F8S DEVELOPMENT MODULE
USER'S MANUAL

FAIRCHILD
SEMICONDUCTOR

AUGUST 14, 1975

REVISION L

F8S DEVELOPMENT MODULE

Introduction

Significant cost reductions may be obtained by using the F8 Microcomputer in a system. These advantages are a result of the unique architecture of the F8 as well as the high volume, low cost characteristics of MOS devices. To facilitate the design and development of these OEM systems, Fairchild now offers a microcomputer development system, the F8S. Fair-Bug, Fairchild's interactive debug program, is an integral part of the F8S.

The F8S is a microcomputer, complete with 1K bytes of RAM memory and 2K words of PROM locations, mounted on a printed circuit board. A connector is provided to bring the I/O ports and interrupt lines off of the board for interfacing with external equipment. Other connectors allow for easy memory and I/O expansion through use of the F8S memory expansion module. The F8S contains operating control switches and data display LEDs mounted on the PC board. The micromodule is a test vehicle which can be connected with the rest of the system and then the entire system can be proved out at full speed. The micromodule is a complete system--only power supplies need to be added. The micromodule is a simple and basic system--the hardware controls and displays eliminate the need for intervening layers of software operating systems which would separate you from your program.

Fair-Bug is a special debug program contained in its own 3851 Program Storage Unit (PSU). This debugging program has been developed by Fairchild to provide the F8S user with a convenient and powerful programming debug facility which is used to aid in the development of software on the F8S. It also provides the user with an interactive system via a teletype or CRT terminal.

TABLE OF CONTENTS

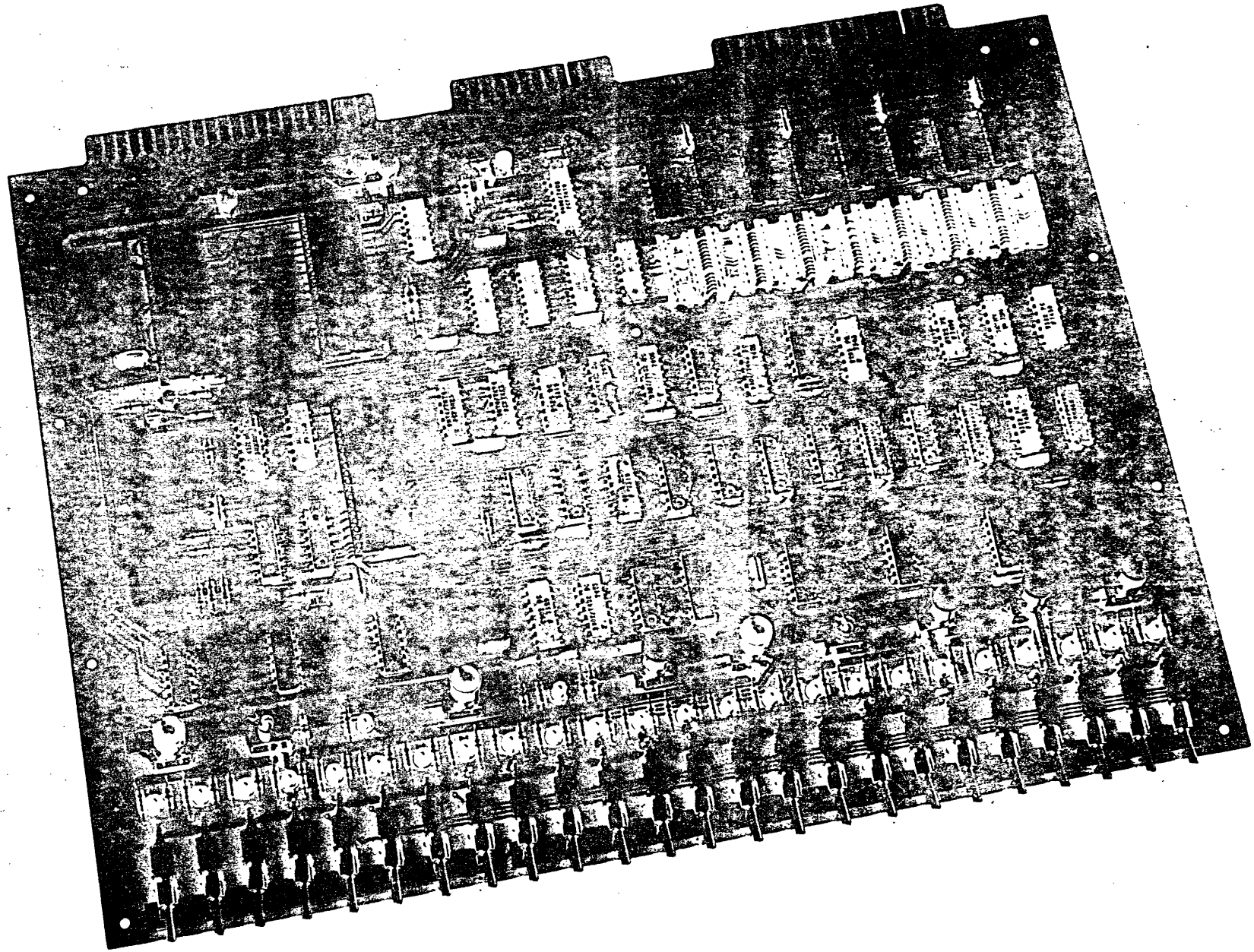
SECTION

APPENDIX

SECTION 1 - F8S MICROMODULE FEATURES

The F8S Micromodule features provide an easy-to-use program development system.

- all the power of the F8 chip set, including
  - Central Processing Unit
  - Memory Interface Circuits
  - 32 pins of TTL Compatible I/O
  - two Interval Timers
  - a complete interrupt system with two independent levels
  - programmable interrupt vector
  - Power-on Reset
  - 2μsec basic instruction time
  - self contained RC clock
- 1K of static RAM
- provisions for up to 2K of PROM memory
- control over memory allocation so that a program may either start from RAM or PROM
- 8 Data Switches and 16 Address Switches to alter memory address and data
- 8 Data LEDs and 16 Address LEDs to display memory address and data
- operating controls for run, halt, single instruction execution, and entry to Fair-Bug program
- a teletype adaptor that converts I/O signals to a four wire full duplex 20 milliamp teletype interface
- memory write data, read data, address, and memory control signals available on a connector
- F8 Data Bus, control lines (ROMC), and clocks available on a connector for I/O expansion
- an external signal, such as an address compare, can halt the system to aid in debug.
- separate power line for memory and memory disable to facilitate standby operation.



Fair-Bug Features

Convenient operator interface thru teletype printer and keyboard gives all the following capabilities:

- Display or Alter Memory Locations
- Display or Alter Scratchpad Registers
- Display or Alter Accumulator, ISAR, Status (W Register)
- Display or Alter PC0, DC0, DC1
- Load Formatted Paper Tape
- Punch Formatted Paper Tape
- Punch Paper Tape in PROM format
- Entry to Fair-Bug from Keyboard or by Program Instruction
- I/O Subroutines available to user

SECTION 2 - OPERATING CONTROL SWITCHES

The operating controls are divided into two groups that correspond to the two modes of the micromodule system.

- EXECUTE mode: F8 program can be executed either continuously or by single instruction steps. Active switches in the execute mode are:

  RUN
  HALT/SINGLE STEP
  DEBUG

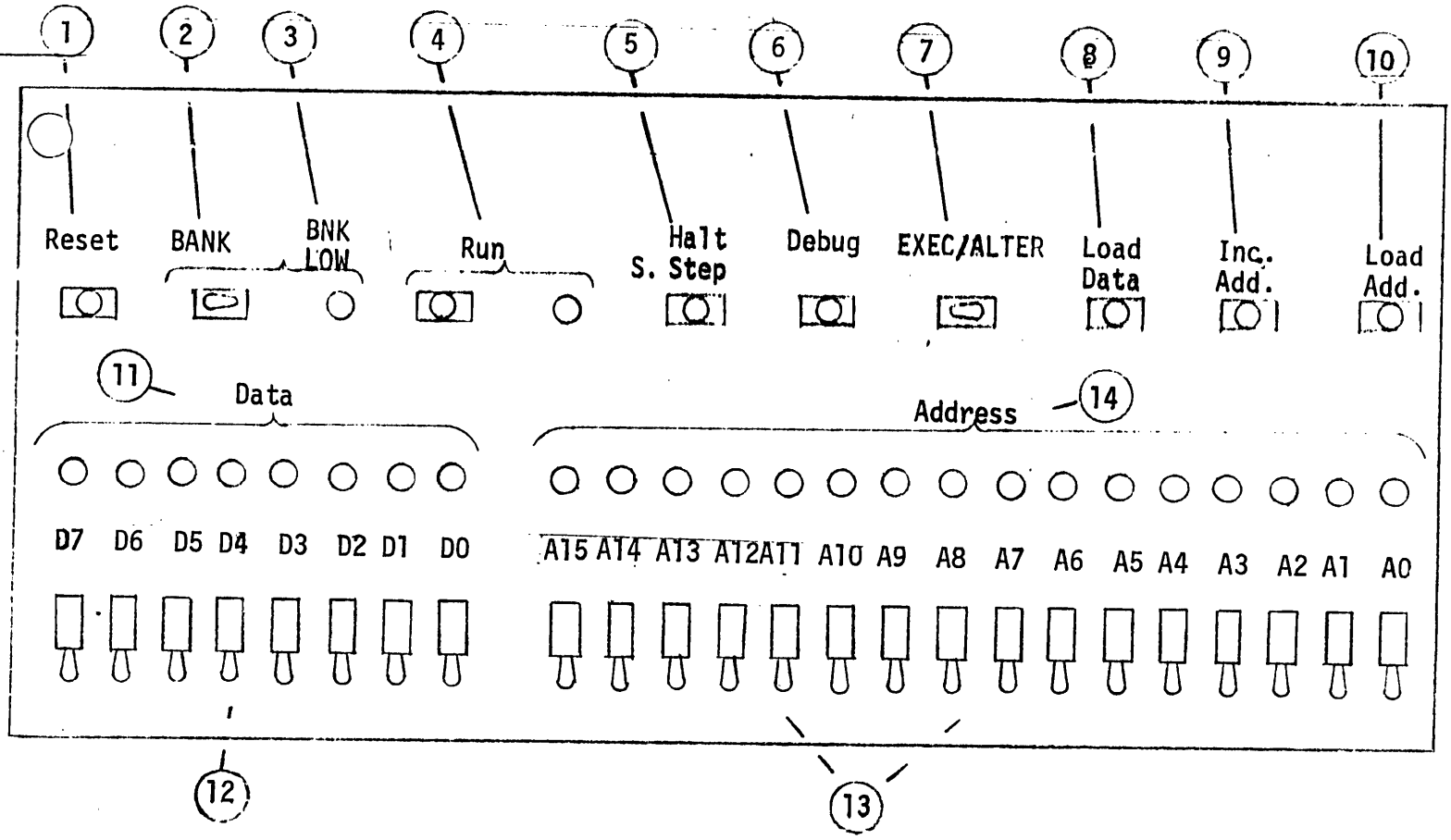
- ALTER mode: Program execution operator controls can be used to alter memory locations and to change the memory address. Active switches in the ALTER mode are:

  LOAD ADDRESS
  INCREMENT ADDRESS
  LOAD DATA

Operator controls that are active in both modes are:

RESET
BANK (Memory Page Allocate)
EXECUTE/ALTER

Figure 1 shows the operating controls and gives an introductory explanation of each switch. The following pages give additional explanation.



CONTROL & STATUS

1. RESET causes the program counter to be set to zero.
2. BANK controls allocation of RAM and PROM memory.
3. BNK LOW LED indicates that location ØØØ is in PROM memory.
4. RUN starts continuous instruction execution. LED indicates RUN status.
5. HALT/S.STEP stops instruction execution. HALT/S.STEP subsequently controls single instruction execution.
6. DEBUG forces a jump into the FAIR-BUG program.
7. EXEC/ALTER sets the mode of the system to EXECUTE or ALTER mode. Memory locations can be changed by the operator only in ALTER mode.
8. LOAD DATA loads the DATA switch setting into memory during ALTER mode.
9. INC ADD increments the memory address during ALTER mode.
10. LOAD ADD sets the memory address during ALTER mode.

MEMORY DATA DISPLAY & SETTING

11. MEMORY DATA LEDs display contents of memory.
12. MEMORY DATA switches provide the data byte that is placed into memory by LOAD DATA.

MEMORY ADDRESS DISPLAY & SETTING

13. MEMORY ADDRESS switches provide the new address that is loaded by LOAD ADD
14. MEMORY ADDRESS LEDs display the memory address.

Figure 1

2.1 EXEC/ALTER toggle switch: This switch sets the mode of the micromodule system. The system will be in the EXECUTE mode when this switch is in the EXECUTE position. The system will be in the ALTER mode when the switch is in the ALTER position and the system has been halted so that the RUN LED is dark.

2.2 RUN pushbutton: Pushing this switch while in the EXECUTE mode will cause the system to continuously execute, beginning with the instruction accessed by the current contents of the PROGRAM COUNTER.

The RUN switch is only functional in the EXECUTE mode. Pushing the switch after the system is already running has no effect. The RUN LED is lit whenever the system is running.

2.3 HALT/S.STEP pushbutton: Depressing this switch while in the EXECUTE mode and while the system is running will suspend CPU execution at the end of the current instruction. Each subsequent depression will cause the system to execute the next instruction. Operation of the switch has no effect while in the ALTER mode.

The RUN LED will go out when HALT is depressed. The ADDRESS LEDs will display the contents of the PROGRAM COUNTER while halted. The PROGRAM COUNTER will point at the next instruction that will be executed; this instruction will be displayed in the DATA LEDs. Execution of previous instructions will be complete; for instance if the last instruction executed before halting was an output instruction, the I/O ports will have already been updated by that output instruction.

The system halt is a psuedo halt--psuedo in that the clocks are not stopped. Hardware on the micromodule forces a NOP instruction into the F8 CPU and a NOP state is forced onto the ROM command lines. Since the clock is not stopped, there will not be any data loss. The interval timers of the PSU and MI chips will continue to count down toward the time-out state even while the system is halted.

If either a timer interrupt or external interrupt occurs while the system is halted, and if the ICB of the CPU is set to accept interrupts, then the interrupt will be acknowledged. The system will execute the four

cycles of the interrupt push sequence and will halt again at the first instruction of the appropriate interrupt service routine--in this case, for example, at address H'0020' if a timer interrupt occurred in the Fair-Bug PSU.

2.4 DEBUG: This momentary switch is used in the EXECUTE mode to set the PROGRAM COUNTER to the entry address of FAIR-BUG and to then start executing the FAIR-BUG monitor as it is released. The DEBUG switch is interlocked with the RUN status so that the system must be halted before the DEBUG switch will be operative. The Interrupt Control Bit (ICB) of the 3850 CPU is reset, disabling interrupt servicing, when the DEBUG switch is pressed.

The DEBUG switch has the same function as the RESET switch while in the ALTER mode. The PROGRAM COUNTER is set to ØØØØ and the ICB of the CPU is reset.

2.5 LOAD ADD pushbutton: Pushing this switch will load the settings of the 16 address switches into the memory address register while in ALTER mode. The switch is inoperative in the EXECUTE mode.

The LOAD ADD loads the memory address register; it does not effect the PROGRAM COUNTER. The PROGRAM COUNTER does not change while the system is in the ALTER mode; the PROGRAM COUNTER will hold the same address it held when the HALT was last used. The memories--both PROM and RAM--are static; the output data from the memories will respond immediately to a change of the address held in the memory address register.

2.6 INC ADD pushbutton: Pushing this switch will increment the memory address while in ALTER mode.

2.7 LOAD DATA switch: Pushing this switch while in ALTER mode will load the settings of the eight DATA switches into the location addressed by the contents of the memory address register, as indicated by the memory ADDRESS LEDs. The memory address will be incremented as the pushbutton is released.

On many occasions it may be necessary to halt the system and modify the next instruction; this is easy to do. The memory address register holds the same value as the PROGRAM COUNTER whenever the system is taken from EXECUTE mode to ALTER mode; it will retain this value, pointing at the next instruction, until it is incremented by an INC ADD step or a LOAD DATA step or until it is replaced with a new value by a LOAD ADD step.

A sequential group of memory locations can be altered conveniently because the memory address is incremented to point at the next location after each location is changed. So in this case the LOAD ADD will only need to be used once, to load the address of the first location.

While you are altering a group of locations, you have a chance to verify your DATA switch settings at each location before moving on to the next. This chance to check occurs while you hold the LOAD DATA switch depressed; at that time the DATA LEDs will show the new contents of the location you just changed. If you have made an error, it can be corrected right then if the LOAD DATA is still being held depressed, as the "write-to-memory" signal continues for as long as the switch is depressed; just correct the DATA switch settings to correct the error.

The address will increment to the next location when the LOAD DATA switch is released. An attempt to write to PROM memory has no effect except for incrementing the memory address.

2.8 RESET pushbutton: This debounced momentary switch is used to restart your program. It will clear PROGRAM COUNTER to ØØØØ and will reset the Interrupt Control Bit (ICB) of the F8 CPU to suspend interrupt servicing as it is released. No other registers are effected. RESET is operative in both EXECUTE and ALTER modes.

When RESET is used in the ALTER mode it will also clear the memory address registers so that memory location ØØØØ will be displayed.

The RESET switch does not change the RUN status of the system. If the system was running, it will still be running after RESET is used. If the system was halted, it will indicate RUN status while the RESET is held depressed but will halt as soon as the switch is released; the running

while RESET is held depressed is harmless, as the system is locked up in a tight loop executing the idle cycle of the reset sequence.

While the RESET button is held down, the system is held in an idle state. The system is forced into this idle state after the completion of the current machine cycle--which will not necessarily be at the end of an instruction in the case of multiple cycle instructions. The PROGRAM COUNTER will be reset to zero after the RESET button is released; specifically this action happens at the end of the first full machine cycle following the release of the switch. The instruction located at address ØØØØ will be fetched during the second full cycle if the system was running; otherwise the system will go back into the halt state.

2.9 BANK: A toggle switch that determines the page assignments for the PROM (2K bytes) and RAM (1K byte) memory. This function will allow assigning PROM to low order memory or RAM to low order memory locations. The BANK switch functions in conjunction with the RESET or MONITOR pushbuttons; the setting of the BANK toggle switch takes effect the next time the RESET or MONITOR switch is pushed. Figure 2 shows the two cases of memory allocation.

