CR-54519
afd

[redacted]

FACILITY FORM 602

N66-18476	
(ACCESSION NUMBER)	(THRU)
131	1
(PAGES)	(CODE)
CR-54519	08
(NASA CR OR TMX OR AD NUMBER)	(CATEGORY)

GPO PRICE $ ______

CFSTI PRICE(S) $ ______

Hard copy (HC) 4.00

Microfiche (MF) 1.25

ff 653 July 65

GIIIIIIID

GENERAL DYNAMICS
~~ASTRONAUTICS~~
NOW COMBINED WITH CONVAIR

LICOS

AN INTERPRETIVE SIMULATION OF THE LIBRASCOPE COMPUTER MODEL -1 AND MODEL -3

GDA63-1141
20 December 1963

NAS3-3232

PREPARED BY Marilyn McLennan
M. McLennan
RESEARCH ENGINEER

CHECKED BY R.C. Foster
R. C. Foster
SENIOR RESEARCH ENGINEER

CHECKED BY John H. Terrell
J. H. Terrell
RESEARCH ENGINEER

APPROVED BY C H Gutzler
C. H. Gutzler
RESEARCH GROUP ENGINEER

GENERAL DYNAMICS | ASTRONAUTICS

FOREWORD

This report is intended to acquaint computer programmers and guidance engineers with computer program Number 2302 (LICOS) which is used to simulate the Librascope Model -1 and Model -3 vehicleborne computers. Information concerning these computers was drawn largely from References 3 and 4.

The LICOS program was created by the GD/A Scientific Programming Group, in coordination with the GD/A Centaur Guidance Analysis Group.

Some familiarity with the Librascope computer on the part of the reader is assumed in the writing of this report.

SUMMARY

18476

LICOS, programmed in 1960 to simulate the Librascope model ASN-24-1 computer and function generator, was modified extensively to permit simulation of both the model ASN-24-3 computer and the telemetry data converter. It has been modified subsequently as required.

This report gives a general description of the Librascope computer followed by a description of the simulation of the computer, function generator, and telemetry data converter. All information required for using the LICOS program is given in detail. In addition, all printed output and magnetic tape output obtainable from LICOS are described.

Because program modifications are frequently made as required, this report may become temporarily outdated until its content can be correspondingly modified by revision pages.

All computer functions are simulated by LICOS with the exactness required by the GD/A Guidance Analysis Group; many operations are represented bit-for-bit. The simulation performed by LICOS is sufficiently accurate to permit thorough testing of preflight and inflight programs before they are actually used in the Librascope computer; it is recommended that LICOS be used fully for such testing. author

TABLE OF CONTENTS

TABLE OF CONTENTS, Contd

LIST OF ILLUSTRATIONS

LIST OF ILLUSTRATIONS, Contd

SECTION 1

LIBRASCOPE ASN-24 COMPUTER (MODELS -1 AND -3)

1.1 DESCRIPTION. The ASN-24 computer is a drum type, general purpose, electronic digital computer which uses two's complement arithmetic and real time inputs to perform arithmetic and logic operations called orders. The drum turns at 6000 revolutions per minute. Sixteen hundred magnetic oxide strips are scribed on its cylindrical surface parallel to the axis of revolution. Electromagnets called writing heads, fixed to the computer case, impart magnetic polarity to small portions of each of these oxide strips as they rotate past; the circular path of magnetized particles (bits) formed around the drum surface by each head is known as a track. The drum is divided into sections containing specific tracks, and each track is subdivided into 64 sectors, each containing 25 (1600/64) bits, the polarity patterns of which comprise the orders or instruction words. These patterns are picked off by other electromagnetic coils known as read heads, and are delivered to the appropriate addresses in the computer to perform the ordered operations.

Data for the program are stored in the main memory section of the drum which contains permanent and temporary track sections, an accumulator register, a sigmator long line, and an MSCAN circuit which scans the memory signals to form instructions known as control words. The main memory capacity for the Model -1 computer is 2560 words. The Model -3 uses eight additional tracks bringing its capacity up to 3072 words.

Tracks in the permanent track section may be read but not written during the program. They are loaded with the guidance program from paper tape. The Model -1 has 37 permanent tracks: Numbers 0, 1 through 27, 31, 0*, 1* through 3*, 16*, and 17* through 19*. The Model -3 computer has 44 permanent tracks: Numbers 0, 1 through 26, 31, 0*, 1* through 5*, 16*, and 17* through 25*.

The temporary tracks may be both written and read by the program. They are used to store intermediate values during computation. The Model -1 computer has three temporary tracks, Numbers 28, 29, and 30. The Model -3 computer has four temporary tracks, Numbers 27, 28, 29, and 30.

The accumulator register (A_0) is a track which recirculates every word time; that is, the read head on this track picks up the 25-bit pattern and feeds it back to the write head so that the track at all times consists of 64 identical words. It is the only register addressable by the program.

The sigmator long line is a 32-word-recirculating track by which velocity, time, and position inputs are available to the program. The torquing and steering commands, main engine cutoff signal, and telemetry are output from this track.

MSCAN is a flip-flop circuit which samples various signals to form a 25-bit pattern known as the control word. The bit positions that have meaning for the program are:

P_0	Ignition Alert
P_2	Exit from Short Loop
P_5 through P_9	Mode Word Number

External commands may be fed to the computer during execution via the MSCAN.

1.2 FUNCTIONAL DESCRIPTION. The Librascope computer uses 25-bit instruction and data words with the most significant bit, P_0, to the right.

The instruction word is divided into three basic parts: the ORDER, the ALPHA ADDRESS, and the BETA ADDRESS (see Figure 1). The ORDER tells which operation is to be performed, the BETA ADDRESS tells where the operand is located, and the ALPHA ADDRESS tells where the next instruction to be executed is located. Each address consists of a track location and a sector location.

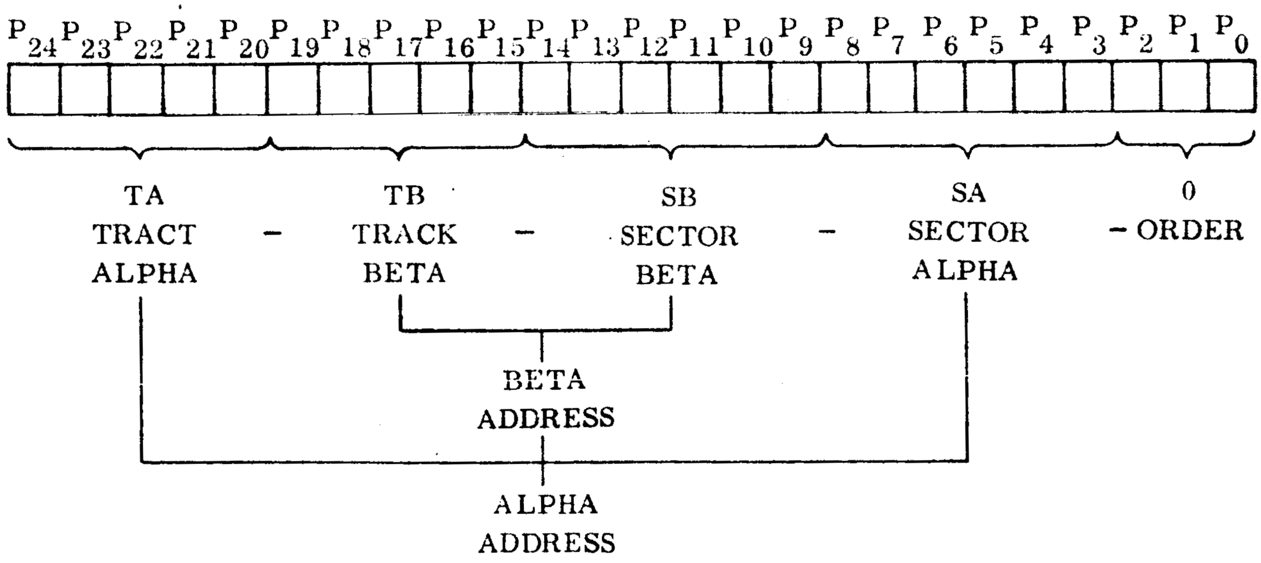


Figure 1. Instruction Word Format

Example: Instruction Format in Decimal and Binary

TA	TB	SB	SA	O	binary word
01001	11001	100111	001101	001	
18 -	19 -	57 -	44 -	4	decimal word

The data word consists of a sign bit (P_0) and 24 data bits (see Figure 2). All negative numbers are carried in the two's complement form. The words represent fixed-point numbers with the binary point between P_0 and P_1. This dictates that all numbers used in the computer must be scaled to absolute values less than one.

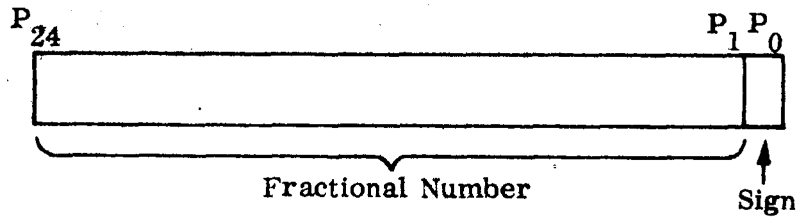


Figure 2. Data Word Format

Example: Binary-to-Decimal Conversion Positive

000000000000000000000101.0		binary word
2^{-1} = .50	.625	decimal value
2^{-3} = .125		
.625		

Example: Binary-to-Decimal Conversion Negative

111111111111111111111001.1		binary word
2^{-2} = .25	-.375	decimal value
2^{-3} = .125		
.375		

Instructions available to the programmer are of 3-bit order and 8-bit order types. The 3-bit order instructions use only bits P_0, P_1, and P_2 to designate the operation, while the 8-bit order instructions use P_0, P_1, P_2, and P_{15} through P_{19} (TB in

Figure 1). In the latter case, either the track location of the operand is implicit in the 8-bit order, or no operand is involved, as in a shifting operation.

There are seven commands in the 3-bit classification. They are:

Order	Mnemonic	Description
0	AT	C(M) to A_0
1	AD	C(M)+A_0 to A_0
2	MU	C(M)xA_0 to A_0
3	SU	C(M)-A_0 to A_0
4	EX	C(M)∩A_0 to A_0
5	DV	A_0/C(M) to A_0
6	TC	A_0 Positive, Transfer to BETA A_0 Negative, Transfer to ALPHA

wherein

C(M) is the contents of a sector from main memory
A_0 is the contents of the accumulator
∩ is a "logical and"

All 8-bit order commands have an order of 7 and are represented by a mnemonic of SR. The value of TRACK BETA determines the operation in these cases. The instructions are:

TRACK BETA	DESCRIPTION
0	A_0 to Sigmator Long Line
1	A_0 to Track 28
2	A_0 to Track 29 (Model -1) A_0 to Track 30 (Model -3)
3	A_0 to Tracks 28, 29 (Model -1) A_0 to Track 27 (Model -3)

TRACK BETA	DESCRIPTION
4	A_0 to Track 30 (Model -1) A_0 to Track 29 (Model -3)
5	A_0 to Tracks 28, 30 (Model -1) A_0 to Tracks 28, 29 (Model -3)
6	A_0 to Tracks 29, 30
7	A_0 to Tracks 28, 29, 30 (Model -1) A_0 to Tracks 27, 28, 30 (Model -3)
8	Not Defined
9	Sigmator (Y02) to A_0
10	Mn to A_0 (Note: Mn = MSCAN)
11	Sigmator (Y01) to A_0
12	$Mn \cap A_0$ to A_0, Right Shift
13	A_0 Right Shift
14	$Mn \cap A_0$ to A_0, Left Shift
15	A_0 Left Shift
16	Stop Computer
17	Power Turn Off
18	Telemetry Discrete Order Code Test Discrete Reset H Flip-Flop (on ground only)
19	Booster Cutoff Vernier Cutoff Mode Accept Discrete

TRACK BETA	DESCRIPTION
20	Reset S6 Flip-Flop Reset H Flip-Flop Reset Main Engine Cutoff
21	Set S6 Flip-Flop
22	Sustainer Cutoff Reorient Discrete Ground Mode Complete
23	A_0 to 9 Time Flip-Flops
24	Not Defined (Model -1) FRU Discrete (Model -3)
25	Set H Flip-Flop
26	Not Defined (Model -1) FRV Discrete (Model -3)
27	Not Defined (Model -1) FRW Discrete (Model -3)
28	Not Defined (Model -1) FRU BAR Discrete (Model -3)
29	Not Defined (Model -1) FRV BAR Discrete (Model -3)
30	Not Defined (Model -1) FRW BAR Discrete (Model -3)
31	Not Defined

1.2.1 Inputs to the Librascope Computer. The two types of input to the computer are 1) the programs controlling the functioning of the computer, and 2) the data inputs to these programs during their execution. In the first group are the guidance program and the S_{0T} or sector address track program, both of which are loaded into the computer permanent storage from paper tape. In the second group are sigmator time, velocity, and position values, and the autopilot ignition alert signal.

Using the velocity, position, and time values from the sigmator long line, the guidance program solves the guidance equations, and computes torquing and steering commands to correct any errors in the trajectory. Cutoff points are also predicted and computed values are telemetered.

The S_{0T} program, a special track loaded with the program and called the sector address track, dictates the operation of the sigmator and MSCAN, controlling the timing of the integration. countdown, and telemetry.

To derive the sigmator time value, an external clock sends 1300 pulses per second to the sigmator short line track; this procedure gives each time pulse a value of 1/1300 second, thus determining the scale factor of this basic input. Every half-revolution of the drum, the accumulated pulses on the short line are added to the time value on the sigmator long line.

To derive sigmator velocity values, an accelerometer sends pulses to the sigmator short line track, each pulse representing 0.1 ft/sec. Like the time values, these velocity pulses accumulate on the short line, and are added to the previously accumulated velocity on the long line every half-revolution of the drum. This new value of velocity is stored on the long line as current velocity.

The position values on the sigmator long line are updated at a rate of 162.5 integrations per second. The integration consists of adding the current long line velocity to the position value. This gives each position pulse a value of (0.1 ft/sec) (162.5 int/sec) or 1/625 feet.

The autopilot signals to the computer that ignition alert has been given by placing a bit in P_0 of the control word in the MSCAN.

1.2.2 Outputs from the Computer. Computer outputs consist of commands to the guidance system and telemetry signals.

Steering and torquing values are computed by the guidance program and stored on specified sectors of the sigmator. They are picked up automatically from the long line and sent to the torquing and steering modules to correct the trajectory of the vehicle. When the guidance program reaches a computed cutoff point, the computer sends a discrete to the autopilot which then turns off the engine.

The guidance program specifies the quantities to be telemetered and stores the appropriate values on a data link of the sigmator long line. The Model -1 computer telemeters one bit every 32 word times; the Model -3 telemeters one bit every eight word times. If no new data are stored on the data link between telemetry discretes, zeros are telemetered. The choice of values, the order of telemetry, and the number of intermediate zeros telemetered are exclusively functions of the guidance program.

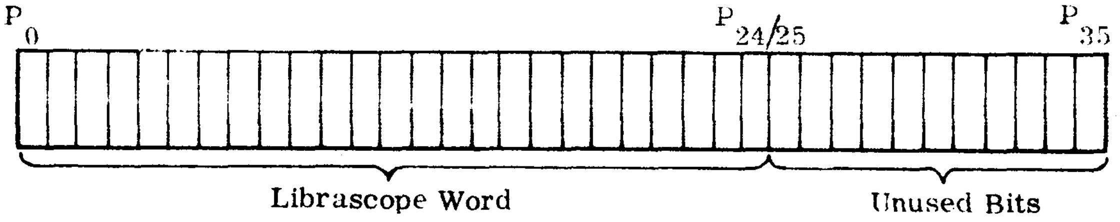
SECTION 2

GENERAL DESCRIPTION OF THE SIMULATION

The purpose of the LICOS program is to perform the functions of the Librascope guidance computer, and produce a printed output of the intermediate and final solutions of the guidance equations through the IBM 7090 computer. Figure 3 shows the various input sources and output capabilities of LICOS in a block diagram constructed around the 7090.

2.1 SIMULATION OF LIBRASCOPE COMPUTER INPUT. The two types of input to the simulation are the guidance program and the data to be analyzed.

2.1.1 Program Inputs. The guidance program is read into the 7090 core from binary cards produced by Decimal Assembler Program No. 2238, GD/A Scientific Computer Programming Library. Each card contains 22 words of data and the location, relative to 1750 (octal), of the first data word on the card. The words are ordered first by track numbers beginning with tracks 0 through 31, continuing through the starred tracks, and then by sector number from 0 through 63. Each 36-bit 7090 word contains the 25-bit Librascope word within it, left-adjusted with the most significant bit, P_0, to the left; thus:



The temporary track storage words and the unused 11 bits P_{25} through P_{35} of the permanent track storage words are set to zero during the loading process.

There is no provision for the input of an S_{0T} program, since the sigmator simulation is not bit-for-bit. The few functions of this program that must be taken into account are incorporated within the individual instruction simulations.

2.1.2 Data Inputs. Data input consists of velocity, position, and time input to the sigmator, and the ignition alert signal from the autopilot.

Sigmator velocity and time pulses are not actually accumulated on the short line, but the approximated values are stored on the long line to give the same results. The four methods available for obtaining velocity and position input to the long line are as follows:

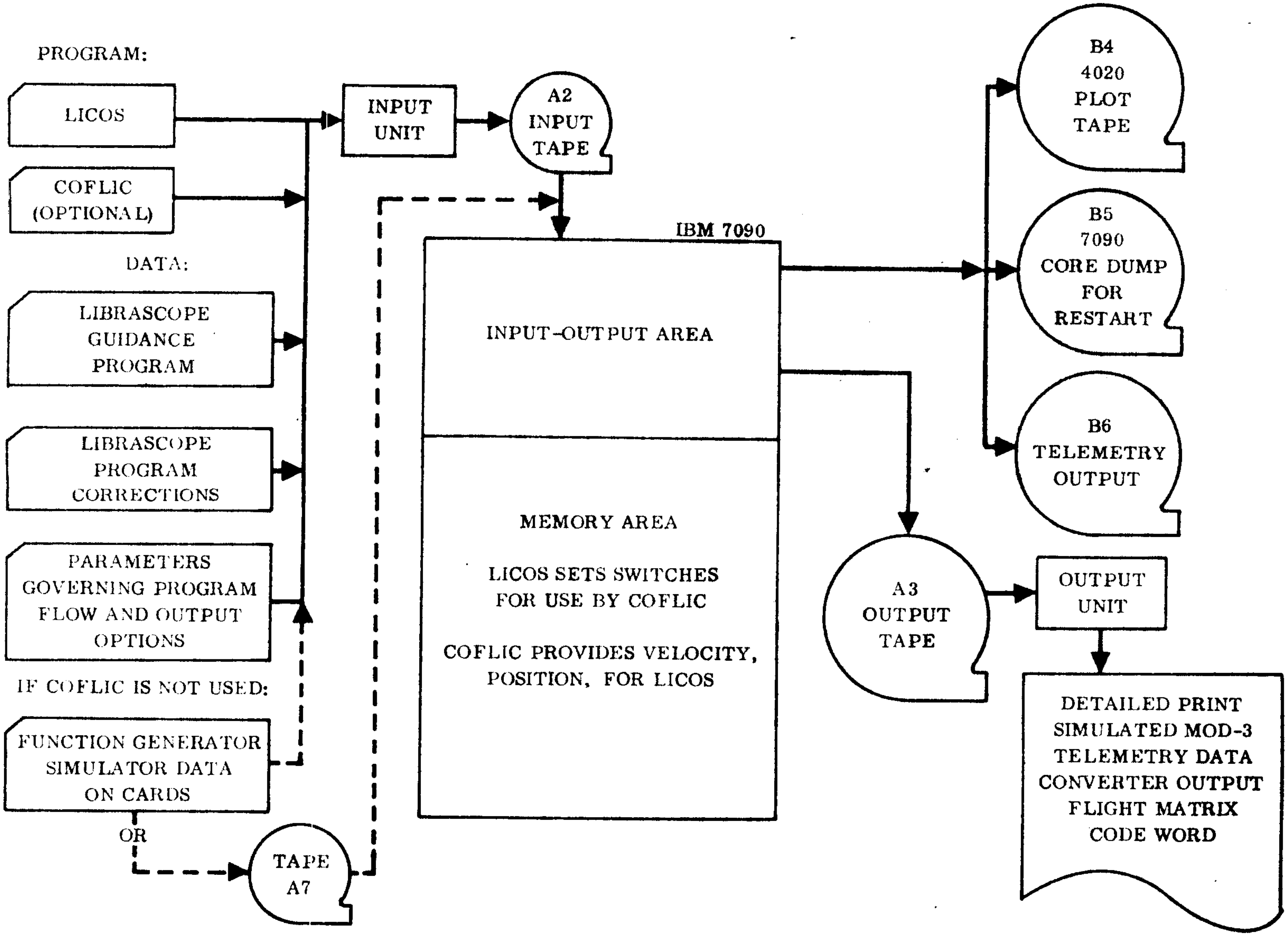


Figure 3. LICOS Simulation - Block Diagram

a. Telemetered Values. The simulation at present is programmed to accept the Thrust values of velocity and position. These values must be rescaled to sigmator scaling before they can be placed on the long line as the original sigmator values. Each set of these values will drive the guidance program through one cycle of computations.

b. Closed-Loop Simulation. The closed-loop operation of LICOS and COMBO is referred to as a separate program, called COFLIC (Reference 2). The COMBO routine computes the velocity and position of the vehicle at all times (Reference 1). LICOS merely scales these values properly and transfers them to the sigmator. A new set of values is required for each computation cycle of the guidance program.

c. Preflight Simulation. The preflight routines compute the velocity input for the sigmator. No position is used.

d. Open-Loop Simulation. Velocity and position are computed from an acceleration profile; i.e., a step function of acceleration vs. time which approximates the expected acceleration curve for the flight (see Figure 4). A subroutine of LICOS, known as FEXT, takes these acceleration values as input and computes the current value of velocity and position in floating point. LICOS then takes these values, converts them to Librascope numbers, and stores them on the sigmator. This process of updating the velocity and position values is performed each time a value is requested from the sigmator.

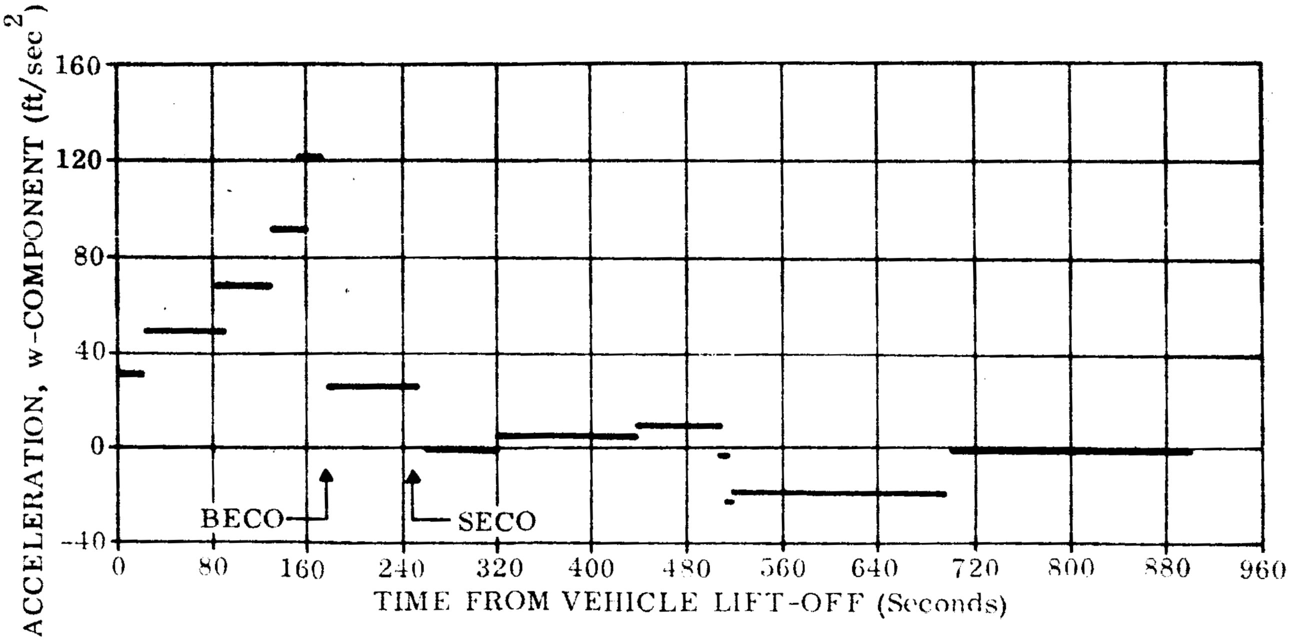


Figure 4. Acceleration Profile

Sigmator time input may be provided from a telemetry tape, or computed from a count of drum revolutions.

When the simulator is using telemetered velocities and positions as input, the telemetered time must also be used in all simulator references to time, to keep the values in phase.

In all simulations except those using telemetry inputs, time is computed by LICOS from a revolution counter. The drum is assumed to turn at a constant rate of 100 revolutions per second, and time pulses are accumulated at a rate of 1300 per second. Therefore, it is a simple matter to compute the number of time pulses that would have accumulated on the long line for a given number of revolutions. From the beginning of a simulation, a count is kept of the number of word times that have elapsed. This value is used in all computations of time.

When the autopilot ignition alert is given, a "one" bit is placed in P_0 of the MSCAN control word. The signal may be given in the simulation by COMBO or by an external input to LICOS.

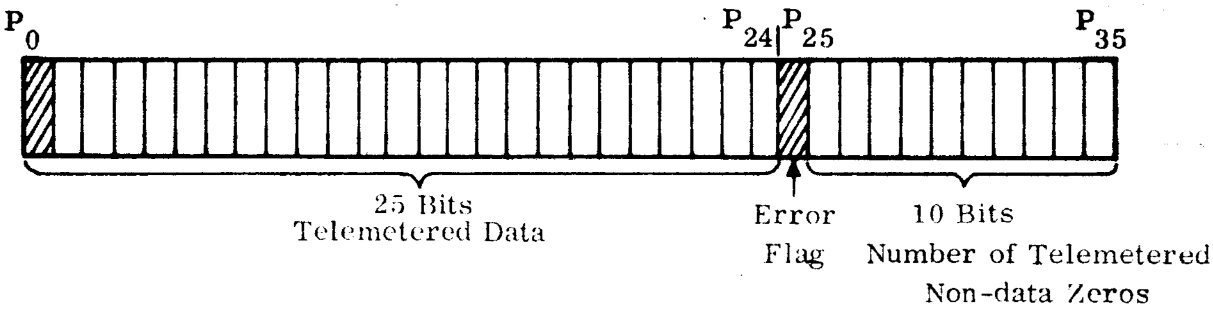
2.2 SIMULATION OF LIBRASCOPE COMPUTER OUTPUT

2.2.1 Steering and Torquing. In an open-loop simulation, steering and torquing values are not used. The output from LICOS is used only as a desired vehicle attitude vector, and not a direct steering output. This attitude vector is operated upon, externally to LICOS, to form commanded steering rates. In the preflight simulation the raw steering and torquing quantities are taken from the sigmator, rescaled to their actual values, and used as input to the preflight routines. In a closed-loop simulation, COMBO uses steering commands as input to correct the simulated flight of the vehicle. LICOS does produce torquing values, but they are not used in closed-loop simulation (see Reference 2).

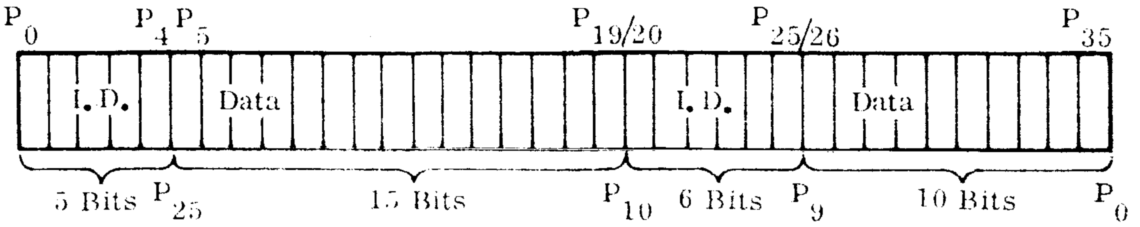
2.2.2 Signals to the Autopilot. Cutoff discretes are simulated by setting switches that are used by COMBO. These switches serve no purpose except in a closed-loop simulation. Comments are printed describing the nature of the discretes, the times at which they were issued, and the location of the instructions that commanded the signals.

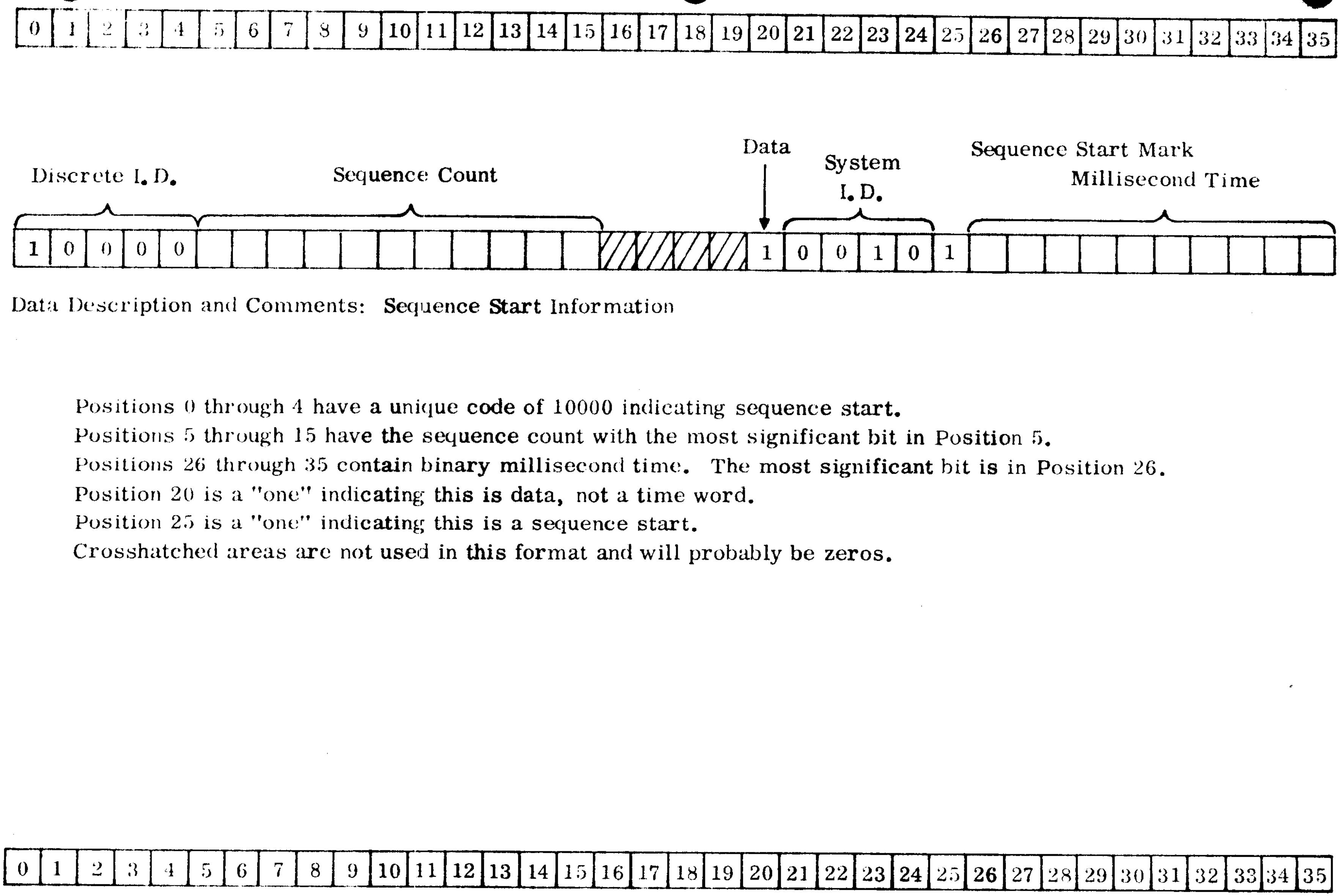
2.2.3 Telemetry. The guidance program stores the words to be telemetered on the data link sectors of the sigmator. In addition, it may give telemetry discretes as identification marks.

The Model -1 and Model -3 computers telemeter at different rates, and use quite different identification schemes. In the Model -1 the telemetry discrete is used to signify the beginning of each sequence. As the guidance program stores values on the data link, the simulator moves them to an output buffer, placing the data in bits 0 through 24. A count is kept of the number of non-data zeros that are telemetered after each word. This zero count is placed in bits 26 through 35 of the data word that follows. As each telemetry discrete is given, the output buffer is printed, or punched on octal cards. The telemetry discrete itself is unique in that it is the first word of the sequence and is a data word of zero with a non-data zero count in bits 26 through 35, thus:



In the Model -3 telemetry scheme the simulation processes the telemetry output a step further than does the actual computer, transforming it into the equivalent of the actual telemetry after it has been processed by the Model -3 telemetry data converter (see Figure 5). A six-word time discrete always signifies the beginning of a telemetry sequence (see Figure 5, Sheet 1). The need for counting non-data zeros is eliminated by giving an identification telemetry discrete at a specified amount of time before each word is telemetered. The length of the identification telemetry discretes varies. An eight-word time discrete means 25 bits of data will follow (Figure 5, Sheet 2). A ten-word time discrete means that 11 data bits will follow (Figure 5, Sheet 3). A twelve-word time discrete signifies that staging has occurred (Figure 5, Sheet 4). In the simulator data converter process the telemetry discrete is converted to an identification number, and the data is reversed in bit order and separated into parts; thus:





Data Description and Comments: **Sequence Start** Information

Positions 0 through 4 have a unique code of 10000 indicating sequence start.
Positions 5 through 15 have the sequence count with the most significant bit in Position 5.
Positions 26 through 35 contain binary millisecond time. The most significant bit is in Position 26.
Position 20 is a "one" indicating this is data, not a time word.
Position 25 is a "one" indicating this is a sequence start.
Crosshatched areas are not used in this format and will probably be zeros.

Figure 5. Centaur Digital Format, -3 Data Converter (Sheet 1 of 4)

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35
Discrete I.D.					Serial Data (Bits 25 - 11)																I.D.					Serial Data (Bits 10 - 1)									
0	X	X	X	X	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	1	1	0	1	1	0	9	8	7	6	5	4	3	2	1	0

Data Description and Comments: This is a normal 25 bit Data word.

Positions 0 through 4 contain the discrete code. The only discretes not permissable are 001100, 10000, and 11111.

Bit 20 will be a "one" indicating this is a data word, not time.

Bit 25 will be a "zero" indicating this is not a sequence start.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35

Figure 5. Centaur Digital Format, -3 Data Converter (Sheet 2 of 4)

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35

Discrete I.D.

I.D.

0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	10	1	1	0	1	1	0	9	8	7	6	5	4	3	2	1	0

Data Description and Comments: 11-bit word

ωw_{i-1} is the first 11-bit word, has a unique discrete I.D. of 001100.
ωv_{i-1} is the 2nd 11-bit word, has a discrete of 01000 (not unique).
ωu_{i-1} is the 3rd 11-bit word, has a discrete of 00010 (not unique).

There are three 11-bit words in each sequence of all phases, ωw_{i-1}, ωv_{i-1}, ωu_{i-1}.
ωw_{i-1} has a unique discrete identification code of 00110. This is the fifth word of the sequence. Words 6 and 7 are also 11-bits long.
Bit 20 will be a "one" indicating this is data not time.
Bit 25 will be a "zero" indicating this is not a sequence start.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35

Figure 5. Centaur Digital Format, -3 Data Converter (Sheet 3 of 4)

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35
Discrete I.D.																					I.D.														
1	1	1	1	1	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	1	0	0	1	0	0	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨

Data Description and Comments: Staging Discrete

This will appear after the Vt_{ui} word (word 9) in all phases.

Bits 0 through 4 will be all "ones", which is unique for this condition.

Crosshatched areas are not used in this format and will probably be zeros.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35

Figure 5. Centaur Digital Format, -3 Data Converter (Sheet 4 of 4)

A special word is generated for staging discretes or sequence start discretes. The sequence start word contains the sequence number in the data portion. All converted telemetry words are saved in an output buffer and written on a magnetic tape in groups of sequences.

2.3 LOGICAL FLOW OF THE SIMULATION. The simulator examines each instruction, separates it into its logical parts, and transfers control to the appropriate section of LICOS. Appendix A depicts the overall operation of the LICOS program in flow diagram form.

The instruction to be simulated is first checked to make sure that it does not direct control to an unused cell, or that it does not transfer control to itself, forming a one-cell loop. If one of these conditions exists, the simulator terminates the simulation and prints the comment **** LIBRASCOPE PROGRAM STOP ****, the location and octal value of the cell in error, and the location and Librascope value of the instruction that transferred control to the cell in error.

The instruction to be simulated is separated into its logical parts, ORDER, ALPHA ADDRESS, and BETA ADDRESS. The state of the S6 flip-flop is checked and modifications to the BETA and ALPHA addresses are made when applicable. Then control is transferred to the proper section of LICOS according to the value of the ORDER.

SECTION 3

INSTRUCTION SIMULATION

In all operations with Librascope words, the most significant 25 bits of the 36-bit word in the 7090 core contain the Librascope value. Only these 25 bits are retained in the Librascope accumulator cell, AC. The operand is located at the coded BETA address plus one.

For 3-bit order commands, control is sent directly to the simulation of the instruction. If the command is an 8-bit order command, further branching is made according to Track BETA.

3.1 THREE-BIT ORDER COMMANDS. Three-bit order commands are processed as follows:

a. Clear and Add. The operand is taken from its location in main memory and placed in the Librascope accumulator cell, AC.

b. Add. The operand from main memory is logically added to the contents of the Librascope accumulator, AC. A check is made for addition overflow. If the sum of like-sign numbers is not of the same sign, the comment, ADDITION OVERFLOW, will appear. One line of detailed print and the octal value of the accumulator before execution of the instruction in error are also printed. The location of the cell in error is saved. If another addition overflow occurs, the old error location is compared with the new. If the locations are the same, no error comment is printed, in order to avoid duplication of output.

c. Multiply. This operation is performed bit for bit according to the algorithm in the Librascopé manual. The number of bits to be multiplied is computed from the ALPHA and BETA values.

If the multiplier is a positive number, P_0 of the multiplier is tested. If P_0 contains a one, the multiplicand is added to the partial product. The multiplicand is right-shifted one bit and the multiplier is left-shifted one bit. Note that the sign bit is propagated on a right shift. These steps are repeated once for each bit of multiplication.

If the multiplier is negative, the two's complement of the multiplicand is added to the partial product and the process proceeds as in the case of a positive multiplier except that the test of P_0 and addition of multiplicand to partial product is omitted. Tests are made to determine if both the multiplicand and multiplier are equal to -1. If so, the comment, MULTIPLICATION WITH AC = -1, AND OPERAND = -1, is printed. As with the addition overflow, the previous value of the accumulator and one line of detailed print are also given and the check to avoid duplication of error printouts is also made.

d. Subtract. The two's complement of the accumulator value is added to the operand as in the Add operation. It should be noted that this subtraction is an abnormal subtraction, i.e., operand-minus-accumulator, rather than the conventional accumulator-minus-operand subtraction. A check is made for an accumulator value of -1. If this condition exists, the comment, SUBTRACTION WITH AC = -1, is printed along with the previous accumulator and detailed print values. The check to avoid duplication of error prints is also made.

e. Extract. A "logical and" is performed between the operand and the Librascope accumulator. The 25-bit result goes to the Librascope accumulator cell.

f. Divide. The divide operation is a bit-for-bit simulation. The dividend is originally in the accumulator, and the operand is the divisor. The length of the divide is determined from the ALPHA and BETA values. The division procedure is as follows:

1) If the sum of signs of divisor and remainder
   = 1, insert 0 in quotient,
   = 0, insert 1 in quotient.

2) Left-shift the remainder.

3) If quotient bit from 1)
   = 0, add divisor to remainder,
   = 1, subtract divisor from remainder.

The above steps are repeated the required number of times. The final step is to add $1 + q \times 2^{-(n-1)}$ to the quotient, where q is the last bit generated in the quotient and n is the number of bits of the division. A test is made to insure that the magnitude of the dividend is smaller than that of the divisor. If this condition is not met, the comment, OVERFLOW HAS OCCURRED IN DIVISION, is printed. If both the dividend and divisor are equal to -1, the comment, DIVISION WITH AC = -1 AND OPERAND = -1, is printed. The previous accumulator value and detailed print are also given, and the usual check is made to avoid duplication of error printouts.

g. Conditional Transfer. The sign of the Librascope accumulator is tested. If the sign is positive, the BETA address is used to compute the location of the next instruction to be simulated. If the sign is negative the ALPHA address is used.

