1965

A tactical data simulation program for the Control Data 1604 computer and the dd65 display console

Borden, Edward L.

Monterey, California: U.S. Naval Postgraduate School

http://hdl.handle.net/10945/13247

A TACTICAL DATA SIMULATION PROGRAM FOR THE CONTROL DATA 1604 COMPUTER AND THE dd65 DISPLAY CONSOLE

EDWARD L. BORDEN
and
ANTHONY C. CASCIATO

A TACTICAL DATA SIMULATION PROGRAM FOR THE CONTROL DATA

1604 COMPUTER AND THE dd65 DISPLAY CONSOLE

* * * * *

Edward L. Borden

and

Anthony C. Casciato

A TACTICAL DATA SIMULATION PROGRAM FOR THE CONTROL DATA

1604 COMPUTER AND THE dd65 DISPLAY CONSOLE

by

Edward L. Borden

Lieutenant, United States Navy

and

Anthony C. Casciato

Lieutenant, United States Navy

Submitted in partial fulfillment of
the requirements for the degree of

MASTER OF SCIENCE
IN
ENGINEERING ELECTRONICS

United States Naval Postgraduate School

Monterey, California

1 9 6 5

A TACTICAL DATA SIMULATION PROGRAM FOR THE CONTROL DATA 1604 COMPUTER AND THE dd65 DISPLAY CONSOLE

by

Edward L. Borden

and

Anthony C. Casciato

This work is accepted as fulfilling

the thesis requirements for the degree of

MASTER OF SCIENCE

IN

ENGINEERING ELECTRONICS

from the

United States Naval Postgraduate School

ABSTRACT

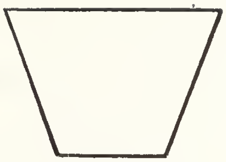
There exists a need for a simulation program that will enable a programmer to simulate systems such as the Naval Tactical Data System (NTDS), Marine Corps Tactical Data System (MTDS), Airborne Tactical Data System (ATDS), or any other system that uses radar or sonar information for operation. A programming project was undertaken using the Control Data 1604 Computer (CDC 1604) and the Data Display Incorporated model dd65 Display Console (DD65) as the basic hardware. The program developed features on-line control of the CDC 1604 by the DD65, display of simulated targets and messages on the DD65, control of tracks by operating personnel from the DD65, continuous graph output for permanent record purposes, and real time or double real time operation. This program can be expanded to provide more features of any of the tactical data systems by the addition of the appropriate subroutines.

TABLE OF CONTENTS

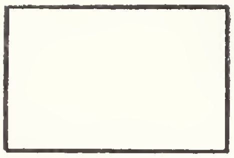
LIST OF ILLUSTRATIONS

TABLE OF SYMBOLS

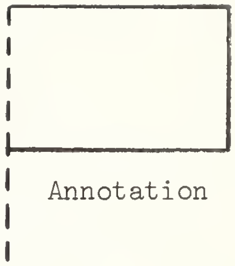
Basic Symbols



Input/Output



Processing



Annotation

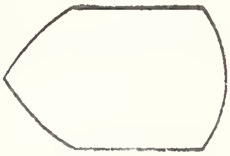
Specialized Input/Output Symbols

Punched Card

Document

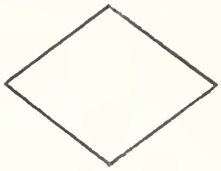
Manual Input

Display

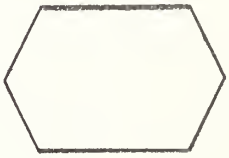


Specialized Processing Symbols

Decision

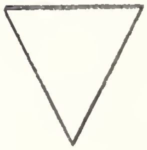


Predefined Process



Additional Symbols

Entry/Exit



Connector

Statement Tag

Off-page connector

1. Introduction.

1.1 Background. With the development of new weapons systems, there has been a renewed interest in the study of tactical methods and the importance of command decisions. This first manifested itself in studies of abstract mathematical models which later provided the basis for pure computer simulation. By a thorough analysis of the results of the simulation, much valuable data on expected system performance was obtained; however, these models did not allow for the man made command decision, nor did they operate in real time with a realistic environment. There must be a more flexible man-machine interface to allow the tactician to construct controlled situations at will, to observe directly the effect of one or more command decision links, to modify parameters assumed for system components, and to provide hard-copy records for a later critique of the situation.

1.2 Proposed Approach. The objective of the thesis project to be described in this paper was to undertake, within the constraints of currently available equipment, the implementation of a simulation system that would be used as a test vehicle to study any problem that involve trajectories. The salient characteristics of the system were to be:

(1) A flexible, fine grain, three dimensional track generator.

(2) A direct man-computer interaction capability (i.e. "on-line" control).

(3) A language independent approach to setting up simulated tactical situations.

(4) A modular approach to inserting sub-systems for testing.

(5) A real time approach to the simulation problem.

A flexible track generator was felt to be a basic necessity in order to provide realistic problem inputs to replace the classical step and ramp input functions. The track generator should have the capability of duplicating the actual position information available to a surveillance system.

In order to take maximum advantage of man-computer interaction capability, it was felt there was a need for direct man-computer communication as well as pre-programmed communication. To this end, an executive routine, that would act as the coordinating routine between the track generator, man communication, display, etc., would be developed. The executive routine would operate in real time in order to provide realistic simulation.

1.3 User Capabilities.

1.3.1 Tactical Data Simulator. A typical tactical data problem (NTDS type) would be one of pre-programmed hostile tracks, one or more surface ships on station, and one or more friendly aircraft, each under computer (pre-programmed) or console control. Then either a canned problem or an actual problem could be run using the console for track modification.

By use of the hard-copy critique output, the user may examine one or more of the following:

(1) The quality of the position information generated by his systems equations.

(2) The effects of various noise environments upon the quality of position information.

(3) The results of tactical decisions made and a comparison of the results, of rerunning the problem, with different tactical decisions being made.

1.3.2 Control System Test Vehicle. An example of a control system test vehicle would be a test of a Carrier Automatic Landing System. The track generator would furnish the fine grain aircraft position data and would receive orders from the landing system computers. The stability of the Carrier Automatic Landing System, as well as its accuracy, versus changing parameters could readily be studied. Using the same approach, many different types of control systems could be studied, such as:

(1) Air to Surface missile systems.

(2) Surface to Surface missile systems.

(3) Surface to Air missile systems.

(4) Subsurface to Air missile systems.

2. Programming.

2.1 Programming Language. The following languages were readily available to the authors:

(1) FORTRAN 60.

(2) FORTRAN SYMBOLIC.

(3) FORTRAN 63.

(4) ALGOL.

(5) SCRAP.

(6) CODAP.

In choosing a language, several criteria must be met:

(1) Ease of manipulating individual bits.

(2) Flexible input/output routines.

(3) Availability of diversified library subroutines.

(4) Flexibility of operation.

(5) Ease of programming.

Some of the available languages satisfied several of the desired criteria, however, it was felt that only FORTRAN SYMBOLIC met all of the requirements. In addition, the FORTBIN compiler associated with FORTRAN SYMBOLIC is very reliable and libraries may be manipulated at will. For these reasons FORTRAN SYMBOLIC was chosen.

2.2 Initial Programming. Initially, it was decided to divide the programming into two areas. These areas were:

(1) Executive Control Routine.

(2) Track Generator.

It was felt that each of these areas were independent enough to allow for separate work. A general outline of the interface was set down,

	EXECUTIVE FLAG TABLE (EFT)	EXECUTIVE TIME TABLE (ETT)	EXECUTIVE JUMP TABLE (EJT)
1		1 second	A
2		6 seconds	B
3		6 seconds	C
4		10 minutes	D
5		2 hours	A
6		1 minute	E
7		30 seconds	F
8		4 seconds	H
9		1 hour	D
M =10			

Figure 1. Example of Executive Table Structure

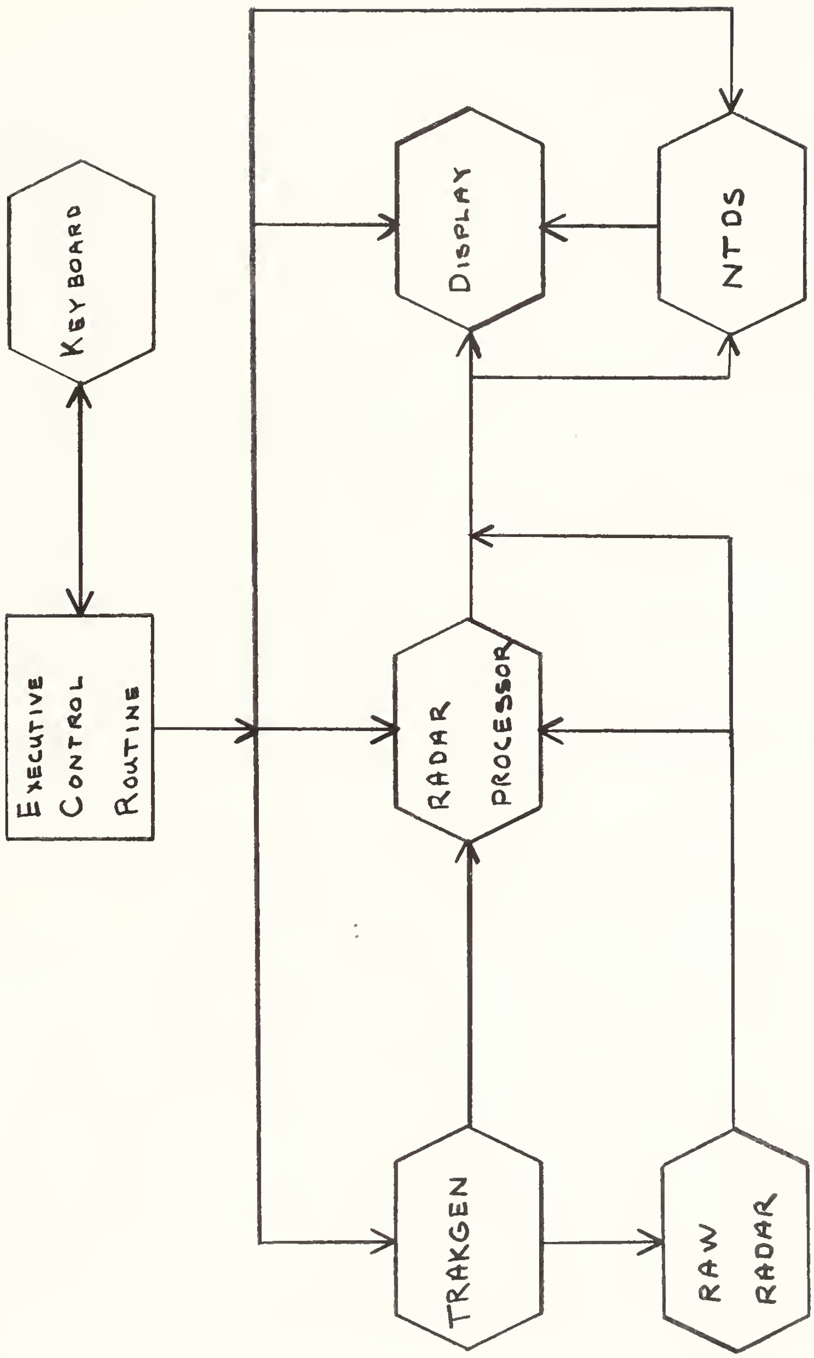


Figure 2. Original Concept of SIMONE

with the final details left to be established as the programs developed. This worked very satisfactorily.

2.2.1 SIMONE.

2.2.1.1. Basic Philosophy. Upon analysis of the problem, the initial frame work was set down as represented by Fig. 2. The system was to operate in real-time, therefore, an executive routine would be needed and would be the heart of the system. The philosophy of the executive routine is found in [14]. Briefly this philosophy is:

(1) Subprograms are considered for execution on the basis of their priorities as system components.

(2) Subprograms would be listed by priority and this list would be scanned sequentially, executing subprograms as they are needed.

(3) After each execution, the scanning process would be resumed starting with the highest priority entry.

(4) Assumption is made that sufficient computational capability exists to perform all assigned tasks.

(5) A subprogram is executed only when its "flag" is set.

The executive control philosophy is implemented through a method of table control. Three tables, Executive Flag Table (EFT), Executive Time Table (ETT), and Executive Jump Table (EJT), contain the necessary information for executive control. See Fig. 1. There is a one-to-one correspondence between these tables. Fig. 3 is a flow chart of the Executive Control Program. Starting with this basic routine and the functions of the following subprograms, the initial programming was undertaken:

(1) Keyboard Processor

(2) Radar Processor

(3) Track Generator

(4) Display Routine

(5) Raw Data Routine

(6) NTDS Routine

2.2.1.2 Programming and Problems. The implementation of Fig. 3 was accomplished with very little difficulty. The executive tables were implemented as arrays with the exception of the EJT. The EJT was programmed as the addresses of jump instructions in the program. By using the philosophy of the computed GO TO of FORTRAN, the appropriate jump will be executed based on the EFT index.

With the basic control routine working, the next step was to determine the method by which the DD 65 would communicate with the main computer. There are two basic methods by which the DD 65 may communicate with the CDC 1604. The first is via the CDC 160 and the second is via Channel 7. The first method involves the use to the CDC 160 to interrogate the DD 65 and then to transfer the information from the DD 65 to a buffer in the CDC 160 and then finally to transfer this stored information to the CDC 1604. This method is similar to that used in satellite operations. [12] The second method provides for direct communications between the CDC 1604 and the DD 65 via Channel 7. The latter method has two modes of operations with respect to the DD 65, active and passive. The active mode involves the use of the interrupt features of the DD 65 while the passive mode involves an interrogation similar to that used by the CDC 160. The first method using the CDC 160 was not

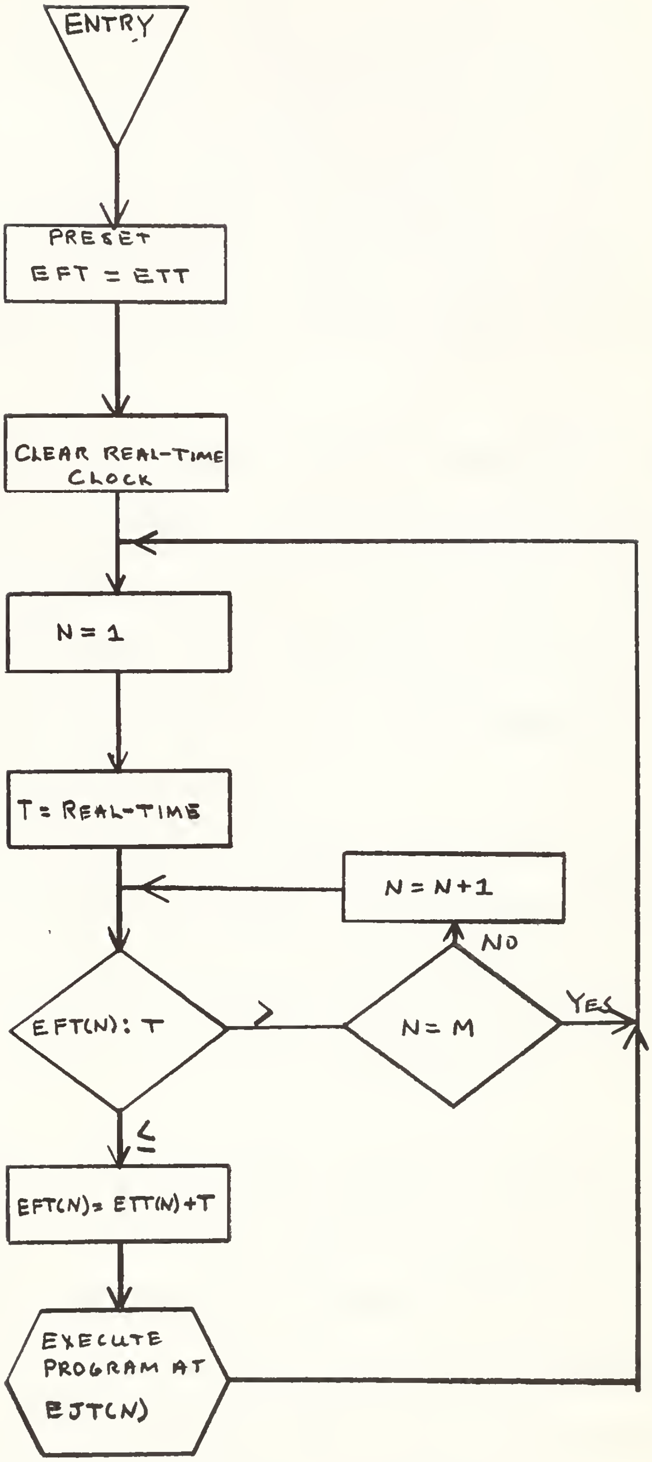


Figure 3. An Executive Control Program

chosen due to the desire to keep the programming as simple as possible. The active mode of the second method was chosen because of its ability to provide continuous control of the CDC 1604. In the passive mode, control of the CDC 1604 would occur only at predetermined times. This may or may not be desirable. In the active mode, the desired control information is transferred to the CDC 1604 instantaneously. This instantaneous transfer allows for the fastest response to the commands originated at the DD 65 console. With these decisions made, the major programming was started.

To program the interrupts, a copy of the MACHINE RESIDENT was used to determine the actual address in memory of the clock processor. By substituting the address of SIMONE's interrupt routine in lower cell 7, and by jumping to the clock processor after checking for interrupts from the DD 65, the interrupt tests were ready to run. This arrangement proved to be satisfactory for the time being.

As the program developed, changes were made to RESIDENT and consequently the program had to change. In order to make SIMONE independent of any address in RESIDENT, the clock processor was incorporated in SIMONE's interrupt processing routine. At this time the double real-time feature was also added.

The next problem was that of how to handle the input functions. The first approach was to use a very complicated jump table and flags within the subroutines. This approach proved very cumbersome, therefore, a new solution was sought. This solution came in the form of a 30 x 3 matrix, which composed a control table for Keyboard 2, and a buffer for storage of both Keyboard 1 and 2 information. See Fig. 5 for Keyboard 2

Control Table. Each row in the matrix corresponds to one of the switches on Keyboard 2. The columns provide information that enables the computer to process the command.

The problem of display was solved with the development of subroutine PRINT. For a complete explanation see Appendix III. For the display, symbols had to be developed by using the vector mode of the DD 65. These symbols are shown in Fig. 6. The results were not entirely satisfactory. The raw radar symbol was changed to just two vectors end to end with the X and Y position located at the junction of the two vectors. The remaining symbols are fine for the smaller scales, however, they are too large when the display scale is set for 256 miles. These symbols are, however, the smallest available with present equipment and logic.

As the philosophy of the program expanded, it became evident that there must be some type of hard-copy record of each problem run. This was the beginning of the critique routine. The initial thoughts centered around the use of scratch tapes to have all information and then at the end of the problem to provide a graph output. The basic idea was to plot one graph that would show all tracks in the xy plane. The critique routine was programmed and tested. The results of the test showed this method to be feasible, however, some difficulty was encountered. When trying to read the scratch tape, parity errors occurred. To eliminate this problem, memory was chosen to be the scratch pad. This proved quite satisfactory, however, due to the limited size of the scratch pad, only 10 minutes of information could be stored. The solution to this problem was another subprogram that would plot a

graph every 10 minutes. During the reprogramming, a second graph was added that would provide for an altitude profile vs. time plot. The last modification included the point plotting of both the smoothed position and the actual position at time intervals to be specified by the user.

The radar processor was made very simple. It included the smoothing equations normally associated with the "Alpha-Beta" tracker. [1]. It is in this routine where there is much to be done. Several types of smoothing and predicting equations are available. Both the NTDS and the MTDS systems were studied.

As the programming continued, the basic frame work was expanded to include many routines. These routines are explained in detail in Appendix I and an explanation of their use is found in Section 4.4.2. See Fig. 4 for final Block Diagram.

2.2.1.3 Future Expansions. There are many areas for expansion. Some of these would be:

(1) On-line graph plotting on an XY plotter.

(2) New smoothing equations.

(3) A more complete intercept routine.

(4) A more sophisticated display routine.

(5) Integration of program with FORTSHARE [12] .

(6) Incorporation of a radar characteristic table.

(7) A more complete aircraft characteristic table.

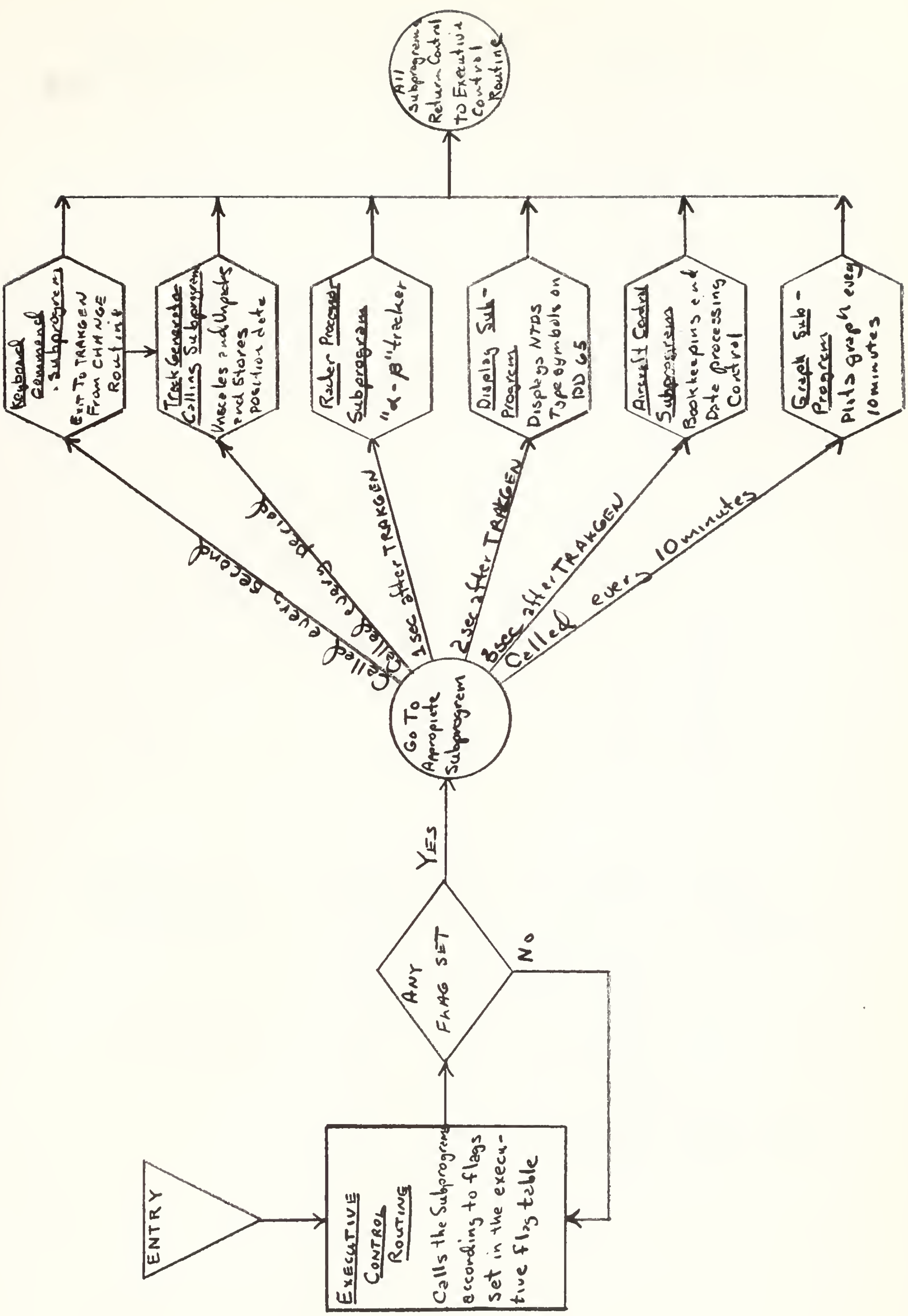


Figure 4. Final Block Diagram of SIMONE

BUTTON TABLE	LIGHT TABLE	TRANSFER TABLE	
01	77202	5	P0
02	77204	6	P1
03	77210	6	P2
04	77220	7	P3
05	77240	5	P4
11	77302	4	P5
12	77304	4	P6
↓	↓	↓	
53	77710	5	P27
54	77720	8	P28
55	77740	8	P29

BUTTON TABLE = Code from DD 65 Keyboard 2 switch.

LIGHT TABLE = External Function Code for turning on Computer Controlled light indicator.

TRANSFER TABLE = Number of Characters in the name of the program called and the address of the name. (Used to transfer program name to the buffer.)

Figure 5. Keyboard 2 Control Table

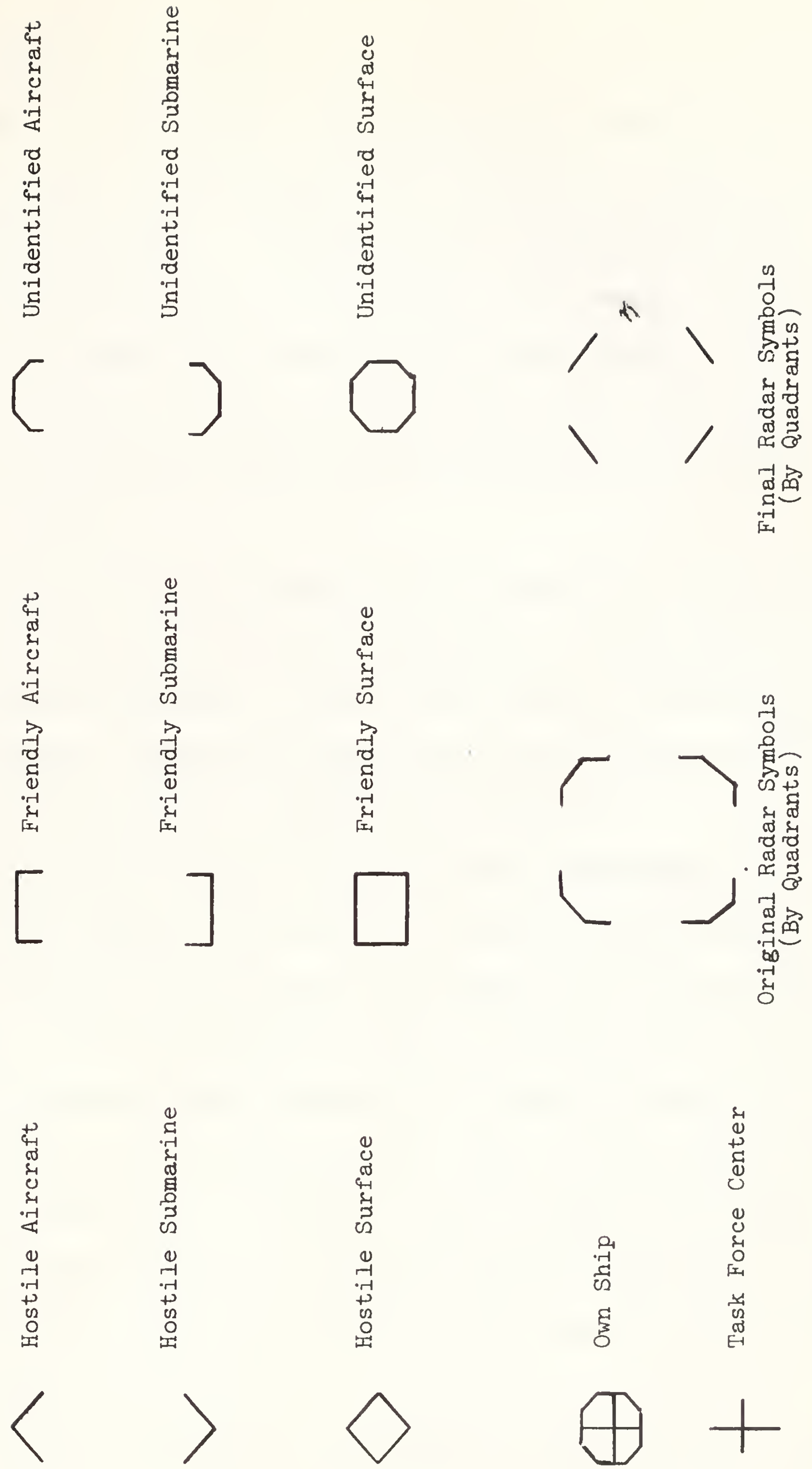


Figure 6. Display Symbols for DD 65

2.2.2 Subroutine TRAKGEN.

2.2.2.1 Early Philosophy. The track generator should be able to generate several tracks, each of which could be maneuvered in three dimensions at the will of the user. It should be able to produce tracks which could simulate aircraft, ships, and submarines. This placed a requirement for a very wide speed range capability on the track generator. The track generator should be capable of being coupled with a track parameter program which would give each track the characteristics of a particular aircraft or vessel.

It was decided to program the track generator as a very fine grain "actual" track generator. This would mean very high resolution capability in three dimensions. The user could then feed this data through a radar simulator program which would deteriorate the data to simulate a specific radar system in a particular environment.

The subroutine should be able to accept preprogrammed track commands and on-line commands from the user. It should then be expanded to allow track control from the tactical data system in the calling program.

2.2.2.2 Initial Programming. The first decision faced in programming the track generator was to determine the number of tracks to be generated. This number would be dependent on user need, available memory space, and program running time. It was felt that there should be a minimum of ten tracks available in order to provide reasonable flexibility. The number 15 was finally chosen mainly because it was convertible to a convenient power of two in binary form.

The first block diagram for TRAKGEN is shown in Fig. 7. Since the subroutine would be generating and storing information on several tracks

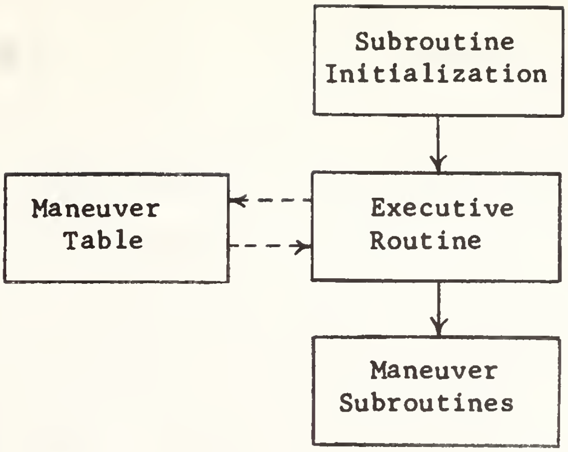


Figure 7. Initial Block Diagram for TRAKGEN

according to a maneuver (command) table, some initialization of matricies and flags would be necessary. It was decided to store the initial position, course, and speed for each track during this phase of the subroutine. Some signal was necessary from the calling program to allow initialization at the proper time. The real-time for the problem would have to be transferred with each call anyway, so a search was made for a way to utilize this as a flag. If the subroutine could be called initially with the real-time equal to zero, the initialization could be accomplished without fear of the flag reappearing later in the program and without the need for a separate flag word.

All track information is stored in a 15 x 6 matrix composed of 48 bit computer words. The track number corresponds to the row number in the matrix. The columns are arranged as shown in Fig. 8. Since the memory requirements for the program were unknown, it was felt necessary that track information be packed into the matrix words rather than stored in individual words. The number of bits assigned to each variable was arrived at by trial and error once the maneuver routines were programmed. The final size was determined by a requirement for slow speed tracking at an angle close to the cardinal compass points. Since the computation is done in part in fixed point arithmetic, a slow track would have to move far enough to change by one bit in order to maintain proper course. This was accomplished by balancing the overall range, the minimum track speed, and the calling period and arriving at the necessary bit resolution. The tracking criterion was that a one or two knot track should be able to track at an angle of five degrees off of a cardinal compass heading with a calling period of one second. At the same time, an overall range capability of $\pm$ 250 miles was desired. With the number of bits allotted, a two knot track can meet the five degree requirement with a one second calling period. A calling period of greater than one second allows a one knot track to meet the requirement. Similar reasoning was used to arrive at the altitude packing. The same number of bits were allotted to altitude as to X and Y position for the sake of symmetry. The track course was stored in tenths of degrees to allow turning rates of less than one degree per second with fixed point computation. Since a floating point capability was available in the computer, all quantities were stored in their original units; i.e., course is stored in tenths of

IPOSIT, IPOSIT1 Arrays:

X Position 24 bits	Y Position 24 bits

IPOSIT contains the X and Y position information for active tracks.

IPOSIT1 initially contains the initial X and Y position for preprogrammed tracks. After the track is started, IPOSIT1 is used to store position information for faded tracks.

IHEAD Array:

				Turn Rate 8 bits	Ordered Crs. 9 bits	Course 12 bits	Speed 12 bits

FFLAG (fade command flag) 1 bit
IFADE (fade flag) 1 bit
ISKIP (keyboard control flag) 1 bit
Maneuver Code 4 bits

JHEAD, JHEAD1 Arrays:

Ordered Altitude 24 bits	Altitude 24 bits

JHEAD contains altitude information for active tracks.

JHEAD1 initially contains the initial altitude for preprogrammed tracks. After the track is started, JHEAD1 is used to store altitude information for faded tracks.

KHEAD Array:

Spare 12 bits	Acceleration Rate 8 bits	Climb Rate 16 bits	Ordered Speed 12 bits

LHEAD Array:

				Maneuver Limit 18 bits	X Position 9 bits	Y Position 9 bits

Track Number 4 bits
Maneuver Code 5 bits
Spare 2 bits
LFLAG 1 bit (a 1 indicates LHEAD command present)

Figure 8. Subroutine TRAKGEN Array Formats

degrees rather than an artificial unit such as BAMS (Binary Angular Measure).

The executive routine divides non-initial calls of the subroutine into two parts:

(1) Update maneuver codes.

(2) Update positions.

The first section checks the maneuver table to see if any tracks are due to be maneuvered at this time. If there are maneuver commands, the proper changes are made in the maneuver code stored in the track matrix. Maneuver limits and rates are also stored in proper portions of the matrix.

The second section updates all 15 tracks according to the maneuver code stored in the track matrix. Initially, a print statement would print out what maneuver had been scheduled for each track. After the executive routine was functioning properly, the maneuver routines were inserted in place of the print statements.

The first maneuver routines were:

(1) Constant course routine.

(2) Turn routine.

(3) Climb routine.

(4) Speed routine.

No provision was made to perform more than one maneuver at a time in the initial programming of TRAKGEN.

The constant course routine takes the track speed per second times the calling period in seconds and computes incremental X and Y distances and adds these to the original position of the track.

The climb and speed routines are similar and simply increment altitude or speed linearly for the number of seconds in the calling period.

The turn routine was the last maneuver section programmed and was the most complicated due to the fact that a track can arrive at a given course by turning in either direction. Two methods of generating a curved track were studied. The first was a computation based on turning radius, bank angle, speed, and turn center position. [18]. The second method was a straight line approximation based on speed and turn rate only. After trying both methods, it was decided that the straight line approximation gave satisfactory results with less computation. The turn routine geometry is shown in Fig. 9. In order to insure stability and uniformity regardless of the calling period, the track is incremented in one second segments. Only the positions corresponding to a calling period time are transmitted to the calling program. If a turn will be completed in less than the number of seconds available in the calling period, the track is incremented in one second segments until just short of the ordered course. Then the track is set to the ordered course for the final increment. See Fig. 10.

Tests were run on the turn routine to check the validity of the approximation. A 1000 knot track was programmed to perform repeated 360 degree turns with a calling period of six seconds. A total of eight revolutions was made and a graph output showed that the track accurately retraced its path on each revolution. This indicates that the approximation results in stable arcs and that circles do not degenerate into logarithmic spirals.

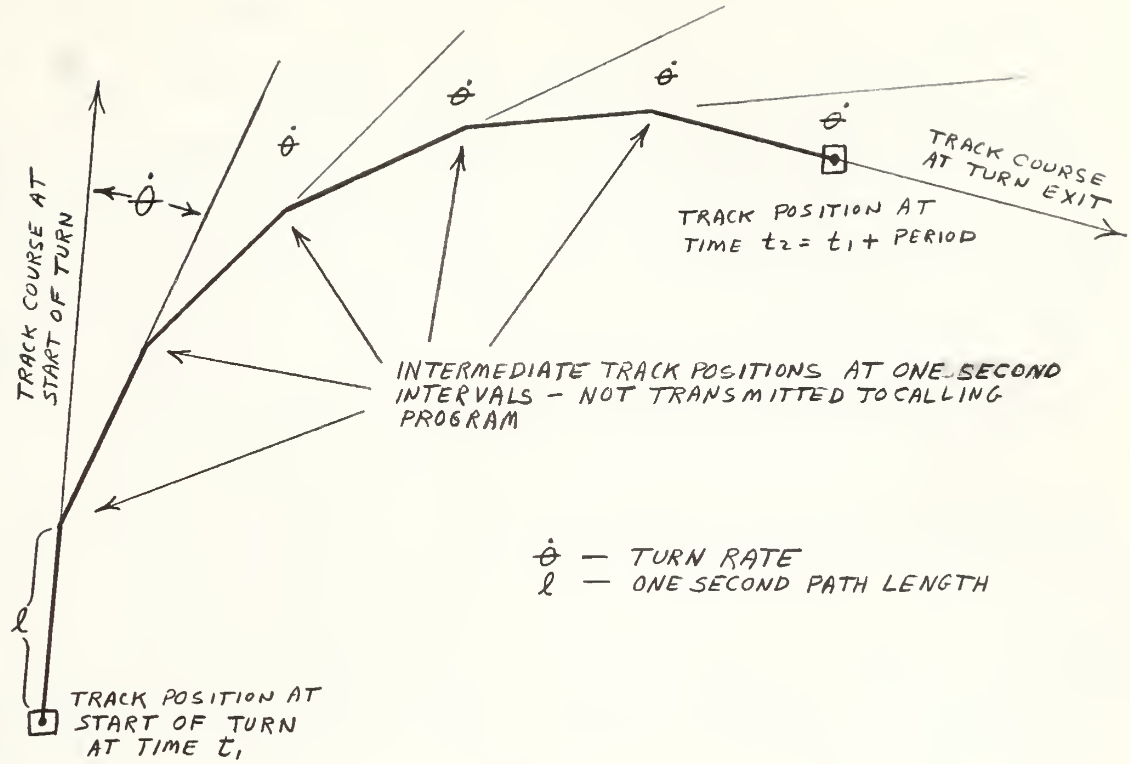


Figure 9. Unfinished Turn

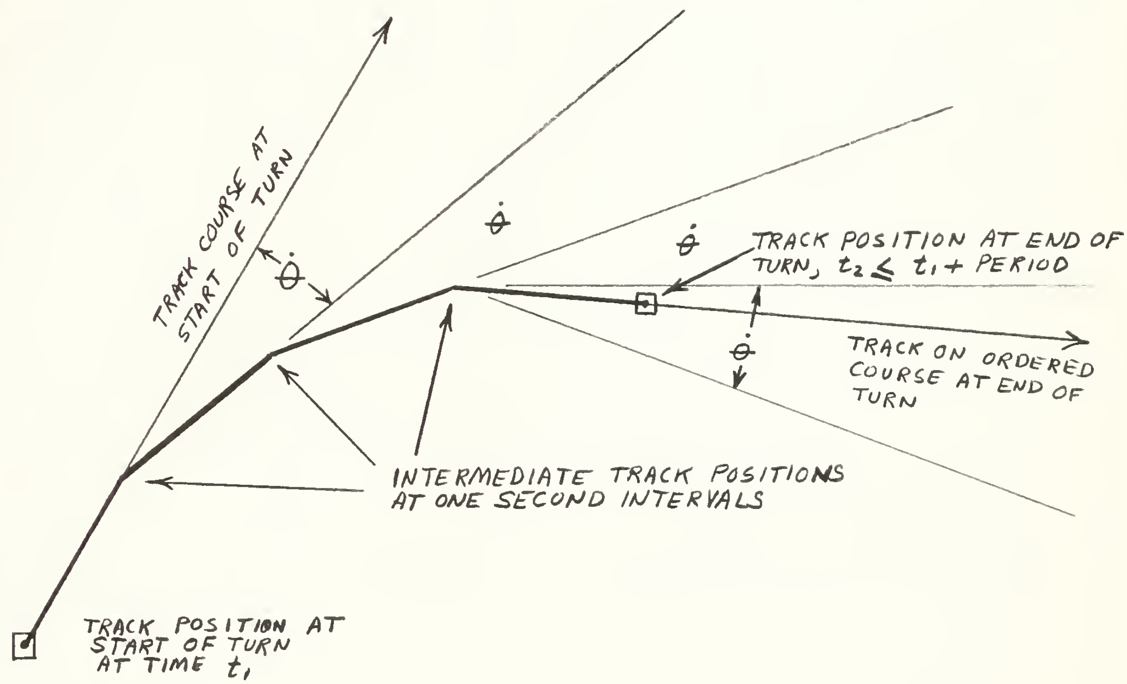


Figure 10. Completed Turn

2.2.2.3 Expanding the Program. At this point, the track generator was capable of generating tracks which could perform only one maneuver at a time. Some method had to be devised to allow multiple maneuvers such as a climbing turn. It was felt that a proper choice of maneuver code table and maneuver sequencing could solve this problem.

There are two possible methods of controlling the tracks by maneuver codes.

(1) Insert a code to initiate the maneuver and insert a separate code to stop the maneuver.

(2) Insert a code to initiate the maneuver along with the desired maneuver limit and rate.

The second method was chosen since it reduces the number of maneuver table entries by a factor of two. It also corresponds to the normal method of giving commands to operating units in the fleet. Many forms of maneuver code table were tried and discarded as being too cumbersome. There are three basic maneuvers, turn, climb, and accelerate. If these are equated to maneuver codes of two, three, and four, the simple addition of desired maneuvers will result in a distinct maneuver code for all possible combinations of maneuvers. These are shown below:

2 Turn (basic command).
3 Climb (basic command).
4 Accelerate (basic command).
5 Climbing turn.
6 Accelerate in a turn.
7 Accelerate in a climb.
9 Accelerate in a climbing turn.

To fit in with the addition or subtraction of maneuver codes, the code of zero was assigned to a constant course, speed, and altitude track.

These maneuver codes are internal to the program; the external codes being turn, climb, and accelerate only.

Since the only external codes being used were two, three, and four, this left all other unused numbers available for other functions. The external maneuver code decided upon is:

1 Start track.
2 Turn.
3 Climb/dive.
4 Change speed.
5 Fade.
6 Unfade.
8 Scrub track.

The fade command allows a track to disappear from the position array while still being capable of performing all maneuvers. The unfade command returns the track to the position array for normal display. The maneuver code seven is unused.

In the initial program, the maneuver table was read in during the initial call and stored in memory. This severely limited the size of the maneuver table. For this reason, the maneuver table was changed to a data deck table and only one maneuver entry at a time is stored in memory. Each maneuver table entry contains:

(1) Maneuver execution time.

(2) Track number.

(3) Maneuver code.

(4) Maneuver limit.

(5) Maneuver rate.

Besides being controlled from the programmed maneuver table, it was desired that tracks be able to maneuver according to the dictates of the user or according to the computations in the data process section

of the calling program. In order to avoid redundency, it was decided to convert the necessary commands into the same form as that read from a maneuver table data card and to insert these into the program in place of data card information. From this point on, the information is processed the same as a maneuver table entry.

Two sections in the program were developed to handle the interface between the input command and the main subroutine. The Keyboard Control Section is designed to handle commands from the DD 65 console keyboard. It performs checks on the command information to determine if the track called for is available for control. It also sets a flag in the track matrix to disallow further preprogrammed commands since they will be meaningless once the track has been displaced from its preprogrammed track. The one exception to this is the keyboard fade or unfade commands which do not alter the track path and, hence, do not disallow maneuver table commands. All tracks are available for keyboard control with the exception of preprogrammed tracks which have not yet been started by the maneuver table command.

The Data Process Section can control any number of tracks simultaneously. It processes the information from the tactical data section of the calling program into a form suitable to be fed directly into the Keyboard Command Section. Input command information flow is shown in Fig. 11.

2.2.2.4 Future Expansion. Subroutine TRAKGEN can be expanded in several areas. The input maneuver code could be changed so that input commands are inserted in plain language for convenience. The Hollerith word then would select the internal numerical maneuver code.

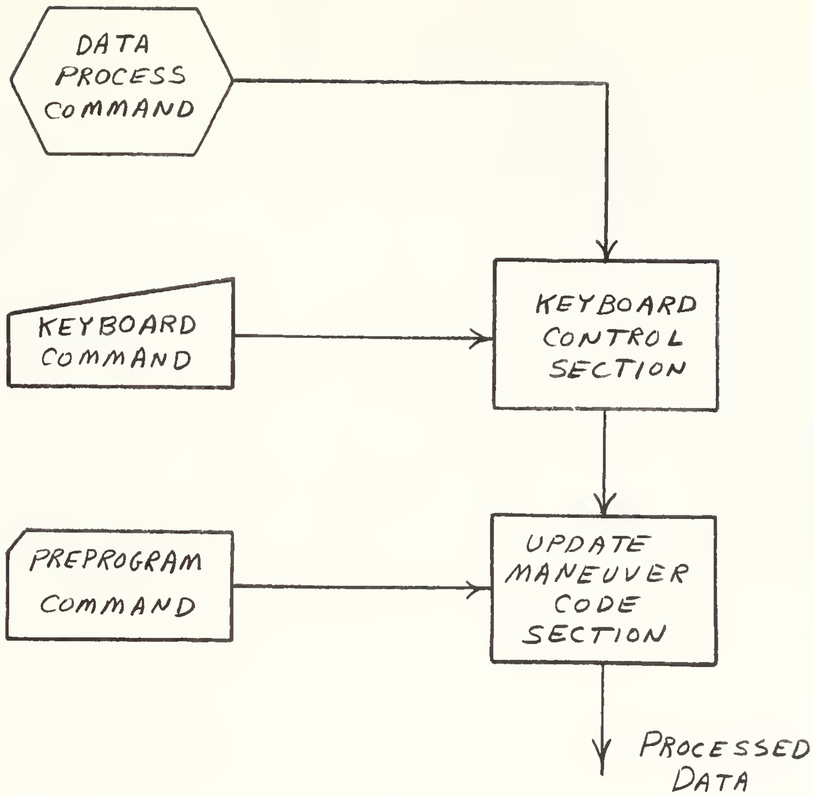


Figure 11. Command Flow Paths

Since the memory capacity of the 1604 has not been fully utilized, it is possible to expand the capability of TRAKGEN to handle more tracks or to have greater range capability. Trakgen requires an additional nine words for each track and the calling program SIMONE requires another 70 cells to allow expanded scratch pad space for each track. The range capability can be expanded by eliminating X, Y, and Z packing and using a complete word for each quantity. This would also allow greater resolution and consequently slower speeds and slower accelerations.

Another expansion of TRAKGEN would be the provision for negative track speeds to allow ships and submarines to back down. This would entail a reworking of the IHEAD array and the pack/unpack sections pertaining to speed and ordered speed.

To keep up with today's high speed aircraft, it would be well if the subroutine could accept maneuver rates in terms of maximum allowable g-forces on the aircraft and pilot. A section could be programmed which would compute these forces as functions of speed and altitude. In this context it would be well to make provision for entering aircraft speeds in mach number as well as well as knots. The g-force section would also prevent aircraft from performing maneuvers which in real life would be physically impossible. It is advisable when doing this to avoid including any classified aircraft parameters to avoid the necessity of classifying the program.

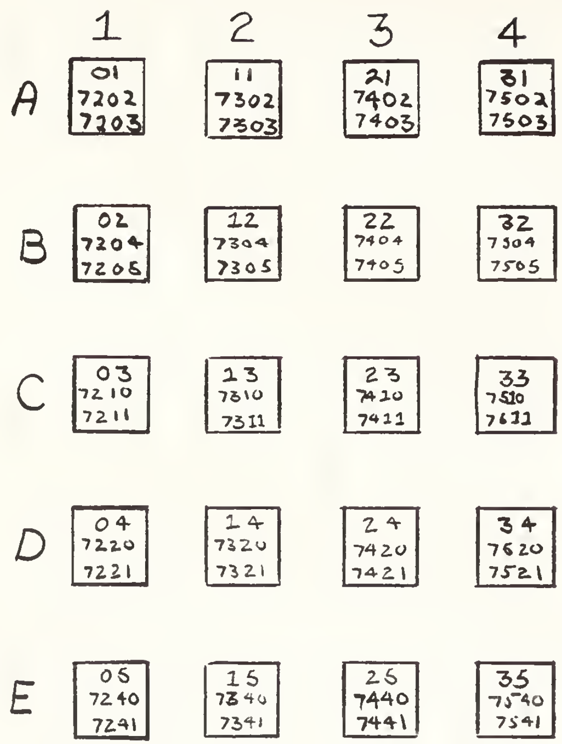
3. EQUIPMENT.

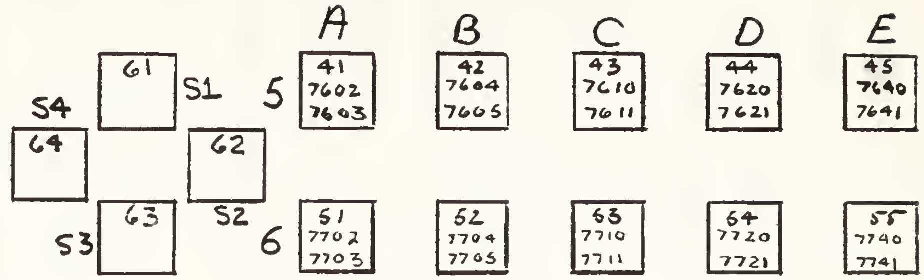
3.1 CDC 1604. The CDC 1604 is a stored-program, general-purpose digital computer, with a storage capacity of 32,768 48 bit words, designed to solve large scale scientific problems and to handle large volume data processing.

The input-output section of the computer handles the flow of information to and from the computer. The computer communicates with external equipment on one of seven channels. Six of these channels are independent buffer channels which provide for the normal input-output exchange of data. The input-output channels are paired, channels 1 and 2, channels 3 and 4, and channels 5 and 6. Channel 7 is a high-speed (4.8 usec per word) transfer channel on which up to eight different equipments may be connected. All communications between the CDC 1604 and the DD 65 were on this channel. By use of the CDC 160 computer as a buffer, communications can be established with the DD 65 using channels 5 and 6. This latter method is similar to that used in the Satellite System. 12 .

3.2 DD 65. The DD 65 consists to two units, the display console and the logic unit. The display console consists of two 12-inch electrostatically deflected tubes. Each tube has a usable display area of 8.5 x 8.5 inches. The left hand tube is normally used to display raw video from one of several radar inputs. The right hand tube is a non-radar tube used to display messages, symbols, etc.

There are several methods of inputting operator commands into the computer. They are by use of an alphanumeric keyboard, a general purpose keyboard, a track-ball controller, and a range switch.





Each of the squares represents a switch on Keyboard #2; the outer characters are switch designators. Depressing any switch generates the 2-digit code shown within the Corresponding square. The 4-digit codes inside the squares are computer codes that control the corresponding indicator lights. Even codes turn the indicators ON; the odd codes turn them OFF.

Figure 12. Switch Designations and Input/Output Codes for Keyboard 2

The logic unit consists of a 1024 24 bit word core memory. Each word is paired to give an effective 512 48 bit word memory. This memory can display data at very high refresh rates. There is a switch inside the logic cabinet that enables one to use only the lower half of memory, thus giving less flicker on the tubes. However, it was found that the flicker was not bad and the advantage of extra memory far out weighed the slight discomfort of the flicker.

3.2.1 Command Inputting Devices. The command inputting devices are designated as Keyboard 1, Keyboard 2, Track Ball, and Range Switch. Keyboard 1 is an alphanumeric typewriter keyboard. There are 64 keys keys on Keyboard 1 of which 60 represent different symbols and three editing keys, i.e., carriage return, space, and tab, and one unassigned key, (code 00).

Keyboard 2 is divided into three functional sections. See Fig. 12. The first section consists of 20 switch units arranged in a 4 x 5 matrix. This group has computer selected light indicators and provisions for insertion of separately coded switch matrix overlays. By use of these overlays, many different functions may be programmed. The second consists of 4 special switch units. These units are grounded to the 60 cps power supply. This enables the switch code to be sent to the computer 60 times a second without having to hit the key but one time. This enables this group of keys to move an editing symbol over the display surface. The third section is a group of 10 switches located over the alphanumeric keyboard. These have associated computer controlled light indicators, also. This set of switches is designed to perform functions common to all overlay programs.

The Track Ball is located to the right of the display tubes on the monitor table top. The X and Y coordinate information is available at all times to the computer and need only be selected.

The last command inputting device is not really a command inputting device, however, it was used for this purpose. This is the Radar Range Switch. This is an 8-position switch which is used to select the range of the display. The 8 positions, ranges and associated codes are shown in Fig. 13.

Switch Position	Code	Switch Position	Code
OFF	0	32	4
4	1	64	5
8	2	128	6
16	3	256	7

Figure 13. Range Switch Coding.

4. Usage.

4.1 General. This routine is intended for use with the normal library currently used by the Computer Facility. Complete compatibility has been maintained.

4.2 Program Format. This routine will normally be called from a reserve library tape. However, if the user wants to modify the program in order to use his own subroutines, the program will be operated under card control. In this case, the following is an example of the normal run using the CDC 405 card reader:

```
SET,5000,66.                          (Memory scratch pad)
CALL,1,FORTBIN.                       (Compiler)
FORTBIN,C,1.                          (Compiler program)
       PROGRAM SIMONE                 (Program)
                -
                -
                -
                -
       END
       END
(Blank card)
(Blank card needed for card reader)
SIMONE.
* Data
    -
    -
    -
    -
..
CONTROL,0.
```

To execute this program, load the card reader with the above cards and then type on the typewriter, "control,c,p.". This will start the card reader and all control statements will be printed on the CDC 1612 printer.

After the user has checked out his program, the following methods are suggested:

```
CALL,1,FORTBIN.
FORTBIN,C,1,4.
       PROGRAM SIMONE
                -
                -
                -
                -
       END
       END
(Blank card)
CONTROL,0.
```

This will cause SIMONE to be compiled and placed in library format on tape 4. A pre-compiled program is ready to run. Due to the fact that cells 5000_8 to 20300_8 are used as a scratch pad, SIMONE cannot be called and run. The following program will allow for the proper space being left for the scratch pad:

```
SET,5000,66.
CALL,1,FORTBIN.
FORTBIN,C,4.
       PROGRAM SIMTWO
       CALL SIMONE
       END
       END
(Blank card)
(Blank card)
SIMTWO.
*Data
   -
   -
   -
..
CONTROL,0.
```

This program reduces the recompile time from about eight minutes to approximately 18 seconds.

4.3 The Data Cards.

4.3.1 The first data card contains job identification. It will be reproduced on the graph output. Use columns 1 through 8. This identification could be in the form of a date, run number, etc.

4.3.2 The second and third data cards contain the necessary arguments for the graph outputs:

(1) XSCALE: Columns 6 through 11. X-scale in units per inch. The units are miles for the second data card and minutes for the third data card. The format is E6.0.

(2) YSCALE: Columns 13 through 18. Y-scale in units per inch. The units are miles for the second data card and feet for the third data card. The format is E6.0.

(3) IXUP: Columns 20 and 21. Distance in inches, of the X-axis from the bottom of the graph. Format I2.

(4) IYRIGHT: Column 23. Distance in inches, of the Y-axis from the left of the graph. Format I1.

(5) MODEXAX: Column 25. Determine the mode of the X-axis location 20 . Format I1.

(6) MODEYAX: Column 27. Determines the mode of the Y-axis location 20 . Format I1.

(7) IWIDE: Column 29. Width of graph in inches. Format I1.

(8) IHIGH: Columns 31 and 32. Height of graph in inches. Format I2.

(9) IGRID: Column 34. If IGRID = 1, a 1" x 1" will be superimposed on the graph. Format I1.

The second card furnishes the information for the X - Y position graph and the third card the information for the altitude vs. time graph.

4.3.3 The fourth data card contains SCALE. Use columns 1 through 10. Format F10.0. Due to the variation in altitude capabilities, provision has been made to scale the depth information. Any negative

altitude will be multiplied by SCALE. This will allow for easier interpretation of negative altitudes.

4.3.4 The fifth data card contains IP and IPP. Use columns 1 and 2 for IP and columns 5 through 10 for IPP. Format I2 and I6. IP is the period in seconds for calling TRAKGEN. All other executive routines are called at times based on this period. IPP is the number of seconds between points on the point plot. If IPP is zero, the point plot will not be plotted.

4.3.5 The sixth data card contains the number of preprogrammed tracks, NTRACKS. $0 \leq \text{NTRACKS} \leq 15$. Right justify in columns 1 and 2.

4.3.6 The seventh data card starts the initial position data. There must be one initial position data card for each preprogrammed track. The position data cards correspond to tracks numbered in consecutive increasing number with the last position being for track number 15. e.g., if three tracks are preprogrammed, the initial position cards are for tracks 13, 14, and 15 in that order.

(1) X-position in miles in columns 1 through 10. X-position may have up to four decimal places. Maximum X is ± 279 miles.

(2) Y-position in miles in columns 11 through 20. The same comments apply as for X-position.

(3) Altitude in integer feet right justified in columns 25 through 30. Maximum altitude is ± 98689 feet.

(4) Course in integer degrees right justified in columns 38 through 40.

(5) Speed in integer knots right justified in columns 47 through 50. Maximum speed is 4095 knots. If a zero speed is assigned, a non-zero course must be given or the track will fail to start when ordered.

4.3.7 This data card begins the maneuver table. There must be one data card for each maneuver. The last card in the table must be a non-executable card with a zero in column ten. There is no limit to the length of the maneuver table data deck. Entries are:

(1) Maneuver execution time in seconds right justified in columns 1 through 10. This time need not agree with a calling time since the maneuver will be executed as soon as real-time is equal to or greater than execution time.

(2) Track number right justified in columns 19 and 20.

(3) Maneuver code in column 30:

1 Start track.

2 Turn.

3 Climb/dive.

4 Change speed.

5 Fade.

6 Unfade.

8 Scrub track.

The first code for any track must be a start track command.

(4) Ordered course in integer degrees, ordered speed in integer knots, or ordered altitude in integer feet as appropriate. Right justify in columns 35 through 40.

(5) Maneuver rate right justified in columns 43 through 50.

Insert turn rate in degrees per second with up to one decimal place. $0.1 \leq$ turn rate ≤ 25.5.

Insert climb/dive rate in integer feet per minute. $43 \leq$ climb rate ≤ 65535. Insert acceleration rate in integer knots per second, maximum 255.

Any number of maneuvers can be scheduled for the same execution time. A track can be ordered to perform more than one maneuver at the same time by inserting each maneuver on succeeding cards, e.g.:

265	12	2	180	1.5
265	12	3	5000	1000

This orders track number 12 to start a climbing turn to 180 degrees at a 1.5 degree per second turn rate and to an altitude of 5000 feet at a climb rate of 1000 feet per minute starting at a time of 265 seconds.

4.4 DD 65 Operations.

4.4.1 Hardware. Before starting operations on the CDC 1604, the DD 65 must be ready in all respects to accept information from the CDC 1604. The following check off list is provided to aid in setting up the DD 65.

1. Check for interrupt/sense relay in place.
2. Check for cables in place.
3. Check for memory switch in WHOLE position.
4. Check for mode switch in 1604 only position.
5. Check for Power ON.
6. If the CDC 160 output/input cables are connected, insure that the CDC 160 is on. If the CDC 160 is not on or is inoperative, disconnect cables.

4.4.1.1 Interrupt/sense Relay. Several years ago, because of some logic problems, a modification to the logic cabinet was undertaken. This modification consisted of putting in series with the interrupt lines a relay, that was actuated by the power supply. It was customarily removed when not needed. This year the sense lines were also run through this relay. With the advent of the routine of the sense lines through this relay, it is possible to leave the switch in the OFF position without loading down the sense lines of the 1604. For this reason, the relay may now be left in permanently. However, before running for the first time it will be wise to check this relay. It is located on the inside bottom of logic chassis 2.

4.4.1.2 Cable Connections. Fig. 15 shows the proper connections for the cables. These cables are located at the base of the logic cabinet just below logic chassis 2.

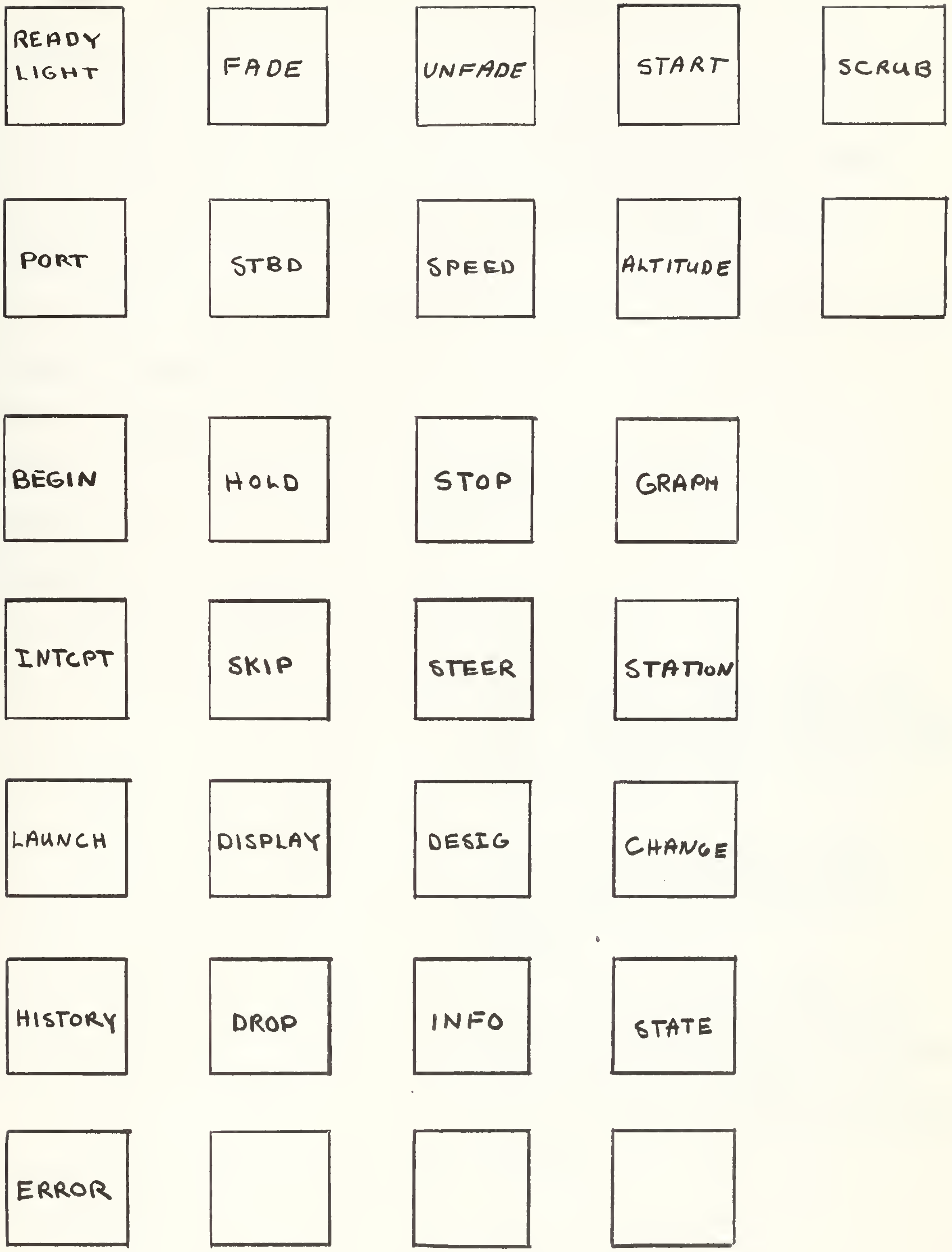


Figure 14. Keyboard 2 Functions

4.4.1.3 Memory Switch. A miniature toggle switch located on the inside edge of logic chassis 1 enables/disables the upper half of memory. Under normal use this switch is in the WHOLE or "down" position, to make maximum use of the memory.

4.4.1.4 Mode Switch. The Mode Switch is located on the front panel of the logic cabinet. This switch has three positions, "160", "Both", and "1604". The normal position for operation of SIMONE is in the "1604" position.

4.4.2 Console Operations. The following Sections deal with the callable routines available to the user.

4.4.2.1 CHANGE.

TASK — To maneuver one of the tracks,

EXAMPLE — CHANGE, N, M, L, R, X, Y, Z. $

ARGUMENTS — 7

N = Track number

M = Maneuver called--START, SCRUB, PORT, STBD, SPEED, ALTITUDE, FADE, OR UNFADE. If the Maneuver called is SCRUB, FADE, or UNFADE, then a period and $ follow M.

L = Limit--initial course for START and desired course for PORT or STBD. This argument is desired speed for SPEED and desired altitude for ALTITUDE.

R = Rate--initial speed for START, ten times the rate of turn for PORT or STBD, rate of climb/dive for ALTITUDE, and acceleration/deceleration for SPEED. If omitted when interpretated as a rate, then standard rates will be assumed.

X = Initial X position for START only.

Y = Initial Y position for START only.

Z = Initial Z position for START only.

4.4.2.2 DESIG.

TASK

To designate a track.

EXAMPLE

DESIG,TYPE.$

ARGUMENTS

2

TYPE = Type of target-- F will assign friendly symbol, H will assign hostile symbol, and if omitted, ownship symbol will be assigned.

HOOK = Target under hook will be assigned symbol designated by TYPE.

4.4.2.3 DISPLAY.

TASK

To change the tube on which raw radar data displayed.

EXAMPLE

DISPLAY.$

ARGUMENTS

0

This routine will cause the raw data to be displayed on the opposite tube from which it is now displayed.

4.4.2.4 DROP.

TASK

To cause a history track to be erased.

EXAMPLE

DROP.$

ARGUMENTS

1

HOOK = Target under hook will have its history track erased.

4.4.2.5 GRAPH.

TASK

To cause a graph to be plotted.

EXAMPLE

GRAPH.$

ARGUMENTS 0

This routine will cause a graph to be plotted that will cover the period from the last regular plot to the present time. The next regular plot will cover the time from now till then.

4.4.2.6 HISTORY.

TASK To cause history points to be displayed.

EXAMPLE HISTORY,A,M.$

ARGUMENTS 3

A = Number of points desired. This must be between 0 and 24. A number over 24 will be reduced to 24. If omitted, 12 points will be plotted.

M = Point mode indicator. If omitted, points will be plotted every period. M will cause points to be plotted every minute.

HOOK = Target under hook will have its history plotted.

4.4.2.7 HOLD.

TASK To cause the problem to pause.

EXAMPLE HOLD.$

ARGUMENTS 0

This routine puts the program into a wait loop. This wait loop is terminated upon the second striking of the HOLD switch.

4.4.2.8 INFO.

TASK To provide information on targets.

EXAMPLE INFO.$

ARGUMENTS 1

HOOK = Information about target under hook will be displayed. See Appendix I for format.

4.4.2.9 INTCPT.

TASK	To initiate an intercept.
EXAMPLE	INTCPT,C.$
ARGUMENTS	2
	C = CAP number of interceptor
	HOOK = Target under hook will be intercepted by CAP

4.4.2.10 LAUNCH.

TASK	To launch a CAP.
EXAMPLE	LAUNCH,C,R,B,T.$
ARGUMENTS	4
	C = CAP number of CAP to be launched
	R = Range from the TASK FORCE center of CAP station
	B = Bearing from the TASK FORCE center of CAP station
	T = Type of CAP (2 through 7)
	If arguments R and B are omitted, Station will be assumed to be under hook. If argument T is omitted, type 7 interceptor is assumed. To designate type and use hook for station, zeros must be placed in R and B.

4.4.2.11 SKIP.

TASK	To cause an intercept to be skipped.
EXAMPLE	SKIP,C.$
ARGUMENTS	1
	C = CAP number of intercepting CAP

4.4.2.12 STATE.

TASK — To display the State of the CAP.

EXAMPLE — STATE,C.$

ARGUMENTS — 1

C = Cap number. State format in Appendix I.

4.4.2.13 STATION.

TASK — To Station a CAP.

EXAMPLE — STATION, C,R,B.$

ARGUMENTS — 3

C = CAP number

R = Range from TASK FORCE center of CAP station

B = Bearing from TASK FORCE center of CAP station

If R = 0, then station will be assumed to be under the hook.

4.4.2.14 STEER.

TASK — To cause a CAP to be sent to TASK FORCE center.

EXAMPLE — STEER,C.$

ARGUMENTS — 1

C = CAP number

4.4.2.15 STOP.

TASK — To terminate the problem.

EXAMPLE — STOP.$

ARGUMENTS — 0

4.4.2.16 BEGIN.

TASK	To start the problem.
EXAMPLE	BEGIN.$
ARGUMENTS	0

This routine starts the problem and the number of tracks under user control is displayed on the DD 65.