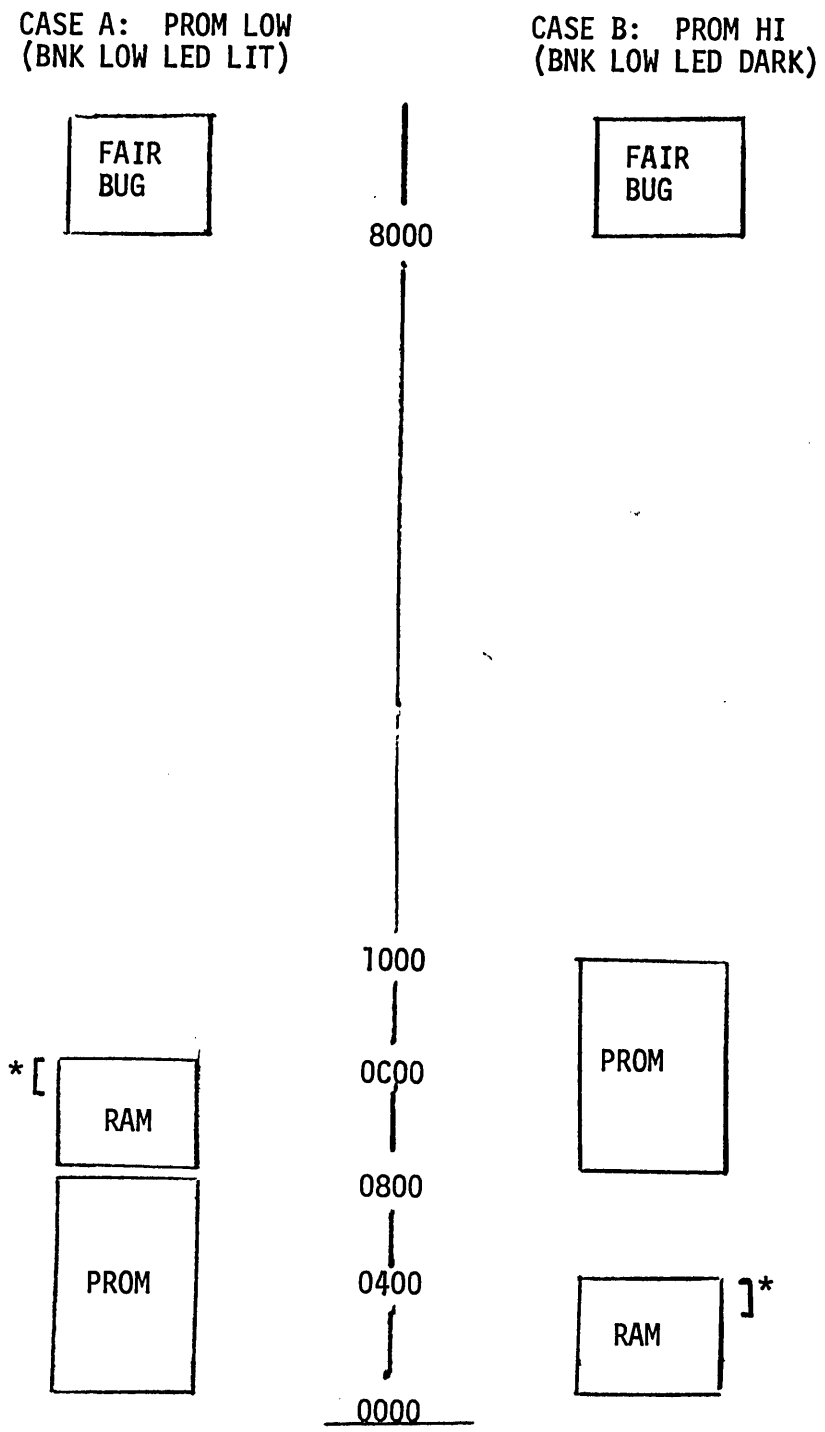
The BNK LOW LED is lit if memory location 0000 is in PROM; the BNK LOW LED is dark if memory location 0000 is in RAM. The exact moment when memory allocation changes is as the RESET or MONITOR switch is pressed down; the clearing of the PROGRAM COUNTER and resumption of instruction execution does not occur until after the RESET or MONITOR switch is released.

Console Displays

2.10 RUN LED: The RUN LED is lit if the system is executing a program. The system can only be running in the EXECUTE mode; the related controls are RUN and HALT/S.STEP.

2.11 BNK LOW LED: The BNK LOW LED is lit if the BANK and RESET controls have been used to make PROM to be low order memory and RAM to be high order memory as shown in Figure 2.

2.12 ADDRESS LEDs: The ADDRESS LEDs display the memory address which is held in the memory address register.



* FAIR-BUG temporary storage : 26 bytes

FIGURE 2: Switching of Memory Allocation

In the ALTER mode, the ADDRESS LEDs will be displaying the memory address that is currently being accessed and that can be modified by the operator. When the ALTER mode is first entered, the address displayed will be the same as currently held by the PROGRAM COUNTER; after that the address will depend on what you, the operator, do with the LOAD DATA, LOAD ADD, and INC ADD switches.

In the EXECUTE mode, the ADDRESS LEDs will display the current contents of the PROGRAM COUNTER while the system is halted. The PROGRAM COUNTER will be accessing the next instruction that will be executed. Whenever the system is running, the ADDRESS LEDs will be displaying the address provided by the F8 MEMORY INTERFACE logic--which may be from the PROGRAM COUNTER, the DATA COUNTER, or the STACK REGISTER, depending on the current instruction; the LED display may be a blur since the addresses are changing rapidly.

2.13 DATA LEDs: The DATA LEDs display the contents of the memory location being accessed, as indicated in the ADDRESS LEDs. The memory location may hold an instruction, an operand, a data constant, or garbage--all depending on what the memory address is and what you have in memory.

SECTION 3 - CONSOLE UTILIZATION

3.1 Debugging from the Console: The Fairchild F8S Program Development Module has both the console panel as described previously and also a special debug PSU to assist in program debugging. Usage of both is described in this section and the next section with reference to block diagrams to lead the user step by step through some common procedures.

The console panel may be used to debug a program; however, this requires reading LED's and changing toggles and pushing buttons to perform the desired functions. It also requires logging much information in order to keep a record of events and their sequence. With so many manual functions there are many chances of making mistakes and not achieving the desired goal of debugging the program and having it execute correctly.

Before describing Fair-Bug, an understanding of the console is useful since debugging may at times be a combination of console functions and Fair-Bug operations. The following examples are offered to familiarize the user with the console; these examples are not necessarily in a logical execution sequence, but instead are in a sequence of progressively more involved functions.

3.2 Program Start from the Console: Figure 3 shows how to start or restart a program execution from address 0.

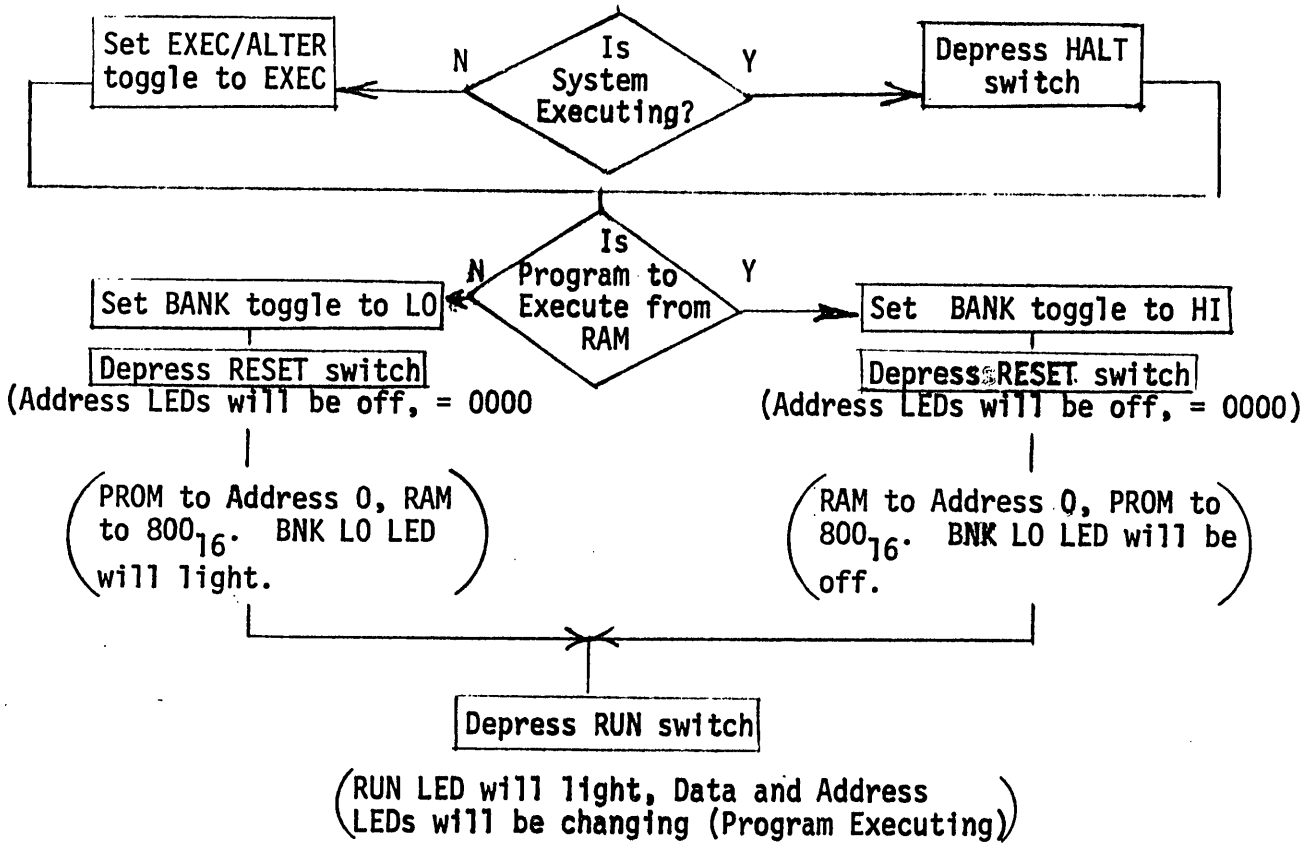

Is System Executing?
N
Y
Set EXEC/ALTER toggle to EXEC
Depress HALT switch
Is Program to Execute from RAM
N
Y
Set BANK toggle to LO
Set BANK toggle to HI
Depress RESET switch
(Address LEDs will be off, = 0000
Depress RESET switch
(Address LEDs will be off, = 0000)
(PROM to Address 0, RAM to 800$_{16}$. BNK LO LED will light.
(RAM to Address 0, PROM to 800$_{16}$. BNK LO LED will be off.
Depress RUN switch
(RUN LED will light, Data and Address LEDs will be changing (Program Executing))

FIGURE 3

3.3 Display or Alter Memory from the Console: Figure 4 depicts how to display or alter memory locations from the console.

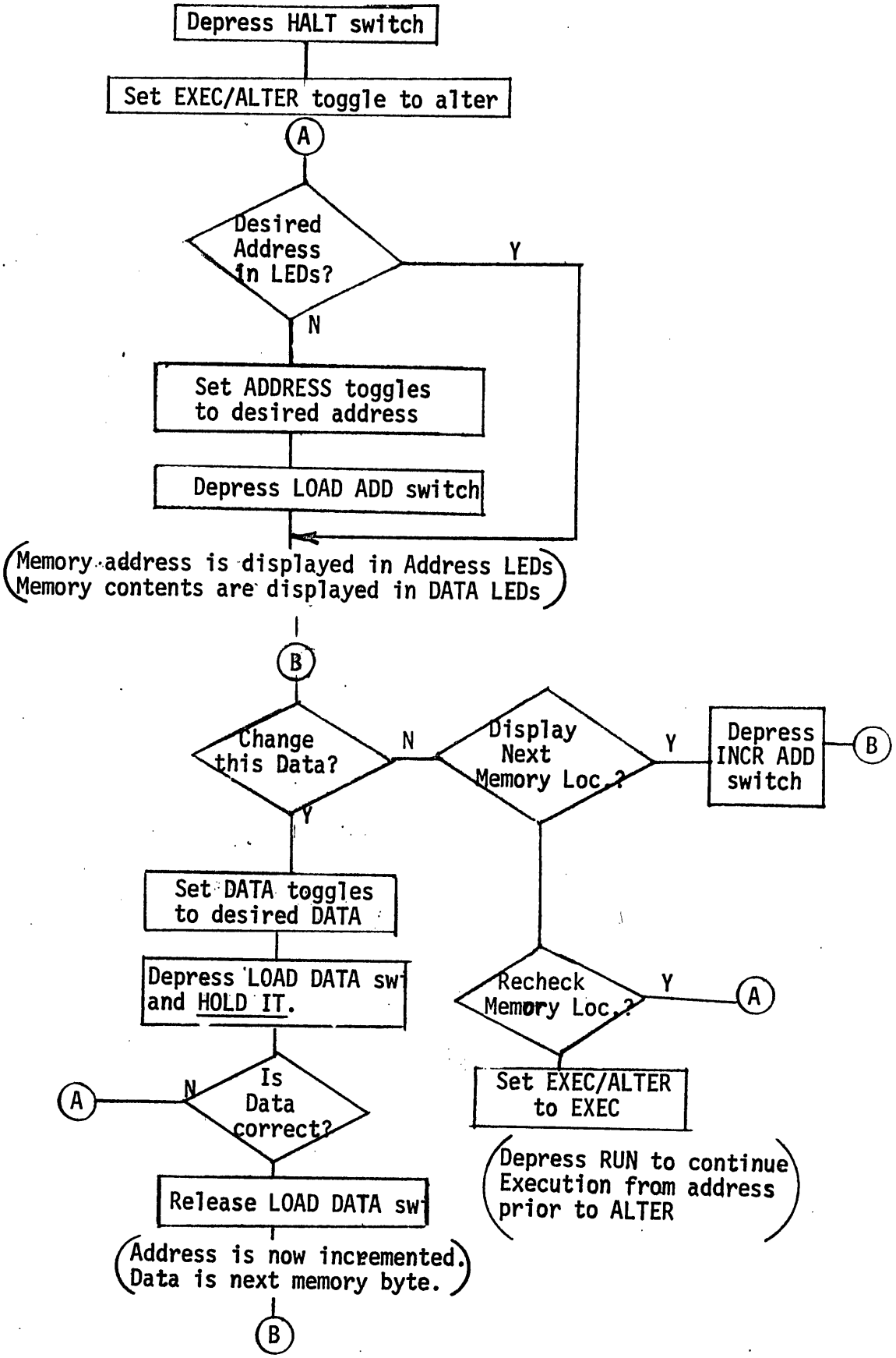


Figure 4

3.4 Loading User's Program: Figure 5 shows how to load a user's program from a high speed paper tape reader when no TTY is available for FAIR-BUG communications.

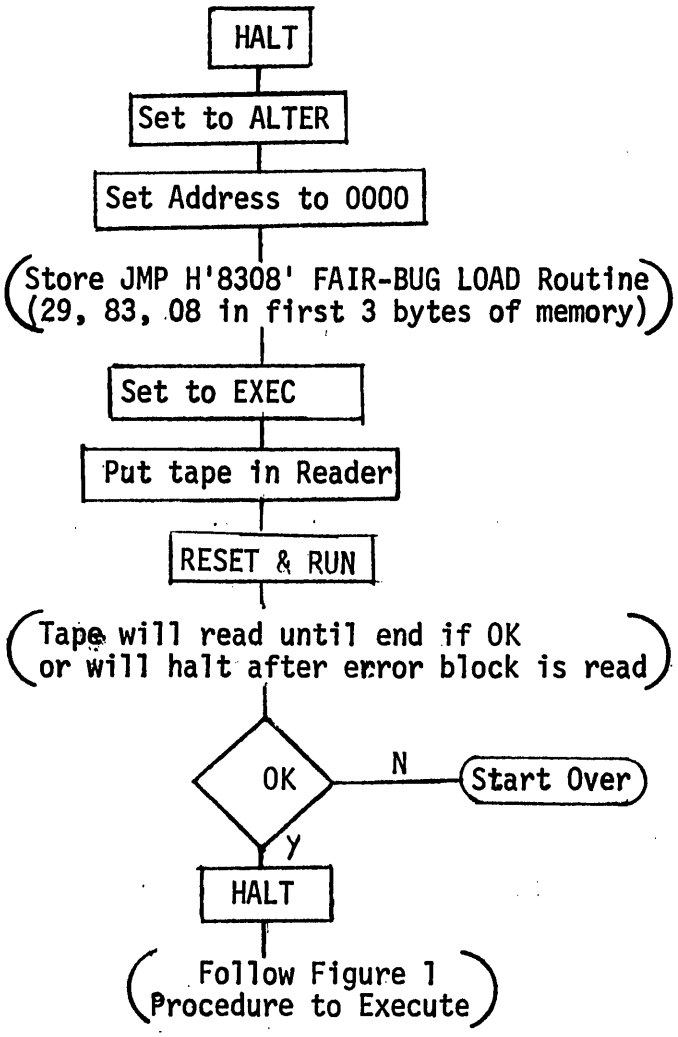


Figure 5

3.5 User Program Trace from Console: Program debugging from the console is a combination of patching instructions or data in memory and single stepping in order to trace instruction execution sequences through the user's program. Figure 4 shows how displaying and patching of memory is accomplished. It is important to log events in order to have a historical reference. Care must be used in using the console or else memory may be changed incorrectly causing wasted time and effort.

While executing a program, the Address LEDs have a quickly changing pattern; however, it does indicate program loops and general address areas being executed. A tight loop will be displayed as an almost constant pattern, the tighter the loop the brighter the LEDs.

In order to trace program execution the program must be modified to determine which instruction paths are being executed. This can be accomplished by substituting a Branch * (branch to itself) instruction for two bytes of memory. This will cause a 1 instruction loop that is easily recognized on the Address LEDs. At this point the HALT switch may be depressed and subsequent action taken.

If you should want to check out a area of code that, for instance, begins at address H'087', the two bytes H'90', H'FF' should be substituted for the instruction at H'087' and for the next byte at H'088' (see "How to Load or Examine Memory Locations", Figure 4, for the procedure). The two bytes, 90 FF, are a branch instruction that branches to itself; once the program gets to this instruction, it will loop there.

After the patch is made, start the program as shown in Figure 1. The program will run, executing instructions until it gets to the "halt"--which for the example chosen is at H'087'.

When the program gets to the Branch loop, the DATA LEDs will look like H'FF'. The ADDRESS LEDs will be a merging of the addresses of the two patched bytes--in this instance the LEDs will look like H'8F'.

At this point, halt the system and restore the original two bytes into locations H'087' and H'088'.

With the original bytes restored, you are ready to single step on into the code of interest.