3.2 EIGHT-BIT ORDER COMMANDS. In processing eight-bit order commands the value of Track BETA is tested and control sent to the simulation section for that instruction.

3.2.1 Accumulator to Track 27, 28, or 30. The BETA value is modified by the head spacing lag for the specified track and combined with the track number to give the main memory location of the desired sector. The value in the Librascope accumulator is then stored on that sector.

3.2.2 Accumulator to Track 29. The BETA value is modified by the head spacing lag for track 29. This value is combined with track number 29 to form the location of the cell in main memory. Since track 29 is an eight-word recirculating track, the value from the Librascope accumulator is stored on the computed sector and on every sector of track 29 that is a multiple of 8 of the computed sector.

Certain codes in the SOT program may stop the recirculation of track 29. This is usually an undesirable feature and such coding is avoided; however, there is no provision in the simulation for such an occurrence at present.

3.2.3 Accumulator to Sigmator. The five conditions under which values are stored on the sigmator long line are: 1) an initial value may be stored as velocity, position, or time; 2) a delta time may be stored on the countdown sector for main engine cutoff; 3) torquing or steering commands may be stored on their respective sectors for output; 4) data may be placed on the data link for telemetry; and 5) a value may be stored on an unused sector of the sigmator, just as on the temporary tracks. In all cases, the BETA value is used to compute the desired sector of the sigmator, modulo 32, since the sigmator is 32-word-recirculating.

3.2.3.1 Storing a Value on the Time Sectors. If an initial value is to be stored on the time sector of the sigmator, certain other values must be computed to insure the correct updating of the sigmator velocity and position values, and of the simulation input and outputs, since they are usually functions of the simulation time.

First, the current time value of the first-word phase of the present instruction is computed. Then the velocity and position values of each component are brought up to their correct values for that time. This value of time is then saved as the last time the velocities and positions were brought up to their current values. This time is also saved as start time.

If the new quantity is to be stored on the upper word of time, the value in the Librascope accumulator is stored on that sector directly. If the new quantity is to be stored on the lower word of time, the value in the Librascope accumulator is stored in that sector and the following additional Functions are performed. The double word of time is saved for computing relative times. The new time value is converted to a floating point number and saved as the simulation time. All quantities involving the number of drum revolutions that have passed are set to zero. The comment, XXXX.XX WAS STORED ON THE SIGMATOR AS TIME AT TIME = YYYY.YY, is printed, where XXXX.XX is the new value of time, and YYYY.YY is the time value before lower time was stored. The new value of time is saved as the last time the velocity, time, and position values were updated.

A test is made for Centaur Phase III initialization. If it is Phase III, the output variables that are functions of the simulation time are modified so that they will be in phase. It should be noted, however, that this logic assumes that the only times that new time values will be stored on the sigmator during a flight will be during Booster Initialization and Centaur Phase III.

3.2.3.2 Storing a New Value on the Velocity or Position Sectors. When a new value is to be stored on one of the velocity or position sectors of the sigmator, the value of time on the sigmator is updated to its value at first-word time of that instruction. The velocity and position values of that component are updated to that time. The value of time is saved as the last time the position and velocity of that component were updated.

If the new value is zero, a test is made in the inflight simulation to insure that no extraneous bits result during the initialization process. To test, one word of position is set to zero first; then within 24 word times the velocity word of that component should set to zero. Finally, within another 24 word times velocity, the other word of position should set to zero. If these conditions are not met, the comment THE SIGMATOR WAS NOT SET TO ZERO PROPERLY is printed along with the location of the instruction in error.

The value in the Librascope accumulator is then stored on the specified sector of the sigmator. The new sigmator value is converted to a floating point number for use by the simulation. The old values of velocity and position are saved as the previous values of velocity and position.

3.2.3.3 Storing a Delta Time on the Countdown Sector. If the delta time is zero, the contents of the accumulator are stored on the countdown sector, and no other simulation tests are made since this is only an initialization procedure.

If the delta time is a negative number, the contents of the accumulator are stored on the countdown sector, and if MECO has been reset, the Main Engine Cutoff discrete is given. The comment MAIN ENGINE CUTOFF, the location of the instruction, and the current value of time are printed.

When the delta time is positive, which is the usual case, the time is computed which is to elapse before MECO discrete is given. This value is added to the current value of time to obtain the predicted value of the cutoff time. The comment is printed, DELTA TIME STORED ON SIGMATOR FOR MAIN ENGINE COUNTDOWN AT TIME = XXXXX. XX, where XXXXX.XX is the current value of time. The value of the delta time is given in both floating point format and Librascope format. The value from the accumulator is stored on the countdown sector, modulo 8.

3.2.3.4 Storing a Value on the Data Link. Values stored on the data link are to be telemetered at a rate determined by the model of the computer. Error checks are made to insure that the values are completely telemetered and properly identified.

a. Model -1 Telemetry. This model telemeters only one bit each 32 word-times. Therefore, to insure that data are not stored on the data link before the previous value has been telemetered completely, a check is made of the word times that have passed since the previous value was stored on the data link. For each 32 word times extra allowed between the data words, a zero is telemetered. The number of preceding zeroes telemetered is stored in the output buffer word of the current data value in bits 27 through 35. The value to be telemetered is also stored in that word in the output buffer in bits 0 through 24.

The count of words in this telemetry buffer is incremented and examined. If the buffer contains 20 words, it is emptied; if not, the value from the accumulator is stored on the sigmator, modulo 32.

If not enough word times were allowed for complete telemetry of the previous word, the number of bits that were successfully telemetered is put in place of the leading zero count of the current value, and a bit is placed in P_{26} of the data word that was clobbered. The comment, DATA WORD NUMBER X WAS CLOBBERED AFTER THE NTH BIT AT LOCATION = TTSS, is printed, where X is count of values in the buffer at the time of the error, N is the last bit successfully telemetered, and TTSS is the location on the data link where the value was stored too soon.

If the word stored on the data link is the first one after a telemetry discrete, the preceding zero count is increased by one because of the 32-word time lag. There is no need to check for overwritten data in this case since the discrete was the last item telemetered.

b. Model -3 Telemetry. The word-time values are computed corresponding to the times when the quantity being stored on the data link will begin and complete its telemetry. The word-time value corresponding to the time when the previous quantity completed its telemetry is saved. The data are then stored in the output buffer in the format specified by the model -3 data converter.

Since the data stored on the data link are not actually telemetered until 32 word times after storing, and a telemetry discrete (which sends its telemetry signal immediately) is expected before the data telemetry begins, all timing error checking is performed after the telemetry discrete has been given. If two data words are stored for telemetry with no intervening identification discrete, the comment, VALUE STORED ON DATA LINK WITHOUT TSS AT TTSS, is printed, where TTSS is not the location of the instruction that stored data without a discrete, but rather the location of the instruction that stored the next word on the data link.

3.2.3.5 Storing Value on a Steering or Torquing Sector. No special checking is done when torquing or steering signals are stored on the sigmator for output. The value from the accumulator is placed in the specified sector of the sigmator.

3.2.4 Sigmator to Accumulator. Two read heads are available for reading the sigmator long line track. The only difference between the simulation of the read from each is in their timing, due to the different spacing of the two heads.

If the value to be read is not time, velocity, position, or the delta time for main engine cutoff, the contents of the specified sector are placed in the Librascope accumulator.

a. Reading Time from the Sigmator. Whenever it is to be read, the value of time on the sigmator is brought up to its current value. This updated value is placed in the accumulator.

b. Reading Velocity or Position from the Sigmator. The sigmator time is updated to the value of time when the quantity to be read (velocity or position, whichever is requested), was last accumulated on the short line. The velocity and position values of the requested component are updated to that time and this new value is stored on the sigmator as the current value. This is also the value that is placed in the Librascope accumulator.

c. Reading Delta Time from the Sigmator Countdown Sector. The time that has elapsed since the delta time was stored on the countdown sector is computed. To this is added the value of delta time. The result is the value that would appear on the countdown sector, just as if it had been constantly incremented. Then a test is made to see if a bit has overflowed into P_0, giving the MECO discrete. If P_0 contains a one and MECO has been reset, the MECO discrete is given. The comment, MAIN ENGINE CUTOFF is printed, along with the location of the instruction that tested the countdown sector and the time that the discrete was given.

It should be noted that the MECO signal may be given late or not at all in the simulation, while the actual computer would give the discrete exactly as predicted. For simulation purposes it was assumed that the guidance program would have to test the countdown sector to find out when the signal was given. Therefore, to economize on the simulation time, testing is done only when the countdown sector is read.

3.2.5 MSCAN to Accumulator. The S_{0T} program restricts the reading of the MSCAN to certain sector codes. The BETA value is tested to see if it is a legitimate code. If not, the comment, MSCAN WAS READ WITH INCORRECT SECTOR BETA AT TTSS, is printed, where TTSS is the location of the instruction in error. In this case, the value of the MSCAN is not placed in the accumulator.

If BETA is a legitimate code, the value of the MSCAN is placed in the accumulator and the P_2 bit of the MSCAN is set to zero.

It should be noted that once a bit is placed in the P_2 position of the MSCAN, the guidance program must detect it the next time the MSCAN is read.

3.2.6 Accumulator Shifts. The number of shifts is computed from the sector addresses. If the number of shifts is greater than or equal to 25, the appropriate comment, A LEFT SHIFT OF 25 OR MORE PLACES, or A RIGHT SHIFT OF 25 OR MORE PLACES, is printed. Zero is placed in the accumulator as the result, except when P_0 contains a one on a right-shift. In this case, since the sign bit is propagated, the accumulator contains all ones.

If the number of shifts is not over 25, the accumulator is shifted the specified number of bits, propagating the sign on a right-shift. On a left-shift the accumulator is tested for a change in sign. The comment, OVERFLOW ON LEFT SHIFT, is printed if a sign change occurs. On all comments, the accumulator value before the shift, in octal, is printed along with a line of detailed print.

3.2.7 Extract MSCAN and Shift. A "logical and" is performed between the accumulator and the MSCAN. The result is placed in the accumulator, and P_2 bit of the MSCAN is removed. The shifting simulation is done exactly as for the accumulator right- or left-shift instructions. The same error comments are printed, but the detailed print enables the programmer to differentiate between these and the accumulator shift instructions.

3.2.8 Stop Instruction. This instruction turns off the computer permanently, immediately terminating the simulation and printing the comment, STOP COMMAND ENCOUNTERED IN SIMULATOR.

3.2.9 Power Turn Off. This instruction is used to turn off the computer temporarily. Checks are made to insure that the instruction was not given accidentally and that proper preparation was made to turn the computer on again. The comment, POWER TURN OFF (PTO), is printed, along with the time and the location of the instruction that commanded the PTO. A check is made to see that time to turn the computer back on has been stored on the 9 flip-flops. If it has not, the comment, PTO COMMANDED WITH FREEFALL TIME ON 9 FLIP-FLOPS = 0 is printed. A test is made to be sure that the time accumulation is stopped; if not, the comment, PTO COMMANDED WITHOUT SETTING H GATE, is printed. If the simulation is closed-loop, the MSCAN is set to 0 for COMBO. A switch is set for LICOS to reinitialize and restart the simulation at location 3100. An OUT TO FACE exit must be made following the PTO command.

3.2.10 Telemetry or Order Code Test Discrete. The main purpose of this discrete is telemetry identification; in preflight, however, since there is no telemetry, this discrete is used to indicate that order code test was passed, and also to reset the H flip-flop.

3.2.10.1 Master Sequence Start Discrete. If this discrete is given for a period of 54 word-times or greater in an inflight simulation, it is treated as a Master Telemetry Sequence Start. The comment, MASTER SEQUENCF START, is printed, and the word-time counters are initialized for telemetry error detection.

3.2.10.2 Telemetry Discrete. This discrete is used by the Model -1 to indicate the beginning of a telemetry sequence, and by the Model -3 as identification for the telemetered data.

a. Model -1 Telemetry Discrete. A test is made to see if the discrete wipes out any telemetry data. If enough time was not allowed for 25 bits to be telemetered, the comment DATA WORD NUMBER X WAS CLOBBERED AFTER THE NTH BIT AT LOCATION - TTSS, is printed, where N is the last bit successfully telemetered, X is the number of words in the output buffer, and TTSS is the location of the discrete command in error. The word that was clobbered is tagged in bit P_{26}. If exactly 25 bits were telemetered, the previous data word is not tagged, but the comment is still printed since the zero which normally separates the values was not telemetered. The sequence counter is updated, and telemetry is output if desired. The comment, TELEMETRY SEQUENCE NUMBER XX, is printed, where XX is the sequence count; the comment is followed by the telemetered words dumped in octal. The same numbers may be punched on octal cards to be used as data for other programs. The number of leading zeroes is then stored in the new sequence start word and the buffer count is reset. The word-time value at which the pulse will go low is computed for the next leading zero computation.

b. Model -3 Telemetry Discretes. The discrete length must be either 6, 8, 10, or 12 word-times. If it is not, the comment, A TSS OF XX WORD TIMES IS NOT DEFINED TTSS, is printed, where XX is the length of the discrete and TTSS is the location of the instruction that gave the discrete. This discrete length is used as the identification code, right-adjusted in bits 0 through 4 of the converted telemetry word.

1. Six-Word-Time Discrete. This signifies the beginning of a telemetry sequence. If telemetry output is desired, the buffer is emptied, either in printed output or on magnetic tape, the buffer counter is reset, and the sequence counter updated. Then a new sequence start word is generated for the next sequence. A zero identification is stored in the first data word of the sequence.

2. Eight-Word-Time Discrete. An eight-word-time discrete signifies that 25 bits of data will follow, one zero after the discrete ends. The identification code is 8.

3. Ten-Word-Time Discrete. Eleven bits of data will follow, one zero after this discrete ends. The word-time that was predicted for telemetry completion of this data is modified for 11 bits. The identification code is 10.

4. Twelve-Word-Time Discrete. This is a staging discrete; no data are expected to follow. A special word is generated and placed in the output buffer. The word-time value for completion of this telemetry signal is computed for future error checking.

After the various functions have been performed according to the length of the identification discrete, the converted telemetry word is stored in the output buffer. Then tests are made to detect any errors in the timing of store operations for telemetry and transmission of discretes.

The Model -3 telemetry data converter may be set to accept either four-, eight-, or twelve-word-times between each identification discrete and its data bits. At present, it is set to accept only eight or twelve. LICOS will accept only eight or twelve, but with a simple coding change could be modified to accept four.

If the identification discrete is given too soon, so that the data bits of the previous data word are not completely telemetered, the comment, DATA WORD NUMBER X WAS CLOBBERED AFTER THE NTH BIT AT LOCATION TTSS, is printed, where X is the number of words in the output buffer, N is the last bit successfully telemetered, and TTSS is the location of the instruction that gave the discrete too soon.

If the discrete is terminated less than 8 words times from the beginning of the data telemetry, the comment, DISCRETE WAS GIVEN TOO LATE AT TTSS, is printed, where TSS is the location of the instruction that gave the discrete.

If the discrete is terminated more than 12 words times from the beginning of the data telemetry, the comment, DISCRETE WAS GIVEN TOO SOON AT TTSS, will be printed, where TTSS is the location of the instruction that gave the discrete.

3.2.10.3 Reset H Flip-Flop. This discrete resets the H flip-flop in a preflight simulation, allowing time to accumulate on the sigmator.

3.2.10.4 Order Code Test Discrete. In a preflight simulation this discrete is used to indicate that order code test was successfully passed. The discrete must be at least 54 word-times long to be recognized; if this condition is satisfied, the comment, ORDER CODE TEST PASSED, is printed.

3.2.11 Booster Cutoff, Vernier Cutoff, or Mode Accept Discrete. In preflight this discrete is a mode accept signal only. For inflight it can be either the Booster or Vernier Cutoff discrete.

a. Booster Cutoff Discrete. If the code word indicates booster phase, the comment, BOOSTER CUTOFF, is printed with the time and location.

b. Vernier Cutoff Discrete. If the code word indicates that the flight is beyond sustainer phase, the comment, VERNIER CUTOFF, is printed along with the time and location.

c. Mode Accept Discrete. The comment, MODE ACCEPT DISCRETE, is printed, along with the time and location of the instruction.

3.2.12 Reset S6, Reset MECO, Reset H Flip-Flop Discrete

a. Reset S6 Flip Flop. If the discrete is at least 54 word-times in length, the S6 flip-flop is reset. This means that no extra tracks may be referenced. This instruction must be on an extra or on an unstarrable track, but its ALPHA address must refer to a regular or unstarrable track. The comment, SELECT REGULAR TRACKS, the time, and the location will be printed.

b. Reset MECO. This discrete also resets MECO, so that the next time the countdown sector goes negative, MECO discrete will be given. If the discrete is not long enough to reset S6 (54 word-times), the comment, RESET MAIN ENGINE CUTOFF, the time, and the location are printed.

c. Reset H Flip-Flop. This discrete also resets the H flip-flop, allowing time to accumulate on the sigmator.

3.2.13 Set S6 Flip-Flop. This discrete must be given for at least 54 word times. If it is not, the comment, S6 WAS NOT SET BECAUSE DISCRETE WAS TOO SHORT, is printed. This discrete means that the starred tracks will be referenced, rather than the regular tracks. This instruction must be on a regular track or an unstarrable track, but its ALPHA address must refer to an extra or an unstarrable track. The comment, EXTRA TRACKS SELECTED, is then printed. Both comments also include the time and the location.

3.2.14 Sustainer Cutoff, or Reorient, or Ground Mode Complete Discrete. This discrete signifies Ground Mode Complete in preflight. For inflight it could be either Cutoff or Reorient.

a. Sustainer Cutoff Discrete. If the code word indicates that the flight is not yet into the Centaur stage, the comment, SUSTAINER CUTOFF, is printed along with the location and time.

b. Reorient Discrete. If the code word indicates that the stage of flight is in Centaur, the comment, REORIENT DISCRETE, is printed.

c. Ground Mode Complete Discrete. The comment, GROUND MODE COMPLETE DISCRETE, is printed, along with the time and the location. If push-to-set has been commanded (1 in bit P_2 of MSCAN), the switch to change modes is not set for the preflight routines.

3.2.15 Load 9 Time Flip-Flops. The comment, LIBRASCOPE VALUE LOADED ON FLIP FLOPS = XX XX XX XX X is printed, where XX XX XX XX X is the value from the Librascope accumulator in Librascope format. A test is made to insure that the H flip-flop has been set. If not, the comment TRIED TO SET 9 FLIP FLOPS WITH H GATE CLOSED , is printed. Otherwise the Librascope value is converted to a floating point number, added to simulation time, and stored in the 7090 core for use by COMBO and LICOS as the time that the computer will be turned back on. The time stored on the 9 flip-flops is identified as the time of free fall either for the transfer ellipse or for the transfer orbit, depending on the phase of the flight. If the flip-flops are loaded during the wrong phase of flight, the comment, ERROR IN OUTS (LIF SET WRONG MODE) MODE = N, is printed, where N indicates the phase of flight. If there is no error, the comment, FLIP FLOP TIME SENT TO COMBO = XXXXX.XX, is printed, where XXXXX.XX is the computed time in free-fall when the computer should come back on.

3.2.16 Set H Flip-Flops. This discrete stops the accumulation of time on the sigmator. The comment, SIGNAL TO SET H GATE AT TIME_____. TIME ACCUMULATION WILL STOP AT SIGMATOR TIME_____, the time, and the location are printed.

3.2.17 FR Discretes. These discretes are not simulated.

3.3 UNDEFINED DISCRETES. If an order of 7 is used with a Track Beta code that is not defined to be some operation, the comment, STORE INSTRUCTION WITH NON-EXISTENT TRACK CODE, the location of the instruction in error, and the value of that location in Librascope format, are printed. Then the simulation is terminated with a core dump.

3.4 NEXT LOCATION TO BE SIMULATED. The location of the next instruction to be simulated is computed from the ALPHA address. If the instruction is a first word command, the next instruction will be located at the ALPHA address plus one. If the instruction is an additional word command, the next instruction will be located at the ALPHA address plus two.

3.5 REVOLUTION COUNTER. This is the counter that is used to compute sigmator time. The revolution counter value at first word time is saved for printout. Each time sigmator time is initialized, so is the revolution counter. After the simulation of each instruction the word time value of the revolution counter is updated to the instruction read-in phase. In the case of a conditional transfer instruction, the revolution counter is updated during the simulation of that instruction.

3.6 FLOATING POINT TIME. This value is the number of real time seconds that have been simulated. It is used as a time standard by all input and output schemes based on time. To obtain floating time, first the value of the revolution counter is converted to seconds. Then to this value is added the value of time when the revolution counter was previously set to zero, the start time.

This value is not computed when using telemetry inputs. Instead the previous value of time read from the telemetry tape is used.

Each time an instruction is simulated, floating point time is checked against the preassigned time to terminate the simulation. If this time-to-stop has not been reached, the simulation continues.

3.7 PRINTED OUTPUT. The two principal types of printed output are instruction traces (detailed prints) of the guidance program and flight matrix printouts of the values computed from the guidance equations.

3.7.1 Instruction Trace. An instruction trace, also known as detailed print, is produced at preset simulation times, or on specific instructions of the guidance program. COMBO may also request a trace based on a specified number of compute cycles of the guidance program.

If detailed print is desired, one line is printed after the simulation of each instruction is completed. It consists of:

a. The location of the instruction that was just simulated, and its value in Librascope format.

b. A mnemonic of the order.

c. The value of the revolution counter at first word-time, separated into complete drum revolutions and word-times.

d. The value of the accumulator at the completion of the operation, in Librascope format and in scaled decimal.

e. The operand, if an operand was used, in Librascope format and scaled decimal.

Example:

```
1253  09  17  56  60  0  AT  000028  53   18  20  08  34  5
.-0.61666000   18  20  08  34  5    -0.61666000   (all on one line)
```

This line of detailed print shows that at location 1253 the contents of location 1757, which equaled -0.61666, were added into a cleared accumulator, giving the same result, -0.61666. The next instruction to be simulated will be at location 0961. The revolution counter at first word phase was 28 drum revolutions and 53 word times. AT is the mnemonic for a zero order, clear and add.

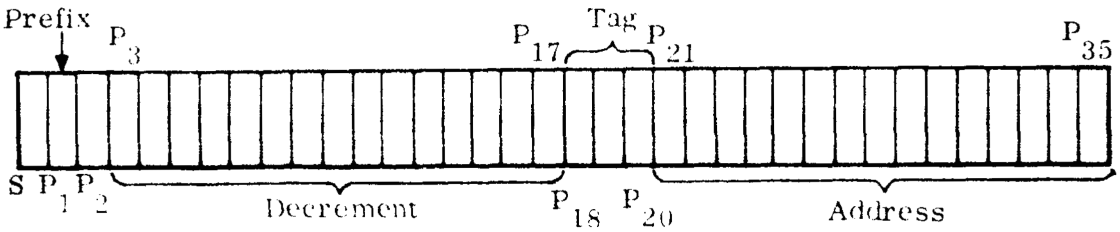
3.7.2 Flight Matrix. The flight matrix has 72 positions which may contain any values, intermediate or final, that are computed by the guidance program. Input to LICOS tells where to find the desired values on the drum, when to pick them up. how to rescale them, and in which positions of the flight matrix to place them. This flight matrix may be printed as often as it is required.

3.8 EXIT ADDRESSES. An exit address is a location in the guidance program where some special simulation function must be performed, usually a function concerned with the flight matrix.

A list of the locations that are exit addresses is read as input to LICOS. LICOS then stores a number in bits 30 through 35 of each of the designated locations on the drum, the numbers starting at 1 and increasing to a maximum of 50, according to the order in which the locations are read in. (Note that storage in bits 30 through 35 limits the exit address locations to permanent tracks, since bits 25 through 35 of the instructions that store on the temporary tracks are cleared during the simulation.)

As the simulation of each instruction is completed, bits 30 through 35 are examined to see if that instruction is an exit address. If it is not, the simulation of the next instruction is begun. If the instruction is an exit address, a corresponding "flow word" will be present which will determine the nature of the special function requested.

3.8.1 Flow Word. The IBM 7090 flow word consists of prefix, decrement, tag, and address.



If some function other than one concerned with the flight matrix is to be performed, the prefix is positive and the address contains a non-zero number, called a patch number. If the function is concerned with the flight matrix, the prefix of the flow word is minus to indicate that no patch number is present: the tag in this latter case describes the function to be performed on the flight matrix. The flow word decrement tells where to continue in the simulation after completing the tasks requested.

3.8.2 Patch Functions. If the flow word prefix is positive, a patch number is present. This patch number transfers control to an instruction in the patch transfer vector. The patch transfer vector in turn transfers control to the desired portion of the program. Although these patches are not usually concerned with the flight matrix values, the most frequently used patch instruction is to print the flight matrix at a specified time rather than on every compute cycle.

The functions currently performed by patches are described below.

PATCH NUMBER	FUNCTION
1	All of core is written on a magnetic tape to be saved for future use.
2	An octal core dump immediately terminates the simulation.
3	Instruction tracing begins immediately with the next location.
4	Instruction tracing initiated by patch 3 is terminated. (Instruction tracing initiated by COMBO, or on a programmed time basis is not affected.)
5	This patch must be used to print the flight matrix at specified intervals (skip-time printouts) over a specified time period. Using the input of desired printout frequencies, this routine computes the time value at which the matrix should be printed. This print time value is compared with floating point time each compute cycle. If floating point time has not yet reached print time value, control is transferred back to the exit address to continue with the matrix functions. If it is time to print, the print switch is set, the matrix is printed, and a new time-to-print is computed. At the end of the specified period control is sent back to the exit address, skipping the matrix operations, to take the exit specified by the flow word. If simulation time exceeds the time duration for which the skip-time printout is requested, printing frequency is set at one print each 40 seconds for the remainder of the run, and the comment, SKIP TIME PRINTOUTS ARE COMPLETE, is printed.
6	Time is stored on the sigmator as a two-word quantity. The upper word contains the sign and the 24 most significant bits; the lower word contains the 25 least significant bits. Patch Number 6 places all one-bits in the upper word, making time a relatively small negative value.
7	This patch sets the print switch. If the matrix is not to be filled or printed on a time basis, this patch may be used. Return is to the matrix operations of the exit address.
10	Patch 10 divides scale factor 66 by 2 for generalized equations.
11	Patch 11 changes FC scale factor to 52428.8, and G scale factor to 26214.4 for generalized equations.
12	This patch is used by the preflight simulation to change phases. The phase counter used in the preflight programs is set back to 1. A new PRINT, PCYCLE, ECORSE, L2 card is

PATCH NUMBER	FUNCTION
12 (contd)	read. The ECORSE values are converted to radians, and the counter for the print time table is set back to 1. Return is to the flow exit of the exit address.
13	Through this patch a data word of 25 zero bits may be telemetered.
14	This patch is meant to restore the capability to simulate the order code test section of the preflight programs. It is assumed that the OUT TO FACE † exit address in the order code test loop is the first exit address in the list. A flag is set for the preflight subroutines, and the flow of the first exit address is set with no patch, no matrix operations, and exit to FACE.
25	This patch changes scale factors for the flight matrix positions 66 and 71. These scaling changes were originally made for the F5 mission guidance program for the Model -3 computer, for sustainer phase. Position 66, which was Pi, is given a scale factor of 0.1537046 x 10^{-5}. Position 71, which was EPSILON, is given a scale factor of 6553.6.
26	This patch changes the scale factors of positions 36, 44, 52, 69, 70, and 71 of the flight matrix. These changes in the flight matrix setup were originally made for the Centaur phases of the F5 guidance program coded for the dash 3 computer. Positions 36, 44, and 52 are given a scale factor of 26214.4. This changes the F* scaling to V*G scaling for all three components. Position 71, which was EPSILON, is given a scale factor of 26214.4. Also, the drum location from which EPSILON is read is changed to 2901. The scale factor for position 69, (1K1), is changed to 0.03417969. The scale factor for position 70, which was K2 DT, is changed to 0.1604153 X 10^{-6}.
27	This patch is used in a preflight guidance program when it is not desired to set velocity back to zero at the beginning of each phase. A switch is set for the preflight subroutines.

If new patches are desired, they may be added by inserting the desired coding into LICOS, and assigning a patch number to that section by placing a transfer to the coding in the patch transfer vector. This may be done either by reassembling or with octal patches.

† FACE is an intermediary routine of COMBO which links the COMBO and LICOS simulations together to comprise the simulation COFLIC. See Reference 1.

Patch numbers not mentioned in this documentation are unused. It should be noted that these patch numbers are octal numbers. The maximum number of transfer allowed in the transfer vector is 41 (octal).

3.8.3 Print Switch. The print switch is set by Patch 7 or Patch 5 and is reset after the matrix is printed. If the flow word is negative, (i.e., contains no patch number) the print switch is tested. If it is reset, the matrix operations are shipped, and the exit specified by the flow is taken If the print switch is set, the matrix operations are performed. By this scheme the matrix is not filled unless it is to be printed.

3.8.4 Blanking the Flight Matrix. If the tag of the flow word contains either a 2 or a 4, the flight matrix will be blanked; that is, a minus zero will be stored in the 72 matrix cells.

3.8.5 Filling the Flight Matrix. If values are to be placed in the flight matrix for print out, it must be known which matrix positions are to be filled, where the values are located on the drum, and what their scale factors are. This information is provided by the "WHICH", by the output addresses in the Librascope memory, and by the appropriate stored scale factors, all in one-to-one correspondence with each other.

a. WHICH. The WHICH tells which matrix positions are to be filled. Each exit address has a corresponding WHICH, consisting of two words made up of 72 bits, each bit representing a position in the flight matrix and corresponding to an output address in the Librascope computer memory.

The WHICH is examined bit by bit. When a 1 bit is found, a value is taken from its correspondent specified word on the drum, scaled with its correspondent scale factor, and put in the corresponding position of the flight matrix. This process continues until all 1 bits have been found.

b. Output Addresses. Each 1 bit discovered in the WHICH designates a specific output address which is a location in the Librascope computer memory containing a value to be put into the flight matrix. This location may be a sector of either a temporary track, the sigmator, or the accumulator. There may be up to 72 output addresses, one for each position in the matrix.

c. Scale Factors. These are conversion values stored in the 7090, which transform Librascope values to the form usable by the 7090. Like the output addresses, there may be 72 scale factors, also in one-to-one correspondence with the matrix positions. The scaled decimal values from the output addresses are multiplied by these scale factors thus rescaling them to their original values.

3.8.6 Printing the Flight Matrix. If the tag of the flow word is a 4 or a 5, the flight matrix is printed. In addition, the values of the code word, revolution counter, and exit address are printed. The code word is printed in binary with P_0 to the left. In preflight simulation, additional preflight data are also printed. After printout the print switch is reset.

3.8.7 Exit to FACE. After the flight matrix is filled or printed, the exit specified by the decrement of the flow word is taken. The simulation continues by simulating the next instruction, or by exiting to FACE.

a. Closed-loop Simulations. In a closed-loop simulation (COFLIC), FACE is the routine that joins LICOS and COMBO. An exit to FACE must be made once at the end of each compute cycle of the guidance program.

b. Open-loop Simulations. In an open-loop inflight simulation, the exit to FACE is replaced by a transfer directly back to LICOS where special functions that need be performed only once per compute cycle are executed.

In an open-loop preflight simulation, the EXIT TO FACE is replaced by a transfer to the preflight subroutine FACE 1, on the first exit. On all subsequent FACE exits, the transfer is to the preflight subroutine FACE 2.

3.9 FUNCTIONS PERFORMED ON A FACE EXIT AND RETURN. Several functions must be performed before LICOS exits to FACE. After the necessary values have been computed for FACE, the simulation time of the FACE exit is saved. This time value is referred to as FTIME in LICOS.

Upon re-entering LICOS from FACE, the simulation functions that need be performed only once per compute cycle are executed. Then the simulation of the next instruction is begun.

3.9.1 Preparing to Exit to FACE. Values of delta time, flight mode, steering, and torquing must be computed for COMBO and preflight subroutine input.

a. Delta Time. A delta time is computed for use by both COMBO and preflight. It must be equal to the length of the compute cycle and is determined by the frequency of the FACE exits.

b. Flight Mode. The value of the flight mode determines the phase of the flight being simulated. It is set to 1 for Booster, 2 for Sustainer, 3 for Centaur Phase I, 4 for Centaur Phase II, and 5 for Centaur Phase III. The setting of the flight mode is determined by the value of the code word. The flight mode is used not only by COMBO, but by LICOS to determine the meaning of ambiguous discretes. (Refer to Paragraphs 3.2.11 and 3.2.14.)

c. Steering and Torquing. The steering values are used by COMBO and the torquing values by the preflight routines. These values are converted to floating point numbers and rescaled to their actual values.

3.9.2 Returning from FACE. LICOS is initialized on the first entry from FACE. (Refer to Paragraph 3.9.2.f.) Thereafter only simulation functions are performed, once per compute cycle.

a. Saving Core on Tape. By saving core on a magnetic tape, and at a later date loading the contents of this tape back into core, great savings are made in simulation time. This save-tape feature allows debugging of a guidance program at high simulation times without first simulating up to the trouble point.

It it is indicated to LICOS that a tape is to be saved at a specific simulation time, the save tape time will be compared with the simulation time on each FACE exit. When the save tape time is reached, the core is written on magnetic tape starting at location 1000 (octal). Sixty-three records of 1000 (octal) words are written. The last record is written containing locations 144 to 777 (octal), followed by an end-of-file mark. Redundancy checks and end-of-tape tests are performed. The comment, TAPE B5 WAS SAVED AT XXXX.XX SECONDS, is printed, where XXXX.XX is simulation time when the tape was saved.

If a tape is requested by PATCH 1, the comment, TAPE B5 WAS SAVED AT XXXX.XX SECONDS USING EXIT ADDRESS TTSS, is printed, where XXXX.XX is the value of the simulation time at which the tape is written, and TTSS is the location of the exit address. The return in this case is to the exit address analysis section. Otherwise the return is to test for telemetry tape inputs.

b. Order Code Exit in Preflight. If preflight is being simulated, the flow value of the first exit address (which would be the OUT exit in Order Code Test) is set not to exit OUT except when the Order Code Test is simulated. This flow value is reset to an OUT exit by PATCH 14.

c. Reading Telemetry Inputs for the Sigmator. In the Model -1, three velocities, three positions, and time, are read in one record from a magnetic tape. The u-, v-, and w-components of velocity and position are divided by D1+1.0, D2+1.0, and D3+1.0, respectively, to convert them from thrust values to sigmator values. These values are then converted from unscaled floating point numbers to scaled Librascope numbers and stored on the sigmator.

An end-of-file mark indicates either the end of the data or the start of a coast phase. The comment FOUND EOF READING INPUT TAPE is printed as each end-of-file is encountered and the simulation is terminated unless more files are expected. If more than one file of telemetry input is to be processed, an input is used to tell LICOS how many files to expect, and reading of this tape continues until the simulation finds a POWER TURN OFF instruction.

In the Model -3 the telemetry tape produced by LITEC (Program NO. 3004 in the GD/A Scientific Programming Library) contains the acutal Librascope values that were telemetered as the thrust values. The thrust values are read one set per compute cycle, shifted to give them sigmator scaling, and then stored directly on the sigmator.

If an end-of-file is encountered on the telemetry tape, the comment, READ AT PTO ALL FILES, is printed and the simulation is terminated.

d. Reinitialize After Power Turn Off. [In order to initialize LICOS properly after a POWER TURN OFF, the instruction that commanded the power turn off must have been an exit address with an OUT exit (refer to Paragraph 3.2.9). Then, as LICOS is re-entered, the switch set at the PTO simulation will be recognized and the simulation properly initialized.] The comment, ENTER SIMULATION AFTER POWER TURN OFF. TIME = XX.XXXX, is printed, where XX.XXXX is the simulated time of re-entry. The value on the flip-flop is then set to zero and the location of the next instruction to be simulated is set to 3100. The revolution counter is set back to zero and the simulation time value is saved as an initial time value called the start time. All other counters connected with drum revolutions are set back to zero.

If the simulation is being driven by a Model -1 telemetry tape, and an end-of-file mark has not yet been reached, the telemetry input tape is advanced to end-of-file. If no more files are to be read, the comment, READ AT PTO ALL FILES, is printed and the run terminated. If more files are expected, the comment, PTO READ THROUGH 1 FILE OK, is printed, and the simulation is continued.

e. Changing the MSCAN Value. The MSCAN content is changed periodically. The time-to-change and the corresponding matrix values are read as input. On the return from each FACE exit, simulation time is checked against the time-to-change the MSCAN. When that time is reached, the new value is stored on the MSCAN, and the comment, MSCAN SET TO NNNNNNNN AT TIME = XXXX.XX is printed, where NNNNNNNN is the new value of the MSCAN in octal (P_0 to the left), and XXXX.XX is the time that the new value was stored on the MSCAN.

The times for changing the MSCAN may for certain purposes be set relative to the time of Reorient if each relative set of times is separated by a zero time (see Paragraph 4.7.3). Therefore, it is necessary for the times to extend beyond the expected Reorient times or the length of simulation time, to avoid having the times taken as relative values prematurely.

f. Initializing LICOS. On the first entrance to LICOS, control is sent to the initialization section.

The Librascope guidance program is read into a section of core referred to as the DRUM. Currently, the DRUM is located beginning at cell 62000 (octal) and ending at 67777 (octal). If the binary cards containing the guidance program are not the first data after the data card, the first card encountered will be printed along with the comment, THIS CARD IS OUT OF ORDER IN THE DATA SETUP, and the simulation will be terminated.

If an unrecoverable redundancy is encountered, the comment REDUNDANCY CHECK WHILE READING DRUM, ADDRESS OF THE CARD IS NNNNN, is printed where NNNNN is the octal address of the card in error. The simulation is then terminated. The entire drum is checked for check sum errors. If any are encountered, the comment ERROR IN CHECK SUM, ADDRESS OF THE CARD IS NNNNN, is printed for each card in error.

If the guidance program has a title card, this title is printed first. The LICOS version title card is printed next. The simulation input is read according to the format specified by the FORIN input routine (see Reference 5). The input is sorted and converted to usable form. Various quantities are initialized. If extra tapes are being used, they begin transmission. The format of the flight matrix is printed. Then the constants from the guidance program are printed. If the simulation is closed-loop, the return is to COMBO. Otherwise, the return is to test for telemetry inputs. In either case, the comment, BEGIN LIBRASCOPE COMPUTER SIMULATION, is printed.

SECTION 4

LICOS INPUT AND OUTPUT

Various data are input to LICOS to specify logical switch setting within the simulation and to describe the types of input and output.

4.1 LOGICAL SWITCH SETTINGS. This input prescribes the type of simulation to be performed, the type of sigmator input to be used, code word locations, code word bits to be used to indicate a change of phase. the computer model to be simulated, duration of the simulation, and types of output desired.

4.2 INPUT AND OUTPUT OPTIONS. Telemetry may be printed and/or placed on cards or tape (depending on the model of the computer being simulated), or omitted. An instruction trace may be given. Overflow error comments may be printed. The flight matrix may be printed on a time basis if PATCH 5 is used. The value of the MSCAN may be changed at any time.

4.3 GUIDANCE PROGRAM CORRECTIONS. A special input is available for making corrections to the guidance program. Any main memory location may be changed.

