As you are now the owner of this document which should have come to you for free, please consider making a donation of £1 or more for the upkeep of the (Radar) website which holds this document. I give my time for free, but it costs me money to bring this document to you. You can donate here https://blunham.com/Misc/Texas

Many thanks.

Please do not upload this copyright pdf document to any other website. Breach of copyright may result in a criminal conviction.

This Acrobat document was generated by me, Colin Hinson, from a document held by me. I requested permission to publish this from Texas Instruments (twice) but received no reply. It is presented here (for free) and this pdf version of the document is my copyright in much the same way as a photograph would be. If you believe the document to be under other copyright, please contact me.

The document should have been downloaded from my website https://blunham.com/, or any mirror site named on that site. If you downloaded it from elsewhere, please let me know (particularly if you were charged for it). You can contact me via my Genuki email page: https://www.genuki.org.uk/big/eng/YKS/various?recipient=colin

You may not copy the file for onward transmission of the data nor attempt to make monetary gain by the use of these files. If you want someone else to have a copy of the file, point them at the website. (https://blunham.com/Misc/Texas). **Please do not point them at the file itself as it may move or the site may be updated.**

It should be noted that most of the pages are identifiable as having been processed by me.

I put a lot of time into producing these files which is why you are met with this page when you open the file.

If you find missing pages, pages in the wrong order, anything else wrong with the file or simply want to make a comment, please drop me a line (see above).

It is my hope that you find the file of use to you.

Colin Hinson
In the village of Blunham, Bedfordshire.

TMS 9900
Floppy Disk Controller
B202 (MOSA5)

Price £1.00

A Texas Instruments
Application Report

PRINTED IN U.K.

TEXAS INSTRUMENTS

TABLE OF CONTENTS

TABLE OF CONTENTS (Continued)

LIST OF ILLUSTRATIONS

LIST OF TABLES

SECTION I

INTRODUCTION

This application report describes a TMS 9900 microprocessor system which controls a floppy disk drive and interfaces to an RS-232C type terminal. In addition to providing useful information for the design of a similar system, this application report also shows many of the design considerations for any TMS 9900 microprocessor system design.

The floppy disk is rapidly becoming the most widely accepted bulk storage medium for microprocessor systems. Using standard encoding techniques, a single floppy disk will contain in excess of 400K bytes of unformatted data. Access time to a random record of data is vastly superior to serial media such as cassettes and cartridges, and the medium is both non-volatile and removable.

The use of a microprocessor in the floppy-disk controller or "formatter" is desirable for a number of reasons. The number and cost of components is reduced: this design contains 24 integrated circuits, while random-logic designs typically contain more than 100. The commands from the user interface (in this case, the terminal) to the controller may be more sophisticated, relying on the microprocessor to intrepret the commands. The microprocessor also enables the controller to perform diagnostic functions, both on the controller itself and on its associated drives, not available with a random-logic system.

The Texas Instruments TMS 9900 microprocessor is particularly well-suited to this application. The TMS 9900 is a 16-bit microprocessor capable of performing operations on single bits, bytes, and words. The CRU provides an economical port for bit-oriented input/output, while the parallel memory bus is available for high-speed data. The speed of operation of the TMS 9900 minimizes additional hardware requirements. The powerful memory-to-memory instruction set and large number of available registers simplify software, both in terms of number of assembly language statements and total program memory requirements.

SECTION II

SYSTEM DESCRIPTION

Figure 1 illustrates the relationship of the system elements. Commands are entered by the user at the terminal. These commands are serially transmitted to the controller. The controller interprets the commands and performs the operations specified, such as stepping the read/write head of the drive to a particular track, and reading or writing selected data.

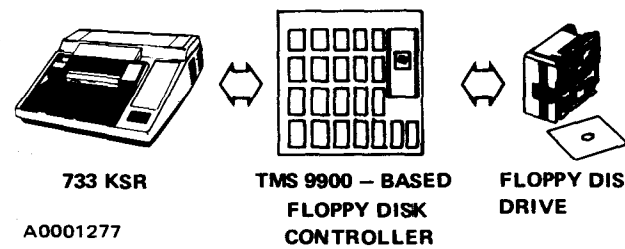


Figure 1. TMS 9900 Floppy Disk Controller System

2.1 DATA TERMINAL

The terminal used in this design is the Texas Instruments 733 KSR Silent Electronic Data Terminal (see Figure 2). Slight modifications to the software will allow the use of virtually any RS-232 terminal.

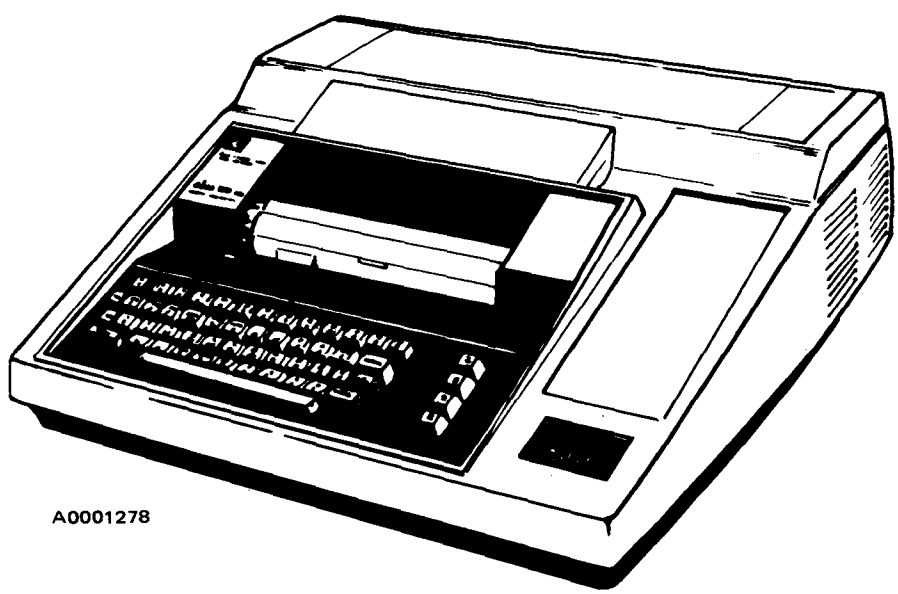


Figure 2. TI 733 KSR Terminal

The 733 KSR consists of a keyboard, printer, and a serial-communication line to the controller. The keyboard enables the operator to enter control commands and data for storage on floppy disc. The printer is used for echoing operator entries, data printout, and reporting of operational errors. The serial interface is full duplex, allowing data transmission both to and from the data terminal simultaneously.

Characters entered on the keyboard are transmitted to the controller in 7-bit ASCII code using asynchronous format, and characters to be printed are sent from the controller to the terminal in the same way. Transmission speed is 300 baud. The format for data transmission is shown in Figure 3.

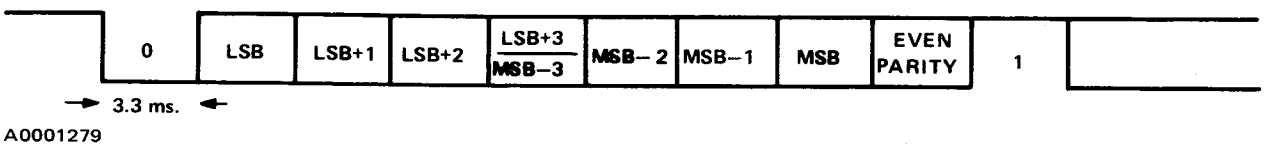


Figure 3. Data Transmission Format

The line idle condition is represented by a logic one. When a character is to be transmitted, the ASCII character is preceded by a zero bit, followed by the 7-bit ASCII code, even parity bit, and the logic-one stop bit. Any amount of idle time may separate consecutive characters by maintaining the logic-one level. Reading data is accomplished by continuously monitoring the line for the one-to-zero transition at the beginning of the start bit. After delaying one-half bit time (1.67 ms) the line is again sampled to ensure that the start bit is valid. If so, the line is sampled each bit time (3.33 ms) until all of the bits of the character have been sampled. The initial one-half bit delay causes subsequent samples to be taken at the theoretical center of each bit, thus providing a margin for distortion due to time base differences between the transmitter and receiver.

The control signals for the terminal are shown in Figure 4.

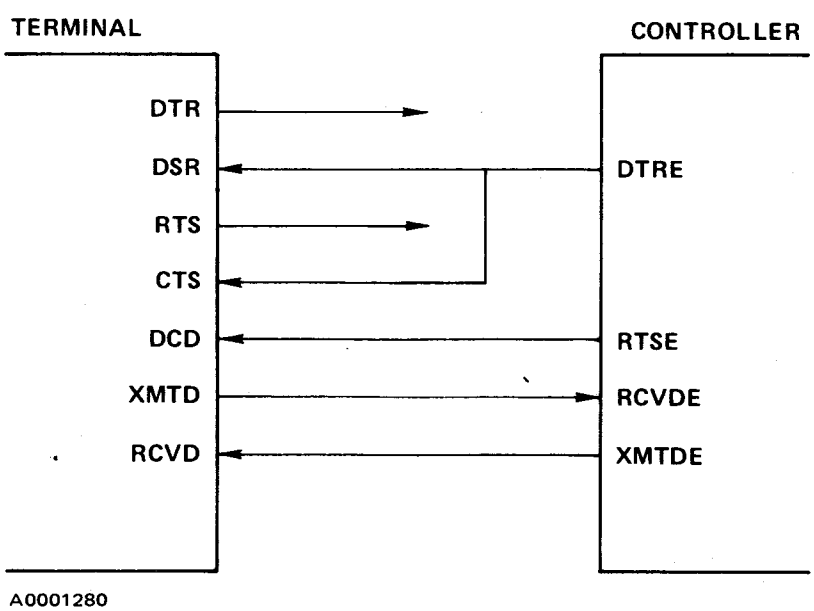


Figure 4. Terminal Interface

Detailed description of the signals is provided in *Electronics Industries Association Standard RS-232C.* The signals used in this design are briefly described below.

DTRE – Data Terminal Ready is always on when power is applied to the controller, enabling operation of the serial interface by the terminal.

RTSE – Request to Send is on when a character is transmitted from the controller to the terminal.

XMTDE – Transmitted Data from the controller to the terminal.

RCVDE – Received Data from the terminal to the controller.

Signal levels conform to EIA Standard RS-232C, as shown in Table 1.

Table 1. RS-232C Signal Levels

Voltage Level	Data (XMTDE,RCVDE)	Control (DTRE,RTSE)
–25 to –3 VDC	1	OFF
+3 to +25 VDC	0	ON

The other important parameter for interfacing to the terminal is the amount of time required for a carriage return by the printer, which is 200 ms maximum for the 733 KSR.

2.2 FLOPPY-DISK DRIVE

The floppy-disk drive (Figure 5) is the electromechanical unit in which the recording medium, the floppy disk is inserted. The drive contains the electronics which control the rotation of the floppy disk, the reading and writing of data, and the positioning of the read/write head to select a particular track on the diskette.

2.2.1 Floppy Disk

The floppy disk, or diskette, is the recording medium (see Figure 6). It is enclosed in a plastic protective envelope which keeps foreign particles away from the recording surface. The inner material of the envelope is specially treated to minimize friction and static electricity discharge. The read/write head opening enables the head to come in contact with the recording surface. The index-access hole enables detection of the index hole.

When the index hole in the diskette becomes aligned with the index-access hole, an optical sensor generates the index pulse, providing a reference point for the beginning of each track. There are 77 concentric tracks for recording data. A particular track is accessed by moving the read/write head radially until the desired track is located.

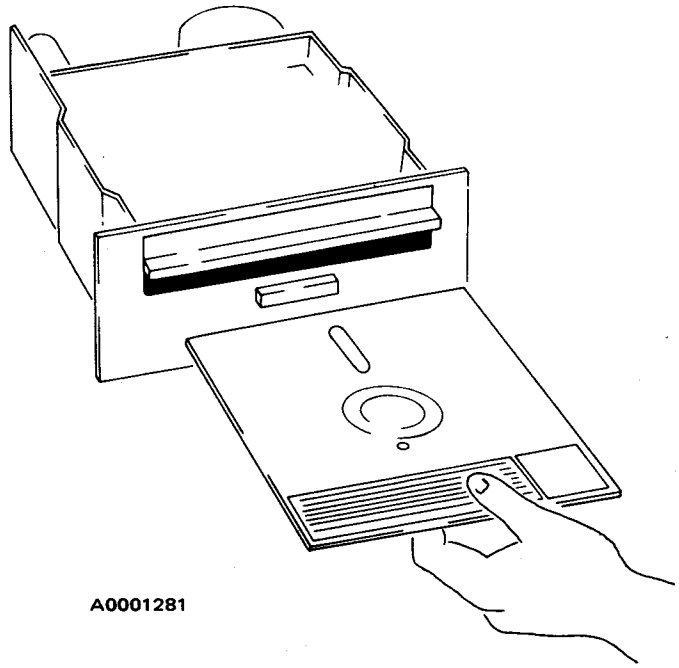


Figure 5. Floppy Disk Drive

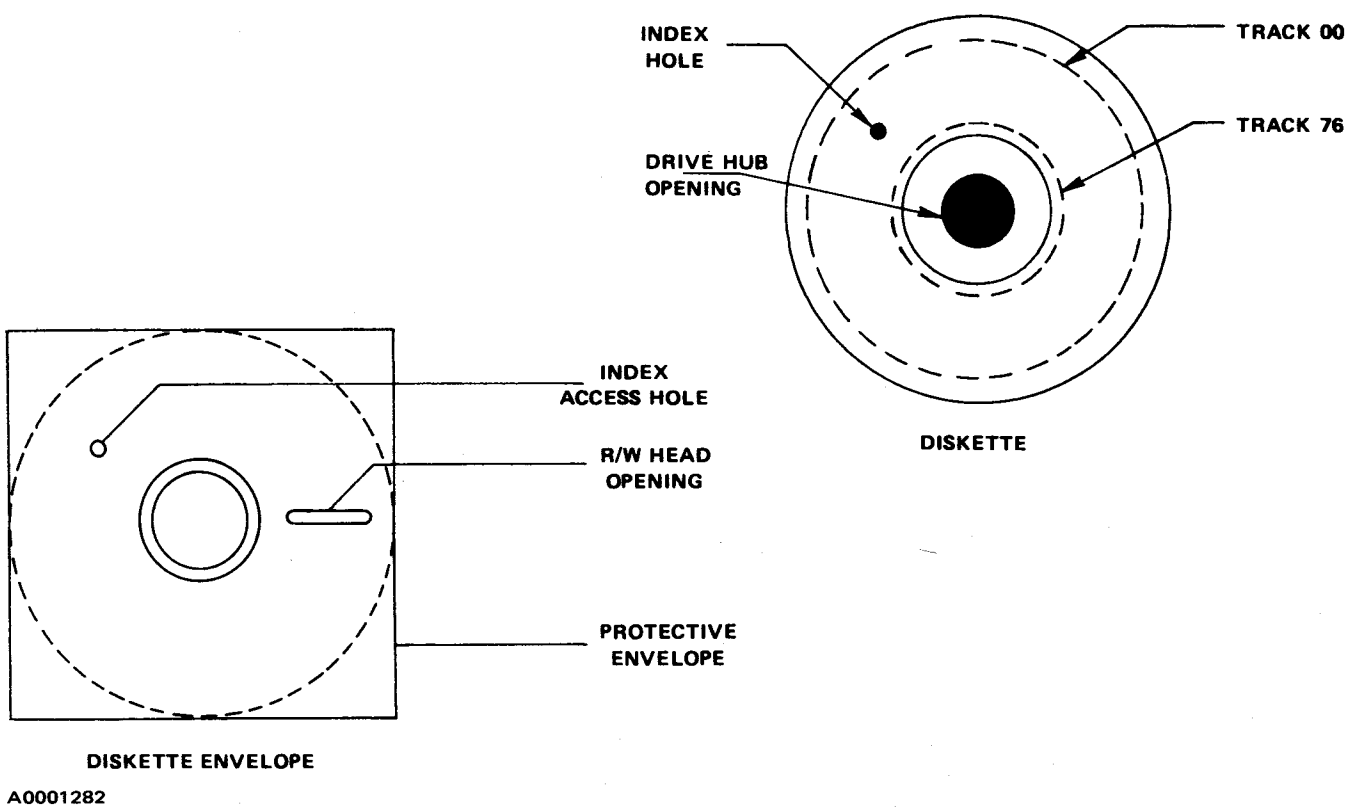


Figure 6. Diskette Envelope and Diskette

2.2.2 Physical Data Structure

The 77 tracks on a diskette are numbered from 00 (outermost) to 76 (innermost). Each track is subdivided into 26 sectors, or records, numbered sequentially from 1 to 26. Each sector consists of two fields: the ID field, which contains sector identification (track and sector number) and the data field, which contains 128 bytes of data.

2.2.3 Encoding Technique

The encoding technique used for representation of data on the diskette is a form of frequency modulation (FM), as shown in Figure 7. Each bit period is 4 microseconds long, resulting in a data-transfer rate of 250K bits per second. A pulse occurs at the beginning of each normal bit period. This pulse is called the clock pulse. If the data bit is a one, a pulse will occur also in the middle of the bit period, 2 μs after the clock bit. If the data bit is a zero, no pulse occurs in the middle of the bit period.

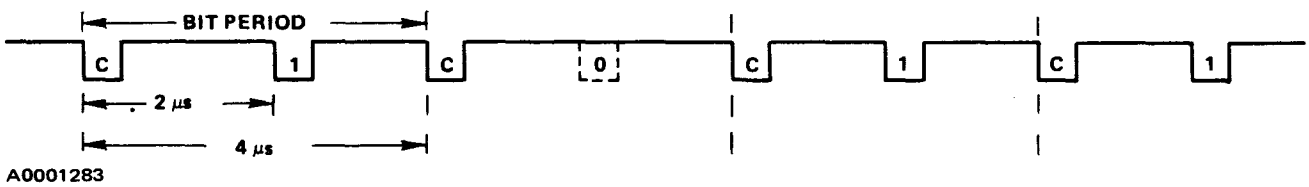


Figure 7. FM Data Pattern 1011

Selected clock bits are deleted in special characters called marks. The absence of the clock bits results in unique sequences, used for synchronization at the beginning of fields.

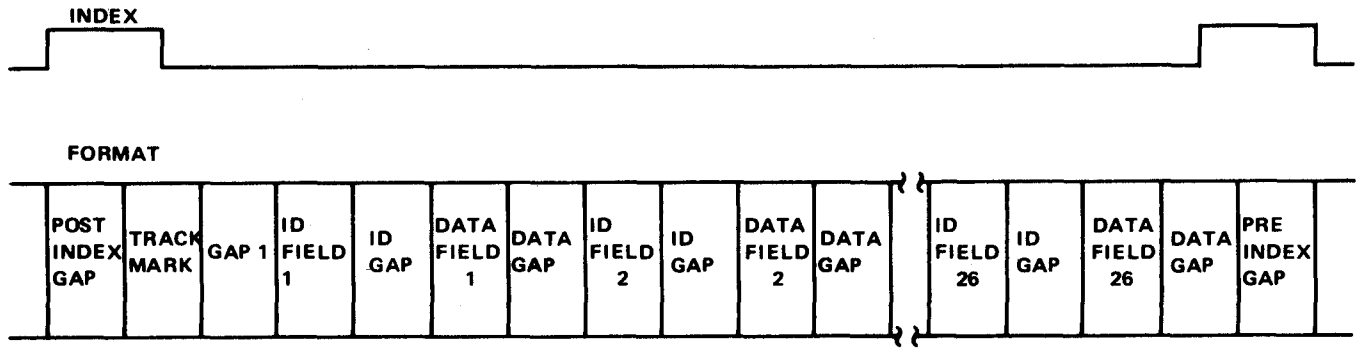
2.2.4 Track Format

Each track is formatted to provide 26 "soft" sectors. The term soft sectoring means that the beginning of each sector is encoded on the medium through a unique bit sequence. Each of the sectors is separated by a gap of dummy data. Each of the two fields (ID and data) in each sector are also separated by a gap. The first byte of each field is a mark in which the clock pattern for the byte is $C7_{16}$ rather than FF_{16}. The organization of data and clock bits on each track is shown in Figure 8.

2.2.5 Cyclic Redundancy Check Character

The last two bytes at the end of each ID and data field comprise the 16-bit cyclic redundancy check character (CRC). The CRC is generated by performing modulo-2 division on the data portion of the entire field (including the mark) by the polynomial $X^{16} + X^{12} + X^5 + 1$. Before generation of the CRC begins, the initial value is $FFFF_{16}$.

The analogous hardware operation is illustrated in Figure 9. All flip-flops are initially set to one. Each data bit in the field, beginning with the MSB of the mark byte, is shifted into the logic at DATAIN. The previous



POST INDEX GAP – 46 BYTES, DATA = 00, CLOCK = FF_{16}

TRACK MARK – 1 BYTE, DATA = FC_{16}, CLOCK = $D7_{16}$

GAP 1 – 32 BYTES, DATA = 00, CLOCK = FF_{16}

ID FIELD – 7 BYTES:

	1	2	3	4	5	6	7
CLOCK	$C7_{16}$	FF_{16}	FF_{16}	FF_{16}	FF_{16}	FF_{16}	FF_{16}
DATA	FE_{16}	TRACK NUMBER	00	SECTOR NUMBER	00	CRC1	CRC2

ID GAP – 17 BYTES, DATA = 00, CLOCK = FF_{16}

DATA FIELD – 131 BYTES:

	1	2-129	130	131
CLOCK	$C7_{16}$	FF_{16}	FF_{16}	FF_{16}
DATA	FB_{16} OR $F8_{16}$	DATA	CRC1	CRC2

DATA GAP – 33 BYTES, DATA = 00, CLOCK = FF_{16}

PRE INDEX GAP – 241 BYTES, DATA = 00, CLOCK = FF_{16}

A0001284

Figure 8. Track Recording Format

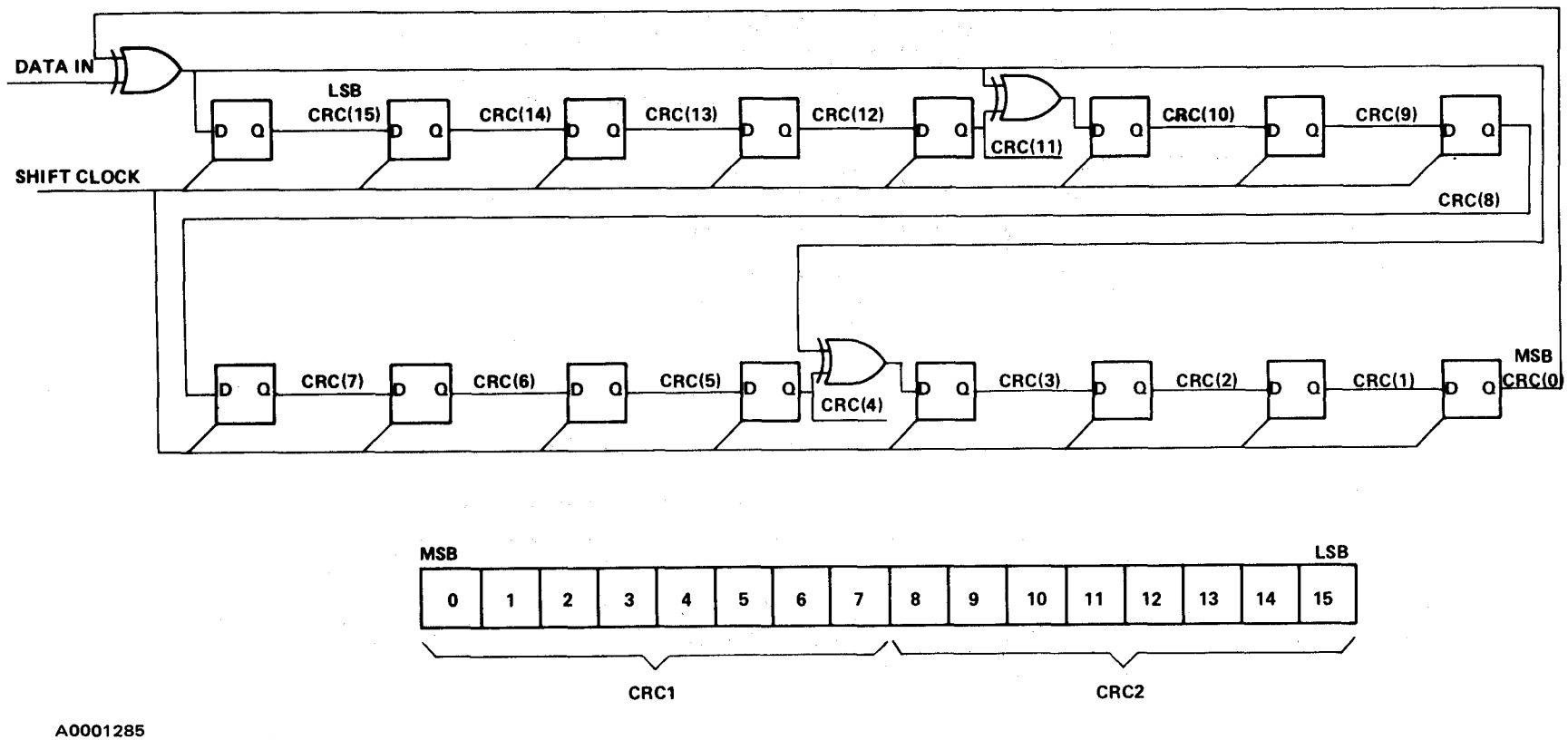


Figure 9. Hardware CRC Generation

MSB is exclusive ORed with the new input bit to generate a feedback term. This feedback term is stored in the LSB of the register, and is also exclusive ORed with other terms of the CRC. After all data bits of the field have been shifted in, the value in the register is the CRC. The most-significant byte is CRC1 and the least-significant byte is CRC2.

