26821-H002-R0-00

CR-151116

FORMULATION OF DETAILED CONSUMABLES MANAGEMENT MODELS FOR THE DEVELOPMENT (PREOPERATIONAL) PERIOD OF ADVANCED SPACE TRANSPORTATION SYSTEM

MBER 1976 CONTRACT NO. NAS9-14264

VOLUME I

DETAILED REQUIREMENTS FOR THE MISSION PLANNING PROCESSOR

Prepared by

L. C. Connelly

Systems Analysis Section

TRW
DEFENSE AND SPACE SYSTEMS GROUP

26821-H002-R0-00

Technical Report
for
Contract NAS 9-14264

Formulation of Detailed Consumables Management Models for the Development (Preoperational) Period of Advanced Space Transportation System

VOLUME I

DETAILED REQUIREMENTS FOR THE MISSION PLANNING PROCESSOR

November 1976

Prepared by
L. C. Connelly

Systems Analysis Section
TRW Defense and Space Systems Group
Houston, Texas

PREFACE

Future manned space programs that will have increased launch frequencies and reusable systems require an implementation of new consumables and systems management techniques that will relieve both the operations support personnel and flight crew activities. These techniques must be developed for the optimum combination of an onboard and ground support consumables management system consistent with the goals of the program. Effective operational performance of the consumables management techniques of a total system requires that a very explicit definition of the time, place, and method of performance of each function be determined by trade studies to ascertain that the operational methods do, indeed, meet these goals. This requires that the complete consumables management cycle be considered by including the mission planning and scheduling functions, prelaunch activities, onboard mission functions, ground mission support functions, and postmission activities.

Formulation of models required for the mission planning and scheduling function and establishment of the relation of those models to prelaunch, onboard, ground support, and postmission functions for the development phase of Space Transportation Systems (STS) was conducted under Contract NAS 9-14264 during the period 1 November 1975 to 31 October 1976. The preoperational Space Shuttle is used as the design baseline for the subject model formulations.

Analytical models were developed which consist of a Mission Planning Processor with appropriate consumables data base, a method of recognizing potential constraint violations in both the planning and flight operations functions, and a Flight Data File for storage/retrieval of information over an extended period which interfaces with a Flight Operations Processor for monitoring of the actual flights.

The Final Report for the Formulation of Detailed Consumables Management Models for the Development Period of Advanced Space Transportation Systems consists of an Executive Summary and five Technical Volumes. The Technical Volumes include information required for the implementation of a Consumables Management System. The individual volumes consist of:

Volume I. Detailed Requirements for the Mission Planning Processor

Volume II. Consumables Data Base Workbook

Volume III. Study of Constraints/Limitations for STS Consumables Management

Volume IV. Flight Data File Contents

Volume V. Flight Operations Processor Requirements

Two additional documents were issued in the course of the contract execution. These reports support the development of the Consumables Management System. The reports are:

Study of Existing Analytical Models for STS Consumables Management, dated February 1976.

Documentation of Computer Routines Developed to Determine Cyclic Probability (CYCPRO) Trends of Shuttle Heater Usage, dated September 1976.

This volume of the technical reports, Volume I, presents the detailed requirements for the Mission Planning Processor. The Mission Planning Processor is a user oriented tool for consumables management incorporating the models developed under this contract.

CONTENTS

LIST OF FIGURES

LIST OF FIGURES (Continued)

LIST OF TABLES

1.0 INTRODUCTION AND SUMMARY

The purpose of this report is to document the detailed requirements for the Mission Planning Processor. The Mission Planning Processor is a user oriented tool for consumables management and is part of the total consumables subsystem management concept presented in Reference 1.

A quasi top-down approach was applied to the design of the Mission Planning Processor. That is, interface requirements, input/output, and data base concepts were considered before computational processing. Existing analytical models (Reference 2) were investigated for applicability before new models (References 3 and 4) were developed. An overview of the Mission Planning Processor is presented in Section 2.0.

The Mission Planning Processor is being designed for an interactive system using demand terminals for input/output/display and interfacing with an updateable mission data bank. The user interface concept is presented in Section 3.0 and the data base handling concept is presented in Section 4.0.

The control and support routines, presented in Section 5.0, provide the user interface, peripheral data handling, program control, and support functions required by the Mission Planning Processor for execution on an interactive system. A description, interface requirements, definition of internal variables, listing of input data, processing flow diagram, and listing of output data are presented for each routine.

The computational routines perform specific manipulations of consumables data base information. The computational routines are Space Shuttle consumables subsystems oriented and are presented in Section 6.0.

The detailed requirements for the Mission Planning Processor presented in this report are independent of computer hardware and programming language.

2.0 OVERVIEW OF THE MISSION PLANNING PROCESSOR

2.1 PURPOSE

Consumables management is a continuous process throughout the mission planning cycle from long-range planning through post-flight analysis. The Mission Planning Processor (Figure 1) is a user oriented tool for consumables management. The user need not be a consumables analyst. The Mission Planning Processor is being designed for an interactive system using demand mode terminals for input/output/display and interfacing with the updateable Flight Data Files. The files for each mission in the data bank are generated and used by the Mission Planning Processor. The amount of detail in the mission files is a function of where the mission lies in the planning cycle.

During long-range planning (Launch - 10 years) mission plans can be developed using discrete event data disassociated from the time of occurrence of the event. The effects of these events on consumable usage can be tallied by the Mission Planning Processor (RUN MODE = EVENT) to determine mission feasibility and consumable subsystem requirements. The results can be stored in the Flight Data Files (FILE 0) for recall.

During near-term planning (Launch - 6 years to Launch) the Mission Planning Processor can be used (RUN MODE = ACTIVE) to build and use mission plans with increasing detail and fidelity to mission time of events. The Mission Planning Processor will provide immediate feedback to the user concerning scheduling conflicts and consumable usage rate limit violations. The user has the option to generate and display event timelines, consumable usage versus mission time, and total consumables used and/or end of mission reserves for each consumable subsystem. The results can be stored in the Flight Data File (FILES 1, 2, and 3) for recall.

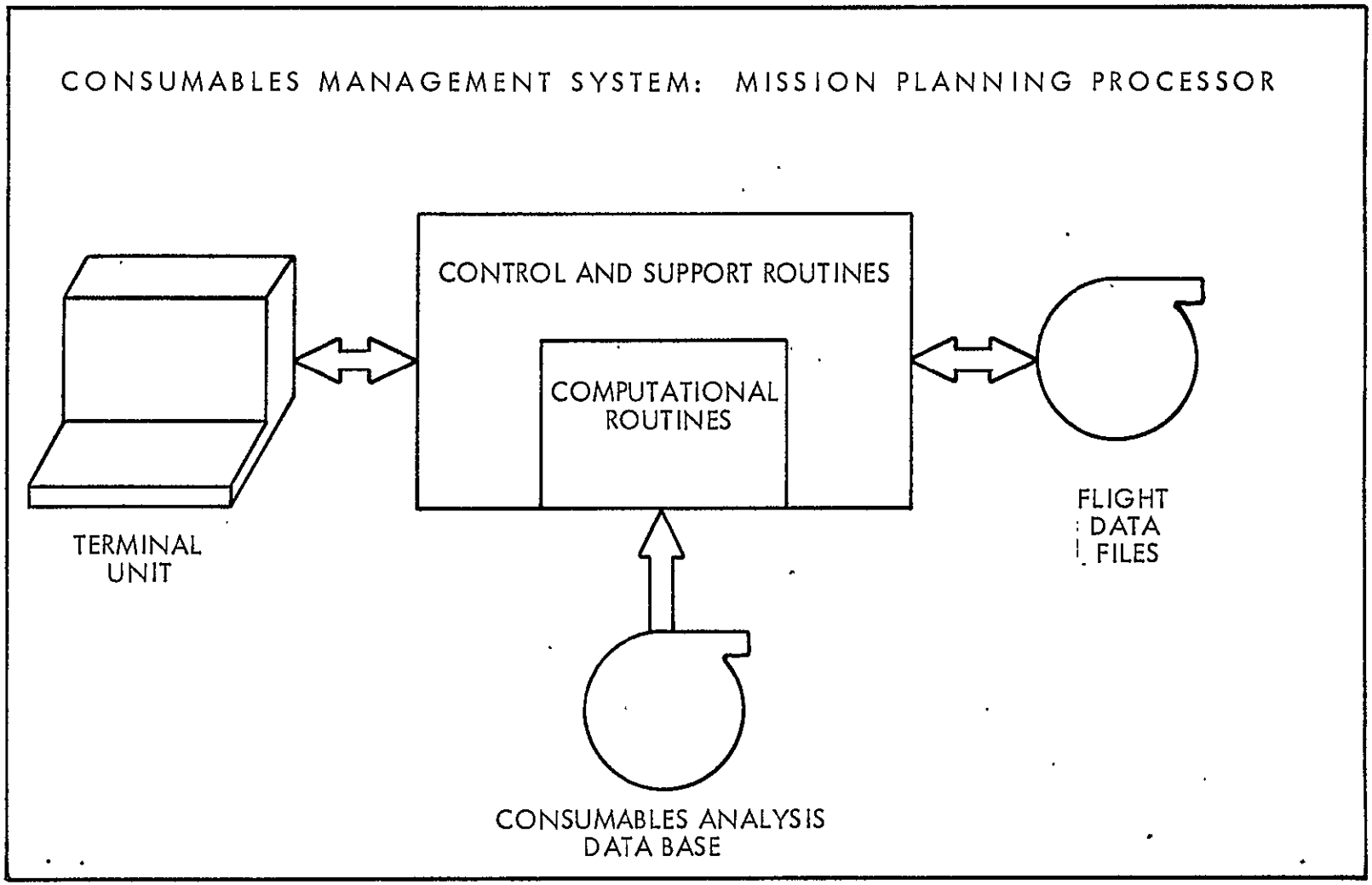


Figure 1. Mission Planning Processor Overview

2.2 FUNCTIONS

The Mission Planning Processor performs the following functions:

a) Provides user interface through interactive CRT displays

b) Generates total mission consumable requirements

c) Acts as a scheduler for mission events that affect consumable usage

d) Provides immediate feedback of scheduling conflicts

e) Provides immediate feedback of consumable usage rate violations

f) Generates and displays detailed consumable analysis data on user request

g) Stores selected generated data in the Flight Data Files on user request.

2.3 ELEMENTS

The Mission Planning Processor consists of the following elements:

a) The displays/user interface

b) The Flight Data Files

c) The consumables analysis data base

d) The control and support routines

e) The computational routines.

The user of the Mission Planning Processor need not be a consumables analyst to perform consumable management functions. The user interface is being designed to promote ease of input, immediate feedback of anomalies, and active display of a minimum data set. The user can request the generation and display of more detailed consumable analysis data if preferred, but the user need not understand the generation process. The user requests are serviced through a series of interactive CRT displays. These displays are discussed in detail in Section 3.0.

The Mission Planning Processor builds and uses the Flight Data Files defined in Volume IV of this report. The amount of detail in the files is a function of where the mission lies in the planning cycle. Four files are identified for each mission:

FILE 0 Contains data to reconstruct the event chart

FILE 1 Contains the minimum data set to operate the Mission Planning Processor in the ACTIVE mode

FILE 2 Contains detailed consumables versus time data for each subsystem

FILE 3 Contains detailed consumables versus time data for individual elements and distribution networks for each subsystem.

The consumables analysis data base contains the characteristic activity and subsystem usage rate data required by the Mission Planning Processor and is documented in Volume II of this report.

The control and support routines provide the user interface, peripheral data handling, program control, and support functions required by the Mission Planning Processor. The control and support routines are presented in detail in Section 5.0.

The computational routines calculate the specific consumables subsystem data required by the Mission Planning Processor. The computational routines are presented in Section 6.0.

2.4 INITIATION

In the EVENT MODE, the Mission Planning Processor requires the consumables analysis influence variables as input. There are two methods to introduce the influence variables into the program:

a) The first time the mission is executed, the influence variables are entered one at a time through the keyboard.

b) In subsequent executions the influence variables are entered from the FILE 0 data set stored in the Flight Data Files.

In the ACTIVE MODE, the Mission Planning Processor requires a mission timeline as input. There are two methods to introduce the mission timeline into the program as a function of where the mission lies in the planning cycle.

a) The first time the mission is executed, the timeline is entered event by event through keyboard entry. Even this mode is semi-automatic. Many standardized events (eat and sleep periods, etc.) are automatically scheduled as a function of the mission configuration.

b) In subsequent executions the mission timeline is entered from the FILE 1 data set stored in the Flight Data Files.

2.5 EXECUTION

In the EVENT MODE the influence variables, regardless of initiation method, are used to calculate consumable usage and requirements. The results may be displayed and/or stored in the Flight Data Files.

In the ACTIVE MODE, the mission timeline, regardless of initiation method, is used to create consumable usage rate blocks. Any scheduling conflicts or rate violations will be fed back to the user and stored in conflict tables for later assessment. At this time in the execution, the user may elect to generate and display detailed consumable analysis data or modify the existing mission timeline.

Mission timeline modification is accomplished by user input through a set of interactive displays. These displays are discussed in Section 3.1. The user may change the start and stop times of mission phases, schedule new events, modify existing events, or unschedule existing events. For each change in the mission timeline, consumable usage rate blocks are built for each consumable subsystem affected by the change. Any scheduling conflicts or rate violations will be fed back to the user and stored in conflict tables for later assessment. At this time in the execution, the user may elect to generate and display detailed consumable analysis data and/or store selected data in the Flight Data Files.

The following detailed consumable analysis data can be generated and displayed at user option:

a) An event chart that lists the number and types of events scheduled without reference to mission time.

b) A timeline listing scheduled events versus mission time without reference to consumables usage.

c) The consumables usage versus time for each consumables subsystem.

d) The total consumables used and end-of-mission quantities for each consumables subsystem.

2.6 OUTPUT

The results of the Mission Planning Processor execution can be output via CRT displays at the user terminal and/or stored in the Flight Data Files. The output displays are discussed in Section 3.2. The Flight Data Files have been addressed previously in this section.

3.0 DISPLAYS

The Mission Planning Processor is being designed for an interactive system and uses the CRT display unit at the user terminal as an integral part of the program. There are two types of Mission Planning Processor displays:

a) The interactive USER INTERFACE displays

b) The read only OUTPUT displays.

The use of these displays in the ACTIVE MODE is discussed in the following subsections.

3.1 USER INTERFACE DISPLAYS

The user interface with the Mission Planning Processor is through a set of interactive displays. Table I lists the user interface displays for the ACTIVE MODE. Except for the first display executed, the progression through the displays in the ACTIVE MODE is controlled by user input.

The first display executed is a function of the data option selected for the ACTIVE MODE. If the INITIAL data option is selected, the CONFIGURATION BLOCK display illustrated in Figure 2 is automatically executed. The user may enter the mission-dependent parameters identified on the display to create a mission configuration. On completion, the user enters the PROCEED instruction to continue the mission planning task. The remaining user interface displays are executed as discussed below for the RESTART data option.

If the RESTART data option is selected, the FLIGHT BLOCK display illustrated in Figure 3 is automatically executed. Entry of the line number corresponding to the requested mission phase will execute the PHASE BLOCK displays. The Ascent, On-Orbit, Deorbit, and Entry/Land phase block displays are illustrated in Figures 4 through 7, respectively. The FLIGHT BLOCK display is used for Prelaunch requests.

On the Prelaunch, Ascent, Deorbit, and Entry/Land displays the user can modify mission times. Any modifications will be noted on the display by the MOD flag. The modification is accomplished by entering the line number corresponding to the phase component to be modified and the time parameter value to be changed.

Table I. Active Mode Display Cross Reference Table

DISPLAY TYPE	DISPLAY NAME	LINE #	DISPLAY ID	ACTION ID
CONFIG	CONFIGURATION	-	90	-
FLIGHT	FLIGHT BLOCK	1	100	-1
PHASE BLOCK	ASCENT	2	200	-2, -5
PHASE BLOCK	ON-ORBIT	3	300	-6
PHASE BLOCK	DEORBIT	4	400	-7, -8
PHASE BLOCK	ENTRY/LAND	5	500	-9, -10
MENU	ORBITAL PHASE MENU	2	320	-
MENU	ORBITAL ACTIVITY MENU	4	340	-
ACTION	OMS MANEUVER	1	321	1
ACTION	RCS TRANSLATION	2	322	2
ACTION	ATTITUDE HOLD	3	323	3
ACTION	RENDEZVOUS	4	324	4
ACTION	STATION KEEPING	5	325	5
ACTION	DOCK	6	326	6
ACTION	UNDOCK	7	327	7
ACTION	PTC	8	328	8
ACTION	EVA	9	329	9
ACTION	IVA	10	330	10
ACTION	MANIPULATOR OPS	11	331	11
ACTION	IMU ALIGNMENT	12	332	12
ACTION	PAYLOAD BAY DOORS	1	341	13
ACTION	PAYLOAD CONSUMABLES	2	342	14
ACTION	COMPUTER	3	343	15
ACTION	TV	4	344	16
ACTION	DOWNLINK	5	345	17
ACTION	UPLINK	6	346	18
ACTION	FUEL CELL PURGE	7	347	19
ACTION	EAT	8	348	20
ACTION	SLEEP	9	349	21
ACTION	WASTE MANAGEMENT	10	350	22
ACTION	APU CHECKOUT	11	351	23

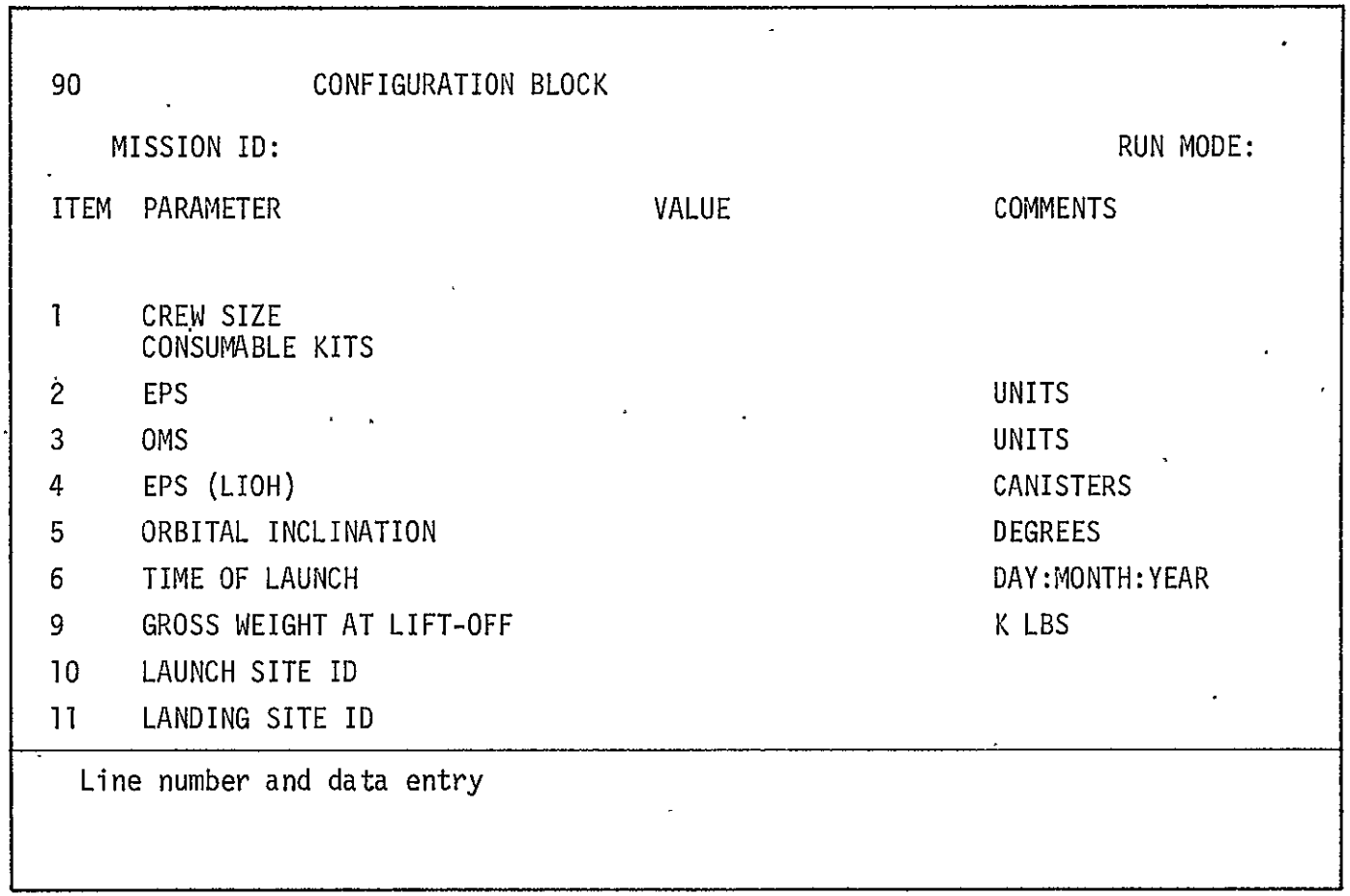


Figure 2. The Configuration Block Display Skeleton

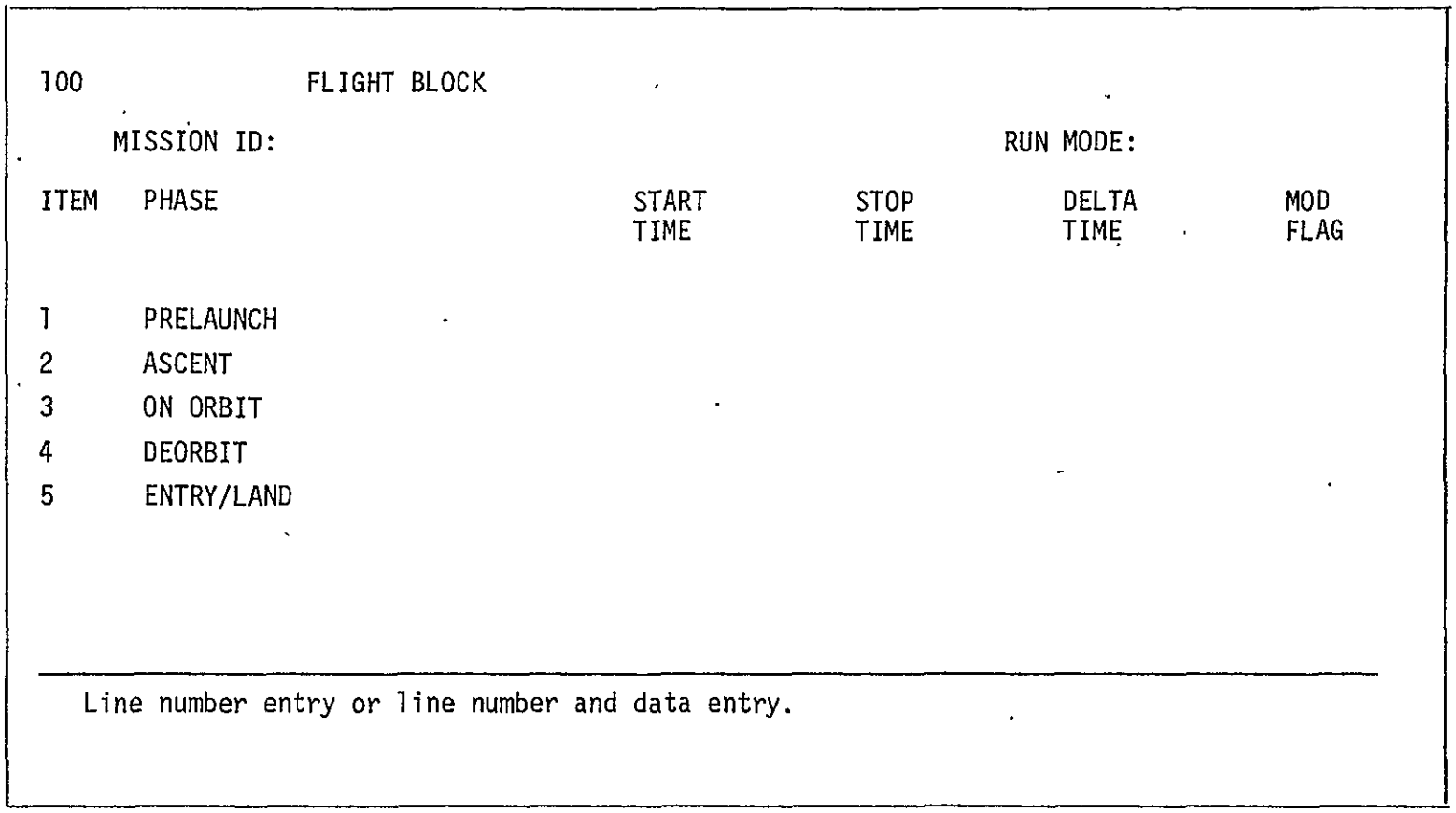


Figure 3. The Flight Block Display Skeleton

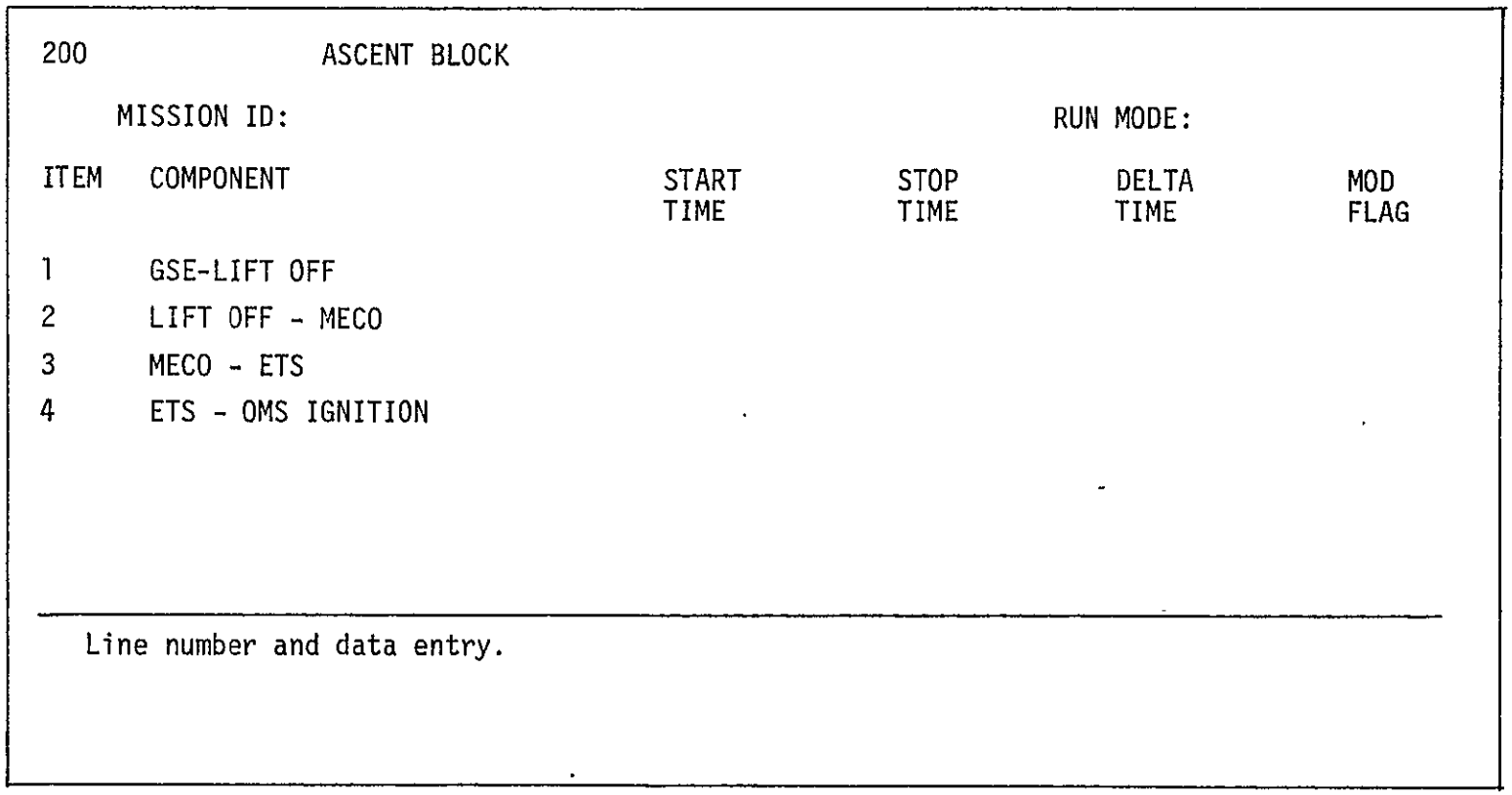


Figure 4. The Ascent Block Display Skeleton

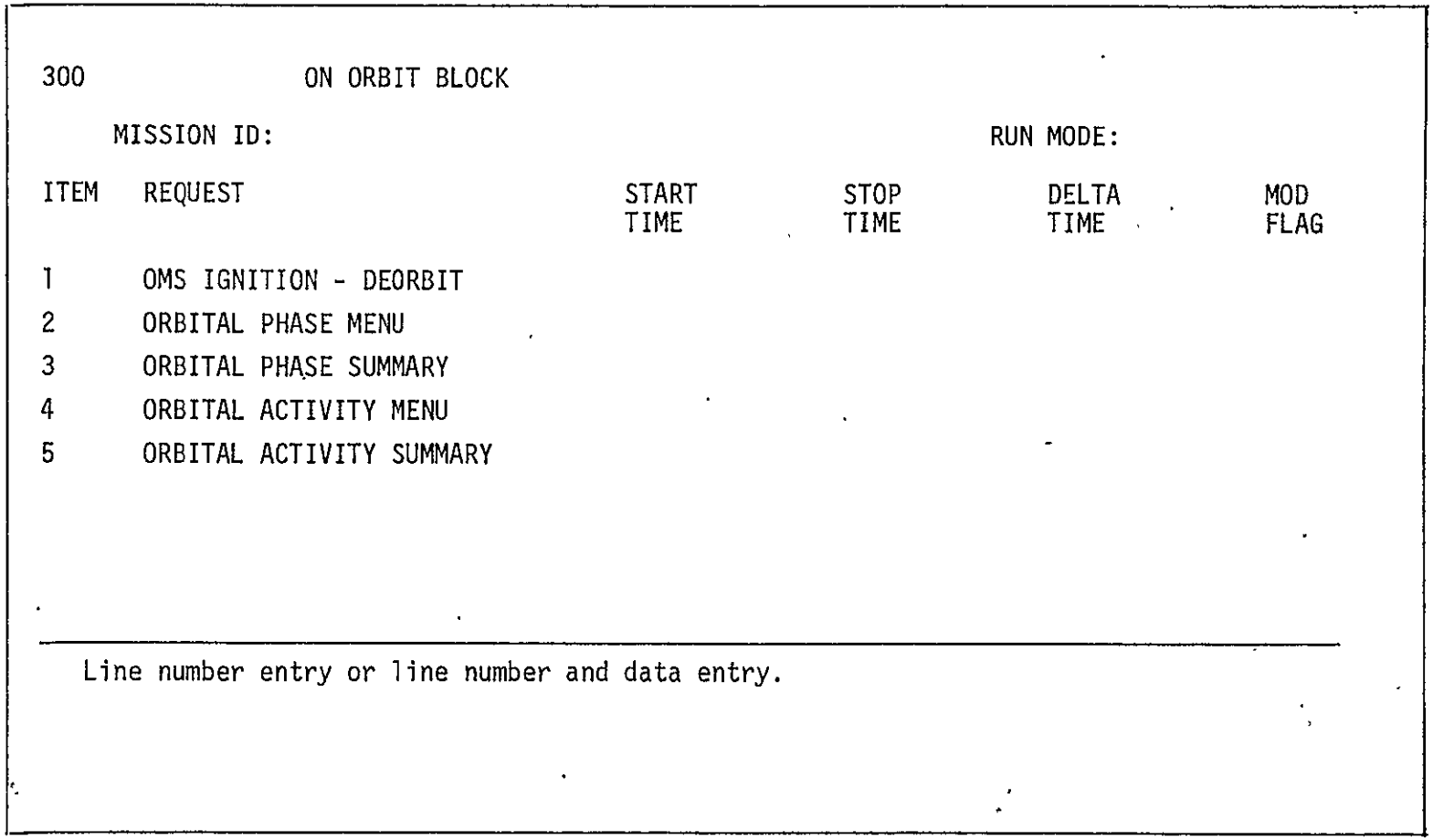


Figure 5. The On-Orbit Block Display Skeleton

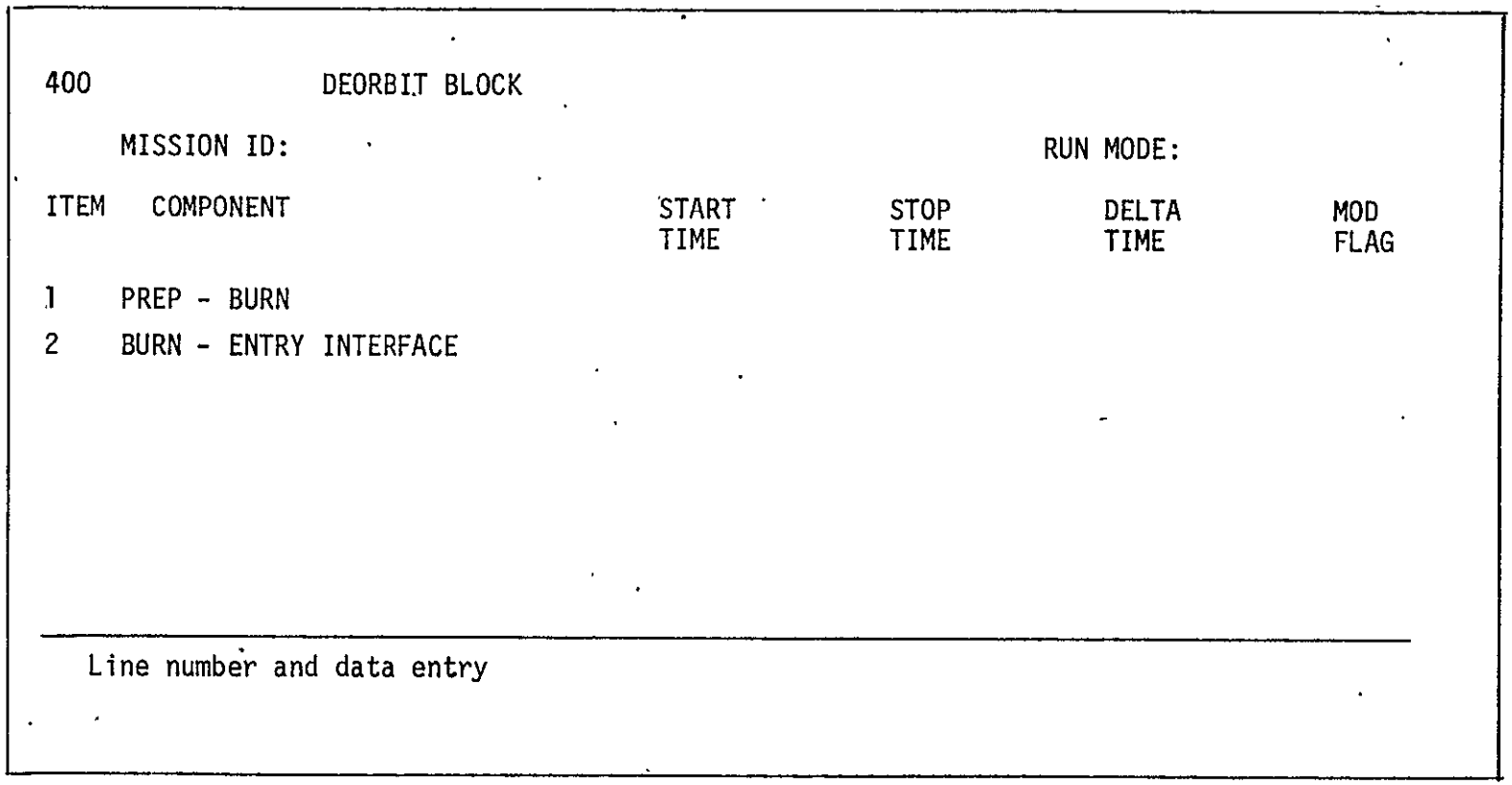


Figure 6. The Deorbit Block Display Skeleton

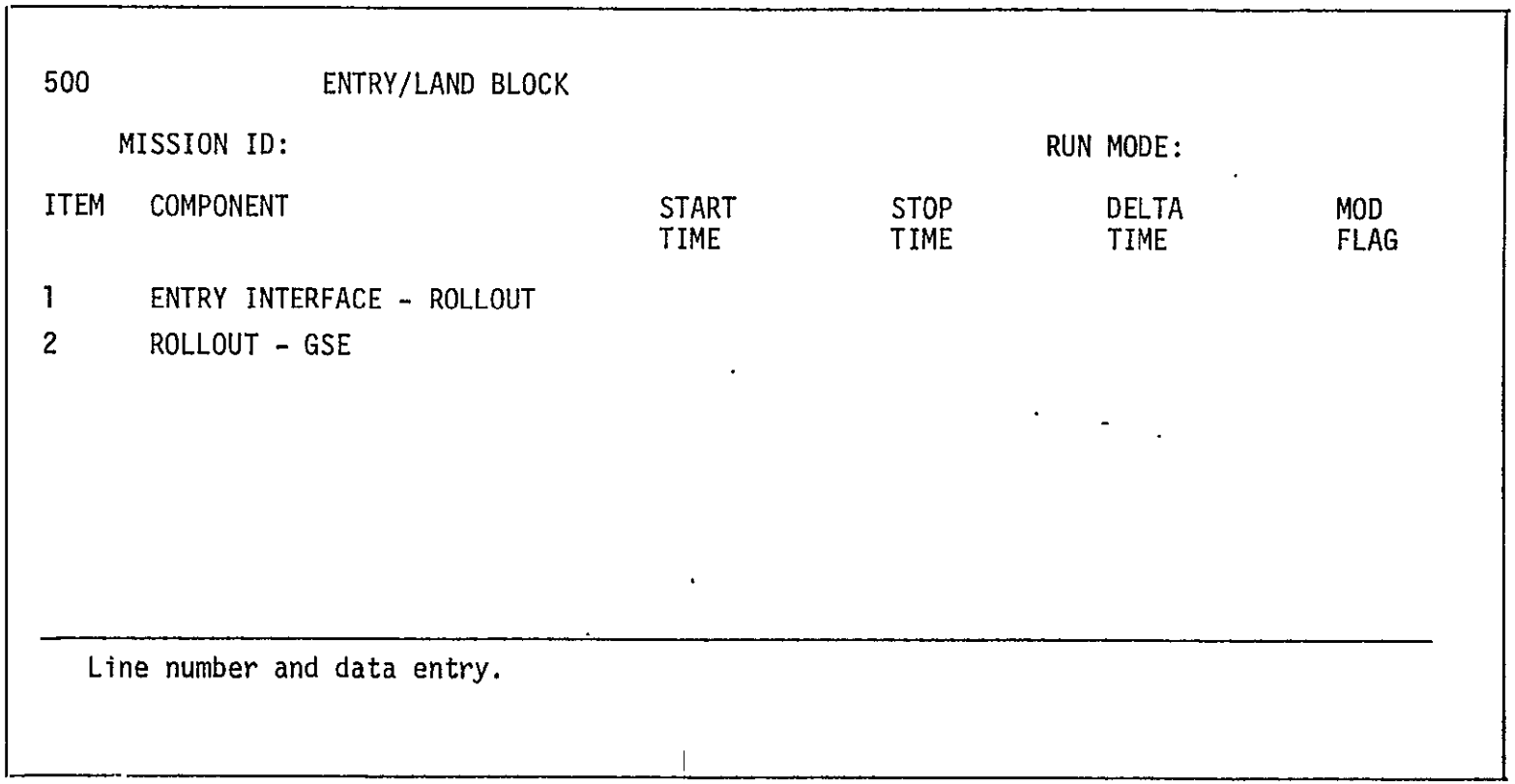


Figure 7. The Entry/Land Block Display Skeleton

The On-Orbit display allows the user more latitude in modifying the mission. The phase time can be changed on line number 1 in the same manner as the Prelaunch, Ascent, Deorbit, and Entry/Land displays. Entering line numbers 2 or 4 will cause the MENU displays to be executed. The MENU displays allow specific mission events to be selected for modification. The Orbital Phase Menu display is illustrated in Figure 8 and the Orbital Activity Menu display is illustrated in Figure 10. Entering line numbers 3 or 5 on the On-Orbit display will allow all the ACTION displays associated with the events listed on a MENU display to be executed in a read only sequence.

On the MENU displays, entering the line corresponding to the selected mission event will cause the ACTION display for the event to be executed. Figure 9 illustrates the ACTION display for the OMS Maneuver selected by entering line number 1 on the Orbital Phase Menu. Figure 11 illustrates the ACTION display for the Payload Door activities selected by entering line number 1 on the Orbital Activity Menu.

Each scheduled occurrence of an event is itemized on the ACTION display corresponding to the event. The user may modify the mission plan by entering the line number (n) of the itemized event occurrence to be modified as follows:

n	Modify occurrence n. The parameters to be changed are also entered.
-n	Unschedule occurrence n.
n+1	Schedule another occurrence of the event. The event data required are also entered.

The results of modifications entered through any display will be fed back to the user on that display. Modifications entered on any display that affect other displays will automatically update the other displays.

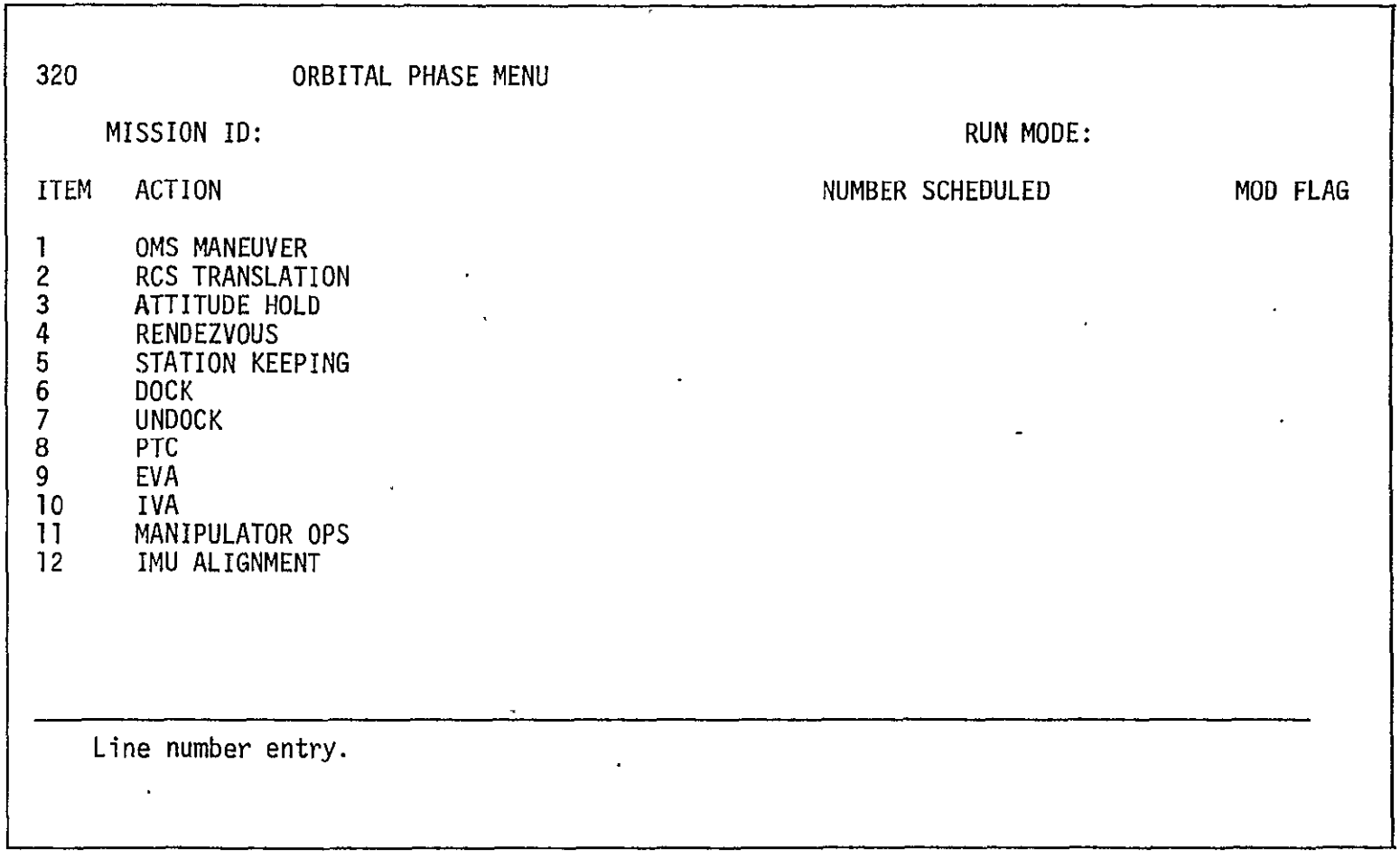


Figure 8. The Orbital Phase Menu Display Skeleton

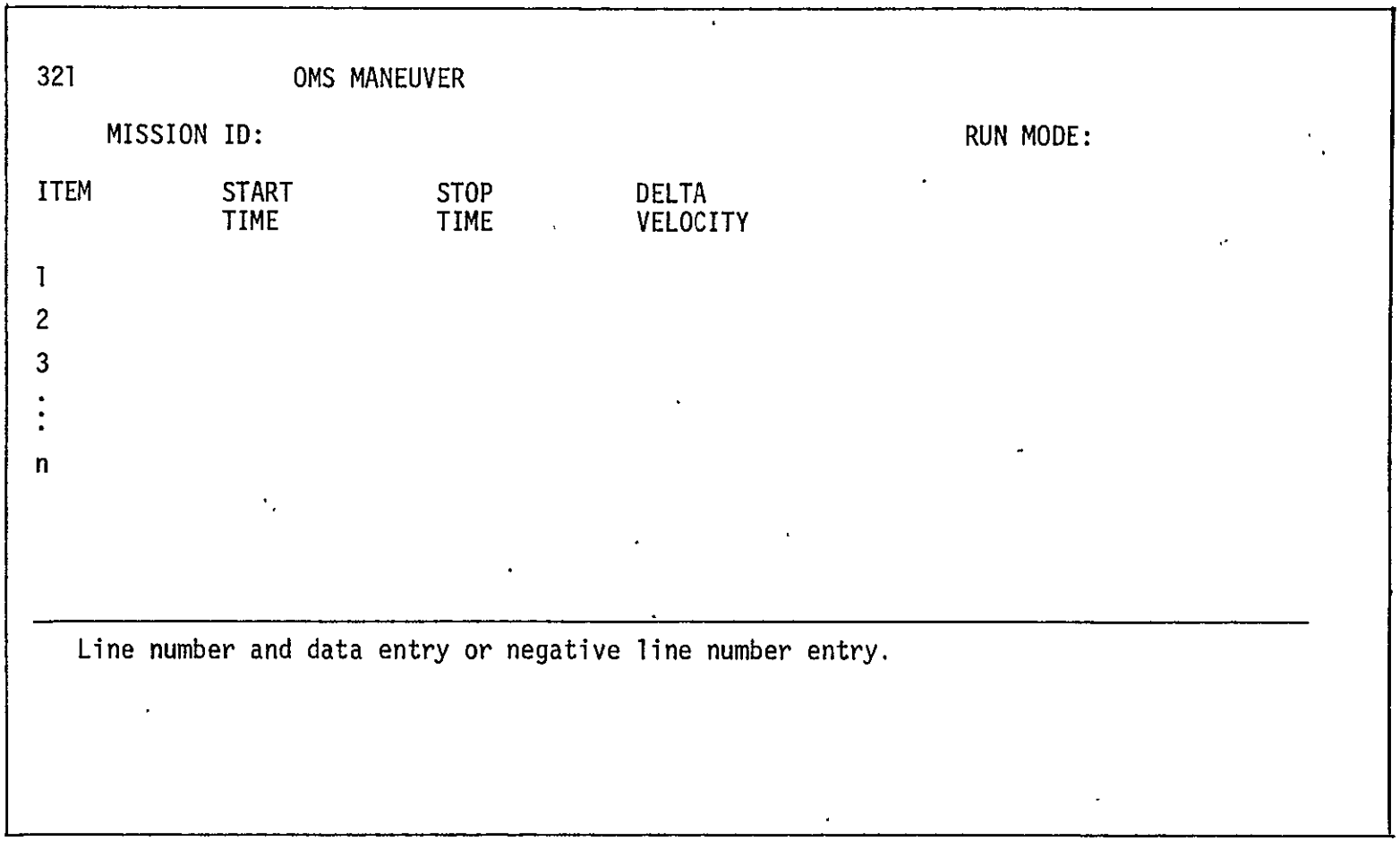


Figure 9. The OMS Maneuver Action Display Skeleton

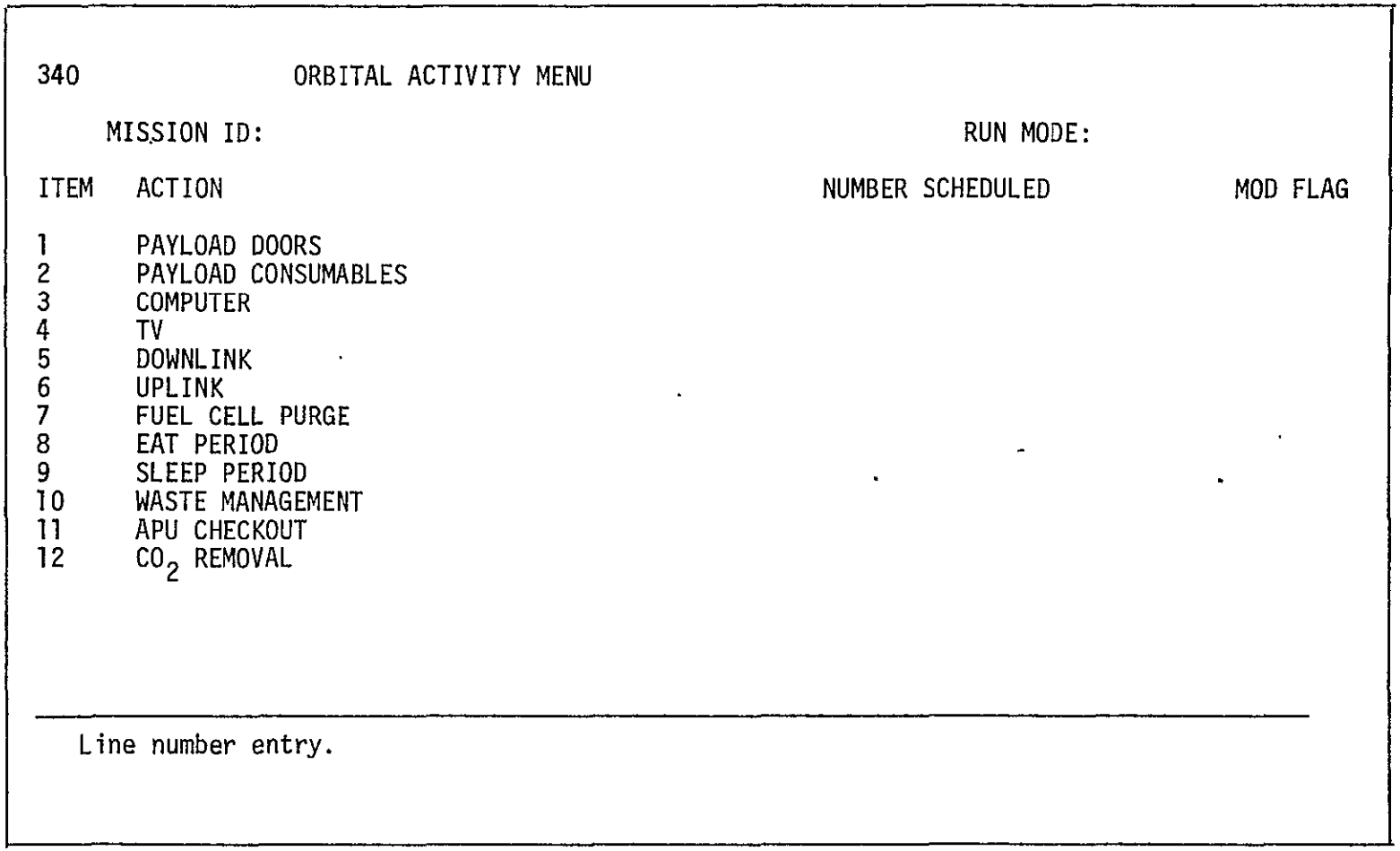


Figure 10. The Orbital Activity Menu Display Skeleton

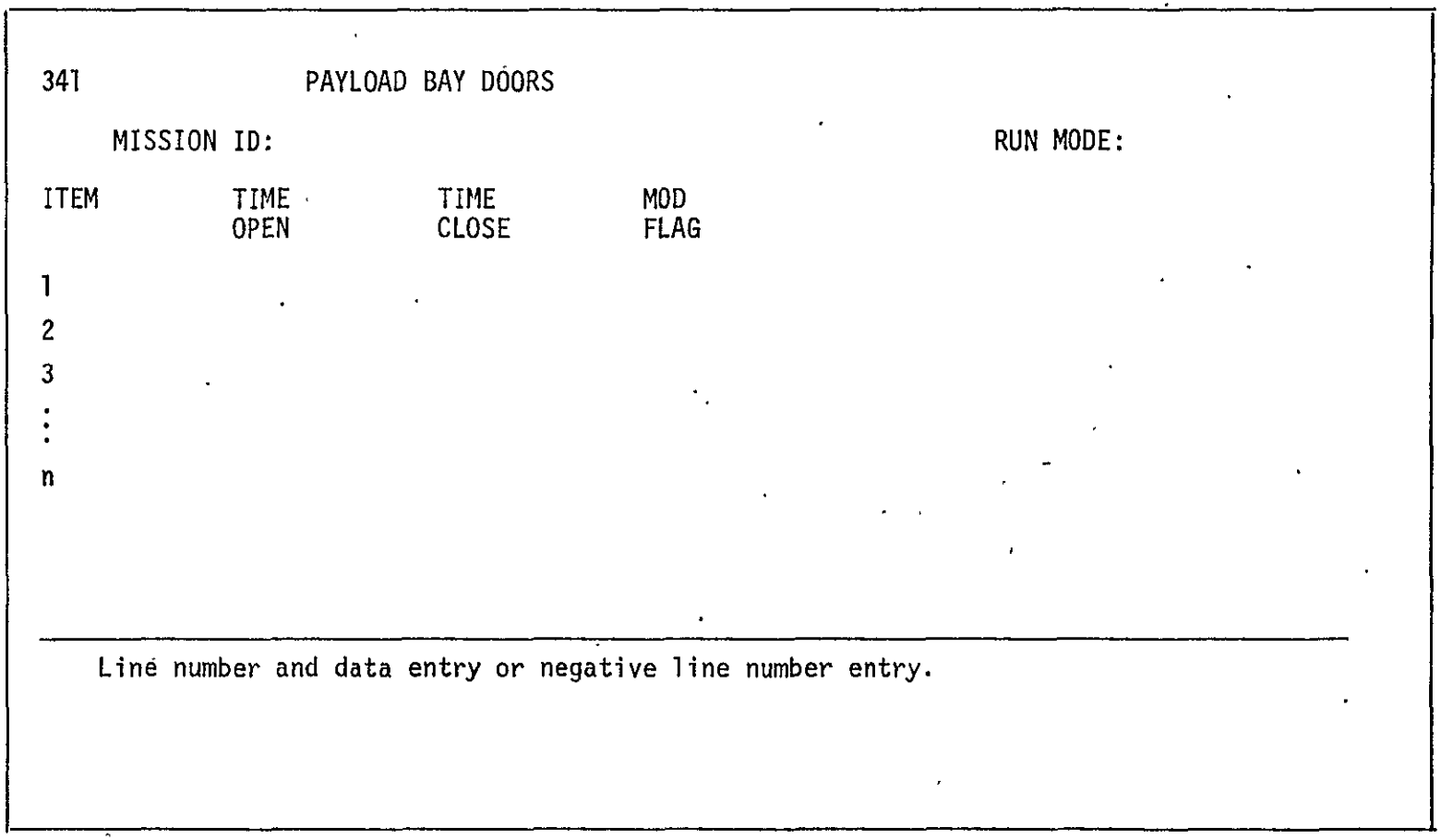


Figure 11. The Payload Bay Doors Action Display Skeleton

3.2 OUTPUT DISPLAYS

The user may request generation and display of more detailed consumable analysis data. Table II lists the Output display options. These displays are executed in a read only mode. The formats for these displays will be defined during program implementation.

Table II. Output Display Cross Reference Table

DISPLAY TYPE	DISPLAY NAME	LINE #	DISPLAY ID	ACTION ID
OUTPUT	CONFLICT TABLE	2	2000	-
OUTPUT	EVENT CHART	3	3000	-
EVENT	SUMMARY	1	3100	-
EVENT	INFLUENCE VARIABLE	2	3200	-
EVENT	OMS CONSUMABLES	3	3300	-
EVENT	RCS CONSUMABLES	4	3400	-
EVENT	EPS CONSUMABLES	5	3500	
EVENT	ECLSS CONSUMABLES	6	3600	-
OUTPUT	TIMELINE	4	4000	-
OUTPUT	CONSUMABLE HISTORY	5	5000	-
HISTORY	OMS HISTORY	1	5100	-
HISTORY	RCS HISTORY	2	5200	-
HISTORY	EPS HISTORY	3	5300	-
HISTORY	ECLSS HISTORY	4	5400	-
OUTPUT	CONSUMABLE QUANTITIES	6	6000	-

4.0 DATA BASE REQUIREMENTS

The Mission Planning Processor interfaces with three distinct data base areas:

a) The Flight Data Files

b) The consumables analysis data base

c) The COMPOOL.

The Flight Data Files contain the data generated and subsequently used by the Mission Planning Processor. The data files have been discussed in Section 2.3 and are defined in detail in Volume IV of this report.

The consumables analysis data base contains the characteristic event data (i.e., prep time, post time, consumables usage rates, etc.) and subsystem data required by the Mission Planning Processor. The data are defined in detail in Volume II of this report.

The COMPOOL is an active area of the data base. That is, the data required by the Mission Planning Processor during execution is temporarily stored in and passed through the COMPOOL.

5.0 DEFINITION OF THE CONTROL AND SUPPORT ROUTINES

The control and support routines for the Mission Planning Processor are defined in the following subsections. A description, interface requirements, definition of internal variables, listing of input data, processing flow diagram (if required), and listing of output data are presented for each routine.

The EXECUTIVE routine is presented first, followed by the remaining routines in alphabetical order. Figure 12 illustrates the control and support routine hierarchy illustrating the relationship between the individual routines.

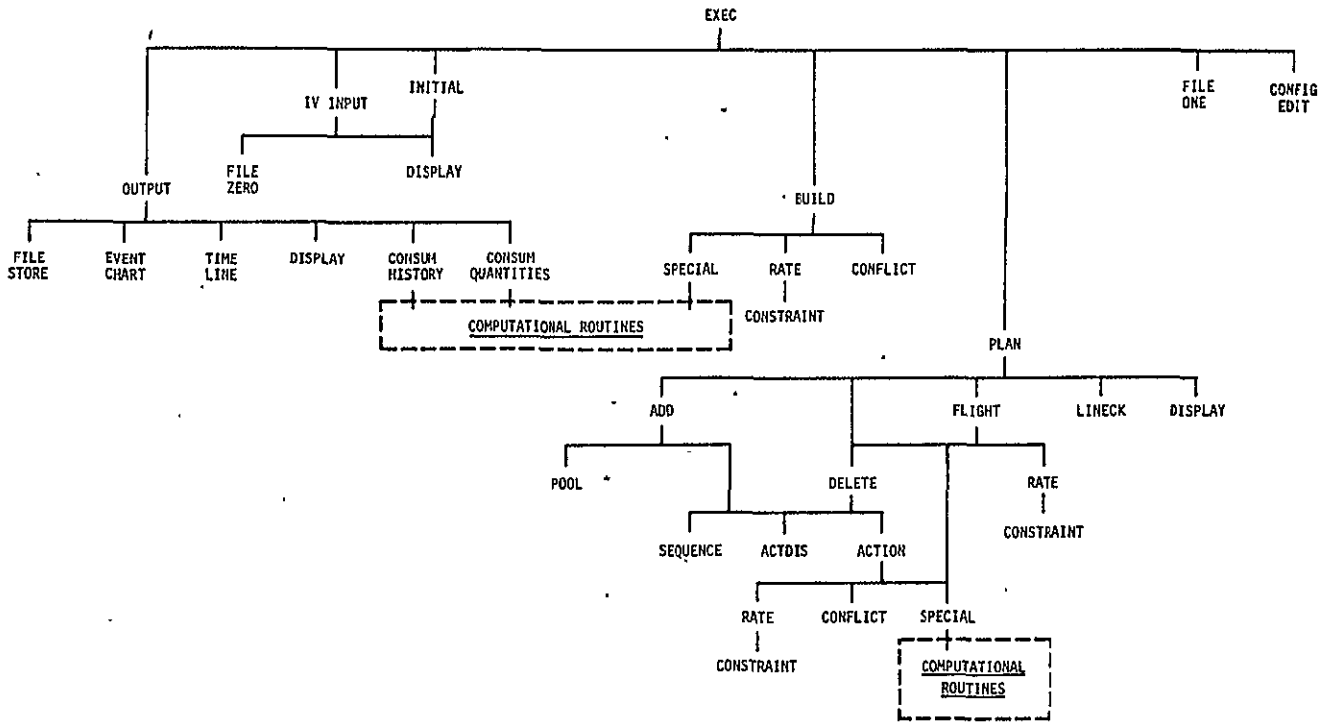


Figure 12. Control and Support Routine Hierarchy

5.1 EXEC ROUTINE

Description - The EXEC routine manages the Mission Planning Processor; calls the other control and support routines as directed by user instructions input through the keyboard unit; and provides tutorial/warning displays to the user through the CRT display unit.

Interface

I/O DEVICES - Terminal KEYBOARD and CRT units.
DATA BASE - COMPOOL for output only.
ROUTINES CALLING EXEC - None.
ROUTINES CALLED BY EXEC - IV INPUT, OUTPUT, FILE ONE, INITIAL, BUILD, CONFIG EDIT, and PLAN routines.