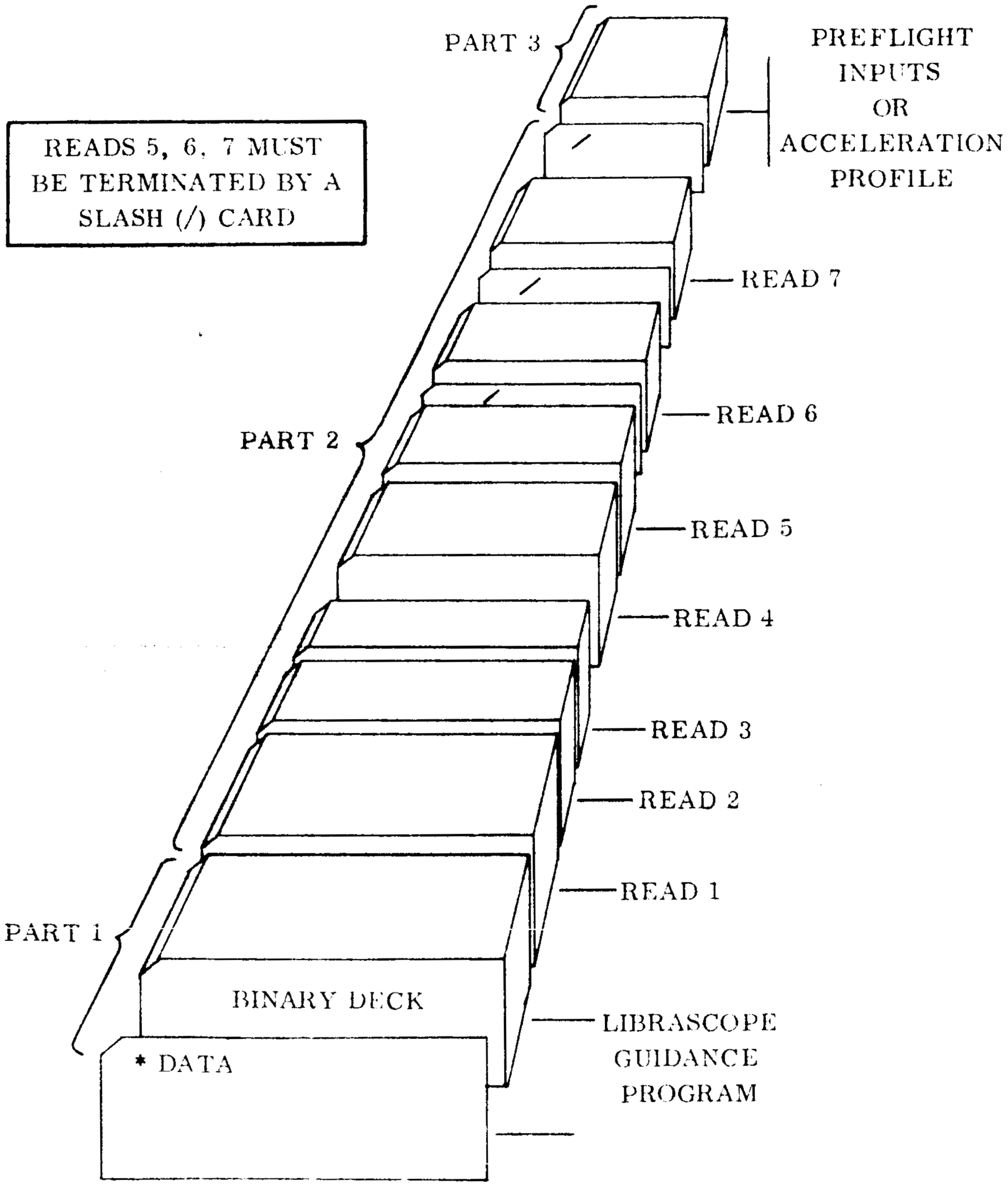
4.4 FLIGHT MATRIX SETUP. A list of exit addresses is read in along with the corresponding FLOW, WHICH, and PATCH values. In addition, the 72 matrix positions are each assigned a name, output address, and scale factor. These completely determine the contents of the matrix.

4.5 DISPLAY OF CONSTANTS. A list of names of constants. their locations on the drum, and their scale factors is read in. Then the names, locations, actual values, and scaled values in both decimal and Librascope form are printed.

4.6 DECK SETUP. The deck consist of three basic parts as shown in Figure 6. The first part is a binary deck of the Librascope guidance program. The second part is the simulation input. These two parts are always present in LICOS. In a preflight simulation, the third part is a section of special preflight program input. In an open-loop simulation, the third part is an acceleration profile. In a closed-loop or tape driven simulation the third part is omitted.

Part two, the simulation input. consists of seven basic sections called READS. Any number of cards may be used for each READ. The card format is specified by the FORIN routine (see Reference 5). Briefly this format is as follows:

a. Each parameter has an alphanumeric name of six or less characters.



READS 5, 6, 7 MUST BE TERMINATED BY A SLASH (/) CARD

Figure 6. LICOS Deck Setup

b. The format for each parameter is: the alphanumeric name of the parameter, an equals sign, and then the value of the parameter.

Example: NAME = 1

c. The form of the input value must be specified as octal, integer, floating point, or hollerith. The convention for representation of these types is as follows:

Octal - An octal number is preceded by a dollar sign ($) and assumed to be right adjusted.

Example: X = $16

Integer - This is a decimal number with no decimal point.

Example: X = 19

Floating Point - A floating point is specified when a decimal point or "E" appears somewhere in the number.

Example:
X = 17698.7688
X = 176987688E-4

Hollerith - If BCD alphanumeric input is desired, the letter H is used. The number immediately preceding the H determines how many characters of information will follow.

Example: X = 4HAB5D

d. Blanks are ignored, so data may appear in any columns not exceeding column 72. FORIN has been modified to print out all cards read unless the letter P appears in column 73.

e. Each parameter must be separated from the next by a card bearing a comma (,), an asterisk (*), or a slash (/). If a parameter is followed by a comma, the next parameter will be read in immediately. If the value is followed by an asterisk or a slash, some special function is to be performed. The READ functions performed by LICOS are described in the following paragraphs.

4.7 LICOS READ FUNCTIONS. In the following discussion:

a. X.X, Y.Y, or Z.Z, etc. are used to denote any decimal floating point number.

b. NNN, MMM, LLL, etc. are used to denote octal numbers.

c. AA, BBB, CCC, etc., are used to denote hollerith information.

d. I, JJ, K, etc., are used to denote integers.

e. TTSS* is some Librascope location, where TT is the track, SS is the sector, and * is used to indicate an extra track.

On all READ input, punctuation and spelling are of the utmost importance, but the order of the parameters within a READ does not matter. For further information on the format, see the writeup of the FORIN routine in Reference 5.

4.7.1 READ 1. Here the logical switch settings for LICOS are read in. Each parameter must be followed by a comma except the last one which must be followed by an asterisk or slash card.

a. TELSW = $0 -- provides printed telemetry output.

TELSW = $1 -- provides telemetry output, on magnetic tape for the Model-3 computer, or on octal cards for the Model-1 computer.

TELSW = $2 -- provides printed and either tape or card telemetry output.

b. SAVETP = $1 -- provides a magnetic tape of core at the simulation time specified by the parameter, B5TIME.

c. B5TIME = X.X -- used in conjunction with SAVETP = $1, this indicates the simulation time (seconds) at which the tape will be written.

d. TPOCD = $0 -- an acceleration profile, used for sigmator output.

TPOCD = $1 -- a telemetry input tape with no I.D. record will be used.

TPOCD = $2 -- calls for use of a telemetry tape with an I.D. record.

e. DTLPRT = 0 -- provides a detailed print given selectively according to the READ 4 input.

f. ERCOM = $1 -- allows printing of Librascope overflow error comments. (See INDEX OF LICOS COMMENTS).

g. ENDRUN = X.X -- prescribes the simulation time at which the run should be terminated. XX is in seconds. If no ENDRUN value is read in, the simulation will be terminated automatically at 5000.0 seconds.

h. **LIBRAS = 0 -- indicates that the Model -1 computer is to be simulated.**

LIBRAS = $1 -- indicates that the Model -3 computer is to be simulated.

i. PREFLT = 0 -- indicates that preflight is to be simulated.

j. HOEFS = 0 -- indicates that LICOS is running with COMBO.

k. FILES = J -- used in conjunction with TPOCD = $1 or TPOCD = $2 to indicate that more than 1 file of data is to be processed (J = the number of files to be processed).

l. LITEC = 1 -- indicates that LICOS is running with LITEC.

m. **CODEW = 4HTTSS -- indicates the LIBRASCOPE drum location of the code word. If omitted LICOS assumes the code word location to be 2818.**

n. CTABLE = $NNNNNN, $NNN....., where CTABLE may have up to 5 octal values. These octal values are left-adjusted bit configurations of the code word that indicate changes in the phase of flight. The first value indicates the end of the booster phase; the second the end of the sustainer phase; the third the end of Centaur Phase 1; the fourth the end of Centaur Phase 2; and the fifth the end of Centaur Phase 3. If these values are omitted, it is assumed that P3 indicates end of the booster phase, P6 indicates end of the sustainer phase, P9 indicates end of Centaur Phase 1. P12 indicates end of Centaur Phase 2.

o. ITIME = N -- indicates the number of times (N) that time on the sigmator is to be initialized. If this value is omitted, N is assumed equal to one.

p. PLOT = 1 -- causes LIPTON PLOT TAPE (B4) to be written.

PLOT = 0 -- prevents writing of LIPTON PLOT TAPE.

On * EXIT READ 3 begins

On / EXIT READ 6 begins

Example: TELSW = $1, DTLPRT = 0, LIBRAS = $1,
TPOCD = $0*

This will cause a telemetry tape to be written, a Librascope trace to be given at the time specified by READ 4, the Model -3 computer to be simulated, and a card acceleration profile to be read as sigmator input.

4.7.2 READ 2. If the matrix is to be printed on a time basis using PATCH 5, this read will specify values of DELTIM (the interval between printouts) and SKPTIM, the periods during which each DELTIM will prevail. Thus:

SKPTIM = W.W, Z.Z, V.V, ...
DELTIM = X.X, Y.Y, ...

which means that the matrix to be printed every X.X seconds from time W.W to time Z.Z, and every Y.Y seconds from time Z.Z to time V.V, etc.

Example: SKPTIM = 0., 10., 100.. DELTIM = .5, 15.*

> This means that the flight matrix will be printed every 0.5 seconds from zero seconds to 10 seconds. Then from 10 seconds to 100 seconds, it will be printed every 15 seconds.

If the simulation time should exceed the larger SKPTIM value, the printout interval is automatically set to 40 seconds for the remainder of the run. There may not be more than 10 SKPTIM values.

4.7.3 READ 3. The parameters of this READ are used to set the value of the MSCAN.

Setting MSCTIM = X.X, Y.Y, ... where X.X, Y.Y, etc., are simulation times at which the MSCAN should be changed

and MSET = $NN, $NNN, ... where NN etc., are the octal values that the MSCAN is to assume (The 25 bits of the MSCAN value are assumed to be left-adjusted.),

means that at time X.X the MSCAN will be set equal to NN; at time Y.Y. the MSCAN will be set equal to NNN, etc.

Example: MSCTIM = 2.0, 4000., MSET = $-0, 0*

> This means that at 2 seconds the MSCAN will have a 1 in P_0 (ignition alert). At 4000 seconds the MSCAN will be set to zero. The MSCTIM values must extend beyond the end of simulation time, and there is a limit of 10 MSCTIM values.

The MSCTIM values may be made relative to Reorient points for Centaur Phases 2 and 3.

Example: MSCTIM = 2.0, 100.0, 0., 4., 600., 0., 10., 4000.,
MSET = $-0, $0, $0, $-0, $0, $0, $-0, $0.*

In this example ignition alert signal would be given at 2 seconds after the beginning of the simulation, 4 seconds after Reorient for Phase 2 and 10 seconds after Reorient for Phase 3.

On * EXIT READ 4 begins

On / EXIT READ 6 begins

4.7.4 READ 4. These are the simulation time intervals for which detailed print is desired. If DTLPRT = 0, a trace (detailed print) of the Librascope program will be given during these intervals. The detailed print patches, PATCH 3 and PATCH 4, have no effect on these detailed print times. (Even if DTLPRT is not set 0, data for READ 4 must be present.)

Example: X.X, Y.Y, V.V, Z.Z, ... where X.X, Y.Y, etc., are the times to turn the print on and off. (Note: TRACE is the same as DTPSTR.)

This means that detailed print will be given from X.X to Y.Y, from V.V to Z.Z, etc.

Example: TRACE = 100.0, 102.0, 400.0, 404.0*

This means that detailed print will be turned on at 100 seconds, off at 102 seconds, on at 400 seconds, and off again at 404 seconds. The maximum allowable number of pairs of times is 10.

On * EXIT READ 5 begins.

On / EXIT READ 6 begins.

4.7.5 READ 5. The usual function of this READ is to make corrections to the Librascope guidance program.

a. LIBINS = 5HTTSS', 9HAAAAAAAAA* is the format for making a correction to an instruction. The location is TTSS', and the nine characters are the new instruction value. The sequence of the nine characters is ORDER, SECTOR ALPHA, SECTOR BETA, TRACK BETA, TRACK ALPHA

Example: LIBINS 2 5H1461., 9H704632103*

Location 1461 will be set to the value, 03-21-63-04-7.

b. LIBCON = 5HTTSS*, X.X, Y.Y is the format for correcting a constant. TTSS* is the location. X.X is the new value of the constant, and Y.Y is its scale factor.

Example: LIBCON = 5H0318*, 28.8, 56.*

The value 28.8 is divided by the scale factor, 56., giving the scaled constant value, .51428571. This value is stored in location 0318*.

c. If a tape is being used to drive LICOS (TPOCD = $1, or TPOCD = $2), the Model -1 computer is being simulated (LIBRAS = 0), and the drift constants in the guidance program, D1, D2, and D3, are not zero, then the values of D1, D2, and D3 must be read into LICOS to convert the tape input from thrust to sigmator values.

DRIFTS = X.X, Y.Y, Z.Z* where X.X, Y.Y, and Z.Z are the unscaled values of D1, D2, and D3 respectively.

It is assumed that the guidance program computes the telemetered thrust values from the sigmator values by adding the factor, D-value · sigmator value. Therefore, each thrust value must be divided by $D_i + 1.0$ to reconvert it to the sigmator value.

d. If a tape is being used to drive LICOS (TPOCD = $1, or TPOCD = $2), the Model -3 computer is being simulated (LIBRAS = $1), and the thrust values on the tape are to be read directly from the sigmator, the input parameter DRIFTS is set not equal to zero.

Example: DRIFTS = 1.*

On * EXIT, another set of LIBINS, LIBCON, or DRIFTS is operated on as described above.

On / EXIT, READ 5 terminates and READ 6 begins.

1.7.6 READ 6. Exit addresses and matrix setups are read here. If an exit address is read, the flow, patch number, if any, and WHICH values, if any, must also be present.

a. XAD = 5HTTSS* -- the location of the exit address. It cannot be a temporary track address.

b. FLO = 11HAAAAAAAAAAAA, 17HBBBBBBBBBBBBBBBBB, $CC -- the flow exit, matrix operation, and patch number respectively. The patch number is omitted if no patch function is to be executed. The flow exit must always be present.

AAAAAAAAAA is one of the two phrases, 1) CONTINUE, (the normal return), or 2) OUT TO FACE, (the special return).

In an inflight simulation an OUT exit must be made once per compute cycle to check the MSCAN setting, change the flight mode, save core on tape, read any tape input and restart after POWER TURN OFF. In addition, if the simulation is run with COMBO, an OUT exit must be made at each cutoff point, at POWER TURN OFF, and when initializing time. If the simulation is from tape input an OUT exit must be present in the initialization section of the guidance program, to read the first set of values.

In a preflight simulation an OUT exit must be made exactly once per compute cycle to convert the torquing values, to receive new velocity input from preflight, and to set up the exit from order code test (OCT) loop of the guidance program. Other OUT exits must be present, one in order code test, and another in integrated test. The simulation requires that the (OCT) OUT exit address be the first card in READ 6.

BBBB ....B is one of the four phrases, 1) NO BLANK NO PRINT, (does nothing), 2) NO BLANK PRINT, (prints the flight matrix), 3) BLANK NO PRINT, (blanks all flight matrix positions with a - 0.), or 4) BLANK PRINT, (blanks all positions of the flight matrix with a - 0., and prints the flight matrix.).

CC is the patch number. This may be any octal value from 1 to 41. Patch numbers used in READ 6 are listed with their functions below. Those numbers not listed are not used.

PATCH NUMBER	FUNCTION OF PATCH
1	Write core on tape for future use.
2	Terminate with core dump.
3	Turn detailed print on.
4	Turn detailed print off.
5	Print matrix on time basis.
6	Store a small negative number on lower time.
7	Fill the flight matrix for printing.

PATCH NUMBER	FUNCTION OF PATCH
10	Change scale factor 66 for generalized guidance equations.
11	Change scale factor 38, 46, 54, 61, and 62 for generalized guidance equations.
12	Change the phase in preflight.
13	Telemeter a word of 0 after telemetry discrete.
14	Set first exit address flow to OUT for O.C.T. (preflight).
25	Change to Sustainer Scaling for F5.
26	Change to Phase 1 Scaling for F5.
27	Do not initialize velocity in preflight.

(For further description of patches, see the documentation section on patches.)

c. WHICH I, J, K, ... specifies which positions in the matrix are to be filled. Each of the numbers (I, J, K...), which may range from 1 to 72, refer to a matrix position, output address, and scale factor. If no positions are to be filled, the WHICH is omitted.

Example: XAD = 4H1763, FLO = 8HCONTINUE, 12HBLANK PRINT, $7,
WHICH 5, 17, 63*

That is, at location 1763 PATCH 7 will set the print switch allowing the rest of the flow to be executed. The flight matrix will be blanked with - 0. values, and the positions 5, 17, and 63 of the flight matrix will be filled with new values.

The matrix is then printed. The simulation is continued by executing the next instruction.

d. If the matrix setup is to be read, a format number, output address, name, and scale factor must all be read.

FMT = JJ -- the format number. It specifies the position of the (9x8) flight matrix that is to be used. As an example: the element which is in row 2, column 6, (2, 6) is FMT = 14.

OUTAD = 5HTTSS* -- the output address where the desired quantity may be found on the drum. TT = 00 indicates the sigmator except when SS = 63; then the output address is the accumulator.

NAME = 6HAAAAAA -- the symbolic name of the variable that will appear in this position of the matrix. All of the names are printed once in their matrix form at the beginning of the run.

SCALE = X.X -- the scale factor of the variable. The value from the drum is multiplied by this value and placed in the matrix for print out.

Example: FMT = 51, NAME = 5HRMUO, OUTAD = 4H2843, SCALE = 677721.6*

This means that position (7, 3) of the flight matrix will contain RMUO, RMUO may be found in location 2843 and its scale factor is 677721.6.

On * EXIT, each value read is converted to usable form and stored in its proper table. Then another READ 6 value is read.

On / EXIT, READ 6 terminates and READ 7 begins.

4.7.7 READ 7

4.7.7.1 Guidance Constants. The constants in the guidance program may be printed out at the beginning of each run in their scaled, unscaled, and Librascope formats.

a. CONST = 6HAAAAAA -- the symbolic name of the desired constant.

b. LOC = 5HTTSS* -- the drum location of the constant.

c. SCALE = X.X -- the scale factor by which the value on the drum is multiplied to obtain the actual value.

Example: CONST = 3HTFF, LOC = 4H3146, SCALE = 12905.5*

d. If a P is in Column 73 of the card (as is usually the case), the input card image will not be printed.

* EXIT prints the values of the constant and reads another READ 7 value.

/ EXIT begins the simulation.

4.7.7.2 Acceleration Profile. Part three in an open-loop simulation deck is the acceleration profile. Three components of acceleration and a time up to which these accelerations should be used are read.

a. ENDTIM = X.X -- the time up to which the accelerations are to be used.

b. ACCEL = X.X, Y.Y, Z.Z -- the three components of acceleration.

Example: ENDTIM = 170., ACCEL = 0., 0.. 48.*

> This means that from the time the card is read until 170 seconds, an acceleration of 48 ft/sec^2 in the W direction is used to compute sigmator input.

When the ENDTIM value is reached, another profile card is read. If no more cards are present, the simulation is terminated.

Each time the sigmator lower time word is stored on, a profile card will be read.

* EXIT returns the simulation to the FEXT routine where the acceleration is integrated to give the velocity and position input necessary to drive LICOS.

/ EXIT would appear only in error, and would cause the simulation to be terminated immediately.

4.8 SPECIAL TAPES. (All special tapes used by LICOS are high density binary mode.)

4.8.1 Telemetry Input. The Model -1 telemetry input tape contains seven-word records, in floating point format. The values are time, thrust velocity in the u-, v-, and w-directions, and thrust position in the u-, v-, and w-directions, in that order.

The Model -3 telemetry input tape contains seven-word records, in Librascope format with P_0 through P_{24} containing the data. Time is the first word, RTU the second, RTV the third, RTW the fourth, VTU the fifth, VTV the sixth, and VTW the seventh.

The telemetry input is read from Tape A7. Control card, *LOAD TAPE HI A7, XXXX, is used, where XXXX is the number of the telemetry input tape.

4.8.2 Telemetry Output. The Model -1 telemetry output is written from the system's output tape A3 in the form of punched octal cards. No special setup is required. Each sequence is a set of cards with the address beginning at zero, five values per card. The number of words in each sequence does not exceed 20. No end-of-file is written.

The Model -3 telemetry output is written on tape B6. The number of words per record is equal to the number of words in each sequence, but may never exceed 30. The format of the words is specified by the Model -3 telemetry data converter. Control card, *SAVE TAPE (EOF) HI B6, is used so that an end-of-file is written by the monitor system.

4.8.3 Save Core on Tape. When core is to be saved for future use, control card *SAVE TAPE HI B5 is used and the core is written on tape B5. Sixty-three records of 512 words each are written, starting at location 1000 (octal). The final record contains 412 words written from location 144 (octal). An end-of-file mark is written.

SECTION 5

FEXT SUBROUTINE

FEXT is the LICOS subroutine that computes the velocity and position values from the acceleration profile in an open-loop simulation. (Refer to Paragraph 4.7.7.)

When this routine is entered, the time to which the quantity is updated must be in the accumulator, and Index Register 2 must contain a value that designates which component is to be updated.

If LICOS is being driven by a telemetry tape, no updating takes place; but the value in the LICOS-COMBO section of the 7090 core (common) is placed in the accumulator for the return to LICOS.

The current time from the accumulator is compared with the end time value of the acceleration profile step.

a. If the latter has not yet been reached, a delta time is computed by taking the difference between current time and the previous time that acceleration values were read. The velocity and position values of the component in question are updated and stored in common. The specific value requested, either velocity or position, is placed in the accumulator as the updated value for the return to LICOS.

b. If the current time is greater than or equal to ENDTIM (Paragraph 4.7.7.2.a), all components are brought up to the ENDTIM values in the following manner.

   For each component a delta time is computed by taking the difference between the ENDTIM and the previous time that the component values were updated. The values of velocity and position for that component are computed. The ENDTIM is saved as the previous time the components were updated. The current velocity and position values are saved as the previous values of velocity and position for that component. When this has been completed for each component, another delta time is computed, this one for the component for which updating is requested in Index Register 2. This delta time is taken from the difference of the current time and the ENDTIM value. Then a new profile card is read with a new end time value and new accelerations. The requested component is updated with the new acceleration to the current time.

The new values of velocity and position are stored in common, and the requested value is placed in the accumulator for the return to LICOS. Upon returning to LICOS, the value in the accumulator is converted to a LIBRASCOPE number and stored on the sigmator.

Velocity and position are updated by the formulas:

$$V = LASTV + (DELTAT) \times (A)$$

$$R = LASTR + (DELTAT \times LASTV) + (DELTAT)^2 \times (A)/2$$

SECTION 6

RESTARTING LICOS FROM TAPE

This is the routine that reads the save tape from a LICOS run into core, and transfers control to the section of LICOS that reads in any new input or corrections and continues the simulation.

The first 63 records are read into core filling locations 1000 to 7777 (octal). Then a "transfer channel in operation" instruction is set up in location 77700 to wait until the last record is read in. The last record is then read into locations 144 to 777 (octal), contro' is sent to the restart section of LICOS, and the simulation is begun.

The card deck for the restart routine must be set up in the following manner. The data card must be followed by three READS and, if the simulation is of an open-loop run, by the remainder of the acceleration profile cards. The required READS are described below.

READ 1

ADDRES = $LLLLL, VALUE = $MMM* where LLLLL is some octal location in core, and MMMMM is an octal value to be patched into core at location LLLLL. This is the only method available for correcting LICOS when restarting from tape.

Example: ADDRES = $70714, VALUE = $1*

*EXIT stores the values in core. Therefore, each set of ADDRES and VALUE must be followed by an asterisk (*) card.

/EXIT terminates READ 1 and expects READ 2 next. A slash must be present for READ 1 even though no corrections are to be made.

READ 2

SAVETP = $1 writes tape for later use.

B5TIME = XX.XXX is the simulation time at which the tape is to be written.

LIBCON = 4HTTSS, XX.XX, YY.Y* makes changes to constants as specified in the LICOS description of LIBCON.

LIBINS = 4HTTSS, 9HABBCCDDEE* makes instruction changes to the drum.

*EXIT converts the LIBINS or LIBCON read and then returns to READ 2. Therefore the B5TIME and SAVETP values must be terminated by commas (,) and LIBINS and LIBCON by an asterisk (*).

/EXIT prints the format of the flight matrix, and then directs the simulation to READ7 of the normal input setup, the initial constant printout.

Even though no constant printout is desired, a slash (/) card must be present. The remainder of the acceleration profile or other input completes the data deck.

SECTION 7

LICOS SYMBOLOGY

7.1 SWITCHES

SYMBOLIC NAME	SETTING	FUNCTION
11BSWH	Not 0	An 11-bit data word was telemetered
	0	A 25-bit data word was telemetered
ASN	Not 0	Simulate Model-3 computer
	0	Simulate Model-1 computer
CONUS		(No longer used)
CNT3BB	0	Compute time for detailed print at the beginning of the section
DAN	Not 0	Check sum error was encountered in the Librascope binary drum input deck
DOIT	Not 0	Print the flight matrix
DATAFG	Not 0	A data word was stored on the data link
	0	A telemetry discrete was given
DTLPRT	Not 0	Do not examine detailed print time list for printing a trace
	0	Give detailed print at the specified times
ERCOM	Not 0	Error comments will be printed as overflows occur
	0	No Librascope overflow error comments will be printed
ERRPNT	Not 0	One line of detailed print will be given as supplement to an error comment
FILR	Not 0	End-of-file was encountered reading the post flight telemetry tape without power turn off

7.1 SWITCHES, Contd

SYMBOLIC NAME	SETTING	FUNCTION
FBOCO	Negative	Booster cutoff discrete for COMBO
FCIMC	Negative	Centaur Phase 1 MECO discrete for COMBO
FCIVC	Negative	Centaur Phase 1 Vernier cutoff discrete for COMBO
FC2MC	Negative	Centaur Phase 2 MECO discrete for COMBO
FC2VC	Negative	Centaur Phase 2 VCO discrete for COMBO
EC3MC	Negative	Centaur Phase 3 MECO discrete for COMBO
FC3VC	Negative	Centaur Phase 3 VCO discrete for COMBO
FLMOD	1	Booster Phase
	2	Sustainer Phase
	3	Centaur Phase 1
	4	Centaur Phase 2
	5	Centaur Phase 3
FSUCO	Negative	Sustainer cutoff discrete for COMBO
FPREFL	Not 0	Simulate inflight
	0	Simulate preflight
FPTOFL	Not 0	Power turn off was given
FXPRIN	Not 0	Detailed print will be given
GET2	0	When converting a BCD Librascope address to binary the drum address in core will not be added
	Not 0	When converting a Librascope address the drum address will be added
GIGI1	0	Skip telemetry output
	Not 0	Give telemetry output when requested
HGATE	Not 0	Sigmator time accumulation is stopped

7.1 SWITCHES, Contd

SYMBOLIC NAME	SETTING	FUNCTION
HOEFS	0	Closed loop simulation using COMBO inputs
L2	Not 0	Read a new ECORSE value (preflight)
LITEC	Not 0	LICOS is running with LITEC
LIBEXT	Not 0	Converting a regular location
	0	Converting a starred location
LRSW	Not 0	One of the words of position of one component was set to 0.
MDCNG	Not 0	Change mode (preflight)
OCTF	Not 0	Simulate O. C. T. (preflight)
	0	Skip O. C. T. simulation
PT0C	Negative	Power turn off was executed
PORTW	Not 0	Velocity of one component was set to 0.
PRSWH	0	Skip the matrix functions of all exit addresses
RTTA3	Not 0	Redundancy light for channel A was turned on by (STH). LICOS will turn off the light, perform operations on A7 unbuffered, and turn the light back on for (STH) buffered output on A3
SAVETP	Not 0	Save a tape of core at B5TIME
	0	Do not save a tape of core on a time basis
SIFLAG	Not 0	Negative number is to be converted
	0	Positive number is to be converted
SWITCH	Not 0	Skips updating time and the revolution counter from ALPHA to BETA when time is initialized. This is to make all time values relative to first word phase when time was initialized.

7.1 SWITCHES, Contd

SYMBOLIC NAME	SETTING	FUNCTION
SINGL	Not 0	A single length word is to be converted
	0	A double length word is to be converted
STOPS	Not 0	Stop the simulation
S6	Not 0	S6 is high. Select the extra tracks
	0	S6 is low. Select the regular tracks
TORT	Not 0	Format statement has not been fulfilled yet for (STH)
	0	Format statement has been completed; print it
TELSW	0	Printed telemetry output only
	1	Either punched or tape output only
	2	Printed and either punched or tape output
TPOCD	0	Acceleration profile card input
	1	Telemetry tape input with no I. D.
	2	Telemetry tape input with I. D.
THEMIS	Not 0	No operand is in the detailed printout
	0	Operand is in the detailed printout
ZCDFL	Not 0	MECO is not reset
	0	MECO is reset

7.2 STORAGE BLOCKS

SYMBOLIC NAME	CONTENTS
3TIME	The value of simulation time when sigmator time is set to zero at Reorient for Centaur Phase 3
3CODEW	Location of the code word on the drum
A	The location of the instruction being simulated is in the address portion of this word
AC	Librascope accumulator register

7.2 STORAGE BLOCKS, Contd

SYMBOLIC NAME	CONTENTS
ATO	Acceleration during previous cycle (preflight)
ATP	Acceleration in platform coordinates (preflight)
ADD4	Format number as read in
AVOIR	Location of last MSCAN reading error
ADD6	BCD name for matrix format as read in
ADDRES	Absolute core location for octal correction made when restarting from a tape
ADD7	Scale factor for matrix printout as read in
AMSCAN	Same as MSCAN
BB1	Input buffer for Librascope drum deck
BABS	Revolution counter at first word phase
B5TIME	The simulation time when a tape containing all of core is to be saved
BEGREV	Revolution count when data will begin telemetry
CUL	Canonical conversion factor for position
CUV	Canonical conversion factor for velocity
CODEW	Code word location in BCD
CNT3BC	Time that COMBO section detailed print is turned off
D	Divisor in the division simulation, multiplicand in the multiplication simulation
DLOC	Location of constant BCD
DRAC	BCD name for initial constant printout

7.2 STORAGE BLOCKS, Contd

SYMBOLIC NAME	CONTENTS
DADD1	Exit address as read in
DCELL	Location of constant in binary
DADD2	Flow value as read in
DRIFT	Gyro drift values, D1 + 1.0, D2 + 1.0, D3 + 1.0 in floating decimal
DADD5	Output address as read in
DELTIM	Matrix printout frequency values in floating decimal to be stored in SKPSTR table
DUMP2	Location from which DUMP was called
EC	Platform misalignments in degrees (preflight)
ENDT	End time value from acceleration profile cards for FEXT
ECELL	Value of constant for initial constant printout
ERCNT	Telemetry buffer count for error print
ECORSE	Platform misalignments in radius (preflight)
ENDREV	Same as TREF
FRZU	Floating point position from FEXT
FVZU	Floating point velocity from FEXT
FBOCT	Time of Booster cutoff for COMBO
FCIMT	Time of Centaur Phase 1 MECO for COMBO
FCIVT	Time of Centaur Phase 1 Vernier cutoff for COMBO
FC2MT	Time of Centaur Phase 2 MECO for COMBO
FC2VT	Time of Centaur Phase 2 VCO for COMBO

7.2 STORAGE BLOCKS, Contd

SYMBOLIC NAME	CONTENTS
FC3MT	Time of Centaur Phase 3 MECO for COMBO
FC3VT	Time of Centaur Phase 3 VCO for COMBO
FCELL	Scaled value of constant for initial constant printout
FDELT	The delta time in floating decimal used by COMBO or preflight. Should be equal to length of compute cycle. It is computed before each exit to FACE by differencing current time and the time of the previous exit to FACE.
FDETP	Number of compute cycles for which detailed print is desired in each COMBO section
FMEGU	Floating point steering for preflight or COMBO
FMGDU	Platform torquing commands in floating point for COMBO or preflight
FSUCT	Time of sustainer cutoff for COMBO
FTFFP	Floating point time stored on the 9 flip-flops for the parking orbit free-fall phase for COMBO
FTFFT	Floating point time stored on the 9 flip-flops for the transfer elipse free-fall phase for COMBO
FEXTDT	Delta time for COFLIC FEXT
FTIME	The value of simulation time on the previous exit to FACE
INKTIM	Total time elapsed
INSTR	Five cells, each containing a left adjusted alphanumeric value for printout. The values are the five parts of a Librascope instruction; Track ALPHA, Track BETA, Sector BETA, Sector ALPHA, Order
ITIME	Number of times sigmator time is initialized
LI	Counter for PRTIME table (preflight)

7.2 STORAGE BLOCKS, Contd

SYMBOLIC NAME	CONTENTS
LOC	Location of next instruction to be simulated
LASTR	Previous position in floating decimal for FEXT
LASTT	Previous time in floating decimal for FEXT
LASTV	Previous velocity in floating decimal for FEXT
LDRUM	Location of drum in core
LIBUF	Buffer for seven input values from telemetry input tape; time, three components of position, three components of velocity
LSIGM	Location of sigmator in core
LASTAC	Librascope accumulator saved from previous instruction for error printout
LOUCI	Operand from drum
LIBCON	Location, unscaled value, and scale factor of a constant correction to the drum
LIBINS	Location and value of an instruction correction to the drum
LISTOP	The ENDRUN input value; the requested time to terminate the simulation
LTFC	Librascope value stored on the 9 flip-flops
MCODT	Floating point delta time to cutoff
MSCAN	Contents of the MSCAN
MSET	MSCAN settings in octal as read in to be stored in MTIMF table
MARVIE	Location of a telemetry error

7.2 STORAGE BLOCKS, Contd

SYMBOLIC NAME	CONTENTS
MSCTIM	Floating point times at which the MSCAN setting should be changed. Stored in MTIME table
NCYCLE	Frequency in computer cycles, with which the flight matrix will be printed (preflight)
OFFLOW	A flow value stored in PEXIT to set the first exit address to OUT TO FACE. (preflight)
OLDIE	Two cells, the current and previous instructions simulated
OTIME	Floating point time that corresponds to the short line accumulation. Used by open loop FEXT
PEXIT	The first FLOW value is set equal to PEXIT. Used in conjunction with beginning and ending the order code test simulation (preflight)
PHASEN	Phase counter (preflight) 1 = Phase A, 2 = Phase B, etc
PNCHID	Address of octal cards for telemetry output
PRTS	Seventy-two cells of the flight matrix
REV	Revolution counter in word times
REVWT	Number of word times at first word phase for printout
REVDR	Number of elapsed drum revolutions at first word phase for printout
RA	Quotient in division simulation
SAMREV	Revolution count at first word phase for printout
SEQCNT	Telemetry sequence count
SKPTIM	Times that frequency of printing the flight matrix should be changed. These value are stored in the SKPSTR table

7.2 STORAGE BLOCKS, Contd

SYMBOLIC NAME	CONTENTS
SCALE	Scale factor for initial constant printout
SKPRS	The next time the flight matrix should be printed
SMITH	Telemetry sequence counter for printout
STIME	Floating point simulation time when the revolution counter was set to zero previously.
SA	Coded value of sector ALPHA +1 of the current instruction
SB	Coded value +1 of sector BETA of the current instruction
S29	Initial sigmator lower word of time
S30	Initial sigmator upper word of time
SECT	Previous value of SA (sector ALPHA +1)
SONT	BCD location for printout
SPAC	Head spacing lag used by the sigmator update routine UPSIG
TSA	Coded ALPHA address +1 of current instruction
TSB	Coded BETA address +1 of current instruction
TBUF	Telemetry output buffer
TEMP	Intermediate storage cells
TIME	Floating point sigmator time used as reference by all input and output parameters
TMCO	Floating point time of main engine cutoff
TREF	Revolution count when last data will be completely telemetered

7.2 STORAGE BLOCKS, Contd

SYMBOLIC NAME	CONTENTS
TAPRC	Input parameter, FILES; count of files to be read from telemetry input tape
TDATA	Current data word being telemetered. Set up in the format specified by the Model-3 telemetry data converter
TELEM	Intermediate telemetry output buffer
TEMPE	Time that core was saved on tape
TITLE	Location from drum deck; 0 for title card
TELCNT	Number of words in the output buffer
TELREV	Revolution count when a value is stored for telemetry
TSSTIM	The length of the telemetry discrete in word-times
URN	Location of the previous instruction
VALUE	Octal correction when restarting from a tape
VTNEW	Velocity for printout (preflight)
WC	Torquing rate (preflight)
WBCT	Omega - earth spin rate (preflight)
WDS	Gyro errors (preflight)
WDRFT	Omega drift - torquing rate (preflight)
WITCH	The current entries from the WHICH, WHIC2 tables being examined
XADCT	Count of exit addresses used
ZREV	Revolution count when delta time is stored in the countdown sector for MECO
ZFLAG	Current entry from SGTRF table

7.3 TABLES

SYMBOLIC NAME	FUNCTION
3SGTBL	Contains the locations of velocity, position, and time sectors on the Model-3 sigmator
3SHORT	Short line lags in word times for the Model-3 sigmator
3XTRAK	Converts the unstarred track addresses to starred tracks for the Model-3 computer
COMPR	Locations of overflow errors to avoid duplications of error comments
COMTB	Locations of format statements for overflow error comments
CODTAB	Bit configurations of the code word that indicate phase branch points in the flight. Used to change the flight mode
DSFAC	Seventy-two scale factor values for the flight matrix printout; all in floating decimal
DTPSTR	Times that detailed print is desired. The times are in pairs, each consisting of a turn-on time and a turn-off time. All values are in floating decimal.
FLOW	Fifty cells containing flags that indicate the functions to be performed at each exit address. Tag contains the matrix handling flag. The decrement contains the exit flag. The address contains a patch number, or 0 if no patch is to be executed.
FORMT	Seventy-two BCD names to be printed in the initial flight matrix
LIBDA	For converting BCD to binary
MTIME	Alternating times and settings for changing the MSCAN setting
OPCOD	Mnemonic orders for detailed print

7.3 TABLES, Contd

SYMBOLIC NAME	FUNCTION
OUTAD	Output address as located in the drum
PATRA	Patch transfer vector
PRTIME	Times relative to the beginning of each phase, that the print out (preflight frequency of the matrix should be changed
SKPSTR	Alternating printout frequencies and times for printing the flight matrix
SGTBL	Sigmator location table for velocity, position, and time sectors of the Model-1 computer
SGTRF	Entries correspond to the sigmator location tables. The address contains the location of the floating point to Librascope conversion routine for each particular value. The decrement contains a 1, 2, or 3 for u-, v-, or w-position; and a 4, 5, or 6 for u-, v-, or w-velocity. The upper word of double words has a prefix of minus.
SHORT	Head spacing of short line read heads in word times for the Model-1 computer
SIGMA	The sigmator storage cells
WHIC2	Output choices for filling the flight matrix; read as WHICH in the inputs. Bits 0 through 35 represent positions 37 through 72 of the flight matrix.
WHICH	Output choices same as WHIC2 only bits 0 through 35 represent positions 1 through 36 of the flight matrix
XADDR	Exit address locations in core
XTRAK	Convert addresses from unstarred tracks to starred tracks for the Model-1 computer
ZSCALE	Sigmator scale factors for velocity, position, and time, floating point decimal

SECTION 8

LIST OF REFERENCES

1. H. B. Hilton, The COMBO Flight Program - A General Purpose Restricted Two-body Simulation, Report No. GDA63-0976, 15 October 1963.

2. T. E. Darnaud, COFLIC - A Simulation for Testing Flight Equations in Their Final Coded Form, Report No. GDA63-1054, 15 November 1963.

3. Operation and Maintenance Manual for Inertial Guidance System DYG8016A1 Computer, Report No. 61501-12, Honeywell Military Products Group, 1 August 1960.

4. Operation and Maintenance Manual for Inertial Guidance System DYG8016B1, Report No. 681781-M-2, Honeywell Military Products Group, 5 April 1963.