When reading the field, the identical operation is performed, presetting all flip-flops and shifting in all data bits. When reading, it is convenient to also shift in the CRC, causing the resultant value in the register to finally become all zeroes.

In this design, the CRC is calculated by software; however, the algorithm is identical.

2.2.6 Reading Data

The procedure for reading diskette data is as follows:

1. Search the serial-bit string for the ID mark (clock = $C7_{16}$, data = FE_{16}).
2. Read the next four bytes to determine if the desired sector has been located. If not, return to 1.
3. Read the CRC for the ID field and compare it to the expected value. If incorrect, report error and/or return to 1.
4. Search the serial-bit string for either the data mark (clock = $C7_{16}$, data = FB_{16}) or the deleted-data mark (clock = $C7_{16}$, data = $F8_{16}$).
5. Read the next 128 bytes and save.
6. Read the CRC for the data field and compare it to the expected value. If incorrect, report error and/or return to 1.

Normally, if the process is not completed before two index pulses are detected, indicating a complete diskette revolution, the try has failed. Either a retry will be performed, or an error is reported.

2.2.7 Writing Data

When writing data, the sector is located as in steps 1 through 3 above. Then, the ID gap, the data field complete with CRC, and a pad byte (data = 0, clock = FF_{16}) are written.

2.2.8 Track Formatting

The formatting process consists of writing all of the gaps, track mark, ID fields, and data fields, putting dummy data into the data bytes of the data field. After a track is formatted, only the ID gap, data field,

and the first byte of the data gap are altered when updating sectors. The number of bytes in the pre-index gap will possibly vary slightly, due to variations in the speed of revolution of the diskette.

2.2.9 Floppy-Disk Timing

Several important timing parameters pertain to the operation of the disk drive:

Bit transfer rate	250,000 bits/second
Track-to-track stepping time	10 milliseconds
Settling time (before read/write)	10 milliseconds
Rotational speed	360 RPM ±2%
Head load time (before read/write)	35 milliseconds

Thus, data is transferred at a rate of 250K bits/second, or 31.25K bytes/second ±2%. Stepping the head each track position requires 10 ms. An additional 10 ms delay must be observed after the final step before reliable data may be written or read. A delay of 35 ms must occur after the head is loaded ($\overline{RDY}$ = 0) before reliable data may be written or read.

SECTION III

HARDWARE DESCRIPTION

A complete logic diagram of the system is contained in the center of this report. The operation of each section is described separately.

3.1 CLOCK GENERATION AND RESET

The TIM 9904 is used to generate the 4-phase MOS clocks for the TMS 9900 (see Figure 10). Ten ohm resistors are connected in series to the clock lines for damping. The TIM 9904 should always be located physically close to the TMS 9900 to minimize the length of the conductor run for the MOS clocks. The $\overline{\phi 3}$ TTL-level output is used in the synchronous disk read/write control logic.

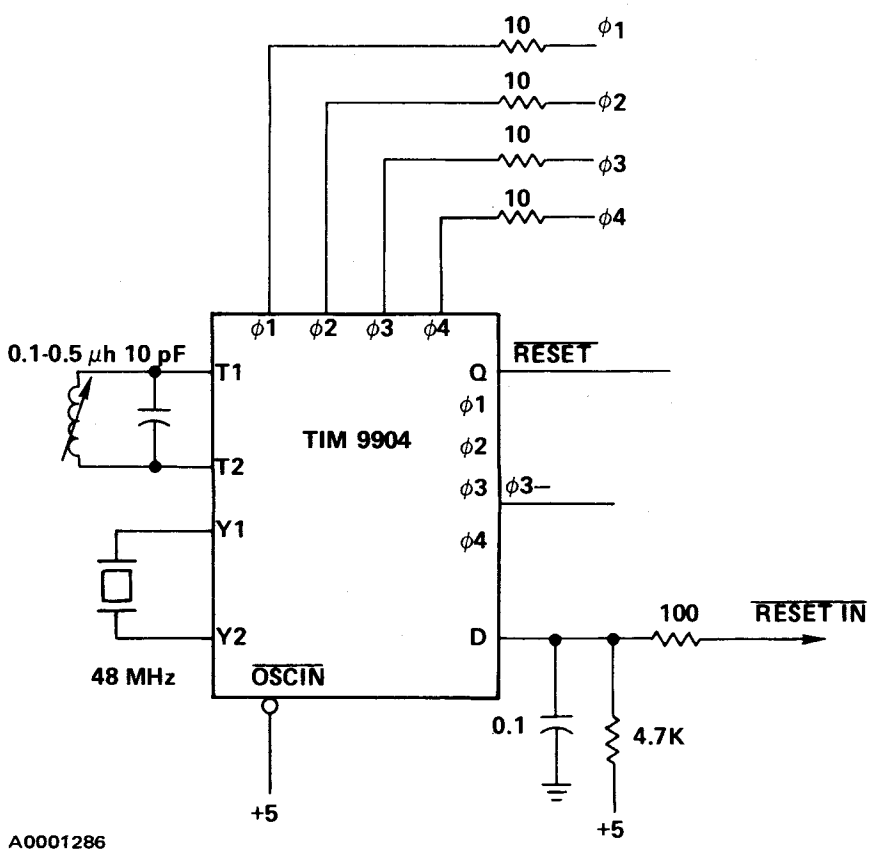


Figure 10. Clock Generation and Reset

A 48 MHz, third overtone crystal causes the clock frequency to be 3 MHz. The inductor of the LC tank circuit need not be variable; however, in wire-wrap prototypes the capacitance due to interconnect is difficult to predict. The $\overline{\text{OSCIN}}$ input is held high to disable the external clock input.

The RC input to the Schmitt-D input provides power-on detection. The $\overline{\text{RESETIN}}$ input is connected to an external pushbutton. The 100 ohm series resistor reduces contact arcing, thereby extending switch life.

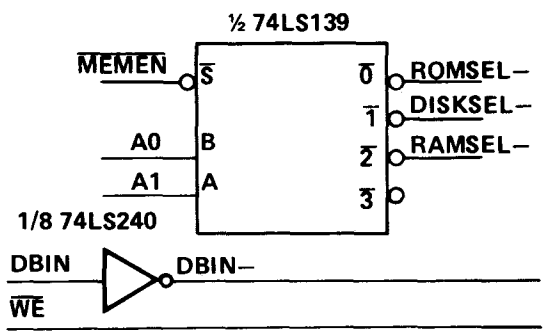
3.2 CPU

The TMS 9900 requires a minimum of external logic. Note that both the data and address buses are connected directly to the memory and disk read/write control logic without buffering as shown in Figure 11. This is due to the ability of the TMS 9900 outputs to sink up to 3.2 mA with 200 pF capacitive load.

The READY input is used to synchronize data transfers to and from the disk read/write control logic, eliminating the need for buffer registers. The $\overline{\text{HOLD}}$, $\overline{\text{LOAD}}$, and interrupt functions are not used in this design and are tied to their inactive (high) level.

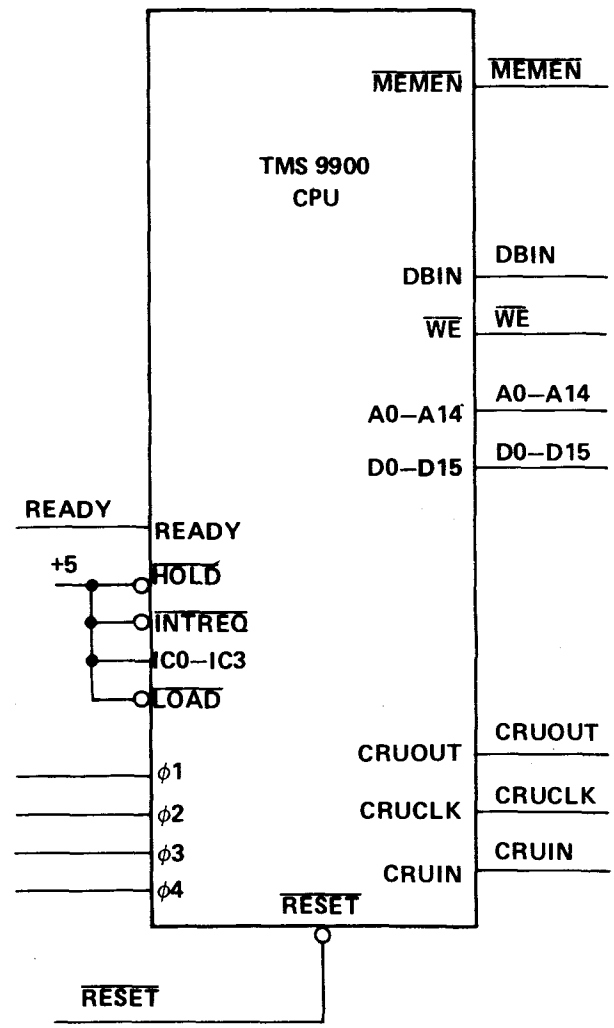
3.3 MEMORY CONTROL

Memory control logic, shown in Figure 12, consists of a simple decode of the high-order address lines, enabled by $\overline{\text{MEMEN}}$. Memory enabling signals are generated for EPROM (ROMSEL–), RAM (RAMSEL–), and the disk interface (DISKSEL–). Table 2 shows the memory address assignments.



A0001287

Figure 12. Memory Control



A0001288

Figure 11. TMS 9900 CPU

Table 2. Memory Address Assignments

Signal	A0	A1	Address Space	Function	Actually Used
ROMSEL–	0	0	000-3FFF	EPROM	000-07FF
DISKSEL–	0	1	4000-7FFF	Disk	7F8E-7FFE
RAMSEL–	1	0	8000-BFFF	RAM	8000-81FF
	1	1	C000-FFFF	Not Used	

Each of the enabling signals will be active when a memory cycle is being performed ($\overline{MEMEN} = 0$) accessing its address space.

3.4 DISK READ/WRITE SELECT

The DISKSEL signal is further decoded to generate separate select lines for disk read (DISKRD–) and disk write (DISKWT–) operations.

$$DISKRD- = \overline{(DISKSEL)\,(DBIN)\,(A14-)}, \text{ and}$$
$$DISKWT- = \overline{(DISKSEL)\,(DBIN-)\,(A14)}.$$

Disk read and write operations are specified by different addresses, and are selected only when the DBIN signal is at the proper level for the direction of transfer (see Figure 13). This is required because of the sequence of machine cycles performed by the TMS 9900 when performing a memory-write operation. In the MOV instruction, the CPU first fetches the contents of the memory location to be altered, then replaces this value with the source operand. In this design, the disk read and write operations are controlled by the READY line to synchronize data transfers. If read and write signals were not generated separately, there would be ambiguity with respect to the type of operation desired.

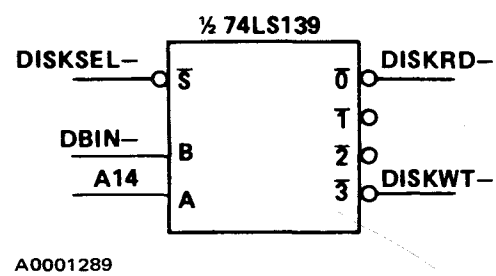


Figure 13. Disk Read/Write Select

This applies to all memory-mapped interfaces in TMS 9900 systems, i.e., the MOV instruction will cause a read operation to precede the write operation to the specified destination address.

3.5 STORAGE MEMORY

Storage memory, shown in Figure 14, is used for implementing workspace registers, maintenance of software pointers and counters, and buffering of a full sector of data.

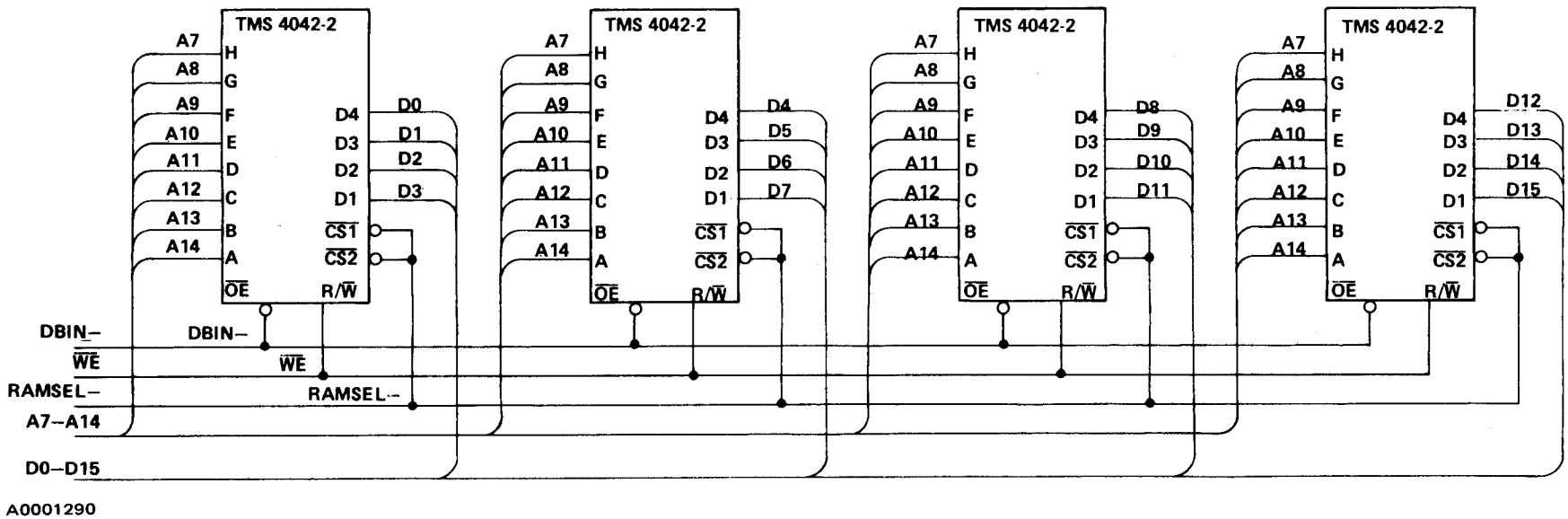


Figure 14. Storage Memory

This design utilizes four TMS 4042-2 RAMs, resulting in a 256-word array of RAM for temporary storage. This 256-word array may be addressed at locations 8000-BFFF, causing each memory location to be multiply defined (e.g., memory address 8000 selects the same word as memory address 8200). For simplificity, RAM will be referred to only as locations 8000-81FF.

Access times for the TMS 4042-2 are sufficiently fast to allow the TMS 9900 to access RAM without any wait states, thus READY will always be true when RAM is addressed. The output enable ($\overline{OE}$) inputs require that the DBIN output from the TMS 9900 be inverted to gate RAM onto the data bus. The $\overline{WE}$ output from the TMS 9900 is directly compatible with the R/$\overline{W}$ input. Data and address lines are connected directly to the CPU.

3.6 PROGRAM MEMORY

Program memory (Figure 15) is used for storage of the machine code program to be executed by the TMS 9900. Also, constants, the RESET vector and XOP vectors are contained in this space.

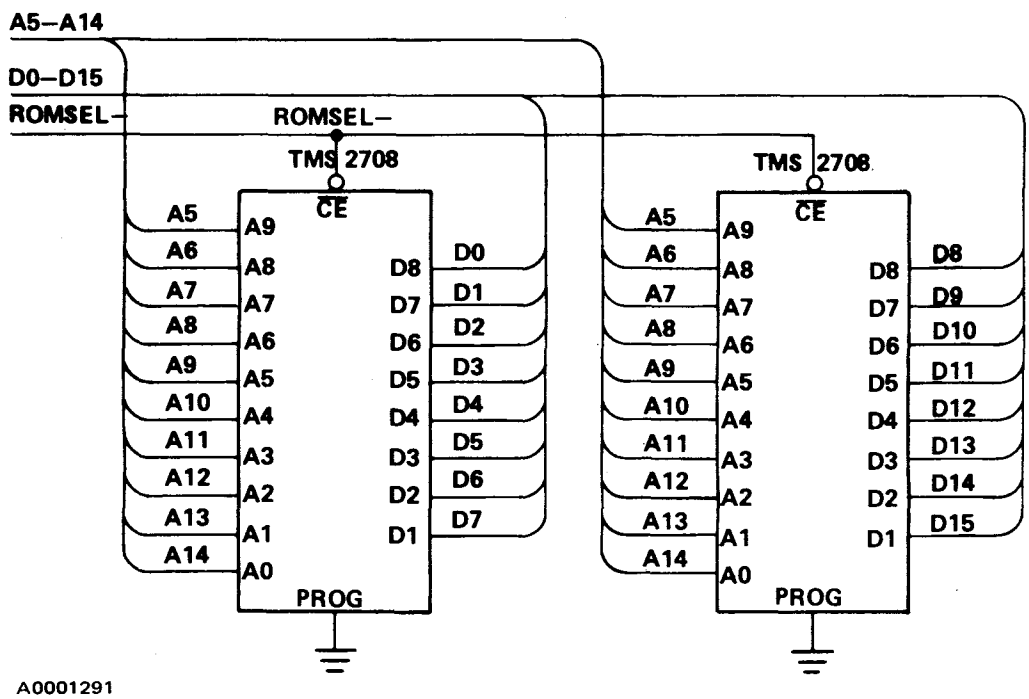


Figure 15. Program Memory

Two TMS 2708 erasable programmable read-only memories (EPROMs) comprise the program memory for this design, resulting in 1024 words of EPROM. EPROM is addressed at memory locations 0000-3FFF. Since these addresses are multiply defined, EPROM will be described only as memory addresses 0000-07FF. Access times for the TMS 2708 are such that no wait states are required.

3.7 CONTROL I/O

All of the control and status signals which require individual testing, setting, or resetting are implemented on the CRU, the bit addressable I/O port for the TMS 9900.

The benefits of using the CRU for these functions is twofold. First, eight bits of input and eight bits of output can be implemented with two 16-pin devices, which are substantially smaller and lower in cost than if these functions were implemented on the parallel-data bus. The second benefit is increased software efficiency. Control and status testing operations can be performed with single one-word instructions, rather than the ORing, ANDing, and maintenance of software images necessary when performing single-bit I/O on the memory bus.

Eight bits of output are implemented with the TIM 9906 8-bit addressable latch. The CRUCLK line must be inverted for input to the TIM 9906. The eight input bits are implemented using the TIM 9905 8-to-1 multiplexer. Individual I/O bits are selected using the three least-significant address lines, A12–A14. The control I/O is illustrated in Figure 16.

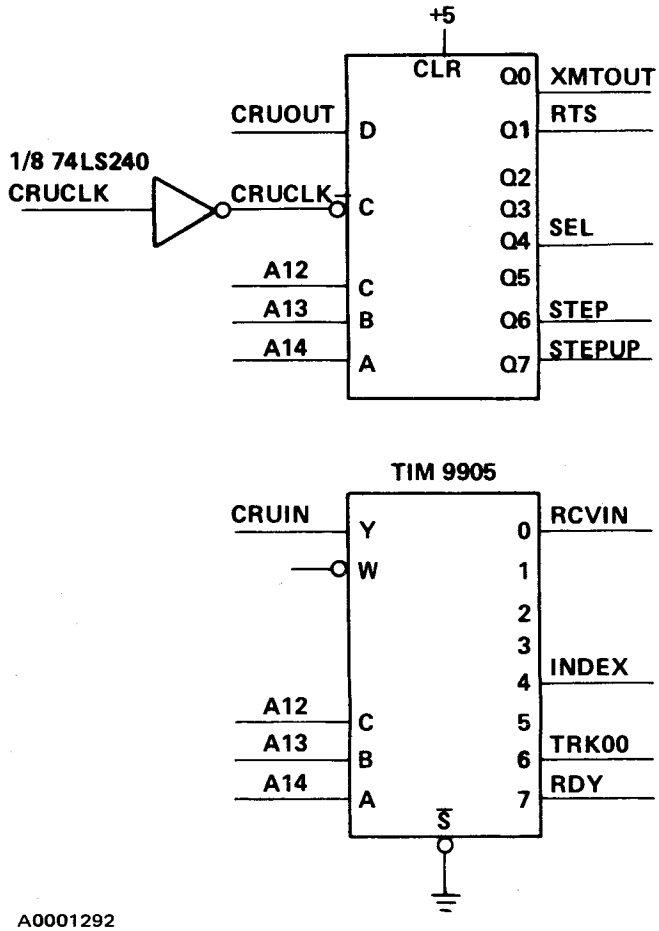


Figure 16. Control I/O

3.8 FLOPPY-DISK-DRIVE INTERFACE

All outputs to the drive are 7406 open-collector, high-voltage and current drivers. Pullups for the output signals are provided in the drive electronics. All inputs are terminated by 150 ohm pullup resistors to +5 volts, and are buffered and inverted. All input and output signals are active low.

$\overline{\text{SEL}}$ – Active when a stepping operation or a data transfer is being performed.

