User's guide for the

MOTOROLA MICROCOMPUTER EVALUATION SYSTEM MES 6800

MOTOROLA Semiconductor Products Inc

USER'S GUIDE

FOR THE

MOTOROLA MICROCOMPUTER

EVALUATION SYSTEM

MES 6800

REVISION 5 - SEPT. 75 - TOULOUSE

TABLE OF CONTENTS

1.0 SYSTEMS OVERVIEW

The Evaluation system MES 6800 is supplied in a suit-case of 45 x 30 x 10 cm. dimensions which contains the Evaluation Card itself, the power supply, different connectors (220V/50Hz power supply connector, TTY or RS-232C connector, 2nd PIA connector) and the control switches for power on/off, restart and bit rate of 10 or 30 characters/second.

The M6800 Evaluation Card is a hardware/software system contained on a single printed circuit card. The function of this system is to allow the user to easily evaluate the various Microcomputer Family parts : MPU, ROM, RAM and PIA. The MES 6800 also provides the user with a 100 % functional system so that he does not have to debug the hardware before bringing up his system the first time.

The MES 6800 demonstrates the minimum system concept and illustrates the ease of use and flexibility of the M6800 System. It is compatible with the Motorola EXORciser* and can be used in conjunction with the other cards in that system.

2.0 FEATURES

The more important features of MES 6800 are as follows :

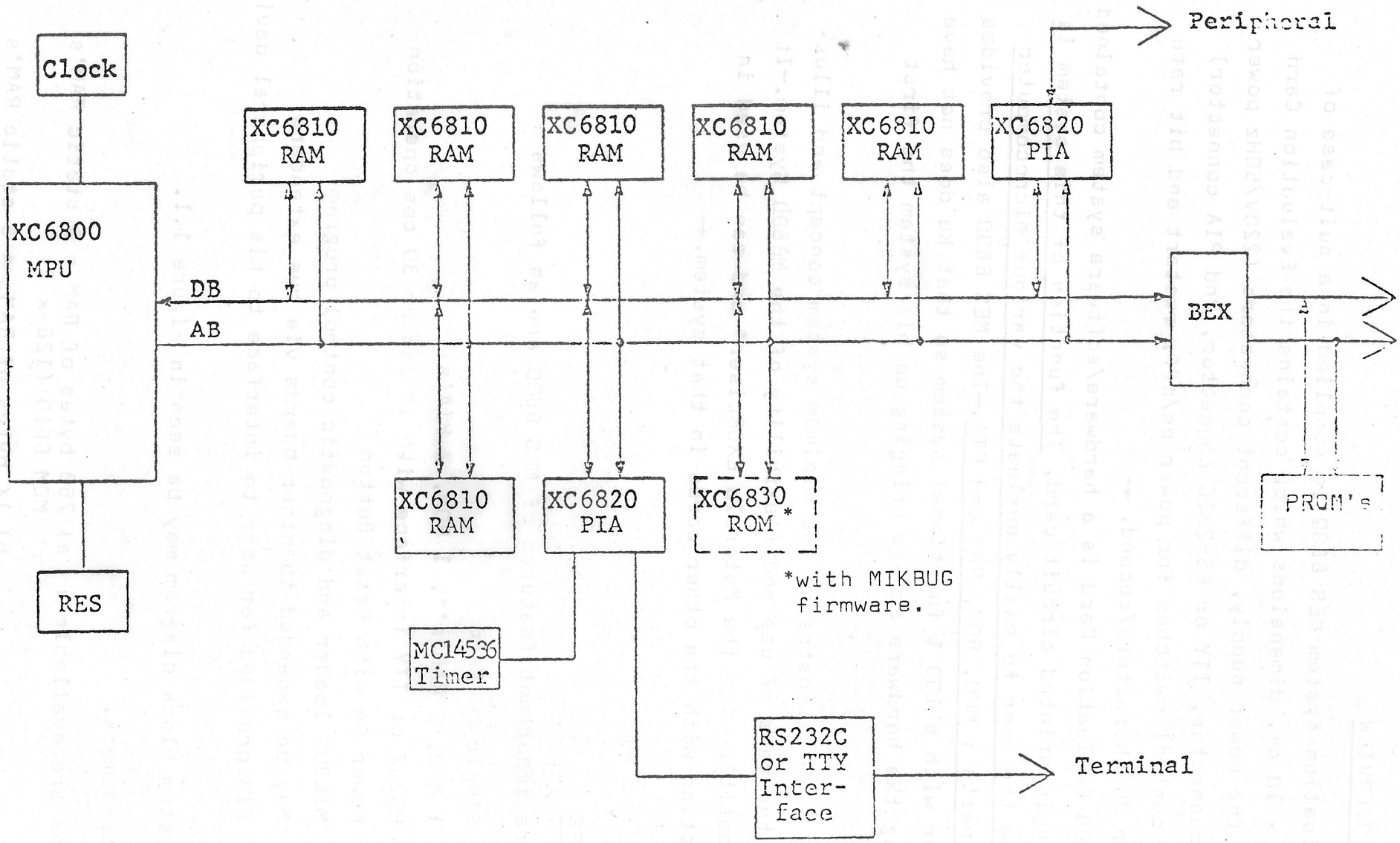
A. Single PC Card
B. 1 MPU, 6 RAM's**, 1 ROM, 2 PIA's
C. RS232 or TTY interface with 10 cps or 30 cps operation
D. Power On with Reset Button
E. MIKBUG loader and diagnostic control program
F. May be expanded to other boards via bus extenders
G. PIA provided for user to interface to his peripheral device.

The system block diagram may be seen in Figure 1.1.

*Motorola trademark.

**2 versions are available : a) 768 bytes of RAM = 6 static RAM's MCM 6810 (128 x 8)

b) 1K bytes of RAM = 8 static RAM's MCM 6602 (1K x 1)



Block Diagram of Evaluation System

Figure 1.1

3.0 USER'S INFORMATION

The MES 6800 System is supplied complete with its power supply. The user needs only to connect the system to 220V/50Hz power plug.

The Evaluation Card itself requires three power supplies : +5V, -12V, +12V. The +5V is for all logic on the Card. The ±12V is for the RS-232C and the Teletype* interface. The +5V supply requires 2.0 a. The ±12V require 250 ma. each.

P1 Connector

This is the connector at the bottom of the PC Card. To use the Card power must be connected to the appropriate pin's on the connector. For pinouts see Section 5.

> Warning : If the +5V and Gnd are connected up backwards to the Card all devices will be destroyed.

P2 Connector

The PIA at location 8008 is connected to this connector per the pinouts in Section 5. It is not necessary that this connector be connected for the Card to operate.

P3 Connector

Teletype Devices

This connector interfaces the Teletype or RS-232 type device to the Evaluation Card. The card is supplied with a cable for interfacing to a Teletype (TTY), 20 ma current loop interface. The Serial Output and Serial Input should be connected to the appropriate points on the TTY.

*Teletype trademark.

Serial Com. is the return line. If the terminal has a reader relay this may also be connected to Reader Control and Reader Control return. If this is not available these two wires may be left open. In this case the turn on and off of the reader must be done manually. The card will also operate with a TTY that has remote device control.

RS-232 Device

To use a RS232 type device with the evaluation card the header must be rewired as follows:

Header Evaluation Card		RS232 Connector (DB-25S) 25 pin female
RS-232C Input	P3-Pin 2	Pin 2
RS-232C Sig. Gnd.	P3-Pin 1	Pin 7
RS-232C Output	P3-Pin 3	Pin 3

Jumper Pins 5, 6, 8, and 20.

Note: Strap between E13 and E14 should be removed for use of RS232 Device and a Strap applied to E11 and E12 if TTY is not to be used.

Bit Rate Adjust

The bit rate is set up for TTY operation when the Card is received by the user. For 10 ch/s operation the zero level of the clock measured at pin 17 of the PIA (U26) should be set between 9,5 and 9,6 ms for 10 ch/s For 30 ch/s operation it should be set for 3.3 ms. This assumes that the MPU is operating at 1 MHz. If the MPU is operating at a lower speed than 1 MHz it will be desirable to set the interval for a smaller time period.

The bit rate is adjusted by the 50K pot (R23).

Ø1, Ø2 Clocks

The clock frequency that the system is operating at may be set by adjusting R9 and R13. R9 adjusts the Ø1 pulse width and R13 adjusts the Ø2 timing. Ø1 should be set for a minimum of 430 ns. at the 90 % point of VDD and Ø2 at 470 ns minimum at the 90 % points of VDD. Note that if slower rams are used in the system it may be necessary to set the clock for a slower period. Also note that adjustment of the MPU clocks will have some effect on the Bit Rate Timing. The pulse widths of Ø1, Ø2 must be measured at the MPU.

