

SENTRY-400

TESTER OPERATING SYSTEM MANUAL

TABLE OF CONTENTS

SECTION I INTRODUCTION

SECTION II GENERAL SYSTEM DESCRIPTION

2.1	Introduction2-1
2.2	System Objectives2-1
2.3	System Operation2-1
2.3.1	Manual Mode Operation2-2
2.3.2	Monitor Operation2-2
2.3.3	Automatic Mode Operation2-2
2.4	Program Sections2-2
2.4.1	Interpreter2-2
2.4.2	Datalogger2-3
2.4.3	Command Processor2-3
2.4.4	Test Programs2-3
2.5	Peripherals Off-Line2-3
2.6	Memory Allocation2-4

SECTION III COMMAND RECORDS

3.1	Introduction3-1
3.2	Command Format3-1
3.3	Operand Parameter Types3-2
3.3.1	String3-2
3.3.2	Name3-2
3.3.3	Integer3-2
3.3.4	Floating Point Number3-2
3.3.5	Special Characters3-3
3.4	Noise Words3-4
3.5	Error - Recovery3-4

TABLE OF CONTENTS (Continued)

SECTION IV COMMAND DESCRIPTIONS

4.1 Introduction4-1
4.1.1 Parentheses4-1
4.1.2 Brackets4-1
4.1.3 Constant Names4-1
4.1.4 Variables4-2
4.2 Load4-2
4.3 Title4-3
4.4 Pause4-3
4.5 Modify4-4
4.6 Datalog4-5
4.7 Set4-6
4.8 Clear4-7
4.9 Switch4-7
4.10 Dopsy4-8
4.11 Sync On/Off4-8

4.12 Alter4-9
4.13 MACT

APPENDIX

A Diagram of Topsy RoutinesA-1
B Control Flow ChartB-1
C Topsy Terminal ErrorsC-1
D Allocation of Main Memory Driving Topsy ControlD-1

SECTION I

INTRODUCTION

The operating system for the Sentry 400 testing consists of a small executive, the TOPSY (Tester Operating System) main program which controls three routines:

- 1) an interpreter to execute user test programs;
- 2) a data-logging routine to output test data; and
- 3) a command processor which enables the user to control the test system (see Appendix A for diagram of the routines).

Other features included within TOPSY are:

- 1) dynamic core management;
- 2) program error detection;
- 3) a maximum object program capability of approximately 132,000 24 bit words; and
- 4) a maximum number of approximately 350 accessible test programs (averaging 1430 words each) from the on-line disc storage.

TOPSY is not a stand alone program. The input for TOPSY must come from the FACTOR compiler via either punched cards (or paper tape) or disc.

In addition, TOPSY comprises a functional tester executing the user's FACTOR program, and a monitor, which allows partial alteration of the user's FACTOR program during execution. The user's program must reside on disc in Data format for the FST-1 system.

The two major areas to be discussed in this manual are:

- 1) the details of the systems organization which either affect or are related to the user's program; and
- 2) the commands by which a user communicates with the system.

For additional details, the user should consult the FACTOR Manual, the Assembler Manual, the DOPSY Manual or the FST-1 Systems Manual.

The minimum configuration in which the system will operate is:

12K Main Memory
Disc
Teletype
Card Reader

SECTION II

GENERAL SYSTEM DESCRIPTION

2.1 INTRODUCTION

This section discusses in general terms the organization of TOPSY. It includes a description of the System Objectives; System Operations; Program Sections; and Memory- Allocations.

2.2 SYSTEM OBJECTIVES

TOPSY is designed to aid the user's interactions with the test system, and also to supplement the capabilities of the hardware. To achieve these objectives, TOPSY executes user test programs in an interpretive mode. This allows the unsophisticated user to be concerned only with the logic of the test program and not with the intricacies of the programming-system and the test-system hardware.

In order to maximize the performance (test rate) of the system, TOPSY user programs are "read-only". This minimizes the number of page transfers (disc accesses) required to execute a program. The variables in arithmetic statements are stored, on a first-in, last-out basis, in an area of core memory called the "run-time stack".

TOPSY controls the test system hardware, datalog output, and command input. In the interpretive mode of operation, TOPSY first analyzes instructions and then initiates hardware functions. While the hardware is responding, the CPU is essentially free to look-ahead and begin analyzing the next instruction to be executed. This procedure overlaps hardware delay time with processing time and enhances the test rate of the system. All functional test sequences are directly accessed from memory by the tester hardware.

2.3 SYSTEM OPERATION

TOPSY is capable of multiplexing four test stations on an ordered sequence (4, 3, 2, 1) operation. Manual, monitor or automatic operations can be independently exercised at each station. To maximize efficiency from a system viewpoint, control is given to the next station in the above sequence whenever TOPSY cannot execute the next test of the presently active test sequence; i.e., a pause, end of test, terminal error, etc. The manual, monitor and automatic modes of operation are discussed as follows:

2.3.1 Manual Mode Operation * See attached note

In the manual mode TOPSY will execute the first test, store away the updated system work area and then return to the scanning routine. A subsequent return by this test station to the TOPSY testing system will initiate testing from the last test if another station had not yet executed, or, if another station had executed, it will execute from the first test up to the next sequential test.

2.3.2 Monitor Operation

In the monitor mode TOPSY allows the user's input commands to be processed. This command processing is independent of the rest of the test system, (which allows commands to be entered while TOPSY is executing the current test program).

2.3.3 Automatic Mode Operation

In the automatic mode the user's tests are executed sequentially until either a pause or an EOT is encountered. In case of a pause TOPSY returns to the scanning routine after storing the updated system work area. In the case of an EOT the system is reset for execution of the next test sequence and then the system work area is stored away. Thus, upon a subsequent re-entry after an EOT the first test is initiated.

2.4 PROGRAM SECTIONS

The TOPSY main program serves as an executive, giving control to the program section which performs the next desired task. This is illustrated by the flowchart in Appendix B. There are three main TOPSY program sections. The

- 1) the interpreter;
- 2) the data logger; and
- 3) the command processor.

