FM-1

FUNCTIONAL DESCRIPTION

The FM-1 board provides permanent non-volatile storage of software for instant execution on power-up. It contains 3K of 1702 type PROM memory and 1K of 2102 type RAM memory, thus the FM-1 occupies 4K of contiguous memory space in the computer. Since any software provided cannot be relocated, the address is fixed at E000 hex.

If the FM-1 is loaded with a system control program (monitor), a front panel is not generally required, since the monitor could perform all of the functions normally fulfilled by the front panel.

If a system configuration does not include a front panel, the FM-1 can be selected to generate the MEM WRITE and EXT CLEAR signals normally associated with the front panel. If desired, these signals can be disabled to avoid conflict when these signals are already present.

A double-entry hardware jump is provided for immediate execution of the stored software. One entry point jumps to E000 hex, which is triggered by a system reset. This could be used for an initialization routine. The other point jumps to E004 hex, which is triggered by a user-defined pin on the backplane. This could serve as a re-entrant point or as an "interrupt" routine.

The FM-1 requires no external memory to function in most situations, since 1024 bytes of on-card RAM is provided. A serial interface is required for a terminal.

A jumper option is available so that the user-defined entry point line may be connected to any of the uncommitted pins on the backplane.

Integrated circuit poewr regulation is provided with high quality tantalum and disc ceramic by-pass capacitors. The board is made on G10-type, 1/16 inch laminate with contact fingers gold-plated over nickel for reliable contact and long life. The remainder of the circuitry is tin/lead-plated for good appearance and reliable solder connections. Plated through-holes eliminate the need for any circuit jumpers.

All circuit options are provided on a 7-position dip switch for convenient and quick change from one system to another.

THEORY OF OPERATION

To enable the FM-1, it must be properly addressed. Address bits A12 through A15 are ANDed by C1. This signal is then NANDed to enable the 8T97 drivers to the data bus, and inverted to enable A1, a 74154 4 to 16 decoder. The latter signal is also used to enable the READY line to the bus.

Address bits A8 through A11 are fed to the 74154 to decode the address of the PROM and RAM array. Outputs 12 through 15 are ORed to form the chip enable for the RAMs, since the RAM array is equivalent to 4 PROMS.

