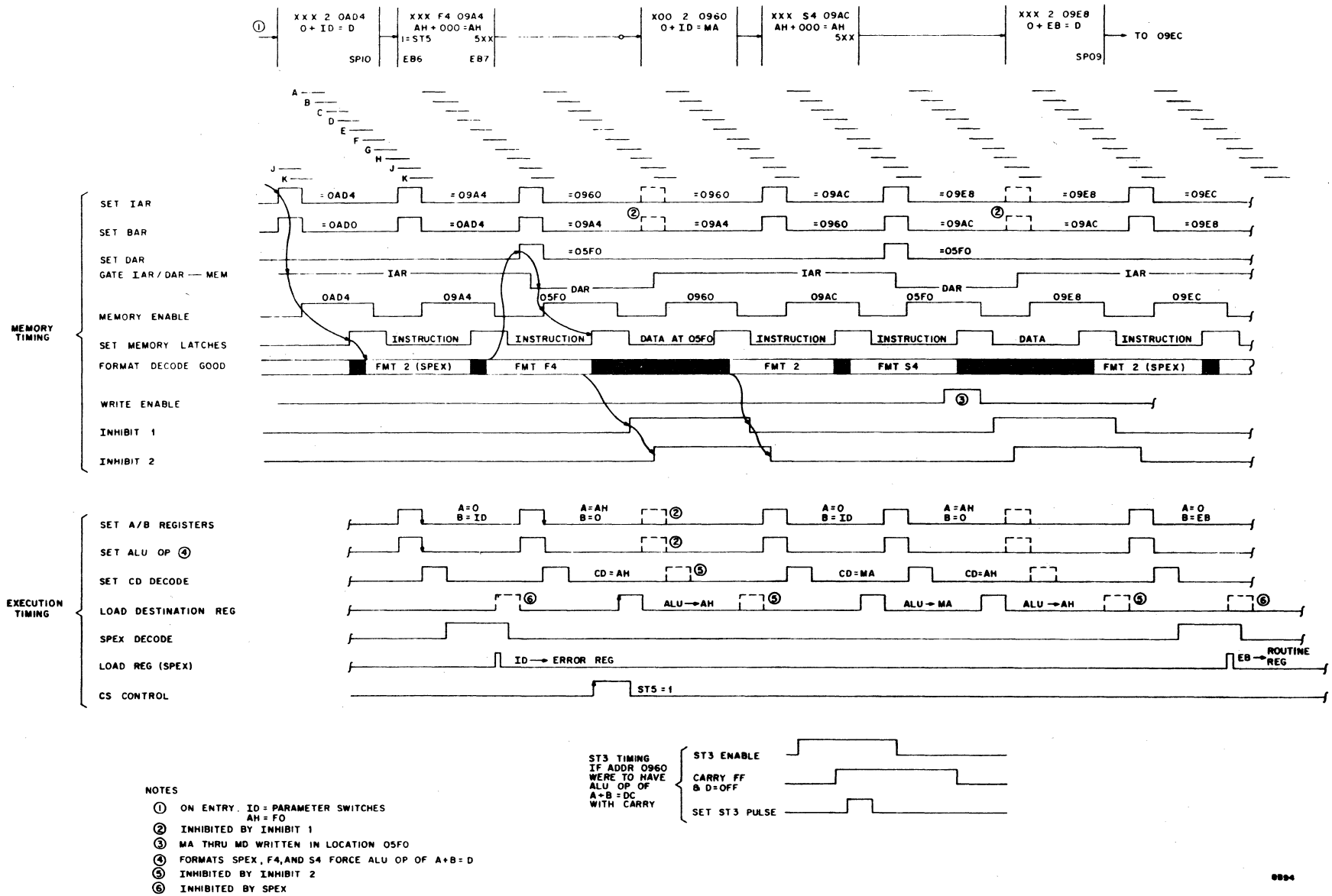


Figure 3-38. Machine Timing

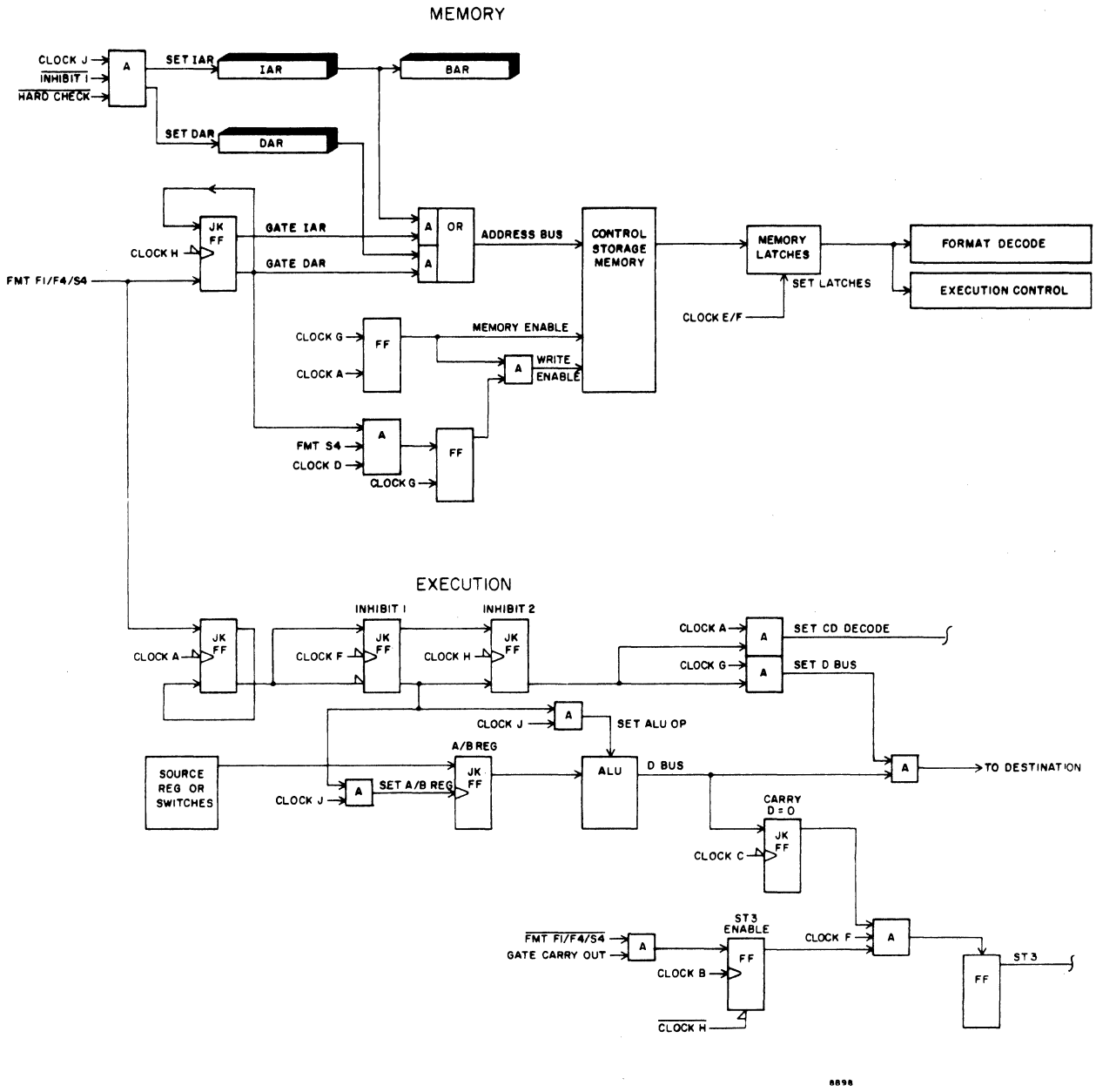


Figure 3-39. Simplified Execution Logic

Each microword may be considered as consisting of two semi-independent operations: memory accessing and instruction execution. They are not wholly independent inasmuch as the results of a computation can affect the address of a word to be executed. In this example, however, all branching conditions were initialized prior to entry into this sequence. Refer to the microprogram discussion for an explanation of the interaction of addresses versus instruction execution.

There are two types of machine cycles, instruction and data.

Instruction Cycle

An instruction cycle performs the following:

1. The address to be executed is gated into IAR. At the same time, the old IAR address is transferred to BAR.
2. The contents of IAR, functioning as the Address Bus, are gated to the control storage memory.
3. Memory Enable is raised to read the 4-byte instruction from memory.
4. Memory ECC corrects the pattern, if necessary, and the memory latches set or clear as applicable.
5. The outputs of the memory latches are applied to the format control logic. This completes the memory accessing function of the cycle.
6. Execution procedes:
   - The format is decoded to determine if the next machine cycle will be in instruction cycle or a data cycle. Data cycle will be enabled if this word is a fetch (format F1 or F4) or store (format S4) operation.
   - The data sources to the A register and B register are determined. The sources (a register or a constant) are available on the A Bus and B Bus as soon as the word is decoded.
   - The type of ALU operation to be used is determined.
   - The CD or SPEX decode is determined.
   - The arithmetic operation is completed and the result is stored in the location specified by the CD decode (if enabled).
   - Auxiliary operations, such as a SPEX function or ST register manipulation, are completed.

Data Cycle

Data cycles are used in the second machine cycle of a fetch or store operation. To understand a 4-byte fetch operation, consider address 09A4 in Figure 3-38:

1. As soon as format F4 is decoded, DAR is enabled and serves as the Address Bus data source to memory for the next machine cycle.
2. The instruction half of the word (AH+000= AH) is executed during the first machine cycle.
3. The four characters in DAR are set to 05F0:
   - Character 0=0 because it is unchanged from the last fetch or store operation.
   - Character 1=5 because the CW field of the memory word at address 09A4 is a 5.
   - Characters 2 and 3=F0 because the contents of the A bus are gated to DAR bits 8 to 15 by an F4 instruction. The A bus contains the contents of the AH register which, in this example, equals F0.

4. The contents of memory location 05F0 are loaded into the MA, MB, MC, and MD registers.

Note that the ALU operation is executed only once; execution is during the instruction half of the operation. The two Inhibit flip-flops are set for one machine cycle for two reasons. One is to prevent an ALU operation during the data cycle and the other is to prevent IAR from changing during the data cycle.

A format S4 operation is demonstrated by the word at address 09AC (Figure 3-38). DAR/memory gating is identical with the fetch operation, except that the data will be stored at 05F0 rather than fetched from 05F0. The address is selected in the same manner as that determined by step 3 of the F4 word explanation. If address 09A4 had changed the contents of the AH register, the storage address would have changed accordingly. But, since AH is still at the same value, DAR will again be set to 05F0. When Write Enable is generated, the four bytes in MA through MD are loaded at address 05F0, destroying the old data already there.

In actual operation, this sequence (addresses 0AD4, 09A4, 096X, and 09AC) is executed four times. Address 096X becomes 0964 on the second execution, and is incremented on each subsequent execution. Each time the sequence is executed, new data from the DATA/ROUTINE/PARAMETER switches is loaded into one of the memory registers. Address 0960 loads the MA register on the first execution. On the second execution, address 0964 (not shown on Figure 3-38) loads data into the MB register. At this point the: MA register contains the data from the first execution, MB register contains data from the second, and MC and MD contain data previously loaded. The sequence continues until, on the final execution, the MD register is loaded and the word stored at 05F0 contains four bytes of new parameter data loaded from the switches.

MACHINE STOPS

The controller normally continuously executes instructions. Even if the unit is not involved in an I/O operation, the wait loop microwords (addresses 1300, 1328, 13A4, 1304, and 13C4) are being executed. To stop execution, the Inhibit flip-flops must be set. This state is attained when one of the following conditions exists (see Figure 3-40):

- Machine Stop
- Machine Reset
- CE Inhibit
- Fetch or Store operation
- Initial Microprogram Load operation

Machine Stop

Machine Stop occurs if a hardware error is detected or if initiated by the microprogram. Hardware errors are of two types: Hard Check and Control Check. These checks are shown in Figure 3-40.

Hard Check indicates that a catastrophic hardware failure occurred, that is, all data processing is totally and irretrievably unreliable. All execution stops immediately. For maintenance purposes only, the OPERATION switch on the maintenance panel may be set to the RUN ON ERROR position. The error is posted in the ERROR DISPLAY indicators on the maintenance panel, but execution continues. Never leave the switch in this position unless all machine operations are closely monitored. One error cannot be bypassed; Clock Error indicates a fault in the clock ring counter.

Hard Check automatically transfers the contents of BAR into FAR to indicate the address at the time of failure. FAR remains unchanged until the next Hard Check occurs.

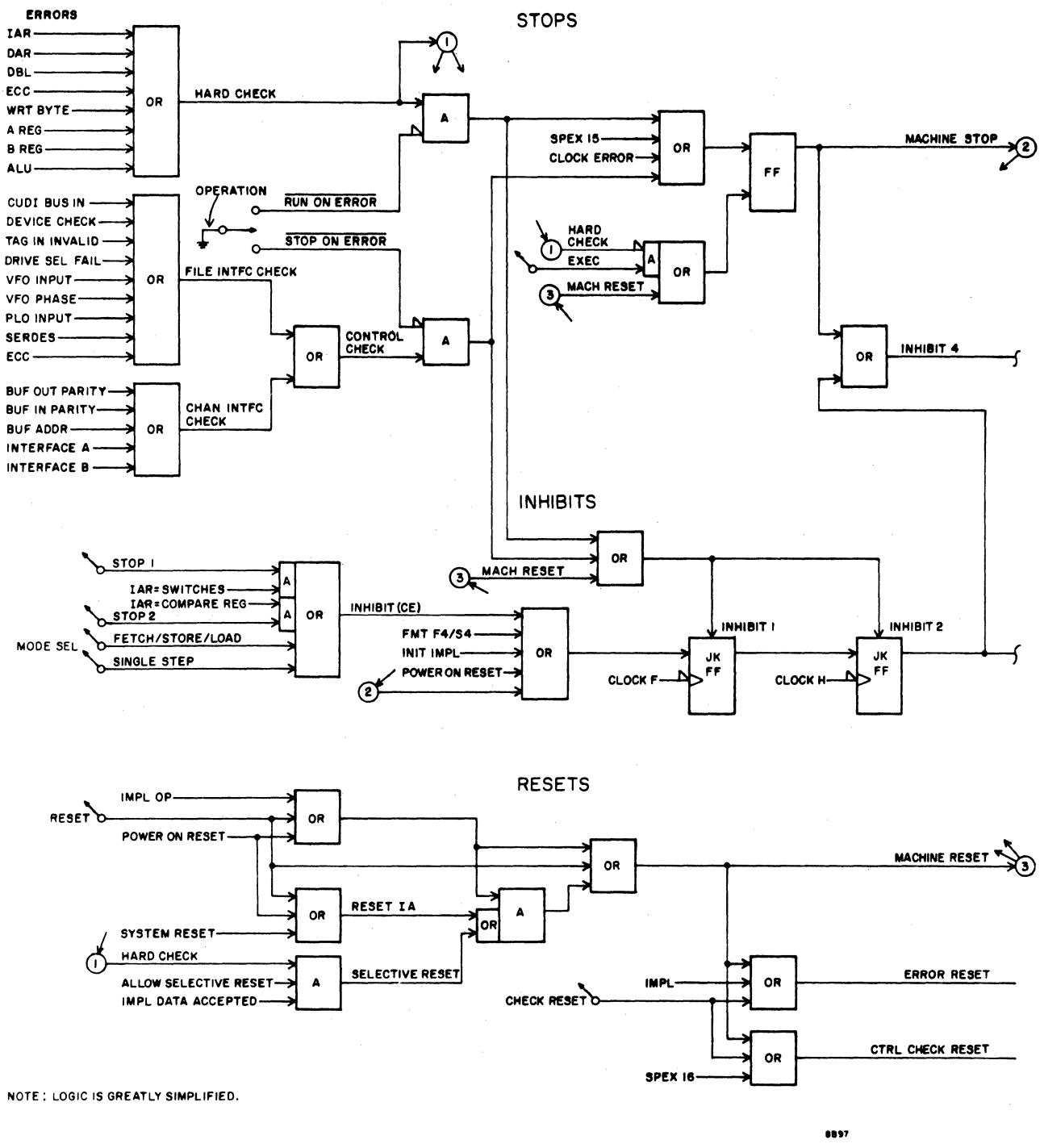


Figure 3-40. Machine Stops and Resets

Control Check indicates that an error condition exists in either the drive/controller or channel/controller interface. These errors are normally monitored by the microprogram since they do not have a direct effect on basic controller operation. If desired, any of these errors can stop the unit for maintenance purposes if the OPERATION switch on the maintenance panel is set to the STOP ON ERROR position.

The microprogram can stop the unit by a SPEX 15 word while executing the hardcore diagnostics. The unit stops with the error-detecting word available for display from BAR. The subsystem Trouble-shooting Manual explains these error stops along with the probable causes and recommended corrective action.

Resets

Five different types of resets are provided to clear machine status and/or error conditions. Figure 3-40 shows the interaction of the resets gating. Refer to Table 3-13 for a list of the effects of these resets.

CE Inhibit

CE inhibit occurs if manual intervention is required. The following conditions generate this signal if in CE mode:

1. If the COMPARE STOP 1 switch is up and the contents of IAR match the CONTROL STORAGE ADDRESS switches.
2. If COMPARE STOP 2 is up and the contents of IAR match the contents of the Compare register. The Compare register must be manually loaded before selecting this function.
3. The MODE SELECT switch is in any position other than RUN or RECYC when EXEC is actuated, that is, data or addresses are being loaded, stored, or displayed.
4. OPERATION switch is set to STOP/SINGLE STEP.

Fetch or Store Operation

A microword that fetches data (format F1 or F4) or stores data (format S4) sets Inhibit for one machine cycle. Refer to the Data Cycle discussion for additional information.

IMPL Operation

Most normal machine operations are inhibited during an Initial Microprogram Load operation. Refer to Microprogram Load theory for additional information.

TABLE 3-13. CONTROLLER RESETS

Reset Type	Circuit Reset	Other Functions
Power On	Switch to Chan A/B SERDES counters	Starts Initial Microprogram Load Drops Controller Out Power (COP) Drops Device In Power (DIP)
Machine	A/B Registers Carry, D=0 SPEX Enable IAR/DAR/BAR Channel Block Timer Block Switch to Chan A/B Channel Read/Write ECC Enable CUDI Checks	Disable memory Stop IMPL in progress Execution starts at address 0000
Reset IA	IA/IC Registers IAR bits 8 thru 13 DAR bits 8 thru 15	

TABLE 3-13. CONTROLLER RESETS (Cont'd)

Reset Type	Circuit Reset	Other Functions
Selective	IAR bits 8 thru 12	Turn on IAR 13
Control Check	All Control Check Errors Overrun (channel interface)	
Error	All Hard Check Errors Routine Register Error Register Compare Register Read Error Pattern Register	

PART 2

MICROPROGRAM THEORY

INTRODUCTION

Microprogram theory explains the basics of what the microprogram is, and how it relates to the hardware to accomplish machine functions. The theory is divided into four areas as follows:

- General Microprogram – Explains the basic functioning of the microprogram. It provides the basic information necessary to understand the material provided in later sections.
- Instruction Word – Explains the basic instruction word, word format, and fields. It also provides a description of word execution.
- Core Map – Explains the layout and use of the core map contained in the Microprogram Manual.
- Flow Diagrams – Explains how to read and interpret the Microprogram Manual flow diagrams which document all program routines.

GENERAL MICROPROGRAM

The microprogram is a set of 40 bit instruction words used to translate channel commands into drive instructions, and control the transfer of data between these units. The microprogram is stored on the cassette tape and read into controller memory either when the unit is first powered up or when commanded from the maintenance panel. Each time the memory is loaded the mainline hardcore diagnostics are executed. This checks the controller hardware and assures the user that the machine is functioning properly. Once the first record is loaded from the tape into memory, the microprogram takes over control of machine functions.

The microword is the basic unit of the microprogram. The 40 bit word contains 32 instruction bits and eight Error Correction Code (ECC) bits. For purposes of this discussion, the eight ECC bits are ignored.

The 32 instruction bits (4 bytes) are arranged in various fields. A field is a bit or group of bits which, when interpreted together, have a specific meaning. Field assignments within a word are the function of word format. This machine uses a variable format concept to allow maximum programming flexability. The microprogrammer establishes word format depending on the function to be accomplished. Fields and formats are explained in detail in the section on the instruction word.

ADDRESSING

In addition to the flexability allowed by variable formatting, this microprogram also provides nonsequential addressing. This means that each instruction generates the address of the next instruction to be executed. In so doing, the program is continually making decisions about what to do next, based on the results of what it has just accomplished. The program may be considered self-steering. This is possible because of the way the microprogram is mechanized. Any given instruction has a minimum of four possible next addresses to which it could branch. By combining some of the special operations available in the program, it is possible to construct one instruction which allows a 4,000 way branch condition. It is also possible to construct an instruction which prohibits branching. In such a case, the programmer arbitrarily sets the next address without regard to what has just occurred.

MEMORY ORGANIZATION

To fully understand the various instruction formats and their uses, memory addressing must be understood. Four hexadecimal characters are used to represent the location of a particular instruction or data word in memory. The four characters represent the divisions of memory: quadrant, sector, subsector, and word. Address 0E34 locates one specific 32 bit word in quadrant zero, sector E, subsector 3, and word 4.

The 32 bits are stored (four to a board) on the eight memory boards A1BB through A1BJ. Each board contains four rows of memory chips, with 16 chips in each row (see Figure 3-41). Each of the 32 chip rows (8 boards x 4 rows) provide storage for one memory bit. Since memory is divided into four quadrants (0 through 3) the chip row is divided into four quadrants, with four chips per quadrant. Each quadrant is then divided into 16 sectors (0 through F). One chip contains one-fourth of the quadrant, or four sectors.

Figure 3-42 illustrates how the entire memory is mapped onto the ten memory boards. Each board contains four specific memory bits. The quadrant and sector divisions are shown along the left-hand margin, card locations are shown across the top, and the specific bit locations are shown (at the bottom) relative to each card and chip location. The illustration also shows that each sector is further divided into 16 subsectors (0 through F), and that each subsector is divided into four words (0, 4, 8, and C). Address 0E34, for example, is stored across all ten boards, in chip locations A3, B3, C3, and D3 on each board. Memory bit zero of this address is stored in chip A3 on the A1BJ card. This specific chip also stores memory bit zero for all addresses from 0C00 through 0FFC. Likewise, chip D8 at card location A1BD contains memory bit 29 of all addresses from 2000 through 23FC.

In order to understand the word addressing system, refer to Figure 3-43. One memory subsector is divided into 16 bytes, but addressing is done on a four byte boundary basis. Each word is made up of four 8-bit bytes. The 32-bit word is layed out in relation to the four memory registers (MA, MB, MC, and MD) through which all information enters and leaves memory. Each byte of the four words stored in a memory subsector, has a byte number (0 through F). The first word in a subsector is comprised of bytes 0 through 3, the second word is bytes 4 through 7, and so forth. It can then be seen that the complete address of a specific word, is made up of the quadrant, sector, and subsector locations, plus the byte number of the first byte of the word within the subsector. Therefore, all word addresses must end with either the character 0, 4, 8, or C.

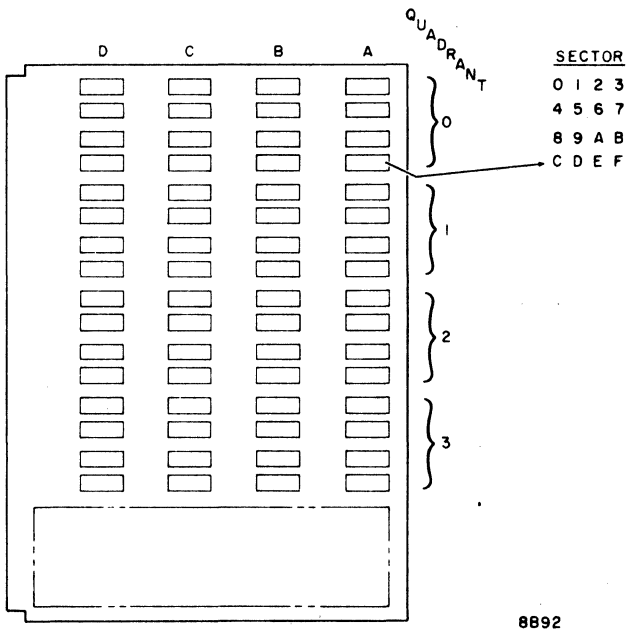
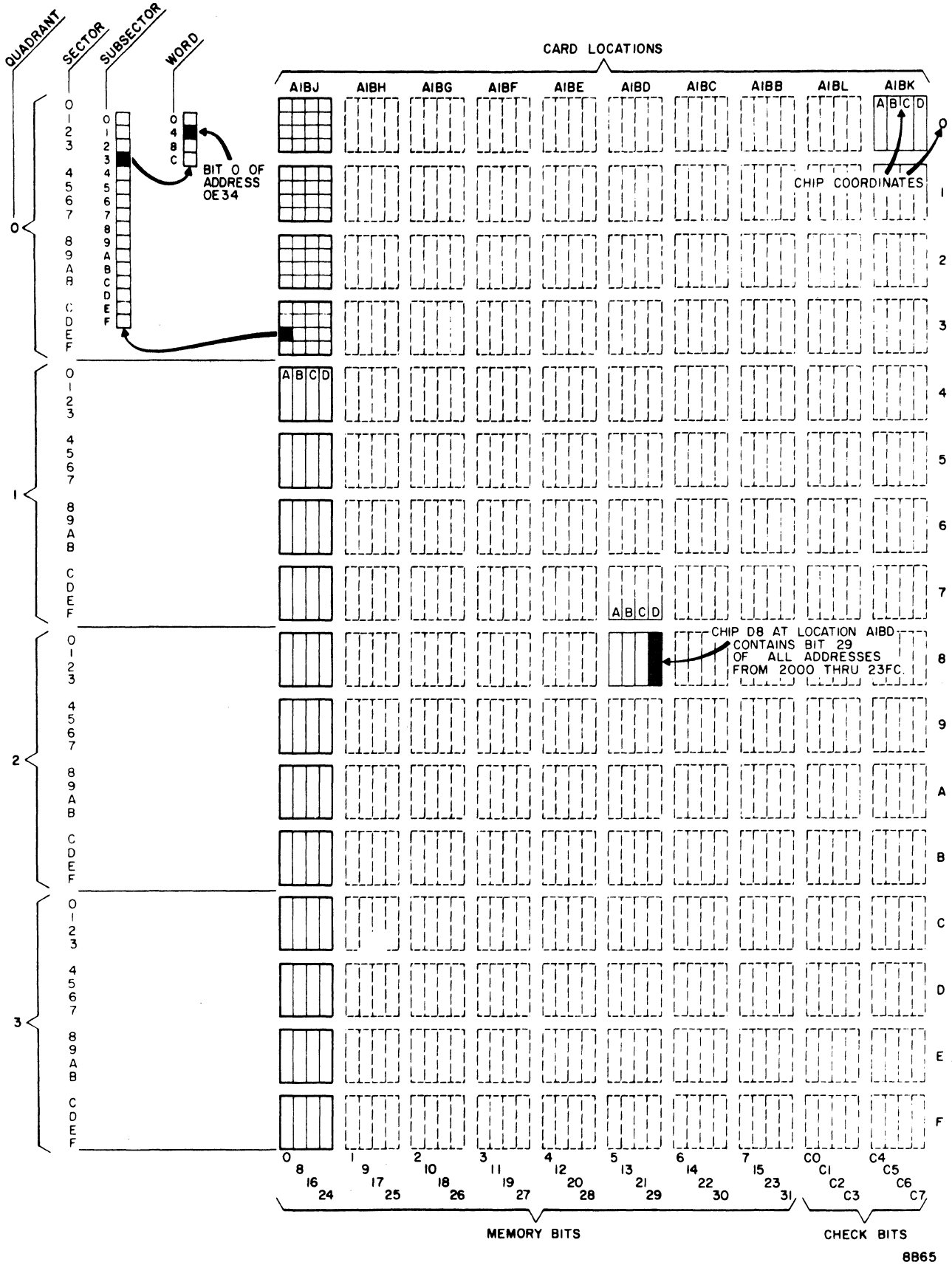


Figure 3-41. Circuit Board Memory Layout



3-42. Memory Organization On Circuit Boards

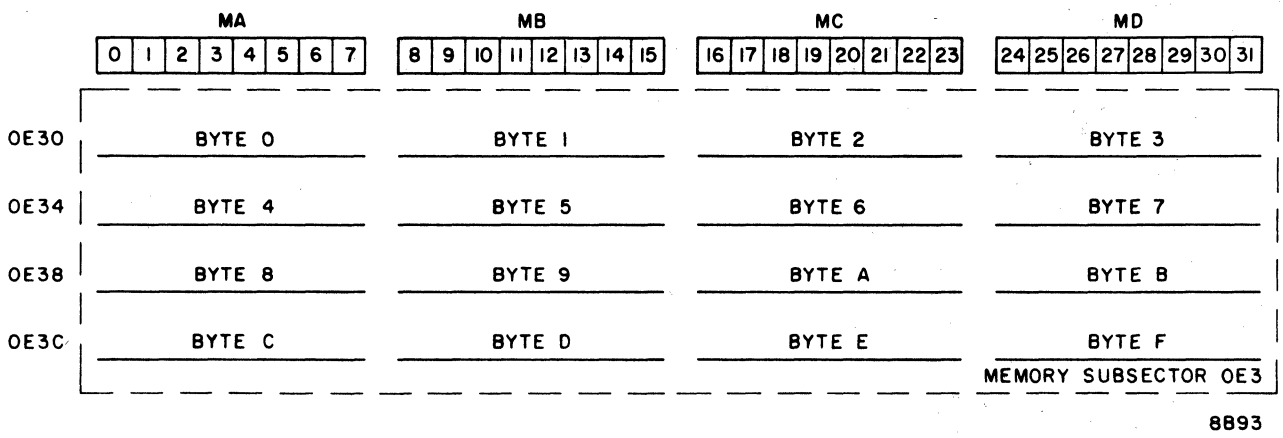


Figure 3-43. Memory Subsector

INSTRUCTION WORD

This material explains the various word formats, fields, and the execution of each of the individual instruction formats.

FORMAT

Basically there are two types of word formats: standard instructions, and special case formats. The standard instructions are identified as formats 1 through 5, F1 (one byte fetch), F4 (four byte fetch), and S4 (four byte store). These formats are recognized, during the instruction decode, by the bit configuration of bits 28 through 31. The special case formats are variations which use part of the standard format, but alter other areas to achieve some particular function. The special case formats are identified as SPEX (special execution), A=IAR, and SPEX 05 + A=IAR. The special case formats are identified, during the instruction decode, by the bit configuration of bits 28 through 31 and a specific code in one or more of the fields. In addition to the standard and special case formats, there is one non-executable instruction, identified as format D (data statement). Format D is used to initialize memory with various constants which are used during program execution.

FIELD

A field is a bit, or group of bits, which when interpreted together, have a unique meaning. Each format (except format D) specifies an arrangement of fields for both the instruction word, and the IAR/DAR. Figure 3-44 illustrates the field arrangements for all the formats. Each field is used to accomplish some part of the instruction word's function. Some fields are used in the generation of the next address, some are used to specify the ALU operation, and some are used to set status information. Table 3-14 explains the functions of each of the fields. Information contained in the bit positions of an instruction word, which comprise a field, is either gated to its intended destination or decoded to interpret its meaning. Table 3-14 also explains whether the field is decoded or used directly. If the field contains coded information, refer to the decode assignment chard provided in Table 3-15. When a SPEX format is decoded it contains a SPEX field which specifies one of 32 possible special execute operations. Table 3-16 defines all the special execute operations.

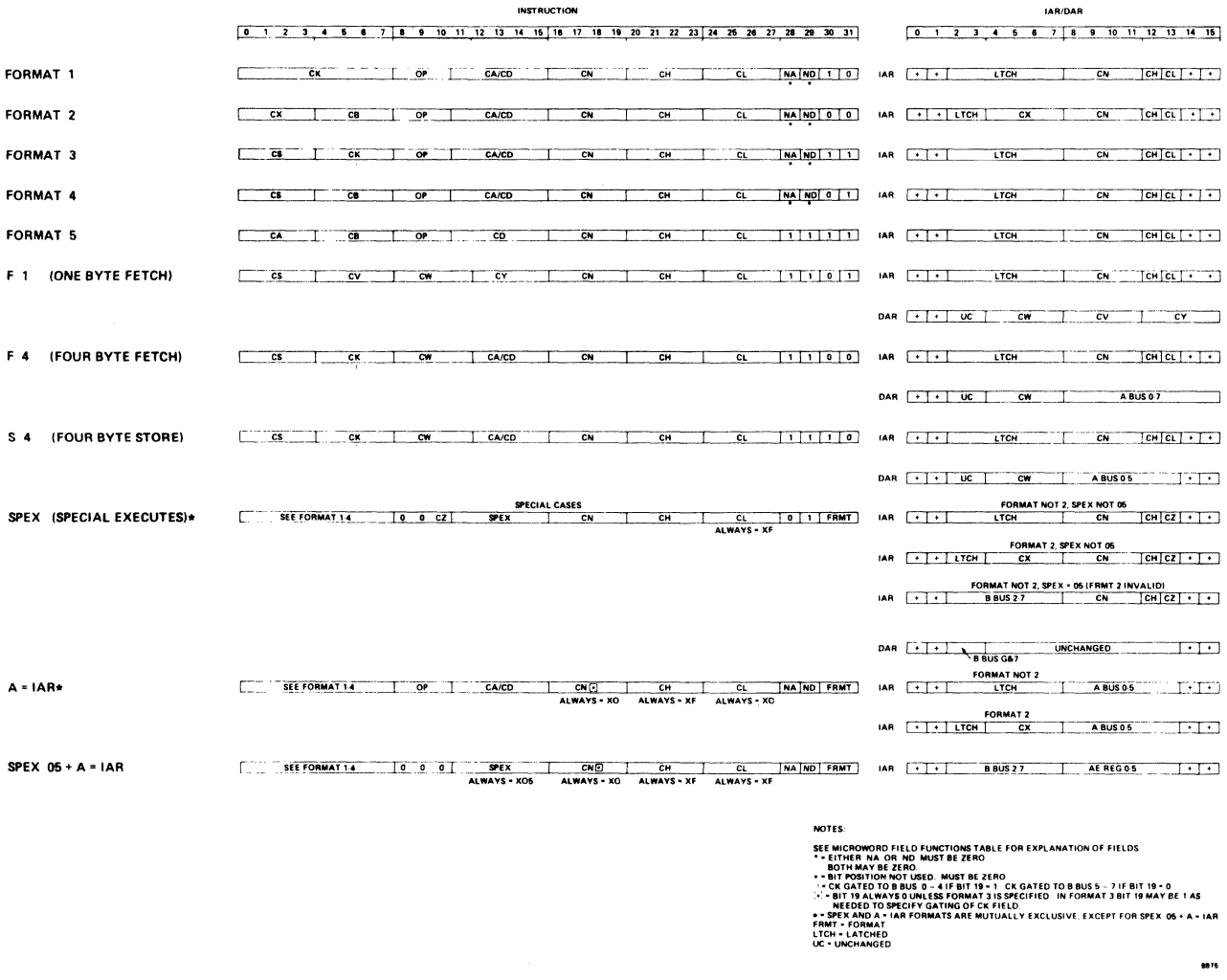


Figure 3-44. Instruction Word Formats

TABLE 3-14. MICROWORD FIELD FUNCTIONS

Field	Function
A Bus X-X	Specifies the source of the second and third hex characters of IAR or DAR. The content of the specified A Bus positions (prior to the ALU operation) are gated to the indicated positions of either IAR or DAR.
B Bus X-X	Specifies the source of the first or the first and second hex characters of IAR or DAR. The content of the specified B Bus positions (prior to the ALU operation) are gated to the indicated positions of either IAR or DAR.
CA	The decode of this field specifies which register is to be gated onto the A Bus. This field is used when the A Bus source and the D Bus destination are different.
CA/CD	The decode of this field specifies which register is to be used, both as the A Bus source and the D Bus destination.
CB	The decode of this field specifies which register is to be gated onto the B Bus.
CD	The decode of this field specifies the register into which the D Bus is to be gated. This field is used when the A Bus source and the D Bus destination are different. It is decoded by looking in the CA/CD column of the decode assignment table. It should be noted, that since this is only a four bit field, a number greater than hex F can not be used.
CH	The decode of this field specifies a particular branch condition to be tested to determine the high order bit on the next address. The specified condition is tested, and if true, next address bit 12 is set. If the condition is not true, next address bit 12 is reset.
CK	This is either a four or an eight bit field which specifies a constant to be loaded onto the B Bus. When the eight bit constant is used (in format 1), the entire field is gated directly to the B Bus. When a four bit field is used (as in formats 3, F4, and S4), bit 19 of the current instruction word determines whether the constant is placed on the high or low half of the B Bus. If bit 19 of the current instruction is 1, the constant is gated to B Bus bits 0 through 3 (high half), and bits 4 through 7 are set to zero. If bit 19 is 0, the constant is gated to B Bus bits 4 through 7, and bits 0 through 3 are set to zero.
CL	The decode of this field specifies the branch condition to be tested to determine the low order bit of the next address. The specified condition is tested, and if true, next address bit 13 is set. If the condition is not true, next address bit 13 is reset.
CN	This field is not decoded on the assignment table. It is used to specify the subsector of the next address. The CN field is gated to IAR bits 8 throuth 11.

TABLE 3-14. MICROWORD FIELD FUNCTIONS (Cont'd)

<table>
<tr><th>Field</th><th>Function</th></tr>
<tr><td>CS</td><td>The decode of this field specifies the setting or resetting of one of the status bits in the ST register. This information is then checked by either the high or low order branching fields (CH or CL) later in the program.</td></tr>
<tr><td>CV</td><td>This field is not decoded on the assignment table. It is used to specify the subsector of the data word to be fetched from memory. The CV field is gated to DAR bits 8 through 11.</td></tr>
<tr><td>CW</td><td>This field is not decoded on the assignment table. It is used to specify the sector of a data word to be fetched from memory. The CW field is gated to DAR bits 4 through 7.</td></tr>
<tr><td>CX</td><td>This field is similar to the CW field. It is used to specify the sector of the next address. The CX field is gated to IAR bits 4 through 7.</td></tr>
<tr><td>CY</td><td>This field is not decoded on the assignment table. It is used to specify the word of the address being fetched from memory. The CY field is gated to bits 12 through 15 of DAR. This field is used only in an F1 (one byte fetch) instruction. The last two bits of the field (14 and 15) specify which byte of the word is fetched as follows:
<table>
<tr><th colspan="2">DAR</th><th>Byte to</th></tr>
<tr><th>14</th><th>15</th><th>MA</th></tr>
<tr><td>0</td><td>0</td><td>0</td></tr>
<tr><td>0</td><td>1</td><td>1</td></tr>
<tr><td>1</td><td>0</td><td>2</td></tr>
<tr><td>1</td><td>1</td><td>3</td></tr>
</table></td></tr>
<tr><td>CZ</td><td>This field is not decoded on the assignment table. It is used to specify the low order branching conditions of the next address. It is a one bit field used only in the SPEX format. The CZ field is gated to IAR bit 13.</td></tr>
<tr><td>FRMT</td><td>This 2-bit field is used in the special case formats to indicate the interpretation of bits 0 through 7. The bit configuration is 10, 00, 11, or 01 to specify formats 1 through 4 respectively. Bits 0 through 7 are then interpreted according to the specified format.</td></tr>
<tr><td>LTCH</td><td>This is not a field as such; but rather, a term used to indicate that the specified bits are latched to the same condition as specified by the current instruction address. Therefore, if the current address of a format 1 instruction is 04XX, the first two hex characters of the next address are latched to 04.</td></tr>
<tr><td>NA</td><td>This field is not decoded on the assignment table. It is used to suppress the entry of a specified register onto the A Bus. This field is used in formats where there is a CA/CD field. Normally the CA/CD field decode specifies a particular register to be loaded onto the A Bus and then</td></tr>
</table>

TABLE 3-14. MICROWORD FIELD FUNCTIONS (Cont'd)

Field	Function
NA (Cont'd)	reloaded with the results of the ALU operation from the D Bus. However, by setting the NA field to a 1, entry of the specified register onto the A Bus is suppressed.
ND	This field is similar to the NA field. It is used to suppress the loading of a register (specified by the CA/CD field) from the D Bus.
OP	The decode of this field specified the ALU operation to be performed.
SEE FORMAT 1-4	This is not a field as such; but rather, is used to indicate that the interpretation of bits 0 through 7 are dependent on the format specified by bits 30 and 31.
SPEX	The decode of this field specifies one of the 32 special executes that may be accomplished. Refer to Table 3-16 for definitions of each of the special executes.
UC	This is not a field as such; but rather, is used to indicate that the specified bits are unchanged from the last execution of the data address register.

TABLE 3-15. DECODE ASSIGNMENTS

Decode	CA/CD	CA	CB	CS	CH	CL	OP
00	--	--	--	--	0	0	A+B=D
01	AH	AH	AA	0=ST4	1	1	A+B=DC
02	AT	AT	AB	0=ST1	ST0	INDEX	A+B+C=DC
03	SG	SG	AC	1=ST1	ST2	ST3	A-B+C=DC
04	SH	SH	AD	0=ST0	ST4	ST5	A-B+1=D
05	ST	ST	AE	1-ST0	ST6	ST7	A/B=D
06	EE	EE	AG	0=ST5	EB0	EB1	A•B=D
07	ED	ED	SA	1=ST5	EB2	EB3	A@B=D
08	SB	SB	MA	0=ST2	EB4	EB5	
09	MB	MB	SC	DNST21	EB6	EB7	+= ADD
0A	MC	MC	SD	0=ST3	TERMD/INTAC	BYRDY	-= 1's Complement
0B	MD	MD	SE	1=ST3	CARRY	SELTD	/= OR
0C	IA	IA	EB	0=ST6	COMND/WHORU	DOPAR/OVRUN	• = AND
0D	IB	IB	EC	1=ST6	CMPAR/INLIN	D=0	@= EXCLUSIVE OR
0E	IE	IE	EA	0=ST7	C-CHK	CUEND	
0F	EG	EG	ID	1=ST7	A=IAR	SPEX	
10	IC						
11	AA						
12	AB						
13	AC						
14	AD						
15	AE						
16	AG						
17	SA						
18	MA						
19	SC						
1A	SD						
1B	SE						
1C	EB						
1D	EC						
1E	EA						
1F	ID						

NOTE: IC register is not transferrable to any other register.

DNST21 = If D Bus is not zero, set ST2 to 1.

TERMD/INTAC = Terminated/Interrupt Active.

CMND/WHORU = Command/Who Are You?

CMPAR/INLIN = Compare/Inline

C-CHK = Control Check

BYRDY = Byte Ready

SELTD = Selected

DOPAR/OVRUN = Data Out Parity/Overrun

CUEND = Control Unit End

TABLE 3-16. SPECIAL EXECUTION DEFINITIONS

Decode (Decimal)	Function
00	ENABLE CUDI CHECKS – A latch is set which allows any CUDI errors to set the CUDI check latches. The Enable latch is reset with Not Tag Gate.
01	CZ CHECK – Dummy SPEX used to check setting and resetting of CZ field in Hardcore Tests.
02	SET REPEAT LATCH – The Repeat Latch is one condition that generates the INLIN branch (CH decode of XOD). The latch is set by an in-line routine to request re-entry.
03	ENABLE MEMORY ECC CHECK LATCHES – Enable setting of memory ECC check bit latches by resetting inhibit latch set by SPEX 17.
04	LOAD DAR – B Bus 5-7 is transferred to DAR 1-3.
05	LOAD IAR (QUADRANT SWITCH) – B Bus 1-7 is transferred to IAR 1-7, which allows the u-code to switch from one quadrant to another. (This SPEX can be used with the A=IAR Branch for a 4K-way branch providing A Bus=AE.)
06	RESET CMND/WHORU BRANCH – Resets the COMND/WHORU Branch. Must be executed at least 200 nsec after the branch becomes true.
07	Unused
08	RESET EXECUTE AND REPEAT LATCHES – The Execute Latch is set by the Execute Switch whenever the Function Switch is in the In-line or CE position. The Execute latch is one condition which generates the INLIN branch. (See SPEX 02 for an explanation of the Repeat Latch.)
09	SET ROUTINE REG – The B Reg is transferred to the Routine Reg, which stores the routine number of the current loaded routine. The Routine Reg may be displayed at the CE panel.
10	SET ERROR REG – The Error Reg stores the most recent in-line error code, which may be displayed at the CE panel.
11	INITIATE CASSETTE ACTION – The cassette command byte is transferred from the B Reg to the cassette latches, which initiates cassette operation as follows: Bit 0, 1, 2, 3, 4, 5 } Identifies number of blocks to space 6 Rewind to BOT 7 Turn on IMPL data accepted
12	UNFREEZE CHANNEL SWITCH – This allows the channel switch to return to neutral.

TABLE 3-16. SPECIAL EXECUTION DEFINITIONS (Cont'd)

Decode (Decimal)	Function
13	ALLOW DISABLE – The channel identified by IE Reg 5 is allowed to disable if the Disable Switch so indicates. The u-code allows disable whenever a given channel is inactive.
14	GATE CASSETTE READ DATA – The cassette read data byte is logically ORed onto the D Bus during the next machine cycle. The next instruction must be "O=MD".
15	PROGRAM STOP – The "hardcore" microdiagnostic issues this command to stop the machine if an error is detected.
16	RESET CONTROL CHECK – The u-code resets all control check error latches after appropriate system action.
17	INHIBIT MEMORY ECC CHECK LATCHES – Inhibit setting of Memory ECC Check bit latches for diagnostics.
18-19	Unused
20	CLEAR BLOCK SWITCH TO CHANNEL – This SPEX resets the Block Channel condition set by 22B.
21	FORCE CHANNEL SWITCH TO X – The wrap-around microdiagnostic uses this to force the channel switch to a particular position.
22	SET INLIN INHIBIT TIMER/BLOCK CHANNEL – 1. Inline Inhibit – This causes the INLIN branch to be inhibited by the hardware for a specified period of time. The value placed on the low order B Bus (bits 4-7), which specifies the number of 250 ms increments that the INLIN branch is to be blocked, is transferred to a timer. Bit 0 on the B Bus specifies the conditions under which the inhibit logic is to be initialized. Bit 0=1 Set the Extended Inhibit latch and initiate the timer. Bit 0=0 Set the Temporary Inhibit latch and initiate the timer. (The hardware automatically blocks this action if the Extended Inhibit latch is already set.) Both inhibit latches reset when the specified number of time increments have elapsed. The Temporary Inhibit latch also resets upon exit from the wait loop due to a channel initiated operation. (SELTD active.)

TABLE 3-16. SPECIAL EXECUTION DEFINITIONS (Cont'd)

Decode (Decimal)	Function
22 (Cont'd)	This function is used by the microprogram to inhibit re-entry to the in-line diagnostics. The extended inhibit function facilitates maximum system throughput during in-line mode by providing the channel a fixed period of time before in-line re-entry is permitted. The temporary inhibit function prevents re-entry during the wait loop only. This allows the channel time to initiate at least one operation before in-lines regain priority. 2. Block Channel – If B Bus bit 1=0, a latch is set to block switching to the channel designated by IE5 for 10 ms.
23	SHIFT PO – The ECC PO register is shifted one bit position. P1, P2, and P3 registers are also shifted one bit position if ECC decode = Allow Descode Write (IEO-2=111).
24	SHIFT P1, P2, AND P3 – The ECC P1, P2, and P3 registers are shifted one bit position.
25	FETCH 4 INHIBIT – Must be executed before an F4 if no modification of MB, MC, and MD is required.
26	SET READY – Set Ready on channel designated by channel switch.
27	SET END – Set End and Service Request on channel designated by channel switch.
28	SET INTERRUPT – Set Interrupt on channel designated by IE5.
29, 30, 31	Unused

INSTRUCTION EXECUTION

As the microprogram executes, it is continually accessing memory to retrieve the microwords. Memory is accessed either through the Instruction Address Register (IAR), or through the Data Address Register (DAR). This is necessary since memory is used both for the storage of instruction words and for the storage of data to be used in the processing of instruction words. In most word formats, only IAR is used to access memory. However, if a memory fetch or store operation is specified by the word format, then both DAR and IAR are used.

As the controller processes an instruction word, the following sequence of events occurs during one machine timing cycle (200 nanoseconds). It should be noted that step 6 is only concerned with status bits which are set under the CS field control. Status bits may be set at other times, as a result of the ALU operation or other machine conditions.

1. The memory is enabled and the data bits are set in the memory latches.
2. The format is identified and decoding of the word begins.
3. Specified branch conditions are tested and the next address is generated.
4. The specified registers are gated to the ALU A and B registers.
5. The specified ALU operation is executed.
6. Status bits are set under control of the CS field.
7. The result of the ALU operation is gated from the D Bus to the specified register.

This sequence is referred to as an instruction cycle.

A data cycle is similar to an instruction cycle; however, a data cycle requires two machine timing cycles (400 nanoseconds) for its execution. A data cycle is used when a data fetch or store operation is specified by the word format. As the controller processes a data word, the following sequence of events occurs in the space of two machine timing cycles.

1. The memory is enabled and the data bits are set in the memory latches.
2. The format is identified and decoding of the word begins.
3. Specified branch conditions are tested and the next instruction and data addresses are generated (IAR/DAR are generated).
4. The specified registers are gated to the ALU A and B registers.
5. The specified ALU operation is executed.
6. Decode inhibits are set to prevent decoding of word during the second timing cycle. DAR addresses memory and memory is enabled. Data is set in memory latches (in the case of a fetch operation).
7. Data is gated from memory latches to Memory registers (fetch), or from Memory registers to memory (store).
8. Status bits are set under control of CS field.
9. The result of the ALU operation is gated from the D Bus to the specified register.

The following discussion explains Figures 3-45 through 3-60, which show each of the format executions individually. On each of the illustrations, the sequence of events are listed along the left margin opposite the illustrated event. When one of the steps in the sequence is altered or not accomplished, the reason is explained in parenthisis. An attempt is made to keep the illustrations consistent with machine timing. However, the purpose of the illustrations is

to show the format execution and field relationships. Therefore, the illustrated sequence of the fetch and store formats have been slightly altered. For a detailed explanation of timing, refer to the Machine Timing heading in the Circuit Theory section of this manual. Each of the illustrations shows the relationship of the microword to the format, and the fields within each format to their decode and machine functions. Each illustration has a block showing the conditions that are assumed to exist at the time of execution. It should be noted that numbers in the constant field (CK), may be expressed in either decimal or hexadecimal notation. The hexadecimal number is always preceded by the character X.

Immediately beneath the byte number/bit assignment block, at the top of the page, is the hexadecimal notation which represents the content of the microword. The hexadecimal notation is shown because that is the form used in the Microprogram Manual Core Map. Below the hexadecimal notation, the word is expressed again in binary form, representing the bits on the output of the memory latches. The illustrations then show the format by which the particular word is decoded. Each field of the word has a block showing the interpretation of the bits in its field. Information contained in the interpretation blocks is summarized from Tables 3-14 and 3-15.

At the bottom of each illustration is the microblock. This block is shown as it appears in the Microprogram Manual. By tracing through the illustrations, the relationship of the various fields to the microblock become apparent. For a detailed description of how to interpret the microprogram, refer to the Flow Diagram heading later in this part of the manual.

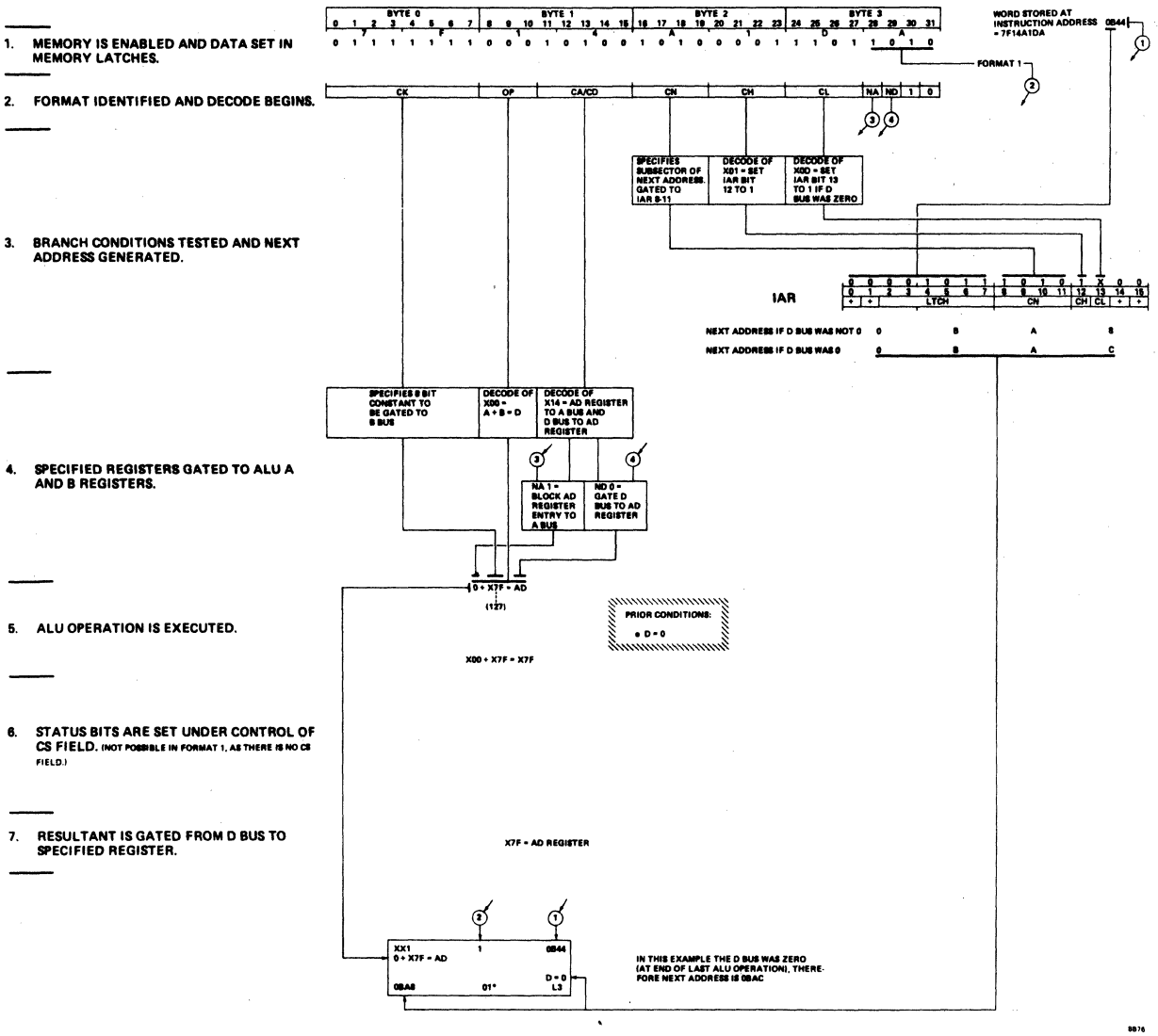


Figure 3-45. Format 1 Execution

Format 1

Format 1 is decoded when bits 30 and 31 of the instruction word are 1 0 respectively (see Figure 3-45). The NA and ND fields (bits 28 and 29) may be 0 1, 1 0, or 0 0. A 1 1 configuration is not valid. Format 1 is used when a full 8 bit constant field is needed. If the ALU operation is specified, the A Bus source and the D Bus destination must be the same if both are specified. The B Bus source is either the 8-bit constant or the character 0. In format 1, a status statement is invalid as there is no CS field control.

The first two hexadecimal characters of the next address are latched to the same condition as the first two characters of the current address. Next address, bits 0 through 7, must be the same as current address bits 0 through 7; unless special execute 05 is specified.

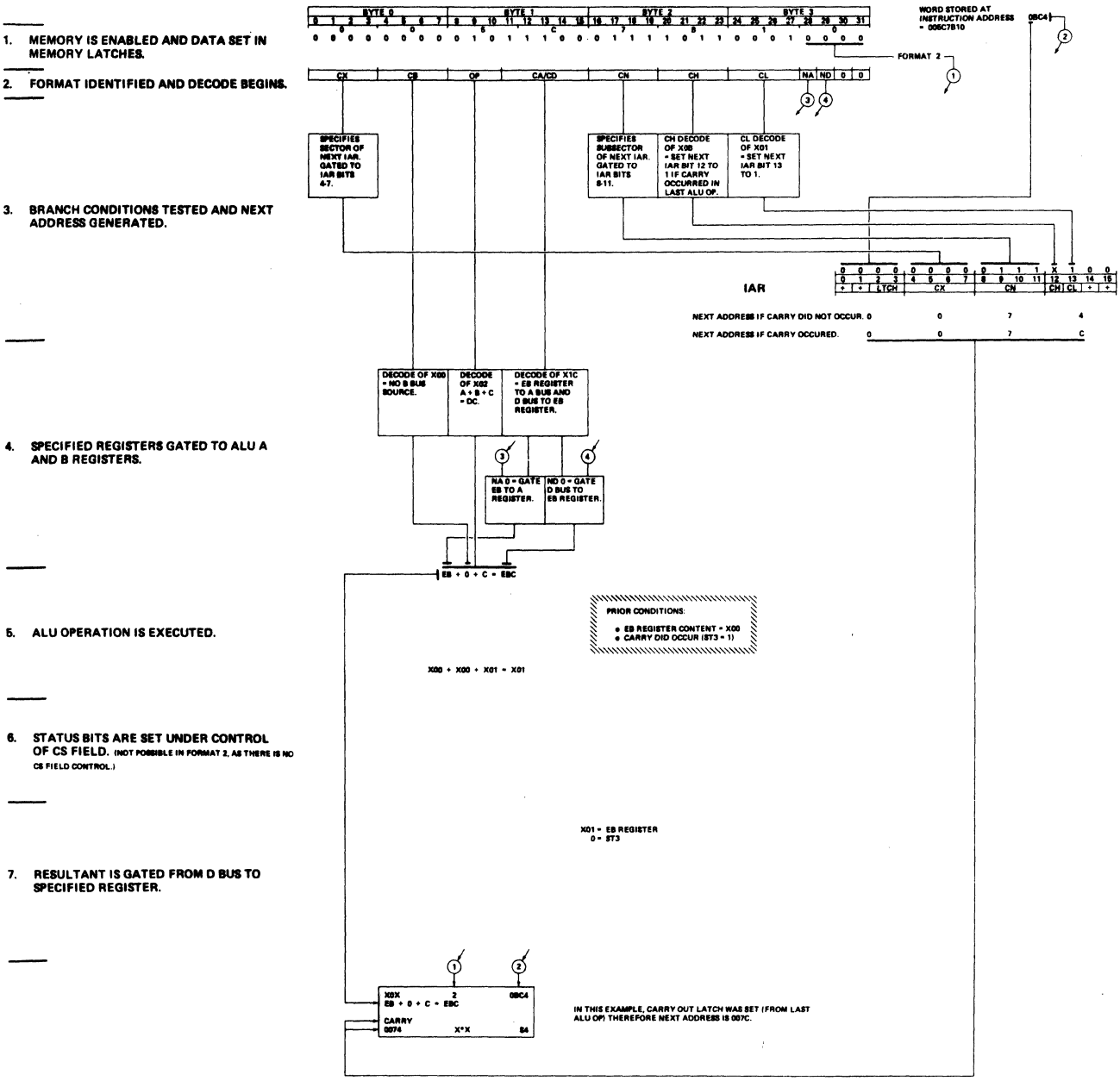


Figure 3-46. Format 2 Execution

Format 2

Format 2 is decoded when bits 30 and 31 of the instruction word are 0 0 (see Figure 3-46). The NA and ND fields (bits 28 and 29) may be 0 1, 1 0, or 0 0. A 1 1 configuration is not valid. Format 2 is used when neither a CK nor a CS field is required, and is used to allow sector switching. This is possible because of the CX field which is used to specify the second hexadecimal character of the next address. If an ALU operation is specified, the A Bus source and the D Bus destination must be the same if both are specified. The B Bus source for the ALU operation is either a register in the CB column of the decode assignment table or the character 0. A status statement is not valid in format 2.

The first hexadecimal character of the next address is latched to the same condition as that of the first character of the current address. Next address, bits 0 through 3 must be the same as current address bits 0 through 3; unless special execute 05 is specified.

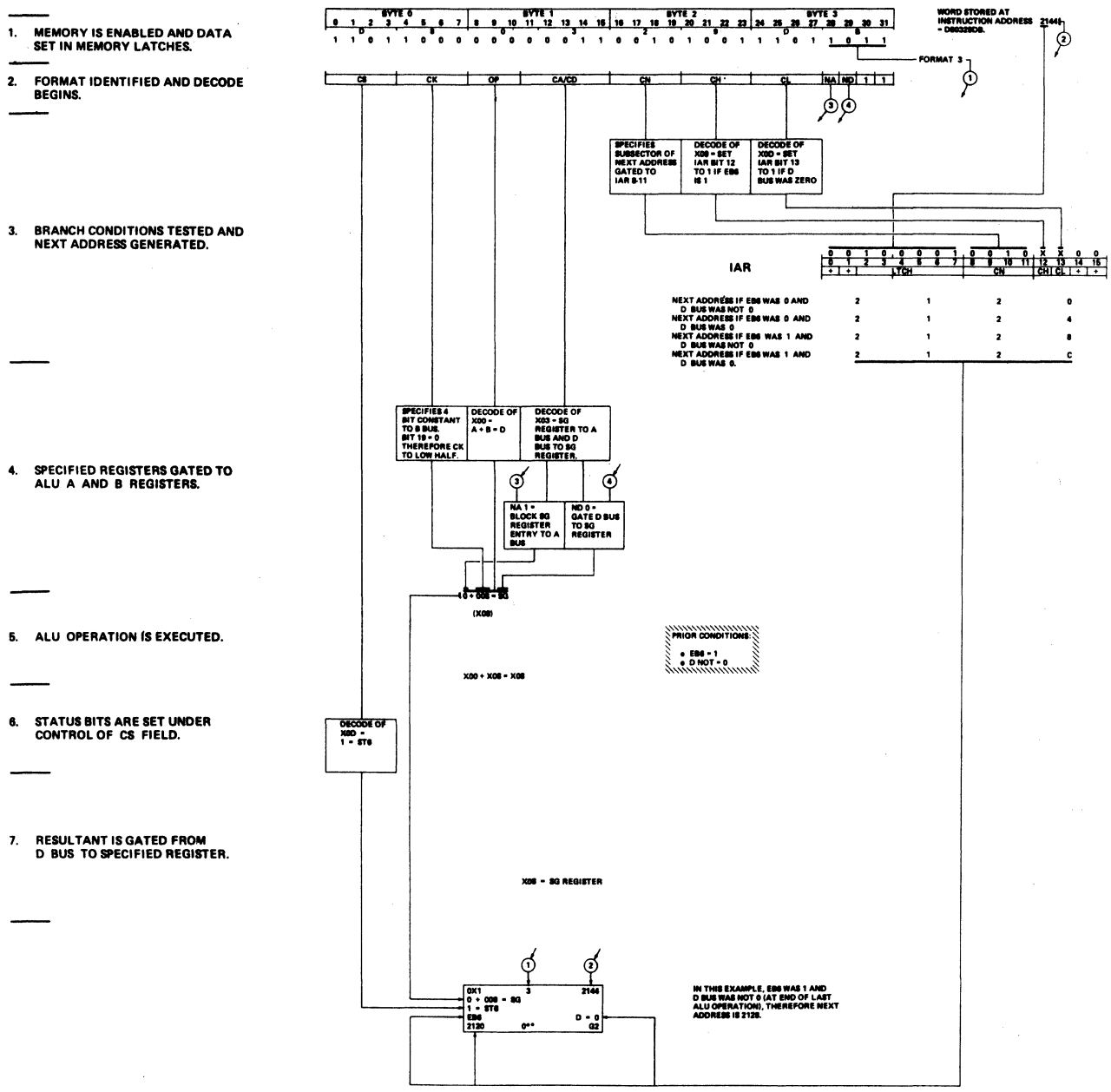


Figure 3-47. Format 3 Execution

Format 3

Format 3 is decoded when bits 30 and 31 of the instruction word are 1 1 (see Figure 3-47). The NA and ND fields (bits 28 and 29) may be 0 1, 1 0, or 0 0. A 1 1 configuration is not valid. Format 3 is used when a four bit constant (CK) and a four bit status (CS) field are required. If an ALU operation is specified, the A Bus source and the D Bus destination must be the same if both are specified. The B Bus source must be a constant or the character 0.