Each of the three program sections are described as follows:

2.4.1 Interpreter

The interpreter controls the test system hardware and analyzes the user test program, instruction by instruction. It executes all arithmetic statements in the user test program and monitors the status of the test system. Note that the interpreter is not activated unless the tester START has been pressed.

TOPSY MANUAL
ADDENDUM #1

In automatic mode operation, while a test is in process the Statement Number Display indicator on the test station control panel will read 1. This is used to indicate to the operator that the station is executing a test sequence.

At the End of Test, Pause, or Read, the Statement Number Display will indicate the number of the last instruction executed.

In automatic mode, a FORCE RESET in a program will reset the Statement Number Display to zero during the test, however, at End of Test or Pause the Statement Number Display will indicate the last executed statement number.

INTERNAL CORRESPONDENCE **RECEIVED**

APR 5 1971

R. E. HUSTON

April 1, 1971

To: File
From: H. Vitale
Subject: PROPOSED SENTRY-400 MANUAL MODE
OPERATION DESCRIPTION

CC: V. DiMucci
R. Huston
R. Forster
A. Smith
J. Spangberg
B. Routh

The SENTRY-400 provides two modes of manual operation when testing in a multiplexed environment. The modes are referred to as N RETURN and N+1 RETURN. The definition for these modes of operation are given below:

N+1 RETURN MODE

In this mode, when a station is executing single test statements in MANUAL, it will not indefinitely restrict testing at other stations. Hence, some other station may execute a test, thus interrupting the manual station. This interruption will cause power to be removed from the device under test since all reference and power supplies are in the mainframe and are multiplexed to other stations. When the manual station requests execution of its next test (test n+1) after being interrupted during its previous pause at test n, the tester automatically executes all tests from 0 to n+1. It is necessary to re-execute the test sequence since the device under test may be sequential (has memory) and when its power is removed it loses all knowledge of previous states. The manual station can then be advanced from test-to-test in this manner with or without interruption by other stations. The station does not execute from 0 to n+1 if it has not been interrupted from the pause at test n; rather it advances from n to n+1.

This mode is enabled by placing console switch 1 down on the FST-1 panel.

N RETURN MODE

This mode of operation is similar to "N+1 RETURN" with the exception that following an interruption, a manual station automatically executes from 0 to n upon request to continue testing (initiated by test station START). This mode is enabled by setting console switch 1 "up" on the FST-1 panel.

If rapid testing (automatic or manual) is being done at other stations, the manual station may not be able to advance past test n unless either;
1) other station operators momentarily halt testing or 2) FST-1 console switch 1 is put "down" to return to the N+1 RETURN mode.

HV

H. Vitale

HV/nr

2.4.2 Datalogger

The datalogger performs all output and data-formatting for test results, terminal errors, and user-programmed WRITE statements. The datalogger will not return control to the TOPSY main program until all requested output is initiated; however, testing will proceed while the output device is busy.

2.4.3 Command Processor

The command processor performs all input for both TOPSY commands and user-programmed READ statements. In addition, it processes the TOPSY commands and initializes the flags which control the operation of the test system, such as PAUSE ON STATEMENT n and PAUSE ON FAIL. The command processor is not activated unless the test system is in MONITOR mode.

Both the command processor and the datalogger are overlays which share the same core-memory area, hence they can never simultaneously be resident in core.

The user test program must be loaded using the TOPSY command LOAD. Once a program has been loaded, pressing tester RESET will restart the program at its first statement, at which point it waits for a START command.

2.4.4 Test Programs

The kind of files which TOPSY will process as test programs are DATA files. DATA records are both word-oriented and fixed-length. The DATA record size is declared at the time the file is created.

2.5 PERIPHERALS OFF-LINE

If a peripheral device is not ready when TOPSY tries to initiate an I/O operation to it, the message 'I/O DEVICE NOT SET' will be typed at the teletype (which is assumed to be on-line at all times). The CPU will NOT halt while in TOPSY. The low-order 6 bits of the A-register will display the device-number which corresponds to the peripheral as shown below:

Card Reader	40	Octal
Card Punch	50	Octal
Line Printer	60	Octal
Disc	70	Octal
Mag Tape	10X	Octal

The operation will be retried as soon as the peripheral is brought to a ready condition.

IMPORTANT

If the peripheral makes an un-recoverable error?

2.6 MEMORY ALLOCATION

The user test program is treated as data which the interpreter acts upon. The program is read into the READ-BUFFER at the top of core. This buffer is variable in size, because the RUN-TIME STACK steals space, whenever necessary, so as to provide additional variable-storage. (Appendix D illustrates the allocation of the main memory when TOPSY has control.)

In general, programs containing loops will execute more slowly than straight line programs, since a loop may cause a different "page" of the program to be read into the READ-BUFFER. Also, programs which perform a large amount of arithmetic computation will tend to run more slowly, since the RUN-TIME STACK will be stealing space from the READ-BUFFER, necessitating a greater number of disc-accesses for a given program length. These considerations apply most strongly to functional test sequences; if the arithmetic portion of a program is contained in a BLOCK not active while executing functional tests, it will not affect the test rate. Similarly, a loop around an entire sequence of functional tests does not affect the test-rate of that sequence.

SECTION III

COMMAND RECORDS

3.1 INTRODUCTION

This section discusses the Command Formats; Operand Parameter Types; Noise Words; and Error Recovery Messages.

The commands recognized by TOPSY may come from either the card reader or the teletype. Commands will not be accepted unless the test system is in the MONITOR mode; they are received as input via the I/O procedures CRI0 and TTRIO.

When entering commands from the keyboard, one should be aware of the limited editing facilities provided by TTRIO. These editing facilities are discussed in detail in the FST-1 Subroutine Library Manual. Briefly, however, TTRIO interprets the character produced by pressing CTRL B as the deletion of a single character, and that produced by CTRL L as a line delete. The former is indicated by echoing "<" for each occurrence, while the latter is indicated by a carriage return, line feed and ":".

