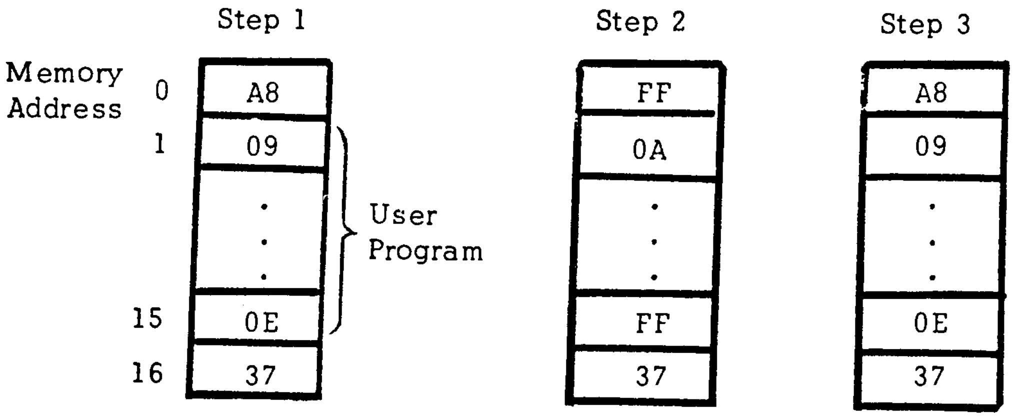
4.0 SYSTEM MONITOR

The INTELLEC 8I System Monitor enables the operator to easily manipulate the contents of memory, read and produce paper tapes, execute programs, and read or initialize PROMs.

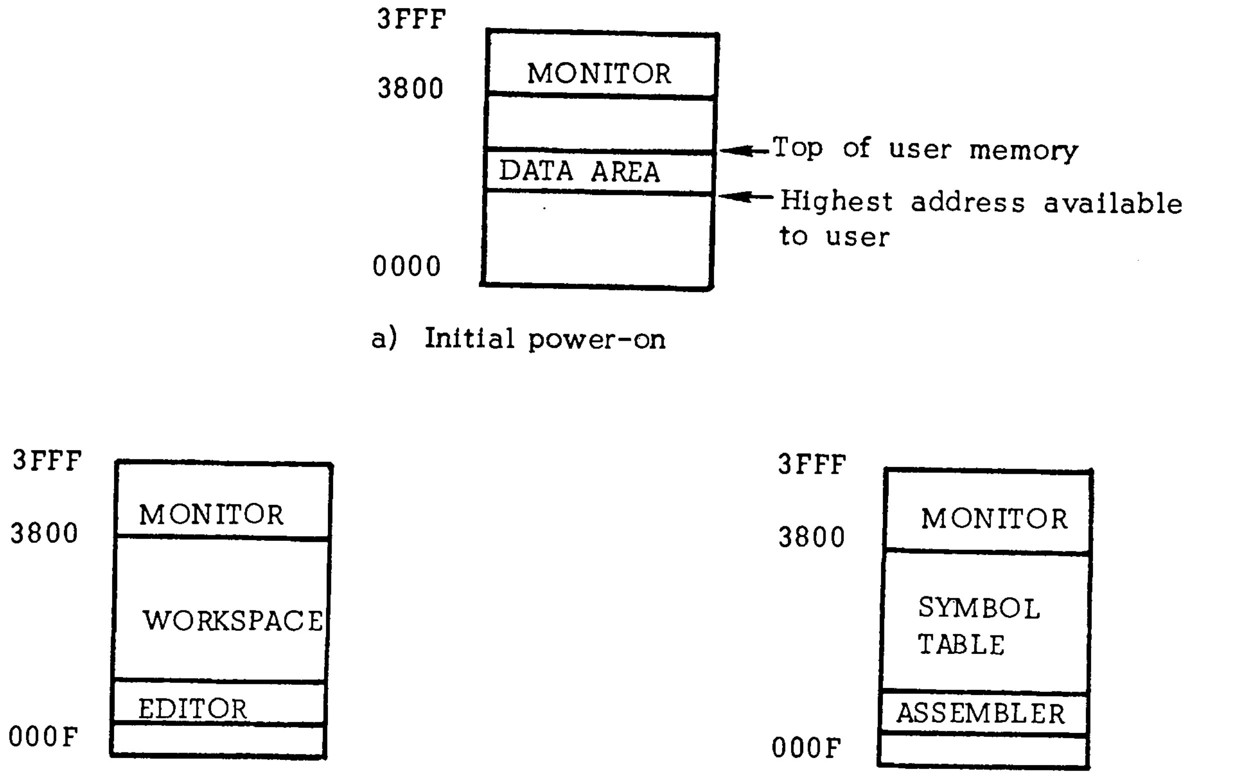
The System Monitor, and all INTELLEC 8I system software in general, use the first 16 memory locations for storage of temporary data. Therefore if the operator runs a program beginning in these locations, and then uses the System Monitor, Text Editor, or Assembler, he must re-load the first 16 bytes of his program before running it again. Alternatively, programs could be written beginning at location 16. Then system programs and user programs could be executed in any order, without requiring the re-load operation.