The constant is gated to the low half (bits 4 through 7) of the B Bus if bit 19 of the instruction word is 0, and to the high half of the B Bus if bit 19 is 1. Either the high or low half of the constant must be zero.

The first two hexadecimal characters of the next address are latched to the same condition as the first two characters of the current address. Next address, bits 0 through 7, must be the same as current address bits 0 through 7; unless special execute 05 is specified.

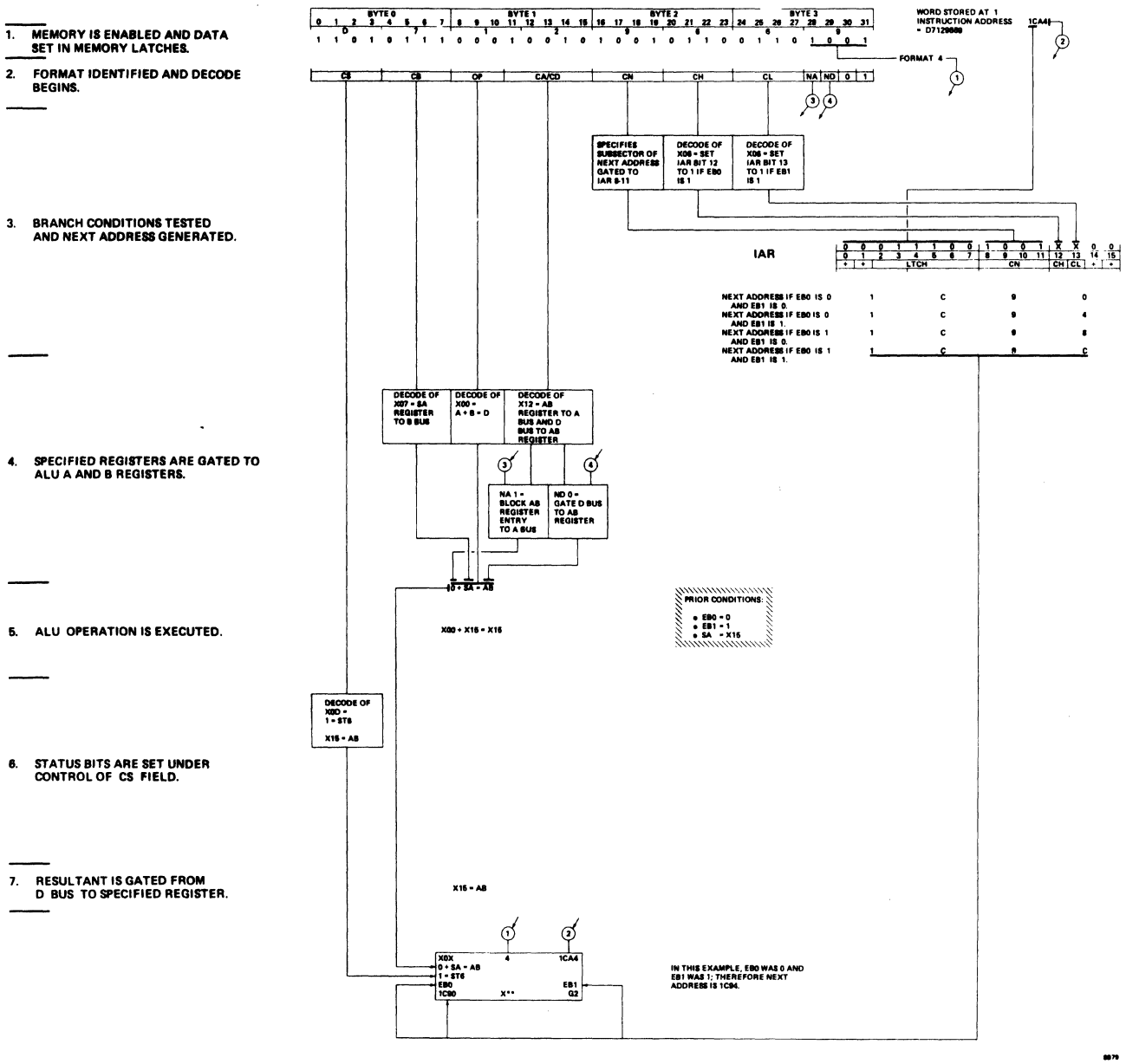


Figure 3-48. Format 4 Execution

Format 4

Format 4 is decoded when bits 30 and 31 of the instruction word are 0 1 respectively (see Figure 3-48). The NA and ND fields (bits 28 and 29) may be 0 1, 1 0, or 0 0. A 1 1 configuration is not valid. Format 4 is used when a four bit status field and a four bit CB field are needed. If the ALU operation is specified, the A Bus source and the D Bus destination must be the same if both are specified. The B Bus source for the ALU operation is either a register in the CB column of the decode assignment table or the character 0.

The first two hexadecimal characters of the next address are latched to the same condition as the first two characters of the current address. Next address, bits 0 through 7, must be the same as current address 0 through 7; unless special execution 05 is specified.

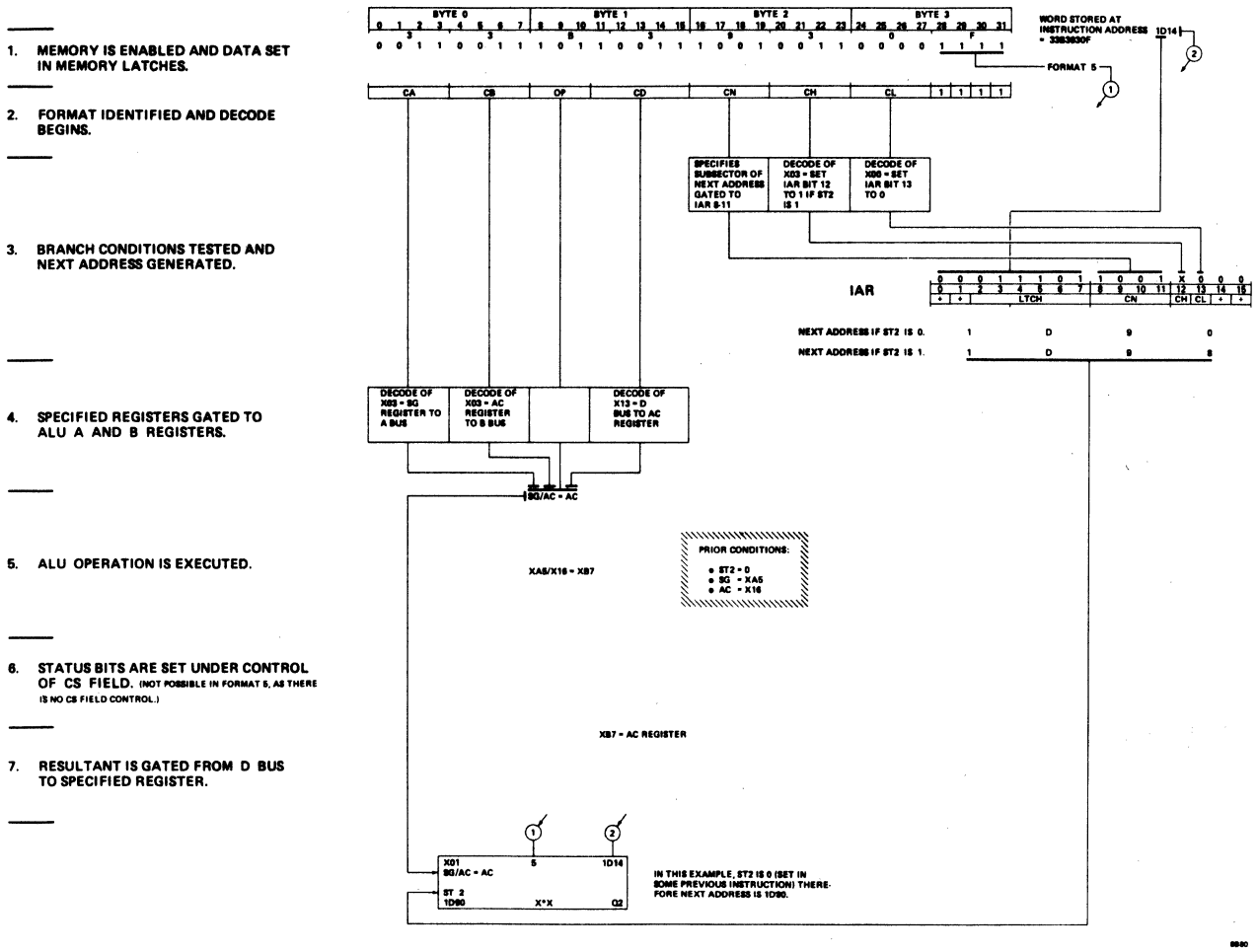


Figure 3-49. Format 5 Execution

Format 5

Format 5 is decoded when bits 28 through 31 of the instruction word are 1 1 1 1 (see Figure 3-49). In this format, the only possible suppression of the A or D Buses is a decode of 0, since there are no NA or ND fields. Format 5 has separate control of the A and D buses and is used when it is necessary to transfer information between registers. Only registers in the CA column of the decode assignment table are valid as the A Bus source. The B Bus source must be a register specified in the CB column of the decode assignment table or the character 0.

Only registers in the upper half (decode 00 through 0F) of the CA/CD column of the decode assignment table are valid for the D Bus destination. A status statement is not valid in a format 5 instruction, as there is no CS field control.

The first two hexadecimal characters of the next address are latched to the same condition as the first two characters of the current address. Next address, bits 0 through 7, must be the same as current address bits 0 through 7; unless special execute 05 is specified.

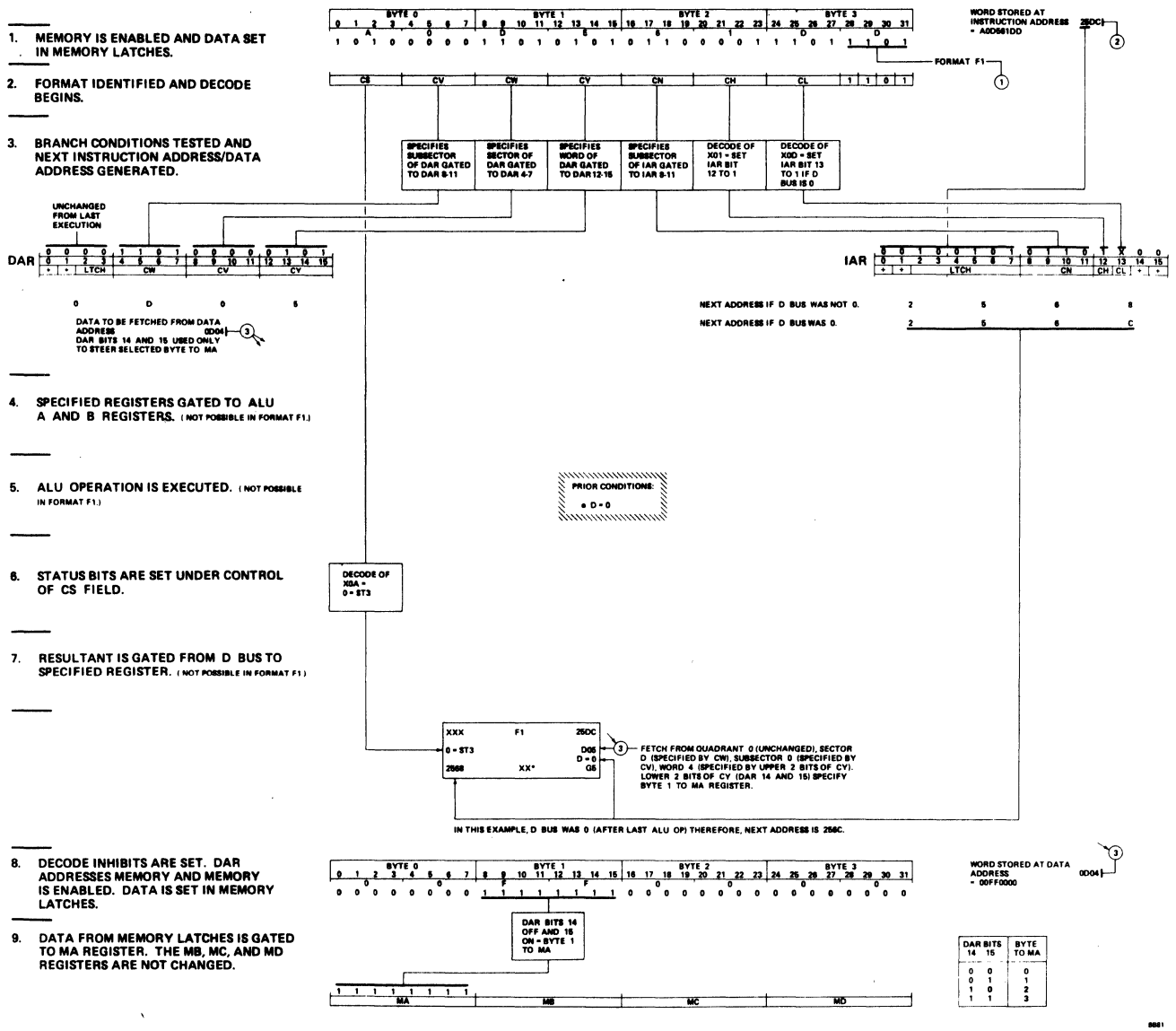


Figure 3-50. Format F1 Execution

Format F1

Format F1 is decoded when bits 28 through 31 of the instruction word are 1 1 0 1 respectively (see Figure 3-50). The decode of this format prohibits an ALU operation. The instruction fetches one 8-bit byte of information from the address constructed in DAR. The specific byte that is fetched is controlled by bits 14 and 15 of DAR. As shown in the lower right-hand corner of the figure, if DAR bits 14 and 15 are 0 0, byte 0 of the data word is transferred to the MA register. If DAR bits 14 and 15 are 0 1, byte 1 is transferred to the MA register.

This instruction generates the address of the data word to be fetched (DAR) as well as the next instruction address. The instruction is capable of controlling the three lower order hexadecimal characters of the Data Address register. The quadrant (high order character) is unchanged from the last execution of DAR. The only way the quadrant of a data address can be changed is by executing a special execute 04 instruction.

The first two characters of the next address are latched to the same condition as the first two characters of the current address. Next address, bits 0 through 7, must be the same as current address bits 0 through 7; unless special execute 05 is specified.

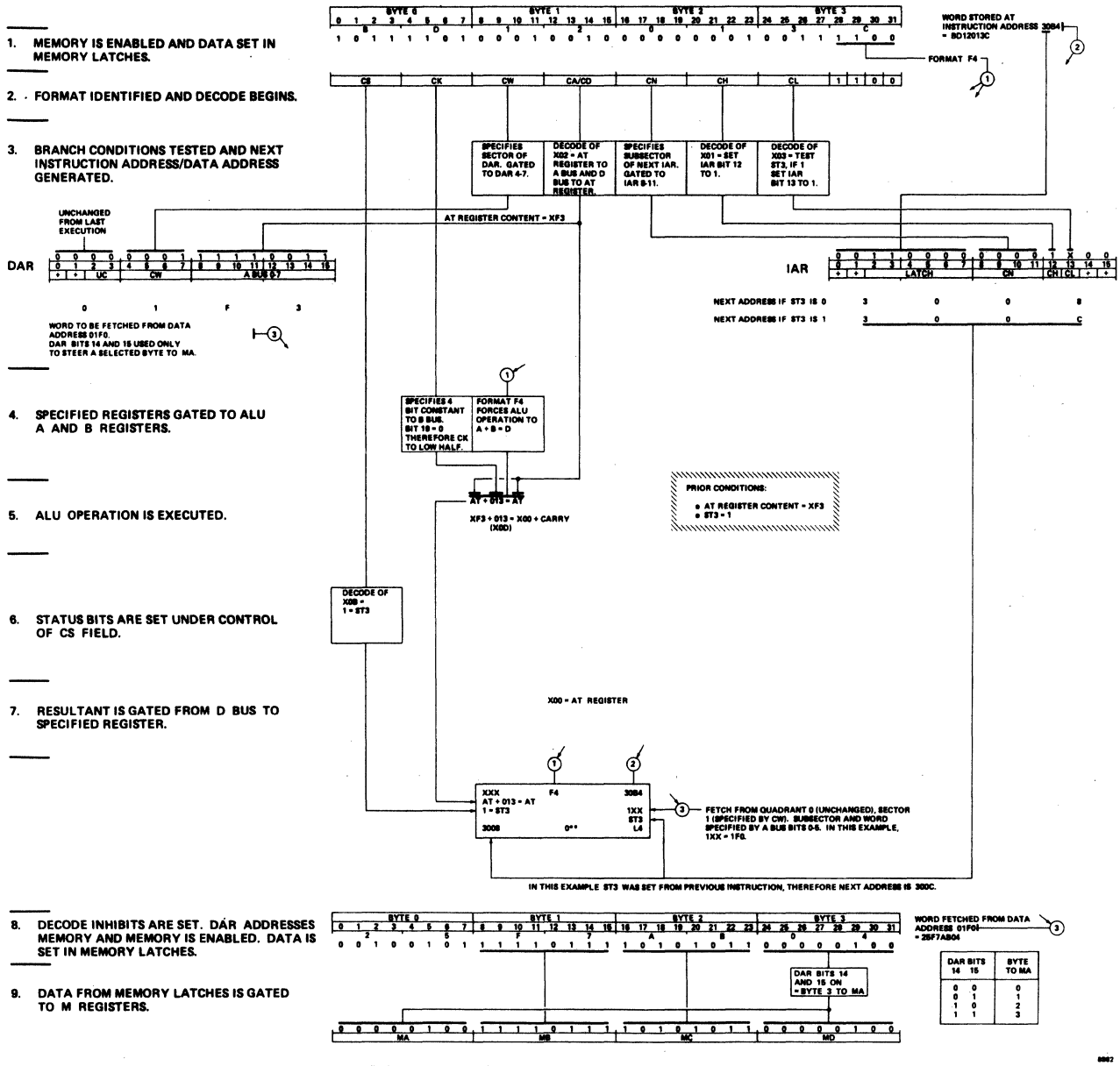


Figure 3-51. Format F4 Execution

Format F4

Format F4 is decoded when bits 28 through 31 of the instruction word are 1 1 0 0 respectively (see Figure 3-51) and used to perform a four byte data fetch. The ALU operation is forced to an A+B=D situation. The A Bus source and the D Bus destination must be the same if both are specified. The B Bus source is a four bit constant or the character 0. The constant is gated to the low half of the B Bus (bits 4 through 7) if bit 19 of the instruction word is 0, and to the high half of the B Bus if bit 19 is 1.

All four bytes of data are transferred from the Memory Latches to the Memory registers. It also has the capability of rearranging the byte order. In the example shown in the figure, byte 3 is transferred to the MA register in accordance with DAR bits 14 and 15. Byte 3 is also loaded into its normal position in the MD register.

This instruction generates the address of the data word to be fetched (DAR) as well as the next instruction address. The instruction is capable of controlling the three lower order hexadecimal characters of the data address register. The quadrant (high order character) is unchanged from the last execution of DAR. The only way the quadrant of a data address can be changed is by executing a special execute 04 instruction. The two lower order characters of DAR, are set prior to the execution of the ALU operation.

The first two characters of the next address are latched to the same condition as the first two characters of the current address.

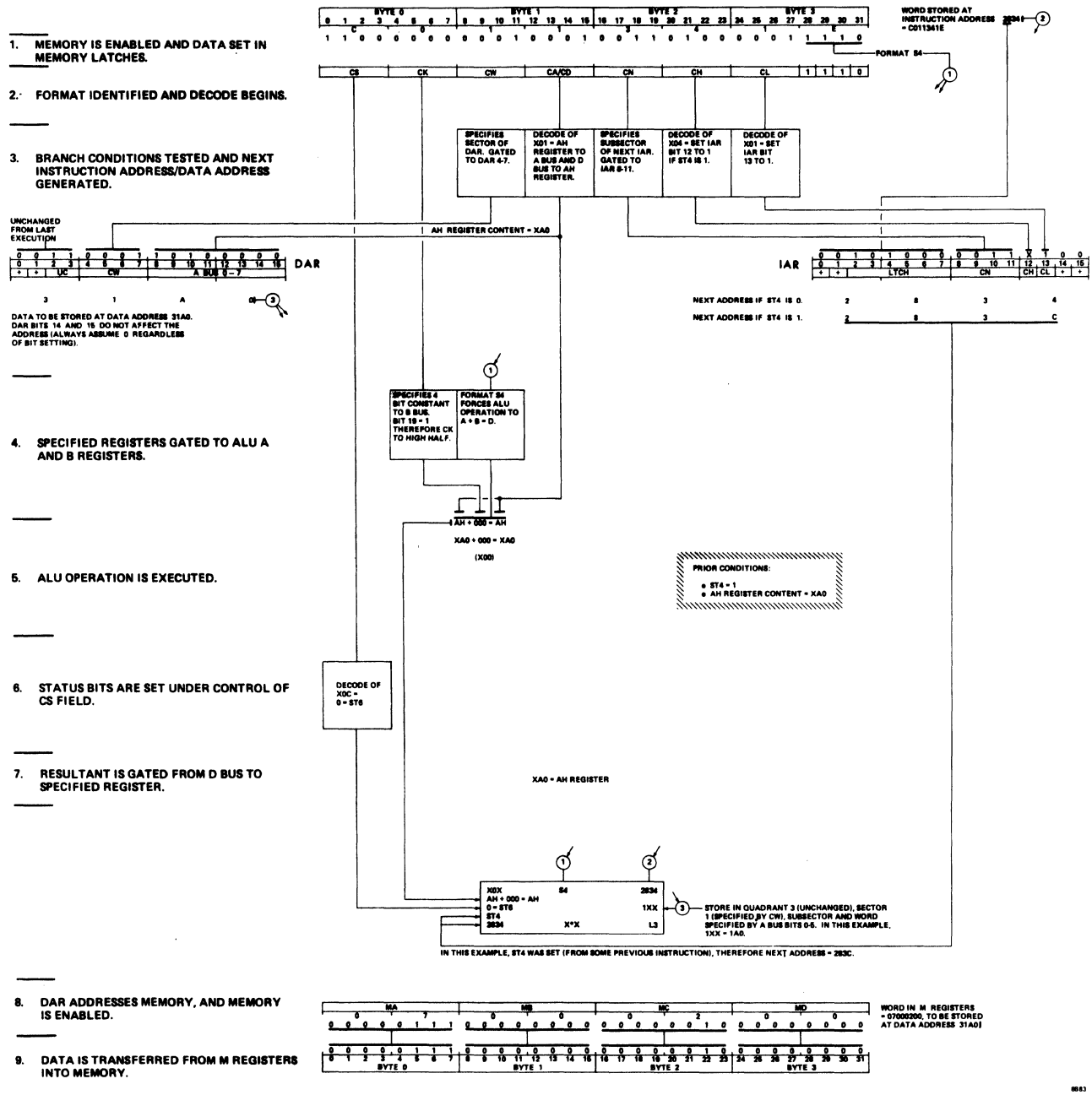


Figure 3-52. Format S4 Execution

Format S4

Format S4 is decoded when bits 28 through 31 of the instruction word are 1 1 1 0 respectively (see Figure 3-52). The format is used to perform a four byte data store operation at the address constructed in DAR. The format forces the ALU operation to an A+B=D situation. The A Bus source and the D Bus destination must be the same, if both are specified. The B Bus source is the four bit constant or the character 0. The constant is gated to the low half of the B Bus (bits 4 through 7) if bit 19 of the instruction word is 0, and to the high half of the B Bus if bit 19 is 1.

This instruction generates the address at which the data is to be stored, as well as the next instruction address. The instruction is capable of controlling the three lower order hexadecimal characters of the Data Address register. The quadrant (high order character) is unchanged from the last execution of DAR. The only way the quadrant of the data address can be changed is by executing a special execute 04 instruction. It should be noted that the two lower order characters of DAR, which are controlled by the content of the A Bus, are set prior to the execution of the ALU operation. The instruction then stores the content of the Memory registers at the data address generated by the instruction.

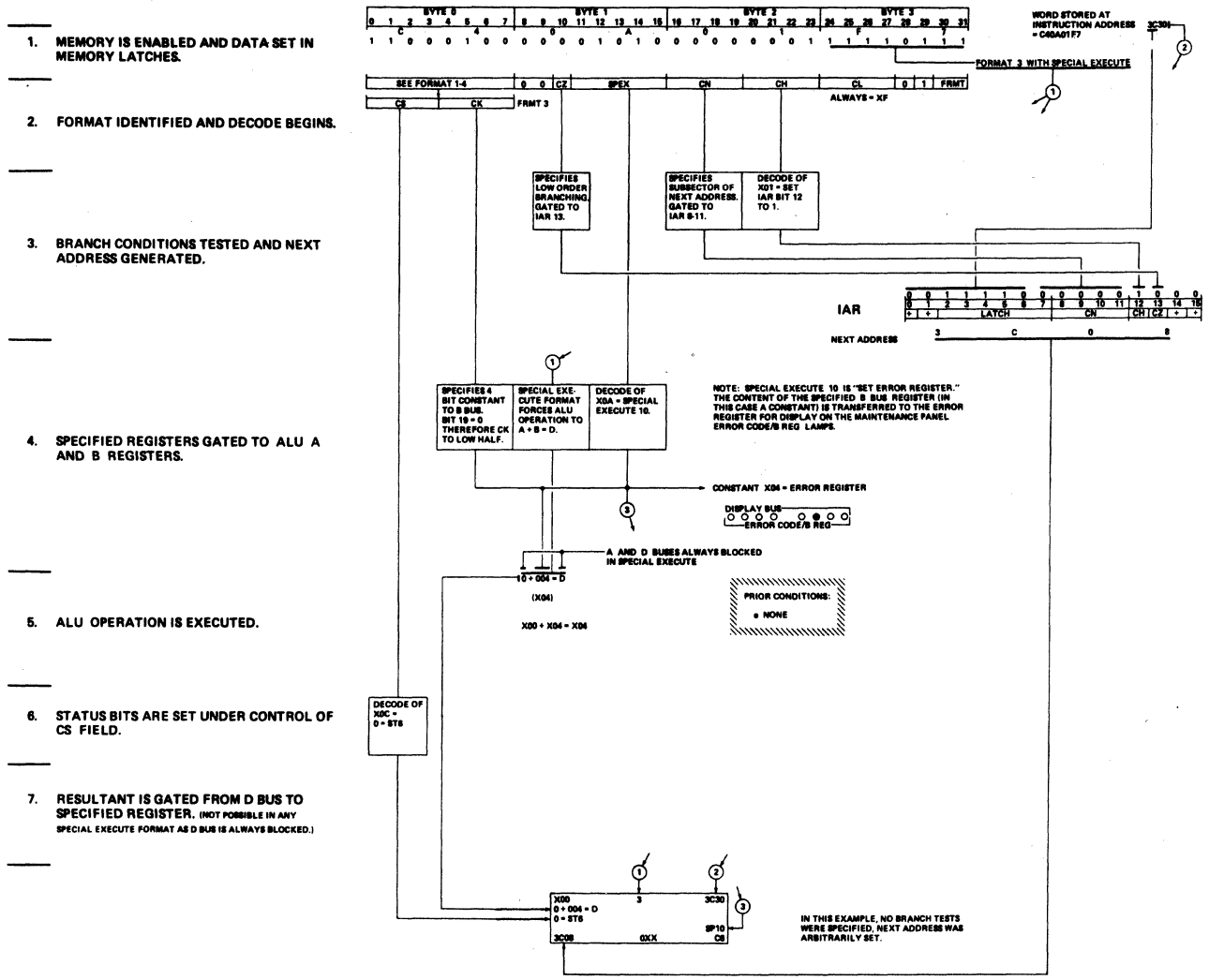


Figure 3-53. Format SPEX Execution - Format not 2, SPEX not 05

Format SPEX

Format SPEX is decoded when the hexadecimal code F appears in the CL field, bits 24 through 27; and bits 28 and 29 are in a 0 1 configuration. Bits 30 and 31 then determine the interpretation of instruction bits 0 through 7. Bits 30 and 31 may specify formats 1, 2, 3, or 4. In the example shown in Figure 3-53, format 3 is specified, therefore, instruction bits 0 through 7 are interpreted as the CS and CK fields. There are four variations of this format depending on the format specified and the special execute specified. The variations differ only in the configuration of the Instruction Address register, and whether or not a Data Address register is specified. If the format is not 2 and the special execute is not 05, Figure 3-53 illustrates the execution and format configuration. If the format is 2 but the special execute is not 05, Figure 3-54 illustrates the execution and format configuration. If the special execute is 05 then format 2 is not valid, see Figure 3-55 for this example. Figure 3-56 illustrates a special execute 04, which requires the construction of a data address. Except for the special execute 05, the SPEX format and the A=IAR format are mutually exclusive.

In each of the variations the format that governs the interpretation of instruction bits 0 through 7, is specified by bits 30 and 31. The SPEX field, instruction bits 11 through 15, controls which of the 32 possible special executes is being accomplished. Refer to Table 3-16 for a definition of the decoded special execute.

When the SPEX format is decoded the ALU operation is forced to an A+B=D situation. The A Bus source and the D Bus destination are always blocked by this format. The B Bus source is dependent on which of the four formats is specified. The B Bus source may be a specified register or a four or eight bit constant. If a four bit constant is specified, instruction bit 19 controls whether it is gated to the high or low half of the B Bus as explained under format 3.

1. MEMORY IS ENABLED AND DATA SET IN MEMORY LATCHES.

2. FORMAT IDENTIFIED AND DECODE BEGINS.

3. BRANCH CONDITIONS TESTED AND NEXT ADDRESS GENERATED.

4. SPECIFIED REGISTERS GATED TO ALU A AND B REGISTERS.

5. ALU OPERATION IS EXECUTED.

6. STATUS BITS ARE SET UNDER CONTROL OF CS FIELD. (NOT POSSIBLE IN FORMAT 1 AS THERE IS NO CS FIELD CONTROL.)

7. RESULTANT IS GATED FROM D BUS TO SPECIFIED REGISTER. (NOT POSSIBLE IN ANY SPECIAL EXECUTE FORMAT AS D BUS IS ALWAYS BLOCKED.)

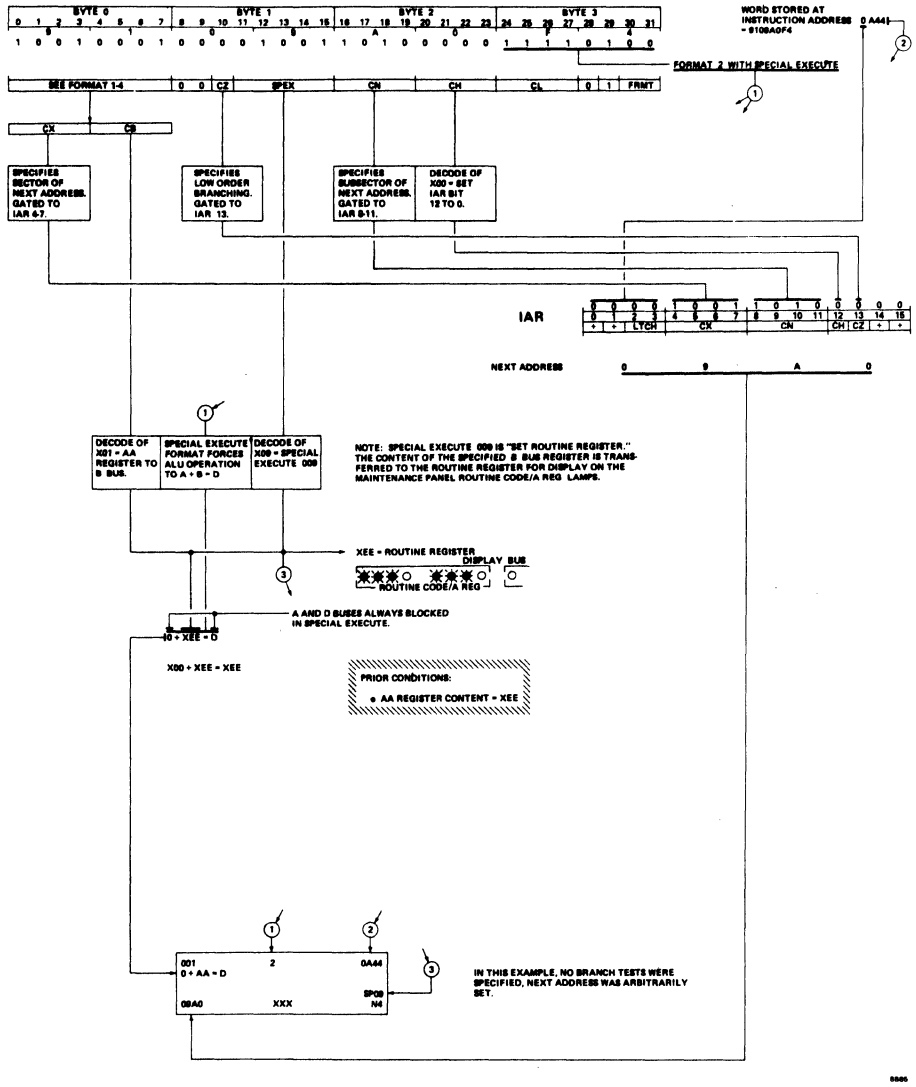


Figure 3-54. Format SPEX Execution - Format 2, SPEX not 05

1. MEMORY IS ENABLED AND DATA SET IN MEMORY LATCHES.

2. FORMAT IDENTIFIED AND DECODE BEGINS.

3. BRANCH CONDITIONS TESTED AND NEXT INSTRUCTION ADDRESS/DATA ADDRESS GENERATED.

4. SPECIFIED REGISTERS GATED TO ALU A AND B REGISTERS.

5. ALU OPERATION IS EXECUTED.

6. STATUS BITS ARE SET UNDER CONTROL OF CS FIELD. (NOT POSSIBLE IN FORMAT 1 AS THERE IS NO CS FIELD CONTROL.)

7. RESULTANT IS GATED FROM D BUS TO SPECIFIED REGISTER. (NOT POSSIBLE IN ANY SPECIAL EXECUTE FORMAT AS D BUS IS ALWAYS BLOCKED.)

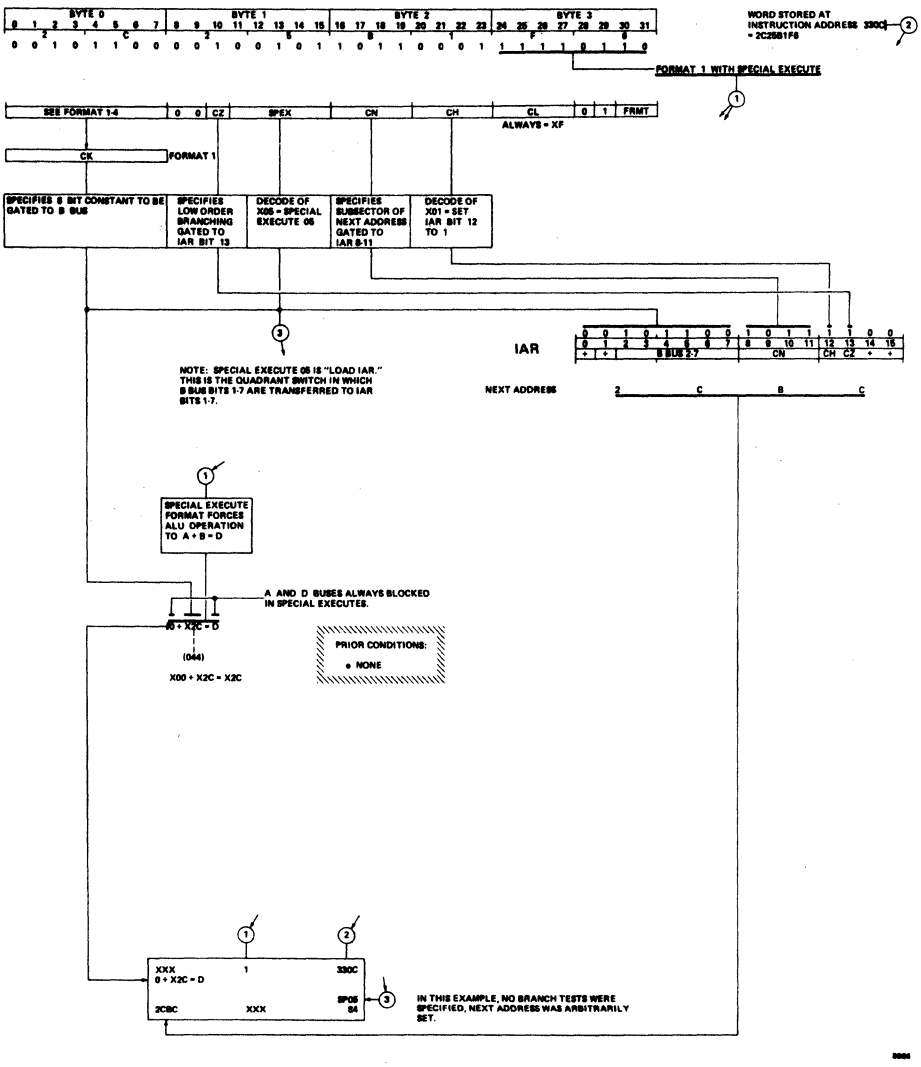


Figure 3-55. Format SPEX Execution - Format not 2, SPEX 05

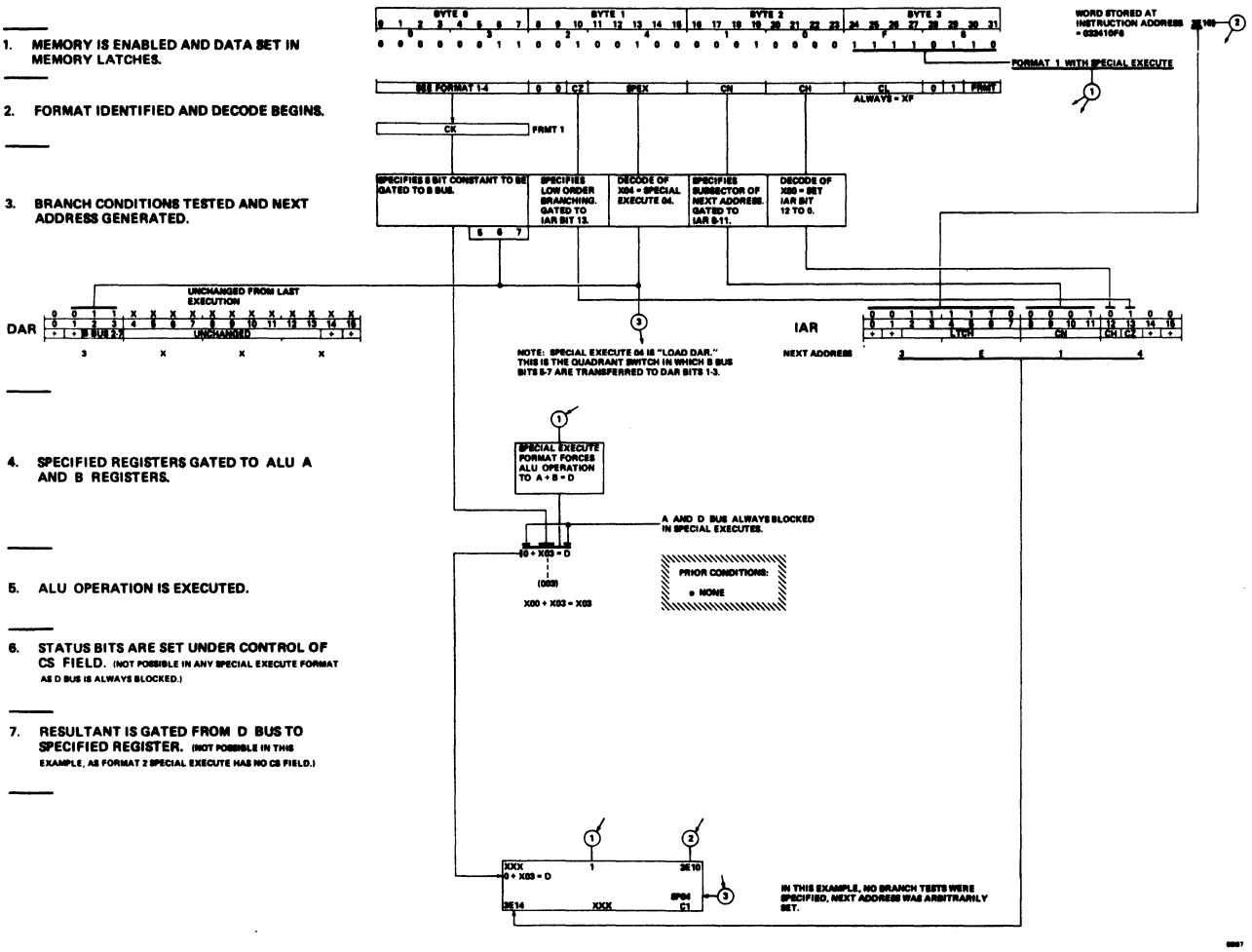


Figure 3-56. Format SPEX Execution - Format 1, SPEX 04

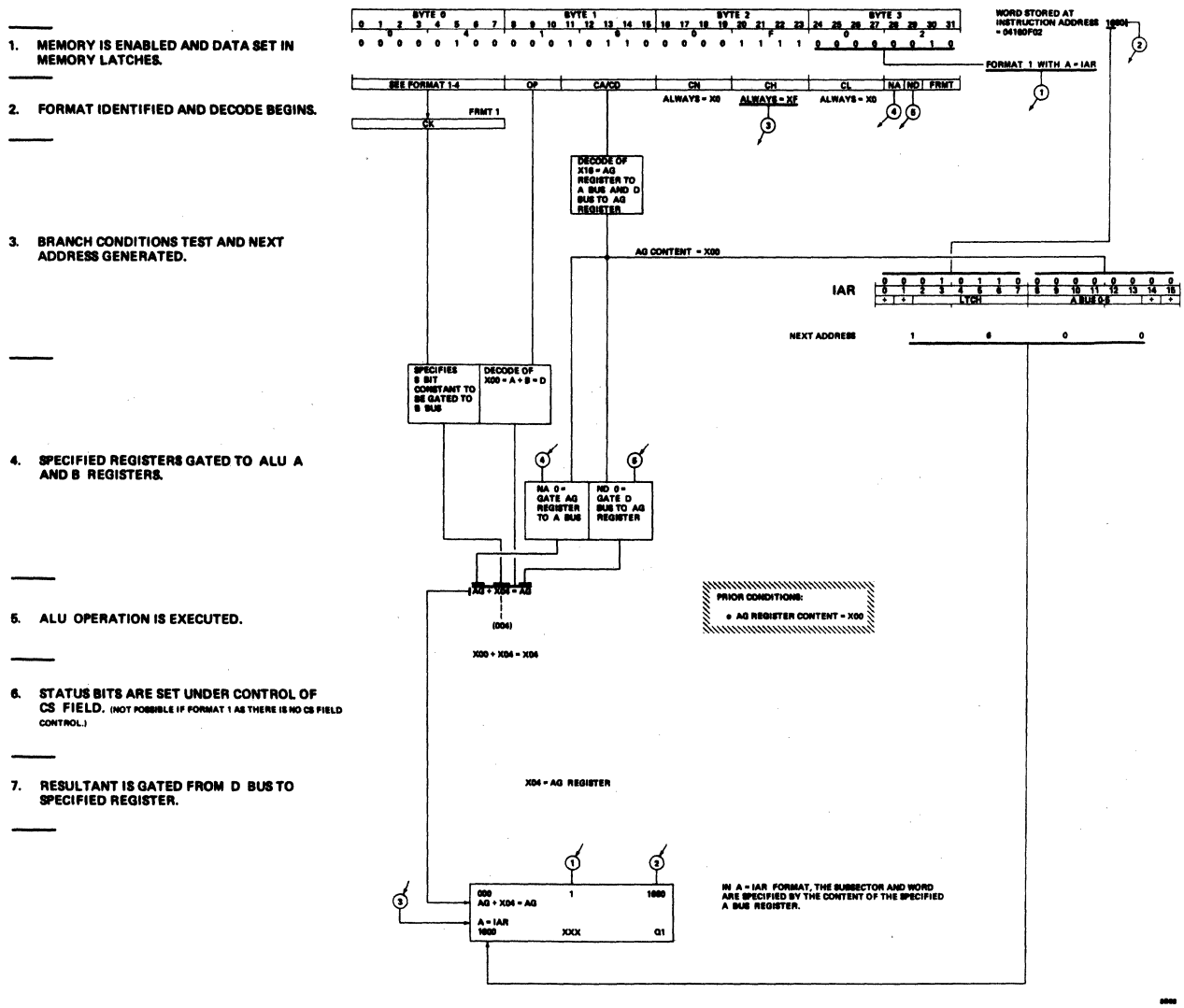


Figure 3-57. Format A=IAR Execution - Format not 2

Format A=IAR

Format A=IAR is decoded when the hexadecimal code F appears in the CH field, instruction bits 20 through 23; and the code 0 appears in the CL field, instruction bits 24 through 27. Bits 30 and 31 then determine the interpretation of instruction bits 0 through 7. Bits 30 and 31 may specify formats 1, 2, 3, or 4. Since the next address is assembled in the same manner in formats 1, 3, and 4; there are really only two variations of this format; format not 2, and format 2. Figure 3-57 illustrates an example of the format not 2 configuration, and Figure 3-58 illustrates the format 2 configuration. Normally in this instruction, the CN field is zero. However, if format 3 is specified, bit 19 of the instruction word is needed to control the gating of the four bit constant as explained under format 3. In this case the CN field may contain hexadecimal 1.

This format is used to transfer the content of the A Bus register to the two low order characters of the Instruction Address register. It should be noted that these two characters are set prior to the execution of the ALU operation. In formats 1, 3, and 4, the first two characters of the next address are latched to the same condition as the first two characters of the current address. Next address, bits 0 through 7, must be the same as current address bits 0 through 7; unless special execute 05 is specified. In format 2, only the first character of the next address is latched, the second character is controlled by the CX field. Except for special execute 05, the SPEX format and the A=IAR format are mutually exclusive.

1. MEMORY IS ENABLED AND DATA SET IN MEMORY LATCHES.

2. FORMAT IDENTIFIED AND DECODE BEGINS.

3. BRANCH CONDITIONS TESTED AND NEXT ADDRESS GENERATED.

4. SPECIFIED REGISTERS GATED TO ALU A AND B REGISTERS.

5. ALU OPERATION IS EXECUTED.

6. STATUS BITS ARE SET UNDER CONTROL OF CS FIELD. (NOT POSSIBLE IN FORMAT 2, AS THERE IS NO CS FIELD CONTROL.)

7. RESULTANT GATED FROM D BUS TO SPECIFIED REGISTER.

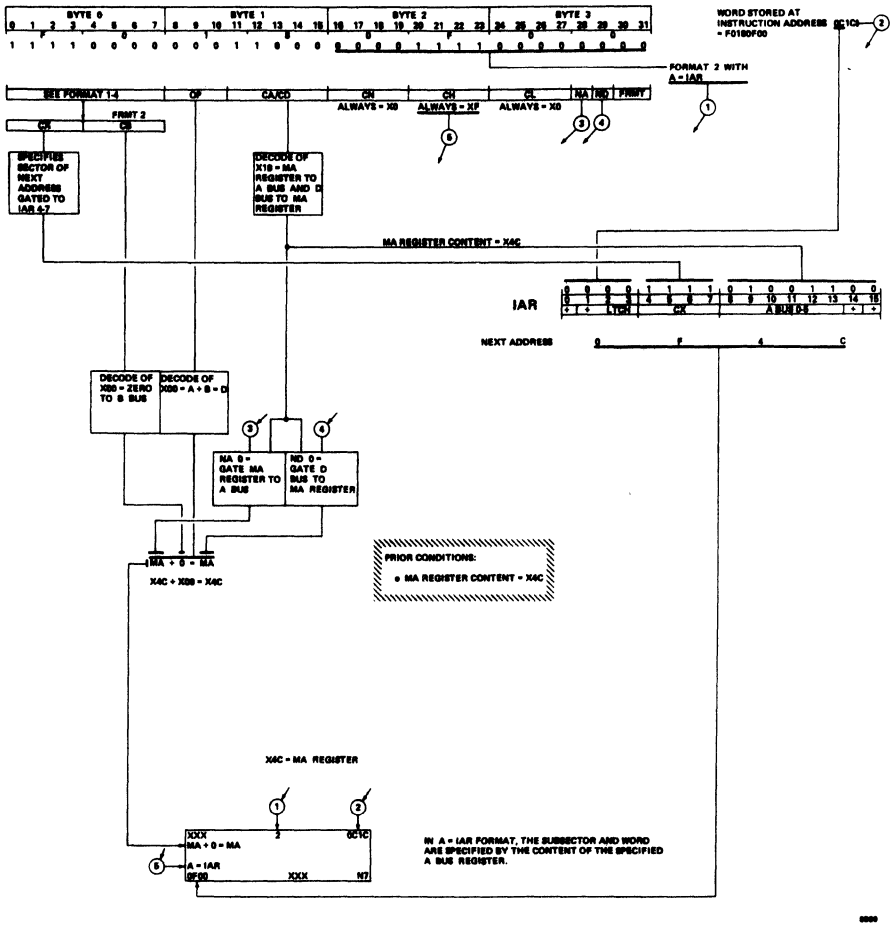


Figure 3-58. Format A=IAR Execution - Format 2

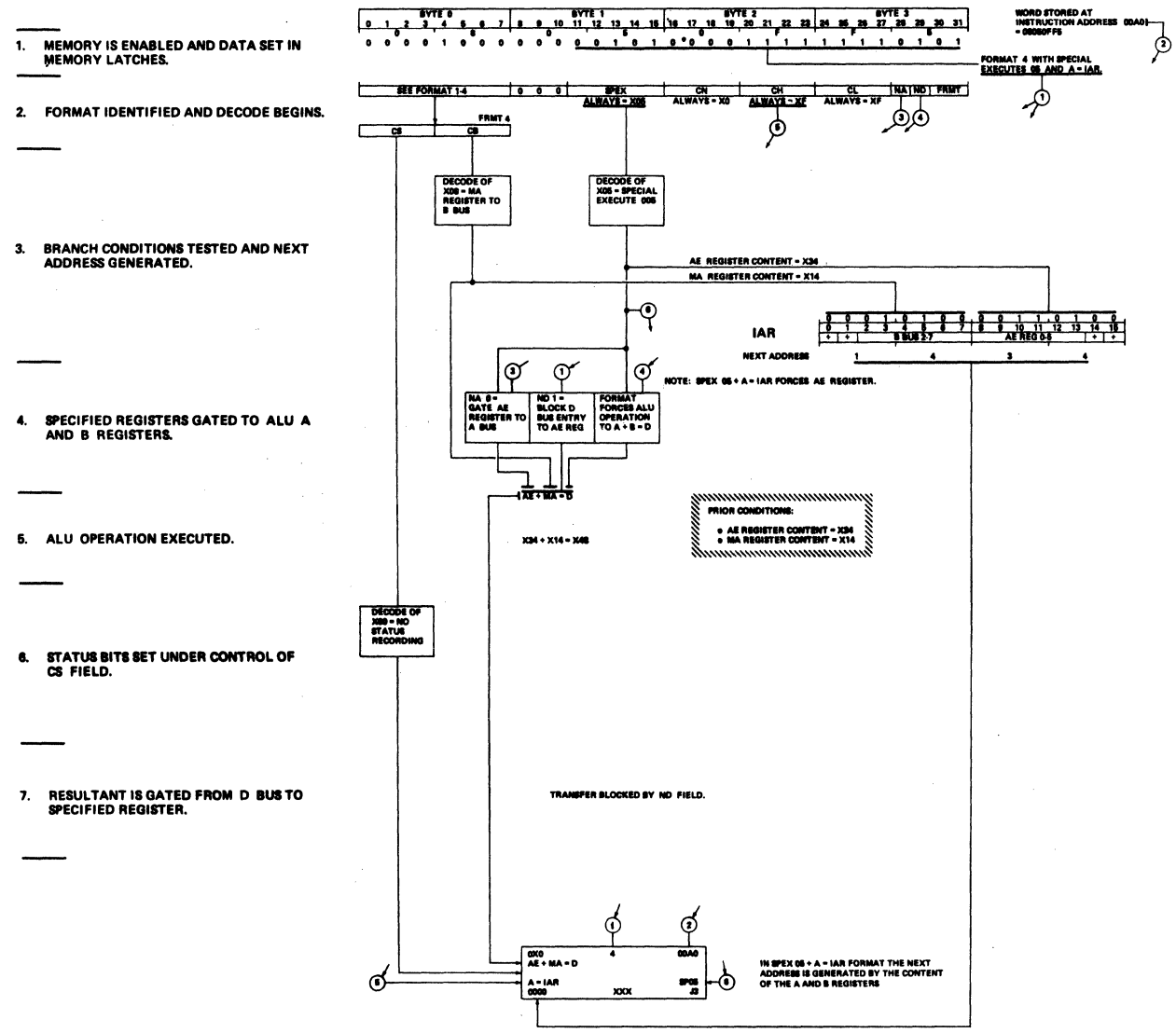


Figure 3-59. Format SPEX 05 + A=IAR Execution

Format SPEX 05 + A=IAR

Format SPEX 05+A=IAR is decoded when the hexadecimal code F appears both in the CH and the CL fields, instruction bits 20 through 27. Bits 30 and 31 then determine the interpretation of instruction bits 0 through 7. Bits 30 and 31 may specify formats 1, 2, 3, or 4. Since all four characters of the next address are gated from the A and B Buses, there is only one from for this instruction. Figure 3-59 illustrates the execution of this format. Normally the CN field is zero; however, if format 3 is specified, bit 19 of the instruction word is needed to control the gating of the four bit constant as explained under format 3. In this situation the CN field may contain hexadecimal 1.

This format is used to transfer the content of the specified B Bus source to the two high order characters of the Instruction Address register, and the content of the A Bus source to the two low order characters. It should be noted that IAR is set prior to the execution of the ALU operation.

The ALU operation is forced by format to an A+B=D situation and the A Bus source is forced to the AE register. The B Bus source depends on which of the four formats is specified. The B Bus source may be a specified register or a four or eight bit constant.

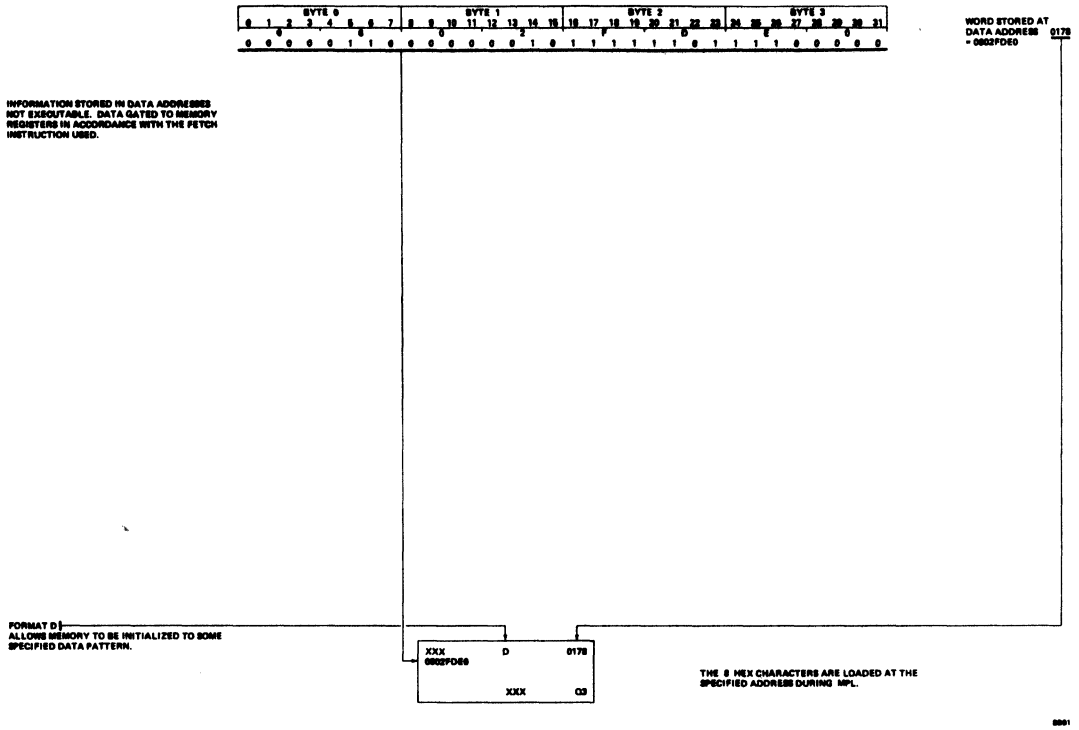


Figure 3-60. Format D

Format D

Format D (see Figure 3-60) is not an executable instruction. Format D is a data statement which allows memory to be initialized to a specified data pattern. The eight hexadecimal characters specified in the ALU statement portion of the microblock are loaded into memory at the current address location during microprogram load.

CORE MAP

The core map is an alphanumeric listing of all 4,096 addresses in memory. It is contained on pages 001 through 062 in Section 1 of the Microprogram Manual. The bit configuration of every instruction in the program is listed in hexadecimal format and cross referenced to its page location within the microprogram flow charts.

PAGE LAYOUT

Each page of the core map provides the microcode and cross reference listing for 200 addresses. The left-hand column on the page lists the basic address. In line on the page with the basic address are 12 columns. The first three columns list in order the microcode, flowchart page number, and microblock identification number for the instruction identified as word 0 of the basic address. The next three columns provide the same information for the instruction identified as word 4 of the basic address. The following three columns pertain to word 8, and the final three columns are for word C. Therefore, to look up the microcode for the word stored at address 0318, find address 0310 in the left-hand column and read across the page to the seventh column.

MICROCODE

There is a microcode listing for every address in the program even though there are some addresses which are not used. This is because the unused addresses are loaded with special execute 15, which is a program stop. This prevents an accidental execution of an unused address. If there is no page and identification number next to a listing of special execute 15 (000F00F4), it is filling an unused address. If page and identification number are listed, that address is actually used as a program stop by the hardcore tests.

FLOW DIAGRAMS

The flow diagrams are symbolic representation of the entire microprogram. This material explains how to read and interpret the symbology. Material covered includes: page layout, edgeconnectors, microblock interpretation.

PAGE LAYOUT

Each page of the program has a four character identification number. The pages are arranged in alphanumeric sequence with page A000 being the first page in Section 2 of the Microprogram Manual, following the MPSS Cross Reference Revision Summary pages. The page numbering is not necessarily sequential. There are many unused page numbers, allowing for necessary revisions to the manual. The MPSS Cross Reference Revision Summary lists all the pages of the flow diagrams along with the current revision level of each page.

In order to be able to locate individual instructions on the page, a grid system is used. Each microblock has a two character identification number which locates it on the page. A full page of instructions would have 80 microblocks on it, although this is a rate occurrance. The block in the upper left-hand corner of the page is identified as A0, and the block in the upper right-hand corner is A7. This is known as the A row. There are 10 rows on the page, each having eight blocks numbered 0 through 7. The 10 rows, from top to bottom are identified A, C, E, G, J, L, N, Q, S, and U. The block identification number is printed in the lower right-hand corner of each microblock.

EDGE CONNECTORS

The microblocks are connected together with dashed lines, known as nets. These nets indicate the flow

of instructions. When a net leaves or enters a page, it does so through an edge connector. This is simply a cross-reference system used to identify where a net is coming from, and where it is going. The edge connector is made up of the four character page number, followed by a period and the two character microblock identifier. The microblock identifier portion is always the number of the block that originated the net. However, the page number changes. The edge connector on a net leaving a page, carries the destination page number. For example, B020Q3 is coming from microblock Q3 and is going to page B020. But the edge connector entering the page, carries the originating page number. To use the same example, the edge connector entering page B020 reads B030.Q3. This is the same edge connector that left page B030 reading B020.Q3. By using this system it is always possible to find the origin and destination of all nets.

MICROBLOCK INTERPRETATION

The microblock layout is illustrated in Figure 3-61. Figures 3-45 through 3-60 are helpful in understanding the relationship between the instruction word, the microblock, and the functions executed. In order to be able to interpret the microprogram it is necessary to understand each entry in the microblock.

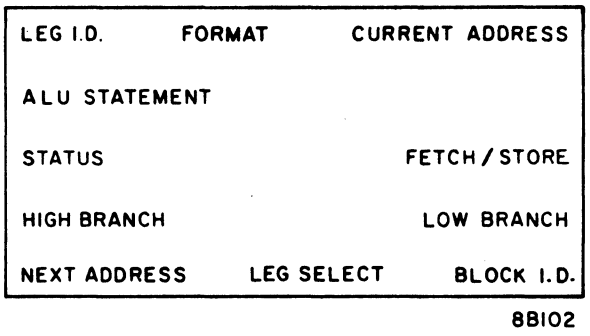


Figure 3-61. Microblock Layout

Leg ID

The Leg ID is defined as – a three bit field corresponding to current address bits 11, 12, and 13. Characters 0, 1, and X are valid for each position (X indicates don't care). The Leg ID assists the reader by expediting identification of the proper address within a branch set for a given state of conditions being tested. This is redundant information. Information contained in the Leg ID field can be easily discerned from the current address field.

Format

This one or two character field defines the format of the instruction. It should be noted that only the standard instruction identification is contained in this field. Special case instructions are defined by appropriate entries in the high or low branch fields. Content of the format field is defined by instruction bits 28 through 31.

Current Address

This is a four hexadecimal field defining this instruction address in memory. In all cases, except SPEX 05 and SPEX 05+A=IAR, the current address has an affect on the next address. Except in the noted cases the first or first and second character of the next address are the same as (latched to) the corresponding characters of the current address.

ALU Statement

This field defines: the A and B Bus sources that are fed to the Arithmetic Logic Unit, the operation to be performed by the unit, and the D Bus destination for the resultant of the operation. Control of the ALU operation and the ALU sources and destination are a function of word format. Refer to Figures 3-45 through 3-60 and the accompanying description of the individual formats, for information pertaining to the ALU execution. Table 3-15, Decode Assignments,

lists all of the possible ALU operations. The character C on the left-hand side of the equation indicates a carry-in, derived from ST register bit 3 (ST3), must be taken into account in the execution of the equation. The character C on the right-hand side of the equation indicates that the state of the ALU carry-out is to be stored in ST3. The numeral 1 on the left-hand side of the equation indicates that the ALU carry-in is forced to a 1.

