NOTES ON USING THE MIL-MOD 8 SYSTEM: by Bro. Thomas McGahee

First, it should be noted that you may experience some trouble in using the LDO (Load Octal) routine if you follow the directions given in the MIL Manual. What the manual fails to say is that not only must you have a / present to initialize each line, but the / MUST be followed by a space, otherwise the last octal digit in each data field may be missing, and/or the data input may sometimes be found to have been dumped 'shifted, such that a 123 input will show up as 230 when the data is examined. If you are using a TV Typewriter, and letting the data you are inputting do an 'automatic carriage return and line feed' at the end of each line, then you may continue dumping data with only a space between each data field. However, once the carriage return and line feed have been entered via the keyboard, you must preface the next line of data with a / followed by a space.

Another matter which is good to know is that in the LDO routine you can delete mistakes by using the backspace key, but only on the first two digits of the three digit data field! When loading symbolic you cannot delete characters, but you may delete octal digits: but once again, the last digit of a 3 digit data field cannot be deleted. In commands using 2 part data fields, (such as JMP and CAL), the last digit of the first part may be deleted, but once a fourth digit has been input, the first part of the data field cannot be deleted. ie.: in JMP 012345 the 0, 1, or 2 may be deleted BEFORE the 3 is entered, but not after. 3 and 4 may be deleted, but not 5 (because the Monitor immediately executes a carriage return and line feed upon receiving the last digit.)

NOTES ON SYMBOLIC LOADING:

The MIL manual is not very clear on how to load symbolic. Probably because it is so simple! Once you are into Monitor, simply type LOC and then input a six digit starting address. Now you may type your symbolics in directly. For example:

_____ Push restart button.

---------- (Monitor responds with dashes and CR/LF)

LOC 012345 (You type LOC, Monitor responds with a space.)

(You enter a six digit starting address. Monitor responds with CR/LF, and awaits instructions.)

XRA (You may now enter symbolic. Monitor will provide
LBA spaces and wait for octal input on any commands that
JMP require data or addresses.)
.....etc.

Any time the monitor has executed a CR/LF, you may input any mnemonic, including Monitor pseudo-ops such as XQT, LOC, EDT, etc. Pseudo-ops are not entered into memory as a program, but result in immediate Monitor execution.

Note the following: NOP is not a valid opcode. Use LAA as a NOP substitute (NO OPERATION.....it is a program filler often used to plug holes or allow for future expansion of a program.)

If an invalid code is input, Monitor responds with a ? : the invalid code is dumped, and the Current Location Pointer remains where it was (it is not advanced, allowing you to immediately correct your error by typing in the correct code.)

Control commands, (with the exception of CTRL/A) are ignored by Monitor, but they are echoed, and if you are using a TV Typewriter, they may be conveniently used for cursor control, erase functions, ringing bells and the like.

NOTES ON THE USE OF MONITOR SUBROUTINES IN USER-WRITTEN PROGRAMS:

The user may find several of the Monitor sub-routines useful in his own programs, especially the input and output routines. It is important to know what registers the sub-routine uses, and what formating is required of any data that is to be manipulated.

For example, the input sub-routine for getting a serial ASCII character (RST 030) uses registers A B C and D. During acceptance of the character, it is automatically "echoed", and after being stripped of the parity bit the 7 bit ASCII code appears in the A register -- it is now ready for use in the next part of the user's program.

When using the ASCII output sub-routine, (RST 020) registers A B C and D are used. The ASCII data must be in the B register, and it must have a start bit added. For example, if the ASCII character is contained in the A register initially (such as after a memory read), then to output it:

```
ADi 200   (add the start bit)
LBA       (load it into the B register)
RST 020   (output the character)
```

Besides the rather obvious use of outputting a character, the output sub-routine may be used for getting a .09 second delay for programming purposes -- simply output a non-used CTRL character, such as Control Z! The TTY or TVT will ignore it, but it will still take .09 second to output the code....this is much better than writing up a string of timing loops!

ADDRESSING CONVENTIONS USED WITH MONITOR:

MONITOR uses a pseudo-octal addresing technique. In this technique there are six addressing bits, N0 through N5. Together they specify Normal or Extended memory and define which BANK and which BYTE location within a BANK is to be accessed.

The following shows how the address bits are combined to form six pseudo-octal bits:

N5	N4	N3	N2	N1	N0
15,14	13,12,11	10,9,8	7,6	5,4,3	2,1,0

The bits N5 and N2 are only capable of octal values 0-3, whereas all the other bits can have values from 0 to 7. Thus, if the BINARY address was 0011111111111111 it would be expressed as 077377.