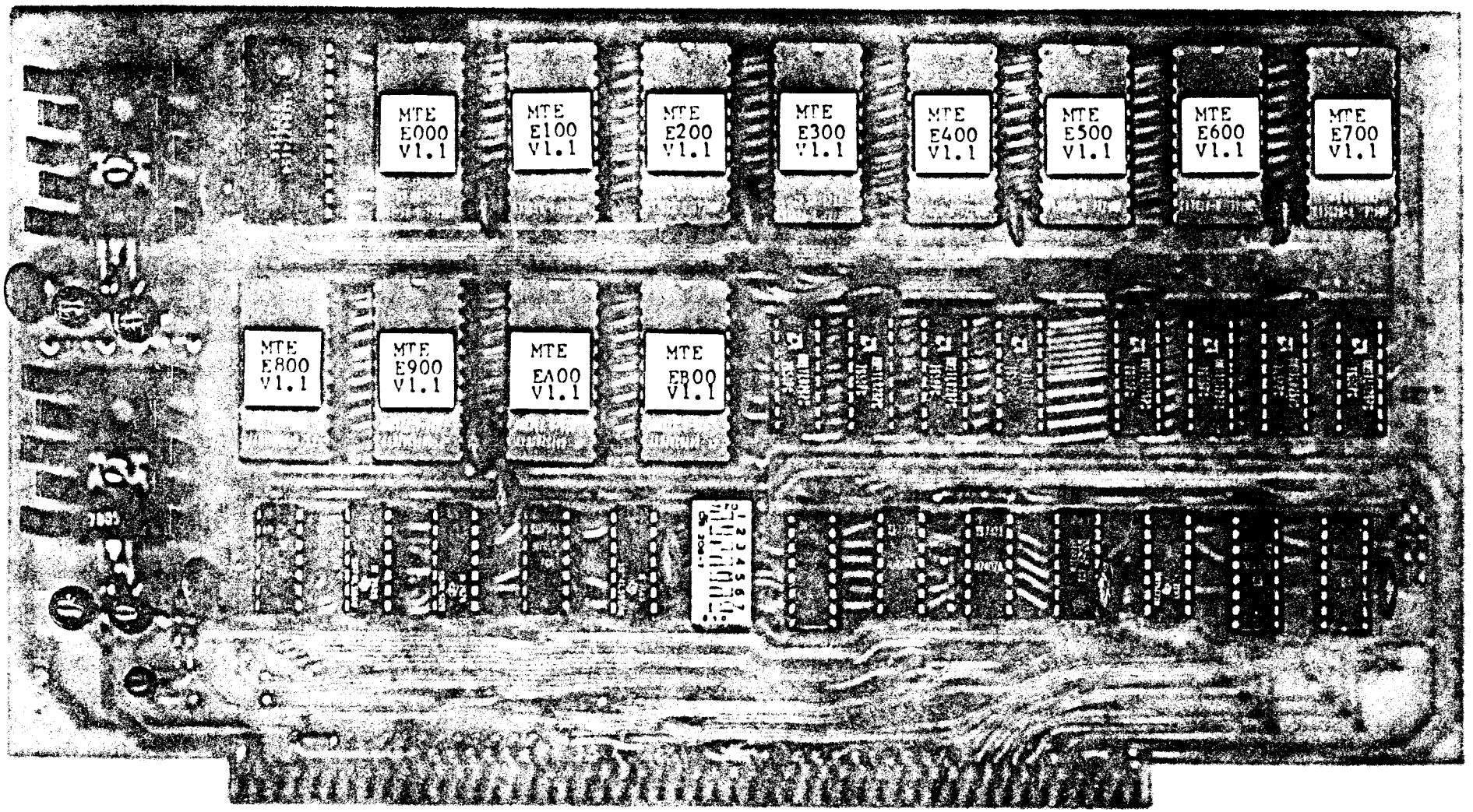
The MEM WRITE signal is formed by NORing the SOUT and PWR signals. This signal is sent to C12, where it may be enabled by closing S7 on the dip switch. The output of NOR gate C3 is inverted directly and fed to the write enable to the RAMs. This is done so that the RAMs get the proper signal whether it is enabled to the bus or not.

The EXT CLEAR signal is accomplished with an open-collector buffer connected to the RESET line. It is enabled by closing S5 on the dip switch.

Since most 1702 proms have access times of more than 500 nanoseconds, one or more wait cycles are required. Wait cycles are generated by C5, a 7495 4 bit shift register. If one of the PROMs is selected, the inputs of the 7495 will be low, driven by Q1. PSYNC loads the register with Ø using the MD input. The Ø2 signal is used to clock C5, and as states progress, outputs A through D, respectivly go back high. Switches S1 through S4 select the number of cycles from zero (no switch closed) to four (S4 closed).

The hardware jump is accomplished by disabling the system address bus, and jamming a temporary address (or addresses). That is, the processor executes those instructions the jammed address points to. In the RESET MODE, the POC line loads C11, a 74177 4 bit counter, and resets flip-flop C4. This causes the A output to go high, enabling C10 to accept the PSYNC signal. This is used to clock the counter to simulate 4 consecutive addresses. Outputs B and C are gated to address bits AØ and A1, respectivly. Output D remains high throughout the 4 cycles, and goes back low after the 4th cycle. This output inverted and connected to the ADDR DSBL line via open

collector buffer C9. This also causes the A output to go low, preventing any further counting. In the USER-DEFINED MODE, this normally high user-defined signal goes low, presetting flip-flop C4 high. This determines address bit A2. On the low to high transition, the other half of C4 is clocked, and the Q output goes low. SM1 and Ø1 are NANDed so that the counter will not start until an instruction fetch is being executed. This signal and Q of C4 are NORed, inverted and gated to start the count sequence.