Internal Variables - None

Input - The EXEC routine requires the following instructions and data input through the KEYBOARD unit:

RUN MODE	Mission Planning Processor operation mode; EVENT - Computer aided event chart generation only. ACTIVE - Interactive mission planning.
DATA OPTION	Required in ACTIVE mode to determine source of initial mission planning data; INITIAL - Data input through terminal. RESTART - Data initialized through Mission Data File.
INSTRUCTIONS	Entered through KEYBOARD unit; PROCEED - Continue processing. DISPLAY - Initiate output options. EXIT - Terminate processing.

Processing - The flow diagram of the EXEC routine is presented in Figure 13.

Output - The EXEC routine transmits the following data through the COMPOOL:

RUN MODE	Mission Planning Processor operation mode;	
	EVENT	- Computer aided event chart generation only.
	ACTIVE	- Interactive mission planning.

The EXEC routine also provides the following warning displays to the user through the CRT display unit:

INCORRECT RUN MODE.
INCORRECT DATA OPTION.
INCORRECT INSTRUCTION.

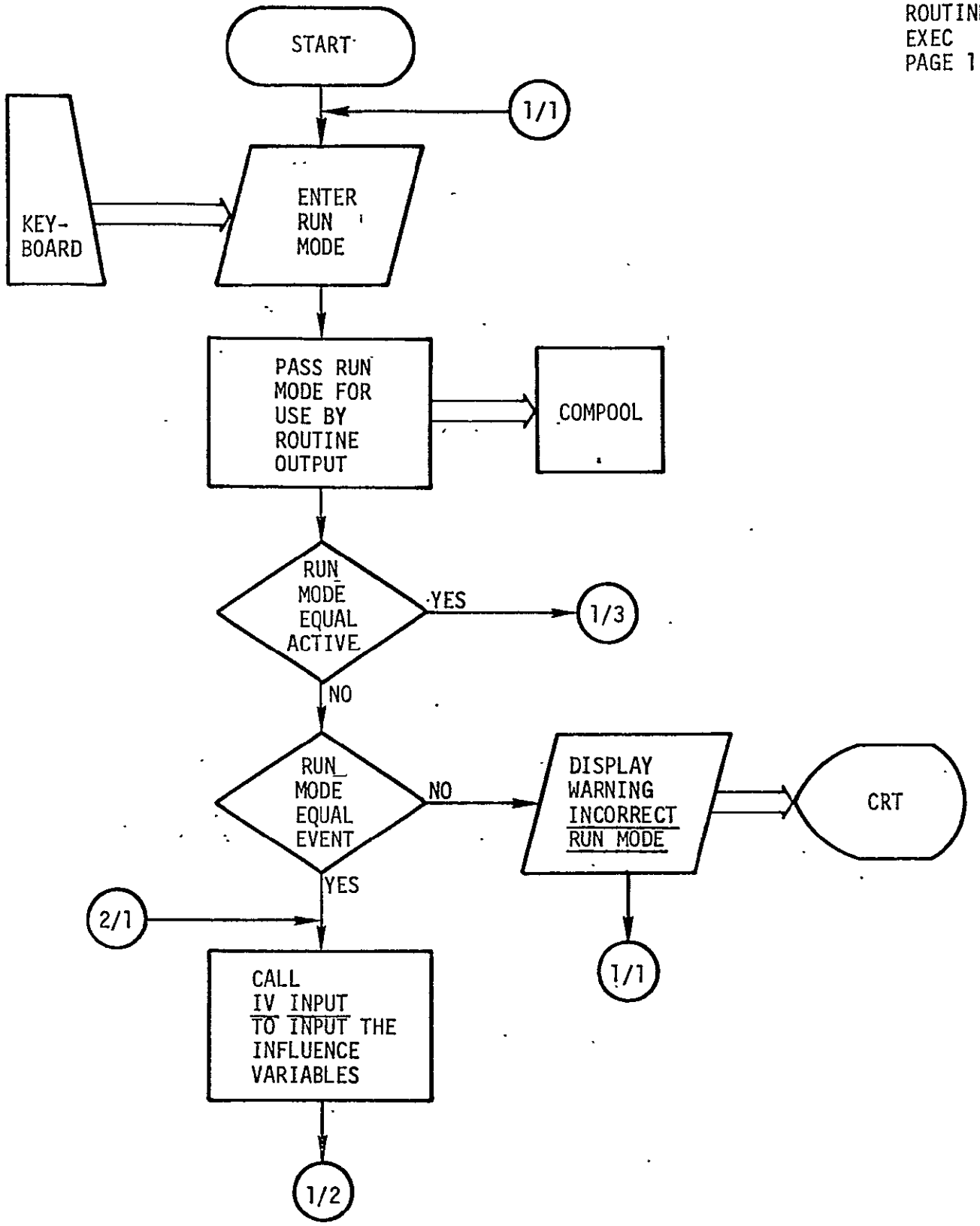


Figure 13. Flow Diagram for the EXEC Routine

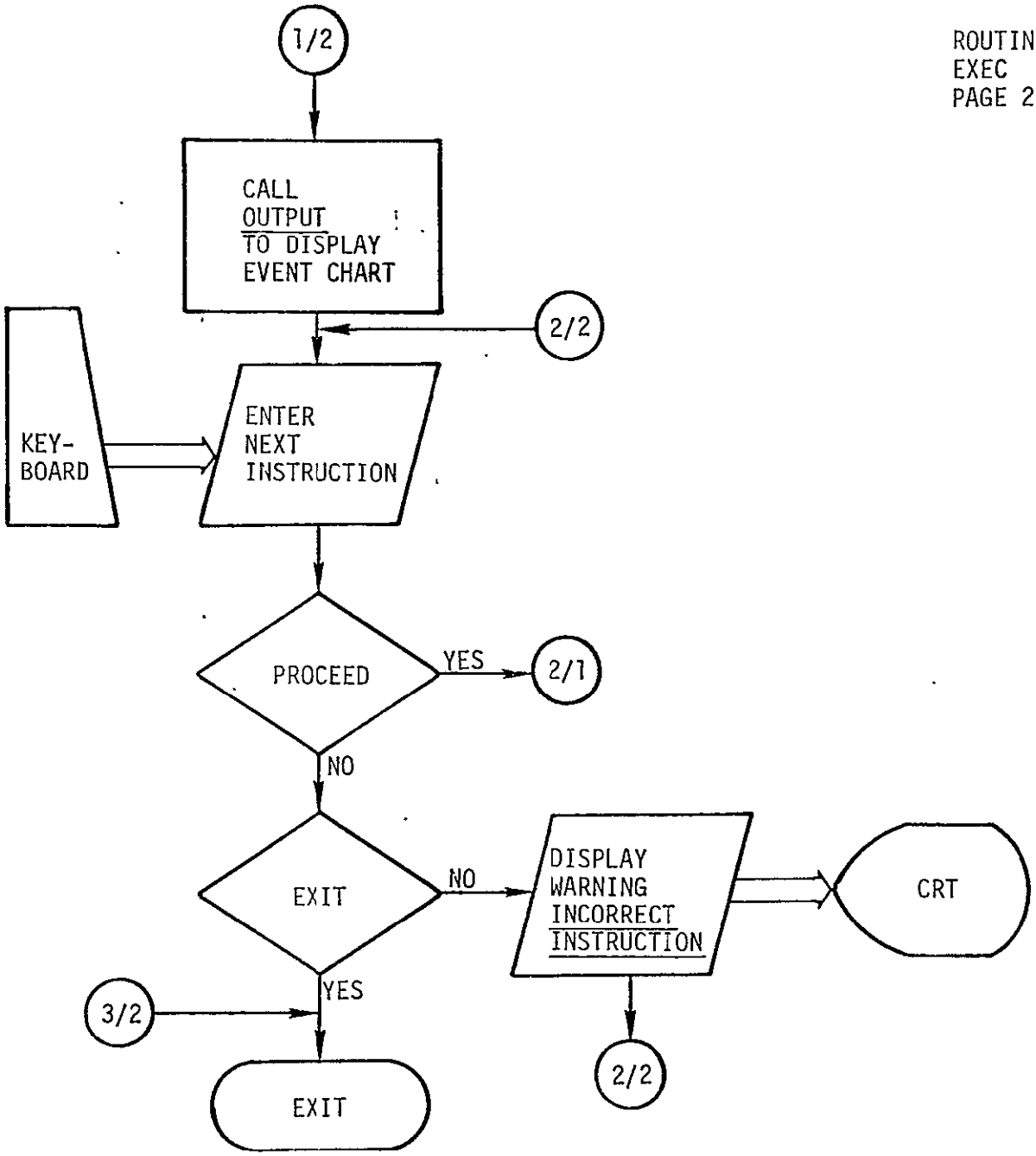


Figure 13. Continued

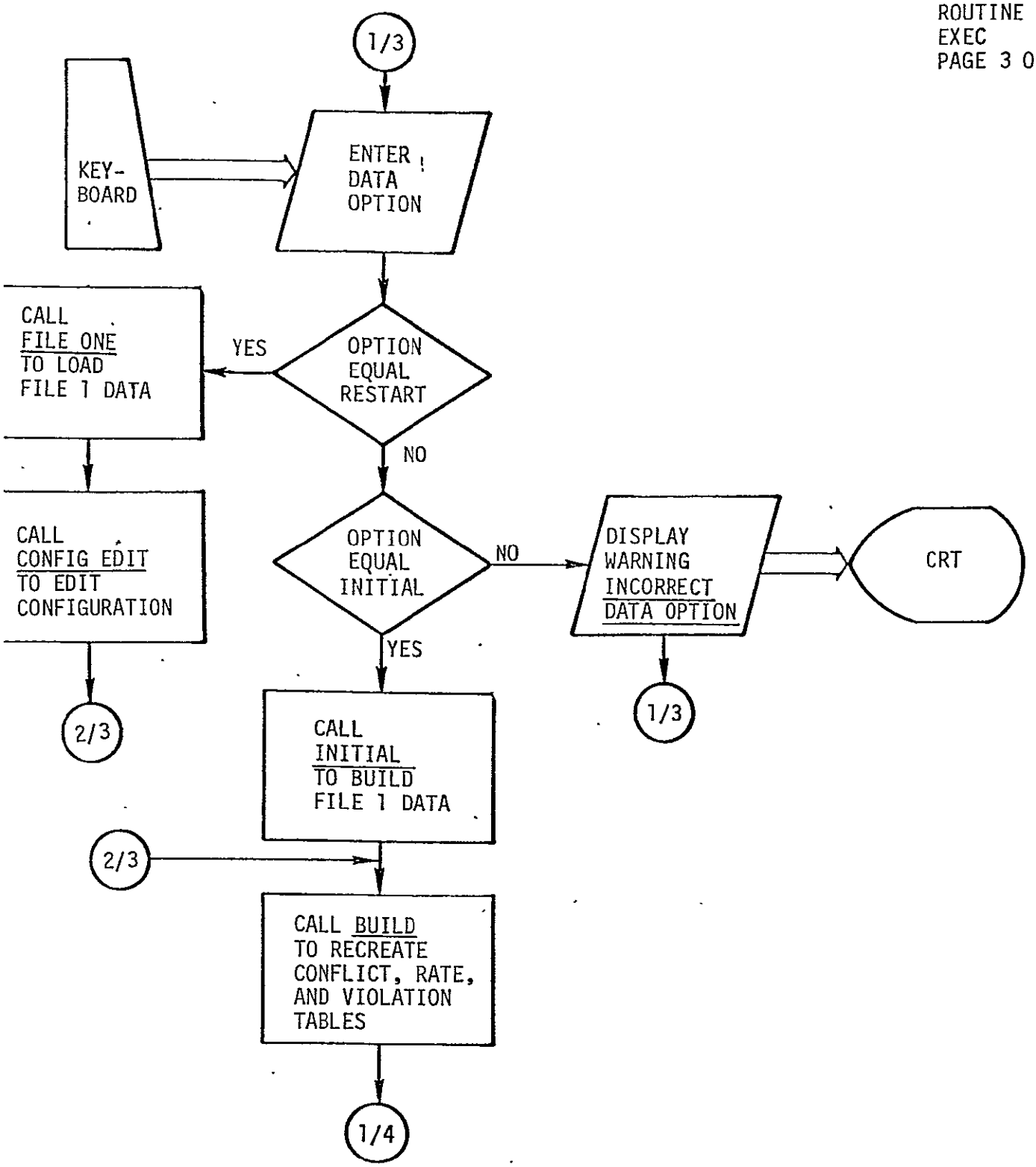


Figure 13. Continued

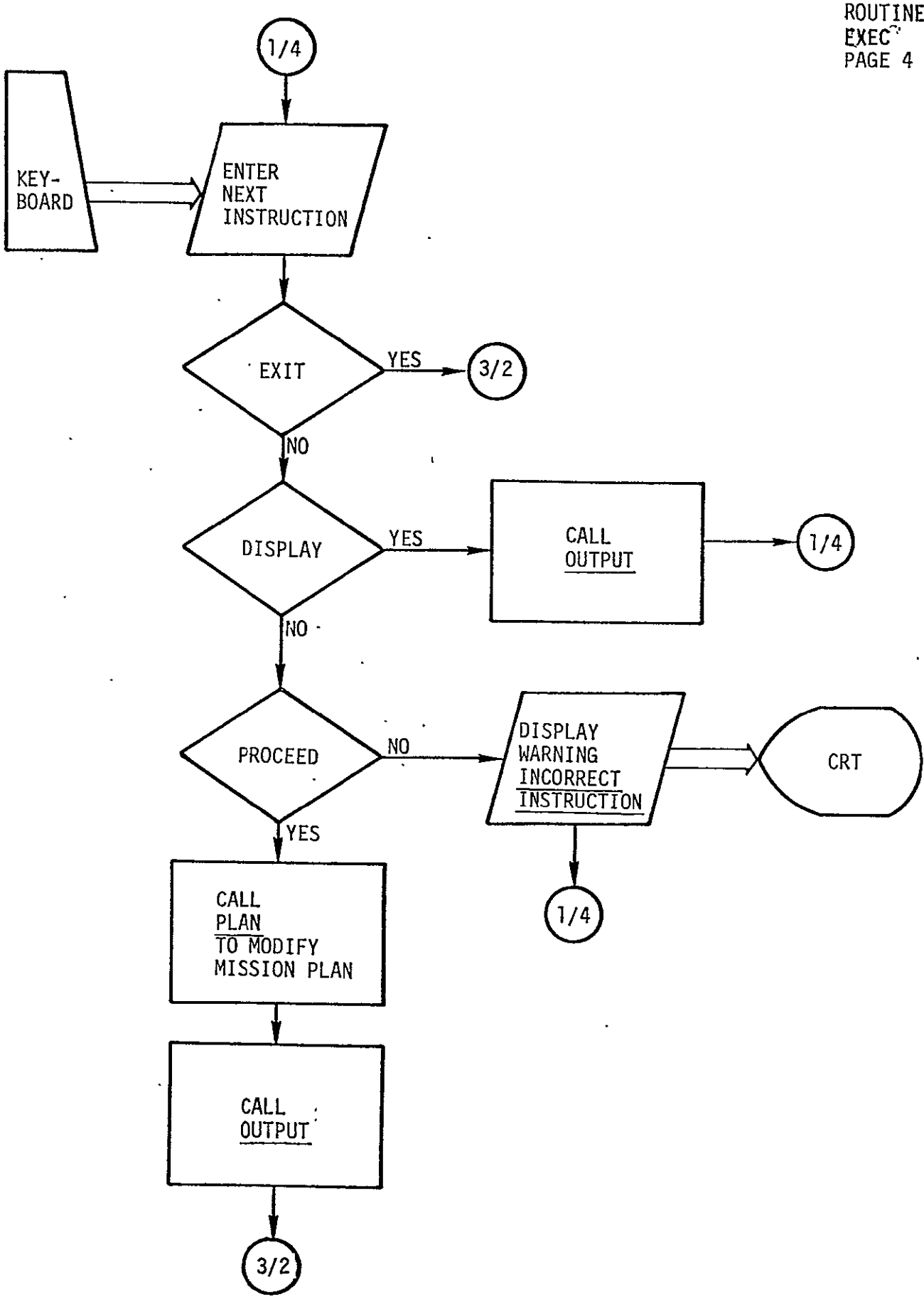


Figure 13. Concluded

5.2 ACTDIS ROUTINE

Description - The ACTDIS routine determines which action identifier is required to schedule or unschedule an event and if the event to be scheduled is cyclic.

Interface

I/O DEVICES - None.
DATA BASE - COMPOOL for both input and output; and Consumables Analysis Data Base for input.
ROUTINES CALLING ACTDIS - ADD and DELETE routines.
ROUTINES CALLED BY ACTDIS - None.

Internal Variables - None.

Input - The ACTDIS routine requires the following input data accessed through the COMPOOL:

Z	The display variable Z = to the identifier of the MENU display being processed (see Table I for ID values).
X	The display variable X = to the identifier of the ACTION display being processed (see Table I for ID values).

The ACTDIS routine requires the following input data from the Consumables Analysis Data Base:

TABLE	The Active Mode Display Cross Reference Table (as defined in Table I).

Processing - The flow diagram of the ACTDIS routine is presented in Figure 14.

Output - The ACTDIS routine transmits the following data through the COMPOOL:

J	The ACTION identifier required by the event to be scheduled or unscheduled (see Table I for the values of J).
CYCLIC	The CYCLIC flag is set if the event to be scheduled is a cyclic event: 0 = NON CYCLIC event 1 = CYCLIC event

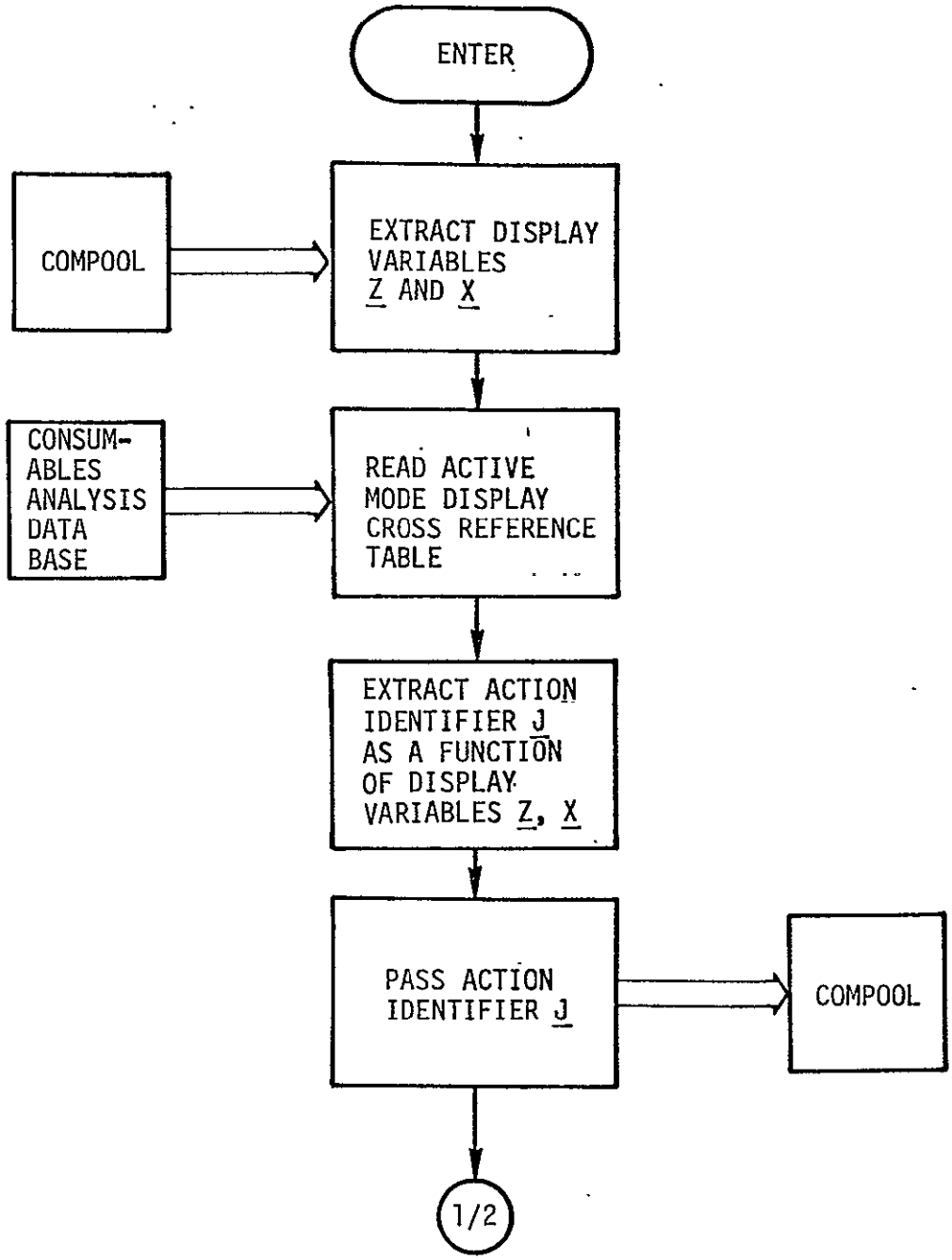


Figure 14. Flow Diagram for the ACTDIS Routine

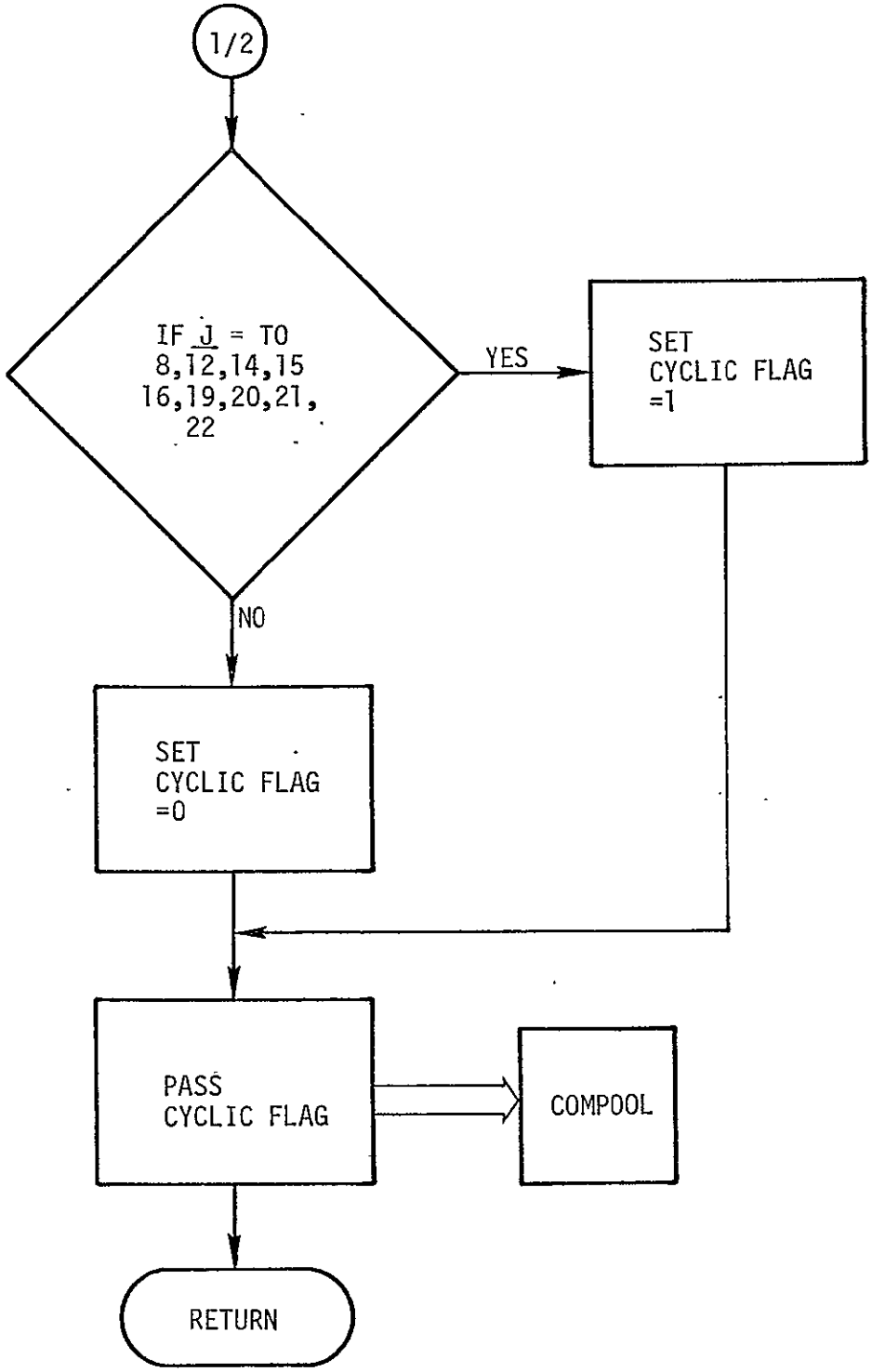


Figure 14. Concluded

5.3 ACTION ROUTINE

Description - The ACTION routine schedules or unschedules an event by updating the File 1 entry data array and calling other control and support routines to update the conflict and consumable rate tables.

Interface

I/O DEVICE - None.
DATA BASE - COMPOOL for both input and output; and the Consumables Analysis Data Base for input.
ROUTINES CALLING ACTION - ADD and DELETE routines.
ROUTINES CALLED BY ACTION - SPECIAL, CONFLICT, and RATE routines.

Internal Variables - None.

Input - The ACTION routine requires the following input data accessed through the COMPOOL:

ACTION MODE	Mode flag for the ACTION routine to schedule or unschedule an event: ADD = schedule an event DELETE = unschedule an event.
J	The ACTIQN identifier required by the event to be scheduled or unscheduled (see Table I for the values of J).
K	The activity number for the event to be scheduled or unscheduled.
REF START	The reference start time for the event to be scheduled or unscheduled.
REF STOP	The reference stop time for the event to be scheduled or unscheduled.

The ACTION routine requires the following input data from the Consumables Analysis Data Base:

DT(J,I)	J=1,23 I=1,2	The delta time array for the action identified by J. I=1 ΔT for preparation period 2 ΔT for post activity period.

Processing - The flow diagram of the ACTION routine is presented in Figure 15.

Output - The ACTION routine transmits the following data through the COMPOOL:

AT(K,I) I=1,5 Entry data array for activity K

- I=1 prep start time
- 2 reference start time
- 3 reference stop time
- 4 post end time
- 5 special parameter, a function of ACTION Identifier J:
  - J=1,2,4,6, or 7; AT(K,5) = ΔV
  - J=9,10,20,21, or 22; AT(K,5) = Number of crew.

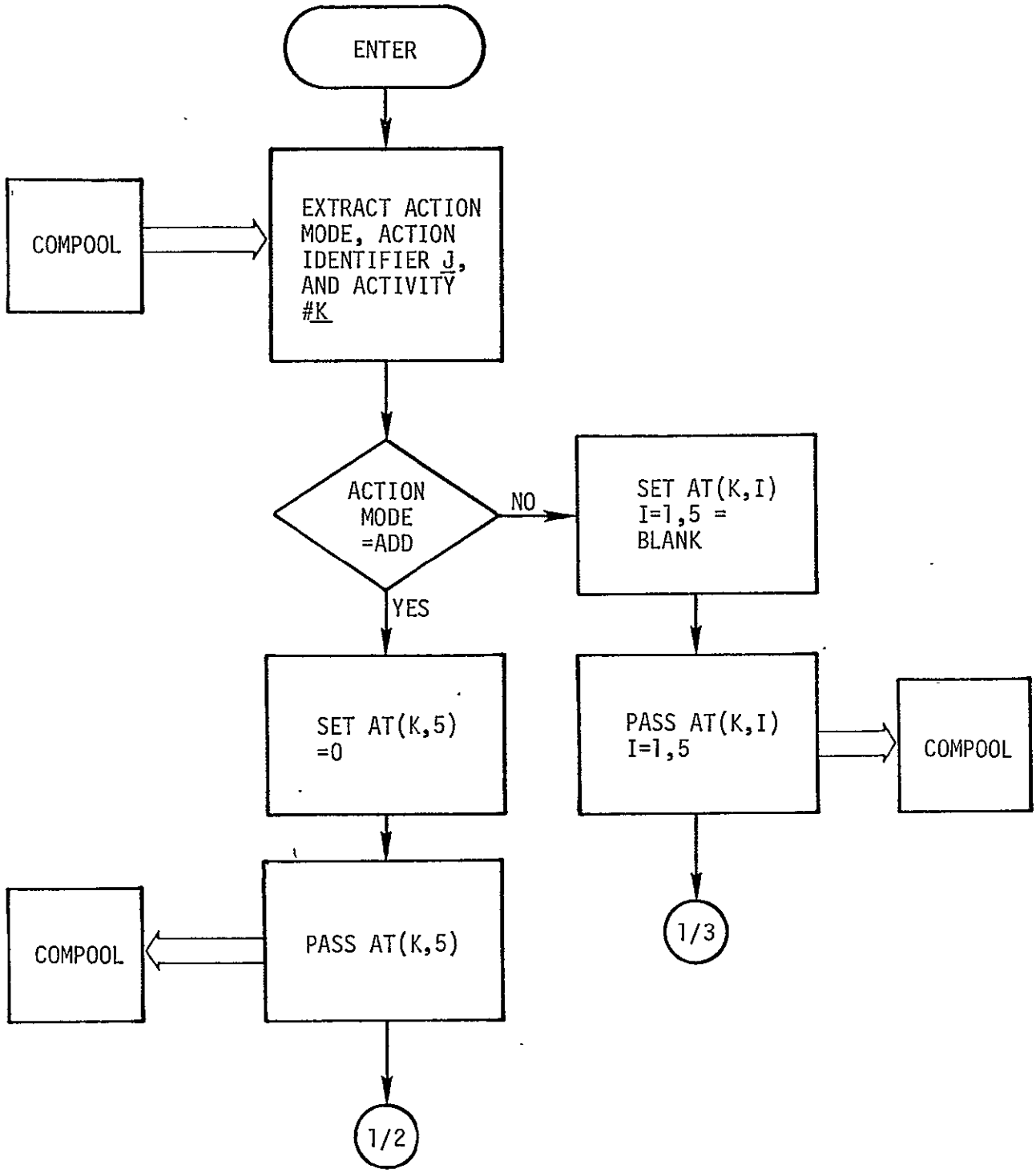


Figure 15. Flow Diagram for the ACTION Routine

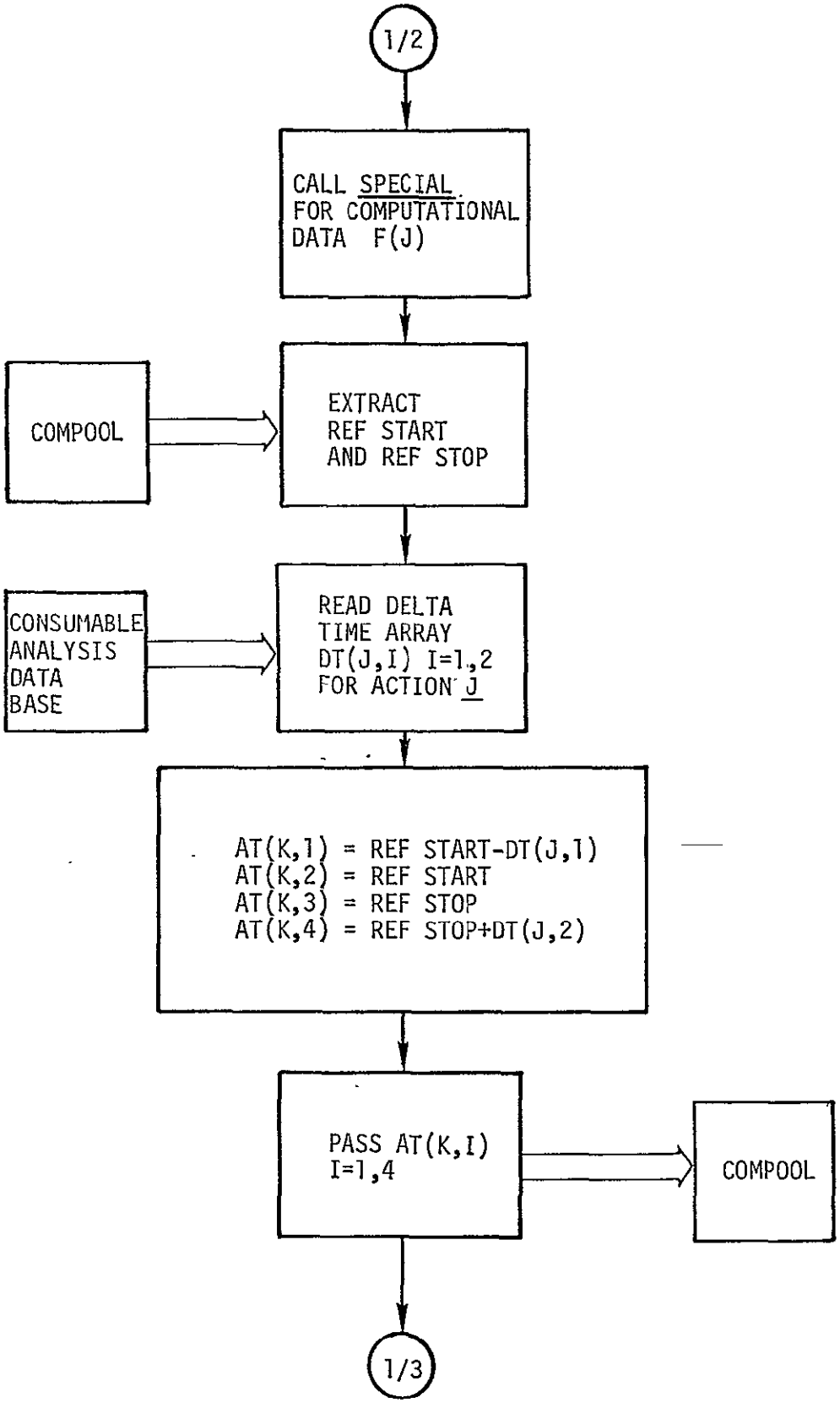


Figure 15. Continued

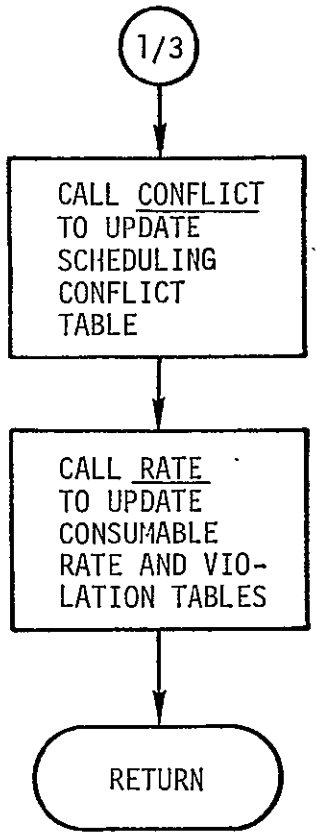


Figure 15. Concluded

5.4 ADD ROUTINE

Description - The ADD routine schedules an event by directly and indirectly updating the File 1 data set and indirectly updating the subsystem rate and conflict tables affected by the new event.

Interface

I/O DEVICES - None.
DATA BASE - COMPOOL for both input and output.
ROUTINES CALLING ADD - PLAN routine.
ROUTINES CALLED BY ADD - POOL, ACTDIS, ACTION, and SEQUENCE routines.

Internal Variables

CYCLIC STOP	If the event to be scheduled is a cyclic event, the CYCLIC STOP time is set.

Input - The ADD routine requires the following input data accessed through the COMPOOL:

K	Activity number for event to be scheduled.
J	Action identifier for the event to be scheduled.
CYCLIC	The CYCLIC flag is set if the event to be scheduled is a cyclic event: 0 = NON CYCLIC event 1 = CYCLIC event.

If CYCLIC = 1, the following additional input data accessed through the COMPOOL is required by the ADD routine:

REF START	The reference start time for the event cycle.
REF STOP	The reference stop time for the event cycle.
Duty	The duration of each event in the cycle.
Period	The duration of each period in the cycle.

Processing - The flow diagram of the ADD routine is presented in Figure 16.

Output - The ADD routine transmits the following data through the COMPOOL:

NN(K)=J	K=1,N	Activity number K is an ACTION J.
IN(J)	J=1,23	Number of ACTION J items scheduled.
NNN(I,J)=K	I=IN(J) J=1,23	Activity number K is the Ith item of ACTION J scheduled.
ACTION MODE=ADD		Mode flag for the ACTION routine to schedule an event.

If CYCLIC=1, the following additional data is transmitted through the COMPOOL:

REF START	The reference start time for each event in the cycle.
REF STOP	The reference stop time for each event in the cycle.

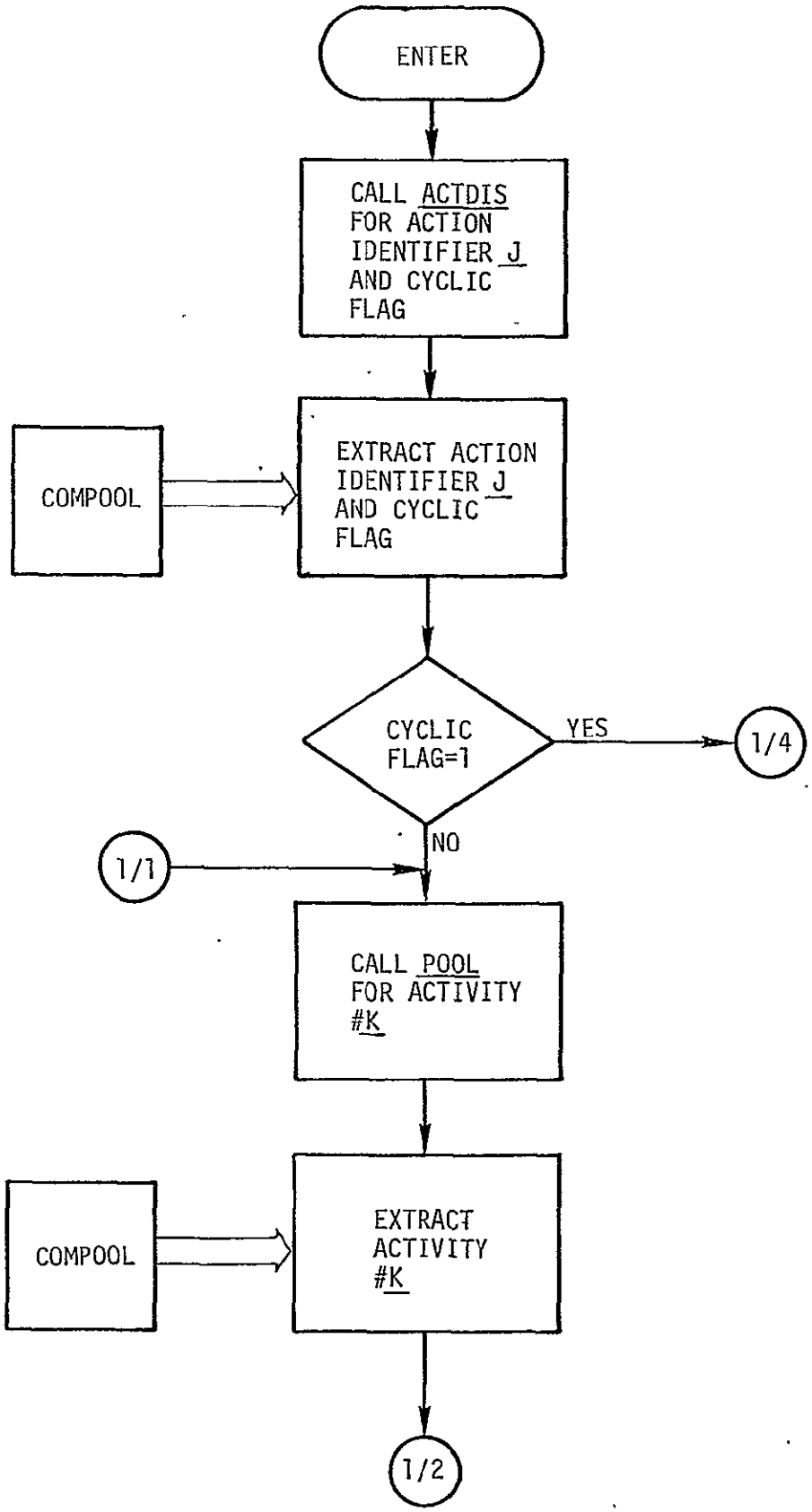


Figure 16. Flow Diagram for the ADD Routine

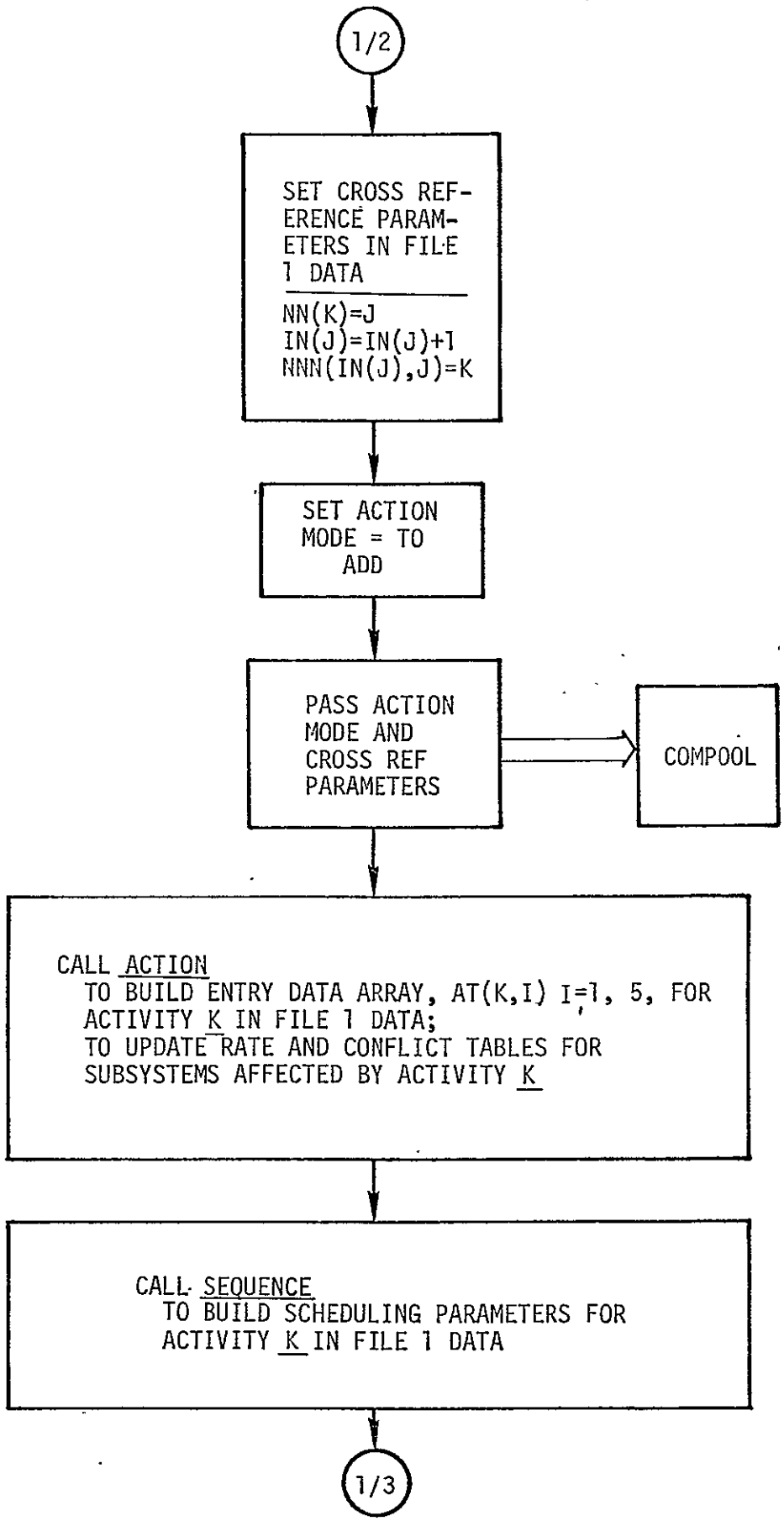


Figure 16. Continued

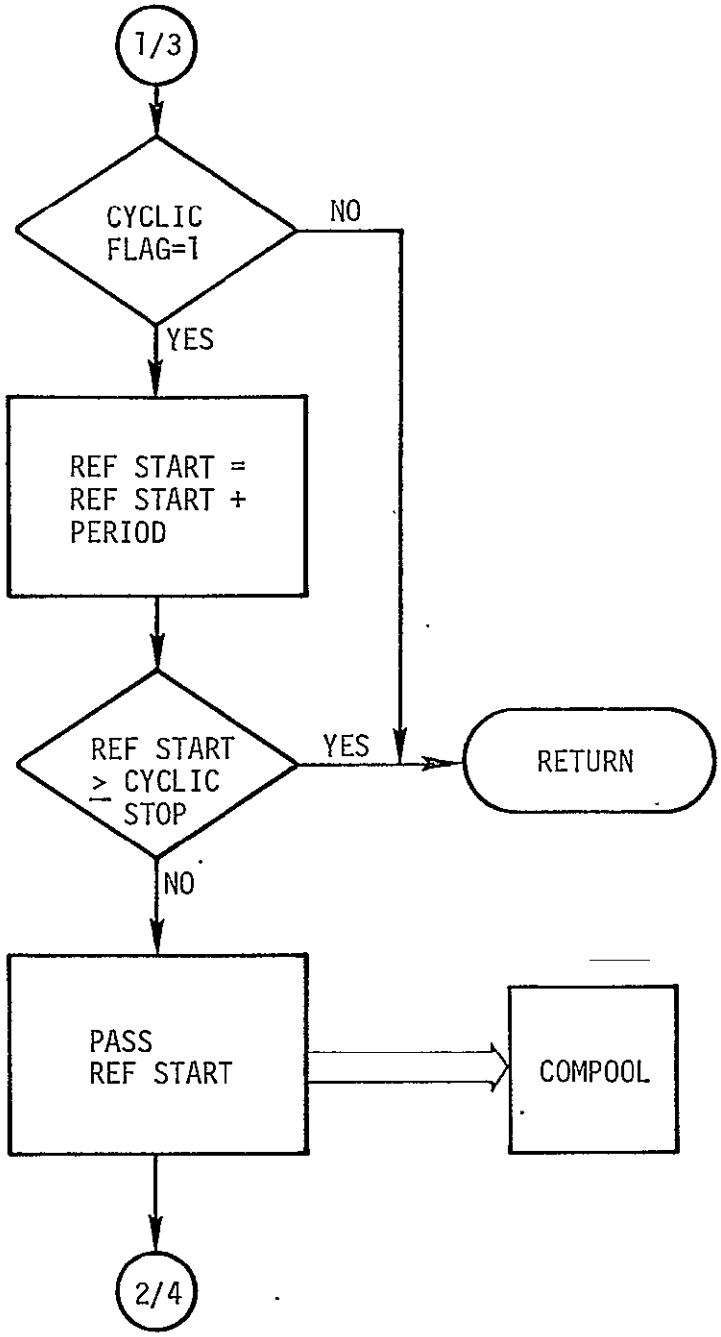


Figure 16. Continued

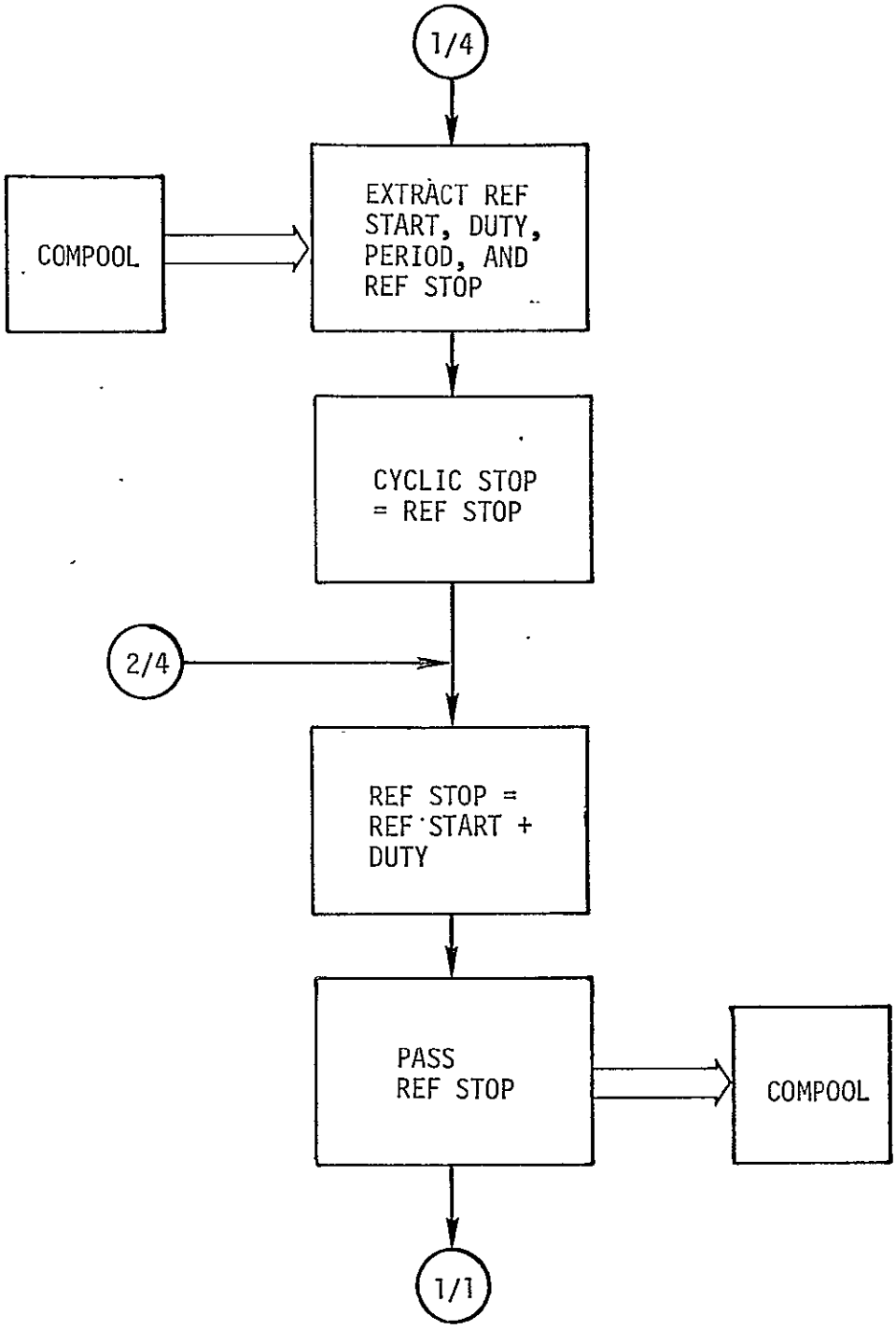


Figure 16. Concluded

5.5 BUILD ROUTINE

Description - The BUILD routine recreates the scheduling conflict table, the consumable rate tables, and the rate violation table from the stored File 1 data set.

Interface

I/O DEVICES - None.
DATA BASE - COMPOOL for both input and output.
ROUTINES CALLING BUILD - EXEC routine.
ROUTINES CALLED BY BUILD - SPECIAL, RATE, and CONFLICT routines.

Internal Variables

JMAX	The maximum number of ACTION Js to be cycled.
JX	The maximum number of events scheduled for a particular ACTION J.
I	The index for a particular event.

Input - The BUILD routine requires the following input data accessed through the COMPOOL:

J		The ACTION identifier required by the event or common block to be scheduled (see Table I for the values of J).
IN(J)	J=1,23	Number of ACTION J items scheduled.
NNN(I,J)=K	I=1,IN(J) J=1,23	Event-Activity cross reference.
N		The entry counter.
AT(K,I)	K=1,N I=1,5	Entry data array for activity K I=1 prep start time 2 reference start time 3 reference stop time 4 post end time 5 special parameter, a function of ACTION Identifier J: J=1,2,4,6, or 7; AT(K,5)=ΔV J=9,10,20,21, or 22; AT(K,5)=Number of crew.

Processing - The flow diagram of the BUILD routine is presented in Figure 17.

Output - The BUILD routine transmits the following data through the COMPOOL:

ACTION MODE=ADD	Mode flag for the ACTION routine to schedule an event.
J	The ACTION identifier required by the event to be scheduled (see Table I for the values of J).
K	Activity number for the event to be scheduled.
REF START	The reference start time for the event to be scheduled.
REF STOP	The reference stop time for the event to be scheduled.

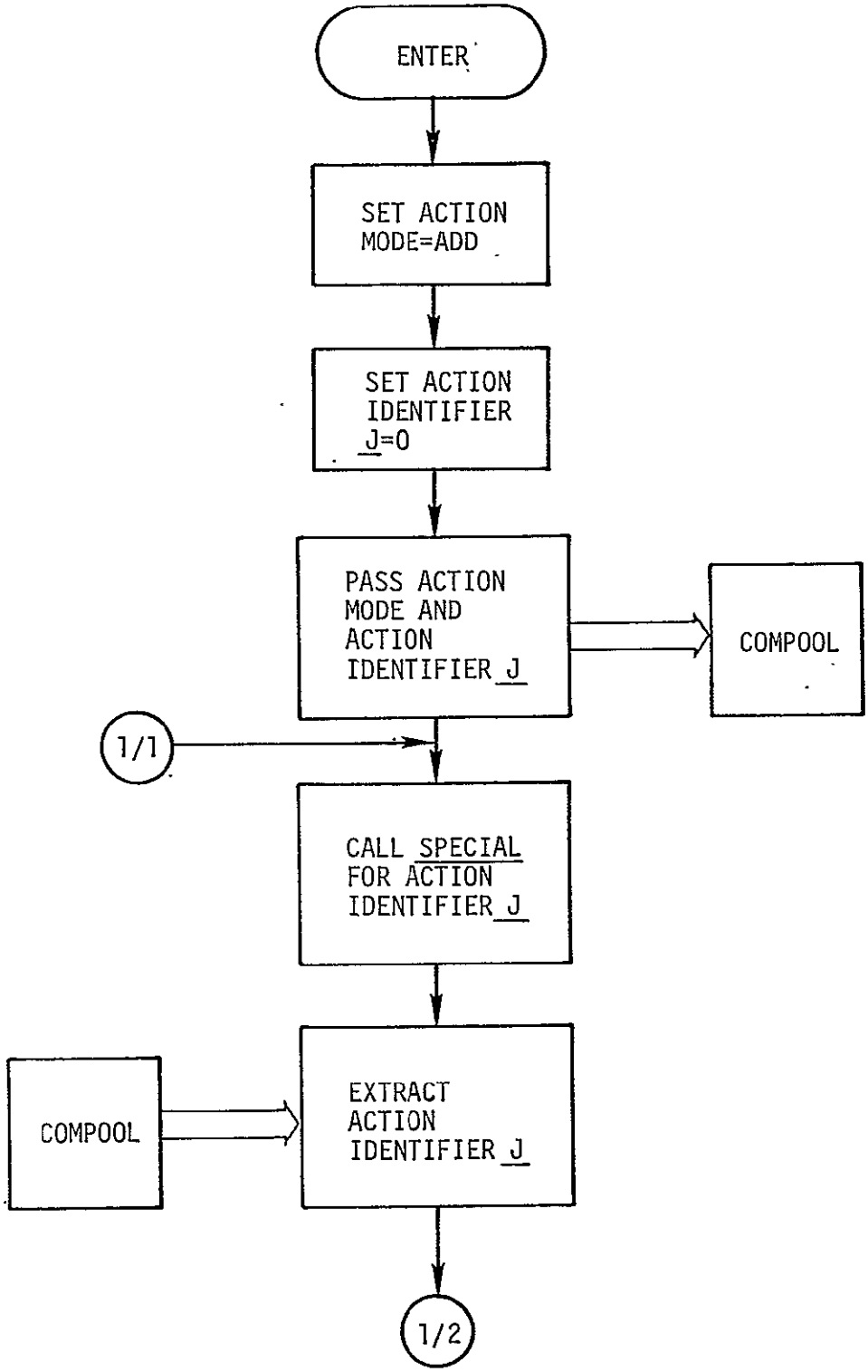


Figure 17. Flow Diagram for the BUILD Routine

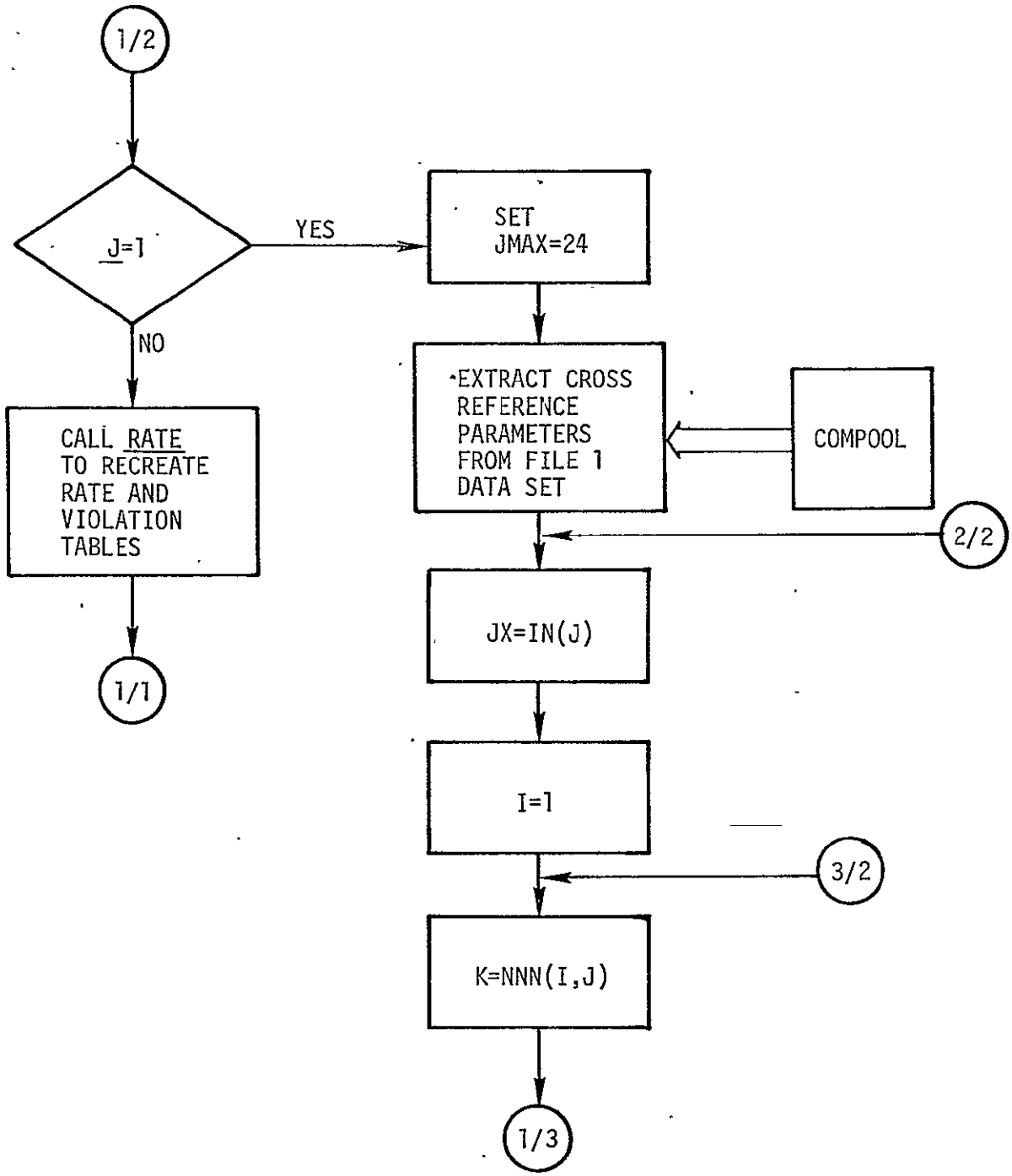


Figure 17. Continued

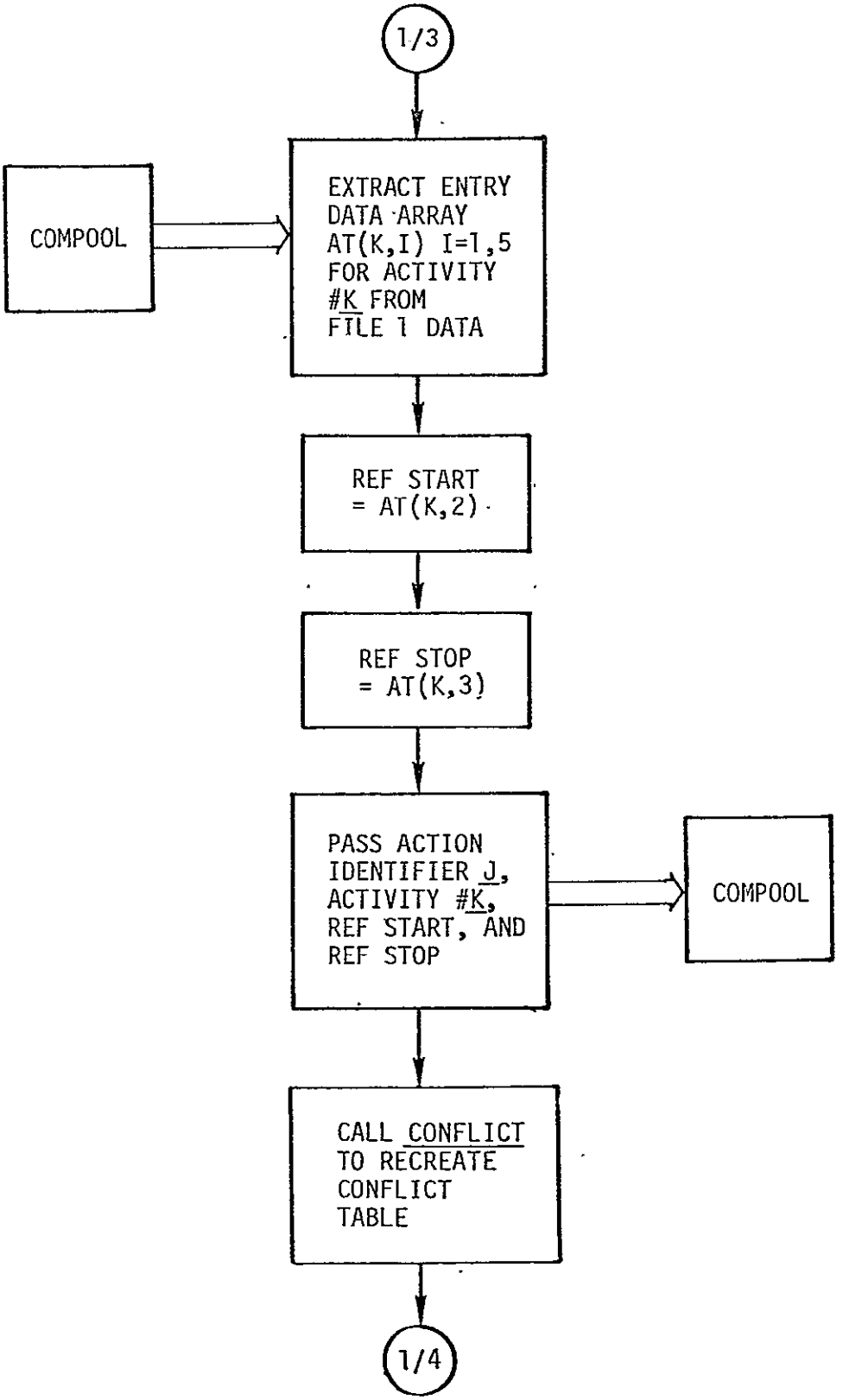


Figure 17. Continued

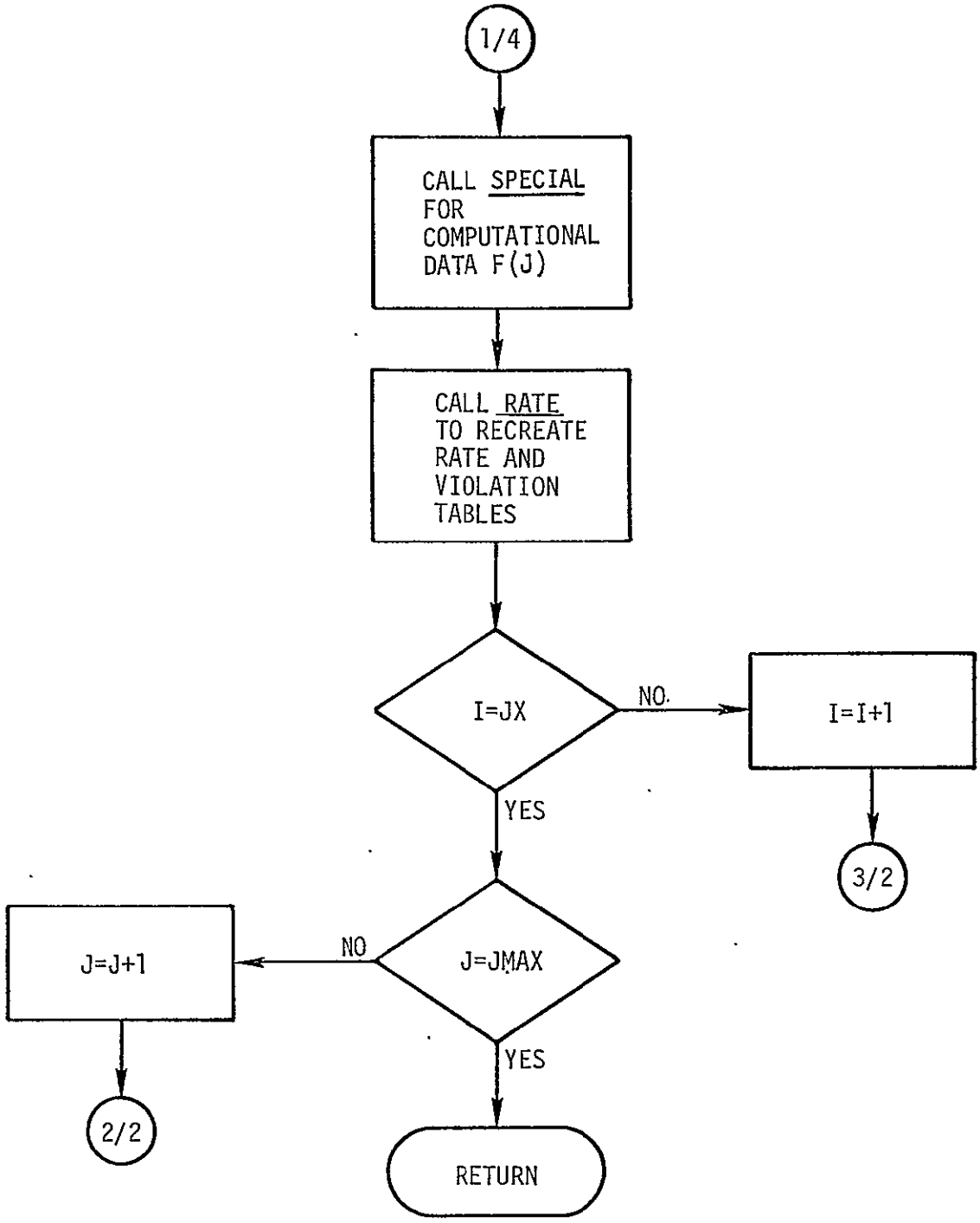


Figure 17. Concluded

5.6 CONFIG EDIT ROUTINE

Description - The CONFIG EDIT routine allows the user to modify the mission configuration data stored in File 1 of the Flight Data Files.

