7300 Processing Unit

Design Description Manual

Volume 3: Dedicated Resources

2501.003

MEMOREX

Computer System Products

February 1973 Edition

Memorex Corporation
Santa Clara, California 95052

Requests for copies of Memorex publications should be made to your Memorex representative or to the Memorex branch office serving your locality.

This publication is provided for informational purposes. Contact Memorex for the latest periodic enhancement.

A readers' comments form is provided at the back of this publication. If the form has been removed, comments may be addressed to the Memorex Corporation, Publications Dept., Santa Clara, California 95052.

7300 Processing Unit Design Description
Volume 3, Dedicated Resources

This volume provides Memorex Field Engineers with logic design descriptions for the 7300 Processing Unit's dedicated resources: Basic Data Channels 1 and 2 and integrated card device adapters, Integrated File Adapter (IFA) and Integrated Communications Adapter (ICA).

The three sections that comprise this volume are a part of the continuum of sections that make up the four volume, 7300 Processing Unit Design Description Manual. The four volumes are organized as follows:

Volume 1, Overview (2501.001)

Section 1. A general description of the 7300 Processing Unit.

Volume 2, Shared Resources (2501.002)

Section 2. A detailed description of main storage, control, timing and arithmetic parts of the 7300 Processing Unit.

Section 3. A detailed description of the formats, characteristics and implementation of the micro instructions associated with the 7300 Processing Unit.

Volume 3, Dedicated Resources (2501.003)

Section 4. A detailed description of the two basic data (selector) channels for the 7300 Processing Unit.

Section 5. A detailed description of the Integrated Communications Adapter (ICA) for the 7300 Processing Unit.

Section 6. A detailed description of the Integrated File Adapter (IFA) for the 7300 Processing Unit.

Volume 4, Power System (2501.004)

Section 7. A detailed description of the 7300 Processing Unit power system.

NOTE

Because Volume 1 provides an overview of the 7300 Processing Unit, it should always be used as an introduction to the other volumes in the set.

TABLE OF CONTENTS

TABLE OF CONTENTS (Continued)

TABLE OF CONTENTS (Continued)

TABLE OF CONTENTS (Continued)

TABLE OF CONTENTS (Continued)

LIST OF TABLES

LIST OF FIGURES

LIST OF FIGURES (Continued)

LIST OF FIGURES (Continued)

4. BASIC DATA CHANNELS (BDC)

INTRODUCTION

These processors' dedicated resources include the Basic Data Channel (BDC) logic (as a minimum), providing interface and control circuitry for communication with various peripheral equipments. Processor 2 circuitry also includes the Integrated Card Reader Adapter (ICRA) and the Integrated Reader/Punch Adapter (IRPA). The processor-state logic, functional capabilities, and characteristics of the two processor states are identical; the difference between them is that of the integrated adapters, absent in the BDC1 processor, and present in the BDC2 processor.

Both channels can communicate in the selector-channel mode with peripheral devices having external controllers. For BDC1, this is the only configuration possible; for BDC2, external assignments may be optional, depending upon the configuration.

Both the integrated adapters and external controllers contain the logic necessary to operate the I/O devices and to adapt device characteristics to the BDC format.

The processor-state logic, of which BDC logic is a principal control component, operates with the following characteristics:

- Only one I/O device is selected at a time.
- Once an I/O device is selected, it stays selected until a block of data (or, if data chaining is specified, a sequence of data blocks) is transferred.
- During basic data channel transfers, other devices may be executing operations not requiring basic data channel activity.

The processor states, in conjunction with the basic data channel logic, can perform the following operations with respect to adapters and control units connected to the channel.

- Select a device, dictate the device operation, and initiate and control the data transfer and termination under BDC2 software control.
- Select a device, dictate the device operation, allow the basic channel logic to initiate and maintain control of the data transfers (hardware operation).
- Respond to a device-initiated request, scan all devices, and determine (in order of priority) which device generated the request (polling).
- Select and read status indications from each I/O device.
- Perform data "turn-around" transfers within the basic data channel logic for diagnostic testing.

The basic data channels each have the following characteristics:

- 8-bit byte data transfers
- 16-bit data transfers to/from main storage when the Assembly/Disassembly option is present
- Parity generation and checking logic
- Basic data channel status indication logic
- Modular design
- Diagnostic capabilities
- Maximum data rate of 1110K bytes (7300) or 555K bytes (7200) per second.
- Hardware and software control of interface signals
- Ability to communicate with up to eight IBM 360/370 compatible external control units

BASIC DATA CHANNELS

GENERAL DESCRIPTION

Channel Configuration

The interconnection and positional relationship of the basic data channel logic, shared resources,

integrated adapters, local I/O devices, and external control units is shown in Figure 4-1. The tag lines (control) determine whether the information transferred on the parallel bus lines is data, status, address, or command information.

The start and termination of signals between the basic data channel logic and the control units and adapters are response-interlocked; that is, termination of one signal depends upon the initiation of a response signal. Interlocking allows channel communications with control units having different circuit speeds and data rates.

The integrated adapters and external control units communicate with the channel logic over a group of common signal lines. The common signal designation and the direction of travel are shown in Figure 4-2. Two cables carry all the common signal lines to the external control units. All the common signal lines are connected in parallel except the Select In and Select Out lines which are connected in a serial selection loop (see Figure 4-3). Each external control unit's selection circuitry is either in series with the Select In or the Select Out line. Regardless of which selection line is controlled by the selection circuitry, the control unit simply passes the other of the two selection signals to the next unit. Each control unit or adapter will disable the selection loop when selected, or will forward the selection signal to the next unit when not selected. This selection loop configuration and the channel logic provide two important features: polling and priority.

Polling enables the processor state program to select and service the control unit(s) and adapter(s), in order of priority, when an I/O operation has been requested by one or more control units or adapters. The priority scheme utilizes the fact that the external control units may be electronically connected into the Select Out or the Select In line with priority levels determined by their placement in this selection loop. For example, unit one in Figure 4-3 has the highest priority because it is connected first on the Select Out line and bypasses the Select In line, but it could become 4th in priority by being connected instead to the Select In line and bypassing the Select Out line.

The integrated adapters are always connected to the Select In line in the order of priority shown in Figure 4-3.

The Basic Data Channel logic and Integrated Adapter logic are contained on printed-circuit modules located in chassis one, row "C". The specific number of PC modules and their designation, location and function are shown in Figure 4-4. The basic data channel logic requires only two PC modules; the Assembly/Disassembly option is contained on an additional module, as is the External Interface logic.

Channel Requirements

Control units and adapters connected to the Basic Data Channel must be compatible with the BDC requirements as described in the following paragraphs.

A complete selection loop is required regardless of the number of control units or adapters in the loop. External control units within the cabling chain may be bypassed and the integrity of the selection loop maintained by disconnecting the cables from a unit and mating the cable ends together. The last physically connected unit in the external cabling chain may be removed, temporarily, for maintenance purposes, and the selection loop integrity retained by disconnecting the two cables and two terminators from the unit and then mating the cable ends and terminators. The basic data channel logic routes the SELECT OUT signal to either the external control units (EXT INTF present) or to the integrated adapters. The channel bypass logic directs the SELECT IN signal to those adapters present. In BDC1, the SELECT IN signal received back from the EXT INTF receiver passes straight through the bypass logic since integrated adapters are not present, thus maintaining the selection loop (Figure 4-3).

A control unit, while in an off-line mode, must gate all of its line transmitters off to prevent interference with communications between the channel logic and the other control units and adapters. Likewise, a control unit in the power-off state must not interfere with the Basic Data Channel operation.

Selection Loop Continuity

While in the off-line mode, each control unit must provide selection loop continuity by electrically bypassing the SELECT OUT signal. Each control unit must ensure that the method of bypassing the SELECT OUT signal in the power-off state does not interfere with the propagation of the SELECT OUT signal.

Each control unit must also ensure that SELECT OUT signal discontinuities, which may occur when another control unit on the external channel is powering up or down, do not affect the propagation of the SELECT OUT signal. One method of accomplishing this is by the use of a latch circuit. The latch is turned on by ANDing the SELECT OUT/IN signal and the HOLD OUT signal. The latch is turned off by the termination of the HOLD OUT signal. The latch circuit is in series with the control unit selection circuitry, providing a constant SELECT OUT/IN signal within the control unit, and therefore to the following control units, regardless of variations in the incoming SELECT OUT/IN signal.

The integrated adapters also ensure SELECT OUT signal continuity by the use of a similar latch circuit.

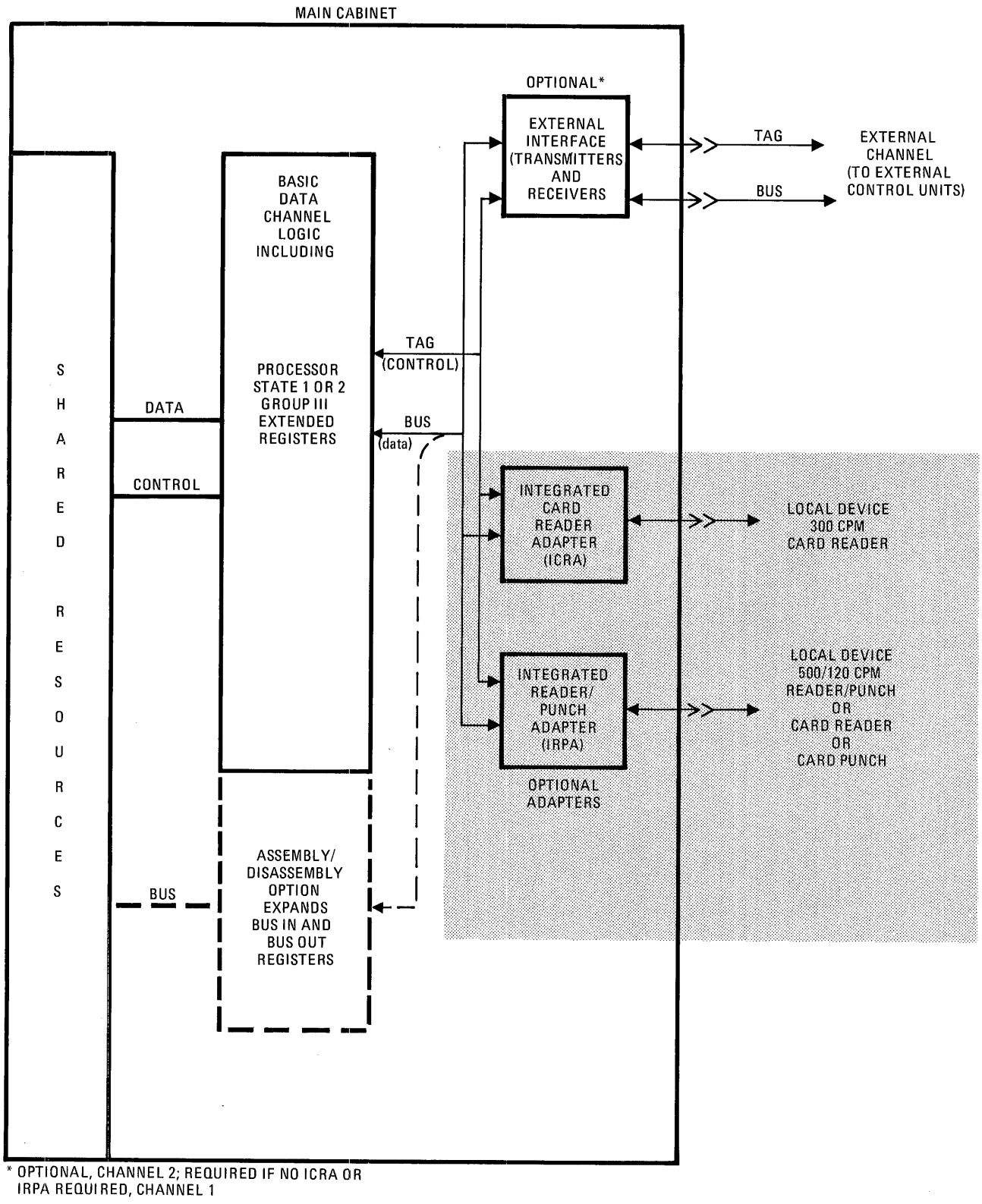


Figure 4-1. Basic Data Channel Processor Configuration

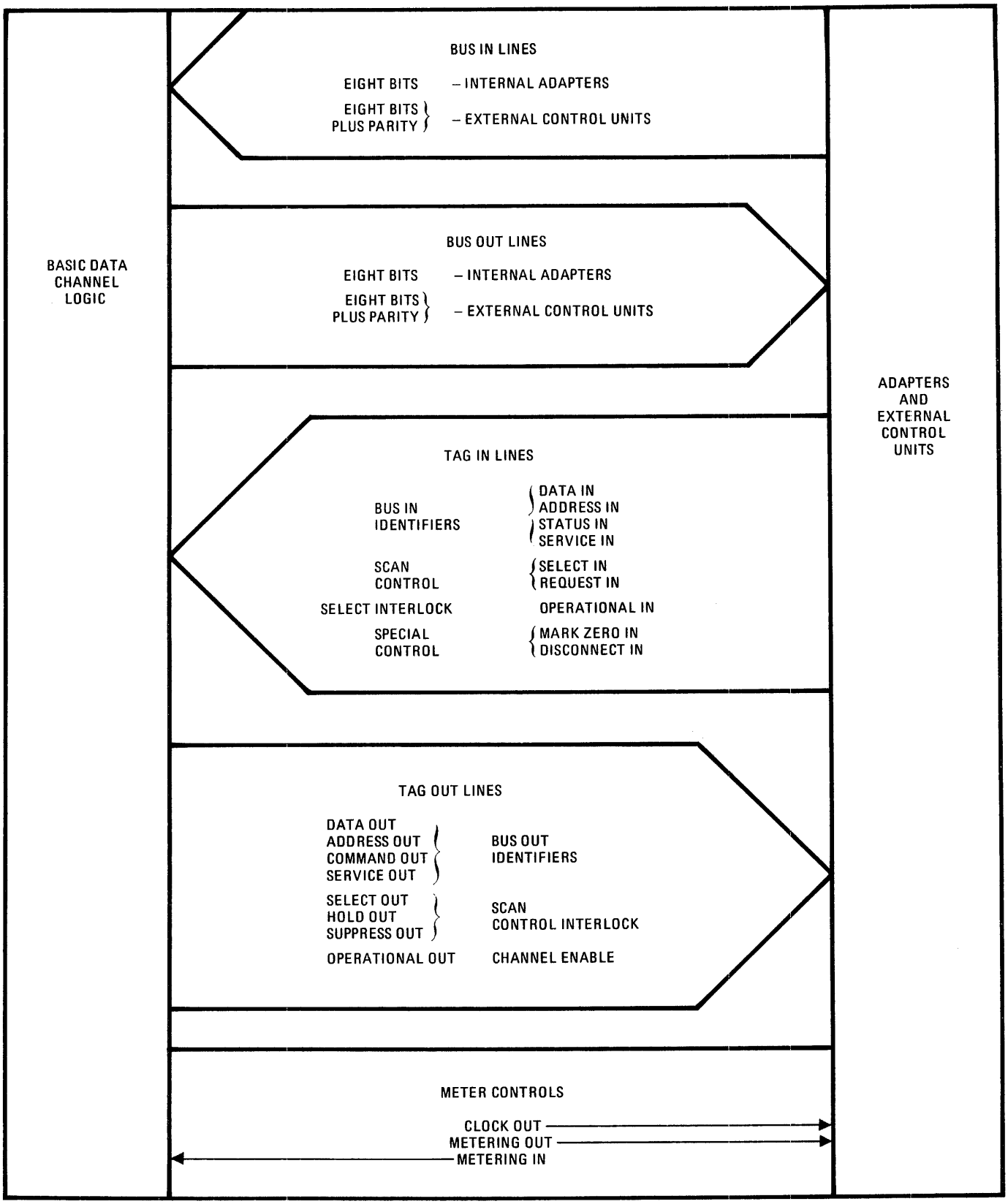


Figure 4-2. BDC Common Signal Lines

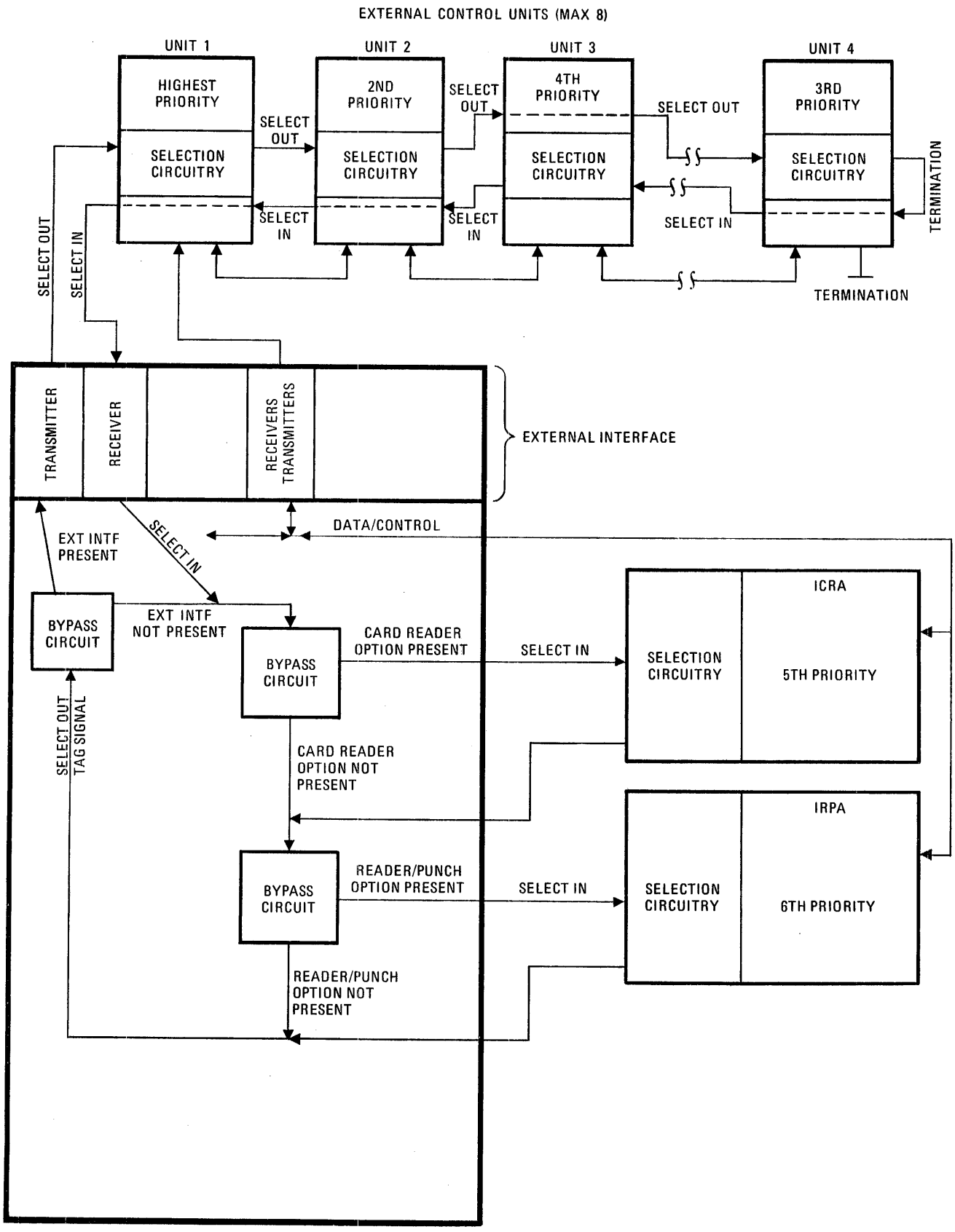


Figure 4-3. BDC Interconnection Diagram

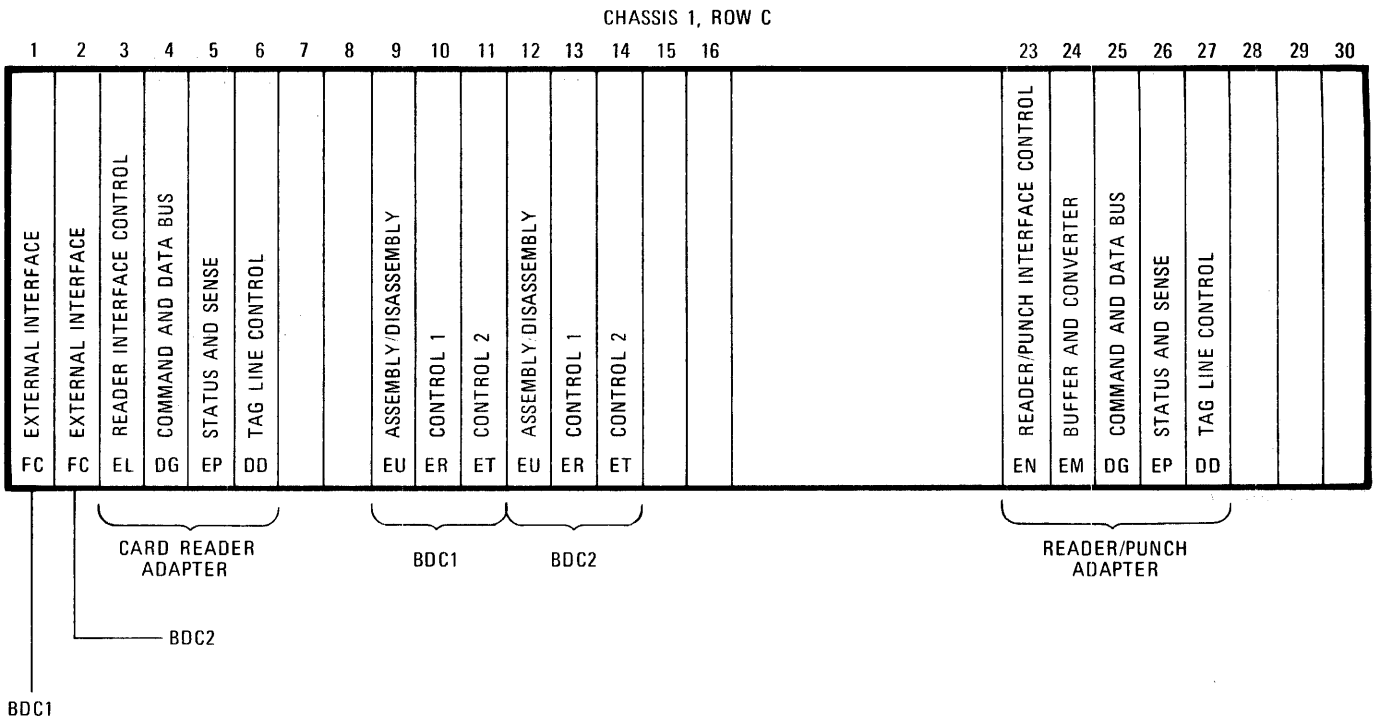


Figure 4-4. Printed-Circuit Module Placement

Addressing

An eight-bit address is used to select the adapters, control units and I/O devices. The adapters and single device external control units are assigned a unique eight-bit address in the range of 0 through 255. The address is assigned to a control unit or adapter at the time of installation.

Multiple I/O devices controlled by one control unit are assigned addresses within a contiguous number set. The higher-order bit positions of the address within a set identify the control unit, and the lower-order bit positions identify the device.

Adapters and control units respond to their address whether their respective device is ready or not-ready. A not-ready condition is indicated by the *Unit Check* bit in the Status Byte and the *Intervention Required* bit in the sense byte.

If none of the adapters or control units recognize an address, the SELECT OUT signal returns as the SELECT IN signal. The addressed device appears as non-operational, which indicates one or more of the following conditions:

- the unit is not installed.
- the unit is in the off-line mode.
- the unit is in a power-down state.

Command Byte

The Command Byte comprises eight bits of command information placed on the Bus Out lines. The Command Byte is decoded by the addressed adapter or control unit only during aBDC processor-initiated selection sequence. The low-order bit positions indicate the basic type of operation, and the high-order bit positions are a modification code for expanding basic operations at the adapter or I/O device.

The particular modifier and operation codes for external control units are specified in the appropriate equipment publication. The integrated adapter operation codes are detailed in the appropriate sections of this manual.

The Command Byte bit designations for the basic operations are listed below:

Command	Bit Positions							
	0	1	2	3	4	5	6	7
Test I/O	0	0	0	0	0	0	0	0
Sense	M	M	M	M	0	1	0	0
Read Backward	M	M	M	M	1	1	0	0
Write	M	M	M	M	M	M	0	1
Read	M	M	M	M	M	M	1	0
Control	M	M	M	M	M	M	1	1

M = Modifier Bit

- The *Read* command initiates the transfer of data from the control unit or adapter to the processor logic. The data is obtained from the recording media of the selected I/O device.

- The *Read Backward* command also initiates data transfer to the channel as does the *Read* command. With the *Read Backward* command, the data bytes are transferred from the adapter or control unit in reverse order to that of the *Read* command.

- The *Write* command initiates the transfer of data from the processor logic to the adapter or control unit.

- The *Control* command initiates a function or operation in the adapter or control unit as determined by the modifier bits. A *Control* command may use second level addressing requiring several bytes of data to complete the control operation, or it may initiate film advance as in the MEMOREX Model 1603 Microfilm Printer.

  A *Control* command with all-zero modifier bits performs no operation at the adapter or control unit but does allow time for certain control units to check conditions before releasing the channel processor.

- The *Sense* command proceeds as does a *Read* command, only the data transferred is obtained from the selected sense points within the unit rather than from a record source.

  A *Sense* command with all-zero modifiers is a basic *Sense* command to all adapters and control units and will not initiate any operation other than sensing. The basic *Sense* command to an available control unit is accepted even though the addressed device is in a not-ready state.

- The *Test I/O* command requests status information from the addressed adapter or control unit. If status information is available, the status bits are placed on the Bus In lines to the channel logic. If no outstanding status information is available, a zero Status Byte is placed on the Bus In lines (see Status Information). If the selected adapter or control unit is still performing the previous operation, status information is not available and, as such, only the *Busy* bit is presented as status information.

  A *Test I/O* command byte during a processor-initiated selection sequence does not initiate an operation as do the other Command Byte codes.

Status Information

The information on the Bus In lines is a Status Byte when the STAT IN signal is active. The Status Byte shows the status indications of the control unit or adapter whose address appeared on the Bus In lines when ADRS IN was active. During the short-busy sequence when no ADRS IN signal occurs, it is assumed that the status information pertains to the addressed control unit, device, or adapter.

The timing and causes of the status information for the external control units is specified in the appropriate equipment publications.

The Status Byte has the following format on the channel Bus In lines.

Bit Position	Designation
P	Parity
0	Attention
1	Status Modifier
2	Control Unit End
3	Busy
4	Channel End
5	Device End
6	Unit Check
7	Unit Exception

Once the status indication is accepted by the processor, it is cleared and not presented again. The Status Byte is transferred to the channel in the following situations:

- To indicate the initial-selection status.
- To indicate the *Channel End* status at the termination of data transfer.
- To indicate the *Device End* and associated conditions to the processor logic. The I/O device remains busy during an operation until the processor accepts the *Device End* status indication.
- To indicate *Control Unit End* or *Device End* status which signals that a control unit or device that was busy had been interrogated and is now free.
- To indicate any previously stacked status when allowed to do so.
- To indicate any externally initiated status to the channel *(Attention* and/or *Device End* because of not-ready-to-ready transition).

The following is a brief description of the Status Byte bit meanings.

- Attention: This state is generated when some asynchronous condition occurs in the

I/O device. *Attention* is not associated with initiation, execution, or termination of any input/output operations, but may occur with *Device End.*

- Status Modifier: This bit is used by control units in the following situations:

  1) During an initial selection sequence control, units unable to provide current status-in response to a *Test I/O* command present the *Status Modifier* bit alone.

  2) Shared control units present the *Status Modifier* bit with the *Busy* bit during initial selection sequence to differentiate a busy control unit from a busy device.

  3) Control units designed to recognize special ending conditions *(Search Equal* on a disk) present the *Status Modifier* bit with *Device End* when the condition occurs.

  4) Control units capable of requesting a retry of a command, in order to recover from a transient error, or because the state of the control unit or device prevented the execution of the command when it was previously issued present the Status Modifier bit with Channel End and Unit Check and Mark In tag line.

- Control Unit End: These conditions are provided only by shared control units or control units with multiple I/O devices, and only when one or both of the following conditions occur:

  1) The control unit was interrogated while it was in the busy state.

  2) The control unit detected an unusual condition while busy, but after Channel End was accepted.

- Busy: This can occur only during a Basic Data Channel-initiated selection sequence, and indicates that the control unit or adapter cannot execute the command because a previously initiated operation is being executed or because outstanding status conditions exist. An operation is being executed from status acceptance during initial selection until *Device End* acceptance.

  Busy is indicated to Test I/O only if a previously initiated operation is still being executed and no end status is available.

- Channel End: This is caused by the completion of the data transfer portion of an input or output operation. Each operation causes only one *Channel End* status.

  For the adapters, the *Channel End* condition occurs on completion of the data transfer between the adapter buffer and the channel. During Control operations, *Channel End* is usually generated after the control information is transferred to the control unit or adapter; although for short operations it may be delayed until end of the operation. Operations that do not involve data transfer provide the *Channel End* condition during the initial selection sequence (immediate commands).

- Device End: This is caused by the completion of an I/O operation at the I/O device; it indicates overall completion of the current operation. Each I/O operation causes only one *Device End* condition. It is generated either simultaneously with the *Channel End* or presented later.

- Unit Check: This situation indicates the I/O device, control unit, or adapter has detected an unusual condition that is detailed by the information available to a *Sense* command. An error condition causes the *Unit Check* indication only when it occurs during execution of a command or during some activity associated with an I/O operation.

  Unless the command is designed to cause *Unit Check* (such as *Rewind And Unload* on magnetic tape), it is not indicated if the command is properly executed even though the device has become not-ready during, or as a result of, the operation. Similarly, *Unit Check* is not indicated if the command can be executed with the device not-ready.

  If, during an initial selection sequence, the device detects that the command cannot be executed, *Unit Check* is presented to the processor and it appears without *Channel End, Control Unit End,* or *Device End.* Such unit status indicates that no action has been taken at the *Device In* response to the command.

- Unit Exception: This means that the I/O device detected an unusual condition such as end-of-file. This code has only one meaning for any particular command and type of I/O device. A *Sense* operation is not required as a response to the acceptance of a *Unit Exception* condition.

  A *Unit Exception* condition can be generated only when the I/O device is executing an I/O operation, or when the device is involved with some activity associated with an I/O operation and the condition is of immediate significance. If a device detects a *Unit Exception* condition

during the initial-selection sequence, it is presented without *Channel End, Control Unit End,* or *Device End.*

Sense Information

This information is presented in response to a *Sense* command during a BDC-initiated selection sequence. (See *Sense* under Command Byte.) The information provided by a basic sense operation is more detailed than that supplied by the Status Byte. The Sense Bytes, for example, may describe reasons for the *Unit Check* indication in the Status Byte or may indicate when a device is in the not-ready state.

All sense information significant to the use of the I/O device is usually provided in the first two Sense Bytes. The amount and meaning of the sense information is peculiar to each type of I/O device and is specified in the appropriate equipment publication.

The sense information from the last I/O operation or control unit action is cleared by the next command addressed to that control unit, provided the *Busy* bit was not set in the *Initial Selection* Status Byte and the command is addressed to the device that causes the sense. The commands *Sense, Test I/O* or *No-Operation,* presented to that device, will not clear the sense information.

Sense information may also be generated or changed as a result of asynchronous actions of the device.

A command code with invalid parity causes the sense information to be replaced with the new sense information which contains Bus Out Check.

The first six bits of the first sense byte have the same designation for all I/O devices:

Bit Position	Description
0	Command Reject
1	Intervention Required
2	Bus Out Check
3	Equipment Check
4	Data Check
5	Over Run

- The *Command Reject* condition occurs when the device receives a command it is not designed to execute, receives a command it cannot execute because of its present state, or recognizes an invalid sequence of commands.
- *Intervention Required* is generated when the last operation could not be executed because of a condition that requires intervention at the I/O device. This bit indicates conditions such as an empty magazine (hopper) in a card punch or a lack of paper in the printer. *Intervention Required* is also turned on when the addressed device is in the not-ready state, is in a test mode, or is not connected to the control unit.
- *Bus Out Check* results when the external control units receive a Data Byte or a Command Byte with invalid parity. Invalid parity in an output Data Byte does not cause the operation to be terminated prematurely, whereas invalid parity in a Command Byte does prevent command execution.
- *Equipment Check* indicates detection of an equipment malfunction in the adapter, control unit or I/O device. It stops transmission and terminates the operation prematurely.
- *Data Check* is generated when invalid data has been detected by the control unit during the reading of data from the record source. The control unit forces correct parity on the data sent to the processor logic via the external channel. During writing operations *Data Check* indicates invalid data may have been recorded at the I/O device. *Data Check* causes premature termination only when the errors prevent meaningful continuation of the operation.
- *Overrun* results when the processor logic fails to respond on time to a request for service from a non-buffered adapter or a control unit operating with a moving medium. On an output operation, *Overrun* indicates the data recorded on the device may be invalid. This normally stops data transfer and terminates the operation in the same manner as a *Stop* sequence control (discussed later).

  Overrun conditions also occur during command chaining when the I/O device receives the new command too late.

Signal Interlock Summary

The following list of direct-current interlocking signal rules must be adhered to by all adapter and control units using either Basic Data Channel.

1. No more than one bus-in identifier may be active at any given time, except for the possible overlap of DAT IN and SRV IN.
2. A bus-in identifier may become active only when all bus-out identifiers are inactive, except during the short-busy or high speed data transfer sequences.
3. A bus-in identifier may go inactive only after the activation of a responding bus-out identifier, except for STATUS IN in the short busy sequence.

4. OPERATIONAL IN cannot go active unless OPERATIONAL OUT is active, and must terminate if OPERATIONAL OUT terminates.
5. OPERATIONAL IN cannot terminate until either:
   - SELECT OUT ends, and a bus-out identifier is active in response to the last bus-in identifier of any sequence.
   - OPERATIONAL OUT terminates.
   - an interface-disconnect control signal is given.
6. During an interface-disconnect operation ADDRESS OUT is the only signal which may be active with other bus-out identifiers.
7. None of the outgoing common signals lines, except SUPPRESS OUT, have meaning when OPERATIONAL OUT is inactive.
8. SELECT OUT can go active only if OPERATIONAL IN and SELECT IN are inactive.
9. SERVICE OUT, DATA OUT and COMMAND OUT can go active only in response to an active bus-in identifier.
10. ADDRESS OUT for a processor-initiated selection sequence can go active only if SELECT IN and SELECT OUT are inactive in the processor logic.
11. Once ADDRESS OUT has become active for the interface disconnect operation it must not end until OPERATIONAL IN terminates.
12. Once ADDRESS OUT and SELECT OUT have become active during a processor-initiated selection sequence, ADDRESS OUT must stay active until after SELECT IN or OPERATIONAL IN becomes active (or if in a short-busy response, until after STATUS IN goes inactive).

Basic Data Channel Control Instructions

The three machine-language instructions (MLI's) used by the processor state to communicate with the I/O devices via the Basic Data Channel processor logic are as follows:

- Input from I/O register (INP)
- Output to I/O register (OUT)
- System I/O (SIO)

These instructions deal with the processor state's Extended register file, Group III, which comprises the five registers residing in the Basic Data Channel logic.

The microprograms which implement the above MLI's use *Store Register* and *Load Register* micro-instructions in communicating with the extended registers. *Store Register* micro-instructions transfer information from a shared resources register to one or more of the Extended registers in the BDC logic. The *Load Register* micro-instructions transfer information from the extended registers in the BDC logic to a shared resources register.

The 4-bit R_2 portion of the *INP* and *OUT* instruction is the Extended register designator. This 4-bit designation is sent out through the I-O register select logic as Extended Register Number Group Three (ERNG3) bits 00, 01, 02 and 03.

The *SIO* instruction does not contain an Extended register designator at the machine-language level. The micro-instructions within the *SIO* microprogram contain the Extended register numbers required.

The five Extended registers in the Basic Data Channel logic are selected using an ERNG3 bit-oriented scheme. The relationship between the ERNG3 bits of a store-register micro-instruction and the Basic Data Channel register(s) selected is shown in Table 4-1.

Table 4-1. ERNG3 Bit Assignment: Store Micro-Instruction

ERNG3 Bits				Extended File Register Written (Destination)
00	01	02	03	
0	0	0	1	Tag Out Register (TOR)
0	0	1	0	Channel Control Register (CCR)
0	1	0	0	Bus Out Register (BOR)
1	0	0	0	Byte Count Register (BCR)
1	1	1	1	TOR, CCR, BOR and BCR
0	0	0	0	None

A *Load Register* micro-instruction transfers information from four Basic Data Channel registers and the Tag In lines. The registers and/or lines read are selected by the ERNG3 bits as shown in Table 4-2.

Table 4-2. ERNG3 Bit Assignment: Load Micro-Instruction

ERNG3 Bits				Extended File Registers Read (Source)
00	01	02	03	
X	X	0	0	Bus In Register (BIR)
X	X	0	1	TAG IN Lines and TOR
X	X	1	0	CCR
X	X	1	1	BCR

LOGIC DESCRIPTION

Block Diagram

The logic for a Basic Data Channel processor state contains two 8-bit data-handling registers (Bus Out and Bus In); two control registers (Channel Control and Tag Out); two 8-bit multiplexers (Byte 0 and Byte 1); a 16-bit Byte Count register; and associated control logic as shown in the block diagram (Figure 4-5). The signal relationships and data paths between the shared resources, the Basic Data Channel logic, and the adapters and external control units are also shown.

The expansion of the Bus In and Bus Out registers and data paths for the Assembly/Disassembly option is indicated by the shaded area. This option also expands the Channel Control register by adding the Assembly/Disassembly flip-flop.

Register Functions

Bus Out Register (BOR)

Information placed in the BOR originates in either a processor state's Basic register file or main storage and is transferred via lines 8 through 15 of the Extended Register Output (ERO) data path (Figure 4-6), when the Assembly/Disassembly option is not present. The contents of the BOR are interpreted by the adapters and control units as *device address, command,* or *data* as directed by individual Tag Out lines.

The BOR is a double rank register with each rank consisting of two 4-bit register MSI elements. The input clocking to the first rank (BOR1) is the WRITE REGISTER (WR REG) signal. It is enabled only when extended register number group 3, bit position one (ERNG3-01) is set. The contents of rank one is gated into rank two (BOR2) by STRBIOR2 signal generated by the control logic. The eight BOR2 outputs are enabled through a rank of inverters to the integrated adapters (BDC2) and the transmitters on the external interface board, with the ODD BYT signal. The ODD BYT signal is produced by the BYT MD signal which is always active when the Assembly/Disassembly option is not present.

A parity bit is also generated and sent to the external control units via the EXT INTF module. The PARITY Bus Out signal is used to maintain odd parity (an odd number of 1's including parity bit) on the nine Bus Out lines. The parity generating element monitors the BOR2 contents and places a logic 1 or a logic 0 on the Parity Bus Out line depending on the number of logic 1's contained in the BOR2.

Information placed in the BOR is transferred from the shared resources using all sixteen ERO data paths, when the Assembly/Disassembly option is present (Figure 4-6). The BOR becomes a sixteen-bit double-rank register utilizing the same enables and clocking signals. The sixteen BOR2 outputs are enabled through a rank of inverters to the integrated adapters (BDC2) and the transmitters on the external interface board one byte at a time.

The Odd Byte flip-flop, on the Extended register board, is used to control enabling by the BDC processor logic. The Odd Byte flip-flop stays set (enabling odd byte) during all sequences except the hardware-controlled data transfer sequences. During this sequence the Odd Byte flip-flop changes state after each byte transfer, thus alternately enabling odd (bits 8-15) and even (bits 0-7) bytes. Of course this only happens when the Assembly/Disassembly option is present and the Assembly/Disassembly Mode flip-flop in the Channel Control register is set. When the Assembly/Disassembly Mode flip-flop is clear (BYT MD) and the option is present the BDC1 still performs straight byte transfers just as though the option were not present.

Bus In Register (BIR)

The BIR is also a double-rank register with each rank consisting of two 4-bit register MSI elements when the Assembly/Disassembly option is not present. The register inputs are clocked into rank one (BIR1) by the STR BIR1 signal when the odd byte enable is present. BIR information can indicate status, device address, or data as determined by a specific Tag In line.

Figure 4-7 is a functional block diagram of the bus-in configuration. The BIR inputs connect directly to the eight Bus In lines from the integrated adapters and to eight of the Bus In Line Receivers on the external interface module. The BOR2 output is also connected through one stage of inversion to the BIR1 input and is enabled only during diagnostic operations.

The eight BIR input lines also connect to the parity check element inputs. The PARITY Bus In signal from the external control units controls the parity check mode. When the PARITY Bus In signal is a logic 1, it places the parity checker in the even mode, checking for an even number of 1's on the BIR1 inputs. Conversely, if the PARITY Bus In signal is a logic 0, it places the parity checker in the odd mode, checking for an odd number of 1's.

The parity check element monitors the nine Bus In lines, checking for an odd number of 1's, and indicates a parity error, when detected, with the CHANNEL PARITY ERROR (CHPARERR) signal.

The contents of BIR1 are gated into BIR2 by the STRBIOR2 signal generated by the control logic. Information leaving the BIR2 is selected via the Byte One multiplexer for the Extended register input (ERI) bits 8 through 15. The BIR2 output is selected when Extended Register Number Group Three bit positions 2 and 3 are clear.

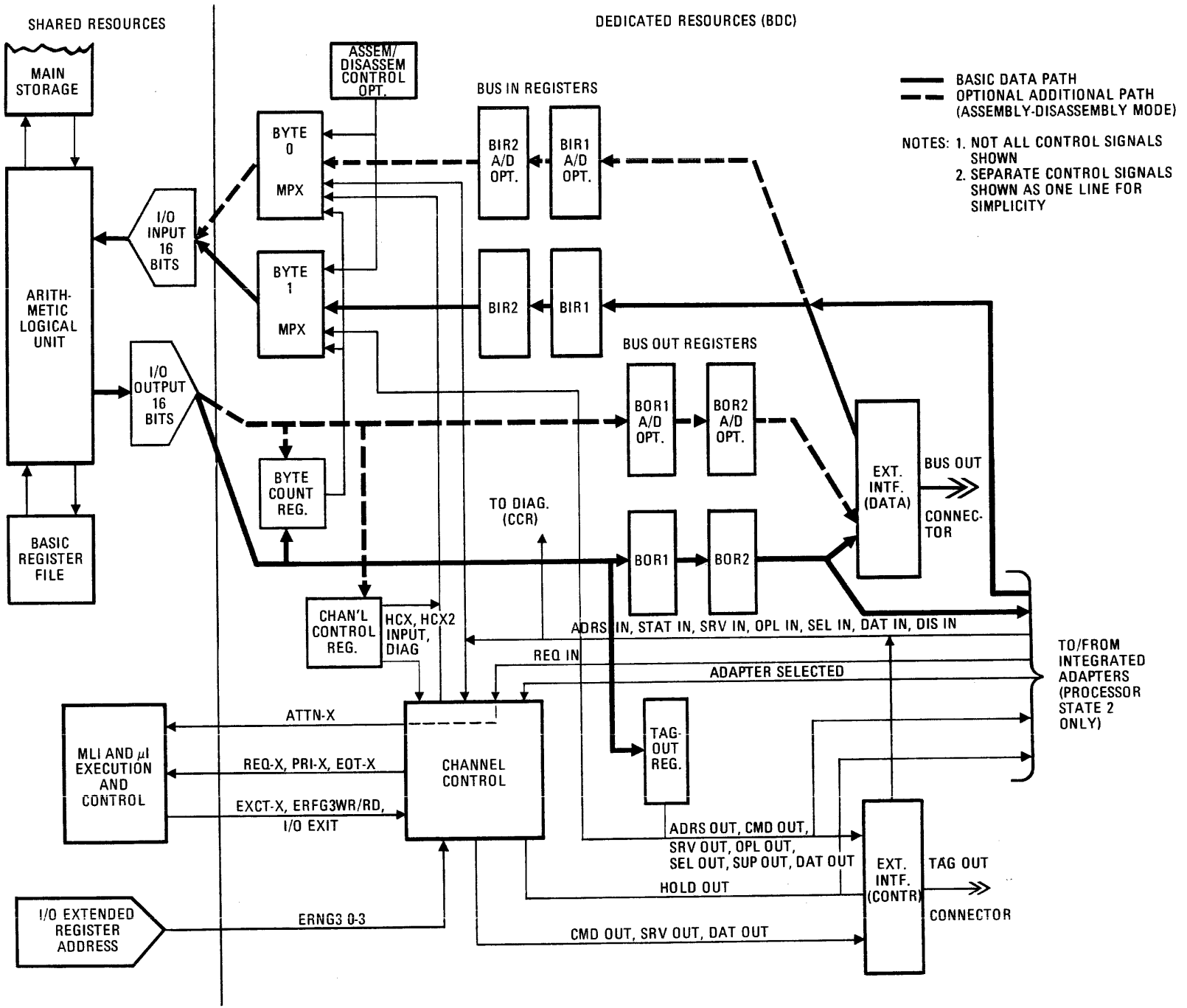


Figure 4-5. BDC Processor Block Diagram

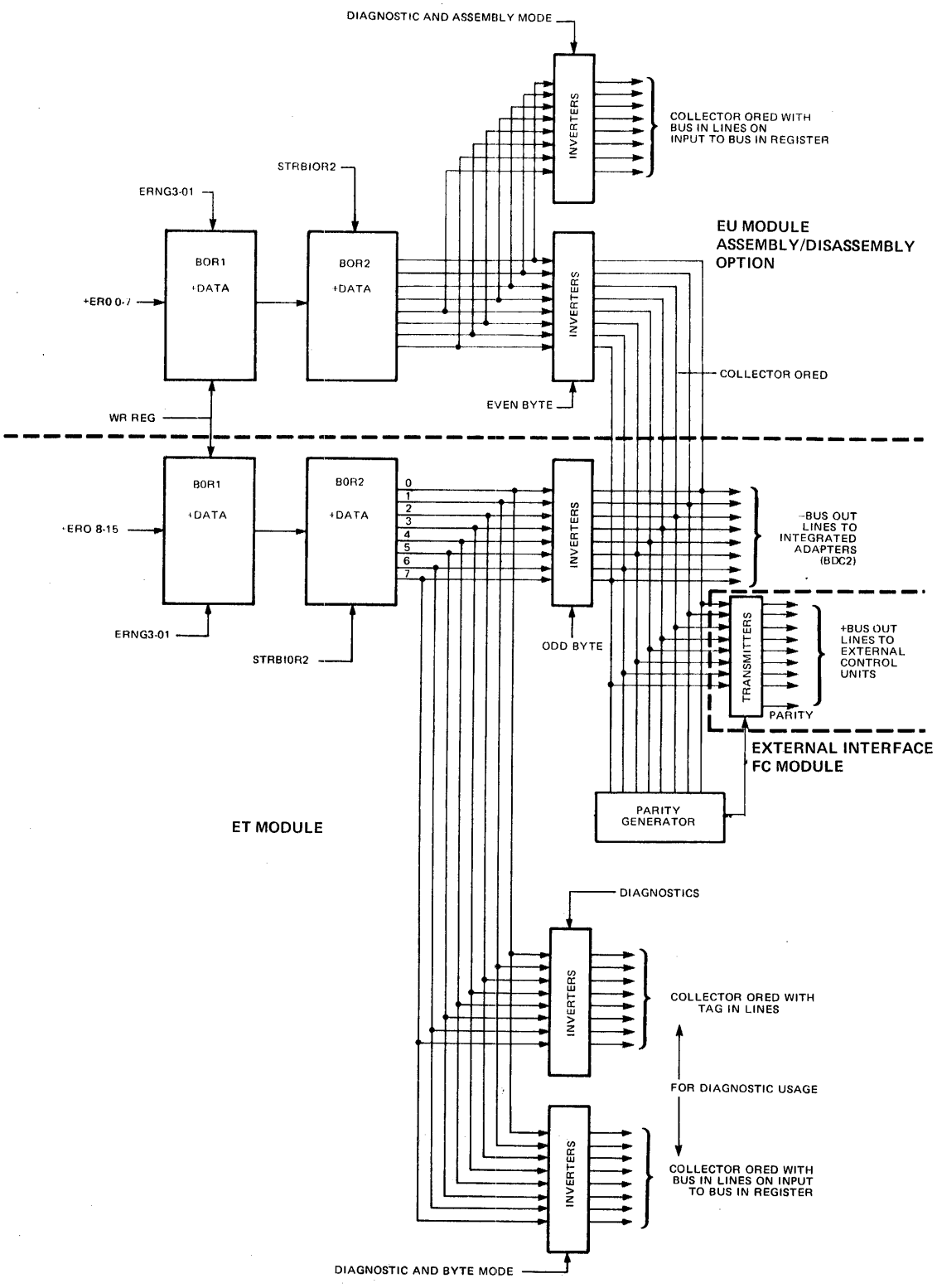


Figure 4-6. Bus Out Configuration

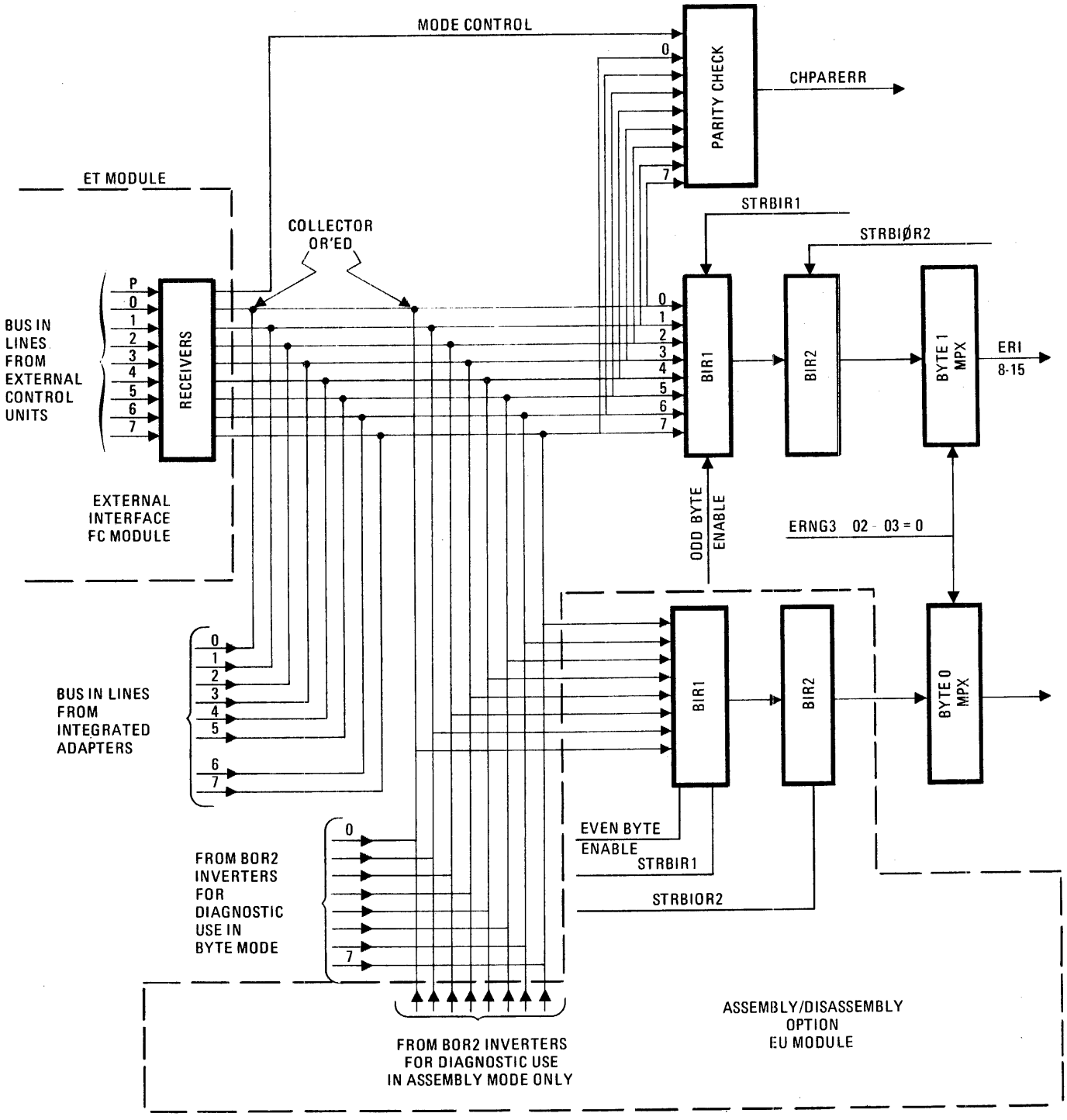


Figure 4-7. Bus In Configuration

The BIR also becomes a 16-bit double-rank register when the Assembly/Disassembly option is present (Figure 4-7). The eight bits of information from the integrated adapters or from the external interface module are clocked into both halves of rank 1 by the STR BIR1 signal. Into which half of the BIR the information is enabled is controlled by the Odd Byte flip-flop. During data transfer sequences the Odd Byte flip-flop changes state after each data transfer, thus alternately enabling the data into the odd byte (bits 8-15) and the even byte (bits 0-7) halves of the register.

The Odd Byte flip-flop is controlled by the hardware data transfer logic on the ER board. As with the BOR, the Odd Byte flip-flop is set for all sequences except the data transfer. Therefore, the even byte (0-7) half of the BIR is only used during data transfers, provided the Assembly/Disassembly option is present and the Assembly/Disassembly Mode flip-flop in the CCR is in the *assembly* state (set).

All sixteen bits of information from BIR2 are selected via the input multiplexers and placed on the ERI 0-15 data paths to the shared resources during data transfer sequences. The BOR2 output which is added when the Assembly/Disassembly module is present is also connected to the BIR input for use during diagnostic operations in the assembly mode.

Tag Out Register (TOR)

The TOR (logic page 1C14) consists of two 4-bit register MSI elements, the same as BOR1. The TOR input is clocked by the WRITE REGISTER (WR REG) signal, but is enabled only when ERNG3-03 is present. TOR inputs connect to the ERO bits 8-15 through one stage of inversion. Seven of the eight TOR FF's hold bus-out identification, interlock and scan control information. The TOR bit assignment is listed in Table 4-3.

Table 4-3. Tag Out Register

Bit Position	Title	Abbreviation	Function
8	Address Out	ADRS OUT	Bus Out ID
9	Command Out	CMD OUT	Bus Out ID
10	Service Out	SRV OUT	Bus Out ID
11	Operational Out	OPL OUT	Channel Enable and Interlock
12	Select Out	SEL OUT	Scan Control
	Hold Out	HLD OUT	Scan Control
13	Suppress Out	SUP OUT	Scan Control and Interlock
14	Data Out	DAT OUT	Bus Out ID
15	Not Used		

All TOR outputs (except SEL OUT) connect directly to the integrated adapters and through the external interface module transmitters to the external control units. The Basic Data Channel control logic also utilizes the TOR information during operational sequences. The contents of the TOR may also be routed through the Byte One Multiplexer to the ERI lines 8 through 15.

The following paragraphs provide a brief definition of the TAG OUT signals. An in-depth discussion of the channel logic usage of the TAG OUT signals and Tag In lines is included in the operational sequences presentation.

- Address Out: This signal indicates the information on the Bus Out lines is an I/O device address and all adapters and control units are to decode it.
- Command Out: This signal is sent to the selected adapter or control unit during initial selection to indicate that the Bus Out lines contain a command byte. During other sequences this signal is sent in response to three of the TAG IN signals: in response to ADRS IN it means *proceed*, in response to STAT IN, it means *stack status*, and in response to SRV IN or DAT IN, it means *stop*.
- Service Out: This signal is sent to the selected adapter or control unit in response to SRV IN or STAT IN signals. A SERVICE OUT signal indicates the processor has accepted or presented data or has accepted status information.

NOTE

The COMMAND OUT, SERVICE OUT and DATA OUT Tag Out signals are also controlled by the channel control logic as well as by the TOR (page 1C13).

- Operational Out: With exception of SUP OUT, all signals from the processor logic to the adapters and control units are valid only while the OPERATIONAL OUT signal is present.
- Select Out: This line, which connects via a serial selection loop to all adapters and control units, is made active during an initial selection sequence or poll sequence. The SELECT OUT signal allows an adapter or control unit to be selected during an initial selection sequence or poll sequence by sending an OPERATIONAL IN signal to the channel logic.

  The Select Out line connects to both the external interface module and to the bypass

logic (page 1C1302) where its signal is enabled to the Integrated Card Reader Adapter (ICRA) if the external interface option (external interface module) is not present. If the option is present (EXTINFOP signal low) the SEL IN signal from the external interface module receiver is enabled to the ICRA as the EXT INTF SEL IN signal. The SEL IN back from the ICRA is enabled to the next adapter (Integrated Reader/Punch Adapter (IRPA) if that option is present). If the ICRA were not present the SEL IN signal would be routed directly to the IRPA. The SEL IN signal is routed to all options present and finally arrives on the SEL IN Tag In line if OPERATIONAL IN is not made active by a control unit or adapter.

- Hold Out (HLD OUT): This signal, in conjunction with the latch circuits in the external control units and adapters, performs three distinct functions:

  1. During selection sequences HLD OUT is ANDed with SEL OUT, in each control unit and adapter, to validate SEL OUT propagation.
  2. During data transfer sequences, it ensures SEL OUT continuity (see Select Out).
  3. When going inactive, it clears the serial select loop by clearing the Select-Out latch in all control units and adapters in parallel.

  Once HLD OUT becomes inactive, it may not become active again for 4 microseconds.

  The control logic (page 1C1401) monitors the *Sel Out* bit of the TOR and generates the HLD OUT signal when the SEL OUT signal is active.

  When the SEL OUT signal goes inactive the HLD OUT signal is also deactivated and cannot be activated by SEL OUT again for 4 microseconds. The HLD OUT signal is disabled by the N121 one-shot circuit, which produces a 4-microsecond logic 0 on the Q output each time the SEL OUT signal goes inactive.

- Suppress Out: This signal is used both alone and in conjunction with the other tag out lines to provide the following special functions: *suppress data, suppress status, command chaining,* and *selective reset.* Each of these functions will be described in subsequent sections.

- Data Out: This signal is sent to the selected adapter or control unit in response to DAT IN and indicates the channel has accepted or presented the data byte. Because of signal turn-around time on the cables, the DAT OUT and DAT IN signals are used in conjunction with and overlap the SRV IN and SRV OUT signals during the hardware controlled data transfer operation to increase the data transfer rate. The control logic only uses the DAT OUT-DAT IN signals with external control units that are equipped to handle it.

Channel Control Register (CCR)

The CCR consists of six edge-triggered flip-flops, five located in the Basic Data Channel logic on the Extended register (ER) module, and one contained on the optional Assembly/Disassembly (EU) module. The CCR receives its information from the processor state via ER0 bits 0 through 3 and 7. The REGISTER CLOCKING signal is generated by ANDing the WR Reg FF output pulse with ERNG3-02.

Outputs from the CCR connect directly to the processor control logic with each flip-flop directing a specific channel operation. The ER0 bit assignment for the CCR is shown in Table 4-4. Bit 0 connects directly to the HCX FF and through one stage of inversion to the HCX2 FF (page 1C13 Loc). When the CCR is clocked with bit 0 set, the HCX FF sets and the HCX2 FF clears. This is the required state of the two flip-flops when beginning a hardware-controlled data transfer.

Table 4-4. ER0 Bit Assignment in Channel Control Register

Bit Position	Title	Abbreviation
0	Hardware Controlled Transfer	HCX
	Hardware Controlled Transfer 2	HCX2
1	Input	INPUT
2	Diagnostic	DIAG
3	Data Chaining	DAT CHNG
7	Assembly/Disassembly Mode	ASMDASMD

The following is a definition of the CCR flip-flop usage.

- Hardware Controlled Transfer: The adapter or control unit initiates each data byte transfer. In response, with HCX set, the channel logic deskews the data, generates or checks parity, initiates the correct response to the control unit or adapter, and requests data transfer to or from the shared resources. The processor program transfers the data, updates the storage address and compares the updated address with the last address plus one.

- Hardware Controlled Transfer 2: While set, this flip-flop enables rank 1 to rank 2 of both the BIR and BOR, thus effectively making them single-rank registers. When clear, the HCX2 FF forces the channel control logic to gate data from rank 1 to rank 2 of the BOR and BIR. The HCX2 FF is cleared at the same time the HCX FF is set. The cleared state of the FF, therefore, is defined as +HCX2.
- Input: This flip-flop controls the direction of byte transfers. When set, it indicates an input operation and enables the control logic to gate data into the BIR1 during the input (read) operation. The Input FF, when cleared, indicates an output (write) operation.

NOTE

The *Input* bit position in the CCR must be set during the processor state's last major cycle prior to the start of an input hardware controlled transfer operation.

- Diagnostic: With the Diag FF set and OPL OUT and SUP OUT signals absent (page 1C14), the information in the BOR2 is placed on the Tag In lines and at the input to the BIR1. The BIR is loaded, when the proper Tag In lines are activated, by setting the appropriate bit positions in the BOR. The basic data channel diagnostic feature provides information turn-around for register operational checks and limited control sequence simulation.
- Data Chaining: This FF set inhibits the clearing of the HCX FF, in the CCR, by the I/O Exit FF output at the end of each data segment transfer. This allows the processor state to set up control for the next data segment and to initiate the hardware-controlled transfer again without the channel logic losing control of the data transfer sequence.
- Assembly-Disassembly Mode: The flip-flop determines whether the data transferred to/from main storage will be one byte or two bytes wide. When it is set, the hardware data transfer logic is allowed to assemble and disassemble bytes in the BIR and BOR and to perform two-byte-wide transfers to and from main storage. When it is clear, the input-output operations are performed in the one-byte-wide mode.

The state of the CCR FF's along with channel logic status information is selected via both multiplexers and routed to the ERI lines when ERNG3-02 is high and ERNG3-03 is low. Table 4-5 indicates the ERI bit assignment during a *Load Register* micro-instruction with the CCR selected.

The channel status bits indicate such things as sequence termination conditions, parity error, interlock signal errors, external control unit status and assembly/disassembly (EU) module presences.

- Control Check: This flip-flop, when set, indicates the existence of too many active TAG IN or TAG OUT signals. The Cntl Chk FF (page 1C13) becomes set when OPL IN and SEL IN are active at the same time, or when any two of the signals ADRS IN, STAT IN, or SRV IN/DAT IN are active at the same time. It also is set for a channel error; that is, when the BDC processor logic and/or software via the TOR activate CMD OUT and SRV OUT or DAT OUT at the same time. The Cntl Chk FF is cleared by the CHANNEL CLEAR (CH CLR) signal which is present any time the BDC processor logic is not active: OPL OUT inactive and not in a diagnostic state. This CH CLR signal clears all channel logic control flip-flops, channel status flip-flops, and the HCX, -HCX2 and Dat Chng FF's in the CCR.
- Transmission Check: The Xmn Chk FF, when set, indicates a parity error on the byte of information received from an external control unit. The Xmn Chk FF (page 1C13 Loc) is clocked each time the BUS IN signals are gated into the BIR1, if the channel is not communicating with an integrated adapter. The CHPAR ERR signal controlled by the parity check element (page 1C14) determines if the Xmn Chk FF will set when clocked.

  Any time one of the integrated adapters is communicating with the channel logic it activates the ADAPTER SELECTED signal. This signal disables the clocking gate to the Xmn Chk FF, thus not allowing parity error indications on information transferred from integrated adapters.

  The Xmn Chk FF is precleared when an interface disconnect sequence is performed, or by CMD OUT active, by SRV OUT active, or by the CH CLR signal.
- Illegal Length: This flip-flop, when set, indicates that the CMD OUT signal was used, in response to either SRV IN or DAT IN, to terminate the data transfer. CMD OUT is used to indicate *stop* to the control unit or adapter when the channel logic has determined all the data has been transferred (byte count=0). The Illegal Length Channel Status flip-flop and the byte count in the

Byte Count register indicates the type of data transfer termination as listed below:

Illegal Length Set = Processor terminated the data transfer before device was ready to.

Illegal Length Clear and byte count zero = Both the processor logic and device terminated data transfer at same time.

Illegal Length Clear and byte count not zero = Device terminated data transfer before processor logic was ready to.

- Assembly/Disassembly Mode: The state of this flip-flop indicates whether the processor logic is in the one-byte-wide or two-byte-wide transfer mode: when set, it indicates two-byte; when clear, it indicates one-byte.
- Mark Zero In: This Tag In Line is activated by an external control unit requesting command retry via the MKO IN signal.
- Assembly/Disassembly Option Present: This channel status line is active when the ASMDASOP (EU module) is present.

Table 4-5. CCR and Status Bit Assignment (ERI)

Source	ERI Bit Position	Title	Abbreviation
CCR 0	0	Hardware Coantrolled Transfer	HCX
1	1	Input	INPUT
2	2	Diagnostic	DIAG
3	3	Data Chaining	DAT CHNG
Channel Logic Status Flip-Flops	4	Control Check	CNTL CHK
	5	Transmission Check	XMN CHK
	6	Illegal Length	ILL LGTH
CCR7 (EU Module)	7	Assembly/Disassembly Mode	ASMDASMD
External Control Units	8	Mark Zero In (Tag In Line)	MKO IN
	9	Not Used	
	10		
	11		
	12		
	13		
	14		
EU Module	15	Assembly/Disassembly Option Present	ASMDASOP

Byte Count Register (BCR)

The BCR consists of four 4-bit counter MSI elements connected together to form a 16-bit *up* counter. The complement of the byte count minus one on the ER0 lines is loaded into the BCR during a *Store-Register* micro-instruction (Figure 4-8) by the LOAD BCR (LD BCR) signal. Only during a *Store-Register* micro-instruction with ERNG3-00 high will the Load Byte Count (LD BYT CNT) flip-flop be set and clear, thus generating the LD BCR pulse (page 1C14).

The BCR counts up by one each time a data byte is transferred between the channel logic and a control unit or an adapter during a hardware controlled data transfer sequence. The COUNT UP signal (SRVO+DATO) into the least-significant bit of the BCR clears each time either SRV OUT or DAT OUT goes active. On the trailing edge of SRV OUT or DAT OUT the BCR counts up by one. The CARRY OUT signal from each MSI element feeds the count up input to the next higher-order element.

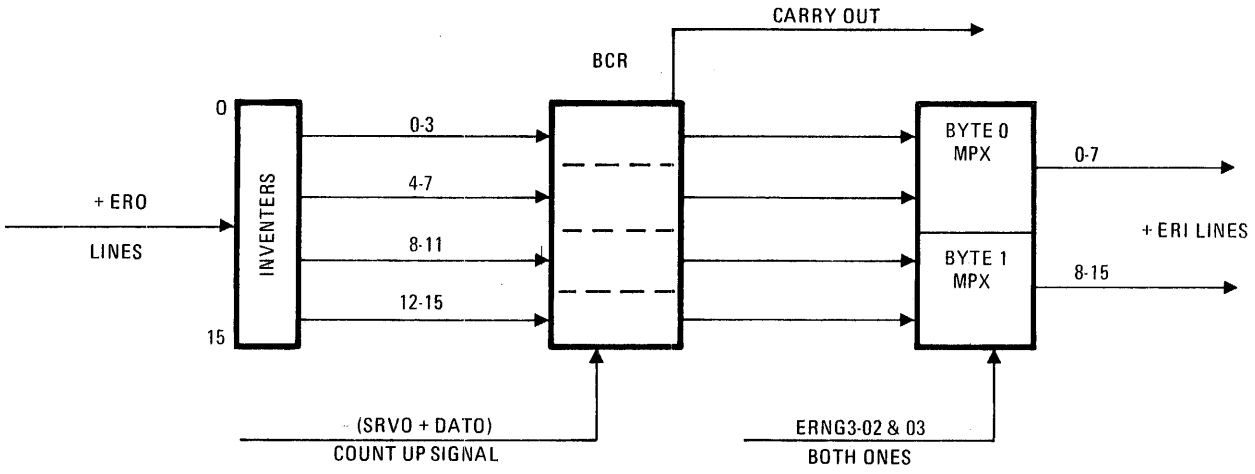


Figure 4-8. Byte Count Register Configuration

When all 16 bits in the counter are set, the next count-up pulse causes a carry out from the most-significant bit position of the counter. For this reason, the byte count minus one is sent to the channel logic to be loaded into the BCR, as shown in the following example.

	0025_{16}	Number of bytes to transfer
	0024_{16}	Byte count minus 1 sent to basic data channel control logic
	$FFDB_{16}$	Value loaded into BCR
	$FFFF_{16}$	Value in BCR after 24_{16} bytes transferred
carry out	0000_{16}	Value in BCR after 25_{16} bytes transferred

The CARRY OUT signal clears the Input FF in the CCR, clocks the Cmd Out Cntl FF and clears the ASMDASMD FF on the optional Assembly/Disassembly (EU) module.

The contents of the Byte Count register may be selected via both Byte Zero and Byte One Multiplexers and routed to ERI lines. This occurs during a *Load Register* micro-instruction with ERNG3-02 and 03 set.

Channel Signals

The nine Tag In lines (TILs) (page 1C13) transfer identification, interlock, and control information from the adapters and control units. The TILs (except that for SEL IN) connect directly to integrated adapters and to the external control units via the external interface module receivers (Figure 4-9). The SEL IN tag line is connected to the channel select-in bypass logic (Figure 4-3). The BOR output is routed through one stage of inversion before being collector OR'd to the TIL. The BOR input is used during diagnostic operations.

The TILs connect to the control logic and to both multiplexers. Six of the TAG IN signals direct, through the control logic, specific operations and/or events. Eight TILs are selected through the Byte Zero multiplexer to the ERI lines 0 to 7, when ERNG3-02 is low and ERNG3-03 is high. The ninth TIL (MARK ZERO IN) is selected through the Byte One Multiplexer when the CCR is read (see Channel Control Register). The bit assignment is listed in Table 4-6, immediately followed by a signal description.

Table 4-6. Tag In Signal Usage

Bit Position	Title	Abbreviation	Function
0	Address In	ADRS IN	Bus In ID
1	Status In	STAT IN	Bus In ID
2	Service In	SRV IN	Bus In ID
3	Operational In	OPL IN	Interlock
4	Select In	SEL IN	Scan Control
5	Request In	REQ IN	Scan Control
6	Data In	DAT IN	Bus In ID
7	Disconnect In	DIS IN	Special Control
8	Mark Zero In	MKO IN	Special Control

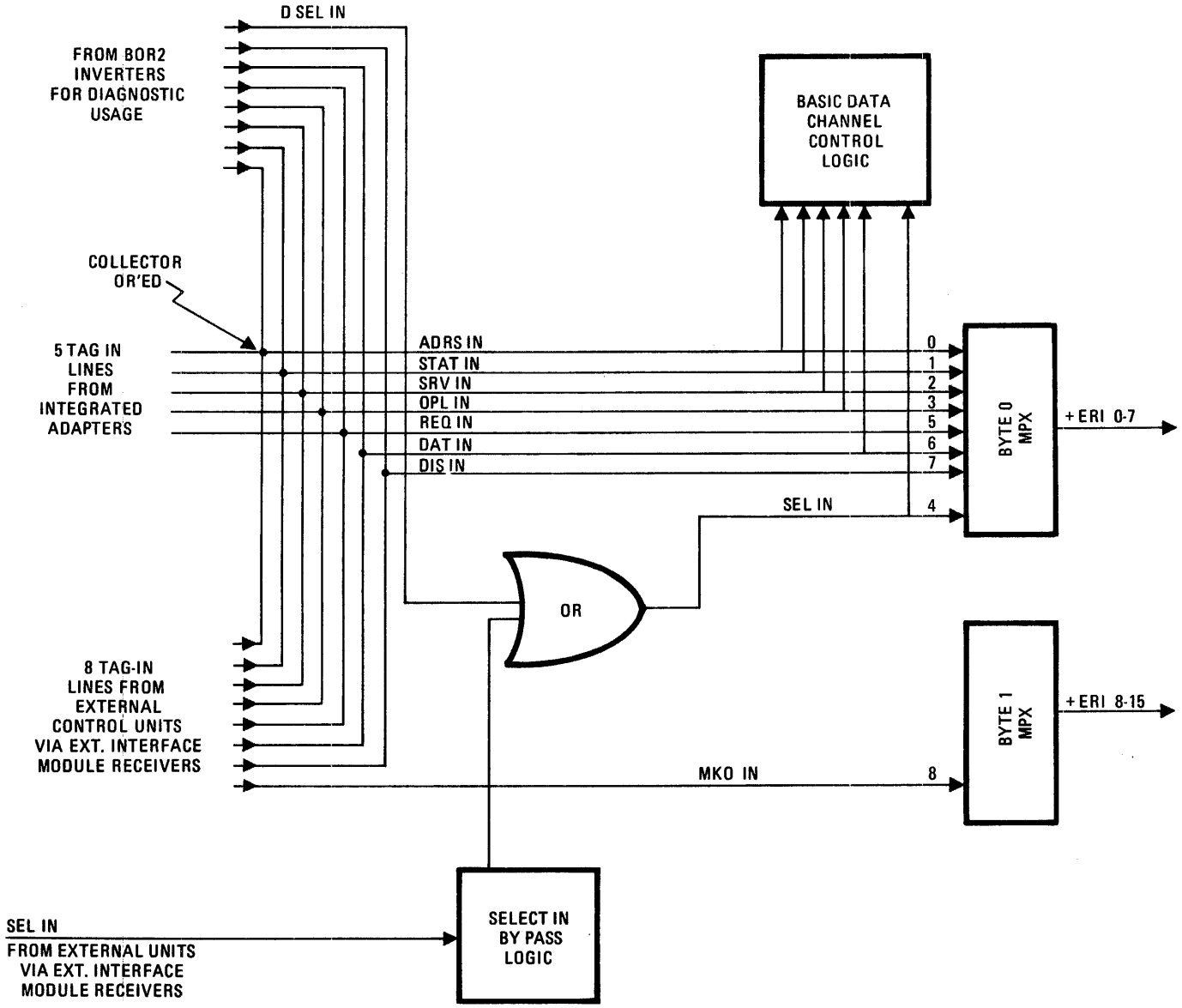


Figure 4-9. Tag In Line Configuration

- Address In: This signal goes active when the selected adapter or control unit has placed its address on the Bus In lines.
- Status In: This signal goes active when the selected adapter or control unit places a Status Byte on the Bus In lines.
- Service In: The selected adapter or control unit activates this signal when it wants to transmit or receive a byte of information.
- Operational In: This signal goes active in response to SEL OUT when an adapter or control unit becomes selected either during an adapter/control unit- or processor-initiated selection sequence and remains active for the duration of the selection.
- Select In: This line provides a return path for the SEL OUT signal via the serial selection loop. This signal goes active during an initial selection sequence indicating the adapter or control unit addressed by the control logic did not become selected.
- Request In: This signal goes active when an adapter(s) or control unit(s) is ready to present data or status information to the processor logic regardless of the immediate sequence being executed by the processor. REQ IN requests a selection sequence to be performed, and remains active until the adapter or control unit activates OPL IN during the selection sequence.
- Data In: The selected external control unit activates this signal when it wants to transmit or receive a byte of information.
- Disconnect In: This signal provides the external control units with the ability to alert the system to malfunctions, which cannot be properly signaled using the other common interface signals.
- Mark Zero In (MKO IN): This signal is activated by an external control unit when a condition requesting command retry is encountered. This signal is enabled through the Byte One multiplexer when the CCR and channel status information is read.

In addition to the TAG IN signals already discussed, the basic data channel control logic deals with other channel signals directly (Figure 4-5). A description of these signals follows.

- Clock Out (CLK OUT): This signal is used in conjunction with the METER ENABLE/DISABLE switch located on the same external control unit or device. CLOCK OUT ensures that the control unit will not change states, regardless of when the switch position changed, until the CLK OUT is inactive and the following conditions prevail:
  - No active communication on the basic data channel that will affect this unit.
  - No pending status information on this unit.
  - No command chaining indicated for this unit.

  The channel control logic (page 1C1302) generates CLK OUT whenever either SEL OUT or OPL IN is active. Pressing the System Control Panel MAINTENANCE MODE pushbutton places the computer in maintenance mode, preventing SEL OUT from activating CLK OUT.

- Metering Out (MTR OUT): This signal is used to condition running-time meters in external control units. The control logic (page 1C1302) also generates MTR OUT whenever SEL OUT is active or whenever the MTR IN signal is active. Maintenance mode disables the SEL OUT signal input to the MTR OUT logic but does not disable the MTR IN input.
- Metering In (MTR IN): This signal is normally activated by an external control unit from the acceptance of a command until the generation of *Device End* for that command. For I/O device metering definitions refer to the pertinent I/O device publication. The control logic simply uses MTR IN to generate MTR OUT, thus conditioning the running-time meter in the external control unit.

The CLOCK OUT, METERING IN and METERING OUT signals are not used by the integrated adapters. METERING IN and METERING OUT are not required, because the adapters do not contain running-time meters. CLOCK OUT is not required to condition state changes in the adapters.

- Adapter Selected: This signal is activated by an integrated adapter while communicating with the basic data channel. When active, this signal disables parity error checking (see Transmission Check).
- External Interface Option (EXTINFOP): This signal, when active, indicates the presence of the external interface printed circuit module in location 1C02. The EXTINFOP signal is used by the select-loop bypass logic (page 1C1302 Loc) (see Select Out).

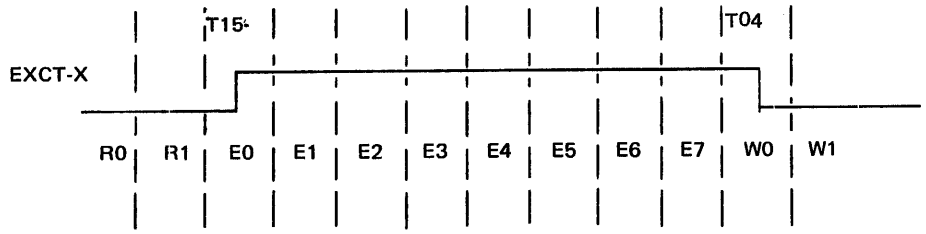
- Card Reader Option (CR OPT): Activated when the ICRA modules are present.
- Card Reader Punch Option (CRP OPT): Activated when the IRPA modules are present.

NOTE

The above option signals are used by the select-loop bypass logic (page 1C13).

In addition to the Extended register data paths and Extended register address, the BDC control logic utilizes several discrete signals in communicating with the instruction execution and control circuitry (Figure 4-5). The following description of these signals, when applicable, includes the timing relationship of each to a processor state's major and minor cycles (Figure 4-10).

- Clock (CLOCK-XX): The five clock signals received from the shared resources timing logic are used by the BDC control logic to generate internal timing signals which allow it to function concurrently with processor state minor cycles. Each clock signal lags the previous clock by 20 nanoseconds, is 30 nanoseconds in duration and repeats every minor cycle.
- Execute (EXCT-X): This signal is active while the processor state is in the execute (E) portion of a time slice. The channel control logic uses the EXCT-X in both the active and inactive state (X) being either a 1 or a 2. EXCT-X active allows the clocking of control registers and flip-flops. In the inactive state, EXCT-X enables the setting of the End-of-Transfer FF.



- Master Clear (MC-I/O): This signal is activated for a power-on clear, a master clear from the System Control Panel SYSTEM RESET switch, an autoload operation, or a reset/load operation. The MC-I/O signal in the processor logic terminates all the TAG OUT signals, clears the Diag FF, and triggers a one-shot circuit which after nine microseconds brings the OPL OUT signal back to the activated state. This one-shot logic, therefore, does not require the processor to activate OPL OUT via the TOR after a master clear.
- Input-Output Exit (I/O EXIT): This signal is activated by a CIO1 (Compare I/O) and the B-Register value compare or when the control logic generates an End of Transfer. The processor control logic ANDs the I/O EXIT signal with EXCT-X to enable the setting of the I/O Exit FF at Clock-80 time (page 1C14). The flip-flop is cleared by CLOCK-40. The resulting I/O EXIT pulse is used to clear the HCX, BIR Cntl and HCX Req FF's.
- Extended Register File Group Three Write (ERFG3WR): This signal is activated by the processor state during the execution of a *Store Register* micro-instruction. The ERFG3WR signal is ANDed with EXCT-2 to enable the setting of the Write Register (WR REG) FF at Clock-60 time (page 1C14). The WR Reg FF is cleared at Clock-00 time, thus generating a WR REG pulse which is used to clock the CCR, TOR, BOR, and BOR 1 Full FFs.

  When the BOR is the register being clocked (ERNGG3-01 high) the ERFG3WR signal also clears the HCX Req FF.
- Extended Register File Group Three Read (ERFG3RD): This signal is activated by the processor state during the execution of a *Load Register* micro-instruction. The ERFG3RD is used to clear the HCX Req FF when ERNG3-02 and 03 are both clear. This is true only when the BIR is being read.
- Request (REQ-X): This signal is used by the channel control logic (page 1C13) to request a time slice for the processor. The REQ-X signal going active immediately sets the processor's Busy FF. If the Busy FF is already set, the REQ-X signal will not set the Busy FF until after the processor has cleared it, providing the REQ-X signal is still active. Once the Busy FF is set, the processor state will be given a time slice dependent upon the existing priority levels for processor states zero to three in the control register, and the number of active processors.

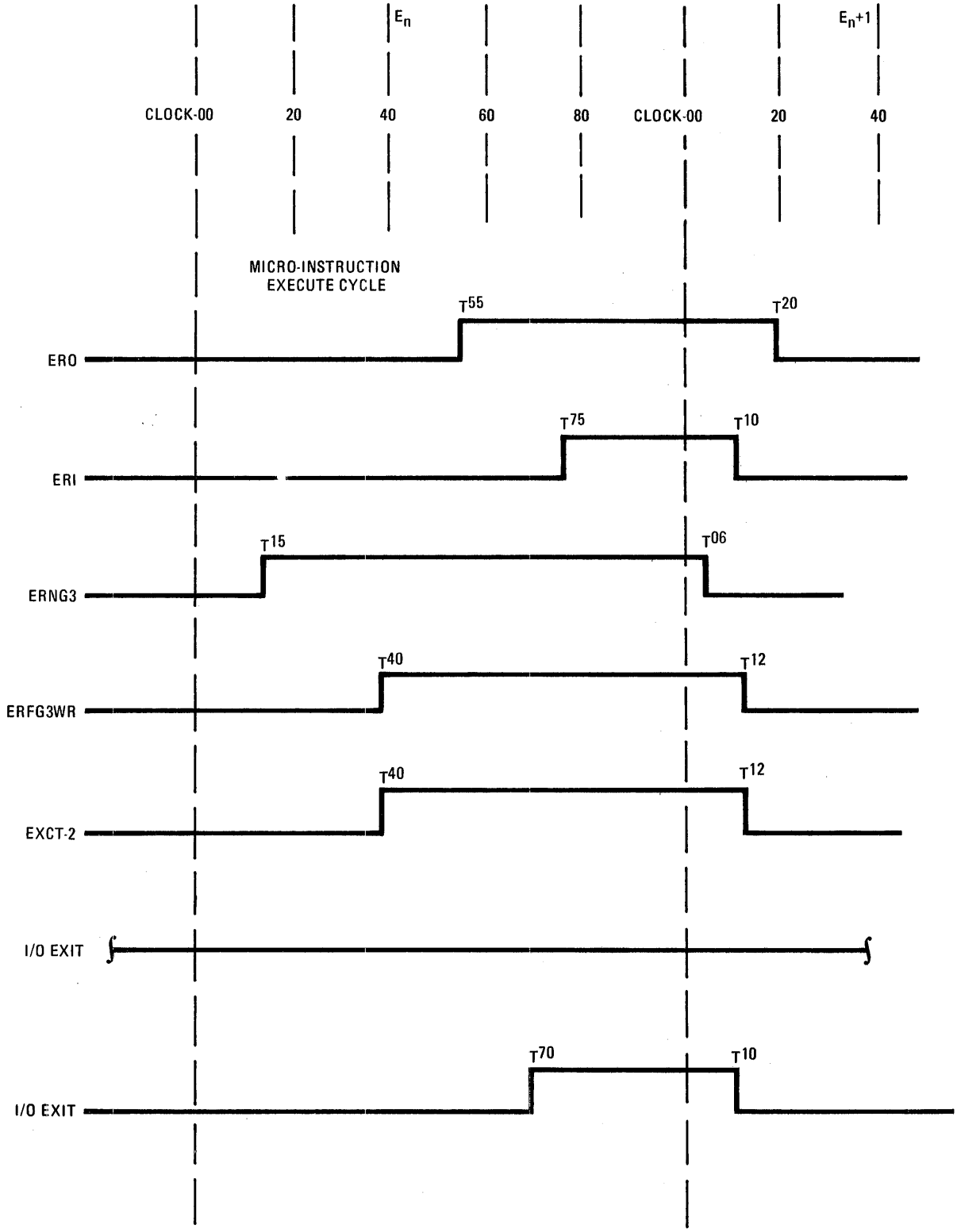


Figure 4-10. Timing Relationship, Minor Cycle to Signal

REQ-X is generated by the BDC request control logic (page 1C13 Loc) in response to Tag In line (TIL) signals, conditioned by bits of the CCR and control flip-flops. The request control logic generates two REQ-X signals, one on the leading edge and one on the trailing edge of a TIL signal, during all sequences but the hardware-controlled data transfers. During a hardware-controlled data transfer the REQ-X signal is generated by the hardware-controlled transfer logic (see Data Transfer operations).

Either the OPL IN or SEL IN signal active generates a REQ-X through the CNTL REQ logic. ADRS IN, STAT IN, SRV IN or DAT IN generate a REQ-X through the Tag Req logic, if enabled by the CCR and Channel Control flip-flops. The REQ-X signal requests a time slice for the processor, indicating it is to react to the TAG IN signal. The adapter or control unit and the request control logic are now interlocked waiting for the processor's response.

The processor responds by activating the proper TAG OUT signal, writing in the TOR with a *Store-Register* micro-instruction. The WR REG pulse produced by processor logic during the *Store* micro-instruction ANDs with ERNG3-03 (TOR designated) to clock the edge-triggered flip-flops in the Request Control logic. Depending upon the active TAG IN signal, the proper edge-triggered flip-flop will set, disabling the REQ-X signal through the exclusive OR logic. Now the Request Control logic and the processor are interlocked, waiting for the adapter or control unit to respond.

In response to the change of the TAG OUT signal, the adapter or control unit terminates the active TAG IN signal. This generates another REQ-X through the exclusive OR. This REQ-X signal indicates the device has responded to the Tag Out change and it is again the processor's turn to react. Not until the processor reacts and executes a *Store-Register* micro-instruction to the TOR will the edge-triggered flip-flops be again clocked. When clocked, the proper flip-flop will clear, since the TAG IN signal is inactive, disabling the REQ-X through the exclusive OR.

- Attention (ATTN-2): This signal is activated by REQ IN going active only if *Opl Out* bit in the TOR is active or the *Diag* bit in the CCR is set. The ATTN-X signal sets the processor state Busy FF if its Active FF is not set. The Active FF not set indicates the processor is not engaged in controlling a Basic Data Channel operation.

- Priority (PRI-X): This signal is activated when the processor logic determines the loss of data is imminent. This condition occurs only during hardware-controlled data transfers when the processor state does not get a time slice often enough to keep up with the I/O device. This is possible due to the fact that during a hardware-controlled data transfer, the logic (not processor state program) controls the SRV OUT and DAT OUT signals to the adapter and control units.

  The PRI-X signal, if enabled by the control register, sets the Over-Ride Resync FF in the shared resources logic (see Priority Network under Time Slicing in Volume 1, General Description).

- End of Transfer (EOT): This signal is activated by the channel logic during a hardware-controlled data transfer. The EOT FF is set when the adapter or control unit terminates the data transfer by activating STAT IN in response to SRV OUT. The active EOT signal forces the Aμ-equal-to-Bμ condition during the execution of the next *CIO1* micro-instruction by the processor. The Aμ=Bμ condition generates the I/O EXIT signal to the processor (see I/O Exit).

OPERATIONAL SEQUENCES

Many different operational sequences can be performed by way of the Basic Data Channel processors, some initiated by the processor, and some adapter or control unit-initiated. All the operational sequences will be discussed in conjunction with the hardware/software interlace tables (Tables 4-7 through 4-12) and Figure 4-11. References to the logic diagrams will be made when necessary to clarify the discussion.

The hardware/software interlace tables present three important features:

1) The Basic Data Channel processor operation and reaction, and the common signal line activation in relationship to processor micro-instruction execution.
2) The Basic Data Channel processor operation and reaction, and processor hardware operation and reaction to common signal line activation by the adapters or control units.
3) The required order of signal generations and time delay duration.

NOTE

Processor state 2 is used in the following descriptions for clarity of signal names. All sequences function the same for processor state 1.

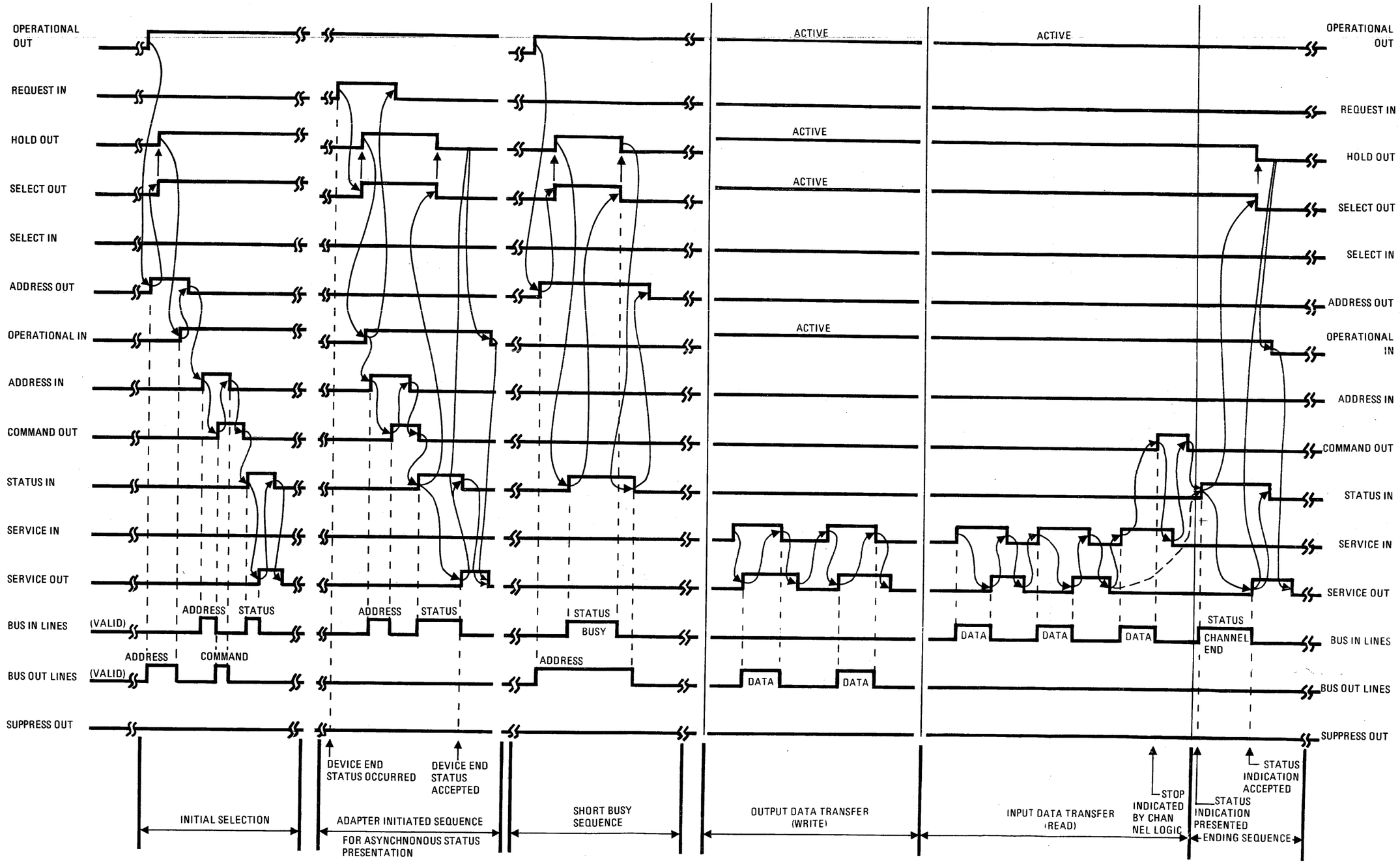


Figure 4-11. BDC Operational Sequences

Initial Selection

Processor state 2 starts an input/output operation by first selecting the I/O device with an initial selection sequence as listed in Table 4-7. Processor 2 starts the selection by activating the CHANNEL ENABLE signal if it is not already active, placing the device address on the Bus Out lines along with activating the proper Tag Out lines in the correct order. When the addressed adapter or control unit receives the SEL OUT signal, it blocks its propagation along the selection loop and indicates selection by activating the OPL IN signal. And, if an integrated adapter is selected, it also activates the ADAPTER SELECTED signal. In response to the OPL IN signal, the processor logic requests a time slice by setting its Busy FF. See the discussion under Request (REQ-2).

Processor 2 responds by reading the TIL's and determining which TIL caused the REQ-2. This is necessary because OPL IN is not the only response possible to SEL OUT. If the addressed adapter or device is not present, not powered up, or not on line, SEL IN will be the response causing the REQ-2.

Table 4-7. Initial Selection

Micro-Instruction and Hardware Functions	Basic Data Channel Logic Operation	Common Signal Lines	Adapter or Control Unit Action
G3WR→TOR	OPL OUT →		→ (channel enable)
G3WR→BOR	ERO→BOR1→BOR2→ BUS OUT →	→ Address →	
250 ns wait			
G3WR→TOR	ADRS OUT →		→ (decode address)
400 ns wait			
G3WR→TOR	SEL OUT →		→
	generate HLD OUT →		→
Set Busy 2 FF←	← Generate REQ-2 ←		← OPL IN (device selected)
G3RD←TIL		←	← ADAPTER SELECTED
(TIL determination routine)			(if adapter)
G3WR→TOR	Clear REQ-2 → End ADRS OUT →		→
	BIR2←BIR1← BUS IN ←	← Address ←	
	Gate BUS IN →		→ ADRS IN
Set Busy 2 FF←	← Generate REQ-2		
G3RD←TIL			
(TIL determination routine)			
G3RD←CCR			
G3RD←BIR			
(address compare and transfer check routine)			
G3WR→BOR	ERO→BOR1→BOR2→ BUS OUT →	→ Command →	→ (direct device operation)
G3WR→TOR	Clear REQ-2 → CMD OUT →		→ (decode command)
Set Busy 2 FF←	← generate REQ-2 ←		← End ADRS IN
G3RD←TIL			
(TIL determination routine)			
G3WR→TOR	Clear REQ-2 → End CMD OUT →		→
	BIR2←BIR1← BUS IN ←	← Status ←	
	Gate BUS IN ←		← STAT IN
Set Busy 2 FF←	← Generate REQ-2		
G3RD←TIL			
(TIL determination routine)			
G3RD←CCR			
G3RD←BIR			
(status check and transfer check routine)			
G3WR→TOR	Clear REQ-2 → SRV OUT →		→
Set Busy 2 FF←	← Generate REQ-2 ←		← End STAT IN
G3WR→TOR	Clear REQ-2 → End SRV OUT →		→

Likewise, if the addressed adapter or control unit is busy, it may respond with STAT IN rather than OPL IN. The discussion under Short-Busy Sequence, has further details.

Once processor 2 has determined that OPL IN was the response, it continues with the selection sequence by deactivating the ADRS OUT signal. The selected device reacts by placing its address on the Bus In lines and activating the ADRS IN signal. The BDC logic responds to the TAG IN signal by generating a REQ 2; then it checks parity on the bus-in information while gating it into BIR1 and BIR2, and clocks the Xmn Chk FF if the unit selected is not an integrated adapter (ADIS PAR signal inactive).

Processor 2 responds by reading the TIL's to determine that ADRS IN is active. If ADRS IN is active the address in the BIR and the status information in the CCR are next read in by processor Execution and Control Circuitry. The address read is compared with the address that was sent to make certain the right device was selected. The CCR status information is checked.

Once it is determined the right device is selected, the type of operation is initiated by placing the Command Byte in the BOR and activating the CMD OUT signal. The selected adapter or control unit accepts the Command Byte and indicates this fact by terminating the ADRS IN signal. The processor logic recognizes the ADRS IN signal termination and generates a REQ-2 signal, again requesting a time slice.

Processor 2 reacts by simply deactivating the CMD OUT signal. The adapter or control unit responds by placing its status information on the Bus In lines and signals this fact with the STAT IN signal.

BDC logic, in response to STAT IN, checks parity on the Bus In lines, generates a REQ-2 signal to the Execution and Control Circuitry and generates a STRB BIR1 signal. The STRB BIR1 signal gates the bus-in information into the BIR1 and BIR2 and clocks the Xmn Chk FF if the ADAPTER SELECTED signal is inactive.

Processor Execution and Control responds by reading the TIL's to determine that STAT IN generated the REQ-2. If STAT IN is active, the processor next reads the CCR status information and the Status Byte from the BIR and performs a status check. The status information received will depend on the operation specified by the Command Byte. All data transfer commands such as *Read, Write, Read Backward* and *Sense* normally cause a zero status return. A non-zero status indication is returned for immediate control commands, the *Test I/O* command and for error conditions. A zero status indicates that a data transfer sequence will follow the initial selection for all commands except the *Test I/O* command.

If the status indication is incorrect, the processor will deselect the adapter or control unit at the end of the initial selection, and no data transfer will take place. This is done by terminating SEL OUT and HLD OUT before activating SRV OUT.