To further discuss patching instructions into a program let us assume an example program segment that occupies RAM memory in the area of 153_{16}.

EXAMPLE PROGRAM

(address)	(op code)			(assembly statements)	
				Base	Hex
153	20	3C	EXMPLE	LI	H'3C'
155	B1			OUTS	01
156	13			SL	1
157	45			LR	A,5
158	77			LIS	7
159	27	14		OUT	H'14'
15B	29	03	00	JMP	BEYOND

Furthermore, let us assume that you have been checking out the program and have single stepped until the ADDRESS LEDs show H'155'. The system is halted. The LI instruction has been executed so that the accumulator will have been loaded with H'3C'. The OUTS instruction will be the next executed. The PROGRAM COUNTER will hold the H'155'.

At this point let us assume that you suddenly realize that two instructions need to be changed:

i) the LR A,5 at H'157' should be LR 5,A (55)
ii) the OUT H'14' at H'159' should be OUT H'15' (2715)

Figure 6 shows several optional ways of making these changes.

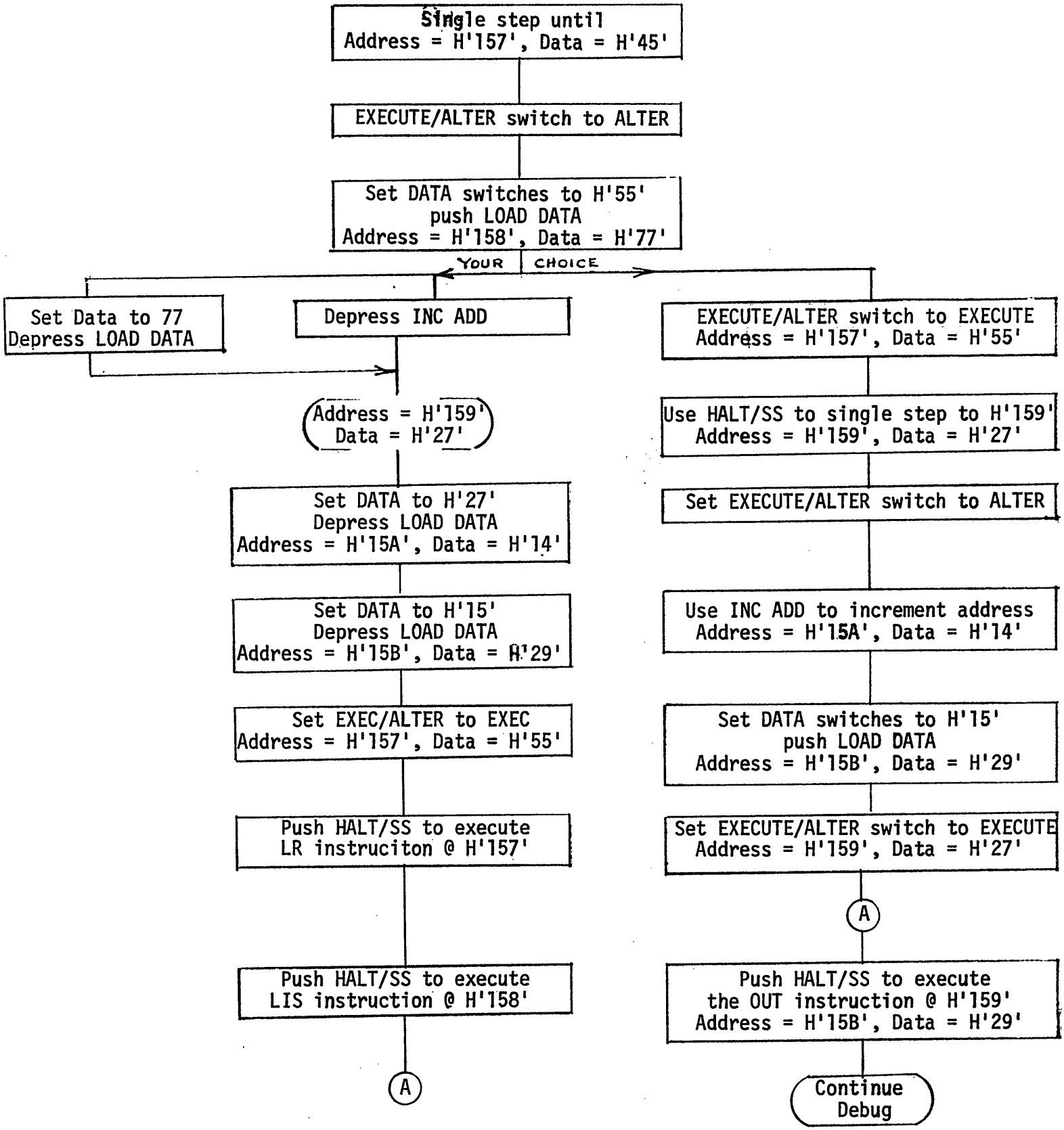

Single step until
Address = H'157', Data = H'45'
EXECUTE/ALTER switch to ALTER
Set DATA switches to H'55'
push LOAD DATA
Address = H'158', Data = H'77'
YOUR CHOICE
Set Data to 77
Depress LOAD DATA
Depress INC ADD
EXECUTE/ALTER switch to EXECUTE
Address = H'157', Data = H'55'
Address = H'159'
Data = H'27'
Use HALT/SS to single step to H'159'
Address = H'159', Data = H'27'
Set DATA to H'27'
Depress LOAD DATA
Address = H'15A', Data = H'14'
Set EXECUTE/ALTER switch to ALTER
Set DATA to H'15'
Depress LOAD DATA
Address = H'15B', Data = H'29'
Use INC ADD to increment address
Address = H'15A', Data = H'14'
Set EXEC/ALTER to EXEC
Address = H'157', Data = H'55'
Set DATA switches to H'15'
push LOAD DATA
Address = H'15B', Data = H'29'
Push HALT/SS to execute
LR instruciton @ H'157'
Set EXECUTE/ALTER switch to EXECUTE
Address = H'159', Data = H'27'
A
A
Push HALT/SS to execute
LIS instruction @ H'158'
Push HALT/SS to execute
the OUT instruction @ H'159'
Address = H'15B', Data = H'29'
Continue
Debug

Figure 6

SECTION 4 - FAIR-BUG UTILIZATION

In order to alleviate many of the problems associated with debugging with a console, a special Debug PSU has been developed by Fairchild to provide the F8S user with a convenient and powerful programming debug facility which is used to aid in the development of software on the F8S. This debugging program (FAIR-BUG) provides the user with an interactive system via a teletype or CRT terminal. The following capabilities are provided:

Display or Alter Memory Locations
Display or Alter Scratchpad Registers
Display or Alter Accumulator, ISAR, Status (W Register)
Display or Alter PC0, DC0, DC1
Load Formatted Paper Tape
Punch Formatted Paper Tape
Punch Paper Tape in PROM format
Entry from Keyboard or by Program Instruction
I/O Subroutines available to user

The FAIR-BUG PSU has the memory addresses 8000 - 83FF with the entry point being H'8080'. Input/Output Ports 4 and 5 are also assigned to this PSU. Entry to the debug program is achieved either by depressing the "ENTER DEBUG" switch or by program instruction PI H'8080'.

FAIR-BUG PSU will save the state of the machine upon entry and will restore it upon return to the users program. Register 8 and PC1 are lost by the system; however, under program control the user can save and restore these if necessary. The save area utilized by FAIR-BUG is scratchpad registers 3C to 3F and also the last 26 bytes of RAM whose address will either be 3E6 to 3FF or BE6 to BFF. The interrupt is disabled by FAIR-BUG which the user may re-enable if desired. See coding examples for methods of saving and restoring volitile registers and software linkage to FAIR-BUG in Section 4.6.

4.1 FAIR-BUG Commands: When FAIR- BUG is entered a prompt character (?) is sent to the output device. The user then has the option of using any of the debug commands. After each debug execution the user is again prompted with (?). All data and input parameters are in hexidecimal notation. C/R following a command indicates a carriage return.

COMMAND TYPE	COMMAND		FUNCTION
Display	A	C/R	Display the contents of the Accumulator
	D0	C/R	Display the contents of DC0
	D1	C/R	Display the contents of DC1
	I	C/R	Display the contents of ISAR
	M XXXX	C/R	Display Memory Location XXXX
	M XXXX-YYYY	C/R	Display Memory Location XXXX to YYYY
	P0	C/R	Display the contents of PC0
	P1	C/R	Display the contents of PC1
	R XX	C/R	Display the contents of Register XX
	R XX-YY	C/R	Display the contents of Registers XX to YY
	S	C/R	Display the contents of W Register, status
	W	C/R	Display the contents of W Register, status
Change	C XX	C/R	Change the previously displayed memory location or register to XX
	C XXXX	C/R	Change the previously displayed PC or DC to XXXX
Examine	E	C/R	Display the last addressed register or memory location
Next	N	C/R	Display the next register or memory location
Load	L	C/R	Load formatted object paper tape. If (CK) prints then checksum error has occurred on block last read
Punch	B XXXX-YYYY-Z		Binary punch PROM format XXXX is starting page address and YYYY is ending page address. Z is number of bytes per block. 0 = 256, 1 = 512. To punch 0 to BFF then enter B0-C00-0.
	F XXXX-YYYY	C/R	Formatted punch for future Load.
Go To	G	C/R	Go to address of PC0
	G AAAA	C/R	Change PC0 to address AAAA, then go to AAAA to execute next instruction
Delete Command	[		Delete command and start a new command input string.

4.2 FAIR-BUG Command Usage: A brief study of Appendix F will assist the reader in understanding FAIR-BUG command utilization. From these examples many advantages over console debugging are quite apparent. First and most important is a history record of the steps taken and the results displayed. Second is the accessability of the scratchpad registers as well as the other working registers of the system. Third is the simple access from the console to FAIR-BUG to the user.

One advantage that is not so apparent is that when a user wishes to end a debug session the user may save his modified program by punching the memory using the F command. In a subsequent debug session he may then load this "updated" program and continue to debug. If program patches were extensive much time and effort is saved. Of course the program may be re-assembled after editing the changes in order to obtain an updated program.

Another advantage and also not so apparent is the fact that Keyboard input is much less error prone. If the user uses a little care and examines his keyed input prior to the carriage return he can almost totally eliminate IP errors. In addition, a debug session achieves more results in a quicker time.

4.3 FAIR-BUG Input/Output: The FAIR-BUG input/output routines assume that port 4 of the PSU is available and is configured as shown in Appendix C. In order to communicate with FAIR-BUG an 11 bit serial type device such as a Teletype ASR 33 or compatible type TTY or CRT is required. FAIR-BUG has options to vary the BAUD rate by changing the counters for delay loops between bits. The timer is not utilized so the user is not deprived of this valuable resource. The counter value for 110 BAUD is 6 which counts by incrementing an 8 bit register until 0; the 6 gives a loop count of 250. For 300 BAUD the counter is initially H'A3' giving a loop count of H'5D' or 93 decimal. Other BAUD rates can be achieved by modifying the counter. These counts assumed a system clock of 2MHz. For a faster clock, the counter must be changed to produce more delay loops, while for a slower clock the counter must be changed to produce less delay loops. This is due to the fact that each instruction time will change and the total loop time will change while the device speed is always constant.

When the BAUD rate is not set to default to either 110 or 300 (it is not one of these, or else a parallel input device is also in the system), then the BAUD rate must be put into RAM location H'3FF' or if this location is a PROM address then into RAM location H'BFF'.

4.4 Load from Parallel Input Device: The loader in FAIR-BUG can load from either a teletype or from a parallel source. The usual parallel source would be a high speed paper tape reader which reads 100 to 300 characters per second. The parallel device is controlled using a "handshaking" protocol. The handshaking eliminates synchronizing problems because it forces the device to wait for the microprocessor and forces the microprocessor to wait for the device.

FAIR-BUG parallel read routine examines DEVICE READY and waits for the ready signal, it then looks for Character Ready and delays 100μsec after detecting the ready, then it reads a character before the output of Step Reader. This sequence is repeated for each character. Only the format shown in Appendix B can be read by FAIR-BUG with the Load command. However, the user may use the subroutine PINP to read other formats.

Bits 1-2 are examined when FAIR-BUG is entered to initialize the baud delay counter and also whenever a Load command is given to determine whether the input is bit serial or parallel on Port 5. See Appendix C for the I/O port pin assignments.

Figure 5 depicts in Flow Chart format the steps necessary to load a paper tape program formatted as shown in Appendix G.

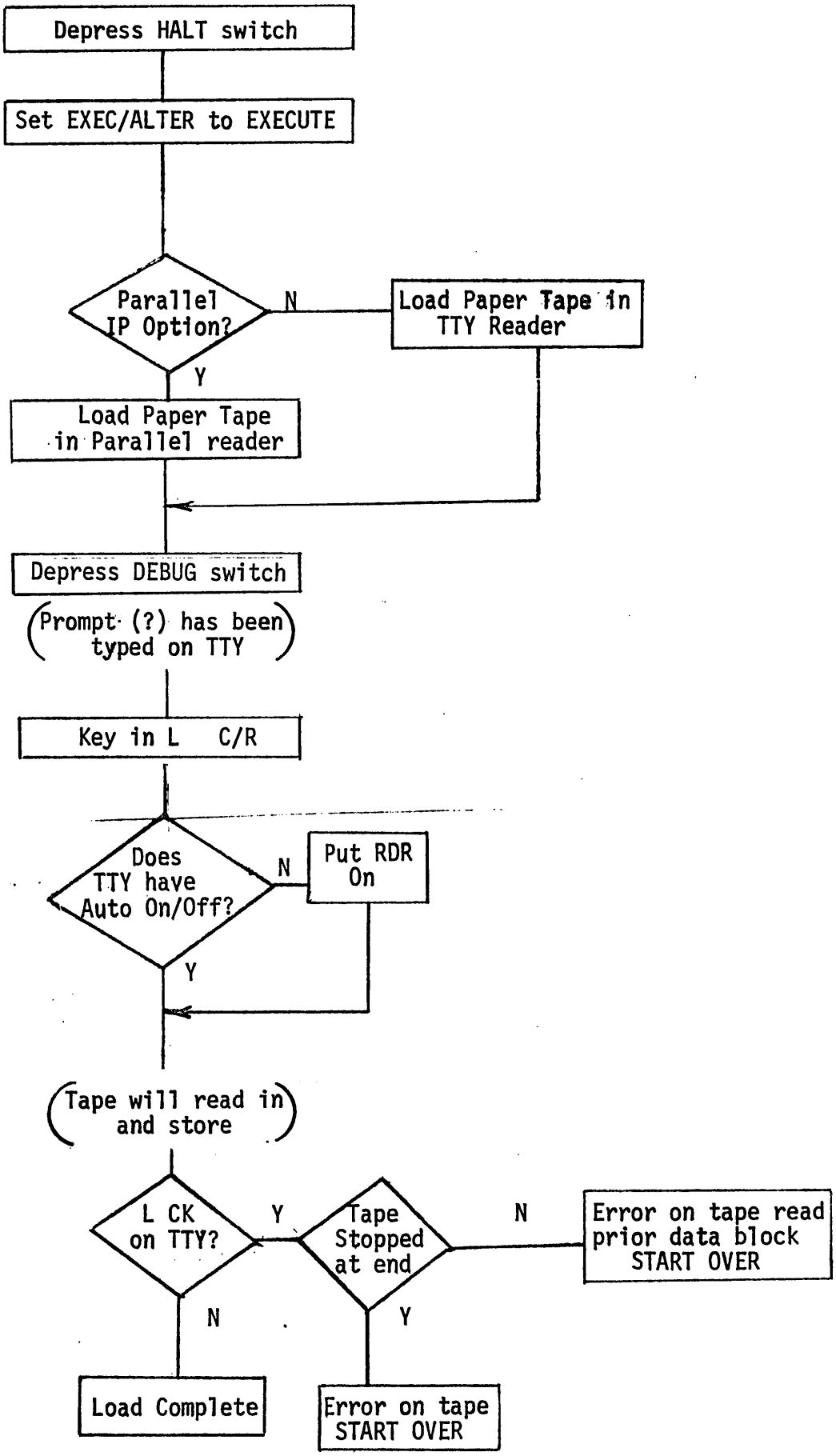

Depress HALT switch
Set EXEC/ALTER to EXECUTE
Parallel IP Option?
N
Load Paper Tape in TTY Reader
Y
Load Paper Tape in Parallel reader
Depress DEBUG switch
Prompt (?) has been typed on TTY
Key in L C/R
Does TTY have Auto On/Off?
N
Put RDR On
Y
Tape will read in and store
L CK on TTY?
Y
Tape Stopped at end
N
Error on tape read prior data block START OVER
N
Y
Load Complete
Error on tape START OVER

Figure 5

4.5 FAIR-BUG Subroutines: I/O Subroutines on the FAIR-BUG PSU are available to the user. These are listed below and documented in Appendix I.

NAME	ENTRY ADDRESS	FUNCTION
TTY1	83AD	Input 1 byte from TTY type device (11 bits serial/character)
TTY0	83E5	Output 1 byte to TTY type device (11 bits serial/character)
TTCR	83E4	Output CR, LF, & Null characters using TTY1 subroutine
PINP	8397	Input 1 byte from the parallel IP device (150μsec minimum delay between characters)
FOP1	80EC	Output 1 or 2 hexidecimal digits in ASCII format from register QL
FOP2	80EE	Output 1 or 2 hexidecimal digits in ASCII format from a memory location
BYTE	837B	Input 2 ASCII characters from a parallel or serial IP device; then convert them to one hexidecimal byte

4.6 Programmable Examples: The following program linkages to FAIR-BUG are cited as examples as how to utilize FAIR-BUG as a "snap shot" display or breakpoint vehicle.

There are two general ways to enter FAIR-BUG; either through programmed instructions, or by manual intervention from the console. The manual entry is normally used for the following situations:

For initial program loading
For program punch and save
To start tracing a runaway program
To display or take further action following a BR *

The procedure to manually enter FAIR-BUG is to HALT, depress DEBUG switch, respond to prompt (?) with desired commands.

Programmed entry to FAIR-BUG can be achieved by building trace routines into the source program prior to assembly or else by patching the object program after loading it. In either case the following instructions provide the necessary linkage.

JMP H'8080'	This is a 3 byte instruction that destroys the accumulator and does not save the program counter.
PI H'8080'	This is a 3 byte instruction that destroys the accumulator and pushes the program counter (PC0) to the stack (PC1).
LR P0,Q	This is a 1 byte instruction that does not destroy the accumulator and does not save the program counter.
PK	This is a 1 byte instruction that does not destroy the accumulator and pushes the program counter (PC0) to the stack (PC1)

These instructions are explained in detail in the F8 USER'S MANUAL. It should be noted that all except the PI are priviledged instructions and that an interrupt cannot be serviced following these instructions. Furthermore, the first instruction executed at H'8080' is a disable interrupt which inhibits interrupts until an Enable instruction is issued.