N5: 0 or 1 = NORMAL RAM or ROM. 2 or 3 accesses the Prom Programmer, allowing access to the data in the Prom. (As such it is considered as an Extended Memory location).

N4 and N3 select the Memory BANK. Each BANK contains 256 bytes (000 to 377 in octal), and N4 and N3 can select 64 different BANKS (00 to 77 in octal).

N2, N1, and N0 specify which byte within the BANK is to be accessed. As mentioned above, there are 256 possible bytes (000 to 377 in octal).

MEMORY ASSIGNMENTS WHEN USING 1 K BLOCKS (2102)

Refers to using Mod 8-5 with Mod 8 system

MONITOR MEMORY LOCATIONS	IC-8 (DECODER) PIN #
000000--003377	(MONITOR ROM)
004000--007377	(MONITOR ROM)
010000--013377	#13 8-5A (013350--013377 RAM locations are used by MONITOR, and should not be accessed by the user) ***
014000--017377	#12 8-5A RAM
020000--023377	#11 8-5B RAM **
024000--027377	#10 8-5B RAM **
030000--033377	#9
034000--037377	#8

* ALL THE FOLLOWING REQUIRE THAT PIN #6 BE CONNECTED TO A13 (connector pad #20), AND PIN #5 OF IC-8 MUST BE HELD LOW (connected to ground).

040000--043377	#15 *
044000--047377	#14 *
050000--053377	#13 *
054000--057377	#12 *
060000--063377	#11 *
064000--067377	#10 *
070000--073377	#9 *
074000--077377	#8 * ROM 077000 to 077377 **

** Used by Brother McGahee in his system.

*** 256 Bytes of RAM is absolutely necessary at this location in order for the Monitor to function.

INPUT PORTS...MOD/8

There are EIGHT (8) possible Input Ports, three of which are already assigned by the Monitor program.

MONITOR CODE #	FUNCTION
000	TTY (Teletype) Input Port. This is a one-bit port used to accept serial data under Monitor program control. The actual port is located on board 5-2, and uses IC's 3A, 3D, 7A, and 4, as well as a 2N3904 transistor. The code # has been assigned by Monitor. To operate properly, this port requires the TTY to be set up for full-duplex operation, using a 20 MA current loop.
001	Prom Station (Programmer) Input Port. Under Monitor control this port is used to check on the progress during Prom programming. It may also be used to read out the contents of a programmed Prom or ROM. It is treated as an extended memory location by the Monitor program, and is accessed whenever the address is in the range from 200000 to 200377 (see ADDRESSING, 1.3, page C-2).
006	Audio Cassette Interface.

THE REMAINING FIVE INPUT PORTS ARE AVAILABLE TO THE USER.

002**, 003, 004, 005, 007

** Brother McGahee used 002 for his Scientific Calculator Interface.

OUTPUT PORTS...MOD/8

There are TWENTY-FOUR (24) possible output ports. Of these, five are already assigned by MONITOR.

MONITOR CODE #	FUNCTION
010	Prom Address: This port supplies the Prom Station with the proper byte address (000 to 377). It is assigned by the Monitor program. (Due to the nature of the Programmer, only the Low Order address is needed...the Block address is assigned to the Prom in a permanent fashion by hardwiring to a decoder AFTER it is programmed). (The Prom Address Port is located on the Programmer Board, and thus if the USER does not have a programmer, he may assign this code of 010 to some other device).
011	Data FOR Prom: This port supplies the Prom with data when it is undergoing programming. The data is properly inverted under program control prior to appearing at this port. Code # is assigned by Monitor.
012	TTY Out: A one-bit port, which under Monitor control, is used to output ASCII data in the proper serial format, including start and stop bits. It is made up of IC's 3B, 4, 7C, 8, and a 2N3906 transistor. It is set up for use with a TTY operating in the full-duplex mode with a 20 MA current loop.
013	Printer Relay Control/Prom Pulse generator Control. This Port number is assigned to two devices. One is a Printer Relay Control one-bit port composed of IC's 3C, 4, 7C, 8, and a 2N3906 transistor. The other is a port on the Programmer which is used to initiate the programming pulses. They operate under Monitor control, and don't interfere with one another in actual use (or at least the two functions don't upset one another.)
016	Cassette. Assigned by Monitor to Cassette Interface.