Status

The status field is controlled by the instruction word CS field. It is used to indicate the arbitrary setting or resetting of a particular ST register bit. This then sets up conditions which are later tested to determine branching conditions.

Fetch/Store

The Fetch/Store field is a three character field specifying the three lower order characters of DAR. This field is used in a fetch or store instruction to indicate the memory address that data is being fetched from or stored in. In a one byte fetch instruction the three characters are controlled by the instruction words CW,CV, and CY fields respectively. The three characters represent the sector, subsector, and word of the data address. In a four byte fetch or store instruction, the last two characters are listed as X's. This is because these characters are controlled by the content of the A Bus. In order to determine the correct subsector and word, it is necessary to know the content of the A Bus, prior to the execution of the indicated ALU operation.

High Branch

The High Branch field is used to specify either a high order branching condition, or that special case format A=IAR is active. Information in this field is controlled by the instruction word CH field. If the High Branch field is left blank, it indicates that the microprogrammer has arbitrarily set the instruction CH field to either 0 or 1. This controls IAR bit 12, and is reflected in the fourth hexadecimal character of the next address. If one of the CH field branch conditions (refer to Table 3-15) is specified, it indicates that that condition is to be tested. If the condition being tested is true, IAR bit 12 is set to a 1. If the condition being tested is not true, IAR bit 12 is set to a 0. Therefore, in order to interpret this field it is necessary to know the state of the condition being tested. Referring to Figure 3-52, the High Branch condition specified is ST4, and the next address field indicates that the lowest possible next address is 2834. Therefore, since the prior conditions indicate that bit 4 of the ST register is turned on (1), bit 12 of IAR is going to be set and the next address is set to 283C.

Low Branch

The Low Branch field is used to specify either a low order branching condition, or that special case format SPEX is active. Information in this field is controlled by the instruction word CL field. If the Low Branch field is left blank, it indicates that the microprogrammer has arbitrarily set the instruction CL field to either 0 or 1. This controls IAR bit 13, and is reflected in the fourth hexadecimal character of the next address. If one of the CL field branch conditions (refer to Table 3-15) is specified, it indicates that that condition is to be tested. If the condition being tested is true, IAR bit 13 is set to a 1. If the condition being tested is not true, IAR bit 13 is set to 0. Therefore, in order to interpret this field it is necessary to know the state of the condition being tested. Referring to Figure 3-51, the Low Branch condition specified is ST3, and the next address field indicates that the lowest possible next address is 3008. Therefore, since the prior conditions indicate the bit 3 of the ST register is turned on (1), bit 13 of IAR is going to be set and the next address is set to 300C.

Next Address

The Next Address field is a four character hexadecimal field specifying the next address to be executed if branching conditions are not specified. If branching conditions are specified, then the Next Address field specifies the lowest possible next address. That is, the address which will be executed if the branching conditions being tested are not true.

Leg Select

The Leg Select is defined as a three bit field representing next address bits 11, 12, and 13. The characters 0, 1, and X are valid for each position (X indicates don't care). The character * is also allowed in the second and third positions to indicate a branch condition for the respective next address bit positions. Information in this field is redundant to the information provided in the High and Low Branch and Next Address fields.

Block ID

The Block ID is a two character alphanumeric code that identifies the location of the microblock on the flow diagram page. For an explanation of the grid system used, refer to the Page Layout heading under Flow Diagrams.

PART 3

MICROPROGRAM FLOWCHARTS

INTRODUCTION

This part contains flowcharts of typical functional microprogram operations. The flowcharts (Figures 3-62 through 3-72) enable the user to follow the sequence of events that occur while analyzing the detailed flowcharts in the Microprogram Manual.

The flowcharts are arranged in the same sequence that a typical chain of channel commands could be issued. Each flowchart stands by itself, that is, it is not necessary to reference between figure numbers to understand command execution. Sheet 1 of each figure provides an overall functional flow of the entire operation; subsequent sheets show details of the operation.

Keep the following concepts in mind while following these charts:

- All operations are preceded by initial selection. The polling loop is contained within initial selection. The loop is expanded in Figure 3-64.
- All charts assume successful completion of the command. The charts not only show the normal execution path, but they also show error testing and branching. Error analysis is illustrated in the sense routine (Figure 3-72).
- No attempt is made to show how sense bytes and error codes are generated.
- The Set File Mask command (Figure 3-65) must precede any other command.
- Except for the initial selection and polling loop routines, go to the End Procedure (Figure 3-71) for an explanation of the final events in successful completion of any command.
- Microprogram Manual page numbers and block identifiers are indicated adjacent to each event in the flowcharts.

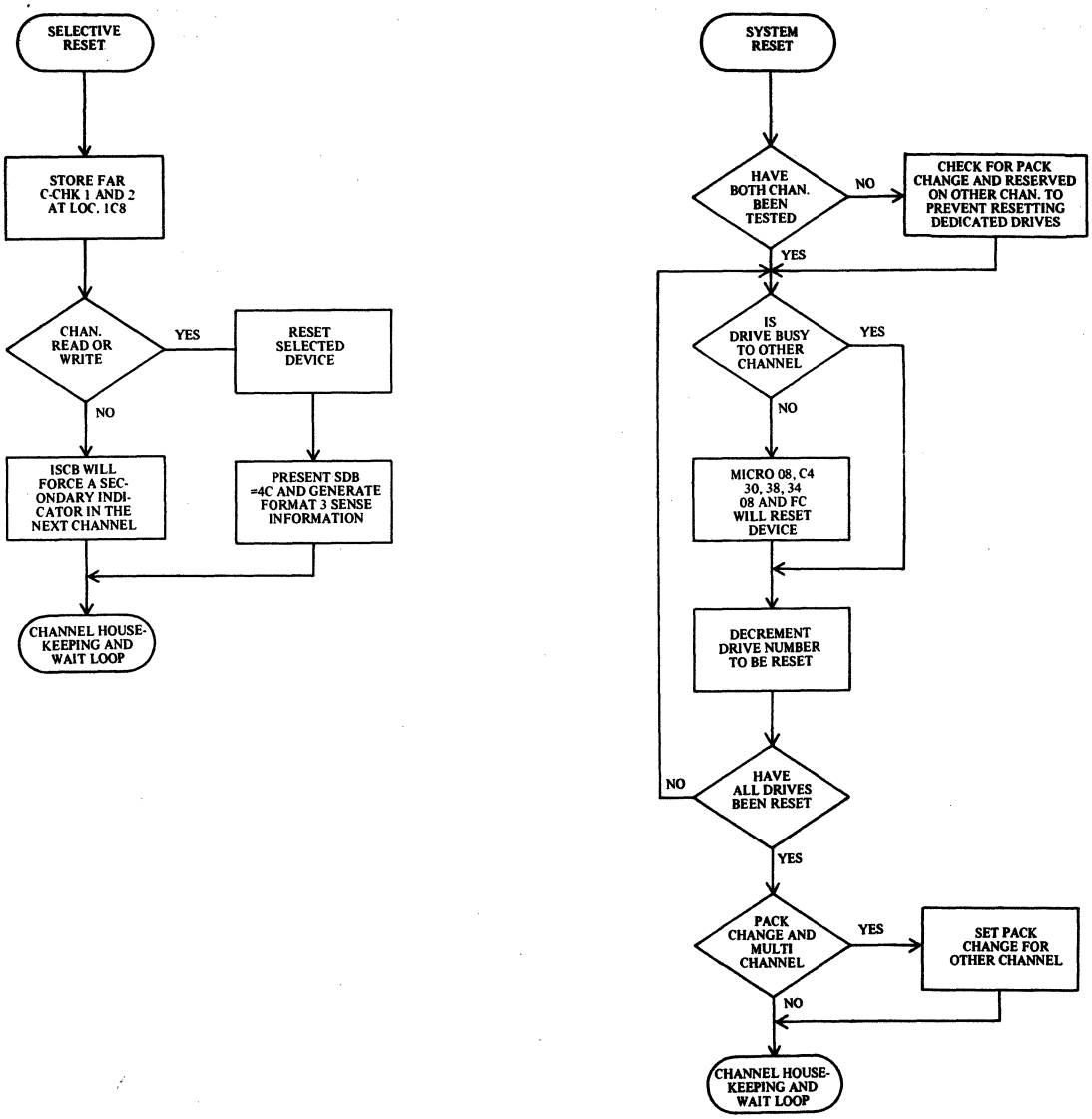
SELECTIVE RESET

When the system actuates Selective Reset, the IAR is forced to Hex address 0004. This address is the start of a store routine which will save Hard Check 1 and 2 as well as the Failing Address Register, which contains the address of the last executed instruction. If channel Read/Write is set, the selected device is reset.

SYSTEM RESET

When the system actuates System Reset, the IAR is forced to Hex address 0000. A General Reset of either channel, whether selected or not, releases reservations related to the resetting channel and does not affect reservations of the other channel. Reset of a channel also resets any pending Attention Interrupts due to Seek Completes.

GENERAL FLOWCHARTS



8B107-1

Figure 3-62. General Reset (Sheet 1 of 5)

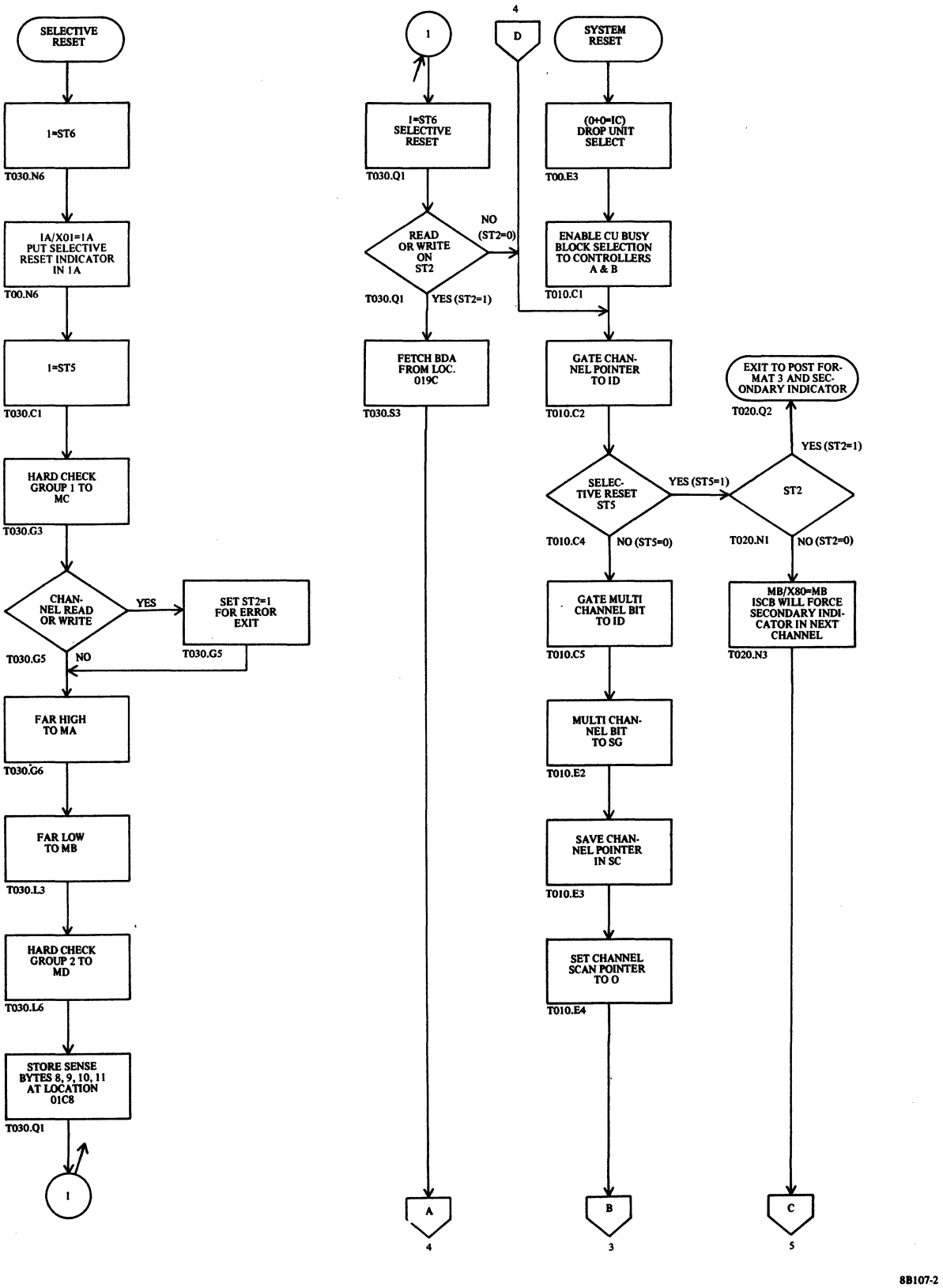


Figure 3-62. General Reset (Sheet 2 of 5)

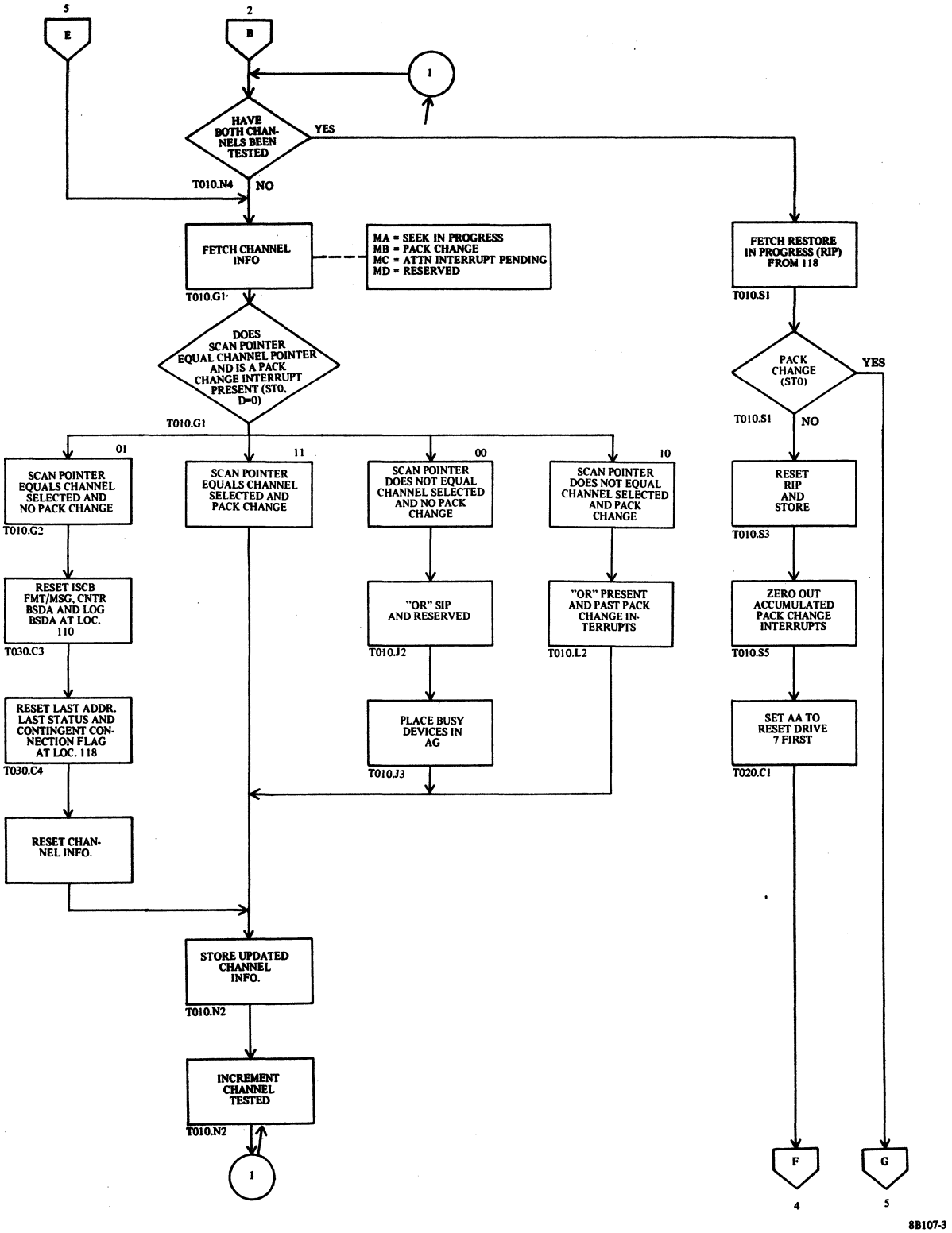


Figure 3-62. General Reset (Sheet 3 of 5)

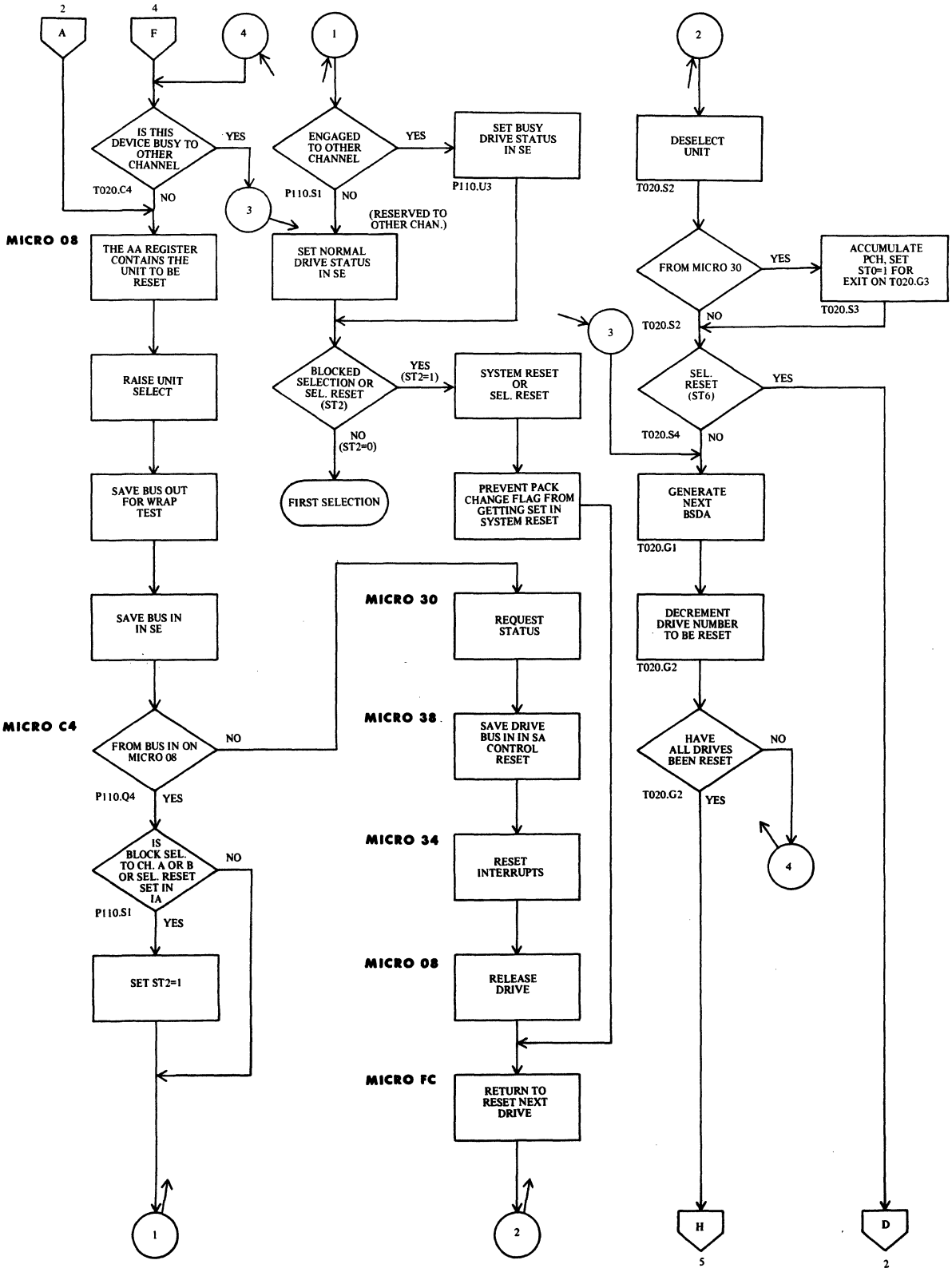
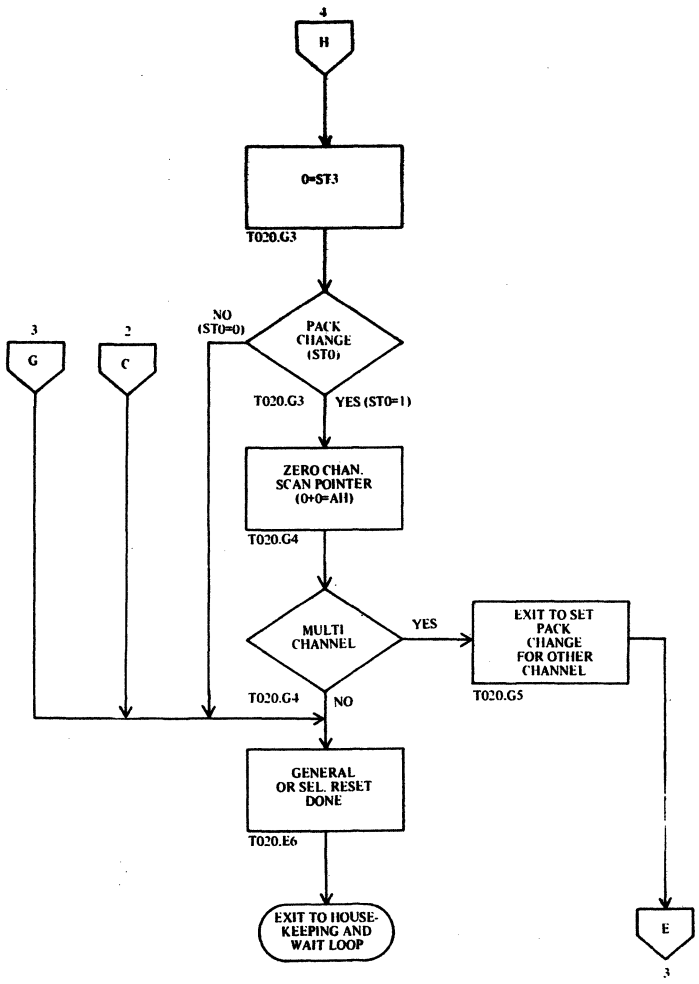


Figure 3-62. General Reset (Sheet 4 of 5)



8B107-5

Figure 3-62. General Reset (Sheet 5 of 5)

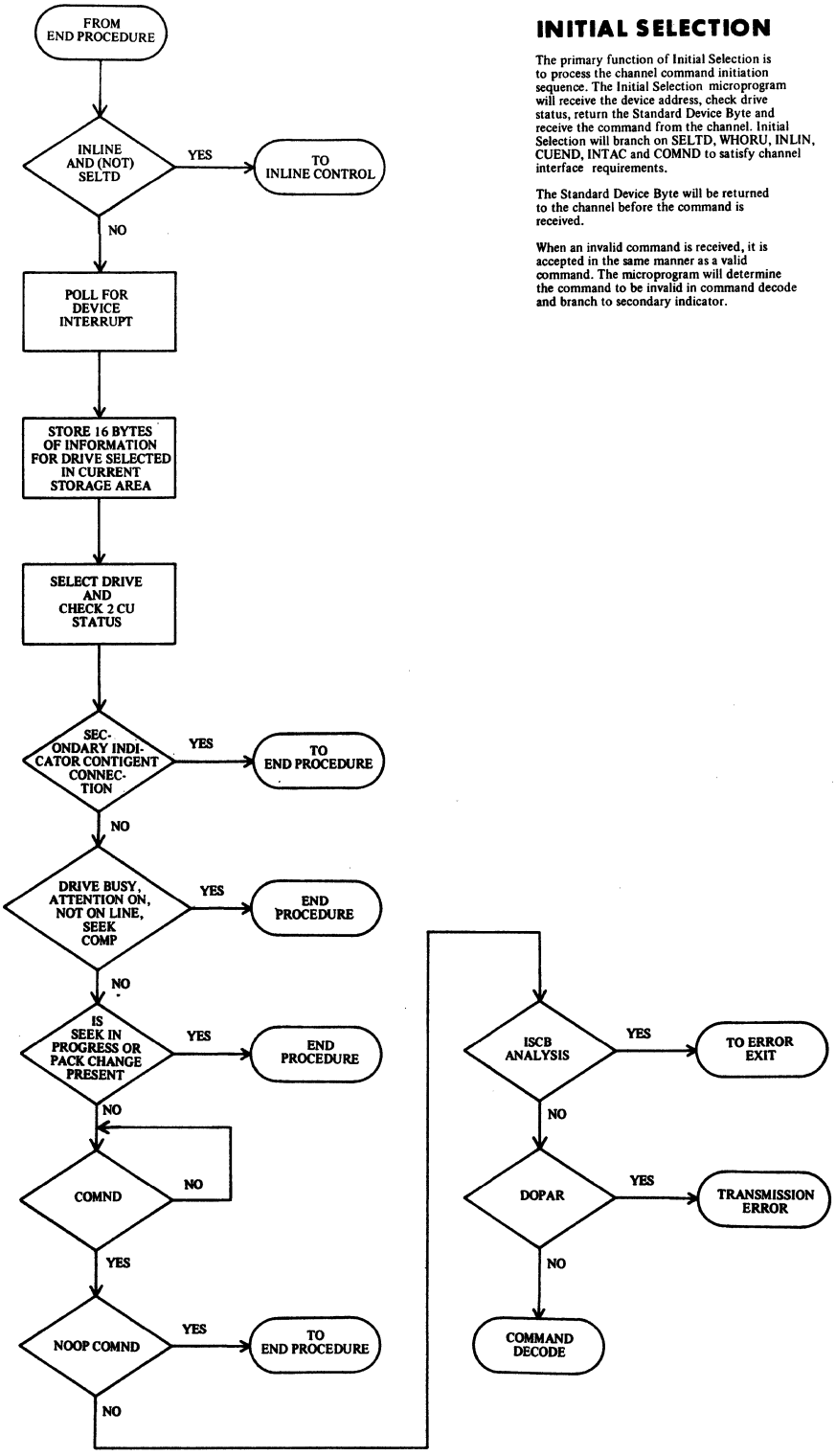
INITIAL SELECTION

The primary function of Initial Selection is to process the channel command initiation sequence. The Initial Selection microprogram will receive the device address, check drive status, return the Standard Device Byte and receive the command from the channel. Initial Selection will branch on SELTD, WHORU, INLIN, CUEND, INTAC and COMND to satisfy channel interface requirements.

The Standard Device Byte will be returned to the channel before the command is received.

When an invalid command is received, it is accepted in the same manner as a valid command. The microprogram will determine the command to be invalid in command decode and branch to secondary indicator.

GENERAL FLOWCHART



8B109-1

Figure 3-63. Initial Selection (Sheet 1 of 8)

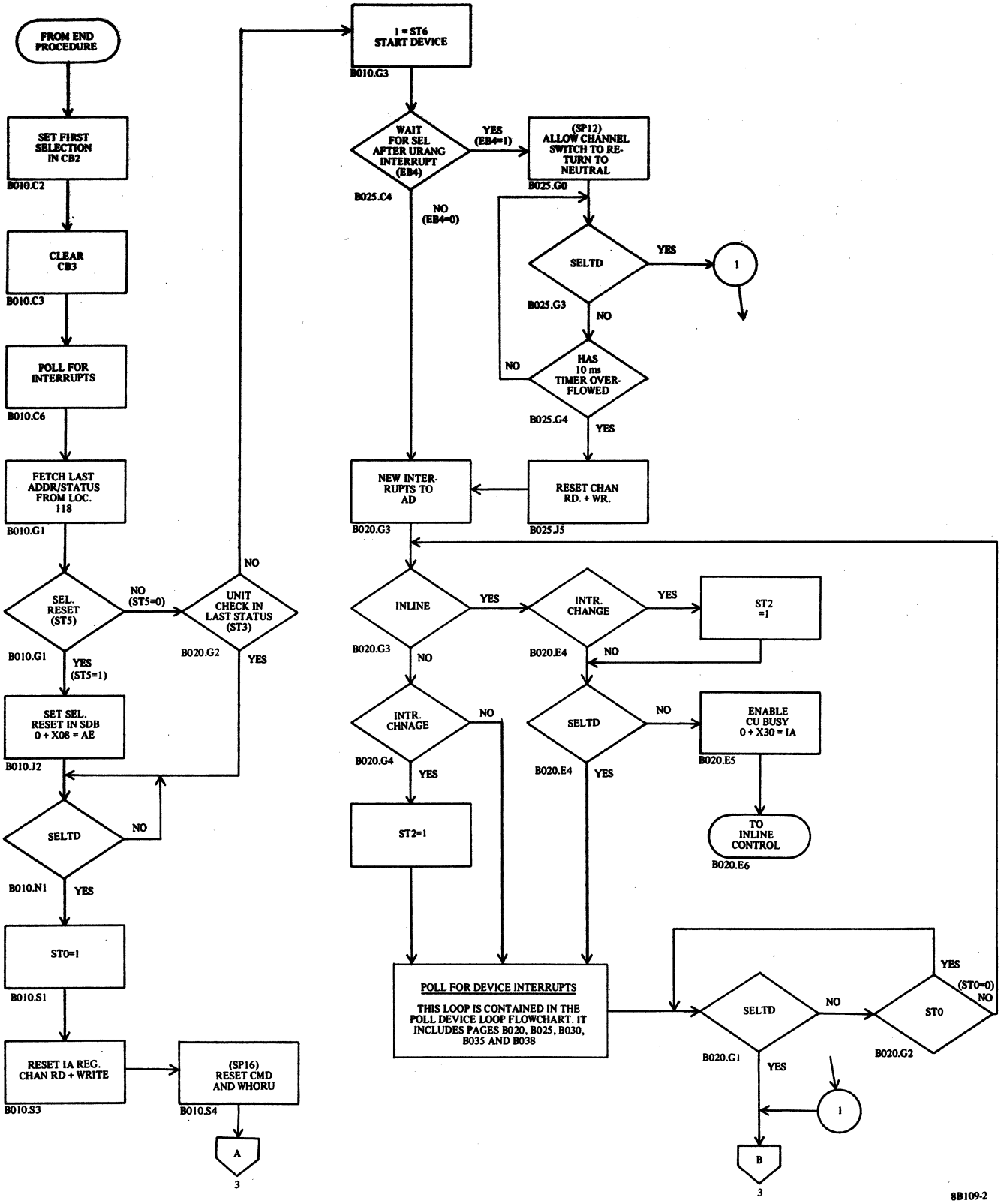


Figure 3-63. Initial Selection (Sheet 2 of 8)

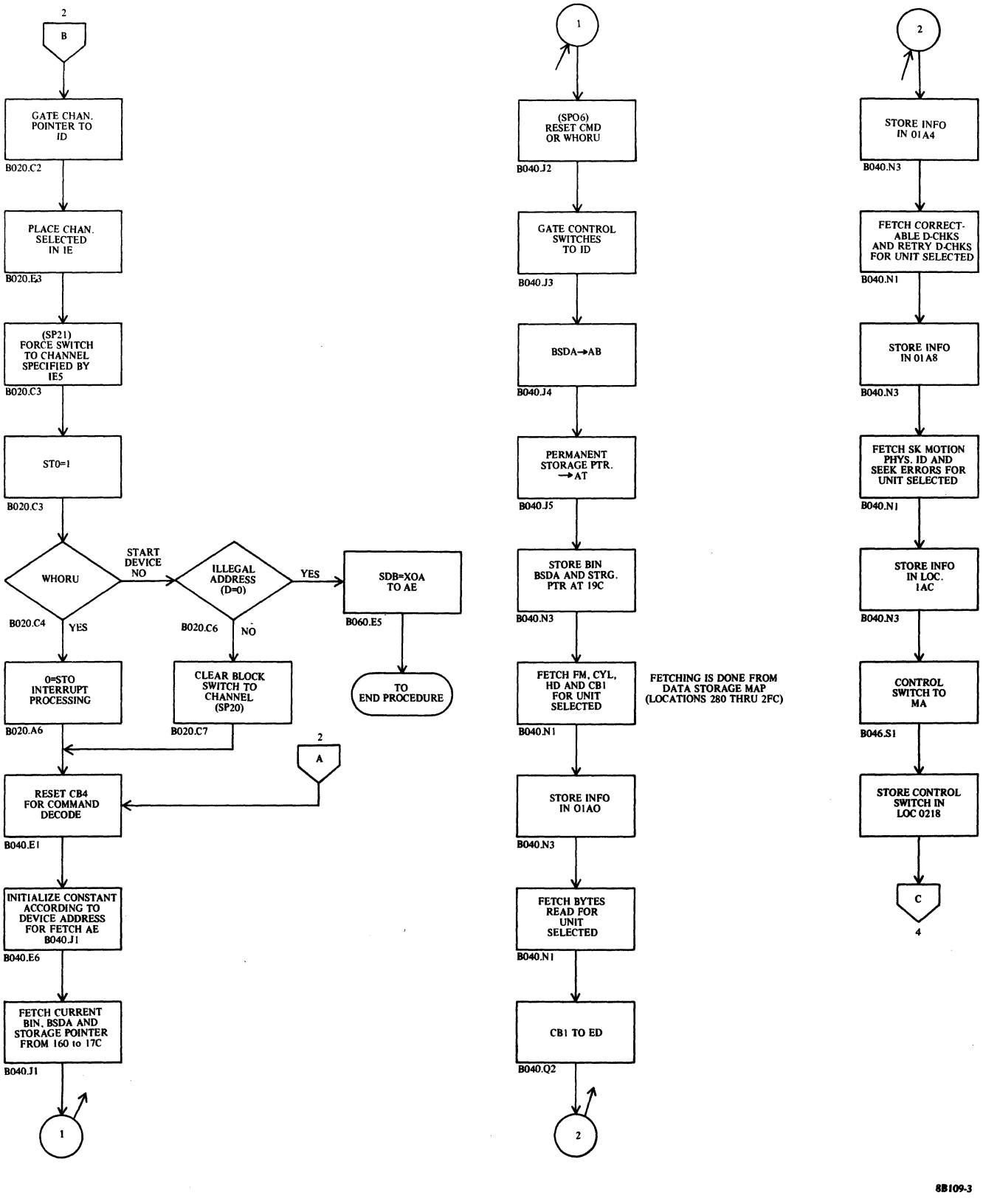


Figure 3-63. Initial Selection (Sheet 3 of 8)

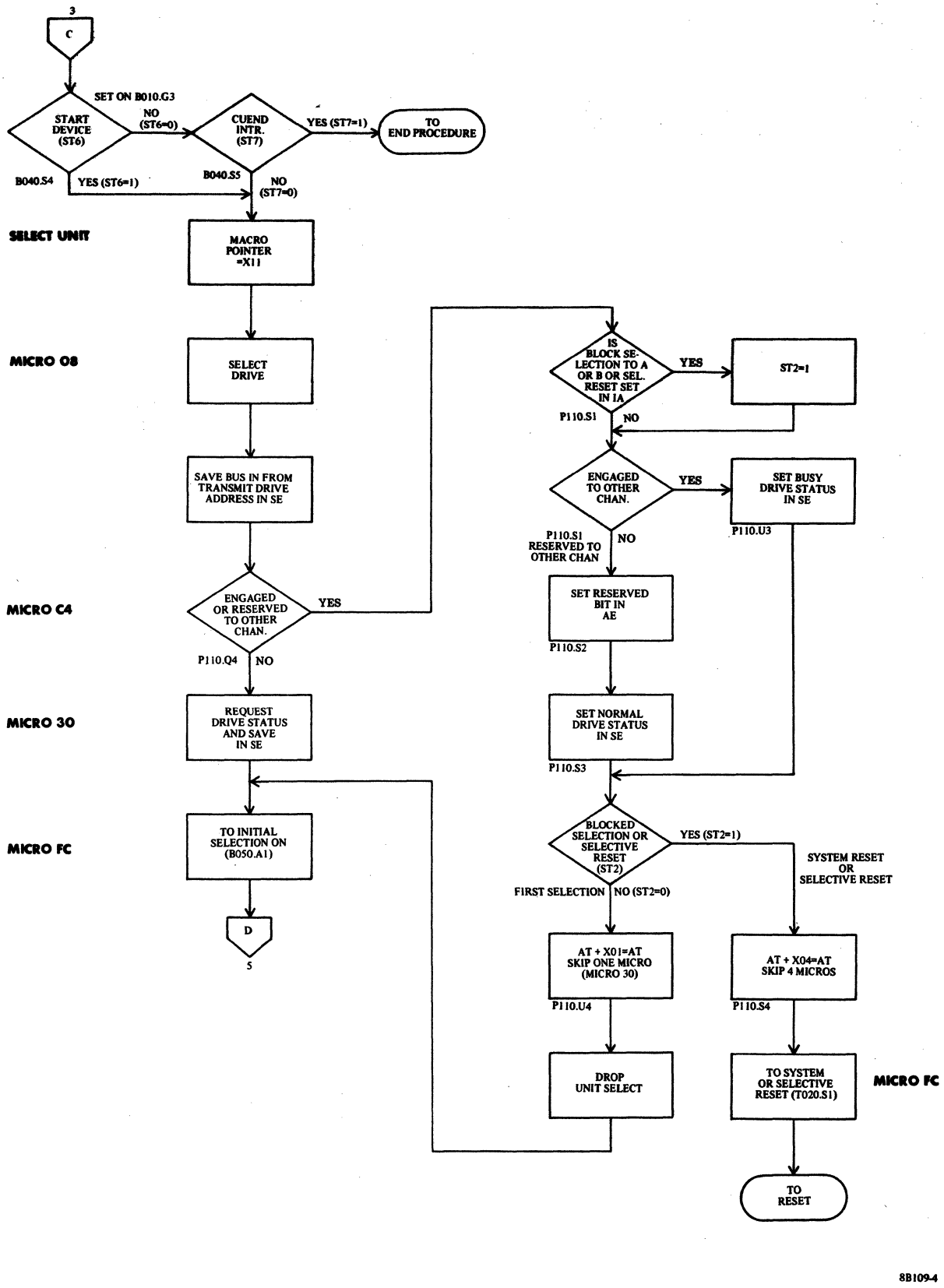


Figure 3-63. Initial Selection (Sheet 4 of 8)

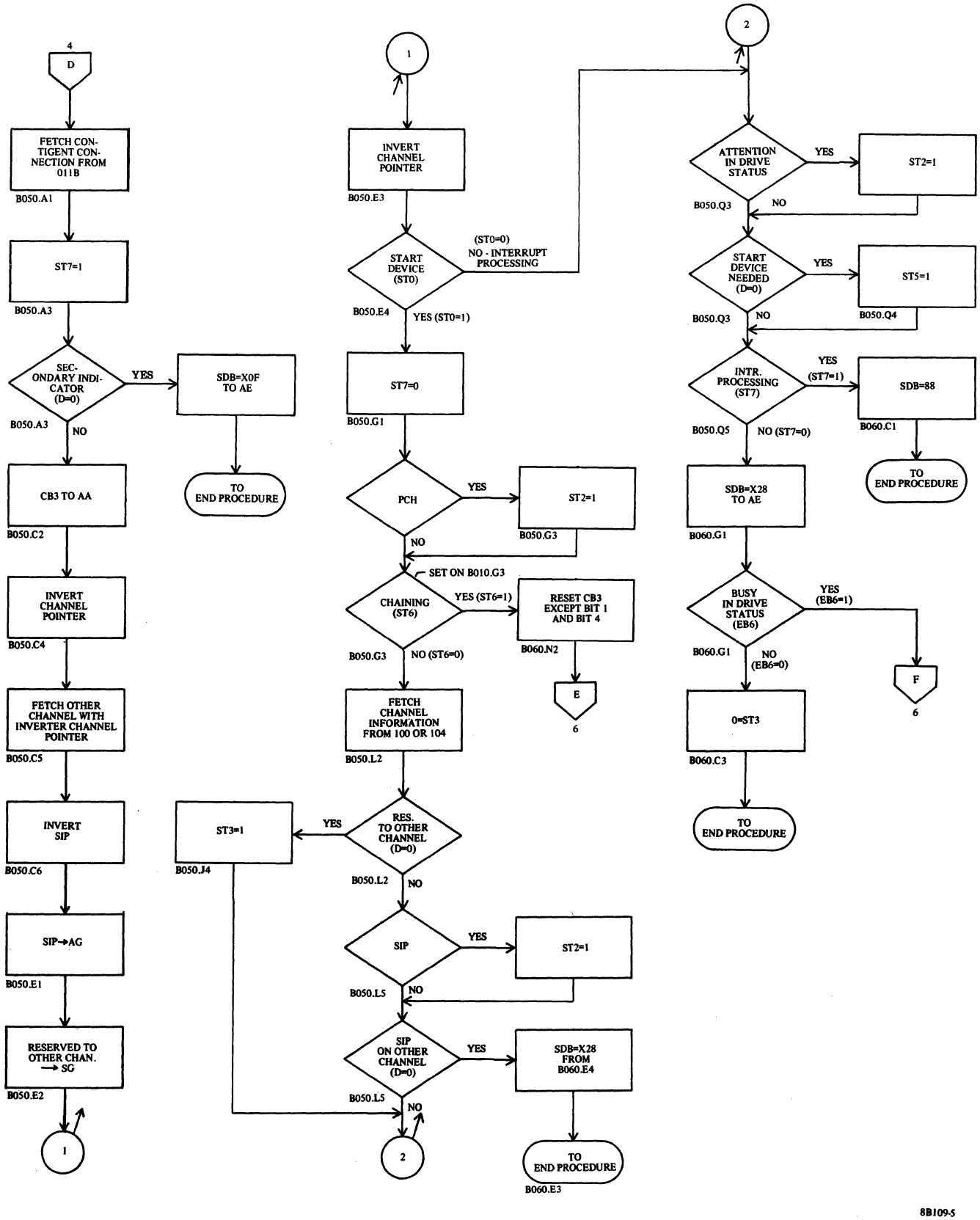


Figure 3-63. Initial Selection (Sheet 5 of 8)

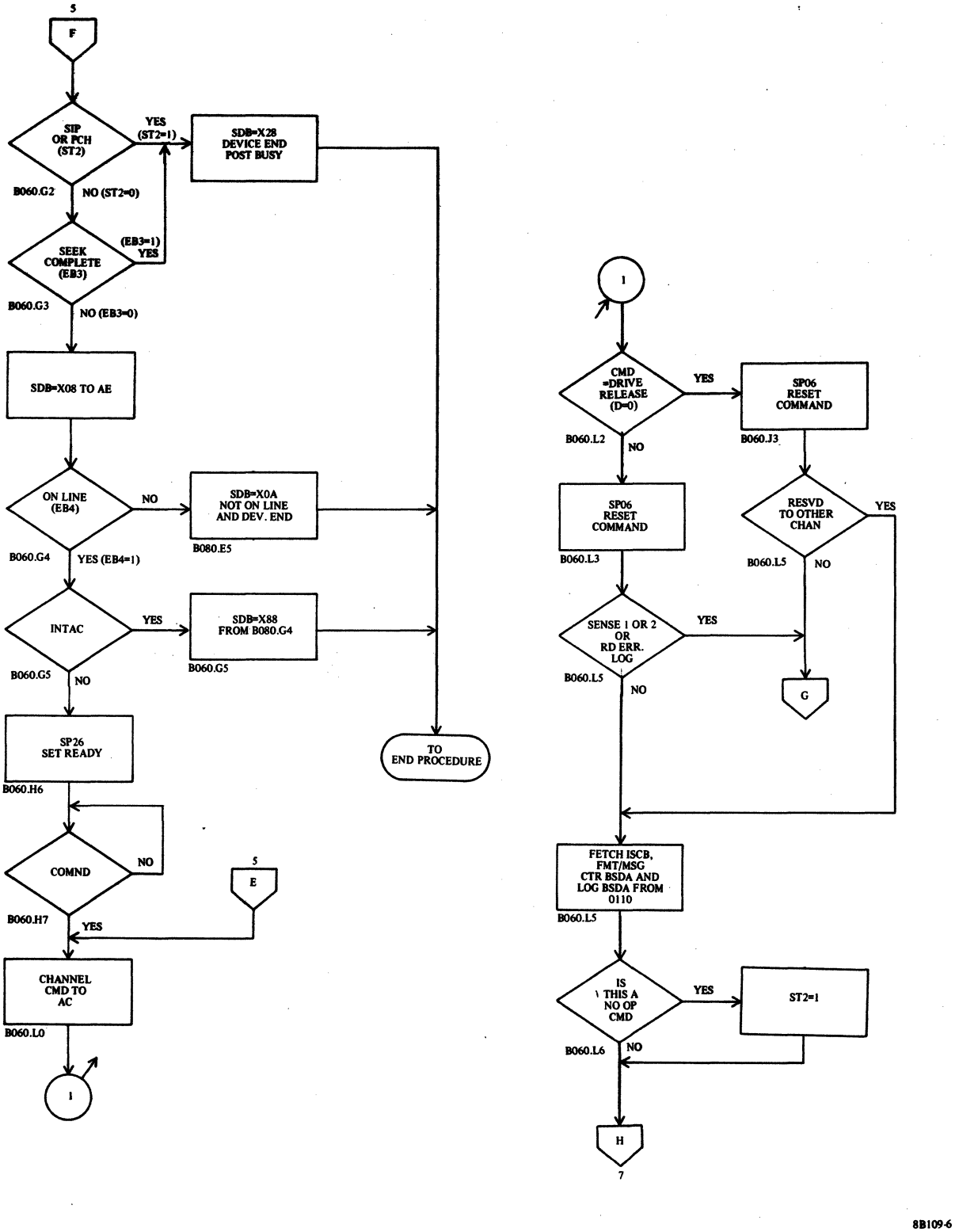


Figure 3-63. Initial Selection (Sheet 6 of 8)

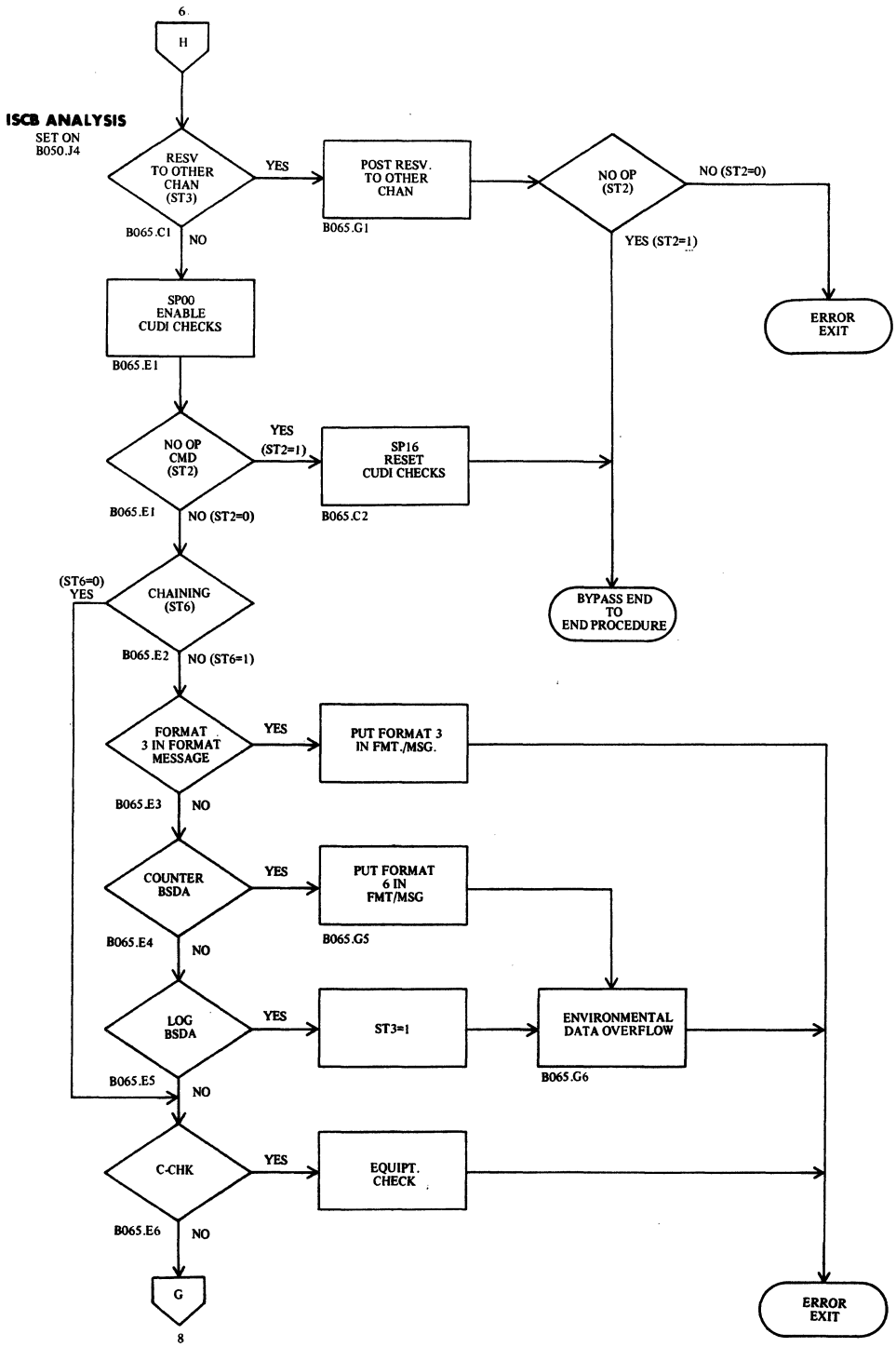


Figure 36 3. Initial Selection (Sheet 7 of 8)

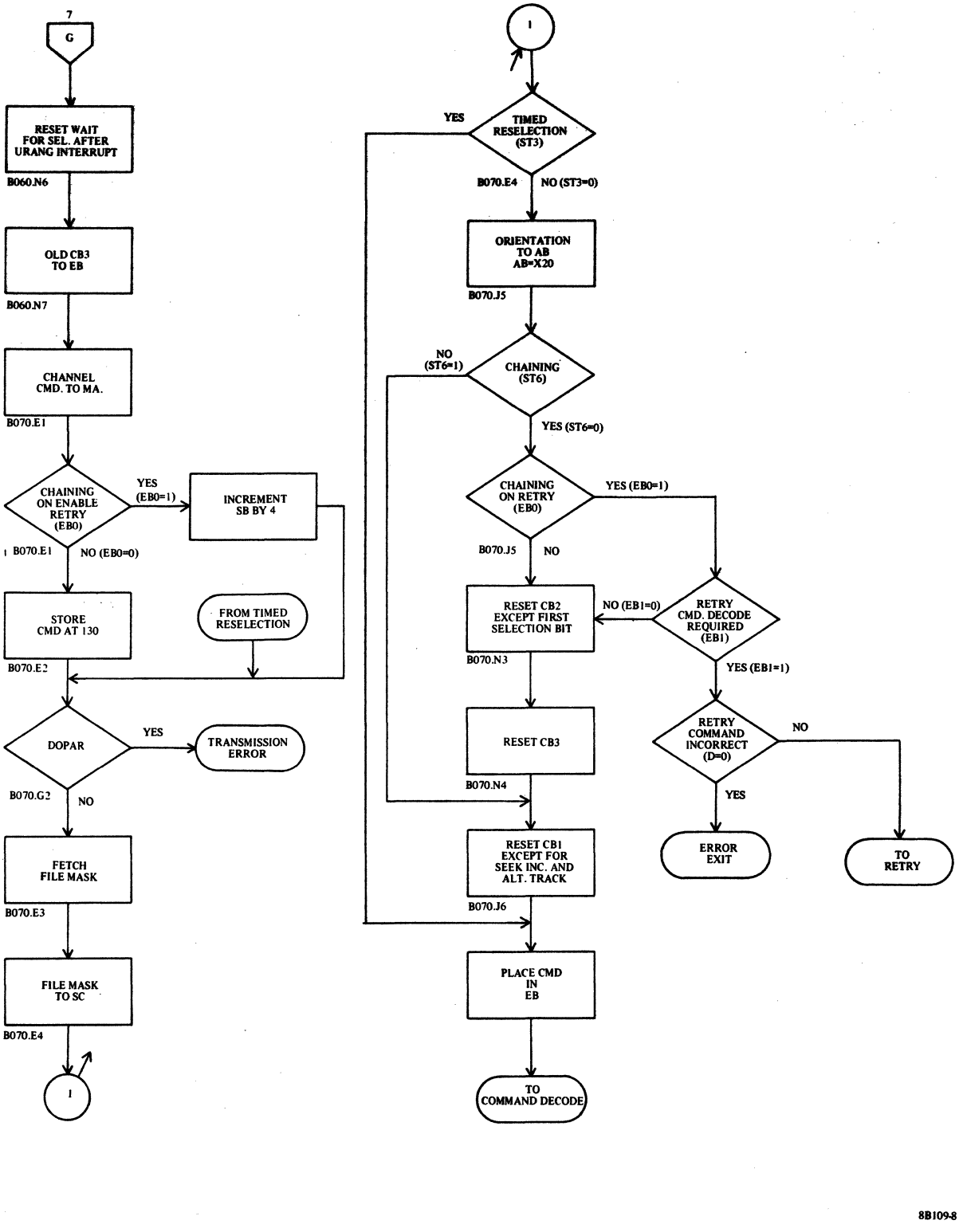


Figure 3-63. Initial Selection (Sheet 8 of 8)

POLL FOR INTERRUPTS

This loop monitors device interrupts. If an interrupt changes, the interrupts are compared against each channel's interrupt steering mask to determine if channel interrupts are to be set for each respective channel. The listed equations define the setting.