Step 1 shows the first 17 bytes of memory while a user program beginning at location 0 is being run.

Step 2 shows these bytes after the System Monitor has been run, changing the first 16 bytes of memory (0-15). Bytes 16 on are unchanged.

Step 3 shows that the user has reloaded the first 16 bytes of his program, and is ready to execute the program again.



a) Initial power-on

b) Load editor and generate source code in workspace.

c) Load assembler and convert source code to object code.

Figure 4-1: Normal Use and Execution Sequence for System Software

The System Monitor is the operator's interface to INTELLEC 8I, and controls loading and execution of the editor and assembler, loading and execution of user programs, and to some extent the debugging of user programs. Figure 4-1 illustrates memory utilization during various stages of system software use. While the System Monitor is running, it uses an area at the top of memory for data storage and scratch work. The user can determine how large this area is via the I/O subroutine MEMCK (described in section 3.3.2), which returns the highest memory address available for the user.

4.1 SYSTEM MONITOR IMPLEMENTATION AND EXECUTION

4.1.1 SYSTEM MONITOR IMPLEMENTATION

The INTELLEC 8I System Monitor program is implemented on eight PROM modules, which are pre-installed into each INTELLEC 8I. This allows System Monitor to be used with great ease, as it is not necessary to wait for lengthy paper-tape loading operations. All that is required to go on-line with System Monitor is to start the INTELLEC 8I, turn the Teletype on-line, perform a program jump to the execution address of System Monitor (3800), and begin execution.

4.1.2 STARTING SYSTEM MONITOR

To begin operating System Monitor, place a "Jump to 3800" command (C30038 H) into the first three locations in the INTELLEC 8I. Press the RESET button and the INTELLEC 8I will automatically jump to the starting address of System Monitor, 3800. (For the exact sequence, see Section 3.2.2).

4.2 SYSTEM MONITOR OPERATION AND COMMANDS

A monitor command consists of a single letter typed into the Teletype keyboard followed by a number of arguments, possibly none. The arguments are separated, if there are more than one, by spaces or commas. A command is terminated and executed by typing a carriage return or space, depending upon the command.

4.2.1 A COMMAND (ASSIGN I/O DEVICES)

The format of the A command is:

```
A ldev = pdev
```

ldev is one of the four logical I/O devices CONSOLE, READER, PUNCH or LIST. Only the first character is required, the rest being optional.

pdev is one of the four legal physical I/O devices corresponding to ldev, as shown in Table 4-1.

Description: The physical device pdev is assigned to logical device ldev. The following table gives all possible A commands, with optional characters shown in lower case. Numeric values for pdev indicate user-supplied devices.

```
AConsole = Tty
AConsole = Crt.
AConsole = Batch
AConsole = 1

AReader = Tty
AReader = Ptr
AReader = 1
AReader = 2

APunch = Tty
APunch = Ptp
APunch = 1
APunch = 2

AList = Tty
AList = Crt
AList = 1
AList = 2
```

Table 4-1: I/O Assignment Commands

For a description of the I/O system, see Section 3.3.