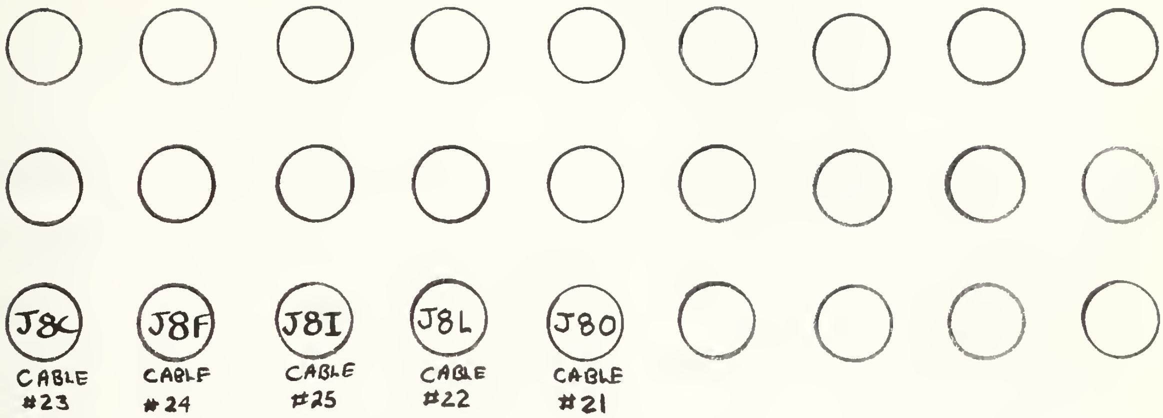


Figure 15. Cable Connections

5. Equipment and Logical Difficulties.

Some difficulties were encountered which were the result of both equipment logical short comings and equipment interaction.

5.1 DD 65 Display Console. Keyboard 1, the alphanumeric keyboard, when selected for input remains selected even after the input has been completed. If the keyboard were again selected for input, the processor is indefinitely delayed when the input is attempted. This limitation is the result of the cumbersome electromechanical keying system used, and has been previously avoided by selecting another display unit external function code. The deselect code previously used was SELECT RADAR TARGET DATA TO AUXILIARY EQUIPMENT. This code seemed to work well and was used to positively deselect both Keyboards 1 and 2.

It was also found that when one used the code SELECT RADAR RANGE SWITCH followed by SELECT TRACK BALL "X" that on some occasions the computer would be delayed indefinitely trying to input the X-cordinate. On the time the 1604 was not delayed, the next time an input from Keyboard 1 was attempted the code 760 was transferred to the computer. To correct for this the code for SELECT RADAR TARGET DATA TO AUXILIARY EQUIPMENT was used to deselect again. It appears that whenever an input code is selected it takes an output code to deselect the input code and vice versa. This points out the fact that there is no single code that deselects previously set selects.

5.2 CDC 1604. At this time there still remains an interaction between the 1604 and the DD 65 that has not been explained. Occasionally, during an update of the DD 65's memory or during an erasing of the DD 65's memory, the 1604 ends up with a 00007 in the upper address portion of

cell 00007. Since the upper instruction is an unconditional jump, the computer just sits there looping on the same cell. The logic prints show this to be impossible and the logic cards that actuate the interrupt lockout have been changed to no avail. At this time there is no answer.

6. Conclusions and Acknowledgments.

6.1 Conclusions. A general purpose track generator has been implemented based on the following concepts:

(1) Suitable for on-line simulation as well as system analysis studies.

(2) Applicable to both anti-air and anti-submarine warfare operations.

(3) Provision for operator interaction made available for exercising "console control" in gaming situations.

The tracking package provided is intended to be used as a basic tool in:

(1) Computer simulation analysis of weapons systems performance, e.g., interceptor-missile mission.

(2) Illustration and evaluation of combat direction system functions, e.g., NTDS, ATDS.

(3) Any system studies requiring complex track motion data on multiple targets, e.g., track-while-scan radar system performance.

The following major routines have been fully checked out under operating conditions:

(1) Subroutine TRAKGEN (Track Generator).

(2) Executive Control Routine.

(3) Display Subprogram.

(4) Radar Processor Subprogram.

(5) Keyboard Command Routine (Program Control Routine).

(6) Graph Output Routine.

(7) All keyboard callable routines except the HISTORY Routine.

In addition, an Aircraft Control Subprogram has been incorporated into the program listing. While reasonably close to final form, it has not yet been successfully tested with the system.

In the course of the work several interface problems on the 1604 - DD 65 channel 7 hookup were brought to light and corrected. These problems were:

(1) The DD 65, when turned off, loading the sense lines of the CDC 1604.

(2) The CDC 1604 being unable to input information from the track ball without first positively deselecting any previous input selection made by the CDC 1604.

(3) Incorrect cable listings in [10] for channel 7 hookup.

In addition, the following engineering difficulties have appeared but have not been corrected:

(1) An apparent random failure of the interrupt lockout feature of the CDC 1604 resulting in the computer remaining in a loop executing the instruction "SLJ (7)" while PAR equals 00007.

(2) Apparent incomplete decoding of certain external function codes or "leaking" of other signals into the external function lines of the DD 65 resulting in random selects of the computer controlled light indicators.

(3) Sensitivity of data transfer to timing adjustments in the CDC 1607 tape units resulting in parity errors if they are out of tolerance (Not corrected, but bypassed by use of memory as a scratch pad).

Upon analysis of the data obtained over the past five months, the duty cycle has been calculated to range from 0.18 to 0.26. The smaller duty cycle occurred during a 15 track problem. These figures were a result of the duty cycle routine embedded in the program. They closely correlate with theoretical data. The duty cycle indicates that the only constraint on expanding the program is that of computer memory size.

6.2 Acknowledgments. The authors wish to express their appreciation to Professor Mitchell L. Cotton for his aid and encouragement in the preparation of this thesis. In addition, the authors would like to thank Major Thomas Kauffman, USMC for his liason with Litton Systems, Inc., Data Systems Division who provided us with much information concerning simulation and smoothing systems; Lieutenant Commander J. V. Reynolds, USN, Fleet Computer Programming Center Pacific, who provided us with data concerning the Naval Tactical Data System and their approach to the simulation problem; Mr. Leon Spors and Mr. John Plesac of Control Data Corporation for their assistance in studying the interface problem; and Mr. Walter Landaker for his excellent maintenance of the DD 65.

BIBLIOGRAPHY

1. Benedict, T. R. and G. W. Bordner. Synthesis of an Optimal Set of Radar Track-While-Scan Smoothing Equations. IRE Transactions on Automatic Control, v. 7, July 1962: 27-32.

2. Borden, E. L. and A. C. Casciato. BCNTDS. Project Report of Digital Control Laboratory, USNPGS, May 1964.

3. Borkum, D. B. Smoothing-Predicting Sample Data. Electronic Industries, v. 9, September 1963: 81-85.

4. Control Data Corporation. Control Data 1604 Computer: FORTRAN System. Control Data Corporation publication 087A, 1961.

5. Control Data Corporation. Control Data 1604 Computer: Programming Manual. Control Data Corporation publication 167b, August 1962.

6. Control Data Corporation. Control Data 1604 Computer: Programming Training Manual. Control Data Corporation publication 015A, November 1961.

7. Control Data Corporation. FORTRAN-60 Operations. Control Data Corporation publication SPD-01, July 1963.

8. Cotton, M. L. Tactical Simulation Methods. Eighth Navy Science Symposium, May 1964.

9. Data Display Incorporated. Data Display Model dd65 Instruction Manual, Vol. 1.

10. Data Display Incorporated. Data Display Model dd65 Instruction Manual, Vol. 2.

11. Fluhr, F. R. NRL Report 5645, Naval Data Handling System Circuits.

12. Leach, G. H. and A. J. Perrella. A Satellite Computer System for On-Line Analysis, Control, and Display. USNPGS Thesis, June 1964.

13. Ogden, D. J. and J. S. Smith. Preparation of a Track Generator. Project Report of Digital Control Laboratory, USNPGS, May 1964.

14. Pickering, G. E., E. G. Mutschler, and G. A. Erickson, Multi-Computer Programming for a Large Scale Real-Time Data Processing System. Proceedings - Spring Joint Computer Conference, 1964: 445-461.

15. Prokop, J. S. Remote Information Retrieval and Display. USNPGS Thesis, June 1964.

16. Rossheim, R. J. Proposed American Standard Flowchart Symbols for Information Processing. Communications of the ACM, v. 6, October 1963: 601-604.

17. Stein, M. L. and W. D. Munro. Computer Programming, A Mixed Language Approach. Academic Press, 1964.

18. United States Naval Academy, Department of Aviation. Introduction to Applied Aerodynamics. 1956.

19. United States Naval Postgraduate School. MACHINE RESIDENT, Monterey version of MACHINE RESIDENT. April 1965.

20. Ward, J. R. General Graph Output Subroutine. Writeup available from Computer Facility, USNPGS, February 1964.

APPENDIX I - A

1. Identification.

Title: SIMONE

Category: Control Routine

Programmer: E. L. Borden

Organization: U. S. Naval Postgraduate School

Date: April 1965

2. Purpose.

This control routine provides a means to control a simulated tactical data system from the DD 65.

3. Executive Called Subprograms.

3.1 1F-Display Subprogram. This program calls PRINT and causes both raw and smoothed data to be displayed on the DD 65.

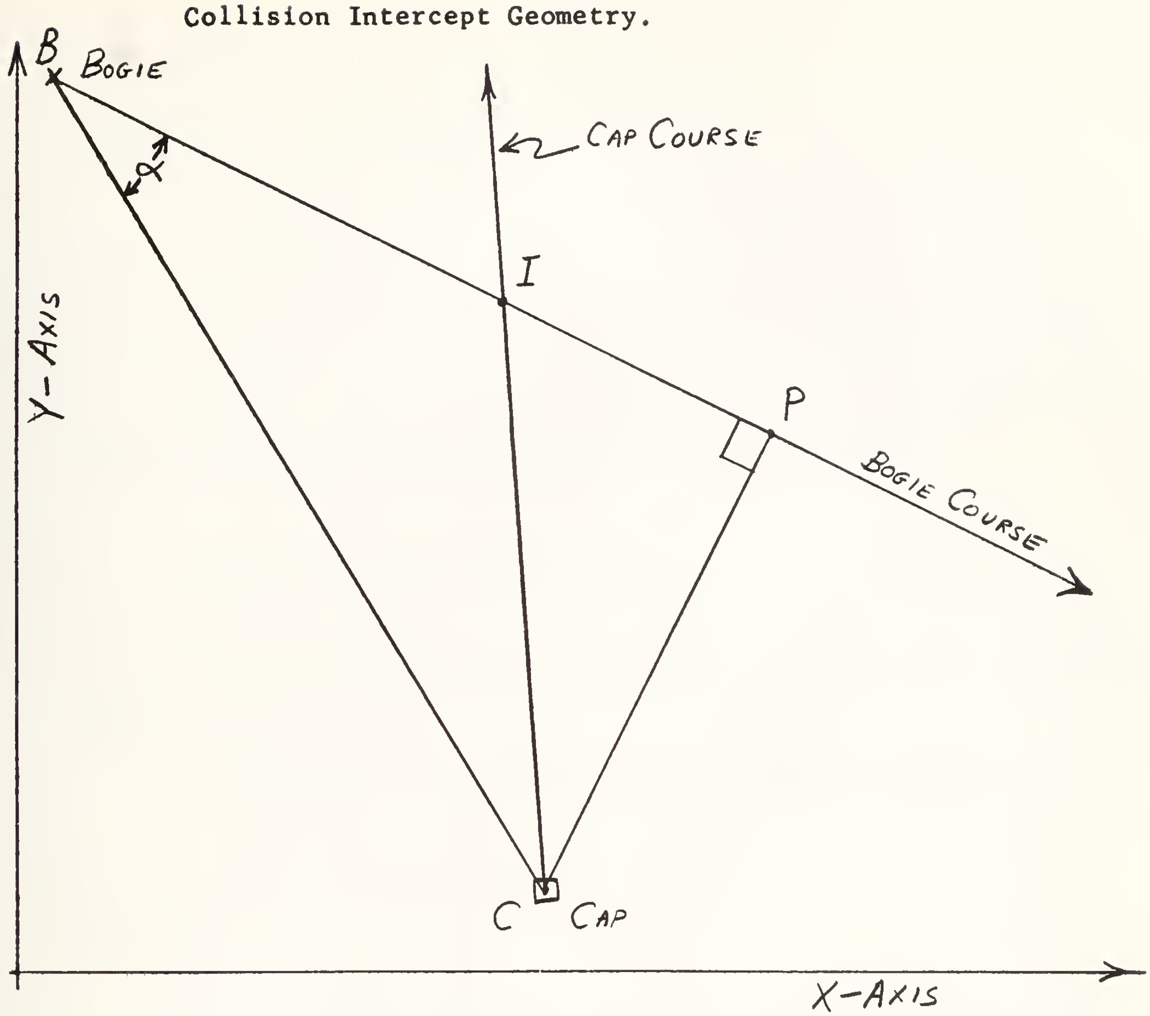
3.2 1G-Radar Processor. This subprogram uses the basic "alpha-beta" smoothing equations to process the radar data. The equations can be found in [1].

3.3 1GR-10 Minute Graph Subprogram. This program reads from the memory scratch pad and processes the data for calling DRAW to output the permanent graph record.

3.4 1INT-Aircraft Control Subprogram. This program performs the bookkeeping on fuel and ammunition loading for friendly aircraft and allows for control of these aircraft by the data process section. See Fig. I-1 for the collision intercept geometry.

APPENDIX I

Collision Intercept Geometry.



Alfa = Absolute value of Bogie Course minus Cap Bearing from bogie.

I = Point of intercept.

CP = N (Perpendicular distance from Cap to Bogie track).

BP = M (Distance from Bogie to point P).

V_b = Velocity of Bogie.

V_c = Velocity of Cap.

Figure I-1

APPENDIX I

In order for the Cap to intercept the Bogie at point I, the following equations must be true:

$$BI = V_b t \quad \& \quad CI = V_c t$$

We assume the following equation to be true. If it is not the case, BI will be negative and a collision intercept will be impossible since the intercept will take place somewhere to the right of P.

$$(CI)^2 = (M - BI)^2 + N^2$$

Solving the above equation simultaneously we end up with a quadratic equation in BI.

$$(V_c - V_b)^2 \cdot (BI)^2 + (2MV_b^2) \cdot (BI) - V_b^2 (M^2 + N^2) = 0$$

Solution:

$$(BI)^2 + \frac{2MV_b^2}{(V_c^2 - V_b)^2} \cdot (BI) - \frac{V_b^2}{(V_c^2 - V_b)^2} \cdot (M^2 + N^2) = 0$$

$$BI = \frac{V_b^2 M}{(V_c^2 - V_b^2)} \cdot \left[-1 \pm \sqrt{1 + \left(1 + \left[\frac{N}{M}\right]^2\right)\left(\left[\frac{V_c}{V_b}\right]^2 - 1\right)} \right]$$

This is the basic equation that must be solved in order to determine if an intercept can be made. If two positive values of BI exist, the smaller of the two will be the shortest intercept.

3.5 1K-Keyboard Command Subprogram. This program sets the range scale factor and positions the hook on the right hand tube of the DD 65. This routine also checks to see if there are any commands from the DD 65 to be executed. If there are commands to be executed then control is passed to the 1P-Program Control Routine.

3.6 1TG-Track Generator Calling Subprogram. This program calls TRAKGEN, unscales and unpacks the X, Y, and Z positions and records these values on the memory scratch pad for future graphs.

4. DD 65 Called Routines.

4.1 1C-Change Routine. This is a seven argument routine, which is used to control tracks.

4.1.1 Arguments.

1A	Track Number
2A	Maneuver Called
3A	Variable Argument (Limit)
4A	Variable Argument (Rate)
5A	X Position
6A	Y Position
7A	Z Position

The variable arguments are interpreted according to the maneuver called. The interpretations are:

Maneuver	3A	4A
START	Course Desired	Speed Desired
SCRUB	Ignored	Ignored
PORT	Course Desired	10 x Turn Rate
STBD	Course Desired	10 x Turn Rate
SPEED	Speed Desired	Acceleration
ALTITUDE	Altitude Desired	Dive/Climb Rate
FADE	Ignored	Ignored
UNFADE	Ignored	Ignored

Arguments 5A, 6A, and 7A are used only when the maneuver START is called, at all other times they are ignored.

When the argument is a rate (4A with PORT, STBD, SPEED, and ALTITUDE), omitting the argument will cause standard rates to be entered. These rates are:

PORT	1.5 Degrees per second
STBD	1.5 Degrees per second
SPEED	10 knots per second
ALTITUDE	5000 feet per minute

4.2 1D-Display Routine. This is a zero argument routine that complements the D-Flag. The Display Subprogram uses the D-Flag to determine on which tube the raw data is to be displayed. A negative D-Flag will cause the raw data to be displayed on the right hand tube while a positive D-Flag will display the data on the left hand tube.

4.3 1J-Designate Target Routine. This routine is a two argument routine that calls the 3J0 routine to determine the second argument. The target under the hook will be designated according to the following:

Argument	Designation
0(zero)	Own Ship
H	Hostile Track
F	Friendly Track

When a track is designated friendly and it is an aircraft, its interceptor type is determined by the condition of the least significant bit in cell 50 (Job time clock). A 0 assigns a type 7 and a 1 assigns a type 3.

4.4 1L-History Routine. This routine is a three argument routine. The first argument is the number of points desired. This argument may be any number between 1 and 24. Any number greater than 24 will be reduced to 24. If the argument is omitted, then 12 points will be plotted. If the second argument is anything other than zero, then the first argument will be interpreted as the number of minutes of past history with one point for each minute. The third argument, the track number, is filled by the Hook Routine. Up to two tracks may have past history plotted. If a third track is requested, the first track will be dropped.

4.5 1PQ-Hold Routine. This is a zero argument routine that places the program in a wait loop. The second hitting of the hold switch will resume the program.

4.6 1Q-Intercept Command Routine. This is a two argument routine that calls the Hook Routine to fill the second argument, the BOGEY track number. The first argument is the CAP number of the interceptor. This routine sets the necessary flags which signal the Aircraft Control Subprogram to initiate an intercept.

4.7 1R-Drop History Routine. This is a one argument routine that has its argument, the track number, filled by the Hook Routine. The history being displayed on the hooked track will be dropped upon execution of this routine.

4.8 1S-Stop and Critique Routine. This is a zero argument routine that causes the problem to be terminated and the final graph to be plotted.

4.9 1ST-State Routine. This is a one argument routine which causes one of the following messages to be displayed on the DD 65 depending upon the actual state of the interceptor.

```
STATE REPORT ON CAP   xx

       PLUS
  AMMO MINUS / OXYGEN PLUS / FUEL xxxx POUNDS / TIME xxxxZ.
       ZERO

  TYPE  x
```

4.10 1TT-Information Routine. This is a one argument routine that has its argument filled by the Hook Routine. The following are examples of the types of information messages that appear:

```
         INFORMATION
   TRACK  NR  14
   BOGEY  E3
   COURSE  350, SPEED 450, ALTITUDE 35000.
   RANGE:   120 MILES.
   BEARING:  80 DEGREES.
   FROM TASK FORCE CENTER

         INFORMATION
   TRACK  NR  13
   SKUNK  D
   COURSE  35, SPEED 13.
   RANGE: 12000 YARDS.
   BEARING  270 DEGREES.
   FROM OWN SHIP
```

INFORMATION
TRACK NR 3
GOBLIN E
COURSE 270, SPEED 4, DEPTH 200.
RANGE: 3000 YARDS.
BEARING: 5 Degrees
FROM OWN SHIP

Speeds are in knots and depth is in feet.

4.11 1U-Steer Command Routine. This is a one argument routine which sets the necessary flags which signal the Aircraft Control Subprogram to steer the CAP specified by the argument.

4.12 1V-Station Command Routine. This is a three argument routine. The arguments are:

1A	CAP number
2A	Range of Station from TF Center
3A	Bearing of Station from TF Center

If argument 2A is zero or omitted, then the station will be assumed to be under the hook. This routine sets the necessary flags which signal the Aircraft Control Subprogram to station the CAP.

4.13 1W-Skip Command Routine. This is a one argument routine that sets the necessary flags which signal the Aircraft Control Subprogram to cause the CAP listed in the argument to disengage and return to station.

4.14 1X-Launch CAP Command Routine. This is a four argument routine. The arguments are:

1A	CAP number
2A	Range of Station from TF Center
3A	Bearing of Station from TF Center
4A	Type of interceptor

If argument 2A is zero or omitted, then the hook is assumed to be over the station. This routine sets the necessary flags which signal the Aircraft Control Subprogram to launch a CAP and send him to the designated station.

4.15 1Z-Begin Routine. This is a zero argument routine that starts the program and calls TRAKGEN with time equal to zero. This routine also causes the number of tracks under Keyboard control to be displayed on the DD 65. The format of the message is:

07 TRACKS UNDER YOUR CONTROL.

5. Control Routines.

5.1 1T Executive Control Routine. This routine follows the basic philosophy of the executive control routine described in Section 2.2.1.1 of the main body and in [14] .

5.2 1P-Program Control Routine. This routine functions as the basic operator-computer communications link, decodes control statements, packs arguments for routines; then passes control to the routine designated.

6. Miscellaneous Routines.

6.1 3JO-Hook Routine. This routine is used by many other routines and cannot be called separately. It hooks any track within the range acquisition gate. The range acquisition gate is variable and changes depending upon the range scale setting. The gate sizes are:

Range Scale (miles)	Gate
256	8.0 x 8.0
128	4.0 x 4.0
64	2.0 x 2.0
32	1.0 x 1.0
16	.5 x .5
8	.25 x .25
4	.125 x .125

There is no acquisition gate for altitude. If two or more targets are within the X and Y gates but at different altitudes, the target with the lowest track number will be hooked. If there is no target within the gate, the error message "TRACK NOT HOOKED" will be displayed and control will return to the Executive Control Routine.

6.2 1Y-Find CAP Number Routine. This routine is used in conjunction with any routine that uses CAP number instead of track number as an argument. This routine searches the TRACK array to determine the track number of the CAP. If the result of the search is negative, the message "CAP xx NOT AIRBORNE" is displayed on the DD 65.

6.3 1ZZ-CAP Number and Station Routine. This routine calls the CAP Number Routine to determine track number and then computes the X and Y coordinates of the designated station.

7. Error Stops.

An incorrect number of tracks, i.e. number greater than 15, on the sixth data card will cause the program to execute a selective stop. The following error message will be displayed.

NTRACKS $\geq$16 CHECK DATA DECK AND TRY AGAIN

The program may be started again after correcting the data deck by hitting RUN on the CDC 1604.

8. Error Messages.

Message	Meaning
PROGRAM NOT CALLED	The program called was not listed in the Program Name Table.
TOO MANY ARGUMENTS	Too many arguments were listed for the program called or an unauthorized argument was used (i.e. $\geqslant$, :, $\wedge$, etc.)
2ND ARGUMENT ILLEGAL	An illegal maneuver was called for or the argument for the Desig Routine was not on H, P, or 0 (zero).
TARGET NOT HOOKED	No target was within the range acquisition gate.
NTRACKS $\geqslant$ 16 CHECK DATA DECK AND TRY AGAIN	See 7.

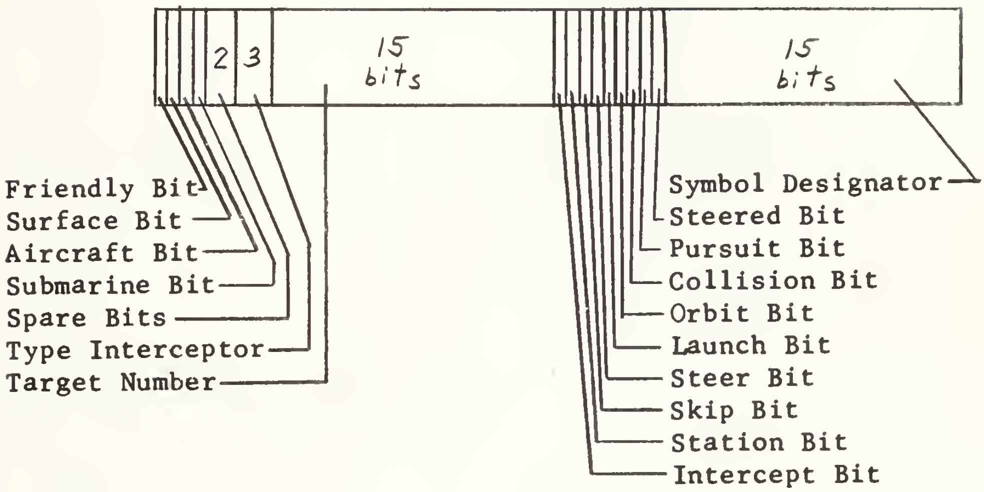
9. Information Messages from Aircraft Control Subprogram.

Message	Meaning
CAP xx UNABLE TO INTERCEPT - FUEL	CAP does not have enough fuel to reach the intercept point
CAP xx UNABLE TO INTERCEPT - AMMO	CAP does not have any ammunition left aboard
CAP xx LOW STATE	CAP has just enough fuel to return to carrier
CAP xx MAYDAY...MAYDAY...	CAP out of fuel, Ditching
CAP xx INTERCEPTING - UNABLE TO RETURN	CAP does not have sufficient fuel to return to the carrier after making an intercept

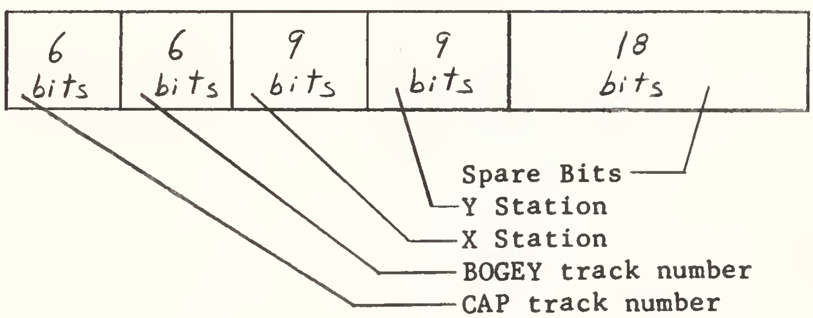
CAP xx FAMISHED	CAP has no station assigned
CAP xx POOR INTERCEPT	CAP unable to make collision intercept and is in a pursuit intercept
SPLASH BOGEY Exx	CAP destroyed BOGEY Exx
HEADS UP BOGEY Exx	CAP unable to destroy BOGEY Exx

10. Array Formats.

10.1 TRACK Array.



10.2 TRACK1 Array.



Interrupt Routine

Interrupt Processor

Clock Processor

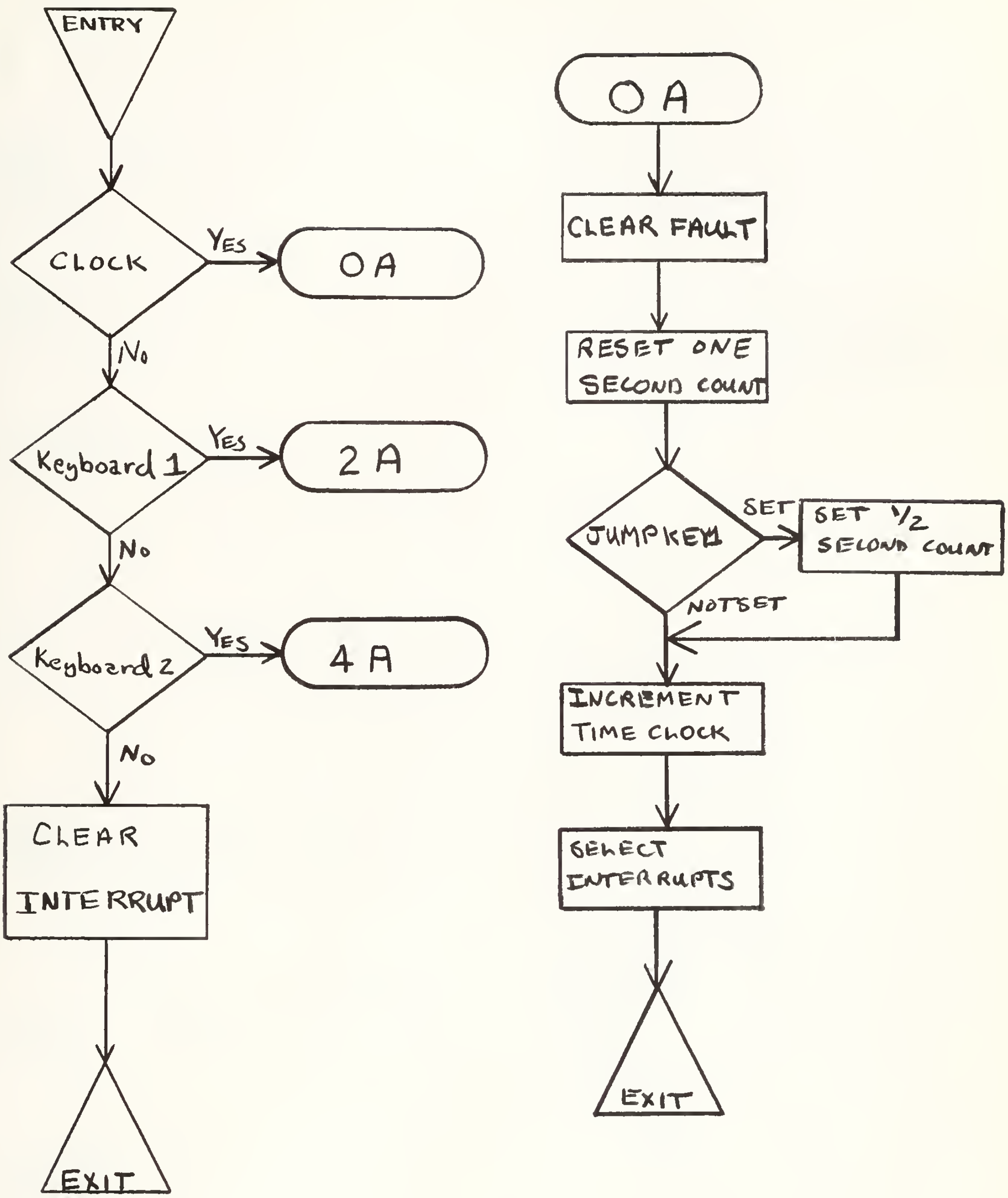


Figure I-2

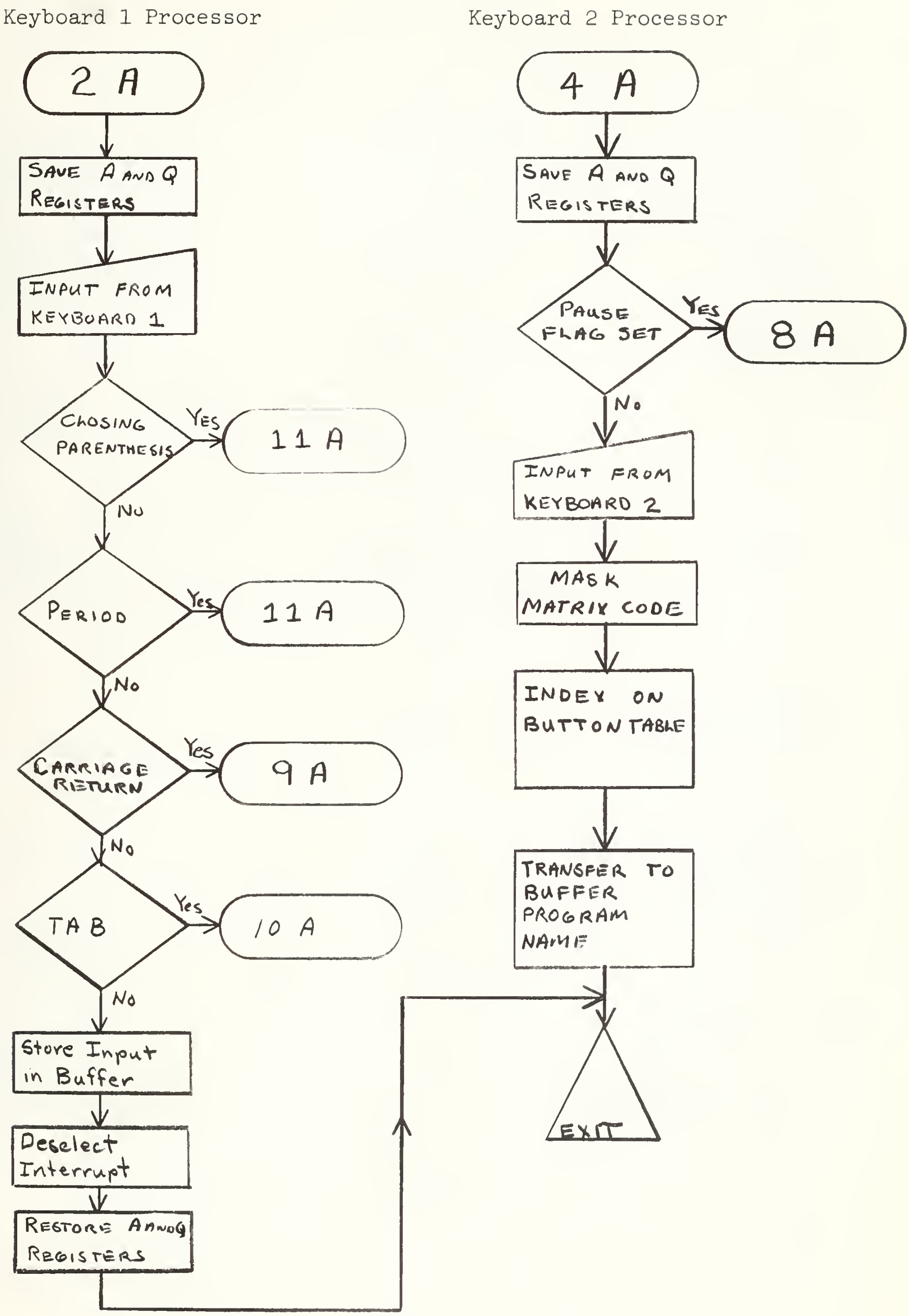

Keyboard 1 Processor
2 A
SAVE A AND Q REGISTERS
INPUT FROM KEYBOARD 1
CLOSING PARENTHESIS
Yes
11 A
No
PERIOD
Yes
11 A
No
CARRIAGE RETURN
Yes
9 A
No
TAB
Yes
10 A
No
Store Input in Buffer
Deselect Interrupt
RESTORE A AND Q REGISTERS
Keyboard 2 Processor
4 A
SAVE A AND Q REGISTERS
PAUSE FLAG SET
Yes
8 A
No
INPUT FROM KEYBOARD 2
MASK MATRIX CODE
INDEX ON BUTTON TABLE
TRANSFER TO BUFFER PROGRAM NAME
EXIT

Figure I-3

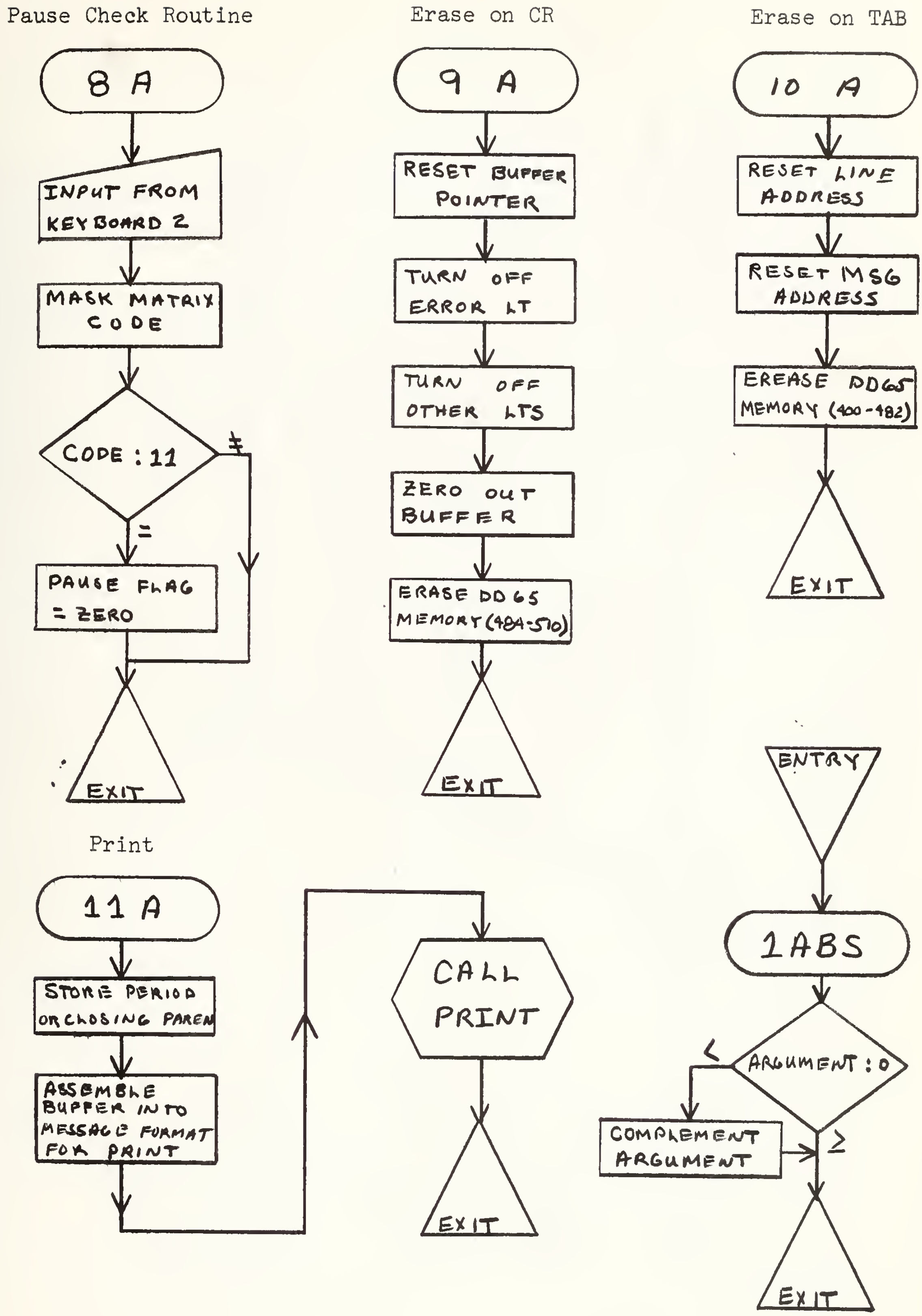

Pause Check Routine
Erase on CR
Erase on TAB
8 A
INPUT FROM KEYBOARD 2
MASK MATRIX CODE
CODE : 11
≠
=
PAUSE FLAG = ZERO
EXIT
9 A
RESET BUFFER POINTER
TURN OFF ERROR LT
TURN OFF OTHER LTS
ZERO OUT BUFFER
ERASE DD 65 MEMORY (484-510)
EXIT
10 A
RESET LINE ADDRESS
RESET MSG ADDRESS
EREASE DD65 MEMORY (400-482)
EXIT
Print
11 A
STORE PERIOD OR CLOSING PAREN
ASSEMBLE BUFFER INTO MESSAGE FORMAT FOR PRINT
CALL PRINT
EXIT
ENTRY
1ABS
ARGUMENT : 0
<
≥
COMPLEMENT ARGUMENT
EXIT

Figure I-4

Change Routine

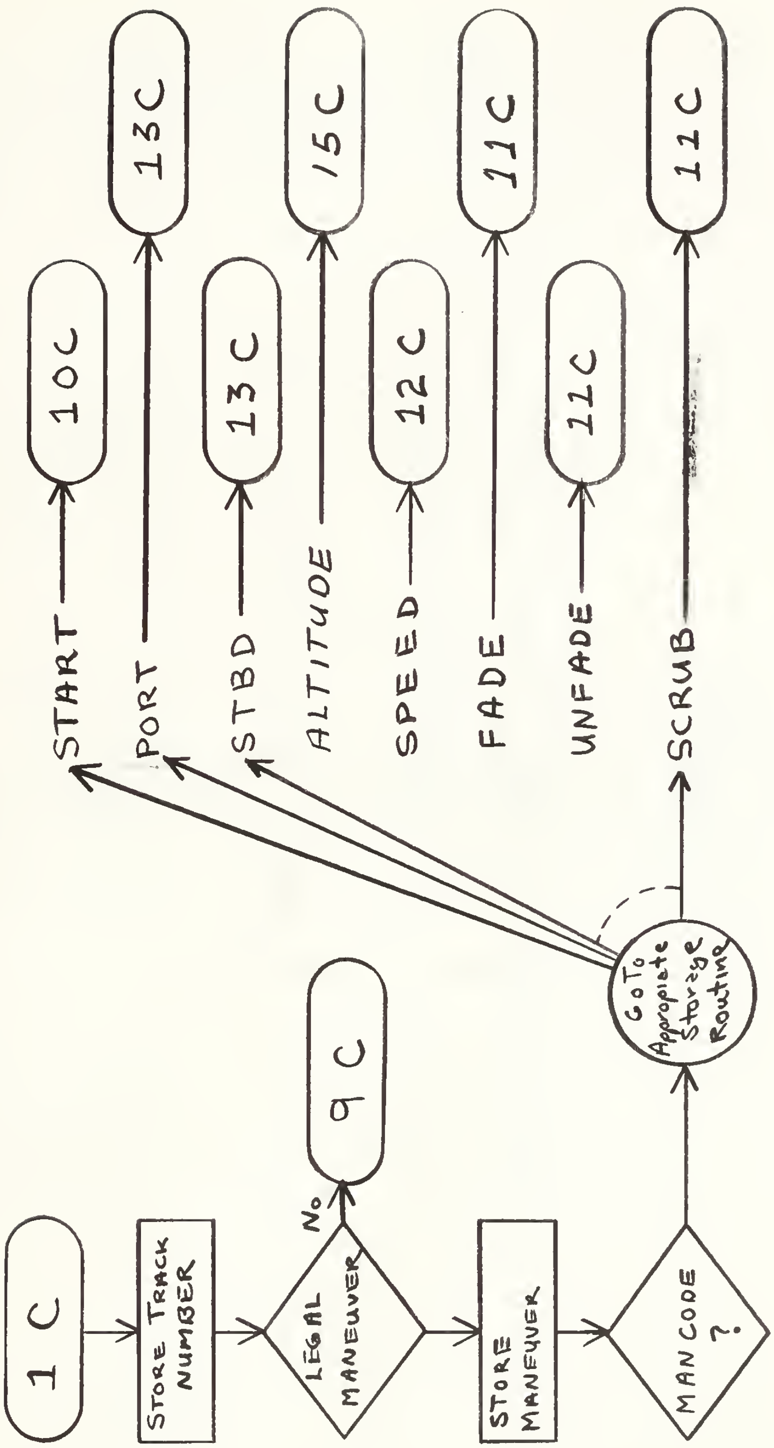


Figure I-5

Storage Routines

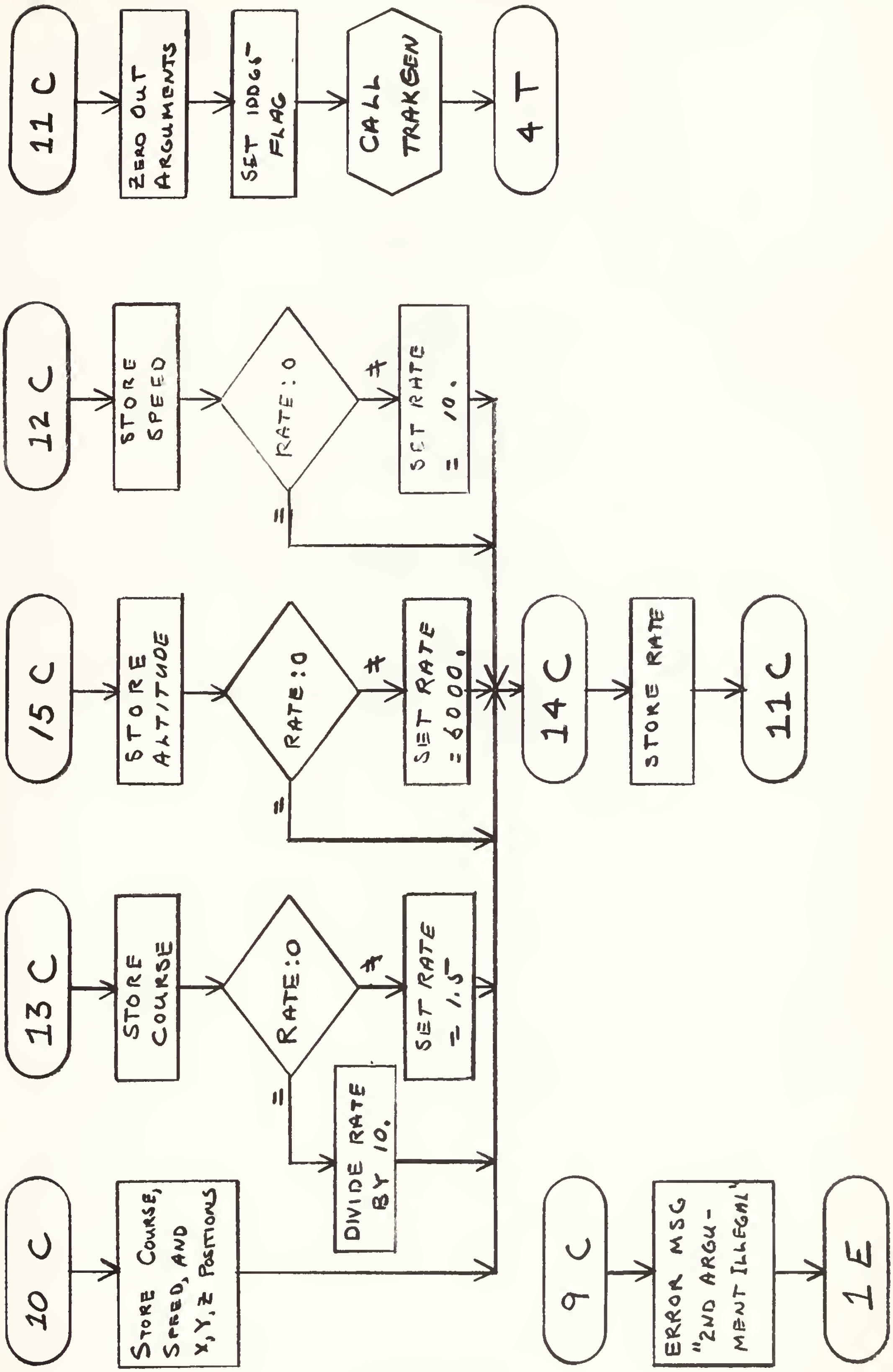


Figure I-6

Error Message Routine

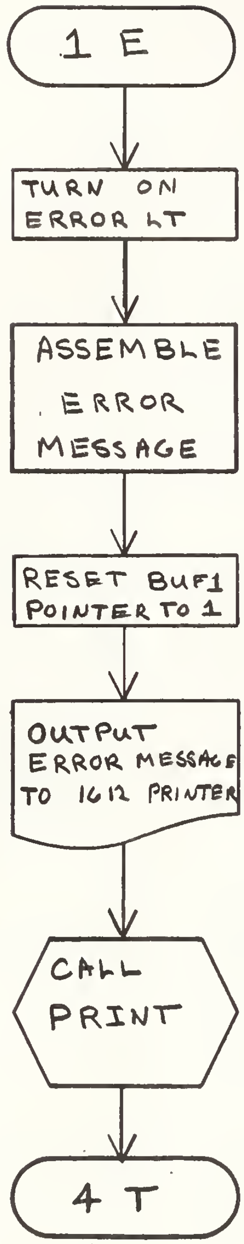


Figure I-7

Display Subprogram

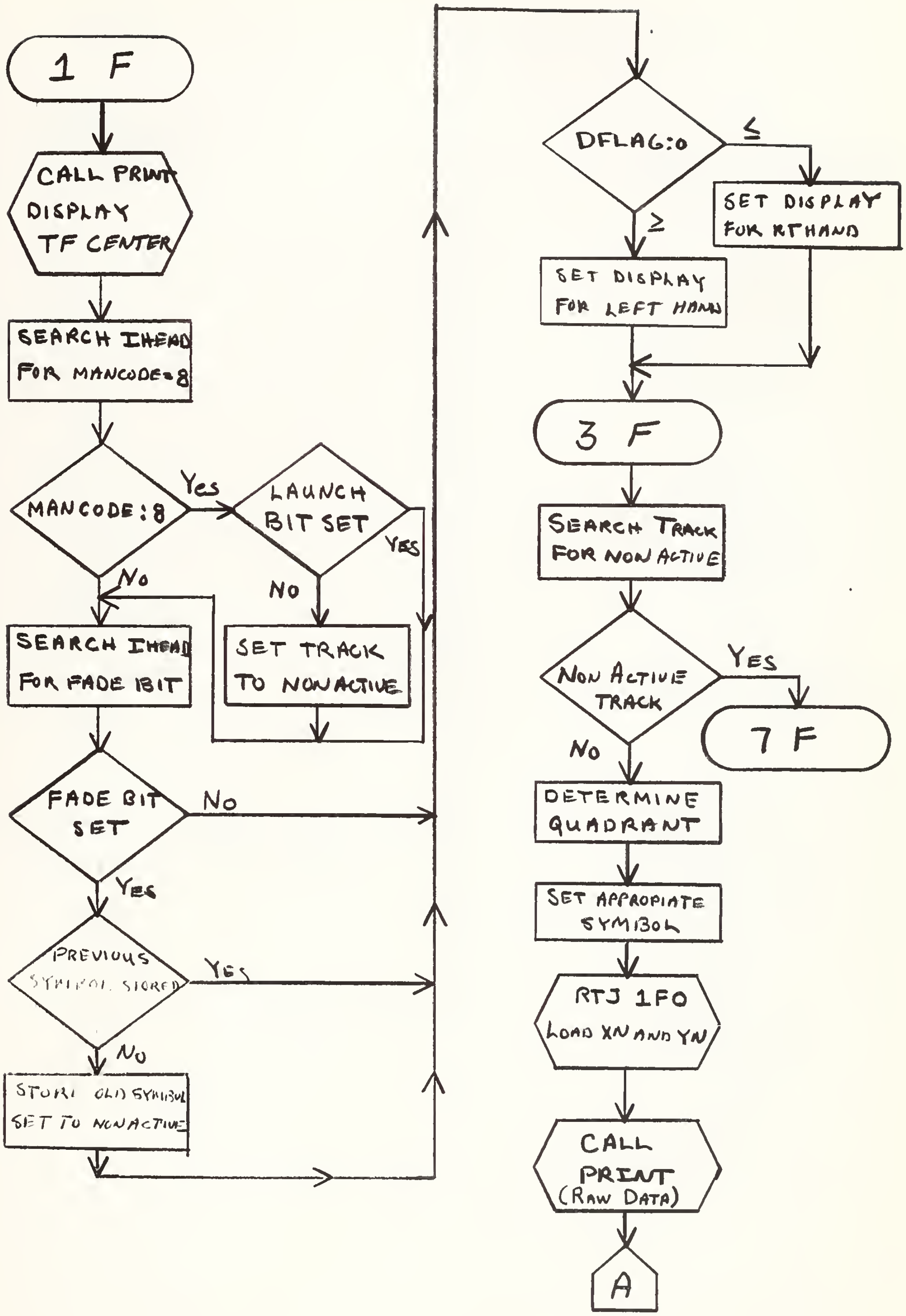


Figure I-8

Display Subprogram (cont)

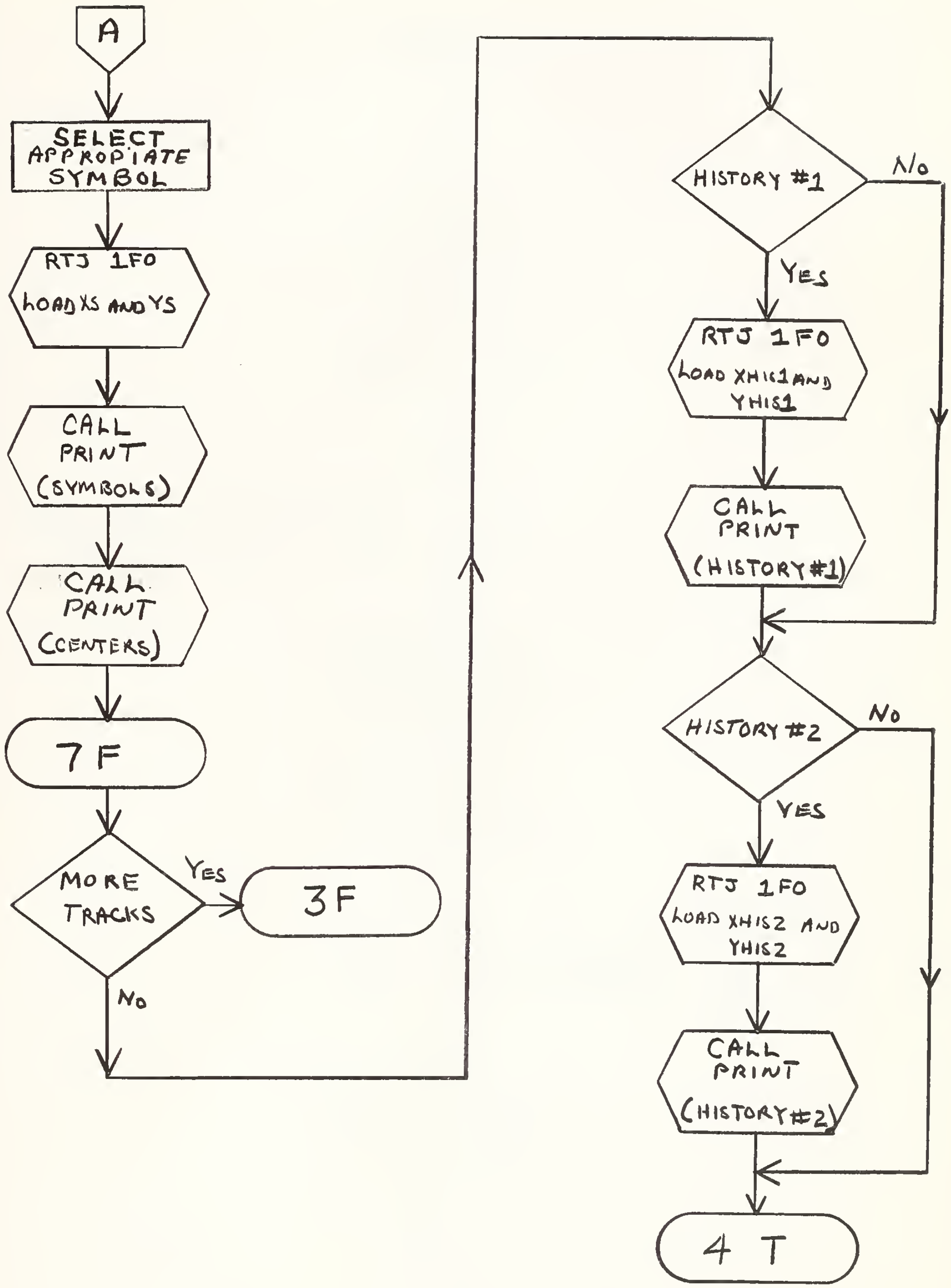


Figure I-9

Load X and Y Routine

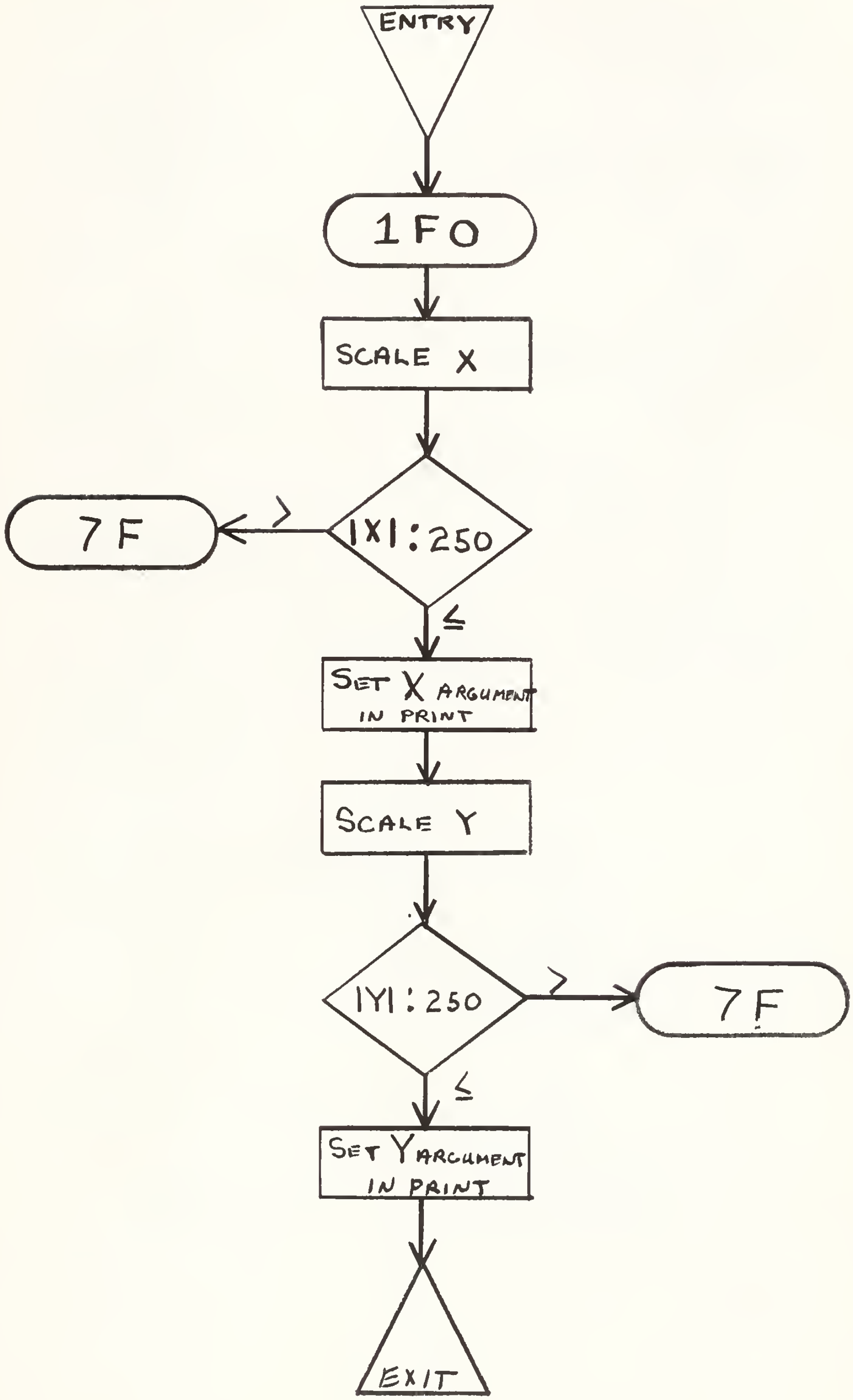


Figure I-10

Radar Processor Subprogram

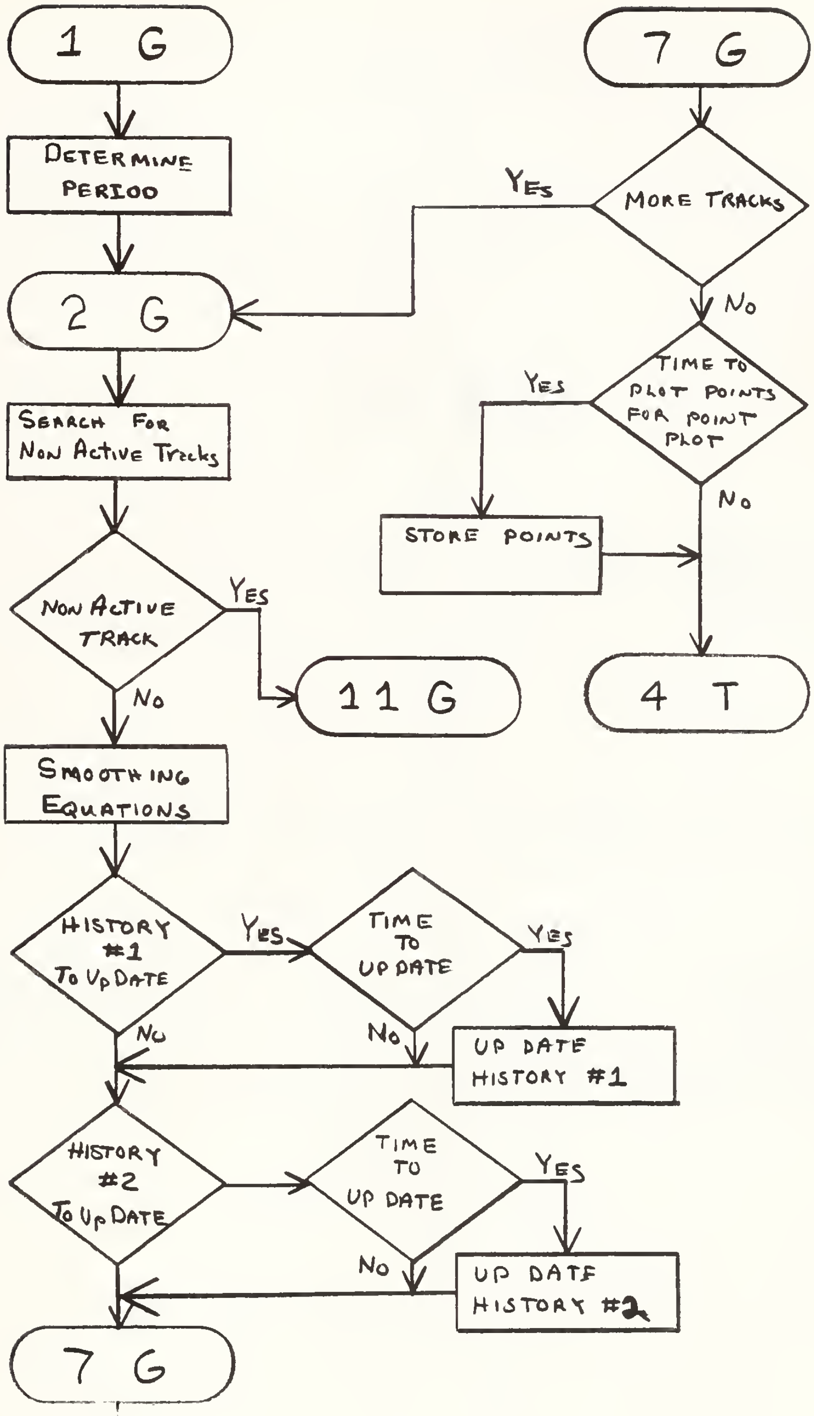


Figure I-11

Radar Processor Subprogram (cont)