Interface

I/O DEVICES - Terminal KEYBOARD and CRT units.
DATA BASE - COMPOOL for both input and output.
ROUTINES CALLING CONFIG EDIT - EXEC routine.
ROUTINES CALLED BY CONFIG EDIT - None.

Internal Variables - None.

Input - The CONFIG EDIT routine can accept the following input data entered through the terminal KEYBOARD unit or through COMPOOL:

CONFG(I)	I=1,11	Mission configuration data (see Table III for parameter definitions).

Processing - The CONFIG EDIT routine is an information management type routine used to manipulate the mission dependent data required as input to the Mission Planning Processor. The data are input using the Configuration Block Display illustrated in Figure 2. No flow diagram is necessary.

Output - The CONFIG EDIT routine transmits the following data through the COMPOOL:

CONFG(I)	I=1,11	Mission configuration data (see Table III for parameter definition).

5.7 CONFLICT ROUTINE

Description - The CONFLICT routine supervises the Scheduling Conflict Table. A proposed Scheduling Conflict Table skeleton is presented in Figure 18. If the ACTION MODE=DELETE, the CONFLICT routine will remove any conflicts noted in the Scheduling Conflict Table attributed to the event being unscheduled. If the ACTION MODE=ADD, the CONFLICT routine will determine if the event being scheduled conflicts with any previously scheduled events. The CONFLICT routine searches for events scheduled during the same time period as the event to be scheduled. If found, the CONFLICT routine determines event compatibility by reading the Master Compatibility Matrix and Compatibility Arrays for the event being scheduled. Figures 19 and 20 present proposed formats for the Master Compatibility Matrix and Compatibility Array, respectively. Compatibility Arrays should be provided for each of the three phases (preparation, activity, and post activity) of each event (J=1,23). If conflicts are created, the Scheduling Conflict Table will be updated to note the conflict and an interactive warning will be provided to the user on the terminal CRT.

Interface

I/O DEVICES - The terminal CRT unit for output only.
DATA BASE - COMPOOL for both input and output; and the Consumables Analysis Data Base for input.
ROUTINES CALLING CONFLICT - ACTION and BUILD routines.
ROUTINES CALLED BY CONFLICT - None.

Internal Variables

TIN	Time to initiate search.
TOUT	Time to complete search.
TFLAG	Flag indicating phase of activity being scheduled: TFLAG = 1 preparation period 2 activity period 3 post activity period.
SFLAG	Flag indicating phase of activity previously scheduled: SFLAG = 1 preparation period 2 activity period 3 post activity period 12 prep and/or activity periods 13 activity and/or post periods.

KSCHED	The activity number of the previously scheduled event.
JSCHED	The ACTION identifier of the previously scheduled event.

Input - The CONFLICT routine requires the following input data accessed through the COMPOOL:

ACTION MODE		Mode flag for the ACTION routine to schedule or unschedule an event: ADD = schedule an event DELETE = unschedule an event.
J		The ACTION identifier required by the event to be scheduled or unscheduled (see Table I for the values of J).
K		The activity number for the event to be scheduled or unscheduled.
AT(K,I)	I=1,5	Entry data array for activity K I=1 prep start time 2 reference start time 3 reference stop time 4 post end time 5 special parameter, a function of ACTION Identifier J: J=1,2,4,6, or 7; AT(K,5)=ΔV J=9,10,20,21 or 22; AT(K,5)=Number of crew.
TABLE		The Scheduling Conflict Table (as defined in Figure 18).

If the ACTION MODE=ADD, the CONFLICT routine requires the following additional input data accessed through the COMPOOL:

NOI		Number of entries in sequence array.
IT(I)	I=1,NOI	Sequence array of activities.
TIM(K,L)	I=1,NOI L=1,2	Start and end times of activities: L=1 minimum start time of activity IT(I) 2 maximum end time of activity IT(I).
NN(K)=J	K=KSCHED	The Activity-Action cross reference for the previously scheduled event.
AT(K,I)	K=KSCHED I=1,5	Entry data array for the previously scheduled activity (see AT(K,I) for values).

If the ACTION MODE=ADD, the CONFLICT routine requires the following additional input data from the Consumables Analysis Data Base:

MATRIX	The Master Compatibility Matrix (as defined in Figure 19).

JCOMP(M1,M2,M3) M1=J M2=TFLAG M3=JSCHED	The Compatibility Array for the event to be scheduled (as defined in Figure 20).

Processing - The flow diagram of the CONFLICT routine is presented in Figure 21.

Output - If a scheduling conflict is detected, the following will be displayed to the user on the terminal CRT unit:

WARNING	Scheduling conflict created. See Scheduling Conflict Table for details.

If a scheduling conflict is detected the CONFLICT routine will transmit the following through the COMPOOL:

ENTRY	Entry to Scheduling Conflict Table (as defined in Figure 18).

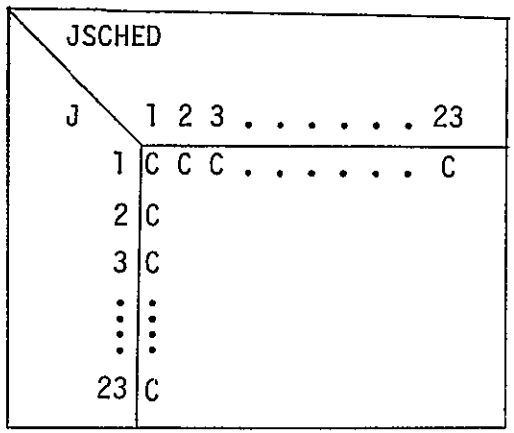
SCHEDULING CONFLICT TABLE

MISSION ID: RUN MODE:

ITEM	TIME OF CONFLICT	CONFLICTING EVENTS	MIN START OF EVENT	MAX END OF EVENT
1				
2				
•				
•				
•				
n				

Figure 18. The Scheduling Conflict Table Skeleton

MASTER COMPATIBILITY MATRIX



JSCHED — The ACTION Identifier for the scheduled event (see Table I for the definition of J=JSCHED). Range = 1 to 23.

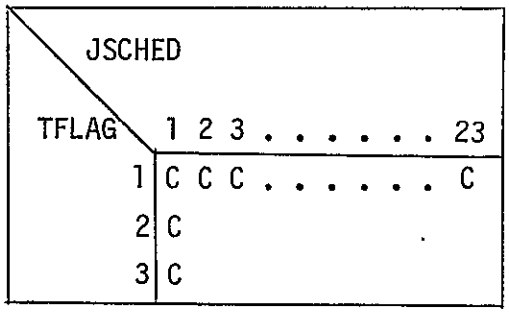
J — The ACTION Identifier for the event being scheduled (see Table I for the definition of J). Range = 1 to 23.

COMPATIBILITY CONSTANT	VALUE	COMPATIBILITY
C	-1	J is completely compatible with JSCHED.
	0	J is incompatible with JSCHED.
	1	J is partially compatible with JSCHED.

Figure 19 The Master Compatibility Matrix

COMPATIBILITY ARRAY

JCOMP(J, TFLAG, JSCHED)



J — The ACTION Identifier for the event being scheduled (see Table I for the definition of J). Range = 1 to 23.

TFLAG — Indicator of event period:
TFLAG = 1 preparation period
2 activity period
3 post activity period

JSCHED — The ACTION Identifier for the scheduled event (see Table I for the definition of J). Range = 1 to 23.

COMPATIBILITY CONSTANT	VALUE	COMPATIBILITY
C	1	J compatible with JSCHED prep period only
	2	J compatible with JSCHED activity period only
	3	J compatible with JSCHED post period only
	12	J compatible with JSCHED prep and/or activity periods
	13	J compatible with JSCHED activity and/or post periods

Figure 20. The Compatibility Array

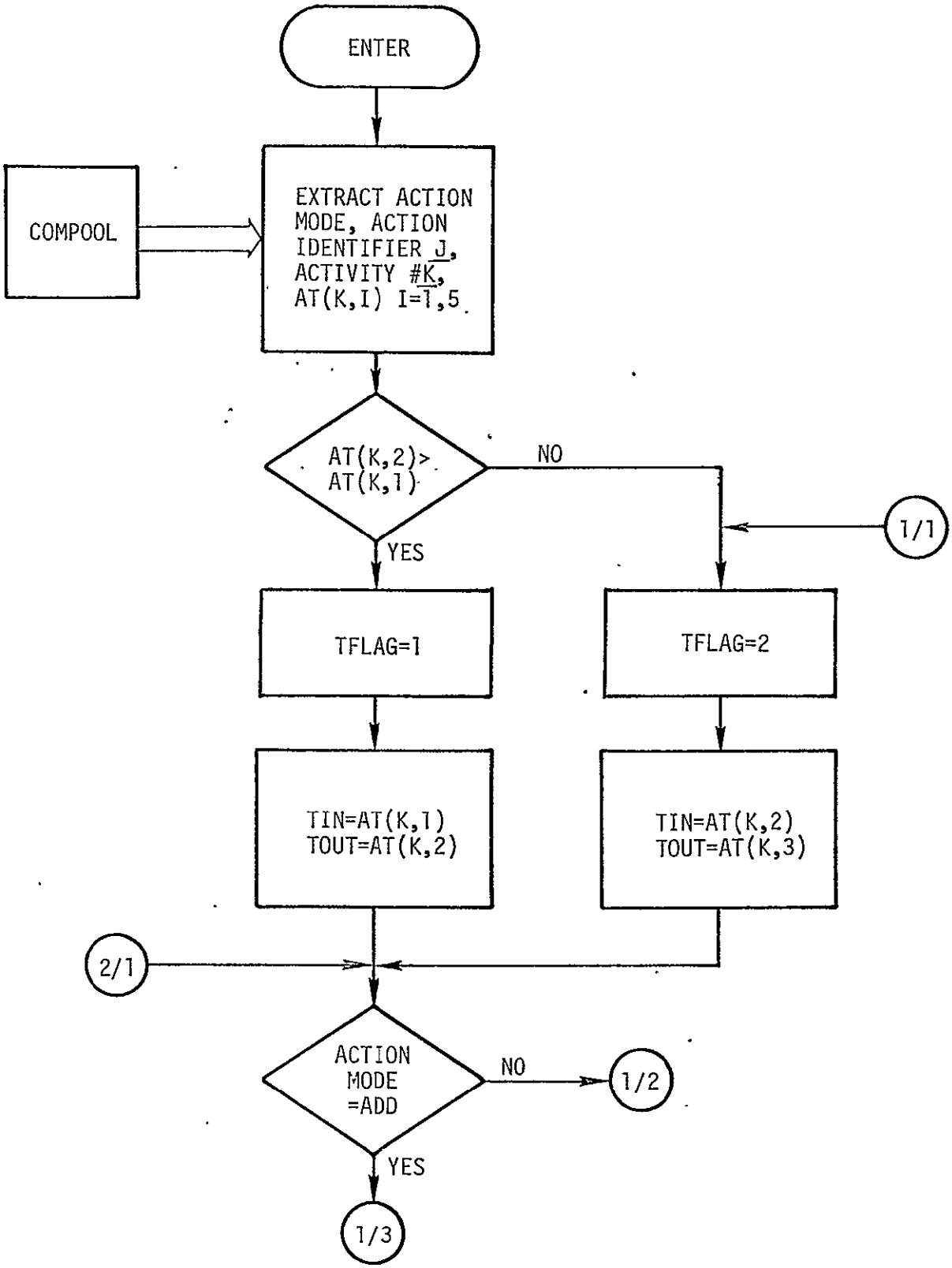


Figure 21. Flow Diagram for the CONFLICT Routine

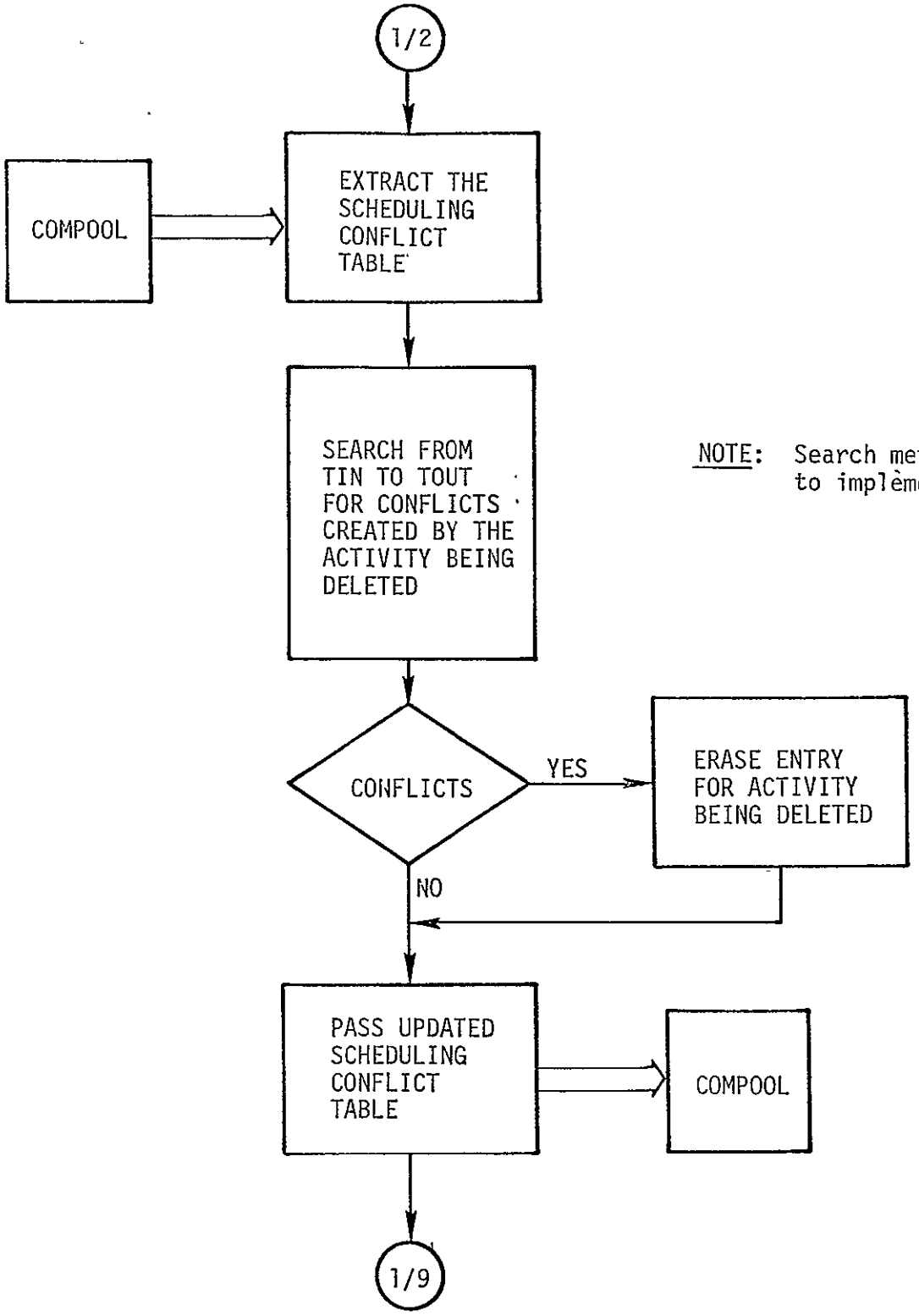


Figure 21. Continued

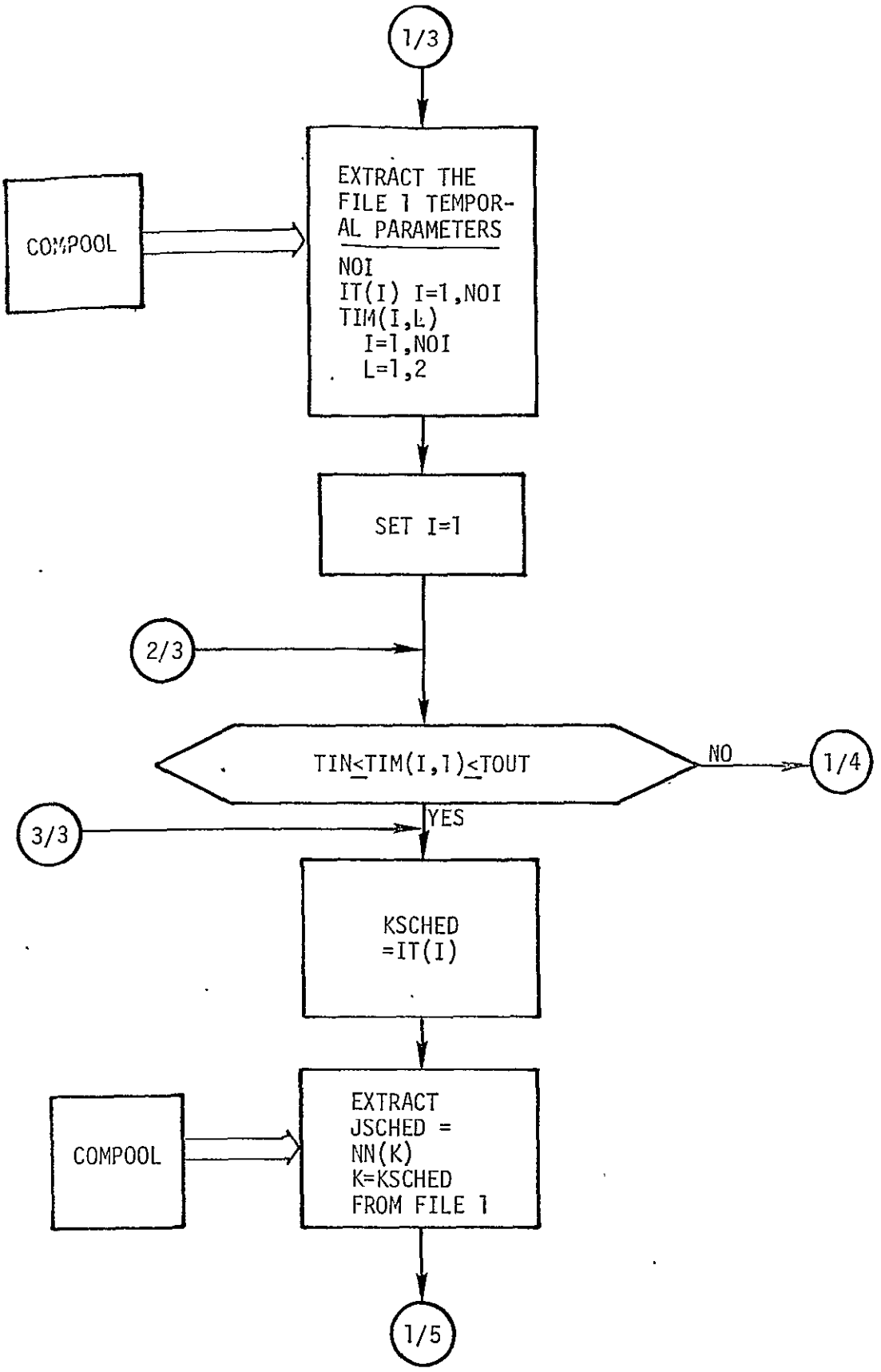


Figure 21. Continued

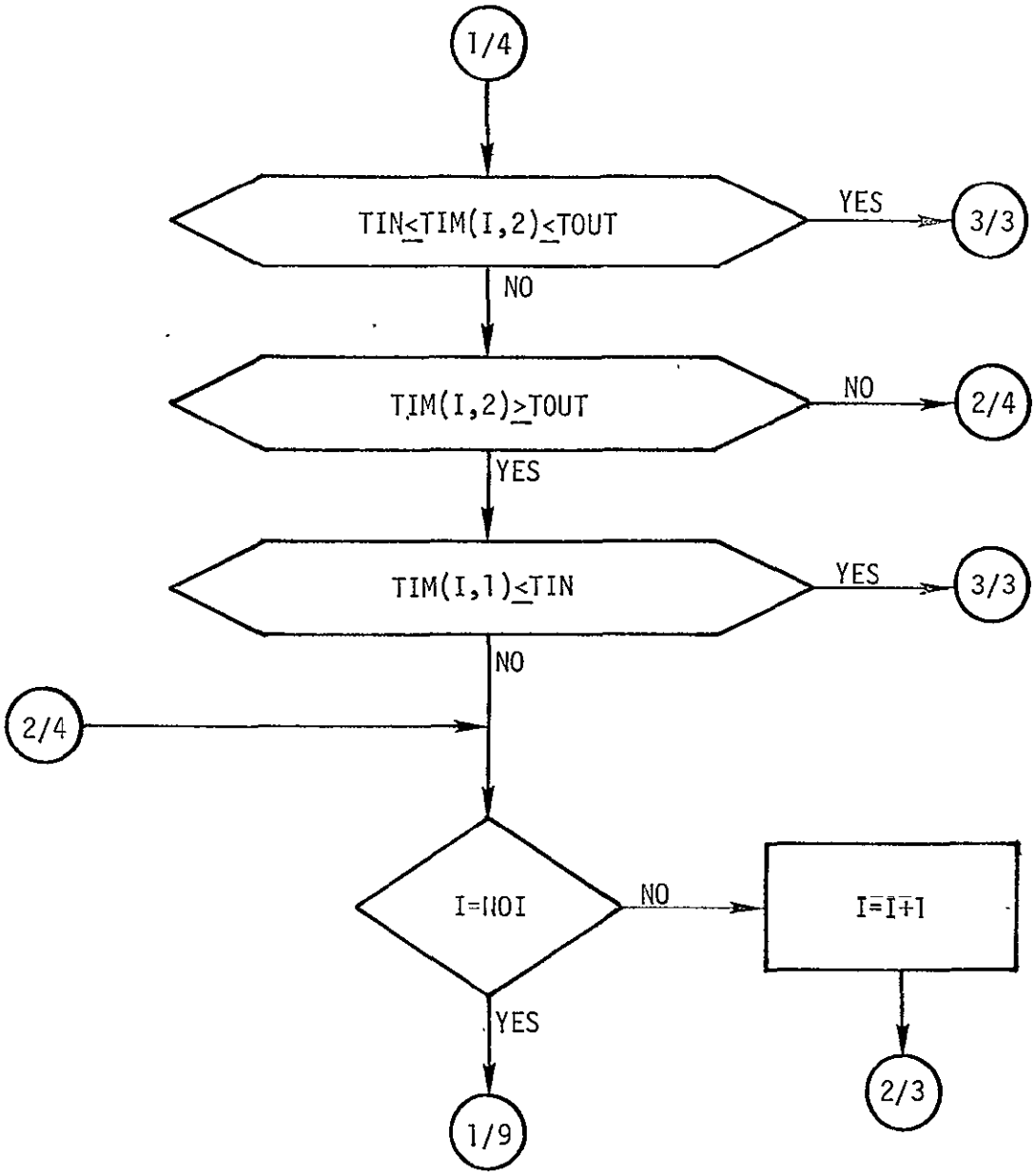


Figure 21. Continued

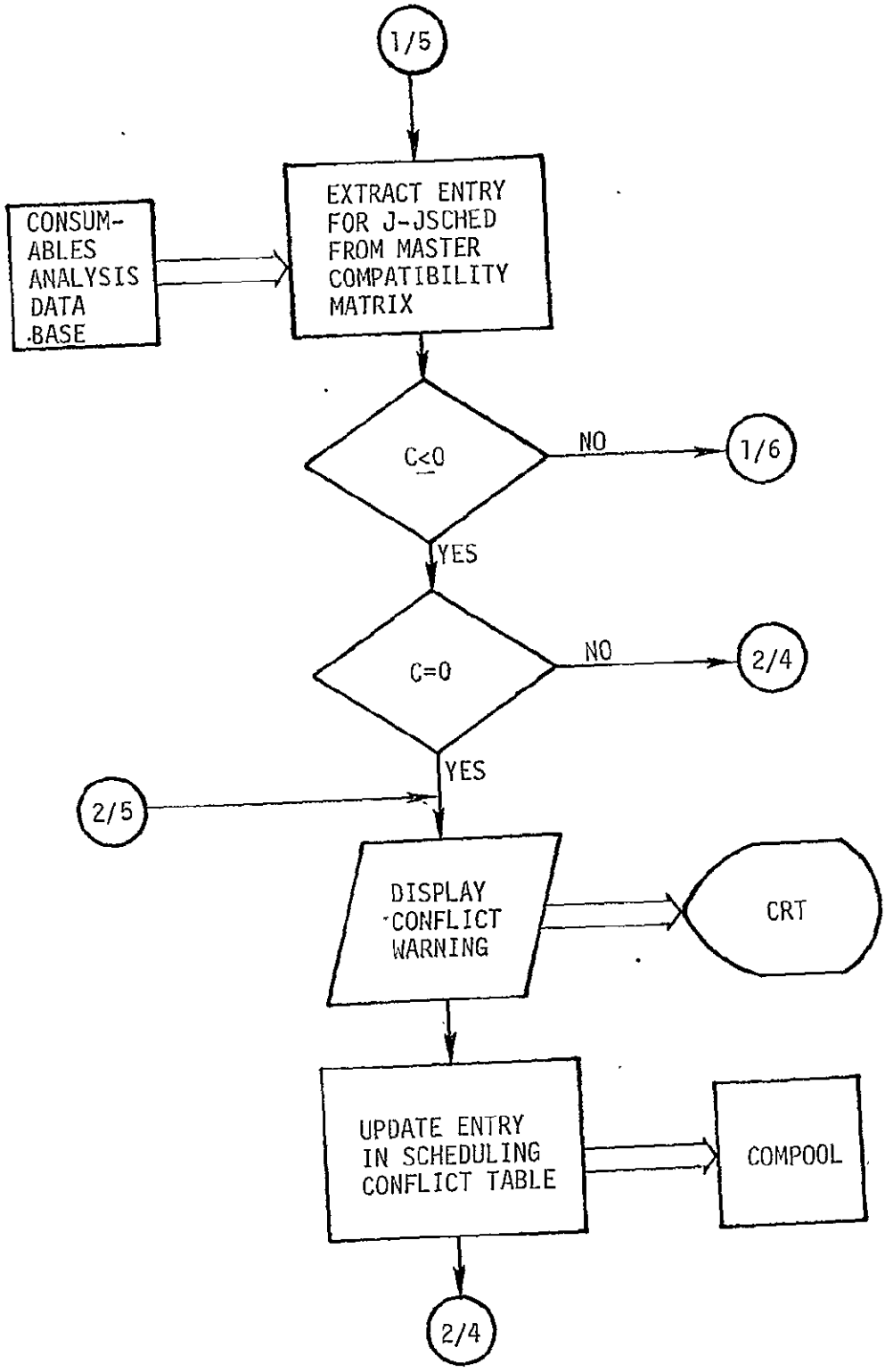


Figure 21. Continued

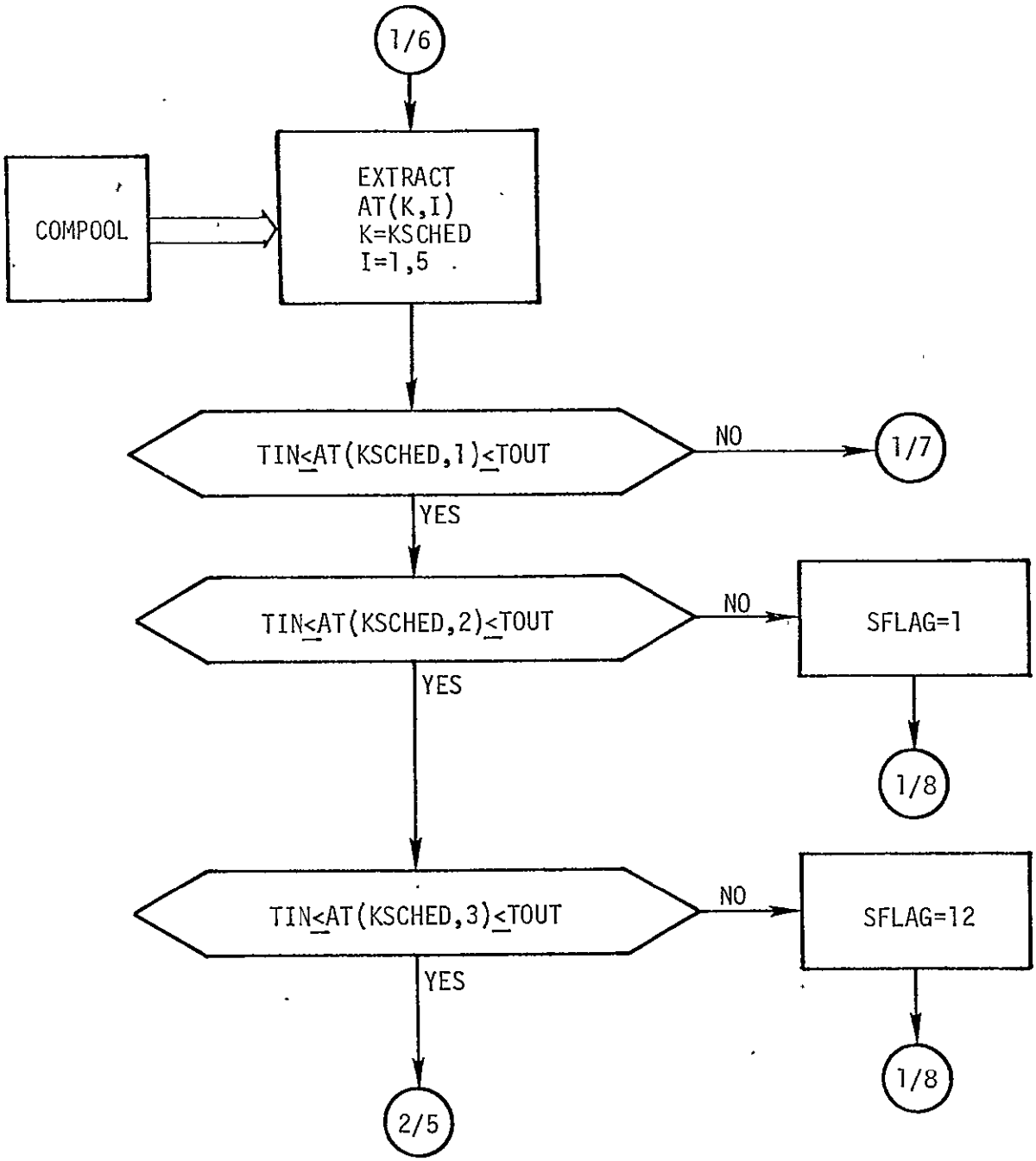


Figure 21. Continued

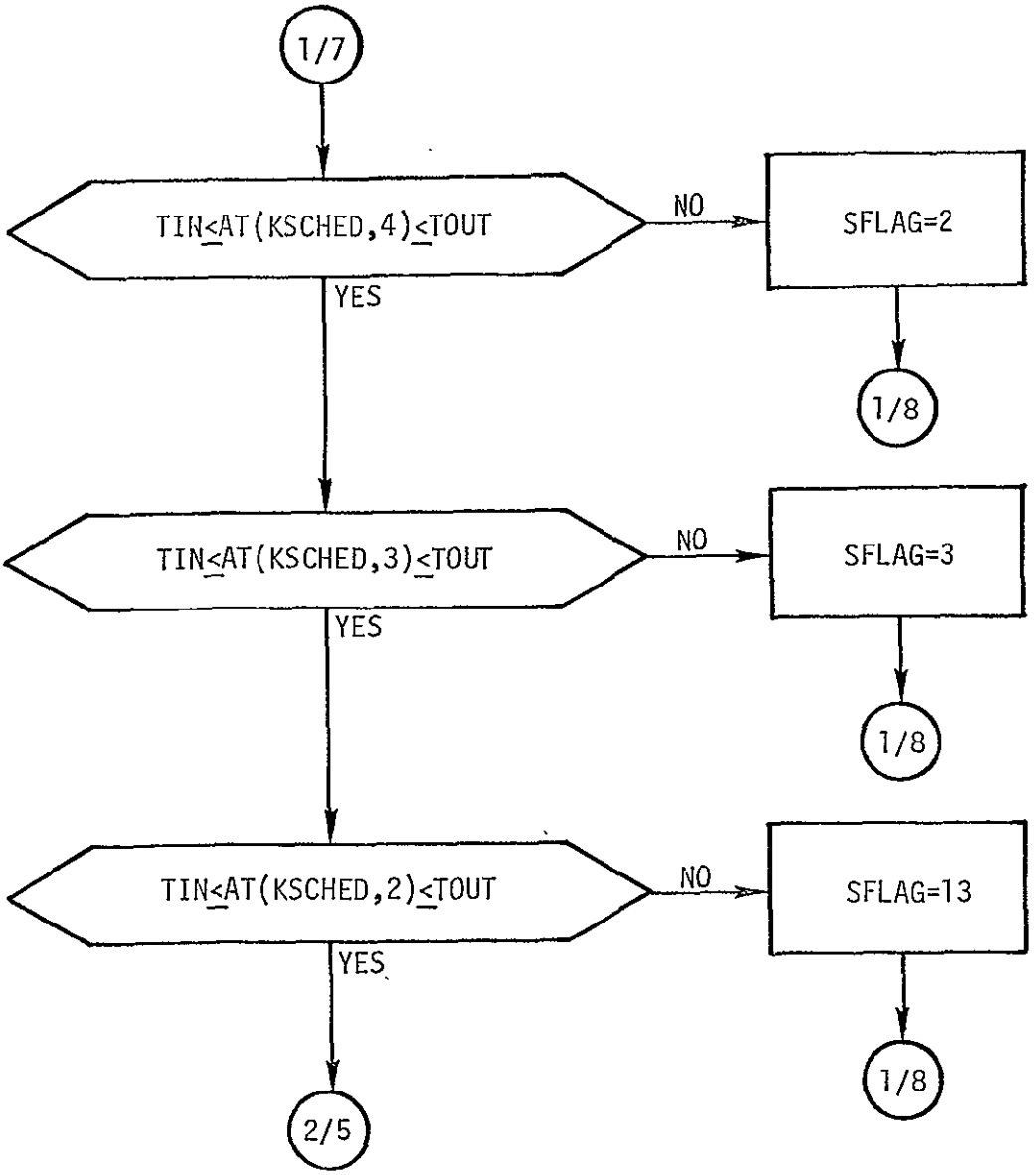


Figure 21. Continued

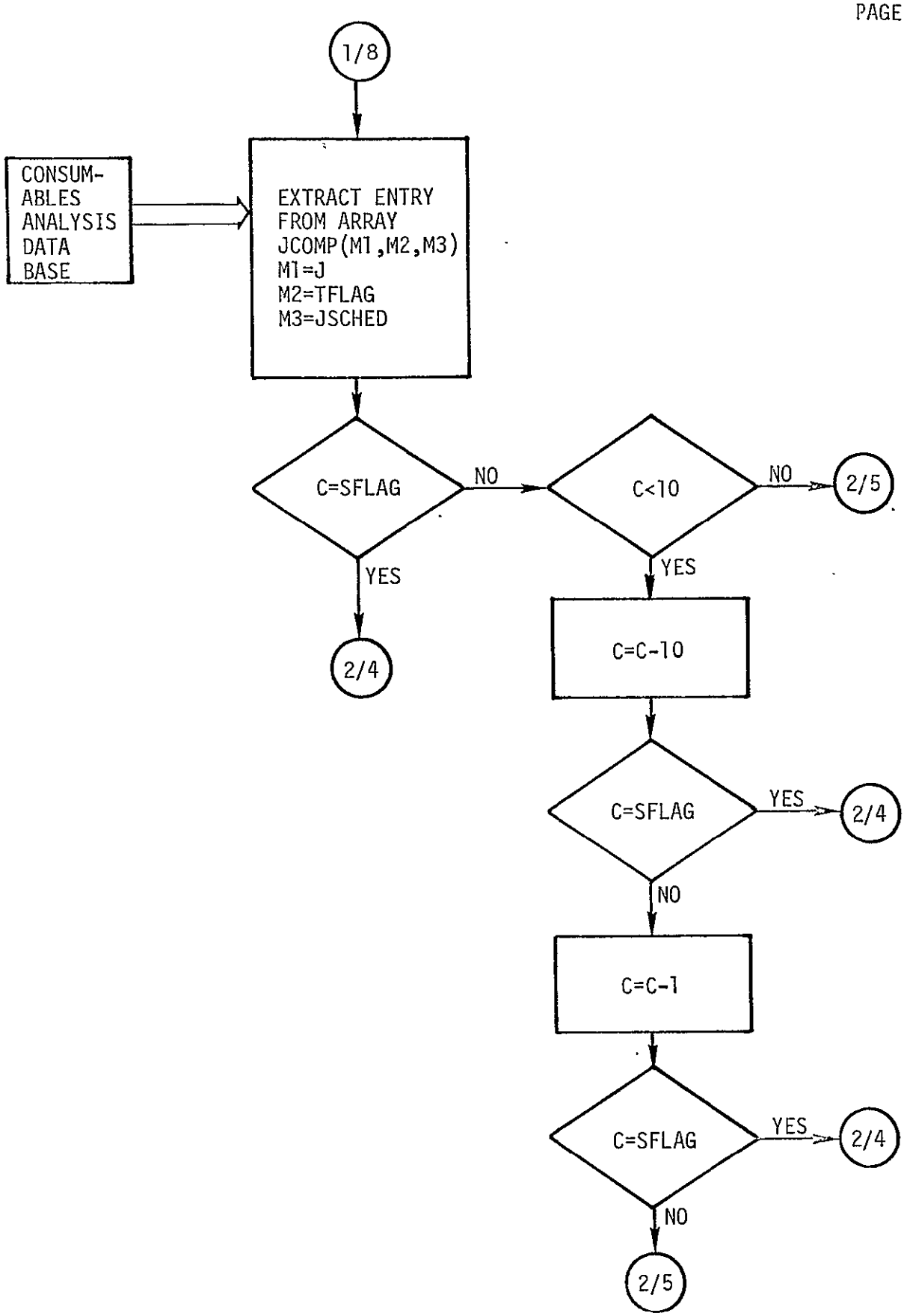


Figure 21. Continued

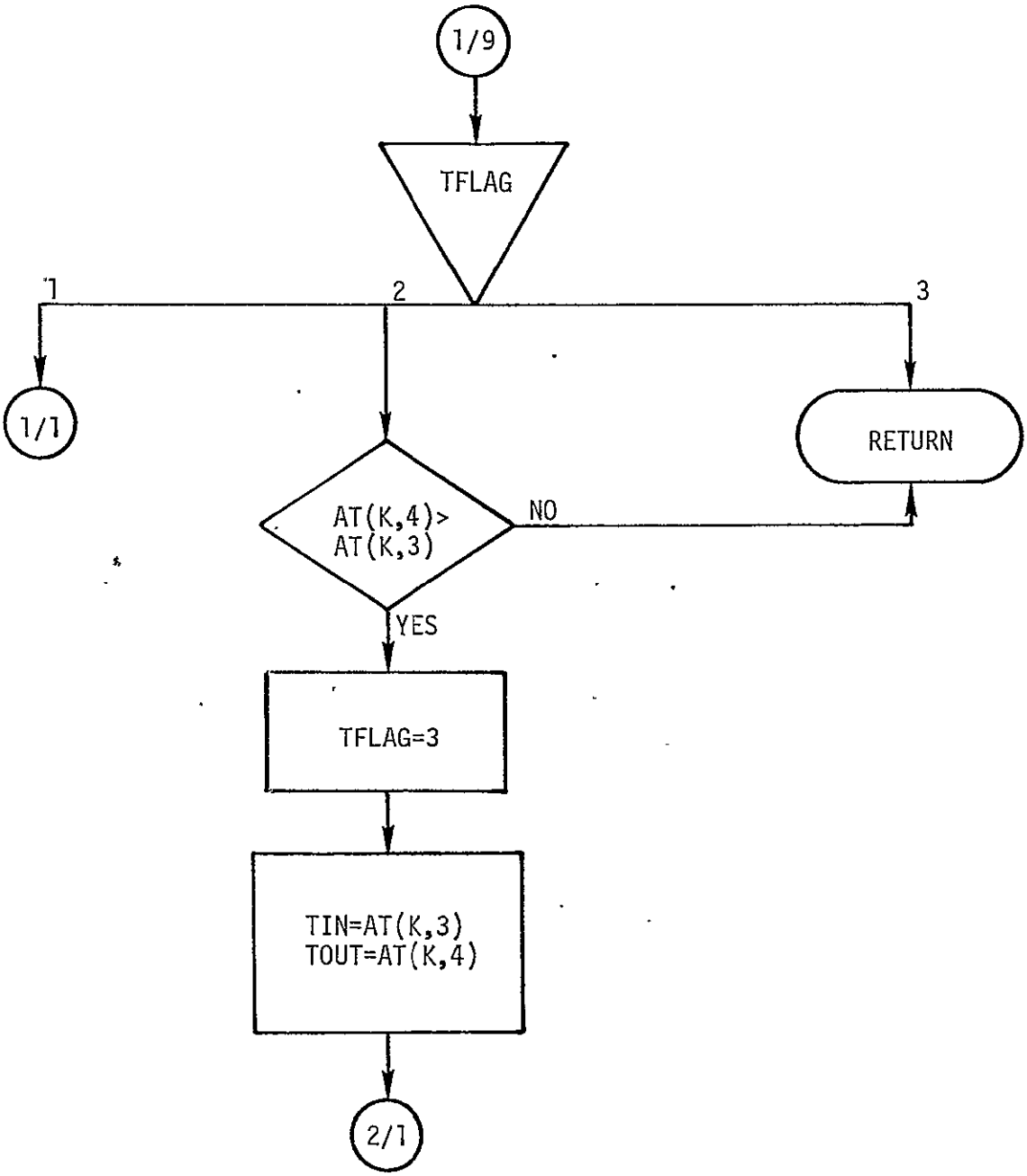


Figure 21. Concluded

5.8 CONSTRAINT ROUTINE

Description - The CONSTRAINT routine determines if an event being scheduled or unscheduled affects the rate limit violation for the consumable subsystems affected by the event. If rate limit violations are created, the Rate Violation Table for the affected consumable subsystem will be updated to note the violation and an interactive warning will be provided to the user on the terminal CRT. Figure 22 presents a skeleton for the Rate Violation Table.

Interface

I/O DEVICES - The terminal CRT unit for output only.
DATA BASE - COMPOOL for both input and output; and the Consumables Analysis Data Base for input.
ROUTINES CALLING CONSTRAINT - RATE routine.
ROUTINES CALLED BY CONSTRAINT - None.

Internal Variables

INR	Index for rate table.
ILR	Index for rate limit array.
SVFLAG	Flag indicating that more than one time point must be considered in checking a time limit: SVFLAG=0 no time point saved 1 time point saved.
TSAVE	If SVFLAG=1, the value of the time point saved.
ISAVE	If SVFLAG=1, the index for the time point saved.
INRSV	Flag indicating search backwards needed: INRSV=0 forward search needed 1 backward search needed.

Input - The CONSTRAINT routine requires the following input data accessed through the COMPOOL:

TMIN	The time to start constraint checking.
TMAX	The time to end constraint checking.
CNUM	The consumable rate table identifier (see Figure 36 for values of CNUM).

RATE(CNUM) CNUM=1,9	The rate tables (rate versus time) for each consumable affected by an event.
TABLE(CNUM) CNUM=1,9	The rate violation tables for each consumable affected by an event (as defined in Figure 22).

The CONSTRAINT routine requires the following input data from the Consumables Analysis Data Base:

TXTND(CNUM) CNUM=1,9	The time to extend constraint checking.
LIMTYP(CNUM)CNUM=1,9	The type of limits to be checked: 1 a minimum limit value 2 a maximum limit value 3 a min/max bounded limit value 4 a maximum/time duration limit value.
LIMNO(CNUM)CNUM=1,9	The number of limits to be checked.
RLIM(CNUM,I)CNUM=1,9 I=1,LIMNO	The value of the rate limit to be checked.
TLIM(CNUM,I)CNUM=1,9 I=1,LIMNO	The time duration for RLIM(CNUM,I).

Processing - The flow diagram of the CONSTRAINT routine is presented in Figure 23.

Output - The CONSTRAINT routine transmits the following data through the COMPOOL:

TABLE(CNUM) CNUM=1,9	The rate violation tables for each consumable affected by an event (as defined in Figure 22).

If a rate limit violation occurs, the following will be displayed to the user on the terminal CRT unit:

WARNING	Rate limit violation created. See Rate Violation Table (CNUM) for details.

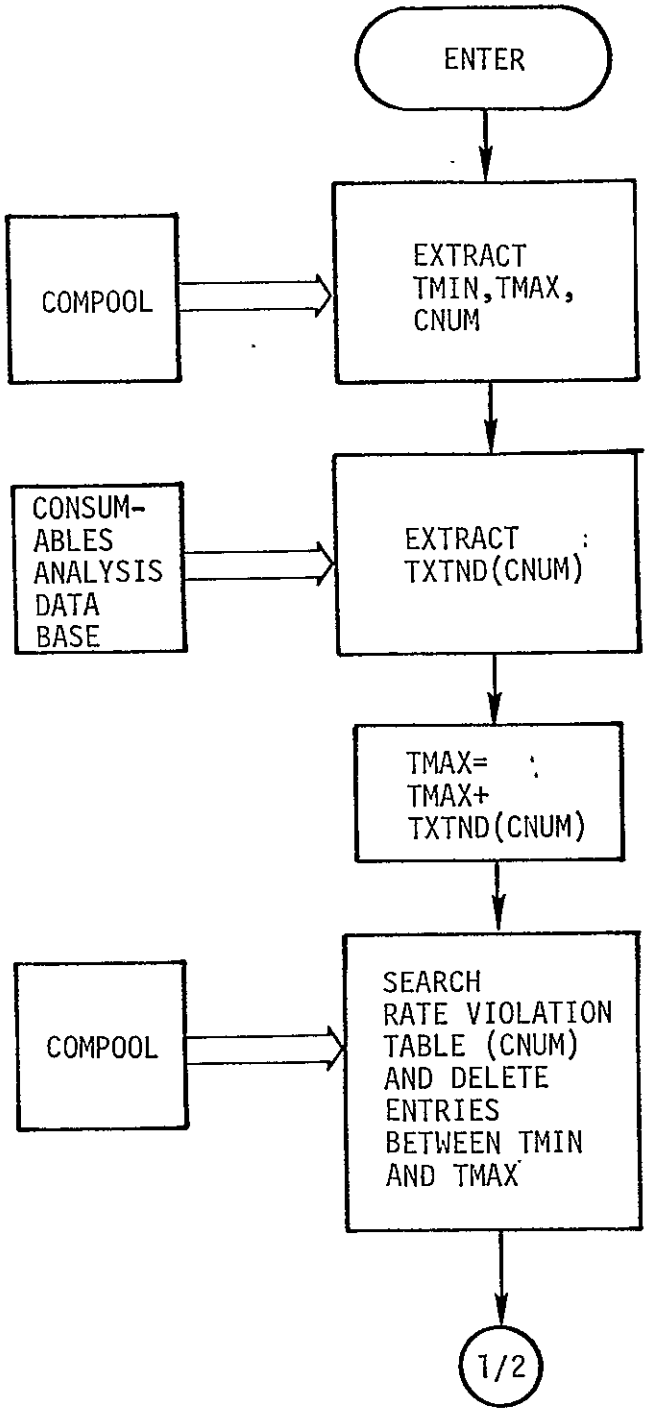
RATE VIOLATION TABLE FOR CNUM

MISSION ID: RUN MODE:

ITEM	TIME OF VIOLATION	RATE AT VIOLATION	LIMIT RATE	LIMIT TIME
1				
2				
•				
•				
•				
n				

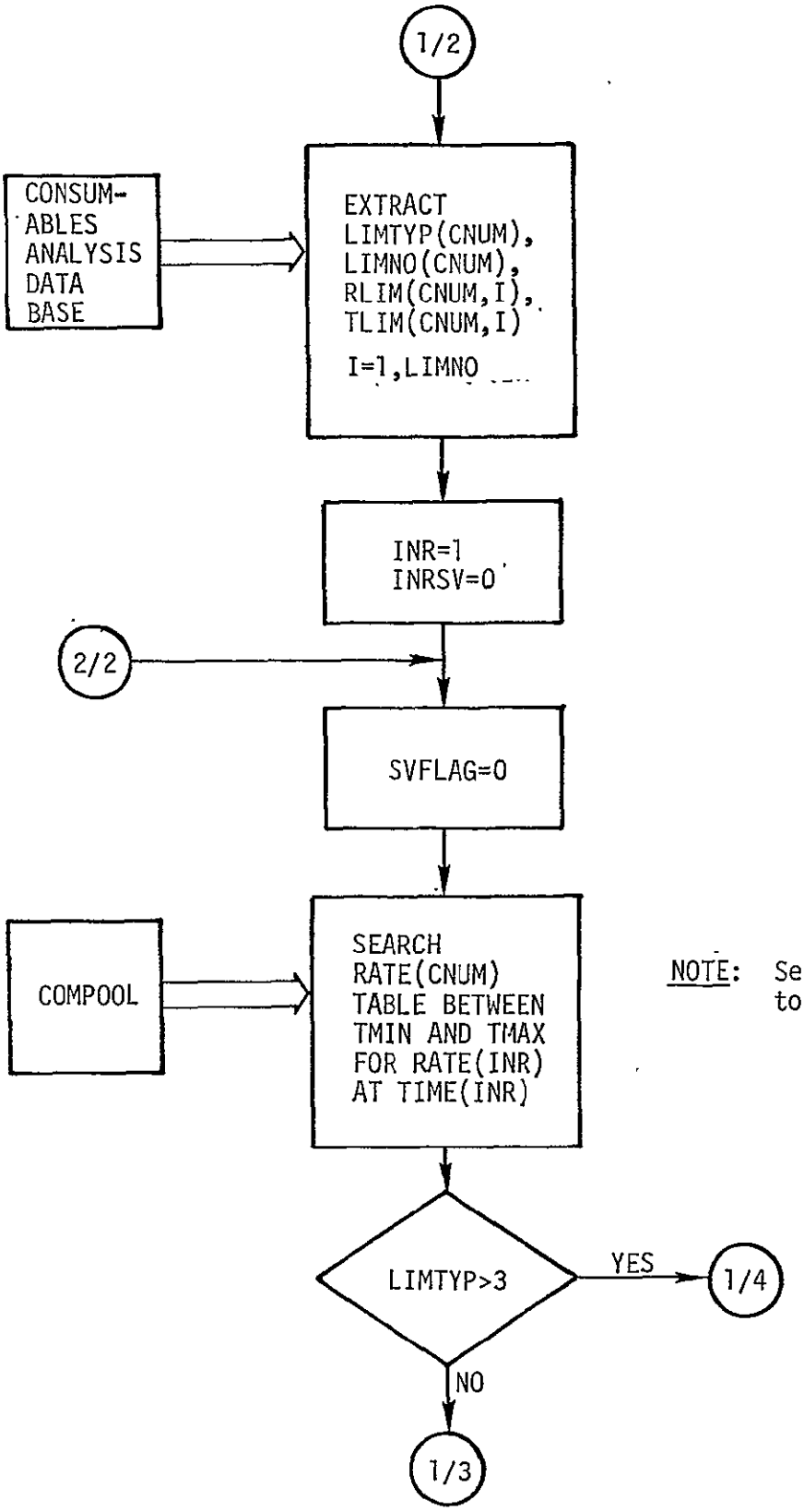
NOTE: CNUM = the consumable rate table identifier (see Figure 36 for the value of CNUM)

Figure 22. The Rate Violation Table Skeleton



NOTE: Search method left to implementor.

Figure 23. Flow Diagram for the CONSTRAINT Routine



NOTE: Search method left to implementor.

Figure 23. Continued

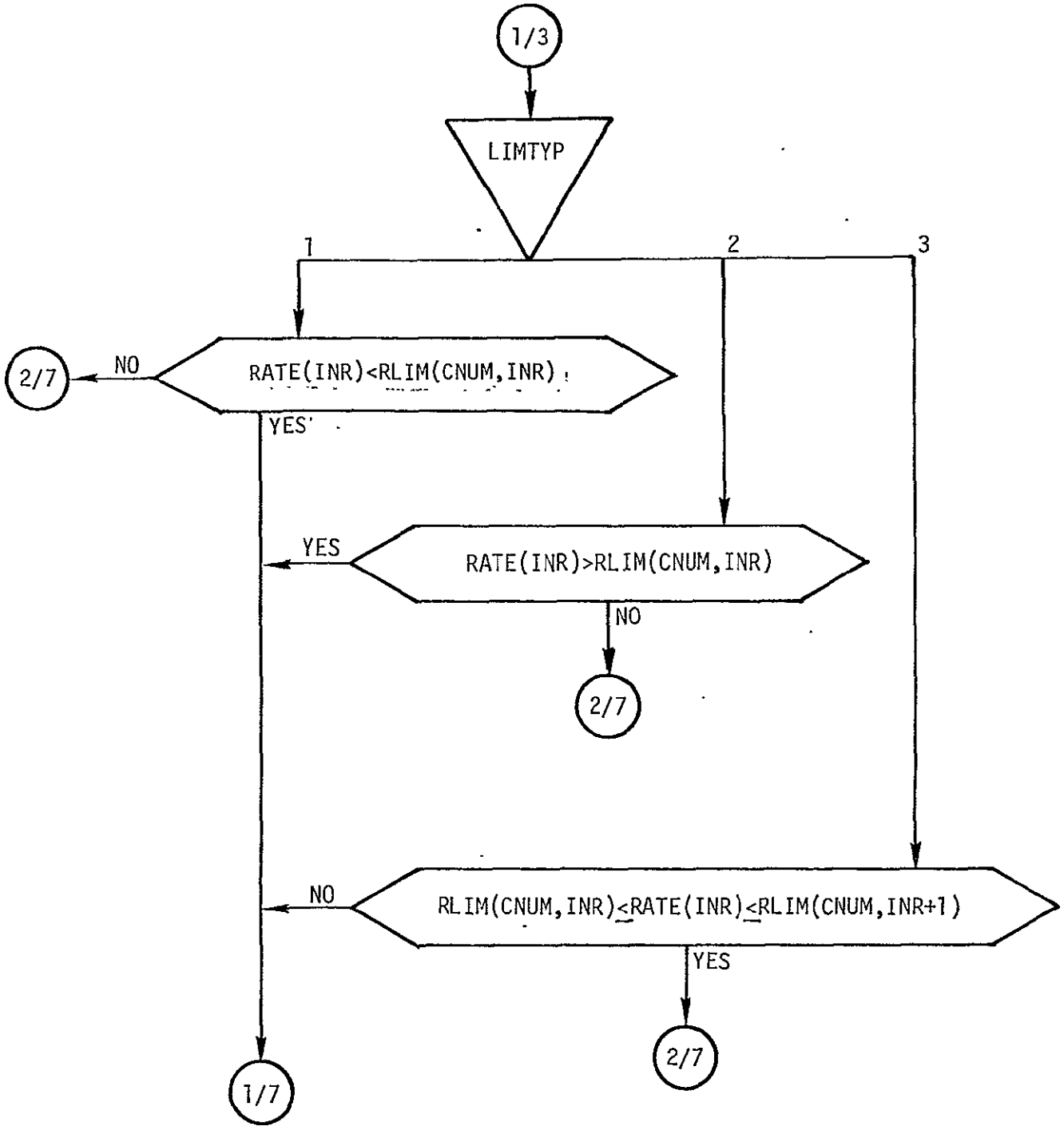


Figure 23. Continued

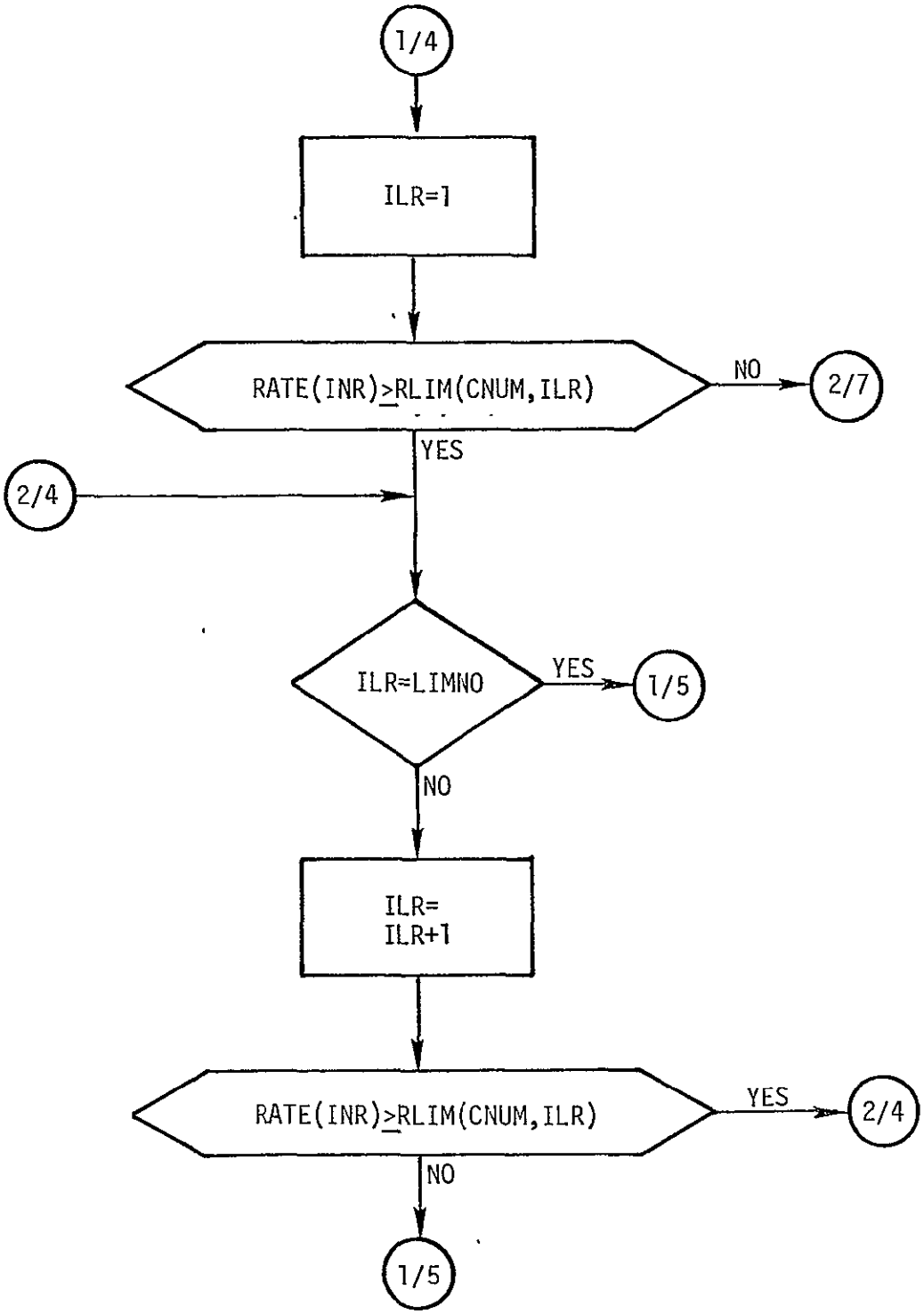
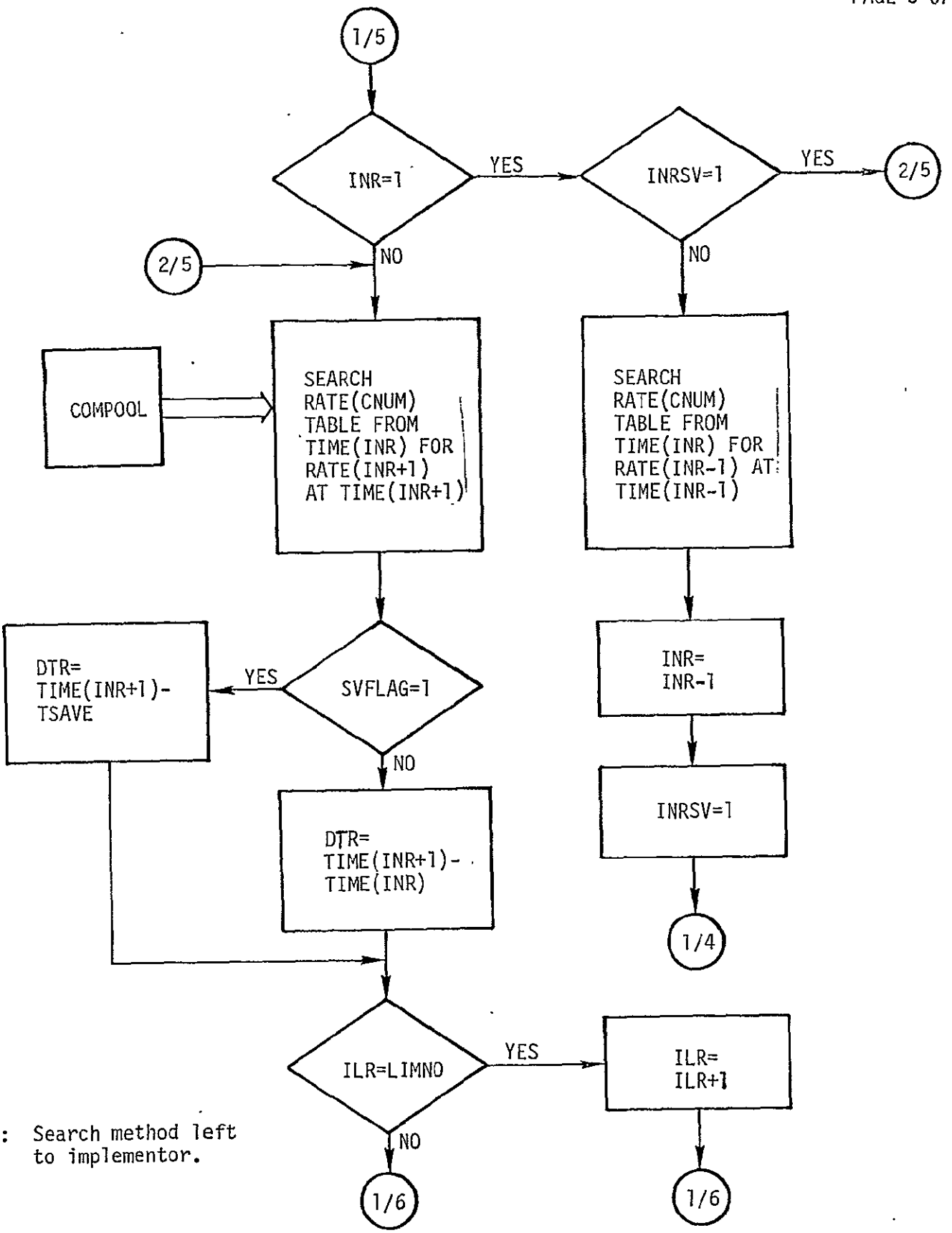


Figure 23. Continued



TE: Search method left to implementor.