5. Vectran: Programming Manual and System Description, Report No. AOE63-0004, 8 February, 1963.

APPENDIX A

LICOS FLOW CHARTS

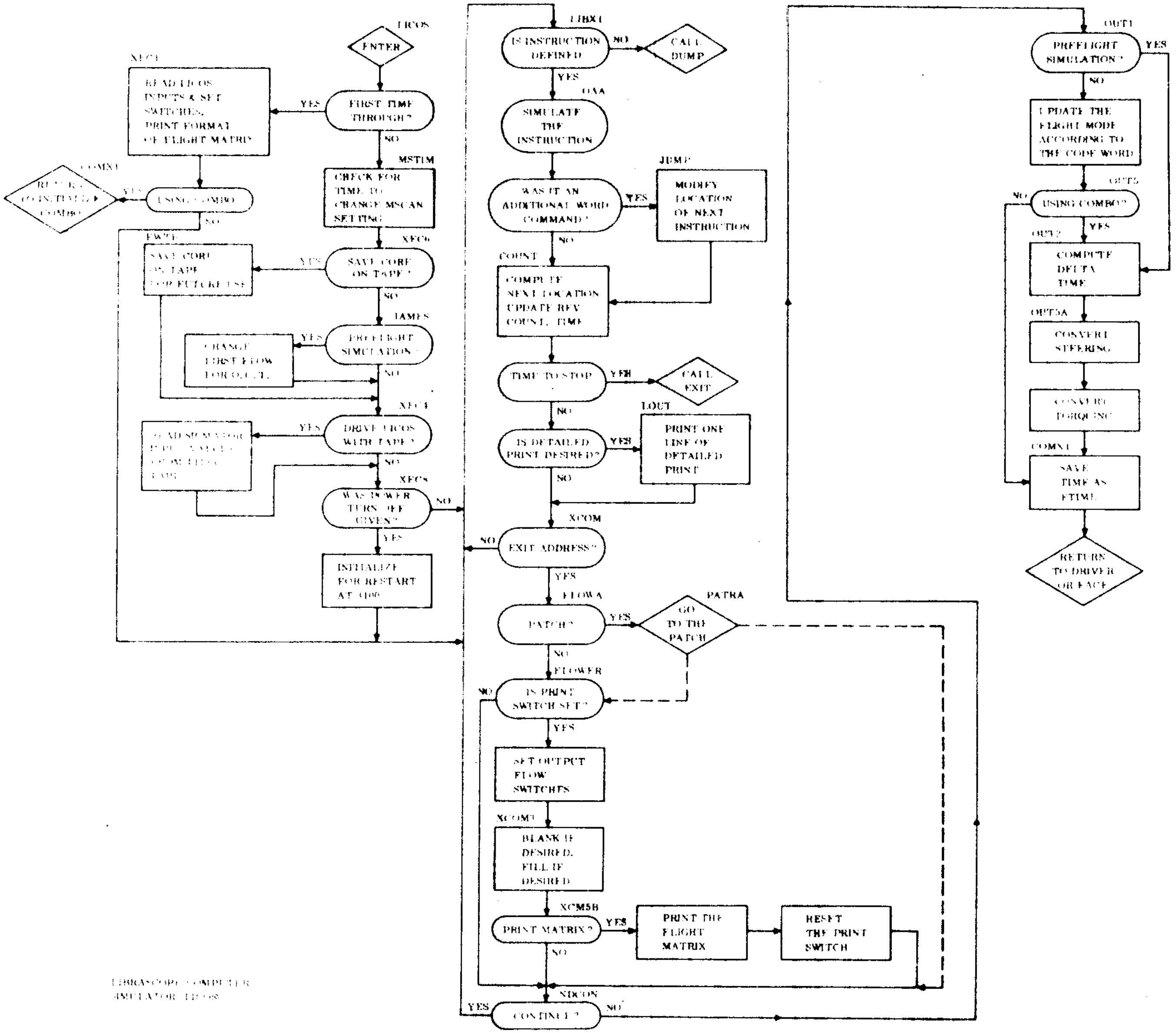
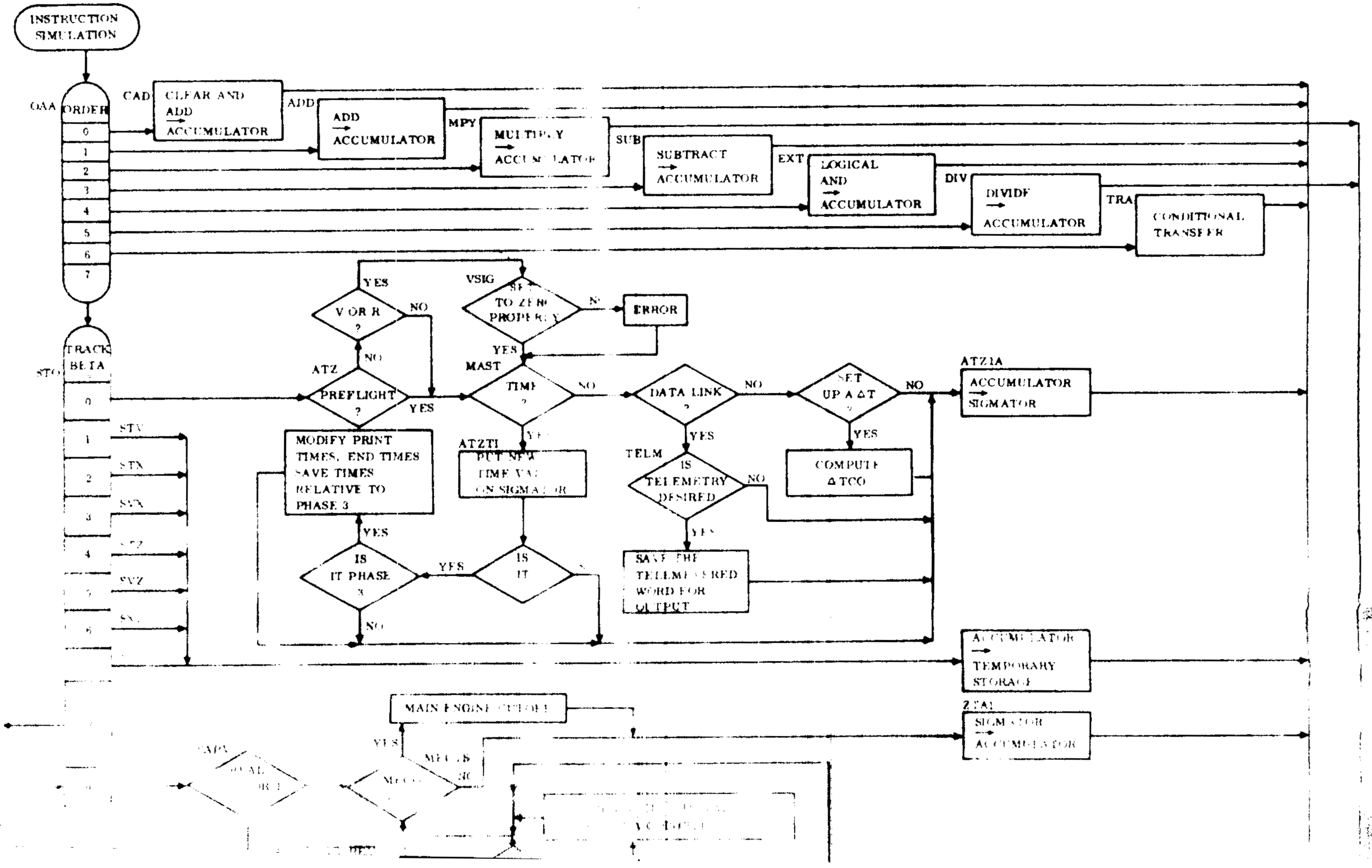


Figure A-1. Generalized Diagram of LICOS


INSTRUCTION SIMULATION
OAA
ORDER
0
1
2
3
4
5
6
7
CAD
CLEAR AND ADD → ACCUMULATOR
ADD
ADD → ACCUMULATOR
MPY
MULTIPLY → ACCUMULATOR
SUB
SUBTRACT → ACCUMULATOR
EXT
LOGICAL AND → ACCUMULATOR
DIV
DIVIDE → ACCUMULATOR
TRA
CONDITIONAL TRANSFER
STO
TRACK BETA
0
1
2
3
4
5
6
STV
STX
SVX
SVZ
ATZ
PREFLIGHT ?
YES
NO
V OR R ?
YES
NO
VSIG
SET TO ZERO PROPERLY ?
NO
ERROR
YES
MAST
TIME ?
NO
YES
ATZTI
PUT NEW TIME VALUE ON SIGMATOR
IS IT
YES
IS IT PHASE 3
YES
NO
MODIFY PRINT TIMES, END TIMES SAVE TIMES RELATIVE TO PHASE 3
DATA LINK ?
NO
YES
TELM
IS TELEMETRY DESIRED
NO
YES
SAVE THE TELEMETERED WORD FOR OUTPUT
SET UP A Δ T ?
NO
YES
COMPUTE Δ TCO
ATZ1A
ACCUMULATOR → SIGMATOR
ACCUMULATOR → TEMPORARY STORAGE
ZTA1
SIGMATOR → ACCUMULATOR
MAIN ENGINE CUTOFF
YES
NO
MECO
CAPY

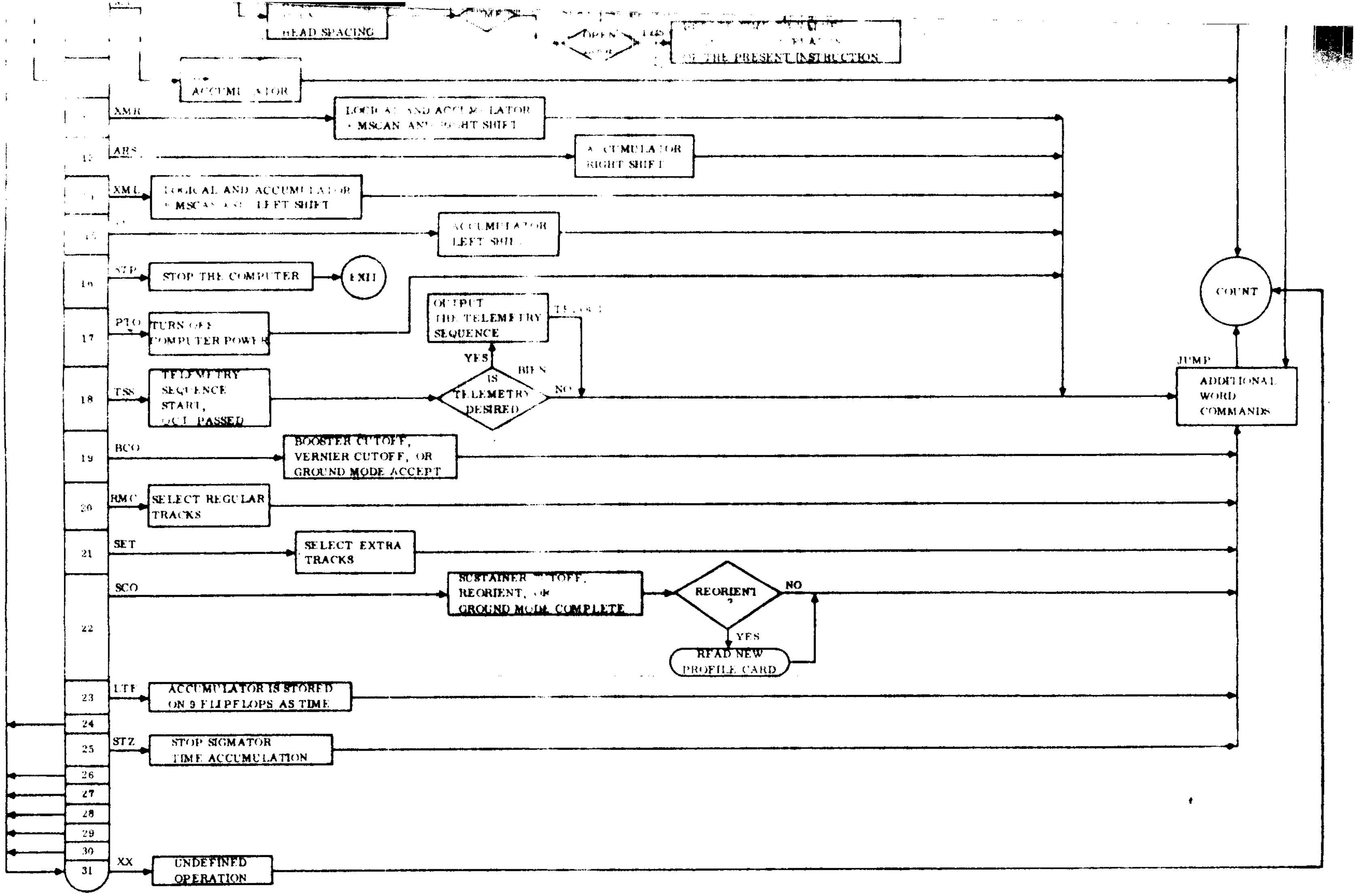

HEAD SPACING
OPEN
OF THE PRESENT INSTRUCTION
ACCUMULATOR
XMR
LOGICAL AND ACCUMULATOR MSCAN AND RIGHT SHIFT
ABS
ACCUMULATOR RIGHT SHIFT
XML
LOGICAL AND ACCUMULATOR MSCAN AND LEFT SHIFT
ACCUMULATOR LEFT SHIFT
STP
16
STOP THE COMPUTER
EXIT
PTO
17
TURN OFF COMPUTER POWER
OUTPUT THE TELEMETRY SEQUENCE
TSS
18
TELEMETRY SEQUENCE START, QCL PASSED
YES
IS TELEMETRY DESIRED
BIEN
NO
BCO
19
BOOSTER CUTOFF, VERNIER CUTOFF, OR GROUND MODE ACCEPT
RMC
20
SELECT REGULAR TRACKS
SET
21
SELECT EXTRA TRACKS
SCO
22
SUSTAINER CUTOFF, REORIENT, OR GROUND MODE COMPLETE
REORIENT ?
NO
YES
READ NEW PROFILE CARD
LTF
23
ACCUMULATOR IS STORED ON 9 FLIPFLOPS AS TIME
24
STZ
25
STOP SIGMATOR TIME ACCUMULATION
26
27
28
29
30
XX
31
UNDEFINED OPERATION
COUNT
JUMP
ADDITIONAL WORD COMMANDS

Figure A-2. General Diagram of Instruction Simulation

2

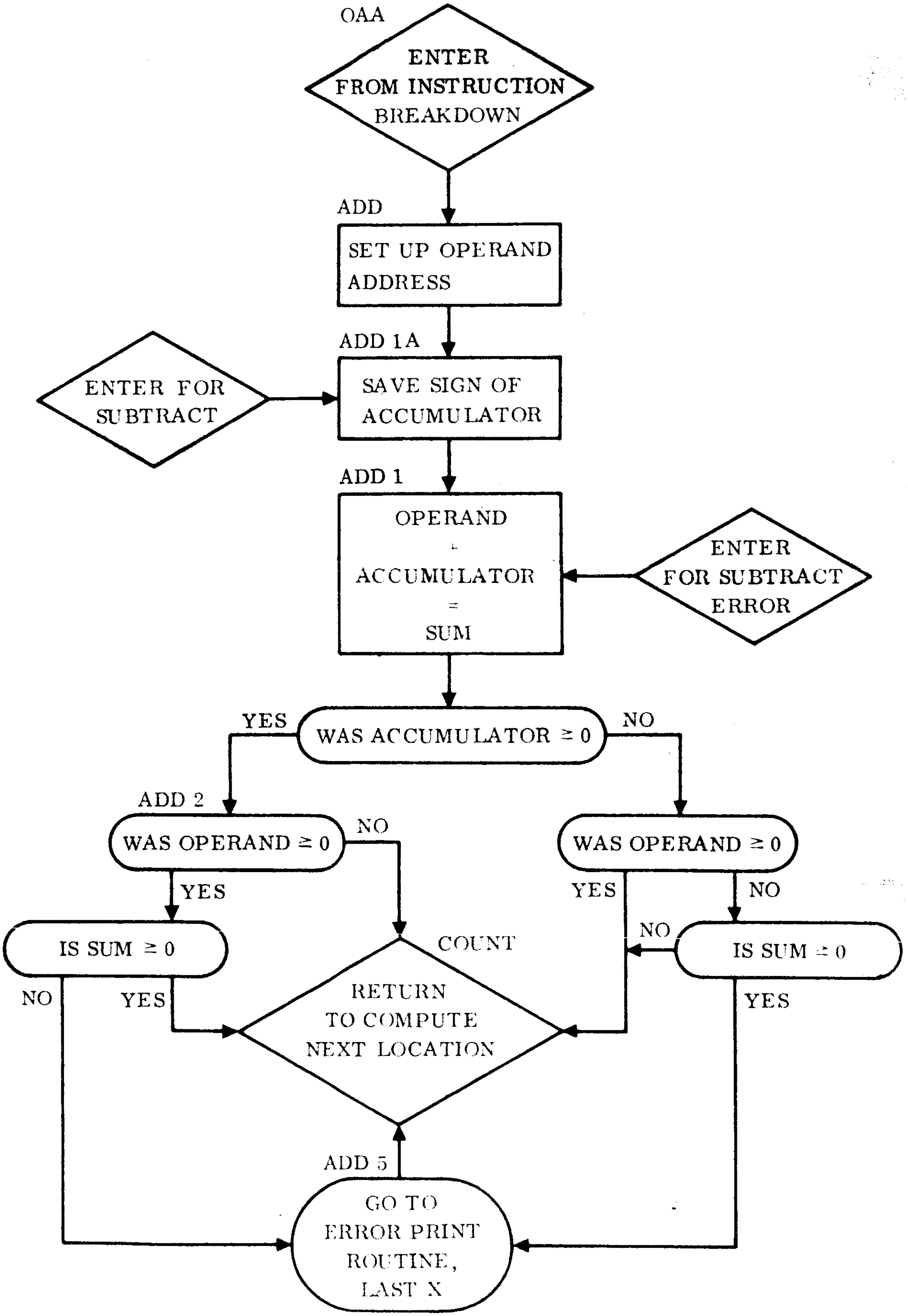


Figure A-3. Instruction Simulation-Add

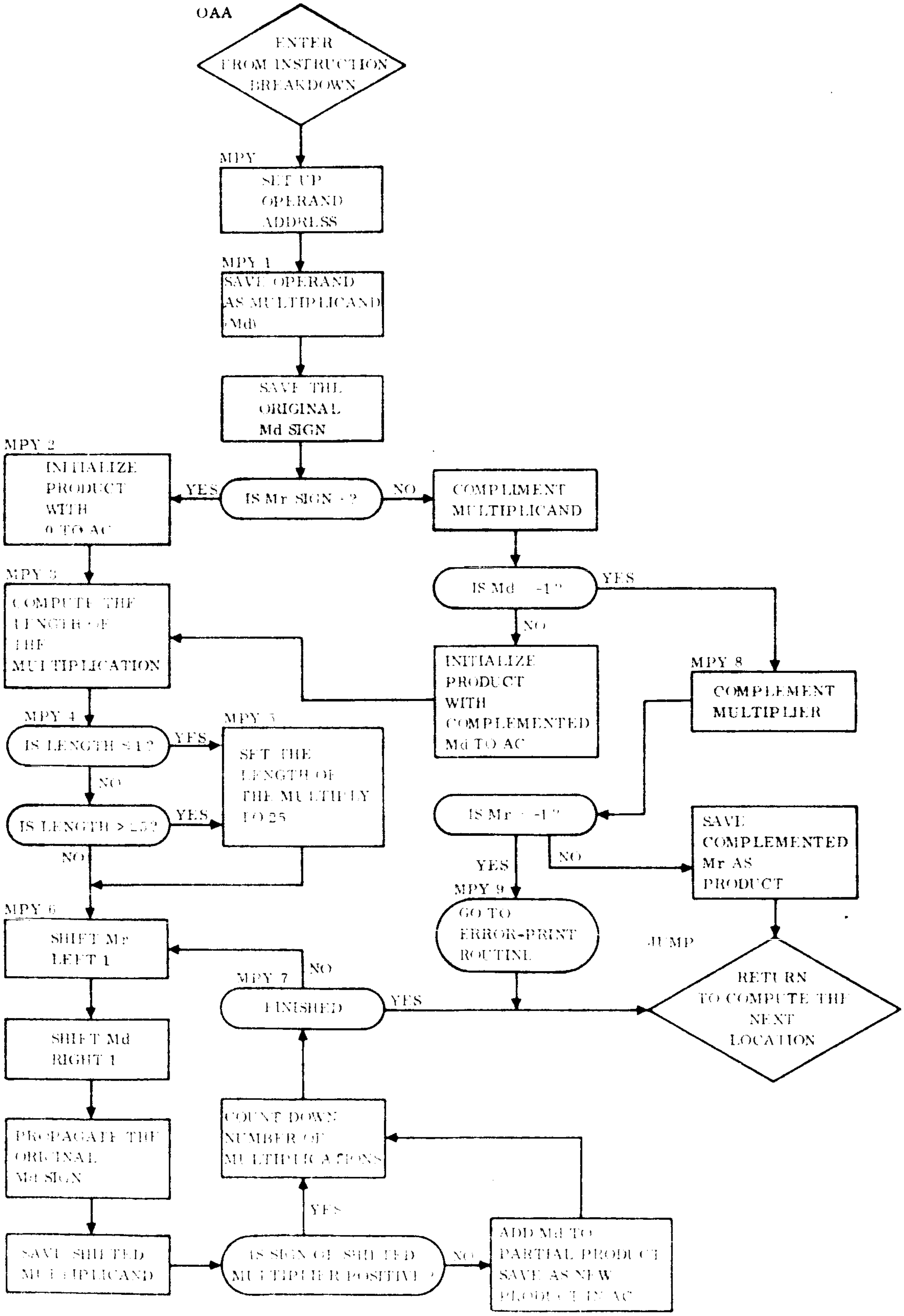

OAA
ENTER FROM INSTRUCTION BREAKDOWN
MPY
SET UP OPERAND ADDRESS
MPY 1
SAVE OPERAND AS MULTIPLICAND (Md)
SAVE THE ORIGINAL Md SIGN
MPY 2
INITIALIZE PRODUCT WITH 0 TO AC
YES
IS Mr SIGN + ?
NO
COMPLIMENT MULTIPLICAND
IS Md = -1 ?
YES
NO
INITIALIZE PRODUCT WITH COMPLEMENTED Md TO AC
MPY 8
COMPLEMENT MULTIPLIER
MPY 3
COMPUTE THE LENGTH OF THE MULTIPLICATION
MPY 4
IS LENGTH ≤ 1 ?
YES
MPY 5
SET THE LENGTH OF THE MULTIPLY TO 25
NO
IS LENGTH > 25 ?
YES
NO
IS Mr = -1 ?
NO
SAVE COMPLEMENTED Mr AS PRODUCT
YES
MPY 9
GO TO ERROR-PRINT ROUTINE
JUMP
MPY 6
SHIFT Mr LEFT 1
NO
MPY 7
FINISHED
YES
RETURN TO COMPUTE THE NEXT LOCATION
SHIFT Md RIGHT 1
COUNT DOWN NUMBER OF MULTIPLICATIONS
PROPAGATE THE ORIGINAL Md SIGN
YES
SAVE SHIFTED MULTIPLICAND
IS SIGN OF SHIFTED MULTIPLIER POSITIVE ?
NO
ADD Md TO PARTIAL PRODUCT SAVE AS NEW PRODUCT IN AC

Figure A-4. Instruction Simulation-Multiply

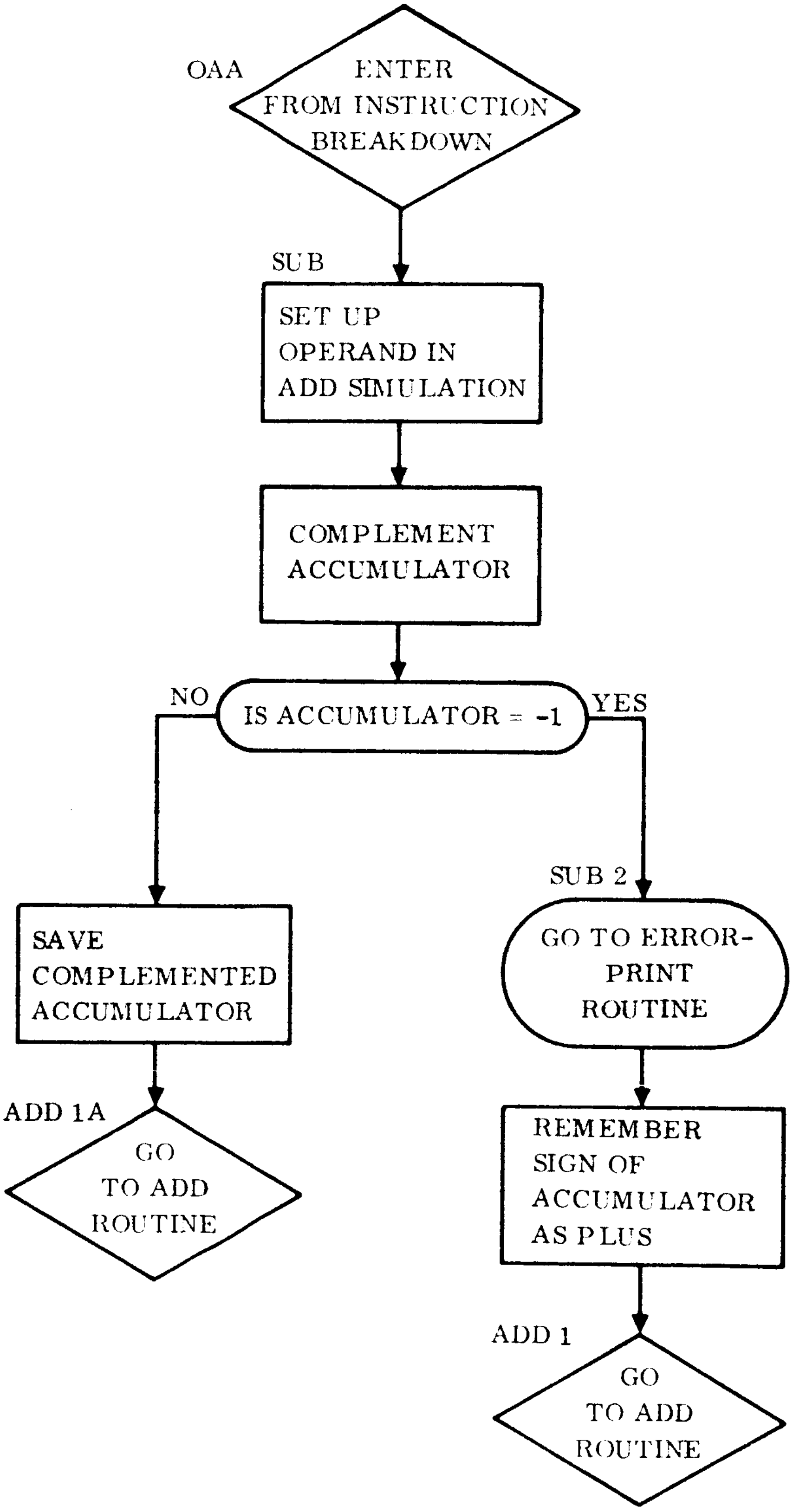

OAA
ENTER FROM INSTRUCTION BREAKDOWN
SUB
SET UP OPERAND IN ADD SIMULATION
COMPLEMENT ACCUMULATOR
NO
IS ACCUMULATOR = -1
YES
SAVE COMPLEMENTED ACCUMULATOR
ADD 1A
GO TO ADD ROUTINE
SUB 2
GO TO ERROR-PRINT ROUTINE
REMEMBER SIGN OF ACCUMULATOR AS PLUS
ADD 1
GO TO ADD ROUTINE

Figure A-5. Instruction Simulation-Subtract

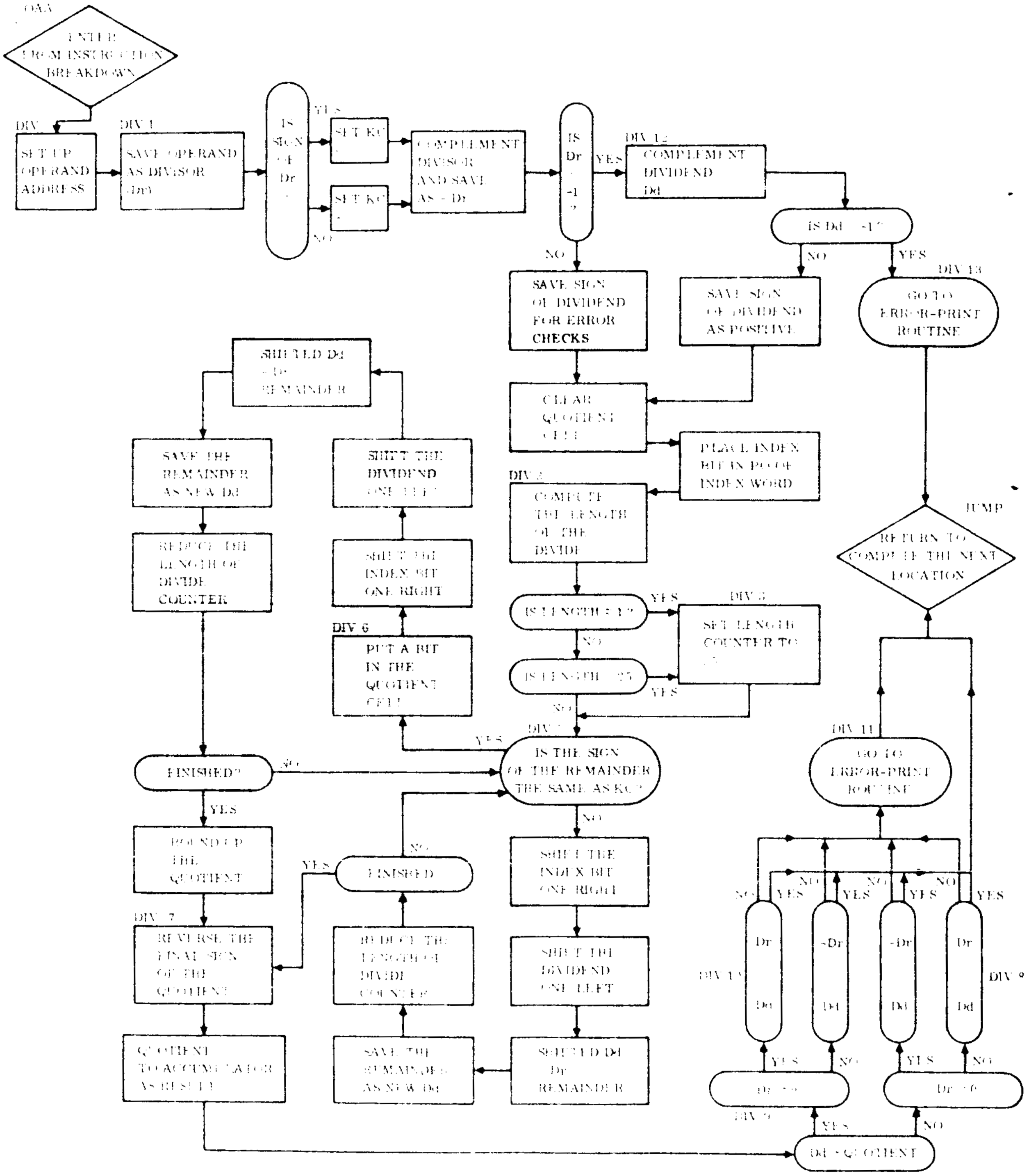

OAA
ENTER FROM INSTRUCTION BREAKDOWN
DIV
SET UP OPERAND ADDRESS
DIV 1
SAVE OPERAND AS DIVISOR (Dr)
IS SIGN OF Dr -
YES
SET KC -
NO
SET KC +
COMPLEMENT DIVISOR AND SAVE AS - Dr
IS Dr = -1 ?
YES
NO
DIV 12
COMPLEMENT DIVIDEND Dd
IS Dd = -1 ?
NO
YES
DIV 13
GO TO ERROR-PRINT ROUTINE
SAVE SIGN OF DIVIDEND FOR ERROR CHECKS
SAVE SIGN OF DIVIDEND AS POSITIVE
CLEAR QUOTIENT CELL
PLACE INDEX BIT IN P0 OF INDEX WORD
DIV 2
COMPUTE THE LENGTH OF THE DIVIDE
JUMP
RETURN TO COMPLETE THE NEXT LOCATION
IS LENGTH ≤ 1 ?
YES
NO
DIV 3
SET LENGTH COUNTER TO 25
IS LENGTH > 25
YES
NO
DIV 4
IS THE SIGN OF THE REMAINDER THE SAME AS KC ?
YES
NO
SHIFTED Dd + Dr REMAINDER
SAVE THE REMAINDER AS NEW Dd
SHIFT THE DIVIDEND ONE LEFT
REDUCE THE LENGTH OF DIVIDE COUNTER
SHIFT THE INDEX BIT ONE RIGHT
DIV 6
PUT A BIT IN THE QUOTIENT CELL
FINISHED?
NO
YES
ROUND UP THE QUOTIENT
FINISHED
YES
NO
DIV 7
REVERSE THE FINAL SIGN OF THE QUOTIENT
REDUCE THE LENGTH OF DIVIDE COUNTER
SHIFT THE INDEX BIT ONE RIGHT
SHIFT THE DIVIDEND ONE LEFT
QUOTIENT TO ACCUMULATOR AS RESULT
SAVE THE REMAINDER AS NEW Dd
SHIFTED Dd Dr REMAINDER
DIV 11
GO TO ERROR-PRINT ROUTINE
NO
YES
NO
YES
NO
YES
NO
YES
Dr Dd
-Dr Dd
-Dr Dd
Dr Dd
DIV 10
DIV 9
YES
NO
YES
NO
Dr > 0
Dr > 0
DIV 8
YES
NO
Dd • QUOTIENT

Figure A-6. Instruction Simulation-Divide

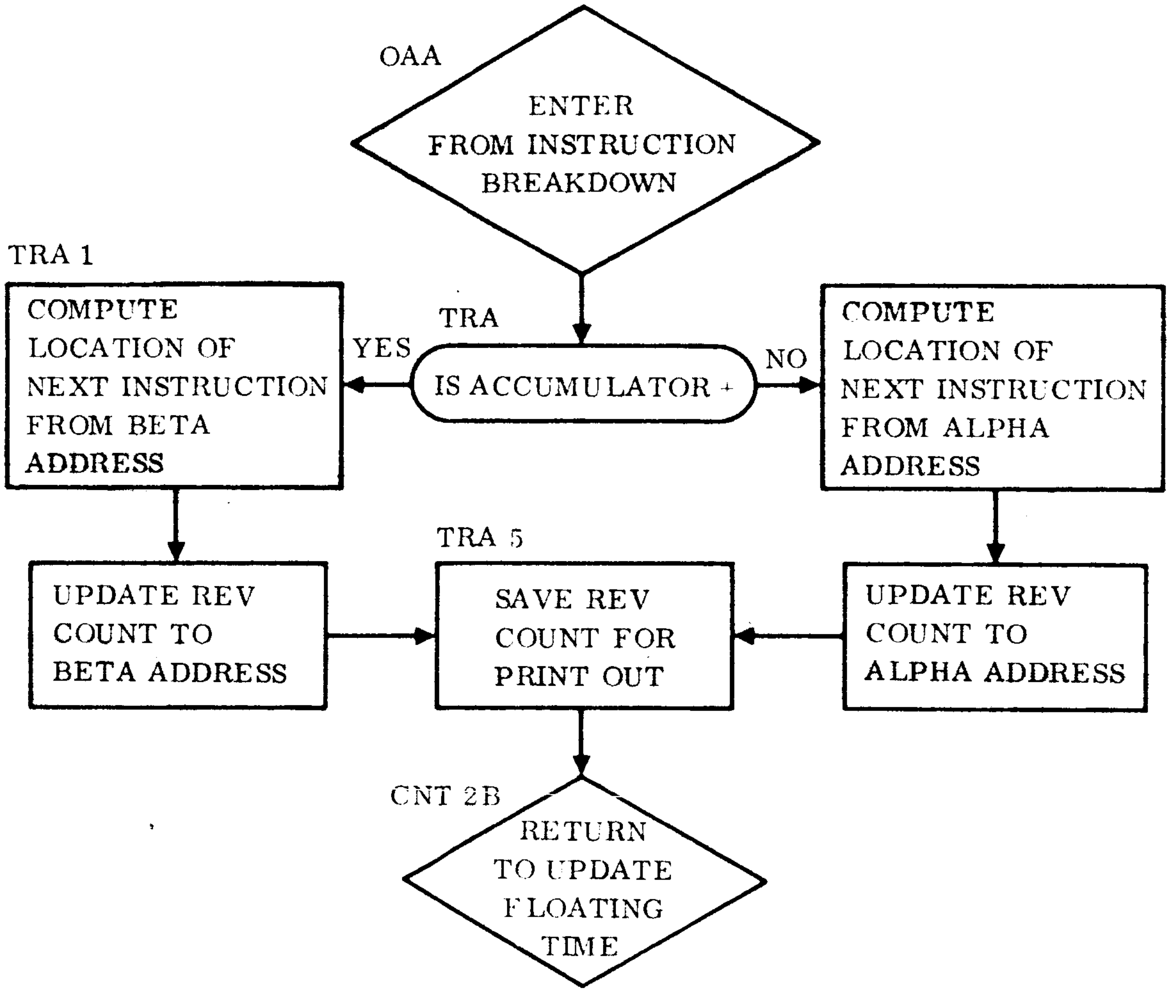

OAA
ENTER FROM INSTRUCTION BREAKDOWN
TRA 1
COMPUTE LOCATION OF NEXT INSTRUCTION FROM BETA ADDRESS
TRA
YES
IS ACCUMULATOR +
NO
COMPUTE LOCATION OF NEXT INSTRUCTION FROM ALPHA ADDRESS
TRA 5
UPDATE REV COUNT TO BETA ADDRESS
SAVE REV COUNT FOR PRINT OUT
UPDATE REV COUNT TO ALPHA ADDRESS
CNT 2B
RETURN TO UPDATE FLOATING TIME

