(1) [illegible]
[illegible] Morns

Honeywell

HONEYWELL
INFORMATION
SYSTEMS INC.

TITLE: No. 43A232215

MTC-500 CONTROLLER, EPS-1
ENGINEERING PRODUCT SPECIFICATION, PART 1

Total Pages 152

DIST: L16-92

REVISION RECORD

REVISION LETTER	DATE	PAGES AFFECTED	APPROVALS	AUTHORITY
A	ISSUED FEB 04 1972	SH.1 THRU 128, 128.1, 129 THRU 151F	[illegible] 728 [illegible] 14 [illegible]	

1.0 GENERAL DESCRIPTION

This document specifies the requirements for control of magnetic tape handlers, with the media recorded in either the phase encoded (PE) mode or in the non-return to zero (NRZI) mode. Both seven track and nine track handlers will be accommodated in this subsystem. The basic units of this subsystem are:

Microprogrammed Peripheral Controller (MPC)
Link Adapter (LA)
Tape Controller Adaptor (TCA)
Magnetic Tape Hnadlers (MTH)
Device Switch (SW)

This document defines the design requirements for the controller. The controller adaptor, switch and handlers are defined in other documents.

1.1 SUBSYSTEM OBJECTIVES

The objectives of the phase encoded subsystem are to:

- Provide single channel and dual simultaneous channel operation
- Provide field upgrade from single channel to dual simultaneous channel, with increases in number of handlers up to maximum capability of the subsystem.
- Provide a high degree of subsystem modularity
- Provide on-line diagnosis, and off-line repair
- Provide 1600 bpi phase encoding (PE) as well as 200, 556, 800 bpi non-return to zero, change on ones (NRZI)
- Operate on systems with a peripheral subsystem interface (PSI) channel on an input/output multiplexer (IOM), or on systems providing an equivalent interface discipline.
- Control the magnetic tape units by microprograms residing in the microprogrammed peripheral controller (MPC)
- Maintain error statistics related to data transfers

1.2 APPLICABLE DOCUMENTS

EPS-1 Microprogrammed Peripheral Controller (MPC), 43A177875
EPS-1 Link Adapter (LA), 43A177879
EPS-1 Peripheral Subsystem Interface (PSI), 43A177874
EPS-1 IOM Peripheral Subsystem Interface Adapter, 43A177880
EPS-1 Input/Output Multiplexor (IOM) Central, 43A219604
EPS-1 Magnetic Tape Subsystem, 43A239623
EPS-1 Magnetic Tape Handler MTH500 Series 59A502064
Common MPC Maintainability 43A237500

1.2 APPLICABLE DOCUMENTS (continued)

EPS-1 General Design Requirements for GE655 System, 43A177851

NPL Device Level Interface (DLI) for Magnetic Tape Subsystem, BL0024

Area Objectives for New Product Line (NPL) Microprogrammed Peripheral Controller (MPC) Subsystems, BL0021

The following documents apply to media characteristics and to techniques for recording the media. In case of conflict between this controller specification and the referenced documents, the referenced documents shall take precedence.

X3B1/402 - April 1970
Draft USA Standard - Unrecorded Magnetic Tape for Information Interchange

X3B1/425 - April 1970
USA Standard - Recorded Magnetic Tape for Information Interchange (200 CPI. NRZI)

X3B1/426 - April 1970
USA Standard - Recorded Magnetic Tape for Information Interchange (800 CPI, NRZI)

X3B1/400 - April 1970
Proposed USA Standard - Recorded Magnetic Tape for Information Interchange (1600 CPI, Phase Encoded)

1.3 DEFINITIONS

MPC	Microprogrammed Peripheral Controller
IOM	Input/Output Multiplexor
TCA	Tape Controller Adaptor
LA	Link Adaptor
MTH	Magnetic Tape Handler
EUS	External User System refers to the central system which is making use of the magnetic tape subsystem
PSIA	PSI channel adaptor in the IOM
PSI	Peripheral Subsystem Interface
IAI	Internal Adaptor Interface
DLI	Magnetic tape handler Device Level Interface

1.3 DEFINITIONS (continued)

Binary Mode - A PSIA mode in which the eight-bit bytes are packed contiguously in core

ASCII Mode - A PSIA mode in which the eight-bit bytes have a ZERO bit between them in core

ips Tape velocity at the MTH head in inches per second

bpi Bits per inch. Data recording density when considered longitudinally along tape. Corresponds to characters per inch in NRZI recording and to bytes per inch in PE recording.

fri Flux reversals per inch

PE Phase Encoded

NRZI Non Return to Zero, change on ones

frame A set of bits when considered laterally across the tape. A frame in 7-channel tape has up to 7 bits; a frame in 9-channel has up to 9 bits.

byte As used in this specification, a byte is a group of 8 bits which are considered as a unit

character - A group of 6 bits considered as a unit

BOT marker - A reflective marker which is located at the logical beginning of tape

EOT marker - A reflective marker located near the physical end of tape

standby-loaded - A handler mode in which the tape is loaded, and operator intervention is required to bring the handler to the ready state

standby-unloaded - A handler state in which the tape is unloaded, but the handler is capable of responding to a Load Tape command; other actions to bring the handler to the ready state require operator intervention

controller - This is the unit consisting of MPC, LA(s), TCA(s) and SW.

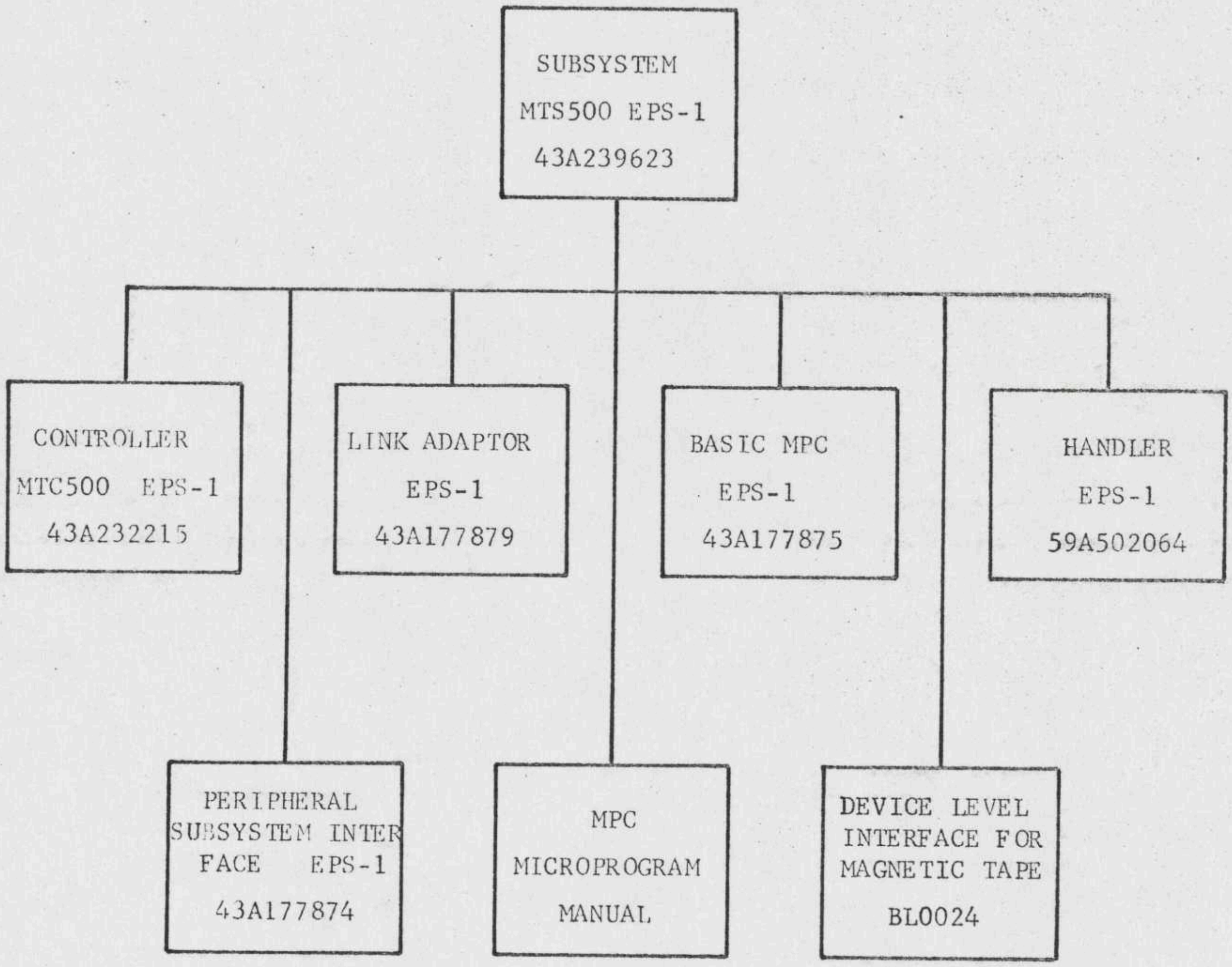
LCC Longitudinal Check Character

CCC Cyclic Check Character

ITR Integrated Test Routine

BLT Basic Logic Test

1.4 DOCUMENTATION TREE



2.0 SU:SYSTEM DESCRIPTION

2.1 SUBSYSTEM CAPABILITIES

This subsystem will control the movement of, recording on, and reading from ½ inch magnetic tape. The tape may be either seven track or nine track, and may be formatted in either the PE mode (for nine track) or NRZI mode (for seven or nine track). The tape may be moved at velocities ranging from 75 ips to 200 ips, and the data packing may vary from 200 bpi NRZI to 1600 bpi PE. The resulting nominal data transfer rates range from 11.25 K-bytes/sec (15 K-char/sec) to 320 K-bytes/sec (426.6 K-char/sec).

Both single channel and dual simultaneous data transfer operations are available, with up to sixteen magnetic tape handlers in the subsystem.

Special features include switched channels to the external user system, and optional code translation including ASCII to BCD (internal 6000 6-bit code), EBCDIC to BCD, and EBCDIC to ASCII. Dependent on the density code set in the handler, the code translation may be written on tape in either NRZI or PE modes.

The capabilities of this subsystem are summarized in figures 2.1.1, 2.1.2 and 2.1.3.

2.2 SUBSYSTEM CONFIGURATIONS

The single channel and dual-simultaneous channels subsystem configurations are illustrated in figures 2.2.1 and 2.2.2. The controller is composed of four major functional units -- link adaptor, microprogrammed controller, controller adaptor and device switch. These units are required in both the single and dual simultaneous configurations.

The link adaptor provides a hardware connection between the PSI cable and the back panel internal adaptor interface of the microprogrammed controller. The controller adaptor and device switch provide a hardware interface to the handlers, and also provide hardware control of certain device oriented functions. Finally, the microprogrammed controller (MPC) is the central intelligence unit which merges these functional units into a magnetic tape controller. This merging of functions is accomplished via microprograms contained in the MPC which personalizes it to become a magnetic tape controller.

SPEED DENSITY	75 ips	125 ips	200 ips
200 bpi	* 15K (20K)	25K (33.3k)	40K (53.3K)
556 bpi	41.7K (55.6K)	69.5K (92.6K)	111.2K (148.26K)
800 bpi	60K (80K)	100K (133.3K)	160K (213.3K)
1600 bpi	120K (160K)	200K (266.6K)	320K (426.6K)

FIGURE 2.1.1

Data Rates for Various Recording Densities and Handler Speeds for 9-Track Handlers

* The data rates are given as bytes/sec (characters/sec), with K = 1000.

SPEED DENSITY	75 ips	125 ips	200 ips
200 bpi	* 11.25K (15K)	18.75K (25K)	30K (40K)
556 bpi	31.275K (41.7K)	52.125K (69.5K)	83.4K (111.2K)
800 bpi	45K (60K)	75K (100K)	120K (160K)

FIGURE 2.1.2

Data Rates for Various Recording Densities and Handler Speeds for 7-track Handlers

* Data rates are given as bytes/sec (characters/sec), with K = 1000.

HANDLER FORMAT INTERNAL 6000 FORMAT	7 TRACK	9 TRACK
6000 CODE (6-bit)	NONE	EBCDIC
6000 CODE (6-bit)	NONE	ASCII (7-bit)
ASCII (7-bit)	NONE	EBCDIC

FIGURE 2.1.3

Optional Code Translation in Subsystem

2.2 SUBSYSTEM CONFIGURATION (continued)

The single channel and dual simultaneous channels subsystem configurations are illustrated in figures 2.2.1 and 2.2.2. The controller is composed of four major functional units -- link adaptor, microprogrammed controller, controller adaptor and device switch. These units are required in both the single and dual simultaneous configurations.

The link adaptor provides a hardware connection between the PSI cable and the back panel internal adaptor interface of the microprogrammed controller. The controller adapter and device switch provide a hardware interface to the handlers, and also provide hardware control of certain device oriented functions. Finally, the microprogrammed controller (MPC) is the central intelligence unit which merges these functional units into a magnetic tape controller. This merging of functions is accomplished via microprograms contained in the MPC which personalizes it to become a magnetic tape controller.

While the PSIA may have one, two, four, or eight logical channels assigned to it, the standard configuration for the MTC501 and MTC502 shall include two logical channels per PSIA.

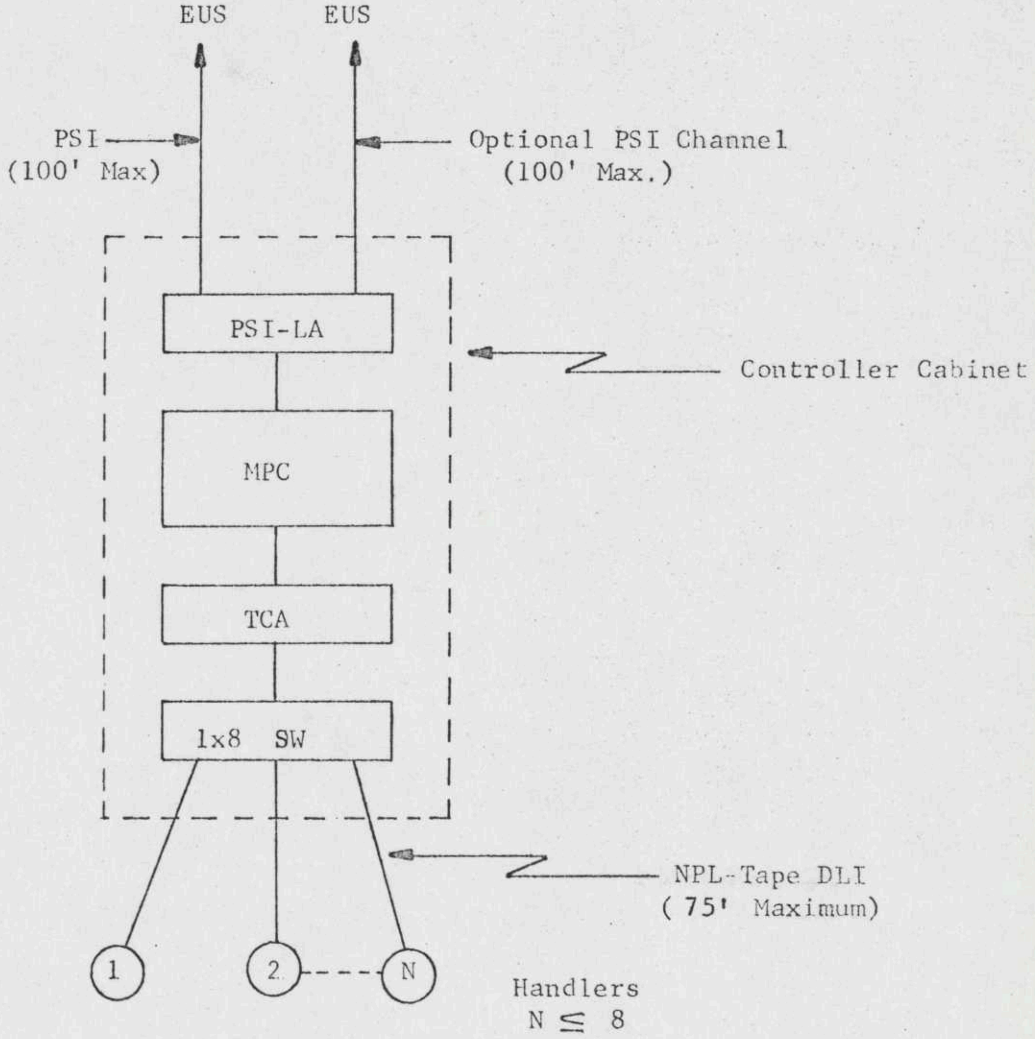


FIGURE 2.2.1 - SINGLE CHANNEL SUBSYSTEM

SHOWING MAJOR FUNCTIONAL UNITS

EXTERNAL USER CHANNELS

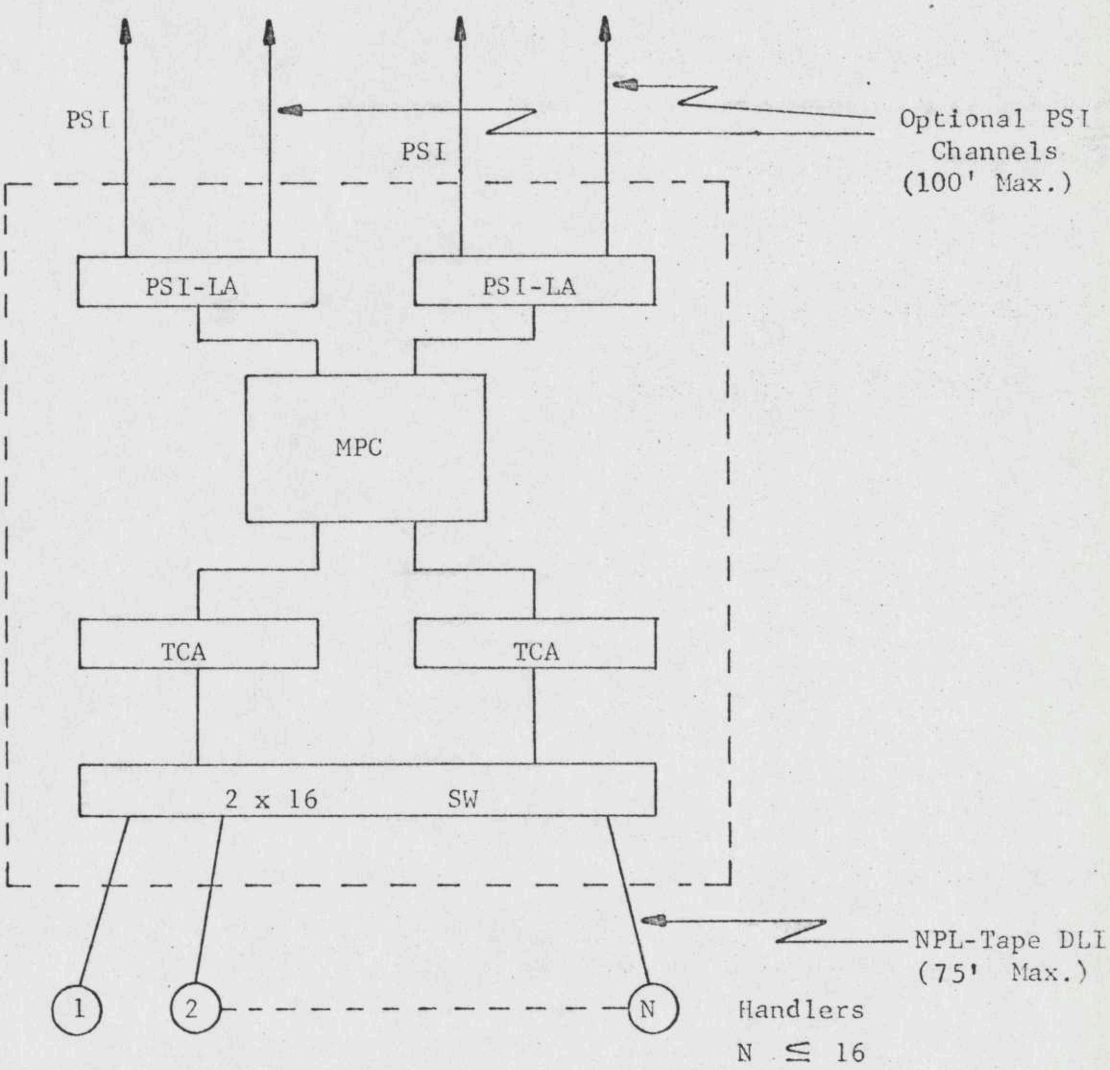


FIGURE 2.2.2

DUAL-SIMULTANEOUS CHANNELS SUBSYSTEM

SHOWING MAJOR FUNCTIONAL UNITS

2.3 GENERAL SUBSYSTEM CHARACTERISTICS

This subsystem will incorporate some features which are new to the 6000 line. This section summarizes the extent of these features.

o The controller does not contain any operator switches. Therefore, the modification of handler address formerly accomplished by configuration switches on magnetic tape controllers is not possible. Also, the set density commands (sections 5.4.15 through 5.4.18) have replaced the over-ride switches on the former Set High (now Set 800 BPI) and Set Low (now Set 556 BPI) commands.

o While the initial offering includes only the MTH502 and MTH505 handlers, it is anticipated that 7-track and 200 ips handlers will be included. Therefore, this controller is specified to include these handlers.

o The controller contains four forms of code translation, three of which are optional (see Section 3.7).

. BCD - Same as in previous 600 line subsystems (see Section 5.4.32 through 5.4.34). Not an option.

. 6-bit to/from EBCDIC - Conversion between 6000 line 6-bit format to/from 8-bit EBCDIC on tape (see Sections 5.4.35 and 5.4.36). (Optional features)

. 6-bit to/from ASCII - Conversion between 6000 line 6-bit format to/from 8-bit ASCII-7 on tape. (See Section 5.4.37 and 5.4.38). Optional feature.

. ASCII to/from EBCDIC - Conversion between ASCII-7 characters in 6000 to/from EBCDIC bytes on tape. (See sections 5.4.39 and 5.4.40). Optional Feature.

o Special Interrupt - When a special interrupt is sent to an external user system, information is included which defines the interrupting device and reason for the interrupt. (See Section 6.6)

o This subsystem is capable of being shared by independent external user systems. This is possible because of such features as the ability for the independent EUS's to communicate via the subsystem (See sections 5.6.5.7 through 5.6.5.9 - Lock Byte Application), and for these independent EUS's to selectively reserve any handler connected to the subsystem.

2.3 GENERAL SUBSYSTEM CHARACTERISTICS (continued)

- The subsystem will perform automatic retries for specified read type command (see Section 7.4) errors which have a Device Data Alert major status. This feature may be inhibited by the EUS using the technique of section 5.2.

- The subsystem will maintain an error log for specified errors occurring within the subsystem (see Section 7.5).

2.4 STANDARDS AND OPTIONS

The features of the handlers and controllers are described in the next two sections.

2.4.1 Controller Features

	MTC501	MTC502
Simultaneous data transfer	No	Yes
Maximum No. Handlers	8	16
Recording Densities		
200, 556, 800 bpi NRZI	Yes	Yes
1600 bpi PE	Yes	Yes
No. Tracks	7 & 9	7 & 9
Standard Code Translation		
BCD	Yes	Yes
Optional Code Translation		
6-bit to/from EBCDIC	Yes	Yes
6-bit to/from ASCII	Yes	Yes
ASCII to/from EBCDIC	Yes	Yes
Device Switch Modularity		
No handlers per module	4	4
Read Reverse Capability	No	No
Multi-System External Users	Yes	Yes

2.4.2 Handler Features

	MTH502	MTH505
No. Tracks	9	9
Speed Forward (ips)	75	125
Rewind Speed (ips)	500	500
Reverse Speed - Backspace (ips)	75	125
Recording Densities		
200, 556, 800 bpi NRZI	Yes	Yes
1600 bpi PE	Yes	Yes
Automatic Load	Yes	Yes
Power Window	Yes	Yes
Options		
7-track, 200,556,800 bpi NRZI	Yes	Yes
9-track, 1600 bpi PE only	Yes	Yes
Canister Load-Factory install	Yes	Yes
Canister Load - Field Upgrade	Yes	Yes
High altitude blower Factory install	Yes	Yes
High altitude blower field upgrade	Yes	Yes
D.C. Power Meter	Yes	Yes
Tape Movement Meter	Yes	Yes
ANSI Compatible Recording (per reference specifications listed in section 1.2)	Yes	Yes

3.0 CONTROLLER DESCRIPTION

The controller utilized in this subsystem will use the MPC as a central control mechanism. The link adaptor will provide a hardware interface to the PSI channel, with microprograms residing in the MPC controlling the dialog on the PSI.

The portion of the controller which communicates with the handlers will be a combination of microprogram, controller adaptor and device switch. The switch provides a hardware interface between the controller adaptor and handlers, with the dialog controlled by microprograms in the MPC.

The controller adaptor provides hardware control for those device-oriented functions which are not readily controlled by microprograms in the MPC. Functions which fall in this category include:

- LCC generation
- CCC generation
- PE data recovery
- PE/NRZI error detection/correction
- Data Buffering
- Code Translation
- Pack/unpack associated with code translation
- Generation of TCA status
- PE Preamble/Postamble Detection and generation
- BOT PE Identification Burst Generation

The MPC, in providing the major subsystem control, must include the following functions in its microprograms:

- Interpret commands received by LA from EUS
- Control dialog on PSI and LAI
- Gather MTH, TCA, MPC status, format and transmit to EUS via LA
- EOF generation/detection

3.0 CONTROLLER DESCRIPTION (continued)

- o Record and file gap control
- o CCC/LCC gap control
- o Write time base generation
- o Blank character time base generation
- o BOT PE identification burst detection

3.1 NPL AREA OBJECTIVES

The design of this controller, including the LA, MPC, and TCA hardware, but excluding the control microprograms, shall not preclude accomplishing the NPL functions as described in the document Area Objectives for New Product Line (NPL) Microprogrammed Peripheral Controller (MPC) Subsystems issued 2/15/71, when the controller hardware is reimplemented in NPL packaging.

3.2 DEVICE ADDRESSING

The controller in this subsystem shall be identified as device zero, and the handlers shall be numbered starting with number one. Thus, the handler numbers in a single-channel subsystem range between one and eight, and in a dual-channel subsystem the handler numbers range between one and sixteen.

Each device contains a five-bit code which is wired into the handler when the handler is installed in the subsystem. This wired-in number occupies bit positions 3 through 7 of summary status byte 4 (bit 3 = MSB) in the handler. This number, and the number on the front of the handler cabinet which is visible to the operator, must be made the same. These numbers are unique within the subsystem, and are not changed for the duration of the life of the handler on that subsystem. Since the visible number corresponds to the device code, the wired-in number also corresponds to the device code. The controller must maintain a table relating device codes to device switch ports. This table must be dynamic so that it is updated whenever a handler is added to or removed from the subsystem. Each time a command requires activity with a particular handler, and prior to initiating this activity. the controller will interrogate the summary status byte containing the wored-in number to be sure that the correct device is addressed.

3.2 DEVICE ADDRESSING (continued)

The legal device codes for these subsystems are as follows:

Device Code	Single-Channel	Dual-Channel
0	Controller	Controller
1	Handler 1	Handler 1
2	Handler 2	Handler 2
3	Handler 3	Handler 3
4	Handler 4	Handler 4
5	Handler 5	Handler 5
6	Handler 6	Handler 6
7	Handler 7	Handler 7
8	Handler 8	Handler 8
9	NL*	Handler 9
10	NL*	Handler 10
11	NL*	Handler 11
12	NL*	Handler 12
13	NL*	Handler 13
14	NL*	Handler 14
15	NL*	Handler 15
16	NL*	Handler 16

* NL means that this device code is not legal for single-channel subsystems.

3.3 CONTROLLER UTILIZATION OF HANDLER INTERFACE

The controller shall take special actions to verify that the correct handler is addressed, and that the interface to/from the handler is working properly. These activities are summarized as follows:

Prior to the execution of any command which will change the state of the handler, the controller will verify that the desired handler is addressed. This is accomplished by reading the address status byte stored in the handler.

Prior to the execution of any command involving a handler operation, and following the address verification mentioned above, the controller shall test the handler interface lines by executing DLI instruction Test.

Also, at the completion of execution of any command involving a handler, the controller shall read the handler status prior to formatting command termination status.

3.4 CONFIGURATION SWITCHES

The controller contains a set of sixteen switches which are readable by microprogram control. The function of these switches include configuration control and operational information. They are defined as follows:

Switch	Application
0	ITR BLT Bypass
1	ITR Loop Control
2	LA to be Used for Booting: 1 = LA on IAI Port 3 0 = LA on IAI Port 2
3	Number of LA's in Subsystem: 1 = One LA (Port 2) 0 = Two LA's (Ports 3 and 2)
4	Inhibit Interval Timer Runout Detection
5	1 = 9 track handler; 0 = 7 track handler
6 & 7	These two switches are defined as;

00	Device 4
01	Device 1
10	Device 2
11	Device 3

The function of switches 5-7 is to provide information for the load-from-device process (see section 5.4.24).

8 - 10	These three switches are defined as follows:

000	Not valid for this controller
001	" " " " "
010	Single channel controller
011	Dual channel controller

3.4 CONFIGURATION SWITCHES (continued)

100 Not valid for this controller
101 " " " " "
110 " " " " "
111 " " " " "

11
Not allocated
12

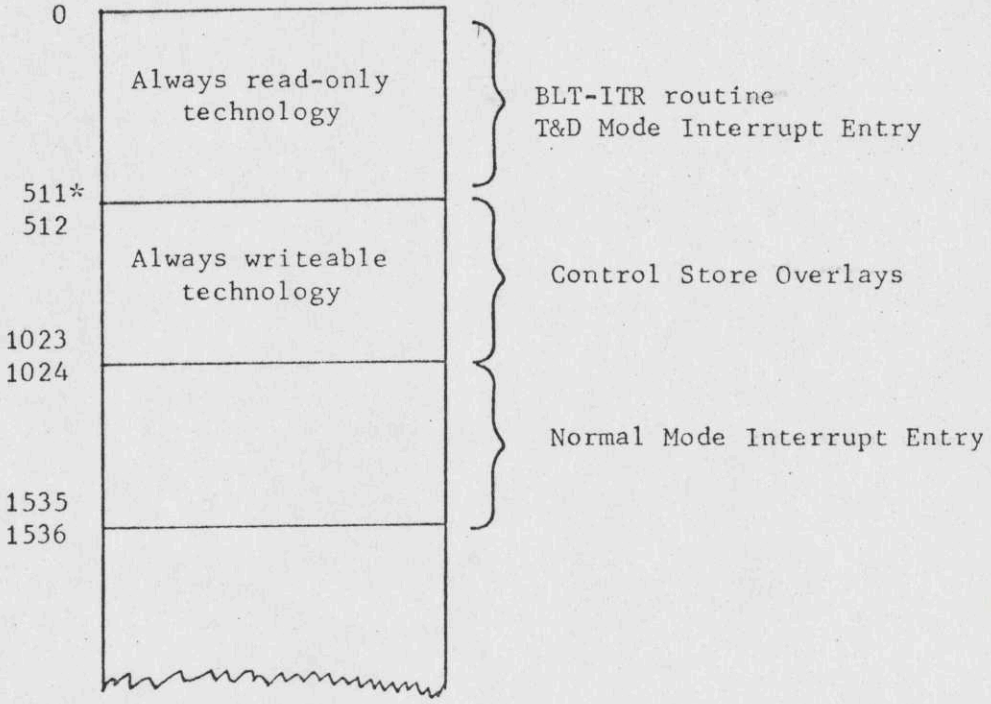
13 Main Memory Size: 0 = 1K Words
1 = 4K Words

14 Error Interrupt Safestore Bypass Control

15 Not allocated

3.5 CONTROLLER MEMORY MAPS

The controller contains two memories -- Main Memory and Control Store. The map of the Main Memory shall be supplied as a part of the design documentation. The Control Store shall be mapped as shown below:



The personality firmware which makes this MPC function as a magnetic tape controller is located beginning at address 512, and extends to the end of the Control Store.