Figure 23. Continued

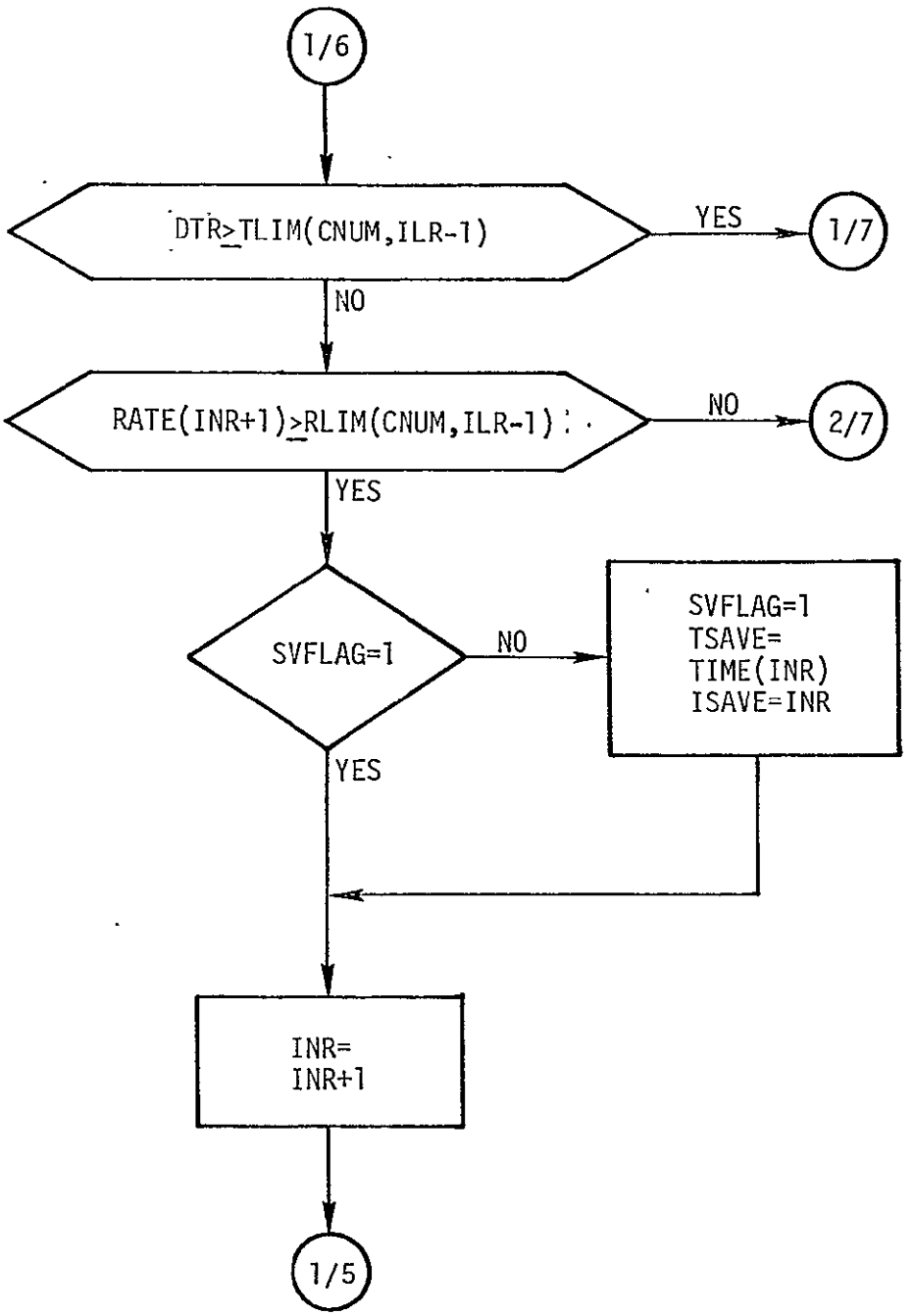


Figure 23. Continued

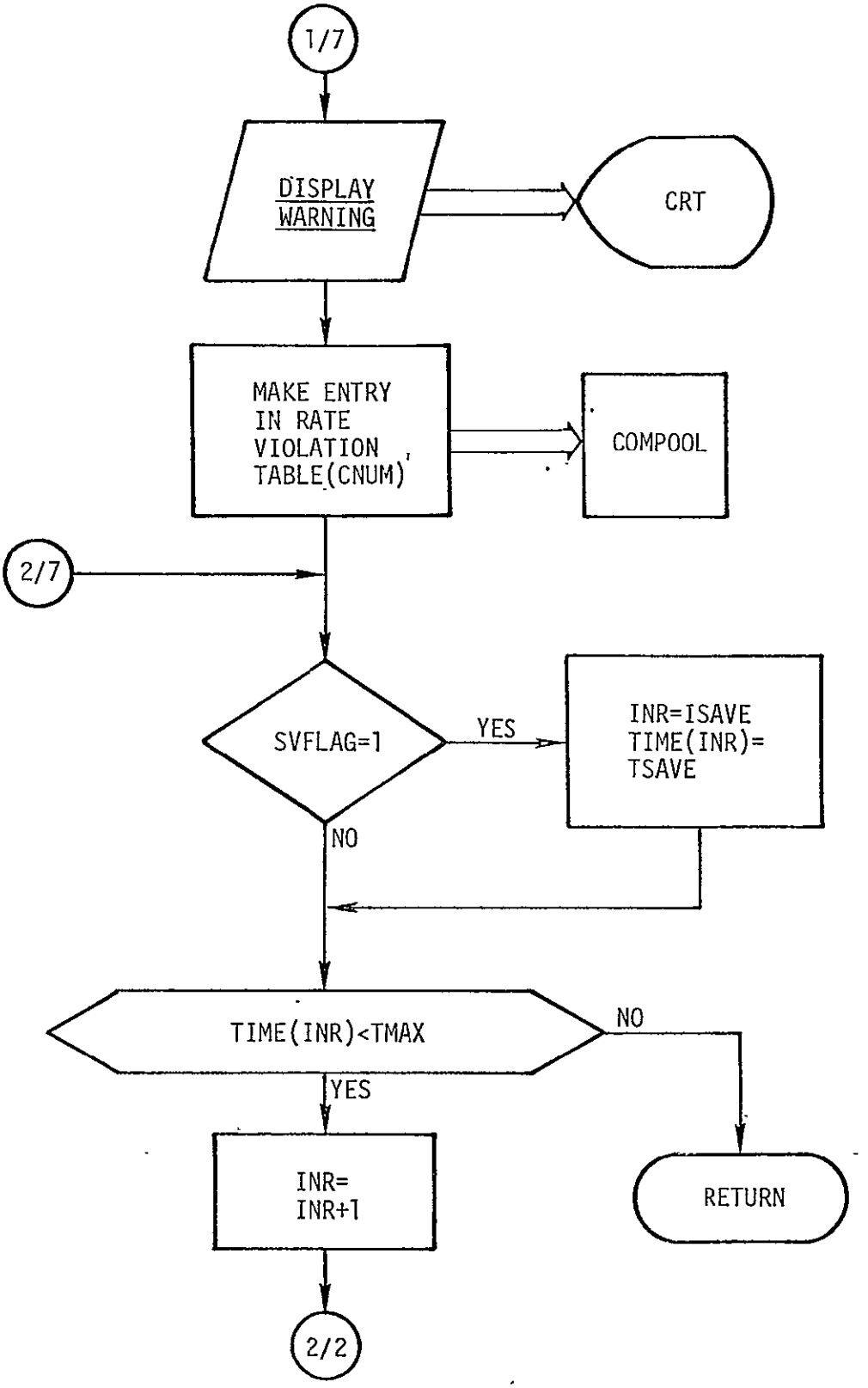


Figure 23. Concluded

5.9 CONSUM HISTORY ROUTINE

Description - The CONSUM HISTORY routine prepares the consumables versus time data for display or storage in the Flight Data Files. The mechanization of this process and the display formats will be defined at time of implementation.

Interface

I/O DEVICES - Terminal KEYBOARD and CRT units.
DATA BASE - COMPOOL for both input and output.
ROUTINES CALLING CONSUM HISTORY - OUTPUT routine.
ROUTINES CALLED BY CONSUM HISTORY - None.

Internal Variables - None.

Input - Will be defined during implementation.

Processing - The CONSUM HISTORY routine is an information management type routine used to prepare the results of consumables analysis for display and/or storage. No flow diagram is necessary.

Output - Will be defined during implementation.

5.10 CONSUM QUANTITIES ROUTINE

Description - The CONSUM QUANTITIES routine calculates the amount used and end-of-mission reserves for each consumable subsystem and prepares the data for display. The mechanization of this process and the display formats will be defined at the time of implementation.

Interface

I/O DEVICES - None.
DATA BASE - COMPOOL for both input and output.
ROUTINES CALLING CONSUM QUANTITIES - OUTPUT routine.
ROUTINES CALLED BY CONSUM QUANTITIES - None.

Internal Variables - None.

Input - Will be defined during implementation.

Processing - The CONSUM QUANTITIES routine is an information management type of routine used to prepare the results of consumables analysis for display. No flow diagram is necessary.

Output - Will be defined during implementation.

5.11 DELETE ROUTINE

Description - The DELETE routine unschedules an event by directly and indirectly updating the File 1 data set. The DELETE routine directly updates the File 1 pool and cross reference parameters to reflect a deleted event and through other control and support routines updates the File 1 temporal parameters and entry data array. The DELETE routine indirectly updates the subsystem rate and conflict tables affected by the unscheduled event.

Interface

I/O DEVICES - None.
DATA BASE - COMPOOL for both input and output.
ROUTINES CALLING DELETE - PLAN and FLIGHT routines.
ROUTINES CALLED BY DELETE - ACTDIS, ACTION, and SEQUENCE routines.

Internal Variables - None.

Input - The DELETE routine requires the following input data accessed through the COMPOOL:

J	Action identifier for the event to be unscheduled.
LINE #	Line # corresponding to the itemized event to be unscheduled.
NNN(I,J)	Activity number for the itemized event to be unscheduled.
ERASE	The ERASE flag is set by the FLIGHT routine in calling the DELETE routine to erase events scheduled beyond the end of a shortened mission: 0 = DELETE called by the PLAN routine. 1 = DELETE called by the FLIGHT routine.
K	Activity number for the itemized event to be unscheduled if ERASE=1.

Processing - The flow diagram of the DELETE routine is presented in Figure 24.

Output - The DELETE routine transmits the following data through the COMPOOL:

K		Activity number for the itemized event to be unscheduled.
M		Pool counter credited for a delete.
NM(I)=K	I=M	The unscheduled activity number is placed in the available activity array.
NNN(I,J)=0	I=LINE # J=J	The specified unscheduled event-activity number cross reference is removed.
IN(J)		The number of ACTION J items scheduled is reduced.
NN(K)=0		The specified unscheduled activity-action cross reference is removed.
ACTION MODE=DELETE		Mode flag for the ACTION routine to unschedule an event.

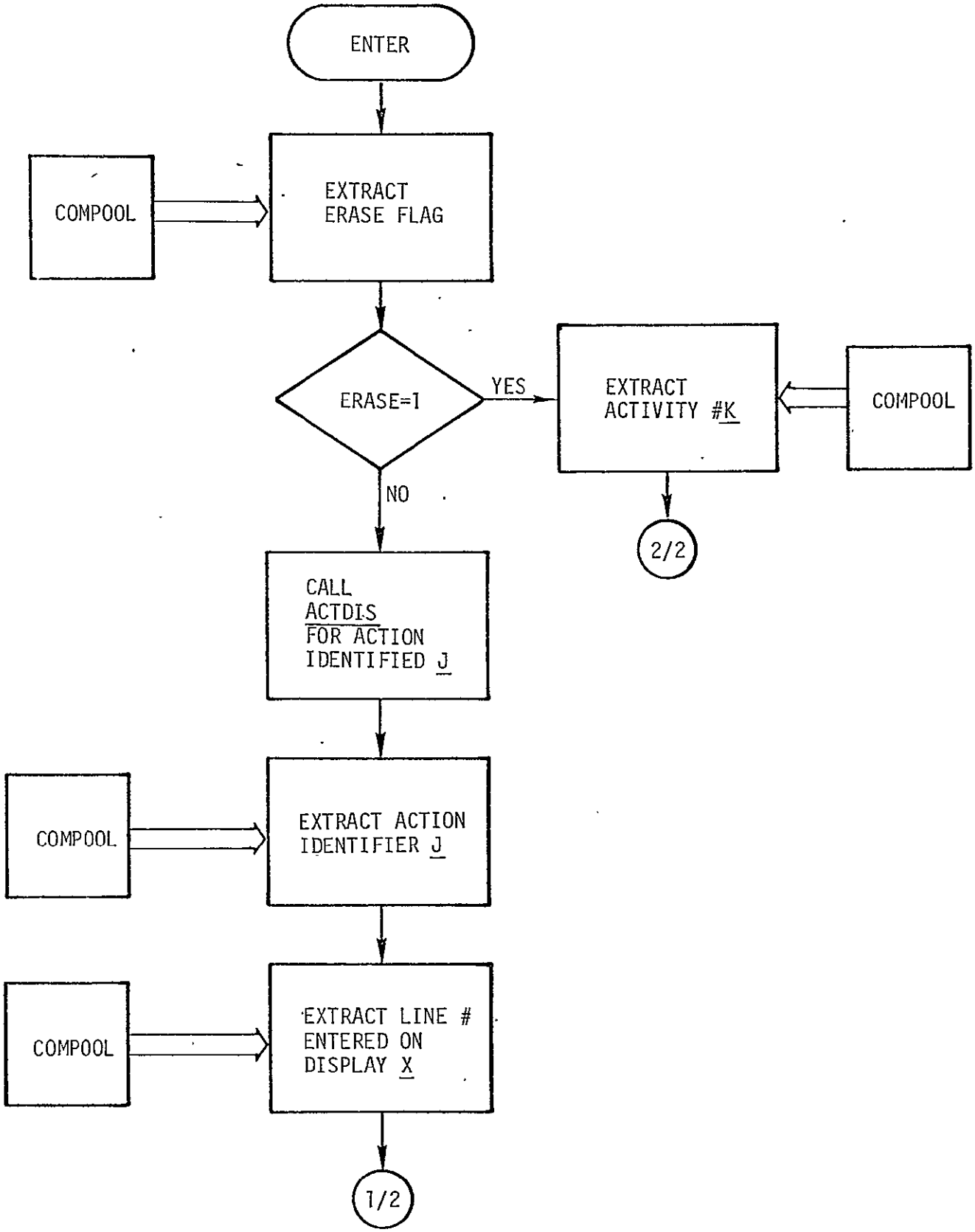


Figure 24. Flow Diagram for the DELETE Routine

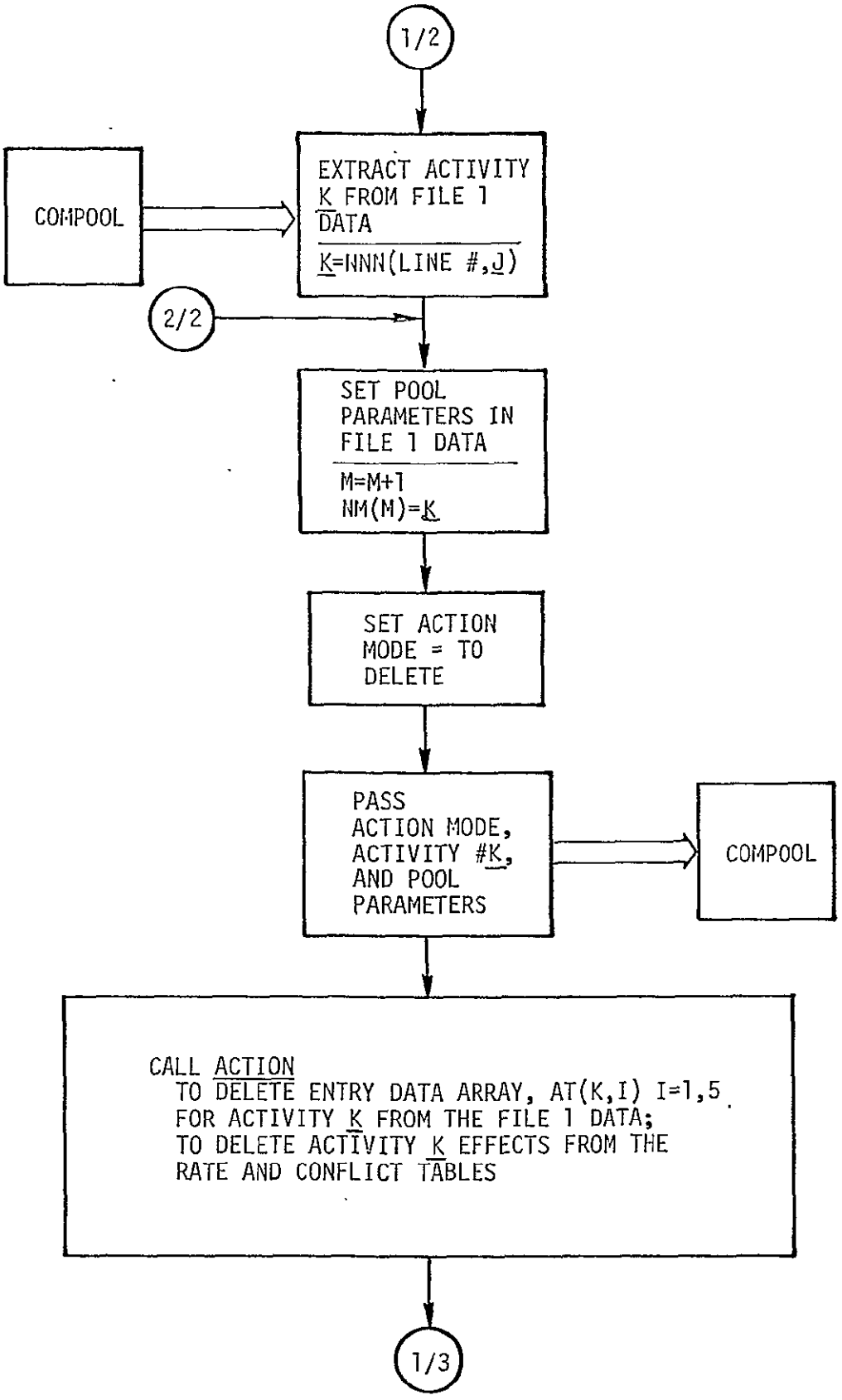


Figure 24. Continued

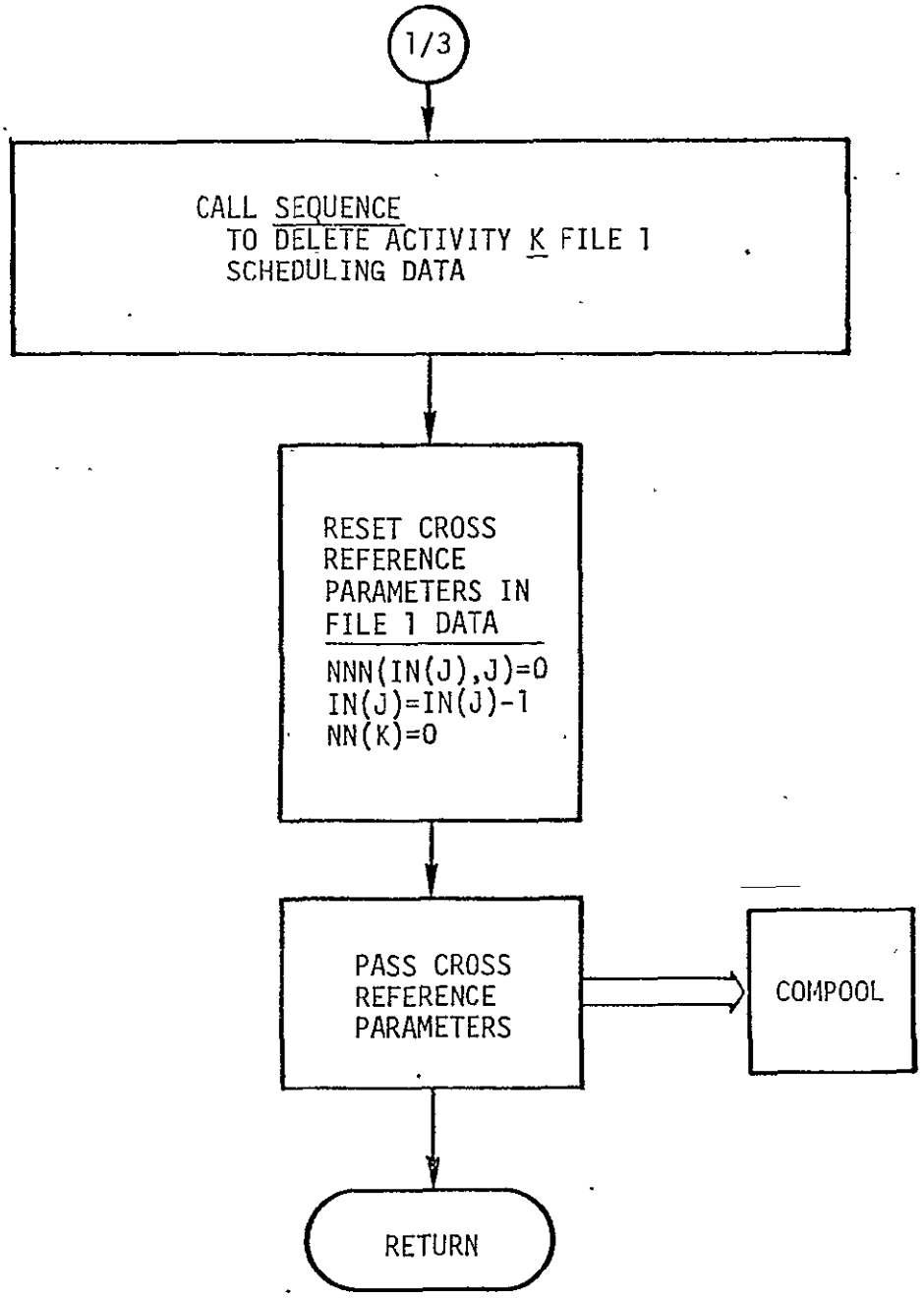


Figure 24. Concluded

5.12 DISPLAY ROUTINE

Description - The DISPLAY routine is a universal routine to execute the interactive User Interface and the read only Output displays on the CRT unit.

Interfaces

I/O DEVICES - CRT unit.
DATA BASE - COMPOOL for both input and output.
ROUTINES CALLING DISPLAY - PLAN and OUTPUT routines.
ROUTINES CALLED BY DISPLAY - none.

Internal Variables - None.

Input - The DISPLAY routine requires the following input data and display skeletons accessed through the COMPOOL:

X	Display variable X = to the Identifier for the display to be executed (see Tables I and II for specific display identifiers)
FLAG(X)	Display skeleton flag for display X
SKELETON(X)	Display skeleton for display X
DATA(X)	Data set for display X
MOD(X)	Mod flag for display X.

Processing - The flow diagram of the DISPLAY routine is presented in Figure 25.

Output - The DISPLAY routine transmits the following data through the COMPOOL:

MAX(X)	The maximum lines on display X.

The DISPLAY routine provides the following to the user through the CRT display unit:

SKELETON(X)	Display skeleton for display X
DATA(X)	Data set for display X
MOD(X)	Mod flag for display X.

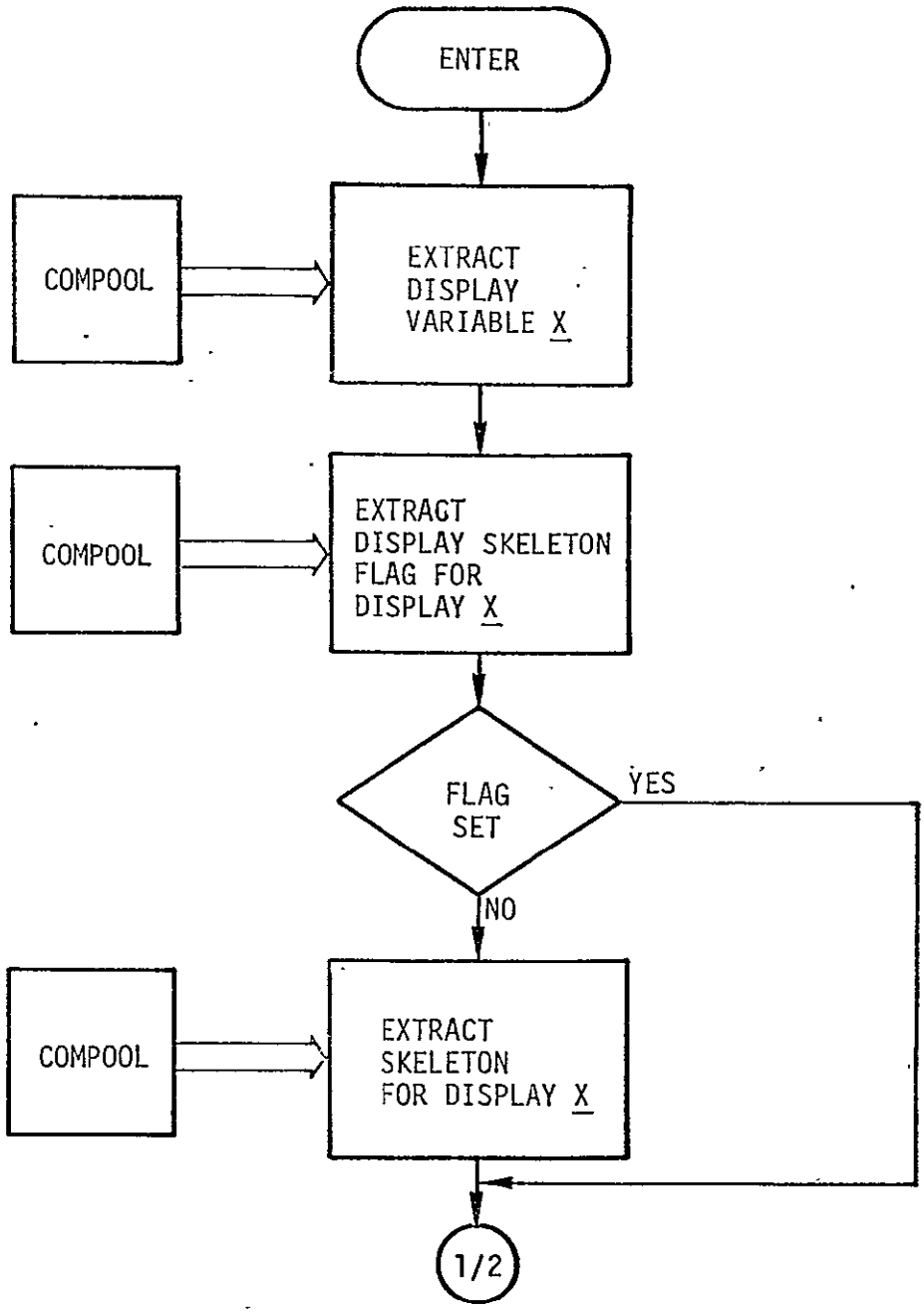


Figure 25. Flow Diagram for the DISPLAY Routine

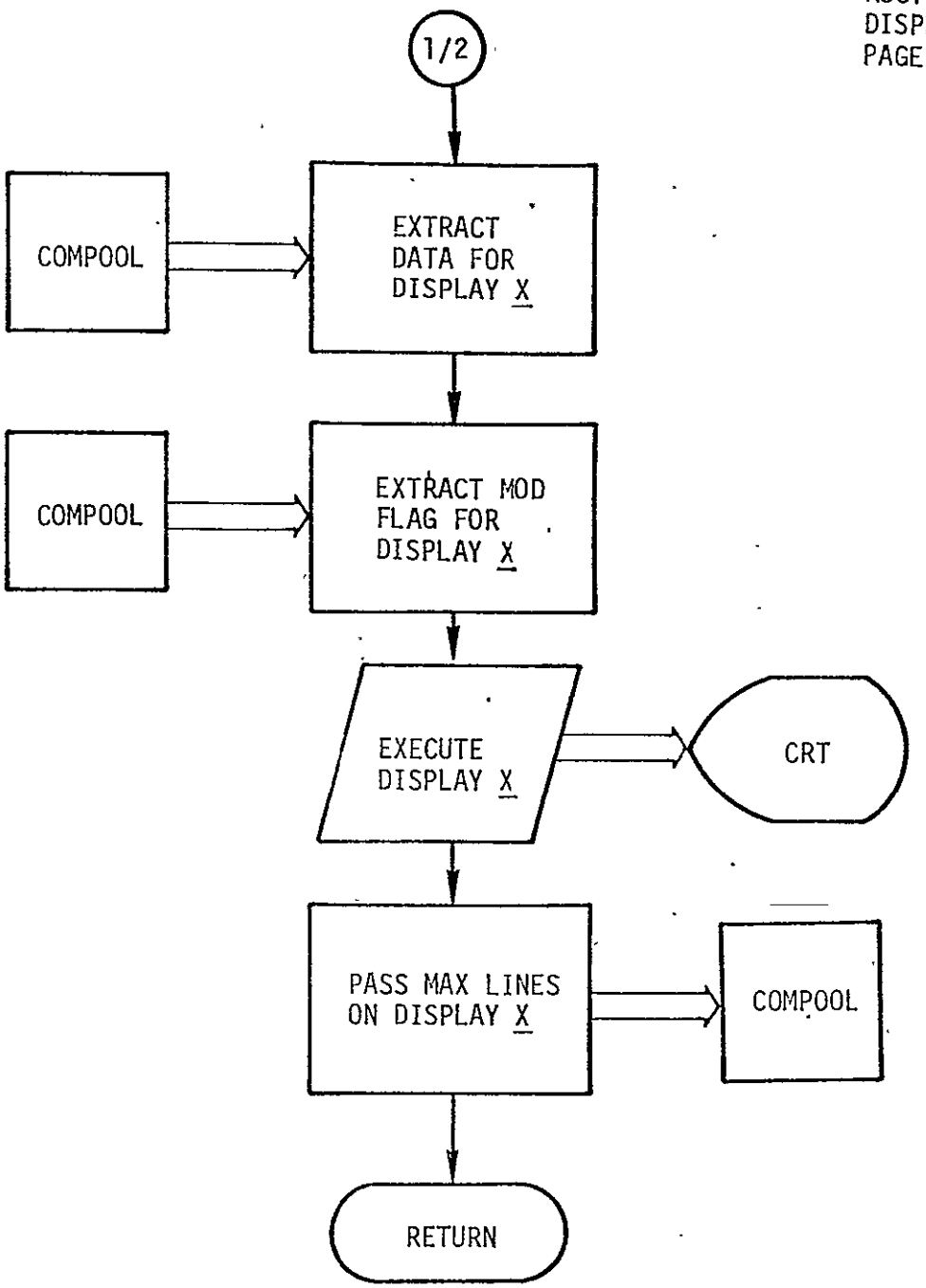


Figure 25. Concluded

5.13 EVENT CHART ROUTINE

Description - The EVENT CHART routine can be called in either the Event Mode or Active Mode to prepare the event chart data for display. These data include the influence variables, factors, consumable usage for each subsystem, and consumable kit requirements. The mechanization of this process and the display formats are will be defined at time of implementation.

Interface

I/O DEVICES - none.
DATA BASE - COMPOOL for both input and output.
ROUTINES CALLING EVENT CHART - OUTPUT routine.
ROUTINES CALLED BY EVENT CHART - none.

Internal Variables - None.

Input - Will be defined during implementation.

Processing - A functional flow diagram of the EVENT CHART routine is presented in Figure 26.

Output - Will be defined during implementation.

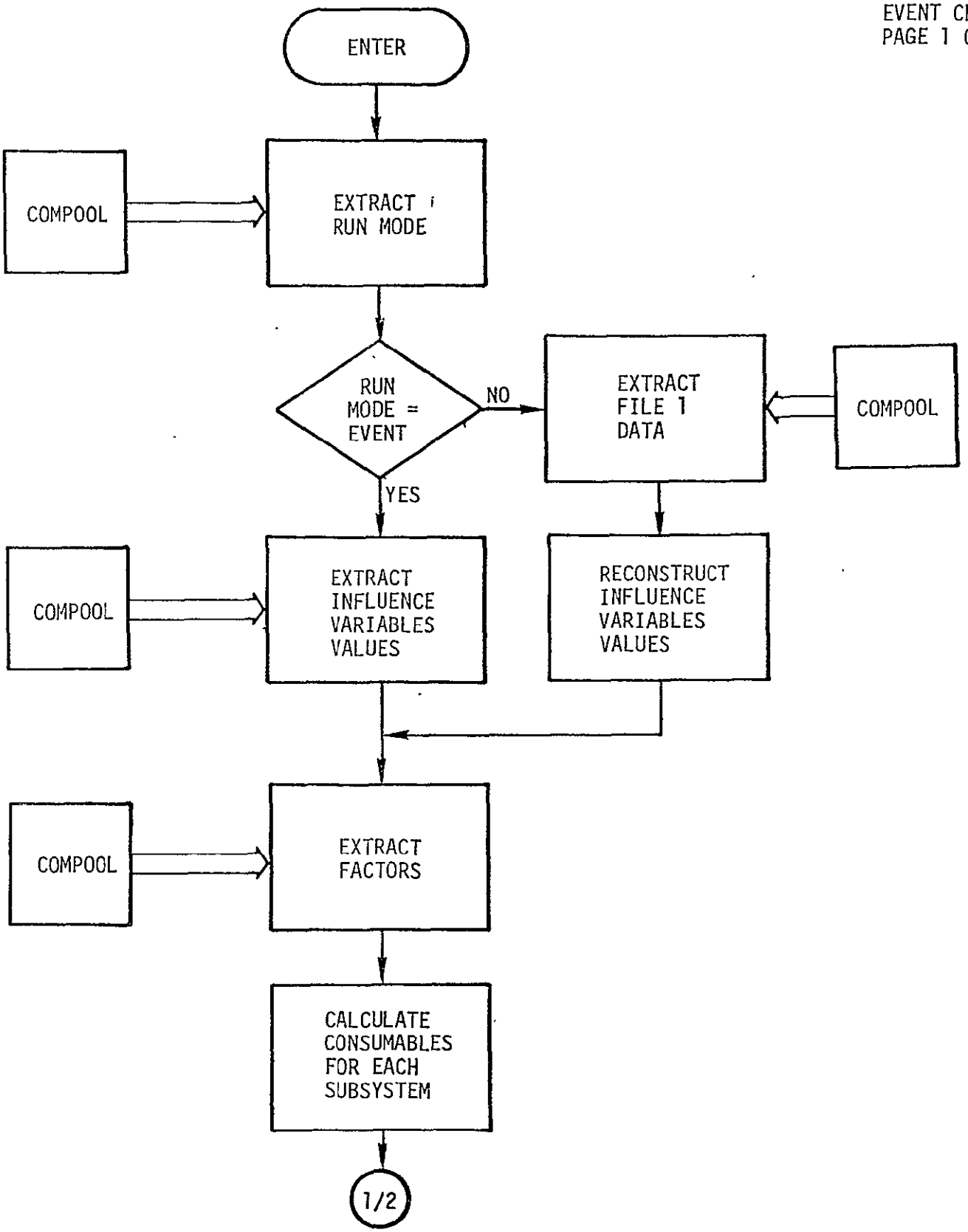


Figure 26. Flow Diagram for the EVENT CHART Routine

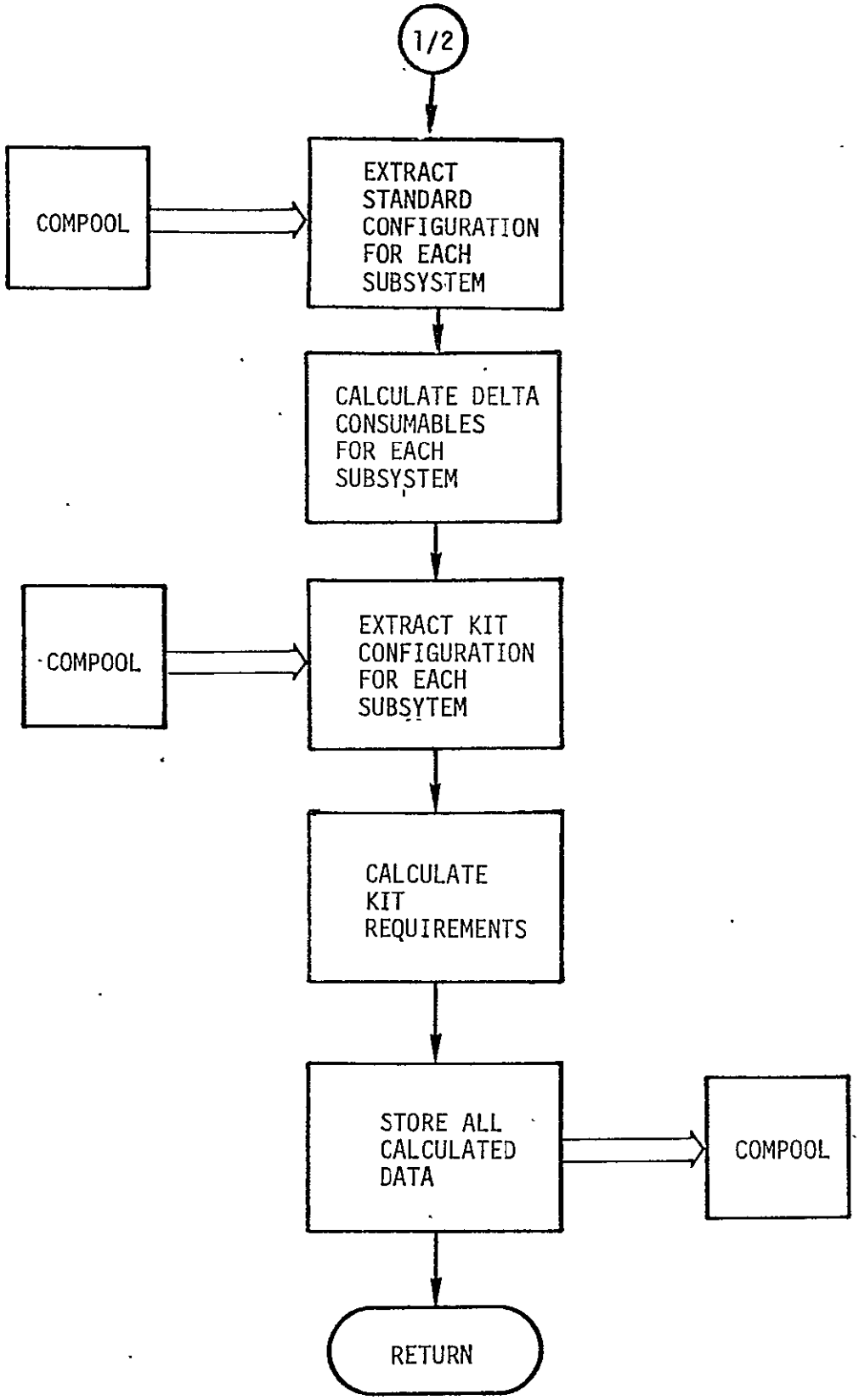


Figure 26. Concluded

5.14 FILE ONE ROUTINE

Description - The FILE ONE routine prepares the flight File 1 data set for use by the Mission Planning Processor.

Interface

I/O DEVICES - Terminal KEYBOARD unit.
DATA BASE - Flight Data File for input and the COMPOOL for output.
ROUTINES CALLING FILE ONE - EXEC routine.
ROUTINES CALLED BY FILE ONE - None.

Internal Variables - None.

Input - The FILE ONE routine requires the following data input through the terminal KEYBOARD unit:

MISSION ID	The mission identifier.

The FILE ONE routine requires as input the File 1 data set stored for the identified mission. Table III defines the File 1 data set parameters.

Processing - The flow diagram of the FILE ONE routine is presented in Figure 27.

Output - The FILE ONE routine transmits the File 1 data set for the identified mission through the COMPOOL. Table III defines the File 1 data set parameters.

C. 2

Table III. File 1 Data Set

Parameter Type	Parameter Name	Parameter Format	Parameter Description
Pool	N	Fixed	Entry counter
Pool	M	Fixed	Pool Counter
Pool	NM(I)	I=1,M	Available activity number array
Temporal	NOI	Fixed	Number of entries in sequence array
Temporal	IT(I)	I=1,NOI	Sequence array of activities
Temporal	TIM(I,L)	I=1,NOI L=1,2	Start and end times of activities L=1 Minimum start time of activity IT(I) 2 Maximum end time of activity IT(I)
Cross Ref.	IN(J)	J=1,23	Number of ACTION J items scheduled
Cross Ref.	NN(K)=J	K=1,N	Activity-Action cross reference (i.e., activity number K is an ACTION J)
Cross Ref.	NNN(I,J)=K	I=1,IN(J) J=1,23	Event-Activity cross reference (i.e., activity number for the Ith event of ACTION J scheduled)
Entry	AT(K,I)	I=1,5	Entry data array for activity K I=1 prep start time 2 reference start time 3 reference stop time 4 post end time 5 special parameter, a function of ACTION Identifier J: J=1,2,4,6, or 7; AT(K,5)=ΔV J=9,10,20,21, or 22; AT(K,5)=Number of crew

Table III. . - Continued

Parameter Type	Parameter Name	Parameter Format	Parameter Description
Data	BPT(I)	I=1,11	Block phase times I=1 prelaunch start 2 prelaunch stop/ascent start 3 liftoff 4 MECO 5 ETS 6 OMS ignition/on-orbit start 7 on-orbit stop/deorbit start 8 deorbit burn ignition 9 deorbit stop/entry start 10 rollout 11 entry/land stop
Data	BPDT(I)	I=1,13	Block phase delta times I=1 prelaunch 2 ascent 3 GSE-liftoff 4 liftoff-MECO 5 MECO-ETS 6 ETS-OMS ignition 7 on-orbit 8 deorbit 9 prep-burn 10 burn to entry interface 11 entry/land 12 entry interface-rollout 13 rollout-GSE

Table III. - Concluded

Parameter Type	Parameter Name	Parameter Format	Parameter Description
Data	CONFG(I)	I=1,11	Mission Configuration Data I=1 crew size 2 number of EPS consumables kits 3 number of OMS consumables kits 4 number of EPS (LIOH) kits 5 orbital inclination 6 day of launch 7 month of launch 8 year of launch 9 gross weight at liftoff 10 launch site ID 11 landing site ID

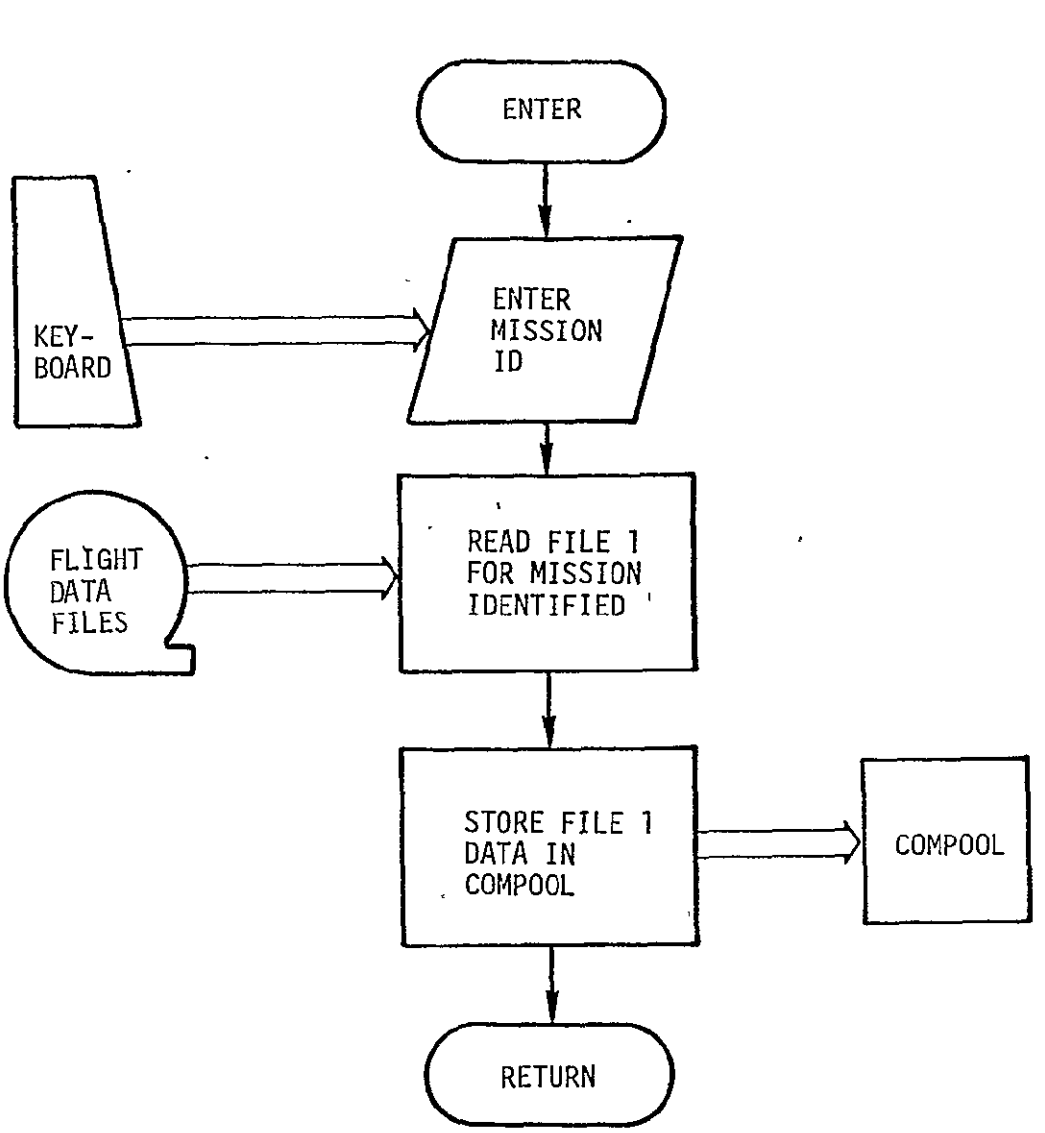


Figure 27. Flow Diagram for the FILE ONE Routine

5.15. FILE STORE ROUTINE

Description - The FILE STORE routine transfers the results of the Mission Planning Processor to permanent storage in the Flight Data File. The user has the option of selecting the type of data to be stored.

Interface

I/O DEVICES - None.
DATA BASE - COMPOOL for input and the Flight Data File for output.
ROUTINES CALLING FILE STORE - OUTPUT routine.
ROUTINES CALLED BY FILE STORE - None.

Internal Variables - None.

Input - The FILE STORE routine requires the following input data accessed through COMPOOL:

Y	Store variable Y Y = 1 store File 1 data set 2 store Files 1 and 2 data sets 3 store Files 1, 2, and 3 data sets 4 store File 0 data set
MISSION ID	Mission identifier
DATA(0)	File 0 data set (parameters are defined in Volume IV)
DATA(1)	File 1 data set (parameters are defined in Table III)
DATA(2)	File 2 data set (parameters are defined in Volume IV)
DATA(3)	File 3 data set (parameters are defined in Volume IV)

Processing - No flow diagram is necessary.

Output - The FILE STORE routine transfers the following data to the Flight Data File for the mission identified:

DATA(0)	File 0 data set (paremeters are defined in Volume IV)
DATA(1)	File 1 data set (parameters are defined in Table III)
DATA(2)	File 2 data set (parameters are defined in Volume IV)
DATA(3)	File 3 data set (parameters are defined in Volume IV).

5.16 FILE ZERO ROUTINE

Description - The FILE ZERO routine prepares the flight File 0 data set for use by the Mission Planning Processor. The File 0 data set contains event chart related data to reinitialize processing in the Event Mode only.

Interface

I/O DEVICES - Terminal KEYBOARD unit.
DATA BASE - Flight Data File for input and the COMPOOL for output.
ROUTINES CALLING FILE ZERO - IV INPUT routine.
ROUTINES CALLED BY FILE ZERO - none.

Internal Variables - None.

Input - The FILE ZERO routine requires the following data input through the terminal KEYBOARD unit:

MISSION ID — The mission identifier.

The FILE ZERO routine requires as input the File 0 data set stored for the identified mission. The File 0 data set is defined in Volume IV.

Processing - The flow diagram of the FILE ZERO routine is presented in Figure 28.

Output - The FILE ZERO routine transmits the File 0 data set for the identified mission through the COMPOOL. The File 0 data set is defined in Volume IV.

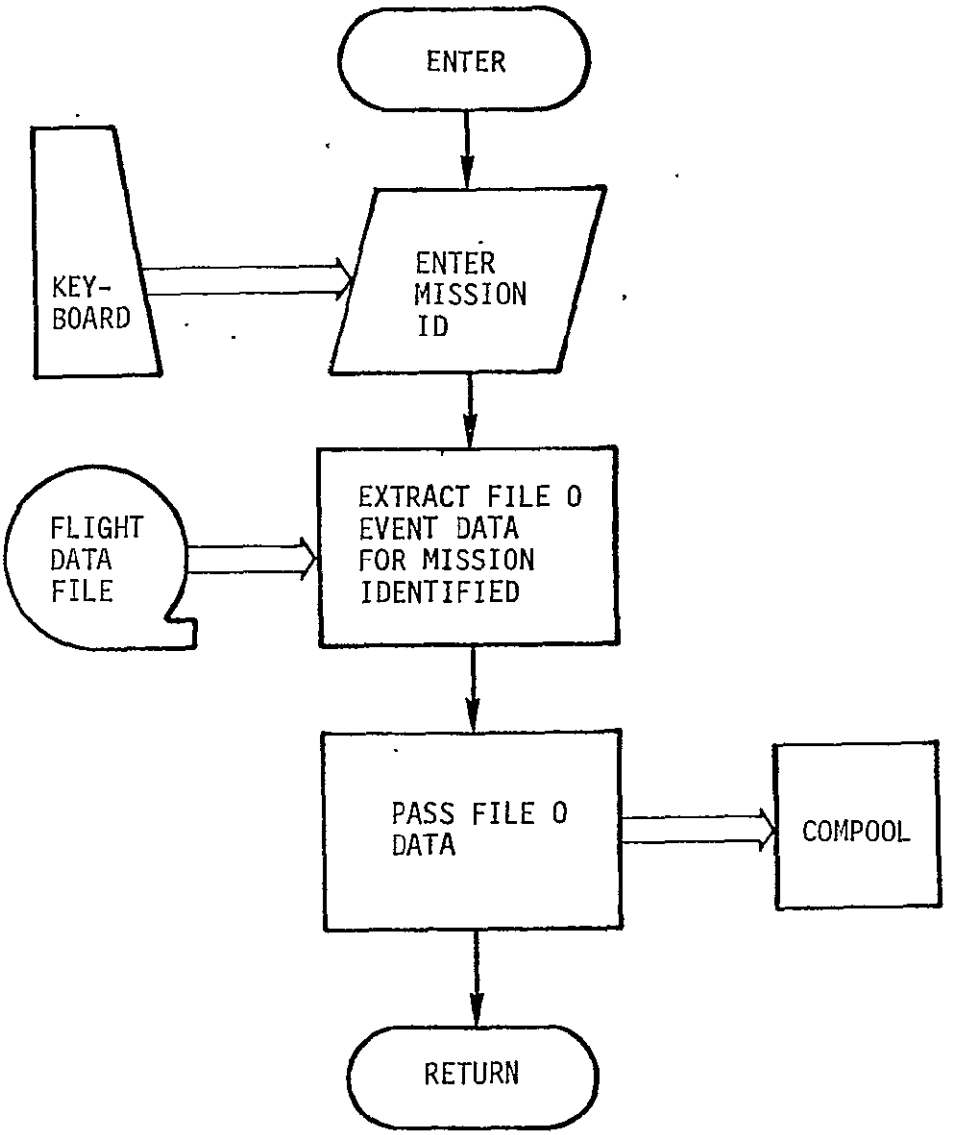


Figure 28. Flow Diagram for the FILE ZERO Routine

5.17 FLIGHT ROUTINE

Description - The FLIGHT routine updates all flight block parameters and common block rates if any phase block time is changed in the active run mode. If the on-orbit phase is shortened, the FLIGHT routine will remove the affects of activities scheduled beyond the new deorbit time.

Interface

I/O DEVICES - None.
DATA BASE - COMPOOL for both input and output.
ROUTINES CALLING FLIGHT - PLAN routine.
ROUTINES CALLED BY FLIGHT - SPECIAL, RATE, and DELETE routines.

Internal Variables - None.

Input - The FLIGHT routine requires the following input accessed through COMPOOL:

APT(I)	I=1,11	Block phase time changes I=1 prelaunch start 2 prelaunch stop/ascent start 3 liftoff 4 MECO 5 ETS 6 OMS ignition/on-orbit start 7 on-orbit stop/deorbit start 8 deorbit burn ignition 9 deorbit stop/entry start 10 rollout 11 entry/land stop
APTD(I)	I=1,13	Block phase delta time changes I=1 prelaunch 2 ascent 3 GSE-liftoff 4 lift-off 5 MECO-ETS 6 ETS-OMS ignition 7 on-orbit 8 deorbit 9 prep-burn 10 burn to entry interface 11 entry/land 12 entry interface-rollout 13 rollout-GSE

J		The ACTION identifier required by the common block to be scheduled or unscheduled (see Table I for the values of J).
BPT(7)		The deorbit phase start time for the File 1 data set block phase time array.
NOI		Number of entries in sequence array.
IT(I)	I=1,NOI	Sequence array of activities.
TIM(I,L)	I=1,NOI L=1,2	Start and end times of activities: L=1 minimum start time of activity IT(I) 2 maximum end time of activity IT(I).
IN(J)	J=1,23	Number of ACTION J items scheduled.
NN(K)=J	K=1,N	Activity-Action cross reference.
NNN(I,J)=K	I=1,IN(J) J=1,23	Event-Activity cross reference.

Processing - The flow diagram of the FLIGHT routine is presented in Figure 29.

Output - The FLIGHT routine transmits the following data through the COMPOOL:

BPT(I)	I=1,11	Block phase times I=1 prelaunch start 2 prelaunch stop/ascent start 3 liftoff 4 MECO 5 ETS 6 OMS ignition/on-orbit start 7 on-orbit stop/deorbit start 8 deorbit burn ignition 9 deorbit stop/entry start 10 rollout 11 entry/land stop
BPTD(I)	I=1,13	Block phase delta times I=1 prelaunch 2 ascent 3 GSE-liftoff 4 liftoff-MECO 5 MECO-ETS 6 ETS-OMS ignition 7 on-orbit 8 deorbit 9 prep-burn 10 burn to entry interface 11 entry/land 12 entry interface-rollout 13 rollout-GSE

J	The ACTION identifier required by the event or common block to be scheduled or unscheduled (see Table I for the values of J).
K	Activity number for the event to be unscheduled.
ACTION MODE	Mode flag for the ACTION routine to schedule or unschedule an event: ADD = schedule an event DELETE = unschedule an event.
ERASE	The ERASE flag is set by the FLIGHT routine in calling the DELETE routine to erase events scheduled beyond the end of a shortened mission: 0 = DELETE called by the PLAN routine 1 = DELETE called by the FLIGHT routine.

