Honeywell

LEVEL 6

HARDWARE

TYPE MTM9102 NINE-CHANNEL NRZI TAPE ADAPTER MANUAL

TYPE MTM9102
NINE-CHANNEL NRZI
TAPE ADAPTER MANUAL

Document No. 71010282-200 Order No. FN18, Rev. 1

This manual has been revised to the -200 level.
It supersedes all previous issues.

RECORD OF REVISIONS

REVISION	DATE	AUTHORITY	AFFECTED PAGES
-100	Nov. 1976	Original issue	-
-200	July 1977	BLCOs 61787, 61873, 70299, and 70395	Entire manual

Hardware Publications, M&TO, Billerica, MA 01821

Printed in the United States of America

0282/FN18

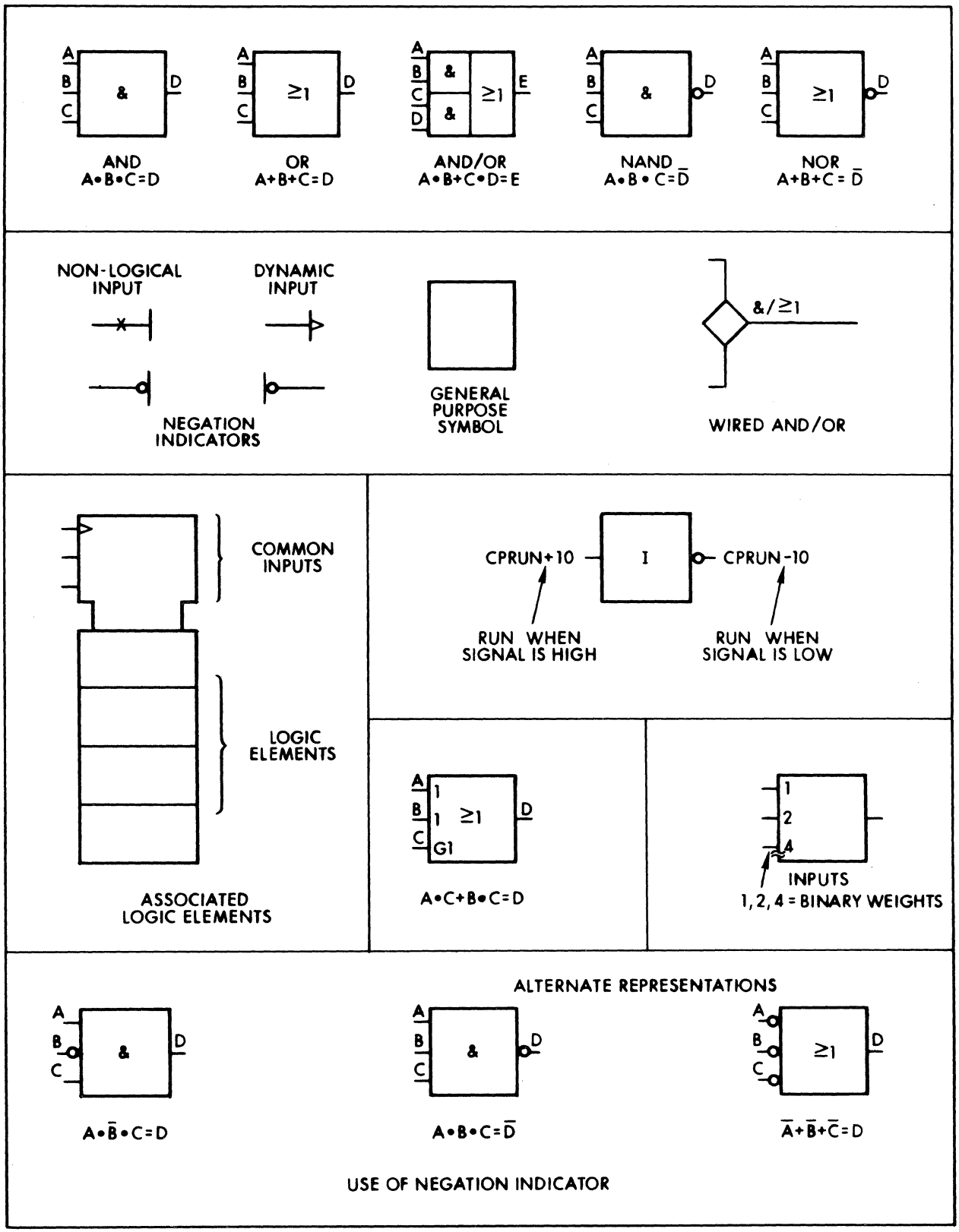
CONTENTS

CONTENTS

ILLUSTRATIONS

TABLES

LOGIC SYMBOLOGY



1 INTRODUCTION

This product manual describes the functionality and operation of the Type MTM9102 Nine-Channel NRZI Tape Adapter, the adapter/device-specific software and the tape medium formats. Since the adapter is designed for use with the Type MTC9101 Magnetic Tape Controller (MTC), this unique firmware for both the adapter and controller is described in this manual. The operational theory within this document acquaints the reader with the functional logic areas of the adapter at both the major and intermediate levels. Refer to the adapter reference manual (listed in subsection 1.7) for detailed logic block diagrams and physical location information.

1.1 GENERAL DESCRIPTION

The MTM9102 Nine-Channel NRZI Tape Adapter (subsequently referred to in this manual as the tape adapter) is a solid-state module (BH2TPA) which, used in conjunction with the magnetic tape controller (BF4MDC), can operate up to four daisy-chained, nine-channel tape devices in Model 34 or 36 configurations of the Series 60 Level 6 computer system. The magnetic tape subsystem, whose attachment configuration is illustrated in Figure 1-1, uses ½-inch magnetic tape for storage and retrieval of NRZI-formatted data. Regardless of the number of drives connected to the tape adapter, only one tape adapter can be utilized with a single controller.

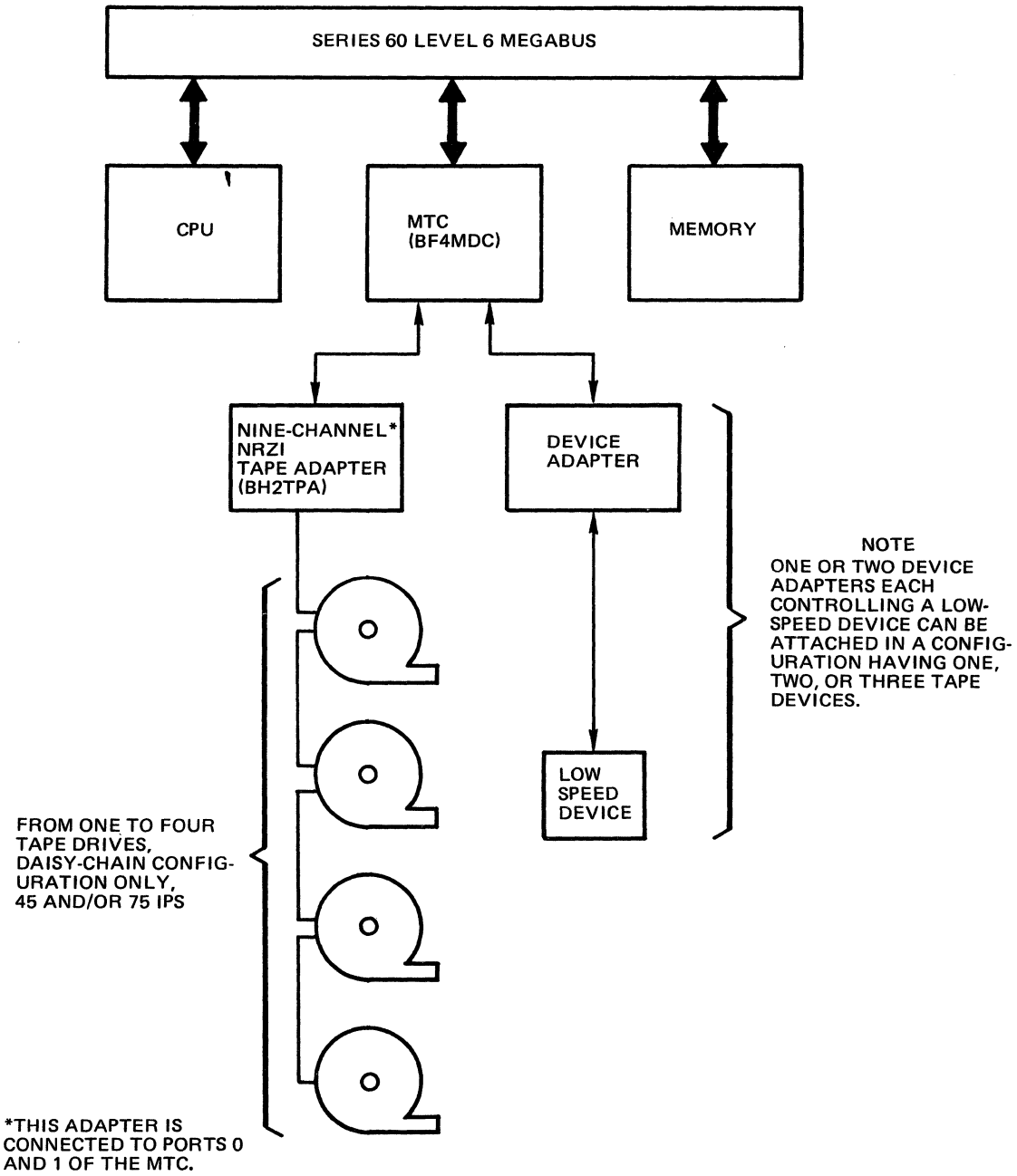


Figure 1-1 Nine-Channel NRZI Tape Adapter Subsystem Block Diagram

The tape adapter consists of dual in-line packages (DIPs) mounted on a double-size Series 60 Level 6 Device-Pac assembled by utilizing printed wiring techniques. The tape adapter is mounted on the MTC package and is plugged into its two 25-pin, in-line connectors. The first rack-mounted nine-channel tape device attaches to the tape adapter via a single cable which has multiple connectors on each end as shown in Figure 1-2. Each subsequent tape device attached to the subsystem uses daisy-chain cabling from the previous device. Individual tape devices derive their power from a power supply located within the device itself.

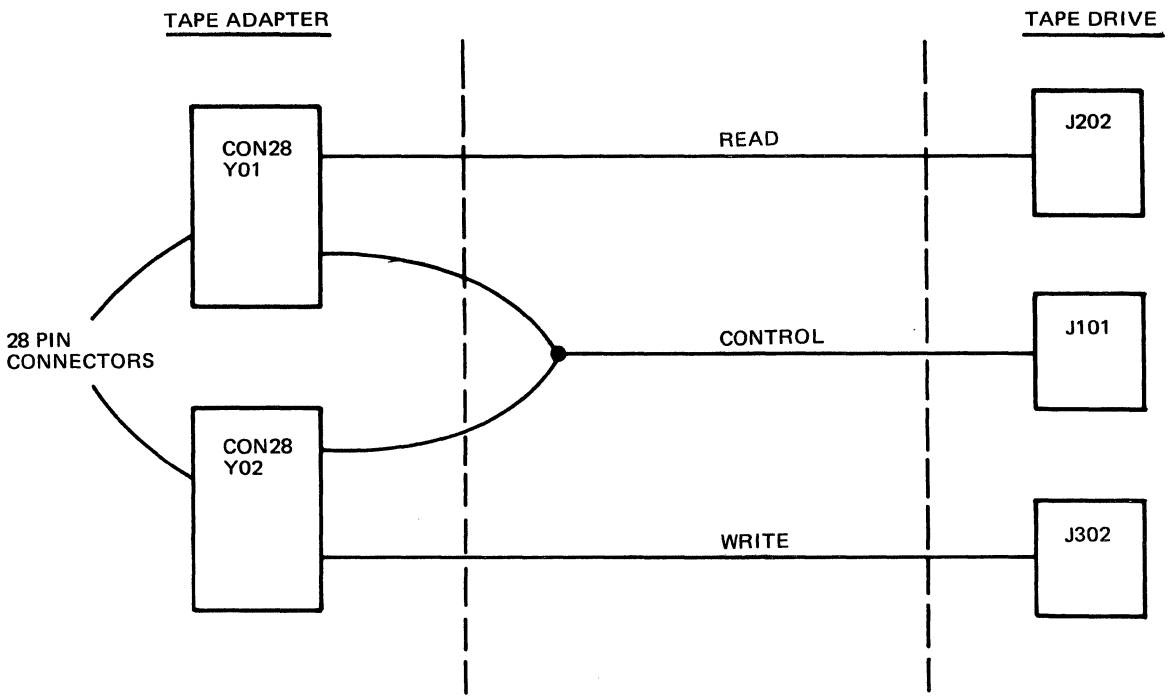


Figure 1-2 Tape Adapter to First Tape Device Cabling

1.2 FUNCTIONAL CHARACTERS

The tape adapter provides the device-specific hardware which, in conjunction with the tape adapter firmware (physically located in the MTC), enables the MTC to control the positioning and the reading/writing of data at various tape speeds on the device in NRZI data format at an 800-bpi density.

1.2.1 Magnetic Tape Controller

The MTC is a microprogrammed peripheral device control unit which, together with the tape adapter, supports from one to four tape devices. Much of the microprocessor portion of the MTC is generalized to facilitate its application as a control element for various unit record and magnetic tape device adapters. The MTC performs the following general purpose control functions:

- Execution of the Series 60 Level 6 Megabus network sequences
- Command decoding
- Status and control register storage
- Data transfer multiplexing to devices/adapters
- Direction of the general flow of command execution.

1.2.2 MTM9102 Nine-Channel Tape Adapter

The tape adapter can support from one to four nine-channel magnetic tape devices. It is attached to these devices via a device level interface (DLI) which allows the intermixing of 45- and 75-ips device speeds. The tape adapter performs the following device-specific functions:

- Controls device interface dialogs
- Generates vertical and longitudinal parity and the CRC character
- Verifies data integrity
- Detects noise and dropped characters from tape
- Detects tape marks
- Allows data to be wrapped from the MTC to the DLI and back to the MTC.

1.2.3 Nine-Channel NRZI Tape Drives

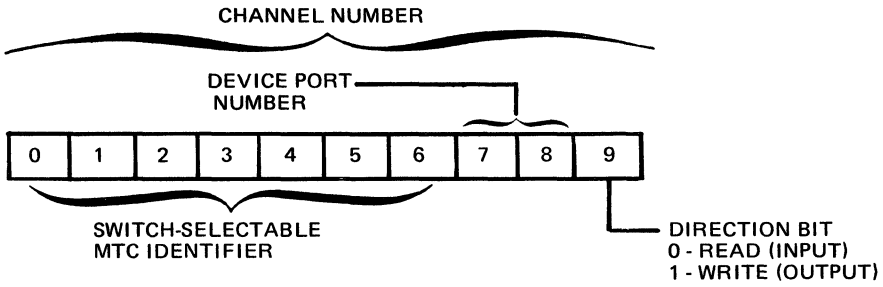
The ½-inch magnetic tape devices are capable of reading or writing nine-channel American National Standard ½-inch magnetic tape at a speed of 45 or 75 ips. (These devices are purchased by Honeywell.)

The tape device performs the following basic functions:

- Receive and generate control signals for device operation
- Position the tape at the beginning of tape marker
- Put the device in an off-line state
- Write data received from the adapter onto tape
- Read data from tape and send it to the adapter
- Perform all digital-analog and analog-digital conversions required for read/write operations.

1.3 SUBSYSTEM OPERATIONAL SUMMARY

Each device attached to the MTC by way of an adapter is addressable by software via channel numbers. A device has two channel numbers assigned, with the numbers differing only in the low-order bit position (called the direction bit). When an I/O Load (IOLD) command for a device is accepted, the direction bit of the channel number specifies whether the command is for an input or output data transfer. The direction bit of a subsequently accepted Output Task Word command is examined to verify that it agrees with the IOLD's direction bit. The following example indicates the composition of a channel number, where bits 0 through 6 are assigned at system installation.



Software usability of the devices attached to the MTC is such that the devices are, in general, independent of one another. For example, operations on one device are independent of the activity of any other device except that the MTC can stall the initiation of a command sequence addressed to one device (channel number) while the MTC is servicing another device. This apparent device independence looks, to software, as if multiple levels of simultaneity exist.

The tape subsystem command sequencing results in any command addressed to a nonbusy channel being accepted, allowing the MTC to accept any command to a tape device while another tape device is executing a data transfer. The accepted command is not invoked until the present data transfer is completed. Only rewind and rewind/unload commands are executed concurrently with data transfer operations. Because channels are serviced on a rotating priority basis, no one channel can dominate adapter usage.

If a tape adapter has fewer than four devices attached, it responds only to channel numbers associated with the installed devices. When a configuration of tape devices is fewer than four, one or two unit record adapters can be implemented by the MTC.

To allow for tape access, the MTC contains a set of software-loadable scratch pad memory locations (refer to Section III of this manual). These locations are assigned to each tape device and contain parameters and control information required for tape operation. In addition to range and address locations, there is a configuration location which contains the mode of operation information, and there is a task word location which stores the command codes.

To perform a specific operation, the software first loads the configuration (and then the address and range) into the scratch pad memory. The task word, which is loaded last, designates the operation to be performed. Upon receipt of the task word, execution of the command is initiated.

Commands addressed to a nonbusy tape device (channel) are always accepted, although their execution can be delayed as previously described. All commands addressed to a busy tape device are rejected (NAK response on the Megabus) with the exception of the Output Control Word command.

1.4 MEDIUM CHARACTERISTICS/FORMATS

1.4.1 Tape Characteristics

Certain physical relationships and data layout dimensions are maintained when nine-channel tape is implemented as the recording medium on the nine-channel tape adapter subsystem. Optimum physical dimensions of the tape and the correlation between tape adapter data bits and tape tracks are shown in Figure 1-3.

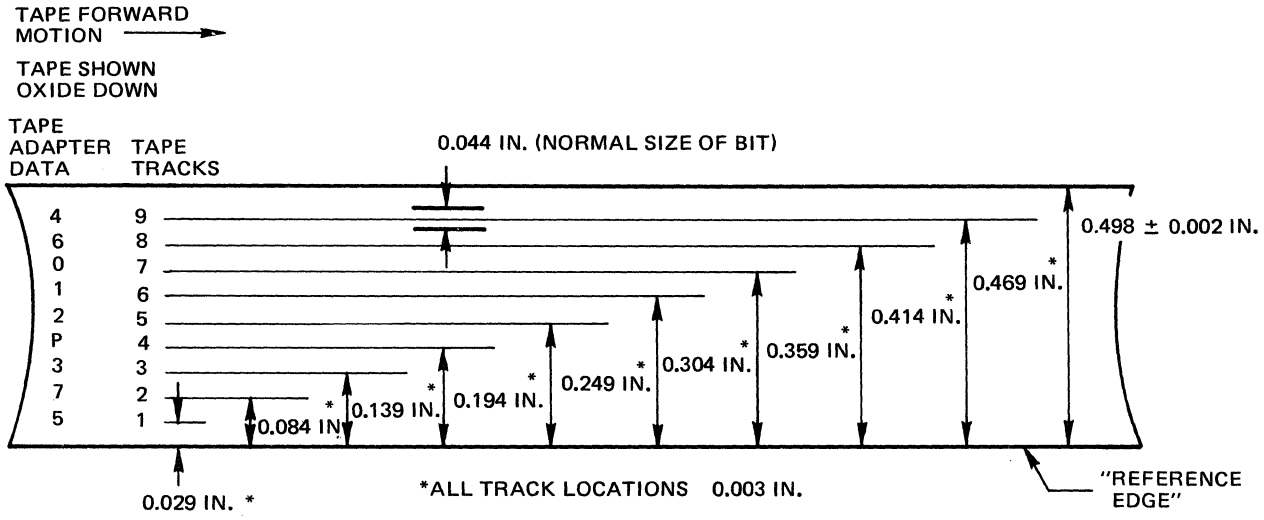


Figure 1-3 Nine-Track Tape Layout

1.4.2 Tape Formats

Beginning and End of Tape

Each tape used on the nine-channel tape adapter subsystem has two markers (see Figure 1-4): a beginning-of-tape (BOT) marker affixed near the reference edge at the start of the tape and an end-of-tape (EOT) marker on the opposite edge at the trailing end of the tape reel. To ensure maximum reliability in the storage of data, an erased area must be maintained at the BOT marker and an unrecorded area is left after the EOT marker.

