plus
LA-3640

291
3/10/67

MASTER

LOS ALAMOS SCIENTIFIC LABORATORY
of the
University of California
LOS ALAMOS • NEW MEXICO

Simulation of the SEL-810A Computer
on MANIAC II (SELMA)

RELEASED FOR ANNOUNCEMENT
IN NUCLEAR SCIENCE ABSTRACTS

UNITED STATES
ATOMIC ENERGY COMMISSION
CONTRACT W-7405-ENG. 36

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency Thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

DISCLAIMER

Portions of this document may be illegible in electronic image products. Images are produced from the best available original document.

LEGAL NOTICE

This report was prepared as an account of Government sponsored work. Neither the United States, nor the Commission, nor any person acting on behalf of the Commission:

A. Makes any warranty or representation, expressed or implied, with respect to the accuracy, completeness, or usefulness of the information contained in this report, or that the use of any information, apparatus, method, or process disclosed in this report may not infringe privately owned rights; or

B. Assumes any liabilities with respect to the use of, or for damages resulting from the use of any information, apparatus, method, or process disclosed in this report.

As used in the above, "person acting on behalf of the Commission" includes any employee or contractor of the Commission, or employee of such contractor, to the extent that such employee or contractor of the Commission, or employee of such contractor prepares, disseminates, or provides access to, any information pursuant to his employment or contract with the Commission, or his employment with such contractor.

This report expresses the opinions of the author or authors and does not necessarily reflect the opinions or views of the Los Alamos Scientific Laboratory.

Printed in the United States of America. Available from
Clearinghouse for Federal Scientific and Technical Information
National Bureau of Standards, U. S. Department of Commerce
Springfield, Virginia 22151

Price: Printed Copy \$3.00; Microfiche \$0.65

LA-3640
UC-32, MATHEMATICS
AND COMPUTERS
TID-4500

CFSTI PRICES

H.C. \$3.00; MN .65

LOS ALAMOS SCIENTIFIC LABORATORY
of the
University of California
LOS ALAMOS • NEW MEXICO

Report written: November 1966

Report distributed: February 27, 1967

**Simulation of the SEL-810A Computer
on MANIAC II (SELMA)**

by

N. Metropolis
John R. Pasta*
Malcolm Fraser

RELEASED FOR ANNOUNCEMENT
IN NUCLEAR SCIENCE ABSTRACTS

*University of Illinois

LEGAL NOTICE

This report was prepared as an account of Government sponsored work. Neither the United States, nor the Commission, nor any person acting on behalf of the Commission:
A. Makes any warranty or representation, expressed or implied, with respect to the accuracy, completeness, or usefulness of the information contained in this report, or that the use of any information, apparatus, method, or process disclosed in this report may not infringe privately owned rights; or
B. Assumes any liabilities with respect to the use of, or for damages resulting from the use of any information, apparatus, method, or process disclosed in this report.
As used in the above, "person acting on behalf of the Commission" includes any employee or contractor of the Commission, or employee of such contractor, to the extent that such employee or contractor of the Commission, or employee of such contractor prepares, disseminates, or provides access to, any information pursuant to his employment or contract with the Commission, or his employment with such contractor.

THIS PAGE
WAS INTENTIONALLY
LEFT BLANK

ABSTRACT

The SEL-810A computer has been simulated on MANIAC II; MADCAP has been used as the programming language. The exercise has several objectives: a detailed knowledge of the logical structure of the SEL-810A and the preparation of programs would be possible; the suitability of MADCAP could be examined for this type of application. The two-pass assembler provided for the SEL computer was tested, and errors in the original versions were detected and corrected. An account is given of other experience.

ACKNOWLEDGMENT

The authors wish to thank Mark Wells and Roger Lazarus of T-7 for useful discussions of the MADCAP language. We also thank Harold Butler of MP-1 for discussions concerning the operation of the SEL computer.

One of us (J.R.P.) is indebted to Carson Mark for the invitation to spend an interesting summer at the Laboratory.