Examination of these instructions discloses that the PK instruction is the most desirable to use since it is 1 byte, saves the accumulator, and saves the PC0; however, it does require the K register (R12 and R13) be preset. If K is not used in the program being tested this is an ideal instruction. A recommended procedure is to code into the users program at location 0 the following instruction:

```
LI  H'80'          LI H'80'
LR  KU,A    or     LR QU,A
LR  KL,A           LR QL,A
```

If one of the above housekeeping procedures are used patching the user's program for tracing or display purposes becomes an easier proposition. Instead of using a 2 byte BR* or 3 byte JMP or PI the simpler PK or LR P0,Q may be used and the prompt (?) will indicate that a desired point has been reached. Display of PC1 will then determine which (if more than one) of the trace points has been reached. The disadvantage of using PK or PI is that the object program PC1 has been lost. If this is detremental for a particular section of code being debugged the JMP or LR P0,Q instructions are available.

It can be noted that the options are numerous and the user must decide for himself which one or which combination is most desirable for his purpose.

It is suggested that the user try a small program as shown in Appendix F and utilize all the options of FAIR-BUG until he feels confident that he understands all the options. This will save much time and effort in future debug sessions.

SECTION 5 - F8S MICROMODULE SYSTEM LOGIC

There are seven major logic blocks of the micromodule system:

- F8 CPU
- F8 PSU
- F8 Static Memory Interface
- RAM and PROM memory
- Operating controls
- Memory Address Interface and Display
- Memory Data Interface and Display

A block diagram of the system is shown in Figure 7.

5.1 F8 CPU CIRCUIT: The computing power of the system is contained in the F8 CPU circuit. It also provides 16 lines of TTL compatible I/O pins. These 16 lines are from two I/O ports; one port is addressed as port 0, the other addressed as port 1. Refer to the F8 Manual for further information.

5.2 F8 PSU CIRCUIT: The F8 PSU circuit contains the FAIR-BUG program, provides two additional I/O ports, an interval timer, an external interrupt input, and interrupt control logic. The I/O ports provide an internal pullup; the I/O circuit configuration is shown on page 4.14 of the F8 Manual. The ports and interrupt vectors of the PSU are:

port 4 port 5	I/O ports
port 6	Local interrupt control
port 7	Interval timer
H'0020'	Timer interrupt vector
H'00A0'	External interrupt vector

FAIR-BUG uses port 4 for teletype interfacing; it uses port 5 only when a paper tape load is requested from a parallel source. These two ports are available to the user when FAIR-BUG is not executing--and port 5 available at all times if a parallel tape reader is not used. FAIR-BUG pin assignments are given in appendix C.

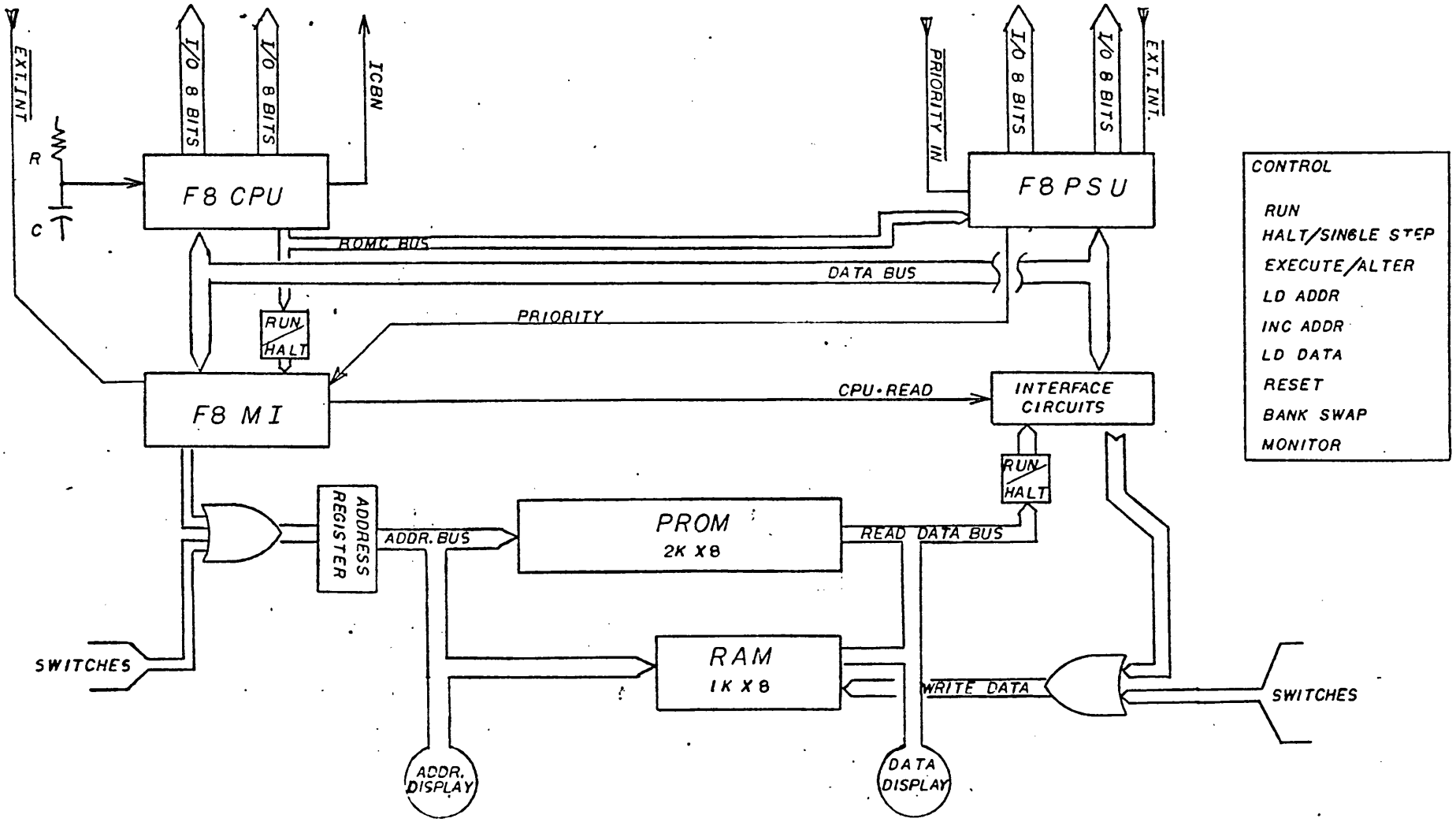


FIGURE 7: F8S PROGRAM DEVELOPEMENT MODULE

5.3 F8 SMI CIRCUIT (3853): This circuit holds a PROGRAM COUNTER, a STACK REGISTER, two DATA COUNTERS, and an ADDRESS ADDER/INCREMENTOR. It also provides a second complete interrupt level, including programmable interrupt vector and interval timer. The circuit also provides 16 bits of memory address and memory control.

The interrupt structure of the 3853 SMI is assigned lower priority than that of the 3851 PSU.

5.4 RAM and PROM MEMORY: The 1024 byte X 8 bit RAM memory consists of eight 2102 memories; each 2102 memory has 1024 X 1 bits of static read/write memory. The micromodule can execute the F8 store to memory instruction (ST).

Sockets are provided for 93446 PROM memory. PROM memory can be as large as 2048 bytes. The 93446 PROM is a fusible link device and has three-state output; its bit organization is 512 X 4. Figure 8 is a map of PROM locations on the micromodule board.

5.5 MEMORY ADDRESS CIRCUITS: The Memory Address register is made from an MSI counter which has parallel asynchronous load provisions. In EXECUTE mode the address provided by the memory interface logic is fed straight thru the register without latching. In ALTER mode the parallel inputs to the address register are from the ADDRESS switches; loading only occurs when the LOAD ADD switch is used. The counting feature of the register is used only in ALTER mode; INC ADD and LOAD DATA cause the register to count.

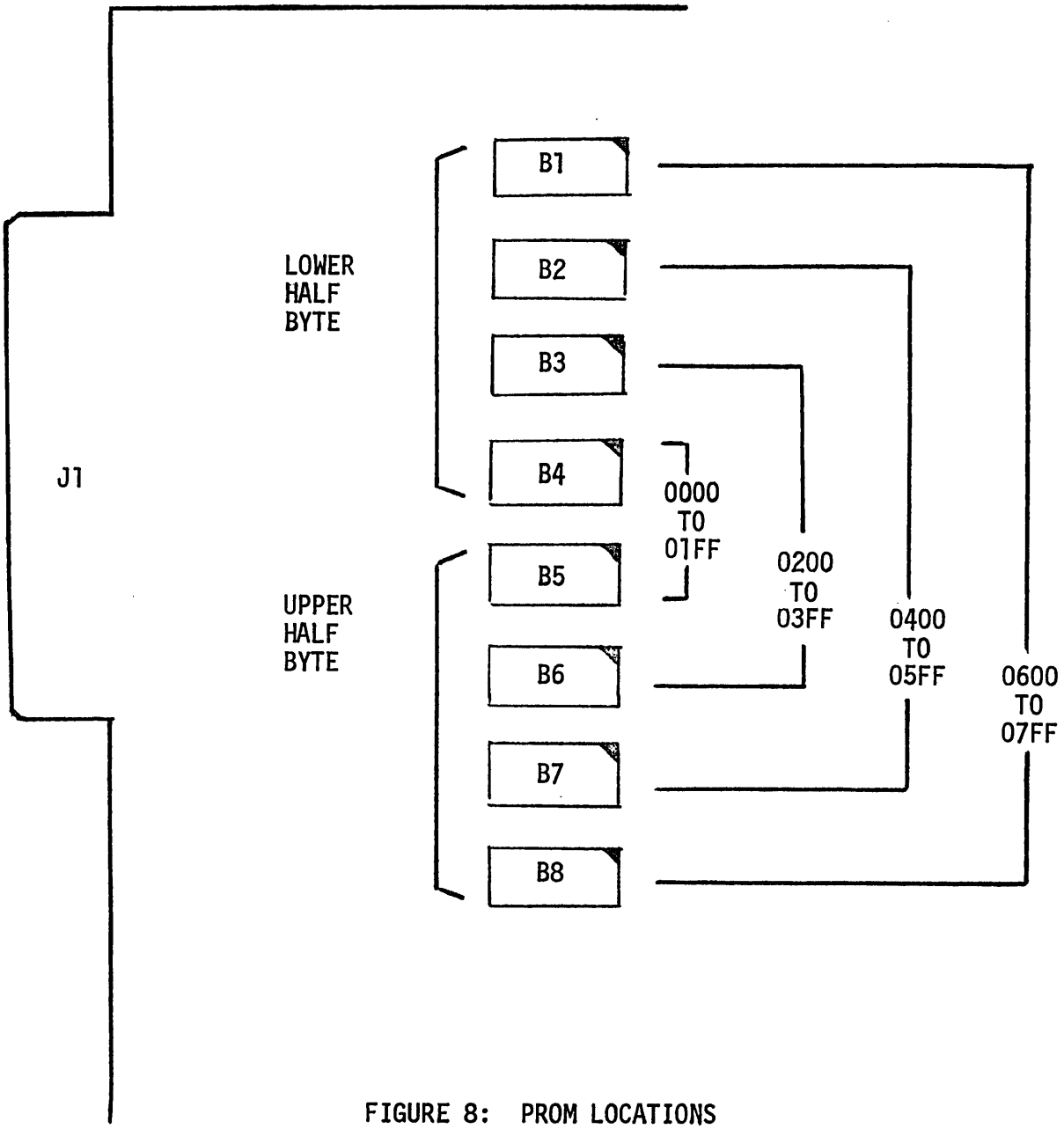


FIGURE 8: PROM LOCATIONS

SECTION 6 - ADJUSTING SYSTEM SPEED

The system clocks are generated by an RC oscillator that is part of the 3850 CPU circuit. The clock frequency is set by an external RC network. There is a variable resistor in the RC network; this resistor can be adjusted with a screw driver to vary system speed over the range 1 MHz to 2 MHz. The capacitor can be changed to a larger value if a speed outside the adjustment range is needed. The capacitor is component C6.

When adjusting system speed, observe the PHI clock at J2-K or pin 1 of the 3850 CPU. The clock must be set at 2MHz (500ns period) when the FAIR-BUG monitor is used; the data rate for the teletype is derived from the PHI clock. There will be some variation of clock rate if there are variations in power supply levels.

SECTION 7 - TURNING POWER ON AND OFF

7.1 The micromodule requires +12 volt and +5 volt power. To insure that the RC oscillator starts, the +5 volt power should be turned on before or at the same time as the +12 volt power. However, no damage will occur if this guideline is violated. A test for whether the oscillator is running is to see if the RUN LED is correctly responding to the RUN and HALT switches.

7.2 There are no restrictions on the power down procedure.

7.3 The micromodule system will execute a reset when it is powered up; this will be the same reset as that executed whenever the RESET pushbutton is used. The reset will initialize the PROGRAM COUNTER, the Interrupt Control Bit of the CPU, and initialize the state of the system to fetch the first program instruction. All the rest of the registers will need to be loaded by your program before any assumptions can be made about what is in those registers. For instance, your program cannot assume that a scratchpad location holds zero--the program will have to have first loaded that location with zero before making that assumption. The list of registers that may need to be initialized include:

Scratchpad locations
I/O ports
Interval timer
Interrupt control
Data Counter
Stack Register

SECTION 8 - I/O PORTS

The micromodule has 4 I/O ports and 6 registers that look like I/O ports to the F8 CPU. The two sets of interrupt structure and interval timers utilize the registers that look like I/O ports. The I/O ports of the system are:

Port Address	Which Circuit	Function
0	3850 CPU	I/O
1	3850 CPU	I/O
4	3851 PSU	I/O
5	3851 PSU	I/O
6	3851 PSU	Intrpt. Cont. (load only)
7	3851 PSU	Timer (load only)
E	3853 SMI	Intrpt. Cont. (load only)
F	3853 SMI	Timer (load only)
C	3853 SMI	Intrpt. Vector, Upper Byte
D	3853 SMI	Intrpt. Vector, Lower Byte

8.1 Each I/O port has 8 bits; each bit is TTL compatible and bidirectional. All I/O ports utilize the standard pull-up configuration; a logic "1" is 0 volts at the I/O pin. Each I/O pin can drive 1.25 TTL Unit Loads. Open collector TTL gates should be used for inputting to the system or some other means used to insure that the short circuit of I/O pins to V_{DD} is limited.

8.2 The 32 I/O pins, interrupt priority, and external interrupt input are brought out to the 44 pin connector J3. Appendix A gives the edge pin assignments.

8.3 FAIR-BUG uses port 4 while executing for teletype interface. It also uses port 5 but only when reading paper tape from parallel devices. At other times ports 4 and 5 are available for general usage. The FAIR-BUG pin assignments are given in Appendix C.

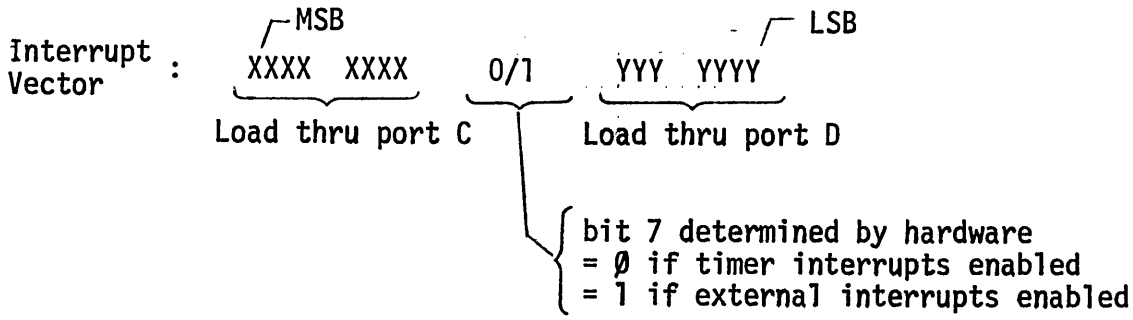
SECTION 9 - INTERRUPTS

The micromodule has external interrupt capability and the capability for interrupts from internal programmable timers. There are two independent interrupt structures in the micromodule--one in the 3851 PSU and the second in the 3853 S-MI. Each can accept an interrupt from an external source or from its own interval timer. Each can be independently programmed to be on, or to be off, and to hold different timer counts. The interrupts of the 3851 PSU are assigned higher priority than those of the 3853 S-MI.

9.1 The interrupt vector addresses are:

3851 - Timer Interrupt H'0020'
3851 - External Interrupt H'00A0'
3853 - Timer Interrupt - Programmable
3853 - External Interrupt = Timer Interrupt Vector + H'0080'

9.2 Programmable interrupt vector of 3853 S-MI: The interrupt vector of the 3853 S-MI is programmable; output instructions to ports H'0C' and H'0D' are used to load it. Fifteen bits may be programmed--the 16th bit (bit 7 of the vector) is used by hardware to distinguish between timer interrupts and external interrupts.



9.3 The Priority-In signal of the 3851 PSU is brought out to pin C of edge connector J3. The active level is Ov; the micromodule is supplied with this pin tied to Ov thru strap E8.

9.4 The external interrupt inputs for both the 3851 PSU and the 3853 S-MI are brought out to edge connector J3.

External interrupt input of 3851 PSU: J3-4
External interrupt input of 3853 S-MI: J3-16

The interrupt circuits respond to the +5V to 0 volt transistion of these signals.

9.5 Interrupts that occur while the system is halted, either in EXECUTE mode or ALTER mode, will be serviced immediately. The system will run momentarily while the interrupt vector is accessed and will then halt after the PROGRAM COUNTER has been changed but before the first instruction of the interrupt service routine has been fetched.

SECTION 10 - TELETYPE CONVERTOR CIRCUIT

The micromodule includes a teletype convertor circuit. The circuit converts the TTL level signal of the I/O ports to signals needed by a 20mA teletype such as a Teletype Corporation Model 33.

10.1 Teletype data moves serially thru two bits of an I/O port. The particular two bits used are not pre-assigned. However, FAIR-BUG has assigned bits Ø and 7 of I/O port 4 for teletype data. A teletype mark is a logic 1--which is a 0 volt level at the I/O port. Baud rate is a function of the software program.

10.2 The teletype hook-up is four wire full duplex. The teletype should be strapped for 20 milliamp loop currents. Appendix D gives directions for strapping a Teletype Model 33 for these options. All strapping options are standard Teletype options; no modifications are necessary. The teletype convertor circuit provide 20mA printer current and 10mA keyboard loop current.