3.5 CONTROLLER MEMORY MAPS (continued)

The Control Store Interrupt Entry locations are:

T & D Mode	:	High-level EN-1 Interrupt	256
		Low-level EN-1 Interrupt	260
		Error Interrupt	272
Normal Mode	:	High-level EN-1 Interrupt	1024
		Low-level EN-1 Interrupt	1028
		Error Interrupt	1040

* All addresses given in this section are decimal addresses.

3.6 ERROR INTERRUPT IMPLEMENTATION

The basic MPC which is being used to implement the MTC500 magnetic tape controller has built-in logic to detect internal hardware errors such as parity errors on internal registers, parity errors on Main Memory data, parity errors on microinstructions accessed from Control Store, etc. The detection of any one of these internal hardware errors will result in the automatic execution of an error interrupt, which forces the MPC to branch to a fixed Control Store location, and establishes an "error interrupt in progress" state of the machine.

The microprogram which is automatically entered as a result of the error interrupt will first test the state of configuration switch number 14 (Section 3.4). If the switch is set, an immediate branch will be made to the Integrated Test Routine (ITR) module, located in the first 512 locations of Control Store. Setting of configuration switch 14 therefore implies that the controller will be put in the ITR mode upon the occurrence of an error interrupt.

If switch 14 is not set, the error interrupt microprogram will take the following actions:

a) Safestore in a fixed Main Memory area the current contents of all pertinent hardware registers.

b) Terminate all existing activity in progress, including device movement. Note however, that a handler which is executing a rewind operation will not be stopped. The execution of an active channel program will be aborted.

c) Reset the error interrupt level, and return the controller to normal operation (i.e., waiting for a command from the external user system).

3.6 ERROR INTERRUPT IMPLEMENTATION (continued)

The occurrence of the error interrupt will force the Operational In line of the PSI to revert to the non-operational state. This line will stay in the non-operational state until the error intterrupt in progress state is reset by the microprogram. Repeated error detections will result in repeated error interrupts, and subsequent fluctuation of the Operational In line.

3.7 CONTROLLER INITIALIZE

Two types of initialization must be considered in the controller. The first is a channel initialize which occurs when the Operational Out line of the PSI reverts from the operational to the non-operational state. This will cause the controller to reset all activity associated with the physical channel over which this action occurred. All allocation of devices, table entries, etc. must be reset. Activity associated with any other physical channel is not affected.

The second type of initialize is a controller initialize which affects the entire controller. This type of initialize causes the controller to be reset, and is caused by any one of the following:

a) Actuation of the Initialize switch on the controller operator panel.

b) Initialize signal from the MPC power supply during any power-up sequence.

c) Remote initialize signal generated via the SCAM module.

d) Signal on the Reset Out line of the PSI.

Initialization by means of (a) and (b) results in the controller being d-c initialized and left in the Halt Mode, and requires actuation of the Start Switch on the operator panel or initialization by means of (c) or (d) to begin microinstruction execution. Microinstruction execution begins (upon fall of the initialize signal) at Control Store location 0, which is the location at which the Basic Logic Test module will start.

The BLT module will execute a test of the basic MPC logic, including all attached Link Adaptors. If a hardware malfunction is detected during this test the BLT module will cause the controller to halt, with fault dictionary symptoms displayed on the controller maintenance panel. Configuration Switch 0 on the maintenance panel may be set to cause the BLT module to bypass the test of the basic MPC logic. Upon completion of the basic MPC logic test (or immediately upon

3.7 CONTROLLER INITIALIZE (continued)

bypassing the test) the BLT module will accept either the Control Store Test Overlay command (11_8) or the Control Store Personality Overlay command (10_8) from the external user system. These commands are defined in Sections 5.4.24 and 5.4.23.

3.8 CODE TRANSLATION & PACK/UNPACK REQUIREMENTS

This subsystem includes the capability of code translation between selected character sets. Some of these translations which are valid only if the user has selected the particular option.

Related to these code translations is a pack/unpack activity which is necessitated in part by the PSI, and in part by the character formats. The controller must therefore be implemented to include the required translation as defined by the command, the related pack/unpack activity, and the control of the PSIA mode. The latter is the selection between the "binary mode" and the"ASCII mode" of the PSI adaptor in the IOM.

The following sections describe the translations and pack/unpack and PSIA mode for various data transfer commands to/from the handler.

3.8.1 Pack/Unpack Requirements for 7-channel Translation

The two basic modes of data transfer to/from a 7-channel handler are the Binary and the BCD modes. These commands require translation and pack/unpack as illustrated in Figure 3.8.1. It should be noted that the PSIA is in the "binary mode" for both data transfer modes.

3.8.2 Pack/Unpack Requirements for 9-channel Translation - Standard

There are three modes of data transfer to/from a 9-channel handler including the Binary, BCD,1 and Tape Nine. These are standard modes of data transfer for 9-channel tapes--the optional modes are described in section 3.8.3.

Only the BCD mode requires translation and pack/unpack as is shown in Figure 3.8.2. It should also be noted that the translation required in the BCD Read is for the single character 001010 as specified in section 8.6.4.4.

3.8.3 Pack/Unpack Requirements for 9-channel Translation - Optional

The three optional code translations offered by this subsystem require the pack/unpack activity illustrated in Figure 3.8.3. The tables defining these translations are given in sections 8.6.1 through 8.6.3.2.

Figure 3.8.1

Pack/Unpack Requirements
for
7-channel Translation

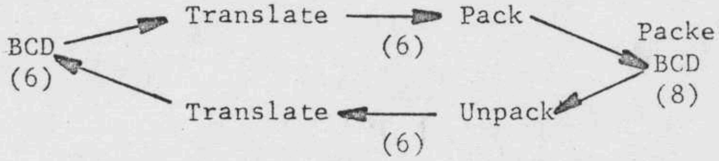
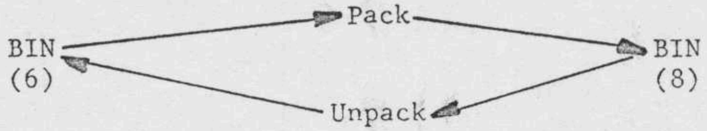
Commands	Translate & Pack/Unpack Activity	PSIA Mode
	Tape Format … PSI Format	
Read BCD Record Reread BCD Record Write BCD Record	 BCD (6) Translate (6) Pack Packed BCD (8) Unpack Translate (6)	Binary
Read Binary Reread Binary Write Binary		Binary

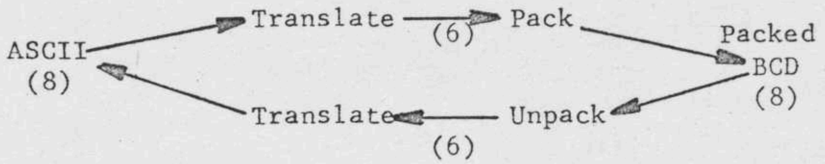
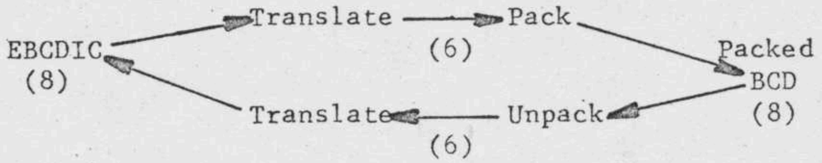
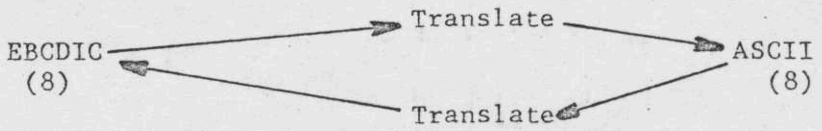
Figure 3.8.2

Pack/Unpack Requirements
for
Standard 9-channel Translation

Commands	Translate & Pack/Unpack Activity	PSIA Mode
	Tape Format → PSI Format	
Read BCD Record Reread BCD Record Write BCD Record	BCD (8) → Unpack → Translate (6) → Pack (6) → Packed BCD (8) Packed BCD (8) → BCD (8)	Binary
Read Binary Record Reread Binary Record Write Binary Record	BIN (8) ↔ BIN (8)	Binary
Read Tape Nine Write Tape Nine	ASCII (8) ↔ ASCII (8)	ASCII

Figure 3.8.3

Pack/Unpack Requirements
for
Optional 9-channel Translaction

Commands	Translate & Pack/Unpack Activity	PSIA Mode
	Tape Format PSI Format	
Read ASCII Record Write ASCII Record	 ASCII (8) Translate (6) Pack Packed BCD (8) Unpack (6) Translate	Binary
Read EBCDIC Record Write EBCDIC Record	 EBCDIC (8) Translate (6) Pack Packed BCD (8) Unpack (6) Translate	Binary
Read ASCII/EBCDIC Record Write ASCII/EBCDIC Record		ASCII

4.0 INTERFACES

4.1 PERIPHERAL SUBSYSTEM INTERFACE

The peripheral subsystem interface (PSI) is the definition of the dialog and interface circuits for the exchange of information between the EUS and the magnetic tape subsystem. This interface exists between the PSI channel of the IOM (PSIA) and the link adaptor (PSI-LA) of the magnetic tape controller. The PSI dialog is controlled on the controller end by microprograms residing in the MPC.

This dialog places the heart of the control of the peripheral operation within the MPC. For example, when a data transfer is to occur, the MPC receives an IDCW from the PSIA and interprets this instruction, places the PSIA in the proper operating mode, and initiates the data transfer. At the termination of the instruction, the MPC formats the status and initiates a status transfer to the PSIA.

4.2 MAGNETIC TAPE HANDLER DEVICE LEVEL INTERFACE

This interface defines the dialog and interface circuitry involved in the transfer of information between the device switch and the handler. It is an all-digital, radial interface which has been optimized for PE operation. Inherent in its definition is the requirement that PE deskew occur in the TCA, and that the NRZI deskew is accomplished in the handler. See the specification titled NPL Device Level Interface (DLI) for Magnetic Tape Subsystems.

4.3 INTERNAL ADAPTOR INTERFACE

This interface is internal to the controller, and is implemented on the back-panel of the card cage. It provides a hardware and firmware interface between the MPC and the adaptors (link adaptors and controller adaptors).

4.4 EFFECTS OF PSI ON SIX-BIT CHARACTER MODE

The PSI presents some unique situations when working with an external user system which is oriented toward the six-bit character mode. This is due to the mis-match between the six-bit EUS and the eight-bit PSI. These situations are detailed in the following two sections.

4.4.1 Effects of PSI on 7-channel Operation

When operating with 7-channel tapes the controller and IOS must take special precautions to ensure that only valid data is written

4.4.1 Effects of PSI on 7-channel Operation (continued)

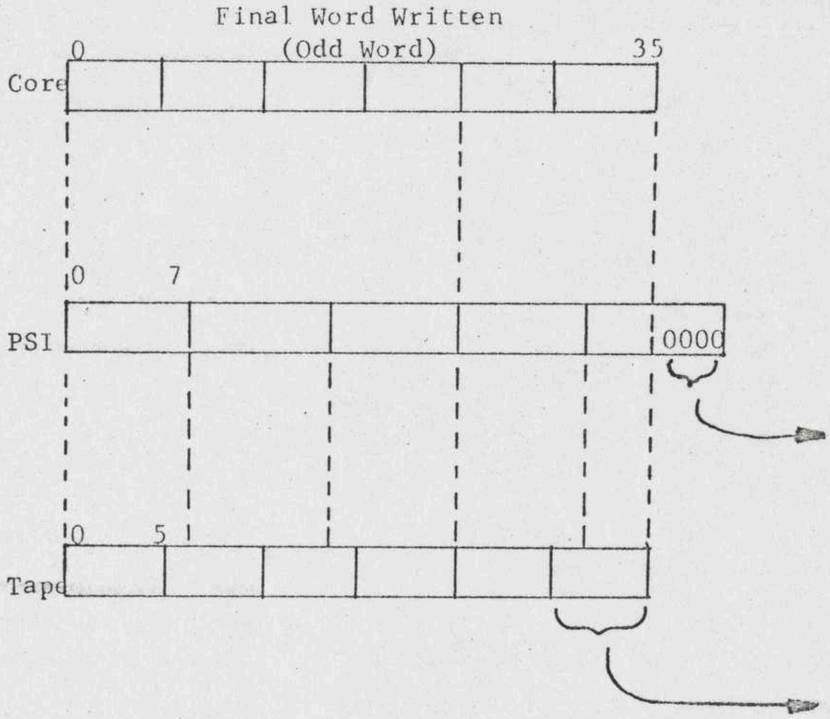
on tape, and later read from tape.

Write mode - In the write mode the controller has the responsibility of writing only valid six-bit characters on tape. As illustrated in Figure 4.4.1.1, the 6000 systems write multiples of 36-bit words. If an odd number of words are written, the final byte transferred to the controller will contain four ZEROs padded by the PSIA. The controller must strip off these four bits, thus writing a valid final character on tape.

Read mode - The situation in the read mode is more complicated because the tape may have been generated by a stranger system. In this case the final character on tape may correspond to any six-bit character position in the 36-bit 6000 word. Depending on the particular position of this final character in relation to the packing of previous characters into bytes for transmission across the PSI, the controller will pad 6, 4, 2, or 0 ZEROs on the final character.

The status supplied by the controller at termination will indicate the amount of ZERO-fill on the final character transferred, along with the terminate character position of the final byte transferred to the EUS. It is then the responsibility of IOS to map this terminate character position to the terminate character position as seen by subsystems operating on the former Common Peripheral Interface systems.

The relationship between characters on tape, bytes on the PSI, and words in 6000 core is illustrated in Figure 4.4.1.2. Also, Figure 4.4.1.3 defines the ZERO-fill performed by the controller, and the mapping required of IOS.



Four ZERO bits added by PSIA during write mode. Must be stripped off by controller before writing final character on tape.

Final character written on tape.

Figure 4.4.1.1

7-channel Write Operation for

Odd Number of Words Written

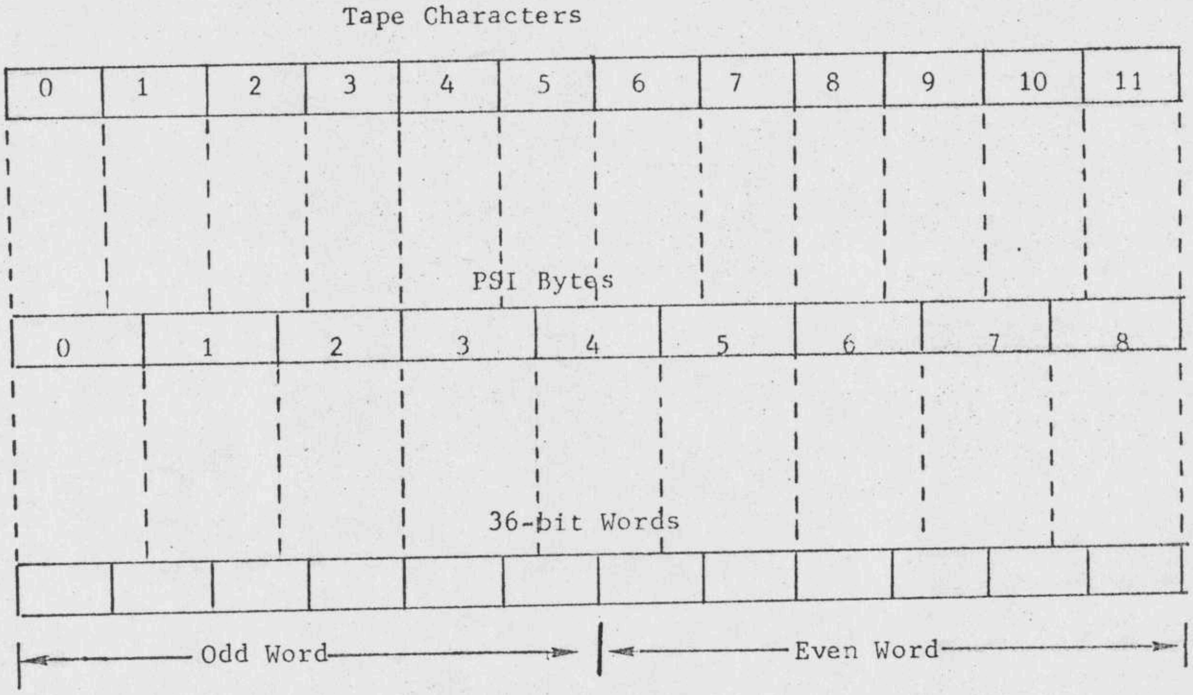


Figure 4.4.1.2

Relationship Between Tape Characters,
PSI Bytes, and Core Words for 7-channel
Tape

The mapping required of the IOS is defined by the following table.

	INFO. for IOS			OUTPUT of IOS			
Final Char. Read	Zero-fill Size[1]	Terminate Char.[2]		Word Count Residue	Terminate Char.[3]		Substatus
(No.)	(No.bits)	TCP	E/O		TCP	E/O	
0	2	2	1	Actual	1	0	00000X
1	4	3	1	"	2	0	"
2	6	4	1	"	3	0	"
3	0	4	1	"	4	0	"
4	2	0	1	"	5	0	"
5	4	1	0	Actual +1	0	0	"
6	6	2	0	Actual	1	0	"
7	0	2	0	"	2	0	"
8	2	4	0	"	3	0	"
9	4	5	0	"	4	0	"
10	6	0	0	"	5	0	"
11	0	0	0	"	0	0	"

[1] The formatting of the substatus under the Ready major status, for each of the Zero-fill conditions, is defined in section 6.5.

[2] The definition of Terminate Character Position for PSI channels is given in 43A177880.

[3] The definition of Terminate Character Position for CPI channels is given in 43A219605.

Figure 4.4.1.3

Status Mapping

for

7-Channel Read Operation

4.4.2 Effects of PSI on 9-channel Operation

When operating with 9-channel handlers the data transferred across the PSI corresponds directly to frames on tape for those operations requiring the PSIA to be in the binary mode. However, there are particular situations which occur, and are of such nature that the user data could be affected.

Write mode - Two situations must be considered in the 9-channel write mode. The first arises when performing write operations with the PSIA in the binary mode, which is the mode of the PSIA for most non-ASCII operations (i.e., most non-Tape Nine operations as shown in Figure 3.8.2 and 3.8.3). In this case, a write operation with an odd number of 36-bit words transferred will result in a final byte with four high-order ZEROs. This final byte will be written on tape, and when read back will extend beyond the expected word count by the four ZERO bits as shown in Figure 4.4.2.1.

When performing code translation from the six-bit to the eight-bit representation during a write operation as shown in Figure 3.8.3, the final unpacked character may be the result of a ZERO-fill in the last byte transferred across the PSI. These four ZERO bits are padded with two ZEROs by the controller, translated, and written on tape. When reading, this last invalid frame will be transferred to the EUS as part of the record. See Figure 4.4.2.2.

Read mode - If a 9-channel tape is read in a mode which does not require any packing/unpacking activity by the controller, the data contained in the bytes on the PSI should be completely valid. However, if any packing/unpacking is required, as would be the case for translation to an EUS six-bit code, the final byte transferred via the PSI may contain some invalid ZERO bits as illustrated in Figure 4.4.2.2. This final byte will result in the transfer to core of an extra word, zero-filled, whose first four bits result from the last four bits of the last byte on tape. This problem has existed since the orignal implementation of 9-channel tapes on the 600 Series. Software (GEFRC) has provided a test for correcting the residue, based on the assumption that for all 600/6000 Series created tape records of odd length, 4.4.2 Effects of PSI on 9-channel Operation (continued)

the four bits in question will be zero bits. However, any "stranger" created 9-channel tape containing valid data in its last byte (of which the last four bits are zero), meeting the above criteria, will suffer the loss of the last four bits of the final byte.

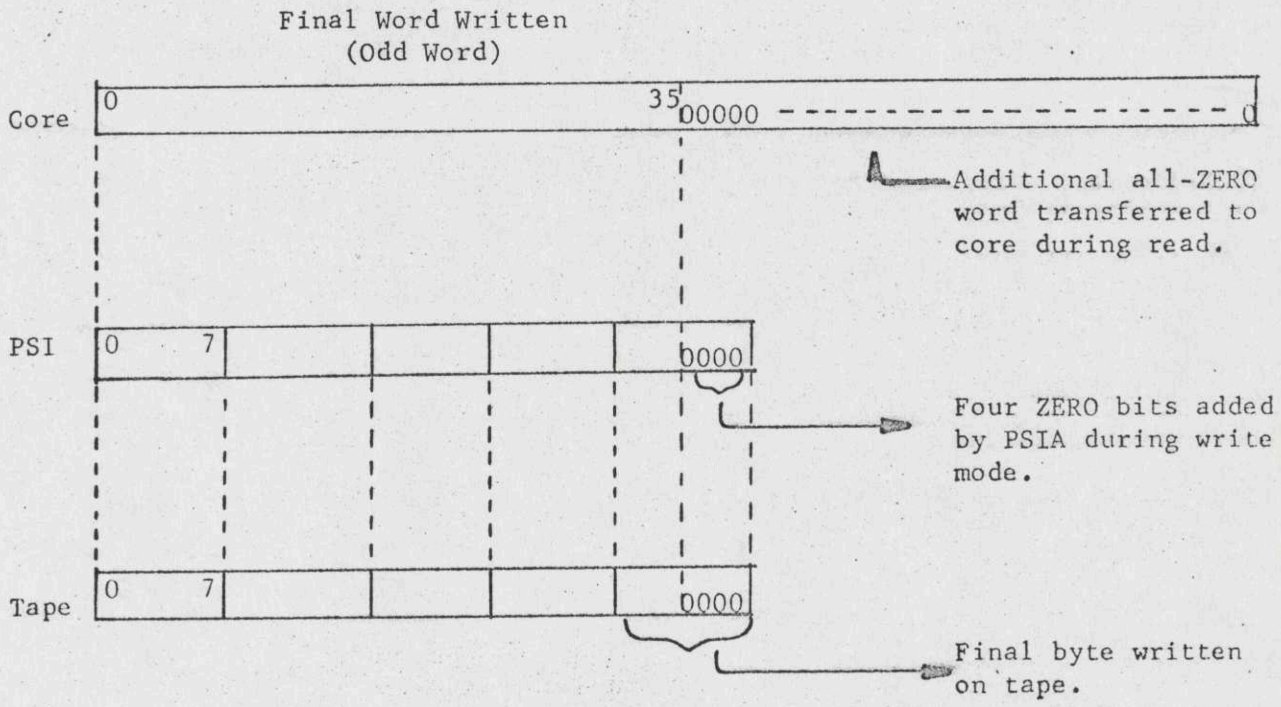

Final Word Written
(Odd Word)
Core
0
35
00000
0
Additional all-ZERO word transferred to core during read.
PSI
0
7
0000
Four ZERO bits added by PSIA during write mode.
Tape
0
7
0000
Final byte written on tape.

Figure 4.4.2.1

9-channel Binary Operations

for Odd Words Written

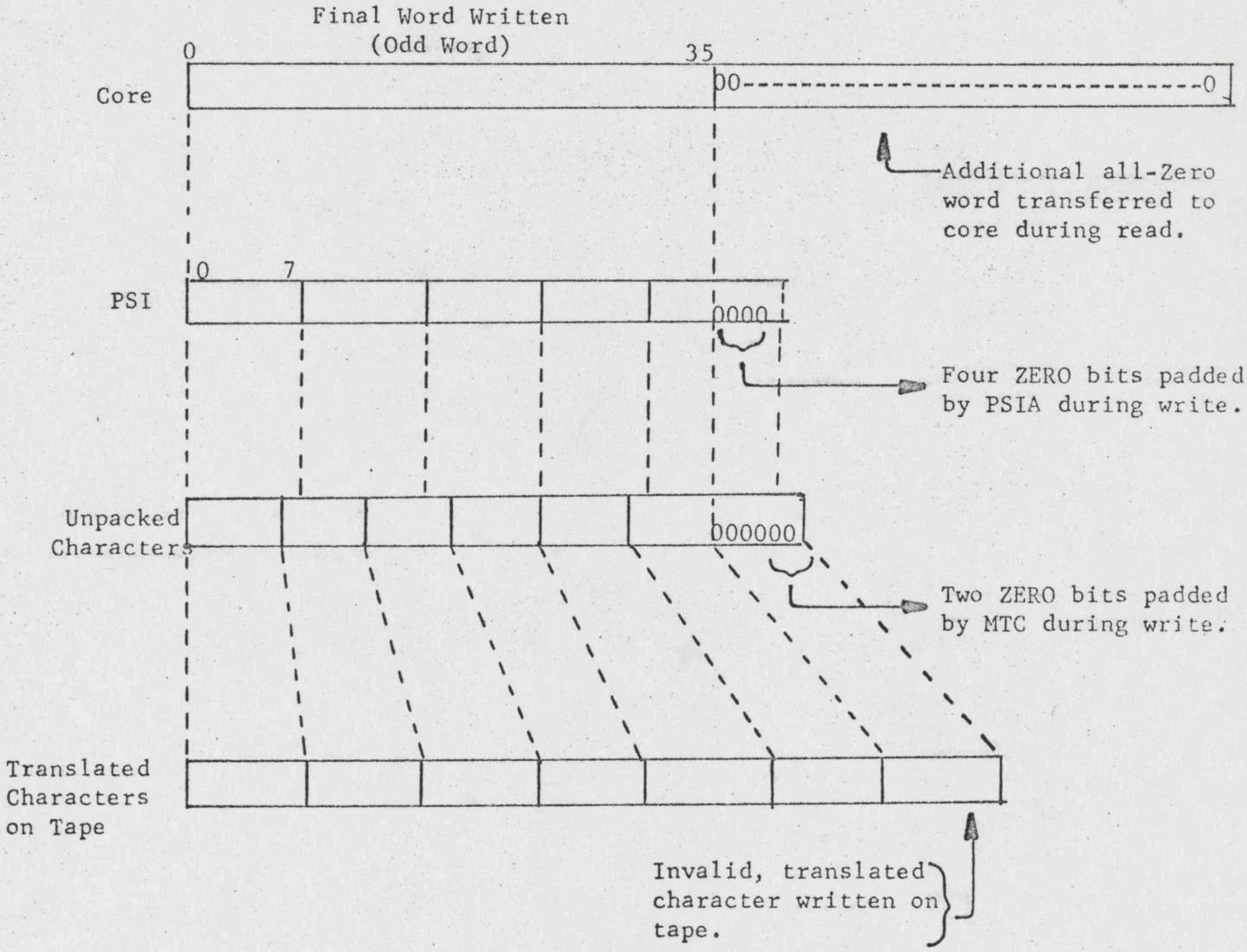

Final Word Written
(Odd Word)
0
35
Core
00----------------------------------0
Additional all-Zero word transferred to core during read.
0
7
PSI
0000
Four ZERO bits padded by PSIA during write.
Unpacked Characters
000000
Two ZERO bits padded by MTC during write.
Translated Characters on Tape
Invalid, translated character written on tape.

Figure 4.4.2.2

Illustration for ZERO-fill Effects for 9-channel Operation with Packing/ Unpacking

5.0 SOFTWARE COMMANDS

The controller shall contain microprograms for proper interpretation of command codes received from the external user system. These command codes will be received via the PSI as described in Section 5.1.

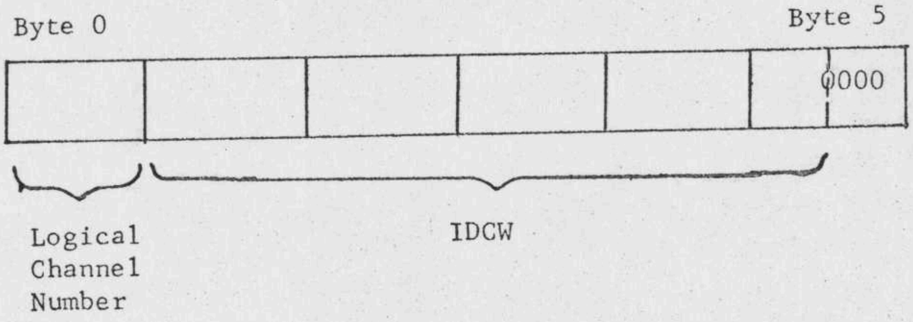
5.1 TYPICAL PSI ACTION OF CONTROLLER

From the controller point of view, a typical command execution will be as described in the following general terms. Specific details for this interface may be found in the specification for the PSI as adapted to the 6000 (43A177874) and in the specification for the PSI adapter of the IOM (43A177880).

1. PSIA raises Channel Program Waiting (CPW) line to the controller.

2. The controller responds with the service code Initiate New Channel Program.

3. The PSIA responds with six bytes of information which contain the logical channel number in the first byte, followed by five bytes containing the IDCW in its standard format. A Terminate Out (TMO) signal will accompany the final byte. Details of the IDCW format will be found in Section 5.2.



4. The legality of various IDCW fields are checked as described in Section 5.2.

5. The controller interprets the command code, and takes appropriate action.

6. The data is transferred by appropriate applications of Strobe Out (STO) and Strobe In (STI) until the operation is terminated either by Terminate Out (TMO) or by Terminate In (TMI).