$\overline{\text{RDY}}$ – Active when the disk is ready to perform a stepping or transfer operation (i.e., $\overline{\text{SEL}}$ = 0, diskette is in place, door is closed, power is furnished to the drive).

$\overline{\text{STEP}}$ – A minimum 10 μs pulse causes the read/write head to move one track position in the direction selected by $\overline{\text{STEPUP}}$.

$\overline{\text{STEPUP}}$ – When $\overline{\text{STEPUP}}$ = 0, the read/write head moves in one track position. When STEPUP = 1, the head will move out (toward track 00).

$\overline{\text{TRK00}}$ – Active when the read/write head is located on the outermost track (track 00).

$\overline{\text{INDEX}}$ – As the diskette rotates in the drive, the index pulse occurs once per revolution, providing a reference point for the beginning of each track.

$\overline{\text{WRITE ENABLE}}$ – This signal must be active a minimum of 4 μs before a write operation begins, and must be maintained active during the entire write operation.

$\overline{\text{WRITE DATA}}$ – This signal contains a series of pulses representing the data to be written to the disk in the FM format previously described.

$\overline{\text{READ DATA}}$ – This signal contains a series of pulses representing the data to be read from the disk in the FM format previously described.

Figure 17 illustrates the floppy-disk-drive interface.

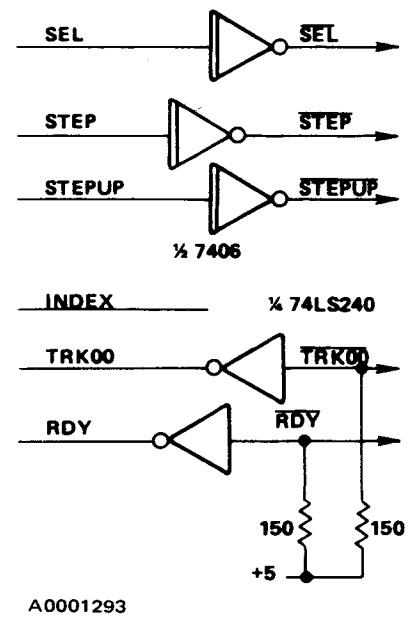


Figure 17. Floppy-Disk Drive Interface

3.9 INDEX PULSE SYNCHRONIZATION

Since the index pulse is a term in some of the expressions that are sampled by the CPU, it must be synchornous to the CPU. The circuit shown in Figure 18 generates a signal one $\phi 3$ clock cycle long at the beginning of each index pulse from the drive. RDY will be inactive when the drive is turned off or the door is open, thus connection of RDY to the preset input of the flip-flop shown causes INDSYN to be active as long as RDY = 0 (see Figure 19). Forcing INDSYN to be one when RDY = 0 prevents the CPU from remaining in a wait state when the drive is disabled during data transfer.

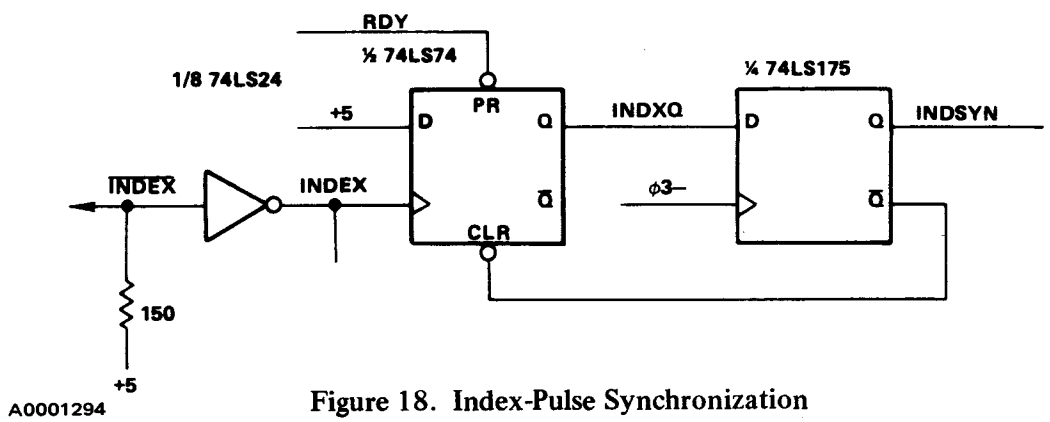


Figure 18. Index-Pulse Synchronization

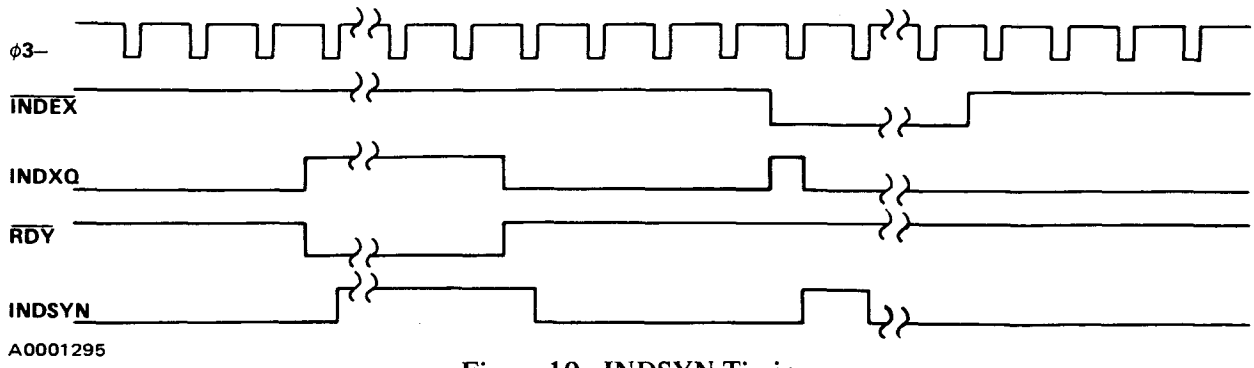


Figure 19. INDSYN Timing

3.10 READ PULSE SYNCHRONIZATION

The read-pulse synchronization logic, Figure 20, generates an active signal, BITIN, one clock cycle long each time a read pulse is detected during read operations. During write operations BITIN is maintained at a logic-one level.

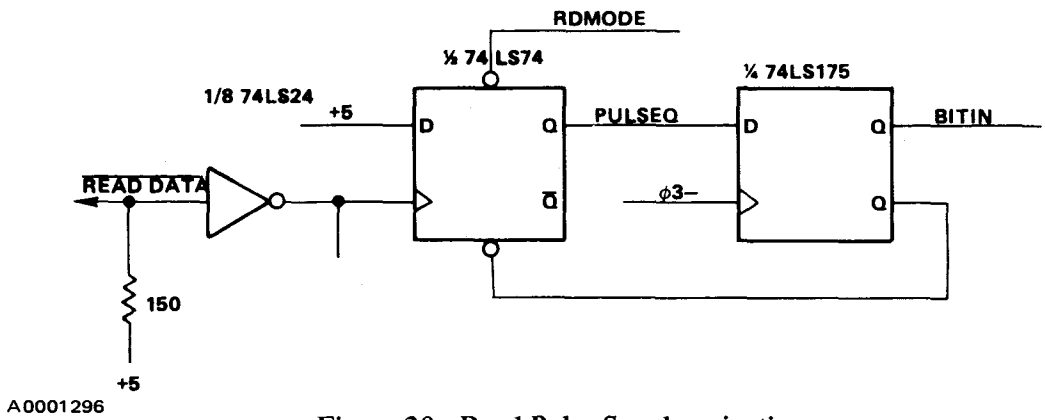


Figure 20. Read-Pulse Synchronization

3.11 BIT DETECTOR

The bit detector, Figure 21, consists of a 74LS163 counter and random logic contained in PROM. During write operations, the counter is used to time the 2 μs spacing between clock bits and data bits. During read operations the bit detector is used to determine the time interval between successive read pulses. The key signal generated by the bit detector is BITTIME, which is active for one clock cycle every 2 μs during disk writing, and which is active each time a one or zero bit is detected during read operations.

3.12 BIT COUNTER

The bit counter, Figure 22, is a 74LS163 used to count the number of bits currently read or written during disk-data transfers. Each time a clock or data bit is detected or written (BITTIME = 1) the bit counter is

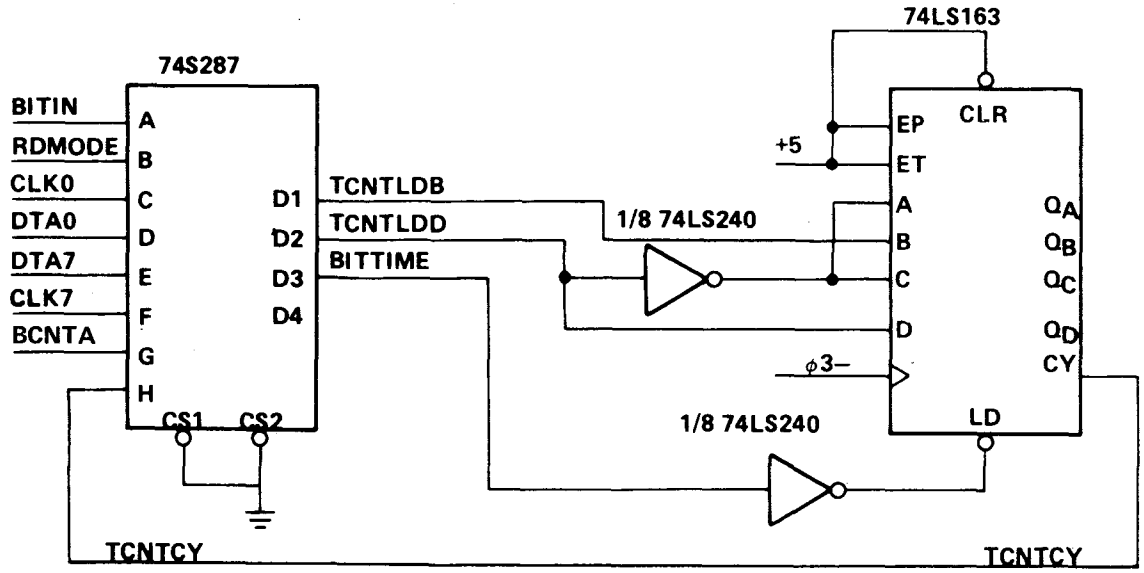


Figure 21. Bit Detector

incremented. The two key outputs are BCNTA and BCNT = 15. BCNTA is the least-significant bit of the counter and is used to alternately select clock (BCNTA = 0) and data (BCNTA = 1) bits as the counter increments. BCNTA = 15 is active when a complete byte has been read or written. This signal establishes byte boundaries for the data and is used to synchronize the parallel data from the CPU to the serial-bit string and from the disk.

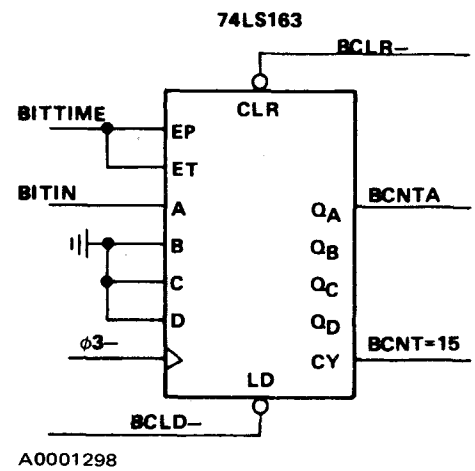


Figure 22. Bit Counter

3.13 WRITE CONTROL AND DATA

Writing to the diskette is controlled by $\overline{\text{WRITE ENABLE}}$, which is the inverted and buffered WTMODE signal. WTMODE is active when a write operation has been initiated by the CPU. The $\overline{\text{WRITE DATA}}$ signal is a series of negative pulses representing FM data to be recorded on the diskette. Figure 23 illustrates write control and data.

3.14 DATA SHIFT REGISTER

The data shift register, see Figure 24, is used for accumulation of data bits during read operations and storage of data bits to be shifted out during write operations. Data is transferred to and from the CPU via the eight most-significant data lines (D0–D7). The data shift register is device type 74LS299.

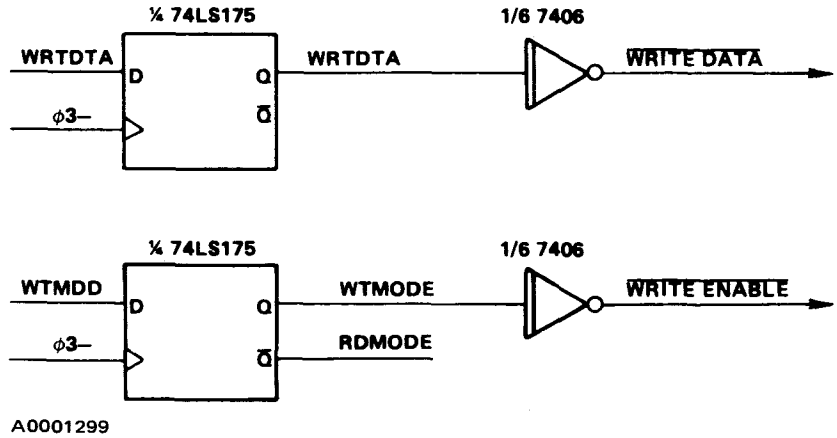


Figure 23. Write Control and Data

3.15 CLOCK SHIFT REGISTER

The clock shift register, Figure 25, is used for accumulation of clock bits during read operations and storage of clock bits to be shifted out during write operations. The clock shift register is device type 74198, which has separate parallel inputs and outputs. Three address lines, A9–A11, are connected to the parallel inputs. As data is loaded into the data shift register during write operations, these three address lines select the clock pattern for that byte (i.e., C7 for ID and data marks, D7 for track mark, FF for normal data). The parallel outputs (CLK0-CLK7) are used to detect mark clock patterns during read operations.

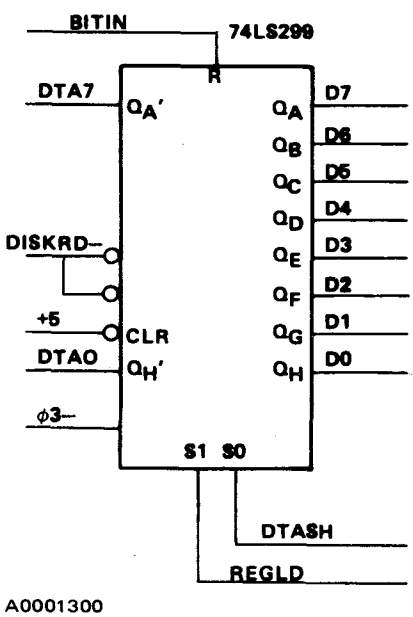


Figure 24. Data Shift Register

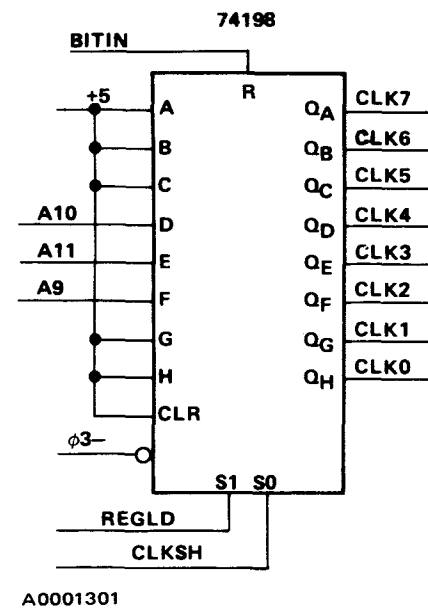


Figure 25. Clock Shift Register

SECTION IV

DISKETTE DATA TRANSFER

The previous section described the various functional blocks in the TMS 9900 floppy-disk controller. However, detailed information was not provided with respect to the logical relationships and timing of the control signal in the read/write control logic.

Most of the read/write control logic varies in function depending on the direction of transfer. This section will describe the operation of the logic separately for read and write operations. After both operations have been completely described, the combined operation will be explained.

4.1 DISK-WRITE OPERATIONS

Disk writing is initiated by executing an instruction which writes data to the data shift register (i.e., when DISKWT– = 0). When this transfer occurs, READY is held low until a byte boundary occurs (BCNT = 15), then READY becomes active, permitting completion of the write cycle. In this way, the data transfers are synchronized to the serial bit string.

To complete the transfer, READY must be active to the CPU, and the CLKSH, DTASH, and REGLD signals to the clock and data shift registers must be active to permit loading. READY = CLKSH = DTASH = REGLD = (DISKWT) (A13) (BCNT = 15) + . . .

The preceding equation indicates that the disk write must be performed with A13 = 1 for data transfer on byte boundaries. When formatting a track, the write operation must be synchronized with the index pulse, and the bit counter must be cleared regardless of its current state. When this type of write operation is to be performed, A13 must be 0.

READY = CLKSH = DTASH = REGLD = (DISKWT) (A13) (BCNT = 15) + (DISKWT) (A13–) (INDSYN) + . . .

$$\text{BCLR}- = \overline{(\text{DISKWT})\,(\text{A13}-)\,(\text{INDSYN}) + \ldots}$$

As the data byte is loaded into the data shift register, address lines A9, A10, and A11 select the clock pattern to be loaded into the clock shift register (see Table 3).

Table 3. Write Clock Patterns

A9	A10	A11	Clock Pattern
0	0	0	C7 (ID and Data Mark)
0	0	1	D7 (Track Mark)
1	1	1	FF (Normal Data)

When the transfer is complete to the clock and data shift registers, the write mode (WTMODE) flip flop is set, causing $\overline{\text{WRITE ENABLE}}$ to become active. If another byte is not written at the next byte boundary, WTMODE is reset, causing the control logic to revert to the read mode (RDMODE = 1). Also, control reverts to read mode and the bit counter is cleared when the index pulse occurs and when no write operation synchronized to the index pulse is being performed. This is useful when formatting a track, since $\overline{\text{WRITE ENABLE}}$ will automatically be turned off when the second index pulse occurs. If an index pulse occurs during a write operation with A13 = 1, the CPU proceeds, but no data transfer takes place.

WTMDD = (WTMODE) (BCNT = 15–) (INDSYN–) + (DISKWT) (A13) (BCNT = 15) + (DISKWT) (A13–) INDSYN)

BCLR– = $\overline{\text{INDSYN}}$ + . . .

READY = (DISKWT) [(A13) (BCNT = 15) + INDSYN)] + . . .

While WTMODE = 1, write data is generated by alternately shifting out bits from the clock and data shift register every two microseconds. Shifting of the clock shift register occurs when CLKSH = 1, and shifting of the data shift register when DTASH = 1. The shift is enabled by BITTIME, which is active for one clock cycle every 2 μs by loading the counter with 10_{10} each time TCNTCY = 1.

BITTIME = (WTMODE) (TCNTCY) + . . .

TCNTLDD = TCNTLDB = WTMODE + . . .

CLKSH = (DISKWT) [(A13) (BCNT = 15) + (A13–) (INDSYN)] + (WTMODE) (BCNTA–) (BITTIME) + . . .

DTASH = (DISKWT) [(A13) (BCNT = 15) + (A13–) (INDSYN)] + (WTMODE) (BCNTA) (BITTIME) + . . .

WRTDTAD = (WTMODE) (BITTIME) [(CLK0) (BCNTA–) + (DTA0) (BCNTA)]

On even bit counts (BCNTA = 0) clock bits are shifted, and on odd bits (BCNTA = 1) data bits are shifted, producing the desired interleaving of clock and data bits. (See Figure 26.)

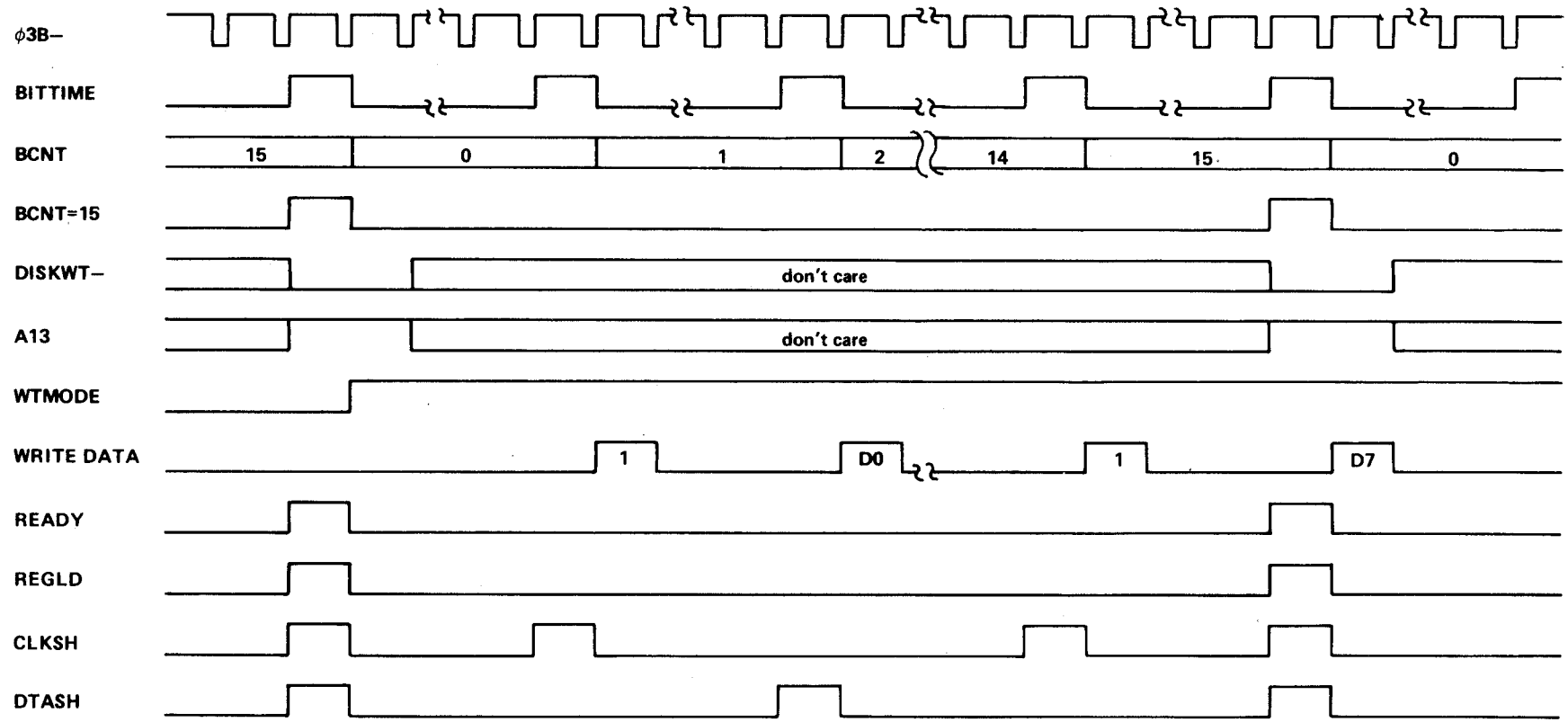


Figure 26. Write Timing

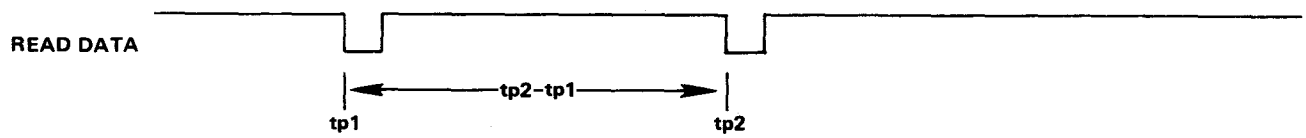
4.2 DISK READ OPERATIONS

Any time disk write operations are not being performed, the read/write control logic defaults to the read mode (RDMODE = 1). The following functions are performed to enable the CPU to read diskette data:

1. Conversion of FM to digital data;
2. Separation of clock and data bits;
3. Byte synchronization of the bit string;
4. Assembly of the seria data into bytes to be ready by CPU.