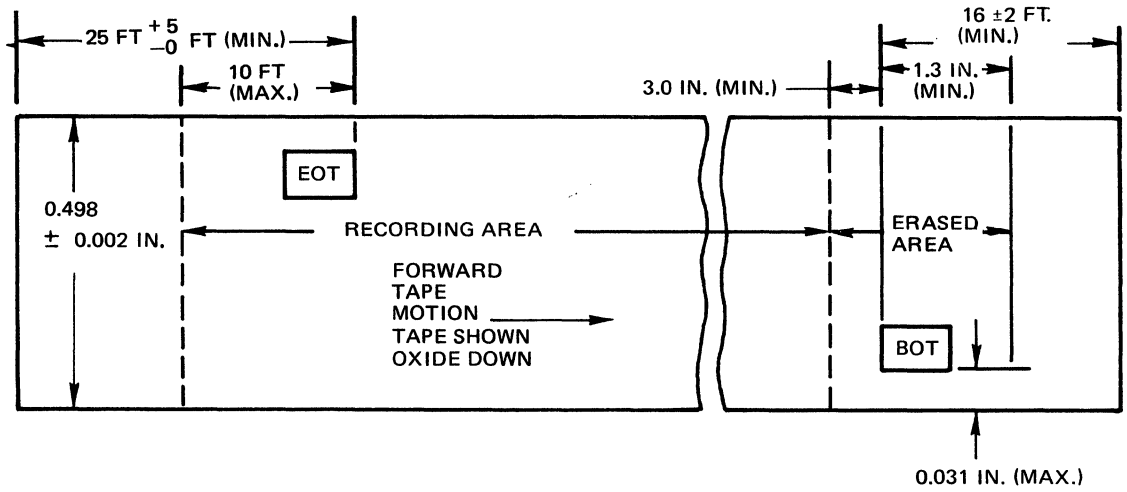


Figure 1-4 BOT and EOT Formats

Non-Return-to-Zero-Inverted (NRZI) Data Recording

In NRZI coding, the logic One bits are reflected on the tape adapter/tape device interface lines as a ground level and the logic Zero bits as a high voltage. However, when recording data on the tape, a logic One is represented by a change of flux during a frame time and a logic Zero as no change of flux (see Figure 1-5) during a frame time.

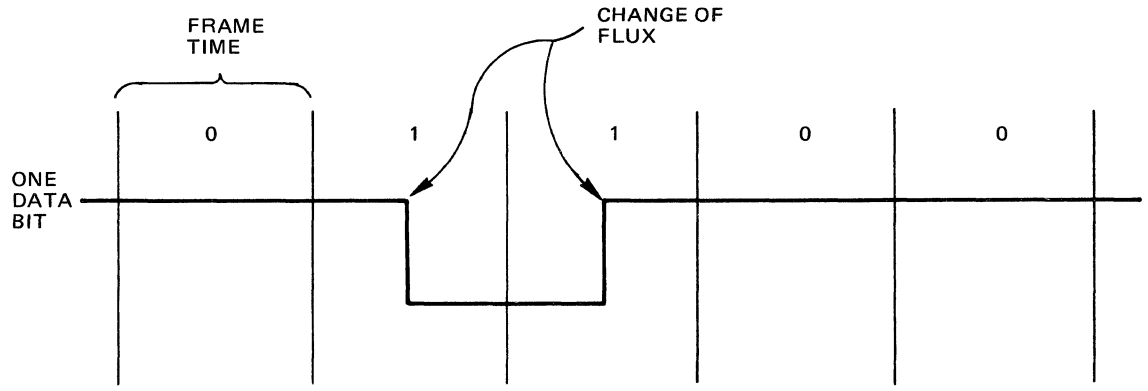


Figure 1-5 NRZI Tape Data Recording

The data from the tape adapter is formatted and recorded on the tape in blocks. Each block consists of a variable data field, a cyclic redundancy check (CRC) character, and a longitudinal redundancy check (LRC) character. The CRC character is positioned

four frame times after the final data byte, and the LRC character must occur four frame times after the CRC. Figure 1-6 shows the relationship between two sequential blocks of data indicating the nominal interblock gap.

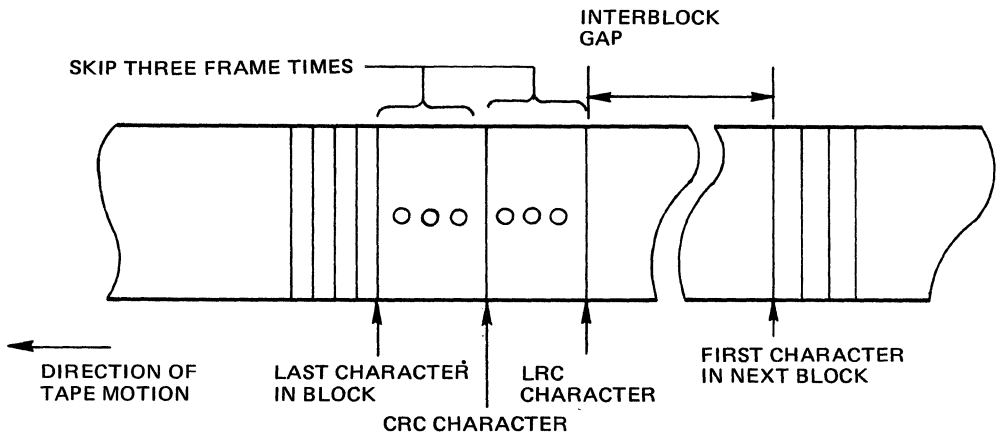


Figure 1-6 Data Block Format

Beginning-of-Tape (BOT) Gap

As shown in Figure 1-4, an erased section of tape must surround the BOT marker. This erased section serves as a defined area in which a read operation can begin. The total erased area is approximately 4.3 inches; it starts a minimum of 1.3 inches prior to the occurrence of the BOT marker and extends for a minimum of 3 inches beyond the marker.

Interblock Gaps

Interblock gaps are areas of tape between data blocks without data, where all tracks are restored to the erase polarity. The nominal length of the gap is 0.6 inch (see Figure 1-6) with a minimum length of 0.5 inch and a maximum length of 25 feet.

Tape Marks

A tape mark character (13 hexadecimal) is distinctive in that it is a single character block followed by seven frame times and then the LRC character. Figure 1-7 shows the relationship between the tape mark character, its check character, and another data block.

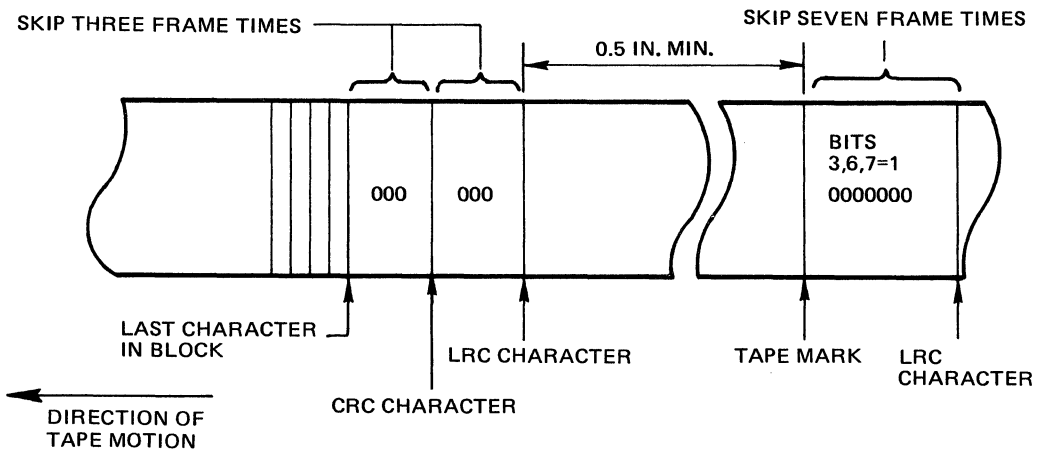


Figure 1-7 Tape Mark Format

Vertical Parity

The nine-channel tape adapter subsystem uses eight tracks for data recording and the remaining track for a parity bit. When performing a write operation, odd parity is generated on the data coming from the MTC. This parity bit accompanies the data to the device and is written in the track shown in Figure 1-3.

When executing a read operation or a read after write (RAW), the parity read from the device is compared against a new parity bit generated on the data byte read from the device. When a no-compare condition arises, the parity error bit is set in the adapter status. However, if an even number of bits in the data byte are dropped or picked up, vertical parity will not detect this condition. Therefore, two other data integrity checks are performed by the tape adapter: CRC and LRC.

Cyclic Redundancy Check (CRC) Character

The CRC character is derived by the adapter hardware and, along with the vertical parity and LRC character, minimizes the possibility of undetected data errors. This character is written on the tape four frame times after the last data byte, as shown in Figure 1-6.

Longitudinal Redundancy Check (LRC)

The LRC character is written on tape following the data portion of each block. It is separated from the end of the data by the CRC character and six skipped frame times as shown in Figure 1-6. The LRC character has one bit per track and is calculated so that an even number of One bits including those of the data, CRC, and LRC are received in each track of a data block. During a read operation an error is detected if the parity is odd in any track. The possibility of not detecting a data error still exists if an even

number of bits is dropped or picked up in a given track. However, when this check is combined with the CRC, the vertical parity check, and the dropped frame check, the probability of not detecting an error is minimal.

1.5 INTERFACES

Figure 1-8 is a simplified block diagram showing the basic interface relationships between the adapter and the MTC and between the adapter and the tape devices. A more detailed description of these adapter interfaces can be obtained from Section II of this manual.

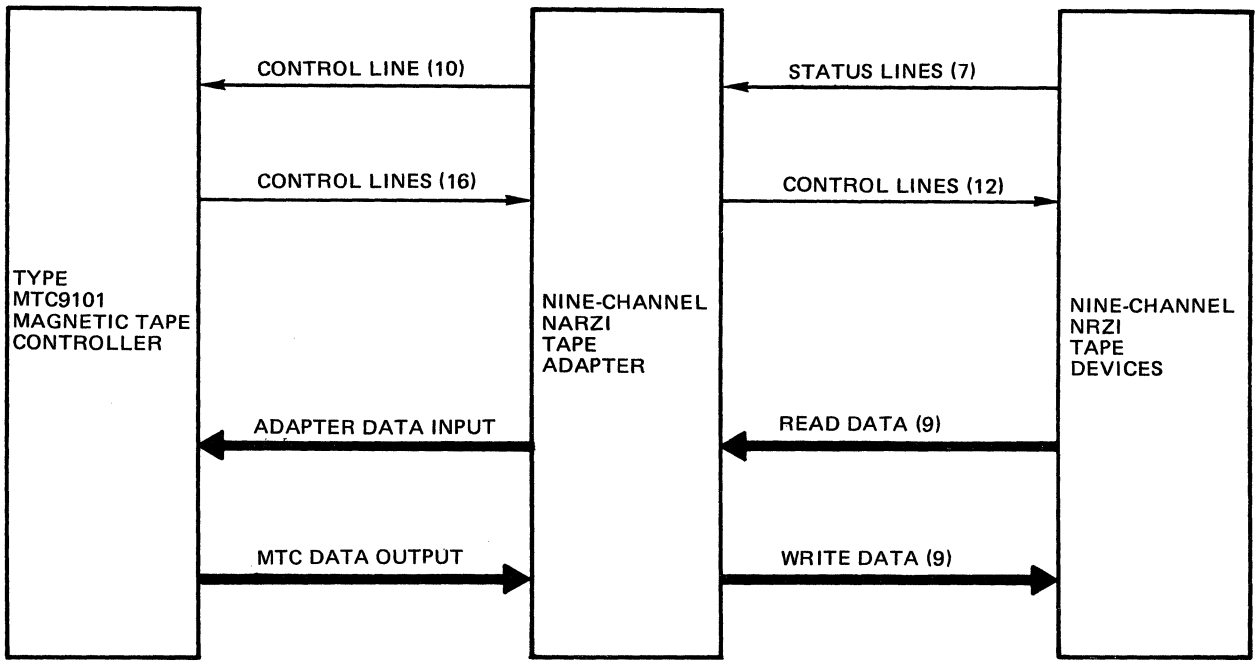


Figure 1-8 MTC/Tape Adapter/Tape Device Interfaces

1.6 OPTIONS

The nine-channel NRZI tape adapter has no options.

1.7 REFERENCE DOCUMENTS

The information contained within the following documents should facilitate an understanding of the tape adapter and the system of which it is a part.

TITLE	DOC. NO.	ORDER NO.
Model 34/36 System Manual	71010200	FL35
Type MTC9101 Magnetic Tape Controller Manual	71010425	FM88
MTM9102 Nine-Channel NRZI Tape Adapter Reference Manual	71010283	FN19
Circuits Description Reference Manual	71010206	FL47
Power Systems Manual	71010290	FL34
Series 60 Level 6 Peripherals Manual	N/A	AT04
Series 60 Level 6 Minicomputer Handbook	N/A	AS22
Type MTC9101 Magnetic Tape Controller Reference Manual (Assy. No. 60130148-003)	71010326	FM89

II
THEORY OF OPERATION

The tape adapter, in conjunction with the MTC and a variable configuration of tape devices (the maximum being four), is a firmware-operated peripheral device adapter. The adapter is attached to the MTC and enables apparent simultaneous control of the connected tape devices. The tape adapter hardware, which implements data transfers and control sequences to and from the tape devices, is divided into six major logic components: Read Path, End-of-Read/RAW, Data Request, Control, Write Path, and Write Clock Logic (see Figure 2-1). These logic components provide for data transfer and efficient control of tape operations.

2.1 SOFTWARE

Operations which affect the tape adapter are the result of two types of commands: software I/O commands that read/write the firmware-accessible locations of scratch pad memory (refer to Section III for the scratch pad memory topology) and those commands contained in the software-supplied task word which cause specific device operations.

2.1.1 I/O Command Set

Table 2-1 lists the input, output, and diagnostic commands applicable to the tape adapter subsystem. These commands load address, range, and control information into a tape device specific segment of the scratch pad memory (SPM). For complete information relative to the command set and the function of the individual commands, refer to the Type MTC9101 Magnetic Tape Controller Manual, Order No. FM88.

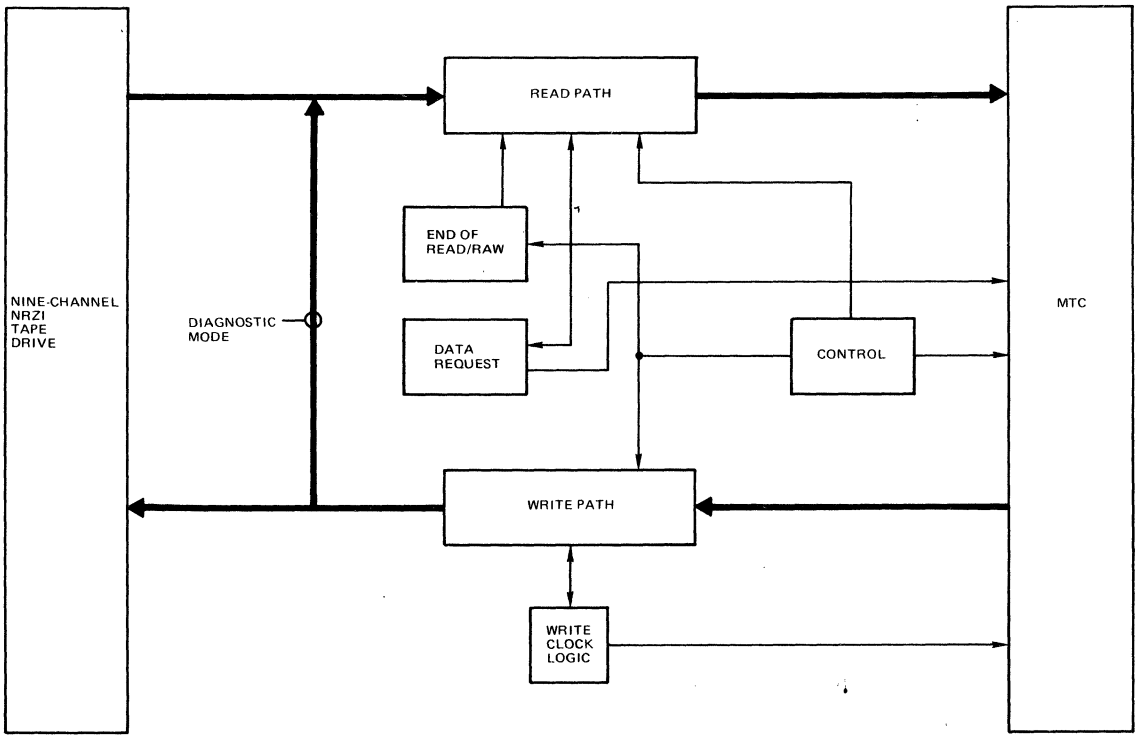


Figure 2-1 Tape Adapter Major Block Diagram

Table 2-1 I/O Command Set

TYPE	FUNCTION CODE	INSTRUCTION
Output	09 (Hex)	IOLD Address (09) Range (0D)
	11	Configuration Word
	03	Interrupt Control
	07	Task Word
	01	Control Word
Input	0C	Range
	10	Configuration Word
	02	Interrupt Control
	26	Identification Code
	06	Task Word
	18	Status Word 1
	1A	Status Word 2
Diagnostic	Any Even Code	Read RWS
	Any Odd Code	Write RWS

2.1.2 Tape Subsystem Device Commands

The device commands are contained in the command code field of the task word. The task word is located in the devices' segment of SPM and is loaded there by the Output Task Word I/O command (see Figure 2-2). The device commands that the tape adapter subsystem is capable of performing are described in the remainder of this subsection.

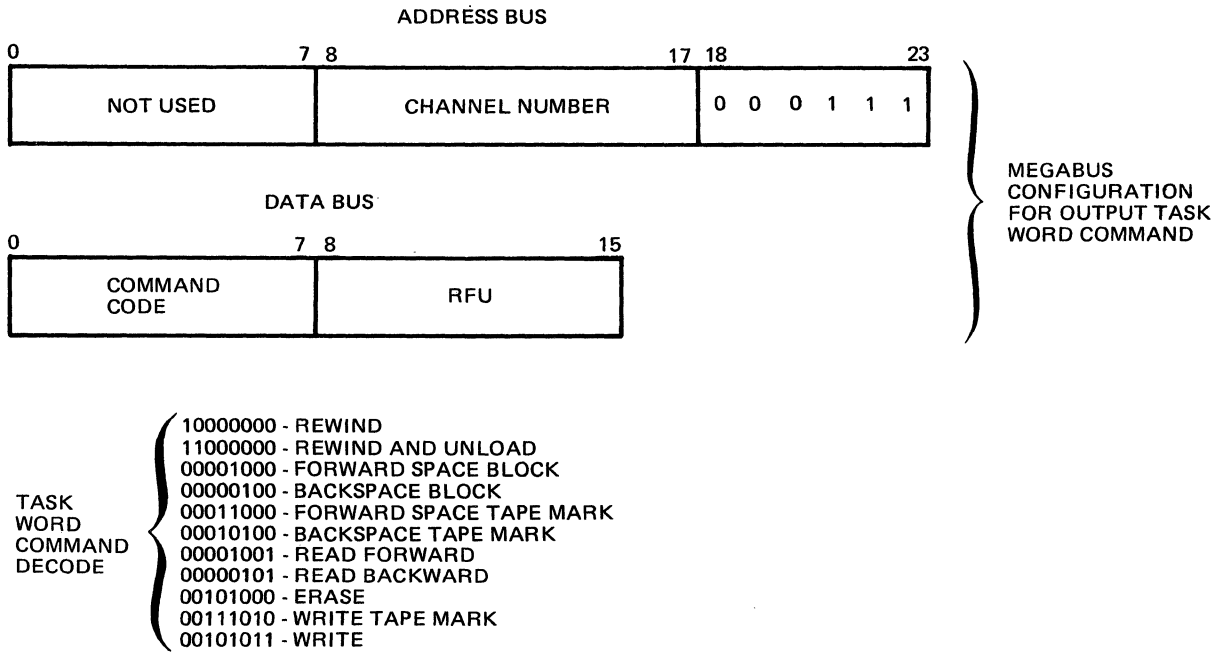


Figure 2-2 Output Task Word I/O Command

Rewind

The Rewind command rewinds the tape to the BOT marker provided that the Rewind command has not been immediately preceded by a write-type command (i.e., Write, Write Tape Mark, Erase). If the previous command is a write, an erase in the forward direction across approximately 1 inch of tape is performed prior to the Rewind operation. The tape device remains in the busy state until the rewind is complete or a firmware-generated timeout occurs. If the tape on the device is at BOT when the command is issued, tape motion is not implemented, and a normal termination of the command results.

The rewinding of a device can be initiated when the device is off-line by depressing the REWIND pushbutton on the device. The firmware polling cycle detects this activity and updates the status of the rewinding, device ready, and attention bits to their present condition (see Figure 3-3). When the manually propagated rewind is

complete, the rewinding status condition is reset, device ready changes state, and attention is set again.

Rewind and Unload

The Rewind and Unload command implements a rewind, as previously described, and following detection of the BOT marker activates the device's unload sequence prior to termination of the command. If the device is at BOT when the command is issued, only the unload sequence is utilized before command termination.

The unload sequence places the selected tape device into the off-line state, extinguishes the ON LINE indicator, and causes tape motion in the reverse direction until tape is wound off the takeup reel.

Forward Space Block

The Forward Space Block command spaces forward over the subsequent block on tape provided this command has not been immediately preceded by a write-type command, (i.e., Write, Write Tape Mark, Erase). If the previous command is a write, an erase in the forward direction across approximately 1 inch of tape is performed prior to the Forward Space Block operation. This command terminates when the tape is in the next interblock gap (refer to subsection 1.4.2) or as the result of a firmware timeout.

