CHAPTER 8 ROM CARTRIDGE

8.1 General

The ROM cartridge, which is provided as a plug-in option of the HX-20, can read 2K to 16K bytes of data from an external ROM memory via the I/O ports using its addressing counter and shift register. The addressing counter is incremental and its value can also be reset to 0.
The ROM cartridge is designed for an output-only file as a ROM file to allow data output in this file format.

8.2 Configuration

Table 8.1 shows the I/O ports related to the ROM cartridge.

	Port	I/O	Description
Master MCU	P17	Input	ROM data (1 bit)
	P266	Output	Shift/load select (0: Load; 1: Shift)
	P267	Output	Clock
Slave MCU	P20	Input	ROM cartridge interface judgment
	P46	Input	ROM cartridge interface judgment
	P42	Output	Shift register clear (0: OFF (Clear) 1: ON (Don't clear)
	P43	Output	Power supply (0: OFF 1: ON)
	P44	Output	Addressing counter clear (0: OFF (Clear); 1: ON (Don't clear))

The ROM cartridge is configured as shown in Fig. 8-1. One byte of ROM data at the address indicated by the addressing counter is input to the shift register, which in turn transfers the ROM data to the master MCU.

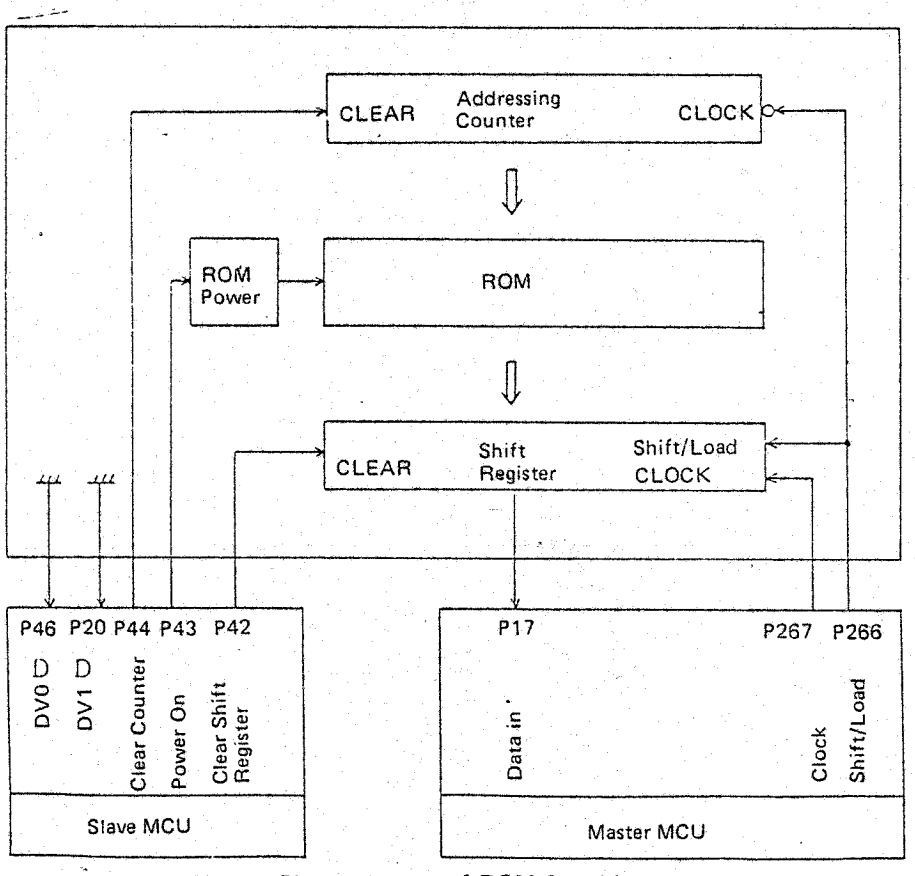

Fig. 8-1 Block Diagram of ROM Cartridge

8.3 Data Input Procedure

Only two types of instructions are applicable to the addressing counter: Clear (by setting the P44 of the slave MCU to "0") and Count-up. Data is fetched by the master MCU from the shift register by inputting one bit of data to the port P17 of the master MCU each time the data bits in the shift register are moved. Data input from the ROM cartridge is performed by the procedure as detailed below.

(1) The power supply of the ROM cartridge is turned ON.
The port P43 of the slave MCU is the power supply port to turn on or off the ROM cartridge. The master MCU instructs the slave MCU to issue a ROM Power ON command to turn on the power supply of the ROM cartridge.

(2) The addressing counter is cleared.
The addressing counter is automatically reset to 0 when the ROM Power ON command is issued to the ROM cartridge from the slave MCU.

(3) The addressing counter is incremented to the address from which data is to be read.
The counter counts up when the voltage level at the port P266 (bit 6 at address 26) of the master MCU changes from High to Low.

(4) When port P266 is at Low level, one byte of data at the address indicated by the addressing counter is loaded into the shift register at the leading edge of a CLOCK signal appearing at the P267 (bit 7 at address 26) of the master MCU. In this case, bit 7 is first loaded into the master MCU through port P17 (Data in).

(5) When port P266 is at High level, the contents of the shift register are shifted by one bit at the trailing edge of the CLOCK signal (P267). By repeating this operation 7 times, one byte of data can be fetched by the master MCU.

(6) If data input from the ROM cartridge is no longer required, the power supply of the ROM cartridge must be turned off by sending a command from the master MCU to the slave MCU to turn off the ROM power supply.

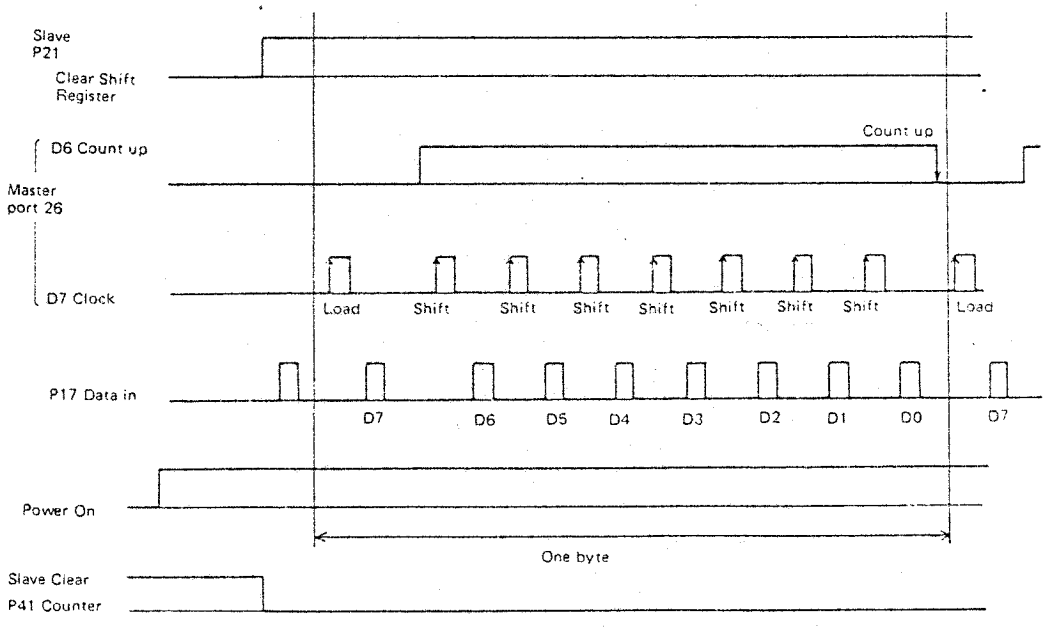

Fig. 8-2 Timing Chart of Data Input from ROM Cartridge

Note: If data is input after clearing the shift register, the data that is input to the master MCU is binary 0. If this Shift Register Clear operation is performed when the optional microcassette drive is connected to the HX-20, binary 1 is input.