Strapping Options (Note : See Schematic Diagram MEC6800-1(A)S)

E1, E2

E1 connected to E2 disables TSC and enables the bus drivers. To use TSC this strap must be removed.

E3, E4

This option grounds NMI which is not functional in MPU A1 and A2 models. For Model B disconnect strap and connect a 2K Ω pullup resistor to +5V at E3.

E5, E6, E7

These options connect the ROM address line either to GND or A9 depending on the amount of the ROM to be used, Normally connect E5 to E6.

E8, E9, E10

E8 should be connected to E10 for operation of the PROM's. For external control of data bus and expansion to larger memory systems, E8 should be connected to E9.

E11, E12

These points should be shorted if the RS232 device is to be used. If the TTY is used these points should be open.

E13, E14

These points should be shorted for TTY operation and open for RS232 operation.

Warning : If +5V and GND are reversed all devices die !

Turn Power On

Check Bit Rate Clock for proper speed. Check MPU clock's for proper speed. Connect I/O device with header to Connector P3.
Press Red reset button. See software Section.

3.20 Software

3.21 MIKBUG General Description

MIKBUG is a loader and diagnostic control program that is supplied in firmware on the evaluation card. The various routines within MIKBUG are called by entering on the keyboard one of the following single character calls:

L - Memory Loader
P - PRINT/PUNCH Memory Dump
M - Memory Change
R - Display Contents of Target Registers
G - GO TO TARGET PROGRAM

Other features include the capability to set breakpoints with memory change and the use of interrupts which will be described in detail in the following pages.

The MIKBUG Program requires either a Teletype or teletypewritten replacement such as the Silent 700 ASR** for input/output.

RESTART FUNCTION

This button, located on the top of the evaluation card should be pushed when power has first been applied. The terminal should respond with a carriage return, line feed and an asterick. This indicates that the MIKBUG control program is now ready for input. Should control be lost during the execution of a user's program control may be regained by pushing the Restart Button.

**Registered trademark of Texas Instrument Inc.

MEMORY LOADER

The Memory Loader function (L) of MIKBUG loads formatted binary object tapes generated by the Assembler or the punch dump function (P) of MIKBUG.

The paper tape format is shown in Figure 3.1.

When loading from paper tape, all characters prior to the Start-of-Record character are ignored. Each record has a checksum which is verified as the data associated with the record is stored to memory. If the checksum is in error, loading is stopped, and a question mark is printed. The operator now has the options to: abort the load, repositioning and rereading the record (entering a L), or continuing the load ignoring the checksum error (entering a L). The continue option is not recommended in that if the error is in the address field the data may destroy data in another area of memory.

Example of Load

(underline indicates user Input)

```
*L
S113000002020202020286AA02023F86FC8810CC7B
S10400102635
S9
*
```

Here a short tape has been loaded using the load (L) command. The "S9" has been entered from the keyboard to exit the loader.

NOTE:

Tapes punched by MIKBUG do not have the End-of-File Record therefore a "S9" must be typed on the keyboard to exit the loader.

PRINT/PUNCH MEMORY DUMP

The PRINT/PUNCH Memory Dump (P) function of MIKBUG punches an absolute formatted binary object tape. The formatted binary object tape format is shown in Figure 3.1. The beginning and ending address must be entered using memory change. The beginning address is to be entered at location A002 and A003 and the ending address at locations A004 and A005. All address and data will be in hexadecimal. The beginning address is the address of the first byte to be punched and the ending address is the address of the last byte to be punched.

PRINT/PUNCH Example

(Underline indicates user input)

```
M A002
*A002 FF 00
*A003 E8 00
*A004 EA 00
*A005 0F 10
*A006 A0 -
*P
S113000002020202039EAA02023F86FC8810CC7B
S1040010B535
```

MEMORY CHANGE

Memory is changed in a three step sequence: 1 opening the location which displays the contents, 2 changing the location and, 3 closing the location. An open location is one whose contents MIKBUG has printed on the teletype for examination, and whose contents are available for change. A closed location is one whose contents are no longer available for change. Any 8 bit byte may be opened. The contents of an open location may be changed by typing a space followed by two hex characters (new contents). MIKBUG will then open the next location. To end the sequence and close a location without any change a space should be typed followed by a carriage return. Once a location has been opened the next location may be opened by typing any character except a space. To open a location respond to the asterick with an "M". MIKBUG will then type a space. Next type the hex address of the location to be opened. It must be 4 hex characters.

```
*M 1000
1000 A5 FC
1001
```

In the example above location 1000 now contains FC and location 1001 is now open. Example: Memory Change

```
*M 0000
*0000 02 .
*0001 02 .
*0002 02 .
*0003 02 .
*0004 02 .
*0005 02 .
*0006 96 -
*
```

In the above example location "0000" was opened followed by 0001-0005. No location or data was changed. If memory change does not cause the contents of a register to change a question mark will be printed.

Display contents of Target Registers

The target program registers can be displayed with the R command. After the registers have been displayed they may be changed by using memory change. The stack pointer points to the first unused location on the stack. The next location contains the condition codes (CC), followed by B accumulator (B), A accumulator (A), Index Register (X), Program counter (P) and stack pointer (S). The registers are stored on the user's stack.

Example Display Registers:

```
    CC  B  A   X    P    S
*R A6 EB DF 7FEC FFD7 A042
*M A042
*A042 01 _
*A043 A6 .   CC
*A044 EB .   B
*A045 DF .   A
*A046 7F .   X
*A047 EC .   X
*A048 FF .   P
*A049 D7 _   P
*
```

In the above example display registers was used to print out the Registers. Memory change was then used to display the registers on the stack. The sequence was ended by typing space and carriage return.

GO TO TARGET PROGRAM

The go to target program function (G) causes the target program registers to be loaded from the user's stack and execution of the program starting at the address pointed to by the program counter on the stack.

The target program will free run until:

1. A wait (WAI) instruction is encountered where the program will wait for a non-maskable interrupt or an interrupt request.

2. A Software interrupt (SWI) is encountered at which time control will jump back into MIKBUG. The user's register will be stored on the user's stack and they will also be displayed by MIKBUG. Therefore SWI along with memory change may be used to set breakpoints.

3. The target program blows (recovery is by RESTART push-button.

Example Go to Target Program

```
*R A6 EB DF 7FEC DFD7 A042
*M A042
*A042 01 .
*A043 A6  00
*A044 EB  00
*A045 DF  00
*A046 7F  00
*A047 EC  00
*A048 DF  00
*A049 D7  00
*A04A FE
*L
S113000000202020202086AA02023F86FC9810CC7B
S1040010B635
S9
*G C8 00 AA 0000 000A A042
*
```

In the example we first set up the registers with memory change. Then we load the program. Then go to target program and return at the software interrupt location 000A.

