MODEL 980 COMPUTER ASSEMBLY LANGUAGE INPUT/OUTPUT

MANUAL NO. 961961-9734
ORIGINAL ISSUE 1 JUNE 1972
REVISED AND REISSUED 1 OCTOBER 1974

TEXAS INSTRUMENTS
INCORPORATED

LIST OF EFFECTIVE PAGES

INSERT LATEST CHANGED PAGES DESTROY SUPERSEDED PAGES

Note: The portion of the text affected by the changes is indicated by a vertical bar in the outer margins of the page.

Model 980 Computer Assembly Language Input/Output (961961-9734)

Original Issue	1 June 1972
Change 2	8 October 1973 (ECN 380749)
Revised and Reissued	1 October 1974 (ECNs 397696 and 397697)

Total number of pages in this publication is 80 consisting of the following:

PAGE NO.	CHANGE NO.	PAGE NO.	CHANGE NO.	PAGE NO.	CHANGE NO.
Title	0				
ii - vi.	0				
1-1 - 1-2	0				
2-1 - 2-36. . . .	0				
3-1 - 3-6	0				
4-1 - 4-22. . . .	0				
App A Div	0				
A-1 - A-2	0				
User's Resp . .	0				
Bus. Reply . . .	0				
Cover Blank . .	0				
Cover	0				

TABLE OF CONTENTS

TABLE OF CONTENTS (Continued)

APPENDIXES

LIST OF ILLUSTRATIONS

LIST OF TABLES

LIST OF TABLES (Continued)

SECTION I

GENERAL INFORMATION

1.1 SCOPE OF MANUAL

This is the second of two manuals describing the Model 980 Computer assembly language, and it is written for the user interested in controlling all input/output (I/O) operations. Section II of this manual describes the two methods of performing I/O with the computer, and contains examples of both methods. Included in Section II is a discussion of interrupts and related programming techniques. Section III discusses some of the considerations applicable to systems programming. Section IV provides a detailed look at I/O programming considerations related to the 733 ASR/KSR data terminals and the Full Duplex EIA Communications Module used to interface the terminals to the computer. The single appendix at the end of the manual contains a list of USASCII characters.

1.2 REFERENCES

The other manual covering the Model 980 assembly language is Model 980 Computer Assembly Language Programmer's Reference Manual. This manual describes both the machine instructions and coding conventions associated with the Model 980 assembly language.

If the user does not want to control all I/O, the Basic System provides two alternate methods of performing I/O. First, if a 980 FORTRAN compiler is part of the Basic System, the Format Editor portion of the compiler can be linked to the user's program to provide easy, formatted I/O. This method is described in Model 980 Computer FORTRAN Library and IAL. The second method is to use the I/O package as described in Model 980 Computer Basic System Use and Operation. This manual contains the information necessary to load and execute a machine language program; or to assemble, load, and execute a stand alone assembly language program. The related software manuals and their respective manual numbers are listed in table 1-1.

Table 1-1. Related Manuals

Manual	Manual No.
Model 980 Computer Assembly Language Programmer's Reference Manual	943013-9701
Model 980 Computer Basic System Use and Operation	961961-9710
Model 980 Computer FORTRAN Library and IAL	961961-9744
Model 980 Computer Programming Card	943000-9701

SECTION II

INPUT/OUTPUT PROGRAMMING

2.1 METHODS

The Model 980 Computer has two types of I/O ports: the data bus and the direct memory access channel (DMAC). The characteristics of the devices attached to these ports, and the methods of performing I/O, vary considerably.

Data bus devices have a relatively slow rate of transfer. Data is moved to or from the device via the computer one word, or one character, at a time. Since only a small amount of data is moved in a given time interval, the software overhead required to handle each word or character is acceptable. The data bus should be used for devices with a slow transfer rate, because the interfaces are less expensive than those required for the DMAC. The 733 ASR/KSR, paper tape reader, and teleprinter are examples of data bus devices.

DMAC devices are relatively fast. Entire buffers of data may be moved to or from these devices without program intervention. The speed of the devices precludes software handling of each word or character as it is moved. Examples of devices that should interface with the DMAC are magnetic tape and disc drives.

Both data bus and DMAC I/O may be performed with or without the use of interrupts. If interrupts are used, the interface issues a signal to the computer each time some action is required by the software. The signal causes the computer to "trap"; that is, rather than executing the next sequential instruction, an instruction at a specific trap location is executed. This instruction is normally a store status and branch instruction (SSB) which transfers control to a program that performs the service requested by the device.

If interrupts are not used, the program performing the I/O operation must periodically test the status of the appropriate device. A change in status is comparable to an interrupt in that both require some action to be taken by the program.

2.2 DATA BUS INPUT/OUTPUT

The data bus provides a path for single word (16-bit) parallel transfers between the computer and external devices. Four data bus ports are standard in the computer; however, the number of ports may be expanded to a maximum of 256. The external devices connected through these ports are addressable in four groups of 64 addresses, and the assignment of a device to an address is completely arbitrary. The computer instruction repertoire includes the Write Direct Single (WDS) and Read Direct Single (RDS) instructions to control data bus operations. The WDS instruction is used to perform

a 16-bit write to the addressed external device and the RDS instruction performs a 16-bit read from the addressed external device.

Both WDS and RDS are two-word instructions, formatted as shown in figure 2-1. Table 2-1 provides a descriptive breakdown of the bit-fields in both words. A data word containing a character and/or control information is associated with each WDS and RDS instruction. The content of the data word depends on the type of instruction (WDS or RDS) and the external device involved in the communications. When a WDS or RDS instruction is executed, the first word of the instruction is applied to the I/O data bus. The device controller addressed by the external register field must then determine if it is ready to transmit or receive, as specified by bit 10 of the first word of the instruction. If the device controller returns an acknowledge signal to the computer signifying the transfer took place, the program continues. If the busy bit in the second word of the instruction is set and the device controller does not return an acknowledge signal, the external device is not ready for the latest WDS or RDS instruction. In this case, the next sequential instruction (commonly a branch to re-try the WDS or RDS instruction) is executed.

Although all I/O device controllers have the capability of being strapped for any external register number, standard assignments have been made for group zero to facilitate documentation. All standard 980 software furnished by Texas Instruments will address the devices with the assigned numbers shown in figure 2-2 and table 2-2.

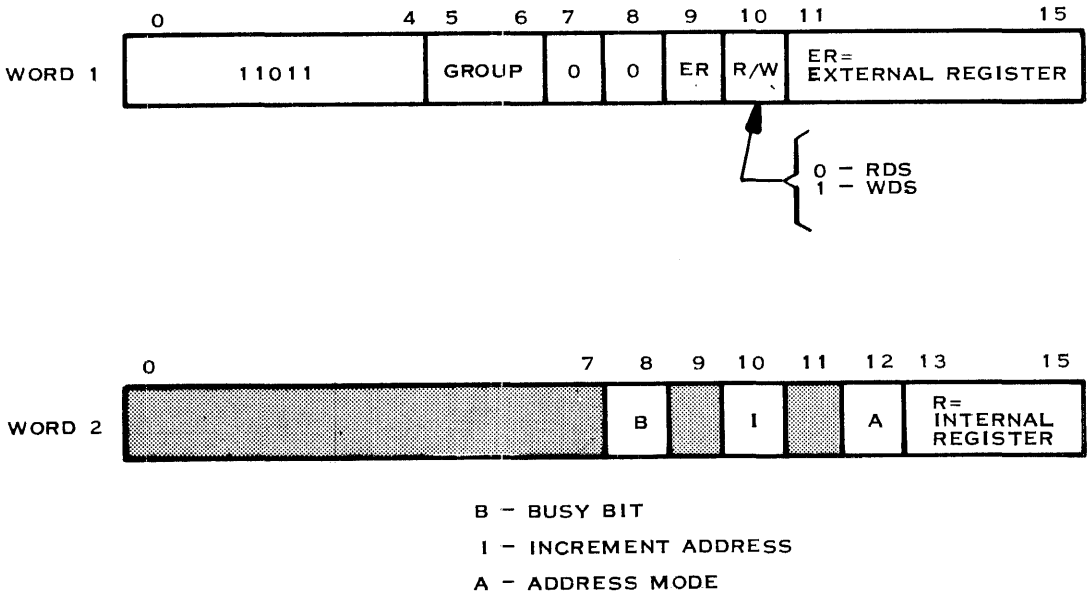


Figure 2-1. WDS/RDS Instruction Format

Table 2-1. WDS/RDS Instruction Field Breakdown

Word/Bit(s)	Description
1/0-4, 7, 8	Op-code bits.
1/5, 6	Group bits, used to define 1 of 4 groups. Each group can be used to specify up to 64 external devices. The group and external register fields combine to specify the external device involved in the WDS/RDS instruction. Typically, the group bits are zero unless the expansion of data bus ports exceeds 64.
1/9, 11-15	External register bits, used to specify 1 of 64 devices in the selected group.
1/10	Read/write bit, used to specify either a RDS or WDS instruction. A logic zero specifies a RDS instruction; a logic one specifies a WDS instruction.
2/0-7, 9, 11	Not used.
2/8	Busy bit, an optional field (B) used with external devices that may not be ready to transfer data when queried by the computer. If this bit is a logic one and a successful data transfer takes place, the instruction following the WDS/RDS instruction is skipped. If this bit is a logic one and no data transfer takes place, the instruction following the WDS/RDS instruction is executed. If this bit is a logic zero, the device transfers data unconditionally and the instruction following the WDS/RDS instruction is executed.
2/10	Increment (I) field, used in conjunction with the A field (bit 12 of word 2) to increment or decrement the contents of the register specified by the R field (bits 13 to 15 of word 2). If both the I and A fields are logic ones, the address contained in the register specified by the R field is incremented by one each time a data word is transferred. If the A field is a logic one, but the I field is a logic zero, the address contained in the register specified by the R is decremented by one each time a data word is transferred.
2/12	Address (A) mode field, used to specify indirect addressing. If the A field is a logic zero, data is transferred to or from the register specified by the R field. If the A field is a logic one, data is transferred to or from the memory location specified by the contents of register R.
2/13-15	Register (R) field, used to specify 1 of 8 internal registers in the computer. The specified register holds the data or the address of data involved in the transfer.

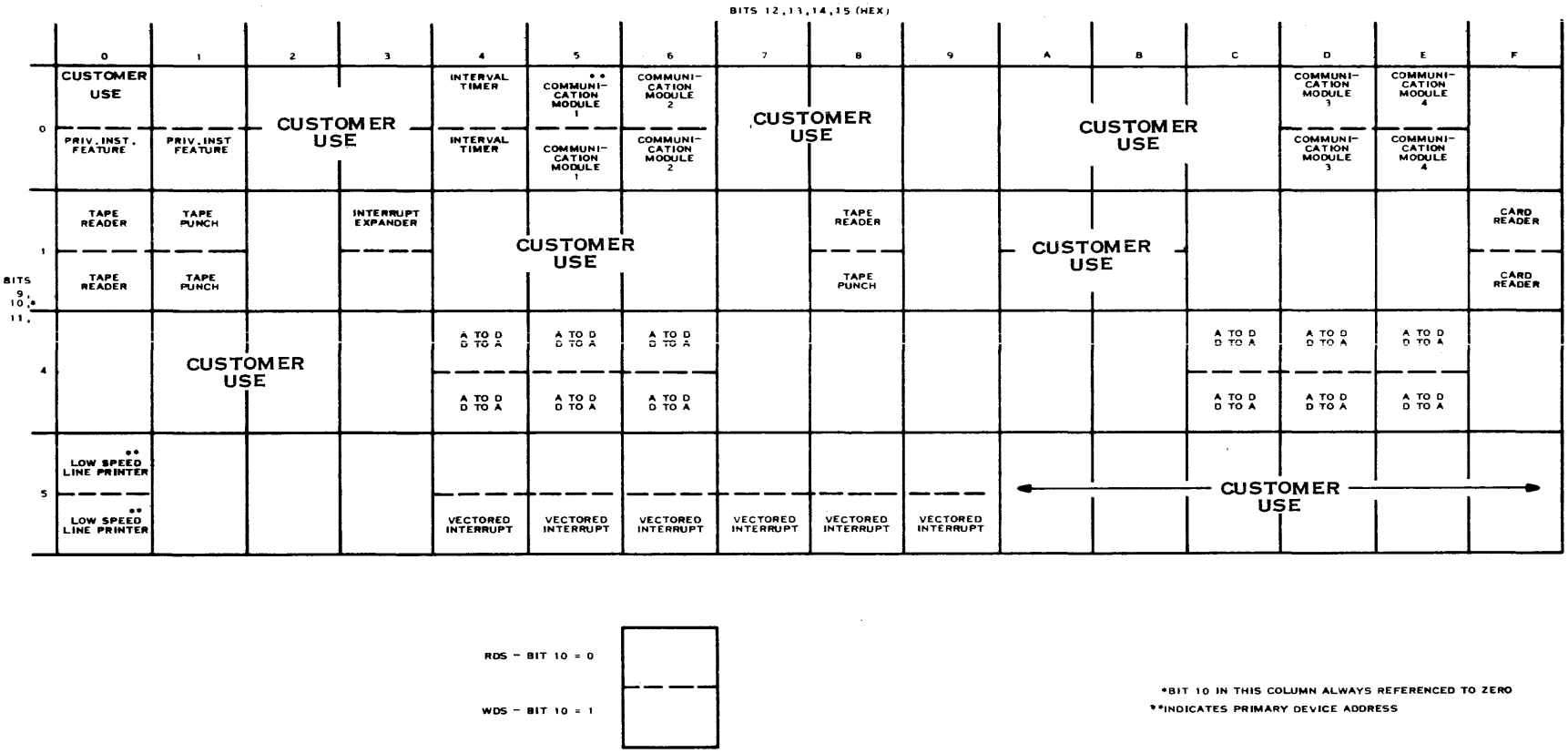


Figure 2-2. Standard 980 External Register Number Assignments

Table 2-2. Standard 980 External Register/Function Details

Register Number (Hex) Bits (9-15)	Device	Function
00	PIF - Privileged Instruction Feature	Out - Set lower memory boundary
01	PIF - Privileged Instruction Feature	Out - Set upper memory boundary
02	Customer Use	
03	Customer Use	
04	Interval Timer	
05	Communication Module (Primary Addr.) 33 ASR, 733 ASR/KSR or CRT New Interface P/N 966637	In - Input data and Status Out - Output data and Commands
06	Communication Module (Secondary Addr.)	In - Input data and Status Out - Output data and Commands
07	Customer Use	
08	Customer Use	
0A	Customer Use	
0B	Customer Use	
0C	Customer Use	
0D	Communication Module (Secondary Addr.)	In - Input data and Status Out - Output data and Commands
0E	Communication Module (Secondary Addr.)	In - Input data and Status Out - Output data and Commands
10	PTR - Paper Tape Reader	In - Read status Out - Write command
11	PTP - Paper Tape Punch	In - Read status Out - Write command

Table 2-2. Standard 980 External Register/Function Details (Continued)

Register Number (Hex) Bits (9-15)	Device	Function
13	INT - Interrupt Expander Address for Expander Chassis Interrupt Expander is relative to board location	In - Read interrupt
14	Customer Use	
15	Customer Use	
16	Customer Use	
18	PTR - Paper Tape Reader	In - Read data
	PTP - Paper Tape Punch	Out - Write data
1A	Customer Use	
1B	Customer Use	
1F	CDR - Card Reader	In - Read status and data Out - Write command
41	Customer Use	
42	Customer Use	
44-46, 4C-4E	ADA - Analog to Digital Converter or Digital to Analog Converter	Data Module - (See ADA User's Manual)
50	LLP - Low Speed Line Printer	Data Module - (See Line Printer User's Manual)
54	Vectored Interrupt (I/O slot 1&2)	Set Mask
55	Vectored Interrupt (I/O slot 1&2)	Clear Mask
56	Vectored Interrupt (I/O slot 1&2)	Set Interrupt (T)

(T) Test Only

Table 2-2. Standard 980 External Register/Function Details (Continued)

Register Number (Hex) Bits (9-15)	Device	Function
57	Vectored Interrupt (I/O slot 3&4)	Set Mask
58	Vectored Interrupt (I/O slot 3&4)	Clear Mask
59	Vectored Interrupt (I/O slot 3&4)	Set Interrupt (T)

(T) Test Only

Programming examples are included in this section for some of the I/O data bus peripheral devices. The 16-bit operands for the RDS and WDS instructions are calculated as follows:

- The assembler puts the correct information in bits 0-4, 7, 8, and 10. Bit 10 designates whether the instruction is an RDS(0) or a WDS(1).
- The external register/function number to be used is designated in bits 9 and 11-15 and is shown in figure 2-2 and listed in table 2-2. The group is zero in all of the examples in the following pages, and is defined in bits 5 and 6.