5.1 TYPICAL PSI ACTION OF CONTROLLER (continued)

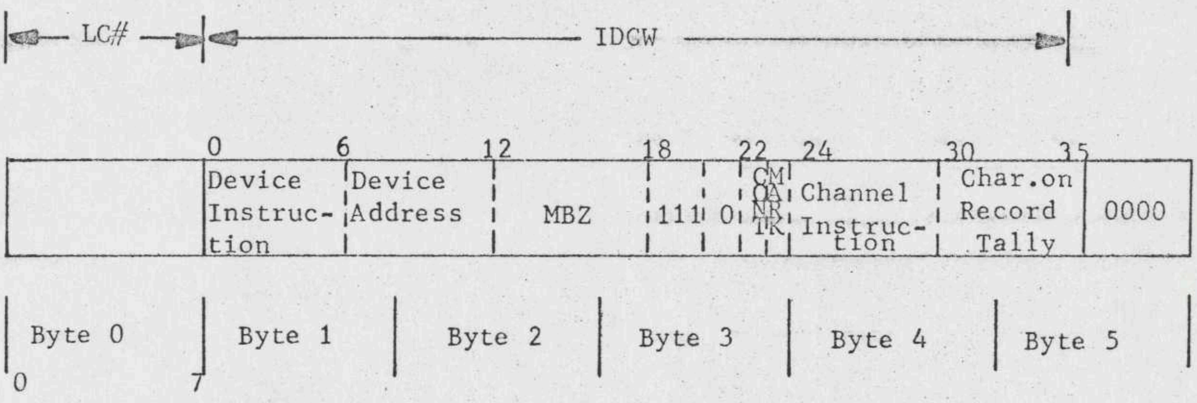
7. The controller will send the service code Store Terminate Status to the PSIA. The five status bytes will contain the major status and substatus as required by the standard 6000 status word, plus additional status information. See Section 6.0.

8. The controller will send the service code Move Pointer and Initiate Command Transfer, if required.

9. If step 8 above occurs, the sequence will start again at step 3.

5.2 REQUIREMENTS FOR CONTROLLER CHECKING OF IDCW

The controller will receive an IDCW from the external user system in the following format.



Bits	Description
0:0-0:7*	Logical Channel Number

Valid logical channel numbers are represented by 00000XXX, where the X's may be ZERO or ONE.

1:0-1:5	Device Instruction (Op. Code)

The codes found here are those of Section 5.3 and Section 5.5. If an illegal code is detected, the error is reflected to the external user system via the terminate status as described in Section 6.5.

* 0:7 defines byte and bit (e.g., 0:7 = bit 7 of byte 0)

5.2 REQUIREMENTS FOR CONTROLLER CHECKING OF IDCW (continued)

Bits	Description
1:6-2:3	Device Address

The legal codes are defined in Section 3.2. This field of the IDCW is valid only for the first IDCW in the channel program, and shall be ignored for all IDCWs other than the first in the program.

3:2-3:4 Must be 111_2.

If this field is not correct, the controller will reflect Channel Status of 011_2 in the termination status as defined in Section 6.2.

3:5 Ignored by the controller.

3:6-3:7 Continue and Marker Bits

These two bits define the action to be taken by the controller upon completion of the operation:

Cont	Mark	
0	X	The controller stores terminate status and causes a terminate interrupt at the end of this IDCW execution. This IDCW is the last in the list. The state of the mark bit is ignored by the controller
1	0	This is not last IDCW in list. Upon completion of execution of this IDCW with no faults detected, the controller will issue a "Move Command Pointer" service code and obtain a new IDCW.
1	1	Upon completion of the execution of the IDCW, with ready status and no faults detected, the controller will store terminate status and cause a marker interrupt (This means that the technique of Section 6.1 is used to store terminate status except that the "Set Marker Interrupt" service code replaces the "Set Terminate Interrupt" service code), and then issue a "Move Command Pointer" service code to obtain a new IDCW.

5.2 REQUIREMENTS FOR CONTROLLER CHECKING OF IDCW (continued)

Bits	Description
4:0-4:5	Channel Instruction

This field has a two-fold purpose. First, it defines whether the command is to be executed as a Special Command (Section 5.5) or as a Regular Command (Section 5.3). Second, it defines the type of transaction which is to take place. This field must contain one of the following configurations:

For Regular Commands;

00_8	Unit record transfer
02_8	Peripheral action (non-data transfer; e.g., request status, backspace N records, set density)
06_8	Multi-record instruction
10_8	Single character record (e.g., write file mark as distinguished from EOF, tape mark, where the single character is contained in the tally field (4:6-5:3) of the IDCW Additional Codes for Regular Data Transfer Commands

Octal Code	Turn Off Automatic Retry & In-flight Error Correction	Set Capstain		Set Low Threshold
		Slow	Fast	
22	No	No	No	No
21	No	No	No	Yes
22	No	No	Yes	No
23	No	No	Yes	Yes
24	No	Yes	No	No
25	No	Yes	No	Yes
26	No	No	No	No
27	No	No	No	Yes
30	Yes	No	No	No
31	Yes	No	No	Yes
32	Yes	No	Yes	No
33	Yes	No	Yes	Yes
34	Yes	No	No	Yes
35	Yes	No	No	Yes
36	Yes	No	No	No
37	Yes	No	No	Yes

5.2 REQUIREMENTS FOR CONTROLLER CHECKING OF IDCW (continued)

for Special Controller Commands**;

40_8 Unit record transfer

42_8 Peripheral action

46_8 Multi-record instruction

50_8 Single character record

* These conditions are turned off for the duration of this current IDCW.

** Channel Instruction fields of $4X_8$ amd $5X_8$ are reserved for Special Controller commands.

All other configurations of this field are illegal, and will result in the controller generating a Channel Status of 010_2 in the termination status (Section 6.4).

4:6-5:3 Character of Record Tally

This field contains the character for a single character record (Channel Instruction field = 10_8 or 50_8), or the number of times the device instruction is to be re-issued by the controller (Channel Instruction field = 02_8, 06_8, 42_8 or 46_8). In the magnetic tape controller application multiple record peripheral action commands are legal only for the Forward Space One Record and Backspace One Record commands. No specific checking of this field is required of the controller.

5:4-5:7 Must be zero.

5.3 PERMISSABLE COMMANDS

The IDCW description given in section 5.2 indicates that the Channel Instruction field determines if the Device Instruction is to be interpreted as a Regular command, or as a Special Controller command. The former are summarized in the following list; the latter are summarized in Section 5.5.

Command	Octal Representation	Section
Request Status	00	5.4.1
Reset Status	40	5.4.2
Request Device Status	50	5.4.3
Reset Device Status	51	5.4.4
Survey Devices	57	5.4.5
Read Control Registers	26	5.4.6
Write Control Registers	16	5.4.7
Set File Protect	62	5.4.8
Set File Permit	63	5.4.9
Rewind	70	5.4.10
Tape Load	75	5.4.11
Rewind/Unload	72	5.4.12
Reserve Device	66	5.4.13
Release Device	67	5.4.14
Set 200 BPI	64	5.4.15
Set 556 BPI	61	5.4.16
Set 800 BPI	60	5.4.17
Set 1600 CPI	65	5.4.18
Forward Space One Record	44	5.4.19
Forward Space One File	45	5.4.20
Backspace One Record	46	5.4.21
Backspace One File	47	5.4.22
Control Store Overlay	10	5.4.23
Load From Device	05	5.4.24
Erase	54	5.4.25
Write End-of-File Record	55	5.4.26
Write Tape Nine	13	5.4.27
Read Tape Nine	03	5.4.28

5.3 PERMISSABLE COMMANDS (continued)

Command	Octal Representation	Section
Write Binary Record	15	5.4.29
Read Binary Record	05	5.4.30
Reread Binary Record	07	5.4.31
Write BCD Record	14	5.4.32
Read BCD Record	04	5.4.33
Reread BCD Record	06	5.4.34
Write EBCDIC Record	34	5.4.35
Read EBCDIC Record	24	5.4.36
Write ASCII Record	37	5.4.37
Read ASCII Record	27	5.4.38
Write ASCII/EBCDIC Record	35	5.4.39
Read ASCII/EBCDIC Record	25	5.4.40
Diagnostic Mode Control	31	5.4.41
Device Wraparound	32	5.4.42
Write Timing Character	30	5.4.43

5.4 DESCRIPTION OF TAPE COMMANDS

5.4.1 Request Status Command (00_8)

This command results in the transmission to the external user system of the status conditions of the addressed device. This status shall be formatted as terminate status (see Section 6).

Since the exception conditions are stored for each handler, this information remains independent of controller communication with other handlers. Therefore, if an exception condition is reflected to the external user system at the termination of a command execution, and the external user system then addresses a second handler, the true status for the second handler will be returned as terminate status for the current command. The exception condition of the first handler remains unchanged.

5.4.2 Reset Status (40_8)

This command results in the resetting of all resettable status conditions within the addressed device. The resettable status conditions are defined as those exception conditions included in the following major status conditions:

Device Data Alert

End of File

The status reflected to the external user system as terminate status will be the status condition of the subsystem after resetting the resettable status.

5.4.3 Request Device Status (50_8)

Extensive status information is stored in each handler. Some of this status is descriptive of handler physical characteristics, and some is descriptive of the operational characteristics of the handler. The Request Device Status command will transfer this status to the external user system.

This command is not legal if addressed to device zero, and shall be rejected with Command Reject- Invalid Device Code status. The controller will place the PSIA in the "binary mode" prior to transmitting this status to the EUS.

Twenty-four bytes of handler status will be transmitted unless the EUS terminates the operation.

The first four (4) bytes are defined in the document defining the handler interface (NPL Device Level Interface (DLI) for Magnetic Tape Subsystem). The remaining bytes will be specified in the document defining the handler.

5.4.4 Reset Device Status (51_8)

This command provides a mechanism for resetting the resettable status bits in the addressed handler. In response to this command, the controller will issue appropriate DLI commands to the addressed handler such that the status bits are reset to the ZERO state. Successful completion of this command will result in Ready status being transmitted to the EUS.

This command is not legal if addressed to Device Zero, and shall be rejected with Command Reject-Invalid Device Code status.

5.4.5 Survey Devices (57_8)

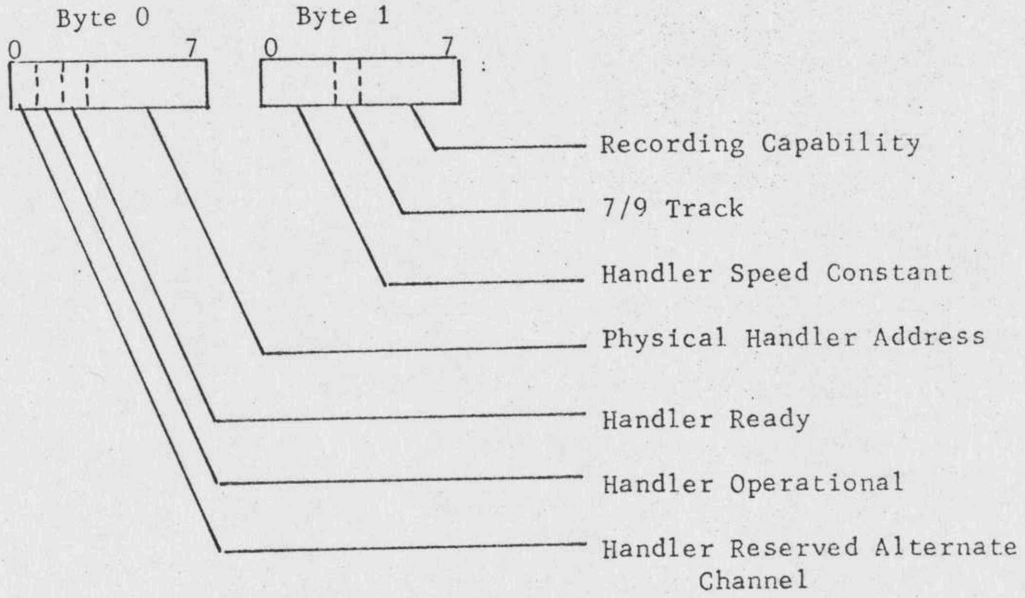
Upon receipt of this command, the subsystem will interrogate each handler, assembling two bytes of information for each handler. The survey proceeds from handler 1 through handler 8 (for single channel controller) or through handler 16 (for dual channel controller). The controller will place the PSIA in the "ASCII mode" prior to transferring the first byte to the PSIA.

This command is legal only if addressed to device zero; otherwise it will be rejected with Command Reject-Invalid Device Code.

This command does not constitute an access request of a handler as described in section 6.6.

Successful completion of this command will result in Ready termination status being sent to the EUS.

The bytes assembled by the controller for each handler will contain the following format and information:



5.4.5 Survey Devices (57_8) (continued)

Bits Description

0:0 Handler Reserved Alternate Channel

A logical ONE in this bit position indicates that the handler is reserved for use by an alternate channel as the result of a Reserve Device command previously issued from another channel. If this bit is a logical ONE all other bits in the byte pair for that handler are invalid.

0:1 Handler Operational

A logical ONE indicates that the handler is capable of communication with the external user system; logical ZERO means handler is incapable of communication with EUS.

0:2 Handler Ready

A logical ZERO in this position indicates that the handler (a) is in the process of executing a rewind-type operation or the Tape Load command, (b) the device is in the standby state, or (c) the handler is in a fail state.

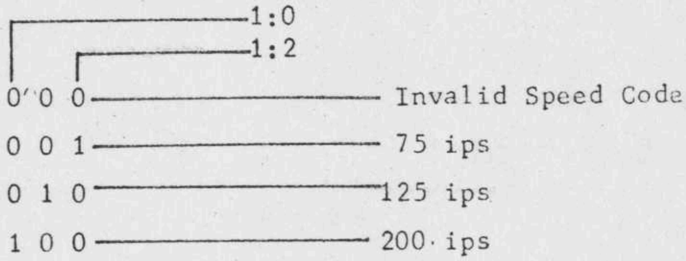
0:3-0:7 Physical Handler Address

This is the 5-bit number which is wired into the handler at subsystem installation. For further description of the significance of this number, refer to section 3.2.

The low order bit is in position 0:7.

1:0-1:2 Handler Speed Constant

This field specifies the forward speed capability of the handler.



5.4.5 Survey Devices (57_8) (continued)

Bits	Description
1:3	7/9 Track Logical ONE = 9-track handler.
1:4-1:7	Recording Capability This field describes the PE and NRZI recording capability of the handler.

```
┌──────────────── 1:4
│      ┌───────── 1:7
│      │
0 0 0 0* ─────────────── 1600 bpi

0 0 0 1 ─────────────── 200,556,800 bpi

0 1 0 0* ───────────── { 1600 bpi
                       { 200,556,800 bpi
```

*These codes are not legal if 7/9 track bit is logical ZERO.

5.4.6 Read Control Register (26_8)

This command provides a method by which the external user system may obtain the contents of error logs maintained in the controller. This command is valid only if addressed to a handler, and shall be rejected with Command Reject - Invalid Device code status if addressed to device ZERO.

When the Read Control Register command is addressed to some device other than zero, the contents of the counters related to the addressed handler will be returned to the EUS. In this case seven counts (two bytes each, plus two special bytes,) will be sent to the external user system. Prior to transmitting the first byte, the controller will place the PSIA in the "binary mode". Refer to section 7.5.2 for a description of the counters maintained by the controller for each handler in the subsystem.

5.4.7 Write Control Register (16_8)

This command provides a mechanism by which the external user system may load particular data in the handler error logs maintained in the controller. This command shall be rejected with Command Reject-Invalid Device Code status if addressed to device zero.

In response to this command, the controller will place the PSIA in the binary mode, and prepare to receive eighteen bytes of information. These bytes shall be formatted by the EUS as shown in Figure 7.5.

The controller will extract the various counts, and store in the appropriate error logs.

5.4.8 Set File Protect (62_8)

The reception of the Set File Protect command will serve to unconditionally inhibit write operations on the selected handler regardless of the presence of a Write Permit Ring in the tape reel.

The handler shall remain in the File Protect condition until a new reel is loaded, or until the Set File Permit command is issued to this handler.

5.4.9 Set File Permit (63_8)

This command will allow writing on a handler which has been protected previously by a Set File Protect command and has a Write Permit Ring in place.

If the Write Permit Ring is not in place, this command will be rejected with Command Reject - Invalid Operation Code status.

5.4.10 Rewind (70_8)

This command results in the tape on the selected tape unit being positioned on the Beginning of Tape (BOT) marker. The tape subsystem is free to accept commands directed to other tape units during the rewind execution. If any command other than a Rewind is directed to a tape unit which is executing a rewind operation, the command will be rejected with the Device Busy - In Rewind status. If a Rewind command is directed to a tape unit which is executing a rewind operation, the Peripheral Subsystem Ready status will be reflected.

5.4.10 Rewind (70_8) (continued)

Upon completion of this command, a special interrupt is sent to the EUS by using the technique of Section 6.6.

If at the time the Rewind command is received, the tape mounted on the specific tape unit is already positioned on BOT, the subsystem will reflect the Peripheral Subsystem Ready status to the EUS with the terminate status mode (section 6.1).

Prior to terminating this command the controller shall interrogate the "In Rewind" bit of the addressed handler to verify that the handler was set to the desired state. (This should take in the order of a few microseconds).

5.4.11 Tape Load (75_8)

This command causes the tape mounted on the supply reel of the selected tape unit to be loaded automatically, and positioned at the beginning of Tape (BOT) marker. The status reflected will be Peripheral Subsystem Ready. The subsystem is free to accept commands directed to other tape units during the load operation. At the termination of the load operation special status will be sent to the EUS per section 6.6.

This command is valid only if the handler is in the Standby-unloaded state, which is the state at the end of a Rewind/Unload command.

If, at the time this command is received in the subsystem, the tape is already positioned at the BOT marker, the controller will send the terminate status Peripheral Subsystem Ready - Positioned at BOT to the EUS.

5.4.12 Rewind/Unload (72_8)

This command results in the tape on the selected tape unit being run back on the supply reel. The tape will stop in such a position that the tape reel may be readily dismounted from the tape unit.

Status reflected to the EUS during the reception of the Rewind/Unload command will be Peripheral Subsystem Ready, provided the command is accepted. All subsequent commands will be rejected with the Device Attention - Standby status, except Tape Load and Request Stat

At termination of the unload operation, the controller will send special status to the EUS by using the method of section 6.6.

5.4.13 Reserve Device (66_8)

This command permits an EUS to reserve an unreserved device for its exclusive use. This causes a device to be reserved to the link adapter port by which this command was received. This reserve condition may be erased by a Release Device command received via the same link adapter port which originally received the Reserve Device command.

When any other EUS (i.e., any other link adapter port) attempts to address a reserved device, the status Device Busy - Device Reserved will be returned as terminate status. The reserved state of the handler is maintained in tables within the controller.

5.4.14 Release Device (67_8)

This command, when received via a link adapter port by which a Reserve Device command directed to the same device had been received, will cause the device reserve condition to be erased. The device is then available for use by any EUS.

When this command has been completed successfully, a special status/interrupt will be transmitted to all as defined in Section 6.6.

5.4.15 Set 200 BPI (64_8)

The reception of this command by the subsystem will cause the addressed handler to be set to 200 bpi, if currently in a NRZI mode, or if at the BOT marker and capable of this density. The addressed tape unit will remain in this density until modified by one of the other set density commands.

If this command is directed to a handler which is not capable of 200 bpi operation, the command shall be rejected with Command Reject-Invalid Op Code.

If this command is directed to a handler which is currently set to 1600 bpi, and the tape is not at the BOT marker, this command will be rejected with Command Reject-Invalid Op Code.

5.4.16 Set 556 BPI (61_8)

The reception of this command by the subsystem will cause the addressed handler to be set to 556 bpi, if currently in an NRZI mode, or if at the BOT marker and capable of this density. The addressed tape unit will remain in this density until modified by one of the other set density commands.

5.4.16 Set 556 BPI (61_8) (continued)

If this command is directed to a handler which is not capable of 556 bpi operation, the command shall be rejected with Command Reject-Invalid Op Code. If this command is directed to a handler which is currently set to 1600 bpi, and the tape is not at the BOT marker, this command will be rejected with Command Reject-Invalid Op Code.

5.4.17 Set 800 BPI (60_8)

The reception of this command by the subsystem will cause the addressed handler to be set to 800 bpi, if currently in a NRZI mode, or if at the BOT marker and capable of this density. The addressed tape unit will remain in this density until modified by one of the other set density commands.

If this command is directed to a handler which is not capable of 800 bpi operation, the command shall be rejected with Command Reject-Invalid Op Code. If this command is directed to a handler which is currently set to 1600 bpi, and the tape is not at the BOT marker, this command will be rejected with Command Reject-Invalid Op Code.

5.4.18 Set 1600 BPI (65_8)

The reception of this command by the subsystem will cause the addressed handler to be set to 1600 bpi, if at the BOT marker and capable of this density. The addressed tape unit will remain in this density until modified by one of the other set density commands (provided the next set density command is issued while at BOT marker).

If this command is directed to a handler which is not capable of 1600 bpi operation, the command shall be rejected with Command Reject-Invalid Op Code. If this command is directed to a handler which is currently set to a NRZI density, and the tape is not at the BOT marker, this command will be rejected with Command Reject-Invalid Op Code.

5.4.19 Forward Space One Record (44_8)

This command results in the tape mounted on the specified handler being moved forward over the number of records specified by the record tally field of the IDCW, and positioned to read or write the next sequential record. There is no data transfer between the tape subsystem and the external user system during the execution of this command, and tape parity is not checked during the command execution.

5.4.19 Forward Space One Record (44_8) (continued)

The tape subsystem remains busy during the execution of the command. Upon completion of the execution of this command, the terminate status is sent to the external user system. If a record spaced over was an End of File record, the status reflected to the external user system will be the End of File status, the operation will terminate, and the residue will reflect the unfulfilled record tally.

5.4.20 Forward Space One File (45_8)

This command results in the tape mounted on the specified handler being moved forward until an End of File record is detected. Tape is positioned to read or write over the record following the detected End of File record. There is no data transfer between the tape subsystem and the external user system during the execution of this command, and tape parity of the data records is not checked during the command execution. The controller will ignore the EOT marker. The tape subsystem remains busy during the execution of this command. Upon the completion of the execution of the command, the End of File terminate status is sent to the external user system.

5.4.21 Backspace One Record (46_8)

This command results in the tape mounted on the specified handler being moved backward over the number of records defined by the record tally field of the IDCW, and positioned to read or write the record just spaced over. There is no data transfer between the tape subsystem and the external user system during the execution of this command, and tape parity for data records is not checked during the command execution.

The tape subsystem remains busy during the execution of this command. Upon the completion of the execution of this command, appropriate terminate status is sent to the external user system. If a record just spaced over was an End of File record, the End of File terminate status will be sent, the operation will terminate, and the residue will reflect the unfulfilled record tally. If, at the time this command is received, the tape mounted on the specific tape unit is positioned at the BOT marker, the tape subsystem will not go busy, but will reflect the Command Reject-Positioned on Leader status to the external user system during the command sequence.

5.4.22 Backspace One File (47_8)

This command results in the tape mounted on the specified handler being moved backward until an End of File record is detected. Tape will be positioned to read or write over the End of File record just spaced over. There is no data transfer between the tape subsystem and the external user system during the execution of this command, and tape parity of data records is not checked during the command execution.

The tape subsystem remains busy during the execution of this command. Upon completion of execution of this command, the End of File terminate status is sent to the external user system.

If, at the time this command is received, the tape mounted on the specific tape unit is positioned at the BOT marker, the tape subsystem will not go busy, but will reflect the Command Reject - positioned at BOT status to the external user system.

If, during the execution of this command, no End of File record is encountered by the time the beginning of tape marker is detected, tape will be halted at the beginning of tape position, and the Peripheral Subsystem Ready - Positioned at BOT status reflected to the external user system.

5.4.23 Control Store Overlay (10_8)

This command is used for loading either Controller Personality or Integrated Test Routine (ITR) microprograms into MPC Control Store. The overlay operation is begun at Control Store location 512, and continues until terminated by the external system. This command is executed by the BLT routine.

A "sum-check" word (two-bytes wide) will follow and be included as a part of the overlay data. This sum check word will represent the twos-complement of the result of the binary addition of all two-byte overlay words received, with end-around carry. The data must be formatted in the ASCII mode.

If the Control Store Overlay command is executed error-free, the Controller will reflect Ready Major Status. Following execution of the Termination interrupt the "WAIT" routine will branch to location 514.

Any error detected during the reception and execution of the Control Store Overlay command will be reflected by the Controller reverting to the "halt" state, which will in turn cause the Operational-In line to the IOM to drop.

If the Control Store overlay was an ITR microprogram, the Operation-In line to the IOM will be dropped immediately upon the start of execution of the ITR overlay. The execution of the ITR overlay may result in the halting of the MPC Controller (fault detected), in which case the Operational-in line to the IOM will remain down.

5.4.23 Control Store Overlay (10_8) continued

If the ITR overlay execution completes without a fault detection, the MPC Controller will raise the Operational-In line to the IOM, and will generate a Special Status storage followed by a Special Interrupt to the external system as indicated in section 6.6.1.

At this point the MPC controller will be idling in the "WAIT" routine, waiting for the next command from the external system.

5.4.24 Load From Device (05_8)

This command will occur only immediately following a controller initialize resulting from activation of the IOM boot button or in response to a PCW generated by software. The controller is then in the "WAIT" mode.

Upon reception of this command the "WAIT" microprogram will branch to Control Store location 1536 (decimal). The microprogram being branched to shall be a non-destructive read of one record from tape. The record read will be the next record encountered on tape, subject to the blank-tape-on-read constraint. The data shall be formatted as a binary record.

Note that the Operation Code is the same as for the Read Binary Record command. However, since the Load From Device command is entered only via the BLT "WAIT" microprogram, and since the Read Binary Record command is not valid when in the BLT routines, these commands will be unique.

When the "WAIT" routine receives this command, the IDCW shall be placed in the controller RBA registers as follows:

IDCW Bits (Byte:Bits)	RBA Register (Reg. No. : Bits)
0:0 - 0:7	0:0 through 0:7
1:0 - 2:7	10:0 through 11:7
3:0 - 4:7	12:0 through 13:7

The Device Address field of the IDCW must be all ZEROs. The microprogram at location 1536 shall interrogate controller configuration switches 5 - 7 to determine the device number and the recording mode (NRZI or PE). This command will read the following combinations of tape densities and tracks:

BPI	MODE	TRACKS
800	NRZI	7
800	NRZI	9
1600	PE	9

5.4.24 Load From Device (05_8) continued

The 7 track/9 track decision is made by the controller reading the appropriate status stored in the addressed handler.

Upon completion of execution, the microprogram branched to at 1536 will set RBA register zero to all-ZEROs, and return to the "WAIT" routine at location 346 (decimal).

5.4.25 Erase (54_8)

This command results in the tape on the specified handler being erased in the forward direction for a nominal distance of 8.5 inches measured from its position at the time the Erase command was received. D-C erase shall be used and the magnetic polarity shall be the same as the gap polarity.

The tape subsystem remains busy during the execution of the command. Upon the completion of the execution of the command, appropriate terminate status is sent to the external user system. If the End of Tape mark is detected during the execution of the Erase command, The End of Tape Data Alert status is reflected to the external user system at the completion of the command.

During the execution of the erase operation, a read check will be made on the portion of tape erased. If a bit is detected in the erased portion of the tape, the status reflected at the termination of the command execution will be Device Data Alert - Bit Detected Curing Erase Operation. Note that approximately 18 bits must be present in PE recording before any will be detected.

5.4.26 Write End-of-File Record (55_8)

This command shall result in the writing of an End-of-File Record. No data transfer shall occur between the processing system and the tape subsystem. The status reflected to the processing system, after the successful completion of this command, shall be the Peripheral Subsystem Ready status. It should be noted that a Read Check shall be made on the erased portion of tape during the generation of the End-of-File Record. If a bit is detected in the erased portion of the tape, the status reflected at the termination of the command execution shall be the Bit Detected During Erase Operation - Data Alert status.

a) End-of-File - 7 Channel. The End-of-File character for a 7-channel unit shall be an octal 17, even parity. See section 8.2.5.

The tape shall be erased nominally 3.75 inches, then the Octal 17 character written, followed by the Longitudinal-Check character. The erased portion of tape shall be monitored for bits and if any are detected, an Error Status shall be reflected to the processing system.

b) End-of-File - 9 Channel NRZI. The write End-of-File command to a 9-channel tape unit shall result in the single tape character octa 23 being written. This single-frame record shall be written in the normal End-of-Record format with an 0.6 inch gap (Section 8.3.5). The LCC at 800 bpi maintains its location of seven characters between the last data character and the LCC. The CCC is forced to be all zeros (Section 8.4.5).

c) End-of-File - PE Mode. The Write End-of-File command issued to a PE handler shall record an octal 23 written in the format defined in section 8.5.5.

5.4.27 Write Tape Nine (13_8)

This command causes the controller-IOM combination to convert 9-bit bytes from memory to 8-bit frames to be written on tape

5.4.27 Write Tape Nine (13_8) continued

by causing the PSIA to ignore the first bit of each 4-bit byte. The conversion from 9-bits to 8-bits takes place in the PSI channel of the IOM. The controller will accept the 8-bit bytes and write them on tape as if they were binary data. When the controller receives the Write Tape Nine command from the IOM, the controller sends the service code "Initiate Data Transfer, Write ASCII" to the PSIA. The PSIA will then select the data bytes so that they are in the "ASCII format".

This command will be rejected with Command Reject - Nine Track Error status if issued to a seven-track tape handler.

5.4.28 Read Tape Nine (03_8)

This command has the effect of converting 8-bit frames read from tape into 9-bit bytes embedded in the 36-bit 6000 word (bits 0, 9, 18, 27 are set to Zero). From the controller standpoint, the data read from tape is transferred as binary data to the IOM. The conversion to 9-bit bytes takes place in the PSI channel of the IOM while it is in the "ASCII mode".

In response to the Read Tape Nine command, and prior to sending data to the IOM, the controller sends the service code "Initiate Data Transfer, Read ASCII" to the IOM. The IOM/PSIA will then format the succeeding data bytes in the "ASCII format".

This command will be rejected with Command Reject - Nine Track Error status if issued to a seven-track tape handler.

5.4.29 Write Binary Record (15_8)

This command results in the writing of one binary record as defined below:

a) Seven Track Handler - With a seven track handler, the packed eight-bit bytes from the EUS are unpacked to six-bit characters, and then written without code conversion.

b) Nine Track Handler - With these handlers, the record is written as binary information without code conversion.

5.4.30 Read Binary Record (05_8)

This command results in the reading of one binary record.

a) Seven Track Handler - The six-bit characters read from tape are packed into eight-bit bytes and transferred to the EUS. No code conversion is performed.

5.4.30 Read Binary Record (05_8) continued

b) Nine Track Handler - The data read from tape is passed to the EUS without packing or code conversion.

5.4.31 Reread Binary Record (07_8)

The execution of this command shall be the same as for the Read Binary Record (05_8) command, with the exception that the controller will set the handler in the low threshold state prior to starting the tape motion.

5.4.32 Write BCD Record (14_8)

This command results in the writing of one BCD record as follows:

a) Seven Track Handler - The eight-bit bytes received from the IOM via the PSI are unpacked to six-bit characters. Then these six-bit characters are converted, as defined in section 8.6.4.1 and written on tape.

b) Nine Track Handler - This command results in the writing of the eight bit bytes received from the EUS as binary data on the tape without code conversion.