10.3 The teletype connections are brought out on the 24 pin edge connector J2. Connections between J2 and a teletype are given in Appendix D ; connections between J2 and a RS-232 device are given in Appendix E . The two TTL level signals going to and from the micromodule's teletype convertor circuit are also brought out on connector J3; these should be tied across to some of the I/O pins available on connector J2.

10.4 The teletype convertor can also be to interface with devices using RS-232 compatible signals. The necessary changes of the micromodule are shown in Appendix E; the change involves strapping options, changing one resistor, and removing another. Signal levels are:

	Input	Output
Mark	$\leq$ -4V	-5V
Space	$\geq$ +1V	+7V
Impedance	3.3K	300Ω

(note: voltages with respect to RS-232 SIGNAL RETURN--pin 7. SIGNAL RETURN is at +5V with respect to VSS)

SECTION 11 - MEMORY EXPANSION/I/O EXPANSION

The F8S Memory Expander Module is available to expand the memory size and the number of I/O ports. Each memory expander provides an additional 4K of 2102 RAM, four I/O ports, and two interrupt levels. More than one expander can be used. The expanders connect to J1 and J2 of the F8S; the memory signals are on J1 while the F8 signals for driving the PSUs are on J2.

The signals on J1 and J2 are also useful for interfacing to other memories or other F8 devices. The memory signals are driven by the 3853 Static Memory Interface; memory timing is given in the 3853 data sheet. The F8 signals are driven by the 3850 CPU. The table below gives the loading of the different lines. The signal MEM ENB can be used to disable all on-board memory; memory is enabled when MEM ENB is left open.

J1

	Direction	Drive TTL U.L. Available	Capacitance Load on F8S	Logic "1" Level
Address 0 - 9	Out	9 - *	50pf	$\leq$ 2.4
Address 10 - 15	Out	8	----	$\leq$ 2.4
Read Data 0 - 7	In	1 TTL Load	50pf	$\geq$ 3.5**
Write Data 0 - 7	Out	0	50pf	$\leq$ 2.4
RAM Write N	Out	0	50pf	$\leq$ 2.4
CPU Read	Out	1.25	25pf	$\leq$ 3.9

J2

Data Bus 0 - 7	In/Out	1 : Out	25pf	$\geq$ 3.5 : In
ROMC 0 - 4	Out	1	25pf	$\leq$ 3.9
PHI N	Out	6	----	$\leq$ 2.4
Write N	Out	2	----	$\leq$ 2.4

* = Prom loads @ 1/6 U.L. each

** = 22kΩ pullup provided on F8S

SECTION 12 - EXTERNAL STOP ABILITY

The F8S can be halted by grounding the signal STOP CMD (J2 - A). This ability to halt the system with a logic signal is another debugging tool. It might be that the logic signal is the control signal for some step in a process. A typical application would be to do a comparison of a desired instruction address as held in a set of switches to the current system address (available on J1); the result of this comparison could be used to halt the machine just after the desired instruction has been executed. This application is shown in Figure 9.

The signal STOP CMD is sampled into a latch at the end of every cycle. After the latch has been set, the system halt will occur at the end of the current instruction. The timing of these events is shown in Figure 10. The loading of STOP CMD is 1 TTL U.L. plus a 10kΩ pullup resistor. The signal has no effect after the system is halted.

Two system status signals and a strobe are provided on J3 and J2. RUN holds the same information as the RUN LED. RCEZN is low at the end of every instruction while the next is being fetched. The strobe is CNTRL STROBE; it has a positive going transistion 1μsec after the start of each cycle and is useful for gating decodes of the ROMC lines.

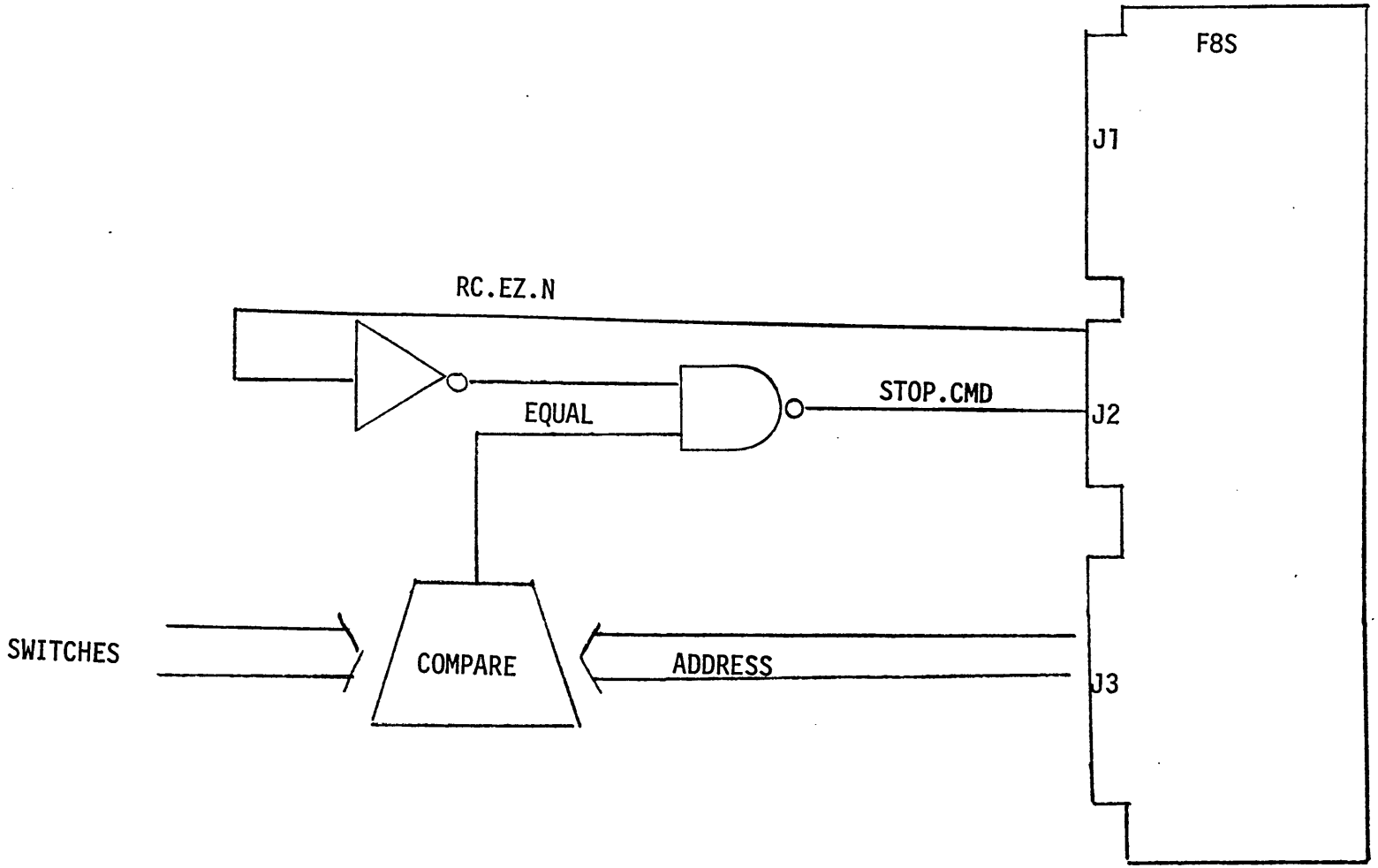


FIGURE 9: STOP ON INSTRUCTION FETCH

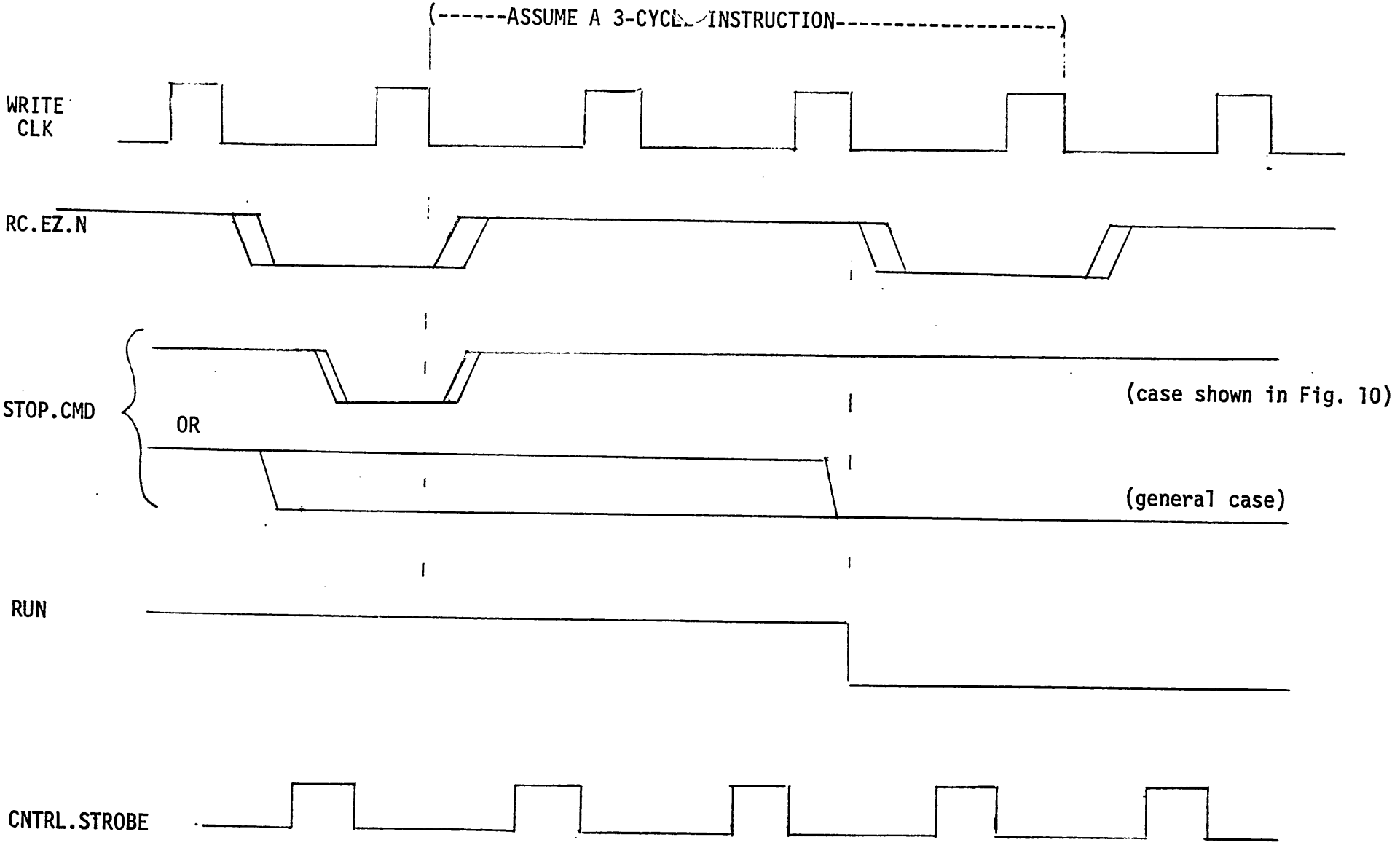


FIGURE 10: EXTERNAL STOP TIMING

SECTION 13 - MEMORY DISABLE/STANDBY

The RAM memory of the F8S may be left powered while the rest of the system is powered down. A separate VCC power lead is brought to just the 2102 RAMs and the chip select circuitry. Also the chip selects can all be forced into the disabled state by the signal MEM ENB (J1 - M). The memory should be disabled first and then system power dropped.

Memory in standby:

MEM ENB (J1 - M) $\leq$ 0.8V (1 TTL U.L.)
RAM +5 (J3 - B) : 5V @ 500mA max

Memory and system active:

MEM ENB (J1 - M) open or +5 volts (10kΩ pullup provided)
RAM +5 (J3 - B) : 5V @ 500mA max

SECTION 14 - ROM VERIFICATION

The F8S may be used to check the program in a PSU--such as for instance in the PSU that you have developed with the F8S for your system. The available controls are RESET, RUN, HALT/S.STEP; the EXECUTE/ALTER switch should be in EXECUTE position. The Address display is valid, the data displays are not valid. The FAIR-BUG PSU should be removed and your's inserted in its place. Strap E10 should be connected and strap E11 removed; these strap changes disable the 3853 SMI and the PROM/RAM memory.

SECTION 15 - INSTALLATION OF F8S AND TELETYPE CABLING TO RUN FAIR-BUG

This section draws together all the pieces if what you'd like to do with the F8S is just to connect it to a teletype and be able to run FAIR-BUG.

1. Requirements:

   F8S

   +5V power supply at 2.5A

   +12V power supply at 1/2A

   44 pin connector (J3) (Amphenol 225-2221-401)

   24 pin connector (J2) (Cinch 50-24A-30)

   teletype - 20ma loop, full duplex, model 33 ASR or equivalent (see Appendix D if there options are not strapped)

2. Power Connections:

   +12V J3 - N

   +5V J3 - 1,A,B

   gnd J3 - 22,Z

3. Teletype Interface to I/O Connections:

   J2 - D to J3 - E

   J2 - 3 to J3 - 3

4. Teletype to F8S Connections:

F8C Connector J2		Teletype Terminal Strip
J2-J	Keyboard	TS-4
J2-7	Keyboard Return	TS-3
J2-H	Printer	TS-7
J2-8	Printer Return	TS-6

5. If the teletype has been modified by the addition of a relay in the reader control circuit (as for DEC or Nova computer terminals), either energize the relay or tie its contacts together.

6. Factory Supplied Straps:

   These straps are supplied connected by traces:

   S3 - teletype circuit

   S8 - priority in : by 3851 PSU

   S11 - DBDRN

These straps are supplied open:

S1 - teletype circuit
S2 - teletype circuit
S10 - DBDRN

Pin J1 - M (MEM ENB) should be open.

7. Teletype Baud Rate:

	110 baud	300 baud
J3 - 6	open	to J3 - Z (0V)
J3 - H	open	open

8. Observe the PHI clock of the CPU at J2-K and adjust trimpot for a 500 $\pm$ 10 nsec period (2MHz).

APPENDIX A: Pin Assignments of Connectors

Connector J1 36 pins: memory expansion signals

1	Addr ØØ	A	Addr 04
2	Read data Ø	B	Addr 02
3	Addr 01	C	Write data Ø
4	Addr 06	D	Addr 03
5	Read data 1	E	Write data 1
6	Addr 05	F	Write data 2
7	Read data 2	H	Addr 9
8	Read data 3	J	Write data 3
9	Addr 10	K	Addr 8
10	Read data 4	L	Write data 4
11	RAM WRITE N	M	MEM ENB
12	Read data 5	N	Write data 5
13	Addr 12	P	Addr 14
14	Read data 6	R	Write data 6
15	Addr 15	S	Addr 13
16	Addr 11	T	Addr 07
17	Read data 7	U	Write data 7
18	BANK LOW	V	CPU READ

Connector Amphenol 225-21821-401 or equivalent
Key: Amphenol 225-594 or equivalent
Key should be installed between pins 3 and 4

Connector J2 24 pins: I/O expansion and teletype

1	RC.EZ.N	A	STOP.CMD.N
2	Data Bus 3	B	CNTRL.STROBE
3	TTL.TO.TTY	C	Data Bus 5
4	Data Bus 2	D	TTL FROM TTY
5	Data Bus 4	E	Data Bus 7
6	Data Bus Ø	F	Data Bus 6
7	KYBD RETURN	H	PNTR
8	PNTR RETURN	J	KYBD
9	WRITE.N	K	PHI.N
10	Data Bus 1	L	ROMC4
11	ROMC 2	M	ROMC3
12	ROMC 1	N	ROMCØ

Connector: Cinch 50-24A-30 or equivalent
Key: Cinch 50-5K-2 or equivalent
Key should be installed between 3 and 4.

Connector J3 44 pin: I/O, interrupts, and power

1	+5 pwr	A	+5 pwr
2	RUN	B	RAM +5
3	I/O 4ØN	C	PRI.01.N
4	EXT.INT.Ø.N	D	I/O 50N
5	I/O 57N	E	I/O 47N
6	I/O 41N	F	I/O 51N
7	I/O 52N	H	I/O 42N
8	I/O 43N	J	I/O 53N
9	I/O 54N	K	I/O 44N
10	I/O 45N	L	I/O 55N
11	I/O 56N	M	I/O 46N
12	I/O 03N	N	+12 pwr
13	I/O 12N	P	I/O 13N
14	I/O 00N	R	I/O 10N
15	I/O 11N	S	I/O 01N
16	EXT.INT.1.N	T	I/O 02N
17	INT.REQ.N	U	I/O 07N
18	I/O 16N	V	I/O 17N
19	I/O 05N	W	I/O 06N
20	I/O 14N	X	I/O 15N
21	RESET N	Y	I/O 04N
22	0V	Z	0V

Connector: Amphenol 225-2221-401 or equivalent.
Key: Amphenol 225-594 or equivalent.
Key should be installed between pins 3 and 4.