Figure A-7. Instruction Simulation-Conditional Transfer

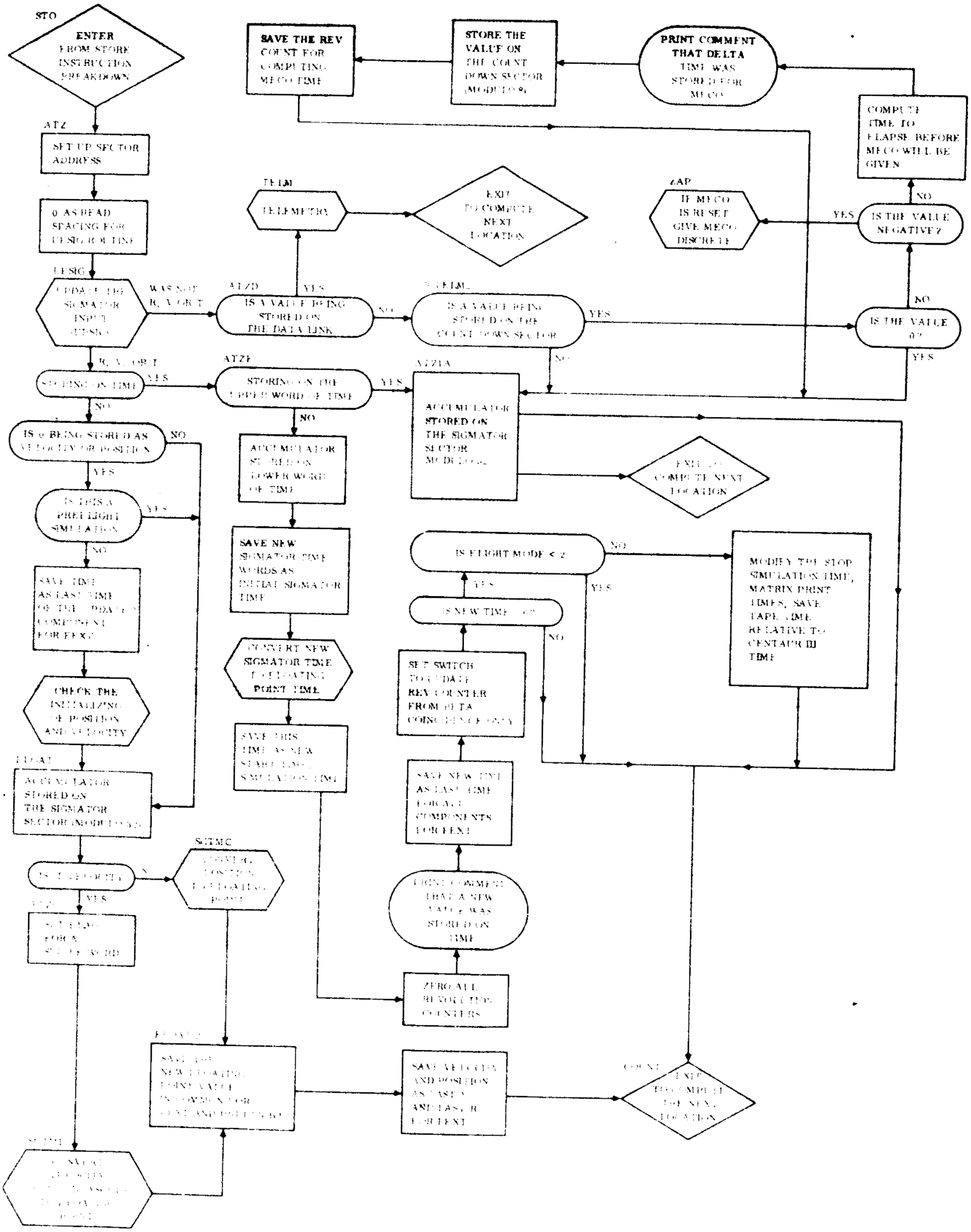

STO
ENTER FROM STORE INSTRUCTION BREAKDOWN
ATZ
SET UP SECTOR ADDRESS
0 AS HEAD SPACING FOR UESIG ROUTINE
UESIG
UPDATE THE SIGMATOR INPUT (UESIG)
WAS NOT R, V OR T
R, V OR T
STORING ON TIME
YES
NO
IS 0 BEING STORED AS VELOCITY OR POSITION
NO
YES
IS THIS A PREFLIGHT SIMULATION
YES
NO
SAVE TIME AS LAST TIME OF THE UPDATED COMPONENT FOR FEXC
CHECK THE INITIALIZING OF POSITION AND VELOCITY
FLOAT
ACCUMULATOR STORED ON THE SIGMATOR SECTOR (MODULO 32)
IS IT VELOCITY
YES
SET FLAG FOR A SCALE WORD
SITMC
CONVERT POSITION TO FLOATING POINT
CONVERT VELOCITY TO FLOATING POINT
SAVE THE NEW FLOATING POINT VALUE IN COMMON FOR CENT AND DYNAMICS
TELM
TELEMETRY
EXIT TO COMPUTE NEXT LOCATION
ATZD
YES
IS A VALUE BEING STORED ON THE DATA LINK
NO
ATZE
STORING ON THE UPPER WORD OF TIME
YES
NO
ACCUMULATOR STORED ON LOWER WORD OF TIME
SAVE NEW SIGMATOR TIME WORDS AS INITIAL SIGMATOR TIME
CONVERT NEW SIGMATOR TIME TO FLOATING POINT TIME
SAVE THIS TIME AS NEW START LAG SIMULATION TIME
ZERO ALL REVOLUTION COUNTERS
PRINT COMMENT THAT A NEW VALUE WAS STORED ON TIME
SAVE NEW TIME AS LAST TIME FOR ALL COMPONENTS FOR FEXT
SET SWITCH TO UPDATE REV COUNTER FROM BETA COINCIDENCE ONLY
IS NEW TIME = 0?
YES
NO
IS FLIGHT MODE < 2
NO
YES
MODIFY THE STOP SIMULATION TIME, MATRIX PRINT TIMES, SAVE TAPE TIME RELATIVE TO CENTAUR III TIME
SAVE VELOCITY AND POSITION AS LAST V AND LAST R FOR FEXT
COUNT
EXIT TO COMPUTE THE NEXT LOCATION
ATZLA
ACCUMULATOR STORED ON THE SIGMATOR SECTOR MODULO 32
EXIT TO COMPUTE NEXT LOCATION
ATELM
IS A VALUE BEING STORED ON THE COUNT DOWN SECTOR
YES
NO
IS THE VALUE 0?
NO
YES
IS THE VALUE NEGATIVE?
YES
NO
ZAP
IF MECO IS RESET GIVE MECO DISCRETE
COMPUTE TIME TO ELAPSE BEFORE MECO WILL BE GIVEN
PRINT COMMENT THAT DELTA TIME WAS STORED FOR MECO
STORE THE VALUE ON THE COUNT DOWN SECTOR (MODULO 8)
SAVE THE REV COUNT FOR COMPUTING MECO TIME

Figure A-8. Accumulator to Sigmator

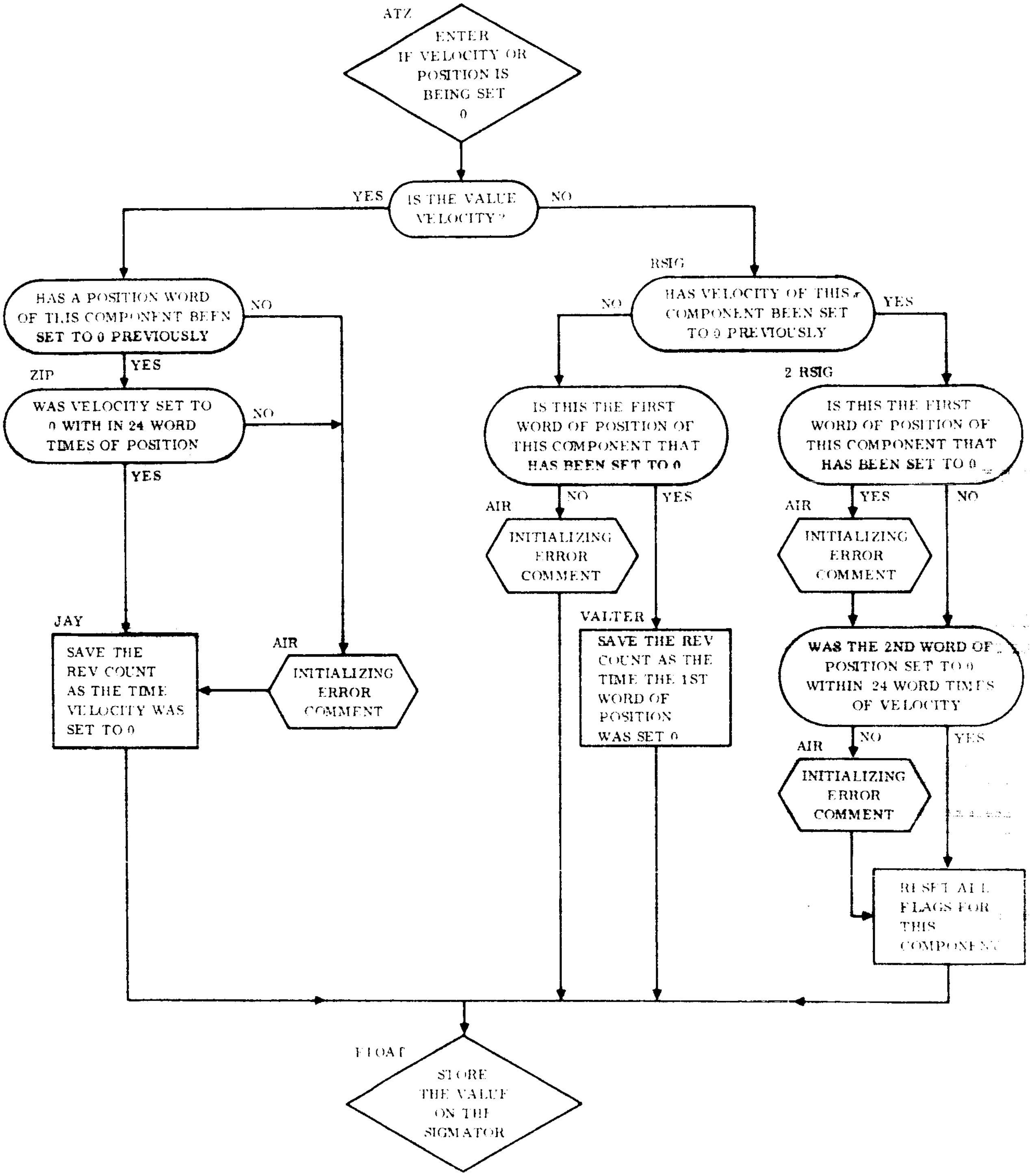

ATZ
ENTER IF VELOCITY OR POSITION IS BEING SET 0
YES
IS THE VALUE VELOCITY ?
NO
HAS A POSITION WORD OF THIS COMPONENT BEEN SET TO 0 PREVIOUSLY
NO
YES
ZIP
WAS VELOCITY SET TO 0 WITH IN 24 WORD TIMES OF POSITION
NO
YES
RSIG
HAS VELOCITY OF THIS COMPONENT BEEN SET TO 0 PREVIOUSLY
NO
YES
2 RSIG
IS THIS THE FIRST WORD OF POSITION OF THIS COMPONENT THAT HAS BEEN SET TO 0
NO
YES
AIR
INITIALIZING ERROR COMMENT
IS THIS THE FIRST WORD OF POSITION OF THIS COMPONENT THAT HAS BEEN SET TO 0
YES
NO
AIR
INITIALIZING ERROR COMMENT
JAY
SAVE THE REV COUNT AS THE TIME VELOCITY WAS SET TO 0
AIR
INITIALIZING ERROR COMMENT
VALTER
SAVE THE REV COUNT AS THE TIME THE 1ST WORD OF POSITION WAS SET 0
WAS THE 2ND WORD OF POSITION SET TO 0 WITHIN 24 WORD TIMES OF VELOCITY
NO
YES
AIR
INITIALIZING ERROR COMMENT
RESET ALL FLAGS FOR THIS COMPONENT
FLOAT
STORE THE VALUE ON THE SIGMATOR

Figure A-9. Check Initializing R & V

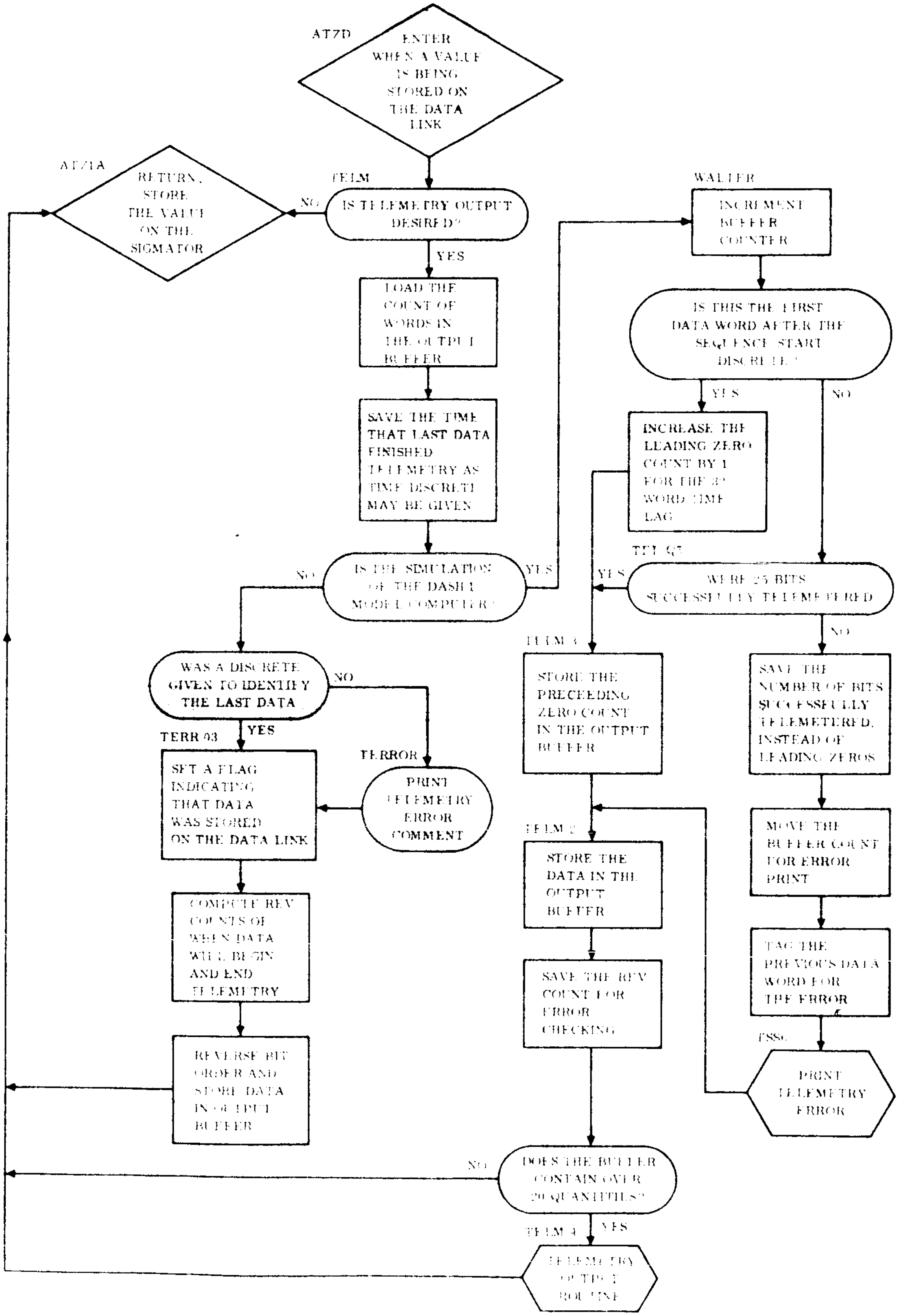

AT7D
ENTER WHEN A VALUE IS BEING STORED ON THE DATA LINK
AT71A
RETURN, STORE THE VALUE ON THE SIGMATOR
TELM
IS TELEMETRY OUTPUT DESIRED?
NO
YES
LOAD THE COUNT OF WORDS IN THE OUTPUT BUFFER
SAVE THE TIME THAT LAST DATA FINISHED TELEMETRY AS TIME DISCRETE MAY BE GIVEN
IS THE SIMULATION OF THE DASH 1 MODEL COMPUTER?
NO
YES
WAS A DISCRETE GIVEN TO IDENTIFY THE LAST DATA
NO
YES
TERR 03
SET A FLAG INDICATING THAT DATA WAS STORED ON THE DATA LINK
TERROR
PRINT TELEMETRY ERROR COMMENT
COMPUTE REV COUNTS OF WHEN DATA WILL BEGIN AND END TELEMETRY
REVERSE BIT ORDER AND STORE DATA IN OUTPUT BUFFER
WALTER
INCREMENT BUFFER COUNTER
IS THIS THE FIRST DATA WORD AFTER THE SEQUENCE START DISCRETE?
YES
NO
INCREASE THE LEADING ZERO COUNT BY 1 FOR THE 32 WORD TIME LAG
TELQ7
WERE 25 BITS SUCCESSFULLY TELEMETERED
YES
NO
TELM 3
STORE THE PRECEEDING ZERO COUNT IN THE OUTPUT BUFFER
SAVE THE NUMBER OF BITS SUCCESSFULLY TELEMETERED, INSTEAD OF LEADING ZEROS
TELM 2
STORE THE DATA IN THE OUTPUT BUFFER
SAVE THE REV COUNT FOR ERROR CHECKING
MOVE THE BUFFER COUNT FOR ERROR PRINT
TAG THE PREVIOUS DATA WORD FOR THE ERROR
FSSO
PRINT TELEMETRY ERROR
DOES THE BUFFER CONTAIN OVER 20 QUANTITIES?
NO
YES
TELM 4
TELEMETRY OUTPUT ROUTINE

Figure A-10. Store on the Data Link

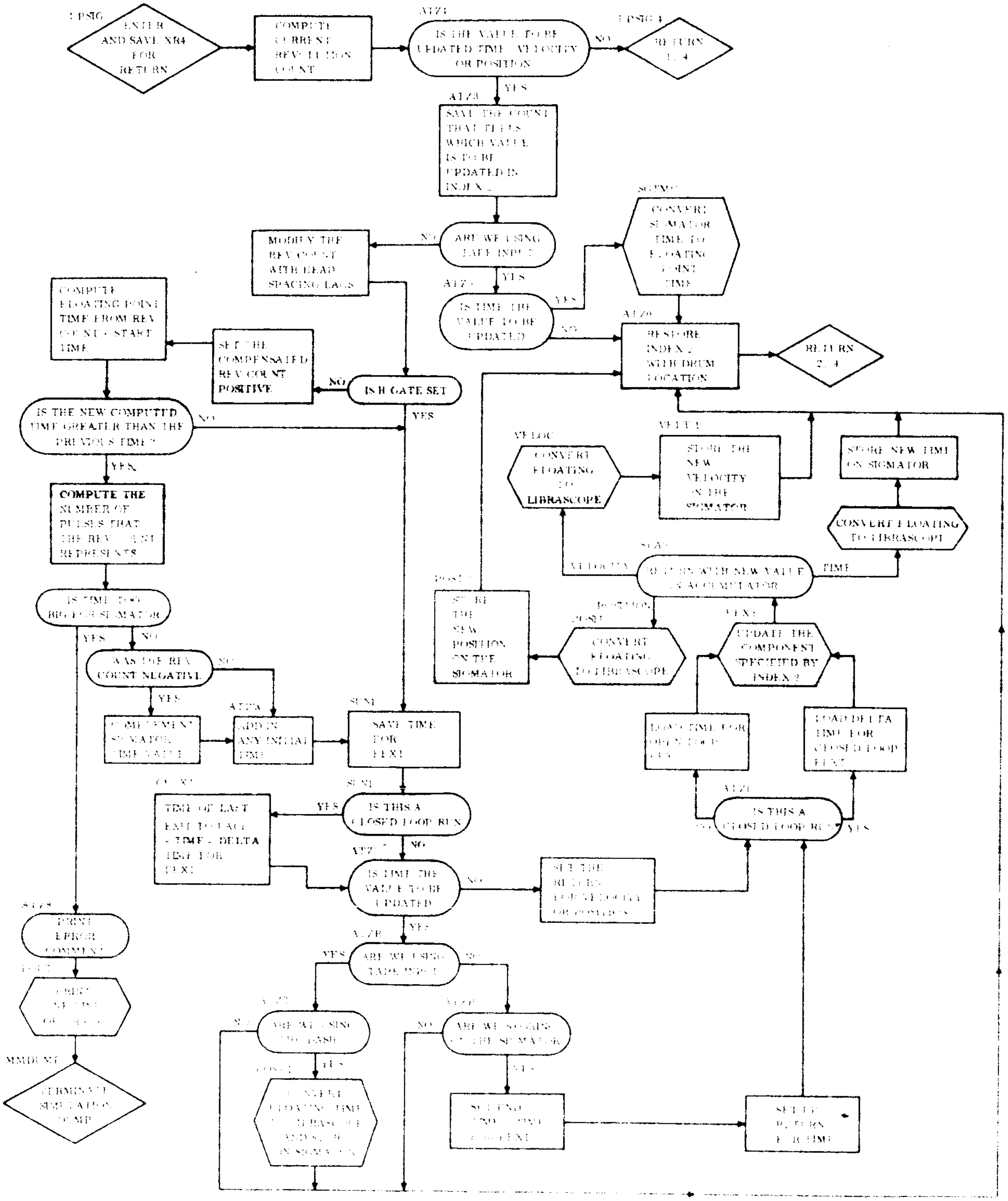

UPSIG
ENTER AND SAVE XR4 FOR RETURN
COMPUTE CURRENT REVOLUTION COUNT
ATZ1
IS THE VALUE TO BE UPDATED TIME VELOCITY OR POSITION
NO
UPSIG 4
RETURN 1, 4
YES
ATZ3
SAVE THE COUNT THAT TELLS WHICH VALUE IS TO BE UPDATED IN INDEX 2
ARE WE USING TAPE INPUT
NO
MODIFY THE REV COUNT WITH HEAD SPACING LAGS
YES
ATZ
IS TIME THE VALUE TO BE UPDATED
YES
SGPMO
CONVERT SIGMATOR TIME TO FLOATING POINT TIME
NO
ATZ0
RESTORE INDEX 2 WITH DRUM LOCATION
RETURN 2, 4
COMPUTE FLOATING POINT TIME FROM REV COUNT + START TIME
SET THE COMPENSATED REV COUNT POSITIVE
NO
IS R GATE SET
YES
IS THE NEW COMPUTED TIME GREATER THAN THE PREVIOUS TIME ?
NO
YES
COMPUTE THE NUMBER OF PULSES THAT THE REV COUNT REPRESENTS
IS TIME TOO BIG FOR SIGMATOR
YES
NO
WAS THE REV COUNT NEGATIVE
NO
YES
COMPLEMENT SIGMATOR TIME VALUE
ATZA
ADD IN ANY INITIAL TIME
SUN1
SAVE TIME FOR FEXT
VELOC
CONVERT FLOATING TO LIBRASCOPE
VELT
STORE THE NEW VELOCITY ON THE SIGMATOR
STORE NEW TIME ON SIGMATOR
CONVERT FLOATING TO LIBRASCOPE
VELOCITY
RETURN WITH NEW VALUE IN ACCUMULATOR
TIME
POST
STORE THE NEW POSITION ON THE SIGMATOR
POSITION
POSIT
CONVERT FLOATING TO LIBRASCOPE
FEXT
UPDATE THE COMPONENT SPECIFIED BY INDEX 2
LOAD TIME FOR OPEN LOOP RUN
LOAD DELTA TIME FOR CLOSED LOOP FEXT
ATZ1
IS THIS A CLOSED LOOP RUN
NO
YES
SUN1
TIME OF LAST EXIT TO FACT + TIME + DELTA TIME FOR FEXT
YES
IS THIS A CLOSED LOOP RUN
NO
ATZ
IS TIME THE VALUE TO BE UPDATED
NO
SET THE RETURN FOR VELOCITY OR POSITION
YES
ATZB
YES
ARE WE USING TAPE INPUT
NO
ATZ8
PRINT ERROR COMMENT
PRINT
MMDUMP
TERMINATE SIMULATION DUMP
ATZ
NO
ARE WE USING THE DASH
YES
CONVERT FLOATING TIME TO LIBRASCOPE AND STORE IN SIGMATOR
ATZ
NO
ARE WE STORING ON THE SIGMATOR
YES
SET END POINT
SET UP RETURN FOR TIME

Figure A-11. Update Sigmator Value

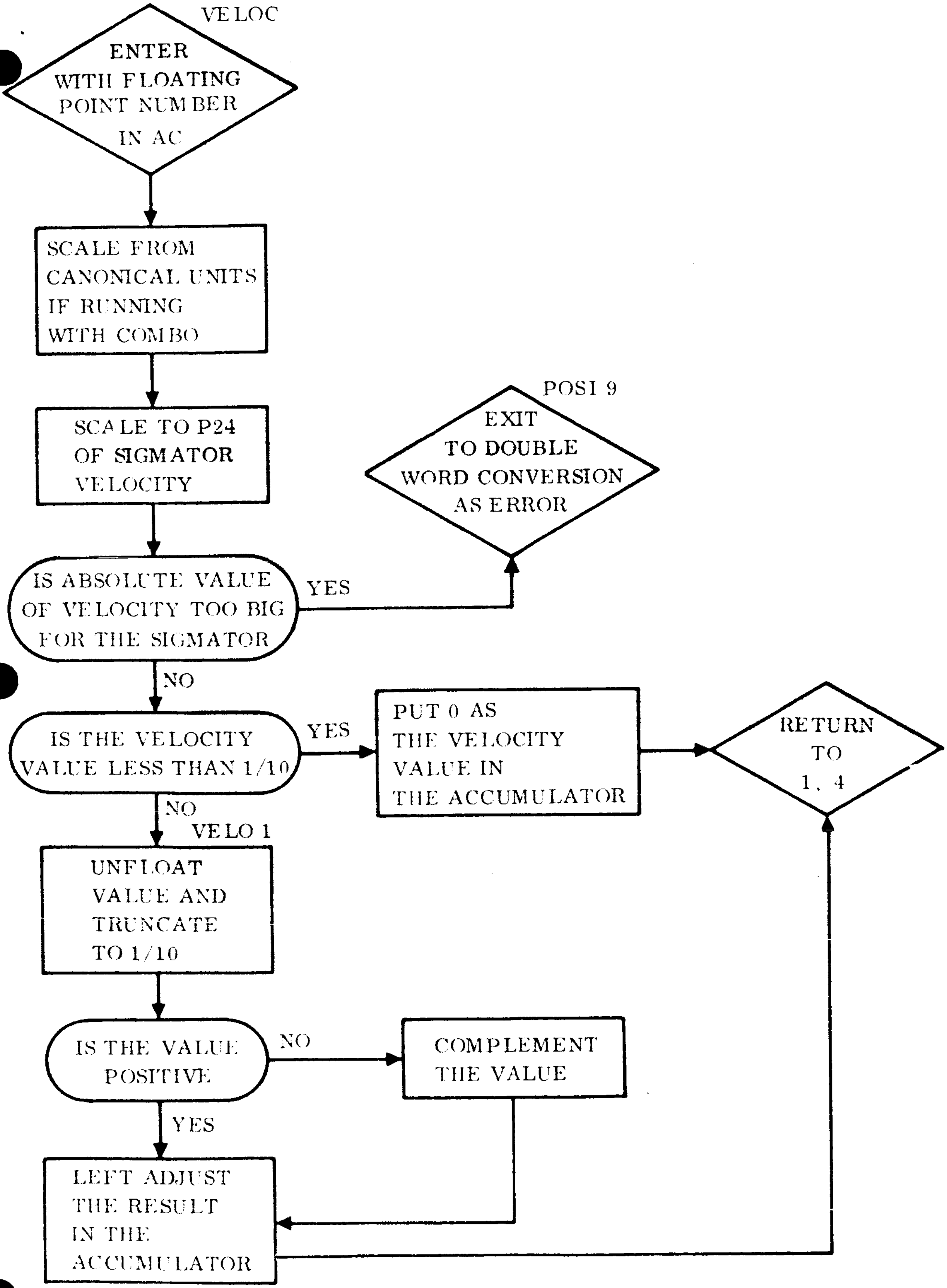

VELOC
ENTER WITH FLOATING POINT NUMBER IN AC
SCALE FROM CANONICAL UNITS IF RUNNING WITH COMBO
SCALE TO P24 OF SIGMATOR VELOCITY
IS ABSOLUTE VALUE OF VELOCITY TOO BIG FOR THE SIGMATOR
YES
POSI 9
EXIT TO DOUBLE WORD CONVERSION AS ERROR
NO
IS THE VELOCITY VALUE LESS THAN 1/10
YES
PUT 0 AS THE VELOCITY VALUE IN THE ACCUMULATOR
RETURN TO 1, 4
NO
VELO 1
UNFLOAT VALUE AND TRUNCATE TO 1/10
IS THE VALUE POSITIVE
NO
COMPLEMENT THE VALUE
YES
LEFT ADJUST THE RESULT IN THE ACCUMULATOR

Figure A-12. Convert Floating Point Number to a Single Librascope Word (Velocity)

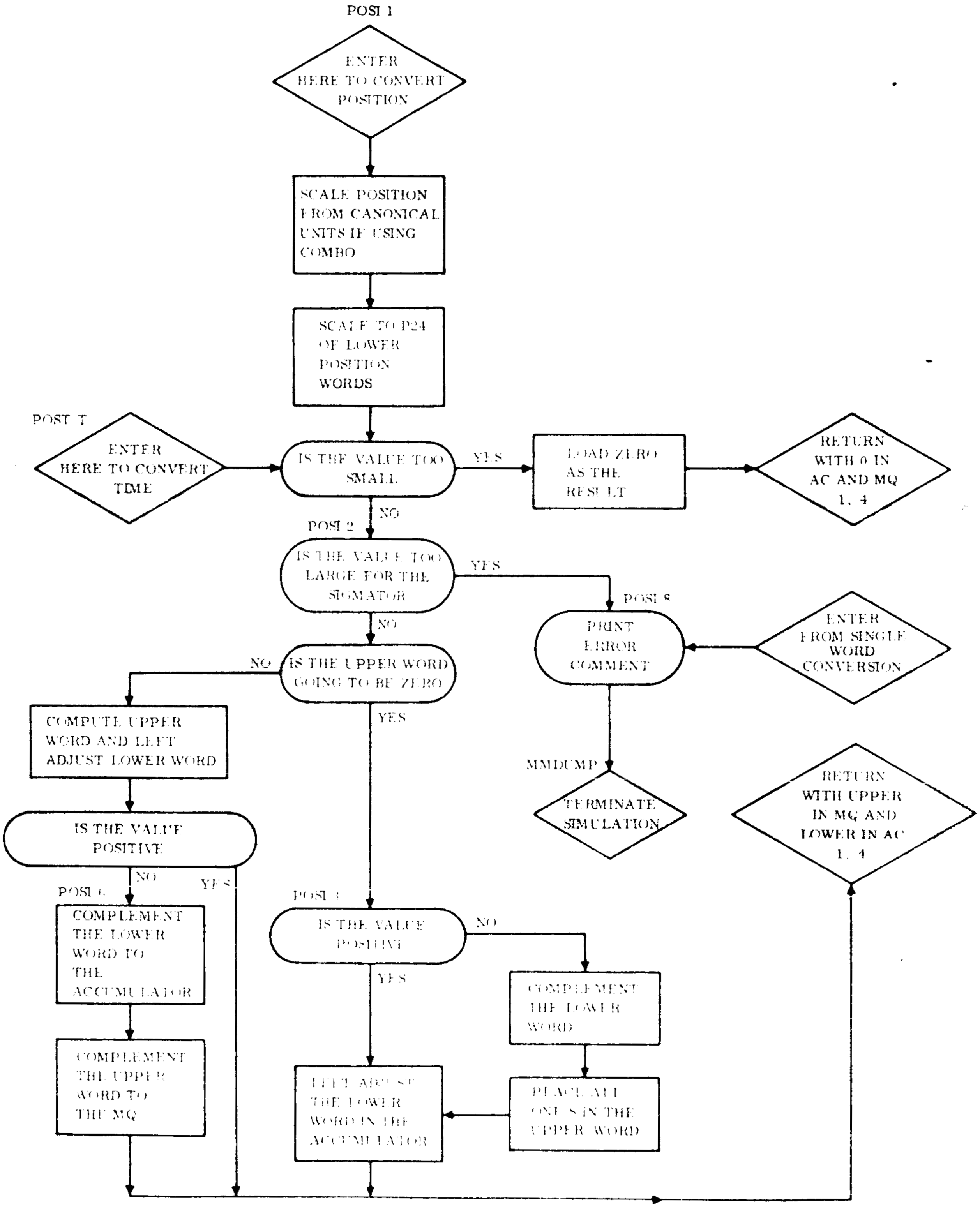

POSI 1
ENTER HERE TO CONVERT POSITION
SCALE POSITION FROM CANONICAL UNITS IF USING COMBO
SCALE TO P24 OF LOWER POSITION WORDS
POST T
ENTER HERE TO CONVERT TIME
IS THE VALUE TOO SMALL
YES
LOAD ZERO AS THE RESULT
RETURN WITH 0 IN AC AND MQ 1, 4
NO
POSI 2
IS THE VALUE TOO LARGE FOR THE SIGMATOR
YES
POSI 8
PRINT ERROR COMMENT
ENTER FROM SINGLE WORD CONVERSION
NO
NO
IS THE UPPER WORD GOING TO BE ZERO
YES
COMPUTE UPPER WORD AND LEFT ADJUST LOWER WORD
MMDUMP
TERMINATE SIMULATION
RETURN WITH UPPER IN MQ AND LOWER IN AC 1, 4
IS THE VALUE POSITIVE
NO
YES
POSI 6
COMPLEMENT THE LOWER WORD TO THE ACCUMULATOR
POSI 3
IS THE VALUE POSITIVE
NO
YES
COMPLEMENT THE LOWER WORD
COMPLEMENT THE UPPER WORD TO THE MQ
LEFT ADJUST THE LOWER WORD IN THE ACCUMULATOR
PLACE ALL ONE'S IN THE UPPER WORD

Figure A-13. Convert Floating Point Number to a Double Librascope Word

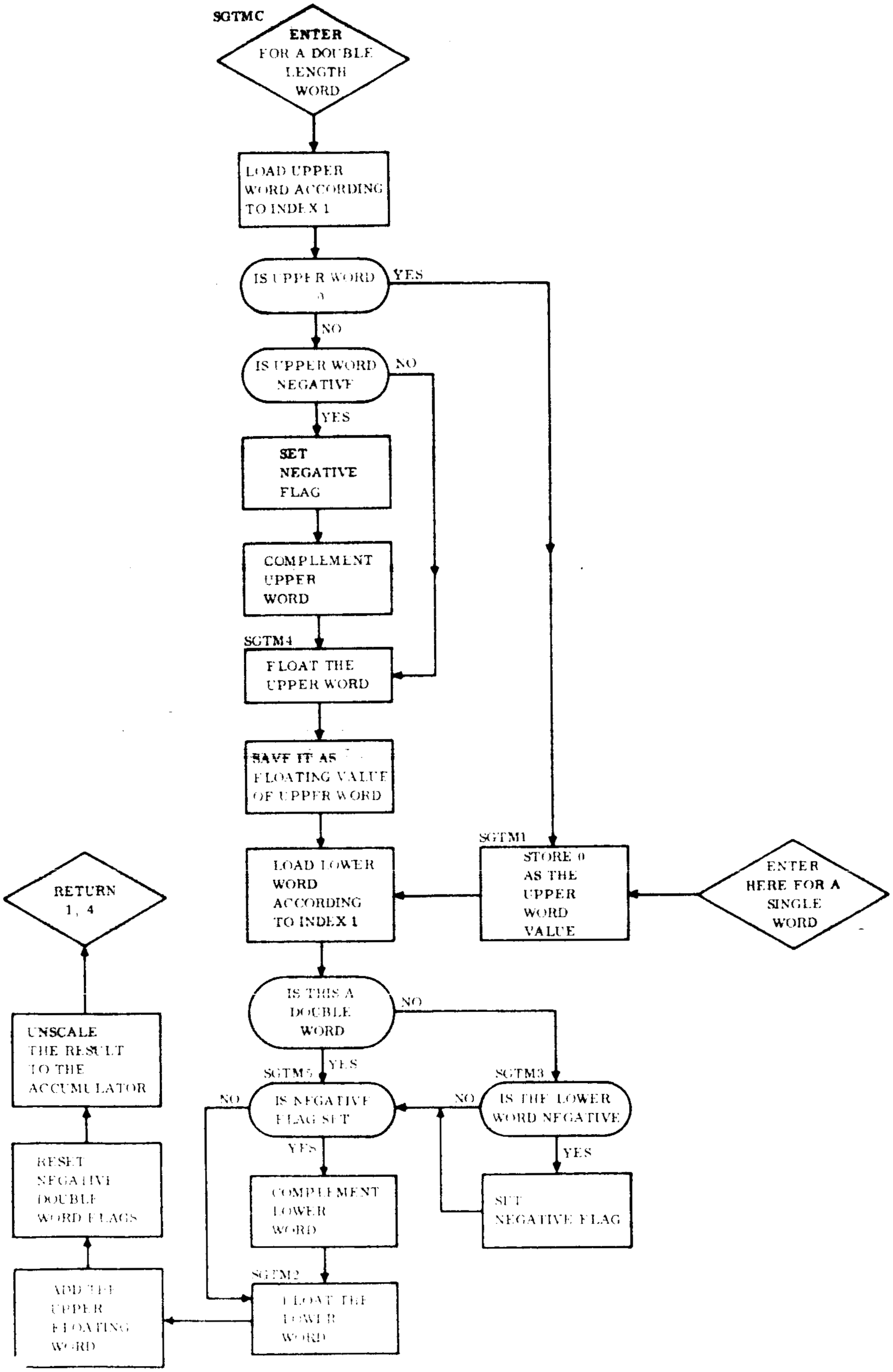

SGTMC
ENTER FOR A DOUBLE LENGTH WORD
LOAD UPPER WORD ACCORDING TO INDEX 1
IS UPPER WORD 0
YES
NO
IS UPPER WORD NEGATIVE
NO
YES
SET NEGATIVE FLAG
COMPLEMENT UPPER WORD
SGTM4
FLOAT THE UPPER WORD
SAVE IT AS FLOATING VALUE OF UPPER WORD
SGTM1
STORE 0 AS THE UPPER WORD VALUE
ENTER HERE FOR A SINGLE WORD
LOAD LOWER WORD ACCORDING TO INDEX 1
RETURN 1, 4
IS THIS A DOUBLE WORD
NO
YES
SGTM5
SGTM3
NO
IS NEGATIVE FLAG SET
NO
IS THE LOWER WORD NEGATIVE
YES
YES
UNSCALE THE RESULT TO THE ACCUMULATOR
RESET NEGATIVE DOUBLE WORD FLAGS
COMPLEMENT LOWER WORD
SET NEGATIVE FLAG
SGTM2
ADD THE UPPER FLOATING WORD
FLOAT THE LOWER WORD

Figure A-14. Convert Librascope Word (Double or Single) to Floating Point

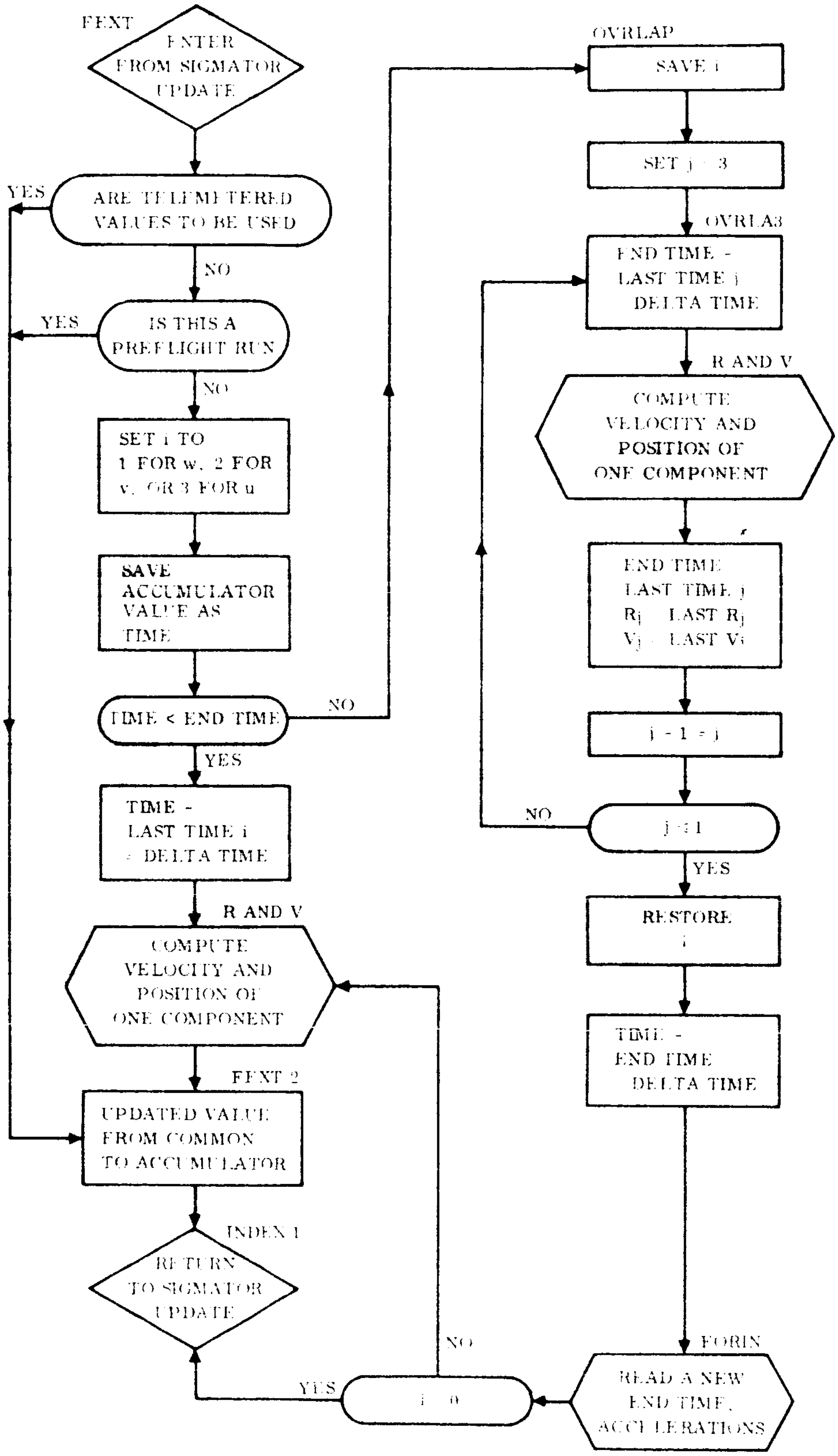

FEXT
ENTER FROM SIGMATOR UPDATE
YES
ARE TELEMETERED VALUES TO BE USED
NO
YES
IS THIS A PREFLIGHT RUN
NO
SET i TO 1 FOR w, 2 FOR v, OR 3 FOR u
SAVE ACCUMULATOR VALUE AS TIME
TIME < END TIME
NO
YES
TIME - LAST TIME i = DELTA TIME
R AND V
COMPUTE VELOCITY AND POSITION OF ONE COMPONENT
FEXT 2
UPDATED VALUE FROM COMMON TO ACCUMULATOR
INDEX 1
RETURN TO SIGMATOR UPDATE
YES
i = 0
NO
OVRLAP
SAVE i
SET j = 3
OVRLA3
END TIME - LAST TIME j = DELTA TIME
R AND V
COMPUTE VELOCITY AND POSITION OF ONE COMPONENT
END TIME = LAST TIME j
R_j = LAST R_j
V_j = LAST V_j
j - 1 = j
NO
j = 1
YES
RESTORE i
TIME - END TIME = DELTA TIME
FORIN
READ A NEW END TIME, ACCELERATIONS