For example, in order to read a character from the high speed paper tape reader (PTR):

1. Locate PTR input in figure 2-2. The register/function numbers are found to be 10_{16} and 18_{16}.
2. Table 2-2 shows that PTR input on register 10_{16} reads status, and that PTR input on register 18_{16} reads data.
3. In this example, register 18_{16} is the required register/function number. Thus 0018_{16} is the required operand for group zero.
4. If some group other than zero is to be used, the group number must be placed in bits 5 and 6. Thus, for group three, the operand would be 0618_{16}.

The second word of the RDS and WDS examples that follow may specify the device busy test in bit 8. Computer register A (zero) is used as the internal

register in bits 13-15 to receive or transmit data. Bits 10 and 12 are zero to indicate that register A does not contain a memory address nor is it to be incremented. Bits 0 through 7, 9, and 11 are not used. Thus, the second word of an RDS to read a character from the high speed paper tape reader into register A is 0080_{16}. The quantity 2080_{16} may be used for compatibility with previous models of the computer. Most of the examples do include 2000_{16} in the second word of the instruction so that they may be assembled and executed on the 980A or 980B as well as previous models of the computer.

The following paragraphs contain specific examples of common I/O operations; however, these examples do not represent the only way the operations may be performed. For further information, the user should consult the appropriate I/O data bus User's Manual for the Model 980 Computer. Each programming example is followed by a table that explains the appropriate command/status bits for the peripheral device being discussed.

NOTE

The I/O examples included in this section of the manual assume that the mnemonics used in the operand fields have been adequately defined. The mnemonics could be defined as follows:

```
         HED     DATA BUS I/O TEST
A        EOU     0
X        EQU     2
IN       BSS     100
OUT      DATA    ' A B C D E F G H I J'
MINSEC   BSS     1
HOURS    BSS     1
DAYS     BSS     1
          .
          .                    INPUT/OUTPUT CODING
          .

         IDL
         END
```

2.2.1 HIGH SPEED PAPER TAPE READER

The high speed paper tape reader has two external registers: data and command/status. Usually these registers are 18_{16} and 10_{16}, respectively. The command/status register bit definitions are listed in table 2-3.

Table 2-3. High Speed Tape Reader Command/Status Bits

Bit	Commands		Status	
	1	0	1	0
10	Disconnect	No disconnect	Interrupt	No interrupt
11	Interrupts enabled	No action	Interrupts enabled	Interrupts disabled
13	Stop reader	No action	-	-
15	Read forward	No action	-	-

The program listed below turns on the reader, reads 10 characters, and turns off the reader.

```
      .
      .
      .
LDA   =1          BIT 15 TURNS ON HSR
WDS   >10         OUT TO COMMAND REG
DATA  >2080       WITH BUSY BIT TEST
BRU   $-2         REPEAT UNTIL HSR ON
LDX   =-10        SET UP FOR 10 CHAR
RDS   >18         READ CHAR RIGHT ADJ
DATA  >2080       WITH BUSY BIT TEST
BRU   $-2         REPEAT IF NOT READY
STA   IN+10,X     SAVE CHARACTER
BIX   $-4         LOOP BACK
LDA   =4          BIT 13 TURNS OFF
WDS   >10         OUT TO COMMAND REG
DATA  >2000       NO BUSY BIT TEST
      .
      .
      .
```

2.2.2 HIGH SPEED PAPER TAPE PUNCH

The high speed paper tape punch has two external registers: data and command/status. Normally these registers are 18_{16} and 11_{16}, respectively. The command/status register bit definitions are listed in table 2-4.

Table 2-4. High Speed Paper Tape Punch Command/Status Bits

Bit	Commands		Status	
	1	0	1	0
6	Disconnect punch	No disconnect	Interrupt	No interrupt
7	Enable interrupts	No action	Interrupts enabled	Disabled

The following program punches a leader, punches 10 characters and punches a trailer.

```
      .
      .
      .
LDX   =-120       SET UP FOR 12 INCH
LDA   =0          BLANK FRAMES LEADER
WDS   >18         OUT TO HSP
DATA  >2080       WITH BUSY BIT TEST
BRU   $-2         REPEAT IF NOT READY
BIX   $-3         LOOP BACK
LDX   =-10        SET UP FOR 10 CHAR
LDA   OUT+10,X    GET CHAR RIGHT ADJ
WDS   >18         OUT TO HSP
DATA  >2080       WITH BUSY BIT TEST
BRU   $-2         REPEAT IF NOT READY
BIX   $-4         LOOP BACK
LDX   =-120       SET UP FOR 12 INCH
LDA   =0          BLANK FRAMES TRAIL
WDS   >18         OUT TO HSP
DATA  >2080       WITH BUSY BIT TEST
BRU   $-2         REPEAT IF NOT READY
BIX   $-3         LOOP BACK
      .
      .
      .
```

2.2.3 CARD READER

The card reader has one external register, normally $1F_{16}$, used for both status/data and command. The command/status register bit definitions are listed in table 2-5. During a read operation, the least significant 12 bits of the register contain the contents of the last column read, and the most significant four bits contain the status. The data is placed in the register with

Table 2-5. Card Reader Command/Status Bits

Bit	Commands		Status	
	1	0	1	0
0	-	-	Timing error	No error
2	-	-	End of card	No end of card
14	Skip to end of card	No skip	-	-
15	Feed card	No feed card	-	-

Note: Errors that inhibit data transfer (i.e., feed and read errors) are indicated by a light on the card reader; operator intervention is required.

the row 9 in bit 4, row 8 in bit 5, etc. The following is a program to feed a card and read its contents. Note that device status is tested.

```
                .
                .
                .
          LDA   =1          BIT 15 FEEDS A CARD
          WDS   >1F         OUT TO CARD READER
          DATA  >2080       WITH BUSY BIT TEST
          BRU   $-2         REPEAT IF NOT READY
          LDX   =-60        SET UP FOR 60 COL
READ      RDS   >1F         READ A COLUMN
          DATA  >2080       WITH BUSY BIT TEST
          BRU   $-2         REPEAT IF NOT READY
          LRD   12          SEPARATE DATA/STAT
          SZE   A           TEST FOR ERROR
          BRU   ERROR       PROCESS ERROR
          LLD   12          RIGHT ADJ DATA
          STA   IN+60,X     SAVE DATA
          BIX   READ        LOOP BACK
                .
                .
                .
```

2.2.4 DATA MODULE

Many peripherals may be connected to the 980 through the I/O data module interface. The command and status information listed in table 2-6 is for the data module in general. Specific information for a peripheral may be obtained from the appropriate User's Manual furnished by Texas Instruments. The three peripheral devices that may be connected through the data module and have been assigned standard external register numbers are listed in table 2-7.

Table 2-6. I/O Data Module Command/Status Bits

Bit	Commands		Status	
	1	0	1	0
0	*Disconnect	*No disconnect	-	-
1	Enable output acknowledged interrupt	Disable	Output interrupt enabled	Disabled
2	Enable input ready interrupt	Disable	Input interrupt enabled	Disabled
3	-	-	Output interrupt pending	Not pending
4	-	-	Input interrupt pending	Not pending

*Note: The I/O data module Disconnect and No Disconnect commands are defined as follows:

Disconnect - Reset all other command/status bits to zero. I/O may continue but command bits 1 and 2 are not interrogated.

No Disconnect - Command bits 1 and 2 are interrogated and the appropriate action is taken.

Table 2-6. I/O Data Module Command/Status Bits (Continued)

Bit	Commands		Status	
	1	0	1	0
5	-	-	Output word ready for external device	Not ready
6	-	-	Output word acknowledged	-
7	-	-	Input data ready	Not ready
8			Input data received	Not received

Table 2-7. Typical Data Module Peripheral Devices

Device	External Register (Hex)		Function
Line Printer - 132 Column	50 51 52		(See the Line Printer User's Manual for the Model 980 Computer.)
Analog to Digital Converter or Digital to Analog Converter	44 45 46	Module 1	(See the A/D/A User's Manual for the Model 980 Computer. Two data modules are used.)
	4C 4D 4E	Module 2	

The I/O data module interface is designed to transfer data immediately upon command, with two exceptions:

- A busy bit test is required to output a word.
- After a bit is output, another bit WDS may not be issued for four microseconds. (The line marked WAIT in the following example performs no function except to provide a 4.75 microsecond delay.)

Since the busy bit test does not guarantee valid input data, one of several other methods can be used to test data validity as follows:

- Interrupts may be enabled and an interrupt entry point may be defined for the RDS so that the RDS is performed only after an input ready from external device interrupt is received. (See three lines of the example beginning at ENABLE plus two lines at INT.)
- If interrupts are not used, status can be read to be sure the input is ready before a data RDS is issued. (See six lines of example beginning at NOINT.) Note that a status RDS or an output register RDS may be issued at any time.
- A bit of the data word can be assigned (through hardware wiring) to designate input data validity and can be tested after issuing the RDS instruction.

The following example of programming the I/O data module interface is general and does not refer to any of the I/O data module devices that have been assigned standard external register/function numbers. The three registers used in the example are 48_{16} through $4A_{16}$. These registers are currently unassigned in figure 2-2. Register 48_{16} is used for command/status, 49_{16} for data, and $4A_{16}$ for read output/bit control.

```
          .
          .
ENABLE  @LDA   =>6000     ENABLE INTERRUPTS
        WDS    >48        COMMAND
        DATA   >0         REG.A, NO BUSY BIT
        LDA    =7         LOAD REG.A WITH WORD
        WDS    >49        OUTPUT WORD
        DATA   >80        WITH BUSY BIT TEST
        BRU    $-2        REPEAT IF NOT READY
        LDA    =>12       TURN BIT 2 ON
        WDS    >4A        IN OUTPUT REGISTER
        DATA   >0         IF ISSUING ANOTHER BIT WDS,
WAIT    CRA    =16        WAIT 4 MICROSECONDS
NOINT   RDS    >48        READ STATUS INTO
        DATA   >0         REGISTER A
        TABO    4         CHECK INPUT INTERRUPT
        BRU    $-3        REPEAT IF NOT READY
INT     RDS    >49        READ THE INPUT
        DATA   >0         WHEN IT IS READY
        RDS    >4A        READ OUTPUT REGISTER
        DATA   >0         NO BUSY BIT TEST
        TABO   2          SEE IF BIT 2 HAS BEEN
          .               TURNED ON IN THE
          .               OUTPUT REGISTER
          .
```

The format for the WDS data word used by the write bit command involving register $4A_{16}$ of the sample program is shown in figure 2-3.

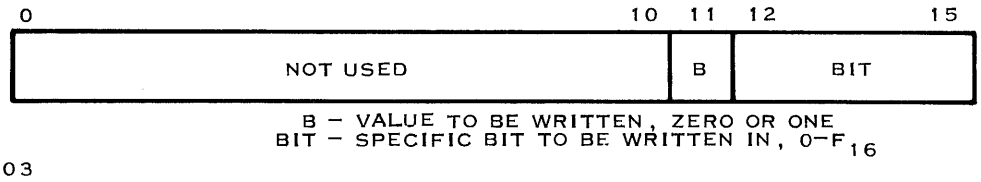


Figure 2-3. WDS Data Word for Write Bit Command

2.2.5 LINE PRINTER

The low-speed line printer has three external registers, numbered 50_{16}, 51_{16}, and 52_{16}. These registers are used for control/status, write/read data, and strobe, respectively. The command/status bit definitions are listed in table 2-8.

Table 2-8. Low Speed Line Printer Command/Status Bits

	Bit	1	0
Control Commands	0	Reset all Control Logic and Flags	No action taken
	1	Enable Interrupts	Disabled Interrupts
	Bits	**07_{16}**	**17_{16}**
Set Bit Commands	11 to 15	Set Strobe On	Set Strobe Off
	Bit	**1**	**0**
Status	1	Interrupts Enabled	Interrupts Disabled
	3	Interrupt	No Interrupt
	6	Printer Requesting Data	Not Requesting Data

ASCII character bit patterns are inverted before being written to the printer. The inverted character sent is placed in bits 9 through 15 of the specified register; bits 0 through 8 must be logic one's.

The strobe is set on and off by an "Output Bit" command. The bit to be set is specified in bits 12 through 15, and the value to be set is put in bit 11 of the specified register. (Note the two uses of the "WDS >52" command in the example below.) No other WDS instruction may be sent for four microseconds following this Output Bit command.

Bytes are loaded into a print buffer before being printed by the printer. The printer will print the contents of the buffer upon receipt of a carriage return.

NOTE

Receipt of the USASCII delete code ($7F_{16}$) will cause contents of the print buffer to be cleared; however, any remaining characters in the record will be entered in the buffer and printed upon receipt of a carriage return, provided another delete code is not received.

The following example will send 10 characters to the printer:

```
          .
          .
          .
          LDX     = -10       SET UP FOR 10 CHAR
          LDA     = 0         SET PRINTER STATUS
          WDS     > 50        GIVE CONTROL COMMAND
          DATA    > 2000      WITHOUT BUSY TEST
LOOP      LDA     OUT+10,X    GET A CHAR
          AND     = >7F       MASK OFF UNWANTED BITS 0 TO 8
          RIV     A,A         INVERT CHAR, PUT ONES IN TOP 9 BITS
          WDS     > 51        OUTPUT CHAR TO PRINTER
          DATA    > 2080      WITH BUSY TEST
          BRU     $-2         REPEAT IF NOT READY
          RDS     > 50        READ STATUS
          DATA    > 2000      WITHOUT BUSY TEST
          TABO    6           TEST FOR DEMAND
          BRU     $-3         REPEAT TILL DEMAND COMES
          LDA     =7          LOAD SET STROBE COMMAND
          WDS     > 52        OUTPUT STROBE
          DATA    > 2000      WITHOUT BUSY TEST
          LLA     13          WAIT 4 MICROSECONDS
          LDA     = >17       LOAD SET STROBE OFF COMMAND
          WDS     > 52        OUTPUT STROBE OFF
          DATA    > 2000      WITHOUT BUSY TEST
          BIX     LOOP        REPEAT
          .
          .
          .
```

2.2.6 MODEL 912 VIDEO DISPLAY TERMINAL

The Model 912 Video Display Terminal interfaces with the computer via the communications module at one of the external register addresses selected from figure 2-2. The single external register assigned to the 912 video terminal handles commands, status, and data in the format described in paragraphs 4.2.1 and 4.2.3. The data field of the WDS format (paragraph 4.2.1)

is used to write USASCII characters to the 912 display and to issue display related commands. The data field of the RDS format (paragraph 4.2.3) is used to read USASCII characters from the 912 keyboard. Figure 2-4 shows the USASCII character codes used by the 912 display (20_{16} to $5F_{16}$) and the display related commands acceptable to the 912 controller.

MOST SIGNIFICANT HEX DIGIT

BITS	7 6 5	0 0 0	0 0 1	0 1 0	0 1 1	1 0 0	1 0 1	1 1 0	1 1 1
4321	HEX	0	1	2	3	4	5	6	7
0000	0			℔	0		P		
0001	1			!	1	A	Q		
0010	2	HOME		"	2	B	R		
0011	3			#	3	C	S		
0100	4			$	4	D	T		
0101	5			%	5	E	U		
0110	6			&	6	F	V		
0111	7	BEL	CLR EOL	'	7	G	W		
1000	8	←		(	8	H	X		
1001	9			)	9	I	Y		
1010	A	↓	↑	*	:	J	Z		
1011	B			+	;	K	|		
1100	C		→	,	<	L	\		
1101	D	RE-TURN		-		M	]		
1110	E			.	>	N	^		
1111	F		CLEAR	/	?	O	_		

LEAST SIGNIFICANT HEX DIGIT

HOME–MOVE CURSOR TO FIRST POSITION OF FIRST LINE

CLEAR–CLEAR DISPLAY AND HOME CURSOR

CLR EOL–CLEAR CURRENT LINE AND RETURN CURSOR TO FIRST POSITION OF CURRENT LINE

↑,↓–SINGLE LINE CURSOR CONTROLS

→,←–SINGLE CHARACTER POSITION CURSOR CONTROLS

BEL–ACTUATE AUDIBLE ALARM (≈1/2 SECOND)

(A)129604

Figure 2-4. 912 Characters and Controls

The following simplified programming example writes a four-character message to the 912 display, assuming external register 6_{16} is used to interface the computer with the associated communications module.

```
          :
          :
          LDX     =-4
WRITE     LDA     OUT+4,X   LOAD CHARACTER
          AND     =>7F      MASK OUT BITS 0-8
          @IOR    =>5000    WRITE DATA READY, CLEAR WRITE REQUEST
          WDS     6         WRITE TO 912
          DATA    >80
          BRU     $-2
          BIX     WRITE     WRITE NEXT CHARACTER
          BRU     CONT      CONTINUE
OUT       DATA    ' H E L P '
          :
          :
```

2.3 DIRECT MEMORY ACCESS CHANNEL

The direct memory access channel (DMAC) is the high speed I/O portion of the computer. One direct memory access port, capable of I/O with one peripheral controller, is included in the basic computer. A direct memory access port expander, however, may be added to the DMAC to allow for up to eight high-speed peripheral controllers. With a single peripheral device, the DMAC can attain a maximum burst rate of 10^6 words per second; when memory access must be granted to more than one peripheral device, the effective maximum transfer rate is 8×10^5 words per second. When the DMAC expander interfaces with multiple high speed devices, a priority scheme selected by the user resolves two or more simultaneous access requests. All access requests from DMAC related peripheral devices have priority over the computer when both require memory access at the same time.

