NRL Report 6446

Computer Program for the NRL Satellite Position Display

R. J. ORSINO AND T. L. FRANCAVILLA

Data Processing Branch
Applications Research Division

December 5, 1966

NAVAL RESEARCH LABORATORY
Washington, D.C.

CONTENTS

This page intentionally left blank.

ABSTRACT

A computer driven satellite display has recently been developed and evaluated at the Naval Research Laboratory. This equipment, designated as the NRL Satellite position Prediction And Display (SPAD), provides a considerable amount of display control versatility. Up to eleven satellites can be selected either individually or by category from a repertoire of 88 satellites. These selected satellites are then updated and displayed in either real time or at an accelerated time. Several display modes are available at an operator's command, namely, world map, rectangular expanded mode, and polar expanded mode.

A commercial equivalent of the AN/UYK-1 computer was used in the research version of SPAD. Since the program was written in a language peculiar to this machine, the program listings and codings are omitted from the present report. However, included in this report are the complete set of computer program flow diagrams and a brief description of the interrupt service routines along with their corresponding subroutines. Characteristics of the AN/UYK-1 computer, operator control panel provisions and possible future program modifications are discussed. The contents of this report should be a sufficient indication of the program philosophy. This philosophy, with appropriate minor changes, could be implemented by any suitable language.

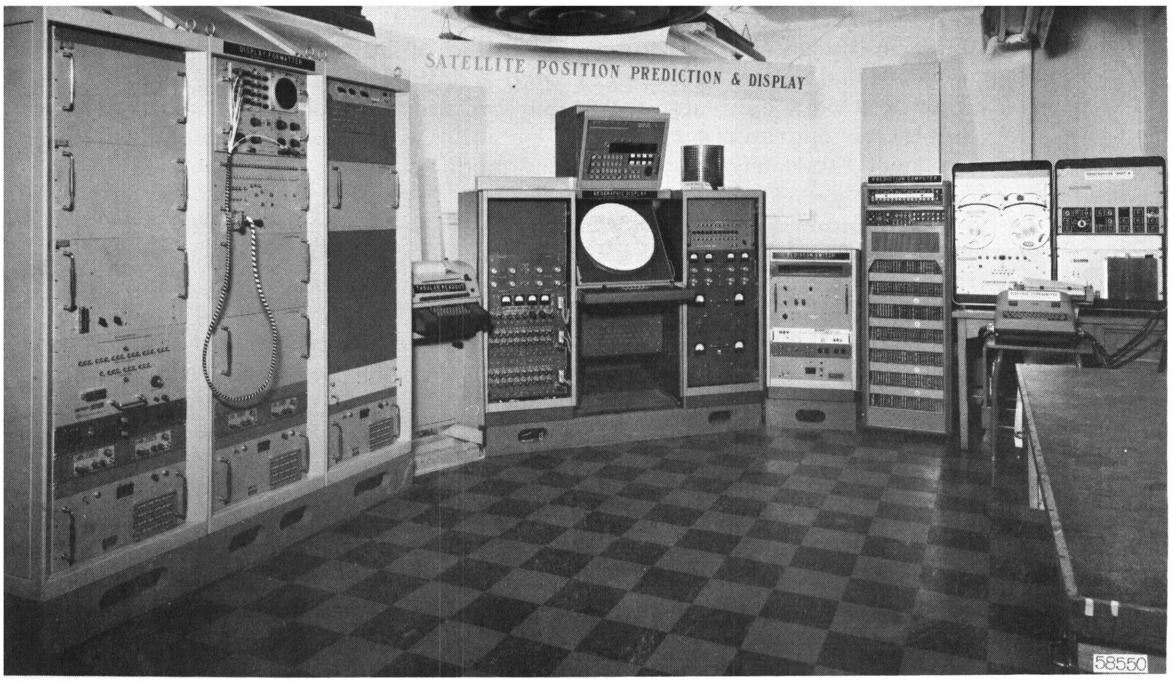
PROBLEM STATUS

This is an interim report; work on the problem is continuing.

AUTHORIZATION

NRL Problems Y01-01
Project SF 019-01-03, Task 6168

Manuscript submitted June 9, 1966.



SPAD system

COMPUTER PROGRAM FOR THE NRL SATELLITE POSITION DISPLAY

INTRODUCTION

A Satellite position Prediction And Display equipment (SPAD) has recently been developed at the Naval Research Laboratory. The information to be displayed is provided by an AN/UYK-1 digital computer and consists of latitude, longitude, and height predicted approximately each second for as many as 11 satellites chosen from a repertoire of 88 satellites. Satellite positions can be updated in real time or at an accelerated time. The system versatility resides mainly in the many operator-selected options which can tailor the display to satisfy a wide range of needs. The normal operating mode is the world map presentation. Expanded rectangular and polar modes are available on command, as well as a complete orbit and tabular readout on any one satellite. The operator can select and delete satellites either individually or in groups according to categories.

In order to perform the many operations demanded the computer program was necessarily complex. Although the program was written in an interpretive mode using a program language peculiar to this particular computer, the program philosophy as reported here could have been implemented by any suitable language. This interpretive mode was chosen as the most expedient manner of obtaining a workable program from personnel not intimately familiar with the machine language used in the machine. Thus, the program as written is not necessarily the most efficient. The program uses approximately 8000 of the possible 8192 memory locations for storing the operational program, the raw data on 88 satellites, the tables of constants, and the necessary working locations.

This report concerns itself primarily with the program philosophy and includes the complete set of flow diagrams, along with a brief description of each of the separate interrupt routines and their corresponding subroutines. Pertinent facts concerning the digital computer, the operator control panel provisions, and possible future SPAD program modifications are also given.

COMPUTER DESCRIPTION

General

The AN/UYK-1 is a medium-size, solid-state digital computer which conforms to specifications for peripheral devices in the Navy Tactical Data System (NTDS). It is a binary parallel word machine, using two's complement arithmetic, with a magnetic core memory containing 8192 15-bit words (expandable to 32,768 words). The computer reads and writes in parallel with memory access time of 3 μsec and a cycle time of 6 μsec. The clock rate is 333 kc.

Communication between the computer and peripheral devices is controlled by the computer's input/output (I/O) section. Three sets of cables are available, each set of which contains an input and an output channel. All the cables contain input and output data lines. Two cables, A and B, transfer 30 bits of parallel data per channel, while the third, cable C, transfers only 15 bits. This latter cable is generally used for communicating with standard peripheral devices such as a page printer, a tape reader and punch, and a typewriter. The distribution to each device is performed through a buffer designated the controller.

Control Lines

Each of the three cables contains two sets of control lines, one for input and one for output. The conventional use of the input lines, which differs somewhat from the SPAD use to be described, is as follows:

1. Interrupt lines—used by an external device in order to interrupt the computer when it is not electrically connected to data lines.

2. Input data request (IDR)—used by an external device that is electrically connected, to signal the computer that data are present on the input lines.

3. Input acknowledge—used by the computer in response to either of the above interrupts, to signal the external device that input data has been sampled or an interrupt has been executed.

The conventional use of the output lines is as follows:

1. Output data request (ODR)—used by an external device to signal the computer that it is ready to accept an output word.

2. Output acknowledge—used by the computer in response to an ODR,* to signal the external device that data are present on the output lines.

3. External function (EF)—used by the computer to signal the external device that a word present on the output lines is to be interpreted as a command or control word as opposed to a data word.

Interrupt Scheme

Type II Interrupts – An external device is normally assigned at least one unique interrupt line over which it has control at all times. In response to a type II interrupt signal the computer examines all the interrupt lines in order to determine the source of the interrupt and then proceeds to electrically connect the data lines to the interrupting device by executing control function (CF) and external function (EF) commands. Following connection if the external device can have more than one reason for generating an interrupt the computer may input an interrupt code word from the input data lines to determine the reason.

Type I Interrupts – A device already connected electrically to the computer data lines can interrupt the computer via the IDR or ODR lines; these interrupts are considered type I interrupts. Since the computer does not have to examine the source of the interrupt, it can respond more rapidly to this type of interrupt than it can for type II.

Input/Output Section as Used by SPAD

The manner in which SPAD uses the computer I/O section is slightly different from the conventional use described in the preceding section. The difference lies mainly in the method for reading out data and also in the use of the interrupt lines.

The primary communication link between the computer and the display is the B cable. Input data, such as time, and operator commands are transferred to the computer via the

*In addition to being explained where first introduced in the text, terms and symbols are also listed alphabetically in Appendix A.

30 data lines of the B_{in} channel of the B cable. All data to be displayed is sent by the computer via the 30 data lines in the B_{out} channel of the B cable. Information to be placed on the page printer is transferred via the 15 bits in the C cable. The A cable is used for placing own ship position and heading into the computer.

The AN/UYK-1 readout rate is such that the external SPAD equipment could accept the data words as fast as the computer could transfer them. Consequently, it was not necessary for the external equipment to send an ODR signal to the computer in the conventional manner when it was ready for a data word. Instead, the computer signals the external equipment. The external function line was thus used to signal the SPAD equipment that a data word was present on the output lines rather than to signal that a command or control word was present as in the conventional use described earlier. The external function operation is exclusively under program control; i.e., no input command or response is required. The program was easily written to place data on the output lines instead of command words. It is evident that if ever the external equipment was not ready to accept a new word from the computer, that word would be lost. However, a careful study indicated that, aside from some catastrophic malfunction of the external equipment, it would always be ready. Furthermore, in only a few situations does the computer read out two words sequentially; all the other data are read out following rather lengthy computations, thus allowing sufficient time between data words for equipment settling.

Although convention calls for one interrupt line to be assigned to each unique external device, the SPAD equipment uses nine of the 11 available type II interrupts, although only one external device, the SPAD computer control panel, is capable of generating interrupts. This unorthodox procedure permitted a simpler interrupt service routine, since the computer does not have to decode the interrupt code word in order to determine what function to perform but merely recognizes each interrupt as an instruction to perform a certain unique function. Several of the interrupts, discussed later in this report, do transfer a code word, but this word contains only information concerning what satellite to perform the function on; the function itself is already known by virtue of the particular interrupt line that was activated. In this manner, if any new function was desired, it would merely have to be programmed and assigned an interrupt line, without modifying any previous programming.

Programming with Stored Logic

In the stored logic technique, most of the logic that is normally wired into the computer is instead stored as part of the program in the computer memory. The AN/UYK-1 responds to three levels of instruction: microcommands, logic commands, and logic programs. The characteristics of each are as follows:

1. Microcommands are wired into the machine and cause actions such as shifts and transfers between the computer's six registers, adder, and address counter. On every clock pulse a number of these actions occur simultaneously.

2. Logands (a contraction of logic commands) are the AN/UYK-1 machine language instructions which, after being written, are stored in the computer memory. The structure of these commands, of which there are approximately 8500 possibilities, cause a corresponding sequence of simultaneous microcommands to be performed. This is by far the most efficient method available to the programmer both in time for execution and in memory space.

3. Lograms (a contraction of logic programs) are sets of logands and provide both simple instructions such as branch unconditional as well as complex instructions such as arctangent. This method is the simplest for those uninitiated in complex programming

techniques. A logram program is written as a tabulation of calling sequences for existing lograms available from the manufacturer as a software package. Thus, the program using lograms can be considered as a listing of subroutine addresses.

The SPAD program consists mainly of the lograms available in the software package. Several special lograms were written by SPAD personnel as peculiar requirements demanded. All the interrupt service routines and the input/output instructions were written in machine language, i.e., in logands.

MODES OF DISPLAY OPERATION

There are three modes of operation available for display: world map (WM), rectangular expanded, and polar expanded. An operator can select either world map (WM) or expanded (EXP). During EXP, the display automatically switches from rectangular to polar whenever the target (own ship or satellite about which expansion takes place) moves to within 15° of either geographic pole. When the target distance from the pole increases to greater than 15°, the display switches back to rectangular expanded mode. The computer program provides the necessary code words to the display in order to effect the appropriate electronic switching for display modes.

World Map Mode

The present SPAD program provides memory space for orbital element data on 88 satellites. Data for each satellite consists of the nine 30-bit words listed in Table 1. Thus, 792 thirty-bit words, or 1584 fifteen-bit memory locations compose the satellite raw data (SRD) table. From this store of 88 satellites, a maximum of 11 satellites chosen to be updated is stored in the satellite-displayed data (SDD) table. In order to reduce the position computation time, several often-used quantities are computed and stored in the SDD table (see Table 2) as part of the routine which moves the orbital element data from SRD table to the SDD table. The move routine has been designated as SUBMV. For each of the 11 satellites there are 32 words of 15 bits stored in the SDD table, or for all 11 satellites a total of 352 memory locations. The preceding events occur prior to actual entry into the WM loop.

Table 1
Satellite Raw Data

Data Word (30 bits each)	Use
1	Most significant 16 bits - Satellite category indicator Least significant 9 bits - Look-cone angle, scaled 2^7
2	Least significant 16 bits - SPADAT, number in binary-coded digits (BCD)
3	T_o - Epoch modified to passing of perigee, in seconds from beginning 1960, scaled 2^{29}
4	a - Orbit semimajor axis, in earth radii, scaled 2^4
5	e - Eccentricity, scaled 2^0
6	i - Angle of inclination, in radians ≥ 0, scaled 2^3
7	ω - Argument of perigee, in revolutions ≥ 0, scaled 2^0
8	Ω - Longitude of ascending node, in revolutions ≥ 0, scaled 2^0
9	$\dot{a}$ - Time rate of change of semimajor axis, in earth radii per second, scaled 2^{-25}

Table 2
Satellite Displayed Data (SDD) and Satellite Being Computed Data (SCD)

Data Word (30 bits each)	Contents	Bit Position	Use
1	-	30 29 27-12 11- 8 7- 1	Section status Fill source (TN or category) Category Blank TN (binary)
2	T_0	All	Epoch modified to perigee in binary seconds since 1960 scaled 2^{29}
3	a	All	Semimajor axis in earth radii scaled 2^4
4	$a^{3/2}$	All	Three-halves power of semimajor axis in earth radii scaled 2^6
5	e	All	Eccentricity scaled 2^0
6	$e/2\pi$	All	Scaled 2^0
7	$(1-e^2)^{1/2}$	All	Scaled 2^1
8	sin i	All	Sine of the angle of inclination scaled 2^0
9	cos i	All	Cosine of the angle of inclination scaled 2^0
10	ω	All	Argument of perigee in revolutions scaled 2^0
11	Ω	All	Longitude of the ascending node revolutions scaled 2^0
12	$\dot{a}$	All	Time rate of change of semimajor axis scaled 2^{-25}
13	$(1-e^2)^2$	All	Scaled 2^0
14	-	30-16 15-13 12- 5 4- 1	View cone radius angle scaled 2^3 Space TN in BCD Category in BCD
15	-	30-22 21-16 15- 7 6- 1	Space Symbol address Space Position address
16	e sin E	All	Revolutions scaled 2^0

Once in the WM loop, the computer updates (computes a position for) each of the 11 satellites in turn in cyclic fashion. The data for the particular satellite presently being updated is transferred from the SDD table to the working data table of the position computation subroutine, SUBP1 or SUBP2. This working table is designated as the satellites-being-computed data (SCD) table. The manner of choosing the satellites whose data is placed in the SDD table will be discussed in later sections of this report.

The sequence of events occurring in the WM loop is shown in Fig. 1a. Upon initial entry in the WM loop, a return address is set to WM. This return address is required so that, following the service of an interrupt, the computer knows whether it was interrupted during a WM or EXP loop and thus can return to the appropriate loop. Subsequent cycles through the WM loop bypass this instruction. The return address is changed only when

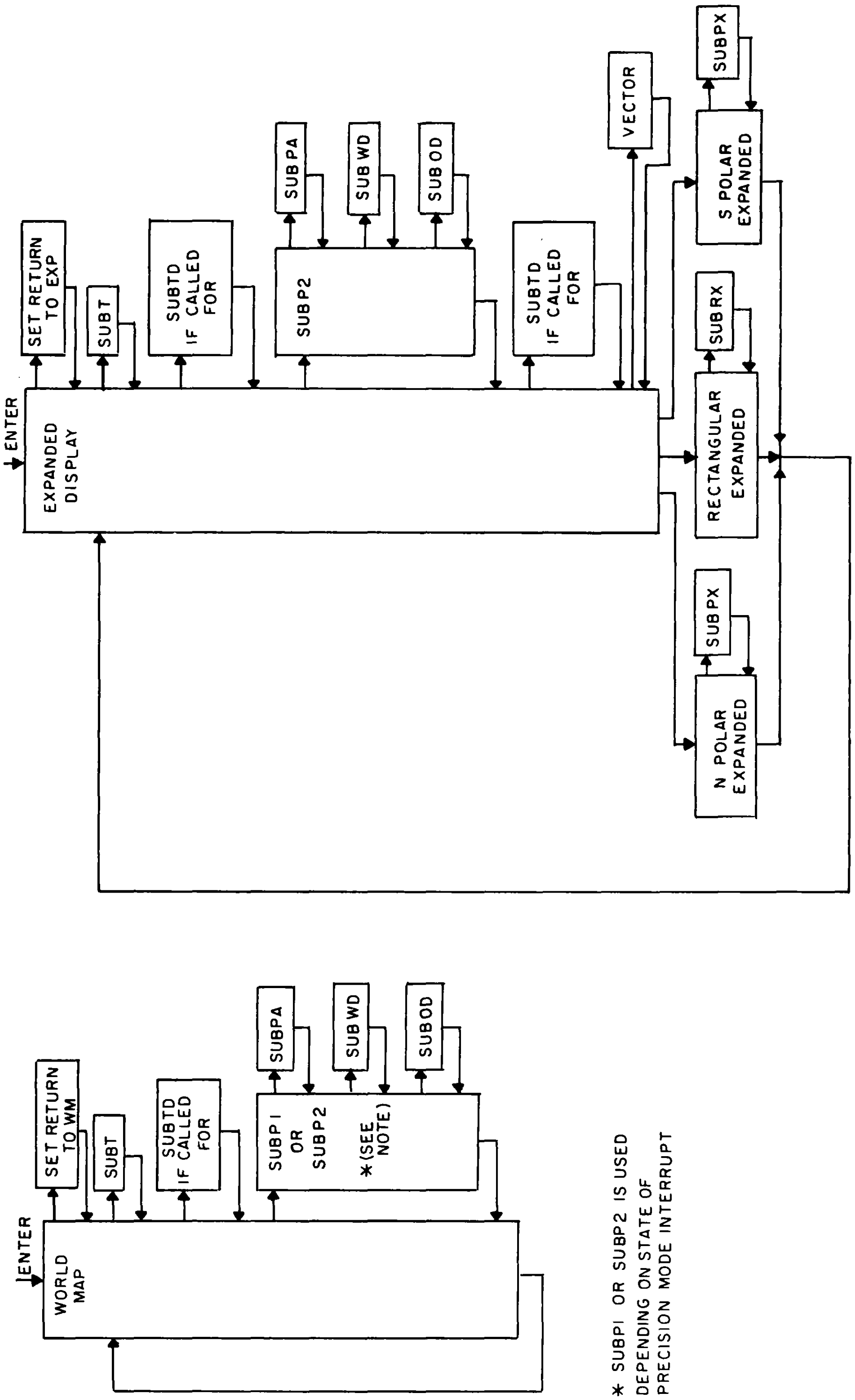


Fig. 1 - Sequences of events for the three modes of display: (a) world map loop; and (b) expanded display loop

the operator changes the display mode i.e., when the operator pushes the MAP SCALE button on the computer control panel and thus switches from WM to EXP, or vice versa.

Prior to the start of a position computation, the latest value of time received by the computer from the computer control panel is converted into seconds by the SUBT routine. This binary number represents the time in seconds, minutes, hours, and days sent to the computer in binary-coded-decimal (BCD) form.

Following the conversion of time, the SUBTD routine checks whether the operator has requested a tabular readout; if he has not, the computer continues with position computation. If he has, the page printer is connected, one character is read out, and the printer is then temporarily disconnected. Since the page printer operates on an interrupt basis, it is necessary to disconnect the printer after each character in order to print out only one character for each WM cycle. Unless disconnected, the printer would request a new character as soon as it completed a previous one, thus interrupting the normal sequence of the program at arbitrary locations. Interleaving of this kind presented many problems, so it was decided to have printout occur only at fixed locations in the WM loop. The data for printout at this time is already computed and stored in the TDU1 table, in the five-bit teletypewriter code for subsequent readout.

After servicing the page printer, if required, the program then enters either of two position computation subroutines: SUBP2 for double-length or 30-bit computations, and SUBP1 for 15-bit computations. The operator selects either of these subroutines by means of the PRECISION SELECT button on the computer control panel. Associated with either of the SUBP subroutines are other subroutines: SUBPA for computing the average anomalistic period, SUBWD for the rotation of perigee, and SUBOD for the regression of the node.

Upon completing a position computation, the computer reads out latitude and longitude coordinates of nine bits each in a position output word and then reads out the category symbol, track number, and height in a symbol output word. The data for the next satellite in the SDD table is then transferred to the SCD table, time is converted to binary seconds, and the cycle repeated. When all 11 satellites are updated, the computer returns again to the first satellite in the SDD table.

Expanded Modes

For the most part, the EXP routine is identical to the WM routine. However, only one satellite, chosen by TN, and stored as the eleventh set of data in the SDD table, is updated and displayed during this mode. The position of a ship known as own ship (OS) is also displayed. The operator can select an expansion about either the satellite or the ship by means of the OFF-CENTER pushbutton on the computer control panel. Regardless of the chosen expansion, both the ship and the satellite will appear on the display if their positions are within the same expanded area.