Errors detected while processing a command, will force the Principal Input Device (PID) to become teletype keyboard (TTK).

If commands are entered through the card reader, the last card must be '/. SET TTK' for an exit from the command processor to occur.

3.2 COMMAND FORMAT

The format of the commands which the system recognizes is:

```
/. name    <operand parameters>
```

The format is free-field, except that the '/.' must be in columns one and two.

The remainder of the command record contains the command name, followed by the operand parameters. The command name is terminated with the first special character (anything other than a letter or digit and is ignored by the system).

The operand parameters occur after the command name and they must be separated from each other by a special character, such as a space or a comma.

3.3 OPERAND PARAMETER TYPES

The types of parameters that may occur in an operand are: string, name, integer, floating point number, and special characters. The meaning of each parameter type is a function of the command. Unless otherwise stated in a particular command description, the order of the parameters is irrelevant. Each of the parameter types are discussed as follows:

3.3.1 String

A string is a sequence of characters (other than a single quote) which are enclosed in single quotes. These are used to reference files. Only the first six characters are retained, except by the TITLE command.

Examples:

```
'TEST4'  
'*QRT'
```

3.3.2 Name

A name is a sequence of letters, including \$ or digits. The first character must be a letter. Only the first eight characters are retained.

Examples:

```
TTK  
MODIFY  
FAIL
```

3.3.3 Integer

An integer is a sequence of digits, which, if terminated by the letter B, is assumed to be octal; otherwise, they are interpreted as decimal. Only the low order 24 bits are retained.

Examples:

```
15  
40B  
7777777B
```

Note: 4095 is equivalent to 7777B.

3.3.4 Floating Point Number

A floating point number is a sequence of digits and special characters in any of the following forms:

+XXX	decimal integer
+YYYY	octal integer
+XX.	} decimal numbers
+XX.XX	
+XX	

$\left. \begin{array}{l} +XX.E+XX \\ +XX.XXE+XX \\ \pm.XXE+XX \end{array} \right\}$ decimal numbers with exponent

where X is a digit, 0-9
 Y is a digit, 0-7
 + may be omitted
 $E \pm XX \equiv 10^{\pm XX}$

Any deviations from the above forms will be treated as errors, as will those numbers which are outside the listed limits.

Examples:

<u>Form</u>	<u>Limits</u>
decimal integer	$-8388608 \leq n \leq 8388607$
octal integer	$4000000B \leq n \leq 37777777B$ (in two's complement form, including sign)
decimal number	The integer which is formed by ignoring the decimal point and any leading zeros. It must be a legal decimal integer. The actual number must be within the legal decimal number limits, including exponent. See below:
decimal number with exponent	$n = 0$ or $2.7105E-20 \leq n \leq 9.2228E18$
Examples:	29 16B 2840.7 14.3E-5

3.3.5 Special Characters

These are generally ignored, but they must be present, since they separate the parameters. Characters, other than letters or digits, are special characters.

Example:

,
;
+

3.4 NOISE WORDS

There are certain names which are used by the commands so as to specify certain options. FAIL, DCT, LOG, TTK and OFF are a few examples. Words and/or names, other than those "reserved" names recognized by the command, may be freely inserted to improve readability. Some examples of these are shown below. The words and characters shown underlined are the significant ones. Words, characters, etc., not shown underlined are considered noise and are ignored.

/. LOAD 'DT4X' STAT1
/. TITLE '3/4/69 RUN NO. 15 ARRAY TYPE 9440' STAT3
/. DATALOG ON LP ALL FGT DCT AND MEASURE STAT2
/. SET CR
/. PAUSE ON FIRST FAIL STAT1
/. PAUSE ON STATEMENT 15B STAT4
/. MODIFY ON 3.5E-6 SECONDS STAT4

3.5 ERROR-RECOVERY

Whenever the system encounters an error which requires user intervention, it types a message on the teletype. (The text describes the error condition.) Next, the PID is set to the TTK and the system types a ":". It then expects a command to be typed; this command may be either the same one, with the error corrected, or else a new one.

Below is a partial list of the error messages produced by TOPSY. (The error messages which are produced by the individual command processors are described in Section IV along with the commands.)

<u>Text</u>	<u>Descriptions</u>
WRONG SEQUENCE	The record is not properly identified as a command.
COMMAND?	The command name is not one recognized by TOPSY.
"NAME":SYS 11 ERR.	"NAME" is the overlay which cannot be located on the disc. DOPSY is restarted.

NOTE

The facilities of the Automatic Restart Routine (ARR) are used to both locate and load TOPSY's COMMAND-PROCESSOR and DATALOGGER overlays. ARR is also used to locate the user's test programs. The user should note that the disc I/O, which resides in ARR, executes a halt at 100B if it is unable to read the disc after 80 retries. Whenever the halt occurs, the contents of the program counter will be 101B.

SECTION IV
COMMAND DESCRIPTIONS

4.1 INTRODUCTION

This section describes each of the commands recognized by the system. The general form of each command is given, along with specific examples. The general form is described using the following syntax notation:

4.1.1 Parentheses

Parentheses, "(",")", are used to enclose items which are optional.

4.1.2 Brackets

Where a choice is to be made from a set, the items in that set will be enclosed in brackets, "[","]", and they will be separated by a slash.

Example:

[TTP/TTK/CR]

A "0" indicates that none of the elements of the set need be chosen.

An underlined item in a set indicates that the item is assumed, if "0" is selected.

4.1.3 Constant Names

Constant-names are shown in upper-case. They must always be written exactly as shown.

Examples:

ON
TTP
FAIL

SECTION IV
COMMAND DESCRIPTIONS

4.1 INTRODUCTION

This section describes each of the commands recognized by the system. The general form of each command is given, along with specific examples. The general form is described using the following syntax notation:

4.1.1 Parentheses

Parentheses, "(",")", are used to enclose items which are optional.

4.1.2 Brackets

Where a choice is to be made from a set, the items in that set will be enclosed in brackets, "[","]", and they will be separated by a slash.