Once the Status Byte is accepted and the channel control status is checked, SRV OUT is activated. The control unit or adapter responds by terminating the STAT IN signal, which causes the BDC logic to generate another REQ-2 signal. The Execution and Control Circuitry, in response, terminates the SRV OUT signal and the initial selection sequence is complete.

Adapter or Control Unit-Initiated Sequence

An adapter or control unit can initiate a sequence whenever it requires service and the channel is enabled, by simply activating its REQ IN signal. The control logic forwards the REQ IN signal to the processor directly (page 1C13) and in doing so changes the signal's name to ATTN-2.

ATTN-2 sets the processor 2 Busy FF, if the processor 2 Active FF is clear or when it becomes clear. The Active FF, cleared, says processor 2 is not engaged in a Basic Data Channel operation; thus the request from the adapter or control unit can be recognized.

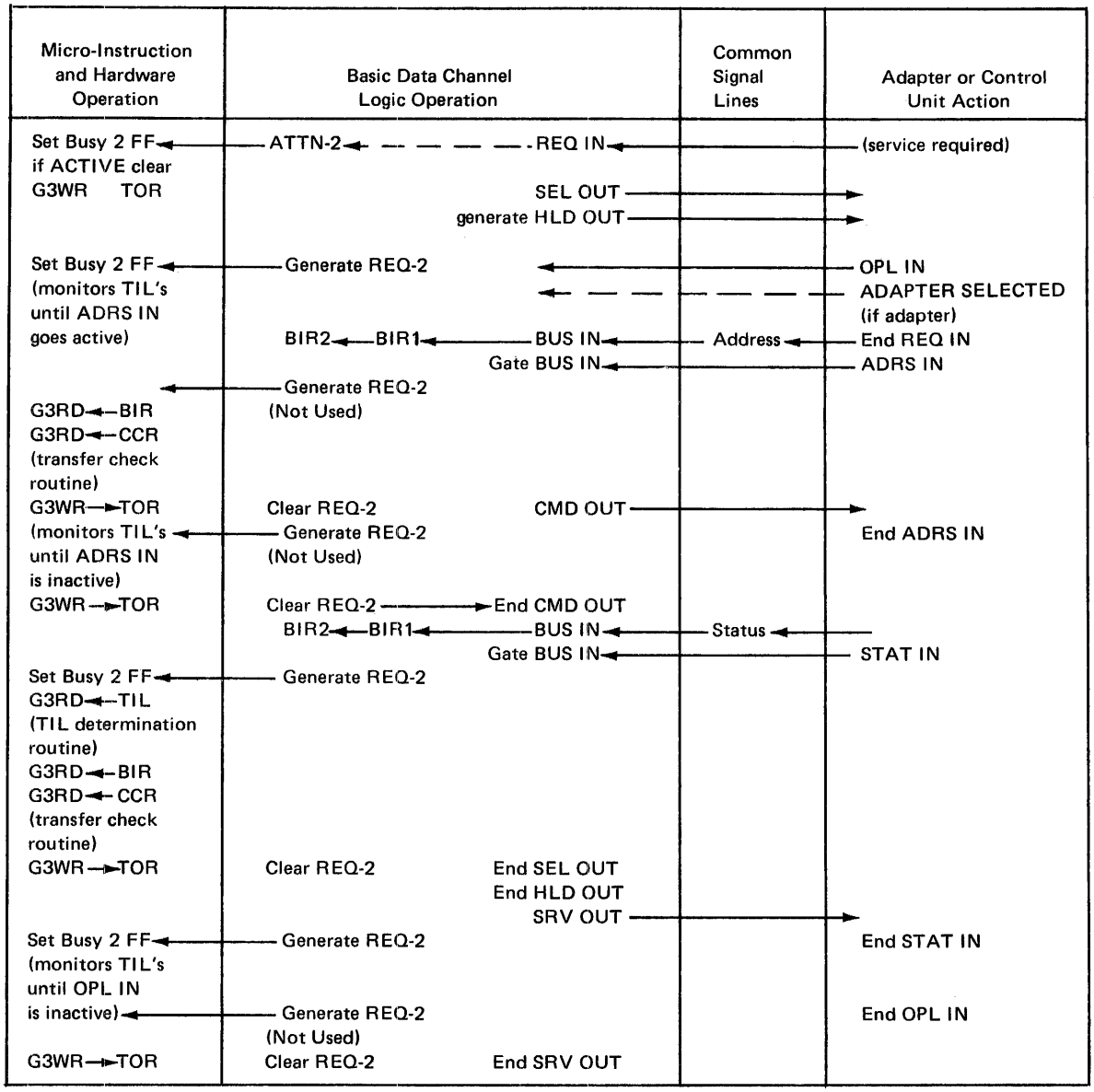
Processor 2 must perform a polling routine to determine which adapter or control unit is requesting service. This is done by activating only the SEL OUT and HLD OUT signals and *not* the ADRS OUT signal (Table 4-8). The SEL OUT signal arrives at the control units in the order of priority. If a control unit is not the one requesting service it passes the SEL OUT signal to the next lower-priority unit in the selection loop. (Channel Configuration.) When the first unit in the selection loop requesting service receives SEL OUT it stops the signal from further propagation. It also activates the OPL IN and ADIS ADAPTER SELECTED (if it is an integrated adapter) and terminates its REQ IN. the OPL IN signal causes the control logic to generate a REQ-2 to processor 2 requesting a time slice.

NOTE

If other units are also requesting service at this time, the REQ IN signal will stay active until all have been serviced.

Next, the adapter or control unit places the device address on the Bus In lines and activates the ADRS IN signal. In response to ADRS IN, the control logic checks parity on the Bus In lines and generates the STRB BIR1 signal which gates the BUS IN signals into the BIR1 and BIR2, clocks the Xmn Chk FF if ADAPTER SELECTED is inactive, and generates another REQ-2 (which is not used by processor 2).

Table 4-8. Adapter Initiated



Processor 2, in response to the first REQ-IN signal, becomes active and reads in and monitors the Tag In lines.

Once ADRS IN goes active, processor 2 reads in the device address from BIR2 and CCR status, and performs an address determination routine. The Xmn Chk FF is checked for a parity error indication. Once the device address has been recognized, processor 2 indicates to the adapter or control unit to proceed by activating CMD OUT, and monitors the TIL's until ADRS IN is terminated.

The adapter or control unit responds to CMD OUT by terminating ADRS IN, causing the generation of a REQ-2 signal by the control logic (although it is not used by processor 2). The termination of ADRS IN causes processor 2 to end CMD OUT.

The adapter or control unit responds to this by placing its Status Byte on the Bus In lines, and activating STAT IN. The Status Byte indicates the reason the adapter or control unit initiated the sequence with REQ IN.

The control logic, in response to STAT IN, checks Bus In line parity, generates a REQ-2, generates a STRB BIR1 pulse which gates the BUS IN signals into BIR1 and BIR2, and clocks the Xmn Chk FF if ADAPTER SELECTED is inactive.

Processor 2, in response to the REQ-2 signal, reads in TIL's and the Status Byte residing in the BIR2, and the channel status information in the CCR. The TIL's are checked for correct response and the Xmn Chk FF is checked for a parity error in the Status Byte. Next, the SEL OUT and HLD OUT are terminated and SRV OUT is activated, indicating the Status Byte has been accepted.

NOTE

The status information from the polling routine is used to direct the processor 2 program.

The adapter or control unit responds by terminating STAT IN, which causes the control logic to generate a REQ-2 signal to Execution and Control Circuitry. The adapter also terminates OPL IN after making sure HLD OUT is inactive.

Processor 2, in response to the REQ-2, monitors the TIL until OPL IN terminates and then ends SRV OUT, thus completing the adapter-initiated sequence and polling routine.

Short Busy Sequence

During initial selection, an adapter or control unit can indicate a busy condition in one of two ways, depending on the unit's design.

1) The busy adapter or control unit presents *busy* status information during an initial selection sequence in response to CMD OUT going inactive.

2) The busy adapter or control unit can force a short-busy sequence by responding to SEL OUT with STAT IN rather than OPL IN during the initial selection.

The short-busy sequence begins as a normal initial selection with processor 2 enabling the channel, placing the device address on the Bus Out lines and activating SEL OUT and ADRS OUT (Table 4-9). The processor logic activates HLD OUT. When SEL OUT is received by the addressed adapter or control unit, it is blocked from further propagation. OPL IN is not activated at this time by the adapter or control unit. The adapter or control unit places the *Busy* Status Byte on the Bus In lines and indicates it is status, rather than address information, by activating the STAT IN signal, not ADRS IN.

The control logic responds to the STAT IN signal by gating the Status Byte into the BIR1 and BIR2 with the STRB BIR1 signal, by clocking the Xmn Chk FF if ADAPTER SELECTED is inactive, and by requesting a processor 2 time slice with the REQ-2 signal.

Table 4-9. Short Busy

Micro-Instruction and Hardware Operation	Basic Data Channel Logic Operation	Common Signal Lines	Adapter or Control Unit Action
G3WR→TOR	OPL OUT →		→ (channel enable)
G3WR→BOR	BOR1→BOR2→ BUS OUT →	→ Address →	→
250 ns wait			
G3WR→TOR	ADRS OUT →		→ (decode address)
400 ns wait			
G3WR→TOR	SEL OUT →		→
	Generate HLD OUT →		→
	BIR2←BIR1← BUS IN ←	← Status ←	←
	Gate BUS IN ←		← STAT IN (unit busy)
Set Busy 2 FF ←	← Generate REQ-2	← - - - -	← ADAPTER SELECTED
G3RD←TIL			(if adapter)
(TIL determination routine)			
G3RD← BIR			
G3RD←CCR			
(transfer check routine)			
G3WR →TOR	Clear REQ-2 End SEL OUT		
Monitor TIL's	End HLD OUT		
until STAT IN ←	← Generate REQ-2		End STAT IN
goes inactive	(Not Used		
G3WR→TOR	End ADRS OUT		

Processor 2 reads the TIL's to determine which TIL caused the REQ-2 signal. Once STAT IN is determined to be the active TIL, the status information in the BIR and CCR status information is read in. Processor 2 signals the adapter or control unit that it has read the status information by deactivating the SEL OUT signal, and continues by monitoring the TIL's. The control logic deactivates HLD OUT as a result of SEL OUT termination.

The adapter or control unit responds by deactivating the STAT IN signal, and thus has completed its portion of the short-busy sequence.

The control logic generates another REQ-2 in response to the STAT IN signal termination. Processor 2 terminates the ADRS OUT signal as a result of the STAT IN signal going inactive, thus completing the short-busy sequence.

Data Transfer Sequence

After a channel-initiated selection sequence has been completed, the control unit or adapter remains selected by processor 2, for the duration of information transfer. The information can be sent in response to a *Sense* command, control information in response to a *Control* Command or data in response to *reads* and *writes*. The number of data bytes is determined by the device record length or the byte count.

Data transfer sequences may be entirely under processor 2 machine-language program control; or under both Basic Data Channel hardware and micro-instruction control, referred to as a hardware-controlled data transfer (HCX transfer). A data transfer sequence may be terminated by either the processor logic or the adapter or control unit.

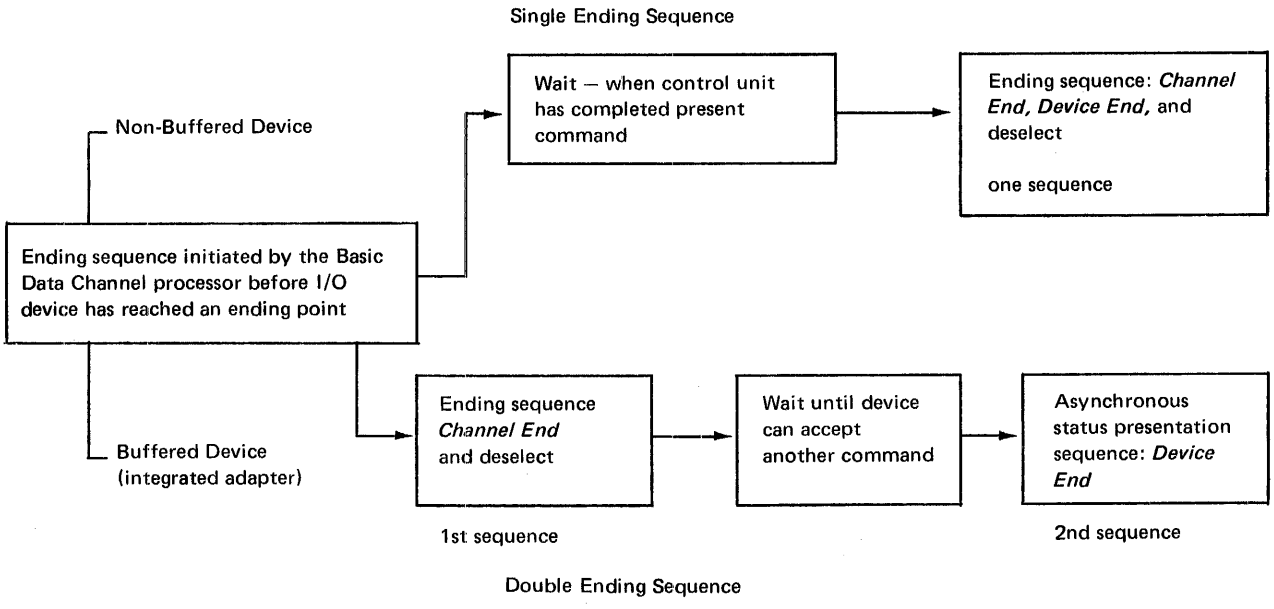
Once the selection sequence has been performed, the control unit or the adapter starts an input data transfer by placing a Data Byte on the Bus In lines and activating SRV IN. The Bus In lines and SRV IN are maintained until the channel logic responds with SRV OUT, which causes the termination of SRV IN. The channel logic responds to SRV IN termination by terminating SRV OUT. This short routine is repeated for each Data Byte transfer (Figure 4-11).

After a selection sequence, the control unit or adapter also may start an output data transfer by activating SRV IN. The first Data Byte is placed on the Bus Out lines, and SRV OUT is activated in response to SRV IN. The SRV OUT signal and the Bus Out lines are maintained until the control unit responds by terminating SRV IN. The processor responds by terminating SRV OUT. Again this routine must be repeated for each Data Byte transfer.

Ending Sequences

An ending sequence terminates a data transfer operation by first presenting device and adapter or control unit status information and then *deselecting* the device from the Basic Data Channel. Either processor 2, through the channel control logic, or an adapter may initiate an ending sequence.

If an ending sequence is initiated by the channel control logic, one or two sequence of events will be performed. The sequence(s) that occur depend on the I/O device, as shown below.



Processor-Initiated

During a data transfer, the BDC processor stops the transfer and signals the beginning of an ending sequence by indicating *stop* to the adapter or control unit. *Stop* is indicated by a response of CMD OUT, instead of SRV OUT, to the SRV IN or DAT IN signal (Table 4-10 and Figure 4-11). The adapter or control unit, in response to CMD OUT, terminates SRV IN because data is no longer to be transferred – note: byte is not transferred. The processor reacts to the ending of the SRV IN by terminating CMD OUT.

Both the adapter or control unit and processor 2 now have registered the fact that the data transfer has stopped and the ending sequence is to follow.

Double Ending Sequence – All integrated adapters, and some control units start the ending sequence immediately by placing status information on the Bus In lines and activating STAT IN. The control logic in response to STAT IN activates the REQ-2 signal, and after a 220-nanosecond delay triggers the Strb BIR1 one-shot circuit. The one-shot output gates the Bus In information into the BIR1 and into BIR2 after μ1 sets -HCX2.

Processor 2 reads in the TIL's to determine that STAT IN caused the REQ-2. Once it is determined STAT IN is active, the Status Byte from the BIR, and the CCR status are read. The Status Byte contains *Channel End* bit because data transfer has stopped. Other status bits may be present, depending on adapter and device conditions. (Status information under Channel Requirements has further details.) Processor 2 now indicates to the adapter that the status information has been accepted, not stacked, by activating SRV OUT. Processor 2 also starts the *deselection* by ending SEL OUT and HLD OUT.

The adapter, in reaction to SRV OUT going active, first makes sure HOLD OUT is inactive and then terminates STAT IN and shortly thereafter OPL IN.

The control logic generates a REQ-2 signal in response to STAT IN going inactive. Processor 2 monitors the TIL until OPL IN terminates at which time processor 2 terminates SRV OUT, completing the ending sequence. The I/O device is now no longer selected by the channel, but only *Channel End* status has been presented to processor 2. This makes it necessary for the adapter or control unit to initiate a sequence for presenting *Device End* status when it occurs; thus it is called an asynchronous status presentation (Figure 4-11).

Single Ending Sequence – Some control units, such as that in the MEMOREX Magnetic Tape Subsystem, use only the one ending sequence at the conclusion of a data transfer operation. These control units will

Table 4-10. Channel Initiated Ending

Program Execution and Hardware Operation	Basic Data Channel Logic Operation	Common Signal Lines	Adapter or Control Unit Action
			STOP indication
	CMD OUT →		→ End SRV IN
	End CMD OUT		
	BIR2 ← BIR1 ← BUS IN	← Status	← (channel end)
	Gate BUS IN		← STAT IN
Set Busy 2 FF ←	← Generate REQ-2		
G3RD ← TIL (TIL determination routine)			
G3RD ← CCR			
G3RD ← BIR (status and transfer check routine)			ENDING
G3WR → TOR	Clear REQ-2	SRV OUT →	
		End SEL OUT	
		End HLD OUT	
Set Busy 2 FF ←	← Generate REQ-2		End STAT IN
G3RD → TIL (monitor TIL's until OPL IN goes inactive)			End OPL IN
G3WR → TOR		End SRV OUT	

start the ending sequence immediately upon completion of the data transfer sequence, as is the case for a *Sense* command. They also wait until the operation is complete before starting the ending sequence, such as for a *Read* or a *Write* command.

Waiting until the operation is complete requires only one ending sequence during which both *Channel End* and *Device End* are presented. This method keeps the channel connected to the control unit until completion of the ending sequence.

Adapter or Control Unit-Initiated

When an adapter or control unit recognizes the end of a data transfer before processor 2 does at the same time, it initiates the ending sequence. In this situation there is no need to indicate *Stop* with the CMD OUT signal. An adapter simply starts the ending sequence by activating STAT IN when SRV OUT terminates after the last Data Byte transfer (dashed arrow on Figure 4-11).

Here again, as described under Double Ending Sequence, the adapter presents *Channel End* and *deselects* during the ending sequence. It then presents *Device End,* when it occurs, with an adapter-initiated asynchronous status presentation sequence. For delayed ending sequence devices, the control unit does not activate STAT IN immediately after SRV OUT ends, but waits until both *Channel End* and *Device End* are available before starting the ending sequence.

SEQUENCE CONTROLS

In addition to the basic operational sequences previously discussed, certain other operations are indicated by the order of the signal activation. The following is a discussion of these various sequence controls including both their initiation sequence and operation.

Suppress Status

Whenever the BDC processor does not want to recognize status conditions, SUPPRESS OUT (SUP OUT) is activated. With SUP OUT active, the control unit or adapter will not attempt to initiate a selection sequence to present suppressible-type status information. Status information that contains *Channel End* is not suppressible until stacked except when interface disconnect is received for that address. Status information that contains *Device End* is not suppressible, when *command chaining* has been indicated, until it is stacked.

SUP OUT must be active 250 nanoseconds before SEL OUT becomes active at the control unit or adapter to ensure suppression of status indications. Suppression of status indications prevents the initiation of an adapter or control unit selection sequence.

Command Chaining

Command chaining is indicated if SUP OUT is active when SRV OUT is activated in response to STAT IN. *Command chaining* means that another command for the I/O device in operation will immediately follow the presentation of *Device End,* provided the current operation executes without errors.

To ensure recognition of *command chaining* by the control unit, SUP OUT is active 250 nanoseconds before SRV OUT is activated in response to STAT IN, and SUP OUT must not end before STAT IN.

Interface Disconnect

The control unit or adapter will recognize *interface disconnect* whenever ADRS OUT is active and SEL OUT and HLD OUT are terminated for at least 250 nanoseconds before the completion of any operational sequence. The adapter or control unit terminates OPL IN in response, and the BDC processor thus terminates ADRS OUT to complete the *interface disconnect.* ADRS OUT must be inactive for 250 nanoseconds before a channel-initiated selection is started.

The adapter or control unit responds to the *interface disconnect* by removing all signals (except possibly REQUEST IN and MTR IN) from the common interface. For an input operation, data on the Bus In lines need not be valid after ADRS OUT becomes active. During an output operation, data on Bus Out lines must be valid until the terminations of SRV IN or OPL IN.

When the device reaches the normal ending point, the adapter or control unit attempts to obtain selection and present any generated status information to the channel. The adapter or control unit will not generate status information as a result of the *interface disconnect* if it is performed before initial status information is accepted, or after *Device End* status for an operation has been accepted by the BDC processor.

While performing an operation, the I/O device or adapter will remain busy after it receives an *interface disconnect* until the *Device End* status is accepted by the processor.

Selective Reset

Selective reset is indicated when SUP OUT is activated and OPL OUT terminated. This order of events causes the active adapter or control unit to terminate OPL IN, clear its status indication and become reset. The I/O device in operation is the only one that is reset, even on a multi-device control unit.

The operation in process will proceed to a normal stopping point, if applicable, with no further data

transfer, and *Device End* may be presented after the *selective reset*. To be effective, SUP OUT is active 250 nanoseconds before OPL OUT terminates, and remains active until 250 nanoseconds after OPL OUT again goes active. OPL OUT must remain inactive until OPL IN terminates or for at least six microseconds, whichever occurs later, for the *selective reset* to be effective. The ready or not-ready state of the adapter or control unit is not changed by a *selective reset.*

System Reset

System reset occurs whenever OPL OUT (the channel enable) and SUP OUT are inactive together and the devices are in on-line mode. *System reset* causes OPL IN to terminate and all adapters, control units and devices, along with their status information, to be reset.

The ready or not-ready state of the adapters or control units will not change because of a *system reset*. To ensure a proper reset, OPL OUT and SUP OUT must be inactive concurrently for at least six microseconds.

The particular interpretation of *system reset* by the adapters is presented in each of the adapter portions of this section.

Stack Status

In response to STAT IN, processor 2 activates CMD OUT rather than SRV OUT to indicate *stack status*. The *stack status* response is not utilized while processor 2 is operating as a selector channel.

Stop

Stop is indicated to an adapter or control unit whenever the data transfer is terminated by the BDC processor. The processor activates CMD OUT instead of SRV OUT in response to SRV IN during a data transfer sequence. The adapter or control unit recognizes the CMD OUT signal as a *stop* indication, and proceeds to the ending sequence. The byte transfer requested by SRV IN does not take place.

DATA TRANSFER OPERATIONS

The processor states perform input-output operations via their Basic Data Channel. These operations can be handled in one of two ways. Either way requires processor activity and strict compliance to the Basic Data Channel requirements and data transfer sequence restrictions.

In one way the operational sequences are controlled by the processor software with the channel control logic simply following directives. An input-output operation using this type of control is referred to as a software operation and is used for diagnostic purposes only.

The other type of input-output operation consists of both Basic Data Channel logic control and micro-instruction execution. Hardware operation is the title given an input-output operation utilizing this type of control.

Both types of input-output operations require data control within the processor and all the necessary operational sequences. Either operation allows communication with buffered units, such as the integrated adapters, or non-buffered devices such as a MEMOREX Magnetic Tape Subsystem.

The two types of input-output operations will be discussed, but only the hardware operation explained in detail along with supporting hardware/software interlace tables.

Software Operation

A software operation performs all the operational sequences required to effect the communications with the I/O device. The initial selection, data transfer and ending sequence are accomplished using INP and OUT instructions.

Each time any Tag Out or Tag In line is to be manipulated or a register read or written, processor 2 software must do it. Even during the data transfer sequence, each Data Byte transferred between the Basic Data Channel and the computer is accomplished using an INP or OUT instruction. Thus, in a software operation in the Basic Data Channel logic functions strictly as an interface reacting to signals from both processor 2 and the control unit or adapter.

The software operation is slower than a hardware operation since the Basic Data Channel processor is not using its hardware data transfer control logic. This feature allows the diagnostic program to analyze responses and reactions by both the channel control logic and the control unit during an operation. Also, since the diagnostic program initiates each step with a machine-language instruction the hardware can be stopped at the first incorrect response providing accurate information as to error location and what was expected.

Hardware Operation

A hardware operation utilizes the *System Input-Output (SIO)* micro-instruction program and the Basic Data Channel control logic. The *SIO* micro-instruction program includes an initial selection, data transfer, and ending sequence, as shown in Appendix 4B, the SIO Instruction flowchart. During the initial selection and ending sequence the channel control logic functions just as it does in a comparable software operation; that is, it is strictly an interface responding and reacting to signals from the processor and the control unit. It is during the data transfer sequence that maximum transfer

speed is obtained by use of the one major cycle data transfer micro-instruction routine and the channel control logic.

Hardware Input Operation

The following discussion will use as an example a hardware input operation from the buffered Integrated Card Reader Adapter. An *SIO* hardware operation begins with a micro-instruction initial selection sequence (Figure 4-12). During the processor 2-initiated initial selection sequence the following events occur:

- The card reader adapter is selected.
- A *Read* command is presented to the adapter.
- The status information received back from the adapter is checked for zero.

If a non-zero status indication is received, the *SIO* microprogram exits to a processor 2 error-checking program rather than starting the data transfer sequence. With a zero status indication returned, the *SIO* microprogram next initializes the channel control logic to control the data transfer by writing in the CCR and the BCR. The Hdwe Cntl Xfr1 and Input FF's are set, the HCX2 FF is cleared and the Byte Count (less one) is placed into the BCR. The *SIO* microprogram then starts the data transfer sequence and ends the selection sequence by terminating SRV OUT (Table 4-11).

The integrated adapters and some control units will respond with the first Data Byte and SRV IN without mechanical I/O device delay. Some external control units may have mechanical delay before responding. For example, with magnetic tape a certain amount of delay is involved before tape movement is up to speed and the first word read.

Once the data byte is on the Bus In lines and SRV IN goes active, the channel control logic responds. SRV IN does not enter the Request Control circuitry, but instead is enabled into the hardware-controlled Data Transfer (HCX) circuitry (page 1C13) by the CCR flip-flops. SRV IN is delayed 185 nanoseconds to allow time for the Bus In lines to deskew and for parity checking. It then triggers the Strobe one-shot circuit (STRB BIR1). The 55-nanosecond STRB BIR1 pulse gates the Bus In data into the BIR1, clocks the Bus In Register Control (BIR CNTL) FF, setting it, but does not clock the Smn Chk FF, because the ADAPTER SELECTED signal is active. The BIR Cntl FF set allows SRV IN to activate the SRV OUT signal to the adapter by setting the Channel Service Out FF (CHSRVOUT). The CHSRVOUT FF, when set, sets BIR1 Full FF if the ODD BYTE FF is set. The BIR1 Full FF, then, clears BIR Cntl and triggers the Strb BIR2 one-shot circuit.

The one-shot output pulse sets the Hardware Controlled (Transfer) Request FF (HCX REQ) generating a REQ-2 to the Execution and Control Circuitry, clears BIR1 Full FF, and gates the data byte from BIR1 to BIR2.

The adapter responds to SRV OUT by terminating SRV IN and begins placing the next Data Byte on the Bus In lines. When the data is available it will again activate SRV IN when SRV OUT is terminated.

The channel control logic uses the termination of SRV IN to clear the CHSRVOUT FF, terminating the SRV OUT signal. The trailing edge of the SRV OUT signal also clocks the BCR, advancing its count by one.

Up to this point in the data transfer, all operations have been performed by the Basic Data Channel control logic. The control logic has effected the acceptance of a Data Byte, told the adapter to proceed with the next one and is now waiting for processor 2 to respond to the REQ-2 signal and transfer the first data byte into the shared resources.

Processor 2, during the execute portion of the time slice, performs a one major cycle routine. This routine consists of six micro-instructions which perform these operations:

- Read the data byte from the BIR2 and cause it to be stored in main storage at the current byte address (CBA)
- Update the CBA by 1 and compare it with the last byte address (LBA) +1
- If CBA + 1 $\neq$ LBA + 1, clear Busy 2 FF and wait for next REQ-2 to transfer another byte.
- If LBA + 1 = CBA + 1 activate the I/O EXIT signal and proceed to the ending sequence in the SIO micro-instruction program.

The *Read Register* micro-instruction which transfers the Data Byte from the BIR2 causes the BDC logic to generate the 30-nanosecond RD WRBR signal. This signal is also generated for every *Write Register* micro instruction to the BOR1. The RDWRBR signal clears the HCX REQ FF terminating the REQ-2 signal.

Processor Not Busy when Adapter Responds – The movement of a single Data Byte from the adapter to main storage has now been completed. This progression of events will occur again for each Data Byte transferred as long as processor 2 is not busy when the adapter responds. The key to the data transfer rate is the Basic Data Channel HCX logic used in conjunction with the micro-instruction routine, which can store a byte of data during each major cycle. Figure 4-13 is a flowchart showing the sequence of events during an input data transfer.

Table 4-11. Hardware Controlled Input Data Transfer

Processor SIO μ-I and Hardware Operation	Basic Data Channel Logic Operation	Common Signal Lines	Adapter Reaction	
Set BUSY 2 FF ← Generate REQ-2			End STAT IN	End of initial selection sequence
G3WR → CCR	Set HDWE CNTL XFR1 and			
	Input FF – clear HCX2 FF			
G3WR → BCR	Byte Count – 1 → BCR			
G3WR → TOR	Clear REQ-2 End SRV OUT			
Clear Busy 2 FF				
	BIR1 ← BUS IN	← Data		
	Gate BUS IN		← SRV IN	
	Activate SRV OUT		→	
	BIR2 ← BIR1			
Set Busy 2 FF ← Generate REQ-2				

E Cycle	μ-I	Result			
0	LSI	CBA→S CBA→A +1→B			
1	LDB	$BIR_{8-15} \rightarrow D_{8-15}$	ERI ← BIR2	Clear REQ-2	1st data byte transfer events
2	SUM	CBA+1 → X			
3	LAW	(x) → A			End SRV IN
4	LBW	LBA+1 → B	Advance BCR –	End SRV OUT	
5	CIO	If A=B update Pμ, activate I/O EXIT and continue if A≠B clear Busy 2			
6 and 7					

	BIR1	BUS IN ←	Data ←	
		Gate BUS IN ←		SRV IN
	Activate →	SRV OUT →		2nd data byte transfer events
	BIR2 ← BIR1			
Set Busy 2 FF perform major cycle μ-I routine				

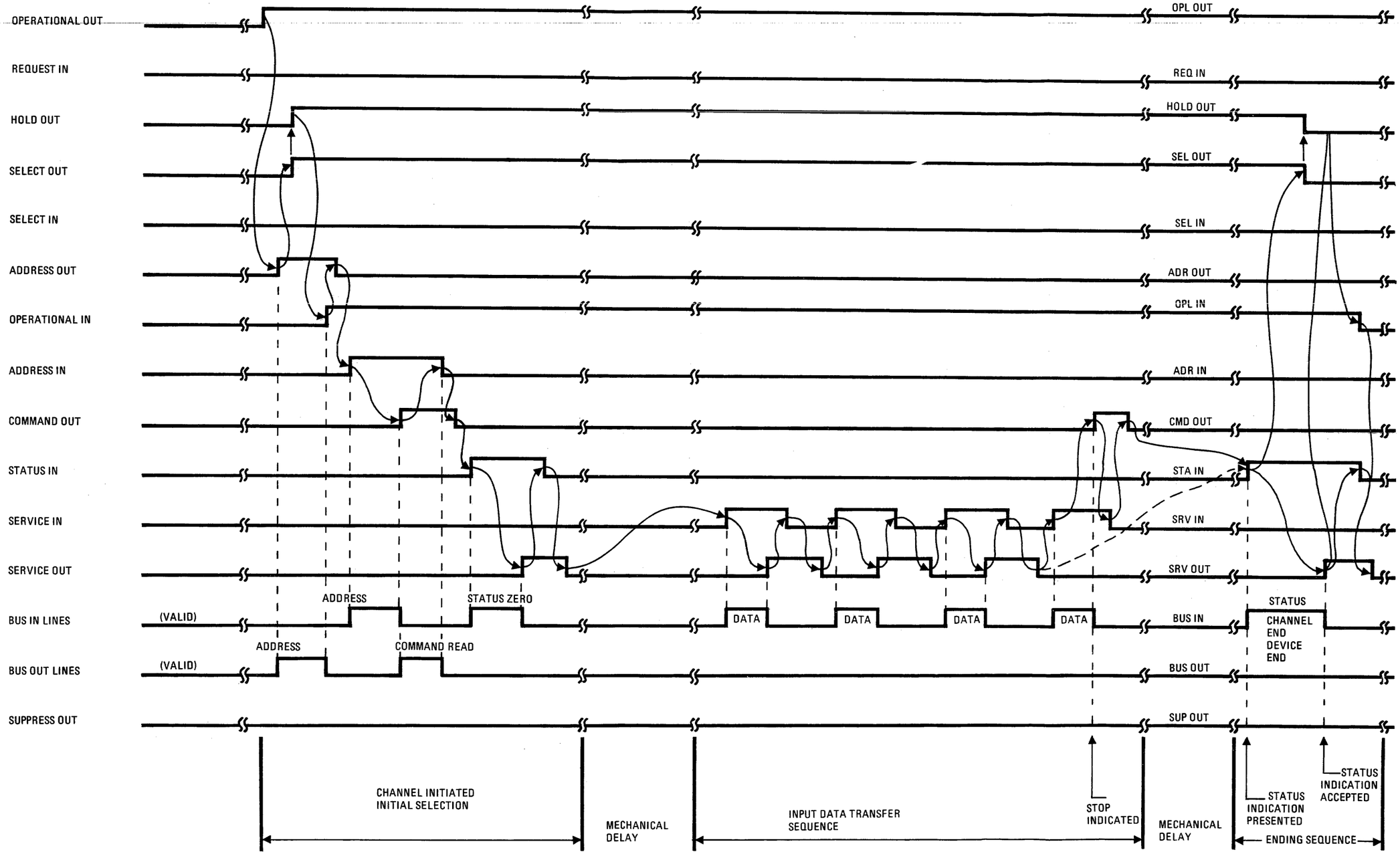


Figure 4-12. SIO Initial Selection Sequence

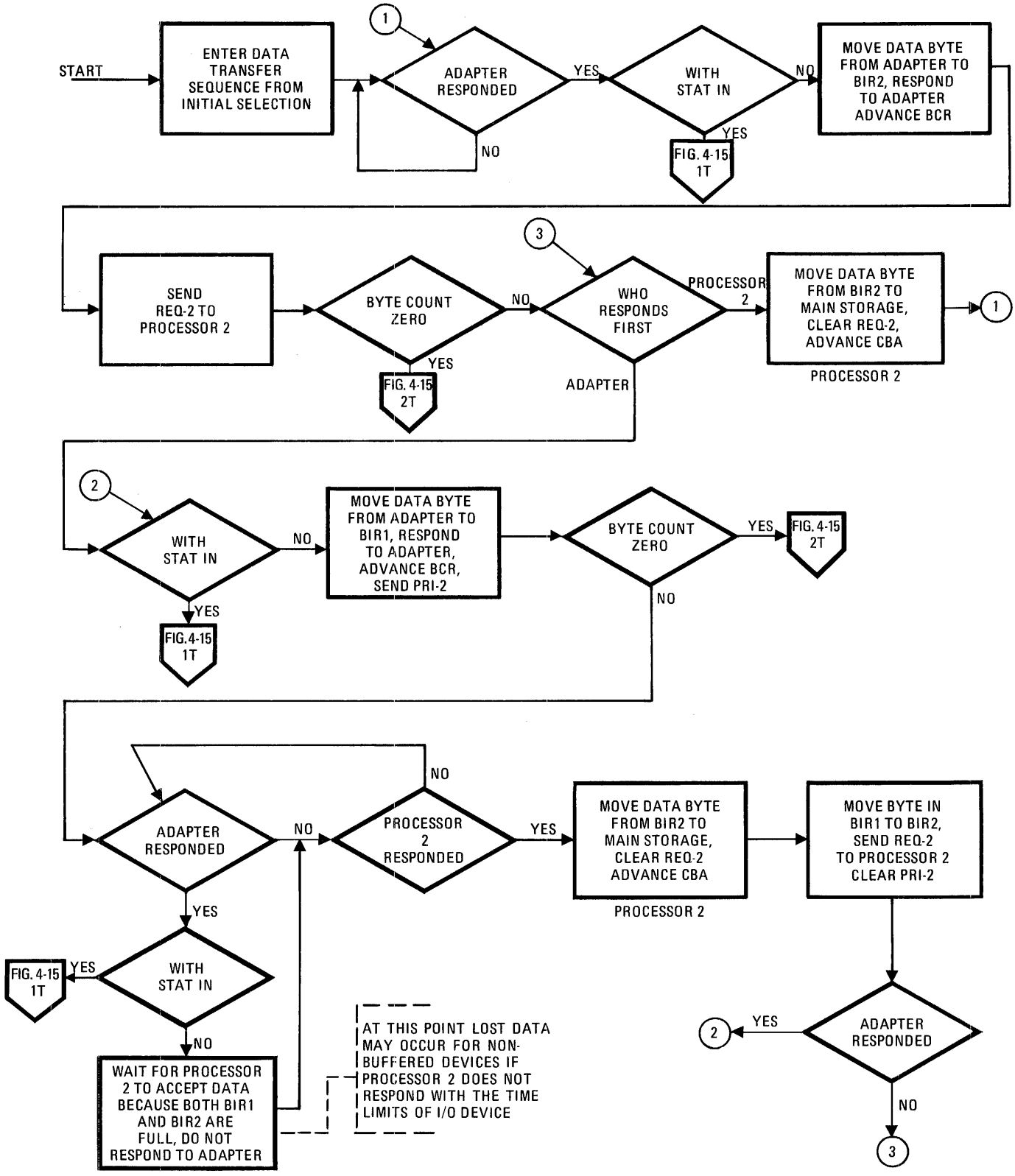


Figure 4-13. Flowchart, Input Data Transfer

Processor Busy when Adapter Responds – Processor 2 performs the major cycle micro-instruction routine asynchronous to the I/O device. The speed of the shared resources enables processor 2 to complete the routine before the adapter can present another Data Byte. However, the priority logic for processors 0, 1, and 3 can prevent processor 2 from receiving time slices. If this happens, processor 2 will not be given a time slice even though the Busy 2 FF is set. The Busy 2 FF is set by REQ-2, which the channel control logic activates when it has accepted a Data Byte from the adapter. This situation is illustrated as Byte 2 on Figure 4-14, the hardware data transfer timing diagram. The BDC logic has accepted a Data Byte, indicated this to the adapter and requested a time slice for processor 2 by activating the REQ-2 signal. Processor 2 has not obtained a time slice; therefore, the HCX Req FF is still set as a result of SRV IN for the Data Byte 2 transfer.

At this point the control logic is waiting for processor two to transfer Data Byte 2. If the adapter activates SRV IN, for Data Byte 3, before processor 2 receives a time slice, Data Byte 3 will be accepted by the control logic. The HCX logic will use SRV IN to trigger the STRB BIR1 one-shot. The one-shot output sets the BIR Cntl FF allowing the SRV OUT to be activated, and gates Data Byte 3 into BIR1 and sets the BIR1 Full FF which clears BIR Cntl FF and allows SRV OUT to drop when SRV IN drops. The STRB BIR2 one-shot will not be triggered because its input gate is disabled by the HCX Req FF, set as a result of BIR2 still being full.

The HCX logic also activates the PRI-2 signal to processor 2 indicating both ranks of the Bus In register are full. At this point the channel logic is waiting for processor 2 to transfer Bytes 2 and 3 and consequently cannot accept another byte from the adapter. The adapter will terminate SRV IN in response to Byte 3 SRV OUT, and the HCX logic terminates SRV OUT because the BIR Cntl FF is cleared. Both the channel control logic and the adapter are now stopped waiting for processor 2.

The PRI-2 signal sets the Priority Override 2 FF, if it is software enabled, placing processor 2 in the time slice operation. During processor 2's next major cycle, the data transfer routine which transfers Byte 2 from BIR2 is performed again causing the HCX logic to clear the HCX Req FF. The clearing action of HCX Req terminates the REQ-2 and PRI-2 signals to the Execution and Control Circuitry and enables the input gate to the STRB BIR2 one-shot circuit. The one-shot output pulse gates Byte 3 from BIR1 into BIR2, again sets the HCX Req FF, which activates the REQ-2 signal, and clears the BIR1 Full FF. BIR1 FF cleared allows the next SRV IN to be serviced.

As shown in this example, the hardware-controlled data transfer operation responds to the speed of both processor 2 and the integrated adapter without "data loss". The speed of the data transfer rate can vary throughout the data transfer operation.

BDC Terminated Transfer – The SRV OUT signal for the last byte, accepted from the adapter, will cause a carry out of the BCR. This is shown in Figure 4-13 by the YES exit to the question "is byte count=zero?". The CARRY OUT signal, as shown for Byte 4 in Figure 4-14, clears the Input FF. When the CARRY signal terminates, the Cmd Out Cntl FF is set if the Dat Chng FF is clear. Figure 4-15, part B shows the progression of events from the data transfer to the ending sequence.

The I/O EXIT signal is activated by the *CIO* micro-instruction near the end of the routine which gates the last Data Byte (Byte 4, Figure 4-14) to the shared resources. The I/O EXIT signal clears the Hdwe Cntl Xfr1 FF in preparation for the following ending sequence.

When the adapter activates SRV IN indicating the next Data Byte is ready, the BDC logic responds by activating CMD OUT rather than SRV OUT. This occurs since the Cmd Out Cntl FF is set. CMD OUT also sets the Illegal Length FF. The HCX logic does not gate Data Byte 5 into the BIR1 because the Input FF is cleared. The adapter responds to CMD OUT by terminating SRV IN, placing the adapter status information on the Bus In lines and activating STAT IN. CMD OUT is terminated by the SRV IN termination. The BDC logic uses STAT IN to generate a REQ-2 through the Request Control logic rather than through the HCX logic. STAT IN also clears the Cmd Out Cntl FF, and triggers the Strb BIR1 one-shot which gates the status information into BIR1.

At this point the data transfer has been completed and the ending sequence started. The ending sequence is under processor 2 micro-instruction control and the channel again is functioning as an interface responding to processor 2 and adapter signals.

Device-Terminated Transfer – During an input data transfer, when the adapter and I/O device determine that no more data is available for input, the STAT IN signal is activated rather than SRV IN. The adapter status information is also placed on the Bus In lines. Figure 4-15, part A, shows the order of events involved in a device-terminated data transfer.

The processor is expecting SRV IN and the next Data Byte. Regardless of when STAT IN is activated by the adapter, the processor does not respond until it has transferred all Data Bytes held in the Bus In register. Once all bytes have been transferred the HCX Req FF is cleared. This then allows STAT IN to generate a Tag Req signal through the Request Control logic. Tag Req activates REQ-2 to processor 2, requesting a time slice.

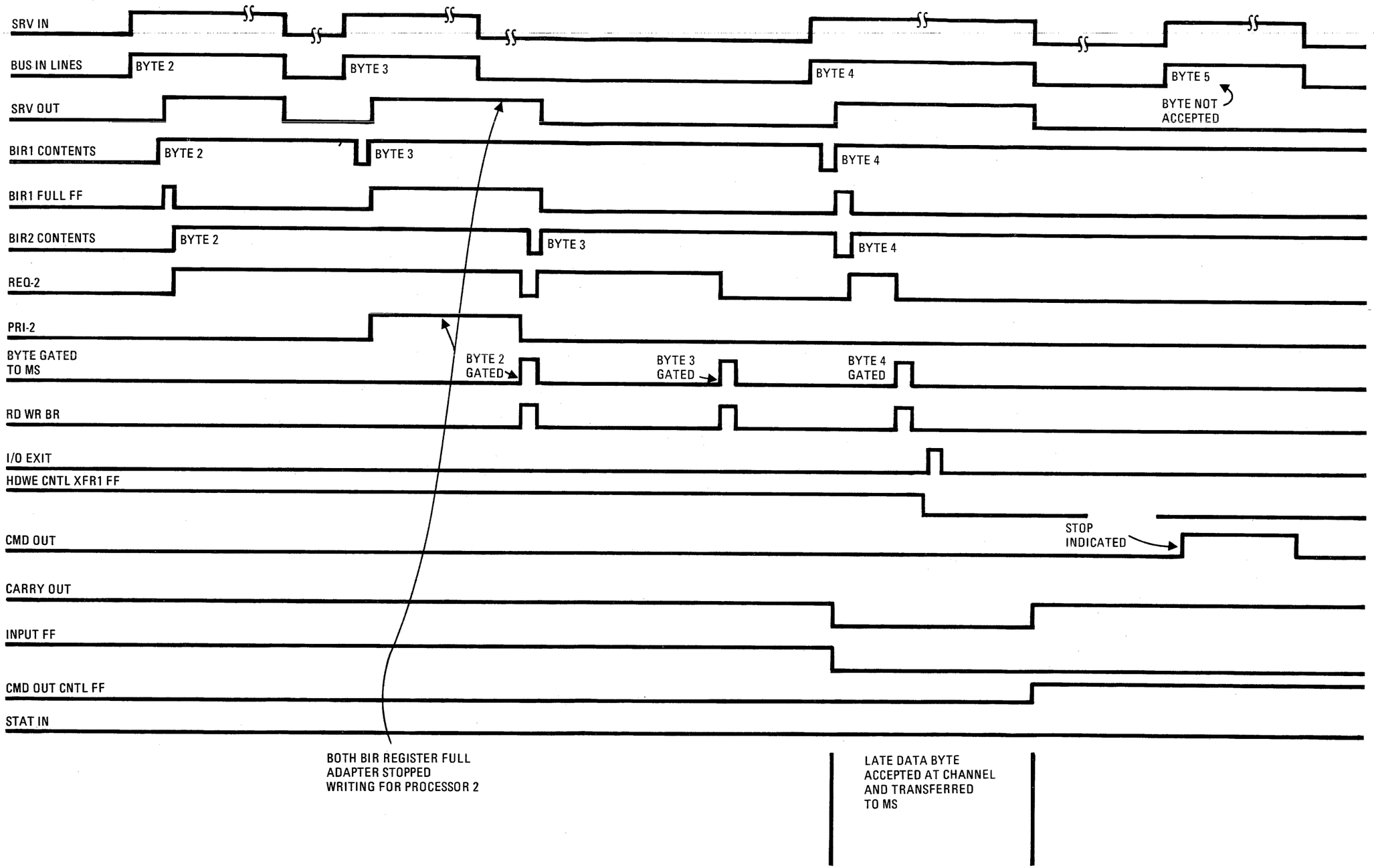


Figure 4-14. Timing Chart, Hardware Controlled-Data Transfer

STAT IN also sets the End Of Transfer (EOT) FF, activating the EOT signal to processor 2 and the HCX logic STAT IN triggers the Strobe BIR1 one-shot circuit in the HCX logic. The one-shot output pulse gates the status information on the Bus In lines into BIR1 and clocks the BIR Cntl FF STAT IN triggers Strobe BIR2 which sets the Hdwe Cntl Req FF. This FF being set disables the STAT In signal from the Request Control logic, terminating the REQ-2, but processor 2's Busy FF is already set.

In response to the Busy 2 FF being set, processor 2 performs the one-major-cycle data-transfer routine. The Status Byte which is in BIR2 is transferred and placed in main storage. At the end of the major cycle, when the *CIO* micro-instruction is executed, the EOT signal forces a compare result which activates the I/O EXIT signal to the control logic and causes processor 2 to proceed to the ending sequence program.

The I/O EXIT clears the Hdwe Cntl Xfr1, Hdwe Cntl Req, and the BIR Cntl FF in the HCX logic since the data transfer sequence is complete. The Hdwe Cntl Xfr1 FF clearing clears the EOT FF.

Assembly/Disassembly Option

During a data transfer sequence the mode of data handling is controlled by the Assembly/Disassembly Mode FF in the CCR. Regardless of the mode of data handling during the data transfer sequence, the processor always performs a one-major-cycle routine for each data transfer. This routine causes one data transfer from the BIR2 register to main storage, updates the Current Byte Address (CBA) and then compares it to the LBA+1 to determine if all data has been transferred.

When the Assembly/Disassembly mode is activated, the transfer from BIR2 to main storage is two bytes wide rather than one byte wide. The two-byte-wide transfer requires the CBA to be increased by 2 during the one-major cycle routine. In one-byte mode the CBA need only be increased by 1 for each transfer to or from main storage.

When in the Assembly/Disassembly mode, the BDC processor requests a time slice only after having transferred two bytes from the external control unit or adapter. The ability to accept two bytes and assemble them before the transfer to main storage reduces by half the number of time slices required by the processor during the data transfer. This effectively doubles the data-transfer rate between main storage and the device.

An assembly mode hardware operation begins with an initial selection sequence, followed by the assembly mode data transfer sequence, as depicted in Table 4-12, and terminates with an ending sequence.

Near the completion of the initial selection sequence, the processor readies for the assembly mode transfer (Figure 4-15) and then completes the initial selection by terminating SRV OUT.

The control unit responds by placing the first Data Byte on the Bus In lines and by activating SRV IN. The CCR state gates SRV IN to the HCX logic where it is delayed, to allow for deskew and parity checking of the Data Byte, before triggering the Strb BIR1 one-shot circuit. Its output pulse gates the bus in data into BIR1 0-7 (page DC091) because the Odd Byte FF is clear, and clocks both the BIR Cntl and Xmn Chk FF's. BIR Cntl sets and Xmn Chk sets if a parity error is detected. BIR Cntl when set, allows SRV IN to activate SRV OUT by setting the Channel Service Out FF (CH SRV OUT). Its setting clears BIR Cntl if Odd Byte is clear, which it now is.

The control unit responds to SRV OUT by terminating SRV IN and begins placing the next Data Byte on the bus in lines. When the data is available it will again activate SRV IN.

While the control unit is preparing the next Data Byte the control logic is clearing the Ch Srv Out FF, terminating SRV OUT. The trailing edge of the SRV OUT signal clocks the BCR, advancing its count by one, and clocks the Odd Byte FF to the set state.

Once SRV IN is activated for the second Data Byte, the control logic again delays it before triggering the Strobe BIR1 one-shot circuit with it. The one-shot output pulse gates the bus in data in BIR1 8-15 this time, since the Odd Byte FF is set. The one-shot output pulse also sets the BIR Cntl FF and strobes the Xmn Chk FF. BIR Cntl set allows SRV IN to activate SRV OUT by setting the Ch Srv Out FF again. This time Ch Srv Out in setting sets BIR1 Full FF because the Odd Byte FF is set one-shot circuit and clears the BIR Cntl FF. Its one-shot output gates the contents of BIR1 to BIR2, sets the HCX Req FF which generates a REQ-2 to processor 2, and clears the BIR1 Full FF.

The control unit responds to SRV OUT by terminating SRV IN and by preparing the next byte for transfer. SRV IN terminating allows the processor to terminate SRV OUT which advances the BCR count again and clocks the Odd Byte FF back to the clear state.

Up to this point all operations have been performed by the BDC control logic and the control unit. The BDC control logic has accepted two bytes from the control unit, assembled them into a 16-bit word and now is waiting for processor 2 to respond to the REQ-2 signal.

When processor 2 responds it will execute the one major cycle routine during which it will transfer the word to main storage, update the CBA and then

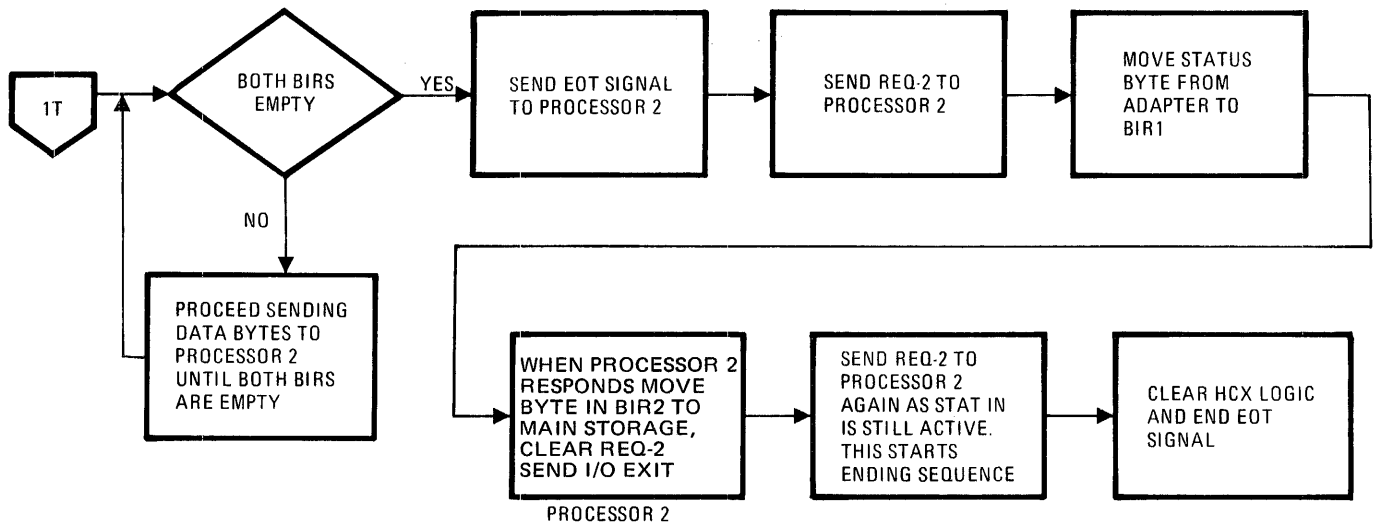
determine whether to continue or stop the transfer if all data has been transferred. This transfer to main storage occurs asynchronously to the control unit; that is, the control unit can activate SRV IN for the next Data Byte at any time and the processor can respond without affecting the transfer to main storage in any way.

Just as in byte mode, the assembly mode data transfer can vary with the speed of both the control unit and processor as affected by the priority network. The processor, in assembly mode, can store four data bytes (BIR1 and BIR2) while compensating for time differences between the processor and the control unit. Once the control circuitry senses four data bytes it activates the PRI-2 signal forcing processor 2 into the time-slicing sequence before data is lost.

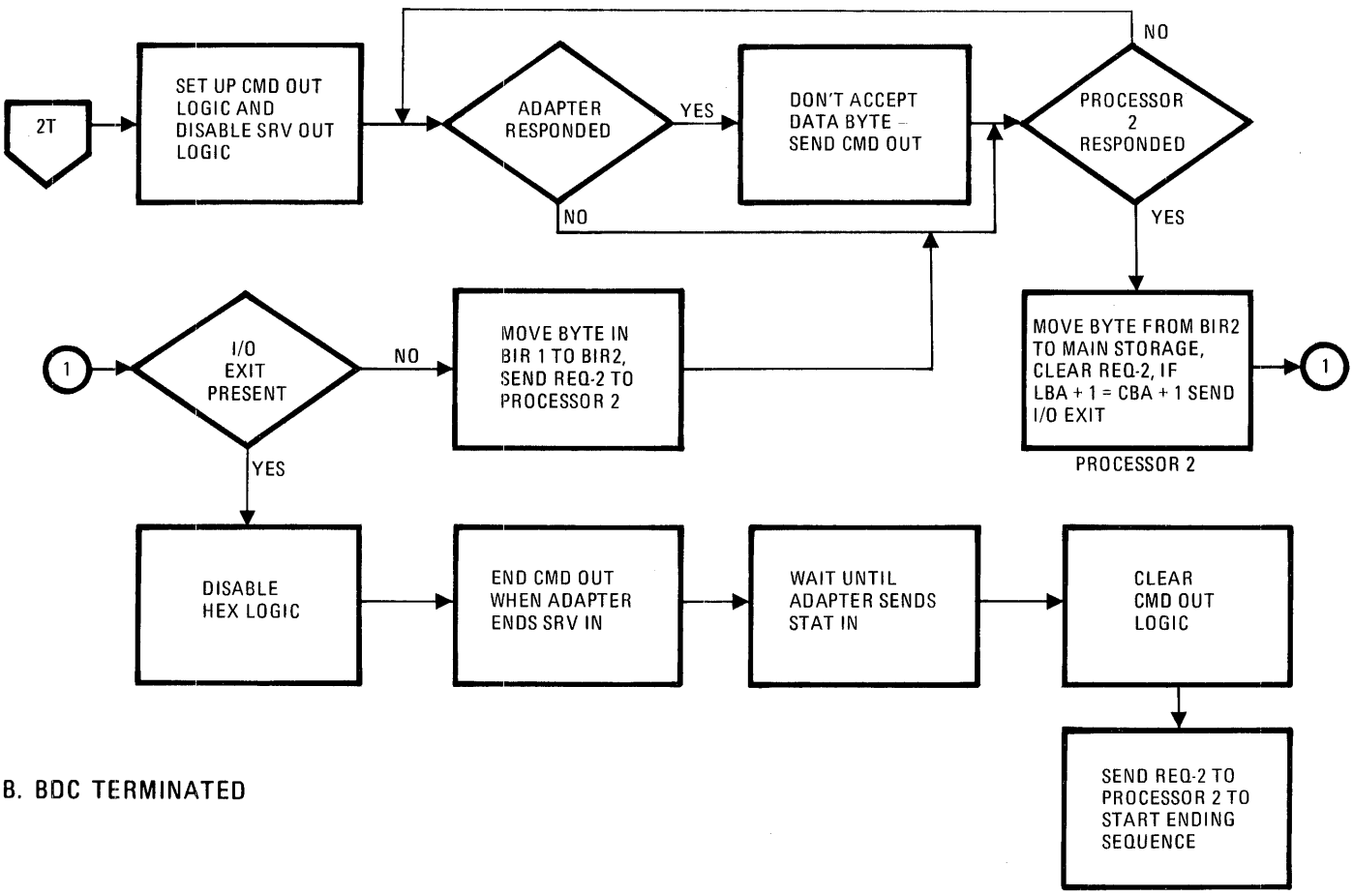
The assembly mode data transfer can be terminated by either the processor or control unit in the same manner as in byte mode data transfers.

Table 4-12. Hardware Assembly Mode Input Data Transfer

Processor SIO μ-I and Hardware Operation			Basic Data Channel Logic Operation	Signal Lines	Control Unit or Adapter Action	
Set Busy 1 FF ←			← Generate REQ-1		End STAT IN	end of initial selection sequence
G3WR → CCR			Set HDWE CNTL XFR1, Input and Assembly/Disassembly mode FF's and clear HCX2 FF	(clear Odd Byte FF to Even Byte state)		
G3WR → BCR			Byte Count-1 → BCR			
G3WR → TOR			Clear REQ-1, End SRV OUT			
			BIR1 ← BUS IN	← Data		
			Gate BUS IN to BIR1 0-7	←	SRV IN	1st data byte
			Activate SRV OUT	→	End SRV IN	
			End SRV OUT			
			Set ODD BYTE FF			
			Advance BCR			
			BIR1 ← BUS IN	← Data		
			Gate BUS IN to BIR1 8-15	←	SRV IN	2nd data byte
			Activate SRV OUT	→		
1st 2 byte-wide transfer to main storage			BIR2 0-15 ← BIR1 0-15			
Set BUSY 1 FF ←			← Generate REQ-1		End SRV IN	
E Cycle	μ-I	Result				
0	LS2	CBA → S	End SRV OUT			
		CBA → A	Advance BCR			
		+2 → B	Clear ODD BYTE FF			
1	LDW	D 0-15	BIR2 0-15 Clear REQ-1			
2	SUM	CBA+2 → X				
3	LAW	(x) → A				
4	LBW	LBA+1 → B	BIR1 ← BUS IN	← Data		
5	CIO1	If A=B update Pμ, activate I/O EXIT and continue if A≠B clear BUSY 1 FF	Gate BUS IN to BIR1 0-7	←	SRV IN	3rd data byte
			Activate SRV OUT	→		
			Set ODD BYTE FF			
					End SRV IN	
			End SRV OUT			
			Advance BCR			
6 and 7						



A. DEVICE–TERMINATED



B. BDC TERMINATED

Figure 4-15. Flowchart, Hardware Transfer Termination

Hardware Output Operations

A hardware output operation to the line printer will be discussed as an example. A hardware output operation begins, as does an input operation, with a micro-instruction initial selection sequence. During the processor 2-initiated selection the following events occur (Figure 4-16):

- The line printer is selected.
- A *Write* command is presented to the adapter.
- The status information received from the adapter is checked for zero.

Once the status indication of 0 is determined, the processor 2 *SIO* microprogram sets up the BDC control logic for a hardware-controlled output data transfer. The Hdwe Cntl Xfr1 FF is set and the HCX2 FF is cleared. The byte count less one is placed in the Byte Count register. The *SIO* microprogram next starts the data transfer sequence and ends the selection sequence by terminating SRV OUT (Table 4-13).

During the time slice following after the selection sequence, processor 2 performs a one-major-cycle *SIO* output routine. This routine is similar to the input routine in that it handles a single data byte each time it is executed. The output routine consists of six micro-instructions which accomplish the following:

- Read the data byte at the CBA of main storage and place it in BOR.
- Update the CBA by one and compare it with the LBA+1.
- If LBA+1 = CBA+1 clear Busy 2 FF and wait for the next REQ-2.
- If LBA=1 = CBA=1 activate the I/O EXIT signal, leave the Busy 2 FF set and proceed to the ending sequence routine.

Within the routine, the *SDB* is the micro-instruction which places the Data Byte on the ERO lines and activates the ERFG3WR signal to the channel logic. The control logic, as a result, generates the WR REG pulse which clocks the Data Byte into BOR1. The WR REG pulse also ANDs with ERNG3-01 to generate a WR BOR pulse which clocks the BOR1 Full FF in the HCX logic (page 1C13). The BOR1 Full FF, being set, triggers the STRB BOR2 one-shot circuit if the BOR2 Full FF is clear, indicating the BOR2 does not contain a Data Byte. The STRB BOR2 output pulse gates the Data Byte in the BOR1 to the BOR2, sets the BOR2 Full FF and clears the BOR1 Full FF. The Data Byte is now in BOR2 and BOR1 can accept another byte from the shared resources. The trailing edge of STRB BOR2 pulse clocks the HCX Req FF, causing it to set, generating another REQ-2 to processor 2. At this point the Data Byte is on the Bus Out lines waiting for the printer to accept it; the processor 2 microprogram has been completed (Figure 4-17, point A).

During the next major cycle, processor 2 will again place a Data Byte in the BOR1 register, perform the byte address compare and the resulting proper exit. The HCX logic will again set the BOR1 Full FF and clear the HCX REQ FF terminating the REQ-2. BOR1 Full does not trigger the one-shot circuit because BOR2 Full FF is still set.

SRV IN is activated by the printer in response to the termination of SRV OUT which occurred at the end-of-selection sequence. SRV IN is activated when the printer is ready to accept the first Data Byte regardless what processor 2 is doing.

Although SRV IN is activated asynchronous to processor 2 it will not be allowed to set the Ch Srv Out FF until a Data Byte has been placed in BOR1, gated to BOR2, BOR2 Full FF set, and a 260-nanosecond delay timed out. The delay allows the Bus Out lines to deskew and for parity to be generated before the SRV OUT signal is activated by the setting of the Ch Srv Out FF.

In the above description, both BOR's have been filled with data before SRV IN was activated. Therefore, the Ch Srv Out FF is set immediately by HCX logic in reaction to SRV IN. The printer responds to SRV OUT by first accepting the data on the Bus Out lines and then terminating SRV IN. The HCX logic responds to SRV IN termination by ending the SRV OUT signal and clearing the Ch Srv Out FF. SRV IN will again be activated when the printer is ready for the next byte.

SRV OUT clearing clocks the BCR, advancing the count by one, and clears the BOR2 Full FF. BOR2 clear enables the STRB BOR2 one-shot to fire. The one-shot pulse gates the Data Byte from BOR1 to BOR2, again sets BOR2 Full FF, and clears the BOR1 Full FF, which in turn sets the HCX REQ FF.

The BDC control logic is now asking processor 2 for another Data Byte with the REQ-2 signal, has a byte on the Bus Out lines and is waiting for the printer to accept it. If the printer responds and accepts the Data Byte before processor 2 fills the BOR1 the HCX logic will activate the PRI-2 signal to the Execution and Control Circuitry. The PRI-2 signal in this case indicates both BOR's are empty.

(This situation is acceptable to the buffered adapters and control units since they can wait until shared resources transfer the next byte to the channel, but for a non-buffered device such as a magnetic tape unit this situation is not acceptable. When the magnetic tape control unit asks for a Data Byte with SRV IN before processor 2 has placed a byte in the BOR's, the HCX logic will not respond with SRV OUT. If the tape control unit does not receive SRV OUT by the time it is ready to write the next frame it will terminate the write operation and proceed into an ending sequence.)

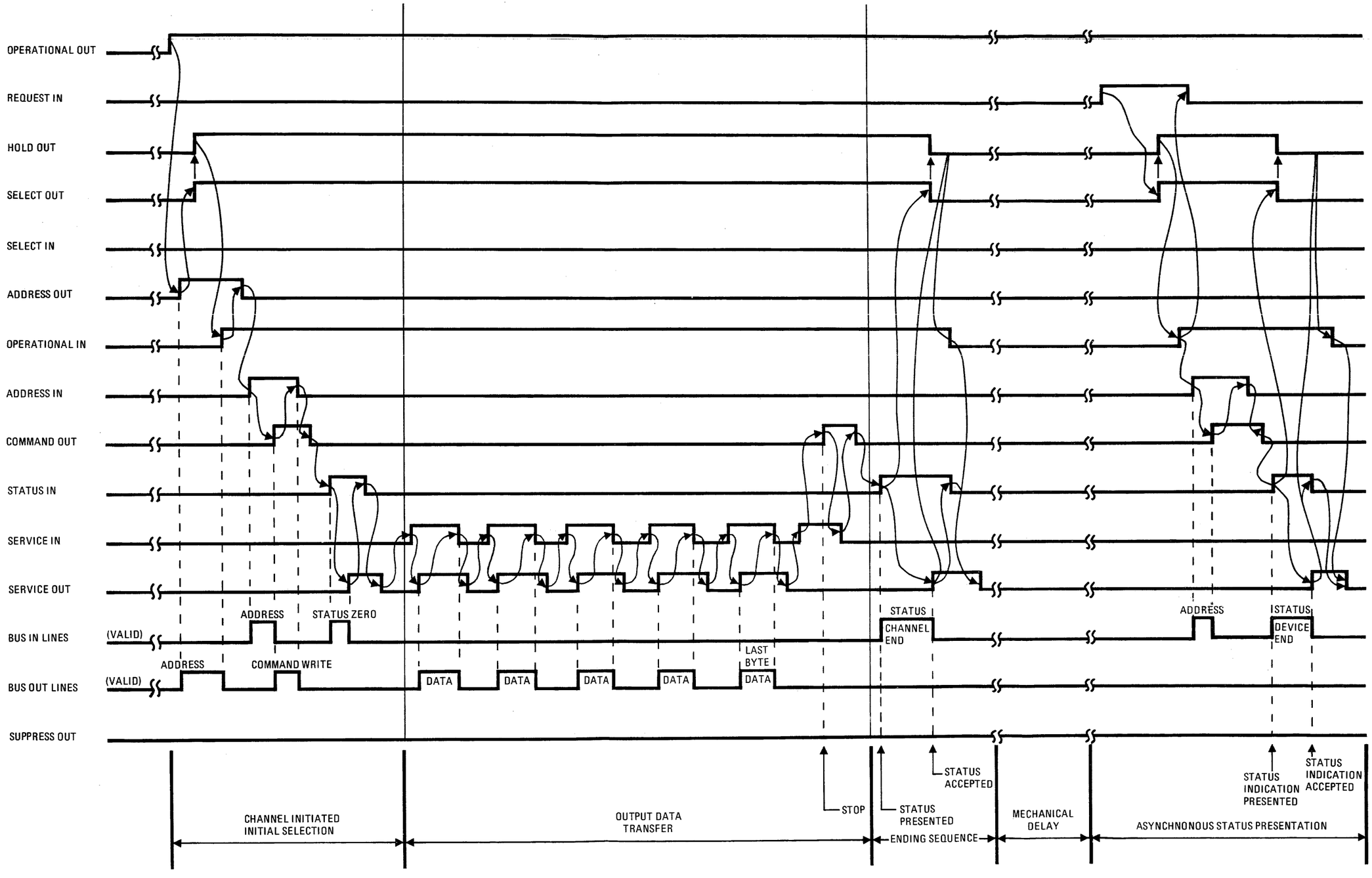


Figure 4-16. Selector Mode Output Sequence

Table 4-13. Hardware Controlled Data Transfer (Output)

Processor Program Execution and Hardware Operation			Channel Logic Operation and Reaction	Common Interface Signals	Adapter Reaction	
Set Busy 2 FF ←			Generate REQ-2		End STAT IN	End of initial selection sequence
G3WR → CCR			Set HDWE CNTL XFR1 and clear HCX2 FF's			
G3WR → BCR			Byte count -1 → BCR			
G3WR → TOR			Clear REQ-2 - End SRV OUT			
E Cycle	μ-I	Result				
0	LSI	CBA→S CBA→A +1 → B				
1	SUM	CBA+1 → X		←	SRV IN	Data byte transferred
2						
3	LAW	(x)→A				
4	LBW	LBA+1 → B				
5	SDB	D_{8-15} → BOR_{8-15}	Data BOR1 → BOR2 → BUS OUT	→ Data	→	
6	CIO	If A=B update Pμ, activate I/O EXIT and continue; if A≠B clear Busy 2 FF	SRV OUT	→	→	Data byte transferred
7					End SRV IN	
Set Busy 2 FF ←			Generate REQ-2 - End SRV OUT			
E Cycle	μ-I	Result				
0	LSI	CBA→S CBA→A +1 → B				
1	SUM	CBA+1 → X				
2						
3	LAW	(x)→A				
4	LBW	LBA+1 → B				
5	SDB	D_{8-15} → BOR	Data BOR1 → BOR2 → BUS OUT	→ Data	→	
6	CIO	A=B activate I/O EXIT, update Pμ and continue			← SRV IN	
7			SRV OUT		→	
Clear Busy 2 FF and wait for REQ-2 generated by STAT IN			← End SRV OUT		End SRV IN	Data byte transferred

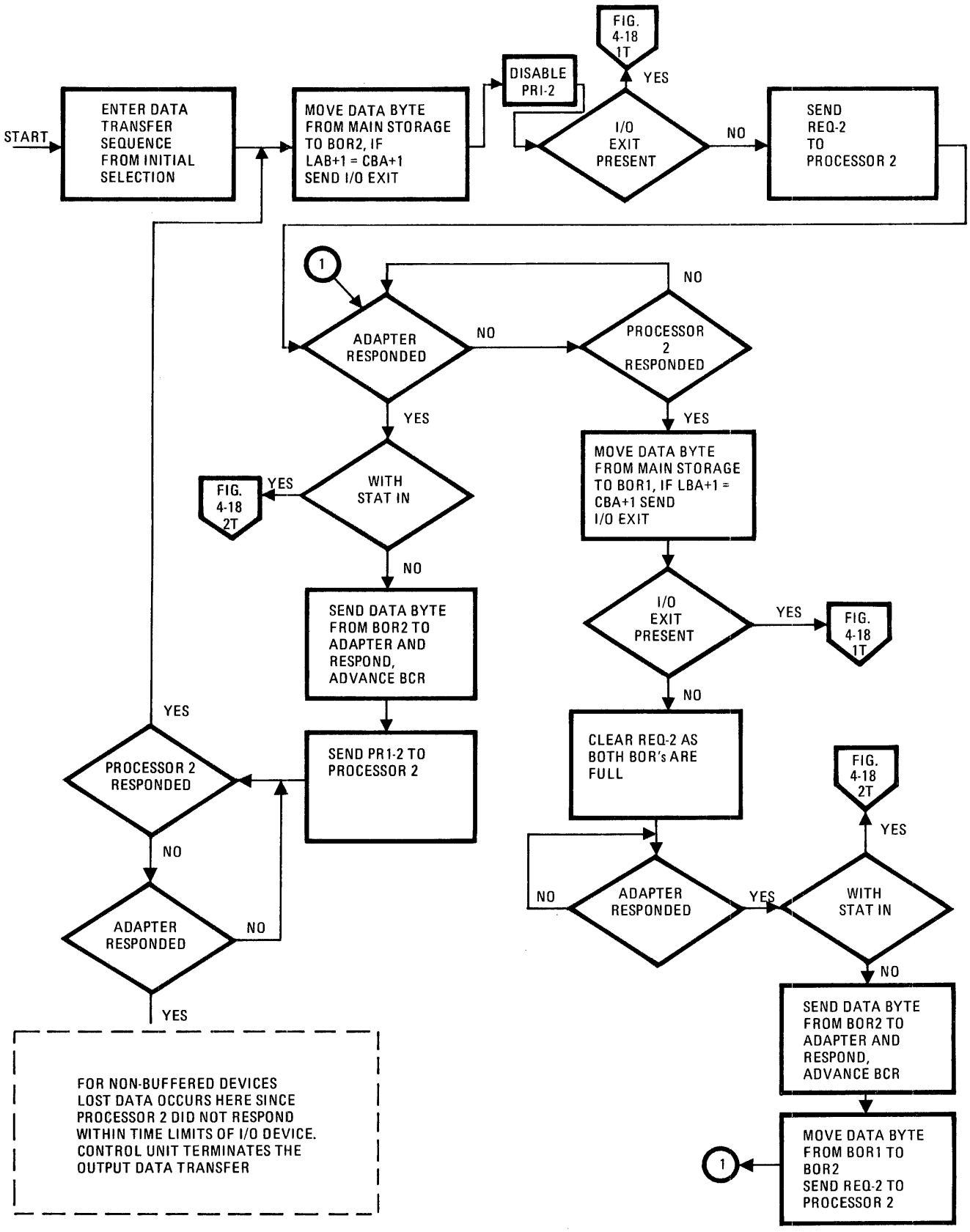


Figure 4-17. Flowchart, Hardware-Controlled Output Data Transfer

To avoid "lost data" the PRI-2 signal sets the Priority Override 2 FF if enabled, placing processor 2 in the time slice operation. Processor 2, during its next major cycle, again performs the routine placing a Data Byte in BOR1. As soon as the BOR1 Full FF sets, the PRI-2 signal is disabled.

Basic Data Channel output data rates are flexible. They can vary with the speed of processor 2, provided they stay within the required time limits of the I/O device.

Processor-Terminated Transfer – Processor 2 determines the output data transfer is complete with the *CIO1*. During the major cycle in which the LBA+1 equals the CBA+1, *CIO1* micro-instruction activates the I/O EXIT signal to the BDC control logic and exits to the ending sequence microprogram. The I/O EXIT clears the Hdwe Cntl Xfr1 FF which prevents the HCX logic from activating the REQ-2 signal again (Figure 4-18, part A).

The HCX logic responds to SRV IN from the control unit in a normal data transfer fashion until both BOR's are empty. The SRV OUT signal for the last Data Byte transferred causes a carry out of the BCR which sets the Cmd Out Cntl FF, if the Dat Chng FF is clear.

When the control unit asks for the next Data Byte with SRV IN the HCX logic will activate CMD OUT indicating *stop*. The control unit now proceeds into an ending sequence and terminates SRV IN. The HCX logic responds by terminating CMD OUT. The BDC control logic uses the Request Control logic to generate a REQ-2 signal in response to STAT IN from the control unit. STAT IN also clears the CMD OUT CNTL FF.

The Execution and Control Circuitry responds to the REQ-2 and starts the ending sequence program without further use of the HCX logic in the BDC control logic.

Device Terminated Transfer – The control unit can terminate the output data transfer by activating STAT IN rather than SRV IN. The control unit at this time also places its status information on the Bus In lines.

STAT IN will not be recognized by the BDC control logic until both BOR's have been filled and HCX logic is no longer requesting time slices (HCX Req FF clear). Although the BOR's must be full before STAT IN is recognized, the two Data Bytes are not transferred to the control unit. Therefore, the count in the BCR is correct because it indicates only the number of bytes actually transferred to the control unit.

STAT IN, when enabled by the clear HCX Req FF, enters the Request Control logic rather than the HCX logic. It sets the EOT FF and generates a REQ-2 signal (Figure 4-18, part B). Processor 2 responds by again performing the one-major-cycle output routine. During the routine, it attempts to transfer another Data Byte to the basic data channel before executing the *CIO1* micro-instruction. Upon completion of *CIO1,* processor 2 proceeds to the ending sequence program because of the compare condition. The ending sequence program is started in response to the still active REQ-2 signal. As previously stated, the ending sequence will proceed completely under microprogram control.

Data Chaining

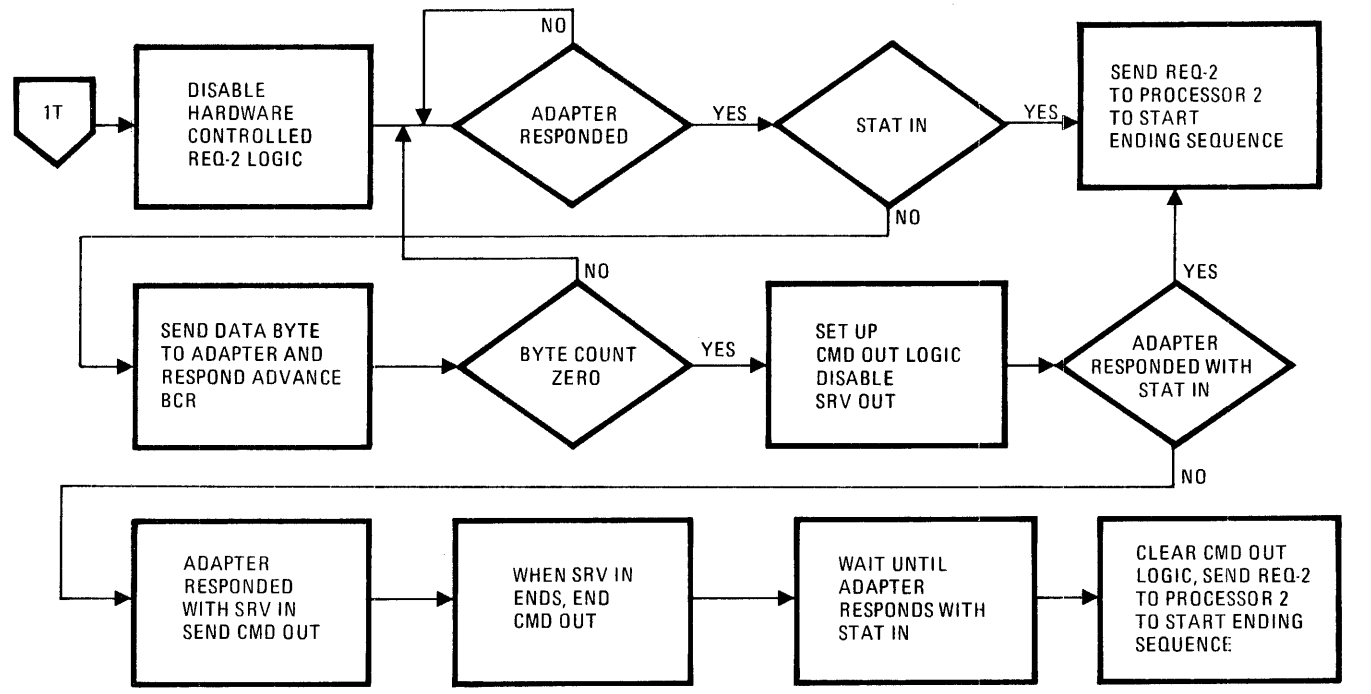
Data chaining may be implemented during hardware operations by utilizing the Dat Chng FF in the CCR. The Dat Chng FF, when set, prevents the Cmd Out Cntl FF from becoming set and prevents the channel control logic from recognizing the I/O EXIT signal from processor 2.

With the *data chaining* bit set in the CCR the *SIO* routine can find LBA+1 equal to CBA+1 and exit from the data transfer routine without clearing the Hdwe Cntl Xfr1 FF. Likewise, the BDC control logic can determine that all bytes have been transferred (by carry out of BCR) and can clear the Input FF without enabling the HCX logic to generate a CMD OUT signal.

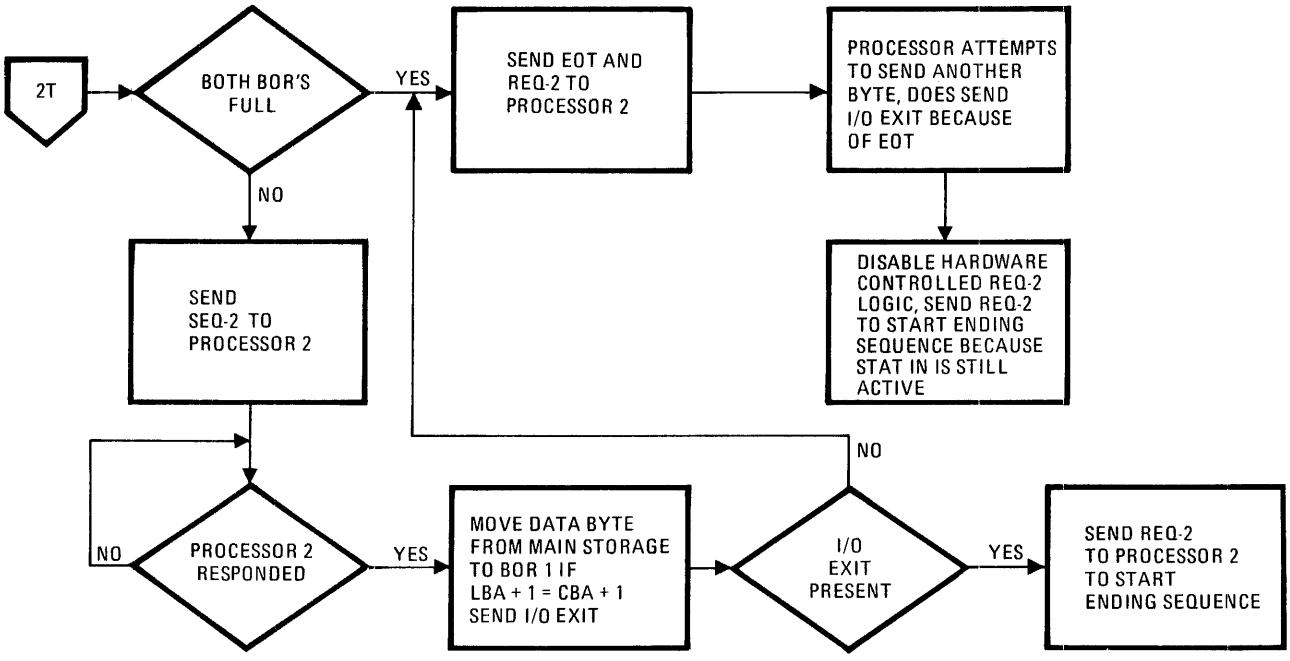
These features allow processor 2 to set up a new CBA and LBA+1 (starting address of the next data area in main storage), write the new byte count minus one in the BCR, and then start the HCX data transfer sequence again without causing an ending sequence. In this manner, many segments of data may be stored in different areas of main storage during one data transfer sequence. To terminate data chaining, processor 2 clears the Dat Chng FF in the CCR prior to starting the data transfer sequence for the last data segment. This will allow an ending sequence to be started when the LBA+1 equals the CBA+1 of the last data segment.

High-Speed Transfer

A higher speed transfer rate is obtained when the DATA IN and DATA OUT signals are used in conjunction with the SRV IN and SRV OUT signals during a hardware-controlled data transfer. BDC control logic reacts to external control units using the high-speed transfer transparent to processor 2, and for this reason the *SIO* microprogram need not change. Processor 2 performs the same one-major-cycle routine each time REQ-2 sets the Busy 2 FF regardless of whether or not it is performing a high-speed transfer or a normal data transfer.



A. BDC TERMINATED



B. DEVICE–TERMINATED

Figure 4-18. Flowchart, Hardware Output Termination

The speed increase is due to the use of two pair of tag lines with each pair overlapping the other. Each pair of tag lines controls alternate Data Bytes during the transfer, thereby reducing the cable length delay of the TAG signals. Figure 4-19 shows a high-speed input data transfer using both pairs of tag lines and the byte-wide data bus.

A control unit utilizing the high-speed transfer may activate DATA IN in response to SRV OUT before terminating SRV IN, but it is not valid until SRV IN is terminated. When SRV IN is terminated, the HCX logic (page DC131) clears the CHSRVOUT FF which terminates SRV OUT. The HCX logic next gates the data in the Bus In lines into BIR1 because both SRV OUT and DATA OUT are inactive. Once the data is in BIR1 it is gated into BIR2 if the HCX REQ FF is clear. The CHDATOUT FF is also set.

HCX Req FF clear indicates that the processor 2 has accepted all the data from the BDC logic. As soon as the data is placed in BIR2 the HCX Req FF is set signaling the shared resources another byte is ready for transfer. As long as processor 2 obtains a time slice and stores the byte in main storage, before the control unit can present its next byte, the input operation will proceed as shown in Figure 4-19 for bytes 1, 2, 6 and 7.

The control unit will activate SRV IN in response to SRV OUT termination, and may do so before it terminates DAT IN in response to DATA OUT. Regardless of when SRV IN is activated, it is not valid until DATA IN terminates. The control unit has placed the next Data Byte on the Bus In lines before activating SRV IN.

DATA IN, going inactive, clears the CHDATOUT FF and causes the HCX logic to gate the Data Byte into BIR1 because SRV IN is active. Once that's completed, the CHSRVOUT FF is set which sets BIR1 Full FF and allows the byte to be transferred into BIR2 if the HCX Req FF is clear. The control unit will again place the next byte on the Bus In lines, terminate SRV IN, and activate DATA In.

The hardware-controlled high-speed transfer will continue, varying with the speed of the control unit and processor 2.

The BDC control logic will activate the PRI signal when processor 2 gets behind the control unit, just as during the normal hardware-controlled transfer. As shown in Figure 4-19 the BDC control logic handles the high-speed transfer over a range of varying speed transparent to processor 2.

INTEGRATED CARD READER ADAPTER (ICRA)

NOTE

Where ICRA descriptions also apply to the Integrated Reader/Punch Adapter (IRPA), they are so generalized and marked. Descriptions unique to the ICRA make specific reference to it; in some generalized descriptions, there are isolated descriptions of the IRPA included because they best fit into the generalized context.

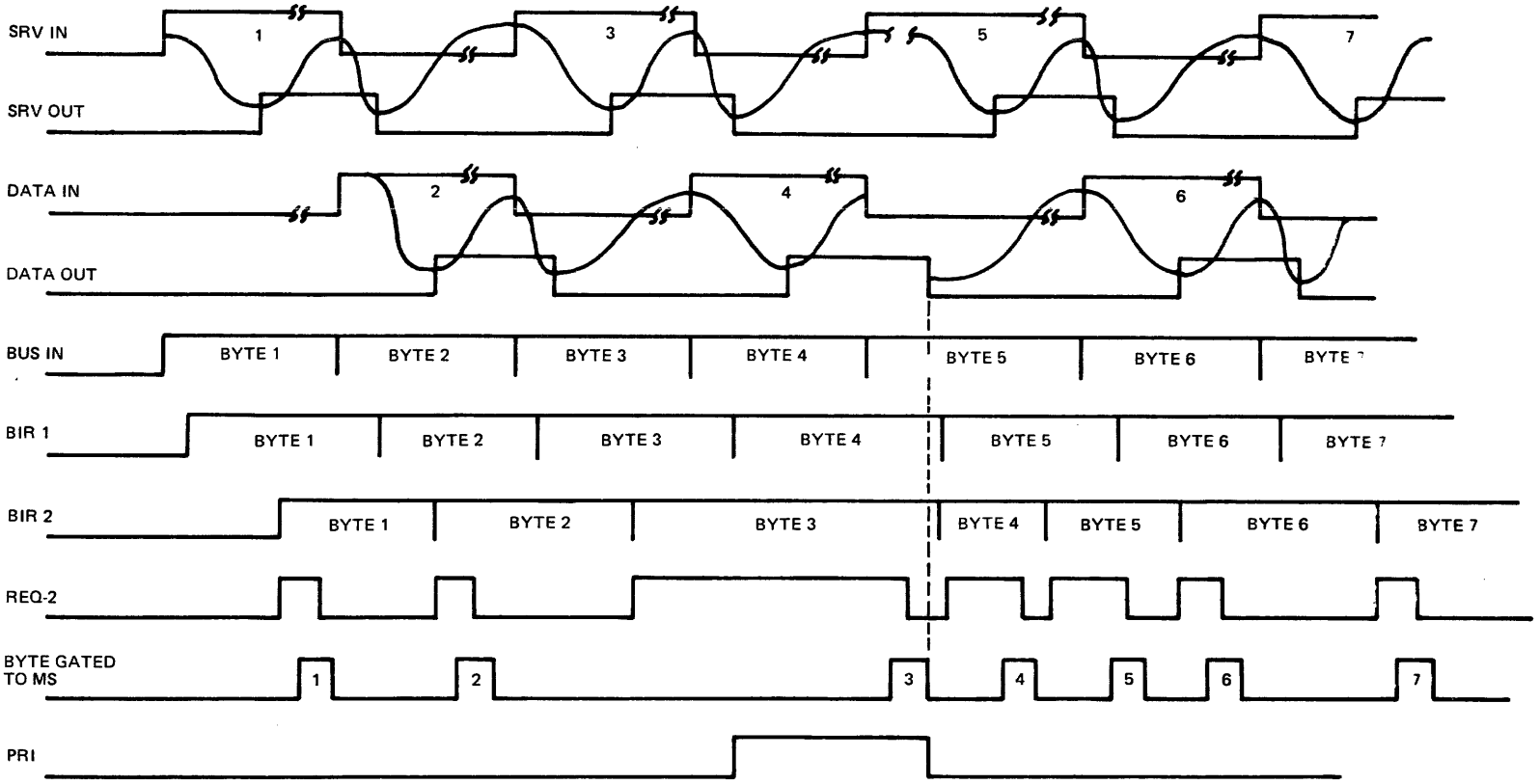
GENERAL DESCRIPTION

The ICRA is the card reader controlling element. It contains the communications path and control logic between processor 2 and the system card reader. The ICRA provides the system with the ability to read information from 80-column cards and place it in either main storage or control storage.

The ICRA performs the following operations:

- Transfers binary card data, or converts EBCDIC card coded data to EBCDIC characters before transfer.
- Performs a validity check when reading and converting EBCDIC data.
- Provides an 80-column data buffer register.
- Transfers the data buffer register contents and reads the next card in response to a single *Read and Feed* command.
- Reads and stores the first card data in the buffer register prior to the first *Read* command.
- Presents all necessary status and sense information when requested by the processor.
- Provides a Control Storage Load (CSL) capability.

The ICRA communication paths and block diagram are shown in Figure 4-20. All ICRA communications via the Basic Data Channel follow the order of events for the BDC2 operational sequences. The control and data signals required by the CSL feature connect to the Control Storage Loader circuitry in card location CB17. The data and control signals from the card reader connect to the ICRA through a back panel connector.



NOTE: SEE TEXT FOR SRV IN AND DAT IN TIMING RELATIONSHIP

Figure 4-19. Timing Chart, High-Speed Input Transfer

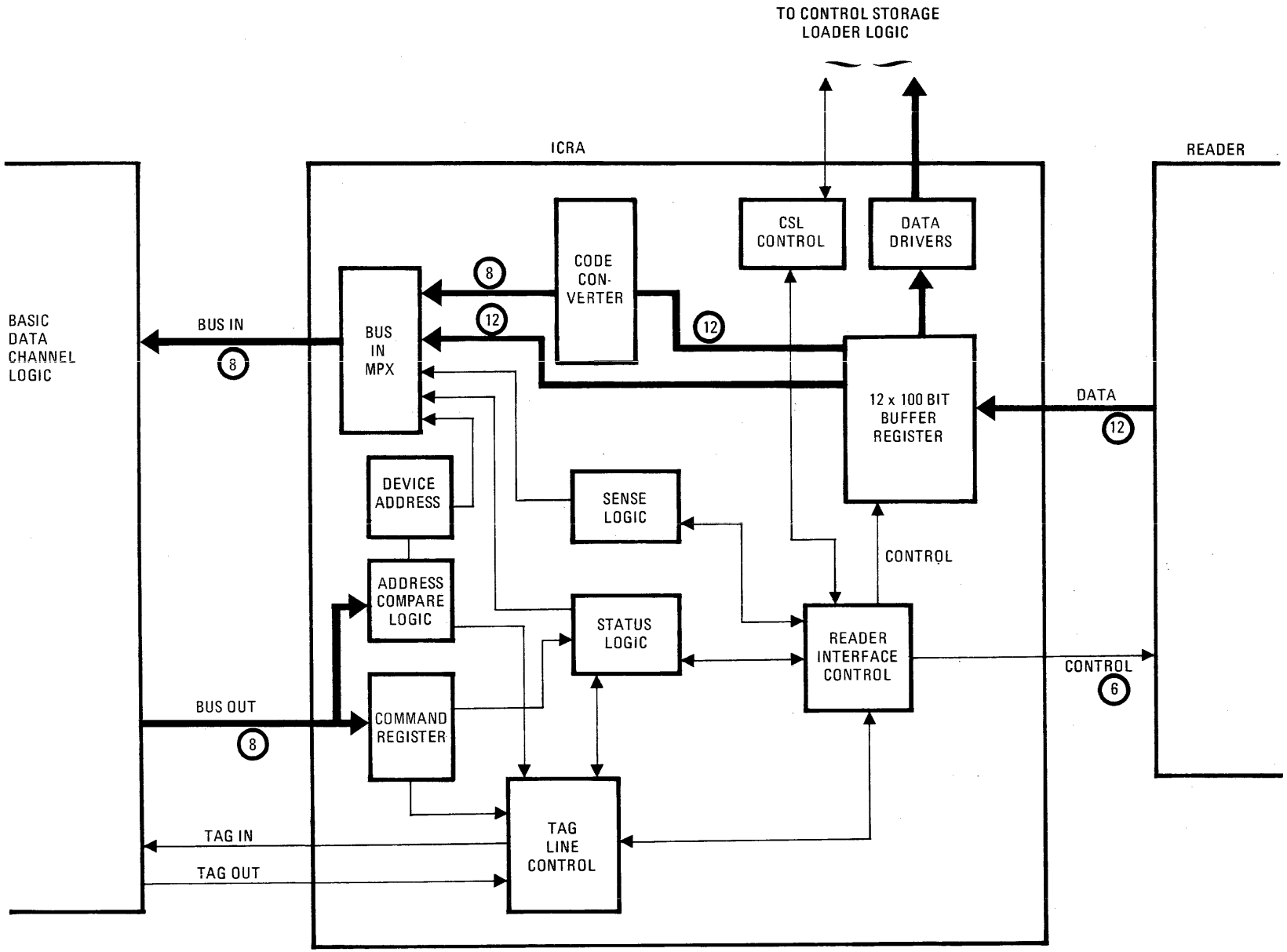


Figure 4-20. ICRA Block Diagram

Reader Commands

The ICRA recognizes as valid the *Read and Feed, Sense, No-operation,* and *Test* commands from the BDC2 processor. In response to the *Read and Feed* command, the Buffer register data is transferred to the Basic Data Channel, and the next card is read. The other three commands cause no card motion during execution.

A description and some programming considerations for each of the reader commands follows.

- NO-OPERATION

  Command Byte:

Bits	0	1					6	7
	0	0	0	0	0	0	1	1

  Normal Status Indications: *Channel End* and *Device End* at initial selection.

  Other Status Indications: *Unit Check* (bit 6) occurs at initial selection if the Data Buffer register has not been loaded. *Busy* (bit 3) occurs at initial selection if the reader has not reached the *Device End* point or if the ICRA has outstanding status.

- TEST I/O

  Command Byte:

Bits	0	1					6	7
	0	0	0	0	0	1	0	0

  Operation Performed: Transfer status information to BDC2 and clear Status flip-flops.

  NOTE

  Busy indication (bit 3) is included in the Status Byte if the reader is executing a *Read* command and the ICRA has no outstanding status.

- SENSE

  Command Byte:

Bits	0	1					6	7
	0	0	0	0	0	1	0	0

  Operation Performed: Transfer the one Sense Byte to the BDC2 during the data transfer sequence. Present *Channel End* and *Device End* during the ending sequence.

  Normal Status Indication: Zero at initial selection.

- READ AND FEED

  Command Byte:

Bits	0	1					6	7
	0	0	X	0	0	0	1	0

  0 = EBCDIC MODE

  1 = CARD IMAGE MODE

  Operation Performed: Transfer the Buffer register data to BDC2. Read next card and place its data in Buffer register. Present *Channel End* during an ending sequence after the last Data Byte is transferred. Present *Device End* during an adapter-initiated sequence after the next card has been read.

  Normal Status Indication: Zero at initial selection.

  Other Status Indications: *Unit Check* (bit 6) occurs at initial selection if the previous card did not feed. Unit Check occurs with CE and DE if an error was detected during the reading of the previous card.

  Busy (bit 3) occurs at initial selection if *Read* command issued while reader is busy.

Status Information

(Data in this discussion is presented in generalized terms, to apply equally to the ICRA and the Integrated Reader/Punch Adapter (IRPA) except where specifically noted.)

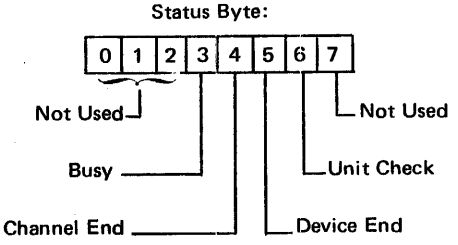
The Status Byte is presented to Basic Data Channel 2 by the adapter under the following conditions:

- During the initial selection sequence after a Command Byte is received.
- During the ending sequence after a data transfer.
- During an adapter-initiated sequence to present previously stacked status indication(s) or *Device End*.