All DMAC I/O is controlled by the Automatic Transfer Instruction (ATI). ATI is a two-word instruction, formatted as shown in figure 2-5. When data must be transferred between computer memory and a device controller on the DMAC, the ATI instruction is used. The first word of the instruction

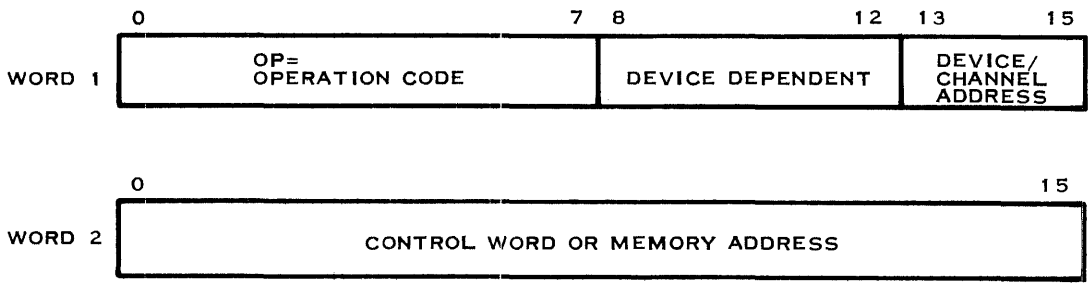


Figure 2-5. ATI Instruction Format

format addresses one of eight possible device controllers (bits 13 to 15) and supplies any necessary device dependent data (bits 8 to 12). The second word of the format is routed to the addressed device controller where it is interpreted as a single word functional command or as an address pointing to a list in memory containing command related data. Command lists typically contain one or more blocks of data, each constructed from the following elements:

- Command Function - Operation to be performed, such as read or write.
- Buffer Address - Memory address for data transfer.
- Buffer Length - Character or word length of record to be transferred.
- External Address - Address within external media of data for transfer.
- Interrupt Enable - Instructs device controller whether or not to generate an interrupt upon completion of data transfer.
- Chaining Information - Linkage to next command list, if any.

In addition to the ATI instruction and any related command lists, the DMAC device controllers interface with the computer by reporting status to dedicated memory locations. The dedicated memory location associated with a device depends on the DMAC port number used to address the device. Table 2-9 lists the port numbers and related memory locations.

Table 2-9. DMAC Device Status Memory Locations

DMAC Port	Memory Location (Hex)
0	98, 99
1	9A, 9B
2	9C, 9D
3	9E, 9F
4	A0, A1
5	A2, A3
6	A4, A5
7	A6, A7

The following is a typical example of code to perform an operation with a DMAC device. The device status location is set to some illegal value (usually all zeros), the ATI is issued, and then a test is performed repeatedly looking for a change in status. This change indicates the operation is complete. The status should then be tested to determine whether or not an error was detected during performance of the operation. The ATI specifies the device port and the command to be executed.

```
      .
      .
      .
LDA   =0          INITIAL VALUE
STA   *STATUS     STORE IN STATUS LOC
ATI   DEVICE      ISSUE ATI
DATA  LIST        ATI LIST POINTER
LDA   *STATUS     OBTAIN STATUS
SNZ   A           TEST FOR CHANGE
BRU   $-2         REPEAT IF NO CHANGE
CPL   LEGAL       TEST FOR OK STATUS
SEQ               CONTINUE IF OK
BRU   ERROR       IF NOT PROCESS ERR
      .
      .
      .
```

The next four paragraphs cover programming details for four types of DMAC devices. For further information, consult the Model 960/980 Computer DMAC Controller manual.

2.3.1 LINE PRINTER

The high speed line printer normally occupies DMAC expander port 5 (if a DMAC expander is used), and it stores status in memory location $00A2_{16}$. The line printer command list words format is shown in figure 2-6 and the returned status words are detailed in table 2-10.

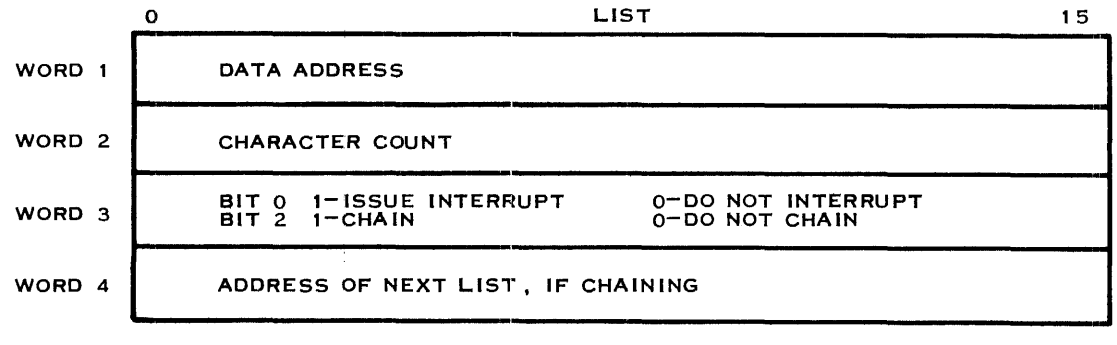


Figure 2-6. Line Printer Command List Words Format

Table 2-10. Line Printer Status Word Bits

	Bit	1	0
Status Word 1	12	Controller busy	Not busy
	14	Printer not ready	Ready
	15	Operation complete	Not complete

Note: Status word 2 not used.

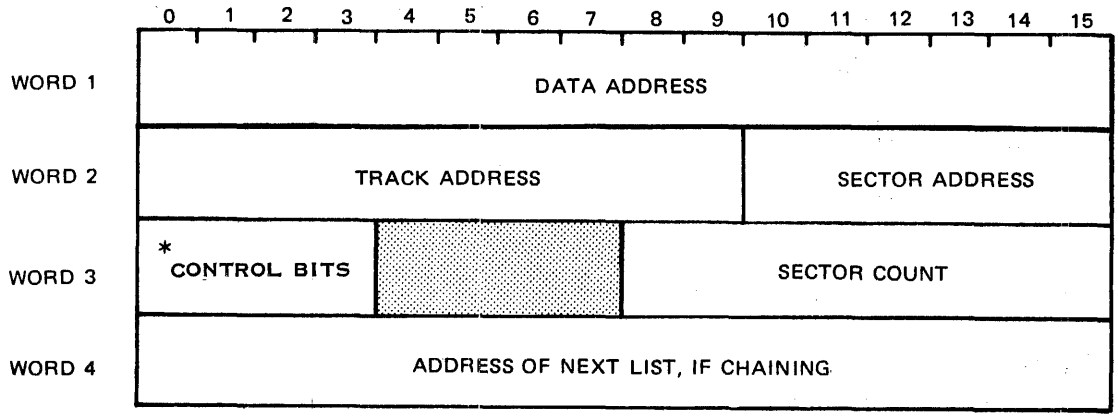
The following programming example prints 100 characters on the line printer from reserved memory named BUFFER. The write is to be done without interrupts or chaining.

```
                .
                .
          LDA   =0
          STA   *STATUS     CLEAR STATUS
          ATI   5           ISSUE ATI
          DATA  LIST        ATI LIST POINTER
          LDA   *STATUS     OBTAIN STATUS
          SNZ   A           TEST FOR CHANGE
          BRU   $-2         REPEAT IF NO CHANGE
          CPL   =1          TEST FOR OPERATION COMPLETE
          SEQ               CONTINUE IF COMPLETE
          BRU   ERROR       BRANCH TO ERROR ROUTINE
          BRU   CONT        CONTINUE
STATUS    DATA  >A2
LIST      DATA  BUFFER,100,0,0
BUFFER    BSS   50
                .
                .
                .
```

2.3.2 FIXED-HEAD DISC

The fixed-head disc controller normally occupies DMAC expander port 0 (if a DMAC expander is used) and stores status in memory locations 0098_{16} and 0099_{16}. The fixed-head disc command list words format is shown in figure 2-7 and the returned status words are detailed in table 2-11.

The first ATI word for the fixed-head disc controller specifies unit number as well as port number. The unit number is specified in bits 10 and 11. A maximum of four discs may be attached to the computer, but a command cannot be issued to one unit while another unit is in operation. The sample



* CONTROL BIT BREAKDOWN

Bit	1	0
0	Issue interrupt	Do not interrupt
1	Write	Read
2	Chain	Do not chain
3	Compare disc with memory	No compare

(A)129607

Figure 2-7. Fixed-Head Disc Command List Words Format

Table 2-11. Fixed-Head Disc Status Word Bits

	Bit	1	0
Status Word 1	0	Next chain list taken	Not yet taken
	8	Timing error	No error
	9	Program error	No error
	10	Compare error	No error
	11	Parity error	No error
	12	Controller busy	Not busy
	13	Unit not ready	Ready
	14	Write lockout enabled	No write lockout
	15	Operation complete	Not yet complete

Note: Status word 2 - address of last list executed.

ATI is for disc number 3 on port 0. The list specifies that three sectors are to be read from track one, beginning at sector two, and the words read are to be placed in the reserved space named BUFFER. The read is to be done without interrupts, chaining, or comparing.

```
                .
                .
                .
        ATI     >30
        DATA    LIST
                .
LIST    DATA    BUFFER,>42,3,0
BUFFER  BSS     96
                .
                .
                .
```

A combination write/compare command may be issued to the disc; however, the read command and a compare only command are issued separately. Average access time to the disc is 8.7 milliseconds. When the compare is issued in conjunction with the write, one extra disc revolution is required. An average disc revolution takes 17.5 milliseconds.

The fixed-head disc available on the 980 computer contains 32 words per sector. The physical structure of the disc sectors and tracks is not covered in this manual, but from a programming viewpoint, the disc contains 56 sectors per track, and from 32 to 1024 tracks. For purposes of disc addressing only, the logical layout of the disc may be thought of as shown in figure 2-8.

Each even-odd track pair, tracks 0 and 1, 2 and 3, etc., requires one disc revolution. The optional write lockout is accomplished manually through a set of 16 switches and cannot be set under program control. Lockout and timing details not covered in this paragraph can be found in the Direct Memory Access Channel Controller User's Manual.

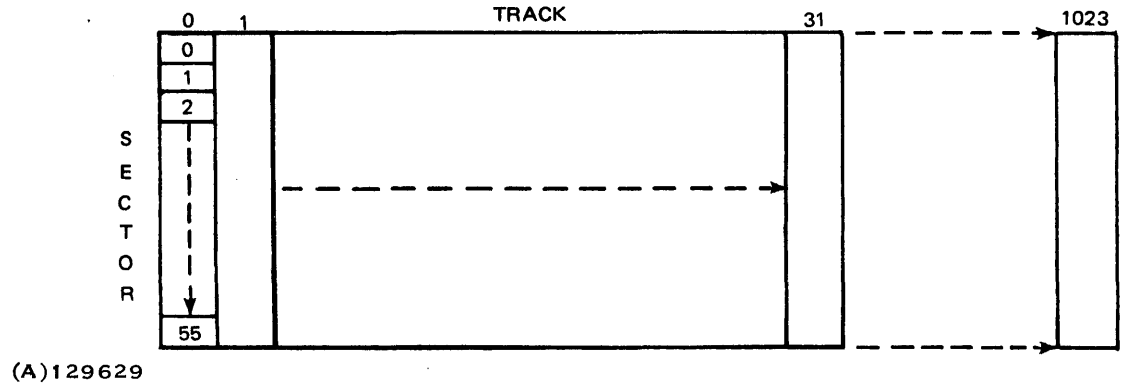


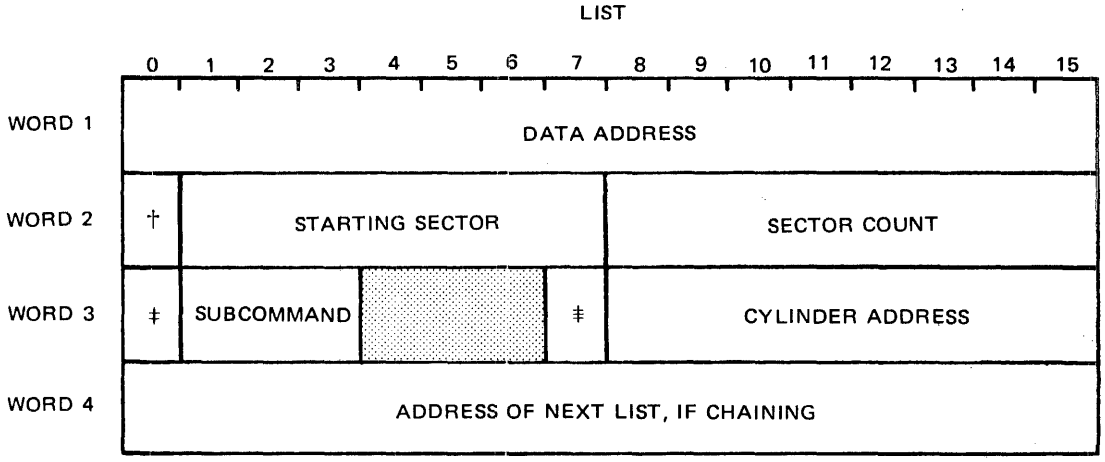
Figure 2-8. Logical Layout of Fixed-Head Disc

2.3.3 MOVING-HEAD DISC

The moving-head disc controller normally occupies DMAC expander port 1 (if a DMAC expander is used) and stores status in memory locations $009A_{16}$ and $009B_{16}$.

In addition to specifying the port, the first ATI word for the moving-head disc controller specifies the unit number in bits 9 and 10. It also specifies that the controller should clear itself (bit 8=1), or acquire a command list and perform a subcommand (bit 8=0). A maximum of four units may be attached to the controller; however, in general, a command may not be issued to one unit while another unit is in operation. The only exception is that a command is accepted for one unit while another is doing an independent seek.

The moving-head disc command list words format is shown in figure 2-9. The subcommand field in word 3 of the list is detailed in table 2-12, and table 2-13 lists the status words returned to the computer.



LIST BIT	MEANING 1	0
†	Issue interrupt	Do not interrupt
‡	Chain	Do not chain
‡	Use head 1	Use head 0

(A)129608

Figure 2-9. Moving-Head Disc Command List Words Format

Table 2-12. Moving-Head Disc Subcommands

List Word 3 Bits 1-3	Subcommand
000	Read data
001	Read i.d.
010	Compare disc and memory
011	Perform independent seek
100	Write data
101	Write i.d. without protect
110	Write i.d. with protect
111	Write data protect override

Table 2-13. Moving-Head Disc Status Word Bits

	Bit	1	0
Status Word 1	0	Operation complete	Not yet complete
	1	Write protect error	No error
	2	Unit off-line	On line
	3	Data transfer error	No error
	4	End-of-disc error	No error
	5	Program error	No error
	6	Compare error	No error
	7	Parity error	No error
	8	ID compare error	No error
	9	Device busy	Not busy
	10	Chain list taken	Not yet taken
	11	Write protect on	Off
	12	Seek complete unit 3	Not yet complete
	13	Seek complete unit 2	Not yet complete
	14	Seek complete unit 1	Not yet complete
	15	Seek complete unit 0	Not yet complete

Note: Status word 2 - address of last list executed.

A new disc pack must be initialized by writing an i.d. in every disc sector. An i.d. may be written with a protect bit set (one) or reset (zero) as specified in the write i.d. command. If set, a write data with protect override is the only command which allows data to be written into a specified sector. From the programmers viewpoint of addressing the moving-head disc, the logical layout may be thought of as shown in figure 2-10.