Backspace Block

The Backspace Block command spaces backwards over the previous block on tape provided this command has not been immediately preceded by a write-type command (i.e., Write, Write Tape Mark, Erase). If the previous command is a write-type command, an erase in the forward direction across approximately 1 inch of tape is performed prior to the Backspace Block operation. The command terminates when the tape is in the previous interblock gap (refer to subsection 1.4.2) or when the firmware times out.

If the command is issued when the tape is positioned at the BOT, tape motion is not initiated, and the order is immediately terminated.

Forward Space Tape Mark

The Forward Space Tape Mark command spaces forward across one or more blocks until a tape mark or EOT is detected provided this command has not been immediately preceded by a write-type command (i.e., Write, Write Tape Mark, Erase). If the previous command is a write, an erase in the forward direction over approximately 1 inch of tape is performed prior to the Forward Space Tape Mark operation. The command terminates when the tape is in the interblock gap (refer to subsection 1.4.2) following the block containing the tape mark. The command can also be terminated upon EOT detection or a firmware timeout.

Backspace Tape Mark

The Backspace Tape Mark command spaces backwards over one or more blocks until the tape mark is detected provided this command was not immediately preceded by a write-type command (i.e., Write, Write Tape Mark, Erase). If the previous command is a write, an erase in the forward direction across approximately 1 inch of tape is performed prior to the Backspace Tape Mark operation. The command terminates when the tape is in the interblock gap (refer to subsection 1.4.2) preceding the block containing the tape mark, when the tape is at the BOT, or when a firmware timeout is detected.

When this command is issued with the tape positioned at BOT, tape motion is not initiated and firmware timeout termination of the command follows.

Read Forward

The Read Forward command reads forward over the next block on tape provided this command has not been immediately preceded by a write-type command. If the previous command is a write, an erase in the forward direction across approximately 1 inch of tape is performed prior to the read forward operation. This command terminates when the tape is in the next interblock gap (refer to subsection 1.4.2) or as the result of a firmware timeout.

In addition to the transfer of data, vertical parity and the LRC character are read, and integrity checks are performed.

Erase

The Erase command erases tape in a forward direction to produce a 2-inch gap on tape provided this command has been immediately preceded by another write-type command (i.e., Write, Write Tape Mark). If the previous command is not a write, an erase in the forward direction across approximately 2 inches is performed prior to the normal 2-inch erase operation.

Write Tape Mark

The Write Tape Mark command erases tape for 2 inches followed by the recording of a tape mark block (refer to subsection 1.4.2) provided this command is immediately preceded by a write-type command (i.e., Write, Erase). If the previous command is not a write, an additional erase in the formward direction across approximately 2 inches of tape is performed prior to the Write Tape Mark operation. The command terminates normally when the tape is positioned in the gap beyond the new tape mark block.

Write

The Write command records data on tape (in a forward direction) in a data block equal to or greater than the minimum block length allowed by the American National Standard (see bit 4 configuration Figure 3-2). If the command has not beem immediately preceded by a

write-type command (i.e., Write Tape Mark, Erase), an erase in the forward direction across approximately 2 inches is performed prior to the Write operation

In addition to the writing of data, vertical parity and the LRC character are written within the data block for read operation integrity checks.

An attempt to write a data block of less than the minimum length or to write on a drive in the write protect mode results in no data transfer, no tape motion, and an operation check reported in the status.

The command terminates when the tape is positioned in the gap beyond the new data block just written.

2.2 FIRMWARE

The firmware that controls the hardware operations of the tape adapter is contained in the MTC microprogram control store. The main function of the firmware is to interpret external and internal events or conditions pertaining to the tape adapter and to react in a prescribed manner (i.e., setting or resetting of hardware functions). Efficient data transfers are also a result of firmware control of hardware components in the data path.

An overview flow chart of the tape adapter firmware routines, as well as a description of the control data and routine utilization, are provided in Section III of this manual.

2.3 HARDWARE OVERVIEW DESCRIPTION

As shown in Figure 2-1 the tape adapter is organized into six fundamental logic areas. Figure 2-3 is an intermediate block diagram of the tape adapter which depicts the components in each of these areas and the interconnections with the MTC and the tape devices. Although the primary function of the tape adapter is to control the data flow between the MTC and the device, the tape adapter performs the following secondary functions:

- Assisting the MTC in formatting data written on the tape device
- Developing signals for device operation
- Generating operational error signals for MTC interrogation
- Generating data and nondata service requests and sending them to the MTC
- Generating status information and sending it to the MTC.

See Figure 2-4 for a listing of the status developed and Figure 3-3 for detailed MTC/tape adapter status relationship. The descriptions in subsections 2.3.1 through 2.3.7 pertain to the logic blocks shown in Figure 2-3.

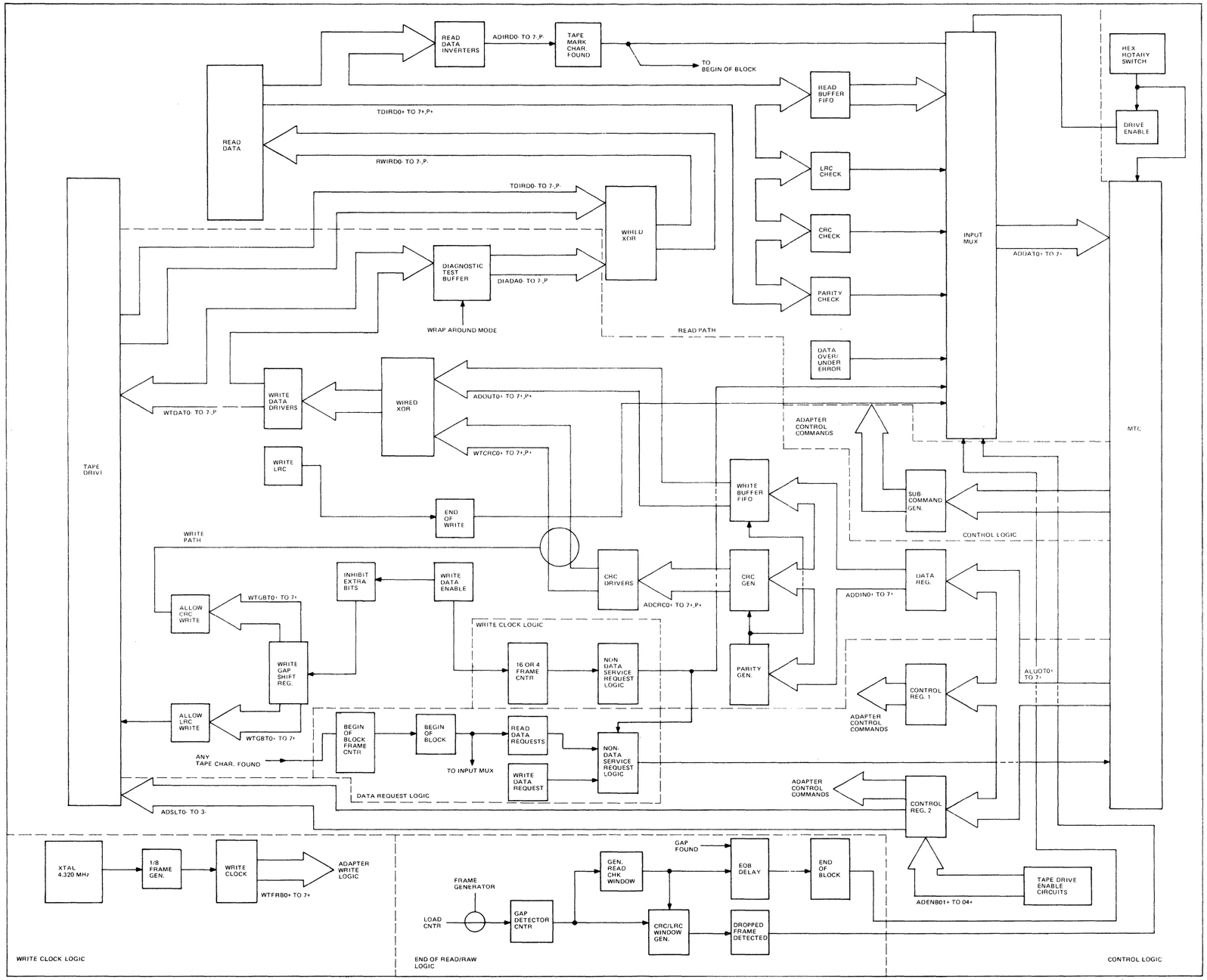
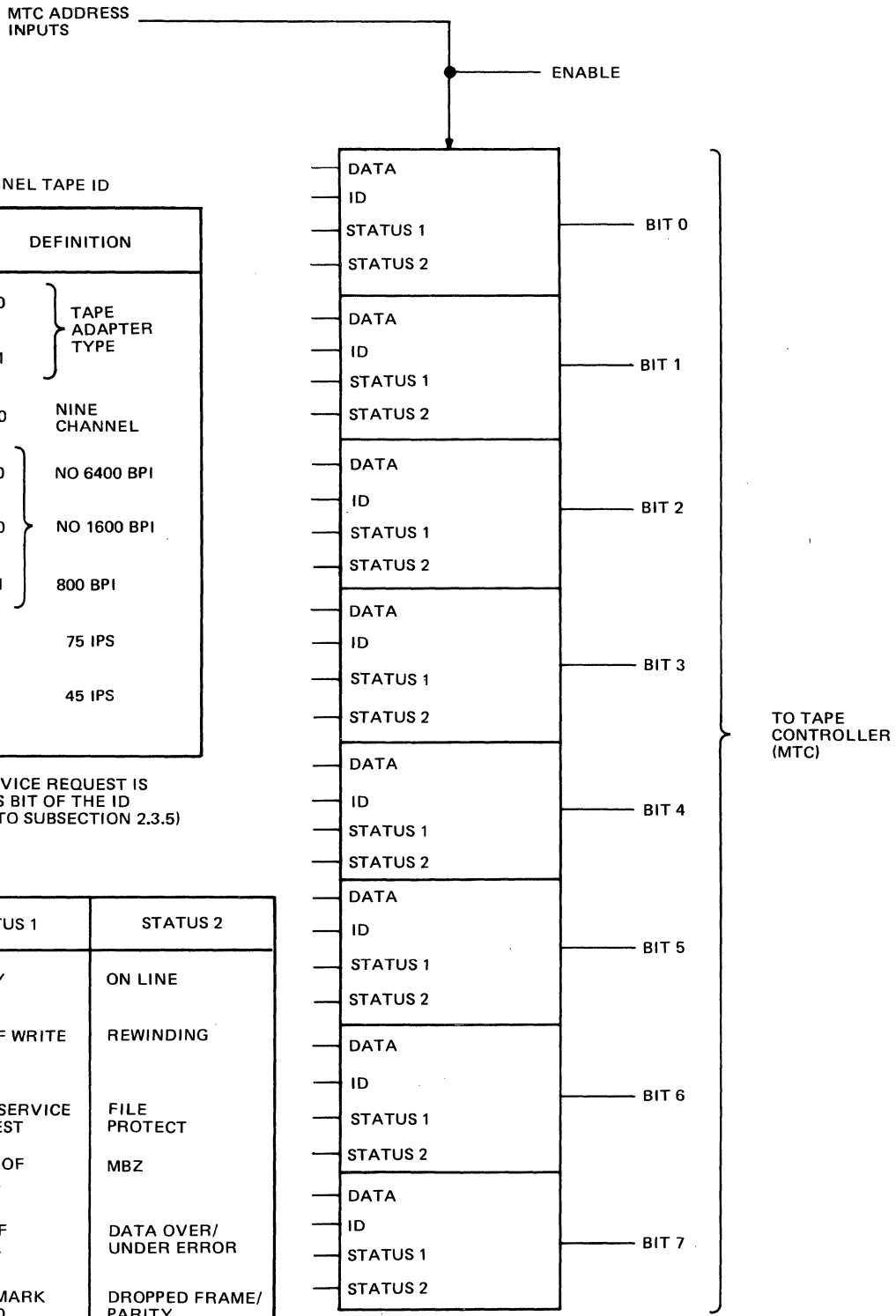


Figure 2-3 Tape Adapter Intermediate Block Diagram



NINE-CHANNEL TAPE ID

BIT		DEFINITION
*0	0	TAPE ADAPTER TYPE
1	1	
2	0	NINE CHANNEL
3	0	NO 6400 BPI
4	0	NO 1600 BPI
5	1	800 BPI
6		75 IPS
7		45 IPS

*NONDATA SERVICE REQUEST IS WIRED TO THIS BIT OF THE ID BYTE (REFER TO SUBSECTION 2.3.5)

BIT	STATUS 1	STATUS 2
0	READY	ON LINE
1	END OF WRITE	REWINDING
2	DATA SERVICE REQUEST	FILE PROTECT
3	BEGIN OF BLOCK	MBZ
4	END OF BLOCK	DATA OVER/ UNDER ERROR
5	TAPE MARK FOUND	DROPPED FRAME/ PARITY
6	BOT	CRC
7	EOT	LRC

Figure 2-4 Input Multiplexer Data Structure

2.3.1 Interface Description

The tape adapter provides the interfaces necessary to transfer data to/from a tape device and to/from the controller. Figure 2-5 is a simplified block diagram showing the basic interface relationships between the tape adapter and the MTC and between the tape adapter and the tape devices.

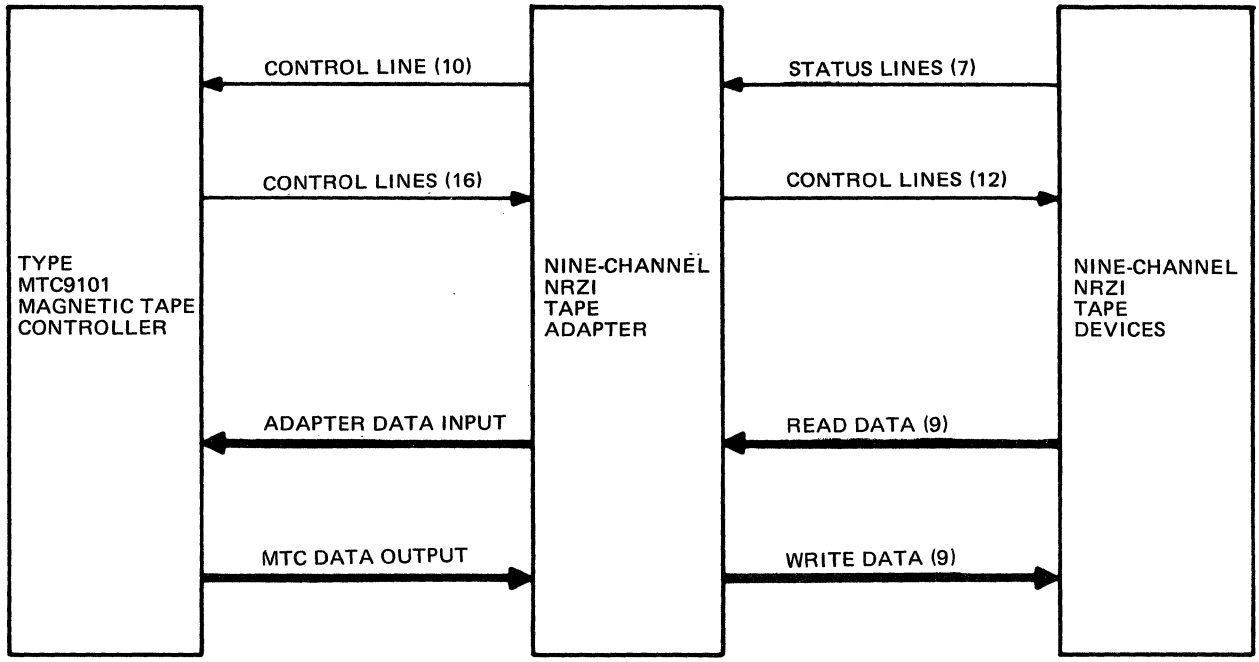


Figure 2-5 MTC/Tape Adapter/Tape Device Interfaces

Adapter/MTC Interface

The interface signal lines between the adapter and the MTC are shown in Figure 2-6. A description of the application of these lines is provided in the Magnetic Tape Controller Manual, Order No. FM88. Signal line mnemonics are shown in Figure 2-6 as they are designated for the adapter. It must be kept in mind that certain lines within the MTC may have different mnemonics.

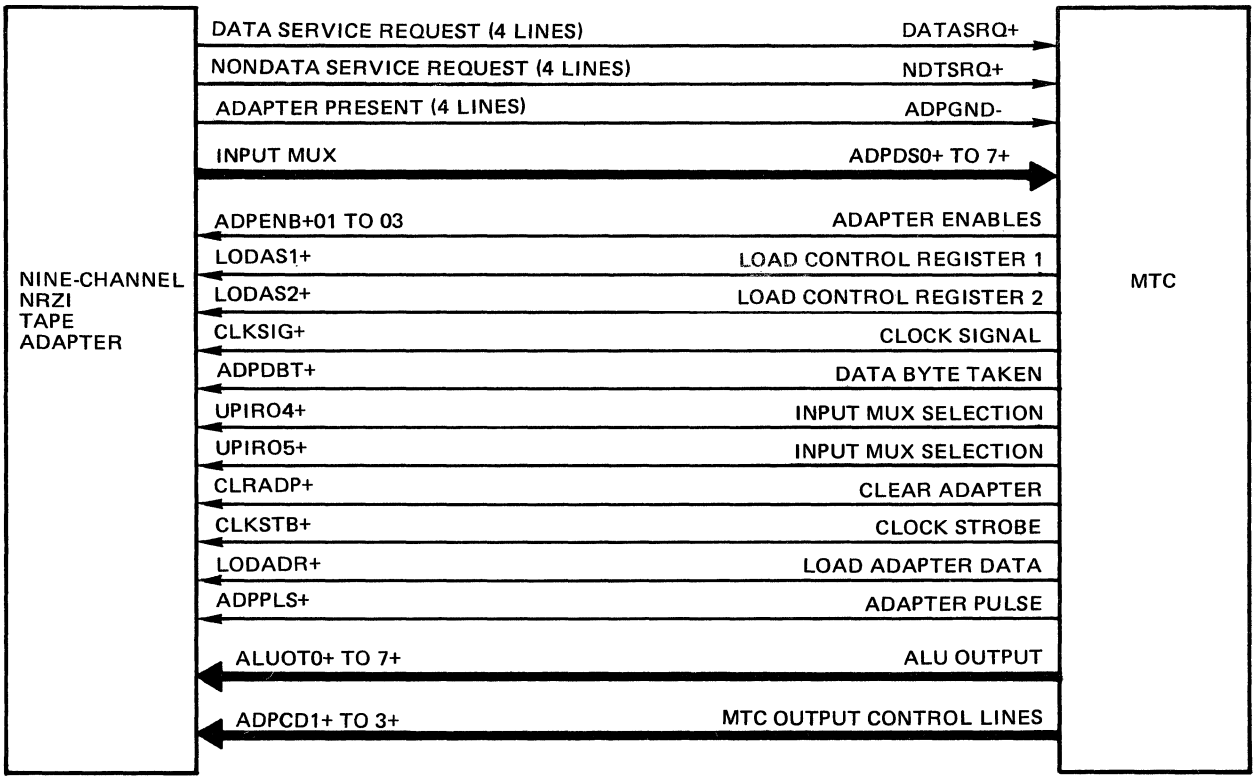


Figure 2-6 Adapter and Controller Interface

Tape Adapter/Device Interface

A diagram of the tape adapter/device interface interconnections is shown in Figure 2-7. This figure identifies the interface lines, their direction, mnemonics, and applications. It depicts the lines for the tape adapter to the device using tape adapter mnemonics. For a detailed description concerning the usage of all the signal lines, refer to Table 2-2.

This interface, which links all four tape devices, allows the adapter hardware and firmware to select a device and perform a designated operation. It also provides the paths for supplying the devices with data, control, and timing pulses, and for supplying the adapter with timing, data, and status from the device.