Interrupt Request

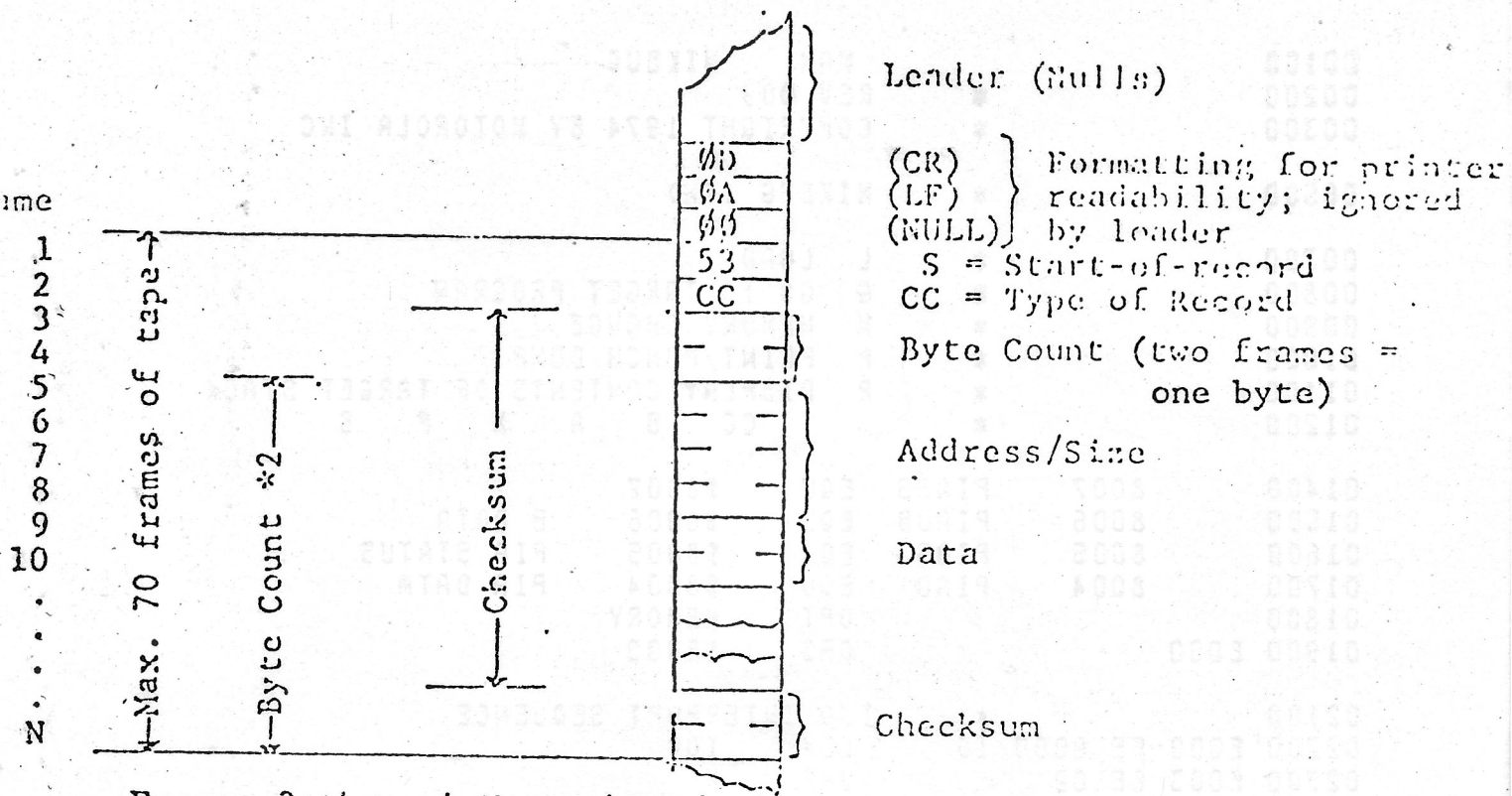
The interrupt request line when pulled will cause a request for interrupt to be input to the MPU. If interrupts are not masked the MPU will jump to the interrupt service routine indirectly through location A000. This is accomplished in MIKBUG by loading the index register with the contents of A000 and then jumping to that address.

Non-Maskable Interrupt

When this interrupt occurs all the registers will be stored on the users stack. The MIU will then wait for the restart button to be pushed. After this has occurred the user may examine the registers that have been saved on the stack.

Allowable Characters

At anytime during the execution of MIKBUG only hex characters, control characters P, M, L, G, R, space, and carriage return are allowed. If an invalid character is input, it will be echoed back to the terminal and the current function will be terminated.



Frames 3 through N are hexadecimal digits (in 7-bit ASCII) which are converted to BCD. Two BCD digits are combined to make one 8-bit byte.

The checksum is the one's complement of the summation of 8-bit bytes.

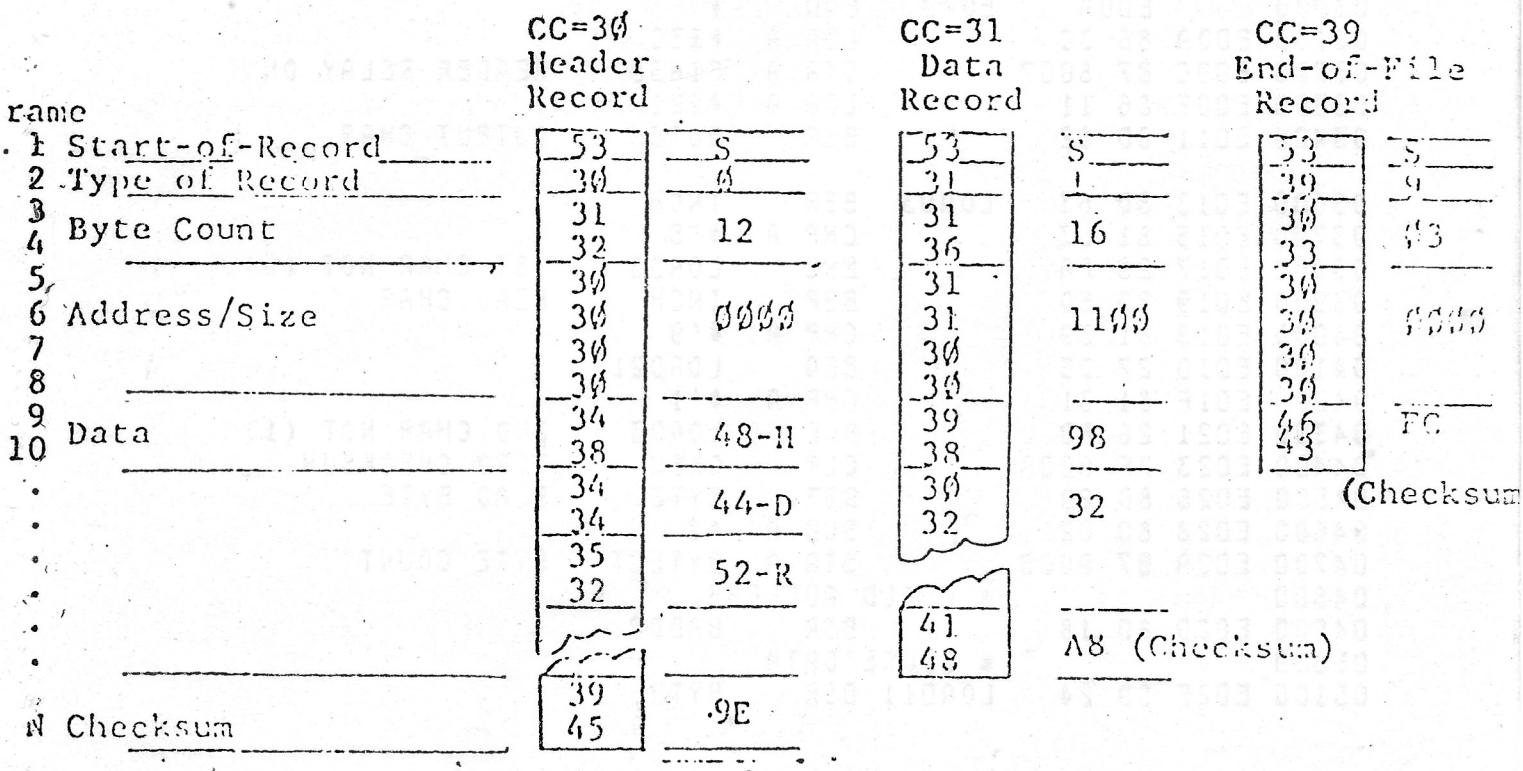


Figure 3.1 Paper Tape Format

3.22 MIKBUG REV. 9 PROGRAM LISTING

```
00100                          NAM    MIKBUG
00200                  *       REV 009
00300                  *       COPYRIGHT 1974 BY MOTOROLA INC

00500                  *       MIKBUG (TM)

00700                  *       L  LOAD
00800                  *       G  GO TO TARGET PROGRAM
00900                  *       M  MEMORY CHANGE
01000                  *       P  PRINT/PUNCH DUMP
01100                  *       R  DISPLAY CONTENTS OF TARGET STACK
01200                  *             CC   B   A   X   P   S

01400        8007      PIASB   EQU    $8007
01500        8006      PIADB   EQU    $8006     B DATA
01600        8005      PIAS    EQU    $8005     PIA STATUS
01700        8004      PIAD    EQU    $8004     PIA DATA
01800                          OPT    MEMORY
01900 E000                     ORG    $E000

02100                  *       I/O INTERRUPT SEQUENCE
02200 E000 FE A000     IO      LDX    IOV
02300 E003 6E 00               JMP    X

  500                  * NMI SEQUENCE
02600 E005 FE A006     POWDWN  LDX    NIO       GET NMI VECTOR
02700 E008 6E 00               JMP    X


03000        E00A      LOAD    EQU    *
03100 E00A 86 3C               LDA A  #$3C
03200 E00C B7 8007             STA A  PIASB     READER RELAY ON
03300 E00F 86 11               LDA A  #@21
03400 E011 8D 62               BSR    OUTCH     OUTPUT CHAR

03600 E013 8D 63       LOAD3   BSR    INCH
03700 E015 81 53               CMP A  #'S
03800 E017 26 FA               BNE    LOAD3     1ST CHAR NOT (S)
03900 E019 8D 5D               BSR    INCH      READ CHAR
04000 E01B 81 39               CMP A  #'9
04100 E01D 27 25               BEQ    LOAD21
04200 E01F 81 31               CMP A  #'1
04300 E021 26 F0               BNE    LOAD3     2ND CHAR NOT (1)
04400 E023 7F A00A             CLR    CKSM      ZERO CHECKSUM
04500 E026 8D 2D               BSR    BYTE      READ BYTE
04600 E028 80 02               SUB A  #2
04700 E02A B7 A00B             STA A  BYTECT    BYTE COUNT
04800                  * BUILD ADDRESS
04900 E02D 8D 18               BSR    BADDR
05000                  * STORE DATA
05100 E02F 8D 24       LOAD11  BSR    BYTE
```