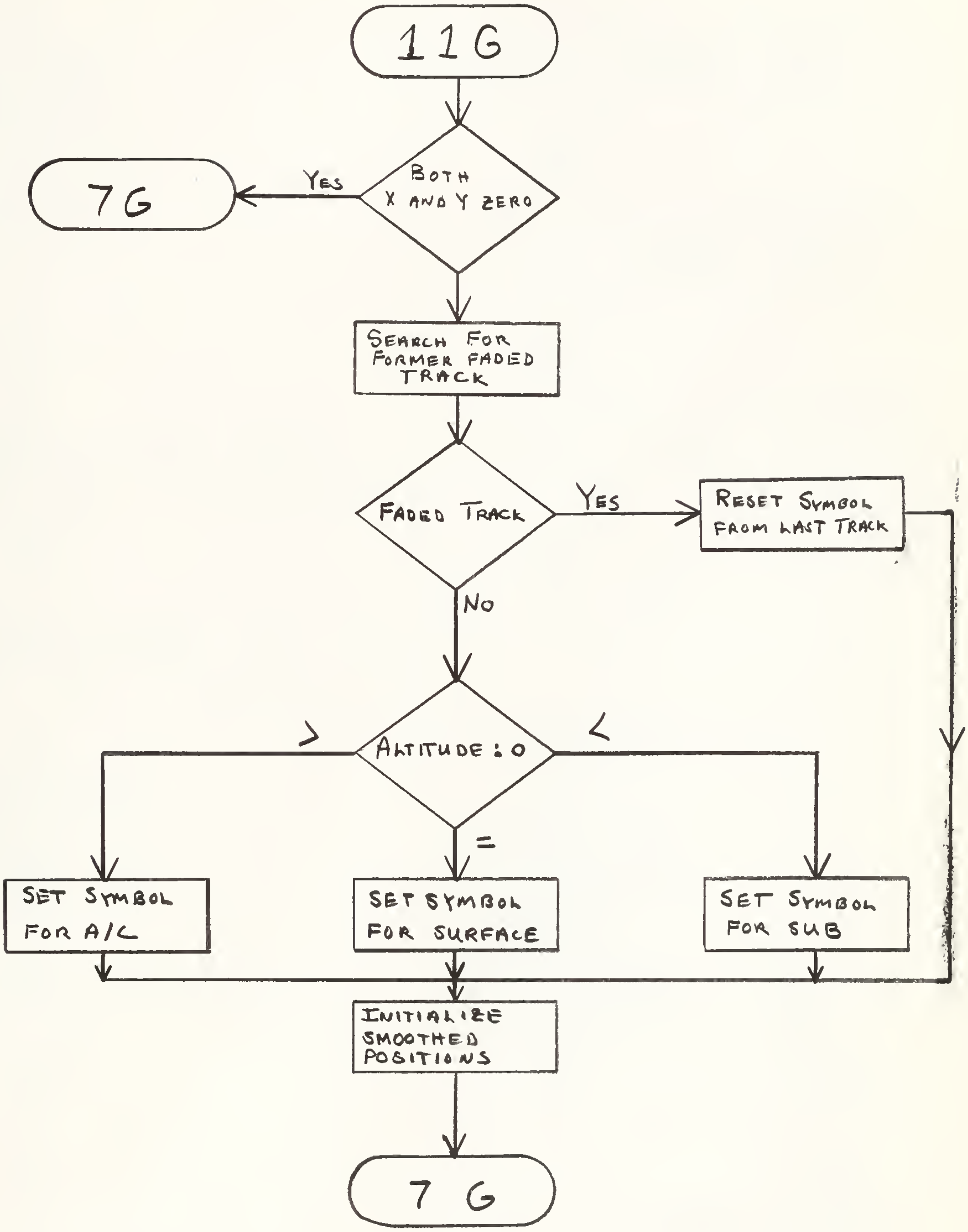


Figure I-12

Aircraft Control Subprogram

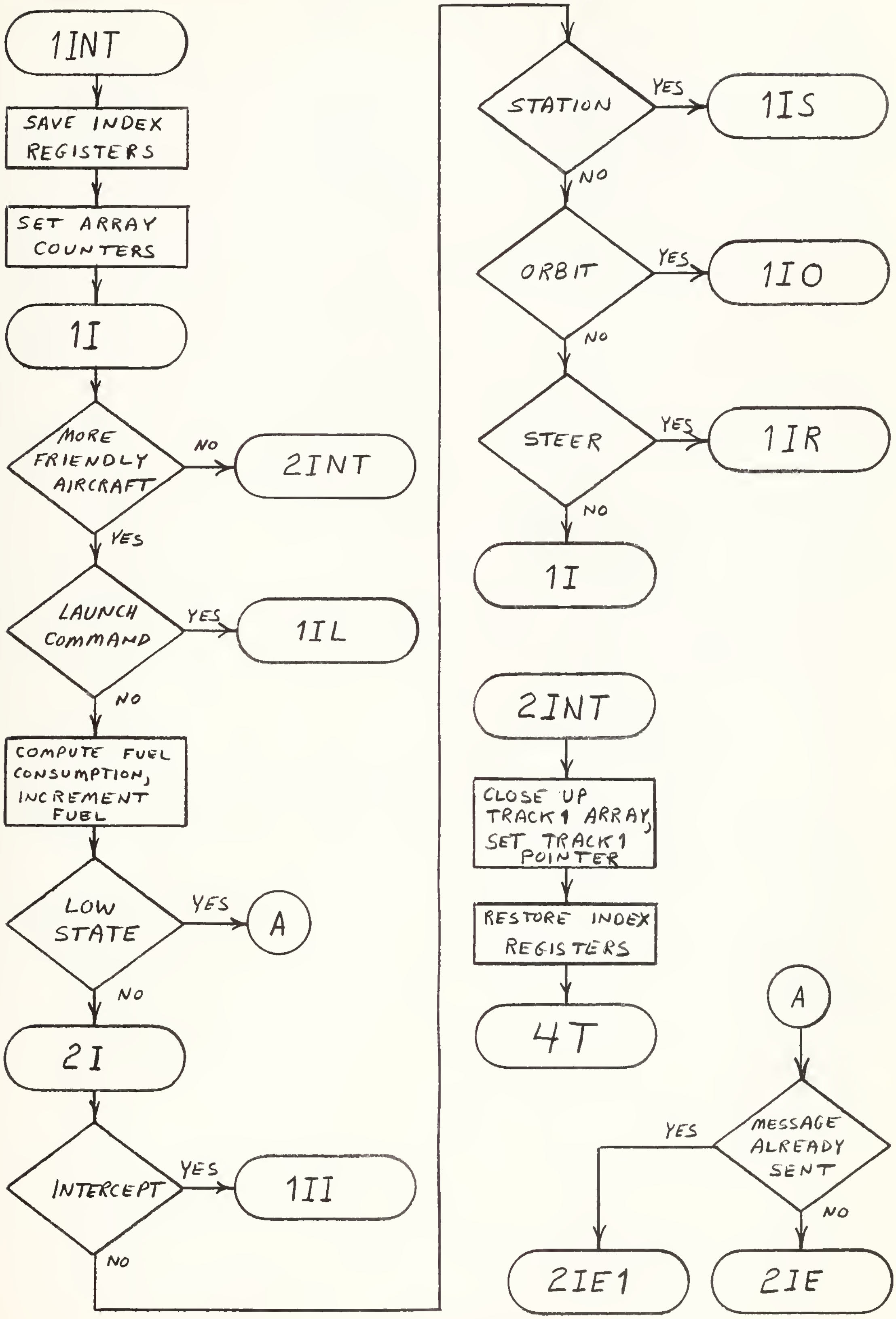


Figure I-13

Send Message Routine

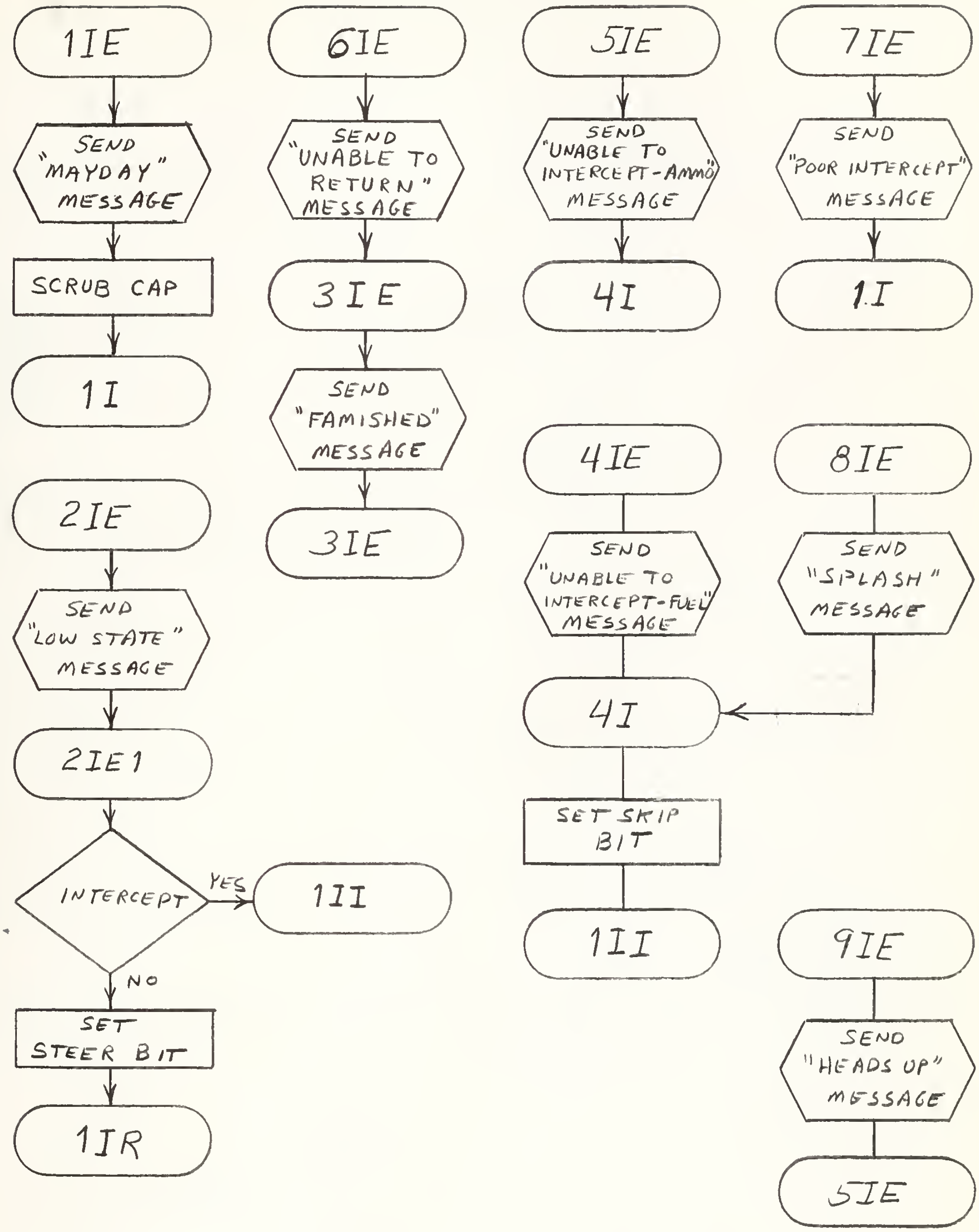


Figure I-14

Intercept and Skip Routines

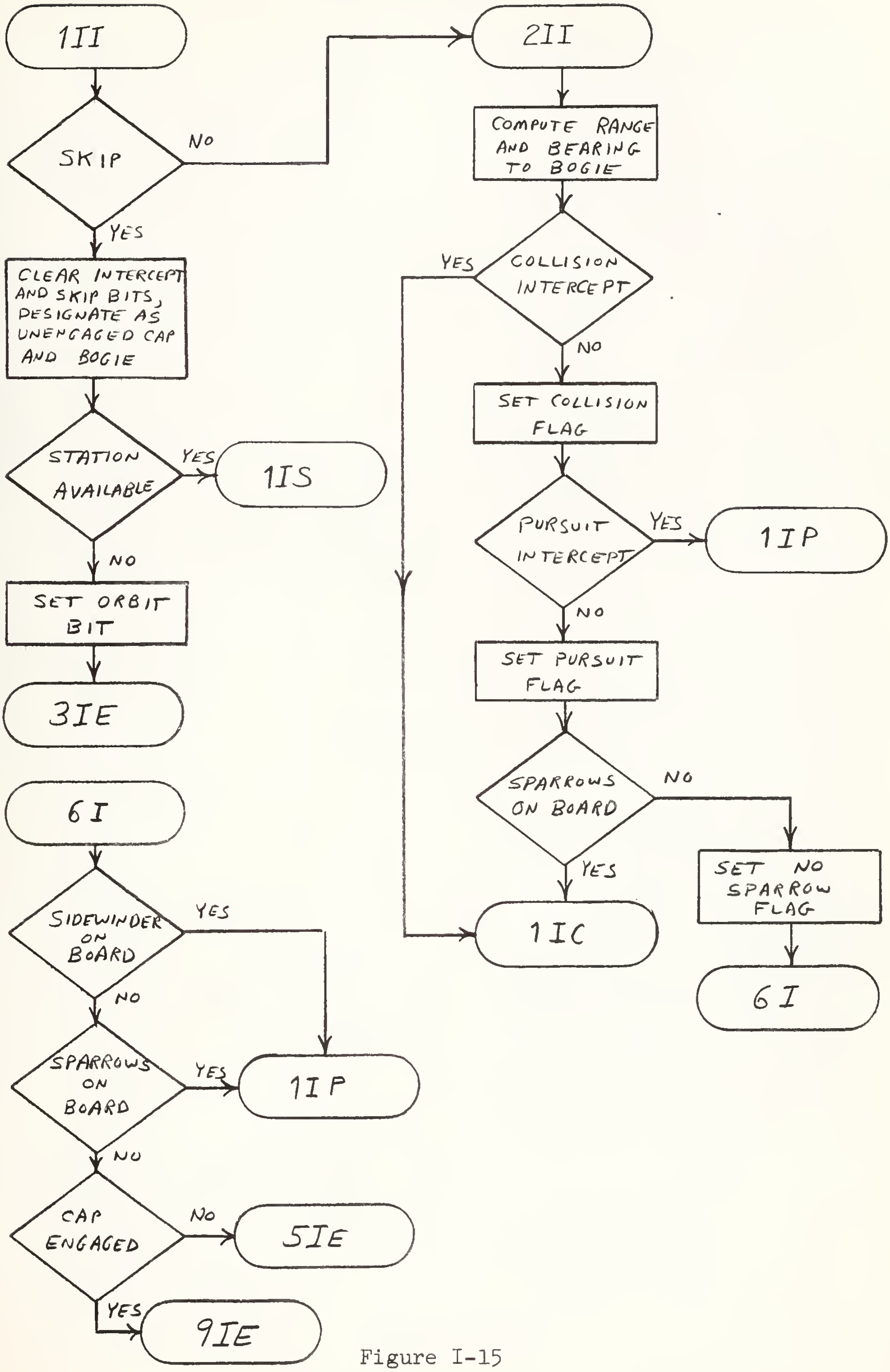


Figure I-15

Collision Intercept and Station Routines

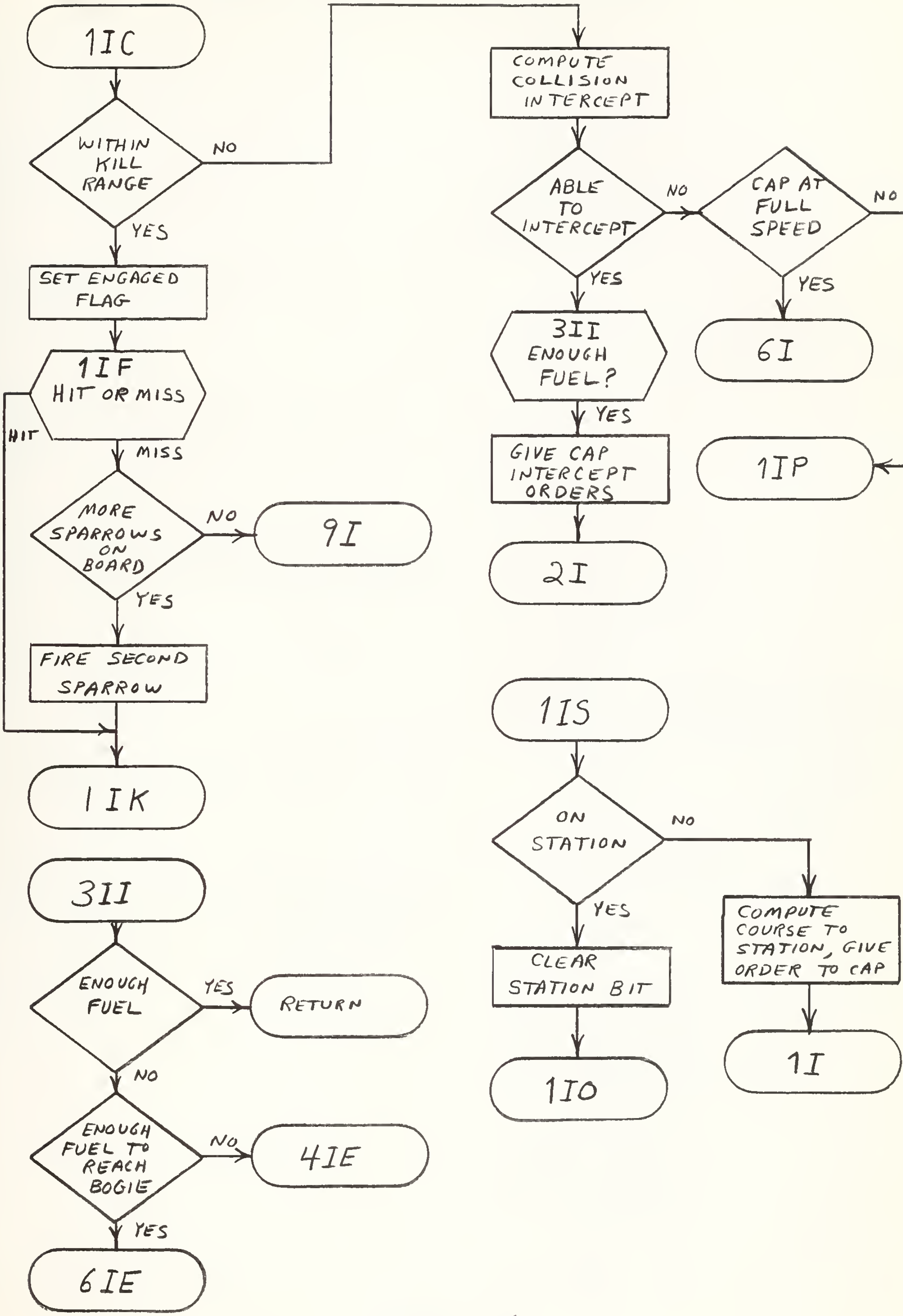


Figure I-16

Pursuit Intercept and Steer Routines

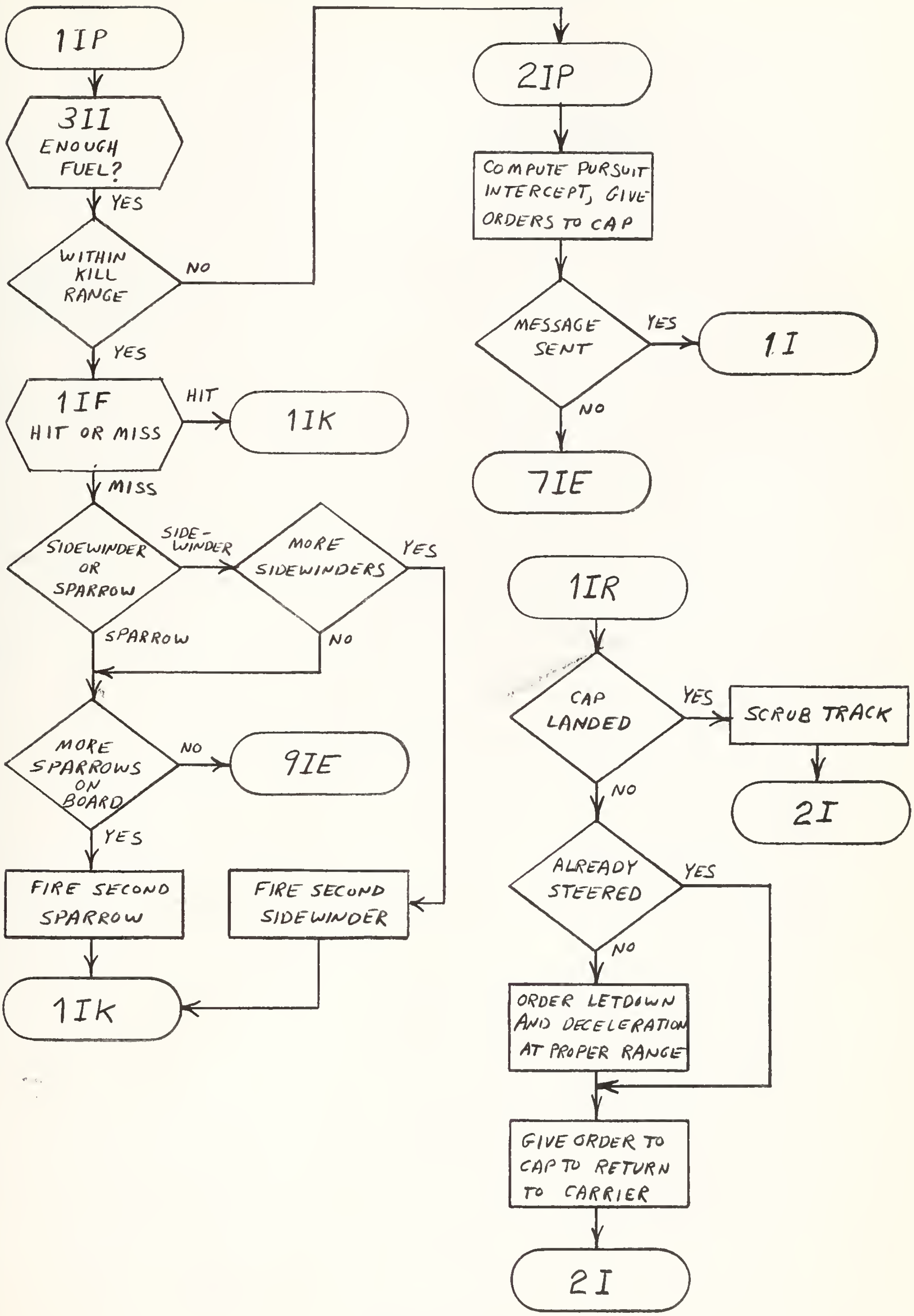


Figure I-17

Orbit Routine

Launch Routine

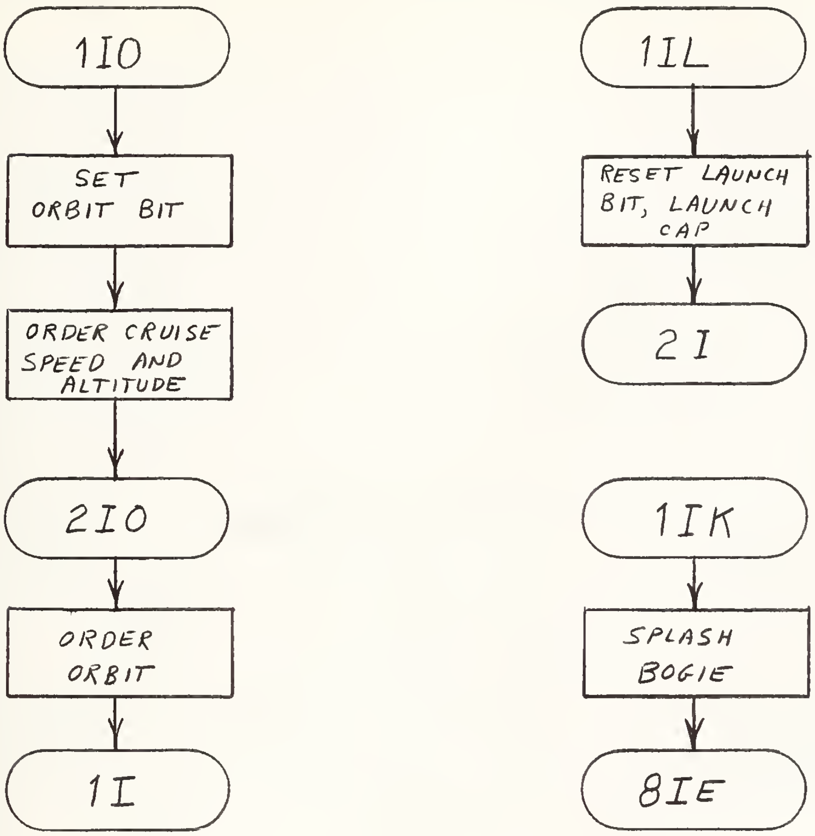


Figure I-18

Designate Track Routine

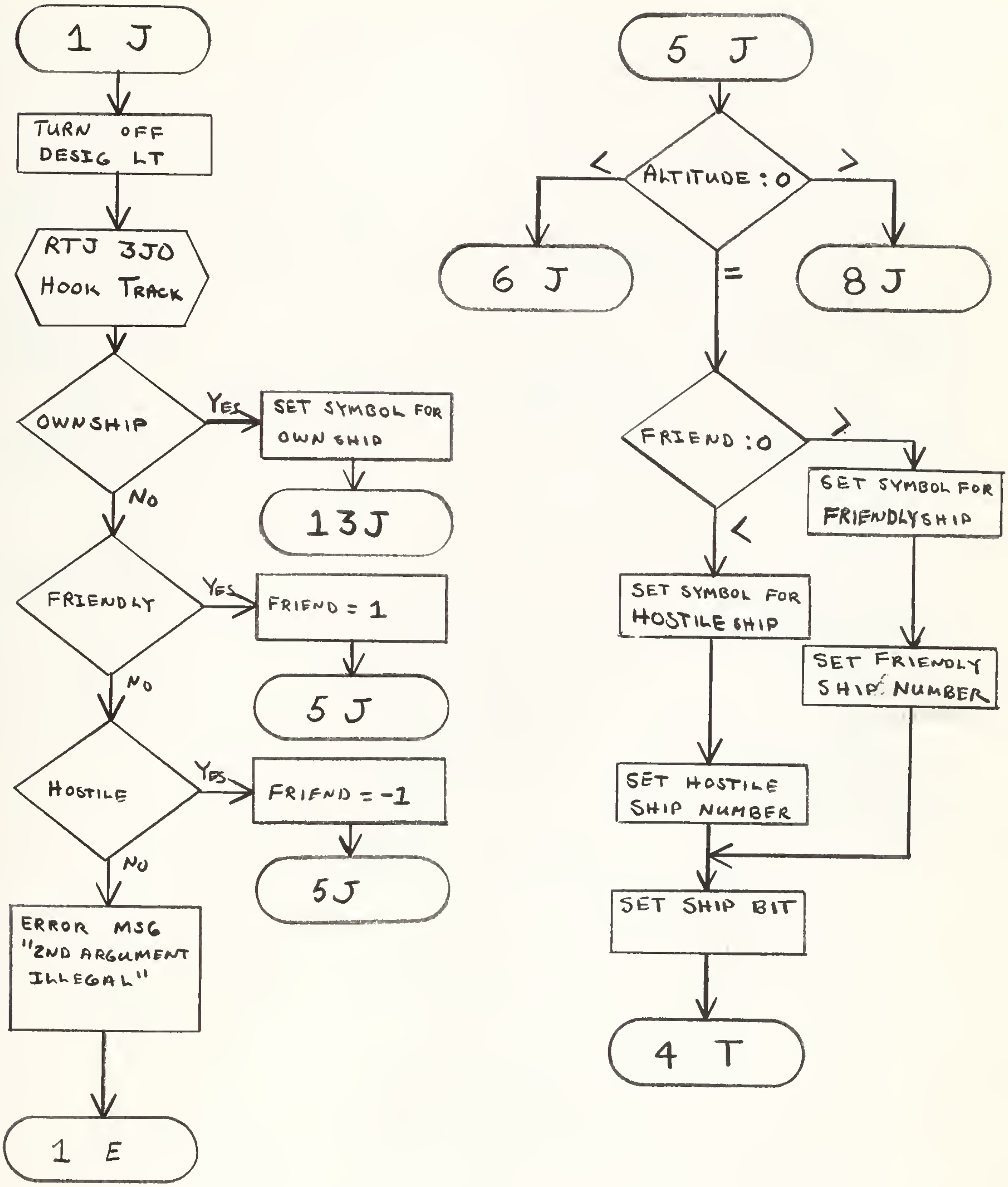


Figure I-19

Designate Track Routine (cont)

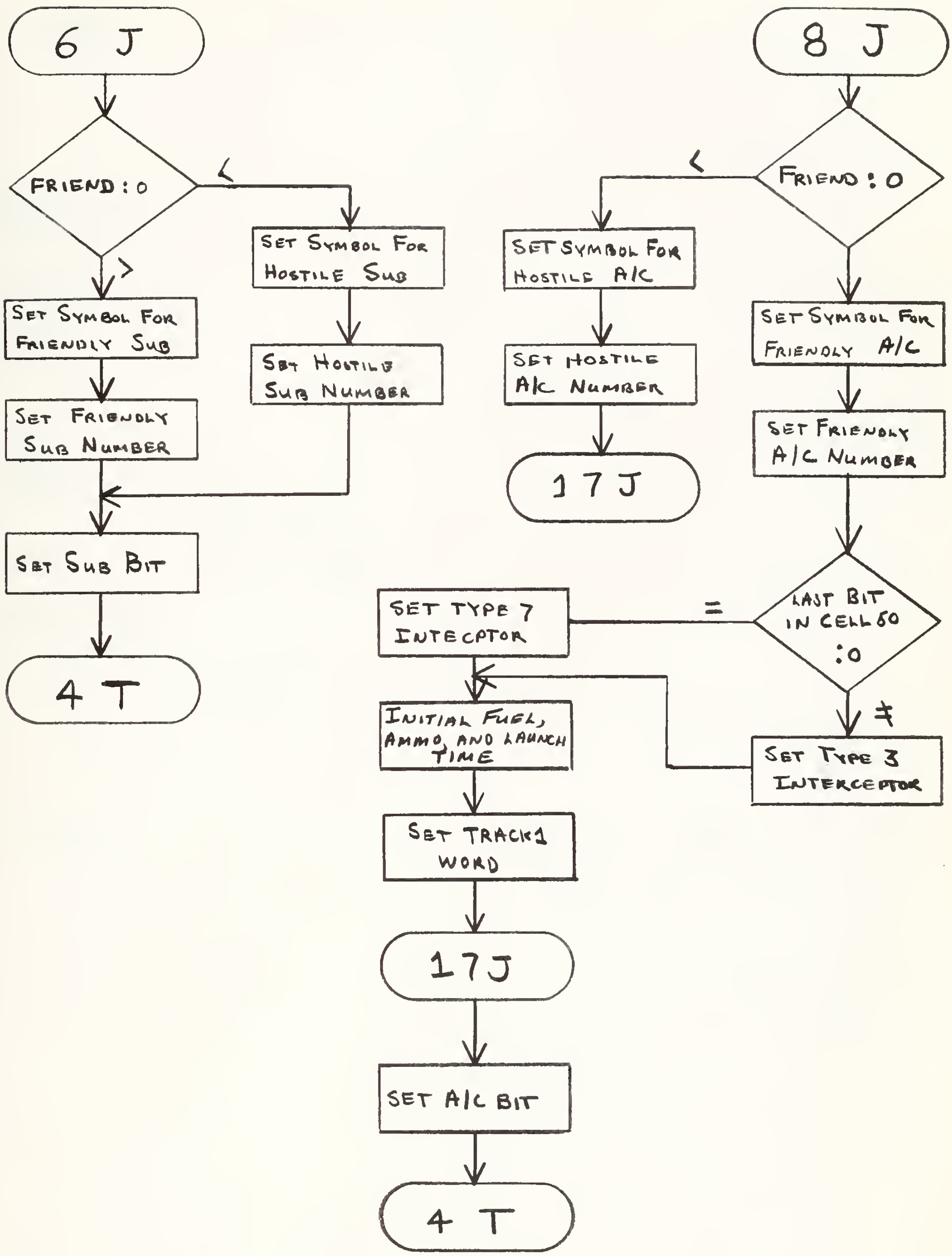
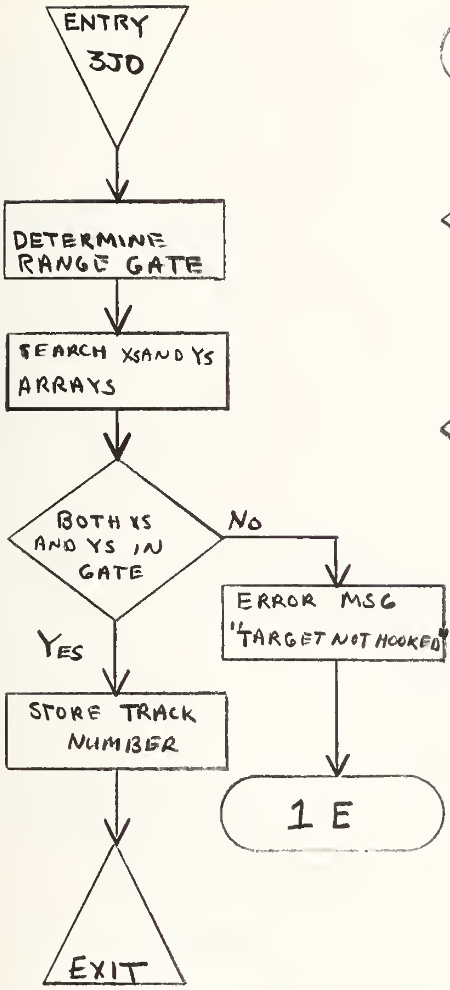


Figure I-20

Hook Track Routine



Keyboard Command Subprogram

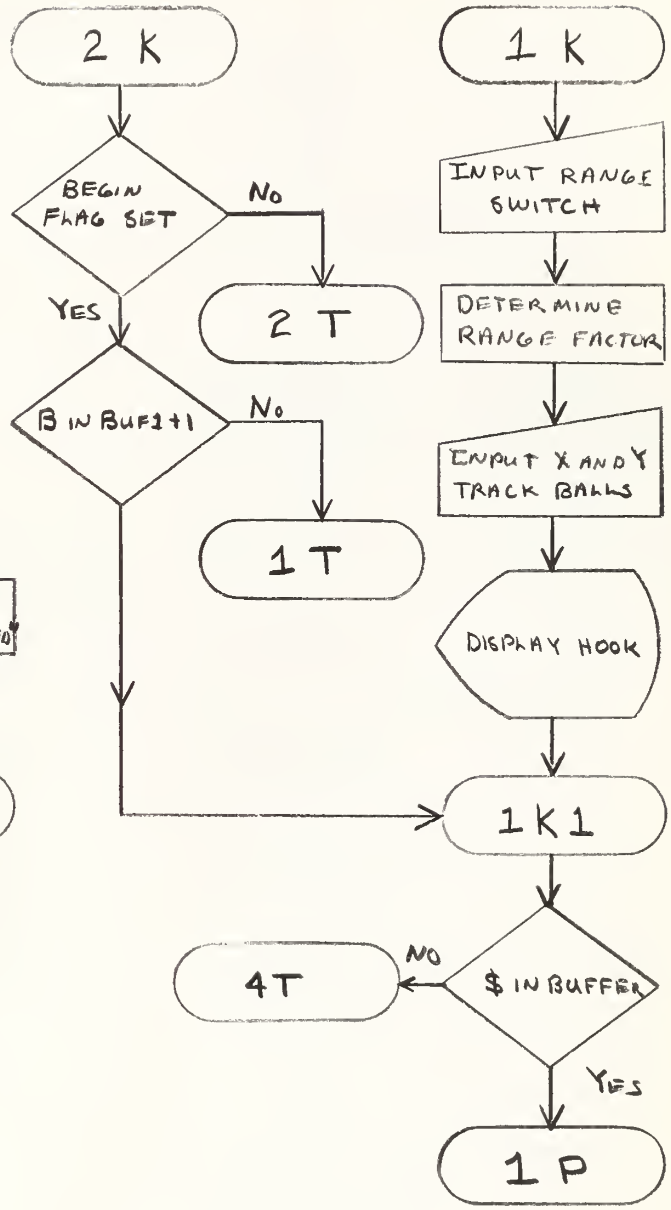


Figure I-21

History Routine

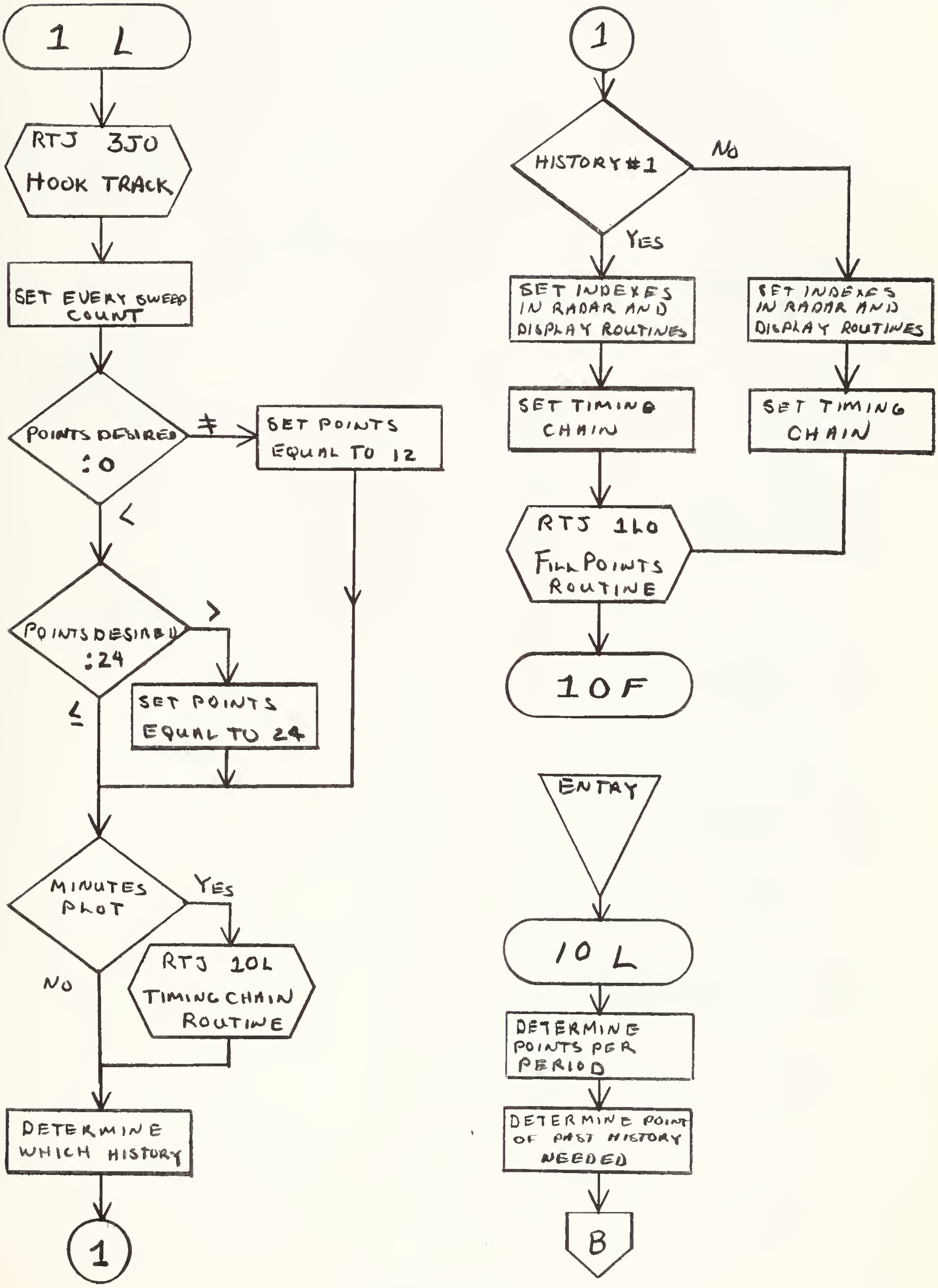


Figure I-22

History Routine (cont)

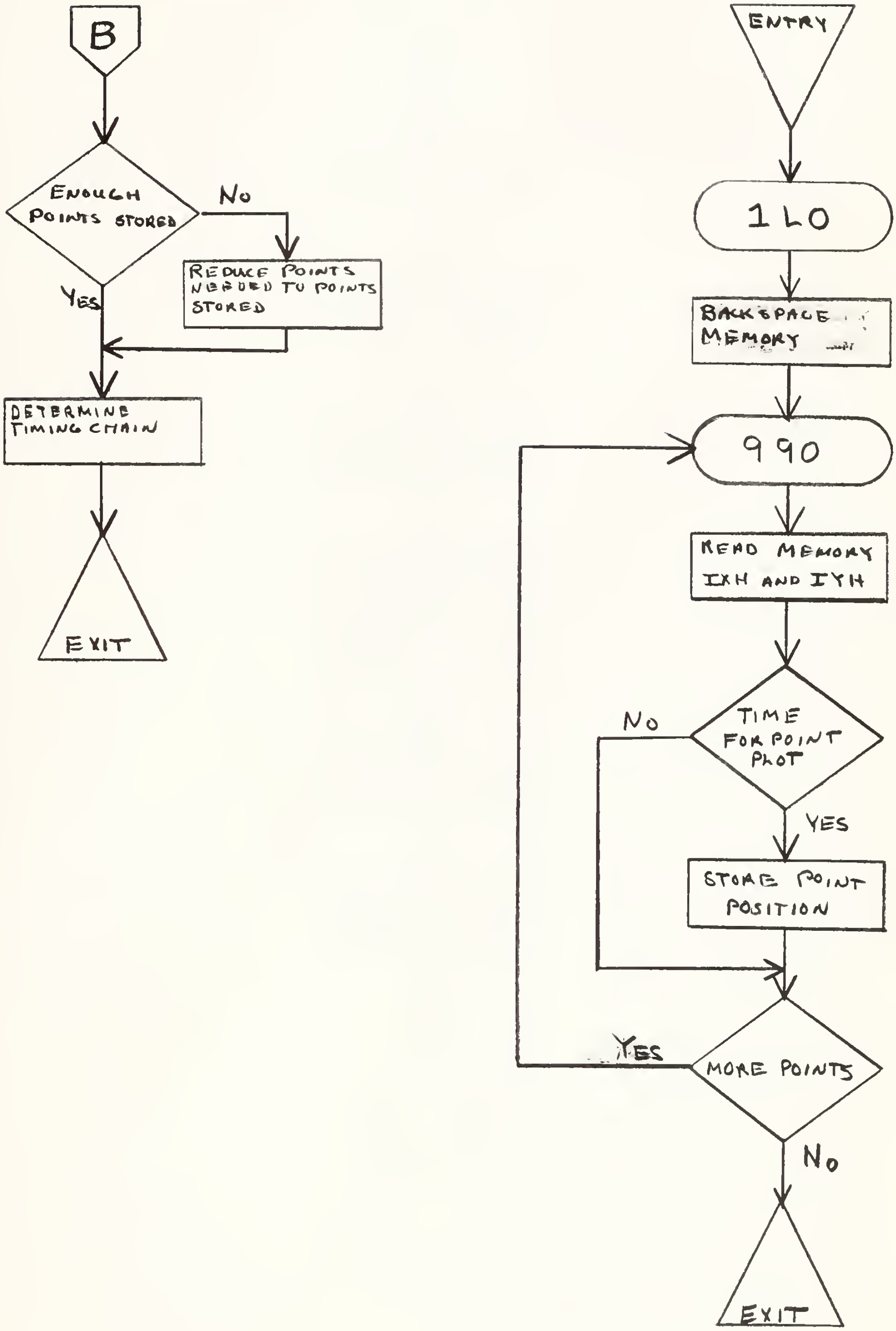


Figure I-23

Erase Routine

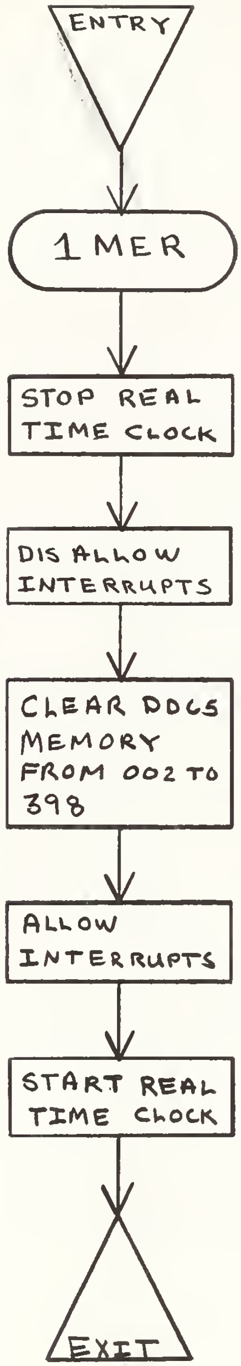


Figure I-24

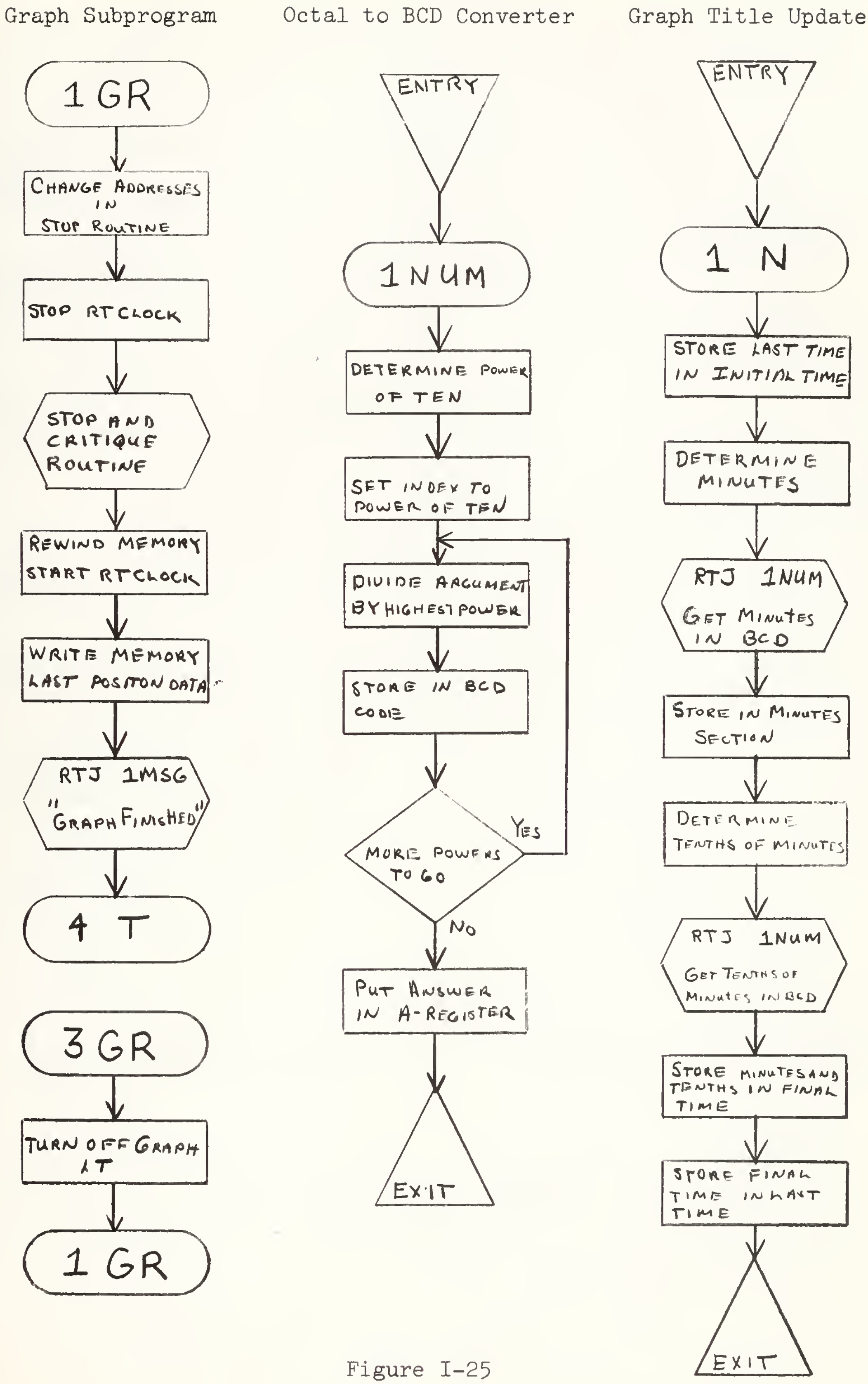
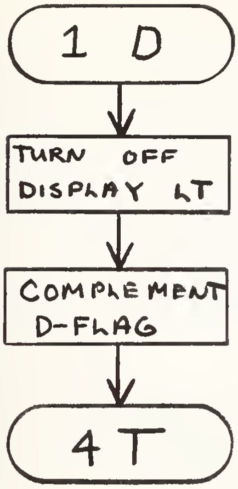
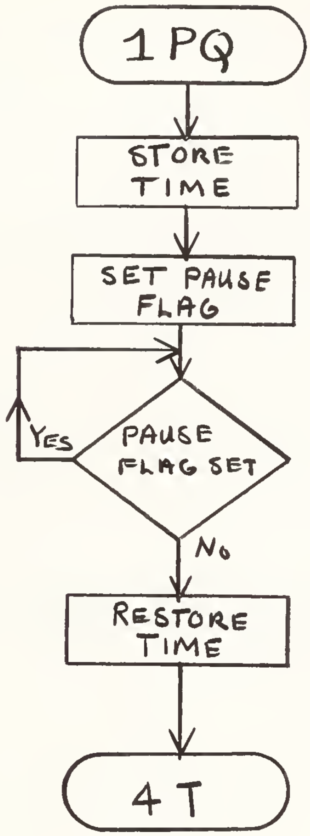

Graph Subprogram
1 GR
CHANGE ADDRESSES IN STOP ROUTINE
STOP RT CLOCK
STOP AND CRITIQUE ROUTINE
REWIND MEMORY START RT CLOCK
WRITE MEMORY LAST POSITON DATA
RTJ 1MSG "GRAPH FINISHED"
4 T
3 GR
TURN OFF GRAPH 1 T
1 GR
Octal to BCD Converter
ENTRY
1 NUM
DETERMINE POWER OF TEN
SET INDEX TO POWER OF TEN
DIVIDE ARGUMENT BY HIGHEST POWER
STORE IN BCD CODE
MORE POWERS TO GO
YES
NO
PUT ANSWER IN A-REGISTER
EXIT
Graph Title Update
ENTRY
1 N
STORE LAST TIME IN INITIAL TIME
DETERMINE MINUTES
RTJ 1NUM GET MINUTES IN BCD
STORE IN MINUTES SECTION
DETERMINE TENTHS OF MINUTES
RTJ 1NUM GET TENTHS OF MINUTES IN BCD
STORE MINUTES AND TENTHS IN FINAL TIME
STORE FINAL TIME IN LAST TIME
EXIT

Figure I-25

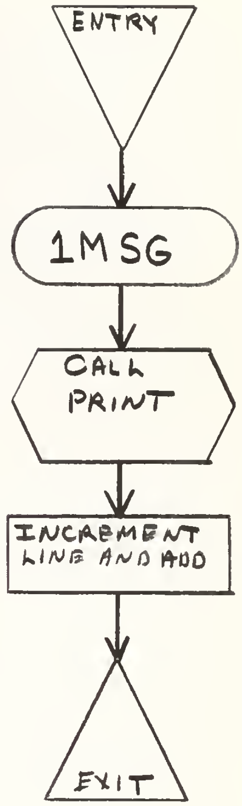
Display Routine



Hold or Pause Routine



Message Routine



Drop History Routine

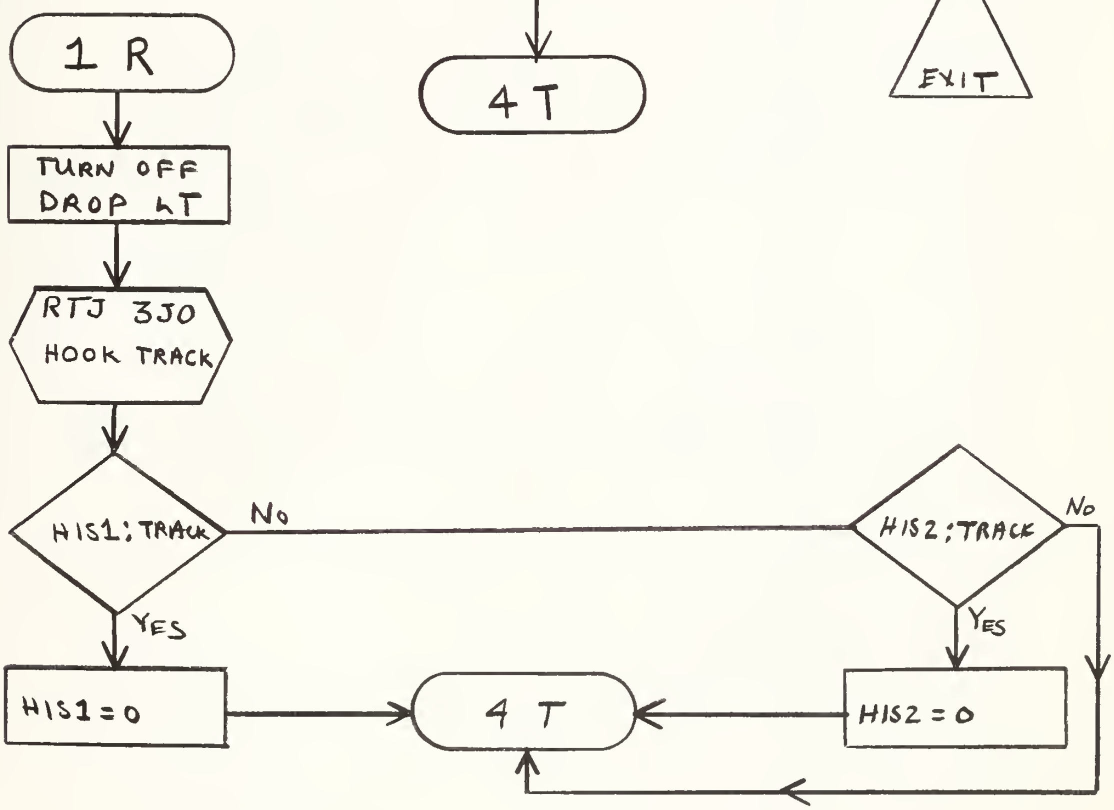


Figure I-26

Program Control Routine

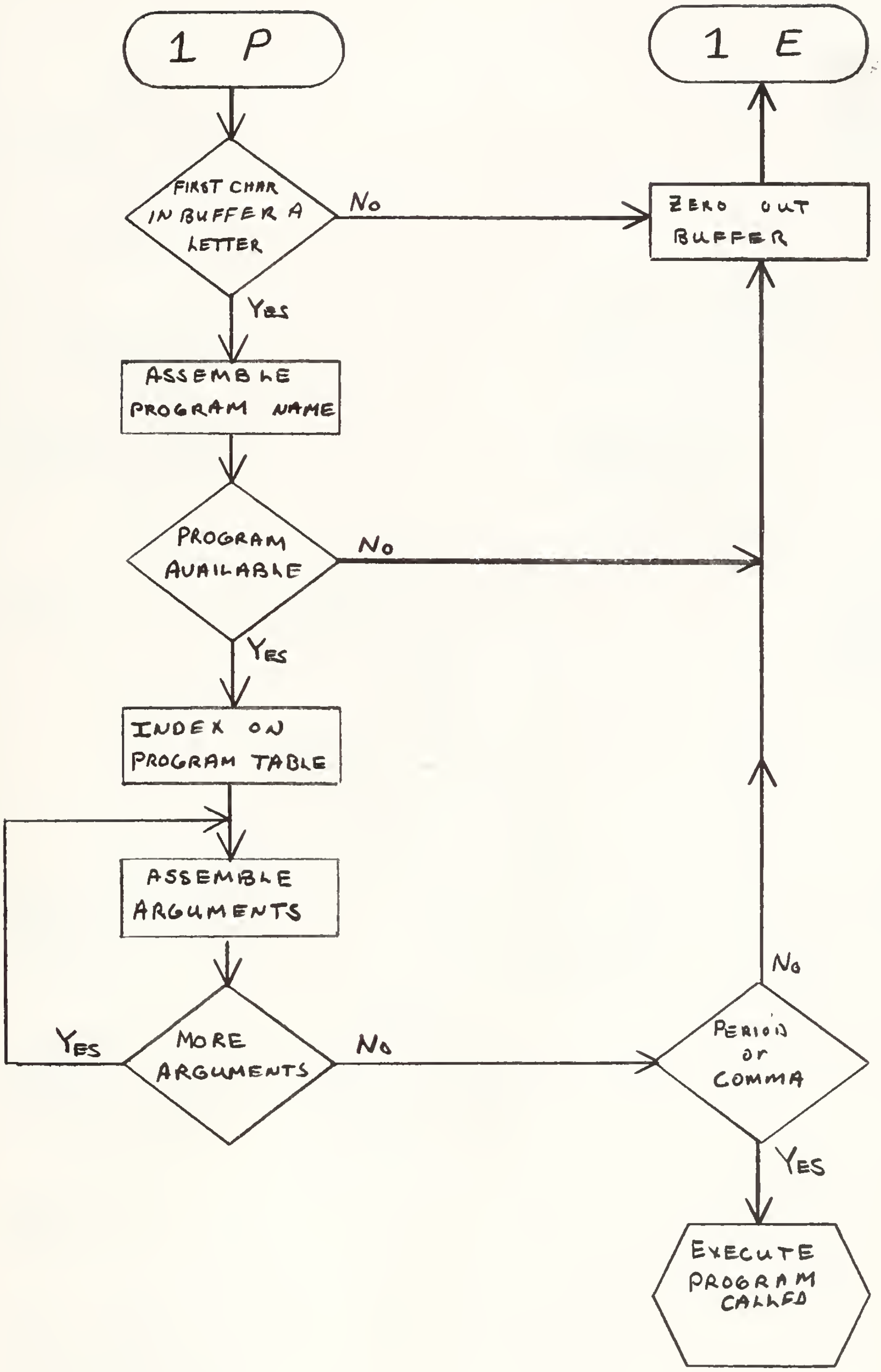


Figure I-27

Intercept Command Routine

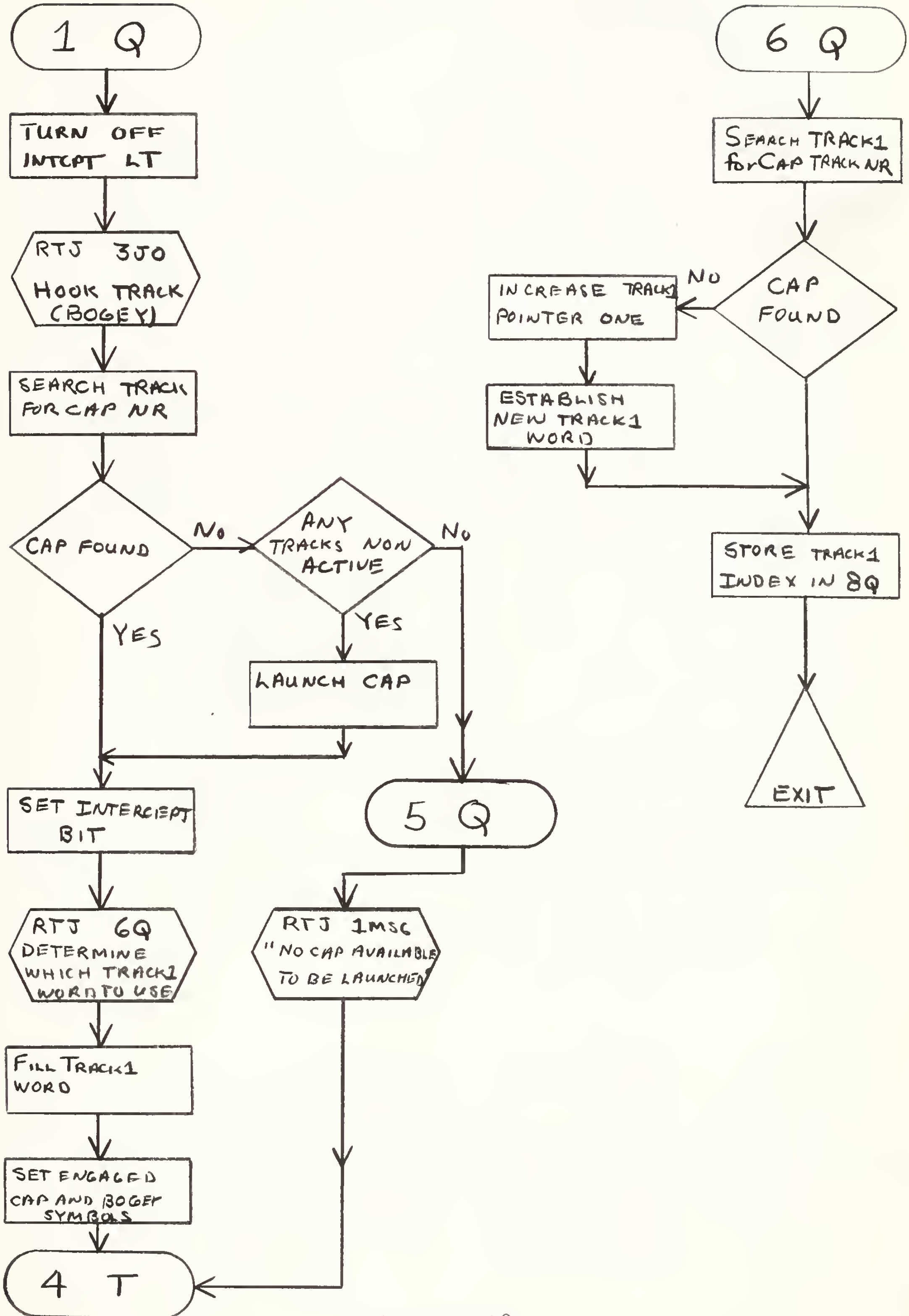


Figure I-28

Stop and Critique Routine

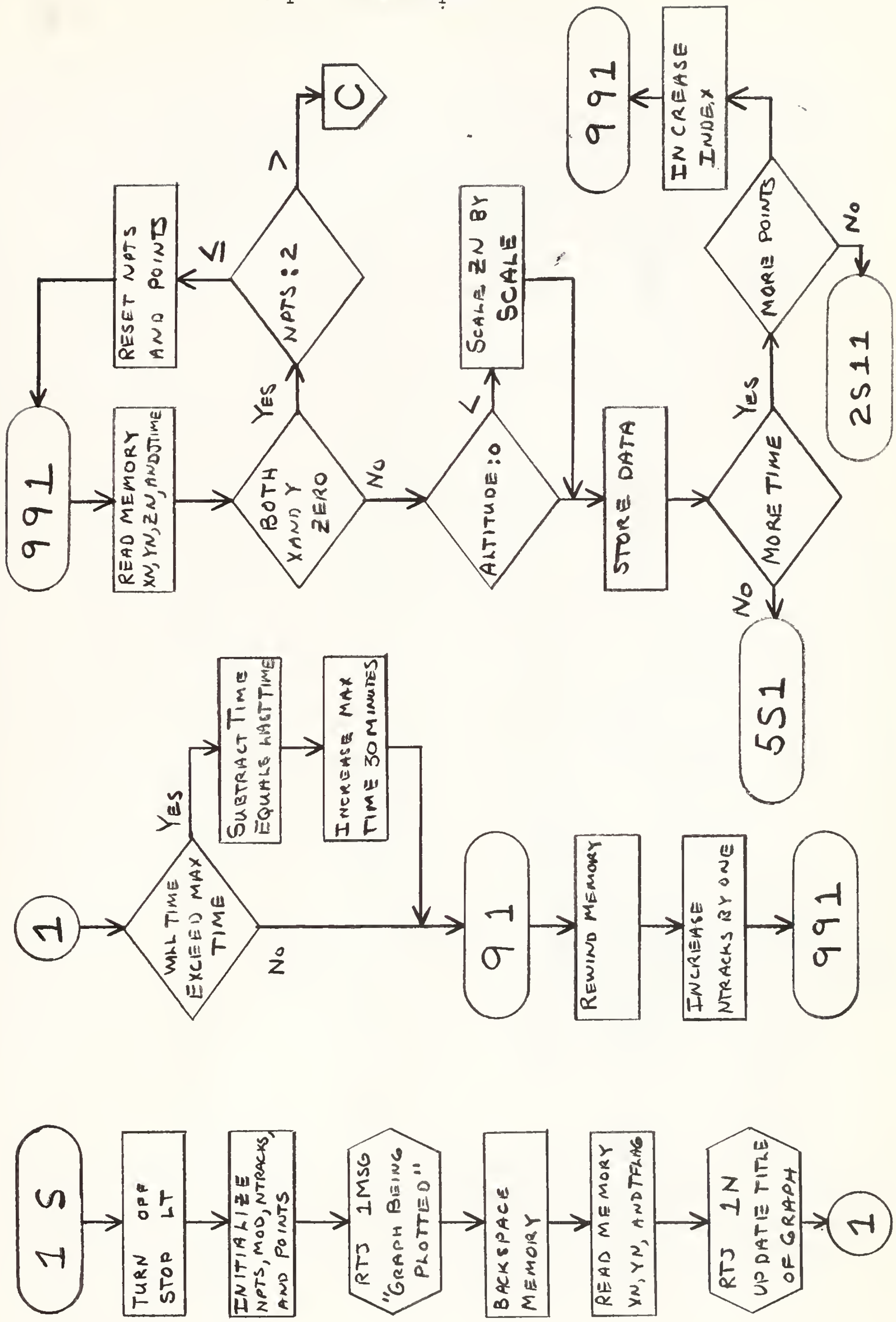


Figure I-29

Stop and Critique Routine (cont)

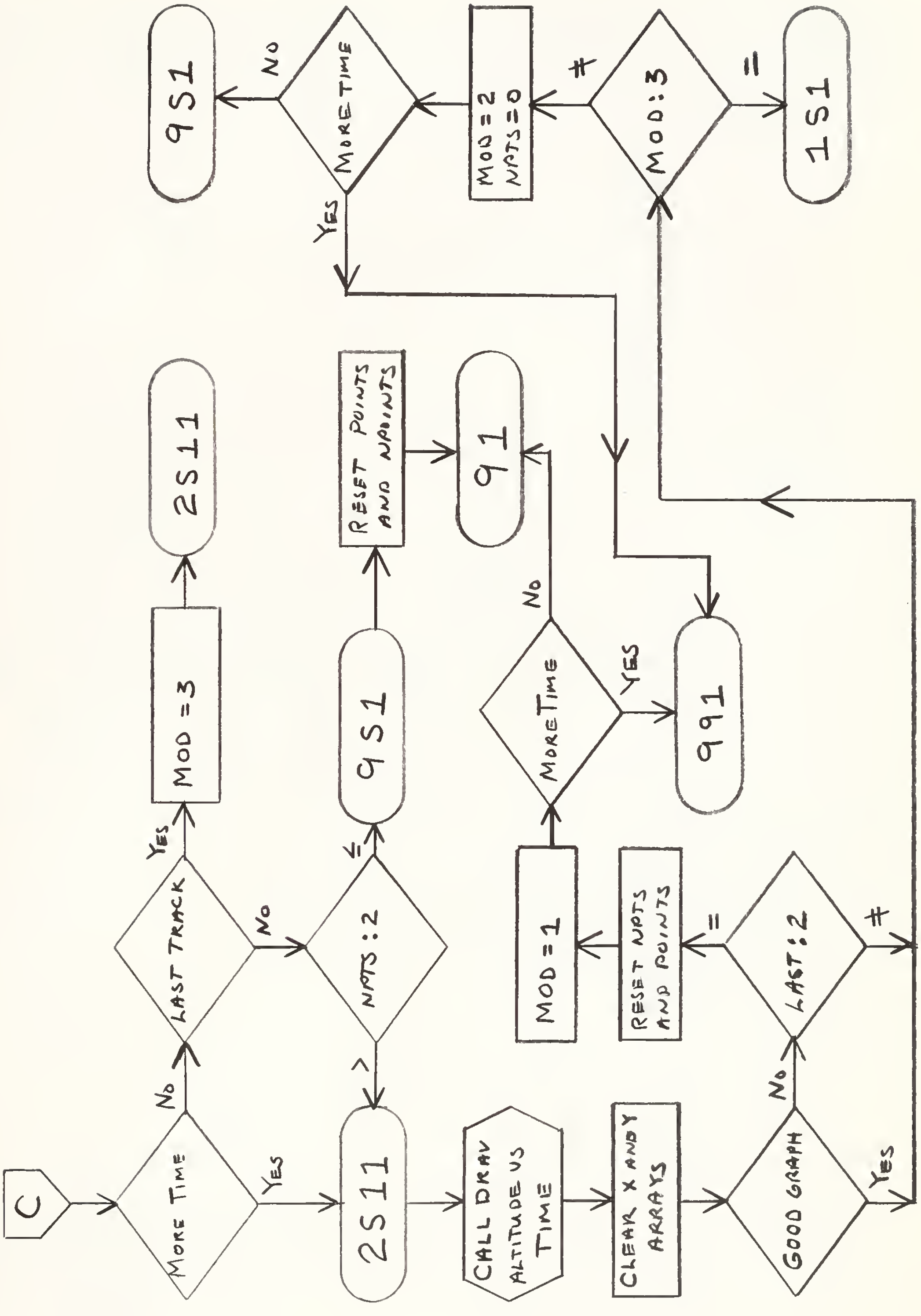


Figure I-30

Stop and Critique Routine (cont)

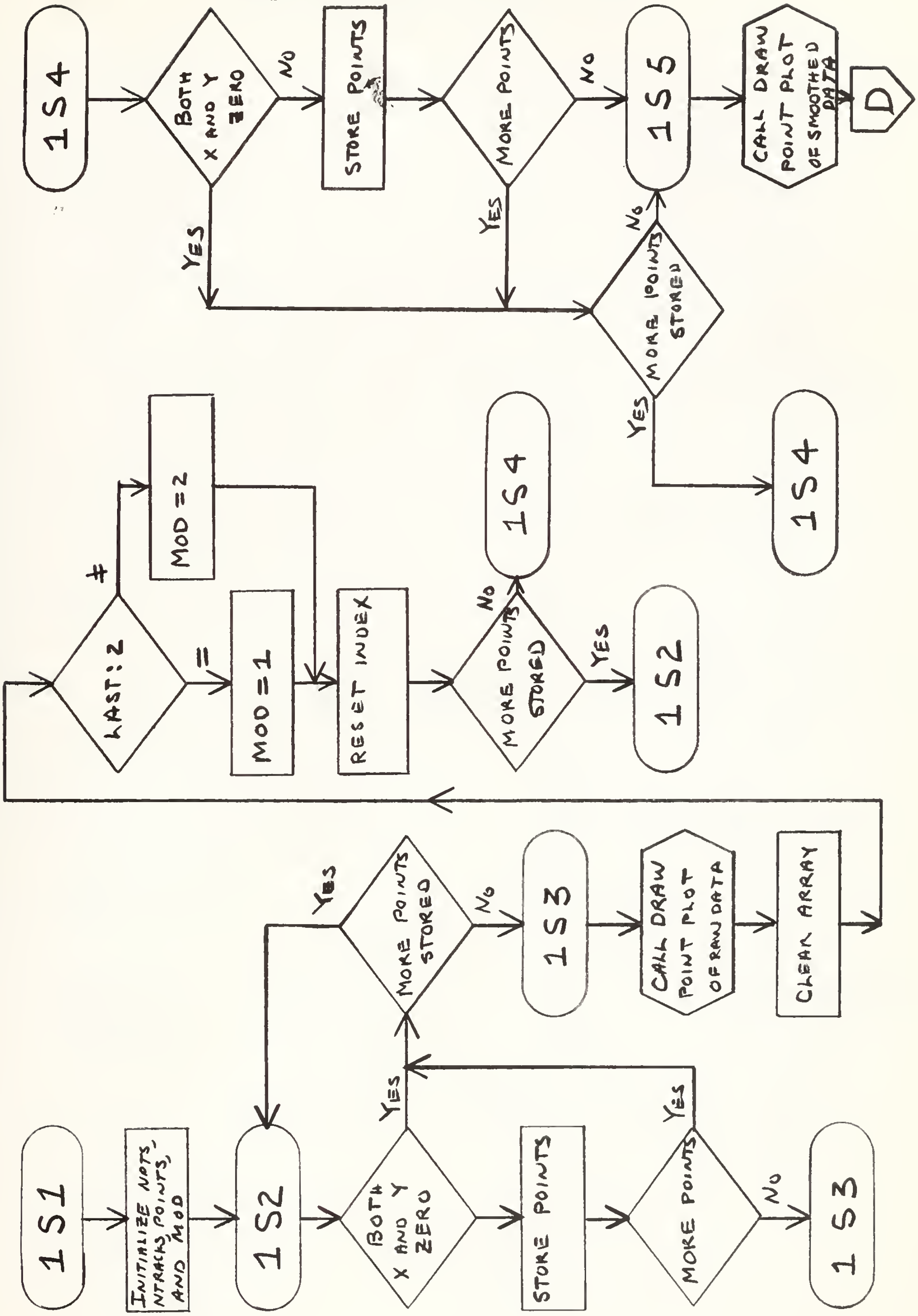


Figure I-31

Stop and Critique Routine (cont)

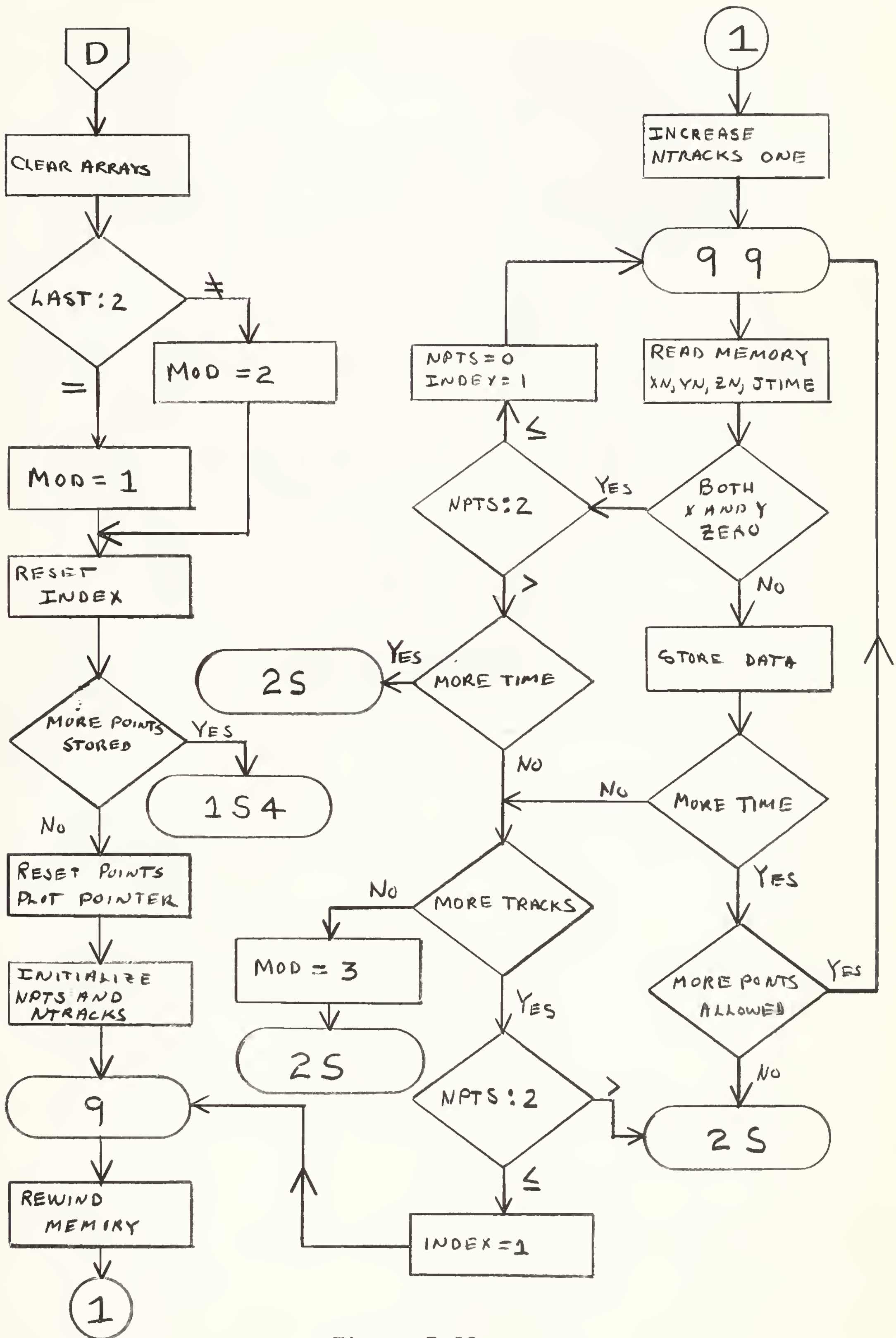


Figure I-32

Stop and Critique Routine (cont)