The sequence of events is shown in Fig. 1b. Upon entry to the expanded routine, the return address is set to EXP rather than WM, in order to indicate which loop the computer will be sequencing through. As in the WM loop, time is converted to binary seconds by the SUBT routine, the page printer is serviced by the SUBTD routine if required, and a position is computed for the satellite by the SUBP2* routine and its associated SUBPA, SUBWD, and SUBOD subroutines.

*The SUBP2 or 30-bit double-precision computation routine is always performed when the computer is in the expanded mode. Thus, a nonprecision mode selected has no effect at this time.

The first major difference between the WM and EXP routines is encountered following the SUBP2 routine, where in the EXP loop the SUBTD routine is entered again (see Fig. 1). Since the total execution time for the expanded routine is about twice as long as the WM routine, two page printer characters are read out for each position computation of the EXP loop rather than one as in the WM loop. This was provided in order to maintain as close as possible the printout rate of the tabular message for both display modes.

Following this second entry into the SUBTD routine, the computed position for the satellite or the position of the ship, depending on whether expansion is to be about the satellite or the ship, is tested for its general location. Specifically, if the position is within 15° of either pole, entry is made into the appropriate polar expanded routine, north or south; if the target does not fall within either polar area, the rectangular expanded routine is entered. Each of these routines, and their associated subroutines, SUBRX and SUBPX, perform the appropriate scale conversions and prepare the position coordinates in the output format for display in an expanded mode. Also performed in these routines is the determination of the particular map area to be displayed. The size of this area is always 20° in latitude by 30° in longitude, with the satellite or ship always falling within the 10° by 20° central area. Thus a 5° border is provided to account for a trailing vector tail dot which may be as much as 5° behind a satellite. Once the area is determined, the computer reads out the off-center map-grid word to the peripheral equipment known as the formatter. This word consists of coded bits designating the lower and the leftmost map grid lines of the previously mentioned central area. The formatter then uses this code word to select the map grid symbol group. Thus, as the target moves from one expanded area to another, the computer sends out the appropriate code defining the area, as well as the expanded coordinates of the target position.

After each position prediction computation in the expanded mode, a vector tail position is computed in the following manner. One minute is subtracted from the position prediction time and another position computation is performed. The result is the position of the vector tail dot, which is the satellite position 1 minute earlier. This is then read out as a separate word after the satellite position information.

OPERATOR-SELECTED INTERRUPTS

List of Interrupts

All those interrupts which the operator selects are generated by pushing appropriate buttons on the computer control panel shown in Figs. 2a and 2b. The complete list of operator-selected interrupts is as follows:

1. Status interrupt
2. Cycle interrupt
3. Cancel interrupt
4. TN interrupt (delete, call-up, orbit)
5. Tabular readout interrupt
6. Precision mode interrupt
7. Month interrupt
8. Year interrupt
9. Own ship interrupt*

*For the evaluation conducted at NRL, an own ship simulator panel permitted position, course, and speed to be set up on switches. This own ship information was then transmitted to the computer via a manually generated interrupt. For a shipboard installation, the own ship information would be automatically generated and the computer would be automatically interrupted.

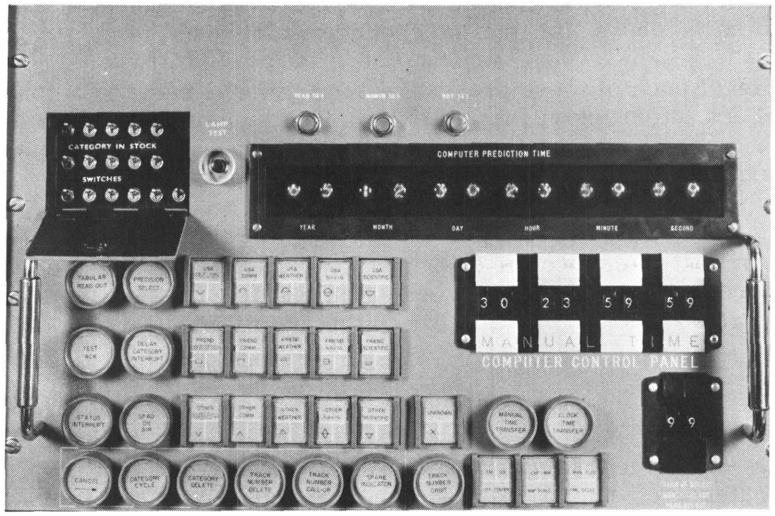


Fig. 2a - Computer control panel

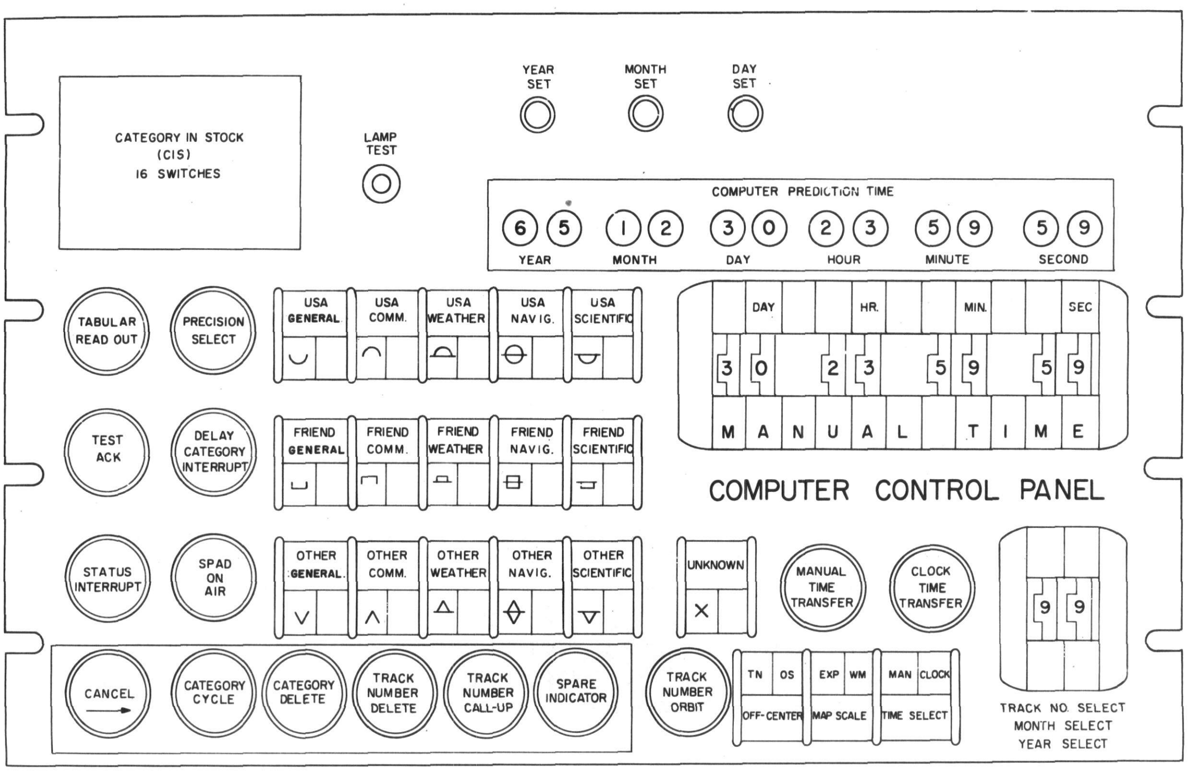


Fig. 2b - Computer control panel

In some cases, the interrupt service routine performs only one function when the interrupt line is activated (e.g., own ship interrupt). In other cases, some processing must go on to determine what service routine must be called in response to a given interrupt. The order given in the above list is of no special significance.

A description of each of these program interrupts will refer to a condensed interrupt flow diagram. Each interrupt will first be described by considering the actions which are observable to the operator generating them; also, pertinent consequences, if applicable, will be pointed out. Then each interrupt will be discussed in detail from the computer program viewpoint, with emphasis on the philosophy of the operations. The operator's viewpoint will be brief, since the details of this phase of the operation have been reported in Ref. 1. In choosing to discuss the interrupts separately, while indeed many interrupt routines are interdependent, certain sequences in the flow of operations covered in this text will not necessarily conform explicitly to the main flow diagrams of Appendix B.

Status Interrupt

The SPAD system categorizes satellites into 16 somewhat arbitrary groups or categories. These categories can be seen in Fig. 2b and are listed in order of priority as follows:

1. Unknown
2. Other general
3. Other communications
4. Friendly general
5. Friendly communications
6. USA general
7. USA communications
8. Other weather
9. Other navigation
10. Friendly weather
11. Friendly navigation
12. Friendly scientific
13. USA weather
14. USA navigation
15. USA scientific
16. Other scientific

The operator can choose any number of these categories for display by activating the appropriate pushbuttons on the computer control panel. Only ten satellites chosen in this manner can be updated and displayed at any one time, since only ten stores in the external buffer equipment were provided for this purpose. Each of the 16 pushbuttons contains three separate computer-controlled and illuminated sections. Briefly, the lower-left section designates CAR, those categories which have been requested by the operator; the upper section designates CAS, the one category in which some satellites are being updated but, due to the ten-satellite limit, a search is continuing for satellites in that category; and the lower right section designates CAO, those categories in which the computer has not been able to update any satellites, and thus in which an overload condition exists. An eleventh satellite can be displayed but is chosen by another interrupt and will be discussed in those sections describing track number (TN) requests.

Each of the 16 categories is assigned a unique bit position in the data word that is sent to the computer along with a status interrupt (Fig. 3). Upon examining the interrupt lines and determining that a status interrupt exists, the computer accepts the input data word and interprets the data as a category request. Also part of the data word is a bit position which indicates whether the state of the MAP SCALE pushbutton is the WM mode or the EXP mode. Thus a push of the MAP SCALE button also generates a status interrupt.

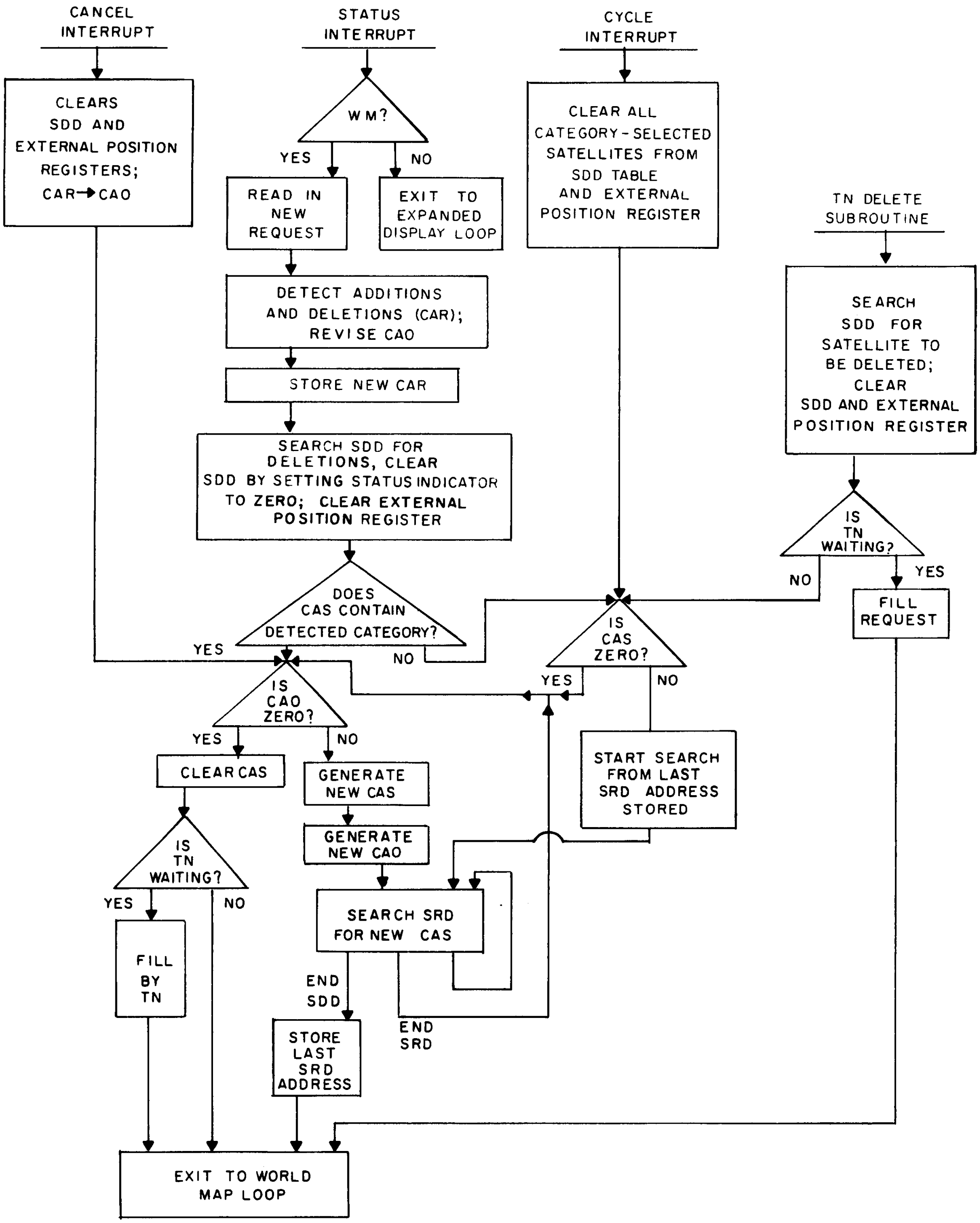


Fig. 3 - Flow of operations of three of the operator-selected interrupts

The computer's immediate task upon receiving a status interrupt is to update the three status registers, i.e., the CAR, CAS, and CAO indicators previously defined. It does this by detecting in the new data word any additions or deletions from the previous data, specifically, from the CAR register. The CAO register is thus revised, adding and removing categories as appropriate. Generally, removal of CAO-indicated categories would not occur, since the operator normally would not cancel a previously requested category unless he had at least observed some satellites in that category. And once satellites are observed, the corresponding category is no longer in an overload condition. Following a revision of the CAO register and the corresponding CAO indicator lamps, the latest request then becomes the new CAR indication.

As mentioned previously, the satellites being displayed have their data stored in the SDD table. Due to a change in status incurred by a status interrupt, the data in these tables must be modified accordingly. Thus, the SDD table is searched in order to delete data for those satellites in categories no longer requested. Upon finding a satellite to be deleted, a "status and fill source" indicator, consisting of two bits in each of the ten satellite stores of the SDD table, is cleared to designate that the corresponding data store is now vacant. The two bits indicate whether the satellite data had been placed in the SDD table due to a category request or to a TN call-up request. Unless either one of these bits is set, the store is considered vacant. Only those satellites selected by category can be so deleted. Selection and deletion of satellites chosen by TN will be discussed under the appropriate heading. Following the deletion of the data from SDD table, a code word is sent out to the peripheral buffer stores to designate which of the satellites being displayed should be deleted from the display.

With vacancies, if any, established in the SDD table and the buffer stores, the computer can now proceed to fill these vacancies with data from the SRD table of satellites requested but not yet displayed, e.g., those in the CAS and CAO indicated categories. As satellites are found in the SRD table and moved to the SDD table, the "fill source bit" for satellites chosen by category is set. The priority sequence for updating satellites of chosen categories has also been somewhat arbitrarily set according to the list at the beginning of this subsection. The position of each category bit in the status interrupt data word indicates the priority, i.e., bit 30, the unknown category, has the highest priority, while bit 15, the other scientific category, has the lowest. The SRD table is first searched for satellites remaining to be displayed in the category designated by the CAS register, unless that category happened to be deleted by the new status interrupt data request. It should be recalled that the CAS register indicates that category in which some but not all the satellites of that category have been displayed. If the category indicated by the CAS register has indeed been deleted, then the next highest category designated in the CAO register becomes the new CAS category. If there are no categories waiting in the CAO condition, a satellite chosen by TN can be placed in the SDD table and subsequently displayed. The manner in which such a satellite could have been selected and waiting will be discussed later under the TN call-up section of the TN interrupt description. If neither of the above conditions exist, the computer exits to the WM loop, where it resumes the normal updating cycle described in the WM mode section of this report.

Cycle Interrupt

As mentioned previously, only ten of the 11 satellites being updated and displayed can be chosen in groups according to category. However, some or all of these ten can be chosen by a TN call-up request. The eleventh satellite is chosen by a TN orbit request or a TN tabular readout request. Details of these TN requested satellites will be described under the appropriate interrupt headings.

The cycle interrupt (Fig. 3) permits an operator to delete from the display the present ten satellites chosen by category and to bring in a maximum of ten satellites that may be waiting, e.g., those in CAS and CAO indicated categories. If, however, some of the ten presently being displayed have been chosen by the TN call-up interrupt, they are not affected by the cycling. Thus, if two of the ten have been so chosen, the cycle interrupt replaces only eight with waiting satellites.

The computer is programmed to perform this function by clearing the two-bit "status and fill source indicator" of the SDD table for all satellites chosen by category. With vacancies now present in the SDD table, the computer branches to that portion of the status interrupt routine described earlier which fills the vacancies with those satellites requested but not yet displayed. New CAS and CAO words are formed in the appropriate registers according to the priority sequence described previously. Through the use of the cycle interrupt, an operator can observe all satellites of those categories he requested. When all the satellites have been seen, the CAS and CAO registers will contain all zeros, and the corresponding CAS and CAO indicator lamps in the pushbuttons on the computer control panel will be extinguished. Thus, the illumination of only CAR indicator lamps on the control panel informs the operator that his entire category request has been completed. That is, all satellites of the chosen categories have been or are being displayed.

Cancel Interrupt

The original category request (stored in the CAR register) can be modified by the individual deletion of satellites in the chosen categories with the TRACK NUMBER DELETE button, and by replacing these with satellites not necessarily in the chosen categories with the TRACK NUMBER CALL-UP button. A cancel interrupt (Fig. 3) was thus provided in order to remove all requests made subsequent to the last category request. Specifically, all TN delete and TN call-up requests are no longer recognized, and the computer resumes operation as if they did not occur.

The cancel interrupt routine is a rather simple one. Since the interrupt is merely a command to perform a specific operation, there is no data transfer connected with this particular interrupt. Upon receipt of the interrupt, the computer is programmed to clear the data for all ten satellites in the SDD table regardless of how they were chosen. It then reads out the appropriate data word for each of the ten external buffer stores or registers, indicating that those stores are to be cleared. In clearing these position stores, all video is removed from the display. However, due to the speed of the computer the interval of time between the removal of video and the subsequent replacement with new video is imperceptible to an observer. After clearing the external stores, the last CAR word stored in the CAR register is then transferred to the CAO register. From this point on, the computer reacts as if it has just received the original category request. The computer proceeds to that portion of the status interrupt routine which searches the SRD table for satellites in normal priority sequence. New CAS and CAO words are formed as necessary.

TN Interrupt

There are two classes of TN requests (Fig. 4) which, although they are different, nevertheless share a common interrupt line. The TN delete and TN call-up requests are used to modify the stores containing the ten satellites chosen by category and thus are in the same class. Data for the satellite requested by the TN orbit interrupt is placed in the eleventh SDD table store; thus the TN orbit request is in a different class than the TN delete and TN call-up requests. Expansion of the display can occur only for the particular satellite requested by the TN orbit interrupt (expansion is also available for the own ship). The tabular readout request, which is an independent interrupt line also can be used to place satellite data in the eleventh store, but this will be discussed as a separate interrupt later. Thus, for expanding about a satellite, either a TN orbit or a tabular readout for that satellite must initially have been requested.

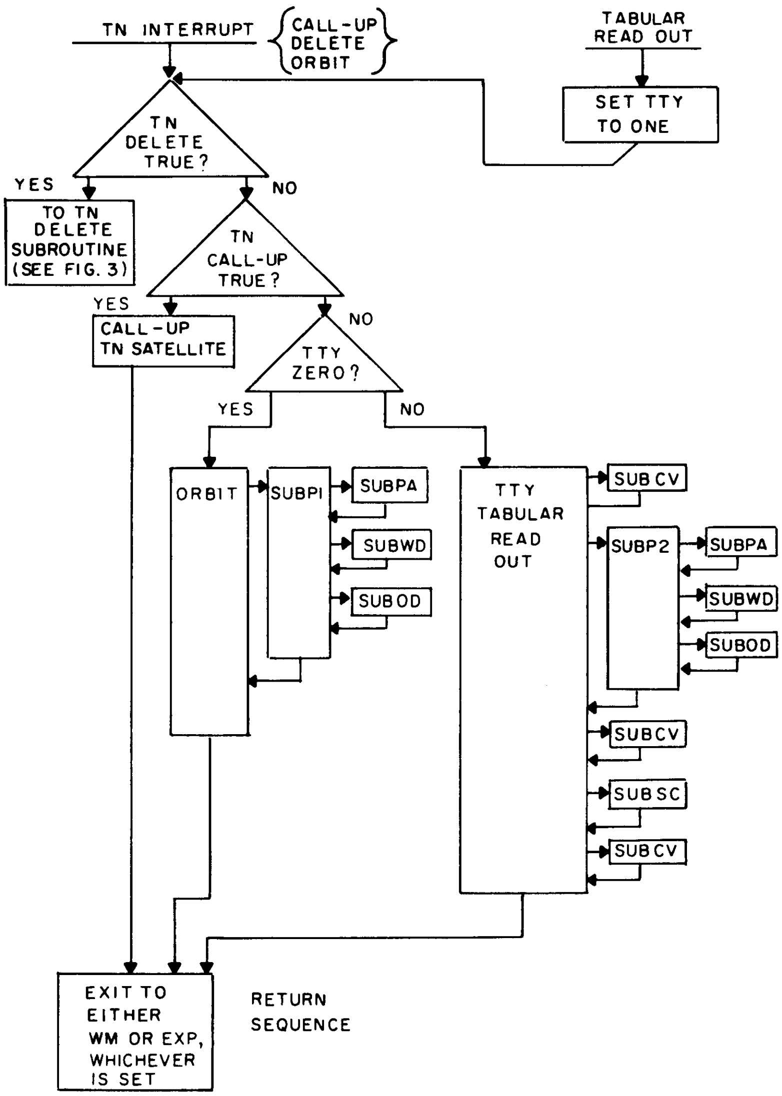


Fig. 4 - Flow of operations of two of the operator-selected interrupts

TN Delete - The data word sent to the computer along with the TN interrupt contains the track number in BCD form for the numbers 00 through 99 as well as a code word designating which of the TN requests was responsible for the interrupt. The first code tested is that corresponding to the TN delete request. If that code is not present, the TN call-up code is tested.