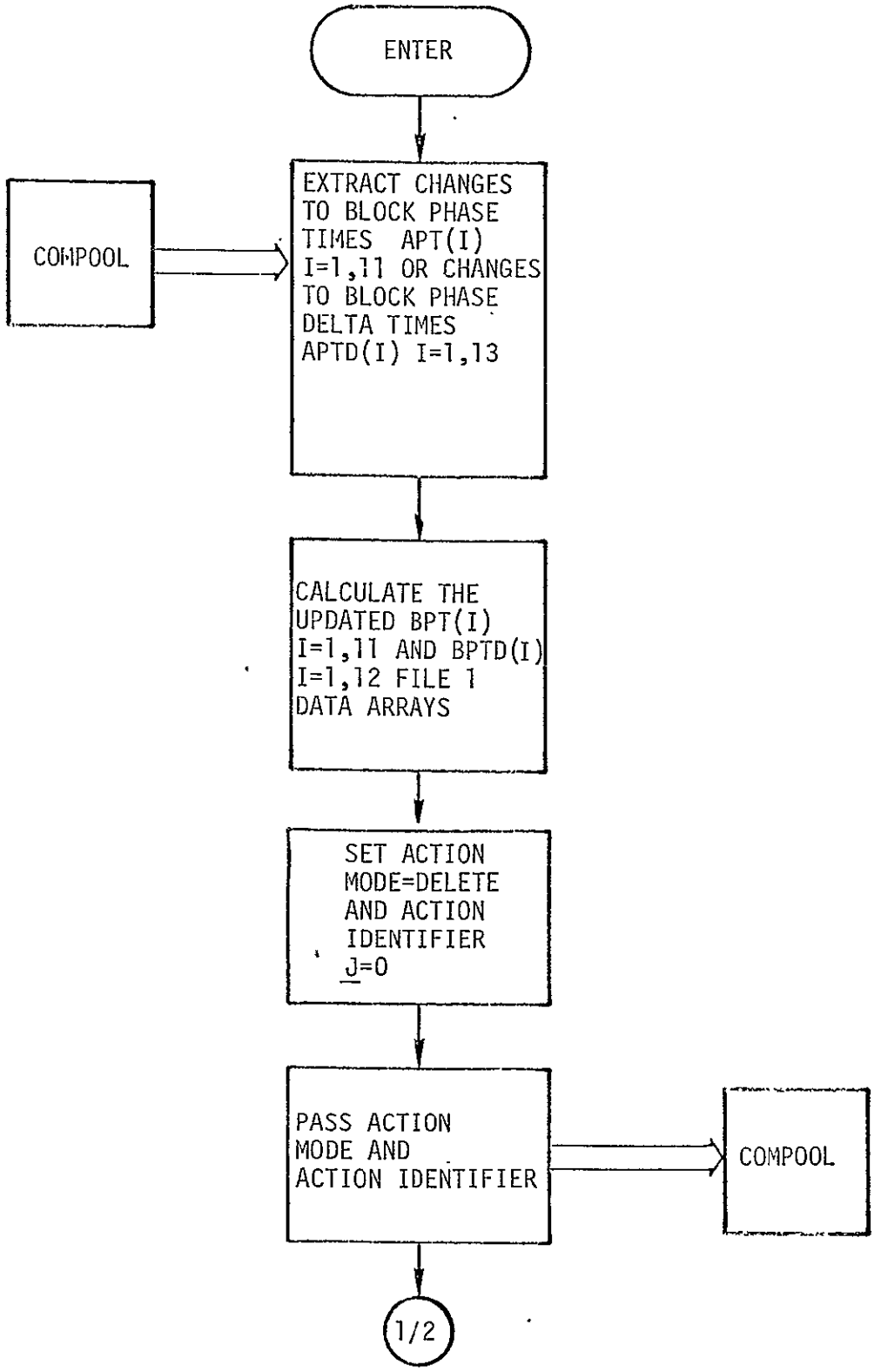


Figure 29. Flow Diagram for the FLIGHT Routine

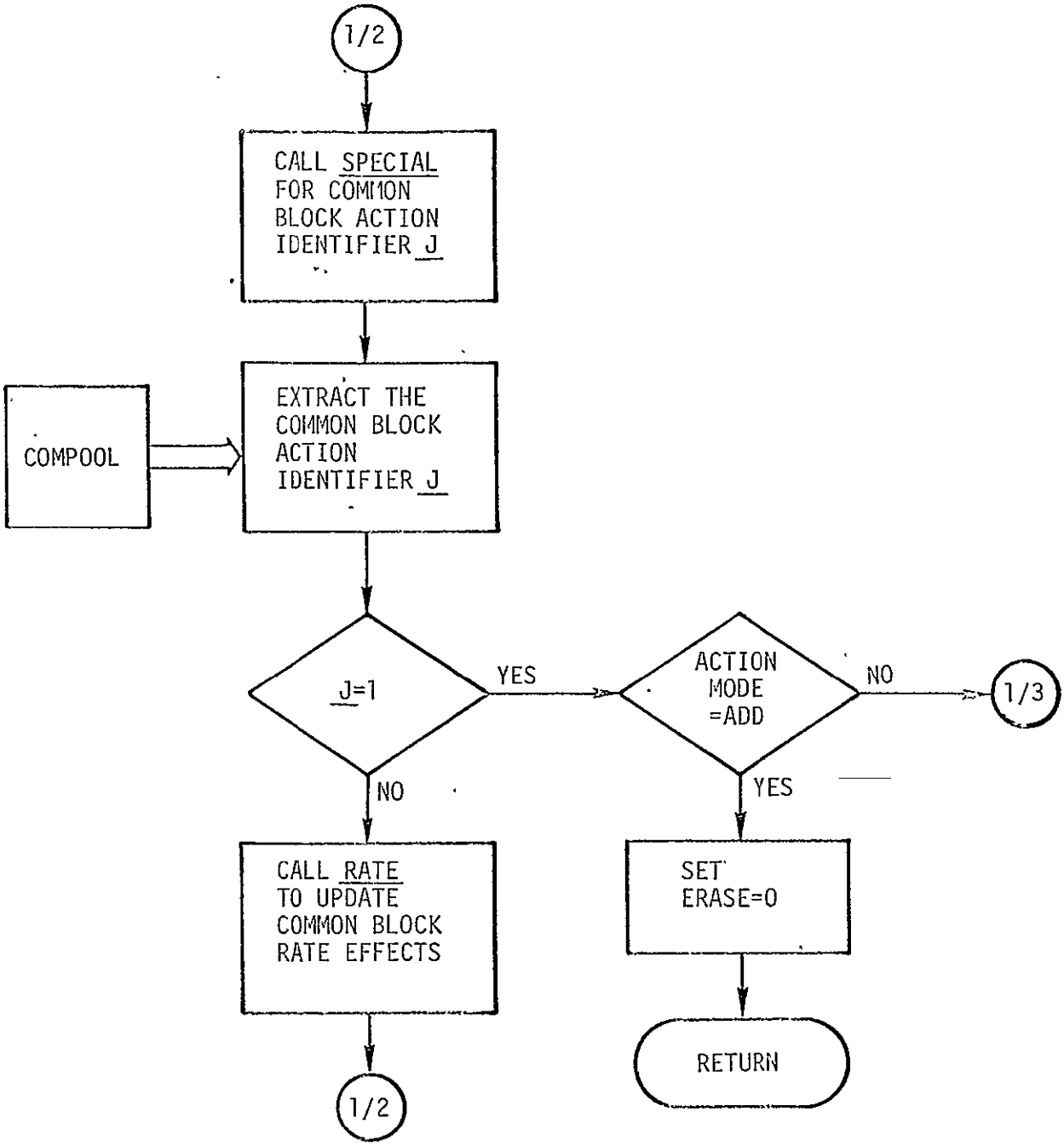


Figure 29. Continued

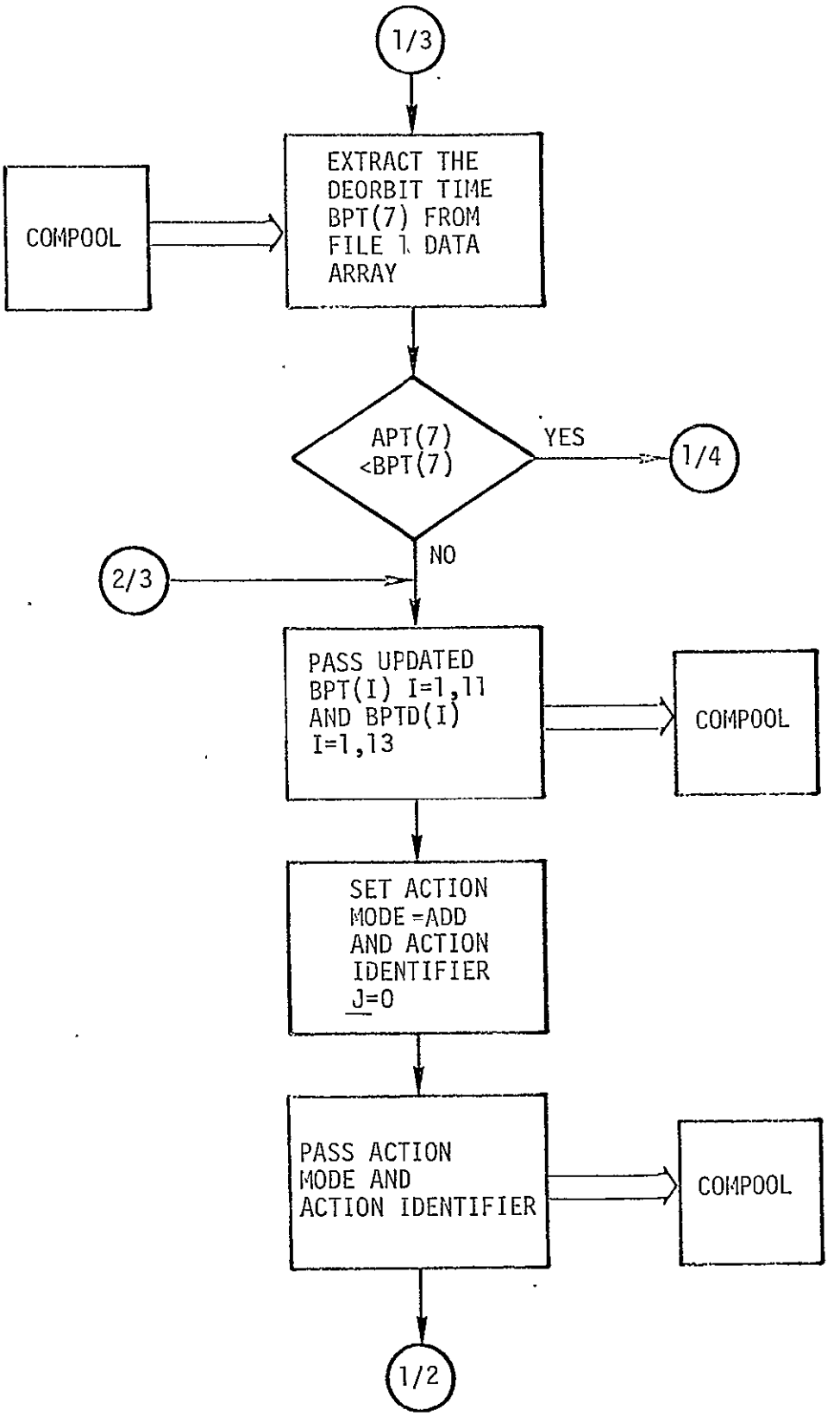


Figure 29. Continued

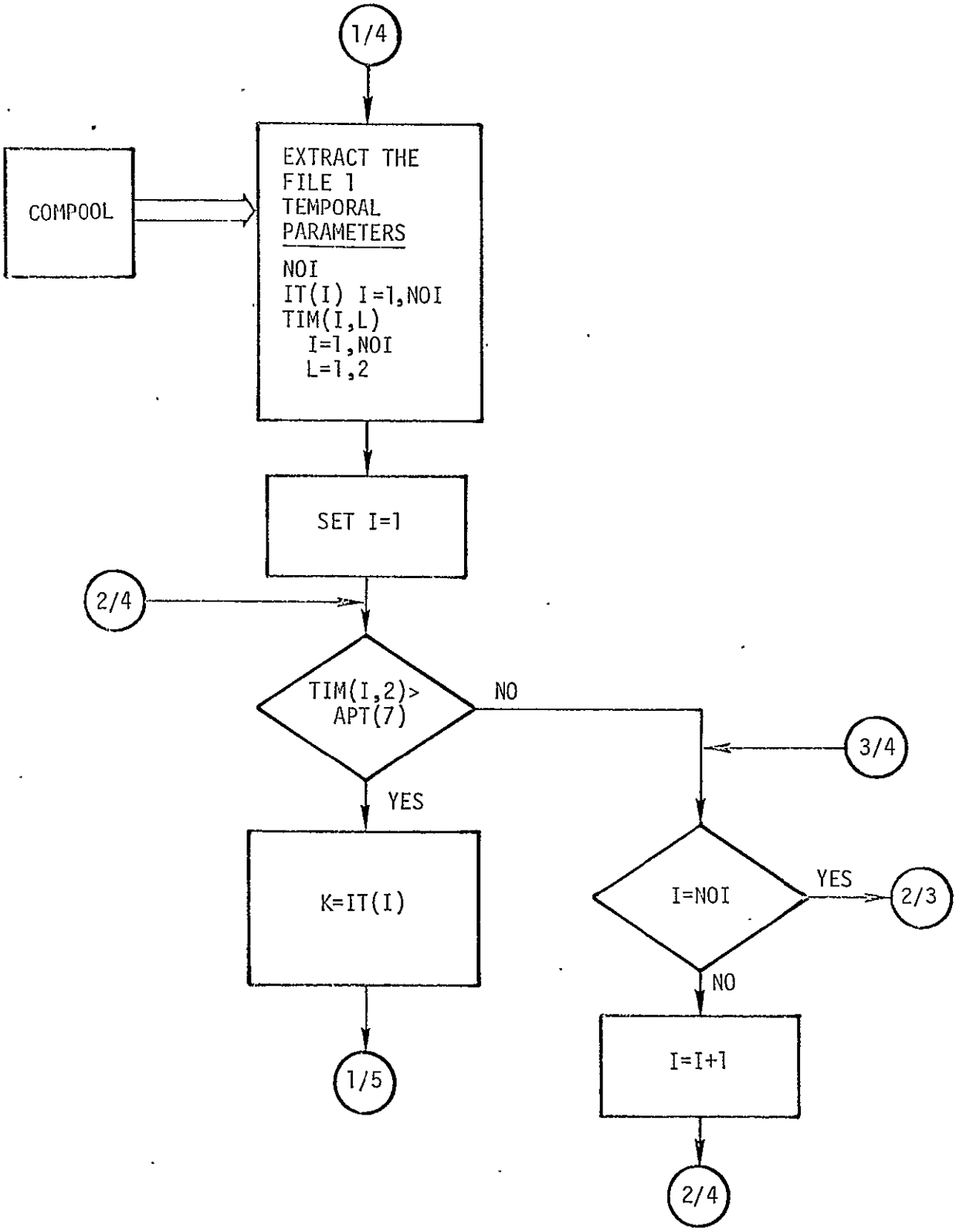


Figure 29. Continued

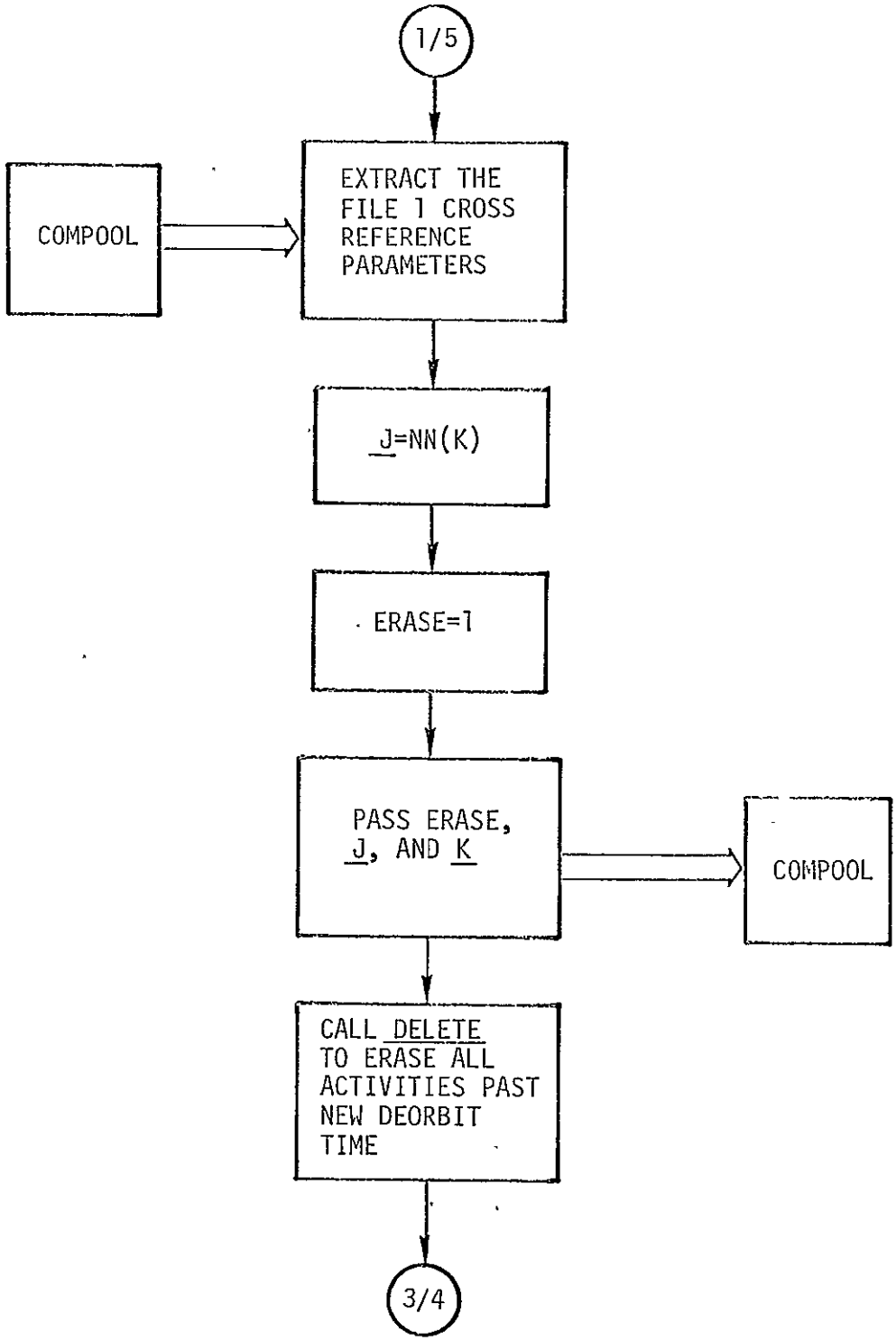


Figure 29. Concluded

5.18 INITIAL ROUTINE

Description - If the data option entered through the EXEC routine is set to Initial, the INITIAL routine will develop the File 1 data set for use by the Mission Planning Processor. The data set is constructed from the mission configuration data input through the terminal keyboard unit and the mission skeleton data from the Consumables Analysis Data Base.

Interface

I/O DEVICES - Terminal KEYBOARD unit.
DATA BASE - Consumables Analysis Data Base for input and COMPOOL for both input and output.
ROUTINES CALLING INITIAL - EXEC routine.
ROUTINES CALLED BY INITIAL - DISPLAY routine.

Internal Variables - None.

Input - The INITIAL routine requires the following data intput through the terminal KEYBOARD unit:

MISSION ID	The mission identifier
CONFG(I) I=1,11	The configuration data for the identified mission. (see Table III for definition of parameters.)

The INITIAL routine also requires the following input data accessed through the COMPOOL:

TABLE	The Active Mode Display Cross Reference Table (as defined in Table I).

Processing - The flow diagram of the INITIAL routine is presented in Figure 30.

Output - The INITIAL routine transmits the following data through the COMPOOL:

X	Display variable X = to the identifier for the configuration display (see Table I for ID values).

The INITIAL routine also transmits the mission configuration data restructured to the File 1 data set format through the COMPOOL. Table III defines the File 1 data set parameters.

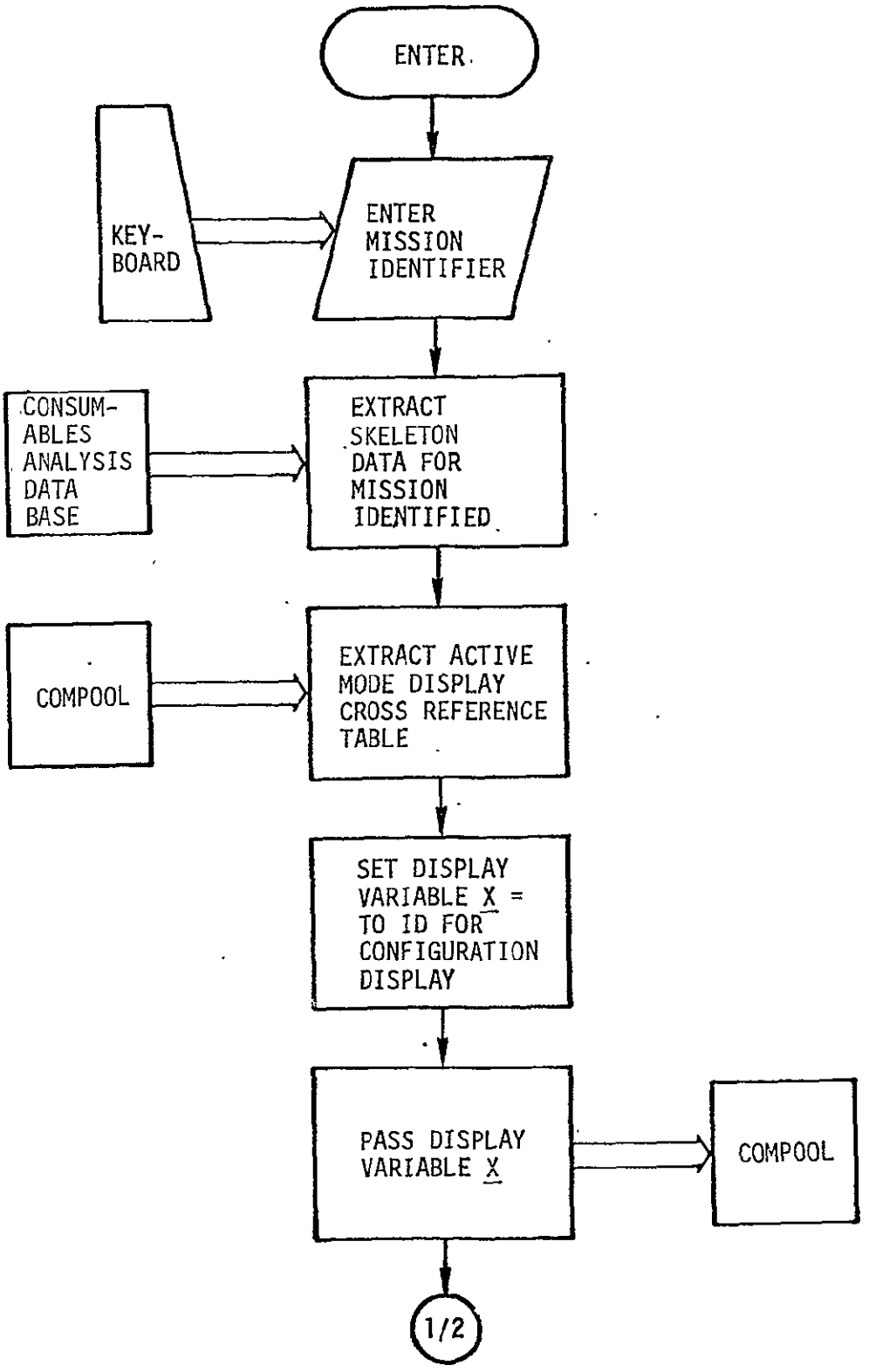


Figure 30. Flow Diagram for the INITIAL Routine

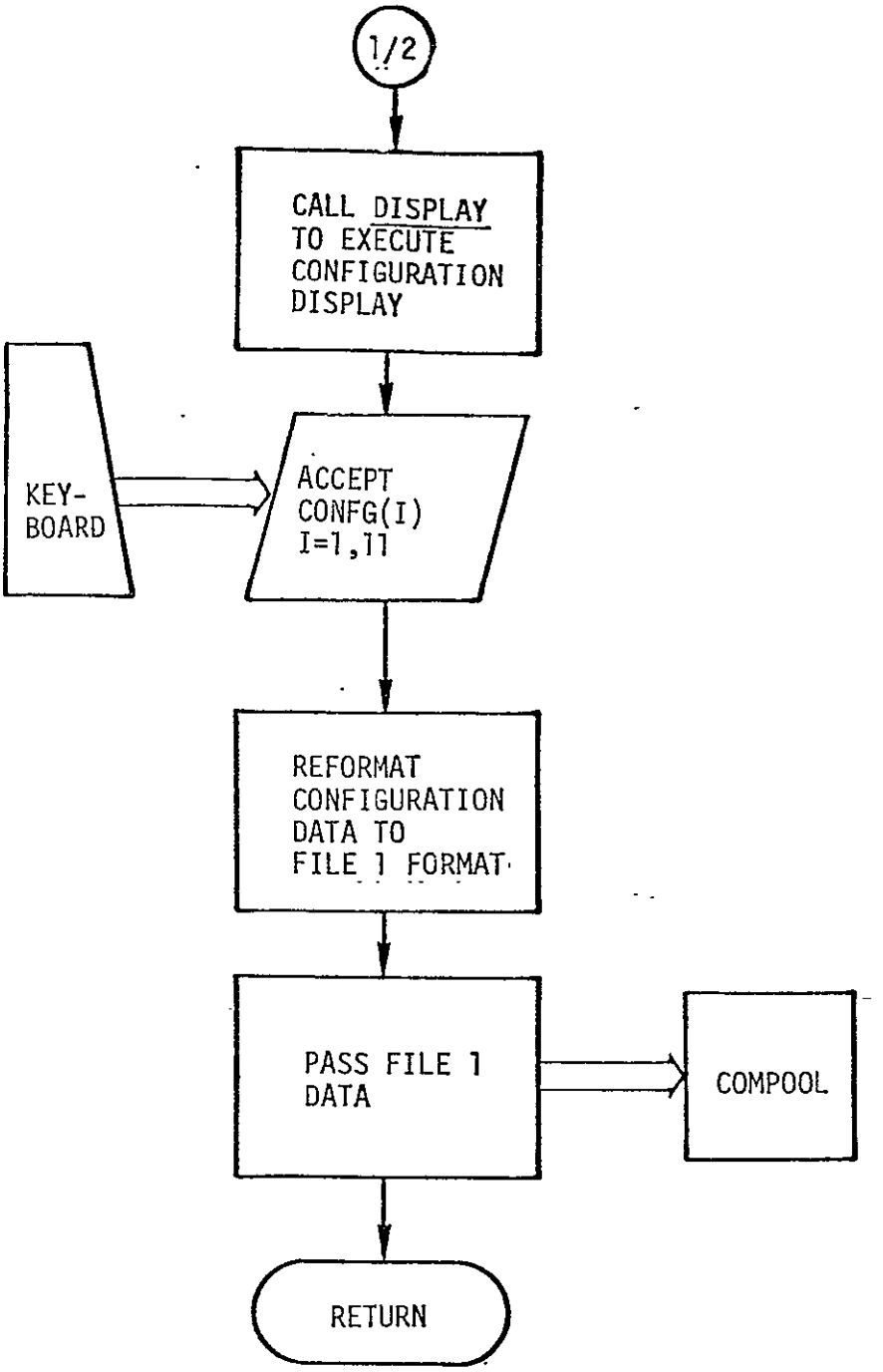


Figure 30. Concluded

5.19 IV INPUT ROUTINE

Description - When the Mission Planning Processor is run in the Event Mode, the IV INPUT routine permits the user to input the influence variables used to generate an event chart. The user has the option to input original influence variable values, use previously stored influence variable values from the File 0 data set, or modify the influence variable values entered.

Interface -

I/O DEVICES - Terminal KEYBOARD unit.
DATA BASE - COMPOOL for input and output.
ROUTINES CALLING IV INPUT - EXEC routine.
ROUTINES CALLED BY IV INPUT - FILE ZERO and DISPLAY routines.

Internal Variables - None.

Input - The IV INPUT routine requires the following data input through the terminal KEYBOARD unit:

MODE OPTION	IV INPUT operation mode; = INITIAL the influence variable values input through terminal KEYBOARD unit = EDIT the influence variable values input from the File 0 data set
VALUE(IV)	The influence variable value.

The IV INPUT routine also requires the following input data accessed through the COMPOOL:

TABLE	Event Mode Display Cross Reference Table (definition is TBD)
DATA(0)	The File 0 data set (parameters are defined in Volume IV).

Processing - The flow diagram of the IV INPUT routine is presented in Figure 31.

Output - The IV INPUT routine transmits the following data through the COMPOOL:

X	The display variable X = to the identifier for the influence variable display (see Table I for identifier)
DATA(0)	The File 0 data set (parameters are defined in Volume IV).

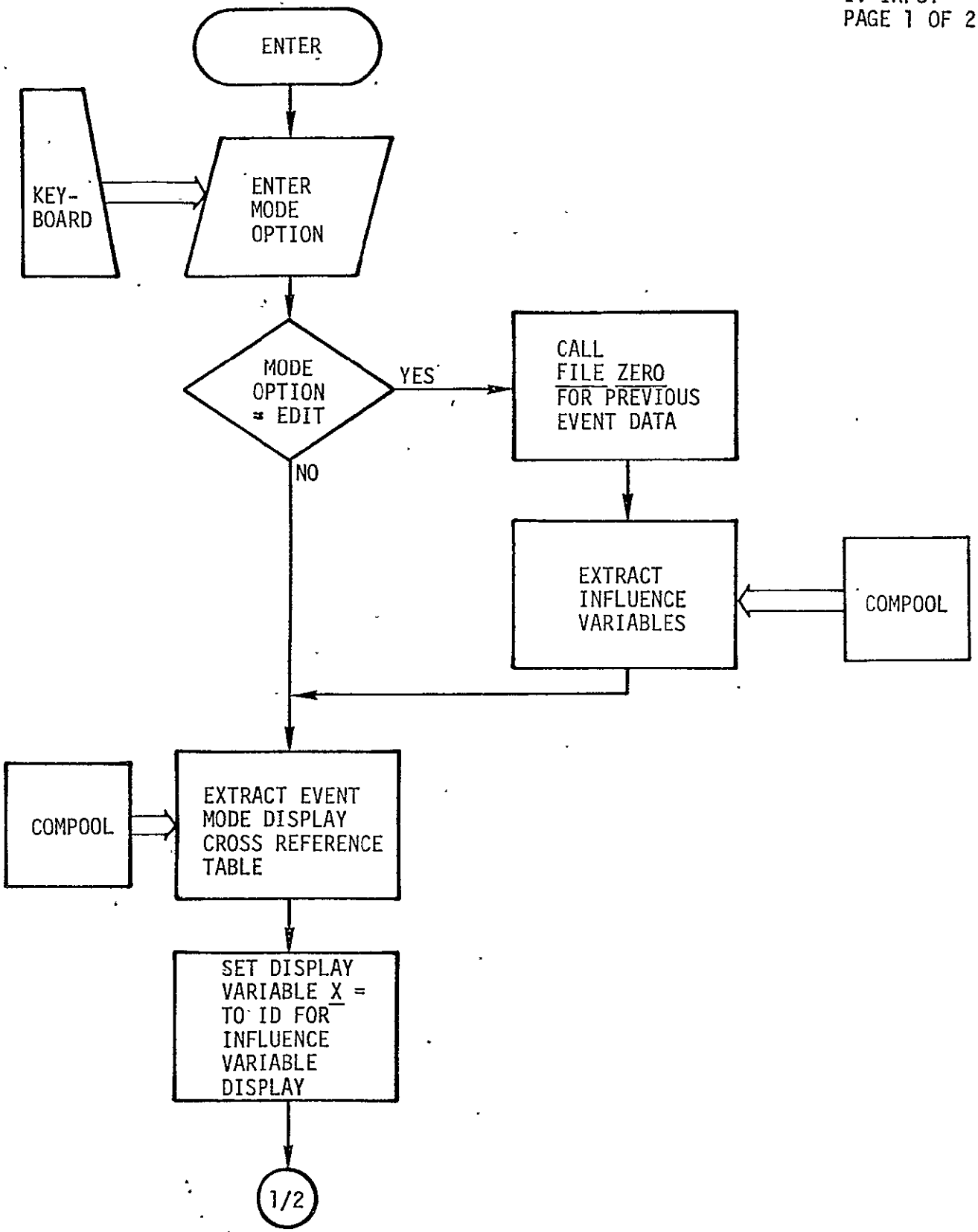


Figure 31. Flow Diagram for the IV INPUT Routine

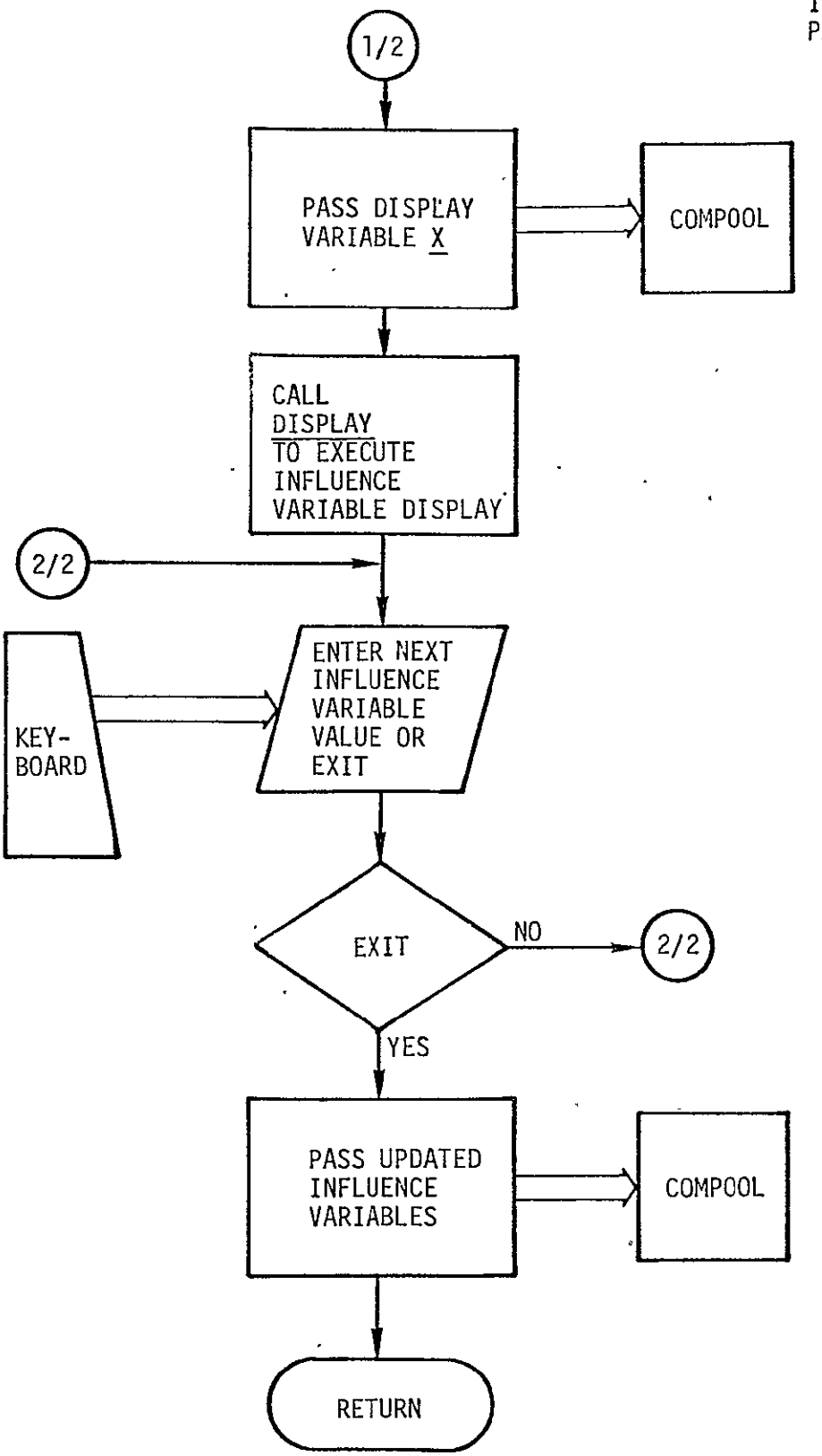


Figure 31. Concluded

5.20 LINECK ROUTINE

Description - The LINECK routine determines the progression through the user interface displays. The line number entered on the active display is queried to determine if that display is to remain active during the next Mission Planning Processor activity or if the next display in the progression is required.

Interface

I/O DEVICES - Terminal KEYBOARD unit.
DATA BASE - COMPOOL for both input and output.
ROUTINES CALLING LINECK - PLAN routine.
ROUTINES CALLED BY LINECK - none.

Internal Variables - None.

Input - The LINECK routine requires the following data input through the terminal KEYBOARD unit:

LINE #	The line number corresponding to the display selected on the display active on the CRT unit.

The LINECK routine requires the following input data accessed through the COMPOOL:

X	Display variable X = to the identifier of the active display (see Table I for specific display identifiers)
MAX(X)	The maximum lines on display X.

Processing - The flow diagram of the LINECK routine is presented in Figure 32.

Output - The LINECK routine transmits the following data through COMPOOL:

LINE #	The line number corresponding to the display selected on the display active on the CRT unit.
CHECK(X)	The display check flag set if the next display in the progression is required.

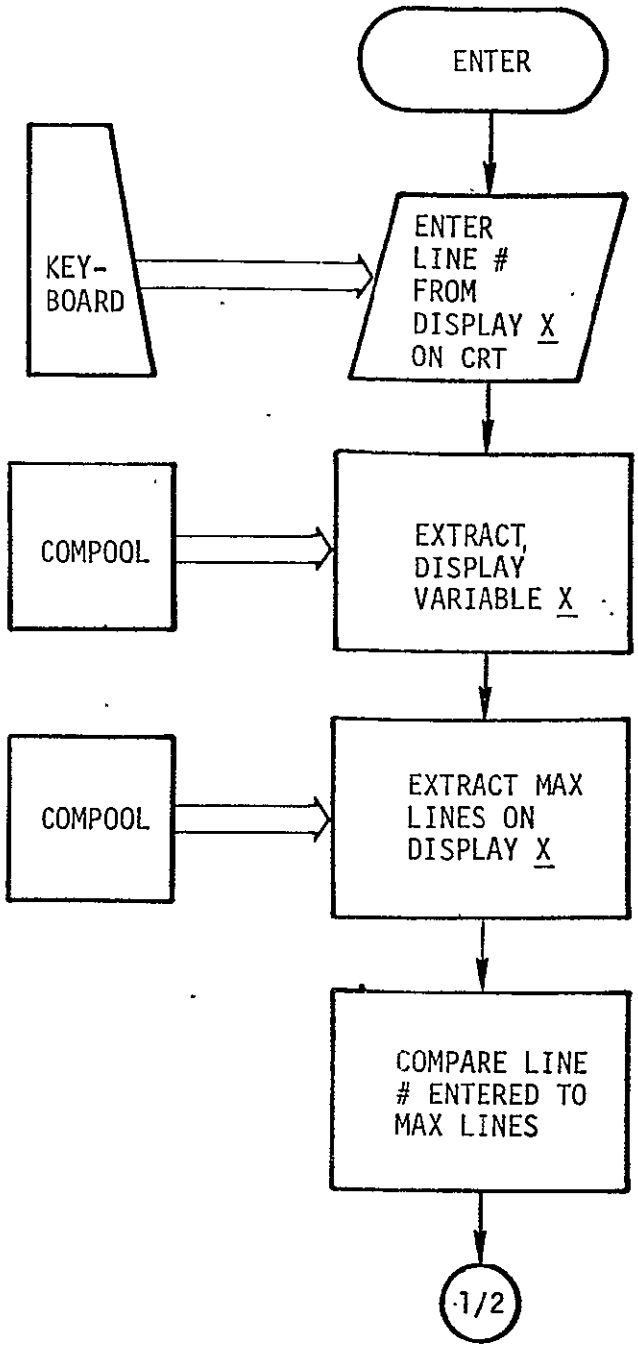


Figure 32. Flow Diagram for the LINECK Routine

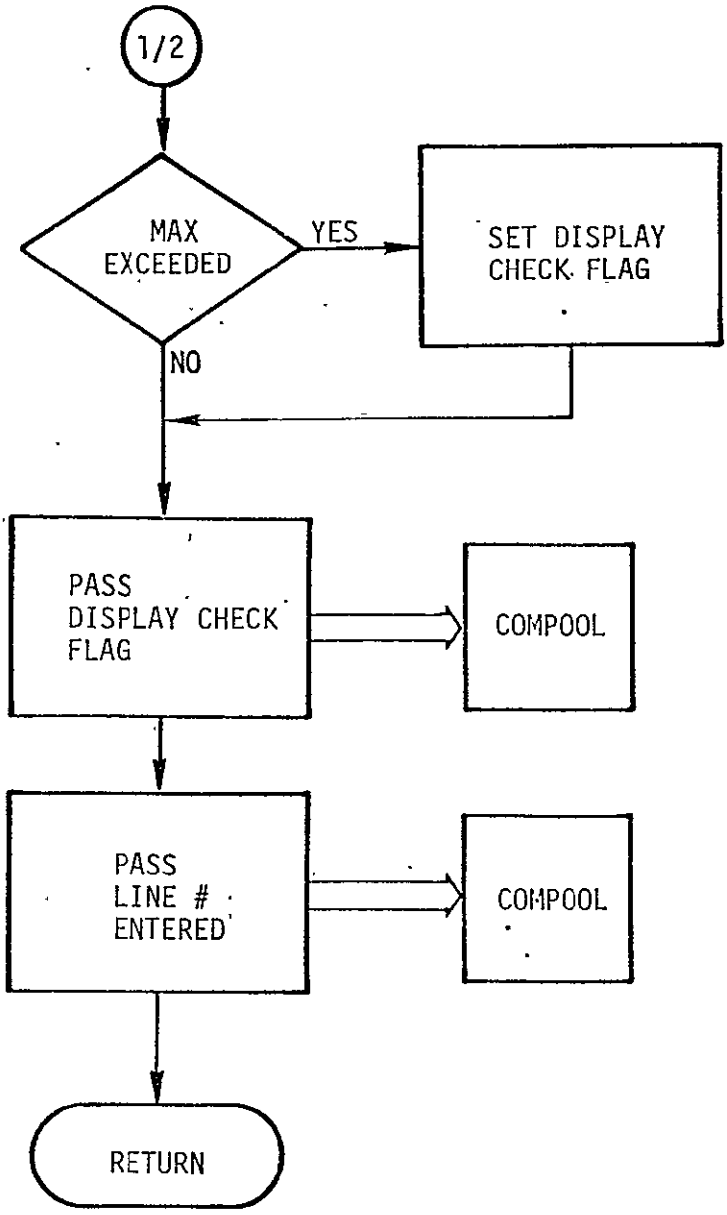


Figure 32. Concluded

5.21 OUTPUT ROUTINE

Description - The OUTPUT routine provides the user with the option to display and/or store selected data generated by the Mission Planning Processor. In the Event Mode, only the Event Chart displays can be executed. In the Active Mode, all Output displays, including the Event Chart displays, can be executed. The Output displays are listed in Table II. The user may select not to store any data. However, the user will be warned of that selection and asked for a confirmation to prevent inadvertent loss of data.

Interface

I/O DEVICES - The terminal KEYBOARD and CRT units.
DATA BASE - COMPOOL for both input and output.
ROUTINES CALLING OUTPUT - EXEC routine.
ROUTINES CALLED BY OUTPUT - EVENT CHART, TIMELINE, CONSUM HISTORY, CONSUM RESERVES, DISPLAY, and FILE STORE routines.

Internal Variables

W	Output variable W = to the display option entered through the terminal KEYBOARD unit.

Input - The OUTPUT routine requires the following data input through the terminal KEYBOARD unit:

DISPLAY OPTION	Display option to select data to be displayed; = 0 no display, return 1 no display, store data only 2 display the CONFLICT TABLE 3 display the EVENT CHART 4 display the TIMELINE 5 display the CONSUMABLES VS TIME 6 display the CONSUMABLES QUANTITIES
STORE OPTION	Store option to select data set to be stored; = 0 store nothing, return 1 store File 1 data set 2 store Files 1 and 2 data sets 3 store Files 1, 2, and 3 data sets 4 store File 0 data set

The OUTPUT routine also requires the following input data accessed through the COMPOOL:

RUN MODE	Mission Planning Processor operation mode; EVENT computer aided event chart generation only ACTIVE Interactive mission planning

TABLE	The output Display Cross Reference Table (as defined in Table II).

Processing - The flow diagram of the OUTPUT routine is presented in Figure 33.

Output - The OUTPUT routine presents the following warning displays through the terminal CRT unit:

EVENT CHART ONLY	Displayed if the RUN MODE = EVENT and a different output display is requested
NO FILE GENERATED	Displayed when STORE OPTION = 0 is entered the first time.

The OUTPUT routine also transmits the following data through the COMPOOL:

X	Display variable X = to the identifier of the requested output display (see Table I for specific identifiers).
Y	Store variable Y = to store option = 0 store nothing, return 1 store File 1 data set 2 store Files 1 and 2 data sets 3 store Files 1, 2, and 3 data sets 4 store File 0 data set

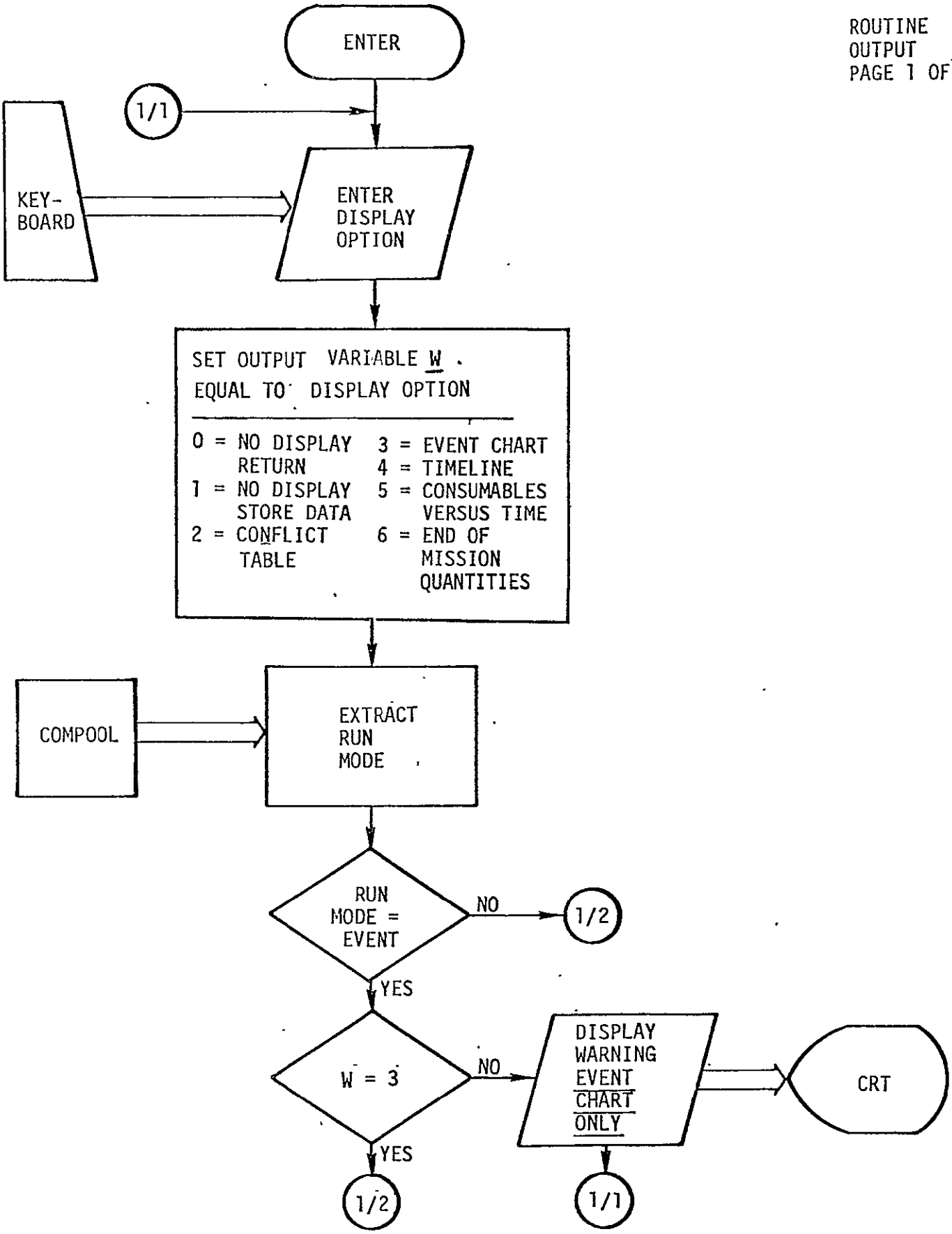


Figure 33. Flow Diagram for the OUTPUT Routine

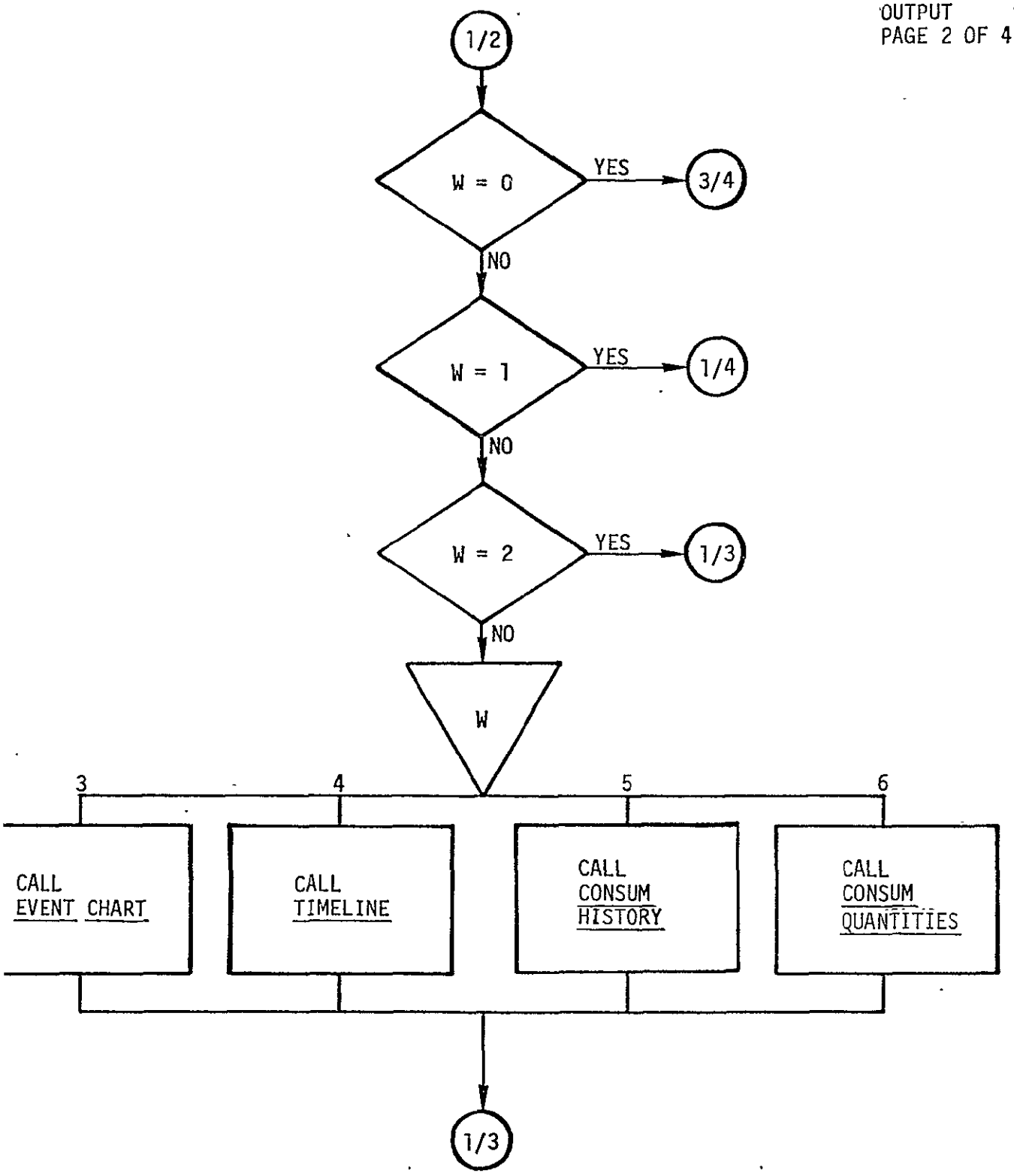


Figure 33. Continued

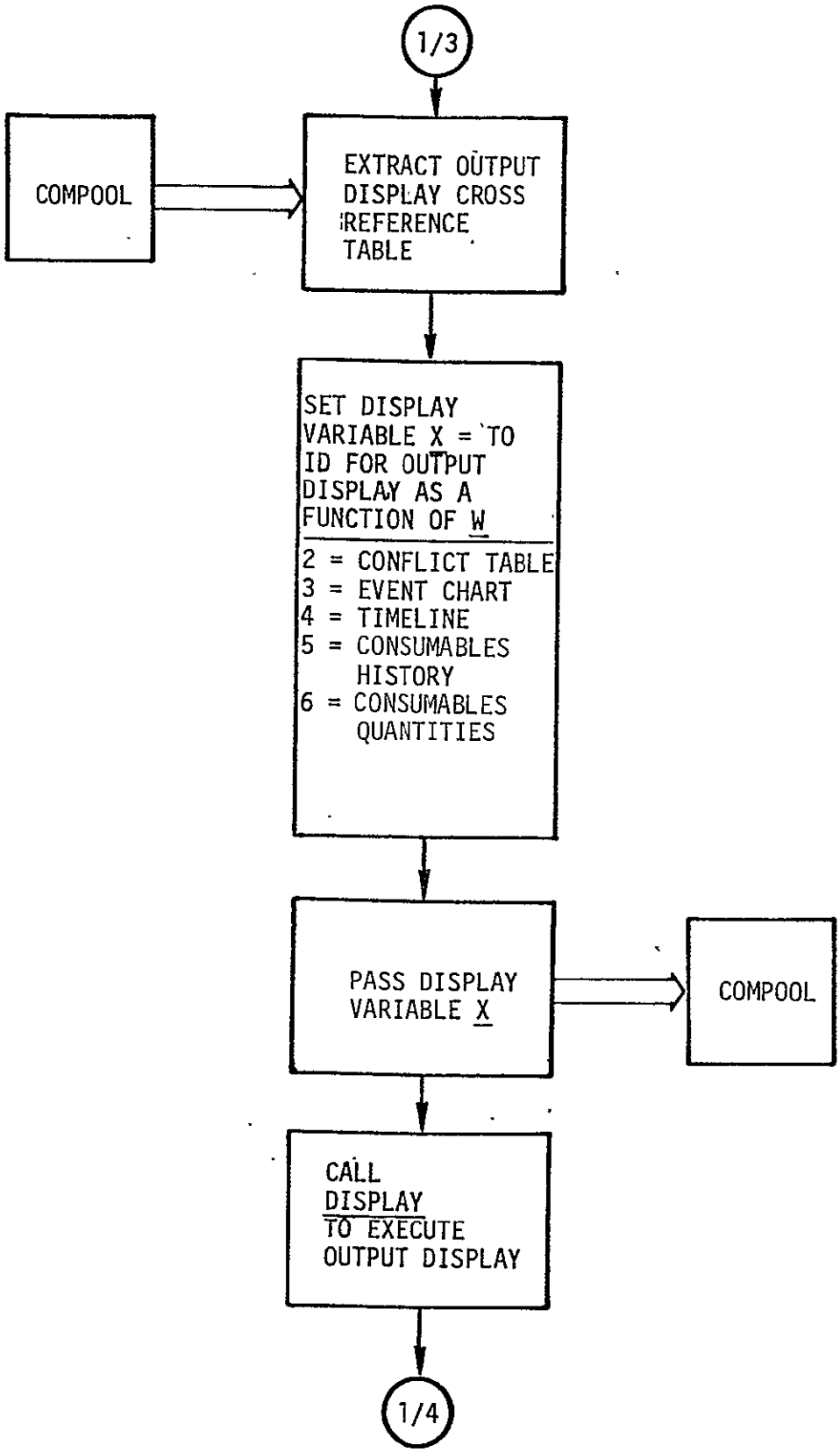


Figure 33. Continued

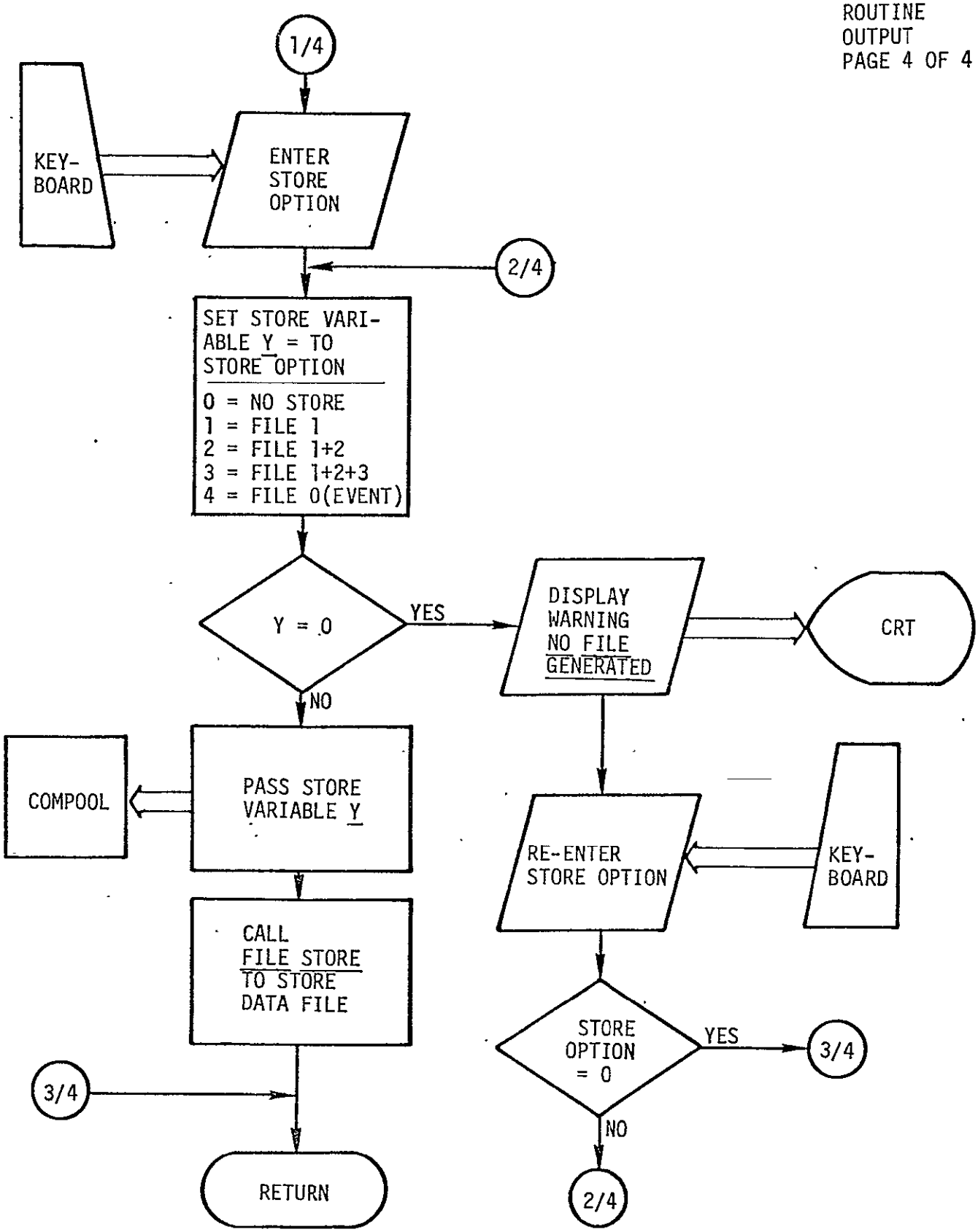


Figure 33. Concluded

5.22 PLAN ROUTINE

Description - The PLAN routine provides the Mission Planning Processor user with the capability to interactively schedule, modify, or unschedule consumable related mission events. The PLAN routine controls the progression through a series of Interactive Displays listed in Table I. Through terminal KEYBOARD unit entry to these displays, the user requests mission planning tasks. The PLAN routine acts as a middle manager calling on other control and support and computational routines to perform the tasks by manipulating the File 1 data set and immediately feeding back the results of the tasks to the user through the same Interactive Displays. The user will also be immediately informed if the results created any scheduling conflicts or consumable rate limit violations. Also, the conflicts and/or violations will be noted in CONFLICT TABLES for later assessment by the user.

Interface

I/O DEVICES - Terminal KEYBOARD unit.
DATA BASE - COMPOOL for both input and output; and the Consumables Analysis Data Base for input.
ROUTINES CALLING PLAN - EXEC routine.
ROUTINES CALLED BY PLAN - DISPLAY, LINECK, FLIGHT, ADD, and DELETE routines.

Internal Variables

CYCLE	If a display summary is requested, CYCLE is used as a counter to cycle through the particular set of displays requested for execution.
LIMIT	If a display summary is requested, LIMIT is set to the maximum number of displays in the particular set requested for execution.

Input - The PLAN routine requires the following data input through the terminal KEYBOARD unit:

PROCEED	If a display summary is requested, the PROCEED instruction will cause the next display in the requested set of displays to be executed.
LINE # = n	The line number corresponding to the itemized event on the ACTION display. The value determines the mission planning task requested; = n modify the scheduled event -n unschedule the scheduled event n+1 schedule the event n+2 return to the MENU display.
.DATA	The specific data set required by the ACTION display to perform the tasks requested (specific parameters are a function of the ACTION display being processed as defined in Table IV).

The PLAN routine requires the following input data accessed through the COMPOOL:

CHECK(X)	The display check flag set if the next display in the progression is required.
LINE #	The line number corresponding to the next selected display in the progression.
MAX(X)	The maximum lines on display X.
Z	The display variable Z = to the identifier of the MENU display being processed (see Table I for ID values).

The PLAN routine requires the following input data from the Consumables Analysis Data Base:

TABLE	The Active Mode Display Cross Reference Table (as defined in Table I).

Processing - The flow diagram of the PLAN routine is presented in Figure 34.

Output - The PLAN routine transmits the following data through the COMPOOL:

X	The display variable X = to the identifier of the ACTION display being executed (see Table I for ID values).

MOD(X)	Mod flag for display X
FLAG(X)	Display skeleton flag for display X
Z	The display variable Z = to the identifier of the MENU display being processed (see Table I for ID values).
Line # = n	The line number corresponding to the itemized event on the ACTION display. The value determines the mission planning task requested; = n modify the scheduled event -n unschedule the scheduled event n+1 schedule the event n+2 return to the MENU display.
DATA	The specific data set required by the ACTION display to perform the task requested (specific parameters are a function of the ACTION display being processed as defined in Table IV).