APPENDIX B: Power Requirements and Physical Size

Power Requirements

+5V ± 5% @ 2.0 amps max

RAM +5V ± 5% @ 500mA max

+12V ± 5% @ 50mA max

Power is supplied through connector J3

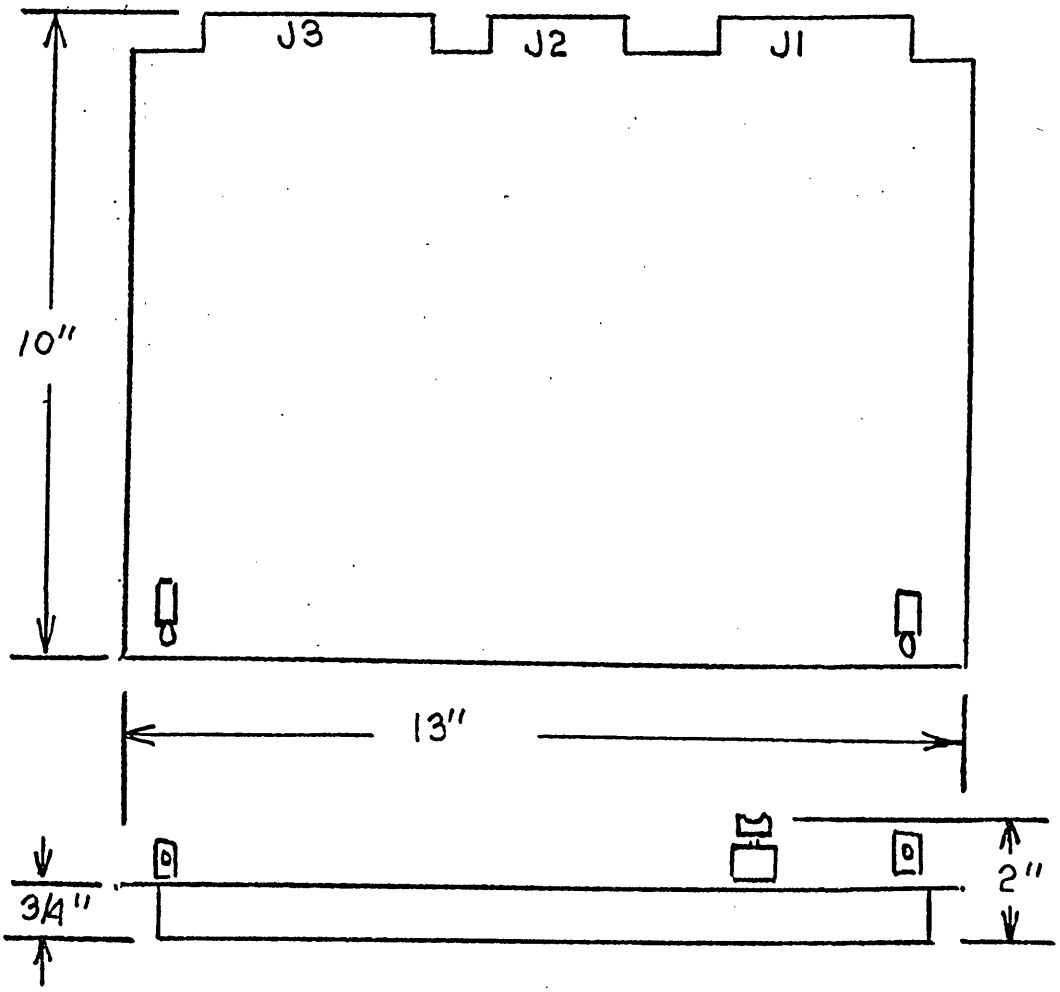
+5V: J3-1 and J3-A

RAM +5V: J3-B

+12V: J3-N

0V: J3-22 and J3-Z

Physical Size



APPENDIX C
FAIR-BUG PORTS ASSIGNMENT

Assignments for Port 4 are:

Connector J3	Bit	Function
J3 - E	7	Serial input
J3 - M	6	Character Ready (Parallel Device)
J3 - 10	5	---
J3 - K	4	Device Ready (Parallel Device)
J3 - 8	3	Step Reader (Parallel Device)
J3 - H	2	bit 2 / bit 1 / Teletype Baud Rate / Input source during tape load
J3 - 6	1	
		0 / 0 / 110 baud / teletype
		0 / 1 / 300 baud / teletype
		1 / 0 / Baud delay in memory (location 3FF or BFF) / parallel source (port 5)
		1 / 1 / Baud delay in memory / teletype
J3 - 3	0	Serial Output

Assignments for Port 5 are:

Connector J3	Bit	Function
J3 - 5	7	Parallel input byte - MSB
J3 - 11	6	Parallel input byte
J3 - L	5	Parallel input byte
J3 - 9	4	Parallel input byte
J3 - J	3	Parallel input byte
J3 - 7	2	Parallel input byte
J3 - F	1	Parallel input byte
J3 - D	0	Parallel input byte - LSB

APPENDIX D: Connecting a Teletype Model 33

The micromodule system can easily interface with a Teletype Model 33 teleprinter. The teletype convertor of the micromodule provides signals for a 20mA full duplex loop.

The recommended teletype is a Teletype Model 33 ASR with automatic reader on/off control. Other Model 33 Teleprinters can also be used.

Teletype Strapping Options

No modifications of the teletype are necessary. Strapping options should be selected to provide 20mA loop currents in place of 60mA loops, and to provide full duplex operation in place of half duplex operation. The options are described in the following paragraphs.

1. Parts location: All option points are on the teletype power supply assembly. The power supply assembly is rightmost in the teletype; prominent are the LINE/OFF/LOCAL switch in the front of it, and a row of three fuse holders in the rear. Changes are made on a ten terminal strip (part # 151411) that is at the lower rear of the power supply assembly. The other change is made on a large flat multi-tap power resistor (part # 181816) that is about three inches behind the LINE/OFF/LOCAL switch.

2. Select 20mA loop currents by performing Note 2 of the TTY Drawing Number 6353WD which states: "For the .020 amp neutral signal line move the purple wire from Terminal 8 to Terminal 9 of the 151411 terminal strip. Also move the blue wire from Terminal 3 of the power resistor 181816 to Terminal 4."

3. Select full duplex operation by performing Note 3 of TTY Drawing Number 6353WD which states: "Move the white-blue wire from Terminal 4 to 5 and the brown-yellow wire from Terminal 3 to 5 on the 151411 terminal strip."

   Caution: The 110V line cord terminates on the terminal strip. Unplug power cord from the AC source before working on the teletype.

Teletype - Micromodule Connections

Teletype Model 33 machines provide two alternative places for attaching an interface cable. One location is the 10 terminal strip that is at the rear of the power supply assembly (the same place as where the option changes were made). The optional location is at the 15 pin connector #2 which is just above the terminal strip. One mating plug for connector #2 is:

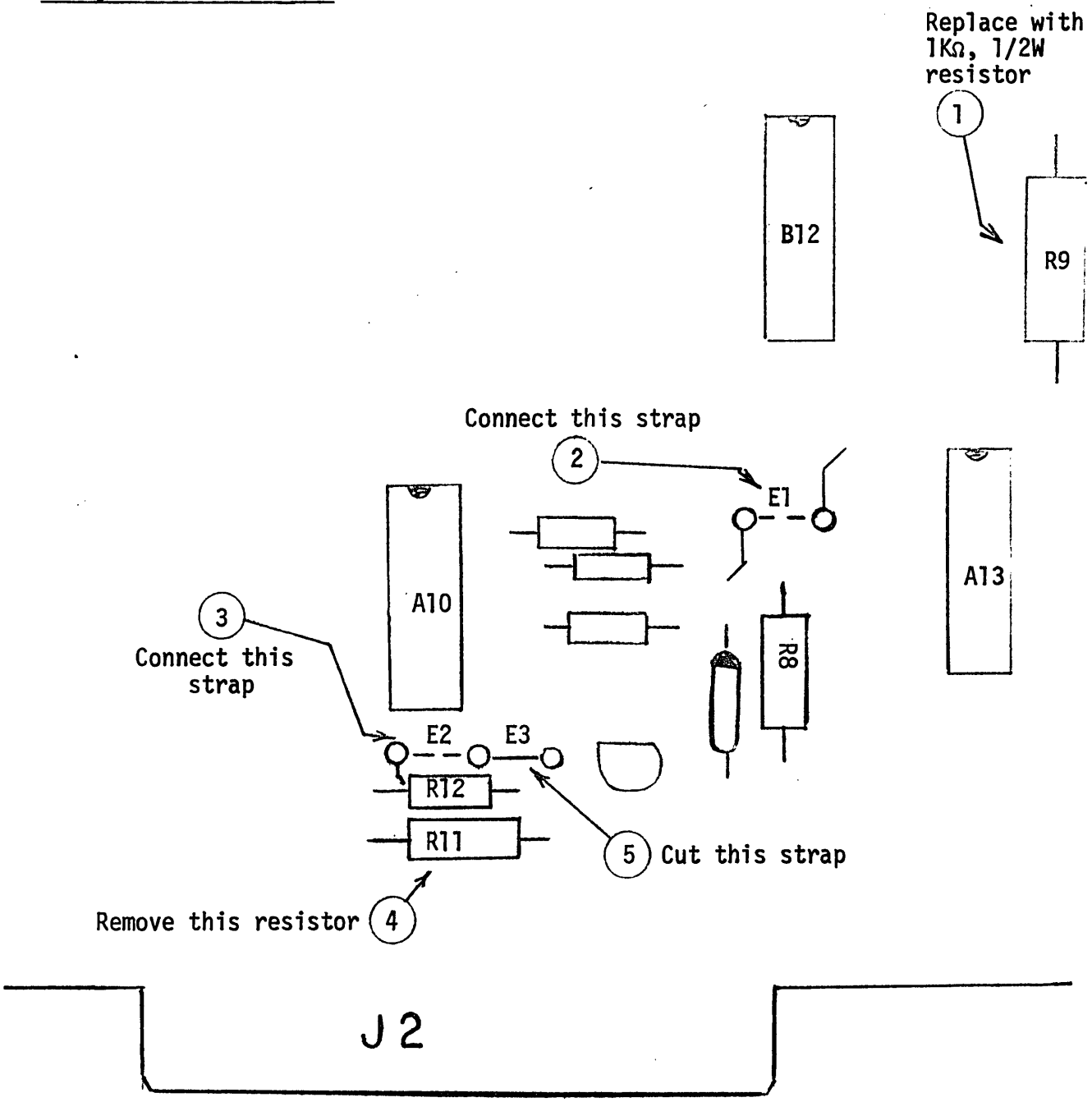
MOLEX part # P(03-09-2151) housing
with MOLEX part # (02-09-2118) terminals

The connections between Teletype and Micromodule are:

Teletype				Micromodule
TS-4	or	J2-6	Keyboard	J2-J
TS-3	or	J2-5	Keyboard Return	J2-7
TS-7	or	J2-8	Printer	J2-H
TS-6	or	J2-7	Printer Return	J2-8

APPENDIX E: EIA OPERATION

Changes on Micromodule



Connections

Micromodule		EIA Connector	
J2-7 (KYBRD)	XMIT DATA	pin 2	
J2-8 (PNTR RTRN)	RCV DATA	pin 3	
J2-H (PNTR)	SIGNAL RETURN	pin 7	
----------	CLEAR TO SEND	pin 5	tie together
----------	DATA SET READY	pin 6	
----------	DATA TERM READY	pin 20	

APPENDIX F

FAIR-BUG EXAMPLE 1

```
?M0
 M0000=00
?C 70
?N M0001=91
?C 0B
?N M0002=00
?C 5C
?N M0003=DD
?C 1F
?N M0004=04
?C 25
?N M0005=10
?C 3F
?N M0006=00
?C 94
?N CF9 M0007=D1
?C F9
?N M0008=00
?C 29
?N M0009=10
?C 80
?N M000A=00
?C 80
```

```
        STORE A PROGRAM TO SET
        SCRATCHPAD TO 0-3F.

              LIS 0
LOOP    LR  IS,A
        LR  S,A
        INC
        CI  H'3F'
        BNZ LOOP
        JMP H'8080'
```

```
?M0-A
 M0000=70 0B 5C 1F 25 3F 94 F9
 M0008=29 80 80 D0 00 10 00 91
```
DISPLAY PROGRAM

```
?R0-3F
 R0000=A4 FF 09 FF 00 00 FF 00
 R0008=83 0A 00 FF 81 97 03 FF
 R0010=0C 20 13 00 1E 00 BF 00
 R0018=9D 40 7D 01 DD 17 55 00
 R0020=B7 F7 7F A2 FF 0E FF 22
 R0028=FF 76 FF 3C FF CE 5F 20
 R0030=18 04 02 00 D9 04 7F 00
 R0038=75 01 57 4A 0F 0A 0A FF
```
DISPLAY SCRATCHPAD
BEFORE EXECUTION

```
?GO
```
GO TO LOC 0 TO EXECUTE

```
?M7
 M0007=F9
```
(PROGRAM LOOP ERR.) MANUAL RESET TO FAIR-BUG

```
?C FA
```
CORRECT BNZ INSTRUCTION

```
?GO
```
GO TO 0

```
?R0-3F
 R0000=00 01 02 03 04 05 06 07
 R0008=80 09 0A 0B 0C 0D 0E 0F
 R0010=10 11 12 13 14 15 16 17
 R0018=18 19 1A 1B 1C 1D 1E 1F
 R0020=20 21 22 23 24 25 26 27
 R0028=28 29 2A 2B 2C 2D 2E 2F
 R0030=30 31 32 33 34 35 36 37
 R0038=38 39 3A 3B 3E 09 09 0B
```
DISPLAY REGISTERS AFTER
EXECUTION.
NOTE; R8, R3C-3F ARE USED
BY FAIR-BUG

```
?P0 0000
```
PC NOT SAVED BY JMP

```
?PI EEEE
?M8
 M0008=29
```
CHANGE JMP TO PI

```
?C28
?GO
```
EXECUTE AGAIN

```
?P0 000B
```
PC NOW SAVED!

```
?
```

APPENDIX F

FAIR-BUG EXAMPLE 2

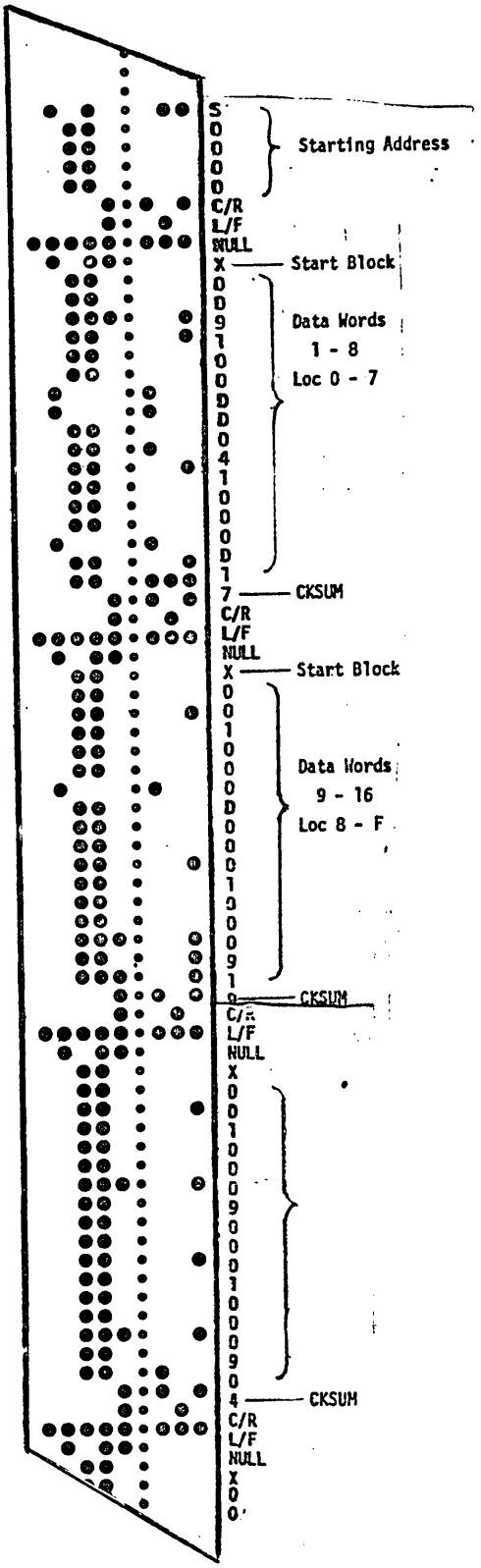
```
 R5
 R0005=00
?R1-2
 R0000=A4 FF 09 00 00 00 FF 00   R1-2 TYPES R0-R7
?N R0008=83                      NEXT=R8
?C55                             CHANGE R8 TO 55
?E R0008=55                      EXAMINE R8
?R8
 R0008=55
?P0 83B0
?D0 807B
?D1 0000
?CFFFF                           DISPLAY DC1, CHANGE, EXAMINE,
?E FFFF                          THEN DISPLAY AGAIN
?D1 FFFF
?I=0F                            ISAR
?S=0A                            STATUS
?W=0A
?M3E0-3FF
 M03E0=04 00 00 10 00 00 00 83
 M03E8=B0 EE EE 80 7B FF FF A4   MEMORY DUMP
 M03F0=FF 09 00 00 00 FF 00 55
 M03F8=0A 00 FF 81 97 03 FF 00
?R0-F
 R0000=A4 FF 09 00 00 00 FF 00   REGISTER DUMP
 R0008=55 0A 00 FF 81 97 03 FF
?R10-40
 R0010=00 20 13 00 1E 00 BF 00
 R0018=9D 40 7D 01 DD 17 55 00
 R0020=B7 F7 7F A2 FF 0E FF 22   REGISTER DUMP
 R0028=FF 76 FF 3C FF CE 5F 20
 R0030=18 04 02 00 D9 04 7F 00   NOTE:  R40-R47 IS FAIR-BUG USE
 R0038=75 01 57 4A 0F 0A 0A FF
 R0040=A4 FF 00 00 44 00 47 EF
?B0-100-00F0BONOBONOBONOFONOFONOF000BNNOBMBM@O@O@M@OCM@M@M@O@MAO@MBO@OFM@NNOBL
                                 PUNCH FORMAT (PROM TAPE)
                                 SEE APPENDIX C FOR FORMAT
?F0-FFS0000                      PUNCH FORMAT (LOAD TAPE)
X009100DD041000D17
X001000D0001000919
X00100090001000904
X000000D1141002D47

                                 SEE APPENDIX B FOR FORMAT
X24DA20C1AFFB20FFF
X26C802F3268164FBD
X20D900FF24F900F9C
X26F900FF2ED92EF58
*                                * (END OF TAPE)
?F0-FF
?A=83                            DISPLAY, CHANGE, EXAMINE ACCUMULATOR
?C00
?E R0008=00
?N R0009=0F                      NOTE:  ACCUMULATOR IS IN R8 THEREFORE NEXT IS R9
?G8080                           GO TO 8080
?PC 00FF
?P0 8080                         PC0 CHANGED TO 8080
?G                               GO TO PC0 (8080)
?
```