When a TN delete has been requested, the computer decodes the BCD track number to binary, and then searches the ten SDD satellite stores for the matching satellite. Upon finding the satellite to be deleted, the corresponding SDD table store is cleared and an instruction is sent to the peripheral equipment to clear the appropriate external position store. If due to an operator error the TN to be deleted is not in the SDD table and therefore not on the display, the computer, in effect, ignores the request.

With a vacancy so obtained in the SDD table, the computer first inquires if there is a satellite, chosen by the TN call-up request, waiting to be displayed. If there is, the data for that satellite is placed in the SDD table and subsequently the satellite is displayed. If a TN call-up satellite is not waiting, the computer branches to that part of the status interrupt routine where a search is made in the SRD table for the next satellite of the CAS indicated category waiting to be displayed. From this description and the one given under the cycle interrupt section, it is evident that with a TN delete interrupt, the computer attempts to fill the SDD vacancy with a waiting TN call-up satellite first. On the other hand, the category cycle interrupt first fills vacancies in the SDD table with satellites chosen by category and then checks for one chosen by a TN call-up request. For the next model SPAD, giving priority to TN call-up satellites regardless of how a vacancy was created is under consideration.

TN Call-Up — If the TN code in the data word sent along with the TN interrupt is not a TN delete request code, the computer tests for the code representing a TN call-up request (Fig. 4). If the interrupt was caused by neither of these, the computer surmises that the remaining TN request, a TN orbit, has been requested.

When a TN call-up request has been made, the TN waiting indicator is set and the track number is stored. If a vacancy exists in the SDD table, the data for the satellite called-up is placed in the vacancy and the TN waiting is cleared. If no vacancy exists, the computer resumes its normal updating of satellites in the SDD table. When vacancies do present themselves due to a TN delete or a category cycle, it will be recalled that in those service routines a check is made for TN waiting satellites. At those points in the program, the computer will retrieve the stored track number of the satellite chosen by TN call-up and place its data in the SDD table. Hence TN call-up requests may not be immediately observed by an operator, and unless he creates a vacancy where none existed at the time the TN call-up request was made, he will never have his request serviced.

TN Orbit - As stated in the previous subsection, when neither a TN delete nor a TN call-up code is present in the TN data word, the computer recognizes this condition as a request for a TN orbit (Fig. 4). The orbit is composed of 128 discrete position points computed in equal increments of time. The orbit is generated for plus and minus one-half the anomalistic or orbital period, P, from the time designated in the time word. A description of how time is handled is contained in a later section. The orbital period as calculated in the SUBPA subroutine is divided into 128 time increments, i.e., P/128. One-half the period is then subtracted from the time word; the resulting time is used for the first orbit point calculation. This calculation is a position prediction and thus uses the same routines as the normal updating sequence. Subsequent points are calculated by adding P/128 seconds to each previous time until the resulting time is equal to or greater than the original time word plus one-half a period. As each orbit point is computed, it is read out to the peripheral equipment, where it is stored in preparation for display. Since each point is quantized 4° in latitude by 4° in longitude, then for more than one point computed as lying within a given 4° by 4° area, just one point will appear at the area center. Thus, fewer than 128 points may be displayed.

Since the computation of the orbit points uses the SUBP1, SUBPA, SUBWD, and the SUBOD subroutines, the normal updating of satellites in the SDD table is sacrificed until the orbit points are generated. Time for the orbit generation has been measured at 5 seconds.

It might be well to note that while the computer is generating orbit points, it cannot be interrupted by a type II interrupt (see p. 2). This is not surprising, because when a type II interrupt occurs, subsequent type II interrupts are blocked until the first is terminated (unblocked). This termination takes place only after the interrupt service routine is completed and entrance to the WM or EXP display loop has been made. Therefore, when a TN interrupt (orbit) is activated, type II interrupts are blocked. Since the complete orbit is generated before entrance to the WM or EXP display loop, no type II interrupt can occur during computation of the 128 orbit points. Upon completion of the entire orbit, the computer resumes its normal sequence in either of the two display modes depending upon which loop it was in at the time the interrupt occurred.

The flow diagram of Fig. 4 shows a "TTY zero" interrogation prior to entry into the orbit routine. This particular set of instructions was added after the program had been completed in order to incorporate a last-minute operational feature. The method used is by no means the most efficient as far as programming is concerned, but it was the most expedient. The particulars concerning this will follow in the next section.

Tabular Readout Interrupt

The original plans for the operation of SPAD specified that activation of the TRACK NUMBER ORBIT pushbutton (Figs. 2a and 2b) would provide the operator with both an orbit and a teletypewriter (TTY) or page printer message. Thus the two routines were written as one. A later decision was made to provide two separate requests for these functions. Hence, a TABULAR READOUT pushbutton was provided and a unique interrupt line assigned to this operation (Fig. 4). In order to eliminate the need for extensive reprogramming, it was decided to merely set an indicator (designated TTY) when the tabular readout was requested and then permit the computer to proceed through the TN interrupt routines as before. However, now, when the TTY was checked, the computer would be steered to the appropriate section of the routine. Due to sequencing through the TN interrupt routine, the TN delete and TN call-up codes are tested even though they are obviously irrelevant for the tabular readout interrupt. Upon reaching the previously mentioned orbit and page printer message routine, the TTY indicator is interrogated. If it is not set, an orbit is generated; if it is set, the teletype message is prepared. When the entire message has been prepared, the TTY indicator is cleared.

A sample message is shown in Fig. 5. The SUBCV subroutine prepares the data into a teletype code and stores it in the TDU1 table. The position and height data are obtained in the usual manner, i.e., by the SUBP1, SUBPA, SUBWD, and the SUBOD subroutines. When this data is completed, the area of view, azimuth, and elevation quantities are computed and stored. The computer then returns to the display mode from which it was interrupted.

SPDT	CATEGORY	MO	DA	HR	MN	SC	LATIT	LONGIT	HEIGHT	SUR	AZ	EL	TN
Ø2Ø5	USA NAVG	Ø2	28	Ø1	Ø6	36	29.6N	118.3W	ØØ,634	2,767	NNN	NN	ØØ

Fig. 5 - Sample message

As was described in the section on modes of display operation, at appropriate points in these loops the SUBTD subroutine is entered. This routine tests for the possibility of there being a message waiting to be read out. If indeed one is waiting, each time the computer sequences to the SUBTD subroutine, one character code is read out to the peripheral equipment. After reading out all the data in the TDU1 table, access is made to the TDU2 table whereby the column headings permanently stored in this table are read out. In this manner, the operator is not required to wait for his data while the column headings are being printed, since the headings are already present from the last printed message. When the contents of both tables are read out, completing two lines of the message, the message waiting indicator is cleared and normal operation is resumed.

The time for the computer to sequence through the display mode loops to update the selected satellites necessarily is somewhat longer when the printed message must be interleaved with the updating cycle. However, the time difference (0.26 sec) had no observable ill effect on the display.

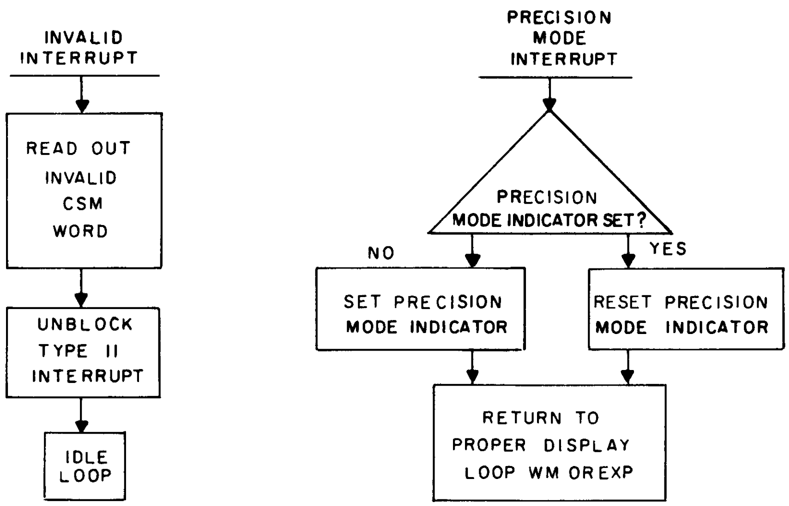


Fig. 6 - Flow of operations of an invalid interrupt and one of the operator-selected interrupts

Precision Mode Interrupt

As the computer enters the precision mode routine (Fig. 6), the precision mode indicator (PMI) is checked. This indicator keeps track of whether the computer is in a precision mode or a nonprecision mode. When the program is running in a precision mode, the PMI is set. This interrupt resets the indicator and turns on the nonprecision mode indicator lamp. Then the program proceeds to either the WM or the EXP display loop. When the program is running in a nonprecision mode, the interrupt sets the PMI and turns off the nonprecision mode indicator lamp. The program then goes to the WM or EXP display loop.

Thus a reciprocating action between precision and nonprecision mode is effected at the push of the PRECISION SELECT button. It should be noted that when the program is in the EXP display loop, the precision mode is always used. Consequently the indicator lamp must be interpreted in this context.

Invalid Interrupts

An invalid interrupt in the SPAD sense may be defined as any interrupt signal received by the computer which is not part of SPAD operation. Whenever an invalid interrupt occurs (Fig. 6), the computer reads out an invalid computer status monitor (CSM) word. This turns on an invalid interrupt indicator lamp (not illustrated). Immediately afterward, type II interrupts are unblocked and the computer goes into an idle loop. Any valid interrupt will correct this condition and cause the computer to return to the ordinary sequence of operation.

Month and Year Interrupts

Month and year interrupt routines (Fig. 7) respond to separate interrupt lines; however the sequence of instructions performed in each are similar and in part identical. Essentially these interrupts take the interrupt code word and extract the settings of the thumbwheel switches. In the month interrupt these bits are interpreted as a month; in the year interrupt they are interpreted as a year.

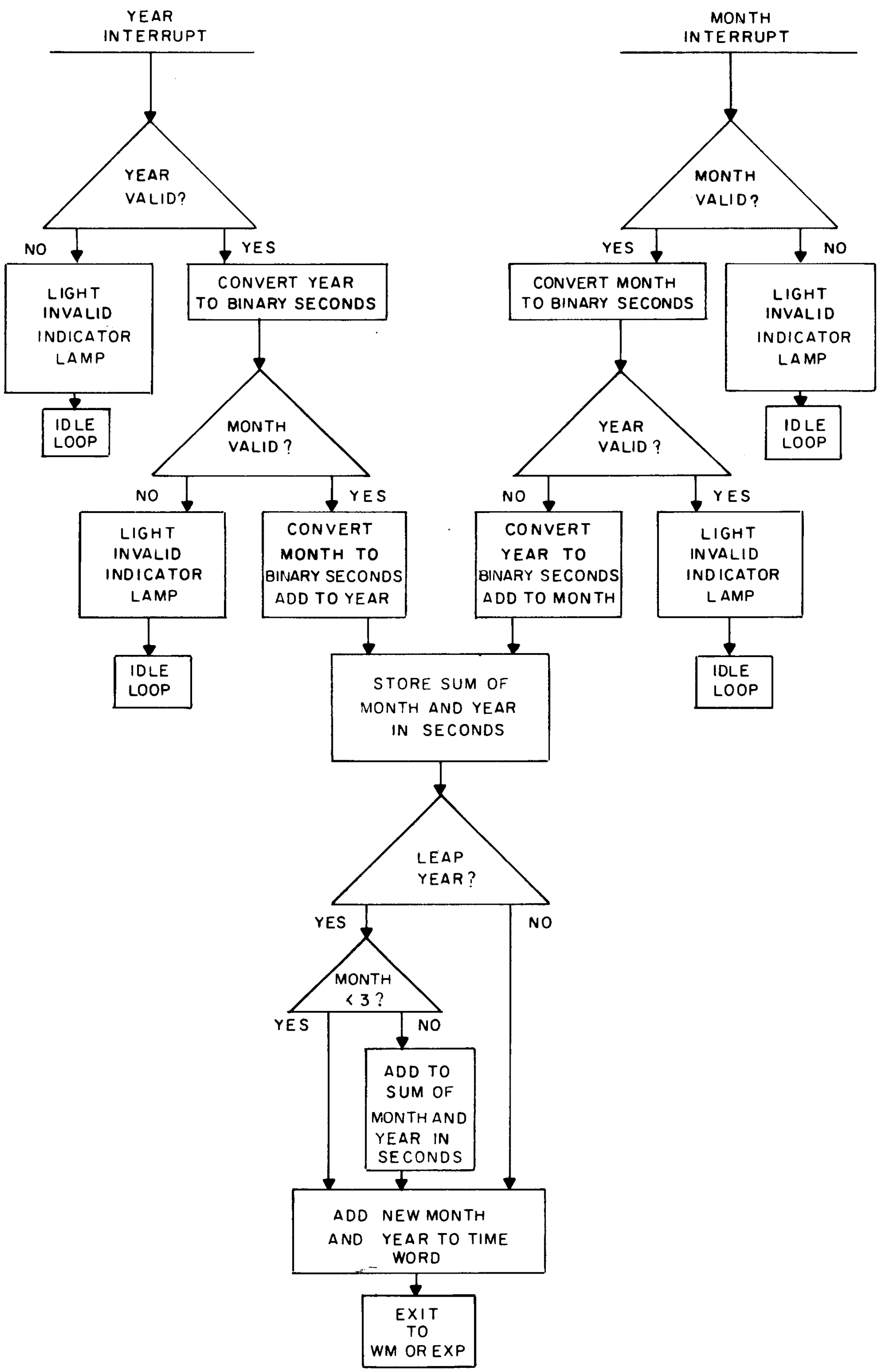


Fig. 7 - Flow of operations of two of the operator-selected interrupts

In both instances the validity of the input values is tested by the following criteria. Any number other than 01 through 12 is considered an invalid month. Any number other than 60 through 75 (for 1960 through 1975) is considered to represent an invalid year. During the year interrupt routine, if the year has been checked and found to be valid, the month is then checked for validity. Similarly during the month interrupt routine, if the month has been found valid, the year is then checked. In the event that an invalid year or month occurs, an indicator lamp is turned on and the computer goes into an idle loop. The operator may terminate this loop by entering a valid month or year as required.

One might have noticed the apparent duplicity of checking in the above sequences. The reason for this duplicity is that the invalid indicator lamp mentioned throughout this section makes no distinction between invalid year and invalid month. Thus if both an invalid year and an invalid month happen to be present, without this double check it would be possible to correct month and still have an invalid year but the indicator lamp would not show this.

Once valid information has been entered, a conversion to binary is made and the sum of year and month in seconds is formed. From this point on, both interrupts share the same instructions. A determination is now made as to whether the prediction time involves a leap year. If not, the new word is formed and the computer enters either the WM or the EXP display loop. If there is a leap year involved, a test is performed to see if prediction time is before March. If so, the new time word is formed and the computer exits to the WM or EXP display loop. If we have passed February, an extra day is added, the new time word formed, and an exit made to the WM or EXP display loop.

Own Ship Interrupt

Information displayed by SPAD includes own ship data as well as satellite positions. The own ship interrupt (Fig. 8) is the means by which own ship information (position, heading, and speed) is fed into the computer. In an operational model, this interrupt will be activated automatically every 10 minutes. At present this sequence is initiated manually at an own ship simulator panel located at the SPAD formatter.

Before the actual interrupt routine is treated, a preliminary discussion will be helpful. SPAD was designed for installation and evaluation on board the USS Kingsport, a ship primarily used for satellite communication work. This ship was equipped with an AN/UYK-1 computer, which therefore was chosen as the SPAD computer. A buffer, interfacing the ship's navigational stores with the computer, set the form of the own ship interrupt. This particular buffer is called the Bendix commutator by the manufacturer and will be referred to as such. Choosing the SPAD computer to be the same as that on board the ship would allow this one computer to be shared between SPAD and other shipboard functions. This also resulted in existing cable assignments with which SPAD had to comply. For example, the A cable carrying ships position and heading would always be connected to the ships navigational stores through the Bendix commutator for both SPAD and other ship functions. This had to be considered when writing the computer program.

During the normal course of operation the B cable consisting of B_{in} and B_{out} cables is connected between the computer and SPAD. The B_{out} cable goes to the SPAD formatter and is the means through which data is sent to the display. The B_{in} cable runs from the computer control panel to the computer. Through this cable comes information which influences computer operation and consequently the SPAD display. In the computer control panel, a thirty-bit register is connected to the B_{in} cable. The way in which the computer interprets the contents of this register depends on the particular interrupt activated. When a type II interrupt occurs, the state of this register is placed in memory location WSIC and the computer begins to execute the interrupt.

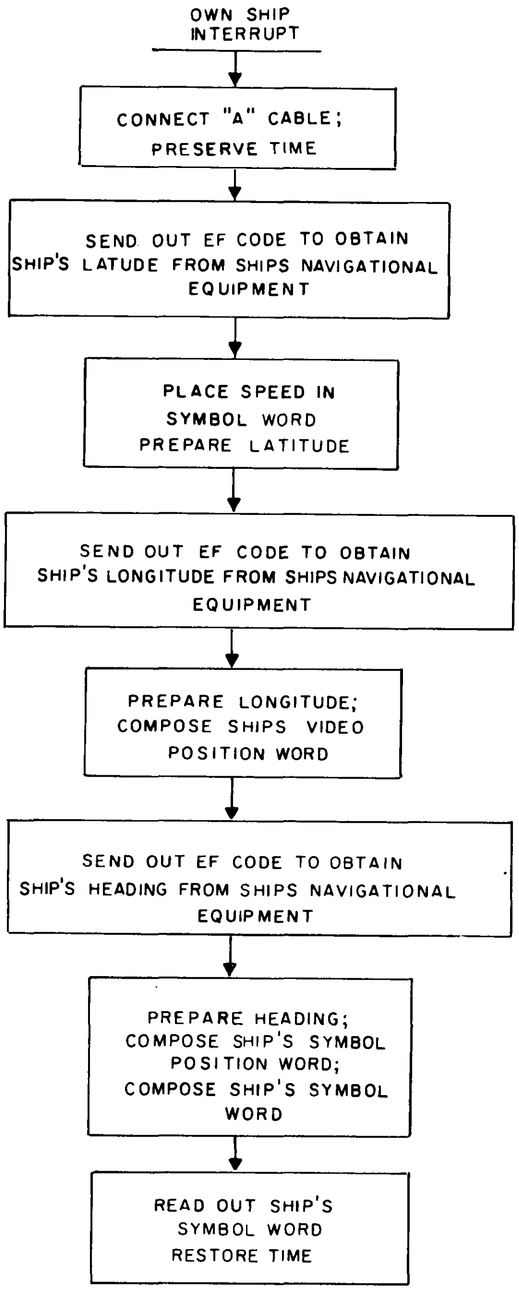


Fig. 8 - Flow of operations of one of the operated-selected interrupts

Ship's speed, which is not available from the Bendix commutator, must be taken directly from the ship's navigational stores. When an own ship interrupt is activated, ship's speed is gated into the register in the computer control panel and thus reaches the computer via the B_{in} cable. This information is then transferred to WSIC for later use. The actual interrupt routine will now be discussed with reference to Fig. 8.

The first instruction in the interrupt routine connects the computer to the A cable. The computer now has access to latitude, longitude, and heading. Ship's speed has already been placed into the computer memory.

At this point, the time word in memory location WTCD is stored away for reasons which will be discussed later in this section. The own ship interrupt now goes on to obtain latitude, longitude, and heading. To acquire each of these, a certain sequence is followed. As shown on the modified flow diagrams, an EF code must be sent requesting each of these. Since there is a slight delay between initiation of the EF code and the time the information appears on the lines(A_{in} cable), the program is written so that other processing occurs which does not involve the requested material. This is to insure that when the information is needed it will have had time to appear.

For example after the computer requests latitude, speed, which is held in memory location WSIC is processed and assembled as part of the own ship symbol word. By this time, latitude is available and has been stored in a suitable working location.

In like manner the computer requests longitude. Immediately after this it processes the latitude, which has been stored, and goes on to process longitude. Latitude and longitude are then combined to form the ship's video position word. This word is then used during WM or EXP mode to update the display.

Finally, ship's heading is requested. In the interim, the ship's symbol position word is composed. This word is also used during WM or EXP mode to update the display. It should be noted that while the ship's video position word and the ship's symbol position word are identical for either the WM or the rectangular expanded modes, they differ for the polar expanded mode. In the polar expanded mode, video is painted in a polar sweep while the symbols are still painted in rectangular coordinates.

The ship's heading, which has arrived by this time, is now placed in the ship's symbol word, which the computer then sends to the formatter (described in Ref. 2). The time word which has been stored away is now restored, and the program returns to either the WM or EXP display loop.