5.4.33 Read BCD Record (04_8)

This command results in the reading of one BCD record.

a) Seven Track Handler - The six-bit character read from tape goes through a two step process. The first step is code conversion as defined in section 8.6.4.2, which converts the six-bit characters for ultimate storage in the EUS. The second step is the packing of these converted six-bit characters into eight-bit bytes for transmission to the IOM.

b) Nine Track Handler - When issued to a nine track handler, this command causes three step processing of the 8-bit bytes read from tape. The first step is the unpacking into six-bit characters, followed by a conversion on a single character as defined in section 8.6.4.4. The resulting six-bit characters are then repacked into eight-bit bytes for transfer to the IOM.

5.4.34 Reread BCD Record (06_8)

The execution of this command shall be the same as for the Read BCD Record (04_8) command, except that the handler is set to the low threshold mode by the controller prior to initiation of tape movement.

5.4.35 Write EBCDIC Record (34_8)

This command results in the writing of one record with translation from the 6000 line six-bit character format to the EBCDIC format. The translation table for this code translation is given in section 8.6.2.1. If this command is issued to a controller which does not contain this option, the command shall be rejected with Command Reject - Invalid Operation Code status.

a) Seven Track Handler - This command is illegal if issued to a seven track handler and must be rejected with Command Reject - Nine Track Error status.

b) Nine Track Handler - With a nine track handler this command causes the translation of six-bit 6000 line characters into eight-bit EBCDIC bytes, and the subsequent writing of these bytes on tape.

In executing this command, the controller will place the PSIA in the "binary mode". Then the eight-bit bytes received from the PSIA are unpacked to six-bit characters, and each six-bit character translated to the corresponding eight-bit byte as defined by the translation table. The resulting eight-bit bytes are then written on tape. The unpacking requirements for this code translation are illustrated in Figure 5.4.

5.4.36 Read EBCDIC Record (24_8)

This command results in the reading of one record with translation to the six-bit character format. If this command is issued to a controller which does not contain this option, this command shall be rejected with Command Reject - Invalid operation Code status.

a) Seven Track Handler - This command is illegal if issued to a seven track handler and must be rejected with Command Reject - Nine Track Error status.

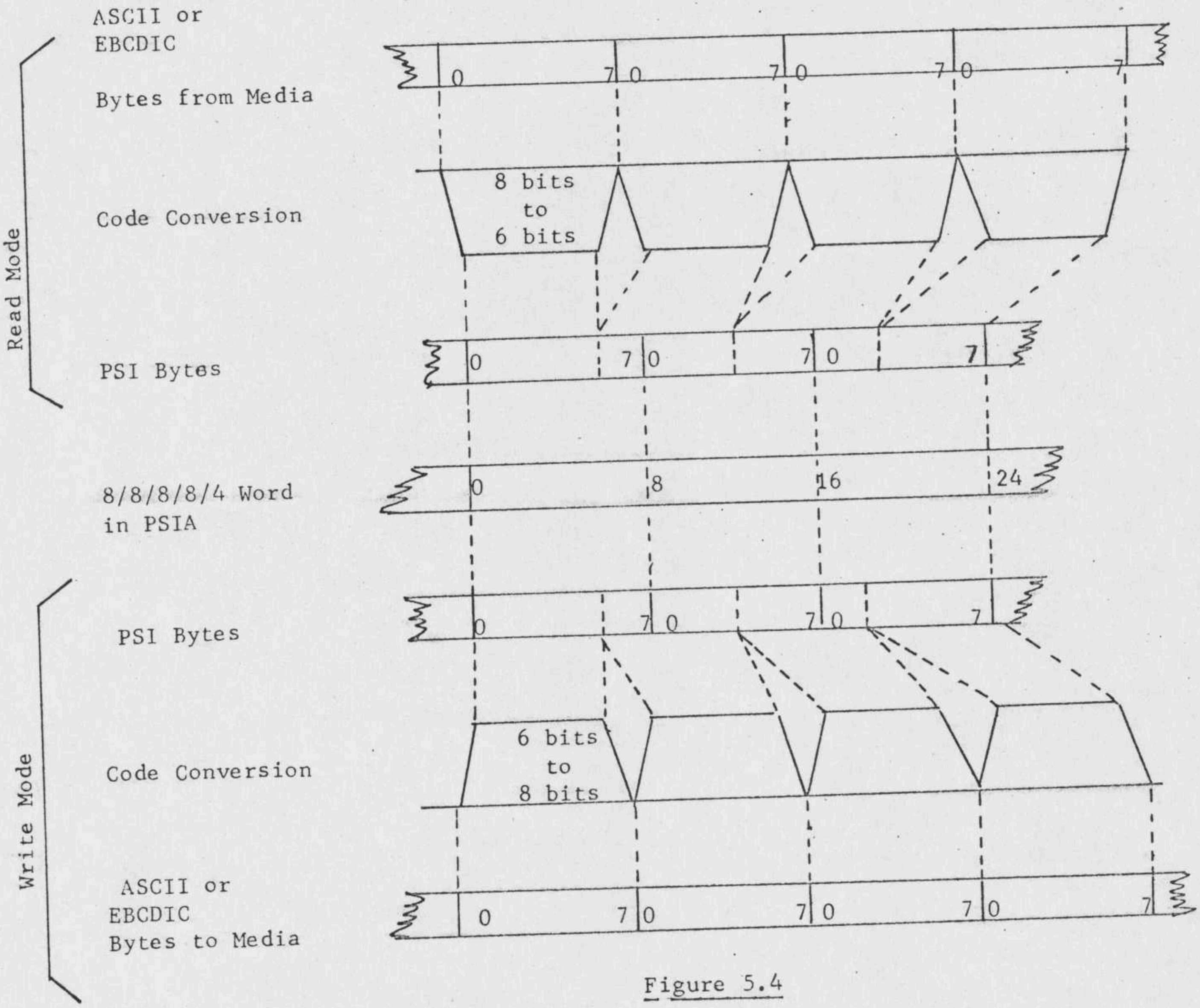


Figure 5.4

Example of Packing and Unpacking Required During Code Conversion

b) Nine Track Handler - In the nine track handler, this command is used to translate from a coded tape (in which the tape frames represent a coded EBCDIC character) to the six-bit 6000 line character set. The procedure for reading these tapes, and converting them to the six-bit character set is performed according to the table given in section 8.6.2.2.

In performing this command, the controller must place the PSI channel in the "binary mode". Then as each eight-bit byte is read from tape, it is translated to the corresponding six-bit character as defined by the EBDIC translation table. These six-bit characters are then packed into eight-bit bytes for transmission to the PSIA. If an eight-bit byte read from tape is not included in the defining table, the controller will substitute the 6000 code for the exclamation point (!, i.e., 77_8), and the error status MPC Device Data Alert - Code Alert is sent to the external user system at termination. Figure 5.4 illustrates the packing required within the controller.

5.4.37 Write ASCII Record (37_8)

This command results in the writing of one record with translation from the 6000 line six-bit character format to the ASCII format. The translation table for this code translation is given in section 8.6.3.1. If this command is issued to a controller which does not contain this option, the command shall be rejected with Command Reject - Invalid Operation Code status.

a) Seven Track Handler - This command is illegal if issued to a seven track handler and must be rejected with Command Reject - Nine Track Error status.

b) Nine Track Handler - With a nine track handler this command causes the translation of six-bit 6000 line characters into eight-bit ASCII bytes, and the subsequent writing of these bytes on tape.

In executing this command, the controller will place the PSIA in the "binary mode". Then the eight-bit bytes received from the PSIA are unpacked to six-bit characters, and each six-bit character translated to the corresponding eight-bit byte as defined by the translation table. The resulting eight-bit bytes are then written on tape. The unpacking requirements for this code translation are illustrated in Figure 5.4.

5.4.38 Read ASCII Record (27_8)

This command results in the reading of one record with translation to the six-bit character format. If this command is issued to a controller which does not contain this option, this command shall be rejected with Command Reject - Invalid Operation Code status.

a) Seven Track Handler - This command is illegal if issued to a seven track handler and must be rejected with Command Reject - Nine Track Error status.

b) Nine Track Handler - In the nine track handler, this command is used to translate from a coded tape (in which the tape frames represent a coded ASCII character) to the six-bit 6000 line character set. The procedure for reading these tapes, and converting them to the six-bit character set is performed according to the table given in section 8.6.3.2.

In performing this command, the controller must place the PSI channel in the "binary mode". Then as each eight-bit byte is read from tape, it is translated to the corresponding six-bit character as defined by the ASCII translation table. These six-bit characters are then packed into eight-bit bytes for transmission to the PSIA. If an eight-bit byte read from tape is not included in the defining table, the controller will substitute the 6000 code for the exclamationpoint (!, i.e., 77_8), and the error status MPC Device Data Alert - Code Alert is sent to the external user system at termination. Figure 5.4 illustrates the packing required within the controller.

5.4.39 Write ASCII/EBCDIC Record (35_8)

This command results in the writing of one record with code translation from an eight-bit byte in the 6000 (i.e., 7-bit ASCII) to another eight-bit byte on tape (i.e., EBCDIC). The table for this translation is given in Section 8.6.1.1. If this command is issued to a controller which does not contain this option, the command shall be rejected with Command Reject - Invalid Operation Code status.

a) Seven Track Handler - If this command is directed to a seven track handler, it will be rejected with the terminate status Command Reject - Nine Track Error.

b) Nine Track Handler - When this command is received in the controller for a nine track handler, the controller will send the service code "Initiate Data Transfer, Write ASCII" to the PSIA. As the eight-bit bytes are received from the PSIA (i.e., the ASCII bytes), each is translated to a corresponding eight-bit byte (i.e., EBCDIC) and written on tape.

5.4.40 Read ASCII/EBCDIC Record (25_8)

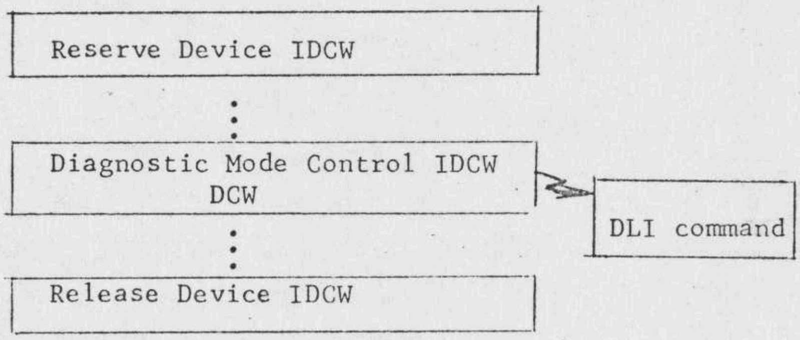
This command will result in one record from tape being read and translated to another code format. The record on tape must be in the EBCDIC format, and the resulting record tansmitted to the PSIA will be in the ASCII-7 format. The table for this translation is given in Section 8.6.1.2. If this command is issued to a controller which does not contain this option, the command must be rejected with Command Reject - Invalid Operation Code status.

a) Seven Track Handler - If issued to a seven track handler, this command will be rejected with Command Reject - Nine Track Error terminate status.

b) Nine Track Handler - In response to this command, the controller will place the PSIA in the"ASCII mode". Then, as the eight-bit EBCDIC bytes are read from tape, each is translated to a corresponding eight-bit byte (i.e., 7-bit ASCII) and transmitted to the PSIA. If a byte is read from tape which contains a code not included in the translation table, the controller will send the ASCII all ONEs (EO, i.e. 377_8) to the external user system, and the terminate status shall be MPC Device Data Alert - Code Alert.

5.4.41 Diagnostic Mode Control (31_8)

This command provides a mechanism for transferring DLI commands directly from the external user system to the addressed handler. Prior to issuing this command, the EUS must have issued a Reserve Device command to the desired handler. If the handler is not reserved when the Diagnostic Mode Control command is received by the controller, the command shall be rejected with MPC Command Reject - Illegal Procedure status. The correct channel program format is illustrated below.



In response to a correct Diagnostic Mode Control IDCW, the controller will place the PSIA in the ASCII mode, and begin requesting the information bytes. The first byte will contain the DLI command which the controller places directly on the control lines to the addressed handler. The other bytes are ignored by the controller.

If this DLI command requests information from the handler, the requested byte is placed in the controller Main Memory in the area containing error status for the addressed handler (See figure 7.5.2). The EUS may than retrieve this byte of handler status by issuing the Read Control Register command to the same handler as addressed by the Diagnostic Mode Control command.

5.4.42 Device Wraparound (32_8)

This command is used in the handler diagnostic mode to set a specific data wraparound point in the addressed handler, and exercise this data path by transmitting data from the EUS to the handler with the data wrapped around in the handler and sent back to the controller. The channel program format is illustrated in Section 5.4.41; the distinction is that the Device Wraparound command replaces the Diagnostic Mode Control command, and the data passed to the controller is different.

5.4.4.2 Device Wraparound (32_8) continued

The Device Wraparound IDCW will cause the controller to prepare to receive a variable number of bytes from the EUS. The controller will place the PSIA in the ASCII mode preparatory to receiving this data. The first byte received by the controller will be a DLI command which is placed directly on the control lines to the handler. This byte will set the density in the handler. The second byte received by the controller from the EUS will be a DLI command which establishes the wraparound path in the handler. This command is also placed directly on the control lines to the addressed handler. The next two bytes received by the controller are ignored.

All of the ensuing bytes received by the controller from the EUS are data bytes to be sent to the handler. Prior to sending this data to the handler, the controller will assume a PE mode of operation. The controller will generate a normal preamble, and will turn off the in-flight error correction capability.

Assuming that the handler is a 9-channel unit, the operation is as follows. Each byte of data received from the EUS is placed on the data lines to the addressed handler, and the controller maintains an eight-bit count of the bytes transferred to the handler. Each data byte transmitted to the handler will be wrapped around at the required level, and returned to the controller. The controller will check each returned byte for correct parity.

If this operation continues without an error until the EUS terminates the operation, the controller will place the final byte returned from the handler in byte 14 of the handler error log (Figure 7.5), and the byte count in byte 15 of this log. The controller then terminates with Ready status.

If a byte returned from the handler contains a parity error, the controller will terminate the operation. The byte in error will be stored in byte 14 of the handler error log, and the byte count will be stored in byte 15. The status sent to the external user system will be MPC Data Alert - T&D Error. All other errors result in normal error termination.

If the handler is a 7-channel unit, the operation is performed in essentially the same manner except that as the data bytes are received from the EUS and transmitted to the handler, the handler will ignore the two high-order bits. The returned data will then be a character rather than a byte.

5.4.42 Device Wraparound (32_8) continued

Note that after completion of execution of this command, the external user system may obtain the byte-in-error (or final byte) and the byte count by sending the Read Control Register command addressed to the same handler.

5.4.43 Write Timing Character (30_8)

This command provides a method of checking handler speed. The channel program is similar to that illustrated in Section 5.4.40 with the exception that the Diagnostic Mode Control IDCW is replaced with the Write Timing Character IDCW. Also, the format of the data transferred to the controller is changed as described below.

In response to the Write Timing Character IDCW, the controller will place the PSIA in the ASCII mode, and request data bytes from the external user system. With the handler performing a normal write command, the data received from the EUS is written on tape. Each byte written on tape is returned to the controller via the normal read-after-write mechanism.

This writing will continue until either (1) terminated by the EUS, or (2) the controller detects a read-after-write character (byte) with the 2^4 bit ON. The respective actions are as follows:

EUS termination - The controller will have maintained a count of the bytes written on tape. This eight-bit count is stored in byte 15 of the error log for that handler (Figure 7.5), with the final byte returned to the controller via the read-after-write path stored in byte 14 of this error log. The termination status is then Ready.

TCA termination - The TCA will signal the controller firmware when a character (byte) containing the 2^4 bit ON is received via the read-after-write path. The byte count maintained by the controller will be stored as if the EUS had terminated the operation. Also, the character (byte) which had the 2^4 bit ON is stored as in the EUS termination. The status returned to the EUS in this case is Ready.

The byte count and final byte returned to the controller via the read-after-write path are both available to the external user system by way of the Read Control Register command issued to the particular handler. If any normal error is detected, normal error status is reported.

5.5 PERMISSABLE SPECIAL CONTROLLER IDCW COMMANDS

As indicated in Section 5.2, particular IDCW channel instruction field contents indicate that the IDCW is defining a Special Controller command. These commands are described in the following section, with a description of their application in channel programs. The permissable commands of this type are listed below.

Command	Octal Representation	Section
Suspend Controller	00	5.6.5.1
Release Controller	20	5.6.5.2
Initiate Read Data Transfer	06	5.6.5.3
Initiate Write Data Transfer	16	5.6.5.4
Read Controller Main Memory (ASCII)	02	5.6.5.5
Write Controller Main Memory (ASCII)	12	5.6.5.6
Read Lock Byte	04	5.6.5.7
Write Lock Byte	14	5.6.5.8
Conditional Write Lock Byte	34	5.6.5.9
Write Control Store	10	5.6.5.10
Execute Control Store Microprogram	30	5.6.5.11
Write Controller Main Memory (BIN)	32	5.6.5.11
Read Controller Main Memory (BIN)	22	5.6.5.13

The device instruction field (device code) for all the above commands must be ZERO.

5.6 SPECIAL CONTROLLER COMMAND IMPLEMENTATIONS

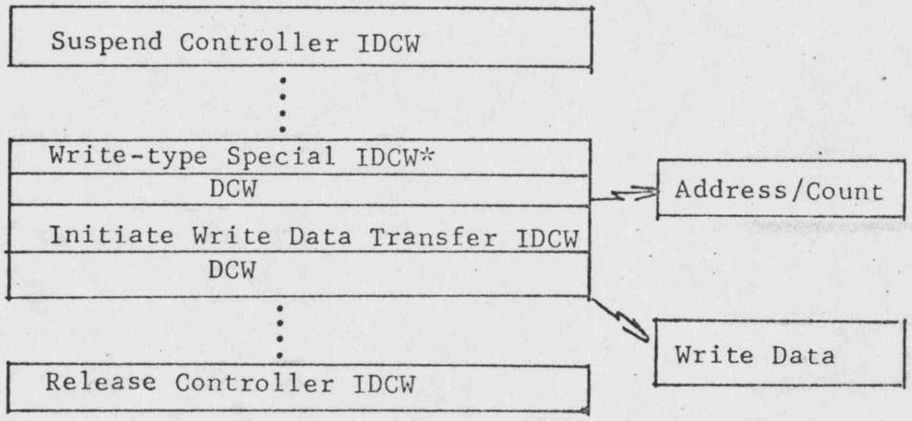
The Special Controller Commands were referenced in Section 5.2, and will be fully described in this section. These commands form a group of commands dedicated to controller-type functions related to MPC controllers. They are identified by a 10XXXX configuration in the Channel Instruction field of the IDCW being in the ONE state. The IDCW format for these commands is identical to the standard IDCW format defined in Section 5.2, and forms a special case for IDCW implementation. Sections 5.6.1 through 5.6.4 describe channel programs using these Special Controller Commands. Section 5.6.5 defines the details of these commands.

5.6.1 Type A Implementation of Special Controller Commands

This implementation of the Special Controller commands is used to write information into the controller Main Memory, or to load firmware into the controller Control Store. Note that the controller must be in the suspended state when the information

5.6.1 Type A Implementation of Speical Controller Commands (continued)

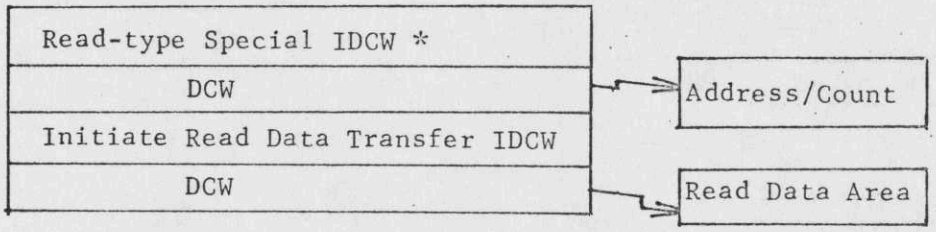
transfer commands are issued, and the controller must be released prior to returning to normal operation.



*The IDCWs which may be used here are
Write Controller Main Memory (BIN)
Write Controller Main Memory (ASCII)
and
Write Control Store

5.6.2 Type B Implementation of Special Controller Commands

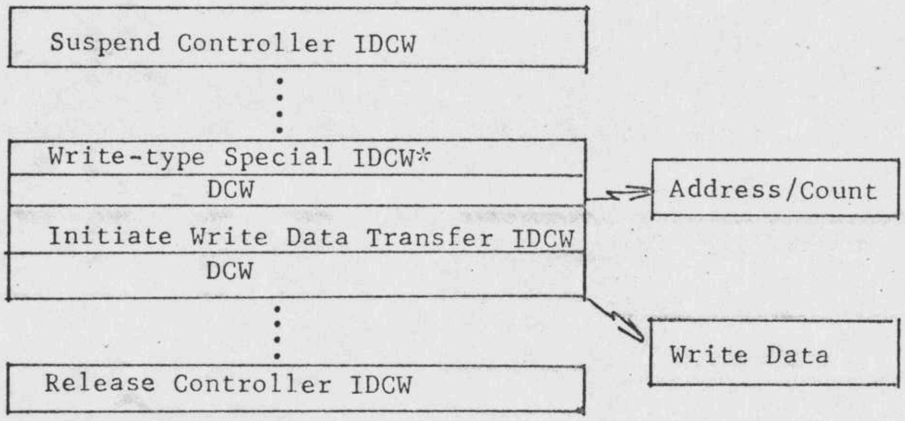
This implementation of the Special Controller commands is used to read selected information from the controller Main Memory. Note that the controller need not be in the suspended state for these commands to be recognized.



*The Special IDCWs which may be used here are
Read Controller Main Memory (BIN)
Read Controller Main Memory (ASCII)
and
Read Lock Byte

5.6.1 Type A Implementation of Speical Controller Commands (continued)

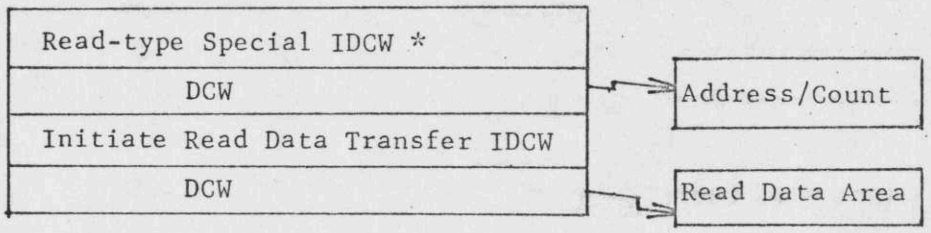
transfer commands are issued, and the controller must be released prior to returning to normal operation.



*The IDCWs which may be used here are
Write Controller Main Memory (BIN)
Write Controller Main Memory (ASCII)
and
Write Control Store

5.6.2 Type B Implementation of Special Controller Commands

This implementation of the Special Controller commands is used to read selected information from the controller Main Memory. Note that the controller need not be in the suspended state for these commands to be recognized.



*The Special IDCWs which may be used here are
Read Controller Main Memory (BIN)
Read Controller Main Memory (ASCII)
and
Read Lock Byte

5.6.5 Description of Special Controller Commands

5.6.5.1 Suspend Controller (00_8)

This command allows a controller to be seized by one external user system when the controller is physically connected to more than one system via two or more PSI interfaces. This command will cause the controller to take the following actions:

- o Ignore all Channel Program Waiting lines from all PSI interfaces, and wait until all current operations in progress at the time this command was received are completed. (This implies that any channel program which has been initiated will be completed).
- o Issue a Special Status Storage and Special Interrupt to all connected PSI interfaces (see Section 6.6).
- o Execute the action defined by the Continue/Marker bits of the Suspend Controller IDCW (see Section 5.2).
- o Ignore all Channel Program Waiting lines from any PSI interface other than the one over which the Suspend Controller command was received. The PSI interface initiating the Suspend Controller command will be serviced normally, with the following exception;

  > Subsequent IDCWs received by the controller over the PSI interface which issued this command must be directed to the same logical channel as the Suspend Controller command or the new command will be rejected with MPC Command Reject - Illegal Logical Channel Number status.

Ths Suspend Controller command when used must be the first IDCW in the channel program.

5.6.5.2 Release Controller (20_8)

This command is used to reset the suspended state set up by a previously issued Suspend Controller command. If the controller is not suspended when this command is received by the subsystem, the command will be terminated with Ready status.

In response to this command the controller will issue a Special Status Storage and Special Interrupt to logical channel ZERO of all connected PSI interfaces as defined in Section 6.6. The controller will then execute status entry and interrupt. The Release Controller command will always be considered to be the last command of the channel program, regardless of state of continue bit of IDCW.

5.6.5.3 Initiate Read Data Transfer (06_8)

This command is used with the Type B implementation of the Special Controller commands, and is used to indicate to the controller that the information transfer is to start. Prior to starting the data transfer the controller will place the PSIA in the proper mode, binary or ASCII mode. Then the data transfer will start at the location specified by the previously issued Read Controller Main Memory command, or by the previously issued Read Lock Byte command. If the IDCW preceding the Initiate Read Data Transfer command was not either the Read Controller Main Memory or Read Lock Byte commands, the controller shall send MPC Command Reject - Illegal Procedure status to the EUS.

5.6.5.4 Initiate Write Data Transfer (16_8)

This command is used in the Type A implementation of the Special Controller command sequence, and indicates to the controller that the information transfer should begin. It must be preceded by one of the following commands; Write Controller Main Memory (ASCII), Write Controller Main Memory (BIN), or Write Control Store. If this condition is not met, the controller shall send MPC Command Reject - Illegal Procedure status to the external user system.

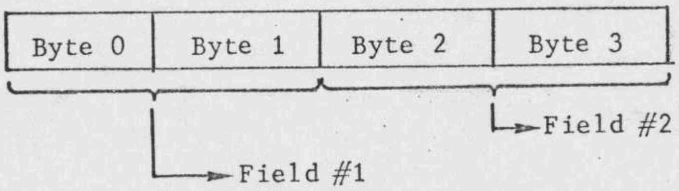
If the Initiate Write Data Transfer IDCW is correct, and the above sequence is followed, the controller will place the PSIA in the proper mode, binary or ASCII prior to requesting the first byte of information. The actual information transfer will then occur in accordance with the previously issued command.

5.6.5.5 Read Controller Main Mamory (ASCII) (02_8)

This command is used to transfer information from the controller Main Memory to the external user system. The controller must check the validity of the channel instruction field of the IDCW, and of the continue bit. If the former is not valid, the controller sends termination status with Channel Status of 010 to the EUS; if the continue bit is not a ONE, the controller sends MPC Command Reject - Continue Bit.

After receiving this IDCW, and determining that it is correct, the controller will issue an Initiate Data Transfer, Read Binary service code to the external user system, and then receive four bytes of command data defined as follows:

5.6.5.5 Read Controller Main Memory (ASCII) (02_8) continued



Field #1; Defines a 16-bit word address in the controller Main Memory at which location the read operation is to start. The most significant bit of this address is bit 0 of byte 0. This address is not checked for legality in the controller.

Field #2; Defines a 16-bit word count for the maximum number of controller words to be accessed by the execution of the command, where controller words refer to two-byte words. All ZEROs is a valid word count.

Following completion of the error-free reception of the command data, the controller will issue a Move Pointer and Initiate Command Transfer service code to the EUS, and receive the next command. If it is not, execution of the read command will be aborted, and the next command executive initiated.

If this second IDCW is correct, the controller will issue the Initiate Data Transfer, Read ASCII service code, and begin to transfer data to the EUS as defined by the Read Controller Main Memory (ASCII) command.

5.6.5.6 Write Controller Main Memory (ASCII) 12_8)

This command is used to write data from the EUS into the controller Main Memory. The checking of the IDCW must occur as in the Read Controller Main Memory (ASCII) command. Also, the command data transferred to the controller will take the same format as in the Read Controller Main Memory (ASCII) command.

Following the completion of the error-free reception of the command data, the controller will issue a Move Pointer And Initiate Command Transfer service code to the EUS. The next IDCW received by the controller must then be the Initiate Write Data Transfer IDCW. If not, controller will abort the write command, and execute the new command.

5.6.5.6 Write Controller Main Memory (ASCII) (12_8) continued

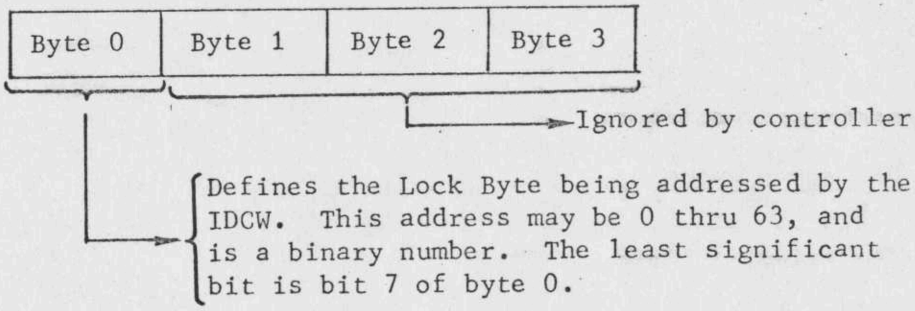
If this second IDCW is correct, the controller issues the Initiate Data Transfer, Write ASCII service code, and proceed with the information transfer. The starting address in controller Main Memory, and the block count, was specified by the command data received by the controller in response to the Write Controller Main Memory command.

Note that if the controller is not in the suspend state when this command is received, the command shall be rejected with MPC Command Reject - Illegal Procedure status.

5.6.5.7 Read Lock Byte (04_8)

This command is used to transfer the address of a lock byte to the controller. As noted in section 5.6.2, this command is followed by an Initiate Read Data Transfer command to actually transfer the contents of the lock byte to the external user system.

Upon receiving the Read Lock Byte IDCW, the controller will issue an Initiate Data Transfer, Write Binary service code to the EUS, and then receive four bytes of command data formatted as follows:



Byte 0	Byte 1	Byte 2	Byte 3

Ignored by controller

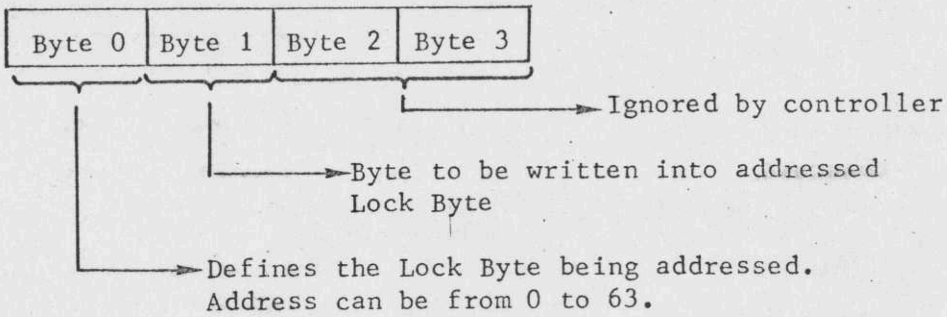
Defines the Lock Byte being addressed by the IDCW. This address may be 0 thru 63, and is a binary number. The least significant bit is bit 7 of byte 0.