MTE E000 V1.1
MTE E100 V1.1
MTE E200 V1.1
MTE E300 V1.1
MTE E400 V1.1
MTE E500 V1.1
MTE E600 V1.1
MTE E700 V1.1
MTE E800 V1.1
MTE E900 V1.1
MTE EA00 V1.1
MTE EB00 V1.1

FM-1
Parts List
Revision A

BOARD: FM-1

Quantity	Description	Identifying Marks
1	Quad 2 Input NOR (Low Power Schottky)	SN74LS02N
1	Hex Inverter (Low Power Schottky)	SN74LS04N
2	Hex Buffer (Open Collector)	SN7407N
1	Dual 4 Input AND	SN7421N
1	Quad 2 Input NAND	SN7437N
1	Dual D Flip-Flop (Low Power Schottky)	SN74LS74N
1	4 Bit Shift Register (Low Power Schottky)	SN74LS95N
1	Presettable 4 Bit Counter	SN74177N/8281
3	Hex Tri-State Buffer	N8T97B
8	1K X 1 RAM	AM91L02BPC
12	256 X 8 PROM	AM1702A
1	4 to 16 Decoder	SN74154N
1	3.9 Volt Zener Diode	1N749
1	NPN Transistor	2N3904
1	5 Volt Positive Voltage Regulator	MC7805CP
1	5 Volt Negative Regulator	MC7905CP
9	1K Ohm, 1/4 Watt Resistor	brown/black/red
4	33 uF Tantalum Capacitor	33-25
17	.1 uF Disc Capactior	.1
2	Heat Sink	Black

FM-1
Parts List
Revision A

Quantity	Description	Identifying Marks
2 ea.	5/16 screw/nut/lockwasher	
13	24 Pin Soldertail Socket	24 Pin Socket
11	16 Pin Soldertail Socket	16 Pin Socket
9	14 Pin Soldertail Socket	14 Pin Socket
1	7 Position Dip Switch	Dip Switch
1	Printed Circuit Board	FM-1
6"	Bus Wire	
	Solder	

USER GUIDE

The address of the FM-1 is fixed at E000 hex, so no address selection is provided. All of the circuit options are provided on one 7 position dip switch in location C6.

The following lists the switch functions;

Switch	Function
1	Select one wait cycle
2	Select two wait cycles
3	Select three wait cycles
4	Select four wait cycles
5	Enable EXT CLEAR to backplane
6	Disable HARDWARE JUMP
7	Enable MEM WRITE to backplane

Select wait cycles according to the access times given for the PROMs used in the FM-1. For example: 1000 nanoseconds, select one wait cycle; 1500 nanoseconds, select two wait cycles, etc. PROMs with access times of more than 2500 nanoseconds cannot be used with the FM-1.

Switches S5 through S7 are selected according to your system configuration. For example, if your system does not have a control panel, S5 and S7 should be switched "on", so these signals are properly generated on the backplane. On the other hand, if a front panel is part of your system, then S5 and S7 should be switched "off" to avoid conflict between the FM-1 and the front panel.

Again, in systems with no front panel, switch S6 must be switched "off" to enable the auto-start jump hardware. Otherwise, the FM-1 would probably never be executed. If you have a front panel, it is your option, since you could manually run the FM-1 at E000 hex.

The RAMs supplied with the kit are fast enough to run with no wait cycles. If slower RAMs are substituted, remove Q1, which will allow the RAMs to run at the same speed as the PROMs.

Since the FM-1 has a fixed address, all software written for the FM-1 should be assembled at E000 hex.

The user-defined jump line may be jumpered to any of the uncommitted pins on the backplane. If this line is to be used, a "bounceless" switch must be made externally to the FM-1. This was done so the FM-2 can use this line to implement a firmware single-step.

SOFTWARE PACKAGE

The software currently shipped with the FM-1 is a system monitor and text editor. The monitor allows total control of memory and I/O, plus program debug routines. The text editor allows the user to create a file and write, modify, list, delete and save the file.

When the FM-1 is executed, the message;
FM-1 MONITOR, VERSION 1.1

is printed. The monitor now awatis command directives.

THE COMMAND LINE

The command line is used to enter command directives. Up to 80 characters may be entered on one line. The carriage return (ctrl/M) key is used to terminate the command line. If a mistake is made, the back arrow (shift O on most teletypes) or backspace (ctrl/H) may be used, or the entire line may be calceled with the cancel (ctrl/X) key. All characters are echoed (in full duplex mode) back to the printer, but non-functional control characters (ie. linefeed, null) are ignored. When the linefeed is issued after the carriage return, the monitor now expects another command directive.