At this point consider the reasons for storing the time word. When the computer is connected to the B cable, time is read in by a type I interrupt. The service routine places this data into memory location WTCD. When the A cable is connected, this same mechanism is used to take own ship information into the computer.

Assume the A cable has been connected. The computer transmits a particular code to the Bendix commutator for some information, say, latitude. After a slight delay, the requested information is available. The commutator then generates a type I interrupt. The interrupt service routine takes this information places it in memory location WTCD, and returns to continue the own ship interrupt. In the same manner, longitude and heading are obtained.

If the program were now to return to either the WM or EXP display loop, it would be entirely possible that the contents of memory location WTCD (which at this point contains heading) could be interpreted as a time word for up to 1 second. If the display is operating in a clock time mode (i.e., in real time), a new time word is read in each second. Thus, if the program returned to either the WM or the EXP loop right after a time word has been read in, then the ship's heading word in memory location WTCD could be interpreted as time for up to 1 second until the actual time word came in. To eliminate this contingency, the time word which was stored away is returned to the WTCD memory location just before the interrupt returns the program to normal operation.

NONOPERATOR SELECTED INTERRUPTS

The list of interrupts covered in the preceding section encompassed all SPAD interrupts save one. This sole nonoperator selected interrupt is time. This interrupt is not complex but performs the important task of supplying the computer with time information used to compute positions. Simplicity of this interrupt precluded the use of any diagram. A discussion of events from a program viewpoint follows.

Time is input to the SPAD prediction computer by means of a type I interrupt generated on the B_{in} cable. The frequency with which the interrupt is generated depends on whether manual or clock time has been selected. For clock time, an input data request (IDR) is generated each second. For manual time, an IDR is generated four times each second. In each case, whenever an IDR is generated, a time word is placed on the computer's B_{in} cable by the computer control panel. This time word consists of day, hour, minute, and second information expressed in BCD form.

When the type I interrupt is sensed, the interrupt service routine takes the time word from the B cable and merely places it into memory locations WTCD in this BCD form. The computer program is then resumed at the point where it was interrupted.

At critical points in the program, i.e., just before a computation, time is tested by the SUBT routine. First, time is checked to see if there is a change. If there is no change, the old time stored in memory location WCRT is used. If there has been a change, a conversion from minutes, hours, etc., to seconds is performed and the new time stored in memory location WCRT. The position computation is then made on the basis of this new time.

CONCLUSIONS AND RECOMMENDATIONS

The computer program is an integral part of SPAD and was tailored to fit SPAD needs and requirements. Recent testing and evaluation have shown that the computer program is capable of performing all tasks required of it. Any major revisions in the SPAD program would probably result only from a change in basic requirements or philosophy.

On the basis of experience with the present program the following conclusions and recommendations have been made.

Accuracy and Speed

Accuracy is defined as the difference between subsatellite positions predicted by SPAD and the position given by more sophisticated methods.* Speed is defined as the time required to update the display. The allowed position error for SPAD was 1 degree in latitude and longitude, nine days from the time of orbital element updating. The speed had to be such that negligible satellite motion occurred between updating of the position stores. As has been stated above, the requirements were adequately met. The particulars may be found in Ref. 3.

The above accuracy requirements were tentatively chosen, since no operational requirements exist. Should further testing reveal that greater accuracy is needed, the program will have to be modified. This modification would involve major revision of the position computation portions (SUBP1 and SUBP2) of the program to include more sophisticated equations. These equations are available, but their incorporation into the SPAD program would greatly increase the length. Thus more memory would be needed to contain the modification. A convenient way to handle these problems would be to use the AN/UYK-3 computer, which for programming purposes is compatible with the AN/UYK-1. The program could be easily adopted to this machine, which has twice the memory capabilities with three times the speed.

Changes in Philosophy

The executive portions of the program will remain adequate only so long as there are no changes in the computer control panel functions or the overall philosophy. The display is very flexible in both the type of information presented and the manner in which the operator can manipulate display contents.

Since there were no operational requirements for a shipboard satellite display before SPAD, it is possible that a comprehensive operational evaluation would reveal further requirements. Thus some functions may be deleted while others may be added. The state of the art has moved ahead, so that a second generation SPAD would probably be very different from the present version.

For example, it now appears feasible to omit the TV raster from the display and paint all information by means of a random or jump scan, with significant savings in circuitry (4). Another possibility is the use of equipment and displays already on ship, such as the Naval Tactical Data System (NTDS). This would reduce SPAD to primarily a software package (i.e., a program tape). These possibilities, now under study, would involve major changes in the SPAD computer program.

*Such as those used at the NRL Research Computation Center.

Optimization and Repackaging

The SPAD computer program evolved into its present form as it was being written. During this process many changes were made, and memory space was left to include further modifications which might arise. Also, as the program was being written, experience was gained which allowed routines to be written more efficiently. Consequently while the program performs well, certain small portions might well be rewritten. Therefore, with repackaging and rewriting, it should be possible to squeeze the program into less space and in so doing obtain more room for raw satellite data, or perhaps other functions.

Input Data

Testing was carried out using orbital element data prepared by SPAD personnel. In the event that SPAD becomes operational, this would not be the most convenient arrangement. The original plans called for data to be converted to the proper format and relayed to the ship. Thus memory space and computation time at the user site would be saved for more important matters.

REFERENCES

1. Talmadge, H.G., Jr., and Orsino, R.J., "Operation and Performance of the NRL Satellite Position Display," NRL Report 6219, Aug. 1965

2. Orsino, R.J., and Boller, J.R., "SPAD Formatter: The Buffer Interface Between the Satellite Prediction Computer and the Display," NRL Report 6431, Oct. 1966

3. Orsino, R.J., and Francavilla, T.L., "Simplified Prediction Equations for the NRL Satellite Position Display," NRL Report 6396, Aug. 1966

4. Boller, J.R., "A Random-Scan Display of Predicted Satellite Positions," NRL Report 6453, Oct. 1966

Appendix A

GLOSSARY

Since many terms and symbols used in this report are peculiar to SPAD, they are assembled in this glossary as a convenience to the reader. Each entry is also explained where it first appears in the body of the report.

AN/UYK-1	Digital Computer. The civilian counterpart TRW-230 was used in the evaluation at NRL.
BCD	Binary Coded Decimal.
CAO	Category Overload. This refers to a word in the memory as well as to a lamp on the computer control panel (pushbutton switch); it refers to those categories which the computer has not gotten to yet because of the ten-satellite limit. The lamp gives this indication to the operator.
CAR	Categories Requested by the operator. This refers to a word in the memory as well as to lamps on the computer control panel. The word in the memory contains ones in positions of categories which have been selected. The lamps give this indication to the operator.
CAS	Category being Searched. This refers to a word in the memory as well as to a lamp on the computer control panel. It designates one category in which some satellites are being updated but, due to the ten-satellite limit, in which a search is continuing for satellites. The lamp gives this indication to the operator.
CF	Control Function. This is used by the computer to perform the following at the choice of the programmer: Connect the required cable combinations, clear the parity indicator, input the interrupt status word to the E register, etc.
CF Command	Control Function Command. This is a machine language command. See the machine language reference manual.
CSM	Computer Status Monitor. This is a control word to light lamps as indicated by the computer.
EF	External Function. This is used by the computer to signal the external device that a word present on the output lines is to be interpreted as a command or control word as opposed to a data word.
EF Command	External Function Command. This is a machine language command.
EXP	Expanded. This is used to designate the display mode or the program updating loop. (It refers to either the polar expanded or the rectangular expanded mode.)

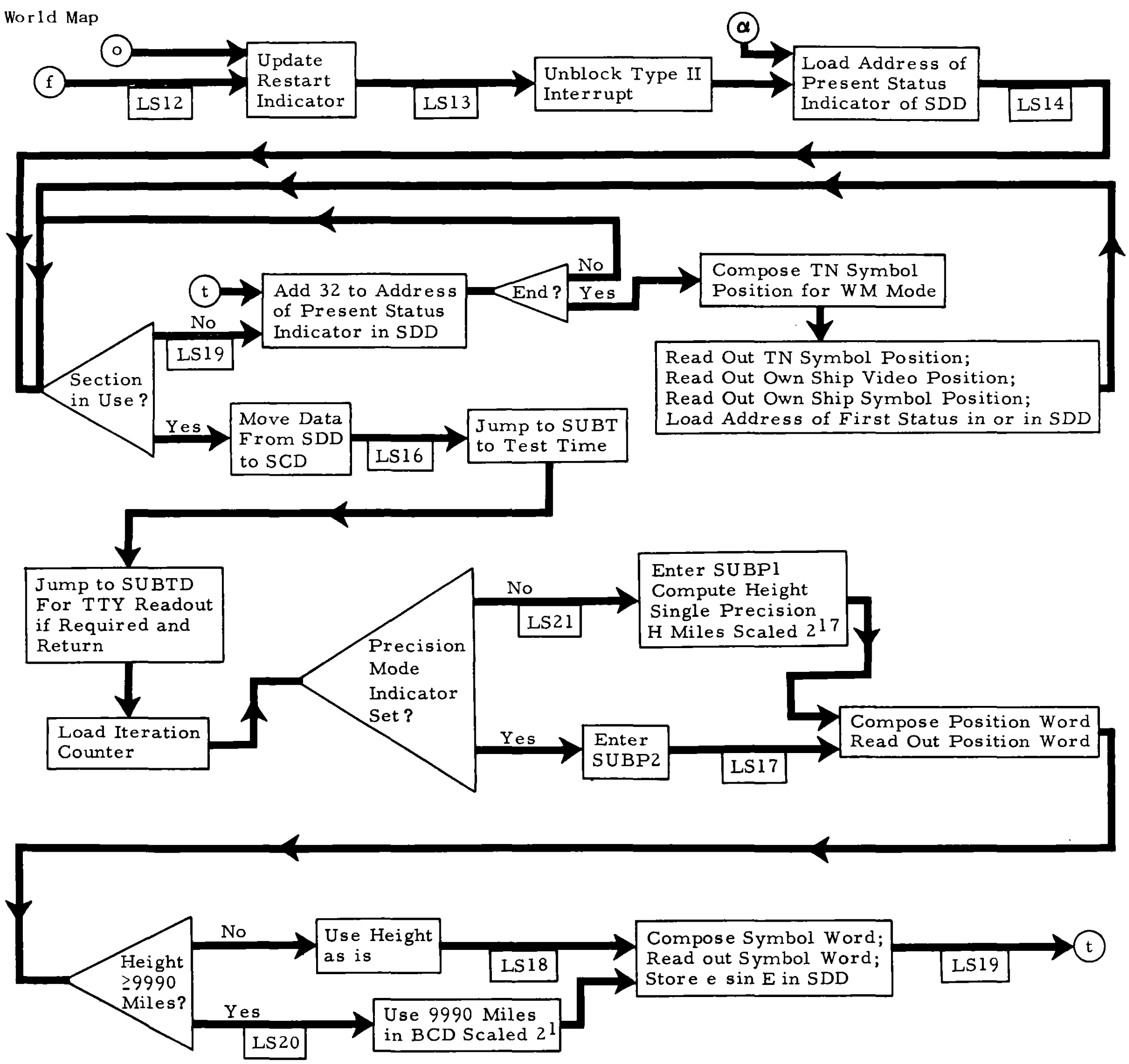
IDR	Input Data Request. This is used by an external device, electrically connected, to signal the computer that data is present on the input lines.
I/O	Input/Output section of the computer.
Input Acknowledge	This is used by the computer in response to either an IDR or an interrupt, to signal an external device that input data has been sampled or that the interrupt has been executed.
Interrupt Line	Used by an external device, not electrically connected to data lines, to interrupt the computer.
Interrupt: Type I	Interrupt of the computer via IDR or ODR lines by a device already electrically connected.
Interrupt: Type II	Interrupt of the computer by an interrupt line. The computer determines which device caused the interrupt, connects it, and services the request.
Logand	Logic Command. This is a single machine-language instruction.
Logram	Logic Program. This is composed of many logands.
NTDS	Navy Tactical Data System.
ODR	Output Data Request. This is used by an external device to signal the computer that it is ready to accept an output word.
Output Acknowledge	This is used by the computer (in response to an ODR) to signal an external device that data is present on the output lines.
PMI	Precision Mode Indicator. This refers to a location in the memory which is set to one or zero depending on whether the precision mode has been requested.
SCD	Satellite being Computed Data. Table holds data for one satellite at a time from SDD. Data held in SCD while particular satellite is being updated.
SDD	Satellite Displayed Data table. This is a table where data for up to eleven satellites are stored. As many as ten of these may be selected by category and one by track number. These are the eleven satellites displayed during WM. The format for each satellite is modified as it is taken from SRD and placed in SDD by SUBMV.
SPAD	Satellite position Prediction And Display equipment. This is the acronym used to describe the NRL satellite display.
SRD	Satellite Raw Data table. This table contains orbital elements and ancillary information on the entire list of satellites (88) in the store.
SUBCV	Subroutine: Converts a BCD character to TTY code and stores it in the TDU1 output list.
SUBMV	Subroutine: Searches for a vacancy in SDD. When a vacancy is located, raw data is moved from SRD to temporary storage. Suitable calculations are performed. The resultant data are stored in the vacancy.

SUBOD	Subroutine: Used to compute the time rate of change of the longitude of the ascending node.
SUBP1	Subroutine: Single precision position computation. It makes use of SUBPA, SUBWD, SUBOD. It iterates Kepler's equation only twice.
SUBP2	Subroutine: Double precision position computation. It makes use of SUBPA, SUBWD, SUBOD. It iterates Kepler's equation either 64 times or until M is accurate to 1.19×10^{-7} revolutions.
SUBPA	Subroutine: Computes the average period from the period at epoch and the period at prediction time. The average period is then used to compute the mean anomaly.
SUBPX	Subroutine: Polar Expanded. It takes into consideration the expanded area of the target and positions the satellite on the display face (polar mode). It is used for latitudes greater than 75°N or 75°S.
SUBRX	Subroutine: Rectangular Expanded. It takes into consideration the expanded area of the target and positions the target on the display face (rectangular mode). It is used for latitudes between 75°S and 75°N.
SUBSC	Subroutine: Computation for the radius of the area of view.
SUBT	Subroutine: Takes the time word from the display clock and converts it to seconds. The day is also checked for validity.
SUBTD	Subroutine: Governs tabular readout.
SUBWD	Subroutine: Used to compute the time rate of change of the argument of perigee.
TDU1	Table containing the data line of a tabular readout message.
TDU2	Table containing the heading line of a tabular readout message.
TN	Track Number.
TTY	Indicator which is set whenever a tabular readout is requested. It is used on occasion to stand for teletype.
TYPE I Interrupt	See Interrupt: Type I
WCRT	Position time in binary seconds since 1960. It also refers to a location in the memory containing the prediction time.
WM	World Map. It is used to designate the display mode, or the program updating loop.
WSIC	Designates a type II interrupt data word. It is also the memory location containing the word put in by a type II interrupt.
WTCD	Clock time word. It is also the memory location containing the word put in by a type I interrupt.

