ASYS MONITOR

Theodore Strollo
Air Force Cambridge Research Laboratories (CRBI)
L. G. Hanscom Field
Bedford, Mass.

FOREWORD

The AIDEX SYSTEM is the result of the work and ideas of many people. I would like to thank:

Mr. Don Easterday who wrote the ASYS LIBRARY Tape Maker

Lt. Richard Smallwood who wrote the two Floating-point systems (FPT-18 and FPT-28)

Dr. F. Corbato, MIT Computation Center, and The Bolt Beranek & Newman Co., Inc. who through their Linking Loaders furnished much of the philosophy for the ASYS Linking Loader.

I also wish to thank Mr. Charlton M. Walter of the Air Force Cambridge Research Laboratories, for making available time on the Experimental Dynamic Processor (DX-1) for the realization of ASYS.

Ted Strollo

INTRODUCTION

The ASYS MONITOR is an executive routine which handles the control of all assemblies, compilations, and loadings made with the AIDEX System. The Monitor accepts commands instructing it to:

1) initiate assemblies or compilations

2) load relocatable binary output from the assembler or compiler

3) search the ASYS LIBRARY and load selected routines from the LIBRARY

4) punch (on paper tape) relocatable or absolute binary versions of programs

5) edit magnetic tapes

6) call the ASYS debugging system

Paper tape handling during assembly, compilation, and loading is kept to a minimum by the MONITOR.

TABLE OF CONTENTS

I. The ASYS SYSTEM TAPE

The ASYS SYSTEM TAPE contains:

1) The ASYS MONITOR, LOADER, and EDITOR

2) AMP Passes 1 and 2, AMP Symbol Table Dump Routine

3) AIDEX Passes 1 - 4, AIDEX Symbol Table Dump Routine

4) Utility Programs (Expensive Typewriter, etc.)

5) The ASYS LIBRARY

The SYSTEM TAPE must be mounted on a tape drive dialed in as unit 1. The MONITOR record is read into core 1 with the "read-in-monitor" paper tape. This paper tape is read into core 0. Access to the remainder of the SYSTEM TAPE is achieved through the MONITOR.

The utility routines on the SYSTEM TAPE are selected with a coding system. Each utility routine is given a code[1]. The code is placed in the test word and the computer is started at 12002 (octal) or 12005 (octal) (rewind before search).

The MONITOR will now search the SYSTEM TAPE for the selected routine and read the routine into core when the search is completed.

The AMP, AIDEX, and ASYS LIBRARY records are accessible through MONITOR control statements. These statements, which may come from the typewriter or the paper tape reader, are described in detail in Sections II - IV.

[1] A table of codes and utitlity routine names is presented in Appendix A.

II. ASYS MONITOR Jobs

A "job: is the work done by the MONITOR on a group of smaller programs resulting in a larger program in core 0 where all of the smaller programs are linked. Operations on these smaller programs must be performed in a specific order with the monitor.

1) Job Initialization

2) Assemblies and/or Compilations

3) Loading of the relocatable binary output

4) Library tape search

5) Corrections to loaded program

6) Execution or punch-off of the linked programs.

Job Initialization

At the beginning of a job the monitor is initialized by starting the computer at 12001 (octal) or by reading the monitor record into core 1 with the "read-in-monitor" paper tape.

When the MONITOR is initialized:

1) The SYSTEM relocatable binary output tape (TU no. 2) is rewound for new assemblies or compilations.

2) The LOADER origin is reset to 100 and the LOADER's symbol table is initialized.

3) The typewriter is set to lower case, black and a carriage return is typed.

4) Control is transferred to the Job Continue entry.

Job Continue

To restart the MONITOR without re-initialization, the computer is started at 12000 (octal).

The MONITOR will read the next control statement whenver Control is transferred to Job Continue or the computer is started at 12000 (octal).

III. ASYS MONITOR Control Statements

The MONITOR receives control statements in the form of lines of text from either the typewriter or paper tape reader.

Typewriter Input

Typewriter input is assumed:

1) If bit 0 of the test word is up

2) If bit 0 of the test word is down and the reader is either unloaded or off.

Mistakes made while typing control statements may be corrected by backspacing over them and retyping. Multiple backspaces may be used, but if an attempt is made to backspace too far (beyond the beginning of the buffer), a space is typed out, and the backspace is otherwise ignored. The command is not interpreted until the type-in of a carriage return. It is impossible to backspace over a carriage return.