MIKBUG REV. 9 PROGRAM LISTING (continued)

```
05200 E031 7A A00B          DEC     BYTECT
05300 E034 27 05            BEQ     LOAD15    ZERO BYTE COUNT
05400 E036 A7 00            STA A   X         STORE DATA
05500 E038 08               INX
05600 E039 20 F4            BRA     LOAD11

05800 E03B 7C A00A LOAD15   INC     CKSM
05900 E03E 27 D3            BEQ     LOAD3
06000 E040 86 3F   LOAD19   LDA A   #'?       PRINT QUESTION MARK
06100 E042 8D 31            BSR     OUTCH
06200      E044    LOAD21   EQU     *
06300 E044 7E E0E3 C1       JMP     CONTRL


06500                  * BUILD ADDRESS
06600 E047 8D 0C    BADDR  BSR     BYTE      READ 2 FRAMES
06700 E049 B7 A00C         STA A   XHI
06800 E04C 8D 07           BSR     BYTE
06900 E04E B7 A00D         STA A   XLOW
07000 E051 FE A00C         LDX     XHI       (X) ADDRESS WE BUILT
07100 E054 39              RTS


07300                  * INPUT BYTE (TWO FRAMES)
07400 E055 8D 53    BYTE   BSR     INHEX     GET HEX CHAR
07500 E057 48              ASL A
07600 E058 48              ASL A
07700 E059 48              ASL A
07800 E05A 48              ASL A
07900 E05B 16              TAB
08000 E05C 8D 4C           BSR     INHEX
08100 E05E 1B              ABA
08200 E05F 16              TAB
08300 E060 FB A00A         ADD B   CKSM
08400 E063 F7 A00A         STA B   CKSM
08500 E066 39              RTS

08700 E067 44       OUTHL  LSR A             OUT HEX LEFT BCD DIGIT
08800 E068 44              LSR A
08900 E069 44              LSR A
09000 E06A 44              LSR A


09300 E06B 84 0F    OUTHR  AND A   #$F        OUT HEX RIGHT BCD DIGIT
09400 E06D 8B 30           ADD A   #$30
09500 E06F 81 39           CMP A   #$39
09600 E071 23 02           BLS     OUTCH
09700 E073 8B 07           ADD A   #$7

09900                  * OUTPUT ONE CHAR
10000 E075 7E E1D1 OUTCH   JMP     OUTEEE
10100 E078 7E E1AC INCH    JMP     INEEE
```

MIKBUG REV. 9 PROGRAM LISTING (continued)

```
10200                    * PRINT DATA POINTED AT BY X-REG
10300 E07B 8D F8   PDATA2 BSR    OUTCH
10400 E07D 08             INX
10500 E07E A6 00   PDATA1 LDA A  X
10600 E080 81 04          CMP A  #4
10700 E082 26 F7          BNE    PDATA2
10800 E084 39             RTS              STOP ON EOT

11000                    * CHANGE MEMORY (M AAAA DD NN)
11100 E085 8D C0   CHANGE BSR    BADDR     BUILD ADDRESS
11200 E087 CE E19D CHA51  LDX    #MCL
11300 E08A 8D F2          BSR    PDATA1    C/R L/F
11400 E08C CE A00C        LDX    #XHI
11500 E08F 8D 37          BSR    OUT4HS    PRINT ADDRESS
11600 E091 FE A00C        LDX    XHI
11700 E094 8D 34          BSR    OUT2HS    PRINT DATA (OLD)
11800 E096 FF A00C        STX    XHI       SAVE DATA ADDRESS
11900 E099 8D DD          BSR    INCH      INPUT ONE CHAR
12000 E09B 81 20          CMP A  #$20
12100 E09D 26 E8          BNE    CHA51     NOT SPACE
12200 E09F 8D B4          BSR    BYTE      INPUT NEW DATA
12300 E0A1 09             DEX
12400 E0A2 A7 00          STA A  X         CHANGE MEMORY
12500 E0A4 A1 00          CMP A  X
12600 E0A6 27 DF          BEQ    CHA51     DID CHANGE
12700 E0A8 20 96          BRA    LOAD19    NOT CHANGED



13100                    * INPUT HEX CHAR
13200 E0AA 8D CC   INHEX  BSR    INCH
13300 E0AC 80 30          SUB A  #$30
13400 E0AE 2B 94          BMI    C1        NOT HEX
13500 E0B0 81 09          CMP A  #$09
13600 E0B2 2F 0A          BLE    IN1HG
13700 E0B4 81 11          CMP A  #$11
13800 E0B6 2B 8C          BMI    C1        NOT HEX
13900 E0B8 81 16          CMP A  #$16
14000 E0BA 2E 88          BGT    C1        NOT HEX
14100 E0BC 80 07          SUB A  #7
14200 E0BE 39      IN1HG  RTS

14500 E0BF A6 00   OUT2H  LDA A  0,X       OUTPUT 2 HEX CHAR
14600 E0C1 8D A4   OUT2HA BSR    OUTHL     OUT LEFT HEX CHAR
14700 E0C3 A6 00          LDA A  0,X
14800 E0C5 08             INX
14900 E0C6 20 A3          BRA    OUTHR     OUTPUT RIGHT HEX CHAR AND R

15100 E0C8 8D F5   OUT4HS BSR    OUT2H     OUTPUT 4 HEX CHAR + SPACE
15200 E0CA 8D F3   OUT2HS BSR    OUT2H     OUTPUT 2 HEX CHAR + SPACE
```

MIKBUG REV. 9 PROGRAM LISTING (continued)

```
15300 E0CC 86 20   OUTS   LDA A  #$20     SPACE
15400 E0CE 20 A5          BRA   OUTCH     (BSR & RTS)

15600                * ENTER POWER ON SEQUENCE
15700       E0D0     START  EQU   *
15800 E0D0 8E A042        LDS   #STACK
15900 E0D3 BF A008        STS   SP        INZ TARGET'S STACK PNTR
16000                * INZ PIA
16100 E0D6 CE 8004        LDX   #PIAD     (X) POINTER TO.DEVICE PIA
16200 E0D9 6C 00          INC   0,X       SET DATA DIR PIAD
16300 E0DB 86 07          LDA A #$7
16400 E0DD A7 01          STA A 1,X       INIT CON PIAS
16500 E0DF 6C 00          INC   0,X       MARK COM LINE
16600 E0E1 A7 02          STA A 2,X       SET DATA DIR PIADB
16700 E0E3 86 34   CONTRL LDA A #$34
16800 E0E5 B7 8007        STA A PIASB     SET CONTROL PIASB TURN READ
16900 E0E8 B7 8006        STA A PIADB     SET TIMER INTERVAL
17000 E0EB 8E A042        LDS   #STACK    SET CONTRL STACK POINTER
17100 E0EE CE E19C        LDX   #MCLOFF

17300 E0F1 8D 8B          BSR   PDATA1    PRINT DATA STRING

17500 E0F3 8D 83          BSR   INCH      READ CHARACTER
17600 E0F5 16             TAB
17700 E0F6 8D D4          BSR   OUTS      PRINT SPACE
17800 E0F8 C1 4C          CMP B #'L
17900 E0FA 26 03          BNE   *+5
18000 E0FC 7E E00A        JMP   LOAD
18100 E0FF C1 4D          CMP B #'M
18200 E101 27 82          BEQ   CHANGE
18300 E103 C1 52          CMP B #'R
18400 E105 27 18          BEQ   PRINT     STACK
18500 E107 C1 50          CMP B #'P
18600 E109 27 32          BEQ   PUNCH     PRINT/PUNCH
18700 E10B C1 47          CMP B #'G
18800 E10D 26 D4          BNE   CONTRL
18900 E10F BE A008        LDS   SP        RESTORE PGM'S STACK PTR
19000 E112 3B             RTI             GO

19200                * ENTER FROM SOFTWARE INTERRUPT
19300       E113     SFE    EQU   *
19400 E113 BF A008        STS   SP        SAVE TARGET'S STACK POINTER
19500                * DECREMENT P-COUNTER
19600 E116 30             TSX
19700 E117 6D 06          TST   6,X
19800 E119 26 02          BNE   *+4
19900 E11B 6A 05          DEC   5,X
20000 E11D 6A 06          DEC   6,X

20200                * PRINT CONTENTS OF STACK
20300 E11F FE A008 PRINT  LDX   SP
20400 E122 08             INX
```