Upon completion of the reception of the command data, the controller will issue a Move Pointer and Initiate Command Transfer service code to the external user system. The next IDCW received by the controller must be an Initiate Read Data Transfer IDCW. If this is not the next command, the controller will abort Read Lock Byte command, and start execution of the new command.

However, if this second command is valid, it will be executed as described in section 5.6.5.3.

5.6.5.8 Write Lock Byte (14_8)

As illustrated in section 5.6.3, this command is used to transfer information to a specified Lock Byte in the controller Main Memory. Upon receiving this IDCW, the controller will issue an "Initiate Data Transfer, Write Binary service code to the external user system. The EUS will then transfer four bytes of command data as shown below.


Byte 0
Byte 1
Byte 2
Byte 3
Ignored by controller
Byte to be written into addressed Lock Byte
Defines the Lock Byte being addressed. Address can be from 0 to 63.

After receiving this IDCW, the controller will store the contents of byte 1 in the controller Main Memory as specified by byte 0.

5.6.5.9 Conditional Write Lock Byte (34_8)

This command is used to conditionally write information into the specified Lock Byte. This Lock Byte is written into if it presently contains all-zeros; otherwise, the command is terminated as defined below.

Upon receiving this IDCW, the controller will issue an Initiate Data Transfer, Write Binary service code to the external user system, and receive four bytes formatted as defined in section 5.6.5.7. After the command data has been received by the controller, it will access the addressed Lock Byte. If this byte is non-zero, the controller will leave it unmodified and reflect this condition by means of termination status MPC Data Alert - Byte Locked Out.

However, if the addressed Lock Byte is zero, the controller will write the contents of byte 2 of the IDCW into the addressed Lock Byte.

5.6.5.10 Write Control Store (10_8)

This command may be used to write microprograms into the controller Control Store. When this command is received by the controller, the controller must be in the suspended state or the command will be rejected with MPC Command Reject - Illegal Procedure status.

The general channel program is illustrated in section 5.6.1 for this command. In response to this command, the controller will issue

5.6.5.10 Write Control Store (10_8) continued

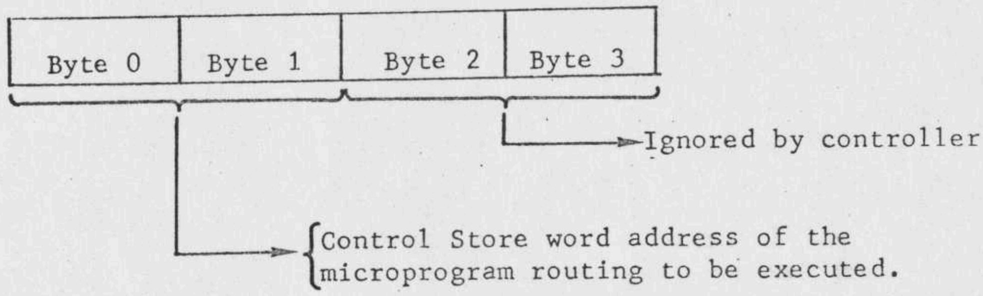
a Move Pointer And Initiate Command Transfer service code to the EUS, and receive the next IDCW. This next IDCW must be Initiate Write Data Transfer, or the write command will be aborted, and the new command execution begun. If it is correct, the controller will issue an Initiate Data Transfer, write ASCII service code to the EUS, and proceed to receive the information to be written into the Control Store.

A sum-check will be sent as the last word (two-byte word) of the information. This sum-check is not written into the Control Store. It will represent the two's complement of the final summation of the addition of all two-byte words sent to the controller, with end-around carry (i.e., as each two-byte word is added to the sum-check accumulation, carry out of the high-order byte of the two-byte accumulation is added back into the sum-check accumulation). The sum-check word is included in the total word count specified for the command execution. Failure of the transmitted sum-check to agree with the controller-computer sum-check will cause the controller to send MPC Data Alert - Sum Check Error status to the EUS.

5.6.5.11 Execute Control Store Microprogram (30_8)

This command may be used to start execution of the controller microprogram beginning at a specified address in the Control Store. The controller must be in the suspended state when this command is received, or the controller shall reject the command with MPC Command Reject - Illegal Procedure.

Upon receiving this command, the controller will issue an Initiate Data Transfer, Write Binary service code to the EUS, and proceed to receive five bytes of command data formatted as below.


Byte 0
Byte 1
Byte 2
Byte 3
Ignored by controller
Control Store word address of the
microprogram routing to be executed.

Upon reception of this command data the controller will branch internally to the addressed microprogram. The mechanism by which the internal microprogram returns will be defined during implementation.

5.6.5.12 Write Controller Main Memory (BIN) (32_8)

This command is used to write data from the EUS into the controller Main Memory, with the data formatted in the EUS core or binary information.

The checking of this IDCW, and the information transferred to the controller will take the same format as defined for the Write Controller Main Memory (ASCII) command (section 5.6.5.5).

The execution of this command sequence will be the same as for the Write Controller Main Memory (ASCII) command (section 5.6.5.6) except that after the reception of the Initiate Write Data Transfer IDCW, the controller will leave the PSIA in the binary mode.

5.6.5.13 Read Controller Main Memory (BIN) 22_8)

Execution of this command is identical to that of the Read Controller Main Memory (ASCII) with the exception that following reception of the Initiate Read Data Transfer IDCW, the controller will leave the PSIA in the binary mode.

6.0 STATUS

There are two classes of status which the controller may send to the PSIA --- terminate status and special status. The controller is required to supply status in a defined format for each class, as well as incorporate status supplied by the PSIA in specific situations.

6.1 STATUS MAPPING FOR "STORE TERMINATE STATUS"

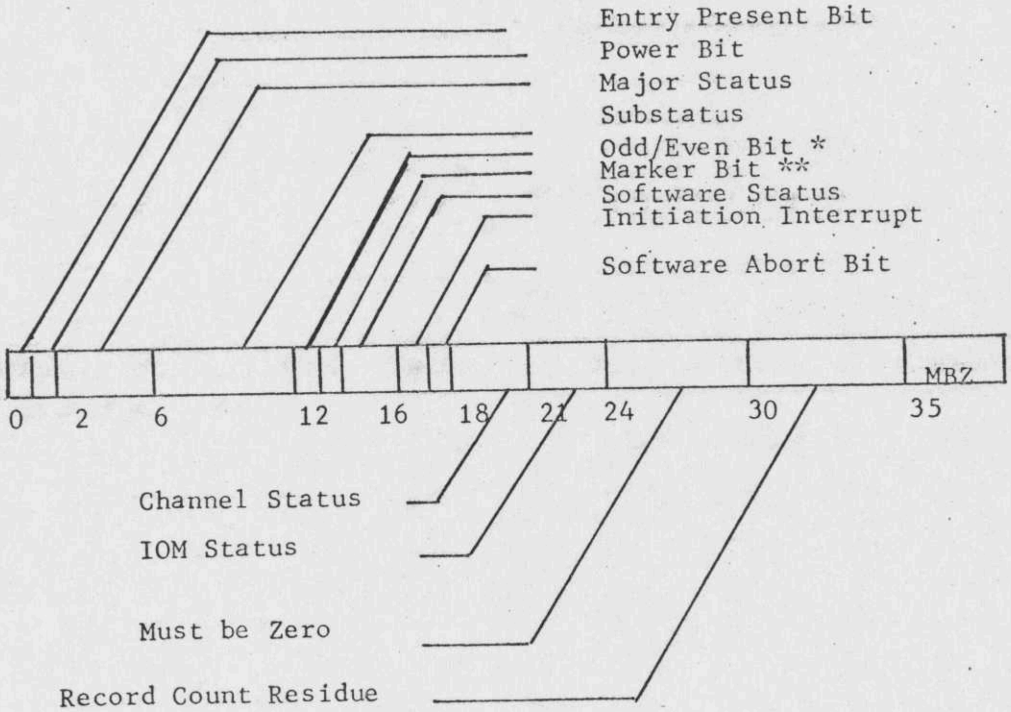
The Store Terminate Status service code is used to store the termination status for a logical channel. Figure 6.1.1 illustrates the mapping of controller-supplied fields into the even status word. Bit 12 is always supplied by the PSIA, and is OR'd by the PSIA with the corresponding ZERO supplied by the controller. This figure does not show the odd status word since it is totally supplied by the PSIA.

Figure 6.1.2 defines the bit positions of the even status word, and the corresponding bit positions within the bytes transmitted by the controller to the PSIA.

The definitions of these bits change as the situation varies. There are three categories in which the terminate status must be used. These categories are defined in the next three sections.

The sequence of operations which results in the terminate status being moved from the controller to the IOM is as follows:

- o Controller issues "Store Status" service code to PSIA.
- o Five bytes of terminate status are sent to PSIA.
- o Controller issues "Set Terminate Interrupt" or "Set Marker Interrupt" (See Section 5.2) service code to PSIA. (See 43A177880 for service code descriptions).



* Controller sets this bit to ZERO.

** Controller sets this bit to ONE if the termination requires a Marker Interrupt; set to ZERO if the termination requires a Terminate Interrupt (see Section 5.2, and Table 6.2).

Even Status Word Definition

Figure 6.1.1

EVEN STATUS WORD BIT	TERMINATE STATUS BYTE : BIT	BIT DESCRIPTION	
0	0:0	Entry Present Bit	
1	0:1	Power Bit	
2	0:2	Major Status	MSB
3	0:3	Major Status	
4	0:4	Major Status	
5	0:5	Major Status	LSB
6	0:6	Substatus	MSB
7	0:7	Substatus	
8	1:0	Substatus	
9	1:1	Substatus	
10	1:2	Substatus	
11	1:3	Sbustatus	LSB
12	1:4	Odd/Even Bit*	
13	1:5	Marker Bit**	
14	1:6	Software Status	
15	1:7	Software Status	
16	2:0	Initiation Interrupt	
17	2:1	Software Abort Bit	
18	2:2	Channel Status	MSB
19	2:3	Channel Status	
20	2:4	Channel Status	LSB
21	2:5	IOM Status	MSB
22	2:6	IOM Status	
23	2:7	IOM Status	LSB
24	3:0	Must be Zero	
25	3:1	Must be Zero	
26	3:2	Must be Zero	
27	3:3	Must be Zero	
28	3:4	Must be Zero	
29	3:5	Must be Zero	
30	3:6	Record Count Residue	
31	3:7	Record Count Residue	
32	4:0	Record Count Residue	
33	4:1	Record Count Residue	
34	4:2	Record Count Residue	
35	4:3	Record Count Residue	
	4:4	Must be Zero	
	4:5	Must be Zero	
	4:6	Must be Zero	
	4:7	Must be Zero	

* Controller sets this bit to ZERO

** Controller sets this bit as defined in Figure 6.1.1.

Mapping Table for Terminate Status Bytes

Figure 6.1.2

6.2 USE OF "STORE TERMINATE STATUS" - CATEGORY A

The first category within the terminate status occurs when the controller detects an illegal IDCW, as defined in section 5.1. In particular, if bits 18-20 of the IDCW are not all ONE's, or if the channel instruction field is not correct, the controller must notify the system via the Store Terminate Status service code, followed by the five status bytes. The status bits not indicated as "must be zero" in Figure 6.1.2 are supplied by the controller as defined by Table 6.2.

6.3 USE OF "STORE TERMINATE STATUS" - CATEGORY B

Category B includes the normal command termination, where this termination may be initiated by either the PSIA or the controller.

It also includes command termination due to faults detected within the subsystem, exclusive of those covered by Category A in section 6.2.

The status bits supplied by the controller for Category B are defined in Table. 6.2.

6.4 USE OF "STORE TERMINATE STATUS" - CATEGORY C

This category of terminate status accounts for those conditions where the IOM/PSIA detects a malfunction. In this situation the Service Code Waiting line (SCW) will be turned ON by the PSIA. In addition, the Terminate Out line (TMO) may be turned ON.

If the SCW line is turned ON, and the TMO line remains OFF, the controller should request that the service code (see TAble 6.4) be sent to the controller just prior to formatting the terminate status for transmission to the PSIA. If both SCW and TMO lines are ON, the controller will halt immediately, request the service code from the PSIA, and format the terminate status for transmission to the PSIA.

The service code sent to the controller contains status information for the channel status and IOM status fields of the even status word. Table 6.4 specifies these service codes. The controller will use this information in formatting the terminate status bytes as described below.

The method of requesting these service codes from the PSIA is as follows. The SCW line in the ON state means that the first byte sent to the controller in response to any controller-generated service code, received from the controller, with SCW in the ON state, and turn it into a request for transmission of the service code containing the status fields to the controller.

The formatting of the terminate status bytes in this category is accomplished as defined by Table 6.2.

TABLE 6.2
TERMINATE STATUS
EVEN STATUS WORD

EVEN STATUS WORD FIELD	CATEGORY A	CATEGORY B	CATEGORY C
Entry Present Bit	ONE	ONE	ONE
Power Bit	ZERO	ZERO	ZERO
Major Status	0000	As defined in section 6.5	As defined in section 6.5
Substatus	000000	As defined in section 6.5	As defined in section 6.5
Odd/Even Bit	ZERO	See Figure 6.1.1	See Figure 6.1.1
Software Status	00	00	00

TABLE 6.2 (continued)

EVEN STATUS WORD FIELD	CATEGORY A	CATEGORY B	CATEGORY C
Initiation Interrupt	ONE	Set = ONE if subsystem did not go busy* as a result of this command.	Set = ONE if subsystem did not go busy* as a result of this command.
Abort Bit	ZERO	ZERO	ZERO
Channel Status	If bits 18-20 of IDCW were not all ONE's, set the channel status bits to 011 where the most significant bit is to the left. If the channel instruction field of the IDCW was not legal, set the c channel status bit to 010.	000	As defined in Table 6.4
IOM Status	000	000	As defined in Table 6.4
Record Count Residue	000000	This is the residue count for a peripheral action or multiple record instruction; stored as a ZERO in all other cases. The record count shall be decremented each time the instruction is executed.	This is the residue count for a peripheral action or multiple record instruction; stored as a ZERO in all other cases. The record count shall be decremented each time the instruction is executed.

TABLE 6.2 (continued)

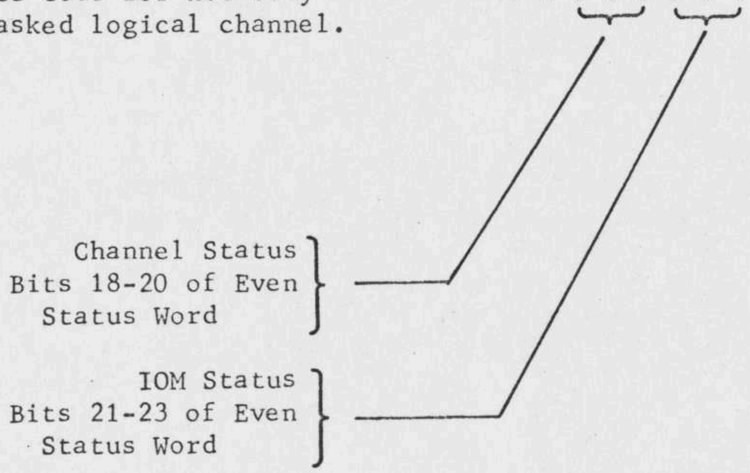
* The busy condition is defined as follows:

1. Tape movement commands. Busy means the controller has issued the DLI Run command to the addressed handler.

2. Data transfer, not including tape movement. Busy means the controller has issued an initiate data transfer type service code to the EUS.

3. No data transfer, with no tape movement. Busy means that the controller has attempted to execute the command.

	Service Code Sent to Controller
IOM DETECTED USER FAULTS	0 1 2 3 4 5 6 7 PSI Bit Positions
LPW tally run out	1 0 0 0 0 0 0 1
2 TDCW's in a row	1 0 0 0 0 0 1 0
Boundary Error	1 0 0 0 0 0 1 1
IDCW in restricted mode	1 0 0 0 0 1 0 1
Char pos/size discrepancy	1 0 0 0 0 1 1 0
Parity Error, PSIA to IOM	1 0 0 0 0 1 1 1
PSIA DETECTED FAULTS	
Connect while busy (unexpected PCW)	1 0 0 0 1 0 0 0
Data parity error, PSI	1 0 1 1 0 0 0 0
Parity error, IOM to PSIA	1 0 1 1 1 0 0 0
Illegal Service Code	0 1 0 0 0 0 0 0
Parity error during service code sequence	0 1 0 0 0 0 0 0
Illegal DCW	1 0 0 1 1 0 0 0
OTHER EVENTS	
IOM System Fault	1 1 0 0 0 0 0 0
Masking PCW	1 1 0 0 0 0 0 0
Service code for not busy or masked logical channel.	1 1 0 0 0 0 0 0



Channel and IOM Status

TABLE 6.4

6.5 TERMINATE STATUS - MAJOR AND SUBSTATUS

The major status conditions contain a priority arrangement such that when two unique major status conditions exist at the same time, the higher priority status will be reflected to the external user system. The priority for the major status is as follows:

Command Reject	Highest Priority
MPC Command Reject	
End of File	
Device Data Alert	
MPC Device Data Alert	
Device Attention	
MPC Device Attention	
Device Busy	Lowest Priority

Either channel may obtain device status. If device is reserved to alternate channel, status is Device Busy - Device Reserved.

The major status and substatus combinations are summarized below; they are detailed in the following sections. The X in the substatus field means that the bit may be a ZERO or a ONE. Substatus codes not listed are illegal.

	Major Status	Substatus	Section
	0:2*	0:6*	
Peripheral Subsystem Ready	0000		6.5.1
Ready		000000	
Write Protected		000XX1	
Positioned at BOT		000X1X	
Nine Track Handler		0001XX	
Two Bit Fill		01000X	
Four Bit Fill		10000X	
Six Bit Fill		11000X	
Device Busy	0001		6.5.2
In Rewind		000001	
Device Reserved		100000	
Alternate Channel in Control		000010	
Device Loading		000100	

*Indicates byte and bit positions of status word (see Figure 6.1.2).

	Major Status	Substatus	Section
Device Attention	0010		6.5.3
Write Protected		00XX01	
No such Handler		000010	
Handler in Standby		0XX10X	
Handler Check		0X1X0X	
Blank Tape on Write		01XX00	
Device Data Alert	0011		6.5.4
Transfer Timing Alert			
Blank Tape on Read		000010	
Bit Detected During Erase Operation		XXXX11	
Transmission Parity Alert		XXX1XX	
Lateral Tape Parity Alert		XX1XXX	
Longitudinal Tape Parity Alert		X1XXXX	
End of Tape Mark		1XXXXX	
End of File	0100		6.5.5
End of File Mark (Seven Track)		001111	
End of File Mark (Nine Track		010011	
Data Alert Condition		111111	
Single Character Record		XXXXXX	
Command Reject	0101		6.5.6
Invalid Operation Code		000XX1	
Invalid Device Code		000X1X	
Invalid IDCW Parity		0001XX	
Positioned at BOT		001000	
Forward Read after Write		010000	
Nine Track Error		100000	
MPC Device Attention*	1010		6.5.7
Configuration Switch Error		000001	
Multiple Devices		000010	
Illegal Device I.D. Number		000011	
Incompatible Mode		001000	
TCA Malfunction		0011XX	
MTH Malfunction		010000	
Multiple BOT		010001	
MPC Device Data Alert*	1011		6.5.8
Transmission Parity Alert		000001	
Inconsistent Command		000010	
Sum Check Error		000011	
Byte Locked Out		000100	
PE-Burst Write Error		001000	
Preamble Error		001001	
T&D Error		001010	
Multi-track Error		010000	
Skew Error		010001	
Postamble Error		010010	
NRZI CCC Error		010011	
Code Alert		010100	

	Major Status	Substatus	Section
MPC Command Reject*	1101		6.5.9
Illegal Procedure		000001	
Illegal L.C. Number		000010	
Illegal Suspended L.C. Number		000011	

*The MPC major status codes will be mapped by software to existing corresponding major status.

6.5.1 Peripheral Subsystem Ready (0000)

This major status return indicates the subsystem is ready to accept a command.

At the time this major status condition is sent, substatus will be encoded and transmitted on the information lines to the external user system as follows:

a) Ready 000000

The device last addressed is ready to accept a command.

b) Write Protected 000X1X

Last handler addressed by the external user system is write protected.

c) Positioned at BOT 000X1X

The tape reel mounted on last handler addressed by the EUS is positioned at the BOT.

d) Nine Track Handler 0001XX

Last handler addressed by the EUS is a nine track unit.

e) Two Bit Fill 01000X

The final byte transmitted to the EUS for a 7-channel read operation contained two low-order zeros padded to final character (see section 4.4.1).

6.5.1 Peripheral Subsystem Ready (0000) continued

f) Four Bit Fill 10000X

The final byte transmitted to the EUS for a 7-channel read operation contained four low-order zero bits padded to final character read from tape.

g) Six Bit Fill 11000X

The final byte transferred to the EUS for a 7-channel read operation contained six low-order zero bits padded to final character read from the tape.

6.5.2 Device Busy (0001)

This major status indicates that the addressed handler is either rewinding , loading, or under the control of another channel.

a) In Rewind 000001

The addressed handler is currently executing a rewind operation.

b) Device Reserved 100000

The addressed handler is currently reserved to an alternate channel as a result of Reserve Device command.

c) Alternate Channel in Control 000010

The addressed handler is currently busy with an I/O operation on the alternate channel. The handler has not been reserved by a Reserve Device command.

Note that the controller will not hold commands waiting for an addressed handler to come Ready.

d) Device Loading 000100

The addressed handler is currently executing a Load command.

6.5.3 Device Attention (0010)

This major status return indicates a condition requiring manual intervention has occurred within the addressed handler. At the time this status condition is sent, substatus will be transmitted as follows:

6.5.3 Device Attention (0010) continued

a) Write Protected 0XXX01

This status return indicates that a Write command was issued to a handler that does not have a Write Permit Ring installed, or that this handler has been write protected with a Set File Protect command (see section 5.4.8).

b) No Such Handler 000010

This status indicates that the handler addressed does not exist, or is in the Off-Line state.

c) Handler Standby 0XX10X

This status indicates that the addressed handler is in the Standby state.

d) Handler Check 0X1X0X

This status indicates that the addressed handler is unable to operate properly.

The detection of the Handler Check condition during the processing of a command will cause the handler to be set to the Handler Standby state. Both substatus conditions will be reflected to the external user system under the Device Attention major status. The first non-Request Status command addressed to the original handler will result in the resetting of the Handler Check substatus. The subsequent commands addressed to the original handler will be rejected with Device Attention - Handler Standby status only.

e) Blank Tape on Write 01XX00

This status indicates that the tape unit receiving the last Write command did not write properly,as no bits were detected by the echo read operation.

The detection of the Blank Tape on Write condition will cause the tape unit to be set to the Tape Unit Standby state. Both substatus conditions will be reflected to the external user system under the Device Attention major status. Following the reflection of this status to the external user system, the first non-Request Status command received will result in the resetting of the Blank Tape on Write substatus, so that subsequent commands addressed to the original handler will be rejected with the Handler Standby-Device Attention status only.

6.5.4 Device Data Alert (0011)

This major status return indicates some type of alert condition occurred during the last command execution within the subsystem. At the time this status condition is sent, substatus will be transmitted as follows:

a) Transfer Timing Alert 000001

This alert indicates that the external user system or the controller did not transfer data at a rate compatible with the data transfer rate of the tape.

If this condition is detected during the execution of a Read command, further data transfer to the external user system will be completed. Upon the completion of tape movement to the inter-record gap, the Transfer Timing - Data Alert Status will be sent to the external user system. See section 7.2.

If this condition is detected during a write operation, the operation is as defined in section 7.1.

b) Blank Tape on Read 000010

This alert indicates that no bits were detected during a read operation for a nominal distance of 25 feet.

Upon the detection of this condition during a read or space operation the handler will be stopped the subsystem will reflect the Blank Tape On Read - Data Alert status.

c) Transmission Parity Alert XXX1XX

This alert indicates that a parity error was detected by the LA on the write data from the external user system.

The detection of this condition, which can only occur during a write operation, does not cause the termination of the write operation. The write operation will proceed to its normal termination, at which time the Transmission Parity - Data Alert status will be reflected to the external user system.

d) Lateral Tape Parity Alert XX1XXX

This alert indicates that incorrect parity across the tape was detected on data transferred.

The detection of this condition, which can occur during either a read or write operation, does not cause the termination of the operation. The operation will proceed to its normal termination, at which time the Lateral Tape Parity - Data Alert status will be reflected to the external user system.

6.5.4 Device Data Alert (0011) continued

e) Longitudinal Tape Parity Alert X1XXXX

This alert indicates that incorrect parity along the length of the tape was detected on data transferred.

The detection of this condition can occur during either a read or write operation. At the termination of the execution of an operation for which this condition was detected, the Longitudinal Tape Parity - Data Alert status will be reflected to the external user system.

f) End of Tape Mark 1XXXXX

The detection of the end of tape reflective mark will occur only during a write or erase operation. It does not cause the termination of the operation. The operation will proceed to its normal termination, at which time the End of Tape Mark - Data Alert status will be reflected to the external user system. This status is transmitted to the EUS at termination of the first write or erase operation following detection of EOT marker.

g) Bit Detected During Erase Operation XXXX11

The detection of this condition during an erase operation will not cause the termination of the erase operation. The Bit Detected During Erase Operation - Data Alert status will be reflected to the external user system at the normal termination of an Erase or Write End of File operation.

6.5.5 End of File (0100)

This major status return is reflected to the external user system at the termination of any one of the following commands:

a) Read command, where the record read was an End of File record (as defined in section 8).

b) Backspace or Forward Space Record command, where the record spaced over was an End of File record.

c) Backspace File command, unless no End of File record is encountered before tape moves to the load point.

d) Forward Space File command, unless no End of File record is encountered before tape is pulled off the supply reel.

6.5.5 End of File (0100) continued

The End of File major status is always accompanied by a substatus character. If the command execution which resulted in the End of File status was completed with no Data Alert condition occurring, this substatus character will be the bit configuration of the single character which comprised the single character End of File record, if that character contained six significant bits. Note that if either of the two high-order bits are ONE for 9-channel tape, the eight-bit byte read from tape is returned to the EUS as a single character record.

If a Data Alert condition was detected during the command execution which resulted in the End of File status, the substatus character will be an octal 77.

Lateral parity error detection is suppressed during the reading of any single character End of File record on a seven track tape unit.

The End of File status also will be returned to the external user system in response to any Request Status command which directly follows an End of File termination.

6.5.6 Command Reject (0101)

This major status condition indicates an unacceptable command was received and rejected by the subsystem. At the time this status condition is sent, substatus will be transmitted as follows:

a) Invalid Operation Code 000XX1

This status is reflected to the external user system when an operation code, not recognizable by the tape subsystem, is received.

b) Invalid Device Code 000X1X

This status is reflected to the external user system whenever a Device Code is detected which is illegal. The legal and illegal Device Codes are defined in section 3.2.

c) Invalid IDCW Parity

This status is reflected to the external user system whenever even Parity is detected on any byte of an IDCW transfer to the controller, or on the byte containing the logical channel number.

d) Positioned at BOT 001000

This status is reflected to the external user system whenever a Backspace type command is issued to a handler with tape positioned at the BOT marker.

6.5.6 Command Reject (0101) continued

e) Forward Read After Write 010000

This status is reflected to the external user system to indicate that a read command was rejected due to the previously addressed handler being in the write mode and that records following the record written should be considered invalid by the external user system.

f) Nine Track Error 100000

This status is reflected to the external user system to indicate that a Nine Track command was issued to a seven track handler.

6.5.7 MPC Device Attention (1010)

This major status indicates a condition requiring manual intervention has occurred within the addressed device. At the time this status condition is sent to the external user system, substatus will be transmitted as follows:

a) Configuration Switch Error 000001

This error condition will occur if the application firmware loaded into a controller does not agree with the configuration switches on the controller operator panel.

b) Multiple Devices 000010

This substatus indicates that the controller has detected at least two devices with the same identification number.

c) Illegal Device I.D. Number 001000

At least one device has an identification number which is outside the range of legal device numbers as defined in Section 3.2.

d) Incompatible Mode 001000

This status indicates that the handler mode (PE or NRZI) and the tape mode do not agree. This substatus may occur in response to a read command issued when the tape is positioned at the BOT marker. If the handler is set to one of the NRZI densities, the controller shall determine if a PE-Burst is present. If it is, this status shall be sent to the EUS and the tape positioned in the gap between the BOT marker and the first record. If the handler is set

6.5.7 MPC Device Attention (1010) continued

to the PE mode, the controller shall determine if the PE-Burst is present. If it is not, this status is sent to the EUS and the tape is positioned in the gap between the BOT marker and the first record.

e) TCA Malfunction 0011XX

This status indicates that a fault has been detected within one of the TCAs connected as a part of the controller. The two low-order bits of the substatus indicate the IAI port to which the malfunctioning TCA is connected.

Port	Substatus
0	000100
1	000101
2	000110
3	000111

f) MTH Malfunction 010000

This substatus indicates that the controller has detected an apparent malfunction with the handler, and the handler did not indicate a malfunction to the controller.

g) Multiple BOT 010001

This substatus indicates that additional BOT markers have been detected after the first one with tape moving in the forward direction.

6.5.8 MPC Device Data Alert (1011)

This major status indicates to the external user system that some type of alert condition occurred during the last command execution. At the time this status is sent to the external user system, substatus will be transmitted as follows:

a) Transmission Parity Error 000001

This substatus indicates that a transmission parity error was detected during the execution of a special controller command defined in Sections 5.5 and 5.6.

b) Inconsistent Command 000010

This substatus is the result of one of the following conditions occurring during the execution of a Special Controller Command.

- o Word count for Read Controller Main Memory, Write Controller Main Memory, and Write Control Store commands is ZERO.
- o Execution of Read Controller Main Memory or Write Controller Main Memory command will require access to non-existant memory.
- o Lock Byte number specified is illegal.
- o The continue bit in the IDCW for the Special Controller Command is ZERO.

c) Sum Check Error 000011

This substatus is the result of an error in the sum check out used by the Write Control Store command.

d) Byte Locked Out 000100

The Lock Byte referenced by the Conditional Write Lock Byte command is non-ZERO.