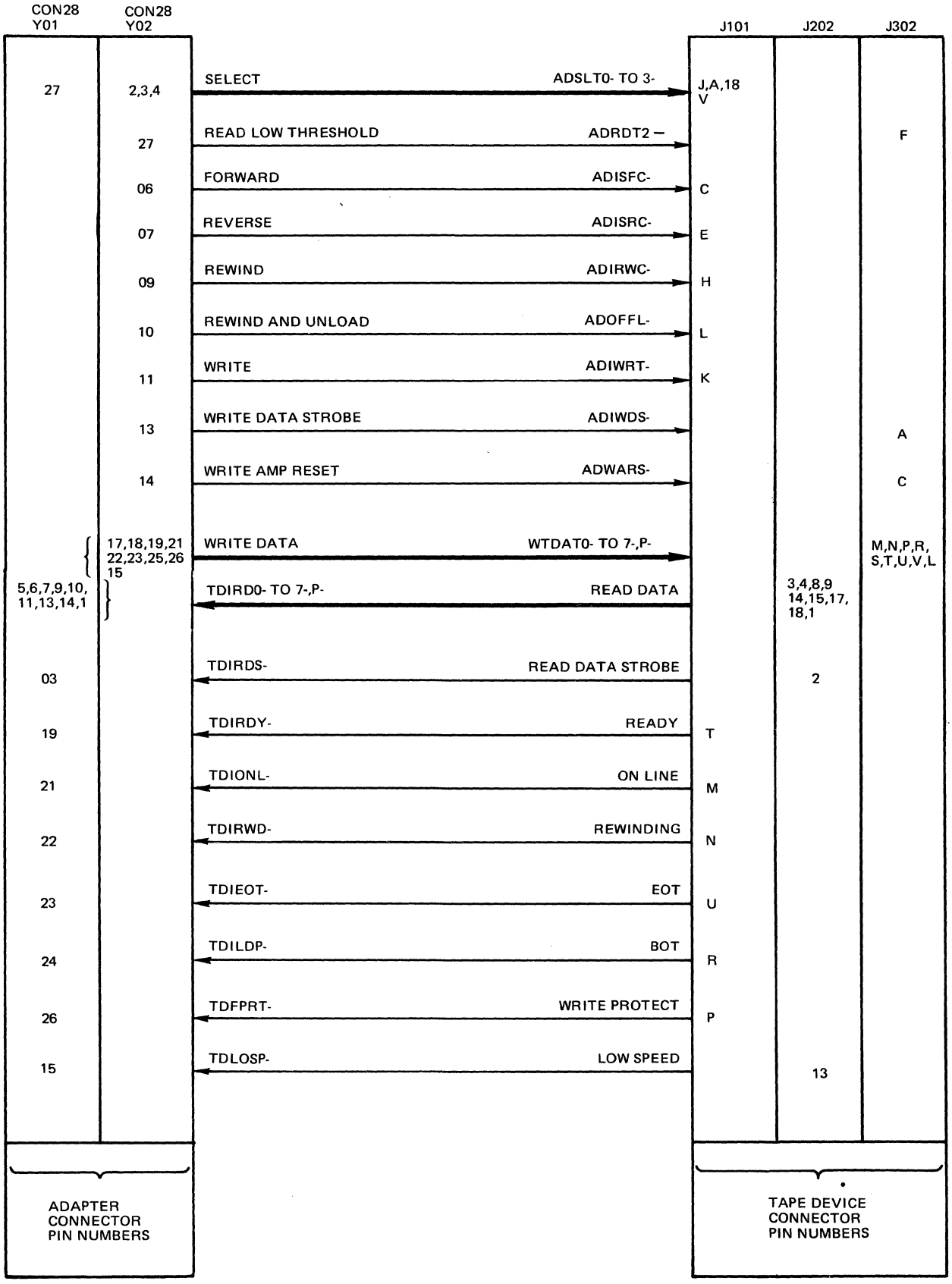


Figure 2-7 Adapter and Tape Device Interface

Table 2-2 Tape Adapter/Device Interface Signal Lines
(Sheet 1 of 3)

OUTPUT LINES TERM AND MNEMONIC	DESCRIPTION
Unit Select (ADSLT0→3)	Four lines used for selection in a magnetic tape configuration. These lines are switch selectable within the device, enabling individual selection by any one of the four lines. All other interface output signals are gated with the unit select lines.
Forward (ADISFC)	This signal line causes the device, when ready, to initiate forward motion or when reset to cease forward motion. Due to acceleration and deceleration times, a delay is required after the change of state of this line.
Reverse (ADISRC)	This signal line causes the device, when ready, to initiate motion in a reverse direction or when reset to cease reverse motion. Due to acceleration and deceleration times, a delay is required after the change of state of this line.
Rewind (ADIRWC)	This pulse, whose minimum width is 2 μs, causes the device, when ready, to move tape in a reverse direction at three/four times the normal forward speed. Upon sensing BOT, the device overshoots the BOT marker and stops, reverses direction, and comes to rest at the marker. The Rewind indicator is illuminated for the duration of the rewind operation.
Rewind and Unload (ADOFFL)	This pulse, whose minimum width is 2 μs, resets the device ON LINE indicator and the interface On-Line signal line to the adapter. The device then executes its rewind (see above) and unload sequences to remove tape from the take-up reel.
Write (ADIWRT)	To perform a write operation, this line is required to be true for a minimum of 20 μs after forward motion is initiated. If a read operation is desired, this line must be false for a minimum of 20 μs after the initiation of forward or reverse motion.

Table 2-2 Tape Adapter/Device Interface Signal Lines
(Sheet 2 of 3)

OUTPUT LINES TERM AND MNEMONIC	DESCRIPTION
Read Low Threshold (ADRDT2)	This line selects a low level for the read circuits in the device. This line is used when it is required to recover very low amplitude data and must remain steady for the duration of the block.
Write Data Strobe (ADIWDS)	This is a pulse whose minimum width is 2 µs. It is sent with each character to be recorded on the media. The frequency of this pulse is equal to the transfer rate and the data must remain steady 0.5 µs before and after the pulse.
Write Amp Reset (ADWARS)	This is a 1-µs pulse (minimum) which resets the device write register, causing the LRC character to be written. This pulse is generated at the end of the data block and occurs eight frame times after the last data character (refer to subsection 2.2.2.8).
Write Data (WTDAT0→7,P)	These lines, when true at write data strobe time with the device in write, result in a flux reversal on the corresponding tape track (see Figures 2-4 and 2-6). These lines must be held steady 0.5 µs before and after write data strobe.
INPUT LINES TERM AND MNEMONIC	DESCRIPTION
Ready (TDIRDY)	This line is valid when the device is prepared to accept an adapter command. This line indicates that the tape is under tension, a rewind is not being performed, and the device is in the on-line state.
On-Line (TDIONL)	This line indicates, when true, that the device is under control from the adapter. When false the line indicates to the adapter that the device is under manual control.
Rewinding (TDIRWD)	This line is valid while the device is performing a rewind operation.

Table 2-2 Tape Adapter/Device Interface Signal Lines
(Sheet 3 of 3)

INPUT LINES TERM AND MNEMONIC	DESCRIPTION
EOT (TDIEOT)	This line being true indicates the EOT reflective marker is positioned under or has passed beneath the photo-sensor during a forward tape motion command. The line is reset after the marker passes the sensor in a reverse tape motion direction.
BOT (TDILDP)	When valid this line indicates the device is ready and the tape is positioned with the BOT marker under the photo-sensor. When the BOT marker leaves the photo-sensor area, this line changes state.
Write Protect (TDFPRT)	This line is valid when tape is loaded and the supply reel has the write enable ring removed.
Low Speed (TDLOSP)	This line is set when the device has been conditioned, via device mechanical hardware, to move tape at the lower of the two available speeds (45 or 75 ips).
Read Data Strobe (RWIRDS)	This line supplies the adapter with a 1 µs pulse for each character read from tape. The read data lines must remain steady for 0.5 µs before and after the read data strobe.
Read Data (RWIRD0→7,P)	These lines each represent one bit of the character recorded on the tape tracks (see Figure 2-4). The state of each of the nine lines must remain steady 0.5 µs before and after the read data strobe.

2.3.2 Read Path Functional Components

The data from the device is received in parallel form and is ORed with the data from the diagnostic test buffer. In normal operation the output of the read data reflects the data from the device, whereas in the diagnostic mode the read data logic accepts the output of the diagnostic test buffer. The read data is fed to a set of read data inverters whose outputs in conjunction with the read data are used in the tape mark character found logic to decode a tape mark from the device. This decoded output is sent to the data request logic and the MTC as a status function.

The read buffer is a 32-byte fall-through FIFO (first-in first-out) buffer which receives its input from the read data logic. The first byte of the FIFO acts as an independent input register, and the last byte acts as an independent output register whose contents are sent back to the MTC.

Four other logic areas of the read path receive the data from the read data: LRC check, CRC check, parity check, and tape mark detection. Each area verifies data integrity in a different manner, and when an error is detected, the appropriate error signal is raised and sent back to the MTC via an adapter status input.

Another read path component is the data over/under error logic. When this logic detects a rate error, an indicator signal is sent to the MTC by way of the status register.

All information (data, status, and ID) sent to the MTC utilizes a four-input multiplexer (see Figure 2-4). The outputs (ADPDS0 through ADPDS7) of the input multiplexer are firmware-controlled and reflect the input selected by the two interface select lines.

2.3.3 Write Path

Data received from the MTC is in parallel form with no parity and is stored in the data register. The output of the data register is simultaneously present at the write buffer (FIFO), the CRC generator, and the parity generator. Although the data is available at the write buffer, it is not stored until a parity bit is sent from the parity generator allowing for the 9-bit transfer to the device. The parity bit is also sent to the CRC generator to allow for computation of a nine-bit CRC character.

The CRC generator's outputs are applied to the CRC drivers, the outputs of which are ORed within the write buffer outputs. During the data transfer (Write Data Enable), the output of the CRC drivers is inhibited, and the write data drivers reflect the write data. The CRC character is recognized when the write shift register and its associated logic (Inhibit Extra Bits) generate the Allow CRC Write at the end of the data transfer. After the CRC character is written, the Allow LRC Write causes the LRC character to be written and the End of Write to be set.

The write data drivers send data to the devices or to the diagnostic test buffer. When executing a normal write command, the data goes to the devices. If the command is a wraparound write, the diagnostic test buffer is enabled, and the data is returned to the MTC via the read path (refer to subsection 2.3.2).

2.3.4 Write Clock Logic

The write clock logic contains a free-running oscillator (4.320 MHz). The oscillator outputs are the basis for producing all the timing pulses required for operation. The 1/8 frame generator, using the oscillator as a source, counts 1/8 segments of a frame regardless of the tape speed. Implementing a strobe from the 1/8 frame generator, the write clock generates eight sequential pulses that are distributed throughout the tape adapter.

The 16-frame/4-frame counter feeding the nondata service request logic enables the adapter to issue a nondata request to the MTC at 0.005-inch intervals of tape duirng the data transfer portion of a unit operation or 0.02-inch intervals of tape at other times.

2.3.5 Data Request Logic

The data request logic requests data from the MTC for a write operation, requests to send data to the MTC for a read operation, or indicates the presence of a nondata service request to the MTC.

The read operation implements the beginning-of-block frame counter to discriminate against noise records encountered while seeking a valid record. Upon detection of a valid record, the beginning of block sets, allowing read data request to generate data requests each time the read buffer (FIFO) has information available.

The write data request causes a data request during the write operation any time there is available storage in the write buffer (FIFO).

Each time a nondata service request is generated in the write clock logic, a data request is issued to the MTC. The MTC responds as if this were a normal data request (high priority) and determines that it is actually a nondata request by reading the ID byte through the input multiplexer (see Figure 2-4) and examining bit 0. This procedure of channeling the nondata requests through the data request logic is required due to the timing considerations encountered in the firmware processing of nondata requests. Because all tape data requests appear on channel 0, it makes tape nondata requests higher priorities than other data requests.

2.3.6 End of Read/Read After Write (RAW) Logic

Two-frame time lapses between read strobes are detected by the gap detector counter. This counter sets and increments the CRC/LRC window generator, which inhibits the setting of the dropped frame detected logic during the time intervals of from three to five and from seven to nine frames after the last read strobe. The time

period of three through five allows the CRC character to be read without producing a dropped frame detected error. The time period of seven through nine allows the LRC character to be read without producing an error. When a read strobe is encountered at any time other than during the CRC/LRC windows, the Dropped Frame Detected signal sets and can be read by the MTC via status information.

The gap detector counter is also used in conjunction with the generate read check window to initiate the EOB delay. The EOB delay counts 32 frame times before the end-of-block indicator is set. Firmware then resets the end-of-block indicator, which has been sent to the MDC via a status register, and allows the EOB delay to count another 32 frame times and set end of block again. If a read strobe is detected during the second count of 32, end of block does not set. This condition is reported to software as an uncorrectable read error.

2.3.7 Control Logic

The tape adapter control logic uses two control registers to store the contents of the MTC data and adapter enable lines during a firmware load control register command. This stored information is distributed throughout the adapter and is sent to the device to establish the type of the mode of operation.

The subcommand generator stores the interface control signals from the MTC. The subcommand generator outputs are "store" and "clear" functions resulting from firmware manipulation of the MTC/ tape adapter interface lines.

The hex rotary switch inhibits the data request, nondata request, and adapter present logic from other adapter positions when a system with more than two devices is supported. This inhibiting allows the request logic of other low-speed device adapters to function if only one, two, or three tape devices are attached. The drive enable circuitry controls the enabling of the input multiplexer for the tape adapter. When there are more than three devices on the tape adapter, the input multiplexer is always enabled. If three or fewer tape devices are supported, the input multiplexer is enabled only when a tape device is addressed.

2.4 INTERMEDIATE HARDWARE DESCRIPTION

Figure 2-3 is an intermediate level block diagram of the tape adapter logic. This illustration is designed to guide the user through the detailed logic block diagrams contained in the Tape Adapter Reference Manual (Order No. FN19), and is the overview diagram applicable to the following text.

2.4.1 Write Operations

The subsequent subsections describe the adapter hardware implemented when a write operation is to be performed on a selected tape device. The data transfer path, as well as the control, timing, and data integrity logic required to execute the write operation,

are described. There are only three functional types of write operation: write data, write tape mark, and erase. The hardware application is basically the same for all three operations. Firmware determines how the transfers will differ. Wherever possible variations in the hardware utilization are noted.

2.4.1.1 Write Control Logic

Since the tape adapter can control up to four devices (each of which is handled as a channel), the logic is designed to allow other adapters to function as if the tape adapter has two or fewer devices. Figure 2-8 shows the hex rotary switch and the associated logic used to allow or inhibit other adapter positions and data request, nondata request, and adapter present functions.

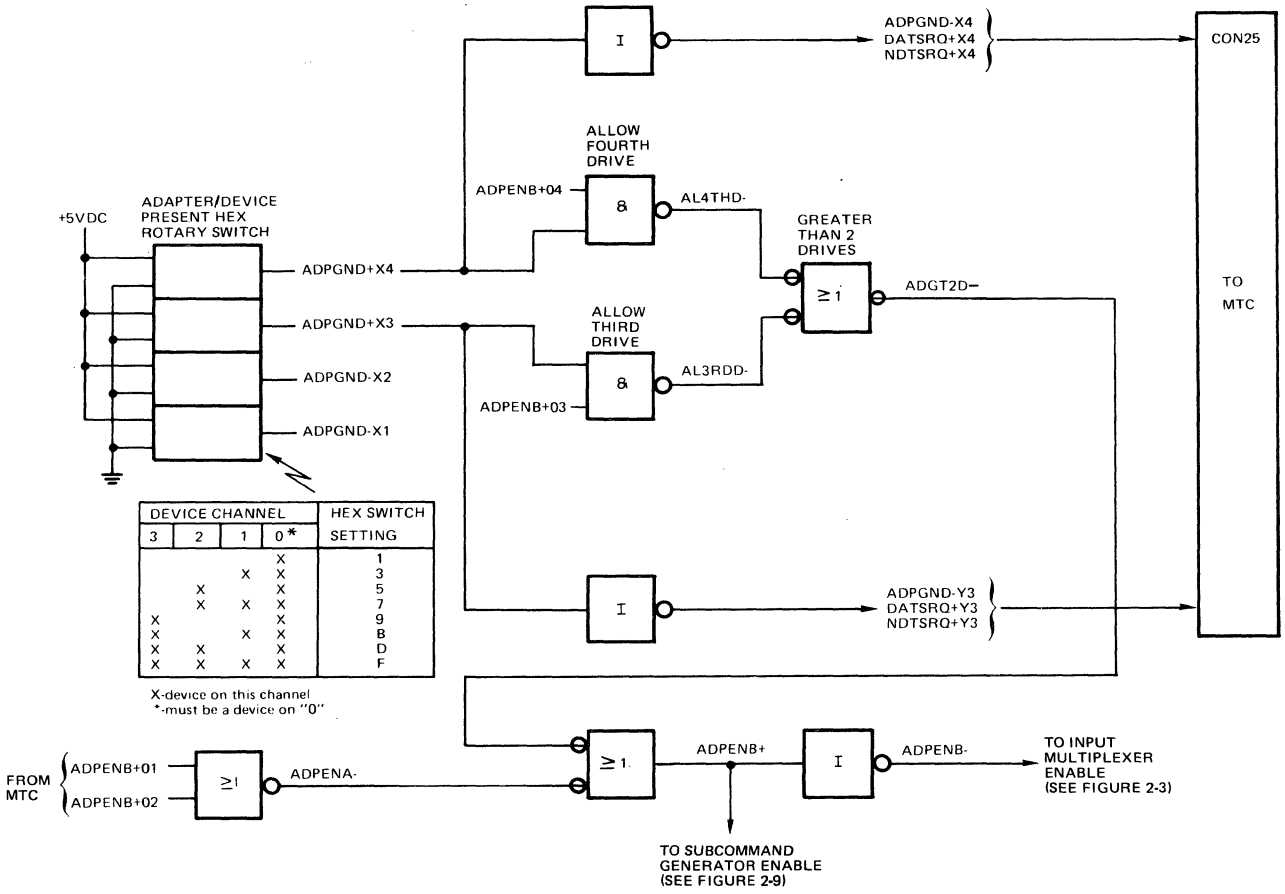


Figure 2-8 Adapter Selection Logic

When the rotary switch is put in positions which select either ADPGND+X3 or ADPGND+X4 (5, 7, 9, B, D, or F), a positive voltage is applied to the corresponding output. This positive voltage is inverted and the ground level is applied to the data request, nondata request, and adapter present functions (ADPGND, DATSRQ, and NDTSRQ) on the MTC interface for the appropriate adapter position. The grounding of these functions inhibits any erroneous requests when channel 2 or 3 is occupied by a tape device.

The positive voltage output of the switch (ADPGND+X3/ADPGND+X4), in conjunction with the applicable adapter enable function (ADENB-03/ADENB+04), allows the output of the greater-than-two device logic (ADGT2D-) to go to ground. As a result of this, the input multiplexer and the subcommand generator are enabled (ADPENB±) for a command sent to an adapter. If ADGT2D- is high, the input multiplexer and the subcommand generator only react as a result of commands sent to device 1 or 2 (ADPENB+01/ADPENB+02).

The tape adapter receives firmware-generated load register commands, reset pulses, and timing pulses through the MTC's use of individual interface lines. As these lines must remain steady for a full firmware cycle, they are latched in the subcommand generator (see Figure 2-9). The subcommand generator latches the interface control lines at the leading edge of Clock Strobe (CLKSTB+) from the MTC when the adapter is enabled as reflected by ADPENB+ from the selection logic (see Figure 2-8). At the leading edge of the following clock strobe (CLKSTB+), the new condition of the interface control lines is latched or all the outputs are reset to Zero if the adapter is no longer selected (ADPENB+ low).

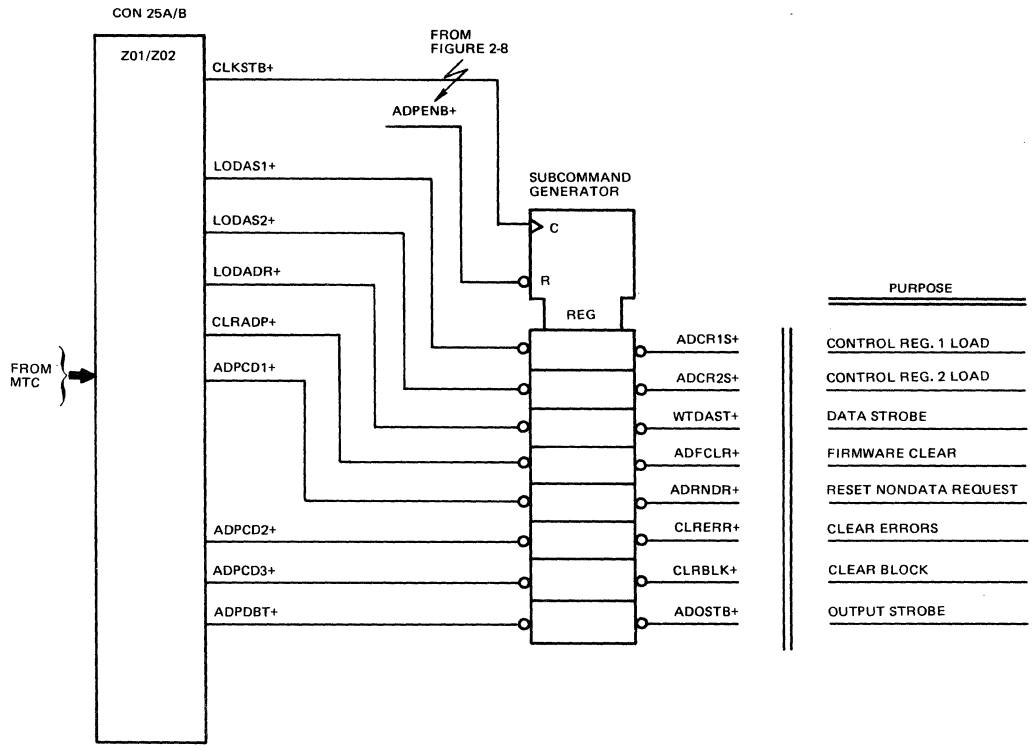


Figure 2-9 Subcommand Generator

The tape adapter has two control registers (see Figure 2-10) that store all the device commands and control information from the scratch pad memory to complete a device operation. These registers (control registers 1 and 2) are loaded with the information on the MTC data output lines (ALUOT0+ to ALUOT7+) when the subcommand generator stores the appropriate load control register command (ADCR1S+ or ADCR2S+). The registers are reset when the firmware loads the subcommand generator with Firmware Clear (ADFLCR+). The high-order bits (0 through 3) of control register 2 are loaded from the MTC interface enable signals (ADPENB+01 through ADPENB+03) and the tape adapter enable signal (ADPENB+04). The outputs of the control register (ADDRV0+ through ADDRV3+) are sent to the device interface drivers and used by the device as selection lines.