Example:

[TTP/TTK/CR]

A "0" indicates that none of the elements of the set need be chosen.

An underlined item in a set indicates that the item is assumed, if "0" is selected.

4.1.3 Constant Names

Constant-names are shown in upper-case. They must always be written exactly as shown.

Examples:

ON
TTP
FAIL

4.1.4 Variables

Variable names and/or quantities are shown in lower case. Their values change with usage and must be supplied by the user.

Examples: string
integer
file name

Note: All commands, except "DOPSY" and "SET", must reference a station, i.e., "STAT(N)", where N = 1, 2, 3 or 4. If "STAT(N)" is not entered, an error message is generated.

Example of error message text: DUPL./MISSING PARM.

4.2 LOAD

General Form:

/. LOAD 'file name' STAT(N) (SAVE)

Description:

This command loads the user's test program from the named disc file and forces tester-reset. It clears all hardware and software, unless the SAVE option exists - in which case the hardware only is cleared.

Restrictions:

The file type of the test program must be DATA. The file must exist under the current job number and the file name must not start with a blank, be \$DIRECT, or \$ARR.

Error Messages:

<u>Text</u>	<u>Description</u>
IMPROPER NAME	The file name either contained a blank or else it was not terminated with a single 'QUOTE'.
MISSING/IMPROPER FILE	The file name either cannot be found in the directory or else it is not of DATA type.

4.3 TITLE

General Form:

/. TITLE 'up to 64 characters' STAT(N)

Description:

This command enters a title line which will be output prior to any DATALOG output. The title will be output each time the addressed station is enabled.

Restrictions:

The two single quotes must both be present. A maximum of 64 characters may appear between them. Also, the command, CLEAR, destroys the current title.

Error Messages:

Text

Description

MISSING/DUPLICATE PARAMETERS

No single quotes were detected.

4.4 PAUSE

General Forms:

/. PAUSE FAIL [ON/OFF] STAT(N)
/. PAUSE integer [ON/OFF] STAT(N)

Description:

PAUSE initializes flags which cause the interpreter to pause (stop testing) when it detects the first failure after the START switch is depressed while in the AUTO mode. It also occurs after executing a specified statement number. In both cases, testing is resumed by pressing tester START. The FAIL and statement-number flags may be initialized as either ON or OFF, independently.

Restrictions:

Either ON or OFF must be present, but not both. Also, either a FAIL or an integer must be present, but not both. The integer must be in the range $n < 177777B$.

Both the FAIL and statement-number flags are turned OFF by the CLEAR command, or PAUSE OFF STAT(N).

Since the statement number is both displayed and listed in octal, it is recommended that the "integer" be written in octal format.

Error Messages:

<u>Text</u>	<u>Description</u>
MISSING/DUPLICATE PARAMETER	The parameters, ON and OFF both appeared or were missing or else FAIL and an integer both appeared or were missing.
INVALID STATEMENT NUMBER	The integer was either negative* or zero, or else it was greater than 177777B.

4.5 MODIFY

General Form:

/. MODIFY [ON floating-point number/OFF] (DC) STAT(N)

Description:

MODIFY alters the value of the programmed time-delay from that specified in the SET DELAY statements in the user's program to the value specified by the floating-point number. The new time delay is assumed to be in seconds.

The "currently active delay" is defined to be the last SET DELAY statement (with or without DC) executed. If no SET DELAY statements have been executed, and if the MODIFY DC condition is enabled (ON), the "currently active delay" is defined to be a standard delay with the value of the last MODIFY command.

The MODIFY command without the DC modifier alters the values of those SET DELAY statements which are without DC, when the SET DELAY statements are executed. If the "currently active delay" is not type DC, the programmed time-delay will be altered to the floating-point value (scaled as a functional-test delay) as soon as the MODIFY command is issued. It will remain altered for all subsequent SET DELAY statements which are without DC.

The MODIFY DC command alters the values of SET DELAY DC statements when they are executed. If the "currently active delay" is type DC, the programmed time-delay will be altered to the floating-point value (scaled as a DC delay) as soon as the MODIFY DC command is issued. It will remain altered for all subsequent SET DELAY DC statements. Execution of SET DELAY statements without DC is not affected.

Prior to the execution of any SET DELAY statements (with or without DC)**, the MODIFY command will load the time-delay register (TD) with the value of the floating-point number (delay).

* Possible only with octal numbers; e.g., 77777777B.

** This is true whenever tester RESET is pressed.

The MODIFY OFF command, without DC, will return control to those user-programmed SET DELAY statements which are without DC. This occurs when the next SET DELAY statement is reached, or when RESET is enabled. In addition, if the "currently active delay" is not DC, the value from the last SET DELAY statement, without DC, will be restored.

The MODIFY DC OFF command will return control to the user-programmed SET DELAY DC statements when the next SET DELAY DC statement is reached, or when RESET is enabled. In addition, if the "currently active delay" is DC, the value from the last SET DELAY DC statement will be restored.

The CLEAR command, along with its other functions, has the same effect as does turning off both MODIFY DC and MODIFY without DC.

Restrictions:

One of the ON or OFF parameters must appear, but not both. The floating-point number must follow the syntax described in Section III, and it must be within the limits shown below:

MODIFY without DC: $n = 0$ or $.35 \times 10^{-6} < n < 5.734 \times 10^{-3}$ seconds.
 MODIFY DC: $n = 0$ or $350 \times 10^{-6} < n < 5.734$ seconds.

Error Messages:

<u>Text</u>	<u>Description</u>
INVALID NUMBER	The time delay was not within the limits shown above, or else the floating-point number did not follow the form described in Section III.
MISSING/DUPLICATE PARAMETER	ON and OFF appeared together, neither appeared, or else a number appeared with OFF.

4.6 DATALOG

General Form:

/. DATALOG (DCT) (FCT) (LOG) (MEASURE) (TRIP) [LP/TTP/POD/0] STAT(N) (OFF)

Description:

DATALOG initializes flags which specify options requested for datalogging:

log all trip failures:	TRIP
log all DC test failures:	DCT
log all functional test failures:	FCT
log all programmed LOG statements:	LOG
log all measurements:	MEASURE
direct data log output to specified device:	LP, etc.
defeat datalogging	OFF

any combination may be specified

All options which are not requested will be turned OFF.