Paper Tape Reader Input

Paper tape input is assumed:

when bit 0 of the test word is down and the reader is both loaded and on.

Mistakes made while preparing the command tapes on a flexowriter may be corrected by punching delete holes over error punches.

Control Format

The MONITOR controls are symbols whose first three characters are used to select the appropriate MONITOR function. These symbols may be preceded by any number of spaces, tabs, or periods. Type A control symbols are followed by argument lists. Type B control symbols may be followed by arbitrary comments. At the termination of the action initiated by most control statements[2] control is returned to Job Continue.

IV. ASYS MONITOR Command Repertoire

The type of control statement (A or B) is shown in parenthesis to the left of each control statement heading.

1 (A) version n

The version number of the AMP-AIDEX system desired is typed after using the "read-in-monitor" paper tape or a library search (at the end of a library search the SYSTEM TAPE is rewound). If the version statement is not used, the normal version (version 1) is assumed. The version control permits the selection of modified versions of the system with only one SYSTEM TAPE. The MONITOR rewinds and then forward spaces the SYSTEM TAPE (n-1) x 2 files whenever version is used.

Versions presently available:

Version 1	FPT-28 library
Version 2	FPT-18 library

2 (B) job

The job control provides another way to initialize the MONITOR. This control will:

1) rewind the system relocatable binary output tape (TU no. 2)

2) read the MONITOR record into core 1 hence initialize the MONITOR

3) transfer control to job continue.

[2] exceptions are shown in next section.

3 (B)

amp

The amp[3] control statement selects the AMP language. AMP pass 1 is read from the SYSTEM TAPE, and AMP proceeds to assemble a symbolic program from the paper tape reader. It is convenient to make the first statement of every AMP symbolic:

" ...   amp   "

This line will cause the MONITOR to automatically select AMP.

4 (B)

aidex

The aidex control statement selects the AIDEX language. AIDEX pass 1 is read from the SYSTEM TAPE, and AIDEX begins compilation of the symbolic program from the paper tape reader. It is convenient to make the first statement of every AIDEX symbolic:

" ...   aidex   "

This line will cause the MONITOR to automatically select AIDEX.

5 (B)

load

The relocatable binary magnetic output tape is loaded into core 0 with the load statement. All of the assemblies are compilations since TU no. 2 was rewound[4] last are loaded. AMP and AIDEX write an END OF FILE on the binary magnetic tape after each assembly or compilation. This END OF FILE is written over by each new assembly or compilation during the same job. The loader rewinds TU no. 2, and loads all programs until it reads the END OF FILE, then it returns to job continue. Since the binary magnetic tape's END OF FILE is always written automatically with the AMP-AIDEX SYSTEM,

[3] The "decal" control statement will also select AMP.

[4] TU no. 2 is rewound by all methods of job initialization.

it is possible to load the binary magnetic tape as often as desired even after a new job initialization.

6 (B) reload

The reload statement is identical in effect to the load statement except the LOADER is initialized first, then the magnetic tape is loaded by reload.

7 (B) bpunch

The relocatable binary magnetic output tape may be reclaimed in punched paper tape form with the bpunch statement. All of the programs assembled and/or compiled will be punched, separated by 256 lines of tape feed. The statement,

" bin "

is punched in FIO-DEC code at the beginning of each binary paper tape. Thus, when the punched paper tape relocatable binaries are presented to the ASYS MONITOR (by starting the computer at job continue), the " . bin " line selects the LOADER which proceeds to load the binary paper tape. At the end of each program loaded, return is to job continue. If the next control read is " . bin " , the LOADER is again selected, thus multiple programs on the same paper tape will be automatically loaded one by one until the reader is emptied. Paper tape binary programs may be loaded either before or after the magnetic tape is loaded by the load statement.

8 (A) porg expression

Binary programs are loaded sequentially, but this may be changed by setting the program origin in one of three ways:

1) An "org" statement in an AMP assembly

2) A reset from the test word switches if switch 2 is up.

3) A reset from the expression following the porg statement. This expression may be a simple octal number or may involve address arithmetic with system symbols,

e.g.

```
"      porg      1200      "

"      porg      matmul + 400   "
```

9 (B) library

The library control mandates a search of the library tape whether programs are needed or not. If a desired program entry is found, the program will be loaded.

10 (B) complete ?