Figure A-15. FEXT

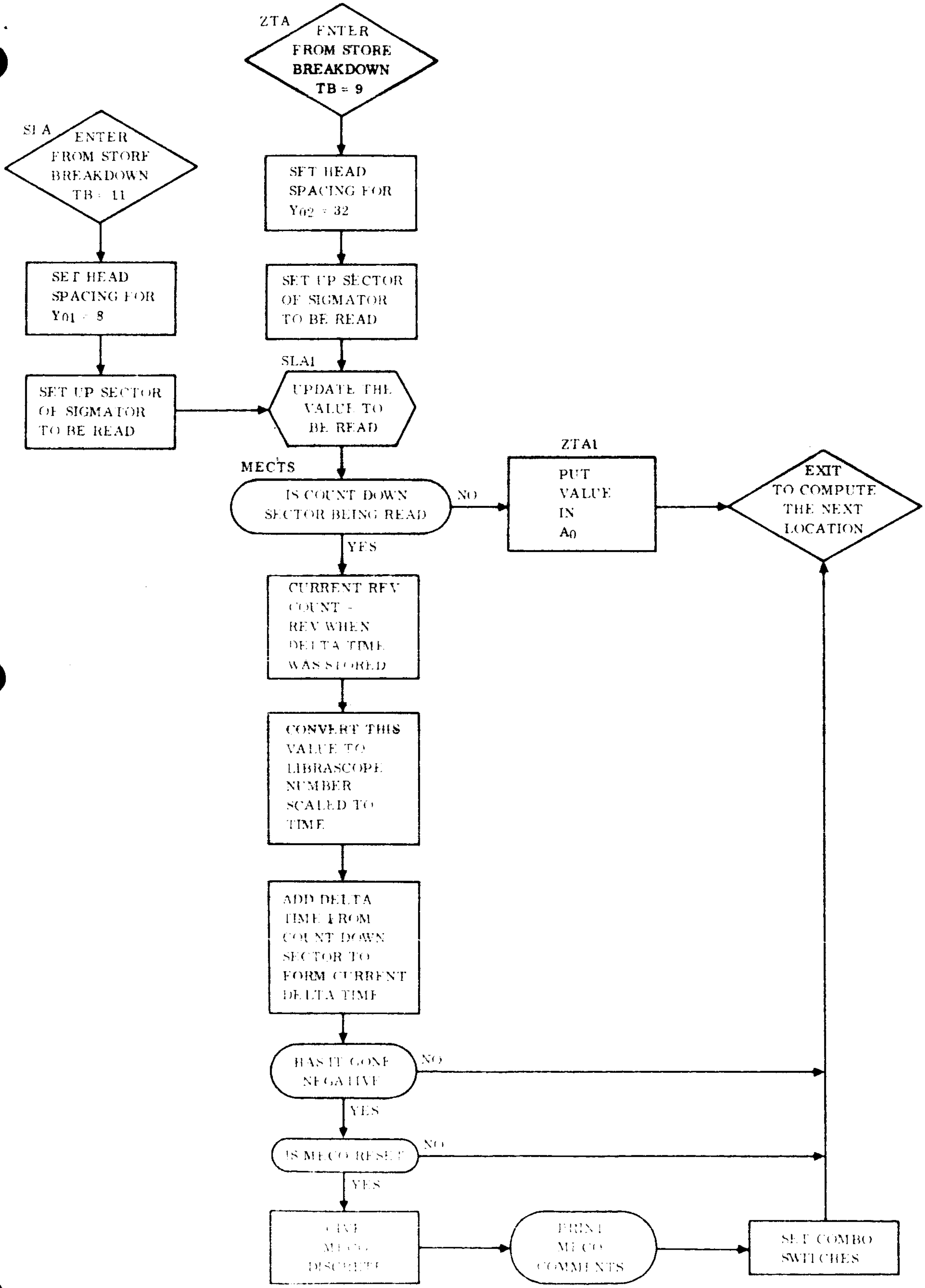

ZTA
ENTER FROM STORE BREAKDOWN TB = 9
SLA
ENTER FROM STORE BREAKDOWN TB = 11
SET HEAD SPACING FOR Y_{02} = 32
SET UP SECTOR OF SIGMATOR TO BE READ
SET HEAD SPACING FOR Y_{01} = 8
SET UP SECTOR OF SIGMATOR TO BE READ
SLA1
UPDATE THE VALUE TO BE READ
ZTA1
MECTS
IS COUNT DOWN SECTOR BEING READ
NO
PUT VALUE IN A_0
EXIT TO COMPUTE THE NEXT LOCATION
YES
CURRENT REV COUNT - REV WHEN DELTA TIME WAS STORED
CONVERT THIS VALUE TO LIBRASCOPE NUMBER SCALED TO TIME
ADD DELTA TIME FROM COUNT DOWN SECTOR TO FORM CURRENT DELTA TIME
HAS IT GONE NEGATIVE
NO
YES
IS MECO RESET
NO
YES
GIVE MECO DISCRETE
PRINT MECO COMMENTS
SET COMBO SWITCHES

Figure A-16. Sigmator to Accumulator and MECO Test

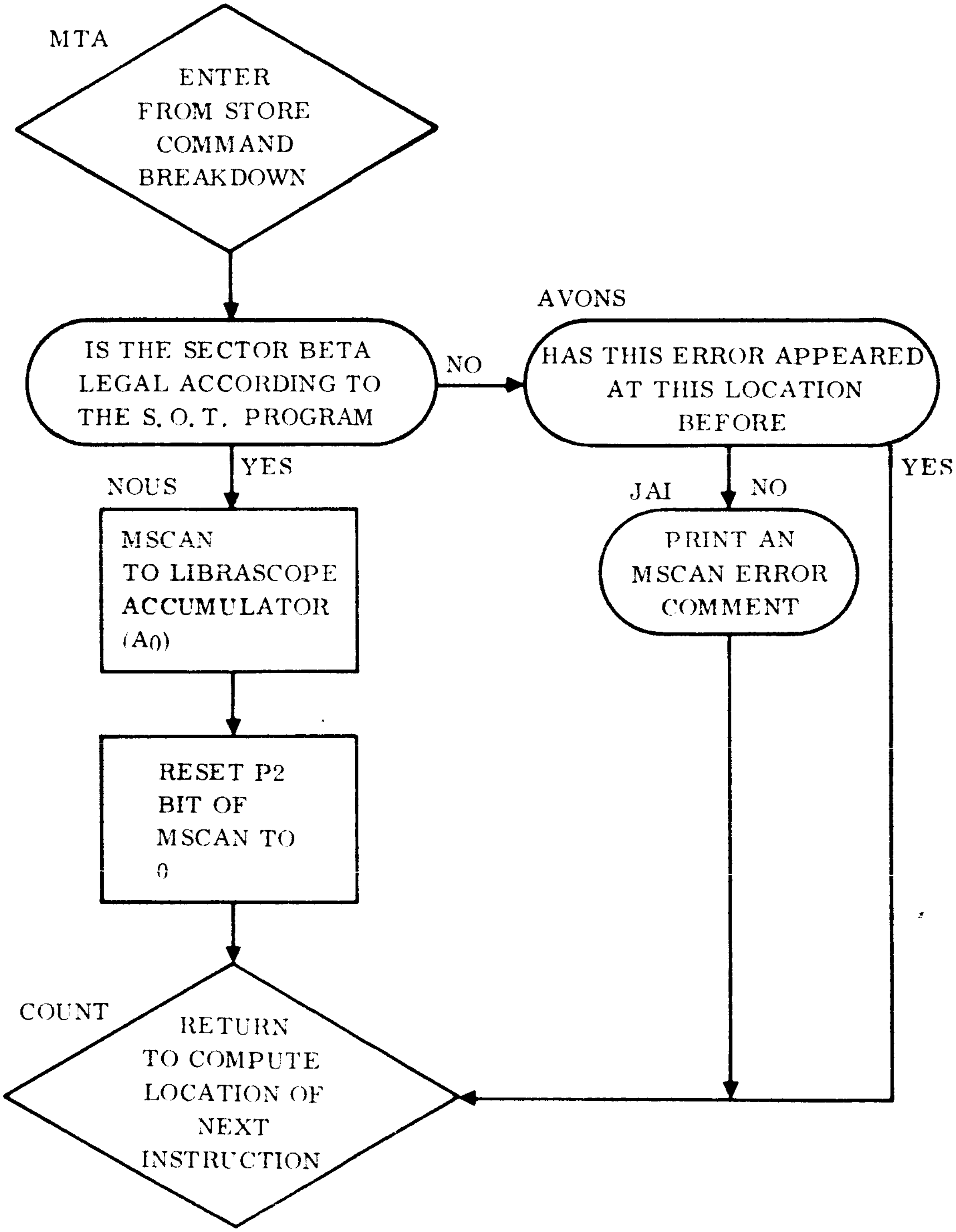

MTA
ENTER FROM STORE COMMAND BREAKDOWN
IS THE SECTOR BETA LEGAL ACCORDING TO THE S. O. T. PROGRAM
NO
AVONS
HAS THIS ERROR APPEARED AT THIS LOCATION BEFORE
YES
NOUS
YES
JAI
NO
MSCAN TO LIBRASCOPE ACCUMULATOR (A0)
PRINT AN MSCAN ERROR COMMENT
RESET P2 BIT OF MSCAN TO 0
COUNT
RETURN TO COMPUTE LOCATION OF NEXT INSTRUCTION

Figure A-17. MSCAN to Accumulator

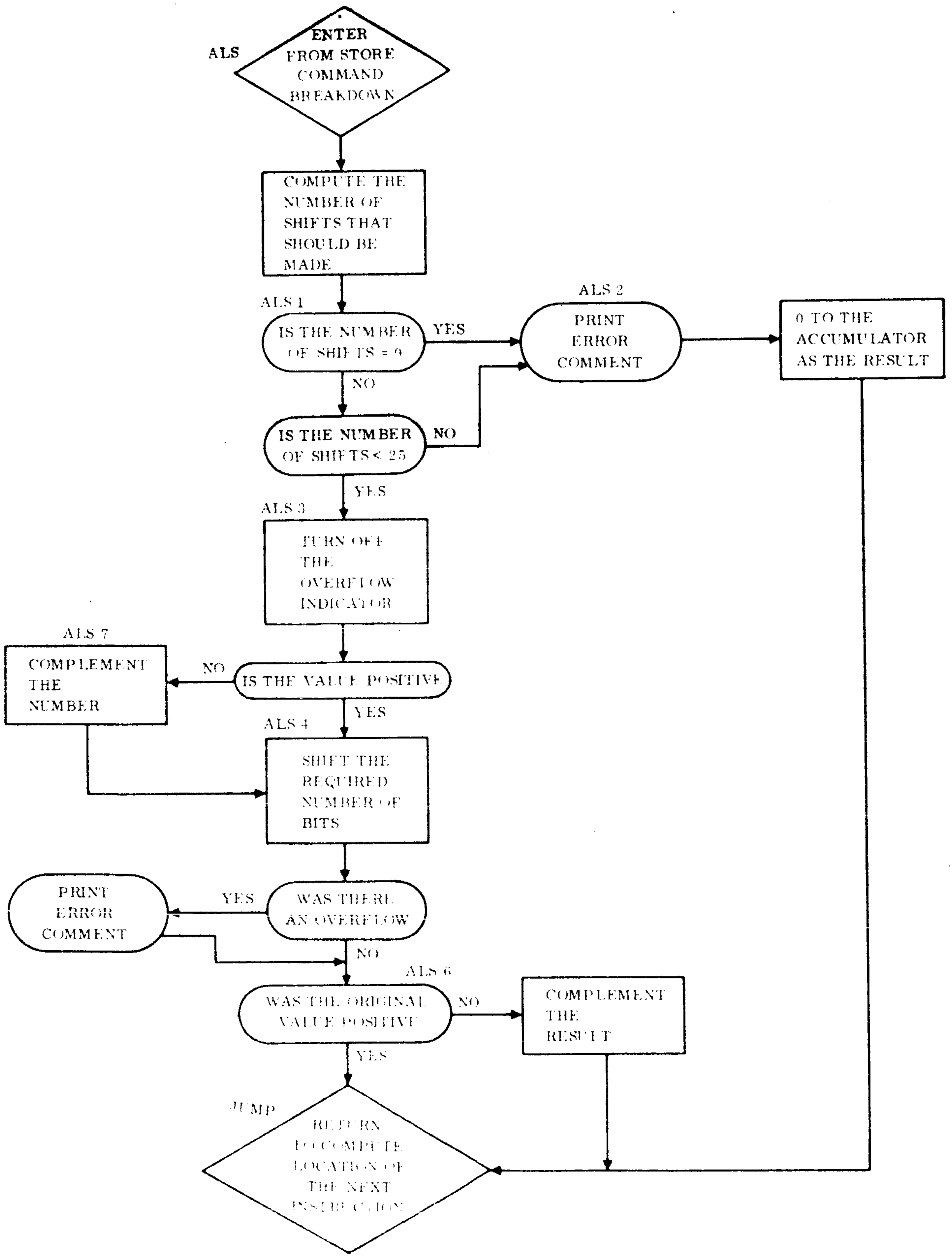

ALS
ENTER FROM STORE COMMAND BREAKDOWN
COMPUTE THE NUMBER OF SHIFTS THAT SHOULD BE MADE
ALS 1
IS THE NUMBER OF SHIFTS = 0
YES
ALS 2
PRINT ERROR COMMENT
0 TO THE ACCUMULATOR AS THE RESULT
NO
IS THE NUMBER OF SHIFTS < 25
NO
YES
ALS 3
TURN OFF THE OVERFLOW INDICATOR
ALS 7
COMPLEMENT THE NUMBER
NO
IS THE VALUE POSITIVE
YES
ALS 4
SHIFT THE REQUIRED NUMBER OF BITS
PRINT ERROR COMMENT
YES
WAS THERE AN OVERFLOW
NO
ALS 6
WAS THE ORIGINAL VALUE POSITIVE
NO
COMPLEMENT THE RESULT
YES
JUMP
RETURN TO COMPUTE LOCATION OF THE NEXT INSTRUCTION

Figure A-18. Accumulator Left Shift

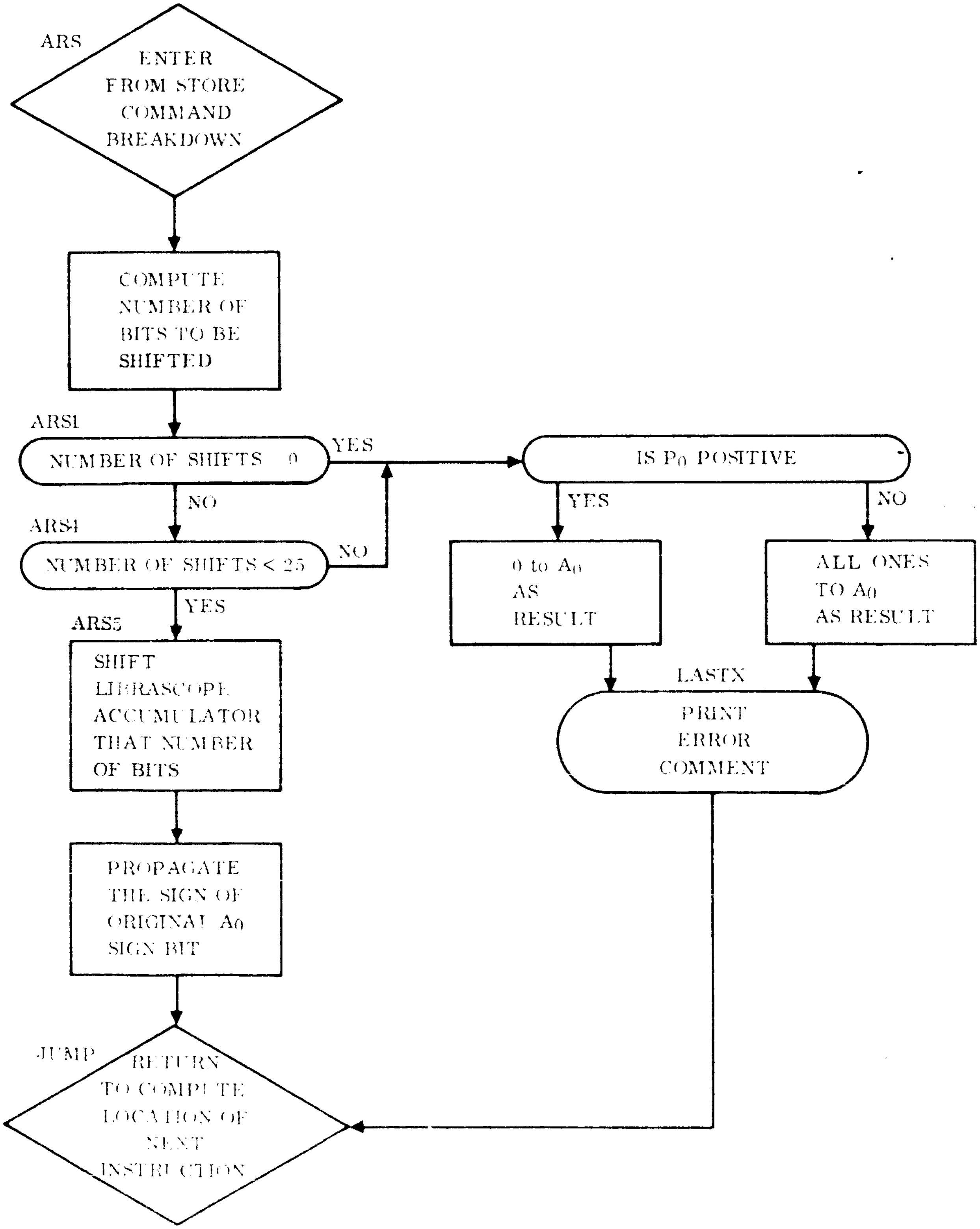

ARS
ENTER FROM STORE COMMAND BREAKDOWN
COMPUTE NUMBER OF BITS TO BE SHIFTED
ARS1
NUMBER OF SHIFTS 0
YES
NO
ARS4
NUMBER OF SHIFTS < 25
NO
YES
ARS5
SHIFT LIBRASCOPE ACCUMULATOR THAT NUMBER OF BITS
PROPAGATE THE SIGN OF ORIGINAL A0 SIGN BIT
JUMP
RETURN TO COMPUTE LOCATION OF NEXT INSTRUCTION
IS P0 POSITIVE
YES
NO
0 to A0 AS RESULT
ALL ONES TO A0 AS RESULT
LASTX
PRINT ERROR COMMENT

Figure A-19. Accumulator Right Shift

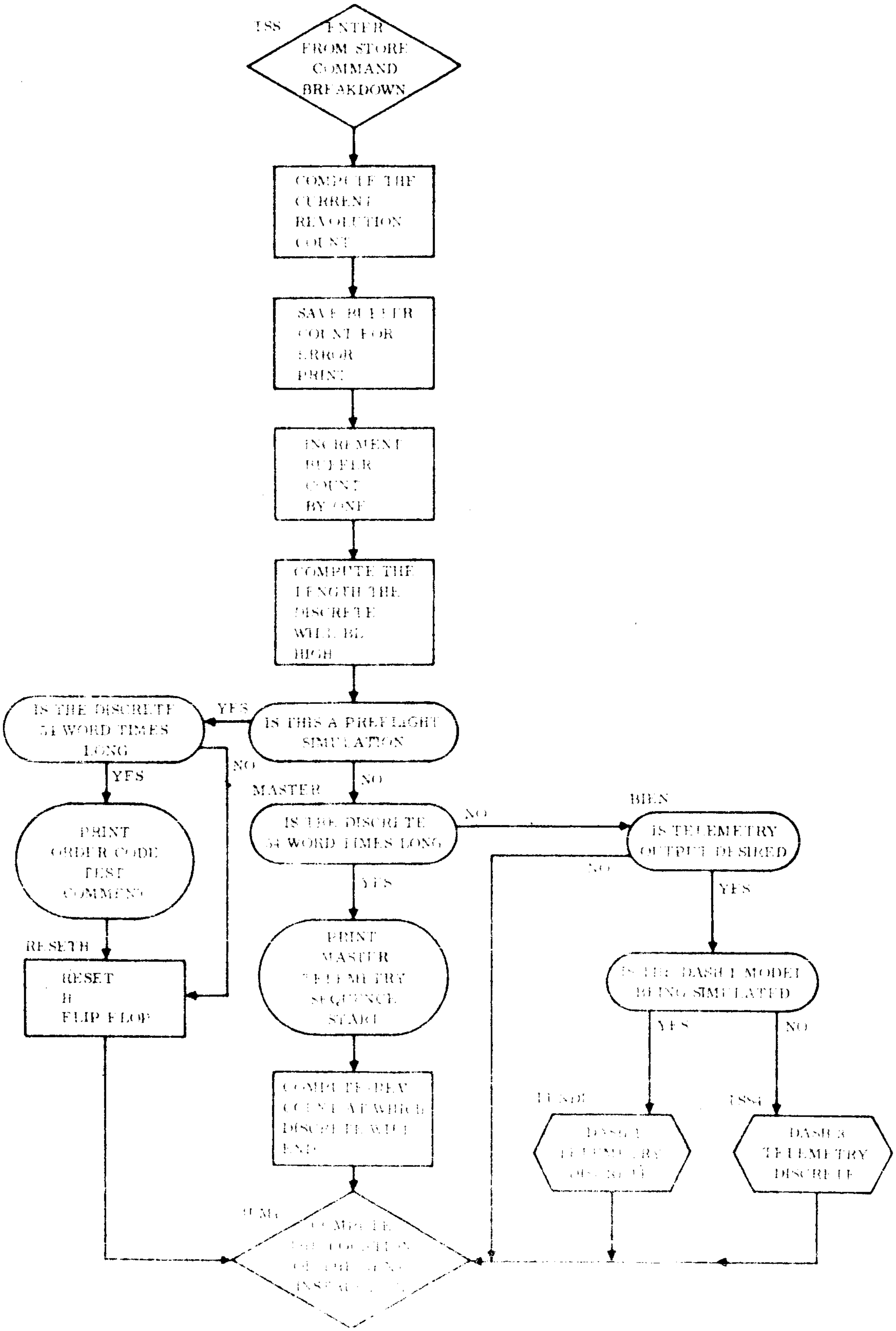

TSS
ENTER FROM STORE COMMAND BREAKDOWN
COMPUTE THE CURRENT REVOLUTION COUNT
SAVE BUFFER COUNT FOR ERROR PRINT
INCREMENT BUFFER COUNT BY ONE
COMPUTE THE LENGTH THE DISCRETE WILL BE HIGH
IS THIS A PREFLIGHT SIMULATION
YES
IS THE DISCRETE 54 WORD TIMES LONG
YES
NO
PRINT ORDER CODE TEST COMMENT
RESETH
RESET H FLIP FLOP
NO
MASTER
IS THE DISCRETE 54 WORD TIMES LONG
YES
PRINT MASTER TELEMETRY SEQUENCE START
COMPUTE REV COUNT AT WHICH DISCRETE WILL END
NO
BIEN
IS TELEMETRY OUTPUT DESIRED
NO
YES
IS THE DASH 1 MODEL BEING SIMULATED
YES
NO
DASH 1 TELEMETRY DISCRETE
DASH 3 TELEMETRY DISCRETE
COMPUTE THE LOCATION OF THE NEXT INSTRUCTION

Figure A-20. Telemetry Discretes

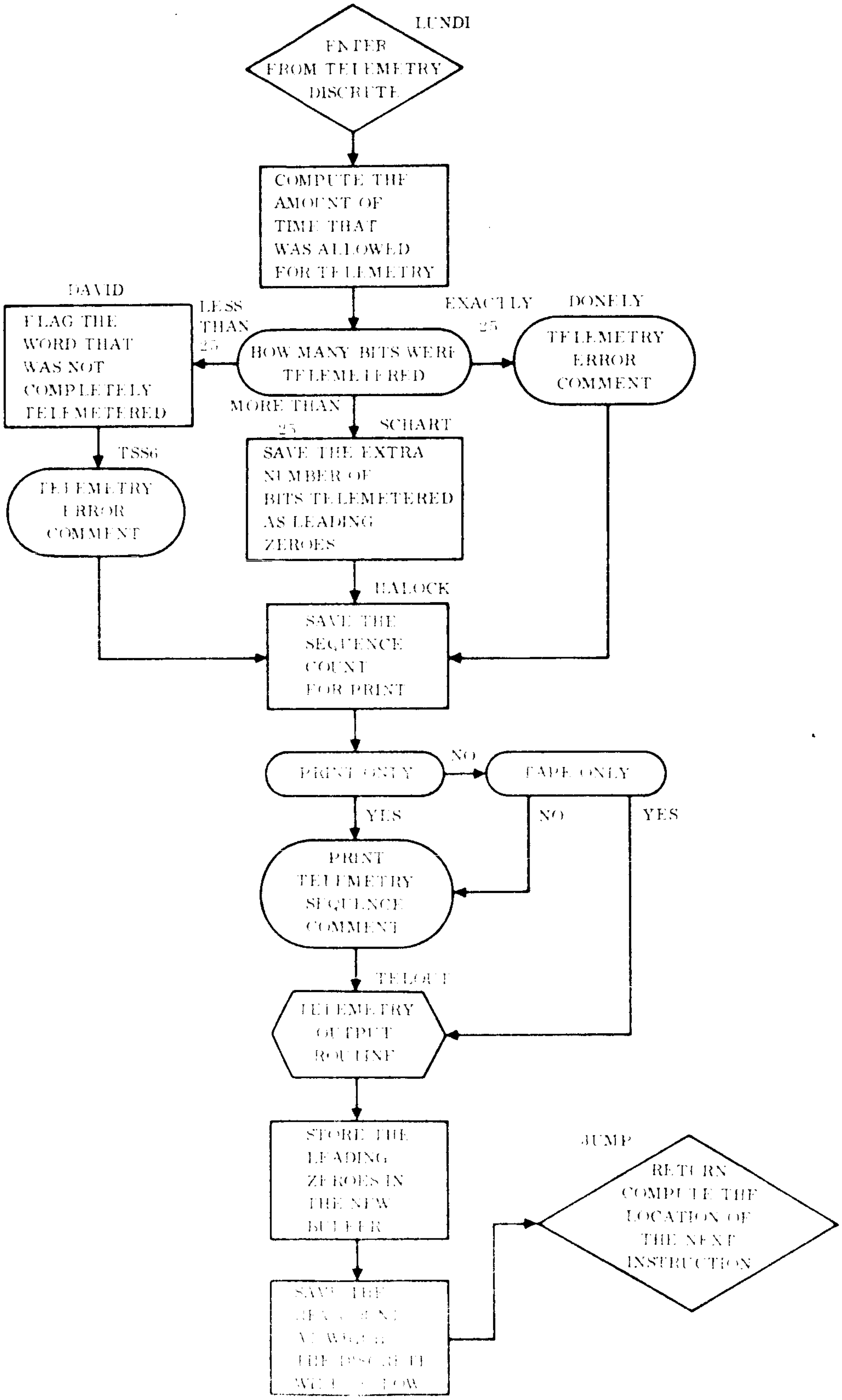

LUNDI
ENTER FROM TELEMETRY DISCRETE
COMPUTE THE AMOUNT OF TIME THAT WAS ALLOWED FOR TELEMETRY
HOW MANY BITS WERE TELEMETERED
LESS THAN 25
EXACTLY 25
MORE THAN 25
DAVID
FLAG THE WORD THAT WAS NOT COMPLETELY TELEMETERED
DONELY
TELEMETRY ERROR COMMENT
TSS6
TELEMETRY ERROR COMMENT
SCHART
SAVE THE EXTRA NUMBER OF BITS TELEMETERED AS LEADING ZEROES
HALOCK
SAVE THE SEQUENCE COUNT FOR PRINT
PRINT ONLY
NO
TAPE ONLY
YES
NO
YES
PRINT TELEMETRY SEQUENCE COMMENT
TELOUT
TELEMETRY OUTPUT ROUTINE
STORE THE LEADING ZEROES IN THE NEW BUFFER
SAVE THE SEQUENCE COUNT AT WHICH THE DISCRETE WILL FOLLOW
JUMP
RETURN COMPUTE THE LOCATION OF THE NEXT INSTRUCTION

Figure A-21. Model [illegible] Telemetry Sequence Start

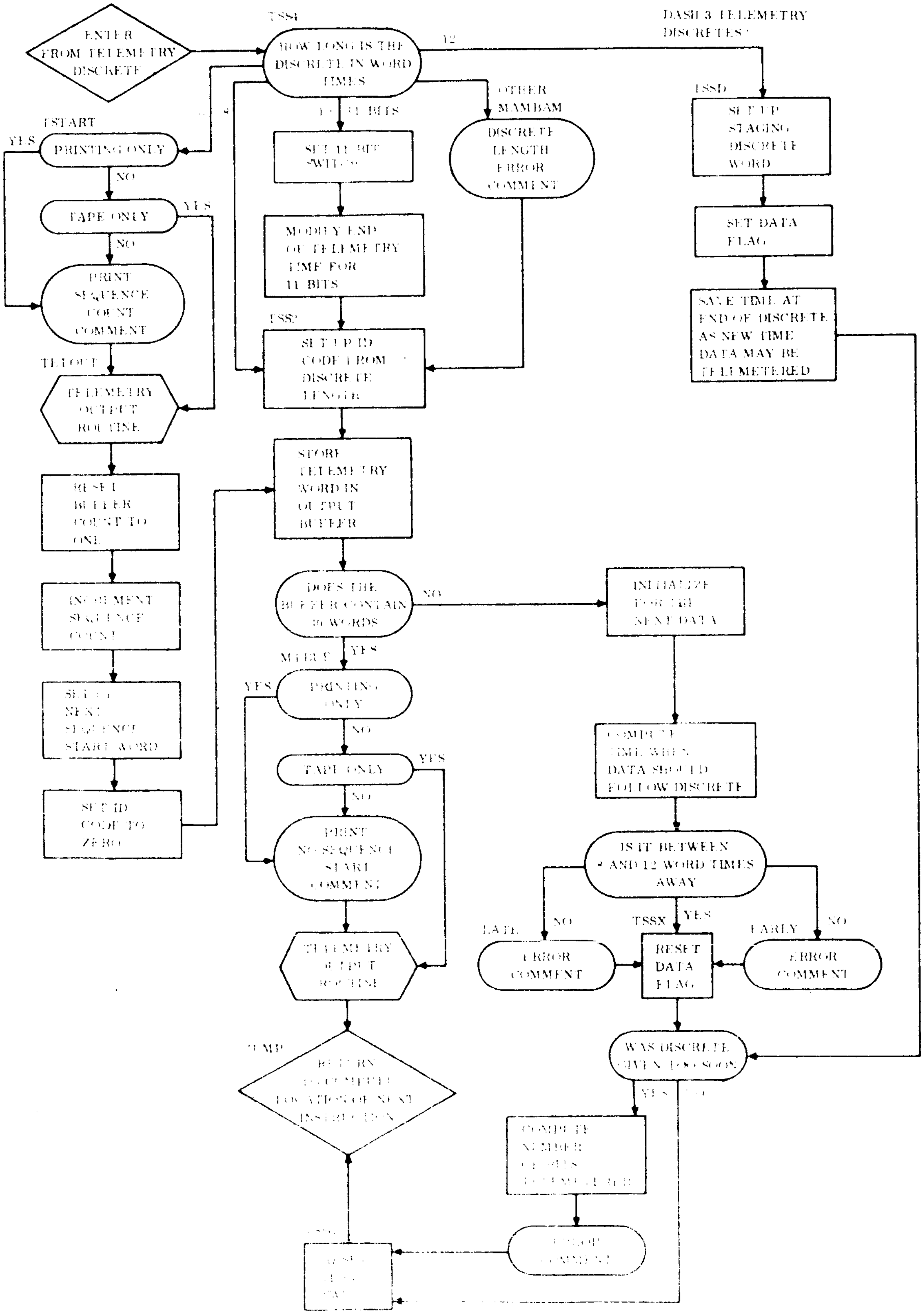

ENTER FROM TELEMETRY DISCRETE
TSS4
HOW LONG IS THE DISCRETE IN WORD TIMES
12
DASH 3 TELEMETRY DISCRETES
8
11 BITS
OTHER
MAMBAM
TSTART
YES
PRINTING ONLY
NO
SET 11 BIT SWITCH
DISCRETE LENGTH ERROR COMMENT
TSSD
SET UP STAGING DISCRETE WORD
TAPE ONLY
YES
NO
MODIFY END OF TELEMETRY TIME FOR 11 BITS
SET DATA FLAG
PRINT SEQUENCE COUNT COMMENT
TSS2
SAVE TIME AT END OF DISCRETE AS NEW TIME DATA MAY BE TELEMETERED
TELOUT
SET UP ID CODE FROM DISCRETE LENGTH
TELEMETRY OUTPUT ROUTINE
STORE TELEMETRY WORD IN OUTPUT BUFFER
RESET BUFFER COUNT TO ONE
DOES THE BUFFER CONTAIN 36 WORDS
NO
INITIALIZE FOR THE NEXT DATA
INCREMENT SEQUENCE COUNT
YES
MTEST
YES
PRINTING ONLY
NO
SET UP NEXT SEQUENCE START WORD
TAPE ONLY
YES
COMPUTE TIME WHEN DATA SHOULD FOLLOW DISCRETE
NO
SET ID CODE TO ZERO
PRINT NO SEQUENCE START COMMENT
IS IT BETWEEN 8 AND 12 WORD TIMES AWAY
LATE
NO
TSSX
YES
EARLY
NO
TELEMETRY OUTPUT ROUTINE
ERROR COMMENT
RESET DATA FLAG
ERROR COMMENT
WAS DISCRETE GIVEN TOO SOON
TEMP
RETURN TO COMPUTED LOCATION OF NEXT INSTRUCTION
YES
NO
COMPUTE NUMBER OF BITS TO TELEMETER
ERROR COMMENT

Figure A-22. Model-5 Telemetry Discretes

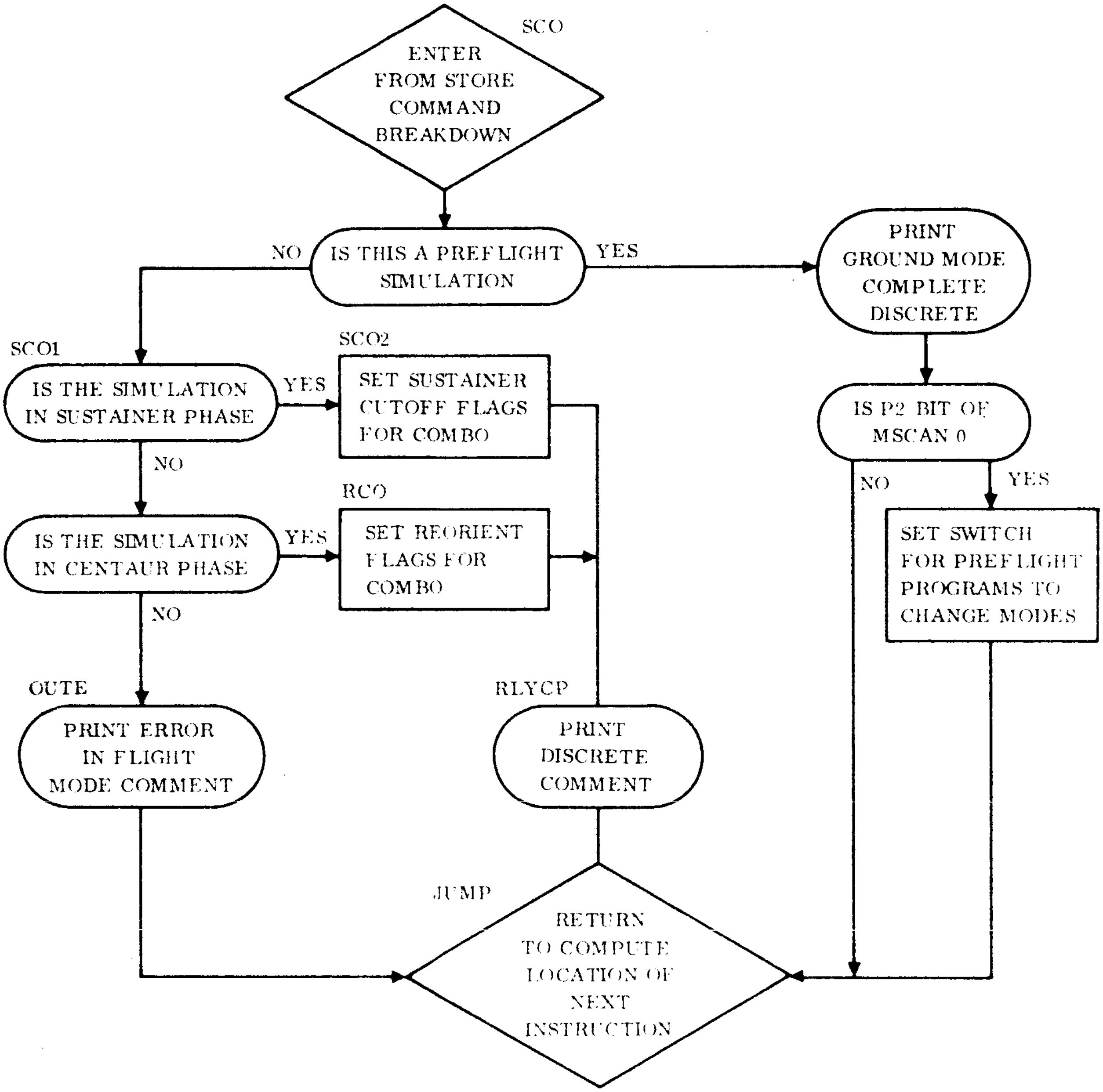

SCO
ENTER FROM STORE COMMAND BREAKDOWN
IS THIS A PREFLIGHT SIMULATION
NO
YES
PRINT GROUND MODE COMPLETE DISCRETE
SCO1
IS THE SIMULATION IN SUSTAINER PHASE
YES
SCO2
SET SUSTAINER CUTOFF FLAGS FOR COMBO
NO
RCO
IS THE SIMULATION IN CENTAUR PHASE
YES
SET REORIENT FLAGS FOR COMBO
NO
IS P2 BIT OF MSCAN 0
NO
YES
SET SWITCH FOR PREFLIGHT PROGRAMS TO CHANGE MODES
OUTE
PRINT ERROR IN FLIGHT MODE COMMENT
RLYCP
PRINT DISCRETE COMMENT
JUMP
RETURN TO COMPUTE LOCATION OF NEXT INSTRUCTION