Error Conditions:

If a selected physical device has not been readied or does not exist, the monitor's execution will be undefined, usually executing an infinite loop. This may be corrected by readying the device or pressing system reset.

4.2.2 B COMMAND (BNPF OUTPUT)

The format of the B command is:

B low address , high address

Low address is a valid 16-bit memory address.

High address is a valid 16-bit memory address equal to or greater than low address.

Description: The B command outputs the content of memory, from (low address) through (high address), to the Teletype printer/punch in BNPF form. Six inches of null tape are punched before and after the data.

BNPF form represents the data word in pure binary form. A B is punched to indicate the beginning of a word, P is punched to indicate a "1" bit, N is punched to indicate a "0" bit, and F is to indicate the end of a word. Thus, 'A5'H would be shown as follows:

```
                    BPNPNNPNPF

        Beginning    10100101    End
```

For reference, the address is punched in <u>decimal</u> form every fourth byte. The reason the addresses are punched in decimal rather than in hexadecimal is to avoid confusion between the character 'B' punched on the tape to indicate the beginning of a byte, and the character 'B' which would have to be punched as part of some hexadecimal addresses. (See Section 4.2.7 for the legal BNPF program tape format.)

Example: If memory bytes 1 through 4 contain 2B 00 FF 55 then the command:

```
.B1,4(Cr)
```

typed into the teletype will cause the following to be punched and printed at the teletype:

```
1      BNNPNPNPPF      BNNNNNNNNF      BNPPPPPPPF
4      BNPNPNPNPF
```

Error conditions:

1. If low address or high address is greater than 16 bits, only the last 4 hex digits of the argument will be used as the address.

Example: The command:

```
.B3C03B00,3CA3C01 (Cr)
```

is equivalent to the command:

```
.B3B00,3C01 (Cr)
```

2. If low address is greater than high address, only the one byte at low address will be punched.

Example: The command:

```
.B4A00,3A02 (Cr)
```

is equivalent to the command:

```
.B4A00,4A00 (Cr)
```

3. Non-existent memory is equivalent to a string of bytes all containing FF H.

Example: If memory addresses 2000 H- 2003 H are not present in a particular INTELLEC 8I system, then the command:

```
.B2000,2003 (Cr)
```

will cause the following to be printed and punched at the teletype:

```
8192    BPPPPPPPPF    BPPPPPPPPF    BPPPPPPPPF
8195    BPPPPPPPPF
```

4. If low address or high address contains an invalid character, or if high address is omitted, the monitor will immediately type '*(Cr)(lf).' and await the next command.

Example: If the user tries to enter the number 3AGE as low address, the following will be printed:

```
.B3AG*
.
```

4.2.3 C COMMAND (COMPARE PROM WITH MEMORY)

The format of the C command is:

```
C address
```

Address is a valid 16 bit memory address.

Description: The C command causes the monitor to read a PROM plugged into the console programming socket and compare its contents with a 256 byte area of memory starting at address. If the contents of the PROM and memory match completely, no message is given. Otherwise, for each location where a mismatch occurs, the memory address, memory data, and PROM data are printed.

Example: If the command

```
.C600(Cr)
```

produces the printout

```
0600     A3          FF
0605     6D          05
```

then the PROM contains the same data as memory locations 0600 H through 06FF H, except at location 0600 H and 0605 H. Memory contains A3 H and 6D H, while the PROM contains FF H and 05 H, respectively.

Error Conditions:

1. If address is greater than 16 bits, only the last 4 hex digits are used as the address.

Example: The command:

```
.C0ABC059(Cr)
```

is equivalent to the command:

```
.CC059(Cr)
```

2. If address specifies a non-existent memory range, the PROM contents are compared to 256 bytes of FF H.

Example: If locations 2000 H through 20FF H are non-existent, and the command

```
.C2000(Cr)
```

is issued, every PROM byte not equal to FF H will be printed.

3. If a PROM is not installed in the socket, memory contents are compared to 256 bytes of FF H.

Example: If a PROM is not plugged into the socket, any C command will print every byte in the memory range not equal to FF H.