6.5.8 MPC Device Data Alert (1011) continued

e) PE-Burst Write Error 001000

This substatus indicates that the controller was not able to properly write the PE-Burst on tape.

f) Preamble Error 001001

This status indicates that an error was detected either with the preamble, or with an apparant lack of data following this preamble. This status is valid only for PE tapes.

g) T&D Error 001010

This status is used with the Device Wraparound command to indicate that the byte in error, and byte count have been stored in Main Memory.

The next five (5) substatus codes are related in that they may occur simultaneously within either the PE or NRZI mode. The highest priority code is described first; the lowest priority code is described last.

h) Multi-track Error 010000

This status indicates that a data record contained errors in more than one track. This condition may occur in either the PE mode or the NRZI mode (CCC on 800 bpi 9-track only).

i) Skew Error 010001

This substatus indicates that excessive skew was detected during a read or write in PE mode, or during a write in the NRZI mode.

j) Postamble Error 010010

This substatus indicates that the postamble of the PE record read may have been in error. The error may have occurred in the data portion of the PE record such that a postamble appeared to be present, or that errors occurred when entering the postamble such that the data appeared to continue beyond the data portion of the PE record; and the postamble was not properly detected.

k) NRZI CCC Error 010011

This substatus is used to indicate to the external system that the record just read contained correctable errors and may be reread.

6.5.8 MPC Device Data Alert (1011) continued

k) Continued

The controller will perform error correction on the data during the next forward read issued to the handler. If automatic retry is turned off (see section 7.3), the EUS must follow the command containing this substatus with a Backspace One Record followed by a read command with the IDCW Channel Instruction field set to 30_8. The controller will hold the track-in-error information during this command sequence so that error-correction may be attempted on the reread. Valid for 9-track, 800 bpi NRZI only.

l) Code Alert 010100

This substatus indicates that a character has been detected that is not in the code translation tables. This status is valid only during those read commands requiring code translation.

End of priority codes.

6.5.9 MPC Command Reject (1101)

This major status indicates that an unacceptable command was received and rejected by the subsystem. At the time this status condition is sent, substatus will be transmitted as follows:

a) Illegal Procedure 000001

This substatus indicates that the channel program is in error due to one of the following conditions.

- o The controller was not in the suspended state when one of the following commands was received by the controller;

  Write Controller Main Memory
  Write Control Store
  Executive Control Store Microprogram

- o A Special Controller Command did not preceed the Initiate Write Data Transfer command, or did not preceed the Initiate Read Data Transfer command.

b) Illegal L.C. Number 000010

This substatus indicates that the logical channel number preceeding the IDCW is not valid.

6.5.9 MPC Command Reject (1101) continued

c) Illegal Suspended L.C. Number 000011

This substatus occurs when the controller is suspended, and an IDCW is addressed to a logical channel other than the one over which the Suspend Controller command was issued.

6.6 CONTROLLER USE OF "STORE SPECIAL STATUS"

The Special Status is used to transmit information identifying which device is causing the interrupt, and the reason for the interrupt. This status is used in two different situations -- normal mode, and test mode. The mapping of the status information is different for these two modes, and will be explained in detail in the following sections.

The format for special status consists of four bytes, of which the first must be all ZEROs because the PSIA will OR other bits with the ZERO bits supplied by the controller. The second byte contains the number of the device causing the interrupt. The high-order bit or byte two (third byte) indicates if the status format is to be interpreted as that of the normal mode or of the test mode. The balance of the third byte, and all of byte are defined per the mode considered.

A set sequence must occur when these four special status bytes are to be sent to the system. The sequence begins with the controller sending the Store Special Status service code to the EUS. Following this service code, the four bytes of special status are sent to the EUS. The sequence is ended by the controller transmitting the "Set Special Interrupt" service code. This six step sequence must occur each time a special interrupt condition is transmitted to the EUS.

This status information will be stored in the ASCII mode. This mode is inherent in the Store Special Status service code -- the PSIA will assume this mode in response to this service code.

6.6.1 Special Status-Normal Mode

This mode of Special Status is used by the handlers to notify the external user system of special events which have occurred. There are six reasons for handler interrupt, and three reasons for controller interrupt.

1) Termination of Rewind Command
2) Termination of Rewind/Unload Command
3) Termination of Tape Load Command
4) Manually bringing a handler to the ready state
5) Device released
6) Controller has been suspended

6.6.1 Special Status-Normal Mode (continued)

7) Controller has been released from suspended state
8) Handler has malfunctioned
9) ITRs overlayed

The special status information for conditions one through three and eight and nine are directed to the PSIA which initially transmitted the command to the controller. If this LA/PSIA path is busy, the special status shall be transmitted when this path becomes available.

The spacial status for conditions 4, 6, and 7 will be sent to all operable PSI channels.

The special status for condition 5 will be sent to those PSI channels which had attempted to access the reserved handler during the period that the handler was reserved.

Table 6.6.1 illustrates the mapping of the status bytes, and the status bits are defined as follows:

Status Bits Description

1:3 - 1:7 These five bits define the handler causing the interrupt, with the least significant bit of this number in bit position 1:7.

2:0 Must be ZERO to define Special Status for the normal mode.

2:6 This bit is set to ONE when the controller is released from the suspended state (see section 5.6.5.2).

2:5 Set to ONE following successful overlay of ITRs(See section 5.4.23)

2:7 This bit is set to ONE when the controller enters the suspended state (see section 5.6.5.1).

3:1 This bit is set to ONE when a rewind is completed.

3:2 This bit is set to ONE when a handler has completed an unload operation.

3:3 When the handler completes an operation which results in a special interrupt with handler ready, this bit is set to ONE. This includes completion of the Rewind command, manually bringing the handler from the not-ready to the ready state, and the completion of the Load Tape command.

Status Bits — Description

3:4 This bit is set to ONE when the handler enters the standby state.

3:5 This bit is a modifier to bit 2:3.

ZERO - Standby Unloaded which is the state of the handler at the completion of a Rewind/Unload command. Handler may be brought ready by manual intervention or by completion of a Tape Load command.

ONE - Standby Loaded which is the state of the handler with the tape loaded, and positioned anywhere. The handler may be brought to the ready state only by manual intervention.

3:6 The handler released bit is used to indicate that the handler previously reserved has been released to all channels. The bit is then in the ONE state.

3:7 This status bit is valid only in response to a Rewind, Load Tape, or Rewind/Unload command. If the addressed handler malfunctions after the command is terminated with Ready status, and before the tape movement is completed normally, special status shall be sent to the EUS which had issued the command. The status shall indicate that the handler malfunctioned in some manner, and that the command was not executed correctly.

TABLE 6.6.1 SPECIAL STATUS BIT DEFINITION FOR NORMAL MODE

SPECIAL STATUS (WORD BIT)	SPECIAL STATUS (BYTE:BIT)	BIT DESCRIPTION
0		
1	0:0	MBZ (Must be Zero)
2	0:1	MBZ
3	0:2	MBZ
4	0:3	MBZ
5	0:4	MBZ
6	0:5	MBZ
7	0:6	MBZ
8	0:7	MBZ
9		
10	1:0	MBZ
11	1:1	MBZ
12	1:2	MBZ
13	1:3	Handler Number MSB
14	1:4	Handler Number
15	1:5	Handler Number
16	1:6	Handler Number
17	1:7	Handler Number LSB
18		
19	2:0	MBZ
20	2:1	MBZ
21	2:2	MBZ
22	2:3	MBZ
23	2:4	MBZ
24	2:5	ITRs Overlayed
25	2:6	Controller Released
26	2:7	Controller Suspended
27		
28	3:0	MBZ
29	3:1	Rewind Completed
30	3:2	Unload Completed
31	3:3	Handler Ready
32	3:4	Handler in Standby
33	3:5	Standby-Loaded
34	3:6	Handler Released
35	3:7	Handler Malfunction

6.6.2 Special Status -Test Mode

This mode of special status may be used to relay particular information from the controller to the external user system. The information is set into the MPC configuration switches 1 - 15, and then this information is transmitted to the EUS via special status by actuation of the Operator Interrupt Switch.

This special status will be sent to all connected PSI interfaces. The formatting of the status bytes is defined by Table 6.6.2.

TABLE 6.6.2 SPECIAL STATUS BIT DEFINITION FOR TEST MODE

SPECIAL STATUS (WORD BIT)	SPECIAL STATUS (BYTE:BIT)	BIT DESCRIPTION
0		
1	0:0	MBZ (must be Zero)
2	0:1	MBZ
3	0:2	MBZ
4	0:3	MBZ
5	0:4	MBZ
6	0:5	MBZ
7	0:6	MBZ
8	0:7	MBZ
9		
10	1:0	MBZ
11	1:1	MBZ
12	1:2	MBZ
13	1:3	MBZ
14	1:4	MBZ
15	1:5	MBZ
16	1:6	MBZ
17	1:7	MBZ
18		
19	2:0	ONE *
20	2:1	Configuration Switches **
21	2:2	
22	2:3	
23	2:4	
24	2:5	
25	2:6	
26	2:7	
27		
28	3:0	
29	3:1	
30	3:2	
31	3:3	
32	3:4	
33	3:5	
34	3:6	
35	3:7	

* This bit must be logical ONE to indicate the status is for the test mode.

** These are bit positions for the MPC configuration switches, with switch number 2 in bit position 2:1, and switch number 15 in bit position number 3:7.

7.0 ERROR DETECTION AND CORRECTION

This subsystem will utilize several modes of error correction/ detection, including:

- o Read-after-write checks during all write type commands.
- o Lateral parity on each data frame written on tape.
- o Longitudinal parity on each NRZI record.
- o Cyclic check character is generated and checked for each 9 track, 800 bpi NRZI record.
- o In-flight error correction for single track read errors in the PE mode.
- o Automatic retry on data-dependent read errors, and write mode errors not involving the media.
- o Verification of command, device number, IDCW format, and device type.
- o Parity generation and check on byte transfer across PSI.
- o Parity check within handler for command codes received from controller.
- o Handler monitors significant points within the drive mechanism, and notifies the controller when a malfunction is detected.

Some of the above-mentioned functions are performed by the handler, and the balance by the controller. Within the controller, the TCA will peform some error detection/correction functions, and the remaining by the controller microprogram, either independent of, or in cooperation with the TCA.

7.1 WRITE MODE ERRORS

When transferring data to the handler from the EUS several types of errors may occur. Depending upon the specific error, particular action is required by the subsystem. These conditions are summarized below to give an indication of the philosophy to be used in this controller.

7.1 WRITE MODE ERRORS (continued)

ERROR TYPE	DETECTED BY	SUBSYSTEM ACTION
Data Parity Errors:		
IOM to PSIA	PSIA	Complete record
PSIA to LA	LA	" "
LA to MPC	MPC	Safe-store MPC registers and attempt to stop handlers
MPC to TCA	TCA	Complete record
TCA to MTH (lateral)	TCA	" "
TCA to MTH (longitudinal)	TCA	" "
Transfer Timing Errors:		
Short Terms:		
IOM to PSIA	TCA	Complete record
PSIA to LA	TCA	" "
LA to MPC	TCA	" "
MPC to TCA	TCA	" "
Long Term;		
LA to MPC	MPC	Safe-store MPC registers and attempt to stop handlers
IAI A/C Error:		
Command error, MPC to TCA	TCA	Immediate Normal Terminate
Skew Error:		
PE Read-after-write	TCA	Complete record
NRZI Read-after-write	TCA	" "

Some of these errors are included in those which will cause the controller to automatically retry the last command. These are described in section 7.4. The subsystem action noted as complete record means that the data transfer should continue to the end of the record, and termination status transmitted to the external user system.

7.2 READ MODE ERRORS

Several types of errors may occur when reading data from tape, and in transferring this data to the external user system. As described in section 7.4, many of these errors are such that the controller may automatically retry the command. This section should be considered in conjunction with section 7.4 as a description of the possible read errors, and the action required of the controller.

7.2 READ MODE ERRORS (continued)

ERROR TYPE	DETECTED BY	SUBSYSTEM ACTION
Data Parity Errors:		
MTH to TCA	TCA	Complete record
TCA to MPC	MPC	Safe-store MPC registers and attempt to stop handlers
MPC to LA	LA	Complete record
LA to PSIA	PSIA	" "
PSIA to IOM	IOM	" "
Non-transfer Timing Errors:		
Lateral Parity	TCA	Complete record
Longitudinal Parity	MPC	" "
Multi-track	TCA	" "
Skew-PE	TCA	" "
PE Preamble	MPC	" "
PE Postamble	MPC	" "
Transfer Timing Errors:		
Short Term;		
MTH to TCA	TCA	Complete record
TCA to MPC	TCA	" "
MPC to LA	TCA	" "
LA to PSIA	TCA	" "
Long Term;		
MPC to LA	MPC	Safe-store MPC registers and attempt to stop handlers

In all cases, the comment in the action column which says that the subsystem will complete the record should be understood to mean that the appropriate termination status is transmitted to the EUS upon completion of the command execution.

7.3 EXTERNAL USER SYSTEM ACTION DURING READ COMMAND RETRIES

The external user system should follow a particular sequence of command executions when issuing a read command, and when attempting to recover from errors resulting during the read command execution. The operation is uder the control of the EUS, and the pertinent control information is transmitted to the controller by way of the Channel Instruction field of the IDCW.

7.3 EXTERNAL USER SYSTEM ACTION DURING READ COMMAND RETRIES (continued)

In general, the EUS should permit the controller to automatically retry those types of errors for which the controller has particular capabilities. These are defined in Section 7.4, as is the technique to be used during the controller retry of the read command.

Under normal operating conditions, the EUS should issue the IDCWs for the read command with the Channel Instruction (CI) field set to 20_8. As shown in Section 5.2, this will permit the controller to retry the command if an error of the type listed in Section 7.4 should occur. If the controller cannot complete a good read operation after eight tries, the command will be terminated with appropriate status. At this point the EUS should reissue the read command, but with the CI field incremented to 21_8. Again, if this should not be completed successfully, the controller will attempt eight tries of the command under the condition of the CI field.

Therefore, as the error recovery process progresses, a cooperative effort occurs between the EUS and the MTC. For each IDCW transmitted to the controller, up to eight tries will be made to execute the command as defined by the IDCW. The EUS should step throught the Channel Insutruction field codes starting with 20_8 and concluding with 27_8, for a total of 64 attempts at reading the record.

If the EUS wishes to keep the controller from performing any automatic retries, the Channel Instruction field should contain codes ranging from 30_8 to 37_8. As defined in Section 5.2, these CI codes direct the controller to try the command just one time.

7.4 AUTOMATIC RETRY IN SUBSYSTEM

The magnetic tape subsystem shall have the capability to retry when certain types of errors are detected within the subsystem, as well as when certain types of errors are detected on the PSI. These shall include the following:

Data Alert Errors Detected Within Subsystem;

a) Transmission Parity Alert

The LA detects an error on data transmitted across the PSI, including data for a normal command (section 5.3), but excluding IDCW parity errors and parity errors for the special controller commands (section 5.5).

b) Lateral Tape Parity Alert (Read mode only)

c) Longitudinal Tape Parity Alert (Read mode only)

7.4 AUTOMATIC RETRY IN SUBSYSTEM (continued)

d) Multi-track Error (implied PE read mode)

e) Skew Error(implied PE read mode)

f) PE Preamble Error (REad mode only)

g) PE Postamble Error (Read mode only)

h) Transfer Timing Error (Read mode only)

i) NRZI CCC Error

Each of the above conditions will result in the controller attempting the current operation up to eight (8) times, provided the Channel Instruction field of the IDCW contains codes between 20_8 and 27_8 (See section 5.2). If the operation is successful within this number of tries, the status to the EUS will be that of a normally successful command execution. If, after attempting eight tries the operation is still unsuccessful, the operation will be terminated with status indicating the reason for the unsuccessful execution.

If the external user system attempts a retry at this time, and the retry involves rereading a record, the Channel Instruction field of the IDCW may indicate that the low threshold mode should be used or that the capstain speed should be changed as described in section 5.2).

If the addressed handler is set to 800 bpi, and is a 9-channel unit, the retries by the controller should alternate between normal read execution and read execution with error correction.

If the addressed handler is other than 800 bpi, 9-channel, the read commands are executed as defined by the IDCW.

As a special case, the controller will perform three (3) retries on the following errors.

a) Parity errors on IDCW transfers if it is not the first IDCW in the channel program.

b) Parity errors on information transferred to the controller via the PSI during the execution of a special controller command (section 5.5).

7.5 ERROR LOGGING IN CONTROLLER

The controller shall maintain counts of specific errors which occur within the subsystem, and to a limited extent, for errors which occur on the PSI. These counts are initialized only when the controller is initialized; at other times the counts are allowed to accumulate and over-flow. The EUS may modify the error log contents by using particular commands. The accumulation of error counts is performed by the controller. The external user system has the responsibility to interrogate these counts whenever desired to determine the count for a particular period of time. This is possible by using appropriate commands as described in the following section and taking the difference between the current count and the previous count.

7.5.1 Errors Logged on a Controller Basis

There are eleven counters which are maintained in the controller Main Memory. The contents of these counters are available to the external user system by proper usage of the Read Controller Main Memory command. Also, the EUS may load these counters with any count by using the Write Controller Main Memory command. The location of these counters will be specified during the design implementation.

7.5.1.1 PSI Parity Error Counter

These one-byte counters record the number of activities containing an error while transferring information across the PSI from PSIA to LA. An activity includes either an IDCW transfer or a record transfer.

There are four of these counters - one for each possible PSI.

7.5.1.2 OPI/PSI Counter

These one-byte counters record the number of times that the OPI line on the PSI is dropped as the result of unsuccessful communication of the LA with the PSIA. The failure may be in the LA (where it appears as a PSI-related fault), in the PSI cable, or in the IOM/PSIA.

There are four of these counters-one for each possible PSI.

7.5.1.3 OPI/IAI Counter

These one-byte counters record the number of times that the OPI line is dropped due to either IAI fualts detected by the LA, or LA faults which appear as IAI faults.

There are two of these counters in the controller - one for each possible LA.

7.5.1.4. Error Interrupt Counter

This one-byte counter records the number of times that the MPC executed an error interrupt.

This is a single counter for the controller.

7.5.2 Errors Logged on a Handler Basis

There are seven counters which are maintained by the controller on an individual handler basis, which required a total of 112 counters for the maximum number of handlers. These counters do not necessarily contain counts of errors which have occurred in the handlers. Rather, they contain counts of error records, and counts of records transferred to/from the handlers.

In response to the read control register command being issued to a particular handler, eighteen bytes are transferred to the external user system. These bytes contain the contents of the counters for the addressed handler plus two diagnostic bytes. Prior to transferring this information to the EUS the controller must place the PSIA in the binary mode. The formatting of these eighteen bytes is illustrated in Figure 7.5 (byte zero transmitted first) while the counters are described in the following sections. Note that the EUS may set any count in these counters by using the Write Control Register command.

7.5.2.1 Number of Read Records

Two-byte counter called "Records Read Count". This counter contains a count of the number of physical records which have been read by the controller during the execution of read type commands.

7.5.2.2 Number of Write Records

Two-byte counter called "Records Written Count". This counter contains a count of the number of physical records which have been written by the controller during the execution of write-type commands.

7.5.2.3 Number of Records with Write Error

Two-byte counter called "Write Error Count". This counter contains a count of the number of physical records which were detected to have an error during the writing process.

7.5.2.4 Number of Records with Read Error

Two-byte counter called "Read Error Count". This counter contains a count of the number of physical records which were detected in error during the writing process.

7.5.2.5 Number of Transfer Timing Errors On Read

Two-byte counter called "Read TTE". This is a count of the number of transfer timing errors detected while transferring information during the read mode.

7.5.2.6 Number of Transfer Timing Errors on Write

Two-byte counter called "WRITE TTE". This counter contains a count of the number of transfer timing errors detected during the write mode.

7.5.2.7 Number of PE Records with Single Track Correction

Two-byte counter called "Single Track Correction". This counter contains the number of PE records which were successfully read, and which required that the in-flight single track correction be performed.

7.5.2.8 Save Area for Diagnostics

This two-byte area is used to store the byte count and saved byte resulting from execution of the following commands:

Diagnostic Mode Control	(section 5.4.41)
Device Wraparound	(section 5.4.42)
Write Timing Character	(section 5.4.43)

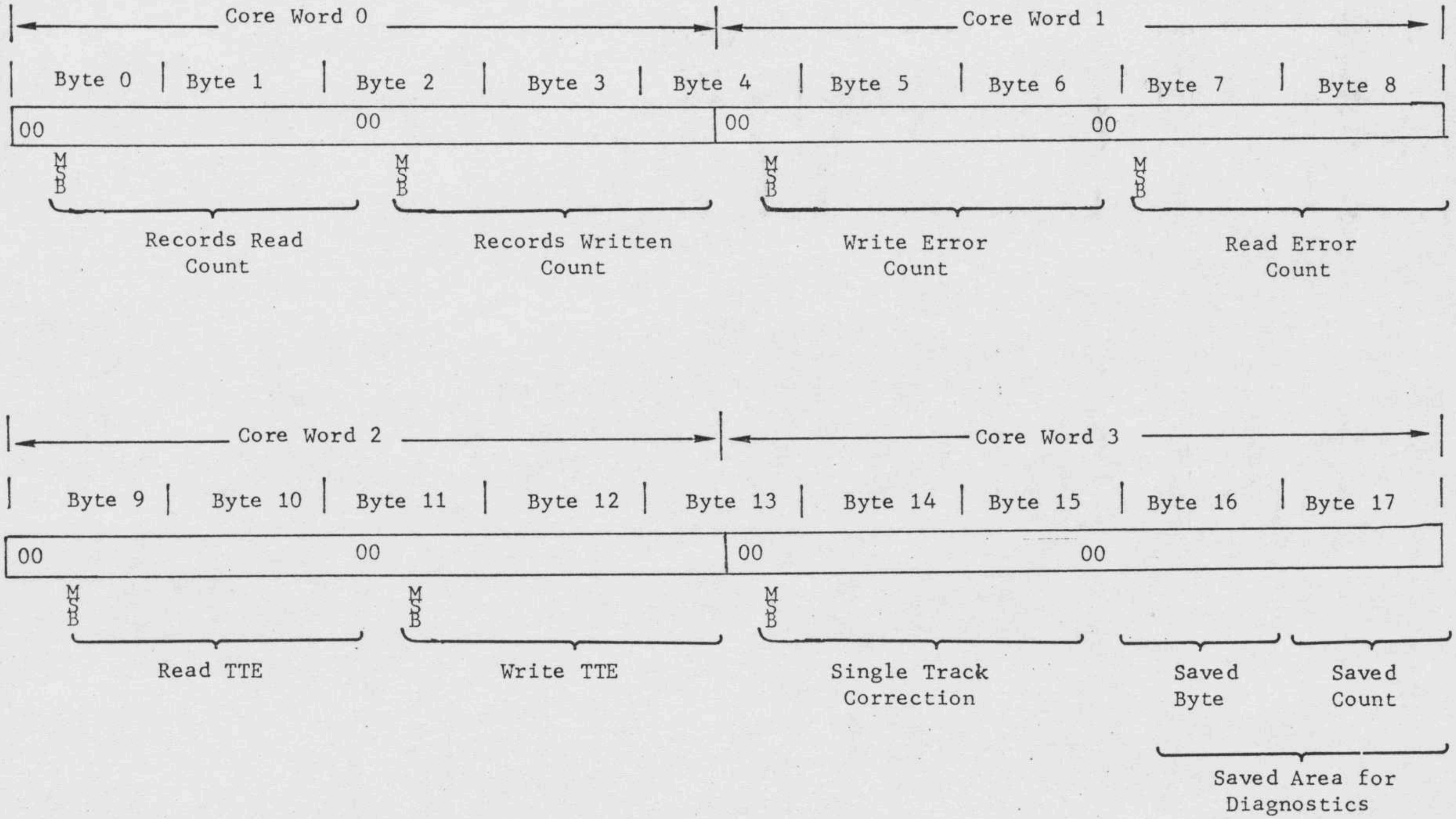


Figure 7.5

Error Counters Logged on a Handler Basis

8.0 RECORDING FORMATS

8.1 RECORDING TECHNIQUES

The recording of digital data on magnetic tape has progressed through several techniques to the present NRZI and PE. The following sections summarize the characteristics of these recording methods.

8.1.1 Non-Return to Zero, Change on Ones - NRZI

NRZI, or "non-return to zero, change on ones", has a flux transition for each one, with no transition for a zero (Figure 8.1). With NRZI, an error is limited to the single bit and does not change the subsequent data.

Data formats are arranged so that at least one flux transition takes place on one of the 7 or 9 parallel tracks for every bit cell. Thus, a clock is provided without assigning a separate track (see Figure 8.1).

8.1.2 Phase Encoding - PE

Phase encoding is a saturation recording technique which combines the clock and data to provide at least one flux transition for each bit cell, and thus achieves self-clocking on a single track. This technique is illustrated in Figure 8.1.

A ONE bit is defined as a flux change to agree with the polarity of the flux contained in the interrecord gap; a ZERO is defined as a flux change to the polarity opposite to the interrecord gap/polarity

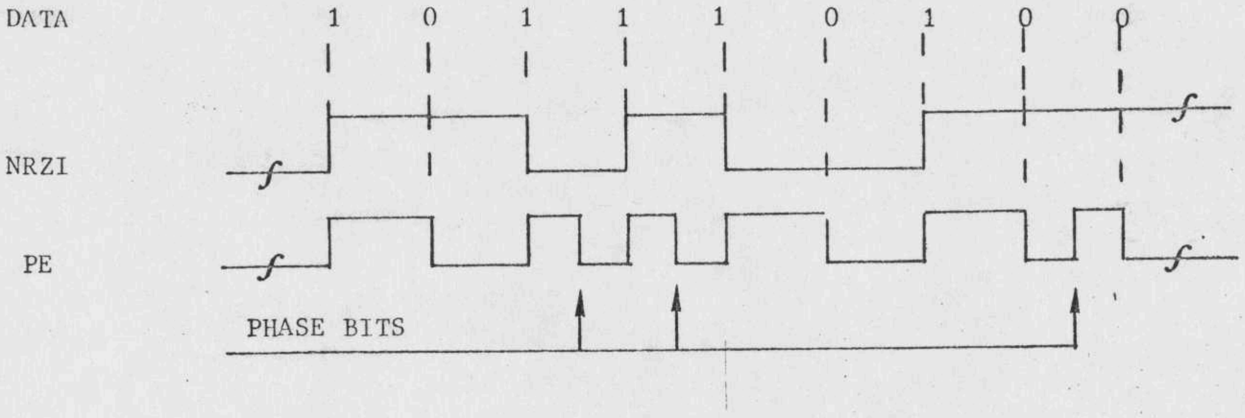


FIGURE 8.1

RECORDING TECHNIQUES

8.1.3 Bit Weights for Tape Tracks

The designation for magnetic tape bit weights differs between seven track and nine track handlers.

Seven Track (NRZI only)

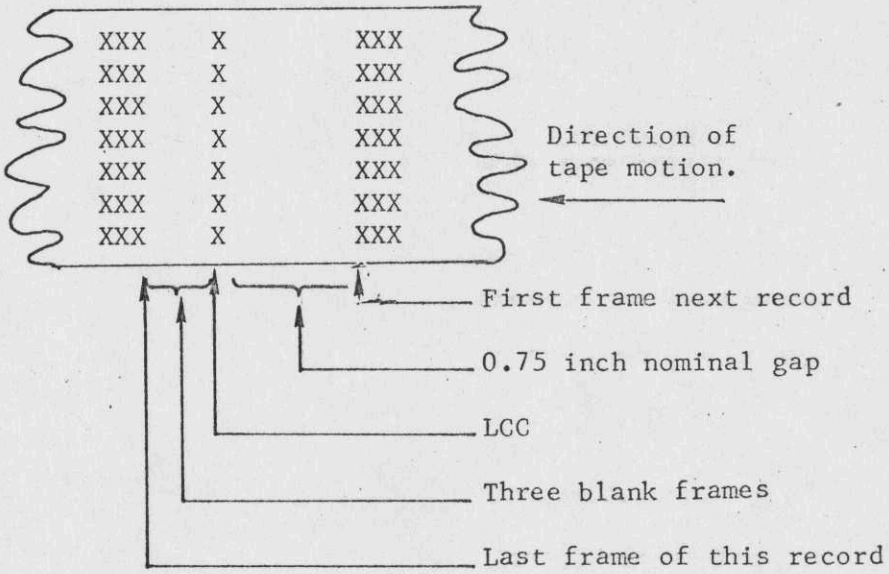
Track No. from Reference Edge			7	6	5	4	3	2	1
Bit Weight			P	2^5	2^4	2^3	2^2	2^1	2^0
Controller Bits	0	1	P	2	3	4	5	6	7

Nine Track (NRZI and PE)

Track No. from Reference Edge	9	8	7	6	5	4	3	2	1
Bit Weight	2^3	2^1	2^7	2^6	2^5	P	2^4	2^0	2^2
Controller Bits	4	6	0	1	2	P	3	7	5

8.2 NRZI SEVEN TRACK

8.2.1 Seven Track Record Format



8.2.2 Seven Track Parity

For data written on seven track tape, lateral parity in all BCD operations shall be even ONE's, and for all binary operations shall be odd ONE's.

8.2.3 Seven Track Longitudinal Parity

After the last data frame of the record has been written, three blank frames shall be generated, and then a longitudinal parity check (LCC) frame generated. Each bit of this LCC represents the even ONE's parity bit for the longitudinal track in which it appears.

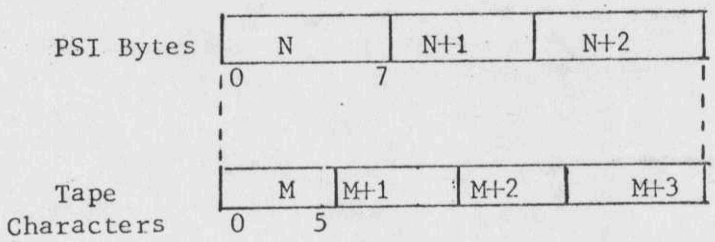
The magnetic polarity on tape after the LCC frame has been written must be the same as the gap polarity.

The lateral parity of the LCC is ignored.