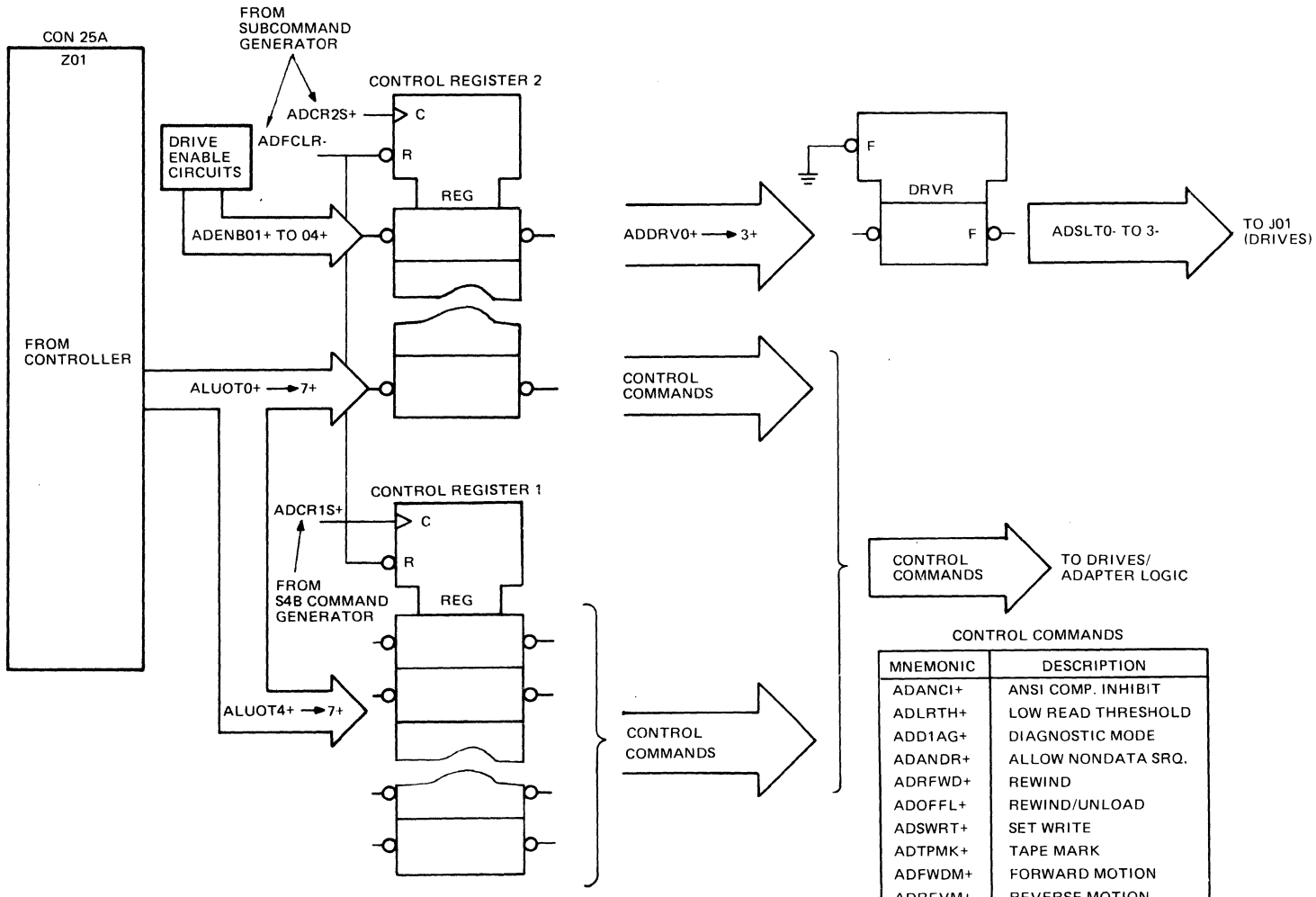
2.4.1.2 Write Clock Logic

The write clock logic is comprised of three primary elements: a crystal oscillator, a 1/8 frame generator, and a write clock. The nondata service request flip-flop and its associated logic are also considered part of the write clock logic (see Figure 2-11).

The crystal oscillator is a free-running 4.320-MHz crystal that generates pulses at a 231-nanosecond rate. This rate is the basis for all timing pulses generated within the adapter.

The 1/8 frame generator creates a pulse (FRMGEN+) at the rate of one per 1/8 frame time regardless of the device speed (75 ips or 45 ips). This is accomplished by utilizing the output of the oscillator (OSCOUT+00) as the clock input to the 1/8 frame generator and the High Speed (ADHISP+) signal derived from the device interface, and +5 Vdc as the preset. When a 75-ips device is attached, ADHISP+ is set and the counter is preset to 7 hex; when a 45-ips device is attached, the counter is preset to 1 hex. This allows FRMGEN+ (1/8 of a frame) to set once every 2 µs for a high-speed device and once every 3.5 µs for a low-speed device. (Both times are approximate.)

The write clock uses FRMGEN+ to clock its eight-output shift register. The first time FRMGEN+ sets, the output of the Clock Preset/Inhibit flip-flop negation (MTLDFB-) is set due to Firmware Clear (ADFCLR-) from the subcommand generator (see Figure 2-9). This allows bit 0 of the write clock (WTFRB0+) to set (this set condition is returned to the Clock Preset/Inhibit flip-flop), inhibiting the insertion of any other bits into the write clock. At each generation of FRMGEN+, the subsequent stage of the write clock is set and the preset stage resets, resulting in the development of eight uniform clock pulses during each frame time. The output of the last write clock stage (WTFRB7+) clocks the ground data input of the Clock Preset/Inhibit flip-flop, setting WTLDFB-. WTLDFB-, in turn, starts the cycle over again.



CONTROL COMMANDS

MNEMONIC	DESCRIPTION
ADANCI+	ANSI COMP. INHIBIT
ADLRTH+	LOW READ THRESHOLD
ADD1AG+	DIAGNOSTIC MODE
ADANDR+	ALLOW NONDATA SRQ.
ADRFWD+	REWIND
ADOFFL+	REWIND/UNLOAD
ADSWRT+	SET WRITE
ADTPMK+	TAPE MARK
ADFWDM+	FORWARD MOTION
ADREVM+	REVERSE MOTION
ADWDAE+	WRITE DATA ENABLE
ADADSV+	ALLOW DATA SERVICE REQUESTS

Figure 2-10 Control Registers 1 and 2

HONEYWELL PROPRIETARY AND CONFIDENTIAL

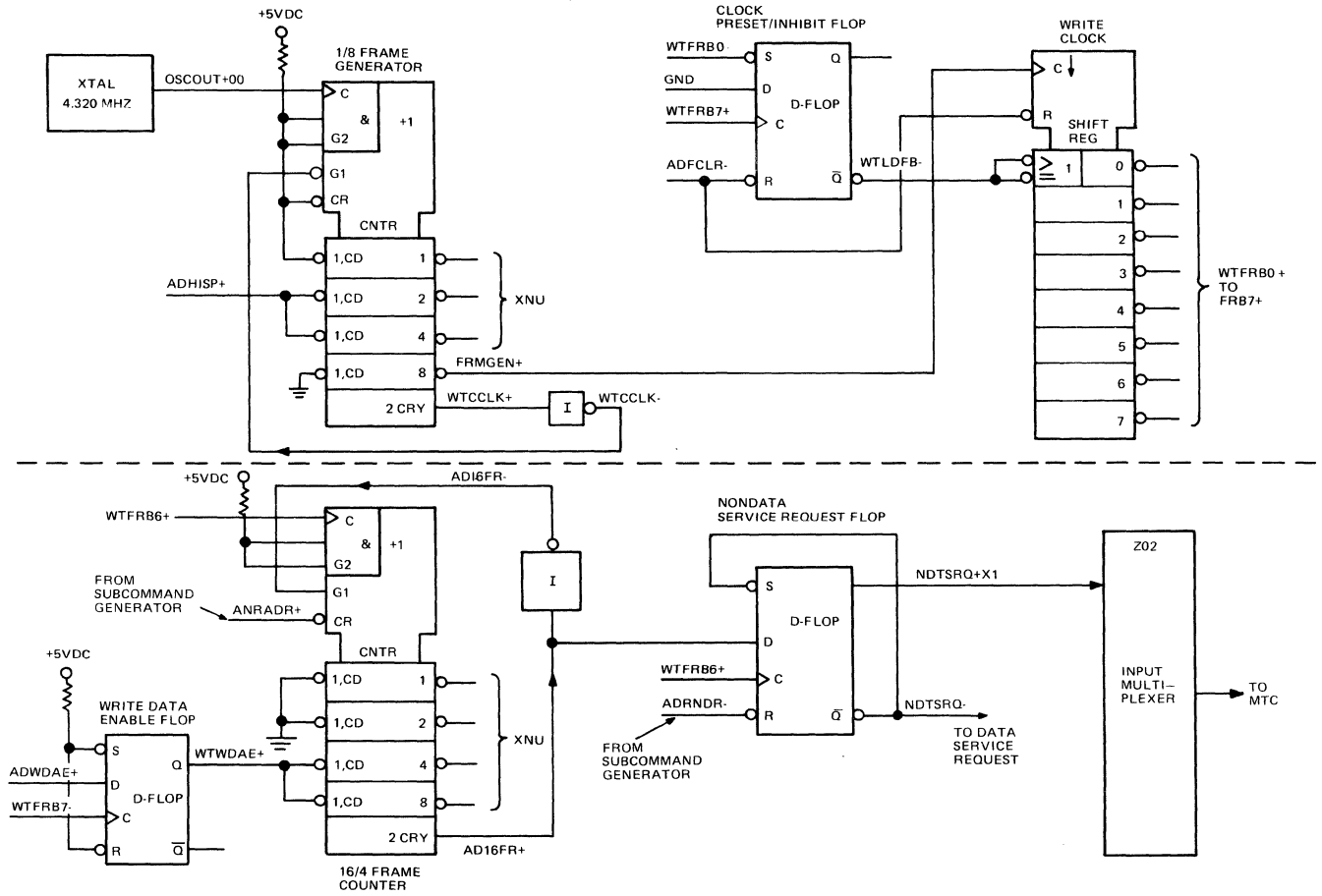


Figure 2-11 Write Clock Logic

The time interval between nondata service requests is determined by the carry-out function (AD16FR+) of the 16/4 frame counter. The counter is preset to 0 hex or C hex as determined by the Write Data Enable flip-flop (WTWDAE+). This preset is done to allow a nondata service request to be generated at 0.005-inch intervals of tape when a write data transfer is taking place or at 0.02-inch intervals of tape during initial gap generation. Firmware uses the nondata service request outputs (NDTSRQ±) to establish distances on tape for gaps, start-up and stay delays, timeouts, etc.

The Write Data Enable flip-flop (WTWDAE+) sets as the result of Write Data Enable (ADWDAE+) from control register 1 (see Figure 2-10) and the negation of the last stage of the write clock (WTFRB7-). The 16/4 frame counter increments at each sixth stage of the write clock (WTFRB6+) with the WTWDAE+ determining how many frames are to be counted (16/4) prior to the setting of the carry-out (AD16FR+) function. After the carry out sets, which is the

data input to the Nondata Service Request flip-flop (NDTSRQ+X1), NDTSRQ+X1 sets at the following sixth write clock output (WTFRB6+). NDTSRQTX1 is sent to the data request logic (see Figure 2-15) to notify the MTC that a request is present. It is also sent to bit 0 of the ID byte (see Figure 3-4) to signify to firmware that the present request is a nondata service request. The Nondata Service Request flip-flop remains set until the request is acknowledged and firmware loads Reset Nondata Service Request (ADRNDR+) into the Subcommand Generator.

2.4.1.3 Write Initiation

Prior to the initiation of a write order the Nondata Service Request flip-flop (NDTSRQ+X1) is reset, using the output of the subcommand generator (ADRNDR+). The status is also reset, using the Clear Errors (CLRERR+) output of the subcommand generator.

Control Register 1 is then loaded with the write command (29 hex) minus the write data enable bit (see Figure 2-10). This sets Write (ADSWRT+), Forward Motion (ADFWDM+), and Allow Data Request (ADADSV+) in the control register. At this point drive motion is started by sending ADFWDM+ from the control register to the device.

2.4.1.4 Write Data Path (See Figure 2-12)

With Write Data Enable reset and the conditions that are now set in control register 1, a Data Request (DATSRQ+X1) is issued to the controller each time the input register to the write FIFO is empty (WTDINE+) but at a rate not to exceed one-frame time intervals. The firmware loads Data Strobe (WTDAST+) into the subcommand generator, clocking the data from the controller (ALUOT0 through ALUOT7) into the data register. WTDAST+ from the subcommand generator is also the input to the Write FIFO flip-flop, and at CLKSTG+ time the output of the flip-flop (WTFIFO+) sets, allowing the output of the data register to be written into the write buffer (FIFO). This process is repeated until all 32 locations of the write buffer are loaded.

When the firmware has determined that the device is up to maximum speed and the proper gap has been allowed from the previous block, it loads control register 1 with the write command (2B hex), including Write Data Enable. With Write Data Enable (WTWDAE+) set, the contents of the write buffer are shifted with Write Data Out (WTDOUT+) each Write Clock Time 7 (WTFRB7+), and the data in the output register of the write buffer (ADOUT0+ through ADOUT7+ and ADOUTP+) is sent through an ORing network to the write data drivers (WTDAT0+ through WTDAT7+ and WTDATP+) and on to the device.

At the same time that data is being sent to the device, data requests (DATSRQ+X1) start being generated (see Figure 2-16) because the FIFO input register will again be empty (WTDINE+).

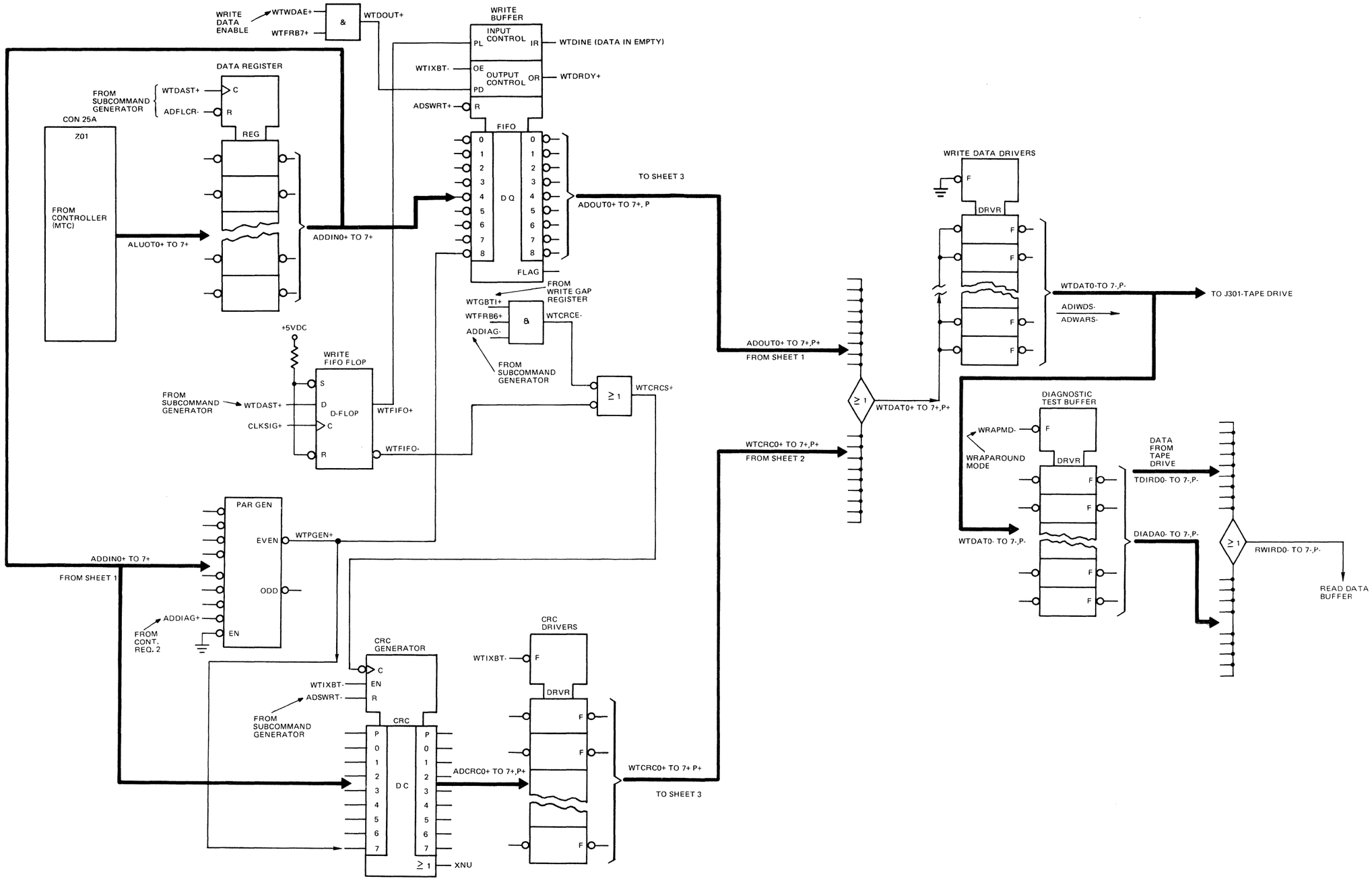


Figure 2-12 Write Data Path

As data is being sent to the adapter and loaded into the data register, it is also applied to a parity generator and a CRC generator. The parity generator creates odd parity (WTPGEN+) on each byte of data received using the data register output (ADDIN0+ through ADDIN7+). When the data is written into the write buffer (FIFO) at CLRSIG+ time, the parity bit (WTPGEN+) is also written.

The data (ADDIN0+ through ADDIN7+) and the parity bit (WTPGEN+) are written into the CRC generator at the same time as it is written into the write buffer. This is accomplished by using the output of the Write FIFO flip-flop (WTFIFO+) to develop the clock input to the CRC generator. The outputs (ADCRC0+ through ADCRC7+ and ADCRCP+) of the CRC generator are isolated from the write data drivers by the tristate CRC drivers (WTCRC0+ through WTCRC7+ and WTCRCP+). These drivers are enabled (WTIXBT-) when the CRC character is to be written on tape. The outputs of the drivers are sent through an ORing network (WTDAT0+ through WTDAT7+ and WTDATP+) to the write data drivers (WTDAT0- through WTDAT7- and WTDATP-), which are always enabled, and then on to the device.

2.4.1.5 Write Termination (See Figure 2-13)

When the firmware recognizes the end of the data transfer (end of range detected), it resets the Allow Data Requests (ADADSV+) in control register 1. This inhibits the adapter from generating any further data requests (DATSRQ+X1) and allows the data buffer (FIFO) to empty. When the last data byte has been transferred from the write buffer to the device, the Data FIFO Empty (WTDFFE+) function sets. WTDFFE+ set in conjunction with the output of the Write Data Enable flip-flop, sets the data input (WTSWPI+) to the write gap shift register (WTGBT0+ through WTGBT7+). At Write Clock Time 3 (WTFRB3+), the first stage of the shift register (WTGBT0+) is allowed; the inverted output is the force set of the Inhibit Extra Bits flip-flop (WTIXBT- reset). WTIXBT- remains set until the firmware resets the write data enable output of control register 1 (ADWDAE+). WTIXBT- from the Inhibit Extra Bits flip-flop prevents any further bits from being inserted into the data input of the write gap shift register and inhibits normal generation of Write Data Strobes (ADIWDS+).

WTIXBT- also enables the CRC drivers (see Figure 2-12) and inhibits the write buffer. This condition permits the write drivers to reflect the CRC output of the ORing network (WTCRC0+ through WTCRC7+ and WTCRCP+) rather than the data output (ADOUT0+ through ADOUT7+ and ADOUTP+).