4.2.1 Clock and Data Bit Detection

Clock and data bits read from the disk are represented as a series of pulses. Each logic one clock or data bit is simply a pulse. Logic zero data and clock bits are indicated by the absence of a pulse between two pulses separated by a full data period (4 μs). Under ideal circumstances, detection of zero bits could be achieved by simply measuring the time between pulses. If $t_{P2}-t_{P1}$ = 2 μs, no zero bit is present; and if $t_{P2}-t_{P1}$ = 4 μs, a zero bit occurs between the two pulses.



Three phenomena make zero-bit detection more complex:

1. Variations in rotational speed of the disk;
2. Uncertainty of measured delays when using synchronous counters;
3. Apparent positional distortion or "bit-shifting" resulting from the tendency of pulses to move away from adjacent pulses.

Disk speed variations are typically specified at ±2% by diskette drive manufacturers. Figure 27 illustrates the bit shifting phenomenon:

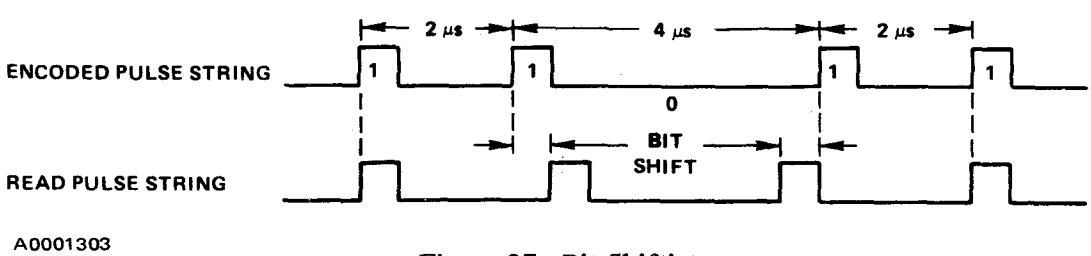


Figure 27. Bit Shifting

Pulses in the string have a tendency to move away from each other, and the closer together the pulses, the stronger the tendency to separate. A zero bit causes contiguous pulses to move toward each other, reducing pulse separation and complicating zero detection.

The bit detector is used to generate the synchronous signal BITTIME, which is active when a one or zero bit has been detected.

BITTIME = (RDMODE) (BITIN) + . . .

Detection of zero bits is accomplished by measuring the time between successive pulses. When TCNTCY = 1 and BITIN = 0, a zero bit is detected.

BITTIME = (RDMODE) (BITIN + TCNTCY) + . . .

Data and clock bits could be detected by measuring the time between read pulses, and if this time is greater than 3 μs, a zero bit is present; otherwise, no zero bit is present. Since the read pulse is asynchronous to the system, the time between pulses can only be measured to an accuracy of 333 ns ($\pm$1 clock cycle). For example, if the counter in Figure 28 is loaded with seven, no zero will be detected if the time between pulses ($t_{P2} - t_{P1}$) is less than 3.0 μs, and a zero will always be detected if $t_{P2} - t_{P1} > 3.333$ μs. If 3.0 $\mu s < t_{P2} - t_{P1} < 3.333$ μs, an ambiguity occurs in that a zero may or may not be detected. Similarly, if the counter is loaded with eight rather than seven, no zero bit will be detected if $t_{P2} - t_{P1} < 2.667$ μs, a zero bit will be detected if $t_{P2} - t_{P1} > 3.0$ μs, and the result is indeterminate if 2.667 $\mu s < t_{P2} - t_{P1} < 3.0$ μs. Most floppy-disk drive manufacturers specify that the maximum shift for any bit is 500 ns. Thus, two consecutive 1 bits may be separated by nearly 3.0 μs, and two 1 bits separated by a zero bit may shift toward each other to result in a minimum separation of nearly 3.0 μs. The combined distortion of consecutive 1 bits never fully reaches 1 μs, but the 667 ns margin provided by loading the counter with either seven or eight does not provide for reliable, accurate reading of data. (See Figure 28.)

As stated previously, adjacent 1 bits affect the direction of distortion of a particular 1 bit, with the closest pulses having the greatest effect. Empirical observation indicates that only the two bit positions on either side of a pulse have significant effect on a pulse, as shown in Table 4.

Table 4. Bit Shift Direction

Bit n-2	Bit n-1	Bit n	Direction of Distortion For Bit n	Bit n+1	Bit n+2
0	1	1	→	0	1
0	1	1	–	1	0
0	1	1	←	1	1
1	0	1	–	0	1
1	0	1	←	1	0
1	0	1	←	1	1
1	1	1	→	0	1
1	1	1	→	1	0
1	1	1	–	1	1

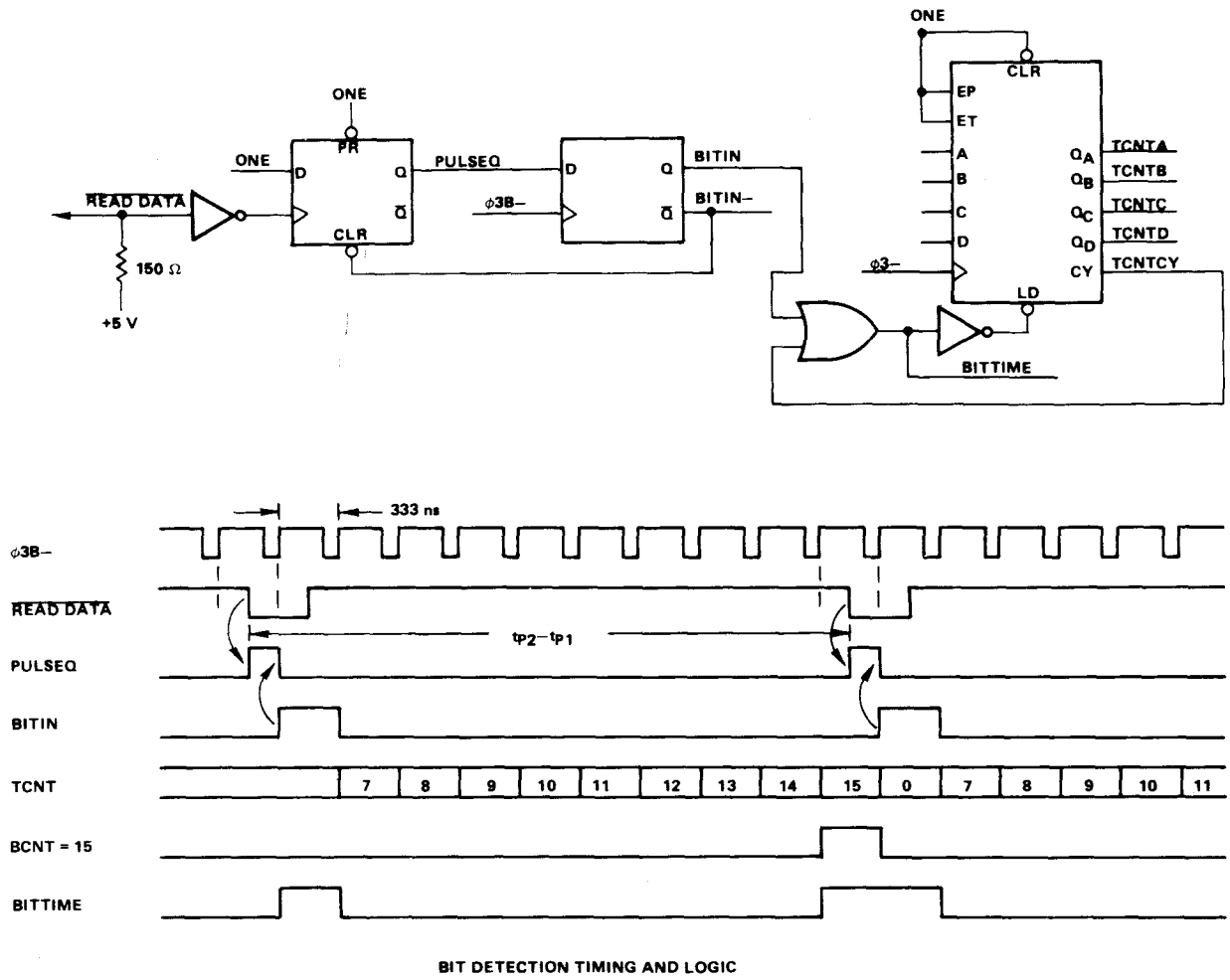


Figure 28. Bit Detection Timing and Logic

The most difficult detection problem is that of differentiating between two contiguous 1 bits which are shifted away from each other (worst case 11) and two 1 bits separated by a zero bit where the 1 bits move toward each other (worst case 101). The worst case 11 occurs in the patterns

			←	→			
Pattern A	0	1	1	1	1	0	, and
Pattern B	1	0	1	1	0	1	.
			→	←			

The worst case 101 occurs in the patterns

			→		←		
Pattern C	0	1	1	0	1	1	, and
Pattern D	1	1	1	0	1	1	.
			→		←		

The timing logic is such that the period of uncertainty does not lie in the area where a severely distorted pulse will occur; that is, when the worst case 11 can occur, and $t_{P2} - t_{P1} < 3.0\,\mu s$, the logic always

indicates that no zero was detected; when the worst case 101 can occur and $t_{p2} - t_{p1} > 3.0\ \mu s$, a zero is always detected. To accomplish this, the value loaded into the counter is shown in Table 5.

Table 5. Worst Case Pattern Load Values

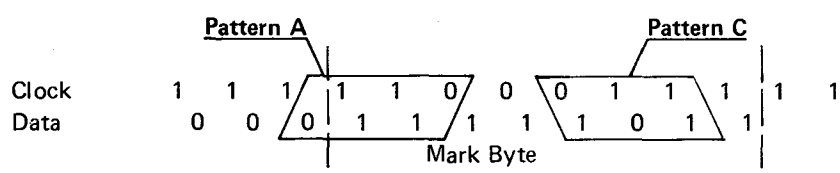
Pattern	Bit n−2	Bit n−1	Bit n	Bit n+1	Bit n+2	Bit n+3	Load Value
A	0	1	1	1	1	0	7
B	1	0	1	1	0	1	7
C	0	1	1	0	1	1	8
D	1	1	1	0	1	1	8

When bit n is detected, the counter is loaded with the value shown, dependent upon the data pattern.

Accommodation of patterns B and D are simple, since bits following that being sampled don't matter. Patterns A and C present the problem that, as the serial pulses are being read, the logic does not know what bits n+1, n+2, and n+3 are going to be.

Further analysis of the data format reveals that patterns A and C occur only when an ID or data mark are being read, see Table 6.

Table 6. Data Mark



Pattern A can only occur at the beginning of an ID, data, or deleted data mark, and pattern C can only occur in a data mark. With pattern A, the first 0 is a data bit, and with pattern C, the first 0 is a clock bit. BCNTA selects whether the current 1 bit is to be shifted into the clock or data shift register. The previous two bits are CLK7 and DTA7, the LSB's of the clock and data shift registers, and the order of these bits is determined by BCNTA. Using this information, the values loaded into the counter are as shown in Table 7.

TCNTLDD = (RDMODE)](CLK7) (DTA7) + (BCNTA−) (DTA7)] + . . .

TCNTLDB = (RDMODE) [(DTA7−) + (BCNTA) (CLK7−)] + . . .

The bit detector will thus adjust its count interval to accommodate the worst-case distortion which can occur for the anticipated data pattern.

Table 7. Bit Detector Counter Load Values

BCNTA	CLK7	DTA7	Load Value
0	0	0	Illegal
0	0	1	8
0	1	1	8
0	1	0	7
1	1	0	7
1	1	1	8
1	0	1	7
1	0	0	Illegal

4.2.2 Clock/Data Separation

Each time BITTIME is active, a new clock or data bit is shifted in. The value of the clock or data bit is BITIN. Since clock and data bits are interleaved, the value of BITIN will be alternately shifted into the clock or data shift register each time BITTIME is active. This is accomplished by incrementing the bit counter each time BITTIME is active, causing BCNTA to toggle. The equations for shifting the clock and data shift registers are:

CLKSH = (BITTIME) (BCNTA–) (RDMODE)+ . . .

DTASH = (BITTIME) (BCNTA) (RDMODE) + . . .

When four consecutive zeroes are detected in the clock shift register, the order in which bits go to the clock and data shift registers is reversed, since four consecutive zero clock bits never occur in the recording format used. This is accomplished by the control signal:

$$BCLD- = \overline{(CLK4-)\ (CLK5-)\ (CLK6-)\ (CLK7-)}.$$

When this signals becomes active, the bit counter is cleared to zero, and remains cleared until the next 1 bit is detected. This 1 bit is directed to the clock shift register, causing BCLD– to become inactive and normal operation is resumed. Synchronization is thus assured at the beginning of each ID and data field because each field is preceded by several bytes with all zero data bits and all one clock bits.

The timing for clock/data separation is shown in Figure 29.

4.2.3 Byte Synchronization

Initial byte synchronization is achieved when reading an ID or data field by detecting the unique clock pattern of $C7_{16}$ which occurs only in ID and data marks. The mark detect signal is expressed by the equation:

MRKDT = (CLK0) (CLK1) (CLK2–) (CLK3–) (CLK4–) (CLK5) (CLK6) (CLK7)

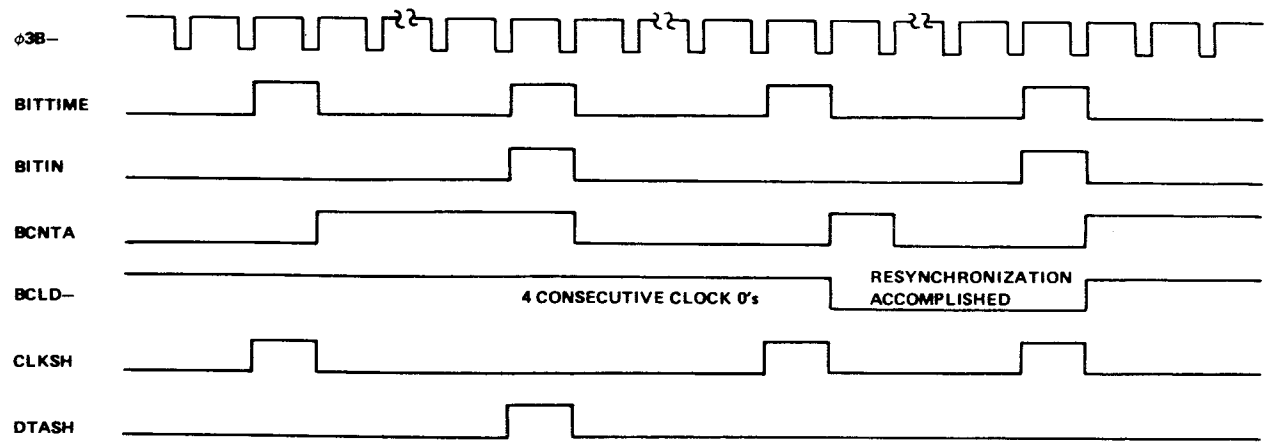


Figure 29. Clock/Data Separation Timing

After the mark is detected, one additional BITTIME must occur, allowing the data bit to be shifted into the data shift register.

4.2.4 Reading Disk Data

Two types of disk reads may be performed. When reading an ID or data field, the first byte read is always the ID or data mark. This is accomplished by performing a disk read with A13 = 0. The READY input signal will not become active until MRKDT = 1 and BITTIME = 1. After the mark is read, byte synchronization is established and subsequent disk reads are performed with A13 = 1. In this case, READY becomes true at each byte boundary when BCNT = 15.

$$READY = (DSKRD)\,[(BCNTA)\,(MRKDT)\,(BITTIME)\,(A13-) + (BCNT = 15)\,(A13) + INDSYN] + \ldots$$

The addresses for the two types of disk reads are $7FF8_{16}$ for reading marks, and $7FFC_{16}$ for reading normal data. The INDSYN term of the above equation causes the read operation to be completed any time the index pulse is detected or when the disk becomes not ready. (See Figure 30.)

4.3 READ/WRITE LOGIC COMBINATION

This subsection summarizes the equations for the control lines resulting from the combination of the read and write control functions.

BCLD–

$$BCLD- = \overline{(CLK4-)\,(CLK5-)\,(CLK6-)\,(CLK7-)}$$

BCLR–

$$BCLR- = \overline{(RDMODE)\,(MRKDT)\,(BCNTA)\,(BITTIME) + (INDSYN)}$$

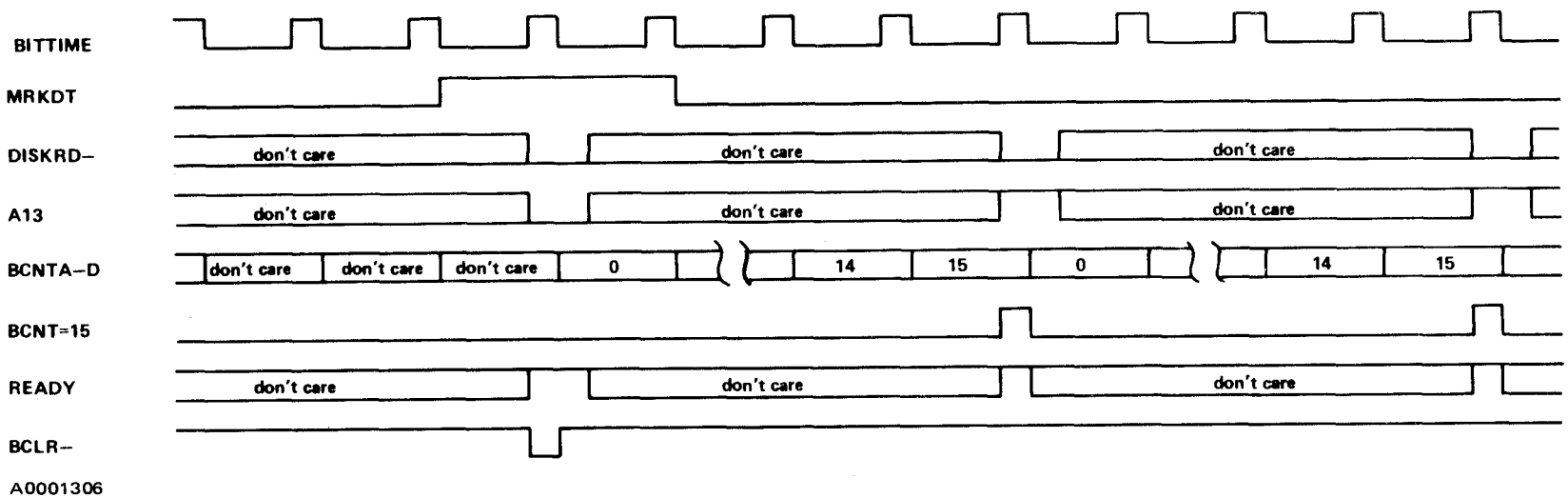
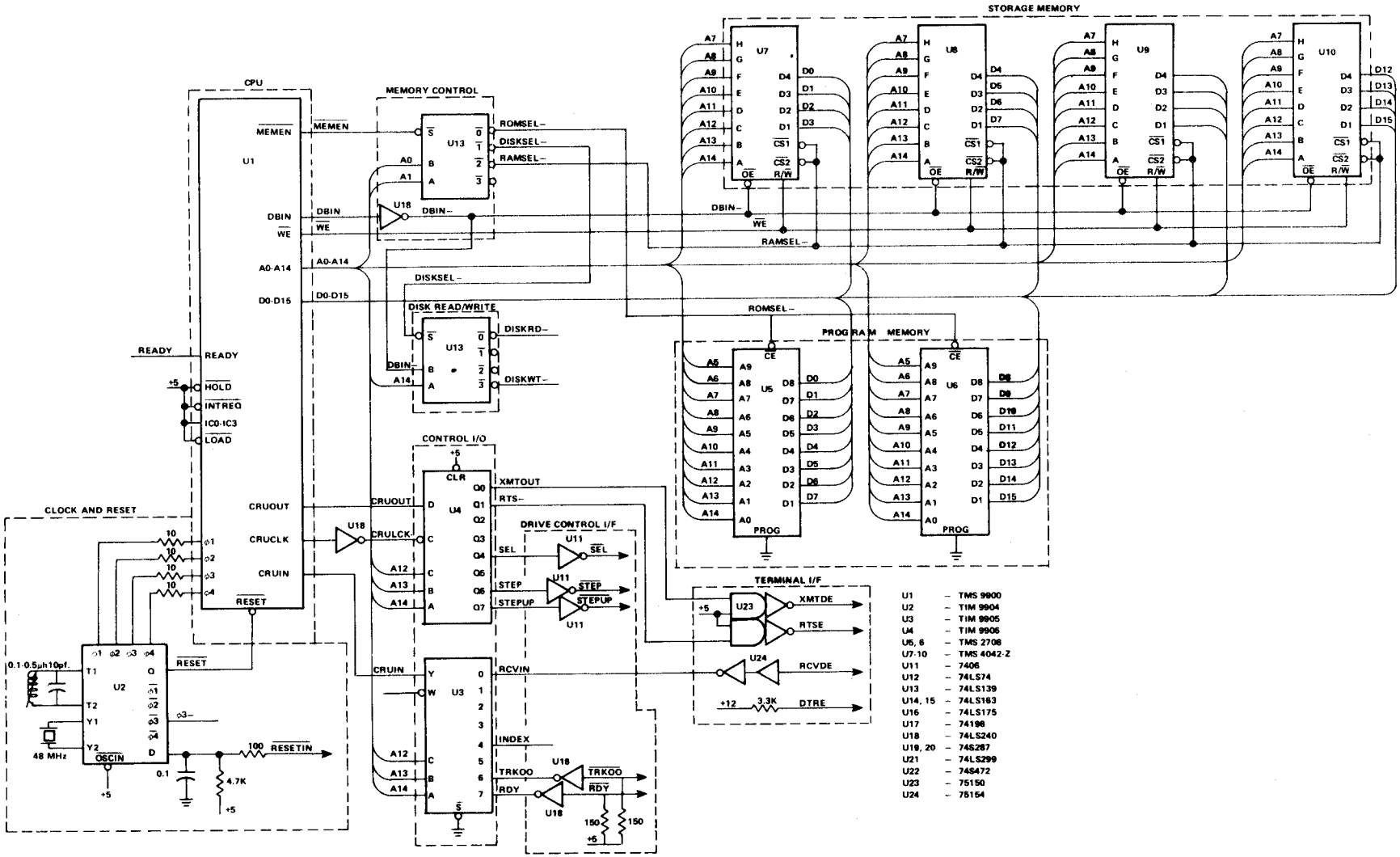
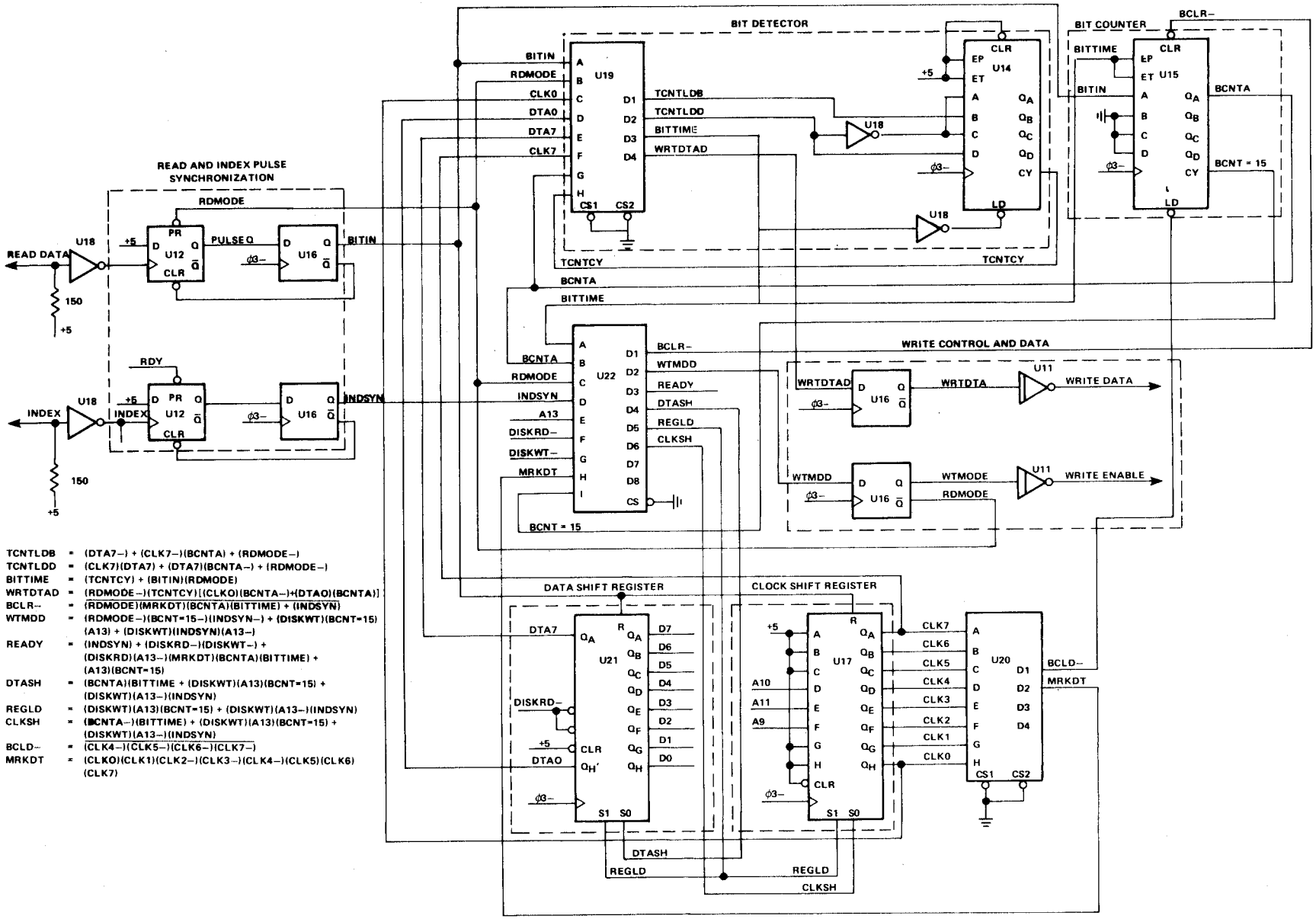