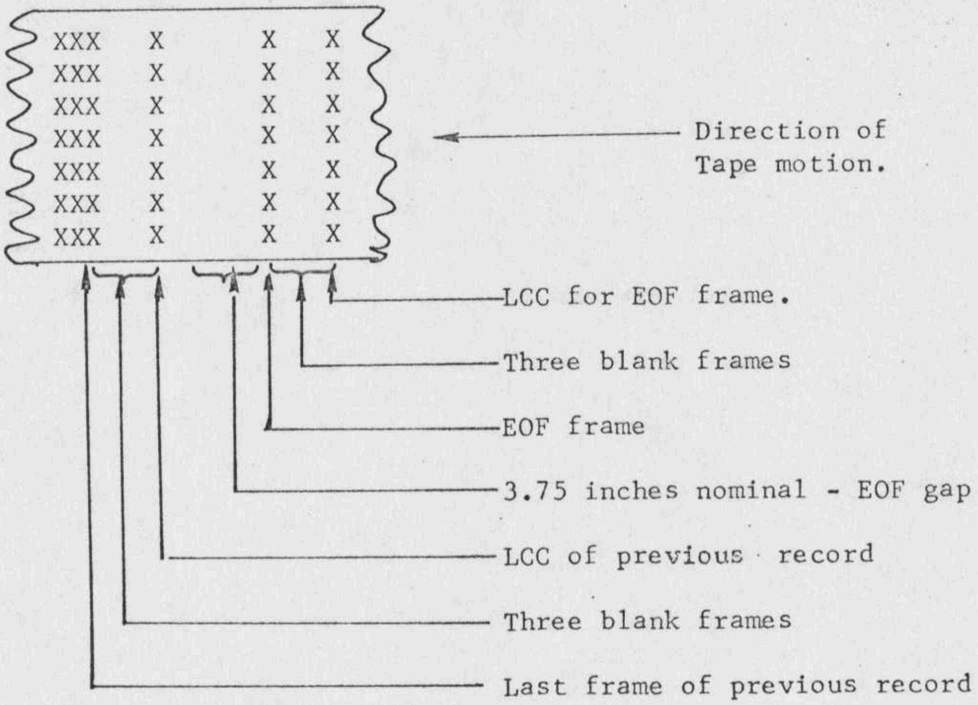
8.2.4 Pack/Unpack Related to Seven Track Data

Since all data transferred over the PSI is in the 8-bit byte format, a pack/unpack operation must occur when transferring data from/to a seven track handler. That is, when writing data to a seven track handler, the controller must unpack the 8-bit bytes received from the PSI, and transfer 6-bit frames to the handler. In the read mode, the 6-bit frames read from tape must be packed into 8-bit bytes before they are placed on the PSI for transmission to the external user system. This is illustrated in the following sketch.

8.2.4 Pack/Unpack Related to Seven Track Data - continued



8.2.5 Seven Track End-of-File



There are two methods available for writing a tape mark on the EOF frame in seven track handlers. The first is by use of the Write End of File command. Upon receipt of the IDCW specifying this command, the controller will cause an octal 17, with even parity, to be written to the designated seven track handler. The second method allows any 6-bit character to be written as an EOF, or file mark. In this case, bits 24-29 of the IDCW contains the 6-bit character to be written as a single frame record. The controller shall format this single character as an EOF frame as defined by the command (e.g., binary or BCD).

8.2.5 Continued

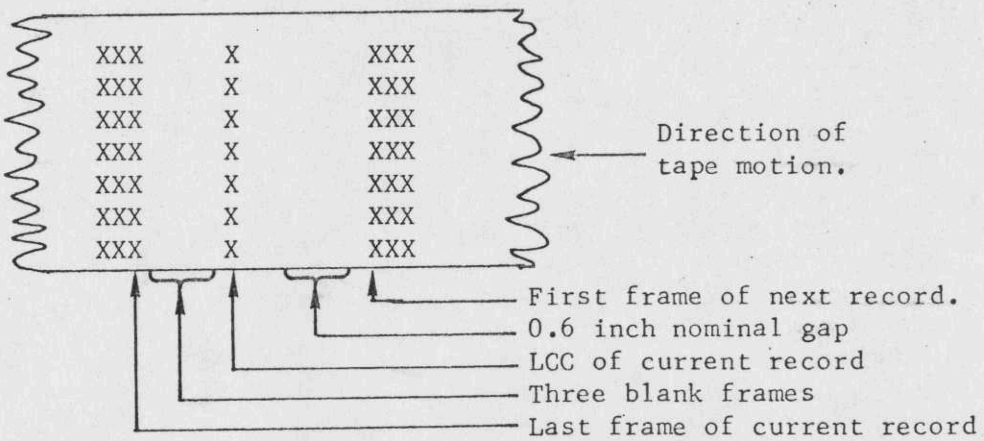
When reading seven track tape any single frame record shall be considered to be an EOF record. The reading of such a record shall result in the End of File status, with the six-bit character read forming the substatus.

8.2.6 NRZI Seven Track Blank Tape on Read

When a tape read command is initiated and no data is encountered within 25 feet, the Blank Tape on Read error status is transmitted to the EUS.

8.3 NRZI NINE TRACK (200 AND 556 BPI)

8.3.1 NRZI Nine Track Record Format (200 and 556 bpi)



8.3.2 NRZI Nine Track Parity (200 and 556 bpi)

The lateral parity shall be odd ONE's for data, and may be odd or even for the LCC.

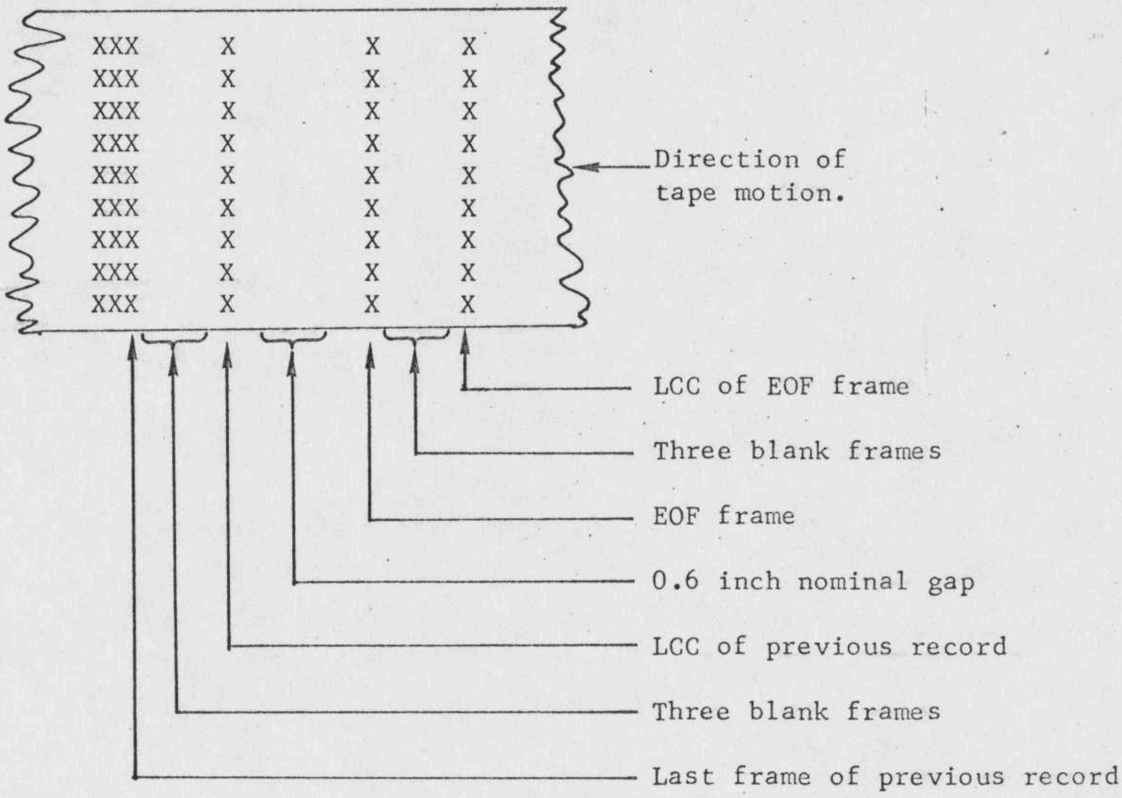
8.3.3 NRZI Nine Track Longitudinal Parity (200 and 556 bpi)

The longitudinal parity (LCC) for 200 and 556 bpi NRZI recording, shall be accomplished in the same manner as in the seven track NRZI mode (see Section 8.2.3).

8.3.4 Pack/Unpack Related to Nine Track NRZI (200 and 556 bpi)

The data transferred over the PSI is in 8-bit bytes, which corresponds to the data width for nine track tapes. Thus, there is no pack/unpack operation except for that required during code translation (see Sections 5.4.34 through 5.4.39), and for the Read BCD Record Command (Section 5.4.31 through 5.4.33).

8.3.5 NRZI Nine Track End-of-File (200 and 556 bpi)



There are three means for writing an End-of-File (EOF) frame for nine track NRZI tapes. The first method is via the Write End-of-File command. When the controller receives this command, the controller will cause an octal 23 to be written to the designated handler according to the above format. The second method of writing an EOF frame is for the IDCW to specify that the tally field contains a single 6-bit character to be written per the particular write command. The controller will append two high-order zeros, and write this 8-bit character in the EOF format. The third method is possible by a combination of Write Tape Nine command, an Initial Character Position specified in the PSIA, and a single byte transferred over the PSI. In this mode, the PSIA channel will send Terminate Out with this first byte of data, and the controller will write this one 8-bit byte as an EOF frame.

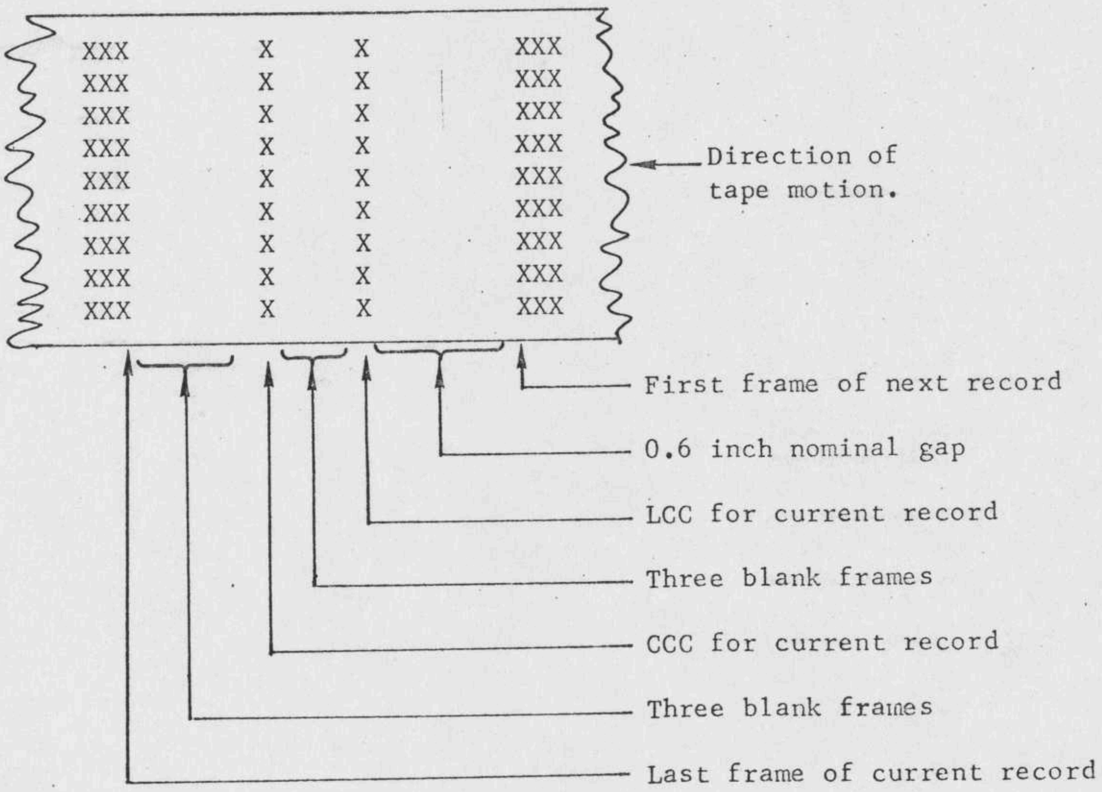
In NRZI recording, an EOF is defined as any single frame record followed at the proper distance by an LCC. If any single frame record is followed by three blank frames and then a second single tape frame, it may be recognized as an EOF frame as follows. If the two high-order bits are zeros, this frame will be recognized as an EOF frame. If the two high-order bits are not zeros, the 8-bit byte will be transferred to the PSIA as a single character record, followed by the normal command termination.

8.3.6 NRZI Nine Track Blank Tape on Read

When a tape read command is initiated and no data is encountered within 25 feet, the Blank Tape on Read error status is transmitted to the EUS.

8.4 NRZI NINE TRACK (800 BPI)

8.4.1 NRZI Record Format (800 bpi)



The record format for 800 bpi, nine track NRZI is shown above. The generation of the Cyclic Check Character (CCC) occurs within the tape controller adapter, and is described in the specification defining that unit. However, the timing of the format is under the control of the controller microprograms.

8.4.2 NRZI Nine Track Parity (800 bpi)

The lateral parity of the data characters for all nine track NRZI tapes is odd so that each frame shall contain an odd number of ONE's. Lateral parity of the CCC may be even or odd. Lateral parity of the LCC is odd.

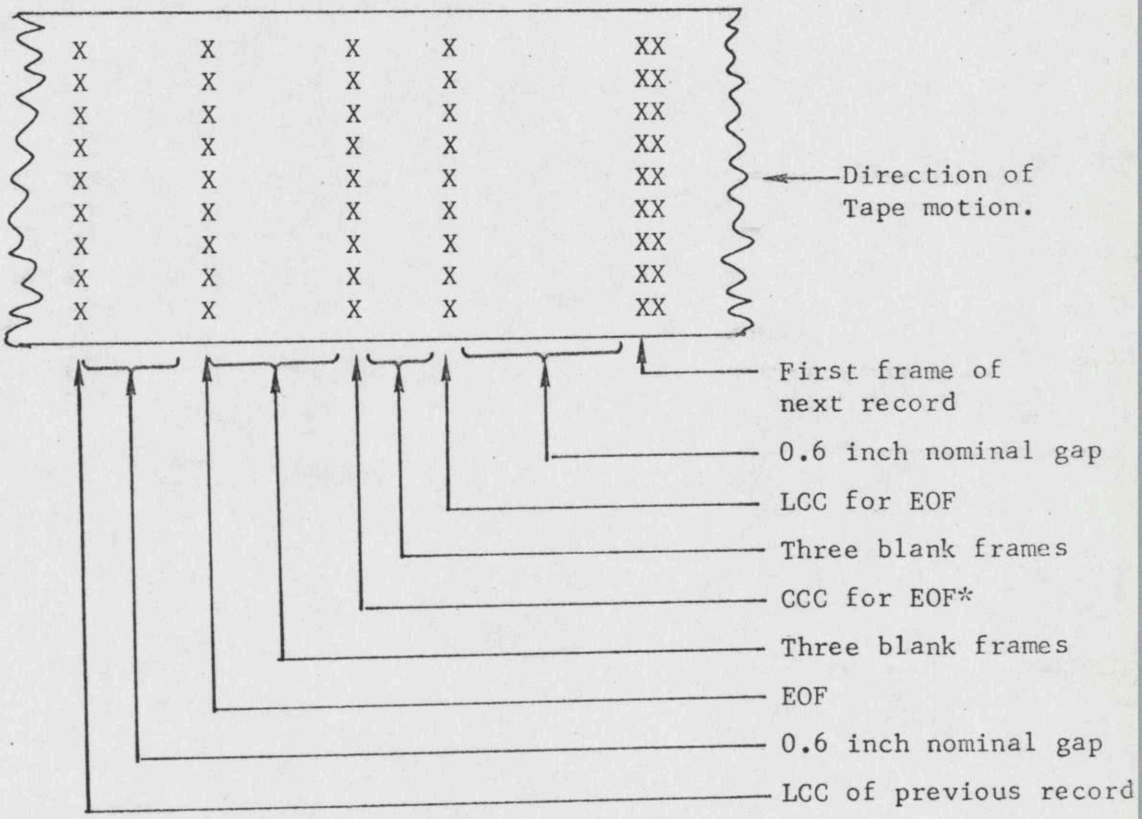
8.4.3 NRZI Nine Track Longitudinal Parity (800 bpi)

The longitudinal parity for 800 bpi NRZI tapes shall be accomplished in the same manner as for seven track NRZI and the nine track 200 and 556 bpi NRZI tapes. The placement differs in that the LCC for 800 bpi must be placed following the CCC with three intervening blank frames.

8.4.4 Pack/Unpack Related to Nine Track NRZI (800 bpi)

The pack/unpack requirements for 800 bpi, nine track NRZI tapes shall be the same as for other nine track NRZI tapes (see Section 8.3.4).

8.4.5 NRZI Nine Track End-of-File (800 bpi)



The three methods of writing an EOF frame at 800 bpi are the same as those for 200 and 556 bpi, nine track NRZI (Section 8.3.5).

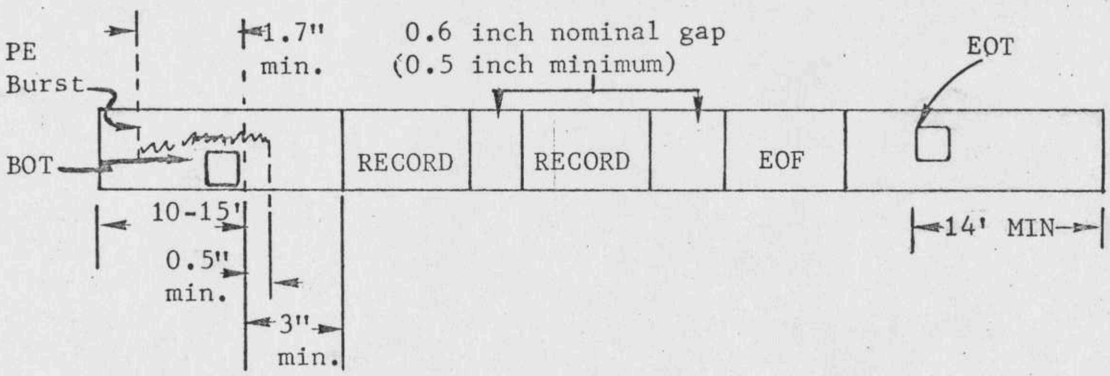
The considerations for reading EOF frames at 800 bpi with nine track NRZI are the same as those for 200 and 556 bpi (see Section 8.3.5) except for the CCC.

*When writing a standard EOF (23_8), CCC will be forced to all ZEROS. When writing a single character type EOF, CCC will be the normal computed value.

8.5 PE RECORDING

8.5.1 PE Tape Format

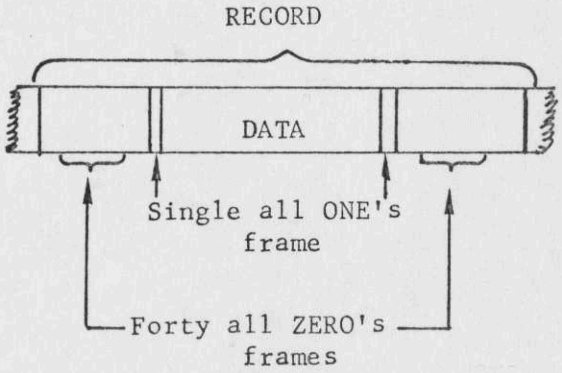
The general tape format for phase encoded (PE) recording is shown in the following diagram. The details of the recording formats for data records and EOF records are specified in the following sections.



PE Identification

The single track recording which starts prior to the BOT marker and extends beyond the BOT marker is called the PC Identification Burst. It is used in the identification of PE tapes by the controller. The PE Burst is written with constant flux reversals (under controller direction) at 1600 flux reversals per inch (FRI) in track 4. The other eight tracks are D-C erased.

8.5.2 PE Record Format



Each PE data record shall contain three parts: preamble, data, postamble. The preamble is written as forty-one frames, where forty frames contain all ZERO's including the parity position, followed by one frame which contains all ONE's. The data is then written as defined in Section 8.1.2. The postamble is written with a single all ONE's frame followed by forty all ZERO's frames

8.5.2 (Continued)

including the parity position. The formatting of the PE record is under the control of the MPC microprogram.

8.5.3 PE Record Parity

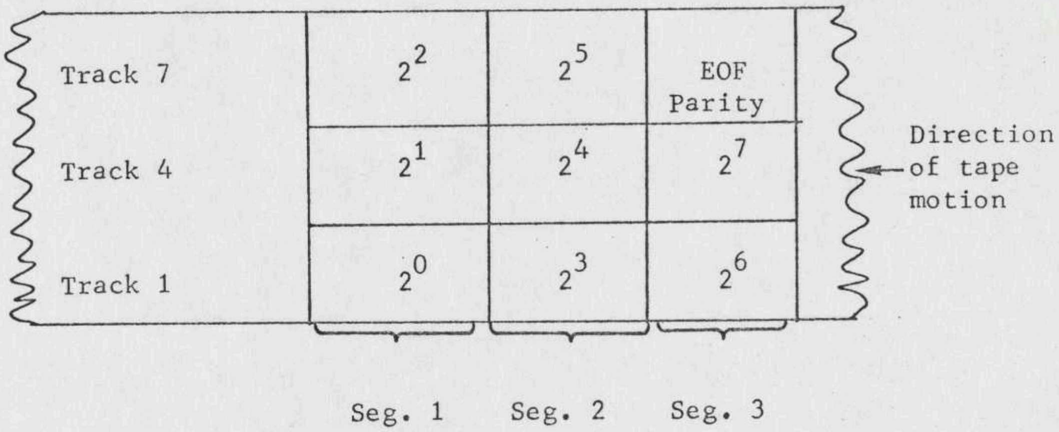
The data portion of the phase encoded record is recorded with odd parity. However, as described in Section 8.5.2, the preamble and postamble portions of the data record require special considerations.

8.5.4 Pack/Unpack Related to PE Data

The data transferred over the PSI is in 8-bit bytes as is the data written on phase encoded tapes. Thus, the only pack/unpack operations required are those which may be necessary during code translation (see Sections 5.4.32 through 5.4.39).

8.5.5 PE End-of-File Format

The PE End-of-File (EOF) does not contain a preamble or postamble like a normal PE record. It contains individual track information as follows. Tracks 2, 5 and 8 contain flux reversals at 3200 per inch. Tracks 3, 6 and 9 are DC erased. Tracks 1, 4 and 7 may contain flux reversals at 3200 per inch, or may be DC erased. This subsystem will generate a PE EOF as shown below for the optional tracks -- tracks 1, 4 and 7.



	Seg. 1	Seg. 2	Seg. 3
Track 7	2^2	2^5	EOF Parity
Track 4	2^1	2^4	2^7
Track 1	2^0	2^3	2^6

The EOF record, which this controller will write, shall be 240 flux reversals in length, and shall contain three sectors of 80 flux reversals each. The bit weight positions for the special EOF character are as indicated in the diagram.

8.5.5 (Continued)

There are three methods of writing a PE EOF record. The first is the result of a Write End-of-File command. In this case, the octal 23 shall be recorded in the EOF record. The second method is for the IDCW to specify that the tally field of the IDCW contains a single 6-bit character to be written on tape. The controller shall append two high-order zeros to this 6-bit character, and write the resulting 8-bit byte as defined by the above EOF diagram. The third method of writing an EOF mark is by a combination of the Write Tape Nine command, an Initial Character Position specified to the PSIA, and a single byte transferred over the PSI to the controller. In this case, the PSIA would send Terminate Out with the first byte of data, and the controller will write this byte as an EOF record per the above EOF format.

In reading end-of-file records in the PE mode, the subsystem must be able to recognize a variety of records as an EOF. As stated above, an EOF record must have tracks 2, 5 and 8 contain constant flux reversals at 3200 reversals per inch, with tracks 3, 6 and 9 DC erased. Tracks 1, 4 and 7 may be DC erased or recorded with flux reversals as are tracks 2, 5 and 8. The length of the EOF record may vary from 64 to 256 flux reversals in length (as defined in X331/400). Thus, this subsystem must be able to recognize (read) a variety of EOF types, and must react accordingly.

8.5.6 PE Blank Tape on Read

When a tape read command is initiated and no data is encountered within 25 feet, the Blank Tape on Read error status is transmitted to the EUS.

8.6 CODE TRANSLATION IN CONTROLLER

8.6.1 Tables for ASCII-EBCDIC Translation

The codes for translating from ASCII-7 to 8-bit EBCDIC, and vice versa, are included in the next two sections.

8.6.1.1 ASCII to EBCDIC Translation

The table for this translation follows as Table 8.6.1.1

TABLE 8.6.1.1 ASCII TO EBCDIC

ASCII-7			EBCDIC		
GRAPHIC	BIT EQUIVALENT	OCTAL	HEXI-DECIMAL	BIT EQUIVALENT	GRAPHIC
0	00110000	60	F0	11110000	0
1	00110001	61	F1	11110001	1
2	00110010	62	F2	11110010	2
3	00110011	63	F3	11110011	3
4	00110100	64	F4	11110100	4
5	00110101	65	F5	11110101	5
6	00110110	66	F6	11110110	6
7	00110111	67	F7	11110111	7
8	00111000	70	F8	11111000	8
9	00111001	71	F9	11111001	9
A	01000001	101	C1	11000001	A
B	01000010	102	C2	11000010	B
C	01000011	103	C3	11000011	C
D	01000100	104	C4	11000100	D
E	01000101	105	C5	11000101	E
F	01000110	106	C6	11000110	F
G	01000111	107	C7	11000111	G
H	01001000	110	C8	11001000	H
I	01001001	111	C9	11001001	I
J	01001010	112	D1	11010001	J
K	01001011	113	D2	11010010	K
L	01001100	114	D3	11010011	L
M	01001101	115	D4	11010100	M
N	01001110	116	D5	11010101	N
O	01001111	117	D6	11010110	O
P	01010000	120	D7	11010111	P
Q	01010001	121	D8	11011000	Q
R	01010010	122	D9	11011001	R
S	01010011	123	E2	11100010	S
T	01010100	124	E3	11100011	T
U	01010101	125	E4	11100100	U
V	01010110	126	E5	11100101	V
W	01010111	127	E6	11100110	W
X	01011000	130	E7	11100111	X
Y	01011001	131	E8	11101000	Y
Z	01011010	132	E9	11101001	Z
a	01100001	141	81	10000001	a
b	01100010	142	82	10000010	b
c	01100011	143	83	10000011	c
d	01100100	144	84	10000100	d
e	01100101	145	85	10000101	e
f	01100110	146	86	10000110	f

TABLE 8.6.1.1 (Continued)

ASCII-7			EBCDIC		
GRAPHIC	BIT EQUIVALENT	OCTAL	HEXIDECIMAL	BIT EQUIVALENT	GRAPHIC
g	01100111	147	87	10000111	g
h	01101000	150	88	10001000	h
i	01101001	151	89	10001001	i
j	01101010	152	91	10010001	j
k	01101011	153	92	10010010	k
l	01101100	154	93	10010011	l
m	01101101	155	94	10010100	m
n	01101110	156	95	10010101	n
o	01101111	157	96	10010110	o
p	01110000	160	97	10010111	p
q	01110001	161	98	10011000	q
r	01110010	162	99	10011001	r
s	01110011	163	A2	10100010	s
t	01110100	164	A3	10100011	t
u	01110101	165	A4	10100100	u
v	01110110	166	A5	10100101	v
w	01110111	167	A6	10100110	w
x	01111000	170	A7	10100111	x
y	01111001	171	A8	10101000	y
z	01111010	172	A9	10101001	z
¦	01111100	174	6A	01101010	¦
`	01100000	140	79	01111001	`
#	00100011	43	7B	01111011	#
@	01000000	100	7C	01111100	@
:	00111010	72	7A	01111010	:
>	00111110	76	6E	01101110	>
?	00111111	77	6F	01101111	?
b̸	00100000	40	40	01000000	b̸
&	00100110	46	50	01010000	&
.	00101110	56	4B	01001011	.
(	00101000	50	4D	01001101	(
<	00111100	74	4C	01001100	<
-	00101101	55	60	01100000	-
$	00100100	44	5B	01011011	$
*	00101010	52	5C	01011100	*
)	00101001	51	5D	01011101	)
;	00111011	73	5E	01011110	;
'	00100111	47	7D	01111101	'
+	00101011	53	4E	01001110	+
/	00101111	57	61	01100001	/
,	00101100	54	6B	01101011	,
%	00100101	45	6C	01101100	%
=	00111101	75	7E	01111110	=
"	00100010	42	7F	01111111	"
!	00100001	41	4F 5A	01001111 01011010	| !

TABLE 8.6.1.1 (Continued)

ASCII - 7

GRAPHIC	BIT EQUIVALENT	OCTAL
[	01011011	133
]	01011101	135
^	01011110	136
_	01011111	137
\	01011100	134
{	01111011	173
}	01111101	175
~	01111110	176
	All Others	

EBCDIC

HEXI- DECIMAL	BIT EQUIVALENT	GRAPHIC
C0	11000000	{
D0	11010000	}
5F	01011111	¬
6D	01101101	_
E0	11100000	\
C0	11000000	{
D0	11010000	}
A1	10100001	~
FF	11111111	

8.6.1.2 EBCDIC to ASCII Translation

Table 8.6.1.2 specifies this translation.

EBCDIC			ASCII-7		
GRAPHIC	BIT EQUIVALENT	HEXIDECIMAL	OCTAL	BIT EQUIVALENT	GRAPHIC
0	11110000	F0	60	00110000	0
1	11110001	F1	61	00110001	1
2	11110010	F2	62	00110010	2
3	11110011	F3	63	00110011	3
4	11110100	F4	64	00110100	4
5	11110101	F5	65	00110101	5
6	11110110	F6	66	00110110	6
7	11110111	F7	67	00110111	7
8	11111000	F8	70	00111000	8
9	11111001	F9	71	00111001	9
A	11000001	C1	101	01000001	A
B	11000010	C2	102	01000010	B
C	11000011	C3	103	01000011	C
D	11000100	C4	104	01000100	D
E	11000101	C5	105	01000101	E
F	11000110	C6	106	01000110	F
G	11000111	C7	107	01000111	G
H	11001000	C8	110	01001000	H
I	11001001	C9	111	01001001	I
J	11010001	D1	112	01001010	J
K	11010010	D2	113	01001011	K
L	11010011	D3	114	01001100	L
M	11010100	D4	115	01001101	M
N	11010101	D5	116	01001110	N
O	11011001	D6	117	01001111	O
P	11010111	D7	120	01010000	P
Q	11011000	D8	121	01010001	Q
R	11011001	D9	122	01010010	R
S	11100010	E2	123	01010011	S
T	11100011	E3	124	01010100	T
U	11100100	E4	125	01010101	U
V	11100101	E5	126	01010110	V
W	11100110	E6	127	01010111	W
X	11100110	E7	130	01011000	X
Y	11101000	E8	131	01011001	Y
Z	11101001	E9	132	01011010	Z

TABLE 8.6.1.2 EBCDIC TO ASCII

8.6.1.2 EBCDIC TO ASCII Translation (continued)

Table 8.6.1.2 (continued)

EBCDIC			ASCII-7		
GRAPHIC	BIT EQUIVALENT	HEXI-DECIMAL	OCTAL	BIT EQUIVALENT	GRAPHIC
a	10000001	81	141	01100001	a
b	10000010	82	142	01100010	b
c	10000011	83	143	01100011	c
d	10000100	84	144	01100100	d
e	10000101	85	145	01100101	e
f	10000110	86	146	01100110	f
g	10000111	87	147	01100111	g
h	10001000	88	150	01101000	h
i	10001001	89	151	01101001	i
j	10010001	91	152	01101010	j
k	10010010	92	153	01101011	k
l	10010011	93	154	01101100	l
m	10010100	94	155	01101101	m
n	10010101	95	156	01101110	n
o	10010110	96	157	01101111	o
p	10010111	97	160	01110000	p
q	10011000	98	161	01110001	q
r	10011001	99	162	01110010	r
s	10100010	A2	162	01110011	s
t	10100011	A3	164	01110100	t
u	10100100	A4	165	01110101	u
v	10100101	A5	166	01110110	v
w	10100110	A6	167	01110111	w
x	10100111	A7	170	01111000	x
y	10101000	A8	171	01111001	y
z	10101001	A9	172	01111010	z
#	01111011	7B	43	00100011	#
@	01111100	7C	100	01000000	@
:	01111010	7A	72	00111010	:
>	01101110	6E	76	00111110	>
?	01101111	6F	77	00111111	?
ƀ	01000000	40	40	00100000	ƀ
&	01010000	50	46	00100110	&
.	01001011	4B	56	00101110	.
(	01001101	4D	50	00101000	(
<	01001100	4C	74	00111100	<
-	01100000	60	55	00101101	-

8.6.1.2 EBCDIC TO ASCII Translation (continued)

Table 8.6.1.2 (continued)

	EBCDIC			ASCII-7	
	BIT	HEXI-		BIT	
GRAPHIC	EQUIVALENT	DECIMAL	OCTAL	EQUIVALENT	GRAPHIC
$	01011011	5B	44	00100100	$
*	01011100	5C	52	00101010	*
)	01011101	5D	51	00101001	)
;	01011110	5E	73	00111011	;
'	01111101	7D	47	00100111	'
+	01001110	4E	53	00101011	+
/	01100001	61	57	00101111	/
,	01101011	6B	54	00101100	,
%	01101100	6C	45	00100101	%
=	01111110	7E	75	00111101	=
"	01111111	7F	42	00100010	"
!	01001111	4F	41	00100001	!