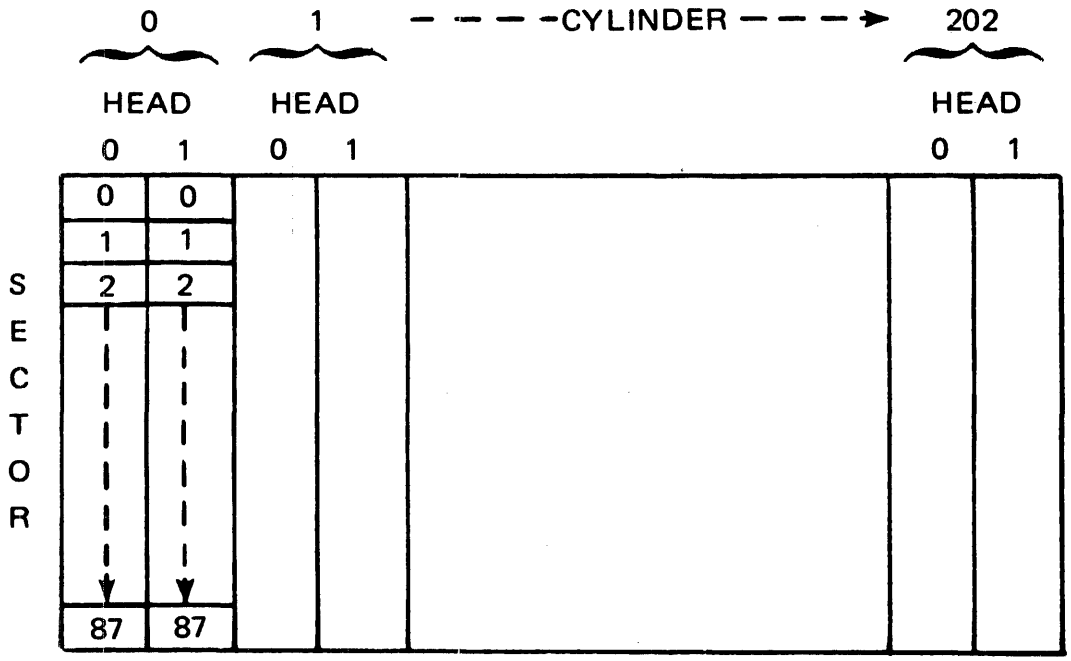


Figure 2-10. Logical Layout of Moving-Head Disc

The following programming example reads 16 sectors on moving-head disc 3, starting at sector 24 of head 1 on cylinder 51. Following the read, a check is made to determine if the operation is complete. Interrupts and chaining are not used.

```
                    .
                    .
                    .
                LDA     =0
                STA     *STATUS     CLEAR STATUS
                ATI     >41         ISSUE ATI
                DATA    LIST        ATI LIST POINTER
                LDA     *STATUS     OBTAIN STATUS
                SNZ     A           TEST FOR CHANGE
                BRU     $-2         REPEAT IF NO CHANGE
                CPL     LEGAL       TEST FOR READ COMPLETE
                SEQ                 CONTINUE IF COMPLETE
                BRU     ERROR       BRANCH TO ERROR ROUTINE
                BRU     CONT        CONTINUE
STATUS          DATA    >9A
LIST            DATA    BUFFER,>1810,>0133,0
LEGAL           EQU     >8000
BUFFER          BSS     512
                    .
                    .
                    .
```

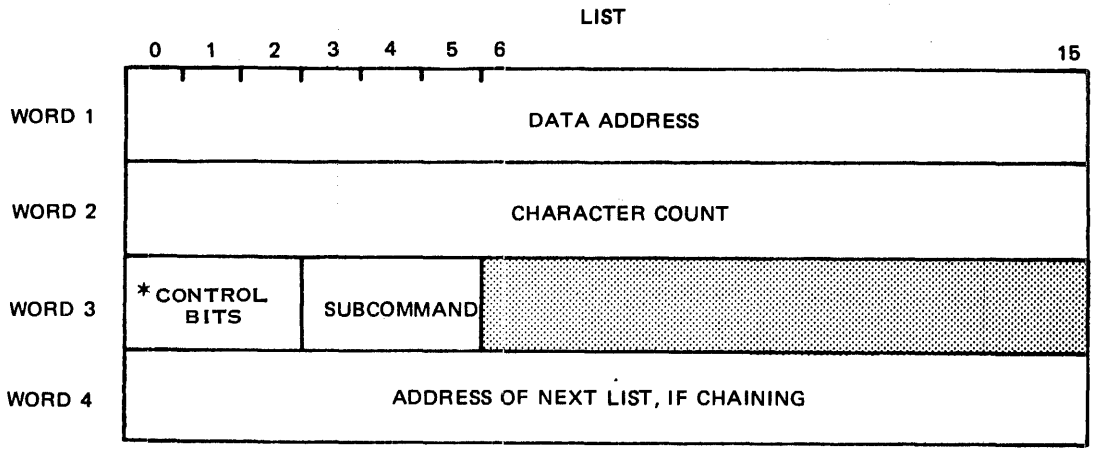
2.3.4 MAGNETIC TAPE

The 800-bpi, 9-track magnetic tape controller normally occupies DMAC expander port 2 (if a DMAC expander is used) and stores status in memory location $009C_{16}$. In addition to specifying the port, the first ATI word to a magnetic tape controller specifies a command in bits 8 and 9 and a unit in bits 11 and 12. Bits 11 and 12 are also used to reset the interface. Table 2-14 lists the commands and unit numbers associated with the various bit patterns.

Table 2-14. Magnetic Tape Command and Unit Designations

ATI Bits 8 and 9	Command	ATI Bits 11 and 12	Unit
00	Rewind selected unit with interrupt	00	Reset
01	Unload selected unit with interrupt	01	Select unit 1
10	Rewind unit without interrupt	10	Select unit 1
11	Perform subcommand (acquire list)	11	Select unit 3

In general, if an operation is in progress on one unit, it is not possible to issue a command for another unit. The only exception is that a command is accepted for one unit while another is being rewound. The magnetic tape command list words format is shown in figure 2-11. The subcommand field in word 3 of the list is detailed in table 2-15 and table 2-16 lists the status words returned to the computer.



CONTROL BIT BREAKDOWN

BIT	1	0
0	ISSUE INTERRUPT	DO NOT INTERRUPT
1	REVERSE MOTION	FORWARD MOTION
2	CHAIN	DO NOT CHAIN

(A)129610

Figure 2-11. Magnetic Tape Command List Words Format

Table 2-15. Magnetic Tape Subcommands

List Word 3 Bits 3-5	Subcommand
000	Read a record
001	Skip a record
100	Write a record
101	Erase forward
110	Write end-of-file

Table 2-16. Magnetic Tape Status Word Bits

	Bit	1	0
Status Word 1	0	Operation complete	Not complete
	1	Write ring error	No error
	2	Unit off-line	On-line
	3	Controller busy	Not busy
	4	DMAC parity error	No error
	5	Tape parity error	No error
	6	End-of-record detected	No end-of-record
	7	End-of-file detected	No end-of-file
	8	End-of-tape detected	No end-of-tape
	9	Beginning-of-tape detected	Not detected
	10	Transfer error	No error
	11	Rewind complete unit 1	Not complete
	12	Rewind complete unit 2	Not complete
	13	Rewind complete unit 3	Not complete
	14	Command error	No error
	15	Chain list taken	Not yet taken

Note: Status word 2 - After read or write is completed or aborted, remaining character count is stored; for commands other than read or write, address of last list executed.

The following programming example writes a 500-character record on magnetic tape unit 1 from reserved memory named BUFFER. Following the write, a check is made to determine if the operation is complete. Chaining is not used but the magnetic tape is to issue an interrupt when the write is complete.

```
                .
                .
                .
          LDA   =0
          STA   *STATUS       CLEAR STATUS
          ATI   >CA           ISSUE ATI
          DATA  LIST          ATI LIST POINTER
          LDA   *STATUS       OBTAIN STATUS
          SNZ   A             TEST FOR CHANGE
          BRU   $-2           REPEAT IF NO CHANGE
          CPL   LEGAL         TEST FOR WRITE COMPLETE
          SEQ                 CONTINUE IF COMPLETE
          BRU   ERROR         BRANCH TO ERROR ROUTINE
          BRU   CONT          CONTINUE
STATUS    DATA  >9C
LIST      DATA  BUFFER,500,>9000,0
LEGAL     EQU   >8000
BUFFER    BSS   250
                .
                .
                .
```

2.4 PROGRAMMING WITH INTERRUPTS

There are four types of interrupts in the Model 980 Computer. These interrupts, in order of priority, are as follows: internal interrupt, vectored interrupt option, DMAC interrupt, and data bus interrupt. It should be noted that no interrupts other than internal interrupts are recognized for one instruction following a WDS, RDS, LSB, LSR, SSB, or a register-to-register instruction which has the status register as the destination register. The status register (SR), which is constantly referenced in this section, is covered in detail in Model 980 Computer Assembly Language Programmer's Reference Manual.

Certain fixed memory addresses are reserved in the computer for preassigned functions. The various reserved hexadecimal locations and their functions are described in table 2-17.

2.4.1 DATA BUS INTERRUPTS

When the computer accepts a data bus interrupt signal from an external device, it causes a program trap to memory location 0006_{16}. A 4-bit data bus interrupt expander (not to be confused with the vectored interrupt option) is standard on the 980. It is possible to have a data bus interrupt expander for each of the four groups of 64 external registers addressable via the data bus. An interrupt expander occupies one of the 64 external addresses. When read by the program with an RDS instruction, it yields the true-false status of up to 16 interrupt signals which have been input to the interrupt expander from other devices. The interrupt expander generates a data bus interrupt if it receives an interrupt signal from one or more of its input devices. Interrupts from the interrupt expander are recognized by examining the bits for

Table 2-17. Reserved Memory

Location (Hex)	Function
0-1	Power restore start-up
2-3	Internal interrupt
4-5	DMAC interrupt
6-7	Data bus interrupt
8-47	Interrupts when using vectored interrupt option
48-87	Reserved
88	*Address of first logical device table entry
89	*Address of first physical device table entry
8A	*Address of first device name table entry
8B	*Address of floating point package
8C	*Status flag for operator communication package
8D-93	*Reserved
94	*MP/PIF upper limit address (absolute)
95	*MP/PIF lower limit address (absolute)
96	DMAC interrupt status
98-99	Status from device on DMAC port 0
9A-9B	Status from device on DMAC port 1
9C-9D	Status from device on DMAC port 2
9E-9F	Status from device on DMAC port 3
A0-A1	Status from device on DMAC port 4
A2-A3	Status from device on DMAC port 5
A4-A5	Status from device on DMAC port 6
A6-A7	Status from device on DMAC port 7

*Function assigned this address by the Basic System software.

logic one's. Bit 1 equal to one corresponds to an interrupt from the module in slot I01, bit 15 set to one corresponds to an interrupt from the module in slot I015 (assuming internal I/O expansion). For more information, see the Model 980 Computer Input/Output Manual (part no. 960964-9701). Data bus interrupts are lower in priority than other interrupts. Therefore, if the

programmer wishes to inhibit other interrupts while processing a data bus interrupt, the interrupt mask bits in the status register must be adjusted. It is not possible to inhibit internal interrupts.

It is a common programming technique to put an SSB instruction in location 0006_{16}. If an instruction other than an SSB or an instruction affecting the PC register (such as a branch or skip) is placed in location 0006_{16}, then only the single instruction is executed. Since the PC is held fixed for execution of the one instruction, program control is immediately returned to the interrupted program.

CAUTION

If it is necessary to use an instruction that does not affect the PC at location 0006_{16}, do not use an instruction that does not change the Memory Address (MA) register. The MA is used to retrieve the trap instruction, so if the trap instruction does not modify the MA, the MA will not be updated to the current PC value at instruction termination.

Further interrupts are prohibited until after execution of one instruction subsequent to that instruction in location 0006_{16}. In other words, the SSB instruction and the first instruction it branches to (often an LSB), are both performed before another interrupt of any kind can be honored by the computer. Immediately upon completion of the SSB instruction in location 0006_{16}, and prior to the instruction to which it branches, the content of the SR is automatically modified to inhibit data bus interrupts (bits 4, 5, 6, 7, and 9 are cleared to zero).

The data bus interrupt sequence is as follows:

1. The instruction in location 0006_{16} is performed. Normally, this instruction should be an SSB. The values stored for SR and PC are those which existed prior to the interrupt response.
2. The subroutine referenced by the SSB determines which devices require attention.
3. For each device requiring service, a sequence of instructions is executed which is peculiar to the specific device interface. Many devices require that the program read a 16-bit status word with an RDS instruction. The act of reading the status serves as an acknowledgment to the device that the program has seen the interrupt. The device will then remove the interrupt request.
4. When all interrupt processing is complete, the interrupted program is reentered with an LSB instruction.

NOTE

Status register bit 7 must be on for future data bus interrupts to be recognized.

2.4.2 DMAC INTERRUPTS

When the computer accepts an interrupt signal from one or more DMAC device controllers, it causes a program to trap to memory location 0004_{16}. If more than one device is attached to the DMAC, a DMAC expander must be used. The DMAC expander stores a status word at location 0096_{16}, which identifies the device controller(s) requiring service. In any case, the device controller itself will store status at a memory location as defined in table 2-17.

As with data bus interrupts, it is a common programming practice to put an SSB instruction in location 0004_{16}. If an instruction other than an SSB or an instruction affecting the PC register (such as branch or skip) is placed in location 0004_{16}, then only the single instruction is executed. Since the PC is held fixed for execution of the one instruction only, program control is immediately returned to the interrupted program.

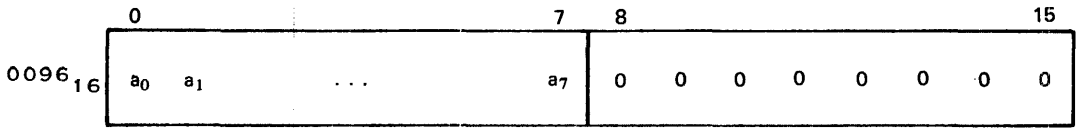
CAUTION

If it is necessary to use an instruction that does not affect the PC at location 0004_{16}, do not use an instruction that does not affect the MA register. The MA is used to retrieve the trap instruction, so if the trap instruction does not modify the MA, the MA will not be updated to the current PC value at instruction termination.

Further interrupts are prohibited until after execution of one instruction subsequent to that instruction in location 0004_{16}. In other words, the SSB instruction and the first instruction it branches to (often an LSB), are both performed before another interrupt of any kind can be honored by the machine. Immediately upon completion of the SSB instruction in location 0004_{16}, and prior to the execution of the instruction to which it branches, the content of the SR is automatically modified to inhibit both DMAC and data bus interrupts (bits 4, 5, 6, 7, 9, and 12 are cleared to zero).

The DMAC interrupt sequence occurs as follows:

1. The main program is suspended.
2. Location 0096_{16} in memory is replaced by a word that indicates which device controllers require service (if the DMAC expander is used).



Location 0096_{16} contains a field of eight bits, a_0 to a_7, where a_i is the service request bit for the device on port i. A one represents a service request is pending.

3. Locations 0098_{16} +2i and 0099_{16} + 2i are replaced with device status information, if any, from the device controller.

4. The instruction in location 0004_{16} is performed. Normally this instruction should be an SSB. The values stored for SR and PC are those which existed prior to the interrupt response.

5. For each device controller requiring service, a program sequence examines the device status. It must then perform whatever processing may be required.

6. The interrupt sequence may terminate with an LSB instruction which restores control to the interrupted program.

NOTE

Bit 12 of the status register must be turned on for future DMAC interrupts to be recognized.

2.4.3 VECTORED (PRIORITY) INTERRUPT OPTION

The methods of servicing I/O interrupts discussed thus far all involve considerable software activity just to decide which of several devices caused an interrupt. Only after this preliminary activity can the actual interrupt servicing begin. For most devices this software overhead is acceptable. However, it may happen that a device requires more rapid servicing than is attainable in this way. In such instances, the vectored (priority) interrupt option should be used. Use of the vectored interrupt option allows each device to have a unique interrupt trap location. The software can then be organized such that execution reaches a given trap location only if the associated device issues an interrupt. Thus, when a trap occurs, immediate action can be taken to service the interrupt.

NOTE

All standard I/O devices supplied by Texas Instruments have a preassigned interrupt structure. Standard Texas Instruments software does not make use of the vectored interrupt option.