Table IV. Input Parameters Required to Schedule/Unschedule an Activity

ACTION ID	ACTIVITY	INPUT PARAMETERS
1	OMS MANEUVER	REF START, ΔV
2	RCS TRANSLATION	REF START, ΔV
3	ATTITUDE HOLD	REF START, REF STOP, ALT, TYPE HOLD
4	RENDEZVOUS	REF START, REF STOP, ΔV
5	STATION KEEPING	REF START, REF STOP
6	DOCK	REF STOP, ΔV
7	UNDOCK	REF START, ΔV
8	PTC	REF START, REF STOP, DUTY, PERIOD
9	EVA	REF START, REF STOP, NUMBER OF CREW
10	IVA	REF START, REF STOP, NUMBER OF CREW
11	MANIPULATOR OPS	REF START, REF STOP
12	IMU ALIGNMENT	REF START, REF STOP, DUTY, PERIOD
13	PAYLOAD BAY DOORS	REF START, REF STOP
14	PAYLOAD CONSUMABLES	REF START, REF STOP, DUTY, PERIOD, AND PAYLOAD related data (see Sec. 5.26)
15	COMPUTER	REF START, REF STOP, DUTY, PERIOD
16	TV	REF START, REF STOP, DUTY, PERIOD
17	DOWNLINK	REF START, REF STOP
18	UPLINK	REF START, REF STOP
19	FUEL CELL PURGE	REF START, REF STOP, DUTY, PERIOD
20	EAT	REF START, REF STOP, DUTY, PERIOD, NUMBER OF CREW
21	SLEEP	REF START, REF STOP, DUTY, PERIOD, NUMBER OF CREW
22	WASTE MANAGEMENT	REF START, REF STOP, DUTY, PERIOD
23	APU CHECKOUT	REF START, REF STOP

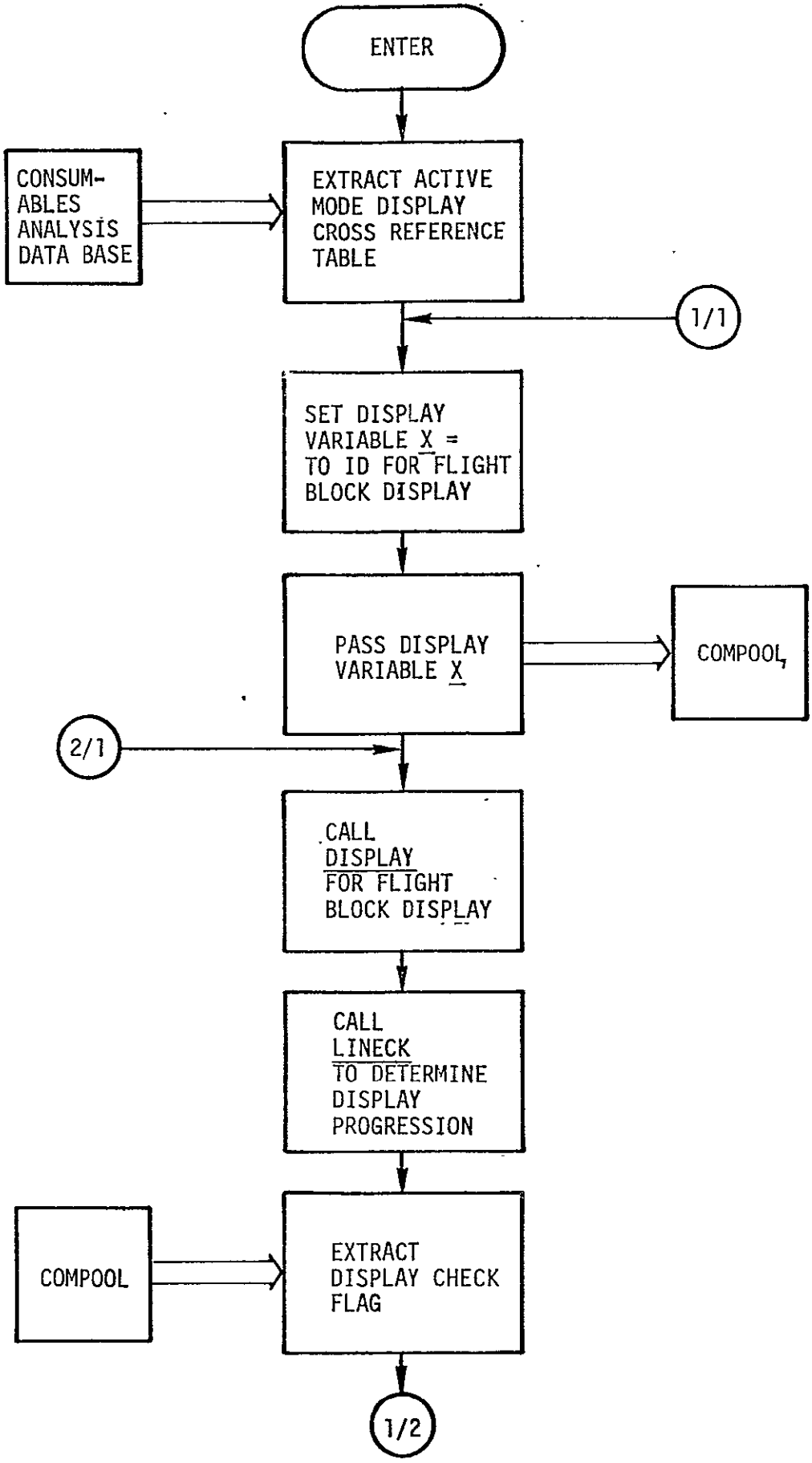


Figure 34. Flow Diagram for the PLAN Routine

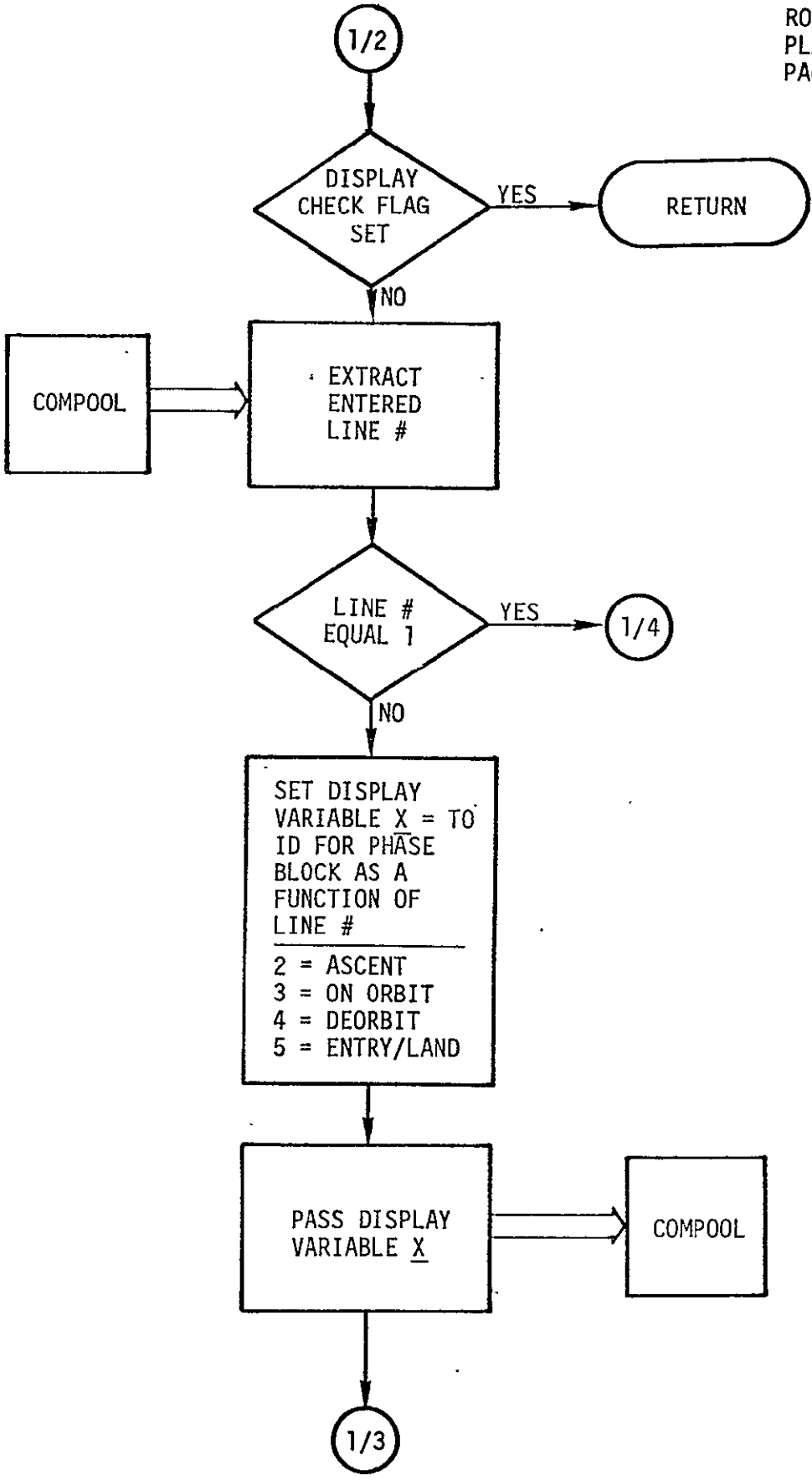


Figure 34. Continued

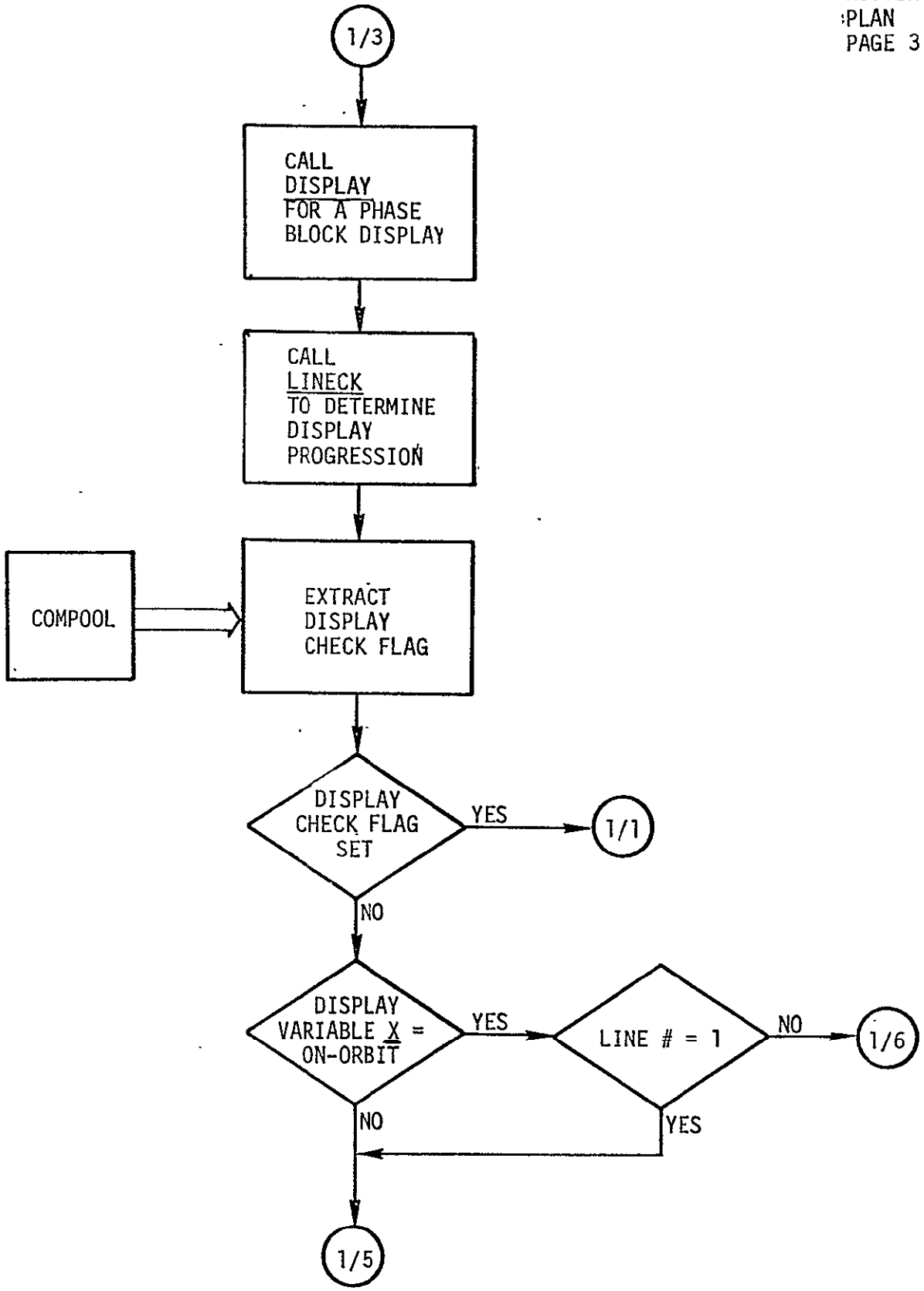


Figure 34. Continued

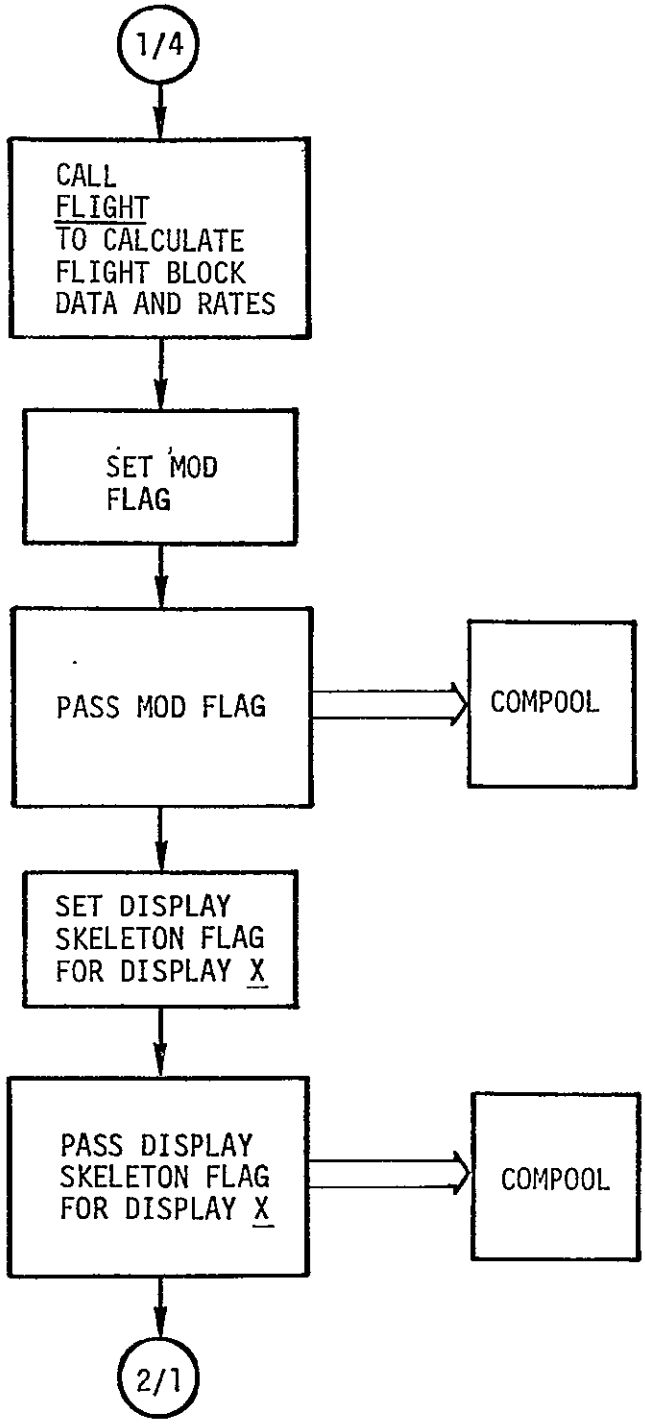


Figure 34. Continued

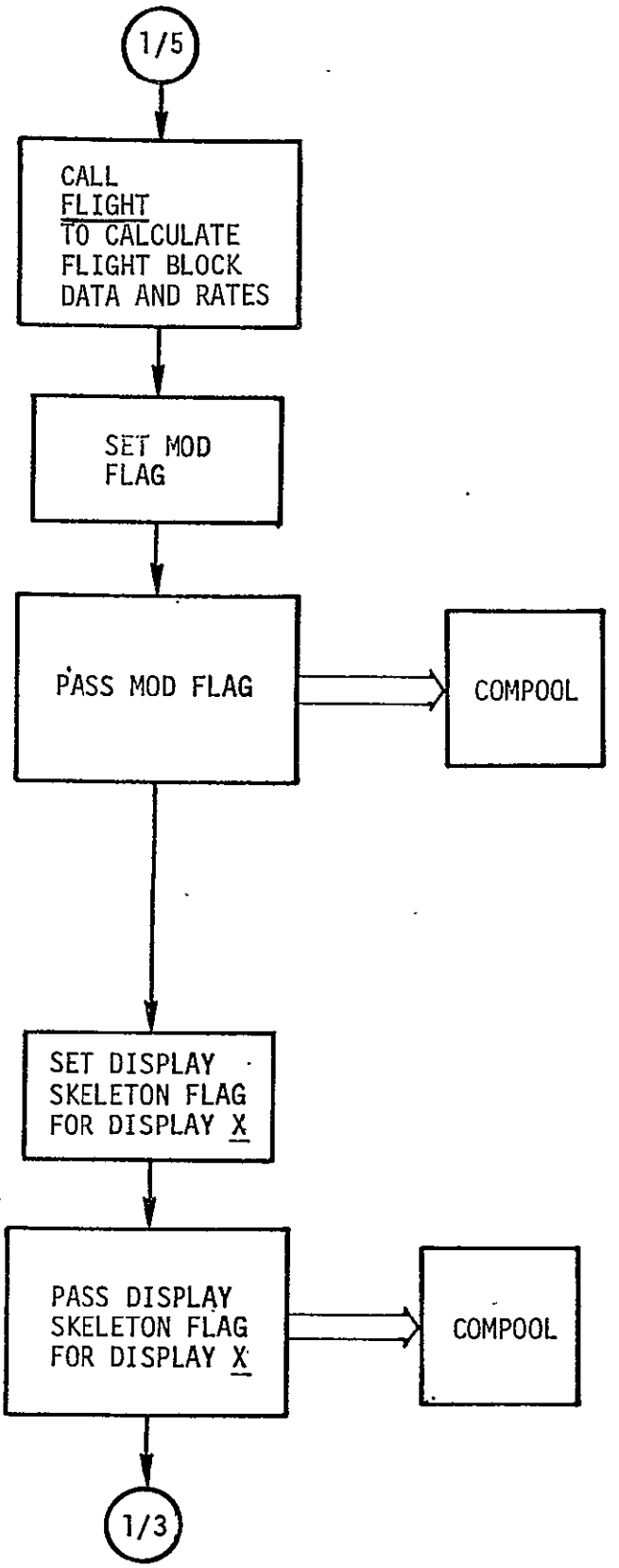


Figure 34. Continued

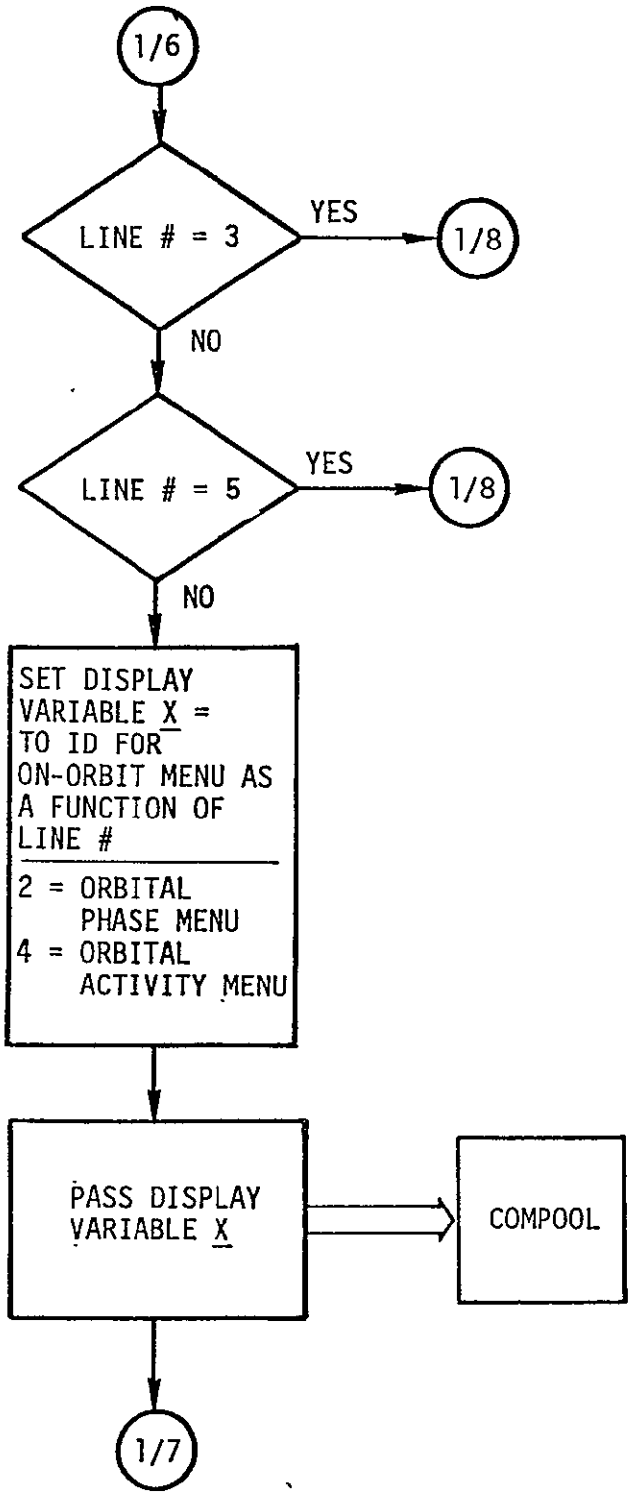


Figure 34. Continued

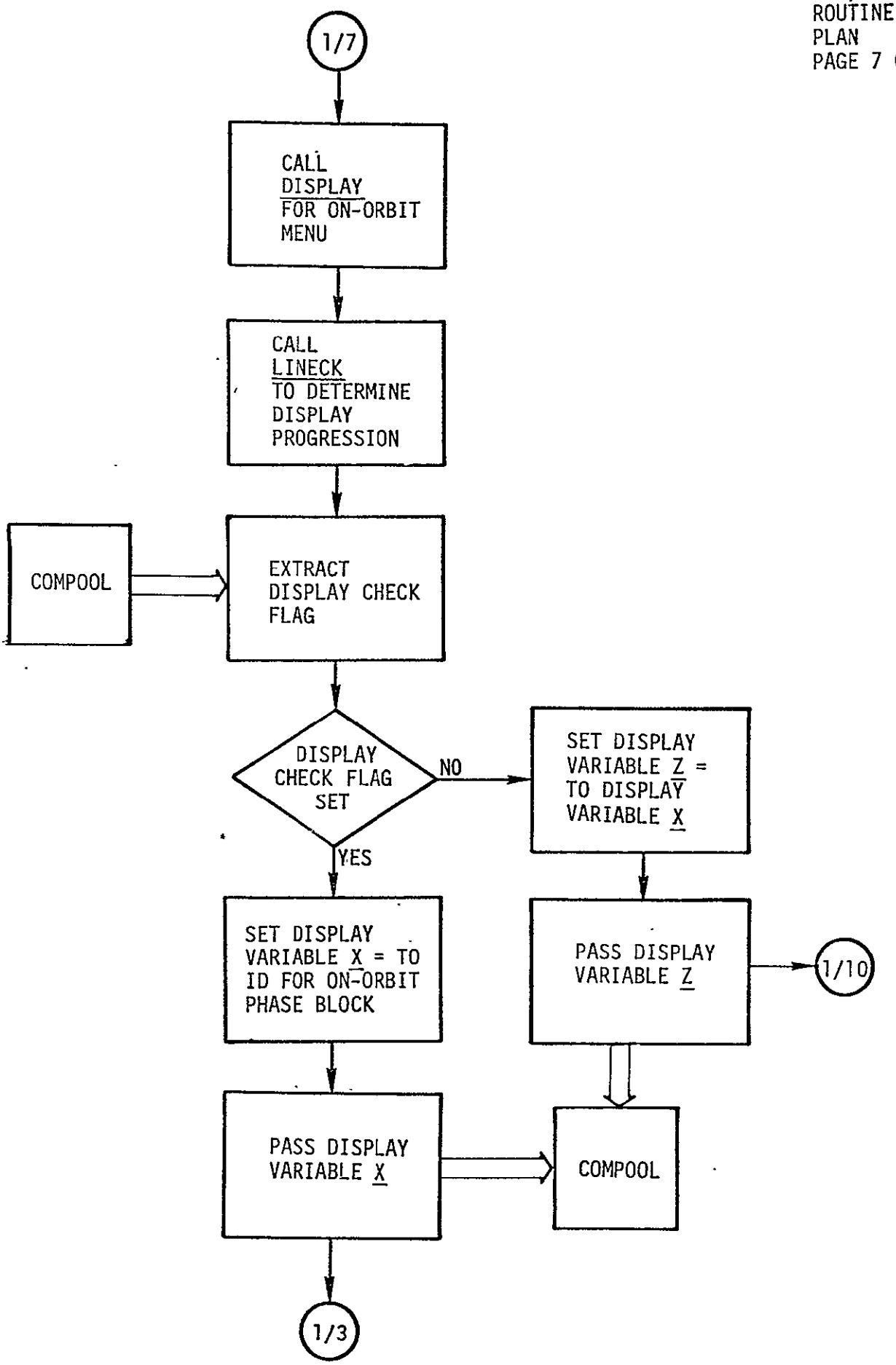


Figure 34. Continued

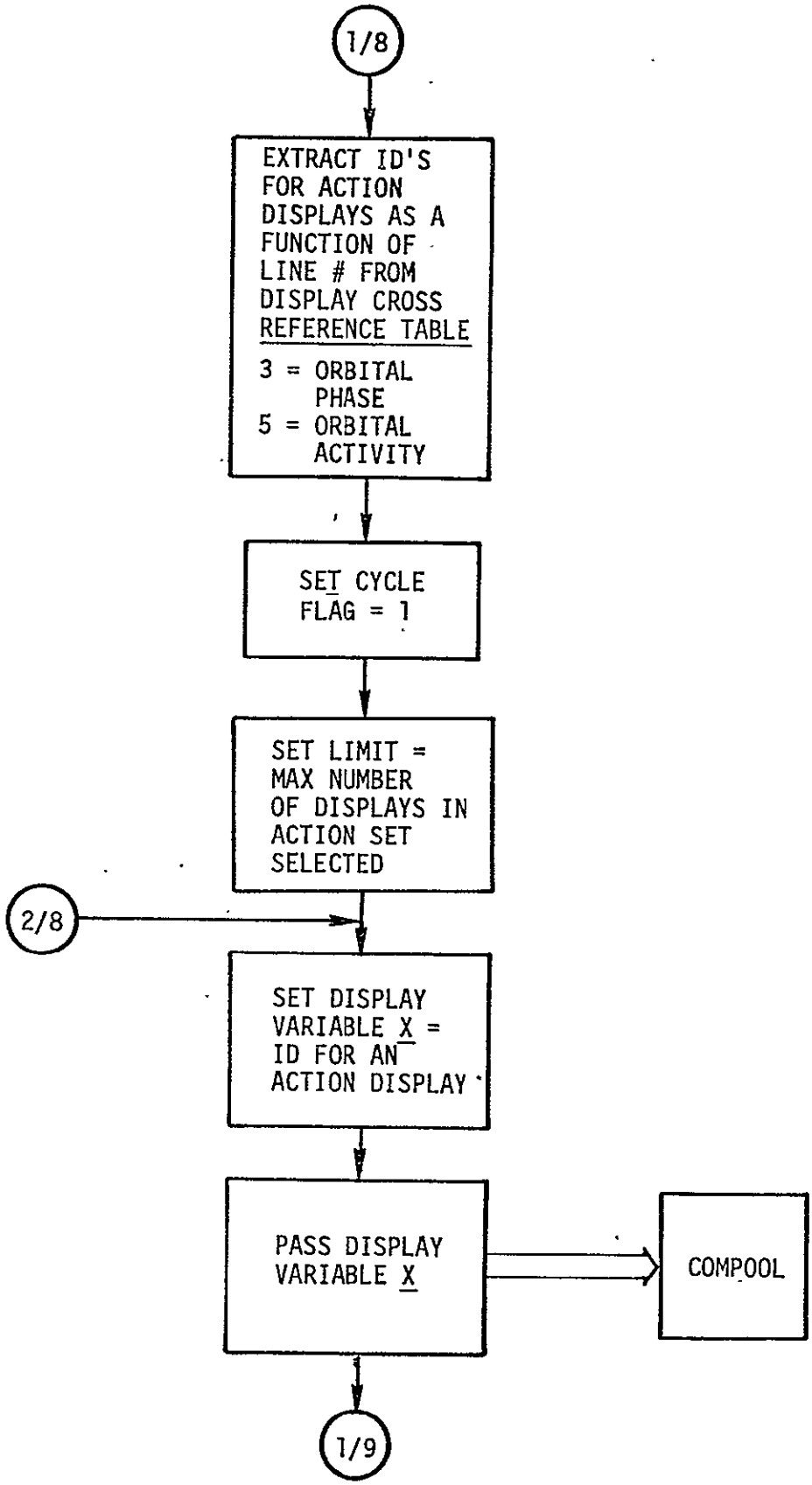


Figure 34. Continued

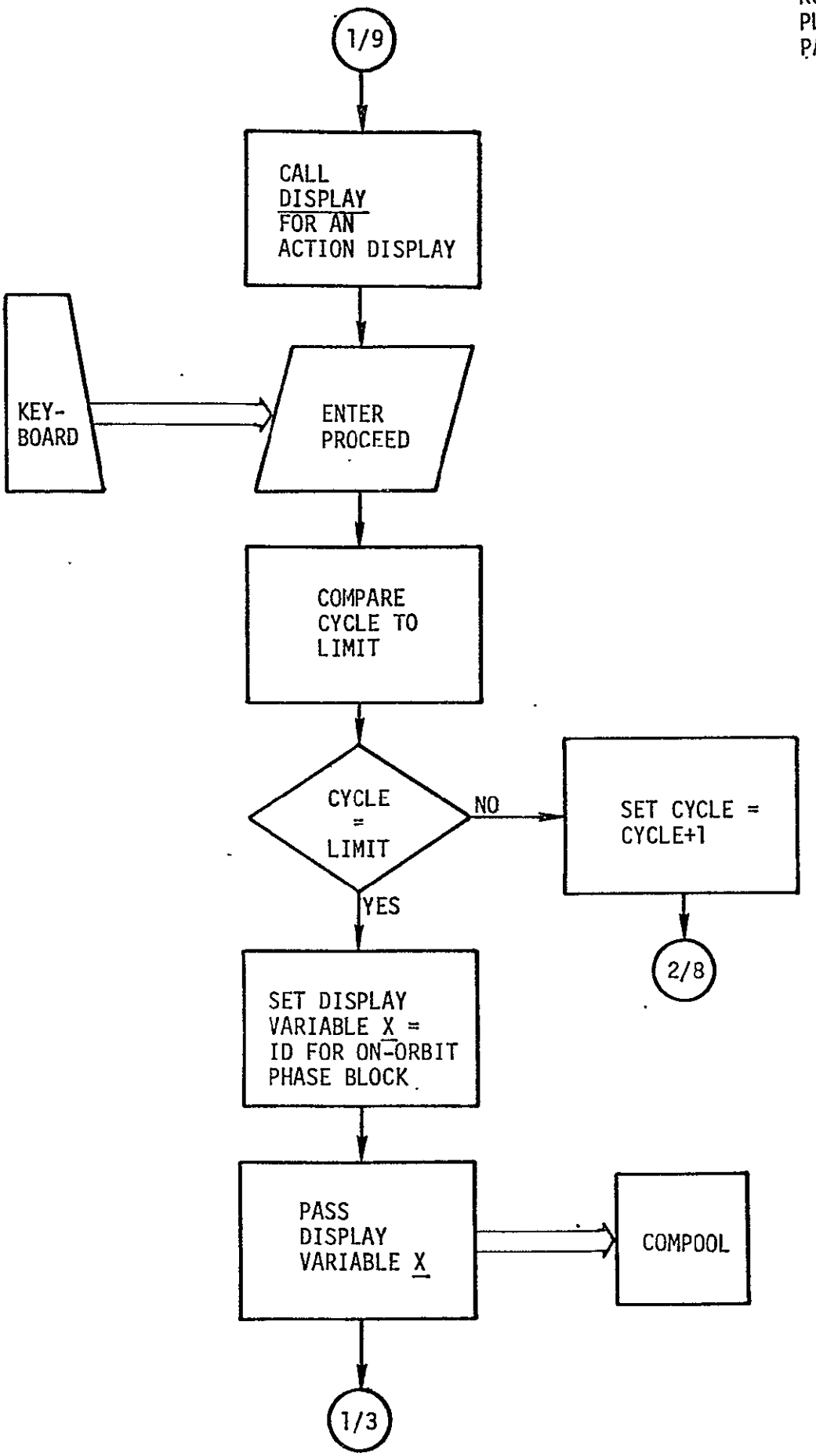


Figure 34. Continued

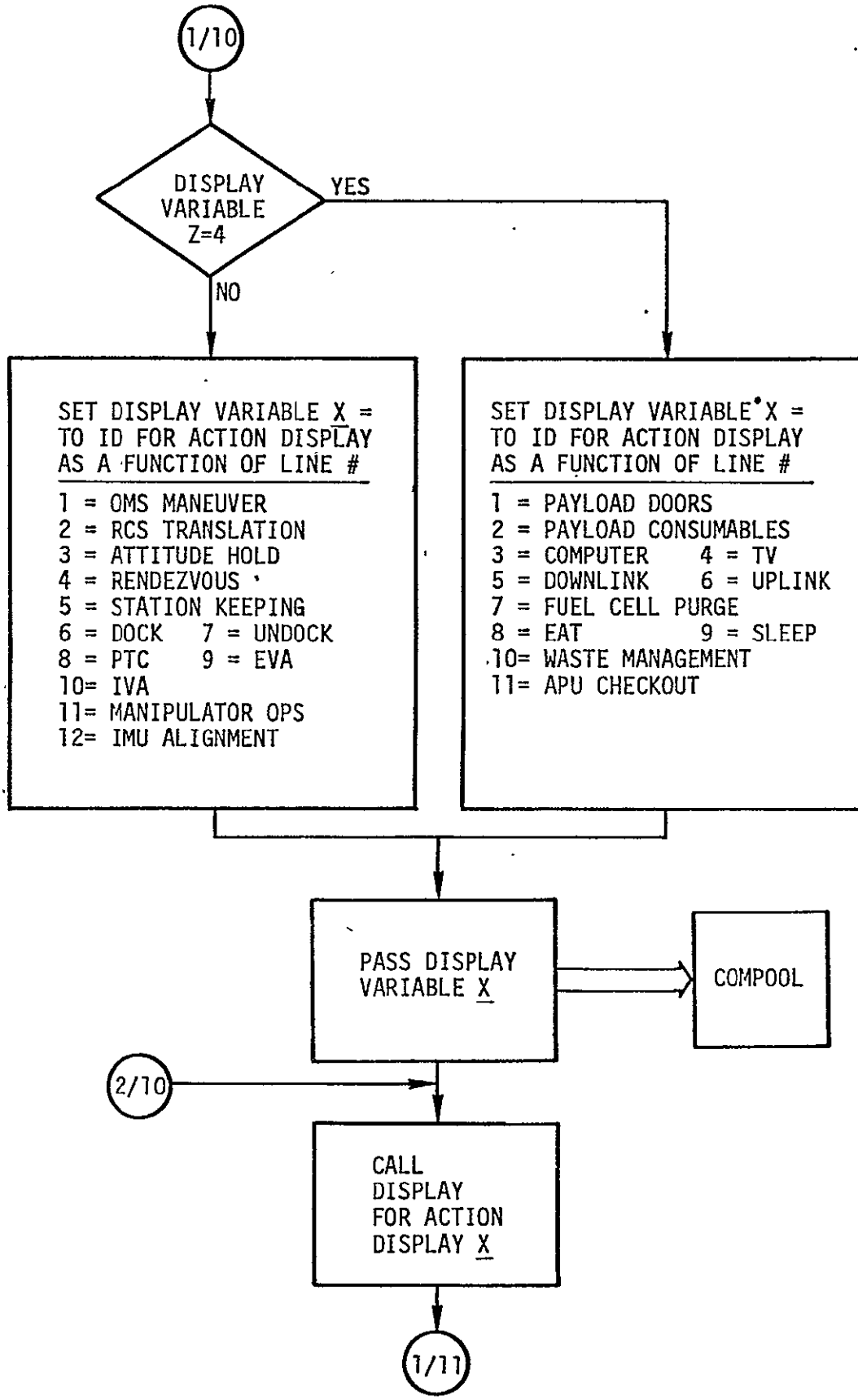


Figure 34. Continued

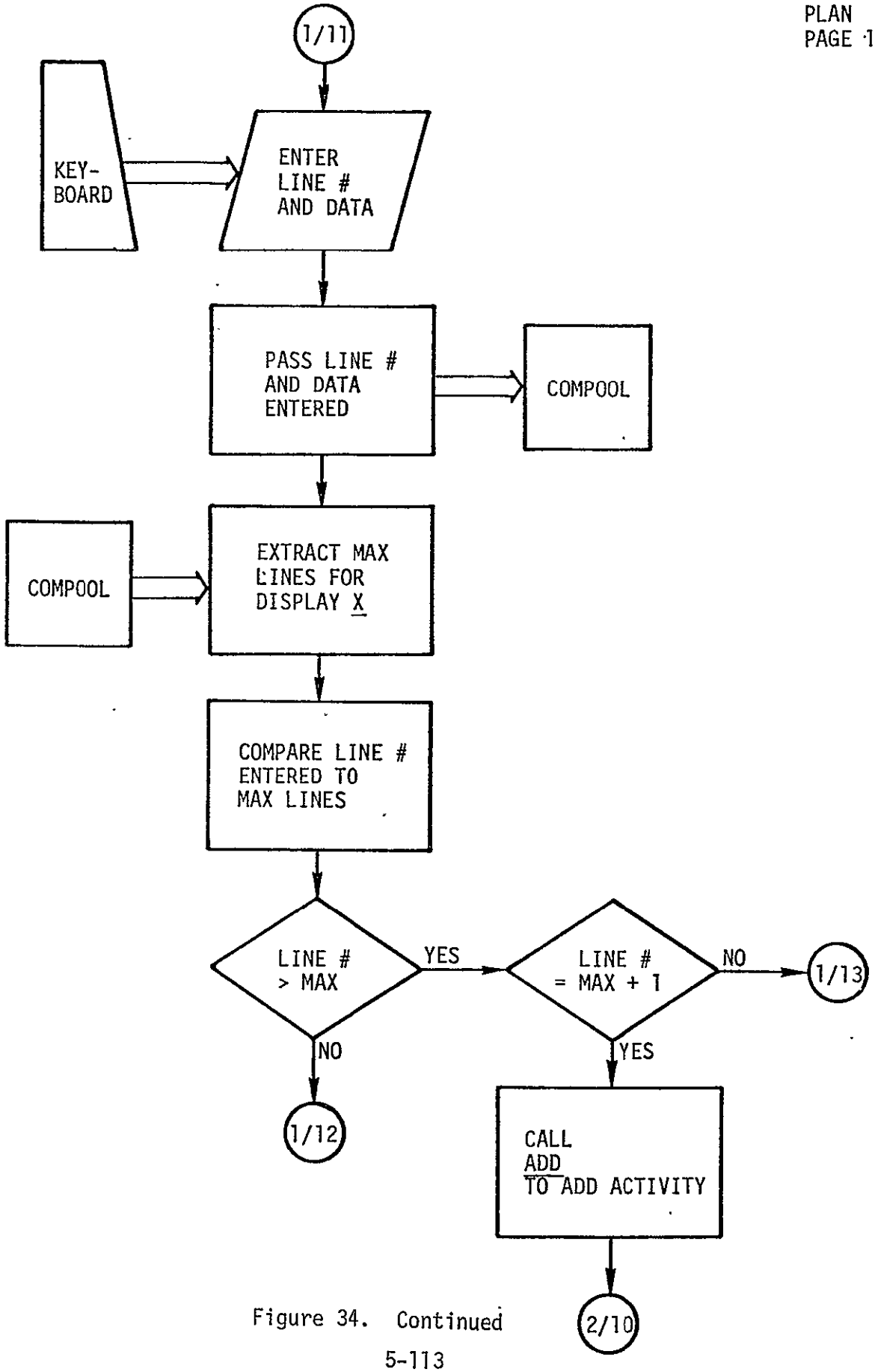


Figure 34. Continued

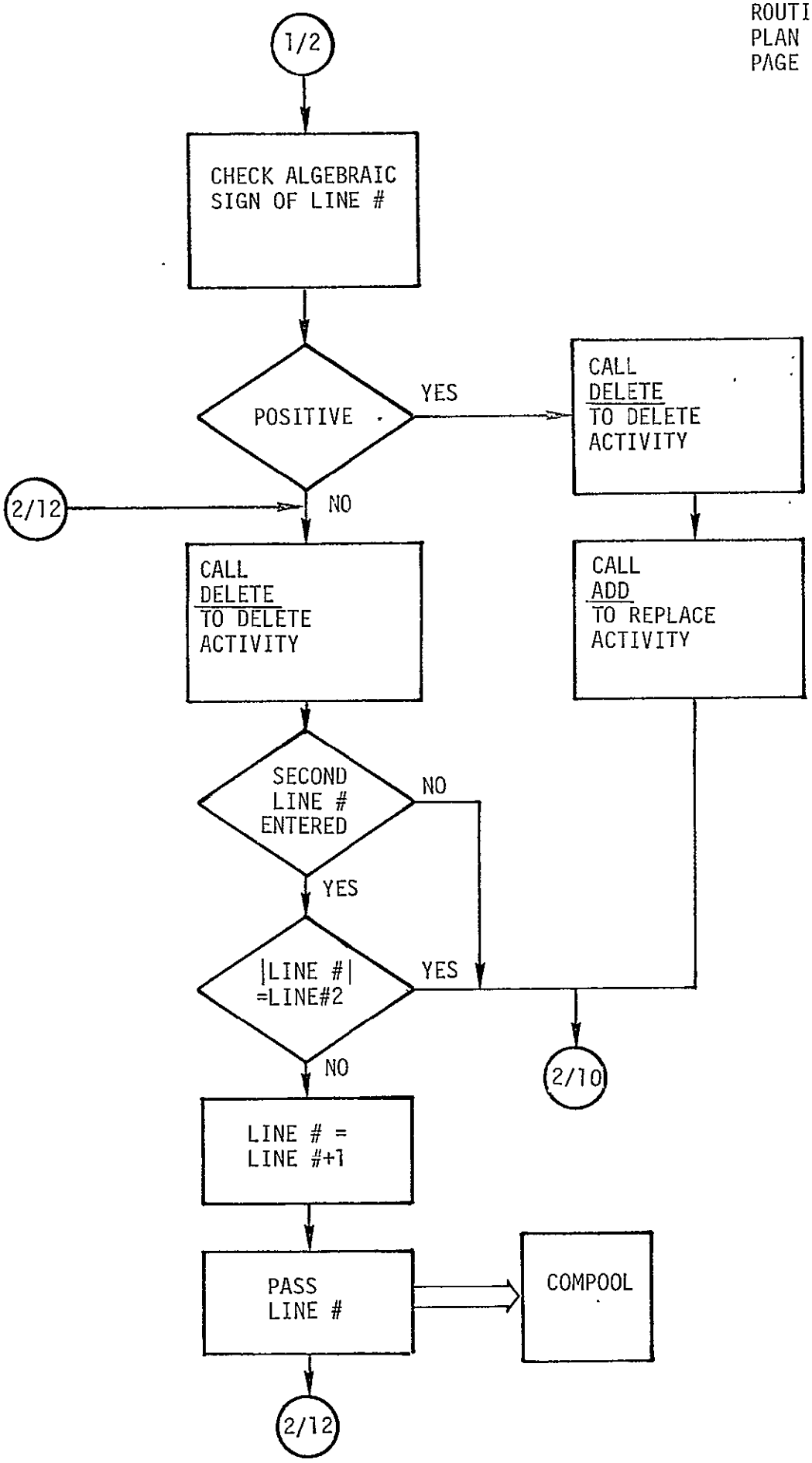


Figure 34. Continued

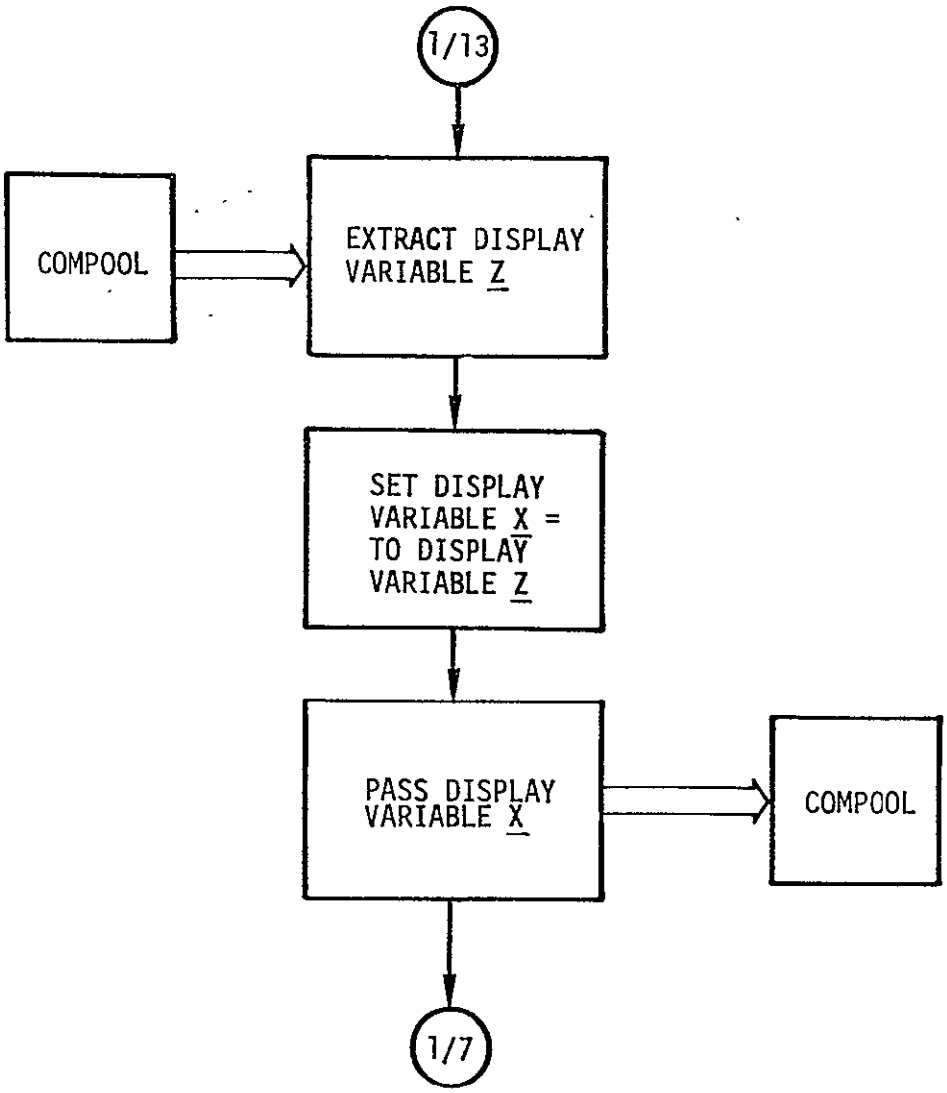


Figure 34. Concluded

5.23 POOL ROUTINE

Description - The POOL routine provides an activity number to index scheduled events in the File 1 data set. If activity numbers of unscheduled events are stored in the available activity number array of the File 1 data set, the last entry of the array will be returned as the activity number for the new event. If the available activity number array is blank, a new activity number will be created for the new event.

Interface

I/O DEVICES - none.
DATA BASE - COMPOOL for both input and output.
ROUTINES CALLING POOL - ADD routine.
ROUTINES CALLED BY POOL - none.

Internal Variables - None.

Input - The POOL routine requires the following input data accessed through the COMPOOL:

N		Entry counter
M		Pool counter
NM(I)	I=1, M	Available activity number array.

Processing - The flow diagram of the POOL routine is presented in Figure 35.

Output - The POOL routine transmits the following data through the COMPOOL:

K	Activity number to index scheduled events
N	Entry counter
M	Pool counter
NM(M)	The Mth entry of the available activity number array.

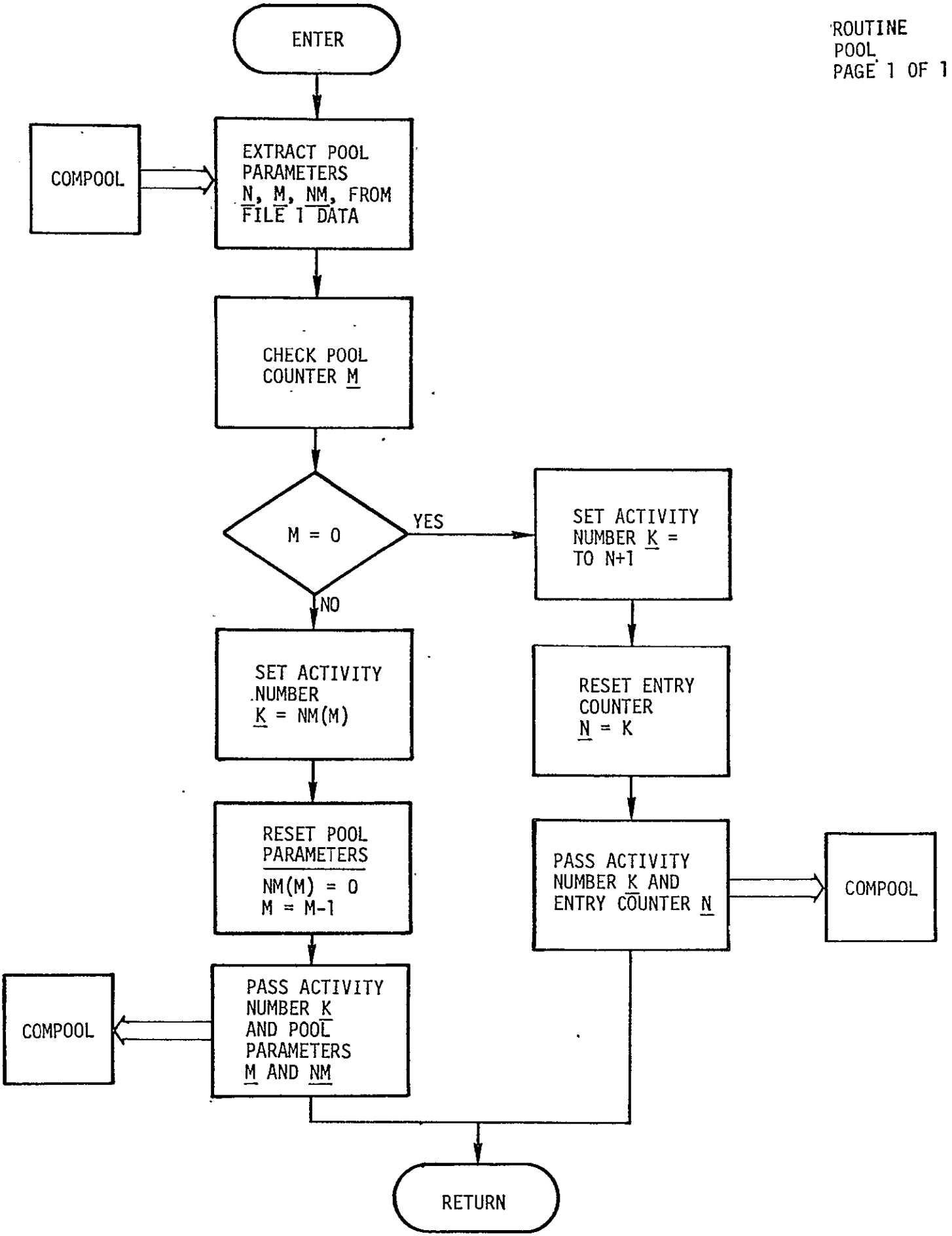


Figure 35. Flow Diagram for the POOL Routine

5.24 RATE ROUTINE

Description - The RATE routine builds the consumable rate tables for each subsystem affected by a scheduled or unscheduled event. The RATE tables are internal tables used by other control and support routines for constraint checking and consumable usage integration. The ACTION-RATE Cross Reference Table will be used to identify the number and specific rate tables affected by an event. Figure 36 represents a skeleton for the ACTION-RATE Cross Reference Table.

Interface

I/O DEVICES - None.
DATA BASE - COMPOOL for both input and output; and the Consumables Analysis Data Base for input.
ROUTINES CALLING RATE - BUILD, FLIGHT, and ACTION routines.
ROUTINES CALLED BY RATE - CONSTRAINT routine.

Internal Variables

TFLAG	Flag indicating phase of activity being scheduled: TFLAG=1 preparation period 2 activity period 3 post activity period.
TIN	Time to initiate search.
TOUT	Time to complete search.
OP	The number of operations in a phase of an activity.
INR	Index for rate table.

Input - The RATE routine requires the following input data accessed through COMPOOL:

J	The ACTION identifier required by the event being scheduled or unscheduled (see Table I for the values of J).
ACTION MODE	Mode flag for the ACTION routine to schedule or unschedule an event: ADD = schedule an event DELETE = unschedule an event.
K	The activity number for the event to be scheduled or unscheduled.

AT(K,I)	I=1,5	Entry data array for activity K I=1 prep start time 2 reference start time 3 reference stop time 4 post end time 5 special parameter, a function of ACTION Identifier J: J=1,2,4,6, or 7; AT(K,5)=ΔV J=9,10,20,21, or 22; AT(K,5)= Number of crew
KRATE(I)	I=1,9	The constant multiplier to be applied to the consumable rate tables.
RATE(CNUM)	CNUM=1,9	The rate tables (rate versus time) for each consumable affected by an event (see Figure 36 for values of CNUM).

If the ACTION identifier J=14 (payload related consumables), the following additional input data is accessed through the COMPOOL:

C(CNUM)	CNUM=1,9	The cross reference constant indicating what consumable rate tables are affected by the event (see Figure 36 for values of CNUM): C=0 The rate table not affected 1 The rate table is affected.
K(J,CNUM)	J=14 CNUM=1,9	The number of operations in the preparation period of the activity.
J(J,CNUM)	J=14 CNUM=1,9	The number of operations in the activity period.
L(J,CNUM)	J=14 CNUM=1,9	The number of operations in the post-activity period.
ΔRATE(J,CNUM,INR)	J=14 CNUM=1,9 INR=1,K/J/L	The Δ rate table (rate versus time) to be added/deleted to the rate (CNUM) table by the scheduling or unscheduling of an event.

The RATE routine requires the following input data from the Consumables Analysis Data Base (unless J=14):

TABLE		The ACTION-RATE Cross Reference Table as defined in Figure 36.
C(CNUM)	CNUM=1,9	The cross reference constant indicating what consumable rate tables are affected by the event (see Figure 36 for values of CNUM): C=0 The rate table not affected 1 The rate table is affected.

K(J,CNUM)	J=1,23 CNUM=1,9	The number of operations in the preparation period of the activity.
J(J,CNUM)	J=1,23 CNUM=1,9	The number of operations in the activity period.
L(J,CNUM)	J=1,23 CNUM=1,9	The number of operations in the post-activity period.
ΔRATE(J,CNUM,INR)	J=1,23 CNUM=1,9 INR=1,K/J/L	The Δ rate table (rate versus time) to be added/deleted to the rate (CNUM) table by the scheduling or unscheduling of an event.

Processing - The flow diagram of the RATE routine is presented in Figure 37.

Output - The RATE routine transmits the following data through the COMPOOL:

TMIN		The time to start constraint checking.
TMAX		The time to end constraint checking.
CNUM		The consumable rate table identifier (see Figure 36 for values of CNUM).
RATE(CNUM)	CNUM=1,9	The rate tables (rate versus time) for each consumable affected by an event.

ACTION-RATE CROSS REFERENCE

CNUM =	1	2	3	4	5	6	7	8	9
J=-10	C	C	C	C	C	C	C	C	C
-9	C								
⋮	⋮								
1	C								
⋮	⋮								
23	C								

Consumable subsystems (CNUM) rate table:

CNUM = 1 EPS-WATTS
2 RCS-ΔV
3 OMS-ΔV
4 ECS-O_2
5 ECS-N_2
6 ECS-LiOH
7 ECS-H_2O
8 APU-Fuel
9 APU-H_2O

J The ACTION Identifier (see Table I for the definition of J). Range = -10 to 23.

CROSS REFERENCE CONSTANT	VALUE	CROSS REFERENCE
C	0	The RATE table for CNUM is not affected by J.
	1	The RATE table for CNUM is affected by J.

