SSM 8080 MONITOR V1

FEATURES

Available in eight 1702's or two 2708's UV PROMs.

Commands:

D----Display a block of memory data between any two addresses within 64K of RAM.

S----Display or Substitute a byte for another anywhere in memory.

F----Fill an area in memory with a specific byte.

M----Move a block of memory data, of any length, from one location in RAM to another.

B----Dump memory to a punch device in a binary format. This format is compatible with MITS paper-tape format.

L----Load a binary tape with an optional bias address.

W----Dump memory to a punch device in hex checksum format. This format is compatible with INTEL paper-tape format.

R----Load a hex checksum tape with an optional bias address.

G----Go to an address and start execution with the option of up to two software breakpoints being preset for a powerful software debugging tool.

X----Display the CPU's registers after a breakpoint is encountered. Any register data can be modified at will and the program resumed.

A----SSM 8Ø8Ø Monitor can logically communicate with any four different physical devices for each of its four main I/O entry points.

Entry points for:

AC = Console I/O Devices
AR = Reader Devices
AP = Punch Devices
AL = Listing Devices

The monitor may be easily adapted to any hardware configuration through redefinition of the "System Configuration Package" section of the monitor.

Starting address of the SSM 8Ø8Ø Monitor is at FØØØ hex.

CONTENTS

1.0 SOLID STATE MUSIC 8080 MONITOR V-1

The Solid State Music 8080 Monitor constitutes a powerful central operating system for an 8080 based microcomputer system. The monitor may be easily adapted to any hardware configuration through redefinition of a separately assembled "System Configuration Package".

Together the monitor and the system configuration package occupy less that 2048 bytes of memory (ROM) and require less than 64 bytes of RAM for scratchpad.

The monitor provides an interface to system peripheral devices through externally referenced subroutines and logical I/O assignments. Operator commands are inplemented for loading, examining, changing and dumping memory, executing user programs with or without breakpoints, examining/changing register contents, assigning physical devices to logical devices, etc. Memory may be loaded from or dumped to external files in either hexadecimal or binary format.

The system configuration package serves as an interface between the standard monitor and the hardware configuration at a particular installation. Consideration is given to a variety of peripheral devices and memory organizations.

Operator Commands

Operator commands to the monitor consist of a single alphabetic character input to the logical console input device and, usually, one or more parameters. When two or more parameters are used they are separated by a comma or a space. Parameters are hexadecimal values consisting of the last four (or two, as appropriate) hexadecimal characters before the parameter is terminated. Leading zeros are assumed.

The command line is terminated by a carriage return in most cases. Any character other than those specified aborts the command.

Memory Commands

D Command (display memory)

D'Low Address','High Address'

Memory from 'Low Address'through 'High Address' is displayed on the logical console output device. If 'High Address' is equal to or smaller than 'Low Address' only the byte at 'Low Address' is displayed.

Memory Commands (continued)

Data bytes are displayed in hexadecimal, 16 bytes per line. The beginning address of each line is displayed before the line.

S Command (substitute memory)

S'Address'

The byte at location 'Address' is displayed on the logical console device followed by a - character. The operator responds with one or more characters from the logical console input device. If the input character is space or comma, the contents of the next location are displayed, etc. If one or more hexadecimal digits are input before the space or comma, the specified value will replace the display value in the memory location. A carriage return terminates the command line.

F Command (fill memory)

F'Low Address','High Address','Data'

Memory from 'Low Address' through 'High Address' is filled with 'Data'. If 'High Address' is equal to or smaller than 'Low Address' only location 'Low Address' is filled.

M Command (move memory)

M'Low Address','High Address','Dest Address'

Data from 'Low Address' through 'High Address' are moved to memory beginning at 'Dest Address'. If 'High Address' is equal to or smaller than 'Low Address' only the byte at 'Low Address' is moved. If 'Dest Address' is between 'Low Address' and 'High Address' the data from 'Low Address' to 'Dest Address' is repeated to fill the destination field.

External File Commands

B Command (binary dump memory)

B'Low Address','High Address'

Data from 'Low Address' through 'High Address' are output to the logical punch device in a binary format compatible with mits paper tape format. ' High Address' must be equal to or greater than 'Low Address', with one exception: If 'High Address' is zero and end-of-file record is output specifying 'Low Address' as the entry point address.

External File Commands (continued)

L Command (load memory, binary)

L'Bias Address'

Data in binary format are read from the logical reader device and stored in memory at the load address specified in the input file plus 'Bias Address'. When an end-of-file record is encountered control is transferred to the specified entry point address of zero terminates loading and the monitor remains in control.

W Command (write memory, hexadecimal)

W'Low Address','High Address'

Data from 'Low Address' through 'High Address' are output to the logical punch device in a hexadecimal format compatible with Intel paper tape format. 'High Address' must be equal to or greater than 'Low Address' with one exception: If 'High Address' is zero an end-of-file record is output specifying 'Low Address' as the entry point address.

R Command (read to memory, hexadecimal)

R'Bias Address'

Data in hexadecimal format are read from the logical reader device and stored in memory at the load address specified in the input file plus 'Bias Address'. When an end-of-file record is encountered control is transferred to the specified entry point address if it is non-zero. An entry point address of zero terminates loading and the monitor remains in control.

N Command (null output)

N

Sixty null bytes (00H) are output to the logical punch device.

K Command (copy files)

Bytes are continuously read from the logical reader device and output to the logical punch device. This process continues until manually interrupted, I.E., by resetting the system.

Execute and Debug Commands

G Command (goto)

G'Address','Breakpoint 1','Breakpoint 2'

If 'Address' is specified, control is transferred to 'Address'. If 'Address' is not specified, control is transferred to the address of the last encountered breakpoint, after program status (CPU registers and flags) is restored.

If 'Breakpoint 1 ' or 'Breakpoint 2 ' is specified, breakpoints (RST 1) replace the bytes at corresponding addresses. These addresses must contain the first byte of an instruction. If breakpoints are specified, a jump instruction is stored at location 0008H to return control to the monitor when a breakpoint, or any RST 1 instruction, is excuted. At this point the monitor will save the program status and restore the bytes replaced by any known breakpoints. Twelve bytes of the users stack are used in this process. The program counter in the saved program status is decremented, so that program execution may be resumed with the instruction formerly replaced by the breakpoint. Monitor commands may then be used to display/modify memory or CPU registers, etc.

When the monitor is entered normally, I.E., by other than breakpoint execution, recording of existing breakpoints is destroyed. Therefore, if breakpoints are set, but not executed before the monitor is re-entered, the contents of the bytes containing those breakpoints must be manually restored.

RST 1 instructions other than known breakpoints may be used as pseudo-breakpoints, subject to certain restrictions. The jump instruction must be stored at location 0008H by previously settins a normal breakpoint. RST 1 instructions other than known breakpoints may be executed thorugh normal program execution (RST 1 stored as part executing program) or instruction jam (interrupt). When such a RST 1 instruction is encountered, the monitor saves the program status and resets known breakpoints. However, the program counter in the saved program status is not decremented, so program execution may be resumed at the next instruction.

X Command (register display/modify)

X'Register'

Register contents as of the last encountered breakpoint are displayed. 'Register' may be specified as A,B,C,D,E,F (flags), H, L,M (H and L combined), P (program counter) or S (stack pointer). The registers are displayed, in the above order, beginning with

Execute and Debug Commands (continued)

the specified 'Register'. After each register content is displayed, the operator may change it by supplying the new value followed by a space or comma. If no new value is entered the old value is retained and the next register is displayed. The command is terminated by a carriage return, or display/modification of register S.

If 'Register' is not specified, all registers are displayed without operator intervention.

Miscellaneous Commands

H Command (hexadecimal arithmetic)

H'Operand1','Operand2'

The sum and difference of 'Operand1' and Operand2' are displayed in hexadecimal on the logical console output device.

A Command (assign I/O devices)

A'Logical'='Physical'

Physical device 'Physical' is assigned to logical device 'Logical'. 'Logical' may be any of the four system logical devices, I.E., console,reader, punch, or list. Only the first character of the device name is required. 'Physical' may be 0,1,2 or 3. The actual physical device assigned is determined by the installation configuration as specified in the "System Configuration Package".

Externally Referenced Subroutines

Several externally referenced subrountines and entry points are provided for use by user written software. These routines, their addresses relative to the beginning of the monitor (LOC), and their functions are listed below.

MONTR LOC

Normal entry point of the monitor. The I/O configuration is initialized so physical device zero is assigned to each logical device.

CI LOC+3

Console input. One character is read from the logical console input device and returned in register A. All registers other than A and F are preserved.

Externally Referenced Subroutines (continued)

RI LOC+6

Reader input. One byte is read from the logical reader device and returned in register A. All registers other than A and F are preserved. If no byte is available from the reader, the carry flag is set upon return.

CO LOC+9

Console output. The byte in register C is output to the logical console output device. All resisters other than A and F are preserved.

PO LOC+0CH

Punch output. The byte in register C is output to the logical punch device. All registers other than A and F are preserved.

LO LOC+0FH

List output. The byte in register C is output to the logical list device. All registers other than A and F are preserved.

CSTS LOC+12H

Console status. The logical console input device is checked for input availability. Register A is set to zero and the zero flag set true if no input is available. Register A is set non-zero and the zero flag set false if a character is available. All registers other than A and F are preserved.

IOCHK LOC+15H

The current setting of IOBYT is returned in register A. IOBYT is the byte of RAM used to record the current logical device/ physical device assignments.

Bits 0,1 Record the physical device currently assigned to the logical console device.

Bits 2,3 Record the physical device currently assigned to the logical reader device.

Bits 4,5 Record the physical device currently assigned to the logical punch device.

Bits 6.7 Record the physical device currently assigned to the logical list device.

Externally Referenced Subroutines (continued)

All registers other A and F are preserved.

IOSET LOC+18H

The contents of register C are stored in IOBYT, thus altering the logical/physical device assignments. All registers are preserved..

MEMCK LOC+1BH

The upper limit of RAM available to user programs is returned in registers B (most significant) and A (least significant). The address returned is that of the first byte not available to the user. All registers other than A,B and F are preserved.

STRNG LOC+1EH

The string of characters pointed to by registers H,L is output to the logical console output device. The character string is terminated before a null character, or after A character with bit 7 set. Registers B,D,E are preserved.