THIS PAGE
WAS INTENTIONALLY
LEFT BLANK

CONTENTS

	Page
Abstract	3
Acknowledgment	3
1. Introduction	7
2. Program Organization	10
3. Declaratory Statements and Initialization	12
4. Input Phase	14
5. Instruction Decoding and Effective Address Computation . .	16
6. Diagnostics	17
7. Interrupt Processor	19
8. Instruction Interpretive Routines	20
9. I/O Instructions	21
10. Procedures	23
11. Concluding Remarks	26

APPENDICES

A. Timing Parameters	28
B. Sense Lights	29
C. SEL Console Stops	33
D. Tag Meanings	35
E. Console Display	37
F. Teletype - Mark 3 - Printer Character Equivalence	38

FIGURES

	Page
1. Simulation procedure	8
2. Simulator organization	11
3. Program sample of MADCAP statements	27

1. Introduction

1.1 A simulator called SELMA which interprets SEL-810A machine language into equivalent MANIAC II (M2) machine language has been written using MADCAP.¹ No attempt has been made to attain maximum speed or memory economy; rather the criteria of the speed of programming the simulator and the clarity of the final program have been selected. The MADCAP language is well-suited for the description as well as the programming of the subject program.

1.2 Given the SEL assembler called MNEMBLER² written in modified SEL machine language and the appropriate SEL loader which converts modified to absolute language, we may then assume the existence of a MNEMBLER machine code; and a program written in MNEMBLER language may be processed as shown in Fig. 1.

1.3 The input-output complement of M2 is not the same as that of the SEL computer, so compromises are necessary. The fast paper tape reader and punch are compatible on the two computers, so no modification is necessary. The M2 line printer has a greater width, which causes no severe difficulty, but the character set and paper control are slightly

¹MADCAP Manual, Group T-7, Los Alamos Scientific Laboratory (1964).

²SEL-810A Reference Manual, Fort Lauderdale, Florida (1966).

Fig. 1. Simulation procedure.

different. The proper correspondence is given later.

The greatest mismatch is in the ASR-33 to Mark 3 typewriter mapping. No reader or punch is available on the Mark 3, and there is a substantial difference in the special character set.

1.4 Interrupts may be introduced into the system by means of a paper tape giving the priority level and the time of the interrupt in cycles or by giving the setting of the SEL program counter at which the interrupt is desired. If it is desired that the program run without interrupts, they can be suppressed by a sense setting on the M2 console.

1.5 Display of the internal state of the SEL-810A can be obtained under M2 sense switch control, and changes can be introduced into the SEL computer memory by means of the M2 Mark 1 keyboard. The actual SEL running time can be exhibited.

1.6 Certain programmed M2 stops indicate possible errors in the SEL code and are provided as an aid in debugging. The meanings of these stops will be spelled out below. The SEL program may also be stopped manually or breakpointed.

1.7 Times, in cycles, for carrying out peripheral operations are entered as parameters N_k , where k is an integer, and are used in the computation of the actual SEL running time.

1.8 The operating speed of the simulator is approximately 1000 times slower than the actual SEL computer. The SEL assembler carries out the full assembly at about two statements per minute.

2. Program Organization

2.1 The general organization of the program is shown in Fig. 2, and each part is described in more detail in the following sections. A description of the self-contained procedures is also given.

2.2 The simulator is stored in the M2 disk file, and calling is carried out under standard M2 operating procedure.

2.3 Although a knowledge of certain parts of MADCAP is needed to understand the program, the procedure descriptions are made independently of that facility; and a view of the simulator can be had independently.

2.4 An overview of the program is given in the following sections.

Fig. 2. Simulator organization.

3. Declaratory Statements and Initialization

3.1 As in most compilers, words may be of various number-types and must be declared or implied to be of a certain type. The number-types used in this simulator are:

3.11 Real: Floating Point Number

3.12 Word-Set: A 48-bit M2 word on which Boolean set-operations may be performed.

3.13 Label: A symbolic location of a MADCAP statement.

3.14 String: A sequence of 8-bit characters of specified length.

They are declared, generally, early in the program.

3.2 Also early in the program parameters are initialized, latches are set, and formats for input and output are declared.

3.21 N1 through N13 are parameters describing the operation times of various peripheral devices and should be set with proper values if interrupt timing is to be simulated properly.

3.22 The parameters prescribing when the different peripheral devices will next be available are initialized to zero.

3.23 The pointers I1, I2, and I3 on the interrupt lists are set to zero.

- 3.24 Tag bit 0 (Teletype clear) is set to one and the remaining Tag bits are set on initial zero. The Tag word is defined in Appendix D.
- 3.25 The Stall Alarm enable bit is set. This bit is not disturbed during operation of SELMA and has the highest priority. All other bits in the interrupt enable set are initialized to zero.
- 3.26 Alatch, a flag signalling that the highest active interrupt is to be turned off at the next LOB or BRU* instruction, is set to the zero or off position.