Restrictions:

The options specified by the DATALOG command are turned off by the CLEAR command. Also, the datalog output device must be an output device.

Datalogging on TTP is suppressed in monitor enable mode.

Error Messages:

<u>Text</u>	<u>Description</u>
WRONG I/O	An input device was specified for datalog output.

4.7 SET

General Form:

/. SET [CR/TTK/TTR/PID/0]

Description:

SET temporarily alters the primary input device (PID) to the device specified. Since PID is the source of TOPSY commands, the SET command can be used to direct TOPSY to read commands from the card reader (CR).

Restrictions:

The TOPSY command will alter PID from its current value before restarting DOPSY. Note, while actively executing a user's test program, PID will always be TTK.

NOTE

If commands are entered from the card reader, the last card must be '/. SET TTK' for an exit from the command processor to occur.

Error Messages:

<u>Text</u>	<u>Description</u>
WRONG I/O	An illegal I/O device was specified.

4.8 CLEAR

General Form:

/. CLEAR STAT(N)

Description:

CLEAR resets all modified conditions established by TITLE, PAUSE, MODIFY, DATALOG and SWITCH. CLEAR restores the last test delay from the user's test program and resets the flags which control functional-failure and DC-failure header-output. CLEAR forces the output of a blank title. In addition, it sets the global variable 'SWITCH' and GLOB1 through GLOB9 to zero.

Restrictions:

Does not alter PID or POD.

Error Messages:

None.

4.9 SWITCH

General Form:

/. SWITCH floating-point number STAT(N)

Description:

SWITCH sets the global variable 'SWITCH' equal to the value of the floating-point number. User's test programs may test this variable to control program execution. The CLEAR command resets 'SWITCH' to zero. The current value of 'SWITCH' is kept active from one test sequence to the next.

Restrictions:

The floating-point number must follow the syntax outlined in Section III, and it must be within the limits specified there.

Error Messages:

<u>Text</u>	<u>Description</u>
MISSING/DUPLICATE PARAMETER	No number was specified.
NUMBER SYNTAX	The number was illegal.

4.10 DOPSY

General Form:

/. DOPSY

Description:

DOPSY writes present TOPSY core onto disc file \$TOPSY. In addition, it resets the tester and calls the DOPSY monitor system into core memory. When TOPSY is recalled, it is in the last state it was in before calling DOPSY.

Restrictions:

None.

Error Messages:

None.

4.11 SYNC ON/OFF

General Form:

/. SYNC ON statement number/OFF STAT(N)

Description:

'SYNC ON' generates an external sync pulse at the test station jack labeled "SYNC" when the given statement number is executed in the test sequence. 'SYNC OFF' nullifies the action of 'SYNC ON'.

Restrictions:

The statement number cannot be larger than 177777B. Also, SYNC ON cannot be used concurrently with PAUSE ON.

Error Messages:

<u>Text</u>	<u>Description</u>
MISSING/DUPLICATE PARAMETER	The parameters ON or OFF both appeared, or else neither appeared.
INVALID STATEMENT NUMBER	The integer was greater than 177777B.

Examples:

/. SYNC ON 27B STAT4
/. SYNC OFF STAT4

USER'S MANUAL ADDENDUM

MANUAL: TOPSY
SYSTEM: SENTRY-400
SUBJECT: CUSTOM PRODUCT J-019
ASSIGNMENT STATEMENT ALTER
AUTHOR: D. G. HARBAUGH
DATE: 21 DECEMBER 1970

The purpose of this custom product is to allow modification of FACTOR test programs at execution time in order to avoid time-consuming editing and re-compiling which can only be done by halting testing. The initial implementation of this custom software was based on REVISION 3 of TOPSY, November 21, 1970, and modules affected are: GLOBAL, MPRO, IAARITH, and CPMAIN.

One, and only one, type of FACTOR statement can be altered, namely the assignment statement. An assignment statement is any statement of the form:

var = exp;

where var is a variable name or an array name followed by a subscript, and

where exp is any legal FACTOR arithmetic expression.

An arithmetic expression can be a single numeric constant, a single variable name, or a combination of variables and constants together with arithmetic operators.

Examples of assignment statements:

```
X = 1.5;  
VOLTS = VCCVLT;  
AXT [25] = VOLTS-X;  
XAMPT = ((AXT [4 ]-AXT [N] ) / (X + 77.6))*FFACT;
```

See the FACTOR user's manual for a complete description of assignment statements and expressions.

This custom software can be used to substitute any numeric value for the value of the expression at the time that the statement is executed. The following new TOPSY command was provided for the purpose of specifying which statement to alter and what the new value should be:

```
:/ .ALTER stmnt val STAT(N)
```

where stmnt is the number of the assignment statement to be altered (given in octal). This number is found in the left margin of the FACTOR compilation listing.

where val is the numeric value to be substituted for the expression.

It is necessary that the statement number and value be given in the order shown.

Once a statement is altered it remains altered until the command to CLEAR that station is given, or until another LOAD command (without SAVE) is given for that station. An alter can be used to override previous alters on the same statement. The combined total number of alters for all stations may not exceed sixty (60) at any time. If an attempt is made to enter the 61st alter, CPMAIN responds with the message "ALTER BUFFER IS FULL" and no other action is taken. If one of the numbers is missing from the command, CPMAIN responds with "DUPL/MISSING PARM. "

Attached is a sample FACTOR program followed by output from several executions of that program, showing various combinations of the use of alters.

```

000001 REM TEST PROGRAM FOR ALTER SOFTWARE.
000001 PROGRAM NAME IS ALTEST.
000001
000001 DCL A,B,C,ARY[10] /2,4,6,8,10,12,14,16,18,20/ ;
000002
000002 A = ARY[1]/2 ; REM VALUE IS 1 ;
000003 B = ARY[10]/4 ; REM VALUE IS 5 ;
000004 C = A+B+A ; REM VALUE IS 6 ;
000005 WRITE(TTP) A,B,C;
000006
000006 A = 100;
000007 B = 101;
000010 C = 102;
000011 WRITE(TTP) A,B,C;
000012
000012 WRITE(TTP) ARY;
000013 FOR N = 1 THRU 10 DO
000014 ARY[N] = N ;
000015 WRITE(TTP) ARY;
000016 END;

```

```

*// TOPSY
:/. LOAD 'ALTEST' STAT4
:

```

Note: This is the nominal program output.