The status flip-flops are cleared when the Status Byte is accepted by the Basic Data Channel except as follows:

- *Unit Check* is not cleared until the condition is corrected by the operator, when the condition causing it requires intervention.
- *Busy* is not cleared until *Device End* is accepted by BDC2.

The bits within the Status Byte presented by the adapter have the following meaning.



Bit 3 – *Busy* is presented to BDC2 only during an initial selection sequence. Once a command

has been accepted by the adapter, *Busy* is presented in response to all commands, except the *Test I/O,* when *Device End* or *Channel End* is included in the Status Byte. The *Busy* bit indicates the card device is still executing the last command or the adapter has outstanding status.

Bit 4 – *Channel End* is presented to BDC2 after completing the data transfer portion of a command or during the initial selection of an immediate command (*Test I/O, No-Operation*). *Channel End* indicates the specified operation has been completed to a point where the Basic Data Channel is free to disconnect from the adapter.

Bit 5 – *Device End* is presented by the adapter when the card device has completed the previous command, indicating the adapter can accept another command once the status indication is accepted. *Device End* is also generated when the device goes from a not-ready to a ready state.

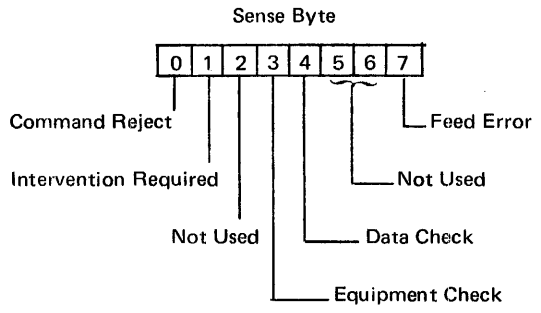
Bit 6 – *Unit Check* indicates an error condition exists in either the card device or the adapter. The nature of the error is defined by the Sense Byte information (see Sense Information). The error recovery requires either program or operator intervention. *Unit Check* is presented during initial selection or during an ending sequence with *Channel End* and/or *Device End*.

Sense Information

(Presented in generalized form to apply to ICRA and IRPA.)

The Sense Byte is transferred by the adapter in response to a *Sense* command from BDC2. It provides the information that indicates why a *Unit Check* occurred in the Status Byte.

All Sense Byte flip-flops except intervention required are cleared during the initial selection of the (ICRA) *Read and Feed;* or the (IRPA) *Read Only; Read, Feed and Select Stacker; Feed and Select Stacker,* and the *Punch,* and *Feed and Select Stacker* commands. The bits within the Sense Byte have the following meaning:



Bit 0 – The Command Reject FF is set during an initial selection when an undefined command is received by the adapter or an improper command sequence occurs. As a result, *Unit Check* will be presented in that initial selection's Status Byte, but not *Channel End* or *Device End*, because no command execution took place.

Bit 1 – The Intervention Required FF is set by a read error, or a feed check detected in the card device, or an invalid EBCDIC card code character detected by the adapter. The error requires manual intervention to correct before the Intervention Required FF can be cleared by the appropriate command.

Bit 3 – The Equipment Check FF is set when a read error is detected by the data checking logic during card reading. The read error will also cause the Intervention Required FF to be set.

Bit 4 – The Data Check FF is set when an invalid EBCDIC card code character is detected while performing an input in the EBCDIC mode. An invalid EBCDIC card code character contains more than one punch in rows 1 through 7 on the card. The set Data Check FF also causes the Intervention Required FF to be set.

Bit 7 – The Feed Error FF is set when a feed error is detected by the card device and signals this fact to the adapter by activating the MOCK signal. A feed error also causes the Intervention Required FF to be set.

Data Formats

(Presented in generalized terms to apply to ICRA and IRPA.)

Information may be either punched in or read from the cards in card image, EBCDIC card code or CSL format. The card image and EBCDIC card code formats are used in conjunction with BDC2 operations, while the CSL format is used for control storage load during deadstart operations.

A *Read* command specifying EBCDIC directs the adapter and reader to read EBCDIC-card-code punched cards, converting each EBCDIC-card-code character (a column) to an 8-bit EBCDIC character before presenting it to BDC2 (Figure 4-21A). Likewise, a *Punch* command specifying EBCDIC directs the IRPA to accept 8-bit EBCDIC data from the BDC2 and convert it to EBCDIC card code format before having it punched in the card. Any combination of punches in a single column is a valid EBCDIC card code character as long as no more than one punch exists in rows 1 through 7.

A *Read* command specifying card image directs the adapter and the reader to read binary punched cards and present the data to the BDC2 without code conversion. Since the bus-in path is only 8 bits wide the 12 bits of data per column are transferred in two sequential bytes. The odd byte (upper six rows) is transferred first followed by the even byte (lower six rows) with bits 0 and 1 of each byte being zero as shown in Figure 4-21B.

During control storage load, the adapter reads and transfers information from only six rows on the card. The data is punched four bits to a column in rows 12, 11, 0, and 1, and a nybl (4 bits) indicator is punched in either row 8 or 9. Four nybls are used by the Control Storage Load logic to generate the 16-bit Control Storage Word. The first nybl of the word contains a punch in row 8 and the fourth nybl contains a punch in row 9 (Figure 4-21C).

A card-image-mode *Punch* command causes the IRPA to accept two bytes from the BDC2 and assemble them into binary-column data before having it punched in the card. The odd byte is presented first by BDC2. In this mode of punching, the IRPA ignores bits 0 and 1 of both bytes.

LOGIC DESCRIPTION

The 4 printed-circuit modules comprised by the ICRA logic can be functionally divided into the areas shown in Figure 4-20. The ICRA logic diagrams are contained in the 7300 Processing Unit Support Diagrams manual, page through . The five modules comprised by IRPA logic are shown in Figure 4-40, and Support Diagram, pages through .

Bus Out

(Generalized)

The eight Bus Out lines connect, within the adapter, to the Address Compare logic and Command register* and to the EBCDIC converter, EBCDIC/Card Image selector and the Card Image Odd Byte register.

The Address Compare logic consists of two 4-bit comparator MSI elements whose inputs are connected to both the Bus In lines and the device address lines. The comparator element at chip location 4A deals with the upper four bits of the address and is constantly looking for an equal condition between the two inputs; (pin 3 – A=B enable signal – is a constant logic 1). When the upper four bits of the bus out information match the upper four bits of the device address, the comparator output signal enables the second comparator to look for an A=B condition.

*Support Diagram pages: ICRA, 1C042; IRPA, 1C251 and 1C241.

When the lower four bits also compare, the second comparator's output signal (ADDRESS COMPARE) goes high. The ADDRESS COMPARE signal is only used during the initial selection sequence.

The device address is determined by the wires installed in chip location 5A. The standard address assigned the card reader is changed to 04_{16}, and that for the reader/punch is $0C_{16}$. The address can be changed by simply removing or adding wires between the pins in chip location 5A.

The Command register consists of two 4-bit, bistable latch, MSI elements. The bus out information is clocked into the command register by the STROBE COMMAND signal during an initial selection sequence. The Command register holds the Command Byte, received from the BDC2 processor, while it is decoded and executed.

Two 1-of-8 decoder elements are used in the Command Decode logic. These decoders activate one of the eight outputs in accordance with the combination of the active inputs. The three inputs provide eight combinations with each corresponding to one of the output signals.

The decode logic decodes many more signals than each devices uses. This is necessary because the DG printed circuit module serves both the ICRA and IRPA. Only the signals used by each adapter are shown on its logic diagram.

As shown in Figure 4-22, the IRPA EBCDIC-to-card-code converter accepts the eight bits of bus-out data. Along with its associated 1-of-8 decoder element and inverter logic it converts the bus-out data to 12-bit card-code data before presenting it to the EBCDIC/Card-Image Data Select multiplexer (page 1C241).

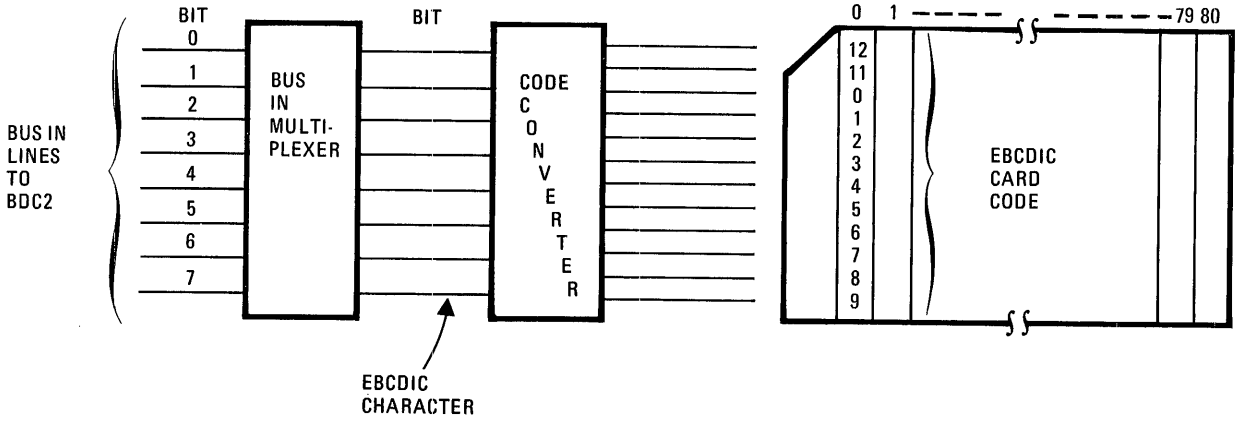
Bits 2 through 7 on the Bus Out lines also are routed to the card image Odd Byte register. It holds the first (odd) byte of the two bytes sent by BDC2 during card-image punching operations. When the processor has the second (even) byte on the Bus Out lines, all 12 bits are then routed through the EBCDIC/Card Image Data Select multiplexer to the Punch/Read Data Select multiplexer.

Bus In

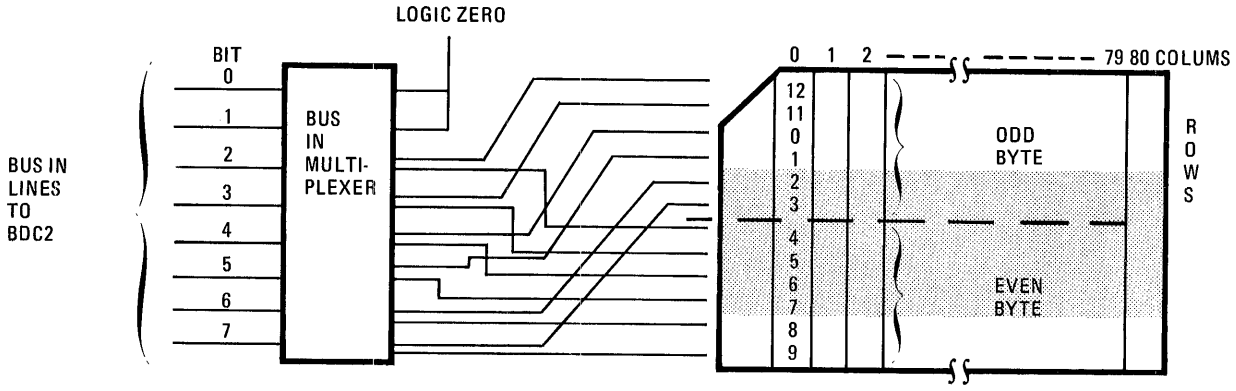
(Generalized)

The Bus In lines are connected to the adapter Bus In multiplexers through open-collector inverters. The bus-in logic and part of the control storage load logic is also located on the DG printed circuit module.

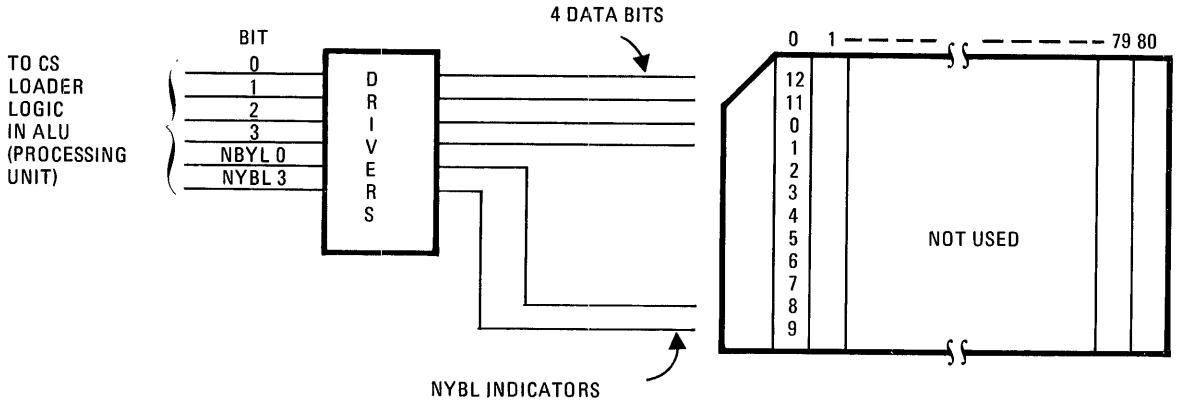
The Bus In multiplexer consists of eight 8-input multiplexer MSI elements. One of the eight inputs is routed through the multiplexer to the open-collector inverter by the three selector signals MUX2, MUX1, and MUX0.



A. EBCDIC CARD CODE FORMAT AND TRANSFER PATH



B. CARD IMAGE FORMAT AND TRANSFER PATH



C. CSL FORMAT AND TRANSFER PATH

Figure 4-21. Card Data Formats

The selector signal generating logic is located on the EP printed circuit module. The Selector Signal logic uses inputs from the Interface Timing logic, Tag Line Control logic, Reader/Punch Interface logic and the Command Decoder in determining the MPX select codes. The select codes and the corresponding selected input are listed in Table 4-14.

Table 4-14. Bus In MPX Select Codes

Select Bits			Input Selected	Input Information
MUX2	MUX1	MUX0		
0	0	0	0	EBCDIC characters from code converter
0	0	1	1	Upper half (odd byte) of card image from Buffer register
0	1	0	2	Lower half (even byte) of card image from Buffer register
0	1	1	3	Not used
1	0	0	4	Sense bits from sense logic
1	0	1	5	Not used
1	1	0	6	Card reader address from Location 5A
1	1	1	7	Status bits from status logic

The multiplexer output is enabled through the inverters to the Bus In lines only while the Tag In signal OPL IN is active. The Bus In lines are collector-OR'd, in the Basic Data Channel processor logic, with the Bus In lines from the other integrated adapters (and the external interface module when present). For this reason open-collector inverters are used to drive the Bus In lines, with the common load resistor in the BDC processor logic (page 1C141).

Buffer Register

The ICRA Buffer register consists of six dual 100-bit shift register MSI elements. These static shift registers hold the data placed in them until it is shifted out.

The 12 bits of input data from the reader are gated into the first rank of flip-flops, and the data in each rank of flip-flops gated to the next rank, each time the PHASE IN signal goes high (Figure 4-23). The outputs from the last rank of flip-flops are **not** recirculated within the register. The output is taken from the last rank of flip-flops.

This Buffer register gives the ICRA the capability of holding all the information from an 80-column card. Since all 80 columns are read and placed in the Buffer register before being transferred to the BDC, no data can be lost.

The Buffer register outputs go directly to the Bus In multiplexer, to the Code Converter logic and to the Invalid Character Checking logic.

The Character Checking logic (page 1C031) constantly monitors outputs 1 through 7 from the buffer register, looking for more than one logic "1". More than one logic "1" between bits 1 and 7 (inclusive) activates the INVALID CH signal which is used by the Sense logic (sheet 1C052). The INVALID CH signal sets the Data Check FF only if the ICRA is reading in EBCDIC mode.

The Code Converter constantly monitors the Buffer register output, and, in conjunction with part of the Character Checking logic, transfers or converts the 12-bit card code coming from the Buffer register to an 8-bit EBCDIC character. The eight-bit Code Converter output is connected to the input 0 of the Bus In multiplexer and is routed through the multiplexer only during EBCDIC read operations.

Sense Logic

The five ICRA Sense flip-flops and the associated Sense logic are contained on the EP printed circuit module (page 1C051). The Sense logic uses command signals from the Command Decode logic and inputs from the Tag Line Control logic in generating *Command Reject*. Signals from the Command Decode logic, the Tag Line Control logic and the INVALID CH signal are used in generating *Data Check*. The *Equipment Check* condition is determined by monitoring the ERROR signal from the reader interface logic and the Channel End Status FF. The Intervention Required FF monitors signals from the Reader Interface Logic, Tag Line Control logic, and the Status logic.

When any one of the five Sense flip-flops sets, the Unit Check status flip-flop sets, if the ICRA is not presently executing a *Sense* command.

NOTE

Since the EP module is also used in the Integrated Reader/Punch adapter many of the inputs have different signals on them when used in the reader/punch. Of course, only the reader signals which are applicable appear in the reader adapter logic diagrams.

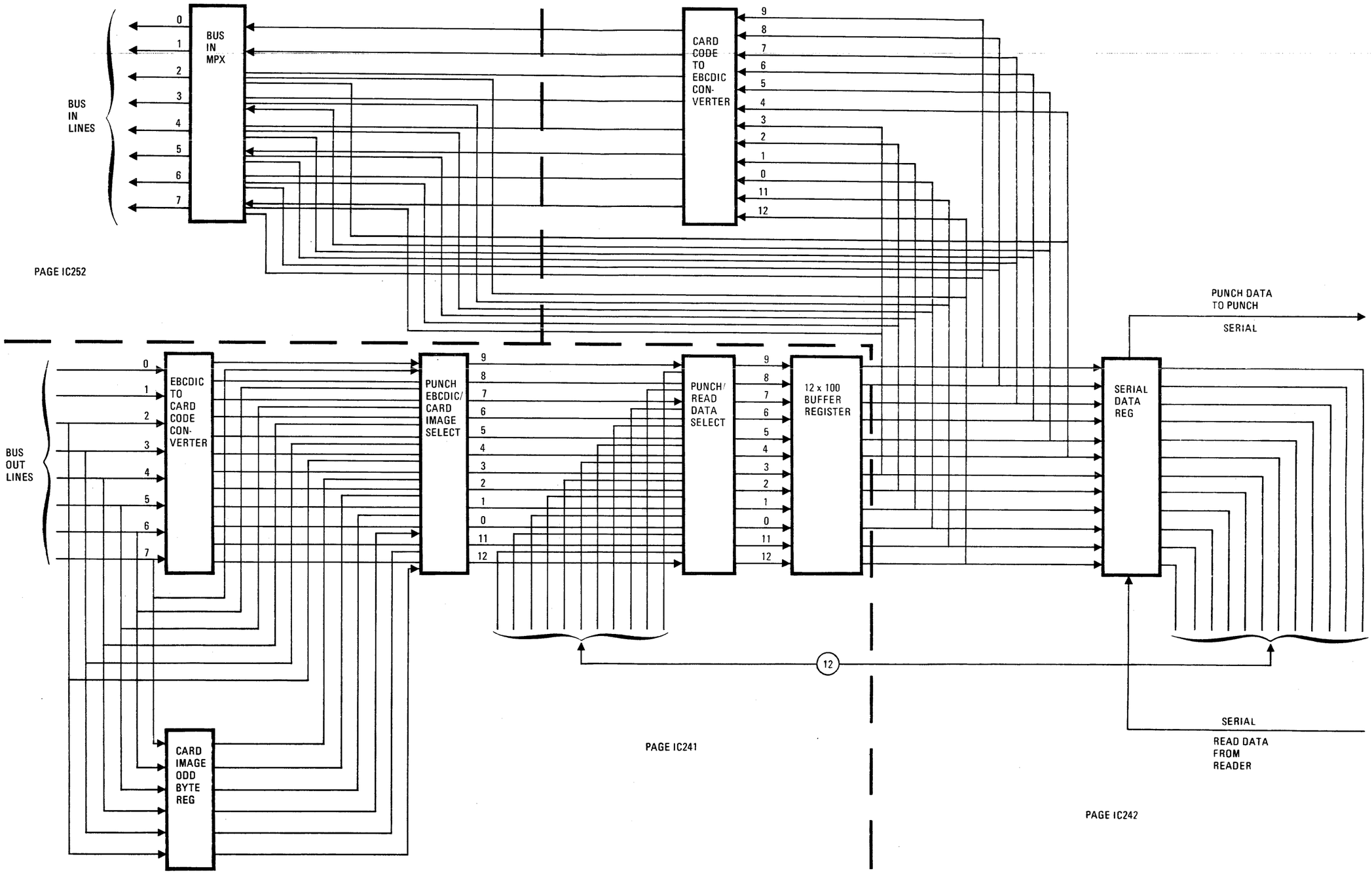


Figure 4-22. Data Paths, Adapter – BDC Processor

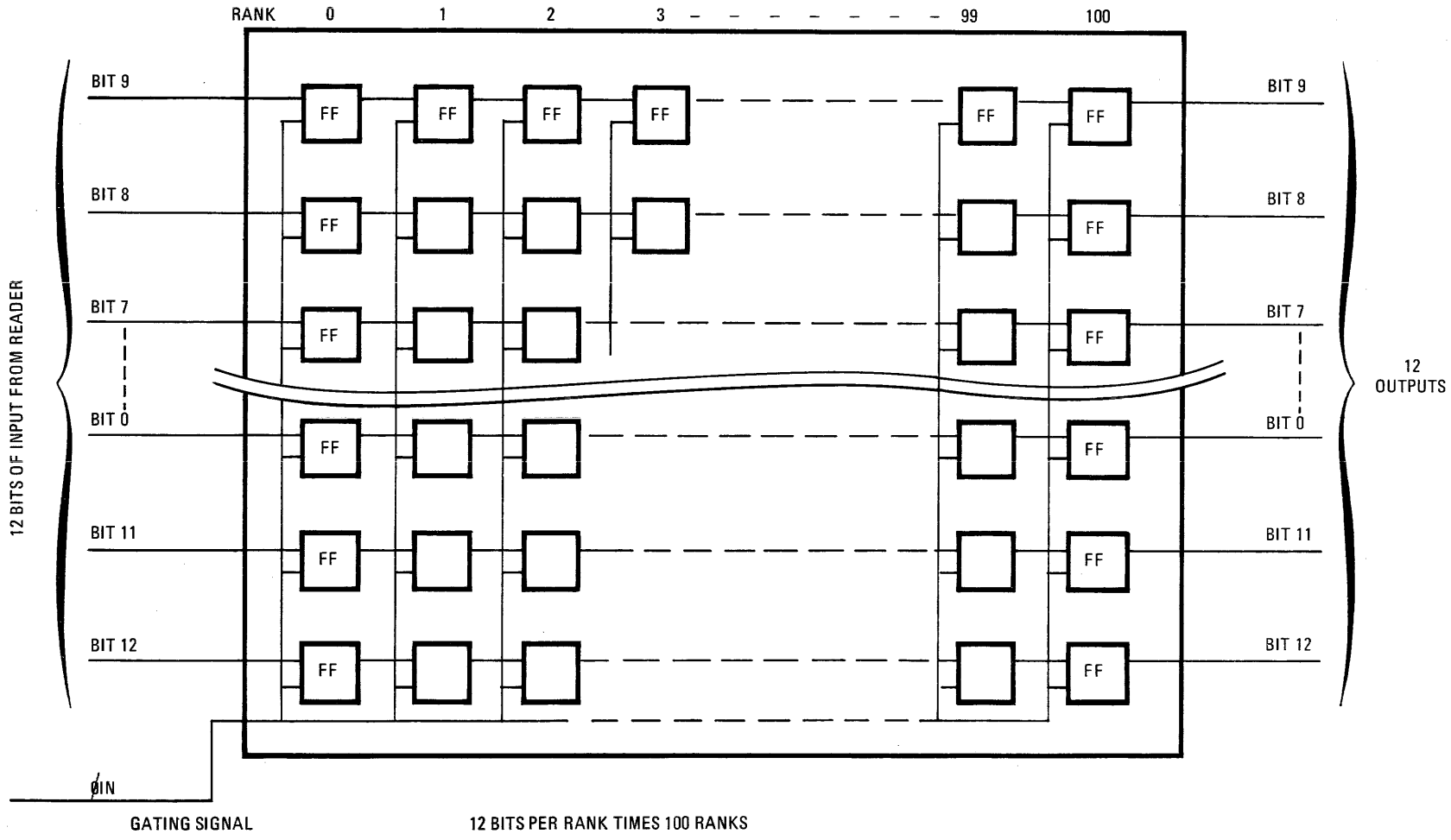


Figure 4-23. Adapter Buffer Register

Status Logic

The ICRA Status logic shown on page 1C052 contains the four Status flip-flops and associated logic. The status logic monitors signals from the Tag Line Control logic, the Reader Interface logic, the Command Decode logic and the Sense logic. The outputs from the status flip-flops are used in the generation of the *Feed* command to the reader, as well as by the Sense logic, by the Tag Line Control logic, and the Reader Interface logic.

Reader Interface Logic

The ICRA Reader Interface logic, shown on page 1C032, contains the five receivers and one transmitter which deal with the card reader interface signals, and the logic controlling the proper loading and unloading of the Buffer register. The following is a brief description of the Card Reader Control signals shown on page 1C032.

- READY: This signal when active indicates the reader has **no** error conditions. The ready condition includes the following elements:

  Cards in the input magazine (hopper)

  Output stacker not full

  Power applied

  START pushbutton pressed.

- BUSY: This signal is activated by the reader from the time the leading edge of a card enters the Read station until the final light check is made at the card's trailing edge.
- INDEX MARK: This signal is active during the time the data from the reader is stable and can be accepted by the ICRA.
- MOCK: This signal indicates a card failed to enter or leave the card track properly. This could be caused by the failure to pick a card from the input magazine (hopper) or by the failure of a card to completely clear the card track.
- ERROR: This signal is activated by the failure of either the light or dark check, and thus signals the fact that a read error has occurred.
- PICK COMMAND (PC): This signal when activated by the ICRA starts a card read cycle, if the reader is READY.

In addition to these reader control signals the reader Interface logic (page CC032) receives a POWER ON MASTER CLEAR (PWRONMC) signal from the Power Sequencer unit. This signal is activated at the beginning of the power up sequence and is terminated once all equipments in the system have been brought up to full power.

The Buffer Register Control and the Reader Interface Control logic consist of a two-phase timing chain, an 8-bit count-down counter, five control flip-flops and associated logic.

The Column counter is implemented by the use of two 4-bit counter MSI elements. The counter is preset to the binary equivalent of either 78 or 99 by the 5 microsecond pulse out of the Preset one-shot circuit. The count is decreased by 1 for each COUNT-DOWN pulse received. The next count-down pulse received after the count has reached zero produces a BORROW OUT signal which clocks the Load Complete (LC) flip-flop.

The PHASE IN and COUNT-DOWN pulses are generated by either the INDEX MARK signal from the reader or the timing chain. The timing chain consists of two one-shot circuits. When enabled, the first one-shot (Ø1) produces a 500 nanosecond pulse (Figure 4-24) which drives both the PHASE IN signal and the second one-shot (Ø2).

As stated earlier, the trailing edge of the PHASE IN signal gates the data from the data lines into the Buffer register and shifts the data right one rank.

The trailing edge of the Ø1 pulse triggers the second one-shot (Ø2) which generates a 1 microsecond COUNT-DOWN pulse. The trailing edge of the Ø2 pulse will trigger the first one-shot if it is still enabled. Each pass through the chain moves the data right one place in the card Buffer register and reduces the count in the Buffer Counter by 1.

Tag Line Control

(Generalized discussion relevant to both ICRA and IRPA.)

Printed circuit module DD contains all the Tag Line Control logic as shown*. The second page contains ten control flip-flops, two deskew one-shot circuits and inverter logic. This logic monitors signals from the Status and Sense logic and the Tag In lines to determine conditions such as *Stack Status, Last Byte Card Image, Status Pending* and others. This logic also controls three Tag In lines, ADRS IN, SRV IN and STAT IN, during operational sequence execution.

Shown on the other page of the DD module logic is the Reset and Disconnect logic, the Select Latch, the Tag Line Timing Chain and associated Inverter logic. This control logic monitors all the Tag In lines and the Address Compare logic in controlling the adapter during the different operational sequences. It generates and sends control signals to all the adapter modules, controls the propagation of SEL IN along the selection loop, generates timing signals used by the adapter during the various operational sequences, and controls the OPL IN and REQ IN Tag In lines.

*Support Diagram pages: ICRA, 1C061/62; IRPA, 1C271/72.

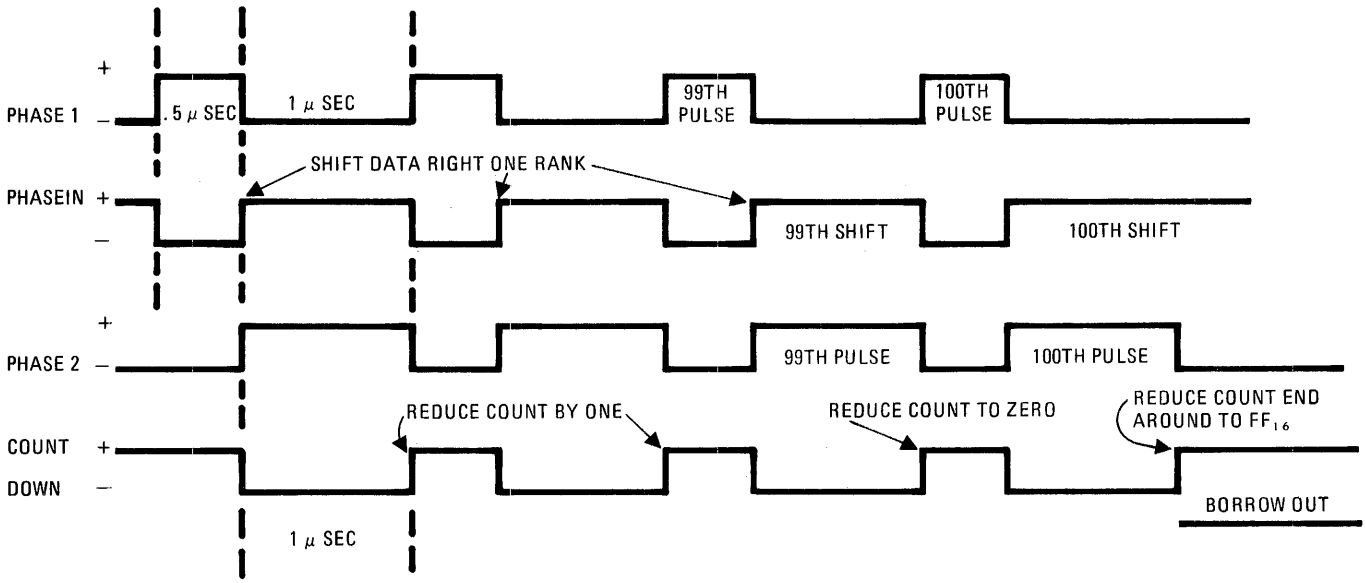


Figure 4-24. Adapter Timing Chain Output

The Sel Latch FF sets when both SEL IN and HLD OUT go active and, when set, enables the SEL IN signal to the next adapter; if the adapter does not have an active REQ IN line, the Opl In FF is clear and the Init Sel FF is clear. The Sel Latch FF will stay set until the HLD OUT signal goes inactive regardless of changes of the SEL IN signal.

A system reset (SYS RST) is recognized by the ICR whenever both OPL OUT and SUP OUT are inactive.

A selective reset (SEL RST) is recognized only while the adapter is selected and has OPL IN active. When SUP OUT goes active and OPL OUT terminates, with OPL IN active the Sel Rst FF sets. The resulting RESET signal within the adapter terminates the OPL IN signal, but the Sel Rst FF is not cleared until SUP OUT is terminated.

If either of the RESET signal sequences is recognized, a CLEAR is generated, which clears all control flip-flops on the DD and EP modules, terminates all signals to the BDC.

The Intf Disc FF becomes set during an operational sequence when ADRS OUT is active, and the adapter is selected (OPL IN active, and HLD OUT terminated by BDC2). HLD OUT going inactive clears the Sel Latch FF which in turn sets the Intf Disc FF and does not allow it cleared until the Sel Latch FF is again set. The Intf Disc FF set disables the OPL IN Tag Line to the BDC2.

Tag Line Timing Chain

(Generalized discussion relevant to both ICRA and IRPA.)

During operational sequences, the adapter requires timing signals in developing the proper activation of the Tag In lines. On the first logic page is shown the Tag Line Timing logic which consists of a five flip-flop (edge-triggered) chain controlled by a delay-line oscillator.

When enabled, the oscillator's output is a 140-nanosecond clock pulse as shown in Figure 4-25. The five flip-flops are set and cleared in order, producing the five timing signals T1 to T5.

Starting at zero (all five flip-flops clear) the oscillator clocks the flip-flops until T4 is set. At this point the oscillator is disabled waiting for a Tag In signal response from the processor. Upon receipt of a response, the oscillator is allowed to advance the flip-flops through T5 back to zero and then is disabled again.

Each complete pass through the timing logic — zero back to zero again — is in response to a Tag In signal going active and then inactive. The timing signals are used by the ICRA to enable Tag In signals, the Bus In Multiplexer Select logic, and the Status and Sense logic.

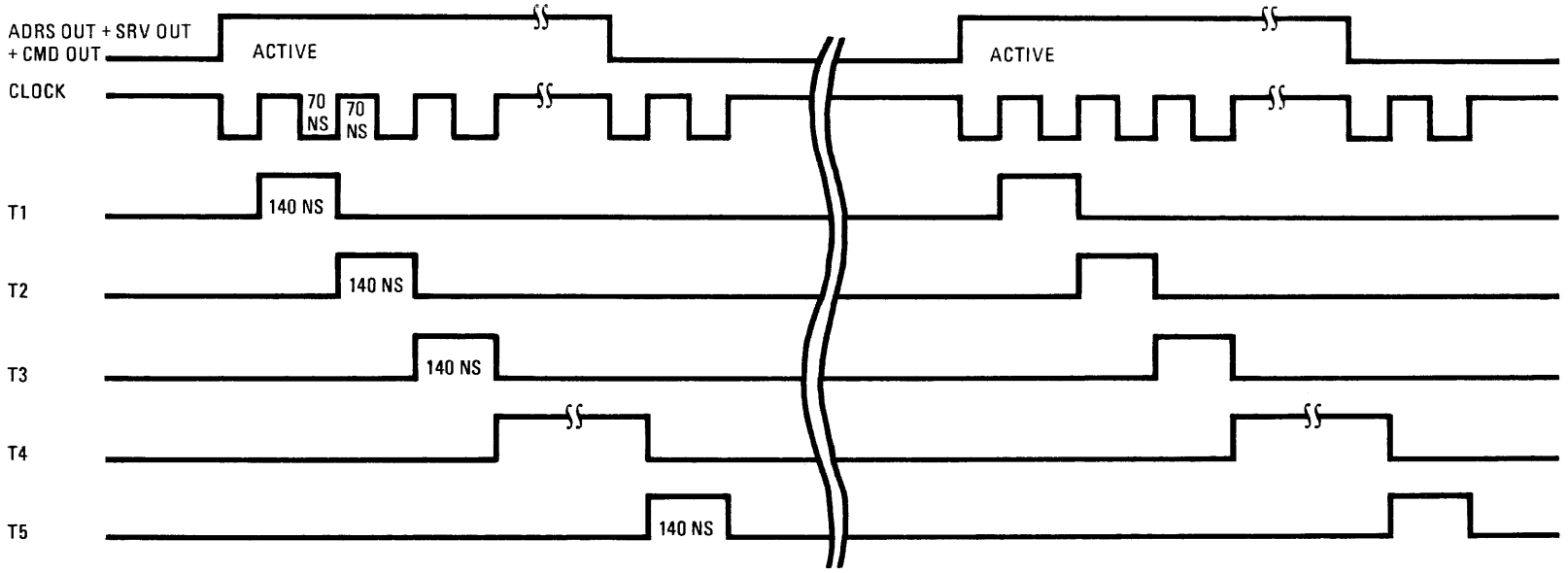


Figure 4-25. Timing Chart, Adapter Tag Line

OPERATING SEQUENCES

The operating sequences which the ICRA performs can be divided into three groups:

- all the operational sequences required to communicate with the BDC.
- read and card load sequence
- control storage load.

Card Load Sequence

During a card load sequence the data on a punched card is read, transferred to the ICRA, and loaded into the Buffer register. During this sequence the card reader is checking for improper operation, which if sensed, it indicates to the ICRA with either the MOCK or the ERROR signal.

A card load sequence is performed for the initial card for for each subsequent card during an input operation as shown in Figure 4-26. The initial card load is performed as soon as the operator readies the card reader. This is accomplished by placing cards in the input hopper and pressing the START switch.

Initial Card Load

An initial card load sequence is started in response to the READY signal from the card reader (Figure 4-27). The READY signal triggers a one-microsecond one-shot circuit which clocks the Feed FF and clears the Load Complete FF. The Feed FF sets if the BUSY and the MOCK signals from the reader are inactive, indicating the reader is not presently reading a card. The Feed FF actives the PICK COMMAND signal to the reader, starting a card read cycle. The Feed FF also triggers the Preset one-shot circuit, the output of which forces the Buffer Counter to 99 if the Block Feed FF is clear.

When the card enters the Read station, the BUSY signal is activated. The data transfer starts with signals from the first row of data being placed on the input lines by the reader. The INDEX MARK signal is activated and then terminated by the reader, generating the PHASE IN and COUNT-DOWN signals within the ICRA. The trailing edge of these signals gates the data into the Buffer register and reduces the buffer count by 1.

After the above events occur 80 times, the buffer count is reduced to 19, and the card data resides between ranks 1 and 80 of the Buffer register.

The card reader terminates the BUSY signal as the card leaves the Read station. Termination of the signal triggers the timing chain causing it to generate 20 PHASE IN and COUNT-DOWN pulses. The trailing edge of the first 02 pulse sets the Data Ready FF. Nineteen COUNT-DOWN pulses reduce the buffer count to zero. The 20th pulse produces a borrow out of the high-order end of the counter which disables the timing chain by setting the Load Complete FF. The 20 PHASE IN pulses move the data in the Buffer register to ranks 21 through 100 in preparation for the output to the BDC. Column 1 data from the card is now at the output of the Buffer register.

Load Complete going set triggers the Preset one-shot circuit which this time forces the buffer counter to 78 since the Block Feed FF is now set. This is in preparation for the output to the BDC. Only 79 shifts are required to transfer the 80 ranks of data since the output of rank 100 is constantly available. The count is set to 78 so that for COUNT-DOWN pulse 79 (all data transferred) the counter produces a BORROW OUT signal.

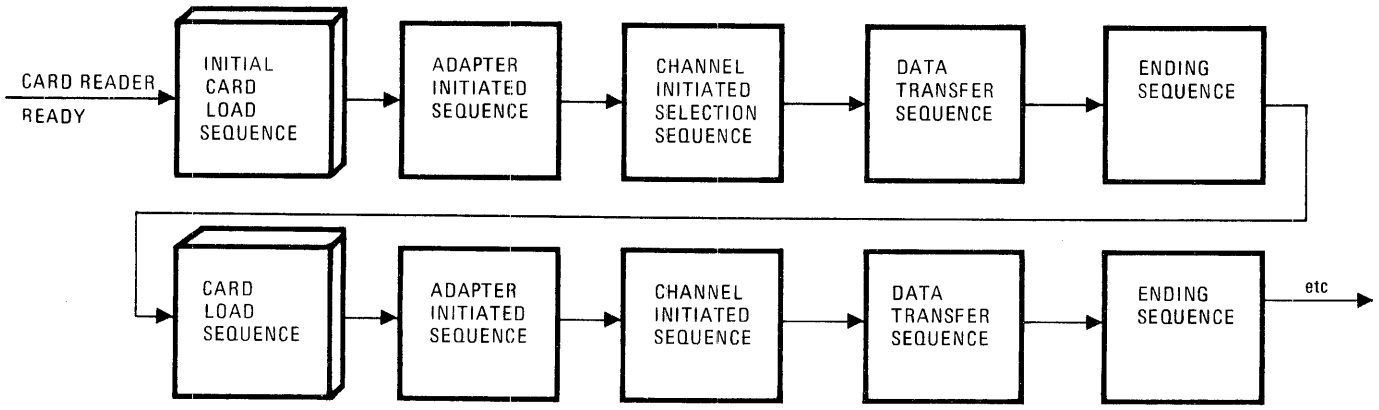


Figure 4-26. ICRA Operating Sequences

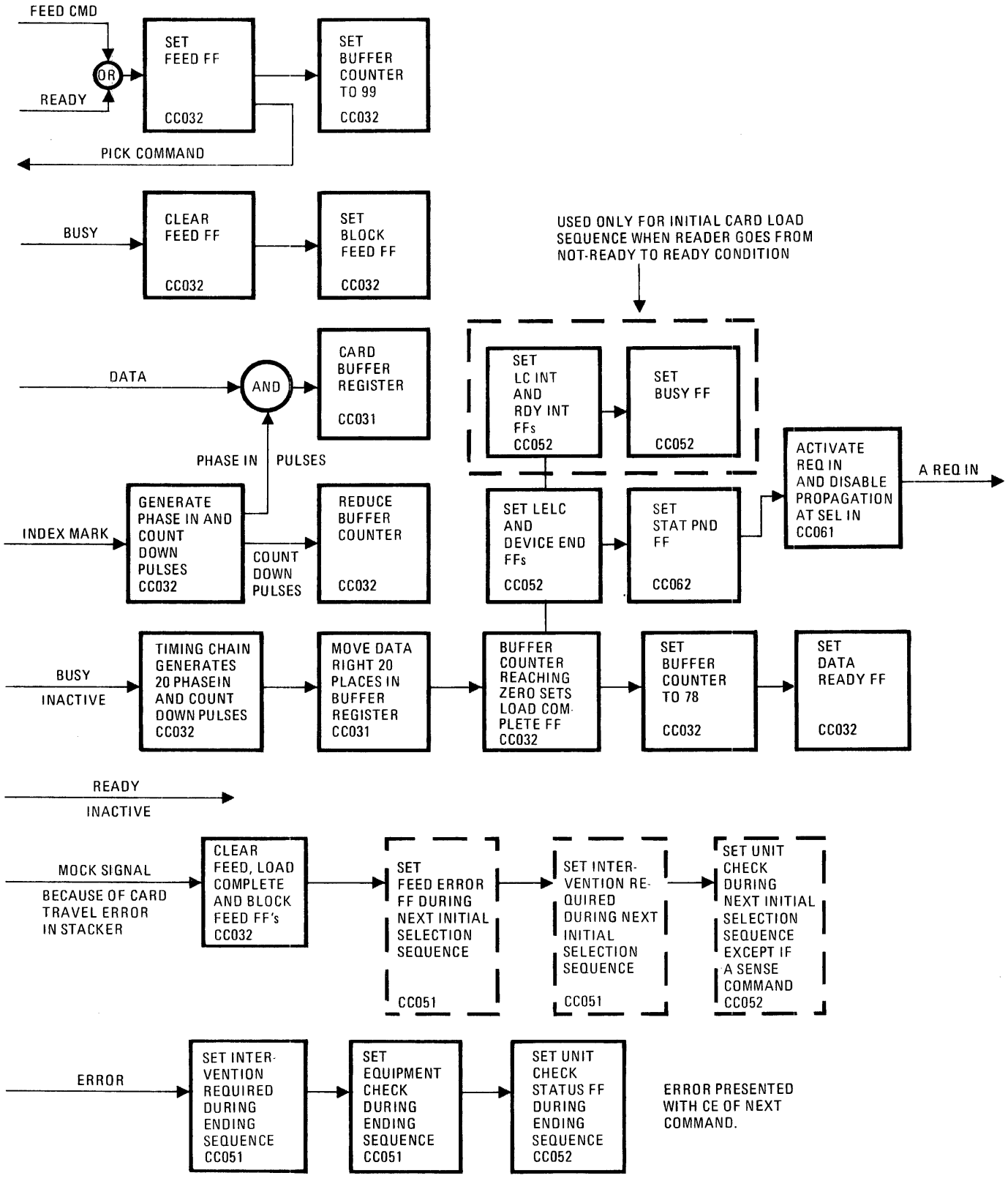


Figure 4-27. ICRA Card Load Sequence

Load Complete set indicates to the BDC Tag Line Control logic that a card has been read, loaded, shifted and the Reader Interface logic is ready. Since the ICRA can now accept and execute *Read* commands it notifies the processor of this fact. The following events are performed in accomplishing this task.

Load Complete setting sets the Leading Edge Load Complete FF (LELC) which in turn sets the Device End FF. Status indication is now waiting acceptance (outstanding status) by the processor; for this reason the ICRA uses Device End set output to set the Stat Pnd FF which activates the REQ IN signal to the processor. REQ IN active also disables the propagation of the SEL IN signal through the ICRA.

In addition, during only an initial card load the Load Complete setting also sets Load Complete Interrupt (LC INT) and Ready Interrupt (RDY INT). Rdy Int setting sets the Busy FF indicating that the reader and the ICRA have gone from a not-ready to a ready state. Busy, when set, clears Rdy Int after a short delay, and it will not be set again until the reader goes ready from a not-ready state.

Here the ICRA waits until the processor responds to the REQ IN signal with a polling routine, which effects an adapter-initiated sequence between the ICRA and the processor. It is during this sequence that the status indication is presented to tell that the ICRA is ready.

Reader Error Signals

The reader, detecting any error during a card read cycle, immediately terminates the READY signal to the ICRA and terminates reader operation. The inactive READY signal prevents the ICRA from generating another *pick* command to the reader. The MOCK signal, activated because of a failure to pick a card, clears the Feed FF in the ICRA. If the failure to pick occurs on an initial card load, no adapter-initiated sequence is started since BUSY is not set (as shown in Figure 4-28A). If the failure to pick occurs on any subsequent card load, the absence of READY AND's with BUSY to start an adapter-initiated sequence (Figure 4-28B).

If the MOCK signal is activated because of a card motion failure in the output stacker, the card has been read and the data already placed in the buffer register; the ICRA in response clears control flip-flops and sets Sense and Status flip-flops as indicated in Figure 4-28.

For any card read error, the reader activates ERROR in addition to terminating READY. The ERROR signal is not used by the ICRA until after the card data has been transferred to processor 2 and the ending sequence is started. At this time the Unit Check status FF and the Intervention Required and Equipment Check sense FF's are set.

Single Card Read

One card in the input hopper can be read into the Buffer register and transferred to processor 2 even though the reader terminates READY as soon as the card leaves. This is possible since the ICRA does not allow the inactive READY signal to set the Device End FF until the Block Feed FF clears. Block Feed sets as soon as the reader becomes busy and is not cleared until the data has been transferred to processor 2. Device End setting sets the Stat Pnd FF and in turn activates REQ IN which is the beginning of an adapter-initiated sequence.

Adapter-Initiated Sequence

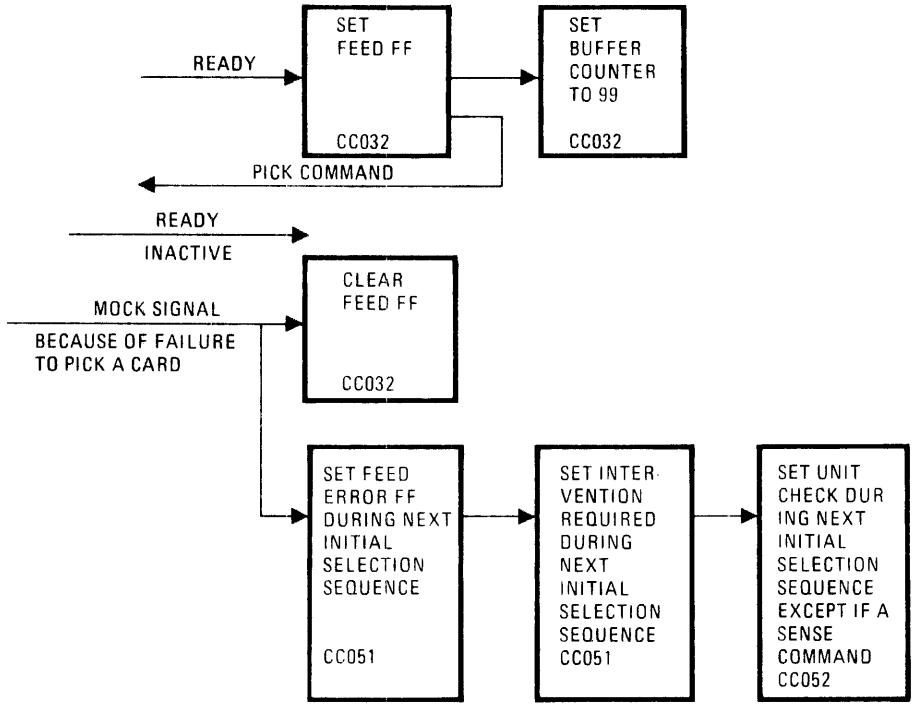
Processor 2 in response to the REQ IN signal performs a polling routine during which the ICRA performs the events within the adapter initiated sequence as shown in Figure 4-29. During this sequence the ICRA presents its address and status indications. If the status indication is accepted by processor 2, the ICRA is deselected and waits for a command via a channel initiated initial selection.

If the status indication is stacked, rather than accepted, the ICRA still is deselected at the end of the sequence, but holds the REQ IN signal active and continues to disable the propagation of SEL IN. The REQ IN will force processor 2 into performing another polling routine. The ICRA will continue to request service with the REQ IN signal until the status indication is either accepted or suppressed by processor 2 during an adapter-initiated sequence.

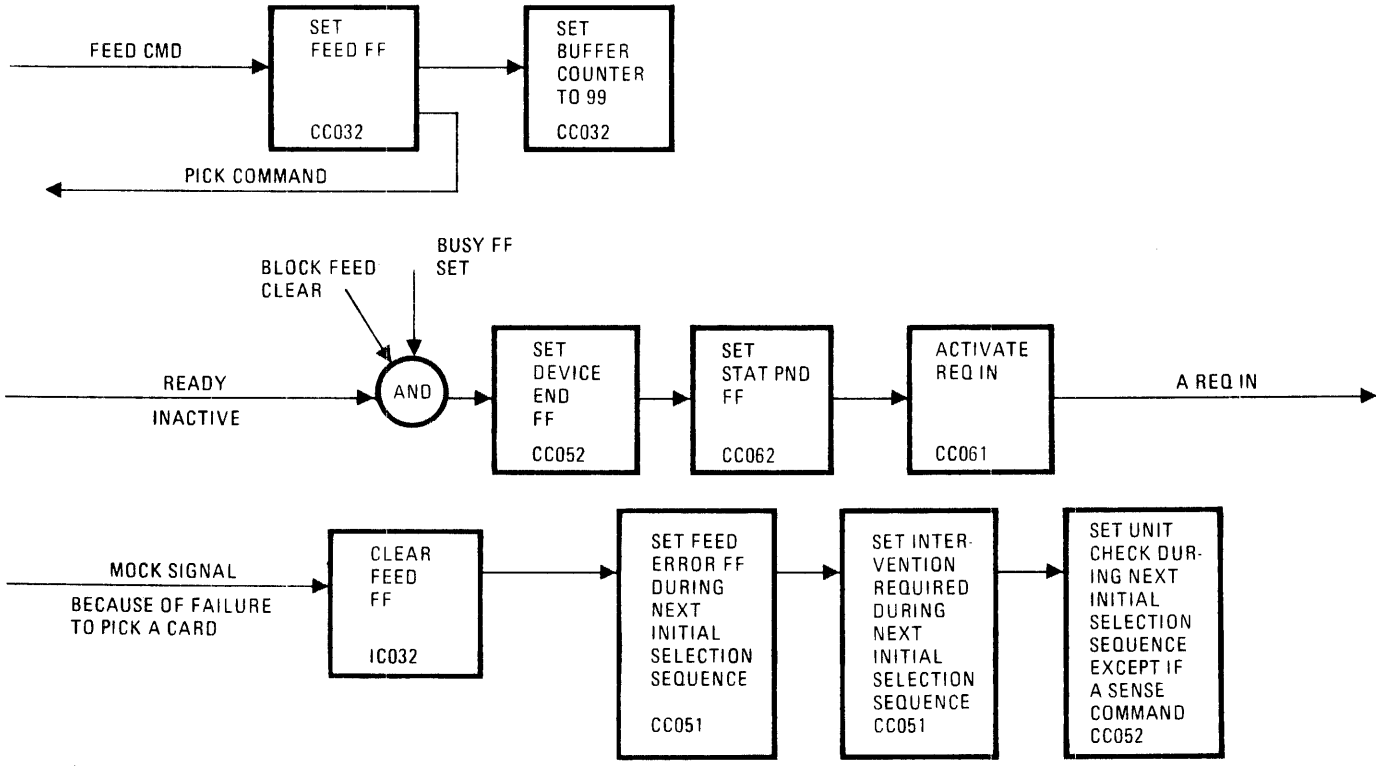
Initial Selection

(Generalized discussion relevant to ICRA and IRPA.)

Once the status indication has been accepted, processor 2 may start an input operation with an initial selection sequence during which the adapter responds to the processor TAG OUT signals and device address by interpreting the Command Byte and presenting another status indication. The adapter develops all control signals following processor requirements. The following discussion presents, in the order of occurrence, the events within the adapter during an initial selection sequence. Appendix 4A has detailed timing.



A. INITIAL CARD LOAD ATTEMPT



B. SUBSEQUENT CARD LOAD ATTEMPT

Figure 4-28. Card Load Attempt Sequences

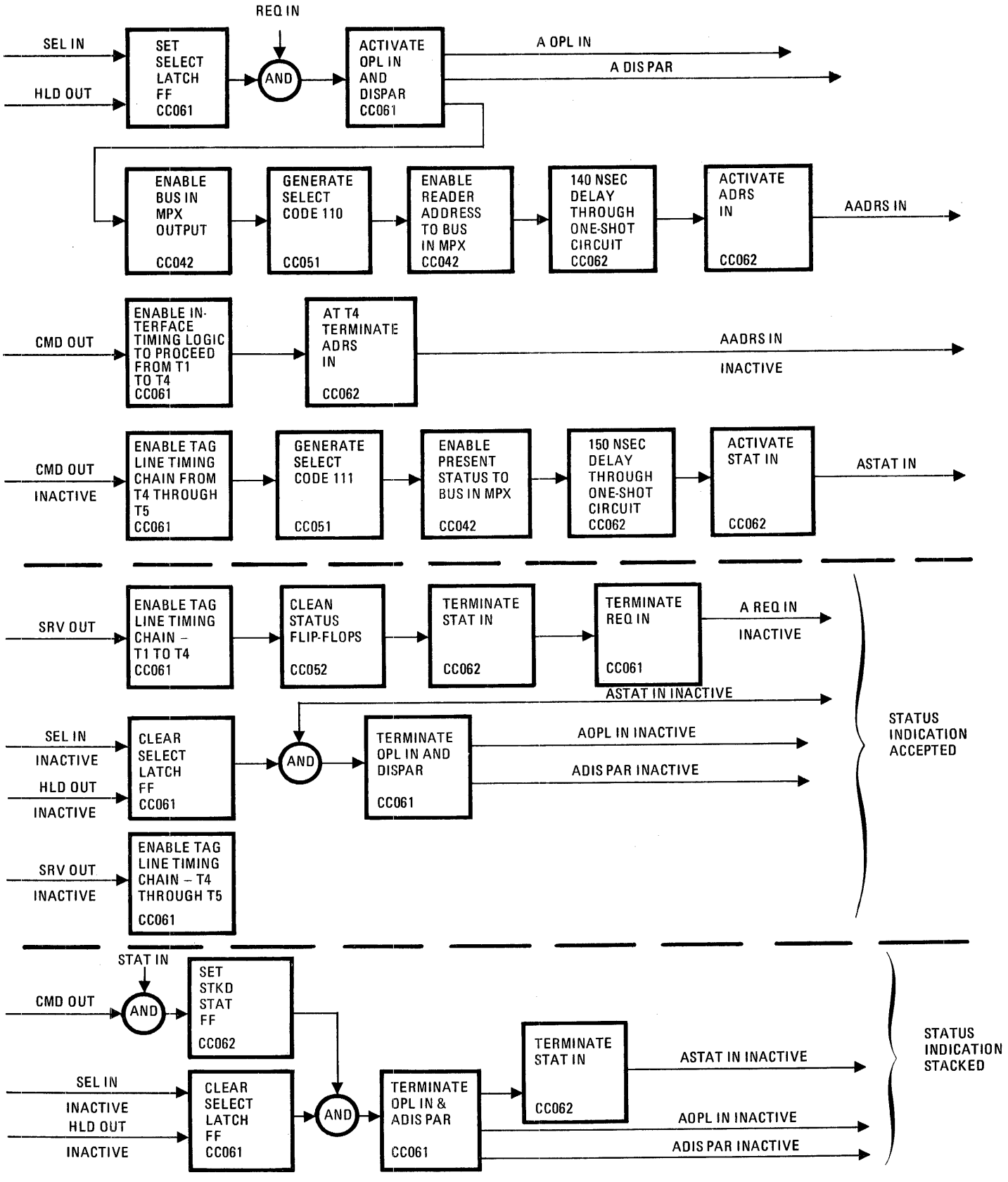


Figure 4-29. ICRA-Initiated Sequence

Address

Tag Out signal ADRS OUT AND's with OPL OUT, enabling the Tag Line Timing Chain, previously described. If the device address compares with the address on the Bus Out lines, the ADRS CMP signal sets the Initial Selection FF, preventing the CR SEL IN signal from propagating back to the bypass logic. The Sel Latch FF sets and then ANDs with INITIAL SELECTION, to activate the AOPL IN and ADISPAR signals to the processor logic and to enable the multiplexer output to the Bus In lines. The Tag Line Timing Chain stops at T4 waiting for processor 2 to respond to AOPL IN by terminating ADRS OUT (Figure 4-30).

The inactive ADRS OUT enables the Tag Line Timing Chain to complete T4 and to advance through T5 to zero. The inactive ADRS OUT ANDs with OPL IN to force (through the setting of the Address Enable FF) the selector logic to generate the select code for the reader/punch address. The Address Enable FF also triggers a 150-nanosecond one-shot circuit. The trailing edge of the 150-nanosecond pulse activates the AADDRESS IN signal to the processor.

Command

Processor 2 responds, after address checking, by placing the Command Byte on the Bus Out lines and activating CMD OUT. The adapter responds to CMD OUT by enabling the Tag Line Timing Chain and generating the COMMAND STROBE signal, unless it is busy and has not reached CE or DE point. COMMAND STROBE gates the Command Byte into the Command register. The Command Decode logic decodes the Command register contents and activates the appropriate signal or signals. If the command is not a *No-Op, Test I/O,* or *Sense*, the Sense flip-flops are cleared at T2. CMD OUT also ANDS with T4 to disable the ADDRESS IN signal.

At this point, if the Command Byte is not one of the four legal ICRA commands, the Command Reject FF will set. If the Buffer register is not full and the ICRA is not busy, the Intervention Required FF will also set. The Unit Check FF will set if any one or more of the sense flip-flops is set.

If the Command Byte is not one of the seven IRPA commands which the reader/punch will recognize, or if the command sequence is incorrect and the IRPA is not busy, the Command Reject FF is set. Likewise at this time if the IRPA is not busy and the reader/punch does not have the READY signal active, the Intervention Required FF is set. The Unit Check FF sets when any of the sense flip-flops set, except for Sense command.

Status

In response to ADDRESS IN, CMD OUT is terminated by processor 2. CMD OUT going inactive enables the Tag Line Timing logic and sets the Service Status FF which ANDS with the Status Pending FF. The result is to force the selector logic to generate the select code for device status indication and to trigger a 150-nanosecond one-shot circuit. The trailing edge of the 150-nanosecond pulse activates the ASTAT IN signal, thus giving the Status Byte time to stabilize on the Bus In lines.

After the status indication has been accepted by processor 2, the processor displays this fact by activating SRV OUT. The ICRA uses SRV OUT to enable the Tag Line Timing Chain. At the end of T1 time the Init Sel FF clears. Init Sel clearing sets the Busy 1 FF if the Unit Check and Device End FF's are clear, indicating no error condition exists. At T3 time the Unit Check, Device End, and Channel End FF's are cleared. At T4 the ASTAT IN signal is terminated, forcing the Selector Signal logic to generate the select code for EBCDIC characters from the converter.

Processor 2 responds by ending the initial selection sequence and begins the data transfer sequence by terminating SRV OUT.

NOTE

Incorrect status causes processor 2 to terminate SEL IN and HLD OUT, deselecting the adapter at the completion of the initial selection.

Data Transfer Sequence

During an ICRA data transfer sequence the 80 ranks of data are shifted out of the Buffer register and sent to the processor via the Bus In lines. The number of bytes transferred depends on the *Read* command. For a card image *Read* there are two bytes per rank transferred, whereas for an EBCDIC *Read* only one byte per rank is transferred. The events within the ICRA during an EBCDIC *Read* data transfer sequence are presented in both the following text and Figure 4-31.

The conclusion of the initial selection (and the beginning of the data transfer) is indicated with the termination of SRV OUT by the processor. SRV OUT going inactive enables the Tag Line Timing Chain and sets the Performing Command FF, which stays set until the data transfer sequence terminates. SRV OUT inactive also AND's with DATA READY and then proceeds through a 600-nanosecond delay before setting the Data Xfr Req FF. The set output of the Data Xfr Req FF in turn triggers a 150-nanosecond one-shot circuit. The trailing edge of the one-shot output pulse sets the Tag En FF, which in turn activates the SRV IN signal to the processor.

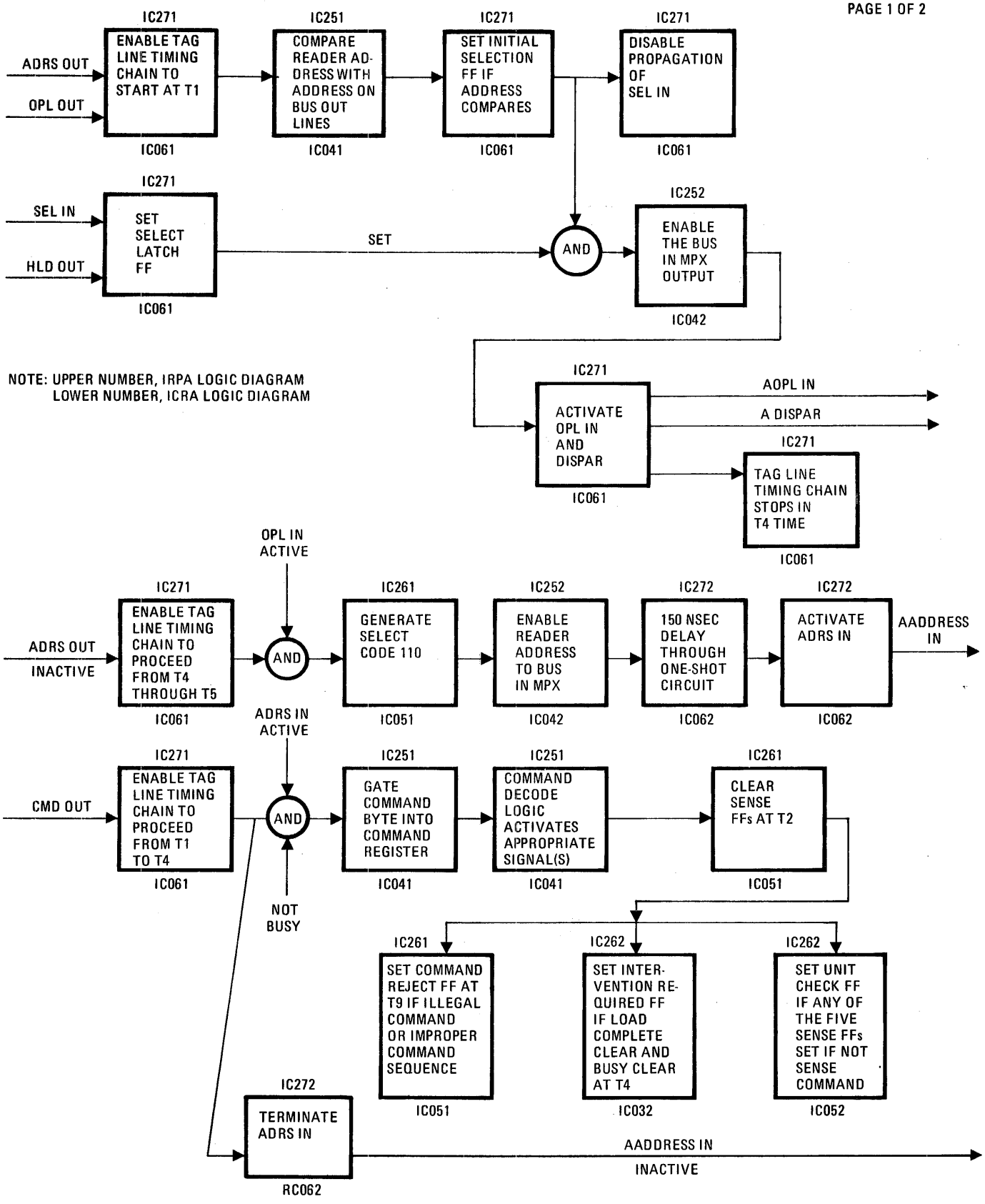


Figure 4-30. Adapter Initial Selection Sequence

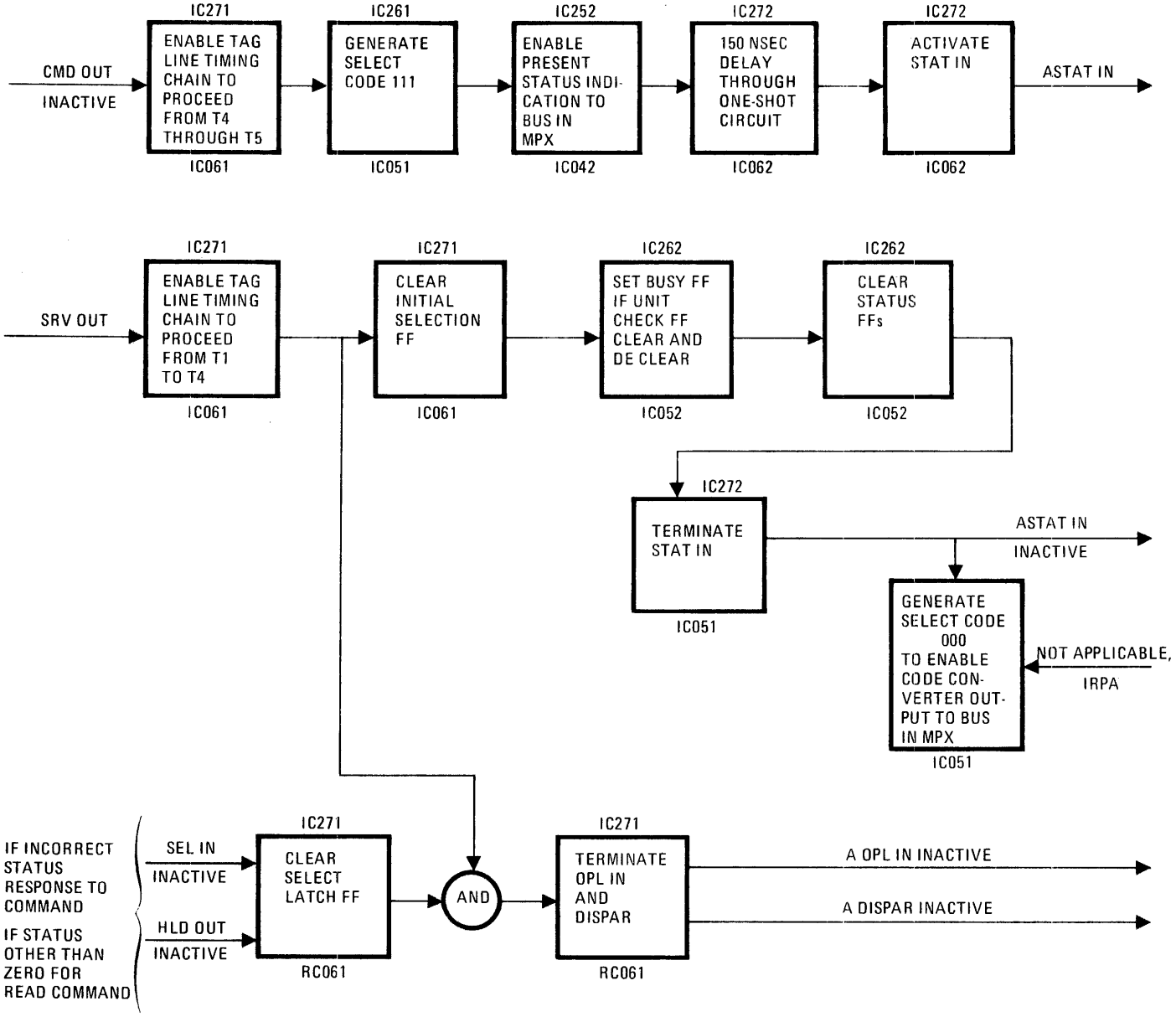


Figure 4-30. Adapter Initial Selection Sequence (continued)

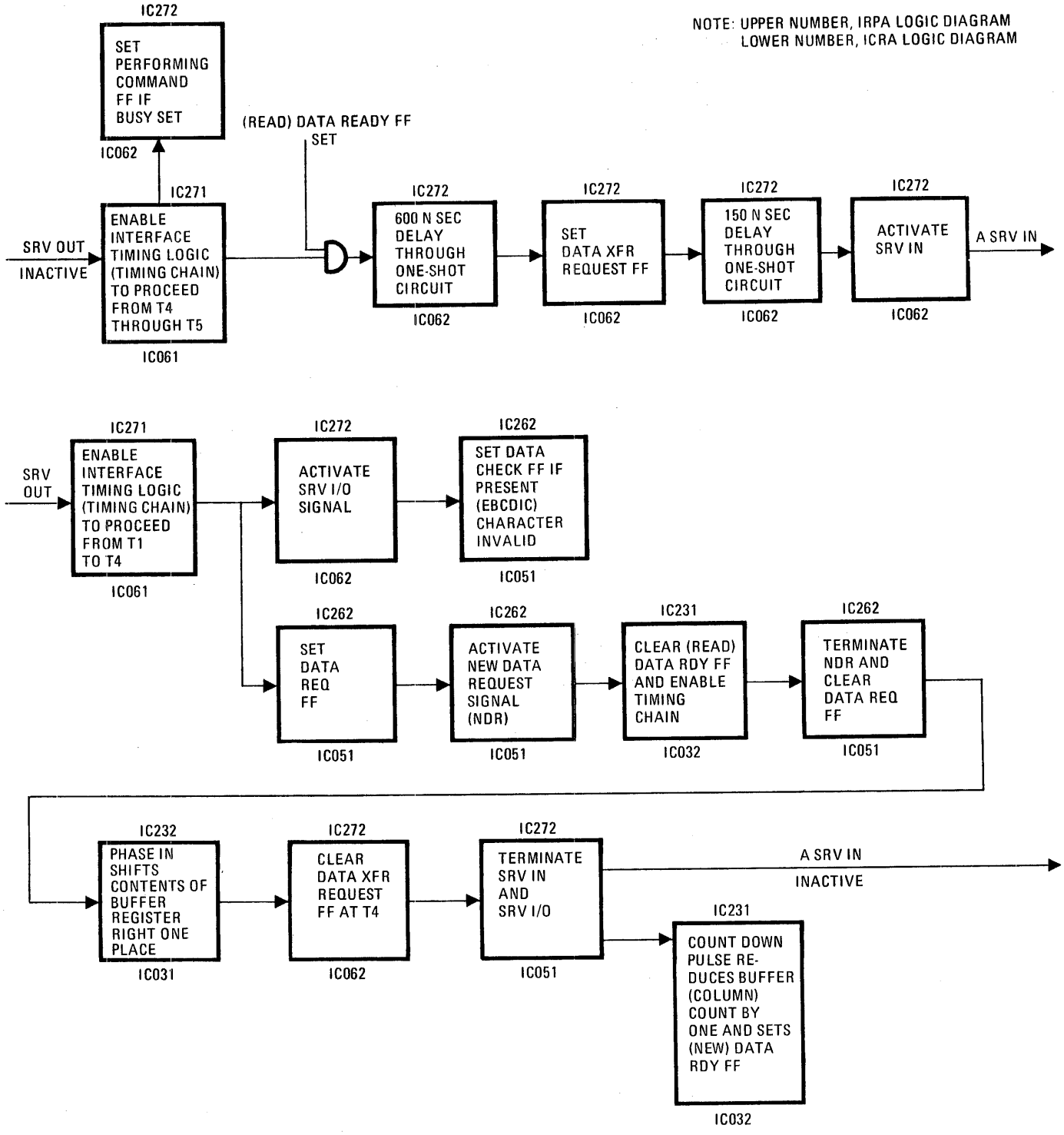


Figure 4-31. Data Transfer Events for One EBCDIC Character

At this point the first converted EBCDIC character is on the Bus In lines. SRV IN is active and the ICRA is waiting for processor 2 to accept the byte and respond.

After accepting the byte, the processor responds by activating SRV OUT. The ICRA uses SRV OUT to prepare the next EBCDIC character for transfer, and then indicates this fact by terminating SRV IN. In doing so, the contents of the Buffer register are right-shifted one place and the count in the Buffer counter decreased by one. The following sequence of events within the ICRA accomplishes the preparation of the next character for transfer. Figure 4-32 shows the timing relationship of the events involved.

- SRV OUT activated by processor 2
- Activate SRV I/O signal
- Set Data Check FF
- Set Data Request FF
- Activate NEW DATA REQUEST (NDR) signal which enables the input to the two-phase timing chain
- Clear Data Ready FF
- Terminate NDR; disable input to timing chain
- Clear Data Request FF
- Shift contents of Buffer register right one place
- Clear Data Xfr Req FF
- Terminate SRV IN and SRV I/O
- Reduce Buffer counter by one
- Set Data Ready FF

As shown in Figure 4-32, the events between SRV OUT going inactive to SRV OUT again inactive are repeated for each EBCDIC character transferred. The data transfer sequence continues until either processor 2 terminates it with CMD OUT, or the Buffer counter reaches a count of zero.

Ending Sequence

(Generalized discussion relevant to both ICRA and IRPA.)

During the ending sequence the adapter status indication is first presented to the processor and then the adapter is deselected. Regardless of whether the processor or the adapter terminates the data transfer, the ending sequence immediately follows the conclusion of that data transfer sequence.

Processor-Terminated Input

The processor can terminate either an input or output data transfer in either mode before all 80 columns of information have been transferred. The activation of CMD OUT in response to SRV IN forces the adapter into an ending sequence (regardless of whether during the odd or even byte portion of a card image transfer). Figure 4-33 shows the sequence of events within the adapter for a channel-terminated input transfer.

The adapter AND's CMD OUT with SRV IN to stop the data transfer and prepare for the ending sequence. It immediately sets the Channel End FF, whose output ANDs with the signal indicating that a *Read* command is being performed, to generate the SHORT BUFFER signal. SHORT BUFFER clears the Load Complete FF, which clears the Block Feed FF. At T4 time the Data Xfr Req FF clears, which terminates the SRV IN signal.

The processor responds by terminating CMD OUT which the adapter uses to start the ending sequence. First the adapter clears the Prf Cmd FF, as the performance portion of the command is now complete, and then sets the Status Pending FF which, when set, causes the Select Code logic to enable the status information through the MPX to the Bus In lines. After a 150-nanosecond delay the adapter activates STAT IN.

Once the processor has accepted the status indication, it responds by activating SRV OUT and deselects the adapter by terminating SEL IN and HLD OUT. The status indication may also at this point be stacked. This would be accomplished by a response to STAT IN of CMD OUT rather than SRV OUT.

Status Indication Accepted – The ICRA uses SRV OUT in response to STAT IN to start a Card Load sequence and to complete its portion of the deselection. The ICRA accomplishes this by generating the FEED CMD signal from the active SRV OUT, if no *Unit Check* conditions exist. The FEED CMD signal sets the Pick FF which generates the PICK command to the reader (starting a read cycle) and triggers the Preset one-shot circuit, which sets the Column Counter to 99.

In completing the deselection, the ICRA clears the status flip-flops – DE, CE and Unit Check – and at T4 clears the Status Pending FF, terminating STAT IN. Also the ICRA in response to HLD OUT going inactive will have cleared the Select Latch FF. Now, with both the Status Pending and Select Latch FF's clear, the ICRA terminates both OPL IN and DISPAR and allows the propagation of SEL IN.

As the last event in the ending sequence, the processor terminates SRV OUT.

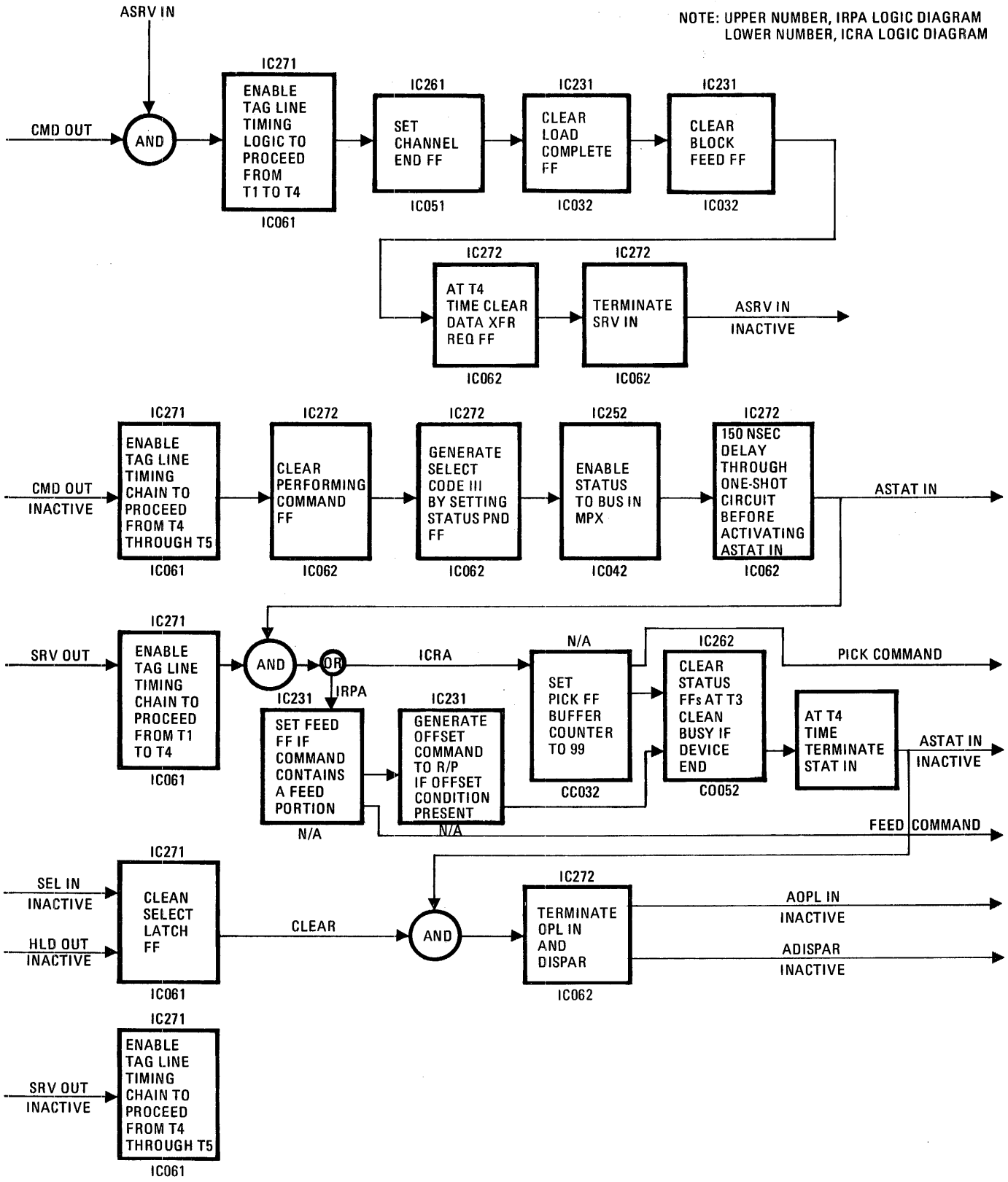


Figure 4-32. Timing Chart, Transfer of One EBCDIC Character

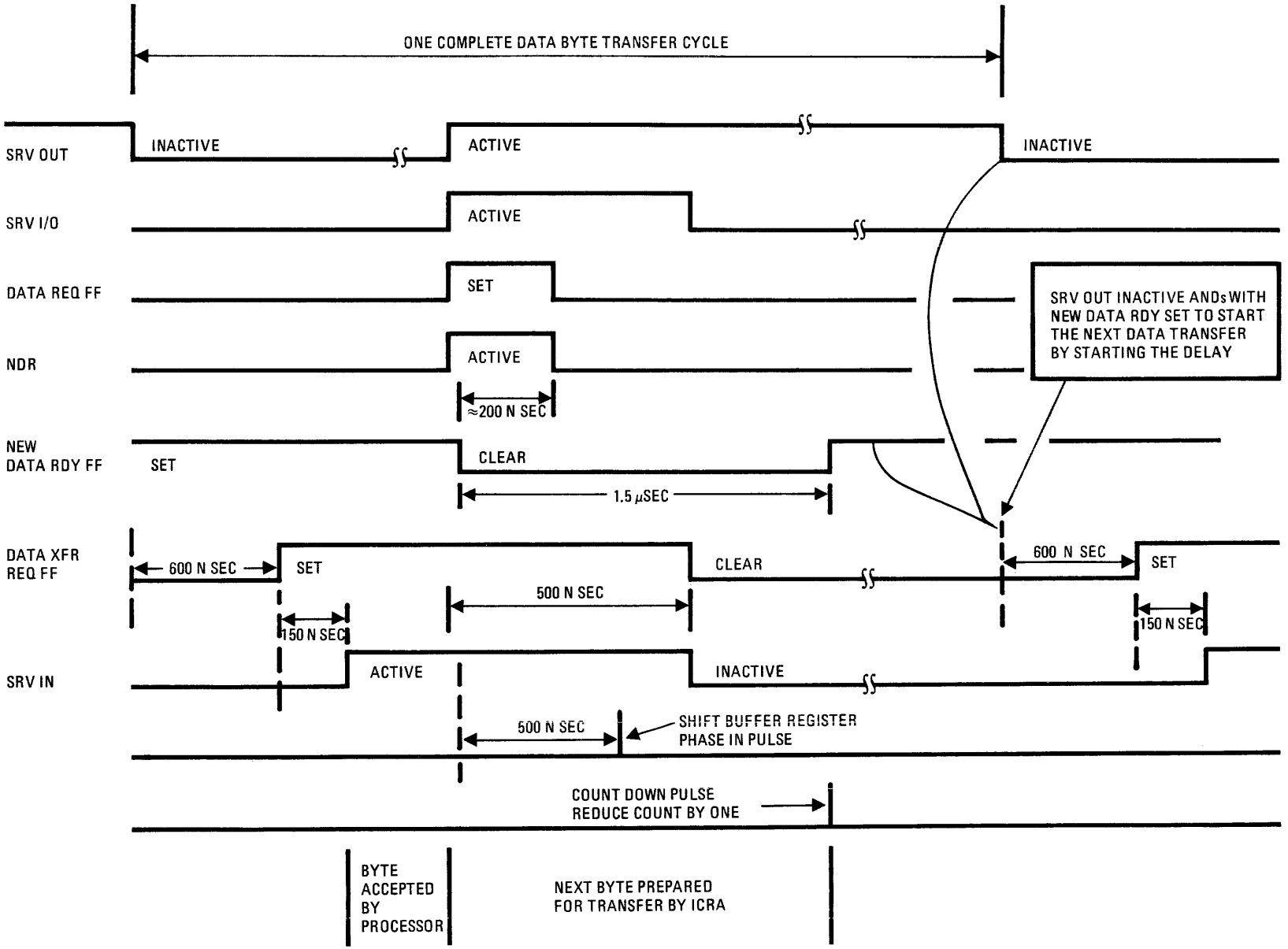


Figure 4-33. Adapter Input Ending Sequence

Status Indication Stacked – The ICRA, in response to CMD OUT, does not start another card read cycle, but instead becomes deselected and activates the REQ IN signal. CMD OUT sets the Stat Pnd FF which the Tag Line logic uses to activate the REQ IN signal. The processor also terminates HLD OUT which clears the Select Latch FF. The ICRA now clears OPL IN and DISPAR but continues to disable the propagation of SEL IN because the REQ IN signal is active. The ending sequence is complete when the processor terminates CMD OUT, with the status indication stacked and the ICRA requesting service with the REQ IN signal.

Adapter-Terminated Transfer

The ICRA will terminate the data transfer and proceed into an ending sequence when all 80 columns of information in the Buffer register have been transferred.

As shown in the adapter-terminated timing diagram, Figure 4-34, the count in the Column Counter is reduced to zero during the preparation for transfer of column 79. After the processor has accepted Byte 79 it indicates this fact by activating SRV OUT. The ICRA prepares Byte 80 for transfer by shifting the contents of the Buffer register right one phase and reducing the Column Counter by one. Because the count is zero the counter generates a BORROW OUT signal which clears the Load Complete and the Block Feed FF's.

The Load Complete FF clear is the key to terminating the data transfer and starting the ending sequence within the ICRA logic. The data transfer proceeds as normal until the processor accepts Byte 80 and activates SRV OUT (point A in Figure 4-34).

Because the Load Complete FF is clear at this point, the ICRA does not activate the NEW DATA REQUEST signal, thus not enabling the Two-Phase Timing Chain in the Reader Interface logic. The ICRA does, however, set the Trailing Edge Load Complete and Channel End flip-flops and then terminates SRV IN.

Processor 2 responds to the SRV IN termination by terminating SRV OUT. The data transfer sequence is now complete; the ICRA starts the ending sequence by enabling the device status through the Bus In MPX to the processor and activating STAT IN.

The processor responds by activating SRV OUT and terminating HLD OUT and SEL OUT. The ICRA uses SRV OUT to generate the FEED command. The FEED command sets the Feed FF, starting a card read cycle in the reader, and sets the Column Counter to 99 through the Preset one-shot circuit. The deselection is accomplished when the ICRA clears Stat In at T4 and then terminates both OPL IN and DISPAR in the same manner as for the adapter-terminated ending sequence.

Next Card Preparation

As pointed out in the two preceding ending-sequence presentations, the ICRA starts the card load sequence for the next card during every ending sequence if no *Unit Check* condition is present and the status indication was accepted.

During the card load sequence for the next card, the ICRA performs the same as for the initial card load (Figure 4-28). The data is read from the card, placed in the Buffer register and then right shifted in preparation for transfer to the processor. Once this is completed, the ICRA starts an adapter-initiated sequence by activating REQ IN.

The ICRA presents its address and status indication to the processor during the adapter-initiated sequence. It is during this sequence that the present ICRA and reader status indications are either accepted or stacked by the processor. Once they are accepted, the ICRA is deselected and again waits until an input operation is started with an initial selection.

If the status indications are stacked, the ICRA will continue to request service with the REQ IN signal until its status indication is accepted and it is deselected. The ICRA and processor continue to perform the described sequences, during card read operations, until the correct number of cards have been read or an error is encountered. Figure 4-35 illustrates, during a card read operation, the relationship between the time an error occurs and when that condition is indicated in the Status Byte. First is shown a normal *read* operation followed by error handling examples.

CARD-IMAGE READ

The ICRA card load and adapter-initiated sequences perform the same regardless of the read mode, whereas the initial selection, data transfer and ending sequences differ depending on the read mode invoked by the *Read and Feed* command. The following presentation points out the differences in these three sequences for a card-image read, as compared to the EBCDIC read already discussed.

Initial Selection

There are only a couple of different logic operations in the ICRA during initial selection for a card-image *Read* as compared to an EBCDIC *Read*. The differences are pointed out with respect to the time they occur by using Figure 4-30 as a time reference.

The first difference occurs when the Command Byte is gated into the Command register in response to CMD OUT. The Command Decode logic activates the CARD IMAGE signal as well as the READ signal. The ICRA also clears the Even Byte FF in response to CMD OUT. This occurs during every initial selection, but is only significant for a card image *Read*.

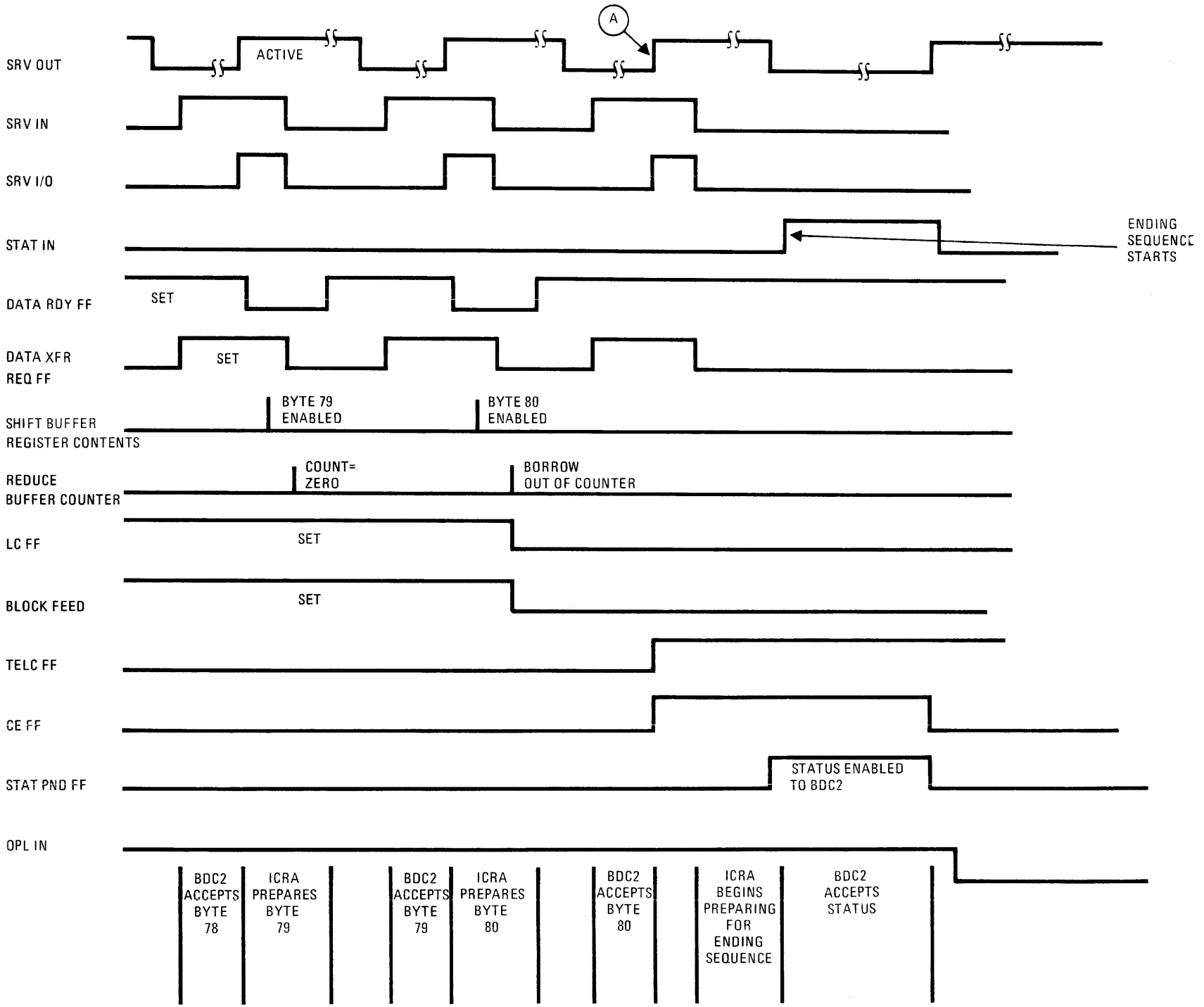


Figure 4-34. Timing Chart, Adapter-Terminated Sequence

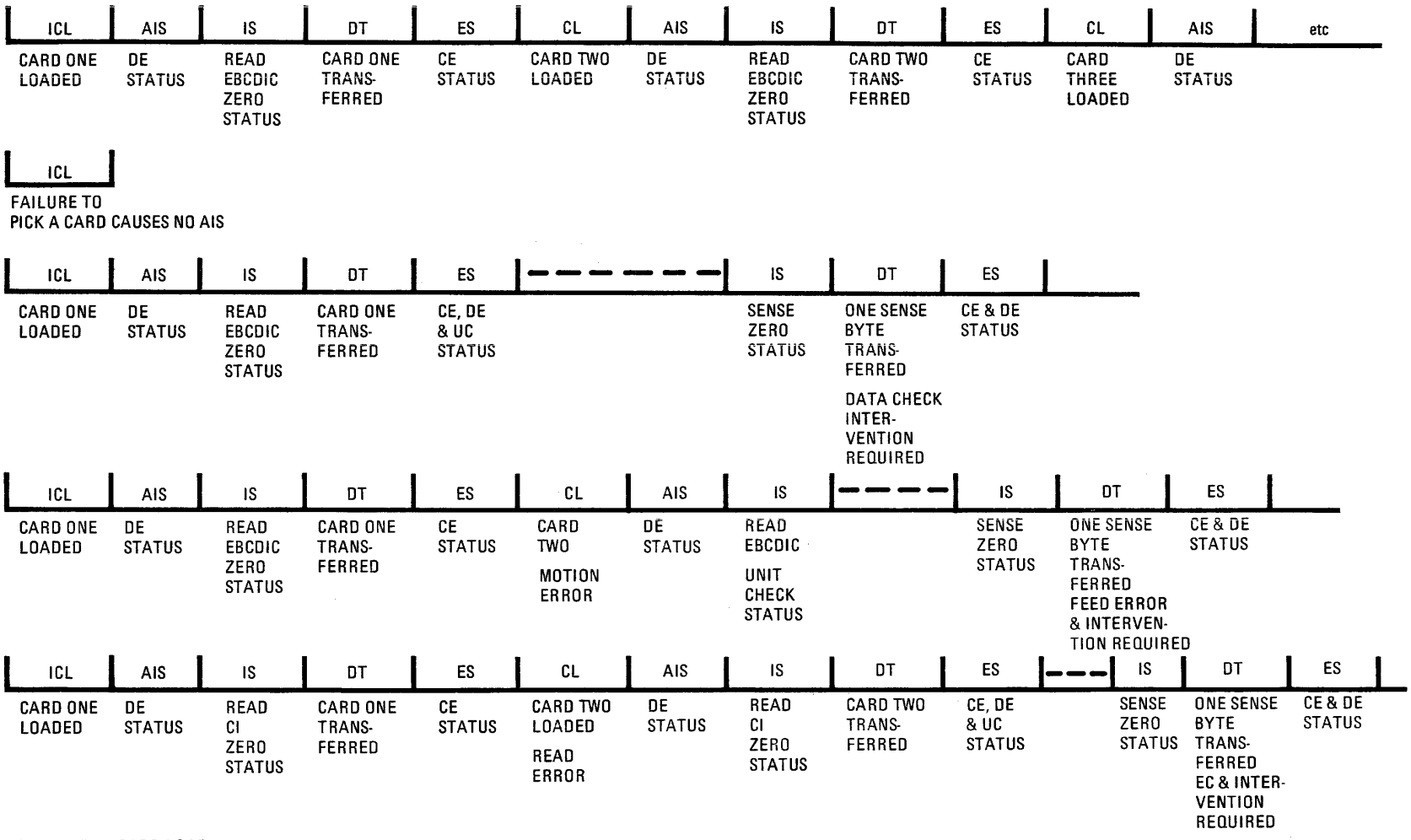


Figure 4-35. Adapter Error Handling

The initial selection proceeds as shown in Figure 4-29 until the STAT IN signal is terminated. At this point the Select Code logic generates the SELECT 001 rather than 000. This is because the CARD IM signal is active and the Even Byte FF is clear. The select code 001 enables the odd byte (bits 12 to 3) in the Buffer register to the Bus In Multiplexer. This then will be the first of the two bytes transferred to the processor for the first column of card image data.

Data Transfer

The data transfer starts when the processor terminates SRV OUT after the ICRA-terminated STAT IN. As shown in Figure 4-36, the ICRA reacts to inactive SRV OUT the first time the same as it does for an EBCDIC read. The odd byte is already on the Bus In lines so all the ICRA does is indicate it's ready by activating SRV IN.

After the processor has accepted the data it activates SRV OUT. The ICRA responds by terminating SRV IN and preparing for transfer the even byte of the first column of card-image data. SRV IN going inactive sets the Even Byte FF, which forces the Select Mode logic to generate the SELECT code 010. This code enables the even byte (bits 4 to 9) in the Buffer register to the Bus In Multiplexer.

The processor responds by terminating SRV OUT, causing the ICRA to again activate SRV IN; this indicates that the second byte is on the Bus In lines ready for transfer.

The processor accepts the even byte and then activates SRV OUT. The ICRA responds to SRV OUT by preparing the next column of card image data for transfer. SRV OUT active generates the NDR signal since the Data Req FF is set. The NDR signal enables the input to the Two-Phase Timing Chain and clears the Data Rdy FF. It in turn disables the NDR signal, thus allowing only one pass through the Two-Phase Timing Chain. The PHASE IN signal shifts the contents of the Buffer register right one place while the output of the second one-shot sets the Data Rdy FF and counts down the Buffer Counter by one.

The ICRA next clears the Data Xfr Request FF, which terminates the SRV IN signal. SRV IN going inactive clears the Even Byte FF, forcing SELECT code 001. The odd byte of the second column of card-image data is now enabled into the Bus In Multiplexer, ready for transfer. The processor responds by terminating SRV OUT to start the transfer of the second column of data.

The events shown in Figure 4-36 are repeated once for each column of card-image data until the data transfer is terminated.

Ending Sequence

The sequence of events for a channel-terminated card-image read from the ICRA are identical to those shown in Figure 4-33 for a channel-terminated EBCDIC read. During an adapter-terminated card-image read, the ICRA performs differently with respect to an EBCDIC read because of the two bytes transferred per card-image column.