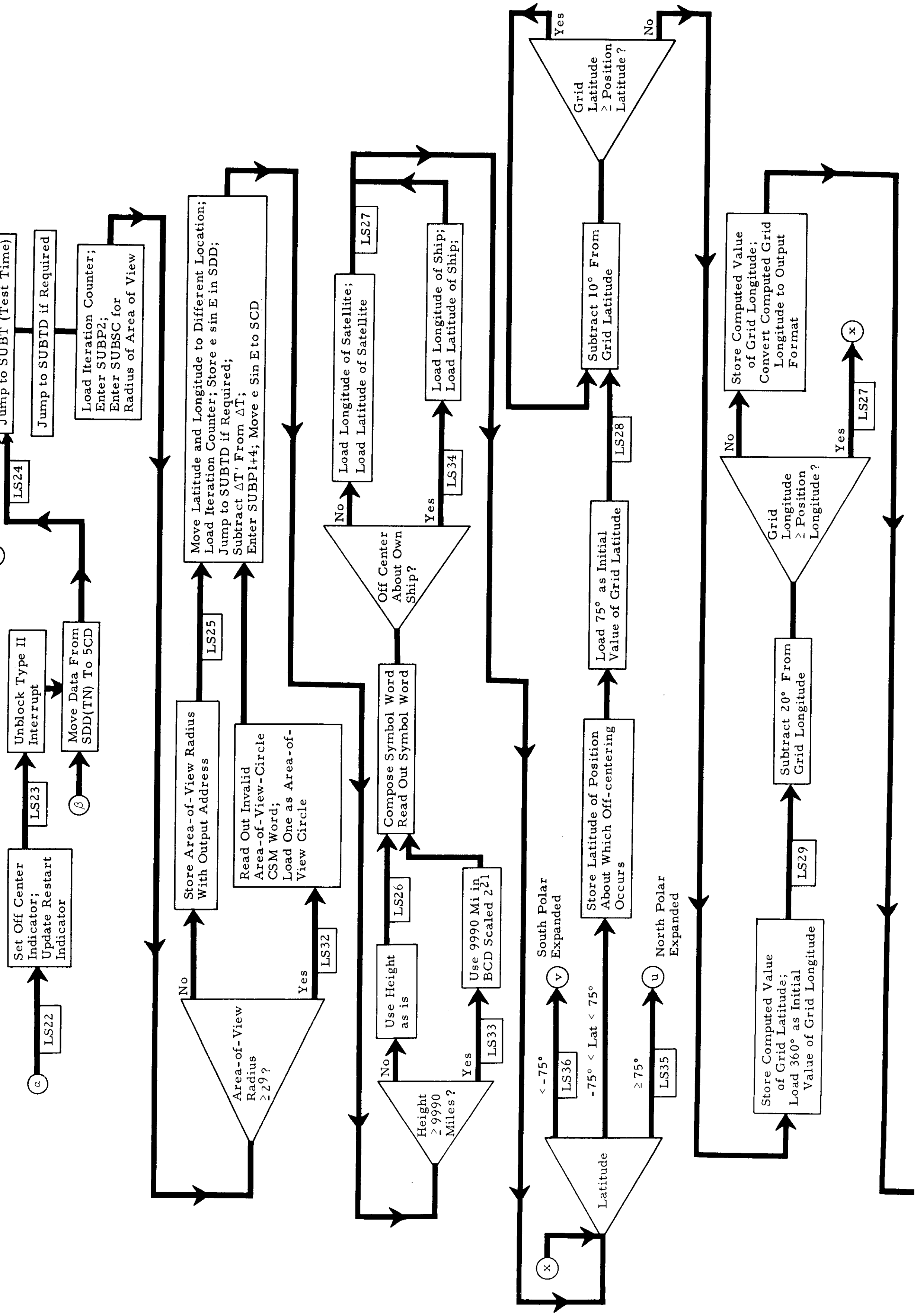
Appendix B

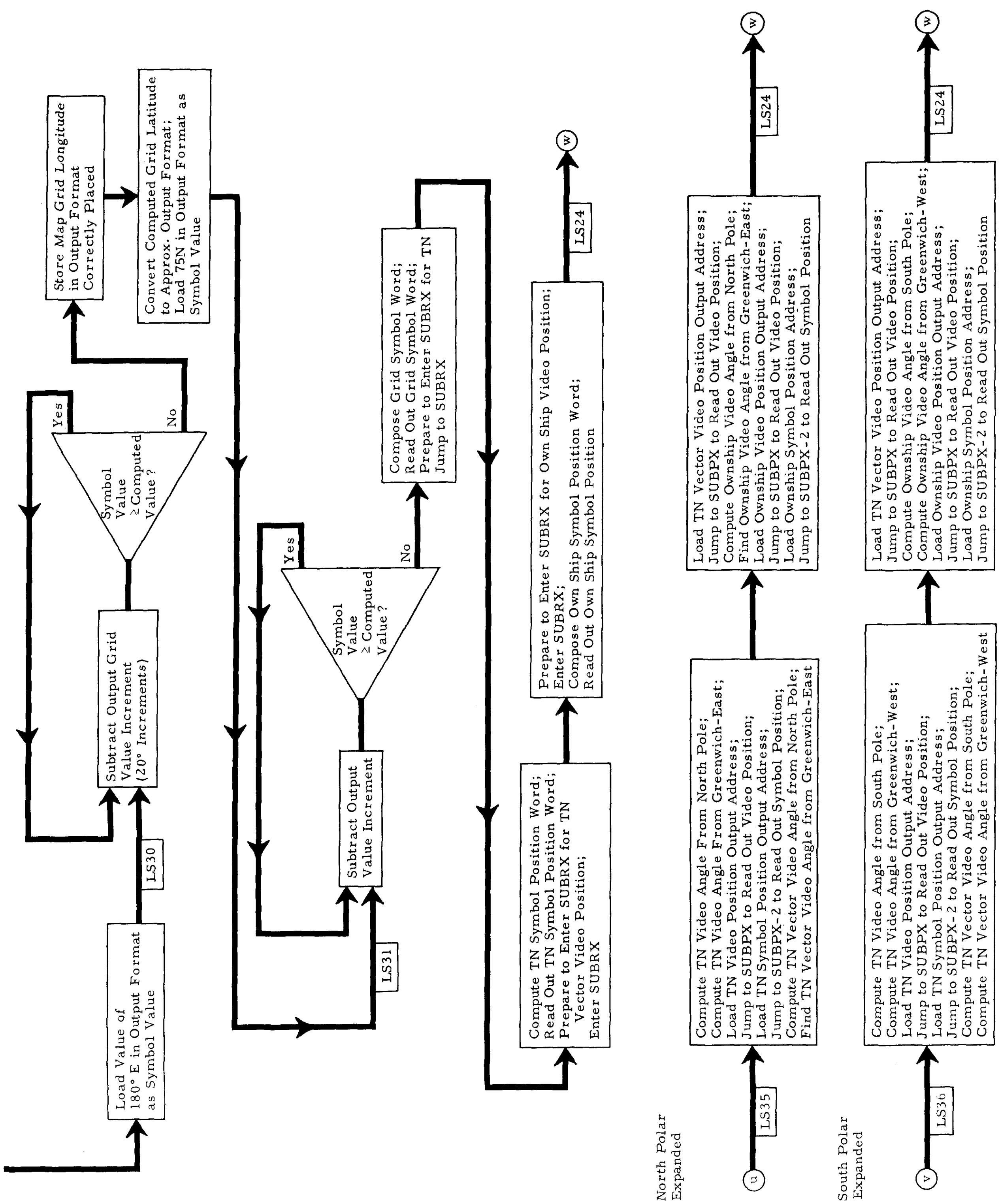
COMPLETE PROGRAM FLOW CHARTS

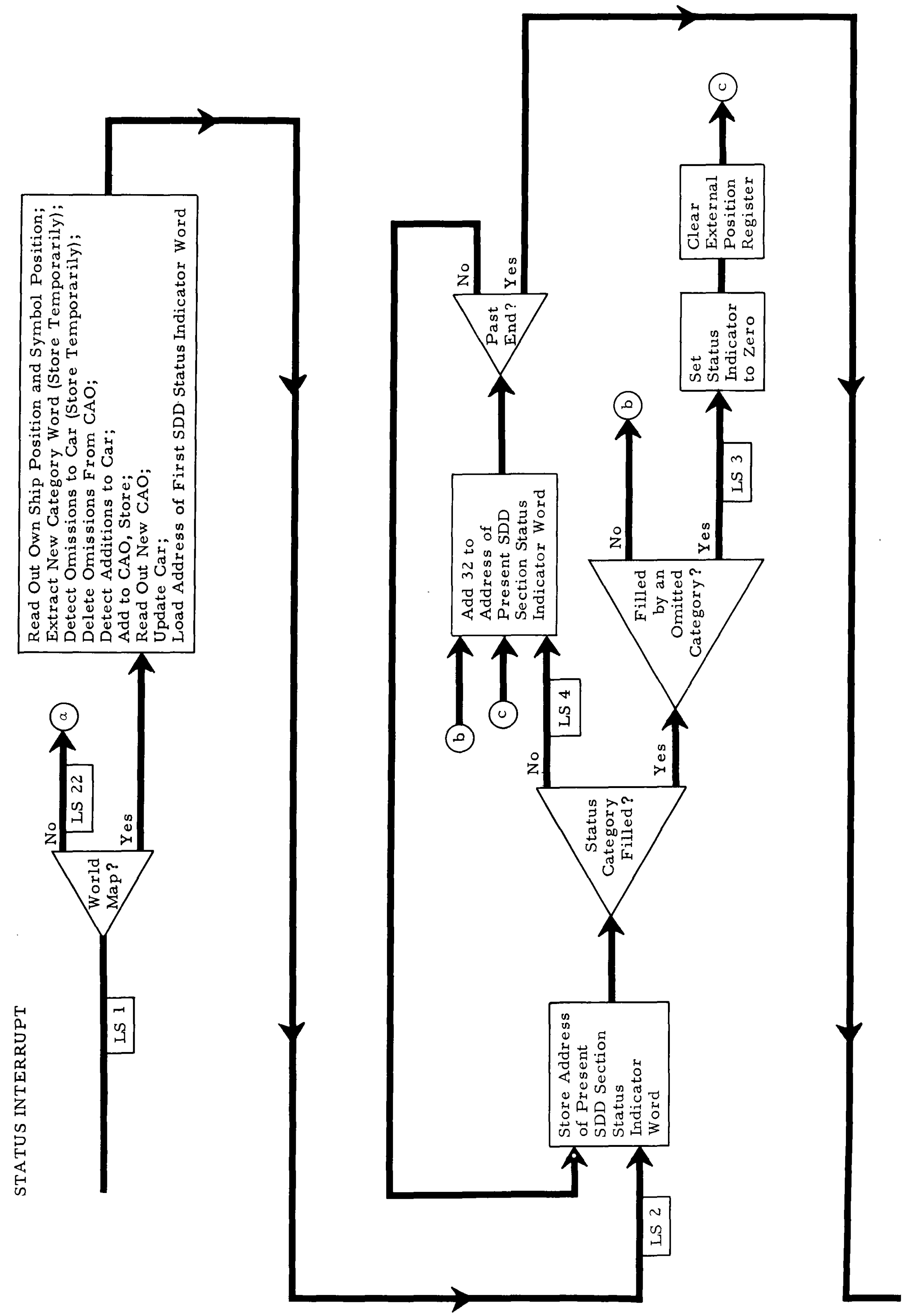
The notation that has been followed for the flow diagrams was that suggested by Grabbe, Ramo, and Wooldridge.* Triangular boxes show decisions, square boxes, enclose operations, and circles point out jumps to other portions of the program with the corresponding symbol. Small squares with LS and a number designate certain locations in the program.

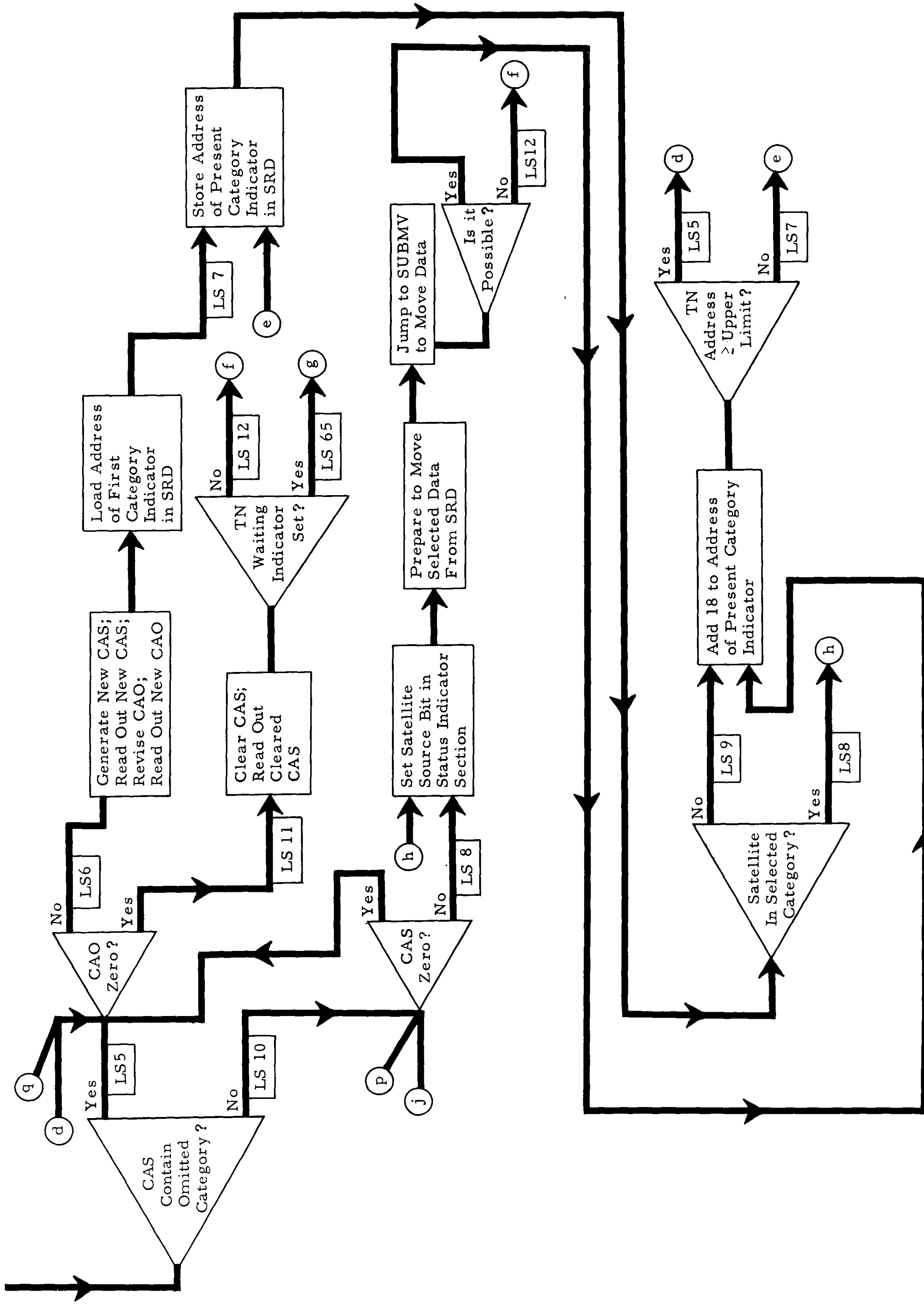


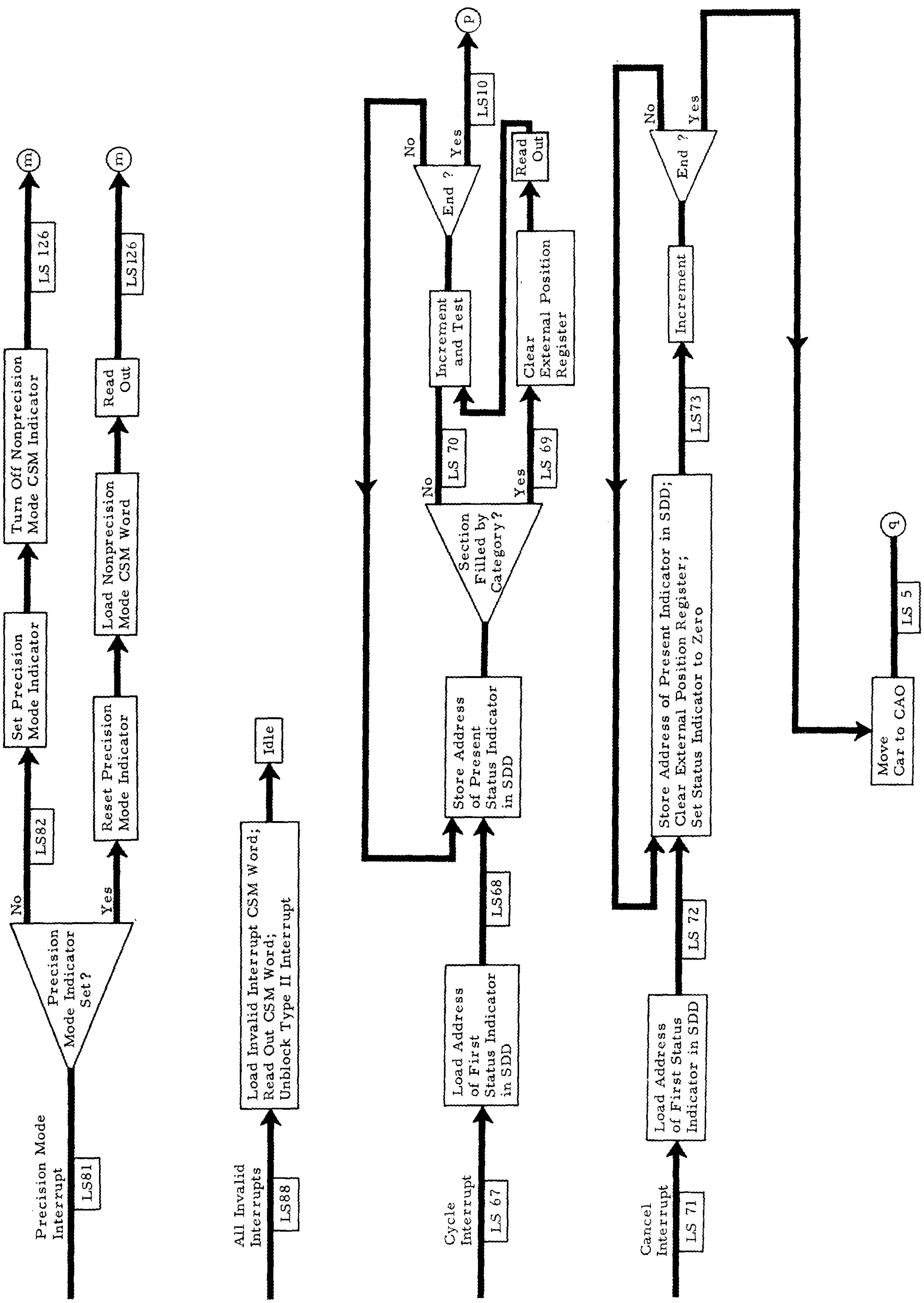
*E.M. Grabbe, S. Ramo, and D.E. Wooldridge, "Handbook of Automation Computation and Control," vol. 2, New York:Wiley, 1959.

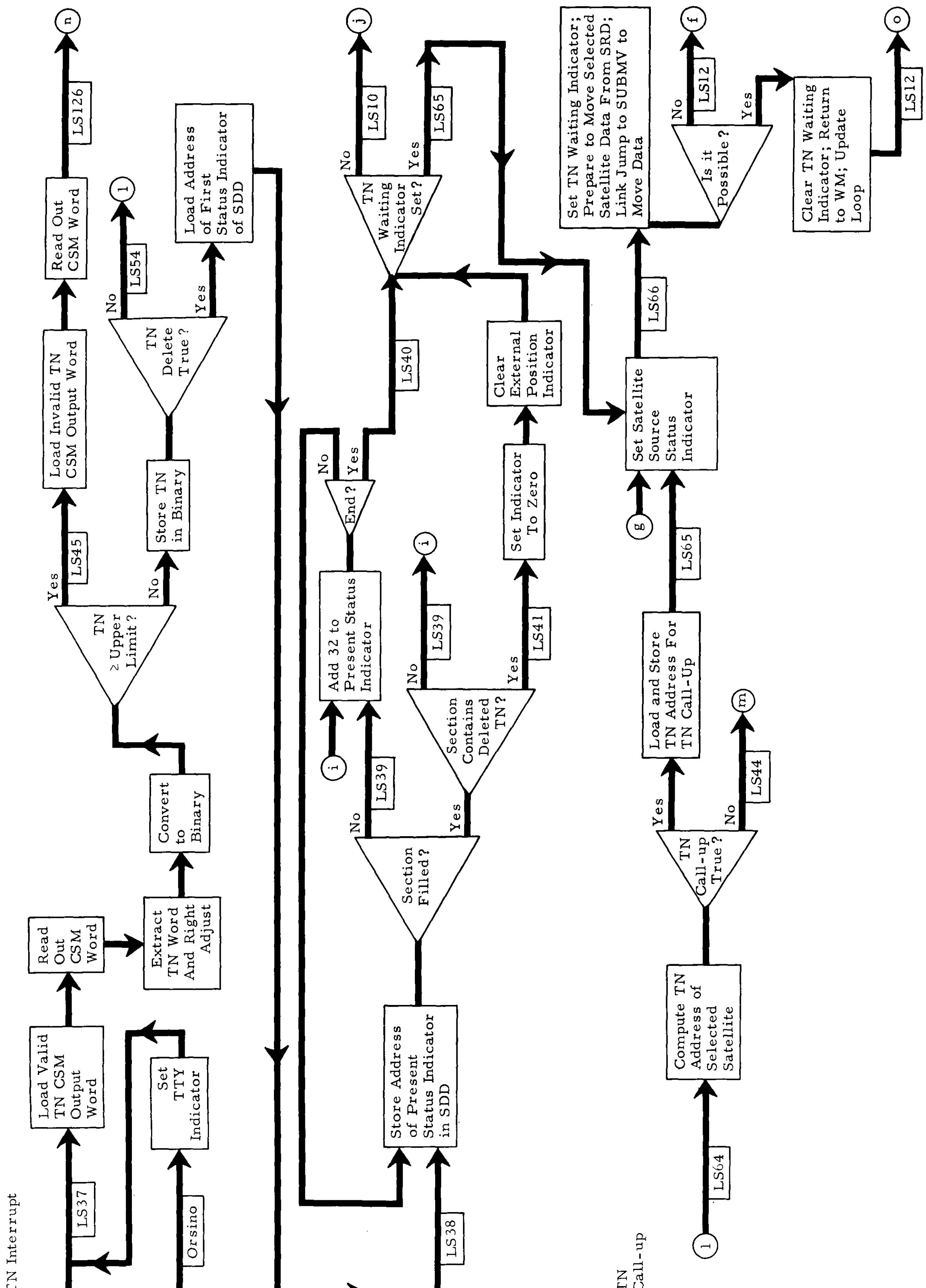

α
LS22
Set Off Center Indicator; Update Restart Indicator
LS23
Unblock Type II Interrupt
β
Move Data From SDD(TN) To 5CD
w
LS24
Jump to SUBT (Test Time)
Jump to SUBTD if Required
Load Iteration Counter; Enter SUBP2; Enter SUBSC for Radius of Area of View
Area-of-View Radius ≥ 29?
No
Yes
Store Area-of-View Radius With Output Address
LS25
LS32
Read Out Invalid Area-of-View-Circle CSM Word; Load One as Area-of-View Circle
Move Latitude and Longitude to Different Location; Load Iteration Counter; Store e sin E in SDD; Jump to SUBTD if Required; Subtract ΔT' From ΔT; Enter SUBP1+4; Move e Sin E to SCD
Height ≥ 9990 Miles?
No
Yes
Use Height as is
LS26
LS33
Use 9990 Mi in BCD Scaled 2^21
Compose Symbol Word Read Out Symbol Word
Off Center About Own Ship?
No
Yes
Load Longitude of Satellite; Load Latitude of Satellite
LS27
LS34
Load Longitude of Ship; Load Latitude of Ship;
x
Latitude
< -75°
LS36
v
South Polar Expanded
-75° < Lat < 75°
Store Latitude of Position About Which Off-centering Occurs
≥ 75°
LS35
u
North Polar Expanded
Load 75° as Initial Value of Grid Latitude
LS28
Subtract 10° From Grid Latitude
Grid Latitude ≥ Position Latitude?
Yes
No
Store Computed Value of Grid Latitude; Load 360° as Initial Value of Grid Longitude
LS29
Subtract 20° From Grid Longitude
Grid Longitude ≥ Position Longitude?
No
Yes
Store Computed Value of Grid Longitude; Convert Computed Grid Longitude to Output Format
LS27
x

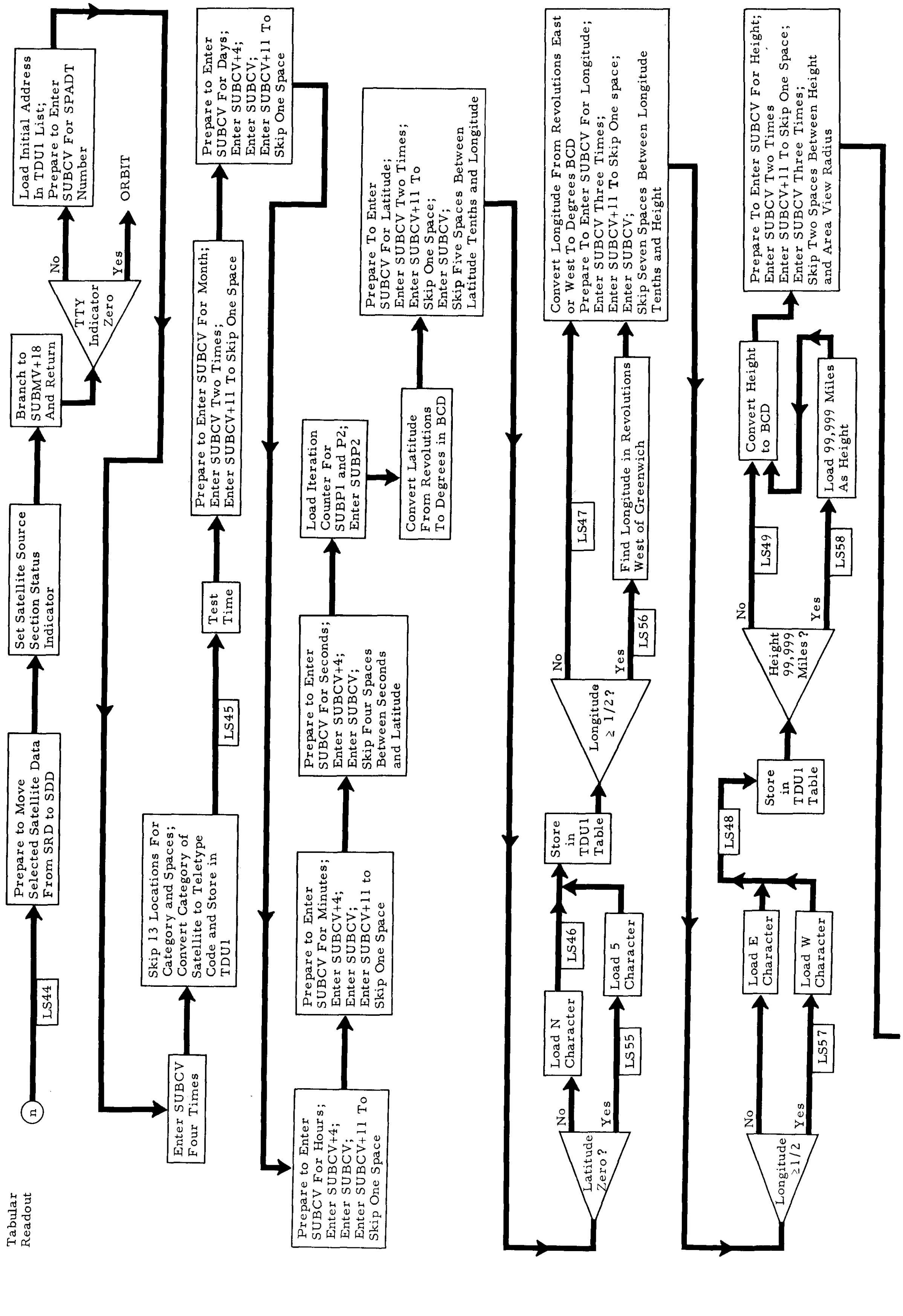

Load Value of 180° E in Output Format as Symbol Value
LS30
Subtract Output Grid Value Increment (20° Increments)
Symbol Value ≥ Computed Value ?
Yes
No
Store Map Grid Longitude in Output Format Correctly Placed
Convert Computed Grid Latitude to Approx. Output Format; Load 75N in Output Format as Symbol Value
LS31
Subtract Output Value Increment
Symbol Value ≥ Computed Value ?
Yes
No
Compose Grid Symbol Word; Read Out Grid Symbol Word; Prepare to Enter SUBRX for TN Jump to SUBRX
Compute TN Symbol Position Word; Read Out TN Symbol Position Word; Prepare to Enter SUBRX for TN Vector Video Position; Enter SUBRX
Prepare to Enter SUBRX for Own Ship Video Position; Enter SUBRX; Compose Own Ship Symbol Position Word; Read Out Own Ship Symbol Position
LS24
w
North Polar Expanded
u
LS35
Compute TN Video Angle From North Pole; Compute TN Video Angle From Greenwich-East; Load TN Video Position Output Address; Jump to SUBPX to Read Out Video Position; Load TN Symbol Position Output Address; Jump to SUBPX-2 to Read Out Symbol Position; Compute TN Vector Video Angle from North Pole; Find TN Vector Video Angle from Greenwich-East
Load TN Vector Video Position Output Address; Jump to SUBPX to Read Out Video Position; Compute Ownship Video Angle from North Pole; Find Ownship Video Angle from Greenwich-East; Load Ownship Video Position Output Address; Jump to SUBPX to Read Out Video Position; Load Ownship Symbol Position Address; Jump to SUBPX-2 to Read Out Symbol Position
LS24
w
South Polar Expanded
v
LS36
Compute TN Video Angle from South Pole; Compute TN Video Angle from Greenwich-West; Load TN Video Position Output Address; Jump to SUBPX to Read Out Video Position; Load TN Symbol Position Output Address; Jump to SUBPX-2 to Read Out Symbol Position; Compute TN Vector Video Angle from South Pole; Compute TN Vector Video Angle from Greenwich-West
Load TN Vector Video Position Output Address; Jump to SUBPX to Read Out Video Position; Compute Ownship Video Angle from South Pole; Compute Ownship Video Angle from Greenwich-West; Load Ownship Video Position Output Address; Jump to SUBPX to Read Out Video Position; Load Ownship Symbol Position Address; Jump to SUBPX-2 to Read Out Symbol Position
LS24
w