014,** 015, 017, 020, 021, 022, 023, 024,.....,036, 037

** Brother McGahee used 014 for his Scientific Calculator Interface.

Bro. Thomas McGahee 80 South Sixth St. Columbus, Ohio 43215

PROGRAM: MONITOR-8P ((parallel input/output))

BANK	BYTE	OCTAL	MNEMONICS	FUNCTION
000	000	250	XRA	* RST 000 = COLD START
	001	133	OUT 015	/ Idle output interface
	002	016	LBI 320	/ B has ASCII for a 'P'
	003	320		
	004	025	RST 020	/ and we print it (P--------)
	005	104	JMP 003000	/ and then go to CONTROLLER.
	006	000		
	007	003		
	010	104	JMP 007000	* RST 010 = CASSETTE ROUTINE
	011	000		/ jump to Cassette Routine.
	012	007		
	013	016	LBI 215	* Print CR + LF
	014	215		/ B has ASCII for CR
	015	025	RST 020	/ and we print it.
	016	016	LBI 212	* Print LF
	017	212		/ B has ASCII for LF (continue..)
	020	250	XRA	* RST 020 = PARALLEL OUTPUT
	021	133	OUT 015	/ Idle output interface.
	022	301	LAB	/ Get character into A.
	023	133	OUT 015	/ Output the character.
	024	000	HLT	/ WAIT for Acknowledge (interrupt).
	025	104	JMP 000064	/ Continued at 000064.
	026	064		
	027	000		
	030	000	HLT	* RST 030 = PARALLEL INPUT
	031	113	INP 005	/ WAIT for interrupt, then get Data
	032	310	LBA	/ and duplicate into B
	033	025	RST 020	/ and ECHO it. (print it).
	034	104	JMP 000075	/ Continued elsewhere.
	035	075		
	036	000		
	037	000	not used	

Bro. Thomas McGahee 80 South Sixth St. Columbus, Ohio 43215

PROGRAM: MONITOR-8P (parallel I/O instead of TTY)

BANK	BYTE	OCTAL	MNEMONICS	FUNCTION
000	040	006	LAI 177	* RST 040 = RUBOUT TEST
	041	177		/ Compare character in B with
	042	271	CPB	/ RUBOUT. IF NOT a rubout,
	043	013	RFZ	/ then Return.
	044	016	LBI 337	/ If it was a rubout, load B
	045	337		/ with ASCII for a left-arrow,
	046	025	RST 020	/ Print it
	047	007	RET	/ and Return.
	050	035	RST 030	* RST 050 = SEARCH FOR "E" CHARACTER
	051	274	CPE	/ Input a character and on a match
	052	053	RTZ	/ Return home.
	053	104	JMP 000050	/ No match ? then try again !!!
	054	050		
	055	000		
	056	000	HLT (not used)	
	057	000	HLT (not used)	
	060	137	OUT 017	* RST 060 = NEW BREAKPOINT ROUTINE
	061	104	JMP 000103	/ STACK A . Continued at 000103...
	062	103		
	063	000		
	064	250	XRA	* Continue output / Clear A
	065	133	OUT 015	/ Idle the output interface,
	066	301	LAB	/ Now both A & B have character.
	067	007	RET	/ Now go on Home !
	070	030	IND	* RST 070 = TIMING LOOP
	071	110	JFZ 000070	/ Increment D... Keep LOOPING
	072	007		/ until D = 000
	073	000		
	074	007	RET	/ Then Return Home.
	075	044	NDI 177	* Continue input / Strip off MSB.
	076	177		
	077	074	CPI 001	/ Check to see if we have CTRL/A,

Bro. Thomas McGehee 80 South Sixth St. Columbus, Ohio 43215

PROGRAM: MONITOR-8P (parallel I/O instead of TTY)

BANK	BYTE	OCTAL	MNEMONICS	FUNCTION
000	100	001		
	101	013	RFZ	/ Return if NOT CTRL/A.
	102	005	RST 000	/ PANIC & RST 000 on CTRL/A .
	103	301	LAB	* Continue Breakpoint
	104	137	OUT 017	/ STACK B
	105	302	LAC	
	106	137	OUT 017	/ STACK C
	107	303	LAD	
	110	137	OUT 017	/ STACK D
	111	304	LAE	
	112	137	OUT 017	/ STACK E
	113	306	LAL	(NOTE: L & H are
	114	137	OUT 017	/ STACK L stored in this
	115	305	LAH	order due to the
	116	137	OUT 017	/ STACK H Monitor routine
	117	056	LHI 013	that retrieves M)
	120	013		/ Set memory up to 013064
	121	066	LLI 064	/ which is where we will store
	122	064		/ characters as we UNSTACK them.
	123	026	LCI 240	/ C has ASCII for a space...
	124	240		/ in case C Flag is False...
	125	342	LEC	/ ..and so does E, for False Z flag.
	126	100	JFC 000133	/ If C Flag is False we keep code
	127	133		/ for a space.
	130	000		/ But if it is true then we store
	131	026	LCI 303	/ ASCII for (C).
	132	303		
	133	110	JFZ 000140	/ Then we check Z Flag, and if
	134	140		/ False, we leave code for space,
	135	000		
	136	046	LEI 332	/ But if True, we store ASCII
	137	332		/ for a (Z).