8.4 ROM File

Data input from the optional ROM cartridge is supported in the form of data input from a ROM file. The ROM file consists of 32 headers and a data area. Each header may contain a maximum of 32 bytes of data as header information. The ROM file may only be accessed sequentially but not randomly.
Fig. 8-3 shows the structure of the ROM file.

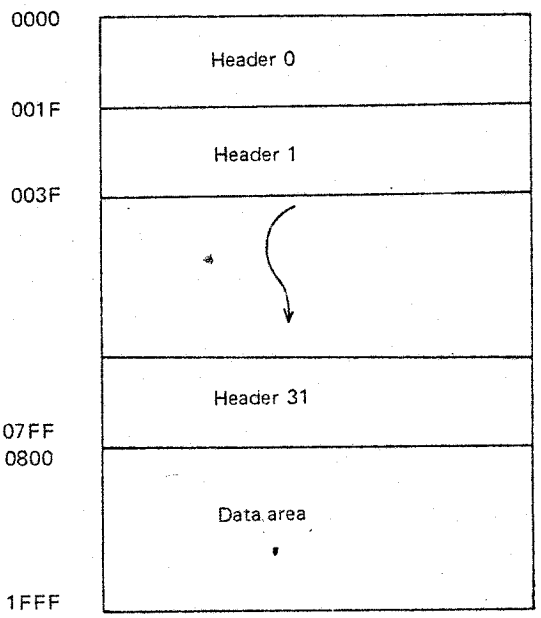

Fig. 8-3 Structure of ROM File

Headers are allocated as fixed areas from address 0000 in units of 32 bytes. Header 0 is from address 0000 to address 001F. A maximum of 32 headers can be set. The first one byte of each header represents the first letter of the filename as well as header information. If the first one byte of a header is "00", it indicates that the file with that header has been deleted. If "FF", it indicates that no subsequent header exists.
If the first one byte of header 2 is "FF", headers 0 and 1 are valid as headers. The contents of the header information are shown in Section 8.7 below.

8.5 Subroutines for ROM Cartridge

The following 4 subroutines are provided for the ROM cartridge:
(1) OPNPRM: Opens the ROM file.
(2) REDPRM: Inputs data from the ROM file in units of one byte.
(3) CLSPRM: Closes the ROM file.
(4) DIRPRM: Inputs the ROM file directory.

8.6 File Input Procedure

A ROM file is processed for data input as follows:

(1) Opening the ROM file
Subroutine "OPNPRM" is used to start the input of data from the ROM file.

(2) Data input
Data is read from the ROM file in units of one byte by subroutine "REDPRM."

(3) Closing the ROM file
Data input from the ROM file is terminated by subroutine "CLSPRM".

Note: Upon opening the ROM file, the ROM cartridge is energized. The ROM file must be closed soon after the data input has been completed particularly when an NMOS type PROM with high power consumption is used.

8.7 Header Format of ROM File

Columns	Bytes	Item	Description
0 to 7	8	Filename	Filename (in ASCII codes.) Column 0 represents ID in addition to the filename. 00: File has been deleted. FF: No subsequent header exists.
8 to 15	8	File type	File type (in ASCII codes)
16 to 19	4	Starting address	The starting address of the ROM area secured as a file. The binary address value is expressed in 4-digit hexadecimal numbers (ASCII codes).
20 to 23	4	Ending address +1	The address next to the ending address of the ROM area secured as a file. The binary address value is expressed in 4-digit hexadecimal numbers (ASCII code).
24 to 29	6	Date	Month, day, and year each expressed in 2-digit ASCII codes.
30 to 31	2		Unused.

8.8 ROM Cartridge Subroutine Table

Subroutine name	Entry point	Description
OPNPRM	FEEC	ROM file input open °Parameters: At Entry (A): This parameter specifies whether or not the filename is to be returned. 01: Return the filename opened in the packet. 00: Do not return the filename opened in the packet. (X): Starting address of packet °Packet 1. Filename (8 bytes) 2. File type (8 bytes) 3. Filename (8 bytes) (Enter the filename opened when the filename is to be returned.) 4. File type (8 bytes) (Enter the file type opened when the filename is to be returned.) Note: In the filename specification for the packet, if the string specifying a filename contains an asterisk (*), the filename matching terminates at the point of the asterisk and the system assumes that both the filenames have matched. In BASIC Version 1.0, when the matching of the filename with an asterisk (*) terminates, the system assumes that both the file types have also matched. (Note that the ROM file open procedure differs from the cassette file open procedure.) At Return (C): Abnormal I/O flag (A): Return codes 00: Normal A0: ROM cartridge not mounted A1: File not found A2: File already open A4: Invalid header data format A5: Invalid header address format (Z): This parameter depends on the value of parameter (A). °Packet (filename, file type) °Registers retained: None °Subroutines referenced: "PRMPON", "PREDBY", "HEXBIN" and "CLSPRM" °Temporary variables used: R0, R1, and R2

Subroutine name	Entry point	Description
REDPRM	FEE9	Input of one byte from ROM file. °Parameters: At Entry None At Return (C) Abnormal I/O flag (Always 00) (A): Input data (B): Return codes 00: Normal 01: End of file A3: File not opened (Z): This parameter depends on the value of parameter (B). °Registers retained: (X) °Subroutines referenced: "ADSTEP" °Temporary variables used: None
CLSPRM	FEE6	ROM file close. °Parameters: At Entry None At Return (C): Abnormal I/O flag °Registers retained: (B) and (X) °Subroutines referenced: "CHKRS" and "SNSCOM" Note: An attempt to close an unopened ROM file is not regarded by the system as an error.
DIRPRM	FEE3	ROM file directory read. This subroutine specifies record number of the directory and inputs the record. °Parameters: At Entry (A): Directory record number from 0 through 63 (D) (X): Starting address of memory locations here the directory record is stored. The size of each record must be 32 bytes. At Return: (C): Abnormal I/O flag (A): Return codes 00: Normal A0: ROM cartridge not connected A3: Invalid specification of the directory record number (Z): This parameter depends on the value of parameter (A). °Registers retained: None °Subroutines referenced: "PRMPON" "ADSTEP", "PREDBY" and "CLSPRM" °Temporary variables used: None