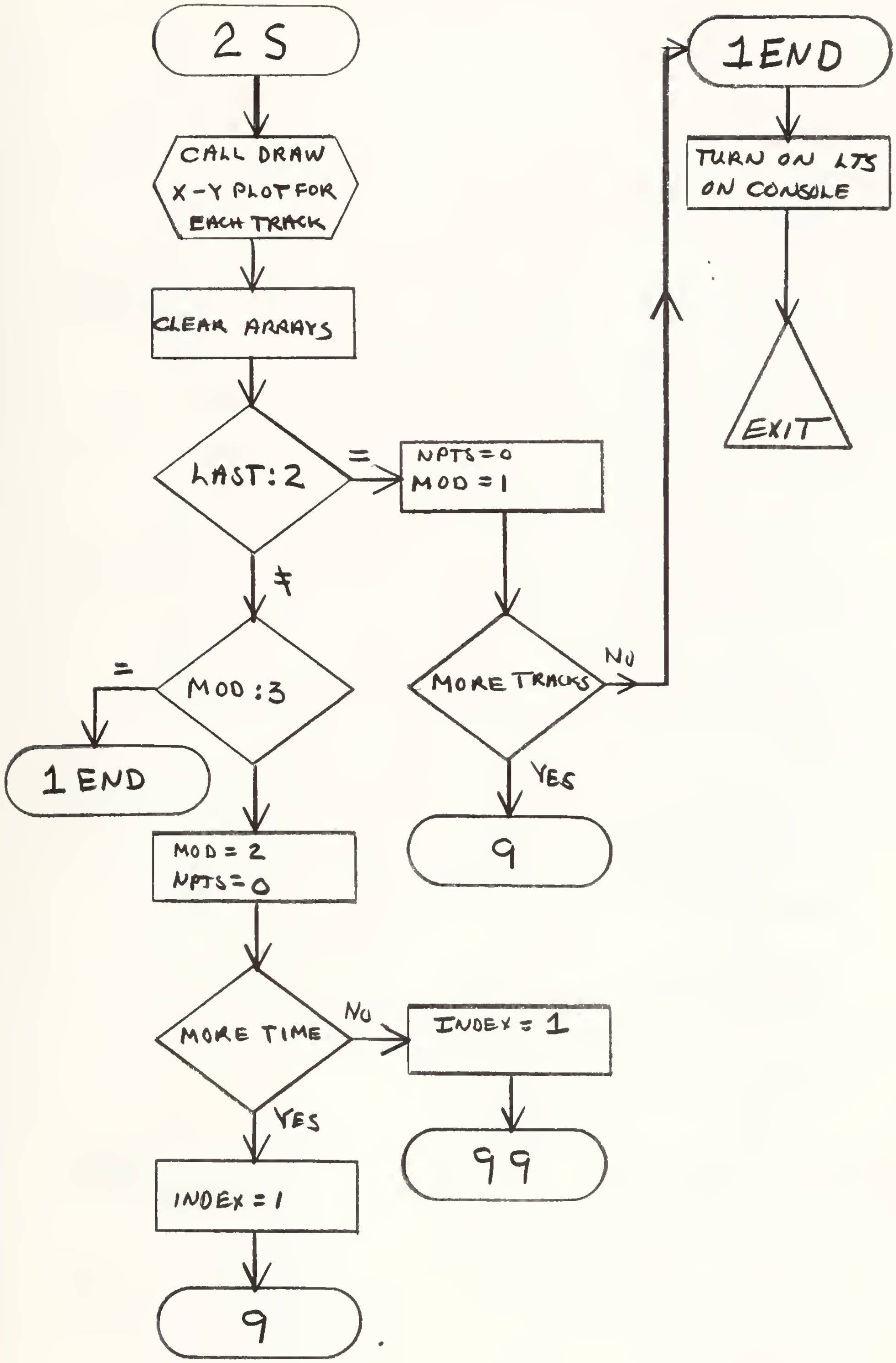
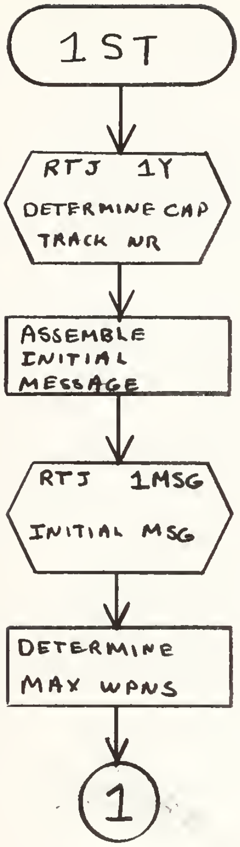
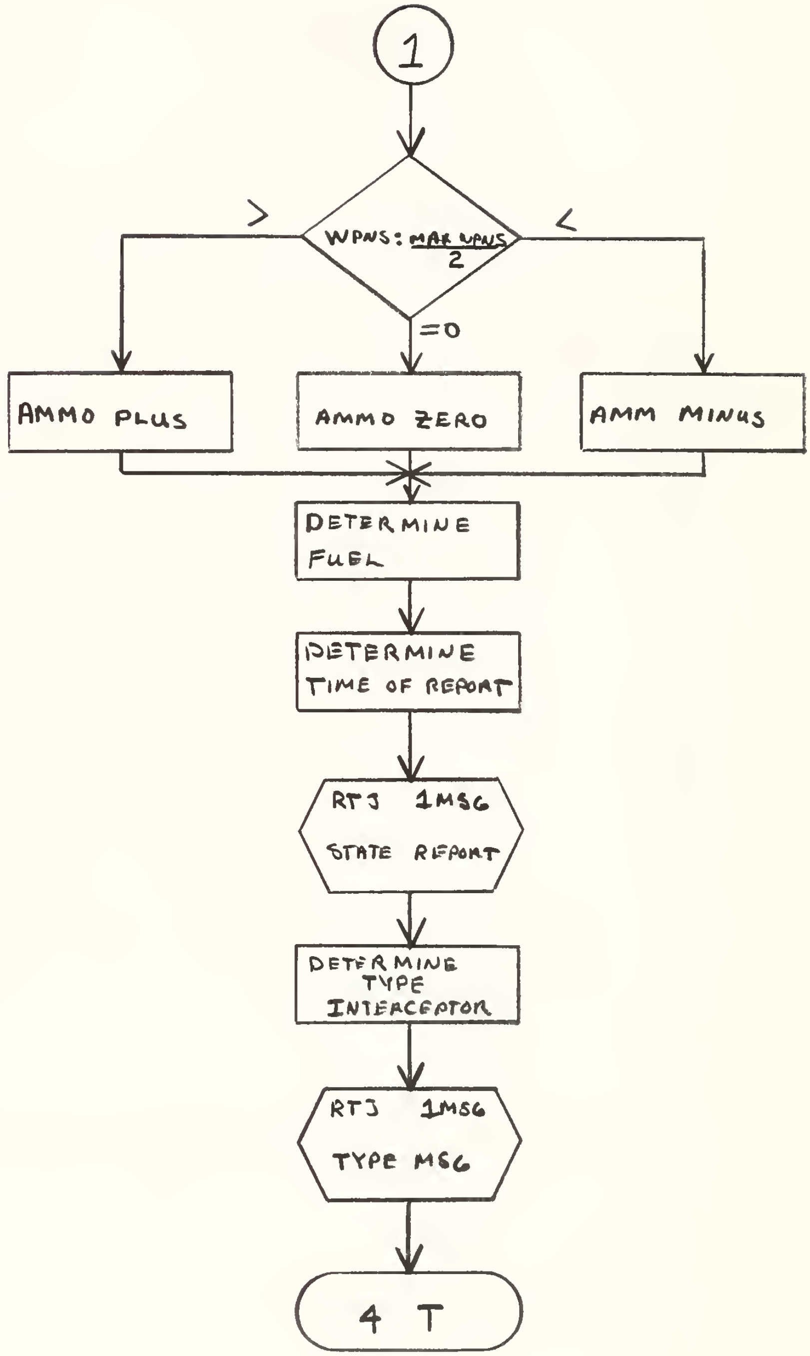


Figure I-33

State Routine





Track Generator
Calling Subprogram

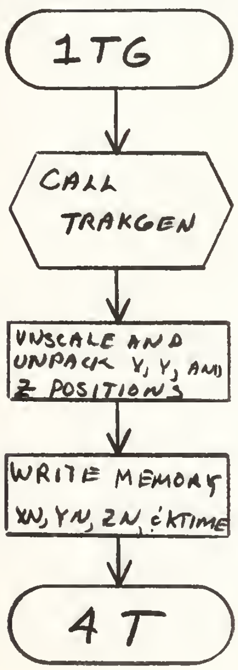


Figure I-34

Executive Routine

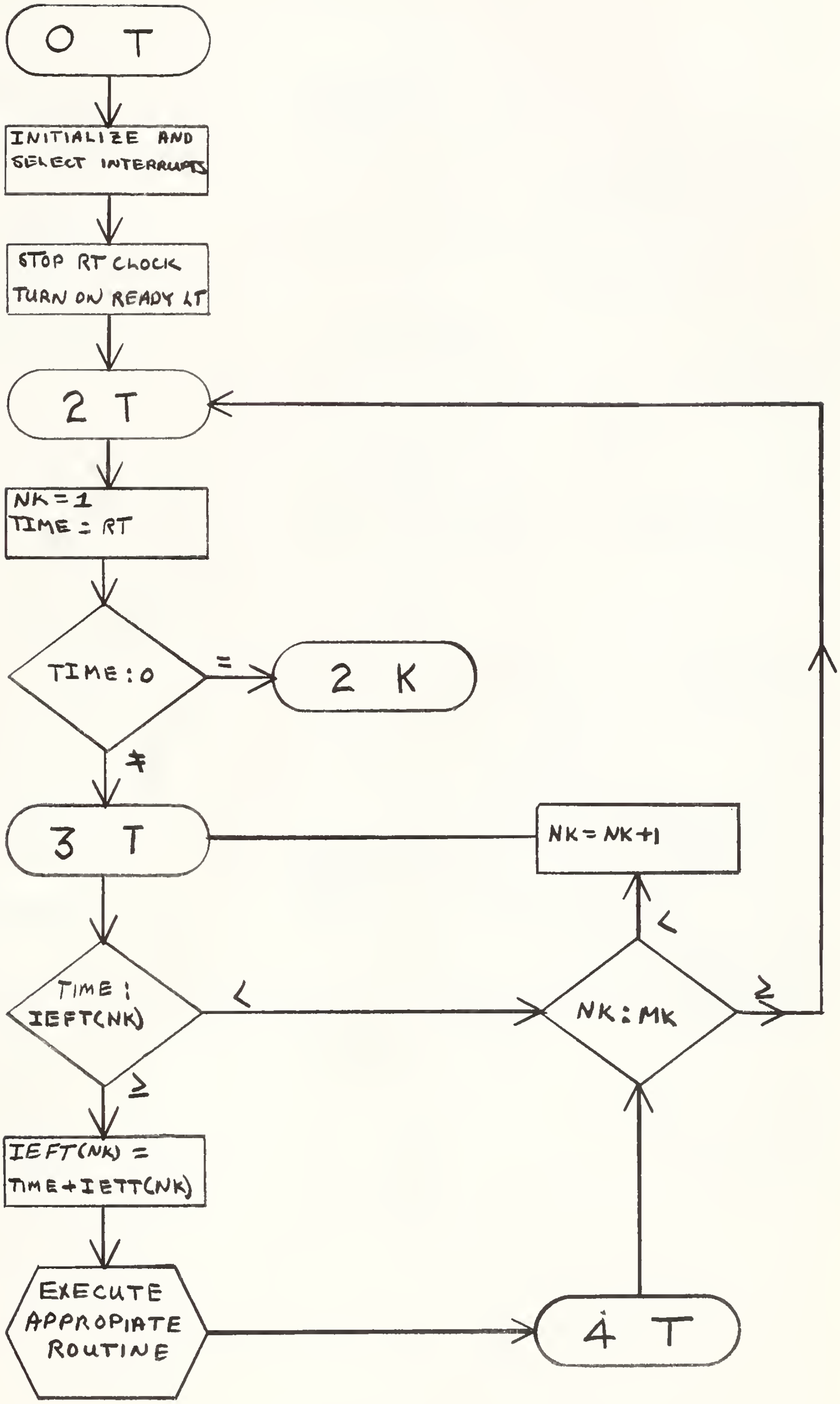


Figure I-35

INFO Routine

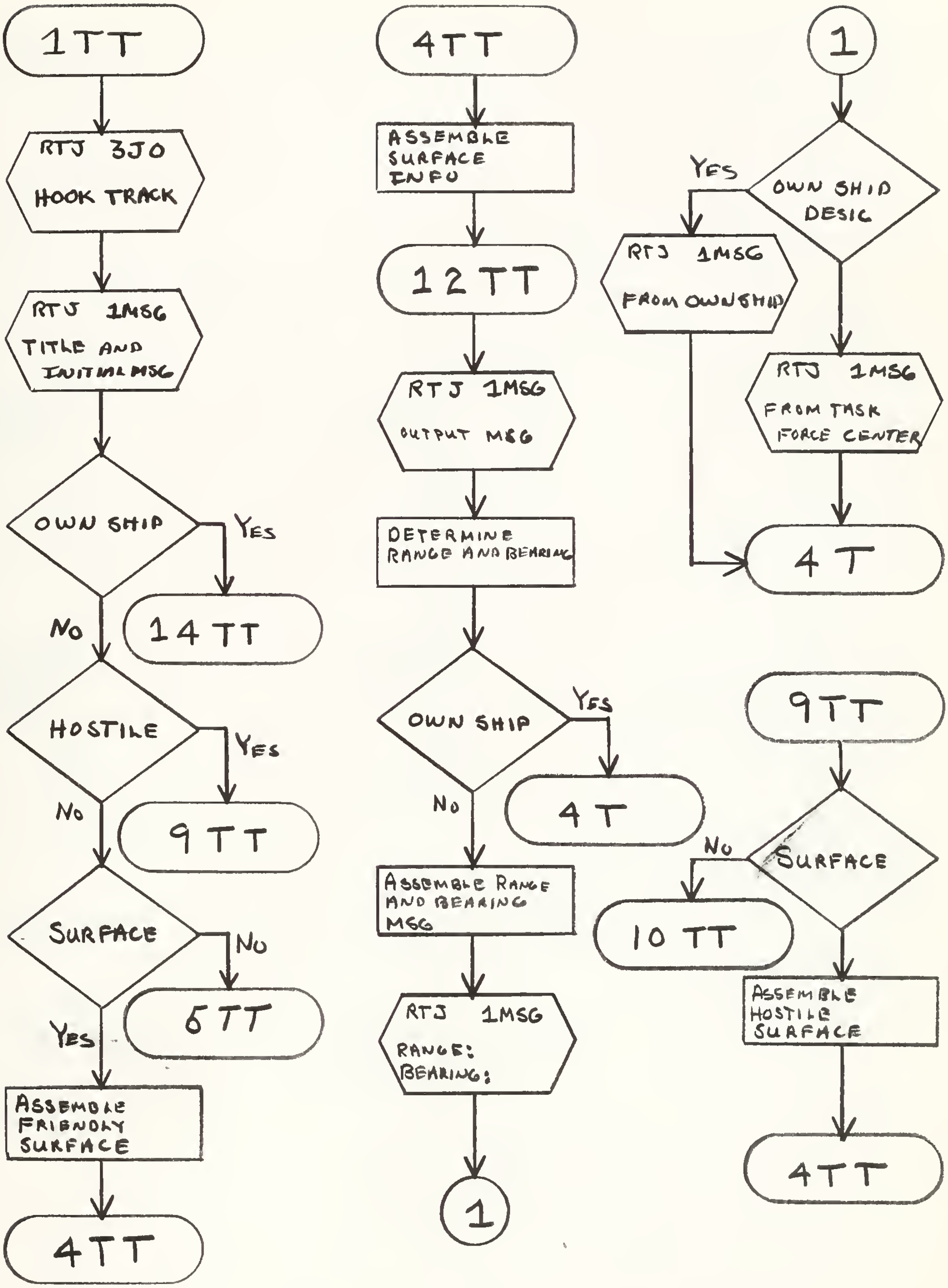


Figure I-36

INFO Routine (cont)

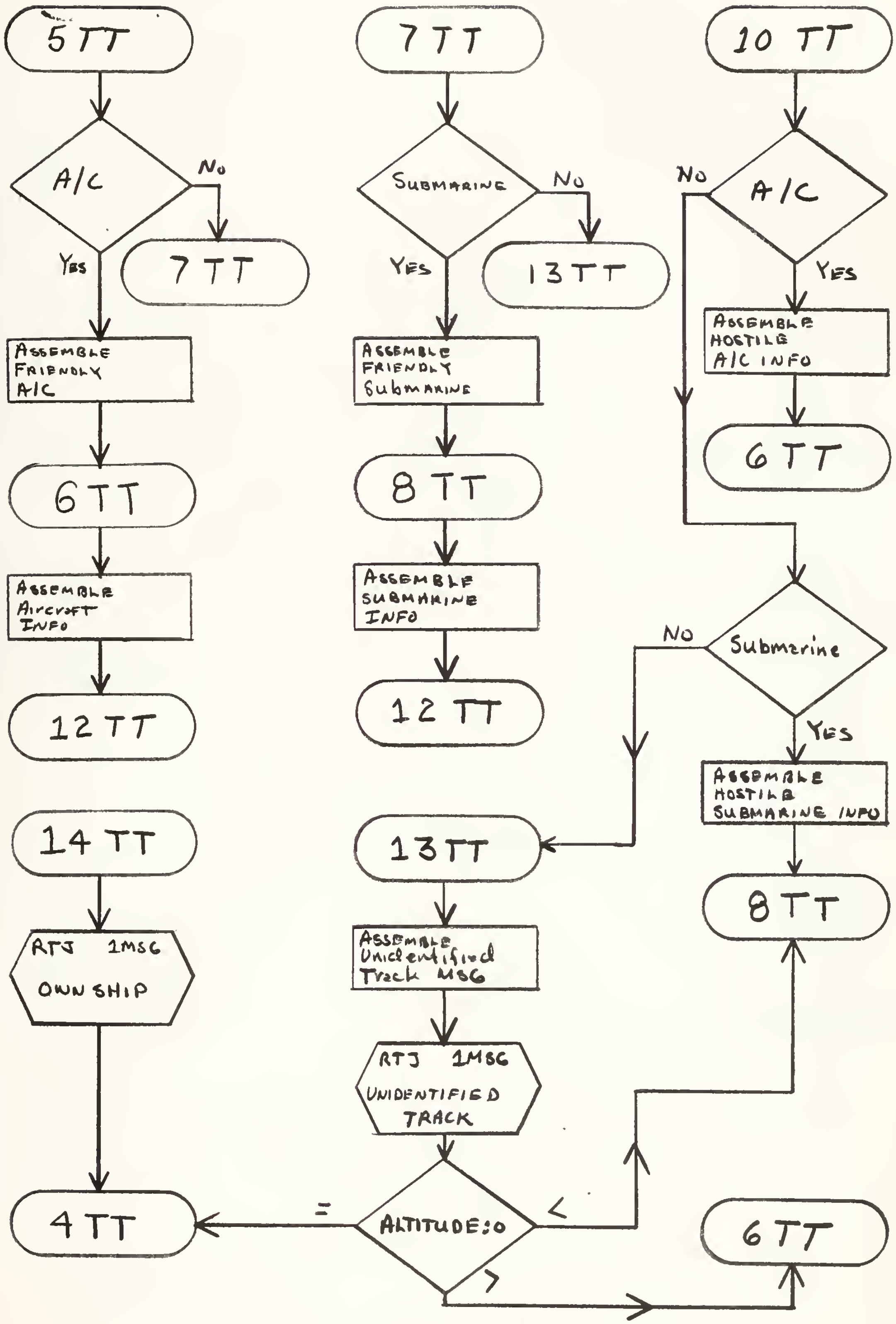


Figure I-37

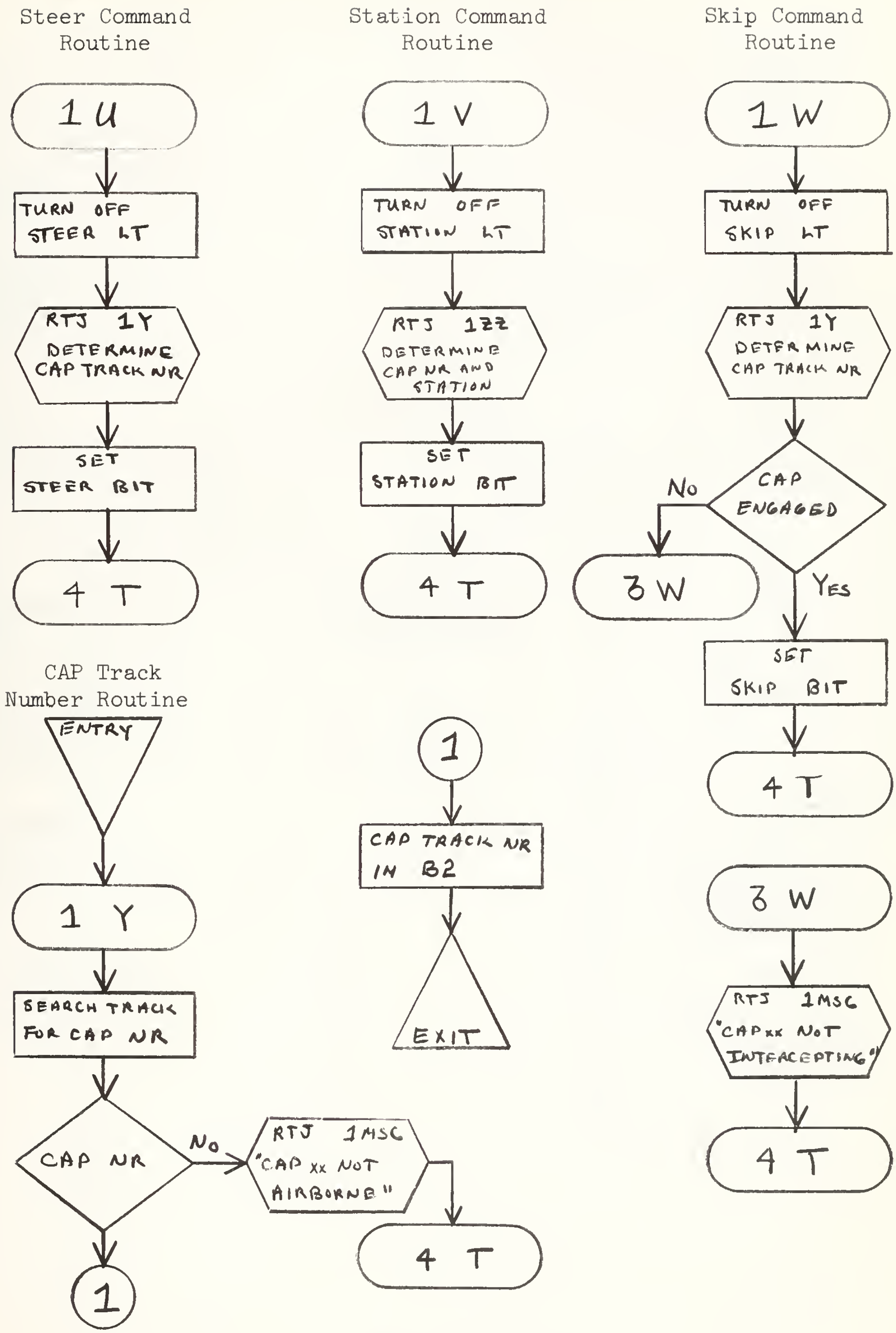

Steer Command Routine
1U
TURN OFF STEER LT
RTJ 1Y DETERMINE CAP TRACK NR
SET STEER BIT
4 T
Station Command Routine
1V
TURN OFF STATION LT
RTJ 1ZZ DETERMINE CAP NR AND STATION
SET STATION BIT
4 T
Skip Command Routine
1W
TURN OFF SKIP LT
RTJ 1Y DETERMINE CAP TRACK NR
CAP ENGAGED
No
3W
Yes
SET SKIP BIT
4 T
CAP Track Number Routine
ENTRY
1 Y
SEARCH TRACK FOR CAP NR
CAP NR
No
RTJ 1MSC "CAP xx NOT AIRBORNE"
4 T
1
1
CAP TRACK NR IN B2
EXIT
3 W
RTJ 1MSC "CAP xx NOT INTERCEPTING"
4 T