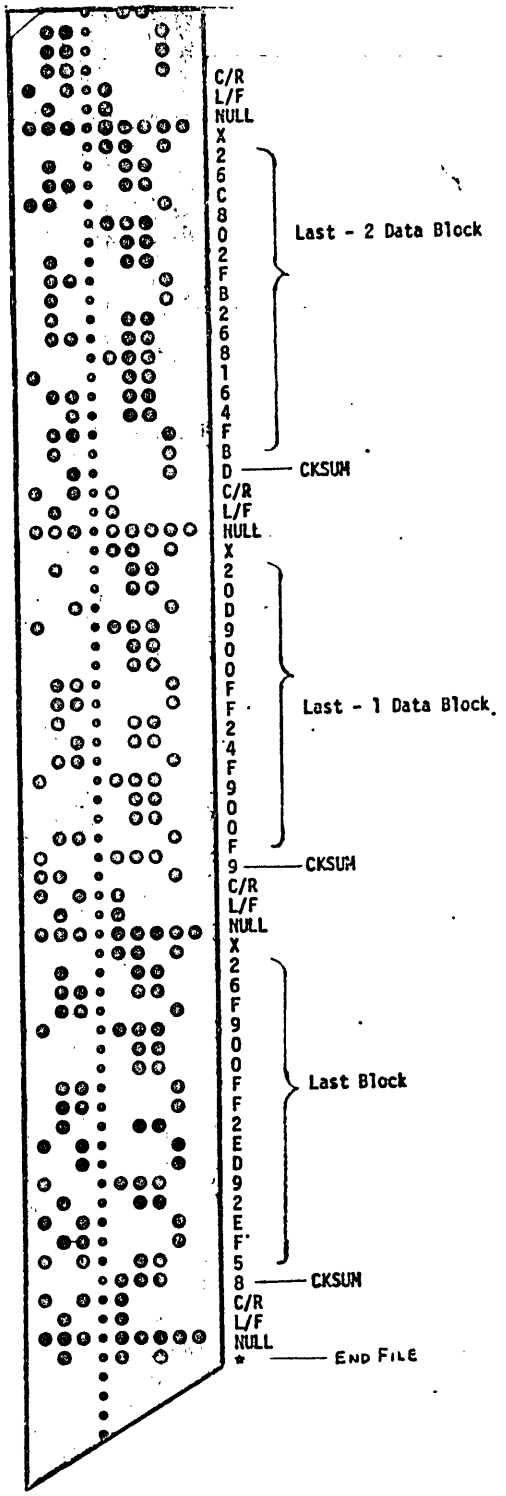
APPENDIX G

FORMATTED TAPE

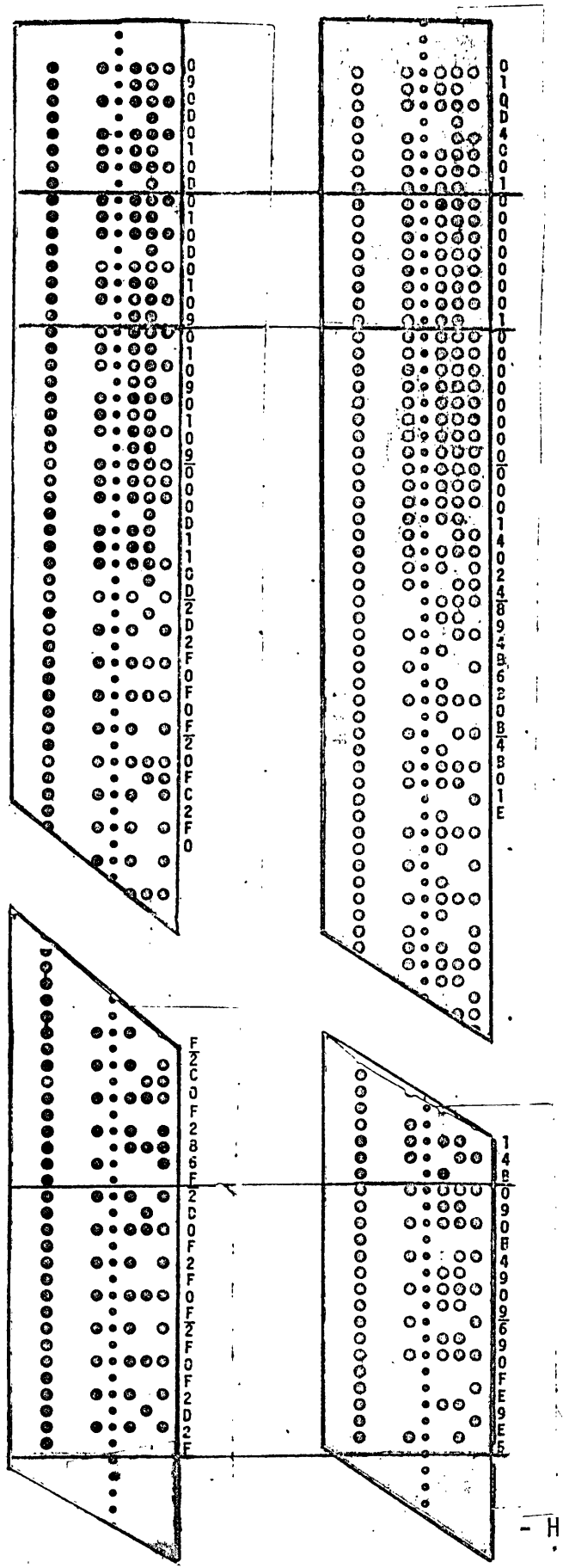
(LOADER FORMAT)



Note: This example was punched by the instruction shown in Appendix A (F0-FF). This is ASCII 7 bit format.



APPENDIX H
BINARY FORMAT
(PROM GENERATION FORMAT)



Note: This example was punched by the instruction shown in Appendix A (B-100-0).

Only the low order 4 bits are significant and are punched in compliment form.

The first block is punched with 256 memory locations hi order bits. The next block has 256 memory locations low order bits.

Subsequent blocks then alternate hi, low, hi, low, etc. with a blank gap between.

The table below shows examples of commands and the results produced.

Input Command B-XX00-YY00-L	Decimal Memory Addresses Punched	# Blocks	Block Length
B 0-100-0	0-255	2	256
B 0-400-0	0-1023	8	256
B 100-200-0	256-511	2	256
B 0-400-1	0-1023	4	512
B 0-1000-1	0-4095	16	512
B 200-400-1	512-1023	2	512

APPENDIX I
FAIR-BUG SUBROUTINES

The following INPUT and OUTPUT subroutines exist in FAIR-BUG and may be called by the users program. All subroutines may be entered by: (PI Address).

TTYI - Input 1 byte from TTY type device, without echo. Data is 11 bits/character being received on Port 4 Pin 7.

Address: H'83AD'
Enter: R0 - Delay Counter
Exit: W Reg - Destroyed
PC1 - User return address
Accum - Input byte
R0 - Unchanged
R1 - Input byte
R2 - -1

TTY0 - Output 1 byte to TTY type device. Data transmitted is 11 bits/character being output on Port 4 Pin 0.

Address: H'83E5'
Enter: R0 - Delay Counter
R1 - Byte to output
Exit: W Reg - Destroyed
PC1 - User return address
Accum - 0
R0 - Unchanged
R1 - -1
R2 - 0

TTCR - Output CR/LF/NULL to TTY type device; subroutine TTY0 is called.

Address: H'83D6'
Enter: R0 - Delay Counter
Exit: W Reg - Destroyed
PC1 - H'83E4'
Accum - 0
K Reg - User return address
R0 - Unchanged
R1 - -1
R2 - 0

PINP - Input 1 byte from parallel input device; minimum delay between characters is 150µsec. Byte is received on Port 5 with control bits on Port 4, pins 3, 4, and 6.

Address: H'8397'
Enter: No setup
Exit: W Reg - Destroyed
PC1 - User return address
Accum - Input byte
R1 - Input byte

BYTE - Input 2 ASCII hexidecimal characters and convert to 1 byte; also accumulate the checksum. If input is not ASCII characters 0-9 or A-F meaningless results will be returned. Either TTYI or PINP is called as input routine.

Address: H'837B'
Enter: Q - H'8397' (for parallel input)
Q - H'83AD' (serial input) R0 = Delay Counter
R7 - Previously accumulated checksum
Exit: W Reg - Destroyed
PC1 - Destroyed
Accum - Input byte
K - User return address
Q - Unchanged
R0 - Unchanged
R1 - Destroyed
R2 - -1 (if serial IP), unchanged for parallel IP
R7 - Checksum
R8 - 0
R11 - Input byte

FOP1 - Output byte of data from memory to TTY type device using TTYO subroutine. Byte is converted to 1 or 2 ASCII hexidecimal characters.

Address: H'80ED'
Enter: R0 - Delay Counter
R8 - Flag Pos# = OP Hi 4 bits, then Lo 4 as ASCII
Neg# = OP Lo 4 bits as ASCII
DC0 - Memory address of data
Exit: W Reg - Destroyed
PC1 - Destroyed
Accum - Destroyed
DC0 - DC0 + 1
K Reg - User return address
QL - Data byte
R0 - Unchanged
R1 - -1
R2 - 0
R7 - Checksum (low 4 bits significant)

FOP2 - Output byte of data from QL. Same routine as FOP1 except DC0 is not used.

Address: H'80EE'
Enter: R0 - Delay Counter
R8 - Same as FOP1
QL - Data byte to output
Exit: Same as FOP1

APPENDIX J
ASCII CHARACTER CODES

Character	7 Bit Hex Code	Character	7 Bit Hex Code	Character	7 Bit Hex Code
(Space)	20	0	30	H	48
!	21	1	31	I	49
"	22	2	32	J	4A
#	23	3	33	K	4B
$	24	4	34	L	4C
%	25	5	35	M	4D
&	26	6	36	N	4E
' (Quote)	27	7	37	O	4F
(	28	8	38	P	50
)	29	9	39	Q	51
*	2A	:	3A	R	52
+	2B	;	3B	S	53
, (Comma)	2C	>	3C	T	54
	2D	=	3D	U	55
.	2E	<	3E	V	56
/	2F	?	3F	W	57
		@	40	X	58
Line Feed	0A	A	41	Y	59
Carriage RTN	0D	B	42	Z	5A
Bell	87	C	43	[	5B
Punch ON	92	D	44	\	5C
Punch OFF	94	E	45	]	5D
Reader ON	91	F	46	↑	5E
Reader OFF	93	G	47	←	5F
Null	7F				
Null	FF				

F8S EXPANSION MODULE
USER'S MANUAL

FAIRCHILD
SEMICONDUCTOR

AUGUST 15, 1975

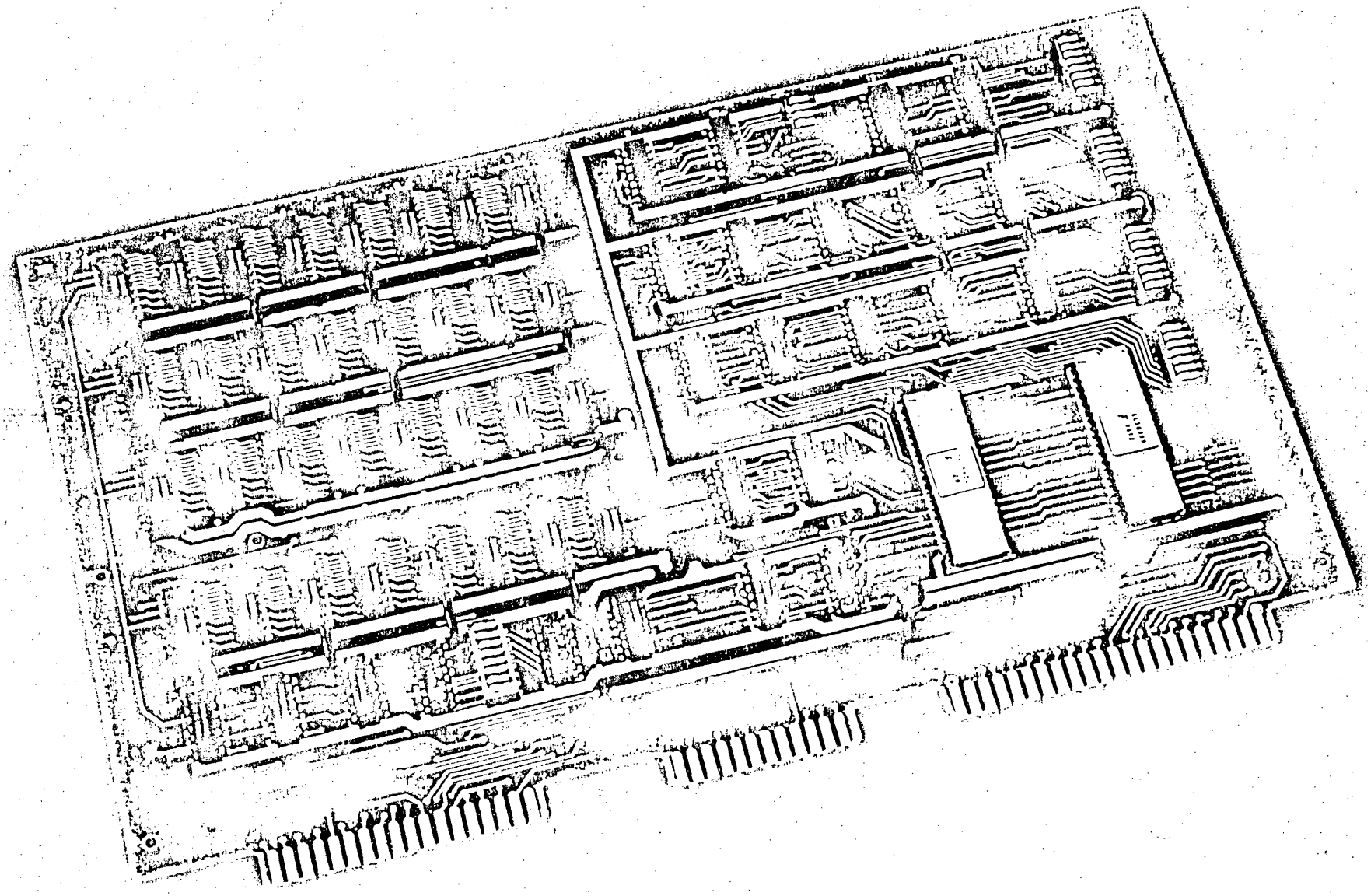


TABLE OF CONTENTS

SECTION 1 - F8S EXPANSION MODULE FEATURES

The F8S Expansion Module features provide for easy expandability of an F8 program development system.

- 4K bytes of Static RAM
- Memory page selection permits expansion to 65K bytes by 4K increments
- 32 I/O pins, TTL compatible
- Programmable I/O addresses permit expansion to 18 I/O ports or 144 I/O pins
- Two Interval Timers
- Two independent levels of interrupt
- Programmable Interrupt Vectors

SECTION 2 - MEMORY ORGANIZATION

The 4K bytes of Static RAM are arranged as a continuous block of 4096 memory locations; the starting address of this block, or page of memory, can be selected anywhere in the 65K space by multiples of 4096 addresses, starting from Zero. Each F8S Expansion Module's page selection is programmable by DIP switches. The page selection switches referenced as SW5 on the schematic diagram, location A6 on the PCB (see Figure 1). The following table gives the 4K page starting address as a function of the switches setting.

Starting Address	SW5 Switches Setting X indicates Closed, 0 indicates Open								
Hexedecimal	1	2	3	4	5	6	7	8	
0 0 0 0	X	0	0	0	X	0	0	0	
1 0 0 0	X	0	0	0	0	X	0	0	
2 0 0 0	X	0	0	0	0	0	X	0	
3 0 0 0	X	0	0	0	0	0	0	X	
4 0 0 0	0	X	0	0	X	0	0	0	
5 0 0 0	0	X	0	0	0	X	0	0	
6 0 0 0	0	X	0	0	0	0	X	0	
7 0 0 0	0	X	0	0	0	0	0	X	
8 0 0 0	0	0	X	0	X	0	0	0	
9 0 0 0	0	0	X	0	0	X	0	0	
A 0 0 0	0	0	X	0	0	0	X	0	
B 0 0 0	0	0	X	0	0	0	0	X	
C 0 0 0	0	0	0	X	X	0	0	0	
D 0 0 0	0	0	0	X	0	X	0	0	
E 0 0 0	0	0	0	X	0	0	X	0	
F 0 0 0	0	0	0	X	0	0	0	X	

Table 1
Memory Page Selection

Note: FAIR-BUG occupies H'8000' to H'83FF'.

2.1 Memory Disable Option: A signal MEM. N is available at the connector J1-M to permit external enable/disable control of the 4K bytes of RAM.

MEM. N	HIGH	MEMORY DISABLED
MEM. N	LOW (0V)	MEMORY ENABLED

A strapping option referenced as E1 on the PCB permits enabling the memory permanently without the use of an external control.

If starting address H'0000' is chosen for the first page of 4K RAM, precaution should be taken to disable the memory of the F8S Development Module. This can be achieved simply by connecting Pin M of connector J1 on F8S Development Module to Pin M of connector J1 on F8S Expansion Module. Enabling the Expansion Module memory will disable the Development Module memory. This precaution is not necessary if the starting address is different than Zero. The F8S Development Module and F8S Expansion Module memories can be enabled both at the same time. Refer to page 2-6 of F8S User's Manual for switching of memory allocation.

Note that the FAIR-BUG memory space requirement remains the same even if the F8S Development Module memory is disabled.

2.2 Memory Standby Option: The power to the 4K bytes of RAM is provided by a separate pin at the connector: J3--pin B. This permits the use of a standby 5V supply independent from the 5V logic supplied at connector J3, pins 1 and A.

The power down procedure to standby will consist of a memory disable followed by removal of the 5V logic supply.

SECTION 3 - I/O PORT ORGANIZATION

Two 3851 PSU devices are supplied with F8S Expansion Module. The I/O port addresses of each of these two PSU's are programmable by the DIP switch referenced SW6 at PCB location E11, see Figure 1. The following table gives the possible addresses and corresponding switch settings.

PSU Location A/B - 11/12

SW6 Switches Setting X indicates Closed O indicates Open		I/O A	I/O B	Interrupt Control	Timer
1	3	Hexadecimal Address			
0	0	4	5	6	7
		Not to be used with F8S Development Module.			
0	X	14	15	16	17
X	0	24	25	26	27
X	X	34	35	36	37

PSU Location A/B - 13

SW6 Switches Setting X indicates Closed O indicates Open		I/O A	I/O B	Interrupt Control	Timer
2	4	Hexadecimal Address			
0	0	8	9	A	B
0	X	18	19	1A	1B
X	0	28	29	2A	2B
X	X	38	39	3A	3B

Table 2

I/O Port Address Selection

SECTION 4 - INTERRUPT ORGANIZATION

The F8S expansion module provides two levels of interrupt: the priority chain is established such that the PSU at location A/B - 11/12 is the first in the priority chain; its External Interrupt line is available at connector J3 pin 4. The second PSU at location A/B - 13 is next in the priority chain; its External Interrupt line is available at connector J3 pin 16. In a system using one expansion module in conjunction with an F8S development module the priority chain will be established as follows:

1) Strap Priority In of the Expansion Module to ground by use of strapping option E2 which is located above location A10.
2) Connect Priority Out of the Expansion Module (connector J3 pin 21) to Priority In at F8S Development Module (connector J3 pin C).
3) Cut strapping option E8 of F8S Development Module, located next to IC at location A13. This opens the ground connection to the PSU at location A/B - 15 in the F8S.'
4) Connect Interrupt request (J3 pin 17) of Expansion Module to J3 pin 17 of F8S.