4. If an invalid character is specified in address, the monitor will type '*(Cr)(lf).' and await the next command.

Example: If the user attempts to enter 3GAB as the address, the following will be printed:

```
.C3G8*
.
```

4.2.4 D COMMAND (DISPLAY DATA)

The format of the D command is:

```
D low address , high address
```

Low address is a valid 16 bit memory address.

High address is a valid 16 bit memory address equal to or greater than low address.

Description: Upon execution of this command, memory data from (low address) to (high address) is displayed upon the list device (normally the Teletype). Data are displayed in hexadecimal form. Up to sixteen bytes per line are printed, preceded by the hexadecimal address of the first byte of that line. A carriage return is forced after a byte having a low order digit of F in.its memory address is printed.

Example: Enter at the teletype the command:

```
.D10F,123(Cr)
```

and the teletype will type back:

```
010F   AA
0110   BB CC DD EE FF 11 22 33 44 55 66 77 88 99 AB CD
0120   EF 12 34 56
```

where memory locations 010F through 0123 are assumed to contain

```
AA BB CC DD EE FF 11 22 33 44 55 66 77 88 99 AB CD EF 12 34 56
```

The D command should be used only to examine memory contents. To punch the memory contents onto a paper tape, either the B command or the W command should be used. These commands produce a punched paper tape in the proper formats, while the D command causes only a simple sequence of characters to be output.

Error conditions:

1. If low address or high address is greater than 16 bits, only the last 4 hex digits of the argument will be used as the address.

Example: The command

```
.D30010,AB0013(Cr)
```

is equivalent to the command

```
.D0010,0013(Cr)
```

2. If low address is greater than high address, only the one byte at low address will be displayed.,

Example: The command:

```
.D10,6
```

is equivalent to the command

```
.D10,10
```

3. Non-existent memory is equivalent to a string of bytes all containing FF H.

Example: If memory address 2000 H- 2010 H are invalid, then the command:

```
.D2000,2010
```

will cause the teletype to print:

```
2000    FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
2010    FF
```

4. If low address or high address contains an invalid character, or if high address is omitted, the monitor will immediately type '*(Cr)(lf). and await the next command.

Example: If the user attempts to enter the number 0G as an address, the following will be printed:

```
.D0G*
.
```

4.2.5 F COMMAND (FILL MEMORY WITH CONSTANT)

The format of the F command is:

```
F low address , high address , data
```

Low address is a valid 16 bit memory address.

High address is a valid 16 bit memory address equal to or greater than low address.

Data is an 8 bit data value.

Description: Execution of this command causes memory locations (low address) through (high address) to be filled with the constant (data).

Example: The command:

```
.F7,14,AA(Cr)
```

will set bytes 0007 through 0014 equal to AA H.

```
0007    AA AA AA AA AA AA AA AA AA
0010    AA AA AA AA AA
```

Error Conditions:

1. If low address or high address is greater than 16 bits (or data is greater than 8 bits), only the last 4 (or 2) hex digits will be used.

Example: The command:

```
.F7AB0007,0014,FFACAA(Cr)
```

is equivalent to the command:

```
.F0007,0014,AA(Cr)
```

2. If low address is greater than high address, data will replace only the byte at low address.

Example: If locations 7,8, and 9 contain AA H, BB H, and CC H, execution of the command:

```
.F7,1,33(Cr)
```

will cause memory to appear as follows:

```
0007    33    BB    CC
```

3. If a non-existent memory address is specified, this command has no effect.

4. If low address, high address, or data contain an invalid character, the monitor will immediately type '*(Cr)(lf).' and await the next command.

Example: If the user tries to enter BQ as data, the following will be printed:

```
.F0012,14,BQ*
.
```

4.2.6 G COMMAND (GO TO)

The format of the G command is:

```
G address, bkptl, bkpt2
```

Address,bkptl, and bkpt2 are valid 16 bit hexadecimal memory addresses.