REENT LOC+21H

Alternate entry point to the monitor. The current I/O configuration is not altered when the monitor is entered at this point.

System Configuration Package

The system configuration package is used to "Tailor" the monitor to a particular system hardware/software configuration. The package consists of a table establishing the I/O configuration, several entry points for establishing certain memory addresses, and driver routines for system I/O devices. The table and entry points, their addresses relative to the beginning of the package (SCP) and their functions are listed below.

IOTAB SCP

This table consists of six sub-tables, each consisting of four entries of two bytes each. Each sub-table entry is the entry point address of a driver routine for a particular physical device. Each sub-table lists the driver routines for the four physical devices that may be assigned to a locical device. Sub-tables are included for console status, console input, console output, reader input, punch output, and list output.

System Configuration Package (continued)

ADSCS SCP+30H

Subroutine to set registers D,E to the address of the top of the monitor scratchpad area of RAM. This subroutine is called without the use of the stack.

ADSCR SCP+33H

Subroutine to set registers H,L to the address of the top of the monitor scratchpad area of RAM. This subroutine is called in the normal way, I.E. by a call instruction. The monitor requires 64 bytes of RAM for scratchpad. Of these 64 bytes, the top four are unused by the monitor itself, and may be used to store IOBYT and up to three other bytes such as those associated with a CRT output device.

ADIOB SCP+36H

Subroutine to set registers H,L to the address of IOBYT, the byte of RAM where the current I/O configuration is recorded.