COMMAND FORMAT

Most of the monitor's commands require arguments, that is additional information that must be given. In general, the command name is typed first, followed by arguments, if any, separated by one or more spaces. All command names may be abbreviated to its first three characters. All addresses are to be given in hexadecimal. Numeric values are assumed to be decimal, unless suffixed by a base identifier. (B: Binary O or Q: Octal, D: Decimal, H: Hexadecimal)

COMMANDS

DEPOSIT Format: DEPOSIT address

This command is used to enter hexadecimal data into memory. After the command and its address argument are entered, the monitor issues a prompt of the address where it will assign the first data item to. The user then types as many hex data items as he wishes, separated by one or more spaces. When the carriage return key is hit, the user is reprompted with the new address where he left off. The command is terminated by typing a colon (:) preceded by one or more spaces.

RUN Format: RUN address

This command is used to execute a user program at a specified address. All CPU registers are restored with this command. If the address was not specified, execution begins at the stored program counter value.

EXAMINE Format: EXAMINE address address numeric

This command lists the contents, in hexadecimal, between the first address and the second address, inclusive. If the second address is not specified, only the first address will be listed. The third number specifies how many bytes per line are to be listed. This number can range from 1 to 22. A value of 16 is assumed by default, if not given.

OUTPUT Format: OUTPUT address numeric

This command outputs the numeric data to the port specified by the address argument. This is useful for setting software-driven peripherals. (ie. software memory protect, programmed output lights)

INPUT Format: INPUT address

This command prints tthe hexadecimal value of the input port specified by the address argument.

NULL Format; NULL numeric

Sets the number of nulls to be printed during CRLF. (Carriage Return/Linefeed) Use for slower printers.

ECHO Format: ECHO ON or ECHO OFF

Echo on will echo the data from the keyboard to the printer. Echo off inhibits this echo. Echo off should be used on half-duplex systems.

LEADER Format: LEADER

This command punches six inches of leader (blanks) for teletypes.

PUNCH Format: PUNCH address address

This command is used to punch a block of memory on paper tape in hexadecimal format. Six inches of leader and trailer are punched before and after the data block. Both arguments must be specified.

TAPE Format: TAPE or TAPE R

This command reads and dumps into memory a hexadecimal tape. If the "R" argument is given, the data will be read from the TR-1 paper tape reader.

CUSTOMER Format: CUSTOMER

This command executes a user-written program at address 20 hex. If the program is terminated with a RET instruction, it will return to the monitor properly.

REGISTER Format: REGISTER register name "=" numeric

This command is used to examine and/or modify the CPU registers. If no arguments are specified, a list of all registers and values is printed. If a particular register is to be worked on, the name of the register is entered after the command. If the carriage return key is hit, the current value of that register will be printed. If that register is to be modified, an equals sign followed by the numeric value should be typed. (NOTE: Hexadecimal values must be suffixed with an "H")