4. Input Phase

4.1 When SELMA is called from the disk, any absolute SEL tapes should be in the reader with sense 0 in the set position if more absolute tapes are to follow.

4.2 The absolute tape will be read in and SELMA will come to stop 0 if there are more absolute tapes. These tapes are in standard SEL format with a local address and negative word count at the start of the tape (SEL Manual, p. D.1, Debug Dump Tape). They will be automatically loaded at the proper place.

4.3 If sense 2 is on, SELMA will halt at stop 2 and print "Load time interrupts." Any time interrupt tape is loaded and sense 2 is left on in order to maintain these interrupts in an active status.

4.4 Sense 3 has a similar action for location interrupts.

4.5 If sense 6 is on, stop 6 will occur, allowing the operator to type an execution starting address into the last 13 bits of the top console register display.

4.6 If sense 6 is off, sense 7 will determine the starting address. It will be 0 (standard assembly location) if sense 7 is off and 6677 (standard location of loader start) if set.

4.7 The last stop is "a" and this is the only stop which always occurs regardless of sense settings. The words, "Insert Data Tape and

Go" will be printed, allowing for any program data including programs to be assembled or loaded (which are data to the assembler or loader programs) to be inserted in the reader..

4.8 The date is printed and the input phase is complete.

5. Instruction Decoding and Effective Address Computation

5.1 The instruction type is recognized by the first four bits of the word and, in the case where they are all zero, by the last 6 bits, which then contain the augmented code.

5.2 A special case is made for the instructions of octal type 13 or 17, which are I/O or interrupt type instructions. In the case of shift instructions, the amount of the shift is extracted for transmission to the interpretive routine.

5.3 The procedure which computes the effective address is called effective and is described later.

6. Diagnostics

6.1 In order to assist in debugging, certain facilities are made available to the operator and programmer.

6.2 If sense 4 is on, at the completion of each SEL instruction there will be a printout of the SEL cycle number, in decimal; the instruction location; the instruction itself; the effective address last computed; the contents of that address; the A and B accumulator contents, all in octal; and the contents of the location called Tag (see Appendix D) in binary.

6.3 Sense 5 will cause printing of cycle and instruction location, in decimal, and the word-sets Enable, Request, and Active in hexadecimal so that interrupts can be studied.

6.4 Sense 4 and 5 may be actuated together. An example of this type of diagnostic printout is:

39 1660 C_p=1111656 e=1656 C_e=034531 A=000000 B=177645 Tag=000110000000010001

39 944 Enable=000000001 Request=000000000 Active=000000000

6.5 In order to breakpoint a SEL instruction, enter the instruction location in the first 16 bits of the panel array in octal and set sense 9. If the program counter assumes the value on the panel, there will be a stop 9 before performing the instruction. Setting sense 9 will not cause the corresponding sense light to remain on when the simulator is

running since it is immediately interrogated and reset. Only one breakpoint at a time can be entered.

6.6 Sense switch 8 is the "manual" switch and will cause a halt after each SEL instruction. This occurs after any diagnostic print requested by sense 4 or 5. The console will display, after every stop, the internal state of the SEL computer (Appendix E). If now the fetch switch is depressed, SELMA will resume simulation to the next stop or will run on automatic, as requested on sense 8.

6.7 The Stall Alarm is implemented as a trap to a level designated as 0 having highest priority.

7. Interrupt Processor

7.1 If sense 2 or 3 is on (Sec. 4), the interrupt list is interrogated, and all time interrupts whose cycle is less than or equal to the machine cycle are entered into Request. If the location interrupt list contains the program counter setting, an interrupt is similarly entered.

7.2 An attempt to interrogate an empty list will result in stop 12, which may be overridden by resetting the sense switch and continuing.

7.3 If the interrupt suppress occurring after certain instructions is on, it is turned off, and SEIMA continues without further action.

7.4 An interrupt is processed by the interrupt procedure described later.

8. Instruction Interpretive Routines

8.1 These are interpretive routines which simulate each SEL instruction. The instructions are sixteen bit 2's complement notation; therefore, many of the interpretive routines involve word-set operations.

8.2 Memory referencing instructions use the procedure effective to get the final effective address before entering the interpretive routine.

8.3 Shift instructions use shift to update the cycle number and to transmit the magnitude of the shift.

8.4 The TOI instruction sets a location called Alatch to the value corresponding to the highest active interrupt. This interrupt is reset at the next LOB or BRU* instruction.