If all the necessary programs have been supplied, the current loader origin is typed out. If programs are missing, the missing program entries are typed followed by the current loader origin.

11 (A) define symbol, expression

Occasionally system symbols are missing from a loading but it is necessary to execute the incomplete job. Such missing symbols may be defined with the define control.

e. g.

```
"   define     routine A,    routine B    "
```

Now all calls to routine A will be routed to routine B.

Parameters can be supplied at load time with the use of define.

e. g.

```
"    define    tape, 1    "
```

will put a 1 into

" law tape " where "tape" has been declared a system symbol but left missing in the job until define was used.

12 (B) <u>link</u>

The quick way to complete a job with missing system symbols is accomplished with the link control statement. This control sets aside one zeroed location for each missing system symbol at the current loader origin (incremented by one for each location linked). All calls to the missing system symbol are linked to its individual location.

13 (A) <u>tran</u>sfer expression

The transfer control does the work of a combination of other controls. First a check is made for missing system symbols. If symbols are missing, library search is made. Then there is another missing system symbol check. If symbols are still missing after the library search, the list of missing symbols is typed; otherwise, the MONITOR transfers control to "expression" which is a core 0 address. The transfer shuts off the extend mode and uses core 0's 7776 and 7777[5] to execute the transfer. The expression may be an absolute octal address or an expression with system symbols. This control will also set up the jump punch address for punch-offs of core using the punch control statement.

14 (A) <u>stra</u>nsfer expression

The stransfer control has the same effect as transfer, but the transfer is not executed (return is to job continue).

15 (A) <u>jmp</u> expression

The jmp controls both sets up the jump punch for the punch control

[5] 7776, lem
7777, jmp expression

and executes an unconditional jump to the core 0 address, supplied by expression. No library searches or missing symbol checks are made.

16 (A) sjmp expression

The sjmp control sets up the jump punch for the punch control but does not execute the jump. (return is to job continue)

17 (A) punch first, last, jump

The punch control calls a modified version of high binary punch and load which punches off selected areas of core with a short read-in-mode loader. If punch is used with no arguments, all of core loaded by the ASYS LOADER during this job will be punched. If first and last (which may be any address expression with or without system symbols) are supplied, all of core between these addresses will be punched. If the jump argument is given, a jump address will be punched after the data block on paper tape. If no jump argument is given but jmp, sjm, tra, or stra were used during this job, the address specified by jmp, sjm, tra, or stra will be punched after the data block. If neither the jump argument is present nor jmp, sjm, tra, or stra were used during this job, no jump address is punched.

18 (A) dpunch first, last, jump

The data punch control causes the punch-off of core from first to last as a data block with no read-in-mode loader. If a jump is specified, it is punched. This control is used after a normal punch control to avoid duplicate punching of the read-in-mode loader. No jump address is specified until the last dpunch control is used.

e.g.

```
punch     a, b
dpunch    c, d
dpunch    e, f, start
```

19 (B) hlt

The hlt control, when read from the paper tape reader, causes the MONITOR to halt. Continue returns control to job continue. If sense switch 3 is up the monitor will halt whenever it reads a FIO-DEC stop code from the paper tape reader. Stop codes are ignored if sense switch 3 is down.

20 (B) editor

The editor control calls the ASYS EDITOR for magnetic tapes. The EDITOR gets its input from the MONITOR input routine; therefore, EDITOR input may be from either typewriter or paper tape reader. The functions of the EDITOR are described in the next section.

21 (B) zero

The zero control stores 0 in every register of the lower core.

22 (B) adt

The adt control calls the ADT debugging system into core 1 and transfers control to ADT. (ADT is discussed in a separate manual).

23 (B) spu

The LOADER symbol table, symbol table pointer, and value of the current LOADER origin are punched out on paper with the spu control. The punch-out is performed by modified high binary punch and load. A short read-in-mode loader is punched out to load the paper tape. This paper tape may be read into core 1 for symbolic debugging with ADT after the LOADER symbol table has been changed (e.g., At some later time when debugging a program from a read-in-mode tape the symbol table may be restored with the spu paper tape.) The computer will halt after reading in the spu tape.

24 (B) save

The entire contents of core 0 are dumped onto the scratch tape mounted on TU no. 3 using the save control.

25 (B) restore

The previous contents of core 0 are reclaimed from the scratch tape mounted on TU no. 3 using the restore control.