STATUS INTERRUPT
LS 1
World Map?
No
LS 22
a
Yes
Read Out Own Ship Position and Symbol Position;
Extract New Category Word (Store Temporarily);
Detect Omissions to Car (Store Temporarily);
Delete Omissions From CAO;
Detect Additions to Car;
Add to CAO, Store;
Read Out New CAO;
Update Car;
Load Address of First SDD Status Indicator Word
LS 2
Store Address of Present SDD Section Status Indicator Word
Status Category Filled?
No
LS 4
Yes
Filled by an Omitted Category?
No
b
Yes
LS 3
Set Status Indicator to Zero
Clear External Position Register
c
b
c
Add 32 to Address of Present SDD Section Status Indicator Word
Past End?
No
Yes

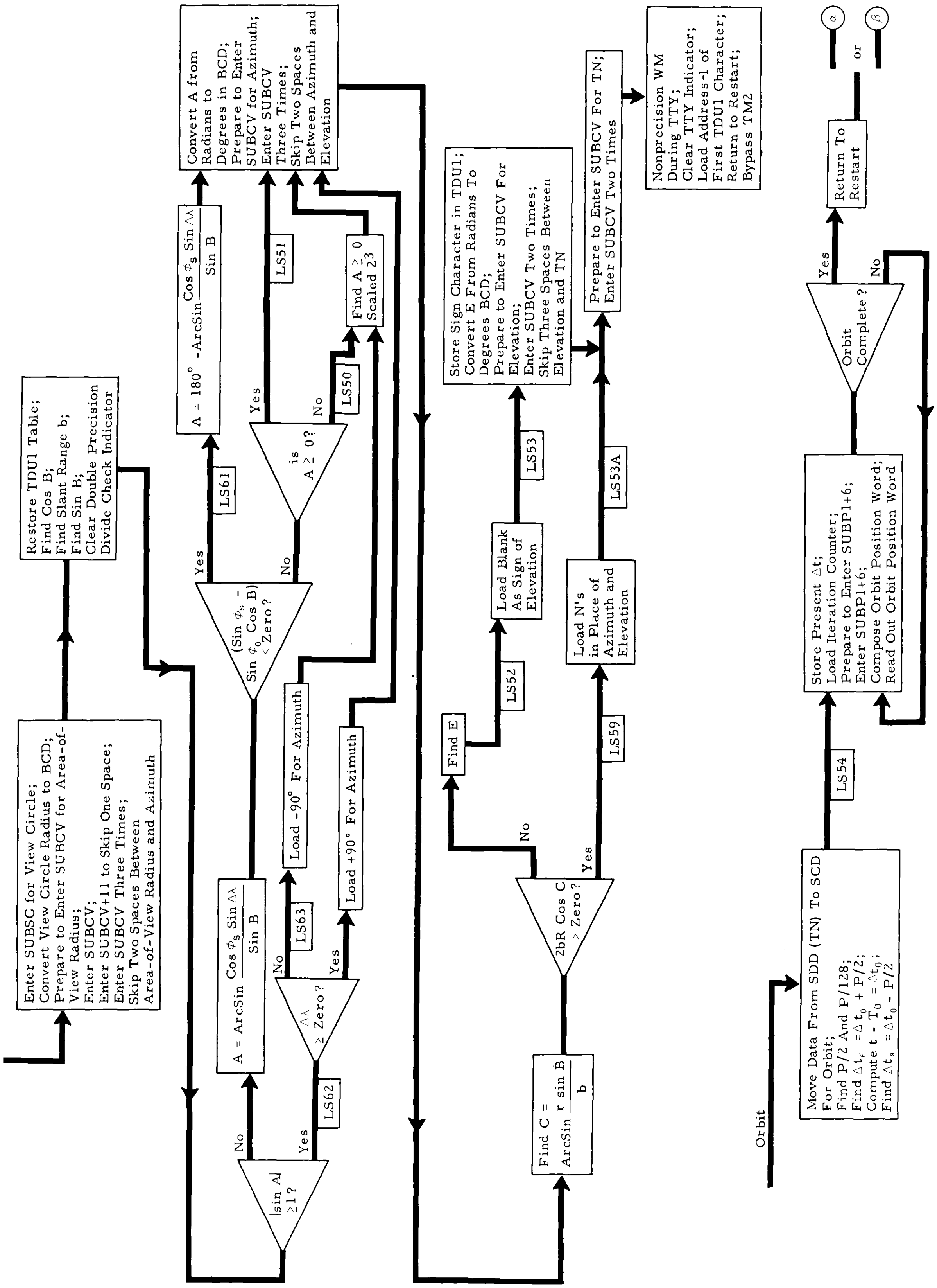

CAS Contain Omitted Category ?
Yes
LS5
No
LS 10
d
q
p
j
CAO Zero?
No
LS6
Yes
Generate New CAS; Read Out New CAS; Revise CAO; Read Out New CAO
Load Address of First Category Indicator in SRD
LS 7
Store Address of Present Category Indicator in SRD
e
LS 11
Clear CAS; Read Out Cleared CAS
TN Waiting Indicator Set?
No
LS 12
f
Yes
LS 65
g
CAS Zero?
Yes
No
LS 8
h
Set Satellite Source Bit in Status Indicator Section
Prepare to Move Selected Data From SRD
Jump to SUBMV to Move Data
Is it Possible?
Yes
No
LS12
f
Satellite In Selected Category?
No
LS 9
Yes
LS8
h
Add 18 to Address of Present Category Indicator
TN Address ≥ Upper Limit?
Yes
LS5
d
No
LS7
e


Precision Mode Interrupt
LS81
Precision Mode Indicator Set?
No
LS82
Set Precision Mode Indicator
Turn Off Nonprecision Mode CSM Indicator
LS 126
m
Yes
Reset Precision Mode Indicator
Load Nonprecision Mode CSM Word
Read Out
LS126
m
All Invalid Interrupts
LS88
Load Invalid Interrupt CSM Word; Read Out CSM Word; Unblock Type II Interrupt
Idle
Cycle Interrupt
LS 67
Load Address of First Status Indicator in SDD
LS68
Store Address of Present Status Indicator in SDD
Section Filled by Category?
No
LS 70
Increment and Test
End ?
No
Yes
LS10
p
Yes
LS 69
Clear External Position Register
Read Out
Cancel Interrupt
LS 71
Load Address of First Status Indicator in SDD
LS 72
Store Address of Present Indicator in SDD; Clear External Position Register; Set Status Indicator to Zero
LS73
Increment
End ?
No
Yes
Move Car to CAO
LS 5
q

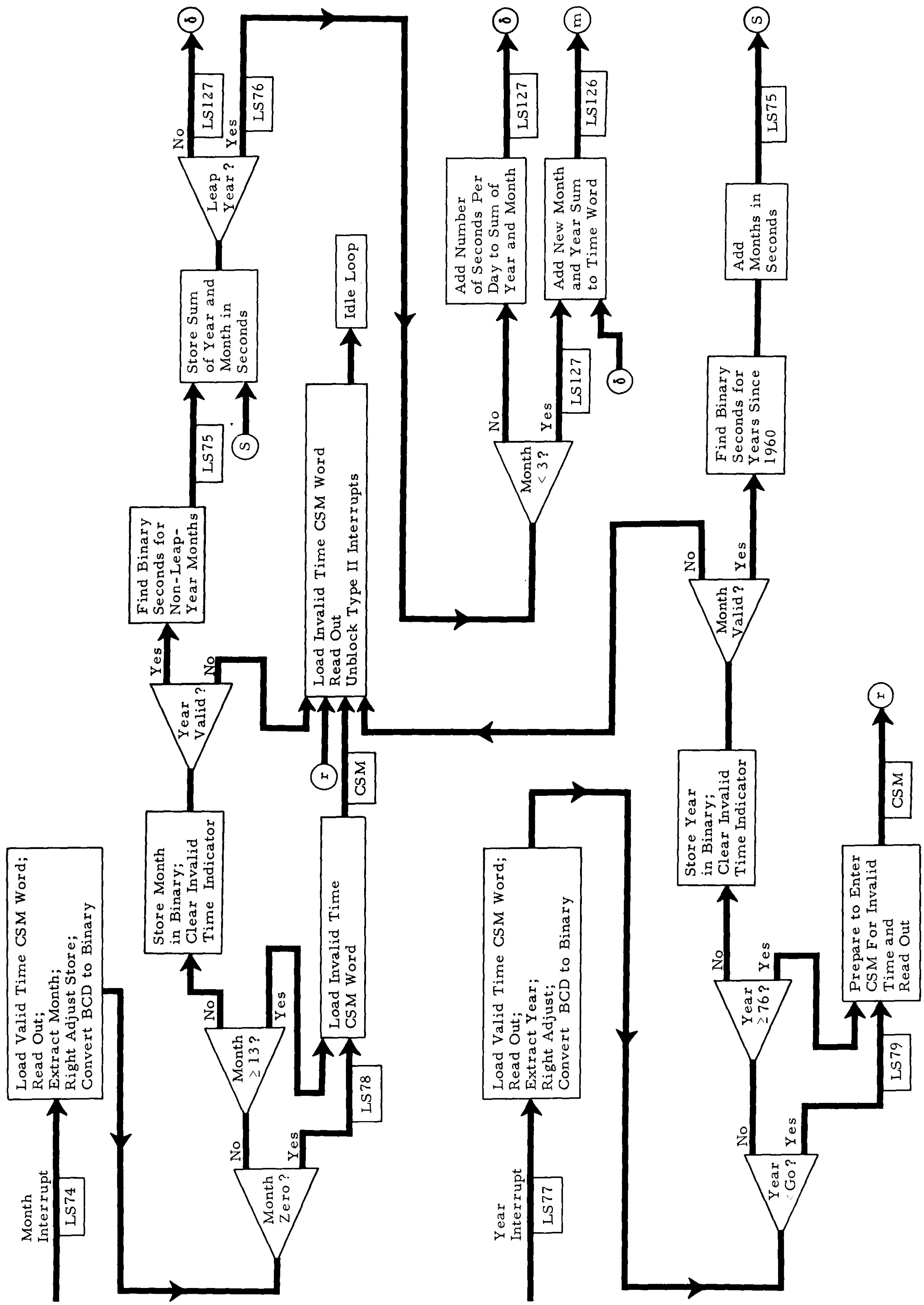

TN Interrupt
LS37
Orsino
Load Valid TN CSM Output Word
Set TTY Indicator
Read Out CSM Word
Extract TN Word And Right Adjust
Convert to Binary
TN ≥ Upper Limit?
Yes
No
LS45
Load Invalid TN CSM Output Word
Read Out CSM Word
LS126
n
Store TN in Binary
TN Delete True?
No
LS54
l
Yes
Load Address of First Status Indicator of SDD
LS38
Store Address of Present Status Indicator in SDD
Section Filled?
No
LS39
Yes
i
Add 32 to Present Status Indicator
End?
No
Yes
LS40
Section Contains Deleted TN?
No
LS39
i
Yes
LS41
Set Indicator To Zero
Clear External Position Indicator
TN Waiting Indicator Set?
No
LS10
j
Yes
LS65
TN Call-up
l
LS64
Compute TN Address of Selected Satellite
TN Call-up True?
Yes
Load and Store TN Address For TN Call-Up
LS65
No
LS44
m
g
Set Satellite Source Status Indicator
LS66
Set TN Waiting Indicator; Prepare to Move Selected Satellite Data From SRD; Link Jump to SUBMV to Move Data
Is it Possible?
No
LS12
f
Yes
Clear TN Waiting Indicator; Return to WM; Update Loop
LS12
o



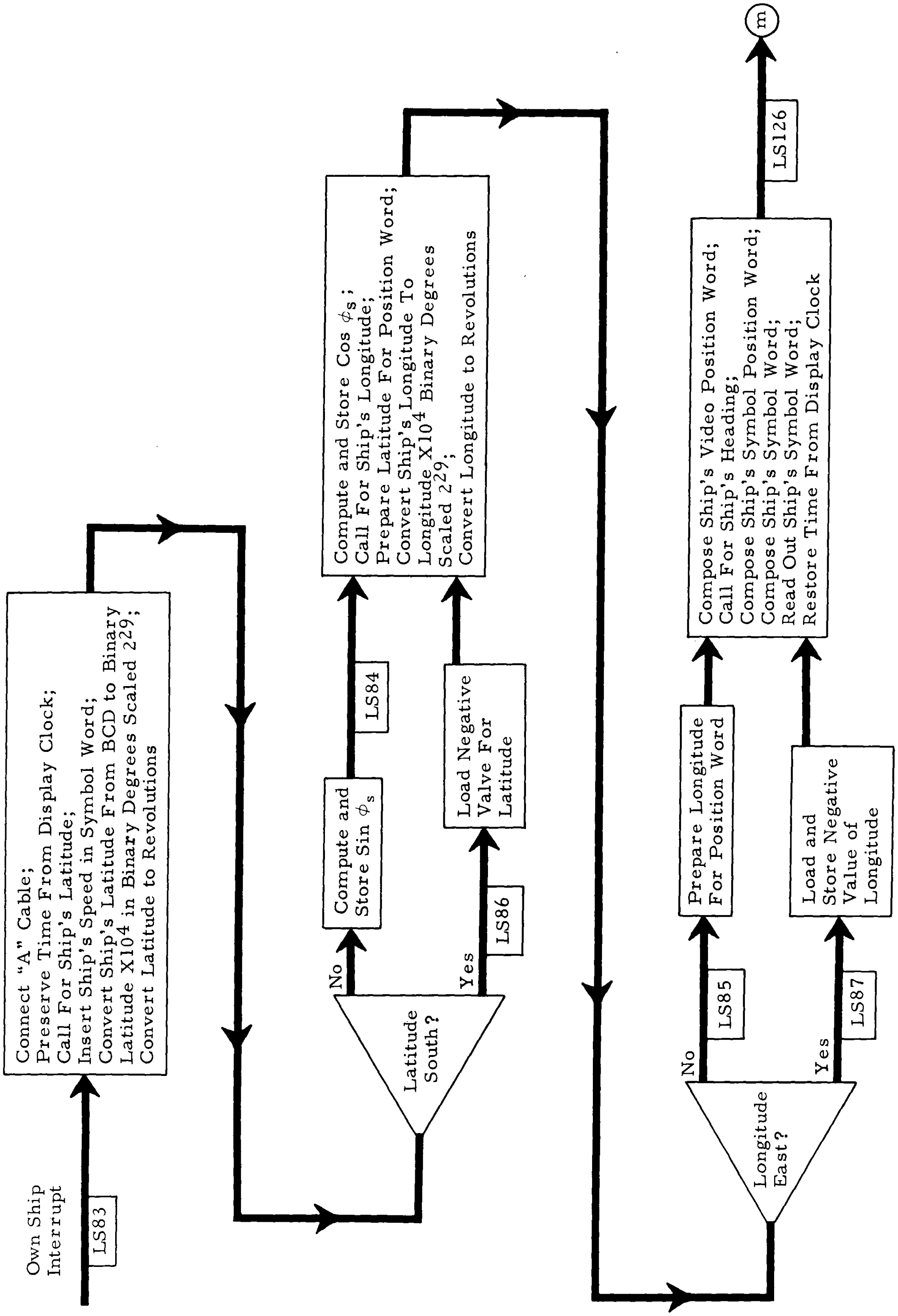
Tabular Readout
n
LS44
Prepare to Move Selected Satellite Data From SRD to SDD
Set Satellite Source Section Status Indicator
Branch to SUBMV+18 And Return
TTY Indicator Zero
No
Yes
Load Initial Address In TDU1 List; Prepare to Enter SUBCV For SPADT Number
ORBIT
Enter SUBCV Four Times
Skip 13 Locations For Category and Spaces; Convert Category of Satellite to Teletype Code and Store in TDU1
LS45
Test Time
Prepare to Enter SUBCV For Month; Enter SUBCV Two Times; Enter SUBCV+11 To Skip One Space
Prepare to Enter SUBCV For Days; Enter SUBCV+4; Enter SUBCV; Enter SUBCV+11 To Skip One Space
Prepare to Enter SUBCV For Hours; Enter SUBCV+4; Enter SUBCV; Enter SUBCV+11 To Skip One Space
Prepare to Enter SUBCV For Minutes; Enter SUBCV+4; Enter SUBCV; Enter SUBCV+11 to Skip One Space
Prepare to Enter SUBCV For Seconds; Enter SUBCV+4; Enter SUBCV; Skip Four Spaces Between Seconds and Latitude
Load Iteration Counter For SUBP1 and P2; Enter SUBP2
Convert Latitude From Revolutions To Degrees in BCD
Prepare To Enter SUBCV For Latitude; Enter SUBCV Two Times; Enter SUBCV+11 To Skip One Space; Enter SUBCV; Skip Five Spaces Between Latitude Tenths and Longitude
Latitude Zero?
No
Yes
Load N Character
LS55
Load 5 Character
LS46
Store in TDU1 Table
Longitude ≥ 1/2?
No
Yes
LS47
LS56
Find Longitude in Revolutions West of Greenwich
Convert Longitude From Revolutions East or West To Degrees BCD Prepare To Enter SUBCV For Longitude; Enter SUBCV Three Times; Enter SUBCV+11 To Skip One space; Enter SUBCV; Skip Seven Spaces Between Longitude Tenths and Height
Longitude ≥1/2
No
Yes
Load E Character
LS57
Load W Character
LS48
Store in TDU1 Table
Height 99,999 Miles?
No
Yes
LS49
LS58
Load 99,999 Miles As Height
Convert Height to BCD
Prepare To Enter SUBCV For Height; Enter SUBCV Two Times Enter SUBCV+11 To Skip One Space; Enter SUBCV Three Times; Skip Two Spaces Between Height and Area View Radius