Figure I-38

Launch Command Routine

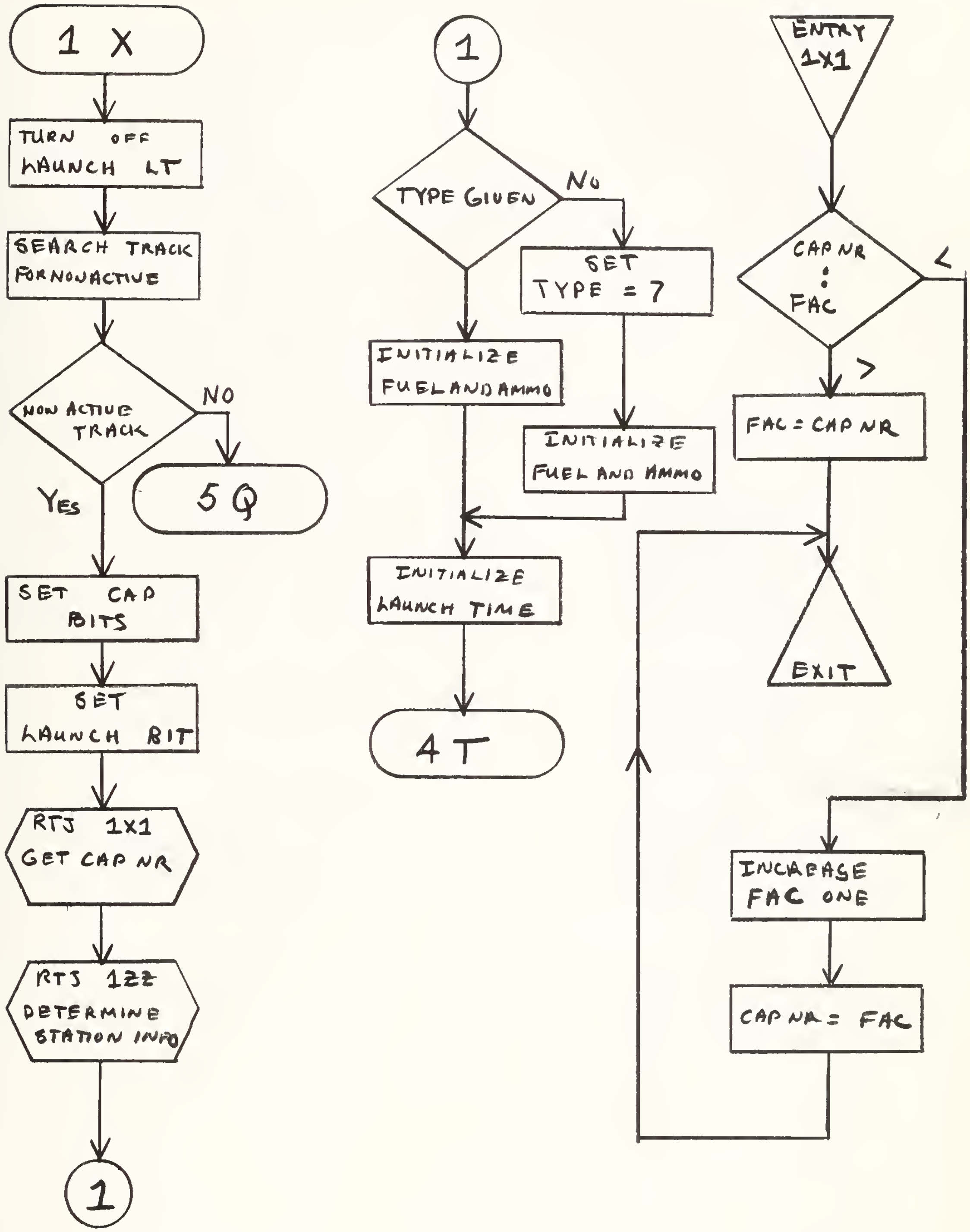


Figure I-39

Begin Routine

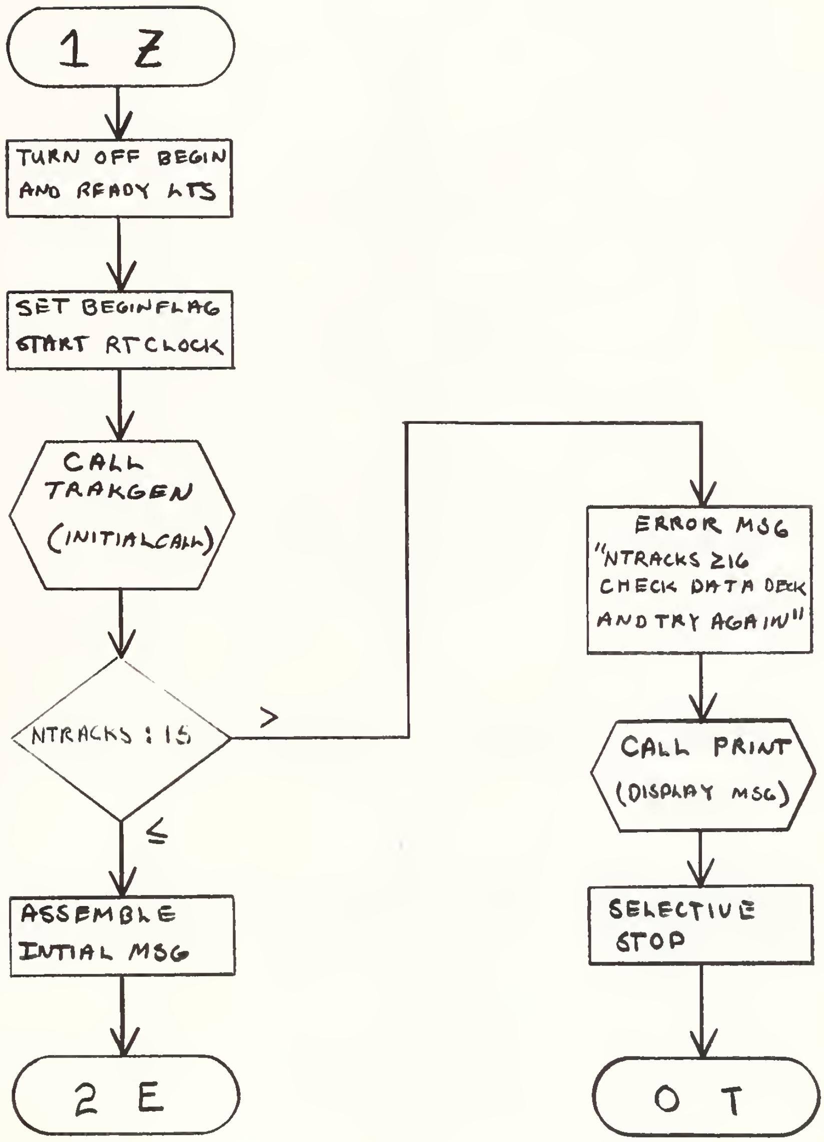


Figure I-40

CAP Track Number and Station Routine

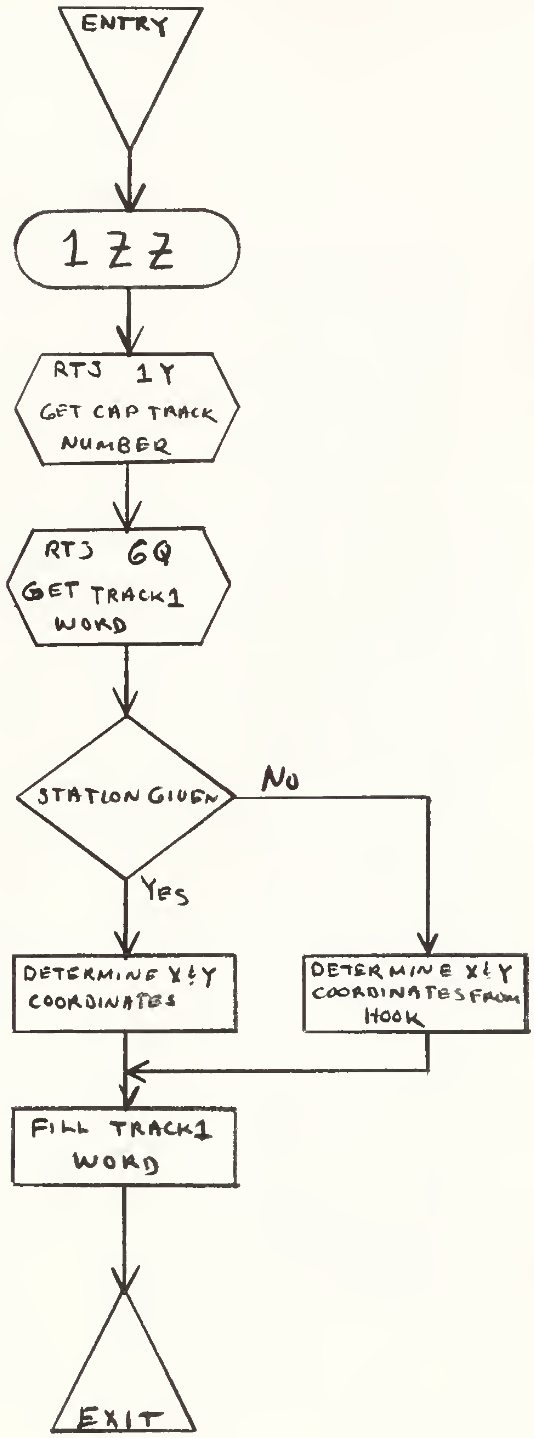


Figure I-41

APPENDIX I-C

SIMONE PROGRAM LISTING

```
      MACHINE SIMONE

      RSV(IPOSIT=15,IHEAD=15,JHEAD=15,IDD65=10,IEFT=6,IETT=6,BUF1=120)
      RSV(T=11,ERRMSG=4,R=4,D=15,F=15,TRACK=15,TRACK1=15,TRACK2=15)
      RSV(SPD=15,CRS=15,KTIME=4,MACH=15)
      RSV(XHIS1=24,YHIS1=24,XHIS2=24,YHIS2=24,IXH=15,IYH=15)
      RSV(XN=15,YN=15,ZN=15,XP=15,YP=15,ZP=15,XS=15,YS=15,ZS=15,
     *    VXS=15,VYS=15,VZS=15,LHEAD=60,KFIG=15)
      RSV(X=299,X1=299,X2=299,Y=299,Y1=299,Y2=299)

      LIB(TRAKGEN,PRINT,DRAW,FLOATFIX=FLOAT,FIXFLOAT=FIX,COURSE)
      LIB(SQRTF=SQ,COSF,SINF)

      COMMON IPOSIT,IHEAD,JHEAD,LHEAD,IDD65,ISCALE1,ISCALE2,SCALE1,
     *       SCALE2,TIME

      LOC(Z=0,ITIME=50)

* BCD CODE TABLE

      CON(F0 =12B,F1 =01B,F2 =02B,F3 =03B,F4 =04B,F5 =05B,F6 =06B,
     *    F7 =07B,F10=10B,F11=11B,F12=61B,F13=62B,F14=63B,F15=64B,
     *    F16=65B,F17=66B,F20=67B,F21=70B,F22=71B,F23=41B,F24=42B,
     *    F25=43B,F26=44B,F27=45B,F30=46B,F31=47B,F32=50B,F33=51B,
     *    F34=22B,F35=23B,F36=24B,F37=25B,F40=26B,F41=27B,F42=30B,
     *    F43=31B,F44=73B,F45=53B,F46=34B,F47=13B,F50=54B,F51=21B,
     *    F52=60B,F53=40B,F54=74B,F55=33B,F56=20B,F57=14B,F60=52B)

* SPECIAL SYMBOLS FOR DD-65 DISPLAY

      CON(S1  = 0700524354455647B, S1A = 5041010454540401B,
     *    S1B = 0001565677000000B, S1C = 0000000000000000B,
     *    S2  = 1046455443427700B, S2A = 0000000000000000B,
     *    S2B = 0000000000000000B, S2C =          0000000B,
     *    S3  = 0700524354455647B, S3A = 5041770000000000B,
     *    S3B = 0000000000000000B, S3C =          0000000B,
```

```
*     S4  = 1042435445467700B, S4A = 0000000000000000B,
*     S4B = 0000000000000000B, S4C =          0000000B,
*     S5  = 1012141416565050B, S5A = 5277000000000000B,
*     S5B = 0000000000000000B, S5C =          0000000B)
 CON(S6  = 1052545456565050B, S6A = 5277000000000000B,
*     S6B = 0000000000000000B, S6C =          0000000B,
*     S7  = 1052545456770000B, S7A = 0000000000000000B,
*     S7B = 0000000000000000B, S7C =          0000000B,
*     S8  = 1053557700000000B, S8A = 0000000000000000B,
*     S8B = 0000000000000000B, S8C =          0000000B,
*     S9  = 1013155751770000B, S9A = 0000000000000000B,
*     S9B = 0000000000000000B, S9C =          0000000B,
*     S10 = 1053555751770000B, S10A= 0000000000000000B,
*     S10B= 0000000000000000B, S10C=          0000000B)
 CON(S11 = 5016155252525215B, S11A= 1657770000000000B,
*     S11B= 0000777777777777B, S11C= 7300000000000000B,
*     S12 = 5053555077000000B, S13 = 5052545456507700B)

* RAW DATA DISPLAY SYMBOLS

 CON(C360 = 4310164377000000B, C090 = 4116144177000000B,
*     C180 = 4310164377000000B, C270 = 4116147700000000B)

* MASK FOR LOADING AND UNPACKING

 CON(MASK4B = 0070000000000000B, MASK8U = 0030000000000000B,
*     MASK9  = 0000000777777777B, MASK9L = 5000000002000030B)
 CON(MASK10 = 7777777777777700B, MASK11 = 0000000000000077B,
*     MASK12 = 0000000000000070B, MASK13 = 7400000000000000B,
*     MASK14 = 4000000000000000B, MASKX  = 7777777700000000B,
*     MASKY  = 0000000077777777B, MASK15 = 7777777777777777B,
*     MASK16 = 1000000000000000B, MASK17 = 2000000000000000B,
*     MASK18 = 0400000000000000B, MASK19 = 0000000040000000B,
*     MASK20 = 0000000020000000B, MASK21 = 0000000010000000B,
*     MASK22 = 0000000004000000B, MASK23 = 0000000002000000B,
*     MASK24 = 0000000001000000B, MASK25 = 0000000000400000B,
```

```
*    MASK26 = 0000000000200000B, MASK27 = 0000000000100000B)
 CON(MASK30 = 0070000000000000B, MASK31 = 0000000770000000B,
*    MASK32 = 0000007700000000B, MASK33 = 0000000050600000B,
*    MASK34 = 0000000007700000B, MASK35 = 0000077000000000B)
 CON(MASK36 = 7700000000000000B, MASK37 = 0000007777777777B,
*    MASK38 = 7777777700777777B, MASK39 = 0000007700777777B)
 CON(MASK40 = 0100000000000000B)

* PROGAM NAMES

 CON(P0  = 6265677145000000B, P1  = 7145236347230000B,
*    P2  = 4361244563700000B, P3  = 7071222346513000B,
*    P4  = 6551514651000000B, P5  = 7046436400000000B,
*    P6  = 2242714700000000B, P7  = 6471224743613000B,
*    P8  = 6451464700000000B, P9  = 4546232024226564B,
*    P10 = 2223464700000000B, P11 = 2223656551000000B,
*    P12 = 6465227167000000B, P13 = 7145664600000000B,
*    P14 = 4546232024226564B, P15 = 6751614770000000B,
*    P16 = 2223612371464500B, P17 = 6370614567650000B,
*    P18 = 2223612365000000B, P19 = 4546232024226564B)
 CON(P20 = 5165616430204323B, P21 = 6661646500000000B,
*    P22 = 2445666164650000B, P23 = 2223615123000000B,
*    P24 = 2263512462000000B, P25 = 4746512300000000B,
*    P26 = 2223626400000000B, P27 = 2247656564000000B,
*    P28 = 6143237123246465B, P29 = 4546232024226564B)

* BCD MESSAGE CODE TABLE

 CON(E0 =4751466751614420B,   E1 =4546232063614343B,
*    E2 =6564000000000000B,   E3 =2346462044614530B,
*    E4 =2061516724446545B,   E5 =2322000000000000B,
*    E6 =0245642061516724B,   E7 =4465452320714343B,
*    E8 =6567614373000000B,   E9 =0002023516163 42B,
*    E10=2220244564655120B,   E11=3046245120634645B,
*    E12=2351464373000000B,   E13=2361516765232045B,
*    E14=4623207046464265B,   E15=6400000000000000B)
```

```
 CON(E16=0000000000000000B,   E17=2445616243652023B,
*     E18=4620714523655163B,   E19=6547232040206624B,
*     E20=6543000000000000B,   E21=0          00000B,
*     E22=2445616243654023B,   E23=4620714523655163B,
*     E24=6547232040206144B,   E25=4446000000000000B,
*     E26=0000000000000000B,   E27=4346262022236123B,
*     E28=6500000000000000B,   E29=00          0000B,
*     E30=4461306461307373B,   E31=7344613064613073B,
*     E32=7373000000000000B,   E33=00         00000B,
*     E34=6661447122706564B,   E35=0000000000000000B)
 CON(E36=0000000000000000B,   E37=7145226551636547B,
*     E38=2371456720402024B,   E39=4561624365202346B,
*     E40=2051652324514500B,   E41=0000000000000000B,
*     E42=4746465120714523B,   E43=6551636547230000B,
*     E44=2247436122702020B,   E45=2062466765302065B,
*     E46=0000000000000000B,   E47=7065616422202447B,
*     E48=2062466765302065B,   E49=0000000000000000B)
 CON(E50=4523516163422220B,   E51=3720010620202020B,
*     E52=6370656342206461B,   E53=2361206465634220B,
*     E54=6145642023513020B,   E55=6167617145730000B)
 CON(E56=4546206361472061B,   E57=2561714361624365B,
*     E58=2023462062652043B,   E59=6124456370656400B,
*     E60=0000000000000000B,   E61=2020454623206171B,
*     E62=5162465145650000B,   E63=0000000000000000B,
*     E64=2020454623204461B,   E65=4271456720614520B,
*     E66=7145236551636547B,   E67=2373000000000000B)
 CON(E68=6751614770206265B,   E69=7145672047434623B,
*     E70=2365647300000000B,   E71=6751614770206671B,
*     E72=4571227073000000B,   E73=2020202071456646B,
*     E74=5144612371464500B,   E75=2020202020202020B,
*     E76=2351616342204551B,   E77=0000000000000000B,
*     E78=2020202020202020B,   E79=0000000000000000B,
*     E80=0000000000000000B,   E81=0          00000B,
*     E82=0000000000000000B,   E83=0*         00000B)
 CON(E84=2223612365205165B,   E85=4746512320464520B,
*     E86=0000000000000000B,   E87=2020206144444620B,
```

```
*     E88=0000000000000000B,
*     E89=4627306765452047B,   E90=4324222021206624B,
*     E91=0000000000000000B,   E92=2047462445642220B,
*     E93=2120237144652020B,   E94=0000000000000000B)
 CON(E95=4626452022707147B,   E96=5161456765152020B,
*     E97=2030615164220000B,   E98=6265615171456715B,
*     E99=2064656751656522B,  E100=6651464420462645B,
*    E101=2022707147000000B,  E102=6651464420236122B,
*    E103=4266465163652063B,  E104=6545236551000000B,
*    E105=2044714365220000B)

* GRAPH TITLES

      CON(ITITLE = 0)
      CON(TITLE1  = 2271442046456520B, TITLE2  = 2020202020614323B,
*         TITLE3  = 7123246465202522B, TITLE4  = 7320237144652020B,
*         TITLE5  = 2020202020202020B, TITLE6  = 2020202020202020B,
*         TITLE7  = 0000000000000000B, TITLE8  = 2023714465202020B,
*         TITLE9  = 0000000000000000B, TITLE10 = 0000000000000000B,
*         TITLE11 = 4524236522202020B, TITLE12 = 2020202020202020B)
      CON(KTITLE = 0)
      CON(TITLE13 = 2271442046456520B, TITLE14 = 2020202020274030B,
*         TITLE15 = 2047462271237146B, TITLE16 = 4520474346232020B,
*         TITLE17 = 2020202020202020B, TITLE18 = 2020202020202020B,
*         TITLE19 = 0000000000000000B, TITLE20 = 2023714465202020B,
*         TITLE21 = 0000000000000000B, TITLE22 = 0000000000000000B,
*         TITLE23 = 4524236522202020B, TITLE24 = 2020202020202020B)

* PTABLE

      CON(PTABLE = 0, P1 = 48, P2 = 24, P3 = 12, P4 = 8, P5 = 6, P6 = 4,
*         P7 = 3, P8 = 2, P9 = 1)

* TIMING CONSTANTS FOR DISPLAY AND READBACK

      CON(PPO = 0000000000000000B, PP1 = 4000000000000000B,
```

```
*    PP2 = 4000000040000000B, PP3 = 4000400040004000B,
*    PP4 = 4010020040100200B, PP5 = 4040404040404040B,
*    PP6 = 4210421042104210B, PP7 = 4444444444444444B,
*    PP8 = 5252525252525252B, PP9 = 7777777777777777B)

* AIRCRAFT CONTROL CONSTANTS

     CON(LHEAD1 = 4020000000000000B, LHEAD2 = 4040000000000036B,
*        LHEAD3 = 4060000000024000B, LHEAD4 = 4100000000000012B,
*        LHEAD5 = 4040000003000017B, LHEAD6 = 4720000000000036B,
*        LHEAD7 = 4060000120024000B, LHEAD8 = 4200000000000000B,
*        LHEAD9 = 4100000170000012B)

* INITITAL INTERCEPTOR INFORMATION

     CON(IFIG0 = 0000000000000000B, IFIG1 = 0000000000000000B,
*        IFIG2 = 2064002317404301B, IFIG3 = 2064004317404301B,
*        IFIG4 = 3260440437644300B, IFIG5 = 3260444437644300B,
*        IFIG6 = 3506440437644300B, IFIG7 = 3506444437644300B)

* SECOND WORD INTERCEPTOR INFORMATION

     CON(FIG  = 0.00, FIG1 = 0.00, FIG2 = 0.67, FIG3 = 0.67,
*        FIG4 = 1.18, FIG5 = 1.15, FIG6 = 1.15, FIG7 = 1.30)

* MANEUVER CODE TABLE

     CON(MAN1 = 05B, MAN2 = 06B, MAN3 = 01B, MAN4 = 10B, MAN5 = -2B,
*        MAN6 = 02B, MAN7 = 04B, MAN8 = 03B)

* BUTTON TABLE

     CON(T0  = 01B, T1  = 02B, T2  = 03B, T3  = 04B, T4  = 05B,
*        T5  = 11B, T6  = 12B, T7  = 13B, T8  = 14B, T9  = 15B,
*        T10 = 21B, T11 = 22B, T12 = 23B, T13 = 24B, T14 = 25B,
*        T15 = 31B, T16 = 32B, T17 = 33B, T18 = 34B, T19 = 35B,
```

```
*     T20 = 41B, T21 = 42B, T22 = 43B, T23 = 44B, T24 = 45B,
*     T25 = 51B, T26 = 52B, T27 = 53B, T28 = 54B, T29 = 55B)

* LIGHT  TABLE

      CON(LT0  = 77202B, LT1  = 77204B, LT2  = 77210B, LT3  = 77220B,
*     LT4  = 77240B, LT5  = 77302B, LT6  = 77304B, LT7  = 77310B,
*     LT8  = 77320B, LT9  = 77340B, LT10 = 77402B, LT11 = 77404B,
*     LT12 = 77410B, LT13 = 77420B, LT14 = 77440B, LT15 = 77502B,
*     LT16 = 77504B, LT17 = 77510B, LT18 = 77520B, LT19 = 77540B,
*     LT20 = 77602B, LT21 = 77604B, LT22 = 77610B, LT23 = 77620B,
*     LT24 = 77640B, LT25 = 77702B, LT26 = 77704B, LT27 = 77710B,
*     LT28 = 77720B, LT29 = 77740B)

* TURN OFF LIGHT CONSTANTS

      CON(OFF0 = 77203B, OFF1 = 77205B, OFF2 = 77211B, OFF3 = 77221B,
*     OFF4 = 77241B, OFF5 = 77303B, OFF6 = 77305B, OFF7 = 77311B,
*     OFF8 = 77321B, OFF9 = 77341B, OFF10= 77403B, OFF11= 77405B,
*     OFF12= 77411B, OFF13= 77421B, OFF14= 77441B, OFF15= 77503B,
*     OFF16= 77505B, OFF17= 77511B, OFF18= 77521B, OFF19= 77541B,
*     OFF20= 77603B, OFF21= 77605B, OFF22= 77611B, OFF23= 77621B,
*     OFF24= 77641B, OFF25= 77703B, OFF26= 77705B, OFF27= 77711B,
*     OFF28= 77721B, OFF29= 77741B)

* OCTAL TO BCD CONVERSION FOR NUMBERS 0-15
      CON(OB0  = 4546B, OB1  = 1201B, OB2  = 1202B, OB3  = 1203B,
*     OB4  = 1204B, OB5  = 1205B, OB6  = 1206B, OB7  = 1207B,
*     OB8  = 1210B, OB9  = 1211B, OB10 = 0112B, OB11 = 0101B,
*     OB12 = 0102B, OB13 = 0103B, OB14 = 0104B, OB15 = 0105B)

* ASSORTED CONSTANTS

      CON(R0 = 01, R1 = 10, R2 = 100, R2A = 1000, R2B = 10000,
*     R2C= 100000, R2D= 1000000, R3 = 0000007300204020B,
```

```
*     R4 = 00000073002044718, R5 = 43412700000000008,
*     R6 = 77777777777777578, R7 = 43310200000000138,
*     R8 = 00022744000000008, R9 = 00332051432320208,
*     R10= 00000000530000008, R11= 62466765302020658,
*     R12= 66517165456443308, R13= 20222245200706008,
*     R14= 00232064654723708, R15= 00730000000000008,
*     R16= 02220425220000008, R17= 67466242714520008,
*     R18= 20644367452010008, R19= 45464540646522718)
 CON(R20= 67456123656473008, R21= 44714524222021208,
*     R22= 31655146202021208, R23= 47432422202021208,
*     R24= 45230000000000008, R25= 0000  0011530000B)
 CON(R26= 03102745000000008, R27= 20201200120000008)
 CON(RHO = 0., RH1 = 64., RH2 = 32., RH3 = 16., RH4 = 8., RH5 = 4.,
*     RH6 = 2., RH7 = 1.)
 CON(PIP = 3000350000005000B, CLK = 3777777777777704B)
 CON(CLK5 = 3777777777777742B)
 CON(ERASE  = 0000000000041000B, ERASE1 = 0000000017101000B)
 CON(ERASE2 = 0000000014405000B)
 CON(XIP = -377B, YIP = -377B)
 CON(ALPHA=.085,BETA=.009)
 CON(AXIS = 450., RADIANS = 0.01745)
 CON(ZERO = 0.0, EIGHT =8.0)
 CON(BUTTONS = 30)
 CON(ZTIME= 1800)

1RET   SLJ   (N)          SLJ   (OT)       . EXIT/ENTRY                                  00000

* FORMATS USED IN PROGRAM

1      FORMAT(4A8)                                                                      00001
100    FORMAT(6A8)                                                                      00002
101    FORMAT(A4,1X,E6.0,1X,E6.0,1X,I2,1X,I1,1X,I1,1X,I1,1X,I1,1X,I2,1X,             00003
*            I1,1X)
102    FORMAT(A8)                                                                       00004
103    FORMAT(F10.0)                                                                    00005
104    FORMAT(I2,2X,I6)                                                                 00006
```

```
200    FORMAT(17H1   DUTY CYCLE =  F7.5, 16H FOR THIS RUN.   )            00007

* INTERRUPT ROUTINE

1A     EXF 7 (00131B)    SLJ   (0A)      . TIME ADVANCE                  00010
       EXF 7 (77173B)    SLJ   (2A)      . KBD 1 INTER ROUTINE           00011
       EXF 7 (77174B)    SLJ   (4A)      . KBD 2 INTER ROUTINE           00012
1AA    EXF   (00070B)    EXF   (77111B)  . CLEAR BOTH ARITHEMTIC AND     00013
       SLJ   (Z+7B)      ZRO   (0)       .   DD65 INTERRUPTS             00014

* TIME ADVANCE ROUTINE

0A     EXF   (00070B)    STA   (ASAVE)   . CLEAR FAULT                   00015
       SSK   (Z)         SLJ   (Z+7B)    . SENSE ZERO OVERFLOW           00016
       LDA   (CLK)       STA   (Z)       . RESET ONE-SECOND COUNT        00017
       SLJ 5 (L+4)       ENI   (0)       . DOUBLE TIME ROUTINE           00020
       RAO   (Z+50B).    EXF   (77105B)  . INCREMENT TIME AT 50          00021
       EXF   (77103B)    LDA   (ASAVE)   . SELECT INTERRUPTS             00022
       SLJ   (Z+7B)      ZRO   (0)       . EXIT THROUGH 7                00023
       SLJ   (N)         LDA   (CLK5)    . RESET ONE HALF SECOUND COUNT  00024
       STA   (Z)         SLJ   (L-1)     .                               00025

* KEYBOARD PROCESSOR   NUMBER 1
2A     SIU 4 (3A)        STA   (ASAVE)   . SAVE INDEX B4                 00026
       STQ   (QSAVE)     LIL 4 (KNR1)    . SAVE A AND Q REGISTERS        00027
       EXF   (77140B)    EXF 7 (77140B)  . B4 = KNR1   SELECT KBD1       00030
       LIL 4 (KNR1)      EXF   (77140B)  . B4 = KNR1   SELECT KBD1       00031
       INT   (BUF1)      LDA   (BUF1)    . INPUT CHARACTER               00032
       EQS   (74B)       SLJ   (L+2)     . CHECK FOR CLOSING PAREN       00033
       SLJ   (11A)       ZRO   (0)       . YES JUMP TO PRINT ROUTINE     00034
       EQS   (73B)       SLJ   (L+2)     . CHECK FOR PERIOD              00035
       SLJ   (11A)       ZRO   (0)       . YES JUMP TO PRINT ROUTINE     00036
       EQS   (76B)       SLJ   (L+2)     . CHECK FOR CR                  00037
       SLJ   (9A)        ZRO   (0)       .    YES JUMP TO 9A             00040
       EQS   (36B)       SLJ   (L+2)     . CHECK FOR TAB                 00041
       SLJ   (10A)       ZRO   (0)       .    YES JUMP TO 10A            00042
```

```
        STA 4 (BUF1)      RAO    (KNR1)      . INCREASE POINTER                 00043
3A      ENI 4 (N)         EXF    (77001B)    . RESTORE B4   DESELECT            00044
        LDA   (ASAVE)     LDQ    (QSAVE)     . RESTORE A AND Q REGISTERS        00045
        EXF   (77111B)    SLJ    (Z+7B)      . EXIT THRU CELL 7                 00046

* KEYBOARD PROCESSOR      NUMBER 2

4A      SIU 3 (6A)        SIL 4 (6A)         . SAVE INDESES                     00047
        SIU 5 (7A)        SIL 6 (7A)         .    3, 4, 5, 6                    00050
        STA   (ASAVE)     STQ   (QSAVE)      . SAVE A AND Q REGISTERS           00051
        LDA   (PFLAG)     AJP 1 (8A)         . CHECK P-FLAG                     00052
        LDQ   (MASK11)    EXF   (77120B)     . SELECT KBD2                      00053
        INT   (KBD2WD)    LDL   (KBD2WD)     . INPUT                            00054
        STA   (KBD2WD)    LIL 3 (BUTTONS)    .                                  00055
        EQS 3 (T0)        SLJ   (6A)         . WHICH BUTTON HIT                 00056
        LDA 3 (LT0)       SAU   (L+1)        .                                  00057
        EXF   (N)         LDA 3 (0TAB)       . TURN ON CORRECT LT               00060
        SAU   (5A)        ARS   (24)         .                                  00061
        STA   (T+1)       LIL 5 (T+1)        . STORE NUMBER OF LETTERS          00062
        LIL 6 (KNR1)      LDA   (KNR1)       .   IN THE NAME                    00063
        INA 5 (0)         STA   (KNR1)       . INCREAS KNR1 BY NR LETTERS       00064
5A      LDA   (N)         ENI   (0)          .                                  00065
        ENQ   (0)         LLS   (6)          . TRANSFER WORD TO BUF1            00066
        STQ 6 (BUF1)      INI 6 (1)          .                                  00067
        IJP 5 (L-2)       ENI   (0)          . LOOP                             00070
6A      ENI 3 (N)         ENI 4 (N)          . RESTORE INDEXES                  00071
7A      ENI 5 (N)         ENI 6 (N)          .   3, 4, 5, 6                     00072
        LDA   (ASAVE)     LDQ   (QSAVE)      .                                  00073
        EXF   (77001B)    EXF   (77111B)     . DESELECT AND CLEAR               00074
        SLJ   (Z+7B)      ZRO   (0)          . EXIT THRU CELL 7                 00075
8A      EXF   (77120B)    INT   (KBD2WD)     . INPUT                            00076
        LDQ   (MASK11)    LDL   (KBD2WD)     . MASK MATRIX CODE                 00077
        EQS   (F11)       SLJ   (6A)         . CHECK FOR PAUSE                  00100
        ENA   (0)         STA   (PFLAG)      . RESET P-FLAG                     00101
        SLJ   (6A)        ZRO   (0)          . RETURN                           00102
```

```
* ERASE ON CARRIAGE RETURN

9A     ENA    (1)          STA    (KNR1)      . RESET POINTERS
       EXF    (77241B)     LIL 4  (BUTTONS)   . TURN OFF ERROR LT
       LDA 4  (OFFO)       SAU    (L+1)       .
       EXF    (N)          IJP 4  (L-1)       . TURN OFF ALL LTS
       ENA    (0)          ENI 4  (120)       . ZERO OUT BUFFER
       STA 4  (BUF1)       IJP 4  (L)         .
       EXF    (02000B)     EXF    (04001B)    . STOP RT CLOCK AND DISALLOW
       ENI 4  (484)        ENI    (0)         . ERRASE DD65 MEMORY
       EXF    (77010B)     OUT    (ERASE1)    .   FROM ADDRESS
       EXF    (77010B)     OUT    (S6C)       .     484 TO 510
       ISK 4  (510)        SLJ    (L-1)       . LOOP
       EXF    (04000B)     EXF    (01000B)    . ALLOW AND START RT CLOCK
       SLJ    (3A)         ZRO    (0)         . RETURN

* ERASE ON TAB

10A    ENA    (353B)       STA    (YMSG)      . RESET LINE
       ENA    (400)        STA    (ADD)       . RESET MSG ADDRESS IN DD65
       ENI 4  (400)        ENI    (0)         .
       EXF    (02000B)     EXF    (04001B)    . STOP RT CLOCK AND DISALLOW
       EXF    (77010B)     EXF 7  (77010B)    . ERASE DD65 MEMORY
       OUT    (ERASE2)     ENI    (0)         .   FROM ADDRESS
       EXF    (77010B)     OUT    (S6C)       .     400 TO 482
       ISK 4  (482)        SLJ    (L-1)       . LOOP
       EXF    (04000B)     EXF    (01000B)    . ALLOW AND START RT CLOCK
       SLJ    (3A)         ZRO    (0)         . RETURN

* PRINT ROUTINE FOR BUFFER

11A    STA 4  (BUF1)       RAO    (KNR1)      . STORE CLOSING PAREN OR PERIOD
       SIU 2  (15A)        SIL 3  (15A)       . SAVE NEEDED INDEXES
       SIU 5  (16A)        SIL 6  (16A)       .   2, 3, 4, 5, 6
       SIU 4  (L+5)        SIL 4  (13A)       .
       ENI 3  (1)          ENI 2  (0)         . ASSEMBLE MESSAGE FOR PRINT
```

```
         ENQ    (0)           STQ    (T+1)                                            00137
         LDA    (T+1)         ENI 5  (1)        .                                     00140
         ALS    (6)           ADD 3  (BUF1)     .                                     00141
         ISK 3  (N)           SLJ    (L+2)      . CHECK FOR END OF LETTERS            00142
         SLJ    (L+4)         ZRO    (0)        .                                     00143
         ISK 5  (8)           SLJ    (L-3)      . CHECK FOR 8 CHARACTERS              00144
         STA 2  (D)           INI 2  (1)        . STORE WORD IN D ARRAY               00145
         SLJ    (L-6)         ZRO    (0)        . LOOP                                00146
         ISK 5  (8)           SLJ    (L+2)      . CHECK FOR POSITION OF LAST          00147
         STA 2  (D)           SLJ    (L+2)      .   WORD                              00150
         ALS    (6)           SLJ    (L-2)      . POSITON LAST WORD                   00151
         ENA    (XP1)         SAL    (PRINT+1)  . SET UP ARGUMENTS FOR PRINT          00152
         ENA    (YP1)         SAL    (PRINT+2)  .                                     00153
         LIL 5  (PRINT+4)     SIL 5  (T+1)      .                                     00154
         LIL 5  (PRO)         SIL 5  (PRINT+3)  .                                     00155
         ENI 5  (484)         SIL 5  (PRINT+4)  .                                     00156
         ENA    (D)           ENI    (0)        .                                     00157
         ENQ    (0B)          SLJ 4  (PRINT)    . JUMP TO PRINT                       00160
         LIL 5  (T+1)         SIL 5  (PRINT+4)  . RESTORE MEMOY ADDRESS IN PRINT      00161
         ENA    (0)           STA 2  (D)        . ERASE D ARRAY                       00162
13A      IJP 2  (L-1)         ENI 4  (N)        . LOOP                                00163
15A      ENI 2  (N)           ENI 3  (N)        . RESTORE INDEXES                     00164
16A      ENI 5  (N)           ENI 6  (N)        .   2, 3, 4, 5, 6                     00165
         SLJ    (3A)          ZRO    (0)        . RETURN THRU CELL 7                  00166

* ABSOLUTE VALUE ROUTINE

1ABS     SLJ    (N)           SLJ    (L+2)      . EXIT/ENTRY                          00167
2ABS     ZRO    (0)           ZRO    (0)        . ARGUMENT                            00170
         LDA 7  (2ABS)        AJP 2  (1ABS)     . A POSITIVE     RETURN               00171
         LAC 7  (2ABS)        SLJ    (1ABS)     . COMPLEMENT     RETURN               00172
         LDL    (MASK24)      AJP    (L+3)      .                                     00173
         LDA 1  (TRACK)       SST    (MASK20)   . CAP HAS ASSIGNED STATION, SET       00174
         SCL    (MASK24)      STA 1  (TRACK)    .   STATION BIT, CLEAR ORBIT BIT      00175
         LDQ 2  (TRACK)       LDL    (MASK10)   . REDESIGNATE AS                      00176
         STA 2  (TRACK)       ENA    (34B)      .   UNENGAGED                         00177
```

```
        RAD 2 (TRACK)      ENI    (0)        .    BOGIE                            00200
        LDQ 1 (TRACK)      LDL    (MASK10)   . REDESIGNATE AS                      00201
        STA 1 (TRACK)      ENA    (30B)      .   UNENGAGED CAP                     00202
        RAD 1 (TRACK)      LDL    (MASK20)   .                                     00203
        AJP   (L+1)        SLJ    (1IS)      . RETURN TO STATION                   00204
        LDA 1 (TRACK)      SST    (MASK24)   . SET ORBIT BIT, NO                   00205
        SLJ   (3IE)        ZRO    (0)        .   STATION ASSIGNED                  00206

* CHANGE ROUTINE

1C      SLJ   (L+8)        ZRO    (0)        . ENTRY                               00207
2C      ZRO   (0)          ZRO    (0)        . TRACK NUMBER                        00210
3C      ZRO   (0)          ZRO    (0)        . MANEUVER CALLED                     00211
4C      ZRO   (0)          ZRO    (0)        . 3RD ARGUMENT                        00212
5C      ZRO   (0)          ZRO    (0)        . 4TH ARGUMENT                        00213
6C      ZRO   (0)          ZRO    (0)        . 5TH ARGUMENT                        00214
7C      ZRO   (0)          ZRO    (0)        . 6TH ARGUMENT                        00215
8C      ZRO   (0)          ZRO    (0)        . 7TH ARGUMENT                        00216
        LDA   (2C)         STA    (IDD65+1)  . STORE TRACK NUMBER                  00217
        LDA   (3C)         ENI 4  (8)        .                                     00220
        EQS 4 (P21)        SLJ    (9C)       . SEARCH FOR MANEUVER                 00221
        LDA 4 (OFF21)      SAU    (L+1)      .                                     00222
        EXF   (N)          EXF    (77511B)   . TURN OFF CHANGE AND MAN LTS         00223
        LDA 4 (MAN1)       STA    (IDD65+2)  . STORE MANCODE                       00224
        LDA 4 (OZ10)       SAU    (L+1)      . SET ENTRY ADDRESS                   00225
        SLJ   (N)          ZRO    (0)        . JUMP TO PROPER ENTRY                00226
9C      ENA   (E6)         SLJ    (1E)       . ENTRY TO ERROR ROUTINE              00227
10C     LDA   (4C)         STA    (IDD65+3)  . STORE CRS                           00230
        LDA   (5C)         STA    (IDD65+4)  . STORE SPD                           00231
        LDA   (6C)         STA    (IDD65+6)  . STORE X POSIT                       00232
        LDA   (7C)         STA    (IDD65+7)  . STORE Y POSIT                       00233
        LDA   (8C)         STA    (IDD65+8)  . STORE Z POSIT                       00234
11C     ENI 4 (6)          ENA    (0)        . BEGIN EXIT ROUTINE                  00235
        STA 4 (2C)         IJP 4  (L)        . ZERO ALL ARGUMENTS                  00236
        ENA   (1)          STA    (IDD65+9)  . SET PERIOD FLAG                     00237
        SLJ 4 (TRAKGEN)    ZRO    (0)        . CALL TRAKGEN                        00240
```

```
        SLJ    (4T)         ZRO    (0)         . RETURN                          00241
12C     LDA    (4C)         STA    (IDD65+4)   . STORE SPD                       00242
        LDA    (5C)         AJP    (18C)       .                                 00243
        SLJ    (16C)        ZRO    (0)         .                                 00244
13C     LDA    (4C)         STA    (IDD65+3)   . STORE CRS                       00245
        LDA    (5C)         AJP    (17C)       .                                 00246
        SLJ 4  (FLOAT)      ZRO    (0)         . FLOAT RATE                      00247
        FDV    (10.)        STA    (5C)        .                                 00250
14C     LDA    (5C)         STA    (IDD65+10). STORE RATE                        00251
        SLJ    (11C)        ZRO    (0)         .                                 00252
15C     LDA    (4C)         STA    (IDD65+5)   . STORE ALTITUDE                  00253
        LDA    (5C)         AJP    (19C)       .                                 00254
16C     LDA    (5C)         SLJ 4  (FLOAT)     . FLOAT RATE                      00255
        STA    (5C)         SLJ    (14C)       .                                 00256
17C     LDA    (1.5)        STA    (5C)        . SET STD RATE TURN               00257
        SLJ    (14C)        ZRO    (0)         .                                 00260
18C     LDA    (10.)        STA    (5C)        . SET STD SPEED RATE              00261
        SLJ    (14C)        ZRO    (0)         .                                 00262
19C     LDA    (5000.)      STA    (5C)        . SET STD CLIMB/DIVE RATE         00263
        SLJ    (14C)        ZRO    (0)         . STORE RATE                      00264

* DATA DISPLAY ROUTINE

1D      EXF    (77311B)     LAC    (DFLAG)     . TURN OFF DISPALY LT             00265
        STA    (DFLAG)      SLJ    (4T)        . COMPLEMENT D-FLAG               00266

* ERROR MESSAGE ROUTINE

1E      SAL    (2E)         SAU    (6E)        .                                 00267
        EXF    (77240B)     ENI 5  (1)         . TURN ON ERROR LT                00270
2E      ENI 4  (7)          LDQ    (N)         . SET LETTER COUNT                00271
        ENA    (0)          ENI    (0)         . ZERO ACCUMLATOR                 00272
3E      LLS    (6)          QJP    (5E)        . ASSEMBLE ERROR MSG              00273
        IJP 4  (L-1)       ENI    (0)         . END WORD CHECK                  00274
        ISK 5  (4)          SLJ    (2E)        . CHECK WORD COUNT                00275
        SLJ    (5E1)        ZRO    (0)         .                                 00276
```

```
4E      LDA     (T+9)        INI 4 (-1)        .
        ALS     (6)          ADD   (20B)       . FILL OUT WORD WITH BLANKS
        IJP 4   (L-1)        STA 5 (ERRMSG)    . STORE WHEN FILLED
        ISK 5   (4)          SLJ   (4E1)       . 4-WORDS   NO JUMP TO 4E1
        SLJ     (5E1)        ZRO   (0)         . YES - JUMP TO 5E1
4E1     ENA     (0)          ENI 4 (8)         . SET INDEX + 1
        STA     (T+9)        SLJ   (4E)        .
5E      STA     (T+9)        ENI   (0)         . STORE MESSAGE
        SIL 4   (T+10)       LDA   (T+10)      . NATURAL END OF WORD
        AJP 1   (4E)         LDA   (T+9)       .   NO JUMP TO FILL OUT ROUTINE
        STA 5   (ERRMSG)     RAO   (2E)        .
        ISK 5   (4)          SLJ   (2E)        . 4-WORDS  NO JUMP TO 2E
5E1     ENA     (1)          STA   (KNR1)      . SET POINTER TO 1
        PRINT 1, ERRMSG
6E      ENA     (N)          STA   (T+8)       . DISPLAY MSG USING
        ENA     (XP1)        SAL   (PRINT+1)   .   PRINT ROUTINE
        ENA     (YP2)        SAL   (PRINT+2)   .
        LDA     (T+8)        LIL 5 (PRINT+4)   .
        SIL 5   (T+6)        LIL 5 (PRO)       .
        SIL 5   (PRINT+3)    ENI 5 (502)       .
        SIL 5   (PRINT+4)    ENQ   (0B)        .
        SLJ 4   (PRINT)      ZRO   (0)         .
        LIL 5   (T+6)        SIL 5 (PRINT+4)   . RESTORE ORIGINAL ADDRESS
        SLJ     (4T)         ZRO   (0)         . RETURN

* EXIT PROGRAM

1END    EXF     (77202B)     EXF   (77302B)    .   X X X X
        EXF     (77402B)     EXF   (77502B)    .   X
        EXF     (77204B)     EXF   (77210B)    .   X
        EXF     (77310B)     EXF   (77410B)    .   X X X X
        EXF     (77510B)     EXF   (77520B)    .         X
        EXF     (77540B)     EXF   (77440B)    .         X
        EXF     (77340B)     EXF   (77240B)    .   X X X X
        LDA     (COUNTER)    SLJ 4 (FLOAT)     . DUTY CYCLE CARD
        ENI     (0)          STA   (COUNTER)   . DUTY CYCLE CARD
```

```
      LDA    (TIME)       SLJ 4 (FLOAT)      . DUTY CYCLE CARD                 00340
      STA    (TIME)       ENI   (0)          . DUTY CYCLE CARD                 00341
      FDV    (PERIOD)     FMU   (5.)         . DUTY CYCLE CARD                 00342
      STA    (TEMP)       LDA   (COUNTER)    . DUTY CYCLE CARD                 00343
      FSB    (TEMP)       FMU   (797.2E-6).  DUTY CYCLE CARD                   00344
      FDV    (TIME)       STA   (TEMP)       . DUTY CYCLE CARD                 00345
      LDA    (1.0)        FSB   (TEMP)       . DUTY CYCLE CARD                 00346
      STA    (TEMP)       ENI   (0)          . DUTY CYCLE CARD                 00347
      PRINT 200, TEMP                                                          00350
      SLJ    (1RET)       ZRO   (0)          . EXIT MAIN PROGRAM               00351

* DISPLAY ROUTINE

1F    SIU 4 (10F1)        SIL 5 (10F1)       . SAVE INDEXES                    00352
      SIU 6 (11F)         SLJ 4 (1MER)       . ERASE                           00353
      ENA    (2)          SAL   (PRINT+4)    .                                 00354
      ENA    (S6C)        SAL   (PRINT+1)    . TASK                            00355
      SAL    (PRINT+2)    ENA   (1)          .      FORCE                      00356
      SAL    (PRINT+3)    ENQ   (-0B)        .            CENTER               00357
      ENA    (S11)        SLJ 4 (PRINT)      .                   SYMBOL        00360
      ENI 4  (16)         LDQ   (MASK13)     . SET SEARCH INDEX                00361
2F    LDA    (MASK14)     ENI 5 (0)          . MANCODE = 8                     00362
      MEQ 4  (IHEAD)      SLJ   (L+7)        .   NO JUMP                       00363
      LDA 4  (TRACK)      STA   (T+10)       .   YES  CHECK LAUNCH BIT         00364
      LDA    (MASK23)     LDQ   (MASK23)     .                                 00365
      MEQ    (T+10)       SLJ   (L+2)        . LAUNCH BIT NOT SET              00366
      LDQ    (MASK13)     SLJ   (2F)         .   DESTROY TRACK WORD            00367
      LDQ    (MASK13)     ENA   (52B)        .                                 00370
      SAL 4  (TRACK)      SLJ   (2F)         .                                 00371
      ENI 4  (16)         LDQ   (MASK40)     . SET INDEX FOR SEARCH            00372
      LDA    (MASK40)     ENI   (0)          . SET A FOR SEARCH                00373
      MEQ 4  (IHEAD)      SLJ   (L+8)        . CHECK FOR FADE BIT              00374
      LDQ 4  (TRACK)      LDL   (77B)        . SAVE TRACK SYMBOL               00375
      EQS    (52B)        SLJ   (L+2)        . PREVIOUS SYMBOL STORED          00376
      SLJ    (L+5)        ZRO   (0)          .    YES JUMP                     00377
      STA 4  (F)          LDQ   (MASK40)     .                                 00400
```

```
        LDA 4 (TRACK)      SCL   (MASK11)   . CLEAR OLD SYMBOL              00401
        ADD   (52B)        STA 4 (TRACK)    . SET SYMBOL TO NON ACTIVE      00402
        SLJ   (L-8)        ZRO   (0)        .                               00403
        LDA   (DFLAG)      AJP 3 (L+3)      . CHECK TUBE FOR                00404
        ENA   (0)          STA   (RAW)      .   DISPLAY OF RAW DATA         00405
        SLJ   (3F)         ZRO   (0)        .                               00406
        ENA   (1)          STA   (RAW)      .                               00407
3F      ENI 4 (1)          ENI   (0)        . RAW DATA DISPLAY              00410
3FF     ENQ   (77B)        LDL 4 (TRACK)    .                               00411
        SUB   (52B)        AJP   (7F)       .                               00412
        LDL 4 (TRACK)      ENI   (0)        .                               00413
        STA   (T+3)        LIL 5 (T+3)      .                               00414
        ENA 4 (XN)         STA   (XD)       . SET DISPLAY ROUTINE           00415
        ENA 4 (YN)         STA   (YD)       .   ADDRESS                     00416
        ENI   (0)          SLJ 4 (1F0)      . LOAD X AND Y                  00417
        LDA 4 (XN)         AJP 3 (4F)       . JUMP IF X NEG                 00420
        LDA 4 (YN)         AJP 3 (5F)       . JUMP IF X POS   Y NEG         00421
        ENA   (C360)       SLJ   (L+5)      . 000 - 090                     00422
4F      LDA 4 (YN)         AJP 3 (6F)       . JUMP IF X NEG   Y NEG         00423
        ENA   (C270)       SLJ   (L+3)      . 270 - 360                     00424
5F      ENA   (C090)       SLJ   (L+2)      . 090 - 180                     00425
6F      ENA   (C180)       ENI   (0)        . 180 - 270                     00426
        ENQ   (-0B)        LIL 5 (RAW)      .                               00427
        SIL 5 (PRINT+3)    SLJ 4 (PRINT)    . DISPLAY RAW DATA              00430
        ENA 4 (XS)         STA   (XD)       . SET DISPLAY ROUTINE           00431
        ENA 4 (YS)         STA   (YD)       .   ADDRESS                     00432
        SLJ 4 (1F0)        ZRO   (0)        .                               00433
        LIL 6 (PRINT+2)    SIL 6 (T+5)      .                               00434
        LIL 6 (PRINT+1)    SIL 6 (T+4)      .                               00435
        LDA   (PRO)        ENQ   (-0B)      .                               00436
        SAL   (PRINT+3)    LIL 5 (T+3)      .                               00437
        ENA 5 (S1)         SLJ 4 (PRINT)    . PRINT SYMBOLS                 00440
        LIL 6 (T+4)        SIL 6 (PRINT+1)  .                               00441
        LIL 6 (T+5)        SIL 6 (PRINT+2)  .                               00442
        LDA   (PRO)        SAL   (PRINT+3)  .                               00443
        ENA   (S11C)       ENQ   (0B)       .                               00444
```

```
       SLJ 4 (PRINT)     ZRO    (0)         . PRINT PERIODS
7F     ISK 4 (15)        SLJ    (3FF)       . LOOP
10F    LDA   (HIS1)      AJP 1 (13F)        . HIS1 TO BE PRINTED YES JUMP
12F    LDA   (HIS2)      AJP 1 (15F)        . HIS2 TO BE PRINTED YES JUMP
       SLJ   (10F1)      ZRO    (0)         . RETURN
13F    ENI 4 (N)         ENI    (0)         . DISPLAY HIS1
14F    ENA 4 (XHIS1)     STA    (XD)        . SET DISPLAY ROUTINE
       ENA 4 (YHIS1)     STA    (YD)        .   ADDRESS
       SLJ 4 (1F0)       ZRO    (0)         .
       LDA   (PRO)       SAL    (PRINT+3)   .
       ENA   (S11C)      ENQ    (0B)        .
       SLJ 4 (PRINT)     ZRO    (0)         .
14F1   IJP 4 (14F)       SLJ    (12F)       . LOOP
15F    ENI 4 (N)         ENI    (0)         . DISPLAY HIS2
16F    ENA 4 (XHIS2)     STA    (XD)        . SET DISPLAY ROUTINE
       ENA 4 (YHIS2)     STA    (YD)        .   ADDRESS
       SLJ 4 (1F0)       ZRO    (0)         .
       LDA   (PRO)       SAL    (PRINT+3)   .
       ENA   (S11C)      ENQ    (0B)        .
       SLJ 4 (PRINT)     ZRO    (0)         .
16F1   IJP 4 (16F)       SLJ    (10F1)      . LOOP
10F1   ENI 4 (N)         ENI 5 (N)          .
11F    ENI 6 (N)         SLJ    (4T)        . RETURN

* LOAD X AND Y ROUTINE

1F0    SLJ   (N)         LDA 7 (XD)         . EXIT/ENTRY
       FMU   (RHO)       SLJ 4 (FIX)        . SCALE X
       STA   (XD)        ENA    (XD)        .
       SAL   (2ABS)      SLJ 4 (1ABS)       .
       THS   (250)       SLJ    (7F)        . CHECK X FOR MAX
       ENA   (XD)        SAL    (PRINT+1)   .
       LDA 7 (YD)        FMU    (RHO)       . SCALE Y
       SLJ 4 (FIX)       ZRO    (0)         .
       STA   (YD)        ENA    (YD)        .
       SAL   (2ABS)      SLJ 4 (1ABS)       .
```

```
      THS   (250)       SLJ   (7F)       . CHECK Y FOR MAX                  00506
      ENA   (YD)        SAL   (PRINT+2)  .                                  00507
      SLJ   (1F0)       ZRO   (0)        . RETURN                           00510

* RADAR PROCESSER ROUTINE

1G    SIL 2 (8G)        SIU 3 (9G)       . SAVE INDEXES                     00511
      SIL 4 (9G)        SIU 5 (10G)      .   2,3,4,5                        00512
      LDA   (KTIME)     SUB   (KTIME+1)  . DETERMINE PERIOD                 00513
      STA   (IP)        SLJ 4 (FLOAT)    . FLOAT PERIOD                     00514
      STA   (PERIOD)    FDV   (3600.)    .                                  00515
      STA   (PERIOD2)   ENI 4 (1)        . SET TRACK INDEX                  00516
2G    ENQ   (77B)       LDL 4 (TRACK)    . PUT TRACK DESIGNATION IN  A      00517
      EQS   (F60)       SLJ   (3G)       .                                  00520
      SLJ   (11G)       ZRO   (0)        . NON DESIGNATED TRACK JUMP        00521
3G    ENA  (0)          STA   (J1)       . CLEAR J1                         00522
      SIL 4 (J1)        LDA 4 (XN)       .                                  00523
      FSB 4 (XP)        FMU   (BETA)     . VXS =  VXS(N-1)                  00524
      FDV   (PERIOD2)   FAD 4 (VXS)      .         + BETA*(XN-XP)/PERIOD2   00525
      STA 4 (VXS)       LDA 4 (XN)       .                                  00526
      FSB 4 (XP)        FMU   (ALPHA)    . XS  =  XP + ALPHA*(XN-XP)        00527
      FAD 4 (XP)        STA 4 (XS)       .                                  00530
      LDA 4 (VXS)       FMU   (PERIOD2)  . XP  =  XS + PERIOD2*VXS          00531
      FAD 4 (XS)        STA 4 (XP)       .                                  00532
      LDA 4 (YN)        FSB 4 (YP)       .                                  00533
      FMU   (BETA)      FDV   (PERIOD2)  . VYS =  VYS(N-1)                  00534
      FAD 4 (VYS)       STA 4 (VYS)      .         + BETA*(YN-YP)/PERIOD2   00535
      LDA 4 (YN)        FSB 4 (YP)       .                                  00536
      FMU   (ALPHA)     FAD 4 (YP)       . YS  =  YP + ALPHA*(YN-YP)        00537
      STA 4 (YS)        LDA 4 (VYS)      .                                  00540
      FMU   (PERIOD2)   FAD 4 (YS)       . YP  =  YS + PERIOD2*VYS          00541
      STA 4 (YP)        ENA 4 (VXS)      .                                  00542
      SAL   (COURSE+1)  ENA 4 (VYS)      . DETERMINE                        00543
      SAL   (COURSE+2)  ENA 4 (CRS)      .   COURSE IN DEGREES TRUE         00544
      SAL   (COURSE+3)  ENA 4 (SPD)      .     SPEED IN MACH NUMBER         00545
      SAL   (COURSE+6)  ENA 4 (MACH)     .                                  00546
```

```
        SAL     (COURSE+4)   ENA 4 (ZN)          .  M = SPD(KNTS)/(661-2.5*ALT)
        SAL     (COURSE+5)   SLJ 4 (COURSE)      .
        LDA 4 (ZN)           STA 4 (ZS)          . STORE Z DATA
        STA 4 (ZP)           LDA   (J1)          .
        EQS   (HIS1)         SLJ   (5G)          . HIS1 TO BE UPDATED  NO JUMP
        SSH   (PPMC1)        SLJ   (7G)          .
4G      ENI 2 (24)           INI 2 (-1)          . UPDATE HIS1
        LDA 2 (XHIS1)        STA 2 (XHIS1+1)     .
        LDA 2 (YHIS1)        STA 2 (YHIS1+1)     .
        IJP 2 (L-2)          LDA 4 (XN)          .
        STA   (YHIS1+1)      LDA 4 (YN)          .
        STA   (YHIS1+1)      SLJ   (7G)          .
5G      EQS   (HIS2)         SLJ   (7G)          . HIS2 TO BE UPDATED  NO JUMP
        SSH   (PPMC2)        SLJ   (7G)          .
6G      ENI 2 (24)           INI 2 (-1)          . UPDATE HIS2
        LDA 2 (XHIS2)        STA 2 (XHIS2+1)     .
        LDA 2 (YHIS2)        STA 2 (YHIS2+1)     .
        IJP 2 (L-2)          LDA 4 (XN)          .
        STA   (XHIS2+1)      LDA 4 (YN)          .
        STA   (YHIS2+1)      ENI   (0)           .
7G      ISK 4 (15)           SLJ   (2G)          . LOOP
7G1     LDA   (KTIME)        SLJ   (L+1)         .
        SUB   (IPP)          AJP 3 (L+9)         .
7G11    ENA   (60)           RAD   (IPP)         . INCREMENT IPP BY 60
        LIL 3 (PPI)          ENI 4 (14)          .
        LDA 4 (XS+1)         STA 3 (X1)          . STORE    X-SMOOTHED POINT
        LDA 4 (YS+1)         STA 3 (Y1)          .          Y-SMOOTHED POINT
        LDA 4 (XN+1)         STA 3 (X2)          .          X-ACTUAL   POINT
        LDA 4 (YN+1)         STA 3 (Y2)          .          Y-ACTUAL   POINT
        INI 3 (1)            IJP 4 (L-4)         . LOOP
        SIL 3 (PPI)          ENI   (0)           .
7G2     ENI 4 (3)            ENI   (0)           .
        LDA 4 (KTIME)        STA 4 (KTIME+1)     .
8G      IJP 4 (L-1)          ENI 2 (N)           . LOOP    RESTORE INDEXES
9G      ENI 3 (N)            ENI 4 (N)           . RESTORE INDEXES
10G     ENI 5 (N)            SLJ   (4T)          . RETURN
```

```
11G     LDA 4 (XN)        AJP 1 (L+2)     .
        LDA 4 (YN)        AJP   (7G)      .
        LDA 4 (F)         AJP   (L+4)     . CHECK FOR PREVIOUS TRACK FADE
        LDQ 4 (TRACK)     QRS   (6)       . CLEAR OLD SYMBOL
        QLS   (6)         ADL   (MASK10)  . RESTORE PREVIOUS SYMBOL
        STA 4 (TRACK)     SLJ   (14G)     .
        LDA 4 (ZN)        AJP   (12G)     . JUMP ACCORDING TO ALTITUDE
        AJP 3 (13G)       ENA   (-36B)    .  UNIDENTIFIED A/C
        RAD 4 (TRACK)     SLJ   (14G)     .
12G     ENA   (-42B)      SLJ   (L-1)     .  UNIDENTIFIED SURFACE
13G     ENA   (-46B)      SLJ   (L-2)     .  UNIDENTIFIED SUBMARINE
14G     LDA 4 (XN)        STA 4 (XS)      . SET XP AND XS = XN
        STA 4 (XP)        LDA 4 (YN)      . SET YP AND YS = YN
        STA 4 (YS)        STA 4 (YP)      .
        ENA   (0)         STA 4 (F)       . RESET F TO ZERO
        SLJ   (7G)        ZRO   (0)       .

* 10 MIN GRAPH ROUTINE

1GR     ENA   (2GR)       SAU   (3S)      . CHANGE ADDRESSES
        SAU   (8S)        EXF   (2000B)   . STOP THE REAL TIME CLOCK
        SLJ   (1S)        ZRO   (0)       . JUMP TO STOP AND CRIT ROUTINE
2GR     ENA   (1END)      SAU   (3S)      .
        SAU   (8S)        EXF   (01000B)  .
        REWIND MM
        WRITE TAPE MM, XN, YN, JTIME
        ENA   (E71)       SLJ 4 (1MSG)    . SEND GRAPH FINISHED MSG
        SLJ   (4T)        ZRO   (0)       . RETURN
3GR     EXF   (77503B)    SLJ   (1GR)     . TURN OFF GRAPH LT

* AIRCRAFT CONTROL ROUTINE

1INT    SIU 1 (1ISV)      SIL 2 (1ISV)    . SAVE
        SIU 3 (2ISV)      SIL 4 (2ISV)    .   INDEX
        SIU 5 (3ISV)      SIL 6 (3ISV)    .     REGISTERS
        ENI 3 (1)         ENI 5 (0)       . SET ARRAY COUNTERS
```

```
11I     ISK 5 (15)        SLJ   (L+2)     . CHECK FOR LOOP END                00651
        SLJ   (2INT)      ZRO   (0)       . JUMP IF TRACK1 ARRAY FINISHED     00652
        LDA 5 (TRACK1)    AJP   (2INT)    . JUMP IF NO FRIENDLY AIRCRAFT      00653
        ENQ   (0)         STQ   (ISFLAG)  . SET FLAGS TO ZERO                 00654
        STQ   (ICFLAG)    STQ   (IPFLAG)  .                                   00655
        STQ   (IEFLAG)    LLS   (6)       .                                   00656
        STQ   (ICNR)      QLS   (36)      . UNPACK CAP TRACK NUMBER           00657
        STQ   (KCNR)      LIL 1 (ICNR)    .                                   00660
        LDQ 1 (TRACK)     LDL   (MASK32)  .                                   00661
        ARS   (24)        STA   (JCNR)    . UNPACK CAP NUMBER                 00662
        LDL   (MASK23)    AJP 1 (1IL)     . JUMP TO LAUNCH CAP                00663
        LDL   (MASK30)    ARS   (39)      .                                   00664
        STA   (ITYPE)     LIL 4 (ITYPE)   . UNPACK A/C TYPE                   00665
        LDA 4 (FIG)       STA   (FLOW)    . UNPACK FUEL RATE                  00666
        FMU   (PERIOD)    SLJ 4 (FIX)     . COMPUTE FUEL CONSUMPTION          00667
        STA   (T+2)       LDQ 1 (KFIG)    . INCREMENT                         00670
        ENA   (0)         LLS   (15)      .   FUEL ON BOARD                   00671
        SUB   (T+2)       AJP 3 (1IE)     . JUMP IF FUEL EXHAUSTED            00672
        STA   (T+11)      LRS   (15)      .                                   00673
        STQ 1 (KFIG)      ENA 1 (XP)      . STORE NEW FUEL QUANTITY           00674
        SAL   (2ID)       ENA 1 (YP)      . COMPUTE                           00675
        SAL   (3ID)       ENA   (0)       .   RANGE                           00676
        STA   (T+2)       ENA   (T+2)     .     FROM                          00677
        SAL   (4ID)       SAL   (5ID)     .       CARRIER                     00700
        SLJ 4 (1ID)       ZRO   (0)       .                                   00701
        STA   (DME)       ENI   (0)       .                                   00702
        LDQ 4 (IFIGO)     LDL   (MASK31)  .                                   00703
        ARS   (21)        SLJ 4 (FLOAT)   .                                   00704
        STA   (RECOVER)   LDQ 4 (IFIGO)   .                                   00705
        LDL   (77700B)    ARS   (6)       .                                   00706
        STA   (ICRUISE)   SLJ 4 (FLOAT)   .                                   00707
        FDV   (3600.)     STA   (CRUISE)  . UNPACK CRUISE SPEED               00710
        LDQ 4 (IFIGO)     LDL   (MASK34)  .                                   00711
        ARS   (15)        STA   (LEVEL)   . UNPACK CRUISE ALTITUDE            00712
        LDL   (1B)        AJP 1 (L+3)     .                                   00713
        ENA   (1800)      STA   (INTSPD)  . MAX SPEED+IS 1800 KNOTS           00714
```

```
        SLJ    (L+2)       ZRO    (0)       .                                   00715
        ENA    (1200)      STA    (INTSPD)  . MAX SPEED IS 1200 KNOTS           00716
        SLJ 4  (FLOAT)     ZRO    (0)       .                                   00717
        STA    (SPDINT)    ENI    (0)       .                                   00720
        LDA    (T+11)      SLJ 4  (FLOAT)   .                                   00721
        STA    (FUEL)      LDA    (DME)     . COMPUTE                           00722
        FDV    (CRUISE)    FMU    (FLOW)    .   STEER                           00723
        FAD    (RECOVER)   FSB    (FUEL)    .     STATE                         00724
        AJP 3  (2I)        LDQ 1  (TRACK2)  . JUMP IF NOT LOW STATE             00725
        LDL    (MASK19)    AJP 1  (2IE1)    . JUMP IF MESSAGE ALREADY SENT      00726
        SLJ    (2IE)       ZRO    (0)       . SEND LOW STATE MESSAGE            00727
2I      LDQ 1  (TRACK)     LDL    (MASK19)  .                                   00730
        AJP 1  (1II)       ENI    (0)       . JUMP TO INTERCEPT                 00731
5I      LDL    (MASK20)    AJP 1  (1IS)     . JUMP TO STATION                   00732
        LDL    (MASK24)    AJP 1  (2IO)     . JUMP TO ORBIT                     00733
        LDL    (MASK22)    AJP 1  (1IR)     . JUMP TO STEER                     00734
3I      SLJ    (1I)        ZRO    (0)       . END OF MAIN LOOP                  00735

* EXIT FROM AIRCRAFT CONTROL ROUTINE

2INT    ENI 5  (1)         ENI 6  (1)       . REARRANGE                         00736
        LDA 6  (TRACK1)    AJP 1  (L+3)     .   TRACK1                          00737
        ISK 6  (15)        SLJ    (L-1)     .     ARRAY                         00740
        SLJ    (L+3)       ZRO    (0)       .                                   00741
        STA 5  (TRACK1)    INI 5  (1)       .                                   00742
        ISK 6  (15)        SLJ    (L-4)     .                                   00743
        SIL 5  (KNR2)      ENI    (0)       . SET TRACK1 POINTER                00744
1ISV    ENI 1  (*)         ENI 2  (*)       . RESTORE                           00745
2ISV    ENI 3  (*)         ENI 4  (*)       .   INDEX                           00746
3ISV    ENI 5  (*)         ENI 6  (*)       .     REGISTERS                     00747
        SLJ    (4T)        ZRO    (0)       . END AIRCRAFT CONTROL ROUTINE      00750

* COMPUTE SQUARE ROOT OF (A-C) SQUARED MINUS (B-D) SQUARED

1ID     SLJ    (*)         SLJ    (L+5)     .                                   00751
2ID     ZRO    (0)         ZRO    (0)       . A                                 00752
```

```
3ID    ZRO    (0)          ZRO    (0)        . B                                   00753
4ID    ZRO    (0)          ZRO    (0)        . C                                   00754
5ID    ZRO    (0)          ZRO    (0)        . D                                   00755
       LDA 7  (2ID)        FSB 7  (4ID)      . A-C                                 00756
       STA    (T+7)        FMU    (T+7)      . (A-C) SQUARED                       00757
       STA    (T+7)        LDA 7  (3ID)      .                                     00760
       FSB 7  (5ID)        STA    (T+8)      . B-D                                 00761
       FMU    (T+8)        FAD    (T+7)      . (B-D) SQUARED + (A-C) SQUARED       00762
       STA    (T+7)        ENA    (T+7)      .                                     00763
       SAL    (SQ+1)       SLJ 4  (SQ)       . TAKE SQUARE ROOT                    00764
       SLJ    (1ID)        ZRO    (0)        . RESULT IN A REGISTER                00765

* COIN FLIPPER

1IF    SLJ    (*)          LDQ    (Z)        . RETURN WITH A REGISTER              00766
       LDL    (1B)         AJP 1  (L-1)      .    + , HEADS                        00767
       ENA    (-1)         SLJ    (L-2)      .    - , TAILS                        00770

* DETERMINE SHORTEST DIRECTION OF TURN TO A GIVEN COURSE

1IT    SLJ    (*)          SLJ    (L+3)      .                                     00771
2IT    ZRO    (0)          ZRO    (0)        . ORDERED COURSE                      00772
3IT    ZRO    (0)          ZRO    (0)        . PRESENT COURSE                      00773
       LDA 7  (2IT)        FSB 7  (3IT)      . THE SIGN OF A REGISTER              00774
       SLJ 4  (FIX)        ZRO    (0)        .   DETERMINES DIRECTION OF TURN      00775
       AJP    (1IT)        ENI    (0)        .                                     00776
       STA    (T+7)        ENA    (T+7)      .                                     00777
       SAL    (2ABS)       SLJ 4  (1ABS)     .   +  STARBOARD TURN                 01000
       SUB    (180)        AJP 2  (L+2)      .   -  PORT TURN                      01001
       LDA    (T+7)        SLJ    (1IT)      .   0  O COURSE                       01002
       LAC    (T+7)        SLJ    (1IT)      .                                     01003

* SEND MESSAGE TO DISPLAY UNIT

1IE    ENA    (E29)        SLJ 4  (1OIE)     . MAYDAY MESSAGE                      01005
```

```
       LDA   (KCNR)     ADD   (LHEAD8)   .                                    01006
       STA 3 (LHEAD)    ENI   (0)        . SCRUB TRACK                        01007
       ENA   (0)        STA 5 (TRACK1)   . ERASE TRACK1 WORD                  01010
       STA 5 (TRACK2)   ENI   (0)        . ERASE TRACK2 WORD                  01011
       ENA   (52B)      STA 1 (TRACK)    . RESET TRACK WORD                   01012
       ISK 3 (60)       SLJ   (L+2)      .                                    01013
       ENI 3 (60)       ENI   (0)        .                                    01014
       SLJ   (1I)       ZRO   (0)        .                                    01015
2IE    ENA   (E26)      SLJ 4 (10IE)     . LOW STATE MESSAGE                  01016
       LDA 1 (TRACK2)   SST   (MASK19)   .                                    01017
       STA 1 (TRACK2)   ENI   (0)        .                                    01020
2IE1   LDQ 1 (TRACK)    LDL   (MASK19)   .                                    01021
       AJP 1 (1II)      LDA 1 (TRACK)    . JUMP TO INTERCEPT                  01022
       SST   (MASK22)   STA 1 (TRACK)    . SET STEER BIT                      01023
       SLJ   (1IR)      ZRO   (0)        . JUMP TO STEER ROUTINE              01024
3IE    ENA   (E33)      SLJ 4 (10IE)     . FAMISHED MESSAGE                   01025
       SLJ   (1IO)      ZRO   (0)        .                                    01026
4IE    ENA   (E16)      SLJ 4 (10IE)     . UNABLE TO INTERCEPT - FUEL MSG     01027
4I     LDA 1 (TRACK)    SST   (MASK21)   . SET SKIP BIT                       01030
       STA 1 (TRACK)    SLJ   (1II)                                           01031
5IE    ENA   (E21)      SLJ 4 (10IE)     . UNABLE TO INTERCEPT - AMMO MSG     01032
       SLJ   (4I)       ZRO   (0)        .                                    01033
6IE    ENA   (E36)      SLJ 4 (10IE)     . UNABLE TO RETURN MESSAGE           01034
       LDA 1 (TRACK2)   SST   (MASK23)   .                                    01035
       STA 1 (TRACK2)   SLJ   (3II)      .                                    01036
7IE    ENA   (E41)      SLJ 4 (10IE)     . POOR INTERCEPT MESSAGE             01037
       SLJ   (1I)       ZRO   (0)        .                                    01040
8IE    ENA   (E44)      SAL   (12IE)     . SPLASH MESSAGE                     01041
       ENA   (E46)      SLJ 4 (11IE)     .                                    01042
       SLJ   (4I)       ZRO   (0)        .                                    01043
9IE    ENA   (E47)      SAL   (12IE)     . HEADS UP MESSAGE                   01044
       ENA   (49)       SLJ 4 (11IE)     .                                    01045
       SLJ   (5IE)      ZRO   (0)        .                                    01046
10IE   SLJ   (*)        STA   (T+11)     .                                    01047
       LDA   (JCNR)     SLJ 4 (1NUM)     .                                    01050
       ALS   (6)        ADD   (R5)       .                                    01051
```

```
        STA 7 (T+11)     LDA   (T+11)    .                                  01052
        SLJ 4 (1MSG)     ZRO   (0)       .                                  01053
        SLJ   (10IE)     ZRO   (0)       .                                  01054
11IE    SLJ   (*)        STA   (T+11)    .                                  01055
        LDA   (JBNR)     SLJ 4 (1NUM)    .                                  01056
        ALS   (39)       SCL   (77B)     .                                  01057
12IE    STA 7 (T+11)     ENA   (*)       .                                  01060
        SLJ 4 (1MSG)     ZRO   (0)       .                                  01061
        SLJ   (11IE)     ZRO   (0)       .                                  01062

* SKIP INTERCEPT

1II     LDL   (MASK21)   AJP   (2II)     . JUMP TO INTERCEPT                01063
        LDA 1 (TRACK)    SCL   (MASK33)  . CLEAR INTERCEPT                  01064
        STA 1 (TRACK)    LDQ 1 (TRACK)   .   AND SKIP BIT                   01065

* INTERCEPT ROUTINE

2II     LDQ 1 (TRACK)    LDL   (MASK20)  . IF CAP HAS ASSIGNED STATION,     01066
        AJP   (L+3)      LDA 1 (TRACK)   .   SET ORBIT BIT AS A FLAG,       01067
        SST   (MASK24)   SCL   (MASK20)  .     CLEAR STATION BIT            01070
        STA 1 (TRACK)    ENI   (0)       .                                  01071
        LDQ 1 (KFIG)     LLS   (15)      .                                  01072
        ENA   (0)        LLS   (3)       .                                  01073
        STA   (SP)       ENA   (0)       . UNPACK NUMBER OF SPARROWS        01074
        LLS   (3)        STA   (SW)      . UNPACK NUMBER OF SIDEWINDERS     01075
        LDQ 5 (TRACK1)   LLS   (6)       .                                  01076
        ENA   (0)        LLS   (6)       .                                  01077
        STA   (IBNR)     LIL 2 (IBNR)    .                                  01100
        ALS   (36)       STA   (KBNR)    .                                  01101
        LDQ 2 (TRACK)    LDL   (MASK32)  .                                  01102
        ARS   (24)       STA   (JBNR)    . UNPACK BOGIE NUMBER              01103
        LDA 2 (XP)       FSB 1 (XP)      . COMPUTE                          01104
        STA   (XDIS)     LDA 2 (YP)      .   RANGE OF                       01105
        FSB 1 (YP)       STA   (YDIS)    .     BOGIE                        01106
        FMU   (YDIS)     STA   (T+2)     .       FROM                       01107
```

```
        LDA   (XDIS)        FMU   (XDIS)      .        CAP                  01110
        FAD   (T+2)         STA   (T+2)       .         .                   01111
        ENA   (T+2)         SLJ 4 (SQ)        .           .                 01112
        STA   (BC)          ENI   (0)         .             .               01113
        ENA   (XDIS)        SAL   (COURSE+1). COMPUTE                       01114
        ENA   (YDIS)        SAL   (COURSE+2).   BEARING                     01115
        ENA   (CAPB)        SAL   (COURSE+3).     OF                        01116
        ENA   (0)           STA   (T+2)       .     BOGIE                   01117
        ENA   (T+2)         SAL   (COURSE+4).         FROM                  01120
        SAL   (COURSE+5)    SLJ 4 (COURSE)    .         CAP                 01121
        SLJ   (7I)          ZRO   (0)         .                             01122

* DOES CAP HAVE ENOUGH FUEL FOR INTERCEPT

3II     SLJ   (*)           ENI   (0)         .                             01123
        LDA   (YBI)         FMU   (YBI)       . COMPUTE AMOUNT              01124
        STA   (T+2)         LDA   (XBI)       .   OF FUEL NEEDED            01125
        FMU   (XBI)         FAD   (T+2)       .     TO TRAVEL FROM          01126
        STA   (T+2)         ENA   (T+2)       .       PRESENT POSITION      01127
        SAL   (SQ+1)        SLJ 4 (SQ)        .   TO COLLISION INTERCEPT    01130
        STA   (T+2)         ENA   (XBI)       .     POINT AND THEN          01131
        SAL   (2ID)         ENA   (YBI)       .       TO CARRIER TO         01132
        SAL   (3ID)         ENA 1 (XP)        .   ARRIVE WITH 600 POUNDS    01133
        SAL   (4ID)         ENA 1 (YP)        .     OF FUEL                 01134
        SAL   (5ID)         SLJ 4 (1ID)       .                             01135
        STA   (T+3)         FAD   (T+2)       .                             01136
        FDV   (CRUISE)      FMU   (FLOW)      .                             01137
        FAD   (600.)        FSB   (FUEL)      .                             01140
        AJP 2 (L+1)         SLJ   (3II)       . A JUMP IF SHORT OF FUEL     01141
        LDA   (T+3)         FDV   (CRUISE)    . COMPUTE AMOUNT OF FUEL      01142
        FMU   (FLOW)        FSB   (FUEL)      .   TO REACH BOGIE            01143
        AJP 2 (4IE)         LDQ 1 (TRACK2)    . JUMP IF UNABLE TO INTERCEPT 01144
        LDL   (MASK23)      AJP 1 (3II)       .                             01145
        SLJ   (6IE)         ZRO   (0)         . JUMP IF UNABLE TO RETURN    01146
7I      LDQ 1 (TRACK)       LDL   (MASK25)    .                             01147
        AJP 1 (1IC)         ENA   (1)         . JUMP IF COLLISION INTERCEPT 01150
```

```
      STA   (ICFLAG)    LDL   (MASK26)   . SET INITIAL COLLISION FLAG      01151
      AJP 1 (1IP)       ENA   (1)        .                                 01152
      STA   (IPFLAG)    LDA   (SP)       . SET INITIAL PURSUIT FLAG        01153
      AJP 1 (1IC)       ENI   (0)        . JUMP IF SPARROWS ON BOARD       01154
9I    ENA   (1)         STA   (ISFLAG)   .                                 01155
      SLJ   (6I)        ZRO   (0)        . JUMP IF NO SPARROWS             01156

* COLLISION INTERCEPT

1IC   LDA   (BC)        FSB   (10.)      .                                 01157
      AJP 2 (L+5)       ENA   (1)        . JUMP IF OUTSIDE KILL RANGE      01160
      STA   (IEFLAG)    SLJ 4 (1IF)      . SET ENGAGED FLAG                01161
      AJP 2 (L+2)       RSO   (SP)       . MISS WITH SPARROW               01162
      AJP   (9I)        ENI   (0)        . JUMP IF NO SPARROWS             01163
      RSO   (SP)        SLJ   (1IK)      . HIT WITH SPARROW                01164
      LDA 2 (CRS)       FSB   (CAPB)     .                                 01165
      STA   (T+2)       ENA   (T+2)      .                                 01166
      SAL   (2ABS)      SLJ 4 (1ABS)     .                                 01167
      FMU   (.0174533)  STA   (ALFA)     .                                 01170
      ENA   (ALFA)      ENI   (0)        .                                 01171
      SAL   (COSF+1)    SLJ 4 (COSF)     .                                 01172
      FMU   (BC)        STA   (BP)       .                                 01173
      ENA   (ALFA)      SAL   (SINF+1)   .                                 01174
      SLJ 4 (SINF)      ZRO   (0)        .                                 01175
      FMU   (BC)        STA   (CP)       .                                 01176
      LDA 1 (SPD)       FDV 2 (SPD)      .                                 01177
      STA   (T+2)       FMU   (T+2)      .                                 01200
      FSB   (1.)        STA   (T+2)      .                                 01201
      LDA   (CP)        FDV   (BP)       .                                 01202
      STA   (T+3)       FMU   (T+3)      .                                 01203
      FAD   (1.)        FMU   (T+2)      .                                 01204
      FAD   (1.)        STA   (ARG)      .                                 01205
      AJP 2 (L+4)       LDA   (SPDINT)   . CHECK IF CAP ACCELERATING       01206
      FSB   (10.)       FSB 1 (SPD)      .                                 01207
      AJP 2 (L+3)       LDA 1 (TRACK)    . JUMP IF UNABLE TO INTERCEPT     01210
      SCL   (MASK25)    STA 1 (TRACK)    . CLEAR COLLISION BIT             01211
```

```
SLJ    (6I)       ZRO    (0)         . JUMP IF UNABLE TO INTERCEPT      01212
LDA 1  (TRACK)    SST    (MASK25)    . SET COLLISION BIT                01213
STA 1  (TRACK)    SLJ    (1IP)       . TEMPORARY PURSUIT INTERCEPT      01214
LDA 1  (SPD)      FMU 1  (SPD)       .                                  01215
STA    (T+2)      LDA 2  (SPD)       .                                  01216
FMU 2  (SPD)      FAD    (T+2)       .                                  01217
STA    (T+2)      LDA 2  (SPD)       .                                  01220
FMU 2  (SPD)      FMU    (BP)        .                                  01221
FDV    (T+2)      STA    (A)         .                                  01222
ENA    (ARG)      SAL    (SQ+1)      .                                  01223
SLJ 4  (SQ)       ZRO    (0)         .                                  01224
STA    (SQARG)    LDA    (-1.)       .                                  01225
FSB    (SQARG)    STA    (B1)        .                                  01226
FMU    (A)        STA    (B3)        .                                  01227
LDA    (-1.)      FAD    (SQARG)     .                                  01230
STA    (B2)       FMU    (A)         .                                  01231
STA    (B4)       LDA    (B3)        .                                  01232
AJP    (L+9)      AJP 2  (L+9)       .                                  01233
LDA    (B4)       AJP 3  (L+2)       .                                  01234
SLJ    (L+5)      ZRO    (0)         .                                  01235
LDA    (SPDINT)   FSB    (10.)       .                                  01236
FSB 1  (SPD)      AJP 2  (L+3)       . JUMP IF UNABLE TO INTERCEPT      01237
LDA 1  (TRACK)    SCL    (MASK25)    . CLEAR COLLISION BIT              01240
STA 1  (TRACK)    SLJ    (6I)        . JUMP IF UNABLE TO INTERCEPT      01241
LDA 1  (TRACK)    SST    (MASK25)    . SET COLLISION BIT                01242
STA 1  (TRACK)    SLJ    (1IP)       . TEMPORARY PURSUIT INTERCEPT      01243
LDA    (B4)       STA    (BI)        . BI = B4                          01244
SLJ    (L+8)      ZRO    (0)         .                                  01245
LDA    (B4)       AJP    (L+4)       .                                  01246
AJP 2  (L+3)      ENI    (0)         .                                  01247
LDA    (B3)       STA    (BI)        . BI = B3                          01250
SLJ    (L+4)      ZRO    (0)         .                                  01251
LDA    (B3)       FSB    (B4)        .                                  01252
AJP    (L-3)      AJP 3  (L-3)       .                                  01253
AJP 2  (L-8)      ZRO    (0)         .                                  01254
LDA 2  (CRS)      FMU    (.0174533). CONVERT COURSE                     01255
```

```
STA    (BCRS)       ENI    (0)          .   TO RADIANS               01256
LDA    (BCRS)       FSB    (4.7124)     .                            01257
AJP 2  (L+2)        LDA    (BCRS)       .                            01260
FSB    (1.5708)     AJP 2  (L+5)        .                            01261
ENA    (BCRS)       SAL    (COSF+1)     .                            01262
SLJ 4  (COSF)       ZRO    (0)          .                            01263
FMU    (BI)         FAD 2  (XP)         .                            01264
STA    (XBI)        SLJ    (L+6)        .                            01265
ENA    (BCRS)       SAL    (COSF+1)     .                            01266
SLJ 4  (COSF)       ZRO    (0)          .                            01267
FMU    (BI)         STA    (T+2)        .                            01270
LDA 2  (XP)         FSB    (T+2)        .                            01271
STA    (XBI)        ENI    (0)          .                            01272
ENA    (BCRS)       SAL    (SINF+1)     .                            01273
SLJ 4  (SINF)       ZRO    (0)          .                            01274
FMU    (BI)         FAD 2  (YP)         .                            01275
STA    (YBI)        LDA    (ICFLAG)     .                            01276
AJP    (L+1)        SLJ 4  (3II)        . CHECK FOR ENOUGH FUEL      01277
ENA    (XBI)        SAL    (COURSE+1).  COMPUTE                     01300
ENA    (YBI)        SAL    (COURSE+2).    ORDERED                   01301
ENA    (CCRS)       SAL    (COURSE+3).      COURSE                  01302
ENA    (0)          STA    (T+2)        .         .                  01303
ENA    (T+2)        SAL    (COURSE+4).              .               01304
SAL    (COURSE+5)   SLJ 4  (COURSE)     .             .              01305
ENA    (CCRS)       SAL    (2IT)        . DETERMINE BEST             01306
LDQ 1  (IHEAD)      LDL    (7777B)      .   DIRECTION TO TURN        01307
ARS    (12)         ENQ    (0)          .                            01310
DVI    (10)         STA    (T+3)        .                            01311
ENA    (T+3)        SAL    (3IT)        .                            01312
SLJ 4  (1IT)        ZRO    (0)          .                            01313
AJP    (L+8)        AJP 3  (L+2)        . ON COURSE                  01314
LDA    (LHEAD2)     SLJ    (L+2)        . STARBOARD TURN             01315
LDA    (LHEAD6)     ENI    (0)          . PORT TURN                  01316
STA    (T+2)        LDA    (CCRS)       .                            01317
SLJ 4  (FIX)        ZRO    (0)          .                            01320
ALS    (18)         ADD    (KCNR)       .                            01321
```

```
          ADD    (T+2)         STA 3 (LHEAD)     . ORDER TURN                        01322
          ISK 3  (60)          SLJ   (L+2)       .                                   01323
          ENI 3  (60)          ENI   (0)         .                                   01324
          LDA 2  (ZP)          FSB   (2500.)     . SET CAP ALTITUDE                  01325
          AJP 2  (L+2)         LDA   (1750B)     .   TO BOGIE ALTITUDE               01326
          STA    (T+2)         SLJ   (L+3)       .     LESS 2000 FEET OR             01327
          LDA 2  (ZP)          FSB   (2000.)     .       TO 1000 FEET IF             01330
          STA    (T+2)         SLJ 4 (FIX)       .         BOGIE IS BELOW            01331
          ALS    (18)          ADD   (LHEAD3)    .           2500 FEET               01332
          ADD    (KCNR)        STA 3 (LHEAD)     .                                   01333
          ISK 3  (60)          SLJ   (L+2)       .                                   01334
          ENI 3  (60)          ENI   (0)         .                                   01335
          LDQ 1  (IHEAD)       LDL   (7777B)     . UNPACK CAP SPEED                  01336
          STA    (T+2)         SUB   (INTSPD)    .                                   01337
          AJP    (L+5)         LDA   (LHEAD4)    . JUMP IF INTERCEPTOR AT SPEED      01340
          ADD    (KCNR)        STA   (T+3)       .                                   01341
          LDA    (T+2)         ALS   (18)        .                                   01342
          ADD    (T+3)         STA 3 (LHEAD)     . ORDER SPEED                       01343
          ISK 3  (60)          SLJ   (L+2)       .                                   01344
          ENI 3  (60)          ENI   (0)         .                                   01345
          LDA 1  (TRACK)       SST   (MASK25)    . SET COLLISION BIT                 01346
          STA 1  (TRACK)       SLJ   (2I)        .                                   01347
6I        LDA    (SW)          AJP 1 (1IP)       . JUMP IF SIDEWINDERS ON BOARD      01350
          LDA    (ISFLAG)      AJP   (1IP)       . JUMP IF AMMO NOT EXHAUSTED        01351
          LDA    (IEFLAG)      AJP 1 (9IE)       . JUMP IF CAP ENGAGED, NO AMMO      01352
          AJP    (5IE)         ENI   (0)         . JUMP IF AMMO EXHAUSTED            01353

* PURSUIT INTERCEPT

1IP       LDA    (ICFLAG)      AJP   (L+4)       . CHECK FOR ENOUGH                  01354
          LDA 2  (XP)          STA   (XBI)       .   FUEL FOR                        01355
          LDA 2  (YP)          STA   (YBI)       .     INTERCEPT                     01356
          SLJ 4  (3II)         ZRO   (0)         .                                   01357
          LDA    (BC)          FSB   (2.5)       .                                   01360
          AJP 2  (2IP)         SLJ 4 (1IF)       . JUMP IF OUTSIDE KILL RANGE        01361
          AJP 3  (L+4)         LDA   (SW)        . JUMP IF MISS                      01362
```

```
      AJP    (L+2)      RSO    (SW)      . HIT WITH SIDEWINDER                01363
      SLJ    (1IK)      ZRO    (0)       .                                    01364
      RSO    (SP)       SLJ    (1IK)     . HIT WITH SPARROW                   01365
      LDA    (SW)       AJP    (L+3)     . JUMP IF NO SIDEWINDERS             01366
      RSO    (SW)       AJP    (L+5)     . JUMP IF NO SECOND SIDEWINDER       01367
      RSO    (SW)       SLJ    (1IK)     . HIT WITH SECOND SIDEWINDER         01370
      LDA    (ISFLAG)   AJP 1  (9IE)     . JUMP IF NO SPARROWS                01371
      RSO    (SP)       AJP    (9IE)     . MISS WITH SPARROW                  01372
      RSO    (SP)       SLJ    (1IK)     . HIT WITH SPARROW                   01373
      LDA    (ISFLAG)   AJP 1  (9IE)     . JUMP IF NO SPARROWS                01374
      RSO    (SP)       SLJ    (1IK)     . HIT WITH SPARROW                   01375
2IP   ENA    (CAPB)     SAL    (2IT)     . DETERMINE BEST                     01376
      LDQ 1  (IHEAD)    LDL    (7777B)   .   DIRECTION TO TURN                01377
      ARS    (12)       ENQ    (0)       .                                    01400
      DVI    (10)       STA    (T+3)     .                                    01401
      ENA    (T+3)      SAL    (3IT)     .                                    01402
      SLJ 4  (1IT)      ZRO    (0)       .                                    01403
      AJP    (L+8)      AJP 3  (L+2)     . ON COURSE                          01404
      LDA    (LHEAD2)   SLJ    (L+2)     . STARBOARD TURN                     01405
      LDA    (LHEAD6)   ENI    (0)       . PORT TURN                          01406
      STA    (T+2)      LDA    (CAPB)    .                                    01407
      SLJ 4  (FIX)      ZRO    (0)       .                                    01410
      ALS    (18)       ADD    (KCNR)    .                                    01411
      ADD    (T+2)      STA 3  (LHEAD)   . ORDER TURN TO COURSE               01412
      ISK 3  (60)       SLJ    (L+2)     .                                    01413
      ENI 3  (60)       ENI    (0)       .                                    01414
      LDA 2  (ZP)       SLJ 4  (FIX)     .                                    01415
      ALS    (18)       ADD    (KCNR)    .                                    01416
      ADD    (LHEAD3)   STA 3  (LHEAD)   . ORDER ALTITUDE EQUAL BOGIE ALT     01417
      ISK 3  (60)       SLJ    (L+2)     .                                    01420
      ENI 3  (60)       ENI    (0)       .                                    01421
      ENI    (0)        LDA 2  (SPD)     .                                    01422
      FMU    (1.15)     STA    (T+2)     . ORDER SPEED EQUAL                  01423
      FSB    (SPDINT)   AJP 3  (L+2)     .   TO BOGIE SPEED                   01424
      LDA    (INTSPD)   SLJ    (L+2)     .     PLUS 15 PERCENT                01425
      LDA    (T+2)      SLJ 4  (FIX)     .                                    01426
```

```
         ALS    (18)      ADD    (KCNR)    .                                01427
         ADD    (LHEAD4)  STA 3 (LHEAD)    .                                01430
         ISK 3 (60)       SLJ    (L+2)     .                                01431
         ENI 3 (60)       ENI    (0)       .                                01432
         LDQ 1 (TRACK)    LDL    (MASK26)  .                                01433
         AJP 1 (1I)       LDA 1 (TRACK)    . JUMP IF MESSAGE ALREADY SENT   01434
         SST    (MASK26)  STA 1 (TRACK)    .                                01435
         SLJ    (7IE)     ZRO    (0)       .                                01436

* INTERCEPTOR KILL, SPLASH BOGIE

1IK      LDA    (LHEAD8)  ADD    (KBNR)    . SPLASH BOGIE                   01437
         STA 3 (LHEAD)    ENI    (0)       .                                01440
         ENA    (0)       STA 5 (TRACK1)   .                                01441
         STA 5 (TRACK2)   ENI    (0)       . ERASE TRACK2 WORD              01442
         ENA    (52B)     STA 2 (TRACK)    . RESET TRACK WORD               01443
         ISK 3 (60)       SLJ    (L+2)     .                                01444
         ENI 3 (60)       ENI    (0)       .                                01445
         SLJ    (8IE)     ZRO    (0)       .                                01446

* LAUNCH CAP FROM CARRIER

1IL      LDA 1 (TRACK)    SCL    (MASK23)  . RESET                          01447
         STA 1 (TRACK)    LDQ 1 (TRACK)    .   LAUNCH BIT                   01450
         LDA    (KCNR)    ADD    (LHEAD1)  . INSERT CAP NUMBER              01451
         STA 3 (LHEAD)    ENI    (0)       .   INTO LAUNCH                  01452
         ISK 3 (60)       SLJ    (L+2)     .     COMMAND WORD               01453
         ENI 3 (60)       ENI    (0)       .                                01454
         SLJ    (2I)      ZRO    (0)       .     COMMAND WORD               01455

* ORBIT ROUTINE

1IO      LDA 1 (TRACK)    SST    (MASK24)  . SET ORBIT BIT                  01456
         ENA    (52B)     STA 1 (TRACK)    . RESET TRACK WORD               01457
         STA 1 (TRACK)    ENI    (0)       .                                01460
         LDA    (ICRUISE) ALS    (18)      .                                01461
```

```
       ADD    (LHEAD4)    ADD    (KCNR)    .                           01462
       STA 3  (LHEAD)     ENI    (0)       . ORDER CRUISE SPEED        01463
       ISK 3  (60)        SLJ    (L+2)     .                           01464
       ENI 3  (60)        ENI    (0)       .                           01465
       LDA    (LEVEL)     ALS    (18)      .                           01466
       ADD    (LHEAD3)    ADD    (KCNR)    .                           01467
       STA 3  (LHEAD)     ENI    (0)       . ORDER CRUISE ALTITUDE     01470
       ISK 3  (60)        SLJ    (L+2)     .                           01471
       ENI 3  (60)        ENI    (0)       .                           01472
2IO    LDA    (LHEAD5)    ADD    (KCNR)    . ORDER ORBIT               01473
       STA 3  (LHEAD)     ENI    (0)       .                           01474
       ISK 3  (60)        SLJ    (L+2)     .                           01475
       ENI 3  (60)        ENI    (0)       .                           01476
       SLJ    (1I)        ZRO    (0)       .                           01477

* STEER ROUTINE

1IR    LDA    (DME)       FSB    (1.)      .                           01500
       AJP 2  (L+8)       LDA    (LHEAD8)  . JUMP IF NOT LANDED YET    01501
       ADD    (KCNR)      STA 3  (LHEAD)   . CAP LANDED,               01502
       ENA    (0)         STA 5  (TRACK1)  .                           01503
       STA 5  (TRACK2)    ENI    (0)       . ERASE TRACK2 WORD         01504
       ISK 3  (60)        SLJ    (L+2)     .   SCRUB TRACK             01505
       ENI 3  (60)        ENI    (0)       .                           01506
       SLJ    (2I)        ZRO    (0)       .   SCRUB TRACK             01507
       LDQ 1  (TRACK)     LDL    (MASK27)  .                           01510
       AJP 1  (8I)        ENI    (0)       . JUMP IF ALREADY STEERED   01511
       LDA 1  (ZP)        FSB    (80.)     . COMPUTE RANGE             01512
       FMU    (.006)      FMU    (CRUISE)  .   AT WHICH TO             01513
       FSB    (DME)       AJP 3  (L+3)     .     BEGIN LET DOWN        01514
       LDA    (LHEAD7)    ENQ    (1)       . ORDER LET DOWN            01515
       STQ    (T+4)       SLJ-   (L+3)     .                           01516
       LDA    (LEVEL)     ALS    (18)      .                           01517
       ADD    (LHEAD3)    ENI    (0)       . ORDER CRUISE ALTITUDE     01520
       ADD    (KCNR)      STA 3  (LHEAD)   .                           01521
       ISK 3  (60)        SLJ    (L+2)     .                           01522
```

```
ENI 3 (60)       ENI   (0)          .                           01523
ENI   (0)        LDA 1 (SPD)        . COMPUTE RANGE             01524
FSB   (120.)     STA   (T+2)        .   AT WHICH TO             01525
FDV   (20.)      FMU   (T+2)        .     BEGIN DECELERATION    01526
FSB   (DME)      AJP 3 (L+5)        .                           01527
LDQ   (T+4)      QJP   (L+3)        .                           01530
LDA 1 (TRACK)    SST   (MASK27)     . SET STEERED BIT           01531
STA 1 (TRACK)    ENI   (0)          .                           01532
LDA   (LHEAD9)   SLJ   (L+3)        . ORDER DECELERATION        01533
LDA   (ICRUISE)  ALS   (18)         .                           01534
ADD   (LHEAD4)   ENI   (0)          . ORDER CRUISE SPEED        01535
ADD   (KCNR)     STA 3 (LHEAD)      .                           01536
ISK 3 (60)       SLJ   (L+2)        .                           01537
ENI 3 (60)       ENI   (0)          .                           01540
ENA 1 (XP)       SAL   (COURSE+1)   .                           01541
ENA 1 (YP)       SAL   (COURSE+2)   .                           01542
ENA   (T+2)      SAL   (COURSE+3)   .                           01543
ENA   (0)        STA   (T+3)        .                           01544
ENA   (T+3)      SAL   (COURSE+4)   .                           01545
SAL   (COURSE+5) SLJ 4 (COURSE)     .                           01546
ENA   (T+2)      SAL   (2IT)        . DETERMINE BEST            01547
LDQ 1 (IHEAD)    LDL   (7777B)      .   DIRECTION TO TURN       01550
ARS   (12)       ENQ   (0)          .                           01551
DVI   (10)       STA   (T+3)        .                           01552
ENA   (T+3)      SAL   (3IT)        .                           01553
SLJ 4 (1IT)      ZRO   (0)          .                           01554
AJP   (2I)       AJP 3 (L+2)        . ON COURSE                 01555
LDA   (LHEAD2)   SLJ   (L+2)        . STARBOARD TURN            01556
LDA   (LHEAD6)   ENI   (0)          . PORT TURN                 01557
STA   (T+3)      LDA   (T+2)        .                           01560
SLJ 4 (FIX)      ZRO   (0)          .                           01561
ALS   (18)       ADD   (KCNR)       .                           01562
ADD   (T+3)      STA 3 (LHEAD)      . ORDER TURN                01563
ISK 3 (60)       SLJ   (L+2)        .                           01564
ENI 3 (60)       ENI   (0)          .                           01565
SLJ   (2I)       ZRO   (0)          .                           01566
```

```
* STATION ROUTINE

1IS    LDQ 5 (TRACK1)    LLS   (21)      .                                01567
       QRS   (39)        STQ   (IYSTA)   . UNPACK STATION Y POSITION      01570
       ENQ   (0)         LRS   (9)       .                                01571
       QRS   (39)        STQ   (IXSTA)   . UNPACK STATION X POSITION      01572
       LDA   (IXSTA)     SLJ 4 (FLOAT)   .                                01573
       STA   (XSTA)      ENI   (0)       .                                01574
       LDA   (IYSTA)     SLJ 4 (FLOAT)   .                                01575
       STA   (YSTA)      LDA 1 (YP)      . IS                             01576
       FSB   (YSTA)      STA   (T+2)     .   CAP                          01577
       ENA   (T+2)       SAL   (2ABS)    .     ON                         01600
       SLJ 4 (1ABS)      ZRO   (0)       .       STATION                  01601
       ENA   (6)         STA   (T+3)     .          .                     01602
       THS   (T+3)       SLJ   (L+6)     .            .                   01603
       LDA 1 (XP)        FSB   (XSTA)    .              .                 01604
       STA   (T+2)       ENA   (T+2)     .                .               01605
       SAL   (2ABS)      SLJ 4 (1ABS)    .                  .             01606
       THS   (T+3)       SLJ   (L+3)     . JUMP IF NOT ON STATION         01607
       LDA 1 (TRACK)     SCL   (MASK20)  . CLEAR STATION BIT              01610
       STA 1 (TRACK)     SLJ   (1IO)     . ON STATION                     01611
       LDA   (KCNR)      ADD   (LHEAD4)  . NOT ON STATION                 01612
       STA   (T+2)       LDA   (ICRUISE) .                                01613
       ALS   (18)        ADD   (T+2)     .                                01614
       STA 3 (LHEAD)     ENI   (0)       . ORDER CRUISE SPEED             01615
       ISK 3 (60)        SLJ   (L+2)     .                                01616
       ENI 3 (60)        ENI   (0)       .                                01617
       LDA   (KCNR)      ADD   (LHEAD3)  .                                01620
       STA   (T+2)       LDA   (LEVEL)   .                                01621
       ALS   (18)        ADD   (T+2)     .                                01622
       STA 3 (LHEAD)     ENI   (0)       . ORDER CRUISE ALTITUDE          01623
       ISK 3 (60)        SLJ   (L+2)     .                                01624
       ENI 3 (60)        ENI   (0)       .                                01625
       LDA 1 (XP)        FSB   (XSTA)    .                                01626
       STA   (T+2)       LDA 1 (YP)      .                                01627
```

```
        FSB     (YSTA)          STA     (T+3)         .                                01630
        ENA     (T+2)           SAL     (COURSE+1).   COMPUTE                          01631
        ENA     (T+3)           SAL     (COURSE+2).     COURSE                         01632
        ENA     (T+4)           SAL     (COURSE+3).       TO                           01633
        ENA     (0)             STA     (T+5)         .     STATION                    01634
        ENA     (T+5)           SAL     (COURSE+4).             .                      01635
        SAL     (COURSE+5)      SLJ 4   (COURSE)      .           .                    01636
        LDA     (T+4)           SLJ 4   (FIX)         .                                01637
        ALS     (18)            STA     (T+4)         . COURSE TO STATION IN T+4       01640
        LDQ 1   (IHEAD)         LDL     (7777B)       .                                01641
        ARS     (12)            ENQ     (0)           .                                01642
        DVI     (10)            STA     (T+3)         .                                01643
        ENA     (T+3)           SAL     (3IT)         .                                01644
        ENA     (T+4)           SAL     (2IT)         .                                01645
        SLJ 4   (1IT)           ZRO     (0)           . DETERMINE WHICH WAY TO TURN    01646
        AJP     (1I)            AJP 2   (L+2)         . ON COURSE                      01647
        LDA     (LHEAD6)        SLJ     (L+2)         . PORT TURN                      01650
        LDA     (LHEAD2)        ENI     (0)           . STARBOARD TURN                 01651
        STA     (T+2)           LDA     (KCNR)        .                                01652
        RAD     (T+2)           LDA     (T+4)         .                                01653
        ALS     (18)            ADD     (T+2)         . ORDER TURN TO STATION          01654
        STA 3   (LHEAD)         ENI     (0)           .                                01655
        ISK 3   (60)            SLJ     (L+2)         .                                01656
        ENI 3   (60)            ENI     (0)           .                                01657
        SLJ     (1I)            ZRO     (0)           .                                01660

*DESIGNATE TRACK ROUTINE

1J      EXF     (77411B)        SLJ     (L+3)         . TURN OFF DESIG LT              01661
2J      ZRO     (0)             ZRO     (0)           . HOSTILE/FRIENDLY               01662
3J      ZRO     (0)             ZRO     (0)           . TRACK NUMBER                   01663
        LDA     (2J)            LLS     (6)           . REPOSITION ARGUMENT            01664
        LDL     (77B)           STA     (2J)          .                                01665
        ENA     (3J)            SLJ 4   (3J0)         . HOOK TRACK                     01666
        LDA     (2J)            AJP     (19J)         . SENSE OWN SHIP                 01667
        INA     (-66B)          AJP     (4J)          . SENSE FRIENDLY                 01670