V. ASYS EDITOR

The ASYS EDITOR performs many useful magnetic tape functions using a command structure very similar to the MONITOR control statement format. The appearance of an initial period in the line of an EDITOR command returns control to the MONITOR which interprets the rest of the statement as a MONITOR command. The EDITOR has been placed in the area of core where the AMP, AIDEX, and LOADER symbol tables build. Therefore, the EDITOR should only be called when the MONITOR has been read in and no assemblies, compilations, or loadings have taken place. The EDITOR was placed in this area because of storage capacity limitations.

All arguments of EDITOR commands must be octal numbers.

EDITOR Command Repertoire

1 rewind unit 1, unit 2, ...

The tapes on the units specified are rewound. As many arguments as desired may be given to rewind.

2 fws unit, count

The tape on the specified unit is forward spaced; count number of blocks.

3 bks unit, count

The tape on the specified unit is back spaced; count number of blocks.

4 write unit, ia, fa + 1

A block is written on the magnetic tape in odd parity on the specified unit from core address ia to core address fa.

5 read unit, ia, fa + 1

A block is read from the magnetic tape in odd parity on the specified unit from core address ia to core address fa.

6 iwrite unit, I.D., ia, fa + 1

A block with 3-word identification header (I.D., ia, fa + 1) is written on the magnetic tape in odd parity on the specified unit. The I.D. is a number code for search and identification purposes.

7 iread unit, I.D.

The tape on the specified unit is searched for the block headed by the specified I.D. When the block is found, it is read into the ia (specified by the second word of the header) to the fa (specified by the third word of the header). The tape must, therefore, be written with the 3-word header (as with iwrite) and each I.D. for sequential records must be numerically 1 greater than the previous I.D.

8 wef unit

An I.B.M. End of File mark is written on the specified tape.

9 fsf unit, (count)

The specified tape unit is forward spaced (count) number of I.B.M. End of File marks. If count is not given, the count is assumed to be 1. The unit is stopped immediately after the last End of File mark read.

10 copy read unit, write unit, n blocks

The specified number of blocks are copied from read unit to write unit. All of core 0 and 100 core 1 cells are used as the copy buffer; thus, the blocks to be copied must be ≤ 10100 words long. If an End of File mark is encountered on the read unit, the copy operation halts. The End of File mark is not copied.

11 fcopy read unit, write unit, (n files)

All of the blocks to and including the next specified number of file marks are copied from the read unit to the write unit. If the file count is not given, 1 file is assumed.

12 ucopy read unit, write unit, update value, (count)

The action of ucopy is similar to copy but each block has its I.D. (first word) changed by the addition of update value to it. This control provides the capability of adding or deleting records from tapes recorded with sequentially increasing I.D. headers.

13 ufcopy read unit, write unit, update value, (f count)

This command is a combination of ucopy and fcopy.

14 rmon unit

The MONITOR record is read into core 0 from the specified unit for debugging the MONITOR.

15 wmon unit

The MONITOR record is written from core 0 onto the tape on the specified unit. The 3-word I.D. block will, however, specify a core 1 positioning of the MONITOR. This control is used after rmon or after assembling and loading the MONITOR into core 0 or after reading in the MONITOR from paper tapes marked "core 0 MONITOR." The tape on the specified unit must be properly positioned before using wmon.

VI. ASYS Error Diagnostics

diagnostic	meaning
cks	checksum error in relocatable binary input (from either a tear in paper tape or a magnetic tape error)
ctx	LOADER constants table exceeded (128 absolute constants/program loaded maximum)
ffe	format error in a MONITOR control statement
fpe	paper tape parity error in a MONITOR or EDITOR command
icd	illegal code in relocatable binary input (from either a tear in paper tape or a magnetic tape error)
ief	illegal end of file on the SYSTEM relocatable binary input magnetic tape (from a magnetic tape error)
ile	illegal EDITOR command
ils	illegal MONITOR statement
mds	multiply defined loader symbol (the symbol name follows)
mmp	missing main program and tra, stra, jmp, or sjmp was used without an argument field
mpe	missing program entries (a list of these entries follows)
olp	program attempts to overlap core 0 and use core 1
utx	LOADER usage table exceeded (128 external system symbols/program loaded maximum)