8.9 ROM Cartridge Work Areas

Address (from)	(to)	Variable name	Bytes	Description
208	208	PRMSTS	1	Status of the ROM file Bit 0: File open status flag 0: File not opened 1: File opened Bits 1 ~ 6: Undefined Bit 7: Power supply for ROM 0: OFF, 1: ON
209	20A	STAPRS	2	ROM addressing counter
20B	20C	FTADRS	2	Starting address of file
20D	20E	EDADRS	2	Ending address of file + 1

```
ERR  SEQ   LOC  OBJECT      PROGRAM  ROMOPT     --- ROM CARTRIDGE INTERFACE ROUTINE ---

00001                              NAM     ROMOPT
00002                              TTL     --- ROM CARTRIDGE INTERFACE ROUTINE ---
00003                          * FILE    'EX57'  BY  K.A
00004                              OPT     PAGE=55
00005                              OPT     LOAD
00006                          *
00007                          *
00008                          * COMMON DEFINITION
00009                          *
00010                          * MPU 6301 I/O PORT
00011          0002  A         PORT1   EQU     $02         * I/O PORT 1 (ADDRESS)
00012          0003  A         PORT2   EQU     $03         * I/O PORT 2 (ADDRESS)
00013                          *
00014                          * OTHER REGISTERS
00015                          * REGISTER MEANINGS
00016                          *
00017                          * PORT1 $02
00018                          *       0:R DATA SET READY (0:HIGH 1:LOW)
00019                          *       1:R CLEAR TO SEND (0:HIGH 1:LOW)
00020                          *       2:R PORT TO SLAVE P34 (SFLAG)
00021                          *       3:R INTERRUPT FROM EXTERNAL PORT (0:INTERRUPT)
00022                          *       4:R POWER FAIL (0:ABNORMAL)
00023                          *       5:R KEY BOARD INTERRUPT FLAG (0:INTERRUPT)
00024                          *       6:R PERIPHERAL STATUS (0:HIGH  1:LOW) (FROM SERIAL)
00025                          *       7:R MICRO CASSETTE COUNTER / MICRO CASSETTE EXIST
00026                          *
00027                          *  $26
00028                          *       0:W LCD COMAND/DATA 1
00029                          *       1:W LCD COMAND/DATA 2
00030                          *       2:W LCD COMAND/DATA 4
00031                          *       3:W LCD CMMAND/DATA SELCTION (0:DATA 1:COMMAND)
00032                          *       4:W KEY BOARD INTERRUPT MASK (0:CLOSE 1:OPEN)
00033                          *       5:W PERIPHERAL CONTROL (TO SERIAL)
00034                          *       6:W TO PLUG IN 1
00035                          *       7:W TO PLUG IN 2 AND SLAVE P40
0036                           *
```

```
EPR   SEQ   LOC  OBJECT      PROGRAM  ROMOPT     --- ROM CARTRIDGE INTERFACE ROUTINE ---

00038                        *  COMMON DEFINITION
00039                        *
00040                        * ZERO PAGE RAM
00041A 004E                          ORG     $4E
00042A 004E     0001  A      PWRFLG  RMB     1          * BIT 0-3: CLOCK POWER ON MODE
00043                        *                          *  $01:POWER ON BY CLOCK IN BASIC MODE
00044                        *                          *  $02:POWER ON BY CLOCK IN APLLICATION MODE
00045                        *                          * BIT 4-7: BEFOR POWER OFF, CALL PROCEDURE MODE
00046                        *                          *  $01:BEFOR POWER OFF, CALL PROCEDURE IN
00047                        *                          *      BASIC MODE.
00048                        *                          *  $02:BEFOR POWER OFF, CALL PROCEDURE IN
00049                        *                          *      APLLICATION MODE.
00050                        *
00051A 004F     0001  A      P26     RMB     1          * VALUE OF ADDRESS $26
00052                        * GENERAL REGISTERS USED BY I/O ROUTINE
00053           0050  A      R0      EQU     *          * 2 BYTES REGISTER  (R0H,R0L)
00054A 0050     0001  A      R0H     RMB     1
00055A 0051     0001  A      R0L     RMB     1
00056           0052  A      R1      EQU     *          * 2 BYTES REGISTER  (R1H,R1L)
00057A 0052     0001  A      R1H     RMB     1
00058A 0053     0001  A      R1L     RMB     1
00059           0054  A      R2      EQU     *          * 2 BYTES REGISTER  (R2H,R2L)
00060A 0054     0001  A      R2H     RMB     1
00061A 0055     0001  A      R2L     RMB     1
00062           0056  A      R3      EQU     *          * 2 BYTES REGISTER  (R3H,R3L)
00063A 0056     0001  A      R3H     RMB     1
00064A 0057     0001  A      R3L     RMB     1
00065                        *
00066A 007C                          ORG     $7C
00067A 007C     0001  A      SIOSTS  RMB     1          * SLAVE I/O STATUS   (EACH BIT  0:OFF,  1:ON)
00068                        *                          * BIT 0: PRINTER
00069                        *                          * BIT 1: EXTERNAL CASSETTE
00070                        *                          * BIT 2: INTERNAL CASSETTE
00071                        *                          * BIT 3: RS232C ON (READ)
00072                        *                          * BIT 4: SPEAKER ON
00073                        *                          * BIT 5: PROM CASSETTE
00074                        *                          * BIT 6: BAR CODE READER
00075                        *                          * BIT 7: BREAK SLAVE CPU (0:ON EXECUTE
00076                        *                          *                        1:BROKEN BY INTERRUPT)
00077A 007D     0001  A      MIOSTS  RMB     1          * MAIN I/O STATUS  EACH BIT (0:OFF 1:ON)
00078                        *                          * BIT 0: LCD ON READ/WRITE CHARACTERS
00079                        *                          * BIT 1: NOW SENDING COMMAND TO SLAVE CPU
00080                        *                          * BIT 2: NOW TRANSMITTING DATA TO SERIAL (1:ON)
00081                        *                          * BIT 3: ON CLOCK INTERRUPT (1:ON)
00082                        *                          * BIT 4: (POWER FAIL)
00083                        *                          * BIT 5: (OFF POWER SWITCH)
00084                        *                          * BIT 6: ON PAUSE KEY
00085                        *                          * BIT 7: ON BREAK KEY
```

```
EPP   SEQ   LOC  OBJECT      PROGRAM  ROMOPT     --- ROM CARTRIDGE INTERFACE ROUTINE ---

      00087                          * ROM CASSETTE WORK AREA
      00088A 0208                      ORG     $208
      00089            0208  A       PRWKTP EQU     *          * ROM WORK TOP
      00090A 0208      0001  A       PRMSTS RMB     1          * ROM STATUS (BIT7:POWER ON  1:ON 0:OFF
      00091                          *                         *             (BIT0:OPEN FLAG  1:OPEND  0:CLOSE
      00092A 0209      0002  A       STADRS RMB     2          * ROM ADDRESS COUNTER
      00093A 0208      0002  A       FTADRS RMB     2          * ADDRESS OF TOP OF FILE
      00094A 020D      0002  A       EDADRS RMB     2          * ADDRESS OF LAST OF FILE +1
```

```
ERR   SEC   LOC  OBJECT      PROGRAM  ROMOPT     --- ROM CARTRIDGE INTERFACE ROUTINE ---

00097A 1000                          ORG     $1000
00098                          *
00099          FF2E  A         CHKPLG EQU     $FF2E
00100          FF19  A         SNSCOM EQU     $FF19
00101          FF16  A         CHKRS  EQU     $FF16
00102          FED4  A         WRTP26 EQU     $FED4
00103          FF28  A         HEXBIN EQU     $FF28
00104                          *
00105          0051  A         CMPRON EQU     $51       * ROM POWER ON COMMAND TO SLAVE MCU
00106          0052  A         CMPROF EQU     $52       * ROM POWER OFF COMMAND TO SLAVE MCU
00107                          *
00108          012E  A         FILBYT EQU     $12E      * REST BYTES IN THE FILE (2 BYTES SIZE)
00109                          *
00110                          *
00111                          * HEADER FORMAT OF PROM
00112                          *  00 - 07 (DEC) : FILE NAME    (00: $00:DELETED   $FF:END OF HEADER)
00113                          *  08 - 15       : FILE TYPE
00114                          *  16 - 19       : TOP ADDRESS OF THE FILE
00115                          *  20 - 23       : BOTTOM ADDRESS + 1
00116                          *  24 - 29       : DATE
00117                          *  30 - 31       : NOT USED
00118                          *
00119                          *
00120                          *
00121                          *  FUNCTION :  OPEN TO READ
00122                          *  ON ENTRY
00123                          *    (A)=READ MODE(0:NOT ANSWER FILE NAME
00124                          *                  1:ANSWER FILE NAME )
00125                          *    (X)=PACKET ADDRESS
00126                          *          PACKET  0-7: FILE NAME
00127                          *                  8-15: FILE TYPE
00128                          *  ON EXIT
00129                          *    (A)=RETURN CODE
00130                          *                    $00:NORMAL
00131                          *                    $A0:WITHOUT ROM CASSETTE
00132                          *                    $A1:FILE IS NOT FOUND
00133                          *                    $A2:ALREADY OPEN
00134                          *                    $A3:DIRECTRY NUMBER ERROR
00135                          *                    $A4:ROM FORMAT ERROR
00136                          *                    $A5:ADDRESSING ERROR
00137                          *    (C)=0
00138                          *    (Z) DEPEND ON VALUE OF (A)
00139                          *    PACKET
00140                          *        16-23: FOUND FILE NAME(WHEN 'ANSWER FILE NAME' MODE)
00141                          *        24-31: FOUND FILE TYPE( ..)
00142                          *  REGISTER PRESERVE
00143                          *    NONE
00144                          *
00145                          *  WORK AREA AS REGISTER
00146                          *   R0: SAVE PACKET ADDRESS
00147                          *   R1H: SAVE MODE WHEN OPEN PROCEDURE WAS CALLED (VALUE OF (A))
00148                          *   R1L: THE FLAG WHETHER FOUND FILE NAME IS MATCHED
00149                          *        (BIT 7:STOP TO COMPARE     0:CONTINUE TO COMPARE 1:STOP)
00150                          *        (BIT 0-4:FLAG FILE NAME IS MATCHED (0:MATCHED,  OTHERS:NO)
00151                          *   R2H: READ CHARACTER (READ BYTE ROUTINE)
```