8.5 I/O instructions are treated in the next section.

9. I/O Instructions

9.1 The I/O instructions for each peripheral device are separate routines.

9.2 The TEU instructions are not implemented, except for the line printer, since the I/O interrupt will be used and each device will be connected to a separate level so that no ambiguity of unit will occur.

9.3 Conflicts on the line printer in CEU instructions are detected, and stop e will signal their occurrence. The CEU "format N" causes a one-line advance, only.

9.4 The meaning of the different stops can be found in Appendix C.

9.5 The teletype routine uses the procedure convert to establish the correspondence between the Mark 3 typewriter and the ASR-33 code. Refer to Appendix F for the equivalence of certain characters. The typing of an illegal character causes a stop 7 and ignoring of character so continuation is possible.

9.6 The timing is carried out using the parameters N 1 through N 13 listed in Appendix G.

9.7 It is possible to insert a Flexowriter-generated tape in the fast reader and have it interpreted as SEL code. This is accomplished by setting sense 1.

9.8 Whenever an I/O instruction is performed, the new time at which the unit will be ready is inserted into the I/O interrupt list at the interrupt level appropriate to the unit. The interrupt processor will enter the request at the appropriate cycle.

10. Procedures

10.1 Complement

comp (F,G)

The intersection of G with the 2's complement of F is transmitted without side effects.

10.2 Shift

shift (none; p, C, Cycle)

The magnitude of the shift is transmitted from SEL instruction p (MANIAC C_p); and as a side effect, Cycle is advanced the appropriate amount.

10.3 I/O Timing

twx (Select, Delay; all)

If the wait flag in the I/O instruction in F is on, set Cycle to $\max(\text{Cycle}, \text{Tunit})$ where Tunit is the last set ready time. Cycle + Delay, the next ready time, is set into Tunit.

If the wait flag is not on but the unit is ready, advance p by 1 (skip) and advance Tunit.

In these cases, Mode is set to Select with the meanings for the ASR-33

-1 keyboard mode

0 clear mode

1 reader mode.

Unit not ready and no wait flag causes an exit with no side effects. Tag 14 is set if this latter exit occurs.

10.4 Typewriter Character Conversion

Convert (T; Tag)

Tag 5 is set to 0 for output and 1 for input before entry.

In the case of output, the leftmost 8 bits of the SEL word Wd are converted to the appropriate MANIAC character string and outputted to the typewriter (no punch). Contrary to the statements in the manual, output to the line printer is from the right 8 bits.

On input, characters typed on the Mark 3 typewriter are converted to an 8-bit character right-justified. The character formation is delayed until the change in case has been completed.

Case designation is carried in tag 3 and carries between procedure calls so that the current state can be determined and unnecessary case change actions are eliminated.

If an illegal character is input, T is set to zero and SELMA will halt on stop 7. A fetch of the next instruction will allow another try with case remembered from the illegal action (i.e. present status).

See Appendix F for character correspondences.

10.5 Effective

effective (none; B, C, p)

This procedure computes the effective address of the SEL instruction at location p whose image is stored in MANIAC location C_p . It will map, index with B, and compute to any indirect depth adding 1 to Cycle for each indirect level.

If the final effective address or the effective address at any indirect level exceeds 8192, the words:

"Address exceed 8192 at p=xxxx"

will be printed and MANIAC will halt.

The location 8192 rather than 8191 is used, because the wired-in SPB* instruction associated with the interrupt hardware is executed at the artificial location 8192.

The procedure has no side effects.

10.6 Basic Interrupt Routine

interrupt (none; Active, Enable, Request, p, C, Cycle)

If the interrupt level "Level" is enabled and no higher levels are active, the wired-in instruction SPB*L is executed at SEL location 8192 and both the "active" and "request" bits for that level are set.

If Level does not meet this condition, the request bit is set and the program exits immediately.

11. Concluding Remarks

11.1 The MADCAP language was particularly suited to the writing of this simulator. The set operations were of great value and had a naturalness which made them simple to use. The great generality of the various types of conditional statements helped in making the logic of the program simply and clearly expressible.

11.2 Because the MADCAP statements are expressed so easily, we believe that logical and syntactic errors are less likely to occur and, indeed, it was our experience that the debugging time was remarkably shorter than for other formal languages.