/ = "OR" . = "AND"
X = Channel specified by IE5
Y = Other Channel

SIP(X) = SEEK IN PROGRESS ON CHANNEL X
RES(X) = RESERVED ON CHANNEL X
NBSY = NOT BUSY TO ANY CHANNEL
NBSY(X) = NOT BUSY TO CHANNEL X
INTD = INTERRUPT FROM DRIVE
INTC = INTERRUPT TO CHANNEL
PCH(X) = ATTENTION INTERRUPT OWED TO CHANNEL X (NOT DEDICATED TO CHANNEL Y)

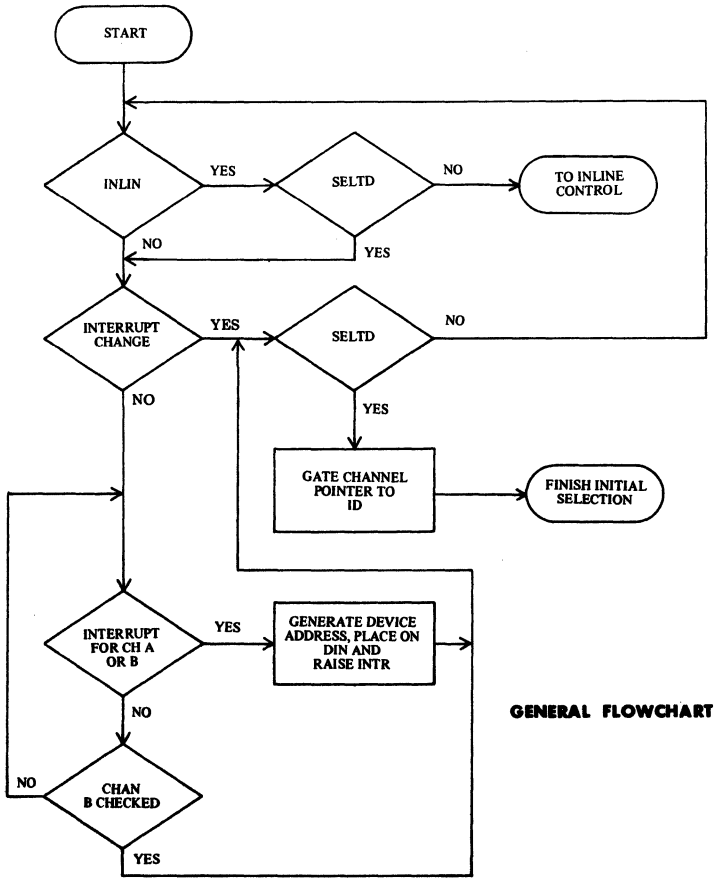
$$INTC(X) = NBSY(X) \cdot [INTD/PCH(X)/CUEND]$$

WHERE

NBSY(X) = SIP(X)/RES(X)/NBSY
AND
NBSY = N[SIP(X)/RES(X)/SIP(Y)/RES(X)]

Reducing + eliminating impossible states
NBSY(X) = N[SIP(Y)/RES(Y)]

Impossible states
SIP(X) · SIP(Y) = 1
RES(X) · RES(Y) = 1



GENERAL FLOWCHART

8B108-1

Figure 3-64. Polling Loop (Sheet 1 of 6)

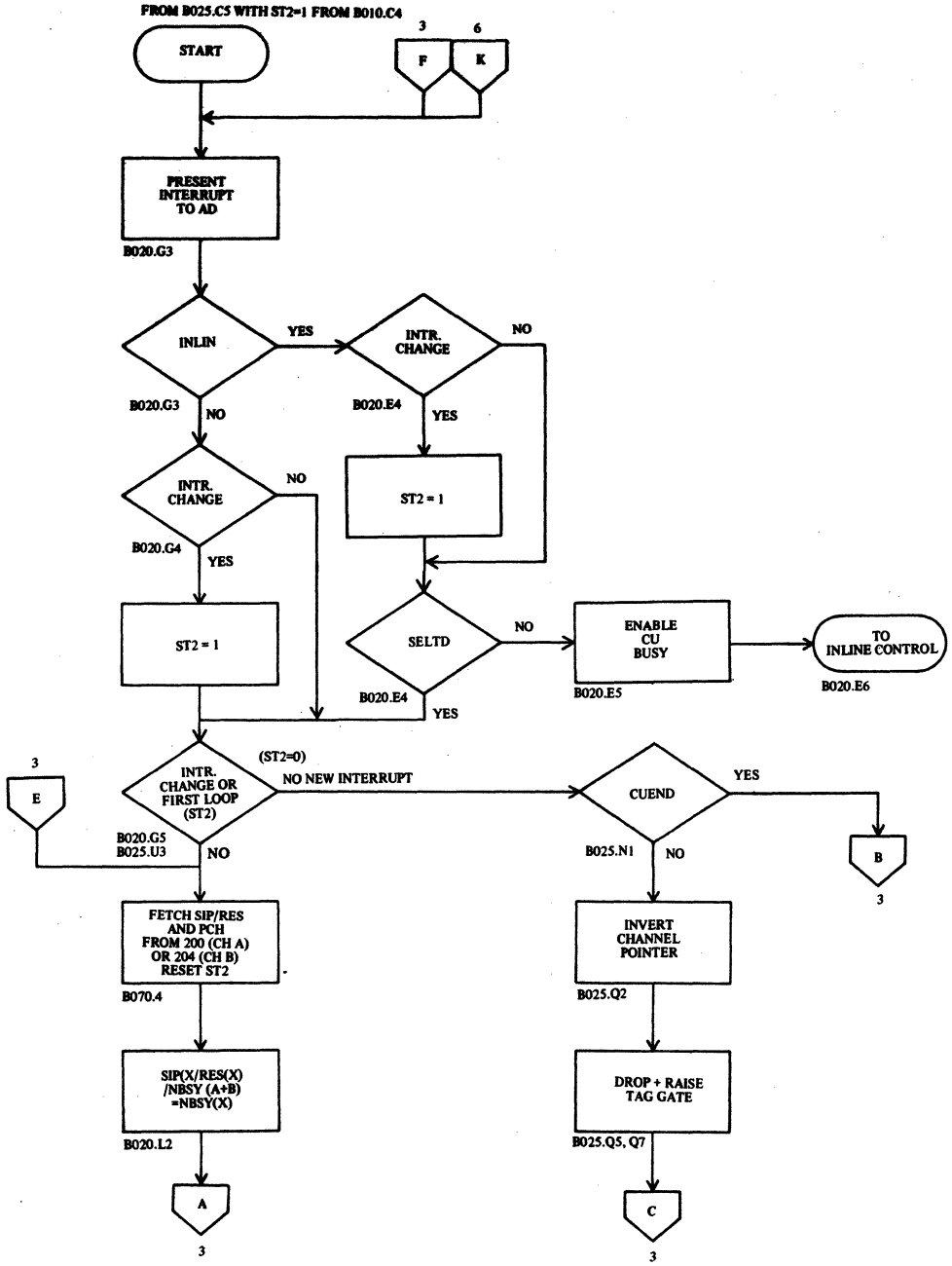


Figure 3-64. Polling Loop (Sheet 2 of 6)

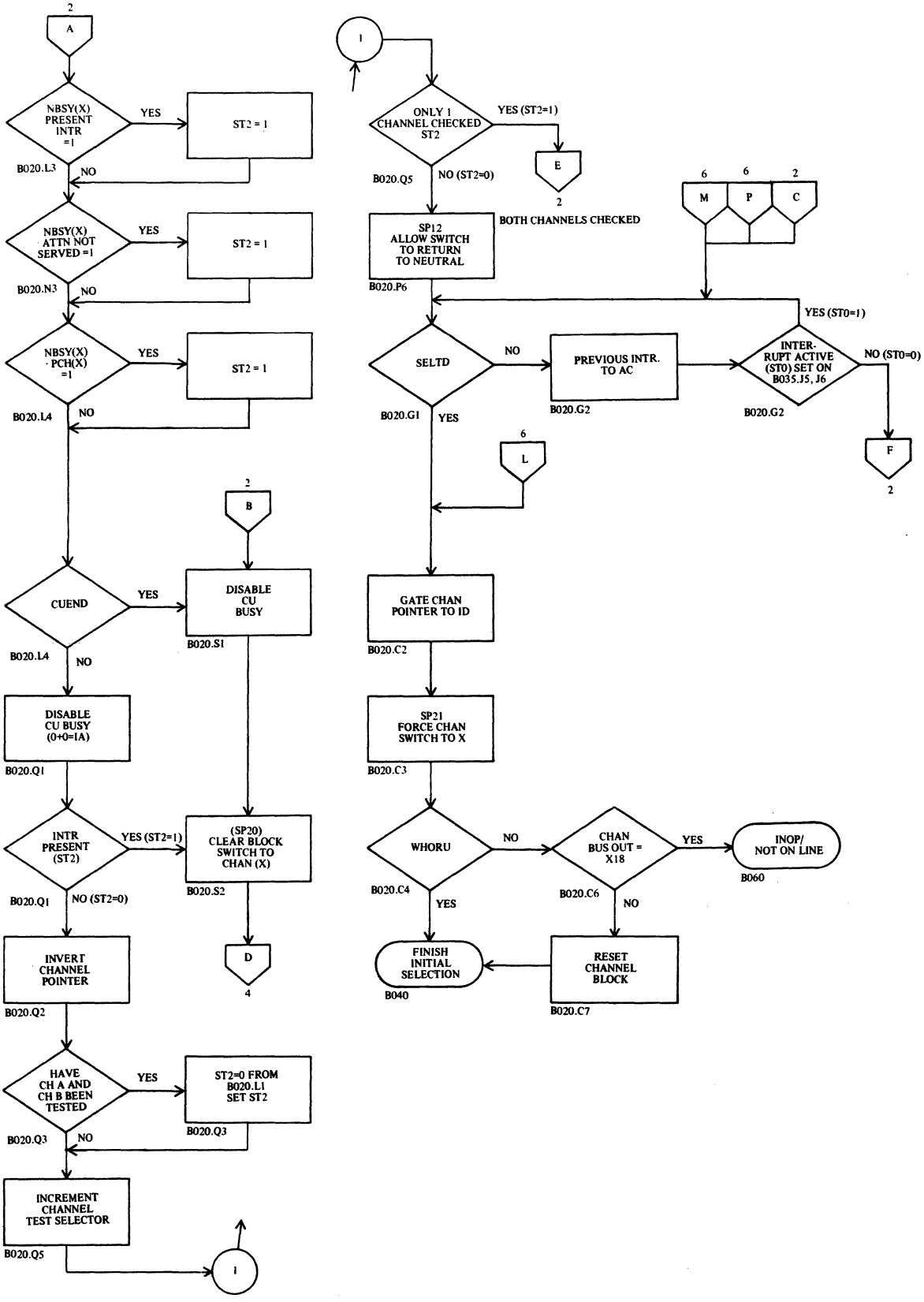


Figure 3-64. Polling Loop (Sheet 3 of 6)

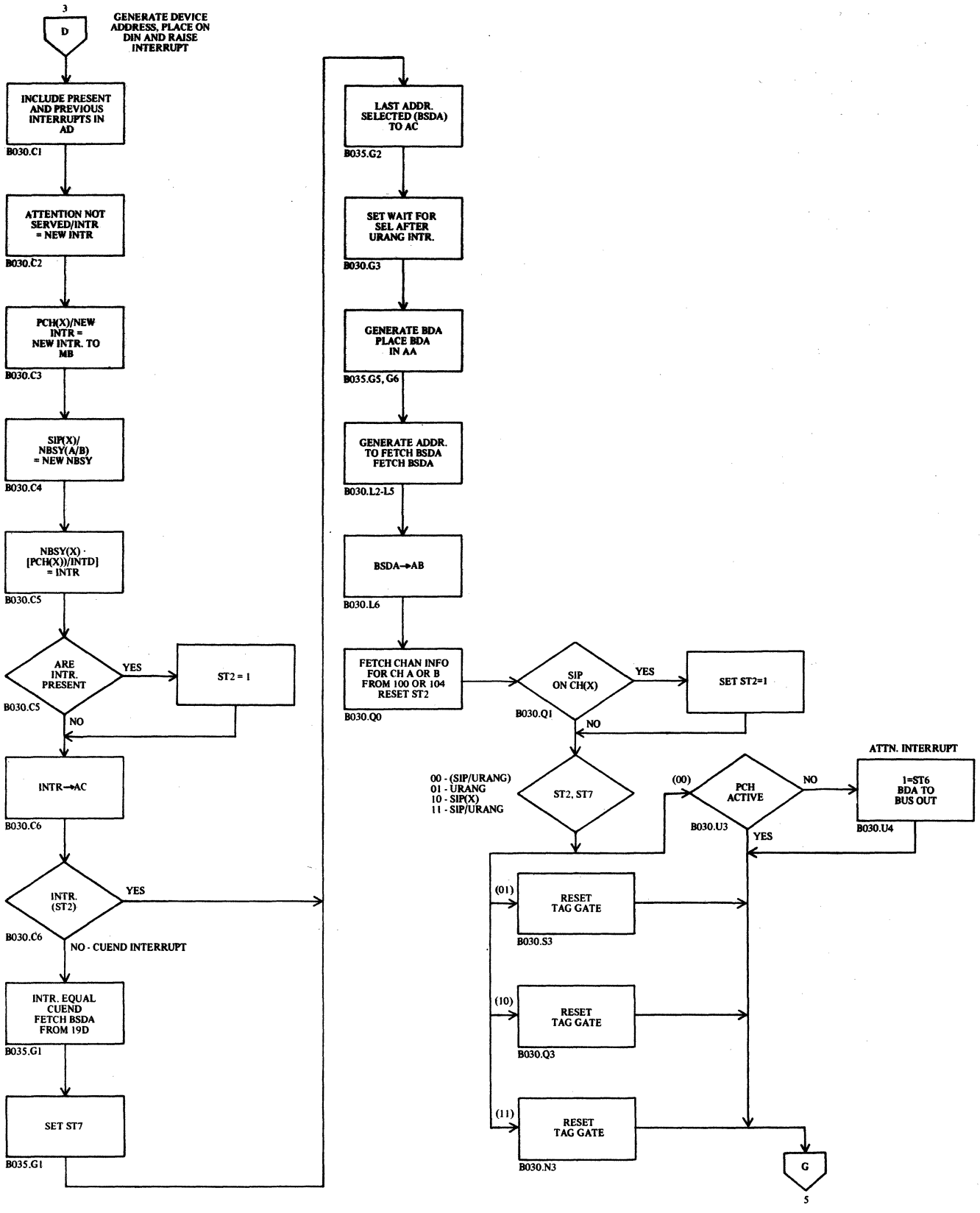


Figure 3-64. Polling Loop (Sheet 4 of 6)

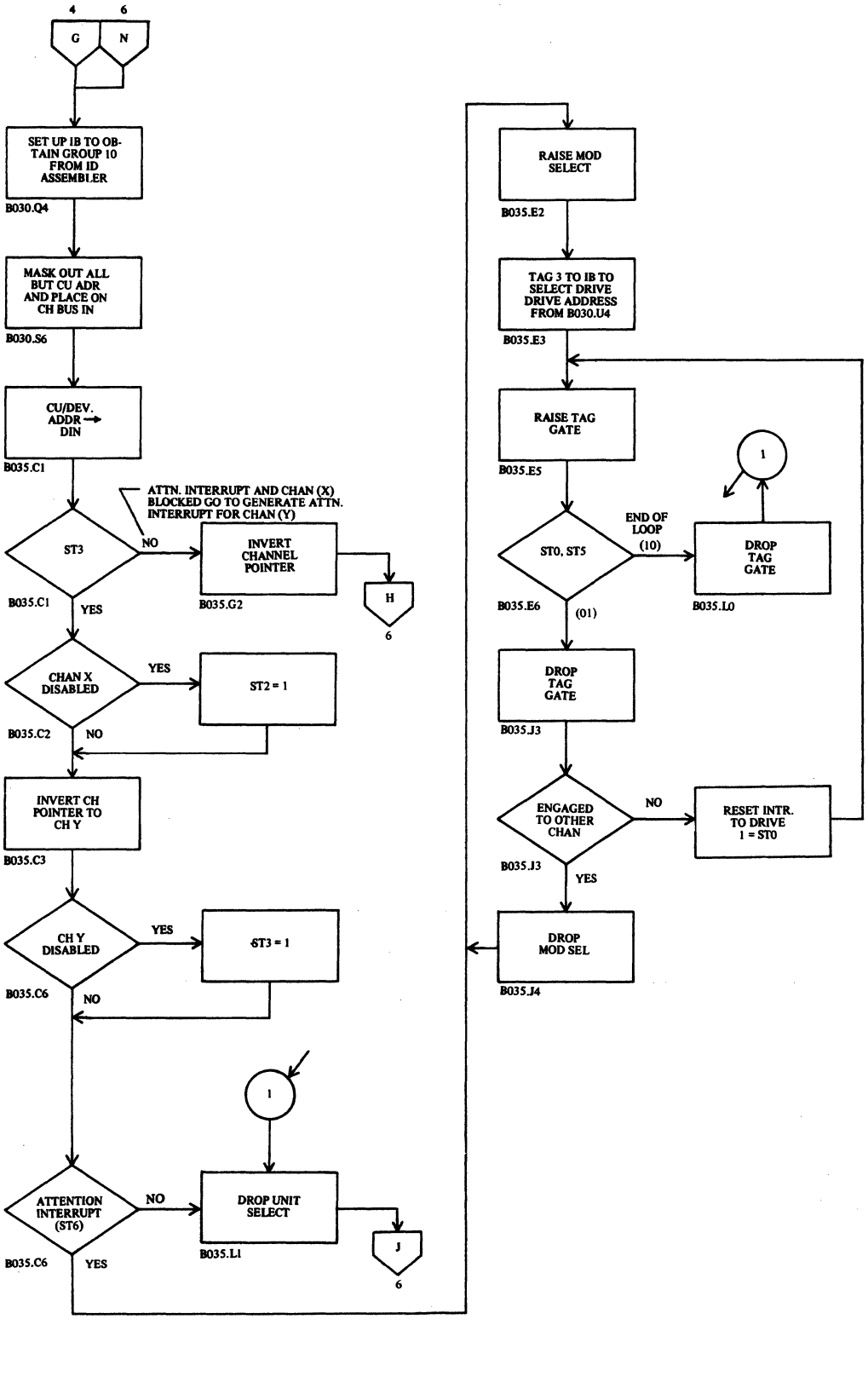


Figure 3-64. Polling Loop (Sheet 5 of 6)

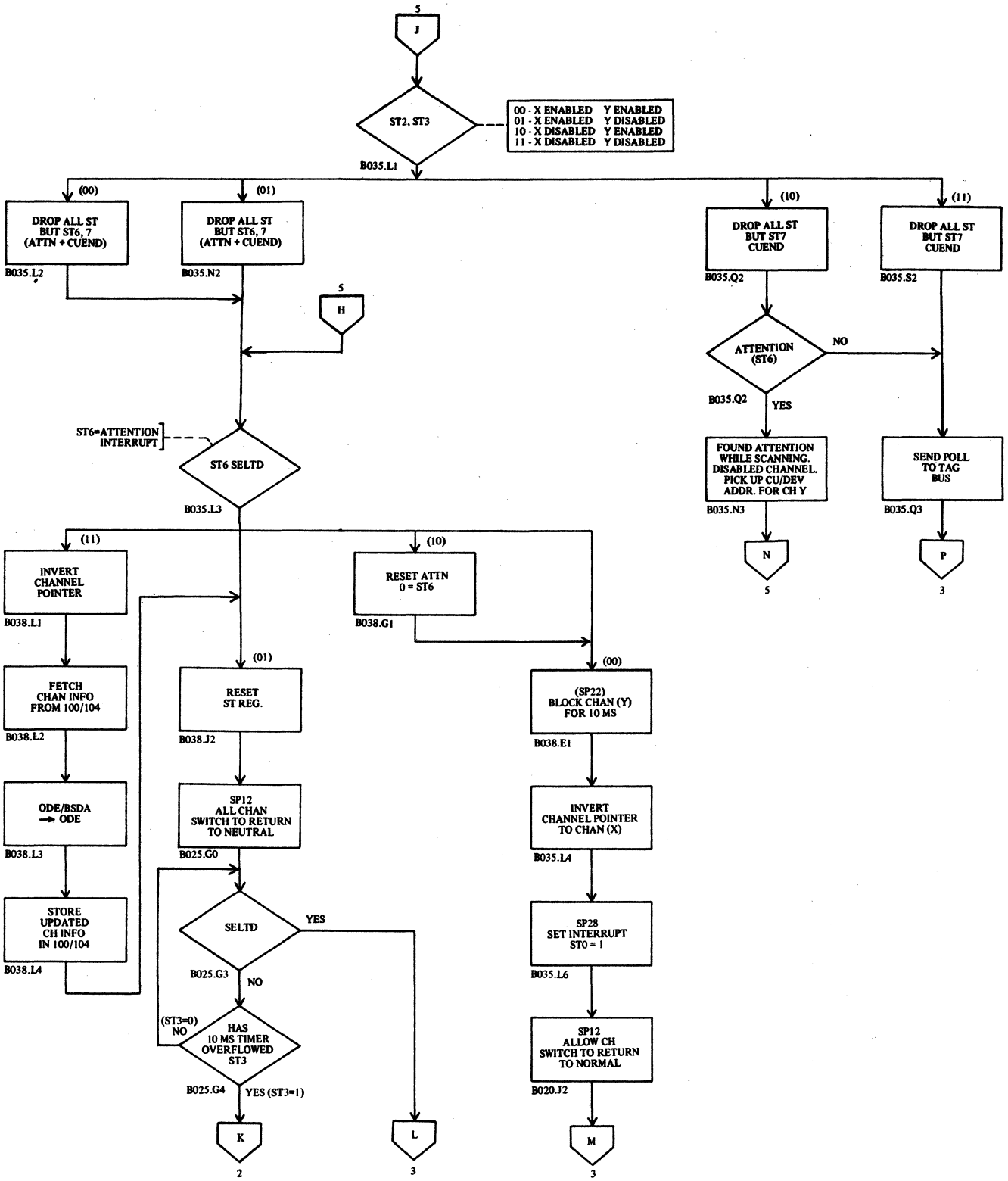
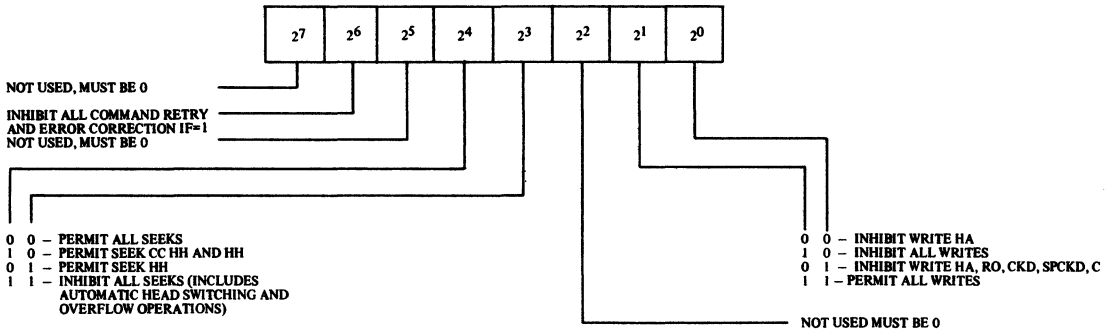


Figure 3-64. Polling Loop (Sheet 6 of 6)

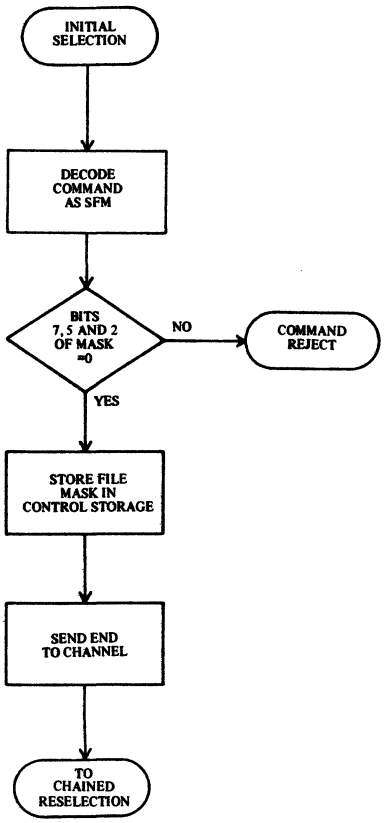
SET FILE MASK

SET FILE MASK

The Set File Mask command causes a byte of data to be transferred from CPU memory to the controller. At the completion of the transfer, End is signalled. The byte of data transferred describes the write and seek operations that can be performed.



GENERAL FLOWCHART



8B110-1

Figure 3-65. Set File Mask (Sheet 1 of 5)

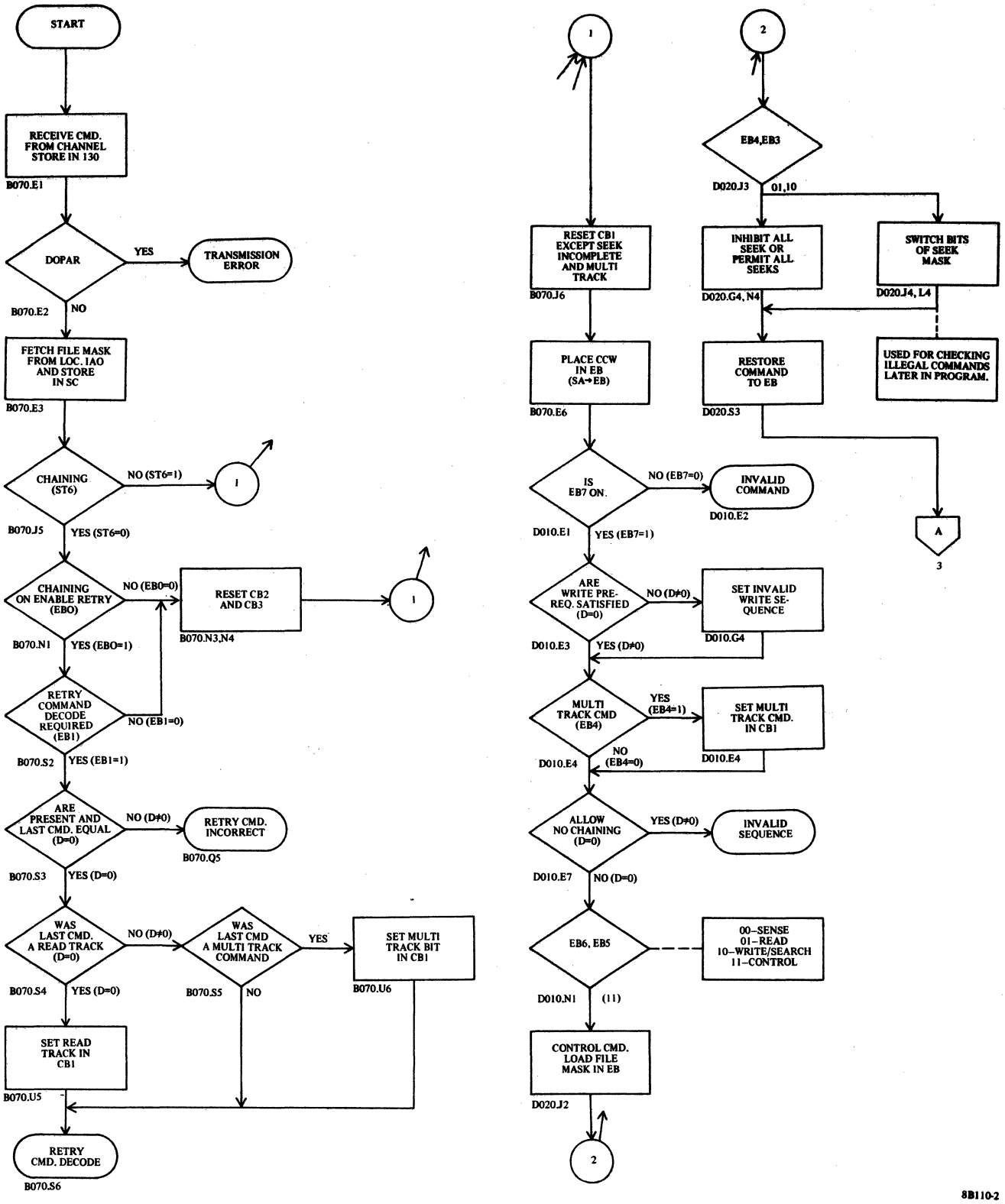


Figure 3-65. Set File Mask (Sheet 2 of 6)

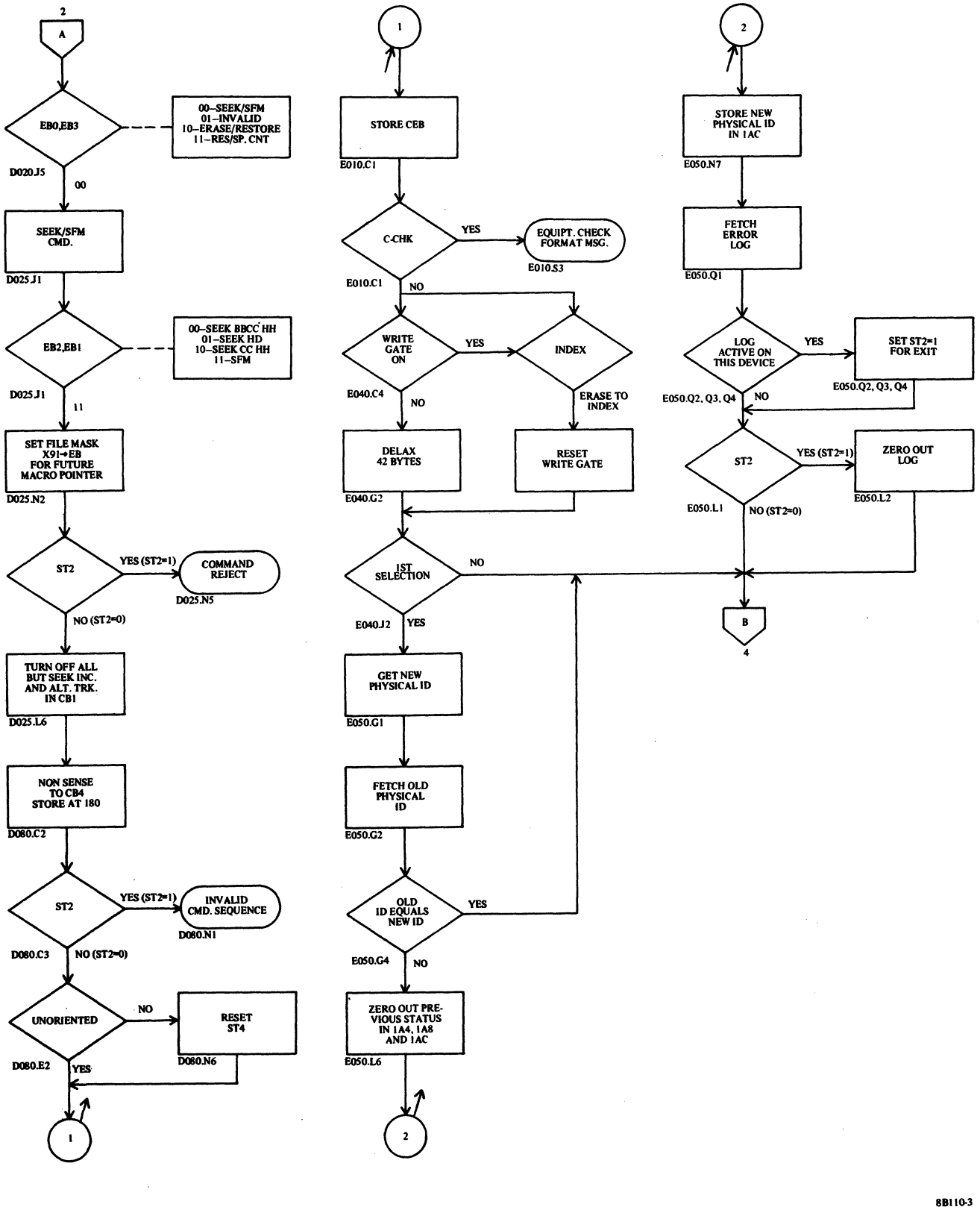


Figure 3-65. Set File Mask (Sheet 3 of 5)

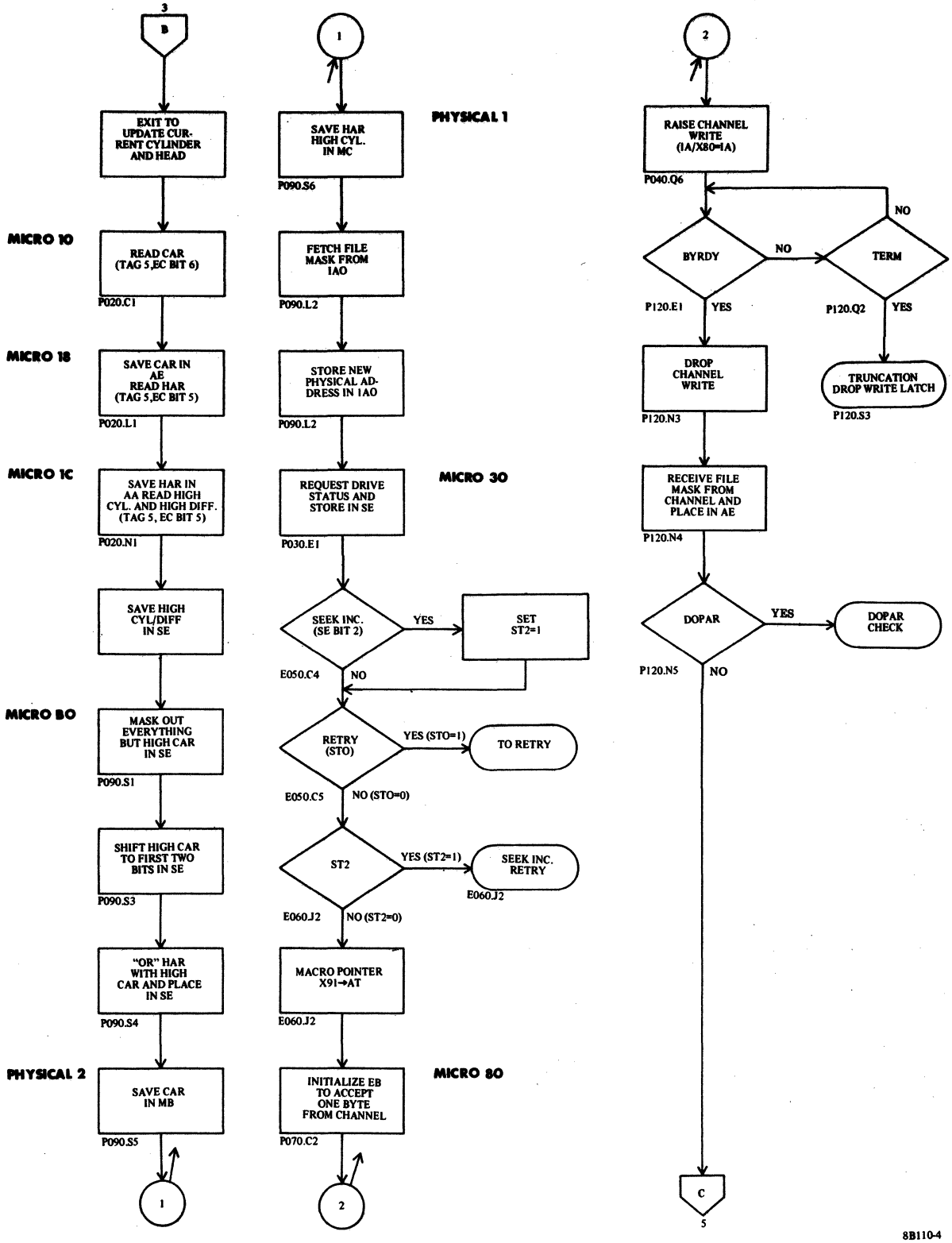


Figure 3-65. Set File Mask (Sheet 4 of 5)

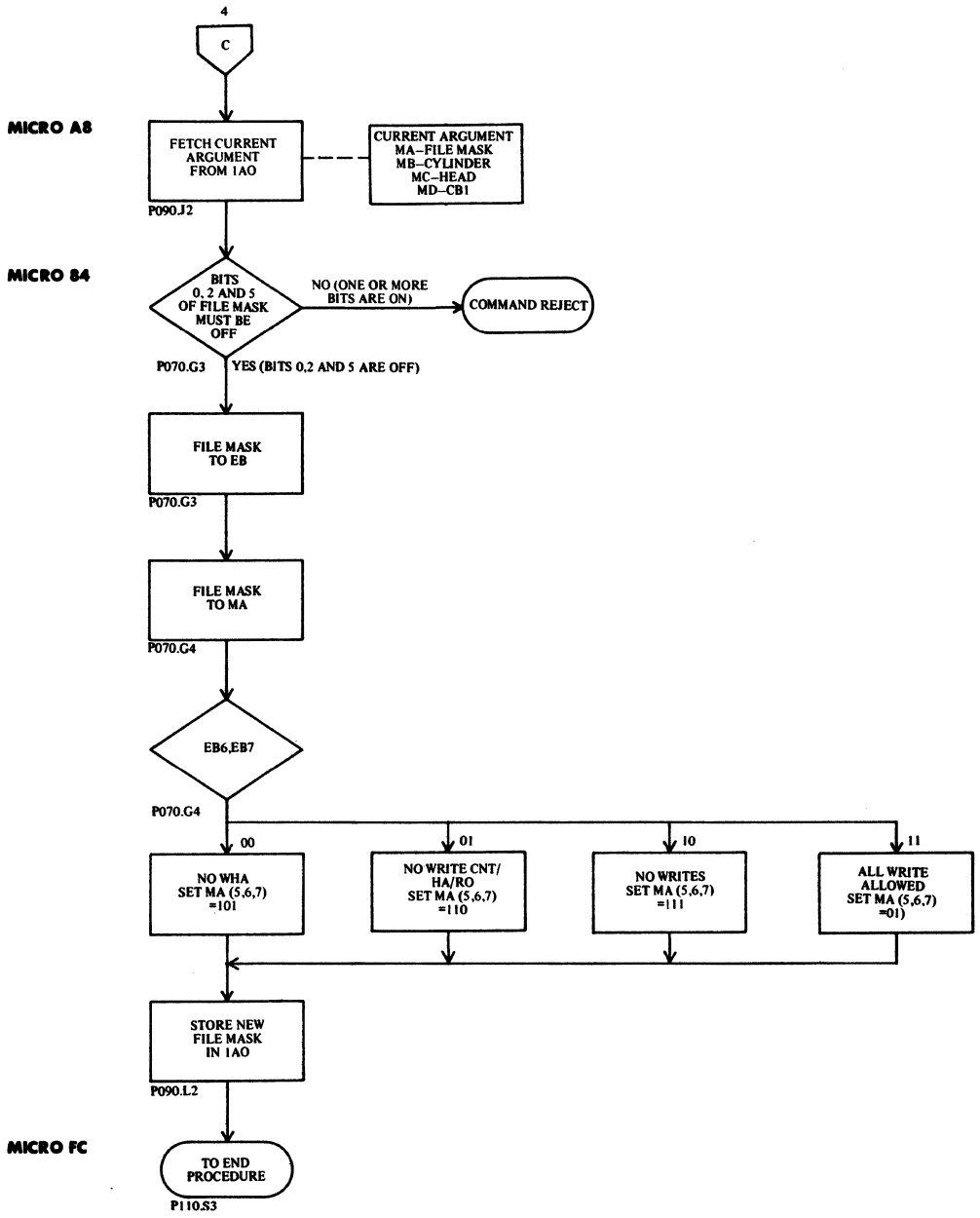
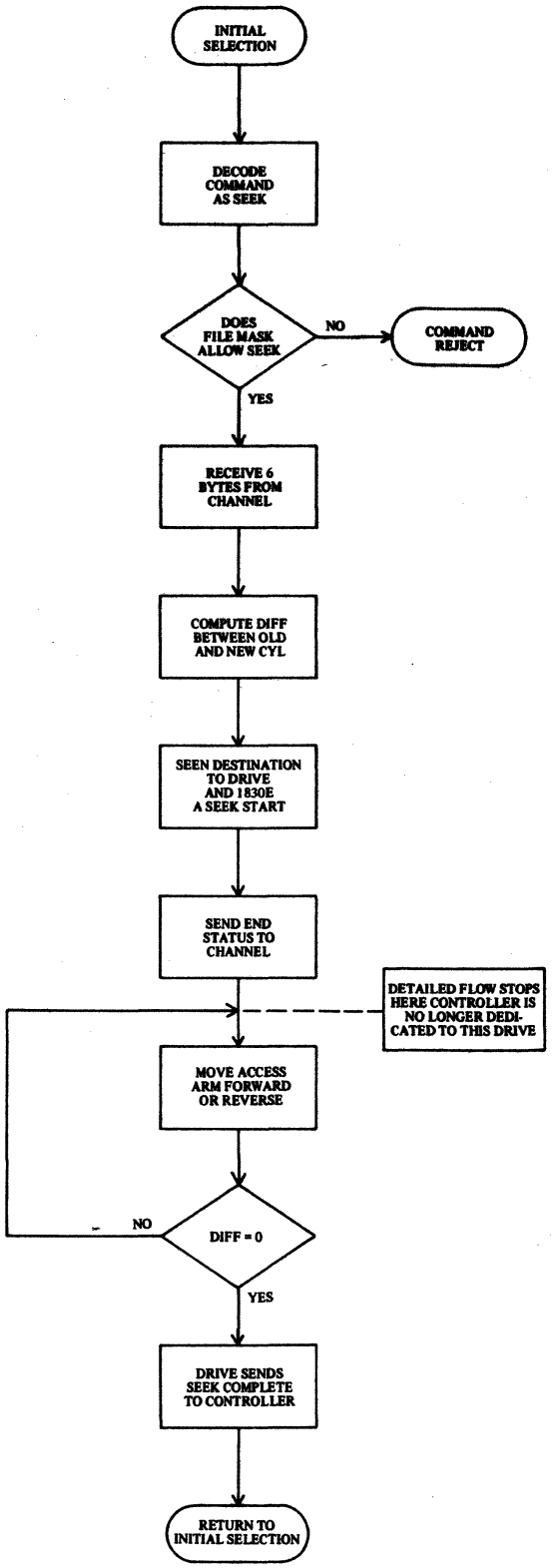


Figure 3-65. Set File Mask (Sheet 5 of 5)

SEEK CYLINDER AND HEAD

This command selects the drive and transfers the seek control information to that drive. The drive then moves the access mechanism to the cylinder and stores the head address.

GENERAL FLOWCHART



8B111-1

Figure 3-66. Seek CCHH (Sheet 1 of 9)

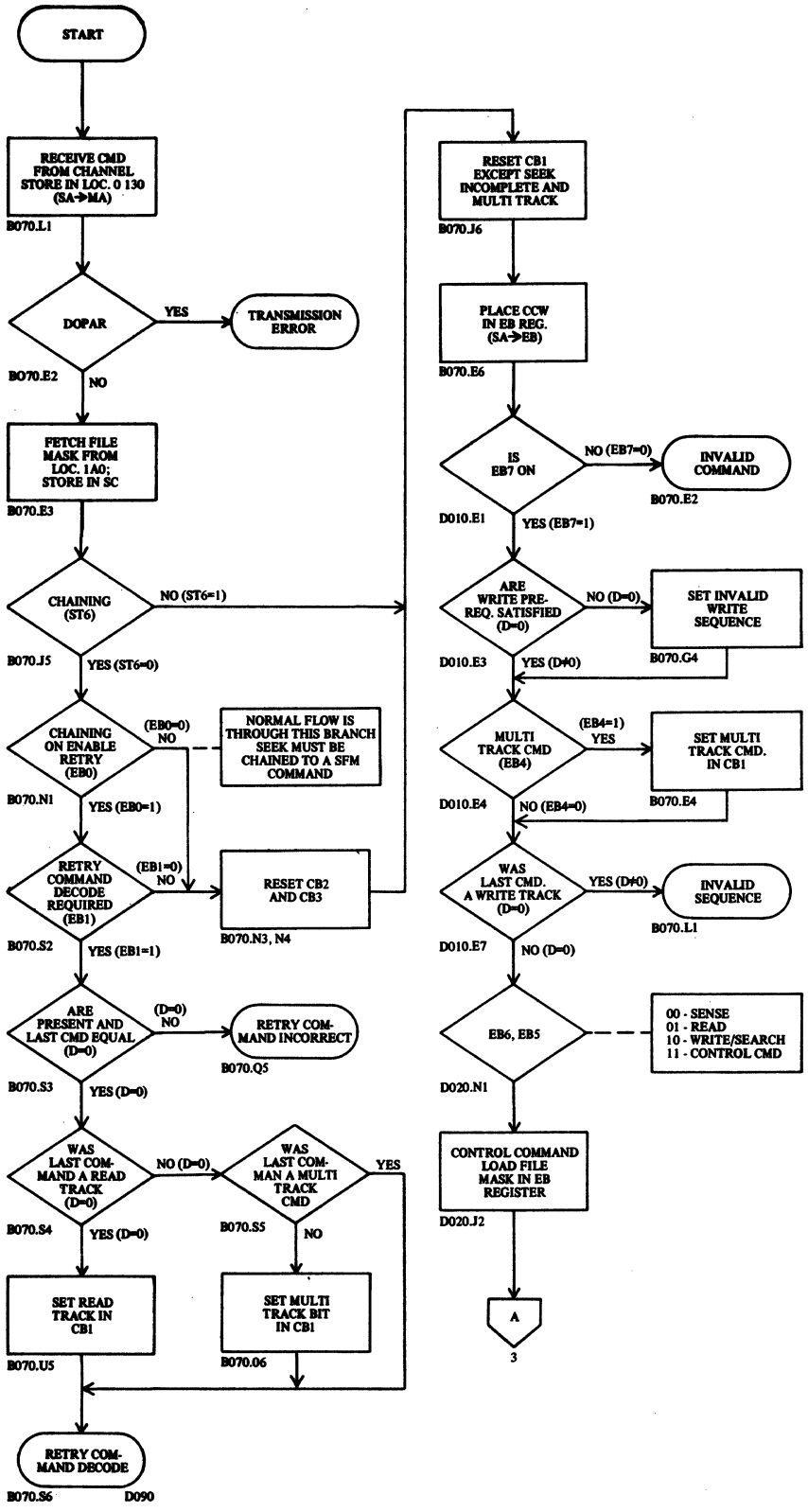


Figure 3-66. Seek CCHH (Sheet 2 of 9)

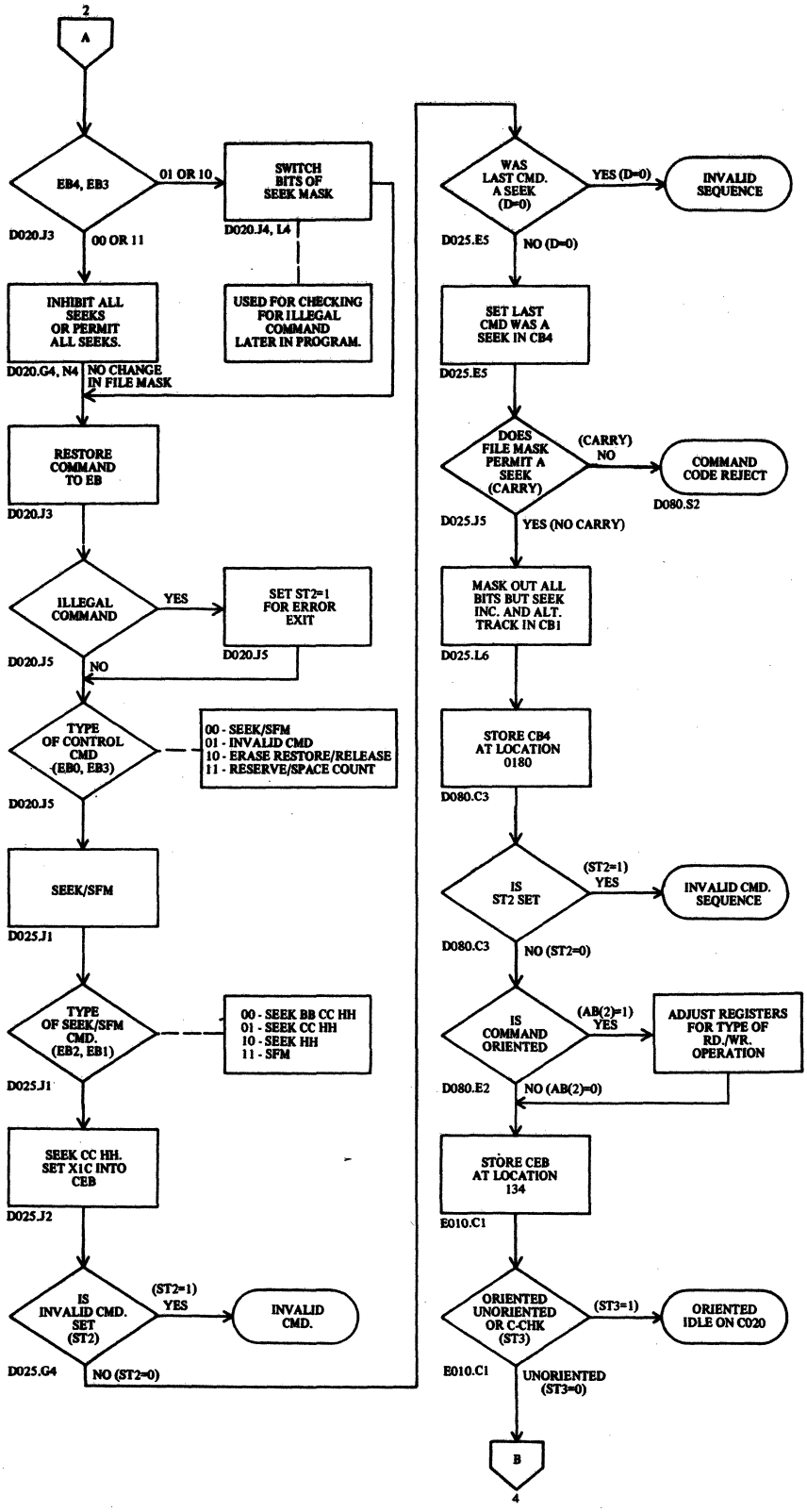


Figure 3-66. Seek CCHH (Sheet 3 of 9)

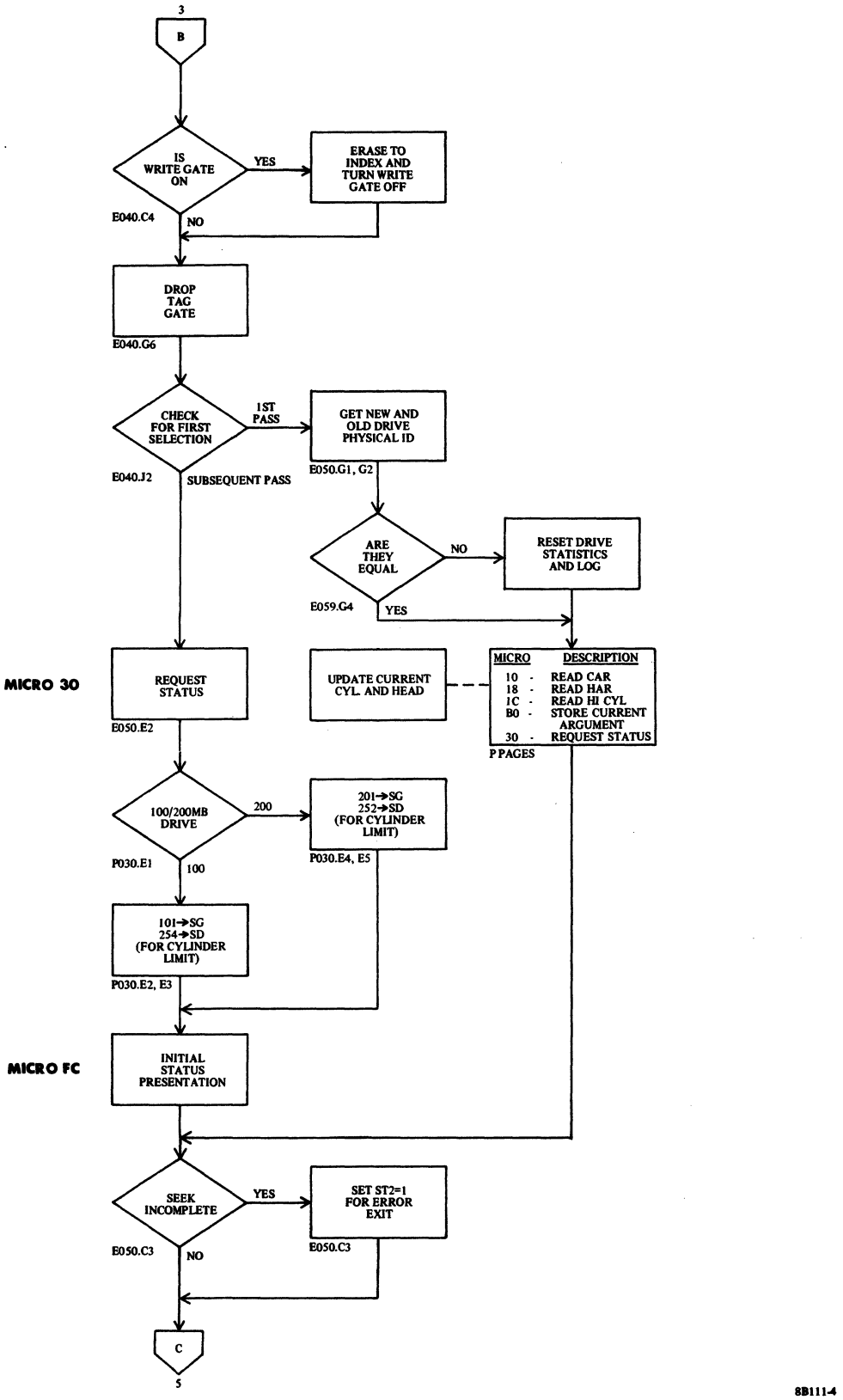


Figure 3-66. Seek CCHH (Sheet 4 of 9)

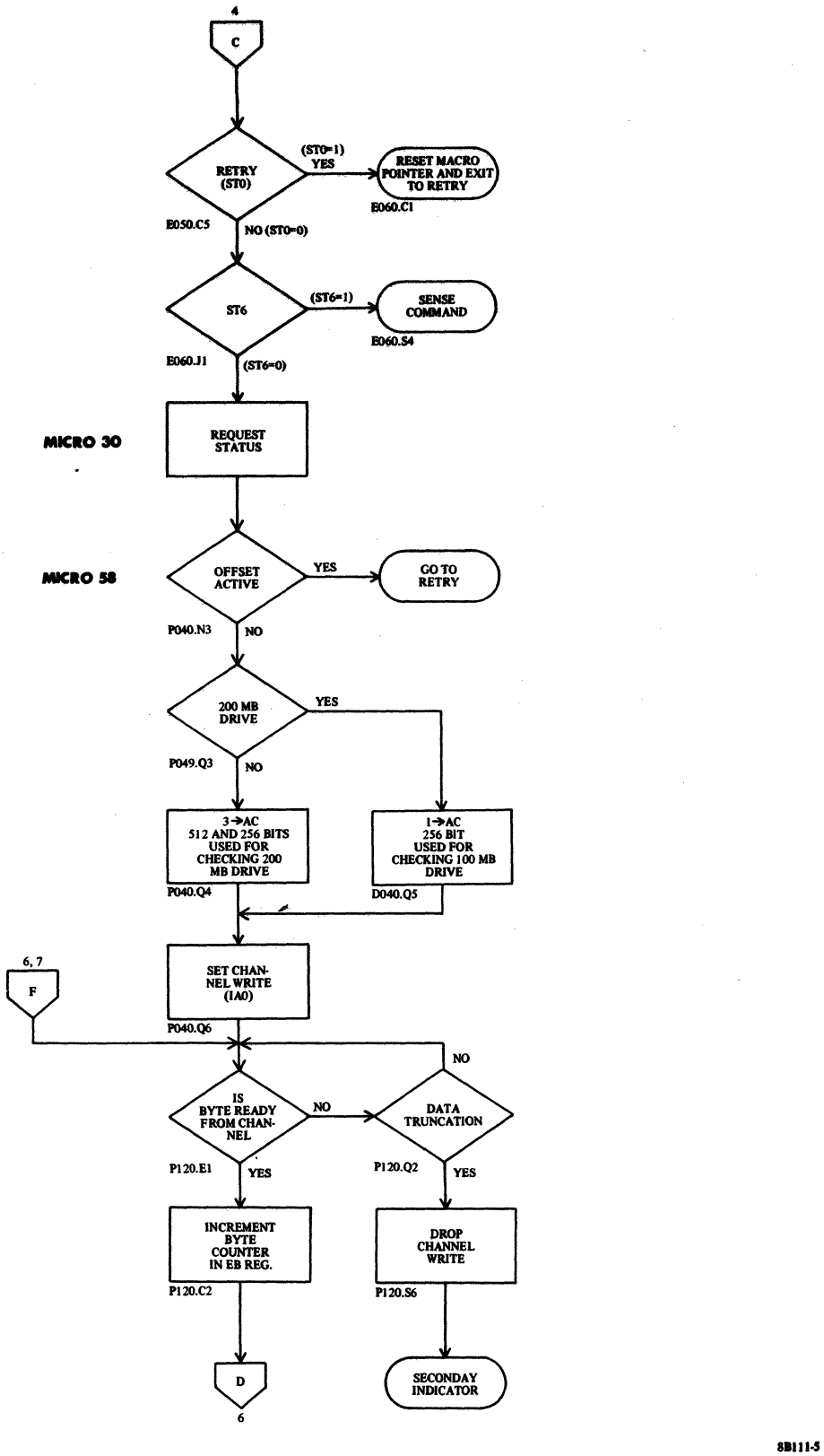


Figure 3-66. Seek CCHH (Sheet 5 of 9)

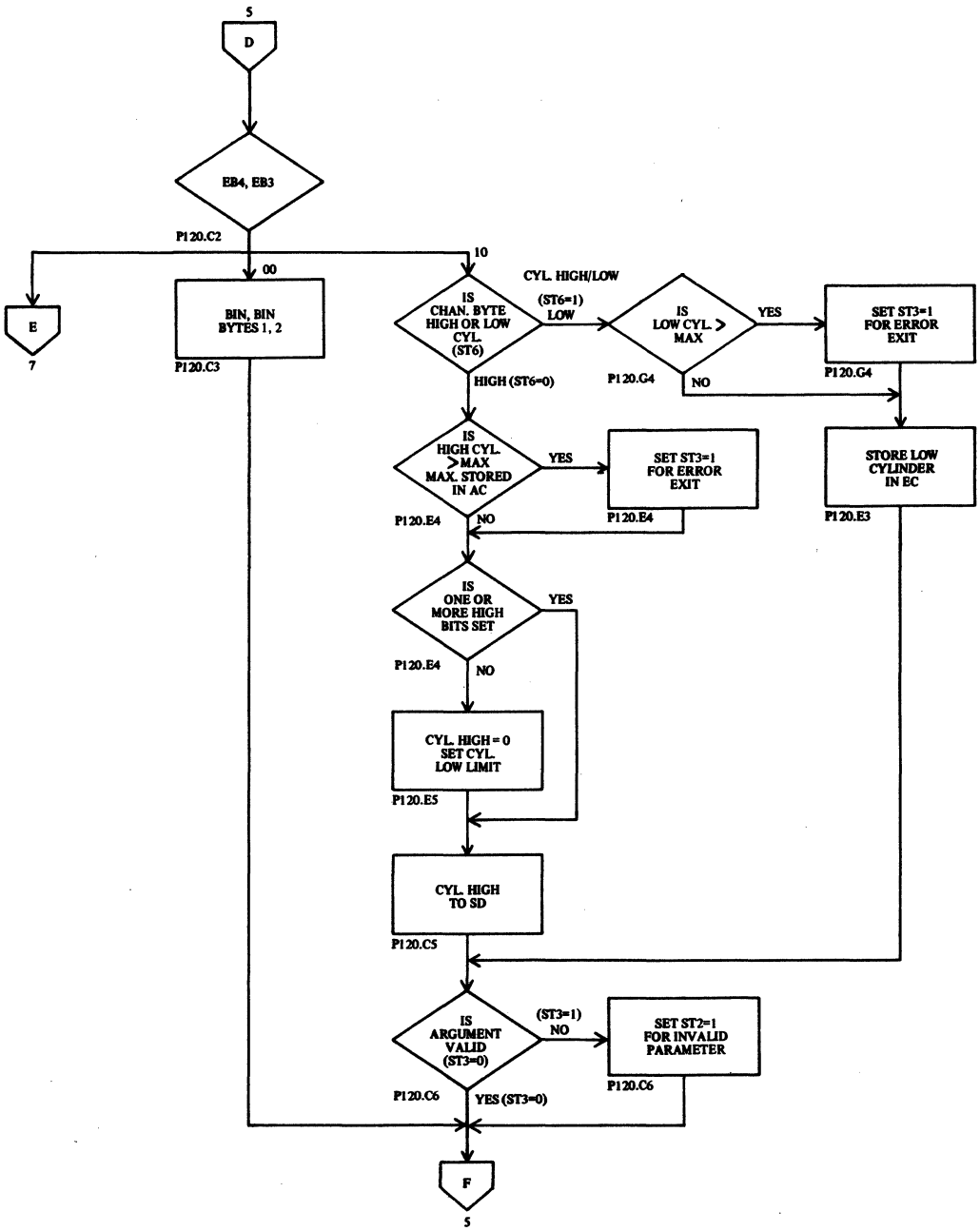


Figure 3-66. Seek CCHH (Sheet 6 of 9)

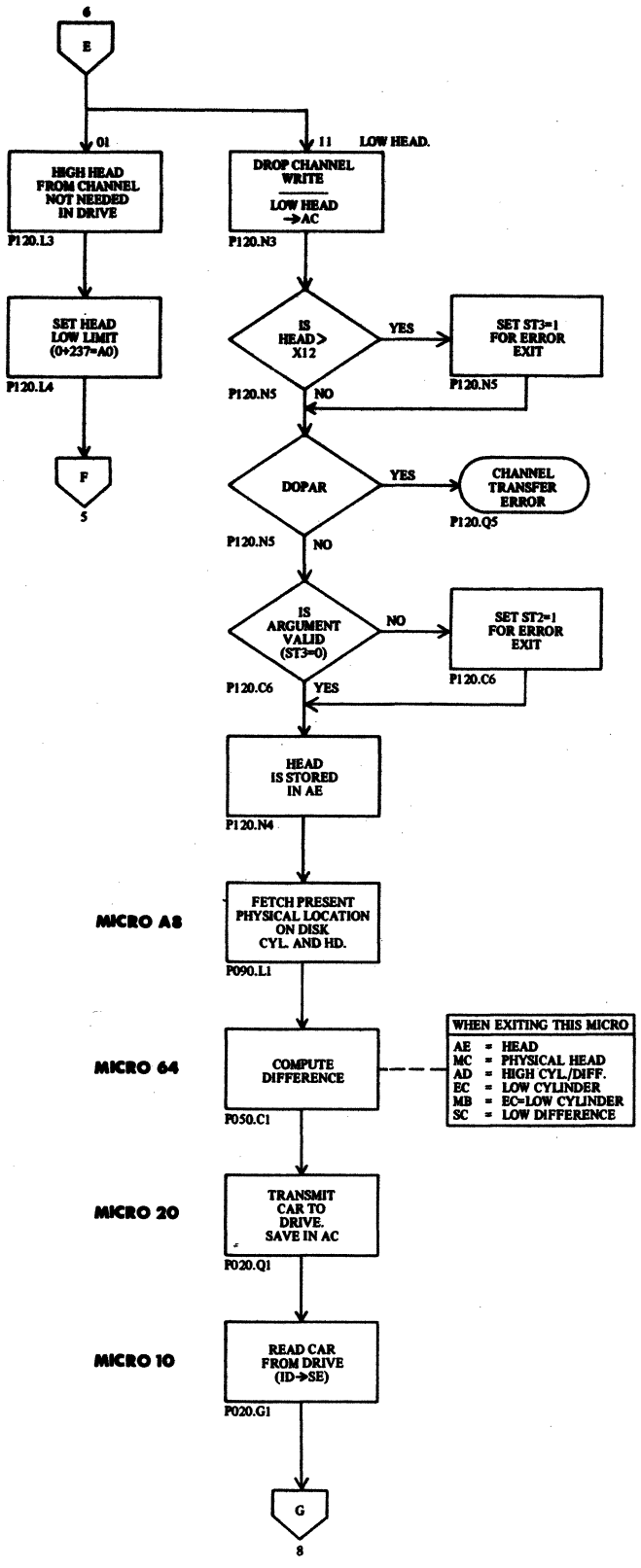


Figure 3-66. Seek CCHH (Sheet 7 of 9)

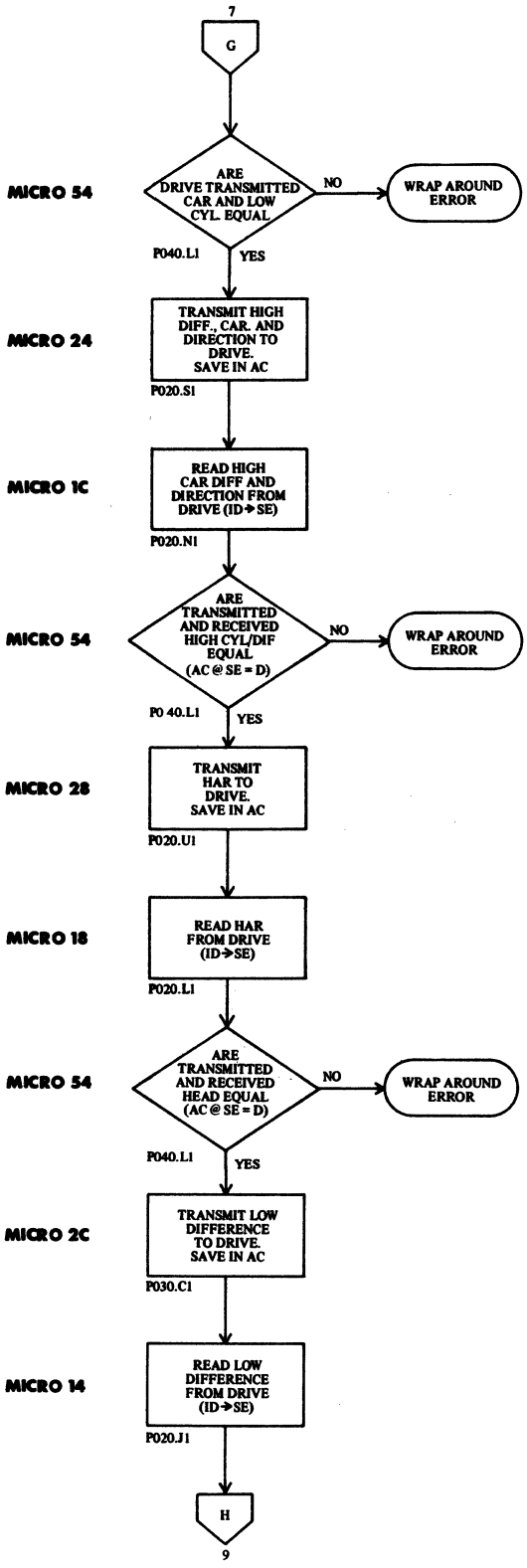


Figure 3-66. Seek CCHH (Sheet 8 of 9)

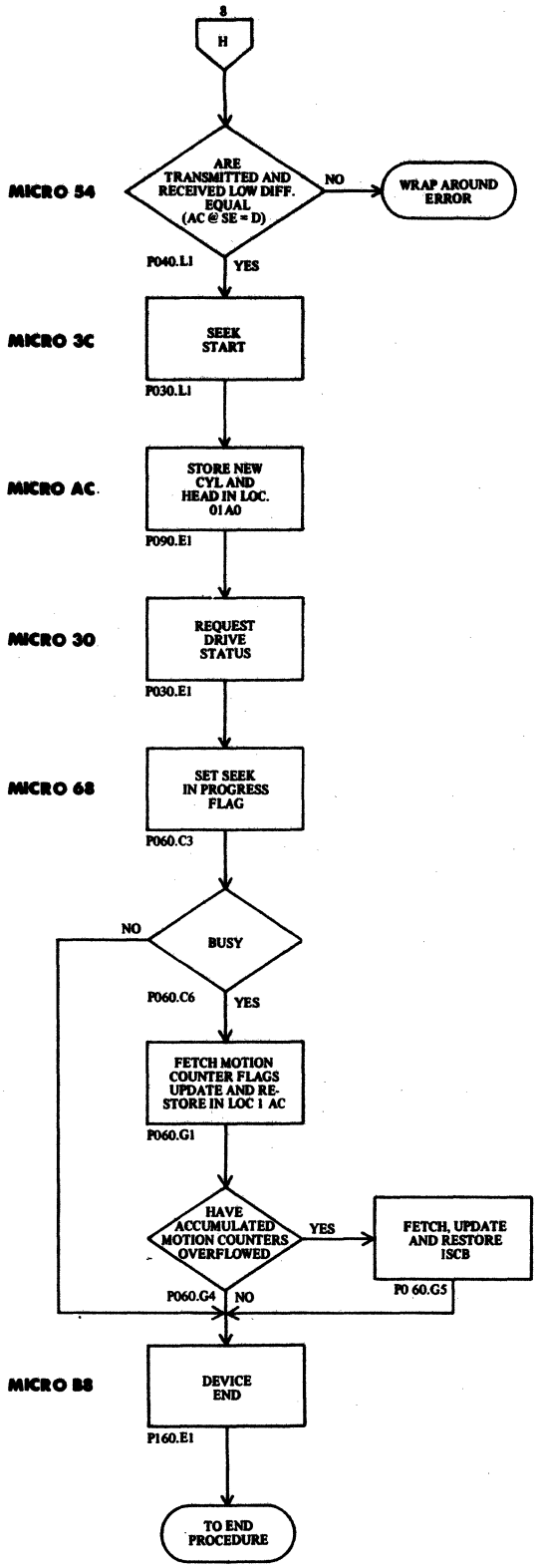
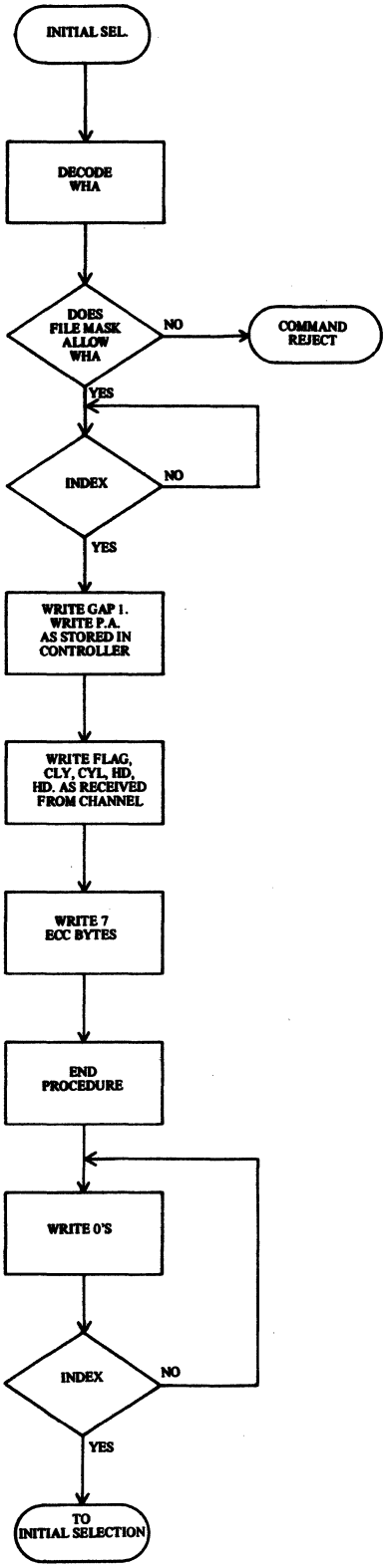


Figure 3-66. Seek CCHH (Sheet 9 of 9)

WRITE HOME ADDRESS

This command writes a new Home Address on the drive and track selected by the system. The procedure consists of writing Gap 1 (92 bytes) followed by a 7-byte Home Address field and a 7-byte controller-generated ECC field.