If more than one Expansion Module are used in a system and if the interrupt levels of these modules are to be used, the priority in and priority out of each module should be daisy chained to the next module, the F8S development module always being the last one in the chain. The Interrupt Request line of each module are wire-ORed together with the F8S development module.

SECTION 5 - INTERRUPT VECTOR PROGRAMMING

The two interrupt levels described in Section 4 correspond to the two Interrupt Vector Addresses that are generated by the PSU when the appropriate interrupt is serviced. These interrupt vectors are switch programmable by the DIP switches identified as SW1, SW2, SW3 and SW4.

SW1 and SW2, respectively at locations E14 and D14, correspond to the interrupt vector of the PSU at location A/B - 11/12. SW3 and SW4, respectively at locations C14 and B14, correspond to the interrupt vector of the PSU at location A/B - 13. The switch programming is described in the following table.

Interrupt Vector Upper Bits	Switch Setting		
	SW1 or SW3	Open	Closed
15	1	1	0
14	2	1	0
13	3	1	0
12	4	1	0
11	5	1	0
10	6	1	0
9	7	1	0
8	8	1	0

Interrupt Vector Lower Bits	Switch Setting		
	SW2 or SW4	Open	Closed
7	Not Switch Programmable		
6	1	1	0
5	2	1	0
4	3	1	0
3	4	1	0
2	5	1	0
1	6	1	0
0	7	1	0

Table 3

Interrupt Vector Programming

SECTION 6 - INSTALLATION OF EXPANDER WITH F8S

6.1 Physical Size: See Figure 2.

6.2 Physical Mounting: The F8S memory expander is designed to mount above the F8S as shown in Figure 3. Holes in the expander align with those in the F8S so that standoffs may be used to tie the boards together. The controls of the F8S are not covered by the expander.

6.3 Power Requirements: +5V ± 5% @ 100mA max.
RAM +5V ± 5% @ 1.6 A max.
+12V ± 5% @ 50mA max.

6.4 Electrical Connections: The F8S memory expander has three printed circuit connectors on its edge--a 44 pin connector that carries power and I/O signals, a 24 pin connector that carries I/O expansion signals, and a 36 pin connector that carries memory expansion signals. The connector types and pin assignments are given in Appendix A. The 44 pin connector, J3, connects to your I/O-driven peripherals . The other two connectors are wired pin to pin to corresponding connectors directly below on the F8S board. The connections are given in Table 4. If more than one expander is used, the connections are the same--bussed pin to pin--except for the PRI.IN.N and PRI.OUT.N signals (J3-21 and J3-C). These two priority signals are "daisy-chained" as given in Table 4; the board that you have chosen to have highest priority will have 0V wired to PRI.IN.BN, J3-C (or strapped at E2); the PRI.IN.BN of each lower priority board will be connected to PRI.OUT.AN (J3-21) of the next higher board.

6.5 Switch Settings: There are six DIP switches on the expander board--one for setting memory page assignment, one for I/O address selection, and four for setting interrupt vectors. Refer to Tables 1, 2, and 3 for switch setting instructions.

	F8S		F8S Memory Expander	
J1:	J1 - xx	to	J1 - xx	
	Pin to pin for all pins except J1 - 18, and J1-V			
J2:	J2 - 2	to	J2 - 2	
	J2 - 4		J2 - 4	
	J2 - 5		J2 - 5	
	J2 - 6		J2 - 6	
	J2 - 9		J2 - 9	
	J2 - 10		J2 - 10	
	J2 - 11		J2 - 11	
	J2 - 12		J2 - 12	
	J2 - C		J2 - C	
	J2 - E		J2 - E	
	J2 - F		J2 - F	
	J2 - K		J2 - K	
	J2 - L		J2 - L	
	J2 - M		J2 - M	
	J2 - N		J2 - N	
J3:	J3 - 1 & J3 - A	to	J3 - 1 & J3 - A	(+5V)
	J3 - B		J3 - B	(RAM +5)
	J3 - N		J3 - N	(+12)
	J3 - 17		J3 - 17	
	J3 - 22 & J3 - Z		J3 - 22 & J3 - Z	(0V)
	J3 - C	to	J3 - 21	
			J3 - C to 0V or J3 - 21 of another expander	

Table 4

F8S/F8S Memory Expander Connections

APPENDIX A: Pin Assignments of Connectors - F8S Memory Expander

Connector J1 36 pins: memory expansion signals

1	Addr ØØ	A	Addr 04
2	Read data Ø	B	Addr 02
3	Addr 01	C	Write data Ø
4	Addr 06	D	Addr 03
5	Read data 1	E	Write data 1
6	Addr 05	F	Write data 2
7	Read data 2	H	Addr 9
8	Read data 3	J	Write data 3
9	Addr 10	K	Addr 8
10	Read data 4	L	Write data 4
11	RAM WRITE N	M	MEM.ENB.N
12	Read data 5	N	Write data 5
13	Addr 12	P	Addr 14
14	Read data 6	R	Write data 6
15	Addr 15	S	Addr 13
16	Addr 11	T	Addr 07
17	Read data 7	U	Write data 7
18	Not used	V	Not used

Connector Amphenol 225-21821-401

Key: Amphenol 225-594 or equivalent

Key should be installed between pins 3 and 4

Connector J2 24 pins: I/O expansion and teletype

1	Not used	A	Not used
2	Data Bus	B	Not used
3	Not used	C	Data Bus 5
4	Data Bus 2	D	Not used
5	Data Bus 4	E	Data Bus 7
6	Data Bus Ø	F	Data Bus 6
7	Not used	H	Not used
8	Not used	J	Not used
9	WRITE.N	K	PHI.N
10	Data Bus 1	L	ROMC4
11	ROMC 2	M	ROMC3
12	ROMC 1	N	ROMCØ

Connector: Cinch 50-24A-30 or equivalent

Key: Cinch 50-5K-2 or equivalent

Key should be installed between 3 and 4.

Connector J3 44 pin: I/O, interrupts, and power

1	+5 pwr	A	+5 pwr
2	Not used	B	RAM +5
3	I/O 14ØN	C	PRI.IN.BN
4	EXT.INT.B.N.	D	I/O 150N
5	I/O 157N	E	I/O 147N
6	I/O 141N	F	I/O 151N
7	I/O 152N	H	I/O 142N
8	I/O 143N	J	I/O 153N
9	I/O 154N	K	I/O 144N
10	I/O 145N	L	I/O 155N
11	I/O 156N	M	I/O 146N
12	I/O 183N	N	+12 pwr
13	I/O 192N	P	I/O 193N
14	I/O 180N	R	I/O 190N
15	I/O 191N	S	I/O 181N
16	EXT.INT.A.N.	T	I/O 182N
17	INT.REQ.N	U	I/O 187N
18	I/O 196N	V	I/O 197N
19	I/O 185N	W	I/O 186N
20	I/O 194N	X	I/O 195N
21	PRI.OUT.AN	Y	I/O 184N
22	OV	Z	OV

Connector: Amphenol 225-2221-401 or equivalent.
Key: Amphenol 225-594 or equivalent.
Key should be installed between pins 3 and 4.

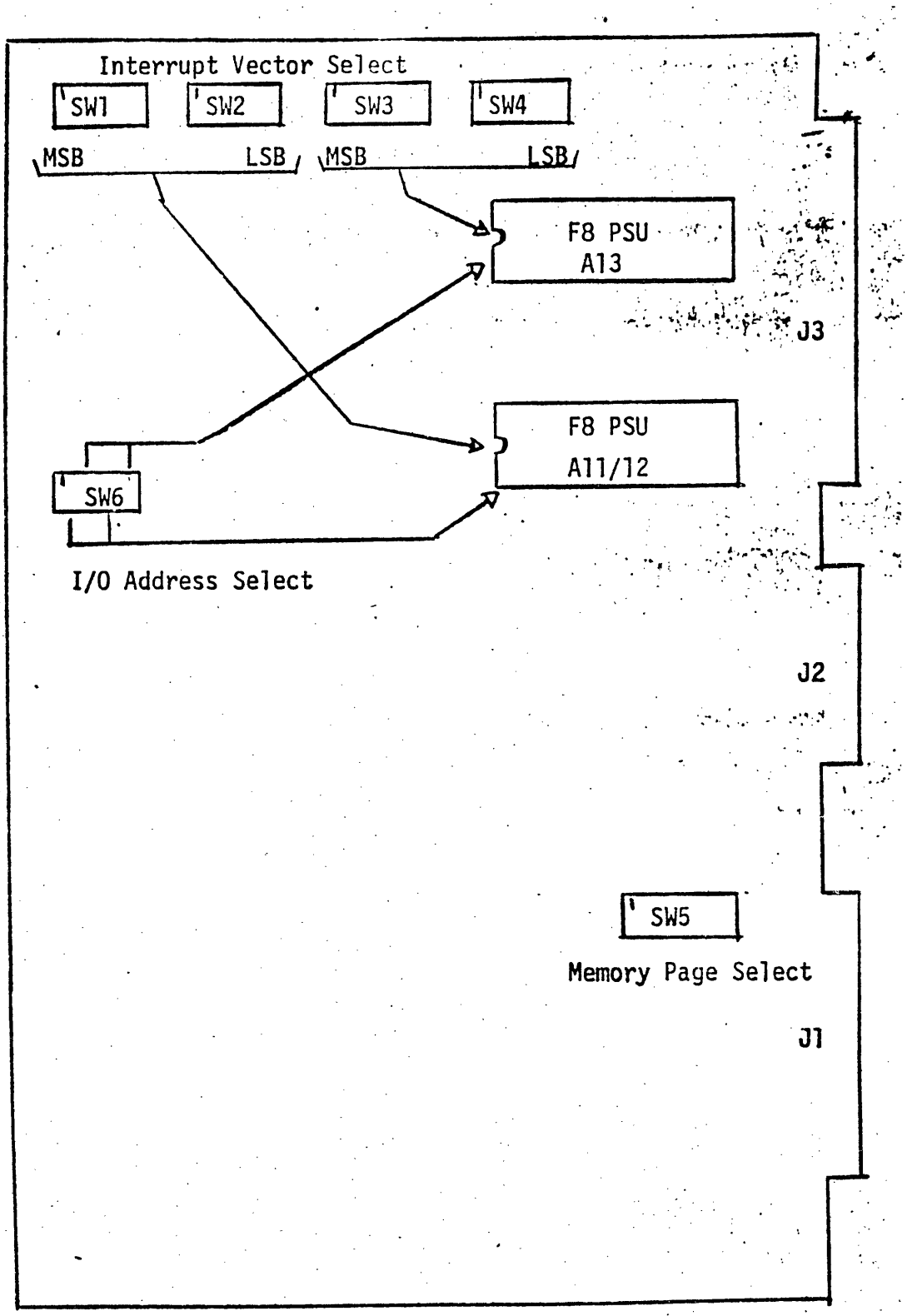


Figure 1

DIP Switch Locations

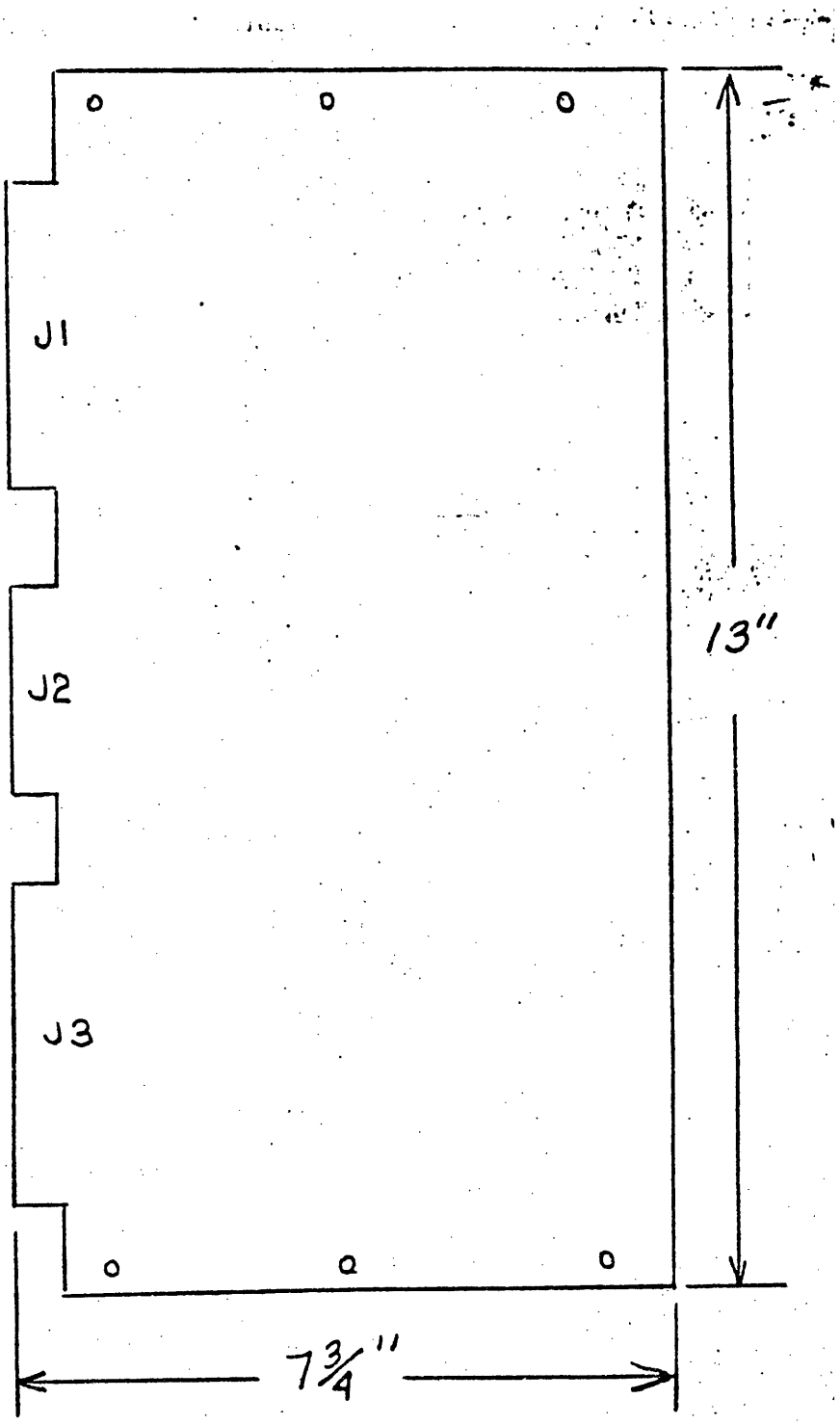
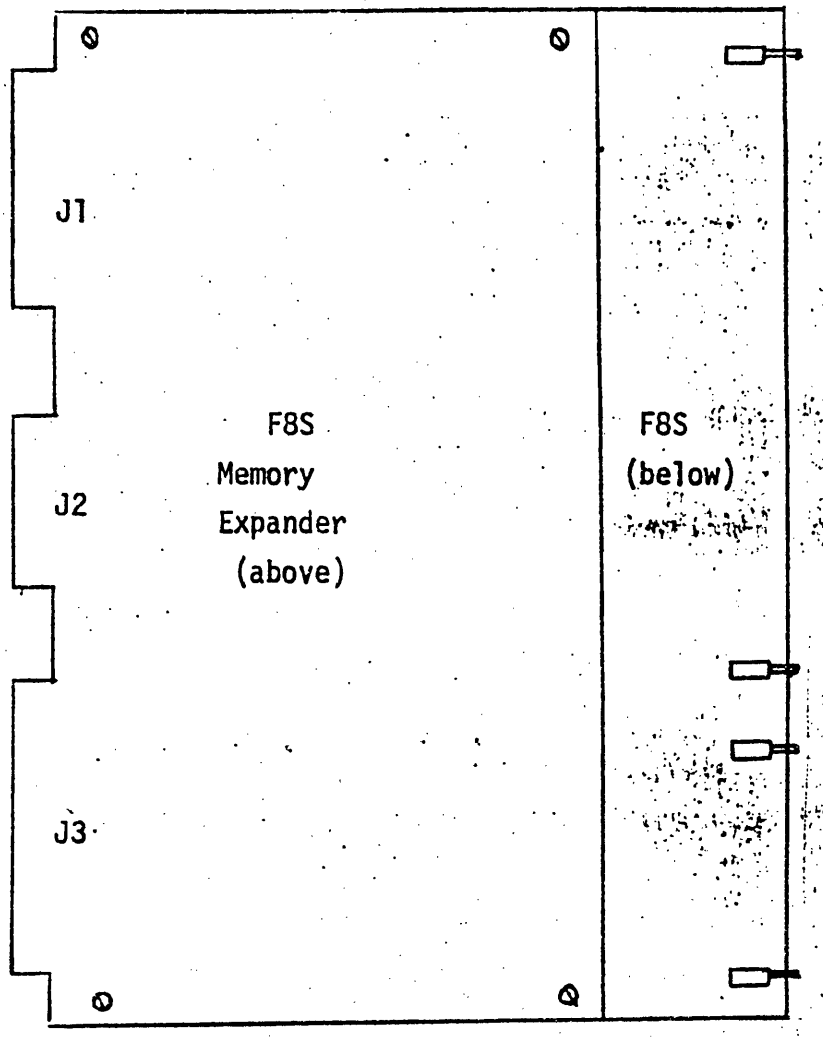


Figure 2.
Physical Size



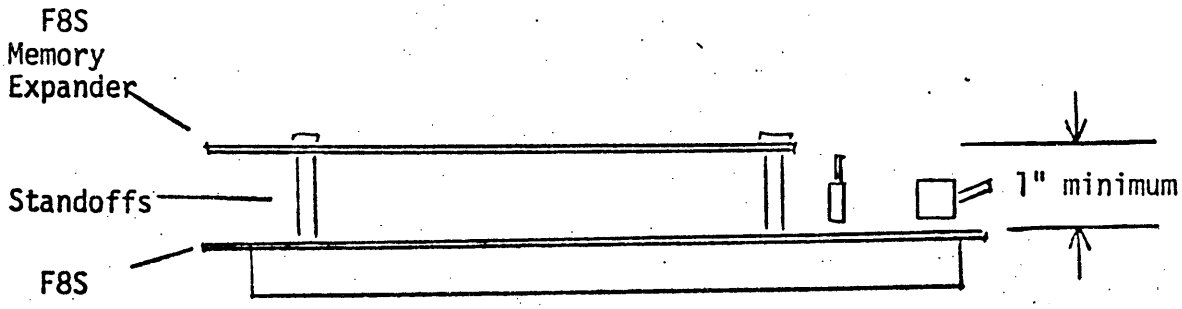


Figure 3

Mounting of Memory Expander