ERR SEQ LOC OBJECT PROGRAM ROMOPT --- ROM CARTRIDGE INTERFACE ROUTINE ---

```
00152                          *    R2L: HEADER NUMBER
00153                          *
00154A 1000 97 52     A        OPNPRM STA A  R1H        * SAVE MODE 'ANSWER FILE NAME OR NOT'
00155A 1002 DF 50     A               STX    R0         * SAVE PACKET ADDRESS
00156                          *
00157A 1004 BD 10EC   A               JSR    PRMPON     * WITH ROM CARTRIDGE ? (RESET ADDRESS COUNTER)
00158A 1007 26 7B 1084*               BNE    OPNP67     * IF NONEZERO, ERROR DETECT.
00159A 1009 97 55     A               STA A  R2L        * HEADER NUMBER = 0
00160A 100B 86 81     A               LDA A  #$81       * SET OPEN AND POWER ON FLAG
00161A 100D B7 0208   A               STA A  PRMSTS
00162                          *
00163                          * READ HEADER AND SEARCH FILE NAME
00164A 1010 5F                 OPNP20 CLR B             * (B): DATA COUNTER (0 - $0F)
00165A 1011 D7 53     A               STA B  R1L        * FLAG (NAME IS MATCHED)
00166A 1013 DE 50     A        OPNP25 LDX    R0         * (X): PACKET ADDRESS
00167A 1015 3A                        ABX
00168A 1016 BD 10C2   A               JSR    PREDBY     * READ ONE CHARACTER FROM THE ROM
00169A 1019 25 72 108D*               BCS    OPNP80
00170A 101B 5D                        TST B            * ADDRESS = FIRST COLUMN OF FILE NAME ?
00171A 101C 26 07 1025                BNE    OPNP26
00172A 101E 81 FF     A               CMP A  #$FF       * NOT FOUND ?   (LAST DIRECTRY MARK= $FF)
00173A 1020 27 6C 108E                BEQ    OPNP90
00174A 1022 4D                        TST A            * DELETED ?  (DELETED FILE MARK = $00)
00175A 1023 27 2A 104F                BEQ    OPNP35
00176A 1025 7D 0052   A        OPNP26 TST    R1H        * 'ANSWER FILE NAME' MODE ?
00177A 1028 27 02 102C                BEQ    OPNP27
00178A 102A A7 10     A               STA A  16,X       * YES, STORE FILE NAME TO DATA PACKET.
00179A 102C 7D 0053   A        OPNP27 TST    R1L        * STOP TO COMPARE (FILE NAME IS MATCHED) ?
00180A 102F 2B 14 1045                BMI    OPNP29
00181A 1031 36                        PSH A            * '*': MARK TO STOP TO COMPARE.
00182A 1032 86 2A     A               LDA A  #'*
00183A 1034 A1 00     A               CMP A  0,X
00184A 1036 32                        PUL A
00185A 1037 26 05 103E                BNE    OPNP28
00186A 1039 72 8053   A               OIM    #$80,R1L   * '*' MARK. SET 'STOP COMPARE' BIT
00187A 103C 20 07 1045                BRA    OPNP29
00188                          *
00189A 103E A1 00     A        OPNP28 CMP A  0,X        * COPARE FILE NAME.
00190A 1040 27 03 1045                BEQ    OPNP29
00191A 1042 7C 0053   A               INC    R1L        * SET 'FILE NOT MATCHED' FLAG
00192                          *
00193A 1045 5C                 OPNP29 INC B             * FINISH TO COMPARE ?
00194A 1046 C1 10     A               CMP B  #16        * FILE NAME AND FILE TYPE HAVE 16 BYTES LENGTH
00195A 1048 26 C9 1013                BNE    OPNP25
00196                          * FILE NAME AND FILE TYPE ARE COPLETED TO COMPARE.
00197A 104A 7B 0F53   A               TIM    #$F,R1L    * OK ?
00198A 104D 27 12 1061                BEQ    OPNP50     *
00199                          * NO, COMPARE NEXT HEADER
00200A 104F 7C 0055   A        OPNP35 INC    R2L        * R2L: HEADER NUMBER (NEXT)
00201A 1052 D6 55     A               LDA B  R2L        * ADDRESS OF HEADER = '32' * 'HEADER NUMBER'
00202A 1054 C1 40     A               CMP B  #64        *
00203A 1056 2A 36 108E                BPL    OPNP90     * LIMIT OF THE HEADER ($000 - $3FF)
00204A 1058 86 20     A               LDA A  #32
00205A 105A 3D                        MUL
00206A 105B 18                        XGDX             *  (X) : NEXT ADDRESSING POINTER
```