The vectored interrupt option allows 8, 16, 24, or 32 separate trap locations. All interrupts have priorities, and the vectored interrupts are placed in the priority list as follows:

1. Internal interrupts
2. Optional vectored interrupts in trap locations

   0008_{16} (highest priority)
   $000A_{16}$
   .
   .
   .
   0046_{16} (lowest priority)

3. Regular DMAC interrupts
4. Regular data bus interrupts

After receipt of an interrupt, a lower priority interrupt is not honored until the original interrupt has been reset; however, an interrupt of higher priority is honored. Thus, while processing the first interrupt, but before resetting that interrupt, a second interrupt with higher priority can cause another trap to occur. After the second interrupt is processed, execution of an LSR instruction resets the higher priority interrupt and returns control to the program processing the first interrupt.

The recommended method of implementing these vectored interrupt-processing programs is as follows:

- Use an SSB instruction in the trap location to branch to the interrupt-processing program.
- Be sure that the last instruction executed in the interrupt-processing program is an LSR, which resets the interrupt and returns control to the interrupted program.

If an instruction other than an SSB or an instruction affecting the PC register (such as branch or skip) is placed at the trap location, then only the single instruction is executed. Since the PC is held fixed for execution of the one instruction, program control is immediately returned to the interrupted program.

CAUTION

If it is necessary to use an instruction that does not affect the PC at the trap location, do not use an instruction that does not affect the MA register. The MA is used to retrieve the trap instruction, so if the trap instruction does not modify the MA, the MA will not be updated to the current PC value at instruction termination.

Further interrupts are prohibited until after execution of one instruction subsequent to that instruction at the trap location. In other words, the SSB instruction and the first instruction it branches to, are both performed before another interrupt of any kind can be honored by the machine. Immediately upon completion of the SSB instruction, and prior to the execution of the instruction to which it branches, the content of the SR is automatically modified to inhibit ordinary data bus and DMAC interrupts (bits 4, 5, 6, 7, and 12 of the SR are cleared to zero).

In addition to the 32 priority levels provided, the vectored interrupt option includes interrupt masking control and programmed interrupts. Associated with each vectored interrupt are three control bits that can be used to enable, disable, or simulate the interrupt. These three control bits for 16 interrupts are grouped together to form three 16-bit registers on the I/O data bus. Since the maximum number of vectored interrupts is 32, six I/O data bus register addresses must be allocated to the mask and programmed interrupt functions. The I/O data bus addresses used in conjunction with WDS and RDS instructions concerned with the various vectored interrupt levels are shown in table 2-18.

Table 2-18. Vectored Interrupt System External Registers

Vectored Interrupts 0 to 15	Vectored Interrupts 16 to 31	Register Function
54_{16}	57_{16}	Set Mask
55_{16}	58_{16}	Clear Mask
56_{16}	59_{16}	Set Interrupt

The set mask registers are used to mask (inhibit) interrupts; the clear mask registers are used to unmask interrupts; the set interrupt registers are used to create programmed interrupts. For example, a logic one placed in bit 6 of the data word of a WDS 54_{16} instruction would mask interrupt 6. The same data word could be used to unmask interrupt 6 with a WDS 55_{16}, or create an interrupt 6 with a WDS 56_{16}.

NOTE

Status register bit 8 must be on for the priority interrupt to be recognized.

SECTION III

SYSTEM PROGRAMMING CONSIDERATIONS

3.1 INTERNAL INTERRUPT

An internal interrupt occurs as a result of detection of imminent power failure or detection of an undefined instruction operation code. It may also occur as the result of a memory parity error or memory protect/privileged instruction feature (MP/PIF) violation. In any case, when an internal interrupt is honored, program execution begins at location 0002_{16}. This interrupt condition is of higher priority than any other interrupt. Although the memory parity and MP/PIF interrupts can be masked, there is no bit in the status register to inhibit internal interrupts caused by illegal operation codes or power failure; therefore, there is no point in program execution at which an internal interrupt cannot occur.

It is common programming practice to put a Store Status Block (SSB) instruction in location 0002_{16}. When an internal interrupt occurs, this instruction links to a program sequence which will:

- Interrogate bit 15 of the saved contents of the status register. If it is a one, power failure is imminent and register contents must be preserved in memory (1 ms (980A), or 20 ms (980B), is available to perform this before execution is terminated). The status register bits are defined in table 3-1.
- Interrogate bit 14 of the saved contents of the status register. If it is a logic one, a memory parity error was detected during execution of the instruction preceding the one pointed to by the saved program counter.
- Interrogate bit 5 of the saved contents of the status register. If it is a logic one, an attempt was made to store data into, or branch to, a location outside the memory protect limits established.
- Interrogate bit 6 of the saved contents of the status register. If it is a logic one, an attempt was made to execute one of the instructions privileged by PIF.
- If none of these conditions is detected, an attempt has been made to execute an illegal instruction. The codes for illegal instructions are listed in table 3-2.
- After all processing is complete, a Load Status Block (LSB) instruction may be executed which will return the machine to the interrupted program sequence.

Table 3-1. Status Register Bit Functions

Status Register Bits	Function
0-1	Compare Indicators - Indicate the result of the last compare operation. 00 - less than 01 - equal to 10 - greater than 11 - not allowed
2	Overflow Indicator - Turned on or off by those instructions which can cause overflow.
3	Carry Indicator - Turned on or off by any add or subtract instruction which can result in a carry into the sign bit of a register.
4	Privileged Instruction and Memory Protect. 0 - Disabled 1 - Enabled
5	Memory Protect Address Violation - May not be set under program control. 0 - No Violation 1 - Violation
6	PIF Instruction Violation - May not be set under program control. 0 - No Violation 1 - Violation
7	Data Bus Interrupt Control 0 - Disabled 1 - Enabled
8	Vectored Interrupt Feature 0 - Disabled 1 - Enabled
9	PIF Lower Limit Address Bias 0 - Disabled 1 - Enabled
10	Index Control 0 - Post Indexing 1 - Pre-indexing

Table 3-1. Status Register Bit Functions (Continued)

Status Register Bits	Function
11	Memory Parity Error Interrupt Control 0 - Disabled 1 - Enabled
12	DMAC Interrupt Control 0 - Disabled 1 - Enabled
13	Not Used
14	Memory Parity Error Indicator - May not be set under program control. 0 - No Error 1 - Error
15	Power Fail Indicator - 1 ms (980A) or 20 ms (980B) warning that power failure is imminent. May not be set under program control. 0 - Power Up 1 - Power Failure Imminent

If an instruction other than an SSB or an instruction affecting the PC register (such as a branch or skip) is placed in location 0002_{16}, then only the single instruction will be executed. Since the PC is held fixed for execution of the one instruction only, program control is immediately returned to the interrupted program.

CAUTION

If it is necessary to use an instruction that does not affect the PC at location 0002_{16}, do not use an instruction that does not affect the MA register. The MA is used to retrieve the trap instruction, so if the trap instruction does not modify the MA, the MA will not be updated to the current PC value at instruction termination.

Further interrupts are inhibited until after execution of one instruction subsequent to that instruction in location 0002_{16}. In other words, the SSB instruction and the first instruction it branches to will both be performed before another interrupt of any kind can be honored. When an internal interrupt is recognized, bits 4, 5, 6, 7, 8, 9, 12, 14 and 15 of the status register are

Table 3-2. Illegal Instruction Codes

Instruction Bits															
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	1	0	0	0	X	X	X	X	X	X	X	1	X	X	X
1	1	0	0	0	0	X	1	1	X	X	X	X	X	X	X
1	1	0	0	0	1	0	1	1	X	X	X	X	X	X	X
1	1	0	0	1	0	0	1	X	X	1	X	X	X	X	X
1	1	0	0	1	0	1	0	1	0	1	X	X	X	X	X
1	1	0	0	1	0	1	0	1	1	X	X	X	X	X	X
1	1	0	0	1	0	1	1	0	0	0	X	X	X	X	X
1	1	0	0	1	0	1	1	1	X	1	X	X	X	X	X
1	1	0	0	1	1	1	0	0	0	0	1	X	X	X	X
1	1	0	0	1	1	1	0	0	0	1	X	X	X	X	X
1	1	0	0	1	1	1	0	0	1	X	X	X	X	X	X
1	1	0	0	1	1	1	0	1	X	X	X	X	X	X	X
1	1	0	0	1	1	1	1	X	0	X	X	X	X	X	X
1	1	0	0	1	1	1	1	0	1	0	X	X	X	X	X
1	1	0	1	0	X	X	X	X	X	X	X	X	X	X	X
1	1	0	1	1	0	1	X	1	X	X	X	X	X	X	X
1	1	0	1	1	1	X	0	X	X	X	X	X	X	X	X
1	1	1	X	X	X	X	X	X	X	X	X	X	X	X	X

X = DON'T CARE (0 or 1)

cleared to zero after execution of the instruction in the trap location. If the machine traps because of an illegal operation, and an illegal instruction is in location 0002_{16}, the machine is halted.

3.2 POWER FAILURE PROTECTION

The 980 is protected against loss of data due to power failure. An input power detector allows 1 millisecond (980A), or 20 milliseconds (980B), of operation when power failure is imminent and then disables the computer and the memory initiate circuit. The operating program is warned of the power failure condition via an internal interrupt. Subsequently, when power is restored, the operating program is automatically started at memory location zero. When power is initially applied to the computer, the situation is internally analogous to a power-restored condition, so program execution automatically starts at memory location zero.

It is a common programming technique to put an LSB instruction in location 0000_{16}. The LSB branches to a sequence of instructions which restores the machine environment that existed prior to power loss. Since a power loss clears all DMAC controllers and operations in progress are not completed, care must be taken in restarting DMAC devices. The Basic System, if used, includes routines to handle the power-restore/start-up condition.

3.3 MEMORY PARITY ERROR DETECTION

The 980 performs a check of parity on all data transferred from memory. In the event a parity error is detected, one of two possible options occurs:

- If bit 11 of the status register (parity error mask) is a zero, the parity error is ignored.
- If bit 11 of the status register is a one, bit 14 of the status register is set to a one, and an internal interrupt is generated.

The first option is usually not an acceptable mode of operation, because it does not even allow a count of parity errors to be kept. The second option allows the user to attempt a recovery, to continue in a degraded mode of operation, or to terminate his program.

3.4 MEMORY PROTECT/PRIVILEGED INSTRUCTION FEATURE (MP/PIF)

When program execution is under direction of a monitor program, it is desirable to restrict a user program in several ways in order to protect the monitor as well as the balance of the operating system. These restrictions are implemented on the 980 through use of the MP/PIF which prevents a user program from:

- Reading, writing, or branching into protected memory.
- Changing the status register.
- Performing I/O.
- Bringing the computer to an idle.

To enable the MP/PIF, it is first necessary to load the MP/PIF lower limit and upper limit registers that define the limits within which execution will be constrained. Both registers are loaded by use of the WDS instruction just as if the MP/PIF registers were external to the computer. Register address zero defines the lower limit and register address one defines the upper limit. These boundary locations and all memory outside of the boundaries are protected by the MP/PIF feature. The monitor then enables MP/PIF through use of an LSB instruction that sets bit 4 of the status register. Control can now be transferred to the user program.

3.5 PROGRAM RELOCATION FEATURE

The lower limit register used by the MP/PIF is also used by the program relocation feature. If the monitor sets bit 9 of the status register, at the time control is transferred to the user program, the contents of the lower limit register plus one is added into the address calculations for each memory access. The effect of this is that a program may be loaded anywhere in memory, but the program operates as though it were loaded starting at location 0000_{16}.

For example, suppose a program is assembled as an absolute program with origin at location 0000_{16}. Also, suppose that the entry point to the program is location 0020_{16}, and that it is convenient for the monitor to load the program at location 1000_{16}. The monitor loads the program starting at 1000_{16}, places $0FFF_{16}$ in the lower limit register, and performs an LSB instruction to transfer to the program. The LSB must set bit 9 of the status register and load the program counter with 0020_{16}. Note that although the instruction executed is at 1020_{16}, the program counter contains 0020_{16}. If, for instance, a trap were to occur, the SSB at the trap location would save the value 0020_{16} for the program counter, not 1020_{16}.

SECTION IV

MODEL 733 ASR/KSR PROGRAMMING CONSIDERATIONS

4.1 GENERAL INFORMATION

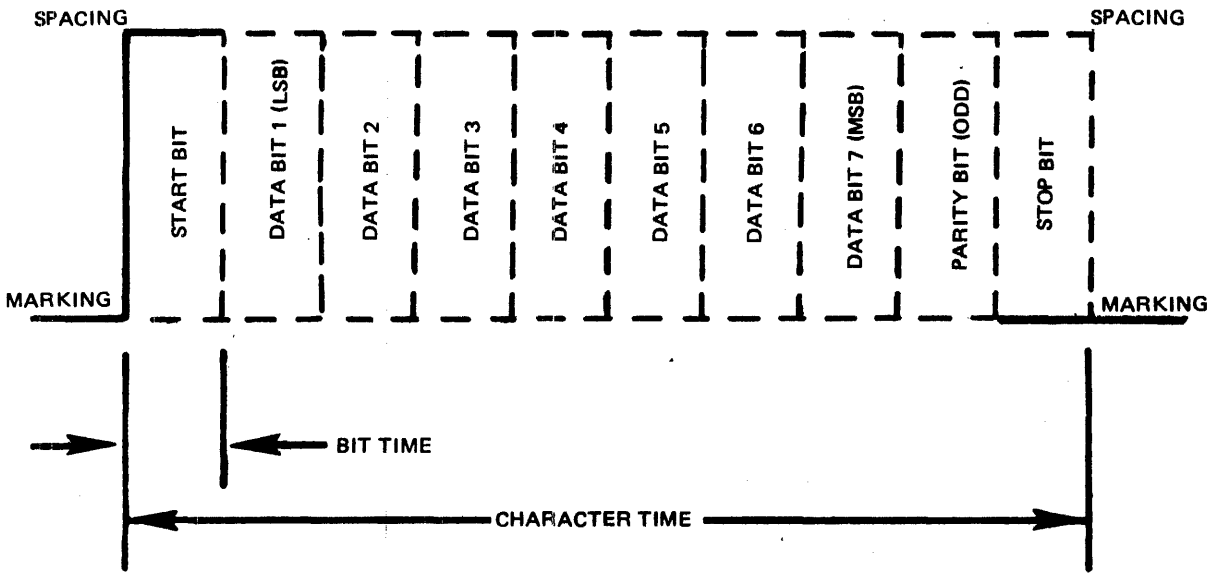
This section contains the information necessary to program all I/O operations involving the Texas Instruments 733 ASR/KSR data terminals. Included in this section is a discussion of character format and transmission rates, a detailed breakdown of control, status, and data interface lines, and programming examples involving the teleprinter, keyboard, and cassette portions of the data terminal.

The 733 ASR/KSR data terminals communicate with the Model 980 Computer on a character-by-character basis via the Full Duplex EIA Communications Module. The communications module plugs into an I/O data bus port of the computer and connects to the 733 ASR/KSR data terminal with a standard 30 foot cable. In this configuration, the communications module performs both the data synchronization and formatting necessary for the computer to interface with the 733 ASR/KSR data terminal. The logic circuits on the communications module perform all the functions required to serially transmit and receive data while supplying a variety of status indicators and control signals to the computer. The transmit and receive circuits on the communications module operate independently of one another (full-duplex operation), and each is able to convert between the computer TTL logic levels and the 733 ASR/KSR EIA levels.

4.1.1 TRANSMISSION RATE AND CHARACTER FORMAT

The 733 ASR (printer with dual cassettes) allows data transfer to and from the cassettes at a rate of 1200 baud. The 733 ASR in printer mode operates at an effective rate of only 300 baud (1200 baud with a programmable delay). All 733 KSR operations are at 300 baud. Synchronous operation at 300 baud is maintained by creating an effective 300-baud transmission rate comprised of an 8.33 millisecond character time followed by a 25.0 millisecond minimum delay time. On the communications module, this is accomplished by setting the Write Request Delay (WRD) bit (WDS control bit 9) to a logic one prior to initiating transmission of an USASCII character data string. It is necessary that this bit be a logic zero for direct cassette communication at 1200 baud.

Regardless of the baud rate, an USASCII character will be represented at the communications module/733 ASR/KSR interface as shown in figure 4-1. The serial bit pattern consists of a start bit, seven data bits with the least significant bit first, a parity bit, and a stop bit. The seven bits of data



	300-BAUD	1200-BAUD
BIT TIME	3.33ms	.833ms
CHARACTER TIME	33.3ms	8.33ms

(A)129611

Figure 4-1. 733 ASR Character Format and Timing

comprise the limited USASCII character code (refer to Appendix A) used in the computer. The start and stop bits surrounding the USASCII character are used for timing purposes at the communications module/733 ASR/KSR interface.