11.3 In order to give support to these statements, we present the following data:

Number of compound MADCAP statements	710
Programming	7 man-weeks
MANIAC II debugging	19.5 hours
	51 sessions
Utilization time, to date	23.5 hours
	47 sessions

11.4 To give some feeling for the appearance of the language, an example is shown in Fig. 3.

"Add Memory to B Accumulator"

Statement number	Madcap Statement	Comments
#1400	$F = C_e; S = (F \cap \{15\}) \Delta (B \cap \{15\})$	Save contents of effective address. $S = 0$ if both sign bits $\{15\}$ of F and B accumulators are the same; otherwise $S \neq 0$.
#1401	$F = \text{word}(\text{number}(F) + \text{number}(B)) \cap 16$	Form algebraic sum of F and B and truncate to 16 bits.
#1402	Unless $F \cap \{15\} \neq B \cap \{15\}$, and $S \equiv 0$, go to #1403	If the signs of F and B are now different and $S = 0$, overflow has occurred.
	$\text{Tag} \cup \{4\} \rightarrow \text{Tag}$	Set overflow latch to 1.
#1403	$B = F$	Result to B accumulator.
#1404	Go to #101	Exit to advance cycle and program counter.

Fig. 3. Program sample of MADCAP statements.

APPENDIX A

Timing Parameters

The following I/O execution times are to be set in SELMA to ensure proper updating of cycle.

N1	Fill reader buffer
N2	Reader enable
N3	Punch cycle
N4	TWX input (read buffer fill time - either mode)
N5	TWX output time
N6	Set TWX mode
N7	Line printer time
N8	Punch turn-on
N9	Next page
N10	Next line
N11	Enable printer buffer
N12	Clear buffer (printer)
N13	Local character in printer buffer

APPENDIX B

Sense Lights

The first ten sense switches on the MANIAC console are interrogated by SELMA and can be used to allow various simulator operation modes. If a console stop occurs as a result of one of these sense interrogations, the console will display the internal state of the SEL computer. Operation can be resumed by depressing the Fetch switch.

<u>Sense</u>	<u>Set Condition</u>
0	<u>Load Absolute and Halt.</u> There is no halt before the first absolute tape is read, so it is assumed that there is an absolute tape in the reader at the start of SELMA if this sense is set. As long as sense 0 is set, SELMA will halt at stop 0 for more tapes. Sense 0 can be reset at any 0 stop. The format for an absolute tape is that given in Appendix D of the SEL manual for the "Debug Dump Tape." This tape can also be loaded with the SEL Debug Loader Program.
1	<u>Flexowriter-Generated Tape in Reader.</u> On reader AIP and MIP instructions, the characters read are assumed to be Flexowriter characters and conversion to ASR-33 equivalents are made automatically. The equivalence is that of Appendix F.

SenseSet Condition

- 2 Time Interrupts. SEIMA will print "Load time interrupts," eject a page, and come to stop 2. The Time Interrupt tape should be inserted before continuing operation. Time interrupts will be processed during the program as long as this sense is left on.
- 3 Location Interrupts. SEIMA will print "Load location interrupts," eject a page, and come to stop 3. The Location Interrupt tape should be inserted before continuing operation. Location interrupts will be processed during the program as long as this sense is left on.
- 4 Print Standard Diagnostics. After each SEL instruction, print a line displaying Cycle, p, C_p , e, C_e , A, B, and Tag, consecutively. Tag is displayed as a binary number.
- 5 Print Interrupt Diagnostics. After each SEL instruction, print a line displaying Cycle, p, Enable, Request, and Active, consecutively. If sense 4 is also set, both diagnostics are printed and a line is skipped after the interrupt diagnostics.
- 6 Halt for Starting Address. Causes console stop 6. The 16-bit number entered into console position 2 (see Appendix E) is taken as the execution starting address in the SEL computer. If this switch is in the reset position, the setting of sense 7 will determine the starting address.

Sense

Set Condition

- 7 Start at Loader. The program will start at 15025 (octal).
This is the location of the loader in the standard state.
If the switch is in the reset position, the starting location is 0, the assembler starting address in the standard state. Setting of sense 6 will always override this sense.
- 8 Manual Operation. The computer will halt after each SEL instruction at console stop 8 (Sec. 6).
This sense switch replaces the MANIAC Manual/Automatic Switch.
- 9 Enter SEL Breakpoint. When this switch is moved to "set", the first 16 bits of the panel array are entered as a breakpoint (Sec. 6).
- 10 Suppress I/O Interrupts. All interrupts from I/O devices are suppressed. Since detection of I/O interrupts can be time-consuming, this sense should be set if no I/O interrupts are expected in order to ensure better SELMA running time.
- 11 Absolute Code Modification. When this switch is "set" after a console stop, the paper tape in the reader is used to modify the SEL absolute code in the SEL simulated memory. The format of the paper tape is
a. number of corrections in decimal followed by a carriage return.