GENERAL FLOWCHART



8B112-1

Figure 3-67. Write Home Address (Sheet 1 of 8)

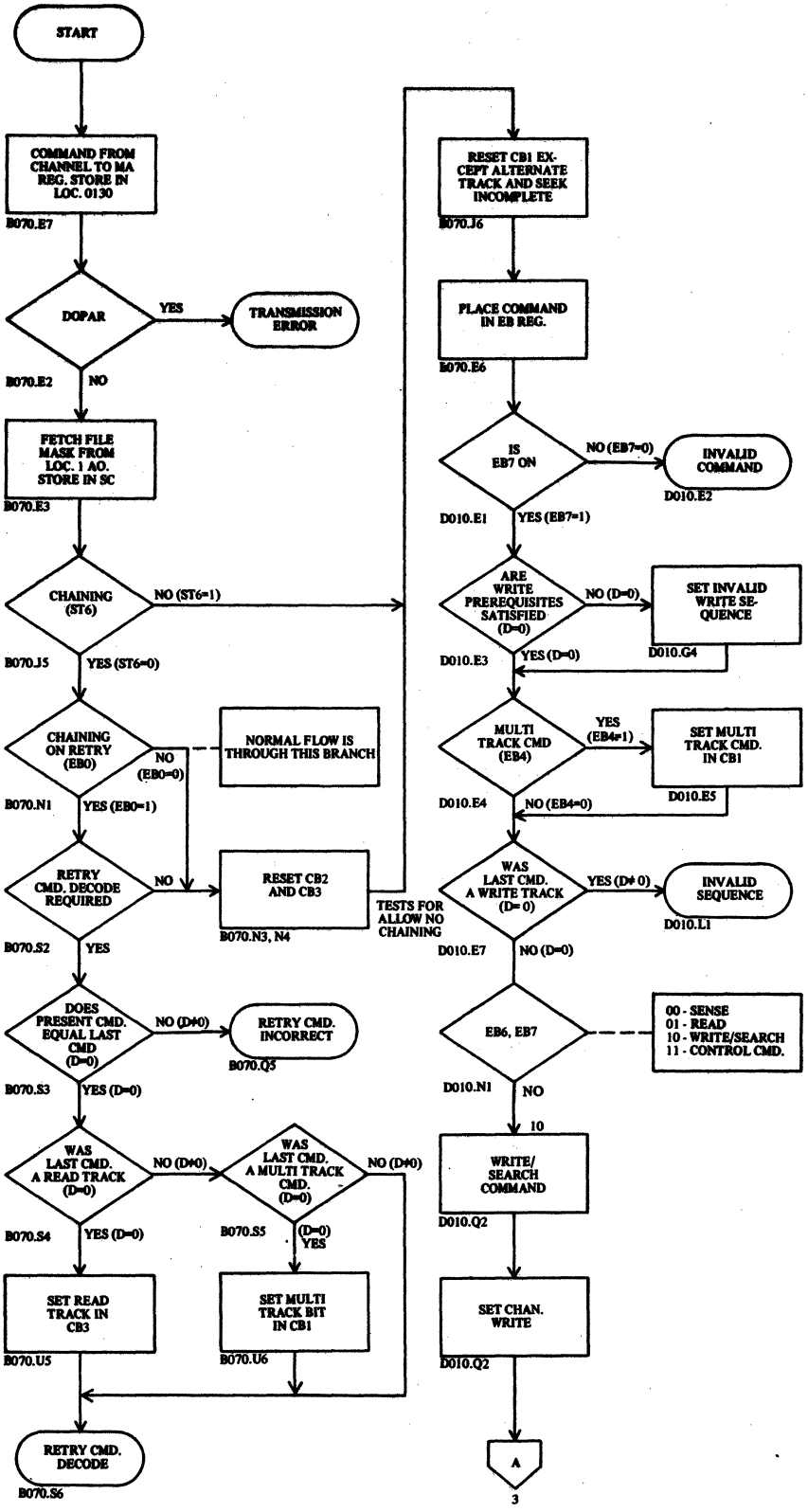
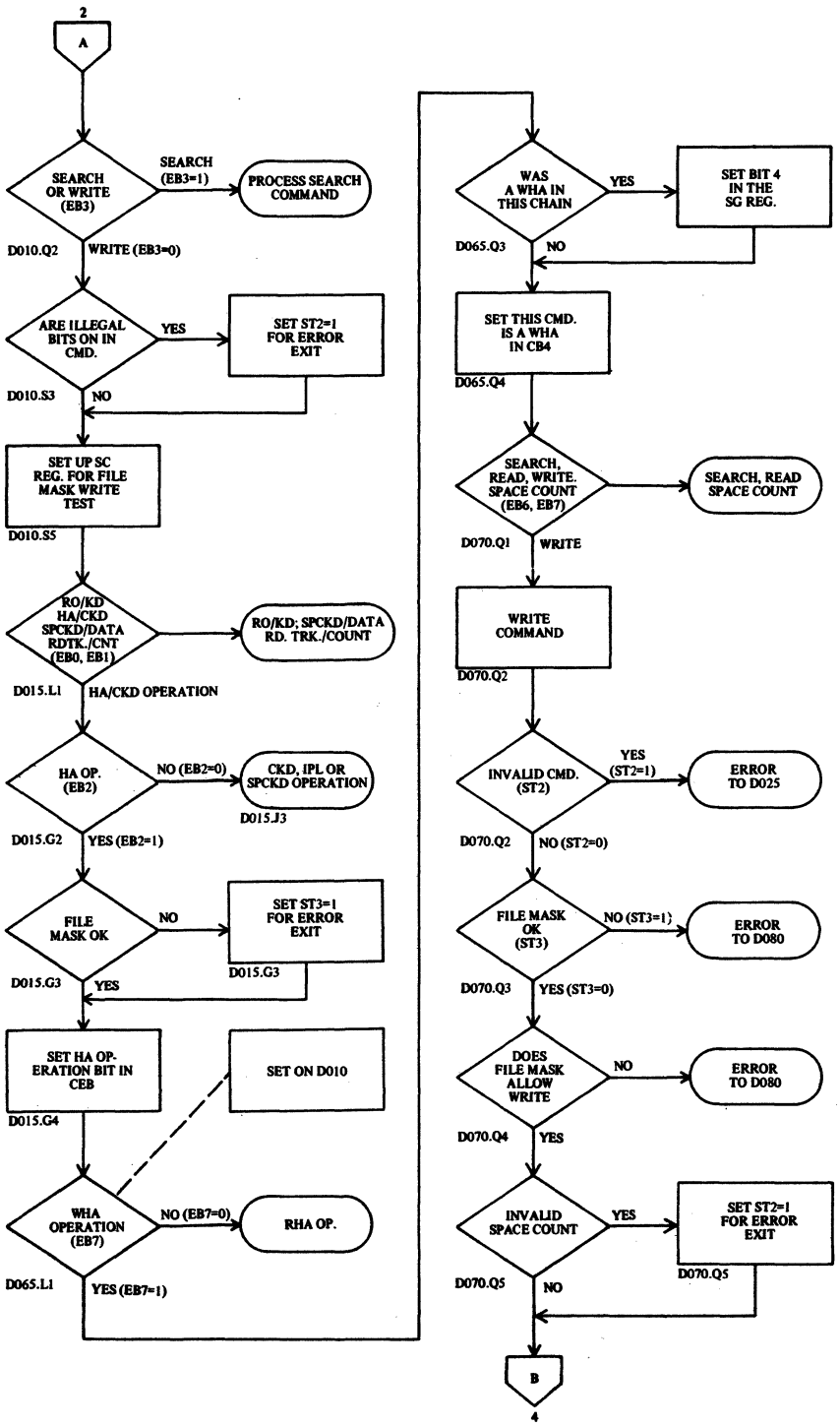


Figure 3-67. Write Home Address (Sheet 2 of 8)



8B112-3

Figure 3-67. Write Home Address (Sheet 3 of 8)

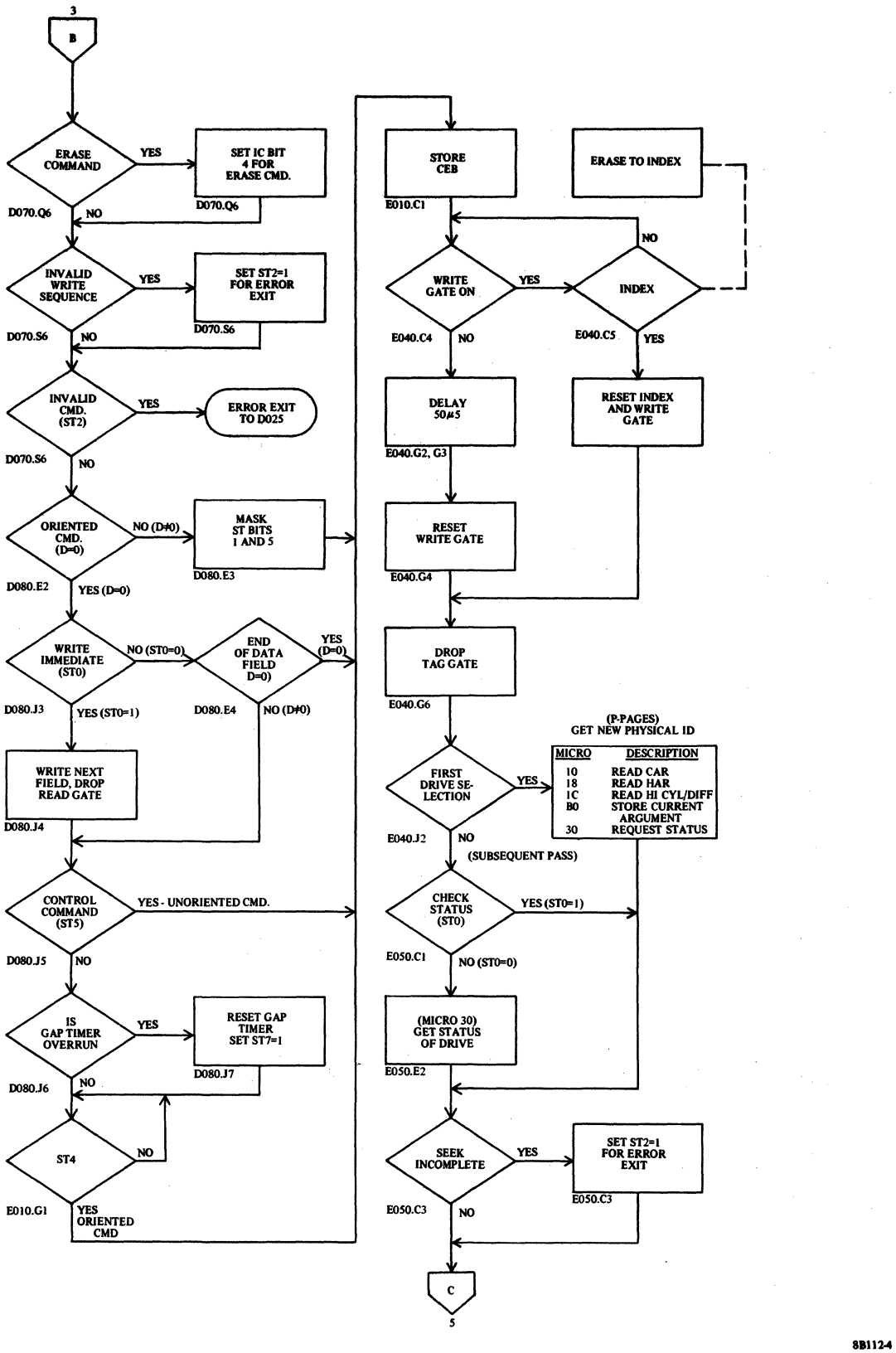


Figure 3-67. Write Home Address (Sheet 4 of 8)

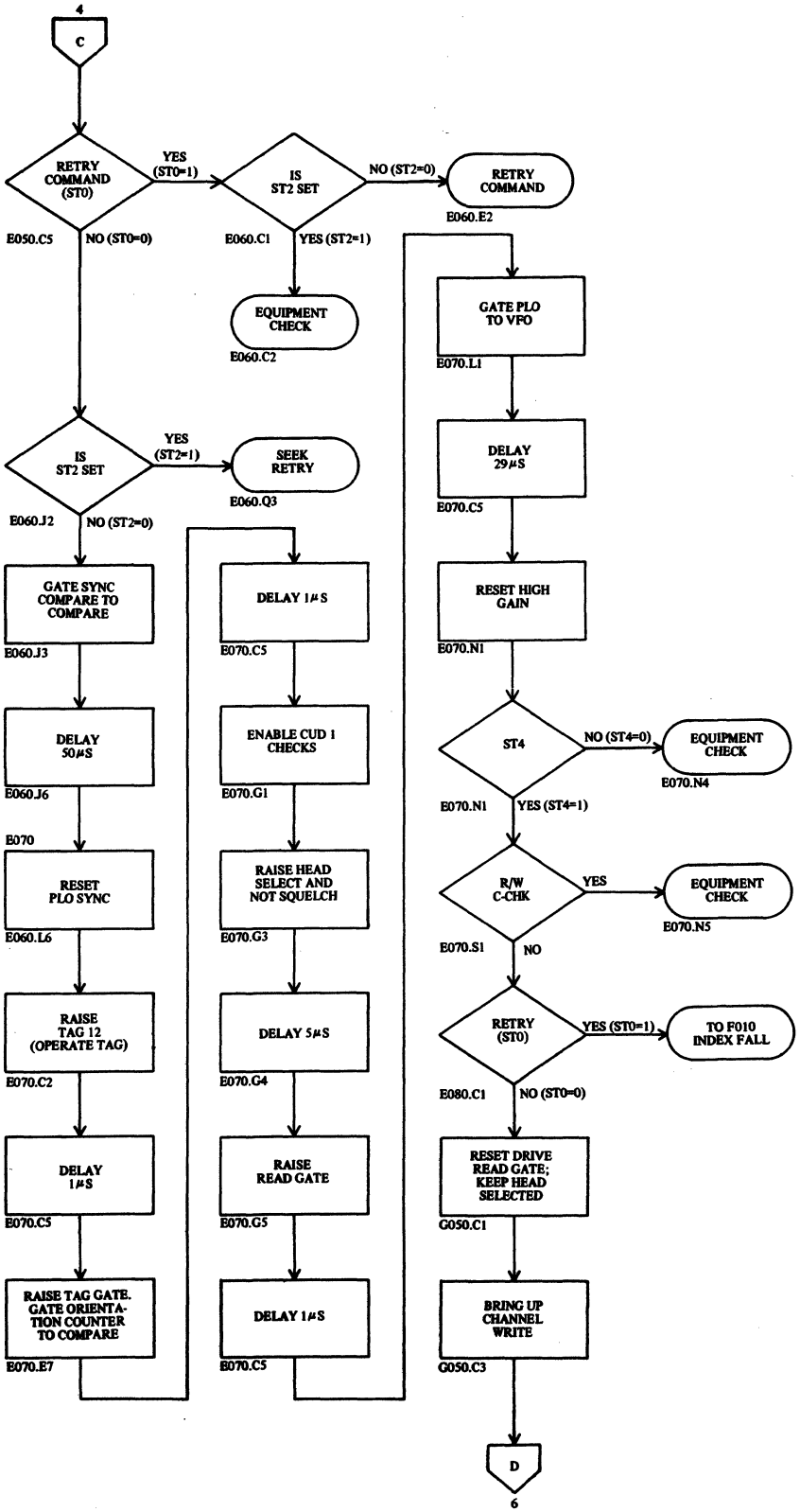


Figure 3-67. Write Home Address (Sheet 5 of 8)

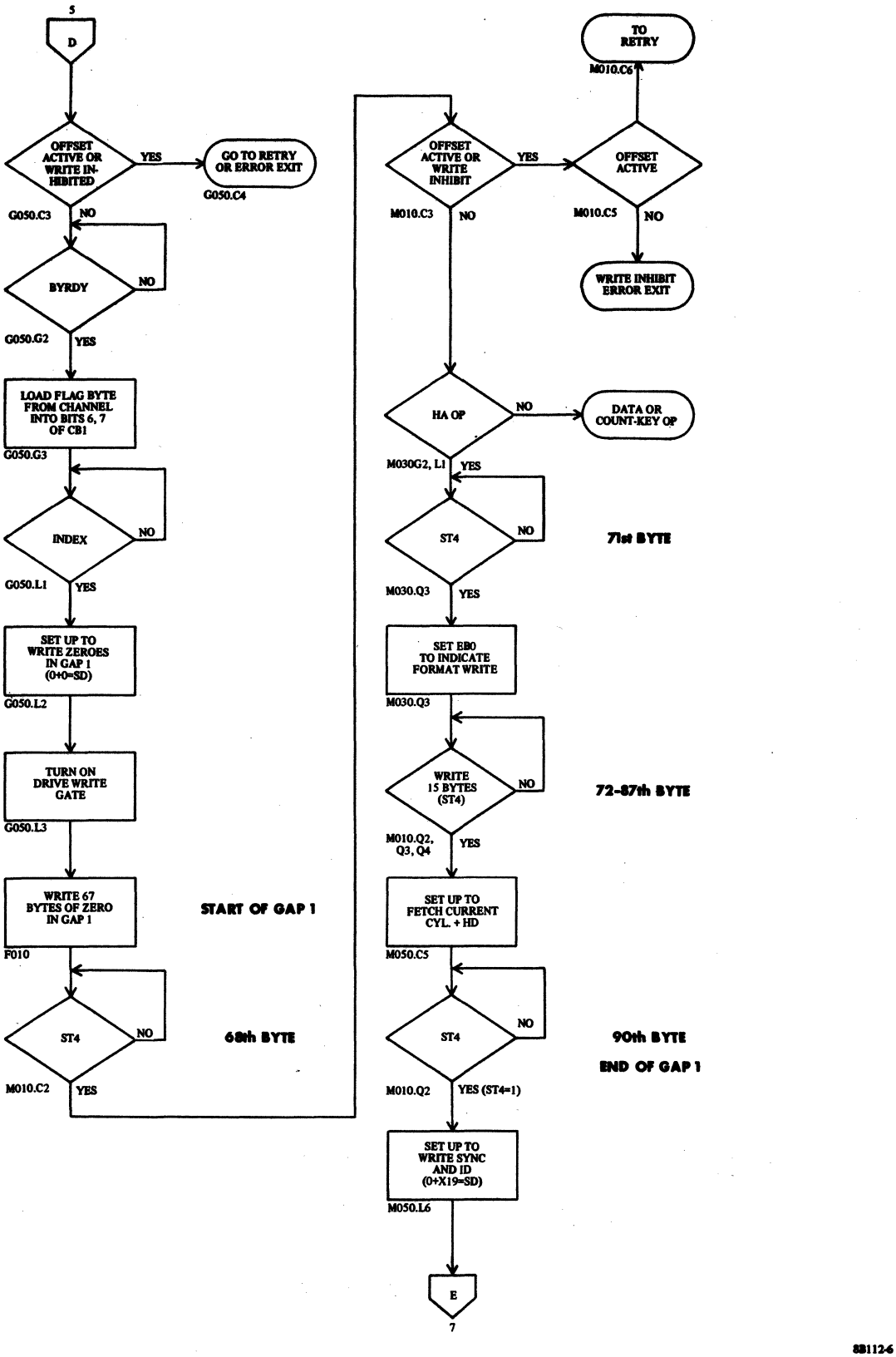


Figure 3-67. Write Home Address (Sheet 6 of 8)

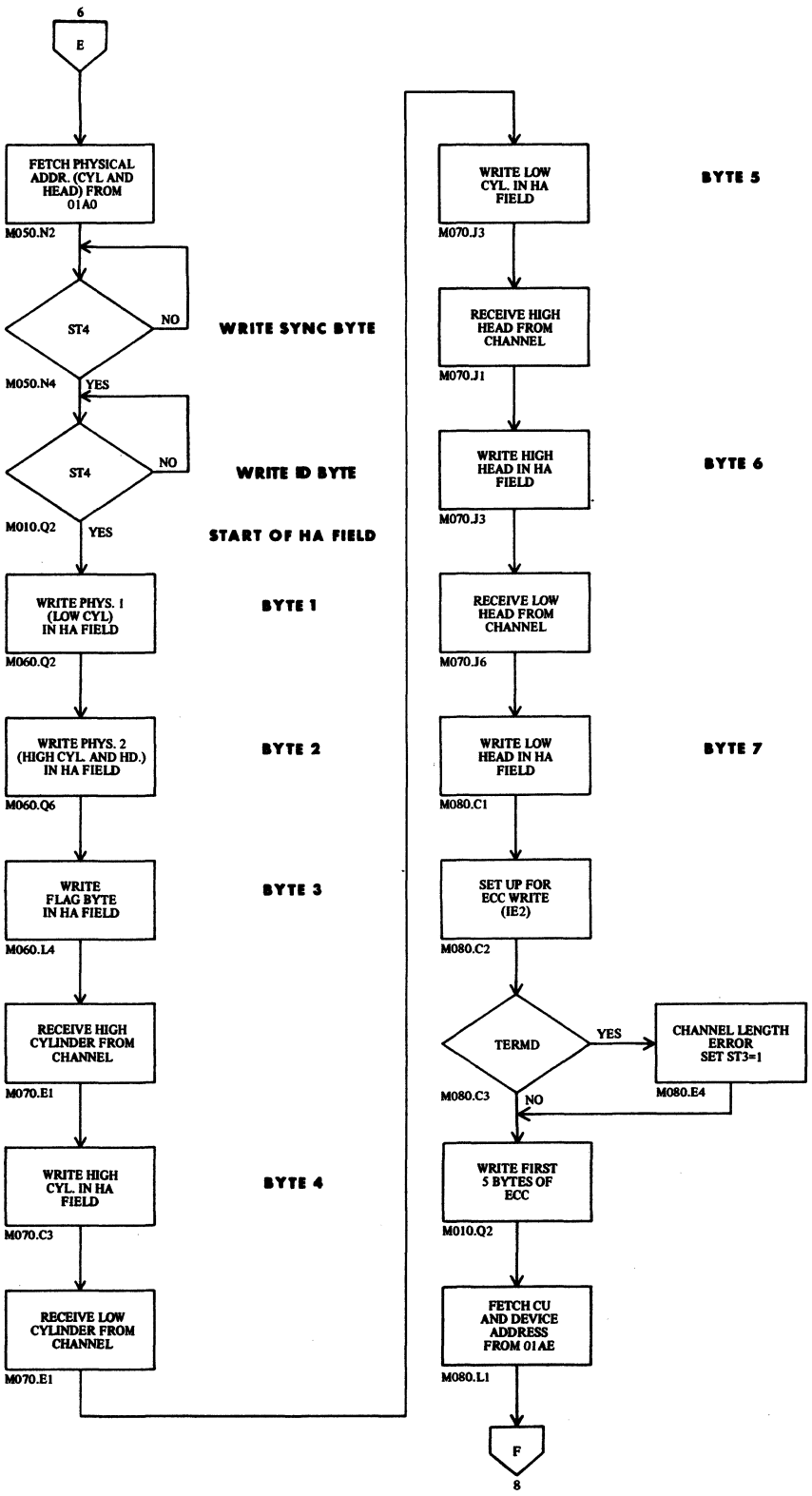


Figure 3-67. Write Home Address (Sheet 7 of 8)

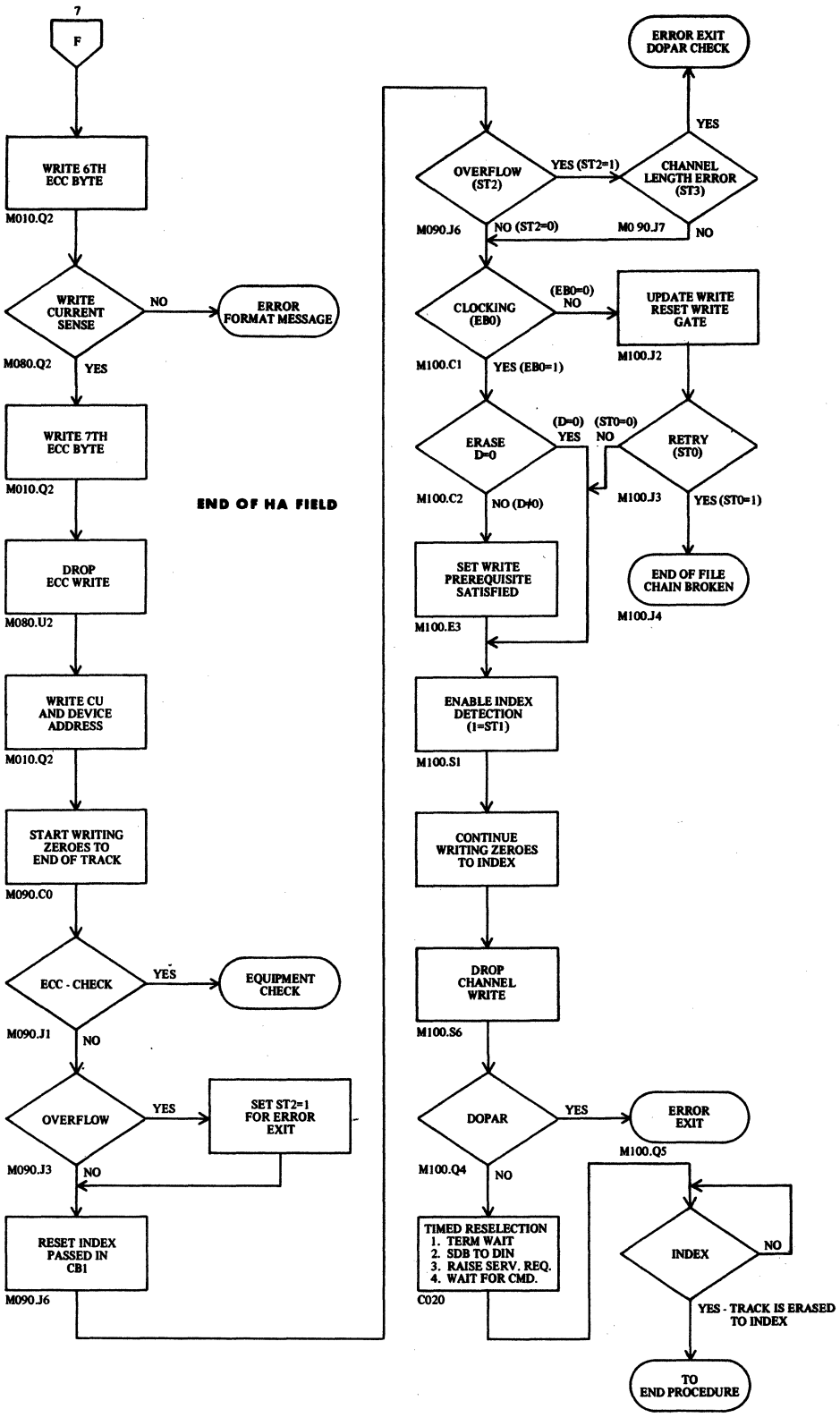
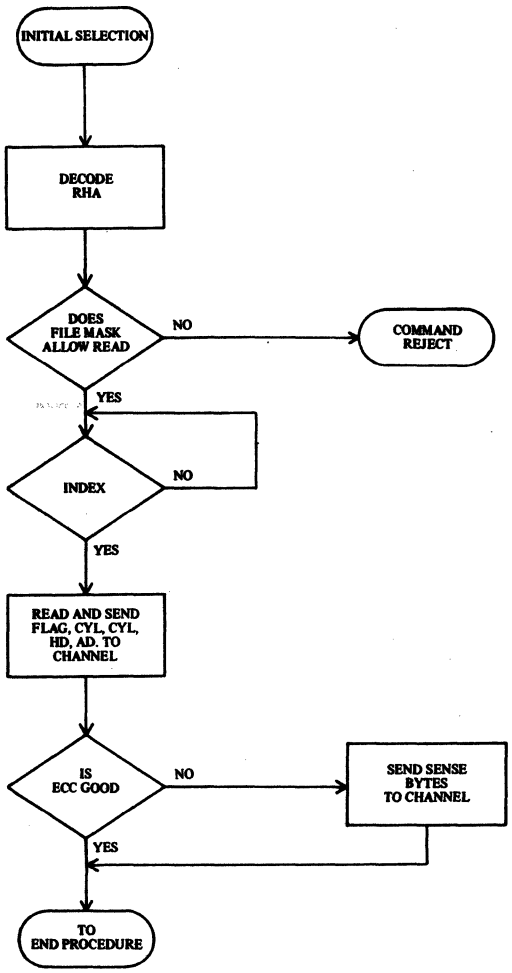


Figure 3-67. Write Home Address (Sheet 8 of 8)

READ HOME ADDRESS

The command reads and transfers a home address field on a drive selected by the channel. Seven bytes are read from the drive: Physical 1 and 2, Flag, High and Low cylinder, High and Low Head. All but Physical bytes 1 and 2 are transferred to the channel.

GENERAL FLOWCHART



8B113-1

Figure 3-68. Read Home Address (Sheet 1 of 8)

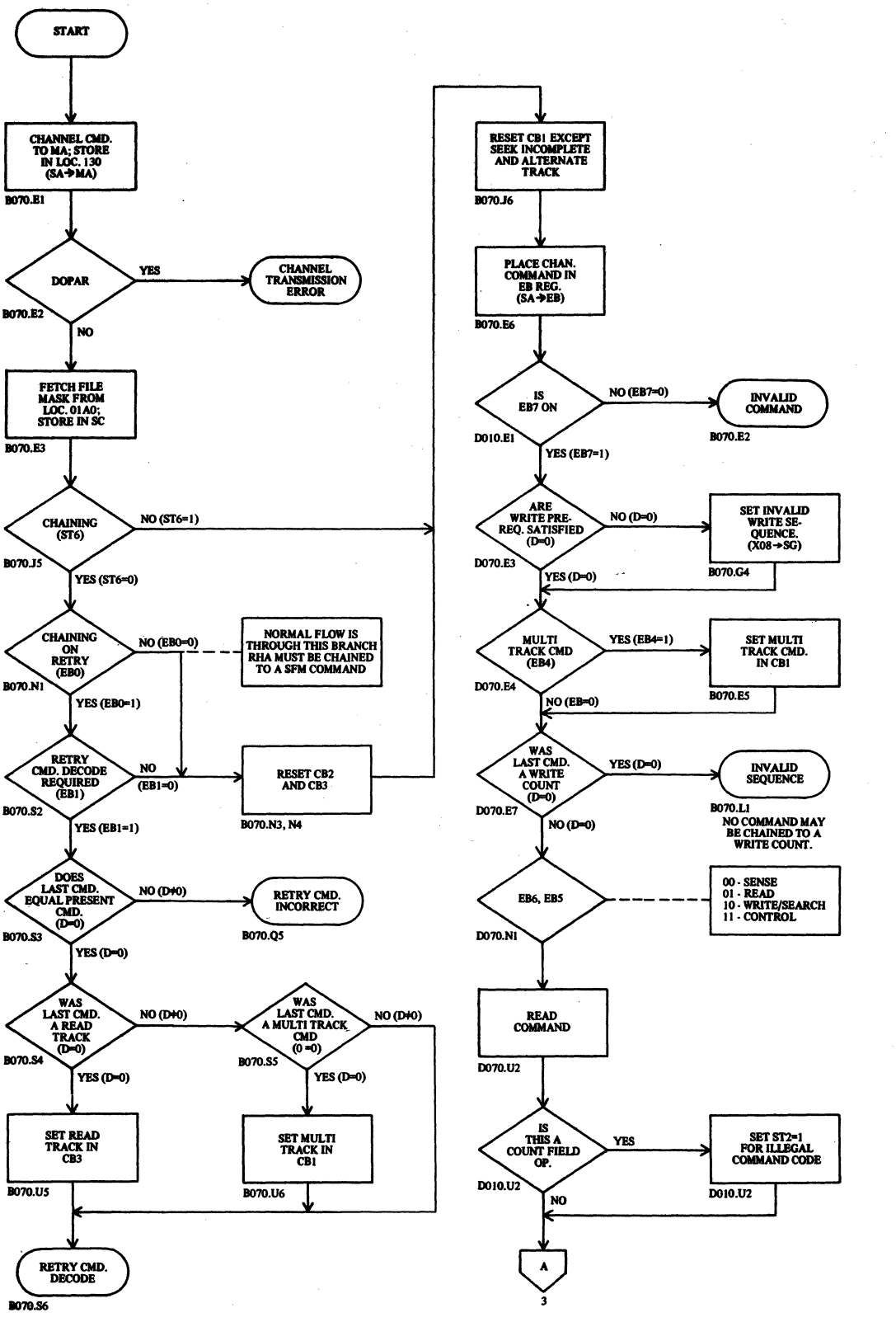
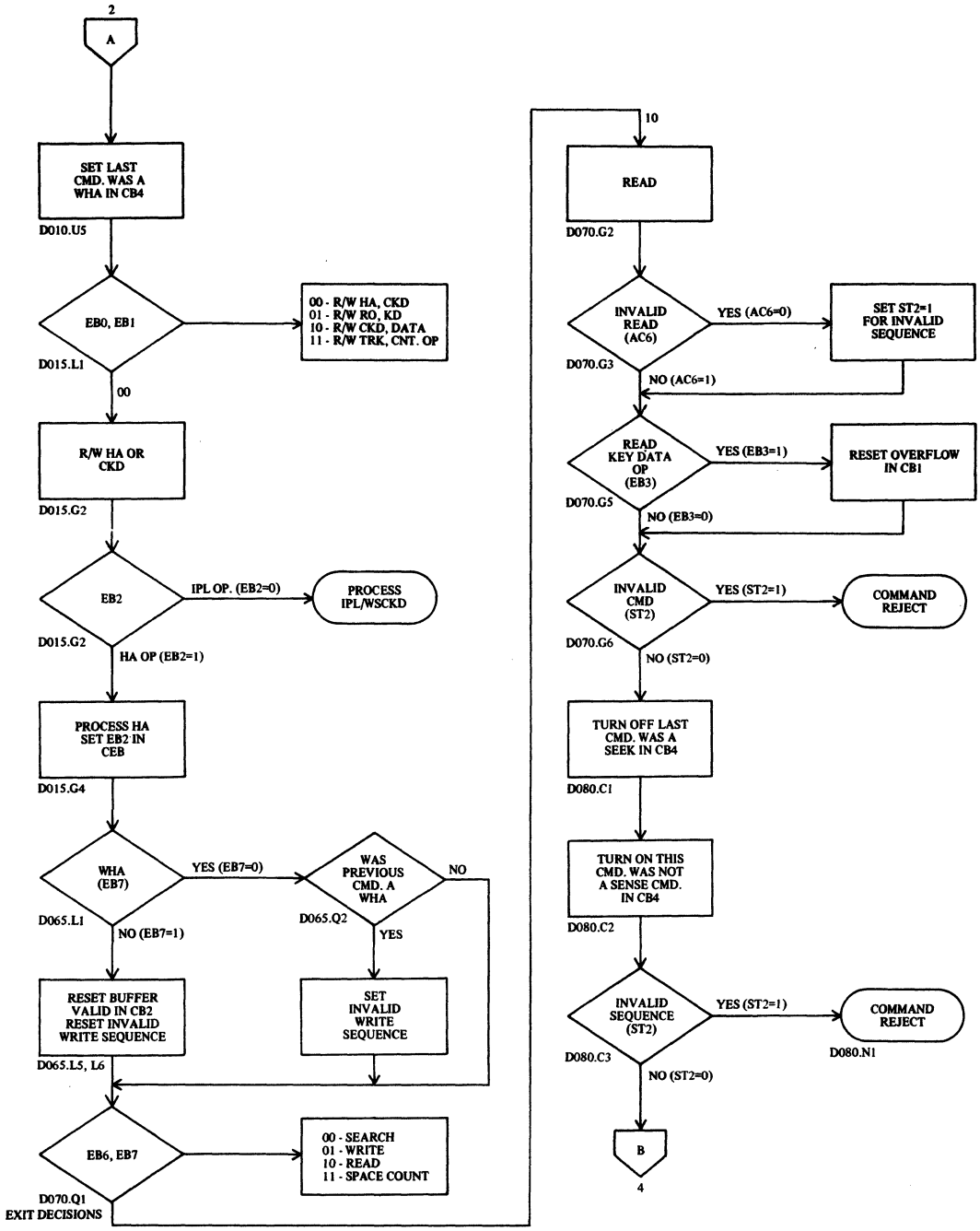


Figure 3-68. Read Home Address (Sheet 2 of 12)



8B113-3

Figure 3-68. Read Home Address (Sheet 3 of 12)

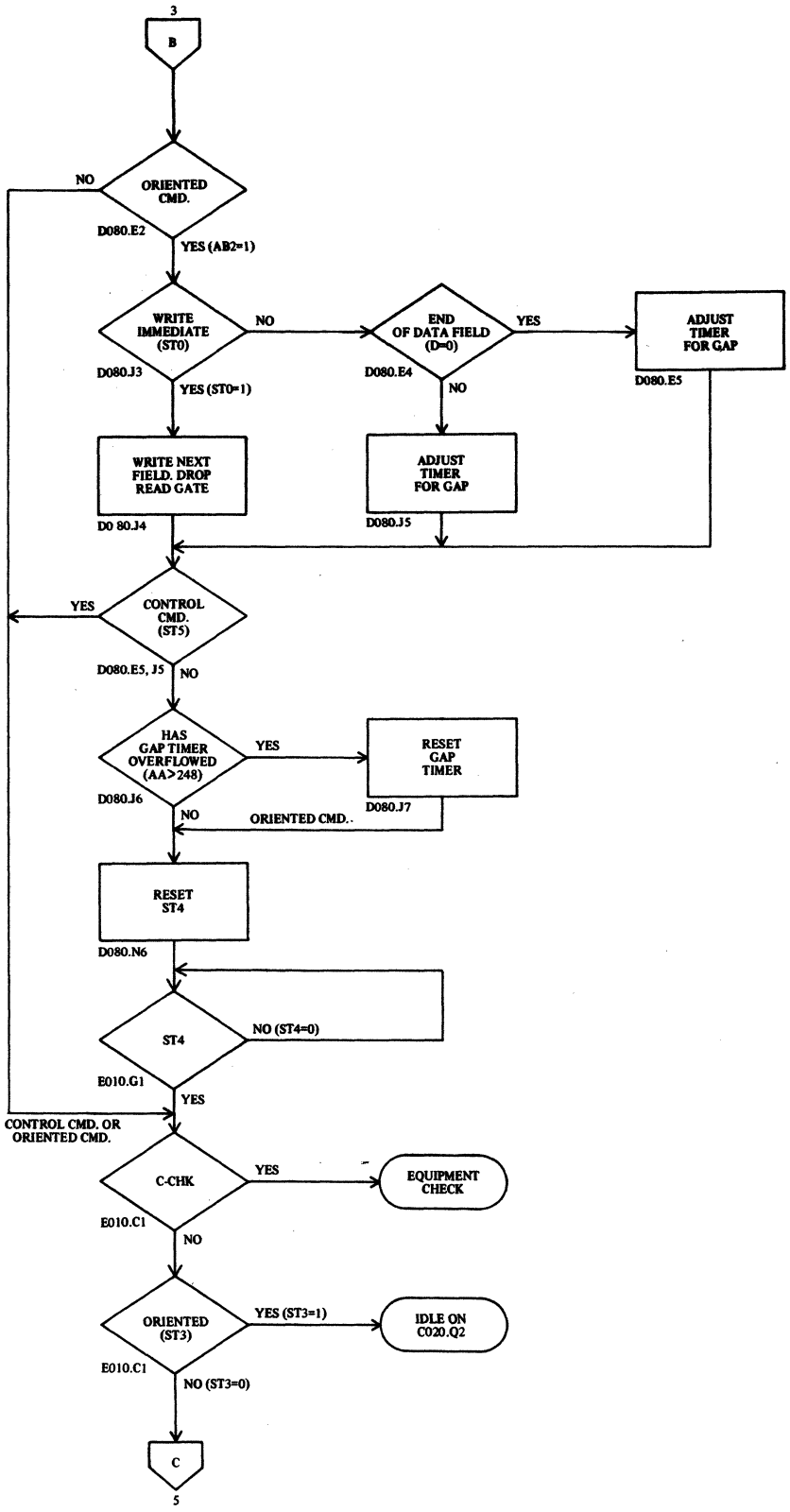


Figure 3-68. Read Home Address (Sheet 4 of 12)

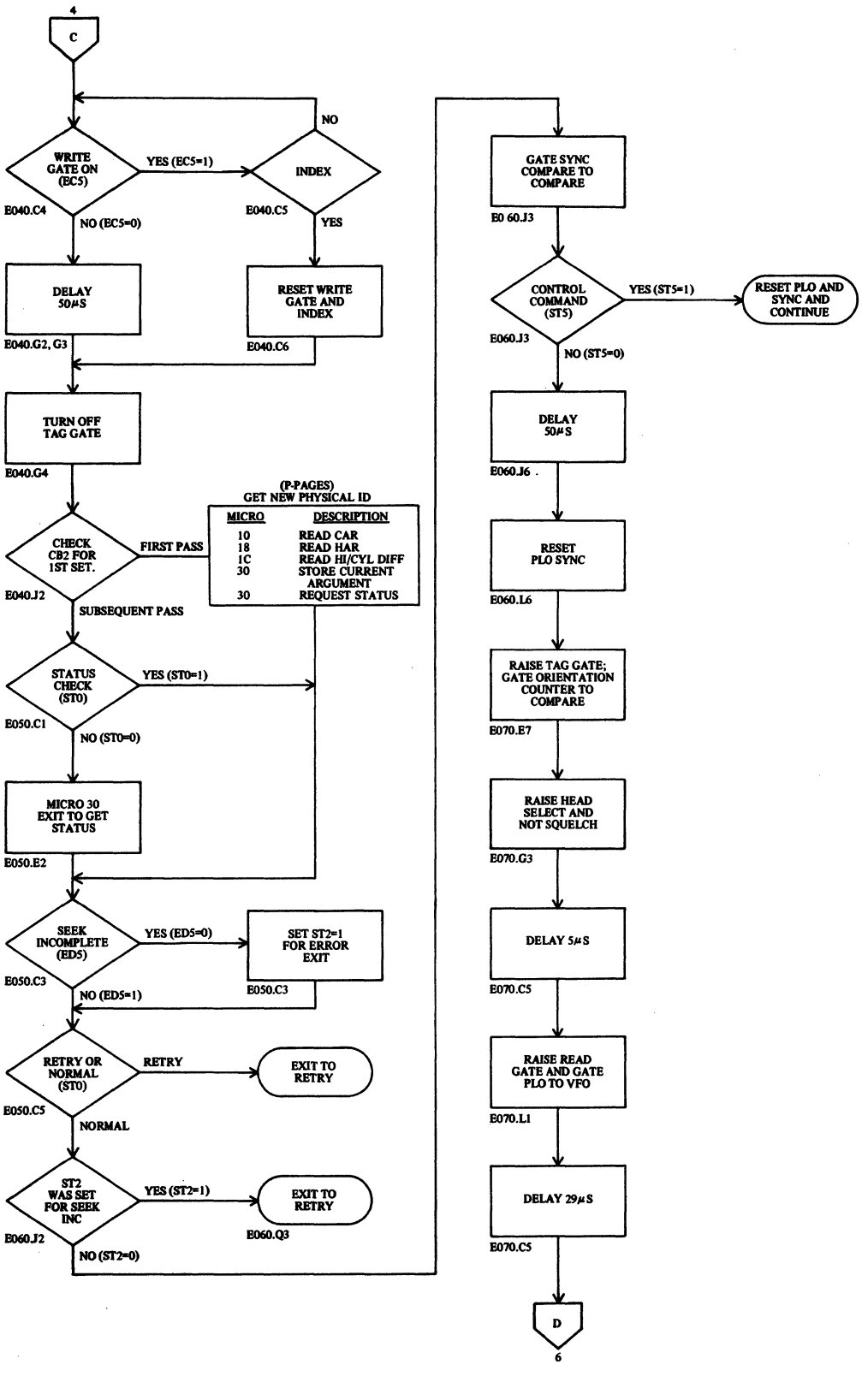


Figure 3-68. Read Home Address (Sheet 5 of 12)

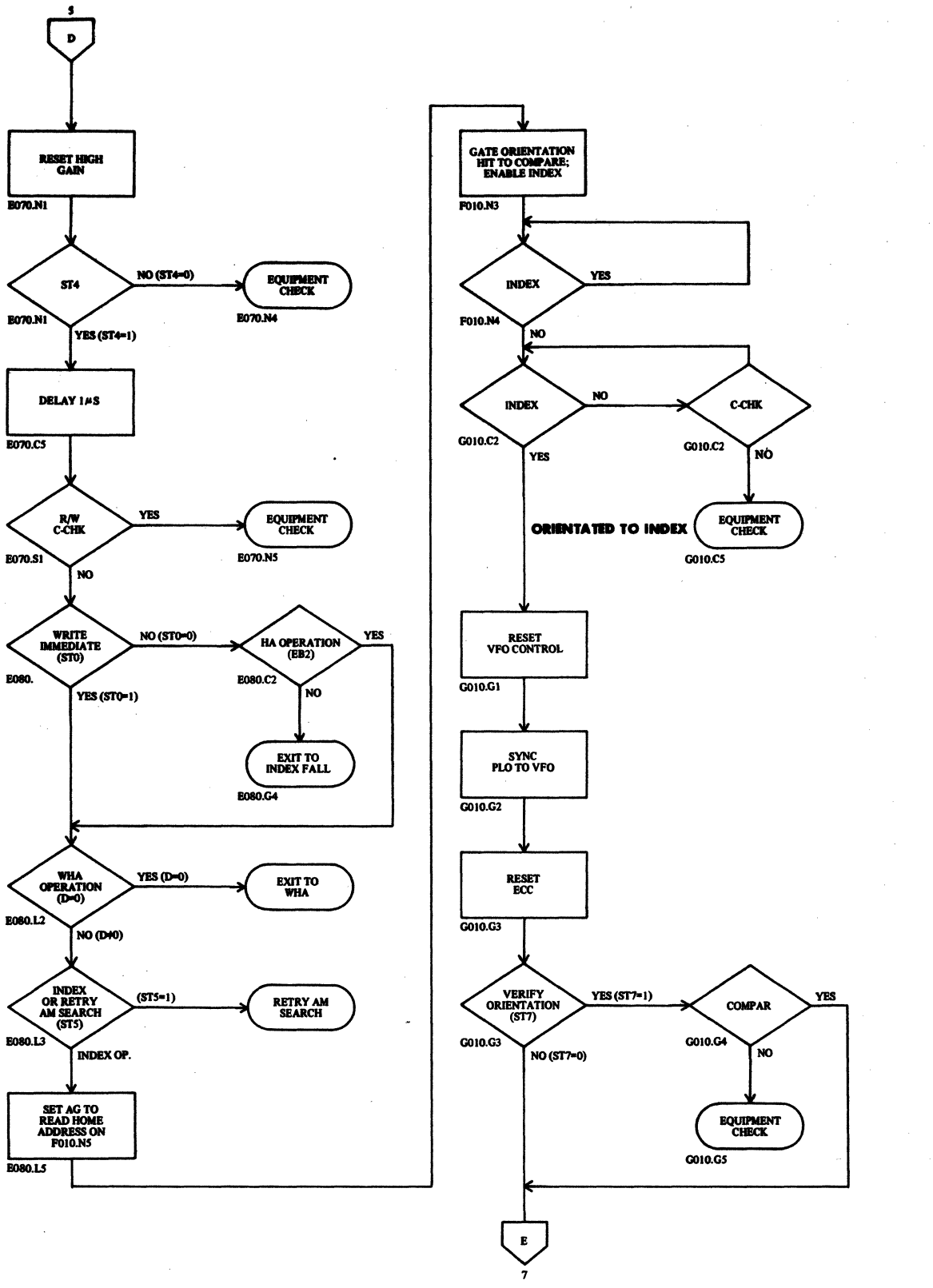
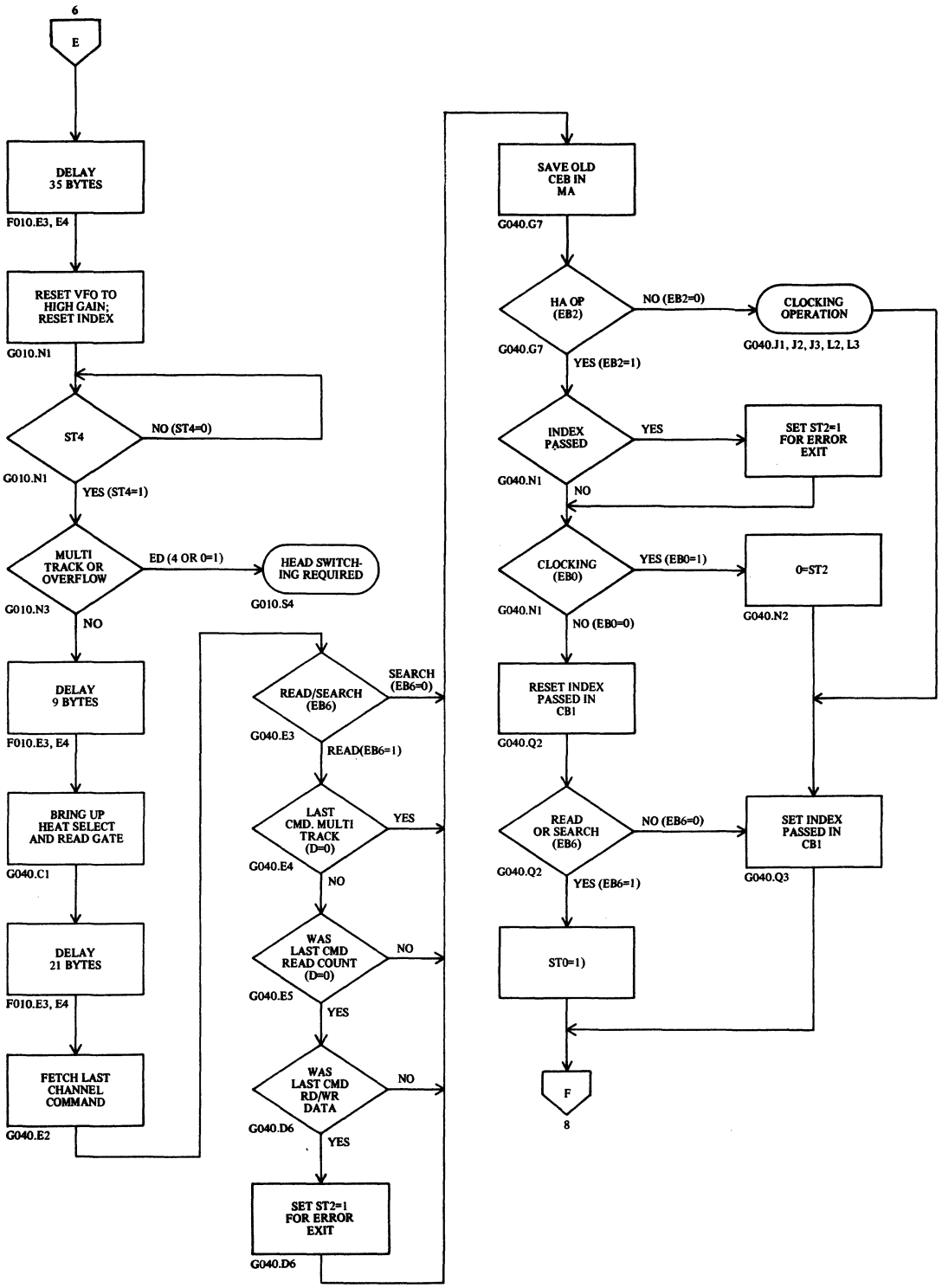


Figure 3-68. Read Home Address (Sheet 6 of 12)



8B113-7

Figure 3-68. Read Home Address (Sheet 7 of 12)

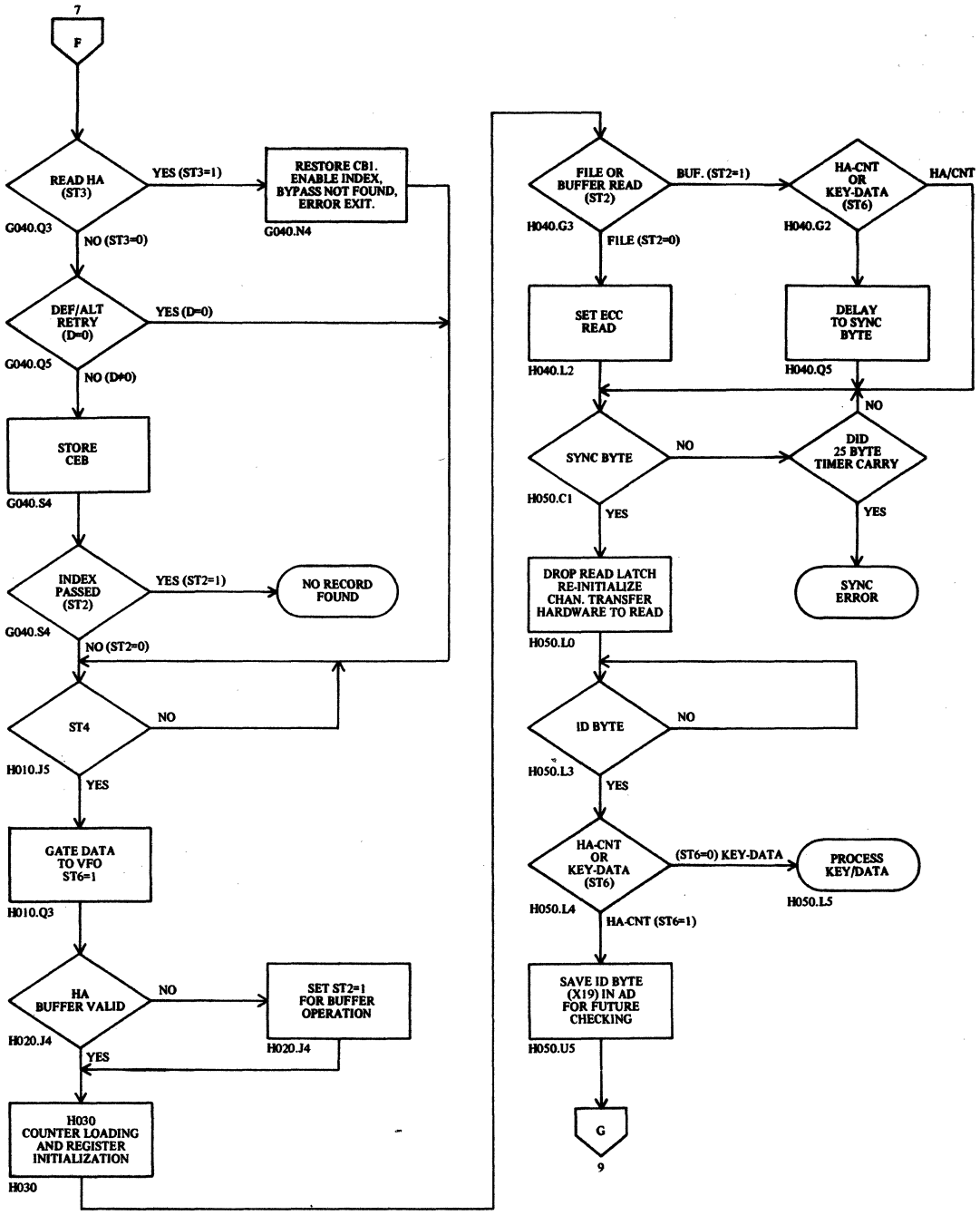


Figure 3-68. Read Home Address (Sheet 8 of 12)

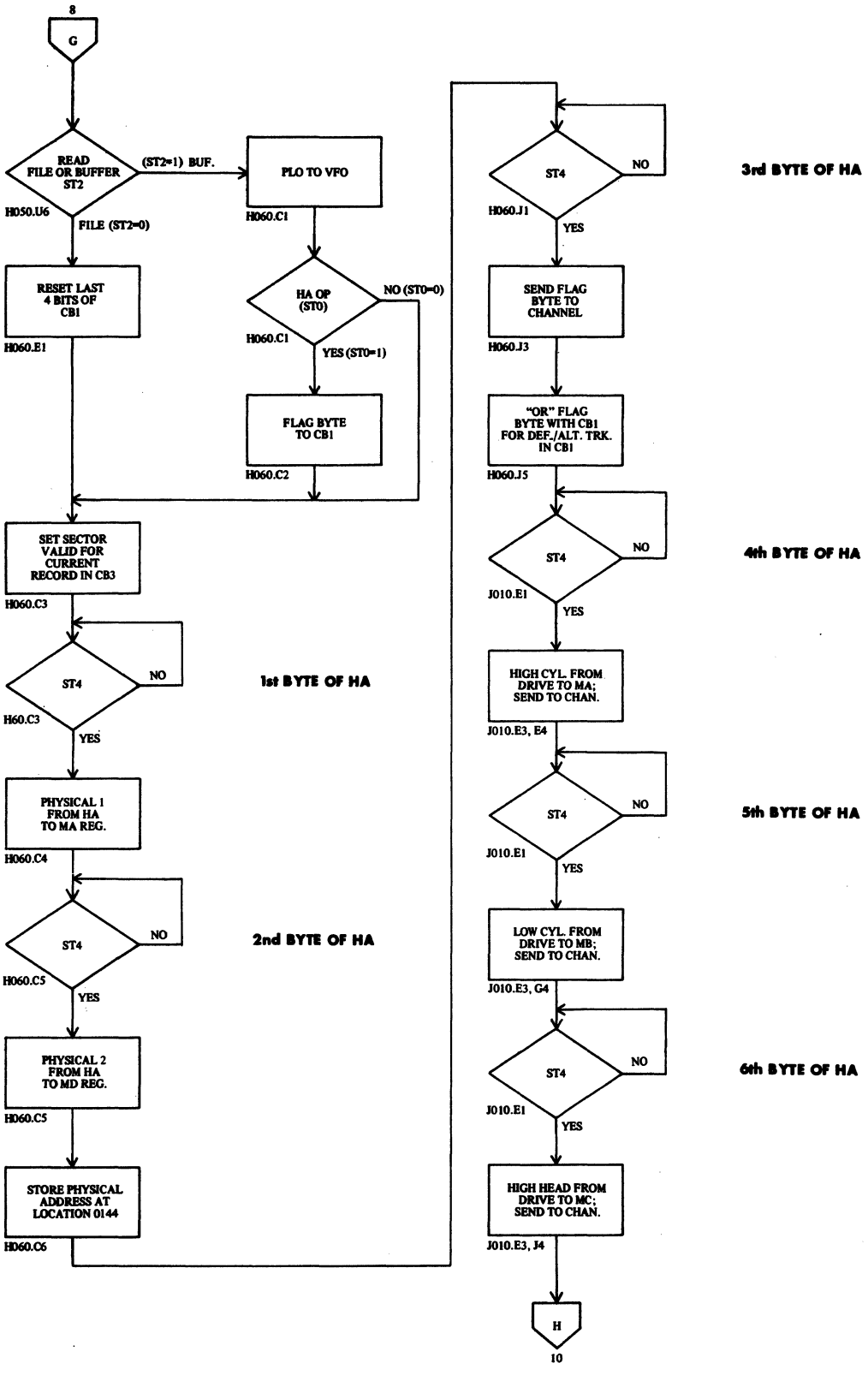


Figure 3-68. Read Home Address (Sheet 9 of 12)

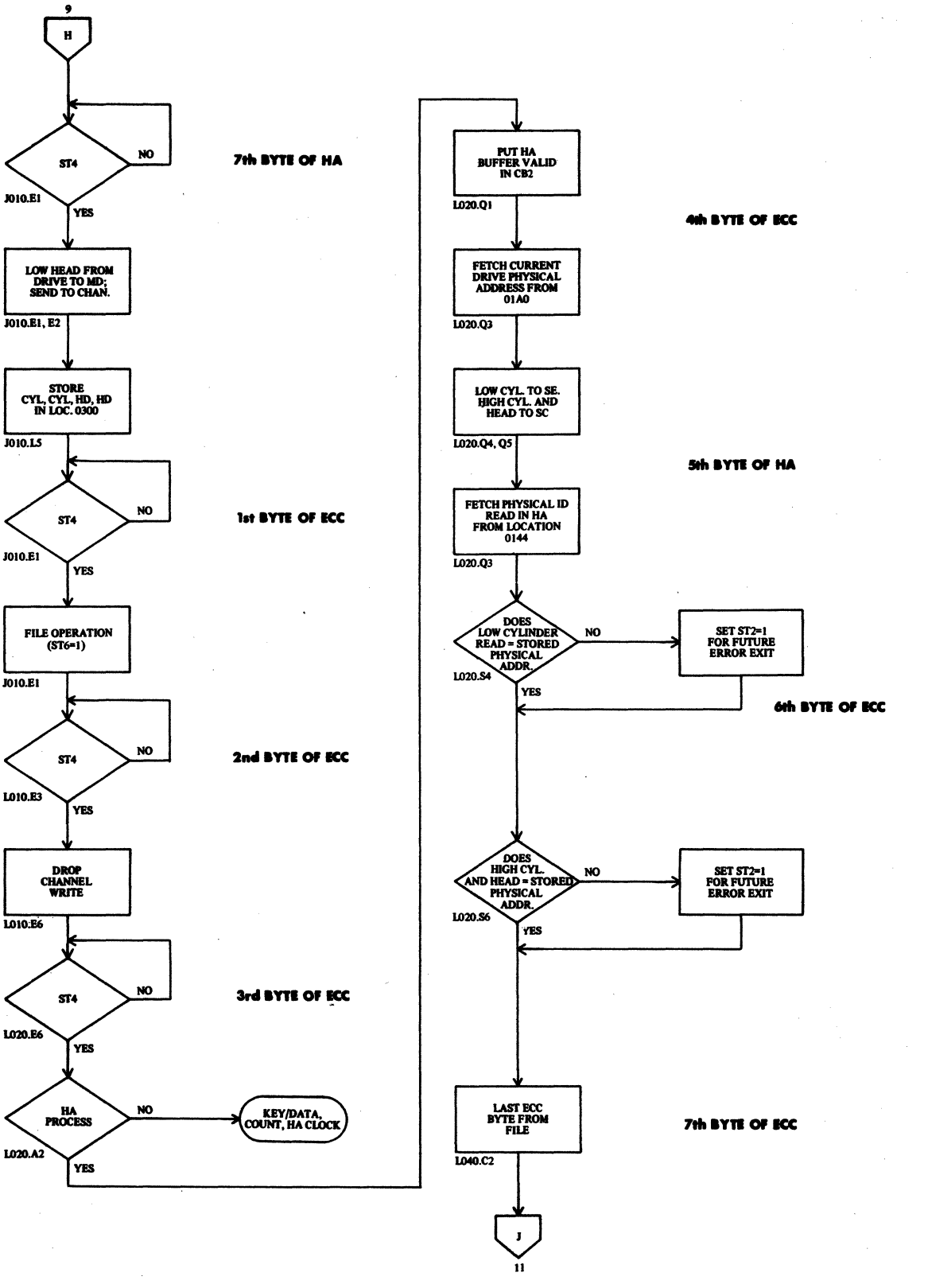


Figure 3-68. Read Home Address (Sheet 10 of 12)