4.1.2 COMMUNICATIONS MODULE OPTIONS

The communications module has a group of Single Pole Single Throw (SPST) switches that affect the communications module interface with both the computer and the 733 ASR/KSR terminal. The switches are located on the component side of the module board in two Dual-Inline-Packages, as shown in figure 4-2. The switch-selectable options include the I/O data bus address

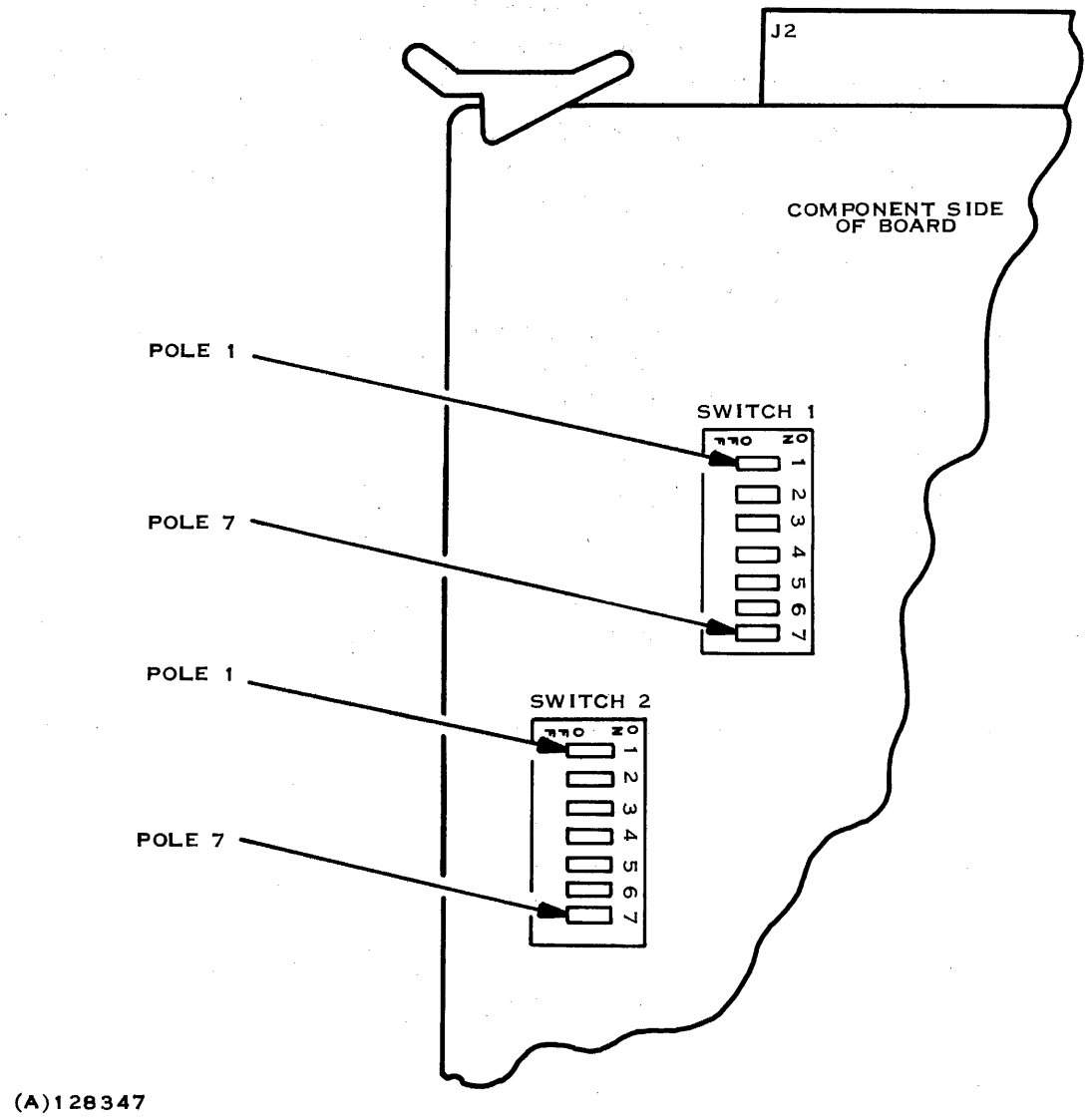


Figure 4-2. Communications Module Switch Locations

assigned to the communications module, the transmit and receive baud rate, word length, and stop bit configuration associated with the interfacing data terminal, and parity bit control. Each of these options is explained in more detail in the following list:

- Module Address Selection - The communications module must be assigned to one of the 64 available I/O data bus addresses. The selected address should not conflict with the address of any other peripheral device on the I/O data bus, and software must be aware of the selected address. The available addresses range from 00_{16} to $1F_{16}$ and 40_{16} to $5F_{16}$, with 05_{16} used as the standard address by Texas Instruments furnished software. Table 4-1 lists the switch poles used in the address selection and the logic value of the two pole positions.

- Baud Rate Selection - The transmit and receive baud rate is determined by switch S2, poles 1 to 3, as listed in table 4-2. The standard rate is 1200 baud for the 733 ASR and 300 baud for the 733 KSR.
- Parity Selection - Poles 4 and 5 of switch S2 are used to select even or odd parity generation, or no parity, as listed in table 4-3. Odd parity is standard with the 733 ASR/KSR data terminals.
- Word Length Selection - The word length used in communications between the computer and data terminal may be set to 5, 6, 7, or 8 bits (in addition to parity). The number of bits is determined by poles 6 and 7 of switch S2 as listed in table 4-4. The 7-bit word is standard with the 733 ASR/KSR data terminals.
- Stop Bit Selection - The number of stop bits is switch selectable. Pole 1 of switch S1 determines the number of stop bits as listed in table 4-5.

Table 4-1. Communications Module Address Selection

Hex Digit of Address	WDS/RDS Word 1 Bits*	Switch S1 Pole	Logic 1 Position	Logic 0 Position
MSD	9	2 (A09)	"OFF"	"ON"
	11	3 (A11)		
LSD	12	4 (A12)		
	13	5 (A13)		
	14	6 (A14)		
	15	7 (A15)		

*Bit 10 of word 1 is used to specify WDS or RDS, and is assumed to be zero in developing the MSD of the module address.

4.2 PROGRAMMING INFORMATION

The 733 ASR/KSR data terminal interfaces with the computer via one I/O data bus external register. The single register, normally addressed at 05_{16} by Texas Instruments furnished software, handles the transmission of data, command, and status information between the computer and 733 ASR/KSR data terminal. The following paragraphs describe the data word formats associated with the 733 ASR/KSR WDS and RDS instructions (including subcommands), and provide programming examples illustrating their use.

Table 4-2. Baud Rate Selection

Baud Rate	Switch S2		
	Pole 1 (R/A)	Pole 2 (R/B)	Pole 3 (R/C)
75 110	"ON" "OFF"	"ON"	"ON"
150 300	"ON" "OFF"	"OFF"	
1200 2400	"ON" "OFF"	"ON"	"OFF"
9600 4800*	"ON" "OFF"	"OFF"	

*4800 Baud for E15-E16 (Standard)
600 Baud for E16-E17

Table 4-3. Parity Selection

Parity Selection	Switch S2 Pole	
	Pole 4 (PI)	Pole 5 (PS)
Parity Inhibited	"OFF"	-
Odd Parity	"ON"	"ON"
Even Parity		"OFF"

4.2.1 733 ASR/KSR WDS DATA WORD DESCRIPTION

The 16-bit WDS data word is used by the programmer to issue commands, subcommands, and USASCII characters to the data terminal. The format of the WDS data word is shown in figure 4-3. Note that when bit 7 of the data word is set, bits 8 through 15 of the data word contain additional control information; when bit 1 of the data word is set, bits 8 through 15 of the

Table 4-4. Word Length Selection

Number of Data Bits	Switch S2 Pole	
	Pole 6 (WL2)	Pole 7 (WL1)
5	"ON"	"ON"
6		"OFF"
7	"OFF"	"ON"
8		"OFF"

Table 4-5. Stop Bit Selection

Stop Bit Selection	Switch S1 Pole 1 (STB)
1 Stop Bit	"ON"
2 Stop Bits	"OFF"

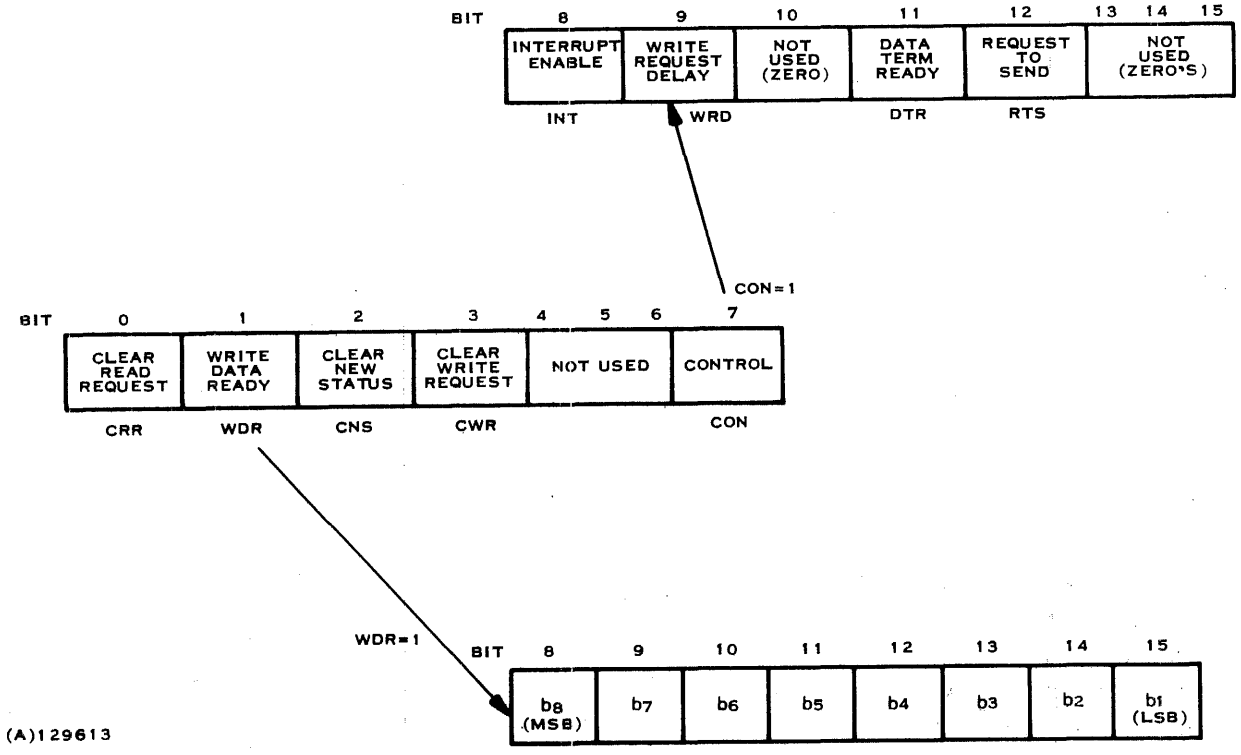


Figure 4-3. 733 ASR/KSR WDS Data Word Format

data word contain a USASCII character. The character may represent a subcommand or data to be written to the data terminal. The two types of subcommands are described in the next paragraph. The individual bits of the WDS data word are described in the following list:

- Clear Read Request (CRR) - A logic one in bit 0 clears the Read Request (RREQ) flag and the associated interrupt (if enabled) on the communications module. The RREQ flag is described in bit 0 of the RDS data word format. A logic zero in bit 0 causes no action.
- Write Data Ready (WDR) - A logic one in bit 1 loads the USASCII character in bits 8 through 15 in the transmit buffer register on the communications module. This action initiates the transmission sequence to the data terminal. A logic zero in bit 1 causes no action.
- Clear New Status (CNS) - A logic one in bit 2 clears the New Data Set Status (DSS) flag and the associated interrupt (if enabled) on the communications module. The DSS flag is described in bit 2 of the RDS data word format. A logic zero in bit 2 causes no action.
- Clear Write Request (CWR) - A logic one in bit 3 clears the Write Request (WREQ) flag and the associated interrupt (if enabled) on the communications module. The WREQ flag is described in bit 1 of the RDS data word format. A logic zero in bit 3 causes no action.
- Not Used - Bits 4, 5, and 6 of the WDS data word are not used by the 733 ASR/KSR data terminal.
- Control (CON) - A logic one in bit 7 indicates to the communications module that bits 8, 9, 11, and 12 contain control information. A logic zero in bit 7 causes no action.

If bit 1 of the WDS format is set, bits 8 through 15 contain a USASCII character with bit 8 the MSB and bit 15 the LSB. If bit 7 of the WDS format is set, the following controls are output to the communications module:

- Interrupt Enable (INT) - A logic one in bit 8 enables the communications module to issue an I/O data bus interrupt when a read request, write request, or data set status change occurs. The interrupt is cleared by clearing the condition that caused the interrupt. A logic zero in bit 8 inhibits the communications module interrupt.
- Write Request Delay (WRD) - A logic one in bit 9 delays the communications module from issuing a write request (bit 1 of the RDS data word format) until 33 milliseconds after loading the transmit buffer register. Normally, a write request is issued when the transmit buffer register becomes empty. This delay option is used to create an effective 300-baud serial transmission rate when addressing the teleprinter portion of the 733 ASR/KSR data terminal through the 1200-baud interface. The delay results in a one-character print period of 33.33 milliseconds, equaling the 30

character per second print rate of the 733 teleprinter. Figure 4-4 shows the timing relationships between the 300-baud and 1200-baud transmission rates. A logic zero in bit 9 inhibits the delay action.

- Data Terminal Ready (DTR) - A logic one in bit 11 maintains the DTR circuit on the communications module in the "ON" condition. The DTR circuit drives the Data Set Ready (DSR) input line of the 733 ASR/KSR data terminal to enable the transmit and receive capabilities of the data terminal. A logic zero in bit 11 maintains the DTR circuit in the "OFF" condition.
- Request to Send (RTS) - A logic one in bit 12 maintains the RTS circuit on the communications module in the "ON" condition. The RTS circuit drives the Data Carrier Detect (DCD) input line of the 733 ASR/KSR data terminal to enable the data terminal to accept USASCII characters from the communications module. A logic zero in bit 12 maintains the RTS circuit in the "OFF" condition.
- Not Used - Bits 10, 13, 14, and 15 of the WDS control data word are not used by the 733 ASR/KSR data terminal.

4.2.2 733 ASR SUBCOMMANDS

The 733 ASR subcommands, or Remote Device Control (RDC) functions, each consist of a USASCII character placed in the least significant half of the data word associated with a WDS instruction. These control instructions are used by the programmer to perform such mechanical tasks as rewind and load cassettes.

The RDC functions are basically divided into two categories: (1) those activated by receipt of a single USASCII control character and (2) those activated by receipt of the non-printable "DLE" character followed by a second predetermined USASCII character code. The first category is known as the Automatic Device Control (ADC), or single-character subset, while the second category is known as the DLE, or two-character subset. The two categories

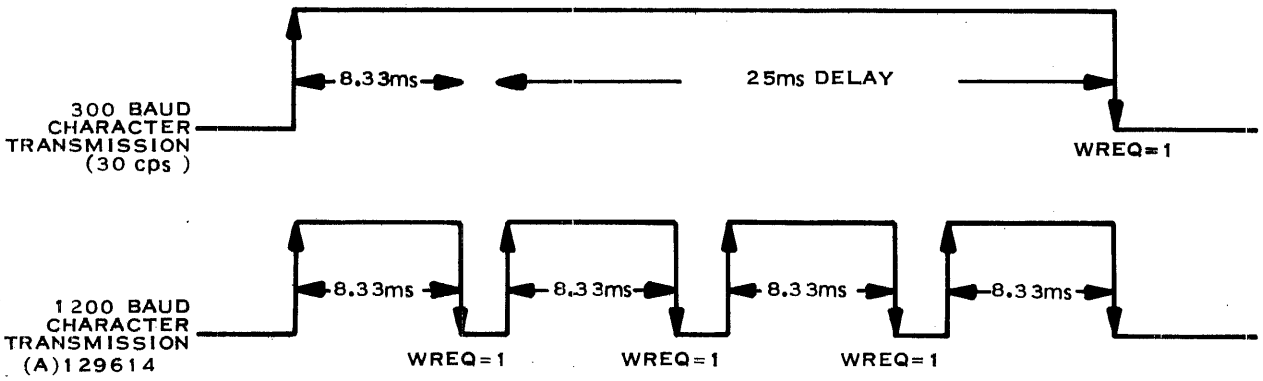


Figure 4-4. 300-Baud/1200-Baud Timing Relationships

Table 4-6. Remote Device Control Functions

Single-Character Functions (ADC)		
Function	Hex Code	USASCII Character
Playback On/Keyboard Off (No busy bit necessary)	>11	DC1
Record On/Printer Off	>12	DC2
Playback Off/Keyboard On	>13	DC3
Record Off/Printer On	>14	DC4
Two-Character Functions ("DLE")		
Function	Hex Code	USASCII Character
Rewind Cassette 1	>31	1
Rewind Cassette 2	>32	2
Load Cassette 1	>33	3
Load Cassette 2	>34	4
Cassette 1 In Record, 2 In Playback	>35	5
Cassette 2 In Record, 1 In Playback	>36	6
Block Forward	>37	7
Block Reverse	>38	8
Printer On (Non-Operable)	>39	9
Printer Off (Non-Operable)	>30	0
Automatic Device Control (ADC Above) On	>3A	:
Automatic Device Control (ADC Above) Off	>3B	;
Request Status Information	>3C	<

of RDC functions and their activation codes are listed in table 4-6. It should be noted that the RDC logic circuitry automatically disables the teleprinter portion of the 733 ASR from printing the first character received after the "DLE" character code. This ensures that all of the two-character functions are, in effect, treated in the same manner as any non-printable USASCII control character, such as those in the ADC (single-character) category.