ADUST SCP+39H

Subroutine to set registers H,L to the address of the users stack. This is a default value used only until replaced by the operator (X command) or by execution of a breakpoint. It is expected that user routines will initialize the stack pointer when they are first entered.

```
                ;       8080 MONITOR V1.0

                ;       PROGRAMMER: C. E. OHME
                ;                   (415)657-8326


                ;
                ;       SYSTEM CONFIGURATION INTERFACE

F600            SCP     EQU     0F600H
F600            IOTAB   EQU     SCP
F630            ADSCS   EQU     SCP+48
F633            ADSCR   EQU     SCP+51
F636            ADIOB   EQU     SCP+54
F639            ADUST   EQU     SCP+57

                ;       ASCII CHARACTERS

000D            CR      EQU     0DH
000A            LF      EQU     0AH

F000                    ORG     0F000H

                ;       EXTERNALLY REFERENCED SUBROUTINE
                ;       JUMP TABLE

F000 C324F0             JMP     BEGIN
F003 C3D6F0             JMP     CI
F006 C3E0F0             JMP     RI
F009 C3D1F0             JMP     CO
F00C C3E5F0             JMP     PO
F00F C3EAF0             JMP     LO
F012 C3DBF0             JMP     CSTS
F015 C30EF1             JMP     IOCHK
F018 C315F1             JMP     IOSET
F01B C3B1F1             JMP     MEMCK
F01E C31EF1             JMP     STRNG
F021 0E00       REENT:  MVI     C,0
F023 210000             LXI     H,0
F024                    ORG     $-2

F024 0E01       BEGIN:  MVI     C,1
F026 112CF0             LXI     D,$+6
F029 C330F6             JMP     ADSCS
F02C EB                 XCHG
F02D 0615               MVI     B,ENDX-EXIT-1
F02F 1176F2             LXI     D,ENDX-1
F032 1B         BG1:    DCX     D
F033 1A                 LDAX    D
F034 2B                 DCX     H
F035 77                 MOV     M,A
F036 05                 DCR     B
```

```
F037 C232F0            JNZ     BG1
F03A F9                SPHL
F03B CD39F6            CALL    ADUST
F03E E5                PUSH    H
F03F 2600              MVI     H,0
F041 E5                PUSH    H
F042 E5                PUSH    H
F043 E5                PUSH    H
F044 79                MOV     A,C
F045 B7                ORA     A
F046 CA4EF0            JZ      BG2
F049 CD36F6            CALL    ADIOB
F04C 3600              MVI     M,0
F04E 218FF0    BG2:    LXI     H,VERS
F051 CD1EF1            CALL    STRNG
F054 B7                ORA     A

               ;       COMMAND RETURN POINT

F055 D266F0    CMNDR:  JNC     START

               ;       ERROR RETURN

F058 CD33F6    LER:    CALL    ADSCR
F05B 11E3FF            LXI     D,EXIT-ENDX-7
F05E 19                DAD     D
F05F F9                SPHL
F060 219DF0            LXI     H,ERM
F063 CD1EF1            CALL    STRNG

               ;       INPUT AND EXECUTE NEXT COMMAND

F066 FB        START:  EI
F067 CD46F1            CALL    CRLF
F06A 0E2E              MVI     C,'.'
F06C CDD1F0            CALL    CO
F06F CD2FF1            CALL    TI
F072 D641              SUI     'A'
F074 FA58F0            JM      LER
F077 FE18              CPI     'X'-'A'+1
F079 F258F0            JP      LER
F07C 87                ADD     A
F07D 2155F0            LXI     H,CMNDR
F080 E5                PUSH    H
F081 219FF0            LXI     H,TBL
F084 1600              MVI     D,0
F086 5F                MOV     E,A
F087 19                DAD     D
F088 7E                MOV     A,M
F089 23                INX     H
F08A 66                MOV     H,M
F08B 6F                MOV     L,A
F08C 0E02              MVI     C,2
F08E E9                PCHL
```

```
F08F 0D0A4D4F VERS:   DB      CR,LF,'MONITOR V1.','0' OR 80H
F093 4E49544F
F097 52205631
F09B 2EB0
F09D 0AAA     ERM:    DB      LF,'*' OR 80H

              ;       COMMAND JUMP TABLE

F09F 77F2     TBL:    DW      ASSIGN
F0A1 C0F2             DW      BIN
F0A3 58F0             DW      LER
F0A5 1CF3             DW      DISP
F0A7 58F0             DW      LER
F0A9 3CF3             DW      FILL
F0AB 4CF3             DW      GOTO
F0AD A0F3             DW      HEXN
F0AF 58F0             DW      LER
F0B1 58F0             DW      LER
F0B3 BCF3             DW      KOPY
F0B5 C6F3             DW      LOAD
F0B7 0CF4             DW      MOVE
F0B9 1EF4             DW      NULL
F0BB 58F0             DW      LER
F0BD 58F0             DW      LER
F0BF 58F0             DW      LER
F0C1 23F4             DW      READ
F0C3 89F4             DW      SUBS
F0C5 58F0             DW      LER
F0C7 58F0             DW      LER
F0C9 58F0             DW      LER
F0CB B1F4             DW      WRITE
F0CD 3DF5             DW      X

              ;       UTILITY SUBROUTINES

F0CF 0E20     BLK:    MVI     C,' '

F0D1 CDEFF0   CO:     CALL    IOBR
F0D4 0110             DB      1,10H

F0D6 CDEFF0   CI:     CALL    IOBR
F0D9 0108             DB      1,8

F0DB CDEFF0   CSTS:   CALL    IOBR
F0DE 0100             DB      1,0

F0E0 CDEFF0   RI:     CALL    IOBR
F0E3 0418             DB      4,18H

F0E5 CDEFF0   PO:     CALL    IOBR
F0E8 0320             DB      3,20H
```

```
F0EA CDEFF0    LO:     CALL    IOBR
F0ED 0228              DB      2,28H

F0EF E3        IOBR:   XTHL
F0F0 C5                PUSH    B
F0F1 46                MOV     B,M
F0F2 23                INX     H
F0F3 4E                MOV     C,M
F0F4 CD36F6            CALL    ADIOB
F0F7 7E                MOV     A,M
F0F8 0F                RRC
F0F9 07        IOB1:   RLC
F0FA 07                RLC
F0FB 05                DCR     B
F0FC C2F9F0            JNZ     IOB1
F0FF E606              ANI     6
F101 81                ADD     C
F102 4F                MOV     C,A
F103 2100F6            LXI     H,IOTAB
F106 09                DAD     B
F107 7E                MOV     A,M
F108 23                INX     H
F109 66                MOV     H,M
F10A 6F                MOV     L,A
F10B C1                POP     B
F10C E3                XTHL
F10D C9                RET
               IOCHK:
F10E E5                PUSH    H
F10F CD36F6            CALL    ADIOB
F112 7E                MOV     A,M
F113 E1                POP     H
F114 C9                RET
               IOSET:
F115 E5                PUSH    H
F116 F5                PUSH    PSW
F117 CD36F6            CALL    ADIOB
F11A 71                MOV     M,C
F11B F1                POP     PSW
F11C E1                POP     H
F11D C9                RET
               STRNG:
F11E 7E                MOV     A,M
F11F E67F              ANI     7FH
F121 C8                RZ
F122 4F                MOV     C,A
F123 7E                MOV     A,M
F124 B7                ORA     A
F125 FAD1F0            JM      CO
F128 CDD1F0            CALL    CO
F12B 23                INX     H
F12C C31EF1            JMP     STRNG
```

```
                TI:
F12F CDD6F0             CALL    CI
F132 E67F               ANI     7FH
F134 C5                 PUSH    B
F135 4F                 MOV     C,A
F136 CDD1F0             CALL    CO
F139 79                 MOV     A,C
F13A C1                 POP     B
F13B C9                 RET
                CONV:
F13C E60F               ANI     0FH
F13E C690               ADI     90H
F140 27                 DAA
F141 CE40               ACI     40H
F143 27                 DAA
F144 4F                 MOV     C,A
F145 C9                 RET
                CRLF:
F146 0E0D               MVI     C,CR
F148 CDD1F0             CALL    CO
F14B 0E0A               MVI     C,LF
F14D C3D1F0             JMP     CO
                EXPR1:
F150 0E01               MVI     C,1
                EXPR:
F152 210000             LXI     H,0000H
F155 CD2FF1     EX0:    CALL    TI
F158 47         EX1:    MOV     B,A
F159 CDD0F1             CALL    NIBBL
F15C DA68F1             JC      EX2
F15F 29                 DAD     H
F160 29                 DAD     H
F161 29                 DAD     H
F162 29                 DAD     H
F163 B5                 ORA     L
F164 6F                 MOV     L,A
F165 C355F1             JMP     EX0
F168 E3         EX2:    XTHL
F169 E5                 PUSH    H
F16A 78                 MOV     A,B
F16B CDE5F1             CALL    P2C
F16E D276F1             JNC     EX3
F171 0D                 DCR     C
F172 C258F0             JNZ     LER
F175 C9                 RET
F176 C258F0     EX3:    JNZ     LER
F179 0D                 DCR     C
F17A C252F1             JNZ     EXPR
F17D C9                 RET
                EXF:
F17E 0E01               MVI     C,1
F180 210000             LXI     H,0000H
F183 C358F1             JMP     EX1
```

```
                HILO:
F186 23                 INX     H
F187 7C                 MOV     A,H
F188 B5                 ORA     L
F189 37                 STC
F18A C8                 RZ
F18B 7B                 MOV     A,E
F18C 95                 SUB     L
F18D 7A                 MOV     A,D
F18E 9C                 SBB     H
F18F C9                 RET
                LADR:
F190 7C                 MOV     A,H
F191 CD95F1             CALL    LBYTE
F194 7D                 MOV     A,L
                LBYTE:
F195 F5                 PUSH    PSW
F196 0F                 RRC
F197 0F                 RRC
F198 0F                 RRC
F199 0F                 RRC
F19A CD9EF1             CALL    HXD
F19D F1                 POP     PSW
                HXD:
F19E CD3CF1             CALL    CONV
F1A1 C3D1F0             JMP     CO
                LEADS:
F1A4 0604               MVI     B,4
                LEAD:
F1A6 0E00               MVI     C,0
F1A8 CDE5F0             CALL    PO
F1AB 05                 DCR     B
F1AC C2A6F1             JNZ     LEAD
F1AF B7                 ORA     A
F1B0 C9                 RET
                MEMCK:
F1B1 E5                 PUSH    H
F1B2 D5                 PUSH    D
F1B3 CD33F6             CALL    ADSCR
F1B6 EB                 XCHG
F1B7 210000             LXI     H,0
F1BA 24         MEM0:   INR     H
F1BB 7E                 MOV     A,M
F1BC 2F                 CMA
F1BD 77                 MOV     M,A
F1BE BE                 CMP     M
F1BF 2F                 CMA
F1C0 77                 MOV     M,A
F1C1 CABAF1             JZ      MEM0
F1C4 2B                 DCX     H
F1C5 44                 MOV     B,H
F1C6 7C                 MOV     A,H
F1C7 BA                 CMP     D
```

```
F1C8 3EC0               MVI     A,0C0H
F1CA D1                 POP     D
F1CB E1                 POP     H
F1CC C8                 RZ
F1CD 3EFF               MVI     A,0FFH
F1CF C9                 RET
                NIBBL:
F1D0 D630               SUI     '0'
F1D2 D8                 RC
F1D3 C6E9               ADI     '0'-'G'
F1D5 D8                 RC
F1D6 C606               ADI     6
F1D8 F2DEF1             JP      NI0
F1DB C607               ADI     7
F1DD D8                 RC
F1DE C60A       NI0:    ADI     10
F1E0 B7                 ORA     A
F1E1 C9                 RET
                PCHK:
F1E2 CD2FF1             CALL    TI
                P2C:
F1E5 FE20               CPI     ' '
F1E7 C8                 RZ
F1E8 FE2C               CPI     ','
F1EA C8                 RZ
F1EB FE0D               CPI     CR