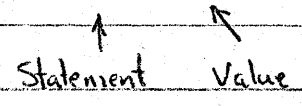
STAT4A

+ 1	+ 5	+ 6		
+100	+101	+102		
+ 2	+ 4	+ 6	+ 8	+ 10
+ 12	+ 14	+ 16	+ 18	+ 20
+ 1	+ 2	+ 3	+ 4	+ 5
+ 6	+ 7	+ 8	+ 9	+ 10

```

:/. ALTER STAT4 6 66.78
:

```



Note: Statement number 6 on compilation listing is circled. Statement numbers are octal.

STAT4A

+ 1	+ 5	+ 6		
+6.678E+01	+101	+102		
+ 2	+ 4	+ 6	+ 8	+ 10
+ 12	+ 14	+ 16	+ 18	+ 20
+ 1	+ 2	+ 3	+ 4	+ 5

(Last line of output suppressed by MONITOR button.)

/. ALTER STAT4 3 55.55

Note: All previous alters remain active until the CLEAR command.

STAT4A

+ 1	+5.555E+01	+5.655E+01		
+6.678E+01	+101	+102		
+ 2	+ 4	+ 6	+ 8	+ 10
+ 12	+ 14	+ 16	+ 18	+ 20

:/ . ALTER STAT4 14B 88

↑
Given in
Octal

STAT4A

+ 1	+5.555E+01	+5.655E+01		
+6.678E+01	+101	+102		
+ 2	+ 4	+ 6	+ 8	+ 10
+ 12	+ 14	+ 16	+ 18	+ 20
+ 88	+ 88	+ 88	+ 88	+ 88
+ 83	+ 88	+ 88	+ 88	+ 88

:/ . CLEAR STAT4

STAT4A

+ 1	+ 5	+ 6		
+100	+101	+102		
+ 2	+ 4	+ 6	+ 8	+ 10
+ 12	+ 14	+ 16	+ 18	+ 20
+ 1	+ 2	+ 3	+ 4	+ 5
+ 6	+ 7	+ 8	+ 9	+ 10

0/. ALTER STAT4 2 22

:

STAT4A

+ 22 + 5 +132

:/ . ALTER STAT4 2 44

:

Example of an alter command
overriding a previous alter on
the same statement.

STAT4A

0 44 + 5 +264
+100 +101 +102
+ 2 + 4 + 6 + 8 + 10

DATA SHEETS FOR REV 5 TOPSY

Updating REV 3 TOPSY has produced the following additions/deletions to the mentioned manuals:

1) Reference to SENTRY-400 TESTER OPERATING SYSTEM Manual

a) Add to 4.3:

NOTE: TOPSY command names may NOT appear in the title text.
The error message:
COMMAND?
will be generated if such a name is found.

b) Add to 4.4 restrictions:

The command PAUSE ON (STATEMENT) should not be used on a CALL or CALL-1 statement. In such cases, TOPSY may skip the next instruction after the CALL.

c) Add 4.12.

4.12 MAGT

General form

/. MAGT [INi/NOIN] [OUTo/NOOUT] [REW_r] where i = 0, 1, 2
o = 0, 1, 2
r = 0, 1, 2

'MAGT' assigns magnetic tape input and/or output unit(s) to be used when FACTOR mag tape commands are executed. The 'i' and 'o' units may not be equal. 'NOIN' and 'NOOUT' produce in effect NOOP when FACTOR mag tape statements are executed. REW allows the rewinding of one unit, unit r.

NOTE

On executing the 'LOAD' command without the SAVE option, MAGT is initialized to NOIN, NOOUT.

ERROR MESSAGE:

Text

WRONG I/O

Description

An illegal I/O combination was specified, i.e., same unit for read and write.

RECEIVED

MAY 31 1971

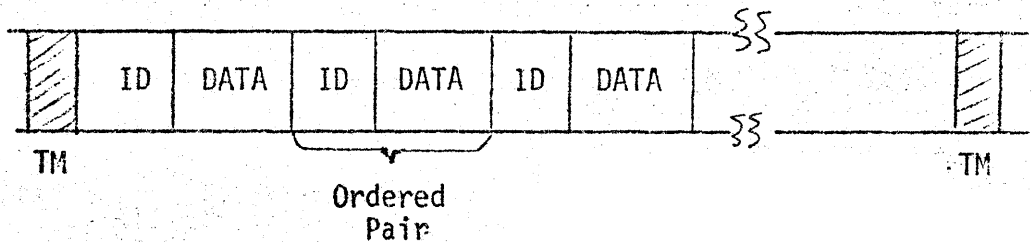
May 27, 1971

R. E. HUSTON

To: J. Spangberg
 From: J. Burnett
 Subject: EXPLANATION OF TOPSY MAG TAPE WRITE FORMAT.

cc: R. Forster
 R. Huston
 A. Perry
 A. Smith
 H. Vitale

TOPSY magnetic tape datalog data is, in general, written on tape in ordered pairs of records. The first of these records contains an identity (ID) and the second contains data associated with the identity. Example:

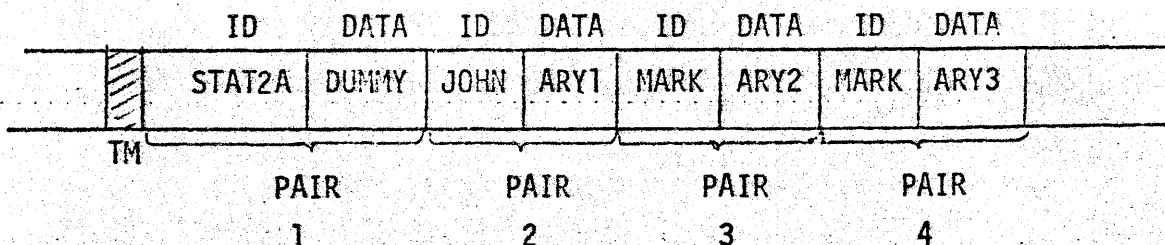


These records are grouped together into logical files delimited by tape marks (TM).