Figure 30. Disk Read Timing



A0001308

Logic Diagram, TMS 9900 Floppy Disk Controller
(Sheet 1 of 2)



TCNTLDB = (DTA7–) + (CLK7–)(BCNTA) + (RDMODE–)
TCNTLDD = (CLK7)(DTA7) + (DTA7)(BCNTA–) + (RDMODE–)
BITTIME = (TCNTCY) + (BITIN)(RDMODE)
WRTDTAD = (RDMODE–)(TCNTCY)[(CLKO)(BCNTA–)+(DTAO)(BCNTA)]
BCLR– = (RDMODE)(MRKDT)(BCNTA)(BITTIME) + (INDSYN)
WTMDD = (RDMODE–)(BCNT=15–)(INDSYN–) + (DISKWT)(BCNT=15)(A13) + (DISKWT)(INDSYN)(A13–)
READY = (INDSYN) + (DISKRD–)(DISKWT–) + (DISKRD)(A13–)(MRKDT)(BCNTA)(BITTIME) + (A13)(BCNT=15)
DTASH = (BCNTA)(BITTIME + (DISKWT)(A13)(BCNT=15) + (DISKWT)(A13–)(INDSYN)
REGLD = (DISKWT)(A13)(BCNT=15) + (DISKWT)(A13–)(INDSYN)
CLKSH = (BCNTA–)(BITTIME) + (DISKWT)(A13)(BCNT=15) + (DISKWT)(A13–)(INDSYN)
BCLD– = (CLK4–)(CLK5–)(CLK6–)(CLK7–)
MRKDT = (CLKO)(CLK1)(CLK2–)(CLK3–)(CLK4–)(CLK5)(CLK6)(CLK7)

A0001307

Logic Diagram, TMS 9900 Floppy Disk Controller
(Sheet 2 of 2)

BITTIME

BITTIME = (WTMODE) (TCNTCY) + (RDMODE) [(BITIN) + (TCNTCY)]
= (TCNTCY) + (RDMODE) (BITIN)

CLKSH

CLKSH = (DISKWT) [(A13) (BCNT = 15) + (A13–) (INDSYN)] + (WTMODE) (BCNTA–) (BITTIME) + (RDMODE) (BCNTA–) (BITTIME)
= (DISKWT) [(A13) (BCNT = 15) + (A13–) (INDSYN)] + (BCNTA–) (BITTIME)

DTASH

DTASH = (DISKWT) [(A13) (BCNT = 15) + (A13–) (INDSYN)] + (WTMODE) (BCNTA) (BITTIME) + (RDMODE) (BCNTA) (BITTIME)
= (DISKWT) [(A13) (BCNT = 15) + (A13–) (INDSYN)] + (BCNTA) (BITTIME)

MRKDT

MRKDT = (CLK0) (CLK1) (CLK2–) (CLK3–) (CLK4–) (CLK5) (CLK6) (CLK7)

READY

READY = (DISKWT) [(A13) (BCNT = 15) + (INDSYN)] + (DISKWT–) (DISKRD–) + (DISKRD) [(A13) (BCNT = 15) + (INDSYN) + (A13–) (MRKDT) (BCNTA) (BITTIME)]
= (DISKWT–) (DISKRD–) + (A13) (BCNT = 15) + (INDSYN) + (DISKRD) (A13–) (MRKDT) (BCNTA) (BITTIME)

REGLD

REGLD = (DISKWT) [(A13) (BCNT = 15) + (A13–) (INDSYN)]

TCNTLDB

TCNTLDB = (WTMODE) + (RDMODE) [(DTA7–) + (BCNTA) (CLK7–)]
= (WTMODE) + (DTA7–) + (BCNTA) (CLK7–)

TCNTLDD

TCNTLDD = (WTMODE) + (RDMODE) [(CLK7) (DTA7) + (BCNTA–) (DTA7)]
= (WTMODE) + (CLK7) (DTA7) + (BCNTA–) (DTA7)

WRTDTAD

WRTDTAD = (WTMODE) (BITTIME) [(CLK0) (BCNTA–) +(DTA0) (BCNTA)]
= (WTMODE) (TCNTCY) [(CLK0) (BCNTA–) + (DTA0) (BCNTA)]

WTMDD

WTMDD = (WTMODE) (BCNT = 15–) (INDSYN–) + (DISKWT) [(A13) (BCNT = 15) + (A13–) (INDSYN)]

SECTION V

SOFTWARE

The software design of a microprocessor system is as important as its hardware design. In this system, several functions which are normally performed by hardware are instead done in software in order to reduce device count. Examples of hardware/software tradeoffs include timing, transmit/receive, and CRC calculation.

5.1 SOFTWARE INTERFACE SUMMARY

The memory map in Figure 31 shows the memory address assignments for program memory, storage memory and the floppy-disk interface.

The CRU bit address assignments are summarized in Table 8 below.

Table 8. CRU Address Assignments

Bit Address	Output	Input
0	XMTOUT	RCVIN
1	RTS–	
2		
3		
4	SEL	INDEX
5		
6	STEP	TRK00
7	STEPUP	RDY

5.2 CONTROL SOFTWARE

Rather than providing individual examples of each individual control and data transfer function, all of the functions are combined to demonstrate complete system operation. The control software is modular, and the various subroutines may easily be adapted to different configurations of a TMS 9900 floppy-disk controller.

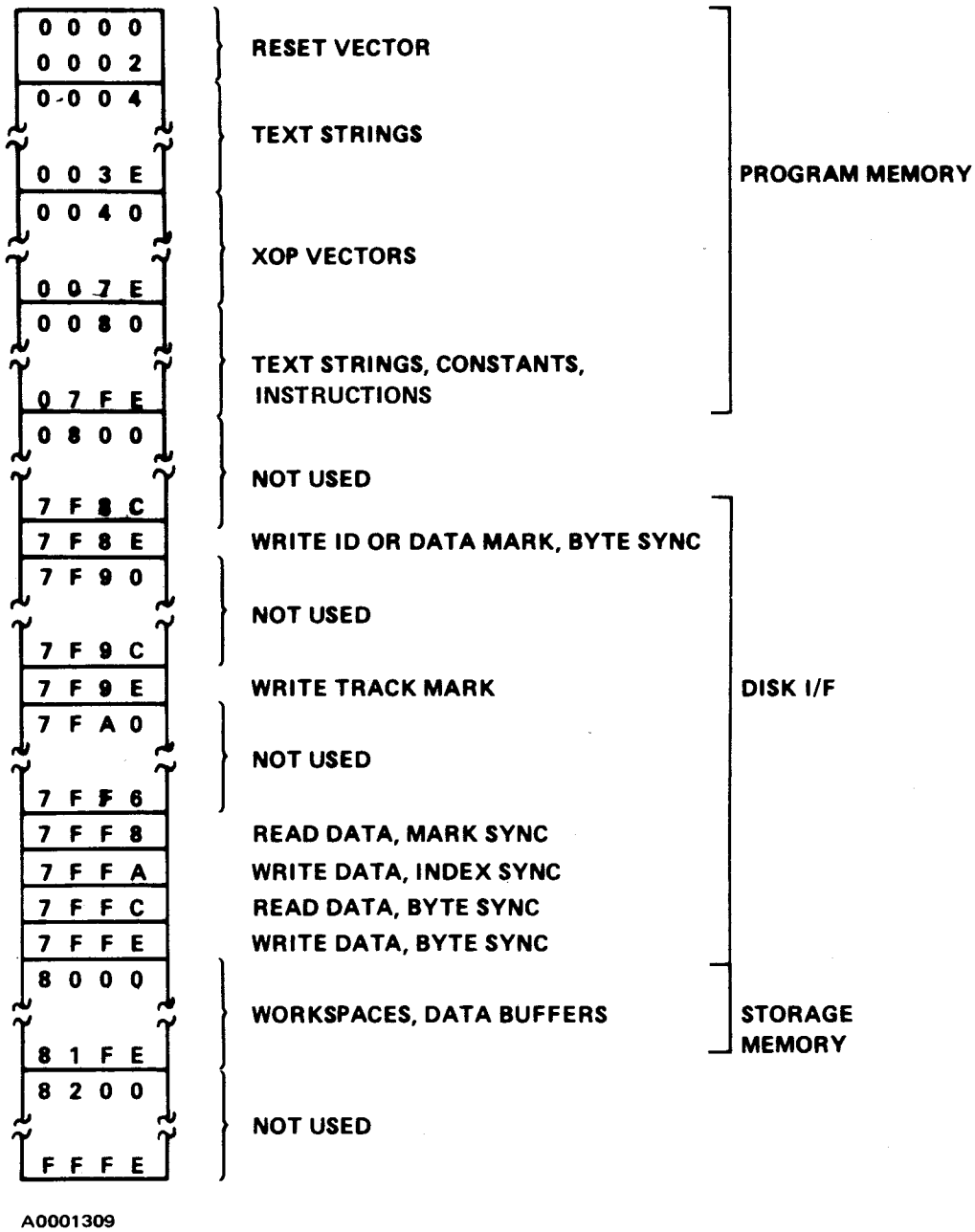


Figure 31. Memory Address Assignments

5.2.1 Floppy-Disk Control Program

This program contains the complete software for interfacing the TMS 9900 floppy-disk controller to both the RS-232 terminal and the floppy-disk drive.

5.2.2 Operator Commands

The commands listed in Table 9 are available to the terminal operator. These commands enable the user to write and read data to and from the diskette, format tracks, display and enter data from memory, and execute from a selected address. The user is able to load and execute diagnostics in addition to performing normal data transfer operations. When errors are encountered, error information is reported at the terminal.

Table 9. Operator Commands

?WA	TRACK = ct st,	SECTOR = cs ss,	NUMBER = sn
?WH	TRACK = ct st,	SECTOR = cs ss,	NUMBER = sn
?WD	TRACK = ct st,	SECTOR = cs ss,	NUMBER = sn
?RA	TRACK = ct st,	SECTOR = cs ss,	NUMBER = sn
?RH	TRACK = ct st,	SECTOR = cs ss,	NUMBER = sn
?FM	TRACK = ct st	END TRACK = st et	
?MD	sadd eadd		
?ME	sadd		
?MX	sadd		

Underscored characters are entered by the user. All others are supplied by the controller. The lower case fields are hexadecimal values. If the users enters a blank into these fields, the default value is used by the controller. Entry of any non-printable character (e.g., Carriage Return, ESCape) during command entry causes the command to be aborted. Entry of a non-hexadecimal value in hexadecimal fields causes the command to be aborted.

Table 10 lists the command entry parameters and Table 11 gives a summary of the commands.

Table 10. Command Entry Parameters

Parameter	Definition	Default Value	Range
ct	Current track number	–	$00 \leqslant ct \leqslant 4C\ (76_{10})$
st	Starting track number	ct	$00 \leqslant st \leqslant 4C$
cs	Current sector number	–	$01 \leqslant cs \leqslant 1A\ (26_{10})$
ss	Starting sector number	cs	$01 \leqslant ss \leqslant 1A$
sn	Number of sectors	01	$01 \leqslant sn \leqslant FF(255_{10})$
et	Ending track number	st	$st \leqslant et \leqslant 4C$
sadd	Starting address	8000	$0 \leqslant sadd \leqslant FFFF$
eadd	Ending address	sadd	$0 \leqslant eadd \leqslant FFFF$

Table 11. Command Summary

Command	Description
WA	Write ASCII. The ASCII character strings entered by the user are written sequentially onto the diskette. Each sector may be terminated, filling remaining bytes with 00, by entry of any non-printable character (ASCII code $< 20_{16}$) other than ESCape. Entry of ESCape aborts the command.
WH	Write Hexadecimal. Hexadecimal bytes entered by the user are written sequentially onto the diskette. Sector termination and abort are performed in the same way as for the WA command.
WD	Write Deleted Data. Same as WH command, except the Deleted Data Mark (Clock = $C7_{16}$ Data = $F8_{16}$) rather than the Data Mark (Clock = $C7_{16}$, Data = FB_{16}) is written at the beginning of the Data Field.
RA	Read ASCII. The specified sectors are read and printed out as ASCII character strings. Each sector is printed beginning at a new line, and printing continues until the end of the sector, or until a non-printable ASCII character is encountered. When more than 80 characters are printed, the controller prints the eighty-first character in the first position of the next line. The command may be aborted at the end of any sector by depressing the BREAK key before the last character of the sector is printed. If a Deleted Data field is encountered, it is reported, and normal operation continues.
RH	Read Hexadecimal. The specified sectors are read and printed out as hexadecimal bytes, 16 bytes per line. The command may be aborted by depressing the BREAK key before the last character of any line is printed. If a Deleted Data field is encountered, it is reported and normal operation continues.
FM	Format Track. The specified tracks are completely rewritten with gaps, Track Marks, ID fields, and Data fields. All zero data is written into the 128 bytes of the data field.
MD	Memory Display. The contents of the specified memory addresses are printed out in hexadecimal byte format. The address of the first word of each line is printed, followed by 16 bytes. The command may be aborted by depressing the BREAK key before the last character of any line is printed.
ME	Memory Enter. Beginning with the selected location, the memory address and contents are printed. If it is to be modified, the user enters a hexadecimal byte value which will be stored at that address. If the value is not to be changed, the user enters a blank character (SPACE bar). The address is then incremented and the process is repeated until a non-hex character is entered, terminating the command.
MX	The CPU begins execution at the selected memory location.

Figure 32 shows the control software for the system described in this application report.

SECTION VI

SUMMARY

This application report has provided a thorough discussion of the TMS 9900 floppy-disk controller hardware and software system design. The economy of the CRU and the high throughput capability of the memory bus result in an economical, powerful system. The memory-to-memory architecture of the TMS 9900, along with its powerful instruction set and addressing capability, make the TMS 9900 ideally suited for applications where large amounts of data manipulation are necessary. Also, software development time is optimized by the minimization of lines of code resulting from the memory-to-memory instructions and large number of working registers.

It is likely that the designer using this application report will have requirements that are not addressed in this design. Variations in the sector length are accommodated with slight software modification. Higher density recording formats such as MFM and M^2FM require changes in the bit detector and data-separation logic. Higher throughput can be achieved by using an LSI terminal interface such as the TMS 9902 asynchronous communication controller and hardware CRC generation. Controlling multiple disks requires only the addition of drive select control lines. In short, variations on this design are easily implemented through slight hardware and software modifications.

```
0002                    IDT  'FDCTRL'
0003                  ********************************************************************
0004                  *
0005                  *          FLOPPY DISK CONTROL PROGRAM
0006                  *
0007                  *          DECEMBER 21, 1976
0008                  *
0009                  *     THIS PROGRAM CONTAINS THE CONTROL SOFTWARE FOR THE
0010                  *     SYSTEM DESCRIBED IN THE "TMS 9900 FLOPPY DISK
0011                  *     CONTROL SYSTEM" APPLICATION REPORT.  THE PROGRAM
0012                  *     ALLOWS THE USER TO READ, WRITE, AND FORMAT DATA ON
0013                  *     FLOPPY DISK.  ADDITIONALLY, THE USER MAY ENTER,
0014                  *     DISPLAY, AND INITIATE EXECUTION FROM
0015                  *     ANY LOCATION IN MEMORY.  IT IS ASSUMED THAT THE
0016                  *     CONSOLE USED FOR COMMAND ENTRY AND DATA DISPLAY IS
0017                  *     A 300 BAUD, RS-232C TYPE TERMINAL.  THE COMMANDS
0018                  *     USED IN INTERFACING THE TERMINAL OPERATOR
0019                  *     INTERFACE TO THE CONTROLLER ARE FULLY
0020                  *     DESCRIBED IN SECTION 5.3 OF THE "TMS 9900
0021                  *     FLOPPY DISK CONTROL SYSTEM" APPLICATION
0022                  *     REPORT.
0023                  *
0024                  ********************************************************************
0025                  *
0026                  *     DISK TRANSFER ADDRESSES
0027                  *
0028        7F8E  MRKWT   EQU  >7F8E            DATA MARK WRITE
0029        7F9E  TKMWT   EQU  >7F9E            TRACK MARK WRITE
0030        7FFA  INDXWT  EQU  >7FFA            INDEX SYNC WRITE
0031        7FFE  DTAWT   EQU  >7FFE            BYTE SYNC WRITE
0032        7FF8  MRKRD   EQU  >7FF8            MARK SYNC READ
0033        7FFC  DTARD   EQU  >7FFC            BYTE SYNC READ
0034                  ********************************************************************
0035                  *
0036                  *     RAM EQUATES
0037                  *
0038        80C0  SECBUF  EQU  >80C0            CRC BUFFER FOR FORMATTING
0039        80F7  IDFLD   EQU  >80F7            ID FIELD IMAGE
0040        80F8  TKNUM   EQU  >80F8            TRACK NUMBER
0041        80FA  SECNUM  EQU  >80FA            SECTOR NUMBER
0042        80FC  IDCRC   EQU  >80FC            CRC FOR ID FIELD
0043        80FF  DTAFLD  EQU  >80FF            DATA FIELD IMAGE
0044        8100  DTABUF  EQU  >8100            128 BYTE DATA BUFFER
0045        8180  DTACRC  EQU  >8180            CRC FOR DATA FIELD
0046        8170  FDWP5   EQU  >8170            WORKSPACE 5
0047        8180  FDWP4   EQU  >8180            WORKSPACE 4
0048        81A0  FDWP3   EQU  >81A0            WORKSPACE 3
0049        81C0  FDWP2   EQU  >81C0            WORKSPACE 2
0050        81E0  FDWP1   EQU  >81E0            WORKSPACE 1
```

Figure 32. Floppy Disk Control Program (Sheet 1 of 28)

```
0052                    ********************************************************************
0053                    *
0054                    *       CRU EQUATES
0055                    *
0056          0000      RIN     EQU  0                  RECEIVE IN
0057          0004      INDEX   EQU  4                  INDEX PULSE
0058          0006      TRK00   EQU  6                  TRACK 00 INDICATOR
0059          0007      RDY     EQU  7                  DRIVE READY
0060          0000      XOUT    EQU  0                  TRANSMIT OUT
0061          0001      RTS     EQU  1                  REQUEST TO SEND
0062          0004      SEL     EQU  4                  DRIVE SELECT
0063          0006      STEP    EQU  6                  HEAD STEP CONTROL
0064          0007      STEPUP  EQU  7                  STEP DIRECTION CONTROL
0065                    ********************************************************************
0066                    *
0067                    *       XOP  EQUATES
0068                    *
0069                            DXOP ERPT,1             ERROR REPORT
0070                            DXOP IDRD,2             READ ID FIELD
0071                            DXOP TKST,3             SET TRACK
0072                            DXOP SINC,4             INCREMENT SECTOR
0073                            DXOP DSON,5             SELECT DRIVE ON
0074                            DXOP AXMT,6             ASCII DATA TRANSMIT
0075                            DXOP CRCI,7             ID FIELD CRC CALCULATION
0076                            DXOP CRCD,8             DATA FIELD CRC CALCULATION
0077                            DXOP TINC,9             INCREMENT TRACK
0078                            DXOP HRC2,10            RECEIVE HEX BYTE
0079                            DXOP HXM2,11            TRANSMIT HEX BYTE
0080                            DXOP NLIN,12            NEW LINE
0081                            DXOP RECV,13            RECEIVE CHARACTER
0082                            DXOP XMIT,14            TRANSMIT CHARACTER
0083                            DXOP DLAY,15            SOFTWARE TIME DELAY
0084                    ********************************************************************
0085                    *
0086                    *       TIME CONSTANTS
0087                    *
0088          00FA      HBDLY   EQU  250                HALF BIT (1.667 MS.)
0089          01F4      FBDLY   EQU  500                FULL BIT (3.333 MS.)
0090          03E8      B2DLY   EQU  1000               2 BITS (6.667 MS.)
0091          7530      B30DLY  EQU  30000              CARRIAGE RETURN
0092                    *                               (200 MS.)
0093          05DC      HSDLY   EQU  1500               HEAD STEP (10 MS.)
0094          1482      HDLDLY  EQU  5250               HEAD LOAD (35 MS.)
0095                    ********************************************************************
0096                    *
0097                    *       POWER ON RESET VECTOR
0098                    *
0099  0000    81E0      RSETVC  DATA FDWP1,START
      0002    ----
```