VII. ASYS APPENDICES

Appendix A

ASYS SYSTEM TAPE STRUCTURE

*routines should be selected only with monitor controls

	UTILITY ROUTINE NAME	Test Word Code
*	ASYS MONITOR	0
*	AMP Pass 1	1
*	AMP Pass 2	2
*	AMP Symbol Table Dumps	3
*	AIDEX Pass 1	4
*	AIDEX Pass 2	5
*	AIDEX Pass 3	6
*	AIDEX Symbol Table Dumps	7
*	AIDEX Pass 4	10
*	ASYS MONITOR RESTORE	11
*	ADT	12
	Expensive Typewriter	13
	Expensive Desk Calculator	14
	Compatible Reproduce	15
	Tape Duplicator	16
	Aid	17
	Analyzer	20
	Prim	21
	ASYS LIBRARY TAPE MAKER	22

* END OF FILE MARK

* ASYS LIBRARY

* END OF FILE MARK

Appendix B

Maintenance of the SYSTEM TAPE

1) The MONITOR record

The MONITOR should seldom need maintenance. However, MONITOR control statements can be added by inserting the new code in the MONITOR table with a pointer to the subroutine for the new code. The MONITOR is now reassembled (for correct linkage the last three assemblies should be the EDITOR, mgfws and mgbks, mgwrti and mgrdi), reloaded, then rewritten on the rewound new SYSTEM TAPE with wmon.

e.g. After assembling the new MONITOR and loading it into core 0 the following sequence of commands should be executed:

```
" job (restore the EDITOR into core 1)

  editor (call the EDITOR)

  rewind 2, 1 (position tapes, 1 is the old SYSTEM
               TAPE 2 is the new SYSTEM TAPE)

  wmon 2 (write the new MONITOR record on tape 2
          from core 0)

  fws 1, 1 (space over the old MONITOR record tape 1)

  fcopy 1, 2, total files (copy the rest of the old
                           SYSTEM TAPE onto the new
                           SYSTEM TAPE)

  rewind 2,1 "
```

The value of the total files is the number of versions of ASYS on the SYSTEM TAPE times 2.

2) AMP records

It will often be useful to add SYSTEM MACROS to AMP. These macros are left in the symbol table and available for all users. The technique for adding these macros is very simple.

First a symbolic tape is prepared with the desired new macros. This tape is assembled with AMP

e. g. to add the call macro prepare a tape with

```
    " call   macro  name  params

             jsp name

             iff 1,_C_ params

             dup 3, 0

             irp params

             loc params

             irp

      call   end     "
```

This tape is subject to normal assembly.

After assembly of the new macros the new SYSTEM TAPE is made by executing the following sequence of commands

```
"    job        (restore the EDITOR into core 1)

     editor     (call the EDITOR)

     rewind 1,2,3 (position tapes, 1 is the old SYSTEM
                   TAPE, 2 is the new SYSTEM TAPE, 3 is
                   the save macros tape)

     write 3, 6500, 10000* (save the new macros)

     copy 1,2,1 (copy the old MONITOR record on the new
                tape)

     iread 1, 1 (read the old AMP pass-1 into core 0)

     read 3,6500,10000* (read in the new macros)

     iwrite 2,1,0,10000* (write the new AMP pass-1 on the
                         new tape)
```

*This top value should be obtained by examining the symbol table pointer (6501) after the assembly of the new system macros.

```
fcopy 1,2, total files (copy the rest of the old tape
                        onto the new tape)

rewind 1,2,3
```

Only the first record (Pass 1) of AMP is changed to add to the AMP symbol table.

Appendix C

ASYS LIBRARY Tape Maker

The ASYS LIBRARY Tape Maker is a routine for creating and editing ASYS LIBRARY tapes. Relocatable binary programs are read from the paper tape reader, converted to library tape format, and written on magnetic tape. Routines from an existing magnetic tape library may be merged and/or intermixed with new paper tape binary programs to form an expanded library tape.

The order of programs on the library is very important. If subroutine A calls subroutine B calls subroutine C, the routines must be placed on the library in the order:

A,B,C[1].

Using the ASYS LIBRARY Tape Maker

Starting the computer at 100 with ss 1-6 down will initialize the ASYS LIBRARY Tape Maker. The computer will type "r = "; then wait for an octal number from the typewriter which is the tape unit number on which the existing LIBRARY is mounted, (r means read from). Now the computer will type "w = "; then wait for an octal number from the typewriter which is the tape unit number on which the new LIBRARY is mounted, (w means write on). Next the computer waits for commands from the typewriter. The read unit must be positioned at the beginning of the LIBRARY being edited before the read unit is used. Starting the computer at 100 with ss 1 up will skip the initialization sequence and cause the computer to wait for commands.