When a particular tape unit is on-line and assigned as a datalog output tape, (see MAGT command in TOPSY manual), each time START is pressed by the station requesting datalogging to tape, a new tape file is initiated and, in order to preserve the "pair" relationship, TOPSY writes an ID record identifying the station in operation, e.g., STAT4A, and a dummy data record which will contain title information if any has been specified. Then, for each execution of a write statement at the station, TOPSY will write an ID record containing both the "name" inserted in the write statement and the word count of the associated data record, and a data record containing one of the arrays as specified in the write statement.

Examples (on STAT 2):

```
WRITE (MTW) "JOHN" ARY1;
:
WRITE (MTW) "MARK" ARY2,ARY3;
```



The file is terminated by a TM when the station either reaches end-of-test or commits a non-recoverable error. If the same station or a new station then presses START and requests tape datalogging, a new file is started in the manner previously described.

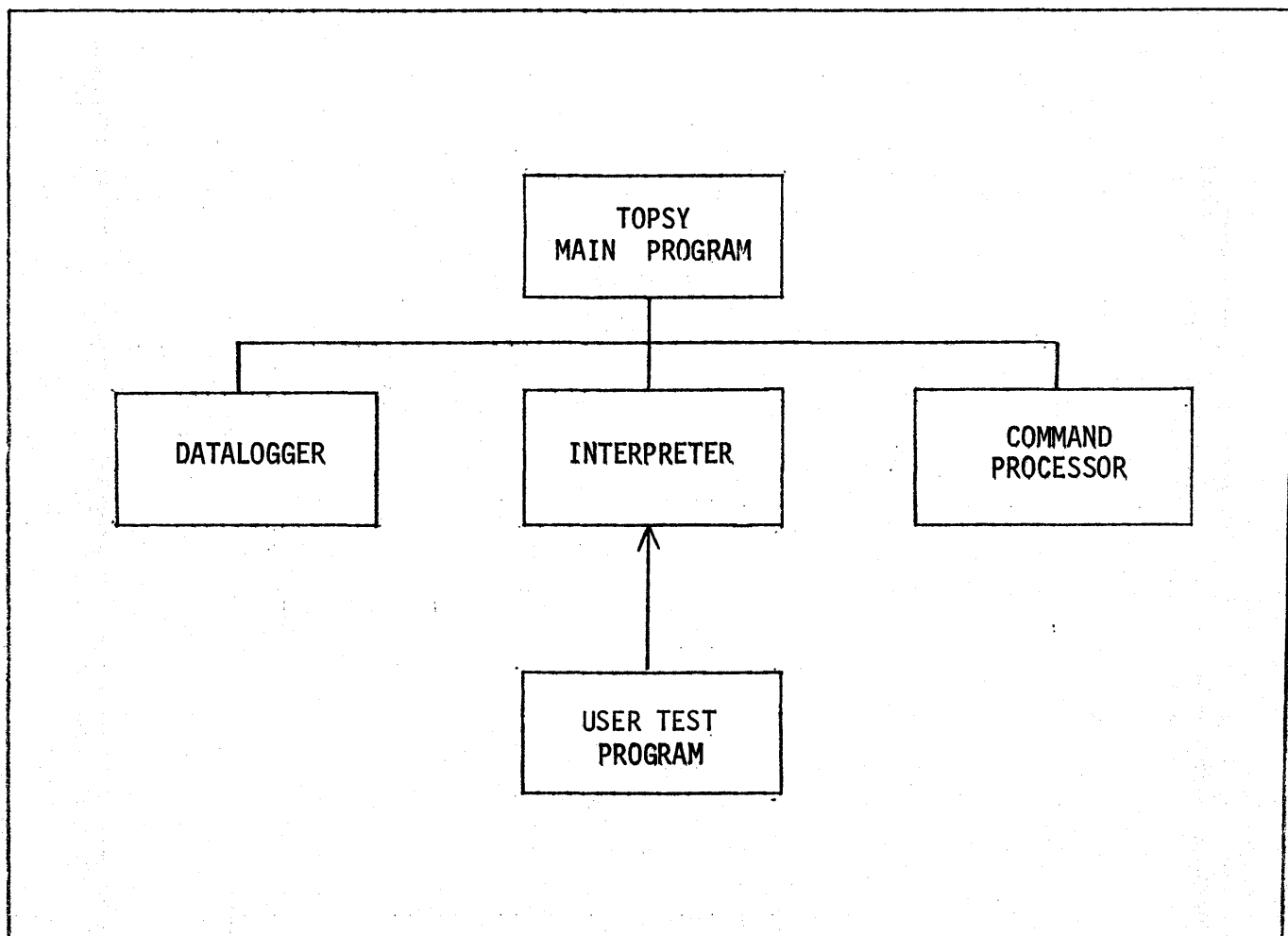
In terminating a tape file, TOPSY actually writes 2 tape marks and backs up over the last one so that

- (1) If datalogging continues, each file will be separated by one tape mark.
- (2) If datalogging is terminated, the end-of-tape will be indicated by the presence of both tape marks in sequence.