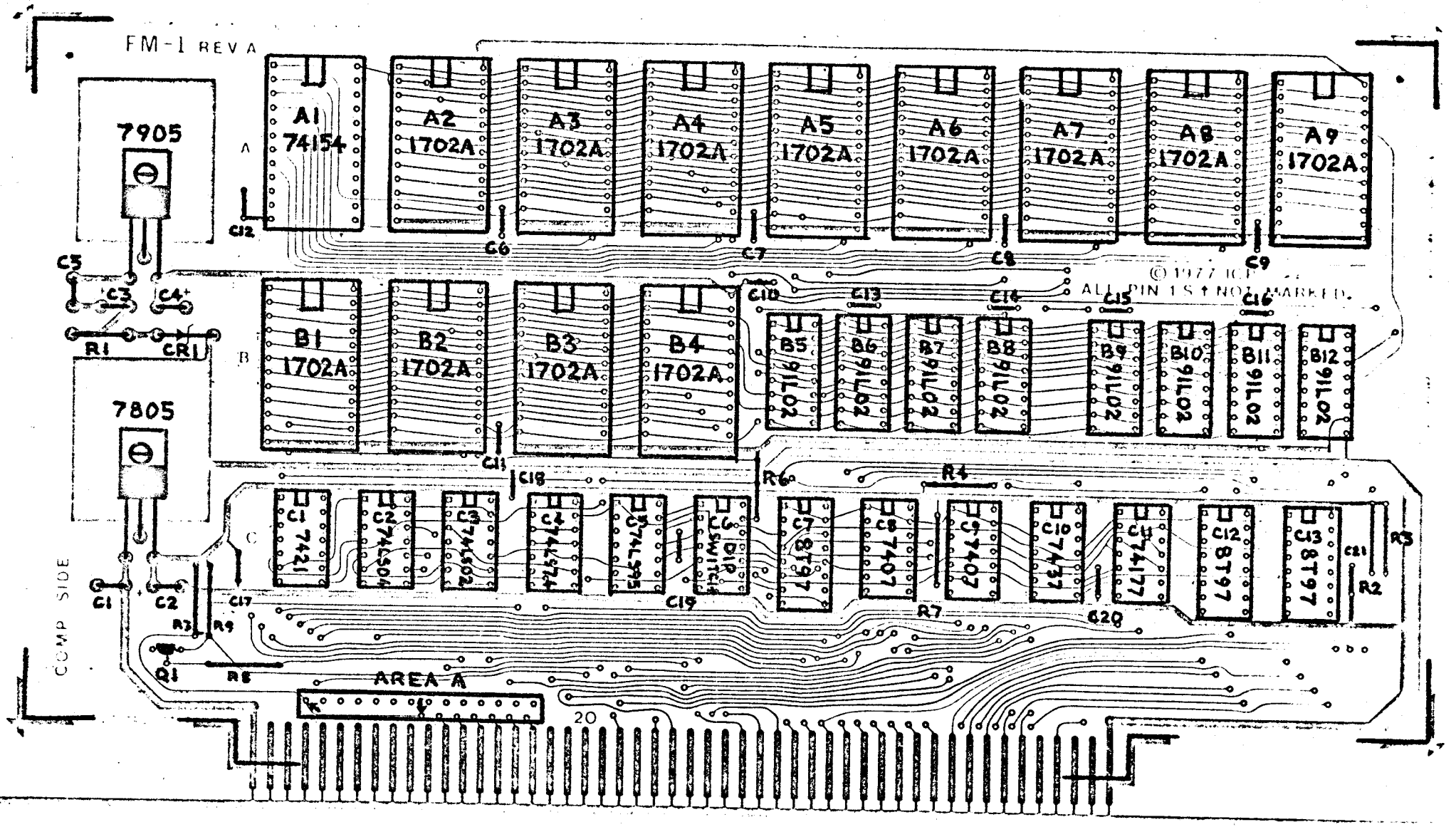
BREAK Format: BREAK address address...

With this command, the breakpiont structure may be displayed of modified. To display all currently set breakpoints, enter the command only. If a new breakpoint is to be entered, the command should be ebtered followed by the address, or addresses to be set. Only eight breakpoints may be set at one time.

EDIT Format: EDIT address address

This command is used to enter the FM-1 text editing system. If the editor is to be initialized, (following power-up or a reset)the command should be followedby the address that the text file is to begin at. The editor scans upward, until it finds a byte of non-existent memory or ROM. This address is used to identify the upper limit of the text buffer. If the entire section of RAM is not to be used by the editor, the desired end address may be entered following the starting address.
If the editor has been previously initialized, no addresses should be entered. This is done so the pointers kept by the editor will not be altered. A SYSTEM RESET DESTROYS THESE POINTERS!

FM-1 ASSEMBLY DIAGRAM
Revision A



ASSEMBLY INSTRUCTIONS

1) Unpack your kit and check all parts against the parts list in this doccumentation.

2) If the gold contacts on the edge connector appear to be corroded, use a pencil eraser to remove any oxidation. NOTE: Do not use Scotchbright or any abrasive material as it will remove the gold plating.

RESISTOR INSTALLATION

3) Insert and solder each of the nine 1K $\frac{1}{4}$ watt resistors (brown/black/red) R1 through R9. See Assembly Diagram for location.

IC SOCKET INSTALLATION

4) Insert but do not solder nine 24 pin sockets in locations A1 through A9. Do the same for the four remaining 24 pin sockets at locations B1 through B4. Eight of the eleven 16 pin sockets are installed at locations B5 through B12. Insert the three remaining 16 pin sockets at locations C7, C12 and C13. The nine 14 pin sockets may be installed at locations C1 through C5, and locations C8 through C11.