MIKBUG REV. 9 PROGRAM LISTING (continued)

```
20500 E123 8D A5          BSR    OUT2HS   CONDITION CODES
20600 E125 8D A3          BSR    OUT2HS   ACC-B
20700 E127 8D A1          BSR    OUT2HS   ACC-A
20800 E129 8D 9D          BSR    OUT4HS   X-REG
20900 E12B 8D 9B          BSR    OUT4HS   P-COUNTER
21000 E12D CE A008        LDX    #SP
21100 E130 8D 96          BSR    OUT4HS   STACK POINTER
21200 E132 20 AF    C2    BRA    CONTRL


21400               *     PUNCH DUMP
21500               *     PUNCH FROM BEGINING ADDRESS (BEGA) THRU ENDI
21600               *     ADDRESS (ENDA)
21700               *
-------------------------------------------------------------------------
      E134 0D       MTAPE1 FCB   $D,$A,0,0,0,0,'S,'1,4   PUNCH FORMAT
      E135 0A
      E136 00
      E137 00
      E138 00
      E139 00
      E13A 53
      E13B 31
      E13C 04

22100       E13D    PUNCH  EQU   *

22300 E13D 86 12          LDA A  #$12      TURN TTY PUNCH ON
22400 E13F BD E075        JSR    OUTCH     OUT CHAR

22600 E142 FE A002        LDX    BEGA
22700 E145 FF A00F        STX    TW        TEMP BEGINING ADDRESS
22800 E148 B6 A005  PUN11 LDA A  ENDA+1
22900 E14B B0 A010        SUB A  TW+1
23000 E14E F6 A004        LDA B  ENDA
23100 E151 F2 A00F        SBC B  TW
23200 E154 26 04          BNE    PUN22
23300 E156 81 10          CMP A  #16
23400 E158 25 02          BCS    PUN23
23500 E15A 86 0F    PUN22 LDA A  #15
23600 E15C 8B 04    PUN23 ADD A  #4
23700 E15E B7 A011        STA A  NCONT     FRAME COUNT THIS RECORD
23800 E161 80 03          SUB A  #3
23900 E163 B7 A00E        STA A  TEMP      BYTE COUNT THIS RECORD
24000               *     PUNCH C/R,L/F,NULL,S,1
24100 E166 CE E134        LDX    #MTAPE1
24200 E169 BD E07E        JSR    PDATA1
24300 E16C 5F             CLR B            ZERO CHECKSUM
24400               *     PUNCH FRAME COUNT
24500 E16D CE A011        LDX    #NCONT
24600 E170 8D 25          BSR    PUNT2     PUNCH 2 HEX CHAR
24700               *     PUNCH ADDRESS
```

MIKBUG REV. 9 PROGRAM LISTING (continued)

```
24800 E172 CE A00F          LDX     #TW
24900 E175 8D 20            BSR     PUNT2
25000 E177 8D 1E            BSR     PUNT2
25100                *      PUNCH DATA
25200 E179 FE A00F          LDX     TW
25300 E17C 8D 19    PUN32   BSR     PUNT2       PUNCH ONE BYTE (2 FRAMES)
25400 E17E 7A A00E          DEC     TEMP        DEC BYTE COUNT
25500 E181 26 F9            BNE     PUN32
25600 E183 FF A00F          STX     TW
25700 E186 53               COM B
25800 E187 37               PSH B
25900 E188 30               TSX
26000 E189 8D 0C            BSR     PUNT2       PUNCH CHECKSUM
26100 E18B 33               PUL B               RESTORE STACK
26200 E18C FE A00F          LDX     TW
26300 E18F 09               DEX
26400 E190 BC A004          CPX     ENDA
26500 E193 26 B3            BNE     PUN11
26600 E195 20 9B            BRA     C2          JMP TO CONTRL


26800                *      PUNCH 2 HEX CHAR, UPDATE CHECKSUM
26900 E197 EB 00    PUNT2   ADD B   0,X         UPDATE CHECKSUM
27000 E199 7E E0BF          JMP     OUT2H       OUTPUT TWO HEX CHAR AND RTS


27020 E19C 13       MCLOFF  FCB     $13         READER OFF
27100 E19D 00       MCL     FCB     $D,$A,$14,0,0,0,'*,4    C/R,L/F,PUNCH
      E19E 0A
      E19F 14
      E1A0 00
      E1A1 00
      E1A2 00
      E1A3 2A
      E1A4 04
27200                *
27300 E1A5 FF A012  SAV     STX     XTEMP
27400 E1A8 CE 8004          LDX     #PIAD
27500 E1AB 39               RTS
27600                * INPUT ONE CHAR INTO A-REGISTER
27700 E1AC 37       INEEE   PSH B               SAVE ACC-B
27800 E1AD 8D F6            BSR     SAV         SAV XR
27900 E1AF A6 00    IN1     LDA A   0,X         LOOK FOR START BIT
28000 E1B1 2B FC            BMI     IN1
28100 E1B3 6F 02            CLR     2,X         SET COUNTER FOR HALF BIT TI
28200 E1B5 8D 3C            BSR     DE          START TIMER
28300 E1B7 8D 36            BSR     DEL         DELAY HALF BIT TIME
28400 E1B9 C6 04            LDA B   #4          SET DEL FOR FULL BIT TIME
28500 E1BB E7 02            STA B   2,X
28600 E1BD 58               ASL B               SET UP CNTR WITH 8
```

MIKBUG REV. 9 PROGRAM LISTING (continued)

```
28700 E1BE 8D 2F    IN3     BSR     DEL         WAIT ONE CHAR TIME
28800 E1C0 0D               SEC                 MARK COM LINE
28900 E1C1 69 00            ROL     0,X         GET BIT INTO CFF
29000 E1C3 46               ROR A               CFF TO AR
29100 E1C4 5A               DEC B
29200 E1C5 26 F7            BNE     IN3
29300 E1C7 8D 26            BSR     DEL         WAIT FOR STOP BIT
29400 E1C9 84 7F            AND A   #$7F        RESET PARITY BIT
29500 E1CB 81 7F            CMP A   #$7F
29600 E1CD 27 E0            BEQ     IN1         IF RUBOUT,GET NEXT CHAR
29700 E1CF 20 12            BRA     IOUT2       GO RESTORE REG


29900                   * OUTPUT ONE CHAR
30000 E1D1 37           OUTEEE PSH B            SAV BR
30100 E1D2 8D D1            BSR     SAV         SAV XR
30200 E1D4 C6 0A        IOUT    LDA B   #$A      SET UP COUNTER
30300 E1D6 6A 00            DEC     0,X         SET START BIT
30400 E1D8 8D 19            BSR     DE          START TIMER
30500                   *
30600 E1DA 8D 13        OUT1    BSR     DEL         DELAY ONE BIT TIME
30700 E1DC A7 00            STA A   0,X         PUT OUT ONE DATA BIT
30800 E1DE 0D               SEC                 SET CARRY BIT
30900 E1DF 46               ROR A               SHIFT IN NEXT BIT
31000 E1E0 5A               DEC B               DECREMENT COUNTER
31100 E1E1 26 F7            BNE     OUT1        TEST FOR 0
31200 E1E3 E6 02        IOUT2   LDA B   2,X      TEST FOR STOP BITS
31300 E1E5 58               ASL B               SHIFT BIT TO SIGN
31400 E1E6 2A 02            BPL     IOS         BRANCH FOR 1 STOP BIT
31500 E1E8 8D 05            BSR     DEL         DELAY FOR STOP BITS
31600 E1EA FE A012  IOS     LDX     XTEMP       RES XR
31700 E1ED 33               PUL B               RESTORE BR
31800 E1EE 39               RTS
31900                   *
32000 E1EF 6D 02        DEL     TST     2,X     IS TIME UP
32100 E1F1 2A FC            BPL     DEL
32200 E1F3 6C 02        DE      INC     2,X     RESET TIMER
32300 E1F5 6A 02            DEC     2,X
32400 E1F7 39               RTS

32600 E1F8 E000             FDB     IO
32700 E1FA E113             FDB     SFE
32800 E1FC E005             FDB     POWDWN
32900 E1FE E000             FDB     START
33000 A000                  ORG     $A000
33100 A000 0002     IOV     RMB     2           IO INTERRUPT POINTER
33200 A002 0002     BEGA    RMB     2           BEGINING ADDR PRINT/PUNCH
33300 A004 0002     ENDA    RMB     2           ENDING ADDR PRINT/PUNCH
33400 A006 0002     NIO     RMB     2           NMI INTERRUPT POINTER
33500 A008 0001     SP      RMB     1           S-HIGH
33600 A009 0001             RMB     1           S-LOW
33700 A00A 0001     CKSM    RMB     1           CHECKSUM
```