The ICRA terminates the data transfer when both bytes of all 80 columns of information in the Buffer register have been transferred.

The processor indicates acceptance of the even byte for column 79 by activating SRV OUT, as shown in point A in Figure 4-37. The ICRA responses by preparing column 80 for transfer. It shifts the contents of the Buffer register one place and reduces the Column Counter by one. The count was reduced to zero during the preparation of column 79 and for this reason the Column Counter generates a BORROW OUT when further reduced for column 80. The BORROW OUT signal clears the Load Complete and Block Feed FF's, because all the necessary shifting has been completed.

Next the Even Byte FF is cleared, which enables the odd byte to the Bus In Multiplexer, and SRV IN is terminated. The data transfer proceeds as for any other byte until the processor accepts the odd byte column 80 and activates SRV OUT (point B). In response, the ICRA sets the TELC and CE FF's because the LC FF is clear. The Last Byte Card Image FF is also set. This keeps the Stat Pnd FF from setting when SRV OUT is terminated (point C) because the even byte column 80 has not been accepted at that point. Finally the Even Byte FF is set, enabling the even byte for column 80 to the Bus In Multiplexer and SRV IN is terminated.

Once even byte column 80 is accepted, the processor activates SRV OUT (point D). The ICRA clears the Last Byte Card Image FF in preparation for the ending sequence, and terminates SRV IN. When the processor terminates SRV OUT, the ICRA sets the Stat Pnd FF because the Last Byte Card Image FF is now clear; and activates STAT IN, starting the end sequence.

NON-MOTION COMMANDS

The *Sense, Test I/O* and *No-Operation* commands do not cause card motion in the reader. *No-Operation* and *Test I/O* require only an initial selection sequence. *Sense* uses an initial selection, data transfer and ending sequence, but only one byte of sense information is transferred to the processor during the data transfer sequence (Figure 4-38).

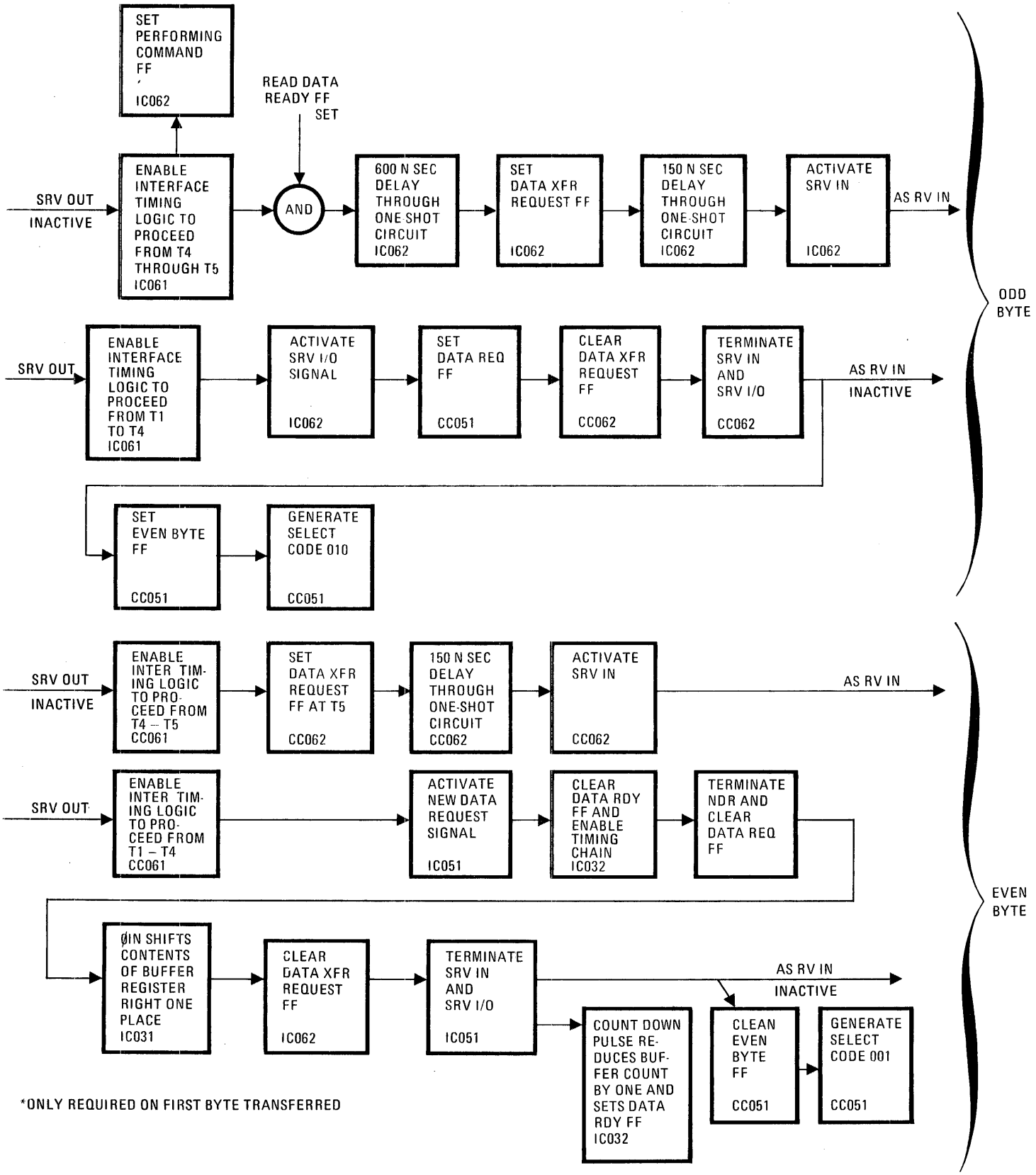


Figure 4-36. ICRA Card Image Data Transfer Sequence

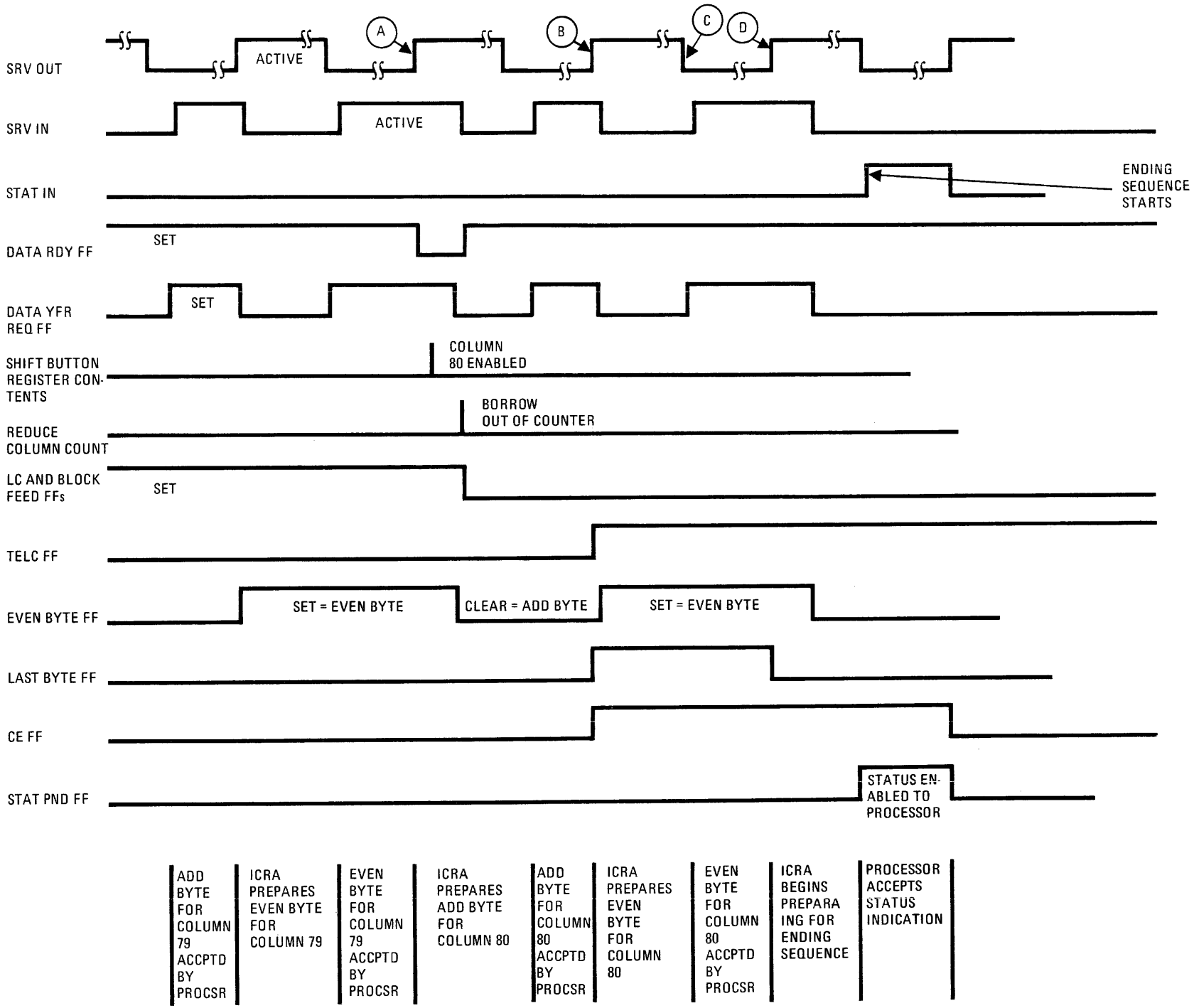


Figure 4-37. Timing Chart, Adapter Termination of Card Image Read

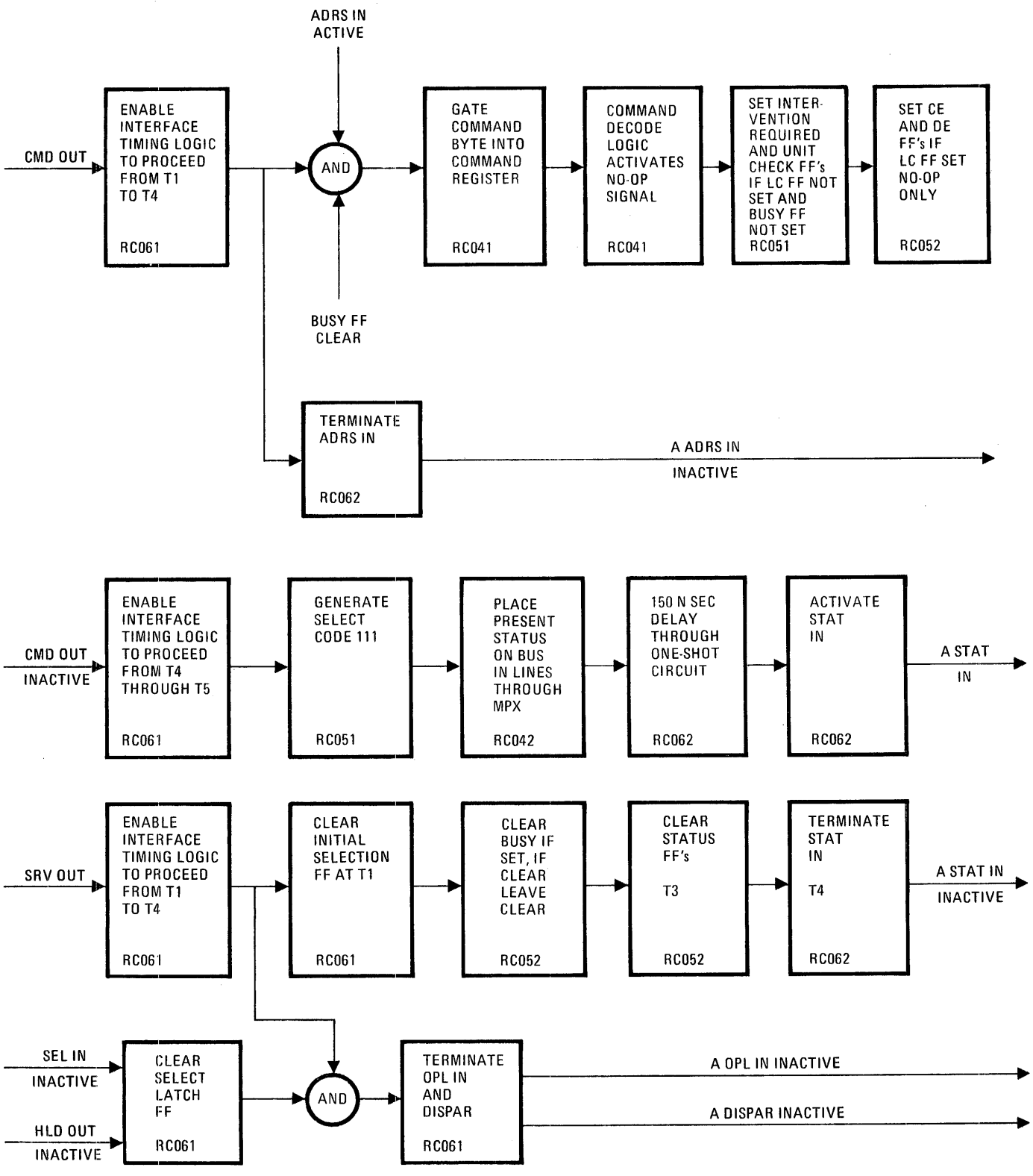
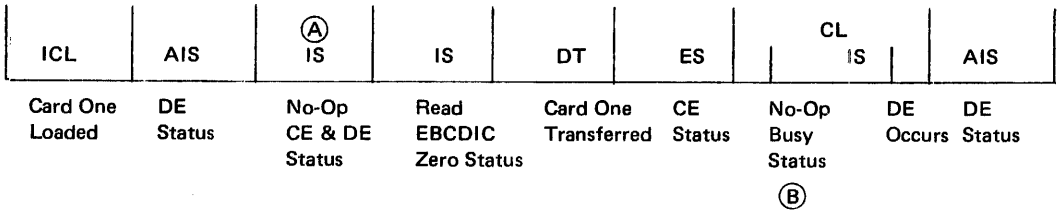


Figure 4-38. ICRA No-Operation and Test I/O Sequences

No-Operation Command

If the *No-Op* command is issued during an initial selection sequence while the ICRA is not busy, indicated by sequence A in the diagram below, it is accepted and executed by the ICRA. The normal status response at this point is *Channel End* and *Device End;* this is shown in sequence A. If *No-Op* is issued during an initial selection sequence while the ICRA is busy and has not reached *Channel End* or *Device End,* the ICRA will reply with a *Busy* status indication (sequence B).



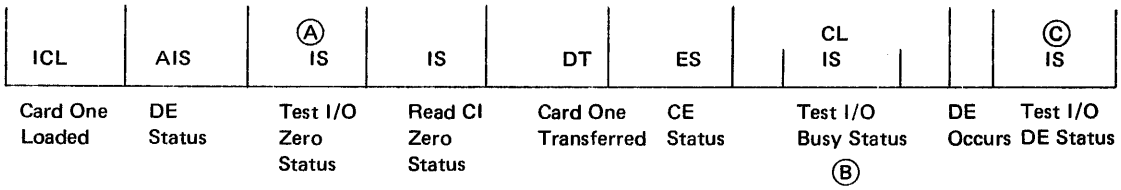
ICL – Initial Card Load
AIS – Adapter Initiated Sequence
IS – Initial Selection Sequence
DT – Data Transfer Sequence
ES – Ending Sequence
CL – Card Load

Figure 4-38 shows only the events which take place from CMD OUT to the completion of a *No-Op* and *Test I/O* initial selection, because the events prior to CMD OUT (Figure 4-30) are identical for all commands.

Test I/O Command

If the *Test I/O* command is issued while the ICRA is not busy and has no outstanding status (sequence A in diagram below), it is accepted and executed by the ICRA. The status response is zero.

Test I/O, like any of the commands, if issued while the ICRA is busy and has not reached *Channel End* or *Device End*, will not be executed. The ICRA will proceed through the initial selection with only the *Busy* bit in the Status Byte (sequence B). The status response to *Test I/O* (when the ICRA is busy and has *Device End* status indication which the processor has not accepted (sequence C)) is *Device End* only, without *Busy*.



Sense Command

A *Sense* command issued while the ICRA is not busy will be accepted and executed. During the initial selection the status indication is zero. During the data transfer sequence, the one byte of sense information is transferred to the processor, and *Channel End* and *Device End* are presented during the ending sequence. If the initial selection status indication is not zero, the processor terminates HLD OUT and SEL IN, causing the ICRA to be deselected at the end of the initial selection by the termination of OPL IN.

CONTROL STORAGE LOGIC

The Control Storage Load (CSL) logic within the ICRA allows the shared resources to load control storage from punched cards. The data read from the cards is transferred by the ICRA directly to the shared resources, rather than through the Basic Data Channel control circuitry (Figure 4-20).

A portion of CSL logic is located on the EP module with the rest of the logic, and the drivers are on the DG module. In order for the CSL logic to

communicate with the shared resources the LOAD SELECT switch must be in the CR position since its function is to enable or disable the CSL logic.

The ICRA sends four data bits and two nybl bits as well as two control signals to the Control Storage Loader logic in the shared resources. The eight drivers for these lines are shown on page CC042 with a definition of each signal listed below.

- End Data Input (ENDI) – This signal indicates that both the card reader and ICRA have properly completed their portion of the control storage load operation.
- Other Data Strobe (ODS) – This signal is active when data is available to the shared resources. It is active for 500 nanoseconds during each nybl transfer.
- The four bits of data transferred from the Buffer register: bits 12, 11, 0 and 1.
- NYBL 0 – This signal is controlled from bit position 8 of the Buffer register. When active, it indicates that the four bits of data on the lines are the first four bits, or nybl zero, of a Control Storage word.
- NYBL 3 – This signal is controlled from bit position 9 of the Buffer register. When active, it indicates that the four bits of data on the lines are the last four bits, or nybl three, of a Control Storage word.

The CSL logic on the EP module (page CC052) receives only two control signals from the shared resources. They both enable and start a Control Storage Load operation in the ICRA. The CR select signal originates at the LOAD SELECT switch on the system control panel. With the switch in the CR position the CR signal is active, thus enabling the ICRA to respond to the DOA signal. DOA is activated by the shared resources at the beginning of CSL operation and held active until its completion.

During a CSL operation the ICRA alternately performs card load sequences and CS data transfer sequences. The CSL operation is terminated by any one of the following three conditions occurring:

- All punched cards have been read and their data transferred to the shared resources. This is the correct termination which uses the ENDI signal.
- Either a read or feed error is detected in the card reader.
- The DOA signal is terminated by the shared resources before all the punched cards have been read. DOA going inactive indicates that control storage is full.

Initial Card Load

A card load sequence is performed for the initial card and, as mentioned earlier, for each subsequent card during the CSL operation. The initial card load is performed as soon as the card reader is made ready by the operator: cards in input magazine (hopper) and START switch pressed. The card load sequence performed by the ICRA reader interface logic (logic page CC032) is identical to the card load sequence represented in Figure 4-27, except for the activating of REQ IN. As explained earlier, during a card load the data is read from the card, placed in the Buffer register and then right shifted 20 places in preparation for output. The REQ IN is not activated since the RESET signal is active.

Once the initial card load is complete, with no errors, the ICRA is ready to perform a CSL operation. The shared resources start the operation by activating the DOA signal. With the CSL switch in the ON position, the CSL logic responds to DOA by performing a CS data transfer sequence, then a card load sequence, then another CS data transfer sequence, and so on alternately until the termination of the CSL operation.

CSL Operation

The LOAD SELECT switch in the CR position not only allows the CSL logic to respond to the DOA signal, but it also enables the contents of Rank 100 of the Buffer register to go through the CSL drivers to the shared resources. The CSL logic checks the status of the card reader before starting the CS data transfer sequence, as shown in Figure 4-39. If the card load sequence performed correctly, the data transfer is started; if it did not, the ICRA stops the CSL operation without activating the ENDI signal.

The CSL logic activates the CSDR signal if the card load sequence had no errors. This signal, delayed through a one-shot circuit before activating the CSNDR signal which starts the two-phase timing chain in the Reader Interface logic, clears the Data Rdy FF. The Data Rdy FF clearing terminates the CSDR signal which, after 200 nanoseconds, terminates the CSNDR signal thereby disabling the timing chain input. The CSL logic uses the timing chain phase one output to generate the ODS signal to the shared resources. The ODS is active for 500 nanoseconds; it is during this time that the shared resources accept the data. On the trailing edge of PHASE ONE pulse the Buffer register contents are right shifted one place.

The timing chain PHASE TWO pulse reduces the Column Counter by one, and when that terminates sets the Data Rdy FF again. The Data Rdy FF set again starts another data transfer as just described.

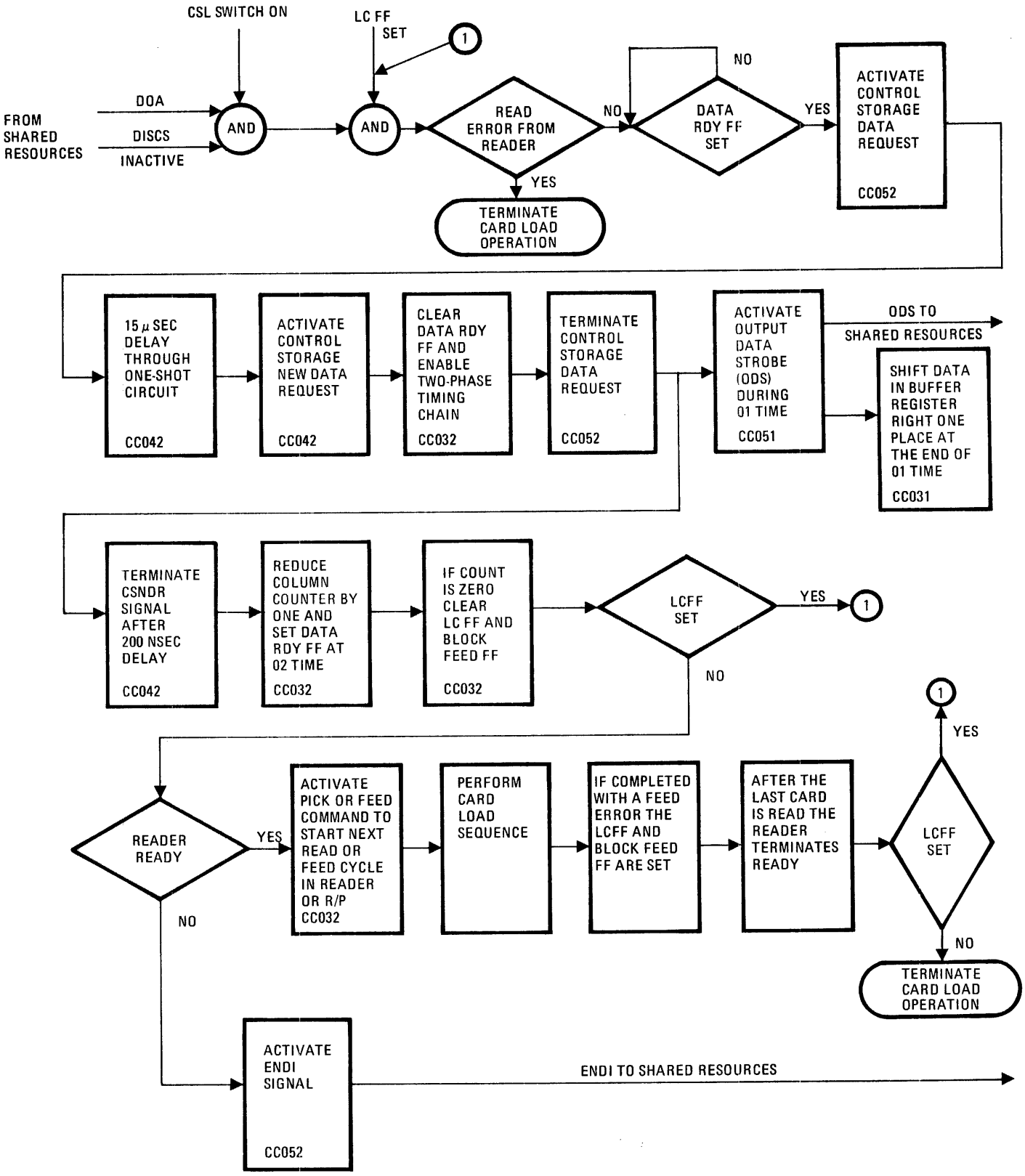


Figure 4-39. Flowchart, Control Storage Operation

This sequence is repeated until all 80 columns of data in the Buffer register have been transferred to the shared resources. During the 80th data transfer the Column Counter produces a BORROW OUT signal which clears the Load Complete FF. The CSL logic now initiates a card load sequence if the reader is still ready.

The card load sequence again reads a card and places the data in the Buffer register ready for transfer. After that, the LC FF is set. LC going set starts another CS data transfer sequence by activating the CSDR signal if no read errors were encountered during the card load sequence. The data transfer events repeat until all 80 columns have been transferred, which is signified by the LC FF being cleared. At this point the CSL logic makes the decision whether to start another card load sequence or to terminate the Control Storage operation. If during the previous card load sequence the last card had been read the reader is now no longer ready. In this case, the CSL logic activates the ENDI signal to the shared resources and stops.

INTEGRATED READER/PUNCH ADAPTER (IRPA)

NOTE

Processes and functions common to both the IRPA and the Integrated Card Reader Adapter (ICRA) are generally not fully described in this portion of this manual. They were described in the ICRA portion. In a few instances, IRPA descriptions were included in those generalized descriptions because they best fit into the generalized context.

GENERAL DESCRIPTION

The IRPA is the reader/punch controlling element. It contains the communications path and control logic between processor 2 and the card reader/punch. The IRPA, in conjunction with the reader/punch, provides a Memorex system with the ability to read information from 80-column cards and place it in main storage or to punch main storage information into 80-column cards.

The IRPA performs the following operations:

- Accepts either EBCDIC or binary data from the processor (punching).
- Transfers either EBCDIC or binary data to the processor (reading).
- Provides an 80-column data buffer register for temporary data storage.
- Transfers the data buffer register contents to the processor and reads the next card in response to a single *Read* command.
- Punches the data sent by the processor in the card and places the card in the specified stacker in response to a single *Punch* command.
- Reads and stores the first card data in the Buffer register prior to the first command.
- Presents all necessary status and sense information when requested by the processor.
- Provides a control storage load (CSL) capability.

The IRPA block diagram is in Figure 4-40. All communications between the processor and the IRPA follow the operational sequence requirements. The control and data signals for the CSL feature connect directly to the control storage loader logic in the shared resources. The data and control signals from the reader/punch connect to the IRPA through a connector on the I/O connector panel.

Commands

The IRPA recognizes as valid seven commands, three of which cause card motion, and four do not. A description and programming considerations for each of the seven commands follows.

- NO-OPERATION

Command Byte:

0	1	Bits				6	7
0	0	0	0	0	0	1	1

Normal Status Indication: *Channel End* and *Device End* at initial selection.

Other Status Indication: *Unit Check* (bit 6) is presented if alone if an *Intervention Required* condition exists. *Busy* (bit 3) occurs if the reader/punch has not reached the *Device End* point or if the IRPA has outstanding status.

- TEST I/O

Command Byte:

0	1	Bits				6	7
0	0	0	0	0	1	0	0

Operation Performed: Transfers status information to the channel and clears Status flip-flops.

NOTE

Busy (bit 3) is included if the reader/punch is executing a *Read* or *Punch* command and the IRPA has no outstanding status.

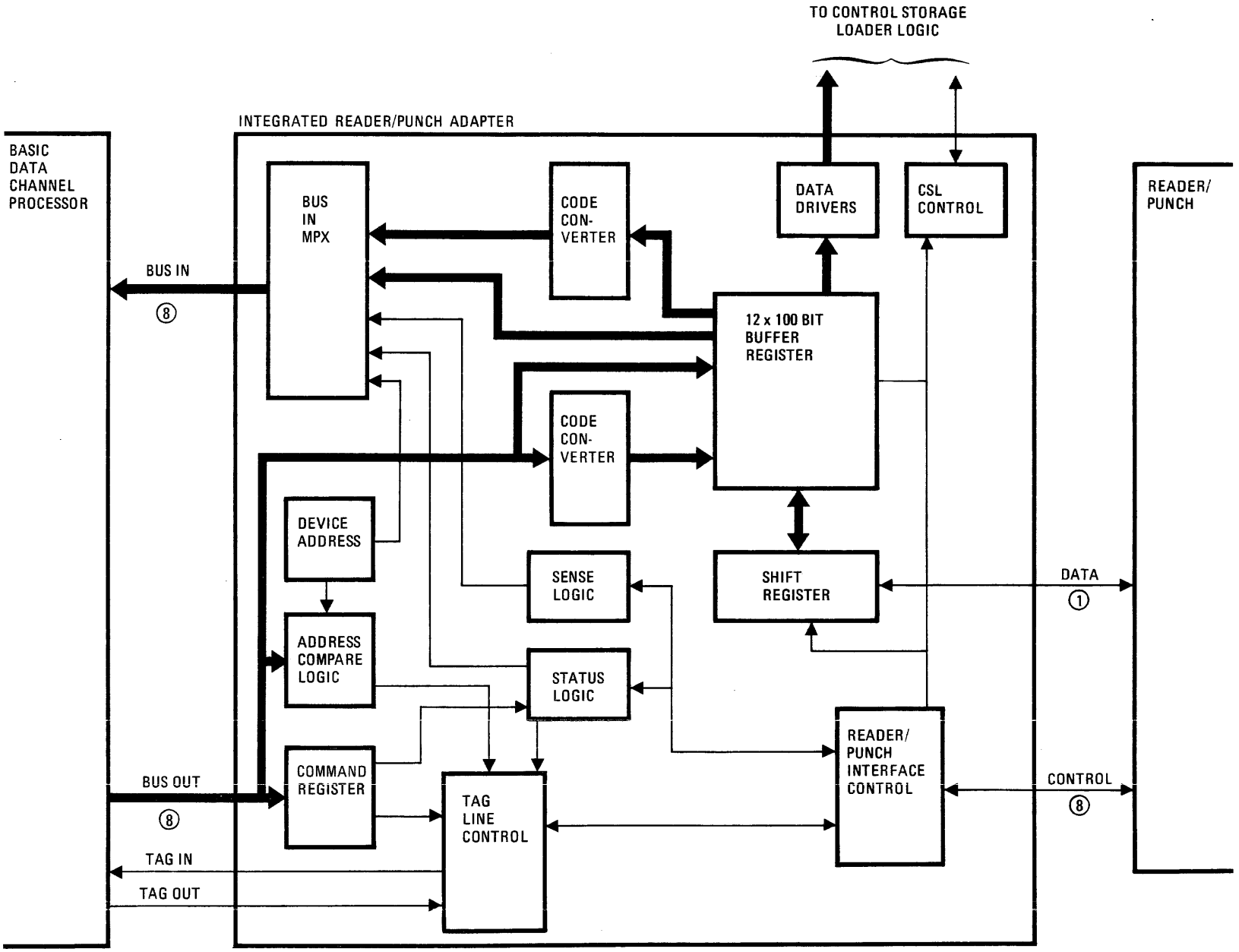
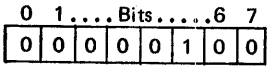


Figure 4-40. IRPA Block Diagram

- SENSE

Command Byte:

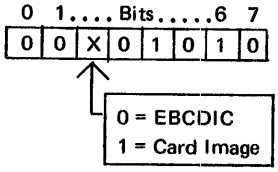


Operation Performed: Transfers one Sense Byte to the channel during the data transfer sequence. Presents *Channel End* and *Device End* during the ending sequence.

Normal Status Indication: Zero at initial selection.

- READ ONLY

Command Byte:



Operation Performed: Transfers the buffer register data to the channel. Presents *Channel End* and *Device End* during an ending sequence after the last Data Byte is transferred.

Normal Status Indication: Zero at initial selection.

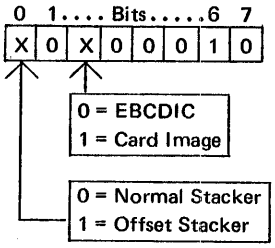
Command Sequence: A *Read Only* command should not follow another *Read Only* command. If it does, *Unit Check* occurs in the initial selection status and *Command Reject* in the Sense Byte.

Other Status Indication: *Unit Check* (bit 6) occurs at initial selection for any error causing *Intervention Required* or *Command Reject* in the Sense Byte.

Busy (bit 3) occurs at initial selection if the IRPA is still executing the previous command.

- READ, FEED AND SELECT STACKER

Command Byte:



Operation Performed: Transfers the Buffer register data to the channel, feeds the card (whose data was transferred) to the specified stacker position, and reads the next card, placing its data in the Buffer register. Presents *Channel End* during an ending sequence after the last Data Byte is transferred. Presents *Device End* during an adapter initiated sequence after the next card has been read.

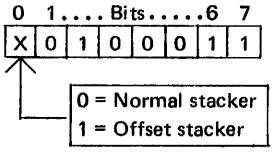
Normal Status Indication: Zero at initial selection.

Command Sequence: This command should not follow a *Read Only* command. If it does, *Unit Check* occurs in the initial selection Status Byte, and *Command Reject* appears in the Sense Byte.

Other Status Indication: *Unit Check* (bit 6) occurs at initial selection for any error causing *Intervention Required* or *Command Reject* in the sense byte. *Unit Check* occurs with *Channel End* for any error causing *Equipment Check* or *Data Check* in the Sense Byte.

- FEED AND SELECT STACKER

Command Byte:



Operation Performed: Feeds the card (whose data was transferred to the channel by the preceding *Read Only* command) to the specified stacker position, and reads the next card, placing its data in the Buffer register. Presents *Device End* during an adapter-initiated sequence after the next card has been read.

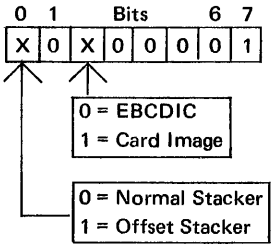
Normal Status Indication: *Channel End* at initial selection (immediate command as no data transfer occurs).

Command Sequence: This command must only follow a *Read Only* command and as such can never be the first command sent to the IRPA.

Other Status Indication: *Unit Check* (bit 6) occurs at initial selection instead of *Channel End* for an error causing *Intervention Required* or *Command Reject* in the Sense Byte.

- PUNCH, FEED AND SELECT STACKER

Command Byte:



Operation Performed: Accepts data from the processor and places it in the Buffer register; punches the Buffer register data into the card at the punch station and feeds it to the specified stacker position. Reads the next card, placing its data in the Buffer register. Presents *Channel End* during an ending sequence after the last Data Byte

is accepted from the channel. Presents *Device End* during an adapter-initiated sequence after the card is punched and the next card has been read.

Normal Status Indication: Zero at initial selection.

Other Status Indication: *Unit Check* (bit 6) occurs at initial selection for any error causing *Intervention Required* in the Sense Byte. *Unit Check* occurs with *Device End* for an error causing *Equipment Check* in the Status Byte.

Status Information
Sense Information
Data Formats

The generalized description presented previously for the Integrated Card Reader Adapter applies; the few items unique to the IRPA are presented there to be in context.

LOGIC DESCRIPTION

The basic logic description relevant to both the ICRA and IRPA is contained in the previous ICRA section. That relevant only to the IRPA is in this section.

Reader/Punch Data

Data is transferred between the reader/punch and the IRPA serially on a single bi-directional data line. Accompanying the data is a STROBE signal, which is active during the serial transfer of each column as shown in Figure 4-41A.

During a serial transfer of data with the reader/punch the IRPA uses the STROBE signal in enabling its clock circuitry. During the STROBE signal interval, 12 CLOCK pulses are generated which clock the 12 bits of data into or out of the Serial Data register (SDR).

The SDR consists of three 4-input right-shift register MSI elements (logic page 1C242). The fourth flip-flop output of an element is connected to the serial input of the next element. The input data from the reader/punch is applied to the serial input of the first element, and the output data for the reader/punch is taken from the output of the last element's last flip-flop. Each time the CLOCK goes low, the data shifts right one place.

For read (input) operations the reader/punch terminates the STROBE signal after it sends the twelfth bit of a column. The IRPA uses the inactive direction of STROBE to generate the PHASE IN signal which gates the 12 bits of data, in parallel, from the Serial Data register to the Buffer register (Figure 4-41B).

During punch (output) operations the IRPA uses STROBE going active to enable and then gate, in parallel, a column of data from Buffer register to the SDR (Figure 4-41C). The CLOCK pulses then shift the data out to the reader/punch.

Buffer Register

The IRPA Buffer register consists of six dual 100-bit, shift register, MSI elements. These are static shift registers and, as such, hold data placed in them until it is shifted out.

The 12 bits of input data come from the Punch/Read Data Select multiplexer. The inputs to this multiplexer come from the Serial Data register and the EBCDIC/Card Image multiplexer. During read operations, data from the Serial Data register is enabled through the Punch/Read multiplexer to the Buffer register inputs, while during punch operations data from the EBCDIC/Card Image multiplexer is enabled through the Punch/Read multiplexer to the Buffer register (Figure 4-22).

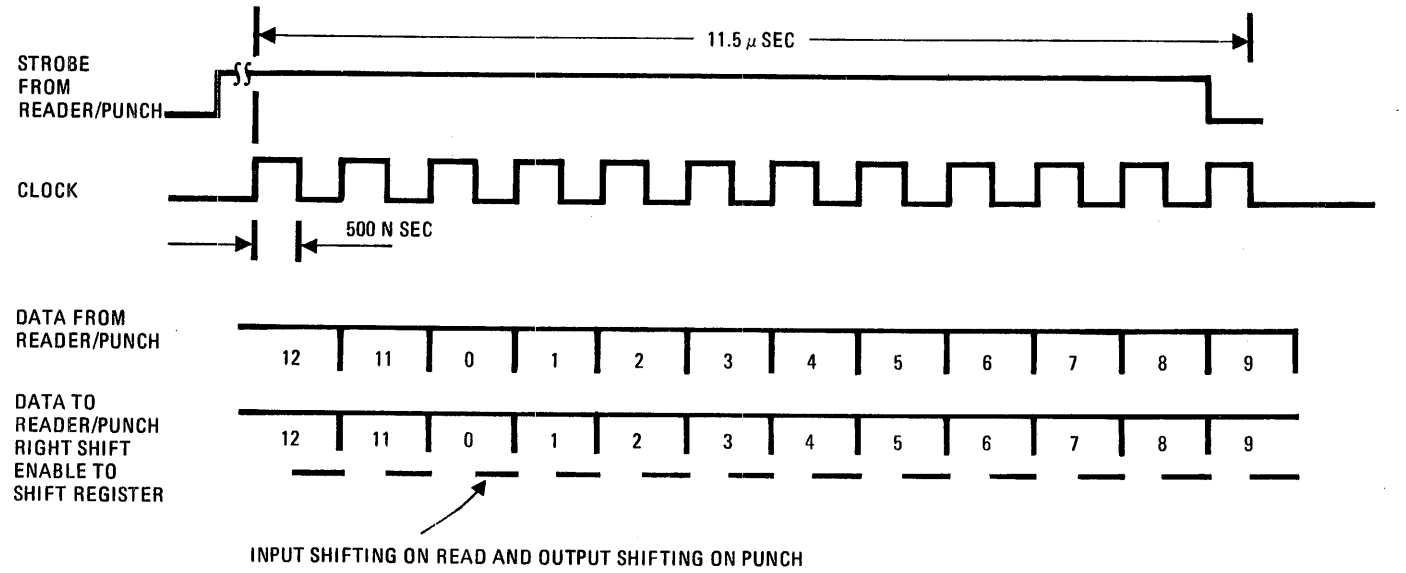
The 12 bits of input data are gated into the first rank of flip-flops and the contents of all other ranks are gated into the subsequent rank each time the PHASE IN signal goes high (Figure 4-23). The outputs from the last rank of flip-flops are **not** recirculated within the register. The output, taken from the last rank of flip-flops, is permanently enabled.

The Buffer register gives the IRPA the capability of holding all the information from either a 80- or 96-column card. Since all columns are read and placed in the Buffer register before being transferred to the channel, data cannot be lost.

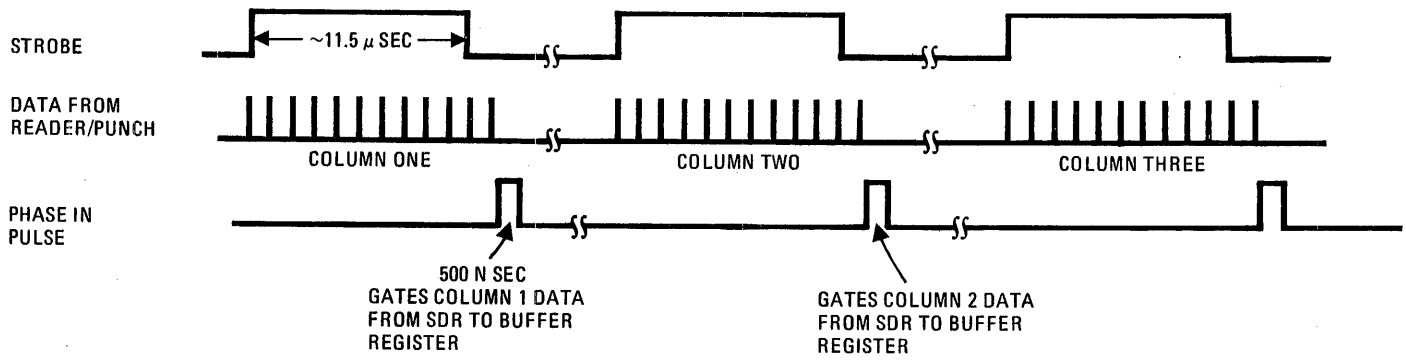
The Buffer register outputs are connected to the Serial Data register, the Bus In multiplexer, the Card-Code-To-EBCDIC converter, and the Invalid Character Checking logic.

The Character Checking logic (page 1C242) constantly monitors Buffer register outputs 1 through 7, looking for more than one logic "1". More than one logic "1" between bits 1 and 7 inclusive activates the INVALID CHARACTER signal (INVALIDCH) which is used by the Sense Logic (page 1C262). An invalid character will cause the setting of the Data Check flip-flop only during EBCDIC *Read* operations.

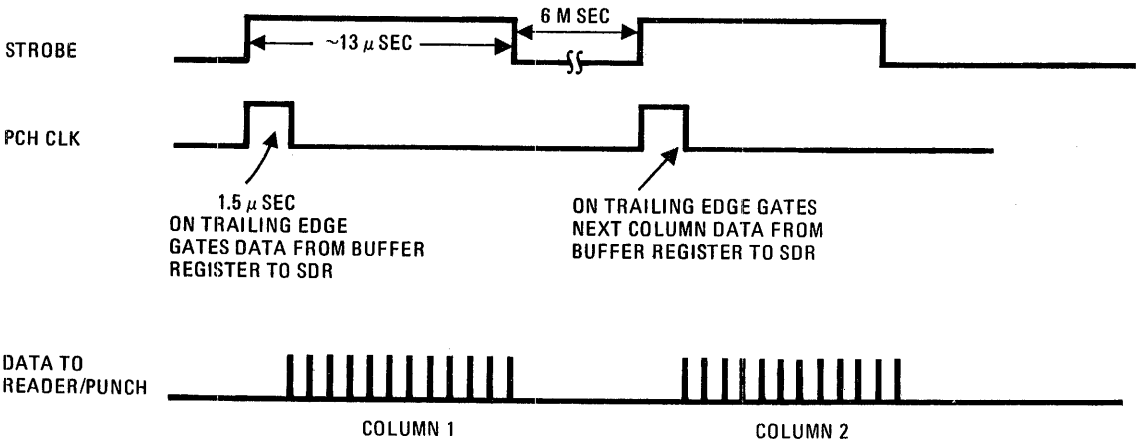
The Card Code converter constantly monitors the Buffer register output and, in conjunction with part of the Character Checking logic, translates or converts the 12-bit card code to an 8-bit EBCDIC character. The eight-bit Code Converter output is connected to input 0 of the Bus In multiplexer. Its output is routed through the multiplexer only during EBCDIC read operations.



A. BI-DIRECTIONAL SERIAL DATA TRANSFER



B. INPUT TRANSFER



C. OUTPUT TRANSFER

Figure 4-41. Timing Chart, IRPA Transfers

Buffer register outputs corresponding to card columns 3 through 12 connect to the Bus In multiplexer input 1, bits 2 through 7; register outputs for columns 9 through 14 connect to input 2, bits 2 through 7. These data paths are used when transferring the 12 bits of card image data from the Buffer register to the processor in two subsequent bytes.

Sense Logic

The four IRPA Sense flip-flops and the associated Sense logic are contained on the EP printed circuit module (page 1C261). The sense logic uses signals from the Command Decode logic and inputs from the Tag Line Control logic in generating *Command Reject*. Signals from the Command Decode logic, the Tag Line Control logic and the INVALID CH signal are used in generating *Data Check*. The *Equipment Check* condition is determined by monitoring the ERROR signal from the Reader Interface logic. The Intervention Required FF monitors signals from the Reader Interface logic, Control logic, and Status logic.

When any one of the five Sense flip-flops sets, the Unit Check FF sets, if the IRPA is not presently executing a *Sense* command.

NOTE

Many of the inputs have different signal names when the EP module is used in the reader adapter (ICRA). Of course, only the applicable reader/punch signal names appear in the reader/punch logic diagrams.

Status Logic

The IRPA Status logic shown on page 1C052 contains the four Status flip-flops and associated logic. The Status logic monitors signals from the Tag Line Control logic, the Reader Interface logic, the Command Decode logic and the Sense logic. The outputs from the Status flip-flops are used in the generation of the FEED command to the reader, by the Sense logic, by the Tag Line Control logic, and the Reader Interface logic.

Reader/Punch Interface Logic

The Reader/Punch Interface logic (EN module), shown on page 1C232, contains the receivers and transmitters for the reader/punch signals and the logic which controls the proper loading and unloading of the Buffer register during read operations. The following is a brief description of the reader/punch signals shown on page 1C232.

- READY – This signal when active indicates the reader/punch has successfully fed a card into the Punch Wait station and one into the Pre-Read Wait station and has no operator-intervention-required error conditions present.
- CARD PRESENCE – This signal is active while a card is passing through the Read station.
- STROBE – This signal is activated whenever the reader/punch is ready to send a column of data or is ready to accept a column of data. STROBE stays active until 12 data bits have been serially transferred to or from the reader/punch.
- FEED CHECK – This signal is activated whenever a feed cycle does not complete correctly and the input magazine (hopper) is not empty.
- ERROR – This signal is activated by a light or dark check failure (read), or by an echo check failure (punch).
- FEED – This signal, when activated by the IRPA, starts a card feed cycle in the reader/punch.
- OFFSET – This signal, active in conjunction with FEED, causes the reader/punch to offset from the normal stacker position the card entering the stacker, as a result of the FEED signal.
- CLOCK – This is the 1-microsecond CLOCK signal used by the reader/punch to gate the serial punch data into or out of its register.
- DATA – A bi-directional signal representing an item of serial read or punch data.

NOTE

The signals from the reader/punch are only activated and the signals from the IRPA are only accepted by the reader/punch when it is in for On-Line mode.

Serial Data Transfer

The single bi-directional Data line is connected to a transmitter and receiver in both the IRPA and the reader/punch, as shown in Figure 4-42. Because of this, a transmitter has three states – one, zero, and off. When a card is being read, the CARD PRESENCE signal is activated by the reader/punch. This signal inhibits the IRPA transmitter (placing it in the off state) and enables the IRPA receiver. The transmitter in the off state presents a high impedance to the Data line thus effectively removing the transmitter from the circuit.

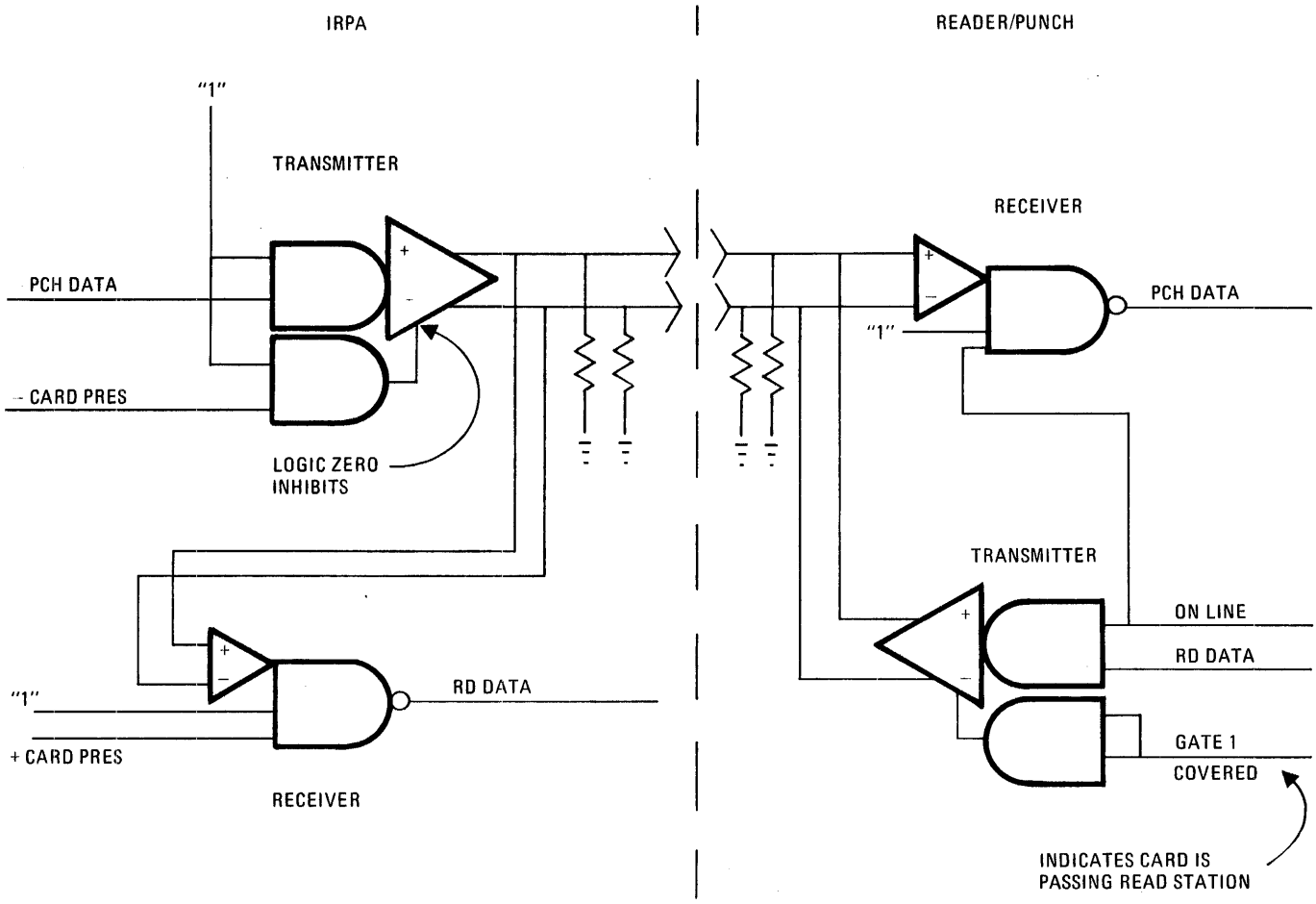


Figure 4-42. IRPA Data Line Configuration

When data is being sent to the reader/punch, the CARD PRESENCE signal is inactive; this disables the IRPA receiver and allows the punch data to control the IRPA transmitter.

The remaining logic on page 1C232 consists of the Two-Phase Timing Chain, the Column Counter and Reader/Punch Interface control logic. The Column Counter is composed of two 4-bit MSI counter elements.

The counter is preset to the binary equivalent of either 78 or 99 by the 5-microsecond pulse out of the Preset one-shot circuit. The Column Counter decreases by one for each COUNT DOWN pulse it receives. After the counter has reached zero, the next COUNT DOWN pulse produces a BORROW OUT signal. This signal pulses the Load Complete FF, which has a toggling input (it changes states each time it is pulsed).

Two-Phase Timing Chain

The IRPA COUNT DOWN and PHASE IN pulses are generated by signals from the reader/punch, the Punching Control logic, or the Two-Phase Timing Chain. The timing chain consists of two one-shot circuits. When enabled, the first one-shot (phase 1) produces a 500-nanosecond pulse (Figure 4-24) which feeds both the second one-shot and the PHASE IN signal. As stated earlier, the trailing edge of the PHASE IN signal gates the data from the SDR into the Buffer register, and shifts all other data right one place.

The trailing edge of PHASE 1 triggers the Phase 2 one-shot which in turn generates a one-microsecond COUNT DOWN pulse. The PHASE 2 trailing edge triggers the Phase 1 one-shot again if it is still enabled. Each pass through the chain moves the data right one place in the Buffer register and reduces the count in the Column Counter by 1.

In addition to the reader/punch signals and IRPA logic signals the Reader/Punch Interface logic receives a POWER ON MASTER CLEAR signal from the Power Sequencer unit. This signal is activated at the beginning of the power-up sequence and is not terminated until all equipments in the system have been brought up to power. When activated, this signal clears Control flip-flops and the Non-Zero Punch Counter on the EN module.

Punch Control Logic

The Punch Control Interface logic and CLOCK signal generating logic is on logic page 1C231. Two one-shot circuits connected in a cyclic fashion produce both the CLOCK and MUX CLOCK signals while enabled by reader/punch signals or the punch control logic.

The punch control logic has these functions:

- Control over loading of the Buffer register when accepting data from the processor.
- The gating of data from the Buffer register to SDR and the serial transfer of it to the reader/punch.
- The counting of zero columns and the subsequent generation of the SKIPOUT signal.

Tag Line Control
Tag Line Timing Chain

Data relevant to both ICRA and IRPA previously presented in ICRA discussion.

OPERATING SEQUENCES

The operating sequences performed by the IRPA are divided into the following three categories: those used to communicate with processor 2, those used to communicate with the reader/punch and those needed for the control storage load function. The sequences used and their order of occurrence during the execution of commands and control storage load are shown in Figure 4-43.

Initial Run-In

In response to pressing the START switch, the reader/punch and the IRPA perform an initial run-in sequence. This sequence includes the execution of two feed cycles by the reader/punch. During the first cycle a card is moved from the input magazine (hopper) to the Pre-Read station. In the second cycle the first card is moved from the Pre-Read station past the Read station, during which the data is read and loaded into the IRPA. The first card stops at the Wait Punch station. Also, while card one is being read a second card is being moved from the input magazine to the Pre-Read station. After both feed cycles have been completed correctly, the reader/punch is ready, and activates the READY signal to the IRPA.

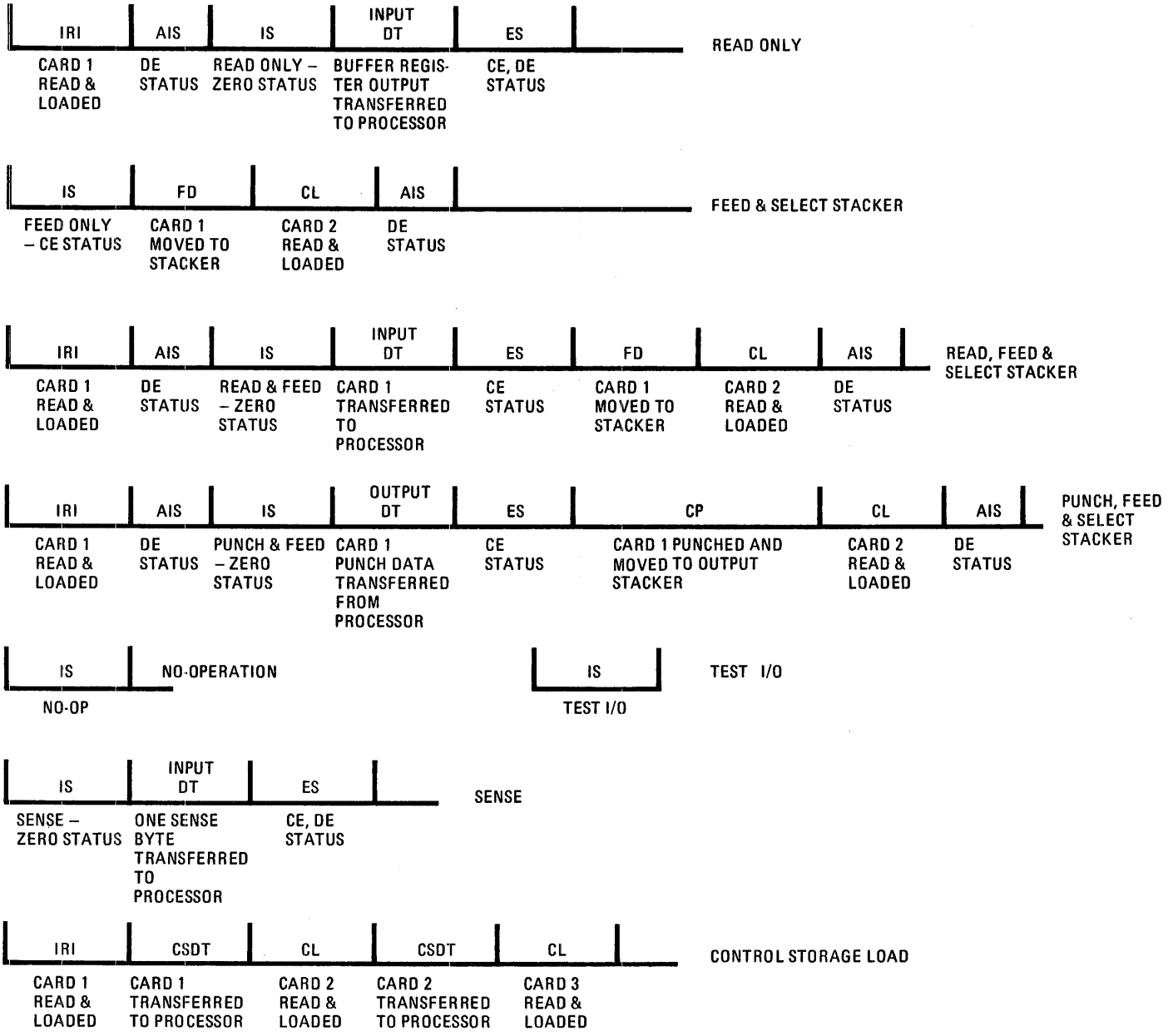
The IRPA performs a number of events, in response to reader/punch signals, during the card load sequence for the first card as illustrated by Figure 4-44. At the point in the second feed cycle at which the first card enters the Read station the reader/punch activates the CARD PRESENCE signal. The IRPA uses CARD PRESENCE to enable the clock logic and to trigger the Preset Counter one-shot circuit whose 5-microsecond output pulse at this point sets the Column Counter to 99. The IRPA also used CARD PRESENCE to initialize three control flip-flops, set the Punch Counter to all 1's, disable the data transmitter to the reader/punch, and enable the data receiver from the reader/punch.

Once the reader/punch has read the first column of data and is ready to transfer it to the IRPA, it activates the STROBE signal. The IRPA clock logic responds by generating the CLOCK pulses as shown in Figure 4-41A. The CLOCK signal is sent to the reader/punch which uses it to serially present the 12 bits of data. Internally the CLOCK signal shifts the 12 bits of data into the SDR.

After 12 CLOCK pulses, the reader/punch terminates the STROBE signal which stops the clock logic. STROBE going inactive also reduces the Column Counter by 1, by forcing the count down input to the counter to a logic "1". It also generates a PHASE IN signal by triggering the 500-nanosecond Read Data Strobe one-shot circuit. The PHASE IN signal gates the data from the SDR to the Buffer register.

At this point, the first column of data has been read from the card, transferred to the IRPA, placed in the Buffer register, and the Column Counter has been reduced by 1. When the second column of data is ready for transfer, the reader/punch again activates STROBE and the above described events again take place. After this sequence of events is performed 80 times, the Column Counter is reduced to a count of 19, and the card data resides between ranks 0 and 80 of the Buffer register.

The reader/punch terminates the CARD PRESENCE signal as the card leaves the Read station. Termination of the signal starts the Two-Phase Timing Chain, causing it to generate 20 PHASE IN and COUNT DOWN pulses (as described in the Two-Phase Timing Chain paragraph). The trailing edge of the first PHASE TWO pulse sets the Read Data Ready FF. Nineteen count-down pulses reduce the Column Counter to zero. The twentieth pulse produces a BORROW OUT from the counter, which disables the timing chain by setting the Load Complete FF. The 20 PHASE IN pulses move the data in the Buffer register to ranks 20 through 100 in preparation for transfer to the channel.



IS – INITIAL SELECTION
IRI – INITIAL RUN-IN
AIS – ADAPTER INITIATED SEQUENCE
DT – DATA TRANSFER
ES – ENDING SEQUENCE
CP – CARD PUNCH
CL – CARD LOAD
FD – FEED
CSDT – CONTROL STORAGE DATA TRANSFER

Figure 4-43. IRPA Operating Sequences

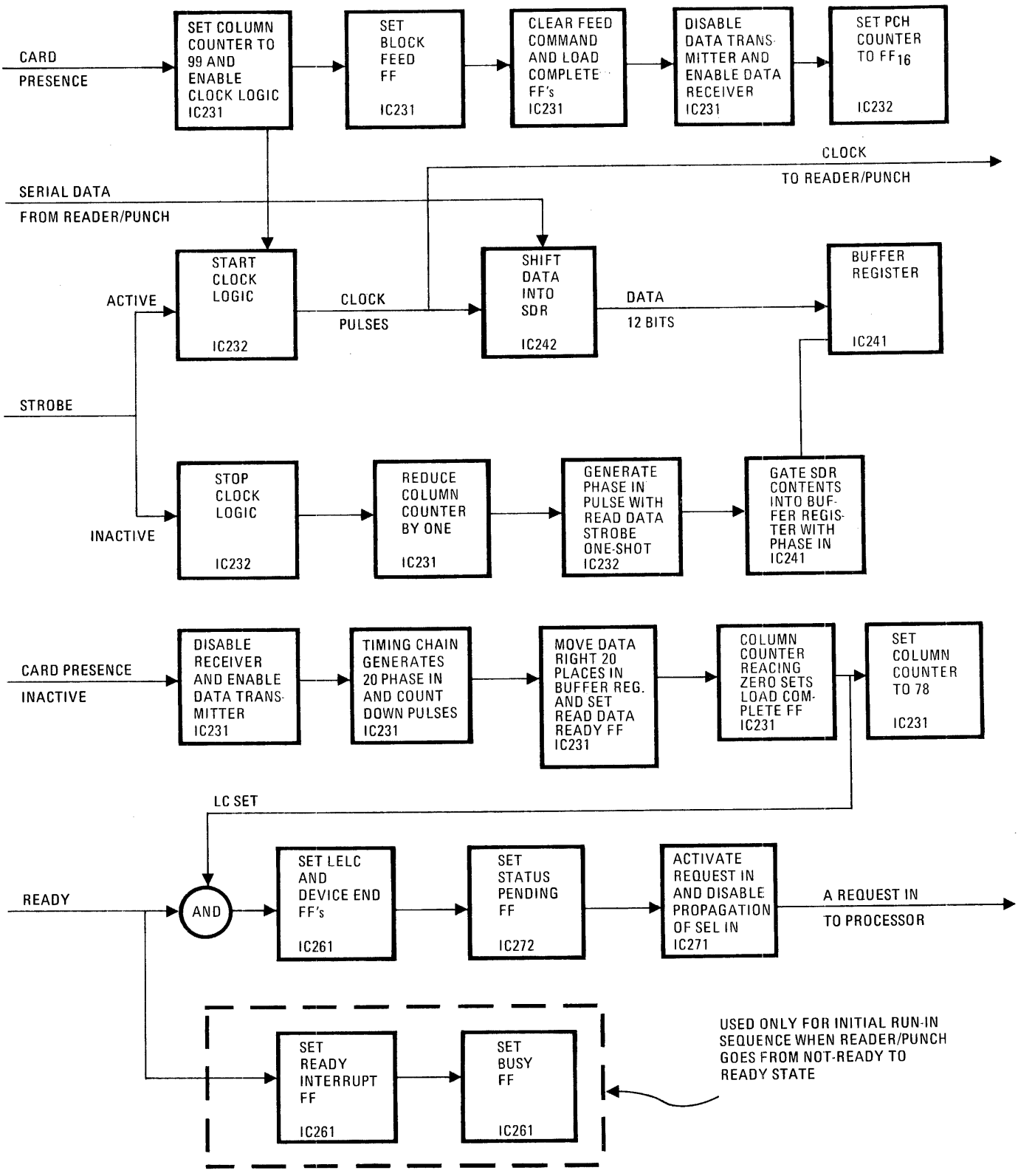


Figure 4-44. IRPA Card Load Sequence

The BORROW OUT signal also triggers the Preset Counter one-shot circuit which this time forces the Column Counter to 78 because the CARD PRESENCE signal is now inactive. The Column Counter is set in preparation for the data transfer to the channel. Only 79 shifts are required to transfer the 80 ranks of data since the output of rank 100 is constantly available. The count is set to 78; thus, for COUNT DOWN pulse 79 (all data transferred) the counter produces a BORROW OUT signal.

READY is activated by the reader/punch at the completion of the second feed cycle. The Load Complete FF set indicates the shifting is complete and the IRPA is ready to accept commands. The IRPA now prepares its status information and thus requests service. Load Complete setting clocked the Leading Edge Load Complete (LELC) causing it to set, since READY was active. LELC setting sets the Device End status FF. Status is now outstanding and for this reason the IRPA uses the Device End output to set the Status Pending FF which activates the REQUEST IN signal to the processor. REQUEST IN active also disables the propagation of the SEL IN signal through the IRPA.

In addition, the READY signal going active sets the Ready Interrupt FF, which in turn sets the Busy status FF indicating that the reader/punch and IRPA have gone from a not-ready to a ready state. Busy, when set, clears the Ready Interrupt FF after a short delay, because it will not be used to set BUSY again until the reader/punch goes not-ready and back to ready.

It is at this point that the IRPA waits until the processor responds to the REQUEST IN signal with a polling routine, which effects an adapter-initiated sequence between the IRPA and the processor. It is during this sequence that the status indication is presented indicating that the IRPA is ready.

A feed error detected by the reader/punch during initial run-in causes it to activate the FEED CHECK signal, disabling further execution of feed cycles and holding READY inactive. As a result, regardless of whether the error occurs during the first or second feed cycles, the IRPA will not activate the REQUEST IN signal at the termination of the run-in. This is the proper response, because no data is transferred to the channel during initial run-in; as such, the error is strictly motion failure which the operator can observe.

If the reader/punch detects a read error during the second feed cycle (reading the first card) it activates the ERROR signal to IRPA. The initial run-in sequence terminates correctly and an adapter-initiated sequence is started as normal. The error status information will be presented to the processor dependent upon the command issued as listed below.

- During the ending sequence of a *Read Only* or a *Read, Feed And Select Stacker* command when executed first after initial run-in.
- With *Device End* status presentation for the *Punch, Feed And Select Stacker* command.

NOTE

The first card will be offset in the stacker at the completion of any command containing a *feed* portion.

Adapter Initiated Sequence

Processor 2, in response to the REQUEST IN signal, performs a polling routine during which the IRPA performs the events within the adapter-initiated sequence as shown in Figure 4-45. During this sequence the IRPA presents its address and status indication. If the status indication is accepted by the Processor the IRPA is deselected and waits for a command via a channel-initiated initial selection sequence.

If the status indication is stacked, rather than accepted, the IRPA still is deselected at the end of the sequence, but holds the REQUEST IN signal active and continues to disable SEL IN propagation. The REQUEST IN again forces Processor 2 into performing another polling routine. The IRPA will continue to request service with the REQUEST IN signal until the status indication is either accepted or suppressed by Processor 2 during an adapter-initiated sequence.

Initial Selection

Data contained in ICRA portion, relevant to both ICRA and IRPA.

Data Transfer Sequence

During an IRPA data transfer sequence, data is either shifted out of the buffer register and sent to the processor via the Bus In lines, or data is accepted from the processor via the Bus Out lines and shifted into the Buffer register. The number of bytes transferred between the adapter and the channel depends on Processor 2 software and the data mode. For card-image mode there are two bytes per rank, or column, transferred; but for EBCDIC mode, only one byte per rank. Also, the processor may elect to send or receive less than the full 80 columns of information during a data transfer sequence.

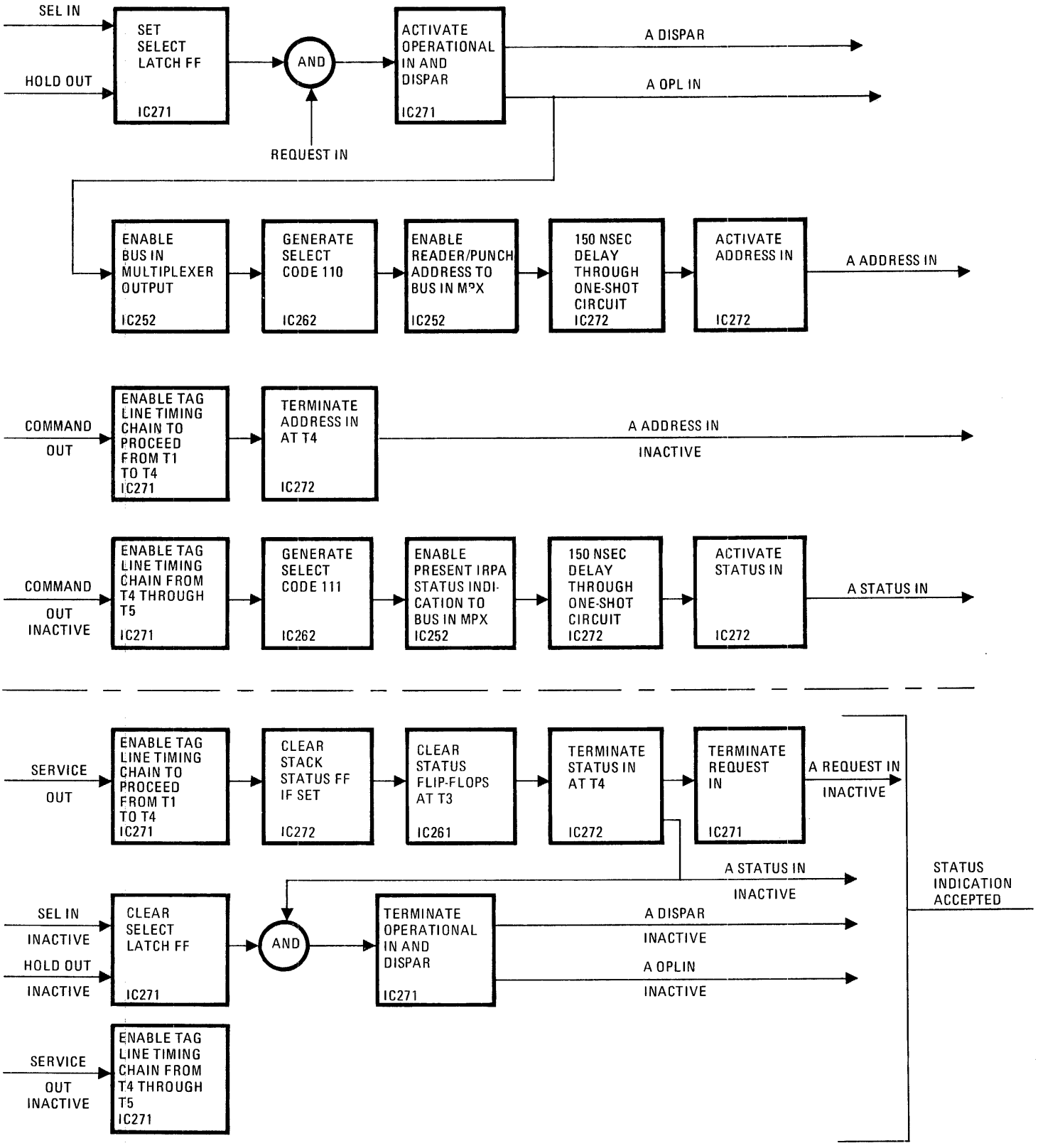


Figure 4-45. IRPA-Initiated Sequence

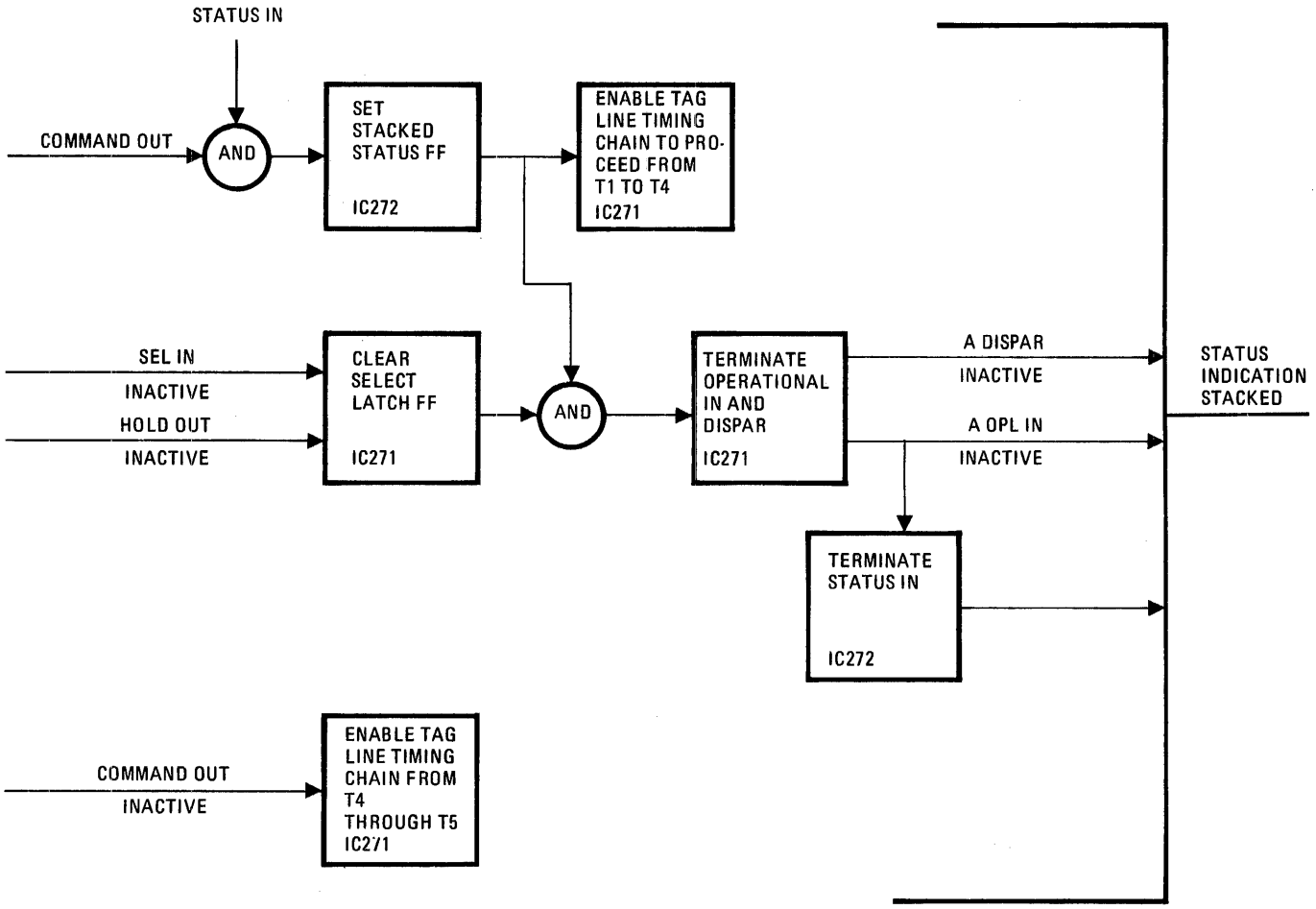


Figure 4-45. IRPA-Initiated Sequence (continued)

EBCDIC Data Transfer to Channel

The conclusion of an initial selection and the beginning of an input data transfer sequence is begun with the termination of SRV OUT by the processor after it has accepted the initial selection status indication. The Selector logic will have generated SELECT code 000 during initial selection for a *Read Only* or *Read, Feed And Select Stacker* command with the EBCDIC mode specified. SELECT code 000 enables the Card Code/EBCDIC Converter output into the Bus In multiplexer and thus to the Bus In lines. Therefore, when the data transfer starts, the proper data paths within the IRPA have already been enabled under control of the command issued.

When SRV OUT becomes inactive, it enables the Tag Line Timing Chain and sets the Performing Command FF, which is not cleared until the completion of the data transfer sequence. SRV OUT inactive also AND's with the output of Read Data Ready, which became set during initial run-in, to trigger a 600-nanosecond one-shot circuit. The trailing edge of the one-shot circuit clocks the Data Transfer Request FF, causing it to be set because the Performing Command FF is set. The output of Data Transfer Request, after a 150-nanosecond delay through a one-shot circuit, sets the Tag Line Enable FF which activates SRV IN to the processor.

At this point the first converted EBCDIC character is on the Bus In lines, and the adapter is awaiting a response from the processor (Figure 4-31). After accepting the byte, the processor responds by activating SRV OUT. The IRPA uses SRV OUT to prepare the next EBCDIC character for transfer, before indicating it is ready by terminating SRV IN. During this preparation, the contents of the Buffer register are shifted right one place and the Column Counter is decreased by one. The following events accomplish this preparation within the IRPA (Figure 4-32 shows their timing relationship):

- Activate SRV OUT (BDC circuitry)
- Activate SRV I/O signal
- Clock Data Check FF
- Set Data Request FF
- Activate NEW DATA REQUEST (NDR) signal which enables the input to the Two-Phase Timing Chain
- Clear Read Data Ready FF
- After a short delay, clear Data Request FF and terminate the NDR signal, disabling input to the timing chain
- Shift contents of Buffer register right one place
- Clear Data Transfer Request FF, which terminates the SERVICE IN and SRV I/O signal to BDC circuitry
- Reduce Column Counter by one
- Set Data Ready FF

Figure 4-32 illustrates the events which are repeated for each input transfer of a EBCDIC character: SRV OUT inactive to SRV OUT inactive again. The data transfer sequence continues, repeating the above-described events, until either it is terminated by the processor, or the Column Counter is reduced to zero.

EBCDIC Data Transfer From Channel

The start of an output data transfer sequence to the IRPA is initiated by the termination of SRV OUT by the processor after it has accepted the initial selection status indication. The Command Decode logic will have generated the *Punch* command without the CARD IMAGE signal during initial selection for a *Punch, Feed And Select Stacker* command with the EBCDIC mode specified. The inactive CARD IMAGE signal selects the EBCDIC-to-Card Code Converter output through the Punch EBCDIC/Card Image Selector (Figure 4-22). The *Punch* command routes the Punch EBCDIC/Card Image Selector output through the Punch/Read Selector to the Buffer register. Since the Bus Out lines are directly connected to the EBCDIC-to-Card Code Converter, when the data transfer begins all the correct data paths in the adapter are enabled.

When SERVICE OUT goes inactive the first time after initial selection, it causes the IRPA to set itself up for the transfer before asking for the first character, as indicated in Figure 4-46. The Performing Command FF output ANDs with a COMMAND DECODE signal from the Decode logic to trigger the Start Write one-shot circuit. Its output triggers the Preset one-shot circuit which sets the Column Counter to 99 once the Channel Write FF sets. The output from both one-shots AND together to set the Channel Write FF, which in turn sets the Block Feed FF. These events now have the adapter ready to accept data from the processor.

The adapter signals the processor it is ready for the first EBCDIC character by triggering the Punch Request one-shot circuit which sets the Data Transfer Request FF and, after a 150-nanosecond delay, activates SERVICE IN.

The processor responds to SERVICE IN by placing the EBCDIC character on the Bus Out lines and activating SRV OUT. The IRPA uses SRV OUT active as the signal to accept the character, and indicates this fact to the processor by terminating SERVICE IN. Here again the data is shifted right one place and the Column Counter reduced by one for each character accepted. The following list of events perform this acceptance task.

- Activate SRV OUT (BDC circuitry)
- Activate SERVICE IN-SERVICE OUT (SISO) signal
- At T4 time, clear Data Transfer Request FF, terminating SRV IN which terminates SISO signal
- SISO going inactive causes:
  - The Column Counter to be reduced by one
  - The PHASE IN signal which shifts the data in the Buffer register right one rank and gates the EBCDIC character sent by the processor into rank one
  - The non-zero Punch Counter to be counted up by 1 if the EBCDIC character accepted is not all 0's
  - The Channel Write FF to clear after the 80th byte.

The IRPA now waits for the processor to terminate SRV OUT, and place the next character on the Bus Out lines. The adapter uses inactive SRV OUT to simply trigger the Punch Request one-shot circuit which leads to the activation of SERVICE IN. Figure 4-47 shows the timing relationship of these events.

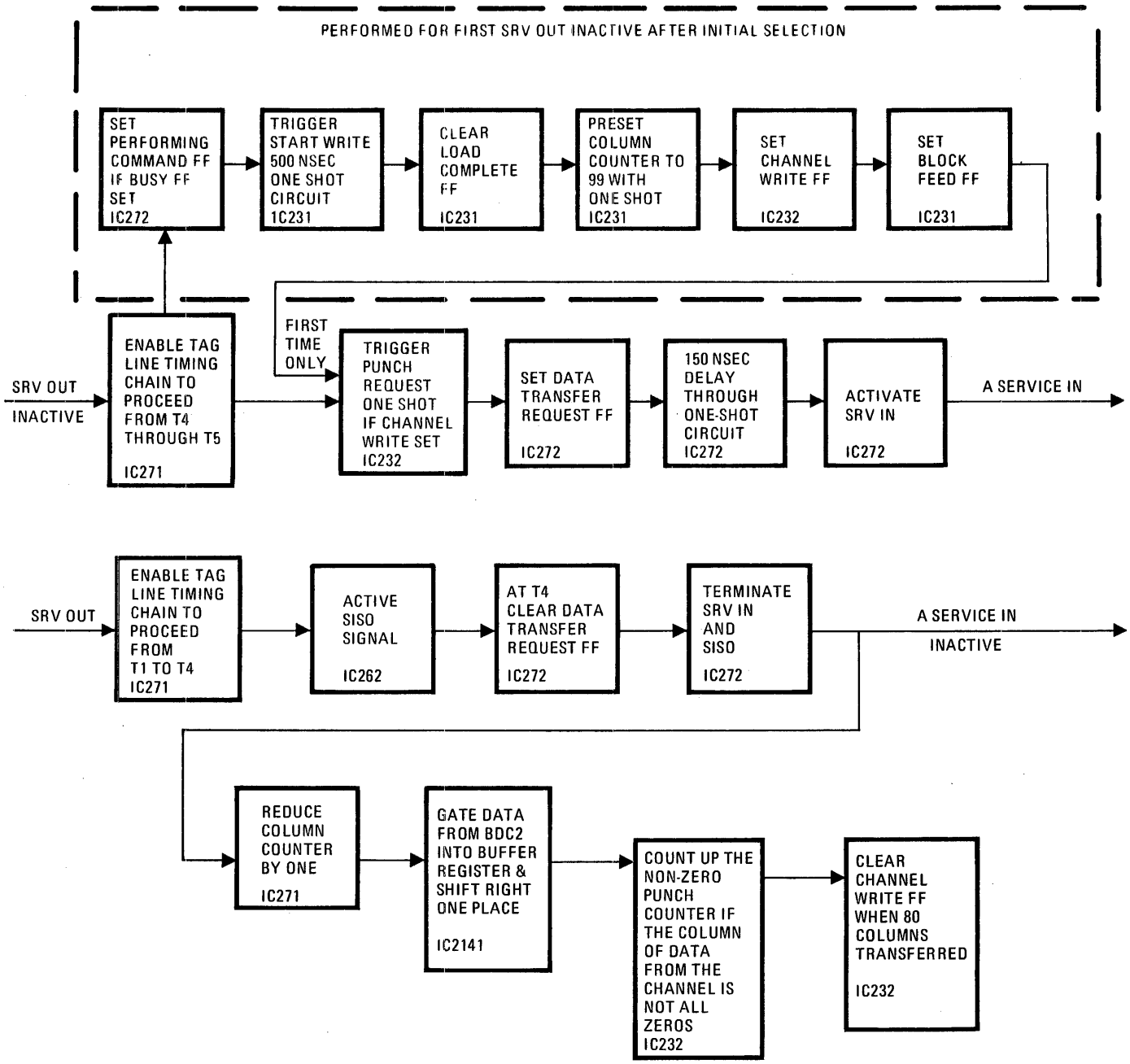


Figure 4-46. IRPA Output Transfer Sequence for One EBCDIC Character

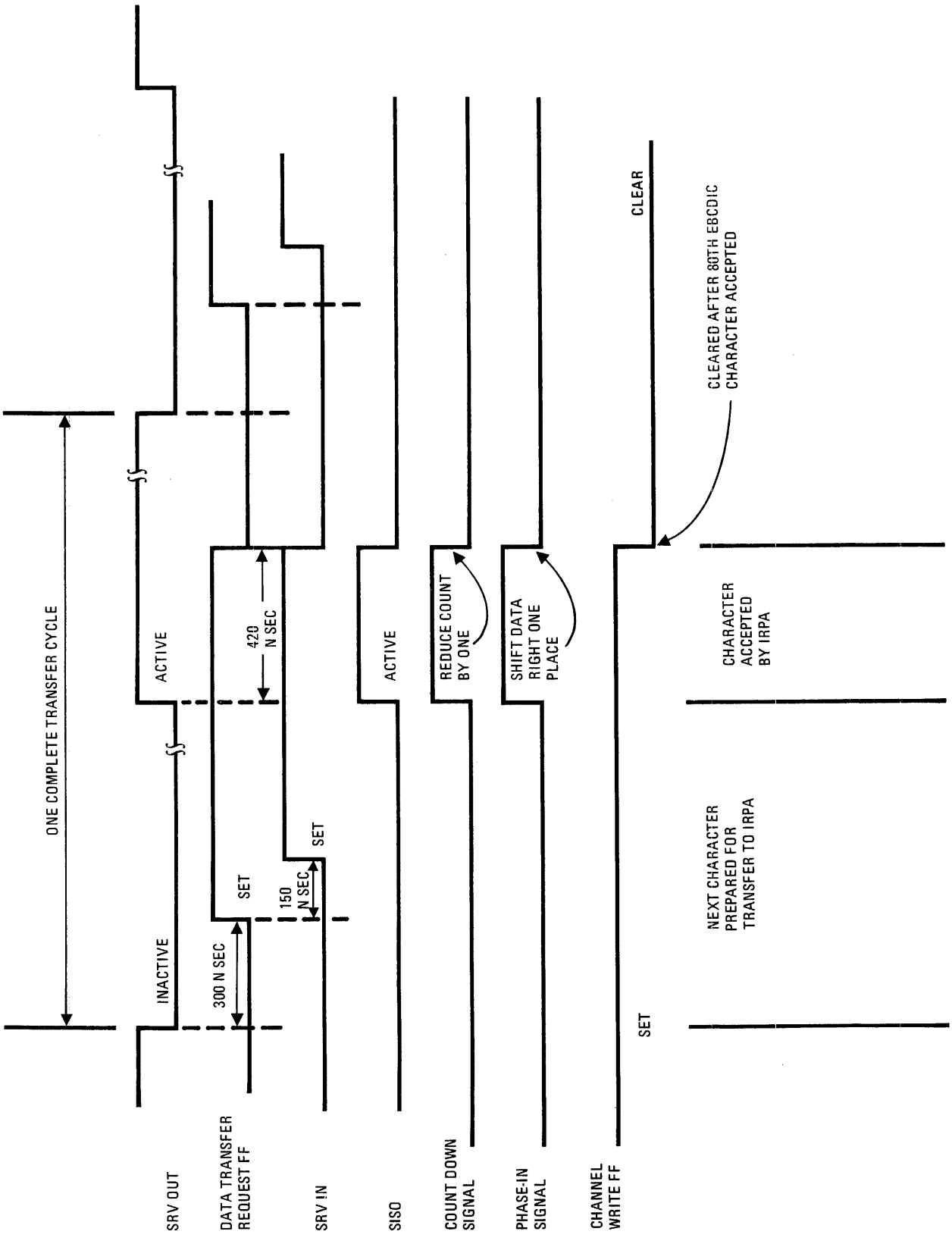


Figure 4-47. Timing Chart, IRPA Input, EBCDIC Characters

These events, from SRV OUT inactive to SRV OUT again inactive, are executed once for each EBCDIC character transferred from the channel to the IRPA. The output transfer continues until either terminated by the processor or until the Column Counter is reduced by 80 (80 bytes transferred).

Card-Image Data Transfer To Channel

During the initial selection for either *Read* command with card-image mode specified, the Command Decode logic activates *Card Image* as well as the *Read* signals. The IRPA also clears the Even Byte FF during initial selection. The Select Code logic thus generates the SELECT code 001 because of the cleared Even Byte FF and the CARD IMAGE signal. The SELECT code 001 routes the odd byte (bits 12 to 3) of the Buffer register to the Bus In multiplexer and thus to the Bus In lines.

The data transfer starts when the processor terminates SRV OUT after the IRPA-terminated STAT IN. As shown in Figure 4-31, the IRPA reacts to the inactive SRV OUT the first time the same as for an EBCDIC read. The odd byte is already on the Bus In lines; thus the IRPA simply indicates it's ready by activating SRV IN.

After the processor has accepted the data it activates SRV OUT. The IRPA responds by terminating SRV IN and preparing the even byte of the first column of card-image data for transfer. When SRV IN goes inactive, it sets the Even Byte FF, which forces the Select Mode logic to generate the SELECT code 010. This code enables the even byte (bits 4 to 9) of the Buffer register into the Bus In multiplexer.

The processor responds by terminating SRV OUT, causing the IRPA to again activate SRV IN; this indicates that the second byte is on the Bus In lines ready for transfer.

The processor accepts the even byte and then activates SRV OUT. The IRPA responds to SRV OUT by preparing the next column of card-image data for transfer. SRV OUT generates the NEW DATA REQUEST signal since the Data Req FF is set. The NDR signal enables the input to the two-phase timing chain and clears the Data Rdy FF; this disables the NDR signal, thus allowing only one pass through the Two-Phase Timing Chain. The PHASE IN signal shifts the contents of the Buffer register right one place, while the output of the second one-shot circuit sets the Read Data Rdy FF and counts down the Column Counter by one.

The adapter next clears the Data Xfr Request FF, which terminates the SRV IN signal. SRV IN going inactive clears the Even Byte FF, forcing the SELECT code 001. The odd byte of the second column of card image data is now enabled into the Bus In Multiplexer ready for transfer. The processor responds by terminating SRV OUT to start the transfer of the second column of data.

The events shown in Figure 4-31 will be repeated once for each column of card-image data until the data transfer is terminated.

Card-Image Data Transfer From Channel

The Command Decode logic generates the CARD IMAGE signal as well as the *Punch* command during initial selection for a *Punch, Feed And Select Stacker* command with card-image mode specified. The CARD IMAGE signal routes the Card Image Odd Byte register output and the Bus Out lines through the Punch EBCDIC/Card Image selector (Figure 4-22). The *Punch* command routes the card-image data through the Punch/Read Selector to the Buffer register. The Even Byte FF is also cleared by the adapter.

The data transfer begins when the processor terminates SRV OUT in response to the termination of STAT IN. The IRPA, in response to SRV OUT inactive the first time, performs the same ready operations as for an EBCDIC output transfer (Figure 4-47), before asking for the first data byte by activating SRV IN.

The processor responds by placing the odd byte of column 1 data on the Bus Out lines and activating SRV OUT. The adapter activates the SISO signal which generates the CARD IMAGE CLOCK signal since the Even Byte FF is clear. This CARD IMAGE CLOCK signal gates the six bits of the byte into the Card Image Odd Byte register. At time T4, the adapter clears the Data Transfer Request FF which terminates SRV IN to the processor, indicating the byte has been accepted; terminates the SISO signal, which disables the clock signal to the Odd Byte register; and sets the Even Byte FF in preparation for the next byte transfer.

The processor responds by terminating SRV OUT and the adapter responds by activating SRV IN. The processor now reacts by placing the second (even) byte of Column 1 data on the Bus Out lines, and then activates SRV OUT which activates the SISO signal in the adapter. At T4 time the adapter clears the Data Transfer Requests FF, which terminates the SRV IN, causing the Even Byte FF to clear. SRV IN going inactive counts down the Column Counter by one and terminates the PHASE IN signal, which clocks the Card Image Odd Byte register contents and the Bus Out data into rank 1 of the Buffer register and shifts the rest of the register right one place. The first column of card-image data is now in the Buffer register and the Column Counter is reduced by 1. SRV IN terminating indicates to the processor that the even byte has been accepted; SRV IN also clocks the Non-Zero Punch Counter, causing it to count up by 1 if all 12 bits of column 1 of the card-image data are not zeros. (This counter is again used during the card punch sequence; its full function will be explained in the description of that sequence.)

The events for both the odd and even byte transfer, as shown in Figure 4-48, are again repeated for each column of data transferred until the process is terminated.

Ending Sequence

Terminated Input

Data contained in previous ICRA section for those elements common to both ICRA and IRPA.

Status Accepted – The IRPA uses SRV OUT, in response to STAT IN, to complete the ending sequence by generating the FEED CMD signal if the command being executed is a *Read, Feed And Select Stacker*. The FEED CMD signal sets the Feed FF which generates the FEED command to the reader/punch to start a *read* cycle. The FEED FF also enables the OFFSET signal if an offset condition exists. Once the next *read* cycle has been started, the adapter clears its status flip-flops and then terminates STAT IN by clearing the Status Pending FF at T4.

Once the processor has terminated HLD OUT, the adapter completes the deselection by clearing the Select Latch FF which terminates the OPL IN and DISPAR signals because the Status Pending FF is clear. OPL IN inactive allows the propagation of the SEL IN signal.

The processor responds by terminating SRV OUT, completing the ending sequence.

Status Stacked – The processor may stack the status indication(s) during an ending sequence by a response of CMD OUT to STAT IN. In this case, the IRPA still becomes deselected; however, it does not clear the status flip-flops, and it requests service. As shown in Figure 4-45, the adapter sets the Stacked Status FF in response to CMD OUT, and, since the Status Pending FF is set, activates the A REQUEST IN signal to the processor. This REQUEST IN signal is asking the processor for a polling sequence to relieve the adapter of its status indication. The adapter will continue to ask for service until the processor responds. The processor also terminates SEL IN and HLD OUT just as if the status indication had been accepted. The adapter uses this termination to deselect itself, as shown in Figure 4-45.

Processor-Terminated Output

The response of CMD OUT to SRV IN during an output data transfer also forces the IRPA into an ending sequence. The adapter reacts to CMD OUT whenever it goes active, whether in the EBCDIC mode or after either byte transfer in the card-image mode. Figure 4-49 shows the sequence of events within the adapter for a channel-terminated output data transfer.

The adapter uses CMD OUT to prepare for the card punch sequence before entering the ending sequence. In preparation for output to the punch, the adapter shifts the Buffer register data right and reduces the count until the first column of data resides in rank 100 of the Buffer register. When CMD OUT goes active, it clears the Channel Write FF. Its cleared output, along with the Load Complete FF clear, starts the Two-Phase Timing Chain. The PHASE 1 output pulse generates the PHASE IN signal which shifts the Buffer register data right one place. The PHASE 2 output pulse (COUNT-DOWN pulse) reduces the Column Counter by 1.

The Two-Phase Timing Chain cycles through the two phases, moving the data and reducing the count until the Column Counter generates a BORROW OUT indicating column 1 data is in rank 100. The number of timing chain cycles will depend upon the number of columns of data sent by the processor during the output data transfer. The BORROW OUT signal disables the timing chain by setting the Load Complete FF. Load complete going set also sets the Channel End FF through the setting of the Leading Edge Load Complete Punch FF.

While the adapter was preparing for card punch sequence, it also responded to the processor by clearing the Data Transfer Request FF at T4, which, in turn, terminated SRV IN. The processor responds by terminating CMD OUT, which indicates the starting of the ending sequence. This may happen before the adapter has completed its card punch sequence preparation since the number of passes through the timing chain depends on the number of columns to be punched on the card. Regardless of when CMD OUT goes inactive, the adapter does not start the ending sequence until the Channel End FF is set as shown in Figure 4-49.

When the Channel End FF is clear and CMD OUT goes inactive, the adapter first clears the Performing Command FF because the data transfer portion of the command is complete. The Status Pending FF is set which causes the Select Code logic to enable the IRPA Status Byte through the multiplexer to the Bus In lines. After a 150-nanosecond delay the adapter activates STAT IN.

Once the processor has accepted the Status Byte, it responds by activating SRV OUT, and deselects the adapter by terminating SEL IN and HLD OUT. The adapter, in response to SRV OUT, starts the card punch sequence and then completes the deselection as shown in Figure 4-49. Of course the Status Byte may also be stacked by the processor, in which case the adapter would still become deselected, but would activate REQUEST IN and would not start the card punch sequence.