ERR SEQ LOC OBJECT PROGRAM ROMOPT --- ROM CARTRIDGE INTERFACE ROUTINE ---

```
00207                                *
00208A 105C BD 1155  A               OPNP20 JSR    ADSTEP    * SET ADDRESSING COUNTER TO FIRST COLUMN OF
00209A 105F 20 AF 1010               BRA    OPNP20   *  THE HEADER.
00210                                *
00211                                * TOP ADDRESS AND LAST ADDRESS WHICH ARE SHOWN BY ASCII CODE ARE
00212                                *  CONVERTED TO BINARY VALUE.
00213A 1061 CE 0204  A               OPNP50 LDX    #PRWKTP-4
00214A 1064 8D 5C 10C2               OPNP65 BSR    PREDBY    * (A,B) <--- ASCII CODED HEXADECIMAL VALUE.
00215A 1066 36                              PSH A
00216A 1067 8D 59 10C2                      BSR    PREDBY
00217A 1069 16                              TAB
00218A 106A 32                              PUL A
00219A 106B BD FF2B  A                      JSR    HEXBIN    * CONVERT HEX TO BINARY.
00220A 106E 26 15 1085                      BNE    OPNP70    * ERROR ?
00221A 1070 A7 07    A                      STA A  FTADRS-PRWKTP+4,X
00222A 1072 08                              INX
00223A 1073 8C 0208  A                      CPX    #PRWKTP
00224A 1076 26 EC 1064                      BNE    OPNP65
00225                                *
00226A 1078 EC 05    A                      LDD    EDADRS-PRWKTP,X * 'EDADRS' <--- LAST ADDRESS
00227A 107A A3 03    A                      SUBD   FTADRS-PRWKTP,X * 'FTADRS' <--- TOP ADDRESS
00228A 107C FD 012E  A                      STD    FILBYT    * 'FILBYT' <---- DATA NUMBER IN THE FILE.
00229                                *
00230A 107F 86 81    A                      LDA A  #$81      * SET 'OPENED FILE' FLAG.
00231A 1081 A7 00    A                      STA A  PRMSTS-PRWKTP,X
00232A 1083 4F                              CLR A
00233A 1084 39                       OPNP67 RTS
00234                                *
00235A 1085 86 A4    A               OPNP70 LDA A  #$A4      * FORMAT ERROR
00236A 1087 36                       OPNP75 PSH A
00237A 1088 BD 113D  A                      JSR    CLSPRM    * ERROR CLOSE
00238A 108B 32                              PUL A
00239A 108C 16                              TAB              * SET (Z), (N)
00240A 108D 39                       OPNP80 RTS
00241                                *
00242A 108E 86 A1    A               OPNP90 LDA A  #$A1      * RETURN CODE (FILE WAS NOT FOUND)
00243A 1090 20 F5 1087                      BRA    OPNP75
00244                                *
00245                                *
00246                                *  FUNCTION :  READ ONE CHARACTER FROM ROM FILE
00247                                *  ON ENTRY
00248                                *    NONE PARAMETER
00249                                *  ON EXIT
00250                                *    (A)=READ DATA
00251                                *    (B)=STATUS   $00:NORMAL  $01:END OF FILE
00252                                *                 $A3:FILE NOT OPEN
00253                                *    (C)=0
00254                                *    (Z)=DEPEND ON VALUE OF (B)
00255                                *  REGISTER PRESERVE
00256                                *    (X)
00257                                *
00258A 1092 3C                       REDPRM PSHX              * SAVE (X)
00259                                *
00260A 1093 C6 A3    A                      LDA B  #$A3      * PRESET ERROR CODE (FILE IS NOT OPEND)
00261A 1095 B6 0208  A                      LDA A  PRMSTS    * IS POWER ON ? (BIT0, BIT7 BOTH ON)
```

```
ERR   SEQ   LOC  OBJECT      PROGRAM  ROMOPT     --- ROM CARTRIDGE INTERFACE ROUTINE ---

     00262A 1098 2A 0D 10A7            BPL    REDP08
     00263A 109A 47                    ASR A
     00264A 109B 24 0A 10A7            BCC    REDP08
     00265                       *
     00266A 109D FC 020D  A      REDP05 LDD    EDADRS     * IS CURRENT ADDRESS BOTTOM IN THE FILE ?
     00267A 10A0 B3 0208  A             SUBD   FTADRS
     00268A 10A3 26 06 10AB            BNE    REDP10
     00269                       *
     00270A 10A5 C6 01    A            LDA B  #1         * EOF RETURN
     00271A 10A7 4F              REDP08 CLR A
     00272A 10A8 5D                    TST B             * SET (Z), (N), CLEAR (C).
     00273A 10A9 38                    PUL X
     00274A 10AA 39                    RTS
     00275                       * READ ON BYTES FROM FILE
     00276          10AB  A      REDP10 EQU    *
     00277A 10AB 18                    XGDX
     00278A 10AC 09                    DEX
     00279A 10AD FF 012E  A            STX    FILBYT     * SET 'REST DATA NUMBER IN THE FILE'
     00280A 10B0 C6 A5    A            LDA B  #$A5       * PRESET 'ADDRESSING ERROR' FLAG
     00281A 10B2 25 F3 10A7            BCS    REDP08
     00282A 10B4 FE 0203  A            LDX    FTADRS     * NON ERROR,
     00283A 10B7 3C                    PSHX
     00284A 10B8 BD 1155  A            JSR    ADSTEP     * ROM ADDRESSING <---- +1 INCREMENT
     00285A 10BB 38                    PUL X
     00286A 10BC 08                    INX
     00287A 10BD FF 0203  A            STX    FTADRS     * ADDRESSING COUNTER <--- +1 INCREMENT
     00288A 10C0 5F                    CLR B             * RETURN CODE = NORMAL
     00289A 10C1 38                    PUL X
     00290                       *
     00291                       *
     00292                       *  ENTRY POINT 'READ NEXT ONE BYTE'
     00293                       * ON ENTRY
     00294                       *  PARAMETER : NONE
     00295                       *  READ ONE BYTE AND INCREMENT ADDRESSING COUNTER
     00296                       * ON EXIT
     00297                       *  (A) READ CHARACTER
     00298                       * REGISTER PRESERVE
     00299                       *  (B), (X).
     00300                       *  WORK AS REGISTER
     00301                       *  R2H: COUNTER FOR 8 TIMES AND WORKAREA FOR READ DATA
     00302                       *      R2H   C  BIT7                          BIT0
     00303                       *              0   0   0   0   0   0   0   1
     00304                       *                      I
     00305                       *                      I
     00306                       *                      V
     00307                       *          0   0   0   0   0   0   0   1   X
     00308                       *                                      X:READ BIT
     00309                       *                      I
     00310                       *                      I
     00311                       *                      V
     00312                       *          0   0   0   0   0   0   1   X   X
     00313                       *                                      X:READ BIT
     00314                       *
     00315                       *
     00316A 10C2 37              PREDBY PSH B
```