Bro. Thomas McGahee 80 South Sixth St. Columbus, Ohio 43215

PROGRAM: MONITOR-8P (parallel I/O instead of TTY)

BANK	BYTE	OCTAL	MNEMONICS	FUNCTION
000	140	106	CAL 000013	/ Call a CR + LF so that the
	141	013		/ output from Breakpoint routine
	142	000		/ will be neatly presented.
	143	025	RST 020	/ Do another CR.
	144	016	LBI 252	/ Load ASCII for (*)
	145	252		
	146	025	RST 020	/ and Print it.
	147	016	LBI 240	/ Now load ASCII for a space
	150	240		
	151	025	RST 020	/ and Print that.
	152	312	LBC	/ Get space or (C) for status
	153	025	RST 020	/ of C Flag, and Print it.
	154	314	LBE	/ Get space or (Z) for status
	155	025	RST 020	/ of Z Flag, and Print it.
	156	106	CAL 000013	/ Get another CR + LF
	157	013		/ so we can eventually print ALL
	160	000		/ Registers on one TVT line !
	161	026	LCI 007	/ We have 7 registers to UNSTACK,
	162	007		/ so set Counter to 007.
	163	117	INP 007	* start UNSTACKING registers
	164	370	LMA	/ then store each in Memory.
	165	061	DCL	/ Then move Memory down one place.
	166	021	DCC	/ Decrement Counter
	167	110	JFZ 000163	/ If not done, LOOP to 000163
	170	163		/ and continue UNSTACKING and
	171	000		/ storing until done.
	172	026	LCI 007	/ Set Counter to 007, the number
	173	007		/ of registers to be printed,
	174	104	JMP 003300	/ NOW jump to Monitor routine
	175	300		/ at 003300 , which will print
	176	003		/ A B C D E L H M !!!
	177	035	RST 030	* TEST FOR OCTAL CHARACTER

THAT'S ALL, FOLKS .

Keyboard Routine for Parallel ASCII Keyboard

```
0140000/ 000   HLT
0140001/ 111   INP 004
0140002/ 074   CPI 015
0140004/ 150   JTZ 014107
0140007/ 040   INE
0140010/ 360   LLA
0140011/ 056   LHI 014
0140013/ 041   DCE
0140014/ 110   JFZ 014031
0140017/ 307   LAM
0140020/ 002   RLC
0140021/ 002   RLC
0140022/ 002   RLC
0140023/ 002   RLC
0140024/ 310   LBA
0140025/ 040   INE
0140026/ 104   JMP 014000
0140031/ 307   LAM
0140032/ 201   ADB
0140033/ 310   LBA
0140034/ 041   DCE
0140035/ 007   RET
0140036/ 056   LLI 377
0140040/ 056   LHI 013
0140042/ 307   LAM
0140043/ 061   DCL
0140044/ 317   LBM
0140045/ 061   DCL
0140046/ 277   CPM
0140047/ 013   RFZ
0140050/ 301   LAB
0140051/ 061   DCL
0140052/ 277   CPM
0140053/ 007   RET
0140054/ 000   HLT
0140055/ 000   HLT
0140056/ 000   HLT
0140057/ 000   HLT
0140060/ 000
0140061/ 001
0140062/ 002
0140063/ 003
0140064/ 004
0140065/ 005
0140066/ 006
0140067/ 007
0140070/ 010
0140071/ 011
0140072/ 106   CAL 014036
0140075/ 013   RFZ
0140076/ 104   JMP 014107
0141001/ 012
0141002/ 013
0141003/ 014
0141004/ 015
0141005/ 016
0141006/ 017
```

JMP to the operating system if a "CR" is input. This program can be used for a cold start.

Anytime a "CR" is seen by this routine, the current program is stopped and the system is initialized.