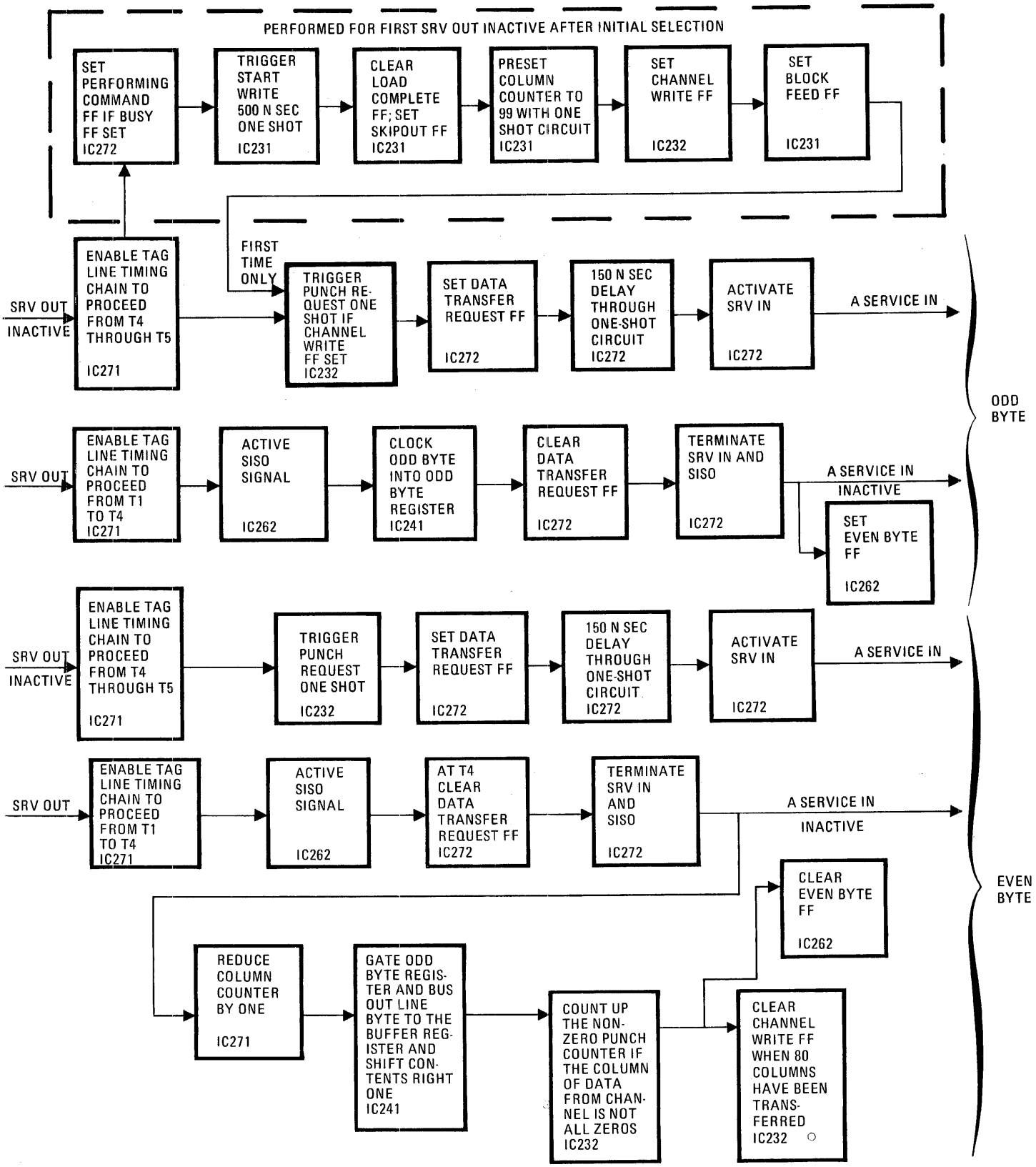


Figure 4-48. IRPA Card-Image Output Transfer Sequence

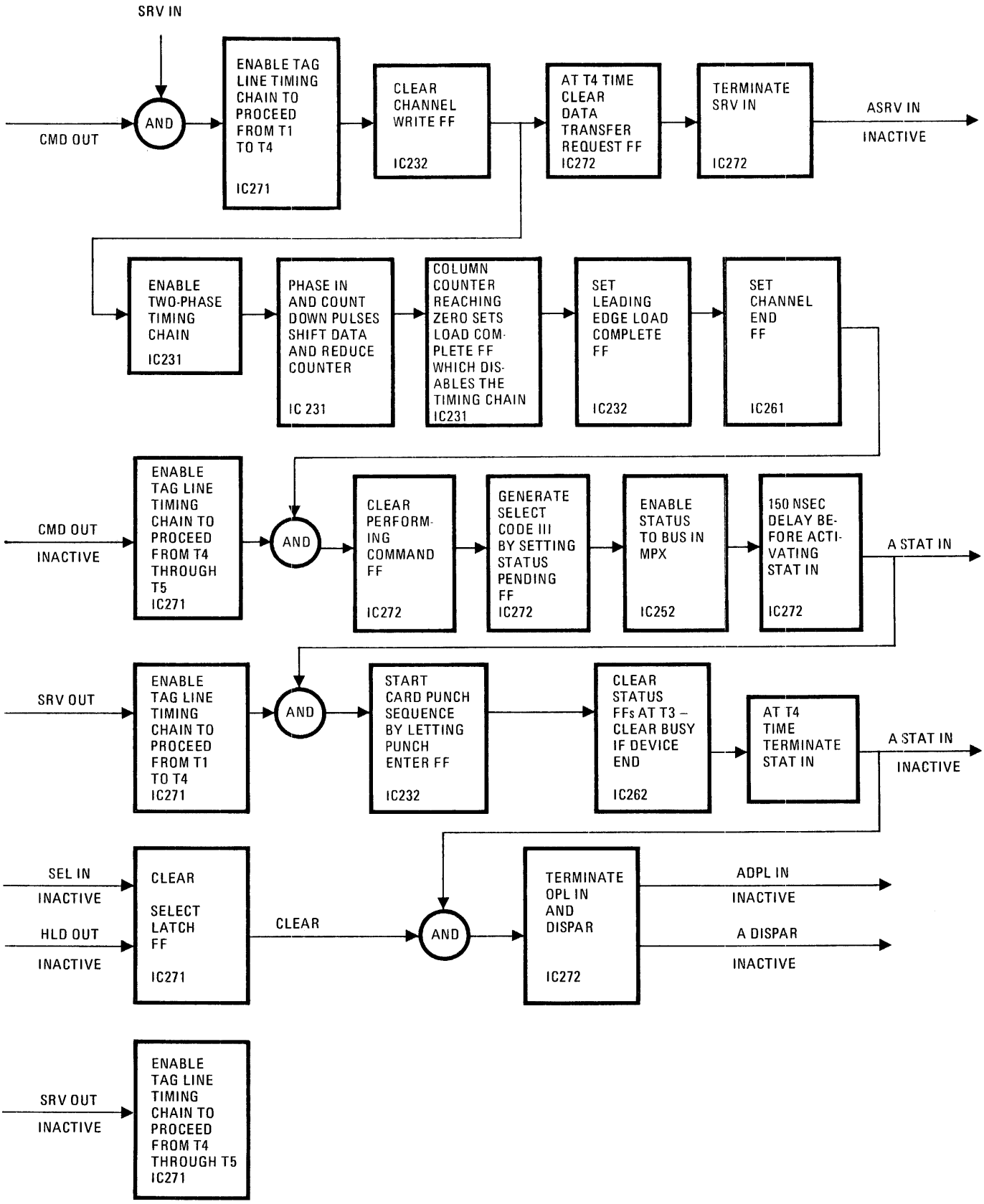


Figure 4-49. Processor-Terminated Output Ending Sequence

Adapter-Terminated

The IRPA will terminate either an input or output data transfer when 80 columns of data transfer have been transferred, even if the processor attempts to continue the transfer. The Column Counter being reduced to zero triggers the termination of an input data transfer, whereas the clearing of the Channel Write FF, when 80 columns have been accepted from the channel, stops an output transfer.

The sequence of events for both an EBCDIC and a card-image adapter-terminated input are identical to those performed by the ICRA, previously described.

The adapter terminates an output data transfer by monitoring the Column Counter and clearing the Channel Write FF after column 80 data is accepted. During transfer of the 80th EBCDIC character from the processor, the COLUMN 80 signal is active. When SRV IN goes inactive it clocks the Channel Write FF, which clears because of the active COLUMN 80 signal. If in the card-image mode, the adapter will not clear the Channel Write FF until after the even byte of the 80th column has been accepted. This is true because in the card-image mode the adapter does not generate the COUNT-DOWN or PHASE IN signal, or clock the Channel Write FF, unless the Even Byte FF is set.

As indicated in Figure 4-50, once the Channel Write FF is cleared, the adapter begins preparing for the card punch sequence and the status presentations. The Two-Phase Timing Chain runs, generating PHASE IN and COUNT-DOWN pulses, until the data in the Buffer register is moved over to rank 100 (ready for output to the punch). It then sets the Channel End FF and waits for the processor to terminate SRV OUT in response to the termination of SRV IN during the last byte accepted (Figure 4-46 for EBCDIC and Figure 4-48 for card-image).

When the processor terminates SRV OUT, the adapter places the Status Byte on the Bus In lines and actives STAT IN. The processor activates SRV OUT after accepting the Status Byte. The adapter, in response, starts the card punch sequence and then completes the ending sequence.

Card Punch Sequence

As illustrated in Figure 4-51, each *Punch, Feed And Select Stacker* command generates an initial selection, output data transfer, ending, card punch, card load and adapter initiated sequence. The card punch sequence is started by the adapter during the ending sequence following the output data transfer. During the card punch sequence, the data in the Buffer register is transferred to the punch unit and punched into the card; the card is then placed in the output stacker.

Upon completion of the card punch sequence, the next card in the input magazine (hopper) is read and the data is routed to the Buffer register. This is in anticipation that the next command might be one of the two *read* commands. If the next command is another *punch* command, the data read is not used.

The card punch sequence is started during the ending sequence (Figure 4-49) when the processor responds with SRV OUT after accepting the adapter's ending Status Byte. The adapter completes the ending sequence while performing the card punch sequence.

Setting the Punch Enter FF triggers the Punch Clock one-shot circuit (Figure 4-52). Its output pulse counts **down** the Non-Zero Punch Counter by 1, if the first column of data is not all 0's. The adapter determines this by monitoring rank 100 of the Buffer register where the first column of data now resides. The Non-Zero Punch Counter was **advanced** by 1 for each column of data placed in the Buffer register during the output transfer. During the card punch sequence the counter will be reduced by 1 for each non-zero column sent to the reader/punch. Thus, when the counter reaches zero, all columns containing data will have been punched in the card and the punch sequence may terminate.

The trailing edge of the PUNCH CLOCK pulse clocks the Punch Enter FF to the clear state, since the set input was removed during the ending sequence when STAT IN was terminated. The trailing edge also triggers the Deskew one-shot circuit and gates the first column of data from the Buffer register to the Serial Data register.

When the Non-Zero Punch Counter is zero, the termination of the PUNCH CLOCK pulse terminates its BORROW OUT signal which sets the Skipout FF. Skipout set indicates all data has been sent to the reader/punch.

The trailing edge of the Deskew one-shot circuit output sets the Punch MUX FF and shifts the Buffer register contents right one place by controlling the PHASE IN signal. The Deskew one-shot circuit delay allows time for line propagation of the first serial data bit from the SDR. The Punch MUX FF set enables the adapter Clock logic which develops the CLOCK signal used for the serial data transfer to the punch.

The Clock logic consists of two 500-nanosecond one-shot circuits cross connected. Thus, once enabled by the Punch MUX FF, it cycles until disabled again, producing a series of 1-microsecond CLOCK signals as shown in Figure 4-41A. The CLOCK signal is used by the adapter to gate the serial data from the SDR register. The signal is also used by the punch to gate the serial data into its SDR.

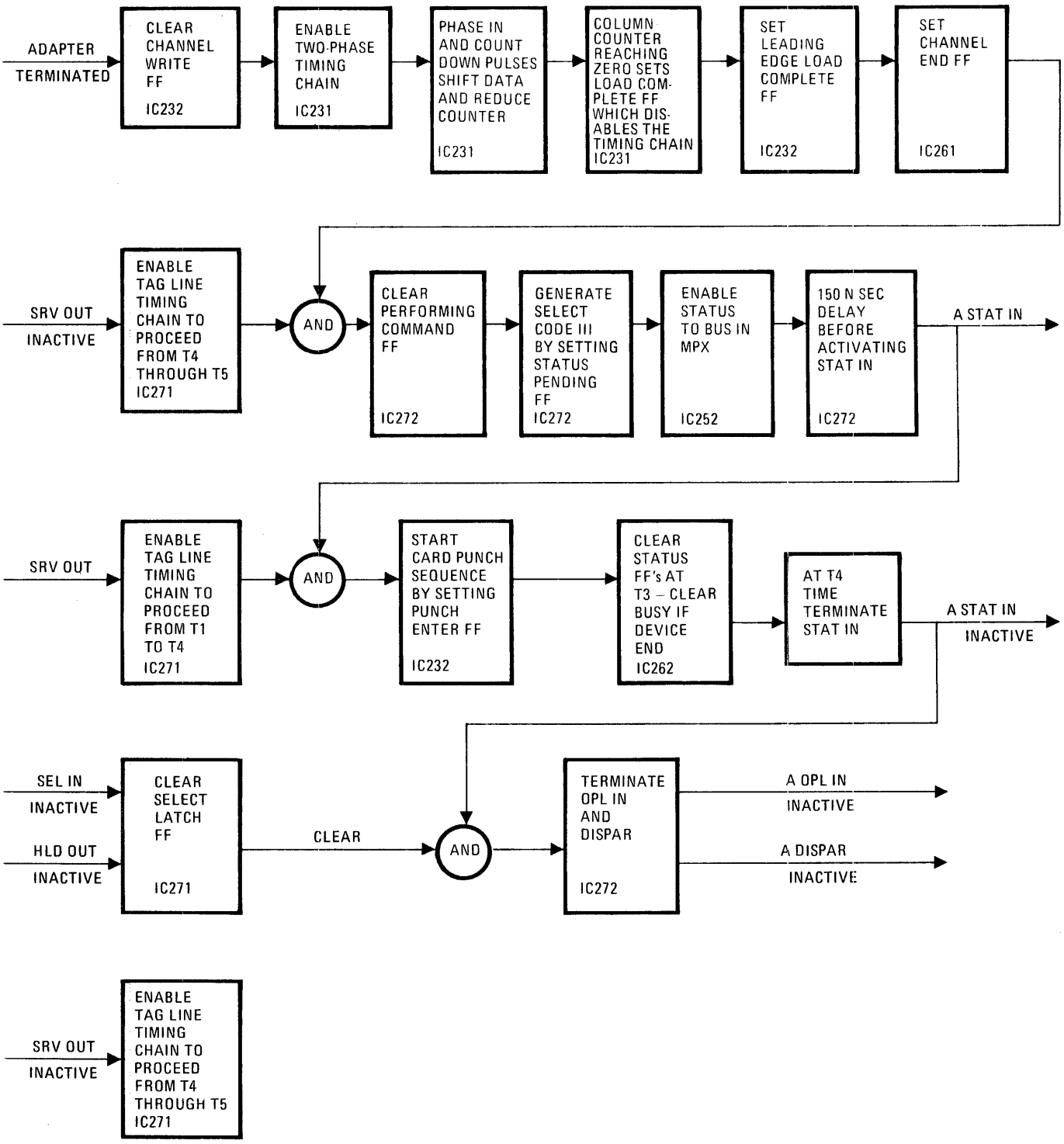


Figure 4-50. Adapter-Terminated Output Ending Sequence

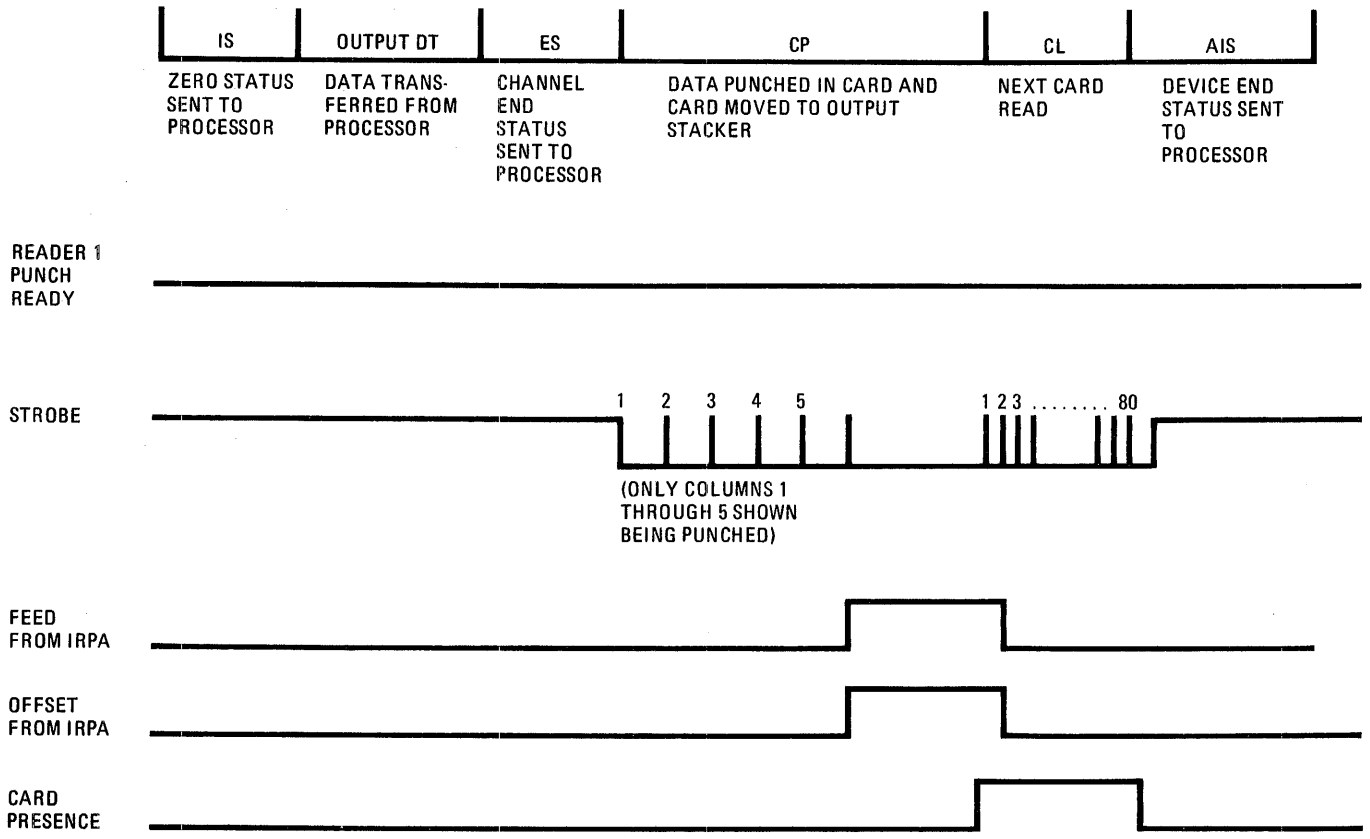


Figure 4-51. Timing Chart, Punch Sequences

The punch unit monitors the serial data from the adapter, and when twelve bits have been received it terminates the STROBE signal. The adapter uses STROBE going inactive to clear the Punch Mux FF which disables the clock logic. The punch then gates the data out of its SDR and punches it in the card. Once the punch is ready to accept the next column of data it activates the STROBE signal again.

STROBE going active ANDs with the LELCP FF-set output to generate a one-shot pulse which sets the Punch Enter FF again. This starts another pass through the card punch sequence, transferring the next column of data to the punch unit. The card punch sequence will be repeated for each column of data until all columns have been punched, indicated by the setting of the Skipout FF. STROBE going active after Skipout is set sets the Feed FF in the adapter as well as the Punch Enter FF. Punch Enter is set, starting another (this time non-functional) pass through the card punch sequence since LELCP FF is not cleared until after the Feed FF is set.

Setting the Feed FF also activates the FEED signal to the punch unit along with OFFSET if either of the following conditions exists:

- *Offset* bit in the command register set
- Echo check failure during the card punch sequence

The reader/punch responds by performing a feed cycle, during which the card just punched is moved to the output stacker, the card in the Pre-Read station is read and moved to the Wait Punch station and the next input card is moved to the Pre-Read station. The IRPA terminates STROBE upon completion of the non-functional card punch sequence.

During the feed cycle when the card enters the Read station, the punch unit activates CARD PRESENCE. The IRPA uses the CARD PRESENCE signal to clear both the Feed and Offset FF's and to start a card load sequence (Figure 4-44). Thus, the adapter loads the Buffer register with the next card data during the

punch unit feed cycle. Upon completion of the card load sequence, the adapter starts an adapter-initiated sequence in order to present the *Device End* status indication to the processor that both the adapter and punch unit are free to accept the next command.

NON-MOTION COMMANDS

The No-Operation Test I/O, and Sense commands are implemented in the IRPA in an identical fashion to that in the ICRA.

The *Read Only* command requires an initial selection, data transfer, and ending sequence. During the data transfer sequence, the Buffer register contents are transferred to the processor. During the ending sequence, the Feed FF is not set, thus not starting a feed cycle in the reader/punch nor a card load sequence in the IRPA. The remainder of the ending sequence events is the same as illustrated in Figure 4-33.

ERROR HANDLING

The relationship between the time an error occurs and when that error is indicated in the IRPA Status Byte is illustrated in Figure 4-53. Error handling during initial run-in is not shown as it was previously discussed in this section. The *Read, Feed, And Select Stacker* command is not shown because it is identical to the *Read and Feed* command discussed for the ICRA.

CONTROL STORAGE LOAD

The Control Storage Load (CSL) logic in the IRPA allows the shared resources to load control storage from the reader/punch directly, without using the Basic Data Channel Processor logic (Figure 4-40). The CSL logic in the IRPA functions much as it did in the ICRA, as both adapters use the EP and DG boards.

The IRPA uses the same signals as the ICRA to communicate with the shared resources during CSL operations. Just as with the ICRA, a CSL operation in the IRPA is started in response to a DOA signal from the processor. During a CSL operation, the IRPA alternately performs card load and CS data transfer sequences until it is terminated because of one of the three same conditions as with ICRA.

Once the initial run-in is completed, the reader/punch is ready to perform a CSL operation. In response to the DOA signal, the IRPA starts the operation by performing a control storage data transfer sequence during which the data in the Buffer register is transferred to the shared resources. This is followed by a card load sequence during which the next card is read and placed in the Buffer register. These events continue until all cards are read or an error is detected (Figure 4-39).

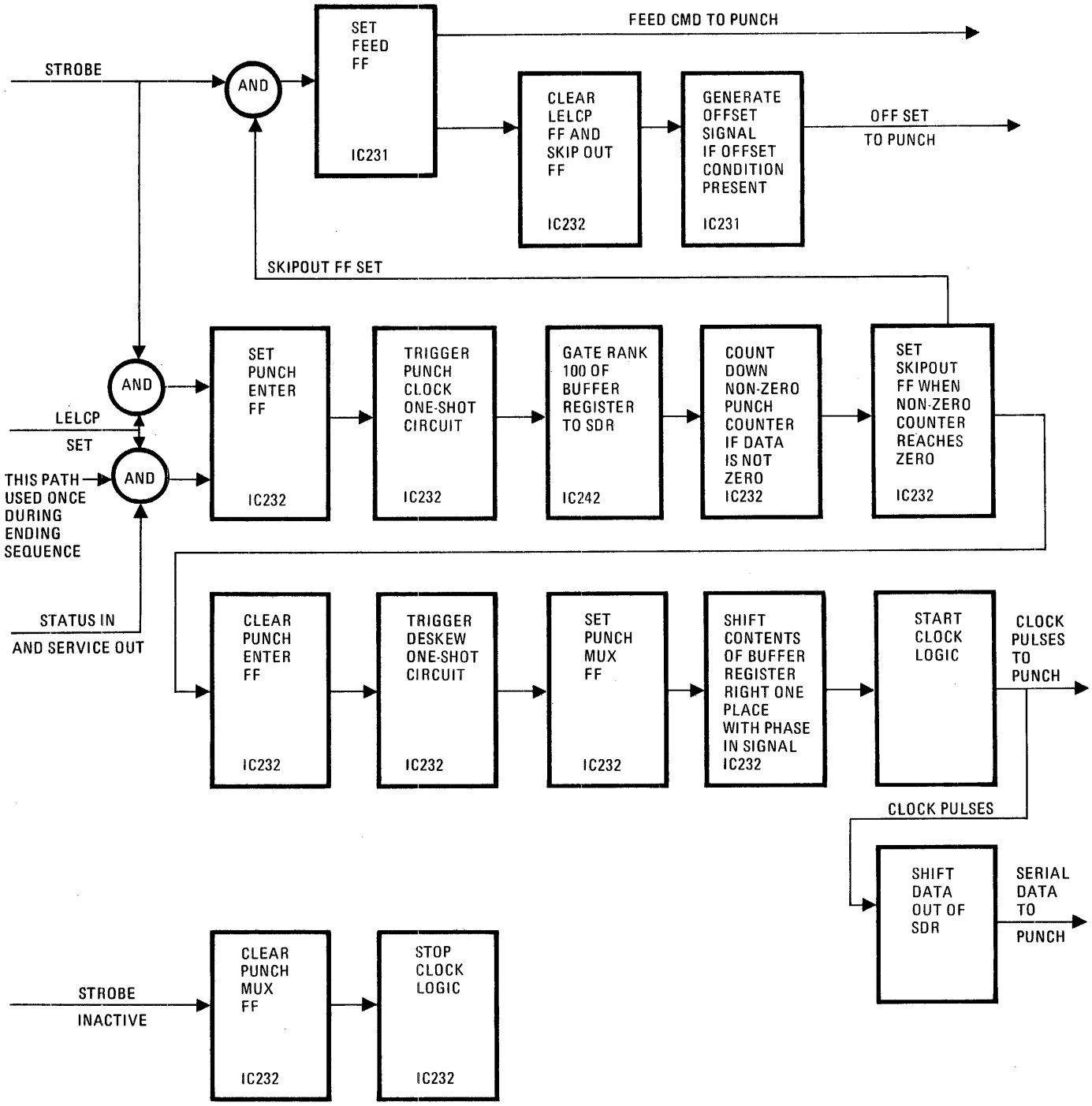


Figure 4-52. Card Punch Sequence

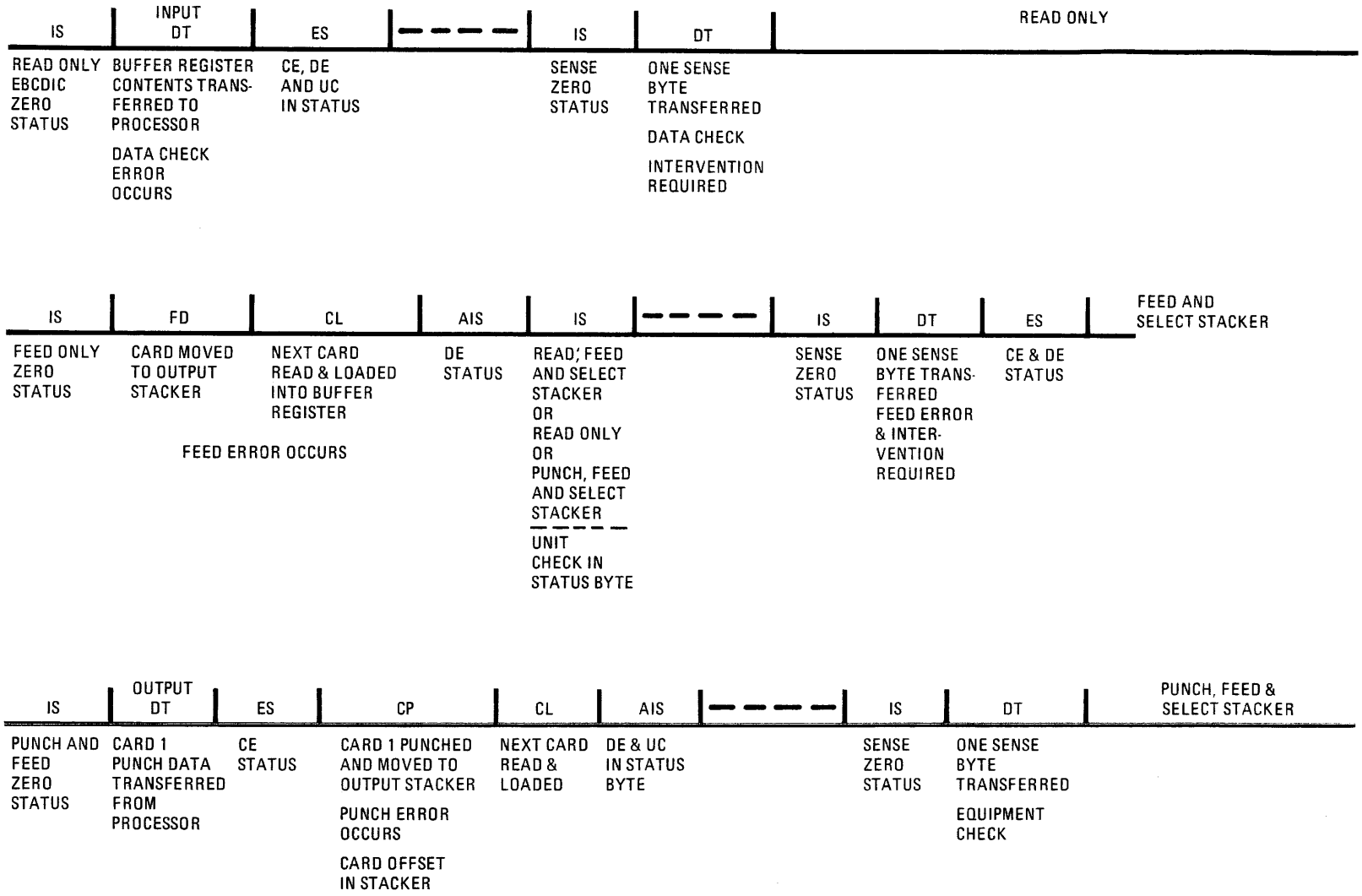


Figure 4-53. Relationship of Error Occurrence and Indications

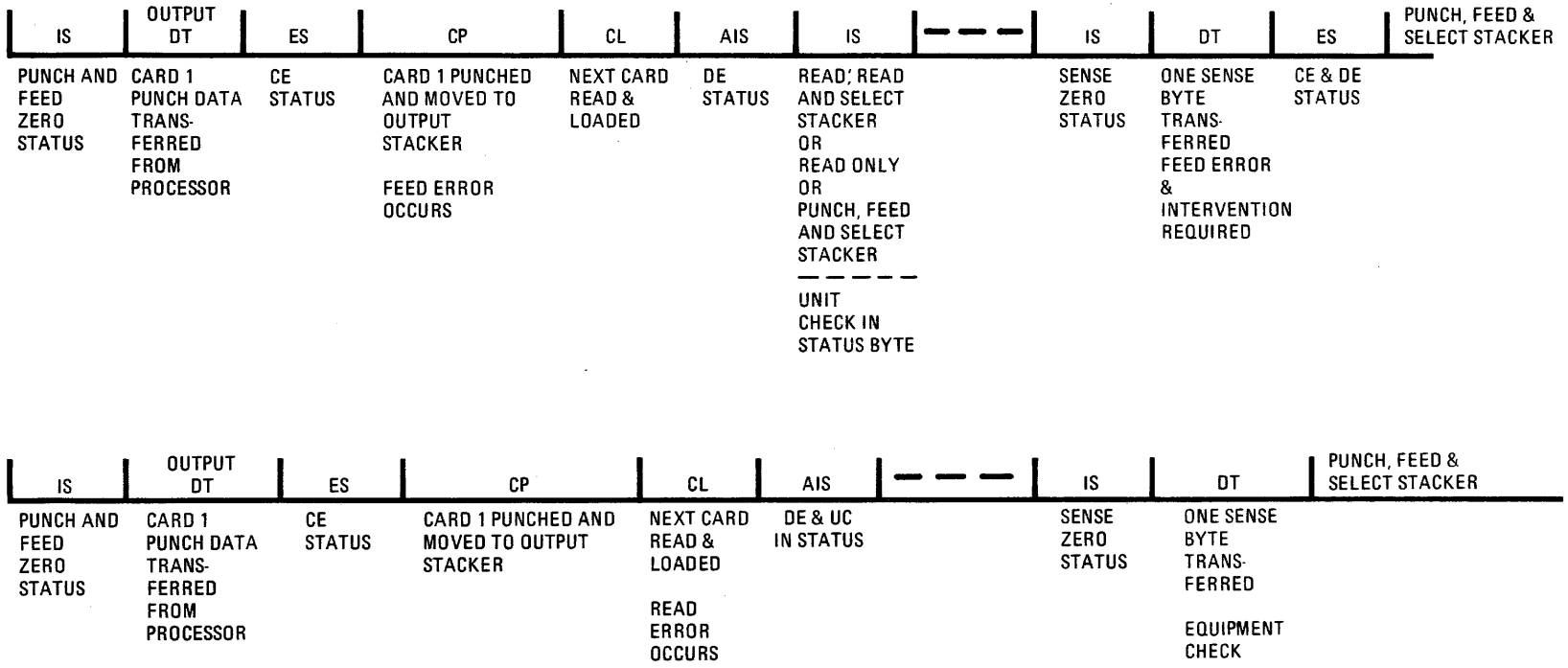


Figure 4-53. Relationship of Error Occurrence and Indications (continued)

APPENDIX 4A

INITIAL SELECTION TIMING

Figure 4A1 shows the relative timing for initial selection.

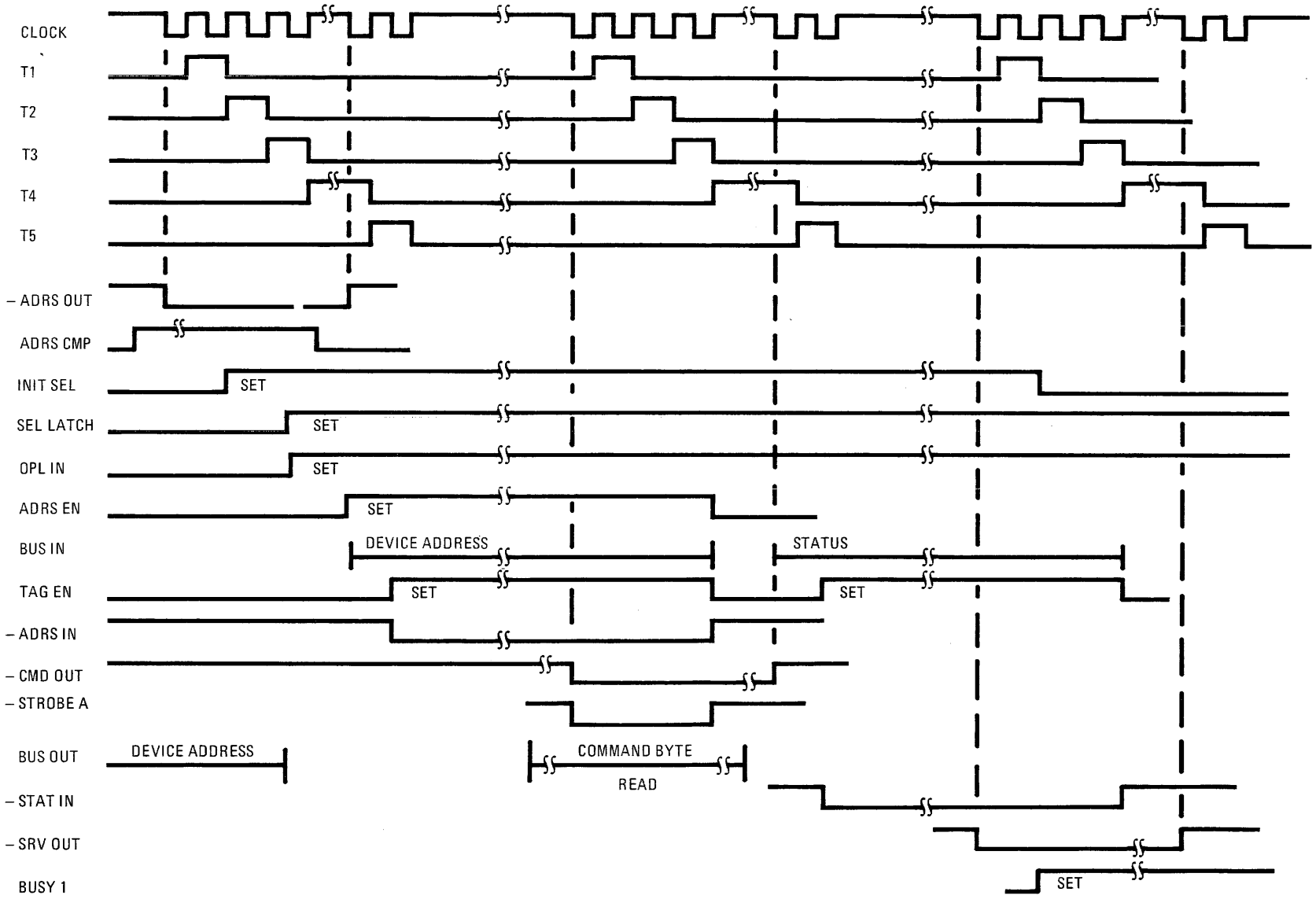


Figure 4-A1. Initial Selection Timing

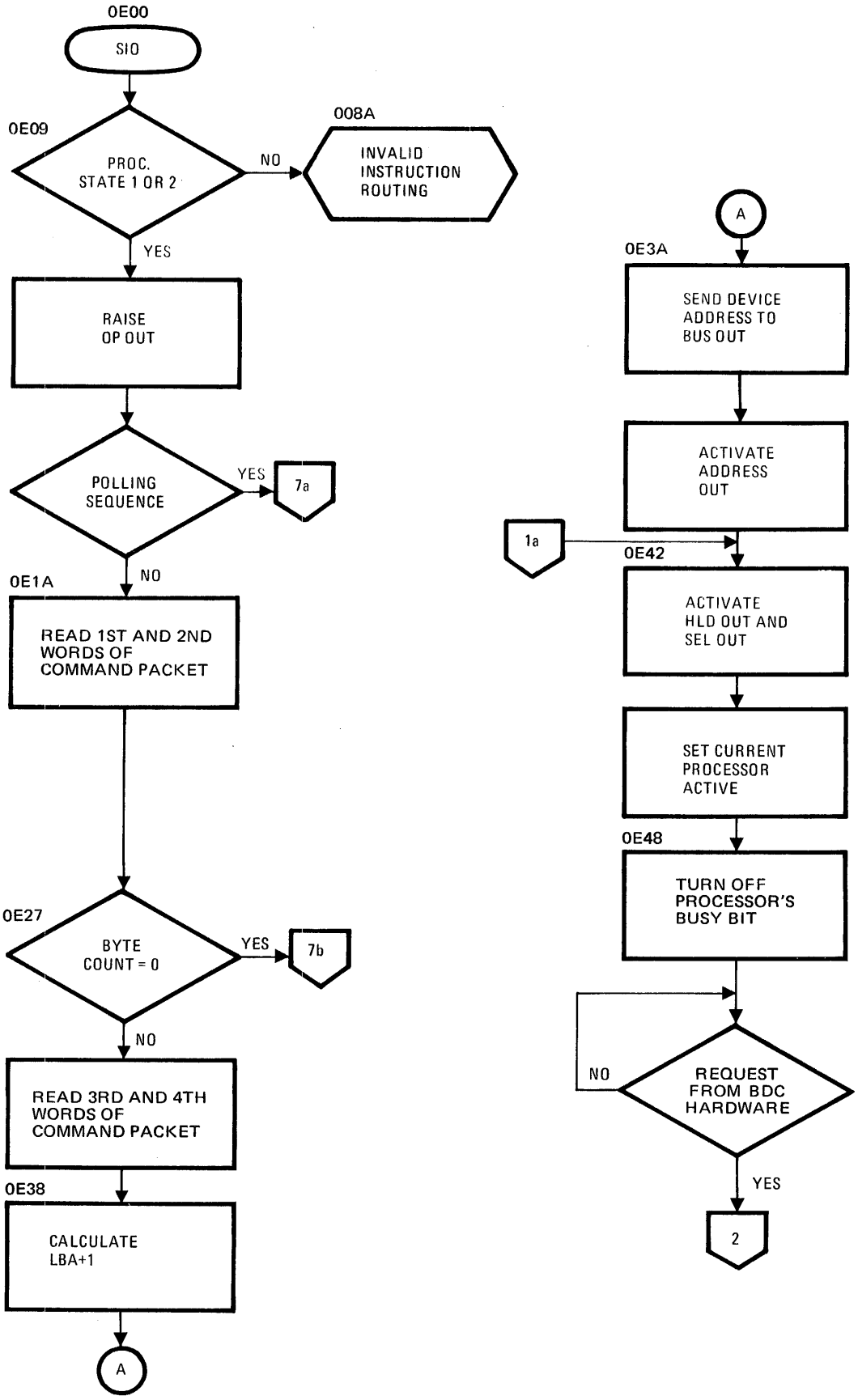


Figure 4-B1. Flowchart, SIO Micro-Instruction Routine

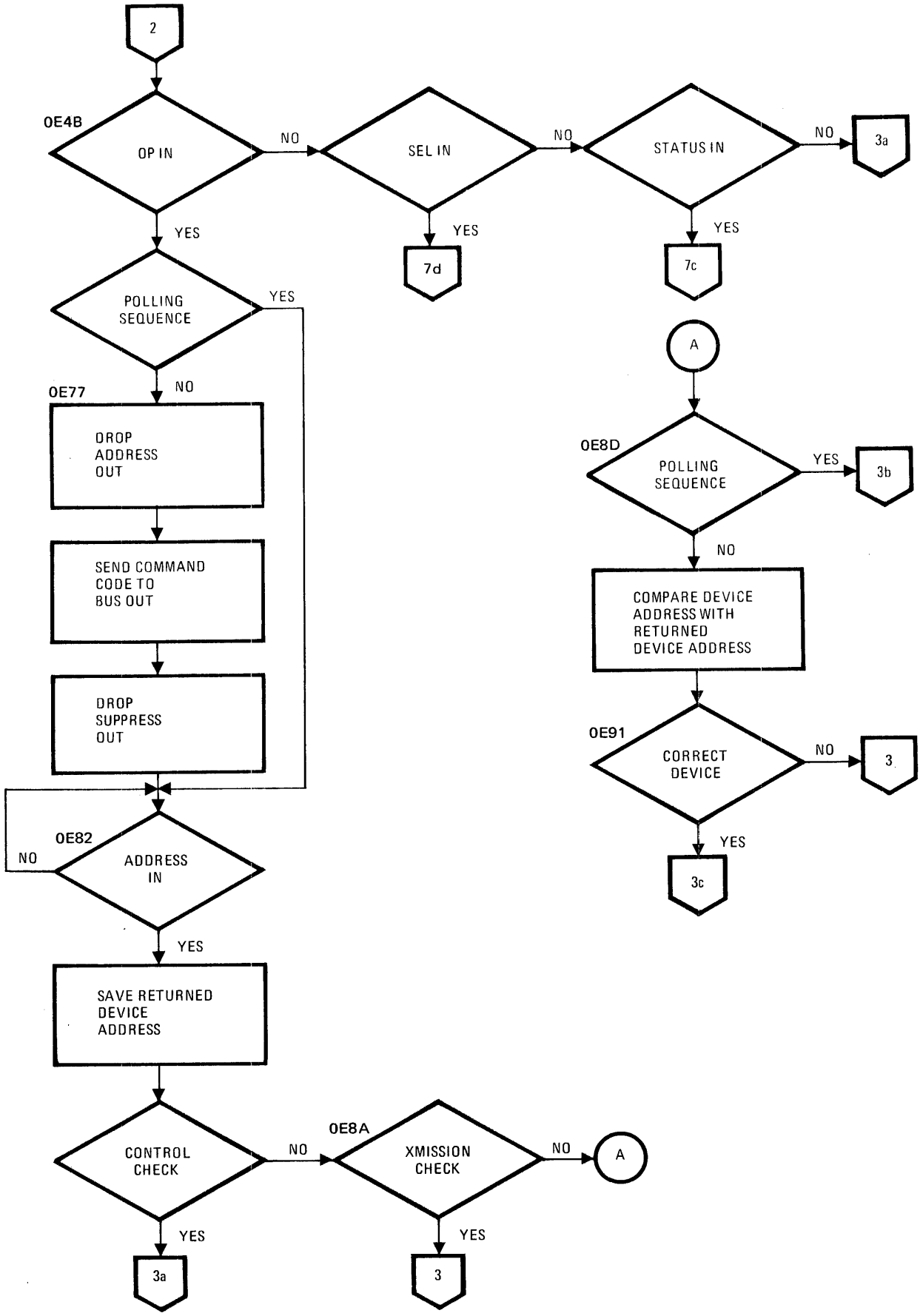


Figure 4-B1. Flowchart, SIO Micro-Instruction Routine (continued)

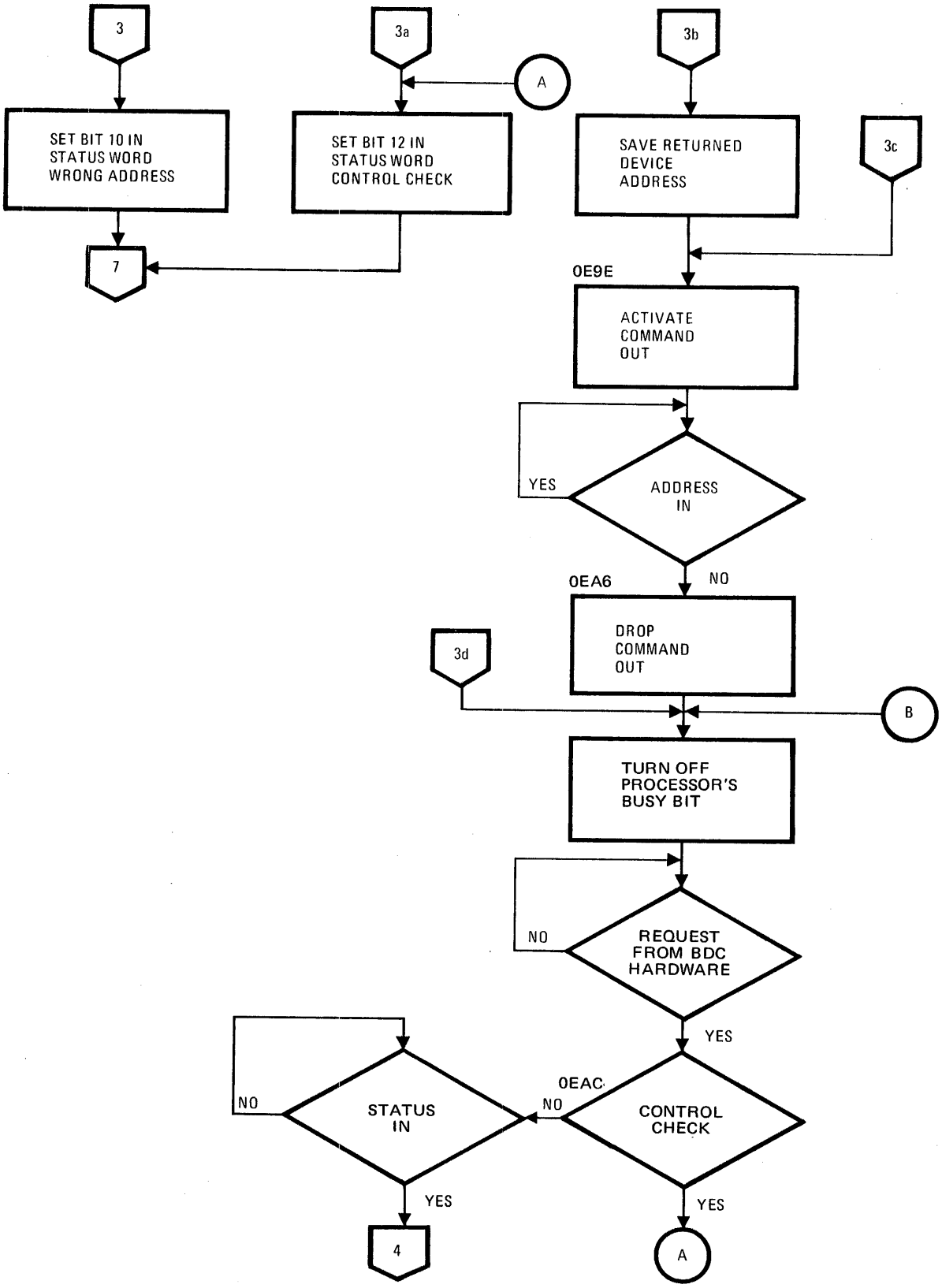


Figure 4-B1. Flowchart, SIO Micro-Instruction Routine (continued)

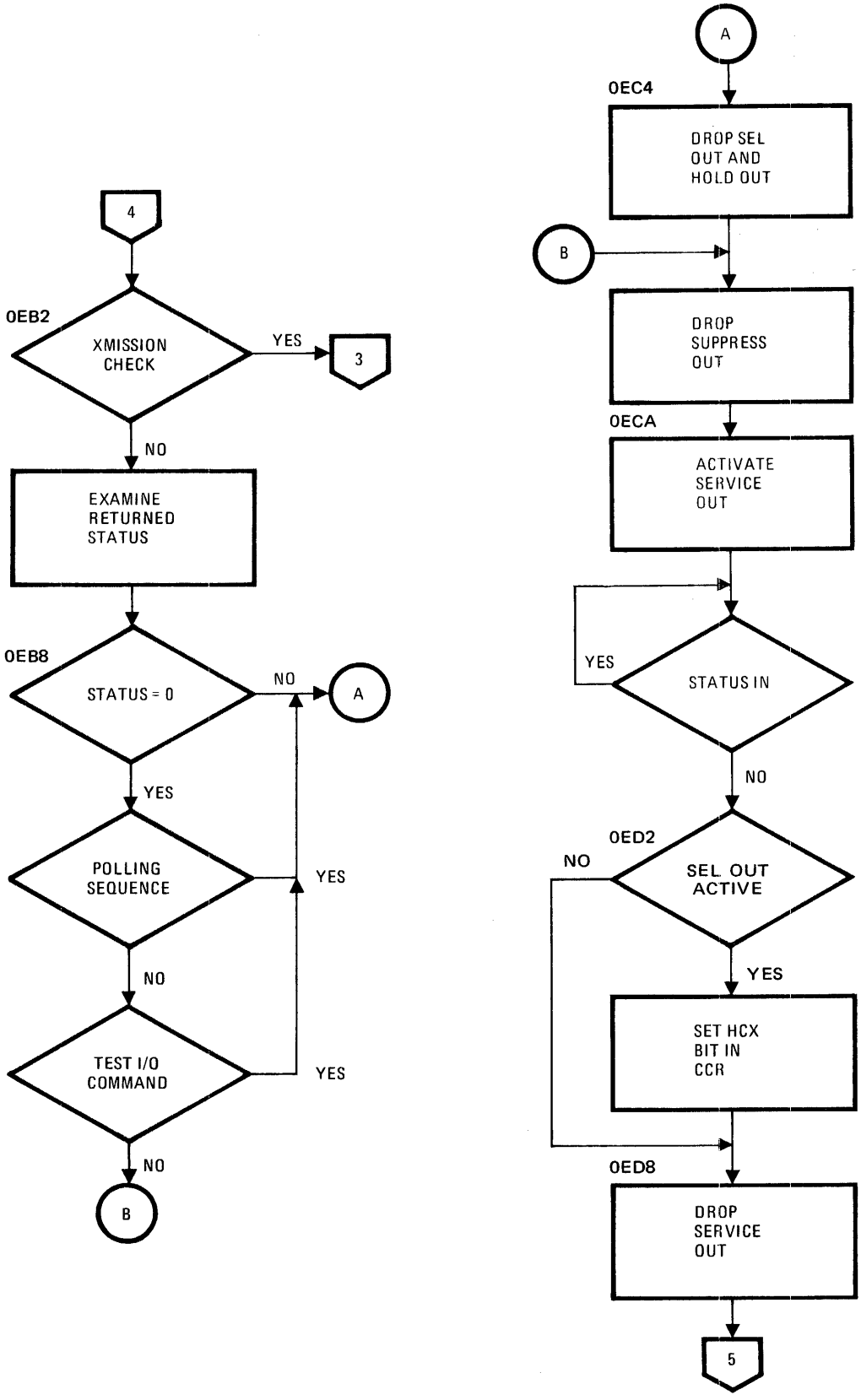


Figure 4-B1. Flowchart, SIO Micro-Instruction Routine (continued)

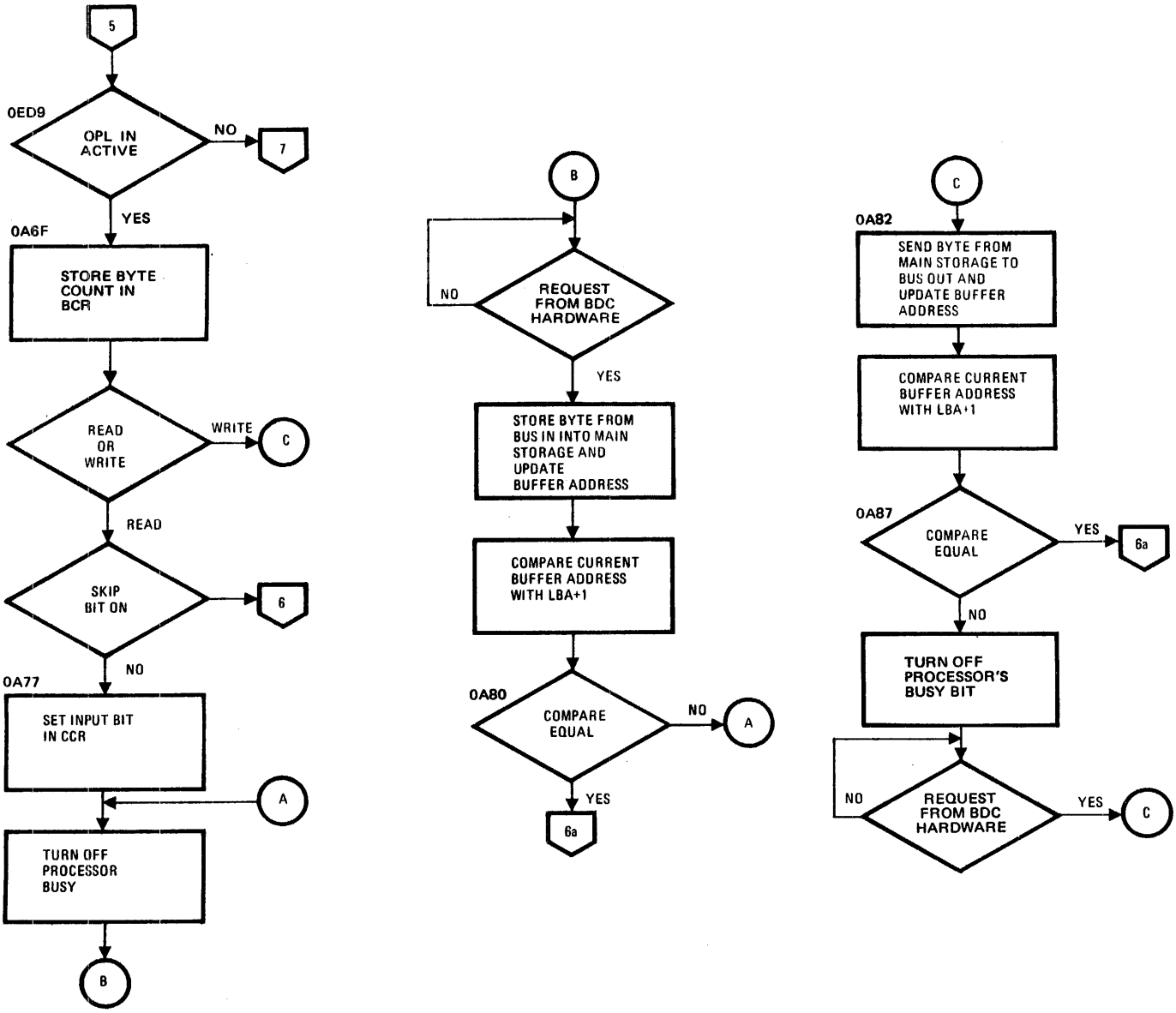


Figure 4-B1. Flowchart, SIO Micro-Instruction Routine (continued)

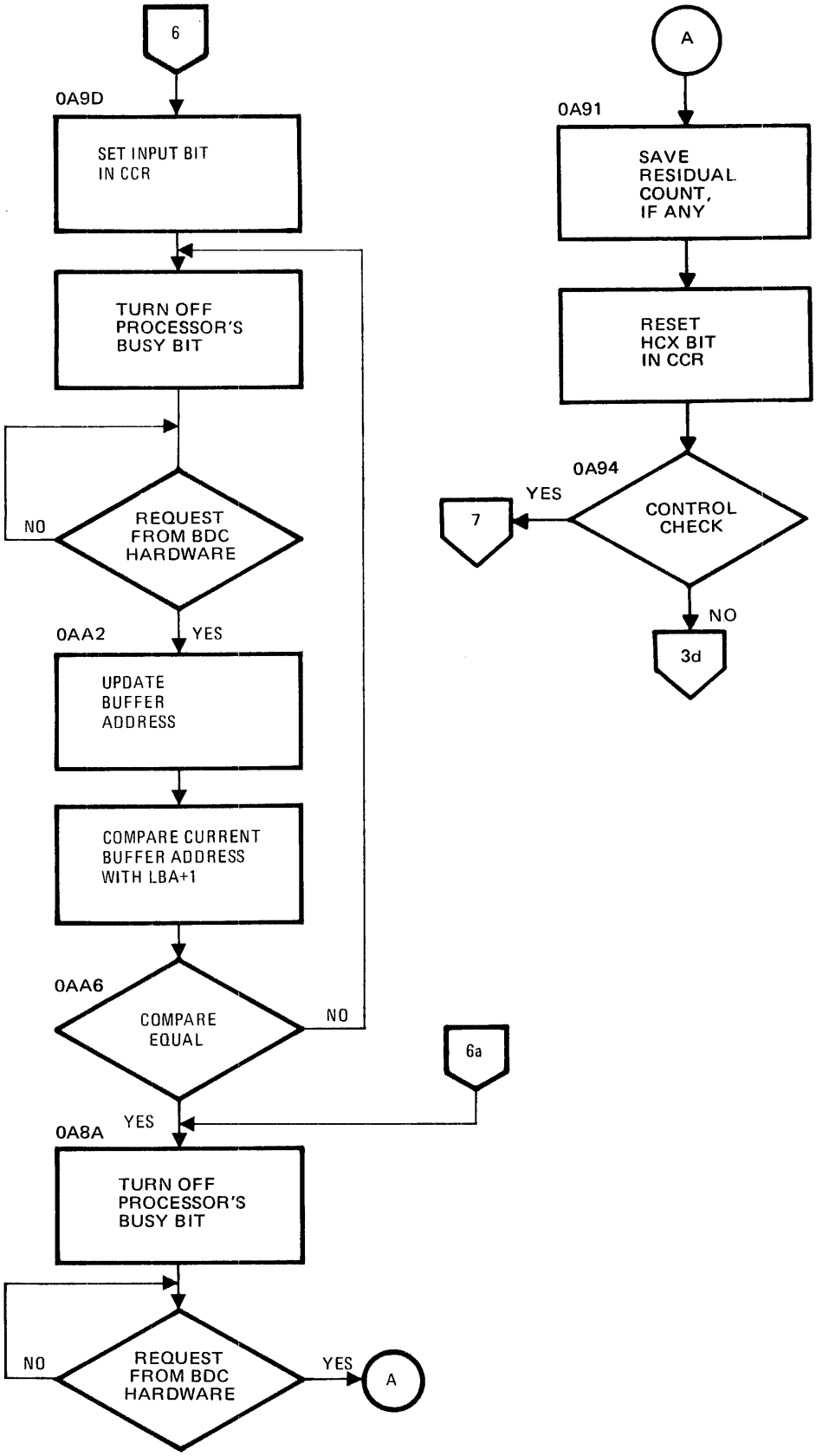


Figure 4-B1. Flowchart, SIO Micro-Instruction Routine (continued)

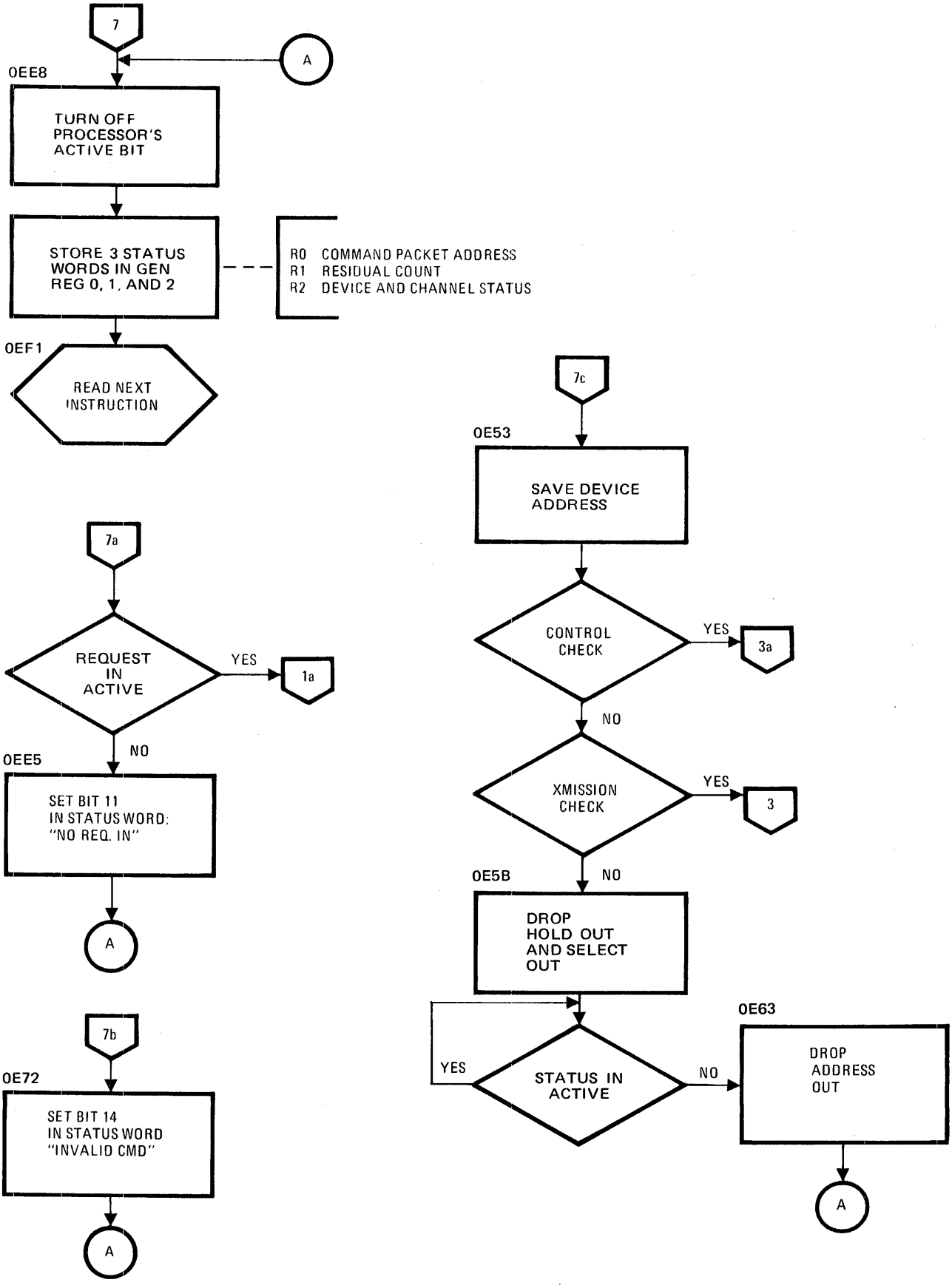


Figure 4-B1. Flowchart, SIO Micro-Instruction Routine (continued)

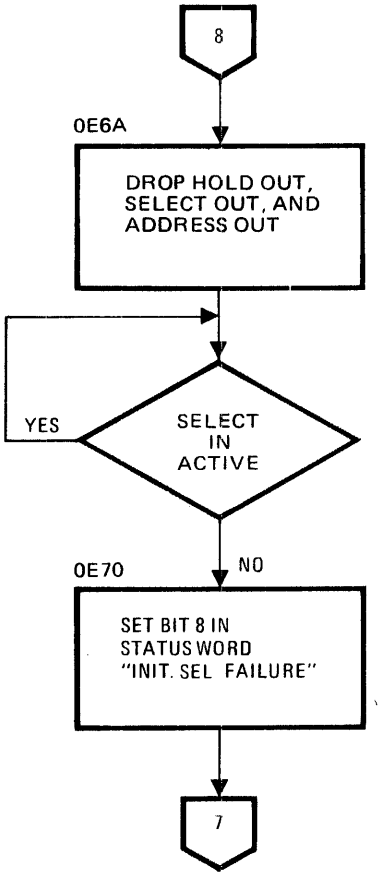


Figure 4-B1. Flowchart, SIO Micro-Instruction Routine (continued)

5. INTEGRATED COMMUNICATIONS ADAPTER (ICA)

(To be supplied)

6. INTEGRATED FILE ADAPTER (IFA)

INTRODUCTION

The Disc Processor functions as a control device through which the Operating System of the 7300 Processing Unit communicates with up to eight MEMOREX 660 Disc Drives.

For maintenance purposes, one or more of the eight drives may be removed from Operating System control, but remain electrically connected to the Processing Unit by removing the numbered (0-7) Unit Select Plug and inserting an S (spare) Unit Select Plug. A ninth disc drive may also be electrically connected to the Processing Unit by inserting an S Unit Select Plug into the unit. The ninth drive may be logically connected to the Disc Processor by interchanging the S Unit Number Plug with a numbered Unit Number Plug from one of the other drives.

The disc drive with the S Unit Number Plug installed is addressable only in maintenance mode.

CAPABILITIES

Logical and control functions are performed by the Integrated File Adapter (IFA) portion of the Disc Processor which decodes instructions sent from shared resources (via processor state 3) and uses this information to initiate the following drive functions:

- Position read/write heads to a specific location on the disc surface (seek) and select a read/write head
- Locate a particular record on the disc surface (search)
- Write a record or part of a record
- Read a record or part of a record
- Transfer selected status information to shared resources

CHARACTERISTICS

In addition to the functional capabilities outlined above, the IFA possesses the following general characteristics:

- 312,000 bytes-per-second data transfer rate
- diagnostic capability
- modular design
- cyclic code error checking
- automatic or manual control storage loading
- optional 100-cylinder access
- IBM 2314 format compatible

DISC ORGANIZATION

To further understand the IFA functions it is necessary to be familiar with the disc organization, as illustrated in Figure 6-1.

A disc pack consists of 11 discs mounted on a common vertical shaft providing 20 recording surfaces numbered 00-19 top to bottom. The read/write heads corresponding to the 20 surfaces on which data may be recorded are mounted on a common movable access mechanism which may be positioned to any one of 203 positions (cylinders) on the disc surface.

Cylinder numbering is from 000 (outermost cylinder) to 202 (innermost cylinder). There is one read/write head for each disc surface. There are 20 read/write heads for each cylinder (00 through 19). Surfaces for cylinders 200 through 202 are reserved as alternates to be used if any surfaces for cylinder 000 through 199 becomes defective.

On systems equipped with 100-cylinder access, cylinder numbering is from 000 through 102 and surfaces for cylinders 100 through 102 are reserved as alternates.

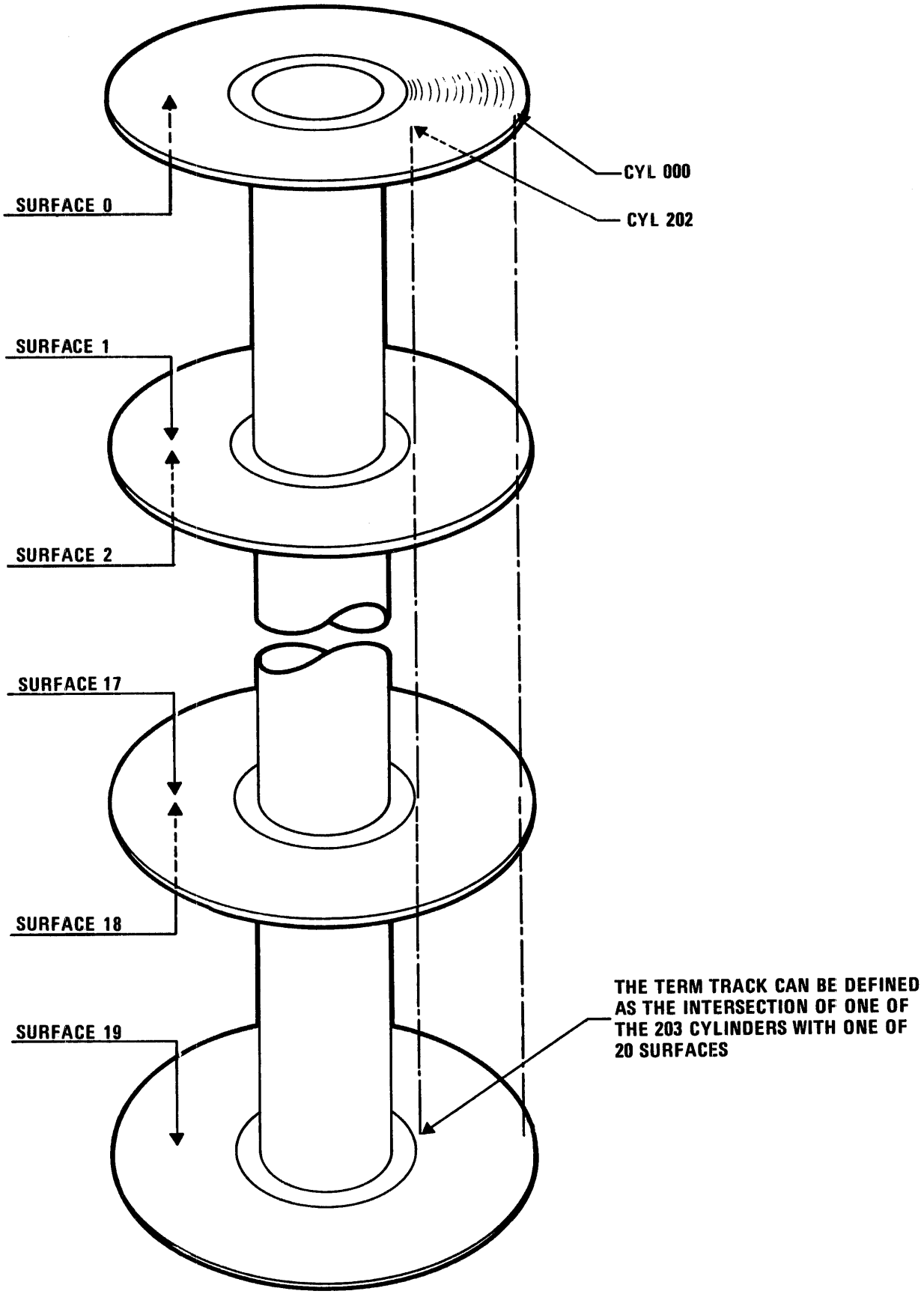


Figure 6-1. Disc Organization

BASIC DRIVE OPERATIONS

The six basic drive operations are briefly discussed in the following paragraphs. All basic drive operations, with the exception of *Seek* and *Restore* are non-concurrent and may be performed only on a selected drive. A detailed description is presented further in this section.

First Seek

The purpose of this operation is to initially position the read/write heads on all of the disc drives to cylinder 000 and set the Head register in all of the drives to zero during power up sequencing. First seek is an automatic operation which occurs as soon as the spindle motor on each of the disc drives reaches 70% of its operating speed and the 60 second or 30 second (optional) timer times out.

Seek Other than First Seek

This operation positions the read/write heads of a selected disc drive to a program specified cylinder address. At the completion of this operation the disc drive sends an ATTENTION signal to the IFA indicating it is ready for further instruction.

Head Selection

This operation is performed in conjunction with a *Seek* operation to select a head (1 of 20) at the addressed cylinder for a read or write operation.

Write

This operation performs the following functions in the IFA:

- Generates all write format requirements.
- Receives word-parallel data from shared resources during the processor state 3 time slice.
- Converts the data to bit-serial mode and writes it on the selected disc drive.
- Generates cyclic check burst format and writes it on the selected disc drive.
- Presents ending status information to processor state 3.

Read

This operation performs the following functions in the IFA:

- Reads data from the selected disc drive in bit-serial mode.
- Converts the data to word-parallel form and sends it to shared resources during the processor state 3 time slice.
- Generates a cyclic check burst and compares it against the one read from the disc drive.
- Presents ending status information to processor state 3.

Reset/Load

This operation loads control storage (CS) with data read from drive 0. This data consists of the micro instructions used to implement the machine language instructions.

At completion of a CS load performed in Operator Mode or Program Mode, control is given to the micro-instruction routine that loads the operating system software into main storage.

PHYSICAL DESCRIPTION

The following paragraphs contain a physical description of the IFA portion of the Disc Processor including a description of the Logic and AC power cabling between the processing unit and each of the drives.

HARDWARE PLACEMENT

The IFA consists of nine printed circuit boards mounted in chassis 1, row B of the 7300 Processing Unit (Figure 6-2).

A detailed drawing which shows the PC board slot locations and the letter designations assigned to each board is provided in Figure 6-3.

The AC line power necessary to energize the DC power supplies and spindle motors in each of the drives is provided by the power sequencer control located inside the center panel beneath the 7300 Processing Unit operator's work table (Figure 6-4).

Figure 6-2. Integrated File Adapter Hardware Placement

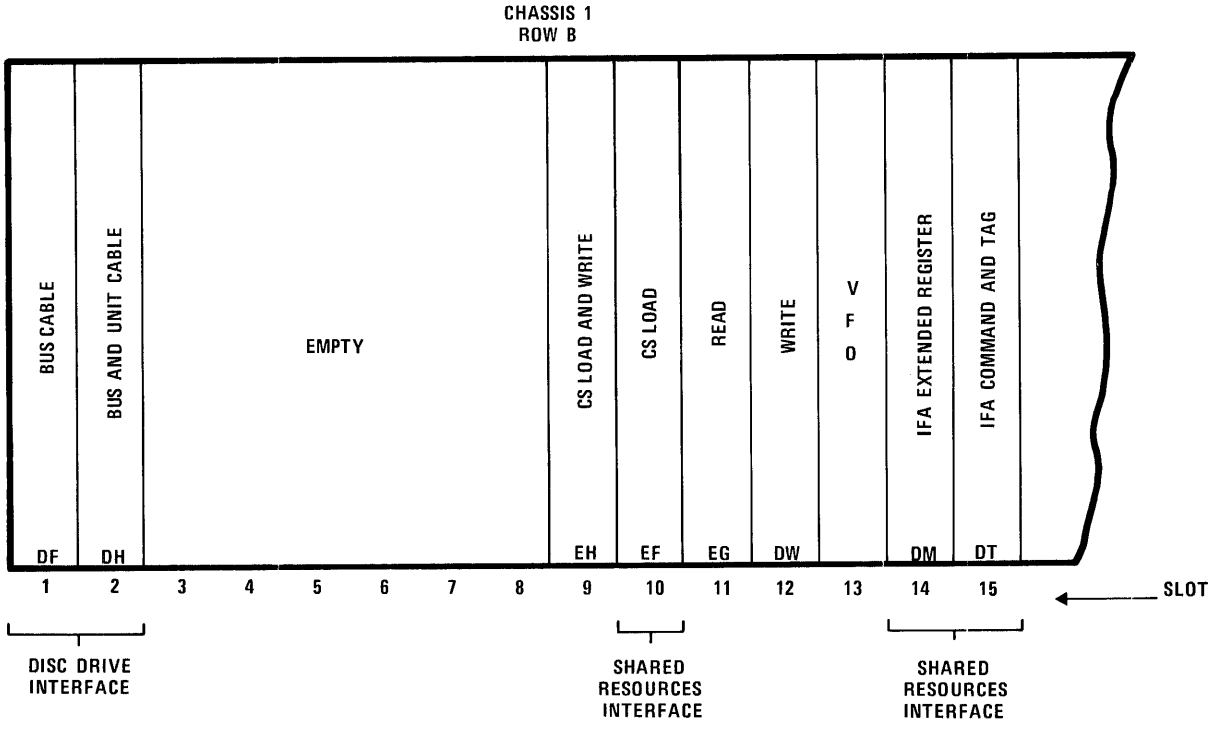


Figure 6-3. IFA Card Type and Locations

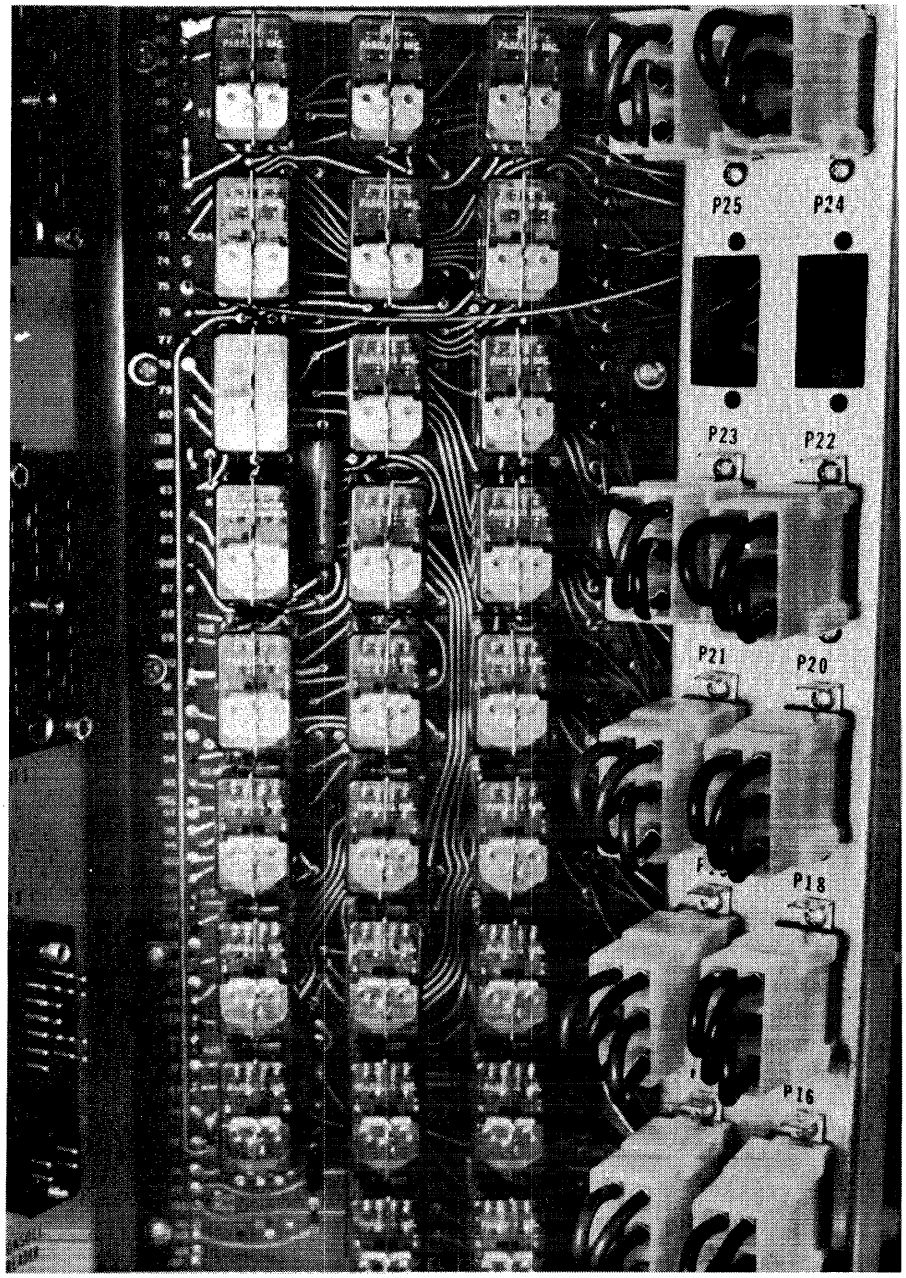


Figure 6-4. Power Sequencer Control

DRIVE LOGIC CABLING

Logic cabling between the IFA and the drives is separated into two classes called Bus (multiplex) cabling and Unit (simplex) cabling. The Bus cable is connected in a *daisy-chain* pattern (serial fashion) to all of the drives and contains the signal lines common to all drives in the string. The Unit cable is connected one cable per disc drive and contains all signal and data lines unique to a particular drive.

Figure 6-5 shows the overall cabling connections required for a maximum system configuration.

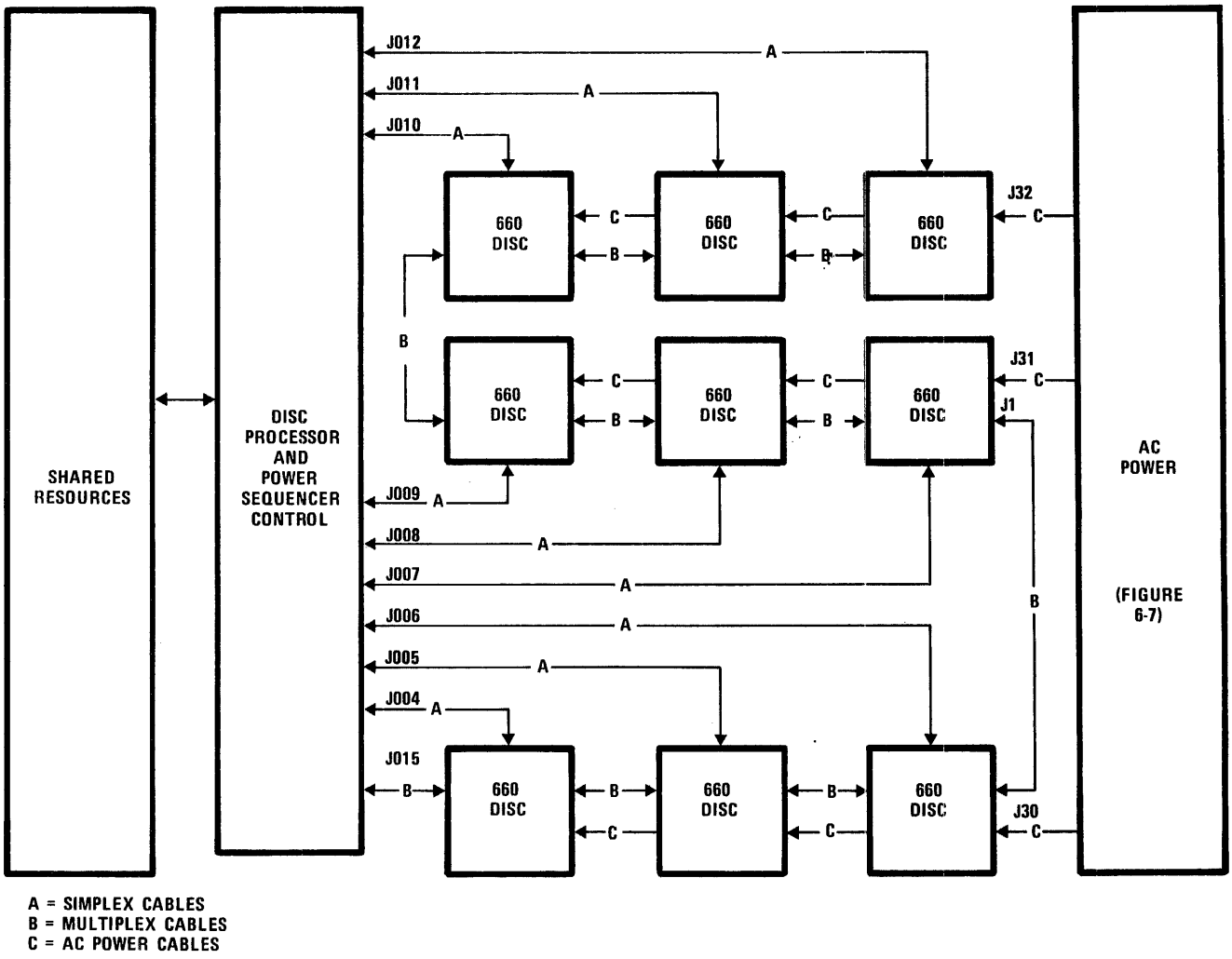


Figure 6-5. Cable Connections

I/O Connector Panel

The I/O connector panel shown in Figure 6-6 is located to the left of the Power Sequence Control. Connector assignments are as follows:

Connector	Cable	Physical Drive Unit
P4	J004	A (1)
P5	J005	B (2)
P6	J006	C (3)
P7	J007	D (4)
P8	J008	E (5)
P9	J009	F (6)
P10	J010	G (7)
P11	J011	H (8)
P12	J012	J (9)

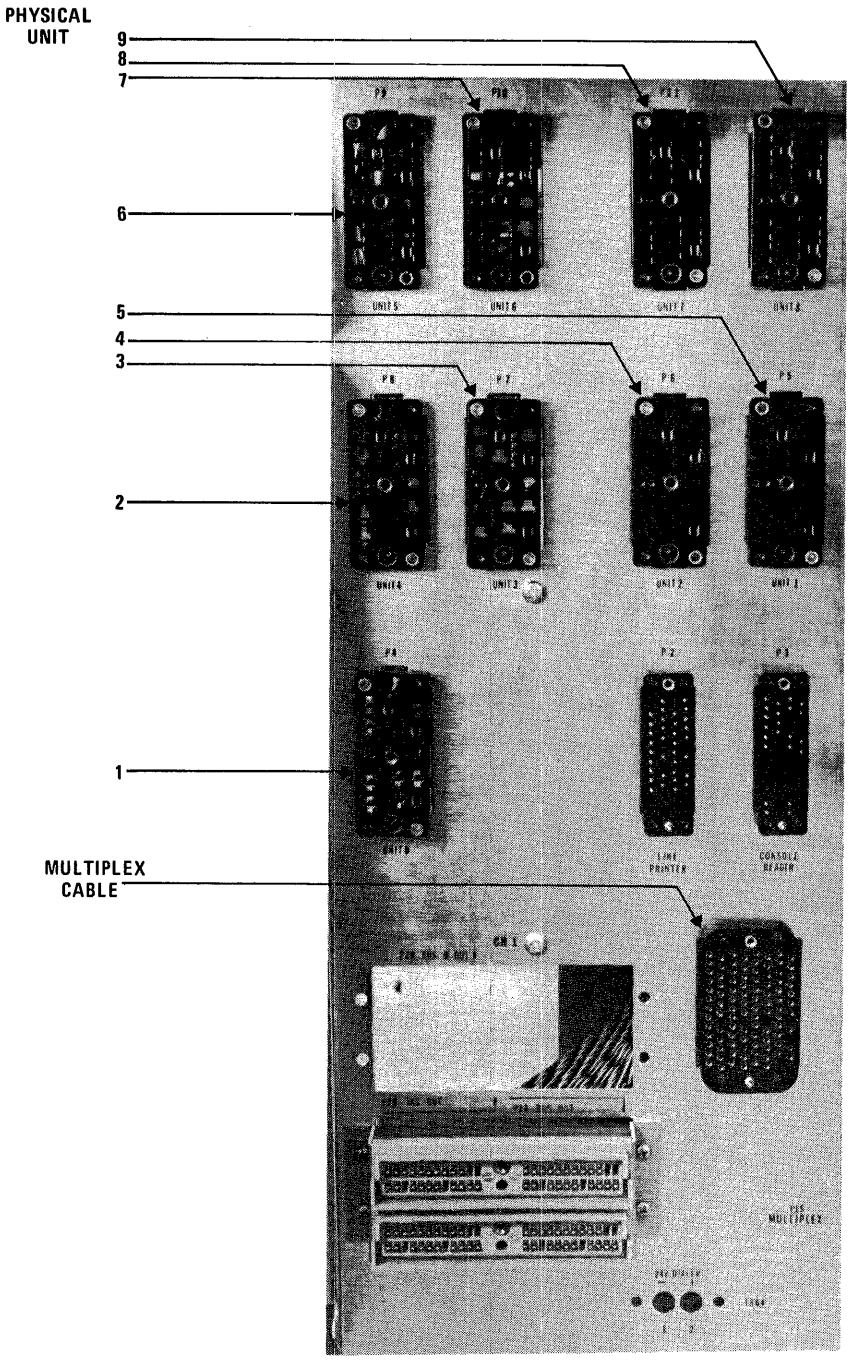


Figure 6-6. I/O Connector Panel

POWER CABLING

Three phase AC power is distributed to each of the drives and certain other peripheral devices in the system from three receptacles (J30, J31 and J32) located inside the operator's table. The cables feeding J30, J31 and J32 are connected to TB3 on the power distribution panel (Figure 6-7). A maximum of three devices may be cabled in a serial fashion from a single three-phase receptacle. Power is fed to each receptacle in such a manner that the first device in each string is energized from a different phase. In addition, phases are rotated between the AC IN and AC OUT connectors of each device.

To prevent introducing noise into the system, disc drives must not share power cables with other peripheral devices, that is, a power cable providing power to disc drives will only provide power to disc drives.

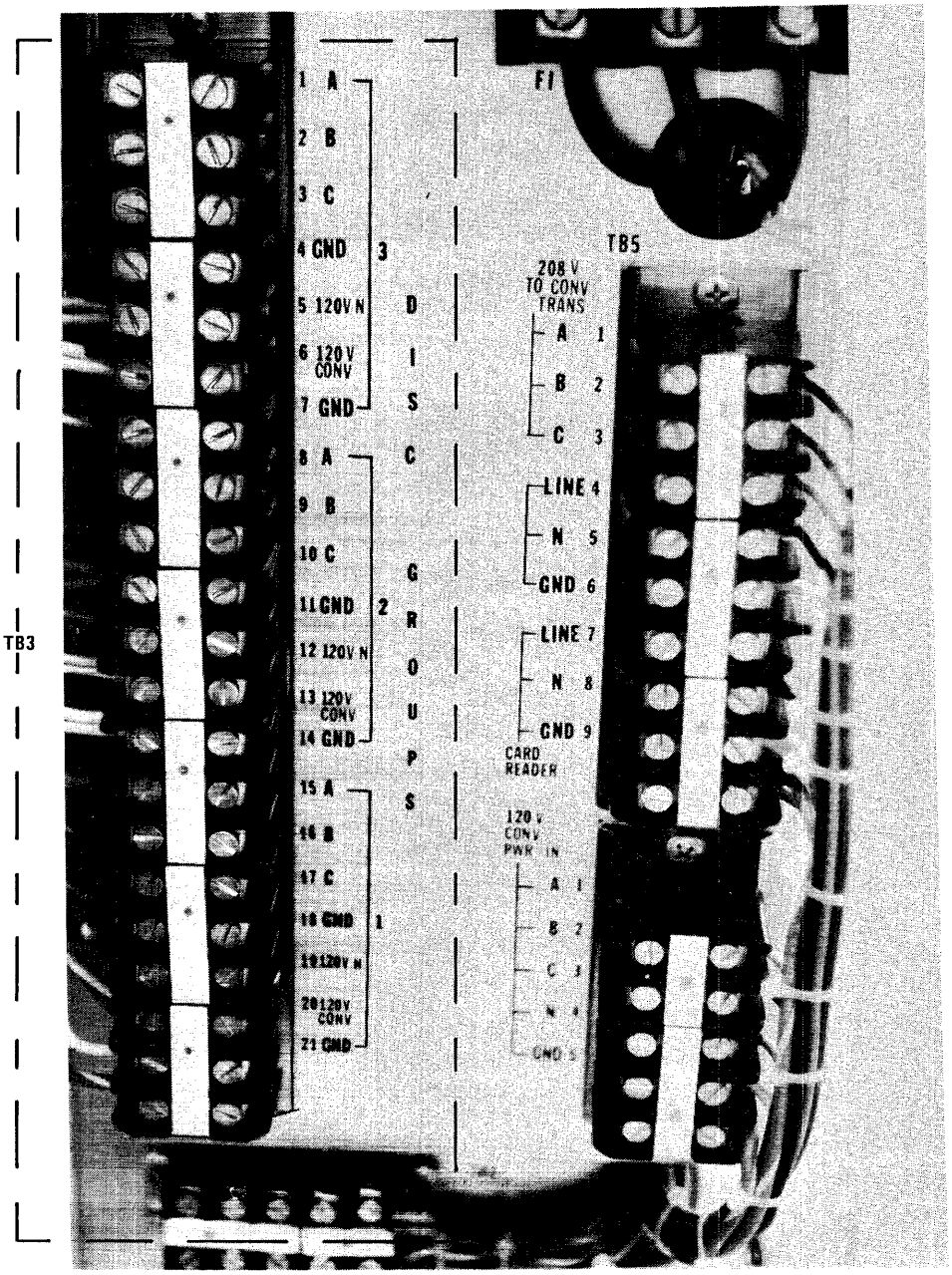


Figure 6-7. Power Distribution

Power Interface

Power sequencer control must provide power to the first device in each string as shown in Table 6-1. Figure 6-8 shows the phase rotation for the AC power to the disc drives.

Table 6-1. Power Connections

7300 Processing Unit		Disc Drive		
Receptacle	Pin	Receptacle	Pin	Description
J30	A	J1	A	Phase A
	B		B	Phase B
	C		C	Phase C
	D		D	System Ground
	E		E	120 VAC
	F		F	120 VAC Neutral
	G		G	Shield Ground
J31	A	J1	A	Phase C
	B		B	Phase A
	C		C	Phase B
	D		D	System Ground
	E		E	120 VAC
	F		F	120 VAC Neutral
	G		G	Shield Ground
J32	A	J1	A	Phase B
	B		B	Phase C
	C		C	Phase A
	D		D	System Ground
	E		E	120 VAC
	F		F	120 VAC Neutral
	G		G	Shield Ground

CABLE LENGTHS

Drive interface cables should conform to the following specifications:

Unit Cable – Separate run for each drive (9 maximum) – 50 ft. maximum per cable

Bus Cable – Jumper connection from drive to drive – 100 ft. maximum accumulated

AC Cable – Jumper connection from drive to drive (three drives per cable maximum, three cables maximum) – 100 ft. maximum accumulated each cable

ELECTRICAL SPECIFICATIONS

Power Requirements

Voltage	208 or 230 VAC ± 10%
	4 wire (3 phase, 1 ground)
Frequency	60 Hz
Phases	3 required, 1 used

660 Maximum Start Current – 25 amperes

660 Maximum Run Current – 4.7 amperes at 208 VAC

660 Nominal Current – 3.6 amperes at 208 VAC

Logic Levels

IFA Internal	Logical 1 = +2.0 to +5.5 VDC
	Logical 0 = -0.7 to +0.8 VDC
Drive Interface	Logical 1 = -1.5 VDC
	Logical 0 = +1.5 VDC

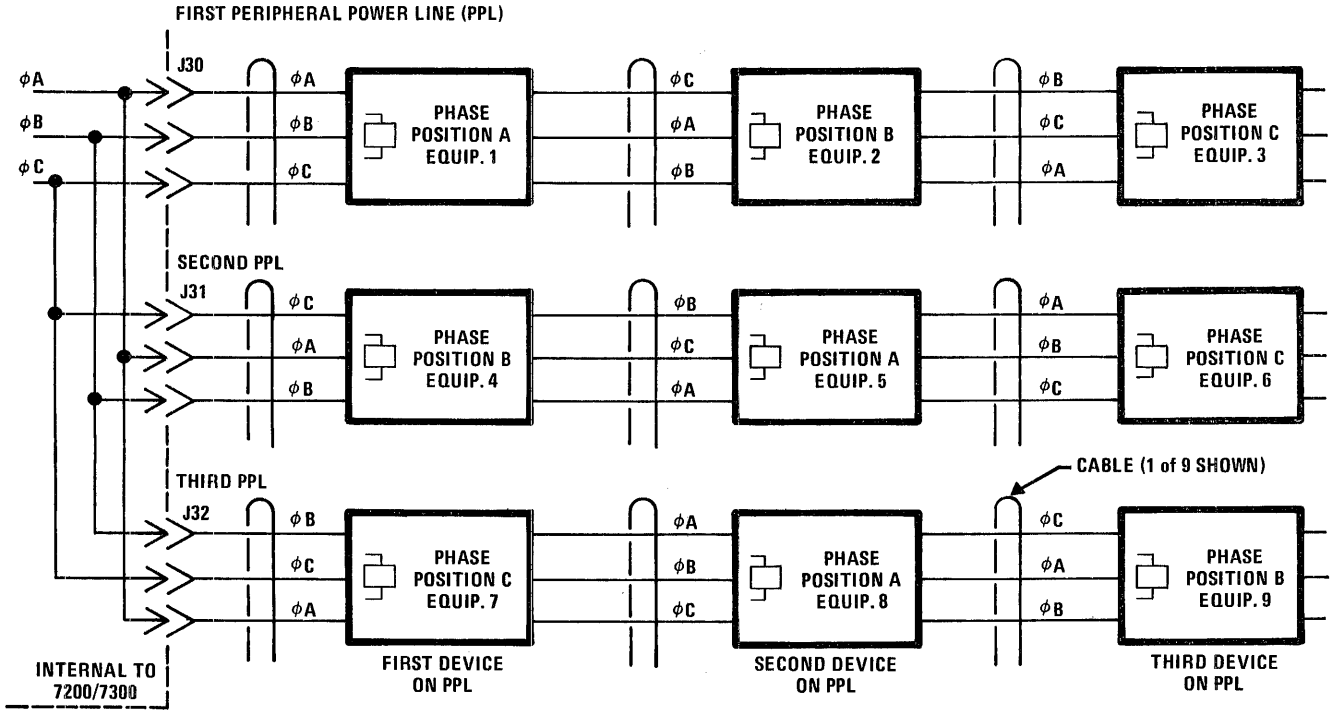


Figure 6-8. Phase Rotation

OPERATIONAL DESCRIPTION

The following paragraphs contain a description of the disc processor command structure and track format including a description of command timing and IFA/Disc status responses.

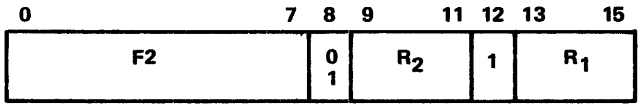
COMMAND STRUCTURE

There are two separate areas involved in the command structure, the machine language instructions with their associated command packets and the IFA commands issued by micro instructions.

Machine Language Instructions

Disc Input/Output (DIO)

This instruction can be executed only by processor 3 and is used to perform read or write operations. The DIO instruction initiates a micro-instruction routine (Appendix A) which operates on a variable length command packet or several such packets located at an address specified by R_2.