F1ED 37                 STC
F1EE C8                 RZ
F1EF 3F                 CMC
F1F0 C9                 RET

                ;       BREAKPOINT ENTRY POINT

F1F1 E5         RESTRT: PUSH    H
F1F2 D5                 PUSH    D
F1F3 C5                 PUSH    B
F1F4 F5                 PUSH    PSW
F1F5 CD33F6             CALL    ADSCR
F1F8 11EBFF             LXI     D,EXIT-ENDX+1
F1FB 19                 DAD     D
F1FC EB                 XCHG
F1FD 210A00             LXI     H,000AH
F200 39                 DAD     SP
F201 0604               MVI     B,4
F203 EB                 XCHG
F204 2B         RST0:   DCX     H
F205 72                 MOV     M,D
F206 2B                 DCX     H
F207 73                 MOV     M,E
F208 D1                 POP     D
F209 05                 DCR     B
F20A C204F2             JNZ     RST0
F20D C1                 POP     B
```

```
F20E 0B              DCX     B
F20F F9              SPHL
F210 211400          LXI     H,TLOC
F213 39              DAD     SP
F214 7E              MOV     A,M
F215 91              SUB     C
F216 23              INX     H
F217 C21FF2          JNZ     RST1
F21A 7E              MOV     A,M
F21B 90              SUB     B
F21C CA2DF2          JZ      RST3
F21F 23      RST1:   INX     H
F220 23              INX     H
F221 7E              MOV     A,M
F222 91              SUB     C
F223 C22CF2          JNZ     RST2
F226 23              INX     H
F227 7E              MOV     A,M
F228 90              SUB     B
F229 CA2DF2          JZ      RST3
F22C 03      RST2:   INX     B
F22D 210F00  RST3:   LXI     H,LLOC
F230 39              DAD     SP
F231 73              MOV     M,E
F232 23              INX     H
F233 72              MOV     M,D
F234 23              INX     H
F235 23              INX     H
F236 71              MOV     M,C
F237 23              INX     H
F238 70              MOV     M,B
F239 C5              PUSH    B
F23A 219DF0          LXI     H,ERM
F23D CD1EF1          CALL    STRNG
F240 E1              POP     H
F241 CD90F1          CALL    LADR
F244 211400          LXI     H,TLOC
F247 39              DAD     SP
F248 1602            MVI     D,2
F24A 4E      RST4:   MOV     C,M
F24B 3600            MVI     M,0
F24D 23              INX     H
F24E 46              MOV     B,M
F24F 3600            MVI     M,0
F251 23              INX     H
F252 79              MOV     A,C
F253 B0              ORA     B
F254 CA59F2          JZ      RST5
F257 7E              MOV     A,M
F258 02              STAX    B
F259 23      RST5:   INX     H
F25A 15              DCR     D
F25B C24AF2          JNZ     RST4
F25E C366F0          JMP     START
```

```
                        ;       SCRATCHPAD TEMPLATE

F261 D1         EXIT:   POP     D
F262 C1                 POP     B
F263 F1                 POP     PSW
F264 E1                 POP     H
F265 F9                 SPHL
F266 FB                 EI
F267 210000             LXI     H,0
F26A             HLX    EQU     $-2
F26A C30000             JMP     0
F26D             PCX    EQU     $-2
F26D 0000       T1A:    DW      0       ;TRAP 1 ADDR
F26F 00                 DB      0       ;TRAP 1 INST
F270 0000               DW      0       ;TRAP 2 ADDR
F272 00                 DB      0       ;TRAP 2 INST
F273 0000               DW      0       ;VIDEO PNTR
F275 00                 DB      0       ;VIDEO HOLD
F276 00                 DB      0       ;IOBYT
                        ENDX:
0005            ALOC    EQU     5
0003            BLOC    EQU     3
0002            CLOC    EQU     2
0001            DLOC    EQU     1
0000            ELOC    EQU     0
0004            FLOC    EQU     4
0010            HLOC    EQU     HLX-EXIT+9
000F            LLOC    EQU     HLX-EXIT+8
0013            PLOC    EQU     PCX-EXIT+9
0007            SLOC    EQU     7
0014            TLOC    EQU     T1A-EXIT+8

                        ;       COMMAND IMPLEMENTATION

                        ;       ASSIGN COMMAND

F277 CD2FF1     ASSIGN: CALL    TI
F27A 0600               MVI     B,0
F27C FE43               CPI     'C'
F27E CA93F2             JZ      AS1
F281 04                 INR     B
F282 FE52               CPI     'R'
F284 CA93F2             JZ      AS1
F287 04                 INR     B
F288 FE50               CPI     'P'
F28A CA93F2             JZ      AS1
F28D 04                 INR     B
F28E FE4C               CPI     'L'
F290 C2BEF2             JNZ     EREXT
F293 CD2FF1     AS1:    CALL    TI
F296 FE3D               CPI     '='
F298 C293F2             JNZ     AS1
```

```
F29B CD2FF1          CALL    TI
F29E D630            SUI     '0'
F2A0 6F              MOV     L,A
F2A1 FABEF2          JM      EREXT
F2A4 FE04            CPI     4
F2A6 F2BEF2          JP      EREXT
F2A9 2603            MVI     H,3
F2AB 05      AS2:    DCR     B
F2AC FAB4F2          JM      AS3
F2AF 29              DAD     H
F2B0 29              DAD     H
F2B1 C3ABF2          JMP     AS2
F2B4 EB      AS3:    XCHG
F2B5 CD36F6          CALL    ADIOB
F2B8 7E              MOV     A,M
F2B9 B2              ORA     D
F2BA AA              XRA     D
F2BB B3              ORA     E
F2BC 77              MOV     M,A
F2BD C9              RET
F2BE 37      EREXT:  STC
F2BF C9              RET

             ;       BINARY COMMAND

F2C0 CD52F1  BIN:    CALL    EXPR
F2C3 CD46F1          CALL    CRLF
F2C6 D1              POP     D
F2C7 E1              POP     H
F2C8 7A      BIN0:   MOV     A,D
F2C9 B3              ORA     E
F2CA C2D7F2          JNZ     B0
F2CD CDA4F1          CALL    LEADS
F2D0 0E78            MVI     C,78H
F2D2 CD11F3          CALL    PHL
F2D5 B7              ORA     A
F2D6 C9              RET
F2D7 7B      B0:     MOV     A,E
F2D8 95              SUB     L
F2D9 7A              MOV     A,D
F2DA 9C              SBB     H
F2DB D8              RC
F2DC 7B      B1:     MOV     A,E
F2DD 95              SUB     L
F2DE 4F              MOV     C,A
F2DF 7A              MOV     A,D
F2E0 9C              SBB     H
F2E1 3F              CMC
F2E2 D0              RNC
F2E3 0C              INR     C
F2E4 C2E9F2          JNZ     B2
F2E7 0EFF            MVI     C,0FFH
F2E9 D5      B2:     PUSH    D
```

```
F2EA 59                 MOV     E,C
F2EB CDA4F1             CALL    LEADS
F2EE 0E3C               MVI     C,3CH
F2F0 CDE5F0             CALL    PO
F2F3 4B                 MOV     C,E
F2F4 CD11F3             CALL    PHL
F2F7 7C                 MOV     A,H
F2F8 85                 ADD     L
F2F9 57                 MOV     D,A
F2FA 4E         B3:     MOV     C,M
F2FB 23                 INX     H
F2FC 7A                 MOV     A,D
F2FD 81                 ADD     C
F2FE 57                 MOV     D,A
F2FF CDE5F0             CALL    PO
F302 1D                 DCR     E
F303 C2FAF2             JNZ     B3
F306 4A                 MOV     C,D
F307 CDE5F0             CALL    PO
F30A D1                 POP     D
F30B 7D                 MOV     A,L
F30C B4                 ORA     H
F30D C8                 RZ
F30E C3DCF2             JMP     B1
F311 CDE5F0     PHL:    CALL    PO
F314 4D                 MOV     C,L
F315 CDE5F0             CALL    PO
F318 4C                 MOV     C,H
F319 C3E5F0             JMP     PO

                ;       DISPLAY COMMAND

F31C CD52F1     DISP:   CALL    EXPR
F31F D1                 POP     D
F320 E1                 POP     H
F321 CD46F1     DI0:    CALL    CRLF
F324 CD90F1             CALL    LADR
F327 CDCFF0     DI1:    CALL    BLK
F32A 7E                 MOV     A,M
F32B CD95F1             CALL    LBYTE
F32E CD86F1             CALL    HILO
F331 3F                 CMC
F332 D0                 RNC
F333 7D                 MOV     A,L
F334 E60F               ANI     0FH
F336 C227F3             JNZ     DI1
F339 C321F3             JMP     DI0

                ;       FILL COMMAND

F33C 0C         FILL:   INR     C
F33D CD52F1             CALL    EXPR
F340 C1                 POP     B
```

```
F341 D1              POP     D
F342 E1              POP     H
F343 71       FI0:   MOV     M,C
F344 CD86F1          CALL    HILO
F347 D243F3          JNC     FI0
F34A B7              ORA     A
F34B C9              RET

             ;       GOTO COMMAND

F34C E1       GOTO:  POP     H
F34D CDE2F1          CALL    PCHK
F350 DA98F3          JC      G03
F353 CA72F3          JZ      G00
F356 CD7EF1          CALL    EXF
F359 D1              POP     D
F35A 211300          LXI     H,PLOC
F35D 39              DAD     SP
F35E 72              MOV     M,D
F35F 2B              DCX     H
F360 73              MOV     M,E
F361 78              MOV     A,B
F362 FE0D            CPI     CR
F364 CA98F3          JZ      G03
F367 3EC3            MVI     A,(JMP 0)
F369 320800          STA     8
F36C 21F1F1          LXI     H,RESTRT
F36F 220900          SHLD    9
F372 1602     G00:   MVI     D,2
F374 211400          LXI     H,TLOC
F377 39              DAD     SP
F378 E5       G01:   PUSH    H
F379 CD50F1          CALL    EXPR1
F37C 58              MOV     E,B
F37D C1              POP     B
F37E E1              POP     H
F37F 78              MOV     A,B
F380 B1              ORA     C
F381 CA8EF3          JZ      G02
F384 71              MOV     M,C
F385 23              INX     H
F386 70              MOV     M,B
F387 23              INX     H
F388 0A              LDAX    B
F389 77              MOV     M,A
F38A 23              INX     H
F38B 3ECF            MVI     A,(RST 1)
F38D 02              STAX    B
F38E 7B       G02:   MOV     A,E
F38F FE0D            CPI     CR
F391 CA98F3          JZ      G03
F394 15              DCR     D
F395 C278F3          JNZ     G01
```

```
F398 CD46F1   GO3:    CALL    CRLF
F39B 210800           LXI     H,0008H
F39E 39               DAD     SP
F39F E9               PCHL

                      ;       HEXADECIMAL COMMAND

F3A0 CD52F1   HEXN:   CALL    EXPR
F3A3 D1               POP     D
F3A4 E1               POP     H
F3A5 CD46F1           CALL    CRLF
F3A8 E5               PUSH    H
F3A9 19               DAD     D
F3AA CD90F1           CALL    LADR
F3AD CDCFF0           CALL    BLK
F3B0 E1               POP     H
F3B1 7D               MOV     A,L
F3B2 93               SUB     E
F3B3 6F               MOV     L,A
F3B4 7C               MOV     A,H
F3B5 9A               SBB     D
F3B6 67               MOV     H,A
F3B7 CD90F1           CALL    LADR
F3BA B7               ORA     A
F3BB C9               RET

                      ;       COPY COMMAND

F3BC CDE0F0   KOPY:   CALL    RI
F3BF 4F               MOV     C,A
F3C0 CDE5F0           CALL    PO
F3C3 C3BCF3           JMP     KOPY

                      ;       LOAD COMMAND

F3C6 CD50F1   LOAD:   CALL    EXPR1
F3C9 C1               POP     B
F3CA CDE0F0   L1:     CALL    RI
F3CD D8               RC
F3CE FE3C             CPI     3CH
F3D0 CADFF3           JZ      L2
F3D3 FE78             CPI     78H
F3D5 C2CAF3           JNZ     L1
F3D8 CD01F4           CALL    LHL
F3DB D8               RC
F3DC B5               ORA     L
F3DD C8               RZ
F3DE E9               PCHL
F3DF CDE0F0   L2:     CALL    RI
F3E2 D8               RC
F3E3 5F               MOV     E,A
F3E4 CD01F4           CALL    LHL