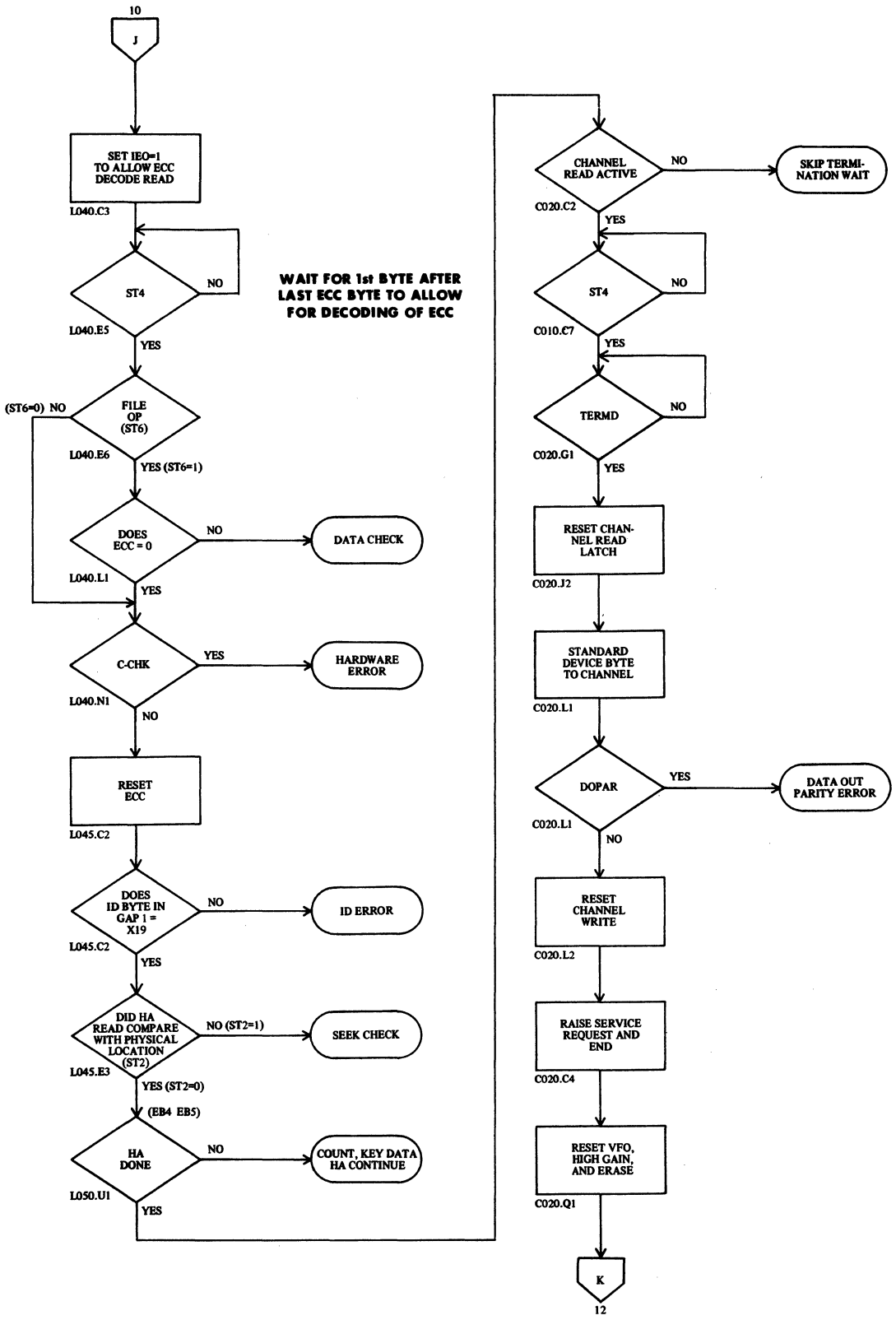
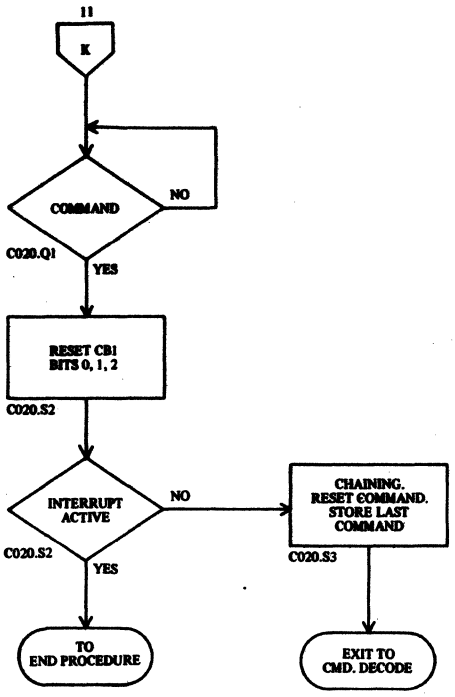


Figure 3-68. Read Home Address (Sheet 11 of 12)



8B113-12

Figure 3-68. Read Home Address (Sheet 12 of 12)

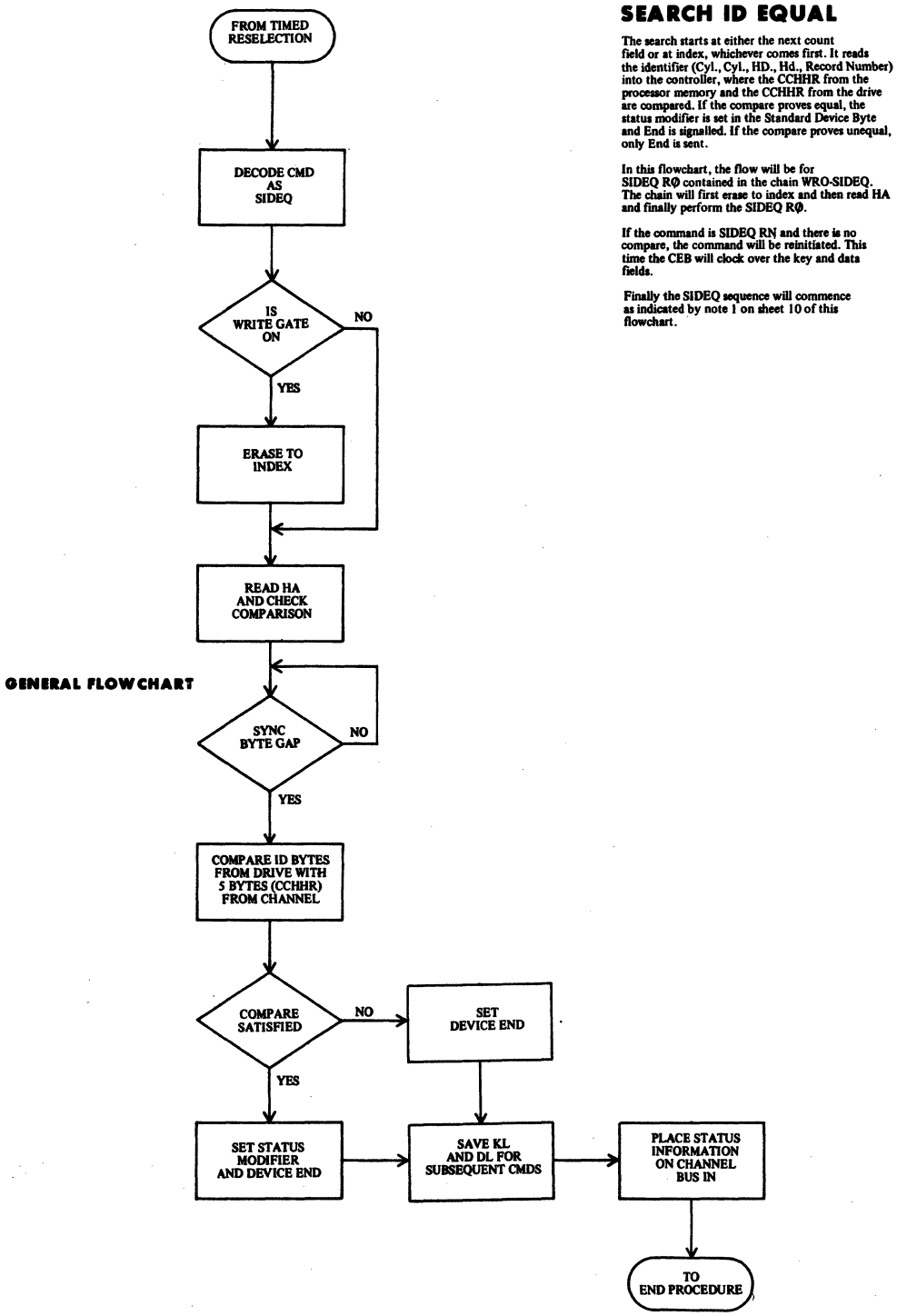


Figure 3-69. Search ID Equal (Sheet 1 of 18)

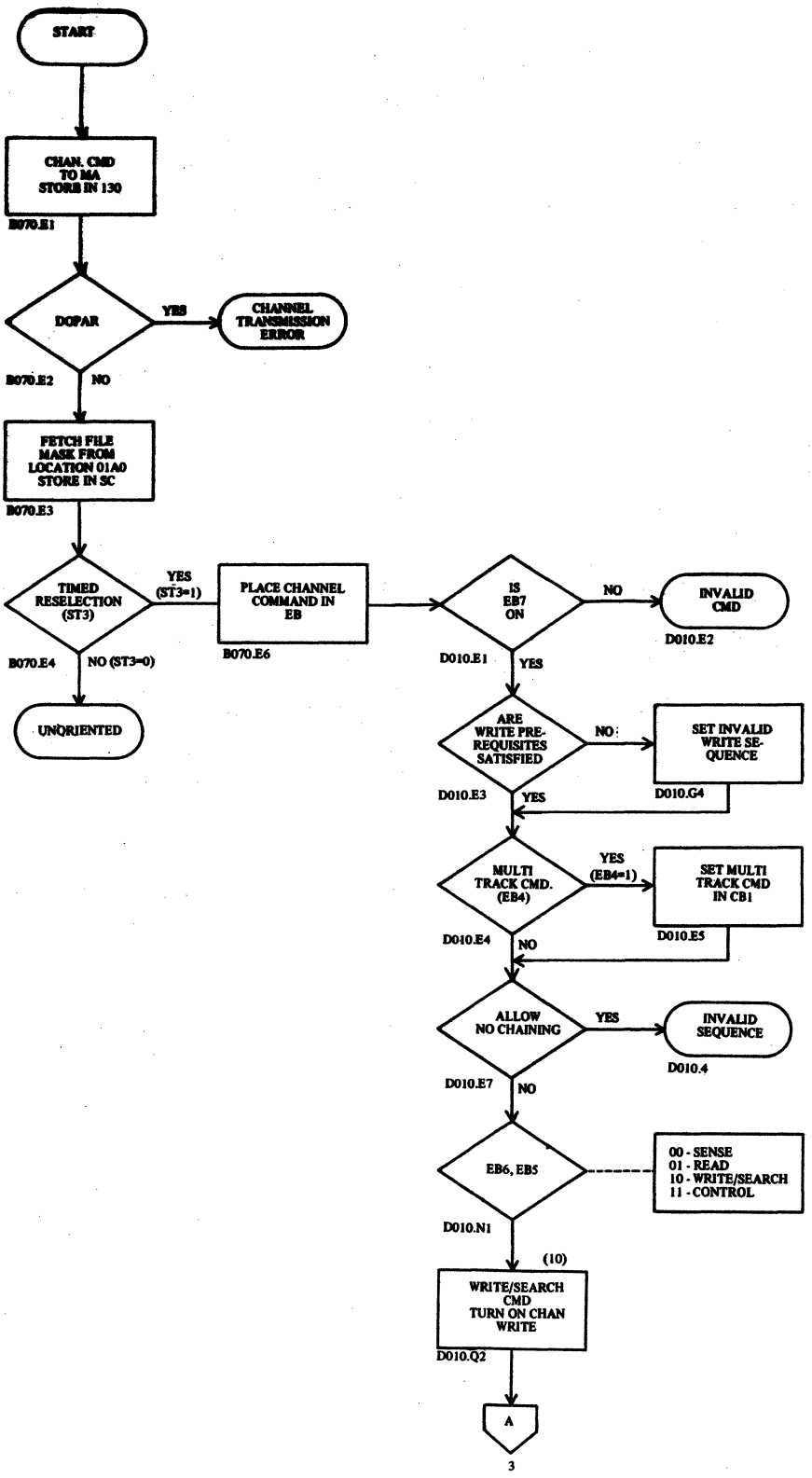


Figure 3-69. Search ID Equal (Sheet 2 of 18)

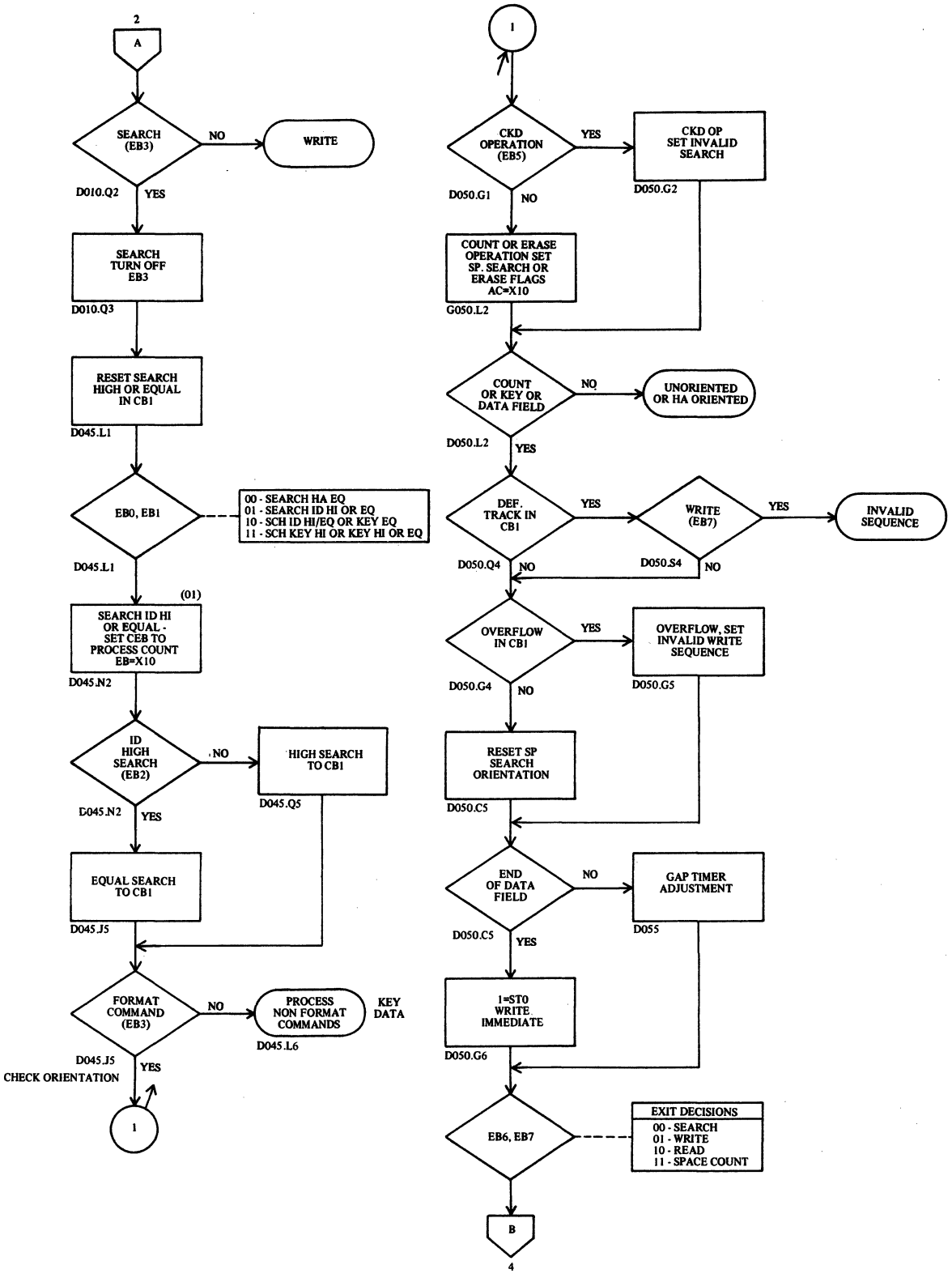


Figure 3-69. Search ID Equal (Sheet 3 of 18)

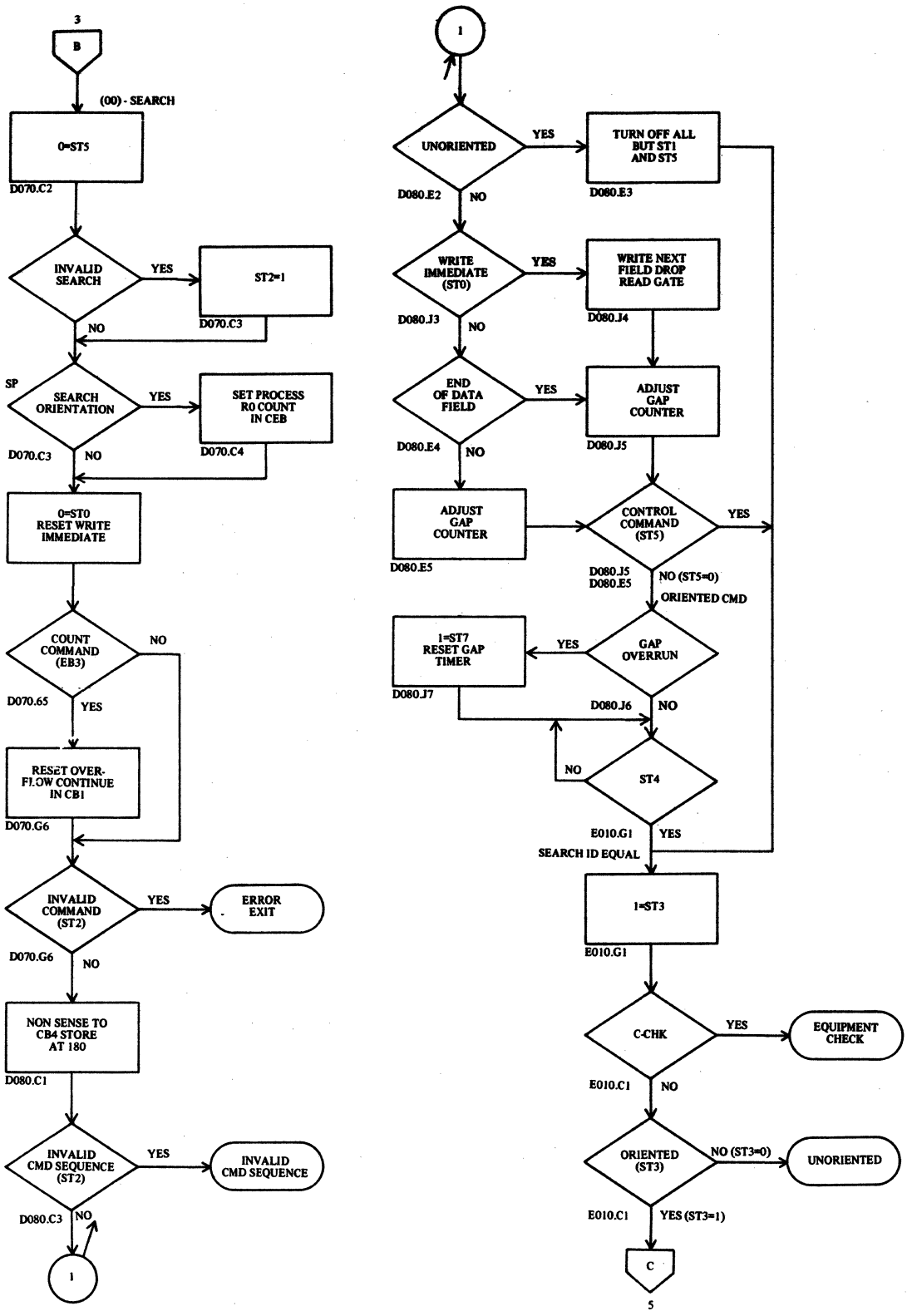


Figure 3-69. Search ID Equal (Sheet 4 of 18)

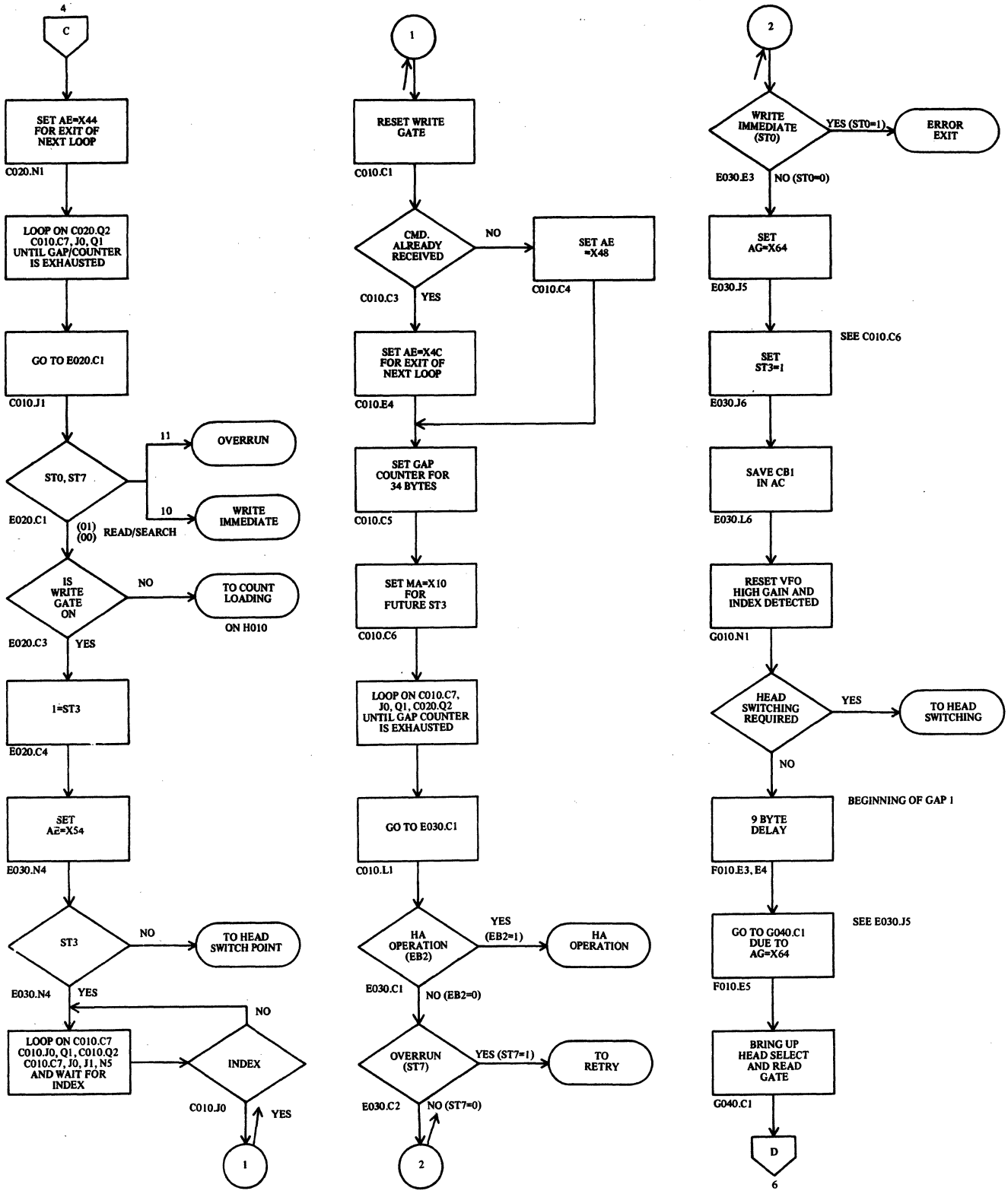


Figure 3-69. Search ID Equal (Sheet 5 of 18)

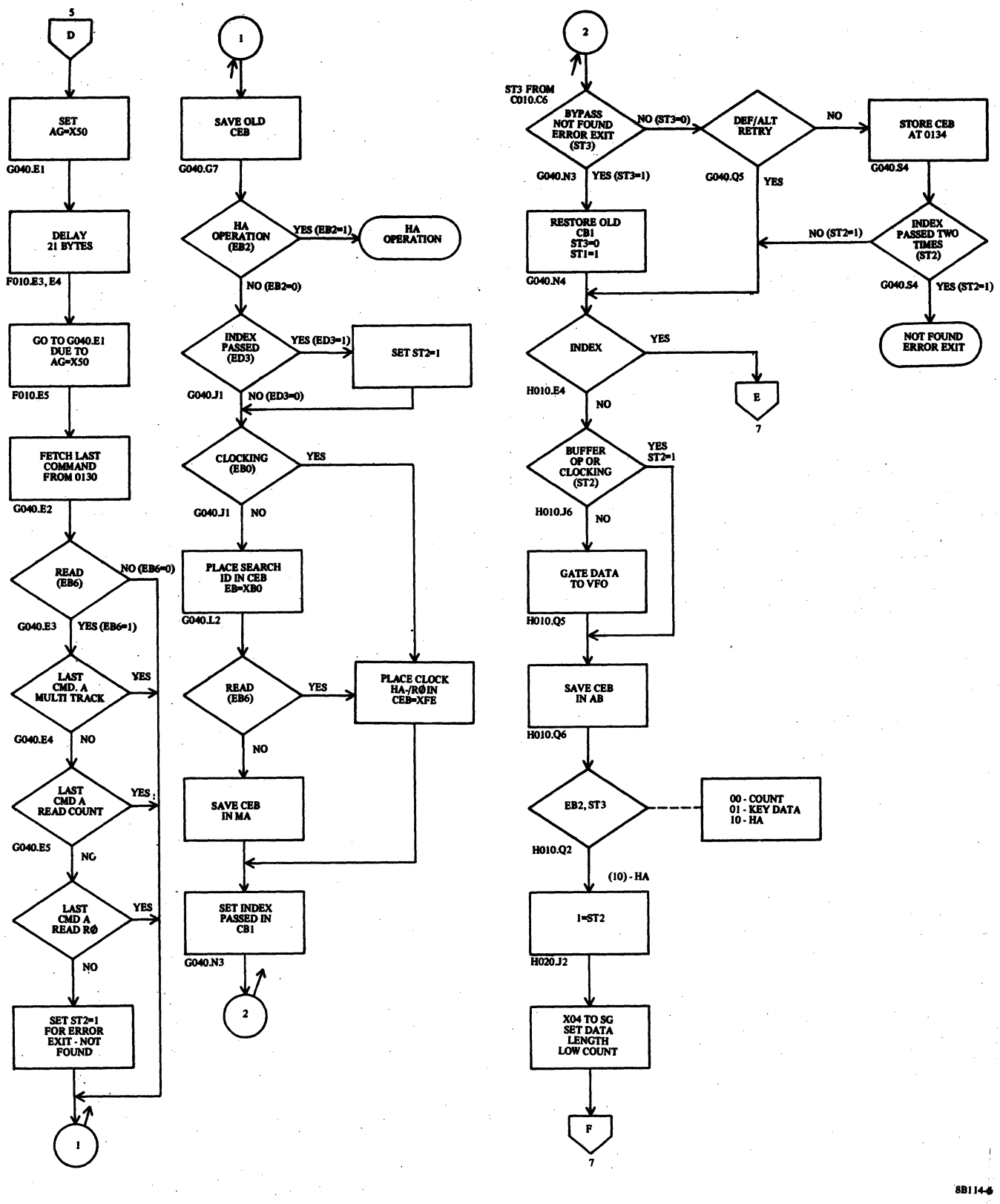


Figure 3-69. Search ID Equal (Sheet 6 of 18)

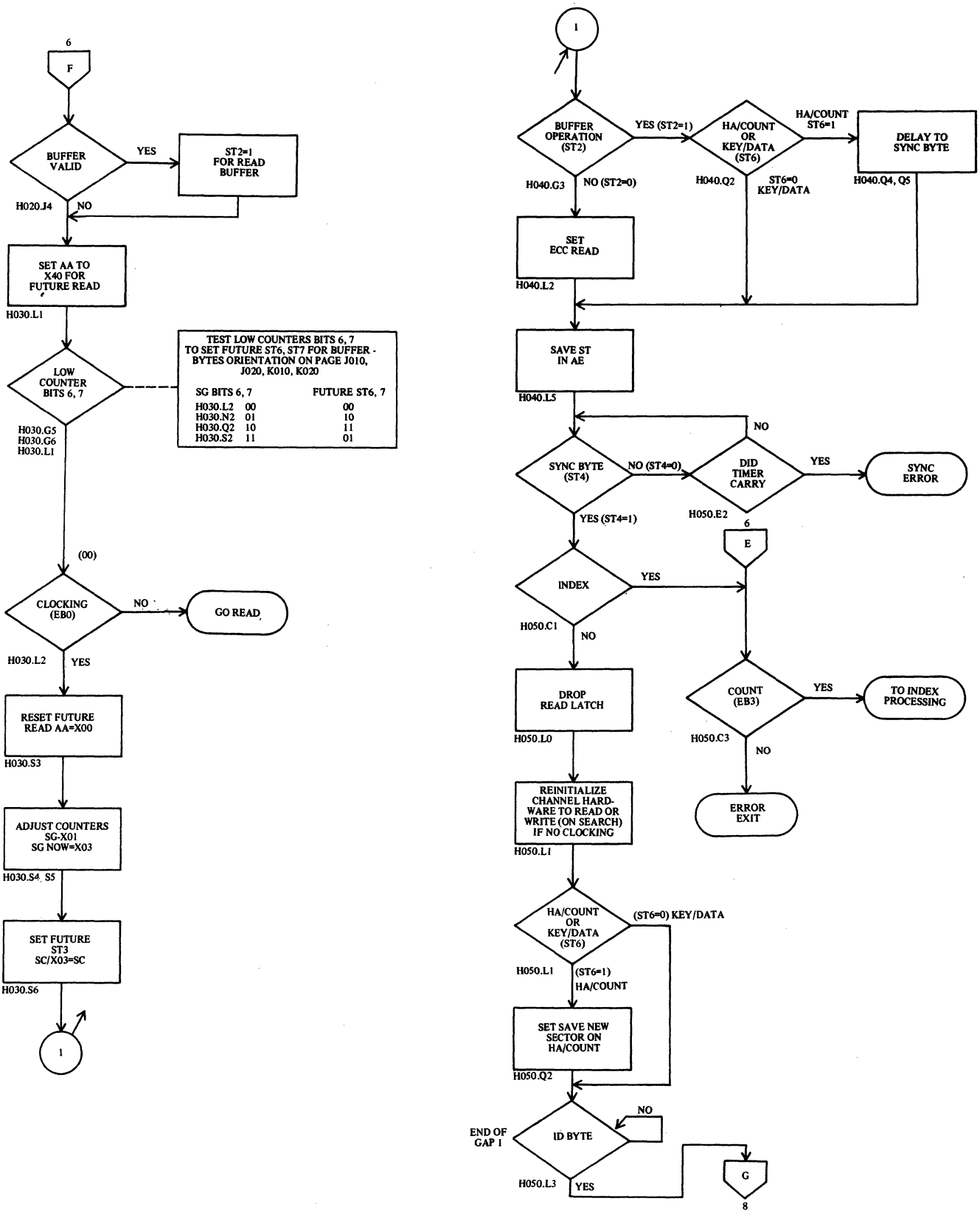


Figure 3-69. Search ID Equal (Sheet 7 of 18)

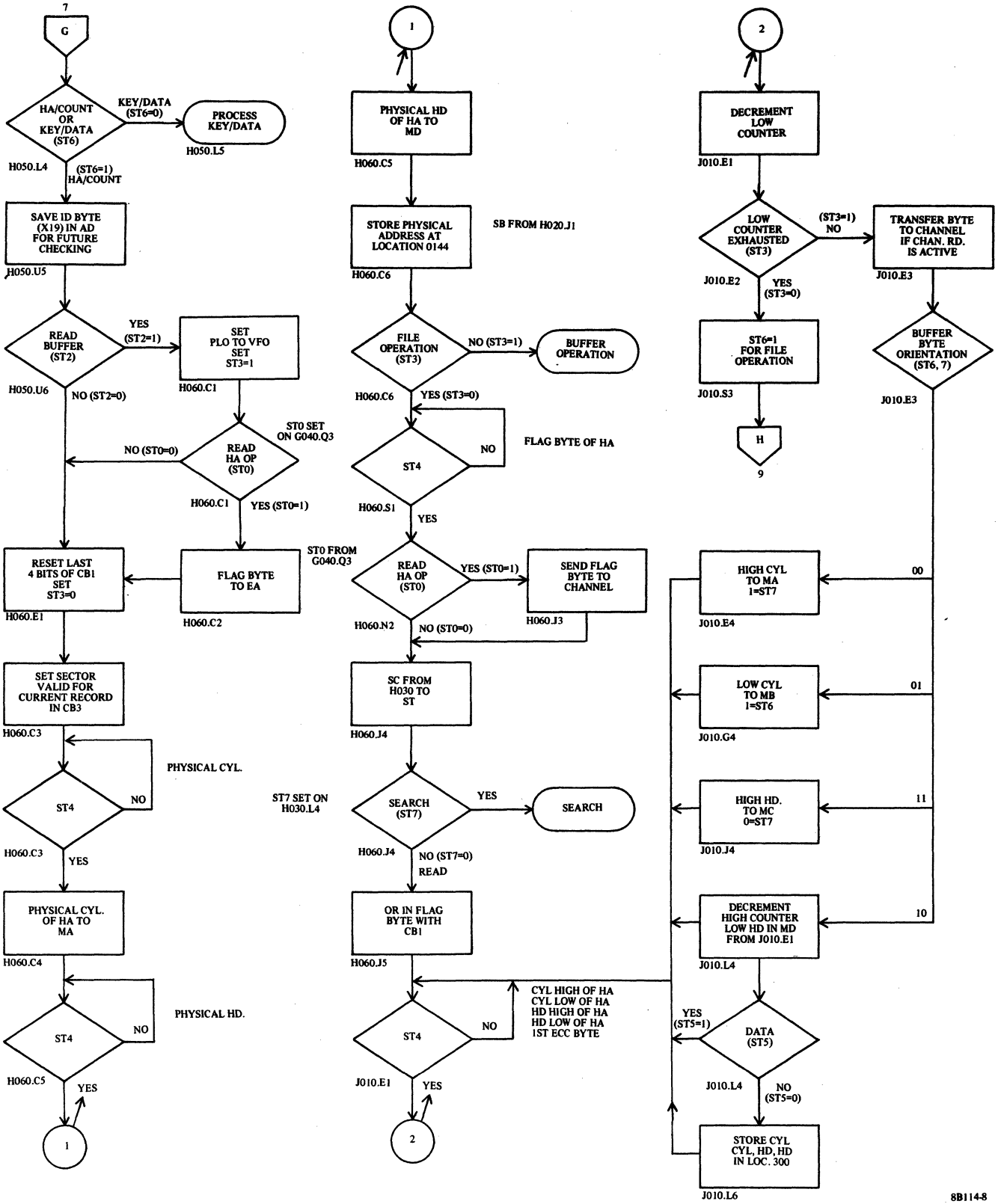


Figure 3-69. Search ID Equal (Sheet 8 of 18)

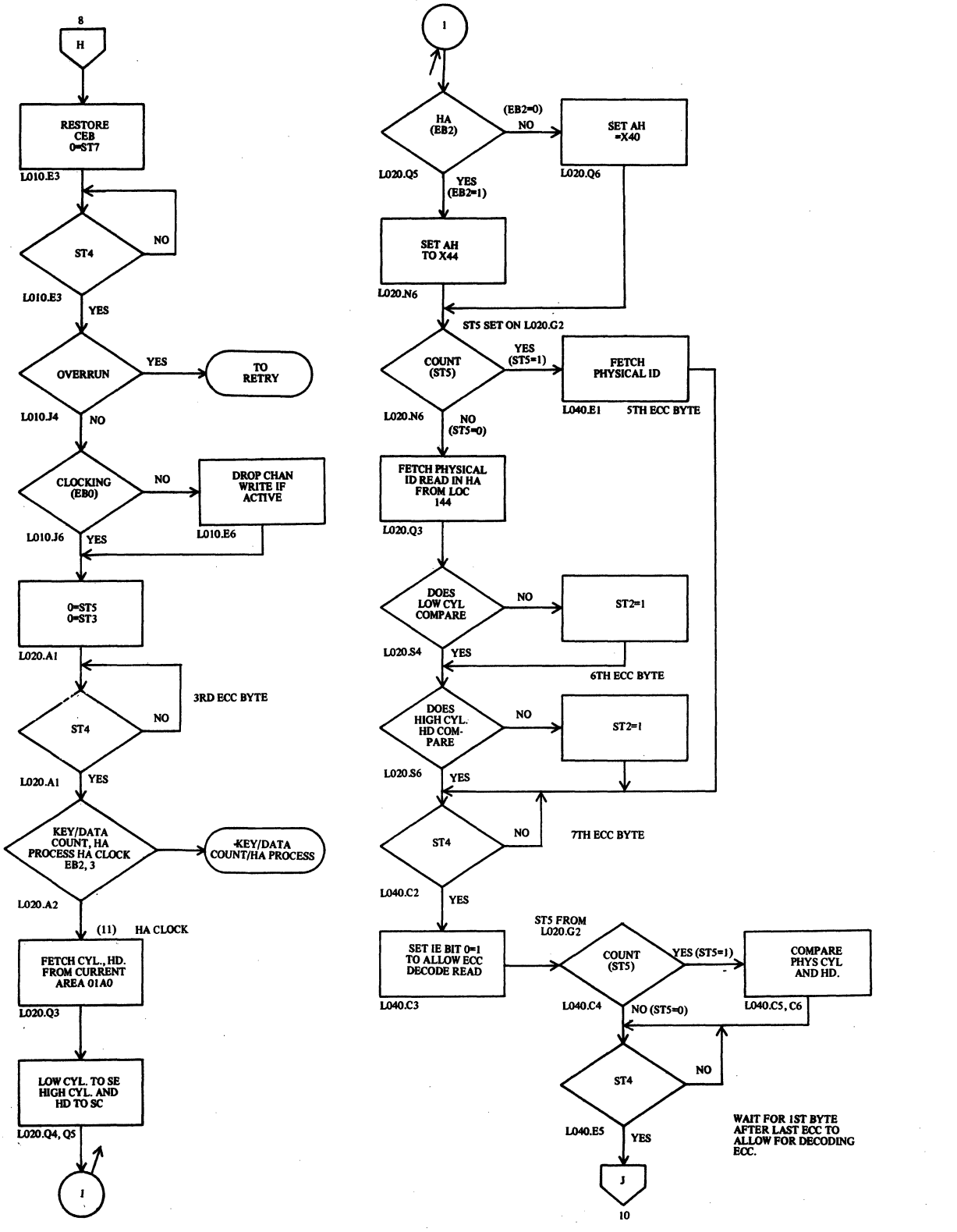


Figure 3-69. Search ID Equal (Sheet 9 of 18)

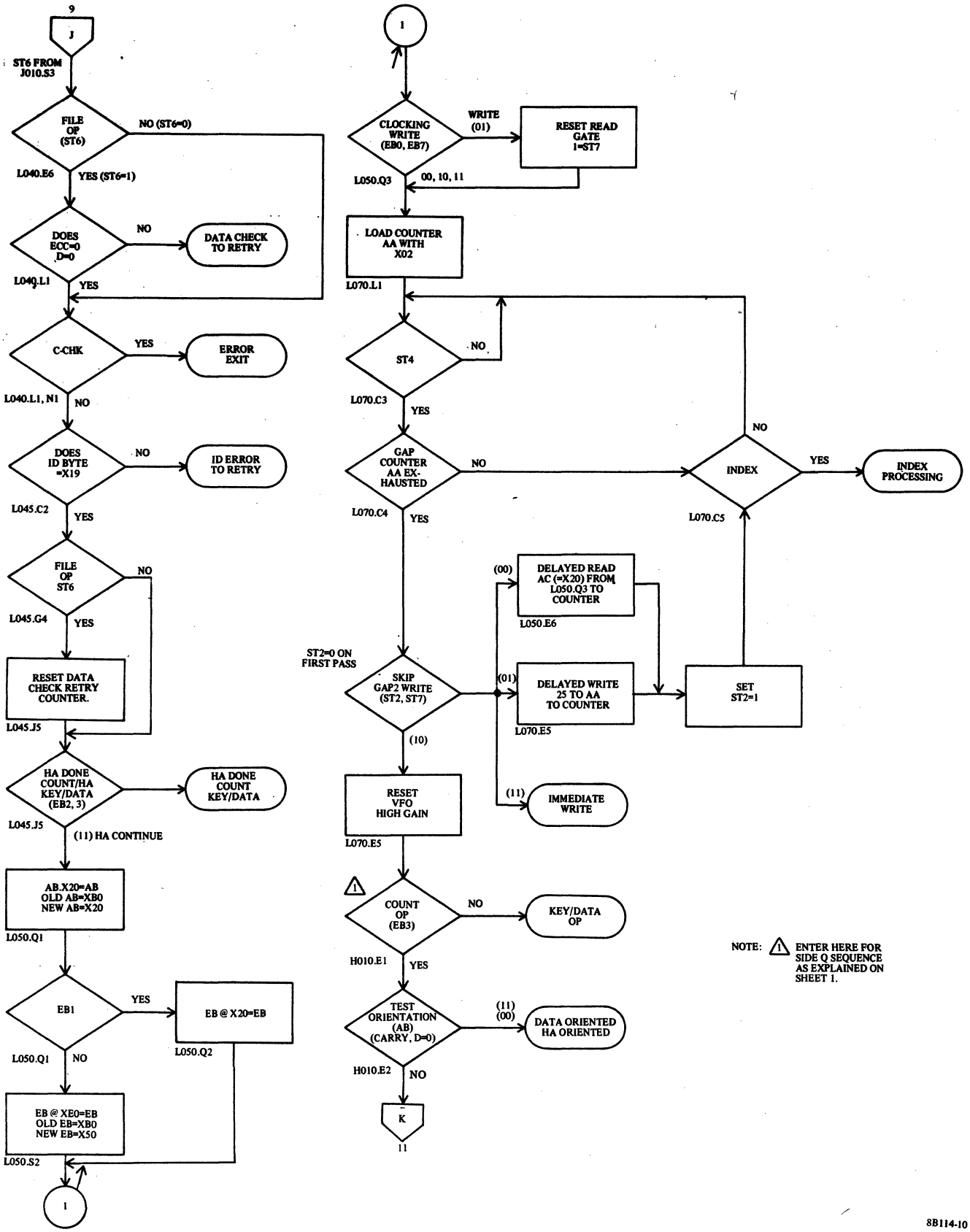


Figure 3-69. Search ID Equal (Sheet 10 of 18)

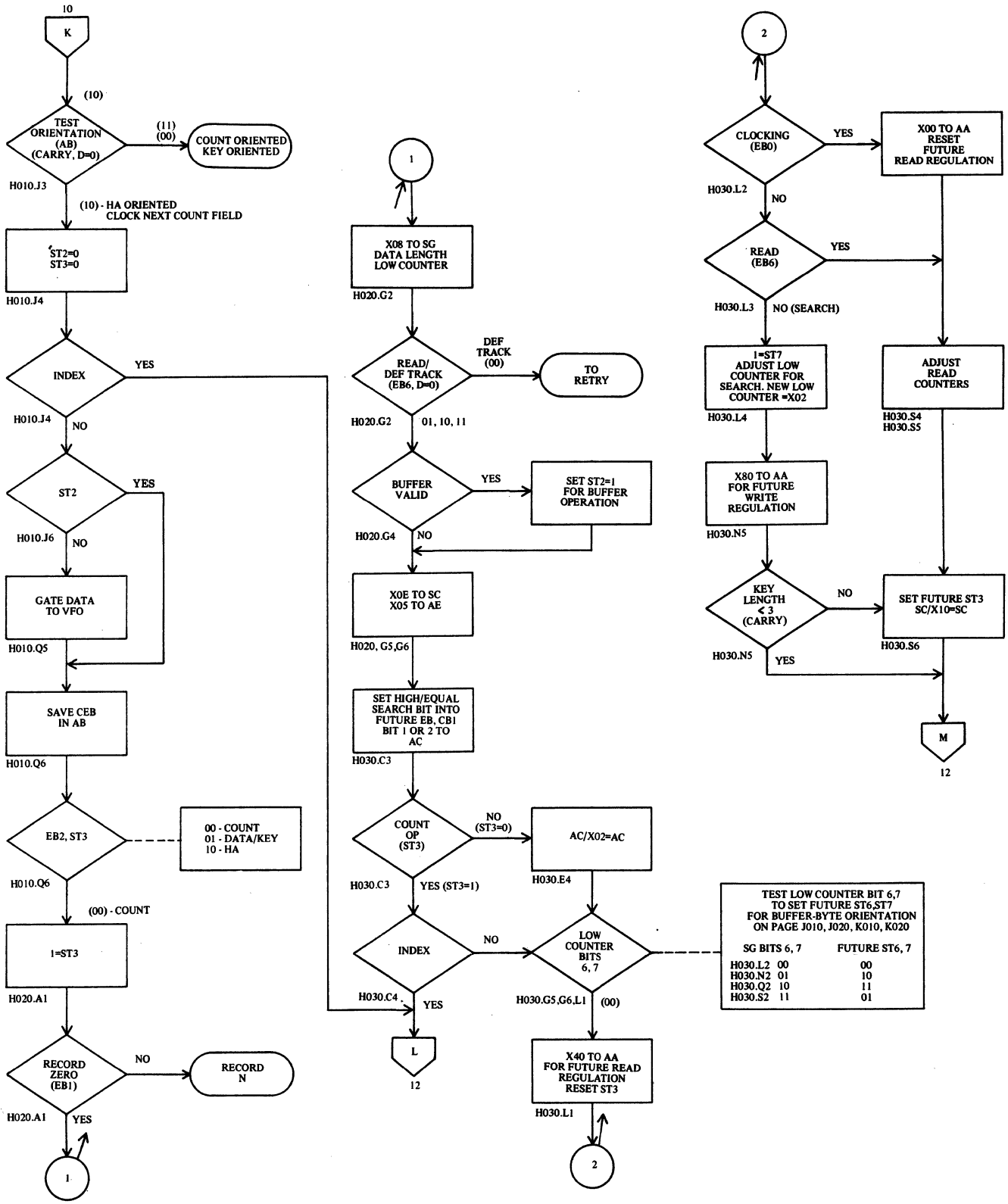


Figure 3-69. Search ID Equal (Sheet 11 of 18)

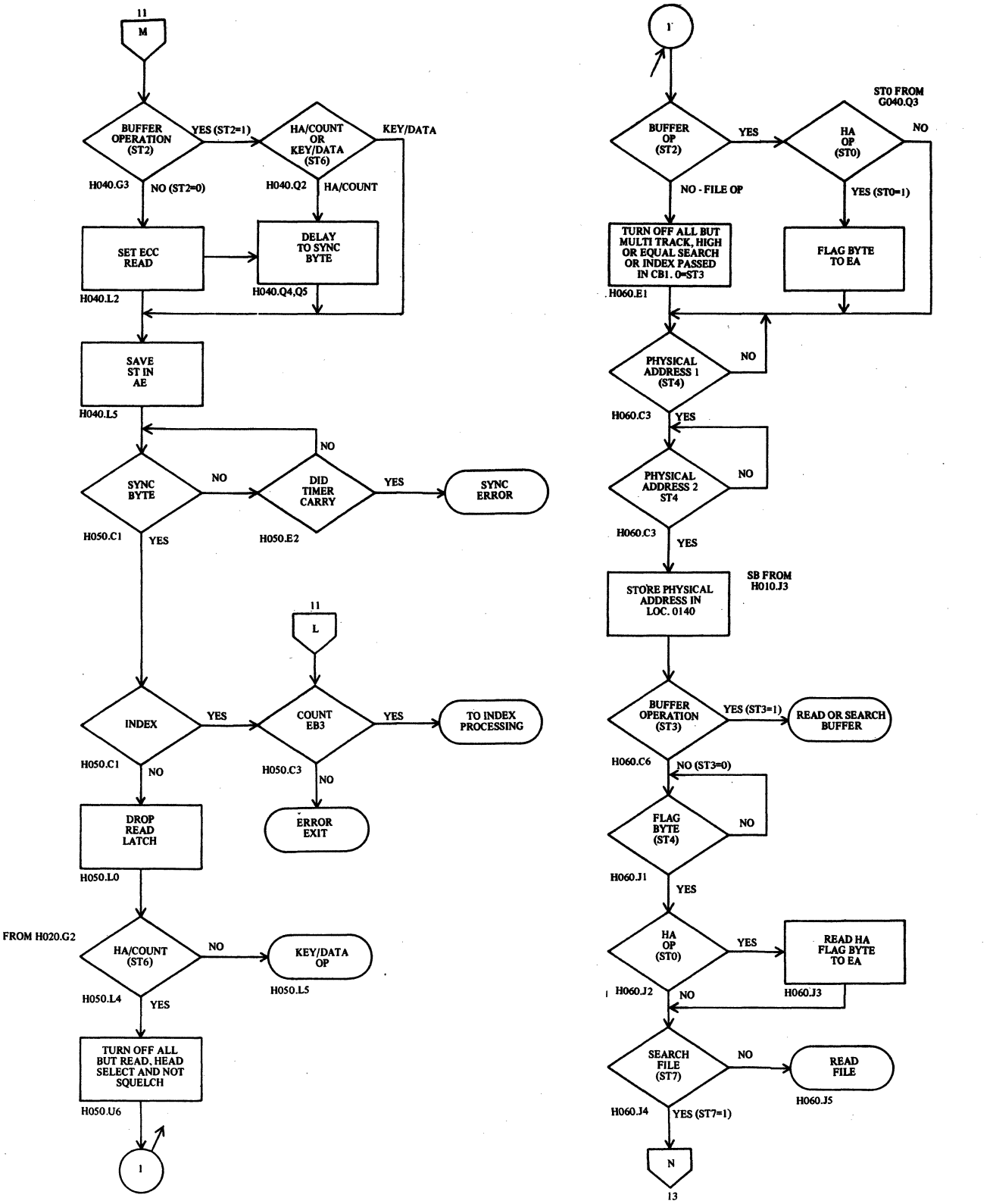
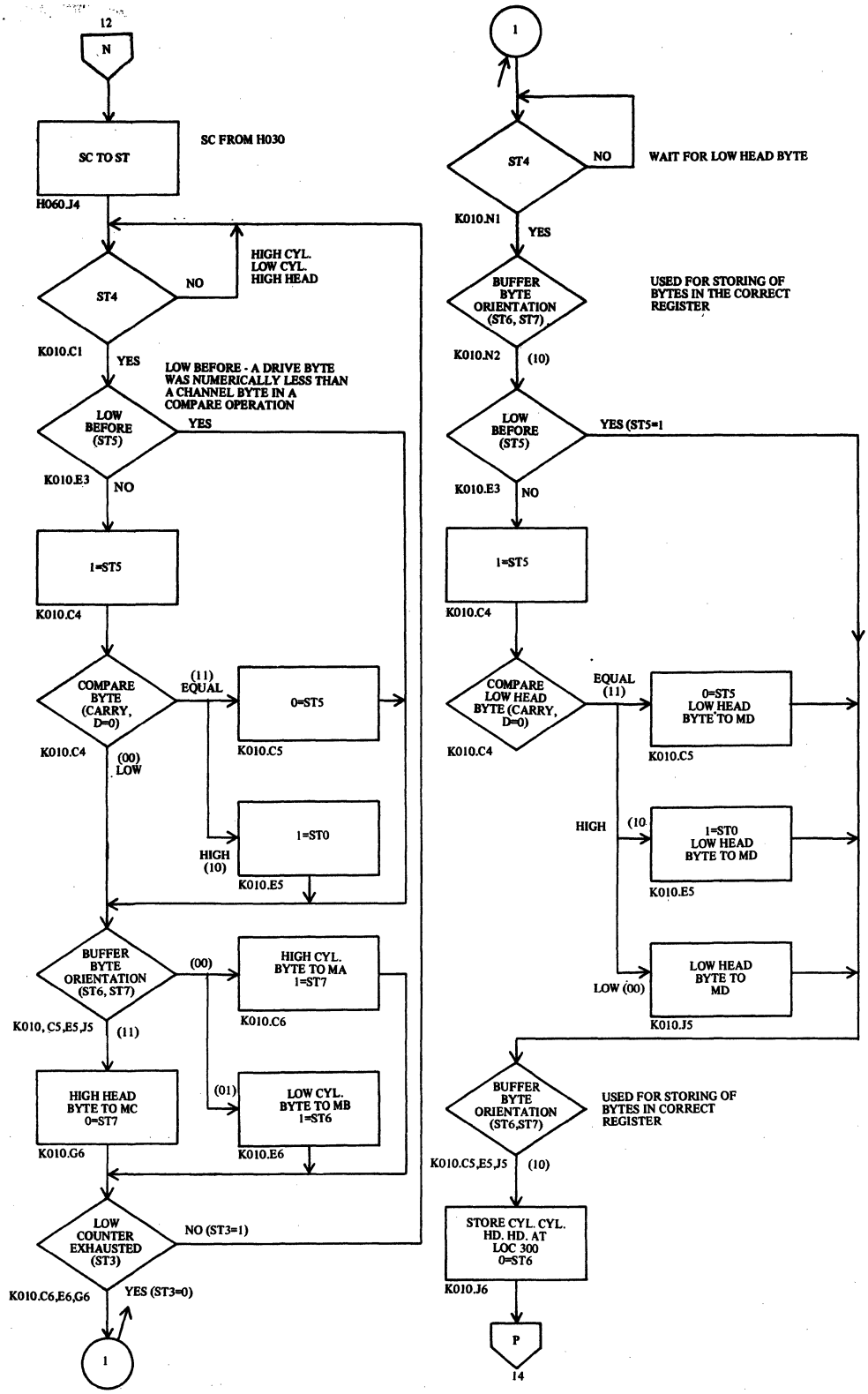


Figure 3-69. Search ID Equal (Sheet 12 of 18)



8B114-13

Figure 3-69. Search ID Equal (Sheet 13 of 18)

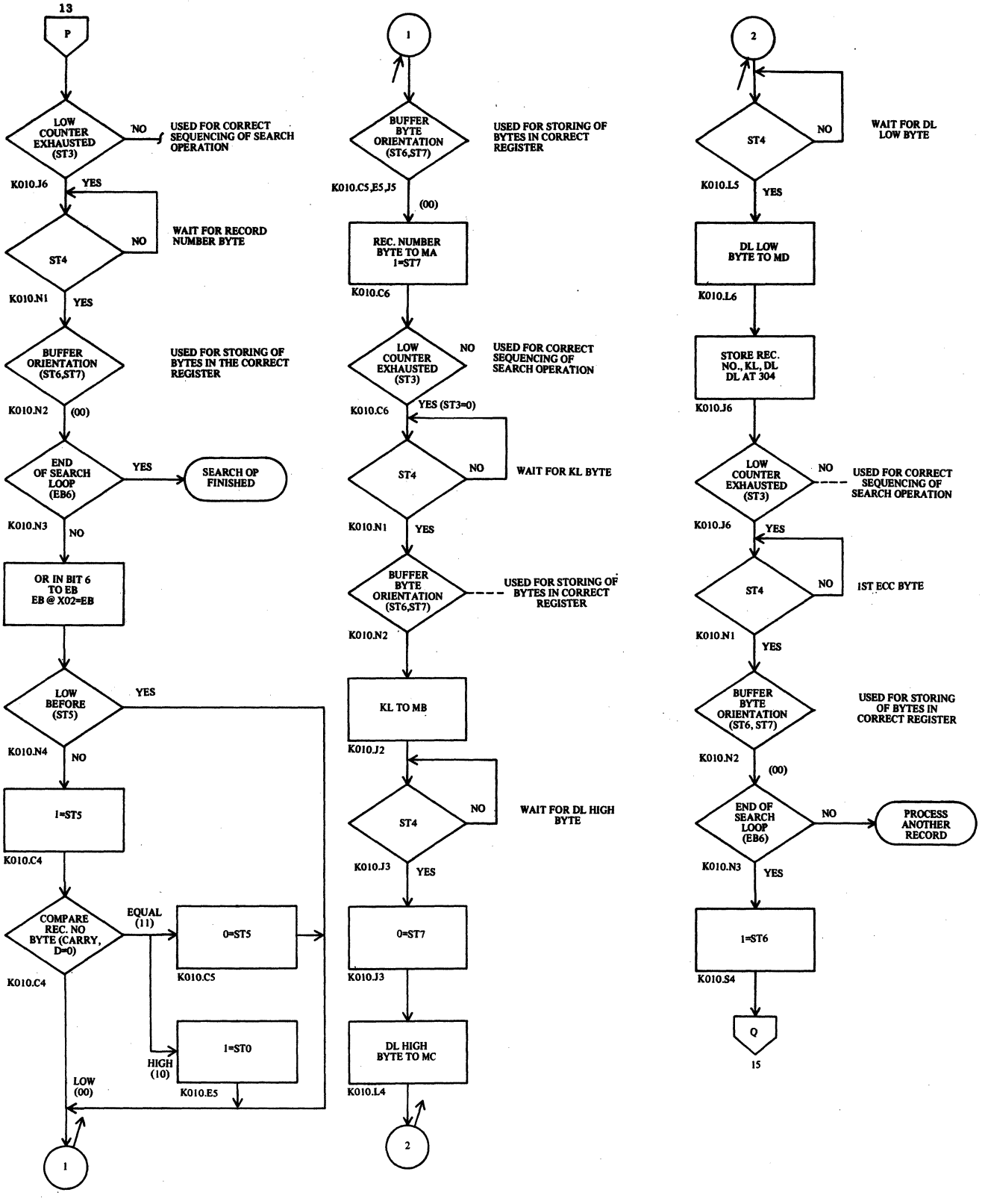
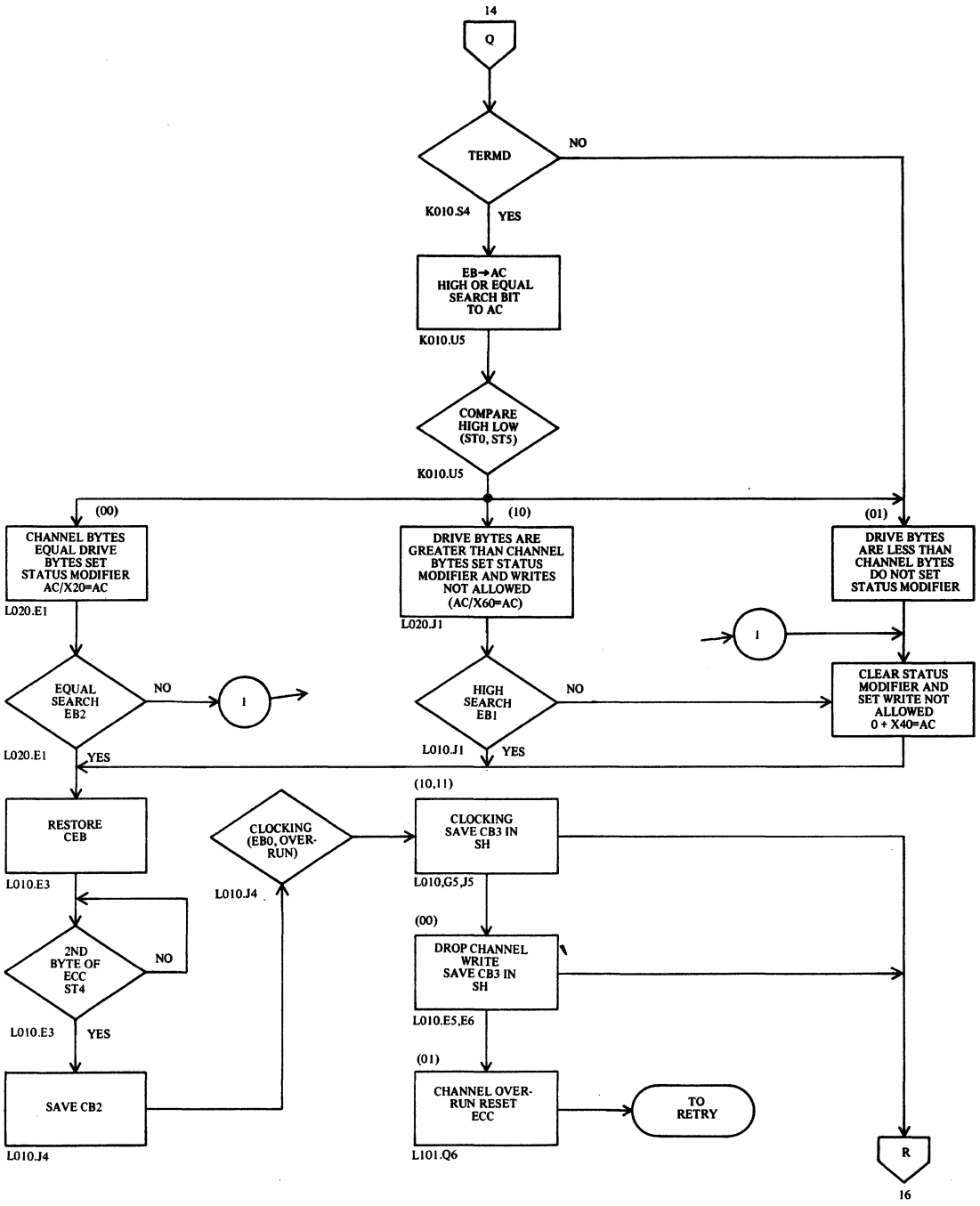


Figure 3-69. Search ID Equal (Sheet 14 of 18)



8B114-15

Figure 3-69. Search ID Equal (Sheet 15 of 18)