MIKBUG REV. 9 PROGRAM LISTING (continued)

```
33800 A00B 0001    BYTECT RMB    1        BYTE COUNT
33900 A00C 0001    XHI    RMB    1        XREG HIGH
34000 A00D 0001    XLOW   RMB    1        XREG LOW
34100 A00E 0001    TEMP   RMB    1        CHAR COUNT (INADD)
34200 A00F 0002    TW     RMB    2        TEMP/
34300 A011 0001    MCONT  RMB    1        TEMP
34400 A012 0002    XTEMP  RMB    2        X-REG TEMP STORAGE
34500 A014 002E           RMB    46
34600 A042 0001    STACK  RMB    1        STACK POINTER


35000                      END

SYMBOL TABLE

PIA5B  8007  PIADB  8006  PIA5   8005  PIAD   8004  IO     E000
POWDWN E005  LOAD   E00A  LOAD3  E013  LOAD11 E02F  LOAD15 E03B
LOAD19 E040  LOAD21 E044  C1     E044  BADDR  E047  BYTE   E055
OUTHL  E067  OUTHR  E06B  OUTCH  E075  INCH   E078  PDATA2 E07B
PDATA1 E07E  CHANGE E085  CHA51  E087  INHEX  E0AA  IN1HG  E0BE
OUT2H  E0BF  OUT2HA E0C1  OUT4HS E0C8  OUT2HS E0CA  OUTS   E0CC
START  E0D0  CONTRL E0E3  SFE    E113  PRINT  E11F  C2     E132
MTAPE1 E134  PUNCH  E13D  PUN11  E148  PUN22  E15A  PUN23  E15C
PUN32  E17C  PUNT2  E197  MCLOFF E19C  MCL    E19D  SAV    E1A5
INEEE  E1AC  IN1    E1AF  IN3    E1BE  OUTEEE E1D1  IOUT   E1D4
OUT1   E1DA  IOUT2  E1E3  IOS    E1EA  DEL    E1EF  DE     E1F3
IOV    A000  BEGA   A002  ENDA   A004  NIO    A006  SP     A008
CKSM   A00A  BYTECT A00B  XHI    A00C  XLOW   A00D  TEMP   A00E
TW     A00F  MCONT  A011  XTEMP  A012  STACK  A042
```

4.0 SYSTEM OPERATION

The M6800 Evaluation System as shown in Figure 1.1 consists of 1 MPU, 6 RAM's, 1 ROM, 2 PIA's, TTY/RS232 interface, Bus Extenders and 512 bytes of PROM. The memory address map as shown in Figure 4.1 shows how these parts are layed out in the system.
In the earlier MES versions the XC6816 ROM with the MIKBUG control program was not available. Therefore the MIKBUG program has been placed in PROM's located at the same address. The part of the system to the left of the bus extenders make up a minimum system and with the ROM it will operate without the bus extenders or PROM's.

The RAM's in hex location 0000 through 027F may be used by the user to program the MPU and PIA's to test and evaluate these devices. The RAM at location A000 is used by the MIKBUG control program for temporary storage and stack area.

The ROM/PROM's are located at location E000 and it is assumed that there are no devices in the system at a higher address.

The PIA's are located at 8004 and 8008. The PIA at location 8004 is used by the MIKBUG control program to interface the TTY and RS232 device. It is also used with the Programmable Timer (MC14536) for the Serial Interface Timing. The PIA at location 8008 is available for use by the user for evaluation of PIA and a cable is provided so that he may connect this PIA to a peripheral if he so desires using the P2 connector.

The bus drivers may be used to expand the system to 32K bytes of memory. To do this it would be necessary to remove the RAM's at locations hex 0000 to 027F. The reason for this is that this address decoding assumes that there is no memory between 027F and 8000. With these RAM's removed it is now necessary to control the data bus drivers so they know when to turn on. This is done by using the enable line with the proper strapping option to turn on the drivers at the proper times. This address information must be decoded external to the PC card.

For further information on the N-Channel parts see the Microcomputer System Reference Handbook.

M6800 Evaluation System Address Map

	A_{15}	A_{14}	A_{13}	A_{12}	A_{11}	A_{10}	A_9	A_8	A_7	A_6	A_5	A_4	A_3	A_2	A_1	A_0	Hex
RAM	0	0	0	0	0	0	0	0	0	X	X	X	X	X	X	X	0000
	0	0	0	0	0	0	0	0	1	X	X	X	X	X	X	X	0080
	0	0	0	0	0	0	0	1	0	X	X	X	X	X	X	X	0100
$64 0_{10}$	0	0	0	0	0	0	0	1	1	X	X	X	X	X	X	X	0180
	0	0	0	0	0	0	1	0	0	X	X	X	X	X	X	X	0200
																	-027F
PIA	1	0	0	0	0	0	0	0	0	0	0	0	0	1	X	X	8004
PIA	1	0	0	0	0	0	0	0	0	0	0	0	1	0	X	X	8008
RAM	1	0	1	0	0	0	0	0	0	X	X	X	X	X	X	X	A000
127_{10}																	A07F
ROM	1	1	1	0	0	0	0	X	X	X	X	X	X	X	X	X	E000

X indicates a variable Address

A000 → A038 USED BY MIKBUG
A042 → A049 " " "

Figure 4.1

Products Division

TITLE	MODEL NO.	ASSY NO.	WORK ORDER	BILL OF MATERIAL NO.
M6800 EVALUATION - CARD				PLMEC6800-1(A)A

SYMBOL	ITEM	QTY	NAME AND DESCRIPTION	VENDOR CODE	PART OR STOCK NO.
	1	1	EVALUATION CARD ASSY		
	2				
	3				
	4				
	5	1	PRINTED WIRING BOARD	DEA	MEC6800-1(A) P
P2	6	1	CONNECTOR, 40 PIN, P2	3M	3432-1002
P3	7	1	CONNECTOR, 16 PIN, P3	TI	C931602
	8	2	SCREW - 2-56 x 3/8 LONG		
	9	2	NUT - 2-56		
	10	1	BUTTON, SNAP-ON	CS	P281R
	11	4	SOCKET, LOW PROFILE, 16 PIN	TI	C931602
	12	7	SOCKET, LOW PROFILE, 24 PIN	TI	C934-002
	13	3	SOCKET, LOW PROFILE, 40 PIN	TI	C932-402
			REF.		
			MEC6800-1(B)S SCHEMATIC DIAGRAM		
			MEC6800-1(A)A ASSEMBLY		

WRITTEN BY | DATE 7-15-74 | REMARKS | PAGE 1 OF 2

Motorola Semiconductor Products Division

TITLE	MODEL NO.	ASSY NO.	WORK ORDER	BILL OF MATERIAL NO
M6800 EVALUATION - CARD				PLMEC6800-1(E)A

SYMBOL	ITEM	QTY	NAME AND DESCRIPTION	VENDOR CODE	PART OR STOCK NO.
C1		1	CAPACITOR, 100UF, 16VDC,	SPRAG	TE-1162
C2, C3		2	CAPACITOR, 30PF, 500V, DIPPED MICA	ELMEN-CO	DM-10-300J
C4		1	CAPACITOR, 620PF. 500V, DIPPED MICA	ELMEN-CO	DM-19-621J
CE THRU C33		29	CAPACITOR, 0.1UF, 50V, ± 20%	USCC	CY20C104M
CR1		1	DIODE		IN4001
CR2		1	DIODE		IN4001
R26		1	RESISTOR, FXD, 150, 1/2W, 5%		
R1 R3, R6, R7		9	RESISTOR, FXD, 2K, 1/4 W, 5%		
R11, R15, R16, R17, R27					
R4, R5		2	RESISTOR, FXD, 20Ω, 1/4W, 5%		
R8 R12		2	RESISTOR, FXD, 1K, 1/4W, 5%		
R9. R13. R23		3	RESISTOR, VAR, 50K, 1/4W, 5%	SPECTROL	43W503
R10. R14		2	RESISTOR, FXD, 5100, 1/4W, 5%		
R18. R2		2	RESISTOR, FXD, 10K, 1/4W, 5%		
R19		1	RESISTOR, FXD, 91K, 1/4W, 5%		
R20		1	RESISTOR, FXD, 1200, 1/4W, 5%		
R21		1	RESISTOR, FXD, 2200, 1/4W, 5%		
R22		1	RESISTOR, FXD, 510, 1/4W, 5%		
R24		1	RESISTOR, FXD, 820, 1 WATT, 10%		
R25		1	RESISTOR, FXD, 1100, 1 WATT, 10%		

DRAWN BY | DATE 7-[illegible]-[illegible] | REMARKS | PAGE 2 OF 2

TITLE	MODEL NO.	ASSY NO.	WORK ORDER	BILL OF MATERIAL NO.
MG800 EVALUATION CARD				PLANE6SDO-1(3,)