```

```
         INA   (-2B)      AJP   (9J)       . SENSE HOSTILE              01671
         ENA   (E6)       SLJ   (1E)       . ERROR EXIT                 01672
4J       ENA   (1)        STA   (FRIEND)   . SET FRIEND                 01673
5J       ENA   (ZN)       ADD   (3J)       . FRIEND = 1 / FRIENDLY      01674
         STA   (T+6)      LDQ 7 (T+6)      . FRIEND =-1 / HOSTILE       01675
         ENA   (TRACK)    ADD   (3J)       .                            01676
         STA   (T+6)      QJP 3 (6J)       . JUMP ACCORDING TO          01677
         QJP   (7J)       QJP 2 (8J)       .    ALTITUDE                01700
6J       LDA   (FRIEND)   AJP 2 (L+3)      .                            01701
         ENA   (40B)      STA 7 (T+6)      . HOSTILE SUB                01702
         SLJ   (12J)      ZRO   (0)        .                            01703
         ENA   (20B)      STA 7 (T+6)      . FRIENDLY SUB               01704
         SLJ   (10J)      ZRO   (0)        .                            01705
7J       LDA   (FRIEND)   AJP 2 (L+3)      .                            01706
         ENA   (44B)      STA 7 (T+6)      . HOSTILE SURFACE            01707
         SLJ   (15J)      ZRO   (0)        .                            01710
         ENA   (24B)      STA 7 (T+6)      . FRIENDLY SURFACE           01711
         SLJ   (13J)      ZRO   (0)        .                            01712
8J       LDA   (FRIEND)   AJP 2 (L+3)      .                            01713
         ENA   (34B)      STA 7 (T+6)      . HOSTILE A/C                01714
         SLJ   (18J)      ZRO   (0)        .                            01715
         ENA   (30B)      STA 7 (T+6)      . FRIENDLY A/C               01716
         SLJ   (16J)      ZRO   (0)        .                            01717
9J       ENA   (-1)       STA   (FRIEND)   . SET HOSTILE                01720
         SLJ   (5J)       ZRO   (0)        .                            01721
10J      LDA   (MASK14)   RAD 7 (T+6)      . SET FRIENDLY               01722
         RAO   (FSUB)     SAU 7 (T+6)      . SET TARGET NUMBER          01723
11J      LDA   (MASK18)   RAD 7 (T+6)      . SET SUB BIT                01724
         SLJ   (4T)       ZRO   (0)        . RETURN                     01725
12J      RAO   (HSUB)     SAU 7 (T+6)      . SET TARGET NUMBER          01726
         SLJ   (11J)      ZRO   (0)        .                            01727
13J      LDA   (MASK14)   RAD 7 (T+6)      . SET FRIENDLY               01730
         RAO   (FSUR)     SAU 7 (T+6)      . SET TARGET NUMBER          01731
14J      LDA   (MASK17)   RAD 7 (T+6)      . SET SURFACE BIT            01732
         SLJ   (4T)       ZRO   (0)        . RETURN                     01733
15J      RAO   (HSUR)     SAU 7 (T+6)      .                            01734
```

```
      SLJ    (14J)      ZRO    (0)        .                               01735
16J   LDA    (MASK14)   RAD 7  (T+6)      . SET FRIENDLY                  01736
      LIL 2  (3J)       LDQ    (ITIME)    . DETERMINE TYPE INTERCEPTOR    01737
      LDL    (1B)       AJP    (L+9)      . ZERO JUMP TO F4               01740
      LDA    (MASK8U)   RAD 7  (T+6)      . F8 TYPE                       01741
      RAO    (FAC)      LDA    (IFIG3)    . SET INITIAL                   01742
      SCL    (MASK9)    STA 2  (KFIG)     .   FUEL AND                    01743
      LDA    (ITIME)    RAD 2  (KFIG)     .     TIME                      01744
      LDA    (3J)       ALS    (42)       .                               01745
      LIL 2  (KNR2)     STA 2  (TRACK1)   . SET UP TRACK1 WORD            01746
      RAO    (KNR2)     LDA    (FAC)      .                               01747
      SAU 7  (T+6)      SLJ    (L+4)      . SET CAP NUMBER                01750
      LDA    (MASK4B)   RAD 7  (T+6)      . F4 TYPE                       01751
      RAO    (FAC)      LDA    (IFIG7)    .                               01752
      SLJ    (L-8)      ZRO    (0)        . JUMP TO INITIALIZE            01753
17J   LDA    (MASK16)   RAD 7  (T+6)      . SET A/C BIT                   01754
      SLJ    (4T)       ZRO    (0)        . RETURN                        01755
18J   RAO    (HAC)      SAU 7  (T+6)      . SET TARGET NUMBER             01756
      SLJ    (17J)      ZRO    (0)        .                               01757
19J   ENA    (TRACK)    ADD    (3J)       .                               01760
      STA    (T+6)      ENA    (0)        .                               01761
      STA 7  (T+6)      LDA    (3J)       . OWN SHIP SYMBOL SET           01762
      STA    (OWN)      SLJ    (13J)      .                               01763

* HOOK ROUTINE

3J0   SLJ    (N)        STA    (TAR)      . STORE ADDRESS OF ANSWER       01764
      LDA    (EIGHT)    FDV    (RHO)      . DETERMINE RANGE GATE          01765
      STA    (INC)      FAD    (XPIP)     .                               01766
      STA    (XDESIG)   LDA    (INC)      . DETERMINE X POSITION          01767
      FAD    (YPIP)     STA    (YDESIG)   . DETERMINE Y POSITION          01770
      ENI 3  (1)        ENI    (0)        .                               01771
3J1   LDA 3  (YS)       FSB    (YDESIG)   . CHECK FOR TARGET IN Y GATE    01772
      STA    (T+2)      ENA    (T+2)      .                               01773
      SAL    (2ABS)     SLJ 4  (1ABS)     .                               01774
      EQS    (INC)      SLJ    (L+2)      .                               01775
```

```
        SLJ    (L+2)       ZRO    (0)        .                                   01776
        THS    (INC)       SLJ    (3J2)      . JUMP IF NO TARGET IN GATE         01777
        LDA 3  (XS)        FSB    (XDESIG)   . CHECK FOR TARGET IN X GATE        02000
        STA    (T+2)       ENA    (T+2)      .                                   02001
        SAL    (2ABS)      SLJ 4  (1ABS)     .                                   02002
        EQS    (INC)       SLJ    (L+2)      .                                   02003
        SLJ    (L+2)       ZRO    (0)        .                                   02004
        THS    (INC)       SLJ    (3J2)      . JUMP IF NO TARGET IN GATE         02005
        SIL 3  (3J)        LDA    (3J)       .                                   02006
        STA 7  (TAR)       SLJ    (3J0)      . STORE ANSWER                      02007
3J2     ISK 3  (15)        SLJ    (3J1)      . LOOP                              02010
        ENA    (E13)       SLJ    (1E)       .                                   02011

* KEYBOARD ROUTINE ENTRY AND CHECK

1K      EXF    (77110B)    INT    (RANGE)    . INPUT RANGE SWITCH                02012
        EXF    (77001B)    LIL 5  (RANGE)    . DESLECT                           02013
        LDA 5  (RHO)       STA    (RHO)      . SET RANGE FACTOR                  02014
        EXF    (77102B)    INT    (XBALL)    . INPUT X-TRACK BALL                02015
        EXF    (77001B)    LDA    (XBALL)    . DESELECT                          02016
        ALS    (39)        ARS    (39)       . EXTEND SIGN BIT                   02017
        SLJ 4  (FLOAT)     ZRO    (0)        .                                   02020
        FDV    (RHO)       STA    (XPIP)     . RANGE SCALE X-POSITION            02021
        EXF    (77104B)    INT    (YBALL)    . INPUT Y-TRACK BALL                02022
        EXF    (77001B)    LDA    (YBALL)    . DESELECT                          02023
        ALS    (39)        ARS    (39)       . EXTEND SIGN BIT                   02024
        SLJ 4  (FLOAT)     ZRO    (0)        .                                   02025
        FDV    (RHO)       STA    (YPIP)     . RANGE SCALE Y-POSITION            02026
        LDA    (XBALL)     ALS    (36)       . ASSEMBLE HOOK WORD                02027
        ADD    (PIP)       STA    (HOOK)     .                                   02030
        LDA    (YBALL)     RAD    (HOOK)     .                                   02031
        EXF    (77010B)    EXF 7  (77010B)   . SELECT MEMORY UPDATE              02032
        OUT    (HOOK)      ENI    (0)        . OUTPUT HOOK TO DD65               02033
1K1     ENI 4  (120)       ENA    (53B)      . CHECK FOR $ IN BUF1               02034
        EQS 4  (BUF1)      SLJ    (4T)       . NO $ EXIT TO EXECUTIVE ROUTINE    02035
        SLJ    (1P)        ZRO    (0)        . JUMP TO PROGRAM CONTROL           02036
```

```
2K    LDA    (IFLAG)     AJP 1 (2T)      . CHECK FOR BEGIN FLAG                    02037
      LDA    (BUF1+1)    AJP   (L+2)     . NOTHING IN BUF1  JUMP TO 2T             02040
      EQS    (F13)       SLJ   (IT)      . CHECK FOR B IN BUFF+1                   02041
      SLJ    (1K1)       ZRO   (0)       . YES JUMP TO 1K1                         02042

* HISTORY ROUTINE

1L    EXF    (77221B)    SLJ   (4L)      . TURN OFF HISTORY LT                     02043
2L    ZRO    (0)         ZRO   (0)       . NUMBER OF POINTS DESIRED                02044
3L1   ZRO    (0)         ZRO   (0)       . MINUTES INSTEAD OF POINTS               02045
3L    ZRO    (0)         ZRO   (0)       . TRACK DESIRED                           02046
4L    ENA    (3L)        SLJ 4 (3J0)     . HOOK TRACK                              02047
      ENA    (9)         STA   (PPM)     . SET EVERY SWEEP COUNT                   02050
      LDA    (2L)        AJP   (8L)      . JUMP IF NR POINTS = ZERO                02051
      THS    (30B)       SLJ   (L+2)     .                                         02052
      SLJ    (9L)        ZRO   (0)       . JUMP IF NR POINTS .GT. 24               02053
5L    LDA    (3L1)       AJP 5 (10L)     .                                         02054
      LAC    (HISTORY)   AJP 2 (7L)      . DETERMINE WHICH ARRAYS TO USE           02055
      LIL 2  (3L1)       LIL 4 (2L)      . HIS1 ARRAYS                             02056
      SIU 4  (13F)       SIU 4 (4G)      . SET INDEXES                             02057
      LIL 3  (3L)        SIL 3 (HIS1)    .                                         02060
      LIL 4  (PPM)       LDA 4 (PP0)     .                                         02061
      STA    (PPMC1)     STA   (PPMC)    . SET TIMING CHAINS                       02062
      ENA    (XHIS1)     SLJ 4 (1L0)     . JUMP TO FILL ROUTINE                    02063
6L    SIU 4  (10F1)      SIL 5 (10F1)    . SAVE INDEXES                            02064
      SIU 6  (11F)       ENA   (0)       .   4, 5, 6                               02065
      STA    (3L1)       STA   (2L)      . ZERO ARGUMENTS                          02066
      SLJ    (10F)       ZRO   (0)       . JUMP TO DISPLAY                         02067
7L    LIL 2  (3L1)       LIL 4 (2L)      . HIS2 ARRAYS                             02070
      SIU 4  (15F)       SIU 4 (6G)      . SET INDEXES                             02071
      LIL 3  (3L)        SIL 3 (HIS2)    .                                         02072
      LIL 4  (PPM)       LDA 4 (PP0)     .                                         02073
      STA    (PPMC2)     STA   (PPMC)    . SET TIMING CHAINS                       02074
      ENA    (XHIS2)     SLJ 4 (1L0)     . JUMP TO FILL ROUTINE                    02075
      SLJ    (6L)        ZRO   (0)       .                                         02076
8L    ENA    (12)        STA   (2L)      . PUT 12 IN 2L                            02077
```

```
         SLJ    (5L)          ZRO   (0)       .                          02100
9L       ENA    (24)          STA   (2L)      . PUT 24 IN 2L             02101
         SLJ    (5L)          ZRO   (0)       .                          02102

* TIMING CHAIN ROUTINE

10L      SLJ    (N)           ENI   (0)       . EXIT/ENTRY               02103
         ENA    (60)          ENQ   (0)       . DETERMINE PERIODS        02104
         DVI    (IP)          STA   (PPM)     .   PER MIN                02105
         ENQ    (0)           MUI   (2L)      . POINTS NEEDED            02106
         STA    (3L1)         LDA   (TIME)    .                          02107
         SUB    (TFLAG)       ENQ   (0)       .                          02110
         DVI    (IP)          ENQ   (0)       .                          02111
         THS    (3L1)         SLJ   (L+2)     . ENOUGH POINTS STORED     02112
         STA    (3L1)         ENI   (0)       .   NO REDUCE 3L1          02113
         LDA    (PPM)         ENI 2 (10)      .                          02114
         EQS 2 (PTABLE)       SLJ   (L+3)     . DETERMINE TIMING CHAIN   02115
         LDA 2 (PPO)          STA   (PPMC)    .                          02116
         SIL 2 (PPM)          SLJ   (5L)      .                          02117
         LDA    (PPM)         ENI 2 (10)      .                          02120
         THS 2 (PTABLE)       SLJ   (4T)      . RETURN                   02121
         LDA 2 (PPO)          STA   (PPMC)    .                          02122
         SIL 2 (PPM)          SLJ   (10L)     . RETURN                   02123

* FILL ARRAYS ROUTINE

1L0      SLJ    (N)           SAL   (1L1)     . EXIT/ENTRY               02124
         INA    (25)          SAL   (1L2)     . BACK UP MEMORY           02125
90       BACKSPACE MM                                                    02126
         IJP 2 (90)           LIL 2 (3L1)     . LOOP                     02127
         LIL 5 (2L)           ENI   (0)       .                          02130
990      READ TAPE MM, IXH,IYH                                           02131
         SSH    (PPMC)        SLJ   (1L3)     . FILL ARRAY               02132
1L1      LDA 3 (IXH)          STA 5 (N)       .                          02133
1L2      LDA 3 (IYH)          STA 5 (N)       .                          02134
```

```
         INI 5 (-1)        ENI   (0)        .                                02135
1L3      IJP 2 (990)       SLJ   (1L0)      . RETURN                         02136

* ERASE ROUTINE

1MER     SLJ   (N)         SIU 4 (2MER)     . EXIT/ENTRY                     02137
         ENI 4 (2)         ENI   (0)        . SET INDEX FOR LOOP             02140
         EXF   (02000B)    EXF   (04001B)   . STOP RT CLOCK  DISALLOW        02141
         EXF   (77010B)    OUT   (ERASE)    . FIRST ERASE WORD               02142
         EXF   (77010B)    OUT   (S6C)      . STRING ERASE WORDS             02143
         ISK 4 (398)       SLJ   (L-1)      . LOOP                           02144
         EXF   (04000B)    EXF   (01000B)   . ALLOW  START RT CLOCK          02145
2MER     ENI 4 (N)         SLJ   (1MER)     . RETURN                         02146

* MESSAGE ROUTINE

1MSG     SLJ   (N)         STA   (T+8)      . EXIT/ENTRY                     02147
         ENA   (XP1)       SAL   (PRINT+1)  . SET UP ARGUMENTS FOR PRINT     02150
         ENA   (YMSG)      SAL   (PRINT+2)  .                                02151
         SIU 5 (2MSG)      LIL 5 (PRO)      .                                02152
         SIL 5 (PRINT+3)   LIL 5 (ADD)      .                                02153
         SIL 5 (PRINT+4)   ENQ   (0B)       .                                02154
         LDA   (T+8)       SLJ 4 (PRINT)    . JUMP TO PRINT                  02155
         LIL 5 (PRINT+4)   SIL 5 (ADD)      . RESET ADD                      02156
         ENA   (-12B)      RAD   (YMSG)     . DECREASE YMSG                  02157
2MSG     ENI 5 (N)         SLJ   (1MSG)     . RETURN                         02160

* GRAPH TITLE UPDATE

1N       SLJ   (*)         LDA   (GTIME)    .                                02161
         ADD   (R3)        STA   (TITLE9)   . STORE                          02162
         STA   (TITLE21)   LDA   (TFLAG)    .   INITIAL TIME                 02163
         ENQ   (0)         DVI   (60)       .                                02164
         STQ   (T+10)      SLJ 4 (1NUM)     .                                02165
         ALS   (30)        SCL   (MASK37)   .                                02166
         ADD   (R4)        STA   (T+9)      . STORE FINAL                    02167
```

```
        LDA   (T+10)      ENQ   (0)        .   INTEGER TIME                02170
        MUI   (10)        ENQ   (0)        .                               02171
        DVI   (60)        SLJ 4 (1NUM)     .                               02172
        ALS   (18)        SCL   (MASK38)   .                               02173
        RAD   (T+9)       STA   (TITLE10)  . STORE FINAL                   02174
        STA   (TITLE22)   SCL   (MASK39)   .   DECIMAL TIME                02175
        STA   (GTIME)     SLJ   (1N)       .                               02176

* OCTAL TO BCD NUMBER CONVERTER 0 TO 9,999,999

1NUM    SLJ   (N)         SIU 2 (3NUM)     . EXIT/ENTRY                    02177
        ENI 2 (N)         ENI   (0)        .                               02200
        THS   (1000000)   INI 2 (1)        .                               02201
        THS   (100000)    INI 2 (1)        . TAKES OCTAL NUMBER            02202
        THS   (10000)     INI 2 (1)        .   CONVERTS IT TO BCD          02203
        THS   (1000)      INI 2 (1)        .     AND PACKS IT IN THE       02204
        THS   (100)       INI 2 (1)        .       LOWER 36 BITS OF        02205
        THS   (10)        INI 2 (1)        .         T WITH THE REST       02206
        LDQ   (TITLE6)    STQ   (T)        .             OF THE WORD BLANKS 02207
        ENQ   (0)         DVI 2 (R0)       .                               02210
        SAL   (2NUM)      LIL 1 (2NUM)     .                               02211
        LDA   (T)         ALS   (6)        .                               02212
        SCL   (77B)       ADD 1 (F0)       . I. E.                         02213
        STA   (T)         LLS   (48)       .   2020NNNNNNNNNNB             02214
        IJP 2 (L-5)       LDA   (T)        . LOOP                          02215
3NUM    ENI 2 (N)         SLJ   (1NUM)     . RETURN                        02216
2NUM    ZRO   (0)         ZRO   (0)        .                               02217

* PROGRAM CONTROL ROUTINE

1P      LIL 4 (KNR)       ENI   (0)        . SET B4 = KNR                  02220
        LDA 4 (BUF1)      ENI 1 (26)       . CHECK TO SEE IF FIRST         02221
        EQS 1 (F12)       SLJ   (2P)       .   CHARACTER IS LETTER         02222
        ENA   (0)         STA   (T)        .                               02223
        LDA 4 (BUF1)      ENI 1 (36)       .                               02224
        EQS 1 (F0)        SLJ   (L+4)      . ASSEMBLE NAME                 02225
```

```
        LDA   (T)       ALS   (6)       .                                  02226
        ADD 4 (BUF1)    STA   (T)       .                                  02227
        INI 4 (1)       SLJ   (L-4)     .                                  02230
        SUB   (F56)     AJP   (L-1)     .                                  02231
        LDA   (T)       ENQ   (0)       . POSITION NAME                    02232
        LRS   (6)       AJP 1 (L)       .                                  02233
        LDL   (-0B)     LIL 3 (NPROG)   .                                  02234
        EQS 3 (P0)      SLJ   (2P)      .                                  02235
        LDA 3 (0Z)      SAU   (L+7)     .                                  02236
        SAL   (4P)      ARS   (24)      .                                  02237
        STA   (5P)      SLJ 4 (3P)      . PROCESS ARGUMENTS                02240
        ENA   (1)       STA   (KNR)     .                                  02241
        STA   (KNR1)    ENI   (0)       .                                  02242
        ENA   (0)       ENI 4 (120)     . CLEAR BUFFER                     02243
        STA 4 (BUF1)    IJP 4 (L)       . LOOP                             02244
        SLJ   (N)       ZRO   (0)       . EXECUTE PROGRAM                  02245
2P      SLJ 4 (13P)     ZRO   (0)       .                                  02246
        ENA   (E0)      SLJ   (1E)      . PROGRAM NOT CALLED               02247
3P      SLJ   (N)       SLJ   (L+3)     . PROCESS ARGUMENTS                02250
4P      ZRO   (0)       ZRO   (0)       . NEXT ARGUMENT ADDRESS            02251
5P      ZRO   (0)       ZRO   (0)       . NUMBER OF ARGUMENTS              02252
6P      LDA 4 (BUF1)    AJP   (8P)      .                                  02253
        INA   (-33B)    AJP   (L+5)     . SENSE COMMA OR OPEN PAREN        02254
        INA   (-1B)     AJP   (L+4)     .                                  02255
        INA   (-37B)    AJP   (3P)      . SENSE PERIOD OR CLOSE PAREN      02256
        INA   (-1)      AJP   (3P)      .                                  02257
        SLJ   (8P)      ZRO   (0)       .                                  02260
        INI 4 (1)       LDA 4 (BUF1)    . SKIP SPACES                      02261
        SUB   (F56)     AJP   (L-1)     .                                  02262
        ENA   (0)       STA   (T)       .                                  02263
        LDA 4 (BUF1)    ENI 1 (10)      .                                  02264
        EQS   (F53)     SLJ   (L+3)     . SENSE MINUS SIGN                 02265
        STA   (NEG)     INI 4 (1)       . SET NEGATIVE FLAG                02266
        LDA 4 (BUF1)    SLJ   (L+3)     .                                  02267
        EQS   (60B)     SLJ   (L+2)     . SENSE PLUS SIGN                  02270
        INI 4 (1)       LDA 4 (BUF1)    .                                  02271
```

```
        EQS 1 (F0)      SLJ   (L+2)     . SENSE OCTAL NUMBER               02272
        SLJ   (9P)      ZRO   (0)       . ASSEMBLE NUMBER                  02273
        LDA 4 (BUF1)    ENI 1 (36)      .                                  02274
        EQS 1 (F0)      SLJ   (L+4)     .                                  02275
        LDA   (T)       ALS   (6)       . ASSEMBLE NAME                    02276
        ADD 4 (BUF1)    STA   (T)       .                                  02277
        INI 4 (1)       SLJ   (L-4)     . LOOP                             02300
        SUB   (F56)     AJP   (L-1)     .                                  02301
        LDA   (T)       ENQ   (0)       . POSITION ARGUMENT                02302
        LRS   (6)       AJP 1 (L)       .                                  02303
        STQ   (T)       ENI   (0)       .                                  02304
7P      RAO   (4P)      RSO   (5P)      .                                  02305
        AJP 3 (8P)      LDA   (T)       . SENSE ARGUMENT LIMIT             02306
        LDQ   (NEG)     QJP   (L+3)     . CHECK NEGATIVE FLAG              02307
        LAC   (T)       ENQ   (0)       .                                  02310
        STQ   (NEG)     ENI   (0)       .                                  02311
        STA 7 (4P)      SLJ   (6P)      . LOOP                             02312
8P      SLJ 4 (13P)     ZRO   (0)       .                                  02313
        ENA   (E3)      SLJ   (1E)      . TOO MANY ARGUMENTS               02314
9P      SIL 4 (10P)     SLJ   (L+2)     .                                  02315
10P     ZRO   (0)       ZRO   (0)       . PRESENT POSITION IN ARRAY        02316
        LIL 2 (10P)     ENI 5 (0)       . DETERMINE WIDTH OF NUMBER        02317
        LDA 2 (BUF1)    ENI   (0)       .                                  02320
        INA   (-33B)    AJP   (11P)     . SENSE COMMA                      02321
        INA   (-01B)    AJP   (11P)     . SENSE OPEN PAREN                 02322
        INA   (-37B)    AJP   (11P)     . SENSE PERIOD                     02323
        INA   (-01B)    AJP   (11P)     . SENSE CLOSING PAREN              02324
        INI 2 (1)       INI 5 (1)       . B5 NOW HAS WIDTH OF NUMBER       02325
        SLJ   (L-6)     ZRO   (0)       .                                  02326
11P     ENA   (0)       STA   (T)       .                                  02327
        ENI 1 (10)      SLJ 4 (12P)     .                                  02330
        EQS 1 (F0)      SLJ   (2P)      . SENSE OCTAL NUMBER               02331
        LDA   (T)       MUI   (F0)      . MULTIPLY BY TEN                  02332
        INA 1 (0)       STA   (T)       . INCREASE BY NEXT DIGIT           02333
        SLJ   (L-4)     ZRO   (0)       . LOOP                             02334
```

```
* DETERMINE OCTAL NUMBER ROUTINE

12P    SLJ    (N)         IJP 5 (L+2)    . EXIT/ENTRY                02335
       SLJ    (7P)        ZRO   (0)      .                           02336
       LDA 4  (BUF1)      SAL   (L+3)    .                           02337
       EQS    (F56)       SLJ   (L+2)    . CHECK FOR SPACES          02340
       LDA    (F0)        SAL   (L+1)    . PUT ZEROS IN FOR SPACES   02341
       INI 4  (1)         ENA   (N)      .                           02342
       SLJ    (12P)       ZRO   (0)      . RETURN                    02343

* ZERO OUT BUF1 UPON ERROR ROUTINE

13P    SLJ    (N)         LIL 4 (KNR1)   . EXIT/ENTRY                02344
       ENA    (0)         STA 4 (BUF1)   . ZERO OUT BUF1             02345
       IJP 4  (L-1)       SLJ   (13P)    . RETURN                    02346

* HOLD OR PAUSE ROUTINE

1PQ    LDA    (TIME)      STA   (PAUSE)  . STORE TIME                02347
       ENA    (1)         STA   (PFLAG)  . SET P-FLAG                02350
       LDA    (PFLAG)     AJP 1 (L)      . WAIT LOOP                 02351
       EXF    (77303B)    LDA   (PAUSE)  . TURN OFF PAUSE LIGHT      02352
       STA    (TIME)      STA   (ITIME)  . RESTORE TIME              02353
       SLJ    (4T)        ZRO   (0)      . RETURN                    02354

* INTERCEPT COMMAND ROUTINE

1Q     EXF    (77321B)    SLJ   (L+3)    . TURN OFF INT LIGHT        02355
2Q     ZRO    (0)         ZRO   (0)      . CAP NR OF INTECPTOR       02356
3Q     ZRO    (0)         ZRO   (0)      . BOGEY HOOKED              02357
       ENA    (3Q)        SLJ 4 (3J0)    . HOOK BOGEY                02360
       ENI 2  (1)         ENI   (0)      . SET INDEX FOR LOOP        02361
       SSK 2  (TRACK)     SLJ   (L+6)    . SENSE FRIENDLY            02362
       LDA    (MASK16)    LDQ 2 (TRACK)  .                           02363
       MEQ    (MASK16)    SLJ   (L+5)    . CHECK FOR A/C BIT         02364
       QRS    (24)        LDL   (77B)    .                           02365
```

```
      EQS    (2Q)        SLJ    (L+2)      . IS CAP CALLED FOR AIRBORNE          02366
      SIU 2  (2Q)        SLJ    (4Q)       . YES JUMP TO 4Q                      02367
      ISK 2  (15)        SLJ    (L-6)      . LOOP                                02370
      ENI 2  (16)        LDQ    (MASK11)   . LAUNCH CAP ROUTINE                  02371
      ENA    (52B)       ENI    (0)        .                                     02372
      MEQ 2  (TRACK)     SLJ    (5Q)       . CHECK FOR NON ACTIVE TRACKS         02373
      ENA    (2Q)        SLJ 4  (1X1)      . GET CAP NUMBER                      02374
      LDA    (2Q)        ALS    (24)       .                                     02375
      RAD 2  (TRACK)     ENI    (0)        . SET CAP NUMBER                      02376
      LDA    (MASK14)    RAD 2  (TRACK)    . SET FRIENDLY SYMBOL                 02377
      LDA    (MASK4B)    RAD 2  (TRACK)    . SET TYPE INTERCEPTOR                02400
      LDA    (MASK23)    RAD 2  (TRACK)    . SET LAUNCH BIT                      02401
      SIU 2  (2Q)        LDA    (IFIG7)    . STORE CAP TRACK NR                  02402
      SCL    (MASK9)     STA 2  (KFIG)     . INITIALIZE FUEL                     02403
      LDA    (ITIME)     RAD 2  (KFIG)     .   AND LAUNCH TIME                   02404
4Q    LDA    (MASK19)    RAD 2  (TRACK)    . SET INTERCEPT BIT                   02405
      LDQ    (2Q)        QRS    (24)       .                                     02406
      LDL    (77B)       ALS    (42)       .                                     02407
      SLJ 4  (6Q)        ZRO    (0)        . JUMP TO DETERMINE TRACK1 WORD       02410
      LIL 3  (8Q)        STA    (T+2)      .                                     02411
      LDA 3  (TRACK1)    SCL    (MASK36)   .                                     02412
      ADD    (T+2)       STA 3  (TRACK1)   . SET UP TRACK1 WORD                  02413
      LDQ 2  (TRACK)     LDL    (MASK10)   .                                     02414
      STA 2  (TRACK)     ENA    (55B)      . SET ENAGAGED CAP SYMBOL             02415
      RAD 2  (TRACK)     LIL 2  (3Q)       .                                     02416
      LDQ 2  (TRACK)     LDL    (MASK10)   .                                     02417
      STA 2  (TRACK)     ENA    (54B)      . SET ENGAGED BOGEY SYMBOL            02420
      RAD 2  (TRACK)     ENI    (0)        .                                     02421
      LDA    (3Q)        ALS    (36)       .                                     02422
      RAD 3  (TRACK1)    ENA    (0)        .                                     02423
      STA    (2Q)        SLJ    (4T)       .                                     02424
5Q    ENA    (E56)       SLJ 4  (1MSG)     . NO CAP AVAILABLE                    02425
      ENA    (0)         STA    (2Q)       .   TO BE LAUNCHED                    02426
      SLJ    (4T)        ZRO    (0)        .                                     02427
```

```
* DETERMINE WHICH TRACK1 WORD TO USE

6Q      SLJ     (N)           SIL 4 (7Q)       . EXIT/ENTRY                          02430
        LIL 4   (KNR2)        LDQ   (MASK36)   .                                     02431
        MEQ 4   (TRACK1)      SLJ   (L+3)      . CHECK FOR CAP TRACK NR              02432
7Q      SIL 4   (8Q)          ENI 4 (N)        . SAVE TRACK1 INDEX                   02433
8Q      SLJ     (6Q)          ZRO   (0)        . RETURN                              02434
        LIL 4   (KNR2)        SIL 4 (8Q)       . USE NEXT TRACK1 WORD                02435
        STA     (9Q)          RAO   (KNR2)     . INCREASE KNR2                       02436
        LDA     (9Q)          SLJ   (7Q)       . RETURN                              02437
9Q      ZRO     (0)           ZRO   (0)        .                                     02440

* DROP HISTORY ROUTINE

1R      EXF     (77321B)      SLJ   (3R)       . TURN OFF DROP LT                    02441
2R      ZRO     (0)           ZRO   (0)        . TRACK TO BE DROPPED                 02442
3R      ENA     (2R)          SLJ 4 (3J0)      . HOOK TRACK                          02443
        EQS     (HIS1)        SLJ   (4R)       . HIS1  NO  --JUMP  4R                02444
        ENA     (0)           STA   (HIS1)     . RESET HIS1                          02445
        ENA     (-0B)         STA   (HISTORY)  . RESET HISTORY                       02446
        SLJ     (4T)          ZRO   (0)        . RETURN                              02447
4R      EQS     (HIS2)        SLJ   (4T)       . HIS2  NO  --JUMP  4T                02450
        ENA     (0)           STA   (HIS2)     . RESET HSI2                          02451
        STA     (HISTORY)     SLJ   (4T)       . RESET HISTORY    RETURN             02452

* STOP AND CRITIQUE ROUTINE

1S      EXF     (77403B)      ENA   (0)        . TURN OFF STOP LT                    02453
        STA     (NPTS)        STA   (NTRACKS)  . SET NPTS AND NTRACKS        =0      02454
        ENA     (1)           STA   (MOD)      . SET MOD                     =1      02455
        ENI 6   (1)           LIL 3 (NTRACKS)  . SET INDEXES                         02456
        ENA     (E68)         SLJ 4 (1MSG)     . GRAPH BEING PLOTTED MSG             02457
9991    BACKSPACE MM                                                                 02460
        READ TAPE MM, XN,YN,ZN, TFLAG                                                02461
        SLJ 4   (1N)          ZRO   (0)        . UPDATE GRAPH TITLE                  02462
        LDA     (TFLAG)       SUB   (MTIME)    . DETERMINE IF THIS GRAPH             02463
```

```
      AJP 6 (12S1)       ENI    (0)        .  WILL EXCEED MAX TIME
91    REWIND MM
      RAO   (NTRACKS)    LIL 3 (NTRACKS)  . INCREASE NTRACKS
991   READ TAPE MM, XN,YN,ZN,JTIME
      LDA 3 (XN)         AJP 1 (L+3)      . IS X ZERO  NO JUMP
      LDA 3 (YN)         AJP 1 (L+2)      . IS Y ZERO  NO JUMP
      SLJ   (4S1)        ZRO   (0)        . BOTH X AND Y ZERO  JUMP
      LDA 3 (ZN)         AJP 7 (13S1)     . JUMP IF SUBMARINE
      STA 6 (Y)          LDA   (JTIME)    . STORE Z DATA
      SUB   (FTIME)      SLJ 4 (FLOAT)    . CHANGE TIME TO MIN
      FDV   (60.)        STA 6 (X)        . STORE TIME
      RAO   (NPTS)       LDA   (JTIME)    . INCREASE NPTS
      SUB   (TFLAG)      AJP   (5S1)      . MORE TIME  NO JUMP
      AJP 2 (5S1)        ENA   (1603B)    . MORE POINTS  YES JUMP
      THS   (NPTS)       SLJ   (6S1)      .
2S11  ENA   (NPTS)       SAL   (DRAW+1)   . FILL DRAW ARGUMENTS
      ENA   (X)          SAL   (DRAW+2)   .
      ENA   (Y)          SAL   (DRAW+3)   .
      ENA   (MOD)        SAL   (DRAW+4)   .
      ENA   (S1C)        SAL   (DRAW+5)   .
      ENA   (LABLE)      SAL   (DRAW+6)   .
      ENA   (ITITLE)     SAL   (DRAW+7)   .
      ENA   (XSCALE1)    SAL   (DRAW+8)   .
      ENA   (YSCALE1)    SAL   (DRAW+9)   .
      ENA   (IXUP1)      SAL   (DRAW+10)  .
      ENA   (IYRT1)      SAL   (DRAW+11)  .
      ENA   (MODXAX1)    SAL   (DRAW+12)  .
      ENA   (MODYAX1)    SAL   (DRAW+13)  .
      ENA   (IWIDE1)     SAL   (DRAW+14)  .
      ENA   (IHIGH1)     SAL   (DRAW+15)  .
      ENA   (IGRID)      SAL   (DRAW+16)  .
      ENA   (LAST)       SAL   (DRAW+17)  .
      SLJ 4 (DRAW)       ZRO   (0)        . JUMP TO DRAW
      ENI 5 (300)        ENA   (0)        . CLEAR ARRAY
      STA 5 (X)          STA 5 (Y)        .
      IJP 5 (L-1)        ENI   (0)        . LOOP
```

```
      LDA   (LAST)      AJP 1 (7S1)    . CHECK TO SEE IF GOOD GRAPH     02530
2S12  ENA   (3)         ENI   (0)      . IS MOD = 3                     02531
      EQS   (MOD)       SLJ   (L+2)    .                                02532
3S1   SLJ   (1S1)       ZRO   (0)      .   YES JUMP TO X-Y PLOT         02533
      ENA   (2)         STA   (MOD)    . SET MOD = 2                    02534
      ENA   (0)         STA   (NPTS)   . SET NPTS TO ZERO               02535
      LDA   (JTIME)     SUB   (TFLAG)  . MORE TIME                      02536
      AJP   (9S1)       AJP 2 (9S1)    .   NO JUMP TO REWIND            02537
      ENI 6 (1)         SLJ   (991)    .   YES JUMP TO READ             02540
4S1   ENA   (2)         ENI   (0)      . IS NPTS .GT. 2                 02541
      THS   (NPTS)      SLJ   (10S1)   . NO - IGNORE GRAPH              02542
      LDA   (JTIME)     SUB   (TFLAG)  .   AND JUMP TO READ             02543
      AJP   (5S1)       AJP 2 (5S1)    . MORE TIME  NO JUMP             02544
      SLJ   (2S11)      ZRO   (0)      .   YES JUMP TO GRAPH            02545
5S1   ENA   (15)        ENI   (0)      . CHECK FOR LAST TRACK           02546
      EQS   (NTRACKS)   SLJ   (11S1)   . LAST TRACK   NO JUMP         02547
      ENA   (3)         STA   (MOD)    .   YES SET MOD = 3              02550
      SLJ   (2S11)      ZRO   (0)      . JUMP TO GRAPH                  02551
6S1   INI 6 (1)         SLJ   (991)    . INCREASE INDEX  JUMP TO READ   02552
7S1   LDA   (LAST)      SUB   (2B)     . LAST = 2                       02553
      AJP 1 (2S12)      STA   (NPTS)   .   NO JUMP TO MOD CHECK         02554
      ENA   (1)         STA   (MOD)    . RESET NPTS AND MOD             02555
      ENI 6 (1)         LDA   (JTIME)  .   AND INDEX                    02556
      SUB   (TFLAG)     AJP   (991)    . MORE TIME                      02557
      AJP 2 (991)       SLJ   (91)     .   YES JUMP TO REWIND           02560
8S1   SLJ   (1S1)       ZRO   (0)      .   NO JUMP TO X-Y POSIT PLOT    02561
9S1   ENA   (0)         STA   (NPTS)   . RESET NPTS                     02562
      ENI 6 (1)         SLJ   (91)     . RESET INDEX  JUMP TO REWIND    02563
10S1  ENA   (0)         STA   (NPTS)   . RESET NPTS AND INDEX           02564
      ENI 6 (1)         SLJ   (991)    . JUMP TO READ                   02565
11S1  ENA   (2)         ENI   (0)      . NPTS .GT. TWO                  02566
      THS   (NPTS)      SLJ   (9S1)    .   NO JUMP TO GRAPH             02567
      SLJ   (2S11)      ZRO   (0)      .   YES JUMP TO GRAPH            02570
```

```
* RESET BASE TIME ROUTINE

12S1   SLJ    (N)          LDA    (JTIME)     . EXIT/ENTRY                        02571
       STA    (FTIME)      ENA    (1800)      . SET FTIME TO LAST JTIME           02572
       RAD    (MTIME)      SLJ    (12S1)      . INCREASE MAX TIME  RETURN         02573

*SCALE SUBMARINE DEPTH ROUTINE

13S1   SLJ    (N)          FMU    (SCALE)     . EXIT/ENTRY                        02574
       SLJ    (13S1)       ZRO    (0)         . SCALE   RETURN                    02575

* X-Y PLOT ROUTINE

1S1    LIL 3 (PPI)         ENI 6 (1)          . SET INDEXES                       02576
       ENA    (0)          STA    (NPTS)      . SET NPTS                  =0      02577
       ENA    (1)          STA    (MOD)       . RESET MOD TO ONE                  02600
1S2    LDA 3 (X1)          AJP 1 (L+3)        . IS X ZERO   NO JUMP               02601
       LDA 3 (Y1)          AJP 1 (L+2)        . IS Y ZERO   NO JUMP               02602
       SLJ    (L+5)        ZRO    (0)         . BOTH X AND Y ZERO JUMP            02603
       LDA 3 (X1)          STA 6 (X)          . STORE X DATA                      02604
       LDA 3 (Y1)          STA 6 (Y)          . STORE Y DATA                      02605
       ISK 6 (30)          SLJ    (L+2)       . CHECK FOR NPTS = 30               02606
       SLJ    (1S3)        ZRO    (0)         . JUMP TO GRAPH                     02607
       IJP 3 (1S2)         SIL 6 (NPTS)       .                                   02610
       SLJ    (L+2)        ZRO    (0)         .                                   02611
1S3    ENA    (30)         STA    (NPTS)      . SET NPTS EQUAL TO THIRTY          02612
       ENA    (NPTS)       SAL    (DRAW+1)    . FILL DRAW ARGUMENTS               02613
       ENA    (X)          SAL    (DRAW+2)    .                                   02614
       ENA    (Y)          SAL    (DRAW+3)    .                                   02615
       ENA    (MOD)        SAL    (DRAW+4)    .                                   02616
       ENA    (F5)         SAL    (DRAW+5)    .                                   02617
       ENA    (LABLE)      SAL    (DRAW+6)    .                                   02620
       ENA    (KTITLE)     STA    (DRAW+7)    .                                   02621
       ENA    (XSCALE)     SAL    (DRAW+8)    .                                   02622
       ENA    (YSCALE)     SAL    (DRAW+9)    .                                   02623
       ENA    (IXUP)       SAL    (DRAW+10)   .                                   02624
```

```
        ENA    (IYRIGHT)   SAL    (DRAW+11)  .                                  02625
        ENA    (MODEXAX)   SAL    (DRAW+12)  .                                  02626
        ENA    (MODEYAX)   SAL    (DRAW+13)  .                                  02627
        ENA    (IWIDE)     SAL    (DRAW+14)  .                                  02630
        ENA    (IHIGH)     SAL    (DRAW+15)  .                                  02631
        ENA    (IGRID)     SAL    (DRAW+16)  .                                  02632
        ENA    (LAST)      SAL    (DRAW+17)  .                                  02633
        SLJ 4  (DRAW)      ZRO    (0)        . JUMP TO DRAW                     02634
        ENI 5  (50)        ENA    (0)        . CLEAR ARRAY                      02635
        STA 5  (X)         STA 5  (Y)        .                                  02636
        IJP 5  (L-1)       LDA    (LAST)     . LOOP                             02637
        EQS    (2B)        SLJ    (L+4)      . CHECK FOR LAST EQUAL TWO         02640
        ENA    (1)         STA    (MOD)      . SET MOD EQUAL ONE                02641
        ENI 6  (1)         IJP 3  (1S2)      . RESET INDEX CHECK MORE POINTS    02642
        LIL 3  (PPI)       SLJ    (1S4)      . NO RESET INDEX  JUMP NEXT PLOT   02643
        ENA    (2)         STA    (MOD)      . SET MOD EQUAL TWO                02644
        ENI 6  (1)         IJP 3  (1S2)      . CHECK FOR MORE POINTS            02645
        LIL 3  (PPI)       ENI    (0)        .                                  02646
1S4     LDA 3  (X2)        AJP 1  (L+3)      . IS X ZERO  NO JUMP               02647
        LDA 3  (Y2)        AJP 1  (L+2)      . IS Y ZERO  NO JUMP               02650
        SLJ    (L+5)       ZRO    (0)        . BOTH X AND Y ZERO JUMP           02651
        LDA 3  (X2)        STA 6  (X)        . STORE X DATA                     02652
        LDA 3  (Y2)        STA 6  (Y)        . STORE Y DATA                     02653
        ISK 6  (30)        SLJ    (L+2)      . CHECK FOR NPTS = 30              02654
        SLJ    (1S5)       ZRO    (0)        . JUMP TO GRAPH                    02655
        IJP 3  (1S4)       SIL 6  (NPTS)     .                                  02656
        SLJ    (L+2)       ZRO    (0)        .                                  02657
1S5     ENA    (30)        STA    (NPTS)     . SET NPTS EQUAL TO THIRTY         02660
        ENA    (NPTS)      SAL    (DRAW+1)   . FILL DRAW ARGUMENTS              02661
        ENA    (X)         SAL    (DRAW+2)   .                                  02662
        ENA    (Y)         SAL    (DRAW+3)   .                                  02663
        ENA    (MOD)       SAL    (DRAW+4)   .                                  02664
        ENA    (F1)        SAL    (DRAW+5)   .                                  02665
        ENA    (LABLE)     SAL    (DRAW+6)   .                                  02666
        ENA    (KTITLE)    STA    (DRAW+7)   .                                  02667
        ENA    (XSCALE)    SAL    (DRAW+8)   .                                  02670
```

```
        ENA    (YSCALE)     SAL    (DRAW+9)   .                                 02671
        ENA    (IXUP)       SAL    (DRAW+10)  .                                 02672
        ENA    (IYRIGHT)    SAL    (DRAW+11)  .                                 02673
        ENA    (MODEXAX)    SAL    (DRAW+12)  .                                 02674
        ENA    (MODEYAX)    SAL    (DRAW+13)  .                                 02675
        ENA    (IWIDE)      SAL    (DRAW+14)  .                                 02676
        ENA    (IHIGH)      SAL    (DRAW+15)  .                                 02677
        ENA    (IGRID)      SAL    (DRAW+16)  .                                 02700
        ENA    (LAST)       SAL    (DRAW+17)  .                                 02701
        SLJ 4  (DRAW)       ZRO    (0)        . JUMP TO DRAW                    02702
        ENI 5  (50)         ENA    (0)        . CLEAR ARRAY                     02703
        STA 5  (X)          STA 5  (Y)        .                                 02704
        IJP 5  (L-1)        LDA    (LAST)     . LOOP                            02705
        EQS    (2B)         SLJ    (L+5)      . CHECK FOR LAST EQUAL TWO        02706
        ENA    (1)          STA    (MOD)      . RESET MOD EQUAL ONE             02707
        ENI 6  (1)          IJP 3  (1S4)      . CHECK FOR MORE POINTS           02710
        ENA    (1)          STA    (PPI)      . RESET PPI TO ONE                02711
        SLJ    (L+4)        ZRO    (0)        .                                 02712
        ENA    (1)          STA    (MOD)      . SET MOD                         02713
        ENI 6  (1)          IJP 3  (1S4)      . CHECK FOR MORE POINTS           02714
        SLJ    (L-4)        ZRO    (0)        .                                 02715
        ENA    (0)          STA    (NPTS)     .                                 02716
        STA    (NTRACKS)    LIL 3  (NTRACKS)  .                                 02717
9       REWIND MM                                                               02720
        RAO    (NTRACKS)    LIL 3  (NTRACKS)  . INCREASE NTRACKS                02721
99      READ TAPE MM, XN,YN,ZN,JTIME                                            02722
        LDA 3  (XN)         AJP 1  (L+3)      . IS X ZERO  NO JUMP              02723
        LDA 3  (YN)         AJP 1  (L+2)      . IS Y ZERO  NO JUMP              02724
        SLJ    (4S)         ZRO    (0)        . BOTH X AND Y ZERO  JUMP         02725
        LDA 3  (XN)         STA 6  (X)        . STORE X DATA                    02726
        LDA 3  (YN)         STA 6  (Y)        . STORE Y DATA                    02727
        RAO    (NPTS)       LDA    (JTIME)    . INCREASE NPTS                   02730
        SUB    (TFLAG)      AJP    (5S)       . MORE TIME  NO JUMP              02731
        AJP 2  (5S)         ENA    (1603B)    . MORE POINTS  YES JUMP           02732
        THS    (NPTS)       SLJ    (6S)       .                                 02733
25      ENA    (NPTS)       SAL    (DRAW+1)   . FILL DRAW ARGUMENTS             02734
```

```
       ENA    (X)         SAL    (DRAW+2)   .                                  02735
       ENA    (Y)         SAL    (DRAW+3)   .                                  02736
       ENA    (MOD)       SAL    (DRAW+4)   .                                  02737
       ENA    (SIC)       SAL    (DRAW+5)   .                                  02740
       ENA    (LABLE)     SAL    (DRAW+6)   .                                  02741
       ENA    (KTITLE)    STA    (DRAW+7)   .                                  02742
       ENA    (XSCALE)    SAL    (DRAW+8)   .                                  02743
       ENA    (YSCALE)    SAL    (DRAW+9)   .                                  02744
       ENA    (IXUP)      SAL    (DRAW+10)  .                                  02745
       ENA    (IYRIGHT)   SAL    (DRAW+11)  .                                  02746
       ENA    (MODEXAX)   SAL    (DRAW+12)  .                                  02747
       ENA    (MODEYAX)   SAL    (DRAW+13)  .                                  02750
       ENA    (IWIDE)     SAL    (DRAW+14)  .                                  02751
       ENA    (IHIGH)     SAL    (DRAW+15)  .                                  02752
       ENA    (IGRID)     SAL    (DRAW+16)  .                                  02753
       ENA    (LAST)      SAL    (DRAW+17)  .                                  02754
       SLJ 4  (DRAW)      ZRO    (0)        . JUMP TO DRAW                     02755
       ENI 5  (600)       ENA    (0)        . CLEAR ARRAY                      02756
       STA 5  (X)         STA 5  (Y)        .                                  02757
       IJP 5  (L-1)       ENI    (0)        . LOOP                             02760
       LDA    (LAST)      AJP 1  (7S)       . CHECK TO SEE IF GOOD GRAPH       02761
2S1    ENA    (3)         ENI    (0)        . IS MOD = 3                       02762
       EQS    (MOD)       SLJ    (L+2)      .                                  02763
3S     SLJ    (1END)      ZRO    (0)        .   YES JUMP TO X-Y PLOT           02764
       ENA    (2)         STA    (MOD)      . SET MOD = 2                      02765
       ENA    (0)         STA    (NPTS)     . SET NPTS TO ZERO                 02766
       LDA    (JTIME)     SUB    (TFLAG)    . MORE TIME                        02767
       AJP    (9S)        AJP 2  (9S)       .   NO JUMP TO REWIND              02770
       ENI 6  (1)         SLJ    (99)       .   YES JUMP TO READ               02771
4S     ENA    (2)         ENI    (0)        . IS NPTS .GT. 2                   02772
       THS    (NPTS)      SLJ    (10S)      . NO - IGNORE GRAPH                02773
       LDA    (JTIME)     SUB    (TFLAG)    .   AND JUMP TO READ               02774
       AJP    (5S)        AJP 2  (5S)       . MORE TIME  NO JUMP               02775
       SLJ    (2S)        ZRO    (0)        .   YES JUMP TO GRAPH              02776
5S     ENA    (15)        ENI    (0)        . CHECK FOR LAST TRACK             02777
       EQS    (NTRACKS)   SLJ    (11S)      . LAST TRACK NO JUMP               03000
```

```
         ENA    (3)         STA    (MOD)      .   YES SET MOD = 3
         SLJ    (2S)        ZRO    (0)        . JUMP TO GRAPH
6S       INI 6 (1)          SLJ    (99)       . INCREASE INDEX  JUMP TO READ
7S       LDA    (LAST)      SUB    (2B)       . LAST = 2
         AJP 1 (2S1)        STA    (NPTS)     .   NO JUMP TO MOD CHECK
         ENA    (1)         STA    (MOD)      . RESET NPTS AND MOD
         ENI 6 (1)          LDA    (JTIME)    .   AND INDEX
         SUB    (TFLAG)     AJP    (L+2)      . MORE TIME
         AJP 2 (L+1)        SLJ    (9)        .   YES JUMP TO REWIND
8S       SLJ    (1END)      ZRO    (0)        .   NO JUMP TO X-Y POSIT PLOT
9S       ENI 6 (1)          SLJ    (9)        . RESET INDEX  JUMP TO REWIND
10S      ENA    (0)         STA    (NPTS)     . RESET NPTS AND INDEX
         ENI 6 (1)          SLJ    (99)       . JUMP TO READ
11S      ENA    (2)         ENI    (0)        . NPTS .GT. TWO
         THS    (NPTS)      SLJ    (9S)       .   NO JUMP TO GRAPH
         SLJ    (2S)        ZRO    (0)        .   YES JUMP TO GRAPH

* STATE ROUTINE

1ST      EXF    (77521B)    SLJ    (L+4)      . TURN OFF STATE LT
2ST      ZRO    (0)         ZRO    (0)        . CAP NUMBER
3ST      ZRO    (0)         ZRO    (0)        . TRACK NUMBER
4ST      ZRO    (0)         ZRO    (0)        . TYPE OF INTERECPTOR
         LDA    (2ST)       SLJ 4 (1Y)        . DETERMINE CAP TRACK NR
         SIL 2 (3ST)        SLJ 4 (1NUM)      . STROE CAP TRACK NR
         ALS    (6)         ADD    (R5)       . ASSEMBLE INITIAL MSG
         SCL    (77B)       STA    (E86)      .
         ENA    (E84)       SLJ 4 (1MSG)      . SEND INITIAL MSG
         LDQ 2 (TRACK)      LDL    (MASK4B)   . DETERMINE MAX WPNS
         ARS    (39)        STA    (4ST)      .
         LIL 2 (4ST)        LDQ 2 (IFIGO)     .
         LDL    (MASK35)    ARS    (27)       .
         STA    (T+9)       ARS    (3)        .
         ENQ    (7B)        ADL    (T+9)      .
         ARS    (1)         STA    (WPNS)     . DIVIDE MAX WPNS BY 2 AND STORE
         LIL 3 (3ST)        LDQ 3 (KFIG)      .
```

```
       LDL    (MASK35)   ARS    (27)      .                                   03042
       STA    (T+9)      ARS    (3)       .                                   03043
       ENQ    (7B)       ADL    (T+9)     .                                   03044
       AJP    (5ST)      ENI    (0)       . AMMO ZERO                         03045
       THS    (WPNS)     SLJ    (6ST)     . AMMO PLUS                         03046
       LDA    (R21)      STA    (E88)     . AMMO MINUS                        03047
       SLJ    (7ST)      ZRO    (0)       .                                   03050
5ST    LDA    (R22)      STA    (E88)     . AMMO ZERO                         03051
       SLJ    (7ST)      ZRO    (0)       .                                   03052
6ST    LDA    (R23)      STA    (E88)     . AMMO PLUS                         03053
7ST    LDA 3  (KFIG)     ARS    (33)      . DETERMINE TIME OF REPORT          03054
       SLJ 4  (1NUM)     ZRO    (0)       .                                   03055
       ADD    (R24)      STA    (E91)     .                                   03056
       LDA    (ITIME)    ENQ    (0)       .                                   03057
       DVI    (60)       ENI 2  (0)       .                                   03060
       THS    (59)       SLJ    (L+2)     . CHECK HOURS                       03061
       SLJ    (L+3)      ZRO    (0)       .                                   03062
       SUB    (60)       INI 2  (100)     .                                   03063
       SLJ    (L-3)      ZRO    (0)       .                                   03064
       ADD    (ZTIME)    INA 2  (0)       .                                   03065
       SLJ 4  (1NUM)     ZRO    (0)       .                                   03066
       ALS    (24)       ADD    (R25)     .                                   03067
       SCL    (77B)      STA    (E94)     .                                   03070
       ENA    (E87)      SLJ 4  (1MSG)    . SEND REPORT                       03071
       LDA    (4ST)      SLJ 4  (1NUM)    . DETERMINE TYPE                    03072
       ADD    (R26)      STA    (E82)     .                                   03073
       ENA    (E82)      SLJ 4  (1MSG)    . SEND TYPE MSG                     03074
       SLJ    (4T)       ZRO    (0)       . RETURN                            03075

* SELECT INTERRUPTS AND INITIALIZING

0T     EXF    (04000B)   EXF    (77105B)  . ALLOW SELECTED INTERRUPTS         03076
       EXF    (77103B)   EXF    (00100B)  . KEYBD1, KEYBD2, AND ARITHMETIC    03077
       ENA    (1A)       SAL    (Z+7B)    . RESET INTERRUPT ADDRESS           03100
       ENA    (400)      STA    (ADD)     . MEMORY ADDRESS DD65 FOR MSG       03101
       ENA    (-377B)    STA    (XP1)     . LEFT SIDE OF PAGE                 03102
```

```
ENA    (377B)       STA    (YP1)       . FIRST LINE OF PRINT                  03103
ENA    (365B)       STA    (YP2)       . SECOND LINE OF PRINT                 03104
ENA    (353B)       STA    (YMSG)      . FIRST LINE OF PRINT FOR MSG          03105
ENA    (44B)        STA    (MM)        . MEMORY INDICATOR                     03106
ENA    (0)          STA    (IFLAG)     . SET I-FLAG                    =0     03107
STA    (NEG)        STA    (DFLAG)     . SET NEG AND D-FLAG            =0     03110
STA    (RAW)        STA    (PFLAG)     . SET RAW AND P-FLAG            =0     03111
STA    (HSUB)       STA    (HSUR)      . SET HSUB AND HSUR             =0     03112
STA    (HAC)        STA    (FAC)       . SET HAC AND FAC               =0     03113
STA    (FSUB)       STA    (FSUR)      . SET FSUB AND FSUR             =0     03114
STA    (FTIME)      ENA    (1)         . SET SUBTRACT TIME             =0     03115
STA    (KNR2)       STA    (PRO)       . SET KNR2 AND PRO              =1     03116
STA    (KNR1)       STA    (KNR)       . SET KNR1 AND KNR              =1     03117
ENA    (20)         STA    (NPROG)     . NUMBER OF CALLABLE PROGRAMS          03120
ENA    (6)          STA    (MK)        . NUMBER OF EXECUTIVE ROUTINES         03121
ENA    (1800)       STA    (MTIME)     . SET MAX TIME FOR GRAPH               03122
ENA    (0)          STA    (COUNTER)   . DUTY CYCLE CARD                      03123
LDA    (R27)        STA    (GTIME)     . INITIALIZE GTIME                     03124
ENI 4  (15)         ENA    (52B)       . SET ALL TRACKS TO NON ACTIVE         03125
STA 4  (TRACK)      IJP 4  (L)         . LOOP                                 03126
READ 102, JTITLE                                                              03127
LDA    (JTITLE)     STA    (TITLE7)    . SET DATE IN GRAPH TITLE              03130
STA    (TITLE19)    ENI    (0)         .                                      03131
READ 101,LABLE,XSCALE,YSCALE,IXUP, IYRIGHT,MODEXAX,MODEYAX,IWIDE,             03132
*        IHIGH,IGRID
READ 101,LABLE,XSCALE1,YSCALE1,IXUP1,IYRT1,MODXAX1,MODYAX1,IWIDE1,            03133
*        IHIGH1,IGRID
READ 103, SCALE                                                               03134
READ 104, IP,IPP                                                              03135
ENA    (1)          STA    (IETT+1)    . INITIALIZE --                        03136
ENA    (1)          STA    (IEFT+1)    .   EXECUTIVE                          03137
ENA    (600)        STA    (IEFT+5)    .     TABLES                           03140
LDA    (IP)         STA    (IEFT+2)    .       .                              03141
STA    (IETT+2)     STA    (IETT+3)    .         .                            03142
STA    (IETT+4)     STA    (IETT+6)    .           .                          03143
INA    (1)          STA    (IEFT+6)    .             .                        03144
```

```
       INA    (1)          STA    (IEFT+4)   .                                  03145
       INA    (1)          STA    (IEFT+3)   .                                  03146
       ENA    (600)        STA    (IETT+5)   .                                  03147
       ENA    (1)          STA    (PPI)      . SET POINT PLOT INDEX             03150
       LDA    (IPP)        AJP 1  (L+2)      .                                  03151
       ENA    (7G2)        SAL    (7G1)      .                                  03152
       SAU    (7G11)       ENI    (0)        .                                  03153
       LDA    (IEFT+2)     RAD    (IPP)                                         03154

* TIME ROUTINE  EXECUTIVE PROGRAM

       EXF    (02000B)     EXF    (77602B)   . STOP REAL TIME CLOCK AND LT 5A   03155
1T     ENA    (0)          STA    (ITIME)    . RESET JOB TIME CLOCK             03156
2T     ENA    (1)          STA    (NK)       . SET PROGRAM INDEX                03157
       LDA    (ITIME)      STA    (TIME)     . TIME = REAL TIME                 03160
       AJP    (2K)         RAO    (COUNTER)  . IF TIME = 0, JUMP TO 2K          03161
3T     LIL 4  (NK)         LDA    (TIME)     . IF(TIME-IEFT(NK)) IS NEG         03162
       SUB 4  (IEFT)       AJP 3  (4T)       .   JUMP TO 4T                     03163
       LDA 4  (IETT)       ADD    (TIME)     . IEFT(NK)=IETT(NK) + TIME         03164
       STA 4  (IEFT)       LDA    (NK)       . GO TO(1K,1TG,1F,1G,1GR,1INT)NK   03165
       INA    (-1)         AJP    (1K)       . KEYBOARD                         03166
       INA    (-1)         AJP    (1TG)      . TRACK GENERATOR                  03167
       INA    (-1)         AJP    (1F)       . DISPLAY                          03170
       INA    (-1)         AJP    (1G)       . RADAR PROCESSOR                  03171
       INA    (-1)         AJP    (1GR)      . 10 MIN GRAPH ROUTINE             03172
       INA    (-1)         AJP    (1INT)     . INTERCEPT ROUTINE                03173
4T     LDA    (MK)         SUB    (NK)       . IF(MK-NK).EQ..LT. ZERO           03174
       AJP    (2T)         AJP 3  (2T)       .   JUMP TO 2T                     03175
       RAO    (NK)         SLJ    (3T)       . NK = NK + 1, JUMP TO 3T          03176

* TRANSFER TABLE

OTAB   ZRO    (5)          ZRO    (P0)       . BEGIN                            03177
       ZRO    (6)          ZRO    (P1)       . INTCPT (INTERCEPT)               03200
       ZRO    (6)          ZRO    (P2)       . LAUNCH                           03201
       ZRO    (7)          ZRO    (P3)       . HISTORY                          03202
```

```
        ZRO   (5)          ZRO   (P4)       . ERROR                     03203
        ZRO   (4)          ZRO   (P5)       . HOLD                      03204
        ZRO   (4)          ZRO   (P6)       . SKIP                      03205
        ZRO   (7)          ZRO   (P7)       . DISPLAY                   03206
        ZRO   (4)          ZRO   (P8)       . DROP                      03207
        ZRO   (8)          ZRO   (P9)       . NOT USED                  03210
        ZRO   (4)          ZRO   (P10)      . STOP                      03211
        ZRO   (5)          ZRO   (P11)      . STEER                     03212
        ZRO   (5)          ZRO   (P12)      . DESIG                     03213
        ZRO   (4)          ZRO   (P13)      . INFO                      03214
        ZRO   (8)          ZRO   (P14)      . NOT USED                  03215
        ZRO   (5)          ZRO   (P15)      . GRAPH                     03216
        ZRO   (7)          ZRO   (P16)      . STATION                   03217
        ZRO   (6)          ZRO   (P17)      . CHANGE                    03220
        ZRO   (5)          ZRO   (P18)      . STATE                     03221
        ZRO   (8)          ZRO   (P19)      . NOT USED                  03222
        ZRO   (8)          ZRO   (P20)      . READY LT                  03223
        ZRO   (4)          ZRO   (P21)      . FADE                      03224
        ZRO   (6)          ZRO   (P22)      . UNFADE                    03225
        ZRO   (5)          ZRO   (P23)      . START                     03226
        ZRO   (5)          ZRO   (P24)      . SCRUB                     03227
        ZRO   (4)          ZRO   (P25)      . PORT                      03230
        ZRO   (4)          ZRO   (P26)      . STBD                      03231
        ZRO   (5)          ZRO   (P27)      . SPEED                     03232
        ZRO   (8)          ZRO   (P28)      . ALTITUDE                  03233
        ZRO   (8)          ZRO   (P29)      . NOT USED                  03234

* TRACK GENERATOR CALLING ROUTINE

1TG     SIU 4 (3TG)        SIL 5 (3TG)      . SAVE INDEXES              03235
        SLJ 4 (TRAKGEN)    ZRO   (0)        . CALL TRAKGEN              03236
        ENI 4 (1)          ENI   (0)        .                           03237
2TG     LDQ 4 (IPOSIT)     LDL   (MASKX)    .                           03240
        ARS   (24)         SLJ 4 (FLOAT)    . UNPACK IPOSIT             03241
        FDV   (SCALE1)     STA 4 (XN)       . UNSCALE X-POSITION        03242
        LDQ 4 (IPOSIT)     ENI   (0)        . STORE IN XN               03243
```

```
       LDL    (MASKY)      ALS    (24)       . UNSCALE Y-POSITION           03244
       ARS    (24)         SLJ 4 (FLOAT)     . STORE IN YN                  03245
       FDV    (SCALE1)     STA 4 (YN)        .                              03246
       LDQ 4 (JHEAD)       ENI    (0)        . UNPACK JHEAD                 03247
       LDL    (MASKY)      ALS    (24)       . UNSCALE Z-POSITION           03250
       ARS    (24)         SLJ 4 (FLOAT)     . STORE IN ZN                  03251
       FDV    (SCALE2)     STA 4 (ZN)        .                              03252
       ISK 4 (15)          SLJ    (2TG)      . LOOP                         03253
       LDA    (TIME)       STA    (KTIME)    .                              03254
       WRITE TAPE MM, XN,YN,ZN,TIME                                         03255

* ADD NOISE ROUTINE HERE OR BEFORE WRITE TAPE

3TG    ENI 4 (N)           ENI 5 (N)         . RESTORE INDEXES              03256
       SLJ    (4T)         ZRO    (0)        . RETURN                       03257

* INFO ROUTINE

1TT    EXF    (77421B)     SLJ    (L+2)      . TURN OFF INFO LT             03260
2TT    ZRO    (0)          ZRO    (0)        . TRACK FOR INFO               03261
       ENA    (2TT)        SLJ 4 (3J0)       . HOOK TRACK                   03262
       ENA    (E73)        SLJ 4 (1MSG)      . OUTPUT TITLE                 03263
       LDA    (2TT)        SLJ 4 (1NUM)      .                              03264
       ALS    (36)         SCL    (MASKY)    .                              03265
       STA    (E77)        ENA    (E75)      .                              03266
3TT    SLJ 4 (1MSG)        ZRO    (0)        . OUTPUT TRACK NUMBER          03267
       LDA    (2TT)        ENI    (0)        . CHECK FOR OWN SHIP           03270
       EQS    (OWN)        SLJ    (L+2)      .                              03271
       SLJ    (14TT)       ZRO    (0)        . OWN SHIP JUMP                03272
       LIL 3 (2TT)         LDA 3 (TRACK)     .                              03273
       STA    (2TT)        ENI    (0)        .                              03274
       SSH    (2TT)        SLJ    (9TT)      . JUMP ON HOSTILE              03275
       SSH    (2TT)        SLJ    (5TT)      . JUMP ON NOT SURFACE          03276
       LDA 3 (TRACK)       STA    (2TT)      . SURFACE TRACK                03277
       LDA    (R12)        STA    (E79)      .                              03300
       LDQ    (2TT)        QRS    (24)       .                              03301
```

```
      LDL   (777B)     SLJ 4 (1NUM)     .                       03302
      SAL   (3TT)      LDQ   (3TT)      .                       03303
      LDL   (77B)      ADD   (R18)      .                       03304
      STA   (E80)      ENA   (0)        .                       03305
      STA   (E81)      ENA   (E78)      .                       03306
      SLJ 4 (1MSG)     ZRO   (0)        .                       03307
4TT   LDA 3 (CRS)      SLJ 4 (FIX)      . SURFACE INFO          03310
      SLJ 4 (1NUM)     ZRO   (0)        .                       03311
      ALS   (6)        ADD   (R7)       .                       03312
      STA   (E79)      LDA 3 (SPD)      .                       03313
      SLJ 4 (FIX)      ZRO   (0)        .                       03314
      SLJ 4 (1NUM)     ZRO   (0)        .                       03315
      LRS   (6)        ADD   (MASK17)   .                       03316
      ADD   (R8)       STA   (E80)      .                       03317
      ENA   (0)        LLS   (6)        .                       03320
      ALS   (42)       ADD   (R15)      .                       03321
      STA   (E81)      ENA   (0)        .                       03322
      STA   (E82)      ENA   (E78)      .                       03323
      SLJ   (12TT)     ZRO   (0)        . OUTPUT SURFACE INFO   03324
5TT   SSH   (2TT)      SLJ   (7TT)      . JUMP ON NO A/C        03325
      LDA 3 (TRACK)    STA   (2TT)      . A/C TRACK             03326
      LDQ   (2TT)      QRS   (24)       .                       03327
      LDL   (777B)     SLJ 4 (1NUM)     .                       03330
      ADD   (R5)       STA   (E79)      .                       03331
      ENA   (0)        STA   (E80)      .                       03332
      ENA   (E78)      SLJ 4 (1MSG)     .                       03333
6TT   LDA 3 (CRS)      SLJ 4 (FIX)      . A/C INFO              03334
      SLJ 4 (1NUM)     ZRO   (0)        .                       03335
      ALS   (6)        ADD   (R7)       .                       03336
      STA   (E79)      LDA 3 (SPD)      .                       03337
      SLJ 4 (FIX)      ZRO   (0)        .                       03340
      SLJ 4 (1NUM)     ZRO   (0)        .                       03341
      LRS   (6)        ADD   (MASK17)   .                       03342
      ADD   (R8)       STA   (E80)      .                       03343
      ENA   (0)        LLS   (6)        .                       03344
      ALS   (42)       ADD   (R9)       .                       03345
```

```
        STA   (E81)     LDA 3 (ZN)      .                                  03346
        SLJ 4 (FIX)     ZRO   (0)       .                                  03347
        SLJ 4 (1NUM)    ZRO   (0)       .                                  03350
        ALS   (12)      ADD   (5300B)   .                                  03351
        SCL   (77B)     STA   (E82)     .                                  03352
        ENA   (0)       STA   (E83)     .                                  03353
        ENA   (E78)     SLJ   (12TT)    . OUTPUT A/C INFO                  03354
7TT     SSH   (2TT)     SLJ   (13TT)    . JUMP IF NON DESIGNATED TRACK     03355
        LDA 3 (TRACK)   STA   (2TT)     . SUBMARINE TRACK                  03356
        LDQ   (2TT)     QRS   (24)      .                                  03357
        LDL   (777B)    SLJ 4 (1NUM)    .                                  03360
        SCL   (MASK10)  ADD   (R13)     .                                  03361
        STA   (E80)     LDA   (R12)     .                                  03362
        STA   (E79)     ENA   (0)       .                                  03363
        STA   (E81)     ENA   (E78)     .                                  03364
        SLJ 4 (1MSG)    ZRO   (0)       .                                  03365
8TT     LDA 3 (CRS)     SLJ 4 (FIX)     . SUBMARINE INFO                   03366
        SLJ 4 (1NUM)    ZRO   (0)       .                                  03367
        ALS   (6)       ADD   (R7)      .                                  03370
        STA   (E79)     LDA 3 (SPD)     .                                  03371
        SLJ 4 (FIX)     ZRO   (0)       .                                  03372
        SLJ 4 (1NUM)    ZRO   (0)       .                                  03373
        LRS   (6)       ADD   (MASK17)  .                                  03374
        ADD   (R8)      STA   (E80)     .                                  03375
        ENA   (0)       LLS   (6)       .                                  03376
        ALS   (42)      ADD   (R14)     .                                  03377
        STA   (E81)     LAC 3 (ZN)      .                                  03400
        SLJ 4 (FIX)     ZRO   (0)       .                                  03401
        SLJ 4 (1NUM)    ZRO   (0)       .                                  03402
        ALS   (24)      ADD   (R10)     .                                  03403
        SCL   (77B)     STA   (E82)     .                                  03404
        ENA   (0)       STA   (E83)     .                                  03405
        ENA   (E78)     SLJ   (12TT)    . OUTPUT SUBMARINE INFO            03406
9TT     SSH   (2TT)     SLJ   (10TT)    . JUMP ON NOT SURFACE              03407
        LDA 3 (TRACK)   STA   (2TT)     . HOSTILE SURFACE TRACK            03410
        LIU 5 (2TT)     LDA 5 (F11)     .                                  03411
```

```
      SUB   (20B)         ADD   (E78)      .                                  03412
      ADD   (R16)         STA   (E79)      .                                  03413
      ENA   (0)           STA   (E80)      .                                  03414
      ENA   (E78)         SLJ 4 (1MSG)     .                                  03415
      SLJ   (4TT)         ZRO   (0)        . JUMP TO SURFACE INFO             03416
10TT  SSH   (2TT)         SLJ   (11TT)     . JUMP ON NOT A/C                  03417
      LDA 3 (TRACK)       STA   (2TT)      . HOSTILE A/C TRACK                03420
      LDQ   (2TT)         QRS   (24)       .                                  03421
      LDL   (777B)        SLJ 4 (1NUM)     .                                  03422
      ALS   (36)          SCL   (77B)      .                                  03423
      STA   (E80)         LDA   (R11)      .                                  03424
      STA   (E79)         ENA   (0)        .                                  03425
      STA   (E81)         ENA   (E78)      .                                  03426
      SLJ 4 (1MSG)        ZRO   (0)        .                                  03427
      SLJ   (6TT)         ZRO   (0)        . JUMP-TP A/C INFO                 03430
11TT  SSH   (2TT)         SLJ   (13TT)     . JUMP ON NON DESIGNATED TRACK     03431
      LDA 3 (TRACK)       STA   (2TT)      . HOSTILE SUBMARINE TRACK          03432
      LIU 5 (2TT)         LDA 5 (F11)      .                                  03433
      SUB   (20B)         ADD   (E78)      .                                  03434
      ADD   (R17)         STA   (E79)      .                                  03435
      ENA   (0)           STA   (E80)      .                                  03436
      ENA   (E78)         SLJ 4 (1MSG)     .                                  03437
      SLJ   (8TT)         ZRO   (0)        . JUMP TO SUBMARINE INFO           03440
12TT  SLJ 4 (1MSG)        ZRO   (0)        . OUTPUT ALL MSG ROUTINE           03441
      LDA 3 (ZN)          STA   (ZTAR)     .                                  03442
      LDA 3 (XN)          STA   (XTAR)     .                                  03443
      LDA 3 (YN)          STA   (YTAR)     .                                  03444
      LIL 3 (OWN)         LAC 3 (XN)       .                                  03445
      FAD   (XTAR)        STA   (XTAR)     .                                  03446
      LAC 3 (YN)          FAD   (YTAR)     .                                  03447
      STA   (YTAR)        ENA   (XTAR)     .                                  03450
      SAL   (COURSE+1)    ENA   (YTAR)     .                                  03451
      SAL   (COURSE+2)    ENA   (DTAR)     .                                  03452
      SAL   (COURSE+3)    ENA   (S6C)      .                                  03453
      SAL   (COURSE+5)    ENA   (RTAR)     .                                  03454
      SAL   (COURSE+6)    SLJ 4 (COURSE)   . JUMP TO COURSE                   03455
```

```
         LDA    (DTAR)     AJP    (4T)      . RETURN OWN SHIP                03456
         LDA    (E96)      STA    (E79)     . FORM RANGE MSG                 03457
         LDA    (ZTAR)     AJP    (L+2)     .                                03460
         AJP 2  (L+6)      ENI    (0)       .                                03461
         LDA    (RTAR)     FMU    (2000.)   .                                03462
         SLJ 4  (FIX)      ZRO    (0)       .                                03463
         SLJ 4  (1NUM)     ZRO    (0)       .                                03464
         STA    (E80)      LDA    (E105)    .                                03465
         SLJ    (L+4)      ZRO    (0)       .                                03466
         LDA    (RTAR)     SLJ 4  (FIX)     .                                03467
         SLJ 4  (1NUM)     ZRO    (0)       .                                03470
         STA    (E80)      LDA    (E97)     .                                03471
         STA    (E81)      ENA    (E78)     .                                03472
         SLJ 4  (1MSG)     ZRO    (0)       . OUTPUT RANGE MSG               03473
         LDA    (E98)      STA    (E79)     .                                03474
         LDA    (DTAR)     SLJ 4  (FIX)     .                                03475
         SLJ 4  (1NUM)     ZRO    (0)       .                                03476
         STA    (E80)      LDA    (E99)     .                                03477
         STA    (E81)      ENA    (0)       .                                03500
         STA    (E82)      ENA    (E78)     .                                03501
         SLJ 4  (1MSG)     ZRO    (0)       . OUTPUT BEARING MESSAGE         03502
         LDA    (OWN)      AJP 1  (L+6)     . JUMP IF OWN SHIP REFERENCE     03503
         LDA    (E102)     STA    (E79)     .                                03504
         LDA    (E103)     STA    (E80)     .                                03505
         LDA    (E104)     STA    (E81)     .                                03506
         ENA    (E78)      SLJ 4  (1MSG)    . OUTPUT REFERENCE MESSAGE       03507
         SLJ    (4T)       ZRO    (0)       . RETURN                         03510
         LDA    (E100)     STA    (E79)     .                                03511
         LDA    (E101)     STA    (E80)     .                                03512
         SLJ    (L-4)      ZRO    (0)       .                                03513
13TT     LDA    (R19)      STA    (E79)     . UNIDENTIFIED TRACK             03514
         LDA    (R20)      STA    (E80)     .                                03515
         ENA    (0)        STA    (E81)     .                                03516
         ENA    (E78)      SLJ 4  (1MSG)    .                                03517
         LDA 3  (ZN)       AJP    (4TT)     . JUMP ACCORDING TO ALTITUDE     03520
         AJP 2  (6TT)      AJP 3  (8TT)     .                                03521
```

```
14TT   ENA    (E95)      STA    (E79)      .                                    03522
       ENA    (0)        STA    (E80)      .                                    03523
       SLJ 4  (1MSG)     ZRO    (0)        . OUTPUT OWN SHIP INFO               03524
       SLJ    (4TT)      ZRO    (0)        . JUMP TO SURFACE INFO               03525

*  STEER COMMAND ROUTINE

1U     EXF    (77405B)   SLJ    (L+2)      . TURN OFF STEER LT                  03526
2U     ZRO    (0)        ZRO    (0)        . CAP NUMBER                         03527
       LDA    (2U)       SLJ 4  (1Y)       . DETERMINE CAP TRACK NR             03530
       LDA    (MASK22)   RAD 2  (TRACK)    . SET STEER BIT                      03531
       SLJ    (4T)       ZRO    (0)        . RETURN                             03532

*  STATION COMMAND ROUTINE

1V     EXF    (77505B)   SLJ    (L+4)      . TURN OFF STATION LT                03533
2V     ZRO    (0)        ZRO    (0)        . CAP NUMBER                         03534
3V     ZRO    (0)        ZRO    (0)        . RANGE FROM CARRIER                 03535
4V     ZRO    (0)        ZRO    (0)        . BEARING DEGREES TRUE               03536
       ENA    (2V)       SLJ 4  (1ZZ)      . DETERMINE CAP TRACK NR             03537
       LDA    (MASK20)   RAD 2  (TRACK)    . SET STATION BIT                    03540
       SLJ    (4T)       ZRO    (0)        . RETURN                             03541

*  SKIP INTERCEPT COMMAND ROUTINE

1W     EXF    (77205B)   SLJ    (L+2)      . TURN OFF INTCPT LT                 03542
2W     ZRO    (0)        ZRO    (0)        . CAP TO SKIP                        03543
       LDA    (2W)       SLJ 4  (1Y)       . DETERMINE CAP TRACK NR             03544
       LDQ    (MASK19)   LDL 2  (TRACK)    . IS CAP MAKING AN INTERCEPT         03545
       AJP    (3W)       LDA    (MASK21)   .   NO JUP -- YES SET SKIP BIT       03546
       RAD 2  (TRACK)    SLJ    (4T)       . RETURN                             03547
3W     LDA    (2W)       SLJ 4  (1NUM)     . SEND MSG                           03550
       STA    (T+1)      LDA    (R5)       .   CAP  XX NOT                      03551
       RAD    (T+1)      STA    (E63)      .     MAKING                         03552
       ENA    (E63)      SLJ 4  (1MSG)     .       INTERECPT                    03553
       SLJ    (4T)       ZRO    (0)        . RETURN                             03554
```

```
* LAUNCH COMMAND ROUTINE

1X    EXF    (77211B)    SLJ    (L+5)      . TURN OFF LAUNCH LT                03555
2X    ZRO    (0)         ZRO    (0)        . CAP NUMBER                        03556
3X    ZRO    (0)         ZRO    (0)        . RANGE                             03557
4X    ZRO    (0)         ZRO    (0)        . BEARING DEGREES TRUE              03560
5X    ZRO    (0)         ZRO    (0)        . TYPE OF INTERCEPTOR               03561
      ENI 2  (16)        LDQ    (MASK11)   .                                   03562
      ENA    (52B)       ENI    (0)        .                                   03563
      MEQ 2  (TRACK)     SLJ    (5Q)       . SEARCH FOR NON ACTIVE TRACK       03564
      LDA    (MASK9L)    STA 2  (TRACK)    . SET CAP BITS                      03565
      LDA    (MASK20)    RAD 2  (TRACK)    .                                   03566
      ENA    (2X)        SLJ 4  (1X1)      . GET CAP NUMBER                    03567
      LDA    (2X)        ALS    (24)       .                                   03570
      RAD 2  (TRACK)     ENI    (0)        . SET CAP NUMBER                    03571
      ENA    (2X)        SLJ 4  (1ZZ)      . DETERMINE STATION INFO            03572
      LDA    (5X)        AJP    (7X)       . JUMP IF NO TYPE GIVEN             03573
      INA    (-1)        AJP    (7X)       . INITIALIZE                        03574
      INA    (1)         ENI    (0)        .   INTERCEPTOR TYPE                03575
      ALS    (39)        RAD 2  (TRACK)    .     FUEL                          03576
      LIL 5  (5X)        LDA 5  (IFIG0)    .       AND                         03577
6X    STA 2  (KFIG)      LDA    (ITIME)    .         LAUNCH TIME               03600
      RAD 2  (KFIG)      SLJ    (4T)       .                                   03601
7X    LDA    (MASK4B)    RAD 2  (TRACK)    . MAKE INTECPTOR F4                 03602
      LDA    (IFIG7)     SLJ    (6X)       .                                   03603

* DETERMINE FRIENDLY A/C NUMBER ROUTINE

1X1   SLJ    (N)         STA    (T+8)      . EXIT/ENTRY                        03604
      LDA 7  (T+8)       ENI    (0)        . CHECK TO SEE IF NUMBER .GT.       03605
      THS    (FAC)       SLJ    (L+3)      .   FAC  YES JUMP  NO USE FAC       03606
      RAO    (FAC)       STA 7  (T+8)      .                                   03607
      SLJ    (1X1)       ZRO    (0)        . RETURN                            03610
      STA    (FAC)       SLJ    (1X1)      . SET NEW FAC   RETURN              03611
```

```
* FIND CAP NUMBER ROUTINE

1Y     SLJ    (N)           STA    (2Y)        . EXIT/ENTRY                           03612
       ENI 2  (1)           ENI    (0)         .                                      03613
       SSK 2  (TRACK)       SLJ    (L+7)       . CHECK FOR FRIENDLY                   03614
       LDQ 2  (TRACK)       LDA    (MASK16)    . CHECK FOR A/C BIT                    03615
       MEQ    (MASK16)      SLJ    (L+5)       .                                      03616
       LDQ 2  (TRACK)       QRS    (24)        .                                      03617
       LDL    (77B)         ENI    (0)         .                                      03620
       EQS    (2Y)          SLJ    (L+2)       . CHECK FOR CAP NUMBER                 03621
       SLJ    (1Y)          ZRO    (0)         . RETURN WITH TRACK NUMBER IN B2       03622
       ISK 2  (15)          SLJ    (L-7)       . LOOP                                 03623
       LDA    (2Y)          SLJ 4  (1NUM)      . ERROR EXIT                           03624
       ALS    (6)           STA    (T+1)       . CAP  XX NOT AIRBORNE                 03625
       LDA    (R5)          RAD    (T+1)       .   MSG TO BE SENT                     03626
       STA    (E60)         ENA    (E60)       .                                      03627
       SLJ 4  (1MSG)        ZRO    (0)         .                                      03630
       SLJ    (4T)          ZRO    (0)         . RETURN                               03631
2Y     ZRO    (0)           ZRO    (0)         . CAP NUMBER                           03632

* PROGRAM ARGUMENT AND ADDRESS TABLE

0Z     ZRO    (0)           ZRO    (1Z)        . BEGIN                                03633
       ZRO    (1)           ZRO    (1Q)        . INTCPT (INTERCEPT)                   03634
       ZRO    (4)           ZRO    (1X)        . LAUNCH                               03635
       ZRO    (2)           ZRO    (1L)        . HISTORY                              03636
       ZRO    (0)           ZRO    (4T)        . ERROR                                03637
       ZRO    (0)           ZRO    (1PQ)       . HOLD/PAUSE                           03640
       ZRO    (1)           ZRO    (1W)        . SKIP                                 03641
       ZRO    (0)           ZRO    (1D)        . DISPLAY                              03642
       ZRO    (0)           ZRO    (1R)        . DROP                                 03643
       ZRO    (0)           ZRO    (2P)        . NOT USED                             03644
       ZRO    (0)           ZRO    (1S)        . STOP                                 03645
       ZRO    (1)           ZRO    (1U)        . STEER                                03646
       ZRO    (2)           ZRO    (1J)        . DESIG                                03647
       ZRO    (0)           ZRO    (1TT)       . INFO                                 03650
```

```
        ZRO   (0)        ZRO   (2P)       . NOT USED                           03651
        ZRO   (0)        ZRO   (3GR)      . GRAPH                              03652
        ZRO   (3)        ZRO   (1V)       . STATION                            03653
        ZRO   (7)        ZRO   (1C)       . CHANGE                             03654
        ZRO   (1)        ZRO   (1ST)      . STATE                              03655
        ZRO   (0)        ZRO   (2P)       . NOT USED                           03656
        ZRO   (0)        ZRO   (4T)       . READY                              03657
0Z10    ZRO   (0)        ZRO   (11C)      . FADE                               03660
        ZRO   (0)        ZRO   (11C)      . UNFADE                             03661
        ZRO   (0)        ZRO   (10C)      . START                              03662
        ZRO   (0)        ZRO   (11C)      . SCRUB                              03663
        ZRO   (0)        ZRO   (13C)      . PORT                               03664
        ZRO   (0)        ZRO   (13C)      . STBD                               03665
        ZRO   (0)        ZRO   (12C)      . SPEED                              03666
        ZRO   (0)        ZRO   (15C)      . ALTITUDE                           03667
        ZRO   (0)        ZRO   (2P)       . NOT USED                           03670

* BEGIN ROUTINE

1Z      EXF   (77203B)   EXF   (77603B)   . TURN OFF BEGIN AND READY LTS       03671
        ENA   (1)        STA   (IFLAG)    . SET IFLAG                          03672
        LDA   (CLK)      STA   (Z)        . RESET CLOCK                        03673
        EXF   (1000B)    SLJ 4 (TRAKGEN)  . INITIAL CALL OF TRAKGEN            03674
        AJP 3 (2Z)       ENI   (0)        . NTRACKS GREATER THAN 16 JUMP       03675
        STA   (T+11)     LIL 4 (T+11)     .                                    03676
        LDA 4 (OBO)      ALS   (36)       .                                    03677
        STA   (R)        LDA   (E9)       .                                    03700
        RAD   (R)        LDA   (E10)      .   INITIAL                          03701
        STA   (R+1)      LDA   (E11)      .     MESSAGE                        03702
        STA   (R+2)      LDA   (E12)      .                                    03703
        STA   (R+3)      ENA   (R)        .                                    03704
        SAL   (2E)       SAU   (6E)       .                                    03705
        ENI 5 (1)        SLJ   (2E)       .                                    03706
2Z      ENA   (XP1)      SAL   (PRINT+1)  . ERRO MSG -- NTRACKS .GT. OR        03707
        ENA   (YP2)      SAL   (PRINT+2)  .   .EQ. 16  CHECK DATA DECK         03710
```

```
        ENA    (1)          SAL    (PRINT+3) .     TRY AGAIN               03711
        ENA    (E50)        ENQ    (03)      .                             03712
        SLJ 4  (PRINT)      ZRO    (0)       .                             03713
        SLS    (0T)         ZRO    (0)       . RESTART                     03714

*  CAP NUMBER AND STATION INFORMATION ROUTINE

1ZZ     SLJ    (N)          SLJ    (L+4)     . EXIT/ENTRY                  03715
2ZZ     ZRO    (0)          ZRO    (0)       . CAP NUMBER                  03716
3ZZ     ZRO    (0)          ZRO    (0)       . RANGE FROM CARRIER          03717
4ZZ     ZRO    (0)          ZRO    (0)       . BEARING DEGREES TRUE        03720
        STA    (T+1)        LDA 7  (T+1)     . TRANSFER                    03721
        STA    (2ZZ)        RAO    (T+1)     .   ARGUMENTS                 03722
        LDA 7  (T+1)        STA    (3ZZ)     .     TO THIS ROUTINE         03723
        RAO    (T+1)        LDA 7  (T+1)     .                             03724
        STA    (4ZZ)        LDA    (2ZZ)     . GET CAP TRACK NR            03725
        SLJ 4  (1Y)         ZRO    (0)       .                             03726
        SIL 2  (2ZZ)        LDA    (2ZZ)     .                             03727
        ALS    (42)         SLJ 4  (6Q)      . DETERMINE TRACK1 WORD       03730
        LIL 3  (8Q)         STA    (T+2)     .                             03731
        LDA 3  (TRACK1)     SCL    (MASK36)  .                             03732
        ADD    (T+2)        STA 3  (TRACK1)  . SET UP TRACK1 WORD          03733
        LDA    (3ZZ)        AJP    (5ZZ)     . STATION GIVEN  NO JUMP TO   03734
        SLJ 4  (FLOAT)      ZRO    (0)       .   HOOK STATION              03735
        STA    (3ZZ)        ENI    (0)       . FLOAT                       03736
        LDA    (4ZZ)        SLJ 4  (FLOAT)   .   ARGUMENTS                 03737
        STA    (4ZZ)        LDA    (AXIS)    . CONVERT TO RADIANS          03740
        FSB    (4ZZ)        FMU    (RADIANS) .                             03741
        STA    (4ZZ)        ENA    (4ZZ)     .                             03742
        SAL    (COSF+1)     SAL    (SINF+1)  .                             03743
        SLJ 4  (COSF)       ZRO    (0)       . DETERMINE X AND Y           03744
        FMU    (3ZZ)        SLJ 4  (FIX)     .   POSITION                  03745
        SCL    (-777B)      ALS    (27)      .     OF                      03746
        RAD 3  (TRACK1)     SLJ 4  (SINF)    .       STATION               03747
        FMU    (3ZZ)        SLJ 4  (FIX)     .                             03750
        SCL    (-777B)      ALS    (18)      . SET TRACK1 WORD             03751
```

```
        RAD 3 (TRACK1)    SLJ    (1ZZ)      . RETURN                     03752
5ZZ     LDA   (EIGHT)     FDV    (RHO)      . DETERMINE                  03753
        STA   (INC)       FAD    (XPIP)     .   X AND Y                  03754
        STA   (XDESIG)    LDA    (INC)      .     POSITION               03755
        FAD   (YPIP)      SLJ 4  (FIX)      .      OF                    03756
        SCL   (-777B)     ALS    (25)       .                            03757
        RAD 3 (TRACK1)    LDA    (XPIP)     .                            03760
        SLJ 4 (FIX)       ZRO    (0)        .                            03761
        SCL   (-777B)     ALS    (34)       .                            03762
        RAD 3 (TRACK1)    SLJ    (1ZZ)      .                            03763

        END
```