Description: The G command causes program control to be transferred to location address. If either bkptl or bkpt2 is specified, a breakpoint will be set in the program at the corresponding address(es). The specified address <u>must</u> correspond to the <u>first</u> byte of a program instruction. If either breakpoint is encountered during program execution, the System Monitor will save all program status (CPU registers and condition bits), clear all existing breakpoints, and take control. The user may then examine and/or modify registers or memory, or use any other monitor commands. This feature allows the user to debug portions of a program.

If address is not specified, the program status is restored and the saved value of the program counter is used as the new starting address.

NOTE: Encountering a breakpoint may be simulated at any time by setting the console Address/Instruction/Data switches to CFH and pressing the INT switch. All breakpoints will be cleared, status will be saved, and the System Monitor will take control.

Example: The command:

```
G24A
```

will cause program execution to begin at location 24AH, with no breakpoints being set.

The command:

```
G,12C
```

will cause a breakpoint to be set at 12CH, and program execution to resume at the address indicated by the saved value of the program counter.

The command:

```
G
```

will cause program execution to resume at the address indicated by the saved value of the program counter, with all status restored and no breakpoints set.

Error Conditions:

1. If address is greater than 16 bits, only the last 4 hex digits of the argument will be used as the address.

Example: The command:

```
.G3C0010(Cr)
```

is equivalent to the command

```
.G0010(Cr)
```

2. If address is a non-existent memory address, the system will attempt to transfer control and then stop with no response. The System Monitor must then be manually restarted.

4.2.7 H COMMAND (HEXADECIMAL ARITHMETIC)

The format of the H command is:

```
.H number , number Sp
```

Number is a 16 bit hexadecimal number.

Description: The H command is designed to aid the user in performing hexadecimal arithmetic while using the System Monitor. It causes the sum and difference its arguments to be printed in two's complement hexadecimal form. This command is terminated by a space, rather than by a carriage return.

Example:

```
.H1E,5C       007A            FFC2
```

Error Conditions:

1. If either number is greater than 16 bits, only the last 4 hex digits are used.

Example: The command:

```
.H00ABC,23Sp
```

is equivalent to the command:

```
.H0ABC,23Sp
```

2. If number contains an invalid character, the monitor will immediately type '*(Cr)(lf).' and await the next command.

Example: If the user attempts to enter 01P, the following will be printed:

```
.H01P*
.
```

4.2.8 L COMMAND (LOAD BNPF TAPE)

The format of the L command is:

```
L low address , high address
```

Low address is a valid 16 bit memory address.

High address is a valid 16 bit memory address greater than or equal to low address.

Description: The L command loads a paper tape punched in BNPF format into memory starting at low address and continuing through high address. BNPF format represents a word in pure binary form. A B is punched to indicate the beginning of a word. Following the B, exactly eight P's and N's must be punched, P indicating a "1" bit and N indicating a "0" bit. The ninth character following the B must be an F, indicating the end of the word. All characters following the F are ignored until another B is encountered. This allows comments (not including the letter B) to appear between data words in BNPF format. For instance, the two hexadecimal data words 3AF0 could be represented on a BNPF tape as follows:

```
BNNPPPNPNF * COMMENT * BPPPPNNNNF
```

Example:

```
.L0,FF(Cr)
```

When executed, this would load the first page (256 bytes) from paper tape into memory.

Error Conditions:

1. If low address or high address is greater than 16 bits, only the last 4 hex digits of the argument will be used as the address.

Example: The command:

```
.L0,AF00FF(Cr)
```

is equivalent to the command:

```
.L0,00FF(Cr)
```

2. If low address is greater than high address, only one byte will be read from the paper tape and transferred into memory at low address.

Example: The command:

```
.L10,0(Cr)
```

is equivalent to the command:

```
.L,0,0(Cr)
```

3. If non-existent memory is referenced, the tape will be read, but no other action will occur.

4. If low address or high address contains an invalid character, the monitor will immediately type ' *(Cr)(lf).' and await the next command.

Example: If the user attempts to enter 3GAB as high address, the following will be printed:

```
.L0,3G*
.
```

4.2.9 M COMMAND (MOVE MEMORY)

The format of the M command is:

```
.M low address , high address , destination address
```

Low address is a valid 16 bit memory address.

High address is a valid 16 bit memory address equal to or greater than low address.