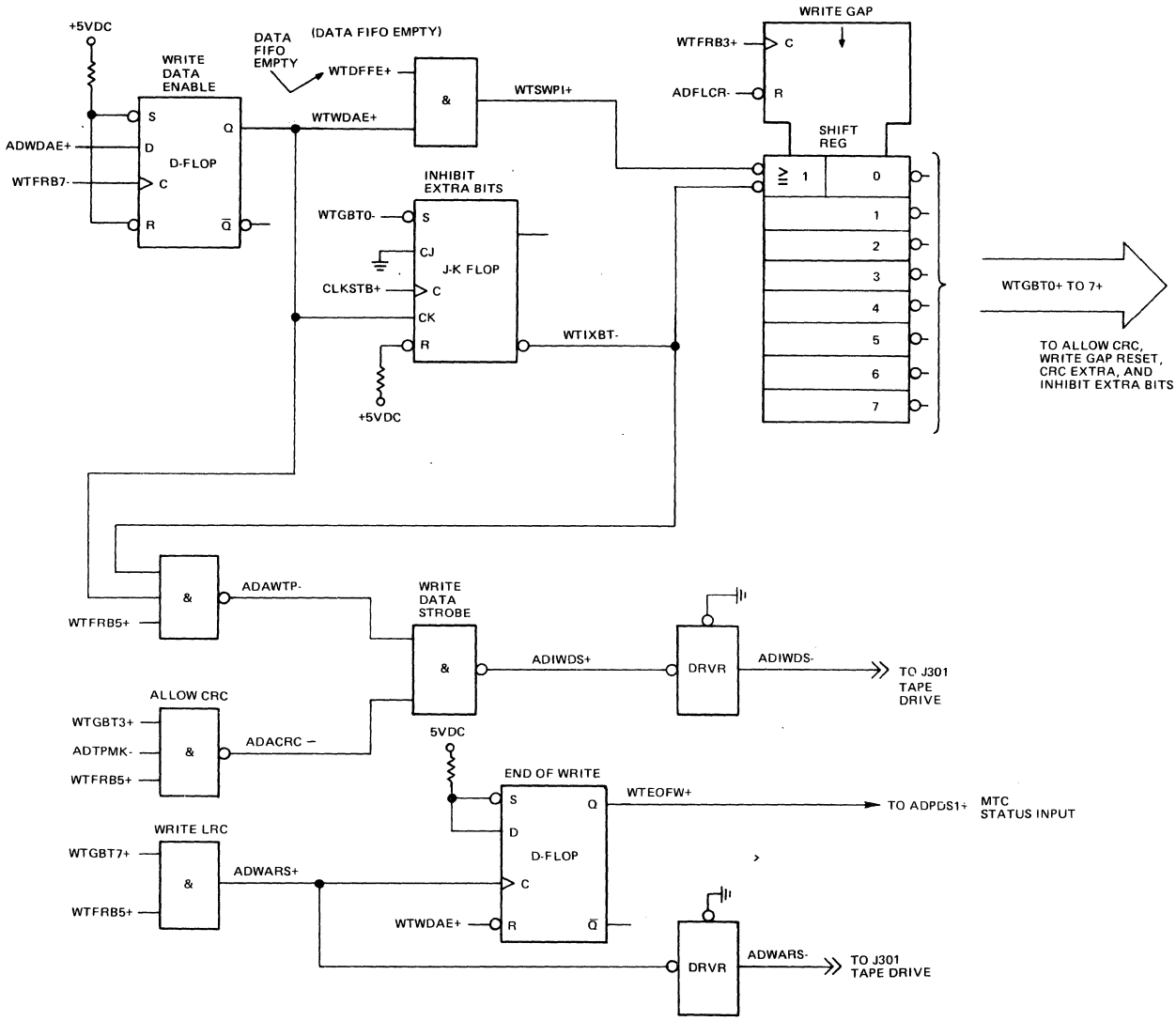


Figure 2-13 End of Write Logic

The write gap shift register is shifted once each frame time, with the output of the fourth stage (WTGBT3+) allowing the CRC character write, and the eighth stage output (WTGBT7+) writing the LRC character. When WTGBT3+ sets and if the operation is not a tape mark write (no CRC character is written for a tape mark block), the final Write Data Strobe (WDIWDS+) is generated at Write Clock Time 5 (WTFRB5+). This last data strobe allows the outputs of the write data drivers to be recorded on tape. At Write Gap Time 7 (WTGBT7+) and Write Clock Time 5 (WTFRB5+), the Write LRC function (ADWARS+) is set. The Write LRC function is fed to the device and causes the write amplifiers in the device to reset to their initial

write state. With the write amplifiers reset even parity is then reset in each of the nine tracks. ADWARS+ is also the clock input to the End of Write flip-flop (WTEOFW+), which signals the MTC firmware that the functional portion of the write operation has been completed.

The MTC firmware, delaying for the period of time necessary to create the gap, allows control register 1 and Write Data Enable (WTWDAE+) to remain set. With the device write current turned on and no data being sent to the tape, the tape motion creates an erased area on tape after the LRC character. When the gap has been generated, the firmware loads ADCR1S+ into the subcommand generator (see Figure 2-9) and sends Zeros (ALUOT0+ through ADLUOT7+) to control register 1 (see Figure 2-10). With control register 1 reset, the Write Data Enable flip-flop (WTWDAE+), the Inhibit Extra Bits flip-flop (WTIXBT-), and the End of Write flip-flop are reset, and tape motion is terminated as the result of Forward (ADISFC-) on the device interface being reset.

2.4.2 Read Operation

Subsections 2.4.2.1 through 2.4.2.4 describe the operation of the adapter hardware when a read operation is to be performed on a selected tape device. The data transfer path and the control, timing, and data integrity logic required to execute the read operation are discussed. The read path and associated logic are used for all space and read operations, for diagnostic write operations, and for the read after write. The hardware application is basically the same in all operations. The firmware determines how the transfers differ. Wherever possible the variations in the hardware utilization are noted.

2.4.2.1 Read Control Logic

The control registers implemented during read operations are the same as those for the write operation. For a description of this logic, refer to subsection 2.4.1.1 and Figures 2-9 through 2-12 of this manual.

2.4.2.2 Read Initiation

Prior to the initiation of a read operation (Read, Space, etc.), the Nondata Service Request flip-flop (NDSRQ+X1) is reset by the ADRNDR+ output from the subcommand generator (see Figure 2-9). The status and read buffer (FIFO) are also reset with the CLRERR+ output of the subcommand generator, and the block detection logic is cleared by the CLRBLK+ output. Tape motion is now initiated in a forward (ADFWDM+) or reverse (ADREVM+) direction, depending upon the type of operation, by loading 08 hex (forward) or 04 hex (reverse) into control register 1 (see Figure 2-10). The firmware now delays to allow the device to reach its maximum speed (start-up delay).

2.4.2.3 Read Data Path (See Figure 2-14)

During the start-up delay which firmware controls, the adapter implements the beginning-of-block (BOB) logic (shown in Figure 2-15) to detect the start of a valid data block. The BOB logic consists of a BOB counter and a BOB flip-flop; it determines valid data blocks when American National Standards are to be adhered to or even when they are not to be adhered to (ADANCI+). The BOB counter is clocked with each read strobe (TDIRDS+) from the device except the first. When the first read strobe comes from the device, the First Character (ADFSTC+) flip-flop (see Figure 2-14) sets and presets the BOB counter to F hex (no American National Standards) or 5 hex (American National Standards, 12 characters minimum record). With the carry-out of the counter (AD2NNR+) as the data input to the BOB flip-flop, either the second or the twelfth data strobe (TDIRDS+) will set the BOB flip-flop. The output of this flip-flop is interrogated by firmware, and when it has been set, the complete task, with allow data service requests (ADADS4+), is loaded into control register 1 by firmware.

The read data (TDIRD0+ through TDIRD7+) is loaded into the read buffer FIFO by the adapter Read Data Strobe (ADIRDS+) as the input control. Because this is a fall-through FIFO which can store up to 32 bytes, the Output Register Ready (ADORDY+) sets as soon as the data becomes valid in the FIFO. If this is not a space or RAW operation, the Allow Data Requests (ADADSV+) output of control register 1 is set. ADADSV+, in conjunction with the Beginning-of-Found Block (ADBOFB+), is the data input to the Read Request (RDDATR-) flip-flop (see Figure 2-15). The flip-flop is clocked each time that the FIFO output register is ready (ADORDY+) and CLKSTB+ is received from the MTC. This generates a Data Request (DATSRQ+) that is sent back to the MTC, it remains set until the data request is serviced.

When the firmware services the data request, it enables the adapter (ADPENB- low) and configures the select lines for the input multiplexer to read the data (UPIR04+ and UPIR05+ low). This process allows the multiplexer outputs (ADPDS0+ through ADPDS7+) to reflect the read buffer FIFO input (ADDAT0+ through ADDAT7+).

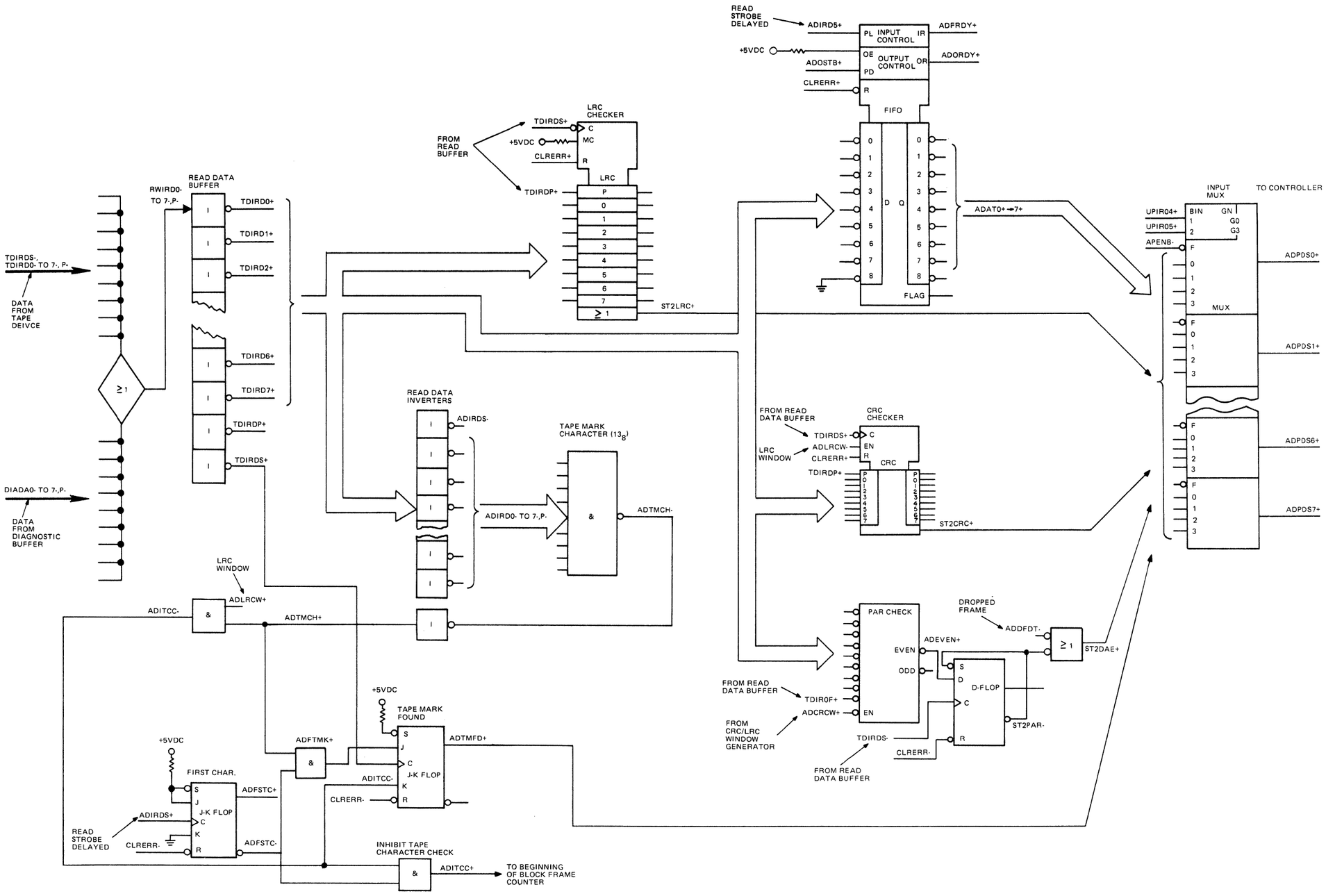


Figure 2-14 Read Operation

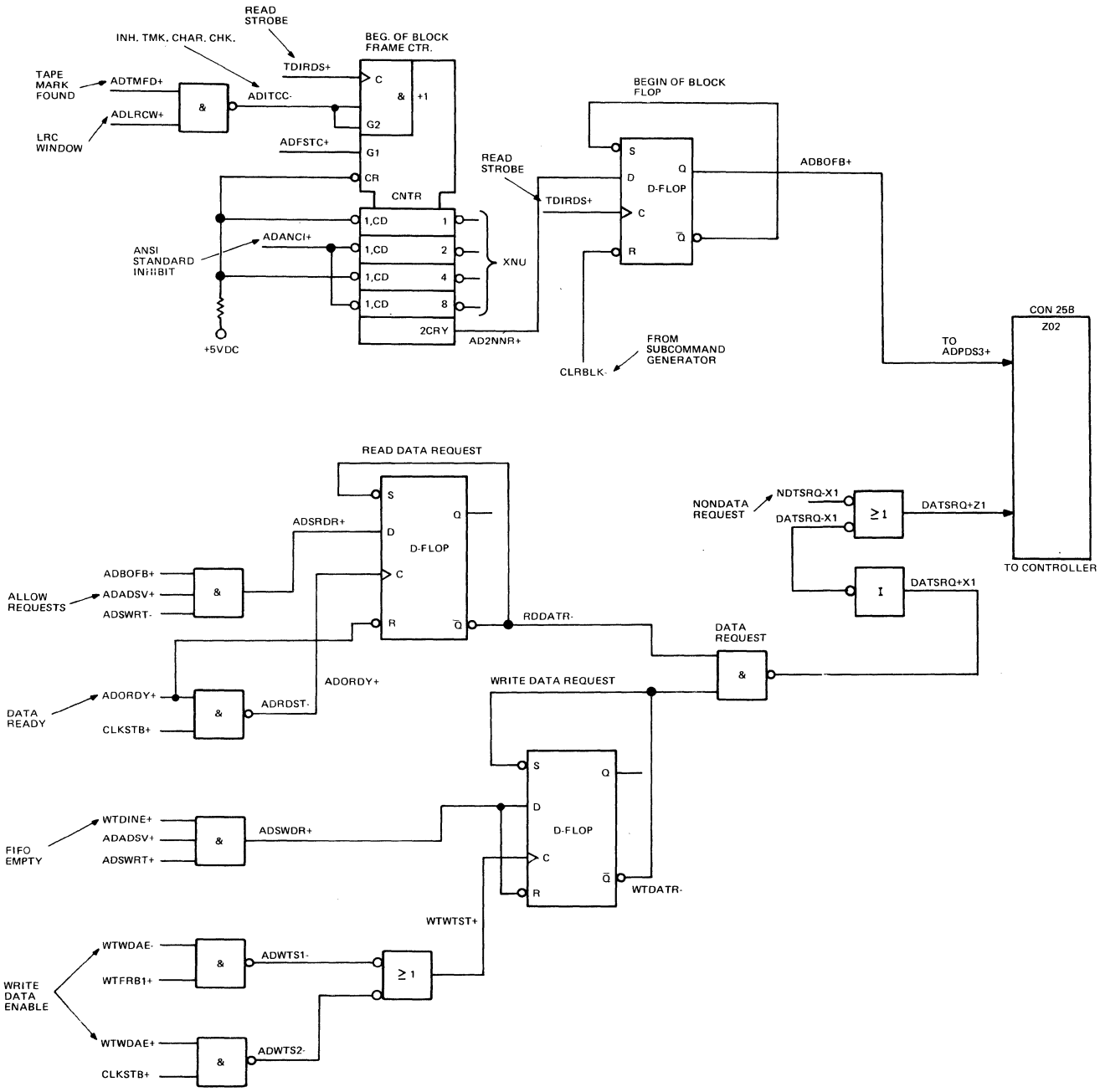


Figure 2-15 Data Request Logic

Tape Mark Detection (See Figure 2-14)

Each time a read operation is performed, the first character is examined to determine if it is a tape mark character (13 hex). The determination is accomplished by inverting the data from the device (TDIRD0+ through TDIRD7+ and TDIRDP+) and using some of the inverted outputs (ADIRD0- through ADIRD7- and ADIRDP-) and some of the device data to compare for a hexadecimal 13. The output of the comparator (ADTMCH-) is inverted and gated with the output of the First Character flip-flop (ADFSTC-). The resultant function (ADFTMK+) is the data input to the Tape Mark Found flip-flop (ADTMFD+). If set, ADFTMK+ is latched in at read strobe time (TDIRDS+). The data input to the Tape Mark Found flip-flop is inhibited after the first read strobe by the setting of the First Character flip-flop (ADFSTC- low). The following LRC character must also be hexadecimal 13 or it resets the Tape Mark Found (ADTMFD+) flip-flop. Tape Mark Found (ADTMFD+) is sent to the adapter/MTC interface and informs the firmware that a successful tape mark operation has been performed.

Data Path Integrity Checking (See Figure 2-14)

Three types of data integrity checks are performed on the data from the device: parity check, CRC check, and LRC check.

As the data is read from the device (TDIRD0+ through TDIRD7+ and TDIRDP+), it is applied to the inputs of the parity checker. The parity checker is always enabled except during the reading of the CRC character (ADCRCW+), as all the other information written on tape (data, LRC, etc.) has valid parity. The even output (ADEVEN+) of the checker is sampled with each read strobe (TDIRDS+) from the device, and if ADEVEN+ is set, the Parity Error flip-flop (ST2PAR+) sets and remains set until it is reset by firmware (CLRERR-).

The CRC checker receives data (TDIRD0+ through TDIRD7+ and TDIRDP+) from the device and generates a running CRC character inclusive of each byte of data. The CRC checker is loaded with each read strobe (TDIRDS+) from the device and is enabled for all data from the device except the LRC character (ADLRCW-). When all the data has been read and the CRC character from the device has been loaded into the CRC checker, the unequal result indicator from the CRC checker (ST2CRC+) should be low.

If there is no further data coming from the device (no read strobes), and if the error output (ST2CRC+) is set, ST2CRC+ remains set until cleared by the firmware (CLRERR+).

The LRC checker receives all data (TDIRD0+ through TDIRD7+ and TDIRDP+) from the device and is loaded for each read data strobe (TDIRDS+) generated by the device. The LRC checker performs a half add on each bit of the data in each track. When all the data has been read (data, CRC, and LRC), the result for each track must be zero, and the accumulation indicator (ST2LRC+) for all the tracks must be low. If the indicator (ST2LRC+) is set when the read strobes stop (after the LRC character), it remains set until cleared by the firmware (CLRERR+).

All three types of integrity checks (parity, CRC, and LRC) supply indicators (ST2PAR+, ST2CRC+, and ST2LRC+) to the adapter input multiplexer which can be interrogated by the firmware's reading of status register 2 (see Figure 2-4 or 3-3).

2.4.2.4 Read Termination (See Figure 2-16)

The end of a data block is detected by the end-of-read/RAW logic in the adapter hardware. The end-of-read/RAW logic has a gap detector counter to determine if more than two frame times have elapsed since a read strobe was received from the device. This counter is clocked by a FRMGEN+ from the write clock logic (see Figure 2-11), which causes a carry-out (ADGPFD+) approximately every two frames. The counter is reloaded (to zero) each time a read strobe (TDIRDS-) is received from the device, thereby inhibiting a carry-out when the read strobes are maintained at a one-frame rate.

When data strobes (TDIRDS-) stop, the counter carries out, setting the Generate Read Check Window (GENRCW+) flip-flop. This condition inhibits the resetting of the counter by the Load Counter flip-flop (LODCTR-) and allows the counter to run fully. GENRCW+ is also the data input to the CRC/LRC window generator that allows the CRC and LRC characters to be read without causing an error. DRDCTD+ from the gap detector counter is the clock input to the window generator; therefore, with GENRCW+ set the first time DRDCTD+ sets (one frame time later), the generator is loaded (three frame times total after last read strobe). The output of the first stage (ADCRCW+) sets the Look for Dropped Frame flip-flop (ADLFDF+), which enables the Dropped Frame Detected flip-flop (ADDFDF-) and inhibits the data input to the CRC/LRC window generator. The CRC Window (ADCRCW+), which is up for two frame times, causes the data input to the Dropped Frame Detected flip-flop (CRCLRC-) to go low for the read strobe coming in with the CRC character. The window generator is stepped by DRDCTD+ twice more, and the LRC window (ADLRCW+) sets and inhibits the Dropped Frame Detected flip-flop from setting in the same manner as did the CRC Window (ADCRCW+). The window generator is stepped once more with DRDCTD+, and this stage is the Clear Read Check Window (CLRRCW+), which resets the Generate Read Check Window flip-flop (GENRCW+) and allows any further read strobes from the device to reset the gap detector counter to zero.

When the CRC and LRC characters have been read, the carry-out (ADGPFD+) of the gap detector counter enables the EOB delay counter to be incremented every two frames. If no read strobes are encountered, the EOB delay counter carry-out (GAPFND+) sets 32 frame times later (40 total from last read strobe: 8 for LRC and 32 for delay), which is the data input to the End-of-Block flip-flop (ADEOFB+).

When the firmware detects that the End-of-Block flip-flop is set, it loads ADCR1S+ into the subcommand generator (see Figure 2-9) and sends Zeros (ALUOT0+ through ALUOT7+) to control register 1 (see Figure 2-10). With control register 1 reset, tape motion terminates as a result of the motion signals (ADISFC- and ADIRWC-) on the device interface resetting.