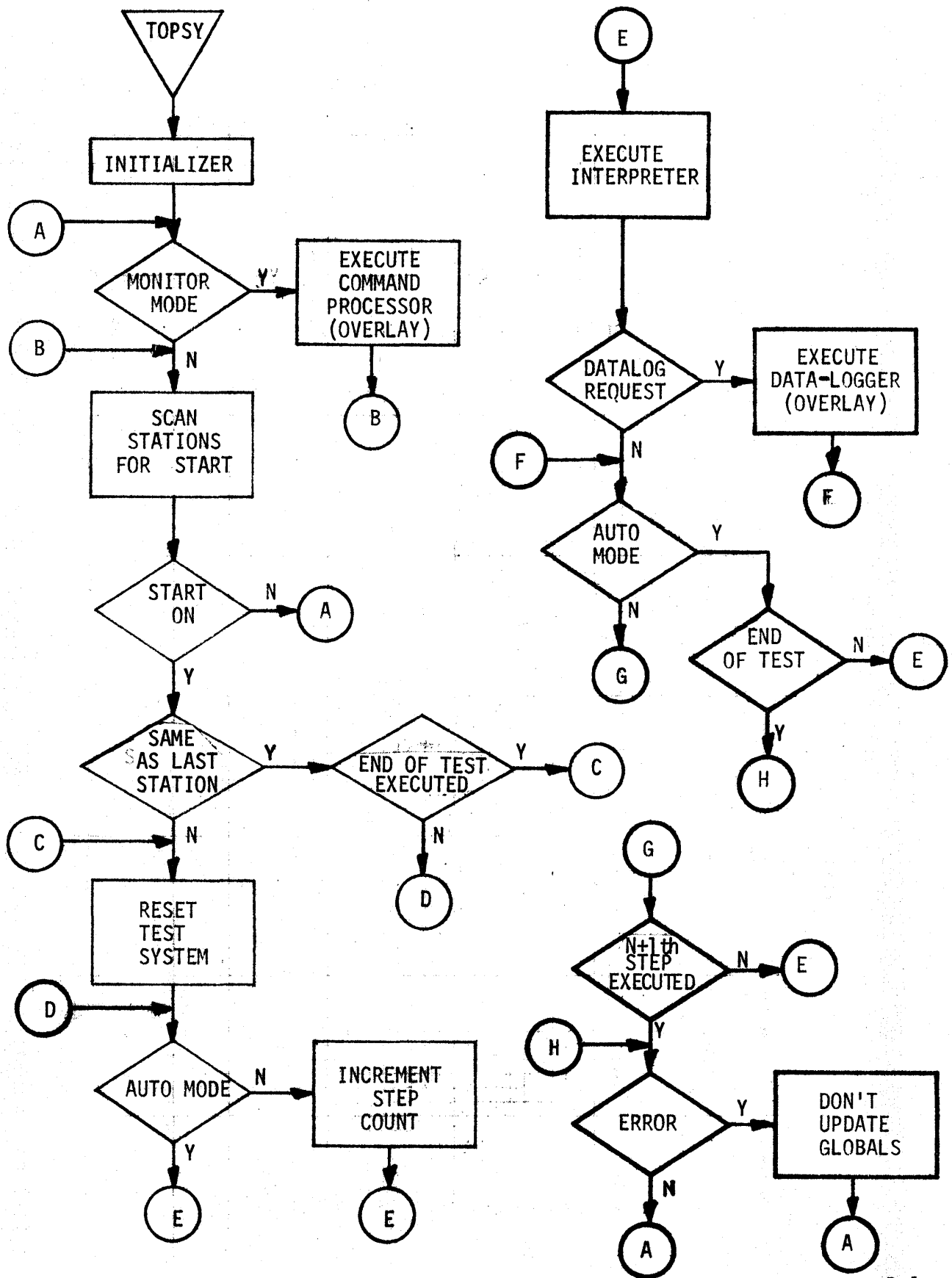
J. W. Burnett *jb*

JB/nr

APPENDIX A
DIAGRAM OF TOPSY ROUTINES



APPENDIX B
CONTROL FLOW CHART



APPENDIX C

TOPSY TERMINAL ERRORS

Terminal numbers have the following format:

INSTRUCTION NUMBER	TERMINAL ERROR NUMBER
n	nn

The instruction number equals n and the terminal error number equals nn, where n is an octal integer and nn is a decimal integer.

<u>ERROR NUMBER</u>	<u>MEANING</u>
1	Program not loaded
2	Station disabled
3	Magnitude or polarity error in pin number, clock count, or time delay
5	Magnitude error in voltage or current (exceeds hardware)
21	Current value not within set limits
22	Voltage value not within set limits
23	Improper pin address
24	Voltage value exceeds 30V
25	Wrong socket address
26	Undefined opcode
31*	Read (file skip forward executed)
33*	Write (file skip backward executed)
35*	EOT tape on write (catastrophic error)
36*	EOT tape on read
37*	Memory protect on tape read
40*	Data count error less than 7 or greater than assigned array
42*	Unrecoverable error
50	Improper vector declaration
51	The number of formal and actual parameters do not agree
52	Subscript violation
53	Empty stack
54	Program too big
55	EOF on test program
56	Illegal op code
57	Improper vector initialization
58	I/O error
59	Improper for loop constants

On terminal error 31, the tape is moved to the next tape file. On terminal error 33, the tape is moved back to the start of the last file. When start is pressed, the program will continue execution from these tape locations.

*Tape status issued is in octal. On terminal errors 35, 36, a tape rewind is executed. The program is aborted on terminal error 35.

APPENDIX D
ALLOCATION OF MAIN MEMORY DURING TOPSY CONTROL

PAGE TRAP		27777B
USER-PROGRAM READ-BUFFER		
RUN-TIME STACK		FIRST AVAILABLE 16500B
DATALOG SUB OVERLAY	COMMAND PROCESSOR	
DATALOG OVERLAY (OUTPUT)	OVERLAY (INPUT)	OVERLAY ORIGIN 143000B
TOPSY INTERPRETER AND SUBROUTINES		
		3700B
TOPSY MAIN PROGRAM		2200B
GLOBAL VARIABLES/CONSTANTS		550B
TESTER TRANSFER VECTOR		510B
AUTOMATIC RESTART ROUTINE COREIMAGE LOADER		77B
INTERRUPT ADDRESS		0