The Command Status Character, shown in figure 4-5, is transmitted by the 733 ASR data terminal to the computer in response to the RDC request status

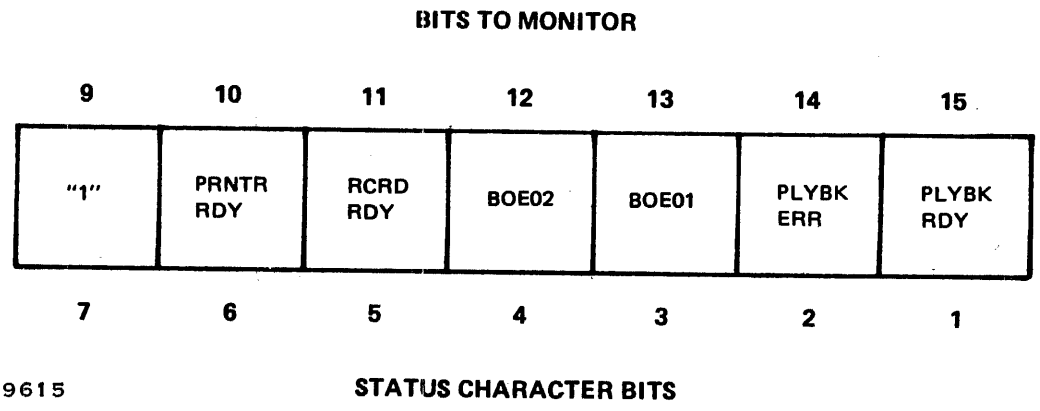


Figure 4-5. Command Status Character Bits

command (command ">3C" of table 4-6). The status character bits, numbered from 7 (MSB) to 1 (LSB), correspond to bits 9 through 15 at the computer interface. Each of the bits is described in the following list:

- Playback Ready (Bit 1) - A logic one level in this bit position indicates the playback cassette is ready to be enabled with the ADC playback on ("DC1") or the RDC block forward ("DLE", "7") function. In the event that a logic zero level is found in this bit position, one or more of the following conditions must exist:
  1. Cassette door open or cassette not in place.
  2. Cassette tape on clear leader.
  3. Playback not in line mode.
  4. Rewind or load operation still in execution.
- Playback Error (Bit 2) - A logic one in this bit position indicates that a playback error has been discovered while an 86 character block of data was being read from the cassette tape. In this event, the transmission of any characters from the block of data is inhibited until the error status is reset to a logic zero level. The RDC block reverse command is normally utilized in the recovery of a playback error. Only block reverse is capable of clearing the error status.
- BOE01 (Bit 3) - A logic one level in this bit position indicates that cassette 1 is positioned on clear leader at either the beginning or the end of the tape. Note that either condition automatically inhibits the RDC rewind command function.
- BOE02 (Bit 4) - The description of this bit position is identical to that for BOE01, except that the information applies to the position of cassette 2.

- Record Ready (Bit 5) - A logic one level in this bit position indicates that the record cassette is ready to be enabled with the ADC record on ("DC2") function. If a logic zero level is found in this bit position, one or more of the following conditions must exist:

    1. Cassette door open or cassette not in place.
    2. Cassette tape on clear leader.
    3. Record not in line mode.
    4. Rewind on load operation still in execution.
    5. Record enable tab removed.

- Printer Ready (Bit 6) - A logic one level in this bit position indicates that the teleprinter is enabled and ready for normal operation. If a logic zero level is found in this bit position, one or both of the following conditions must exist:

    1. Printer not in line mode.
    2. Printer is disabled as a result of the ADC record-on function ("DC2").

- Bias Bit (Bit 7) - This bit position will always be held to a logic one level to ensure that the status character is not an element of the USASCII control character subset.

The following list describes both the single-character and double-character RDC commands:

- Playback On/Keyboard Off (DC1) - The "playback on" function initiates the reading and transmitting of characters from the cassette tape, provided the playback cassette is loaded and ready as indicated by bit 1 of the status character. This function is performed only on data received over the communication lines or generated in the terminal's local loop.

- Record On/Printer Off (DC2) - The "record on" function initiates the receipt and recording of characters onto the cassette tape, provided the record cassette is loaded and ready as indicated by bit 5 of the status character. Upon receipt of "DC2", the record function is turned on and the printer enters a print cycle regardless of the fact that "DC2" is a non-printable character. The "DC2" character is neither entered into the hardware buffer nor recorded on the tape. This function is performed only on data received over the communication lines or generated in the local loop.

- Playback Off/Keyboard On (DC3) - The "playback off" function terminates the reading and transmitting of characters from the cassette tape under the condition that if the "DC3" (X-OFF) code is on the

playback tape then one more character after the "DC3" will be played back before the cassette is acutally turned off. This function is performed on both transmitted and received data as well as data generated in the terminal's local loop. "DC3" and "RUB OUT" (delete) are the last two codes, respectively, entered by the terminal operator to conclude each source line.

- Record Off/Printer On (DC4) - The "record off" function terminates the receipt and recording of characters onto the cassette tape. Upon receipt of the "DC4" character, the record function is turned off and the printer is turned on, however, a print cycle is not entered as is the case with the "DC2" command. The "DC4" character is then entered into the hardware record buffer and all succeeding characters sent to the printer (including the "DC2" character) are also "overpunched" on the "DC4" character in the buffer. The resulting character (usually a $7F_{16}$ delete) is recorded on the tape along with the data in the normal manner (i.e., when a carriage-return is received in the Line format mode of operation or when 86 characters have been entered into the buffer in either the Line or Continuous format modes of operation). Note that a string of data is not automatically recorded onto the cassette tape simply by issuance of the "record off" command. This function is performed only on data received over the communication lines or generated in the local loop.

- Rewind Cassette 1 (DLE, 1) - Upon receipt of the rewind command code (under the condition that the cassette tape is not already on clear leader at either end of the tape), cassette 1 will slew in the reverse direction until clear leader is sensed at the beginning of the tape. From the time that the cassette begins performing the rewind operation, either bit 1 or bit 5 of the status character (depending on whether cassette 1 is in the playback or record mode respectively) will be held to a logic zero, indicating the cassette is not ready. When the rewind operation has been completed, bit 3 of the status character will become a logic one indicating the presence of the clear leader. Only the issuance of a load cassette 1 command will reset the appropriate bits of the status character and ready the cassette. Note that the cassette cannot be remotely rewound once it has entered the clear leader at the far end of the tape.

- Rewind Cassette 2 (DLE, 2) - The description of this command code is identical to that for rewind cassette 1, except that bit 4 of the status character indicates the presence of clear leader on cassette 2.

- Load Cassette 1 (DLE, 3) - Upon receipt of the load command code (under the condition that the clear leader is sensed at the beginning of the tape), cassette 1 will advance the tape until the beginning of tape marker is sensed. During the time that the load operation is being performed, either bit 1 or bit 5 of the status character (depending on whether cassette 1 is in the playback or record mode, respectively) will be held to a logic zero, indicating that the cassette is not ready. Upon completion of the load operation, the appropriate bit of the status character will become a logic one, indicating the cassette is ready to be enabled with either the "record on" or the "playback on" command code.

- Load Cassette 2 (DLE, 4) - The description of this command code is identical to that for load cassette 1, except that bit 4 of the status character is the bit which indicates the presence of clear leader on cassette 2.

- Cassette 1 In Record, 2 In Playback (DLE, 5) - Upon receipt of the appropriate two-character command code, cassette 1 will be placed in the record mode of operation and cassette 2 will be placed in the playback mode. This operation will be performed regardless of the state of any bits in the status character, although its execution may change the state of several bits in the status character.

- Cassette 2 In Record, 1 In Playback (DLE, 6) - The description of this command code is similar to that for cassette 1 in record, 2 in playback, except that the result of this command code is the complement of that description.

- Block Forward (DLE, 7) - Upon receipt of the block-forward command code (under the condition that the playback ready bit of the status character is true), the cassette controller will read an 86-character block of data from the playback cassette and will initiate transmission of the resulting string of USASCII characters. Transmission of the string will continue until all 86 characters have been exhausted or until a playback-off ("DC3") character code is encountered in the transmission. In the event that transmission is terminated by occurrence of the "DC3" character on the tape, the block forward command code may be reissued to initiate transmission of those characters remaining in the 86-character block of data originally read from the tape. The block forward command code may have to be reissued several times, depending on the number of "DC3" characters in the original 86-character block of data.

  In the event that transmission was never initiated because a playback error was discovered while reading the 86-character block from the cassette tape, this command code will cause the following: the playback error bit of the status character (bit 2) will be reset; the block containing the error will be bypassed; a new 86 character

block will be read from the cassette tape; transmission of the new block will be initiated, provided that the new block did not also contain a playback error.

- Block Reverse (DLE, 8) - Upon receipt of the block-reverse command code (under the condition that the playback ready bit of the status character is true), the cassette controller will rewind the tape one full block of data (86 characters) and clear the contents of the hardware read buffer. In the event that this command code is received while the playback error bit of the status character (bit 2) is true, this command will also reset the error flag, thereby preparing the block to be reread.
- Printer-On (DLE, 9) - This RDC command code is non-operable since this function has already been automatically included as a part of the ADC record-off command code ("DC4").
- Printer-Off (DLE, 0) - This RDC command code is non-operable since this function has already been automatically included as a part of the ADC record-on command code ("DC2").
- Automatic Device Control On (DLE, :) - This command enables the playback on/off ("DC1"/"DC3") and record on/off ("DC2"/"DC4") ADC commands to become effective after having been disabled with the "ADC OFF" command.
- Automatic Device Control Off (DLE, ;) - This command disables the playback on/off ("DC1/"DC3") and record on/off ("DC2"/"DC4") ADC commands. It is particularly useful for recording the "DC1" through "DC4" characters onto a cassette tape without affecting the operation of the 733 ASR data terminal.
- Request Status (DLE, <) - Upon receipt of the request status command code, the 733 ASR data terminal will transmit the specially configured USASCII Command Status Character to the computer interface. This character is particularly useful in determining when such mechanical functions such as rewind and load operations are completed.

4.2.3 733 ASR/KSR RDS DATA WORD DESCRIPTION

The 16-bit RDS data word is used by the programmer to read status and USASCII characters from the data terminal. The format of the RDS data word is shown in figure 4-6. Note that when bit 0 of the data word is a logic one, bits 8 through 15 of the data word contain an USASCII character; when bit 0 of the data word is a logic zero, bits 8 through 15 of the data word contain additional status information. The only exception to this generality on bit 0 of the data word occurs following a request status subcommand to the 733 ASR. In this case, the Read Request flag (bit 0 of the RDS data word)

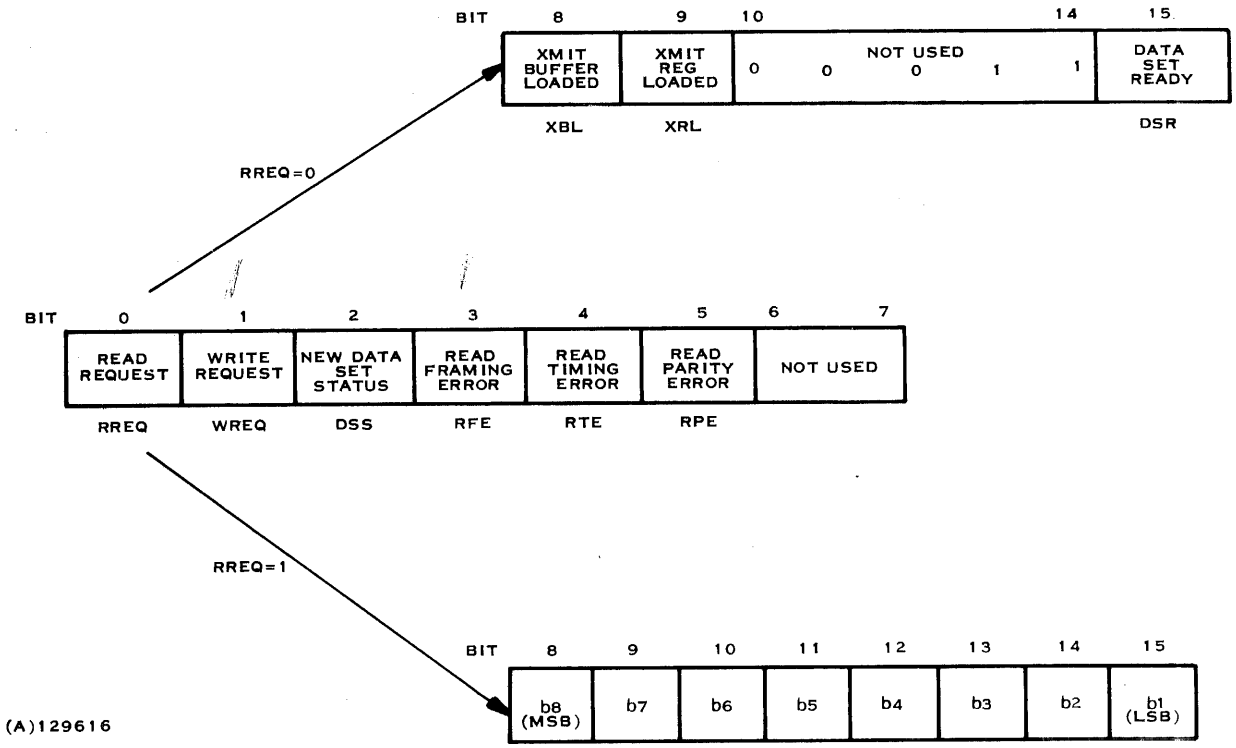


Figure 4-6. 733 ASR/KSR RDS Data Word Format

is set to indicate that the cassette Command Status Character has been loaded in bits 9 through 15 of the data word. The individual bits of the RDS data word are described in the following list:

- Read Request (RREQ) - A logic one in bit 0 indicates that a character has been transferred from the data terminal to the communications module, and is available to the computer in bits 8 through 15 of the data word. The set RREQ flag will cause an I/O data bus interrupt if the communications module interrupt is enabled (via bit 8 of the WDS data word). A logic zero in bit 0 indicates that no character has been read from the data terminal and bits 8, 9, and 15 of the data word contain status information.
- Write Request (WREQ) - A logic one in bit 1 indicates that the transmit buffer register on the communications module is empty, and is ready to accept a new character from the computer for transmission to the data terminal. The set WREQ flag will cause an I/O data bus interrupt if the communications module interrupt is enabled (via bit 8 of the WDS data word). A logic zero in bit 1 indicates that the communications module is not requesting a new character. This may be due to a full transmit buffer register or the clearing of the WREQ flag following the transmission of the last USASCII character in a data string.

- New Data Set Status (DSS) - A logic one in bit 2 indicates that a change of state has occurred in the Data Set Ready (DSR) signal from the 733 ASR/KSR. This usually means the keyboard ON-LINE/LOCAL switch on the 733 ASR/KSR has been repositioned. The set DSS flag will cause an I/O data bus interrupt if the communications module interrupt is enabled (via bit 8 of the WDS data word). A logic zero in bit 2 indicates that no change has occurred in the DSR signal.

- Read Framing Error (RFE) - The RFE signal is valid only as long as the Read Request flag of the interface is set. A logic one in bit 3 indicates that improper character framing was detected by the communications module receive circuitry. This condition may be caused by an improperly selected baud rate or an erroneous number of stop bits. The RFE flag is cleared when another character is properly received, provided that a Clear Read Request has been previously issued (bit 0 of WDS data word). A logic zero indicates that the character was properly received with respect to both timing and stop bit considerations.