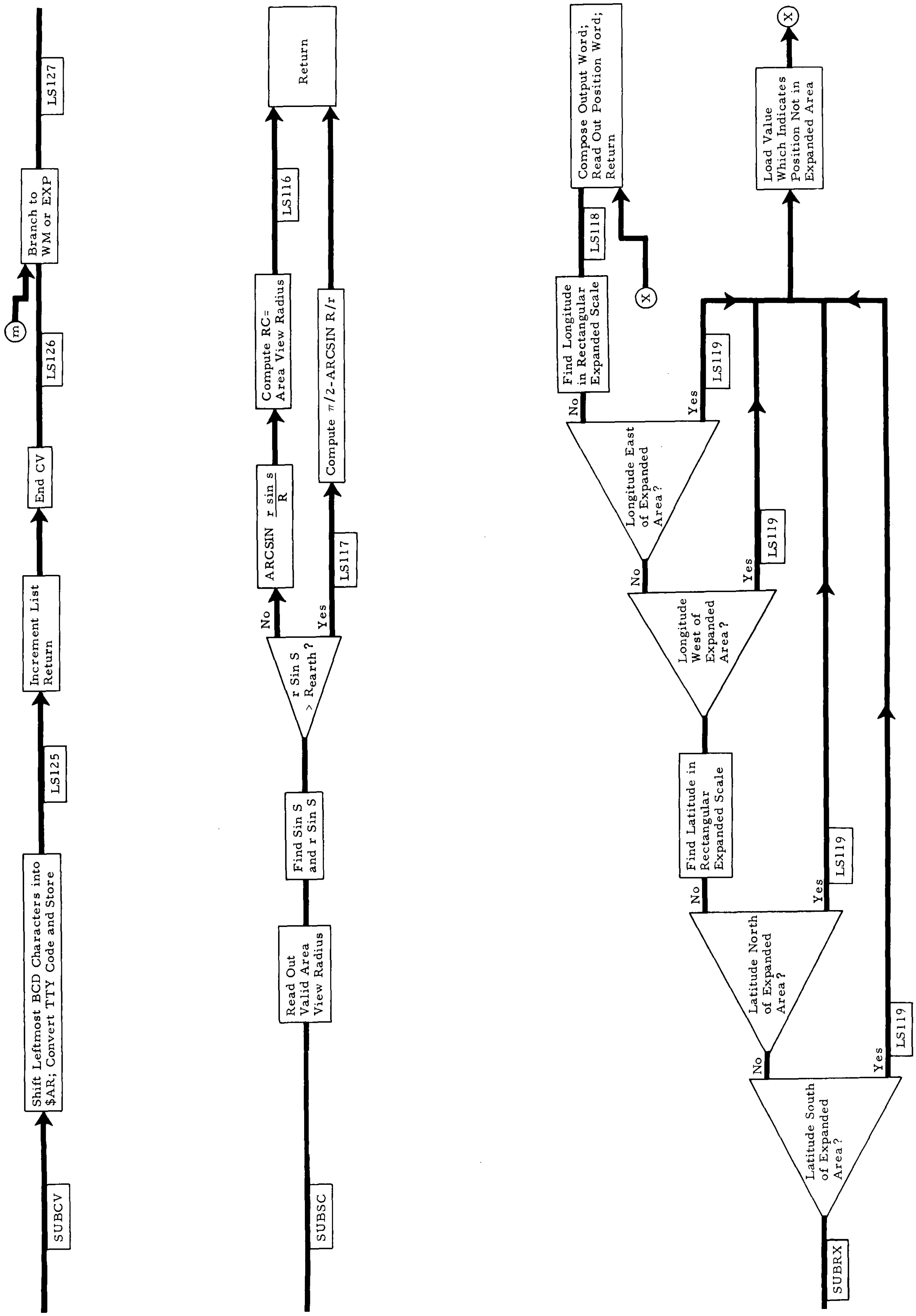


Enter SUBSC for View Circle; Convert View Circle Radius to BCD; Prepare to Enter SUBCV for Area-of-View Radius; Enter SUBCV; Enter SUBCV+11 to Skip One Space; Enter SUBCV Three Times; Skip Two Spaces Between Area-of-View Radius and Azimuth
Restore TDU1 Table; Find Cos B; Find Slant Range b; Find Sin B; Clear Double Precision Divide Check Indicator
|sin A| ≥ 1 ?
No
Yes
LS62
A = ArcSin (Cos φs Sin Δλ / Sin B)
Δλ ≥ Zero ?
No
Yes
LS63
Load -90° For Azimuth
Load +90° For Azimuth
(Sin φs - Sin φ0 Cos B) ≤ Zero ?
Yes
No
LS61
A = 180° - ArcSin (Cos φs Sin Δλ / Sin B)
is A ≥ 0?
Yes
No
LS50
LS51
Find A ≥ 0 Scaled 2^3
Convert A from Radians to Degrees in BCD; Prepare to Enter SUBCV for Azimuth; Enter SUBCV Three Times; Skip Two Spaces Between Azimuth and Elevation
Find C = ArcSin (r sin B / b)
2bR Cos C ≥ Zero ?
No
Yes
Find E
LS52
Load Blank As Sign of Elevation
LS53
Store Sign Character in TDU1; Convert E From Radians To Degrees BCD; Prepare to Enter SUBCV For Elevation; Enter SUBCV Two Times; Skip Three Spaces Between Elevation and TN
LS59
Load N's in Place of Azimuth and Elevation
LS53A
Prepare to Enter SUBCV For TN; Enter SUBCV Two Times
Nonprecision WM During TTY; Clear TTY Indicator; Load Address-1 of First TDU1 Character; Return to Restart; Bypass TM2
Orbit
Move Data From SDD (TN) To SCD For Orbit; Find P/2 And P/128; Find Δtε = Δt0 + P/2; Compute t - T0 = Δt0; Find Δts = Δt0 - P/2
LS54
Store Present Δt; Load Iteration Counter; Prepare to Enter SUBP1+6; Enter SUBP1+6; Compose Orbit Position Word; Read Out Orbit Position Word
Orbit Complete ?
Yes
No
Return To Restart
α
or
β

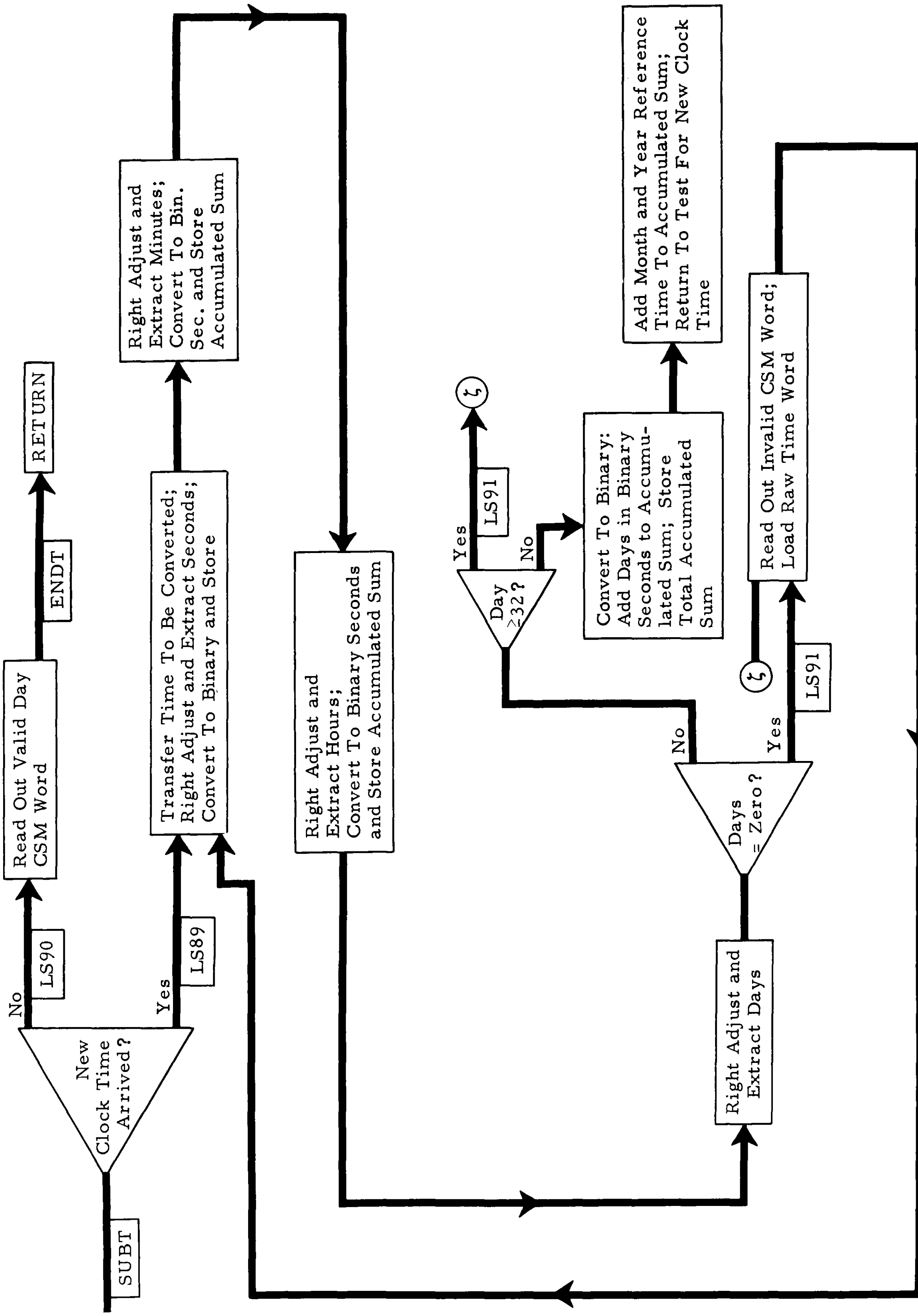


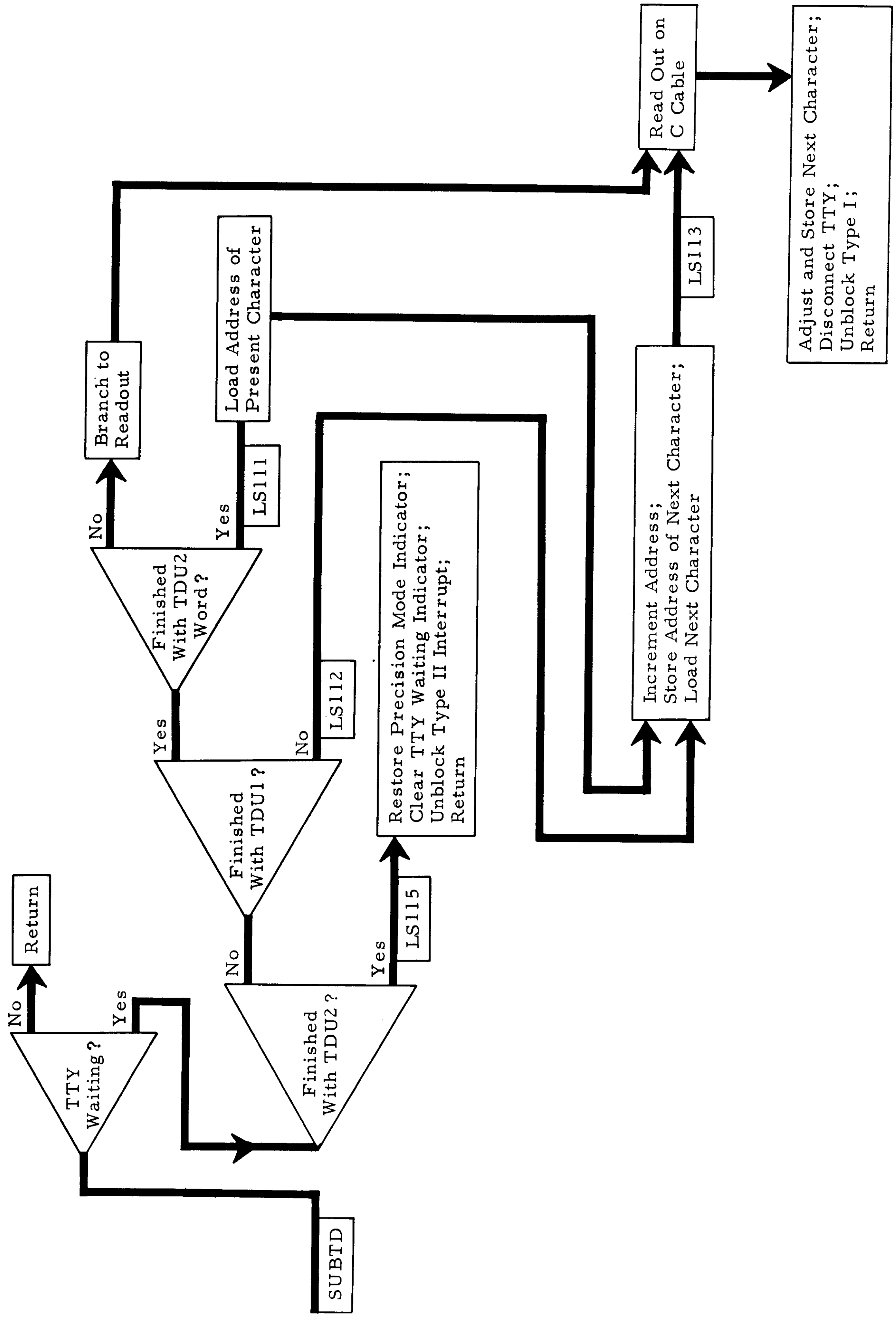
Month Interrupt
LS74
Load Valid Time CSM Word; Read Out; Extract Month; Right Adjust Store; Convert BCD to Binary
Month Zero?
No
Yes
Month ≥ 13?
No
Yes
LS78
Load Invalid Time CSM Word
CSM
r
Store Month in Binary; Clear Invalid Time Indicator
Year Valid?
Yes
No
Find Binary Seconds for Non-Leap-Year Months
LS75
S
Store Sum of Year and Month in Seconds
Leap Year?
No
Yes
LS127
LS76
δ
Load Invalid Time CSM Word Read Out Unblock Type II Interrupts
Idle Loop
Month < 3?
No
Yes
LS127
Add Number of Seconds Per Day to Sum of Year and Month
LS127
δ
Add New Month and Year Sum to Time Word
LS126
m
δ
Year Interrupt
LS77
Load Valid Time CSM Word; Read Out; Extract Year; Right Adjust; Convert BCD to Binary
Year Go?
No
Yes
Year ≥ 76?
No
Yes
LS79
Prepare to Enter CSM For Invalid Time and Read Out
CSM
r
Store Year in Binary; Clear Invalid Time Indicator
Month Valid?
No
Yes
Find Binary Seconds for Years Since 1960
Add Months in Seconds
LS75
S

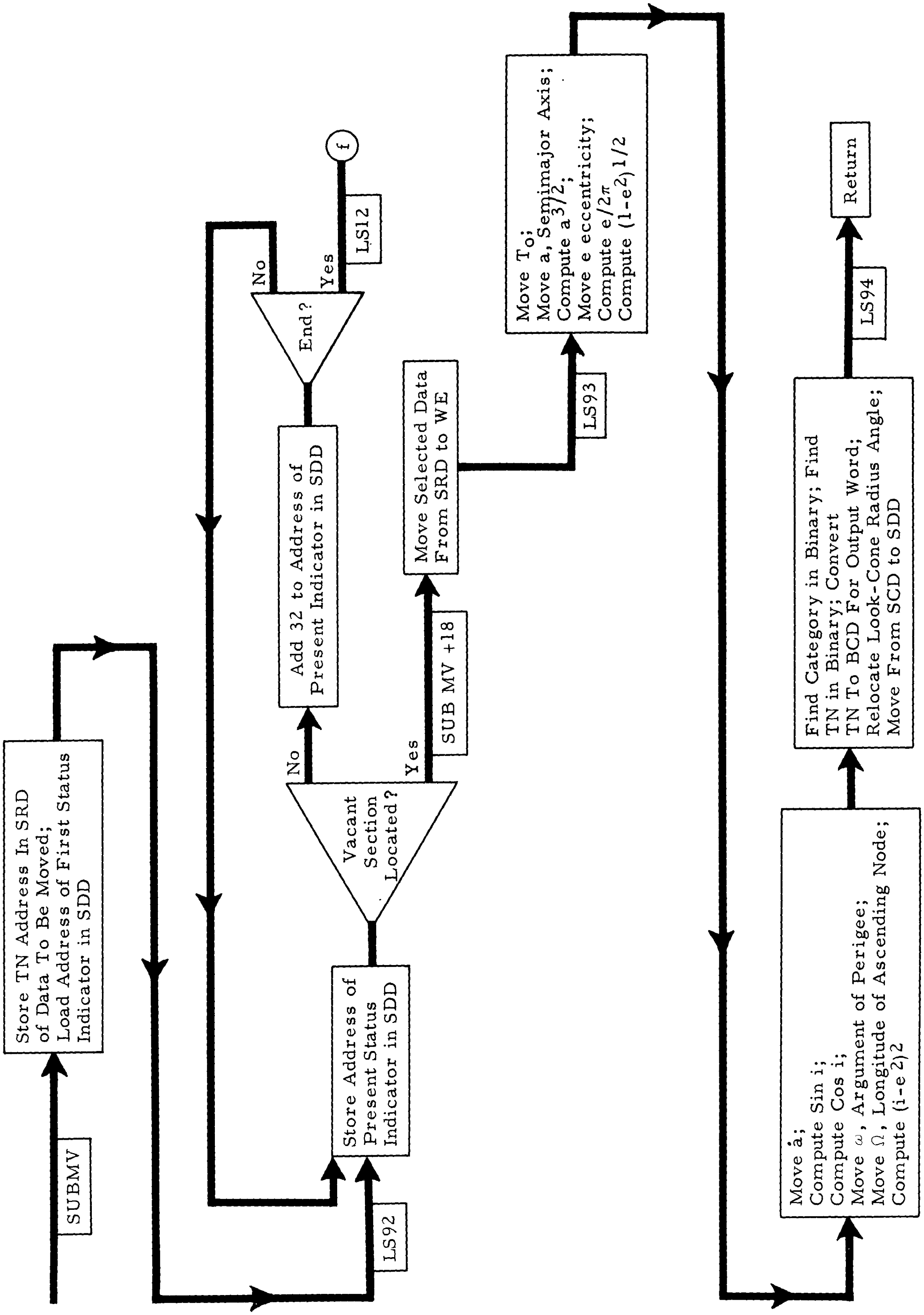

Own Ship Interrupt
LS83
Connect "A" Cable;
Preserve Time From Display Clock;
Call For Ship's Latitude;
Insert Ship's Speed in Symbol Word;
Convert Ship's Latitude From BCD to Binary
Latitude X10^4 in Binary Degrees Scaled 2^29;
Convert Latitude to Revolutions
Latitude South?
No
Yes
Compute and Store Sin φs
LS84
LS86
Load Negative Valve For Latitude
Compute and Store Cos φs ;
Call For Ship's Longitude;
Prepare Latitude For Position Word;
Convert Ship's Longitude To
Longitude X10^4 Binary Degrees
Scaled 2^29;
Convert Longitude to Revolutions
Longitude East?
No
Yes
LS85
LS87
Prepare Longitude For Position Word
Load and Store Negative Value of Longitude
Compose Ship's Video Position Word;
Call For Ship's Heading;
Compose Ship's Symbol Position Word;
Compose Ship's Symbol Word;
Read Out Ship's Symbol Word;
Restore Time From Display Clock
LS126
m