TABLE 8.6.1.2 (Continued)

EBCDIC

GRAPHIC	BIT EQUIVALENT	HEXI-DECIMAL
	01001010	4A
!	01011010	5A
¬	01011111	5F
_	01101101	6D
\	11100000	EO
{	11000000	CO
}	11010000	DO
~	10100001	A1
¦	01101010	6A
`	01111001	79
EO	11111111	FF
	All others*	

ASCII-7

OCTAL	BIT EQUIVALENT	GRAPHIC
377	11111111	EO
41	00100001	!
136	01011110	^
137	01011111	_
134	01011100	\
173	01111011	{
175	01111101	}
176	01111110	~
174	01111100	¦
140	01100000	`
377	11111111	EO
377	11111111	EO

* All other eight-bit codes on tape result in an octal 377 ASCII character, with terminate status MPC Device Data Alert - Code Alert.

8.6.2 Tables for Six-Bit - EBCDIC Translation

The tables for translating from the 6000 line 6-bit code to EBCDIC, and vice versa, are given in the following two sections.

8.6.2.1 Six-Bit to EBCDIC Translation

The table for translating from the six-bit code to EBCDIC is as follows:

6000 SIX-BIT CODE

GRAPHIC	BIT EQUIVALENT	OCTAL
0	000000	00
1	000001	01
2	000010	02
3	000011	03
4	000100	04
5	000101	05
6	000110	06
7	000111	07
8	001000	10
9	001001	11
A	010001	21
B	010010	22
C	010011	23
D	010100	24
E	010101	25
F	010110	26
G	010111	27
H	011000	30
I	011001	31
J	100001	41
K	100010	42
L	100011	43
M	100100	44
N	100101	45
O	100110	46
P	100111	47
Q	101000	50
R	101001	51
S	110010	62
T	110011	63
U	110100	64
V	110101	65
W	110110	66

EBCDIC

HEXI-DECIMAL	BIT EQUIVALENT	GRAPHIC
F0	11110000	0
F1	11110001	1
F2	11110010	2
F3	11110011	3
F4	11110100	4
F5	11110101	5
F6	11110110	6
F7	11110111	7
F8	11111000	8
F9	11111001	9
C1	11000001	A
C2	11000010	B
C3	11000011	C
C4	11000100	D
C5	11000101	E
C6	11000110	F
C7	11000111	G
C8	11001000	H
C9	11001001	I
D1	11010001	J
D2	11010010	K
D3	11010011	L
D4	11010100	M
D5	11010101	N
D6	11010110	O
D7	11010111	P
D8	11011000	Q
D9	11011001	R
E2	11100010	S
E3	11100011	T
E4	11100100	U
E5	11100101	V
E6	11100110	W

TABLE 8.6.2.1 SIX-BIT TO EBCDIC

TABLE 8.6.2.1 (Continued)

6000 SIX-BIT CODE

GRAPHIC	BIT EQUIVALENT	OCTAL
X	110111	67
Y	111000	70
Z	111001	71
#	001011	13
@	001100	14
:	001101	15
>	001110	16
?	001111	17
b̸	010000	20
&	011010	32
.	011011	33
(	011101	35
<	011110	36
-	101010	52
$	101011	53
*	101100	54
)	101101	55
;	101110	56
'	101111	57
+	110000	60
/	110001	61
,	111011	73
%	111100	74
=	111101	75
"	111110	76
!	111111	77
\	011111	.37
[	001010	12
]	011100	34
↑	100000	40
←	111010	72

EBCDIC

HEXIDECIMAL	BIT EQUIVALENT	GRAPHIC
E7	11100111	X
E8	11101000	Y
E9	11101001	Z
7B	01111011	#
7C	01111100	@
7A	01111010	:
6E	01101110	>
6F	01101111	?
40	01000000	b̸
50	01010000	&
4B	01001011	.
4D	01001101	(
4C	01001100	<
60	01100000	-
5B	01011011	$
5C	01011100	*
5D	01011101	)
5E	01011110	;
7D	01111101	'
4E	01001110	+
61	01100001	/
6B	01101011	,
6C	01101100	%
7E	01111110	=
7F	01111111	"
5A	01011010	!
E0	11100000	\
C0	11000000	{
D0	11010000	}
5F	01011111	¬
6D	01101101	_

8.6.2.2 EBCDIC to Six-Bit Translation

Table 8.6.2.2 gives the translation for EBCDIC to the 6000 line six-bit code.

EBCDIC			6000 SIX-BIT CODE		
GRAPHIC	BIT EQUIVALENT	HEXI-DECIMAL	OCTAL	BIT EQUIVALENT	GRAPHIC
0	11110000	F0	00	000000	0
1	11110001	F1	01	000001	1
2	11110010	F2	02	000010	2
3	11110011	F3	03	000011	3
4	11110100	F4	04	000100	4
5	11110101	F5	05	000101	5
6	11110110	F6	06	000110	6
7	11110111	F7	07	000111	7
8	11111000	F8	10	001000	8
9	11111001	F9	11	001001	9
A	11000001	C1	21	010001	A
B	11000010	C2	22	010010	B
C	11000011	C3	23	010011	C
D	11000100	C4	24	010100	D
E	11000101	C5	25	010101	E
F	11000110	C6	26	010110	F
G	11000111	C7	27	010111	G
H	11001000	C8	30	011000	H
I	11001001	C9	31	011001	I
J	11010001	D1	41	100001	J
K	11010010	D2	42	100010	K
L	11010011	D3	43	100011	L
M	11010100	D4	44	100100	M
N	11010101	D5	45	100101	N
O	11010110	D6	46	100110	O
P	11010111	D7	47	100111	P
Q	11011000	D8	50	101000	Q
R	11011001	D9	51	101001	R
S	11100010	E2	62	110010	S
T	11100011	E3	63	110011	T
U	11100100	E4	64	110100	U
V	11100101	E5	65	110101	V
W	11100110	E6	66	110110	W
X	11100111	E7	67	110111	X
Y	11101000	E8	70	111000	Y
Z	11101001	E9	71	111001	Z

TABLE 8.6.2.2 EBCDIC TO SIX-BIT CODE

TABLE 8.6.2.2 (Continued)

EBCDIC

GRAPHIC	BIT EQUIVALENT	HEXI-DECIMAL
a	10000001	81
b	10000010	82
c	10000011	83
d	10000100	84
e	10000101	85
f	10000110	86
g	10000111	87
h	10001000	88
i	10001001	89
j	10010001	91
k	10010010	92
l	10010011	93
m	10010100	94
n	10010101	95
o	10010110	96
p	10010111	97
q	10011000	98
r	10011001	99
s	10100010	A2
t	10100011	A3
u	10100100	A4
v	10100101	A5
w	10100110	A6
x	10100111	A7
y	10101000	A8
z	10101001	A9
#	01111011	7B
@	01111100	7C
:	01111010	7A
>	01101110	6E
?	01101111	6F
b̸	01000000	40
&	01010000	50
.	01001011	4B
(	01001101	4D
<	01001100	4C
\	11100000	E0
-	01100000	60
$	01011011	5B
*	01011100	5C
)	01011101	5D
;	01011110	5E
'	01111101	7D

6000 SIX-BIT CODE

OCTAL	BIT EQUIVALENT	GRAPHIC
21	010001	A
22	010010	B
23	010011	C
24	010100	D
25	010101	E
26	010110	F
27	010111	G
30	011000	H
31	011001	I
41	100001	J
42	100010	K
43	100011	L
44	100100	M
45	100101	N
46	100110	O
47	100111	P
50	101000	Q
51	101001	R
62	110010	S
63	110011	T
64	110100	U
65	110101	V
66	110110	W
67	110111	X
70	111000	Y
71	111001	Z
13	001011	#
14	001100	@
15	001101	:
16	001110	>
17	001111	?
20	010000	b̸
32	011010	&
33	011011	.
35	011101	(
36	011110	<
37	011111	\
52	101010	-
53	101011	$
54	101100	*
55	101101	)
56	101110	;
57	101111	'

TABLE 8.6.2.2 (Continued)

EBCDIC			6000 SIX-BIT CODE		
GRAPHIC	BIT EQUIVALENT	HEXI-DECIMAL	OCTAL	BIT EQUIVALENT	GRAPHIC
+	01001110	4E	60	110000	+
/	01100001	61	61	110001	/
,	01101011	6B	73	111011	,
%	01101100	6C	74	111100	%
=	01111110	7E	75	111101	=
"	01111111	7F	76	111110	"
|	01001111	4F	77	111111	!
!	01011010	5A	77	111111	!
¬	01011111	5F	40	100000	↑
_	01101101	6D	72	111010	←
}	11010000	D0	34	011100	]
{	11000000	C0	12	001010	[
EO	11111111	FF	77	111111	!
			77	111111	!
	All Other Codes*		77	111111	!

*All other eight-bit codes on tape result in an octal 77 six-bit character, with terminate status MPC Device Data Alert - Code Alert.

8.6.3 Tables for Six-Bit - ASCII Translation

The tables for translating from the 6000 line 6-bit code to ASCII-7, and vice versa, are given in the following two sections.

8.6.3.1 Six-Bit to ASCII Translation

The table for translating from the 6-bit code to ASCII-7 is as follows.

6000 SIX-BIT CODE			ASCII-7		
GRAPHIC	BIT EQUIVALENT	OCTAL	OCTAL	BIT EQUIVALENT	GRAPHIC
0	000000	00	60	00110000	0
1	000001	01	61	00110001	1
2	000010	02	62	00110010	2
3	000011	03	63	00110011	3
4	000100	04	64	00110100	4
5	000101	05	65	00110101	5
6	000110	06	66	00110110	6
7	000111	07	67	00110111	7
8	001000	10	70	00111000	8
9	001001	11	71	00111001	9
A	010001	21	101	01000001	A
B	010010	22	102	01000010	B
C	010011	23	103	01000011	C
D	010100	24	104	01000100	D
E	010101	25	105	01000101	E
F	010110	26	106	01000110	F
G	010111	27	107	01000111	G
H	011000	30	110	01001000	H
I	011001	31	111	01001001	I
J	100001	41	112	01001010	J
K	100010	42	113	01001011	K
L	100011	43	114	01001100	L
M	100100	44	115	01001101	M
N	100101	45	116	01001110	N
O	100110	46	117	01001111	O
P	100111	47	120	01010000	P
Q	101000	50	121	01010001	Q
R	101001	51	122	01010010	R
S	110010	62	123	01010011	S
T	110011	63	124	01010100	T
U	110100	64	125	01010101	U
V	110101	65	126	01010110	V
W	110110	66	127	01010111	W
X	110111	67	130	01011000	X
Y	111000	70	131	01011001	Y
Z	111001	71	132	01011010	Z

TABLE 8.6.3.1 SIX-BIT to ASCII-7

TABLE 8.6.3.1 (Continued)

6000 SIX-BIT CODE			ASCII-7		
GRAPHIC	BIT EQUIVALENT	OCTAL	OCTAL	BIT EQUIVALENT	GRAPHIC
#	001011	13	43	00100011	#
@	001100	14	100	01000000	@
:	001101	15	72	00111010	:
>	001110	16	76	00111110	>
?	001111	17	77	00111111	?
b̸	010000	20	40	00100000	b̸
&	011010	32	46	00100110	&
.	011011	33	56	00101110	.
(	011101	35	50	00101000	(
<	011110	36	74	00111100	<
-	101010	52	55	00101101	-
$	101011	53	44	00100100	$
*	101100	54	52	00101010	*
)	101101	55	51	00101001	)
;	101110	56	73	00111011	;
'	101111	57	47	00100111	'
+	110000	60	53	00101011	+
/	110001	61	57	00101111	/
,	111011	73	54	00101100	,
%	111100	74	45	00100101	%
=	111101	75	75	00111101	=
"	111110	76	42	00100010	"
!	111111	77	41	00100001	!
\	011111	37	134	01011100	\
[	001010	12	133	01011011	[
]	011100	34	135	01011101	]
↑	100000	40	136	01011110	^
←	111010	72	137	01011111	_

8.6.3.2 ASCII to Six-Bit Translation

The table for translating from ASCII-7 to the 6-bit code is as follows.

ASCII-7

GRAPHIC	BIT EQUIVALENT	OCTAL
0	00110000	60
1	00110001	61
2	00110010	62
3	00110011	63
4	00110100	64
5	00110101	65
6	00110110	66
7	00110111	67
8	00111000	70
9	00111001	71
A	01000001	101
B	01000010	102
C	01000011	103
D	01000100	104
E	01000101	105
F	01000110	106
G	01000111	107
H	01001000	110
I	01001001	111
J	01001010	112
K	01001011	113
L	01001100	114
M	01001101	115
N	01001110	116
O	01001111	117
P	01010000	120
Q	01010001	121
R	01010010	122
S	01010011	123
T	01010100	124
U	01010101	125
V	01010110	126
W	01010111	127
X	01011000	130
Y	01011001	131
Z	01011010	132

6000 SIX-BIT CODE

OCTAL	BIT EQUIVALENT	GRAPHIC
00	000000	0
01	000001	1
02	000010	2
03	000011	3
04	000100	4
05	000101	5
06	000110	6
07	000111	7
10	001000	8
11	001001	9
21	010001	A
22	010010	B
23	010011	C
24	010100	D
25	010101	E
26	010110	F
27	010111	G
30	011000	H
31	011001	I
41	100001	J
42	100010	K
43	100011	L
44	100100	M
45	100101	N
46	100110	O
47	100111	P
50	101000	Q
51	101001	R
62	110010	S
63	110011	T
64	110100	U
65	110101	V
66	110110	W
67	110111	X
70	111000	Y
71	111001	Z

TABLE 8.6.3.2 ASCII-7 to SIX-BIT

TABLE 8.6.3.2 (Continued)

ASCII-7

GRAPHIC	BIT EQUIVALENT	OCTAL
a	01100001	141
b	01100010	142
c	01100011	143
d	01100100	144
e	01100101	145
f	01100110	146
g	01100111	147
h	01101000	150
i	01101001	151
j	01101010	152
k	01101011	153
l	01101100	154
m	01101101	155
n	01101110	156
o	01101111	157
p	01110000	160
q	01110001	161
r	01110010	162
s	01110011	163
t	01110100	164
u	01110101	165
v	01110110	166
w	01110111	167
x	01111000	170
y	01111001	171
z	01111010	172
#	00100011	43
@	01000000	100
:	00111010	72
>	00111110	76
?	00111111	77
ƀ	00100000	40
&	00100110	46
.	00101110	56
(	00101000	50
<	00111100	74
-	00101101	55
$	00100100	44
*	00101010	52
)	00101001	51
;	00111011	73
'	00100111	47
+	00101011	53

6000 SIX-BIT CODE

OCTAL	BIT EQUIVALENT	GRAPHIC
21	010001	A
22	010010	B
23	010011	C
24	010100	D
25	010101	E
26	010110	F
27	010111	G
30	011000	H
31	011001	I
41	100001	J
42	100010	K
43	100011	L
44	100100	M
45	100101	N
46	100110	O
47	100111	P
50	101000	Q
51	101001	R
62	110010	S
63	110011	T
64	110100	U
65	110101	V
66	110110	W
67	110111	X
70	111000	Y
71	111001	Z
13	001011	#
14	001100	@
15	001101	:
16	001110	>
17	001111	?
20	010000	ƀ
32	011010	&
33	011011	.
35	011101	(
36	011110	<
52	101010	-
53	101011	$
54	101100	*
55	101101	)
56	101110	;
57	101111	'
60	110000	+

TABLE 8.6.3.2 (Continued)

ASCII-7

GRAPHIC	BIT EQUIVALENT	OCTAL
!	00100001	41
/	00101111	57
,	00101100	54
%	00100101	45
=	00111101	75
"	00100010	42
[	01011011	133
]	01011101	135
\	01011100	134
_	01011111	137
^	01011110	136
{	01111011	173
}	01111101	175
	All other codes*	

6000 SIX-BIT CODE

OCTAL	BIT EQUIVALENT	GRAPHIC
77	111111	!
61	110001	/
73	111011	,
74	111100	%
75	111101	=
76	111110	"
12	001010	[
34	011100	]
37	011111	\
72	111010	←
40	100000	↑
12	001010	[
34	011100	]
77	111111	!

*All other eight-bit codes on tape result in an octal 77 six-bit character, with terminate status MPC Device Data Alert - Code Alert.

8.6.4 BCD Code Conversion

When writing and reading seven track handlers in the BCD mode, code conversion is accomplished as shown in Sections 8.6.4.1 and 8.6.4.2. When working with nine track handlers in the BCD mode, the code conversion requirements is limited to one character as defined in section 8.6.4.4.

8.6.4.1 Seven Track BCD Code Conversion - Write

UNPACKED CHARACTER FROM EUS	CHARACTER WRITTEN ON TAPE
000000	001010
All other 00XXXX characters written as	00XXXX
010000	010000
011010	110000
All other 01XXXX characters written as	11XXXX
100000	101010
101010	100000
All other 10XXXX characters written as	10XXXX
110000	111010
All other 11XXXX characters written as	01XXXX

8.6.4.2 Seven Track BCD Code Conversion - Read

Character Read From Tape	Character to be Packed & Transmitted to EUS
000000	Cannot exist on tape because BCD tape character parity is even and a bit must be present for the self-clocking read operation.
001010	000000
All other 00XXXX characters transmitted as	00XXXX
010000	010000
All other 01XXXX characters transmitted as	11XXXX
100000	101010
101010	100000
All other 10XXXX characters transmitted as	10XXXX
110000	011010
111010	110000
All other 11XXXX characters transmitted as	01XXXX

8.6.4.3 Nine Track BCD Code Conversion - Write

During a write operation, no code conversion is made.

8.6.4.4 Nine Track BCD Code Conversion - Read

When the 6-bit character 001010 is detected, it shall be converted to 000000 before being packed. All other 6-bit characters are packed into 8-bit bytes and transferred to the external user system without code conversion.

9.0 RELIABILITY AND MAINTAINABILITY

9.1 Reliability

The MTC 500 is a specific application of the MPC as a magnetic tape controller. Reliability is not a design criteria for this controller since the components and technology used as outlined in Section 10 are specified per existing standards for component reliability.

A forecast of subsystem reliability will be determined later.

9.2 Maintainability

The Maintainability System for the MTC 500 MPC Tape Controller and magnetic tape handlers will be implemented in accordance with the requirements of the Common MPC Maintainability Specification 43A237500.

This section summarizes the common requirements and covers those details that are peculiar to the magnetic tape subsystem.

9.2.1 Test and Diagnostic Requirements

9.2.1.1 Logic Modules

Common MPC Logic modules (Basic Logic, Control Store, and Main Memory) and the magnetic tape controller adapter will be covered by Isolation Test Routines (ITR's) as defined in the common specification. The ITR system consists of a series of comprehensive test sequences and control routines for which all predictable failure modes have been preanalyzed and cataloged in a hard copy fault dictionary.

9.2.1.1 Logic Modules (continued)

The ITR system utilizes test routines in both "read only" and writable control store that are executed in a predetermined sequence under control of the central system. The ITR system design assumes a failure may exist in any portion of the logic tested in such a fashion that there is no hard core of logic within the MPC that must be operable in order to obtain isolation information.

Full ITR capability will be provided in both the on-line mode (under GECOS) and off-line (under MONITOR II) with some limited capability (basic logic test only) in a completely free-standing mode. In general ITR resolution will be to the integrated circuit level.

9.2.1.2 Device T&D

Electro-mechanical or device oriented electronic failures will be covered by functional T&D programs executed from the central system under control of either OPTS-600 (GECOS) or off-line under control of MONITOR II. In general these T&D programs will be executed with the normal magnetic tape firmware loaded into the MPC; but, if necessary to achieve optimum diagnostic testing, provision for overlaying normal firmware with special microroutines will be provided.

The device producing organization will provide a detailed outline of the test sequences necessary to take full advantage of all device diagnostic commands and features to optimize functional and physical failure diagnosis.

9.2.1.2 Device T&D (Continued)

The detailed outline of test sequences provided by the device producing organization will include associated physical failure references and will be applicable for implementation within the framework of the 600 Line T&D system.

The design goal of device T&D will be to achieve comprehensive physical failure isolation while requiring a minimum of detailed hardware knowledge on the part of the field engineer. When tradeoffs are absolutely necessary, priority will be placed on that portion of the device hardware that cannot be comprehensively tested via local maintenance control with the device in a freestanding mode. At a minimum, device T&D will provide detailed functional diagnosis and operating options sufficient to expedite conventional trouble-shooting of all detected failures.

Device T&D will include the testing of marginal or stress conditions as necessary to detect high probability incipient failure modes or maladjustments, such that successful T&D execution will provide a high confidence level that no maintenance activity of any kind is required on the subject device. Normal execution time for magnetic tape device T&D shall not exceed 6 minutes.

9.2.2 Hardware Requirements

9.2.2.1 Logic Modules

The Logic Design Guidelines for maintainability described in the Common MPC Maintainability Specification 43A237500 will be followed for all common MPC logic modules and device adapters. These logic design guidelines include:

9.2.2.1 Logic Modules (Continued)

1. The use of JK type F/F's for all bistable elements except when RS type, cross-coupled nands, or special storage registers are absolutely necessary because of logic timing or hardware cost (in the case of large register banks such as the MPC RBA registers). The use of cross-coupled nands or special storage register elements should be justified on an individual basis.

2. Full DC initialization capability should be provided on all bistable elements unless technically prohibited, (i.e., no provision on storage register chips). Connector pin limitation is not a valid reason for by-passing this requirement; DC initialization of all bistables (including data registers) is an extremely important maintainability requirement. Note that the requirement is DC initialization. This means no clock pulse required. When clocking is absolutely necessary for initialization, the clock pulse shall result from the initialization signal and be completely separated from the normal data or control clock distribution.

3. All free running counters and asynchronous or configuration variable interfaces will be designed to allow testing in a synchronous mode of operation, i.e., provision for setting free running counters and substituting synchronous clock (such as execution clock) will be provided in a diagnostic mode under program control. Program controlled wraparound will be provided at asynchronous or configuration variable interfaces.

4. The state of all common MPC logic module bistables should be observable in a static mode (no clock) via the MPC maintenance panel. As a goal all controller adapter bistables will be

9.2.2.1 Logic Modules (continued)

4. (continued).
accessible via the IAI interface without intermediate bistable levels. The capability for setting the controller adapter bistables to any desired state should be provided, but not necessarily by a direct load path.

5. The capability of simulating error conditions under program control will be provided to test all error detection logic.

6. Whenever possible, logic partitioning should optimize physical failure resolution. The fewest possible physical components or subassemblies should be placed in any given logic path, i.e., two 2 F/F chips with internal input gating structures are preferable to one 4 F/F chip that requires a separate chip for input gating.

7. Logic implementation plans should be reviewed with the appropriate maintainability function as early as possible in the design cycle.

9.2.3 Software Requirements

Will be supplied later.

9.2.4 Performance Requirements

This section defines the minimum acceptable maintainability system performance characteristics for the initial ship and unlimited production phases of the MPC magnetic tape subsystem.

9.2.4 Performance Requirements (continued)

All characteristics are specified relative to the occurrence of testing of "solid" failures as defined in Section 2.1.5 of the Common Maintainability Specification 43A237500.

In general, a solid failure is defined as any failure mode which exists during the entire execution of an appropriate test sequence and which has a consistent effect on associated functions or logic networks. Logic failure modes covered by the ITR system include:

1. Open output junctions or pins.
2. Open input junctions or pins.
3. Shorted output junctions (always low).
4. Open IC voltage pins.
5. Open connector pins.
6. Shorted input junctions.

	CHARACTERISTIC	DESIGN OBJECTIVE	DEMONSTRABLE PERFORMANCE	
			INITIAL SHIP.	UNLIM.PRODUC.
9.2.4.1	Logic Failures in Common Modules and Device Adapter			
	1. Test Comprehensiveness	99%	90%	95%
	2. Effectiveness	95%	80%	90%
	3. Average Resolution	2 IC's	3 IC's	3 IC's
	4. Worst Case Resolution	5 IC's	10 IC's	6 IC's
	5. Procedure Time (Eff.%)	.4 Hr.	.8 Hrs.	.6 Hrs.
	6. Detailed Knowledge Required	No 95% Yes 5%	No 80% Yes 20%	No 85% Yes 15%
9.2.4.2	Power Supply Failures			
	1. Comprehensiveness	99%	99%	99%
	2. Effectiveness	95%	70%	85%
	3. Replacement Level	Brd/Component	Brd/Module	Brd/Module
	4. Procedure Time (Eff.%)	.6 Hrs.	1.0 Hrs.	1.0 Hrs.
	5. Detailed Knowledge Required	No 95% Yes 5%	No 70% Yes 30%	No 85% Yes 15%
9.2.4.3	Device Failures			
	1. Test Comprehensiveness	99%	90%	95%
	2. Incipient Failure Modes Tested			
	A. Skew	Yes	No	Yes
	B. Capstan Speed	Yes	Yes	Yes
	C. Write/Erase Current	Yes	Yes	Yes
	3. Test Execution Time	.1 Hr.	.2 Hr.	.2Hr.
	4. Avg.Part Replacement	.2 Hr.	.5 Hr.	.3 Hr.

10.0 GENERAL DESIGN REQUIREMENTS

10.1 Packaging Requirements

10.1.1 Controller Packaging

The magnetic tape controller shall be packaged to conform with the requirements of 43A177851, General Design Requirements for GE-655 System. The LA, MPC and TCA shall be packaged as a single control unit such that the cabling between each module is held to the minimum practical.

Wherever possible standard NPL components, or standard PCO components, shall be used in the implementation of this controller.

The assignment of functional units within the controller shall not exceed any of the following:

FUNCTIONAL UNIT	NO. MQX BOARDS
MPC (including control and R/W stores)	8
LA	1
TCA	
Basic Module	3
Data Recovery	5
Switch	
1 x 8	2
2 x 16	4
TOTALS	
Single Channel	19
Dual Channel	30

10.1.1 Controller Packaging (Continued)

The height and depth profiles of the control unit shall not exceed any of the following:

width	55	inches
depth	27½	inches
height	38	inches.

10.1.2 Handler Packaging

The handlers shall not exceed the dimensions of width, 30 inches; depth, 29 inches, height, 61 inches (64 inches to top of operator panel). The control electronics shall be mounted on replaceable PWB's, with the power supplied from a power supply mounted within the handler.

10.2 AC Power Requirements

10.2.1 Controller Power Requirements

AC power requirements for the controller shall conform to General Design Requirements (43A177851), paragraph 4.1.3, titled AC Power for Peripherals.

10.2.2 Handler Power Requirements

A-C power for each handler shall conform to the document referenced in section 10.2.1.

10.3 Cooling Requirements

The anticipated cooling requirements for the controller are as follows:

MTC-501	2900	BTU/hr.
MTC-502	4040	BTU/hr.

MTC-500 CONTROLLER

EPS-1

SIGNATURE SHEET

Prepared by: D A Barney 6/29/71
D.A. Barney Date
Advanced Subsystem Design

Reviewed by: F D Strout 7-7-71
F.D. Strout, Manager Date
Advanced Subsystem Design

T J Beatson 7/9/71
T.J. Beatson, Manager Date
System Software Design

G R Williams [illegible]
G.R. Williams, Manager Date
Systems Design

L I Wilkinson 7/16/71
L.I. Wilkinson, Manager Date
Systems Engineering

R L Mynatt 7/16/71
R.L. Mynatt, Manager Date
APL Design Engineering

R L Ruth 7/16/71
R.L. Ruth, Manager Date
Computer Design Engineering

6/30/71
P.G. Smee, Manager Date
Product Engineering

Ross C. Higbee 7/8/71
R.C. Higbee, Manager Date
Large Systems Marketing Req.

W Bramall/Kiern 7/14/71
W. Bramall, Manager Date
Tape Memories Engineering

Reviewed by: M J Tobias 7/9/71
M.J. Tobias, Manager Date
Advanced System Design

C A Conover 7-9-71
C.A. Conover, Manager Date
Advanced System Engineering

S F Gangi 8/5/71
S.F. Gangi, Manager Date
600 I/O System Software

with objections as noted.
R F Stevens 8/5/71
R.F. Stevens, Manager Date
Operating System Engineering

G. B. Krekeler 7/13/71
G.B. Krekeler, Manager Date
Language & Data Mgt. Sys. Software

7/8/71
P.F. Straka, Manager Date
System Maintainability Engrg.

R F Marshall 7/7/71
R.F. Marshall, Manager Date
Memory & Circuits Engrg.

J D Hann 7/8/71
J.D. Hann, Manager Date
Systems & Software Planning

APPROVED BY:

C W Dix 8/5/71
C.W. Dix, Director Date
Engineering

D K Lundberg /KRG
D.K. Lundberg Date
Associate Director
Product Marketing
July 13 1971