- Read Timing Error (RTE) - The RTE signal is valid only as long as the Read Request flag of the interface is true set. A logic one in bit 4 indicates that the communications module received another USASCII character from the 733 ASR/KSR before the computer completed loading the previous character. In this event, the communications module replaces the previous character with the one most recently received and, as such, this flag indicates that one or more characters have been lost in transmission. The RTE flag is cleared when another character is properly received, provided that a Clear Read Request has been previously issued (bit 0 of WDS data word). A logic zero indicates that no characters have been lost in transmission.

- Read Parity Error (RPE) - The RPE signal is valid only as long as the Read Request flag of the interface is set. A logic one in bit 5 indicates that improper parity generation was detected by the communications module receive circuitry. Both the communications module and the 733 ASR/KSR data terminals expect an odd parity bit generation and, as such, this flag indicates that the received character contained an even parity bit. The RPE flag is cleared when another character is properly received, provided that a Clear Read Request has been previously issued (bit 0 of WDS data word). A logic zero indicates that the character was properly received with an odd parity bit.

- Not Used - Bits 6 and 7 of the RDS data word are not used by the 733 ASR/KSR data terminal.

If bit 0 of the RDS data word (RREQ) is a logic one, bits 8 through 15 contain an USASCII character with bit 8 the MSB and bit 15 the LSB. If bit 0 of the RDS data word is a logic zero, the following status data is returned to the computer:

- Transmit Buffer Register Loaded (XBL) - A logic one in bit 8 indicates that the communications module transmit buffer register is presently loaded with a USASCII character and unable to accept another. A logic zero indicates that the USASCII charcter has been transferred to the transmit shift register and that the transmit buffer register is now ready to accept another USASCII character.
- Transmit Shift Register Load (XRL) - A logic one in bit 9 indicates that the communications module character is in process of being serially transmitted to the 733 ASR/KSR data terminal. A logic zero indicates that the module transmit shift register is empty and that no serial transmission is presently in progress.
- Not Used - Bits 10 through 14 of the RDS status data word are not used by the 733 ASR/KSR data terminal. Bits 10, 11, and 12 are wired to logic zero's and bits 13 and 14 are wired to logic one's.
- Data Set Read (DSR) - A logic one in bit 15 indicates that the 733 ASR/KSR data terminal is in the on-line mode of operation. This circuit is driven by the Data Terminal Ready output line of the 733 ASR/KSR data terminal and is maintained in the "ON" condition as long as the 733 ASR/KSR keyboard ON-LINE switch is engaged and the ASR-to-computer connection is maintained.

4.2.4 PROGRAMMING EXAMPLES

The programming examples in the following paragraphs include writing to both the teleprinter and cassette portions of the 733 ASR, reading from both the keyboard and cassettes, and issuing subcommands. All of the examples address I/O data bus register 05_{16} in conjunction with the 733 ASR/KSR data terminal.

The busy bit test, associated with both WDS and RDS instructions, should not be used when writing a data word consisting entirely of control data or reading a data word consisting entirely of status data. In addition, the busy bit test should not be used with the first WDS instruction in a program or when writing the DC1 subcommand (playback on) to the terminal. These restrictions make it advantageous to make the first WDS instruction one that writes all control data or issues the DC1 subcommand. If this is not possible, the busy bit test can be replaced with a check on the Write Request (WREQ)

flag to determine when the WDS data word character bits have been accepted by the data terminal. The following example illustrates this alternate method:

```
                LDA     => 12       SEND DC2 COMMAND (RECORD-ON CHARACTER)
                @IOR    => 5000     LOAD DATA, CLR WRITE REQ
                WDS     5           WRITE TO ASR FROM REG A
                DATA    0           NO BUSY BIT (1ST WDS)
Replaces        RDS     5           READ STATUS (BIT 0 = 0)
Busy Bit        DATA    0           REGISTER A
                TABO    1           TEST BIT 1 (WREQ=1?)
                BRU     $-3         LOOP IF WREQ = 0
                .
                .
                .
```

In general, the guidelines listed below should be followed in programming the 733 ASR/KSR data terminal:

- If the ASR portion of the data terminal is the subject of the program, issue the appropriate subcommand(s) (refer to table 4-6) to ready the desired cassette for the I/O data transfer.
- If the busy bit test is required on the first WDS instruction of the program, use the alternate method shown in the previous example.
- Before each WDS instruction to write an USASCII character, clear the Write Request (WREQ) flag. If the teleprinter is being written to, the Write Request Delay (WRD) bit should be set prior to any character transmission to initialize the interface for 300 baud.
- Before each RDS instruction to read an USASCII character, clear the Read Request (RREQ) flag.

4.2.4.1 WRITE/READ KSR EXAMPLE. The following program writes 10 characters (each character is right-justified in a memory word) to the printer and then reads 10 characters from the keyboard. The data word of 0158_{16} associated with the first WDS instruction disables the I/O data bus interrupts from the data terminal, sets the transmission rate to 300 baud, and sets the Data Terminal Ready and Request to Send control bits. Refer to the WDS and RDS data word formats in figures 4-3 and 4-6, respectively, for aid in following the example.

```
          :
          :
          @LDA    =>158              CONTROL DATA,300 BAUD,TERMNL RDY
          WDS     5                  OUT TO INTERFACE
          DATA    0                  NO BUSY BIT (NO CHARACTER)
          LDX     =-10
WRITE     LDA     OUT+10,X           GET CHAR,RIGHT ADJUST
          AND     =>7F               MASK OUT MSB
          @IOR    =>5000             LOAD ASCII DATA,CLR WREQ
          WDS     5                  WRITE CHARACTER & CONTROL DATA
          DATA    >80                BUSY BIT
          BRU     $-2
          BIX     WRITE              LOOP BACK
          LDX     =-10
READ      @LDA    =>8000             CLR READ REQUEST
          WDS     5                  OUT TO INTERFACE
          DATA    0                  NO BUSY BIT
          RDS     5                  READ CHARACTER
          DATA    >80                BUSY BIT
          BRU     $-2
          AND     =>7F               SAVE 7-BIT LSB CHARACTER
          IOR     =>80               OR IN 8TH USASCII BIT (ONE), MARK PARITY
          STA     IN+10,X
          BIX     READ
OUT       DATA    ' C H A R A C T E R S '
IN        BSS     10
          :
          :
```

4.2.4.2 ASR SUBCOMMAND EXAMPLE. The following program puts cassette 1 in the record mode (and cassette 2 in the playback mode), loads cassette 1, and initiates the recording process. Note that the DLE character code (0010_{16}) accompanies each of the two-character functions. The program assumes a WDS instruction has already been executed and cassette 1 is positioned on the clear leader at the beginning of the tape.

```
      .
      .
      .
LDA   =>10      DLE USASCII CODE
@IOR  =>5000    LOAD USASCII DATA, CLR WREQ
WDS   5         ASR
DATA  >80       BUSY BIT
BRU   $-2       BRANCH IF NO DATA XFER
LDA   =>35      CASSETTE 1 TO RECORD MODE CODE
@IOR  =>5000    LOAD USASCII DATA, CLR WREQ
WDS   5         ASR
DATA  >80       BUSY BIT
BRU   $-2       BRANCH IF NO XFER
LDA   =>10      DLE USASCII CODE
@IOR  =>5000    LOAD USASCII DATA, CLEAR WREQ
WDS   5         ASR
DATA  >80       BUSY BIT
BRU   $-2       BRANCH IF NO XFER
LDA   =>33      LOAD CASSETTE 1 CODE
@IOR  =>5000    LOAD USASCII DATA, CLEAR WREQ
WDS   5         ASR
DATA  >80       BUSY BIT
BRU   $-2       BRANCH IF NO XFER
LDA   =>12      DC2, RECORD-ON CODE
@IOR  =>5000    LOAD USASCII DATA, CLEAR WREQ
WDS   5         ASR
DATA  >80       BUSY BIT
BRU   $-2       BRANCH IF NO XFER
      .
      .
      .
```

Additional examples of the ADC (single-character) functions are provided in the ASR read and write programs in the following paragraphs.

4.2.4.3 WRITE ASR EXAMPLE. The following program initiates the cassette recording process, writes 10 USASCII characters to the cassette, and terminates the cassette recording process. Note the alternate method of the busy bit test associated with the first WDS instruction. This program does not use interrupts.

```
                LDA     =>12            SEND DC2 (RECORD-ON)
                @IOR    =>5000          LOAD USASCII DATA, CLR WREQ
                WDS     5               TO ASR
                DATA    0               FROM REG A, NO BUSY BIT
                RDS     5
Replaces        DATA    0
Busy Bit        TABO    1
                BRU     $-3
                LDX     =-10
WRITE           LDA     OUT+10,X
                AND     =>7F            MASK BITS 0-TO-8 TO ZEROES
                @IOR    =>5000          LOAD USASCII DATA, CLR WREQ
                WDS     5
                DATA    >80
                BRU     $-2
                BIX     WRITE
                LDA     =>14            SEND DC4 (RECORD-OFF)
                @IOR    =>5000          LOAD USASCII DATA, CLR WREQ
                WDS     5
                DATA    >80
                BRU     $-2
                LDA     =>7F            WRITE OVER RECORD-OFF CHARACTER (DELETE)
                WDS     5
                DATA    >80
                BRU     $-2
OUT             DATA    ' C H A R A C T E R S '
```

4.2.4.4 READ ASR EXAMPLE. The following program reads 10 source records from a cassette, assuming each source record is less than or equal to 200 bytes. Note that the count of source records is maintained by keeping track of the number of DC3 characters (playback off) encountered. The DC3 character is used in programs that run under a monitor (supervisor), where each source line is terminated by a carriage return, line feed, DC3, and RUB OUT (delete). Stand-alone programs can use whatever conventions are necessary to end source lines. This program does not use interrupts; a wait loop is entered prior to reading each character from the cassette.

```
Designate Amt of
Records to Read
                    LDA     =10         AMT RECORDS TO READ
                    STA     COUNT       STORE AMT OF RECORDS
                    LDX     =0          CHARACTER STORAGE POINTER
                START   LDA     =>11        PLAYBACK ON (DC1), BEGIN READING RECORD
                    @IOR    =>D000      CLEAR RREQ & WREQ, WRITE CHARS
                READ    WDS     5           WRITE COMMAND & CHARACTER DATA
                    DATA    0           NO BUSY BIT (DC1 IS 1ST WDS)
Wait                RDS     5           READ DATA FROM ASR
for                 DATA    >80         BUSY BIT, REGISTER A
Data                BRU     $-2
                    AND     =>7F        MASK OUT 9 MSB
End-of-Record       CPL     =>13        CHK IF DC3 (PLYBCK OFF) READ
    Check           SNE                 DC3 READ? SKIP NEXT IF NOT READ
                    BRU     FINISH      BRANCH IF DC3 READ
                    IOR     =>80        7 BIT ASCII TO 8 BIT ASCII, MARK PARITY
                    STA     TABLE,X     STORE CHARACTER
                    RIN     X,X         INCREMENT X-REG
                    @LDA    =>8000      SET CLR READ REQ (CRR)
                    BRU     READ        RESTART CHARACTER CYCLE
Count of        FINISH  DMT     COUNT       DECREMENT COUNT, SKIP NEXT IF COUNT = 0
Records Read            BRU     START       GO TO NEXT RECORD
                    IDL     1           IDLE WHEN COUNT = 0
                COUNT   DATA    0           CONTAINS AMT OF RECORDS
                TABLE   BSS     1000        STORAGE AREA
```

APPENDIX A
USASCII CHARACTERS

APPENDIX A

USASCII CHARACTERS BY NUMERICAL SEQUENCE

000 0000	00	Null	010 0000	20	Space	100 0000	40	@	110 0000	60	`
000 0001	01	Start Heading	010 0001	21	!	100 0001	41	A	110 0001	61	a
000 0010	02	Start Text	010 0010	22	"	100 0010	42	B	110 0010	62	b
000 0011	03	End Text	010 0011	23	#	100 0011	43	C	110 0011	63	c
000 0100	04	End Transmission	010 0100	24	$	100 0100	44	D	110 0100	64	d
000 0101	05	Enquiry	010 0101	25	%	100 0101	45	E	110 0101	65	e
000 0110	06	Acknowledge	010 0110	26	&	100 0110	46	F	110 0110	66	f
000 0111	07	Bell	010 0111	27	'	100 0111	47	G	110 0111	67	g
000 1000	08	Backspace	010 1000	28	(	100 1000	48	H	110 1000	68	h
000 1001	09	Horizontal Tab	010 1001	29	)	100 1001	49	I	110 1001	69	i
000 1010	0A	Line Feed	010 1010	2A	*	100 1010	4A	J	110 1010	6A	j
000 1011	0B	Vertical Tab	010 1011	2B	+	100 1011	4B	K	110 1011	6B	k
000 1100	0C	Form Feed	010 1100	2C	,	100 1100	4C	L	110 1100	6C	l
000 1101	0D	Carriage Return	010 1101	2D	-	100 1101	4D	M	110 1101	6D	m
000 1110	0E	Shift Out	010 1110	2E	.	100 1110	4E	N	110 1110	6E	n
000 1111	0F	Shift In	010 1111	2F	/	100 1111	4F	O	110 1111	6F	o
001 0000	10	Data Link Escape	011 0000	30	0	101 0000	50	P	111 0000	70	p
001 0001	11	Device Control 1	011 0001	31	1	101 0001	51	Q	111 0001	71	q
001 0010	12	Device Control 2	011 0010	32	2	101 0010	52	R	111 0010	72	r
001 0011	13	Device Control 3	011 0011	33	3	101 0011	53	S	111 0011	73	s
001 0100	14	Device Control 4	011 0100	34	4	101 0100	54	T	111 0100	74	t
001 0101	15	Negative Acknowledge	011 0101	35	5	101 0101	55	U	111 0101	75	u
001 0110	16	Synchronous Idle	011 0110	36	6	101 0110	56	V	111 0110	76	v
001 0111	17	End Transmission Block	011 0111	37	7	101 0111	57	W	111 0111	77	w
001 1000	18	Cancel	011 1000	38	8	101 1000	58	X	111 1000	78	x
001 1001	19	End Medium	011 1001	39	9	101 1001	59	Y	111 1001	79	y
001 1010	1A	Substitute	011 1010	3A	:	101 1010	5A	Z	111 1010	7A	z
001 1011	1B	Escape	011 1011	3B	;	101 1011	5B	[	111 1011	7B	{
001 1100	1C	File Separator	011 1100	3C	<	101 1100	5C	\	111 1100	7C	|
001 1101	1D	Group Separator	011 1101	3D	=	101 1101	5D	]	111 1101	7D	}
001 1110	1E	Record Separator	011 1110	3E	>	101 1110	5E	^ or ↑	111 1110	7E	~
001 1111	1F	Unit Separator	011 1111	3F	?	101 1111	5F	_ or ←	111 1111	7F	Delete

NOTES: 1. In memory, the most significant bit is undefined and either a zero or one is acceptable.
2. The limited USASCII character set includes only 20_{16} through $5F_{16}$.

USER'S RESPONSE SHEET

Manual Title: Model 980 Computer Assembly Language Input/Output (961961-9734)

Date of Manual: 1 October 1974 Date of This Letter: _____

User: _____ Office/Dept. No.: _____

Company: _____

Street Address: _____

City/State/Zip: _____

Please list any discrepancy found in this manual by page, paragraph, figure, or table number in the following space. If there are any other suggestions that you wish to make, feel free to include them. Thank you.

Location in Manual	Comment/Suggestion
_____	_____
	_____
	_____
	_____
_____	_____
	_____
	_____
	_____
_____	_____
	_____
	_____
	_____

CUT ALONG LINE

NO POSTAGE NECESSARY IF MAILED IN U.S.A.
FOLD ON TWO LINES (LOCATED ON REVERSE SIDE), STAPLE AND MAIL

First Class
PERMIT NO. 3135
Austin, Texas

BUSINESS REPLY MAIL
No Postage Necessary if Mailed in the United States

Postage Will Be Paid by

TEXAS INSTRUMENTS INCORPORATED
DIGITAL SYSTEMS DIVISION

P.O. BOX 2909 · AUSTIN, TEXAS 78767

Attn: TECHNICAL PUBLICATIONS, MS 2146

Sales and Service Offices of Texas Instruments are located throughout the United States and in major countries overseas. Contact the Digital Systems Division, Texas Instruments Incorporated, P.O. Box 1444, Houston, Texas 77001, or call (713) 494-5115, for the location of the office nearest to you.

TEXAS INSTRUMENTS
INCORPORATED