Figure 36. The ACTION-RATE Cross Reference Table Skeleton

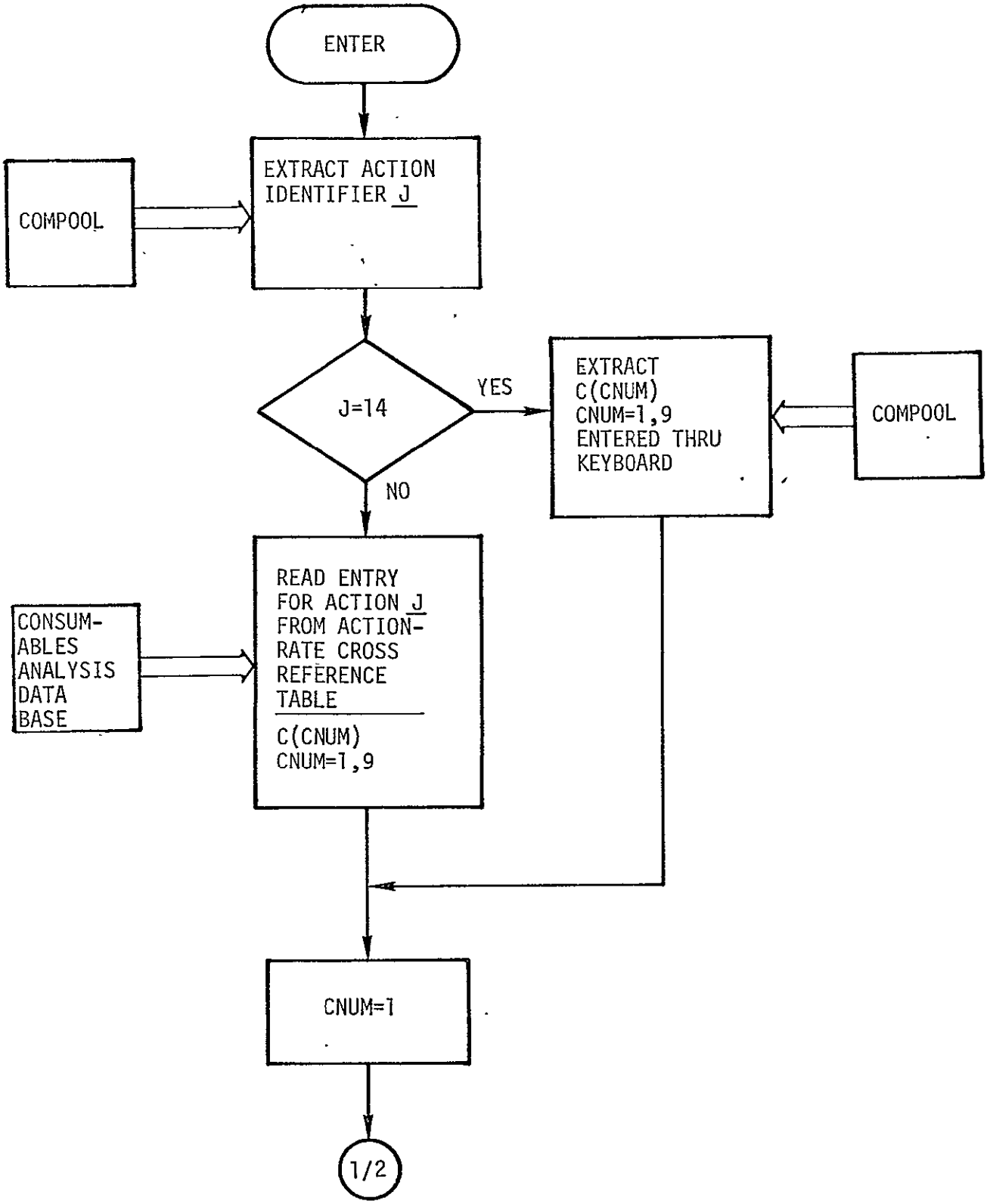


Figure 37. Flow Diagram for the RATE Routine

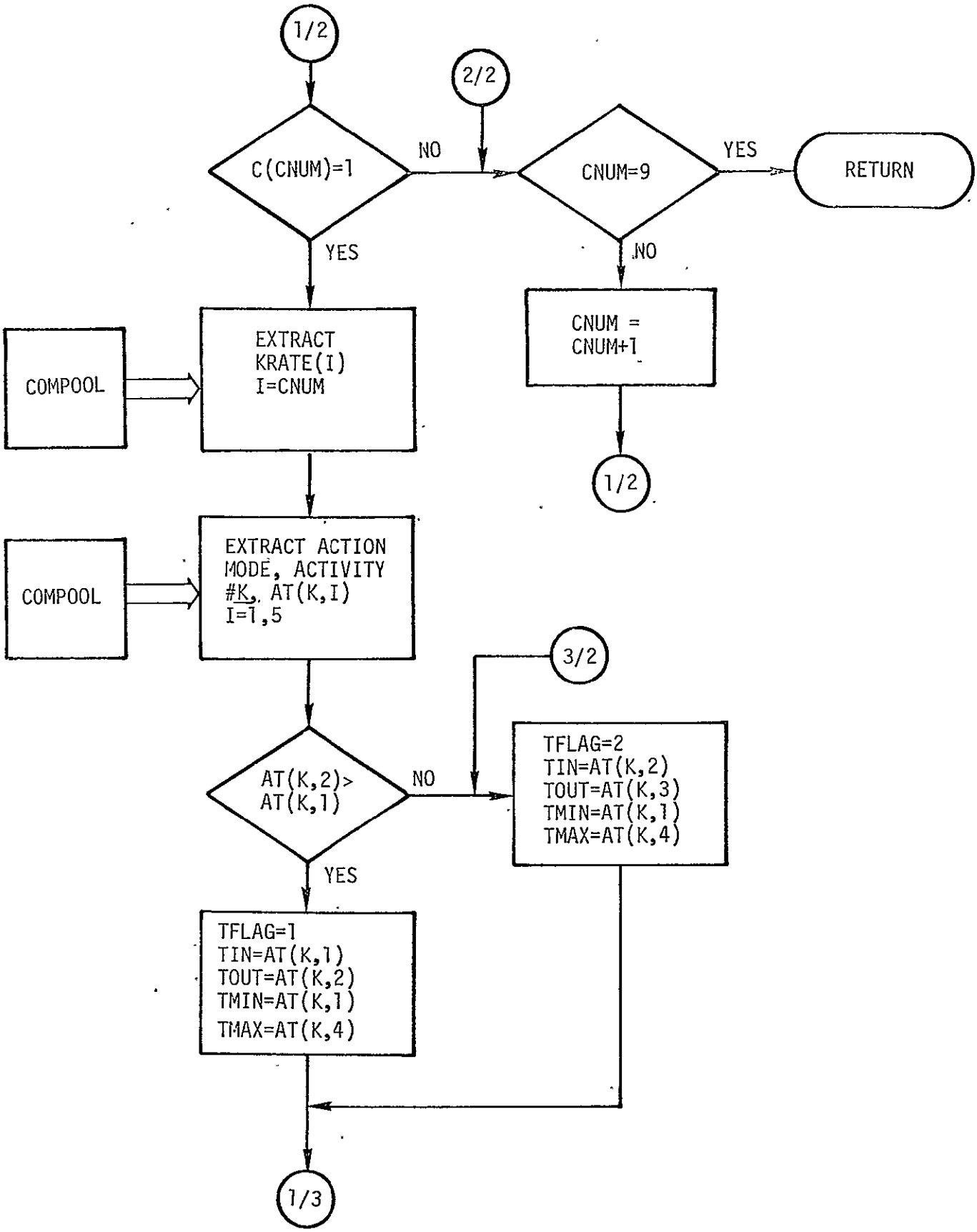


Figure 37. Continued

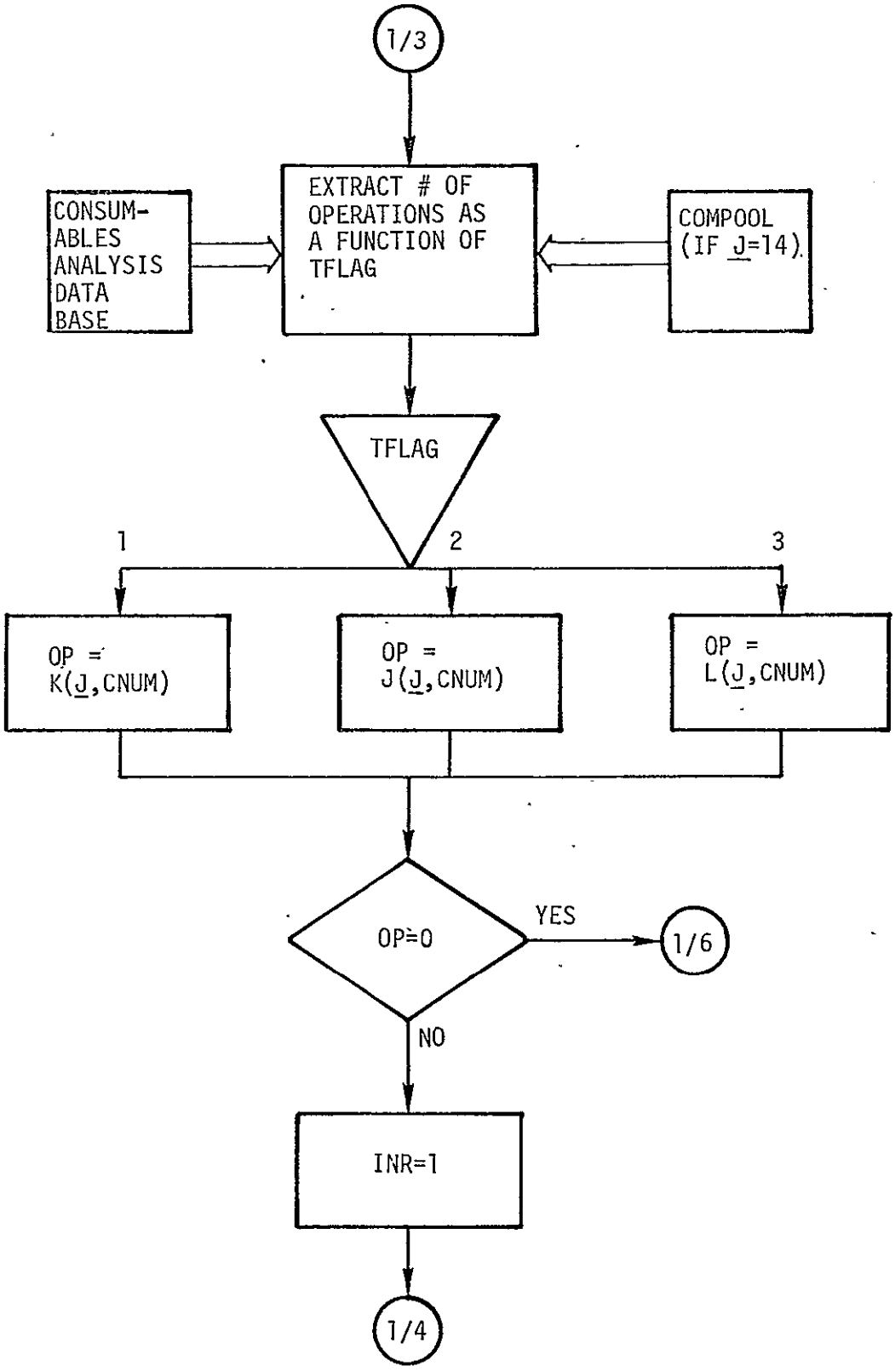


Figure 37. Continued

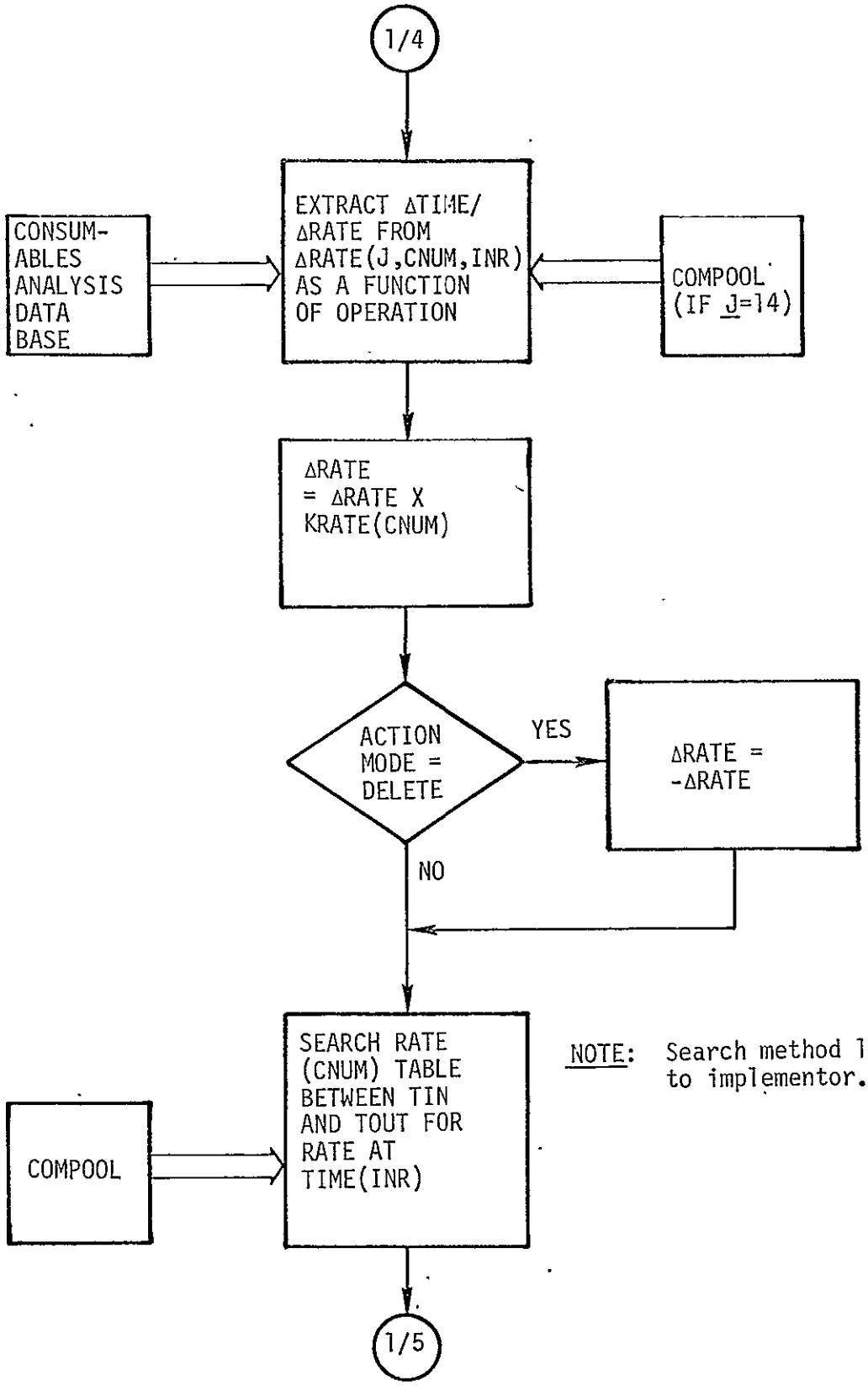


Figure 37. Continued

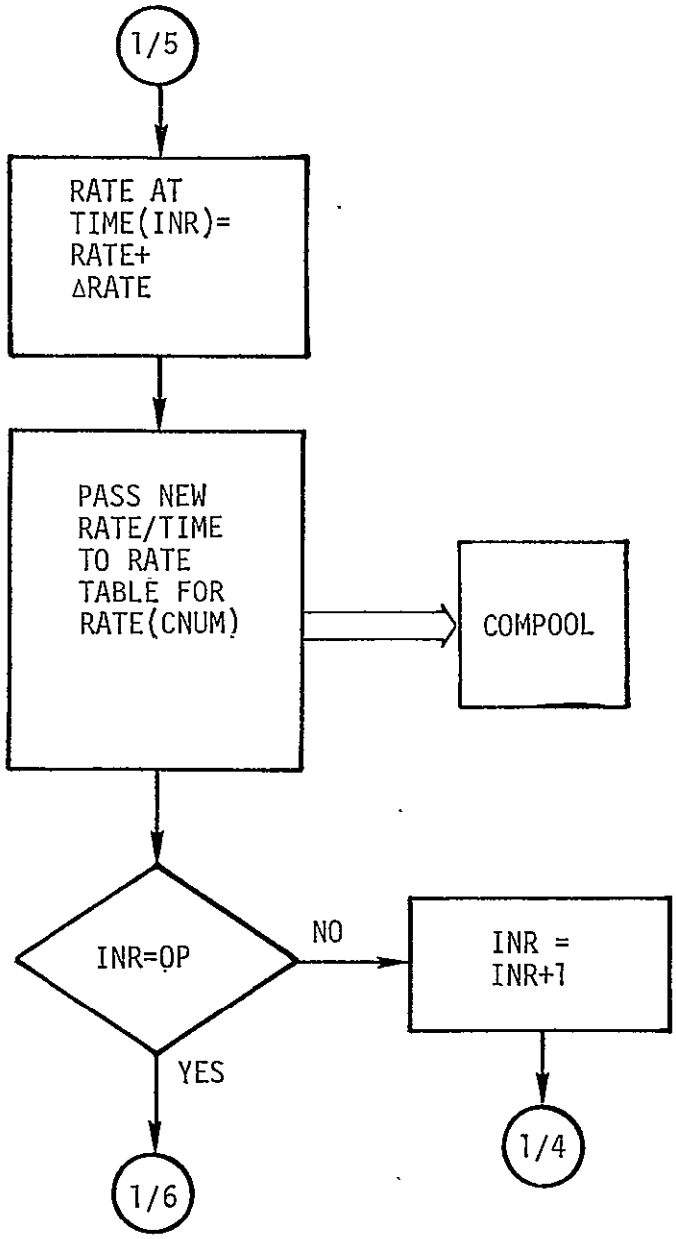


Figure 37. Continued

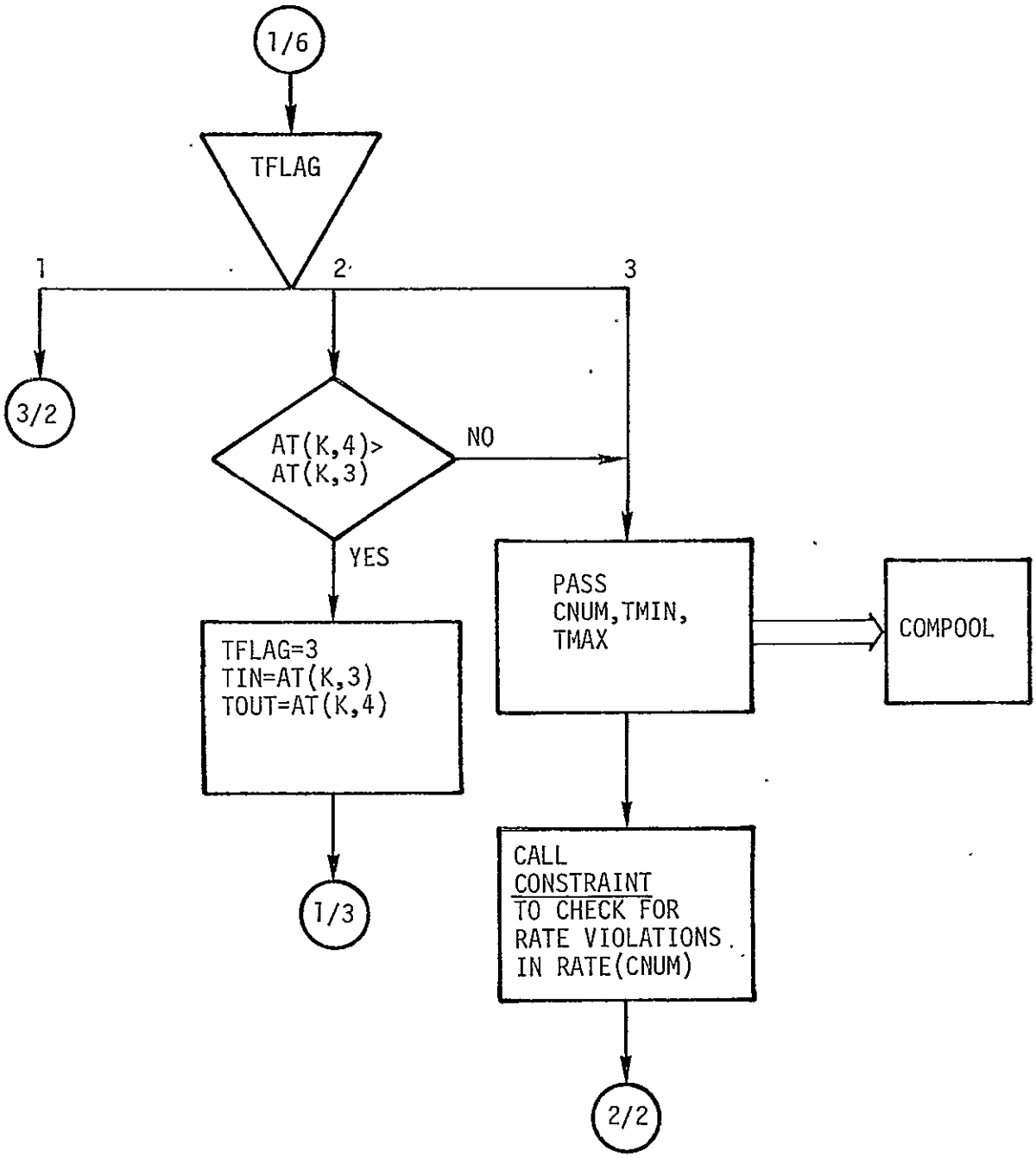


Figure 37. Concluded

5.25 SEQUENCE ROUTINE

Description - The SEQUENCE routine establishes the time relationship between scheduled events by setting the temporal parameters in the File 1 data set. If an added/modified/deleted event affects the order of subsequent events, the order will be adjusted and the temporal parameters of all subsequent events will reflect the adjustment.

Interface

I/O DEVICES - None.
DATA BASE - COMPOOL for both input and output.
ROUTINES CALLING SEQUENCE - ADD and DELETE routines.
ROUTINES CALLED BY SEQUENCE - None.

Internal Variables

ISAVE	The index for the position in the temporal parameters affected by the scheduled or unscheduled event.

Input - The SEQUENCE routine requires the following input data accessed through the COMPOOL:

NOI		The number of entries in sequence array.
IT(I)	I=1,NOI	Sequence array of activities.
TIM(I,L)	I=1,NOI L=1,2	Start and end times of activities: L=1 minimum start time of activity IT(I) 2 maximum end time of activity IT(I).
ACTION MODE		Mode flag for the ACTION routine to schedule or unschedule an event: ADD = schedule an event DELETE = unschedule an event.
K		The activity number for the event to be scheduled or unscheduled.
AT(K,I)	I=1,5	Entry data array for activity K I=1 prep start time 2 reference start time 3 reference stop time 4 post end time 5 special parameter, a function of ACTION Identifier J: J=1,2,4,6, or 7: AT(K,5)=ΔV J=9,10,20,21, or 22; AT(K,5)=Number of crew.

Processing - The flow diagram of the SEQUENCE routine is presented in Figure 38.

Output - The SEQUENCE routine transmits the following data through the COMPOOL:

NOI		The number of entries in sequence array.
IT(I)	I=1,NOI	Sequence array of activities.
TIM(I,L)	I=1,NOI L=1,2	Start and end times of activities: L=1 minimun start time of activity IT(I) 2 maximum end time of activity IT(I).

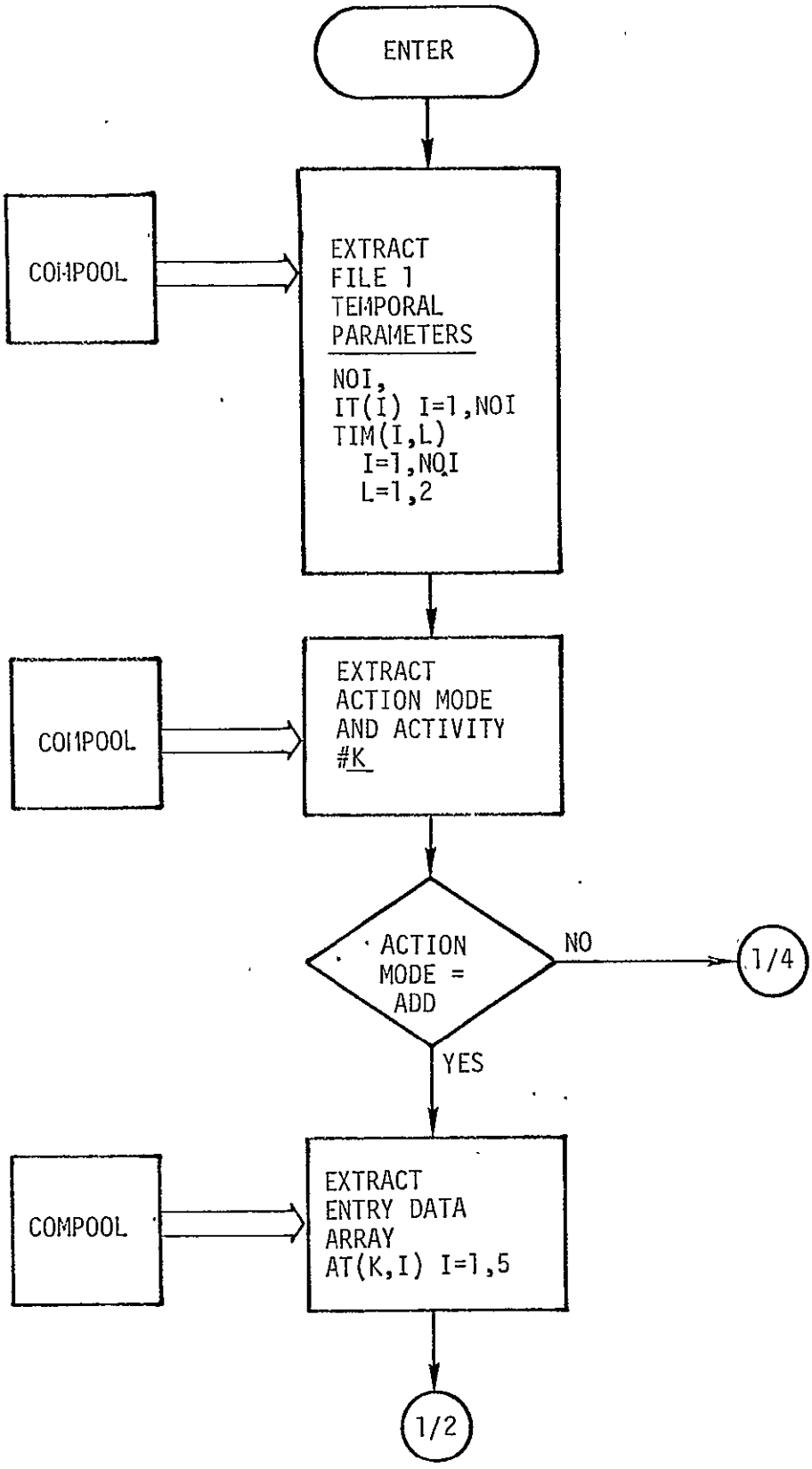


Figure 38. Flow Diagram for the SEQUENCE Routine

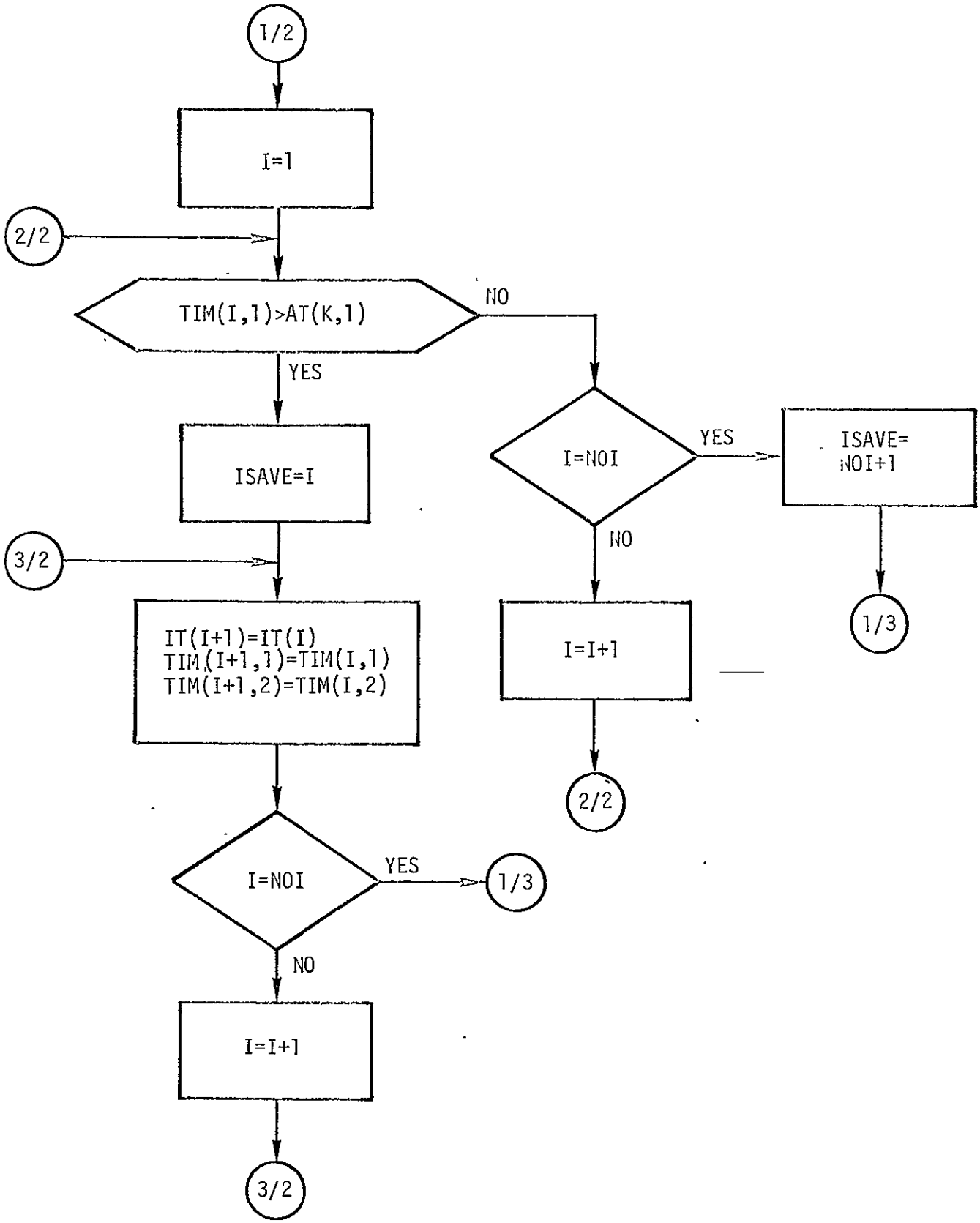


Figure 38. Continued

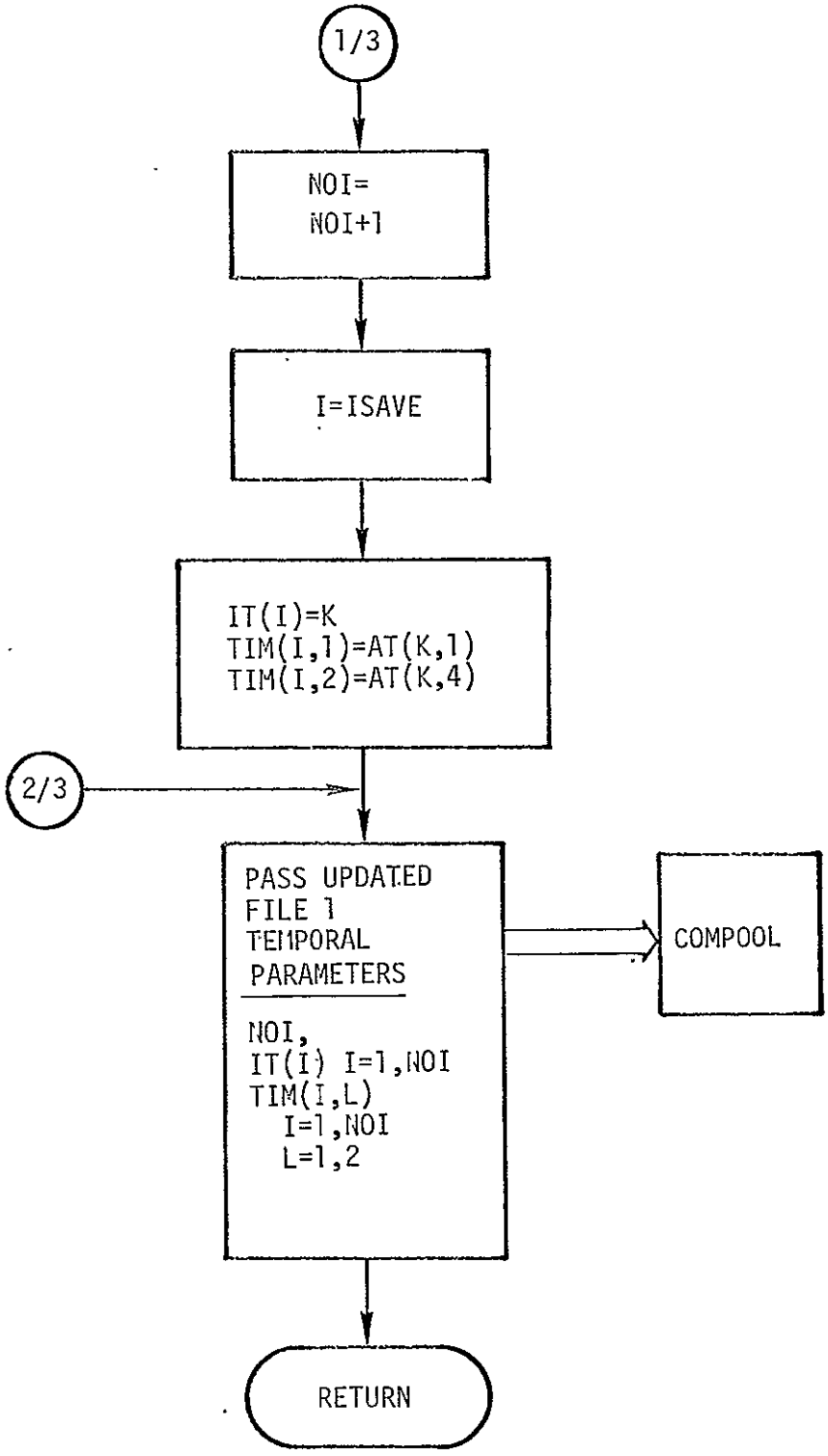


Figure 38. Continued

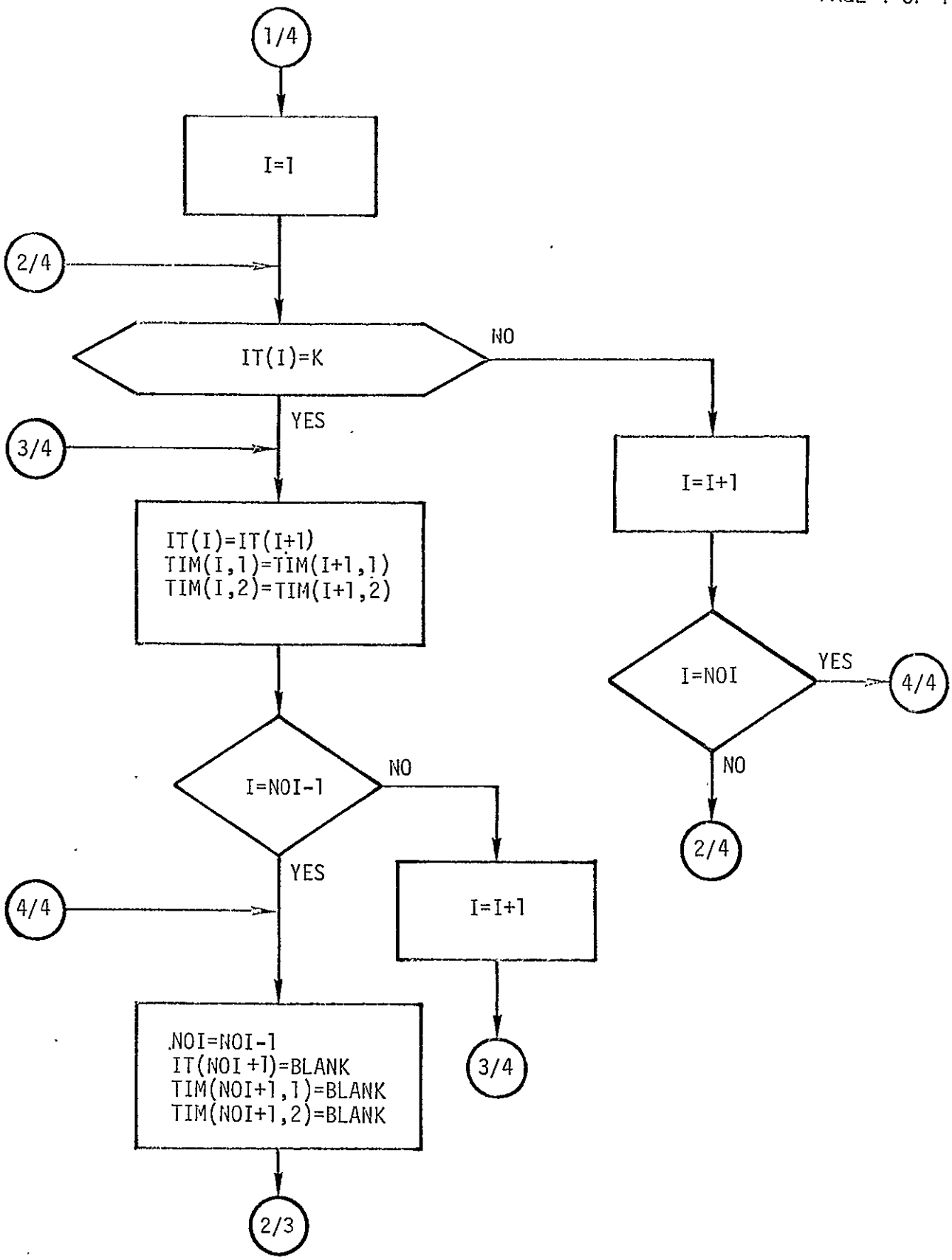


Figure 38. Concluded

5.26 SPECIAL ROUTINE

Description - The SPECIAL routine provides special handling required to schedule or unschedule specific events as a function of the ACTION Identifier J. The special handling includes: setting the special parameter AT(K,5) in the File 1 data entry array; setting constant multipliers to be used in applying consumable rates to the rate tables; calling computational routines that utilize spacecraft subsystem modeling; processing specific payload consumable rates input by the user; and preparing the data required to impose common block ($J \leq 0$) rates on the consumable rate tables.

Interface

I/O DEVICES - The terminal KEYBOARD unit for input.
DATA BASE - COMPOOL for both input and output.
ROUTINES CALLING SPECIAL - ACTION, BUILD, and FLIGHT routines.
ROUTINES CALLED BY SPECIAL - CFLT, COMS, CRCS, CATH, CREND, CDOCK CUDOCK, CEVA, CIVA, CES, and CWM computational routines.

Internal Variables - None.

Input - The SPECIAL routine requires the following input data accessed through the COMPOOL:

J	The ACTION identifier required by the event to be scheduled or unscheduled (see Table I for the values of J).
K	Activity number for the event to be scheduled or unscheduled.
NCREW	The number of crew required to perform an event.

If the ACTION identifier $J \leq 0$, the following additional data are required accessed through the COMPOOL:

BPT(I)	I=1,11	Block phase times I=1 prelaunch start 2 prelaunch stop/ascent start 3 liftoff 4 MECO 5 ETS 6 OMS ignition/on-orbit start 7 on-orbit stop/deorbit start 8 deorbit burn ignition 9 deorbit stop/entry start 10 rollout 11 entry/land stop.

If the ACTION identifier J=14 (payload related consumables), the following input data is accessed through the terminal KEYBOARD unit:

REF START		The reference start time for the event to be scheduled or unscheduled.
REF STOP		The reference stop time for the event to be scheduled or unscheduled.
KRATE(I)	I=1,9	The constant multiplier to be applied to the consumable rate tables.
C(CNUM)	CNUM=1,9	The cross reference constant indicating what consumable rate tables are affected by the event (see Figure 36 for values of CNUM): C=0 The rate table not affected 1 The rate table is affected.
K(J,CUNM)	J=14 CNUM=1,9	The number of operations in the preparation period of the activity.
J(J,CNUM)	J=14 CNUM=1,9	The number of operations in the activity period.
L(J,CNUM)	J=14 CNUM=1,9	The number of operations in the post-activity period.
ΔRATE(J,CNUM,INR)	J=14 CNUM=1,9 INR=1,K/J/L	The Δ rate table (rate versus time) to be added/deleted to the rate (CNUM) table by the scheduling or unscheduling of an event.

Processing - The flow diagram of the SPECIAL routine is presented in Figure 39.

Output - The SPECIAL routine transmits the following data through the COMPOOL:

AT(K,5)		The special parameter in the entry data array for activity K as a function of ACTION Identifier J: J=1,2,4,6, or 7; AT(K,5)=ΔV J=9,10,20,21, or 22; AT(K,5)=Number of crew.
KRATE(I)	I=1,9	The constant multiplier to be applied to the consumable rate tables.

If the ACTION identifier J=14, the following additional output data will be transmitted through the COMPOOL:

C(CNUM)	CNUM=1,9	The cross reference constant indicating what consumable rate tables are affected by the event (see Figure 36 for values of CNUM): C=0 The rate table not affected 1 The rate table is affected.
K(J,CNUM)	J=14 CNUM=1,9	The number of operations in the preparation period of the activity.
J(J,CNUM)	J=14 CNUM=1,9	The number of operations in the activity period.
L(J,CNUM)	J=14 CNUM-1,9	The number of operations in the post-activity period.
ΔRATE(J,CNUM,INR)	J=14 CNUM=1,9 INR=1,K/J/L	The Δ rate table (rate versus time) to be added/deleted to the rate (CNUM) table by the scheduling or unscheduling of an event.
REF START		The reference start time for the event to be scheduled or unscheduled.
REF STOP		The reference stop time for the event to be scheduled or unscheduled.

If the ACTION identifier $J \leq 0$, the following additional output data will be transmitted through the COMPOOL:

REF START	The reference start time for the common block to be scheduled or unscheduled.
REF STOP	The reference stop time for the common block to be scheduled or unscheduled.
J	The ACTION identifier for the common block to be scheduled or unscheduled (see Table I for the values of J).

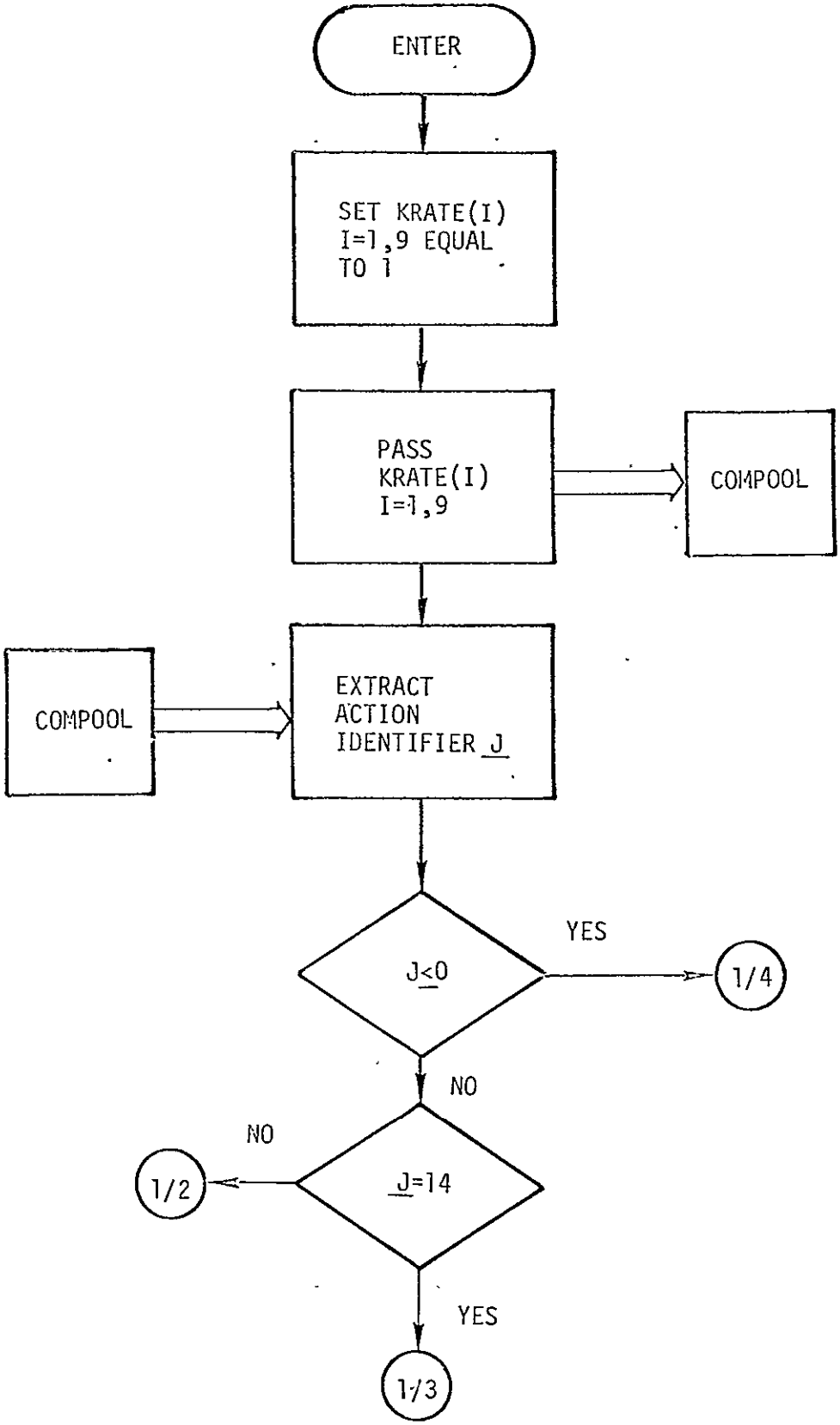


Figure 39. Flow Diagram for the SPECIAL Routine

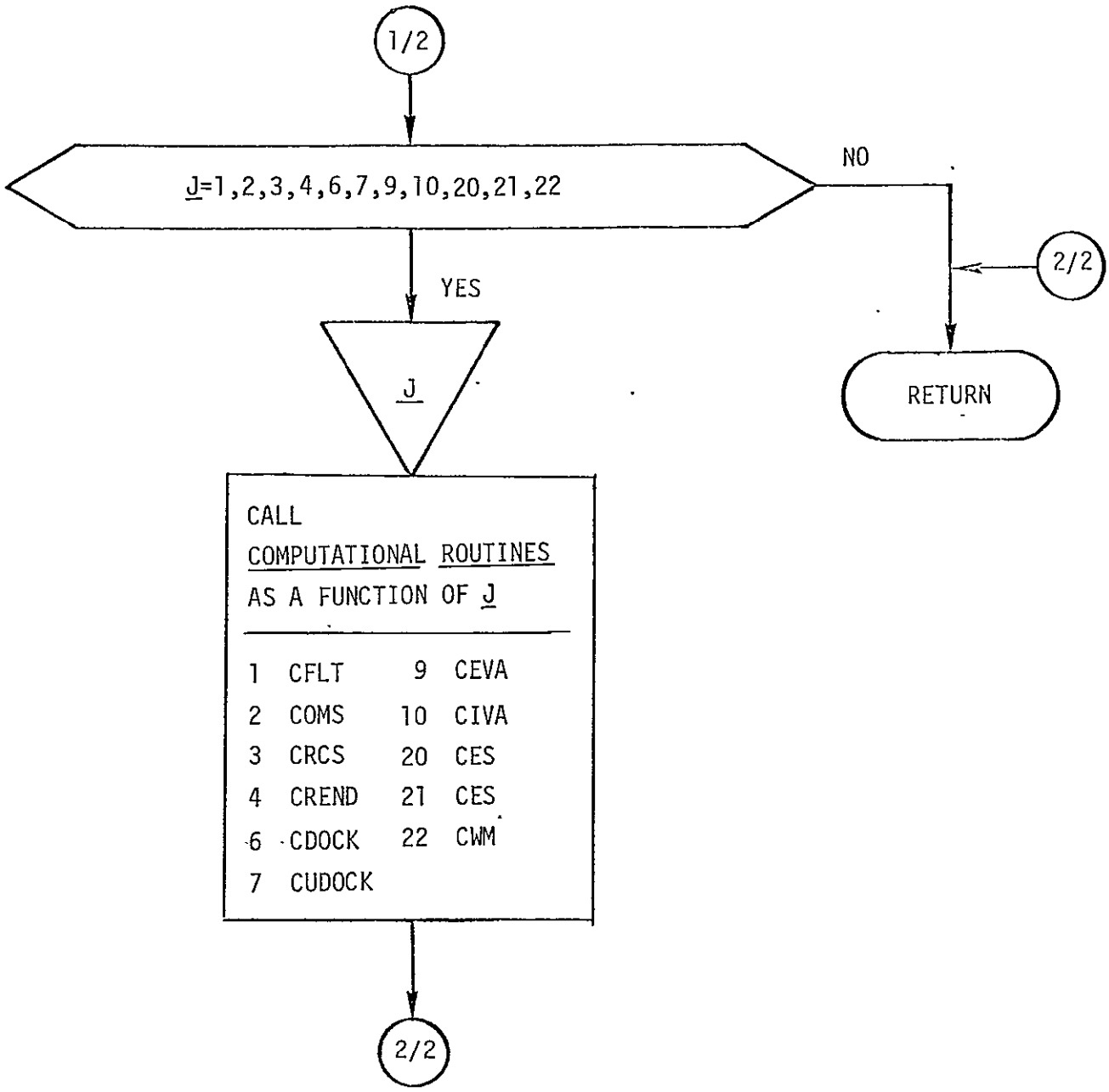


Figure 39. Continued

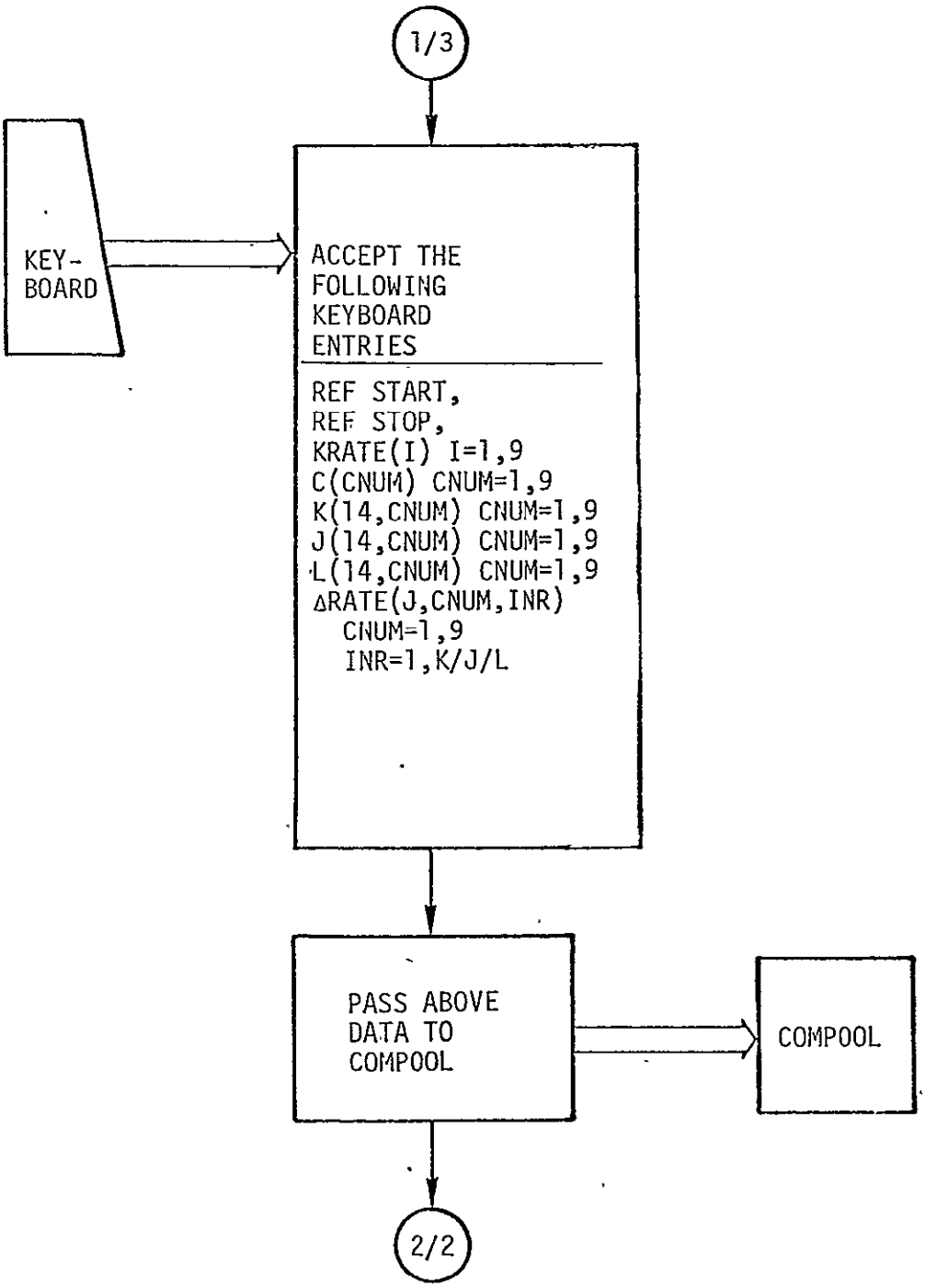


Figure 39. Continued

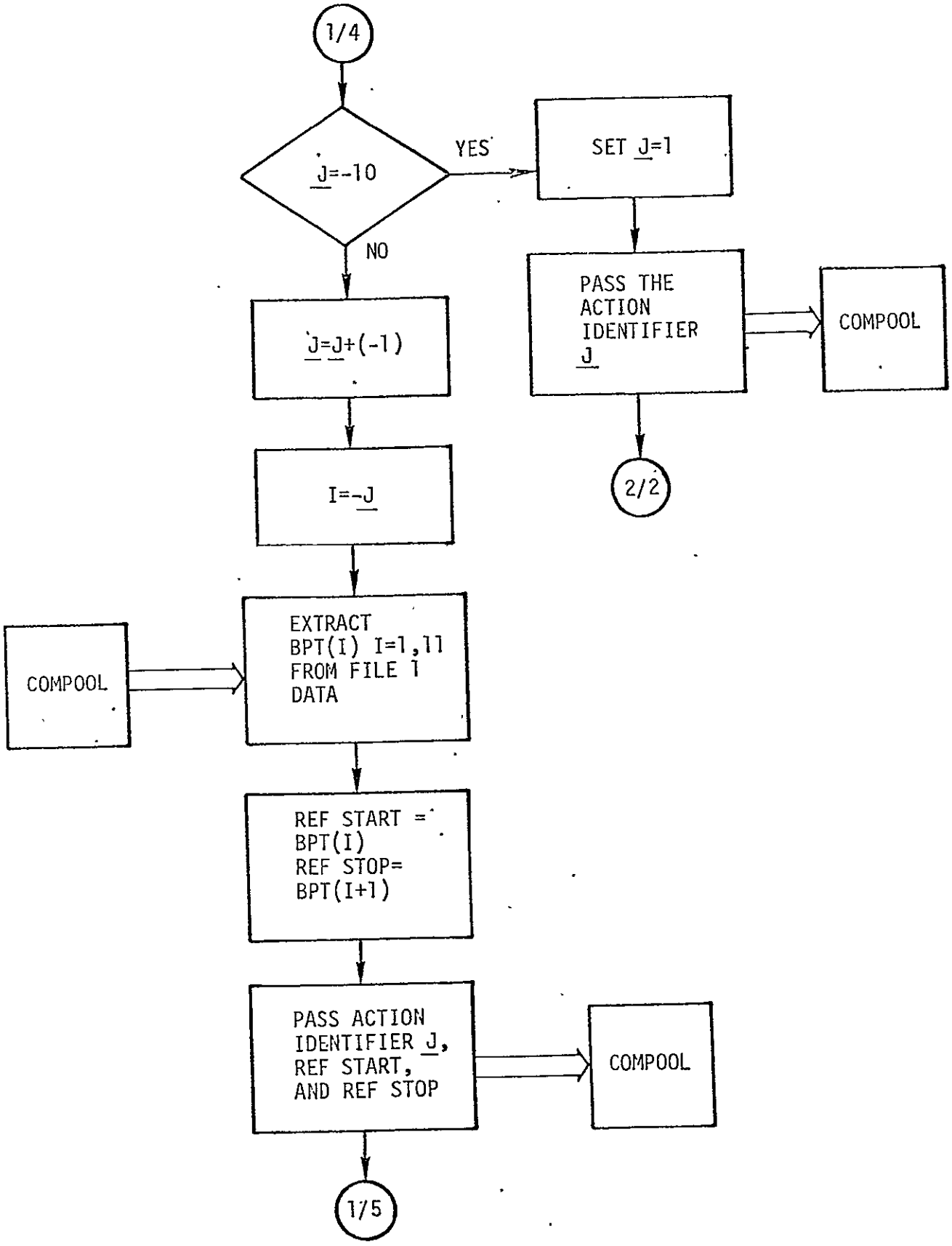


Figure 39. Continued

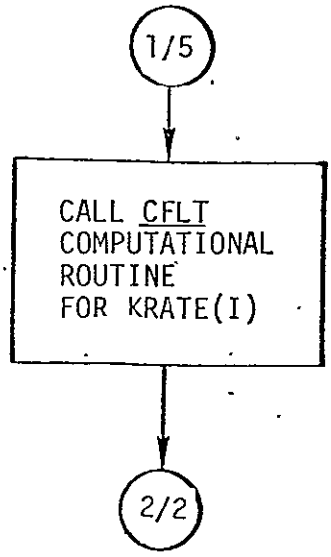


Figure 39. Concluded

5.27 TIMELINE ROUTINE

Description - The TIMELINE routine prepares the scheduled events versus time data for display. The mechanization of this process will be defined at time of implementation.

Interface

I/O DEVICES - None.
DATA BASE - COMPOOL for both input and output.
ROUTINES CALLING TIMELINE - OUTPUT and CONSUM HISTORY routines.
ROUTINES CALLED BY TIMELINE - None.

Internal Variables - None.

Input - Will be defined during implementation.

Processing - The TIMELINE routine is an information management type of routine used to prepare the results of consumables analysis for display. No flow diagram is necessary.

Output - Will be defined during implementation.

6.0 DEFINITION OF THE COMPUTATIONAL ROUTINES

The computational routines defined in this section perform specific manipulation of the input and/or consumables data base information required prior to processing several of the activities. For propulsion-related activities, these routines calculate the equivalent acceleration, burn time, and activity start of stop time. For crew-related activities, these routines evaluate the impact of the number of crew members involved in the activity on the consumables usage rates.

6.1 CFLT Routine

Description - The CFLT routine calculates the influence of the crew size on the consumables rates resident in the consumables analysis data base for the Flight Common activity. Appropriate factors to be applied to the resident rates are made available for subsequent processing by the Control and Support subroutines.

Interface

I/O DEVICE - None.
DATA BASE - COMPOOL for both input and output, Consumables Analysis Data Base (CADB) for input only.
ROUTINES CALLING CFLT - SPECIAL.
ROUTINES CALLED BY CFLT - None.

Internal Variables

CNUM	Index to consumable affected by activity.
INR	Index to particular operation within activity consumables data base.

Input - The CFLT routine requiring the following input data accessed through COMPOOL:

J	ACTION identifier.
FCREW	Crew number.

The CFLT routine requires the following input data accessed through CADB:

K(J,CNUM)	Number of operations in the preparation period of the activity.
ΔRATE(J,CNUM,INR)	The Δ rate table.

Processing - The flow diagram of the CFLT routine is shown in Figure 40.

Output - The CFLT routine transmits the following data through the COMPOOL:

KRATE(CNUM,INR)	Factor to multiply ΔRATE by.

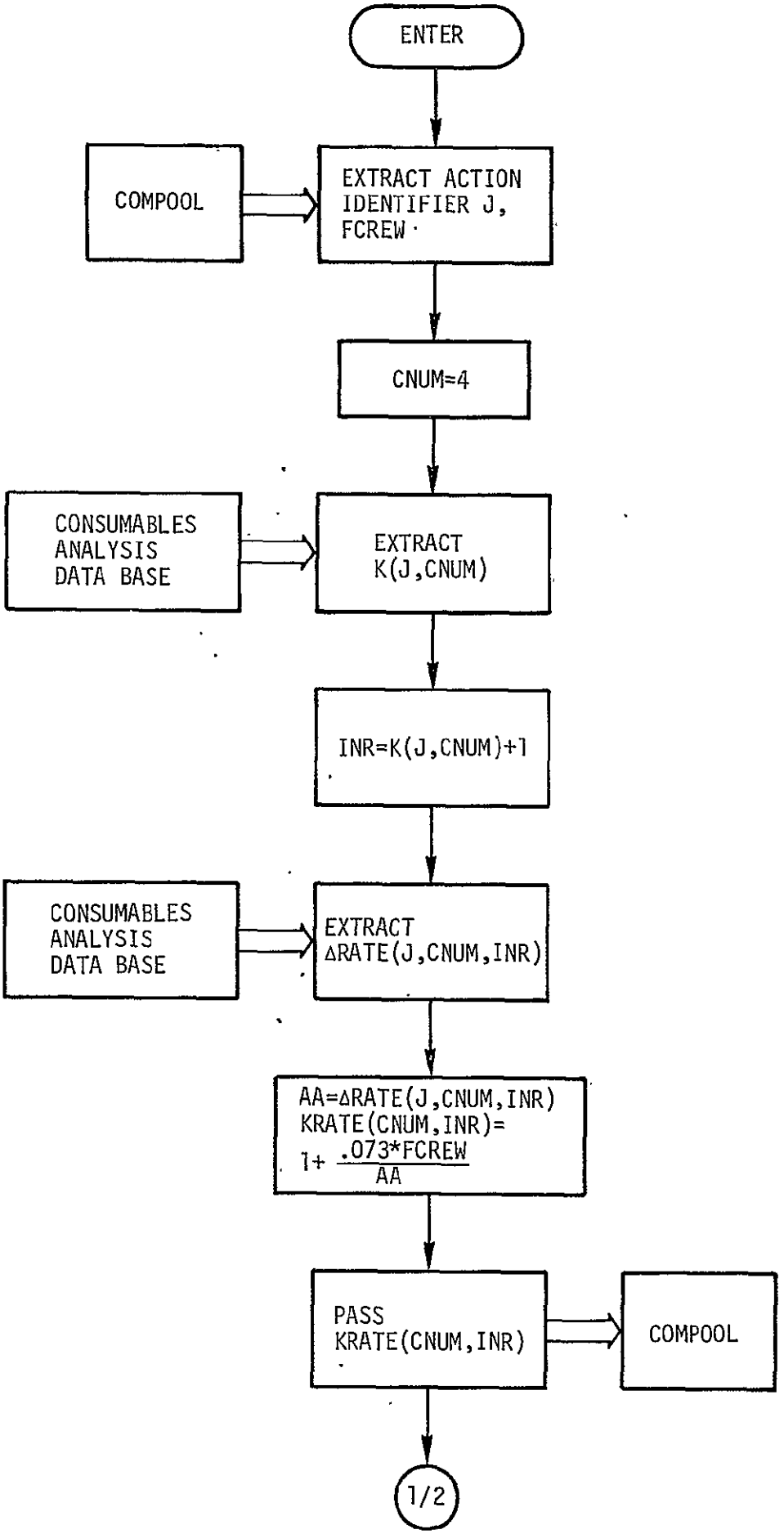


Figure 40. Flow Diagram for the CFLT Routine

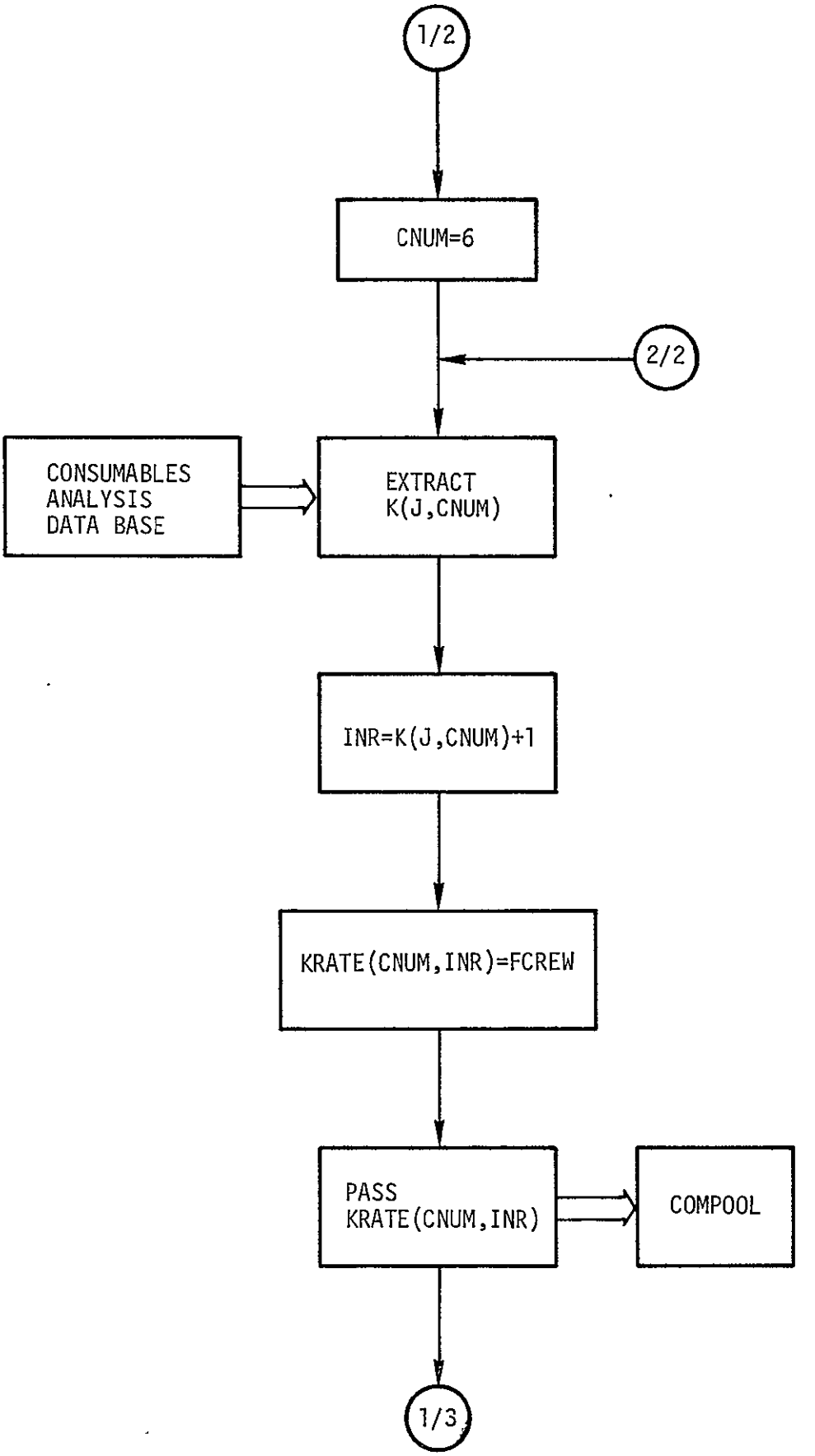


Figure 40. Continued

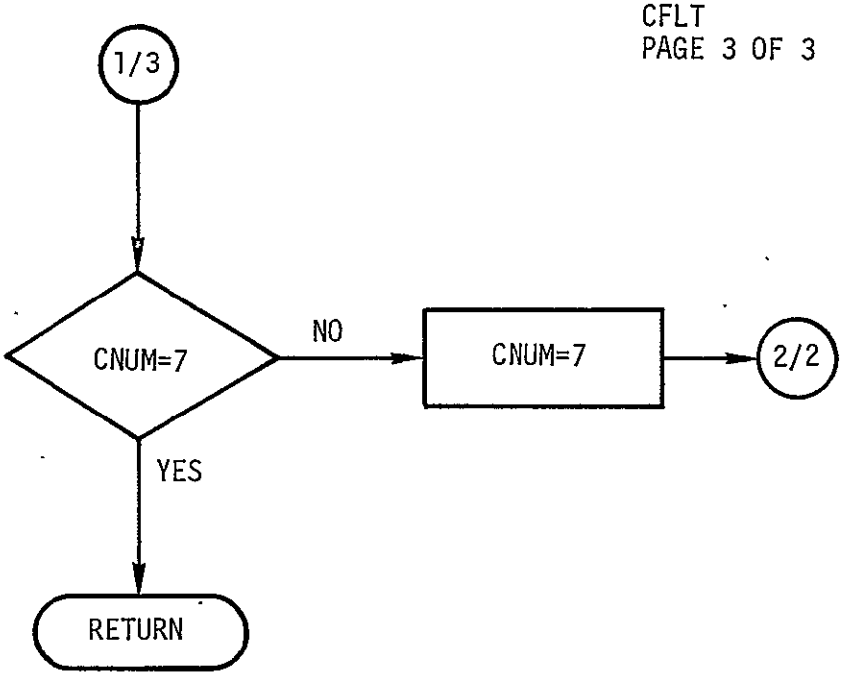


Figure 40. Concluded

6.2 COMS Routine

Description - The COMS routine calculates the burn time, activity reference stop time, and the acceleration rate for an OMS maneuver.