Figure 32. Floppy Disk Control Program (Sheet 2 of 28)

```
0101                    ********************************************************************
0102                    *
0103                    *       ERROR MESSAGES
0104                    *
0105  0004    49        NIDMSG  TEXT 'ID NOT FOUND'
0106  0010    00                BYTE 0
0107  0011    44        NDMMSG  TEXT 'DATA MARK NOT FOUND'
0108  0024    00                BYTE 0
0109  0025    44        NRDYMS  TEXT 'DRIVE NOT READY'
0110  0034    00                BYTE 0
0111  0035    43        CRCMSG  TEXT 'CRCC ERROR'
0112  003F    00                BYTE 0
0113                    ********************************************************************
0114                    *
0115                    *       XOP VECTORS
0116                    *
0117  0040  FFFF                DATA -1,-1
      0042  FFFF
0118  0044  81E0                DATA FDWP1,ERPTPC
      0046  ----
0119  0048  81C0                DATA FDWP2,IDRDPC
      004A  ----
0120  004C  8180                DATA FDWP4,TKSTPC
      004E  ----
0121  0050  81C0                DATA FDWP2,SINCPC
      0052  ----
0122  0054  8180                DATA FDWP4,DSONPC
      0056  ----
0123  0058  81C0                DATA FDWP2,AXMTPC
      005A  ----
0124  005C  81A0                DATA FDWP3,CRCIPC
      005E  ----
0125  0060  81A0                DATA FDWP3,CRCDPC
      0062  ----
0126  0064  81A0                DATA FDWP3,TINCPC
      0066  ----
0127  0068  81A0                DATA FDWP3,HRC2PC
      006A  ----
0128  006C  81A0                DATA FDWP3,HXM2PC
      006E  ----
0129  0070  81A0                DATA FDWP3,NLINPC
      0072  ----
0130  0074  8180                DATA FDWP4,RECVPC
      0076  ----
0131  0078  8180                DATA FDWP4,XMITPC
      007A  ----
0132  007C  8170                DATA FDWP5,DLAYPC
      007E  ----
```

Figure 32. Floppy Disk Control Program (Sheet 3 of 28)

```
0134                    ********************************************************************
0135                    *
0136                    *       ASCII VALUES
0137                    *
0138  0080     41       ASCIIA TEXT 'A'
0139  0081     46       ASCIIF TEXT 'F'
0140  0082     4D       ASCIIM TEXT 'M'
0141  0083     1B       ESC    BYTE >1B
0142  0084     20       BLANK  BYTE >20
0143  0085     3F       QUEST  BYTE >3F
0144  0086     07       BELL   BYTE >07
0145  0087     08       BACKSP BYTE >08
0146  0088     0D       CARRET BYTE >0D
0147  0089     0A       LINEFD BYTE >0A
0148                    ********************************************************************
0149                    *
0150                    *       ADDITIONAL TEXT MESSAGES
0151                    *
0152  008A     20       TKMSG  TEXT '  TRACK ='
0153  0093     00              BYTE 0
0154  0094     20       ENDMSG TEXT '  END'
0155  0099     00              BYTE 0
0156  009A     2C       SCTMSG TEXT ', SECTOR ='
0157  00A4     00              BYTE 0
0158  00A5     2C       NUMMSG TEXT ', NUMBER ='
0159  00AF     00              BYTE 0
0160  00B0     20       ADDMSG TEXT '  ADDRESS ='
0161  00BB     00              BYTE 0
0162  00BC     44       DLDMSG TEXT 'DELETED DATA FIELD'
0163  00CE     00              BYTE 0
0164                    ********************************************************************
0165                    *
0166                    *       DISK MARK CONSTANTS
0167                    *
0168  00CF     F8       DLDMRK BYTE >F8
0169  00D0     FE       IDMRK  BYTE >FE
0170  00D1     FB       DTMRK  BYTE >FB
0171  00D2     FC       TKMRK  BYTE >FC
0172                    EVEN
0173                    ********************************************************************
0174                    *
0175                    *       SUBROUTINE:  DLAY
0176                    *
0177                    *       CALLING SEQUENCE:  DLAY @COUNT
0178                    *
0179                    *       A SOFTWARE LOOP WILL BE EXECUTED THE NUMBER
0180                    *       OF TIMES SPECIFIED BY THE CALLING PROGRAM.
0181                    *       EACH ITERATION OF THE LOOP RESULTS IN A
0182                    *       DELAY OF 6.67 MICROSECONDS.
0183                    *
0184  00D4   060B     DLAYPC DEC  R11                DECREMENT COUNT
       007E**00D4'
0185  00D6   16FE            JNE  DLAYPC            LOOP IF NOT 0
0186  00D8   0380            RTWP                   RETURN
```

Figure 32. Floppy Disk Control Program (Sheet 4 of 28)

```
0188                ********************************************************************
0189                *
0190                *       SUBROUTINE:  RECV
0191                *
0192                *       CALLING SEQUENCE:  RECV @LOCATN
0193                *
0194                *       AN ASCII CHARACTER WITH CORRECT FORMATTING
0195                *       IS RECEIVED AND THEN RETRANSMITTED
0196                *       AT 300 BAUD.  THE RECEIVED CHARACTER IS STORED
0197                *       AT THE SPECIFIED LOCATION.
0198                *
0199  00DA  04CC   RECVPC CLR  R12                SET CRU BASE
      0076**00DA'
0200  00DC  1F00   RCV    TB   RIN                TEST RECEIVE INPUT
0201  00DE  13FE          JEQ  RCV                LOOP UNTIL RIN = 0
0202  00E0  2FE0          DLAY @HBDLY             DELAY HALF BIT TIME
      00E2  00FA
0203  00E4  1F00          TB   RIN                TEST RECEIVE INPUT
0204  00E6  13FA          JEQ  RCV                IF RIN = 0, VALID START BIT
0205  00E8  020A          LI   R10,>3F            INITIALIZE ACCUMULATOR
      00EA  003F
0206  00EC  2FE0   RCVLP  DLAY @FBDLY             DELAY FULL BIT TIME
      00EE  01F4
0207  00F0  1F00          TB   RIN                TEST RECEIVE INPUT
0208  00F2  16--          JNE  RCVOFF             SET MSB OF ACCUMULATOR
0209  00F4  026A          ORI  R10,>8000          IF RIN = 1
      00F6  8000
0210  00F8  091A   RCVOFF SRL  R10,1              SHIFT ACCUMULATOR
      00F2**1602
0211  00FA  18F8          JOC  RCVLP              IF CARRY, RECEIVE NEXT BIT
0212  00FC  2FE0          DLAY @B2DLY             DELAY 2 BIT TIMES
      00FE  03E8
0213  0100  1F00          TB   RIN                TEST RECEIVE INPUT
0214  0102  16EC          JNE  RCV                IF RIN = 0, FRAMING ERROR
0215  0104  D6CA          MOVB R10,*R11           MOVE RECEIVED CHARACTER
0216                *                             TO SPECIFIED LOCATION AND
0217                *                             RETRANSMIT.
```

Figure 32. Floppy Disk Control Program (Sheet 5 of 28)

```
0219                 ********************************************************************
0220                 *
0221                 *       SUBROUTINE:  XMIT
0222                 *
0223                 *       CALLING SEQUENCE:  XMIT @LOCATN
0224                 *
0225                 *       AN ASCII CHARACTER WITH CORRECT FORMATTING
0226                 *       AND EVEN PARITY IS TRANSMITTED AT 300 BAUD.
0227                 *       THE LOCATION OF THE CHARACTER TO BE TRANS-
0228                 *       MITTED IS SPECIFIED AS THE CALLING PARAMETER.
0229                 *
0230  0106  04CC    XMITPC CLR  R12              INITIALIZE CRU BASE
      007A**0106'
0231  0108  020A           LI   R10,3            INITIALIZE ACCUMULATOR
      010A  0003
0232  010C  0209           LI   R9,>8000         INITIALIZE PARITY MASK
      010E  8000
0233  0110  D29B           MOVB *R11,R10         FETCH CHARACTER
0234  0112  1C--           JOP  PARADJ           IF ODD PARITY INVERT MSB
0235  0114  04C9           CLR  R9               ELSE, CLEAR PARITY MASK
0236  0116  2A89    PARADJ XOR  R9,R10           XOR PARITY MASK
      0112**1C01
0237                 *                           WITH CHARACTER
0238  0118  1E01           SBZ RTS               TURN ON RTS
0239  011A  0B8A           SRC  R10,8            ROTATE CHARACTER
0240  011C  18--    XMTLP1 JOC  XOUTON           TEST TRANSMIT BIT
0241  011E  1E00           SBZ  XOUT             IF 0, RESET XOUT
0242  0120  10--           JMP  XFRDLY           AND SKIP
0243  0122  1D00    XOUTON SBO  XOUT             ELSE, SET XOUT
      011C**1802
0244  0124  2FE0    XFRDLY DLAY @FRDLY           DELAY FULL BIT TIME
      0126  01F4
      0120**1001
0245  0128  091A           SRL  R10,1            SHIFT ACCUMULATOR 1 BIT
0246  012A  16F8           JNE XMTLP1            IF NOT ZERO, TRANSMIT NEXT BIT
0247  012C  1D01           SBO  RTS              TURN OFF RTS
0248  012E  0380           RTWP                  RETURN
```

Figure 32. Floppy Disk Control Program (Sheet 6 of 28)

```
0250                    ********************************************************************
0251                    *
0252                    *       SUBROUTINE:  DSON
0253                    *
0254                    *       CALLING SEQUENCE:  DSON 0
0255                    *
0256                    *       THE FLOPPY DISK CRIVE IS SELECTED AND
0257                    *       THE SELECT DELAY PERIOD IS EXECUTED.  IF THE
0258                    *       DEVICE IS NOT READY, AN ERROR MESSAGE IS
0259                    *       PRINTED AND THE OPERATION IS ABORTED.
0260                    *       OTHERWISE, CONTROL RETURNS TO THE CALLING
0261                    *       PROGRAM.
0262                    *
0263  0130  04CC  DSONPC CLR  R12                INITIALIZE CRU BASE
      0056**0130'
0264  0132  1D04         SBO  SEL                SELECT DRIVE
0265  0134  2FE0         DLAY @HDLDLY            DELAY FOR HEAD LOAD
      0136  1482
0266  0138  1F07         TB   RDY                TEST DRIVE STATUS
0267  013A  13--         JEQ  DSONRT             IF READY, NORMAL RETURN
0268  013C  2C60         ERPT @NRDYMS            ELSE, ABORT AND PRINT
      013E  0025'
0269                    *                       ERROR MESSAGE.
0270  0140  0380  DSONRT RTWP                    NORMAL RETURN
      013A**1302
0271                    ********************************************************************
0272                    *
0273                    *       SUBROUTINE:  HRC2
0274                    *
0275                    *       CALLING SEQUENCE:  HRC2 @LOCATN
0276                    *
0277                    *       A BLANK IS TRANSMITTED AND 2 CHARACTERS
0278                    *       ARE RECEIVED.  IF EITHER CHARACTER IS A
0279                    *       BLANK, NO OPERATION IS PERFORMED AND THE
0280                    *       NORMAL RETURN IS EXECUTED.  IF TWO HEXADEC-
0281                    *       IMAL VALUES ARE ENTERED, THE HEXADECIMAL
0282                    *       BYTE IS STORED AT THE LOCATION SPECIFIED
0283                    *       AS THE CALLING PARAMETER.  IF EITHER CHARACTER
0284                    *       IS AN ESCAPE, CONTROL IS RETURNED TO THE MAIN
0285                    *       PROGRAM AT THE POINT WHERE OPERATOR COMMANDS
0286                    *       ARE REQUESTED.  IF ANY OTHER CHARACTER IS
0287                    *       RECEIVED, NO OPERATION IS PERFORMED AND THE
0288                    *       RETURN PC VALUE WILL BE THE CONTENTS OF REG-
0289                    *       ISTER 10 OF THE CALLING PROGRAM.
0290                    *
0291  0142  81E0  RTRNVC DATA FDWP1,TOP          RETURN VECTOR
      0144  ----
0292  0146  2FA0  HRC2PC XMIT @BLANK             TRANSMIT BLANK
      0148  0084'
      006A**0146'
0293  014A  04CA         CLR  R10                CLEAR HEX ACCUMULATOR
0294  014C  0708         SETO R8                 INITIALIZE CHARACTER COUNTER
0295  014E  2F49  HRC2LP RECV R9                 FETCH CHARACTER
0296  0150  9909         CB   R9,@ESC            COMPARE TO ESCAPE
```

Figure 32. Floppy Disk Control Program (Sheet 7 of 28)

```
                0152  0083'
0297            0154  16--        JNE   NOTESC          IF NOT,CONTINUE
0298            0156  0420        BLWP  @RTRNVC        ELSE, ABORT COMMAND
                0158  0142'
0299            015A  9809  NOTESC CB    R9,@BLANK      COMPARE TO BLANK
                015C  0084'
                0154**1602
0300            015E  13--        JEQ   HRC2RT         IF = BLANK, RETURN
0301                        *                          ELSE, CONVERT TO HEXADECIMAL
0302            0160  0229        AI    R9,->3000      SUBTRACT ASCII BIAS
                0162  D000
0303            0164  11--        JLT   HRC2AB         IF LESS THAN >30, ABORT
0304            0166  0289        CI    R9,>A00        TEST FOR NUMERIC
                0168  0A00
0305            016A  11--        JLT   NOHAJ          IF NUMERIC, SKIP
0306            016C  0229        AI    R9,->700       ELSE, SUBTRACT ALPHA BIAS
                016E  F900
0307            0170  0289        CI    R9,>A00        IF LESS THAN >41, ABORT
                0172  0A00
0308            0174  11--        JLT   HRC2AB
0309            0176  0289        CI    R9,>FFF        COMPARE TO ASCII F
                0178  0FFF
0310            017A  15--        JGT   HRC2AB         IF GREATER THAN, ABORT
0311            017C  F289  NOHAJ  SOCB  R9,R10         STORE HEX VALUE IN
                016A**1108
0312                        *                          ACCUMULATOR
0313            017E  0588        INC   R8             INCREMENT CHARACTER COUNT
0314            0180  16--        JNE   HRC2ND         IF NOT 0, SKIP
0315            0182  0A4A        SLA   R10,4          SHIFT HEX ACCUMULATOR
0316            0184  10E4        JMP   HRC2LP         FETCH SECOND CHARACTER
0317            0186  D6CA  HRC2ND MOVB  R10,*R11       STORE HEX VALUE
                0180**1602
0318                        *                          AT SPECIFIED LOCATION
0319            0188  0380  HRC2RT RTWP                RETURN
                015E**1314
0320            018A  C3AD  HRC2AB MOV   @20(R13),R14  MODIFY RETURN PC
                018C  0014
                0164**1112
                0174**110A
                017A**1507
0321            018E  10FC        JMP   HRC2RT         RETURN
```

Figure 32. Floppy Disk Control Program (Sheet 8 of 28)

```
0323                    ********************************************************************
0324                    *
0325                    *       SUBROUTINE:  HXM2
0326                    *
0327                    *       CALLING SEQUENCE:  HXM2 @LOCATN
0328                    *
0329                    *       THE HEXADECIMAL EQUIVALENT OF THE VALUE CONTAINED
0330                    *       IN THE LOCATION SPECIFIED BY THE PARAMETER IS
0331                    *       TRANSMITTED, PRECEDED BY A BLANK
0332                    *
0333  0190  2FA0  HXM2PC XMIT @BLANK             TRANSMIT BLANK
      0192  0084'
      006E**0190'
0334  0194  D29B        MOVB *R11,R10           FETCH BYTE
0335  0196  06A0        BL   @HEXXMT            TRANSMIT FIRST CHARACTER
      0198  ----
0336  019A  0A4A        SLA  R10,4              SHIFT BYTE
0337  019C  06A0        BL   @HEXXMT            TRANSMIT SECOND CHARACTER
      019E  ----
0338  01A0  0380        RTWP                    RETURN
0339  01A2  C24A  HEXXMT MOV  R10,R9            MOVE CHARACTER
      0198**01A2'
      019E**01A2'
0340  01A4  0949        SRL  R9,4               SHIFT RIGHT 4 BITS
0341  01A6  0289        CI   R9,>A00            TEST FOR NUMERIC
      01A8  0A00
0342  01AA  11--        JLT  NHADJ              IF SO, SKIP
0343  01AC  0229        AI   R9,>700            ELSE,ADD ALPHA BIAS
      01AE  0700
0344  01B0  0229  NHADJ  AI   R9,>3000          ADD ASCII BIAS
      01B2  3000
      01AA**1102
0345  01B4  2F89        XMIT R9                 TRANSMIT HEX ASCII CHARACTER
0346  01B6  045B        RT                      RETURN
0347                    ********************************************************************
0348                    *
0349                    *       SUBROUTINE:  NLIN
0350                    *
0351                    *       CALLING SEQUENCE:  NLIN 0
0352                    *
0353                    *       THE PRINTER IS ADVANCED TO THE BEGINNING
0354                    *       OF THE NEXT LINE
0355                    *
0356  01B8  2FA0  NLINPC XMIT @CRRET             CARRIAGE RETURN
      01BA  0088'
      0072**01B8'
0357  01BC  2FE0        DLAY @B30DLY            CARRIAGE RETURN DELAY
      01BE  7530
0358  01C0  2FA0        XMIT @LINEFD            LINE FEED
      01C2  0089'
0359  01C4  0380        RTWP                    RETURN
```

Figure 32. Floppy Disk Control Program (Sheet 9 of 28)

```
0361                    ********************************************************************
0362                    *
0363                    *       SUBROUTINE:  IDRD
0364                    *
0365                    *       CALLING SEQUENCE:  IDRD 0
0366                    *
0367                    *       EACH ID FIELD OF THE CURRENT DISKETTE TRACK
0368                    *       IS READ UNTIL THE ID FIELD WITH THE CORRECT
0369                    *       TRACK, SECTOR, AND CRC IS FOUND, AT WHICH
0370                    *       TIME THE ROUTINE IS EXITED.  IF THE CORRECT
0371                    *       FIELD IS NOT FOUND WITHIN A COMPLETE DISK
0372                    *       REVOLUTION (IE BEFORE 2 INDEX PULSES ARE
0373                    *       DETECTED), THE OPERATION IS ABORTED AND AN
0374                    *       ERROR MESSAGE IS REPORTED.
0375                    *
0376  01C6  2DC0       IDRDPC CRCI 0                  UPDATE ID FIELD IMAGE CRC
      004A**01C6'
0377  01C8  2D40              DSON 0                   TURN ON DRIVE
0378  01CA  0209              LI   R9,2                INITIALIZE INDEX PULSE COUNT
      01CC  0002
0379  01CE  020A       IDMRD  LI R10,IDFLD            SET POINTER TO ID FIELD
      01D0  80F7
0380  01D2  9EA0              CB   @MRKRD,*R10+        COMPARE DISK BYTE TO
      01D4  7FF8
0381                    *                              MARK CHARACTER
0382  01D6  13--              JEQ  MRKFND              IF MARK, CONTINUE
0383  01D8  1F04              TB   INDEX               ELSE, TEST FOR INDEX SIGNAL
0384  01DA  16F9              JNE  IDMRD               IF NO INDEX, REREAD DISK
0385  01DC  0609              DEC  R9                  IF INDEX,DECREMENT INDEX COUNT
0386  01DE  16F7              JNE  IDMRD               IF NOT 0, REREAD DISK
0387  01E0  2C60              ERPT @NIDMSG             ELSE, REPORT ID READ ERROR
      01E2  0004'
0388  01E4  0209       MRKFND LI   R9,6                LOAD BYTE COUNT
      01E6  0006
      01D6**1306
0389  01E8  983A       IDRDLP CB   *R10+,@DTARD       COMPARE DISK DATA
      01EA  7FFC
0390                    *                              TO ID FIELD IMAGE
0391  01EC  16F0              JNE  IDMRD               IF NOT EQUAL, START OVER
0392  01EE  0609              DEC R9                   DECREMENT BYTE COUNT
0393  01F0  16FB              JNE IDRDLP               IF NOT 0, READ NEXT BYTE
0394  01F2  0380              RTWP                     ELSE, ID FOUND, RETURN
```

Figure 32. Floppy Disk Control Program (Sheet 10 of 28)

```
0396                 ********************************************************************
0397                 *
0398                 *       SUBROUTINE:  ERPT
0399                 *
0400                 *       CALLING SEQUENCE:  ERPT @MESSAGE
0401                 *
0402                 *       THE MESSAGE WHOSE ADDRESS IS CONTAINED IN R11
0403                 *       WHEN THE ROUTINE IS ENTERED IS PRINTED,
0404                 *       FOLLOWED BY THE CURRENT TRACK AND SECTOR
0405                 *       NUMBER.  THE DRIVE IS TURNED OFF AND CONTROL
0406                 *       IS RETURNED TO THE COMMAND ENTRY PROGRAM.
0407                 *
0408  01F4  2F00     ERPTPC NLIN 0                    NEW LINE
      0046**01F4'
0409  01F6  2D9B            AXMT *R11                PRINT SELECTED MESSAGE
0410  01F8  2DA0            AXMT @TKMSG              PRINT TRACK MESSAGE
      01FA  008A'
0411  01FC  2EE0            HXM2 @TKNUM              PRINT TRACK NUMBER
      01FE  80F8
0412  0200  2DA0            AXMT @SCTMSG             PRINT SECTOR MESSAGE
      0202  009A'
0413  0204  2EE0            HXM2 @SECNUM             PRINT SECTOR NUMBER
      0206  80FA
0414  0208  1E04            SBZ  SEL                 TURN OFF DISK DRIVE
0415  020A  0420            BLWP @RTRNVC             RETURN TO COMMAND
      020C  0142'
0416                 *                                ENTRY PROGRAM
```

Figure 32. Floppy Disk Control Program (Sheet 11 of 28)

```
0418                ********************************************************************
0419                *
0420                *       SUBROUTINE:  AXMT
0421                *
0422                *       CALLING SEQUENCE:  AXMT @MESSAGE
0423                *
0424                *       THE ASCII CHARACTER STRING, THE
0425                *       BEGINNING ADDRESS OF WHICH IS CONTAINED
0426                *       IN R11, IS TRANSMITTED.  THE END OF THE
0427                *       STRING IS INDICATED BY A NON-PRINTABLE
0428                *       CHARACTER (IE LESS THAN HEX 20)
0429                *
0430  020E  020A   AXMTPC LI   R10,80              LOAD MAX CHARACTERS
      0210  0050
      005A**020E'
0431                *                             PER LINE
0432  0212  D27B   AXMTLP MOVB *R11+,R9           FETCH CHARACTER
0433  0214  9809          CB   R9,@BLANK          PRINTABLE CHARACTER?
      0216  0084'
0434  0218  11--          JLT  AXMTRT             IF NOT, RETURN
0435  021A  2F89          XMIT R9                 ELSE, PRINT CHARACTER
0436  021C  060A          DEC  R10                DECREMENT MAX CHAR COUNT
0437  021E  16F9          JNE  AXMTLP             IF NOT 0, FETCH NEXT CHAR
0438  0220  981B          CB   *R11,@BLANK        ELSE, IS NEXT CHAR
      0222  0084'
0439                *                             PRINTABLE?
0440  0224  11--          JLT  AXMTRT             IF NOT, RETURN
0441  0226  2F00          NLIN 0                  NEW LINE
0442  0228  10F4          JMP  AXMTLP             PRINT REST OF STRING
0443  022A  0380   AXMTRT RTWP                    STRING PRINTED, RETURN
      0218**1108
      0224**1102
```