2.4.3 Rate Error Detection (See Figure 2-17)

When data is being transferred on a write or read operation, a check is maintained to determine that no data is lost due to a rate error. For this purpose the adapter supplies a data overrun/underrun indicator to the MTC interface for interrogation by the firmware. This indicator (ST2DOU+) is a flip-flop that is clocked with each read strobe (TDIRDS+) for a read operation or at each Write Clock Time 4 (WTFRB4+) for a write operation.

The data input is set when: there is not data in the write buffer (WTDRDY-), the Write Data Enable (WTWDAE+) is set, and adapter service requests are allowed (ADADSV+) for a write operation. This combination of conditions represents an underrun condition (ADWRUN-) and sets the indicator.

The data input is also set when the read buffer is full (ADFRDY-), adapter service requests are allowed (ADADSV+), and Read Strobe (TDIRDS+) from the tape device occurs for a read operation. This represents an overrun condition (ADRDOV-) and sets the indicator.

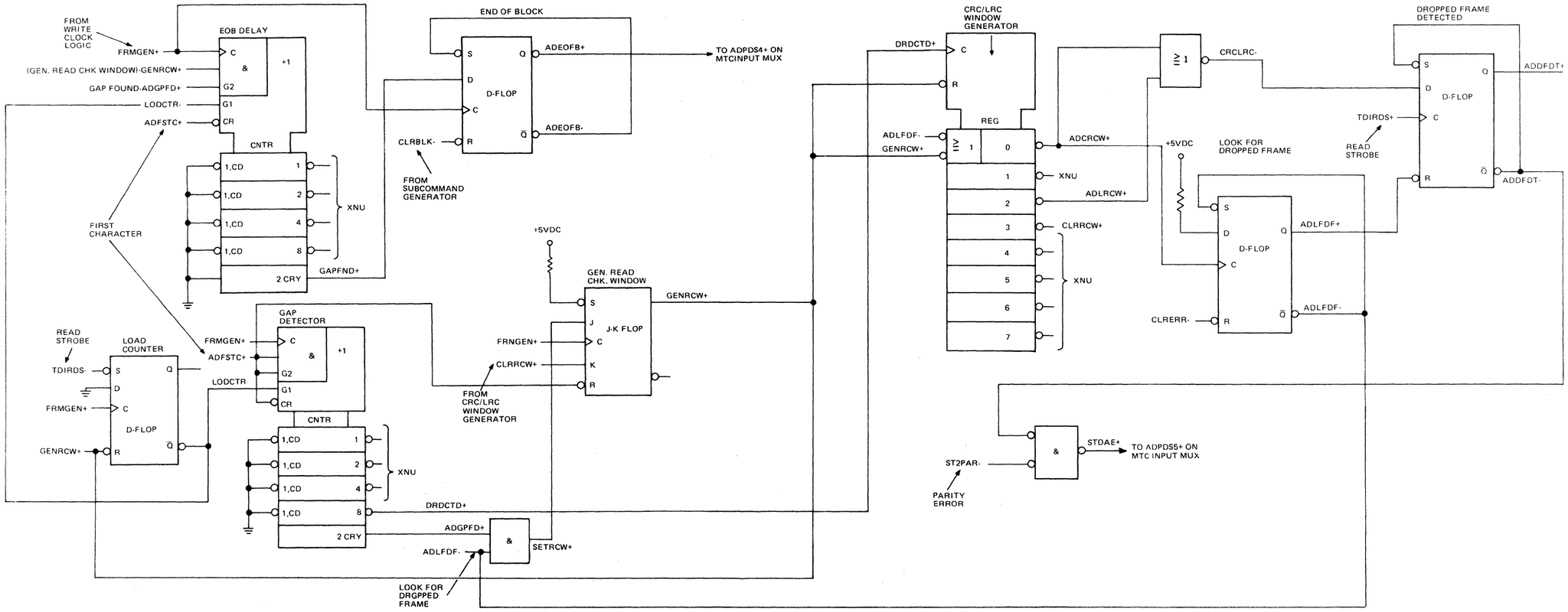


Figure 2-16 End-of-Read/RAW Logic

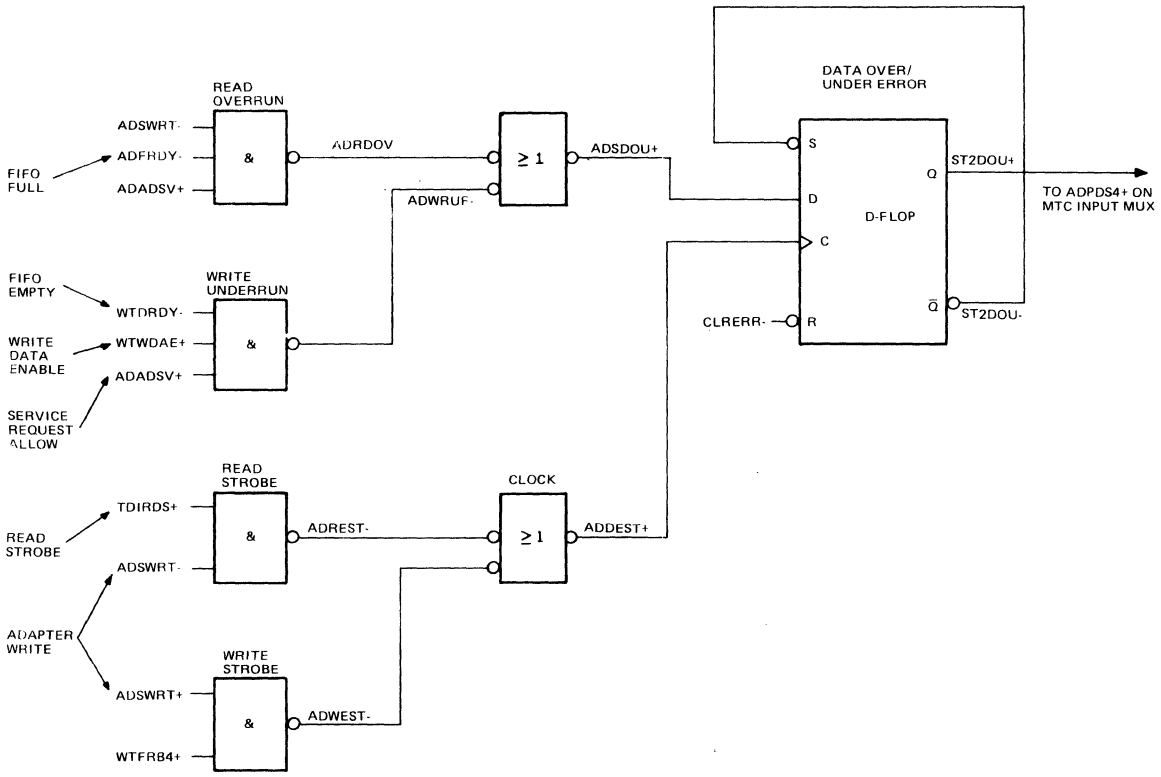


Figure 2-17 Rate Error Detection

III
THEORY OF OPERATION - CYCLE FLOW

The tape adapter firmware is comprised of routines which are formed from various types of firmware commands. Firmware commands, 16-bit encodings of microinstructions, are housed in the device support routine area of the MTC microprogram control store (UPCS). When read from the UPCS and decoded, firmware commands result in the enabling of certain functions or in the specific action of various hardware elements. Hardware operations occur sequentially as designated firmware commands are executed in series.

The tape adapter firmware routines, in conjunction with the MTC firmware routines, provide an operational link between the Level 6 software and the nine-channel tape device subsystem. Software commands are interpreted and executed by firmware decoding of the command. The result of the MTC firmware decoding causes the MTC to exit from the scanning process and enter the appropriate tape adapter firmware routine.

The adapter-supporting firmware is divided into two major functional areas: process management routines and execution routines. The routines within each of these two areas are described in subsections 3.2 and 3.3, respectively. Adapter firmware sequencing continues until all the routines required to perform a software-designated operation have been processed. Upon completion of the prescribed processing, the adapter firmware exits to the MTC routines. These routines determine and implement any further software commands.

The MTC, in addition to the segment of the UPCS dedicated to the tape adapter firmware, also allocates one quadrant of a 256-location Scratch Pad Memory (SPM) for each tape drive attached to the adapter. These quadrants contain device-specific information, and the locations within the quadrant can be interrogated or manipulated by the firmware for interpreting or updating status, configuration, control, etc., information.

3.1 FIRMWARE COMMANDS

The various types of firmware commands, utilized to perform the tape adapter firmware routines, are the same as those implemented by the MTC. (Refer to Section IV of the Type MTC9101 MTC Manual, Order No. FM88.)

3.2 SCRATCH PAD MEMORY

The tape adapter is provided with scratch pad memory availability of up to 256 locations, each of which is 8 bits deep. The alterable contents of the SPM are utilized to store information or to store control bytes which direct the actions of each device attached to the adapter. The 256 locations are divided into 64 location areas (see Figure 3-1) for dedication to each specific device attached to the adapters. Table 3-1 is a listing of the topology of a single quadrant indicating the address for each byte relative to the base location of the segment selected by the scratch pad memory index register (refer to the MTC product manual hardware description). Where the application of a control or parameter byte is unique to the nine-channel tape adapter subsystem, a general usage of and the terminology associated with it will be supplied in Table 3-2. Figures 3-2 through 3-6 are utilized to indicate the purpose of bytes whose bit structure has a bit significant application.

3.3 FIRMWARE FLOW

3.3.1 Overview

The tape adapter firmware is divided into two major segments: process management routines and execution routines. The Process Management routines perform all the functions required for setup prior to the initiation of a specific task, and the Execution routines are responsible for the physical activity necessary to execute the task. Figure 3-7 is a diagram of the routines which comprise these two segments and the general paths between them.

3.3.2 Process Management Routines

3.3.2.1 Start Routines

The Start routine is the initialization segment of the tape adapter firmware. It is the initial entry point, and any subsequent entries into adapter firmware are made via the return address registers of the MTC which are loaded with the appropriate addresses by tape adapter firmware.

Entry into this routine is from the MTC Point routine for initialization, and from the MTC Bus Request routine via the Point routines for a Stop I/O command. This routine performs the necessary setup to implement these two operations.

3.3.2.2 Next Routine

The Next routine is a steering segment of firmware which is utilized during the idle time between task execution to process Nondata Service Requests generated as a result of the internal timer. The routine essentially makes the decisions to initiate a task for execution if any are stacked or to poll device status if no tasks are stacked.

3.3.2.3 Go Routines

The tape adapter Go routine accepts a task issued from the Bus Request routine of the common MTC firmware. This routine performs the necessary setup functions associated with the acceptance of a task, the most important of which is to enqueue the channel (in TSKQ) which is to be set busy.

The Go routine examines the task pointer to determine if another channel is already active. If a channel is active, the MTC Wait routine is entered leaving the task stacked in the task queue. When no other channel is presently executing a task, the adapter Roll-In routine is entered to unstack channels from the task queue.

3.3.2.4 Module Access Routines

The three segments of the Module Access routines are steering routines which compensate for the tape adapter hardware characteristic of issuing all requests to the MTC on channel 0. Port 0 needs no access routine because it is directly accessed.

The Access routines are used only during task execution on their respective channels. These routines select the proper channel of the MTC to replace channel 0 which was selected via hardware action when the adapter request was issued.

As an example, the Module Access routine for channel 2 will perform the following operations:

- Entry to the tape adapter firmware is made using return register 0 with MTC port 0 selected
- The firmware forces selection of port 2
- Entry is made via return register 2 to the proper point in the execution routines for the device active or channel 2.

3.3.2.5 Roll-In Routine

The Roll-In routine performs the process of unstacking previously enqueued tasks. The channel number in the topmost Task Queue Cell (TSKQ) is removed and placed into the task pointer to denote to firmware its status as the currently executing channel.

The remaining tasks (channel numbers) in the other three task queues are shifted up one position to reorder the sequence of priorities. The new task pointer is analyzed to determine which channel (0, 1, 2, 3) is to become active and the appropriate channel is set to a busy condition. Exit is made to the Task Decode routine for task examination and entry to the correct execution routine.

3.3.2.6 Poll Routine

The Poll routine effectively maintains updated status information in the MTC scratch pad memory. Since device level interrupts are not generated as a result of device status changes, the current device status must be kept by sequential interrogation of each device at some interval to detect any status change.

The Poll routine is entered from the Next routine when it has been established that no tasks are pending for execution (i.e., the task queue is empty). This routine references the Polling Pointer (POLP), an indicator of the last channel polled on the previous polling cycle, and increments the pointer to the next MTC channel. This new channel number is compared to a Polling Queue (POLQ) to ascertain if the new channel selected is actually a tape channel. The Polling Queue is an indicator of all the available tape channels on the MTC. The new channel to be polled is selected, the device status is read, and the scratch pad memory is updated if necessary.

3.3.2.7 Task Decode

The Task Decode routine examines the task word location of the scratch pad memory and exits to the appropriate execution routine to execute the task (i.e., Rewind, Write, etc.).

3.3.3 Execution Routines

The execution routines available to the tape adapter subsystem for the execution of device specific operations are:

- Rewind/Rewind and Unload routine
- Jog Erase routine
- Forward/Reverse routine (all read and space operations)
- Write routine
- Termination routine.

Each of the routines performs the functions designated and any in-process setup or verification required. The Termination routine ceases device motion, determines error and status conditions, and exits to the MTC Interrupt routine to report the status to software.

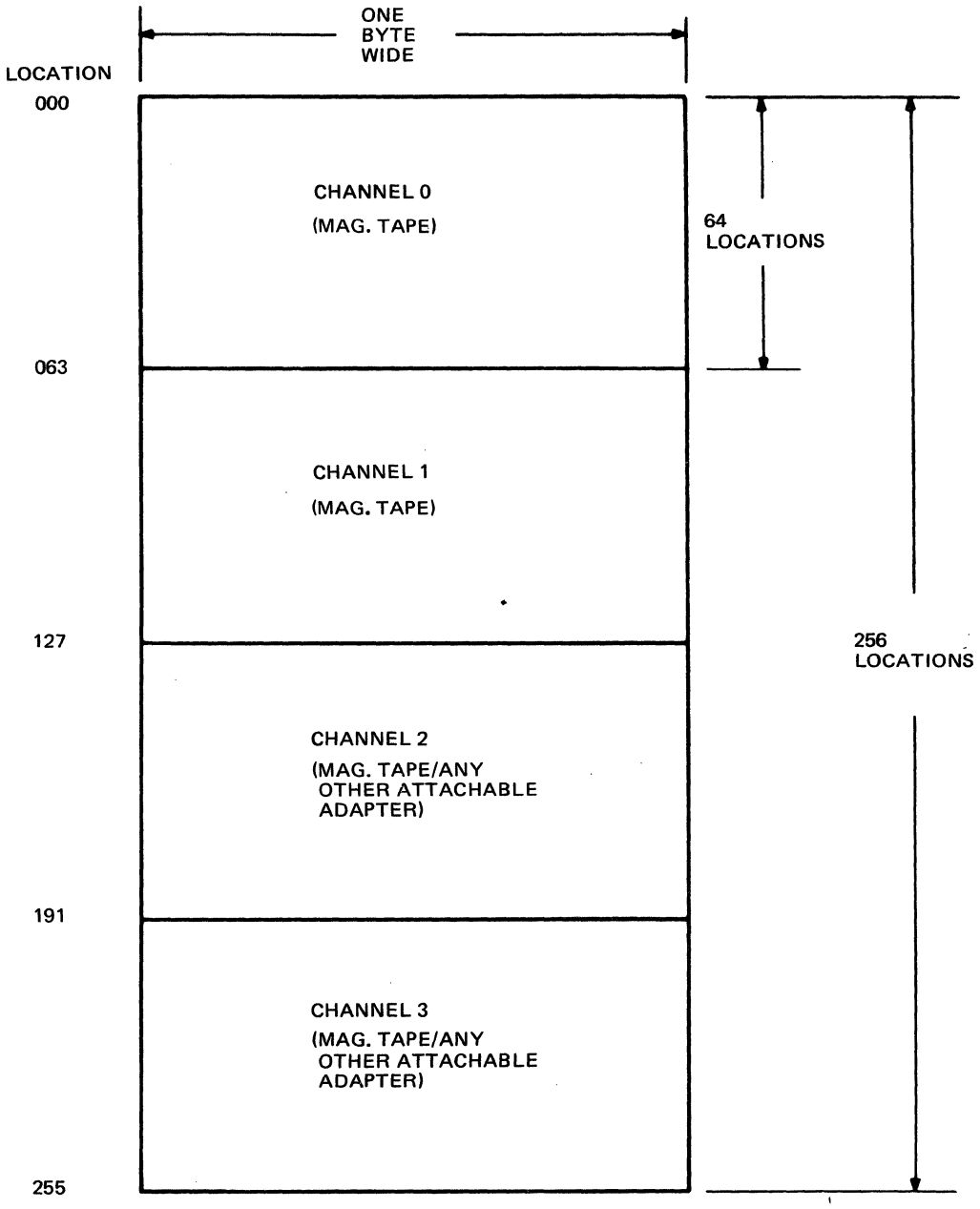


Figure 3-1 Scratch Pad Memory

Table 3-1 Scratch Pad Memory Topology
(Sheet 1 of 2)

ADDRESS	MNEMONIC	TERM
00	CWD1	Control Word, LSB[a]
01	CWD2	Control Word, MSB[a]
02	ILC1	Interrupt Level, LSB
03	ILC2	Interrupt Level, MSB
04	SFC1	Startup function code
05	FWRV[b]	Firmware revision
06	TSK1[c]	Task, LSB
07	TSK2[c]	Task, MSB
08	ADR1	Address, LSB
09	ADR2	Address, MSB
0A	MOD1	Module Address
0B	ENQB	Enqueue Buffer
0C	RNG1	Range, LSB
0D	RNG2	Range, MSB
0E	Spare	-
0F	Spare	-
10	CNF1[c]	Configuration Word 1, LSB
11	CNF2[c]	Configuration Word 2, MSB
12	CNF3[c]	Configuration Word 3, LSB
13	CNF4[c]	Configuration Word 4, MSB
14	Spare	-
15	Spare	-
16	Spare	-
17	Spare	-
18	STS1[c]	Status Word 1, LSB
19	STS2[c]	Status Word 1, MSB
1A	STS3[c]	Status Word 2, LSB
1B	STS4[c]	Status Word 2, MSB
1C	POLQ[c]	Polling Queue
1D	POLP[c]	Polling Pointer
1E	TSKQ[c]	Task Queue
1F	TSKP[c]	Task Pointer
20	DTA1	Data, LSB
21	DTA2	Data, MSB
22	CMSK	Channel Mask
23	Spare	-
24	MON1	Channel Monitor
25	DMA1	DMA Control
26	DID1[c]	Device ID, LSB
27	DID2[c]	Device ID, MSB
28	CHN1	Channel Number, LSB
29	CHN2	Channel Number, MSB
2A	CPC1	CP address, LSB
2B	CPC2	CP address, MSB
2C	IDF1	Interrupt vector, LSB
2D	IDF2	Interrupt vector, MSB

Table 3-1 Scratch Pad Memory Topology
(Sheet 2 of 2)

ADDRESS	MNEMONIC	TERM
2E	WL01[c]	Work Location 1
↓	↓	↓
3C	WL15[c]	Work Location 15
3D	RICP	Resume Interrupt Control
3E	TMW1[b]	Test Mode Work Location 1
3F	TMW2[b]	Test Mode Work Location 2

[a]Least significant bit = LSB and most significant bit = MSB.

[b]Utilized by software only.

[c]Device-specific location.

Table 3-2 SPM Word Description
(Sheet 1 of 3)

MNEMONIC	TERM	DESCRIPTION
TASK1 and TSK2	Task Word	The task word consists of two bytes. The least significant byte is not used by the nine-channel tape adapter, and the most significant byte contains a hexadecimal code which determines the operation to be performed on the tape drive.
CNF1 through CNF4	Configuration Words	The two configuration words consist of four bytes. Of the two available words, only one is utilized. The most significant byte of this word is used to generate control information on the adapter to send to the tape drive. Figure 3-2 is a bit-relevant representation of configuration word 1.
STS1 through STS4	Status Words	The two SPM status words (four bytes) are composites of information in the two adapter status registers and firmware-generated status. Figure 3-3 shows the relationship of the device-reported status, the adapter status registers, and the two SPM status words.