COMPARE CLP TO STOP Address

TABLE LOOK UP

START COMPARE CLP TO Stop Address

TABLE LOOK UP

Operating System used with Parallel ASCII Keyboard

```
014107/ 250   XRA
014110/ 310   LBA
014111/ 320   LCA
014112/ 330   LDA
014113/ 340   LEA
014114/ 006   LAI 377
014116/ 121   OUT 010
014117/ 106   CAL 014000          Load CLP
014122/ 066   LLI 377
014124/ 056   LHI 013
014126/ 371   LMB
014127/ 061   DCL
014130/ 306   LAL
014131/ 121   OUT 010
014132/ 106   CAL 014000
014135/ 066   LLI 376
014137/ 056   LHI 013
014141/ 371   LMB
014142/ 061   DCL
014143/ 076   LMI 104
014145/ 106   CAL 015361
014150/ 121   OUT 010
014151/ 000   HLT
014152/ 111   INP 004
014153/ 074   CPI 112             JMP to Address in CLP if "J" is input.
014155/ 150   JTZ 013375
014160/ 074   CPI 130             Read Data at CLP address on Output #1
014162/ 110   JFZ 014173          When "X" is input.  Increment CLP also.
014165/ 106   CAL 015343
014170/ 104   JMP 014145
014173/ 074   CPI 114             Load next two hexadecimal characters as
014175/ 110   JFZ 014107          one byte at address in CLP.  INR CLP.
014200/ 306   LAL                 Output #1 displays the lower address in
014201/ 121   OUT 010             CLP.
014202/ 106   CAL 014000
014205/ 106   CAL 015361
014210/ 371   LMB
014211/ 106   CAL 015343
014214/ 104   JMP 014201
--------
```

Editor Routine

```
014217/ 006   LAI 014
014221/ 121   OUT 010
014222/ 026   LCI 013
014224/ 036   LDI 377
014226/ 106   CAL 014000
014231/ 352   LHC
014232/ 363   LLD
014233/ 371   LMB
014234/ 061   DCL
014235/ 325   LCH
014236/ 336   LDL
014237/ 306   LAL
014240/ 074   CPI 367
014242/ 121   OUT 010
014243/ 110   JFZ 014226
014246/ 106   CAL 015361
014251/ 044   NDI 305
014253/ 074   CPI 004
014255/ 150   JTZ 014344
014260/ 044   NDI 301
014262/ 074   CPI 100
014264/ 110   JFZ 014347
014267/ 060   INL
014270/ 060   INL
014271/ 307   LAM
014272/ 066   LLI 373
014274/ 056   LHI 013
014276/ 277   CPM
014277/ 110   JFZ 014341
014302/ 061   DCL
014303/ 317   LBM
014304/ 106   CAL 015361
014307/ 301   LAB
014310/ 060   INL
014311/ 277   CPM
014312/ 150   JTZ 014320
014315/ 100   JFC 014341
014320/ 056   LHI 013
014322/ 066   LLI 371
014324/ 317   LBM
014325/ 061   DCL
014326/ 327   LCM
014327/ 106   CAL 015361
014332/ 060   INL
014333/ 307   LAM
014334/ 201   ADB
014335/ 370   LMA
014336/ 060   INL
```

Output #1 will show a 06

Input:
1. Start Address
2. Stop address
3. Insertion Address
4. No. of byte to inserted
5. Upper Address

The insertion address is the point in the program where the bytes are to be inserted or deleted.

If bytes are to be deleted, just two's complement the number and a subtraction will occur. Example: To subtract 1, add FF.

Any JMP or Call Commands with the same upper address as the insertion address will be changed to the Upper Address. This allows the program to be used for block changes. The portions of the modified program can then be load on to cassette tape and reloaded into the proper location and any insertions keyed in from the keyboard.

```
014337/ 372   LMC
014340/ 300   LAA
014341/ 106   CAL 015343
014344/ 106   CAL 015343
014347/ 106   CAL 015343
014352/ 106   CAL 014072
014355/ 104   JMP 014246
014360/ 056   LHI 013
014362/ 066   LLI 376
014364/ 307   LAM
014365/ 121   OUT 010
014366/ 000   HLT
014367/ 060   INL
014370/ 307   LAM

014371/ 121   OUT 010
014372/ 000   HLT
014373/ 007   RET
014374/ 000   HLT
014375/ 000   HLT
014376/ 000   HLT
014377/ 000   HLT
--------
```

End of Editor
Start CLP Display Routine

The Editor Routine, the Operating System , and the Keyboard Routine are courtesy of Mini Micro Mart and C. W. Blevins, Graduate Institute of Technology, UNIVERSITY OF ARKANSAS, 1201 McAlmont St., Little Rock, Arkansas 72203.