Figure 32. Floppy Disk Control Program (Sheet 12 of 28)

```
0445                    ********************************************************************
0446                    *
0447                    *       SUBROUTINE:  CRCI
0448                    *
0449                    *       CALLING SEQUENCE:  CRCI 0
0450                    *
0451                    *       THE CRC IS CALCULATED FOR THE ID FIELD IMAGE
0452                    *       CONTAINED IN MEMORY AND STORED IN THE LAST 2
0453                    *       BYTES OF THE FIELD.
0454                    *
0455  022C  020A  CRCIPC LI    R10,IDFLD         SET UP ID FIELD POINTER
      022E  80F7
      005E**022C'
0456  0230  0209        LI    R9,5              SET UP ID FIELD COUNT
      0232  0005
0457  0234  06A0        BL    @CRCALC           CALCULATE CRC
      0236  ----
0458  0238  0380        RTWP                    RETURN
0459                    ********************************************************************
0460                    *
0461                    *       SUBROUTINE:  CRCD
0462                    *
0463                    *       CALLING SEQUENCE:  CRCD 0
0464                    *
0465                    *       THE CRC IS CALCULATED FOR THE DATA FIELD IMAGE
0466                    *       CONTAINED IN MEMORY AND STORED IN THE LAST 2
0467                    *       BYTES OF THE FIELD.
0468                    *
0469  023A  020A  CRCDPC LI    R10,DTAFLD        SET UP DATA FIELD POINTER
      023C  80FF
      0062**023A'
0470  023E  0209        LI    R9,129            SET UP DATA FIELD COUNT
      0240  0081
0471  0242  06A0        BL    @CRCALC           CALCULATE CRC
      0244  ----
0472  0246  0380        RTWP                    RETURN
```

Figure 32. Floppy Disk Control Program (Sheet 13 of 28)

```
0474                ********************************************************************
0475                *
0476                *       SUBROUTINE:  CRCALC
0477                *
0478                *       CALLING SEQUENCE:  LI   R10,FLDADD
0479                *                          LI   R9,FLDCNT
0480                *                          BL   @CRCALC
0481                *
0482                *       THE CYCLIC REDUNDANCY CHECK CHARACTER (CRC) FOR
0483                *       THE FIELD ADDRESSED BY R10 IS CALCULATED
0484                *       AND STORED IN THE LAST 2 BYTES OF THE
0485                *       FIELD.  THE LENGTH OF THE FIELD (EXCLUDING CRC)
0486                *       IS SPECIFIED BY R9.  THE CRC POLYNOMIAL IS
0487                *       X**16+X**12+X**5+1.  BEFORE CRC CALCULATION
0488                *       BEGINS, THE PARTIAL CRC IS PRESET TO ALL ONES.
0489                *       R7, R8, R9, AND R10 ARE DESTROYED.
0490                *
0491  0248  0708   CRCALC SETO R8                  PRESET PARTIAL CRC
      0236**0248'
      0244**0248'
0492  024A  04C7   CRCLP  CLR  R7                  CLEAR SCRATCH REGISTER
0493  024C  D1FA          MOVB *R10+,R7           FETCH NEXT BYTE
0494  024E  2A07          XOR  R7,R8              XOR NEW BYTE WITH CRC
0495  0250  C1C8          MOV  R8,R7              MOVE TO SCRATCH REG
0496  0252  0947          SRL  R7,4               SHIFT SCRATCH RIGHT 4
0497  0254  29C8          XOR  R8,R7              XOR CRC WITH SCRATCH
0498  0256  0247          ANDI R7,>FF00           MASK OFF LOWER BYTE
      0258  FF00
0499  025A  0947          SRL  R7,4               SHIFT SCRATCH RIGHT 4
0500  025C  2A07          XOR  R7,R8              XOR SCRATCH WITH CRC
0501  025E  0B77          SRC  R7,7               ROTATE SCRATCH RIGHT 7
0502  0260  2A07          XOR  R7,R8              XOR SCRATCH WITH CRC
0503  0262  06C8          SWPB R8                 REVERSE BYTES IN CRC
0504  0264  0609          DEC  R9                 DECREMENT BYTE COUNT
0505  0266  16F1          JNE  CRCLP              IF NOT 0, FETCH NEXT BYTE
0506  0268  DE88          MOVB R8,*R10+           ELSE, TRANSFER
0507  026A  06C8          SWPB R8                 CRC TO THE END
0508  026C  D688          MOVB R8,*R10            OF THE FIELD
0509  026E  045B          RT                      RETURN
```

Figure 32. Floppy Disk Control Program (Sheet 14 of 28)

```
0511                    ********************************************************************
0512                    *
0513                    *       SUBROUTINE:  SINC
0514                    *
0515                    *       CALLING SEQUENCE:  SINC 0
0516                    *
0517                    *       THE SECTOR NUMBER IS INCREMENTED BY 1.
0518                    *       IF THE NEW VALUE IS GREATER THAN 26,
0519                    *       THE SECTOR NUMBER IS SET TO 1 AND
0520                    *       THE TRACK NUMBER IS INCREMENTED,
0521                    *       AND THE HEAD IS STEPPED TO THE NEXT TRACK.
0522                    *
0523  0270  D2A0  SINCPC MOVB @SECNUM,R10       FETCH SECTOR NUMBER
      0272  80FA
      0052**0270'
0524  0274  022A        AI   R10,>100          ADD 1 TO SECTOR NUMBER
      0276  0100
0525  0278  028A        CI   R10,27*>100       COMPARE TO 27
      027A  1B00
0526  027C  14--        JHE  SECNXT            IF HIGH OR EQUAL,
0527                    *                      INCREMENT TRACK
0528  027E  D80A  SECXIT MOVB R10,@SECNUM       RESTORE SECTOR NUMBER
      0280  80FA
0529  0282  0380        RTWP                   RETURN
0530  0284  2E40  SECNXT INC  0                 INCREMENT TRACK NUMBER
      027C**1403
0531  0286  020A        LI   R10,>100          LOAD NEW SECTOR NUMBER
      0288  0100
0532  028A  10F9        JMP  SECXIT            STORE SECTOR NUMBER
0533                    ********************************************************************
0534                    *
0535                    *       SUBROUTINE:  TKST
0536                    *
0537                    *       CALLING SEQUENCE:  TKST @TRACK
0538                    *
0539                    *       THE READ/WRITE HEAD OF THE DISK DRIVE IS
0540                    *       STEPPED TO THE TRACK NUMBER SPECIFIED BY THE
0541                    *       LEFT BYTE OF R11, UNLESS THE DISK IS NOT
0542                    *       READY, IN WHICH CASE THE OPERATION IS ABORTED.
0543                    *       IF THE SPECIFIED TRACK IS OUT OF RANGE (IE
0544                    *       GREATER THAN 76, THE HEAD IS STEPPED TO TRACK
0545                    *       0.  THE NEW TRACK NUMBER REPLACES THE OLD
0546                    *       TRACK NUMBER IN MEMORY.  IF THE NEW TRACK
0547                    *       NUMBER IS TO BE 0, THE TRACK IS STEPPED
0548                    *       UNTIL THE TRK00 STATUS SIGNAL IS DETECTED.
0549                    *       IF THE OLD TRACK NUMBER WAS 0, THE HEAD IS
0550                    *       STEPPED TO TRACK 0 BEFORE THE NEW STEPPING
0551                    *       OPERATION BEGINS.
0552                    *
0553  028C  04CC  TKSTPC CLR  R12               INITIALIZE CRU BASE
      004E**028C'
0554  028E  1D04        SBO  SEL               SELECT DRIVE
0555  0290  2FE0        DLAY @HDLDLY           HEAD LOAD DELAY
      0292  1482
```

Figure 32. Floppy Disk Control Program (Sheet 15 of 28)

```
0556  0294  1F07          TB    RDY              CHECK DRIVE STATUS
0557  0296  13--          JEQ   TKCNTU           IF READY, CONTINUE
0558  0298  2C60          ERPT  @NRDYMS          ELSE, REPORT ERROR
      029A  0025'
0559  029C  C24B  TKCNTU  MOV   R11,R9           SAVE NEW TRACK NUMBER
      0296**1302
0560  029E  0989          SRL   R9,8             TO RIGHT BYTE OF R9
0561  02A0  13--          JEQ   TKT00            IF 0, CLEAR TRACK
0562  02A2  0289          CI    R9,76            NEW TRACK NUMBER IN RANGE?
      02A4  004C
0563  02A6  12--          JLE   TKNZRO           IS SO, SKIP
0564  02A8  04C9          CLR   R9               ELSE, CLEAR NEW TRACK NUMBER
0565  02AA  06A0  TKT00   BL    @TKCLR           STEP TO TRACK 00
      02AC  ----
      02A0**1304
0566  02AE  10--          JMP   TKSTRT           RETURN
0567  02B0  D2A0  TKNZRO  MOVB  @TKNUM,R10       FETCH OLD TRACK NUMBER
      02B2  80F8
      02A6**1204
0568  02B4  098A          SRL   R10,8            MOVE TO RIGHT BYTE
0569  02B6  16--          JNE   TKNZR1           IF NOT 00, CONTINUE
0570  02B8  06A0          BL    @TKCLR           ELSE, STEP TO TRACK 00
      02BA  ----
0571  02BC  8289  TKNZR1  C     R9,R10           COMPARE NEW TRACK
      02B6**1602
0572                      *                      TO OLD TRACK NUMBER
0573  02BE  11--          JLT   STPOUT           IF LESS THAN, STEP OUT 1 TRACK
0574  02C0  13--          JEQ   TKSTRT           IF EQUAL, RETURN
0575  02C2  1D07          SBO   STEPUP           ELSE, STEP IN 1 TRACK
0576  02C4  058A          INC   R10              INCREMENT OLD TRACK
0577  02C6  10--          JMP   TKGO             STEP HEAD
0578  02C8  1E07  STPOUT  SBZ   STEPUP           SELECT STEP OUT
      02BE**1104
0579  02CA  060A          DEC   R10              DECREMENT OLD TRACK
0580  02CC  06A0  TKGO    BL    @TKSTEP          STEP HEAD
      02CE  ----
      02C6**1002
0581  02D0  10F5          JMP   TKNZR1           REPEAT FOR NEXT STEP
0582  02D2  06C9  TKSTRT  SWPB  R9               MOVE NEW TRACK NUMBER
      02AE**1011
      02C0**1308
0583                      *                      TO LEFT BYTE
0584  02D4  D809          MOVB  R9,@TKNUM        UPDATE TRACK NUMBER
      02D6  80F8
0585  02D8  0380          RTWP                   RETURN
```

Figure 32. Floppy Disk Control Program (Sheet 16 of 28)

```
0587                    ********************************************************************
0588                    *
0589                    *       SUBROUTINE:  TKCLR
0590                    *
0591                    *       CALLING SEQUENCE:  BL   @TKCLR
0592                    *
0593                    *       THE READ/WRITE HEAD IS STEPPED OUT UNTIL
0594                    *       THE TRK00 STATUS SIGNAL BECOMES ACTIVE.
0595                    *       THE CONTENTS OF R8 AND R11 ARE DESTROYED.
0596                    *
0597  02DA  C20B  TKCLR MOV  R11,R8             SAVE RETURN LINKAGE
      02AC**02DA'
      02BA**02DA'
0598  02DC  1F07  TKCLP TB   RDY               TEST DRIVE STATUS
0599  02DE  16--        JNE TKCABT             IF NOT READY, ABORT
0600  02E0  1F06        TB   TRK00             TEST TRACK 00 STATUS SIGNAL
0601  02E2  16--        JNE  TKICNT            IF NOT ACTIVE,CONTINUE
0602  02E4  0458        B    *R8               ELSE, RETURN
0603  02E6  1E07  TKICNT SBZ  STEPUP           SET TO STEP OUT
      02E2**1601
0604  02E8  06A0        BL   @TKSTEP           STEP HEAD
      02EA  ----
0605  02EC  10F7        JMP  TKCLP             CONTINUE LOOP
0606  02EE  04C8  TKCABT CLR  R8               SET TRACK
      02DE**1607
0607  02F0  D808        MOVB R8,@TKNUM         NUMBER TO 00
      02F2  80F8
0608  02F4  2C60        ERPT @NRDYMS           REPORT ERROR AND ABORT
      02F6  0025'
0609                    ********************************************************************
0610                    *
0611                    *       SUBROUTINE:  TKSTEP
0612                    *
0613                    *       CALLING SEQUENCE:  BL   @TKSTEP
0614                    *
0615                    *       THE STEP PULSE IS GENERATED FOR 11.3
0616                    *       MICROSECONDS AND THE HEAD STEP DELAY
0617                    *       IS OBSERVED.
0618                    *
0619  02F8  1D06  TKSTEP SBO  STEP              SET STEP SIGNAL
      02CE**02F8'
      02EA**02F8'
0620  02FA  1000        NOP                    DUMMY DELAY
0621  02FC  1E06        SBZ  STEP              RESET STEP SIGNAL
0622  02FE  2FE0        DLAY @HSDLY            DELAY FOR HEAD STEP
      0300  05DC
0623  0302  045B        RT                     RETURN
```

Figure 32. Floppy Disk Control Program (Sheet 17 of 28)

```
0625                    ********************************************************************
0626                    *
0627                    *       SUBROUTINE:  TINC
0628                    *
0629                    *       CALLING SEQUENCE:  TINC 0
0630                    *
0631                    *       THE HEAD IS MOVED TO THE NEXT CONSECUTIVE
0632                    *       POSITION.  IF ON THE INNERMOST TRACK (76),
0633                    *       THE HEAD IS MOVED TO TRACK 00.
0634                    *
0635  0304  D2E0        TINCPC MOVB @TKNUM,R11          FETCH TRACK NUMBER
      0306  80F8
      0066**0304'
0636  0308  022B              AI   R11,>100             ADD 1 TO TRACK NUMBER
      030A  0100
0637  030C  2CDB              TKST *R11                 MOVE HEAD
0638  030E  0380              RTWP                      RETURN
0639                    ********************************************************************
0640                    *
0641                    *       COMMAND CHARACTER LIST
0642                    *
0643  0310    57        CMDLST TEXT 'WAWHWDRARHFMMDMEMX'
0644                    *
0645                    *       COMMAND ENTRY POINT TABLE
0646                    *
0647  0322  ----        CMDENT DATA WTASCI,WRTHEX,WRTDEL,RDASCI,RDHEX
      0324  ----
      0326  ----
      0328  ----
      032A  ----
0648  032C  ----              DATA FORMAT,DUMP,ENTER,EXECUT
      032E  ----
      0330  ----
      0332  ----
```

Figure 32. Floppy Disk Control Program (Sheet 18 of 28)

```
0650                    ********************************************************************
0651                    *
0652                    *       POWER-ON RESET ENTRY POINT
0653                    *
0654  0334  04CC  START  CLR  R12                 INITIALIZE CRU BASE
      0002**0334'
0655  0336  020B        LI   R11,>300           LOAD CRU INITIALIZATION VALUE
      0338  0300
0656  033A  320B        LDCR R11,8              AND OUTPUT TO CRU
0657  033C  020B        LI   R11,IDFLD          SET ID FIELD IMAGE POINTER
      033E  80F7
0658  0340  020A        LI   R10,>100           SET INITIAL SECTOR VALUE
      0342  0100
0659  0344  DEE0        MOVB @IDMRK,*R11+       ID MARK DATA PATTERN TO
      0346  00D0'
0660                    *                       FIRST BYTE OF ID FIELD IMAGE
0661  0348  DECC        MOVB R12,*R11+          0 TO SECOND BYTE
0662                    *                       (TRACK NUMBER)
0663  034A  DECC        MOVB R12,*R11+          0 TO THIRD BYTE
0664  034C  DECA        MOVB R10,*R11+          01 TO FOURTH BYTE
0665                    *                       (SECTOR NUMBER)
0666  034E  D6CC        MOVB R12,*R11           0 TO FIFTH BYTE
0667  0350  2CDC        TKST *R12               SET READ/WRITE HEAD TO TRACK 0
0668  0352  1E04        SBZ  SEL                TURN OFF DRIVE
0669                    ********************************************************************
0670                    *
0671                    *       OPERATOR COMMAND REQUEST ENTRY POINT
0672                    *
0673  0354  2F00  TOP   NLIN 0                   NEW LINE
      0144**0354'
0674  0356  2FA0        XMIT @QUEST             PRINT PROMPTING MESSAGE
      0358  0085'
0675  035A  2FA0        XMIT @BELL              (QUESTION MARK, BELL)
      035C  0086'
0676  035E  2F4A        RECV R10                READ FIRST CHARACTER
0677                    *                       OF COMMAND
0678  0360  06CA        SWPB R10                SAVE IN RIGHT BYTE
0679  0362  2F4A        RECV R10                READ SECOND CHARACTER
0680                    *                       OF COMMAND
0681  0364  06CA        SWPB R10                REVERSE CHARACTERS IN R10
0682  0366  0208        LI   R8,CMDLST          SET COMMAND LIST POINTER
      0368  0310'
0683  036A  0209        LI   R9,CMDENT-2        SET COMMAND ENTRY POINTER
      036C  0320'
0684  036E  C1F8  CMDLP  MOV  *R8+,R7            FETCH COMMAND IN LIST
0685  0370  13F1        JEQ  TOP                IF LIST VALUE = 0, NOT
0686                    *                       A LEGAL COMMAND
0687  0372  05C9        INCT R9                 INCREMENT ENTRY POINTER
0688  0374  81CA        C    R10,R7'            COMPARE ENTERED COMMAND
0689                    *                       TO LIST
0690  0376  16FB        JNE  CMDLP              IF NOT EQUAL, REPEAT
0691  0378  C259        MOV  *R9,R9             ELSE, COMMAND FOUND,
0692                    *                       FETCH ENTRY POINT
0693  037A  020A        LI   R10,TOP            COMMAND PROGRAM RETURN
```

Figure 32. Floppy Disk Control Program (Sheet 19 of 28)

```
      037C  0354'
0694                ♦                                   ADDRESS
0695  037E  9807          CB    R7,@ASCIIM              TEST FOR MD,ME, OR
      0380  0082'
0696                ♦                                   MX COMMANDS
0697  0382  13--          JEQ   ADDFCH                  IF SO, FETCH ADDRESS ENTRY
0698  0384  2DA0          XMT   @TKMSG                  PRINT TRACK MESSAGE
      0386  008A'
0699  0388  D220          MOVB  @TKNUM,R8               FETCH CURRENT TRACK
      038A  80F8
0700                ♦                                   NUMBER
0701  038C  2EC8          HXM2  R8                      PRINT TRACK NUMBER
0702  038E  2E88          HRC2  R8                      READ NEW TRACK NUMBER
0703  0390  0288          CI    R8,77♦256               NEW TRACK NUMBER LEGAL?
      0392  4D00
0704  0394  14DF          JHE   TOP                     IF NOT, ABORT
0705  0396  2CD8          TKST  ♦R8                     STEP HEAD TO NEW TRACK
0706  0398  1E04          SBZ   SEL                     TURN OFF DRIVE
0707  039A  9807          CB    R7,@ASCIIF              FORMAT COMMAND?
      039C  0081'
0708  039E  16--          JNE   SECFCH                  IF NOT, CONTINUE
0709  03A0  0459          B     ♦R9                     ELSE, EXECUTE COMMAND
0710  03A2  2DA0  SECFCH  XMT   @SCTMSG                 PRINT SECTOR MESSAGE
      03A4  009A'
      039E♦♦1601
0711  03A6  D1A0          MOVB  @SECNUM,R6              FETCH CURRENT SECTOR
      03A8  80FA
0712  03AA  2EC6          HXM2  R6                      PRINT CURRENT SECTOR
0713  03AC  2E86          HRC2  R6                      READ NEW SECTOR NUMBER
0714  03AE  0286          CI    R6,>100                 LESS THAN 1
      03B0  0100
0715  03B2  11D0          JLT   TOP                     IF SO, ABORT
0716  03B4  0286          CI    R6,27♦256               GREATER THAN 26?
      03B6  1B00
0717  03B8  14CD          JHE   TOP                     IF SO, ABORT
0718  03BA  D806          MOVB  R6,@SECNUM              UPDATE SECTOR NUMBER
      03BC  80FA
0719  03BE  2DA0          XMT   @NUMMSG                 PRINT NUMBER MESSAGE
      03C0  00A5'
0720  03C2  0205          LI    R5,>100                 LOAD DEFAULT NUMBER
      03C4  0100
0721  03C6  2E85          HRC2  R5                      READ NUMBER
0722  03C8  0985          SRL   R5,8                    MOVE TO RIGHT BYTE
0723  03CA  13C4          JEQ   TOP                     IF NUMBER = 0, ABORT
0724  03CC  0459          B     ♦R9                     EXECUTE COMMAND
0725  03CE  0208  ADDFCH  LI    R8,>8000                LOAD DEFAULT ADDRESS
      03D0  8000
      0382♦♦1325
0726  03D2  2E88          HRC2  R8                      READ FIRST BYTE OF ADDRESS
0727  03D4  06C8          SWPB  R8                      SAVE IN RIGHT BYTE
0728  03D6  2FA0          XMIT  @BACKSP                 BACKSPACE PRINTER
      03D8  0087'
0729  03DA  2E88          HRC2  R8                      READ SECOND BYTE OF ADDRESS
0730  03DC  06C8          SWPB  R8                      CORRECT ADDRESS BYTES
0731  03DE  0459          B     ♦R9                     EXECUTE COMMAND
```