Destination address is a valid 16 bit memory address.

Description: The M command causes the block of memory from low address through high address to be moved to the locations in memory beginning at destination address.

Example: If memory appears as follows:

LOCATIONS		DATA
0300-0304	contain	01020304
0200-0204	contain	A1A2A3A4

then the command:

```
M200,204,300
```

will cause the following:

LOCATIONS		DATA
0300-0304	contain	A1A2A3A4
0200-0204	contain	A1A2A3A4

Note: The movement is performed byte by byte: the byte at low address is moved to destination address, then low address+1 is moved to destination address+1, etc. Therefore, the MOVE command may be used to fill memory with a byte or sequence of bytes.

Example: If location 0300 H contains FF H, the command

```
.M300,310,301(Cr)
```

will cause locations 300 through 310 to contain FF H. The FF at 300 is moved to 301, then the byte at 301 (which is now FF), is moved to 302, and so on.

Error Conditions:

1. If any address is greater than 16 bits, only the last 4 hex digits are used as the address.

Example: The command:

```
.M00302,303,00405(Cr)
```

is equivalent to the command:

```
.M302,303,405(Cr)
```

2. If low address is greater than high address, only one byte will be moved from low address to destination address.

Example: The command:

```
.M300,2F0,100(Cr)
```

is equivalent to the command:

```
.M300,300,100(Cr)
```

3. If low address through high address specifies a non-existent range of memory, bytes of FF H will be moved to the memory locations specified by destination address.

Example: If locations 2000 H through 2005 are non-existent, the command:

```
.M2000,2005,100(Cr)
```

will cause locations 0100 H through 0105 H to contain FF H.

4. If an invalid character is entered in an address, the monitor will type '*(Cr)(lf).' and await the next command.

Example: If the user attempts to enter 0BAG as the destination address, the following will be printed:

```
.M100,10F,0BAG*
.
```

4.2.10 R COMMAND (READ HEX FILE)

The format of the R command is:

R bias address

Bias Address is a 16 bit two's complement hexadecimal number.

Description: This command loads paper tape punched in hexadecimal format (using the W command) into memory. The address at which the tape is loaded is determined by adding the address punched on the tape to the bias address using two's complement arithmetic. The bias may be negative, but in this case must be in two's complement form. If the tape was produced using an E command with a non-zero entry point address (see section 4.2.13), control will be transferred to that location in memory. Otherwise, the System Monitor will remain in control and request another command.

Example: If a tape was punched which began at location 0100 H, the following command:

```
.RFFB0(Cr)
```

will cause the tape to be read and loaded into location 50 H. (1000+FFB0= 50).

NOTE: If an error occurs while reading the tape (such as a checksum error), the monitor will immediately stop reading the tape, type '*(Cr)(Lf).' and await the next command. The operation may be retried by backing up the tape to any point before the last colon and issuing another R command, since each data word specifies the address at which it is to be loaded. The monitor will read up to the first colon it encounters, and then begin loading data.

Note that this means that, if you wish to change data in locations in memory, it is not necessary to regenerate an entirely new tape with the change; instead you may read in the original tape, then read in a patch tape which reloads only the erroneous locations.

Error Conditions:

1. If the bias address is greater than 16 bits, only the last 4 hex digits are used as the bias address.

Example: The command:

```
.R00FFB0 (Cr)
```

is equivalent to the command:

```
.RFFB0(Cr)
```

2. If an invalid character is present in the bias address, the monitor will immediately type '*(Cr)(lf).' and await the next command.

Example: If the user attempts to enter G00 as a bias address, the following will be printed:

```
[redacted]RG[redacted]
```

4.2.11 S COMMAND (SUBSTITUTE MEMORY)

The S command is used to display and/or modify the contents of individual memory locations. It is used as follows:

1. Type an S, followed by the hexadecimal address of the first memory location you wish to display. Type space.

2. The data from the selected address is displayed, followed by a dash (-).

3. To modify memory, type in the new data followed by a space or a carriage return. If you do not wish to modify the contents of that location, do not type any data in, but only type a space or carriage return.