5) Place a stiff flat object (book, heavy cardboard, etc.) over the sockets. Very carefully invert the whole apparatus, making sure not to let any of the sockets slip. When this has been successfully done, solder the corner pins ONLY of each socket. (ie. pins 1 and 8 of a 14 pin socket)

6) Now that the sockets are held in place, they must be pressed flush against the surface of the PC board. This is done by applying pressure with the thumb and re-melting the corner pins of that socket. The socket will "click" into place when it is seated. Follow this procedure for all sockets.

7) Solder all of the remaining unsoldered pins for all of the IC sockets.

DISCRETE COMPONENT INSTALLATION

8) Insert and solder the one zener diode. See Assembly Diagram for location.

9) Insert and solder each of the seventeen .1 mf capacitors. See Assembly Diagram for locations.

10) Insert and solder the four 33 mf tantalum capacitors at locations C1 through C4. NOTE: Observe polarity as indicated on the board.

11) Following the procedure for IC sockets, insert and solder the one 7 position DIP switch at location C6. Switch #1 is oriented towards the top of the board.

12) Insert and solder the one 2N3904 transistor. Observe orientation as shown in the Assembly Diagram.

REGULATOR AND HEAT SINK INSTALLATION

13) Take the one 7905 regulator (do not confuse the 7805) and bendthe leads at 90 degree angles to a length which will match the hole pattern on the board. Place a heat sink at the location marked 7905 on the Assembly Diagram. Use a #6 screw on the component side of the board and a lockwasher and nut on the solder side of the board. Tighten the screw carefully to insure proper alignment of the heat sink to prevent shorting to adjacent traces.

14) In a similar manner, install the one 7805 regulator at the location marked on the Assembly Diagram.

IC INSTALLATION

15) Insert the one 74LS02 in the socket at location C3. Pin 1 on this and ALL other IC's point towards the TOP of the board.

16) Insert the one 74LS04 in the socket at location C2.

17) Insert the two 7407's in the sockets at locations C8 and C9.

18) Insert the one 7421 in the socket at location C1.

19) Insert the one 7437 in the socket at location C10.

20) Insert the one 74LS74 in the socket at location C4.

21) Insert the one 74LS95 in the socket at location C5.

22) Insert the one 74177 in the socket at location C11.

23) Insert the three 8T97's in sockets at locations C7, C12 and C13.

24) Insert the one 74154 in the socket at location A1.

25) Insert the eight 91L02's in sockets at locations B5 through B12.

26) Insert eight of the twelve 1702A's marked E000 through E700 in locations A2 through A9 respectively.

27) Insert four 1702A's marked E800 through EB00 in sockets at locations B1 through B4 respectively.

28) Set dip switches to your system according to the User Guide.

FM-1
Corrections
Revision A

CORRECTIONS

There are some artwork problems on the PC board. They may be remedied by the following modifications.

1) Install a jumper from pin 19 of B4 (1702A) to pin 2 of B5 (91L02).

2) Cut trace going to pin 8 of A8 (1702A).

3) Cut trace going to pin 10 of A8 (1702A).

4) Cut trace going to pin 4 of C12 (8T97).

5) Cut trace going to pin 6 of C12 (8T97).

6) Install a jumper from pin 7 of A8 (1702A) to pin 12 of B9 (91L02).

7) Install a jumper from pin 8 of A8 (1702A) to pin 6 of C12 (8T97).

8) Install a jumper from pin 9 of A8 (1702A) to pin 4 of C12 (8T97).

9) Cut two traces coming from feed-thru's between pins 4 and 13 of C7 (8T97) on solder side.

10) Install a jumper from pin 11 of B5 (91L02) to feed-thru closest to pins 13 and 14 of C7 (8T97).

11) Install a jumper from feed-thru near pin 11 of B5 (91L02) to feed-thru near pin 4 of C7 (8T97).

12) Cut trace going to pin 21 of B3 (1702A).

13) Cut trace going to pin 20 of B3 (1702A).

14) Install a jumper from pin 21 of B3 (1702A) to feed-thru near pin 77 of edge connector at bottom.

15) Install a jumper from pin 20 of B3 (1702A) to feed-thru near pin 78 of edge connector at bottom.