F3E7 D8               RC
```

```
F3E8 85                 ADD     L
F3E9 57                 MOV     D,A
F3EA 09                 DAD     B
F3EB CDE0F0     L3:     CALL    RI
F3EE D8                 RC
F3EF 77                 MOV     M,A
F3F0 82                 ADD     D
F3F1 57                 MOV     D,A
F3F2 23                 INX     H
F3F3 1D                 DCR     E
F3F4 C2EBF3             JNZ     L3
F3F7 CDE0F0             CALL    RI
F3FA D8                 RC
F3FB BA                 CMP     D
F3FC CACAF3             JZ      L1
F3FF 37                 STC
F400 C9                 RET
F401 CDE0F0     LHL:    CALL    RI
F404 D8                 RC
F405 6F                 MOV     L,A
F406 CDE0F0             CALL    RI
F409 D8                 RC
F40A 67                 MOV     H,A
F40B C9                 RET

                ;       MOVE COMMAND

F40C 0C         MOVE:   INR     C
F40D CD52F1             CALL    EXPR
F410 C1                 POP     B
F411 D1                 POP     D
F412 E1                 POP     H
F413 7E         MV0:    MOV     A,M
F414 02                 STAX    B
F415 03                 INX     B
F416 CD86F1             CALL    HILO
F419 D213F4             JNC     MV0
F41C B7                 ORA     A
F41D C9                 RET

                ;       NULL COMMAND

F41E 063C       NULL:   MVI     B,60
F420 C3A6F1             JMP     LEAD

                ;       READ COMMAND

F423 CD50F1     READ:   CALL    EXPR1
F426 E1                 POP     H
F427 CDE0F0     RED0:   CALL    RI
F42A D8                 RC
F42B E67F               ANI     7FH
F42D D63A               SUI     ':'
```

```
F42F C227F4            JNZ     RED0
F432 57                MOV     D,A
F433 E5                PUSH    H
F434 CD65F4            CALL    BYTE
F437 CA59F4            JZ      RED2
F43A 5F                MOV     E,A
F43B CD65F4            CALL    BYTE
F43E 47                MOV     B,A
F43F CD65F4            CALL    BYTE
F442 4F                MOV     C,A
F443 09                DAD     B
F444 CD65F4            CALL    BYTE
F447 CD65F4    RED1:   CALL    BYTE
F44A 77                MOV     M,A
F44B 23                INX     H
F44C 1D                DCR     E
F44D C247F4            JNZ     RED1
F450 CD65F4            CALL    BYTE
F453 E1                POP     H
F454 CA27F4            JZ      RED0
F457 37                STC
F458 C9                RET
F459 CD65F4    RED2:   CALL    BYTE
F45C 67                MOV     H,A
F45D CD65F4            CALL    BYTE
F460 C1                POP     B
F461 6F                MOV     L,A
F462 B4                ORA     H
F463 C8                RZ
F464 E9                PCHL
F465 CD76F4    BYTE:   CALL    RNBBL
F468 07                RLC
F469 07                RLC
F46A 07                RLC
F46B 07                RLC
F46C 4F                MOV     C,A
F46D CD76F4            CALL    RNBBL
F470 B1                ORA     C
F471 4F                MOV     C,A
F472 82                ADD     D
F473 57                MOV     D,A
F474 79                MOV     A,C
F475 C9                RET
F476 CDE0F0    RNBBL:  CALL    RI
F479 DA85F4            JC      RNBER
F47C E67F              ANI     7FH
F47E CDD0F1            CALL    NIBBL
F481 DA85F4            JC      RNBER
F484 C9                RET
F485 E1        RNBER:  POP     H
F486 E1                POP     H
F487 E1                POP     H
F488 C9                RET
```

```
                ;       SUBSTITUTE COMMAND

F489 CD50F1     SUBS:   CALL    EXPR1
F48C CDE5F1             CALL    P2C
F48F E1                 POP     H
F490 D8                 RC
F491 7E         SU0:    MOV     A,M
F492 CD95F1             CALL    LBYTE
F495 0E2D               MVI     C,'-'
F497 CDD1F0             CALL    CO
F49A CDE2F1             CALL    PCHK
F49D 3F                 CMC
F49E D0                 RNC
F49F CAADF4             JZ      SU1
F4A2 E5                 PUSH    H
F4A3 CD7EF1             CALL    EXF
F4A6 D1                 POP     D
F4A7 E1                 POP     H
F4A8 73                 MOV     M,E
F4A9 78                 MOV     A,B
F4AA FE0D               CPI     CR
F4AC C8                 RZ
F4AD 23         SU1:    INX     H
F4AE C391F4             JMP     SU0

                ;       WRITE COMMAND

F4B1 CD52F1     WRITE:  CALL    EXPR
F4B4 CD46F1             CALL    CRLF
F4B7 D1                 POP     D
F4B8 E1                 POP     H
F4B9 7A         WRIT0:  MOV     A,D
F4BA B3                 ORA     E
F4BB C2DBF4             JNZ     W0
F4BE CD33F5             CALL    PEOL
F4C1 0E3A               MVI     C,':'
F4C3 CDE5F0             CALL    PO
F4C6 AF                 XRA     A
F4C7 57                 MOV     D,A
F4C8 CD1CF5             CALL    PBYTE
F4CB CD17F5             CALL    PADR
F4CE 3E01               MVI     A,1
F4D0 CD1CF5             CALL    PBYTE
F4D3 AF                 XRA     A
F4D4 92                 SUB     D
F4D5 CD1CF5             CALL    PBYTE
F4D8 C31EF4             JMP     NULL
F4DB 7B         W0:     MOV     A,E
F4DC 95                 SUB     L
F4DD 7A                 MOV     A,D
F4DE 9C                 SBB     H
F4DF D8                 RC
```

```
F4E0 7B        WRI0:   MOV     A,E
F4E1 95                SUB     L
F4E2 4F                MOV     C,A
F4E3 7A                MOV     A,D
F4E4 9C                SBB     H
F4E5 3F                CMC
F4E6 D0                RNC
F4E7 79                MOV     A,C
F4E8 E60F              ANI     0FH
F4EA 3C                INR     A
F4EB D5                PUSH    D
F4EC 5F                MOV     E,A
F4ED 1600              MVI     D,0
F4EF CD33F5            CALL    PEOL
F4F2 0E3A              MVI     C,':'
F4F4 CDE5F0            CALL    PO
F4F7 7B                MOV     A,E
F4F8 CD1CF5            CALL    PBYTE
F4FB CD17F5            CALL    PADR
F4FE AF                XRA     A
F4FF CD1CF5            CALL    PBYTE
F502 7E        WRI3:   MOV     A,M
F503 23                INX     H
F504 CD1CF5            CALL    PBYTE
F507 1D                DCR     E
F508 C202F5            JNZ     WRI3
F50B AF                XRA     A
F50C 92                SUB     D
F50D CD1CF5            CALL    PBYTE
F510 D1                POP     D
F511 7D                MOV     A,L
F512 B4                ORA     H
F513 C8                RZ
F514 C3E0F4            JMP     WRI0
F517 7C        PADR:   MOV     A,H
F518 CD1CF5            CALL    PBYTE
F51B 7D                MOV     A,L
F51C F5        PBYTE:  PUSH    PSW
F51D 0F                RRC
F51E 0F                RRC
F51F 0F                RRC
F520 0F                RRC
F521 CD3CF1            CALL    CONV
F524 CDE5F0            CALL    PO
F527 F1                POP     PSW
F528 F5                PUSH    PSW
F529 CD3CF1            CALL    CONV
F52C CDE5F0            CALL    PO
F52F F1                POP     PSW
F530 82                ADD     D
F531 57                MOV     D,A
F532 C9                RET
F533 0E0D      PEOL:   MVI     C,CR
```

```
F535 CDE5F0            CALL    P0
F538 0E0A              MVI     C,LF
F53A C3E5F0            JMP     P0

                ;      REGISTER COMMAND

F53D CD2FF1    X:      CALL    TI
F540 21CCF5            LXI     H,ACTBL
F543 FE0D              CPI     CR
F545 CA9FF5            JZ      X6
F548 47                MOV     B,A
F549 BE        X0:     CMP     M
F54A CA57F5            JZ      X1
F54D 7E                MOV     A,M
F54E 17                RAL
F54F D8                RC
F550 23                INX     H
F551 23                INX     H
F552 23                INX     H
F553 78                MOV     A,B
F554 C349F5            JMP     X0
F557 CDCFF0    X1:     CALL    BLK
F55A 23        X2:     INX     H
F55B 7E                MOV     A,M
F55C EB                XCHG
F55D 6F                MOV     L,A
F55E 2600              MVI     H,0
F560 39                DAD     SP
F561 EB                XCHG
F562 23                INX     H
F563 46                MOV     B,M
F564 23                INX     H
F565 1A                LDAX    D
F566 CD95F1            CALL    LBYTE
F569 05                DCR     B
F56A CA72F5            JZ      X3
F56D 1B                DCX     D
F56E 1A                LDAX    D
F56F CD95F1            CALL    LBYTE
F572 04        X3:     INR     B
F573 0E2D              MVI     C,'-'
F575 CDD1F0            CALL    CO
F578 CDE2F1            CALL    PCHK
F57B 3F                CMC
F57C D0                RNC
F57D CA95F5            JZ      X5
F580 E5                PUSH    H
F581 C5                PUSH    B
F582 CD7EF1            CALL    EXF
F585 E1                POP     H
F586 F1                POP     PSW
F587 C5                PUSH    B
F588 F5                PUSH    PSW
```

```
F589 7D              MOV     A,L
F58A 12              STAX    D
F58B C1              POP     B
F58C 05              DCR     B
F58D CA93F5          JZ      X4
F590 13              INX     D
F591 7C              MOV     A,H
F592 12              STAX    D
F593 C1      X4:     POP     B
F594 E1              POP     H
F595 7E      X5:     MOV     A,M
F596 B7              ORA     A
F597 F8              RM
F598 78              MOV     A,B
F599 FE0D            CPI     CR
F59B C8              RZ
F59C C35AF5          JMP     X2
F59F CD46F1  X6:     CALL    CRLF
F5A2 CDCFF0  X7:     CALL    BLK
F5A5 7E              MOV     A,M
F5A6 23              INX     H
F5A7 B7              ORA     A
F5A8 F8              RM
F5A9 4F              MOV     C,A
F5AA CDD1F0          CALL    CO
F5AD 0E3D            MVI     C,'='
F5AF CDD1F0          CALL    CO
F5B2 7E              MOV     A,M
F5B3 23              INX     H
F5B4 EB              XCHG
F5B5 6F              MOV     L,A
F5B6 2600            MVI     H,0
F5B8 39              DAD     SP
F5B9 EB              XCHG
F5BA 46              MOV     B,M
F5BB 23              INX     H
F5BC 1A              LDAX    D
F5BD CD95F1          CALL    LBYTE
F5C0 05              DCR     B
F5C1 CAA2F5          JZ      X7
F5C4 1B              DCX     D
F5C5 1A              LDAX    D
F5C6 CD95F1          CALL    LBYTE
F5C9 C3A2F5          JMP     X7
F5CC 410701  ACTBL:  DB      'A',    ALOC+2, 1
F5CF 420501          DB      'B',    BLOC+2, 1
F5D2 430401          DB      'C',    CLOC+2, 1
F5D5 440301          DB      'D',    DLOC+2, 1
F5D8 450201          DB      'E',    ELOC+2, 1
F5DB 460601          DB      'F',    FLOC+2, 1
F5DE 481201          DB      'H',    HLOC+2, 1
```

```
F5E1 4C1101             DB      'L',    LLOC+2, 1
F5E4 4D1202             DB      'M',    HLOC+2, 2
F5E7 501502             DB      'P',    PLOC+2, 2
F5EA 530902             DB      'S',    SLOC+2, 2
F5ED FF                 DB      -1

0000                    END
```

```
                ;       SYSTEM CONFIGURATION PACKAGE


F600                    ORG     0F600H

                ;       LOGICAL DEVICE/DEVICE DRIVER TABLES
                ;
                ;       EACH 4 ENTRY TABLE LISTS THE ADDRESSES
                ;       OF THE DRIVER ROUTINES TO BE USED FOR
                ;       THE PHYSICAL DEVICES WHICH MAY ASSIGNED
                ;       TO THAT LOGICAL DEVICE.