ERR SEQ LOC OBJECT PROGRAM ROMCPT --- ROM CARTRIDGE INTERFACE ROUTINE ---

```
00317A 10C3 86 01     A             LDA A  #$1        * 1: MARK FOR 8 TH TIME.
00318A 10C5 97 54     A             STA A  R2H
00319A 10C7 5F                      CLR B
00320A 10C8 C4 7F     A      REDP20 AND B  #$FF-$80   * BIT 7 LOW (D7)
00321A 10CA 86 C0     A             LDA A  #$C0       * BIT6,7 EFFECTIVE
00322A 10CC BD FED4   A             JSR    WRTP26     * CLOCK LOW (FIRST TIME: D6 LOW)
00323A 10CF CA 80     A             ORA B  #$80       * CLOCK HIGH (FIRST TIME :READ DATA
00324A 10D1 BD FED4   A             JSR    WRTP26     *              SECOND TIME:SHIFT DATA)
00325                        *
00326A 10D4 96 02     A             LDA A  PORT1      * INPUT DATA (BIT7, BIT6 ,...)
00327A 10D6 48                      ASL A
00328A 10D7 79 0054   A             ROL    R2H        * R2L:SHIFT ONE BIT WHICH WAS GET.
00329A 10DA CA 40     A             ORA B  #$40       * FOR D6:HIGH
00330A 10DC 24 EA 10C8              BCC    REDP20     * COMPLETE TO READ 8 BITS ?
00331                        *
00332A 10DE FC 0209   A             LDD    STADRS     * ASSRESSING POINTER <--- +1 INCREMENT
00333A 10E1 C3 0001   A             ADDD   #1
00334A 10E4 FD 0209   A             STD    STADRS
00335                        *
00336A 10E7 96 54     A             LDA A  R2H        * (A) <--- READ DATA
00337A 10E9 33                      PUL B
00338A 10EA 5D                      TST B             * CLEAR (C), SET (Z) FOR 'REDPRM' ROUTINE
00339A 10EB 39                      RTS
00340                        *
00341                        *
00342                        * POWER ON ROM
00343                        *  PROCEDURE
00344                        *  1:CHECK PLUGIN OPTION (ROM) ?
00345                        *  2:CLEAR ADDRESSING COUNTER
00346                        *  3:POWER ON
00347                        * PARAMETER
00348                        *   ON ENTRY    NONE
00349                        *   ON EXIT
00350                        *      (A):RETURN CODE    00:NORMAL  OTHERS:ERROR
00351                        *      (C):I/O ERROR FLAG
00352                        *      (Z):DEPEND ON VALUE OF (A)
00353                        * REGISTER PRESERVE
00354                        *       (X)
00355                        *
00356A 10EC BD FF2E   A      PRMPON JSR    CHKPLG     * CHECK PLUG-IN OPTION
00357A 10EF 25 26 1117              BCS    PRMP80
00358A 10F1 16                      TAB
00359A 10F2 26 23 1117              BNE    PRMP80
00360A 10F4 72 207C   A             OIM    #$20,SIOSTS * SLAVE ROM CASSETTE ON
00361A 10F7 FD 0209   A             STD    STADRS     * ROM ADDRESS = 0    (A,B)=0
00362A 10FA 86 C0     A             LDA A  #$C0
00363A 10FC BD FED4   A             JSR    WRTP26     * SET D6,D7 LOW (COUNT, CLOCK)
00364A 10FF 86 51     A             LDA A  #$51
00365A 1101 BD FF19   A             JSR    SNSCOM     * SEND 'PROM ON COMMAND' TO SLAVE MCU.
00366A 1104 25 11 1117              BCS    PRMP80
00367A 1106 3C                      PSHX
00368A 1107 CE 0190   A             LDX    #400       * WAIT 2 M SEC
00369A 110A 09               PRMP20 DEX
00370A 110B 26 FD 110A              BNE    PRMP20
00371A 110D FE 0208   A             LDX    PRMSTS     * SET POWER ON FLAG (ON BIT7)
```

```
ERR  SEQ   LOC  OBJECT      PROGRAM  ROMOPT     --- ROM CARTRIDGE INTERFACE ROUTINE ---

00372A 1110 62 8000  A            OIM    #$80,0,X
00373A 1113 38                    PULX
00374A 1114 4F                    CLR A
00375A 1115 20 33 114A            BRA    CLSP10     * (JMP  CHKRS)
00376                      *
00377A 1117 86 A0    A     PRMP80 LDA A  #$A0        * WITHOUT ROM CASSETTE (ERROR)
00378A 1119 39                    RTS
00379                      *
00380                      *
00381                      *  FUNCTION :  READ DIRECTORY
00382                      *  ON ENTRY
00383                      *    (A):DIRECTORY NUMBER (FROM 0 TO 63)
00384                      *    (X):ADDRESS WHERE HEADER ARE STORED
00385                      *  ON EXIT
00386                      *    (A):RETURN CODE  $00: NORMAL
00387                      *                     $A0: WITHOUT ROM CASSETTE
00388                      *                     $A3: DIRECTRY NUMBER ERROR
00389                      *    (C)=0
00390                      *    (Z)=DEPEND ON VALUE OF (A)
00391                      *  REGISTER PRESERVE
00392                      *   NONE
00393                      *
00394A 111A 16             DIRPRM TAB                * SAVE DIRECTORY NUMBER
00395A 111B 86 A3    A           LDA A  #$A3         * (A) <--DIRECTORY ERROR FLAG   (PRESET)
00396A 111D C1 40    A           CMP B  #64          * IS DIRECTRY NO. LIMIT (00 - 63) OK ?
00397A 111F 24 29 114A           BCC    CLSP10
00398                      *
00399A 1121 DF 50    A           STX    R0           * SAVE ADDRESS OF DIRECTORY
00400                      *
00401A 1123 37                   PSH B
00402A 1124 8D C6 10EC           BSR    PRMPON       * POWER ON (CHECK PROM)
00403A 1126 33                   PUL B
00404A 1127 26 21 114A           BNE    CLSP10
00405                      *
00406A 1129 86 20    A           LDA A  #32          * CALCULATE HEADER ADDRESS (32 * 'NUMBER')
00407A 112B 3D                   MUL
00408A 112C 18                   XGDX
00409A 112D 8D 26 1155           BSR    ADSTEP       * SET ROM ADDRESS
00410A 112F C6 20    A           LDA B  #32
00411A 1131 DE 50    A           LDX    R0
00412A 1133 37             DIRP10 PSH B
00413A 1134 8D 8C 10C2*          BSR    PREDBY       * READ ONE CHARACTER
00414A 1136 A7 00    A           STA A  0,X
00415A 1138 08                   INX
00416A 1139 33                   PUL B
00417A 113A 5A                   DEC B
00418A 113B 26 F6 1133           BNE    DIRP10
00419                      *
00420                      *
00421                      *  FUNCTION :  CLOSE ROM CASSETTE
00422                      *  ON ENTRY
00423                      *    PARAMETER  NONE
00424                      *  ON EXIT
00425                      *    (C): I/O ERROR FLAG
00426                      *  REGISTER PRESERVE
```

```
ERR   SEQ   LOC  OBJECT      PROGRAM  ROMOPT     --- ROM CARTRIDGE INTERFACE ROUTINE ---

     00427                          *     (B),(X)
     00428                          *
     00429A 113D 7F 0208  A         CLSPRM CLR    PRMSTS    * SET ROM STATUS  'POWER OFF', 'CLOSED FILE'
     00430A 1140 86 52    A                LDA A  #CMPROF   * SEND 'POWER OFF COMMAND' TO SLAVE MCU.
     00431A 1142 BD FF19  A                JSR    SNSCOM
     00432A 1145 71 DF7C  A                AIM    #$FF-$20,SIOSTS * SET FLAG ('ROM CASSETTE IS OFF')
     00433A 1148 86 00    A                LDA A  #0        * (DO NOT CHANGE (C) BIT)
     00434A 114A 7E FF16  A         CLSP10 JMP    CHKRS     * RECOVER RS232 (OPEN TO READ RS232)
     00435                          *
     00436                          *
     00437                          *  FUNCTION :  SET PROM ADDRESS TO DESTINATED VALUE
     00438                          *  ON ENTRY
     00439                          *   (X)= TARGET ADDRESS
     00440                          *  ON EXIT
     00441                          *   (C): I/O ERROR FLAG
     00442                          *  REGISTER PRESERVE
     00443                          *   NONE
     00444                          *
     00445          114D  A         ADST00 EQU    *          * CASE OF (NEW ADDRESS < CURRENT ADDRESS)
     00446A 114D 8D 9D 10EC                ESR    PRMPON     * WITHOUT ROM ? (CLEAR ADDRESSING COUNTER)
     00447A 114F 26 20 1171                BNE    ADST80     *  WITHOUT ?
     00448A 1151 5F                        CLR B             * IF ROM  (A):0
     00449A 1152 FD 0209  A                STD    STADRS     * ROM ADDRESSING COUNTER <--- 0
     00450                          *
     00451                          *  ENTRY POINT OF 'ADSTEP' ROUTINE
     00452A 1155 3C                 ADSTEP PSHX              * (A,B)<---(X)
     00453A 1156 32                        PUL A
     00454A 1157 33                        PUL B
     00455A 1158 B3 0209  A                SUBD   STADRS     * NEW ADDRESS >= CURRENT ADDRESS ?
     00456A 1158 27 14 1171                BEQ    ADST80     * = ?
     00457A 115D 25 EE 114D                BCS    ADST00
     00458                          *                          * CASE OF 'TARGET ADDRESS > CUPRENT ADDRESS'
     00459A 115F FF 0209  A                STX    STADRS     * SET NEW ADDRESS TO 'STADRS'
     00460A 1162 18                        XGDX              * (X)<--- STEP COUNT
     00461                          *
     00462A 1163 5F                 ADST30 CLR B             * COUNT UP ADDRESSING COUNTER
     00463A 1164 86 C0    A                LDA A  #$C0
     00464A 1166 BD FED4  A                JSR    WRTP26
     00465A 1169 C6 40    A                LDA B  #$40
     00466A 116B BD FED4  A                JSR    WRTP26
     00467A 116E 09                        DEX
     00468A 116F 26 F2 1163                BNE    ADST30
     00469A 1171 39                 ADST80 RTS
     00470                          *
     00471          0000  A                END
***** TOTAL ERRORS     0
```

CHAPTER 9 LOAD MODULE

9.1 General

The module format for output of data by the SAVEM command in BASIC or the W command in the Monitor is a special format called a "Binary Load Module format". One file is divided into a number of records each containing memory addresses and data (Fig. 9-1).