TEXT EDITOR

The FM-1 Text Editor is a character-oriented text editing program allowing the user to create and modify an ASCII text file. The editor is generally used to create source files for the FM-2 assembler. The editor can lend itself to other uses, however, such as letter writing, page composition, preparing off-line programs for another system, etc. Provisions are also made for saving and loading the file from a mass storage device for future editing needs.

When the Text Editor is entered, using the monitor's EDIT command, the area for the text buffer is scanned to make sure you have RAM to edit in. After this is done, internal pointers are set and

FM-1 TEXT EDITOR, VERSION 1.1

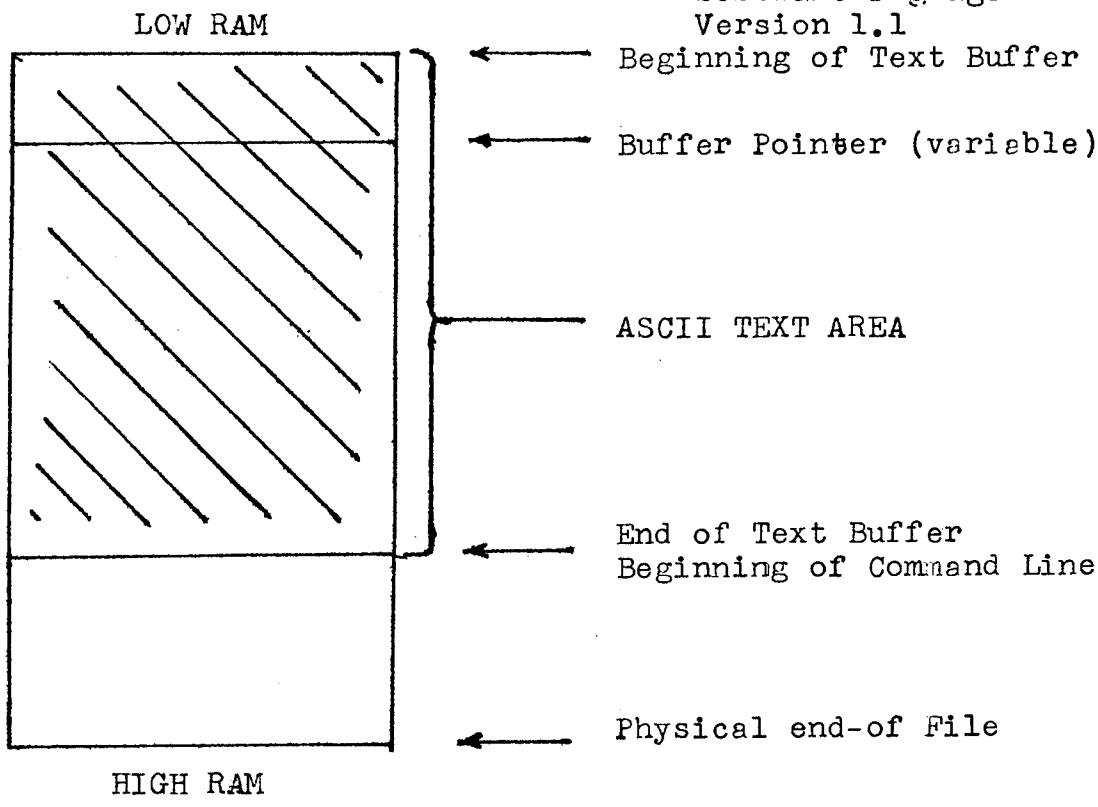
is printed. An asterisk "*" appears on the next line indicating the editor is ready for use.