Interface

I/O DEVICE - None.
DATA BASE - COMPOOL for both input and output, Consumables Analysis Data Base (CADB) for input only.
ROUTINES CALLING COMS - SPECIAL.
ROUTINES CALLED BY COMS - None.

Internal Variables

CNUM	Index to consumable affected by activity.
INR	Index to particular operation with activity consumables data base.
DEL	OMS burn time.

Input - The COMS routine requiring the following input data accessed through COMPOOL:

J	ACTION identifier.
K	Activity number.
AT(K,I)I=1,5	Entry data array for activity.

The COMS routine requires the following input data accessed through CADB:

K(J,CNUM)	Number of operations in the preparation period of the activity.

Processing - The flow diagram of the COMS routine is shown in Figure 41.

Output - The COMS routine transmits the following data through the COMPOOL:

KRATE(CNUM,INR)	Factor to multiply ΔRATE by.
AT(K,I)I=1,5	Entry data array for activity.

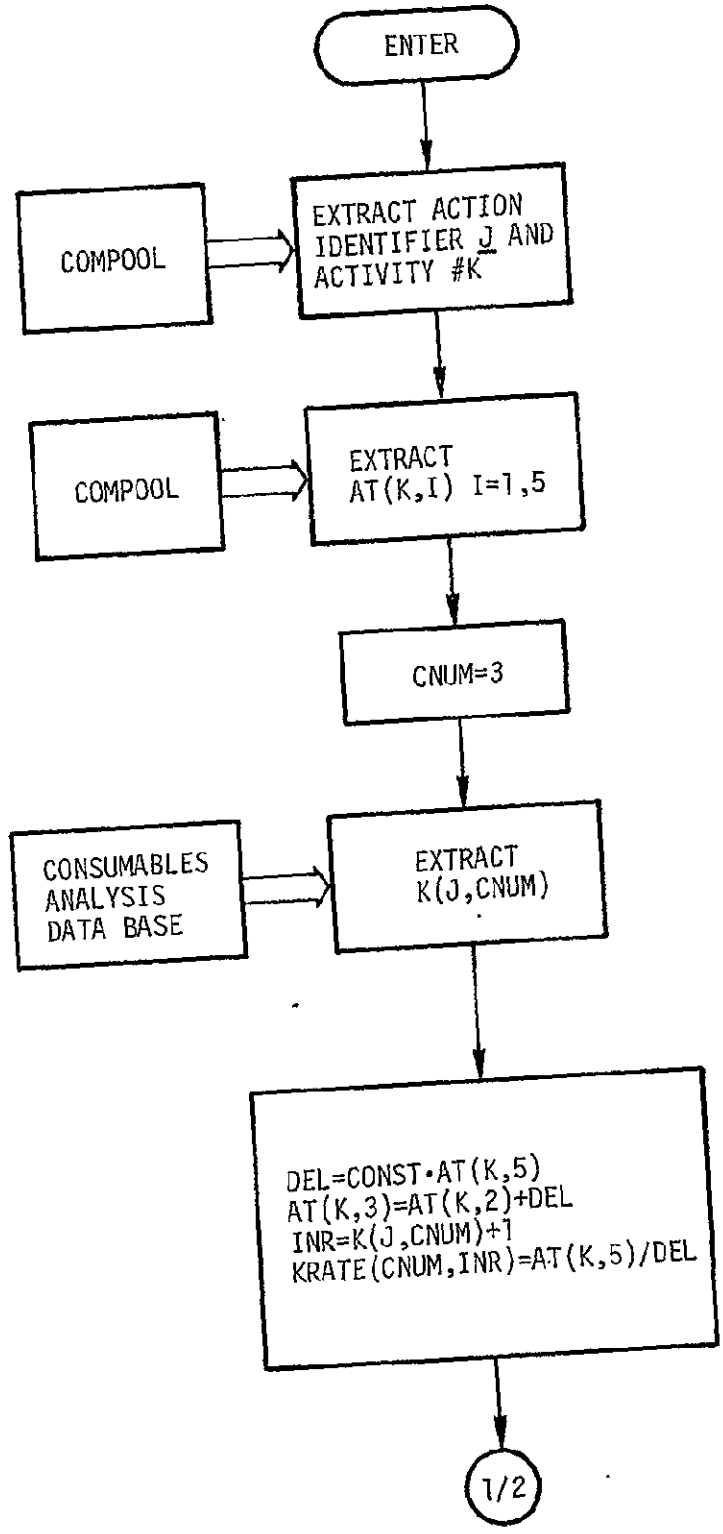


Figure 41. Flow Diagram for the COMS Routine

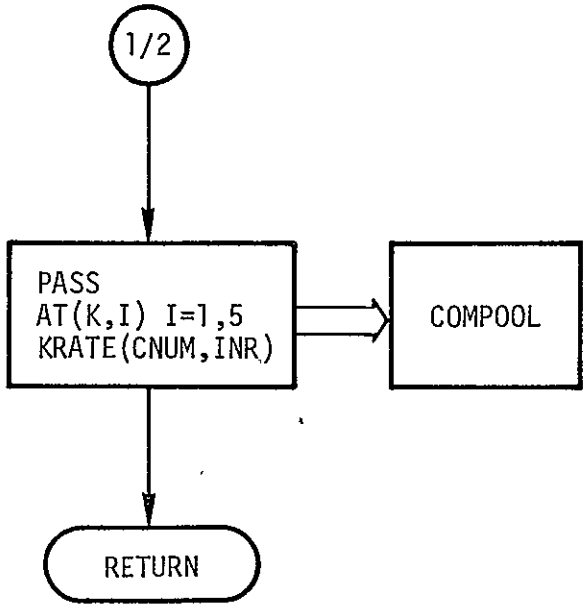


Figure 41. Concluded

6.3 CRCS Routine

Description - The CRCS routine calculates the burn time, activity reference stop time, and the acceleration rate for an RCS translation maneuver.

Interface

I/O DEVICE - None.
DATA BASE - COMPOOL for both input and output, Consumables Analysis Data Base (CADB) for input only.
ROUTINES CALLING CRCS - SPECIAL.
ROUTINES CALLED BY CRCS - None.

Internal Variables

CNUM	Index to consumable affected by activity.
INR	Index to particular operation with activity consumables data base.
DEL	RCS burn time.

Input - The CRCS routine requiring the following input data accessed through COMPOOL:

J	ACTION identifier.
K	Activity number.
AT(K,I)I=1,5	Entry data array for activity.

The CRCS routine requires the following input data accessed through CADB:

K(J,CNUM)	Number of operations in the preparation period of the activity.

Processing - The flow diagram of the CRCS routine is shown in Figure 42.

Output - The CRCS routine transmits the following data through the COMPOOL:

KRATE(CNUM,INR)	Factor to multiply ΔRATE by.
AT(K,I)I=1,5	Entry data array for activity.

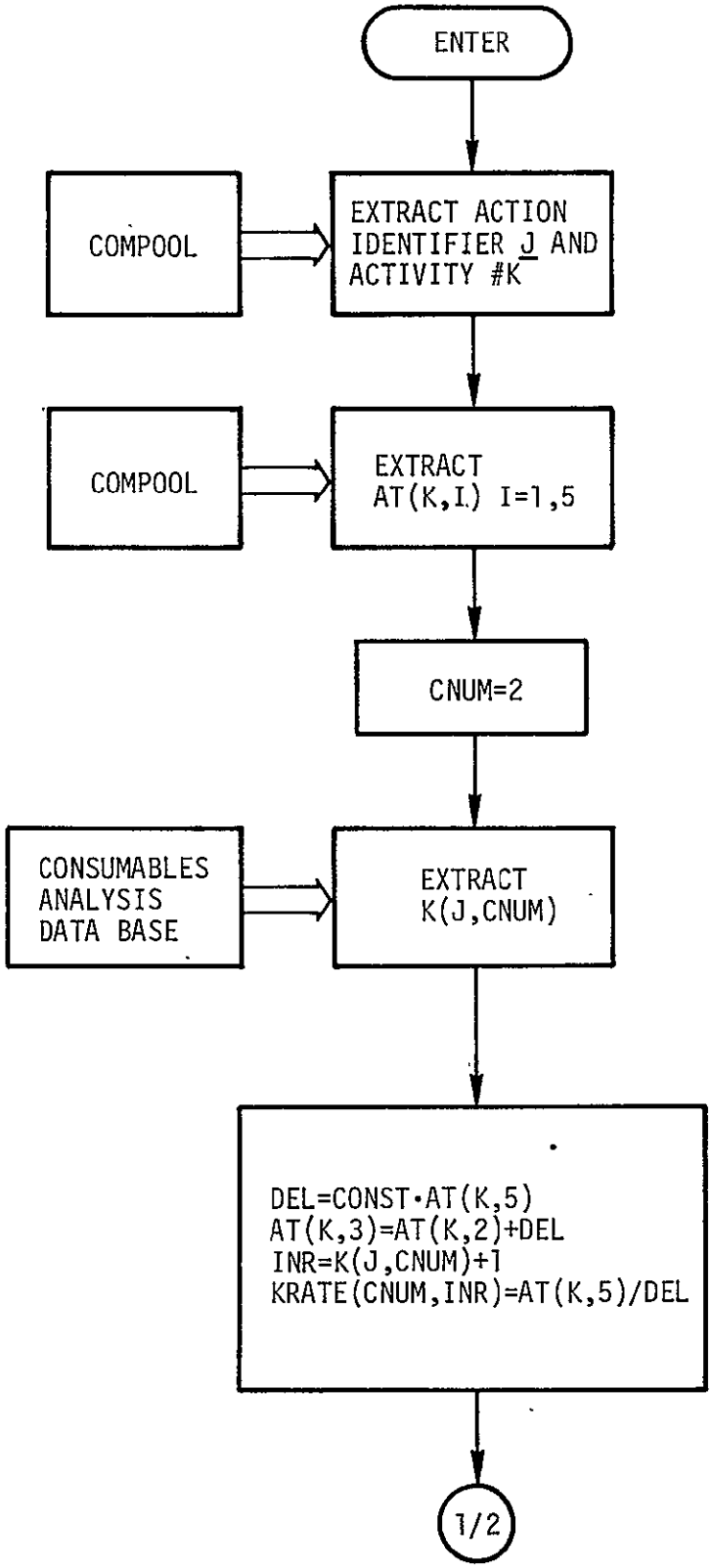


Figure 42. Flow Diagram for the CRCS Routine

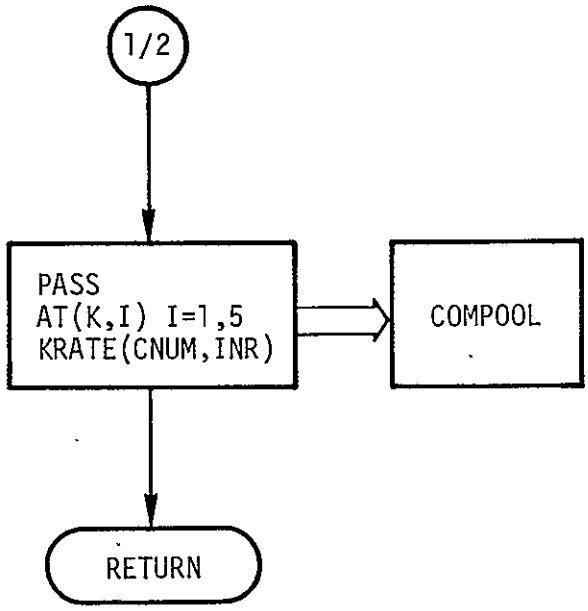


Figure 42. Concluded

6.4 CATH Routine

Description - The CATH routine calculates the RCS propellant required for an attitude hold at various altitudes.

Interface

I/O DEVICE - None.
DATA BASE - COMPOOL for both input and output, Consumables Analysis Data Base (CADB) for input only.
ROUTINES CALLING CATH - SPECIAL.
ROUTINES CALLED BY CATH - None.

Internal Variables

CNUM	Index to consumable affected by activity.
INR	Index to particular operation with activity consumables data base.

Input - The CATH routine requiring the following input data accessed through COMPOOL:

J	ACTION identifier.
K	Activity number.
AT(K,I)I=1,5	Entry data array for activity.

The CATH routine requires the following input data accessed through CADB:

K(J,CNUM)	Number of operations in the preparation period of the activity.
F(AT(K,5))	Equivalent acceleration rate for attitude hold.

Processing - The flow diagram of the CATH routine is shown in Figure 43.

Output - The CATH routine transmits the following data through the COMPOOL:

KRATE(CNUM,INR)	Factor to multiply ΔRATE by.

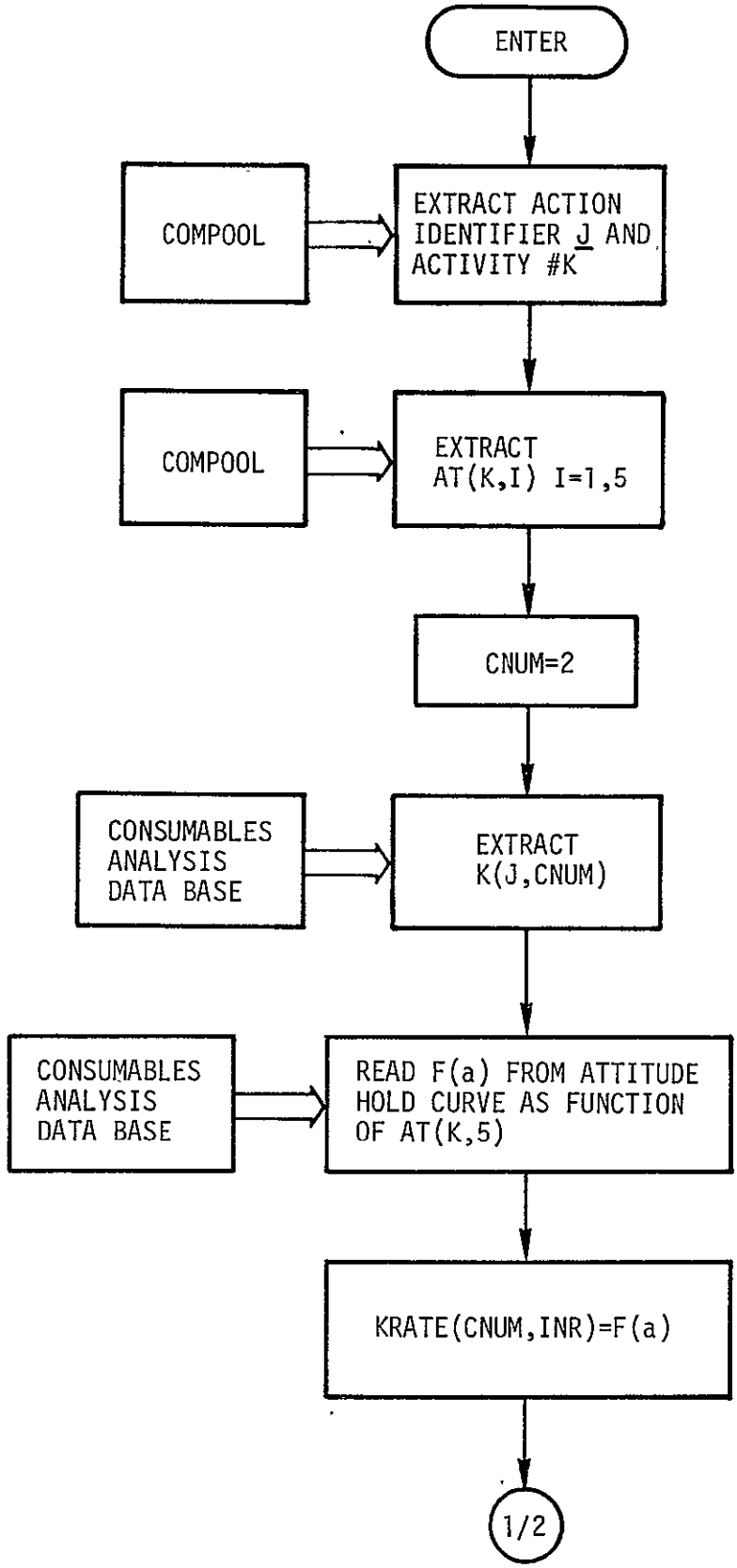


Figure 43. Flow Diagram for the CATH Routine

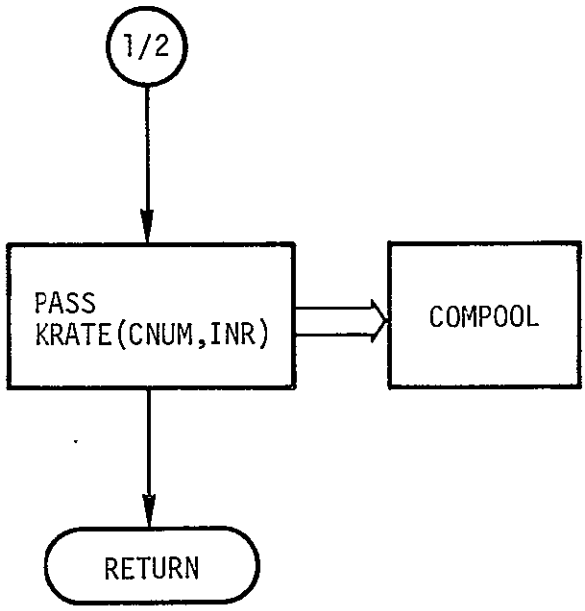


Figure 43. Concluded

6.5 CREND Routine

Description - The CREND routine calculates the burn time, activity reference stop time, and the acceleration rate for a rendezvous braking burn.

Interface

I/O DEVICE - None.
DATA BASE - COMPOOL for both input and output, Consumables Analysis Data Base (CADB) for input only.
ROUTINES CALLING CREND - SPECIAL.
ROUTINES CALLED BY CREND - None.

Internal Variables

CNUM	Index to consumable affected by activity.
INR	Index to particular operation with activity consumables data base.
DEL	RCS burn time.

Input - The CREND routine requiring the following input data accessed through COMPOOL:

J	ACTION identifier.
K	Activity number.
AT(K,I)I=1,5	Entry data array for activity.

The CREND routine requires the following input data accessed through CADB:

K(J,CNUM)	Number of operations in the preparation period of the activity.

Processing - The flow diagram of the CREND routine is shown in Figure 44.

Output - The CREND routine transmits the following data through the COMPOOL:

KRATE(CNUM,INR)	Factor to multiply ΔRATE by.
AT(K,I)I=1,5	Entry data array for activity.

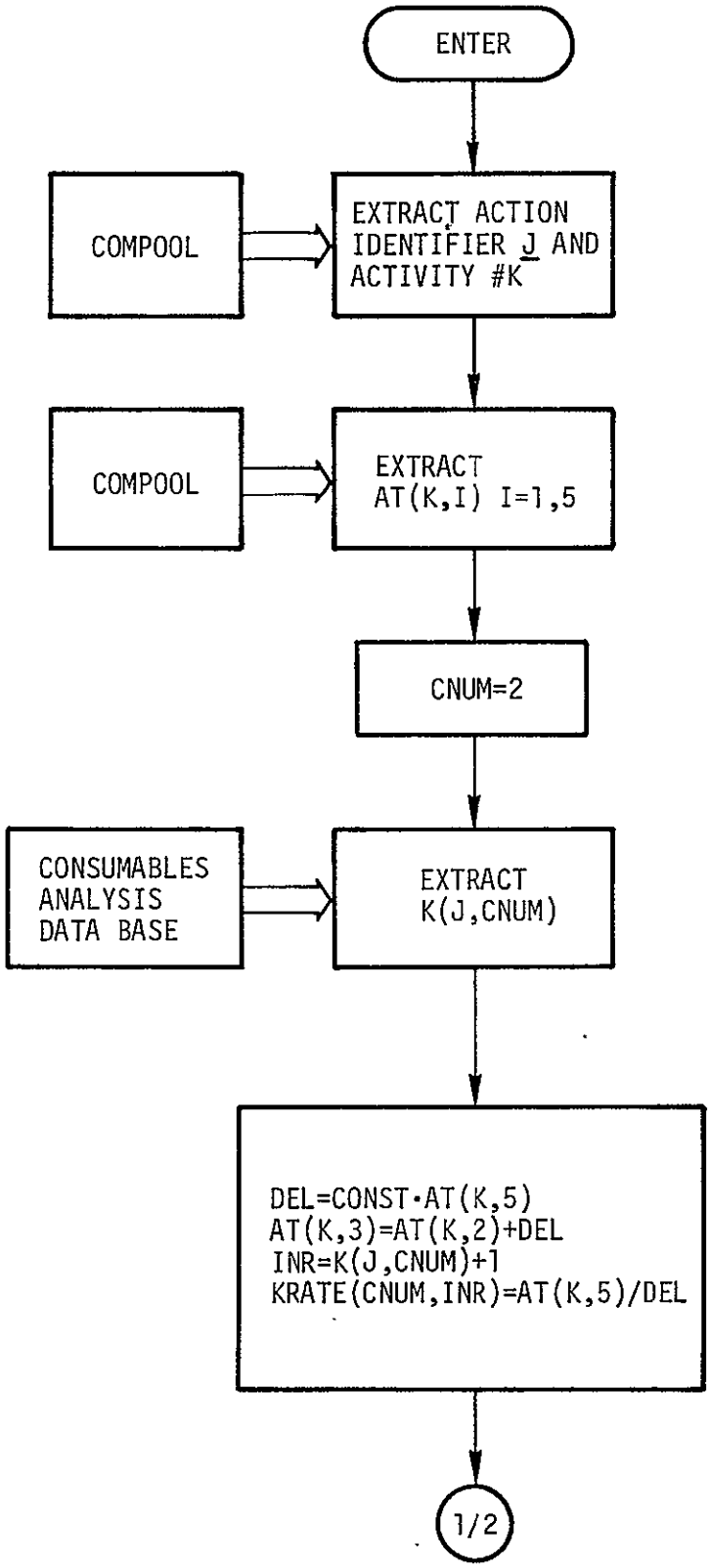


Figure 44. Flow Diagram for the CREND Routine

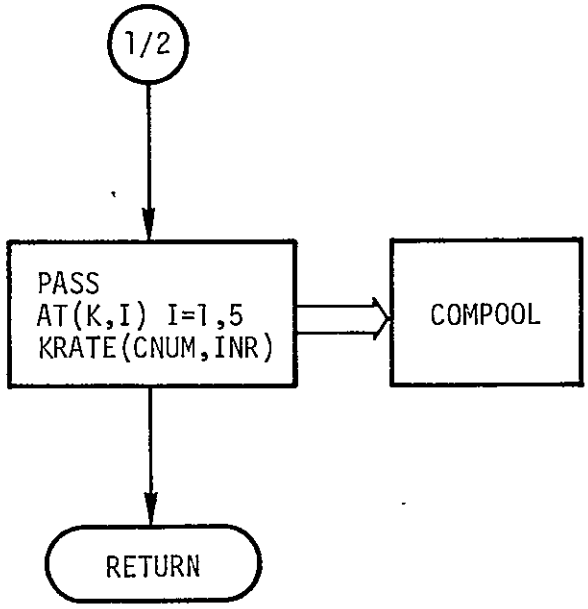


Figure 44. Concluded

6.6 CDOCK Routine

Description - The CDOCK routine calculates the burn time, activity reference start time, and the acceleration rate for a docking maneuver.

Interface

I/O DEVICE - None.
DATA BASE - COMPOOL for both input and output, Consumables Analysis Data Base (CADB) for input only.
ROUTINES CALLING CDOCK - SPECIAL.
ROUTINES CALLED BY CDOCK - None.

Internal Variables

CNUM	Index to consumable affected by activity.
INR	Index to particular operation with activity consumables data base.
DEL	RCS burn time.

Input - The CDOCK routine requiring the following input data accessed through COMPOOL:

J	ACTION identifier.
K	Activity number.
AT(K,I)I=1,5	Entry data array for activity.

The CDOCK routine requires the following input data accessed through CADB:

K(J,CNUM)	Number of operations in the preparation period of the activity.

Processing - The flow diagram of the CDOCK routine is shown in Figure 45.

Output - The CDOCK routine transmits the following data through the COMPOOL:

KRATE(CNUM,INR)	Factor to multiply ΔRATE by.
AT(K,I)I=1,5	Entry data array for activity.

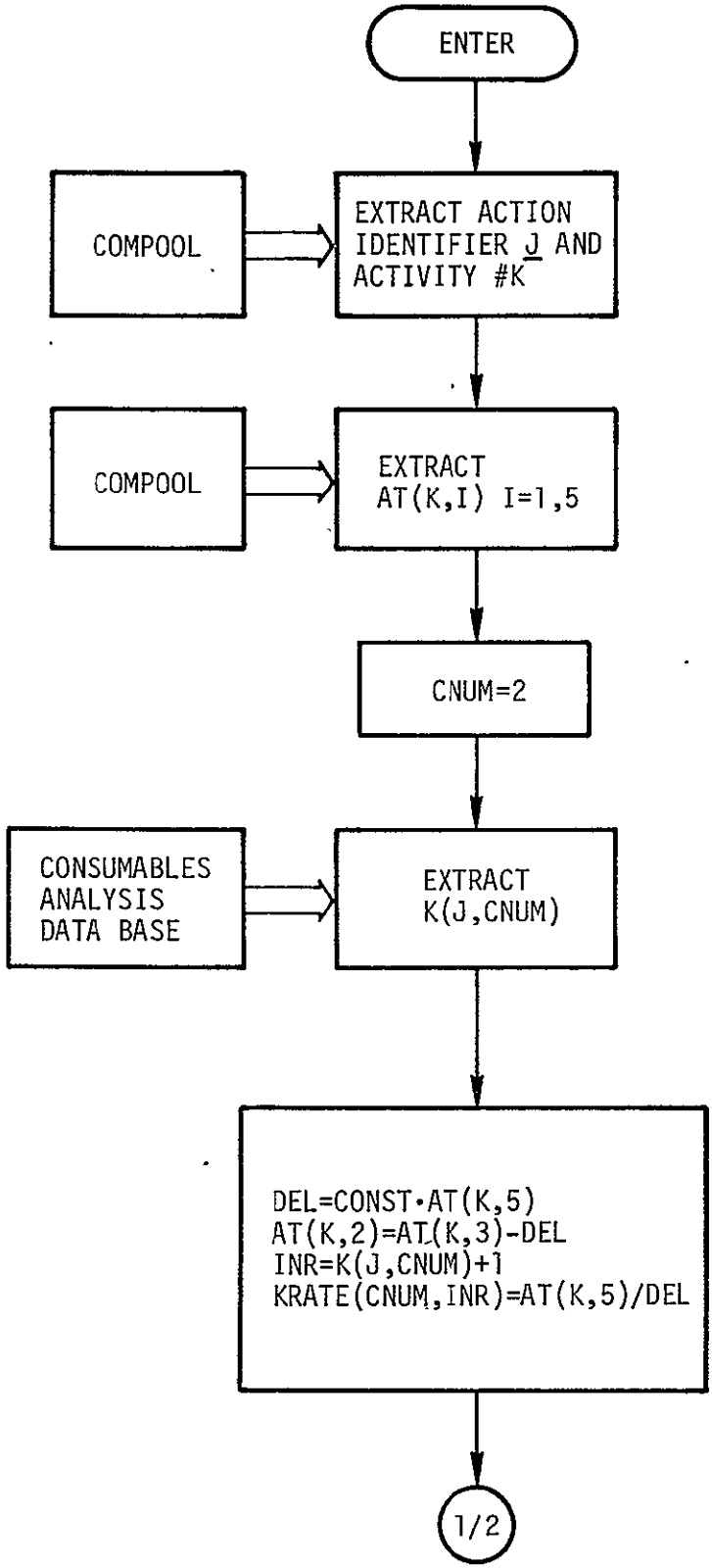


Figure 45. Flow Diagram for the CDOCK Routine

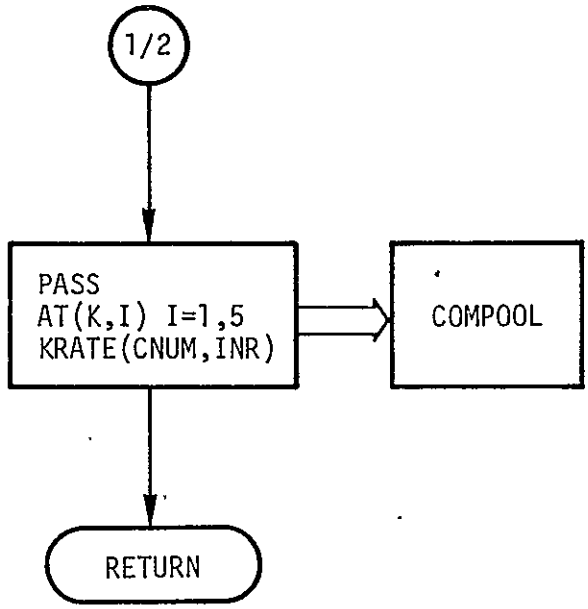


Figure 45. Concluded

6.7 CUDOCK

Description - The CUDOCK routine calculates the burn time, activity reference stop time, and the acceleration rate for an undocking maneuver.

Interface

I/O DEVICE - None.
DATA BASE - COMPOOL for both input and output, Consumables Analysis Data Base (CADB) for input only.
ROUTINES CALLING CUDOCK - SPECIAL.
ROUTINES CALLED BY CUDOCK - None.

Internal Variables

CNUM	Index to consumable affected by activity.
INR	Index to particular operation with activity consumables data base.
DEL	RCS burn time.

Input - The CUDOCK routine requiring the following input data accessed through COMPOOL:

J	ACTION identifier.
K	Activity number.
AT(K,I)I=1,5	Entry data array for activity.

The CUDOCK routine requires the following input data accessed through CADB:

K(J,CNUM)	Number of operations in the preparation period of the activity.

Processing - The flow diagram of the CUDOCK routine is shown in Figure 46.

Output - The CUDOCK routine transmits the following data through the COMPOOL:

KRATE(CNUM,INR)	Factor to multiply ΔRATE by.
AT(K,I)I=1,5	Entry data array for activity.

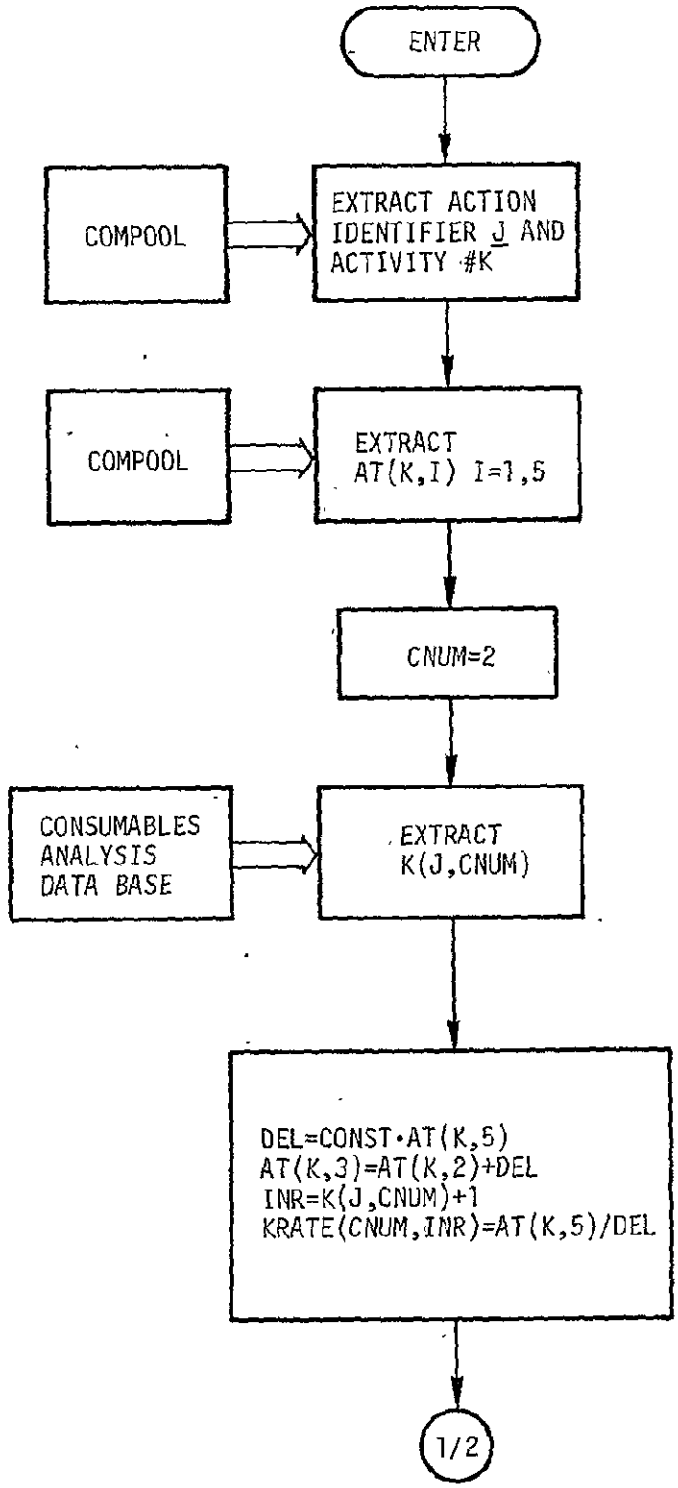


Figure 46. Flow Diagram for the CUDOCK Routine

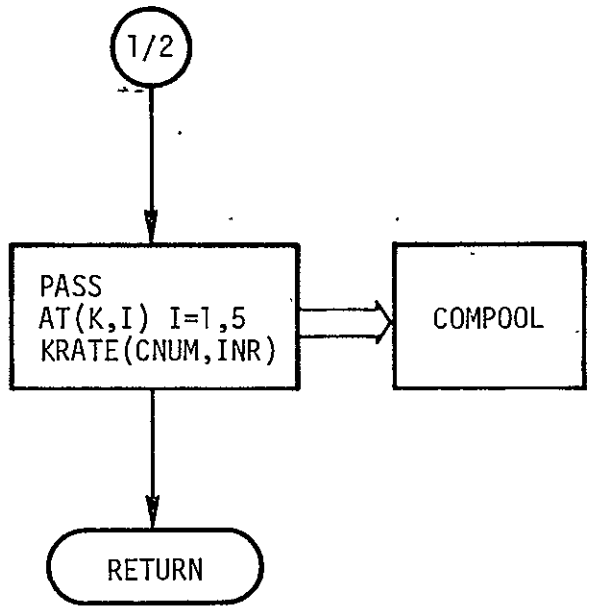


Figure 46. Concluded

6.8 CEVA Routine

Description - The CEVA routine calculates the influence of the EVA team size on the consumables rates resident in the consumables analysis data base for the EVA activity. Appropriate factors to be applied to the resident rates are made available for subsequent processing by the Control and Support subroutines.

Interface

I/O DEVICE - None.
DATA BASE - COMPOOL for both input and output, Consumables Analysis Data Base (CADB) for input only.
ROUTINES CALLING CEVA - SPECIAL.
ROUTINES CALLED BY CEVA - None.

Internal Variables

CNUM	Index to consumable affected by activity.
INR	Index to particular operation within activity consumables data base.
II(1)	Internal indexing of K(J,CNUM).
II(2)	Internal indexing of J(J,CNUM).
II(3)	Internal indexing of L(J,CNUM).
JJ	Operations counter.
I	Operations counter index.
KK	Flag to skip particular operation.

Input - The CEVA routine requiring the following input data accessed through COMPOOL:

J	ACTION identifier.
K	Activity number.
AT(K,I)I=1,5	Entry data array for activity.

The CEVA routine requires the following input data accessed through CADB:

K(J,CNUM)	Number of operations in the preparation period of the activity.

C.3

J(J,CNUM) Number of operations in the activity period.

L(J,CNUM) Number of operations in the post-activity period.

Processing - The flow diagram of the CEVA routine is shown in Figure 47.

Output - The CEVA routine transmits the following data through the COMPOOL:

KRATE(CNUM,INR) Factor to multiply ΔRATE by.

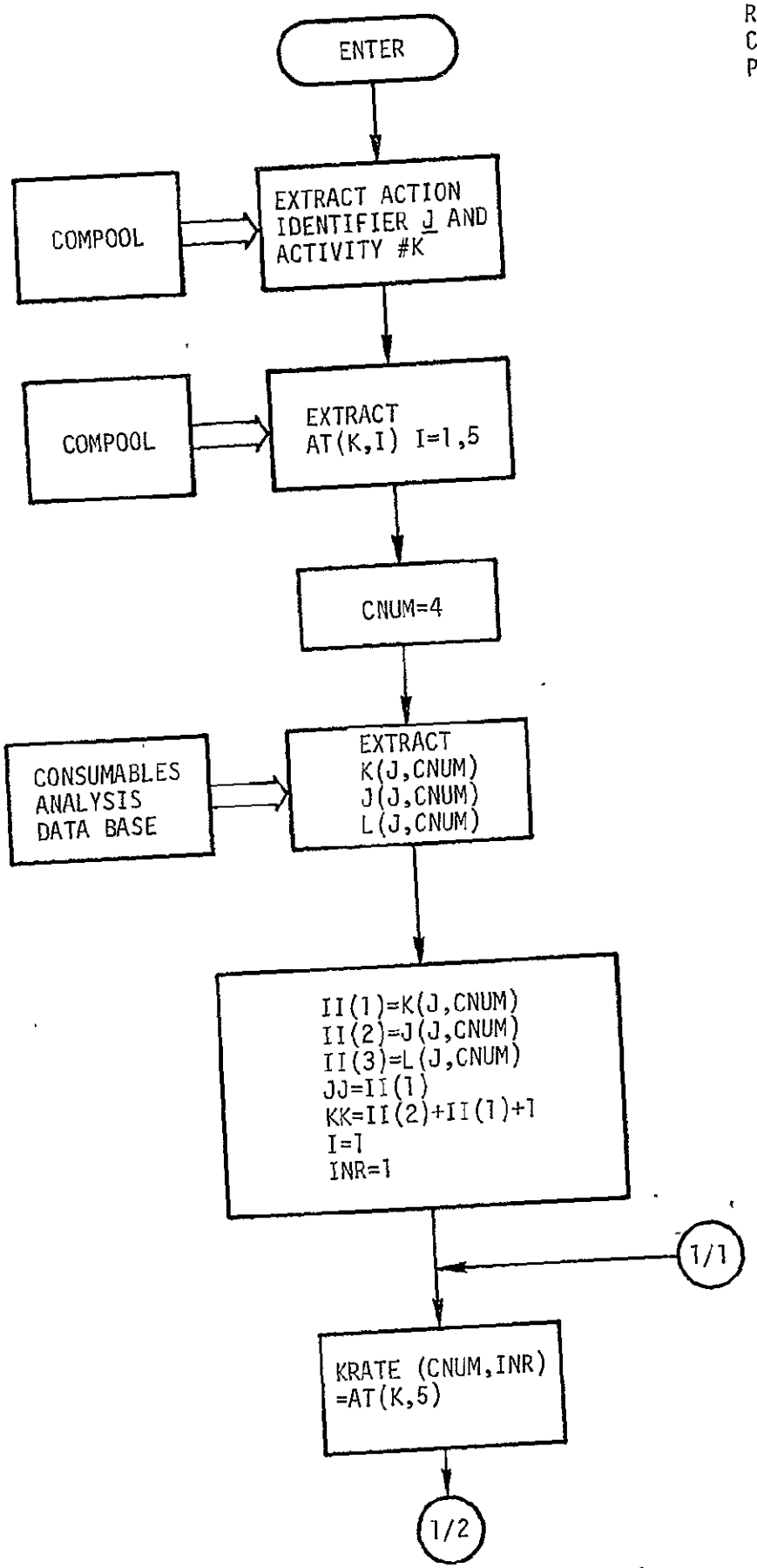


Figure 47. Flow Diagram for the CEVA Routine

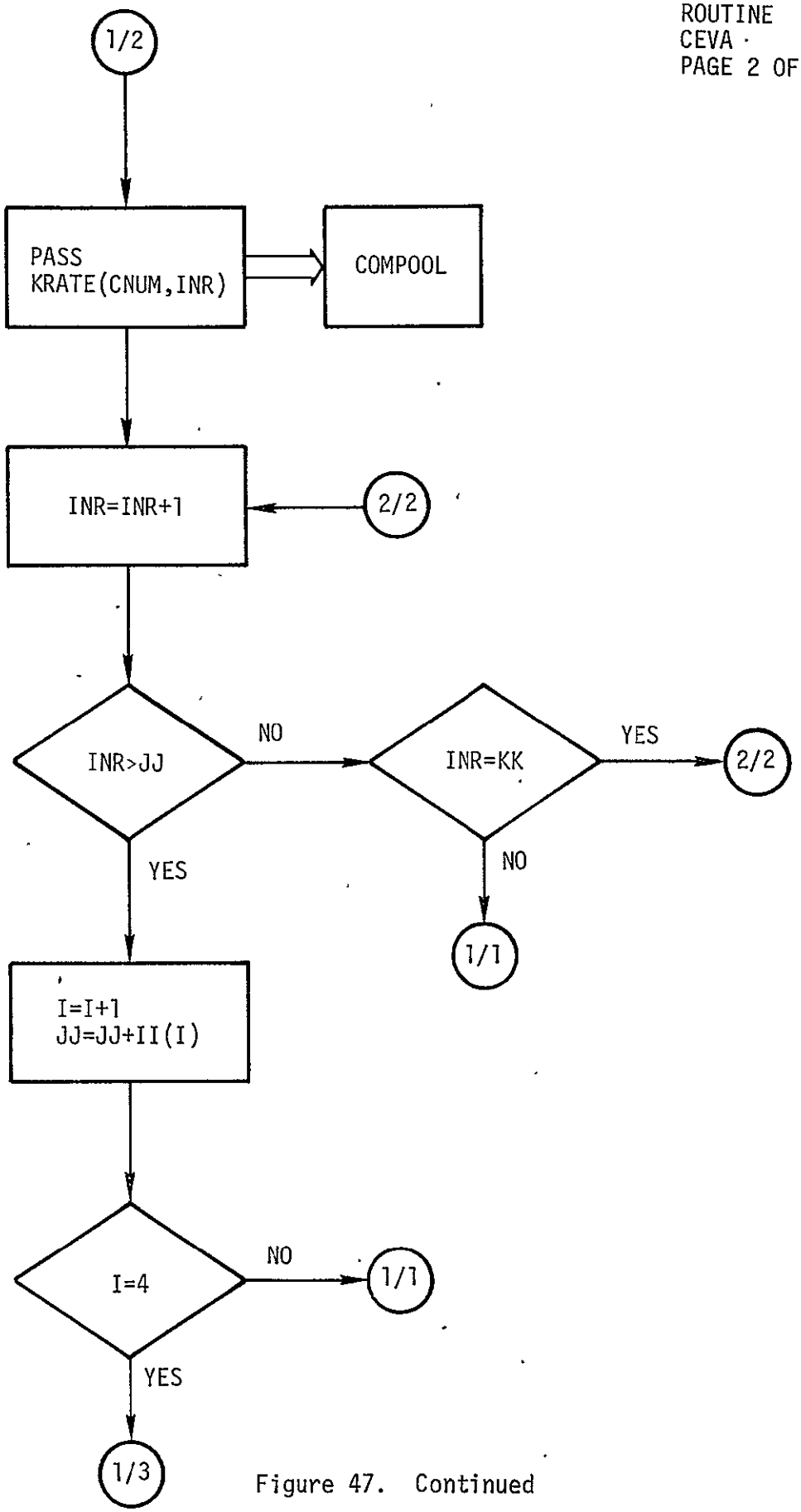


Figure 47. Continued

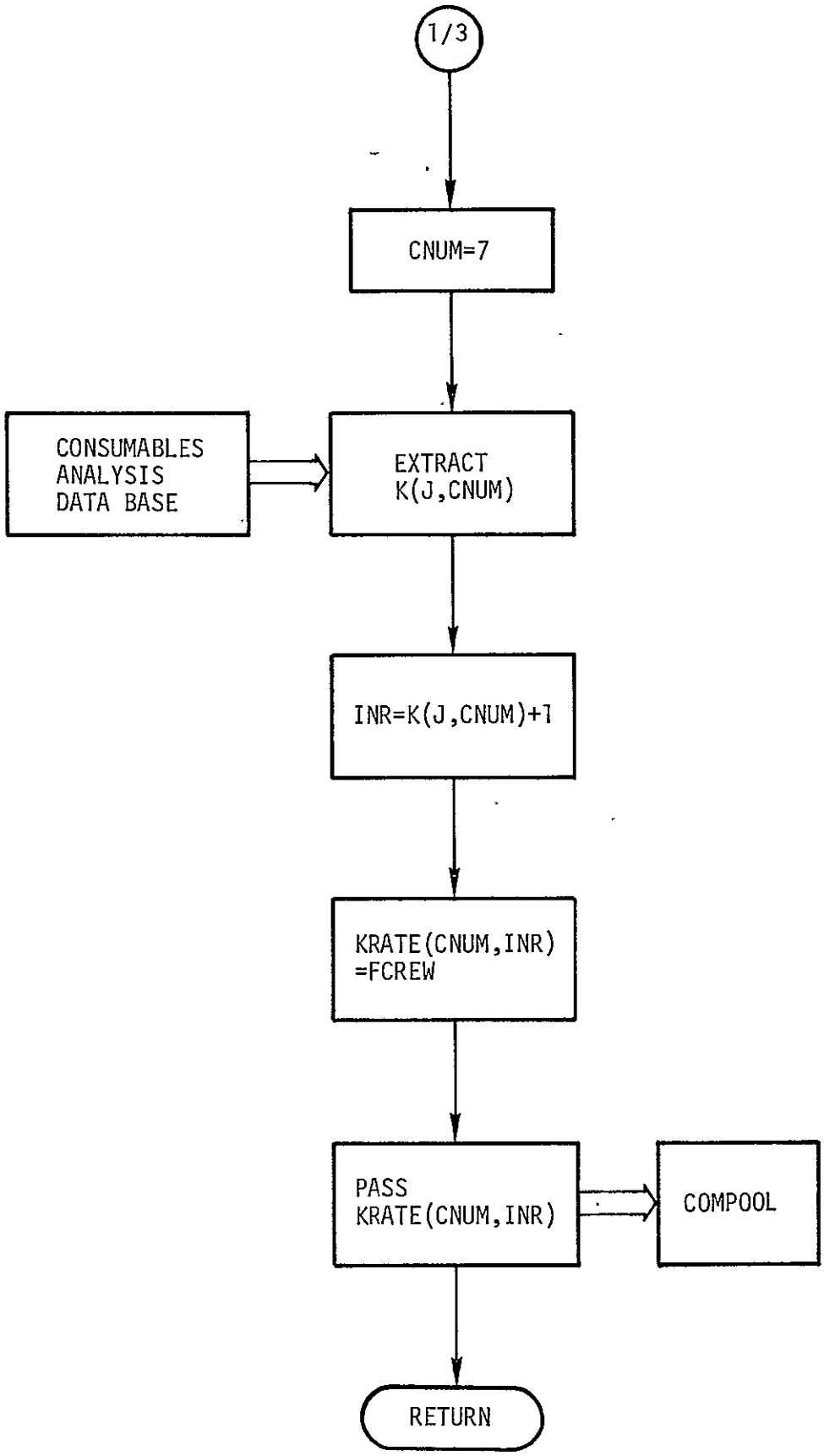


Figure 47. Concluded

6.9 CIVA Routine

Description - The CIVA routine calculates the influence of the IVA team size on the consumables rates resident in the consumables analysis data base for the IVA activity. Appropriate factors to be applied to the resident rates are made available for subsequent processing by the Control and Support subroutines.

Interface

I/O DEVICE - None.
DATA BASE - COMPOOL for both input and output, Consumables Analysis Data Base (CADB) for input only.
ROUTINES CALLING CIVA - SPECIAL.
ROUTINES CALLED BY CIVA - None.

Internal Variables

CNUM	Index to consumable affected by activity.
INR	Index to particular operation within activity consumables data base.

Input - The CIVA routine requiring the following input data accessed through COMPOOL:

J	ACTION identifier.
K	Activity number.
AT(K,I)I=1,5	Entry data array for activity.

The CIVA routine requires the following input data accessed through CADB:

K(J,CNUM)	Number of operations in the preparation period of the activity.

Processing - The flow diagram of the CIVA routine is shown in Figure 48.

Output - The CIVA routine transmits the following data through the COMPOOL:

KRATE(CNUM,INR)	Factor to multiply ΔRATE by.

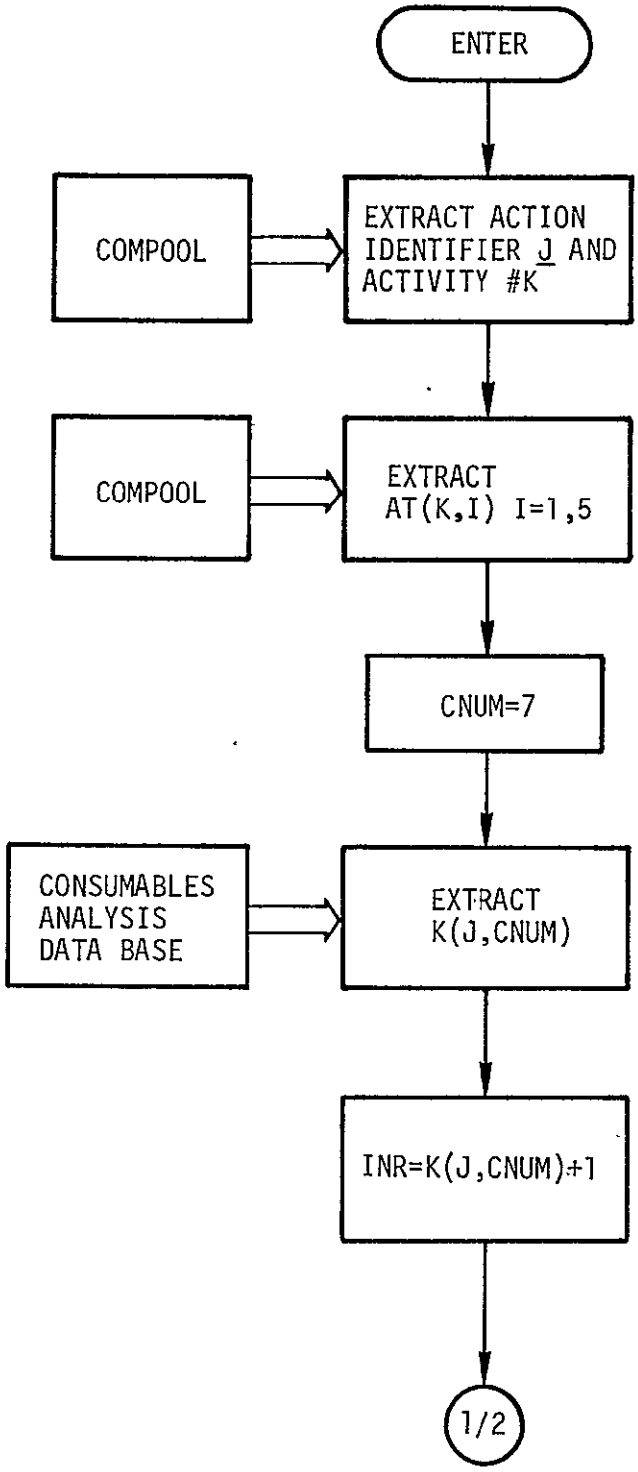


Figure 48. Flow Diagram for the CIVA Routine.

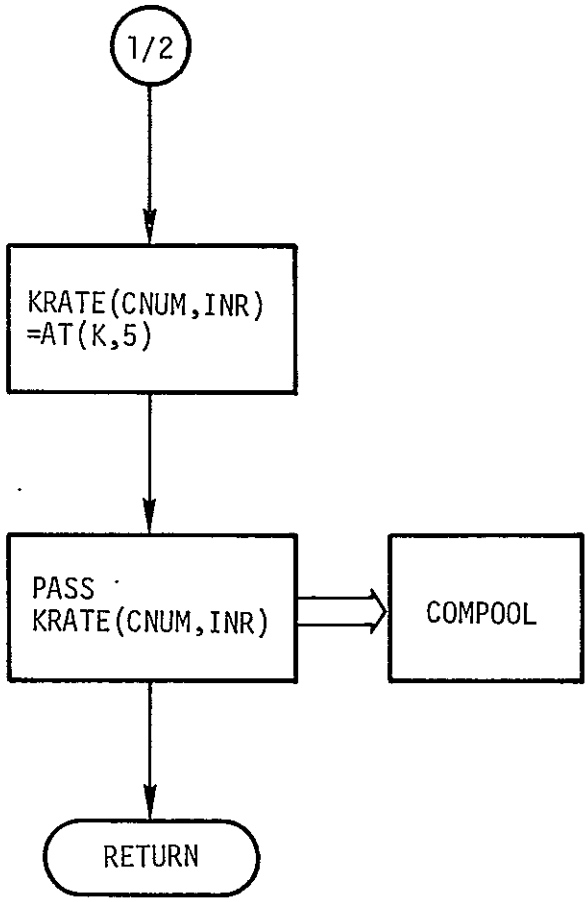


Figure 48. Concluded

6.10 CES Routine

Description - The CES routine calculates the influence of the number of crew members sleeping or eating on the consumables rates resident in the consumables analysis data base for the eat or sleep activity. Appropriate factors to be applied to the resident rates are made available for subsequent processing by the Control and Support subroutines.

Interface

I/O DEVICE - None.
DATA BASE - COMPOOL for both input and output, Consumables Analysis Data Base (CADB) for input only.
ROUTINES CALLING CES - SPECIAL.
ROUTINES CALLED BY CES - None.

Internal Variables

CNUM	Index to consumable affected by activity.
INR	Index to particular operation within activity consumables data base.

Input - The CES routine requiring the following input data accessed through COMPOOL:

J	ACTION identifier.
K	Activity number.
AT(K,I)I=1,5	Entry data array for activity.

The CES routine requires the following input data accessed through CADB:

K(J,CNUM)	Number of operations in the preparation period of the activity.

Processing - The flow diagram of the CES routine is shown in Figure 49.

Output - The CES routine transmits the following data through the COMPOOL:

KRATE(CNUM,INR)	Factor to multiply ΔRATE by.

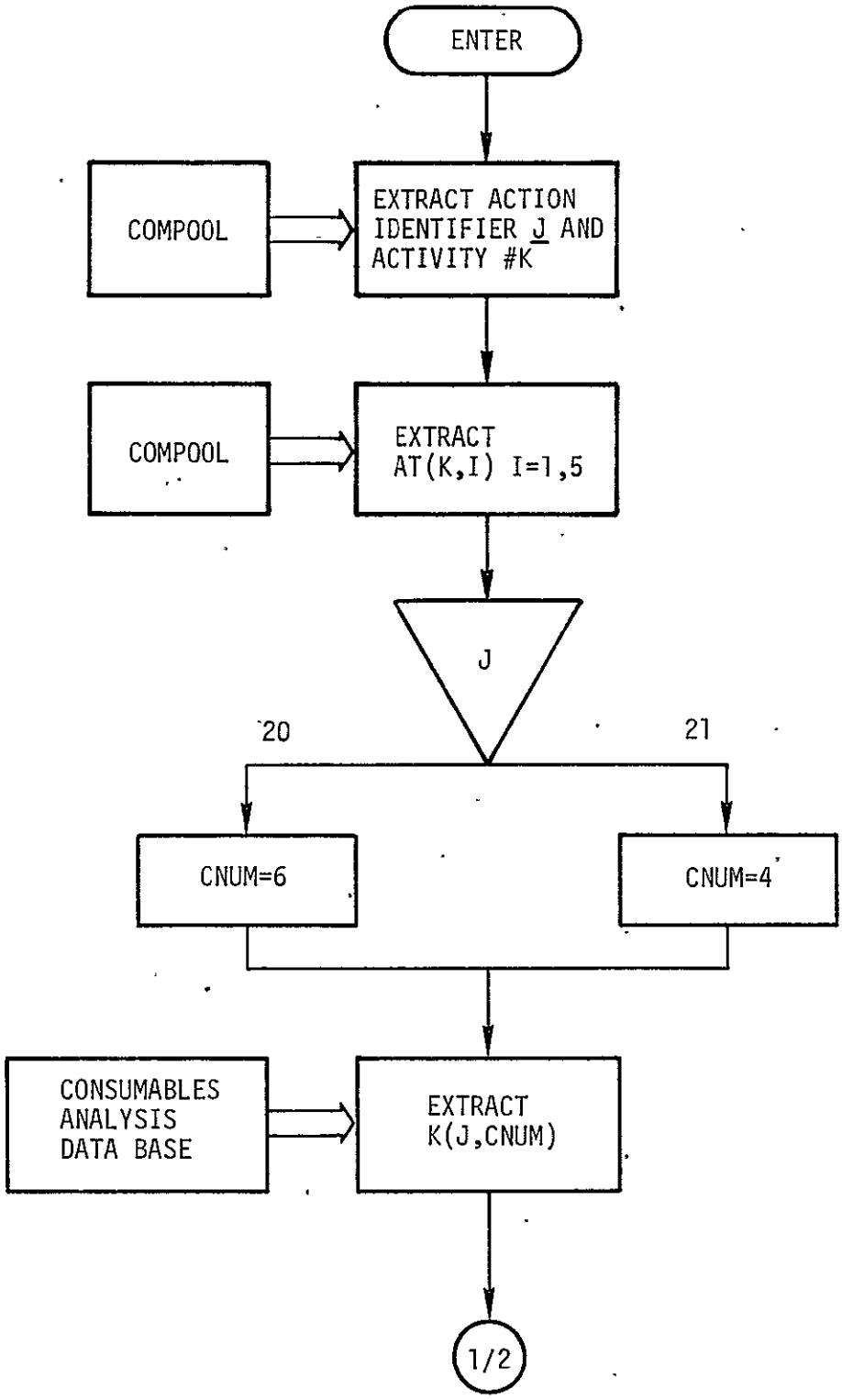


Figure 49. Flow Diagram for the CES Routine

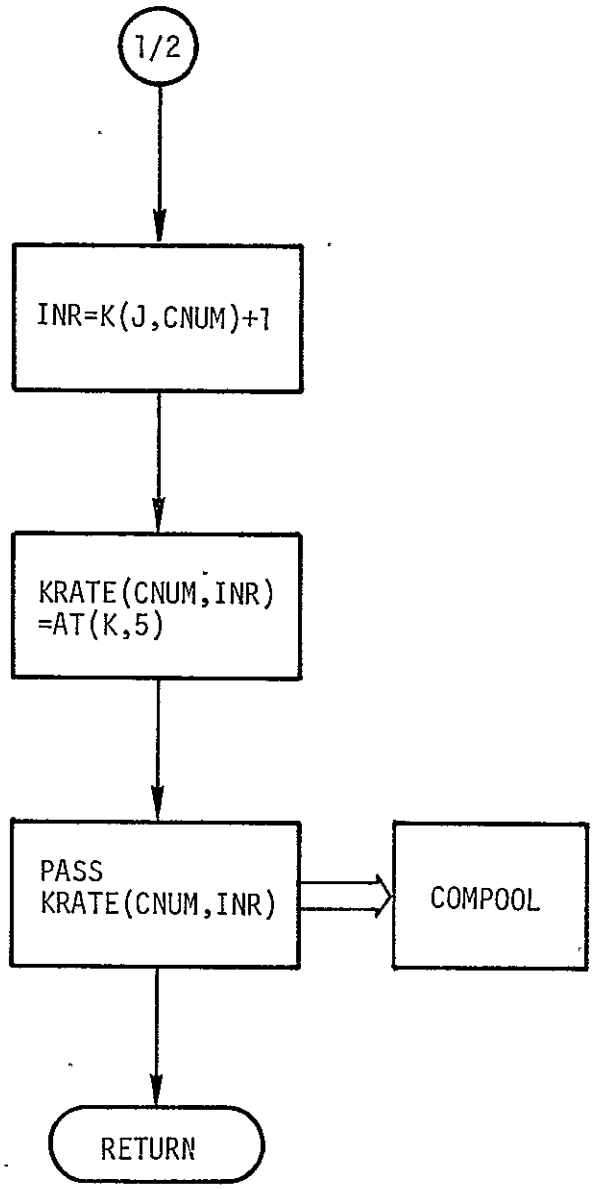


Figure 49. Concluded

6.11 CWM Routine

Description - The CWM routine calculates the influence of the crew size on the consumables rates resident in the consumables analysis data base for the Waste Management activity. Appropriate factors to be applied to the resident rates are made available for subsequent processing by the Control and Support subroutines.

Since the Waste Management activity has not been defined at this time, this routine is TBD.