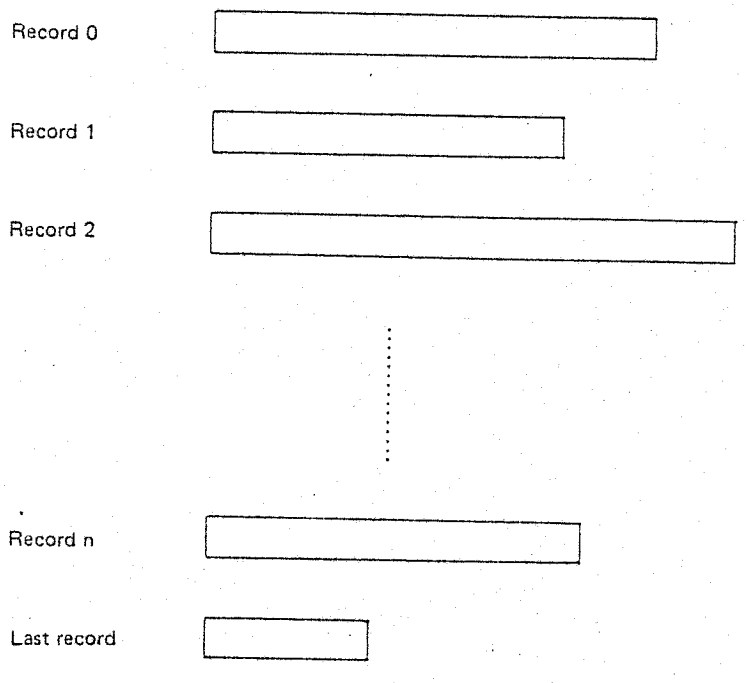

Fig. 9-1 Division of File into Records

Each record has a maximum length of 259 bytes and each data contained in the record is represented in binary numbers in units of one byte. The format of each record is shown below.

9.2 Load Module (Machine Language) Format

(1) Intermediate record

Column	Size (bytes)	Item	Description
0	1	Record length	Indicates the length of the data contained in the record in binary numbers (00 through FF).
1 ~ 2	2	Address	Indicates the address of the first data in the record in binary numbers 0000 through FFFF (in order of the upper and lower digits).
3	1	Data	Data 1. Namely, first data (00 through FF).
4	1	Data	Data 2
⋮			
n	1	Data	Data n (n must be a value in the range of 0 to 255.)
n + 1	1	Checksum	This value must be such that the low-order 8 bits of the sum of the data values in columns 0 through n + 1 become 0.

(2) Last record

Column	Size (bytes)	Item	Description
0	1	Record length	This value must always be 0.
1 ~ 2	2	Address	Indicates the entry point of a program in binary numbers (0000 through FFFF in order of the upper and lower digits).
3	1	Checksum	This value must be such that the low-order 8 bits of the sum of the data values in columns 0 through 3 become 0.

9.3 Dump/Load Procedures

9.3.1 I/O devices

The basic I/O routines support the following devices:

(1) Input
    (a) External audio cassette
    (b) Built-in microcassette
    (c) ROM cartridge
(2) Output
    (a) External audio cassette
    (b) Built-in microcassette

9.3.2 Dump/load procedures

The memory contents in the binary load module format are transferred to and from an external storage as follows:

(1) Output to the external storage
    (a) File opening
    Subroutine "OPNDMP" is provided to open the specified file (device) for output. Subroutine "OPNWCS" is called if the specified file is an external audio cassette.
    (b) Output of the memory contents
        Subroutine "BIDUMP" is provided to output the memory contents in the binary load module format to the opened file and closes it upon completion of the dumping.
(2) Input from the external storage
    (a) File opening
        Subroutine "OPNLOD" is provided to open the specified file (device) for input. Subroutine "OPNPRM" is called if the specified file is a ROM cartridge.
    (b) Loading into memory
        Subroutine "BILOAD" is provided to store the input data in the binary load module format in the main memory and closes the file upon completion of the loading.

9.4 Binary Dump/Load Subroutine Table

Subroutine name	Entry point	Description
OPNDMP	FEEO	Binary memory dump open. This subroutine opens the file to be dumped in a binary absolute format and supports an external cassette and the built-in microcassette drive.
		Parameters: At Entry (X): Top address of a data packet (B): Device name 'M': Microcassette drive 'C': External audio cassette Packet 1. Interblock tape stop mode (1 byte) for external audio cassette or microcassette 00: Stop the tape between blocks. 01: Do not stop the tape between blocks. 2. Top address of buffer (2 bytes). The buffer size is 260 bytes. 3. Filename (8 bytes) 4. File type (8 bytes) 5. Dump start address (2 bytes) 6. Dump end address (2 bytes) 7. Offset value (2 bytes) 8. Program entry point (2 bytes) Note: The offset value is added to the dump start address, dump end address, or the program entry point as an unsigned binary number. Parameters: At Return (C): Abnormal I/O flag (A): Return code (This parameter is dependent on subroutines OPNNCS and OPNWMS.) Registers retained None Subroutines referenced OPNWMS, OPNWCS Variables to be used: R0, R1, R2, R3, R4, R5, R6, and R7
BIDUMP	FEDD	Binary memory dump. This subroutine dumps the memory contents in a binary absolute format to the file opened by subroutine OPNDMP and closes the file upon completion of the dumping.
		Parameters: At Entry None

		At Return Depend on subroutines WRTCS, WRTMS. Registers retained None Subroutines referenced WRTCS, WRTMS Variables used R0, R1, R2, R3, R4, R5, R6 and R7
OPNLOD	FEDA	Binary memory load. This subroutine opens the file to be loaded and loads the contents of the file dumped in binary absolute format, into memory.
		Parameters: At Entry (X): Top address of a data packet (B): Device name 'M': Microcassette 'C': External audio cassette 'P': ROM cartridge (A): Specifies whether or not the filename is to be returned. 00: Return the filename. 01: Do not return the filename. Packet 1. Interblock tape stop mode (1 byte) for external audio cassette or microcassette. 00: Stop the tape between blocks. 01: Do not stop the tape between blocks. FF: Depends on the header. 2. Top address of buffer (2 bytes) The buffer size is 260 bytes. 3. Filename (8 bytes) 4. File type (8 bytes) 5. Lower limit of the memory address to be loaded (2 bytes) 6. Upper limit of the memory address to be loaded (2 bytes) 7. Offset value (2 bytes) Note: The offset value is added to the address information of a file (load start address, load end address, or program entry point) as an unsigned binary number. The interblock tape stop mode is effective only for 'M' or 'C' but not for 'P'. If the return of a filename is specified by register (A), the filename is returned after the 19th byte of the packet. (In this case, the packet contents after the lower limit of the memory address are destroyed.) Since subroutines OPNRCS, SRCRCS, OPNRMS, SRCRMS, and OPNPRM are actually called, the packet depends on these subroutines.