4. If a space was typed in step 3, the next memory location will be displayed as in step 2. If a carriage return was typed, operation will be returned to the System Monitor.

Example: The contents of the first four bytes of memory is 00 Al CE FF.
You wish to change it to 00 A3 CE 11.

```
.S0000Sp00Sp Al - A3Sp CE - Sp FF - 11Cr
```

User entries areunshaded. Printback is shaded.

Error Conditions:

1. If address is greater than 16 bits, or the data to be substituted is greater than 8 bits, only the last 4 or 2 hex digits respectively are used.

Example: The following sequence is equivalent to the previous example:

```
.S0AB0000Sp 00 - Sp Al - BA3Sp CE - Sp FF - 011Cr
```

2. If an invalid character is encountered, the monitor will immediately type '*(Cr)(lf). and await the next command.

4.2.12 X COMMAND (EXAMINE AND MODIFY REGISTERS)

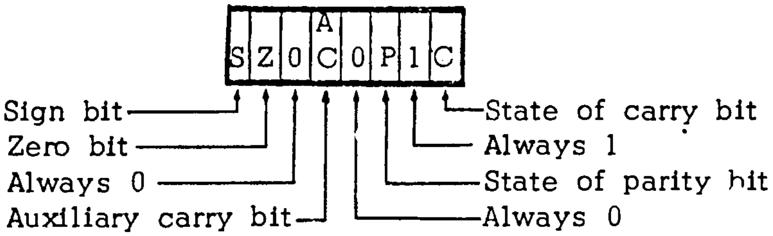
The format of the X command is:

```
X reg ident
```

Reg ident is a single character specifying a CPU register as follows:

A = A register
B = B register
C = C register
D = D register
E = E register
F = Flag byte, displayed in the form as it is stored by the instruction PUSH PSW
H = H register
L = L register
M = H and L registers combined (16 bits)
P = Program counter (16 bits)
S = Stack pointer (16 bits)

Note: The format of the flag byte F is:



Description: The X command is used to display and/or modify CPU registers. It operates similar to the S command, as follows:

1. Type an X, followed by the register identifier.
2. The data from the selected register is displayed, followed by a dash (-). Four hexadecimal digits are displayed for M, P, and S; two hex digits for the other register identifiers.
3. To modify the register, type in the new data followed by a space or a carriage return. If you do not wish to modify the register, type only the space or carriage return.
4. If a space was typed in step 3, the next register in alphabetical order is displayed. If carriage return was typed, the X command is terminated. If a space is typed after register S has been displayed, the command is terminated, this being the last register identifier in the list.

Example: The A, B, C, and D registers contain AAH, BBH, CCH, and DDH, respectively. You wish to change the B and C registers to 00H and FFH, respectively.

```
XASp AA- Sp BB-00Sp CC- FFSp DD-Cr
```

Note: Values set by the X-command will become the actual contents of the registers after execution of the next GO command.

The values displayed by the X-command are the contents of the registers prior to the execution of the last breakpoint set by the GO command. These displayed values, however, will reflect any changes of register "contents" made by the execution of X-commands since this last breakpoint.

Type an X followed by a carriage return. The contents of all internal registers will be printed out in the same form as above.

Note: Values set by the X-command will become the actual contents of the registers after execution of the next GO command.

The values displayed by the X-command are the contents of the registers prior to the execution of the last breakpoint set by the GO command. These displayed values, however, will reflect any changes of register "contents" made by the execution of X-commands since this last breakpoint.

Error Conditions:

1. If the data to be substituted is greater than 16 bits for registers M, P, S,or 8 bits for the other register identifiers, only the last 4 or 2 hex digits respectively are used.

2. If an invalid register identifier or character is encountered, the monitor will immediately type '*(Cr)(Lf).' and await the next command.

4.2.13 E COMMAND (END FILE)

The format of the E command is:

E address

Address is a valid 16 bit memory address.

Description: The E command causes an end-of-file mark and sixty null characters to be punched at the end of a hexadecimal output file. The end of file mark is hexadecimal record of length 00. (See Appendix D). If address is 0 or absent, the R command which loads the file will return control to the System Monitor. If address is non-zero, the R command will transfer control to that memory address immediately after loading the file.