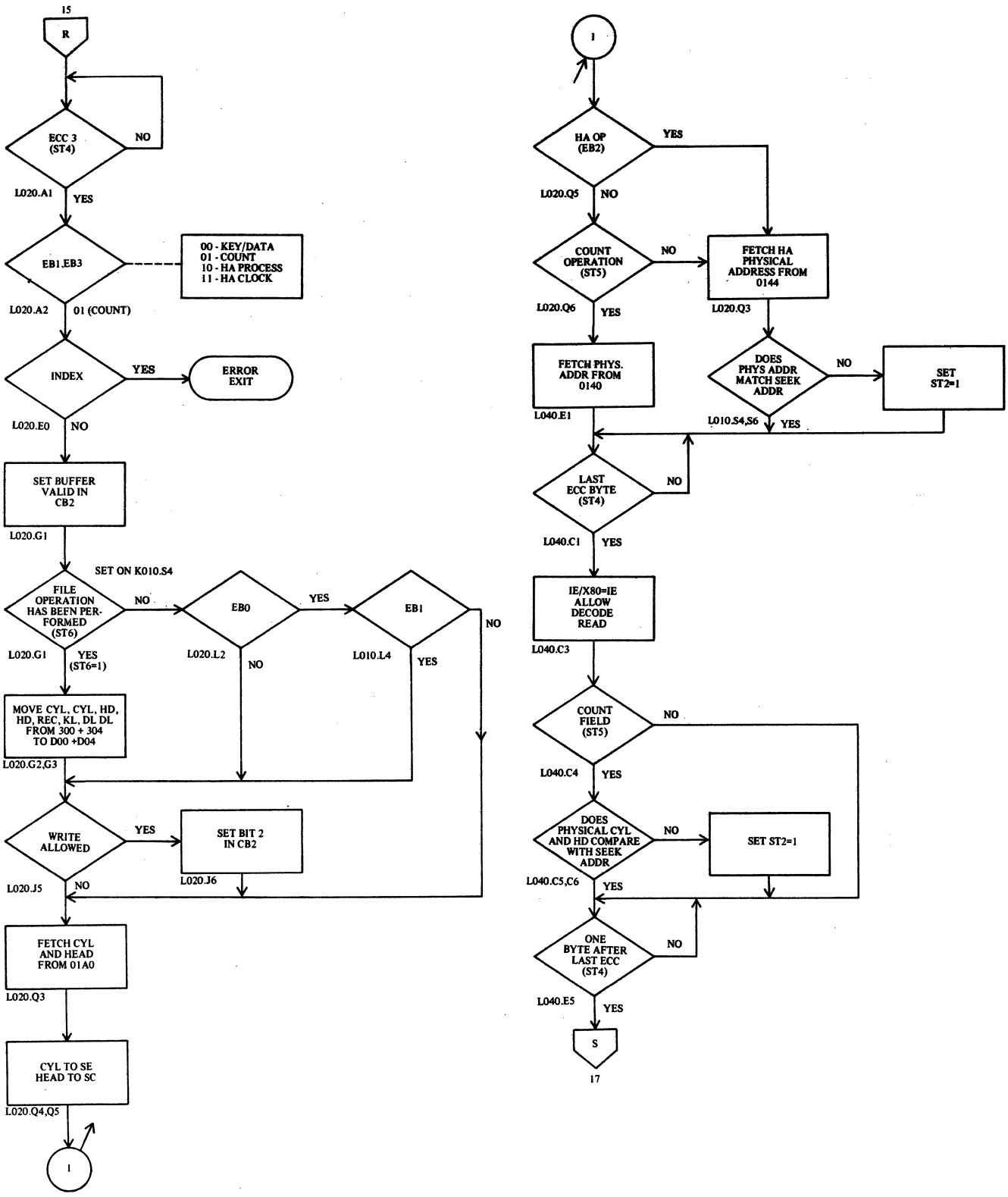


Figure 3-69. Search ID Equal (Sheet 16 of 18)

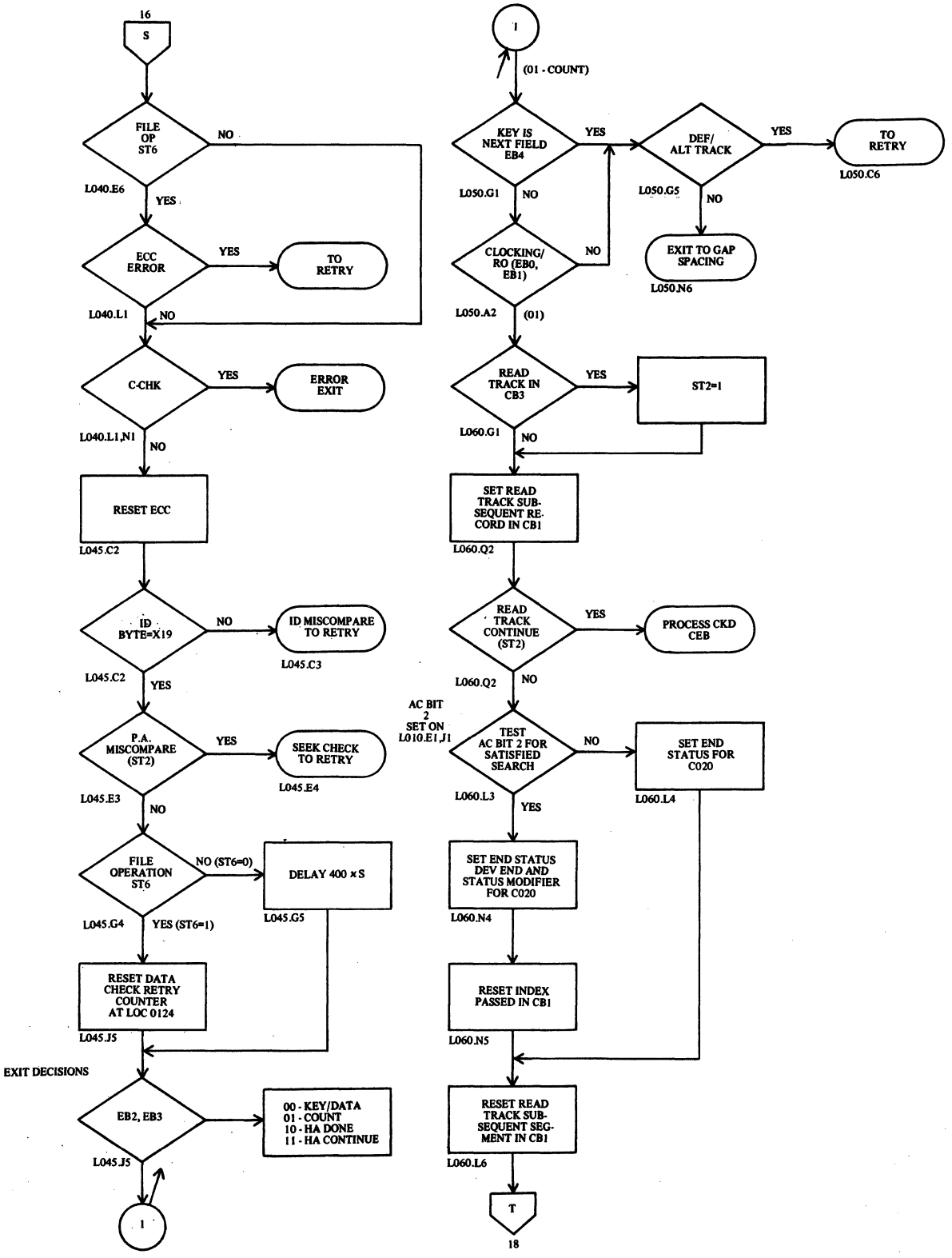
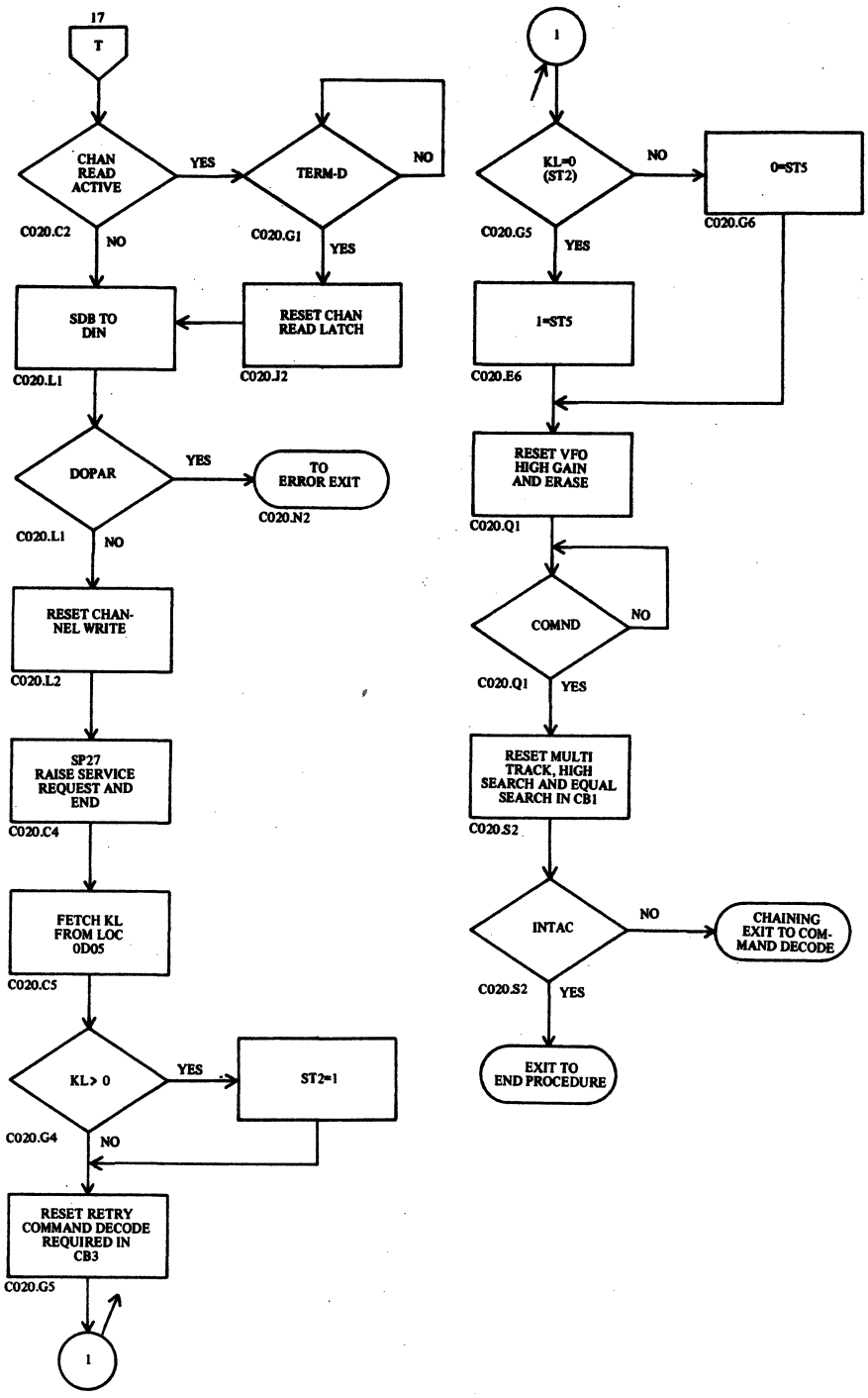


Figure 3-69. Search ID Equal (Sheet 17 of 18)



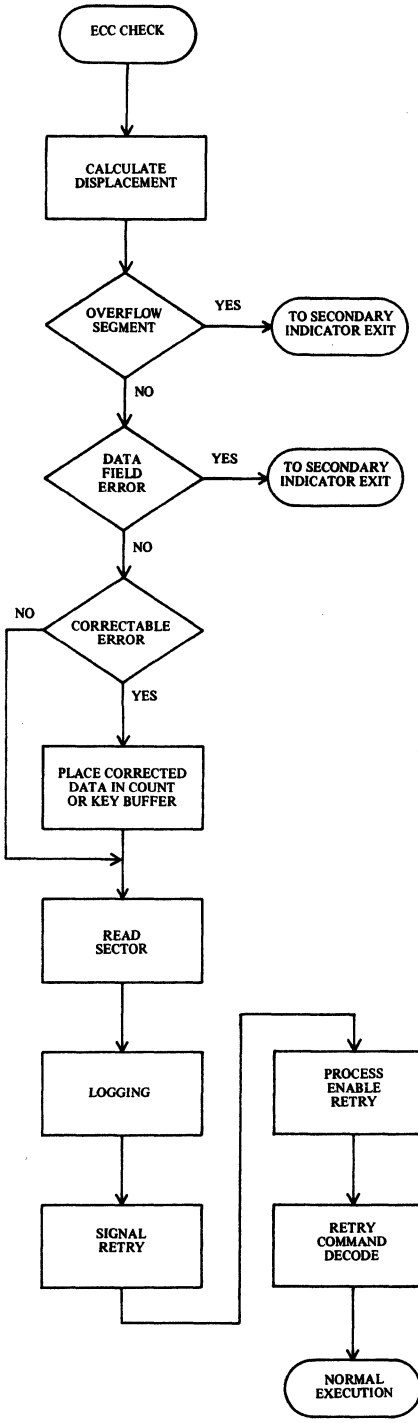
8B114-18

Figure 3-69. Search ID Equal (Sheet 18 of 18)

If the data error is detected at the completion of a read or search in the count or key field and is correctable, the control unit signals retry status to the channel. Then the control unit reorients on the failing track and signals device end to the channel to begin retry. The failing field, which was buffered in control storage, is corrected by the control unit. A read in the count or key field results in the corrected data from control storage to the channel and then the CCW continues.

This flowchart assumes that an ECC correctable error has been detected in a count field operation. The following preparations were made during the read operation:

1. The sector count was saved
2. The data was saved
3. The ECC bytes were read
4. The operation code (CEB) was saved



GENERAL FLOWCHART

8B115-1

Figure 3-70. Command Retry for a Correctable Data Check in the Count Field (Sheet 1 of 18)

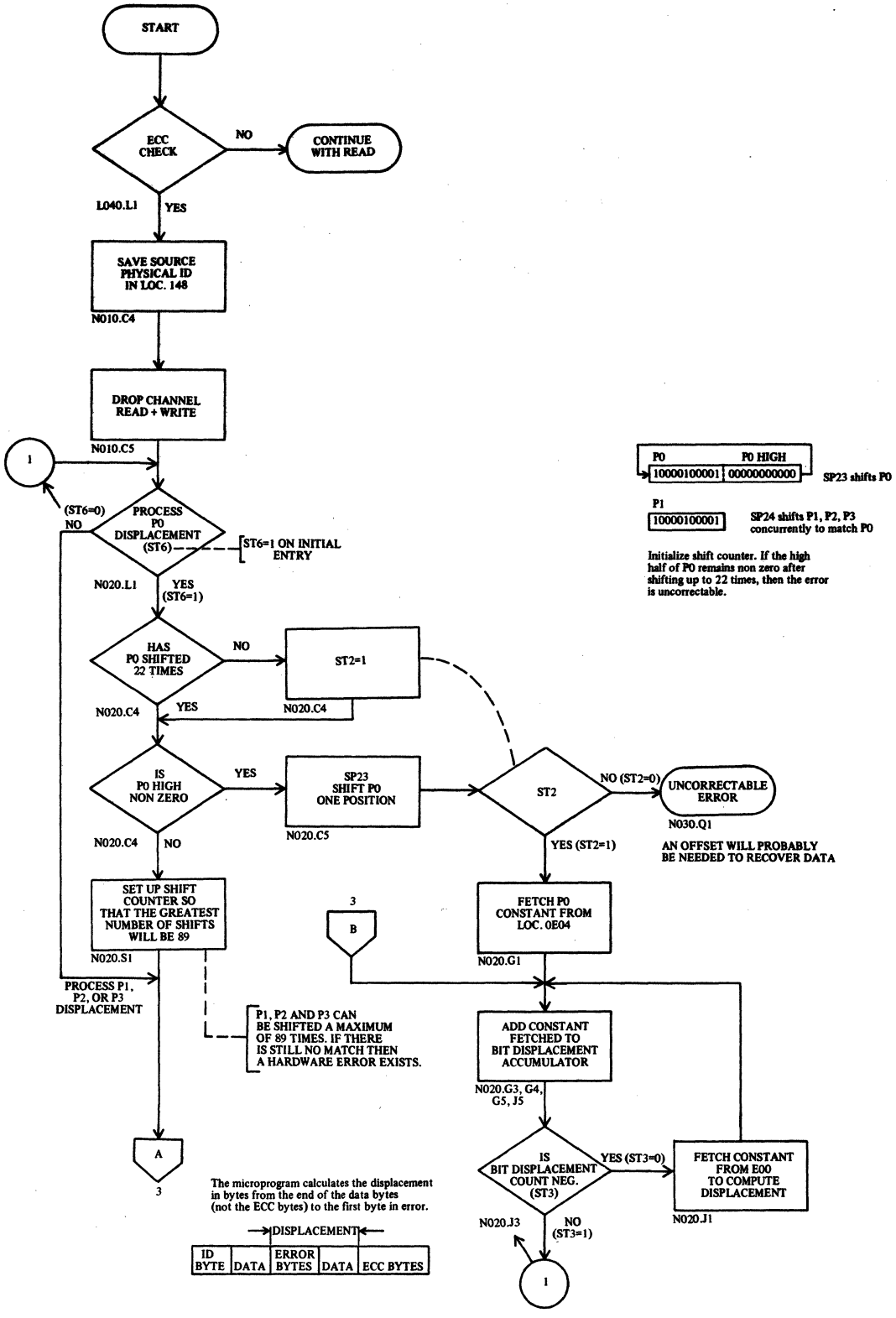


Figure 3-70. Command Retry for a Correctable Data Check in the Count Field (Sheet 2 of 18)

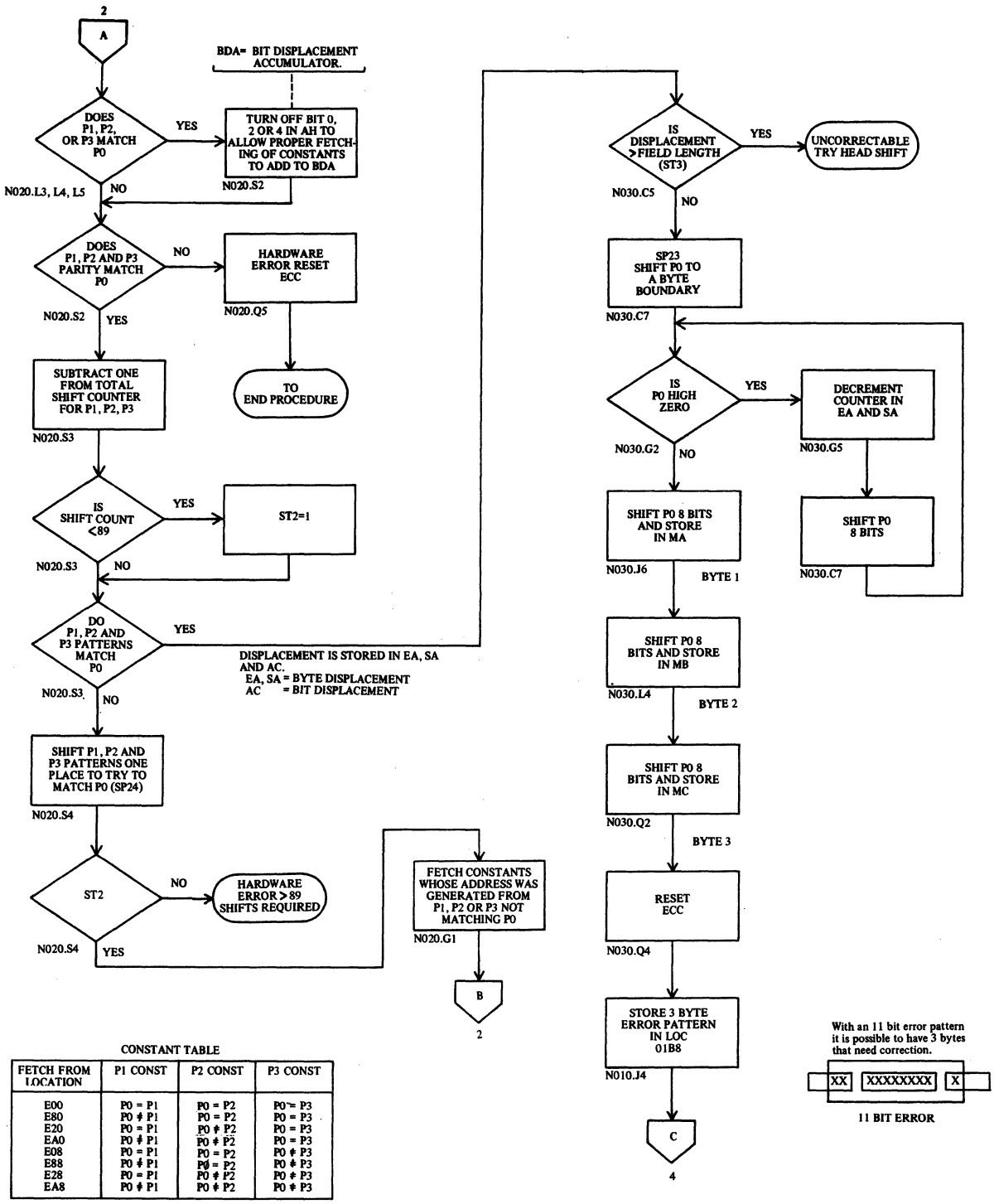


Figure 3-70. Command Retry for a Correctable Data Check in the Count Field (Sheet 3 of 18)

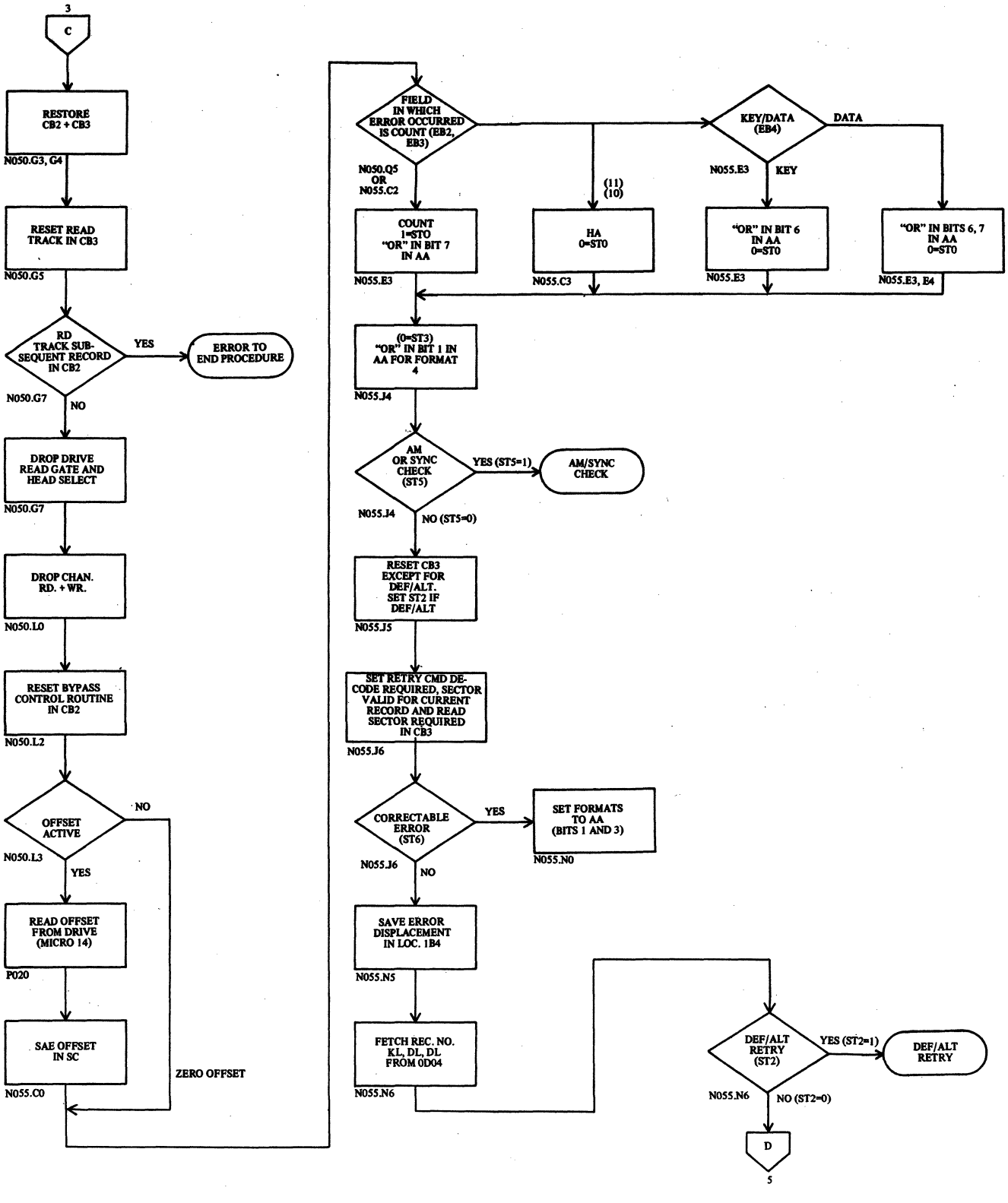


Figure 3-70. Command Retry for a Correctable Data Check in the Count Field (Sheet 4 of 18)

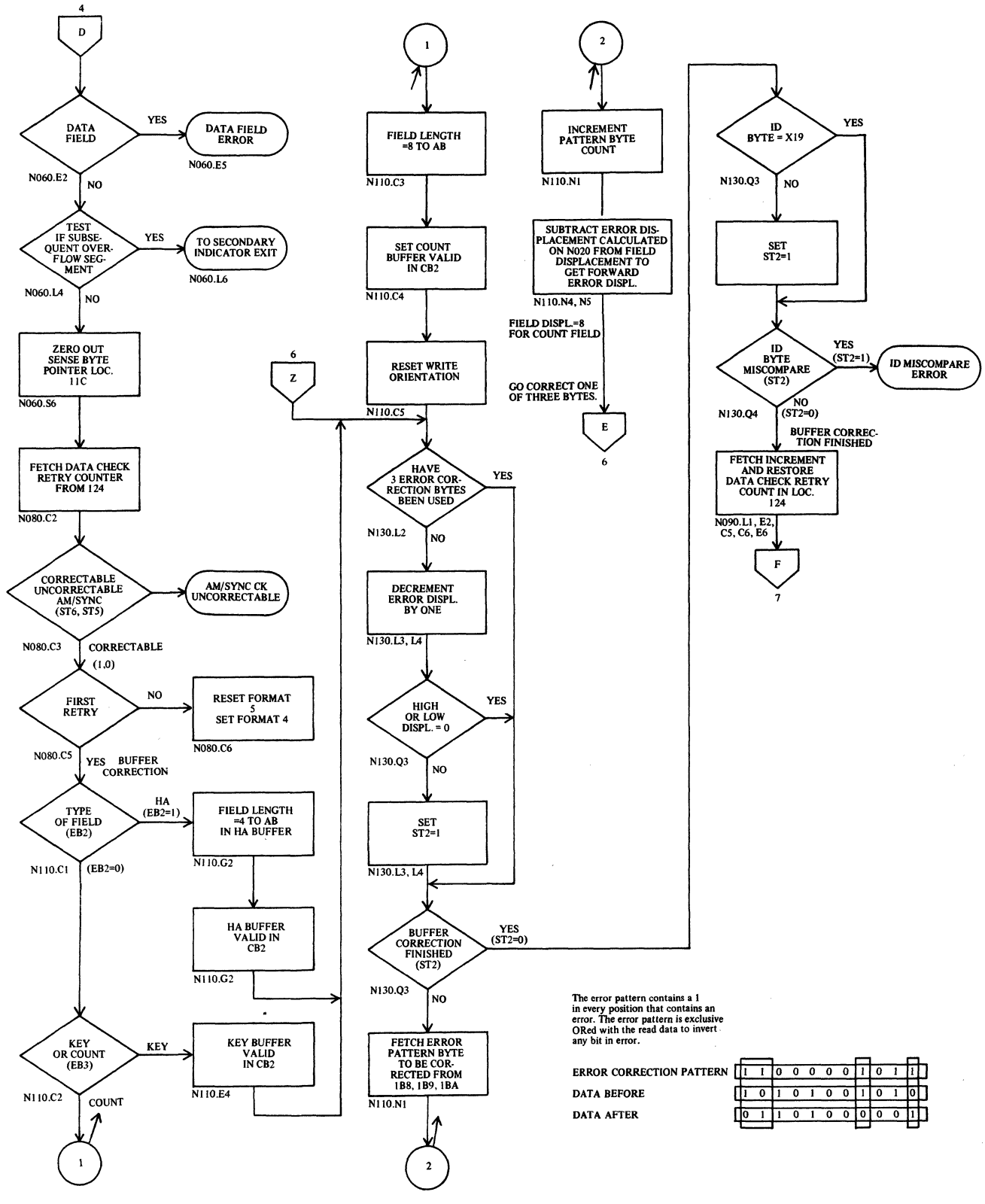


Figure 3-70. Command Retry for a Correctable Data Check in the Count Field (Sheet 5 of 18)

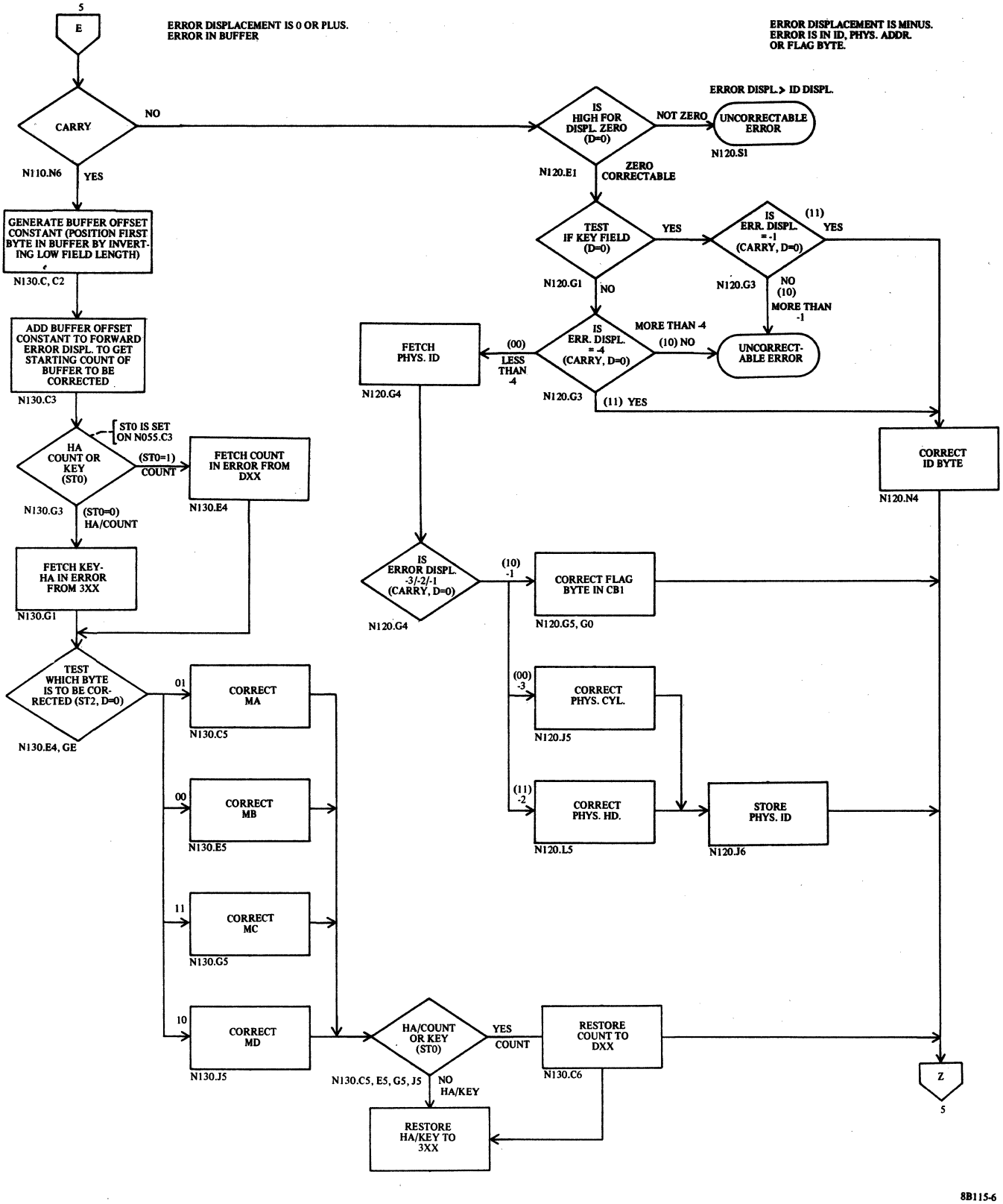


Figure 3-70. Command Retry for a Correctable Data Check in the Count Field (Sheet 6 of 18)

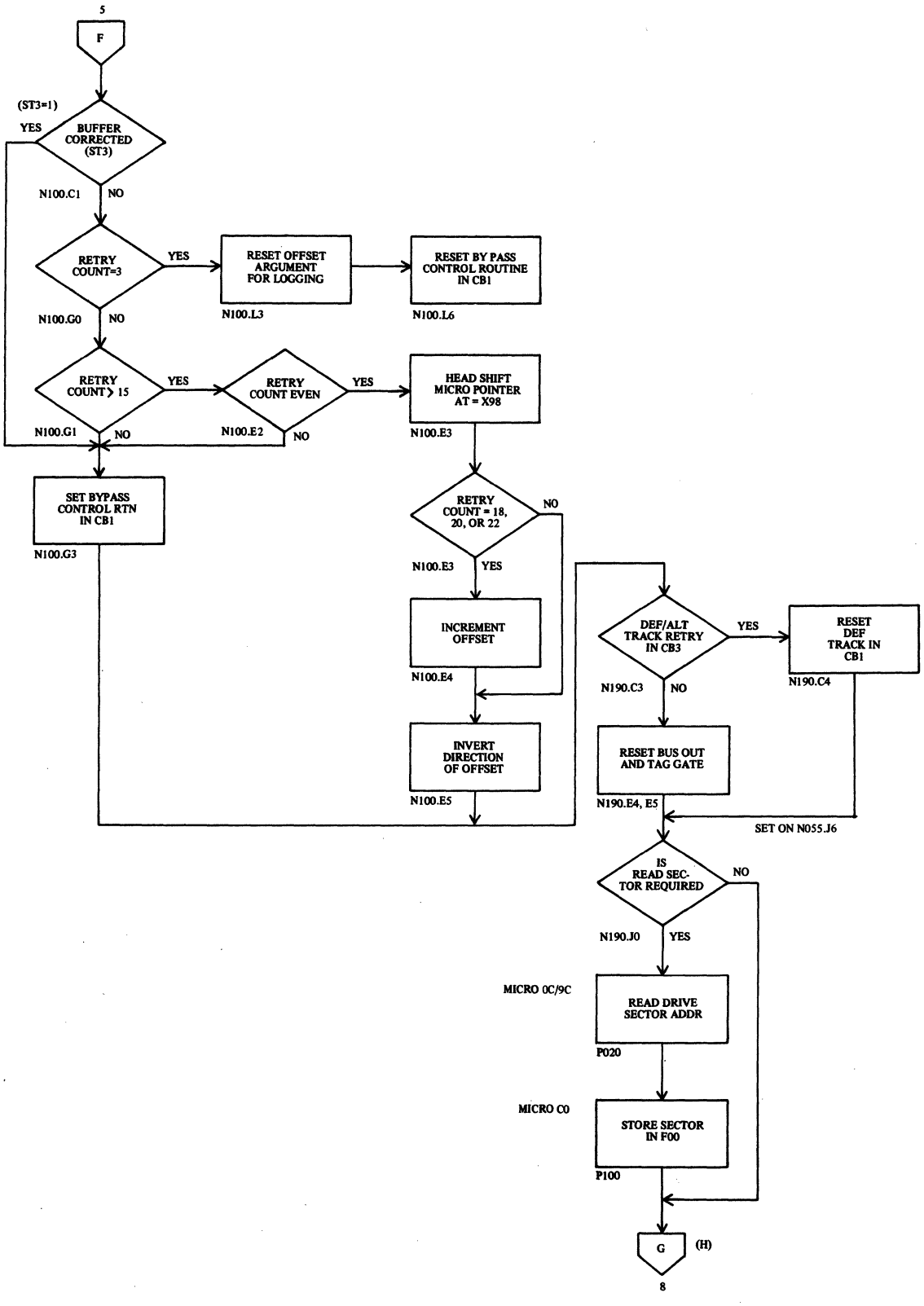


Figure 3-70. Command Retry for a Correctable Data Check in the Count Field (Sheet 7 of 18)

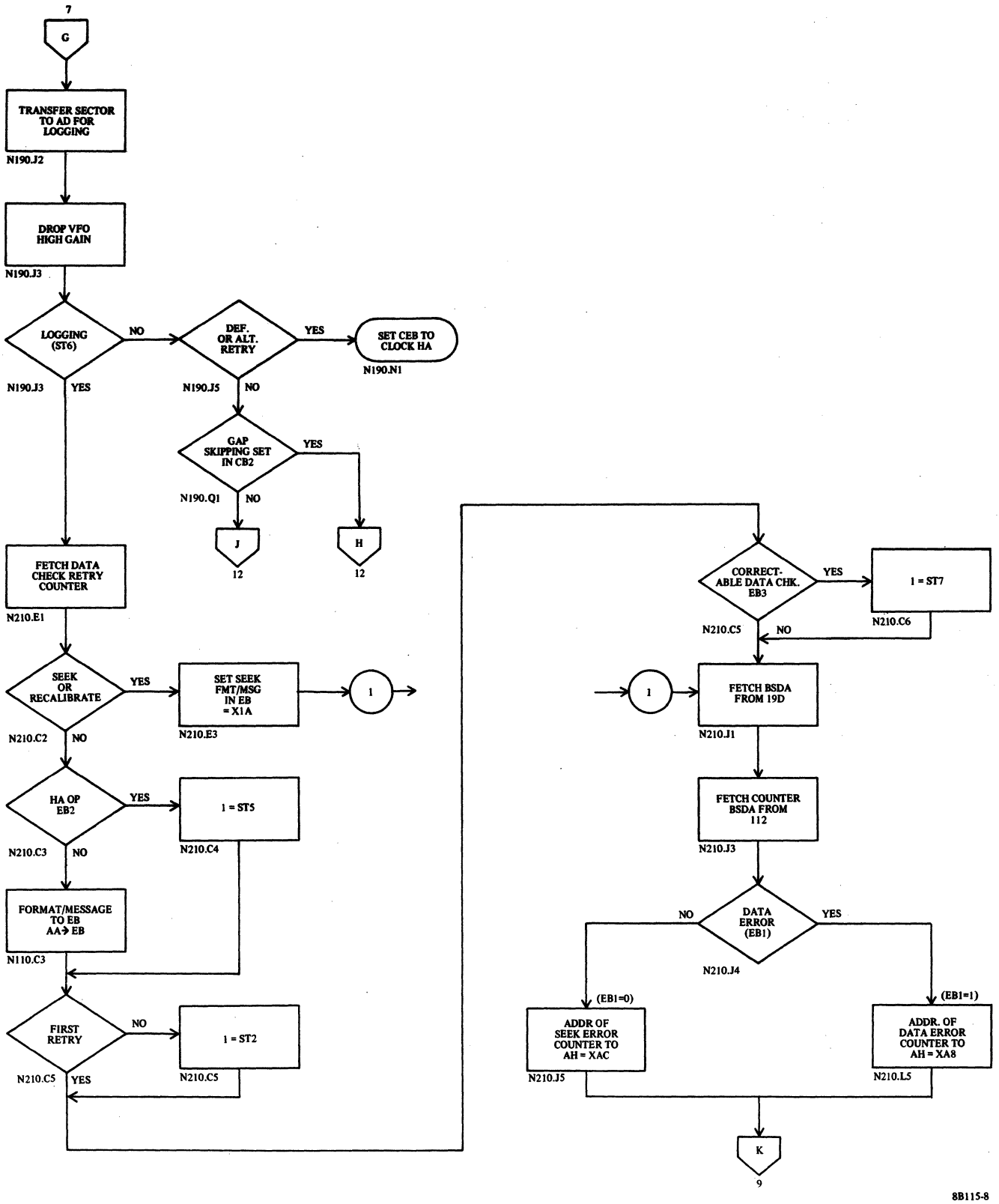


Figure 3-70. Command Retry for a Correctable Data Check in the Count Field (Sheet 8 of 18)

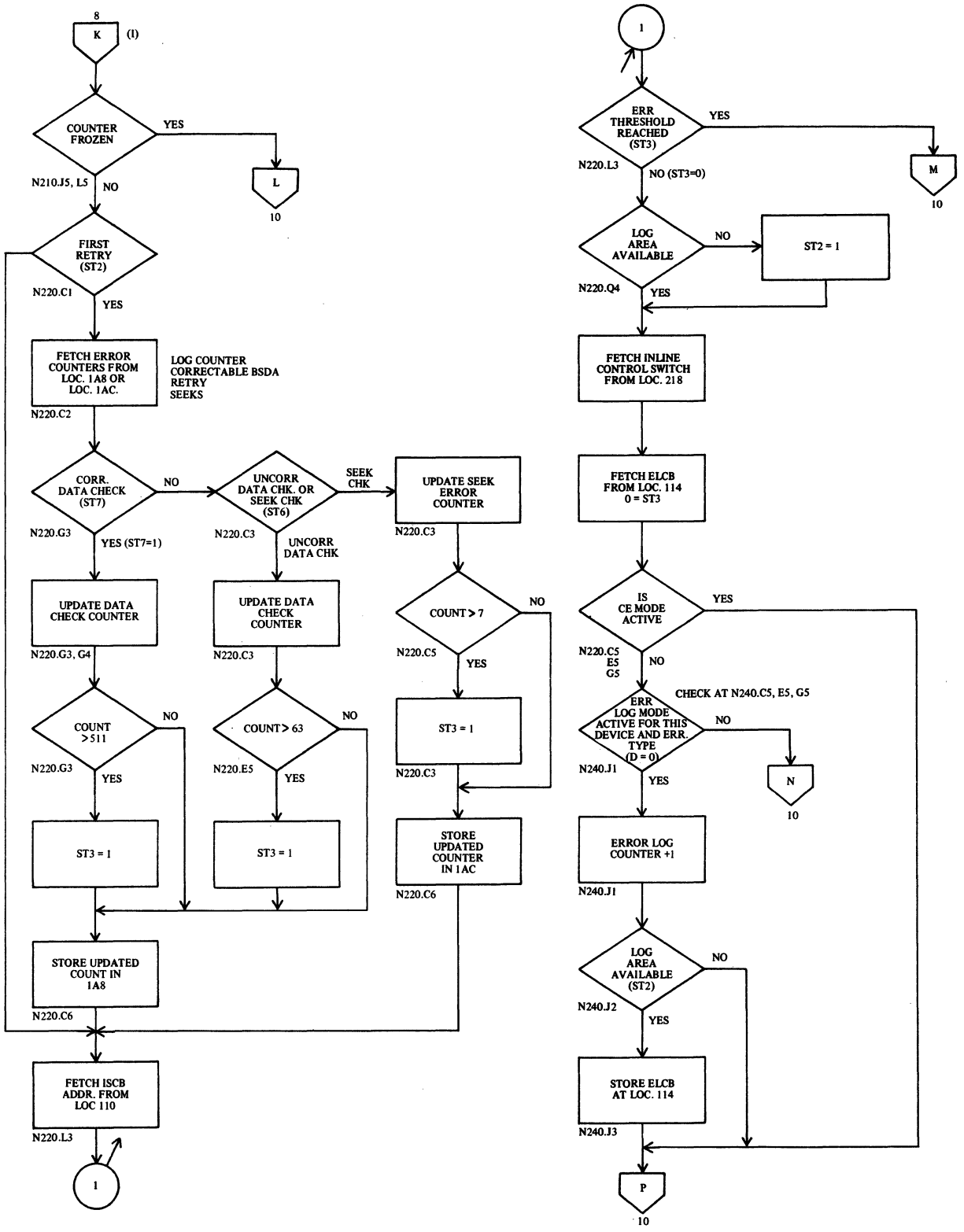


Figure 3-70. Command Retry for a Correctable Data Check in the Count Field (Sheet 9 of 18)

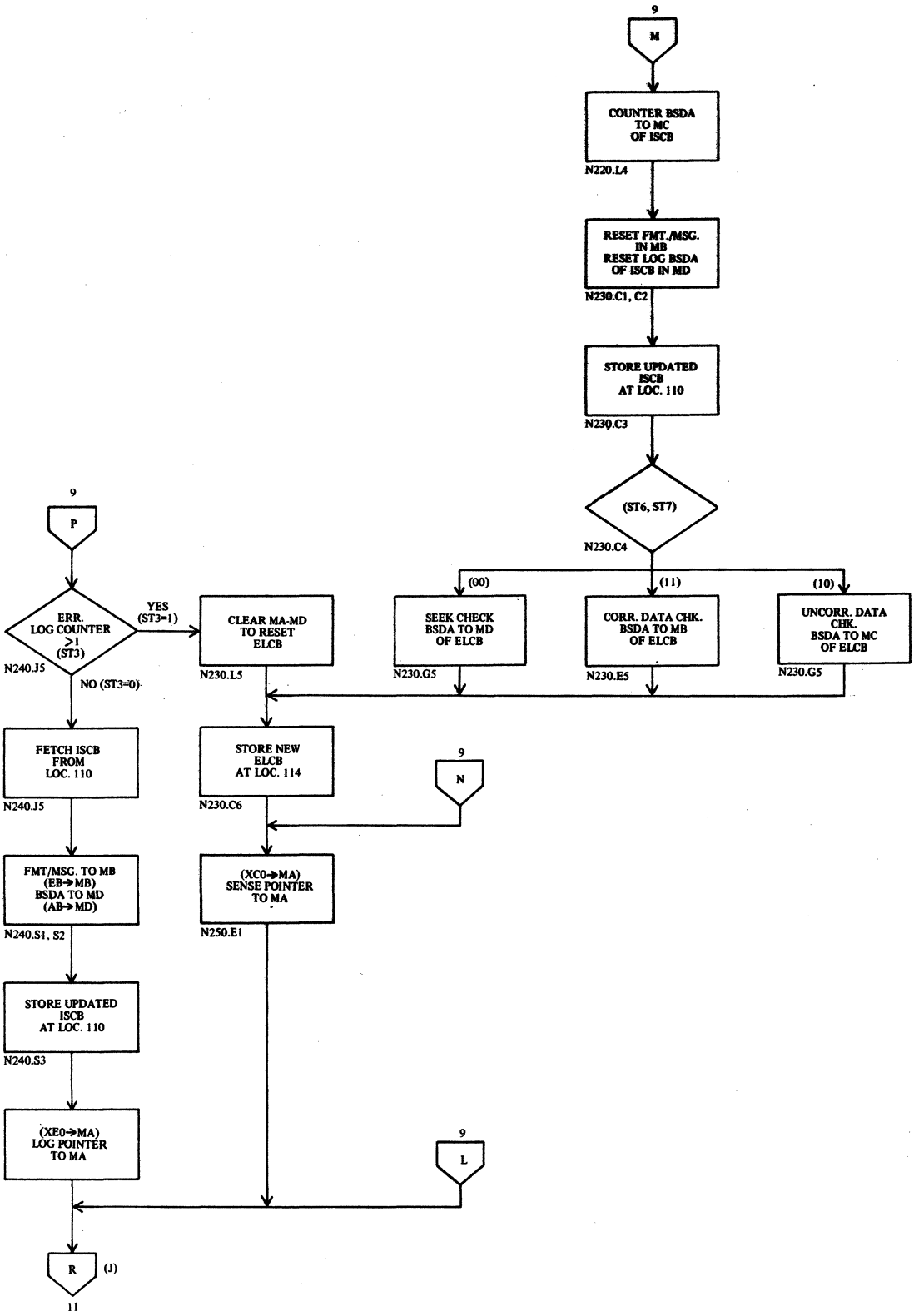


Figure 3-70. Command Retry for a Correctable Data Check in the Count Field (Sheet 10 of 18)

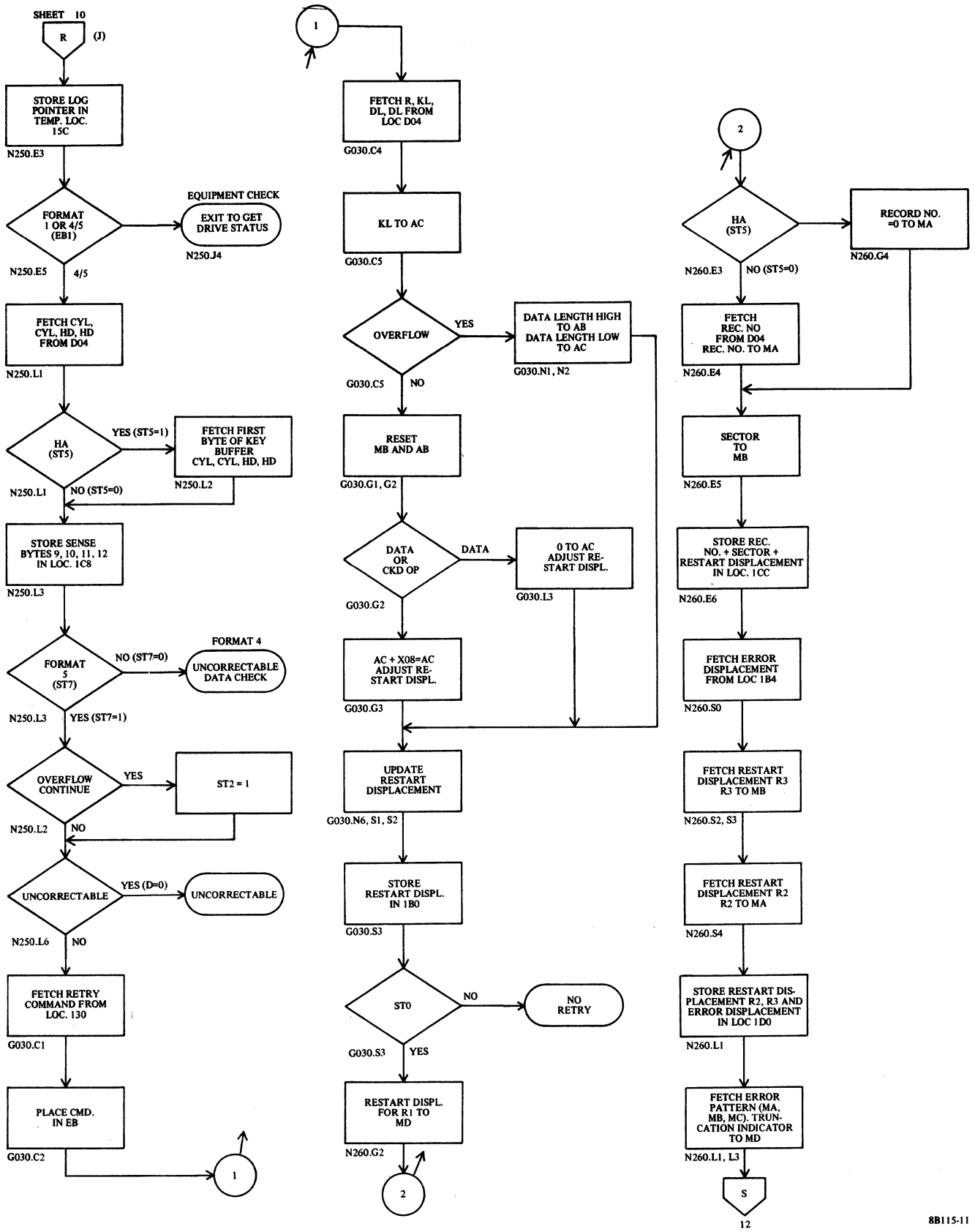


Figure 3-70. Command Retry for a Correctable Data Check in the Count Field (Sheet 11 of 18)

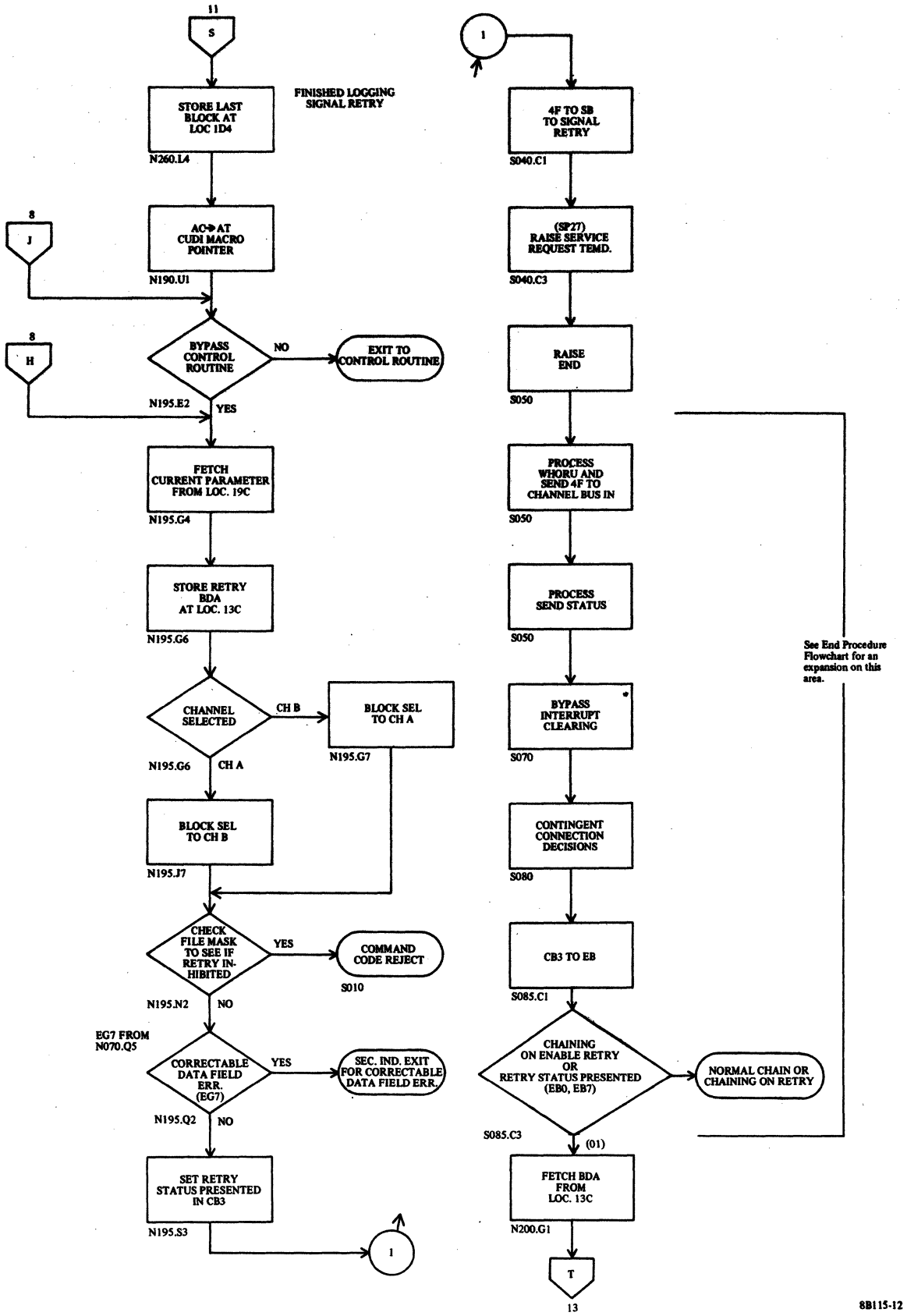


Figure 3-70. Command Retry for a Correctable Data Check in the Count Field (Sheet 12 of 18)

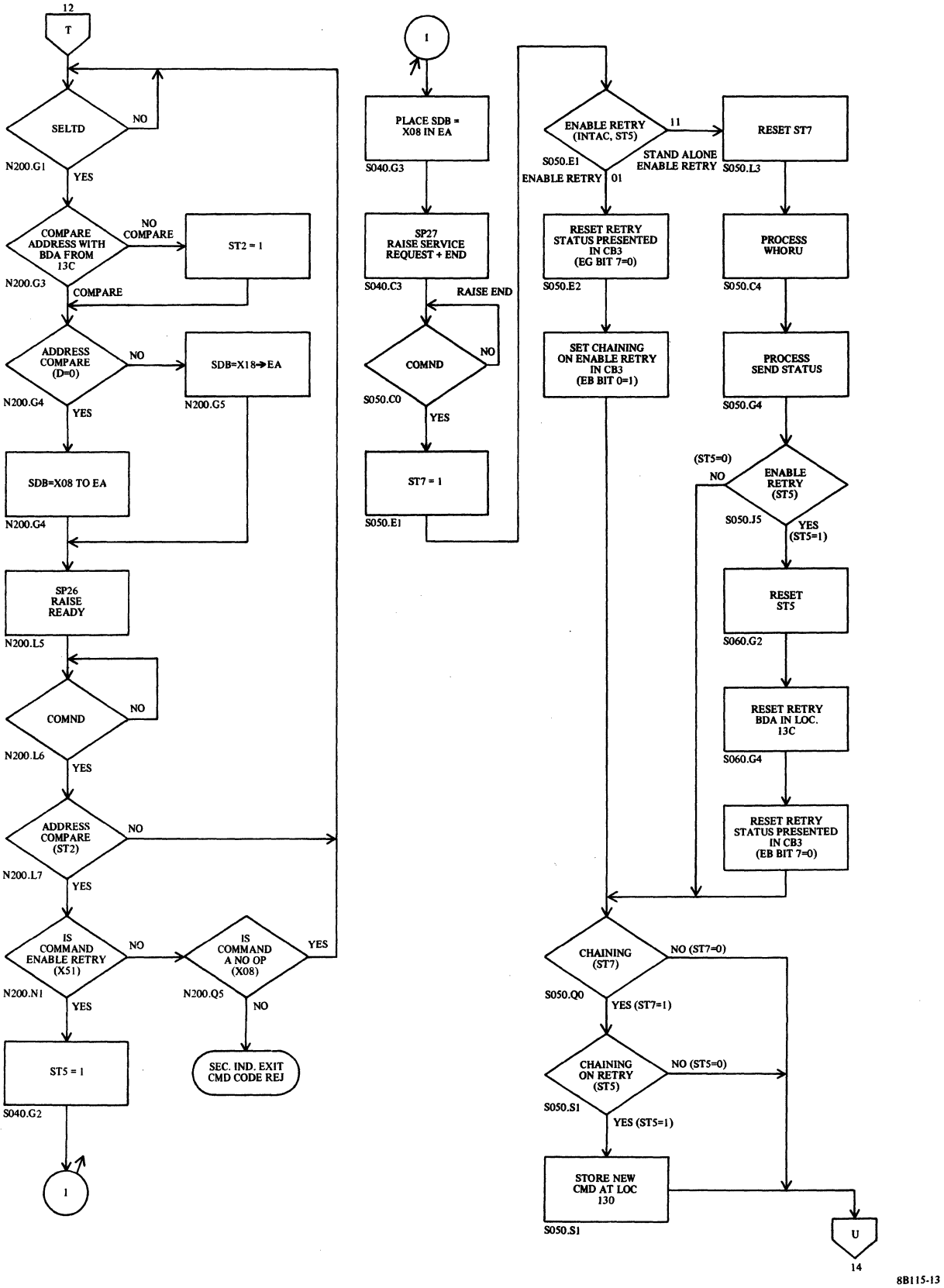


Figure 3-70. Command Retry for a Correctable Data Check in the Count Field (Sheet 13 of 18)

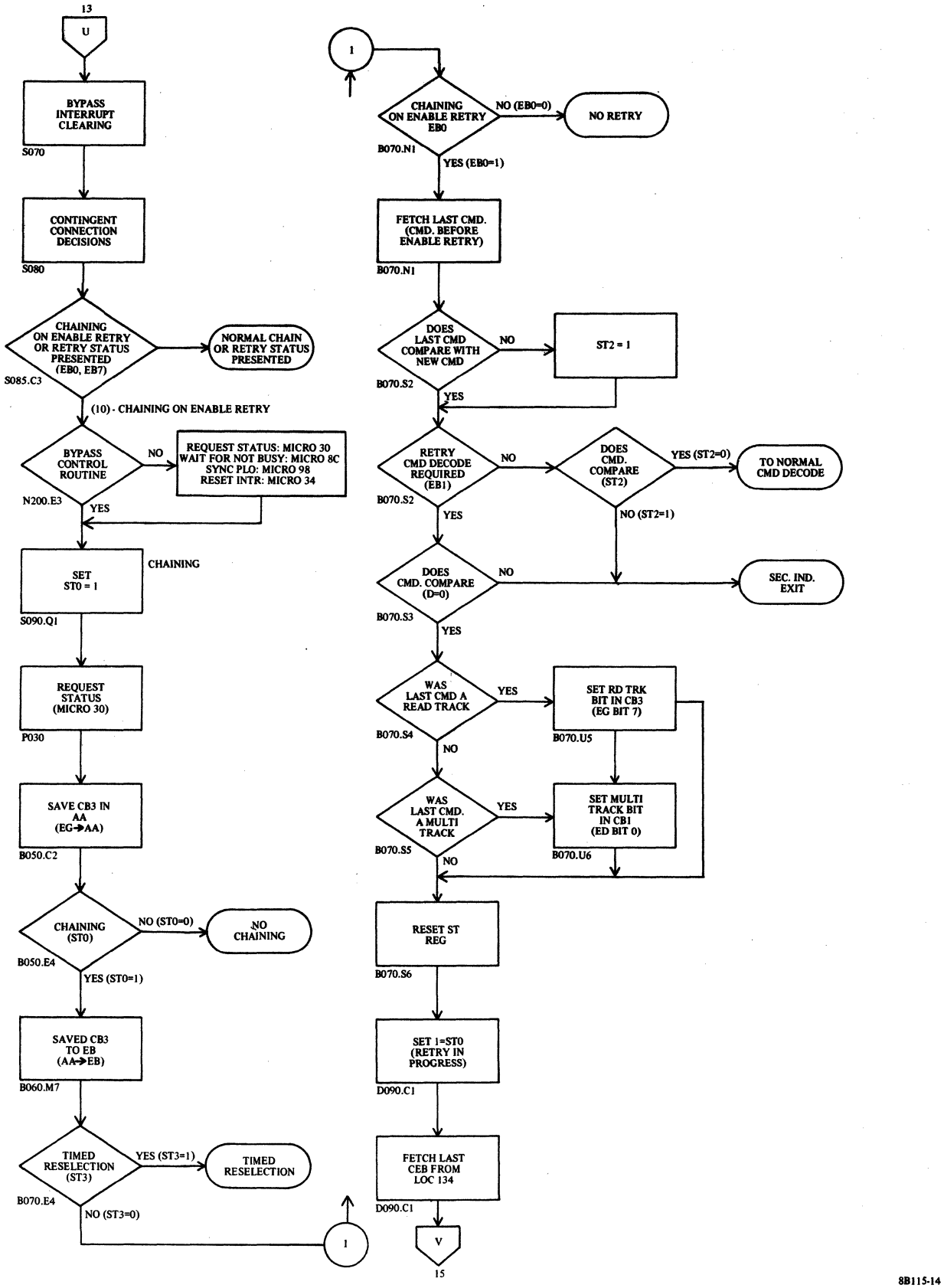
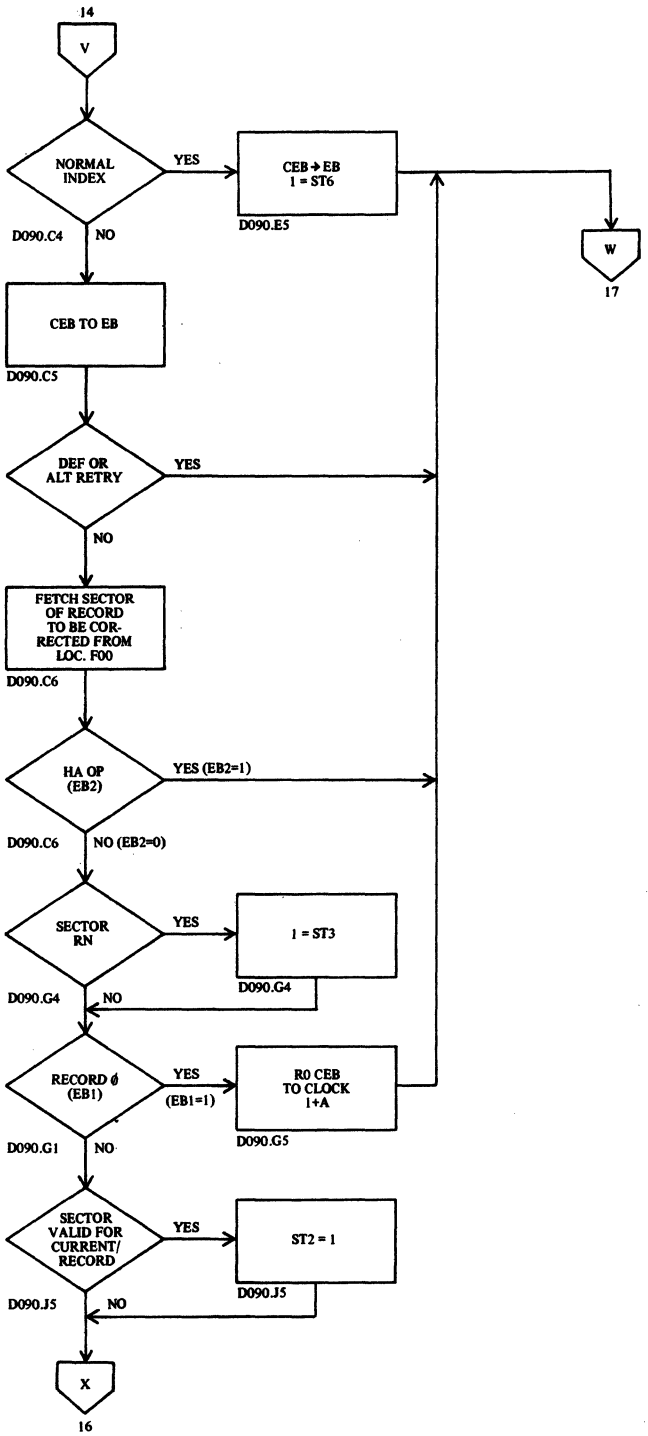