Figure A-23. Sustainer, Reorient Discretes

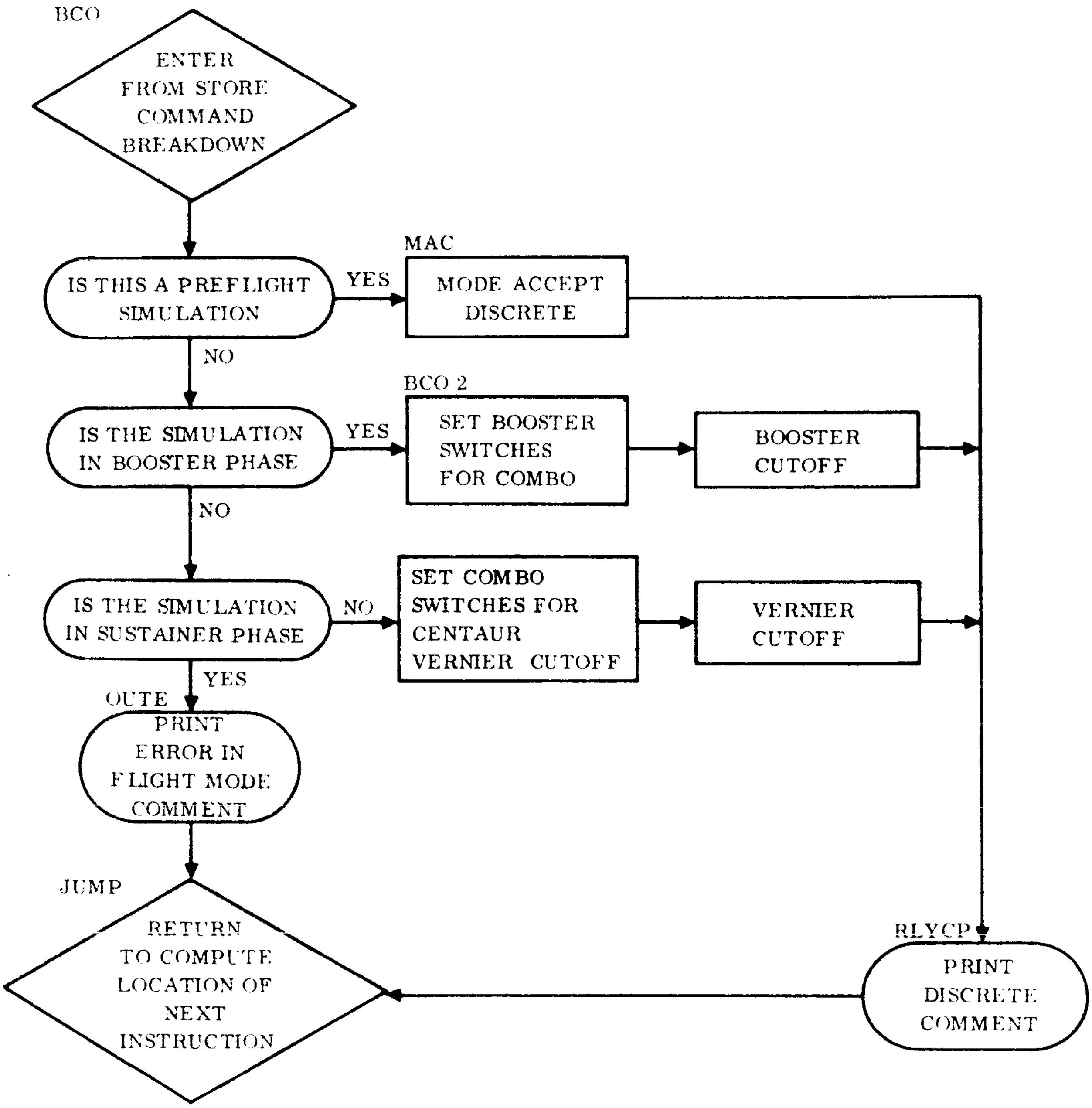

BCO
ENTER FROM STORE COMMAND BREAKDOWN
IS THIS A PREFLIGHT SIMULATION
YES
MAC
MODE ACCEPT DISCRETE
NO
IS THE SIMULATION IN BOOSTER PHASE
YES
BCO 2
SET BOOSTER SWITCHES FOR COMBO
BOOSTER CUTOFF
NO
IS THE SIMULATION IN SUSTAINER PHASE
NO
SET COMBO SWITCHES FOR CENTAUR VERNIER CUTOFF
VERNIER CUTOFF
YES
OUTE
PRINT ERROR IN FLIGHT MODE COMMENT
JUMP
RETURN TO COMPUTE LOCATION OF NEXT INSTRUCTION
RLYCP
PRINT DISCRETE COMMENT

Figure A-24. Booster, Mode Accept. Vernier Discretes

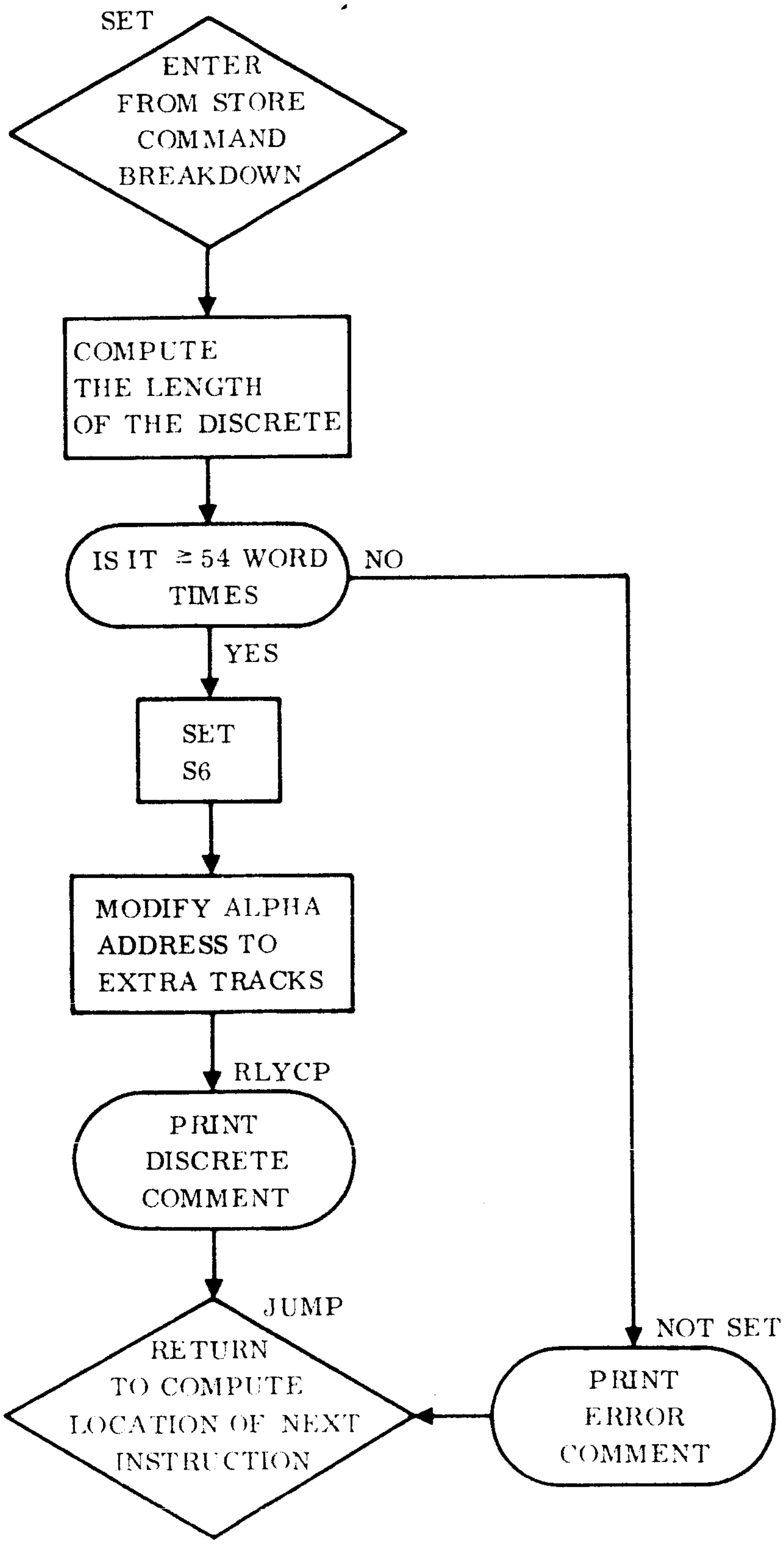

SET
ENTER FROM STORE COMMAND BREAKDOWN
COMPUTE THE LENGTH OF THE DISCRETE
IS IT ≥ 54 WORD TIMES
NO
YES
SET S6
MODIFY ALPHA ADDRESS TO EXTRA TRACKS
RLYCP
PRINT DISCRETE COMMENT
JUMP
RETURN TO COMPUTE LOCATION OF NEXT INSTRUCTION
NOT SET
PRINT ERROR COMMENT

Figure A-25. Set S6

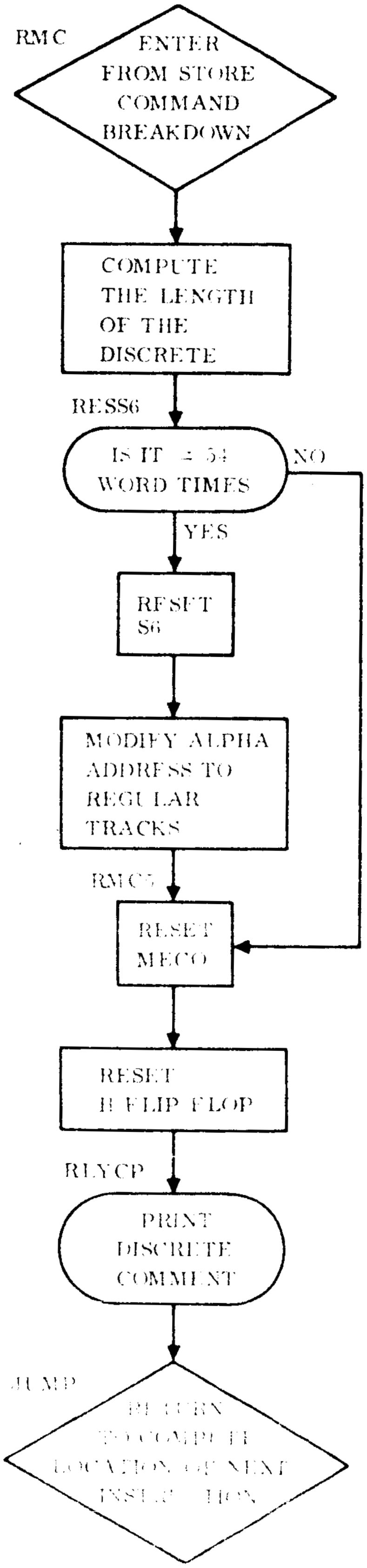

RMC
ENTER FROM STORE COMMAND BREAKDOWN
COMPUTE THE LENGTH OF THE DISCRETE
RESS6
IS IT ≥ 54 WORD TIMES
NO
YES
RESET S6
MODIFY ALPHA ADDRESS TO REGULAR TRACKS
RMC5
RESET MECO
RESET H FLIP FLOP
RLYCP
PRINT DISCRETE COMMENT
JUMP
RETURN TO COMPUTE LOCATION OF NEXT INSTRUCTION

Figure A-26. Reset S6, MECO, H Flip-Flop

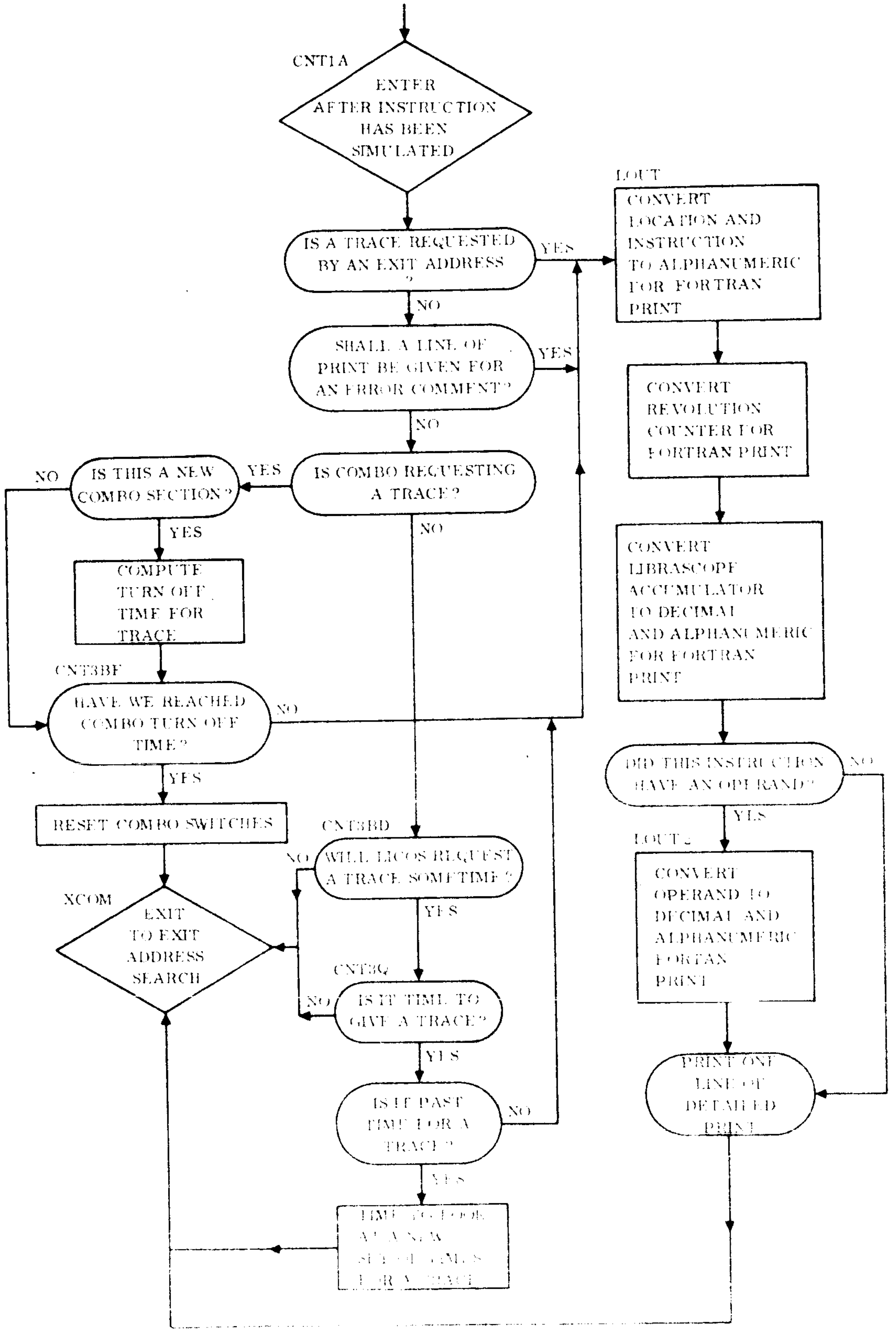

CNT1A
ENTER AFTER INSTRUCTION HAS BEEN SIMULATED
IS A TRACE REQUESTED BY AN EXIT ADDRESS ?
YES
NO
SHALL A LINE OF PRINT BE GIVEN FOR AN ERROR COMMENT?
YES
NO
IS COMBO REQUESTING A TRACE?
YES
NO
IS THIS A NEW COMBO SECTION?
NO
YES
COMPUTE TURN OFF TIME FOR TRACE
CNT3BF
HAVE WE REACHED COMBO TURN OFF TIME?
NO
YES
RESET COMBO SWITCHES
XCOM
EXIT TO EXIT ADDRESS SEARCH
CNT3BD
WILL LICOS REQUEST A TRACE SOMETIME?
NO
YES
CNT3Q
IS IT TIME TO GIVE A TRACE?
NO
YES
IS IT PAST TIME FOR A TRACE?
NO
YES
TIME TO LOOK AT A NEW SET OF TIMES FOR A TRACE
LOUT
CONVERT LOCATION AND INSTRUCTION TO ALPHANUMERIC FOR FORTRAN PRINT
CONVERT REVOLUTION COUNTER FOR FORTRAN PRINT
CONVERT LIBRASCOPE ACCUMULATOR TO DECIMAL AND ALPHANUMERIC FOR FORTRAN PRINT
DID THIS INSTRUCTION HAVE AN OPERAND?
NO
YES
LOUT2
CONVERT OPERAND TO DECIMAL AND ALPHANUMERIC FORTAN PRINT
PRINT ONE LINE OF DETAILED PRINT

Figure A-27. Detailed Print Test

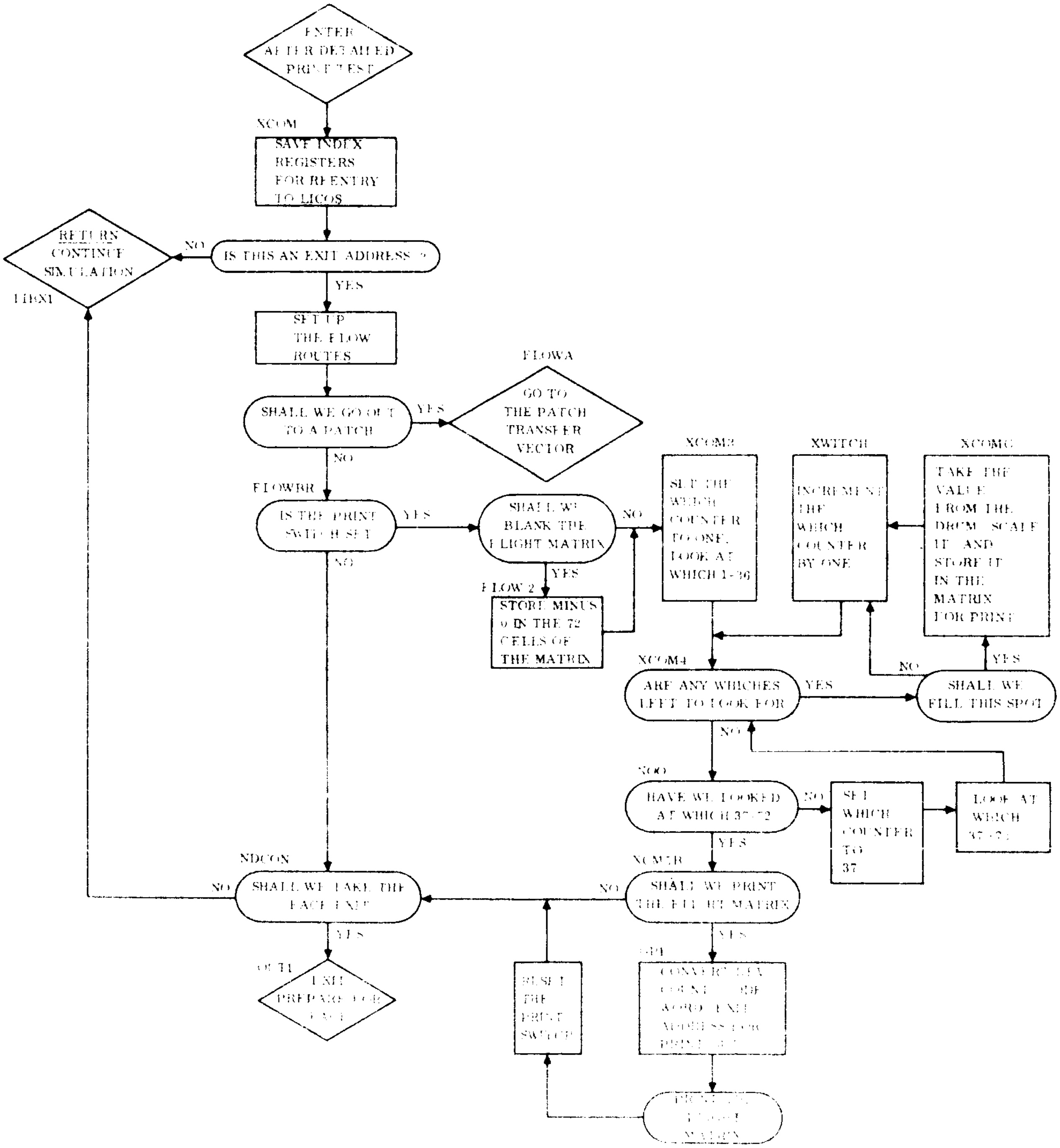


Figure A-28. Exit Address

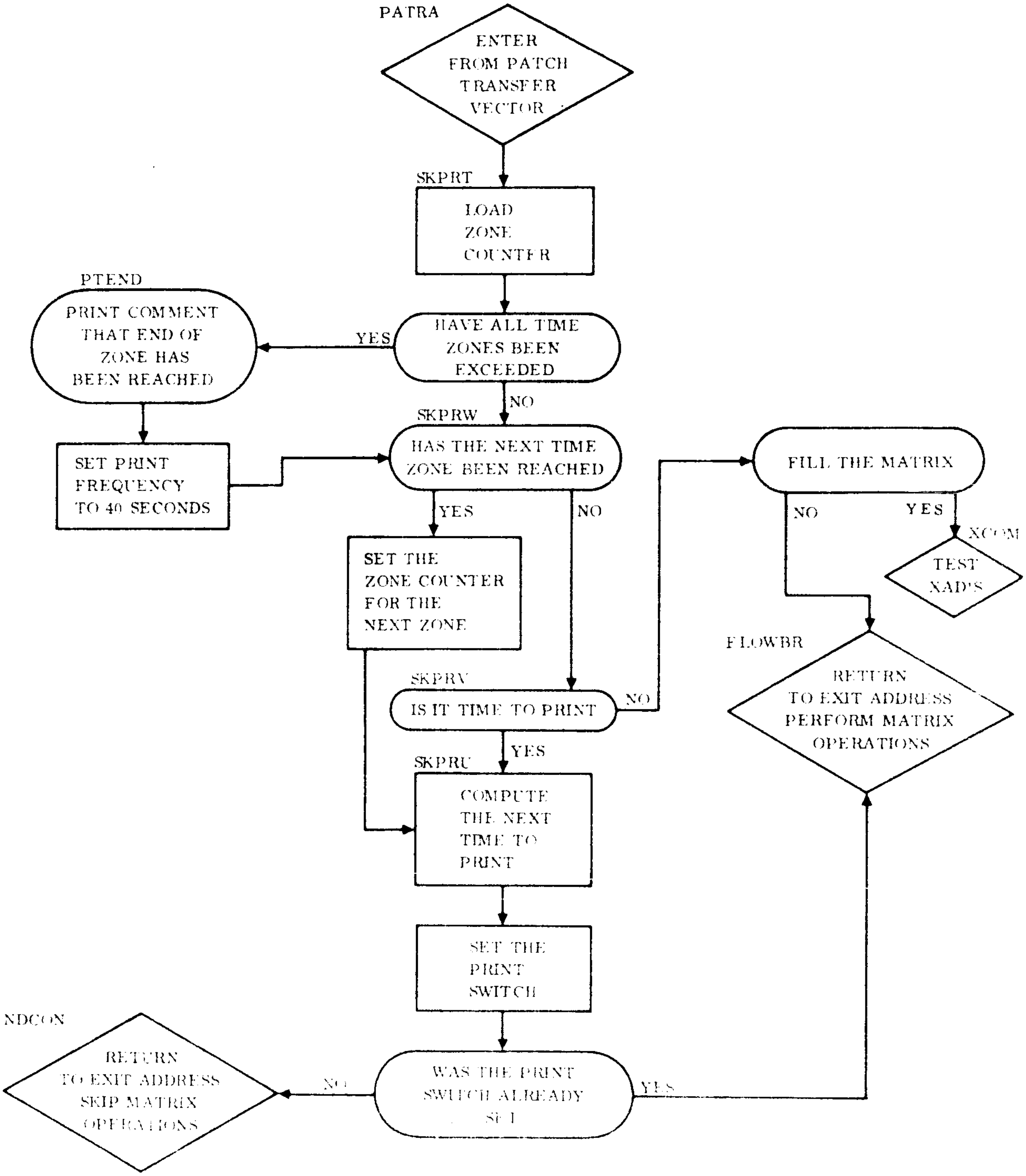

PATRA
ENTER FROM PATCH TRANSFER VECTOR
SKPRT
LOAD ZONE COUNTER
PTEND
PRINT COMMENT THAT END OF ZONE HAS BEEN REACHED
YES
HAVE ALL TIME ZONES BEEN EXCEEDED
NO
SET PRINT FREQUENCY TO 40 SECONDS
SKPRW
HAS THE NEXT TIME ZONE BEEN REACHED
YES
NO
SET THE ZONE COUNTER FOR THE NEXT ZONE
FILL THE MATRIX
NO
YES
XCOM
TEST XAD'S
FLOWBR
RETURN TO EXIT ADDRESS PERFORM MATRIX OPERATIONS
SKPRV
IS IT TIME TO PRINT
NO
YES
SKPRU
COMPUTE THE NEXT TIME TO PRINT
SET THE PRINT SWITCH
NDCON
RETURN TO EXIT ADDRESS SKIP MATRIX OPERATIONS
NO
WAS THE PRINT SWITCH ALREADY SET
YES

Figure A-29. Print Matrix on Time Basis, Patch 5

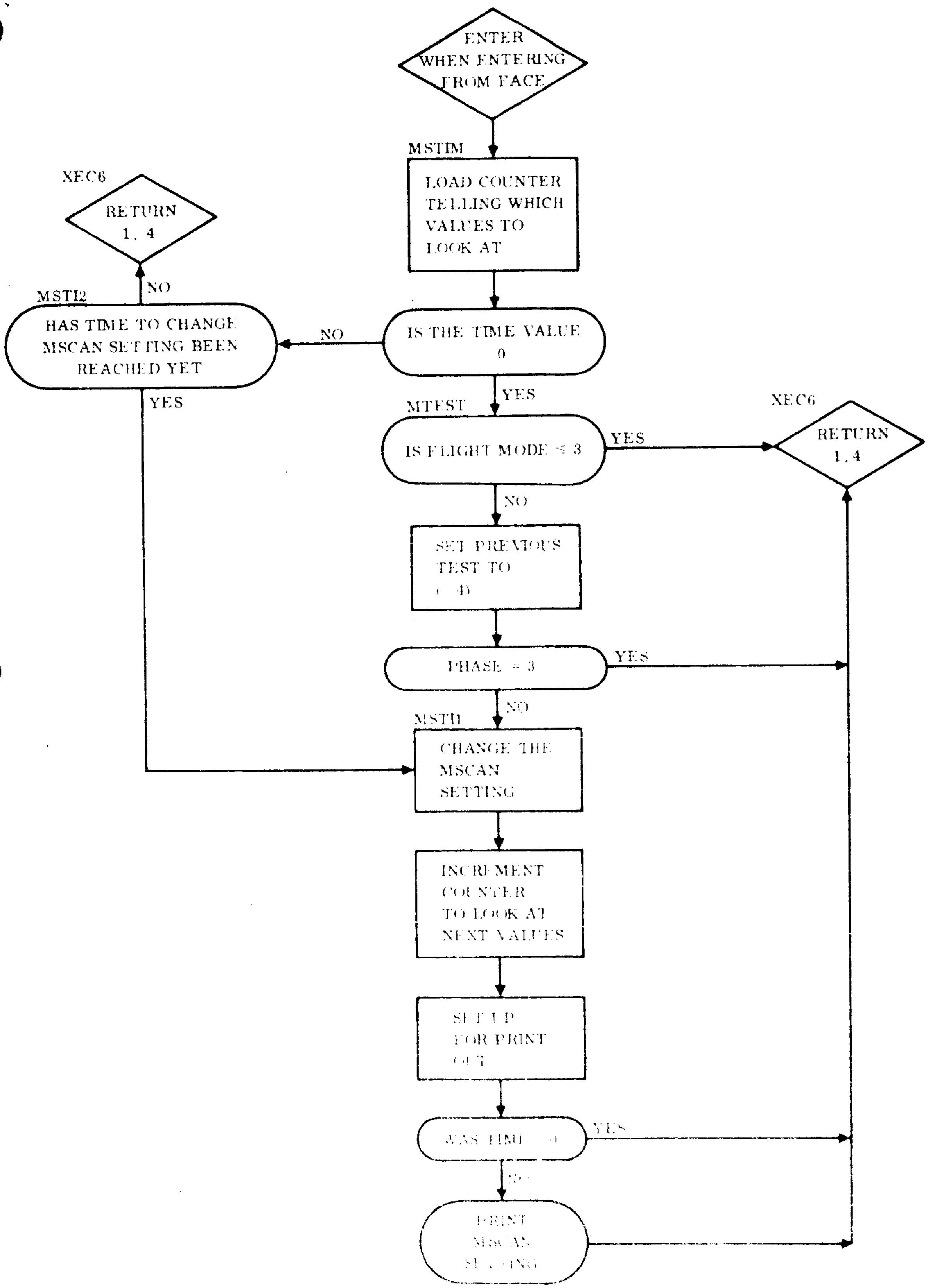


Figure A-30. Change MSCAN Setting

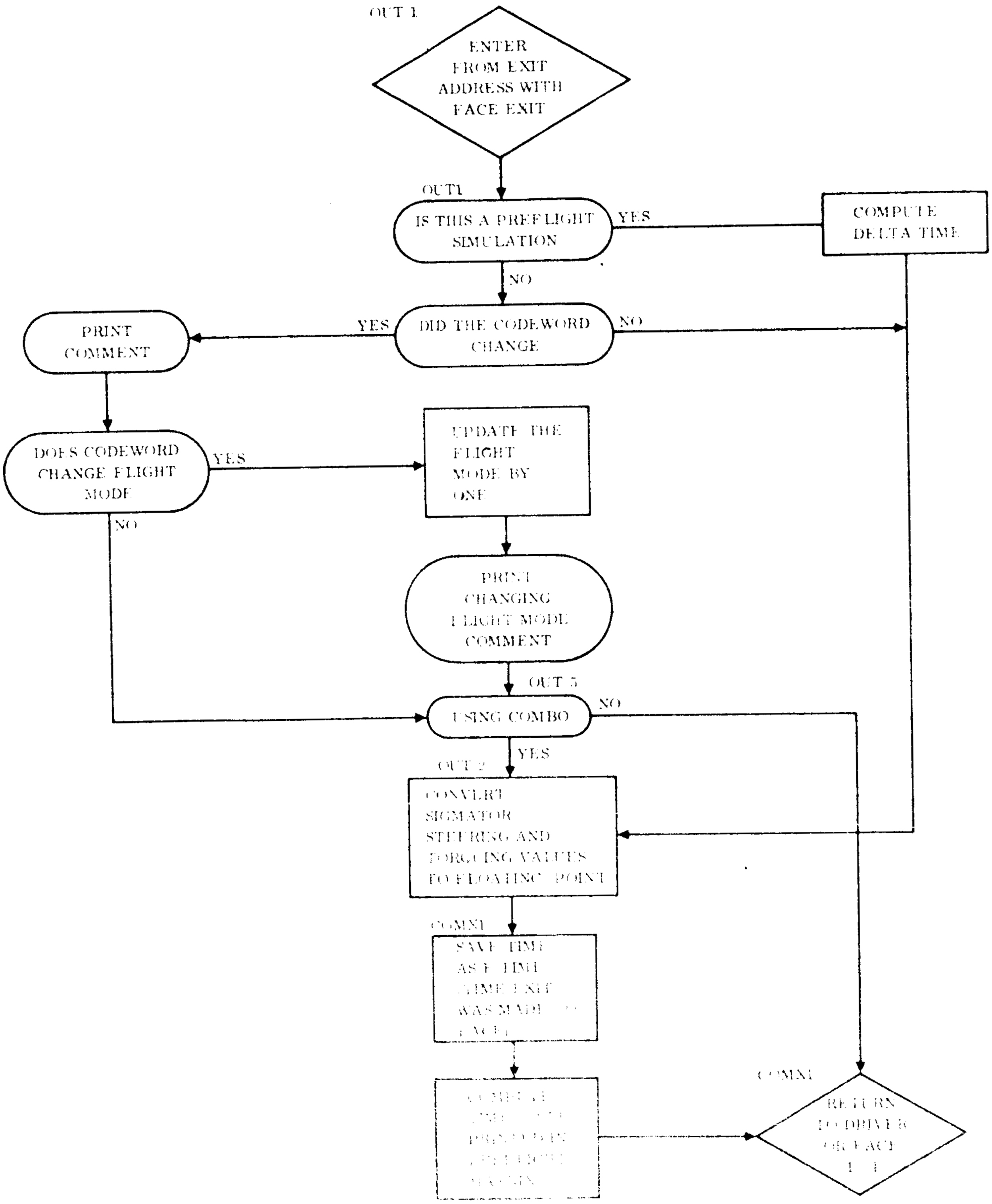

OUT 1
ENTER FROM EXIT ADDRESS WITH FACE EXIT
OUT1
IS THIS A PREFLIGHT SIMULATION
YES
COMPUTE DELTA TIME
NO
DID THE CODEWORD CHANGE
YES
NO
PRINT COMMENT
DOES CODEWORD CHANGE FLIGHT MODE
YES
NO
UPDATE THE FLIGHT MODE BY ONE
PRINT CHANGING FLIGHT MODE COMMENT
OUT 5
USING COMBO
NO
YES
OUT 2
CONVERT SIGMATOR STEERING AND FORCING VALUES TO FLOATING POINT
COMXI
SAVE TIME AS THE TIME EXIT WAS MADE TO FACE
COMPUTE
PRINTOUT IN
MARGIN
COMXI
RETURN TO DRIVER OR FACE
1 4

Figure A-31. Prepare for Face

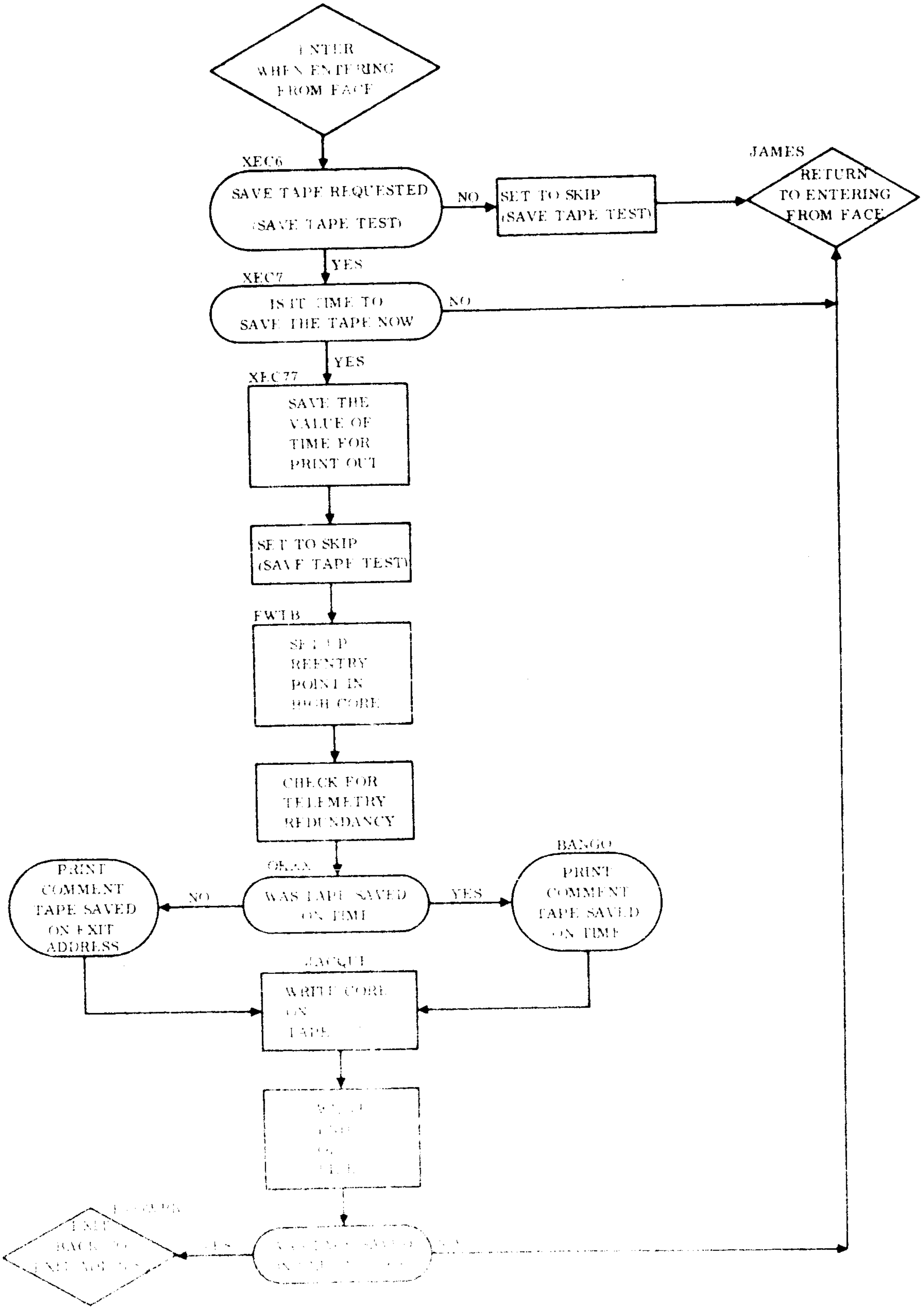


Figure A-[illegible]. Save a Tape

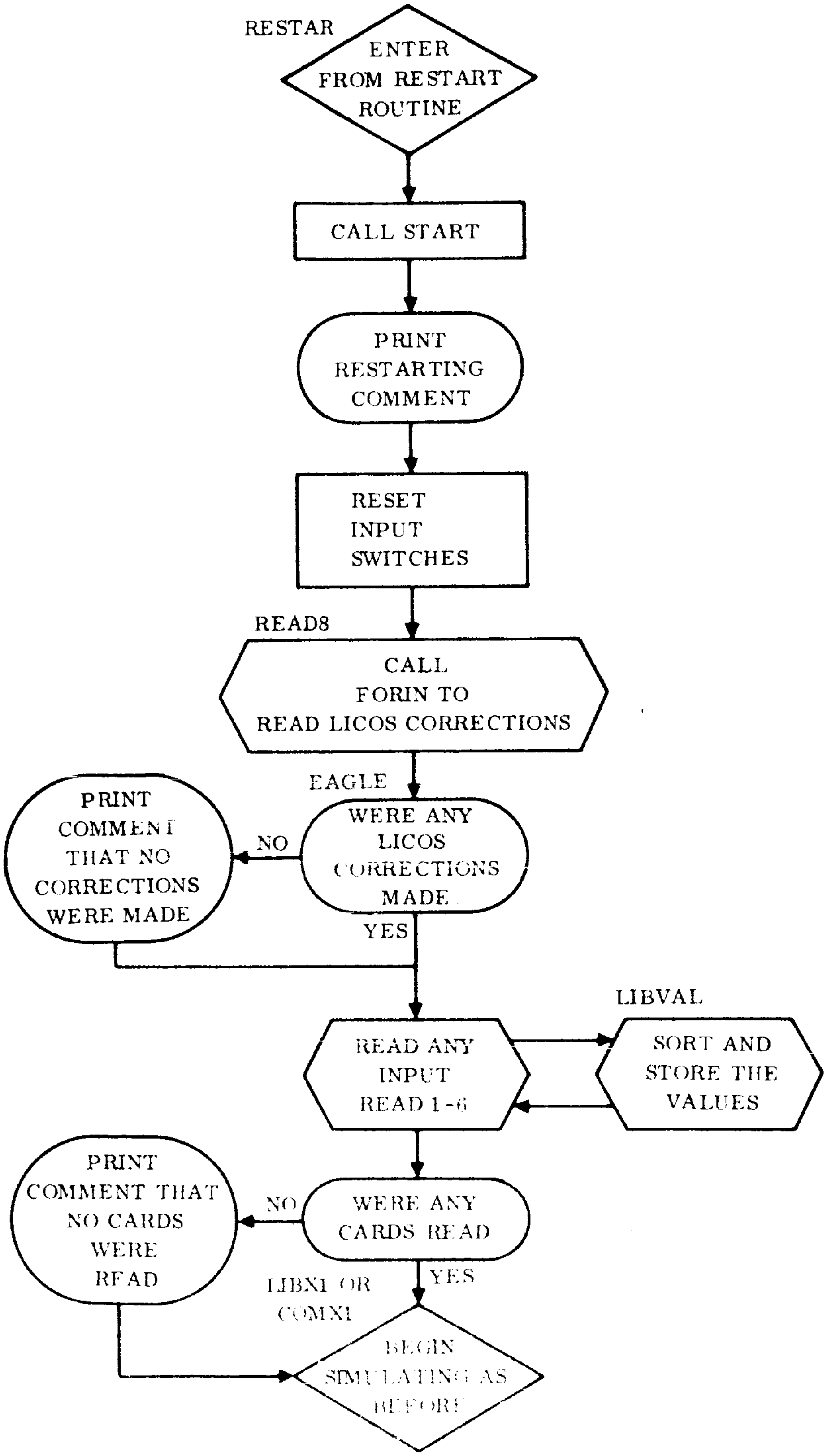

RESTAR
ENTER FROM RESTART ROUTINE
CALL START
PRINT RESTARTING COMMENT
RESET INPUT SWITCHES
READ8
CALL FORIN TO READ LICOS CORRECTIONS
EAGLE
WERE ANY LICOS CORRECTIONS MADE
NO
PRINT COMMENT THAT NO CORRECTIONS WERE MADE
YES
LIBVAL
READ ANY INPUT READ 1-6
SORT AND STORE THE VALUES
WERE ANY CARDS READ
NO
PRINT COMMENT THAT NO CARDS WERE READ
YES
LIBXI OR COMXI
BEGIN SIMULATING AS BEFORE