                IOTAB:


                ;       CONSOLE STATUS

                ;       RETURN WITH REGISTER A = 0 IF NO
                ;       CONSOLE CHARACTER AVAILABLE.

F600 A0F6       CSTAB:  DW      TTST    ;0
F602 7FF6               DW      KYST    ;1
F604 7FF6               DW      KYST    ;2
F606 7FF6               DW      KYST    ;3

                ;       CONSOLE INPUT

                ;       RETURN CONSOLE INPUT CHARACTER
                ;       IN REGISTER A.

F608 A8F6       CITAB:  DW      TTI     ;0
F60A 66F6               DW      KYBD    ;1
F60C 66F6               DW      KYBD    ;2
F60E 66F6               DW      KYBD    ;3

                ;       CONSOLE OUTPUT

                ;       OUTPUT BYTE IN REGISTER C
                ;       TO CONSOLE OUTPUT DEVICE.

F610 B7F6       COTAB:  DW      TTO     ;0
F612 B7F6               DW      TTO     ;1
F614 D4F6               DW      THRM    ;2
F616 59F6               DW      CRT     ;3

                ;       READER INPUT

                ;       RETURN READER INPUT BYTE IN
                ;       REGISTER A, CARRY OFF.  SET
                ;       CARRY IF NO BYTE AVAILABLE.
```

```
F618 C2F6      RITAB:  DW      TTR     ;0
F61A 87F6              DW      RDR     ;1
F61C 66F6              DW      KYBD    ;2
F61E F0B8              DW      0B8F0H  ;3 DISK READ

               ;       PUNCH OUTPUT

               ;       OUTPUT BYTE IN REGISTER C
               ;       TO PUNCH DEVICE.

F620 B7F6      POTAB:  DW      TTO     ;0
F622 DFF6              DW      PUNCH   ;1
F624 59F6              DW      CRT     ;2
F626 73B9              DW      0B973H  ;3 DISK WRITE

               ;       LISTING OUTPUT

               ;       OUTPUT BYTE IN REGISTER C
               ;       TO LISTING DEVICE.

F628 B7F6      LOTAB:  DW      TTO     ;0
F62A 59F6              DW      CRT     ;1
F62C D4F6              DW      THRM    ;2
F62E B7F6              DW      TTO     ;3



               ;       SPECIAL SUBROUTINE TO LOCATE MONITOR
               ;       SCRATCH RAM
               ;
               ;       THE ADDRESS OF THE TOP OF THE SCRATCH
               ;       RAM AREA USED BY THE MONITOR IS RETURNED
               ;       IN REGISTERS D,E.
               ;NOTE:  THIS SUBROUTINE IS NOT CALLED IN THE
               ;       USUAL WAY:  INSTEAD, THE RETURN ADDRESS
               ;       IS PLACED IN REGISTERS D,E AND THE
               ;       SUBROUTINE IS ENTERED BY A JUMP INSTRUCTION.
               ;       RETURN IS DONE BY PLACING THE RETURN
               ;       ADDRESS IN H,L AND EXECUTING A PCHL INST.

F630 C33DF6    ADSCS:  JMP     ADS2


               ;       SUBROUTINE TO LOCATE MONITOR SCRATCH
               ;       RAM
               ;
               ;       THE ADDRESS OF THE TOP OF THE 64 BYTES
               ;       OF RAM TO BE USED BY THE MONITOR FOR
               ;       SCRATCHPAD IS RETURNED IN REGISTERS
               ;       H,L.
```

```
F633 C34FF6    ADSCR:  JMP     ADS1

                ;       SUBROUTINE TO SET ADDRESS
                ;       OF IOBYT
                ;
                ;       THE ADDRESS OF THE BYTE USED TO
                ;       RECORD THE CURRENT PHYSICAL/LOGICAL
                ;       DEVICE ASSIGNMENTS IS RETURNED IN
                ;       REGISTERS H,L.

F636 C34FF6    ADIOB:  JMP     ADS1

                ;       SUBROUTINE TO SET THE USER STACK
                ;       ADDRESS.
                ;
                ;       THE ADDRESS TO BE USED AS THE
                ;       DEFAULT VALUE OF THE USER STACK
                ;       ADDRESS IS RETURNED IN REGISTERS H,L.

F639 210001    ADUST:  LXI     H,0100H
F63C C9                RET

F63D 210000    ADS2:   LXI     H,0
F640 24        ADS3:   INR     H
F641 7E                MOV     A,M
F642 2F                CMA
F643 F3                DI
F644 77                MOV     M,A
F645 BE                CMP     M
F646 2F                CMA
F647 FB                EI
F648 77                MOV     M,A
F649 CA40F6            JZ      ADS3
F64C 2B                DCX     H
F64D EB                XCHG
F64E E9                PCHL

F64F D5        ADS1:   PUSH    D
F650 1156F6            LXI     D,$+6
F653 C33DF6            JMP     ADS2
F656 EB                XCHG
F657 D1                POP     D
F658 C9                RET

                ;       PHYSICAL DEVICE DRIVER ROUTINES

                ;       REQUIREMENTS
                ;               MAINTAIN CONTENTS OF ALL
                ;               REGISTERS EXCEPT A AND F.
                ;               EXIT BY RETURN INST.
```

```
                ;       VIDEO DRIVER

F659 79         CRT:    MOV     A,C
F65A B7                 ORA     A       ;CHECK FOR NULL
F65B C8                 RZ
F65C E5                 PUSH    H
F65D CD36F6             CALL    ADIOB
F660 2B                 DCX     H
F661 2B                 DCX     H
F662 2B                 DCX     H
F663 C304F7             JMP     0F704H

                ;       KEYBOARD DRIVER

F666 DB02       KYBD:   IN      2
F668 E601               ANI     1
F66A C266F6             JNZ     KYBD
F66D DB03               IN      3
F66F E67F               ANI     7FH
F671 FE61               CPI     61H     ;LOWER CASE  A
F673 DA7DF6             JC      KB1
F676 FE7B               CPI     7AH+1   ;LOWER CASE Z +1
F678 D27DF6             JNC     KB1
F67B E6DF               ANI     0DFH    ;DELET ONE BIT
F67D B7         KB1:    ORA     A
F67E C9                 RET

                ;       KEYBOARD STATUS DRIVER

F67F DB02       KYST:   IN      2
F681 E601               ANI     1
F683 D601               SUI     1
F685 9F                 SBB     A
F686 C9                 RET

                ;       READER DRIVER

F687 E5         RDR:    PUSH    H
F688 210000             LXI     H,0
F68B DB04       RD:     IN      4
F68D E601               ANI     1
F68F CA9BF6             JZ      RD2
F692 2B                 DCX     H
F693 7C                 MOV     A,H
F694 B5                 ORA     L
F695 C28BF6             JNZ     RD
F698 37                 STC
F699 E1                 POP     H
F69A C9                 RET
```

```
F69B DB05      RD2:    IN      5
F69D B7                ORA     A
F69E E1                POP     H
F69F C9                RET

               ;       TELETYPE STATUS DRIVER

F6A0 DB00      TTST:   IN      0
F6A2 E601              ANI     1
F6A4 D601              SUI     1
F6A6 9F                SBB     A
F6A7 C9                RET

               ;       TELETYPE INPUT DRIVER

F6A8 AF        TTI:    XRA     A
F6A9 D300              OUT     0
F6AB DB00      TTI1:   IN      0
F6AD E601              ANI     1
F6AF C2ABF6            JNZ     TTI1
F6B2 DB01              IN      1
F6B4 E67F              ANI     7FH
F6B6 C9                RET

               ;       TELETYPE OUTPUT DRIVER

F6B7 DB00      TTO:    IN      0
F6B9 E680              ANI     80H
F6BB C2B7F6            JNZ     TTO
F6BE 79                MOV     A,C
F6BF D301              OUT     1
F6C1 C9                RET

               ;       TELETYPE READER DRIVER

F6C2 3E01      TTR:    MVI     A,1
F6C4 D300              OUT     0
F6C6 3E00              MVI     A,0
F6C8 D300              OUT     0
F6CA DB00      TTR1:   IN      0
F6CC E601              ANI     1
F6CE C2CAF6            JNZ     TTR1
F6D1 DB01              IN      1
F6D3 C9                RET
```

```
                ;       THERMAL PRINTER DRIVER

F6D4 DB02       THRM:   IN      2
F6D6 E680               ANI     80H
F6D8 C2D4F6             JNZ     THRM
F6DB 79                 MOV     A,C
F6DC D303               OUT     3
F6DE C9                 RET

                ;       PUNCH DRIVER

F6DF DB04       PUNCH:  IN      4
F6E1 E680               ANI     80H
F6E3 C2DFF6             JNZ     PUNCH
F6E6 79                 MOV     A,C
F6E7 D305               OUT     5
F6E9 C9                 RET

0000                    END
```

```
                ;       VIDEO BOARD  DRIVER


                ;       THIS SUBROUTINE FACILITATES THE USE
                ;       OF THE SOLID STATE MUSIC VBI BOARD
                ;       AND A VIDEO DISPLAY DEVICE AS A
                ;       CONSOLE OUTPUT DEVICE.
                ;       ASCII CHARACTERS PRESENTED TO THE
                ;       SUBROUTINE IN THE C REGISTER ARE
                ;       DISPLAYED ON THE SCREEN.  CERTAIN
                ;       CHARACTERS, LISTED BELOW, RECEIVE
                ;       SPECIAL TREATMENT.  ALL REGISTERS
                ;       ARE PRESERVED BY THIS SUBROUTINE.

                ;       LOC IS THE BEGINNING ADDRESS OF THE
                ;       SUBROUTINE.  IT MAY BE IN RAM OR ROM.

F700            LOC     EQU     0F700H

                ;       VID IS THE BEGINNING ADDRESS ASSIGNED
                ;       TO THE DISPLAY RAM LOCATED ON THE VBI
                ;       BOARD.

EC00            VID     EQU     0EC00H  or B0006

                ;       THREE BYTES OF RAM ARE REQUIRED FOR
                ;       HOUSEKEEPING.  THESE BYTES MUST BE
                ;       IN AN AREA UNUSED BY OTHER PROGRAMS.

BC44            VDPTR   EQU     0BC44H  ;CURSOR POINTER
BC46            VDHLD   EQU     VDPTR+2 ;CHARACTER HOLD


                ;       NON-DISPLAYABLE CHARACTERS

000C            FF      EQU     0CH     ;FORM FEED
                                        ;CLEAR SCREEN, HOME CURSOR
000A            LF      EQU     0AH     ;LINE FEED
                                        ;DOWN ONE LINE, CLEAR LINE
000D            CR      EQU     0DH     ;CARRIAGE RETURN
                                        ;MOVE CURSOR TO LEFT MARGIN


                ;       NORMAL ENTRY POINT

F700                    ORG     LOC

F700 E5         VDTTY:  PUSH    H       ;SAVE HL
F701 2144BC             LXI     H,VDPTR ;ADDR OF CURSOR POINTER

                ;       ALTERNATE ENTRY POINT
```

```
                ;       THIS ENTRY POINT  MAY BE USED IF
                ;       THE CURSOR POINTER AND CHARACTER
                ;       HOLD ARE AT LOCATIONS OTHER THAN
                ;       THOSE SPECIFIED ON THIS LISTING.
                ;       THE USER MUST SUPPLY SUBROUTINE
                ;       ENTRY CODE AS FOLLOWS:
                ;ENTR:  PUS H    H       ;SAVE HL
                ;       LXI     H,PNTR  ;ADDR OF CURSOR POINTER
                ;       JMP     ALTVD   ;JOIN THIS CODE

F704 D5         ALTVD:  PUSH    D       ;SAVE DE
F705 C5                 PUSH    B       ;SAVE BC
F706 5E                 MOV     E,M     ;LPTR
F707 23                 INX     H       ;
F708 7E                 MOV     A,M     ;HPTR
F709 E603               ANI     3       ;CONVERT TO VIDEO
F70B C6EC               ADI     VID SHR 8 ;RAM ADDRESS
F70D 57                 MOV     D,A     ;
F70E 23                 INX     H       ;
F70F 46                 MOV     B,M     ;CHAR UNDER CURSOR
F710 EB                 XCHG            ;PNTR TO HL
F711 70                 MOV     M,B     ;RESTORE PREV CHAR

                ;       IDENTIFY INPUT CHAR