APPENDIX II-A

1. Identification.

Title: TRAKGEN

Category: Track Generator

Programmer: A. C. Casciato

Organization: U. S. Naval Postgraduate School

Date: April 1965

2. Purpose.

This subroutine is designed to generate fine grain, three dimensional position information in order to simulate radar or sonar track information. The subroutine generates up to 15 tracks simultaneously.

3. Program Sections.

3.1 1RET - Subroutine Entry, Initial Call. This section checks for the initial call and initializes all arrays and flags, reads in the initial position data for preprogrammed tracks, and sets all tracks to standby.

3.2 1TRAK - Process Maneuver Table Entries. This section reads in maneuver command data from the Data Process Section, the Keyboard Command Section, and the Maneuver Table and updates the maneuver code in the track matrix. It selects the proper maneuver section for track processing.

3.3 1WAIT - Utility Section. This section contains several small sections used by other portions of the subroutine.

3.4 1MOVE - Update Tracks. This section unpacks the track matrix information and routes the data through the proper maneuver routines

in the necessary order. After processing, the data is repacked into the track matrix.

3.5 1EXIT - Exit From Subroutine. This section restores the index information and returns control to the calling program.

3.6 1GO - Constant Course Section. This section increments X and Y position information when the track is not in a turn.

3.7 1TURN - Turn Section. This section processes the X and Y position information when the track is in a turn.

3.8 IDIVE - Climb Section. This section increments the altitude information when the track is climbing/diving.

3.9 1SPD - Speed Change Section. This section increments the speed information when the track is changing speed.

3.10 1CT - Multiple Maneuver Section. This section sets the proper flags to route the track information through the necessary maneuver sections when performing more than one maneuver.

3.11 1KEY - Keyboard Command Section. This section processes keyboard command information into a form similar to that of a maneuver table data card.

3.12 7KEY - Data Process Command Section. This section converts tactical data command information into the form of keyboard command information.

4. Limitations on Subroutine Quantities.

4.1 The maximum and minimum values for subroutine TRACKGEN input quantities are:

X and Y position:	$\pm$ 279 miles.
Altitude	$\pm$ 98,689 feet.
Course:	0 to 360 degrees.
Speed:	0 to 4095 knots.
Turn Rate:	0.1 to 25.5 degrees per second.
Climb Rate:	43* to 65,535 feet per minute.
Acceleration:	1 to 255 knots per second.

* The climb rate has a minimum of 43 feet per minute with a calling period of one second. With the six second calling period, a minimum climb rate of eight feet per minute is possible.

5. Rules Governing Track Commands.

5.1 The first command for a preprogrammed track must be a "Start Track" command.

5.2 More than one command may be given to the same track at the same time.

5.3 Normal maneuver commands may be given to a track even if it has been faded.

5.4 Once a track has been maneuvered by the keyboard console or by the data process section, further preprogrammed commands are ignored. This does not apply to the keyboard "Fade" or "Unfade" commands.

5.5 A track which has been scrubbed by any means cannot be restarted from the preprogrammed data deck.

5.6 Any maneuver may be continued to a new assigned limit by inserting a maneuver command with the new limit. The previous maneuver does not have to be completed before ordering the new limit.

APPENDIX II-B

SUBROUTINE TRAKGEN FLOW DIAGRAMS

Subroutine Entry and Exit

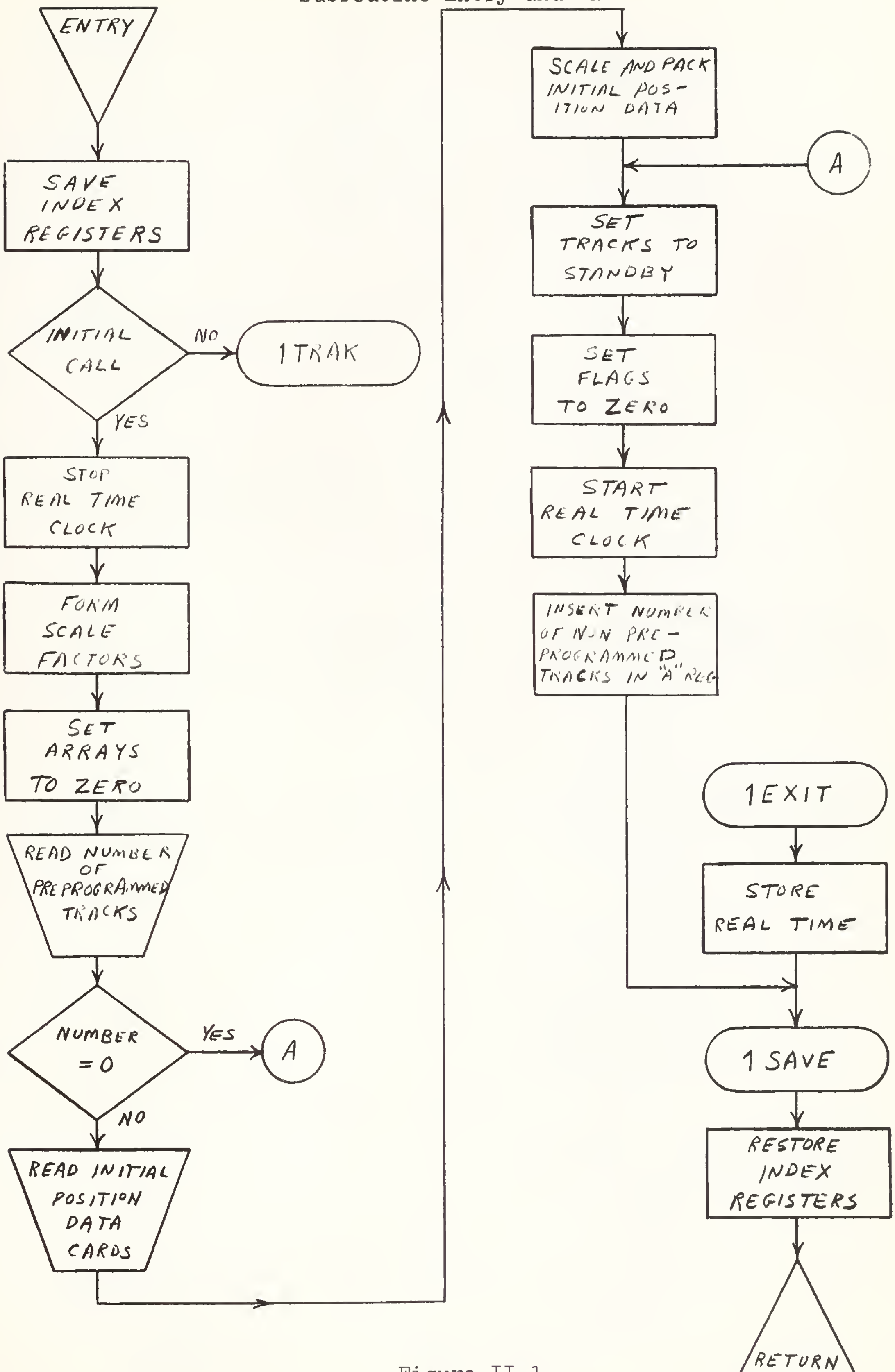


Figure II-1

Update Maneuver Codes

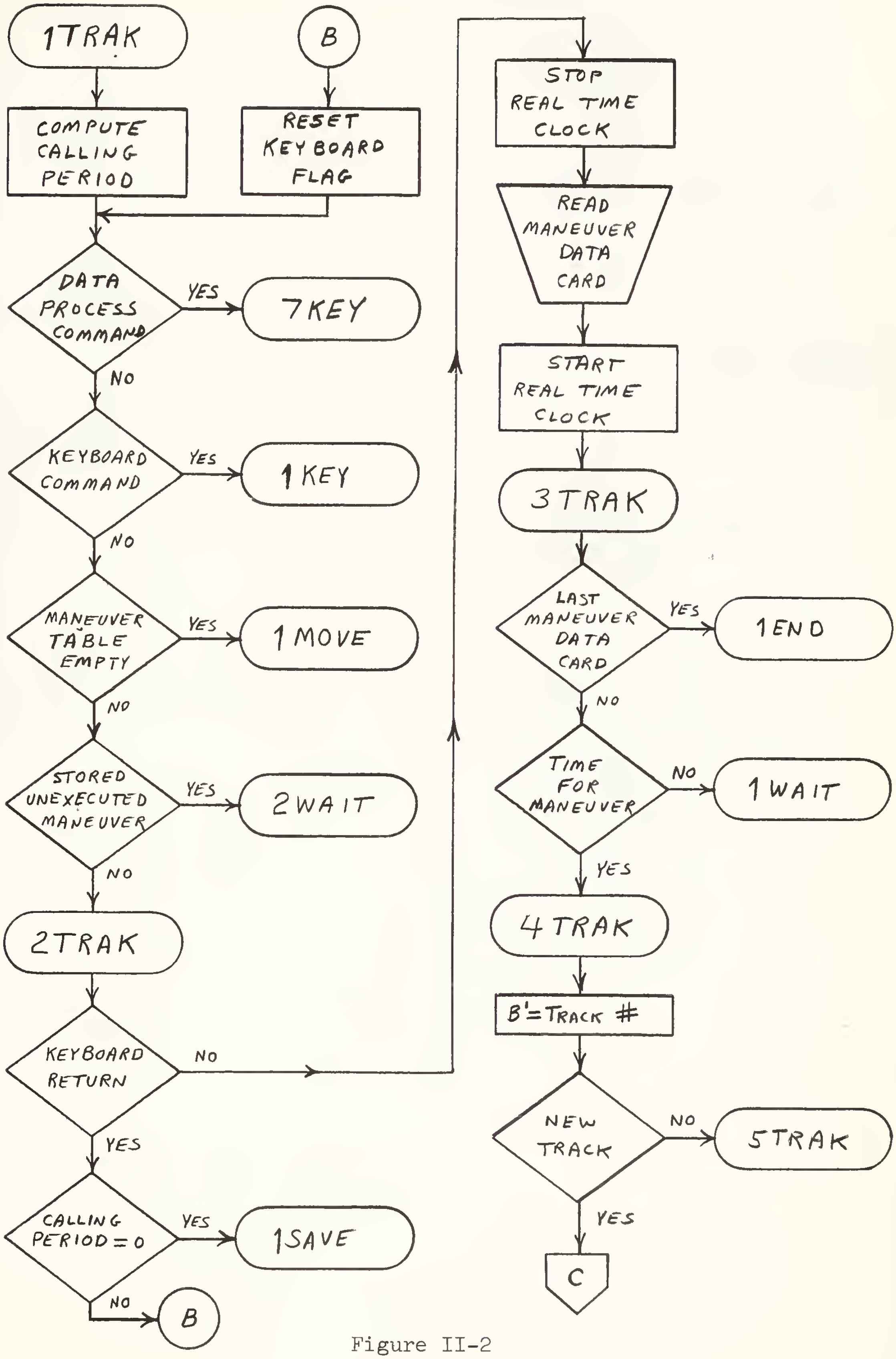


Figure II-2

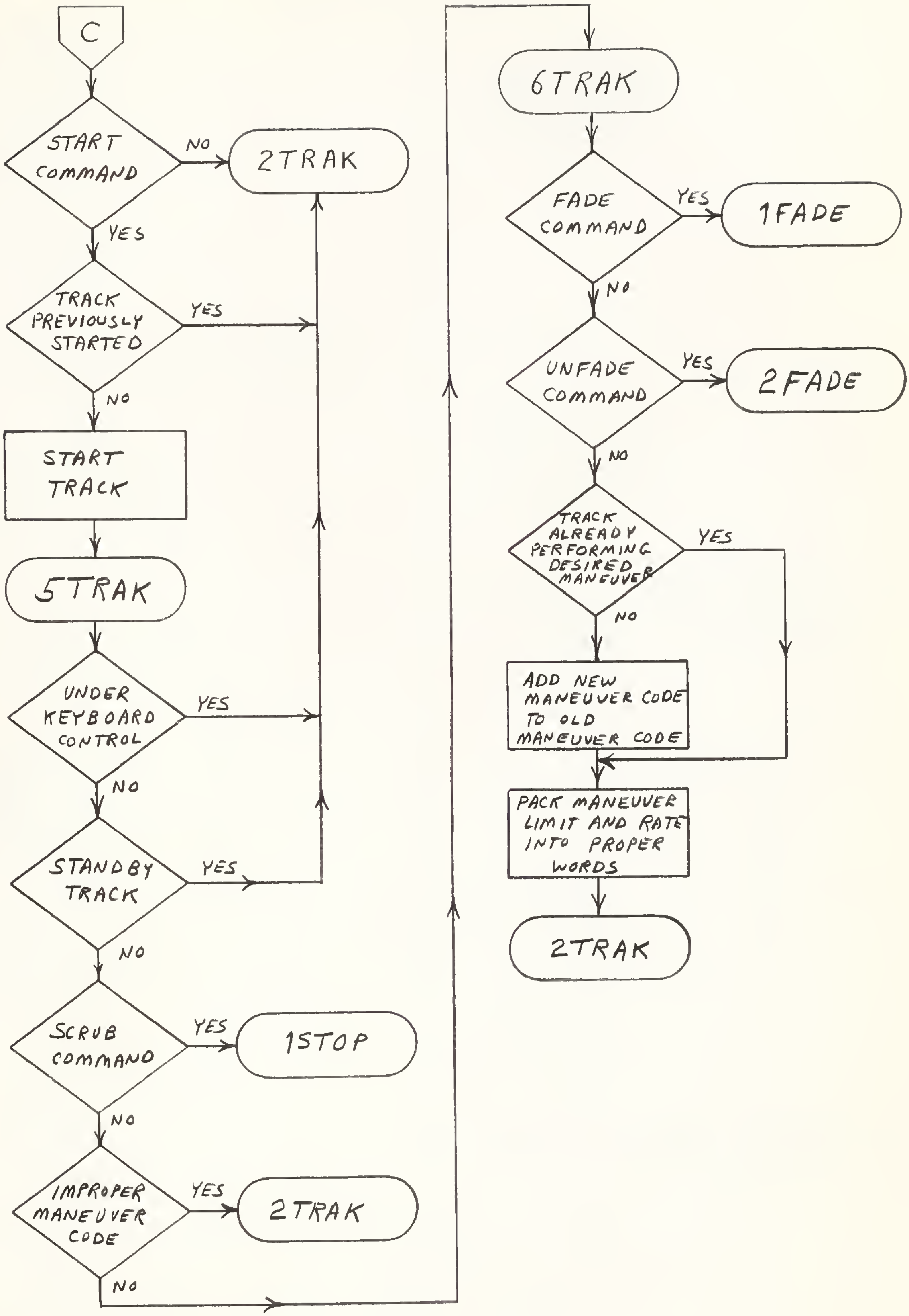

C
START COMMAND
NO
2TRAK
YES
TRACK PREVIOUSLY STARTED
YES
NO
START TRACK
5TRAK
UNDER KEYBOARD CONTROL
YES
NO
STANDBY TRACK
YES
NO
SCRUB COMMAND
YES
1STOP
NO
IMPROPER MANEUVER CODE
YES
2TRAK
NO
6TRAK
FADE COMMAND
YES
1FADE
NO
UNFADE COMMAND
YES
2FADE
NO
TRACK ALREADY PERFORMING DESIRED MANEUVER
YES
NO
ADD NEW MANEUVER CODE TO OLD MANEUVER CODE
PACK MANEUVER LIMIT AND RATE INTO PROPER WORDS
2TRAK

Figure II-3

Update Track Positions

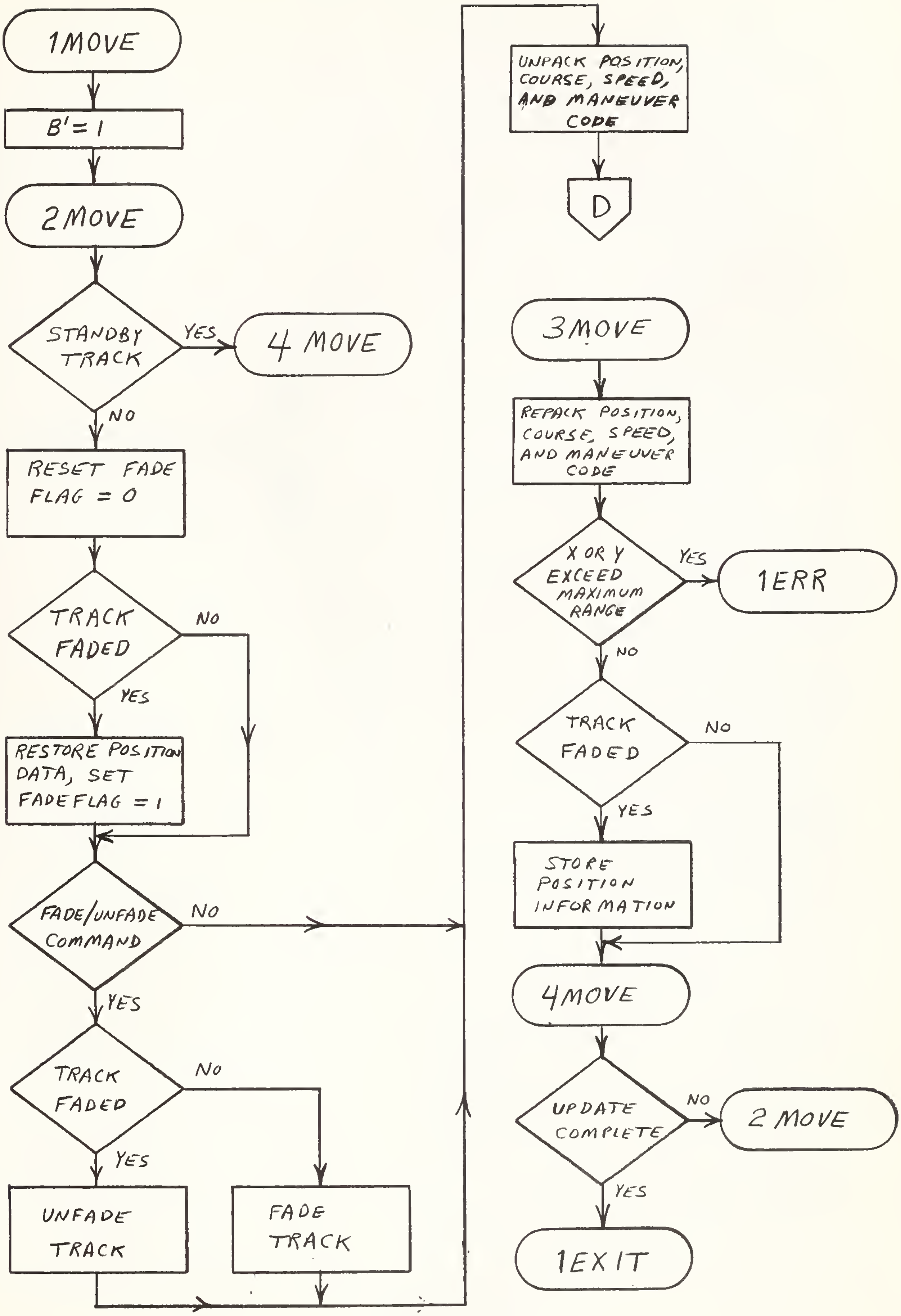


Figure II-4

Select Maneuver

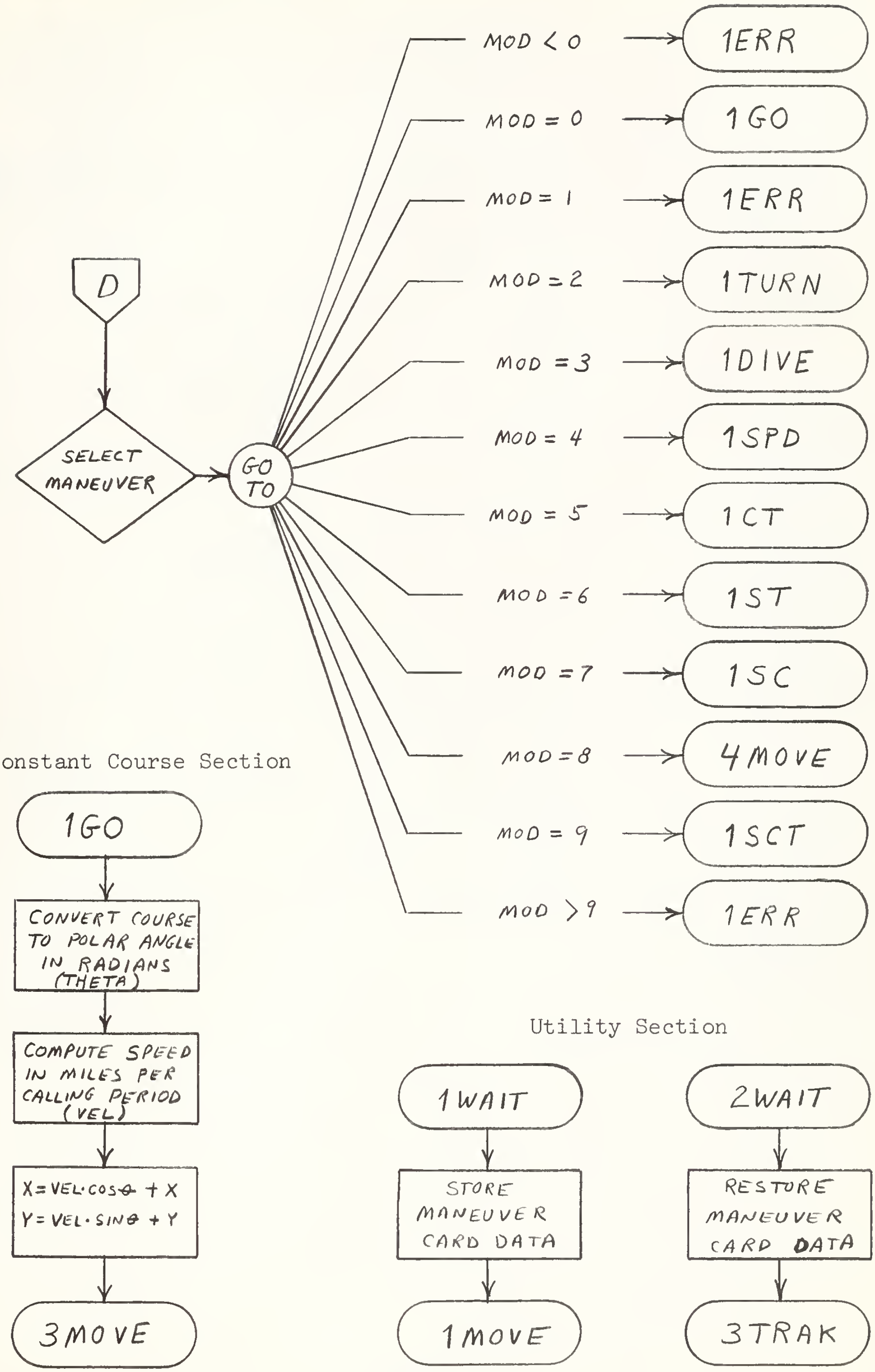


Figure II-5

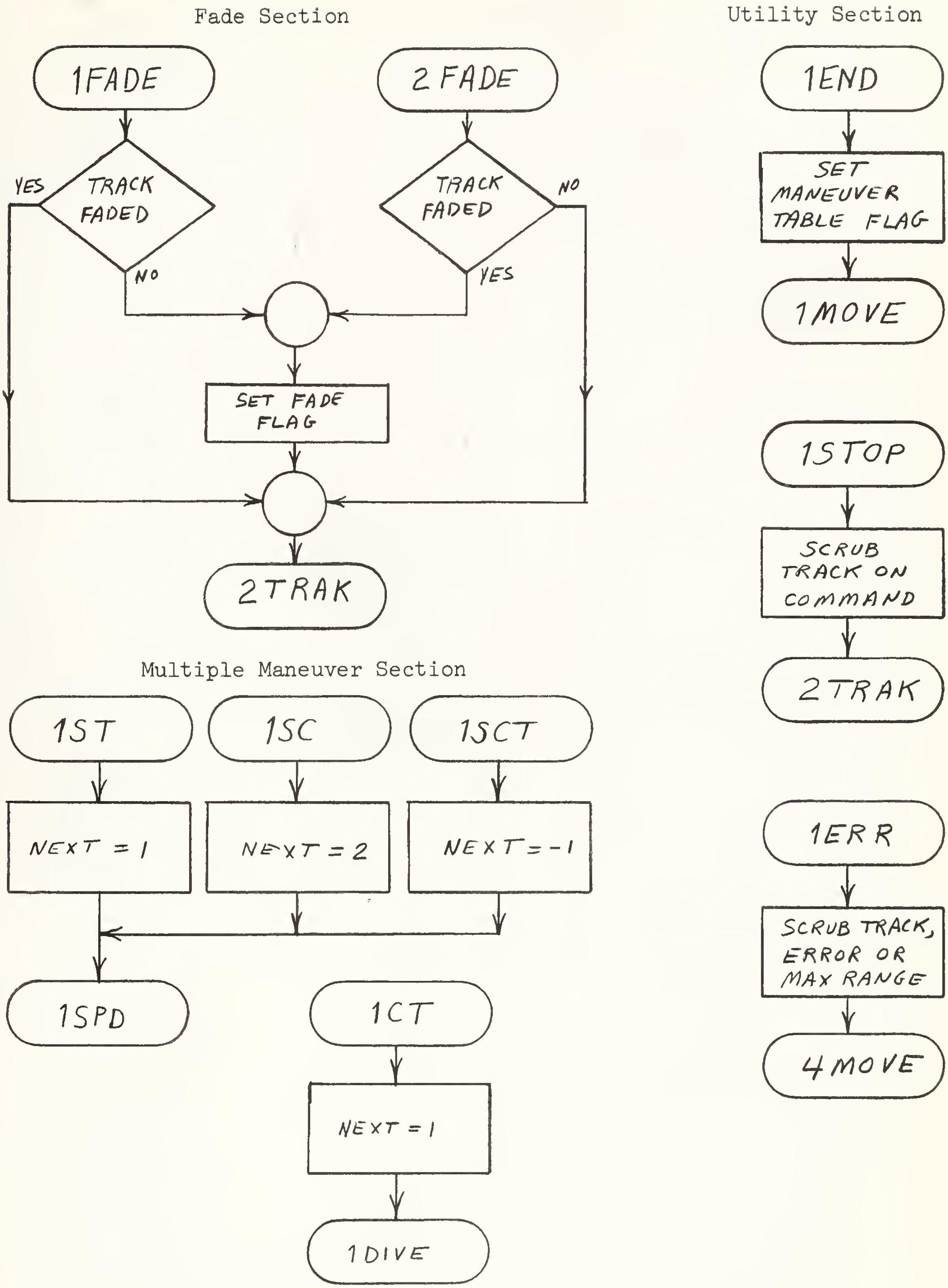

Fade Section
1FADE
TRACK FADED
YES
NO
2 FADE
TRACK FADED
NO
YES
SET FADE FLAG
2TRAK
Utility Section
1END
SET MANEUVER TABLE FLAG
1MOVE
1STOP
SCRUB TRACK ON COMMAND
2TRAK
1ERR
SCRUB TRACK, ERROR OR MAX RANGE
4MOVE
Multiple Maneuver Section
1ST
1SC
1SCT
NEXT = 1
NEXT = 2
NEXT = -1
1SPD
1CT
NEXT = 1
1DIVE

Figure II-6

Turn Section

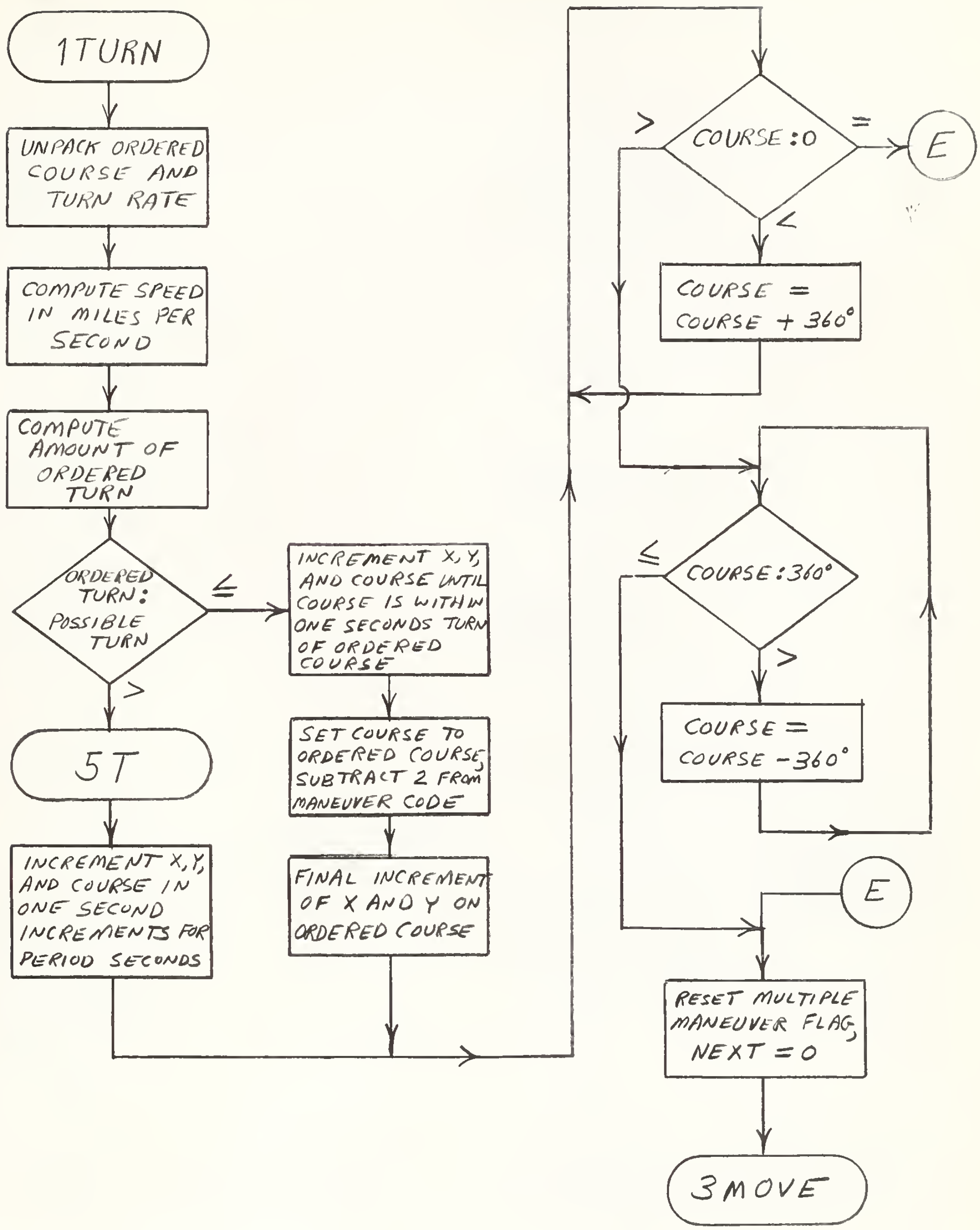


Figure II-7

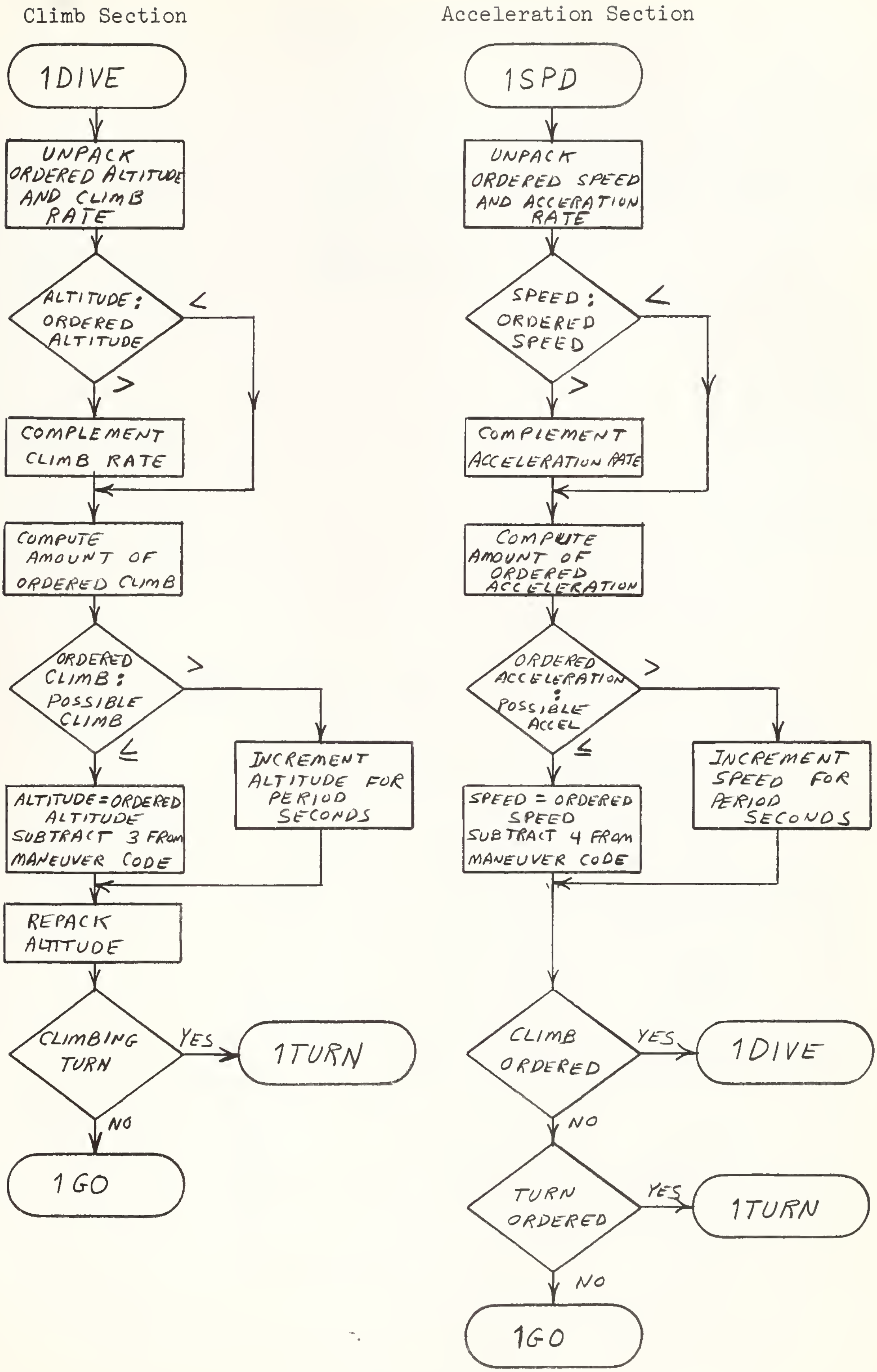

Climb Section
1DIVE
UNPACK ORDERED ALTITUDE AND CLIMB RATE
ALTITUDE : ORDERED ALTITUDE
<
>
COMPLEMENT CLIMB RATE
COMPUTE AMOUNT OF ORDERED CLIMB
ORDERED CLIMB : POSSIBLE CLIMB
>
≤
INCREMENT ALTITUDE FOR PERIOD SECONDS
ALTITUDE = ORDERED ALTITUDE SUBTRACT 3 FROM MANEUVER CODE
REPACK ALTITUDE
CLIMBING TURN
YES
1TURN
NO
1GO
Acceleration Section
1SPD
UNPACK ORDERED SPEED AND ACCERATION RATE
SPEED : ORDERED SPEED
<
>
COMPLEMENT ACCELERATION RATE
COMPUTE AMOUNT OF ORDERED ACCELERATION
ORDERED ACCELERATION : POSSIBLE ACCEL
>
≤
INCREMENT SPEED FOR PERIOD SECONDS
SPEED = ORDERED SPEED SUBTRACT 4 FROM MANEUVER CODE
CLIMB ORDERED
YES
1DIVE
NO
TURN ORDERED
YES
1TURN
NO
1GO