Bits 0-7 = function code

Bit 8 = indirect addressing indicator
- 0 = direct
- 1 = indirect — R_2 field specifies a register containing a memory location containing the address of the first word in the command packet.

Bits 9-11 = register address containing memory location of the first word in the command packet (direct).

Bit 12 = always 1

Bits 13-15 = register address containing memory location where ending status information will be stored.

At the end of a DIO operation, the status information of the last operation performed is transferred to memory beginning at the address specified by the R_1 field. An information table is placed in main storage with the following format:

- First word: current command program address
- Second word: hardware status information of last operation
- Third word: last byte address (+1 if byte count is even, +2 if byte count is odd)

The DIO micro-instruction routine operates on the command packets illustrated in Figure 6-9. The following paragraphs describe the entries on the figure format.

BRS (Bit Ring Sync): determine the type of field (home address, count, key or data) to be read or written.

Byte Count: specify the number of bytes in the command argument (search packet); or the total number of bytes to be read or written.

Tag: add base register displacement to the buffer address (64K or larger machines only).

Buffer Address: specify the starting address of the argument (search packet); or the starting address which will be used to store data from the disc; or where data to be recorded on the disc is located. Buffer address must be an even memory location.

Flag: indicate track usage and condition. Refer to Figures 6-12 and 6-13 for an explanation of the use of the Flag byte.

C1, C2: specify the cylinder address. C1 is always set to zero. C2 can have a value between 000 and 202 (hexadecimal 00 and CA).

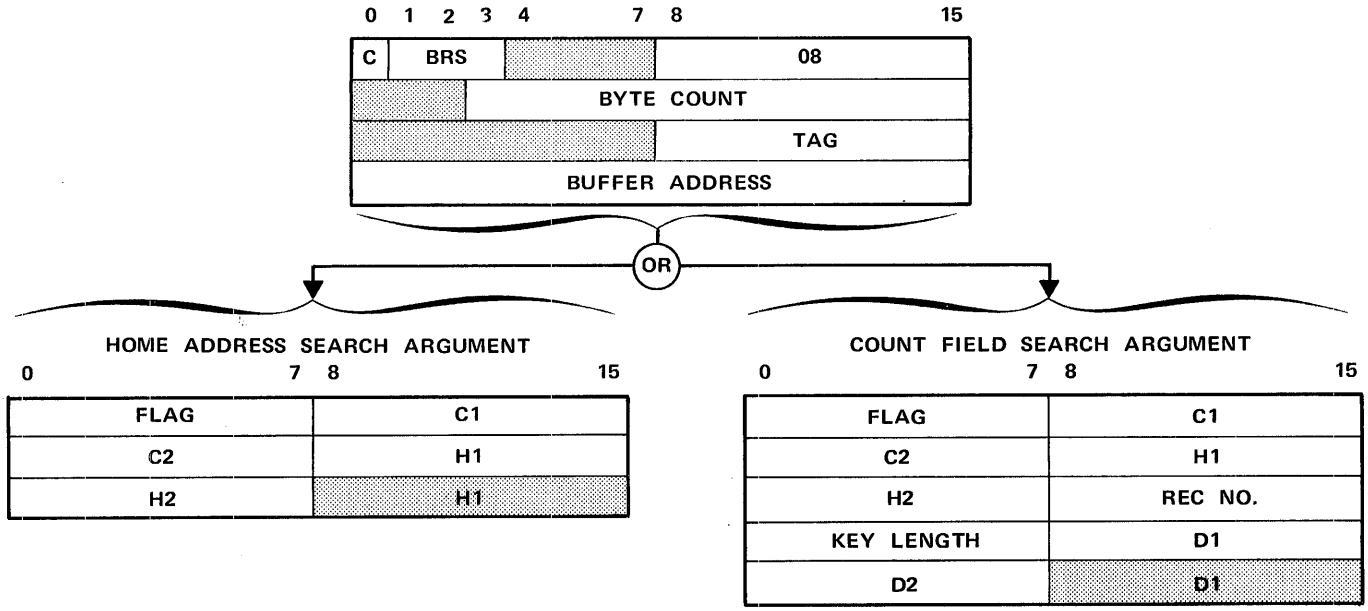
Gap Length: specify the one's complement in hexadecimal of the number of supplementary gap bytes which must precede a write field (refer to "Write Commands").

H1, H2: specify the head address. H1 is always set to zero. H2 can have a value between 00 and 19 (hexadecimal 00 and 13).

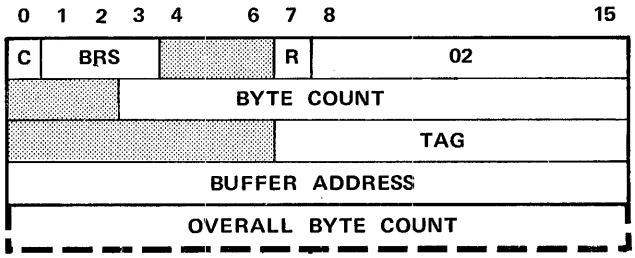
Key Length: specifies the number of bytes in the key field. If no key field is included, key length must be set to zero.

Overall Byte Count: specifies total length of all data fields to be read during repetitive read (multiple of byte count when R is 1). When R is 0 this word is not used and must not be included in the command packet.

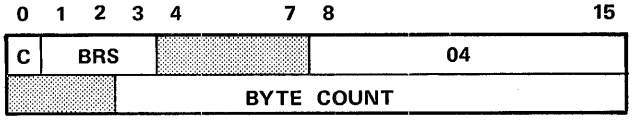
SEARCH (08)



READ (02)



READ WITHOUT TRANSFER (04)



WRITE OR FORMAT WRITE (01/10)

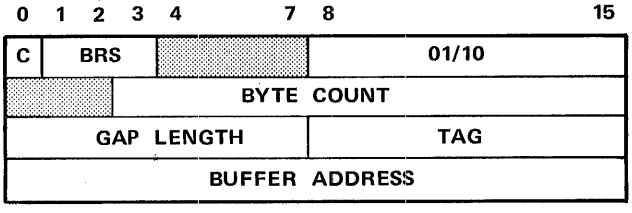


Figure 6-9. DIO Command Packets

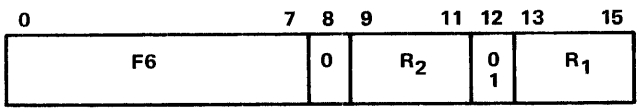
R = Repetitive Read Indicator

0 = No multiple read
1 = Multiple read requested — all records must be the same length

C (chain): if this bit is 1, the DIO will expect to find another command packet immediately following the first command packet. After executing the first command, the microprogram will proceed to execute the second command. Chaining will continue until DIO reaches a command packet with a chain bit of 0 which the microprogram will assume to be the last packet to be executed. Whenever a packet in the chain does not complete successfully the chain is terminated and the status word of the command packet being executed is returned to shared resources.

Output to I/O Register (OUT)

When executed in processor 3, this instruction is used to transfer information from the general register specified by R_1 (if bit 12 is 0) or from the memory location specified by R_1 (if bit 12 is 1) to the extended register in the IFA.



Bits 0-7 = function code

Bit 8 = always 0

Bits 9-11 = extended register to be written (Table 6-2)

Table 6-2. Extended Register Addressing

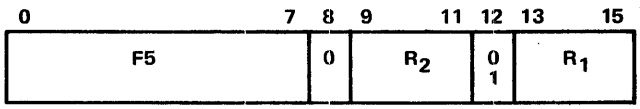
R_2 Field	Register Address	Function Category
0000	Write Register 10	Select Drive
0001	Write Register 11	Control Command
0010	Write Register 12	Request Hardware Status
0011	Write Register 13	Request Attention and Physical Drive Address (Status)
0111	Write Register 17	Request Attention and Cylinder Address (Status)

Bit 12 = indirect addressing indicator
0 = direct
1 = indirect — R_1 field specifies a register containing the address of the word to be written

Bits 13-15 = specifies a general register containing the word to be written (direct)

Input from I/O Register (INP)

When executed in processor 3, this instruction is used to transfer status information from the extended register in IFA to the general register specified by R_1 (if bit 12 is 0) or to the memory location specified by R_1 (if bit 12 is 1). An INP instruction must follow an OUT instruction that requests status (Write Register 12, 13 or 17). A command argument need not be specified.



Bits 0-7 = function code

Bit 8 = always 0

Bits 9-11 = extended register from which information is to be read (normally 0)

Bit 12 = indirect addressing bit
0 = direct
1 = indirect — R_1 field specifies the register containing the address where information is to be placed.

Bits 13-15 = general register to receive the information (direct)

IFA Commands

The IFA commands are generated by micro instructions as a direct result of executing the machine language instructions (MLI's) OUT and DIO.

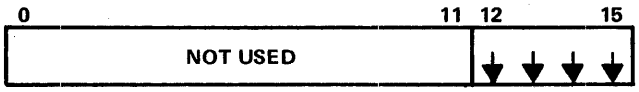
IFA Commands Specified by OUT MLI

The following IFA commands and command groups are issued as a result of the OUT MLI:

- Select Drive Command (Write Extended Register 10)
- CS Load Command (Write Extended Register 11)

- Diagnostic Commands (Write Extended Register 11)
- Control Commands (Write Extended Register 11)
- Write Hardware Status Command (Write Extended Register 12)
- Write Other Status Commands (Write Extended Register 13 or 17)

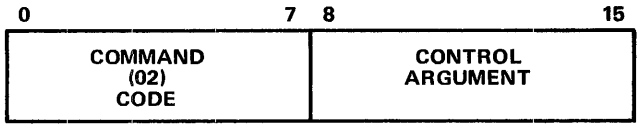
Select Drive Command – This command selects the drive which will respond to the instructions that follow. The *Select Drive* command is transferred to the IFA via an *OUT* machine language instruction specifying Write Register 10. With the exception of a *Write Attention* instruction, all IFA commands require that a drive be selected. A drive remains selected until another drive is selected by a *Select Drive* command. Selecting a drive clears read/write control logic. MASTER CLEAR selects drive 0.



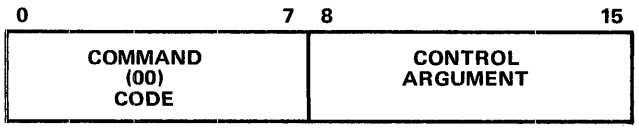
0	0	0	0	Select Drive 0
0	0	0	1	Select Drive 1
0	0	1	0	Select Drive 2
0	0	1	1	Select Drive 3
0	1	0	0	Select Drive 4
0	1	0	1	Select Drive 5
0	1	1	0	Select Drive 6
0	1	1	1	Select Drive 7
1	0	0	0	Select Drive S (Spare Drive)

Diagnostic Commands – These commands are used by maintenance personnel to perform on-line testing on a selected disc drive. Diagnostic commands are transferred to the IFA via an *OUT* machine language instruction specifying Write register 11.

Latch: This command may be used to check the error logic on a selected drive by placing an illegal command argument on the Unit bus lines. An illegal command argument forces a *File Unsafe* condition in the selected drive. A *Latch* command may also be used to clear a *File Unsafe* condition by activating control tag and bus lines 1 and 3 for a minimum of 25 milliseconds.

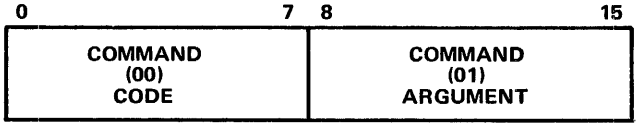


Pulse: This command or any of the control commands (00XX, 08XX or 04XX) may be used to clear a *Latch* command.

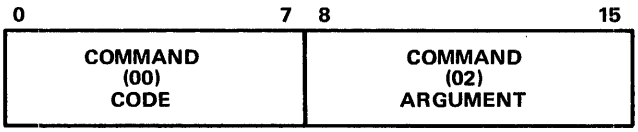


Control Commands – Control commands are used to precondition the selected drive for data transfer operations. Control commands are denoted by a 00XX, 04XX or 08XX format with XX defining the command argument. All control commands are transferred to the IFA via an *OUT* machine language instruction specifying Write Register 11.

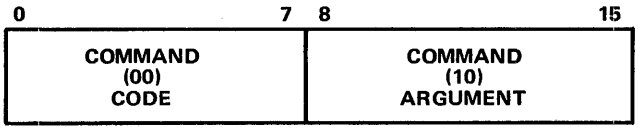
Set Head Advance: This command increments the head register in the selected drive by one.



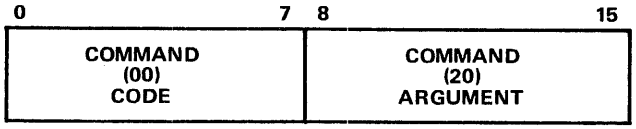
Restore: This command moves the read/write heads to cylinder 000 and clears the head register on the selected drive; this operation is identical to a *First Seek* operation. A *Restore* command is ignored if executed while a *Seek* operation is in progress.



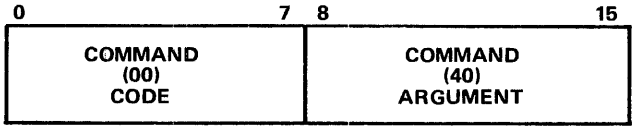
Reset Head Register: This command clears the Head register on the selected drive.



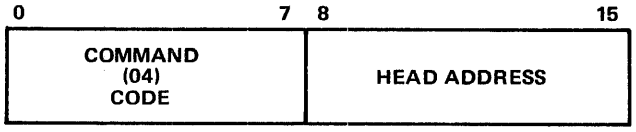
Start Seek: This command initiates forward or reverse movement of the read/write heads to a program specified cylinder address. An ATTENTION signal is generated by the selected drive on completion.



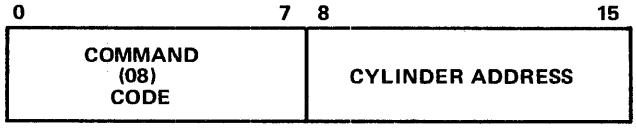
Reset Attention: This command clears the ATTENTION signal on the selected drive.



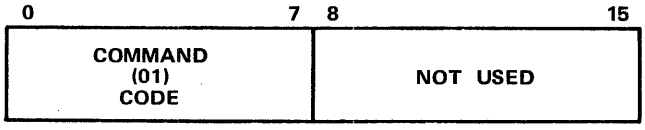
Set Head Register: This command loads the Head register in the selected drive with the address contained in the lower byte of the command. A *Set Head Register* command must be preceded by a *Reset Head Register* command.



Set Cylinder Register: This command loads the Cylinder register in the selected drive with the address contained in the lower byte of the command.



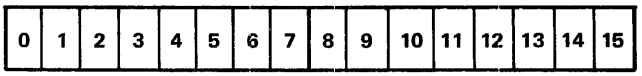
CS Load Command – A *CS Load* command loads control storage from disc. The command may be executed in maintenance mode at any time or within one second following a Power On Reset/Load if in operator mode or program mode. If executed in operator mode or program mode, logical drive 0 is automatically selected. Loading starts from the current cylinder address and head address. If a burst check error occurs in maintenance mode, the operation is terminated. The *CS Load* command is sent to the IFA via an *OUT* instruction specifying Write Extended Register 11.



Write Hardware Status Command – This type of status indication provides information on the condition of the IFA and the selected disc drive preparatory to and during a *Read* or *Write* operation.

Request IFA and Drive Status

This command writes IFA and disc status information into the extended register and clears the IFA status and selected drive status indications. A *Request IFA* and *Drive Status* instruction is transferred to the IFA by an *OUT* machine language instruction or by a micro-instruction routine specifying Write Register 12. The content of the command word is immaterial and must be followed by an *INP* machine language instruction to enable the status response into shared resources.



Bit	Meaning
0	IFA Status Not Valid or Command Early
1	IFA Missed Window or Command Early
2	IFA Window
3	IFA Track Boundary
4	IFA Read/Write Termination
5	IFA Burst Check Error
6	IFA Lost Data
7	IFA No Sync Compare
8	IFA 3rd Rev Sync Find
9	Disc Not On Line or Seek Incomplete and Not File Unsafe
10	Disc File Unsafe or Seek Incomplete and Not File Unsafe
11	Disc Read Only
12	Disc Pack Change
13	Disc End of Cylinder
14	Disc Write Current Sense
15	Disc Busy

Bit	Meaning
0	IFA Status Not Valid or Command Early
1	IFA Missed Window or Command Early
2	IFA Window
3	IFA Track Boundary
4	IFA Read/Write Termination
5	IFA Burst Check Error
6	IFA Lost Data
7	IFA No Sync Compare
8	IFA 3rd Rev Sync Find
9	Disc Not On Line or (Seek Incomplete and Not File Unsafe)
10	Disc File Unsafe or (Seek Incomplete and Not File Unsafe)
11	Disc Read Only
12	Disc Pack Change

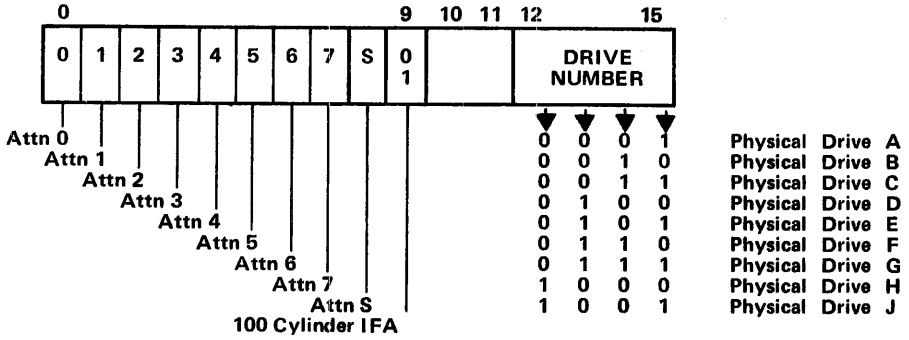
13 Disc End of Cylinder

14 Disc Write Current Sense

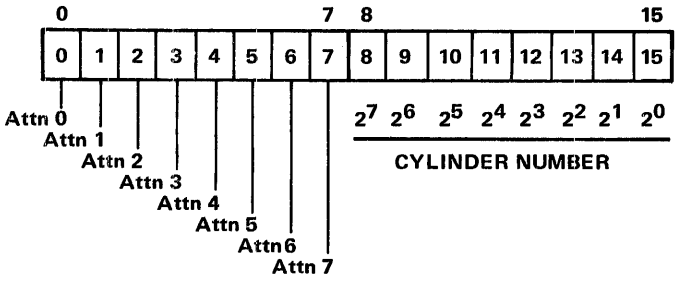
15 Disc Busy

Write Other Status Commands — This type of status indication provides information on the physical configuration of the IFA and selected disc drive and also provides information on the readiness of each of the drives to perform data transfer operations.

Request Attention O-S and Physical Number: Executing an *OUT* MLI specifying Write Register 13 (no command argument required) places the status information, as illustrated below in the extended register and clears the status lines in the IFA. The status information is sent to shared resources by the INP MLI (no command argument required). Prior to attempting another operation on the selected drive, the ATTENTION signal line must be cleared by either a *Reset Attention* command or a *Read* command.



Write Attention and Cylinder Address: Executing an *OUT* MLI specifying Write Register 17 (no command argument required) places the status information, as illustrated below, in the extended register and clears the status lines in the IFA. The status information is sent to shared resources by the *INP* MLI (no command argument required). Prior to attempting another operation on the selected drive, the ATTENTION signal line must be cleared by either a *Reset Attention* command or a *Read* command.



IFA Commands Specified by DIO MLI

The following IFA commands are issued as a result of executing a *DIO* machine language instruction:

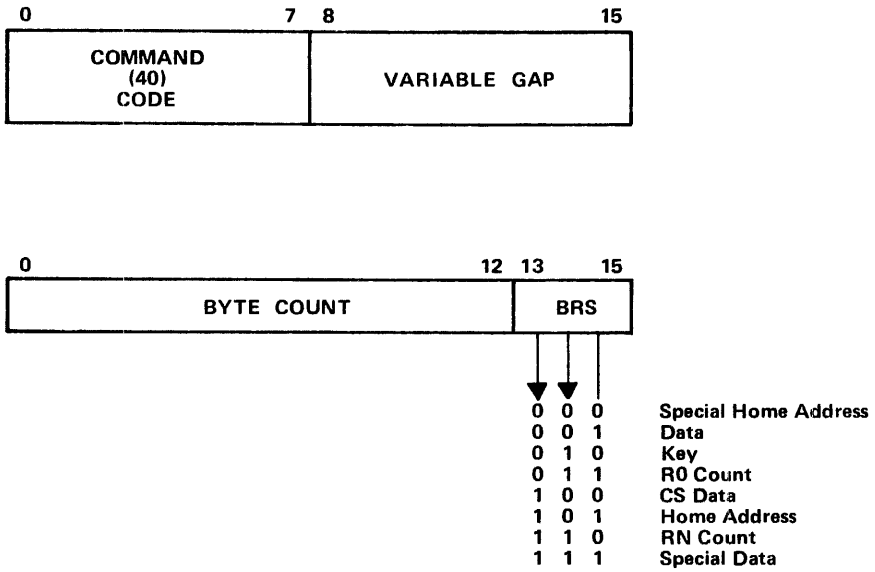
- *Write* Command
- *Format Write* Command
- *Read* Command
- *Read Without Transfer* Command
- *Search* Command-Packet

These commands are all two words in length. The command words are placed in the IFA extended register by the *DIO* micro program. The micro program accomplishes this with a Write Extended Register 11 for the first command word and a Write Extended Register 15 for the second command word.

Write Command — This two-word command transfers data from main storage (shared resources) to a selected field within a record on the disc. A *Write* command must be preceded by an untruncated *Read, Write* or *Search* command except when writing the home address. Additionally, a *Write* command must be immediately followed by a *Request Status* command. A minimum of 1.2 microseconds must be programmed between Word 1 and Word 2 of this command.

If the previous operation was an untruncated *Read* or *Search* operation, shared resources must transmit the first word of this command within a time interval starting at 1.6 microseconds following the previous burst check and ending at 38.4 microseconds after the previous burst check. If the previous operation was an untruncated *Write*

operation, shared resources must transmit the first word of this command within a time interval starting at 22 microseconds following the previous burst check and ending at 38.4 microseconds after the previous burst check. At the conclusion of the operation hardware status information is checked as previously explained.



BRS: Specifies the type of field to be written

Byte Count: Specifies the length of the field to be written

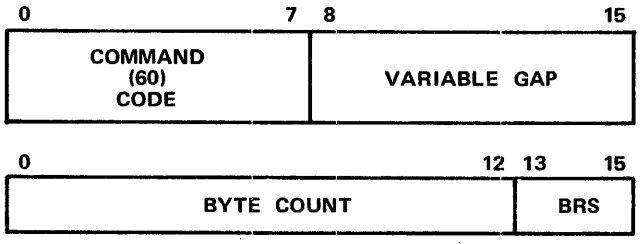
Variable Gap: Specifies the one's complement (hexadecimal) of the number of supplementary gap bytes which must precede the field to be written. IFA write logic automatically generates 41 gap bytes before writing a field of information. Table 6-3 specifies the number of additional gap bytes which must be appended by the command to each type of field.

Table 6-3. Variable Gap Byte Entries

Type of Field	Entry
R0 Count, R0 Data, RN Data, RN Key	FF_{16} (0 bytes)
Home Address*	DF_{16} (32 bytes)
Special Home Address*, Special Data	Not Defined
RN Count	0.043 x (Key Length + Data Length) not including BURST Check bytes

*In the event of home address error, a gap length entry of 164 bytes (205 minus 41 bytes) will be inserted by the utility program.

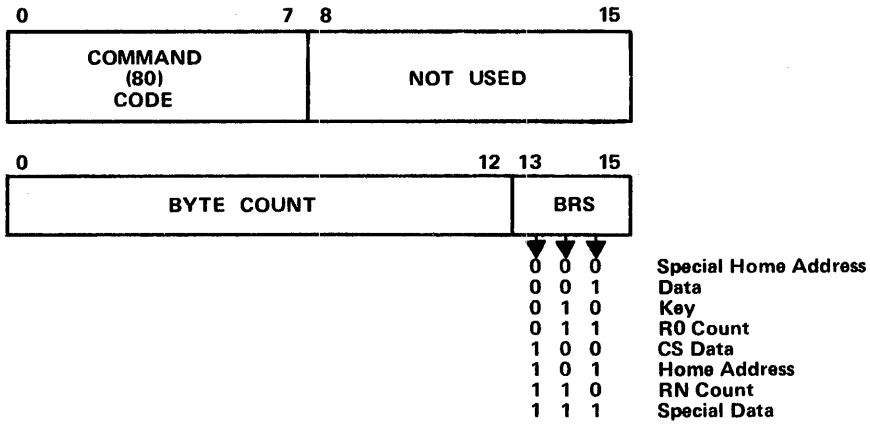
Format Write Command – This two-word command performs the same operation as a *Write* command but in addition writes an all 1's pattern on the remainder of the track following burst check. In all other respects the command is identical to a *Write* command.



Read Command – This two-word command transfers data to storage from a field within a record on the disc. The command must be preceded by an untruncated *Read, Write* or *Search* command except when reading home address, special home address, R0 count or Rn count. The command must be immediately followed by a *Request Status* command. A minimum of 1.2 microseconds must be programmed between Word 1 and Word 2 of this command.

If the previous operation was an untruncated *read* or *search* operation, shared resources must transmit the first

word of this command within a time interval starting at 1.6 microseconds following the previous burst check and ending at 38.4 microseconds after the previous burst check. If the previous operation was an untruncated *Write* operation, shared resources must transmit the first word of this command within a time interval starting at 22 microseconds following the previous burst check and ending at 38.4 microseconds after the previous burst check. At the conclusion of the operation hardware status information is checked as previously explained.

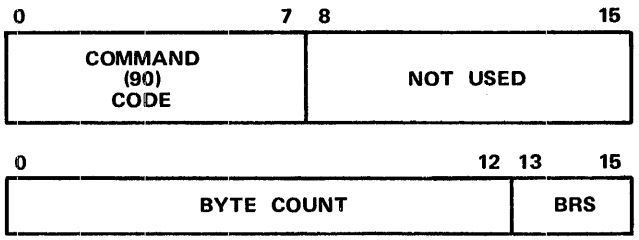


BRS: specifies the type of data to be read

Byte Count: specifies the length of the field to be read

Read Without Transfer Command – This two-word command performs the same operation as a *Read* command with respect to the IFA; however, data is not transmitted from the IFA to main storage. On completion or termination, IFA sends a REQ-3 Signal to shared resources forcing processor 3 into a busy mode (request time slice). *Lost Data* status will not occur with this command.

Read Without Transfer is used to skip over fields within a record and to check cyclic burst. In this instruction BRS specifies the type of field to be skipped and Byte Count specifies the length of the field to be skipped over.



Search Command-Packet – When a *Search* command packet is processed by a DIO machine language instruction, the micro instruction issues an IFA *Read* command. The IFA operates in the normal read mode transferring data to shared resources where the micro code compares it with a *Search* argument, taken from main storage. The *Search* continues until one of the following occur:

1) The desired field is found or

2) The time allotted for the *Search* by the real time clock has expired or

3) A status error condition has occurred.

COMMAND TIMING

All of the basic drive functions except *First Seek* are performed by enabling one of the tag lines and one or more of the Unit bus lines. A *First Seek* operation is performed as part of the power-on sequence and is not controlled by IFA Logic. A programmed *Seek*, by contrast, involves several drive commands executed in sequence. The timing and sequence requirements associated with each drive command are shown in Figure 6-10.

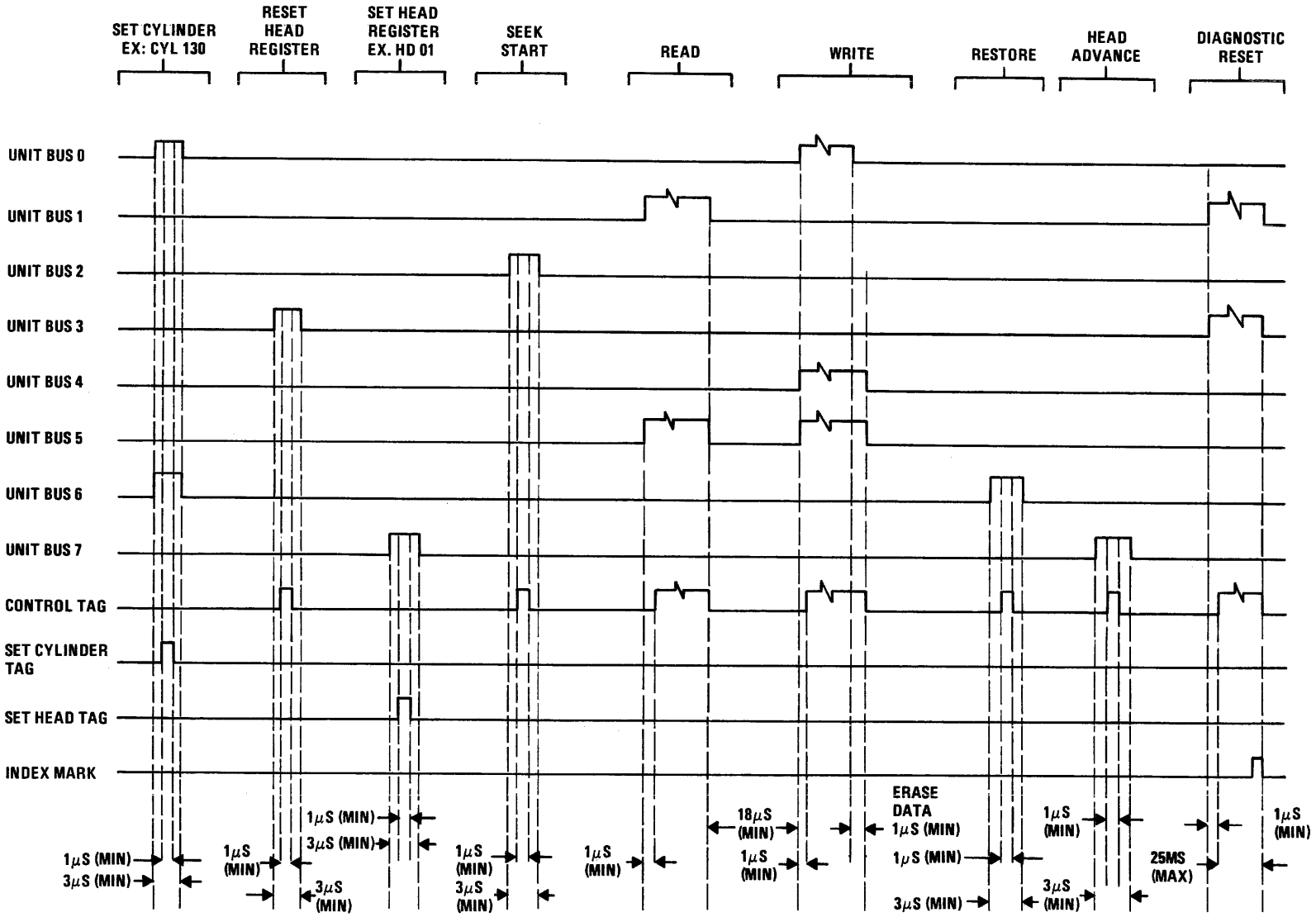


Figure 6-10. Disc Operation Timing

Seek to a Specified Cylinder and Head

A *Seek* operation causes the selected drive read/write heads to move to a new cylinder address (000-202) and also enables one of the read/write heads (00-19). To do this the IFA must perform the following sequence of function commands:

1. Set Cylinder Register –

   (a) Enable Unit Bus Lines 0-7 in accordance with binary equivalent of the desired address:

   Bus Line 0 = Cylinder 128
   Bus Line 1 = Cylinder 64
   Bus Line 2 = Cylinder 32
   Bus Line 3 = Cylinder 16
   Bus Line 4 = Cylinder 8
   Bus Line 5 = Cylinder 4
   Bus Line 6 = Cylinder 2
   Bus Line 7 = Cylinder 1

   (b) Enable SET CYLINDER TAG line

2. Reset Head Register –

   (a) Enable Unit Bus Line 3

   (b) Enable CONTROL TAG line

3. Set Head Register –

   (a) Enable Unit Bus Lines 3-7 in accordance with the binary equivalent of the desired address:

   Bus Line 3 = Head 16
   Bus Line 4 = Head 8
   Bus Line 5 = Head 4
   Bus Line 6 = Head 2
   Bus Line 7 = Head 1

   (b) Enable SET HEAD REGISTER CONTROL TAG line

4. Start Seek –

   (a) Enable Unit Bus Line 2

   (b) Enable CONTROL TAG line

Read

When the carriage arms are positioned at the desired cylinders and the head register in the selected drive has been loaded the IFA may initiate a *Read* operation by executing a *Read* or *Read Without Transfer* command. Executing either of these commands causes the following to occur:

1. Enable Unit Bus Line 1 (read gate)

2. Enable Unit Bus Line 5 (select head)

3. Enable CONTROL TAG line

A *Read* operation may be initiated whenever the selected drive is not busy *(Selected Unit Busy Status* is low) and Bus 0 (write) is not active. Valid read data appears on the READ/WRITE data lines 10 microseconds after the CONTROL TAG signal enables SELECT HEAD and READ BUS lines.

When performing a *Read Without Transfer* operation REQ-3 is disabled until the end of the operation thereby preventing data transfer between IFA and shared resources.

Write

When the carriage has been positioned at the specified cylinder and a read/write head has been selected IFA may initiate a *Write* or *Format Write* command. Executing either of these commands will cause the following to occur:

1. Enable Unit Bus Lines 0 and 4 (write gate and erase gate)

2. Enable Unit Bus Line 5 (select head)

3. Enable CONTROL TAG line

Sending CONTROL TAG and enabling Bus Line 5 must not occur sooner than 18 μ sec after a *Read* or *Write* operation. When concluding a *Write* operation, the signal on Bus Line 4 must be held active for 18 microseconds after Bus Line 0 goes inactive to allow the erase head to side trim erase for a distance equal to 6 bytes after burst check.

A *Format Write* operation is performed in the same manner as a *Write* operation except that the Bus lines and CONTROL TAG signal line are allowed to remain up until *Index Time* thereby allowing write logic to enter all 1's from the end of data to index mark.

The write and erase gates may be selected any time the BUSY STATUS signal to IFA is not active, the drive is not

reading and the READ/WRITE-READ ONLY switch on the selected drive is in the READ/WRITE position. Valid write data appears on the disc surface no later than 10 microseconds after the CONTROL TAG signal line enables the SELECT HEAD and WRITE signal lines.

Head Advance

When a *Read* or *Write* operation uses the tracks of a cylinder in a consecutive sequence, head selection advances one head at a time from the first head selected to the end of the cylinder. To accomplish this, the IFA must perform the following:

1. Enable Unit Bus Line 7
2. Enable CONTROL TAG line

If the head address increments to 20 (one more than the highest head address), the drive sends an *End of Cylinder* status signal to IFA.

Restore

Whenever the selected unit presents a *Seek Incomplete* status signal the head carriage can be moved to a known position (cylinder 000) by initiating a *Restore* (recalibrate) operation. To accomplish this it must perform the following:

1. Enable Unit Bus Line 6
2. Enable CONTROL TAG line

A *Restore* command may be issued whenever the selected drive is not busy (BUSY status line inactive).

Reset Attention

The drive logic which generates the ATTENTION signal to the IFA must be cleared by a *Reset Attention* command or *Read* command from the IFA. To accomplish a *Reset Attention* the IFA must perform the following:

1. Enable CONTROL TAG line
2. Enable Unit Bus Line 1

Reset Attention may be issued at any time except during a *Write* operation.

NOTE

Timing for this operation is the same as that shown for control commands in Figure 6-10.

Diagnostic Reset

The *Diagnostic Reset* command is used to clear the file unsafe logic in the drive when that logic has been set as part of a diagnostic routine. To issue a *Diagnostic Reset* command, the controller must perform the following:

1. Enable Unit Bus Line 1 and Unit Bus Line 3
2. Enable CONTROL TAG line

A *Diagnostic Reset (Latch* command) may be issued at any time except during a *Write* operation.

Load CS

A *Deadstart CS Load* operation may be performed in response to a power on sequence, pressing the RESET/LOAD pushbutton or executing *CS Load* command. If performed in normal mode as the result of a power-on sequence, or pressing the RESET/LOAD pushbutton the IFA will execute the following sequence of commands:

1. *Restore*
   (a) Enable Unit Bus Line 6
   (b) Enable CONTROL TAG line
2. *Set Head Advance*
   (a) Enable Unit Bus Line 7
   (b) Enable CONTROL TAG line
3. *Read*
   (a) Enable Unit Bus Line 1 (read gate)
   (b) Enable Unit Bus Line 5 (select head)
   (c) Enable CONTROL TAG line

If these are performed in maintenance mode as the result of pressing the RESET/LOAD pushbutton the IFA will execute the following function commands:

1 *Set Head Advance*

(a) Enable Unit Bus Line 7

(b) Enable CONTROL TAG line

2 *Read*

(a) Enable Unit Bus Line 1 (read gate)

(b) Enable Unit Bus Line 5 (select head)

(c) Enable CONTROL TAG line

If these are performed in either normal or maintenance mode as the result of a *CS Load* command, the IFA will execute the following function command:

1 *Read*

(a) Enable Unit Bus Line 1 (read gate)

(b) Enable Unit Bus Line 5 (select head)

(c) Enable CONTROL TAG line

A *CS Load* command can be executed whenever the IFA is not performing a *Read* or *Write* operation.

IFA Status Responses

Status Not Valid: occurs when a *Read* or *Write* command is in the Function Register. During a *Read* operation, the Status Not Valid flip-flop clears at bit 4 of the byte following the last burst check. During a *Write* operation, Status Not Valid clears at bit 7 of the byte following the last burst check During a *Format Write,* Status Not Valid clears at index mark. The current operation is not terminated

Command Early Occurs whenever a read/write command is in the Function register and another *Write* register 11 command is detected. The Command Early F/F is cleared by a *Request Status* command or MASTER CLEAR. The current operation is terminated (EOT signal generated) and IFA generates a REQ-3 signal for 30 microseconds every revolution at index mark until a *Request Status* command is received.

Missed Window: Occurs when the first word of a *Read* command (key or data field only) or a *Write* command (count, key, CS data or data field only) is in the Function register and the second word of the command is received during not-window-time or when a CS Load Command is received during not CS window time while operating in operator mode or program mode. The Missed Window F/F is cleared by a *Request Status* command or MASTER CLEAR. The current operation is terminated (EOT signal generated) and IFA generates a REQ-3 signal until a *Request Status* command is received.

IFA Window Exists: The time interval during which shared resources may issue a *Read* command (key or data field only) or a *Write* command (count, key or data field only). Following a *Read* operation, the Window F/F is set at bit 4 of the first byte following the last burst check. Following a *Write* operation, the Window F/F is set at bit 6 of the 7th byte following the last burst check. The Window F/F is cleared in one of the following ways:

- When bit 0 of the 25th byte following burst check is detected during a *Write* or *Read Count Field* operation.
- When AM1 is detected during a *Read Key* or *Data Field* operation.
- When index mark is detected

Track Boundary Error: The program has attempted to read or write across an index mark. The Track Boundary Error F/F is cleared by a *Request Status* instruction or master clear. The current operation is terminated (EOT signal generated) and IFA generates a REQ-3 signal until a *Request Status* command is received.

Read/Write Terminate: The Data Byte Counter (DBC) reached a count of zero either before or after an I/O EXIT signal was received. This status condition forces the DBC to zero and causes the write logic to start writing the burst check immediately. When performing a *Read* or *Search* operation resulting in a hardware terminate, burst check is not computed.

When a *Read* operation is terminated early REQ-3 is generated as follows:

- If the *Read* operation is terminated early by the IFA, REQ-3 is generated immediately.
- If the *Read* operation is terminated early by micro instructions, REQ-3 is generated when the Data Byte Counter reaches a count of zero.

When a *Write* operation is terminated early REQ-3 is generated as follows:

- If the *Write* operation is terminated early by micro instructions, REQ-3 is generated when the Data Byte Counter reaches a count of zero.
- If the *Write* operation is terminated early by the IFA, REQ-3 is generated immediately.

Read/Write Terminate is cleared by a *Request Status* command or MASTER CLEAR. The current operation is terminated (EOT signal generated) and IFA generates a REQ-3 signal until a *Request Status* command is received.

Burst Check: A burst check error has been detected during a *Read* operation. Burst check is set whenever the recompiled cyclic check code or bit count code do not compare with the equivalent information recorded on the disc. The Burst Check F/F is cleared by a *Request Status* command or MASTER CLEAR. The current operation is terminated (EOT signal generated) and IFA generates a REQ-3 signal until a *Request Status* command is received.

Lost Data: Two consecutive data word requests have not been acknowledged by shared resources. The Lost Data F/F is cleared by a *Request Status* command or MASTER CLEAR. The current operation is terminated (EOT signal generated) and IFA generates a REQ-3 signal until a *Request Status* command is received. Lost Data may not occur when executing a *Read Without Transfer* command.

No Sync Compare: This F/F sets during a *Read* or *Search* operation for any of the following conditions:

- Four consecutive index marks detected while looking for a BRS (computed from the time the command is translated).
- *Read* command and not on-line or Seek Incomplete Status.
- AM1 detected twice and no BRS detected while performing a *Read Key* or *Data Field* command.

The No Sync Compare F/F is cleared by a *Request Status* command or MASTER CLEAR. The current operation is terminated (EOT signal generated) and IFA generates a REQ-3 signal until a *Request Status* command is received.

Third Revolution Sync Find: A BRS has been found on the third revolution of a *BRS Search* operation. Three consecutive index marks were sensed and two attempts (search logic disabled during second revolution) were made prior to sensing the correct BRS. Index mark detection is computed from the time the instruction is translated. The Third Revolution Sync Find F/F is cleared by a *Request Status* command or MASTER CLEAR. The current operation is not terminated.

Disc Status

Selected Unit Disc Not On Line: This line becomes high when the selected drive is non-existent, in a powered-down condition, no Unit Select Plug is installed or the ENABLE/DISABLE switch on the selected drive is in the DISABLE position. The current operation is not terminated.

The ENABLE/DISABLE switch will logically change the enable/disable function of a drive only when it is deselected or the Unit Select Plug is removed.

Selected Unit Seek Incomplete: This line becomes high whenever the drive fails to reach a normal detent within 100 milliseconds or a cylinder address greater than 102 has been selected on a 103 cylinder drive (3662 option). If due to a hardware time out, the Seek Incomplete F/F may be cleared by a *Restore* command. If due to a cylinder address selection greater than 102, the Seek Incomplete F/F may be cleared by a *Restore* command or a *Request Status* instruction but a *Restore* command is expected following a Seek Incomplete. The current operation is not terminated.

Selected Unit File Unsafe: This line is high whenever the drive becomes unsafe for operation for any of the following reasons:

- More than one head selected.
- Read bus and write or erase bus selected simultaneously.
- Write or erase selected when the drive is not on-line.
- Write-Current-Sense without write bus selected.
- Write bus selected without write data.
- Erase-Current-Sense without erase bus selected.
- Write bus selected without Erase-Current-Sense.
- Write or erase current supplied to two heads simultaneously.
- A Read or Write is initiated when the head address register is greater than 1011 (19).
- Over-temperature or servo system unsafe.

File Unsafe may be cleared by manual intervention (turn drive off and then back on). A File Unsafe condition caused by illegal bus-line selection may also be cleared by activating the CONTROL TAG line and enabling unit bus lines 1 and 3. A File Unsafe condition deselects the read/write head, clears the Busy status signal, lights the FILE UNSAFE indicator on the selected drive operator control console and terminates the current operation

Selected Unit Read Only This line is high whenever the READ ONLY/READ WRITE switch on the selected drive operator control console is in the READ ONLY position. The current operation is not terminated.

Selected Unit Pack Change: This line is high whenever a unit number plug is changed or the drive is powered up. The PACK CHANGE signal line is cleared by executing a *Read* command or *Reset Attention* command. The current operation is not terminated.

Selected Unit End of Cylinder: This line is high whenever the Head Address register on the selected drive exceeds 19. The current operation is not terminated.

Selected Unit Write Current Sense: This line becomes high within 3 microseconds after the second word of a *Write Count, Key* or *Data Field* command has been received by the IFA or when Index Mark is detected during a home address field write. The WRITE CURRENT SENSE Signal line is cleared 10 byte times after the last burst check when performing a *Write* command. When performing a *Format Write* operation, the WRITE CURRENT SENSE Signal line is cleared at index mark time. The current operation is not terminated.

Selected Unit Busy: This line is high whenever the drive is performing a read/write operation or seeking to a new address. The current operation is not terminated.

STANDARD TRACK FORMAT

Data is written in tracks on the disc surface. Within each track the data is formatted into one or more uniform length records. Every record consists of at least two fields separated by gap information. Several records can be read selectively via micro instruction.

A typical track (Figure 6-11) consists of the following:

- Home address field
- Record zero (R0)
- One or more uniform length records (R1-RN)

The physical starting point for each track is a fixed index mark notched into the bottom cover plate of the disc pack.

Each record contains the following:

- Count field
- Key field (optional)
- Data field

The key field and/or part of the count field facilitate locating data during *Search* operations. Gaps (G0, G2, G4) provide inter-field and inter-record separation for identification purposes and command decoding in a chain sequence.

Every record is self formatting in that it contains information defining its own length. The total amount of data that can be recorded on a track is dependent on the total number of gap and data bytes dependent in turn on the number and length of the fields.

The preceding discussion of specific track areas is valid for the initial writing of the record structure and for those instances when data is rewritten in part. When a field is rewritten, however, the gaps between adjoining fields may vary slightly due to variations in clock oscillator speed and disc rotational speed.

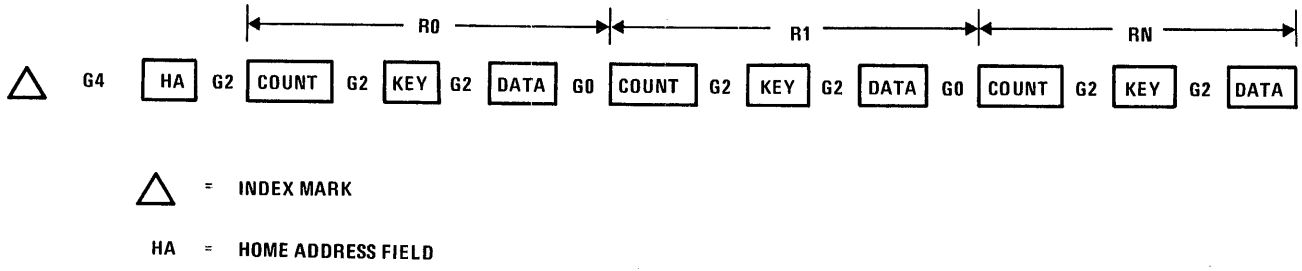


Figure 6-11. Track Format

Home Address Field

A home address (HA) field is written at the beginning of each track during initialization to define track address and condition. It is available through the commands to write or read HA. The information portion of the HA field contains 5 bytes followed by a 4 byte burst check (Figure 6-12). Home address is written after a 73 byte gap from index marker.

Record Zero

Record zero (R0) is normally used to enable an entire track of data to be moved to an alternate track if a portion of the recording surface becomes defective. The use of R0 for this purpose is called *flagging* and R0 is called the *track description record*. When programming does not require flagging, R0 can be used as the first information record on the track.

R0 must contain a count and data field and may also contain a key field the same as subsequent records (Figure 6-13).

- Record length is defined in the count field.
- The key field (when included) usually contains supplementary data information as an additional means of record identification.
- The data field contains the number of data bytes specified by the count field (variable dependent on the space available and restricted to uniform length within a single track).

Data Records

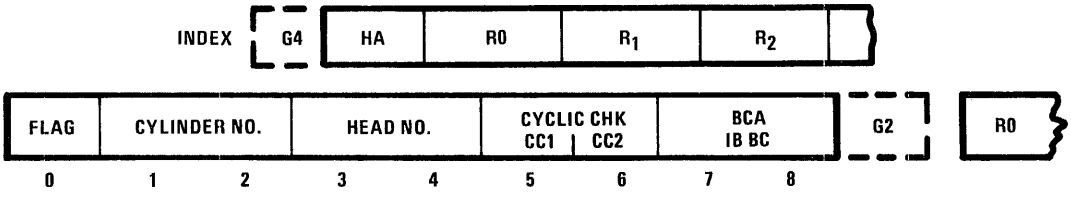
Records R1 through RN contain the normal program provided data information on the tracks. This format with the exception of the bit-ring-sync (BRS) byte preceding each field and the count field flag byte is identical to that for R0.

Gaps

Gaps are used to separate records and fields within records on the tracks. All gap areas contain two address mark (AM) bytes and one program provided bit-ring-sync (BRS) byte which together serve to locate and identify the type of field during *Read* operations (Figure 6-14).

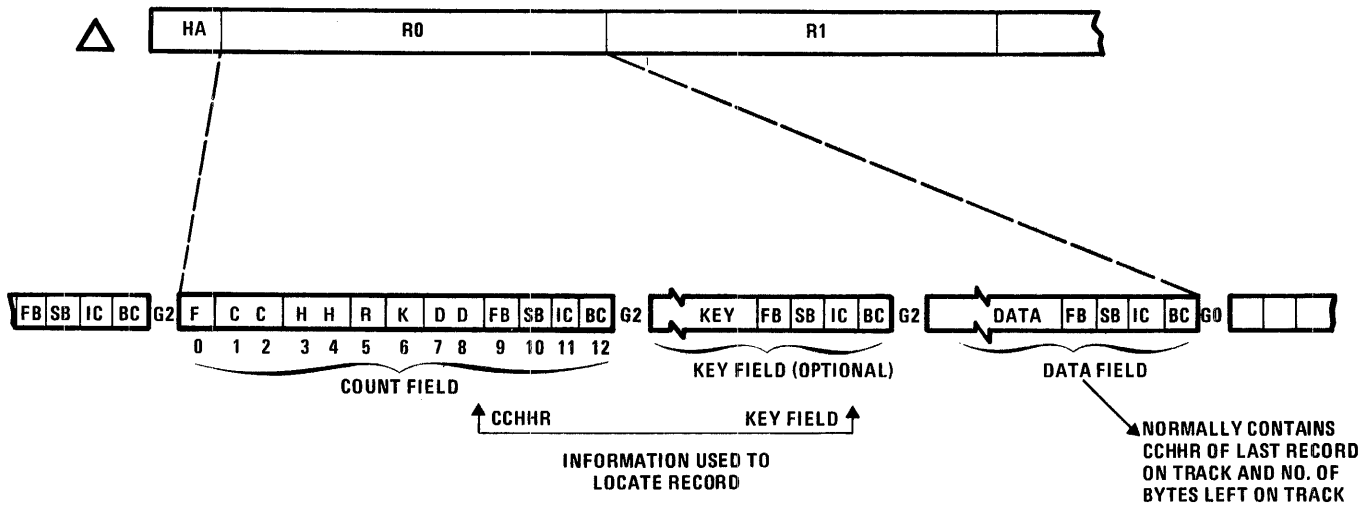
There are three types of gaps, as follows: (Figures 6-11 and 6-14):

- Gap 4 (G4): Gap between index marker and home address field — fixed length of 73 bytes.
- Gap 2 (G2): Gap between home address field and R0, count field, between RN count and key fields, and between RN key and data fields — fixed length of 41 bytes.
- Gap 0 (G0): Gap between data field of one record and count field of next record — variable length of 41 bytes plus 0.043X ($K_L + D_L$) bytes of the previous record (does not include burst check bytes).



FLAG (F)	1 BYTE	INDICATES TRACK CONDITION. BITS 0-5 ALWAYS ZERO. BIT 6 INDICATES TRACK CONDITION, 0=OPERATIVE, 1=DEFECTIVE,BIT 7 INDICATES TRACK USE 0=PRIMARY, 1=SECONDARY.
CYLINDER (C)	2 BYTES	CONTAIN CYLINDER ADDRESS OF TRACK. FIRST BYTE ALWAYS ZERO.
HEAD (H)	2 BYTES	CONTAIN HEAD ADDRESS OF TRACK (00-19). FIRST BYTE ALWAYS ZERO.
CYCLIC CHECKS	2 BYTES	CONTAINS RESIDUAL COUNT OF COMPLEMENTED EXCLUSIVE OR OF ALL DATA BITS IN THE FIELD (DOES NOT INCLUDE BRS).
INDICATOR BYTE (IB)	1 BYTE	ALWAYS SET TO ZERO.
BIT COUNT (BC)	1 BYTE	CONTAINS COMPLEMENT OF RESIDUAL COUNT OF ALL ONE BITS IN THE FIELD. INCLUDED ARE BITS IN THE SYNC BYTE, DATA FIELD AND FIRST CYCLIC CODE CHECK BYTE.

Figure 6-12. Home Address



F = FLAG	1 BYTE		
CC = CYLINDER NUMBER**	2 BYTES		
HH = HEAD NUMBER**	2 BYTES		
R = TRACK RECORD NUMBER	1 BYTE	– INDICATES SEQUENTIAL NUMBER OF RECORD ON THE TRACK	
K = KEY FIELD LENGTH	1 BYTE	– DOES NOT INCLUDE BURST CHECK BYTES	
DD = DATA FIELD LENGTH	2 BYTES	– DOES NOT INCLUDE BURST CHECK BYTES	
FB = FIRST BYTE OF CYCLIC CODE	1 BYTE		BURST CHECK
SB = SECOND BYTE OF CYCLIC CODE	1 BYTE		
IC = INDICATOR BYTE*	1 BYTE		
BC = BIT COUNT BYTE	1 BYTE		

* IC ALWAYS SET TO ZERO
** FIRST BYTE ALWAYS SET TO ZERO

R0 – RN COUNT FIELD FLAG BYTE

BIT 0 AM DETECTION – ALWAYS ZERO IN R0, ALTERNATES ZERO TO ONE IN SUBSEQUENT RECORDS
BITS 1-5 ALWAYS ZERO
BIT 6 TRACK CONDITION
0 = OPERATIVE
1 = DEFECTIVE
BIT 7 TRACK USE
0 = PRIMARY
1 = SECONDARY

Figure 6-13. Record Zero

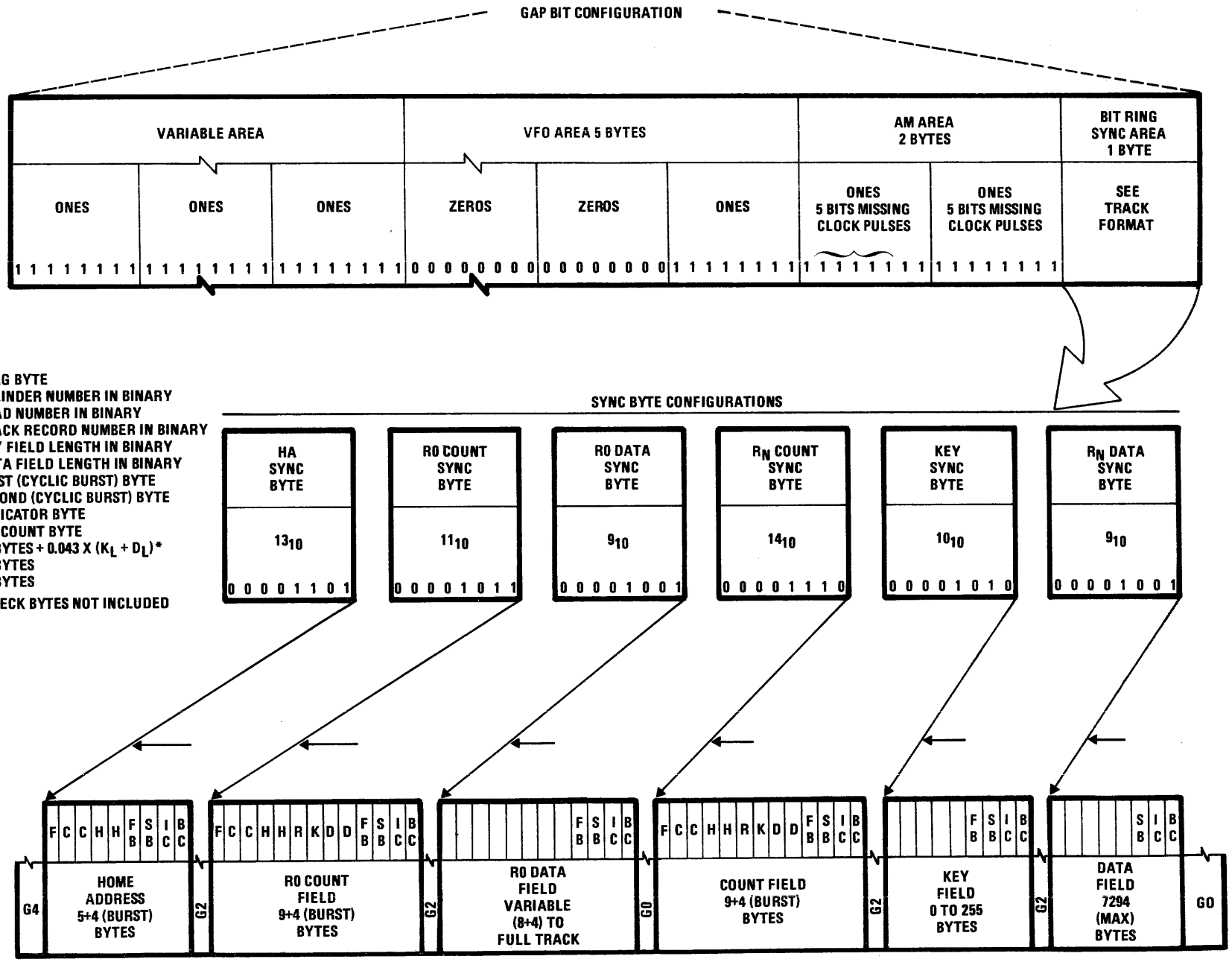


Figure 6-14. Gaps

SPECIAL TRACK FORMAT FOR CS LOAD

A non-standard track format and BRS code are used for CS Load operations (Figure 6-15) to facilitate hardware recognition of CS data. The R1 count field on each CS track is modified to include only micro instructions and is preceded by a special BRS which is recognized by CS load logic in the IFA. All remaining fields on the track are ignored.

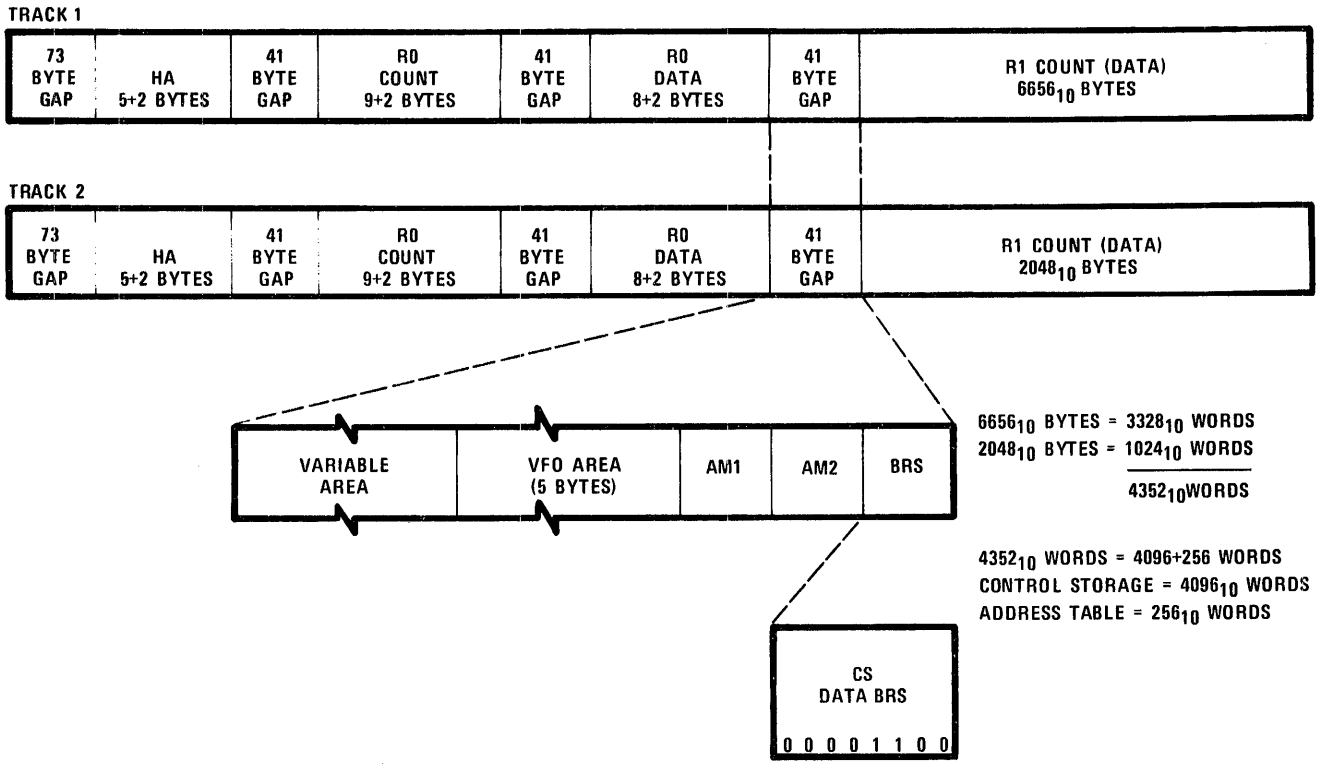


Figure 6-15. Special Track Format for CS Load (Drive Zero)

POWER SEQUENCING

Two types of power sequencing may be used in the 7300 Computer system:

Remote Mode – Power on sequence controlled by the system control unit

Local Mode – Power on sequence controlled by the 7300 Processing Unit

In order for the Power Sequencer Control to power up a drive, all of the following conditions must be met: disc pack installed, disc pack cover closed, main power switch ON, START/STOP switch in START position.

Applying power to the 7300 Processing Unit simultaneously supplies AC power to the DC power supplies in all disc drives connected to the Processing Unit (Figure 6-16). Thereafter, Power Sequencer Control enters a clear state for approximately 1 second to allow the drive power supplies to stabilize. At the end of this delay, Power Sequencer Control completes a ground path for all of the drives applying +36 VDC to the first drive. This activates a relay network in the first drive allowing AC power to be applied to the spindle motor. When the spindle motor in the first drive reaches 70% of its operating speed, a signal is sent to the next drive to allow AC power to be applied to its spindle motor. This process is repeated until all drives have been powered up. The last drive in the string returns a signal to Power Sequencer Control terminating the disc power on sequence. Convenience outlets on all of the drives are energized when system AC power is applied to the computer.

In the event that a drive has not met all of the power up conditions (i.e., main power switch OFF, etc.) it is bypassed and the power up control signals are fed to the next drive in the string.

NOTE

Failure to place a disc pack in one of the drives or spindle motor malfunction will disrupt power sequencing on all succeeding drives in the string if the START switch on the unit is left in the ON position.

First Seek

When the spindle motor on each drive reaches 70% of its operating speed and a 60 second (30 second optional) warm up delay has timed out the read/write heads are forced out to the forward stop. After coming to rest at the forward stop, the head mechanism performs a normal *Reverse Seek* to cylinder 000 and stops. As each drive positions at cylinder 000 it sends an ATTENTION signal to the IFA to indicate that it is ready for further instruction. Failure to complete a normal *Reverse Seek* operation within 100 milliseconds causing a SEEK INCOMPLETE signal and an ATTENTION signal to be sent to the IFA.

NOTE

Removing and reinserting the Unit Select Plug will also cause the drive to perform a FIRST SEEK operation.

Power Down

A *power down* sequence is initiated by de-energizing relay K10 in Power Sequence Control removing the common ground from all of the drives (Figure 6-16). After a delay of approximately 1 second, AC line power is removed provided relay K12 in Power Sequence Control is de-energized. Relay K12, de-energized, indicates that the heads in all of the drives have been retracted.

The following abnormal operating conditions will automatically retract the read/write heads and *power down* the affected drive:

1. Loss of primary (AC) power

2. Loss of or out-of-tolerance logic voltages

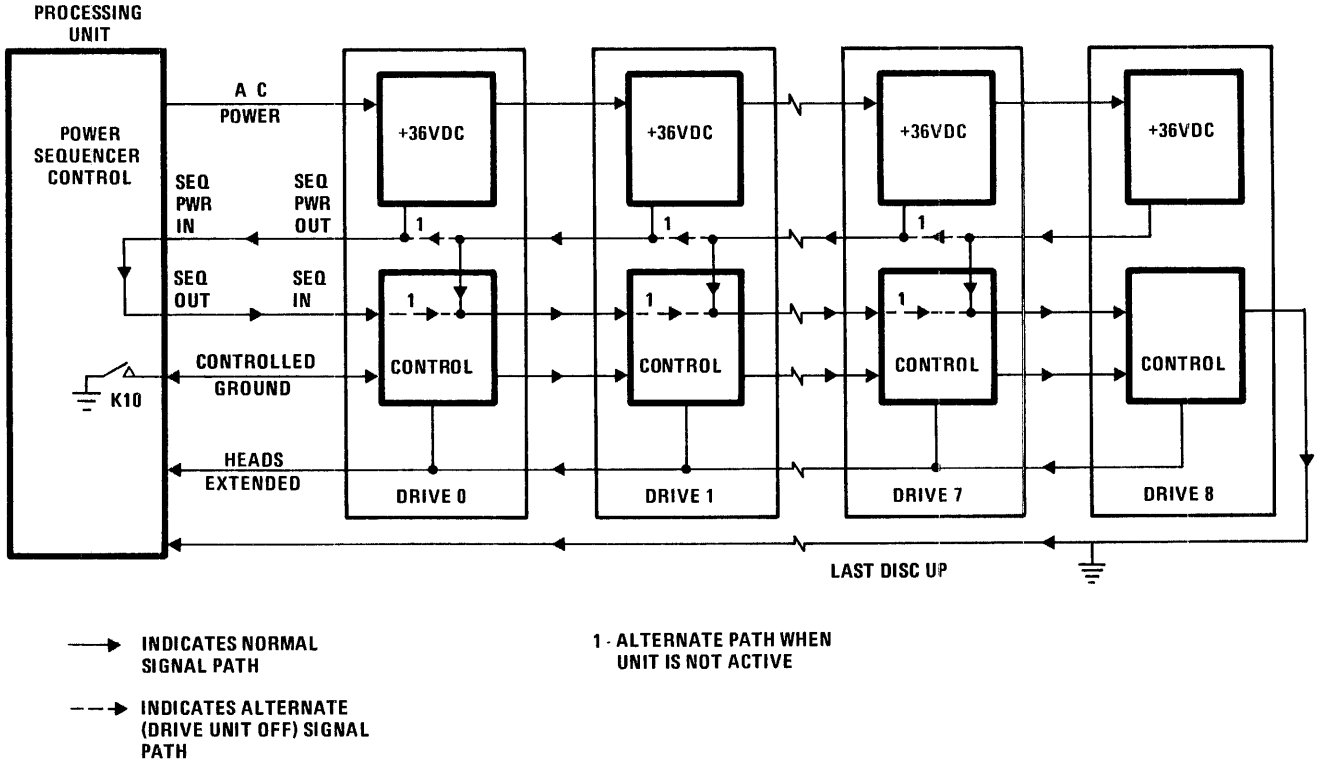


Figure 6-16. Power Sequencing Control Lines

3. Opening the disc compartment door before pressing the STOP switch

4. Pressing the STOP switch with the heads extended

FUNCTIONAL LOGIC DESCRIPTION

The following paragraphs provide a description of the major logical components of the IFA including all interface signals exchanged between the IFA and shared resources and between the IFA and each of the drives (Figure 6-17). For this discussion the IFA logic has been separated into eleven major functional categories as follows:

- Shared Resources Interface
- Drive Interface
- Drive Selection
- Control Commands
- Request Status Logic
- Read/Write Control
- Data Detection Logic
- Read/Write Timing
- Data Format
- Error Detection
- CS Load Control

SHARED RESOURCES INTERFACE

A single 16-bit Group III extended register is used to interface all instructions, data, and status responses between the IFA and shared resources. The type of operation to be performed by the IFA is determined by the number ($10\text{-}1F_{16}$) used to address the extended register. Table 6-4 lists the register addresses and corresponding instruction categories recognized by the IFA.

Table 6-4. Instruction Selection

Register Designation (Hex)	Instruction Type
Write Register 10	Select Drive
Write Register 11	Commands
Write Register 12	Request Status
Read Register 12	Read Hardware Status
Write Register 13	Request Attention and Physical Drive Number Status
Read Register 13	Read Attention and Physical Drive Number Status
Write Register 14	Write Data
Write Register 15	2nd Word of Read or Write Command
Write Register 17	Request Cylinder Number Status
Read Register 17	Read Cylinder Number Status
Read Register 10-1F	Read Data

Tables 6-5 and 6-6 name and describe the interface lines between shared resources and the IFA.

Table 6-5. Signals from Shared Resources

Name	Description
EXTENDED REGISTER NUMBER (ERNG-3)	Designates extended register address ($10\text{-}1F_{16}$)
EXTENDED REGISTER OUT (ERO-3)	Instructions or data to the extended register (MSB is bit 00)
MASTER CLEAR INPUT/OUTPUT (MCI/O)	MC I/O performs all of the following: 1. Clears all bus gate selections 2. Selects drive 0 3. Clears read/write and CS load control logic 4. Presets the Data Byte Counter (DBC) to 6400_8 words

Table 6-5. (Continued)

Name	Description
	MCI/O is generated by: 1. Power on sequence or 2. Pressing SYSTEM RESET pushbutton or 3. AUTOLOAD signal from the IFA or 4. Pressing AUTOLOAD pushbutton or 5. Pressing RESET/LOAD pushbutton if in normal mode 6. RSDS signal from IFA if in normal mode
DISC SOURCE (DISCS)	AUTOLOAD SELECT switch is in the PRIMARY (down) position. Forces micro instruction execution to begin at address 0113_{16} following *CS Load*. DISCS must be high during *CS Load* and *Autoload*.
MAINTENANCE OUT (MANO)	The computer is operating in maintenance mode (MAINTENANCE MODE pushbutton has been pressed).
DEADSTART INITIATE (DOA)	Initiates a *CS Load* operation. DOA is generated on the trailing edge of a power on/MC sequence, on the trailing edge of a Deadstart MC sequence resulting from a RSDS signal from IFA, or by pressing the RESET/LOAD pushbutton (Normal mode only). DOA drops when: 1. ENDO signal is received or 2. SYSTEM RESET pushbutton is pressed or 3. RSDS signal is received
END OUT (ENDO)	Control storage and address table are full or SYSTEM RESET pushbutton has been pressed. ENDO is high 1-2 microseconds after the next to the last word has been transferred, minimum width is 3 microseconds.
POWER ON/MC	Clears first seek detect logic prior to a *Deadstart CS Load*.
EXTENDED REGISTER WRITE (ERFG3WR)	Shared resources is attempting to store an instruction or data word in the extended register. Active whenever an AND, CLR, EOR, IOR, SDB, SDW, STA or STB micro-instruction is being executed that addresses the extended register.
EXTENDED REGISTER READ (ERFG3RD)	Shared resources is attempting to load data or status information from the extended register. Active whenever a CLA, LAB, LAW, LAW/, LBL, LBW or LBW/ micro-instruction is being executed that addresses the extended register.
EXECUTE-3 (EXCT-3)	Processor 3 is in the execute portion of a time slice.
SHARED RESOURCES CLOCK (T00, T20, T40, T60, T80)	Leading edge occurs at T00, T20, T40, T60 and T80. Trailing edge occurs 30 nanoseconds after leading edge.
INPUT/OUTPUT EXIT (I/O EXIT)	Terminates a *Read* or *Write* operation when: 1. Register a=b during execution of a CI01 command or a≠b during execution of CI02 command or 2. An EOT-3 signal is generated.

Table 6-6. Signals to Shared Resources

Name	Description
EXTENDED REGISTER IN (ERI-3)	Data or status information to shared resources (MSB is bit 00).
ATTENTION-3 (ATTN-3)	ORed ATTENTION signal lines from all of the drives except drive S indicating: 1. One or more of the drives has successfully completed a *Seek* operation or 2. 100 milliseconds has elapsed since a *Seek* operation was initiated on one of the drives and detenting did not occur. ATTN-3 is delayed following a *First Seek* operation until a *Request Attention* command is executed by the IFA. ATTN-3 sets the processor 3 Busy FF (request time slice) provided the processor 3 Active FF is not set. ATTN-3 drops when all of the logical attentions in the drives have been cleared.
DEADSTART RESTART (DSRS)	A status error has been detected by the IFA during a *CS Load* operation or a *CS Load* command has been translated by the IFA, 50 nanosecond minimum width. DSRS will be high for any of the following status indications: 1. Burst Check Error 2. End of Cylinder 3. No Sync Find 4. Seek Incomplete A DSRS signal generates a DOA signal for all status conditions except Burst Check Error in maintenance mode. The DSRS signal is also used internally in the IFA forcing a *CS Load Complete* condition when operating in maintenance mode.
DISC DATA STROBE (DDS)	A word of CS data is available in the extended register (a 70 nanosecond pulse).
END OF TRANSFER-3 (EOT-3)	Terminates a *Read* or *Write* operation and generates an I/O EXIT signal when any of the following status conditions occur: 1. Command Early 2. Missed Window 3. Track Boundary 4. Read/Write Termination 5. Lost Data 6. No Sync Find EOT-3 is also generated on Missed Window Status while executing a *CS Load* command in normal mode. EOT-3 becomes inactive when the status condition is cleared.
AUTOLOAD	Initiates a MC Sequence. On the trailing edge of MC, Autoload activates processor 4 by setting its Busy FF and forces the Control Storage Address register to 0113_{16} on the first major cycle allocated to processor 4 (provided AUTOLOAD SELECT is set to PRIMARY).
RESTART DEADSTART DISABLE (RSDSDIS)	Prevents DSRS from generating a DOA signal when a Burst Check Error occurs in maintenance mode – occurs 50 nanoseconds before DSRS pulse and remains active for 50 nanoseconds.
REQUEST-3 (REQ-3)	The IFA is ready to transmit or receive data or status information – sets the Busy FF (request time slice). When transferring data, REQ-3 becomes inactive when EXCT-3 goes active. When transferring status responses REQ-3 becomes inactive when a *Request Status* instruction is executed.

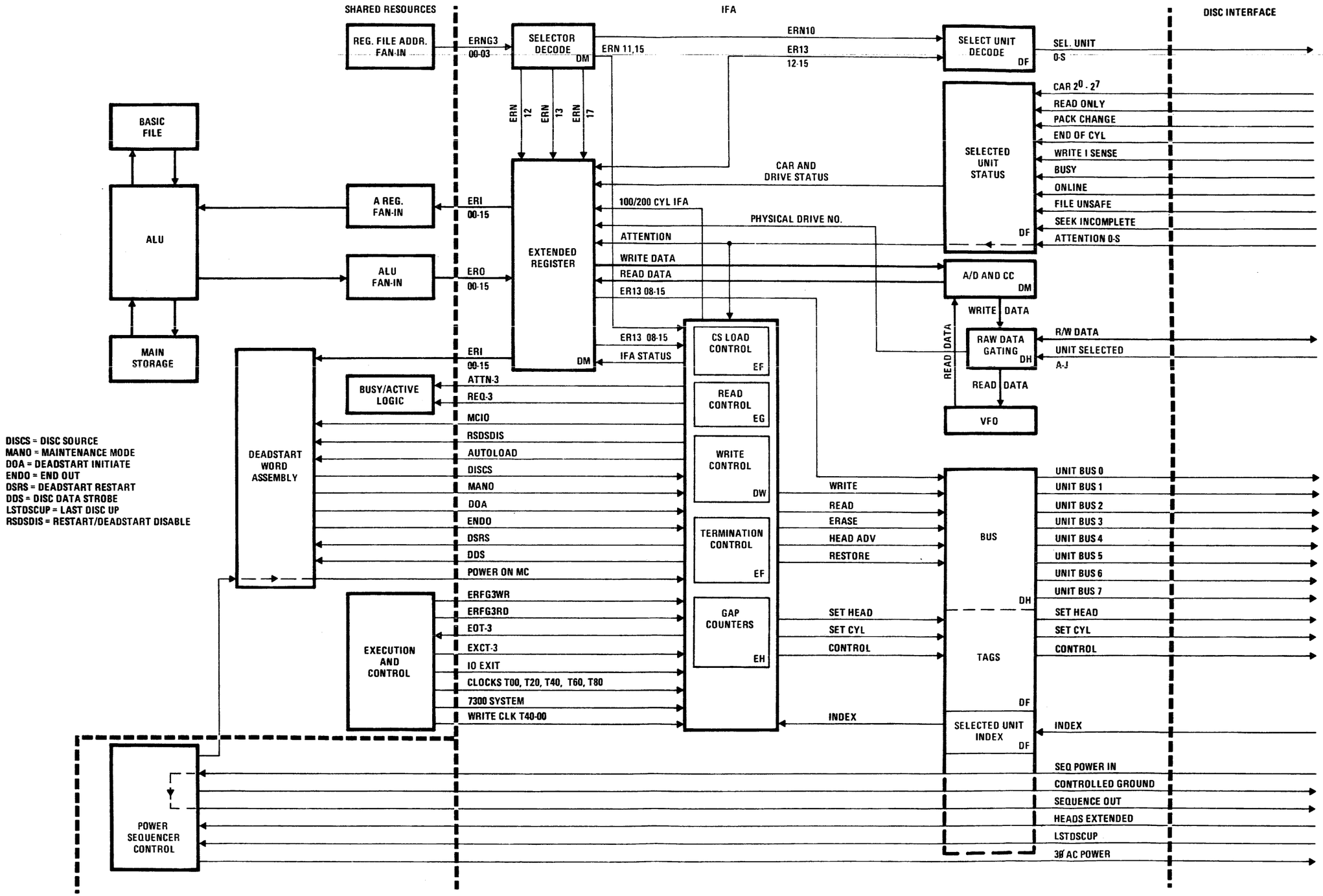


Figure 6-17. IFA Signal Exchange

DRIVE INTERFACE

All outgoing signal lines are terminated in the last drive in a string (except read/write data), and that drive requires +3 volts from the control unit for the terminator. All incoming signal lines are terminated in the IFA and each drive requires -3 volts for the line drivers.

Disc Line Drivers (Figure 6-18)

- The high (true) level (logical 1) occurs when the line signal is -1.52 volts or less.
- The low (false) level (logical 0) occurs when the line signal is +1.15 volts or more.

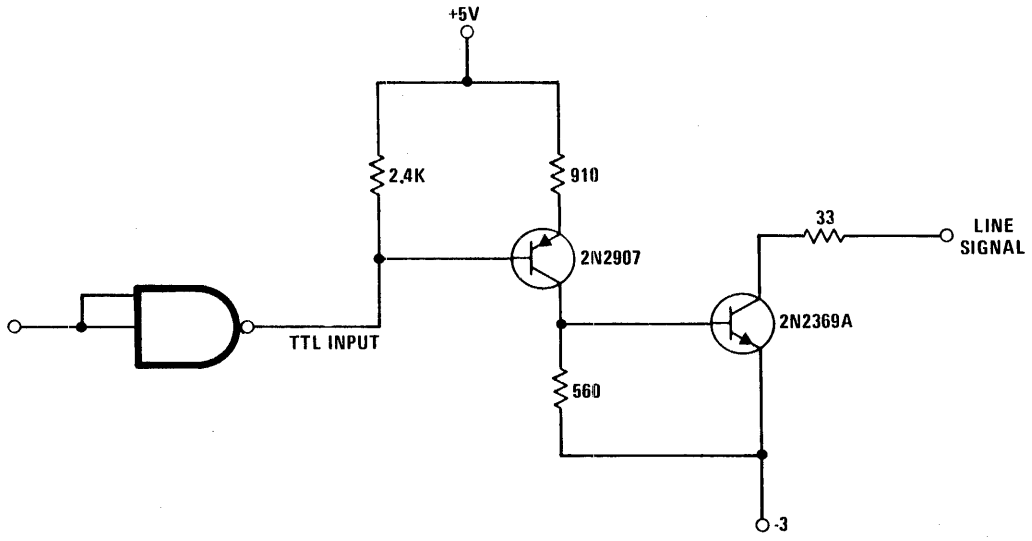


Figure 6-18. Typical Line Driver

Disc Line Receivers (Figure 6-19)

- The high (true) level (logical 1) occurs when the line signal is -0.2 volt or less.
- The low (false) level (logical 0) occurs when the line signal is +0.4 volt or more.

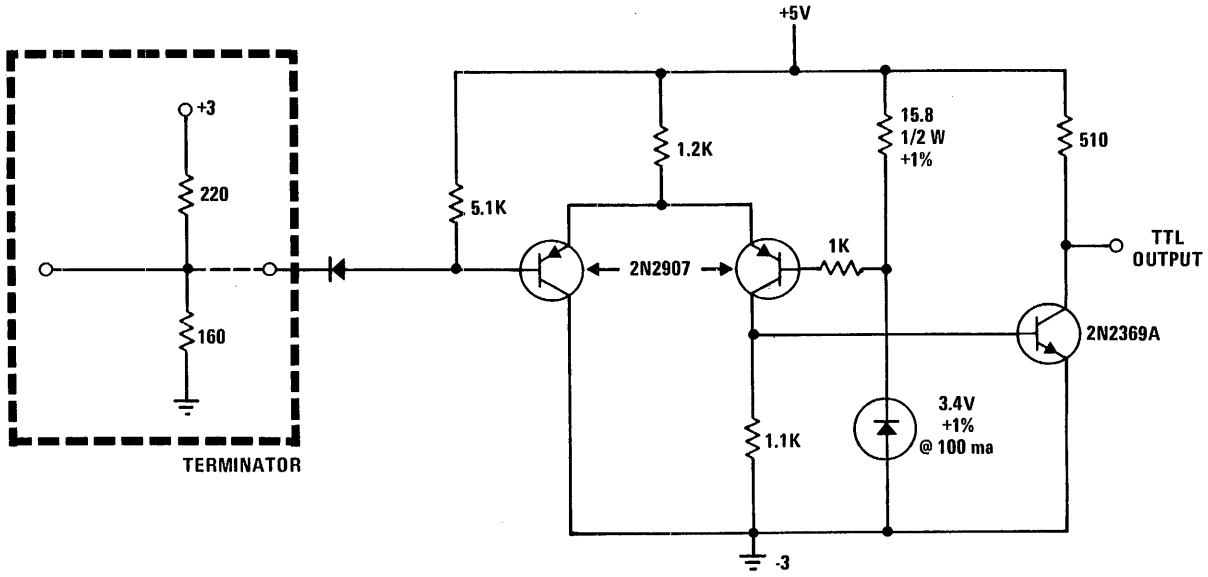


Figure 6-19. Typical Line Receiver and Terminator

Disc Read/Write Multiplexer (Figure 6-20)

- The high (true) level (logical 1) occurs when the line signal for the receiver (write data input) is +1.92 volts or more.
- The low (false) level (logical 0) occurs when the line signal for the receiver (write data input) is +1.23 volts or less.
- The high (true) level (logical 1) occurs when the line signal for the driver (read data output) is +2.86 volts or more.
- The low (false) level (logical 0) occurs when the line signal for the driver (read data output) is +1.23 volts or less.

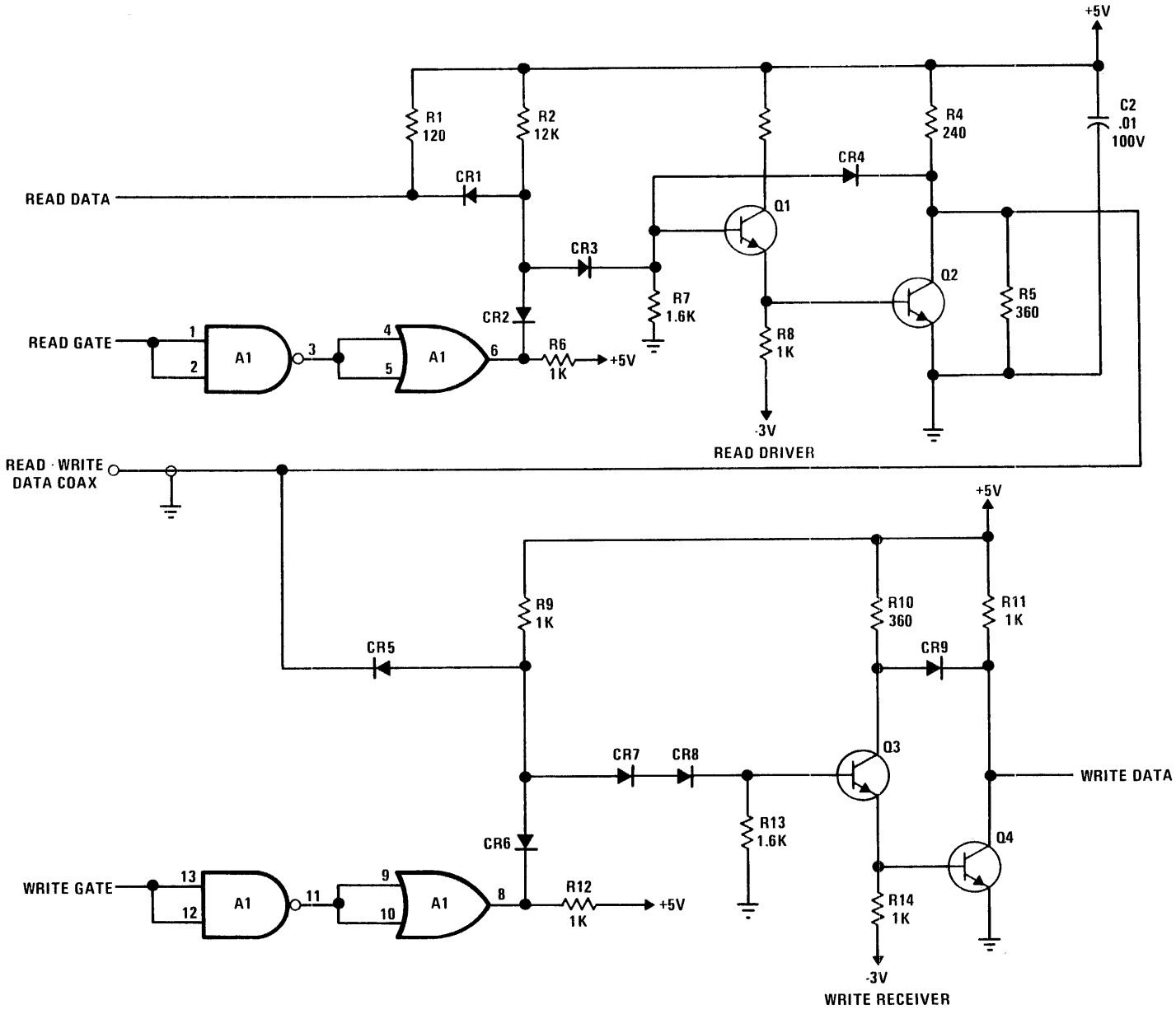


Figure 6-20. Read/Write Data Input

Drive Interface Lines

Drive interface signals are grouped by cable class (Bus or Unit). The signals are described in Tables 6-7 through 6-11 below. Pin assignments for the drive interface logic cables (Bus and Unit) are found in Appendix 6B.

Table 6-7. Bus Line Signals to the Drives from IFA

Name	Description
UNIT BUS LINES (eight)	These time-shared lines are used to select command operations or transmit new cylinder or head address information under *Tag Line* control. A drive must be selected before these lines are active. Refer to individual *Tag Line* descriptions for precise translation of these lines.
CONTROL TAG	Enables the Unit Bus Lines, to a selected drive for command functions as follows: Line — Definition Unit Bus 0 — Write Gate Unit Bus 1 — Read Gate Unit Bus 2 — Seek Start Unit Bus 3 — Reset Head Register Unit Bus 4 — Erase Gate Unit Bus 5 — Select Head Unit Bus 6 — Restore Unit Bus 7 — Head Advance NOTE: Unit Bus 1 and 3 enabled simultaneously for a minimum of 25 milliseconds by Control Tag causes a *Diagnostic Reset* of a program initiated File Unsafe condition.
SET CYLINDER TAG	Enables the Unit Bus Lines to a selected drive for the transmission of new cylinder address information as follows: Line — Definition Unit Bus 0 — Cylinder 128 Unit Bus 1 — Cylinder 64 Unit Bus 2 — Cylinder 32 Unit Bus 3 — Cylinder 16 Unit Bus 4 — Cylinder 8 Unit Bus 5 — Cylinder 4 Unit Bus 6 — Cylinder 2 Unit Bus 7 — Cylinder 1
SET HEAD TAG	Enable the Unit Bus Lines a selected drive for the transmission of new head address information as follows: Line — Definition Unit Bus 3 — Head 16 Unit Bus 4 — Head 8 Unit Bus 5 — Head 4 Unit Bus 6 — Head 2 Unit Bus 7 — Head 1 NOTE: The Head Address register in the selected drive must be cleared by a *Reset Head Register Control* command prior to transmitting new head address information.
SELECT UNIT LINES (nine)	Selects the logical drive unit. For all drive selections except unit S, the drive remains selected until another drive is selected. A MASTER CLEAR (MC) automatically selects unit 0.

Table 6-8. Bus Line Signals to the Drives from Power Sequencer Control

Name	Description
CONTROLLED GROUND	This line provides a ground path for the +36 volt power supplies in each of the disc drives for powering the drives *up* and *down* along with the processing unit. The only prerequisite is that the START/STOP switch is set to START on the drives to be controlled. This line must be at a ground potential (power up on the processing unit) for the START/STOP switch to power up any drive.
SEQUENCE IN	This line provides +36 VDC from the preceding drive or from power sequencer control to start the power up sequence of spindle motors in each drive. SEQUENCE IN is activated approximately 1 second after CONTROLLED GROUND is activated to allow the bulk power supplies in each drive to stabilize.
SEQUENCE OUT	This line passes +36 VDC to the succeeding drive in the multiplex cable string to power up the spindle motor in each drive whenever any of the following conditions exist. • Preceding drive MAIN POWER switch set to OFF • Preceding drive START/STOP switch set to STOP • Preceding drive MAIN POWER switch ON, START/STOP switch set to START and spindle drive motor at 70% of rated speed. NOTE: By connector definition this line runs only between drive units. In the cable segment between Power Sequencer Control and the first drive unit, this signal becomes SEQUENCE IN to the first drive (Figure 6-16).

Table 6-9. Bus Line Signals to IFA from the Drives

Name	Description
CYLINDER ADDRESS LINES (eight)	These lines transmit the current binary address of the head carriage on the selected drive provided a *Start Seek* command has been executed since the Cylinder Address register was loaded.
ATTENTION LINES (nine)	This line is activated when: a normal *Seek* is completed; or 100 milliseconds has elapsed since a *Seek* operation was initiated and detention did not occur; or when the corresponding Unit Number Plug is changed. This line does not require unit selection. An individual ATTENTION line may be deactivated by selecting the appropriate drive and executing a *Reset Attention* or Read command. All ATTENTION lines except ATTENTION-S are ORed together to produce ATTN-3 which is sent to shared resources. NOTE: During a *First Seek* operation ATTN-3 is delayed until a *Request Attention* command is executed by the IFA.
STATUS LINES (eight)	These lines inform the IFA of certain error conditions occurring in the selected drive. Drive status information is made available to shared resources where a *Request Status* instruction is executed by the IFA. Refer to the section on operation for a detailed description of these lines.
SELECTED INDEX	This line is active for 30 microseconds after an index mark is detected on the selected drive. Index mark occur every 25 milliseconds.

Table 6-10. Bus Line Signals to Power Sequencer Control from the Drives

Name	Description
SEQUENCE POWER IN	This line provides +36 VDC from a drive in the string to be used for power sequencing.
SEQUENCE POWER OUT	This line passes +36 VDC to a drive in the string to be used for power sequencing. Power is derived from the drive internal power supply when the drive is up to speed or from a succeeding drive when the drive in question is off. This voltage is returned to the drive in question as SEQUENCE IN. NOTE By connector definition this line runs only between drive units in the cable segment between Power Sequencer Control and the first drive unit. SEQUENCE POWER OUT from the first drive becomes SEQUENCE POWER IN to Power Sequencer Control (Figure 6-16).
LAST DISC UP	This line becomes active when the spindle motor on the last drive in the string has been brought up to 70% of running speed. LAST DISC UP is used by Power Sequencer Control to terminate the power up sequence.
HEADS EXTENDED	This signal prevents Power Sequencer Control from removing AC power until the read/write heads on all drives in the system have been retracted.

Table 6-11. Unit Line Signals to/from Each Drive

Name	Description
READ/WRITE DATA (1 line for each drive)	This line transmits data to or from the selected drive.
UNIT IS SELECTED (1 line for each drive)	This line indicates which of the drives (0-8) is selected.

DRIVE SELECTION LOGIC

Selecting Write Register 10 with an *OUT* machine language instruction performs an *Initial Clear* of the IFA logic and then gates bits 12-15 of the command word into the Drive Number register. At the next clock 20, the command argument is gated to the Drive Unit Decoder. This activates one of the select drive lines which is then compared against the Unit Number Plug in each of the drives (Figure 6-21).

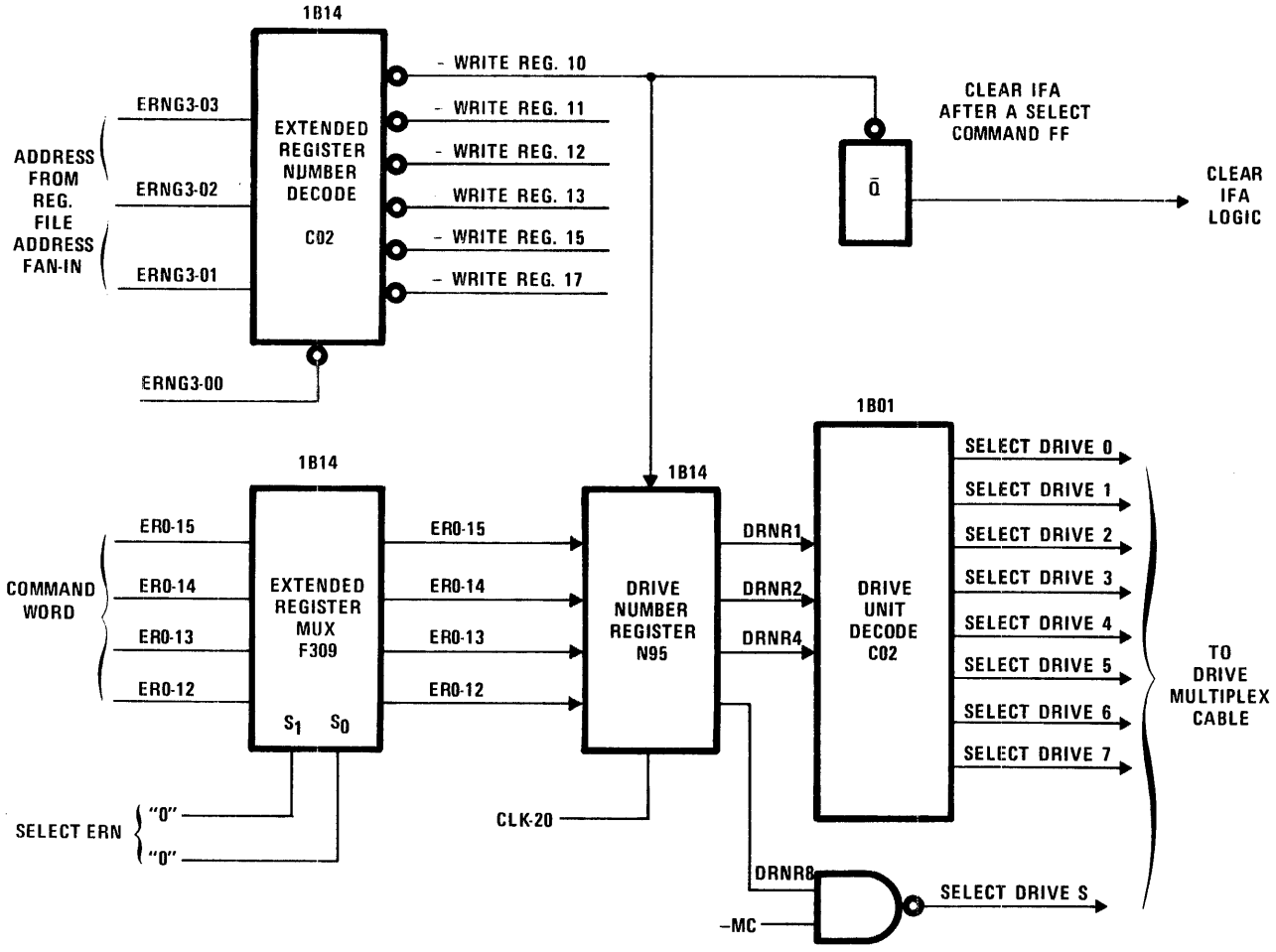


Figure 6-21. Drive Selection

CONTROL COMMANDS LOGIC

Selecting a Write Register 11 control command via an *OUT* machine language instruction gates the command argument into bits 8-15 of the extended register mux. At the following clock 00 the command argument is transmitted to the Bus Gate Mux where it is enabled to the selected drive by the BUS GATE ONE SHOT (BGOS) signal. BGOS is a three microsecond enable generated by addressing Write Register 11 for a non-data transfer operation (Figure 6-22).