[1] A rare ordering problem occurs when A calls B yet B calls A, and either A or B (not both) are called from the main program. Here duplication is necessary on the library (either ABA or BAB).

ASYS LIBRARY Tape Maker Commands

Command	Description
a	append
c	copy
d	delete
e	editor
f	file
i	initialize
l	list
s	stop

append (a) The computer reads binary paper tape programs from the reader adding them to the new library. (the write tape) Normally the computer will continue to add programs to the library as soon as the paper tapes are placed in the reader until a new command is typed. If ss 3 is up, the computer will add only one program to the library, then return to the wait for new commands loop.

copy (c) The computer copies all programs from the read tape (except those which are to be deleted) onto the write tape until a routine is copied which is in the stop list or the end of the read tape library is encountered. The "a" and "c" commands may be used in any order.

delete (d) A list of entry names (one entry per program deleted) is typed to cause the deletion of the named routines from the new library. These routines are deleted during the copy. The list is terminated by a slash (/).

editor (e) The ASYS magnetic tape EDITOR is called. The read unit can be propoerly oriented with the EDITOR (using "FSF"), etc.

file (f) The library end code as well as an I.B.M. END OF FILE is written on the w unit. This command must be the last operation performed on the new library.

initialize (i) The stop and delete lists are emptied, pointers are reset, and, in general, everything done by starting at 100 with ss 1 down is performed except the old "r" and "w" are retained.

list (l) This control is used to list all of the program entries on the read or write tape. After typing "l", either "r" or "w" must be typed to designate read or write tape respectively. The selected tape must be positioned at the beginning of the library for "l" to work properly. If ss 2 is up, the program lengths are also typed out.

stop (s) A list of entry names (one entry per "stop after" program) is typed to cause the termination of the copy after any of the named routines are encountered. The named routine will be copied unless it is also in the delete list. When the copy operation ceases, the computer waits for a new command. Usually a binary paper tape program is inserted here (using append (a) , then the copy is continued (using a copy (c)).

Example of using the ASYS LIBRARY Tape Maker

A new SYSTEM TAPE is to be made with a new "fadd" subroutine for the FPT-28 library (version 1) and with the subroutine "random" added to the FPT-28 library. The following sequence of commands should be executed:

```
    job         (restore the EDITOR to core 1)

    editor      (call the EDITOR)

    rewind 1,2 (position tapes, 1 is the old SYSTEM TAPE
                                2 is the new SYSTEM TAPE)

    fcopy 1,2,1 (copy 1 file from the old to the new tape)

(now read in the ASYS LIBRARY Tape Maker indicate r = 1, w = 2)

    d fadd/  (delete the old "fadd")

    s fadd/  (stop after encountering "fadd")

    c        (copy from 1 to 2 the library up to "fadd")

    a        (the binary paper tape of the new "fadd" is
              placed in the reader and added to the new
              library)

    c        (the remainder of the old library is copied
              on the new library)

    a        (the binary paper tape of "random" is placed in
              the reader and added to the new library)

    f        (the library end code and end file records are
              written on tape 2)

    e        (calls the EDITOR) fcopy 1,2, remaining files
              rewind 1,2

    fsf 2,1  (space out the new SYSTEM TAPE to the library file)

(now start the computer at 100 with ss - 1 up)

    1 w       (the entries to routines on the new library are listed
               on-line)
```

Appendix D

Sense Switch Control of ASYS

ss 1

Throughout the AIDEX SYSTEM sense switch 1 is used to control typeouts. When MONITOR control statements are read from paper tape, they are typed on-line unless either ss 1 is up or TW bit 2 is up. The LOADER may typeouts are deleted if ss 1 is up. AMP and AIDEX on-line symbol table dumps are deleted if ss 1 is up. Error diagnostics are unconditionally typed.

ss 3

Sense switch 3 up tells the MONITOR to stop reading paper tape on stop codes and halt. Sense switch 3 down tells the MONITOR to ignore stop codes and eject paper tape.

TW bit 1

When TW bit 1 is up, the MONITOR types out the date on-line before all loadings, assemblies, or compilations. The date is stored in register 17777.

	First 6 bits	Next 6 bits	Next 6 bits
17777			
	month in octal	day in octal	year - 1964 in octal