Figure II-8

Keyboard Command Section

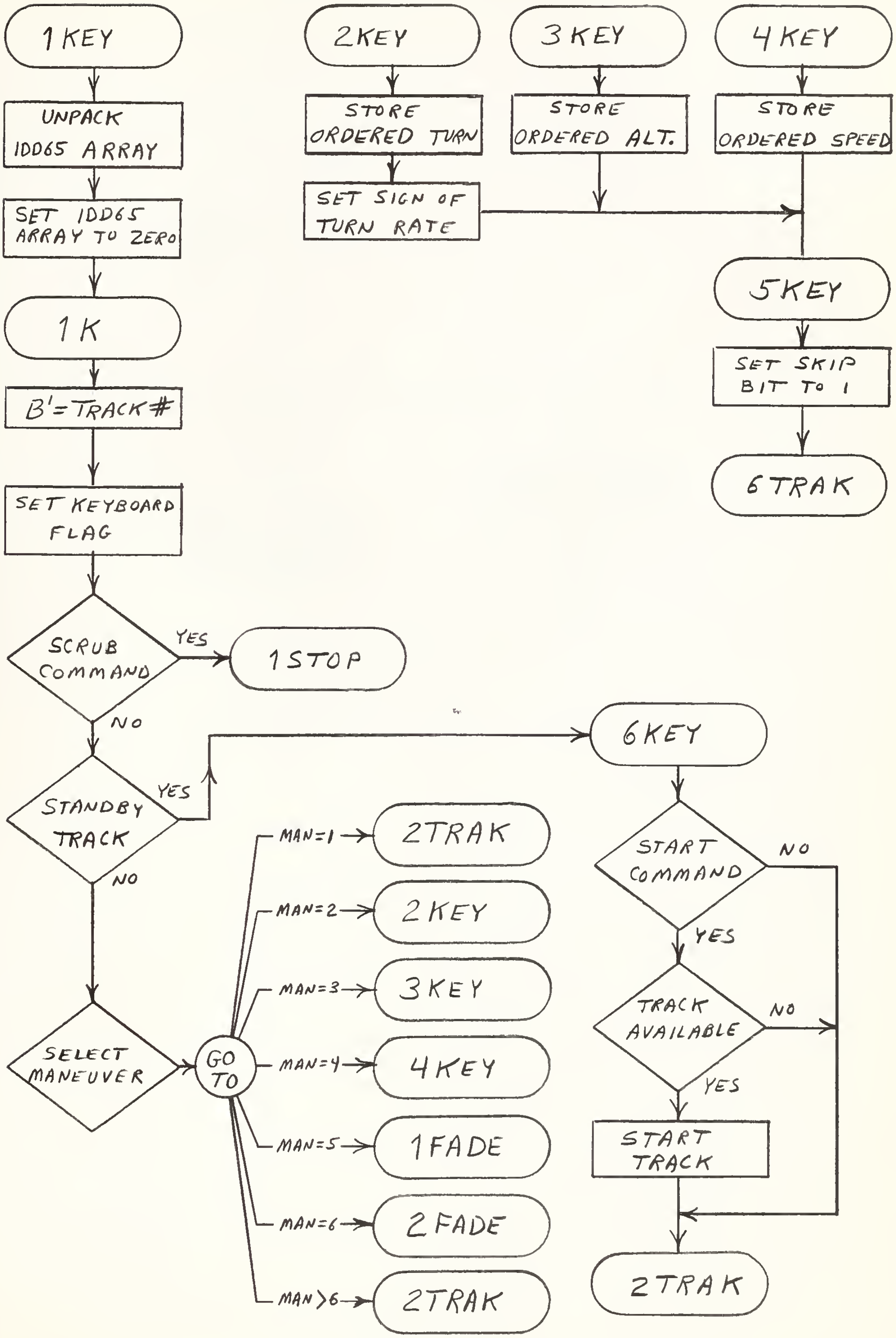


Figure II-9

Data Process Command Section

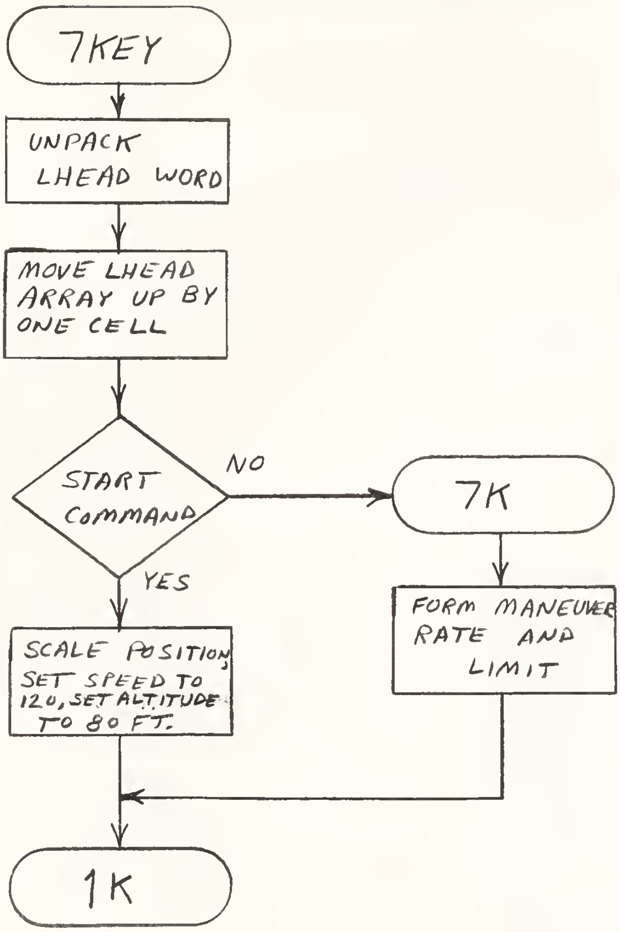


Figure II-10

APPENDIX II-C

SUBROUTINE TRAKGEN PROGRAM LISTING

```
      MACHINE TRAKGEN

*  DATA CARDS - EACH ENTRY MUST BE RIGHT JUSTIFIED IN SPECIFIED COLUMNS
*       1. NUMBER OF PREPROGRAMMED TRACKS, 15 MAXIMUM, COLUMNS 1-2
*       2. INITIAL POSITION CARDS, ONE FOR EACH PREPROGRAMMED TRACK IN
*          ORDER OF INCREASING TRACK NUMBER WITH THE LAST ONE BEING 15
*          A. X POSITION IN MILES, +- 279 MILES MAXIMUM, COLUMNS 1-10
*          B. Y POSITION IN MILES, +- 279 MILES MAXIMUM, COLUMNS 11-20
*             NOTE - X AND Y POSITIONS MAY HAVE UP TO 4 DECIMAL PLACES
*          C. ALTITUDE IN FEET, +- 98689 FEET MAXIMUM, COLUMNS 25-30
*          D. COURSE IN DEGREES, COLUMNS 38-40
*          E. SPEED IN KNOTS, 4095 KNOTS MAXIMUM, COLUMNS 47-50
*       3. MANEUVER CARDS, ONE FOR EACH MANEUVER
*          A. TIME FOR MANEUVER IN SECONDS, COLUMNS 1-10
*          B. TRACK NUMBER, COLUMNS 19-20
*          C. MANEUVER CODE, COLUMN 30
*             1   START TRACK
*             2   TURN
*             3   CLIMB/DIVE
*             4   CHANGE SPEED
*             5   FADE TRACK
*             6   UNFADE TRACK
*             8   SCRUB TRACK
*          D. ORDERED COURSE IN DEGREES, ORDERED SPEED IN KNOTS, OR
*             ORDERED ALTITUDE IN FEET, COLUMNS 35-40
*          E. MANEUVER RATE, COLUMNS 43-50, THIS ENTRY MAY HAVE ONE.
*             DECIMAL PLACE FOR TURN RATE
*                POSITIVE TURN RATE SIGNIFIES A STARBOARD TURN,
*                NEGATIVE TURN RATE SIGNIFIES A PORT TURN
*             INSERT TURN RATE IN DEGREES PER SECOND, 25.5 MAXIMUM,
*             INSERT CLIMB RATE IN FEET PER MINUTE, 65535 MAXIMUM, 43
*             MINIMUM, INSERT ACCELERATION RATE IN KNOTS PER SECOND,
*             255 MAXIMUM.
*          LAST MANEUVER CARD MUST HAVE MANEUVER TIME EQUAL TO ZERO
```

```
*            TRAKGEN VARIABLES AND CONSTANTS

*       AXIS          CONVERSION FACTOR - DEGREES TO RADIANS
*       CRS           TRACK COURSE
*       DEGREES       360 DEGREES SCALED BY TEN
*       DPS           TURN RATE, TENTHS OF DEGREES PER SECOND
*       FFLAG         FADE/UNFADE COMMAND BIT
*       IALT          ORDERED ALTITUDE
*       ICHECK        ABSOLUTE VALUE OF TURN RATE
*       ICRS          TRACK COURSE
*       IDD65         KEYBOARD COMMAND ARRAY
*       IDPS          TURN RATE, TENTHS OF DEGREES PER SECOND
*       IFADE         FADE BIT, 1 - TRACK FADED
*       IFPS          CLIMB RATE, FEET PER SECOND
*       IHEAD         COURSE, SPEED, AND MANEUVER CODE ARRAY
*       IKPS          ACCELERATION RATE, KNOTS PER SECOND
*       IPERIOD       TIME BETWEEN SUBROUTINE CALLS
*       IPOSIT        X AND Y POSITION ARRAY
*       IPOSIT1       INITIAL X AND Y POSITION ARRAY, FADE ARRAY
*       IRATE         SCRATCH WORK CELL FOR MANEUVER RATE
*       ISCALE1       SCALE FACTOR FOR X AND Y POSITION
*       ISCALE2       SCALE FACTOR FOR ALTITUDE
*       ISKIP         KEYBOARD CONTROLLED BIT, 1 - KEYBOARD CONTROLLED
*       ISPD          TRACK SPEED
*       ITER          NUMBER OF ITERATIONS IN TURN ROUTINE
*       IVEL          ORDERED SPEED
*       IX            TRACK X POSITION
*       IY            TRACK Y POSITION
*       IZ            TRACK ALTITUDE
*       JCHECK        ICHECK * IPERIOD
*       JCRS          ORDERED COURSE IN DEGREES
*       JFPS          ABSOLUTE VALUE OF CLIMB RATE
*       JHEAD         ALTITUDE ARRAY
*       JHEAD1        INITIAL ALTITUDE ARRAY, FADE ARRAY
*       JUMP          SCRATCH WORK CELL FOR AMOUNT OF MANEUVER
*       KCRS          ORDERED COURSE IN TENTHS OF DEGREES
```

```
*        KEY           KEYBOARD RETURN FLAG
*        KHEAD         ORDERED SPEED ARRAY
*        KMAN          KEYBOARD MANEUVER CODE
*        KPS           ABSOLUTE VALUE OF ACCELERATION RATE
*        LAST          MANEUVER TABLE FLAG,   1 - MANEUVER TABLE EMPTY
*        LEFT          STORE MANEUVER DATA FLAG
*        LHEAD         DATA PROCESS COMMAND ARRAY
*        LIMIT         DESIRED MANEUVER LIMIT
*        LIMITH        MANEUVER LIMIT FOR DELAYED MANEUVER
*        LTIME         TIME OF LAST SUBROUTINE CALL IN SECONDS
*        MAN           MANEUVER CODE
*        MANH          MANEUVER CODE FOR DELAYED MANEUVER
*        MAX           MAXIMUM X OR Y RANGE BEFORE OVERFLOW
*        MOD           MANEUVER CODE
*        MODPOS        CURRENT TRACK MANEUVER CODE, LEFT JUSTIFIED
*        MOD8          MANEUVER CODE FOR STANDBY TRACK
*        MTIME         EXECUTION TIME FOR MANEUVER
*        MTIMEH        DELAYED EXECUTION TIME FOR MANEUVER
*        NACTIVE       NUMBER OF NON PREPROGRAMMED TRACKS
*        NEXT          MULTIPLE MANEUVER FLAG
*        NORTH         360 DEGREES SCALED BY TEN
*        NR            TRACK NUMBER
*        NRH           TRACK NUMBER FOR DELAYED MANEUVER
*        NTIME         TIME OF PRESENT SUBROUTINE CALL IN SECONDS
*        NTRACKS       NUMBER OF PREPROGRAMMED TRACKS
*        PERIOD        TIME BETWEEN SUBROUTINE CALLS
*        RATE          MANEUVER RATE
*        RATEH         MANEUVER RATE FOR DELAYED MANEUVER
*        SCALE1        SCALE FACTOR FOR X AND Y POSITION
*        SCALE2        SCALE FACTOR FOR ALTITUDE
*        SPD           TRACK SPEED
*        TEMP          SCRATCH WORK CELL
*        THETA         COURSE POLAR ANGLE IN RADIANS
*        VEL           TRACK SPEED IN MILES/SECOND OR MILES/PERIOD
*        X             TRACK X POSITION
*        Y             TRACK Y POSITION
```

```
      RSV(IPOSIT=15,IPOSIT1=15,IHEAD=15,JHEAD=15,JHEAD1=15,KHEAD=15,
     *     LHEAD=60,IDD65=10)
      COMMON IPOSIT,IHEAD,JHEAD,LHEAD,IDD65,ISCALE1,ISCALE2,SCALE1,
     *        SCALE2,NTIME
      LIB(FLOATFIX = FLOAT, FIXFLOAT = FIX, COSF, SINF, XABSF, SIGNF)

 1000 FORMAT(I2)                                                        00000
 1001 FORMAT(2F10.0,4X,I6,7X,I3,6X,I4)                                  00001
 1002 FORMAT(I10,8X,I2,9X,I1,4X,I6,2X,F8.0)                             00002

      CON(MASK1  = 7777777700000000B, MASK2  = 7577777777777777B,
     *    MASK3  = 7777777777770000B, MASK4  = 0377777777777777B,
     *    MASK5  = 7777777777777000B, MASK6  = 7400000000000000B,
     *    MASK7  = 7777777777777760B, MASK8  = 0037777700000000B,
     *    MASK9  = 7777777777777400B, MASK10 = 7777777700007777B,
     *    MASK11 = 0000000077777777B, MASK12 = 7777700077777777B,
     *    MASK13 = 7740077777777777B, MASK14 = 7400000777777777B,
     *    MASK15 = 7437777700000000B, MASK16 = 7777777777600000B,
     *    MASK17 = 7777776000007777B, MASK18 = 7777001777777777B,
     *    MASK19 = 0000001777770000B, MASK20 = 0000776000007777B)
      CON(MT1 = 4400000000000000B, MT2 = 3000000000000000B,
     *    MT3 = 2400000000000000B, MT4 = 1000000000000000B,
     *    MC1 = 4400000000000000B, MC2 = 3400000000000000B,
     *    MC3 = 2400000000000000B, MC4 = 1400000000000000B,
     *    MS1 = 4400000000000000B, MS2 = 3400000000000000B,
     *    MS3 = 3000000000000000B, MS4 = 2000000000000000B)
      CON(IZERO  = 0, IONE   = 1, ITWO   = 2, ITHREE = 3, IFOUR  = 4,
     *    IFIVE  = 5, ISIX   = 6, ISEVEN = 7, IEIGHT = 8, ININE  = 9,
     *    ITEN  = 10, I15   = 15)
      CON(ISKIP = 0200000000000000B, MOD8 = 4000000000000000B, I85 = 85,
     *    I30000 = 30000, AXIS = 4500., RADIANS = .0017453293,
     *    DEGREES = 3600., MAX = 37777777B, IFADE = 0100000000000000B,
     *    FFLAG = 0040000000000000B, SIXTY = 60., NORTH = 3600,
     *    TEN = 10.)
```

```
* ENTRY TO SUBROUTINE

1RET  SLJ   (*)          ENI   (0)        . RETURN JUMP
      SIU 1 (1SAVE)      SIL 2 (1SAVE)    . SAVE
      SIU 3 (2SAVE)      SIL 4 (2SAVE)    .   INDEX
      SIU 5 (3SAVE)      SIL 6 (3SAVE)    .     REGISTERS
      LDA   (NTIME)      AJP 1 (1TRAK)    . JUMP IF NOT INITIAL CALL
      EXF   (2000B)      ENI   (0)        . STOP REAL TIME CLOCK
      LDA   (I30000)     STA   (ISCALE1)  . FORM ISCALE1
      LDA   (I85)        STA   (ISCALE2)  . FORM ISCALE2
      LDA   (ISCALE1)    SLJ 4 (FLOAT)    .
      STA   (SCALE1)     ENI   (0)        . FORM SCALE1
      LDA   (ISCALE2)    SLJ 4 (FLOAT)    .
      STA   (SCALE2)     ENI 1 (1)        . FORM SCALE2
      ENA   (0)          ENI   (0)        . SET IPOSIT, IHEAD,
      STA 1 (IPOSIT)     STA 1 (IHEAD)    .     JHEAD, AND KHEAD
      STA 1 (JHEAD)      STA 1 (KHEAD)    .         ARRAYS TO ZERO
      ISK 1 (15)         SLJ   (L-2)      .
      ENI 1 (1)          ENA   (0)        . SET LHEAD
      STA 1 (LHEAD)      ENI   (0)        .   ARRAY TO
      ISK 1 (60)         SLJ   (L-1)      .     ZERO
      ENI 1 (1)          ENA   (0)        . SET IDD65
      STA 1 (IDD65)      ENI   (0)        .   ARRAY TO
      ISK 1 (10)         SLJ   (L-1)      .     ZERO

* READ, SCALE, AND PACK INITIAL POSITION DATA

      READ 1000, NTRACKS
      LDA   (NTRACKS)    AJP   (2EXIT)    . JUMP IF NO PREPROGRAM TRACKS
      SAU   (2READ)      ENI 1 (1)        .
1READ SIL 1 (TEMP)       LDA   (TEMP)     . FORM
      ADD   (I15)        SUB   (NTRACKS)  .   ARRAY
      STA   (TEMP)       LIL 2 (TEMP)     .     SUBSCRIPT
      READ 1001, X,Y,IZ,ICRS,ISPD
      LDA   (X)          FMU   (SCALE1)   .
      SLJ 4 (FIX)        ZRO   (0)        .
```

```
      SCL   (MASK1)     ALS   (24)       .                              00037
      STA 2 (IPOSIT1)   LDA   (Y)        . PACK INITIAL X POSITION      00040
      FMU   (SCALE1)    SLJ 4 (FIX)      .                              00041
      SCL   (MASK1)     RAD 2 (IPOSIT1)  . PACK INITIAL Y POSITION      00042
      LDA   (IZ)        MUI   (ISCALE2)  .                              00043
      STA 2 (JHEAD1)    LDA   (ICRS)     . PACK INITIAL ALTITUDE        00044
      MUI   (ITEN)      ALS   (12)       .                              00045
      STA 2 (IHEAD)     LDA   (ISPD)     . PACK INITIAL COURSE          00046
      SCL   (MASK3)     RAD 2 (IHEAD)    . PACK INITIAL SPEED           00047
2READ ISK 1 (*)         SLJ   (1READ)    .                              00050

* SET UNUSED TRACKS TO STANDBY, SET FLAGS TO ZERO

2EXIT ENA   (15)        SUB   (NTRACKS)  .                              00051
      STA   (NACTIVE)   ENI 1 (1)        .                              00052
      LDA 1 (IHEAD)     SST   (MOD8)     . SET TRACKS                   00053
      STA 1 (IHEAD)     ENI   (0)        .   TO STANDBY                 00054
      ISK 1 (15)        SLJ   (L-2)      .                              00055
      ENA   (0)         STA   (LTIME)    . SET FLAGS                    00056
      STA   (LAST)      STA   (LEFT)     .   TO ZERO                    00057
      STA   (NEXT)      EXF   (1000B)    . START REAL TIME CLOCK        00060
      LDA   (NACTIVE)   SLJ   (1SAVE)    . END OF INITIAL CALL          00061

* NON INITIAL CALL, PROCESS MANEUVER TABLE ENTRIES

1TRAK LDA   (NTIME)     SUB   (LTIME)    . COMPUTE CALLING PERIOD       00062
      STA   (IPERIOD)   SLJ 4 (FLOAT)    . STORE IPERIOD                00063
      STA   (PERIOD)    ENI   (0)        . STORE PERIOD                 00064
      LDA   (LHEAD+1)   AJP 3 (7KEY)     . JUMP IF DATA PROCESS COMMAND 00065
      LDA   (IDD65+9)   AJP 1 (1KEY)     . JUMP IF KEYBOARD COMMAND     00066
      LDA   (LAST)      AJP 1 (1MOVE)    . JUMP IF MANEUVER TABLE EMPTY 00067
      LDA   (LEFT)      AJP 1 (2WAIT)    . JUMP IF HOLDING A MANEUVER   00070
2TRAK LDA   (KEY)       AJP   (L+3)      . JUMP TO READ NEXT DATA CARD  00071
      AJP 3 (1SAVE)     ENA   (0)        . JUMP IF PERIOD EQUALS ZERO   00072
      STA   (KEY)       SLJ   (L-6)      . RESET KEYBOARD FLAG          00073
      EXF   (2000B)     ENI   (0)        . STOP REAL TIME CLOCK         00074
```

```
      READ 1002, MTIME,NR,MAN,LIMIT,RATE
      EXF    (1000B)      ENI    (0)        . START REAL TIME CLOCK
3TRAK LDA    (MTIME)      AJP    (1END)     . JUMP IF MANEUVER TABLE TO END
      SUB    (NTIME)      AJP    (4TRAK)    . JUMP IF TIME FOR MANEUVER
      AJP 2  (1WAIT)      ENI    (0)        . JUMP IF NOT TIME FOR MANEUVER
4TRAK LIL 1  (NR)         LDA 1  (IHEAD)    . SET INDEX ONE TO TRACK NUMBER
      SCL    (MASK4)      SUB    (MOD8)     . UNPACK OLD MANEUVER CODE
      AJP 1  (5TRAK)      ENA    (1)        . JUMP IF OLD TRACK
      SUB    (MAN)        AJP 1  (2TRAK)    . JUMP IF NOT START COMMAND
      LDA 1  (IHEAD)      SCL    (MOD8)     . CLEAR OLD MANEUVER CODE
      AJP    (2TRAK)      STA 1  (IHEAD)    . JUMP IF PREVIOUSLY STARTED
      LDA 1  (IPOSIT1)    STA 1  (IPOSIT)   . START
      LDA 1  (JHEAD1)     STA 1  (JHEAD)    .   NEW
      SLJ    (2TRAK)      ZRO    (0)        .     TRACK
5TRAK LIL 1  (NR)         LDA 1  (IHEAD)    . UPDATE MANEUVER CODES
      SCL    (MASK2)      SUB    (ISKIP)    . UNPACK ISKIP BIT
      AJP    (2TRAK)      LDA 1  (IHEAD)    . JUMP IF KEYBOARD CONTROLLED
      SCL    (MASK4)      STA    (MODPOS)   . UNPACK OLD MANEUVER CODE
      SUB    (MOD8)       AJP    (2TRAK)    . JUMP IF STANDBY TRACK
      LDA    (MAN)        SUB    (IEIGHT)   . CHECK FOR SCRUB COMMAND
      AJP    (1STOP)      LDA    (MAN)      . JUMP IF TRACK TO BE SCRUBBED
      AJP    (2TRAK)      ENI    (0)        . JUMP IF MAN IS ZERO (ERROR)
6TRAK LDA    (MAN)        ENI    (0)        . SELECT MANEUVER
      INA    (-1)         AJP    (2TRAK)    .    MAN ERROR
      INA    (-1)         AJP    (L+6)      .    TURN
      INA    (-1)         AJP    (L+8)      .    CLIMB/DIVE
      INA    (-1)         AJP    (L+10)     .    CHANGE SPEED
      INA    (-1)         AJP    (1FADE)    .    FADE
      INA    (-1)         AJP    (2FADE)    .    UNFADE
      SLJ    (2TRAK)      ZRO    (0)        . JUMP IF MAN GT SIX (ERROR)
      ENI 2  (4)          LDA    (MODPOS)   . SEARCH FOR TURN
      EQS 2  (MT1)        SLJ    (L+8)      . JUMP IF TRACK NOT TURNING
      SLJ    (1PACK)      ZRO    (0)        . JUMP IF TRACK ALREADY TURNING
      ENI 2  (4)          LDA    (MODPOS)   . SEARCH FOR CLIMB
      EQS 2  (MC1)        SLJ    (L+5)      . JUMP IF TRACK NOT CLIMBING
      SLJ    (1PACK)      ZRO    (0)        . JUMP IF TRACK ALREADY CLIMBING
```

```
      ENI 2 (4)        LDA   (MODPOS)  . SEARCH FOR ACCELERATION
      EQS 2 (MS1)      SLJ   (L+2)     . JUMP IF TRACK NOT ACCELERATING
      SLJ   (1PACK)    ZRO   (0)       . JUMP IF TRACK ACCELERATING
      LDA   (MAN)      SCL   (MASK7)   . MASK MANEUVER CODE
      ALS   (44)       RAD   (MODPOS)  . ADD NEW MANEUVER CODE
      LDA 1 (IHEAD)    SCL   (MASK6)   . CLEAR OLD MANEUVER CODE
      ADD   (MODPOS)   STA 1 (IHEAD)   . REPLACE MANEUVER CODE
      SLJ   (1PACK)    ZRO   (0)       .
1FADE LDQ 1 (IHEAD)    LDL   (IFADE)   . CHECK FOR FADE
      AJP 1 (2TRAK)    SLJ   (L+3)     . A-JUMP IF ALREADY FADED
2FADE LDQ 1 (IHEAD)    LDL   (IFADE)   . CHECK FOR FADE
      AJP   (2TRAK)    ENI   (0)       . JUMP IF NOT FADED
      LDA 1 (IHEAD)    SST   (FFLAG)   . VALID FADE/UNFADE COMMAND,
      STA 1 (IHEAD)    SLJ   (2TRAK)   .   SET FADE FLAG
1PACK LDA   (MAN)      ENI   (0)       . SELECT PACK ROUTINE
      INA   (-2)       AJP   (2PACK)   .    TURN
      INA   (-1)       AJP   (3PACK)   .    CLIMB/DIVE
      INA   (-1)       AJP   (4PACK)   .    ACCELERATE
2PACK LDA   (RATE)     FMU   (TEN)     . TURN
      ENI   (0)        SLJ 4 (FIX)     .   CALLED
      STA   (IRATE)    LDA 1 (IHEAD)   .     FOR,
      SCL   (MASK8)    STA 1 (IHEAD)   .       PACK
      LDA   (LIMIT)    SCL   (MASK5)   .         INTO
      ALS   (24)       RAD 1 (IHEAD)   .           ORDERED COURSE
      LDA   (IRATE)    SCL   (MASK9)   .             AND
      ALS   (33)       RAD 1 (IHEAD)   .               TURN RATE
      SLJ   (2TRAK)    ZRO   (0)       .                 .
3PACK LDA   (LIMIT)    MUI   (ISCALE2) . CLIMB
      STA   (LIMIT)    ENI   (0)       .   CALLED
      LDA   (RATE)     FMU   (SCALE2)  .     FOR,
      FDV   (SIXTY)    SLJ 4 (FIX)     .       PACK
      STA   (IRATE)    LDA 1 (JHEAD)   .         INTO
      SCL   (MASK1)    STA 1 (JHEAD)   .           ORDERED ALTITUDE
      LDA   (LIMIT)    SCL   (MASK1)   .             AND
      ALS   (24)       RAD 1 (JHEAD)   .               CLIMB RATE
      LDA 1 (KHEAD)    SCL   (MASK19)  .                 .
```

```
       STA 1 (KHEAD)     LDA   (IRATE)    .
       SCL   (MASK16)    ALS   (12)       .
       RAD 1 (KHEAD)     SLJ   (2TRAK)    .
4PACK  LDA   (RATE)      SLJ 4 (FIX)      . ACCELERATION
       STA   (IRATE)     LDA 1 (KHEAD)    .   CALLED FOR,
       SCL   (MASK20)    STA 1 (KHEAD)    .     PACK
       LDA   (LIMIT)     SCL   (MASK3)    .       INTO
       RAD 1 (KHEAD)     LDA   (IRATE)    .         ORDERED SPEED
       SCL   (MASK9)     ALS   (28)       .           AND
       RAD 1 (KHEAD)     SLJ   (2TRAK)    .             ACCELERATION RATE

* UTILITY SECTION

1WAIT  ENA   (1)         STA   (LEFT)     . STORE
       LDA   (MTIME)     STA   (MTIMEH)   .   MANEUVER
       LDA   (LIMIT)     STA   (LIMITH)   .     CARD
       LDA   (RATE)      STA   (RATEH)    .       DATA FOR
       LDA   (MAN)       STA   (MANH)     .         MANEUVER
       LDA   (NR)        STA   (NRH)      .            BEING
       SLJ   (1MOVE)     ZRO   (0)        .             HELD
2WAIT  ENA   (0)         STA   (LEFT)     . RESTORE
       LDA   (MTIMEH)    STA   (MTIME)    .   MANEUVER
       LDA   (LIMITH)    STA   (LIMIT)    .     CARD
       LDA   (RATEH)     STA   (RATE)     .       DATA FOR
       LDA   (MANH)      STA   (MAN)      .         MANEUVER
       LDA   (NRH)       STA   (NR)       .           BEING
       SLJ   (3TRAK)     ZRO   (0)        .             HELD
1END   ENA   (1)         STA   (LAST)     . MANEUVER TABLE EMPTY
       SLJ   (1MOVE)     ZRO   (0)        .
1STOP  LDA   (MOD8)      STA 1 (IHEAD)    . SCRUB TRACK
       ENA   (0)         STA 1 (IPOSIT)   .
       STA 1 (JHEAD)     STA 1 (KHEAD)    .
       SLJ   (2TRAK)     ZRO   (0)        .
1ERR   LDA   (MOD8)      STA 1 (IHEAD)    . SCRUB TRACK,
       ENA   (0)         STA 1 (IPOSIT)   .   MOD ERROR, OR
       STA 1 (JHEAD)     STA 1 (KHEAD)    .     MAX RANGE EXCEEDED
```

```
       SLJ    (4MOVE)      ZRO    (0)         .                                      00246

 * UPDATE TRACKS

 1MOVE ENI 1 (1)           ENI    (0)         . INITIALIZE TRACK NUMBER INDEX        00247
 2MOVE LDA 1 (IHEAD)       SCL    (MASK4)     . UNPACK MANEUVER CODE                 00250
       SUB    (MOD8)       AJP    (4MOVE)     . JUMP IF STANDBY TRACK                00251
       ENA    (0)          STA    (TEMP)      . RESET TEMP FADE FLAG                 00252
       LDQ 1 (IHEAD)       LDL    (IFADE)     . CHECK FOR FADE                       00253
       AJP    (L+4)        LDA 1 (IPOSIT1)    . JUMP IF NOT FADED                    00254
       STA 1 (IPOSIT)      LDA 1 (JHEAD1)     . TRACK PREVIOUSLY FADED,              00255
       STA 1 (JHEAD)       ENA    (1)         .   RESTORE POSITION INFORMATION       00256
       STA    (TEMP)       ENI    (0)         . SET FADED FLAG                       00257
       LDQ 1 (IHEAD)       LDL    (FFLAG)     . CHECK FOR FADE COMMAND               00260
       AJP    (L+6)        LDA    (TEMP)      . JUMP IF NO COMMAND                   00261
       AJP    (L+3)        LDA 1 (IHEAD)      . JUMP IF NOT FADED                    00262
       SCL    (IFADE)      SCL    (FFLAG)     . UNFADE TRACK,                        00263
       STA 1 (IHEAD)       SLJ    (L+3)       .   RESET FADE BITS                    00264
       LDA 1 (IHEAD)       SST    (IFADE)     . FADE TRACK,                          00265
       SCL    (FFLAG)      STA 1 (IHEAD)      .   SET FADE BIT                       00266
       LDA 1 (IHEAD)       SCL    (MASK4)     . UNPACK                               00267
       ARS    (44)         SCL    (MASK7)     .   MANEUVER MODE,                     00270
       STA    (MOD)        LDA 1 (IHEAD)      .     COURSE,                          00271
       SCL    (MASK10)     ARS    (12)        .       SPEED,                         00272
       STA    (ICRS)       LDA 1 (IHEAD)      .         X POSITION                   00273
       SCL    (MASK3)      STA    (ISPD)      .           AND                        00274
       LDA 1 (IPOSIT)      SCL    (MASK11)    .             Y POSITION               00275
       ARS    (24)         ENI    (0)         .               .                      00276
       STA    (IX)         LDA 1 (IPOSIT)     .                 .                    00277
       SCL    (MASK1)      ALS    (24)        .                   .                  00300
       ARS    (24)         STA    (IY)        .                     .                00301
       LDA    (ICRS)       SLJ 4 (FLOAT)      .               .                      00302
       STA    (CRS)        ENI    (0)         .                 .                    00303
       LDA    (ISPD)       SLJ 4 (FLOAT)      .                   .                  00304
       STA    (SPD)        ENI    (0)         .                     .                00305
       LDA    (IX)         SLJ 4 (FLOAT)      .               .                      00306
```

```
      STA   (X)       ENI   (0)       .                              00307
      LDA   (IY)      SLJ 4 (FLOAT)   .                              00310
      STA   (Y)       LDA   (MOD)     .                              00311
      AJP 3 (1ERR)    AJP   (1GO)     . SELECT MANEUVER              00312
      INA   (-1)      AJP   (1ERR)    .    MANEUVER CODE ERROR       00313
      INA   (-1)      AJP   (1TURN)   .    TURN                      00314
      INA   (-1)      AJP   (1DIVE)   .    CLIMB/DIVE                00315
      INA   (-1)      AJP   (1SPD)    .    ACCELERATION              00316
      INA   (-1)      AJP   (1CT)     .    CLIMBING TURN             00317
      INA   (-1)      AJP   (1ST)     .    ACCELERATION IN TURN      00320
      INA   (-1)      AJP   (1SC)     .    ACCELERATION IN CLIMB     00321
      INA   (-1)      AJP   (4MOVE)   .    SCRUBBED TRACK            00322
      INA   (-1)      AJP   (1SCT)    .    ACCELERATION IN CLIMB-TURN 00323
      SLJ   (1ERR)    ZRO   (0)       .    MANEUVER CODE ERROR       00324

* REPACK SECTION

3MOVE LDA   (IX)      AJP   (L+4)     . REPACK                       00325
      AJP 3 (L+2)     SUB   (MAX)     .   X POSITION,                00326
      AJP 3 (L+2)     SLJ   (1ERR)    .     Y POSITION,              00327
      LAC   (IX)      SLJ   (L-2)     .       COURSE,                00330
      LDA   (IX)      SCL   (MASK1)   .         SPEED,               00331
      ALS   (24)      STA 1 (IPOSIT)  .           AND                00332
      LDA   (IY)      AJP   (L+4)     .             MANEUVER MODE    00333
      AJP 3 (L+2)     SUB   (MAX)     .                              00334
      AJP 3 (L+2)     SLJ   (1ERR)    .   IF EITHER                  00335
      LAC   (IY)      SLJ   (L-2)     .     X OR Y IS                00336
      LDA   (IY)      SCL   (MASK1)   .       GREATER                00337
      RAD 1 (IPOSIT)  LDA 1 (IHEAD)   .         THAN MAXIMUM         00340
      SCL   (MASK14)  STA 1 (IHEAD)   .           RANGE,             00341
      LDA   (ICRS)    SCL   (MASK3)   .             SCRUB            00342
      ALS   (12)      RAD 1 (IHEAD)   .               TRACK          00343
      LDA   (ISPD)    SCL   (MASK3)   .                              00344
      RAD 1 (IHEAD)   LDA   (MOD)     .                              00345
      SCL   (MASK7)   ALS   (44)      .                              00346
      RAD 1 (IHEAD)   ENI   (0)       .                              00347
```

```
      LDQ 1 (IHEAD)      LDL   (IFADE)    . CHECK FOR FADE
      AJP   (4MOVE)      LDA 1 (IPOSIT)   . JUMP IF NOT FADED
      STA 1 (IPOSIT1)    LDA 1 (JHEAD)    . TRACK FADED, SAVE
      STA 1 (JHEAD1)     ENA   (0)        .   POSITION INFORMATION,
      STA 1 (IPOSIT)     STA 1 (JHEAD)    .     SET POSITION TO ZERO
4MOVE ISK 1 (15)         SLJ   (2MOVE)    . END OF UPDATE LOOP

* EXIT FROM SUBROUTINE

1EXIT LDA   (NTIME)      STA   (LTIME)    . STORE REAL TIME
1SAVE ENI 1 (*)          ENI 2 (*)        . RESTORE INDEX
2SAVE ENI 3 (*)          ENI 4 (*)        .   REGISTERS, RETURN
3SAVE ENI 5 (*)          ENI 6 (*)        .     TO CALLING PROGRAM
      SLJ   (1RET)       ZRO   (0)        . RETURN

* CONSTANT COURSE SECTION

1GO   LDA   (AXIS)       FSB   (CRS)      . CONVERT COURSE TO POLAR
      FMU   (RADIANS)    STA   (THETA)    .   ANGLE IN RADIANS
      LDA   (SPD)        FMU   (SCALE1)   . COMPUTE SPEED IN
      FMU   (PERIOD)     FDV   (DEGREES)  .   MILES PER CALLING PERIOD
      STA   (VEL)        ENA   (THETA)    . INCREMENT X
      SAL   (COSF+1)     SLJ 4 (COSF)     .   .
      FMU   (VEL)        FAD   (X)        .     .
      ENI   (0)          SLJ 4 (FIX)      .       .
      STA   (IX)         ENA   (THETA)    . INCREMENT Y
      SAL   (SINF+1)     SLJ 4 (SINF)     .   .
      FMU   (VEL)        FAD   (Y)        .     .
      ENI   (0)          SLJ 4 (FIX)      .       .
      STA   (IY)         SLJ   (3MOVE)    .         .

* TURN SECTION

1TURN LDA 1 (IHEAD)      SCL   (MASK12)   . UNPACK
      ARS   (24)         STA   (JCRS)     .   ORDERED COURSE
      LDA 1 (IHEAD)      SCL   (MASK13)   . UNPACK TURN RATE
```

```
      ALS   (7)       ARS   (40)      . EXTEND SIGN BIT
      STA   (IDPS)    LDA   (SPD)     . COMPUTE SPEED
      FMU   (SCALE1)  FDV   (DEGREES) .   IN MILES
      STA   (VEL)     ENI   (0)       .     PER SECOND
      LDA   (IDPS)    SLJ 4 (FLOAT)   . FLOAT TURN RATE
      STA   (DPS)     LDA   (JCRS)    . CONVERT ORDERED COURSE
      MUI   (ITEN)    STA   (KCRS)    .   TO TENTHS OF DEGREES
      SUB   (ICRS)    STA   (JUMP)    .
      LDA   (IDPS)    AJP 2 (L+3)     . JUMP IF STARBOARD TURN
      LDA   (JUMP)    AJP 3 (1T)      . COMPUTE
      AJP   (2T)      AJP 2 (L+3)     .   AMOUNT
      LDA   (JUMP)    AJP 3 (L+2)     .     OF ORDERED
      AJP   (2T)      AJP 2 (3T)      .       TURN
      ENA   (JUMP)    SAL   (XABSF+1) . TURN
      ENI   (0)       SLJ 4 (XABSF)   .   PASSES
      STA   (TEMP)    LDA   (NORTH)   .     THROUGH
      SUB   (TEMP)    STA   (JUMP)    .       NORTH
      SLJ   (L+5)     ZRO   (0)       .
1T    ENA   (JUMP)    SAL   (XABSF+1) . TURN DOES
      ENI   (0)       SLJ 4 (XABSF)   .   NOT PASS
      STA   (JUMP)    SLJ   (L+2)     .     THROUGH NORTH
2T    LDA   (NORTH)   STA   (JUMP)    . 360 DEGREE TURN
3T    ENA   (IDPS)    SAL   (XABSF+1) . REMOVE SIGN
      ENI   (0)       SLJ 4 (XABSF)   .   OF TURN RATE
      STA   (ICHECK)  MUI   (IPERIOD) . COMPUTE AMOUNT OF TURN
      STA   (JCHECK)  LDA   (JUMP)    .   POSSIBLE THIS PERIOD
      SUB   (JCHECK)  AJP   (L+2)     . ORDERED TURN = POSSIBLE TURN
      AJP 2 (5T)      ENI   (0)       . ORDERED TURN GT POSSIBLE TURN
      ENQ   (0)       LDA   (JUMP)    . ORDERED TURN LT POSSIBLE TURN
      DVI   (ICHECK)  STA   (ITER)    . DETERMINE NUMBER OF
      LDA   (ITER)    SUB   (IONE)    .   ONE SECOND ITERATIONS
      AJP 3 (4T)      AJP   (4T)      . ONE OR LESS ITERATIONS
      ENI 2 (1)       LDA   (ITER)    . INITIALIZE
      SAU   (3TURN)   ENI   (0)       .   ITERATION INDEX
2TURN LDA   (CRS)     FAD   (DPS)     . INCREMENT
      STA   (CRS)     LDA   (AXIS)    .   X
```

```
      AJP 2 (L+3)        LDA   (CRS)       . JUMP IF COURSE POSITIVE               00510
      FAD   (DEGREES)    STA   (CRS)       . COURSE NEGATIVE, ADD 360 DEG          00511
      SLJ   (L-3)        ZRO   (0)         . JUMP TO RECHECK                       00512
      LDA   (CRS)        FSB   (DEGREES)   . CHECK FOR COURSE                      00513
      AJP   (L+1)        AJP 3 (L+3)       .   GREATER THAN 360 DEGREES            00514
      LDA   (CRS)        FSB   (DEGREES)   . COURSE GT 360, SUBTRACT 360           00515
      STA   (CRS)        SLJ   (L-3)       . JUMP TO RECHECK                       00516
      LDA   (CRS)        SLJ 4 (FIX)       . CONVERT COURSE                        00517
      STA   (ICRS)       ENA   (0)         .   TO FIXED POINT                      00520
      STA   (NEXT)       SLJ   (3MOVE)     . RESET MULTIPLE MANEUVER FLAG          00521

* CLIMB SECTION

1DIVE LDA 1 (JHEAD)      SCL   (MASK1)     . UNPACK ALTITUDE                       00522
      ALS   (24)         ARS   (24)        . EXTEND SIGN BIT                       00523
      STA   (IZ)         LDA 1 (JHEAD)     . STORE ALTITUDE                        00524
      SCL   (MASK11)     ARS   (24)        . UNPACK                                00525
      STA   (IALT)       LDA 1 (KHEAD)     .   ORDERED ALTITUDE                    00526
      SCL   (MASK17)     ARS   (12)        . UNPACK                                00527
      STA   (IFPS)       LDA   (IALT)      .   CLIMB RATE                          00530
      SUB   (IZ)         AJP 2 (L+2)       . JUMP IF CLIMB CALLED FOR              00531
      LAC   (IFPS)       STA   (IFPS)      . DIVE CALLED FOR                       00532
      LDA   (IALT)       SUB   (IZ)        . COMPUTE                               00533
      STA   (TEMP)       ENA   (TEMP)      .   AMOUNT                              00534
      SAL   (XABSF+1)    SLJ 4 (XABSF)     .     OF ORDERED                        00535
      STA   (JUMP)       ENA   (IFPS)      .       CLIMB/DIVE                      00536
      SAL   (XABSF+1)    SLJ 4 (XABSF)     . COMPUTE ABSOLUTE CLIMB RATE           00537
      STA   (JFPS)       MUI   (IPERIOD)   . COMPUTE AMOUNT OF CLIMB               00540
      STA   (TEMP)       LDA   (JUMP)      .   POSSIBLE THIS PERIOD                00541
      SUB   (TEMP)       AJP   (L+2)       . JUMP IF CLIMB = POSSIBLE CLIMB        00542
      AJP 2 (L+4)        ENI   (0)         . JUMP, CLIMB GT POSSIBLE CLIMB         00543
      LDA   (IALT)       STA   (IZ)        . CLIMB COMPLETE,                       00544
      LDA   (MOD)        SUB   (ITHREE)    .   SET ALTITUDE TO                     00545
      STA   (MOD)        SLJ   (L+3)       .     ORDERED ALTITUDE                  00546
      LDA   (IPERIOD)    MUI   (IFPS)      . INCREMENT                             00547
      ADD   (IZ)         STA   (IZ)        .   ALTITUDE                            00550
```

```
       LDA 1 (JHEAD)     SCL    (MASK11)  . CLEAR AND                         00551
       STA 1 (JHEAD)     LDA    (IZ)      .   REPLACE                         00552
       SCL   (MASK1)     RAD 1  (JHEAD)   .     ALTITUDE                      00553
       LDA   (NEXT)      AJP    (1GO)     . CHECK MULTIPLE MANEUVER FLAG      00554
       SLJ   (1TURN)     ZRO    (0)       . JUMP IF CLIMBING TURN             00555

* SPEED CHANGE SECTION

1SPD   LDA 1 (KHEAD)     SCL    (MASK3)   . UNPACK                            00556
       STA   (IVEL)      LDA 1  (KHEAD)   .   ORDERED SPEED                   00557
       SCL   (MASK18)    ARS    (28)      .     AND                           00560
       STA   (IKPS)      LDA    (IVEL)    .       ACCELERATION RATE           00561
       SUB   (ISPD)      AJP 2  (L+2)     . JUMP IF SPEED INCREASE            00562
       LAC   (IKPS)      STA    (IKPS)    . SPEED DECREASE CALLED FOR         00563
       LDA   (IVEL)      SUB    (ISPD)    . COMPUTE                           00564
       STA   (TEMP)      ENA    (TEMP)    .   AMOUNT                          00565
       SAL   (XABSF+1)   SLJ 4  (XABSF)   .     OF ORDERED                    00566
       STA   (JUMP)      ENA    (IKPS)    .       ACCELERATION AND            00567
       SAL   (XABSF+1)   SLJ 4  (XABSF)   .   MAXIMUM POSSIBLE                00570
       STA   (KPS)       MUI    (IPERIOD) .     ACCELERATION THIS PERIOD      00571
       STA   (TEMP)      LDA    (JUMP)    .       .                           00572
       SUB   (TEMP)      AJP    (L+2)     . JUMP IF ACCELERATION TO END       00573
       AJP 2 (L+4)       ENI    (0)       . ACCELERATION WILL NOT END         00574
       LDA   (IVEL)      STA    (ISPD)    . ACCELERATION COMPLETE,            00575
       LDA   (MOD)       SUB    (IFOUR)   .   SET SPEED TO                    00576
       STA   (MOD)       SLJ    (L+3)     .     ORDERED SPEED                 00577
       LDA   (IPERIOD)   MUI    (IKPS)    . INCREMENT                         00600
       ADD   (ISPD)      STA    (ISPD)    .   SPEED                           00601
       LDA   (ISPD)      SLJ 4  (FLOAT)   .     .                             00602
       STA   (SPD)       LDA    (NEXT)    . CHECK FOR SPEED                   00603
       AJP   (1GO)       AJP 3  (1DIVE)   .   CHANGE DURING CLIMB             00604
       SUB   (IONE)      AJP    (1TURN)   . CHECK FOR                         00605
       AJP 3 (1TURN)     ENA    (0)       .   TURN DURING CLIMB               00606
       STA   (NEXT)      SLJ    (1DIVE)   . RESET MULTIPLE MANEUVER FLAG      00607
```

```
* MULTIPLE MANEUVER SECTION

1CT    ENA    (1)          STA    (NEXT)       . CLIMBING TURN
       SLJ    (1DIVE)      ZRO    (0)          .
1ST    ENA    (1)          STA    (NEXT)       . SPEED CHANGE IN TURN
       SLJ    (1SPD)       ZRO    (0)          .
1SC    ENA    (2)          STA    (NEXT)       . SPEED CHANGE IN CLIMB
       SLJ    (1SPD)       ZRO    (0)          .
1SCT   ENA    (-1)         STA    (NEXT)       . SPEED CHANGE IN CLIMBING TURN
       SLJ    (1SPD)       ZRO    (0)          .

* KEYBOARD COMMAND SECTION

1KEY   LDA    (IDD65+1)    STA    (NR)         . STORE TRACK NUMBER
       LDA    (IDD65+2)    STA    (KMAN)       . STORE MANEUVER CODE
       LDA    (IDD65+3)    STA    (ICRS)       . STORE COURSE
       LDA    (IDD65+4)    STA    (ISPD)       . STORE SPEED
       LDA    (IDD65+5)    STA    (IALT)       . STORE ORDERED ALTITUDE
       LDA    (IDD65+6)    MUI    (ISCALE1)    . SCALE X POSITION
       STA    (IX)         LDA    (IDD65+7)    . STORE X POSITION
       MUI    (ISCALE1)    STA    (IY)         . STORE Y POSITION
       LDA    (IDD65+8)    MUI    (ISCALE2)    . SCALE ALTITUDE POSITION
       STA    (IZ)         LDA    (IDD65+10)   . STORE ALTITUDE POSITION
       STA    (RATE)       ENI 2  (1)          . STORE MANEUVER RATE
       ENA    (0)          ENI    (0)          . SET IDD65
       STA 2  (IDD65)      ENI    (0)          .   ARRAY
       ISK 2  (12B)        SLJ    (L-1)        .     TO ZERO
1K     LIL 1  (NR)         ENA    (KMAN)       . SET INDEX ONE TO TRACK NUMBER
       SAL    (XABSF+1)    SLJ 4  (XABSF)      . SET MANEUVER CODE EQUAL TO
       STA    (MAN)        LDA    (IPERIOD)    .   ABSOLUTE VALUE OF KMAN
       AJP 1  (L+2)        ENA    (-1)         . SET
       STA    (KEY)        SLJ    (L+2)        .   KEYBOARD
       ENA    (1)          STA    (KEY)        .     FLAG
       LDA    (MAN)        AJP    (2TRAK)      . JUMP IF MAN ERROR
       SUB    (IEIGHT)     AJP    (1STOP)      . JUMP TO SCRUB TRACK
       LDA 1  (IHEAD)      SCL    (MASK4)      . UNPACK OLD MANEUVER CODE
```

```
      STA    (MODPOS)     SUB    (MOD8)     . SELECT KEYBOARD MANEUVER          00647
      AJP    (6KEY)       LDA    (MAN)      .   START TRACK                     00650
      INA    (-1)         AJP    (2TRAK)    .   ERROR                           00651
      INA    (-1)         AJP    (2KEY)     .   TURN                            00652
      INA    (-1)         AJP    (3KEY)     .   CLIMB/DIVE                      00653
      INA    (-1)         AJP    (4KEY)     .   CHANGE SPEED                    00654
      INA    (-1)         AJP    (1FADE)    .   FADE                            00655
      INA    (-1)         AJP    (2FADE)    .   UNFADE                          00656
      SLJ    (2TRAK)      ZRO    (0)        .   ERROR                           00657

* KEYBOARD TURN SECTION

2KEY  LDA    (ICRS)       STA    (LIMIT)    . STORE ORDERED COURSE              00660
      ENA    (RATE)       SAL    (SIGNF+1)  . INSERT PROPER                     00661
      ENA    (KMAN)       SAL    (SIGNF+2)  .   SIGN INTO                       00662
      ENI    (0)          SLJ 4  (SIGNF)    .     TURN RATE                     00663
      STA    (RATE)       SLJ    (5KEY)     . END KEYBOARD TURN SECTION         00664

* KEYBOARD CLIMB SECTION

3KEY  LDA    (IALT)       STA    (LIMIT)    . STORE ORDERED ALTITUDE            00665
      SLJ    (5KEY)       ZRO    (0)        .                                   00666

* KEYBOARD SPEED CHANGE SECTION

4KEY  LDA    (ISPD)       STA    (LIMIT)    . STORE ORDERED SPEED               00667

* EXIT FROM KEYBOARD ROUTINE

5KEY  LDA 1 (IHEAD)       SST    (ISKIP)    . SET ISKIP                         00670
      STA 1 (IHEAD)       SLJ    (6TRAK)    .   TO ONE                          00671

* START TRACK FROM KEYBOARD

6KEY  LDA    (MAN)        SUB    (IONE)     . CHECK FOR START COMMAND           00672
      AJP 1 (2TRAK)       LDA 1 (IHEAD)     . JUMP IF NOT START (ERROR)         00673
```

```
      SCL    (MOD8)      AJP 1 (2TRAK)   . JUMP IF NOT PREVIOUSLY STARTED
      LDA    (IX)        SCL   (MASK1)   .
      ALS    (24)        STA 1 (IPOSIT)  . PACK X POSITION
      LDA    (IY)        SCL   (MASK1)   .
      RAD 1  (IPOSIT)    LDA   (IZ)      . PACK Y POSITION
      SCL    (MASK1)     STA 1 (JHEAD)   . PACK ALTITUDE
      LDA    (ICRS)      MUI   (12B)     . CONVERT COURSE TO
      SCL    (MASK3)     ALS   (12)      .   TENTHS OF DEGREES
      STA 1  (IHEAD)     LDA   (ISPD)    . PACK COURSE
      SCL    (MASK3)     SST   (ISKIP)   . SET SKIP BIT
      RAD 1  (IHEAD)     SLJ   (2TRAK)   . PACK SPEED

* DATA PROCESS COMMAND SECTION

7KEY  LDQ    (LHEAD+1)   LDL   (777777B) . UNPACK MANEUVER RATE
      STA    (TEMP)      LDA   (LHEAD+1) . STORE MANEUVER RATE
      LRS    (9)         QRS   (39)      . UNPACK Y POSITION
      STQ    (IY)        LRS   (9)       . STORE Y POSITION
      QRS    (39)        STQ   (IX)      . STORE X POSITION
      LRS    (18)        QRS   (30)      . UNPACK MANEUVER LIMIT
      STQ    (LIMIT)     LRS   (4)       . STORE MANEUVER LIMIT
      QRS    (44)        STQ   (NR)      . STORE TRACK NUMBER
      LRS    (5)         QRS   (43)      . UNPACK MANEUVER CODE
      STQ    (KMAN)      LDA   (NR)      . STORE MANEUVER CODE
      SCL    (MASK7)     STA   (NR)      . MASK TRACK NUMBER
      ENI 1  (1)         ENI 2 (1)       . SHIFT
      INI 2  (1)         LDA 2 (LHEAD)   .   LHEAD
      STA 1  (LHEAD)     ENI   (0)       .     ARRAY
      ISK 1  (59)        SLJ   (L-2)     .       UP BY
      ENA    (0)         STA   (LHEAD+60).         ONE CELL
      LDA    (KMAN)      INA   (-1)      . SELECT MANEUVER
      AJP 1  (7K)        LDA   (IX)      . START TRACK
      MUI    (ISCALE1)   STA   (IX)      . SCALE X POSITION
      LDA    (IY)        MUI   (ISCALE1) .
      STA    (IY)        ENA   (120B)    . SCALE Y POSITION
      MUI    (ISCALE2)   STA   (IZ)      . SET ALTITUDE TO 80 FEET
```

```
      ENA   (170B)     STA   (ISPD)   . SET SPEED TO 120 KNOTS      00735
      LDA   (LIMIT)    STA   (ICRS)   . STORE ORDERED CPURSE        00736
      SLJ   (1K)       ZRO   (0)      .                             00737
7K    INA   (-1)       AJP   (L+3)    .   STARBOARD TURN            00740
      INA   (-1)       AJP   (L+6)    .   CLIMB/DIVE                00741
      INA   (-1)       AJP   (L+7)    .   CHANGE SPEED              00742
      LDA   (LIMIT)    STA   (ICRS)   .   TURN OR ERROR             00743
      LDA   (TEMP)     SLJ 4 (FLOAT)  . FORM                        00744
      FDV   (10.)      STA   (RATE)   .   MANEUVER RATE             00745
      SLJ   (1K)       ZRO   (0)      .                             00746
      LDA   (LIMIT)    STA   (IALT)   .   CLIMB/DIVE                00747
      SLJ   (L+2)      ZRO   (0)      .                             00750
      LDA   (LIMIT)    STA   (ISPD)   .   CHANGE SPEED              00751
      LDA   (TEMP)     SLJ 4 (FLOAT)  . FORM MANEUVER RATE          00752
      STA   (RATE)     SLJ   (1K)     .                             00753

      END
```

1. Identification.

Title: PRINT

Category: On Line Display

Programmer: E. L. Borden

Organization: U. S. Naval Postgraduate School

Date: April 1965

2. Purpose.

This subroutine is designed to allow users to display characters and/or vectors on the Data Display model dd 65. The routine enables the user to display data without the trouble of packing special words for the dd 65.

3. Usage.

3.1 Normal Operation.

3.1.1 PRINT provides for entry of the medium size characters. Vectors of any size allowed will be entered by PRINT.

3.1.2 In order to illustrate the usage of PRINT, assume that you wanted to output to the dd 65 the following message:

"NO TRACKS UNDER YOUR CONTROL."

In the memory of the 1604 you would have the following stored:

Address	Contents of cell
22222	4546202351616342
22223	2220244564655120
22224	3046245120634645
22225	2351464373000000

PRO = 1 - right hand tube
0 - left hand tube

ADD = Address in dd 65 memory where message to be stored

X = X position of First Character/Vector

Y = Y position of First Character/Vector

To display the message on the dd 65 the following set of coding is necessary:

```
ENA   (X)           SAL   (PRINT+1)  .
ENA   (Y)           SAL   (PRINT+2)  .
ENA   (PRO)         SAL   (PRINT+3)  .
ENA   (ADD)         SAL   (PRINT+4)  .
ENA   (22222B)      ENQ   (0)        .
SLJ 4 (PRINT)       ZRO   (0)        .
```

The address of the message to be displayed is entered in the Accumulator and the Q-Register is set to zero for character mode or to minus zero for vector mode. If the argument PRINT+4 is deleted the address will be the next available cell in the dd 65 memory.

3.1.3 When the main program executes the subroutine, the subroutine assembles the message and packs it into a designator word and subsequent string words. The packing ends when the end of the message is sensed. To sense the end of the message, PRINT uses two codes. The code 00 is used to sense the end of a character message since this code is illegal. The code 77 is used to sense the end of a vector message since this code is an illegal vector code.

3.1.4 PRINT has a built in sense for end of line and end of page. Before any character or vector is assembled into a word the end of line is sensed. If the end of the line has been reached, the subroutine will reset the X position to an initial position of 400_8 and the Y position will then be checked for end of page. If the end of page has not been reached, the Y position will be decreased by 12_8 and a new designator word will begin with the new X and Y positions. If the end of page is sensed, then the Y

position will be reset to an initial value of 377_8. At this time the new designator word will be formed. The routine also checks to insure that memory is not overflowed. Before packing any word, a memory check is made and if it is found that memory has overflowed, then the memory address is reset to 002_8 and a new designator word is started.

APPENDIX III-B

Subroutine PRINT

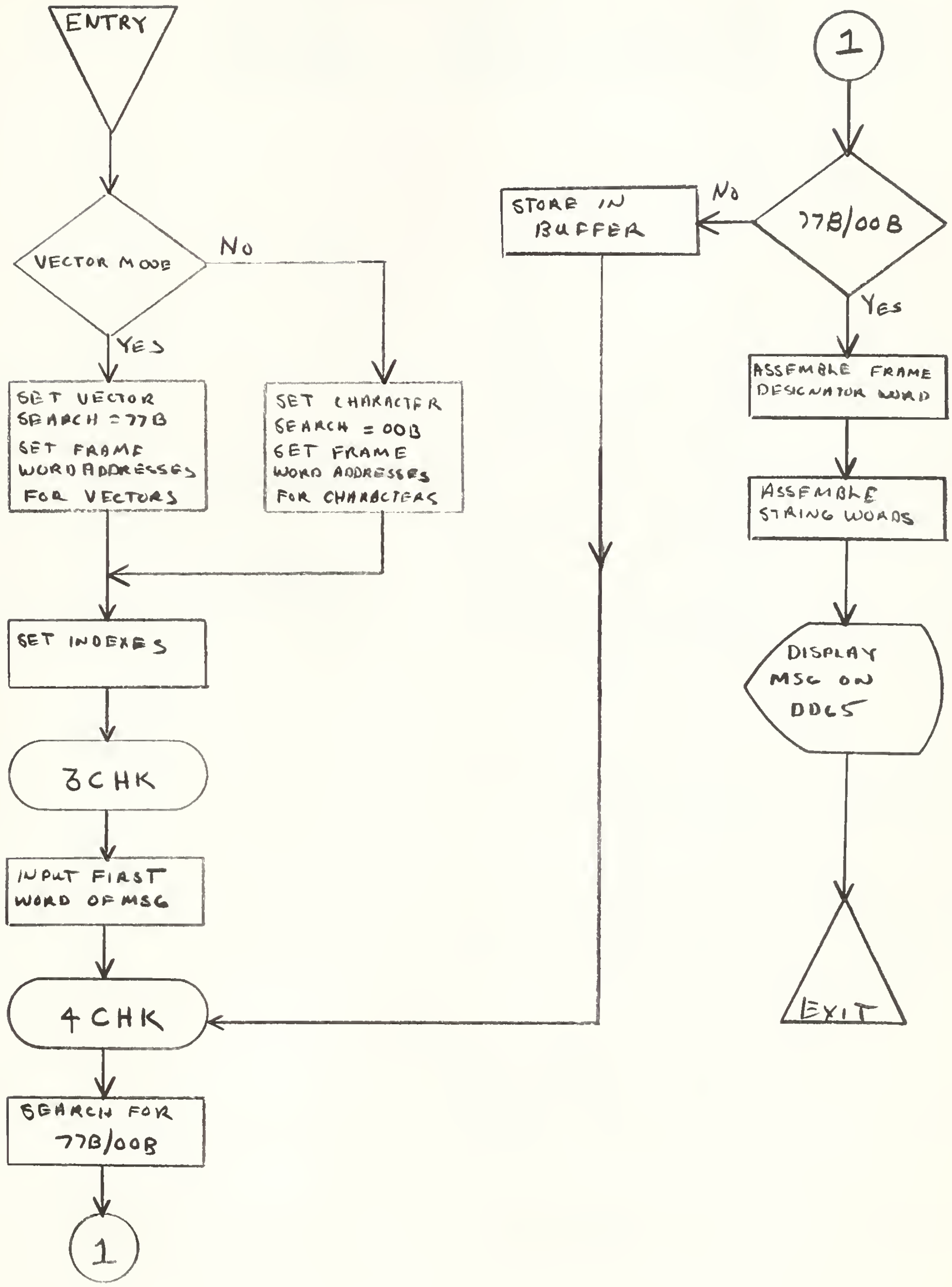


Figure III-1

APPENDIX III-C

SUBROUTINE PRINT PROGRAM LISTING

```
       MACHINE PRINT (1A,2A,3A,4A)

       RSV(BUF=121,OUTPUT=18)

       CON(DW= 3000000000000000B, LTFWC = 1000B, RTFWC = 5000B,V1 = 776B,
      *     LTFWV = 3000B, RTFWV = 7000B, LTSWC = 0B, RTSWC = 4000B,
      *     LTSWV = 2000B, RTSWV = 6000B, XI = -377B, YI = 377B, V = 77B,
      *     MASK1 = 7777777777777000B, DX = 10B, DY=12B)

1RET   SLJ    (N)          SLJ    (1CHK)     . EXIT / ENTRY                     00000
1A     ZRO    (0)          ZRO    (0)        . X-POSITION                       00001
2A     ZRO    (0)          ZRO    (0)        . Y-POSITION                       00002
3A     ZRO    (0)          ZRO    (0)        . 0-LEFT TUBE    1-RT TUBE         00003
4A     ZRO    (0)          ZRO    (0)        . MEMORY ADDRESS                   00004

* SET UP SEARCH WORDS

1CHK   SIU 2 (1SAVE)       SIL 3 (1SAVE)     . SAVE INDEXES                     00005
       SIU 4 (2SAVE)       SIL 5 (2SAVE)     .   2,3,4,5                        00006
       STA   (TEMP)        QJP 3 (2CHK)      . VECTOR OR CHARACTER MODE         00007
       ENA   (LTSWC)       SAU   (5CHK)      . SET SEARCH = ZERO                00010
       ENA   (LTFWC)       SAL   (1C1)       . SET FIRST WORD                   00011
       ENA   (LTSWC)       SAU   (2C1)       . SET STRING WORD                  00012
       SLJ   (L+4)         ZRO   (0)         .                                  00013
2CHK   ENA   (V)           SAU   (5CHK)      . SET SEARCH = 77B                 00014
       ENA   (LTFWV)       SAL   (1C1)       . SET FIRST WORD                   00015
       ENA   (LTSWV)       SAU   (2C1)       . SET STRING WORD                  00016
       ENI 2 (1)           ENI 3 (7)         . B2 = BUFFER INDEX, B3 = LETTER   00017

* FILL BUFFER SECTION

3CHK   LDQ 7 (TEMP)        ENA   (0)         .                                  00020
4CHK   LLS   (6)           ENI   (0)         .                                  00021
5CHK   EQS   (N)           SLJ   (6CHK)      . SEARCH FOR 00/77                 00022
       SLJ   (7CHK)        ZRO   (0)         .                                  00023
6CHK   STA 2 (BUF)         ENA   (0)         . STORE IN BUFFER                  00024
```

```
       ISK 2 (121)      SLJ   (L+2)    . CHECK FOR FULL BUFFER              00025
       SLJ   (7CHK)     ZRO   (0)      . JUMP IF BUFFER FILLED              00026
       IJP 3 (4CHK)     ENI   (0)      . LOOP FOR 8 CHARACTERS              00027
       RAO   (TEMP)     ENI 3 (7)      . INCREASE WORD ONE ADDRESS          00030
       ENA   (0)        SLJ   (3CHK)   .                                    00031
7CHK   INI 2 (-1)       LDA   (4A)     .                                    00032

* ASSEMBLY OF DESIGNATOR WORD

       SUB   (V1)       AJP 6 (1MEM)   . CHCEK FOR MEMORY OVERFLOW          00033
8CHK   ENI 4 (18)       LDA   (DW)     . ASSEMBLE FIRST WORD FRAME          00034
       STA 4 (OUTPUT)   LDA 7 (1A)     .                                    00035
       STA   (X)        SCL   (MASK1)  .                                    00036
       ALS   (36)       RAD 4 (OUTPUT) . PACK INITIAL X POSITION            00037
       LDA 7 (2A)       STA   (Y)      .                                    00040
       SCL   (MASK1)    RAD 4 (OUTPUT) . PACK INITIAL Y POSITION            00041
       LDA   (4A)       ALS   (13)     .                                    00042
       RAD 4 (OUTPUT)   ENI 5 (1)      .                                    00043
       SIL 2 (CHECK)    SLJ   (1C)     . SET CHECK                          00044

*RESET MEMORY TO 002 SECTION

1MEM   SLJ   (N)        ENA   (2)      . RESET MEMORY                       00045
       STA   (4A)       SLJ   (1MEM)   .                                    00046

* ASSEMBLY OF STRING WORDS

1C     LDA 5 (BUF)      ALS   (30)     .                                    00047
       RAD 4 (OUTPUT)   SLJ 4 (1XY)    .                                    00050
       INI 5 (1)        LDA 5 (BUF)    .                                    00051
       ALS   (24)       RAD 4 (OUTPUT) . CHECK X-Y POSITION                 00052
1C1    INI 5 (1)        ENA   (N)      .                                    00053
       ADD   (3A)       STA   (TEMP)   . ASSEMBLE STRING WORD FRAME         00054
       LDA 7 (TEMP)     RAD 4 (OUTPUT) . PACK MODE, TUBE,AND STRING BIT     00055
       INI 4 (-1)       ENA   (2)      .                                    00056
       RAD   (4A)       ENA   (-2B)    .                                    00057
```

```
        RAD    (CHECK)       AJP    (1OUT)      . CHECK FOR END OF MSG              00060
        AJP 3  (1OUT)        ENI    (0)         .                                   00061
2C      LDA    (4A)          SUB    (V1)        .                                   00062
        AJP 6  (1MEM)        ENI    (0)         . CHECK FOR MEMORY OVERFLOW         00063

* FILL STRING WORDS

2C1     ENA    (N)           ENI    (0)         .                                   00064
        ADD    (3A)          STA    (TEMP)      .                                   00065
        LDA 7  (TEMP)        STA 4  (OUTPUT)    .                                   00066
        RSO    (CHECK)       SLJ 4  (1XY)       .                                   00067
        LDA 5  (BUF)         ALS    (42)        .                                   00070
        RAD 4  (OUTPUT)      INI 5  (1)         . FIRST CHARACTER                   00071
        RSO    (CHECK)       SLJ 4  (1XY)       .                                   00072
        LDA 5  (BUF)         ALS    (36)        .                                   00073
        RAD 4  (OUTPUT)      INI 5  (1)         . SECOND CHARACTER                  00074
        RSO    (CHECK)       SLJ 4  (1XY)       .                                   00075
        LDA 5  (BUF)         ALS    (30)        .                                   00076
        RAD 4  (OUTPUT)      INI 5  (1)         . THIRD CHARACTER                   00077
        RSO    (CHECK)       SLJ 4  (1XY)       .                                   00100
        LDA 5  (BUF)         ALS    (24)        .                                   00101
        RAD 4  (OUTPUT)      INI 5  (1)         . FOURTH CHARACTER                  00102
        RSO    (CHECK)       SLJ 4  (1XY)       .                                   00103
        LDA 5  (BUF)         ALS    (18)        .                                   00104
        RAD 4  (OUTPUT)      INI 5  (1)         . FIFTH CHARACTER                   00105
        RSO    (CHECK)       SLJ 4  (1XY)       .                                   00106
        LDA 5  (BUF)         ALS    (12)        .                                   00107
        RAD 4  (OUTPUT)      INI 5  (1)         . SIXTH CHARACTER                   00110
        RSO    (CHECK)       SLJ 4  (1XY)       .                                   00111
        LDA 5  (BUF)         RAD 4  (OUTPUT)    . SEVENTH CHARACTER                 00112
        INI 5  (1)           IJP 4  (3C)        .                                   00113
        SLJ    (1OUT)        ZRO    (0)         .                                   00114

* INCREMENT MEMORY ADDRESS

3C      ENA    (2)           RAD    (4A)        .                                   00115
```

```
        LDA    (CHECK)     INI 4 (1)       . CHECK FOR END OF MSG              00116
        AJP 3 (1OUT)       AJP   (1OUT)    .                                   00117
        INI 4 (-1)         SLJ   (2C)      . LOOP FOR STRING WORDS             00120

* X-Y POSITION CHECK

1XY     SLJ    (N)         LDA   (CHECK)   .                                   00121
        AJP 3 (1OUT)       LDA   (DX)      . CHECK FOR END OF MSG              00122
        RAD    (X)         SUB   (377B)    . INCREASE X                        00123
        AJP 2 (L+1)        SLJ   (1XY)     . X OVERFLOW  NO-RETURN             00124
        LDA    (XI)        STA   (X)       .   YES - INCREASE Y                00125
        LAC    (DY)        RAD   (Y)       .                                   00126
        ADD    (400B)      AJP 3 (1XY2)    . Y OVERFLOW  JUMP IF YES           00127

* NEW DESIGNATOR WORD SECTION

1XY1    IJP 4 (1X)         SLJ   (1OUT)    .                                   00130
1X      LDA    (DW)        ENI   (0)       . ASSEMBLE DESIGNATOR WORD          00131
        STA 4 (OUTPUT)     ENQ   (777B)    .                                   00132
        LDL    (X)         ALS   (36)      .                                   00133
        RAD 4 (OUTPUT)     LDL   (Y)       . PACK X POSITION                   00134
        RAD 4 (OUTPUT)     ENA   (2)       . PACK Y POSITION                   00135
        RAD    (4A)        SUB   (V1)      .                                   00136
        AJP 6 (1MEM)       LDA   (4A)      . CHECK FOR MEMORY OVERFLOW         00137
        ALS    (13)        RAD 4 (OUTPUT)  . INCREMENT MEMORY ADDRESS          00140
        SLJ    (1C)        ZRO   (0)       .                                   00141

* RESET Y TO Y-INITIAL SECTION

1XY2    LDA    (YI)        STA   (Y)       .                                   00142
        SLJ    (1XY1)      ZRO   (0)       .                                   00143

* OUTPUT TO DD65 SECTION

1OUT    ENI 5 (1)          ENI   (0)       .                                   00144
2OUT    LDA 4 (OUTPUT)     STA 5 (OUTPUT)  .                                   00145
```

```
      INI 5 (1)          ENI   (0)        .                                00146
      ISK 4 (18)         SLJ   (2OUT)     . POSITION OUTPUT ARRAY          00147
      INI 5 (-1)         SIL 5 (KT)       .                                00150
      EXF   (02000B)     EXF   (04001B)   . STOP RT CLOCK   - DISALLOW     00151
      EXF   (77010B)     EXF 7 (77010B)   .                                00152
      OUT 5 (OUTPUT+1)   ENI   (0)        . OUTPUT TO DD65                 00153
      EXF   (04000B)     EXF   (01000B)   . ALLOW  -  START RT CLOCK       00154
      ENA   (2)          RAD   (4A)       . INCREASE MEMORY ADDRESS        00155
      ENI 5 (121)        ENA   (0)        . ZERO BUFFER                    00156
      STA 5 (BUF)        IJP 5 (L)        .                                00157
      ENI 5 (18)         ENA   (0)        .                                00160
      STA 5 (OUTPUT)     IJP 5 (L)        . ZERO OUTPUT BUFFER             00161
1SAVE ENI 2 (N)          ENI 3 (N)        . RESTORE INDEXES                00162
2SAVE ENI 4 (N)          ENI 5 (N)        .                                00163
      ENA   (0)          STA   (CHECK)    . SET CHECK TO ZERO              00164
      SLJ   (1RET)       ZRO   (0)        . RETURN                         00165

      END
```

APPENDIX IV-A

1. Identification.

Title: COURSE

Category: Course and Velocity determination

Programmer: E. L. Borden

Organization: U. S. Naval Postgraduate School

Date: April 1965

2. Purpose.

This subroutine is designed to take an x-velocity, a y velocity, and a z-position and determines a course in degrees true, a velocity in knots, and a velocity in mach number for any z-position greater than zero.

3. Usage.

3.1 Normal Operation

3.1.1 The normal calling sequence of COURSE is as follows:

VX = X-velocity in knots
VY = Y-velocity in knots
CRS = Course in degrees true
SPD = Speed in knots
MACH = Mach number
Z = Z-position in feet

```
ENA    (VX)          SAL    (COURSE+1)  .
ENA    (VY)          SAL    (COURSE+2)  .
ENA    (CRS)         SAL    (COURSE+3)  .
ENA    (MACH)        SAL    (COURSE+4)  .
ENA    (Z)           SAL    (COURSE+5)  .
ENA    (SPD)         SAL    (COURSE+6)  .
SLJ 4  (COURSE)      ZRO    (0)         .
```

3.1.2 COURSE determines course and speed in the normal manner. Mach number is determined by the following formula:

$$\text{MACH} = \frac{\text{SPD(knots)}}{661 - 2.5 \times \text{ALT(Kft)}}$$

The above formula is valid for ALT less than 35,000 feet. If ALT is greater than 35,000 feet, it is reduced to 35,000 feet.

APPENDIX IV-B

Subroutine COURSE Chart 1

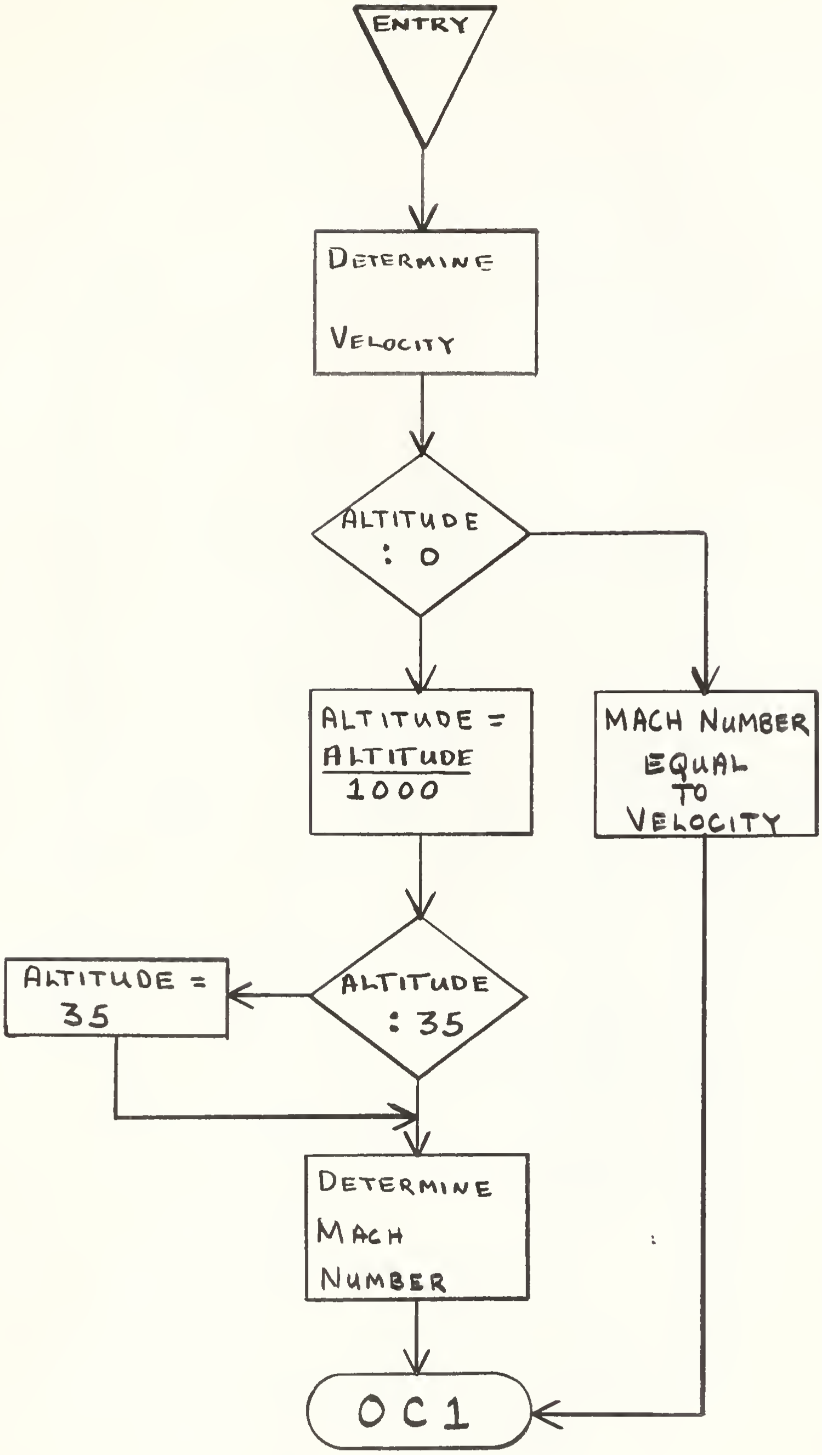


Figure IV-1

Subroutine COURSE Chart 2

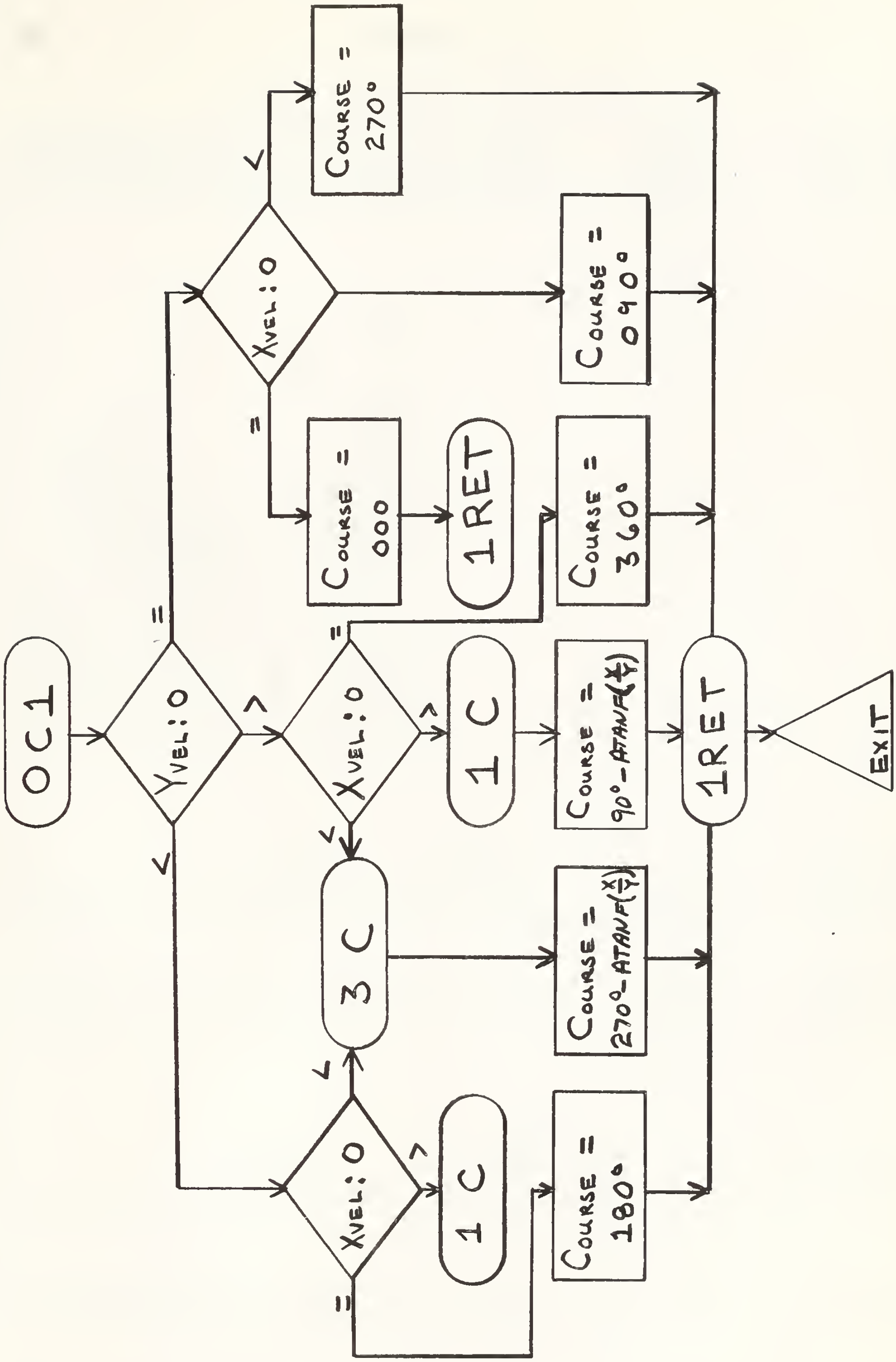


Figure IV-2

APPENDIX IV-C

SUBROUTINE COURSE PROGRAM LISTING

```
       MACHINE COURSE(1A,2A,3A,4A,5A,6A)
       LIB(SQRTF=SQ,ATANF=AT)
1RET   SLJ   (N)         SLJ   (L+7)    . EXIT/ENTRY                        00000
1A     ZRO   (0)         ZRO   (0)      . X VELOCITY IN KNOTS               00001
2A     ZRO   (0)         ZRO   (0)      . Y VELOCITY IN KNOTS               00002
3A     ZRO   (0)         ZRO   (0)      . COURSE IN DEGREES TRUE            00003
4A     ZRO   (0)         ZRO   (0)      . SPEED IN MACH NUMBER              00004
5A     ZRO   (0)         ZRO   (0)      . ALTITUED IN FEET                  00005
6A     ZRO   (0)         ZRO   (0)      . SPEED IN KNOTS                    00006
       LDA 7 (1A)        FMU 7 (1A)     . SQUARE X VELOCITY                 00007
       STA   (TEMP)      LDA 7 (2A)     .                                  000110
       FMU 7 (2A)        FAD   (TEMP)   . SQUARE Y VELOCITY                 00011
       STA   (TEMP)      ENA   (TEMP)   .                                   00012
       SAL   (SQ+1)      SLJ 4 (SQ)     . DETERMINE VELOCITY                00013
       STA 7 (6A)        LDA 7 (5A)     . CHECK Z-POSITION                  00014
       AJP   (0C)        AJP 3 (0C)     . JUMP IF Z .EQ. OR .LT. ZERO       00015
       FDV   (1000.)     FSB   (35.)    . DETERMINE MACH NUMBER             00016
       AJP 6 (0C2)       FAD   (35.)    .                                   00017
       FMU   (2.5)       STA   (TEMP1)  .                                   00020
       LDA   (661.)      FSB   (TEMP1)  .                                   00021
       STA   (TEMP1)     LDA 7 (6A)     .                                   00022
       FDV   (TEMP1)     STA 7 (4A)     .                                   00023

* DETERMINE COURSE

0C1    LDA 7 (2A)        AJP   (6C)     .                                   00024
       AJP 3 (4C)        LDA 7 (1A)     .                                   00025
       AJP   (2C)        AJP 3 (3C)     .                                   00026

* 000-090 DEGREES

1C     LDA 7 (2A)        FDV 7 (1A)     .                                   00027
       STA   (TEMP)      ENA   (TEMP)   .                                   00030
       SAL   (AT+1)      SLJ 4 (AT)     .                                   00031
       FMU   (57.3)      STA   (TEMP)   .                                   00032
       LDA   (90.)       FSB   (TEMP)   .                                   00033
```

```
        STA 7 (3A)        SLJ   (1RET)  .          00034

*  360 DEGREES

2C      LDA   (360.)      STA 7 (3A)    .          00035
        SLJ   (1RET)      ZRO   (0)     .          00036

*  180-270 DEGREES

3C      LDA 7 (2A)        FDV 7 (1A)    .          00037
        STA   (TEMP)      ENA   (TEMP)  .          00040
        SAL   (AT+1)      SLJ 4 (AT)    .          00041
        FMU   (57.3)      STA   (TEMP)  .          00042
        LDA   (270.)      FSB   (TEMP)  .          00043
        STA 7 (3A)        SLJ   (1RET)  .          00044

*  090 -180 DEGREES

4C      LDA 7 (1A)        AJP   (5C)    .          00045
        AJP 3 (3C)        SLJ   (1C)    .          00046

*  180 DEGREES

5C      LDA   (180.)      STA 7 (3A)    .          00047
        SLJ   (1RET)      ZRO   (0)     .          00050

*  270-360 DEGREES

6C      LDA 7 (1A)        AJP   (8C)    .          00051
        AJP 3 (7C)        LDA   (90.)   .          00052
        STA 7 (3A)        SLJ   (1RET)  .          00053

*  270 DEGREES

7C      LDA   (270.)      STA 7 (3A)    .          00054
        SLJ   (1RET)      ZRO   (0)     .          00055
```

```
* 000 DEGREES

8C      ENA    (0)           STA 7 (3A)     .          00056
        SLJ    (1RET)        ZRO   (0)      .          00057

* SURFACE OR SUBMARINE MACH NUMBER CLEARER

0C      LDA 7  (6A)          STA 7 (4A)     .          00060
        SLJ    (0C1)         ZRO   (0)      .          00061

* RESET Z TO 35000 FEET

0C2     SLJ    (N)           LDA   (35.)    .          00062
        SLJ    (L-1)         ZRO   (0)      .          00063

        END
```