SYMBOL	ITEM	QTY	NAME AND DESCRIPTION	VENDOR CODE	PART OR STOCK NO.
U1		1	INTEGRATED CIRCUIT - NAND GATE	MOT	MC7400P
U2, U3, U4, U5, U6		7	INTEGRATED CIRCUIT - BUS DRVR/RCVR	SIG	8T26
U7, U8					
U9		1	INTEG CKT - 1024 X 8 ROM	MOT	XC6816
U10 U17, U20, U13		4	INTEG CKT - 1024 PROM	HAR	HPROM-1024
U11		1	INTEG CKT - MPU	MOT	XC6800
U12, U16, U19		6	INTEG CKT - RAM	MOT	XC6810
U23, U24, U28					
U14		1	INTEG CKT - 2-INPUT QUAD NAND GATE	MOT	MC3459L
U15		1	INTEG CKT - 8-INPUT NAND GATE	MOT	MC7430
U16		1	INTEG CKT - MULTIVIBRATOR	MOT	MC8602P
U21		1	INTEG CKT - NAND BUFFER	MOT	MC7437P
U22 U31, U32		3	INTEG CKT - ISOLATOR	MOT	4N33
U25, U26		2	INTEG CKT - PIA	MOT	XC6820
U27		1	INTEG CKT - PROGRAMMABLE TIMER	MOT	MC14536AL
U29		1	INTEG CKT - QUAD MDTL LINE RCVR	MOT	MC1489L
U30		1	INTEG CKT - QUAD MDTL LINE DRVR	MOT	MC1488L
S1		1	SWITCH, PUSHBUTTON, LATCHING	CS	E8600

MES POWER SUPPLY

SYMBOL	ITEM	QTY	NAME AND DESCRIPTION	Vendor code	PART OR STOCK NO.
M1	1	1	MAINS PLUG		
F1	2	1	FUSE 315 mA		
TR1	3	1	TRANSFORMER 220 V; 50 Hz		
			15 V~/0,3 A; 15 V~/0,3 A; 9 V~/2,5 A		
G1	4	1	MDA 922-4 BRIDGE RECTIFIER		
G2	5	1	MDA 970-3 BRIDGE RECTIFIER		
RG1	6	1	VOLTAGE REGULATOR +12V		
RG2	7	1	VOLTAGE REGULATOR -12V		
RG3	8	1	VOLTAGE REGULATOR +5V		
R1	9	1	RESISTOR 51 Ω/0,25 W; 1%		
Z1.Z2	10	2	ZENER DIODE 1N4746		
Z3	11	1	" 1N4735		
C1	12	4	CAPACITOR ELECTROLYT. 0,1 µF/35V		
C2	13	2	" " 470 µF/25V		
C3	14	1	" " 10000 µF/16V		

5.2 Connector Pin Assignments

Evaluation Card Assignment Connector P1

Component Side Pin Number	Function	Circuit Side	Function
A	+5VDC	1	+5VDC
B	+5VDC	2	+5VDC
C	+5VDC	3	+5VDC
D	IRQ	4	
E	NMI	5	RESET
F	VMA	6	R/W
H	GND	7	
J	Ø2	8	
K	GND	9	
L		10	VUA
M	-12VDC	11	-12VDC
N	ENABLE	12	
P		13	
R		14	
S		15	
T		16	+12VDC
U		17	
V		18	TSC
W		19	
X		20	
Y		21	
Z		22	
A		23	
B		24	
C		25	
D		26	
E	BA	27	
F	H/G	28	
H	D3	29	D1
J	D7	30	D5
K	D2	31	D0
L	D6	32	D4
M	A14	33	A15
N	A13	34	A12
P	A10	35	A11
R	A9	36	A8
S	A6	37	A7
T	A5	38	A4
U	A2	39	A3
V	A1	40	A0
W	GND (?)	41	GND
X	GND	42	GND
Y	GND	43	GND

Evaluation Card Connector Pin Assignments

Connector P3

Pin Number	Function
1	RS232C Signal Ground
2	RS232C Input
3	RS232C Ouput
4	Not Used
5	Not Used
6	Not Used
7	Not Used
8	Not Used
9	Not Used
10	Reader Control Return
11	Reader Control
12	Serial Common
13	Serial Input
14	Not Used
15	Serial Output
16	Not Used

Evaluation Card Connector Pin Assignments

Connector P2

Pin Number	Function	
1, 2, 7, 8 , 9, 10	GND	
39, 40, 15, 16, 17,	GND	
18, 23, 24, 25, 26,	GND	
31, 32, 33, 34	GND	
21	PA7	
22	PA6	
27	PA5	
28	PA4	
29	PA3	A Peripheral Data Bus
30	PA2	
35	PA1	
37	PA0	
38	CA1	A Peripheral Control
36	CA2	
5	PB7	
6	PB6	
11	PB5	
12	PB4	
13	PB3	B Peripheral Data Bus
14	PB2	
19	PB1	
20	PB0	
4	CB1	B Peripheral Control
3	CB2	

5.3 Assembly Drawing

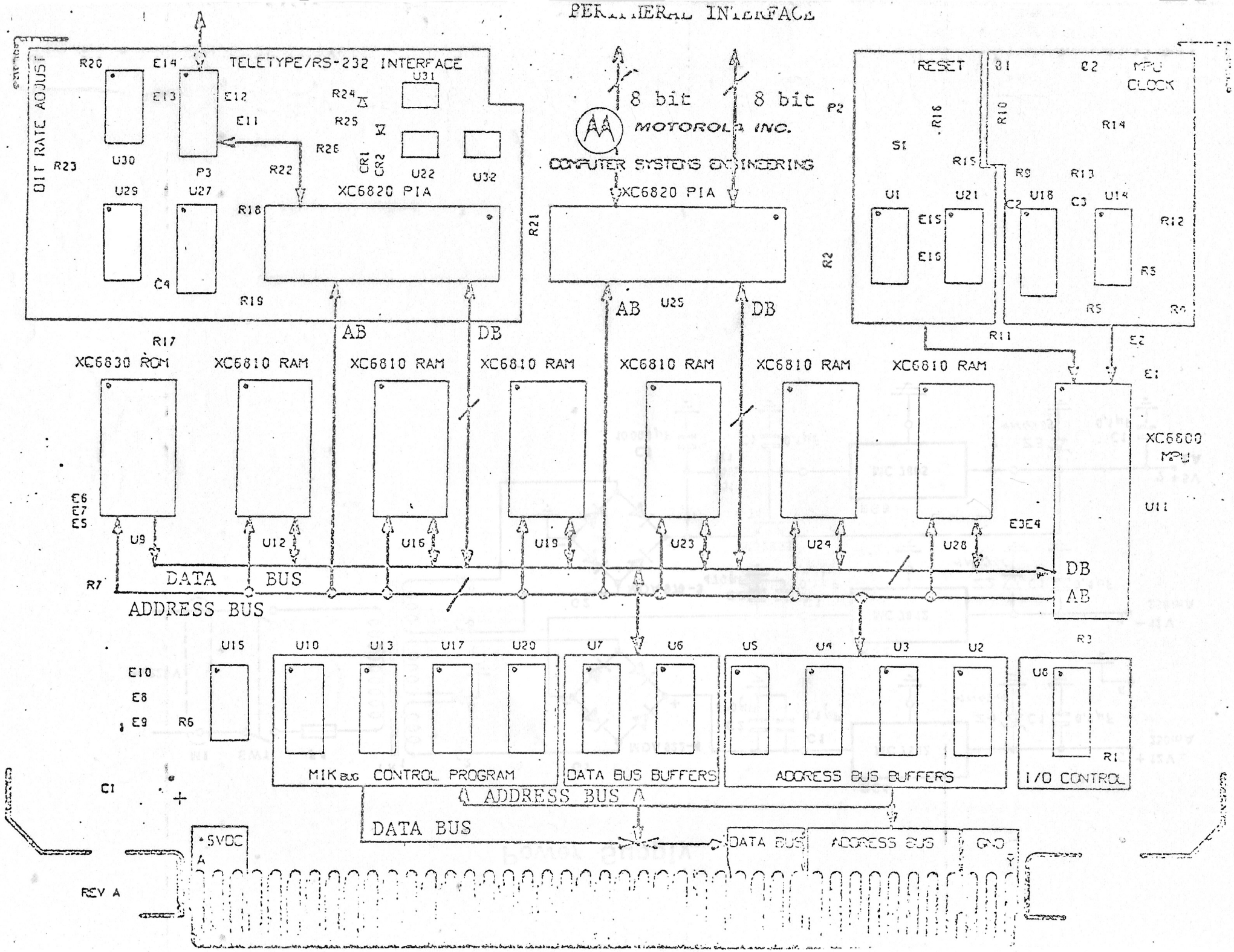
See page 35.