Table 3-2 SPM Word Description
(Sheet 2 of 3)

MNEMONIC	TERM	DESCRIPTION
POLQ	Polling Queue (one location for all channels)	The polling queue is the result of setting a bit for each attached tape drive. As the tape system is a multiple of a two-drive system, the only possible polling queue bytes are 00, C0, and FF hex. This byte is used to designate if the requested channel is a magnetic tape channel for status polling purposes.
POLP	Polling Pointer (one location for all channels)	The polling pointer is a byte used as an indicator to designate the channel which is currently being polled for status changes. One bit is used to signify each channel: Bit 0 = Channel 0 (80 hex) Bit 1 = Channel 1 (40 hex) Bit 2 = Channel 2 (20 hex) Bit 3 = Channel 3 (10 hex)
TSKQ(0-3)	Task Queue (Four locations)	Each channel (four) has a task queue location. These locations are used to stack the channel numbers of those channels that have tasks stalled. The four task cells of the queue (TSKQ0-3) may contain the channel numbers (80, 40, 20, and 10 hex) in any order depending upon the order of the task initiations. TSKQ0 has the number of the next channel to be executed, and a task to be stacked will be put in the first empty cell. When TSKQ0 is utilized each task cell content is shifted one cell position (see Figure 3-4).
TSKP	Task Pointer (one location for all channels)	The task pointer is a byte used to store the previous contents of TSKQ0 (80, 40, 20, and 10 hex) after the task queues have been shifted. This byte represents the channel which is currently executing its task (see Figure 3-4).

Table 3-2 SPM Word Description
(Sheet 3 of 3)

MNEMONIC	TERM	DESCRIPTION
CMSK	Channel Mask	The channel mask is divided into two parts: Bits 0 to 3 and 4 to 7. The first four bits are defined in the Type MTC9101 MTC Manual. A bit definition of the last four bits can be found in Figure 3-5.
DID1 and DID2	Device Identification	Device identification is coded in two bytes. DID2 (MSB) is generated by the MTC. DID1 (LSB) has a bit-specific application illustrated in Figure 3-6.

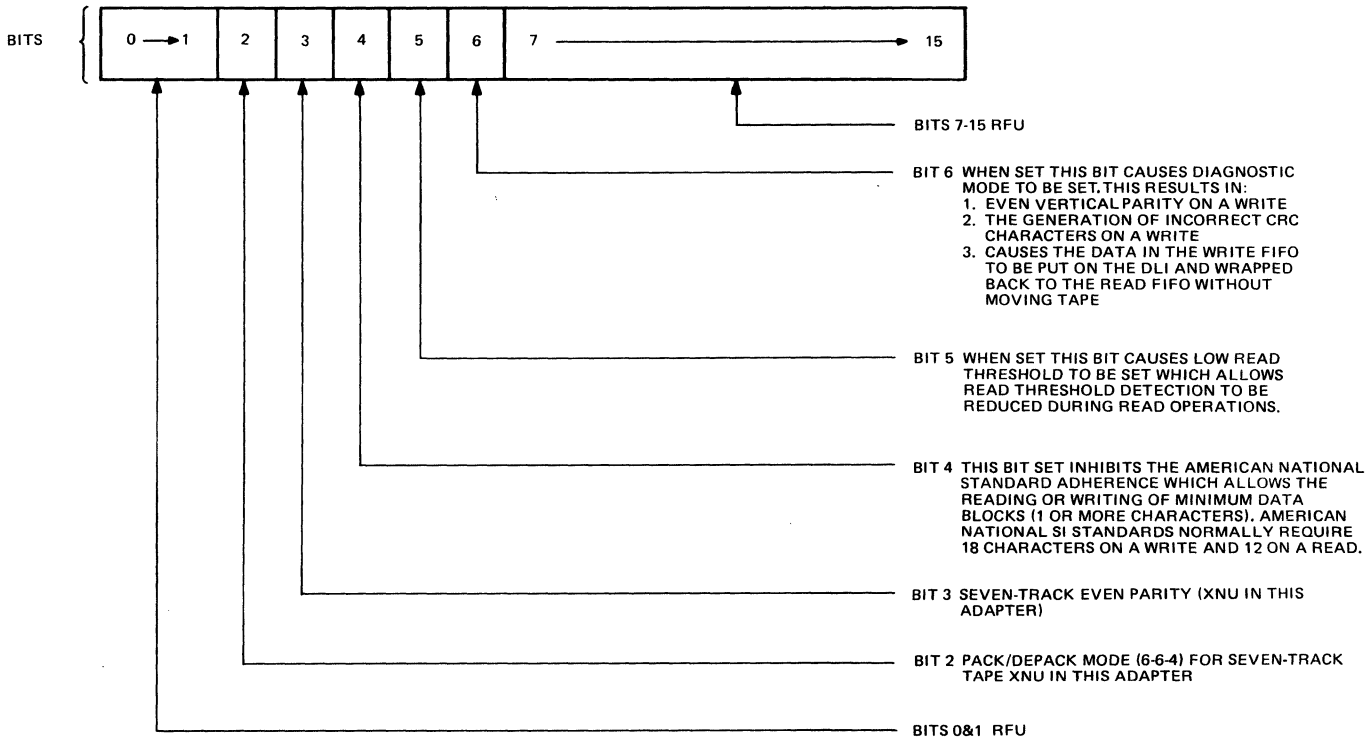


Figure 3-2 Configuration Word 1 Bit Significance

MTC SPM STATUS WORD BIT CONFIGURATION

STATUS BIT NOMENCLATURE	BIT	STATUS WORD ONE		STATUS WORD TWO	
		STS 2	STS 1	STS 4	STS 3
* DEVICE READY	0	1			
ATTENTION	1	1			
RETRYABLE MEDIA ERROR	2	1			
RFU	3	0			
NOISE RECORD DETECTED	4	1			
** TAPE MARK DETECTED	5	1			
* BEGINNING OF TAPE	6	1			
* END OF TAPE	7	1			
UNEQUAL LENGTH CHECK	8		1		
NON-RETRYABLE ERROR	9		1		
RFU	10		0		
OPERATION CHECK	11		1		
CORRECTED MEMORY ERROR	12		1		
NON-EXISTENT RESOURCE ERROR	13		1		
MEGABUS PARITY ERROR	14		1		
UNCORRECTABLE MEMORY ERROR	15		1		
* DEVICE ON LINE	0			1	
* DEVICE REWINDING	1			1	
* DEVICE FILE PROTECT	2			1	
* DEVICE HI DENSITY (SEVEN-TRACK)	3			1	
** DATA RATE ERROR	4			1	
** DROPPED CHAR./FRAME PARITY ERROR	5			1	
** CRC ERROR	6			1	
** LRC ERROR	7			1	
RFU	8				0
RFU	9				0
TIMEOUT	10				1
FUNCTIONALITY CHECK	11				1
BOB EARLY	12				0
BOB LATE	13				0
EOB EARLY	14				0
EOB LATE	15				0

*-DEVICE LINE INTERFACE INPUT, THROUGH ADAPTER, TO MTC STATUS WORDS

**-ADAPTER GENERATED STATUS TO MTC STATUS WORDS

BIT	ADAPTER STATUS REGISTER 1 INPUT TO MTC	DEVICE INPUT
0	READY	IRDY
1	END OF WRITE	
2	NONDATA SERVICE REQUEST	
3	BEGINNING OF BLOCK	
4	END OF BLOCK	
5	TAPE MARK DETECTED	IEOT
6	BEGINNING OF TAPE	IBOT
7	END OF TAPE	

BIT	ADAPTER STATUS REGISTER 2 INPUT TO MTC	DEVICE INPUT
0	ON LINE	IONL
1	REWINDING	IRWD
2	FILE PROTECT	IFPT
3	HI-DENSITY (SEVEN-TRACK)	
4	DATA RATE ERROR	
5	DROPPED CHARACTER/FRAME PARITY ERR.	
6	CRC ERROR	
7	LRC ERROR	

Figure 3-3 Adapter/MDC Status Relationship

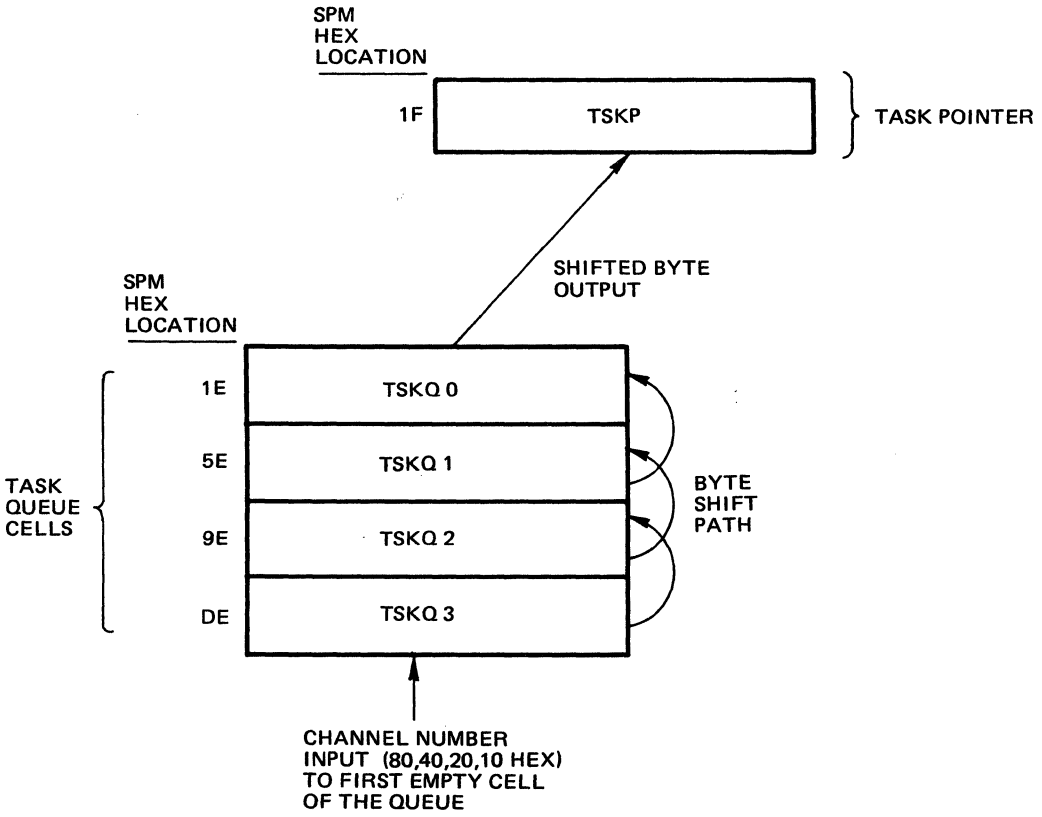


Figure 3-4 Task Queue/Task Pointer Relationship

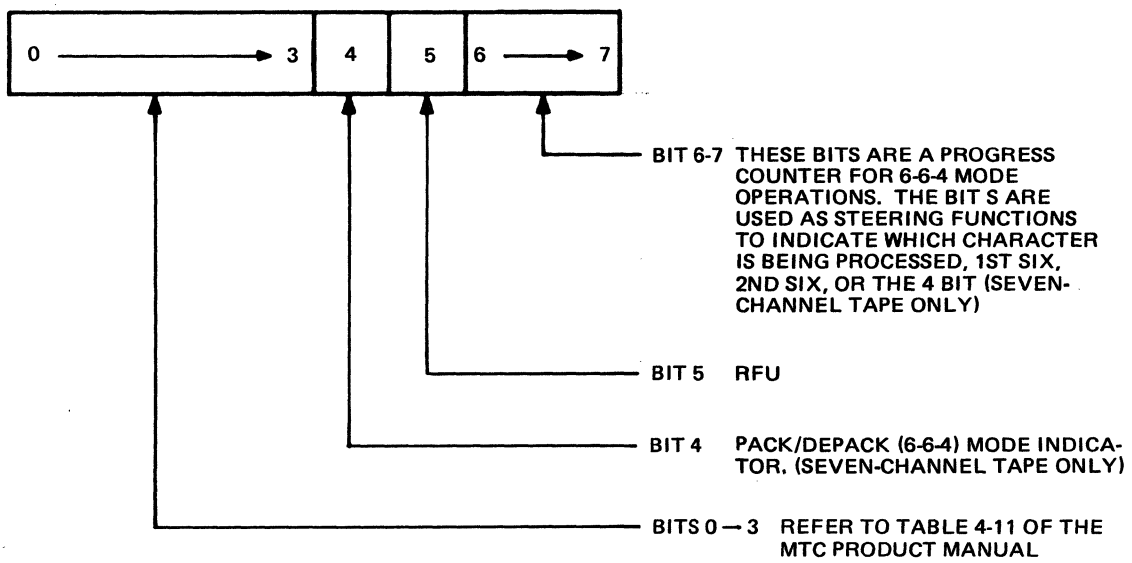


Figure 3-5 Channel Mask Magnetic Tape Application

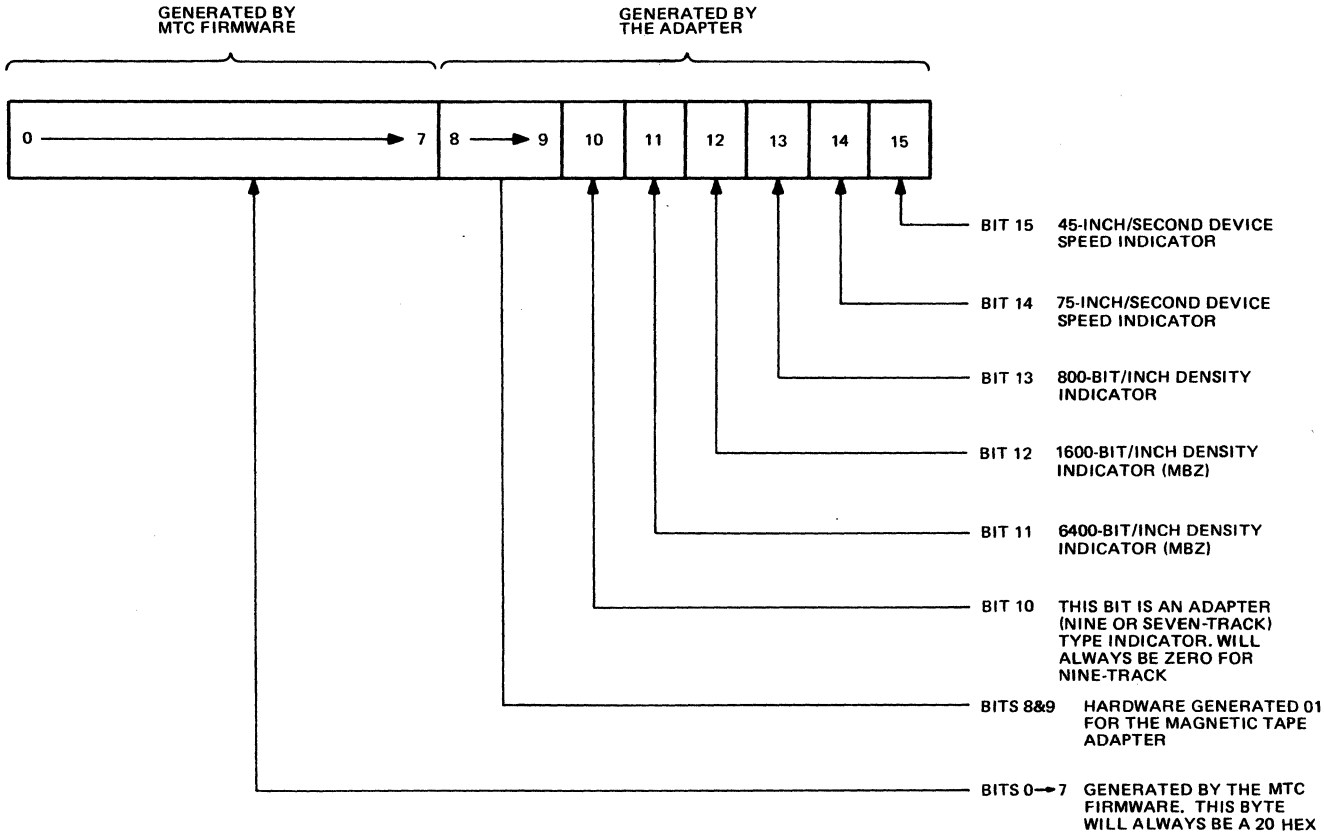


Figure 3-6 Device Identification Word

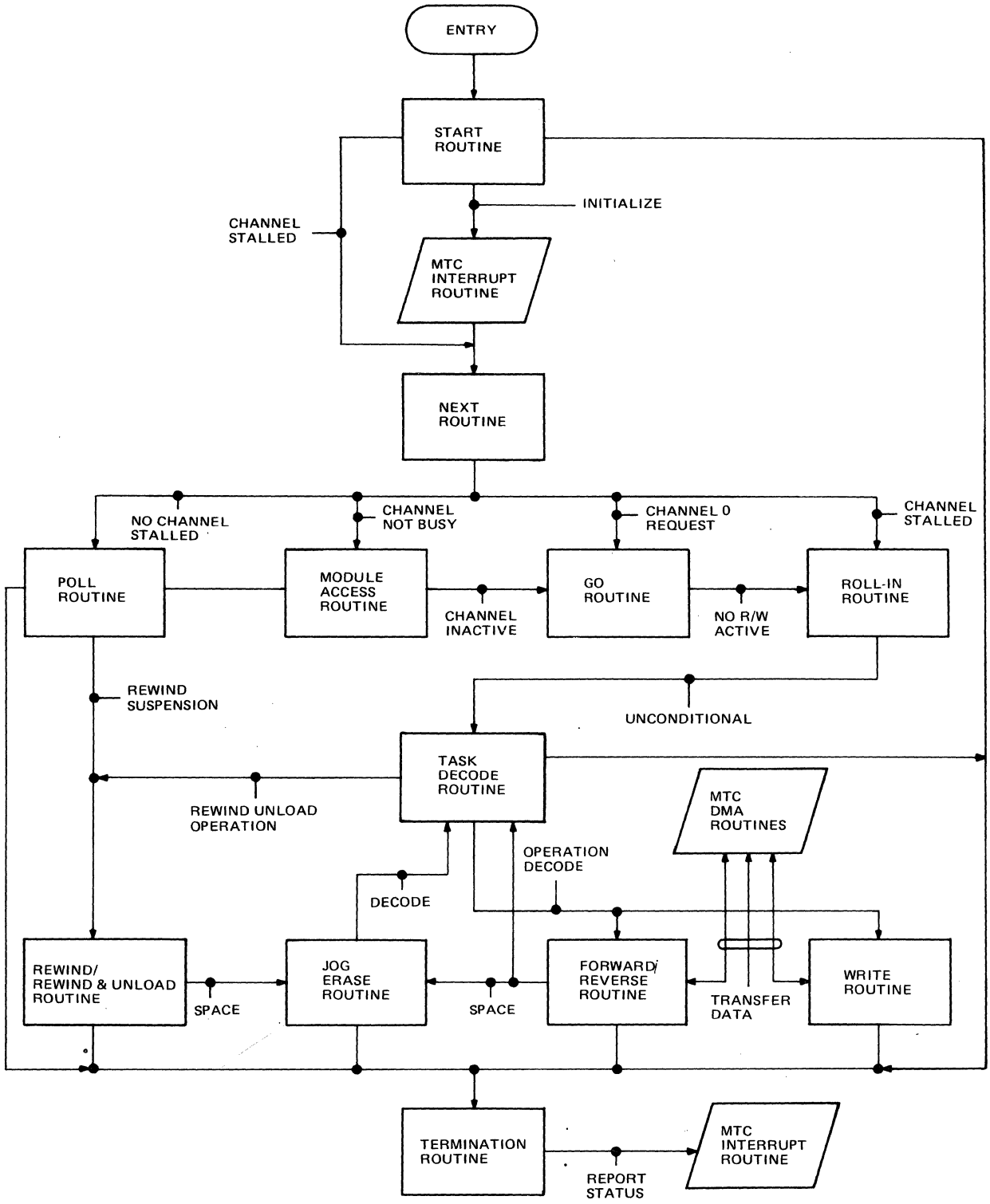


Figure 3-7 Nine-Channel Tape Adapter Overview Flow Chart

PLEASE FOLD AND TAPE –
NOTE: U. S. Postal Service will not deliver stapled forms

M&TO HARDWARE PUBLICATIONS
USER COMMENTS FORM

DOCUMENT TITLE: _____

PART NO.: _____

ORDER NO.: _____

ERRORS:

HOW DO YOU USE THIS DOCUMENT?

THEORY ☐

MAINTENANCE ☐

TROUBLESHOOTING ☐

OTHER: _____

_____

DOES THIS MANUAL SATISFY YOUR REQUIREMENTS?

YES ☐ NO ☐

IF NOT, PLEASE EXPLAIN _____

_____

_____

FROM: NAME _____ DATE _____

TITLE _____

COMPANY _____

ADDRESS _____

FIRST CLASS
Permit No. 39531
Waltham, Ma.

Business Reply Mail NO POSTAGE STAMP NECESSARY IF MAILED IN THE UNITED STATES

POSTAGE WILL BE PAID BY

HONEYWELL INFORMATION SYSTEMS INC.
200 SMITH STREET
WALTHAM, MA. 02154

MAIL STATION 872A
HARDWARE PUBLICATIONS, BILLERICA

Honeywell

Honeywell

Honeywell Information Systems
In the U.S.A.: 200 Smith Street, MS 486, Waltham, Massachusetts 02154
In Canada: 2025 Sheppard Avenue East, Willowdale, Ontario M2J 1W5
In Mexico: Avenida Nuevo Leon 250, Mexico 11, D.F.

FN18, Rev. 1