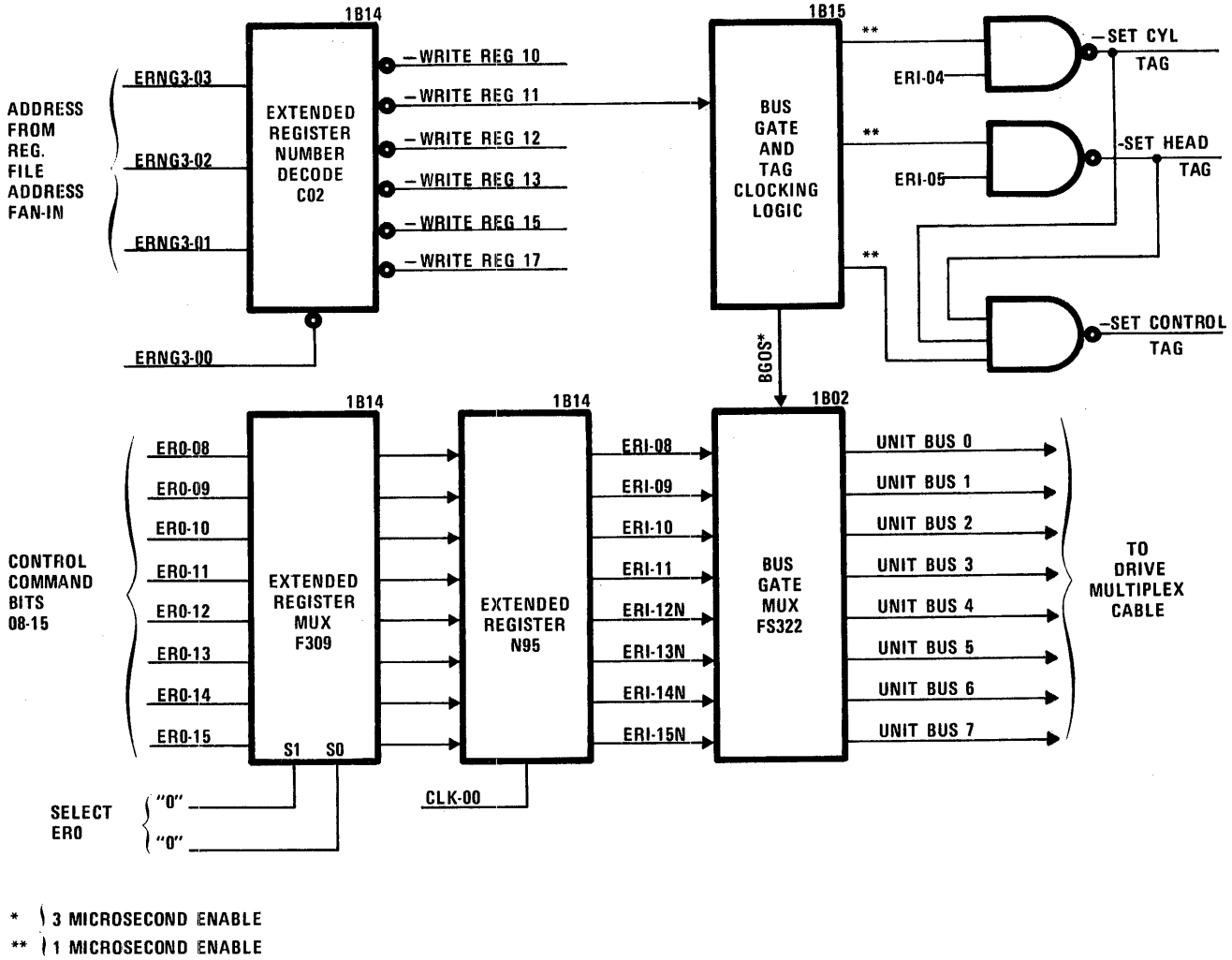
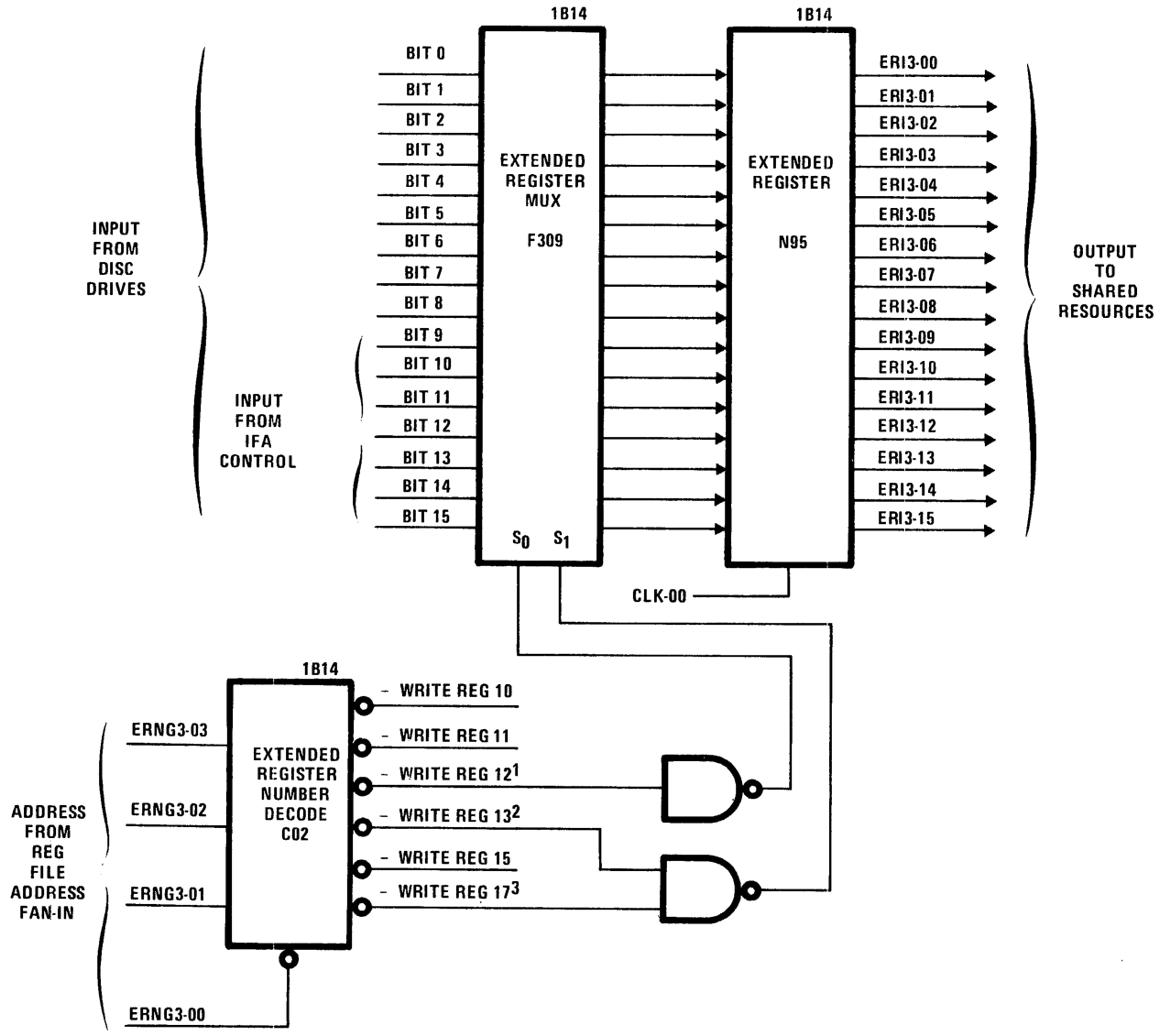


Figure 6-22. Control Commands

REQUEST STATUS LOGIC

A *Status Request* operation is performed by executing an *OUT* machine language instruction specifying Write Register 12, 13 or 17. A command argument is not required. Executing a *Status Request* places the appropriate information into the extended register and clears the corresponding status lines in the drive. To sample the output of the extended register, the operating system must execute an *INP* machine language instruction specifying the same extended register as the previous *OUT* instruction (Figure 6-23).



[1]READ/WRITE STATUS
[2]ATTENTION 0-S, 100/200 CYL IFA, PHYSICAL DRIVE A-J
[3]ATTENTION 0-7, CYLINDER NUMBER

Figure 6-23. Request Status

READ/WRITE CONTROL LOGIC

Control logic interprets commands and control signals from shared resources and uses the information to enable the appropriate timing logic. During a *Write* operation, control logic performs the following functions:

- Stores the command specified data byte count in the Data Byte Counter
- Generates 41 gap bytes and examines the contents of the program loaded Variable Gap Counter to determine the proper number of additional Variable Gap bytes required to precede the information field.
- Stores the command specified Bit Ring Sync in the BRS register for insertion on the disc surface
- Generates REQ-3 to shared resources (activate processor 3) one word time in advance of when IFA is ready to accept each data word
- Gates parallel data from the extended register to data format logic.
- Activates write terminate logic at the end of the burst check to ensure that one additional byte of 1's is written for a normal *Write* operation (or 1's to the end of the track for *Format Write* operations), side trim erase is performed for the equivalent of 6 bytes after the end of data and window logic is set (Figure 6-24).
- Generates REQ-3 to shared resources at the end of every data transfer to activate a micro instruction status check

During a *Read* operation, control logic performs the following functions:

- Stores the command specified data byte count in the Data Byte Counter
- Activates Data Detection logic (VFO)
- Compares the Bit Ring Sync received from shared resources with the Bit Ring Sync received from the disc
- Gates parallel data from data format logic to the extended register
- Generates REQ-3 to shared resources (activate processor 3) when a word of data has been assembled in the extended register
- Activates read terminate logic at the end of burst check to ensure that the window logic is set in preparation for the next *Read* or *Write* command (Figure 6-25)
- Generate REQ-3 to shared resources at the end of every data transfer forcing a micro instruction status check

During a *Read* operation, the VFO logic in IFA examines the pattern of clock and data pulses sent from the disc to differentiate data fields from the gaps between data fields. After locating a field of information, the IFA reformats the data from serial-by-bit to parallel-by-word and transmits the data to shared resources during processor state 3's time slice. A series of flowcharts showing the Read sequence is provided on Figures 6-26 through 6-30.

During a *Write* operation, the IFA receives data from shared resources during processor state 3's time slice, reformats the data from parallel-by-word to serial-by-bit and records the data on a selected disc surface. Clock pulses are generated by the IFA to synchronize the recording of data pulses. Every 400 nanoseconds (one bit cell time) a clock pulse is recorded on the disc. If no other pulse is recorded between two clock pulses the bit cell time represents a logical 0. Two pulses recorded during a bit cell time (one pulse for each 200 nanoseconds) represents a logical 1. A series of flowcharts showing the write sequence is provided on Figures 6-31 through 6-35.

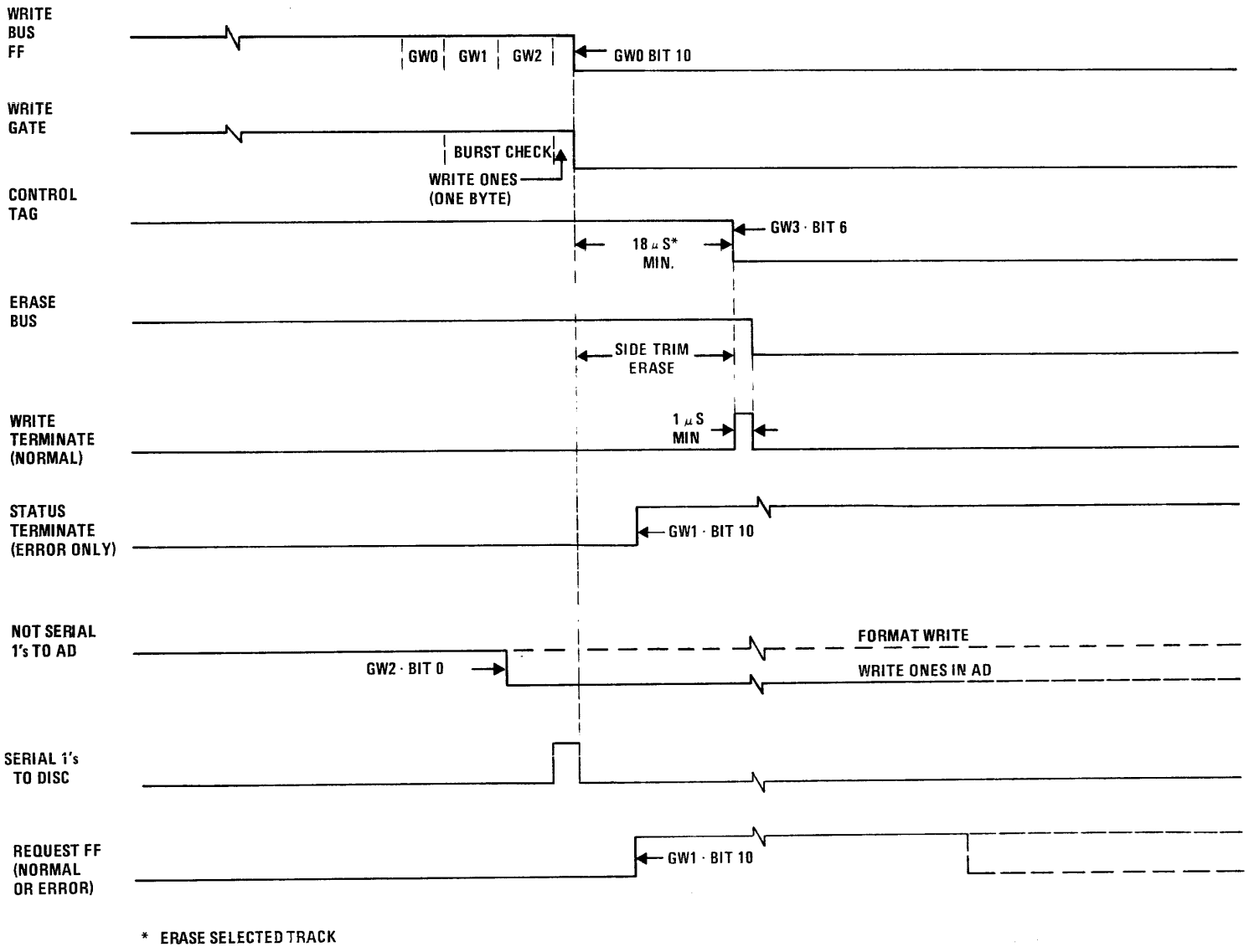
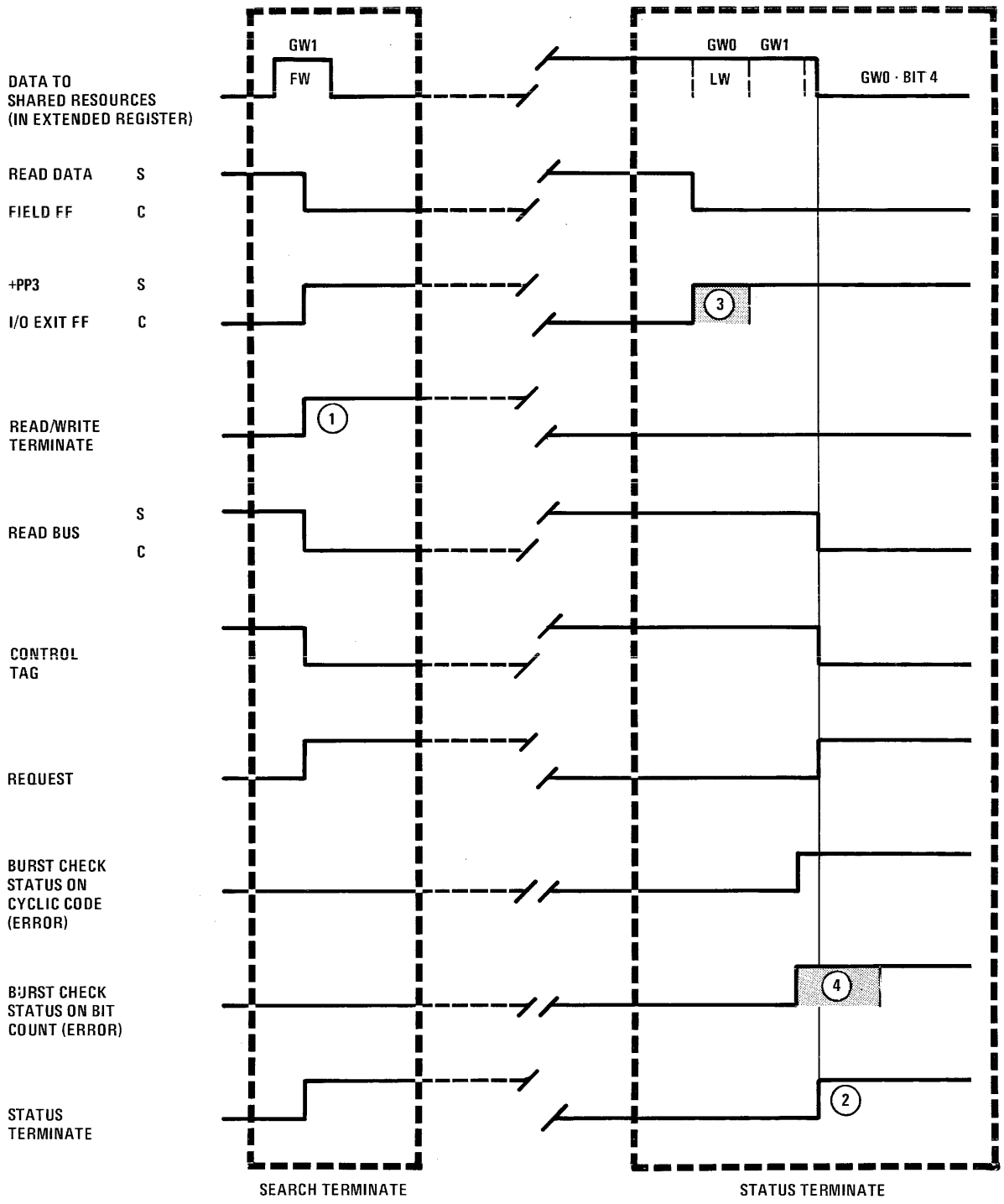


Figure 6-24. Write Terminate



1. READ/WRITE TERMINATE CAN OCCUR ANY TIME BETWEEN THE FIRST WORD (FW) TO SR AND THE NEXT TO THE LAST DATA WORD TO SR.
2. STATUS TERMINATE CAN OCCUR ON NORMAL (COMPLETE) READ, SEARCH MISCOMPARE ON LAST DATA WORD (LW) BURST CHECK ON CC OR BURST CHECK ON BC.
3. I/O EXIT ON LAST WORD CAN OCCUR ANY TIME DURING GW0.
4. BURST CHECK ON BIT COUNT CAN OCCUR ON ANY BIT OF BIT COUNT.

Figure 6-25. Read Terminate

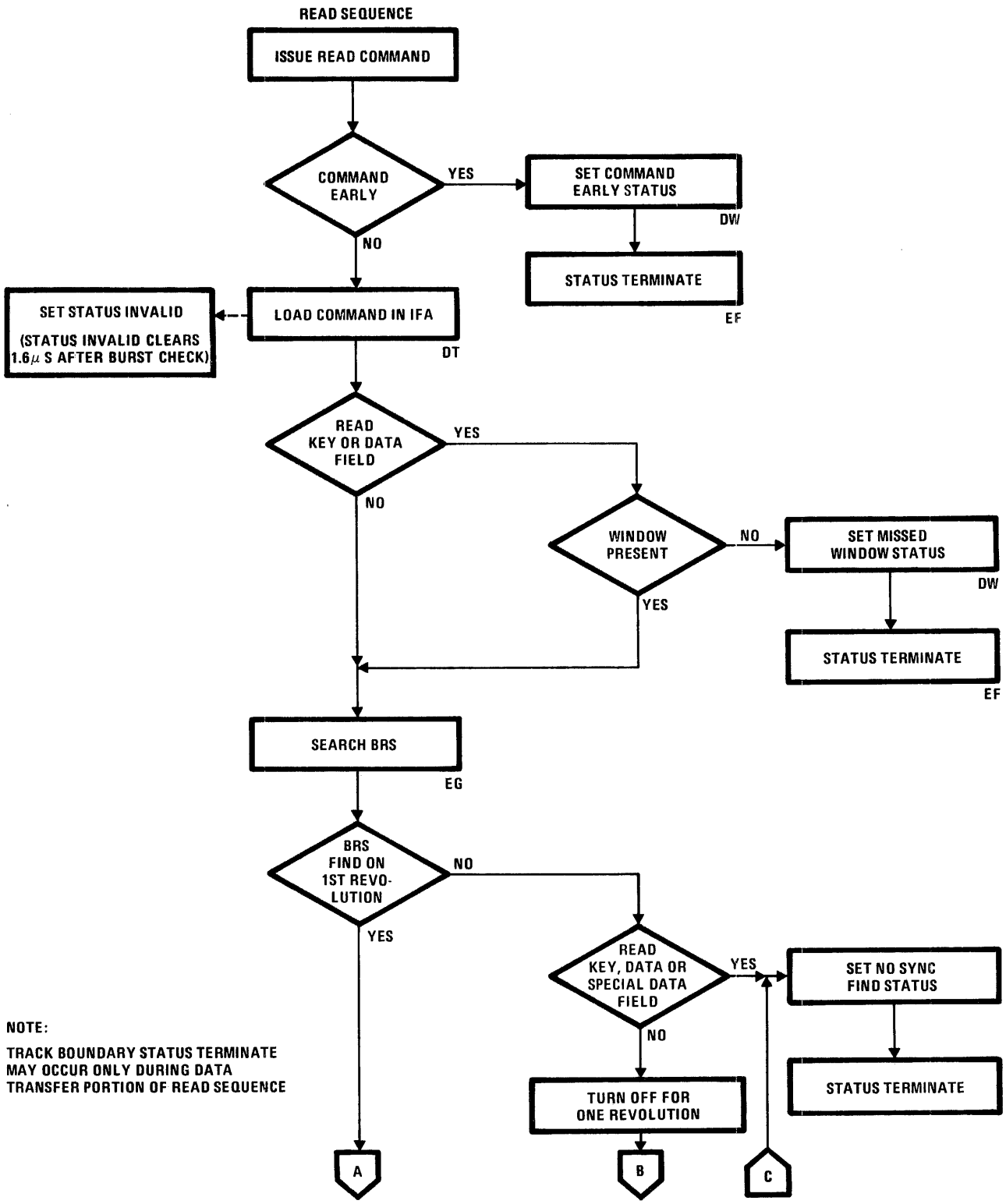


Figure 6-26. Read Sequence

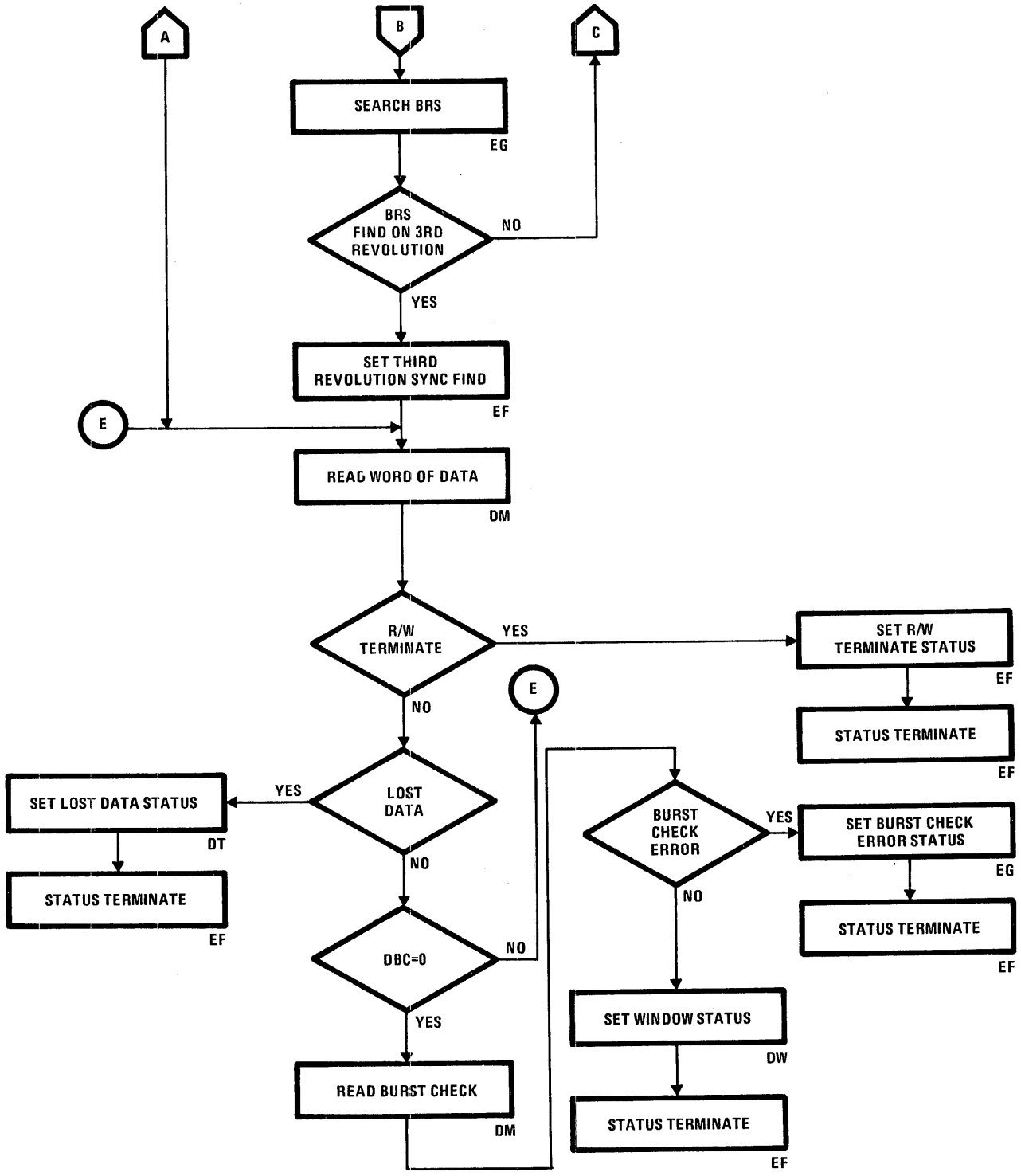


Figure 6-26. Read Sequence (Continued)

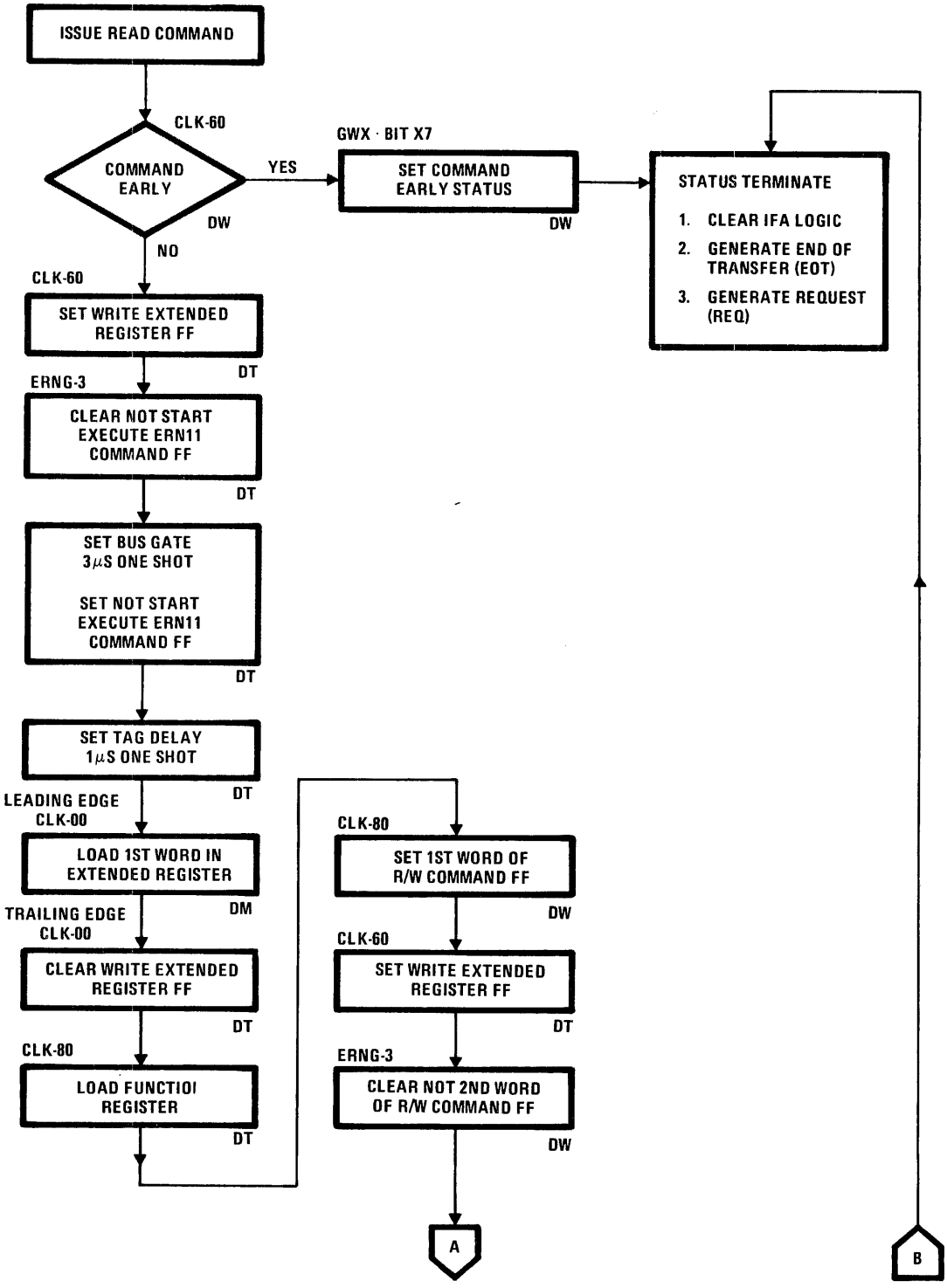


Figure 6-27. Load Read Command

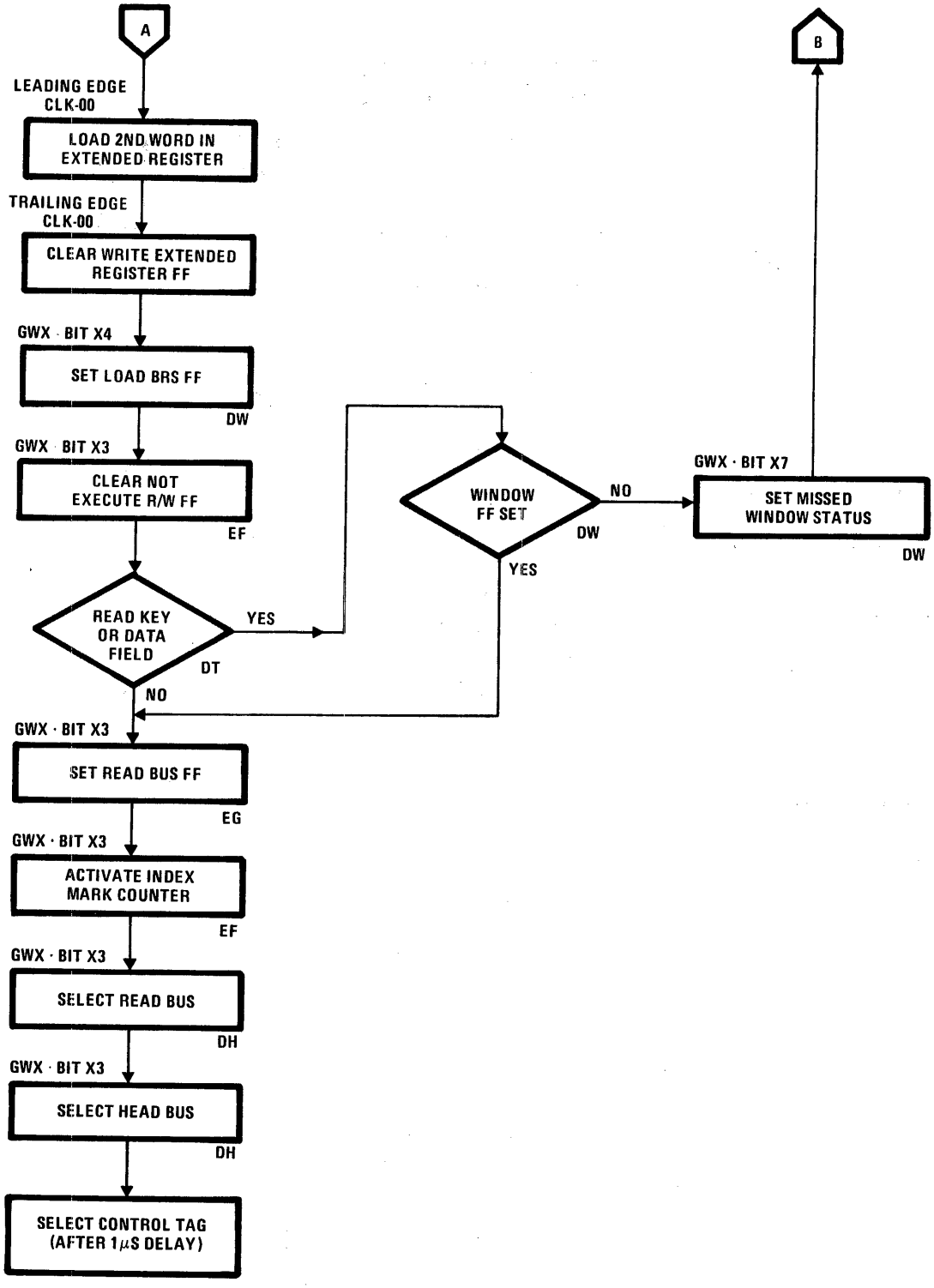


Figure 6-27. Load Read Command (Continued)

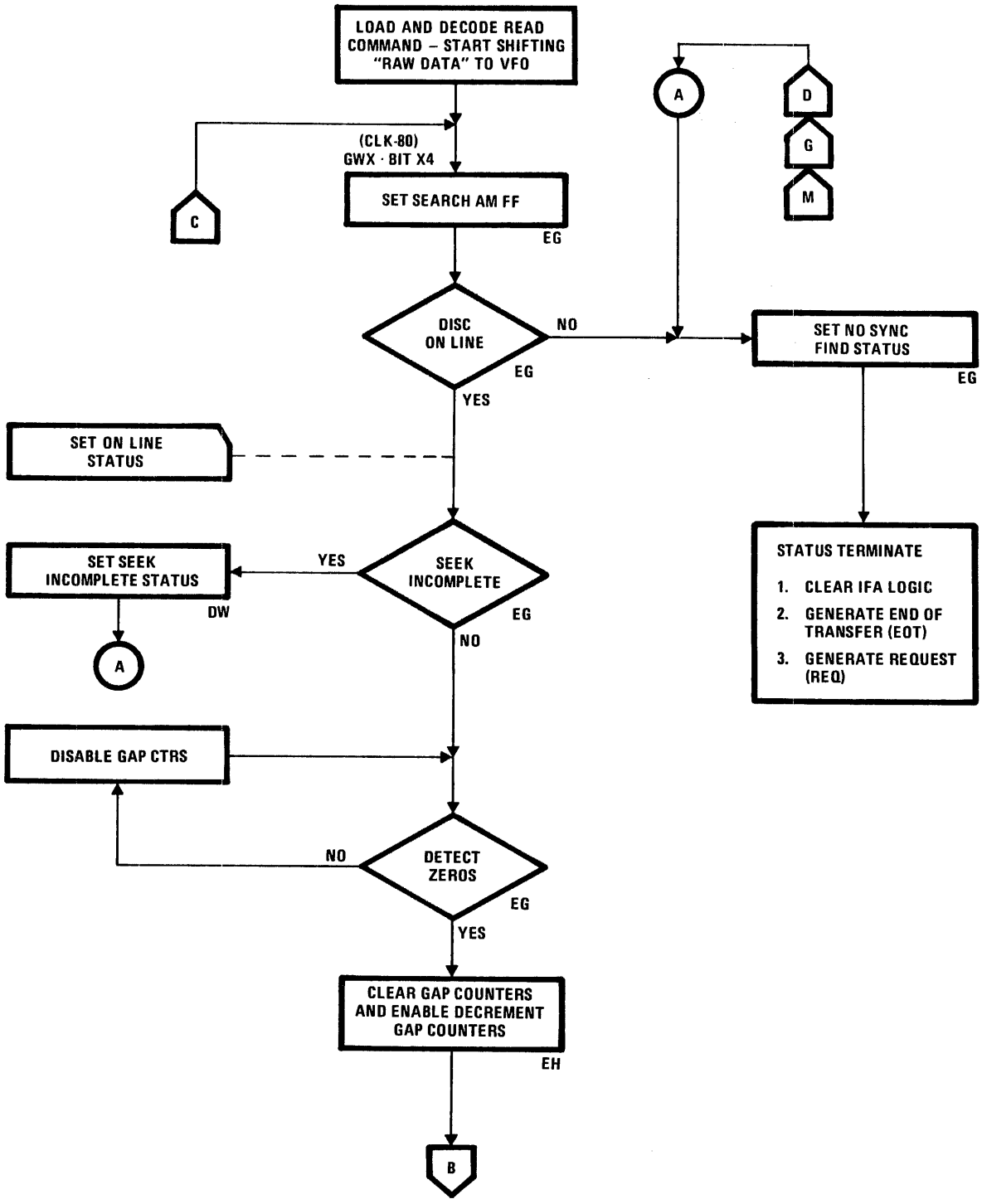


Figure 6-28. Search BRS

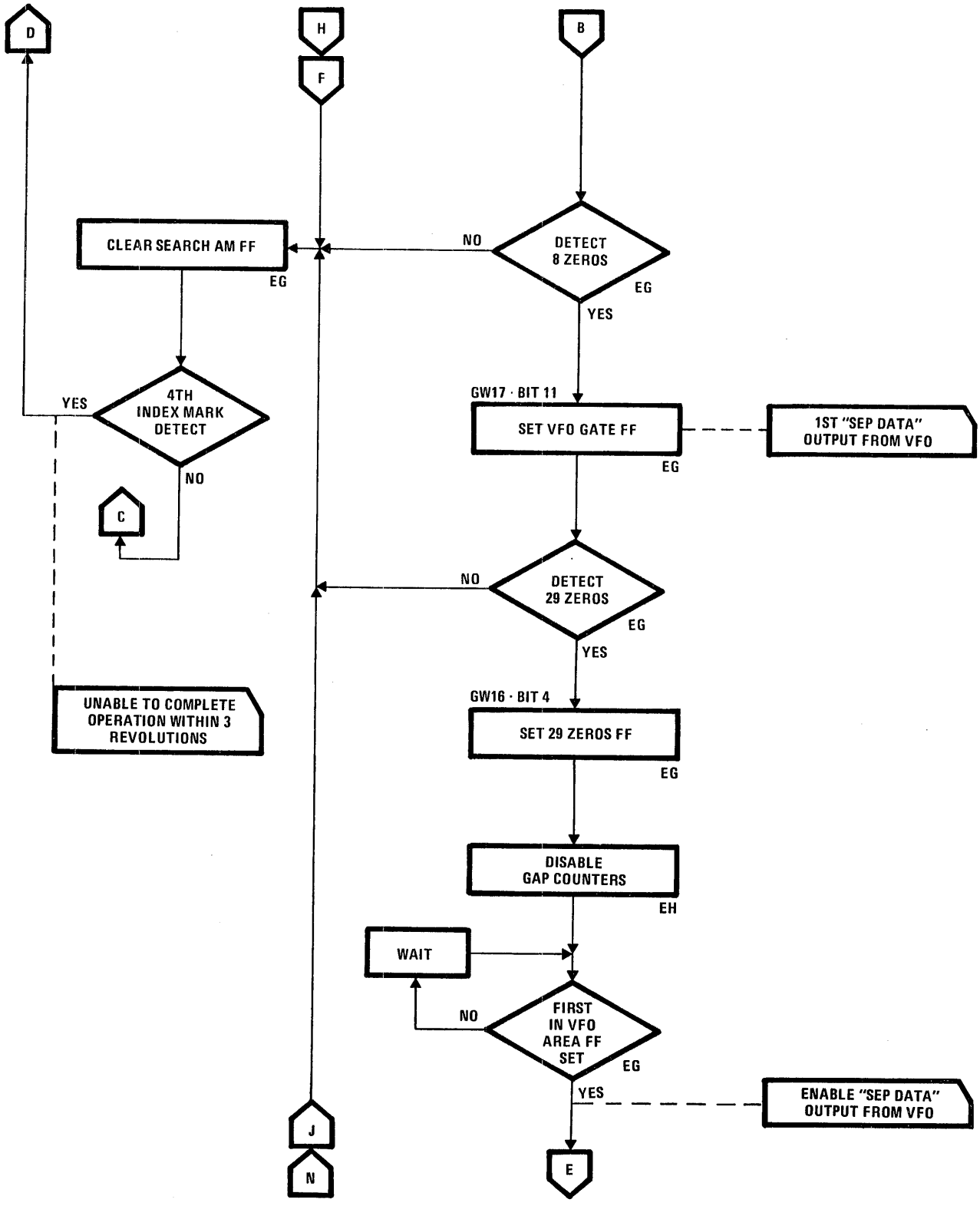


Figure 6-28. Search BRS (Continued)

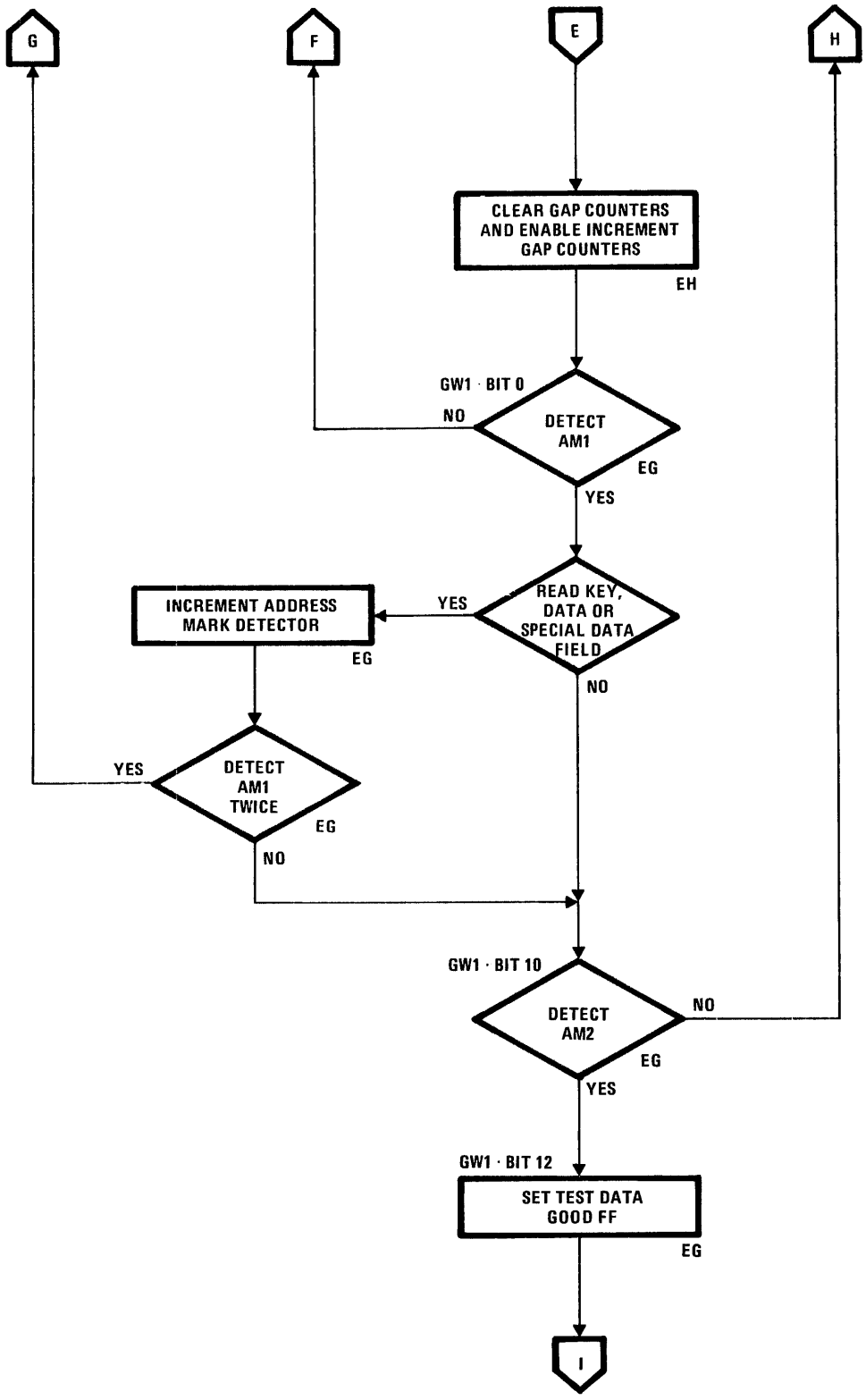


Figure 6-28. Search BRS (Continued)

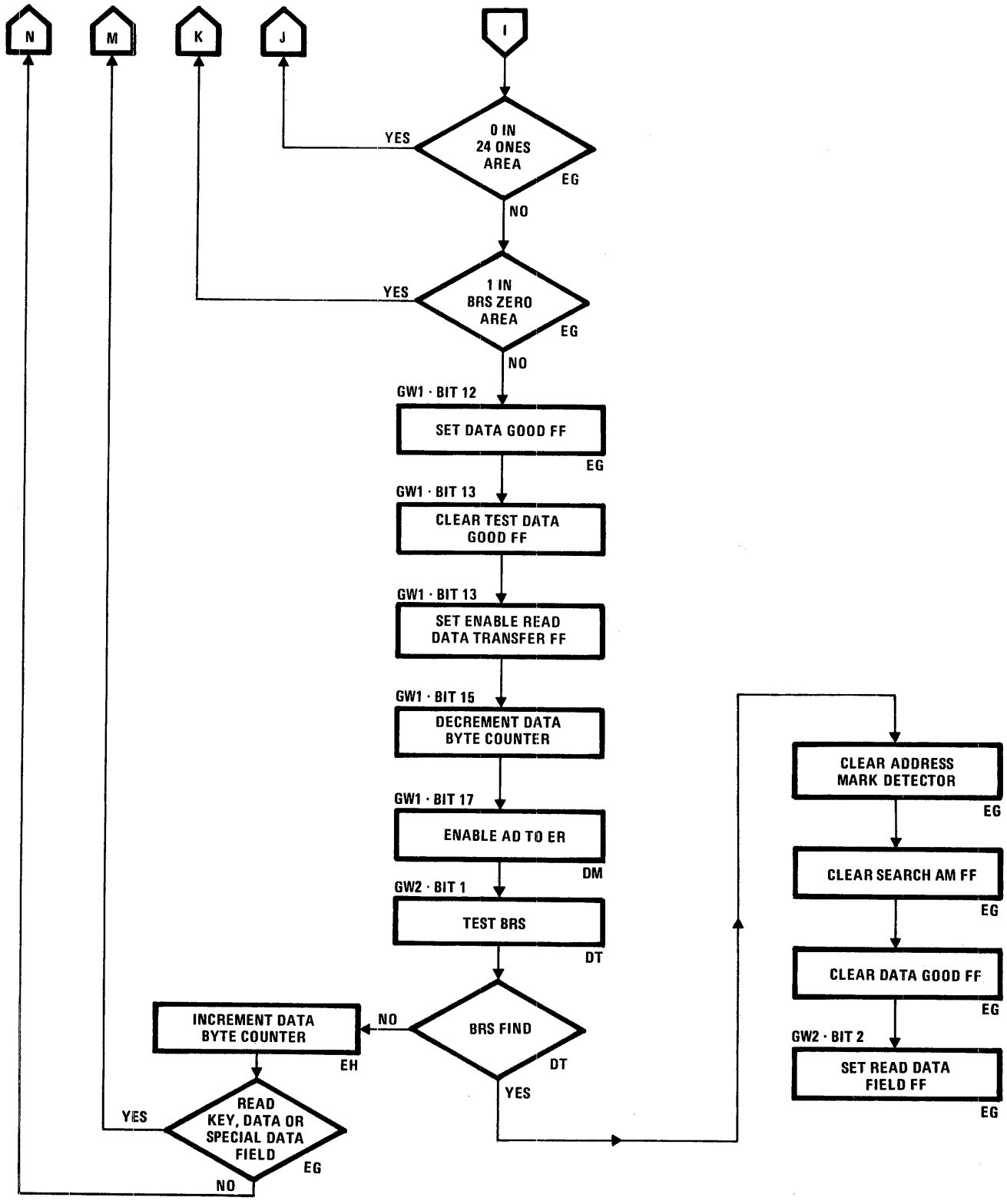


Figure 6-28. Search BRS (Continued)

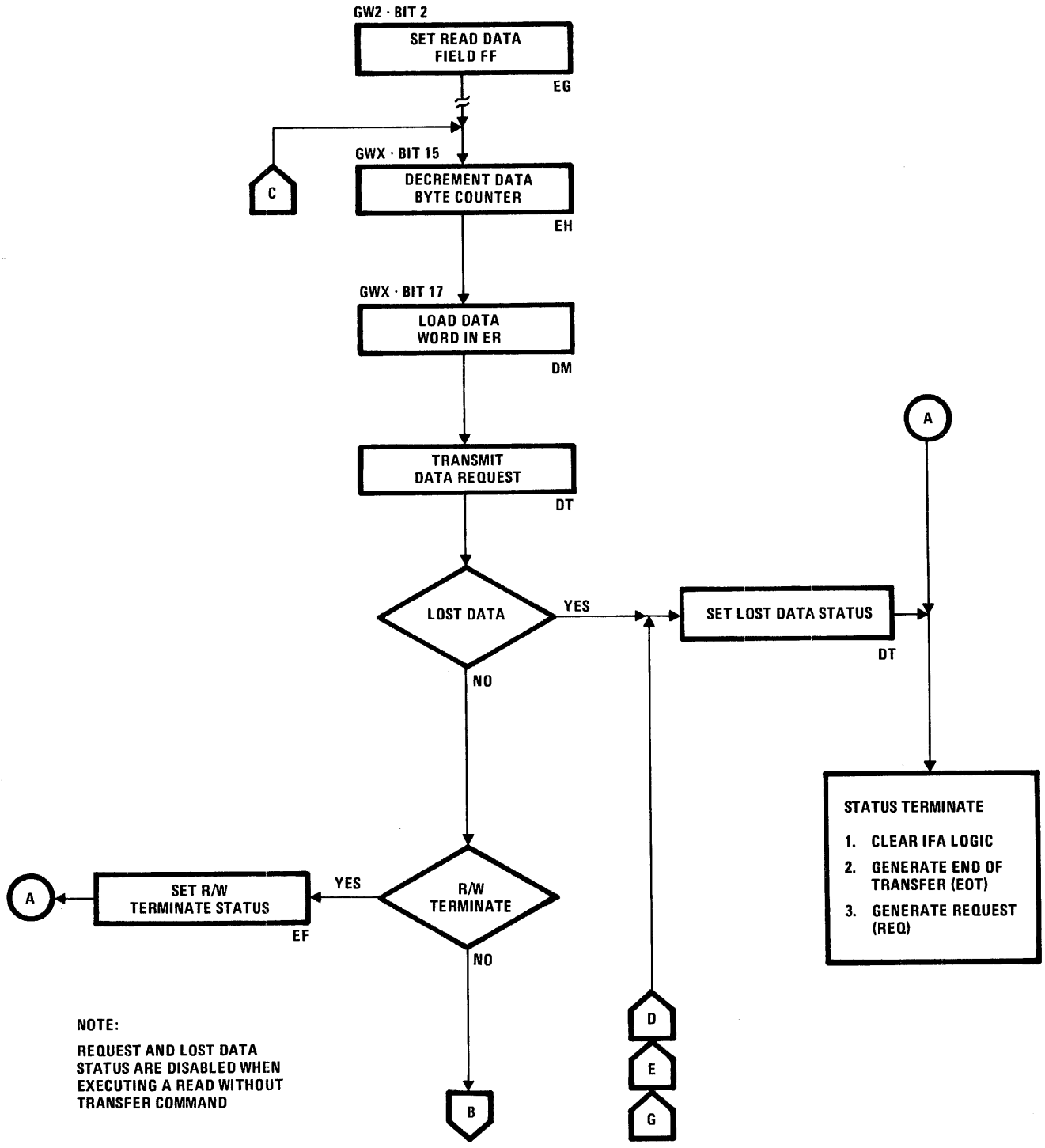


Figure 6-29. Read Data Transfer

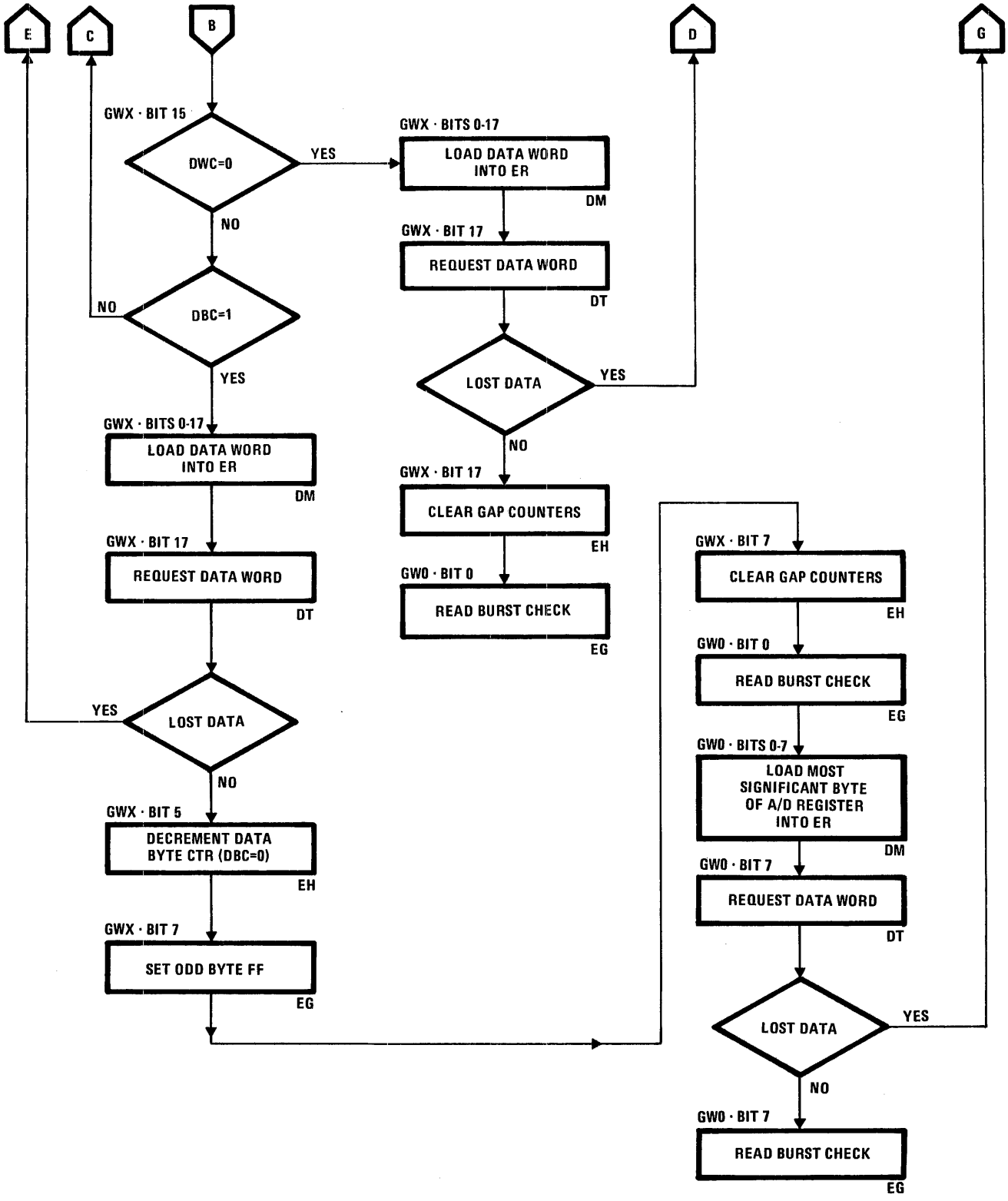


Figure 6-29. Read Data Transfer (Continued)

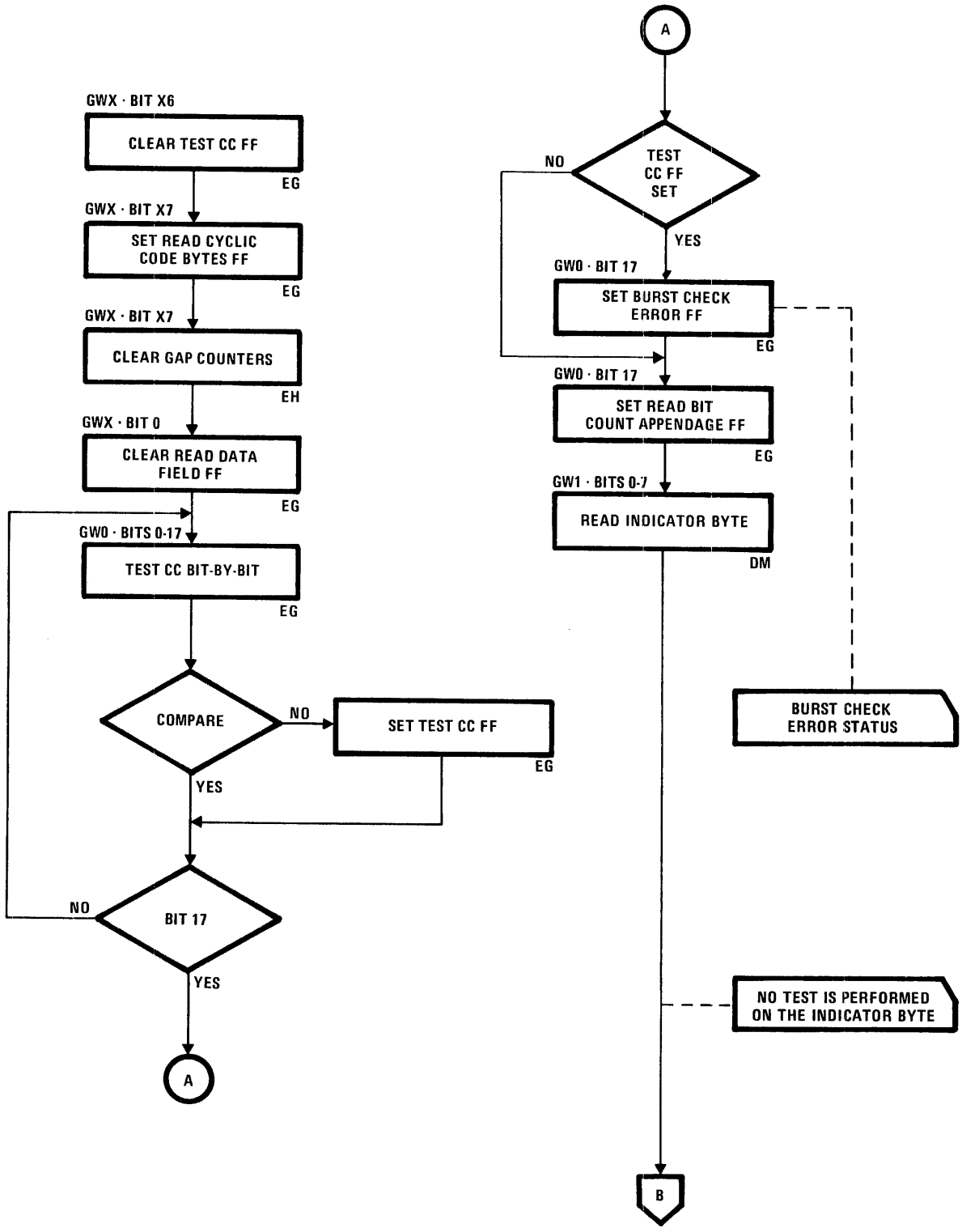


Figure 6-30. Read Burst Check

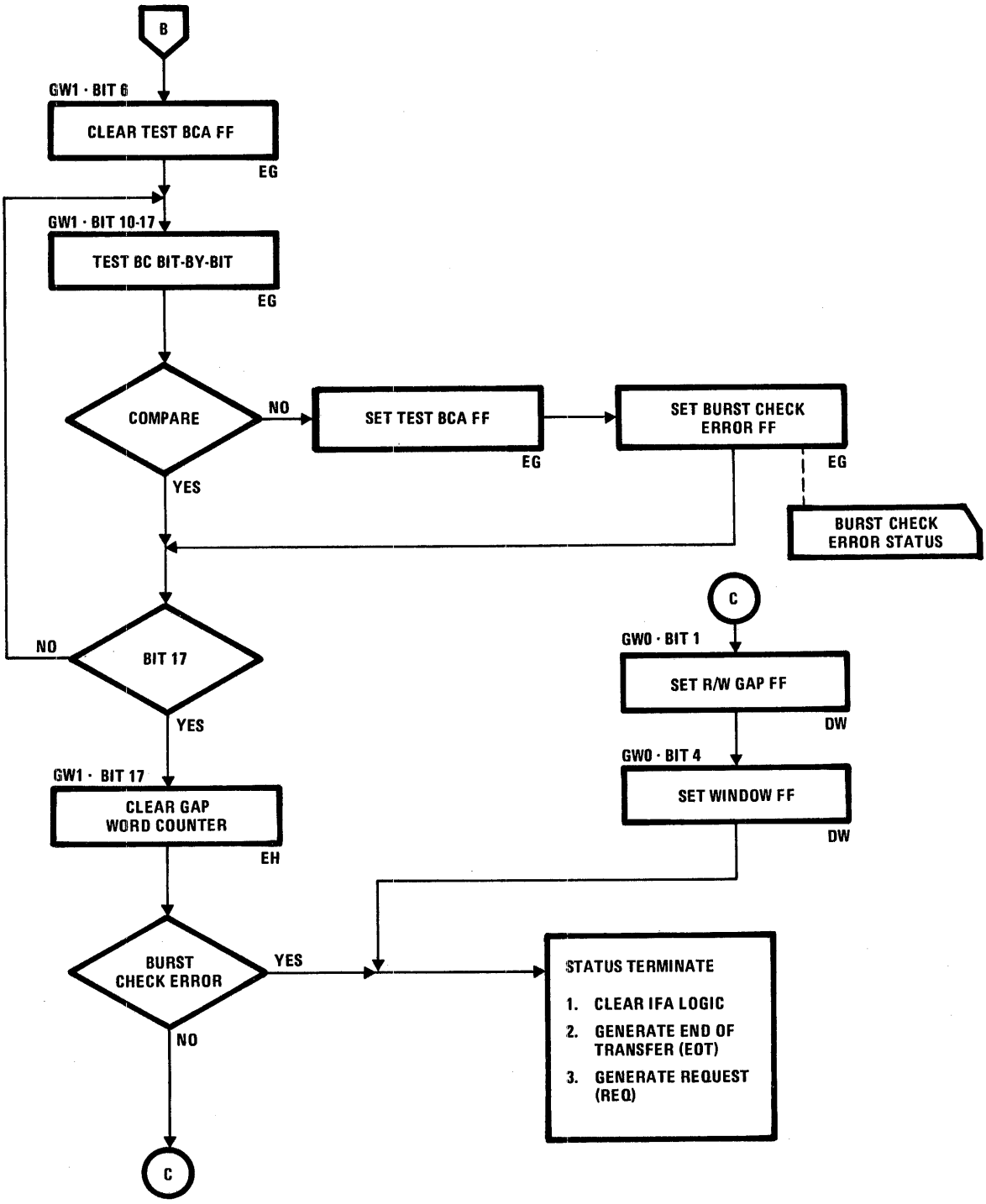


Figure 6-30. Read Burst Check (Continued)

WRITE SEQUENCE

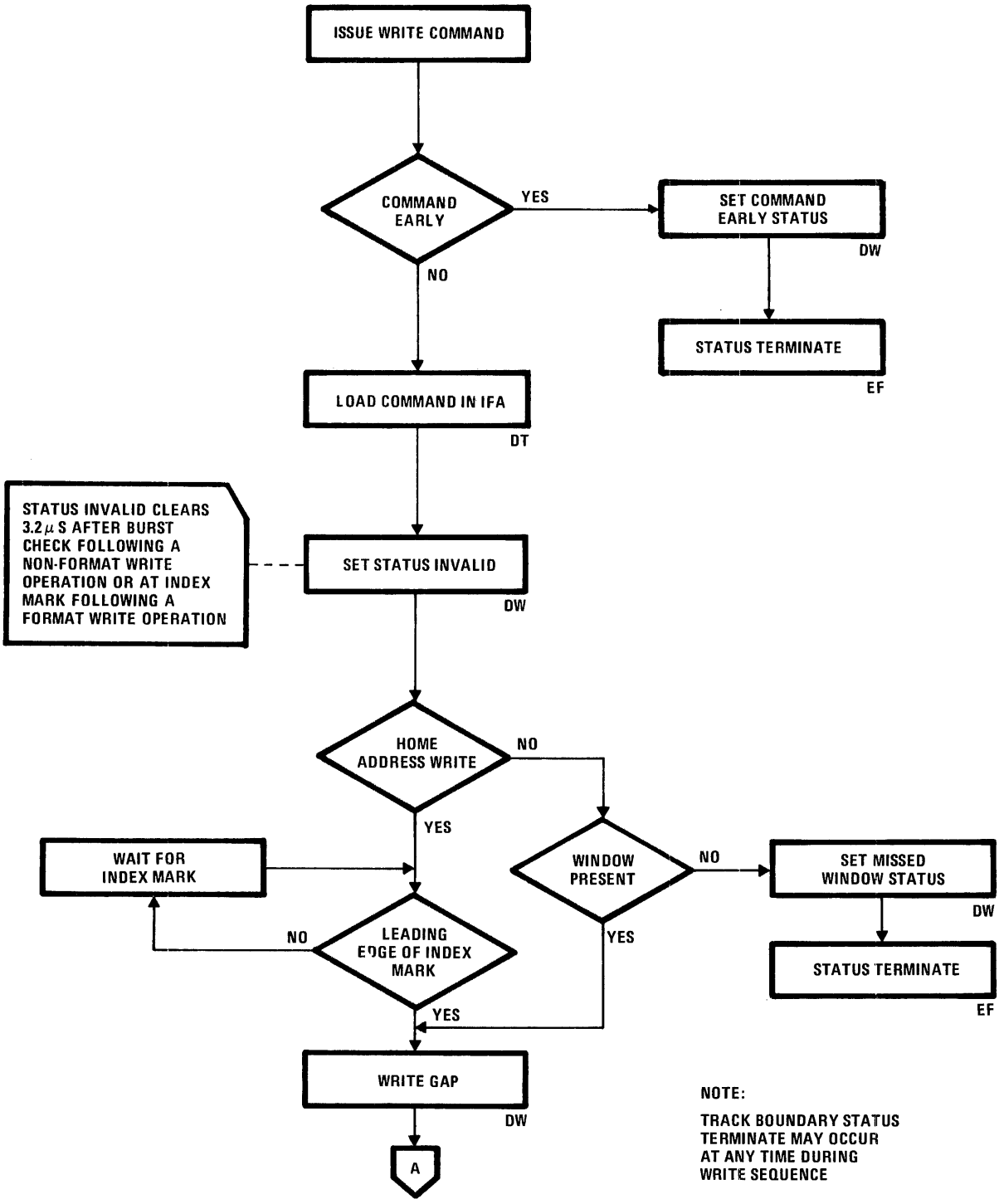


Figure 6-31. Write Sequence

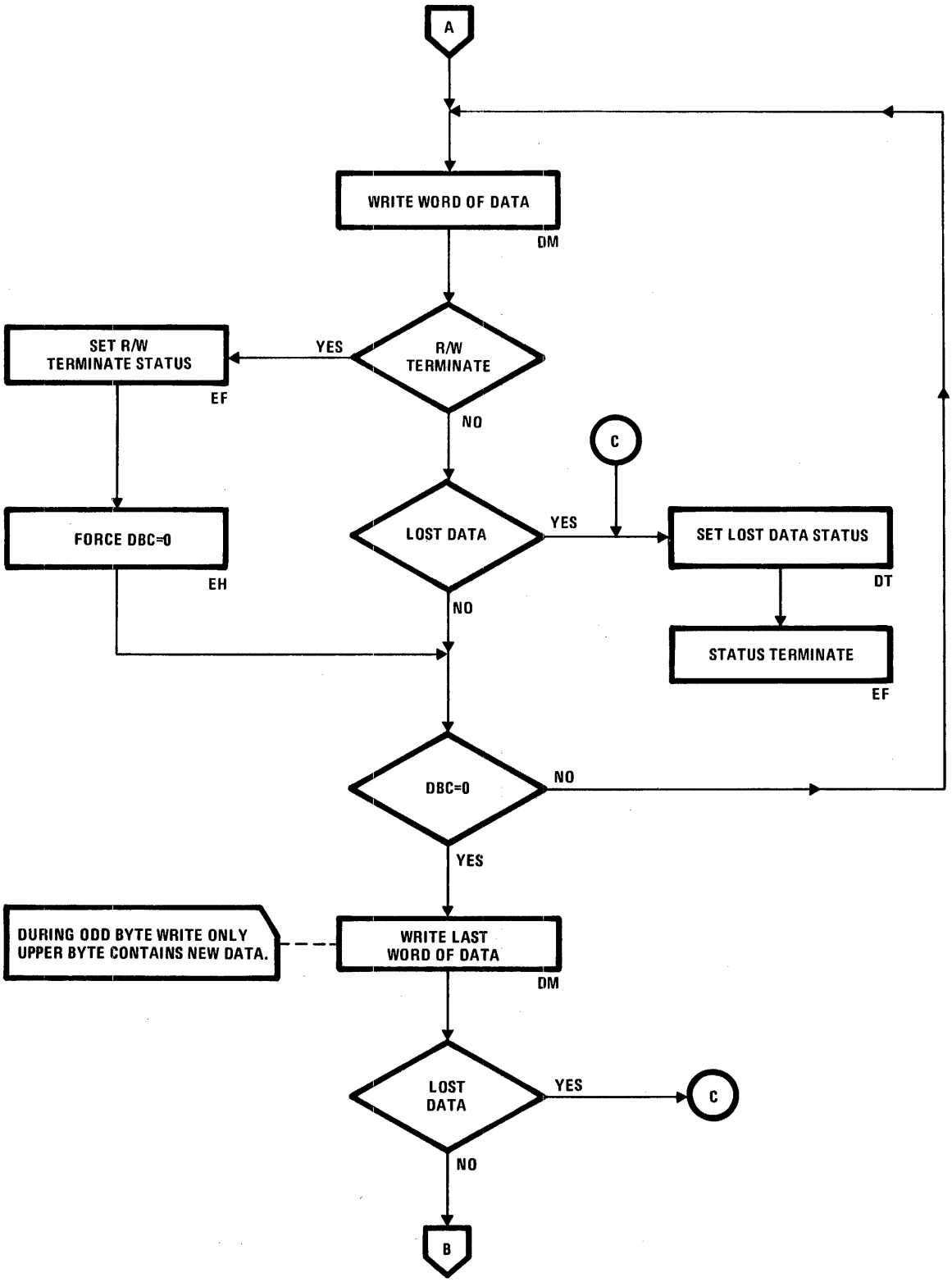


Figure 6-31. Write Sequence (Continued)

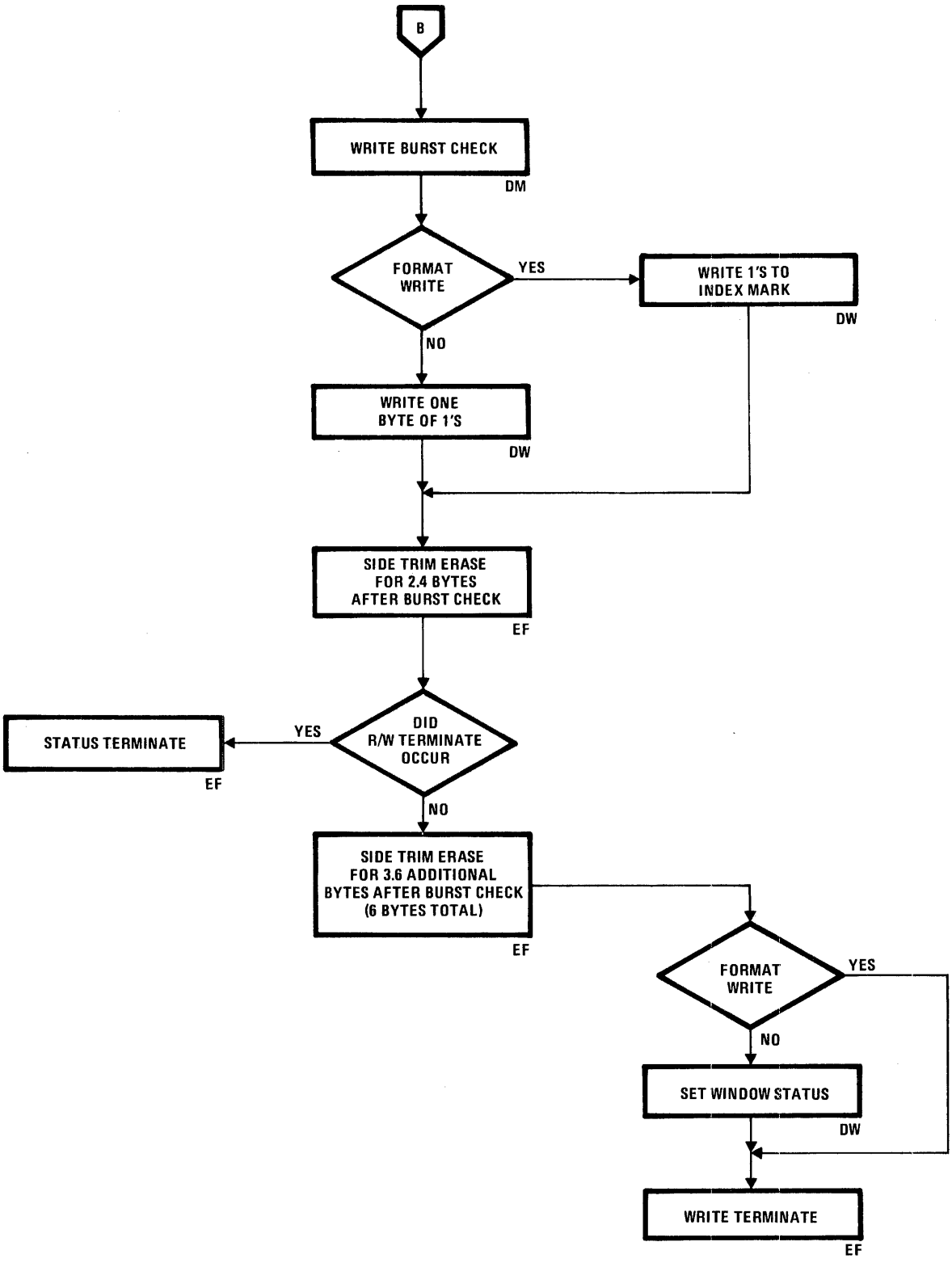


Figure 6-31. Write Sequence (Continued)

LOAD WRITE COMMAND

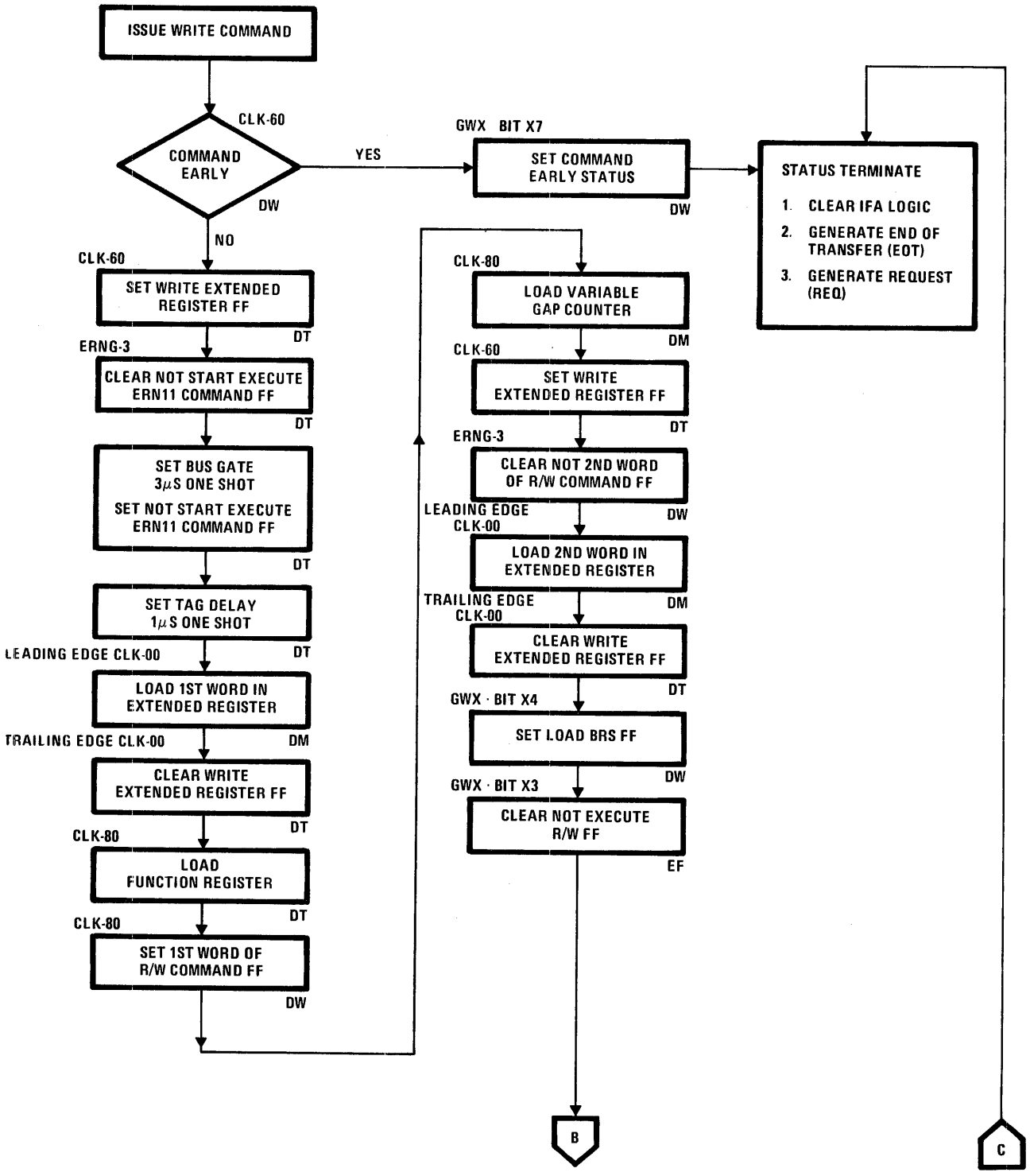


Figure 6-32. Load Write Command

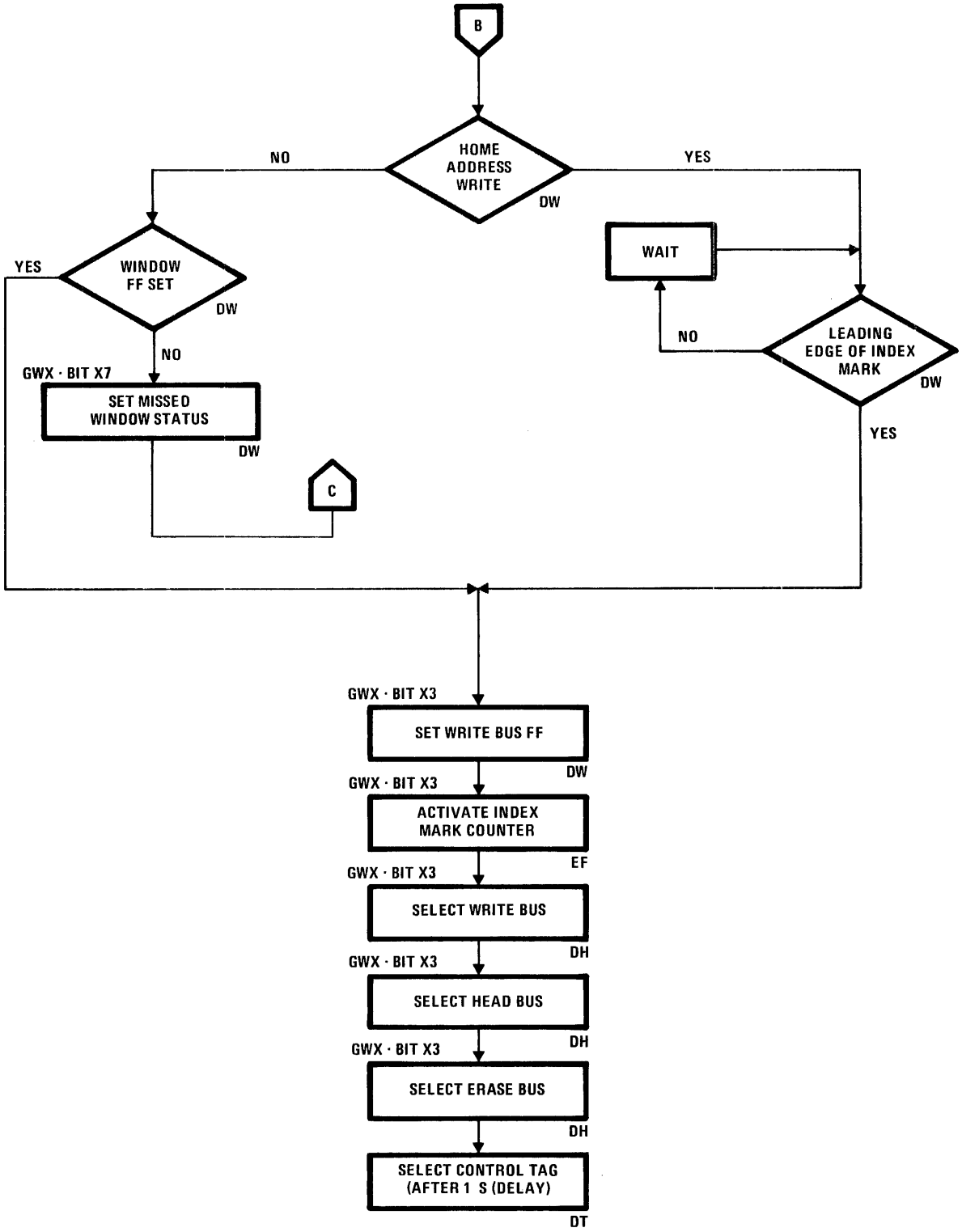


Figure 6-32. Load Write Command (Continued)

WRITE GAP

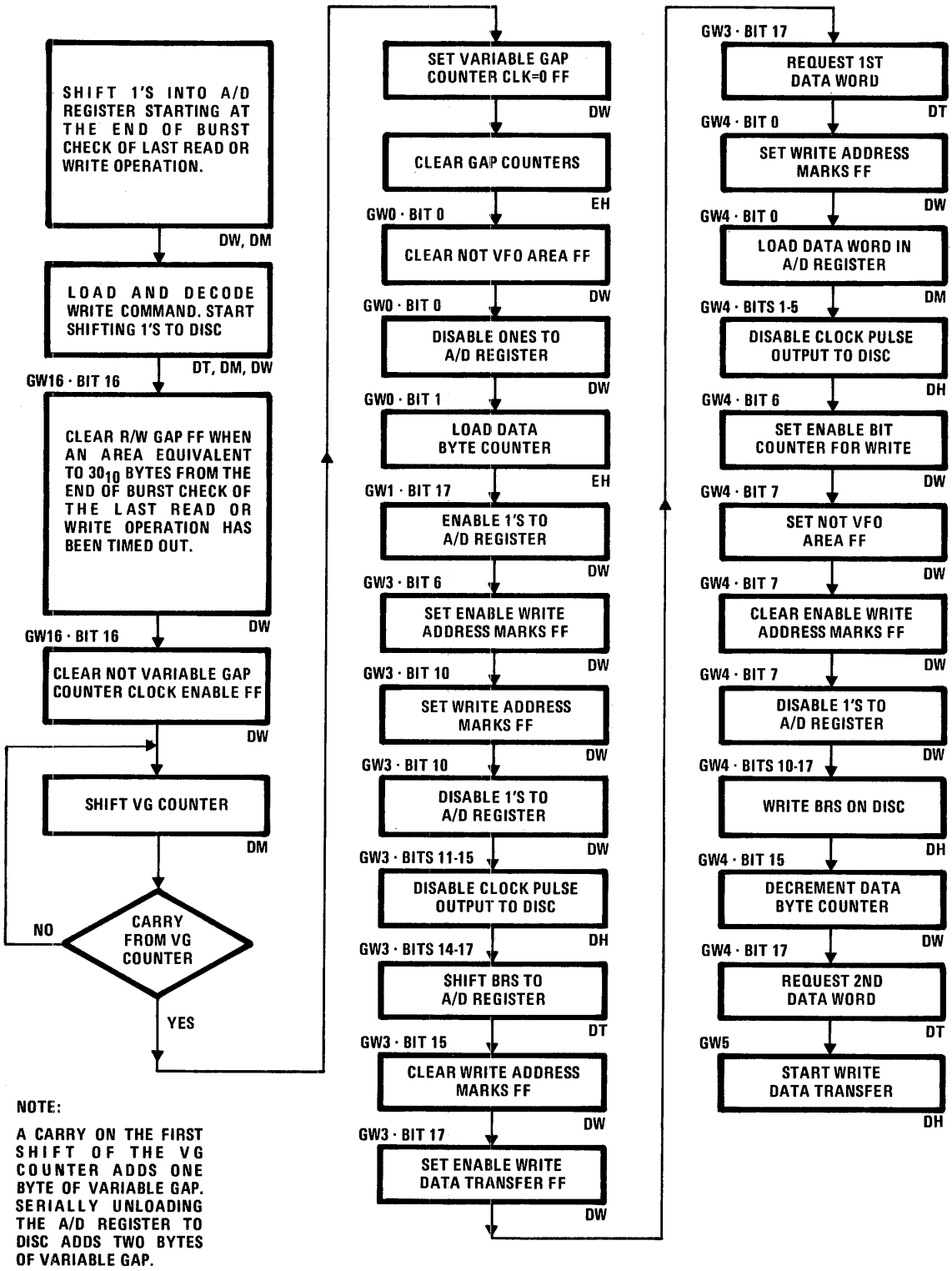


Figure 6-33. Write Gap

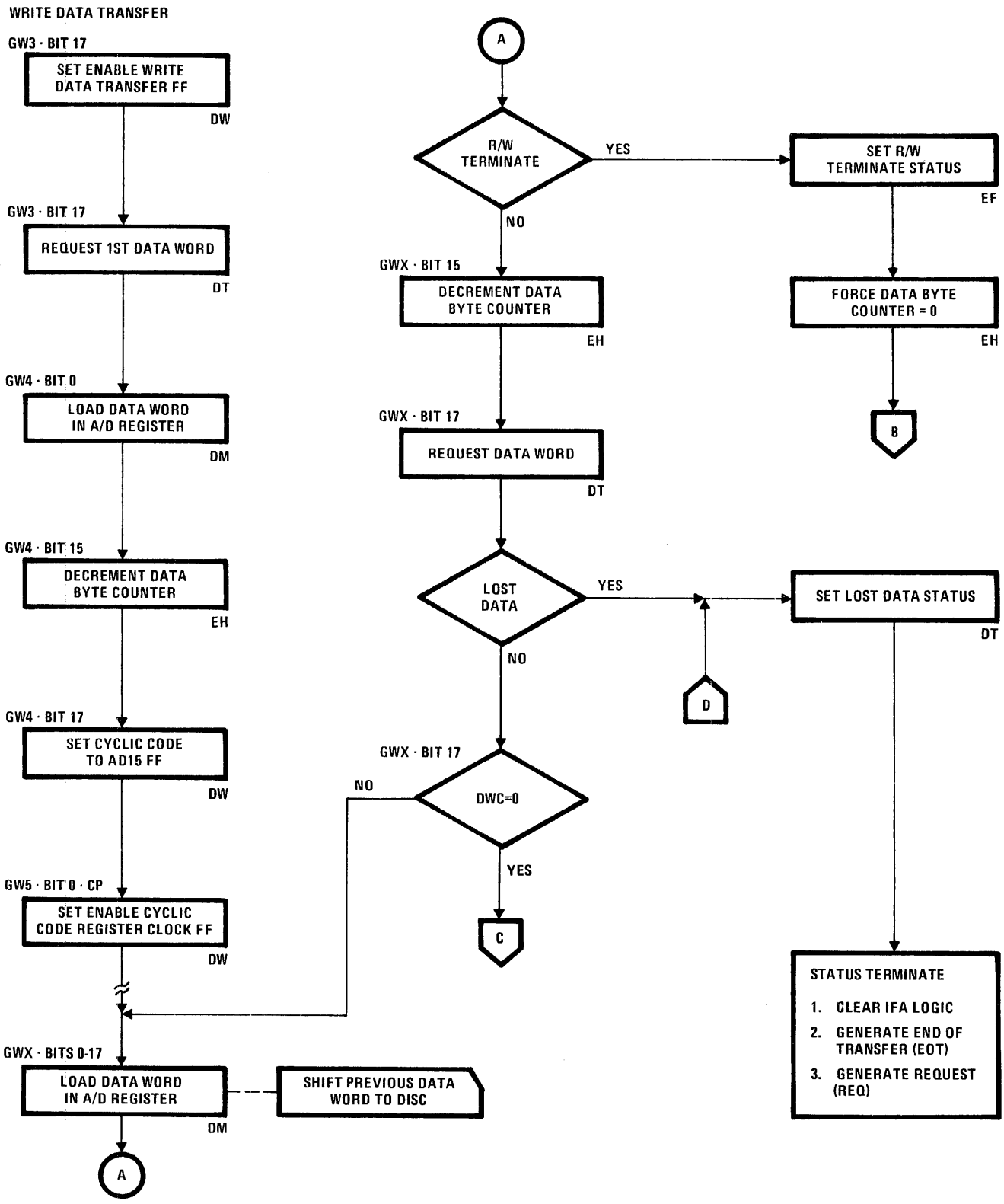


Figure 6-34. Write Data Transfer

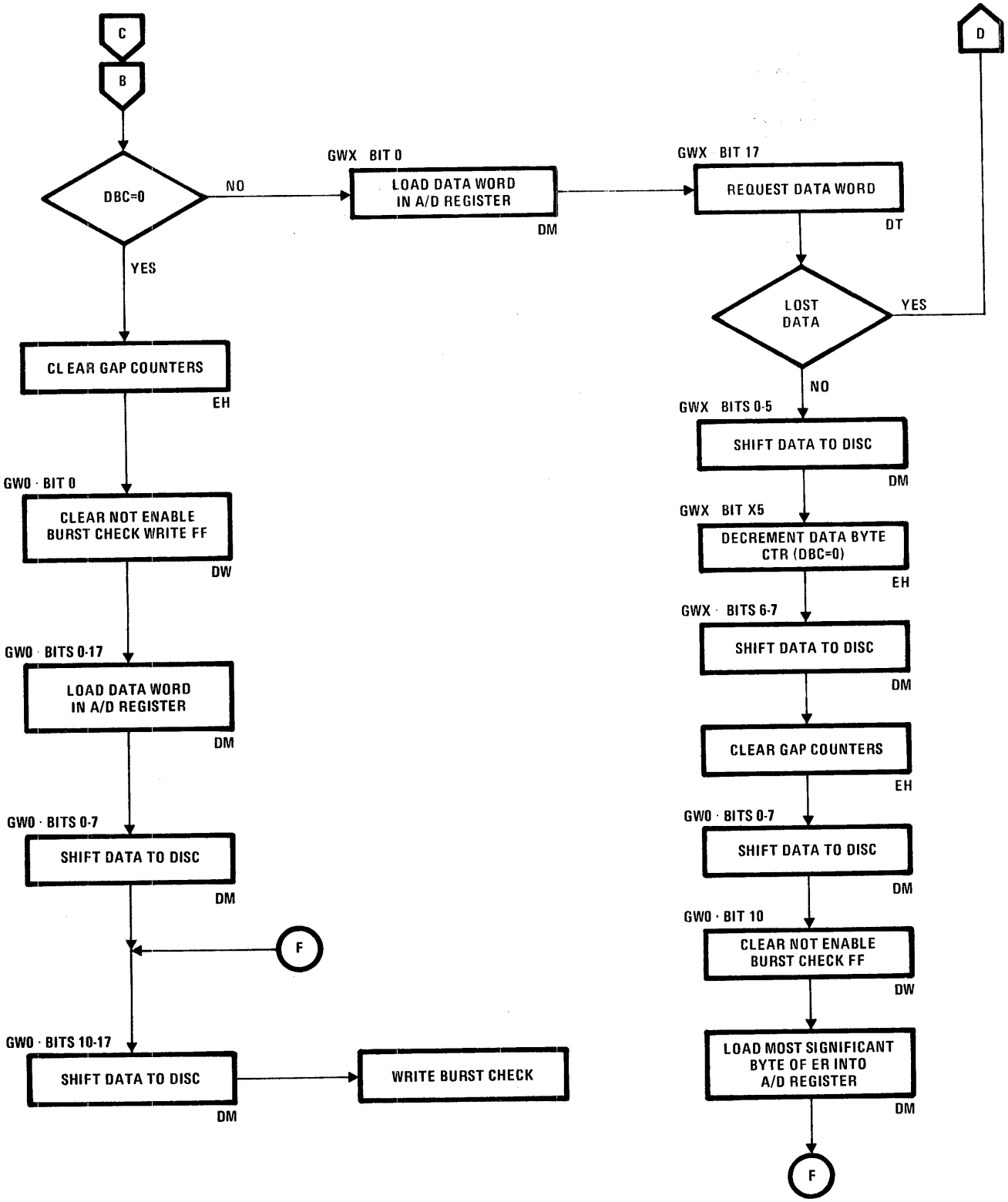


Figure 6-34. Write Data Transfer (Continued)

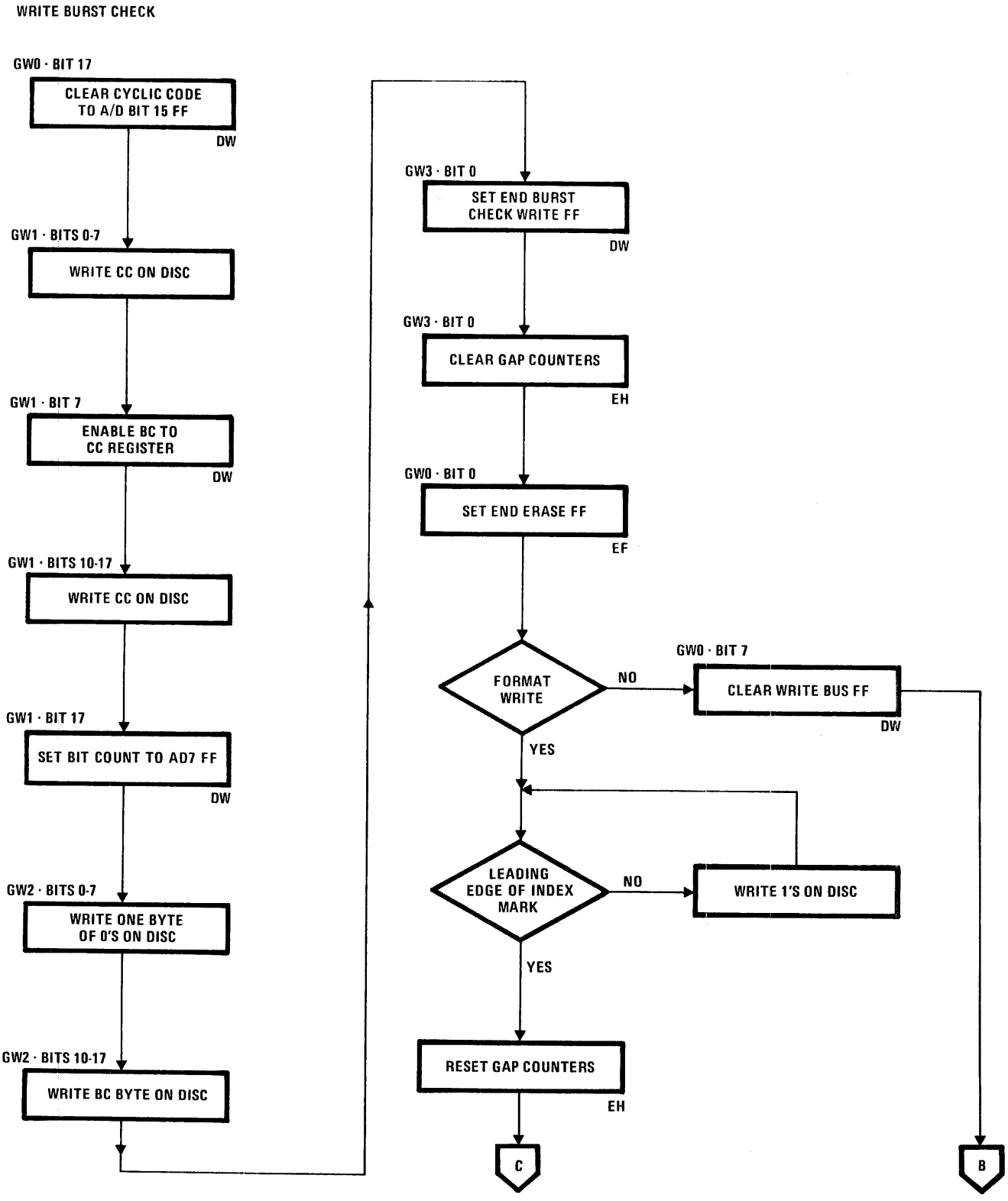


Figure 6-35. Write Burst Check

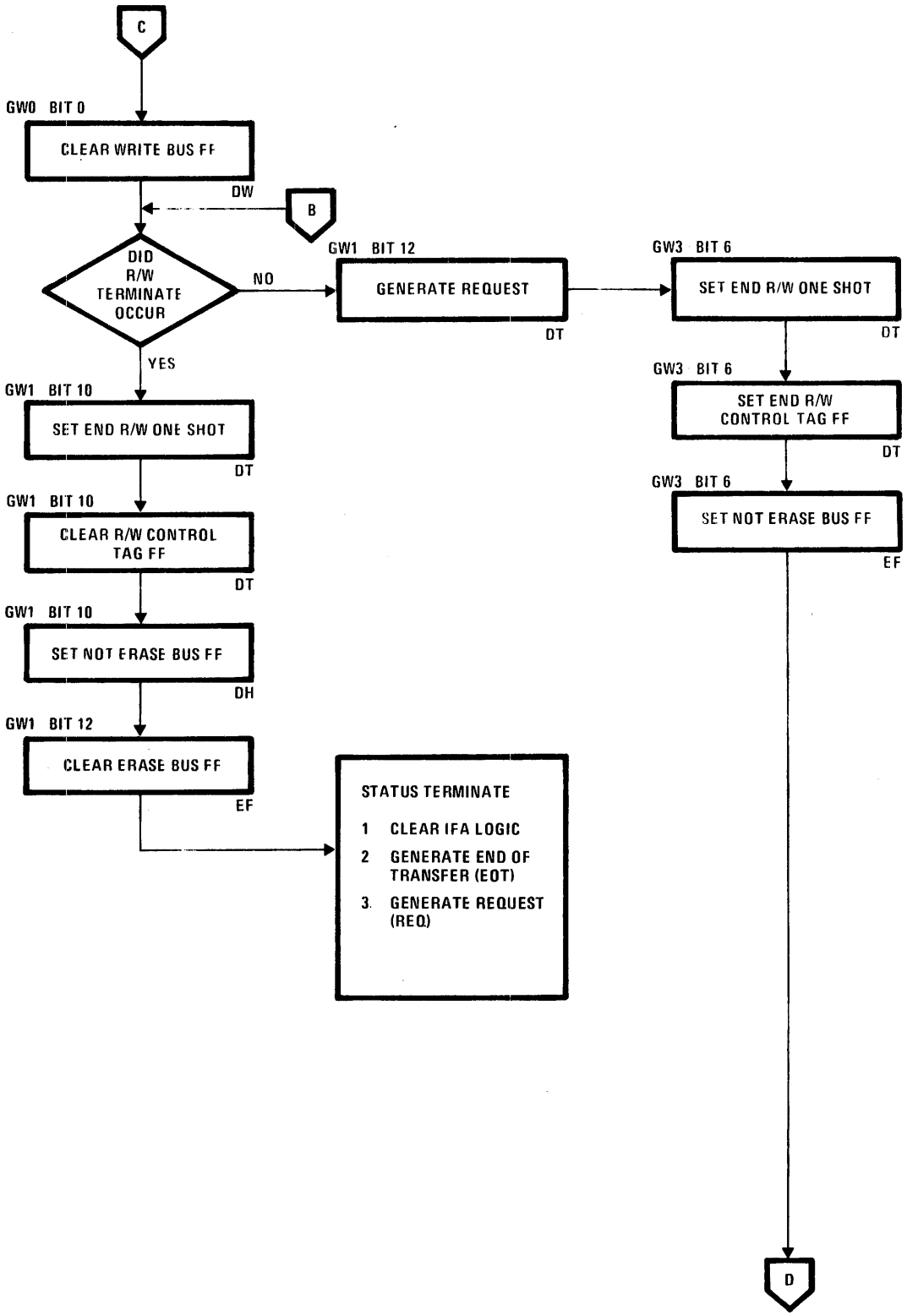


Figure 6-35. Write Burst Check (Continued)

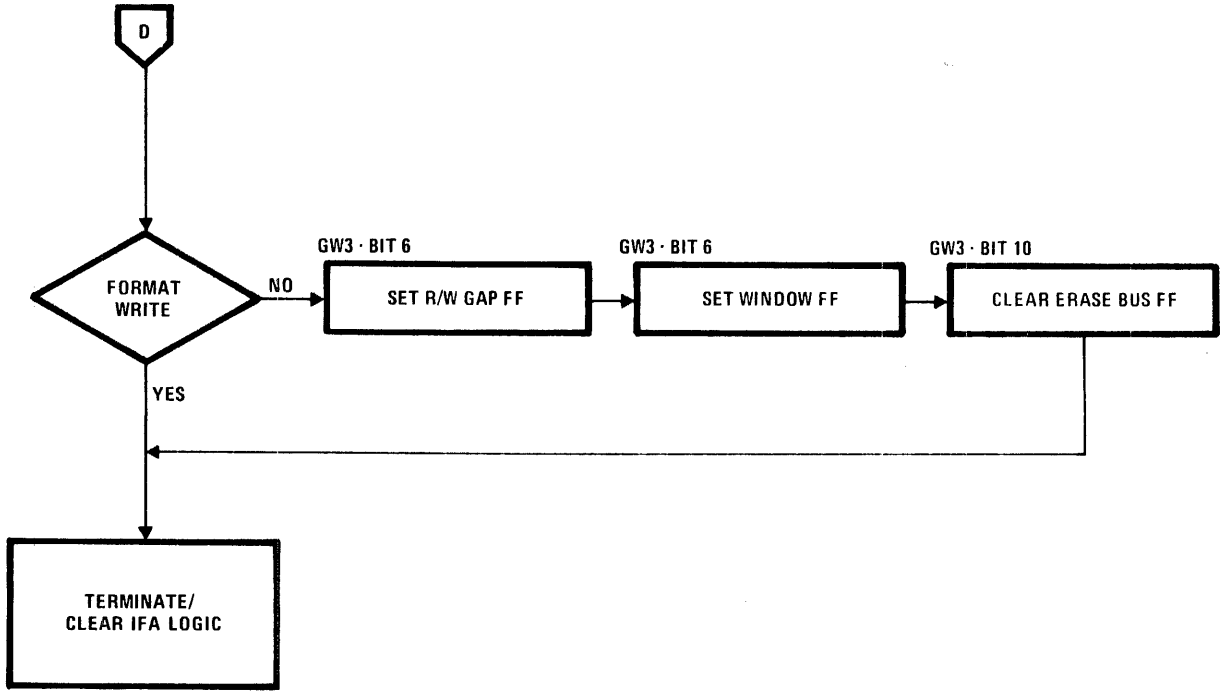


Figure 6-35. Write Burst Check (Continued)

DATA DETECTION

During *Read* operations, the variable frequency oscillator (VFO) is *phase-locked* to the returning data enabling the IFA to distinguish data fields from gap information. Additionally, the VFO provides synchronized clock pulses for accurate gating of detected data into the Assembly/Disassembly (A/D) register.