5.4 Schematic Drawings

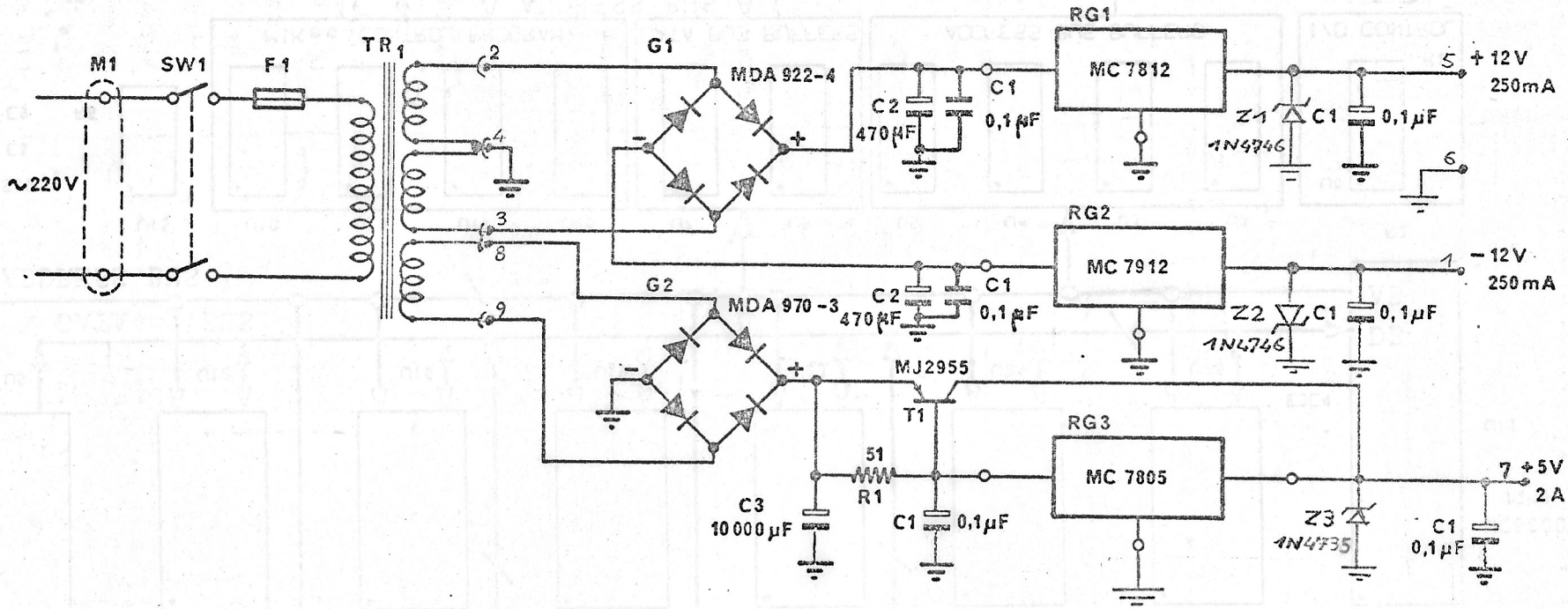
5.4.1 The detailed schematic diagram of the Evaluation Card is supplied with the suit-case.

5.4.2 Schematic drawing of power supply

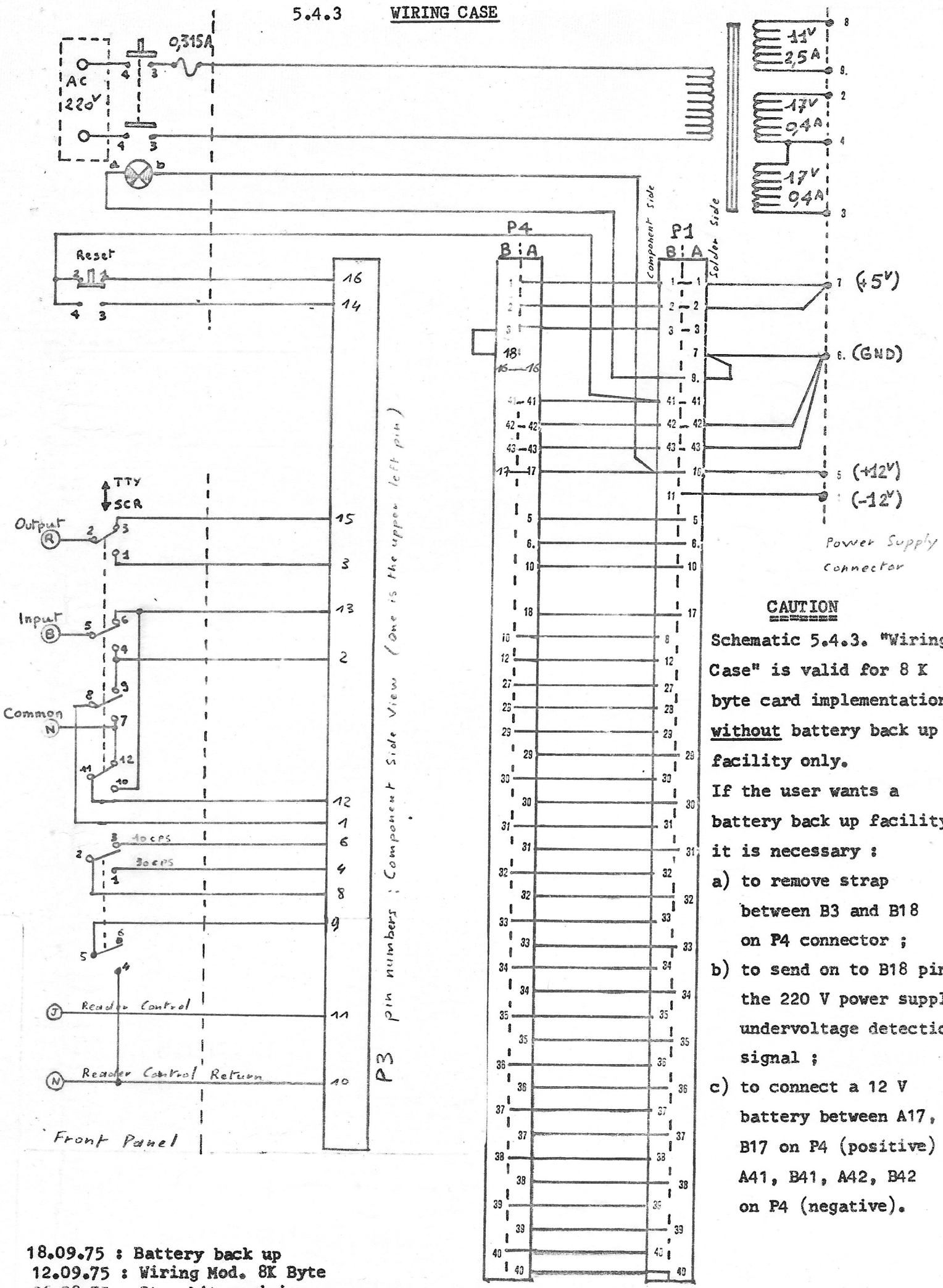
See page 36.


PERIPHERAL INTERFACE
TELETYPE/RS-232 INTERFACE
BIT RATE ADJUST
8 bit
8 bit
MOTOROLA INC.
COMPUTER SYSTEMS ENGINEERING
XC6820 PIA
XC6820 PIA
RESET
MPU CLOCK
AB
DB
AB
DB
XC6830 ROM
XC6810 RAM
XC6810 RAM
XC6810 RAM
XC6810 RAM
XC6810 RAM
XC6810 RAM
XC6800 MPU
DATA BUS
ADDRESS BUS
DB
AB
MIKbug CONTROL PROGRAM
DATA BUS BUFFERS
ADDRESS BUS BUFFERS
I/O CONTROL
ADDRESS BUS
DATA BUS
+5VDC
DATA BUS
ADDRESS BUS
GND
REV A

Power Supply



5.4.3 WIRING CASE



CAUTION

Schematic 5.4.3. "Wiring Case" is valid for 8 K byte card implementation <u>without</u> battery back up facility only.

If the user wants a battery back up facility it is necessary :

a) to remove strap between B3 and B18 on P4 connector ;

b) to send on to B18 pin the 220 V power supply undervoltage detection signal ;

c) to connect a 12 V battery between A17, B17 on P4 (positive) A41, B41, A42, B42 on P4 (negative).

18.09.75 : Battery back up
12.09.75 : Wiring Mod. 8K Byte
26.08.75 : Stop bit revision