		At Return (C): Abnormal I/O FLAG (A): Return codes 8C: Load area exceeds the specified memory space range. Other return codes depend on OPNRCS, SRCRCS, OPNRMS, SRCRMS and OPNPRM Registers retained None Subroutines referenced OPNRCS, SRCRCS, OPNRMS, SRCRMS, and OPNPRM Variables used R0, R1, R2, R3, R4, R5, R6, and R7
		Notes: Assuming that the upper- and lower-limit values of the memory addresses that can be loaded by the packet are μ and ℓ, respectively and that the address of a data to be loaded is α, the data can be loaded only when the following condition is satisfied. $\ell \leq \alpha \leq \mu$ If the address is not within this range, return code 8C (load area error) is output to interrupt the loading operation by force. The file is closed upon completion of the loading.
BILOAD	FED7	This subroutine loads the contents of the file opened by subroutine "OPNLOD" into the memory and closes the file upon completion of the loading.
		PaRAMETERS: At Entry (A): Specifies whether or not the contents of the file are to be loaded into the memory. 00: Load the contents of the file into the memory. 01: Check the load module format only. Do not load the file contents. At Return (X): Program entry point The value specified by the offset value is added to the value of the entry point recorded in the file. (A): Return codes 00: Normal 8C: Load area exceeds the specified memory space range. Others: Depend on the return codes of a file input routine. Registers retained None Subroutines referenced READMS, READCS, REDPRM Variables used R0, R1, R2, R3, R4, R5, R6 and R7

9.5 Binary Dump/Load Work Area

Address (From)	(To)	Variable name	Bytes	Description
20F	210	DLTPAD	2	First dump address
211	212	DLBTAD	2	Last dump address
213	214	DLOFAD	2	Offset value
215	216	DLSTAD	2	Program entry point
217	217	DLDVID	1	Dump/load device
218	218	DLSTS	1	Status work area (dummy)
219	21A	DLDVIX	2	Table address of a dump/load routine

ERR SEQ LOC OBJECT PROGRAM CLOCK --- CLOCK SAMPLE PROGRAM ---

```
00001                                   NAM     CLOCK
00002                           *
00003                           * DISPLAY CURRENT TIME ON THE PHISICAL SCREEN.
00004                           *  MPU IS SLEEP IF CLOCK UPDATE IS NOT CAUSED.
00005                           *
00006                           * FILE NAME     'EX$8'    BY  K.A
00007                                   TTL     --- CLOCK SAMPLE PROGRAM ---
00008                                   OPT     PAGE=55
00009                                   OPT     LOAD
00010A 1000                             ORG     $1000
00011                           *
00012                           * SUBROUTINE ENTRY POINT
00013          FFA9  A          SLEEP   EQU     $FFA9       * SLEEP CPU
00014          FF4C  A          DSPLCH  EQU     $FF4C       * DISPLAY ONE CHARACTER ON THE PHISICAL SCREEN
00015          FF49  A          DSPLCN  EQU     $FF49       * DISPLAY SOME CHARACTERS ON THE PHISICAL SCRE
00016                           *
00017                           *
00018A 1000                             ORG     $1000
00019                           *
00020A 1000 C6 00     A                 LDA B  #0          * CLEAR SCREEN
00021A 1002 BD FF49   A                 JSR     DSPLCN
00022A 1005 86 FF     A                 LDA A  #$FF        * ALARM INTERRUPT TIME
00023A 1007 97 41     A                 STA A  $41         *  = ANY TIME WHEN SECOND IS UPDATED.
00024A 1009 97 43     A                 STA A  $43
00025A 1008 97 45     A                 STA A  $45
00026                           *
00027A 100D 72 2048   A          CLCK10 OIM     #$20,$48   * ENABLE ALARM INTERRUPT.
00028A 1010 BD FFA9   A                 JSR     SLEEP      * MCU IS SLEEP FOR SAVE POWER.
00029A 1013 96 44     A                 LDA A  $44         * LOAD 'HOUR'
00030A 1015 16                          TAB                *  DISPLAY 'HOUR'
00031A 1016 84 F0     A                 AND A  #$F0        *  (HIGH ORDER)
00032A 1018 47                          ASR A
00033A 1019 47                          ASR A
00034A 101A 47                          ASR A
00035A 101B 47                          ASR A
00036A 101C 8A 30     A                 ORA A  #'0
00037A 101E CE 0502   A                 LDX     #$0502
00038A 1021 37                          PSH B
00039A 1022 BD FF4C   A                 JSR     DSPLCH
00040A 1025 32                          PUL A              * DISPLAY (LOW ORDER)
00041A 1026 84 0F     A                 AND A  #$0F
00042A 1028 8A 30     A                 ORA A  #'0
00043A 102A BD FF4C   A                 JSR     DSPLCH
00044A 102D 86 3A     A                 LDA A  #':         *
00045A 102F BD FF4C   A                 JSR     DSPLCH
00046A 1032 96 42     A                 LDA A  $42         * LOAD 'MINUTE'
00047A 1034 16                          TAB                *  DISPLAY 'MINUTE'
00048A 1035 84 F0     A                 AND A  #$F0        *  (HIGH ORDER)
00049A 1037 47                          ASR A
00050A 1038 47                          ASR A
00051A 1039 47                          ASR A
00052A 103A 47                          ASR A
00053A 103B 8A 30     A                 ORA A  #'0
00054A 103D CE 0302   A                 LDX     #$0302
00055A 1040 37                          PSH B
```

```
        *** 6301 CROSS MACROASSEMBLER   VER1.0 ***                    10/21/82    09:43:29
ERR   SEQ   LOC  OBJECT      PROGRAM  CLOCK      --- CLOCK SAMPLE PROGRAM ---

     00056A 1041 BD FF4C  A              JSR    DSPLCH
     00057A 1044 32                      PUL A                * DISPLAY (LOW ORDER)
     00058A 1045 84 0F    A              AND A  #$0F
     00059A 1047 8A 30    A              ORA A  #'0
     00060A 1049 BD FF4C  A              JSR    DSPLCH
     00061A 104C 86 3A    A              LDA A  #':           *
     00062A 104E BD FF4C  A              JSR    DSPLCH
     00063A 1051 96 40    A              LDA A  $40           * LOAD 'SECOND'
     00064A 1053 16                      TAB                  *  DISPLAY 'SECOND'
     00065A 1054 84 F0    A              AND A  #$F0          *  (HIGH ORDER)
     00066A 1056 47                      ASR A
     00067A 1057 47                      ASR A
     00068A 1058 47                      ASR A
     00069A 1059 47                      ASR A
     00070A 105A 8A 30    A              ORA A  #'0
     00071A 105C CE 0B02  A              LDX    #$0B02
     00072A 105F 37                      PSH B
     00073A 1060 BD FF4C  A              JSR    DSPLCH
     00074A 1063 32                      PUL A                * DISPLAY (LOW ORDER)
     00075A 1064 84 0F    A              AND A  #$0F
     00076A 1066 8A 30    A              ORA A  #'0
     00077A 1068 BD FF4C  A              JSR    DSPLCH
     00078A 1068 20 A0 100D              BRA    CLCK10-
     00079                       *
     00080          0000  A              END
***** TOTAL ERRORS    0

  ---- BINARY DUMP FORMAT OF OBJECT CODE ----

     13 10 00 C6 00 BD FF 49 86 FF 97 41 97 43 97 45 72 20 48 BD FF A9 BD
     13 10 13 96 44 16 84 F0 47 47 47 47 8A 30 CE 05 02 37 BD FF 4C 32 4A
     14 10 26 84 0F 8A 30 BD FF 4C 86 3A BD FF 4C 96 42 16 84 F0 47 47 47 62
     14 10 3A 47 8A 30 CE 08 02 37 BD FF 4C 32 84 0F 8A 30 BD FF 4C 86 3A 43
     12 10 4E BD FF 4C 96 40 16 84 F0 47 47 47 47 8A 30 CE 0B 02 37 40
     0D 10 60 BD FF 4C 32 84 0F 8A 30 BD FF 4C 20 A0 34
     00 10 00 F0
```