F712 79                 MOV     A,C     ;NEW CHAR
F713 FE0C               CPI     FF      ;
F715 CA4BF7             JZ      VIDFF   ;FORM FEED
F718 FE0D               CPI     CR      ;
F71A CA59F7             JZ      VIDCR   ;CARRIAGE RETURN
F71D FE0A               CPI     LF      ;
F71F CA60F7             JZ      VIDLF   ;LINE FEED

F722 71                 MOV     M,C     ;
F723 010100     CRRT:   LXI     B,1

                ;       ADJUST CURSOR POINTER

F726 09         CRADJ:  DAD     B       ;

                ;       CHECK FOR OVERFLOW

F727 7C                 MOV     A,H     ;
F728 FEF0               CPI     (VID+1024) SHR 8 ;
F72A C236F7             JNZ     VIDRT   ;
F72D 21C0EF             LXI     H,VID+960 ;
F730 CD7FF7             CALL    ROLL0   ;
F733 C33CF7             JMP     VIDR1   ;

                ;       COMMON EXIT CODE
                ;       NORMALIZE CURSOR POINTER
```

```
F736 26EF      VIDRT:  MVI     H,(VID+960) SHR 8 ;
F738 7D                MOV     A,L       ;
F739 F6C0              ORI     0C0H      ;
F73B 6F                MOV     L,A       ;
F73C 7E        VIDR1:  MOV     A,M       ;CHAR UNDER CURSOR
F73D 367F              MVI     M,7FH     ;CURSOR
F73F EB                XCHG              ;PNTR TO DE
F740 77                MOV     M,A       ;CHAR UNDER CURSOR
F741 2B                DCX     H         ;
F742 72                MOV     M,D       ;HPTR
F743 2B                DCX     H         ;
F744 73                MOV     M,E       ;LPTR

                ;      RESTORE REGISTERS, EXIT

F745 C1                POP     B         ;
F746 D1                POP     D         ;
F747 E1                POP     H         ;
F748 79                MOV     A,C       ;
F749 B7                ORA     A
F74A C9                RET               ;

                ;      PROCESS FORM FEED
                ;      FILL SCREEN WITH SPACES ,
                ;      MOVE CURSOR TO TOP LEFT

F74B 2100EC    VIDFF:  LXI     H,VID     ;
F74E E5                PUSH    H         ;
F74F 3620      VIDFC:  MVI     M,' '     ;
F751 23                INX     H         ;
F752 7C                MOV     A,H       ;
F753 FEF0              CPI     (VID+1024) SHR 8 ;
F755 DA4FF7            JC      VIDFC     ;
F758 E1                POP     H         ;

                ;      PROCESS CARRIAGE RETURN
                ;      MOVE CURSOR TO BEGINNING
                ;      OF LINE

F759 7D        VIDCR:  MOV     A,L       ;
F75A E6C0              ANI     0C0H      ;
F75C 6F                MOV     L,A       ;
F75D C336F7            JMP     VIDRT     ;

                ;      PROCESS LINE FEED
                ;      MOVE CURSOR DOWN ONE LINE,
                ;      FILL NEW LINE WITH SPACES

F760 D5        VIDLF:  PUSH    D         ;
F761 114000            LXI     D,64      ;
F764 19                DAD     D         ;
F765 7C                MOV     A,H       ;
```

```
F766 FEF0              CPI     (VID + 1024) SHR 8  ;
F768 D1                POP     D           ;
F769 C236F7            JNZ     VIDRT       ;


                ;      THE FOLLOWING INSTRUCTION
                ;      (MARKED XXXX) MAY BE REMOVED
                ;      IF SENSE SWITCHES ARE NOT
                ;      TO BE USED.

                ;      WAIT UNTIL SENSE SWITCH 1 IS ON
                ;      BEFORE ROLLING UP ONE LINE.

F76C DBFF       VDLF2: IN      0FFH        ;XXXX
F76E E601              ANI     1           ;XXXX
F770 CA6CF7            JZ      VDLF2       ;XXXX

                ;      ROLL THE WHOLE DISPLAY UP ONE
                ;      LINE.

F773 CD7FF7            CALL    ROLL0       ;
F776 7D                MOV     A,L         ;
F777 F6C0              ORI     0C0H        ;
F779 6F                MOV     L,A         ;
F77A 26EF              MVI     H,(VID+960) SHR 8 ;
F77C C336F7            JMP     VIDRT       ;
                ;
                ;      ROLL SUBROUTINE

F77F D5         ROLL0: PUSH    D           ;
F780 E5                PUSH    H           ;
F781 1100EC            LXI     D,VID       ;
F784 2140EC            LXI     H,VID+64    ;
F787 7E         ROLL1: MOV     A,M         ;
F788 12                STAX    D
F789 3620              MVI     M,20H       ;
F78B 13                INX     D           ;
F78C 23                INX     H           ;
F78D 7C                MOV     A,H         ;
F78E FEF0              CPI     (VID+1024) SHR 8 ;
F790 C287F7            JNZ     ROLL1       ;
F793 E1                POP     H           ;



F794 D1         ROLL2: POP     D           ;
F795 C9                RET                 ;

                ;      GRAPHICS INTERFACE SUBROUTINES

                ;      THESE SUBROUTINES FACILITATE THE
                ;      USE OF THE SOLID STATE MUSIC VB1
```

```
                ;       BOARD AND A VIDEO DISPLAY DEVICE
                ;       AS A GRAPHICS DISPLAY DEVICE.

                ;       THESE SUBROUTINES TREAT THE DISPLAY
                ;       SCREEN AS A MATRIX OF DOTS, 48 DOTS
                ;       HIGH BY 128 DOTS WIDE.  EACH DOT IS
                ;       SPECIFIED IN TERMS OF ITS VERTICAL
                ;       COORDINATE(0-47) AND ITS HORIZONTAL
                ;       COORDINATE(0-127).  DOT 0,0 IS AT
                ;       THE LOWER LEFT CORNER OF THE SCREEN.

                ;       THE SUBROUTINES HAVE SIMILIAR
                ;       INTERFACES WITH THEIR CALLING
                ;       PROGRAMS.  REGISTER B IS PRESERVED.
                ;
                ;       ENTRY CONDITIONS:
                ;               H = VERTICAL COORDINATE
                ;               L = HORIZONTAL COORDINATE
                ;       EXIT CONDITIONS
                ;               A = DIFFERS BY SUBROUTINE
                ;               B = PRESERVED
                ;               C = BIT MASK FOR SPECIFIED DOT
                ;               DE= MEMORY ADDRESS OF DOT
                ;               H = VERTICAL COORDINATE
                ;               L = HORIZONTAL COORDINATE
                ;       H AND L ARE CONVERTED(IF NECESSARY)
                ;       MODULO 48 AND 128 RESPECTIVELY.


                ;       THE CHECK SUBROUTINE SETS THE ZERO
                ;       FLAG TO INDICATE WHETHER THE SPECIFIED
                ;       DOT IS WHITE OR BLACK.  IF THE DOT
                ;       IS CURRENTLY WHITE THE ZERO FLAG IS
                ;       SET ON, IF THE DOT IS BLACK THE FLAG
                ;       IS SET OFF.  THE A REGISTER CONTAINS
                ;       ZERO IF THE DOT IS WHITE, THE BIT
                ;       MASK IF IT IS BLACK.

F796 CDB0F7     CHECK:  CALL    CNVRT   ;
F799 A1                 ANA     C       ;
F79A C9                 RET             ;

                ;       THE WHITE SUBROUTINE SETS THE
                ;       SPECIFIE D DOT WHITE.  REGISTER
                ;       A CONTAINS THE NEW CONTENTS OF
                ;       THE MEMORY LOCATION.

F79B CDB0F7     WHITE:  CALL    CNVRT   ;CONVERT
F79E E6BF               ANI     0BFH    ;CLEAR UNUSED BIT
F7A0 F680               ORI     80H     ;SET GRAPHICS BIT
F7A2 B1                 ORA     C       ;SET THIS DOT
F7A3 A9                 XRA     C       ;CLEAR THIS DOT
F7A4 12                 STAX    D       ;UPDATE BYTE
```

```
F7A5 C9                 RET             ;

                ;       THE BLACK SUBROUTINE SETS THE
                ;       SPECIFIED DOT BLACK.  REGISTER
                ;       A CONTAINS THE NEW CONTENTS OF
                ;       THE MEMORY LOCATION.

F7A6 CDB0F7     BLACK:  CALL    CNVRT   ;CONVERT
F7A9 E6BF               ANI     0BFH    ;CLEAR UNUSED BIT
F7AB F680               ORI     80H     ;SET GRAPHICS BIT
F7AD B1                 ORA     C       ;SET THIS DOT
F7AE 12                 STAX    D       ;UPDATE BYTE
F7AF C9                 RET             ;

                ;       THE CNVRT SUBROUTINE PERFORMS
                ;       THE COORDINATE TO ADDRESS -
                ;       BIT MASK CONVERSION.  REGISTER
                ;       A CONTAINS THE CURRENT CONTENTS
                ;       OF THE MEMORY LOCATION.

F7B0 C5         CNVRT:  PUSH    B       ;

                ;       NORMALIZE THE COORDINATES

F7B1 7D                 MOV     A,L     ;
F7B2 E67F               ANI     7FH     ;
F7B4 6F                 MOV     L,A     ;
F7B5 7C                 MOV     A,H     ;
F7B6 D630       D1:     SUI     48      ;
F7B8 F2B6F7             JP      D1      ;
F7BB C630       D2:     ADI     48      ;
F7BD FABBF7             JM      D2      ;
F7C0 67                 MOV     H,A     ;
F7C1 E5                 PUSH    H       ;

                ;       CONVERT COORDINATES TO ADDRESS
                ;       IN DE

F7C2 44                 MOV     B,H     ;
F7C3 4D                 MOV     C,L     ;
F7C4 5C                 MOV     E,H     ;
F7C5 1600               MVI     D,0     ;
F7C7 210100             LXI     H,1     ;
F7CA 19                 DAD     D       ;
F7CB 29                 DAD     H       ;
F7CC 29                 DAD     H       ;
F7CD 19                 DAD     D       ;
F7CE 29                 DAD     H       ;
F7CF 29                 DAD     H       ;
F7D0 19                 DAD     D       ;
F7D1 54                 MOV     D,H     ;
F7D2 7D                 MOV     A,L     ;
F7D3 E6C0               ANI     0C0H    ;
```

```
F7D5 5F                 MOV     E,A     ;
F7D6 19                 DAD     D       ;
F7D7 19                 DAD     D       ;
F7D8 29                 DAD     H       ;
F7D9 29                 DAD     H       ;
F7DA 78                 MOV     A,B     ;
F7DB 94                 SUB     H       ;
F7DC 47                 MOV     B,A     ;
F7DD 3EC0               MVI     A,(VID+960) AND 0FFH
F7DF 93                 SUB     E       ;
F7E0 5F                 MOV     E,A     ;
F7E1 3EEF               MVI     A,(VID+960) SHR 8
F7E3 9A                 SBB     D       ;
F7E4 57                 MOV     D,A     ;
F7E5 79                 MOV     A,C     ;
F7E6 1F                 RAR             ;
F7E7 B3                 ORA     E       ;
F7E8 5F                 MOV     E,A     ;

                ;       GENERATE BIT MASK

F7E9 79                 MOV     A,C     ;
F7EA 1F                 RAR             ;
F7EB 78                 MOV     A,B     ;
F7EC 17                 RAL             ;
F7ED 4F                 MOV     C,A     ;
F7EE 0600               MVI     B,0     ;
F7F0 21FAF7             LXI     H,DTAB  ;
F7F3 09                 DAD     B       ;
F7F4 7E                 MOV     A,M     ;

                ;       PREPARE FOR EXIT

F7F5 E1                 POP     H       ;
F7F6 C1                 POP     B       ;
F7F7 4F                 MOV     C,A     ;
F7F8 1A                 LDAX    D       ;
F7F9 C9                 RET             ;

F7FA 04         DTAB:   DB      04H
F7FB 20                 DB      20H
F7FC 02                 DB      02H
F7FD 10                 DB      10H
F7FE 01                 DB      01H
F7FF 08                 DB      08H

0000                    END
```