THE COMMAND LINE

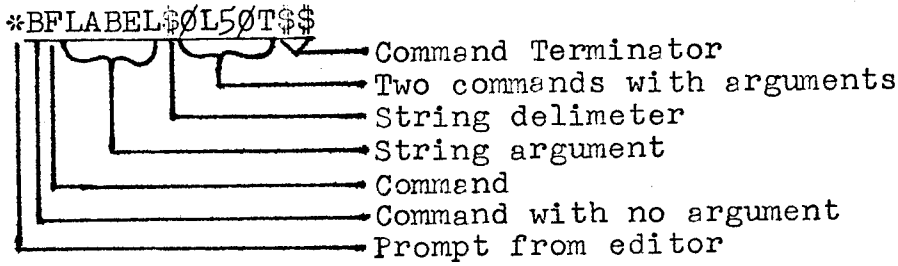
The command line is used to enter commands and text into the Text Editor. The length of the command line is limited only by the amount of RAM allowed in the Text Buffer. A pair of ESC (Escape) or alt. mode characters are used to terminate the command line and tells the editor to begin execution. The carriage return (CR) key has no effect, other than echoing an additional linefeed character. The RUBOUT key is used to correct mistakes. This causes an echo of the character previously typed, as an indication of where the "cursor" is. The backspace (ctrl/H) key may also be used, but no character echo will occour. The CANCEL (ctrl/X) key is used to delete the entire line, and re-prompt for a new command. The horizontal TAB (ctrl/I) character is accepted, and is echoed as the proper number of spaces necessary to move to the next tab stop. Tab stops are located on every eighth column. Multi-statement lines may be aborted with a ctrl/C. A new command line may be entered only when prompted by the "*".

THE TEXT BUFFER

The text buffer is the area in RAM set aside for the ASCII character text. Its size is variable, shrinking as text is deleted, and enlarging as text is added. When the editor is initialized, the size is zero. The following "maps" a typical Text Buffer.



The ASCII Text Area begins at the first location of RAM allocated by the EDIT command from the monitor. The buffer expands upward as more text is added. Command strings are entered starting at the current "end of buffer", expanding upward until no more file space is avaliable.



The above example is a command in which the editor moves to the beginning of the Text Buffer, finds the string "LABEL", moves to the beginning of that line and types 50 lines of text starting at that point. Note that multi-statements may be put on one line, which can be a real time saver. Note also that numeric arguments are placed in front of the command letter, and that all commands require a single letter.

COMMANDS

B Begin buffer Format: B
Move buffer pointer to beginning of Text Buffer

Z Zero buffer Format: Z
Move the buffer pointer to end of Text Buffer.

I Insert text Format: Istring to be inserted$
Insert the text following the command into the text buffer following the buffer pointer. ("$" is the ESC key in all examples.

E Exit Format: E
Exits to FM-1 monitor leaving all pointers for re-entry with another EDIT command.

T Type out text Format: nT
Types out n number of lines following the buffer pointer, if n is positive. If n is negative, the n lines behind the buffer pointer are typed up to the buffer pointer. If n is zero, characters from the beginning of the line up to the buffer pointer are typed.

P Punch out text Format: nP
Punches out to TTY punch (or cassette tape) lines of text similar to the T command, with the exception of the tab characters, which are not expanded.

A Append text Format: A
Appends lines of text to the text buffer from the cassette device, in a similar manner as the I command. A ctrl/z is used to indicate the end of tape. NOTE: The A command must be the LAST command on a line.

R Read text Format: R
Performs the same function as the A command, except text is read from the paper tape reader device.

N Null (leader) Format: N
Punches sixty (six inches) of leader (nulls) to the paper tape punch on the TTY. Several N commands may be used in a row for longer leaders.

C Character Format: nC
Moves the buffer pointer n characters to the left, if n is negative, or to the right, if n is positive. No action is taken if n is zero.

D Delete character Format: nD
Deletes n characters to the left of the buffer pointer, if n is negative, or to the right if n is positive.

L Line Format: nL
Move the buffer pointer n lines back, if n is negative. If n is positive, the buffer pointer is moved n lines ahead. If n is zero, the buffer pointer is moved to the beginning of the current line.

K Kill Format: nK
If n is positive, kill the next n lines following the pointer. If n is negative, kill the last n lines behind the pointer. If n is zero, kill from the beginning of the line to the buffer pointer.

F Find string Format: Fstring$
Search for the string of characters following the command until the esc character. when the string is found, the buffer pointer is left following the last character of the string.

S Substitute Format: Sstring$string$
Substitute the first occourence, following the buffer pointer, of the first string with the second string specified. The null string should NOT be used as either of the arguments.

NOTES ON USING THE TEXT EDITOR

1) All ESC or alt. mode characters are echoed as dollar signs ($). This is done to make it easier to see these delimeters.

2) If the buffer pointer is positioned at some point within a line, and you wish to type out the line without disturbing the buffer pointer, use ØTT instead of ØLT.

3) When a numeric argument is expected, a 1 is assumed by default if it is left out. This makes it easier since many times you may wish to do something once (oTT instead of oT1T)