The VFO generates four signals used by read control logic for address mark (AM) detection:

- ZEROS COUNT provides a pulse for every 0 received from the disc.
- ONES RESET provides a constant high output while reading an area containing 1's.
- DATA GAP SENSE detects 3 or more consecutive missing separated data pulses.
- CLOCK GAP SENSE detects 3 or more consecutive missing clock pulses.

The VFO consists of nine separate circuits each responsible for a specific aspect of AM detection or data synchronization as shown in Figure 6-36.

The VFO circuit components are discussed in the following paragraphs.

Ramp Generator

The Ramp generator is a relaxation sawtooth oscillator consisting of a capacitor and three current sources. A discharge circuit operates when the ramp wave reaches a fixed reference level. (One current source adjusts the range of the VFO and is factory sealed.) A switched current source is used to change the period of the oscillator waveform on alternate cycles. A variable current source, driven by the error detection circuit, supplies the feedback.

Error Detector

A diode bridge sampling gate compares the phase of the ramp waveform to the time position of any incoming pulse, and generates an error signal proportional to the phase error.

Filter

An RC filter is designed to smooth the error pulses from the error detector for application to the error current amplifier.

60/40 Trigger

The negative peak of each ramp cycle complements a JK trigger. The output of this trigger is used to drive the switched current source described under "Ramp Generator" as well as supplying the gating needed for detecting data and clock pulses. As a result of the operation of this trigger, the time period of two successive ramp cycles is divided into a 60% cycle and a 40% cycle. This is done to provide a greater detecting tolerance for clock bits than for data bits.

Phase Detector and Trigger

The phase detector and trigger provides the clock pulses for gating and shifting the data in the A/D register. A trigger is complemented each time the ramp wave passes a fixed voltage reference. Phasing provides correct timing for the register.

Starting Logic

Correct phase starting of the ramp wave and all triggers is performed by the starting logic. Each time the VFO is used for reading, the starting logic first shuts off the 2F oscillator pulses used to maintain the VFO frequency within 4% of that needed. Then the ramp wave is clamped to a reference for at least one clock time and released in phase with the incoming data from a selected file. This is done to minimize the time required to synchronize the VFO to the incoming data.

Window Extender AND Gates

The output of the 60/40 trigger supplies the gating for the window extender. The window extender is used to extend the trailing edge of the detection gates to include the trailing edge of any clock or data pulse. This prevents a late pulse from being gated into the following data or clock cell.

Adjustable Delay Line

The leading edges of the data and clock pulses are centered within their respective windows by selecting a tap on the delay line at the factory. This permits the maximum permissible time shift allowed for data and clock pulses to be used without incurring an error. Component tolerances from one VFO card to the next may require the tap chosen to differ between any two cards.

Address Mark and Format Decode

Address mark and format decoding is accomplished by four separate circuits. The 0's Count Data Gap Sense and Clock Gap Sense are each retriggerable single shots. The timing of the 0's Count single shot is less than the period between two clock pulses but greater than the period between a clock and a data pulse. An output will be generated when there is 400 ns between pulses.

The two gap sensing circuits are timed to change level after three consecutive missing bits from their driving source. The 1's reset circuit is a JK trigger. The trigger is set by any pulse occurring while the 0's Count Single Shot is set. It is cleared by the use of 0's Count.

READ/WRITE TIMING

The IFA clock is used during read/write operations to:

- Generate clock pulses for use in NRZ double frequency recording
- Advance the Gap Word and Gap Field Bit counters to sequence basic read/write control functions

The IFA clock on the DH board provides the basic timing pulses for all data transfer operations. During *Write* operations, the IFA clock records a 50 nanosecond clock pulse on the disc surface every 400 nanoseconds in synchronization with the recording of data pulses.

The IFA clock also generates internal CP and DP timing pulses, corresponding to the clock and data pulses recorded on the disc surface. These are used to update the Gap Word and Gap Field Bit counters. Whenever the IFA is not performing a *Read* operation, CP and DP pulses are derived from the shared resources clock. During *Read* operations, CP and DP pulses are derived from the clock and data pulses detected by the VFO.

Gap Word Counter and Gap Field Bit Counters

These free-running octal counters are used to sequence the execution of read/write commands. The Gap Field Bit Counter is updated every 400 nanoseconds and this counter in turn updates the Gap Word counter every 6.4 microseconds. Synchronization is achieved by clearing the counters to zero at various points in the operation.

When performing a *Write* operation (Figures 6-37 and 6-38), the Gap Word and Gap Field Bit Counters are cleared to zero at the following times:

- at the end of the variable portion of the gap area (beginning of *Write Zeros*)
- at the end of the data field
- at the end of burst check

The Gap Word and Gap Field Bit counters are cleared to zero when a minimum of 33 bytes of gap space have been written (additional variable gap bytes can be written under program control). This portion of the gap provides data separation and allows sufficient time (window) to decode a *Write Count, Key* or *Data Field* command. The Gap counters may be cleared to zero either in the middle of a gap word or at the end of a gap word depending on whether the variable portion of the gap area contained an odd or even number of bytes.

The Gap Word and Gap Field Bit counters are cleared a second time at the end of the data to sequence the writing of burst check information. This may occur either in the middle of a gap word or at the end of a gap word depending on whether the information field contained an odd or even number of bytes (Figure 6-39).

The Gap Word and Gap Field Bit counters are cleared a third time at the end of burst check. Clearing the counters at this time activates the terminate logic. For normal *Write* operations (not *Format Write*), terminate logic causes one byte of 1's to be written after burst check During *Format Write* operations, terminate logic causes delete 1's to be written on the track until the index mark is detected. Terminate logic also causes the selected drive to side trim erase for a minimum of 6 bytes after the end of burst check (provided R/W Terminate did not occur) or until index mark is detected. At the conclusion of Side Trim erase for a normal write operation, terminate logic produces a "Window" for the next operation.

During a *Read* operation (Figure 6-40) the Gap Word and Gap Field Bit Counters are cleared to zero at the following times.

- at the end of the variable portion of the gap area (detect 0's area)
- when the first "1" is detected in the VFO portion of the gap area (separated data)
- at the end of the data in each field
- at the end of the burst check (Gap Word Counter only)

During a *Read* operation the variable gap area is required to provide sufficient time (window) to decode a Key or data field command and to activate the VFO. When reading the four bytes of 0's which follow the variable area, the Gap Word counter is decremented end-around through a count of 17 to a count of 16. The Gap Field Bit Counter is then set to zero for the first bit and decremented end-around through 17 to a count of 1 for each word. At a count of 8 zeros (GW17 Bit 11) the VFO Gate is activated allowing separated data output from the VFO. A count of 29 zeros (GW16·Bit 4) disables both counters until the first separated data pulse is transmitted from the VFO.

The first 1 bit transmitted by the VFO after a field of 32 0's clears both gap counters. This allows the counters to increment in the forward direction until the Data Byte Counter reaches zero at the end of the data transfer.

The gap counters are cleared at the end of the data transfer and used to sequence the re-computation and checking of burst check information. This may occur either in the middle of a gap word or at the end of a gap word depending on whether the information field contained an odd or even number of bytes (Figure 6-41).

The Gap Word Counter (but not the Gap Field Bit counter) is cleared at the end of burst check to terminate the *Read* operation. This activates the status terminate logic. Status Terminate sets the Window FF in preparation for the next data transfer command.

Both counters are cleared to zero when an index mark is detected to prevent attempting to read or write beyond the end of the track

The window, during which a *Write Count, Key* or *Data Field* command or a *Read Key* or *Data Field* command may be translated, is dependent on the previous *Read* or *Write* operation (Figure 6-42). If the previous operation was a *Read* command, the window extends from bit 4 of the 1st byte after burst check to bit 0 of the 25th byte after burst check. If the previous operation was a *Write* operation, the window extends from bit 6 of the 7th byte after burst check to bit 0 of the 25th byte after burst check.

DATA FORMAT LOGIC

Data formatting is performed by the Assembler/Disassembler (A/D) register. The A/D register performs the following functions.

- changes parallel-by-word data into serial-by-byte data when writing on a disc surface
- changes serial-by-bit data into parallel-by-word data when reading from a disc surface.

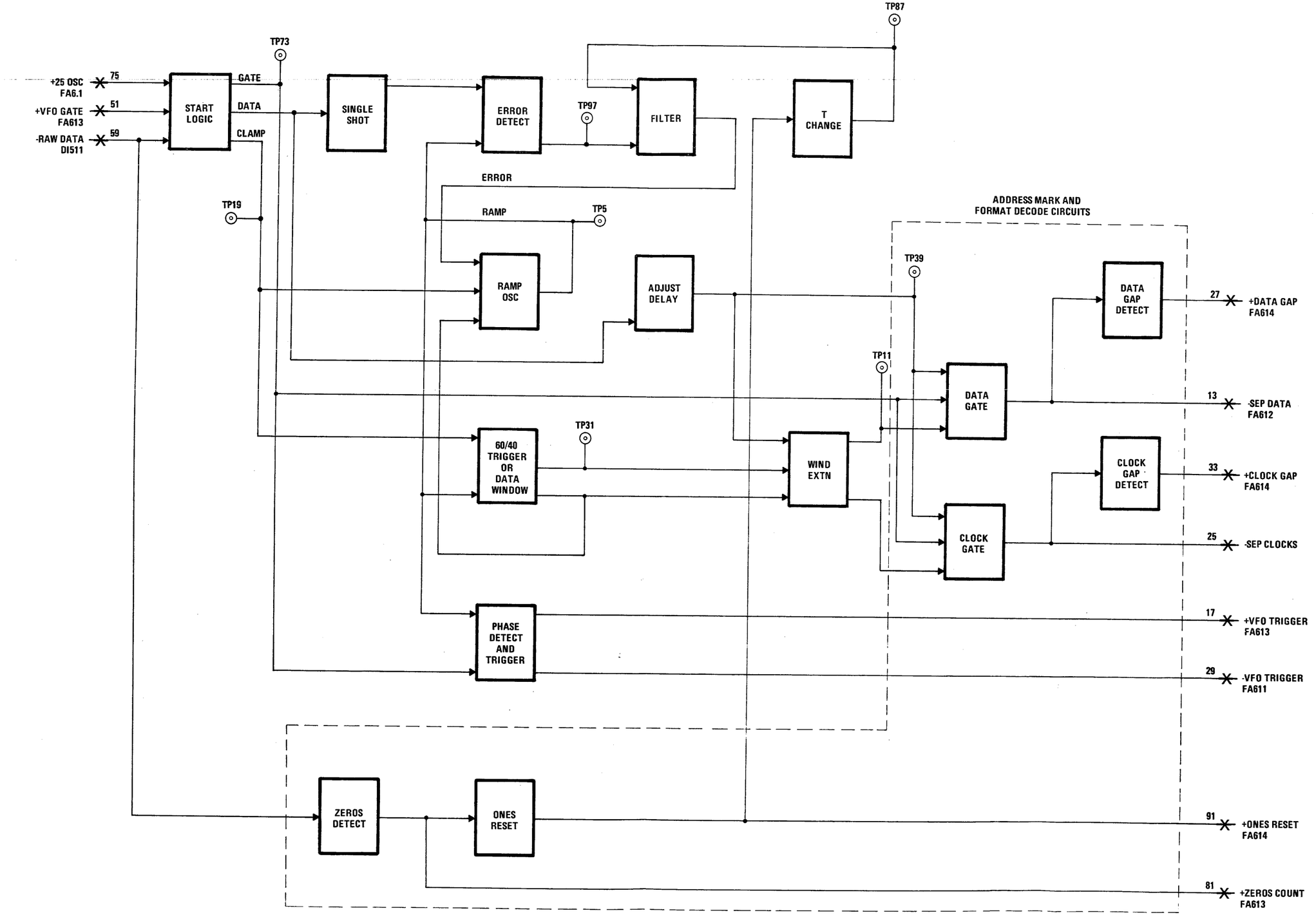


Figure 6-36. VFO Block Diagram

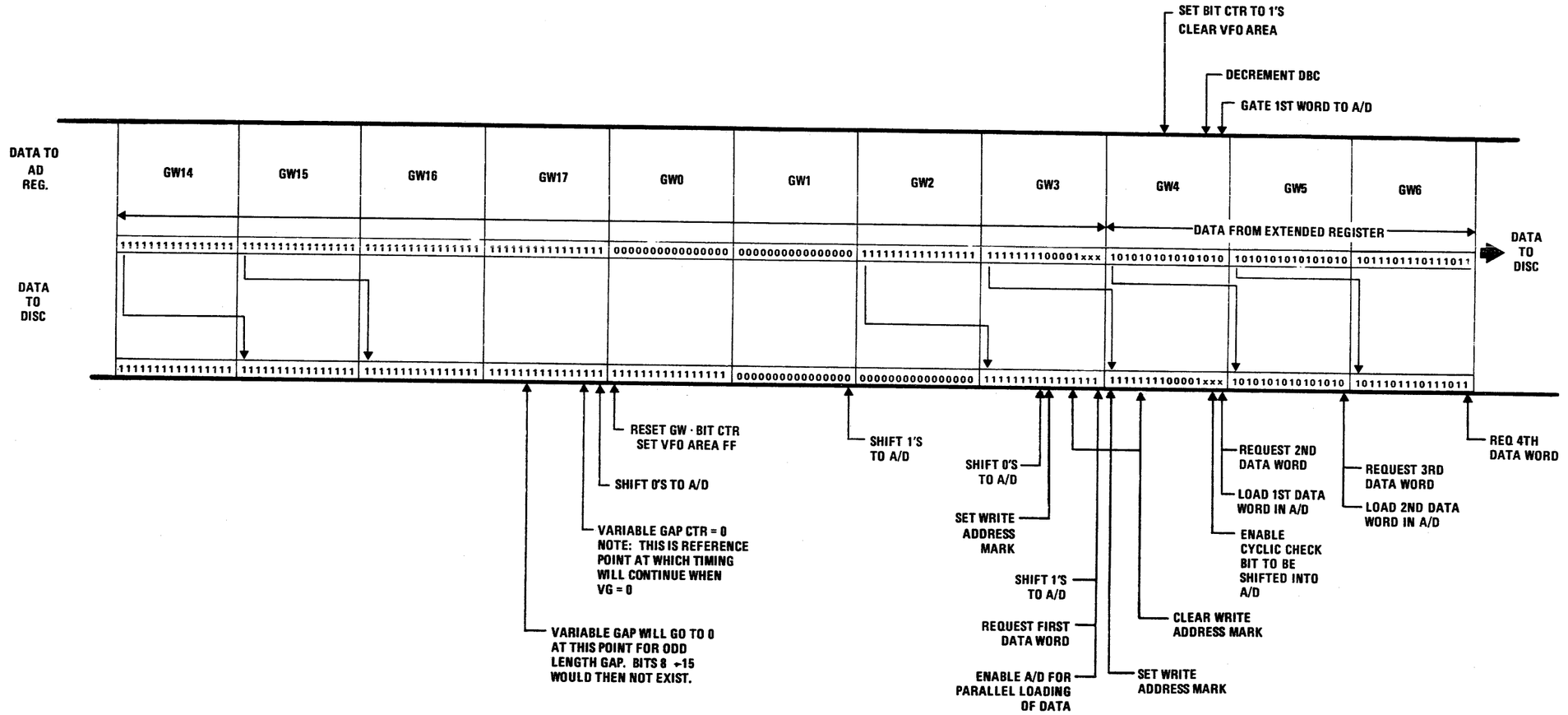


Figure 6-37. Write Timing 1 (Start)

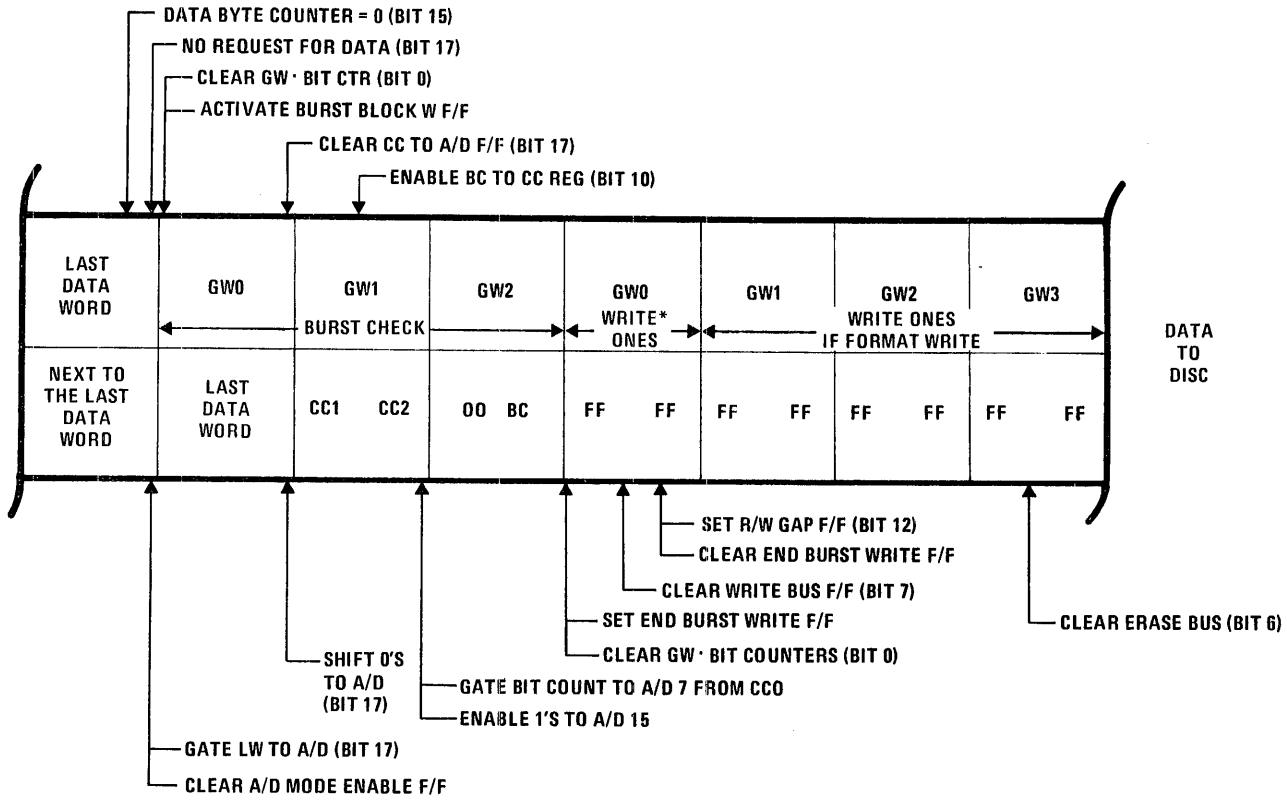


Figure 6-38. Write Timing 2 (End)

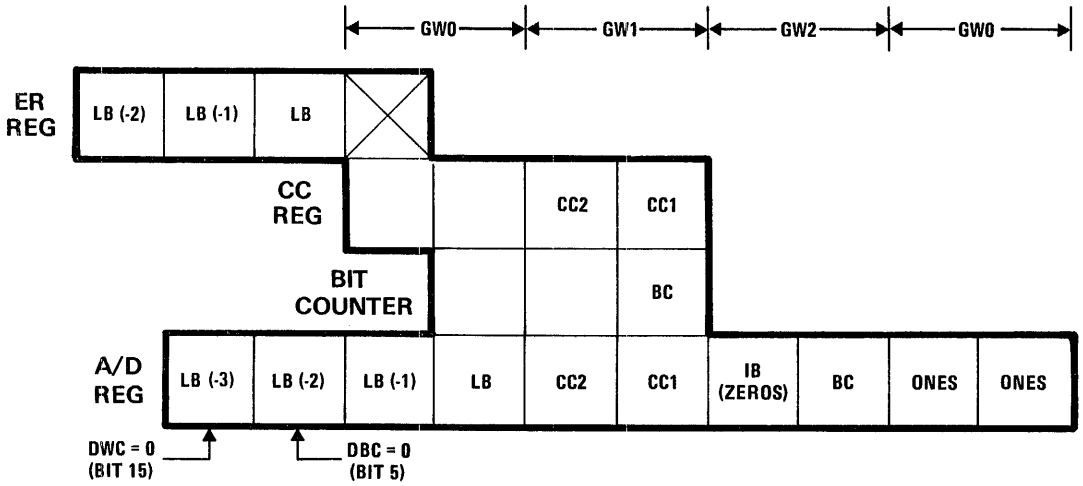
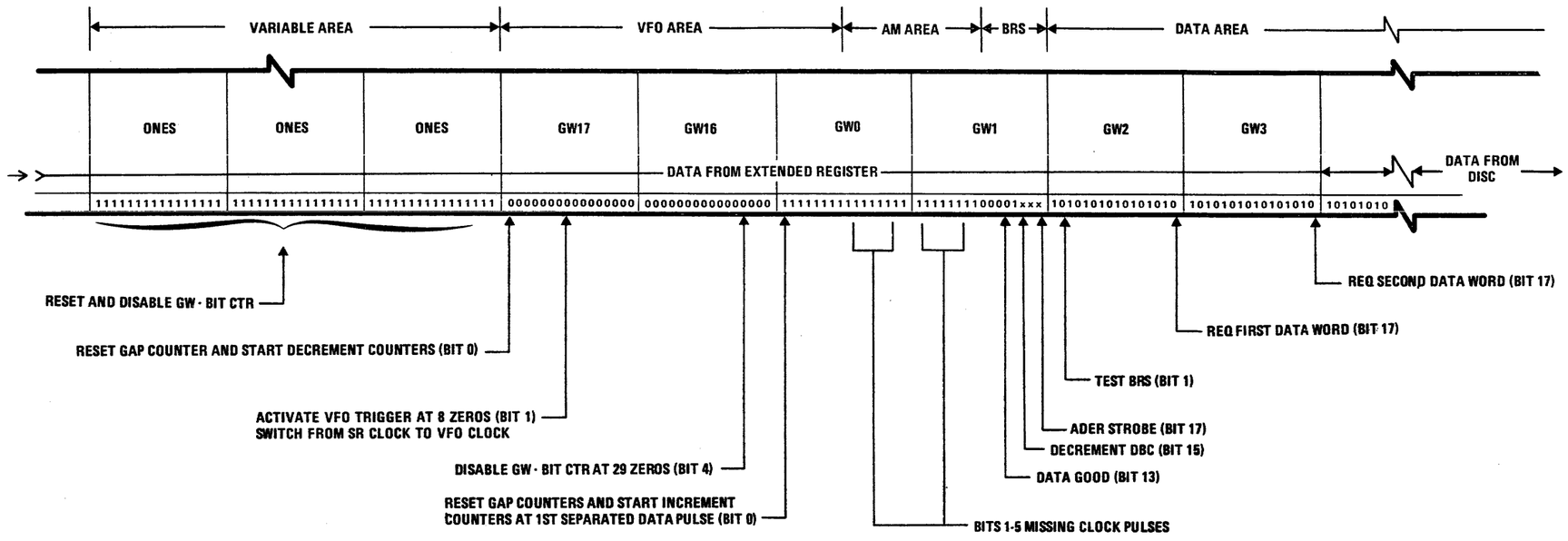


Figure 6-39. Odd Byte Write Timing



VARIABLE AREA BYTE COUNT

G4: 65 BYTES (NORMAL), 197 BYTES (HA ERROR)
G2: 33 BYTES
G0: 33 BYTES + 0.043 $(K_L + D_L) - 8$ BYTES

GW16/GW17 BIT SEQUENCE = 0, 17, 16, 15, 14, 13, 12, 11, 10, 7, 6, 5, 4, 3, 2, 1

NORMAL BIT SEQUENCE = 0, 1, 2, 3, 4, 5, 6, 7, 10, 11, 12, 13, 14, 15, 16, 17

Figure 6-40. Read Timing

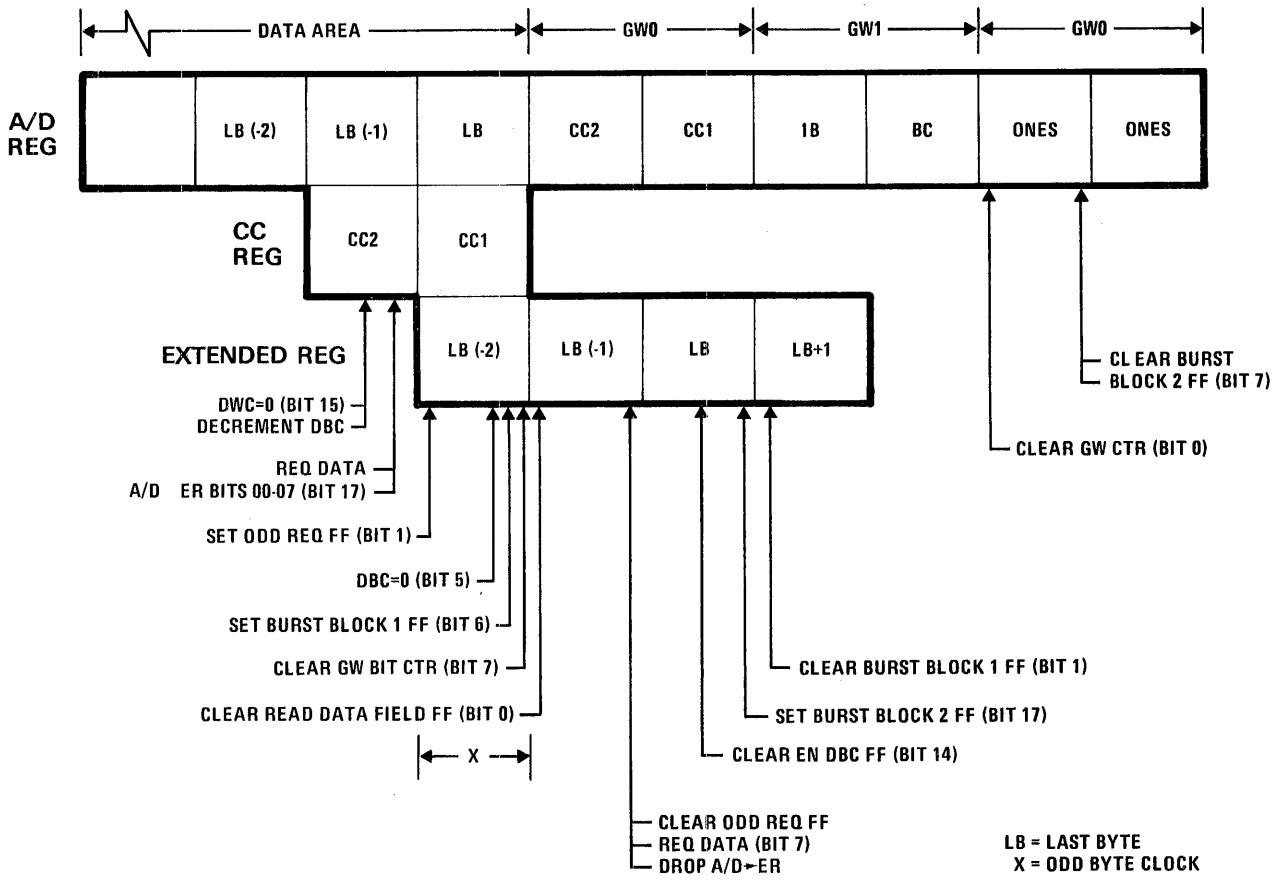


Figure 6-41. Odd Byte Read Timing

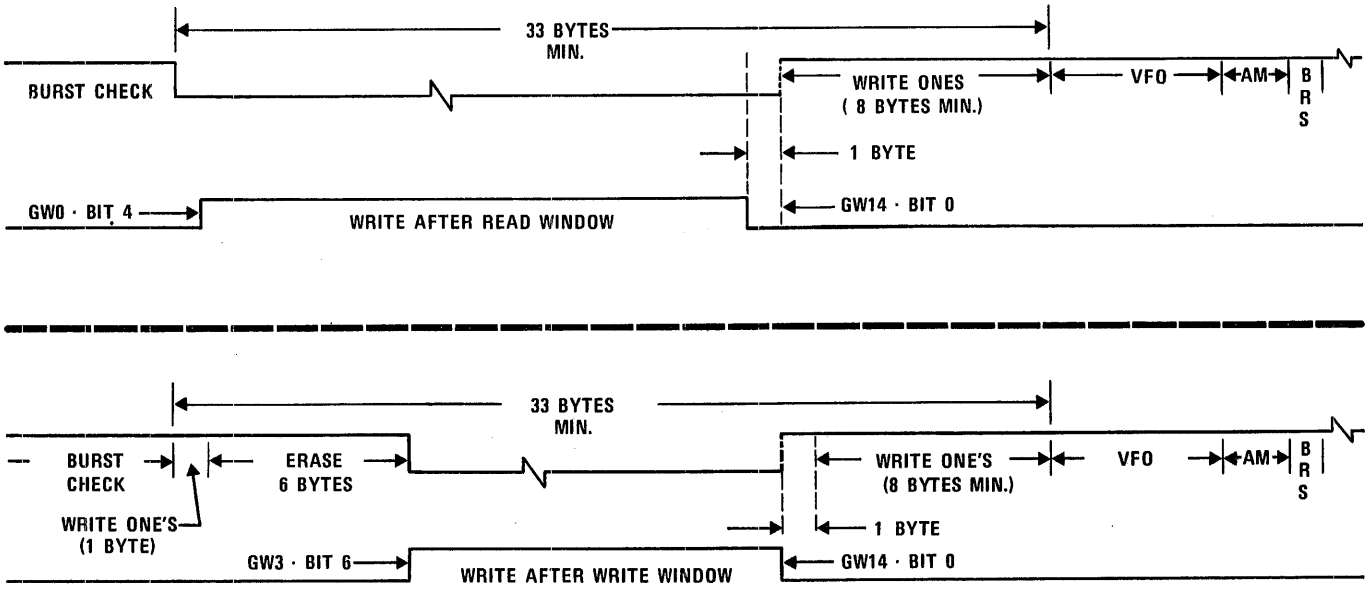


Figure 6-42. Write After Read/Write After Write Ones Generation and Window

ERROR DETECTION LOGIC

A four-byte burst-check is appended to every field recorded on the disc to check the validity of the data. The two bytes immediately following the data are termed the cyclic burst (CB) and consist of a residual count of the complemented exclusive OR of all bits in the appended field.

During a *Write* operation, CB is compiled and recorded on the disc surface with the data. During a *Read* operation, CB is recompiled and compared with the CB received from the disc surface.

The bit count appendage (BCA) which follows the CB consists of one indicator byte (IB) and one bit count (BC) byte. The IB is not used by the IFA and is always set to zero. The BC byte contains a residual count of the complement of all 1 bits in the appended field (including BRS and the first CB byte). During a *Write* operation, the BCA is compiled and recorded on the disc surface after the CB bytes. During a *Read* operation, the BC is recompiled and compared to the BC recorded on the disc surface.

Figures 6-43 and 6-44 illustrate the timing considerations for sequencing CC and BCA information through the BC, CC and A/D registers.

Bit Count Appendage (BCA) Generation and Checking (Figures 6-47 and 6-48)

During a *Write* operation, the first byte of BCA (indicator byte) is generated by forcing A/D bit 16 clear and clocking 0's into the eight low order positions of the A/D register. During the next eight counts, the 0's are shifted to the high order positions and the low order positions are filled with 1's. During the next eight counts the 1's residing in A/D are replaced by BC information from the CC register and the BC information is simultaneously shifted up to the high order positions. The low order positions are filled with 1's.

The BC information and 1's residing in the A/D register are then shifted out and written on the disc. From this point forward the A/D register is continually filled with 1's which are transferred to disc only if the operation being performed is a *Format Write*.

During a *Read* operation, the first byte of BCA is placed in the A/D register but is not sent to the error checking logic. During the next eight counts, the BC information is simultaneously sent to the A/D register from the disc and to the error checking logic where it is compared bit-by-bit with the recompiled BC information from the CC register.

Cyclic Code Generation and Checking (Figures 6-45 and 6-46)

The cyclic code (CC) bytes are compiled by performing a bit-by-bit exclusive OR on the data output of the A/D register and the output of the Cyclic Code register (Cyclic Code register initially set to zero). The serial accumulation of the exclusive OR is stored in the Cyclic Code register and the complement of the serial accumulation is stored in the A/D register following the data as it is shifted out to the disc drive. If additional data is to be read or written, the complemented CC information residing in the A/D register is destroyed. When the last data word has been shifted out of the A/D register, the complemented CC left in the A/D register is shifted out and written on the disc. When reading, the recomputed CC information residing in the Cyclic Code register is serially exclusive ORed with the cyclic code received from the disc and sent to the cyclic code error checking circuitry.

If the last data word from shared resources contains only one byte of information cyclic code byte 2 (CC2) is recorded on the disc prior to cyclic code byte 1 (CC1). When the field is read the CC bytes are recomputed and compared as described above.

DEADSTART CONTROL STORAGE LOAD LOGIC

The Deadstart Control Storage Load Logic interprets control signals generated by switch settings on the computer Control Panel and uses this information to initiate a *Read* operation. With the exceptions noted below, a *Deadstart CS Read* is performed in the same manner as a programmed read.

- If performed in operator mode or programmer mode the *Read* operation is preceded by a hardware initiated *Restore* function and *Set Head Advance* function
- If performed in maintenance mode, the *Read* operation is preceded by a hardware initiated *Set Head Advance* function only
- a MASTER CLEAR or TERMINATE function preceding *Deadstart CS Load* presets the Data Byte Counter to 6400_8 words and selects drive 0

This operation loads control storage (CS) in the 7300 Computer from logical drive 0. A *CS Load* may be initiated by the following:

- Pressing the POWER ON switch (provided the system is not powered up when the switch is pressed) or
- Pressing the RESET/LOAD switch or
- Executing a *CS Load* command

Figure 6-49 shows the procedure for executing a *CS Load* and/or a Main Storage Load operation using either the Deadstart Logic or a *CS Load* command.

A power on CS load initiated in operator mode, programmer mode or maintenance mode, or a *Deadstart CS Load* initiated in operator mode or programmer mode, starts loading from cylinder 000, track 01. A *Deadstart CS Load* initiated in maintenance mode starts loading from the current cylinder address and the current head address+1.

A program initiated *CS Load* may be performed in maintenance mode at any time, or within one second following a power-on *CS Load* if in operator mode or programmer mode. In either case, loading starts from the current cylinder address and the current head address.

If a Burst Check Error, No Sync Find, End of Cylinder or Seek Incomplete status indication occurs while loading in operator mode or programmer mode, the current operation is terminated and loading automatically restarts from cylinder 000, track 01. A status error detected while loading in maintenance mode terminates a *CS Load* operation and no attempt is made to re-read the CS data. A status error detected in either mode lights the BURST CHECK and I/O FAULT indicators on the Processing Unit Control Panel

At the conclusion of a *CS Load* performed in operator mode or programmer mode, the IFA transmits an AUTOLOAD signal to shared resources which activates the executive processor and forces the CS Address register in the processing unit to the starting location of the micro-instruction routine used to load the operating system. If the *CS Load* was performed in maintenance mode, the operator must press the AUTOLOAD pushbutton to activate the executive processor and load the operating system. A flow chart showing the logical operation of CS load logic is provided on Figure 6-50.

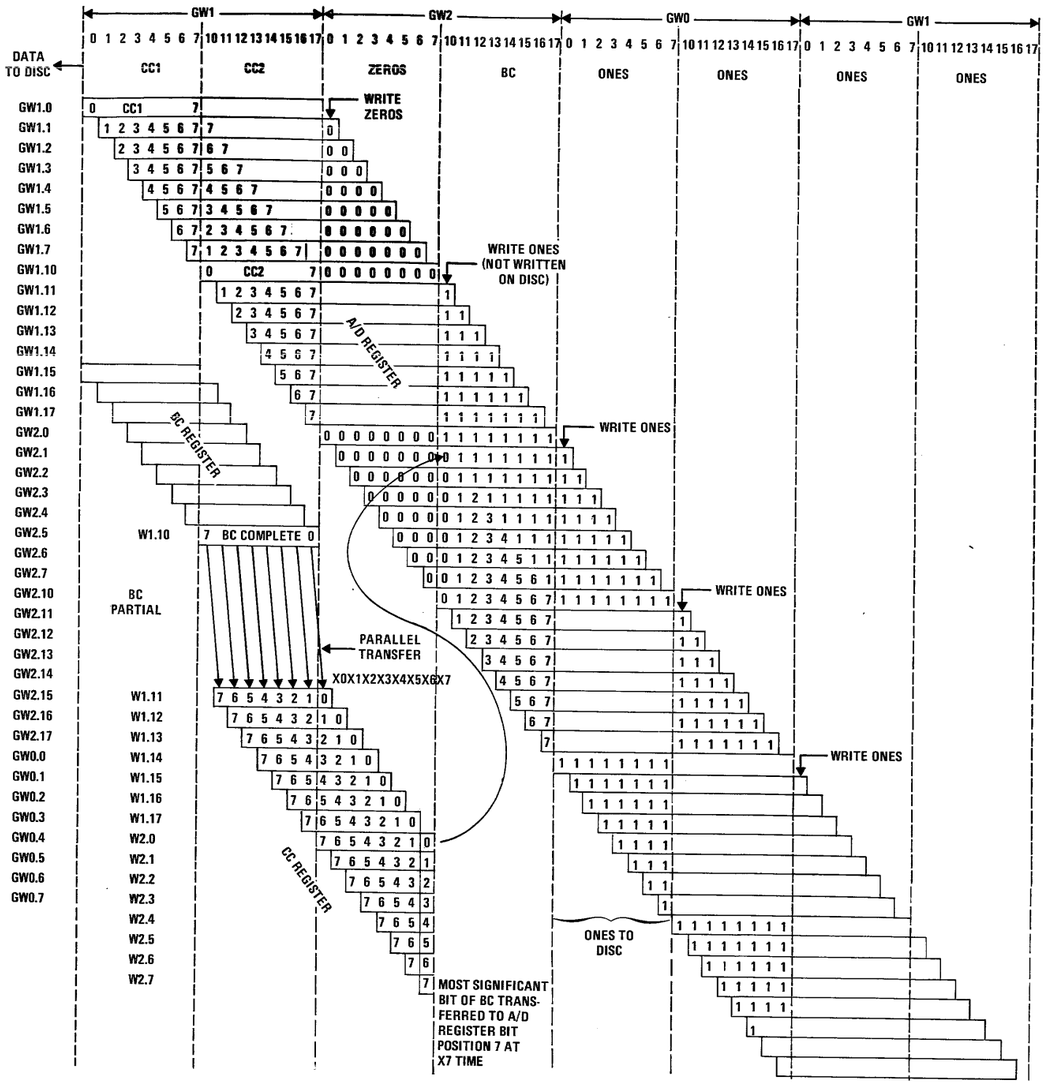


Figure 6-43. Write CC and BCA

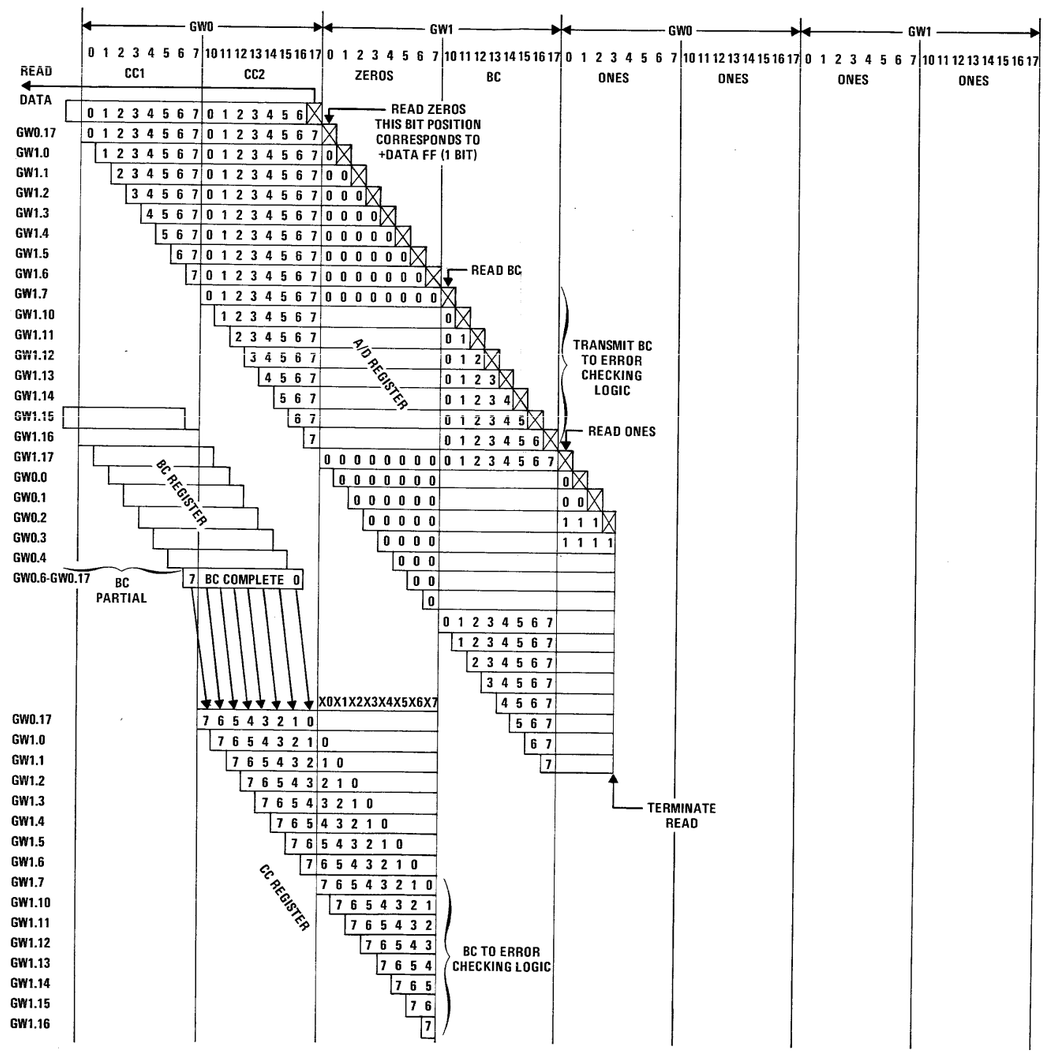


Figure 6-44. Read CC and BCA

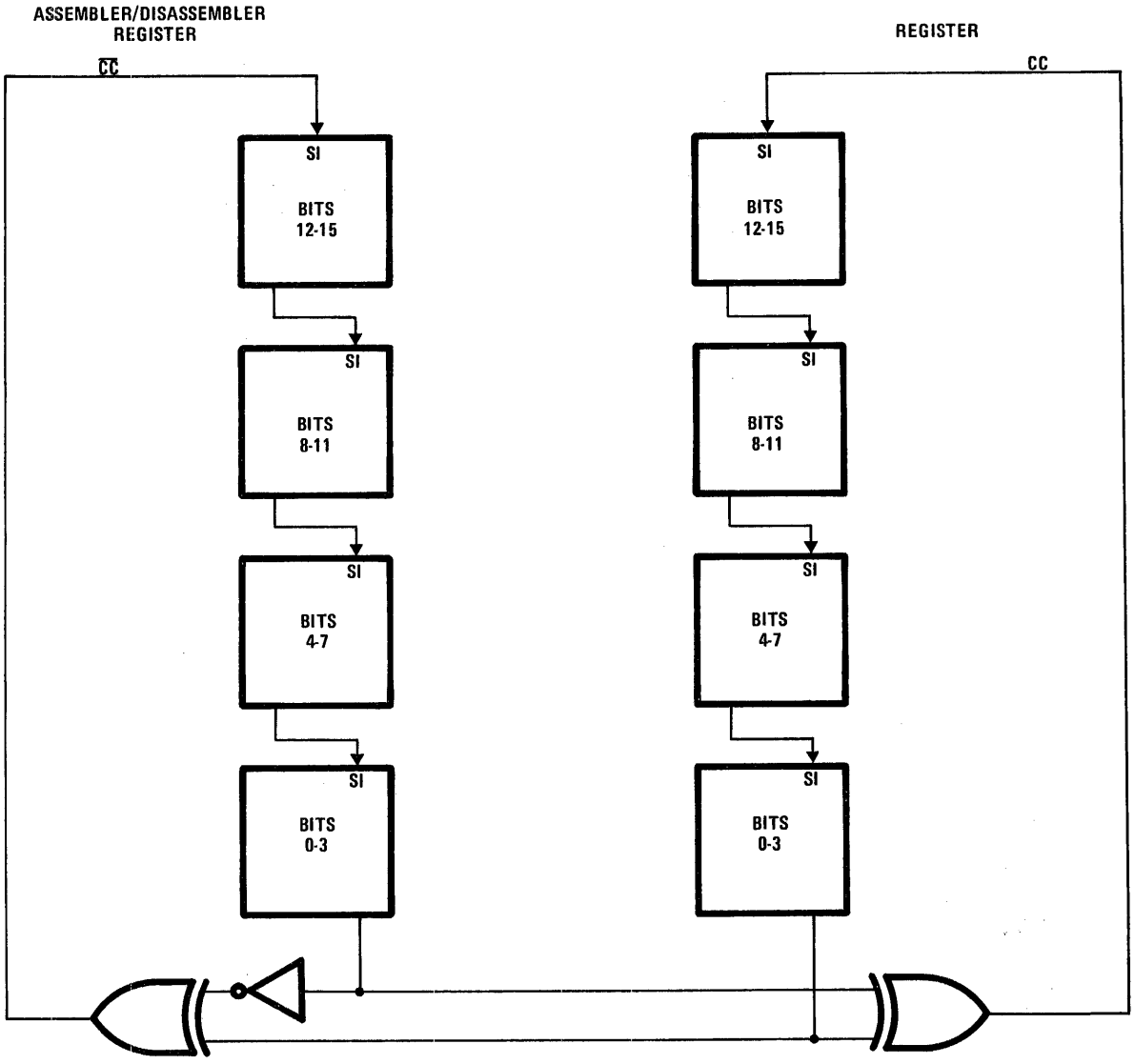


Figure 6-45. Generate Cyclic Code

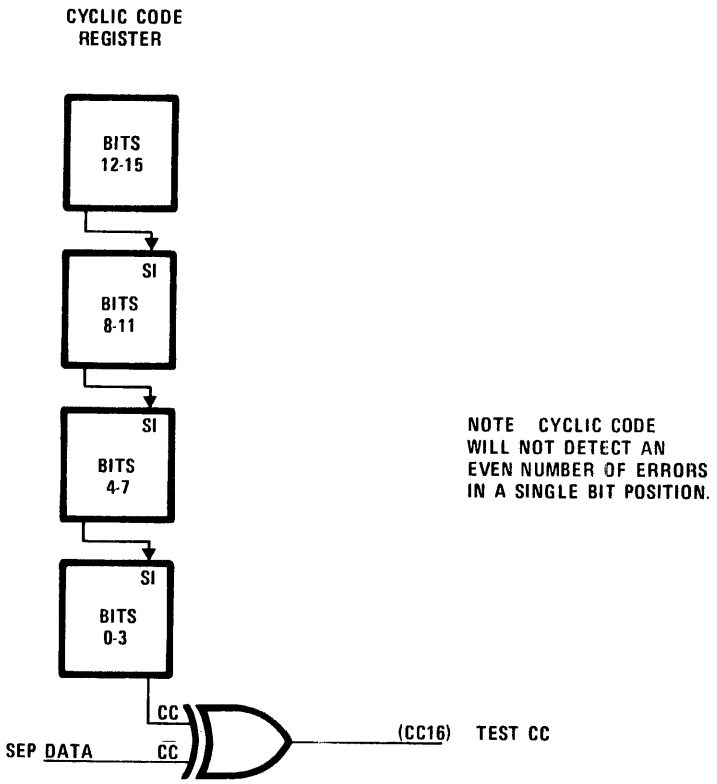


Figure 6-46. Read Cyclic Code

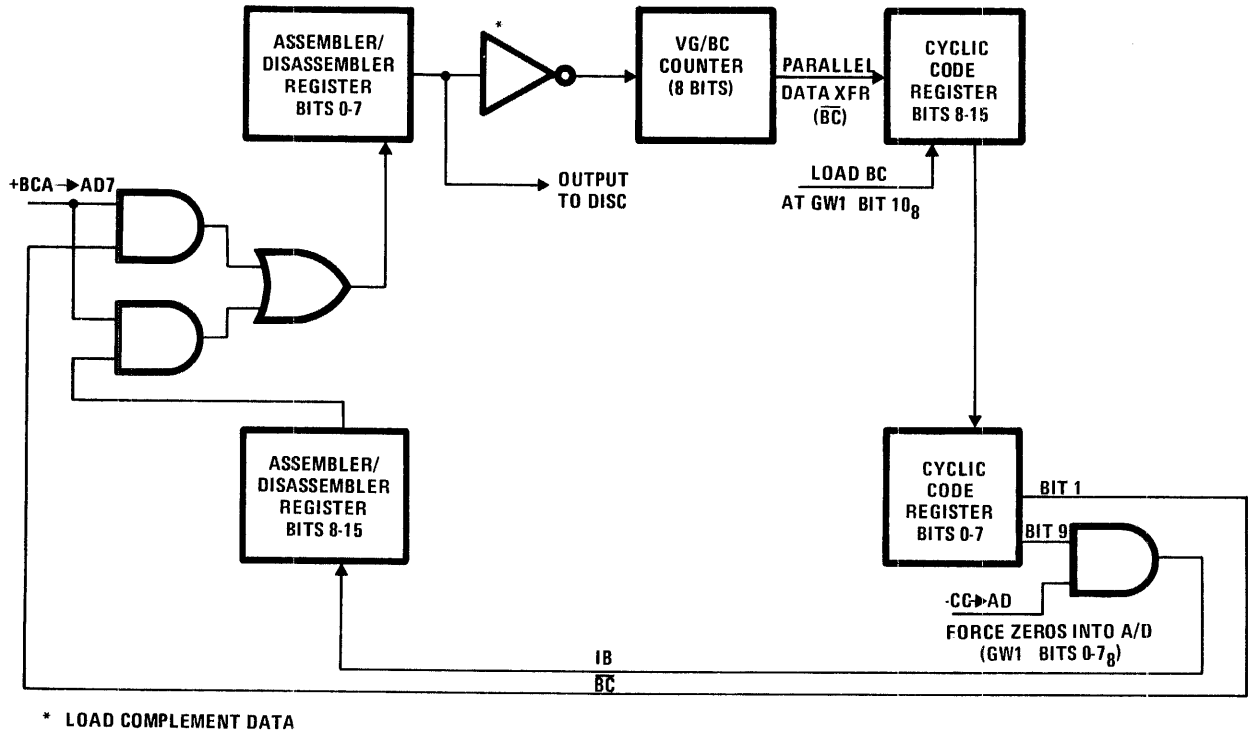


Figure 6-47. Generate BCA

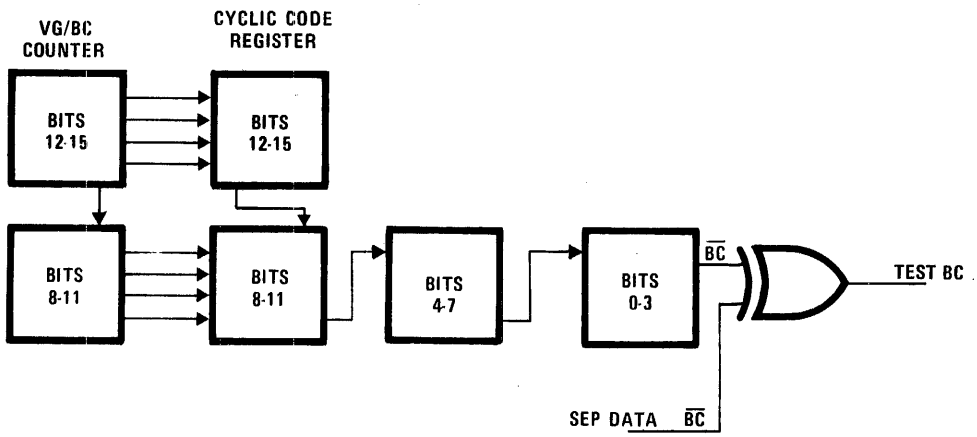


Figure 6-48. Read Bit Count

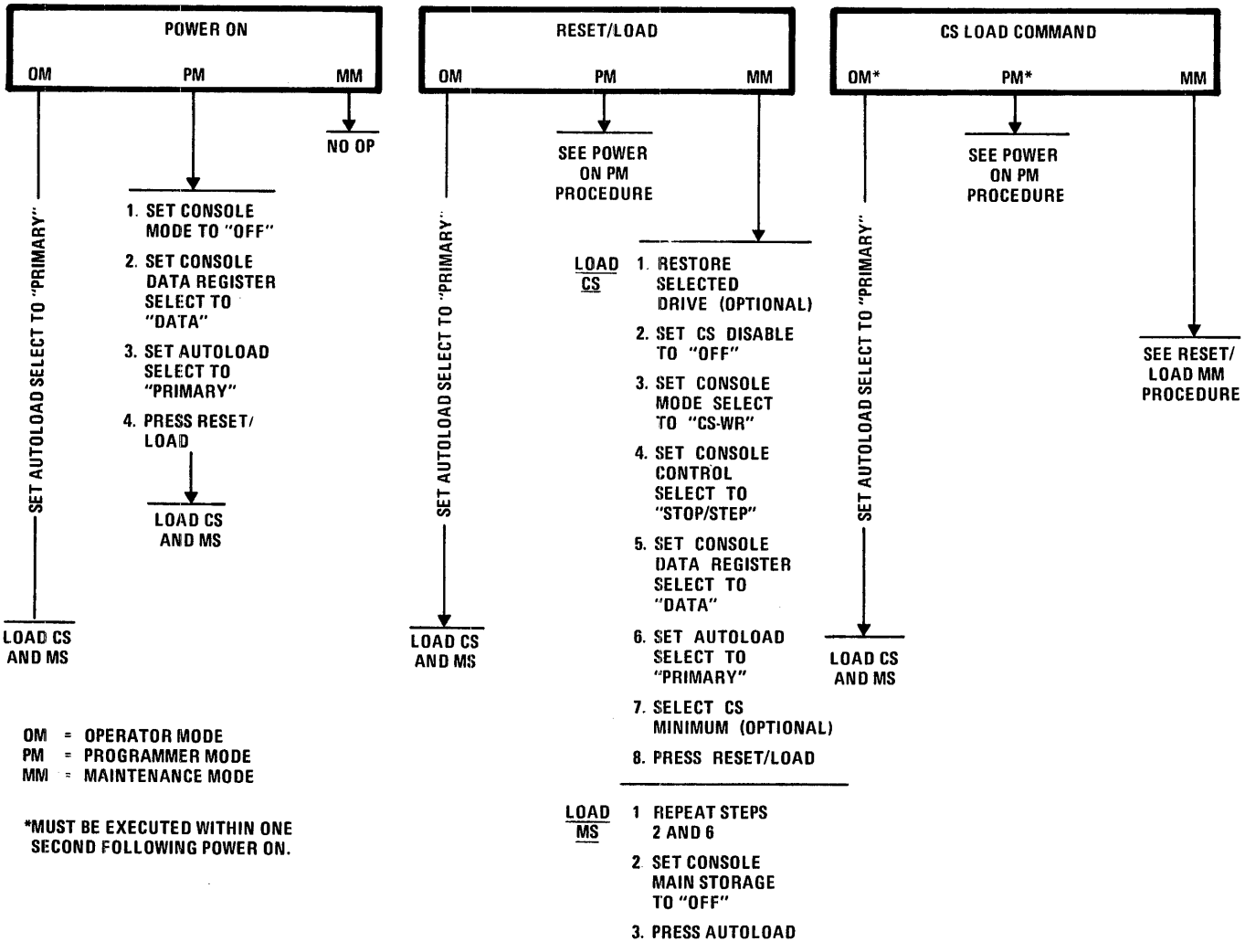


Figure 6-49. CS Load/MS Load Procedure

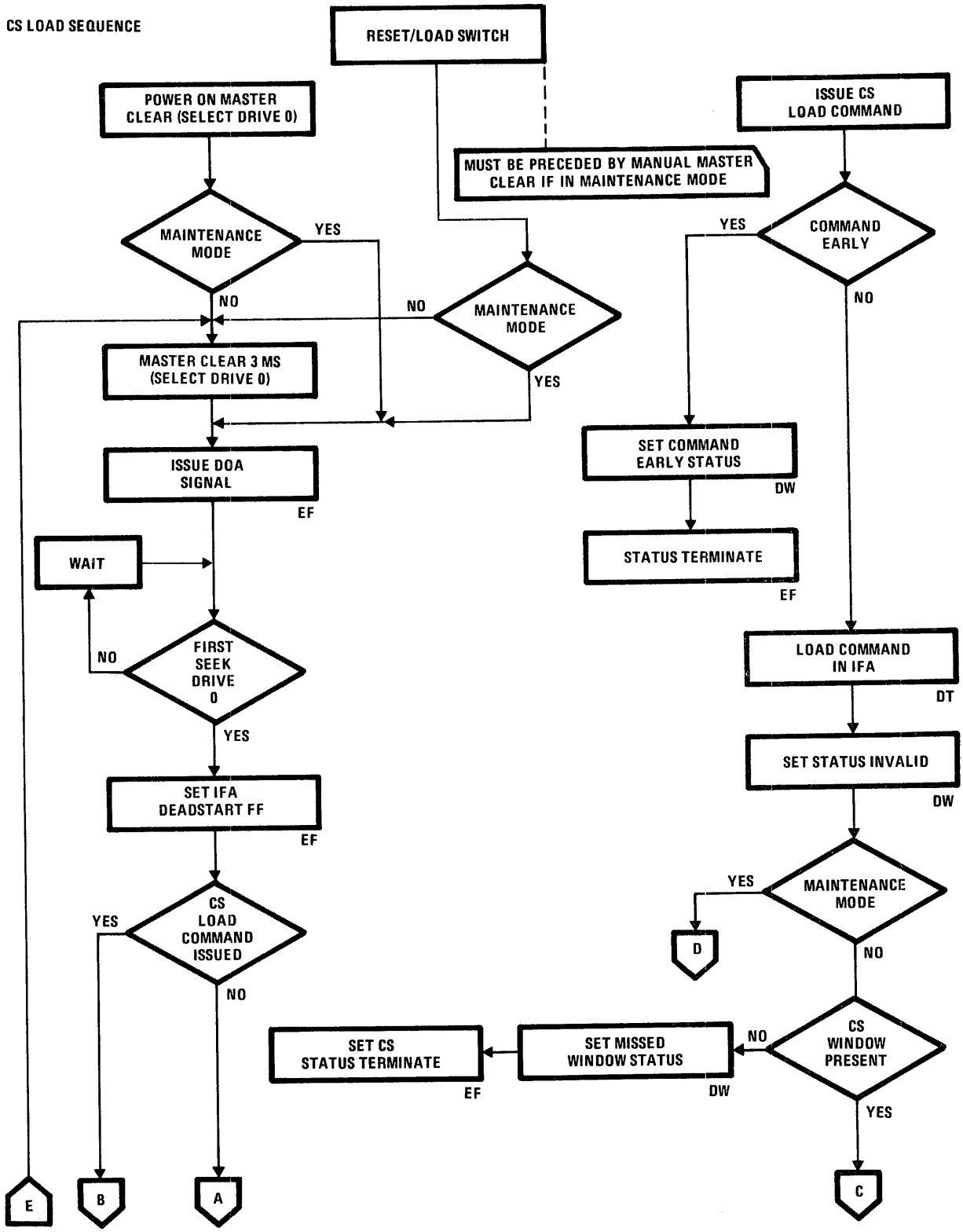


Figure 6-50. CS Load Sequence

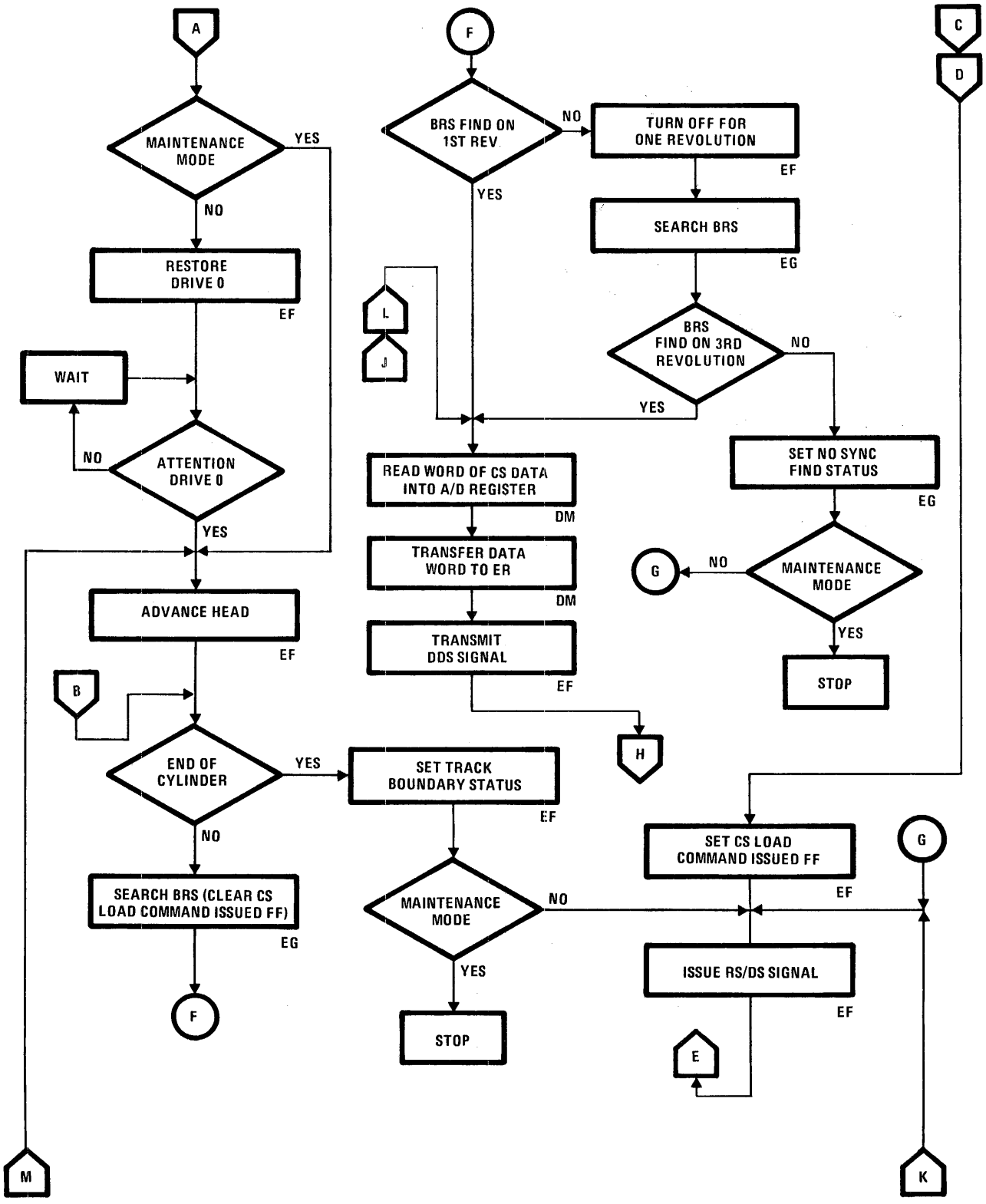


Figure 6-50. CS Load Sequence (Continued)

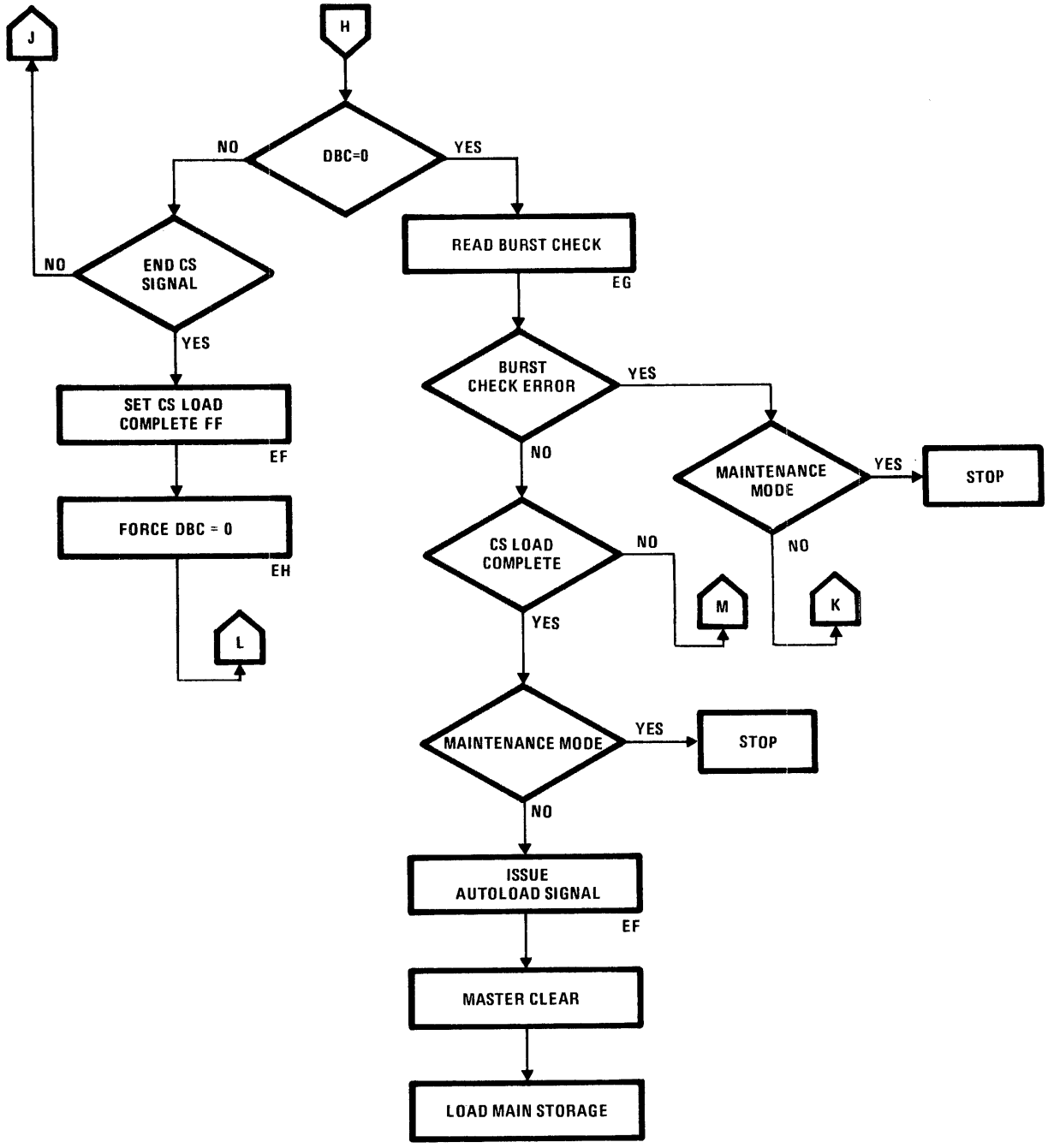


Figure 6-50. CS Load Sequence (Continued)

APPENDIX 6A

DISC MLI FLOW DIAGRAMS

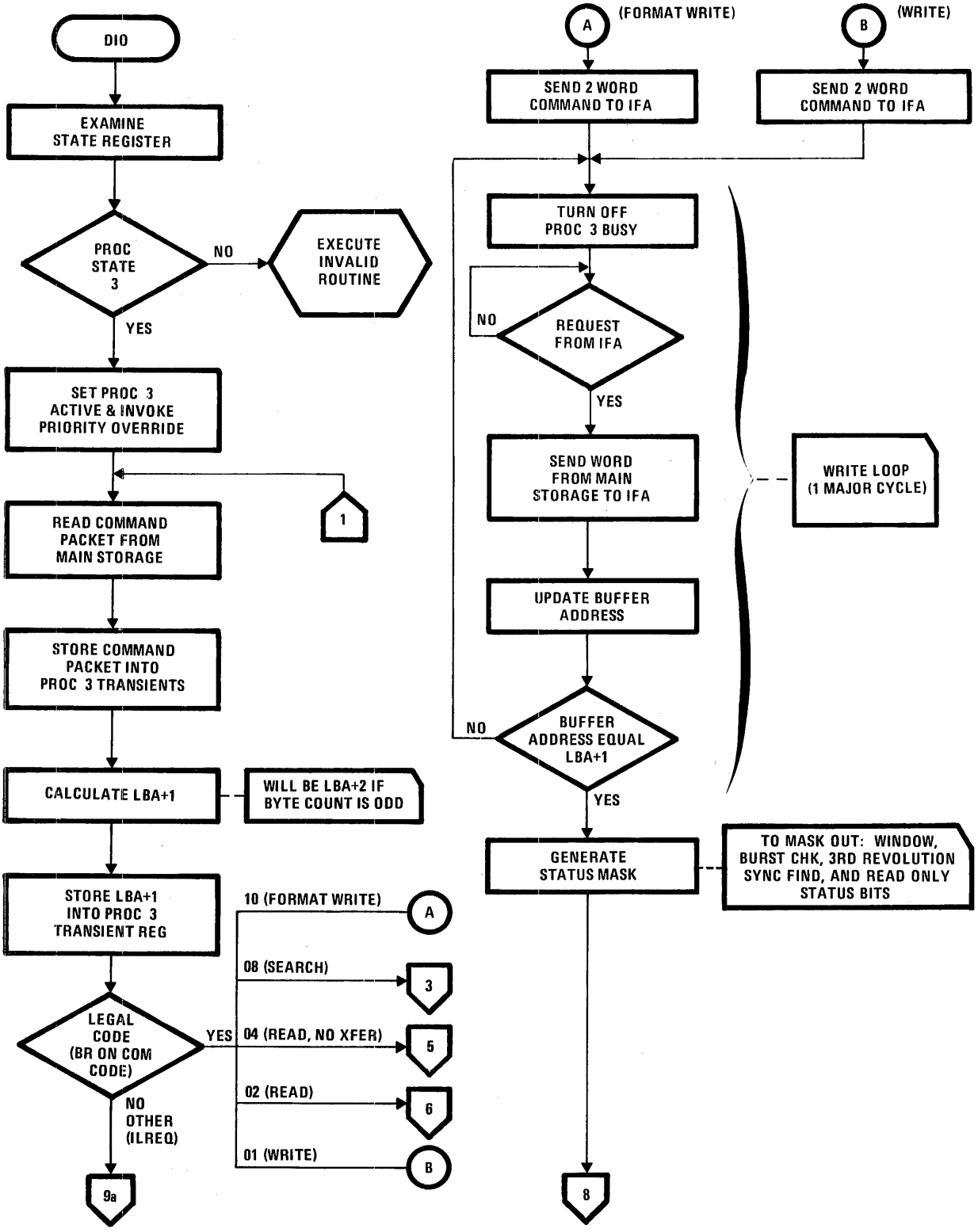


Figure 6A-1. Disc MLI Flow Diagram

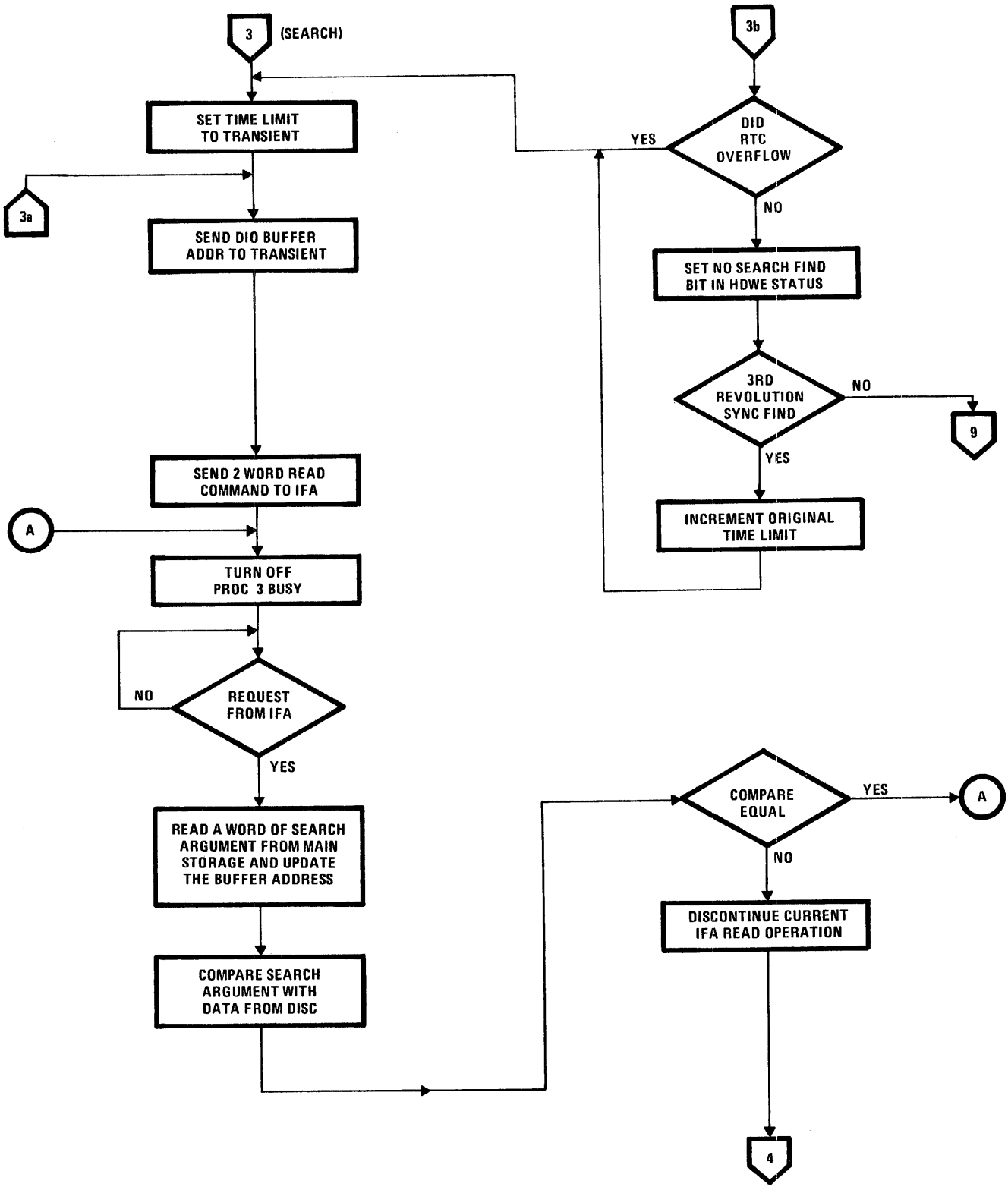


Figure 6A-1. Disc MLI Flow Diagram (Continued)

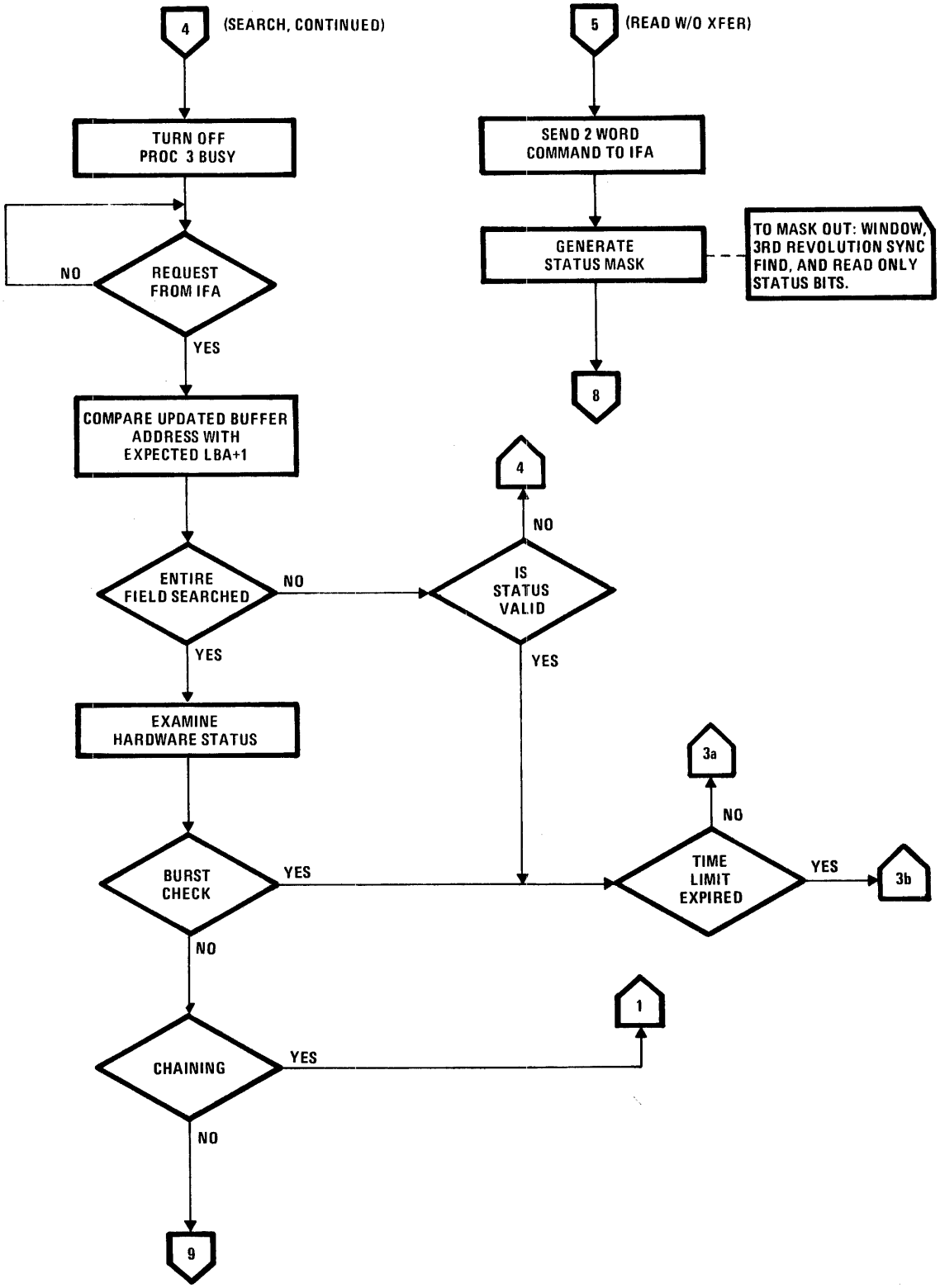


Figure 6A-1. Disc MLI Flow Diagram (Continued)

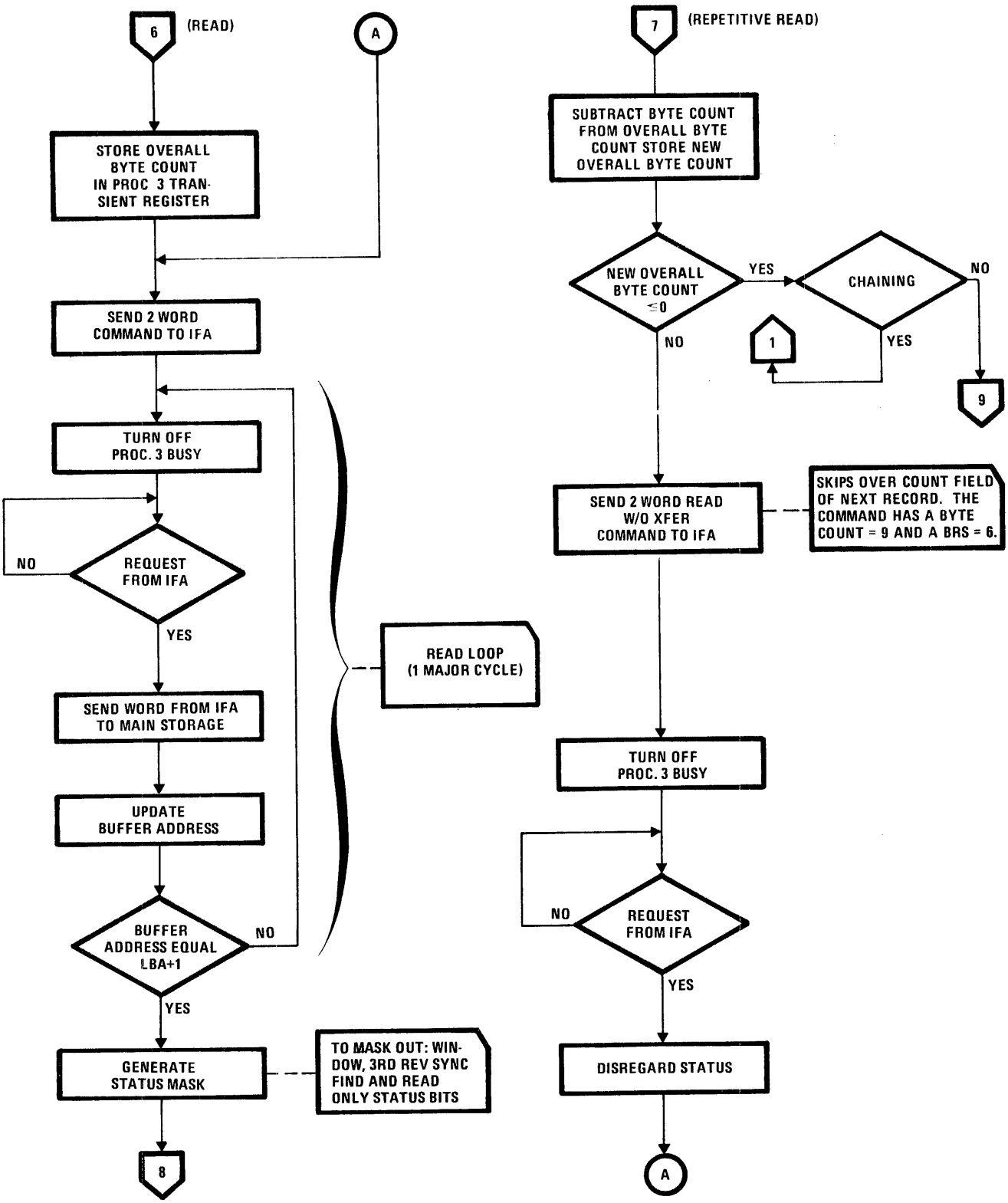


Figure 6A-1. Disc MLI Flow Diagram (Continued)

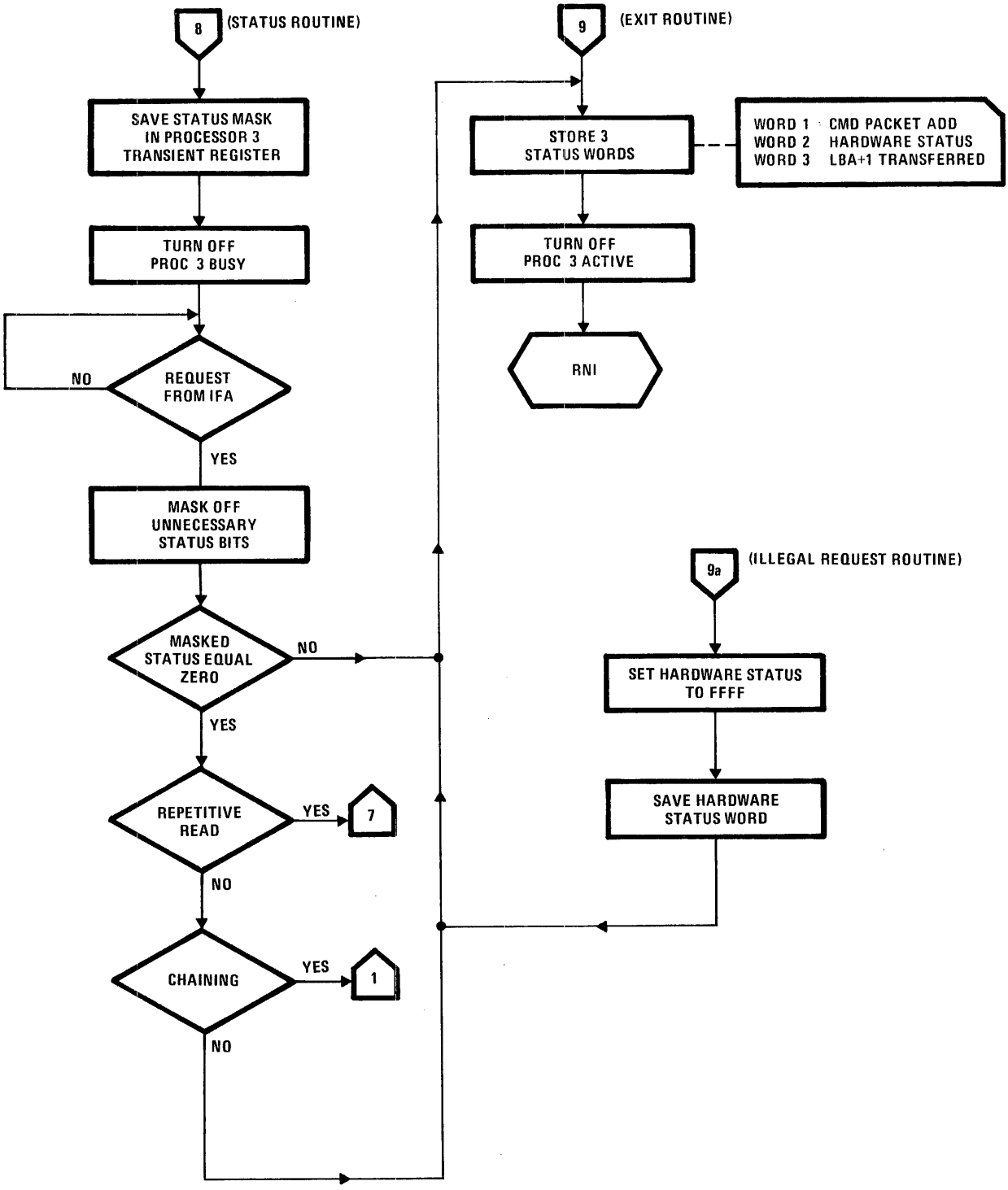


Figure 6A-1. Disc MLI Flow Diagram (Continued)

APPENDIX 6B

DRIVE INTERFACE CABLE CONNECTIONS

- Bus Cable

This is a shielded cable consisting of 52 twisted pairs with a characteristic impedance of approximately 95 ohms. It contains all signal lines which are common to all drives in the string. (Refer to Table 6B1.)

Table 6B1. Bus Cable (Multiplex)

Input Pin	Line Name	Output Pin
A	Unit Bus 0	A
B	return ground	B
C	Unit Bus 1	C
D	return ground	D
E	Unit Bus 2	E
F	return ground	F
H	Unit Bus 3	H
J	return ground	J
K	Unit Bus 4	K
L	return ground	L
M	Unit Bus 5	M
N	return ground	N
P	Unit Bus 6	P
R	return ground	R
S	Unit Bus 7	S
T	return ground	T
U	Last Disc Up	U
V	return ground	V
W	Set Cylinder (tag line)	W
X	return ground	X
Y	Set Head (tag line)	Y
Z	return ground	Z
a	Control (tag line)	a
b	return ground	b
c	Select Unit 0	c
d	return ground	d
f	Select Unit 1	f
g	return ground	g
h	Select Unit 2	h
i	return ground	i
j	Select Unit 3	j
k	return ground	k
m	Select Unit 4	m
n	return ground	n
p	Select Unit 5	p
q	return ground	q
r	Select Unit 6	r
s	return ground	s
t	Select Unit 7	t
u	return ground	u
v	Select Unit S	v
w	return ground	w
x	Attention Unit 0	x
y	return ground	y
z	Attention Unit 1	z
AA	return ground	AA
AB	Attention Unit 2	AB
AC	return ground	AC
AD	Attention Unit 3	AD
AE	return ground	AE
AF	Attention Unit 4	AF
AH	return ground	AH
AJ	Attention Unit 5	AJ
AK	return ground	AK
AL	Attention Unit 6	AL
AM	return ground	AM
AN	Attention Unit 7	AN
AP	return ground	AP

Table 6B1. (Continued)

Input Pin	Line Name	Output Pin
AR	Attention Unit S	AR
AS	return ground	AS
AT	Reserved	AT
AU	Reserved	AU
AV	Not Used	
AW	Reserved	AW
AX	Cylinder Address Reg. 1	AX
AY	return ground	AY
AZ	Cylinder Address Reg. 2	AZ
BA	return ground	BA
BB	Cylinder Address Reg. 4	BB
BC	return ground	BC
BD	Cylinder Address Reg. 8	BD
BE	return ground	BE
BF	Cylinder Address Reg. 16	BF
BH	return ground	BH
BJ	Cylinder Address Reg. 32	BJ
BK	return ground	BK
BL	Cylinder Address Reg. 64	BL
BM	return ground	BM
BN	Cylinder Address Reg. 128	BN
BP	Reserved	BP
BR	Selected Unit Busy	BR
BS	return ground	BS
BT	Selected Unit on Line	BT
BU	return ground	BU
BV	Selected Index	BV
BW	return ground	BW
BX	Selected File Unsafe	BX
BY	return ground	BY
BZ	Selected Unit Seek Incomplete	BZ
CA	return ground	CA
CB	Selected Unit End of Cylinder	CB
CC	return ground	CC
CD	Selected Unit Pack Change	CD
CE	return ground	CE
CF	Selected Unit Write Current Sense	CF
CH	return ground	CH
CJ	Heads Extended (switch)	CJ
CK	Controlled Ground	CK
—	Not Used	CL
CL	Sequence In (+36 vdc)	
—	Sequence Out (+36 vdc)	CM
CM	Not Used	
CN	Selected Unit Read Only	CN
CP	return ground	CP
CR	Sequence Power Out	—
—	Sequence Power In	CR
CS	Reserved	CS

- Unit Cable

This is a shielded cable consisting of two-95 ohm coaxial cables, two No. 10 AWG wires, and four No. 22 twisted pairs. It contains all signals unique to the particular drive. (Refer to Table 6B2.)

Table 6B2. Unit Cable (SIMPLEX)

Pin	Line Name
1	
2	-3 vdc
3	
4	Return ground (pin 2)
5	
6	
7	
8	
9	
10	
11	
12	Read Data/Write Data
13	+3 vdc
14	
15	
16	Spare
17	
18	Spare
19	
20	
21	
22	Unit Is Selected
23	
24	
25	Frame ground
26	Return ground (pin 16)
27	Return ground (pin 22)
28	Return ground (pin 18)

INDEX

COMMENTS FORM

7300 Design Description VOL 3 2501.003

Please send us your comments, to help us produce better publications. Use the space below to qualify your responses to the following questions, if you wish, or to comment on other aspects of the publication. Please use specific page and paragraph/line references where appropriate. All comments become the property of the Memorex Corporation.

- Is the material:

	Yes	No
Easy to understand?	☐	☐
Conveniently organized?	☐	☐
Complete?	☐	☐
Well illustrated?	☐	☐
Accurate?	☐	☐
Suitable for its intended audience?	☐	☐
Adequately indexed?	☐	☐

- For what purpose did you use this publication (reference, general interest, etc.)?

  _____

- Please state your department's function: _____

  _____

Please use the space below to describe any specific comments which you feel will help us to produce a better publication. _____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

MEMOREX

First Class

Permit No. 250
Santa Clara
California 95050

Business Reply Mail

No Postage Necessary if Mailed in the United States

Postage Will be Paid By

Memorex Corporation

Santa Clara Publications
Mail Stop 00-21
1200 Memorex Drive
Santa Clara, California 95052

Thank you for your information.....

Our goal is to provide better, more useful manuals, and your comments will help us to do so.

.....Memorex Publications