Figure 3-70. Command Retry for a Correctable Data Check in the Count Field (Sheet 14 of 18)



8B115-15

Figure 3-70. Command Retry for a Correctable Data Check in the Count Field (Sheet 15 of 18)

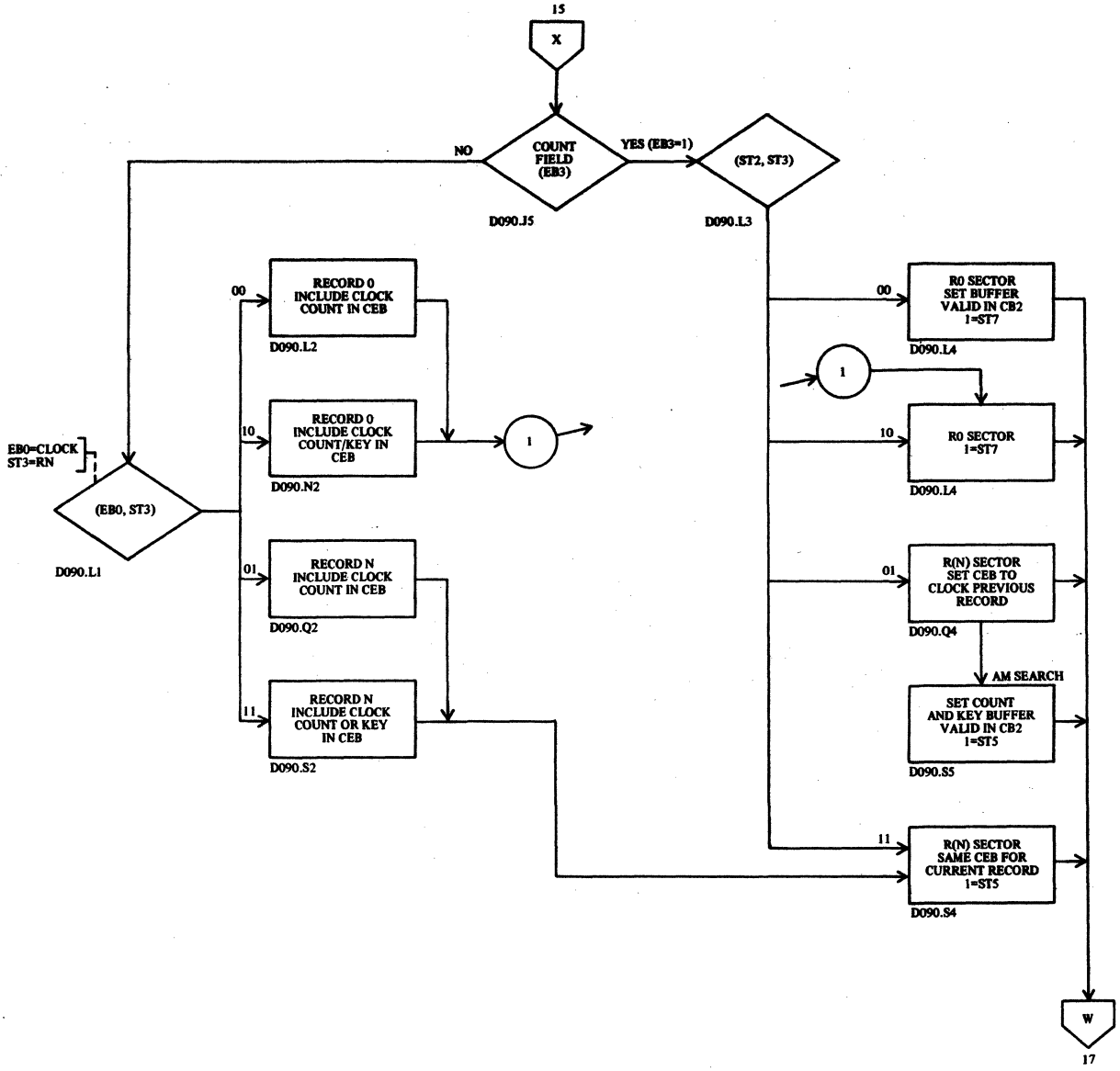


Figure 3-70. Command Retry for a Correctable Data Check in the Count Field (Sheet 16 of 18)

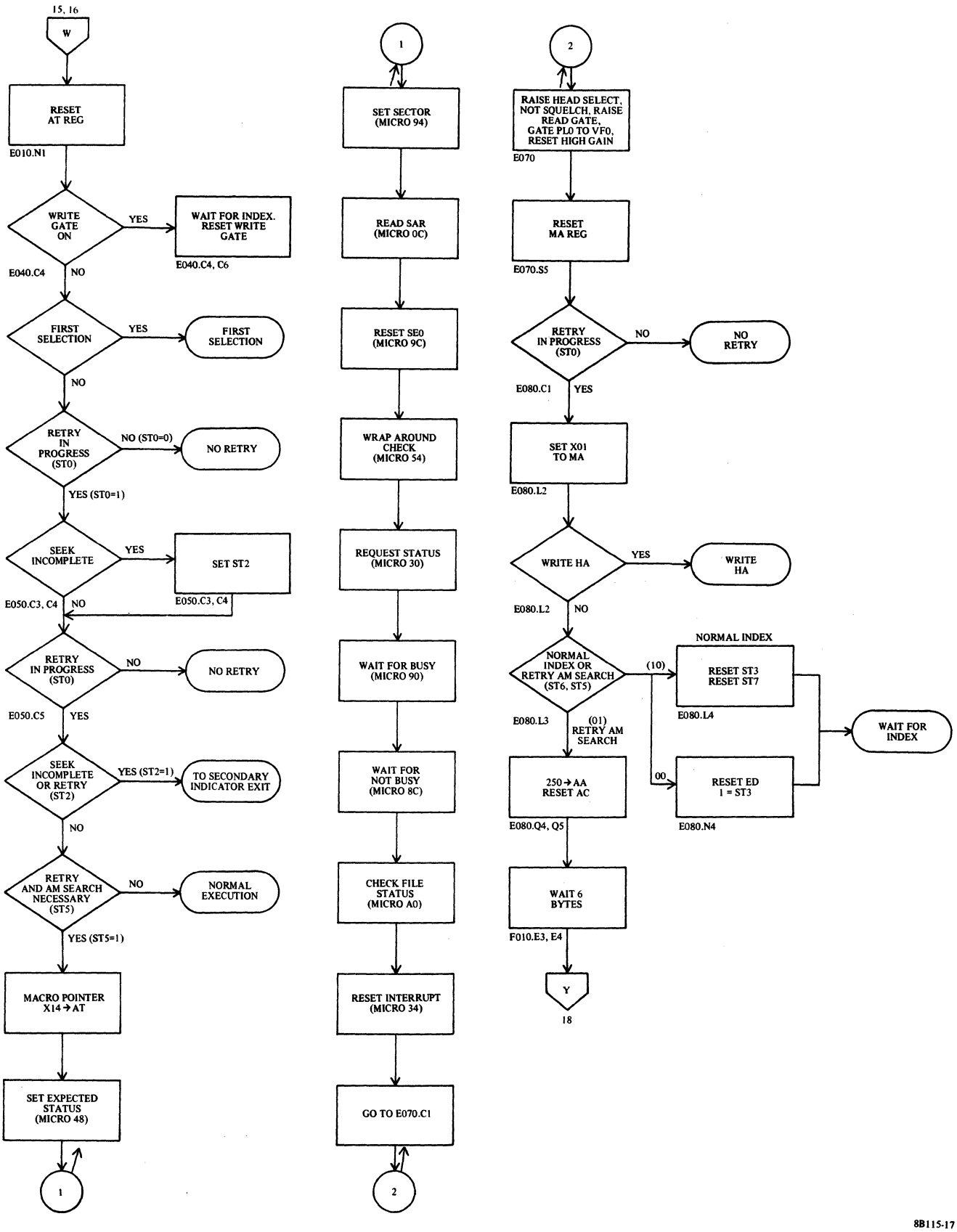


Figure 3-70. Command Retry for a Correctable Data Check in the Count Field (Sheet 17 of 18)

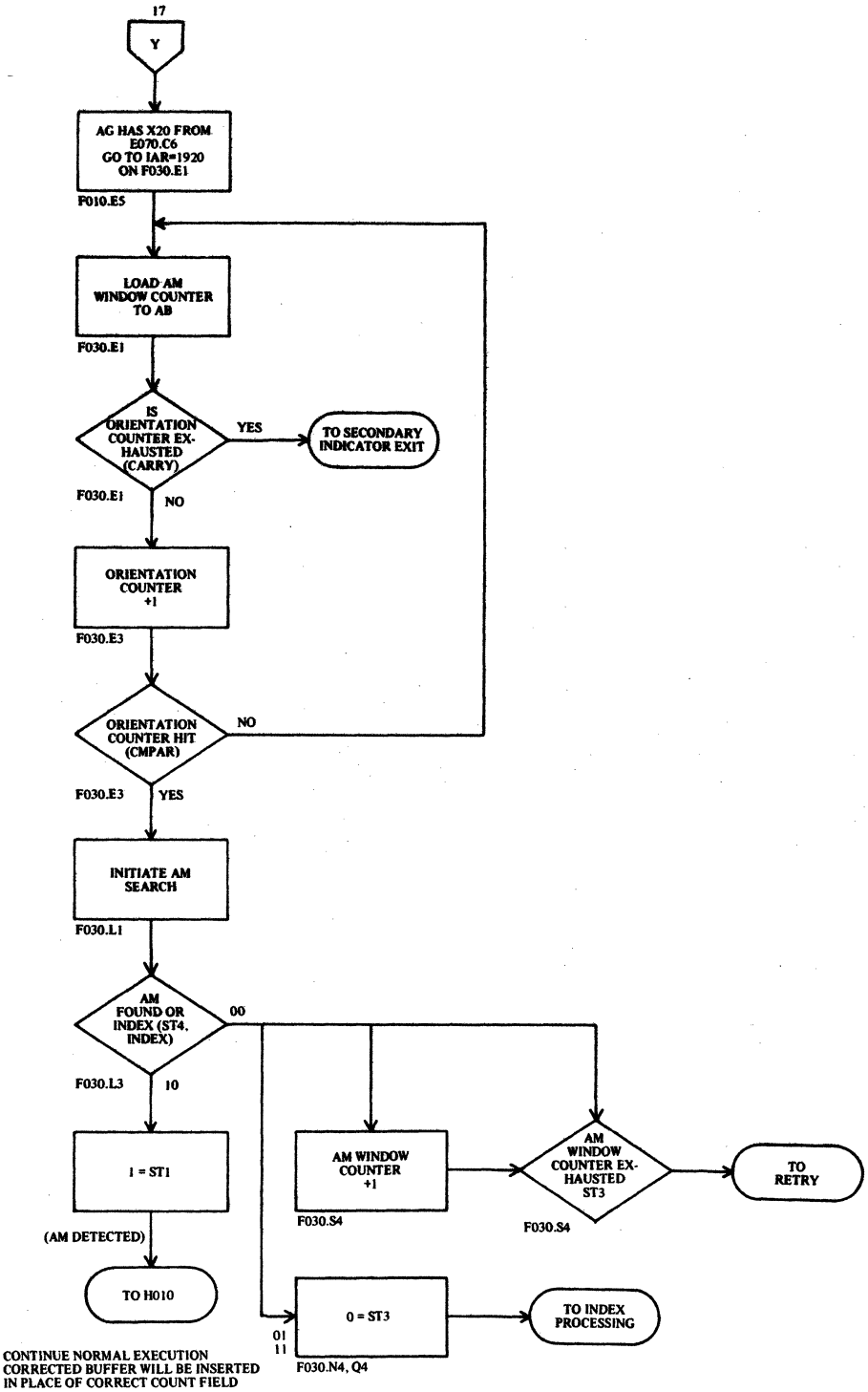


Figure 3-70. Command Retry for a Correctable Data Check in the Count Field (Sheet 18 of 18)

END PROCEDURE

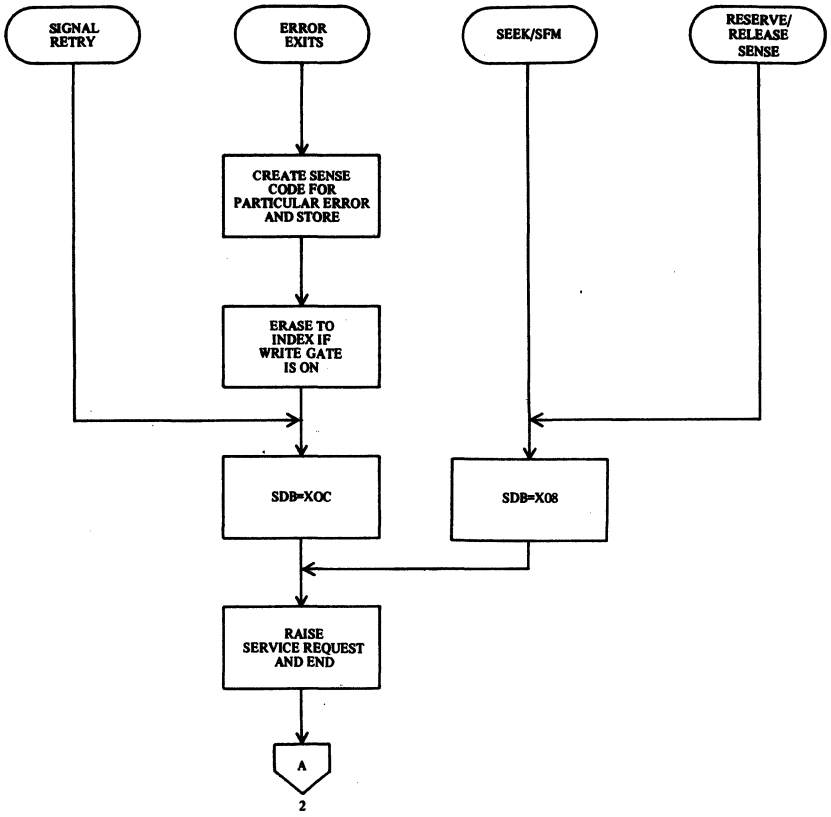
End Procedure is the steering mechanism for ending commands and errors.

It signals the channel with the correct Standard Device Byte so that the correct next channel operation may follow.

All of the following are also under the direction of End Procedure:

1. Unit Check status generation
2. Test if Retry is in progress
3. Channel quiescence
4. Monitor index
5. Process end status - including Retry status
6. Pull for interrupts if chaining and seek hold off
7. Reset interrupts if requested
8. Process channel Devise End flags (SIP, PCH)
9. Restore current parameters
10. Generate channel interrupt mark for wait loop
11. Allow channel disable
12. Set channel block and/or inline inhibit timers
13. Deselect drive

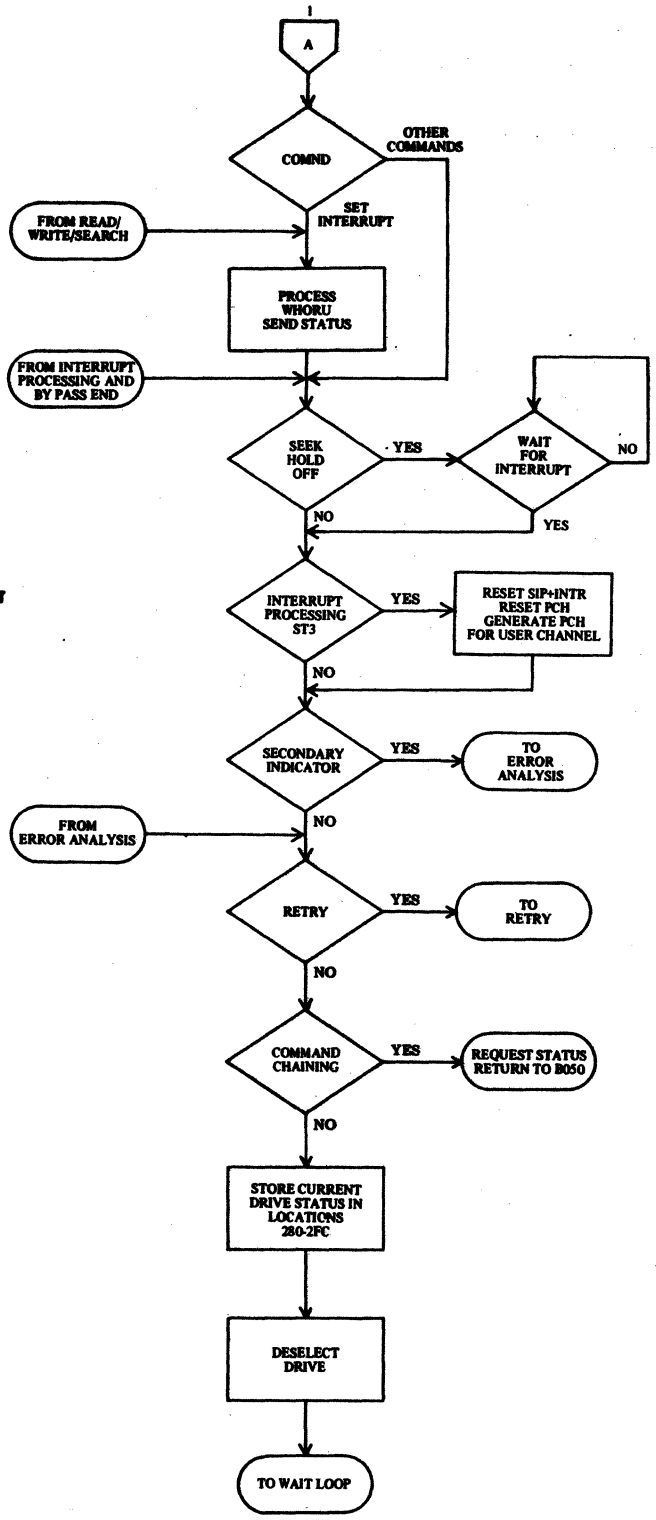
GENERAL FLOWCHART



8B117-1

Figure 3-71. End Procedure (Sheet 1 of 10)

GENERAL FLOWCHART
(continued)



8B117-2

Figure 3-71. End Procedure (Sheet 2 of 10)

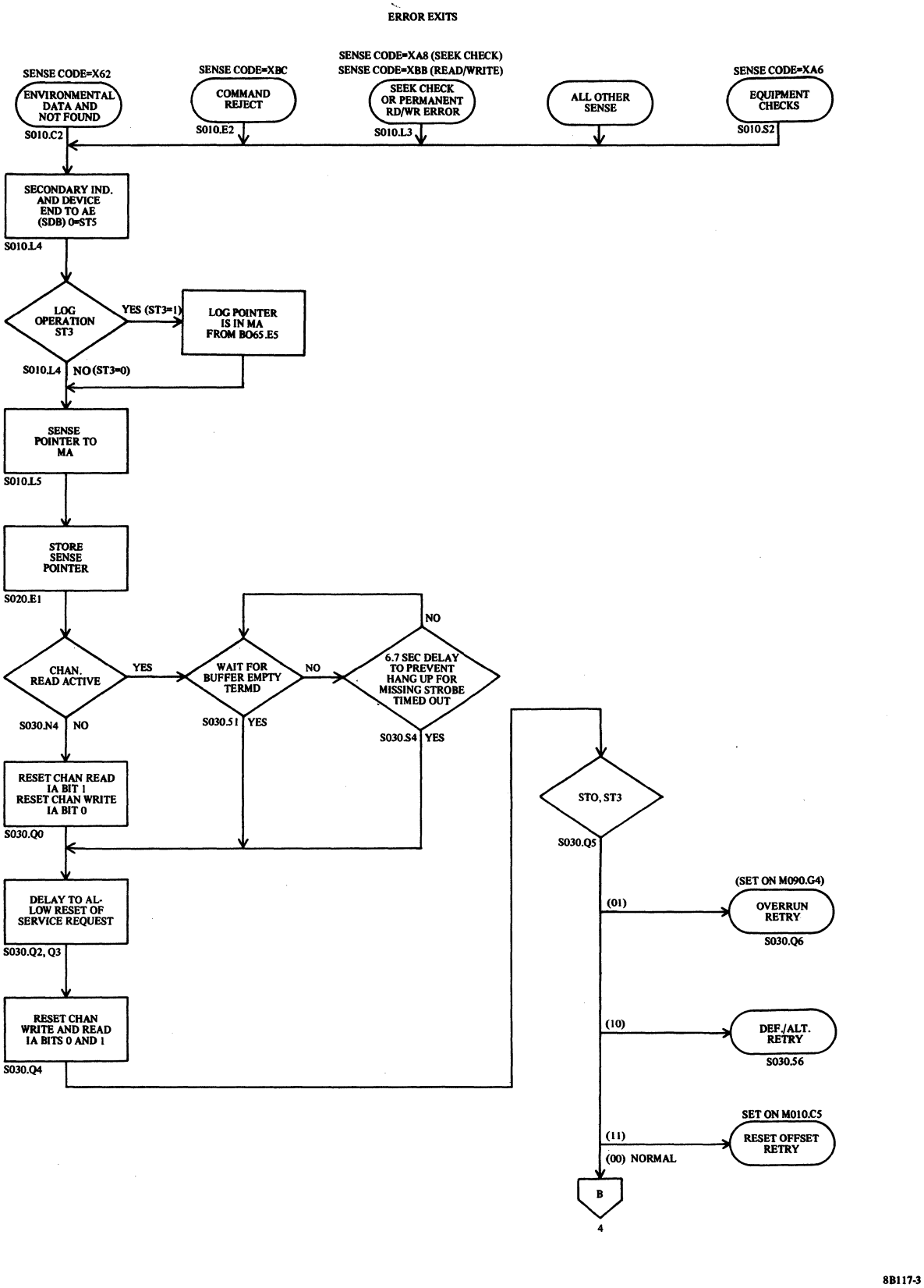


Figure 3-71. End Procedure (Sheet 3 of 10)

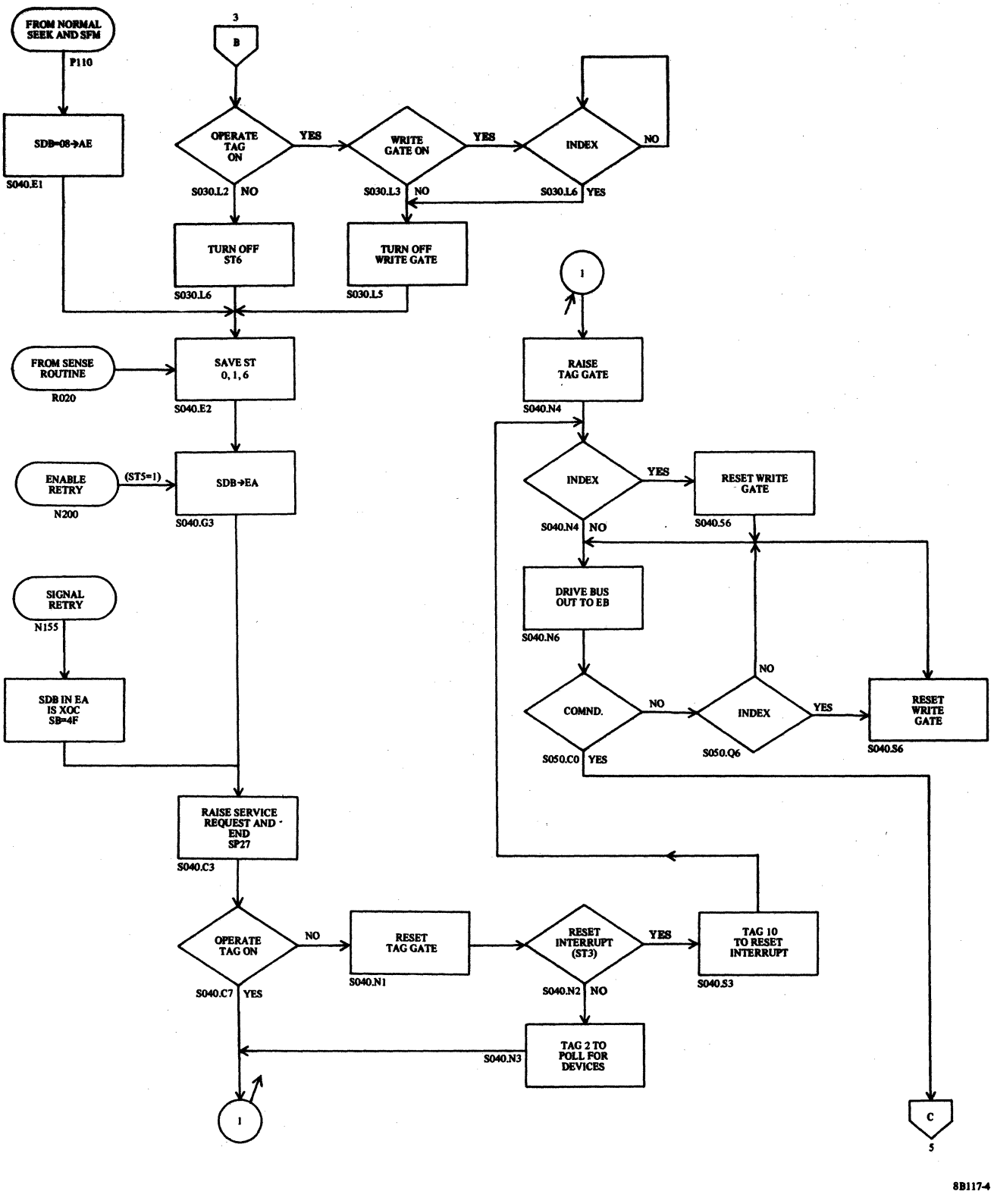


Figure 3-71. End Procedure (Sheet 4 of 10)

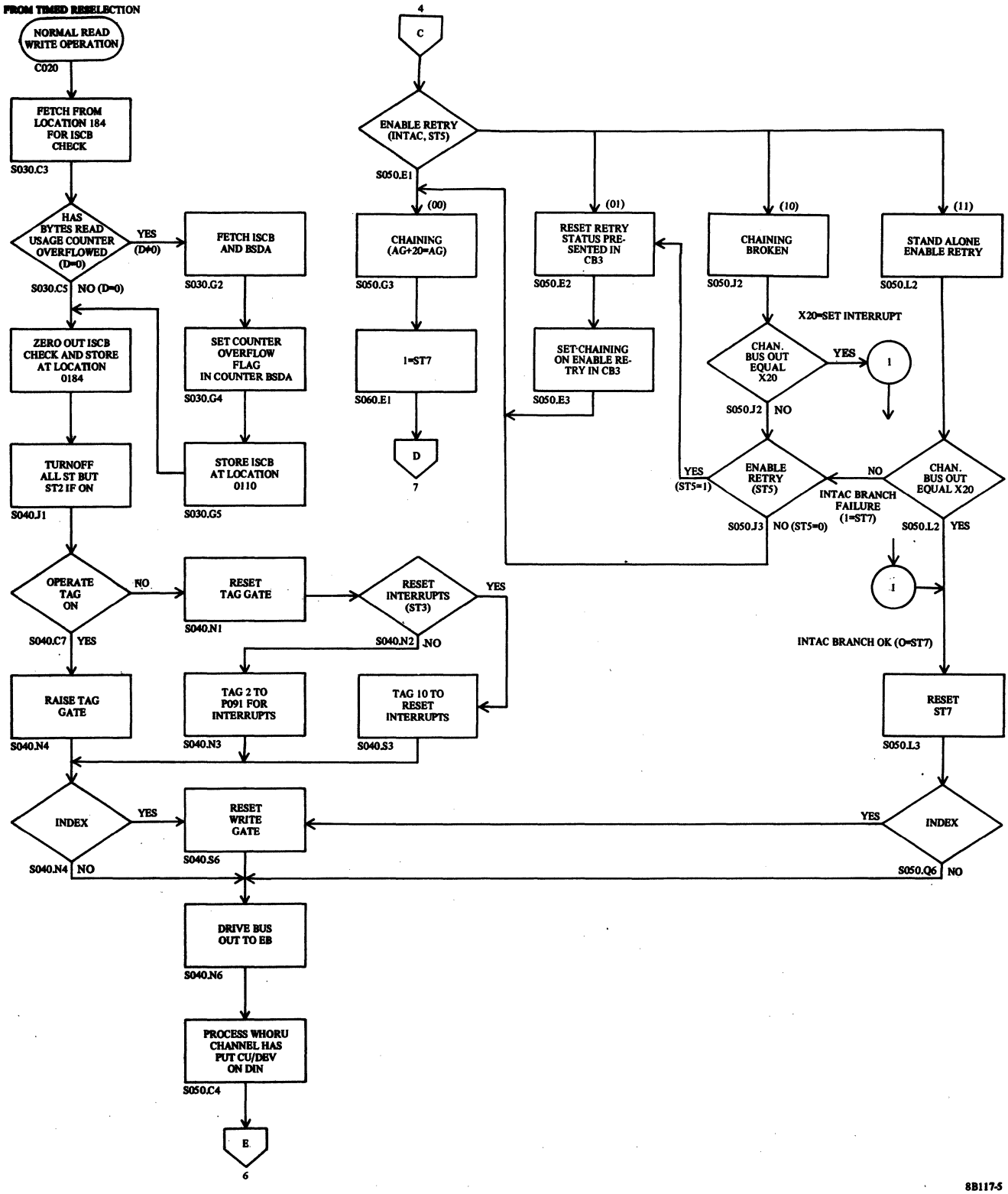


Figure 3-71. End Procedure (Sheet 5 of 10)

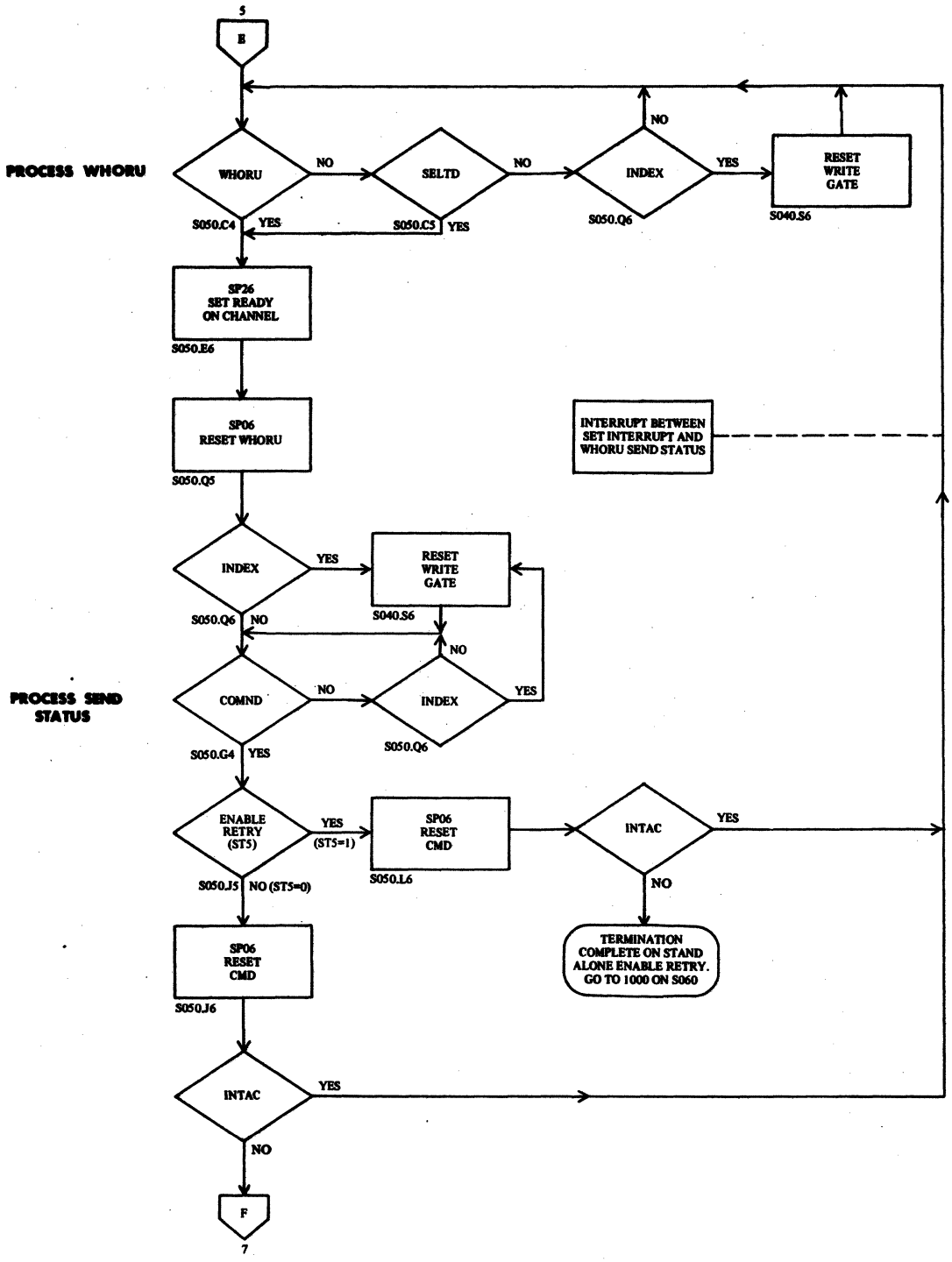


Figure 3-71. End Procedure (Sheet 6 of 10)

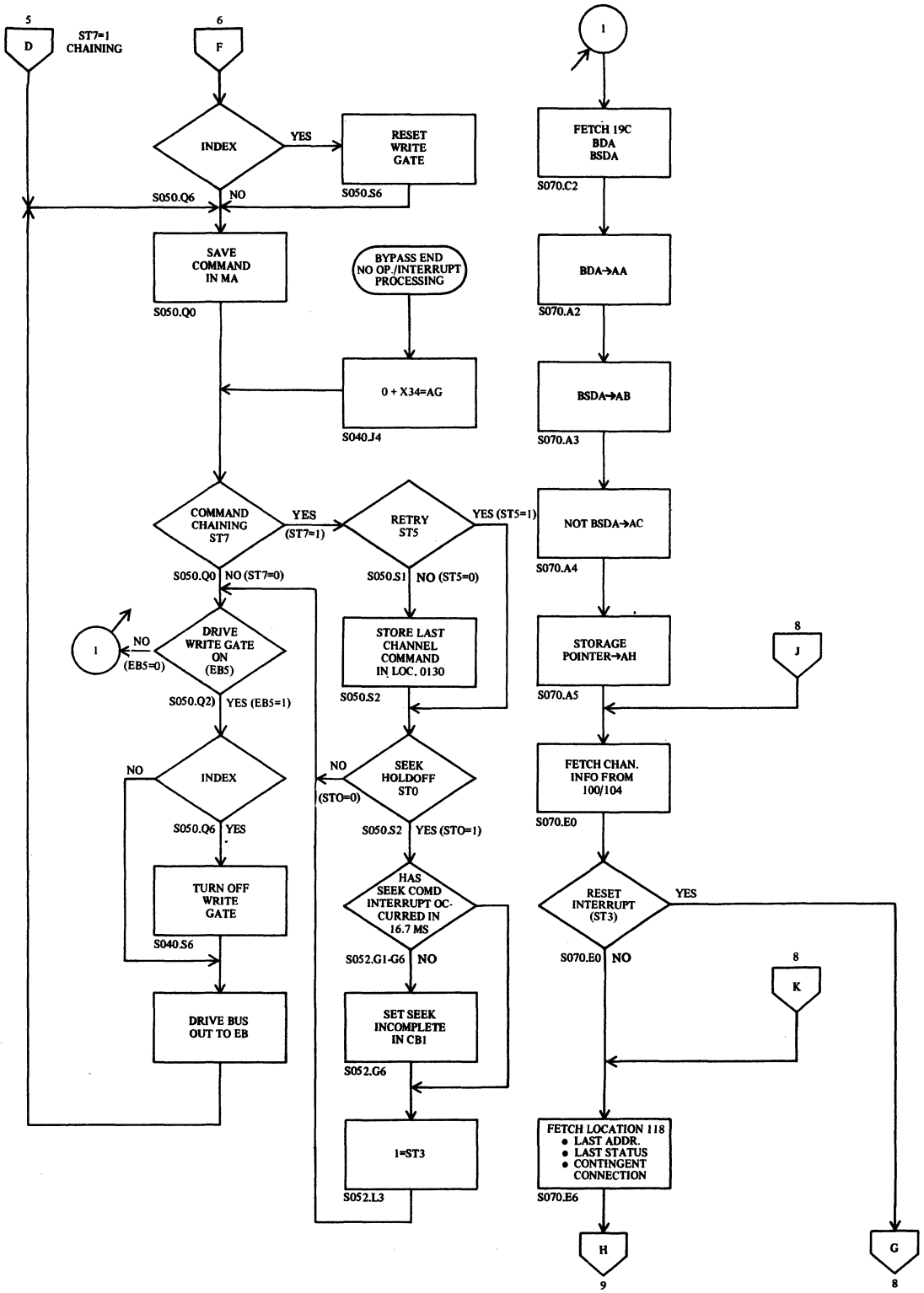


Figure 3-71. End Procedure (Sheet 7 of 10)

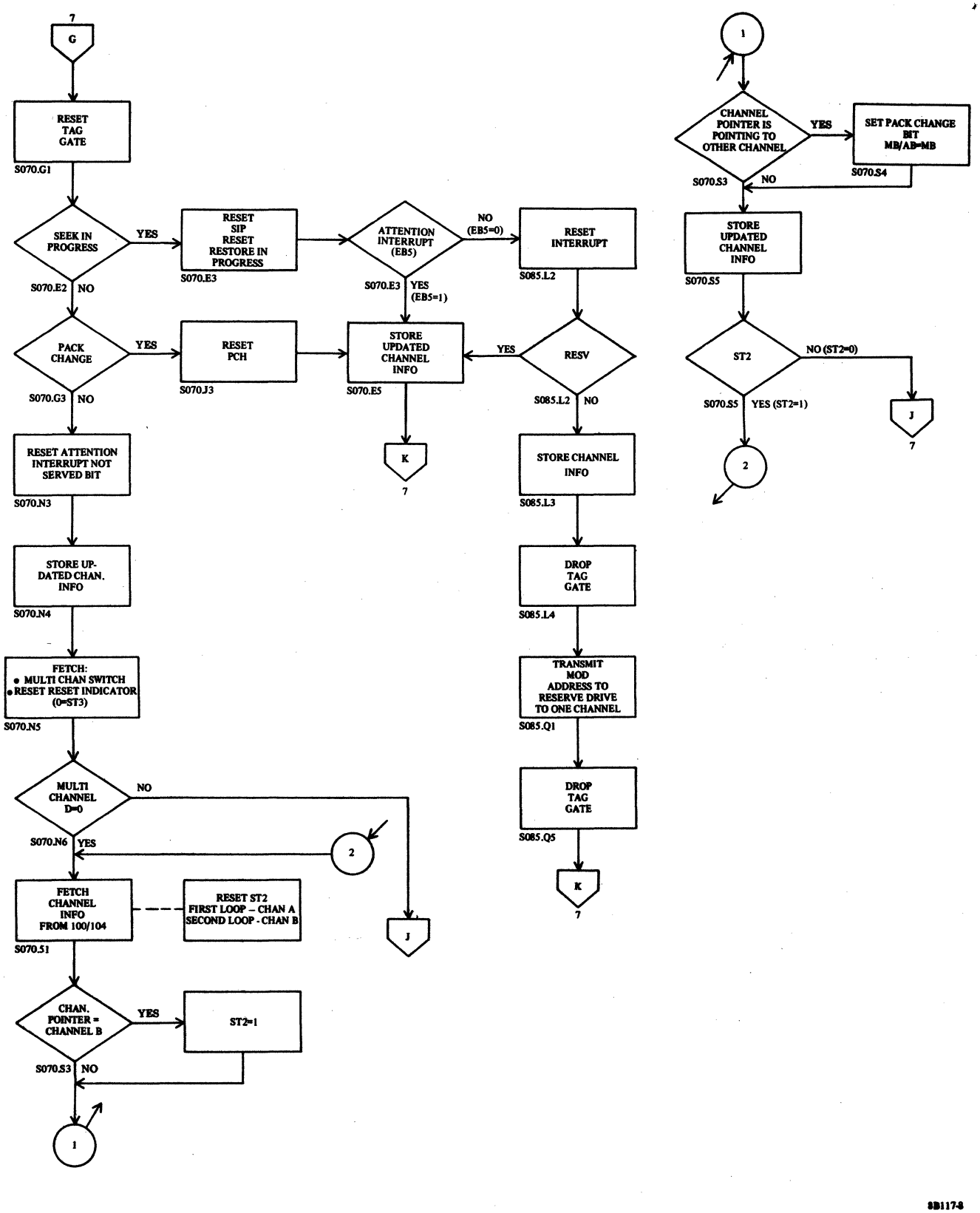


Figure 3-71. End Procedure (Sheet 8 of 10)

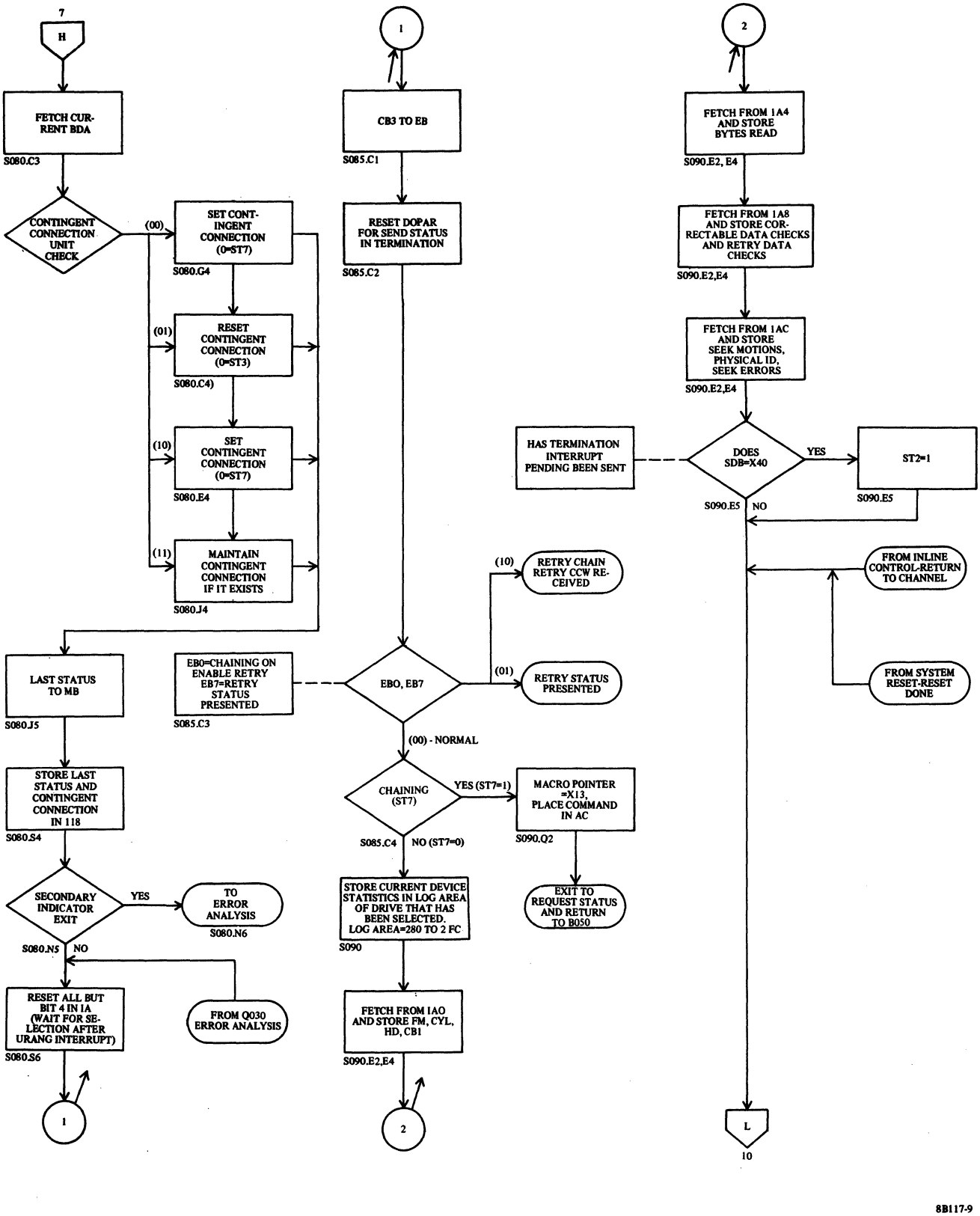


Figure 3-71. End Procedure (Sheet 9 of 10)

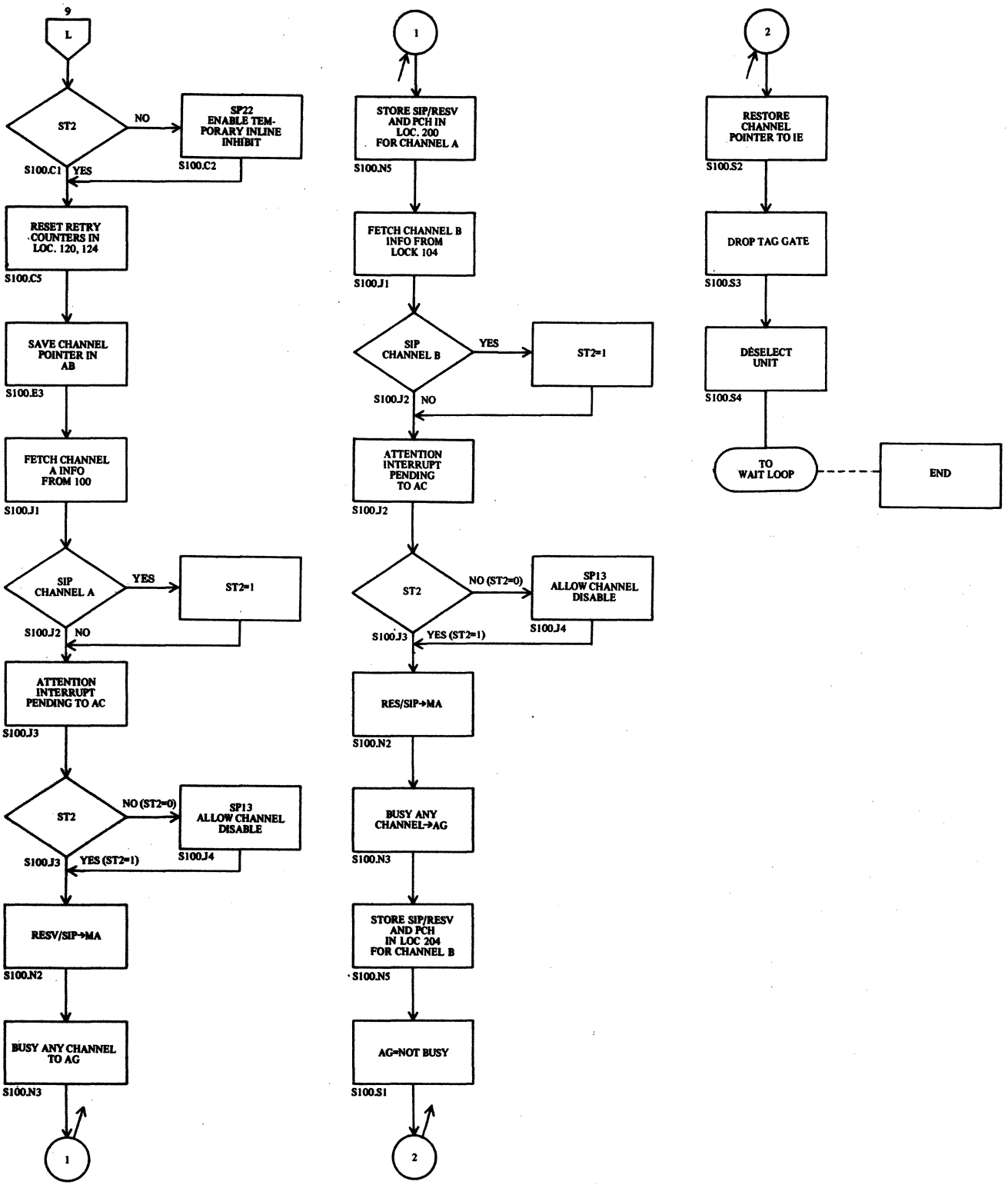
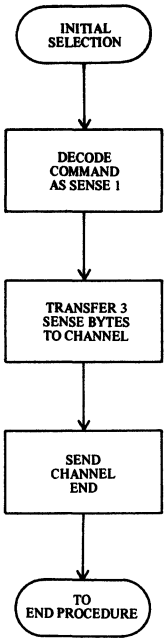


Figure 3-71. End Procedure (Sheet 10 of 10)

SENSE 1

The Sense 1 command is initiated in response to an indication from the controller that a secondary status condition has been detected (2^2 of the Standard Device Byte is true). The execution of this command results in the transfer of up to three Operational Status Bytes from the controller to channel.

GENERAL FLOWCHART



8B116-1

Figure 3-72. Sense 1 Operation (Sheet 1 of 4)

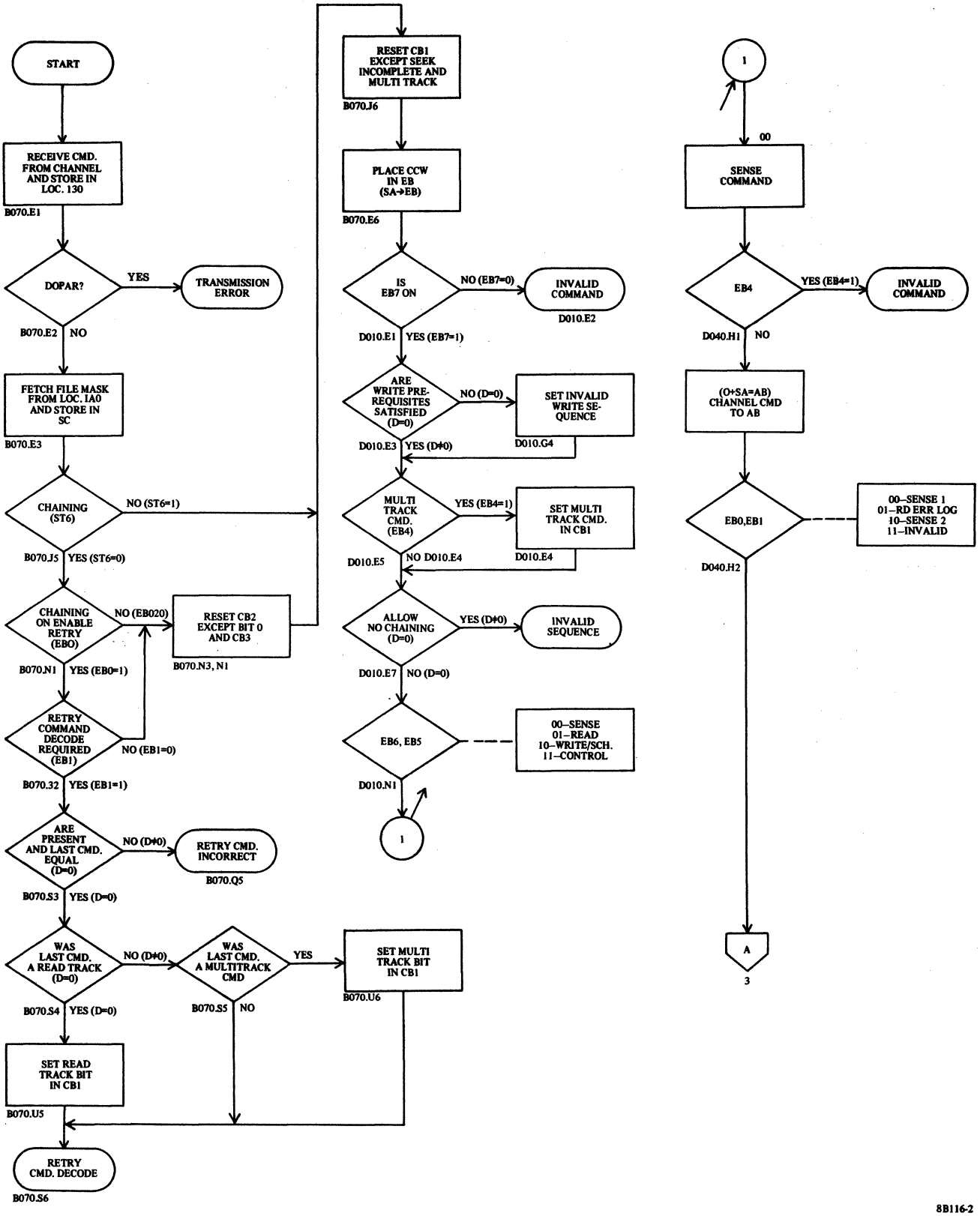


Figure 3-72. Sense 1 Operation (Sheet 2 of 4)

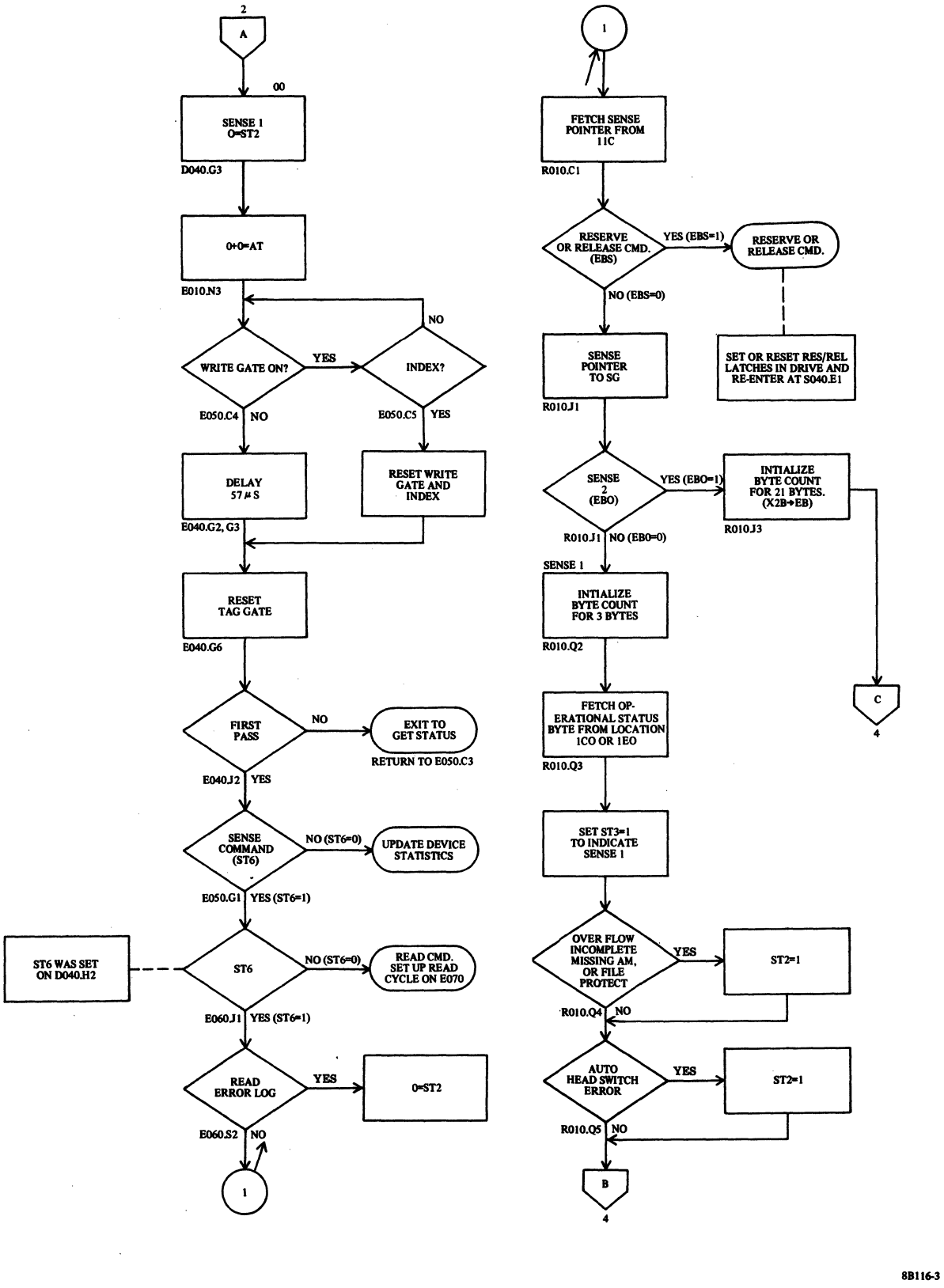


Figure 3-72. Sense 1 Operation (Sheet 3 of 4)

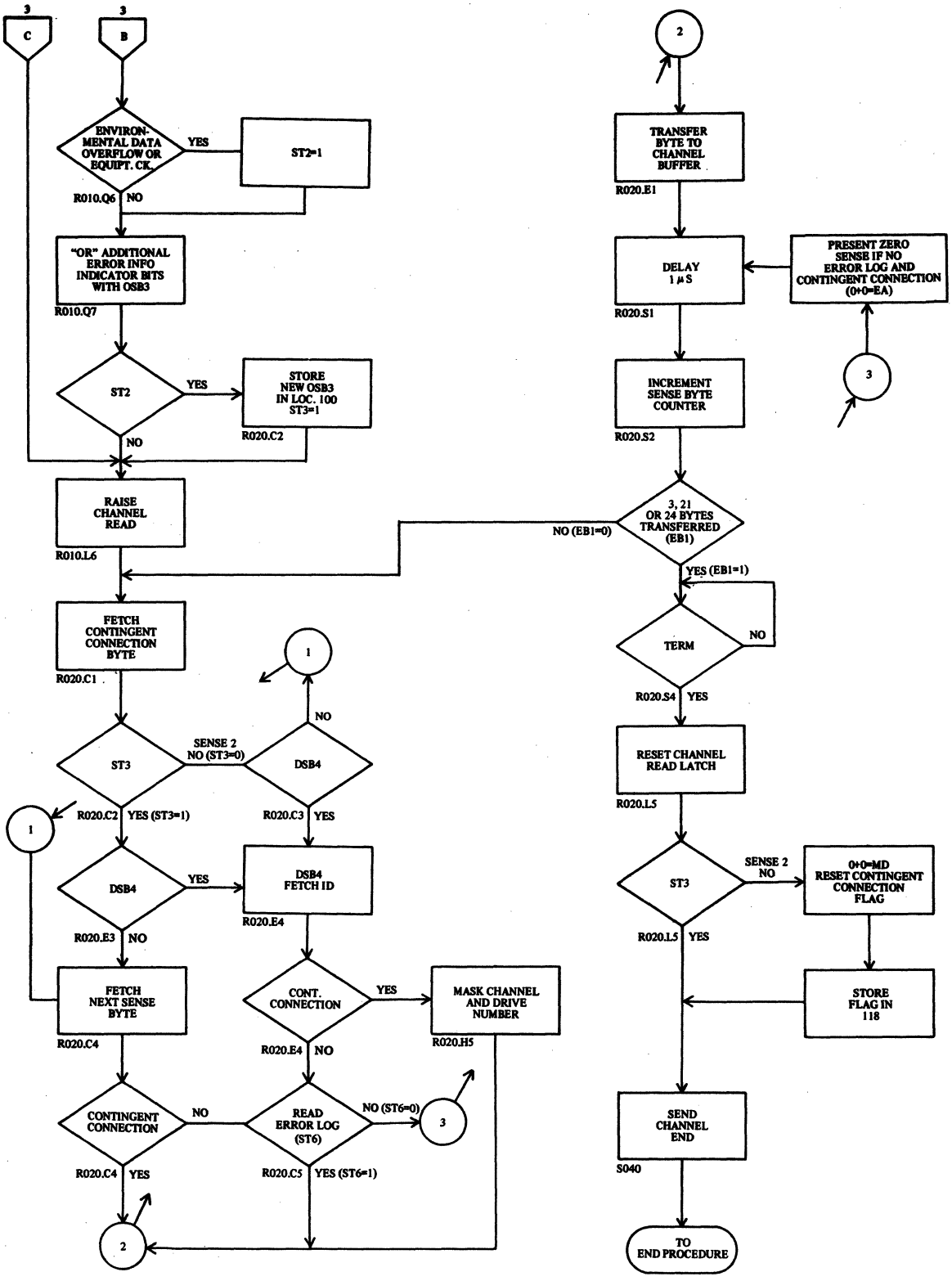


Figure 3-72. Sense 1 Operation (Sheet 4 of 4)

COMMENT SHEET

PLEASE COMPLETE ITEMS 1 THRU 11

From

(1) NAME
(2) DEPARTMENT OR ATTENTION OF
(3) STREET ADDRESS
(4) CITY AND STATE

Manual Information (From Revision Record)

(5) MANUAL TITLE	
(6) PUBLICATION NO.	(7) REVISION
(8) FCO'S INCORPORATED INTO MANUAL	

Equipment Information (From Equipment Nameplate & FCO Log)

(9) EQUIPMENT NO. AND DESCRIPTION
(10) SERIES CODE
(11) FCO'S INCORPORATED INTO EQUIPMENT

Comments

This form is not intended to be used as an order blank. Your evaluation of this manual will be welcomed by Control Data Corporation. Any errors, suggested additions or deletions, or general comments may be made above. Please include references to figure number, assembly number and design configuration level.

NO POSTAGE STAMP NECESSARY IF MAILED IN U. S. A.

FOLD ON DOTTED LINES AND STAPLE

CUT ALONG LINE

KSR 0502 PRINTED IN U.S.A.

STAPLE

STAPLE

FOLD

FOLD

FIRST CLASS
PERMIT NO. 8241

MINNEAPOLIS, MINN.

BUSINESS REPLY MAIL
NO POSTAGE STAMP NECESSARY IF MAILED IN U.S.A.

POSTAGE WILL BE PAID BY

CONTROL DATA CORPORATION
Technical Publications Department
7801 Computer Avenue
Minneapolis MN 55435

CUT ALONG LINE

FOLD

FOLD