4.2.14 W COMMAND (WRITE MEMORY)

The format of the W command is:

W low address ; high address

Low address is a valid 16 bit memory address.

High address is a valid 16 bit memory address equal to or greater than low address.

Description: The W command is used to output memory locations low address through high address to the system punch device in hexadecimal format.
A series of W commands may be issued in order to punch various non-contiguous memory locations onto a continuous strip of tape.

Any series of W commands should be terminated with an E command in order to punch a termination character, so that when the tape is read it will be handled properly.

Example: If memory locations 1 through 3 contain 53F8EC, the command:

```
.W0001,0003(Cr)
```

produces:

```
:0300010053F8ECC5
```

(See Appendix D for an explanation of paper tape format.)

Error Conditions:

1. If low address or high address is greater than 16 bits, only the last 4 hex digits of the argument will be used as the address.

Example: The command:

```
WAB0010,100(Cr)
```

is equivalent to the command:

```
.W0010,100(Cr)
```

2. If low address is greater than high address, only the one byte at low address will be punched.

Example: The command:

```
.W10,0(Cr)
```

is equivalent to the command:

```
.W10,10(Cr)
```

3. Non-existent memory is equivalent to a string of bytes all containing FF H.

4. An invalid character in either address will cause the monitor to print '*(Cr)(lf).' and await the next command.

Example: If the user attempts to enter 3Z as low address, the following will be printed:

```
.W3Z*
.
```

4.2.15 N COMMAND (NULL PUNCH)

The N command consists only of the letter N followed by a carriage return and causes 60 null characters to be punched.

4.2.16 T COMMAND (TRANSFER FROM PROM TO MEMORY)

The format of the T command is:

```
T address
```

Address is a valid 16 bit memory address.

Description: The T command causes the contents of a PROM in the front panel programming pocket to be transferred to memory beginning at address and continuing for 256 bytes.

Error Conditions:

1. If address is greater than 16 bits, only the last 4 hex digits will be used as the address.

Example: The command:

```
.TAB0100(Cr)
```

is equivalent to the command:

```
.T100(Cr)
```

2. If address specifies a non-existent memory range, no data is transferred.

3. If no PROM is present in the socket, 256 bytes of FF H will be transferred into memory.

Example: If no PROM is present, and the command:

```
.T100(Cr)
```

is issued, memory locations 0100 H through 01FF H will contain FF H.

4. If address contains an invalid character, the monitor will immediately type '*(Cr)(lf). and await the next command.

4.2.17 P COMMAND (PROGRAM PROM)

The format of the P command is:

```
P low address , high address , PROM address
```

Low Address is a valid 16 bit memory address.

High Address is a valid 16 bit memory address equal to or greater than low address.

PROM Address is an 8 bit data value.

Description: The P command causes the contents of memory from low address to high address to be programmed into a PROM module in the PROM programming socket.

Data are programmed into the PROM beginning at the PROM address.

Before attempting to program a PROM, ensure that the PROMPRGR POWER switch is ON, and that the PROM module is inserted firmly into the socket.

Error Conditions:

1. If low address or high address is greater than 16 bits, or if PROM address is greater than 8 bits, only the last 4 or 2 hex digits respectively will be used.

Example: The command:

```
.PAB0100,1FF,3FF(Cr)
```

is equivalent to the command

```
.P100,1FF,FF(Cr)
```

2. If low address is greater than high address, only one byte of data at low address will be transferred.

Example: The command:

```
.P400,300,0(Cr)
```

is equivalent to the command

```
.P400,400,0(Cr)
```

3. If low and high address refer to a non-existent memory range, a string of bytes containing FF H will be transferred to the PROM.

4. If an invalid character is typed in any address, the monitor will immediately type '*(Cr)(lf).' and await the next command.

Example: If the user attempts to type 3R as the PROM address, the following will be printed:

```
▨P400,410,3R▨
▨
```

5. If the PROM operation fails, the monitor will type a '$' and re-try the operation. Up to 3 retries will be performed, and, if still unsuccessful, the monitor will type the PROM address at which the failure occurred.