Figure 32. Floppy Disk Control Program (Sheet 20 of 28)

```
0733                ********************************************************************
0734                *
0735                *       COMMAND CONTROL PROGRAM:  RDASCI,RDHEX
0736                *
0737                *       THESE COMMANDS ENABLE THE OPERATOR TO ACCESS
0738                *       A SPECIFIED NUMBER OF SECTORS BEGINNING AT THE
0739                *       CURRENT TRACK AND SECTOR LOCATION, PRINTING
0740                *       THE CONTENTS OF EACH SECTOR IN EITHER ASCII (RA)
0741                *       OR HEXADECIMAL FORMAT (RH).  IF A DELETED DATA
0742                *       FIELD IS DETECTED, IT IS REPORTED AND READING
0743                *       CONTINUES.  IF THE ID FIELD OR DATA MARK ARE
0744                *       NOT FOUND, OR IF A CRC ERROR OCCURS, THE ERROR
0745                *       IS REPORTED AND THE COMMAND IS ABORTED.
0746                *
0747                *       ENTRY PARAMETERS:       R10 = RETURN ADDRESS
0748                *                                R7 = COMMAND CHARACTERS
0749                *                                R5 = NUMBER OF SECTORS
0750                *                                     TO READ
0751                *
0752          03E0' RDASCI EQU  $                   RA COMMAND ENTRY POINT
      0328**03E0'
0753          03E0' RDHEX  EQU  $                   RH COMMAND ENTRY POINT
      032A**03E0'
0754    03E0  06C7         SWPB R7                  SWAP COMMAND CHAR BYTES
0755    03E2  0206  READ   LI   R6,DTAFLD           LOAD DATA FIELD IMAGE
        03E4  80FF
0756                *                                POINTER
0757    03E6  0208         LI   R8,DTARD            LOAD DISK DATA READ
        03E8  7FFC
0758                *                                ADDRESS
0759    03EA  2C80         IDRD 0                   READ ID FIELD
0760    03EC  DDA0         MOVB @MRKRD,*R6+         READ DATA MARK
        03EE  7FF8
0761    03F0  0200         LI   R0,130              REPEAT NEXT INSTRUCTION
        03F2  0082
0762                *                                130 TIMES
0763    03F4  DD98  RDLPL1 MOVB *R8,*R6+            MOVE DISK DATA TO
0764                *                                DATA FIELD IMAGE
0765    03F6  0600         DEC  R0
0766    03F8  16FD         JNE  RDLPL1
0767    03FA  1E04         SBZ  SEL                 TURN OFF DRIVE
0768    03FC  9820         CB   @DTAFLD,@DTMRK      NORMAL DATA MARK?
        03FE  80FF
        0400  00D1'
0769    0402  13--         JEQ  DMRKOK              IF SO, CONTINUE
0770    0404  9820         CB   @DTAFLD,@DLDMRK     DELETED DATA MARK?
        0406  80FF
        0408  00CF'
0771    040A  13--         JEQ  DDMXMT              IF SO, SKIP
0772    040C  2C60         ERPT @NDMMSG             PRINT ERROR MESSAGE
        040E  0011'
0773    0410  2F00  DDMXMT NLIN 0                   NEW LINE
      040A**1302
0774    0412  2DA0         AXMT @DLDMSG             REPORT DELETED DATA MARK
```

Figure 32. Floppy Disk Control Program (Sheet 21 of 28)

```
      0414  00BC'
0775  0416  C220  DMRKOK MOV  @DTACRC,R8      FETCH READ CRC
      0418  8180
      0402**1309
0776  041A  2E00        CRCD 0                RECALCULATE CRC
0777  041C  8220        C    @DTACRC,R8       CRC CORRECT?
      041E  8180
0778  0420  13--        JEQ  RDPRT            IF SO CONTINUE
0779  0422  2C60        ERPT @CRCMSG          ELSE, REPORT ERROR
      0424  0035'
0780  0426  2F00  RDPRT  NLIN 0                NEW LINE
      0420**1302
0781  0428  04E0        CLR  @DTACRC          CLEAR END OF DATA
      042A  8180
0782                    *                     FIELD IMAGE
0783  042C  0206        LI   R6,DTABUF        LOAD FIELD IMAGE
      042E  8100
0784                    *                     POINTER
0785  0430  9807        CB   R7,@ASCIIA       RA COMMAND?
      0432  0080'
0786  0434  13--        JEQ  ASCIRD           IF SO, PRINT IN ASCII
0787                    *                     FORMAT
0788  0436  0209        LI   R9,8             LOAD LINE COUNT
      0438  0008
0789  043A  0208  HXPTLP LI   R8,16           LOAD BYTE COUNT
      043C  0010
0790  043E  2F00        NLIN 0                NEW LINE
0791  0440  2ED6  HXPLP1 HXM2 *R6              PRINT DATA BYTE
0792  0442  0586        INC  R6               INCREMENT DATA POINTER
0793  0444  0608        DEC  R8               DECREMENT BYTE COUNT
0794  0446  16FC        JNE  HXPLP1           IF NOT 0, PRINT NEXT BYTE
0795  0448  1F00        TB   RIN              OPERATOR INTERRUPT?
0796  044A  16--        JNE  READRT           IF SO, ABORT
0797  044C  0609        DEC  R9               DECREMENT LINE COUNT
0798  044E  16F5        JNE  HXPTLP           IF NOT 0, PRINT NEXT LINE
0799  0450  10--        JMP  NXTSCT           CONTINUE
0800  0452  2D96  ASCIRD AXMT *R6              PRINT DATA FIELD
      0434**130E
0801                    *                     IN ASCII
0802  0454  2D00  NXTSCT SINC 0                UPDATE SECTOR NUMBER
      0450**1001
0803  0456  1F00        TB   RIN              OPERATOR INTERRUPT?
0804  0458  16--        JNE  READRT           IF SO, ABORT
0805  045A  0605        DEC  R5               DECREMENT SECTOR COUNT
0806  045C  16C2        JNE  READ             IF NOT 0, READ NEXT SECTOR
0807  045E  1E04  READRT SBZ  SEL              TURN OFF DRIVE
      044A**1609
      0458**1602
0808  0460  045A        B    *R10             RETURN
```

Figure 32. Floppy Disk Control Program (Sheet 22 of 28)

```
0810                    ********************************************************************
0811                    *
0812                    *       COMMAND CONTROL PROGRAM:  WRTHEX,WTASCI,WTDDTA
0813                    *
0814                    *       THESE COMMANDS ENABLE THE OPERATOR TO WRITE
0815                    *       A SPECIFIED NUMBER OF SECTORS OF DATA BEGINNING
0816                    *       AT THE CURRENT TRACK AND SECTOR LOCATION, IN
0817                    *       EITHER ASCII (WA) OR HEXADECIMAL (WD,WH) FORMAT.
0818                    *       THE WD COMMAND CAUSES A DELETED DATA MARK TO
0819                    *       PRECEDE THE DATA, AND THE WA AND WH COMMANDS
0820                    *       WRITE THE DATA MARK.  IF THE ID FIELD OF
0821                    *       ANY SECTOR IS NOT FOUND, AN ERROR IS
0822                    *       REPORTED.
0823                    *
0824                    *       ENTRY PARAMETERS:       R10 = RETURN ADDRESS
0825                    *                                R7 = COMMAND CHARACTERS
0826                    *                                R5 = NUMBER OF SECTORS
0827                    *                                     TO WRITE
0828                    *
0829  0462  D820  WRTDEL MOVB @DLDMRK,@DTAFLD    LOAD DELETED DATA MARK
      0464  00CF'
      0466  80FF
      0326**0462'
0830  0468  10--          JMP  WRITE            CONTINUE
0831        046A' WRTHEX  EQU  $                WH COMMAND ENTRY POINT
      0324**046A'
0832  046A  D820  WTASCI MOVB @DTMRK,@DTAFLD    LOAD DATA MARK
      046C  00D1'
      046E  80FF
      0322**046A'
0833  0470  06C7  WRITE  SWPB R7                MOVE SECOND COMMAND
      0468**1003
0834                    *                       CHARACTER TO LEFT BYTE
0835  0472  0208  WRITLP LI   R8,DTABUF         LOAD DATA FIELD
      0474  8100
0836                    *                       IMAGE POINTER
0837  0476  0200          LI   R0,64            REPEAT 64 TIMES
      0478  0040
0838  047A  04F8  WTLPL1 CLR  *R8+              CLEAR DATA BUFFER
0839  047C  0600          DEC  R0
0840  047E  16FD          JNE  WTLPL1
0841  0480  0208          LI   R8,DTABUF        LOAD DATA BUFFER POINTER
      0482  8100
0842  0484  9807          CB   R7,@ASCIIA       WA COMMAND?
      0486  0080'
0843  0488  13--          JEQ  WRTASC           IF SO, READ ASCII STRING
0844  048A  C10A          MOV  R10,R4           SAVE RETURN ADDRESS
0845  048C  020A          LI   R10,WTBRDY       LOAD HRC2 SUBROUTINE
      048E  ----
0846                    *                       RETURN ADDRESS
0847  0490  0209          LI   R9,8             LOAD LINE COUNT
      0492  0008
0848  0494  0206  WTHLP1 LI   R6,16             LOAD BYTE COUNT
      0496  0010
```

Figure 32. Floppy Disk Control Program (Sheet 23 of 28)

```
0849  0498  2F00        NLIN 0              NEW LINE
0850  049A  2E98  WTHLP2 HRC2 *R8            READ BYTE
0851  049C  0588        INC  R8             INCREMENT BUFFER POINTER
0852  049E  0606        DEC  R6             DECREMENT BYTE COUNT
0853  04A0  16FC        JNE  WTHLP2         IF NOT 0, READ NEXT BYTE
0854  04A2  0609        DEC  R9             DECREMENT LINE COUNT
0855  04A4  16F7        JNE  WTHLP1         IF NOT 0, READ NEXT LINE
0856  04A6  C284  WTBRDY MOV  R4,R10         RESTORE RETURN ADDRESS
      048E**04A6'
0857  04A8  10--        JMP  WTCRCD         CONTINUE
0858  04AA  0206  WRTASC LI   R6,128         LOAD CHARACTER COUNT
      04AC  0080
      0488**1310
0859  04AE  2F00        NLIN 0              NEW LINE
0860  04B0  2F58  WTASLP RECV *R8            READ CHARACTER
0861  04B2  9818        CB   *R8,@ESC       ESCAPE CHARACTER?
      04B4  0083'
0862  04B6  13--        JEQ  WRITRT         IF SO, RETURN
0863  04B8  9838        CB   *R8+,@BLANK    NON-PRINTABLE?
      04BA  0084'
0864  04BC  11--        JLT  WTCRCD         IF SO, END OF SECTOR
0865  04BE  0606        DEC  R6             DECREMENT CHARACTER COUNT
0866  04C0  16F7        JNE  WTASLP         IF NOT 0, READ NEXT CHAR
0867  04C2  2E00  WTCRCD CRCD 0              GENERATE DATA FIELD CRC
      04A8**100C
      04BC**1102
0868  04C4  0209        LI   R9,DTAWT       DISK DATA WRITE ADDRESS
      04C6  7FFE
0869  04C8  0208        LI   R8,DTAFLD      DATA FIELD IMAGE POINTER
      04CA  80FF
0870  04CC  2C80        IDRD 0              READ ID FIELD
0871  04CE  0200        LI   R0,16          REPEAT 16 TIMES
      04D0  0010
0872  04D2  04D9  WTLPL2 CLR  *R9            WRITE LAST 16 BYTES OF
0873                    *                   ID GAP (FIRST BYTE SKIPPED FOR
0874                    *                   BYTE SYNCHRONIZATION)
0875  04D4  0600        DEC  R0
0876  04D6  16FD        JNE  WTLPL2
0877  04D8  D838        MOVB *R8+,@MRKWT    WRITE DATA MARK
      04DA  7F8E
0878  04DC  0200        LI   R0,130         REPEAT 130 TIMES
      04DE  0082
0879  04E0  D678  WTLPL3 MOVB *R8+,*R9       WRITE DATA FIELD
0880  04E2  0600        DEC  R0
0881  04E4  16FD        JNE  WTLPL3
0882  04E6  04D9        CLR  *R9            REWRITE FIRST BYTE OF
0883                    *                   DATA GAP
0884  04E8  2D00        SINC 0              UPDATE SECTOR NUMBER
0885  04EA  1E04        SBZ  SEL            TURN OFF DRIVE
0886  04EC  0605        DEC  R5             DECREMENT SECTOR COUNT
0887  04EE  16C1        JNE  WRITLP         IF NOT 0, WRITE NEXT SECTOR
0888  04F0  045A  WRITRT B    *R10           ELSE, RETURN
      04B6**131C
```

Figure 32. Floppy Disk Control Program (Sheet 24 of 28)

```
0890                    ********************************************************************
0891                    *
0892                    *       COMMAND CONTROL PROGRAM:  FORMAT
0893                    *
0894                    *       THIS COMMAND ENABLES THE OPERATOR TO FORMAT
0895                    *       A NUMBER OF TRACKS BEGINNING AT THE CURRENT TRACK
0896                    *       AND SPECIFYING THE LAST TRACK.  ALL GAPS,
0897                    *       TRACK, ID, AND DATA MARKS, AND TRACK AND
0898                    *       SECTOR NUMBERS ARE WRITTEN.  THE DATA IN THE
0899                    *       DATA FIELDS IS ALL ZEROES.
0900                    *
0901                    *       ENTRY PARAMETERS:       R10 = RETURN ADDRESS
0902                    *
0903  04F2  2DA0  FORMAT AXMT @ENDMSG             PRINT END MESSAGE
      04F4  0094'
      032C**04F2'
0904  04F6  2DA0        AXMT @TKMSG             PRINT TRACK MESSAGE
      04F8  008A'
0905  04FA  D260        MOVB @TKNUM,R9          FETCH TRACK NUMBER
      04FC  80F8
0906  04FE  2EC9        HXM2 R9                 PRINT TRACK NUMBER
0907  0500  2E89        HRC2 R9                 READ LAST TRACK NUMBER
0908  0502  0289        CI   R9,77*256          LEGAL VALUE?
      0504  4D00
0909  0506  14--        JHE  FRMTRT             IF NOT, RETURN
0910  0508  0208        LI   R8,DTAFLD          LOAD DATA FIELD POINTER
      050A  80FF
0911  050C  DE20        MOVB @DTMRK,*R8+        LOAD DATA MARK
      050E  00D1'
0912  0510  0200        LI   R0,64              REPEAT 64 TIMES
      0512  0040
0913  0514  04F8  FFLPL1 CLR  *R8+               CLEAR DATA BUFFER
0914  0516  0600        DEC  R0
0915  0518  16FD        JNE  FFLPL1
0916  051A  2E00        CRCD 0                  CALCULATE THE CRC FOR THE
0917                    *                       DATA FIELD
0918  051C  9809  FRMTLP CB   R9,@TKNUM          LAST TRACK LESS
      051E  80F8
0919                    *                       THAN CURRENT TRACK?
0920  0520  11--        JLT  FRMTRT             IF SO, RETURN
0921  0522  0208  FRMT1  LI   R8,>100            LOAD INITIAL SECTOR
      0524  0100
0922                    *                       VALUE
0923  0526  0207        LI   R7,SECBUF          LOAD SECTOR BUFFER
      0528  80C0
0924                    *                       POINTER
0925  052A  D808  FRIDBL MOVB R8,@SECNUM         UPDATE SECTOR
      052C  80FA
0926                    *                       NUMBER
0927  052E  2DC0        CRCI 0                  CALCULATE CRC FOR ID FIELD
0928  0530  CDE0        MOV  @IDCRC,*R7+        SAVE CRC IN BUFFER
      0532  80FC
0929  0534  0228        AI   R8,>100            INCREMENT SECTOR NUMBER
      0536  0100
```

Figure 32. Floppy Disk Control Program (Sheet 25 of 28)

```
0930  0538  0288          CI    R8,27*256          LAST SECTOR?
      053A  1B00
0931  053C  16F6          JNE   FRIDBL             IF NOT, REPEAT FOR
0932                      *                        NEXT SECTOR
0933  053E  0207          LI    R7,SECBUF          LOAD SECTOR BUFFER
      0540  80C0
0934                      *                        POINTER
0935  0542  0208          LI    R8,>100            LOAD INITIAL SECTOR
      0544  0100
0936                      *                        NUMBER
0937  0546  2D40          DSON  0                  TURN ON DRIVE
0938  0548  0206  FMINDX  LI    R6,DTAWT           DISK DATA WRITE
      054A  7FFE
0939  054C  04E0          CLR   @INDXWT            WRITE 0 AT INDEX PULSE
      054E  7FFA
0940  0550  0200          LI    R0,45              REPEAT 45 TIMES
      0552  002D
0941  0554  04D6  FFLPL2  CLR   *R6                 WRITE REST OF POST-INDEX
0942                      *                        GAP
0943  0556  0600          DEC   R0
0944  0558  16FD          JNE   FFLPL2
0945  055A  D820          MOVB  @TKMRK,@TKMWT      WRITE TRACK MARK
      055C  00D2'
      055E  7F9E
0946  0560  0200  SECTLP  LI    R0,32              REPEAT 32 TIMES
      0562  0020
0947  0564  04D6  FFLPL3  CLR   *R6                 WRITE 32 BYTE GAP
0948  0566  0600          DEC   R0
0949  0568  16FD          JNE   FFLPL3
0950  056A  D820          MOVB  @IDMRK,@MRKWT      WRITE ID MARK
      056C  00D0'
      056E  7F8E
0951  0570  D5A0          MOVB  @TKNUM,*R6         WRITE TRACK NUMBER
      0572  80F8
0952  0574  D58C          MOVB  R12,*R6            WRITE SECOND BYTE
0953  0576  D588          MOVB  R8,*R6             WRITE SECTOR NUMBER
0954  0578  0228          AI    R8,>100            INCREMENT SECTOR NUMBER
      057A  0100
0955  057C  D58C          MOVB  R12,*R6            WRITE FOURTH BYTE
0956  057E  D5B7          MOVB  *R7+,*R6           WRITE CRC1
0957  0580  D5B7          MOVB  *R7+,*R6           WRITE CRC2
0958  0582  0200          LI    R0,17              REPEAT 17 TIMES
      0584  0011
0959  0586  04D6  FFLPL4  CLR   *R6                 WRITE ID GAP
0960  0588  0600          DEC   R0
0961  058A  16FD          JNE   FFLPL4
0962  058C  0204          LI    R4,DTAFLD          LOAD DATA FIELD
      058E  80FF
0963                      *                        IMAGE POINTER
0964  0590  D834          MOVB  *R4+,@MRKWT        WRITE DATA MARK
      0592  7F8E
0965  0594  0200          LI    R0,130             REPEAT 130 TIMES
      0596  0082
0966  0598  D5B4  FFLPL5  MOVB  *R4+,*R6           WRITE DATA AND CRC
0967  059A  0600          DEC   R0
```

Figure 32. Floppy Disk Control Program (Sheet 26 of 28)

```
0968  059C  16FD          JNE  FFLPL5
0969  059E  04D6          CLR  *R6                WRITE PAD BYTE
0970  05A0  0288          CI   R8,>27*256         LAST BYTE?
      05A2  2700
0971  05A4  16DD          JNE  SECTLP             IF NOT, FORMAT NEXT SECTOR
0972  05A6  04D6  PREILP  CLR  *R6                WRITE PRE-INDEX GAP
0973  05A8  1F04          TB   INDEX              UNTIL INDEX
0974  05AA  16FD          JNE  PREILP             PULSE OCCURS
0975  05AC  2E40          TINC 0                  STEP HEAD TO NEXT TRACK
0976  05AE  10B6          JMP  FRMTLP             FORMAT NEXT TRACK
0977  05B0  1E04  FRMTRT  SBZ  SEL                TURN OFF DRIVE
      0506**1454
      0520**1147
0978  05B2  045A          B    *R10               RETURN
0979                ********************************************************
0980                *
0981                *       COMMAND CONTROL PROGRAM:  EXECUT
0982                *
0983                *       THIS COMMAND ENABLES THE OPERATOR TO BEGIN
0984                *       EXECUTION OF A PROGRAM AT ANY LOCATION
0985                *       IN MEMORY.
0986                *
0987                *       ENTRY PARAMETERS:       R8 = ENTRY POINT
0988                *
0989  05B4  0458  EXECUT B    *R8                BRANCH TO ENTRY POINT
      0332**05B4'
0990                ********************************************************
0991                *
0992                *       COMMAND CONTROL PROGRAM:  ENTER
0993                *
0994                *       THIS COMMAND ENABLES THE OPERATOR TO ENTER
0995                *       DATA INTO SEQUENTIAL MEMORY LOCATIONS.
0996                *
0997                *       CALLING PARAMETERS:     R8 = BEGINNING MEMORY
0998                *                                    LOCATION
0999                *
1000  05B6  0209  ENTER  LI   R9,8                LOAD BYTE COUNT
      05B8  0008
      0330**05B6'
1001  05BA  2F00          NLIN 0                  NEW LINE
1002  05BC  2EC8          HXM2 R8                 PRINT FIRST BYTE OF ADDRESS
1003  05BE  06C8          SWPB R8                 REVERSE BYTES
1004  05C0  2FA0          XMIT @BACKSP            BACKSPACE
      05C2  0087'
1005  05C4  2EC8          HXM2 R8                 PRINT SECOND BYTE OF ADDRESS
1006  05C6  06C8          SWPB R8                 RESTORE BYTES
1007  05C8  2ED8  ENTLP   HXM2 *R8                PRINT MEMORY CONTENTS
1008  05CA  2E98          HRC2 *R8                READ AND STORE NEW VALUE
1009  05CC  0588          INC  R8                 UPDATE ADDRESS POINTER
1010  05CE  0609          DEC  R9                 DECREMENT BYTE COUNT
1011  05D0  13F2          JEQ  ENTER              IF 0, NEW LINE
1012  05D2  10FA          JMP  ENTLP              ELSE, FETCH NEXT BYTE
```

Figure 32. Floppy Disk Control Program (Sheet 27 of 28)

```
1014                ********************************************************************
1015                *
1016                *       COMMAND CONTROL PROGRAM:  DUMP
1017                *
1018                *       THIS COMMAND ENABLES THE OPERATOR TO
1019                *       DISPLAY THE CONTENTS OF MEMORY IN
1020                *       HEXADECIMAL FORMAT.
1021                *
1022                *       CALLING PARAMETERS:     R8 = BEGINNING
1023                *                                    ADDRESS
1024                *
1025  05D4  C248    DUMP    MOV   R8,R9             LOAD DEFAULT END
      032E**05D4'
1026                *                               ADDRESS
1027  05D6  2E89            HRC2  R9                READ FIRST BYTE OF
1028                *                               END ADDRESS
1029  05D8  06C9            SWPB  R9                SAVE IN RIGHT BYTE
1030  05DA  2FA0            XMIT  @BACKSP           BACKSPACE
      05DC  0087'
1031  05DE  2E89            HRC2  R9                READ SECOND BYTE OF
1032                *                               END ADDRESS
1033  05E0  06C9            SWPB  R9                SWAP BYTES
1034  05E2  2F00    DUMPLP  NLIN  0                 NEW LINE
1035  05E4  0207            LI    R7,16             LOAD BYTE COUNT
      05E6  0010
1036  05E8  2EC8            HXM2  R8                PRINT FIRST BYTE OF ADDRESS
1037  05EA  06C8            SWPB  R8                REVERSE BYTES
1038  05EC  2FA0            XMIT  @BACKSP           BACKSPACE PRINTER
      05EE  0087'
1039  05F0  2EC8            HXM2  R8                PRINT SECOND BYTE OF ADDRESS
1040  05F2  06C8            SWPB  R8                CORRECT ADDRESS
1041  05F4  2ED8    DMPLP1  HXM2  *R8               PRINT MEMORY CONTENTS
1042  05F6  8209            C     R9,R8             CURRENT ADDRESS = LAST
1043                *                               ADDRESS
1044  05F8  13--            JEQ   DUMPRT            IF SO, RETURN
1045  05FA  0588            INC   R8                INCREMENT ADDRESS
1046  05FC  0607            DEC   R7                DECREMENT BYTE COUNT
1047  05FE  16FA            JNE   DMPLP1            IF NOT 0, PRINT NEXT
1048                *                               BYTE
1049  0600  1F00            TB    RIN               OPERATOR INTERRUPT?
1050  0602  13EF            JEQ   DUMPLP            IF NOT, PRINT NEXT LINE
1051  0604  045A    DUMPRT  B     *R10              ELSE, RETURN
      05F8**1305
1052                        END

0000 ERRORS

ASM/TERM? T
```

Figure 32. Floppy Disk Control Program (Sheet 28 of 28)