Figure A-53. Restart from Tape

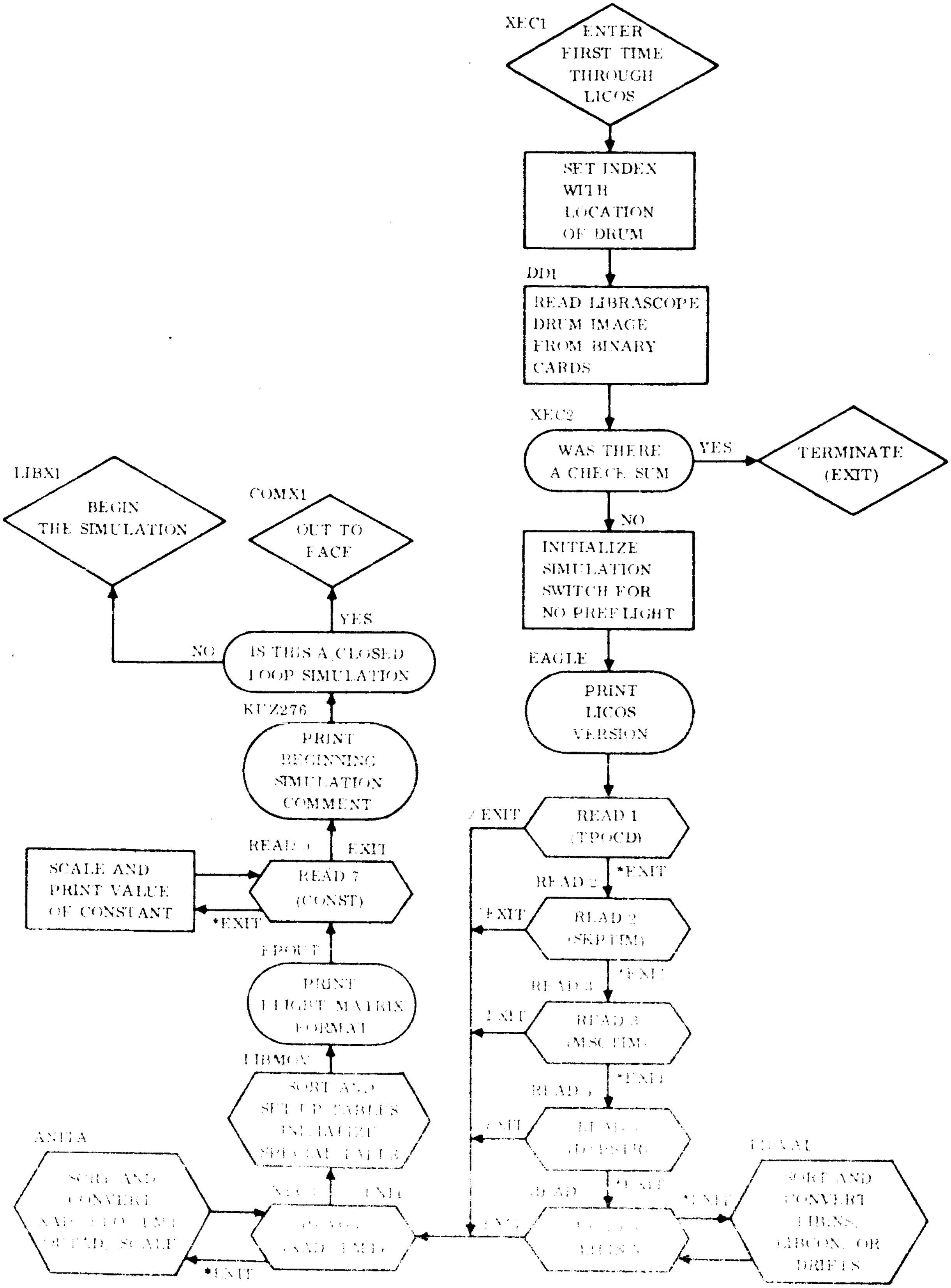

XEC1
ENTER FIRST TIME THROUGH LICOS
SET INDEX WITH LOCATION OF DRUM
DD1
READ LIBRASCOPE DRUM IMAGE FROM BINARY CARDS
XEC2
WAS THERE A CHECK SUM
YES
TERMINATE (EXIT)
NO
INITIALIZE SIMULATION SWITCH FOR NO PREFLIGHT
EAGLE
PRINT LICOS VERSION
EXIT
READ 1 (TPOCD)
*EXIT
READ 2
EXIT
READ 2 (SKPTIM)
*EXIT
READ 3
EXIT
READ 3 (MSCTIM)
*EXIT
READ 5
EXIT
*EXIT
READ
*EXIT
EXIT
SORT AND CONVERT LIBNS, LIBCON, OR DRIFTS
LIBX1
BEGIN THE SIMULATION
COMX1
OUT TO FACE
YES
NO
IS THIS A CLOSED LOOP SIMULATION
KUZ276
PRINT BEGINNING SIMULATION COMMENT
READ
EXIT
SCALE AND PRINT VALUE OF CONSTANT
READ 7 (CONST)
*EXIT
FPOUT
PRINT FLIGHT MATRIX FORMAT
LIBMOV
SORT AND SET UP TABLES INITIALIZE SPECIAL TABLE
ANITA
SORT AND CONVERT
EXIT
*EXIT

Figure A-[illegible]. Initializ[illegible]

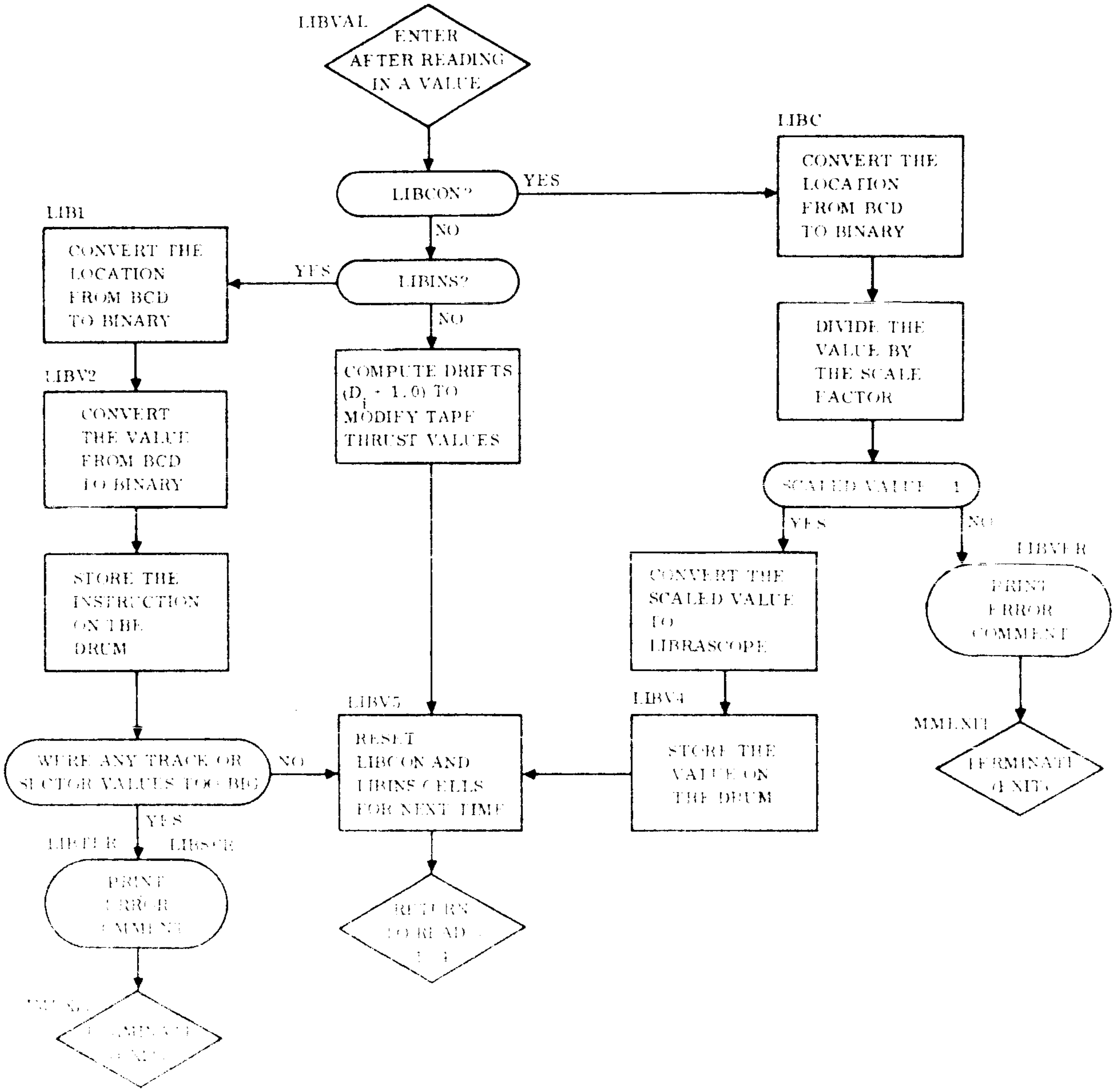

LIBVAL
ENTER AFTER READING IN A VALUE
LIBCON?
YES
LIBC
CONVERT THE LOCATION FROM BCD TO BINARY
DIVIDE THE VALUE BY THE SCALE FACTOR
SCALED VALUE 1
YES
NO
NO
LIB1
CONVERT THE LOCATION FROM BCD TO BINARY
YES
LIBINS?
NO
COMPUTE DRIFTS $(D_i + 1.0)$ TO MODIFY TAPE THRUST VALUES
LIBV2
CONVERT THE VALUE FROM BCD TO BINARY
STORE THE INSTRUCTION ON THE DRUM
CONVERT THE SCALED VALUE TO LIBRASCOPE
LIBVER
PRINT ERROR COMMENT
LIBV5
LIBV4
MMLXIT
TERMINATE (EXIT)
WERE ANY TRACK OR SECTOR VALUES TOO BIG
NO
RESET LIBCON AND LIBINS CELLS FOR NEXT TIME
STORE THE VALUE ON THE DRUM
YES
LIBTER
LIBSCR
PRINT ERROR COMMENT
RETURN TO READ
1 4
MMLXIT
TERMINATE (EXIT)

Figure A-35. Sort Read 5

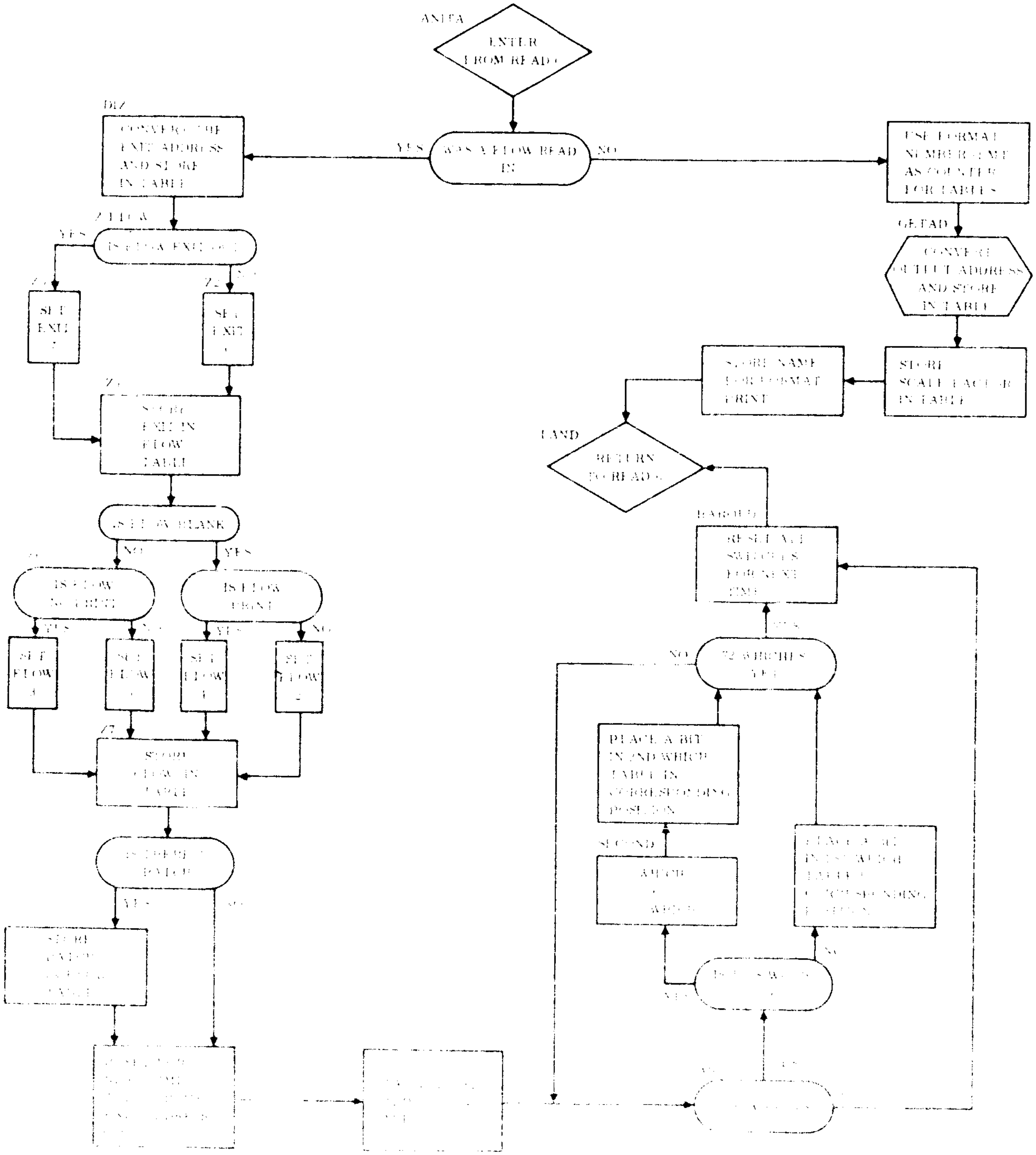

ANITA
ENTER FROM READ 6
WAS A FLOW READ IN
YES
NO
DIZ
CONVERT THE EXIT ADDRESS AND STORE IN TABLE
ZFLOA
IS FLOW A EXIT OUT
YES
NO
Z2
SET EXIT
SET EXIT
STORE EXIT IN FLOW TABLE
IS FLOW BLANK
NO
YES
IS FLOW IN CHECK
IS FLOW PRINT
YES
NO
YES
NO
SET FLOW 3
SET FLOW
SET FLOW 1
SET FLOW 2
Z7
STORE FLOW IN TABLE
PATCH
YES
NO
STORE
USE FORMAT NUMBER AS COUNTER FOR TABLES
GETAD
CONVERT OUTPUT ADDRESS AND STORE IN TABLE
STORE SCALE FACTOR IN TABLE
STORE NAME FOR FORMAT PRINT
LAND
RETURN TO READ 6
RESET ALL SWITCHES FOR NEXT TIME
YES
NO
72 WHICHES YET
PLACE A BIT IN 2ND WHICH TABLE IN CORRESPONDING POSITION
SECOND
WHICH
PLACE A BIT IN 1ST WHICH TABLE CORRESPONDING POSITION
NO
YES

Figure A-36. Sort Read 6

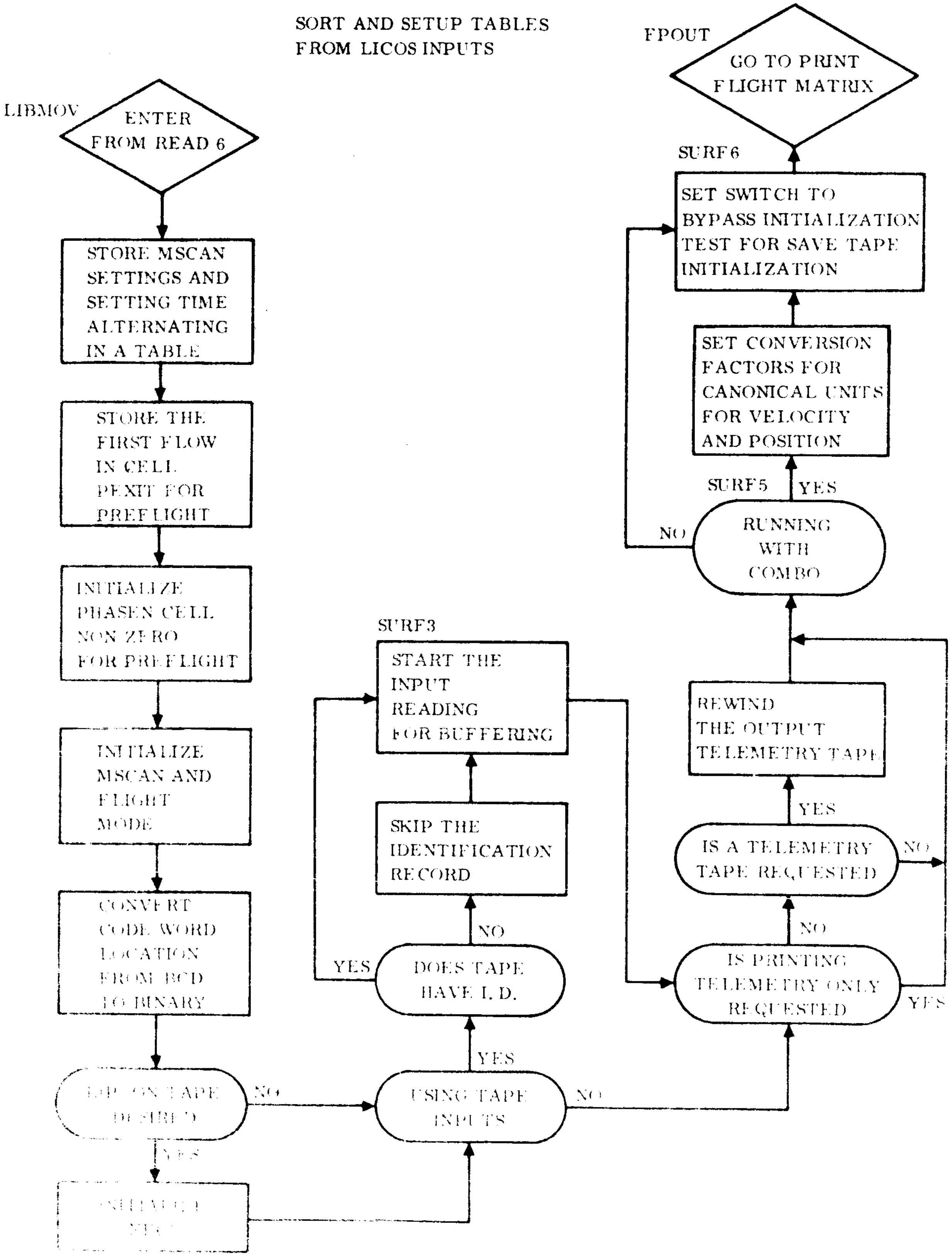

SORT AND SETUP TABLES
FROM LICOS INPUTS
LIBMOV
ENTER FROM READ 6
STORE MSCAN SETTINGS AND SETTING TIME ALTERNATING IN A TABLE
STORE THE FIRST FLOW IN CELL PEXIT FOR PREFLIGHT
INITIALIZE PHASEN CELL NON ZERO FOR PREFLIGHT
INITIALIZE MSCAN AND FLIGHT MODE
CONVERT CODE WORD LOCATION FROM BCD TO BINARY
I.D. ON TAPE DESIRED
NO
YES
USING TAPE INPUTS
YES
NO
DOES TAPE HAVE I. D.
YES
NO
SKIP THE IDENTIFICATION RECORD
SURF3
START THE INPUT READING FOR BUFFERING
IS PRINTING TELEMETRY ONLY REQUESTED
NO
YES
IS A TELEMETRY TAPE REQUESTED
YES
NO
REWIND THE OUTPUT TELEMETRY TAPE
SURF5
RUNNING WITH COMBO
YES
NO
SET CONVERSION FACTORS FOR CANONICAL UNITS FOR VELOCITY AND POSITION
SURF6
SET SWITCH TO BYPASS INITIALIZATION TEST FOR SAVE TAPE INITIALIZATION
FPOUT
GO TO PRINT FLIGHT MATRIX

Figure A-57. Sort Reads 2, 3, 4, Other Initialization

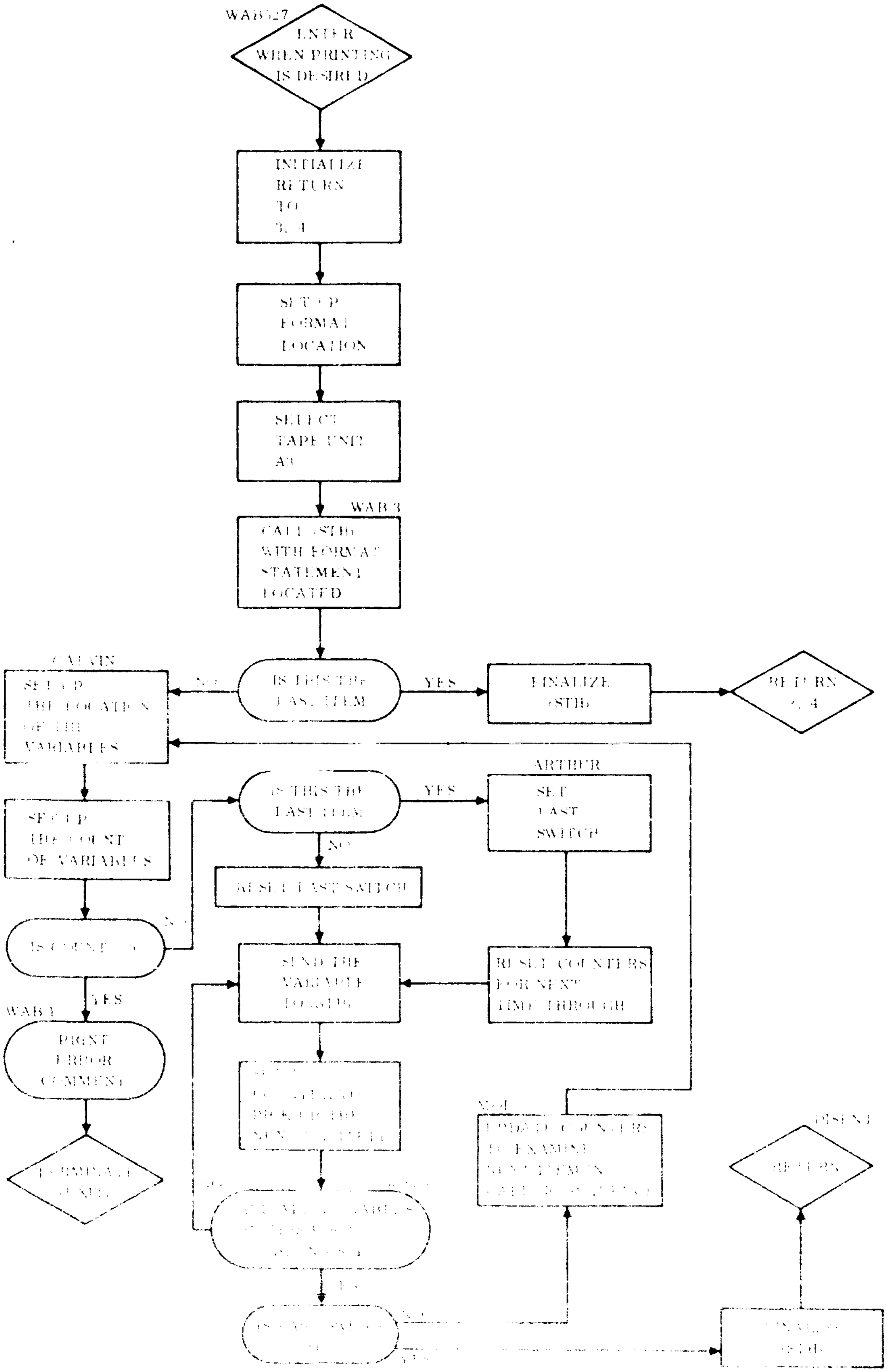


Figure A-38. Fortran Output with (STH)

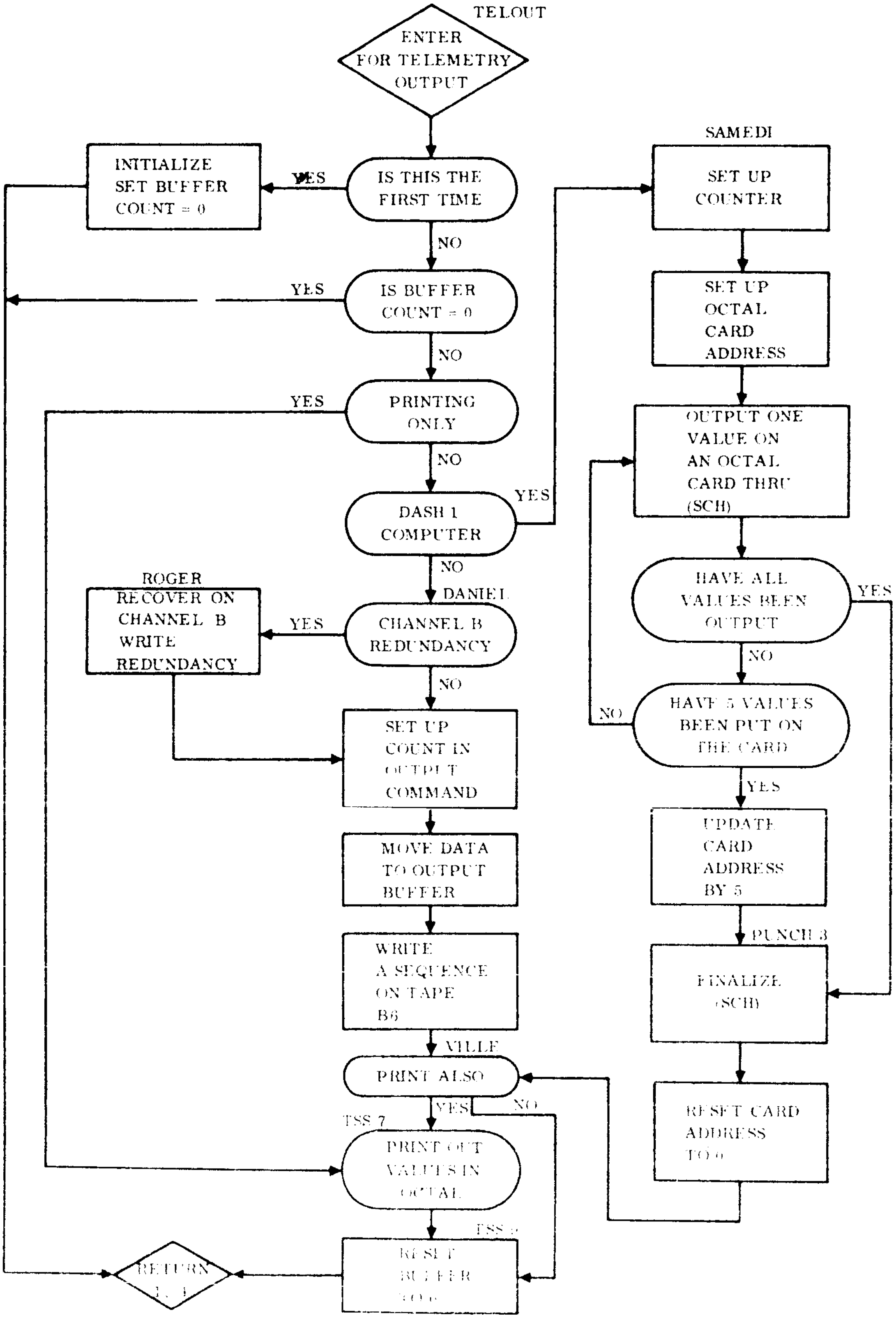

TELOUT
ENTER FOR TELEMETRY OUTPUT
INITIALIZE SET BUFFER COUNT = 0
YES
IS THIS THE FIRST TIME
NO
SAMEDI
SET UP COUNTER
YES
IS BUFFER COUNT = 0
NO
SET UP OCTAL CARD ADDRESS
YES
PRINTING ONLY
NO
OUTPUT ONE VALUE ON AN OCTAL CARD THRU (SCH)
YES
DASH 1 COMPUTER
NO
ROGER
RECOVER ON CHANNEL B WRITE REDUNDANCY
DANIEL
YES
CHANNEL B REDUNDANCY
HAVE ALL VALUES BEEN OUTPUT
YES
NO
NO
HAVE 5 VALUES BEEN PUT ON THE CARD
NO
SET UP COUNT IN OUTPUT COMMAND
YES
UPDATE CARD ADDRESS BY 5
MOVE DATA TO OUTPUT BUFFER
PUNCH 3
WRITE A SEQUENCE ON TAPE B6
FINALIZE (SCH)
VILLE
PRINT ALSO
YES
NO
TSS 7
RESET CARD ADDRESS TO 0
PRINT OUT VALUES IN OCTAL
TSS 8
RETURN 1, 4
RESET BUFFER TO 0

Figure A-39. Telemetry Output with (SCH)

APPENDIX B

The following tabulation lists the comments printed out in LICOS operation. The TEXT REFERENCE column entries refer to paragraphs in this report in which the COMMENT is mentioned. The NOTES REFERENCE entries refer to the Section B.2 NOTES of this appendix in which some of the COMMENTS are discussed briefly. The COMMENTS are divided into groups according to the circumstances under which they are initiated. These are: normal operation, "non-fatal" errors, "fatal" errors, and special tape output.

B.1 LICOS COMMENTS

COMMENT	TEXT REFERENCE	NOTES REFERENCE
B.1.1 Normal Output		
Starting Simulation		
BEGIN LIBRASCOPE COMPUTER SIMULATION	3.9.2f	B.2.1
ENTER SIMULATION AFTER PTO, TIME = ___	3.9.2d	B.2.2
RESTARTED SIMULATION AT ___ SECONDS	3.9.2b	B.2.3
Discretes		
MAIN ENGINE CUTOFF	3.2.3.3 3.2.4c	B.2.4
REORIENT DISCRETE	3.2.14b	B.2.4
SUSTAINER CUTOFF	3.2.14a	B.2.4
MODE ACCEPT DISCRETE	3.2.11c	B.2.4
RESET MAIN ENGINE CUTOFF	3.2.12b	B.2.4
GROUND MODE COMPLETE DISCRETE	3.2.14c	B.2.4
SELECT REGULAR TRACKS	3.2.12a	B.2.4
ORDER CODE TEST PASSED	3.2.10.4	B.2.4
EXTRA TRACKS SELECTED	3.2.13	B.2.4
STOP COMMAND ENCOUNTERED IN SIMULATOR	3.2.8	B.2.4
MASTER SEQUENCE START	3.2.10.1	B.2.4
BOOSTER CUTOFF	3.2.11a	B.2.4
VERNIER CUTOFF	3.2.11b	B.2.4
SIGNAL TO SET B GATE AT TIME ___ TIME	3.2.16	B.2.4
ACCUMULATION WILL STOP AT SIMULATOR TIME___	3.2.16	B.2.4
POWER TURN OFF (PTO)	3.2.9	B.2.4
LIBRASCOPE VALUE LOADED ON FLIP FLOP ___	3.2.16	B.2.4
FLIP FLOP TIME SENT TO COMP ___	3.2.15	B.2.4

B.1 LICOS COMMENTS, Contd

Simulation Comments

COMMENT	TEXT REFERENCE	NOTES REFERENCE
SKIP TIME PRINT OUTS ARE COMPLETE	3.8.2	B.2.5
MSCAN SET TO___AT TIME = ___	3.9.2c	B.2.6
FLIGHT MODE WAS CHANGED FROM ___TO___	3.9.1b	B.2.7
___WAS STORED ON THE SIGMATOR AS TIME AT TIME = ___	3.2.3.1	B.2.8
DELTA TIME TO CUTOFF = ___OR IN LIBRASCOPE FORMAT = ___	3.2.3.3	B.2.8
DELTA TIME STORED ON SIGMATOR FOR MAIN ENGINE COUNTDOWN AT TIME - ___		B.2.8

B.1.2 Non Fatal Errors

Overflow or Computational Errors

COMMENT	TEXT REFERENCE	NOTES REFERENCE
OVERFLOW ON LEFT SHIFT	3.2.6	B.2.9a
A RIGHT SHIFT OF 25 OR MORE PLACES	3.2.6	B.2.9a
LEFT SHIFT OF 25 OR MORE PLACES	3.2.6	B.2.9a
SUBTRACTION WITH AC = -1	3.1d	B.2.9a
DIVISION WITH AC = -1 AND OPERAND = -1	3.1f	B.2.9a
OVERFLOW HAS OCCURRED IN DIVISION	3.1f	B.2.9a
ADDITION OVERFLOW	3.1b	B.2.9a
MULTIPLICATION WITH AC = -1 AND OPERAND = -1	3.1c	B.2.9a

Telemetry Errors

COMMENT	TEXT REFERENCE	NOTES REFERENCE
DISCRETE WAS GIVEN TOO SOON AT ___	3.2.10.2	B.2.9b
DISCRETE WAS GIVEN TOO LATE AT TTSS	3.2.10.2	B.2.9b
VALUE STORED ON DATA LINK WITHOUT TSS AT TTSS	3.2.3.4	B.2.9b
A TSS OF ___WORD TIMES IS NOT DEFINED TTSS	3.2.10.2b	B.2.9b

B.1 LICOS COMMENTS, Contd

COMMENT	TEXT REFERENCE	NOTES REFERENCE
DATA WORD NUMBER ___ WAS CLOBBERED AFTER THE ___TH BIT	3.2.3.4 3.2.10 3.2.10.2	B.2.9b
Other		
THE SIGMATOR WAS NOT SET TO ZERO PROPERLY	3.2.3.2	B.2.9c
TRIED TO SET 9 FLIP FLOPS WITH H GATE CLOSED	3.2.15	B.2.9d
S6 WAS NOT SET BECAUSE DISCRETE WAS TOO SHORT	3.2.13	B.2.9e
PTO COMMANDED WITH FREEFALL TIME ON 9 FLIP FLOPS = 0	3.2.9	B.2.9f
PTO COMMANDED WITHOUT SETTING H GATE	3.2.9	B.2.9g
ERROR IN OUTS (LIF SET WRONG MODE) MODE ___	3.2.15	B.2.9h
MSCAN WAS READ WITH INCORRECT SECTOR BETA AT ___	3.2.5	B.2.9i

B.1.3 Fatal Errors

COMMENT	TEXT REFERENCE	NOTES REFERENCE
Input		
† THE ADDRESS ___ HAS TOO MANY DIGITS	4.2	
† VALUE GIVEN AT ADDRESS ___ HAS TOO MANY DIGITS	4.2	
SECTOR TOO BIG	4.2	
TRACK TOO BIG	4.2	
SCALED VALUE = TO OR GREATER THAN 1.	4.2	
ERROR IN CHECK SUM. ADDRESS OF THE CARD IS ___	3.9.2e	
REDUNDANCY CHECK WHILE READING DRUM. ADDRESS OF THE CARD IS ___	3.9.2f	

B.1 LICOS COMMENTS, Contd

COMMENT	TEXT REFERENCE	NOTES REFERENCE
THIS CARD IS OUT OF ORDER IN THE DATA SETUP	3.9.2f	
NO MORE THAN 50 EXIT ADDRESSES MAY BE USED	4.4	

During the Simulation

COMMENT	TEXT REFERENCE	NOTES REFERENCE
† ___ FROM COMBO TOO LARGE (CUL AND CONUL) ___	3.2.46	
†STORE INSTRUCTION WITH NON-EXISTENT TRACK CODE	3.3	
†TIME IS TOO BIG	3.2.4a	
IMPROPER CALLING SEQUENCE TO WAB527 AT ___	3.3	B.2.10c
† ******* LIBRASCOPE PROGRAM STOP **********	2.3	

B.1.4 Special Tape Comments

Reading the Telemetry Input Tape

COMMENT	TEXT REFERENCE	NOTES REFERENCE
REDUNDANCY ON CHANNEL A		B.2.11a
PTO READ THROUGH 1 FILE OK	3.9.2d	B.2.11b
FOUND EOF READING INPUT TAPE	3.9.2c	B.2.11c
READ AT PTO ALL FILES	3.9.2c 3.9.2d	B.2.11d
READ AT NORMAL ALL FILES		B.2.11e
TAPE CHECK (PTO READ)		B.2.11f

Writing the Model-3 Telemetry Output Tape

COMMENT	TEXT REFERENCE	NOTES REFERENCE
TELEMETRY SEQUENCE NUMBER ___	3.2.10.2a	B.2.12a
REDUNDANCY ON TELEMETRY TAPE, TELEMETRY TERMINATED		B.2.12b

B.1 LICOS COMMENTS, Contd

COMMENT	TEXT REFERENCE	NOTES REFERENCE
TELEMETRY BUFFER WAS EMPTIED WITHOUT SEQUENCE START		B.2.12a
TELEMETRY BUFFER WAS EMPTIED BY SEQUENCE START		B.2.12d
Writing Core on Tape		
BAD TAPE B5		B.2.13a
END OF TAPE B5		B.2.13b
TAPE B5 WAS SAVED AT ___ SECONDS	3.9.2a	B.2.13c
USING EXIT ADDRESS ____	3.9.2a	B.2.13d

†THESE COMMENTS ARE FOLLOWED BY THE COMMENT PROGRAMMER CALLED DUMP. ALL OTHER COMMENTS ARE FOLLOWED BY THE COMMENT PROGRAMMER CALLED EXIT FROM XXXXXXX WHERE XXXXX IS THE ABSOLUTE LOCATION IN CORE FROM WHICH THE EXIT WAS MADE

B.2 NOTES

1. Printed when all input, except sigmator values, has been read successfully and simulation is begun at 3100.

2. Printed when the simulation is restarted at 3100 after a power turn off discrete.

3. Printed when simulation is restarted from a tape.

4. Printed when discrete commands are given by the guidance program. See section on discrete simulation for further description of each discrete.

5. See section on Printing the Flight Matrix on a Time Basis, PATCH 5.

6. See Section on setting the MSCAN.

7. See section on Flight Mode.

8. See section on accumulator to sigmator instruction simulation, (storing on the countdown sector).

9. These comments have no effect on the simulation but are merely to inform the programmer of a possible error condition.

   a. Not printed if the ERCOM switch is set non-zero. See the individual instruction simulations for further information.

   b. Printed if telemetry output is requested; even if printed telemetry is not requested.

   c. See Accumulator to Sigmator instruction simulation (velocity and position).

   d. See load time flip-flop instruction simulation.

   e. See set S6 flip-flop instruction simulation.

   f. See power turn off instruction simulation.

   g. See power turn off instruction simulation.

   h. See section on Flight Mode.

   i. See MSCAN to accumulator instruction simulation.

B.2 NOTES, Contd

10. When an error of this type is encountered, the simulation is terminated; a core dump may be given, depending upon the type of error committed.

    a. This refers to values read in when restarting a simulation from tape.

    b. Same as a.

    c. This comment refers to a LIBRASCOPE value or location that has been read as input.

    d. Same as c.

    e. This comment refers to a LIBCON value.

    f. This comment refers to the binary drum deck.

    g. Same as f.

    h. This comment refers to the card that immediately follows the * DATA card.

    i. Maximum of 50 exit addresses allowable.

    j. A velocity or position input is too large to be handled by the LIBRASCOPE Computer.

    k. See section on undefined operations.

    l. The value of time too big to be handled by the LIBRASCOPE Computer.

    m. This is a programming error in LICOS. The calling sequence for printing refers to a variable location, but has a count of zero.

    n. See section on testing for the logical flow of the Librascope program.

11. Currently the telemetry input is unit A7.

    a. Printed each time a redundancy is encountered. The seven values that were read are printed in octal.

    b. Printed when a POWER TURN OFF instruction is simulated, the input tape is spaced ahead to the next file, and another file of data is expected.

B.2 NOTES, Contd

c. Printed when an end-of-file mark is encountered on a Model-1 floating point input tape during the normal compute cycle of the guidance program.

d. Printed when an end of file is encountered during the normal computer cycle of the Model-3 computer, or if all files of data have been successfully read, a power turn off instruction was simulated, and no more files are expected.

e. Printed when an end of file mark is encountered in the normal compute cycle of the Model-1 computer, but a power turn off instruction had been previously simulated and the input tape spaced ahead to the next file, and no more files are expected.

f. When a POWER TURN OFF instruction has been simulated, LICOS reinitializes and advances the input tape to the next file. If more data are expected, the first record of the new file is read. If a redundancy check occurs, this comment appears. After 3 unsuccessful attempts a core dump is called.

12. Currently the telemetry output tape for the dash 3 computer is unit B6.

a. This is the number of sequences output by LICOS, modulo 2 for the Model-3 computer, but the actual number for the Model-1 computer.

b. Printing of telemetry continues as requested, but an end-of-file mark is written on B6 and no more output is written on tape.

c. For the Model-3 computer, the record size is 30 words and no sequence start word is in the record.

d. This is the normal situation with the record size varying with the number of words per sequence.

13. Currently the save core tape is unit B5.

a. If the redundancy cannot be recovered after two trys, the run is terminated with this comment.

b. If the end of tape is encountered before core is completely written the run is terminated.

c. Printed when core is being written on tape, with the simulation time at which this action took place.

d. Printed when [illegible] other than on a [illegible].