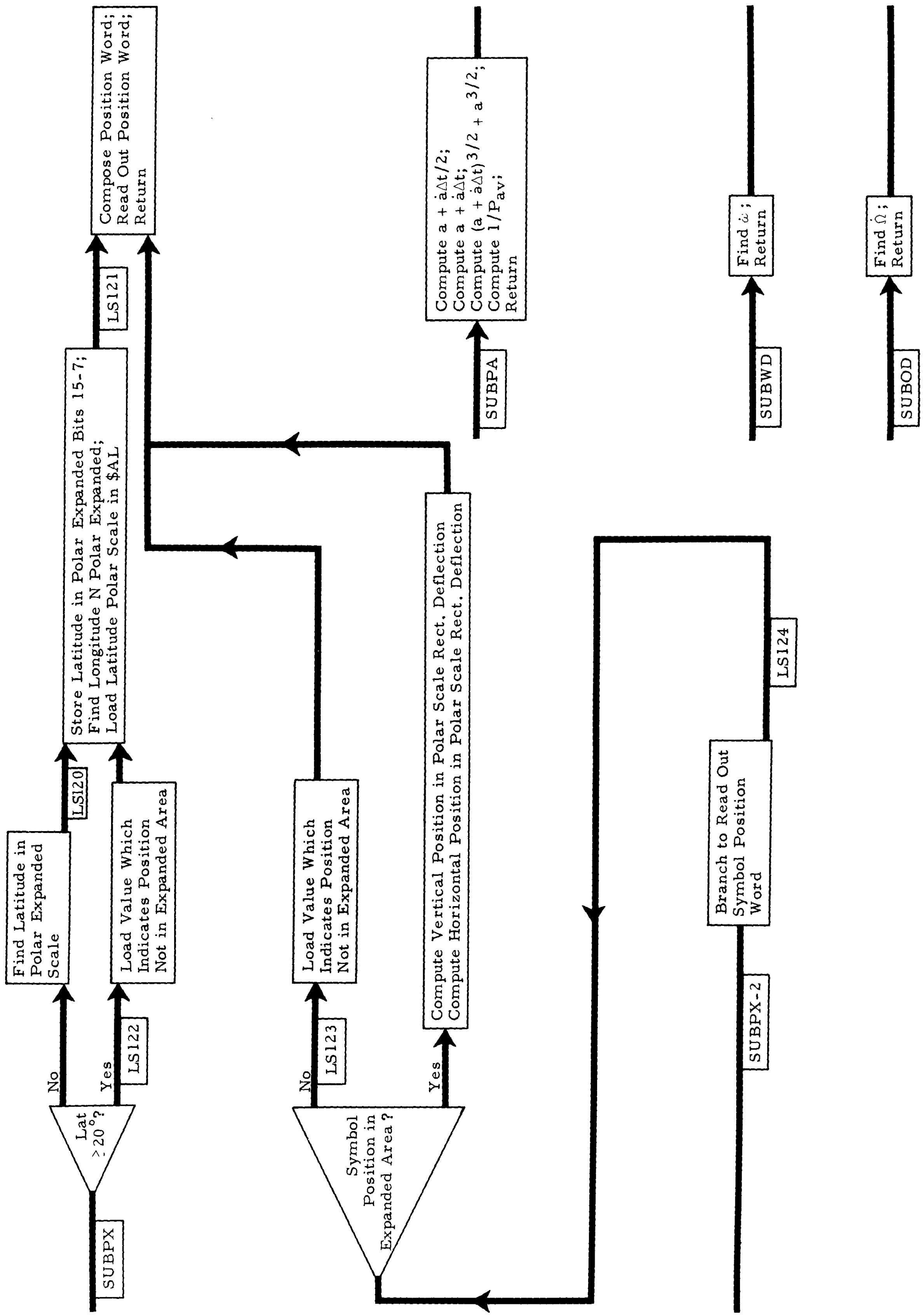


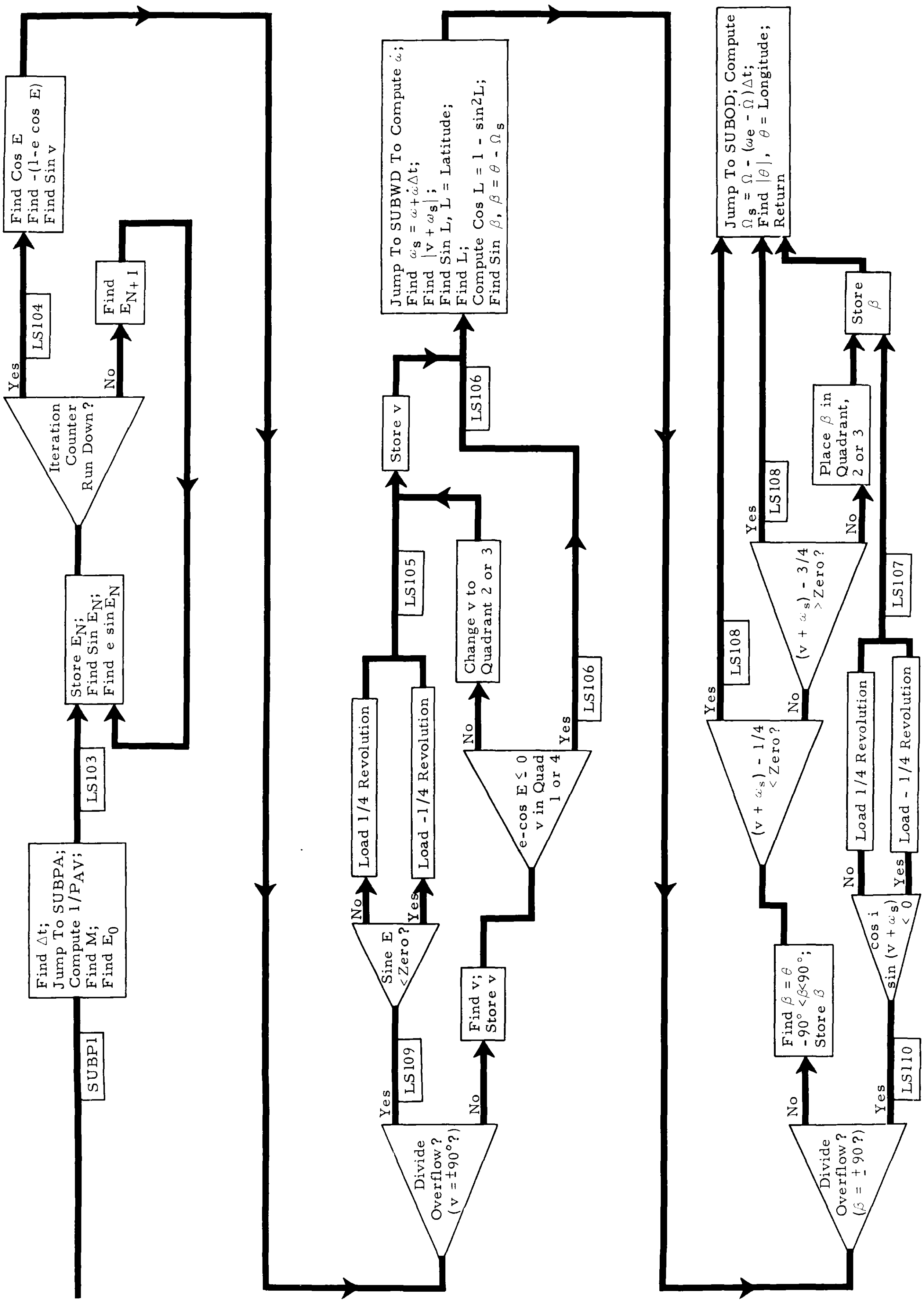
SUBCV
Shift Leftmost BCD Characters into $AR; Convert TTY Code and Store
LS125
Increment List Return
End CV
LS126
m
Branch to WM or EXP
LS127
SUBSC
Read Out Valid Area View Radius
Find Sin S and r Sin S
r Sin S > Rearth ?
No
Yes
ARCSIN r sin s / R
LS117
Compute π/2-ARCSIN R/r
Compute RC= Area View Radius
LS116
Return
SUBRX
Latitude South of Expanded Area ?
No
Yes
LS119
Latitude North of Expanded Area ?
No
Yes
LS119
Find Latitude in Rectangular Expanded Scale
Longitude West of Expanded Area ?
No
Yes
LS119
Longitude East of Expanded Area ?
No
Yes
LS119
Find Longitude in Rectangular Expanded Scale
LS118
X
Compose Output Word; Read Out Position Word; Return
Load Value Which Indicates Position Not in Expanded Area
X

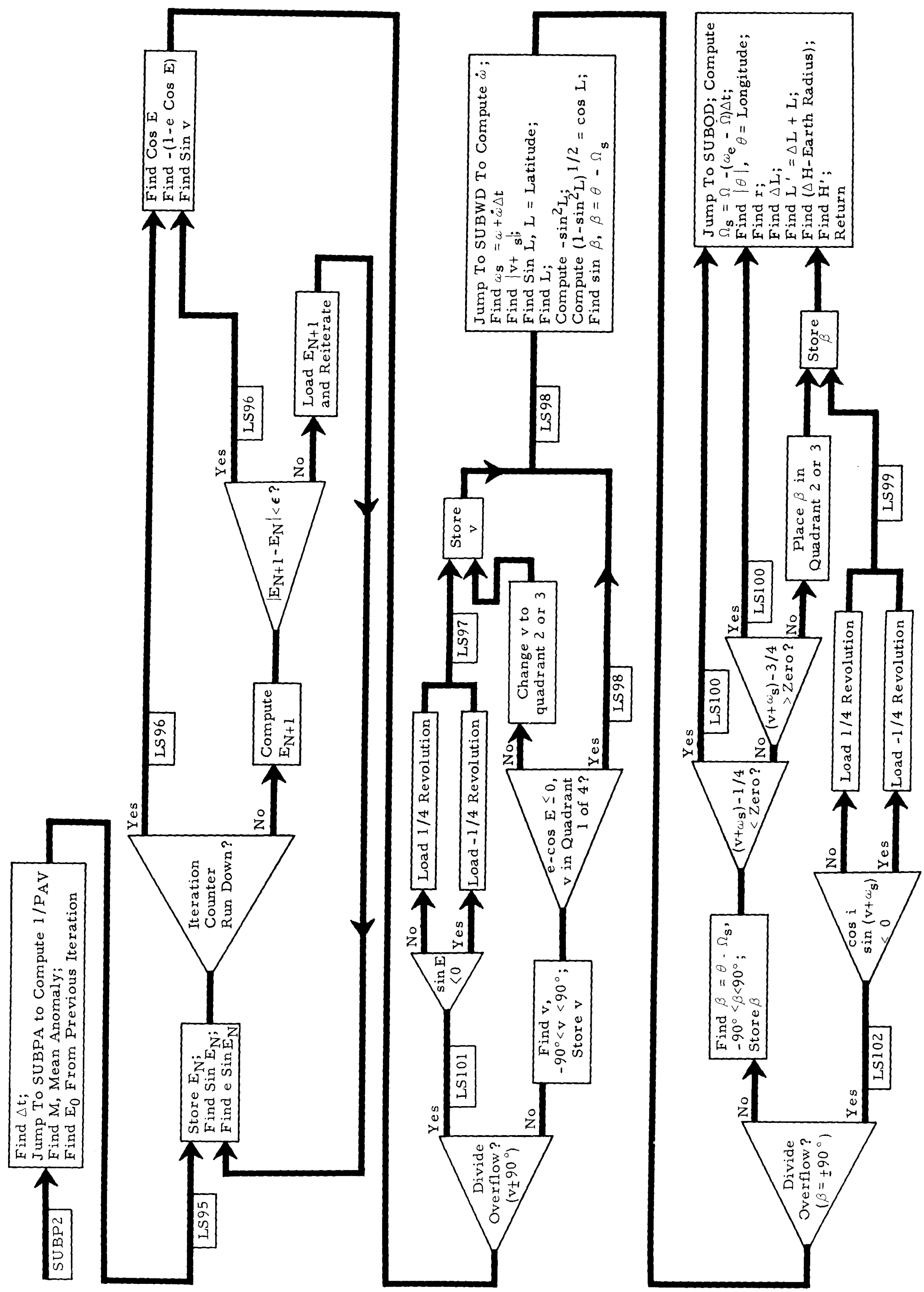

SUBT
New Clock Time Arrived?
No
LS90
Read Out Valid Day CSM Word
ENDT
RETURN
Yes
LS89
Transfer Time To Be Converted; Right Adjust and Extract Seconds; Convert To Binary and Store
Right Adjust and Extract Minutes; Convert To Bin. Sec. and Store Accumulated Sum
Right Adjust and Extract Hours; Convert To Binary Seconds and Store Accumulated Sum
Right Adjust and Extract Days
Days = Zero?
No
Yes
LS91
Day ≥32?
Yes
LS91
No
Convert To Binary: Add Days in Binary Seconds to Accumulated Sum; Store Total Accumulated Sum
Add Month and Year Reference Time To Accumulated Sum; Return To Test For New Clock Time
Read Out Invalid CSM Word; Load Raw Time Word


SUBTD
TTY Waiting?
No
Return
Yes
Finished With TDU2?
No
Yes
LS115
Restore Precision Mode Indicator;
Clear TTY Waiting Indicator;
Unblock Type II Interrupt;
Return
Finished With TDU1?
Yes
No
LS112
Finished With TDU2 Word?
No
Yes
LS111
Branch to Readout
Load Address of Present Character
Increment Address;
Store Address of Next Character;
Load Next Character
LS113
Read Out on C Cable
Adjust and Store Next Character;
Disconnect TTY;
Unblock Type I;
Return


SUBMV
Store TN Address In SRD of Data To Be Moved; Load Address of First Status Indicator in SDD
LS92
Store Address of Present Status Indicator in SDD
Vacant Section Located?
No
Yes
Add 32 to Address of Present Indicator in SDD
End?
No
Yes
LS12
f
SUB MV +18
Move Selected Data From SRD to WE
LS93
Move T_o; Move a, Semimajor Axis; Compute $a^{3/2}$; Move e eccentricity; Compute $e/2\pi$; Compute $(1-e^2)^{1/2}$
Move $\dot{a}$; Compute Sin i; Compute Cos i; Move ω, Argument of Perigee; Move Ω, Longitude of Ascending Node; Compute $(i-e^2)^2$
Find Category in Binary; Find TN in Binary; Convert TN To BCD For Output Word; Relocate Look-Cone Radius Angle; Move From SCD to SDD
LS94
Return


SUBPX
Lat ≥20°?
No
Yes
Find Latitude in Polar Expanded Scale
LS120
LS122
Load Value Which Indicates Position Not in Expanded Area
Store Latitude in Polar Expanded Bits 15-7; Find Longitude N Polar Expanded; Load Latitude Polar Scale in $AL
LS121
Compose Position Word; Read Out Position Word; Return
Symbol Position in Expanded Area?
No
LS123
Load Value Which Indicates Position Not in Expanded Area
Yes
Compute Vertical Position in Polar Scale Rect. Deflection
Compute Horizontal Position in Polar Scale Rect. Deflection
SUBPA
Compute a + ȧΔt/2;
Compute a + ȧΔt;
Compute (a + ȧΔt)3/2 + a3/2;
Compute 1/Pav;
Return
SUBPX-2
Branch to Read Out Symbol Position Word
LS124
SUBWD
Find ω̇ ;
Return
SUBOD
Find Ω̇ ;
Return


SUBP1
Find Δt; Jump To SUBPA; Compute 1/P_AV; Find M; Find E_0
LS103
Store E_N; Find Sin E_N; Find e sinE_N
Iteration Counter Run Down?
Yes
No
LS104
Find E_N+1
Find Cos E Find -(1-e cos E) Find Sin v
Divide Overflow? (v = ±90°?)
Yes
No
LS109
Find v; Store v
Sine E <Zero?
No
Yes
Load 1/4 Revolution
Load -1/4 Revolution
e-cos E ≤ 0 v in Quad 1 or 4
No
Yes
LS106
Change v to Quadrant 2 or 3
LS105
Store v
LS106
Jump To SUBWD To Compute ω̇; Find ω_s = ω+ω̇Δt; Find |v + ω_s|; Find Sin L, L = Latitude; Find L; Compute Cos L = 1 - sin2L; Find Sin β, β = θ - Ω s
Divide Overflow? (β = ± 90?)
No
Yes
LS110
Find β = θ -90° <β<90°; Store β
cos i sin (v + ω_s) < 0
No
Yes
Load 1/4 Revolution
Load - 1/4 Revolution
LS107
(v + ω_s) - 1/4 <Zero?
Yes
No
LS108
(v + ω_s) - 3/4 >Zero?
Yes
No
LS108
Place β in Quadrant, 2 or 3
Store β
Jump To SUBOD; Compute Ω_s = Ω - (ωe - Ω)Δt; Find |θ|, θ = Longitude; Return


SUBP2
LS95
Find Δt;
Jump To SUBPA to Compute 1/PAV
Find M, Mean Anomaly;
Find E0 From Previous Iteration
Store EN;
Find Sin EN;
Find e Sin EN
Iteration Counter Run Down?
Yes
No
LS96
Compute EN+1
|EN+1-EN| < ϵ?
Yes
No
LS96
Load EN+1 and Reiterate
Find Cos E
Find -(1-e Cos E)
Find Sin v
Divide Overflow? (v±90°)
Yes
No
LS101
sin E < 0
No
Yes
Load 1/4 Revolution
Load -1/4 Revolution
LS97
Store v
Find v, -90° < v < 90°; Store v
e-cos E ≤ 0, v in Quadrant 1 of 4?
No
Yes
Change v to quadrant 2 or 3
LS98
LS98
Jump To SUBWD To Compute ω̇;
Find ωs = ω + ω̇Δt
Find |v + ωs|
Find Sin L, L = Latitude;
Find L;
Compute -sin²L;
Compute (1-sin²L)1/2 = cos L;
Find sin β, β = θ - Ωs
Divide Overflow? (β = ±90°)
Yes
No
LS102
Find β = θ - Ωs, -90° < β < 90°; Store β
(v+ωs)-1/4 < Zero?
Yes
No
LS100
(v+ωs)-3/4 > Zero?
Yes
No
LS100
cos i sin (v+ωs) < 0
No
Yes
Load 1/4 Revolution
Load -1/4 Revolution
LS99
Place β in Quadrant 2 or 3
Store β
Jump To SUBOD; Compute
Ωs = Ω - (ωe - Ω̇)Δt;
Find |θ|, θ = Longitude;
Find r;
Find ΔL;
Find L' = ΔL + L;
Find (ΔH-Earth Radius);
Find H';
Return

UNCLASSIFIED
Security Classification

DOCUMENT CONTROL DATA - R&D		
(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)		
1. ORIGINATING ACTIVITY *(Corporate author)* Naval Research Laboratory Washington, D.C. 20390	2a. REPORT SECURITY CLASSIFICATION Unclassified 2b. GROUP	
3. REPORT TITLE COMPUTER PROGRAM FOR THE NRL SATELLITE POSITION DISPLAY		
4. DESCRIPTIVE NOTES *(Type of report and inclusive dates)* Interim report on the problem		
5. AUTHOR(S) *(Last name, first name, initial)* Orsino, R.J., and Francavilla, T.L.		
6. REPORT DATE December 5, 1966	7a. TOTAL NO. OF PAGES 50	7b. NO. OF REFS 4
8a. CONTRACT OR GRANT NO. NRL Problem Y01-01 b. PROJECT NO. SF 019-01-03, Task 6168 c. d.	9a. ORIGINATOR'S REPORT NUMBER(S) NRL Report 6446 9b. OTHER REPORT NO(S) *(Any other numbers that may be assigned this report)*	
10. AVAILABILITY/LIMITATION NOTICES Distribution of this document is unlimited		
11. SUPPLEMENTARY NOTES	12. SPONSORING MILITARY ACTIVITY Department of the Navy (Naval Ship Systems Command), Washington, D.C. 20360	

13. ABSTRACT

A computer driven satellite display has recently been developed and evaluated at the Naval Research Laboratory. This equipment, designated as the NRL Satellite position Prediction And Display (SPAD), provides a considerable amount of display control versatility. Up to eleven satellites can be selected either individually or by category from a repertoire of 88 satellites. These selected satellites are then updated and displayed in either real time or at an accelerated time. Several display modes are available at an operator's command, namely, world map, rectangular expanded mode, and polar expanded mode.

A commercial equivalent of the AN/UYK-1 computer was used in the research version of SPAD. Since the program was written in a language peculiar to this machine, the program listings and codings are omitted from the present report. However, included in this report are the complete set of computer program flow diagrams and a brief description of the interrupt service routines along with their corresponding subroutines. Characteristics of the AN/UYK-1 computer, operator control panel provisions and possible future program modifications are discussed. The contents of this report should be a sufficient indication of the program philosophy. This philosophy, with appropriate minor changes, could be implemented by any suitable language.

DD FORM 1 JAN 64 1473

UNCLASSIFIED
Security Classification

14. KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
Data storage systems Satellites (artificial) Programming (computers) Display systems Shipborne SPAD (satellite position prediction and display)						

INSTRUCTIONS

1. ORIGINATING ACTIVITY: Enter the name and address of the contractor, subcontractor, grantee, Department of Defense activity or other organization *(corporate author)* issuing the report.

2*a*. REPORT SECURITY CLASSIFICATION: Enter the overall security classification of the report. Indicate whether "Restricted Data" is included. Marking is to be in accordance with appropriate security regulations.

2*b*. GROUP: Automatic downgrading is specified in DoD Directive 5200.10 and Armed Forces Industrial Manual. Enter the group number. Also, when applicable, show that optional markings have been used for Group 3 and Group 4 as authorized.

3. REPORT TITLE: Enter the complete report title in all capital letters. Titles in all cases should be unclassified. If a meaningful title cannot be selected without classification, show title classification in all capitals in parenthesis immediately following the title.

4. DESCRIPTIVE NOTES: If appropriate, enter the type of report, e.g., interim, progress, summary, annual, or final. Give the inclusive dates when a specific reporting period is covered.

5. AUTHOR(S): Enter the name(s) of author(s) as shown on or in the report. Enter last name, first name, middle initial. If military, show rank and branch of service. The name of the principal author is an absolute minimum requirement.

6. REPORT DATE: Enter the date of the report as day, month, year; or month, year. If more than one date appears on the report, use date of publication.

7*a*. TOTAL NUMBER OF PAGES: The total page count should follow normal pagination procedures, i.e., enter the number of pages containing information.

7*b*. NUMBER OF REFERENCES: Enter the total number of references cited in the report.

8*a*. CONTRACT OR GRANT NUMBER: If appropriate, enter the applicable number of the contract or grant under which the report was written.

8*b*, 8*c*, & 8*d*. PROJECT NUMBER: Enter the appropriate military department identification, such as project number, subproject number, system numbers, task number, etc.

9*a*. ORIGINATOR'S REPORT NUMBER(S): Enter the official report number by which the document will be identified and controlled by the originating activity. This number must be unique to this report.

9*b*. OTHER REPORT NUMBER(S): If the report has been assigned any other report numbers *(either by the originator or by the sponsor)*, also enter this number(s).

10. AVAILABILITY/LIMITATION NOTICES: Enter any limitations on further dissemination of the report, other than those imposed by security classification, using standard statements such as:

(1) "Qualified requesters may obtain copies of this report from DDC."

(2) "Foreign announcement and dissemination of this report by DDC is not authorized."

(3) "U. S. Government agencies may obtain copies of this report directly from DDC. Other qualified DDC users shall request through
____."

(4) "U. S. military agencies may obtain copies of this report directly from DDC. Other qualified users shall request through
____."

(5) "All distribution of this report is controlled. Qualified DDC users shall request through
____."

If the report has been furnished to the Office of Technical Services, Department of Commerce, for sale to the public, indicate this fact and enter the price, if known.

11. SUPPLEMENTARY NOTES: Use for additional explanatory notes.

12. SPONSORING MILITARY ACTIVITY: Enter the name of the departmental project office or laboratory sponsoring *(paying for)* the research and development. Include address.

13. ABSTRACT: Enter an abstract giving a brief and factual summary of the document indicative of the report, even though it may also appear elsewhere in the body of the technical report. If additional space is required, a continuation sheet shall be attached.

It is highly desirable that the abstract of classified reports be unclassified. Each paragraph of the abstract shall end with an indication of the military security classification of the information in the paragraph, represented as *(TS)*, *(S)*, *(C)*, or *(U)*.

There is no limitation on the length of the abstract. However, the suggested length is from 150 to 225 words.

14. KEY WORDS: Key words are technically meaningful terms or short phrases that characterize a report and may be used as index entries for cataloging the report. Key words must be selected so that no security classification is required. Identifiers, such as equipment model designation, trade name, military project code name, geographic location, may be used as key words but will be followed by an indication of technical context. The assignment of links, roles, and weights is optional.