Sense

Set Condition

- b. location to be modified expressed in six octal digits followed by a Tab.
- c. contents, in octal, of the modified location followed by a carriage return.
- d. steps b and c are repeated for remaining modifications.

APPENDIX C

SEL Console Stops

<u>Console Stop</u>	<u>Meaning</u>
0	Load next absolute tape or reset (sense 0).
1	HLT program halt. (Cycle count is reset to 0.)
2	Load time interrupts (sense 2).
3	Load location interrupts (sense 3).
4	Program counter overflow (8192).
5	Priority Interrupt group too large.
6	Enter starting address in S (sense 6).
7	Illegal I/O character (deleted).
8	Manual step alteration stop (set dial to 0 to continue) (sense 8).
9	Panel breakpoint stop. <ul style="list-style-type: none">a. Insert data tape and go.b. Incorrect peripheral unit number.c. Non-implemented TEU.d. Peripheral unit in wrong mode.e. Printer conflict condition.f. Printer "input" called.
10	Printer buffer full.

Console StopMeaning

11	Space in interrupt list exhausted.
12	No further interrupts (sense 2 and 3).

APPENDIX D

Tag Meanings

The word "Tag", which is displayed in console position 3 after each SEL stop, has the following bit-by-bit meaning. Initially, there is a one in position 0 with the rest zero.

<u>Bit Location</u>	<u>"0" meaning</u>	<u>"1" meaning</u>
0	Neutral	Teletype Clear Mode
1	Neutral	Teletype Keyboard Mode
2	Neutral	Teletype Reader Mode

NOTE: Initially bit 0 is a one and bits 2 and 3 are zero. CEU actions can cause any of bits 0, 1, and 2 to be set to a one but the other two will then be set to zero.

3	Mark 3 in Upper Case	Lower Case
---	----------------------	------------

NOTE: The ASR-33 equivalent of Mark 3 input depends on the case of the Mark 3 character, so the case must be remembered from character to character on input. SELMA assumes that the initial state is upper case.

4	No overflow	Overflow
5	Character Output	Character Input

<u>Bit Location</u>	<u>"0" meaning</u>	<u>"1" meaning</u>
-------------------------	--------------------	--------------------

NOTE: Communicates to convert procedure whether conversion is for input or output.

6	Carry "0"	Carry "1"
---	-----------	-----------

NOTE: Set by CSB instruction, reset by AMA and SMA.

7	Neutral	Presence of interrupt to be processed
---	---------	--

8	Reader disabled	Reader enabled
---	-----------------	----------------

9	Punch disabled	Punch disabled
---	----------------	----------------

10	Neutral	BTC on
----	---------	--------

11-13	not used	
-------	----------	--

14	No I/O action	I/O action
----	---------------	------------

NOTE: The TWX procedure will set this tag to indicate I/O action. A "not ready", no wait I/O operation will set the tag to zero, otherwise to one.

15	Interrupt Suppress off	Suppress on
----	------------------------	-------------

NOTE: The interrupt hardware is suppressed after SPB, PIE, PID, TOI, and CSB instructions. This tag effects that suppression.

APPENDIX E

Console Display

	0	1	2	
S	Stop #	C _p	p	16 bit alter
	3	4	5	
R	Tag	C _e	e	
	6	7	8	
U	CS	A	B	

	<u>Dial</u>	
	0	go
alter code	1	C _p
	2	p
	3	Tag
	4	C _e
	5	e
	6	CS (control switches)
	7	A
	8	B
	>8	go

APPENDIX F

Teletype - Mark 3 - Printer Character Equivalence

All teletype characters have their Mark 3 and printer equivalents except:

<u>Teletype</u>	<u>Mark 3</u>
@	a
\	
↑	Δ
←	— (c on printer)
!	f (i on printer)
\$	s
ℓ	√
&	x
;	π (; on printer)
?	z (π on printer)
CR	CR (with line feed)
LF ³	→

³Note that the line feed is not automatic on the Teletype and must be explicitly struck. Thus each new line will begin with the symbol → even though many SEL programs will detect and skip line feed characters.