2.0 DESCRIPTION OF A EXAMPLE MONITOR LISTING
("The care and feeding of the SSM 8080 Monitor")

2.1 Loading

Paper-tape: Load the paper-tape into the computer per the separate instruction sheet. The monitor will reside in memory from FØØØ Hex to F7FF Hex.

Prom IC's: Place the eight 1702 PROMs (or two 2708's) into the ROM card. Note that the high address is marked on each ROM as FØ, F1, etc. Set the high address of the ROM card, so the set of PROMs will address appropriately from FØØØ Hex and up.
Card kits are available from SSM to support both the 1702 & 2708 PROMs.

MB-3 ROM Board: Mount the 1702's in order into the top row of sockets on the board. FØ is the first ROM to the left.

MB-8 ROM Board: Mount the 2708's in order into the fifth and sixth socket from the left in the bottom row of sockets. FØ is the fifth socket.

2.2 Execution

TTY Version (Suffix "T"): To initialize and run the *SSM 8Ø8Ø Monitor, after loading it into memory, just examine address FØØØ and hit the run switch. The Monitor will output to port 1 (using port Ø as status) the statement MONITOR V1.Ø and then wait for a keyboard command on port 1.

Video Version (Suffix "V"): To initialize and run the *SSM 8Ø8Ø Monitor, after loading it into memory, just examine address FØØØ, set sense switch A8 to a one, and hit the run switch. (Do not worry, if you don't have a front panel or a sense switch at port FF, the Monitor doesn't need the sense switch to work. The sense switch is only an additional feature in the Video software driver allowing the user to stop or start a long listing on the video screen. If you don't have a sense switch port at FF, then A8 will be a one.) The Monitor will output to ECØØ (1702 PROM Monitor) or BØØØ (2708 PROM Monitor) the statement MONITOR V1.Ø and then wait for a keyboard command on port 3 (status port 2, $\overline{DAV}$ = DØ).

2.2 (continue)

The 2708 Monitor has its video driver changed for a VB-1B card at BØØØ, instead of ECØØ to allow a 16K Rom card to be used in the computer with no address conflicts. In the Source Listing, starting at address F7ØØ, the label VID is equated to BØØØ so some bytes will differ from the listing.

*Note: You must have at least 256 bytes of RAM or more starting at ØØØØ. The SSM 8Ø8Ø Monitor creates a scratch pad table (64 bytes) automatically at the top of contiguous memory when intialized.

Address	2708's byte	1702's byte
F7ØC	BØ	EC
F729	B4	FØ
F72E	CØ	CØ
F72F	B3	EF
F737	B3	EF
F74C	ØØ	ØØ
F74D	BØ	EC
F754	B4	FØ
F767	B4	FØ
F77B	B3	EF
F782	ØØ	ØØ
F783	BØ	EC
F786	BØ	EC
F78F	B4	FØ
F7E2	B3	EF

To make the Monitor power-up using the Video card, instead of the Teletype for an output device, six bytes were altered in the I/O tables at F6ØØ.

The AC commands have been changed only in the Video version of the Monitor, so AC=Ø is the parallel keyboard and CRT and AC=3 is the teletype.

Address	Video Version	TTY Version
F6ØØ	7F	AØ
F6Ø6	AØ	7F
F6Ø8	66	A8
F6ØE	A8	66
F61Ø	59	B7
F616	B7	59

2.3 Example Operation of Monitor (see Example Monitor Listing)

[1]... The monitor outputs its name and types a period, when executed at FØØØ Hex.

[2]... If an asterisk was outputted to the printing device, then a character from the input device was not of the correct notation expected by the monitor.

[3]... The substitute command is now used at address zero (SØ) and the space-bar is depressed. The computer responds with "27-", or any random Hex number that happens to be at address ØØØØ. The user typed "C3" to change this byte at address ØØØØ. If the space-bar is hit with no other character typed, then consecutive bytes in memory are displayed without being modified.

[4]... A display of memory is made from ØØØØ to ØØ1F Hex. After typing in DØ 1F a carriage-return is typed. Note: The starting address of each row up to 16 bytes is displayed on the left, as four hex digits.

[5]... A fill command is used. Zero's are placed into memory from ØØ1Ø H to ØØ1F Hex.

[6]... The memory fill command is checked with a display command.

[7]... A small program was now written into memory using the substitute command.
Program:

Address	Code	Mnevmonics
ØØ4Ø	3E,ØØ,	MVI A,Ø
ØØ42	Ø1,Ø2,Ø1	LXI B,Ø1Ø2H
ØØ45	11,Ø4,Ø3	LXI D,Ø3Ø4H
ØØ48	21,Ø6,Ø5	LXI H,Ø5Ø6H
ØØ4B	31,Ø8,Ø7	LXI sp,Ø7Ø8H
ØØ4E	C3,4Ø,ØØ	JMP 4ØH

[8]... A display of the small program.

[9]... A goto command is used to execute the program at ØØ4Ø and also one of the two optional software breakpoints is set to ØØ4Ø (see notes in Source Listing). Note: The breakpoints will not work on a program in ROM. RST 1 is used for software breakpoints.

[10]... The program is stopped at *ØØ4Ø, then the user typed "X" and a carriage-return to display the 8080's registers A thru E,H,L and the flag code "F" also the memory location (M), program counter (P) and the stack pointer (S).

2.3 Use of the monitor (continued)

[11] ... To continue the program, a "G" was typed followed by a space and another optional breakpoint address was set at ØØ48.

[12] ... At step-[3], a jump instruction to the monitor was written into memory. Using the goto command, the monitor was re-in-initialized.

[13] ... A block of memory data will now be changed by using the move command (M). Memory data from ØØ3Ø to ØØ3F will be placed at ØØ2Ø and up.

[14] ... A dump is made of the small program from ØØ4Ø to ØØ5Ø in a format compatiable with INTEL's paper-tape format.

[15] ... Using the "H" command the sum and difference of two hex numbers, all the way up to FFFF, can be displayed.

[16] ... The user now changed the I/O configuration of the monitor.

```
AC = 1 Console I = parallel keyboard in.
       Console 0 = teletype for output.

AR = 1 Reader    = Special reader device.
AP = 2 PUNCH     = CRT for a punch.
AL = 1 LIST DEVICE = CRT for listing.
```

3.0 I/O CONFIGURATION

The Input-Output configuration is established in tables and routines between F6ØØ to F6FF. There are 6 tables starting at F6ØØ which are for storing the main entry addresses (2 bytes each) for the routines to Check-status (CSTAB), Input data (CITAB), Output data (COTAB), Reader device (RITAB), Punch device (POTAB), and Listing device (LOTAB). Typing the command AC,AR,AP or AL followed by an equals sign and a number will set up an I/O configuration. The I/O configuration is coded into just one byte (IOBYT) at the very last byte of contiguous RAM. (Top of memory)

The "IOBYT" is broken down into four parts, where every two bits represents the binary number zero through three of the selected I/O configuration.

3.0 I/O CONFIGURATION (continued)

IOBYT:

List Device	Punch	Reader	Console
D7			DØ

If the Monitor has been initialized and an I/O configuration has been set-up and you wish to go to the monitor without resetting the I/O, then jump to FØ21.

3.1 Preset I/O

The SSM 8080 Monitor comes with the following preset I/O configurations:

Logical Command	Status Port	*Status Bits	Data Port	Comments
AC=Ø	Ø	$\overline{DAV}$=DØ $\overline{DAK}$=D7	1	Main console I/O.
AC=1	2 Ø	$\overline{DAV}$=DØ $\overline{DAK}$=D7	3 1	Keyboard input Console output
AC=2	2 2	$\overline{DAV}$=DØ $\overline{DAK}$=D7	3 3	Keyboard input Output device
AC=3	2 memory mapped	$\overline{DAV}$=DØ I/O	3	Keyboard input Output to VB-1
AR=Ø	Ø	$\overline{DAV}$=DØ	1	Console input
AR=1	4	$\overline{DAV}$=DØ	5	Reader
AR=2	2	$\overline{DAV}$=DØ	3	Keyboard
AR=3	Vector to read from disk			**ICOM Microfloppy
AP=Ø	Ø	$\overline{DAK}$=D7	1	Console output
AP=1	4	$\overline{DAK}$=D7	5	Output to punch
AP=2	Memory mapped I/O			Output to VB-1
AP=3	Vector to write on disk			ICOM Microfloppy
AL=Ø	Ø	$\overline{DAK}$=D7	1	Console output
AL=1	Memory mapped I/O			Output to VB-1
AL=2	2	$\overline{DAK}$=D7	3	Listing Device
AL=3	Ø	$\overline{DAK}$=D7	1	Console output

*Data-Available=DAV, Data-Acknowledge=DAK

**Trade name of a Floppy-Disk system form ICOM Microperipherals.