CCTC

DEFENSE COMMUNICATIONS AGENCY

COMPUTER SYSTEM MAN
CSM MM 9-77
VOLUME II
1 JUNE 1977

12

COMMAND & CONTROL TECHNICAL CENTER

THE CCTC QUICK-REACTING GENERAL WAR GAMING SYSTEM (QUICK)

VOLUME II
WEAPON/TARGET IDENTIFICATION SUBSYSTEM

PROGRAM MAINTENANCE MANUAL

COMMAND AND CONTROL TECHNICAL CENTER

Computer System Manual CSM MM 9-77

1 June 1977

THE CCTC QUICK-REACTING GENERAL WAR GAMING SYSTEM

(QUICK)

Program Maintenance Manual

Volume II - Weapon/Target Identification Subsystem

SUBMITTED BY:

C. C. THOMPSON
Project Officer

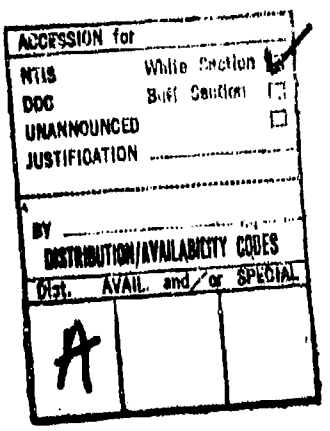
APPROVED BY:

R. E. HARSHBARGER
Acting Deputy Director
NMCS ADP

See Form 1473

ACKNOWLEDGMENT

This document was prepared under the direction of the Chief for Military Studies and Analyses, CCTC, in response to a requirement of the Studies, Analysis, and Gaming Agency, Organization of the Joint Chiefs of Staff. Technical support was provided by System Sciences, Incorporated under Contract Number DCA100-75-C-0019.



CONTENTS

ILLUSTRATIONS

TABLES

ABSTRACT

The computerized Quick-Reacting General War Gaming System (QUICK) will accept input data, automatically generate global strategic nuclear war plans, provide output summaries, and produce input tapes to simulator subsystems external to QUICK. QUICK has been programmed in FORTRAN for use on the CCTC HIS 6000 computer system.

The QUICK Program Maintenance Manual consists of four volumes: Volume I, Data Management Subsystem; Volume II, Weapon-Target Identification Subsystem; Volume III, Weapon Allocation Subsystem; Volume IV, Sortie Generation Subsystem. The Program Maintenance Manual complements the other QUICK Computer System Manuals to facilitate maintenance of the war gaming system. This volume, Volume II, provides the programmer/analyst with a technical description of the purpose, functions, general procedures, and programming techniques applicable to the modules and subroutines of the Weapon/Target Identification subsystem. Companion documents are:

a. USERS MANUAL
   Computer System Manual CSM UM 9-77, Volume I
   Computer System Manual CSM UM 9-77, Volume II
   Computer System Manual CSM UM 9-74, Volume III
   Computer System Manual CSM UM 9-74, Volume IV
   Provides detailed instructions for applications of the system

b. TECHNICAL MEMORANDUM
   Technical Memorandum TM 153-77
   Provides a nontechnical description of the system for senior management personnel

SECTION 1. GENERAL

1.1 Purpose

This volume of the QUICK Program Maintenance Manual describes the modules which are part of the QUICK Weapon/Target Identification subsystem, detailing the modules, subroutines, and functions which it comprises. The information contained herein is presented on a module-by-module basis. The module-by-module discussions are structured so that a maintenance programmer can understand the program functions and programming techniques. The computer subjects are structured to inform the maintenance programmer of overall system programming techniques and conventions.

Subsequent subsections present general descriptions of the overall QUICK system and Weapon-Target Identification subsystem.

1.2 General Description

The Weapon/Target Identification subsystem of QUICK selects and processes the Red and/or Blue forces which are prespecified for a particular plan. The subsystem consists of modules JLM, DBMOD, INDEXER, and PLANSET, as shown in figure 1. Figure 2 shows the relationship of the Weapon/Target Identification subsystem to other QUICK subsystems in terms of procedural and information flow.

The modules of this subsystem are used to assemble selected target data from the CCTC JAD files, and reformat the data in a manner which is acceptable to QUICKs Integrated Data Base and to further develop a plan for allocation.

Modules within this subsystem are executed in the order of: JLM, DBMOD, INDEXER, and PLANSET. All modules perform updates to the Integrated Data Base; no other data files are used (other than internal temporary scratch files).

The first module, JLM, builds the target portion of the data base. Note that the remaining data base is created by modules within the Data Management subsystem. These modules may be executed at any stage of the entire QUICK processing, i.e., before or after INDEXER, etc. An order of module execution pertains only to modules not defined within the Data Management subsystem.

The next module normally run is DBMOD. Its primary purpose is to alter the content or characteristics of a data base to the specific scenario for which the plan is being developed, in accordance with prespecified user input.

Module INDEXER is designed to assign index numbers (attribute INDEXNO) and perform the task of forming complex targets.

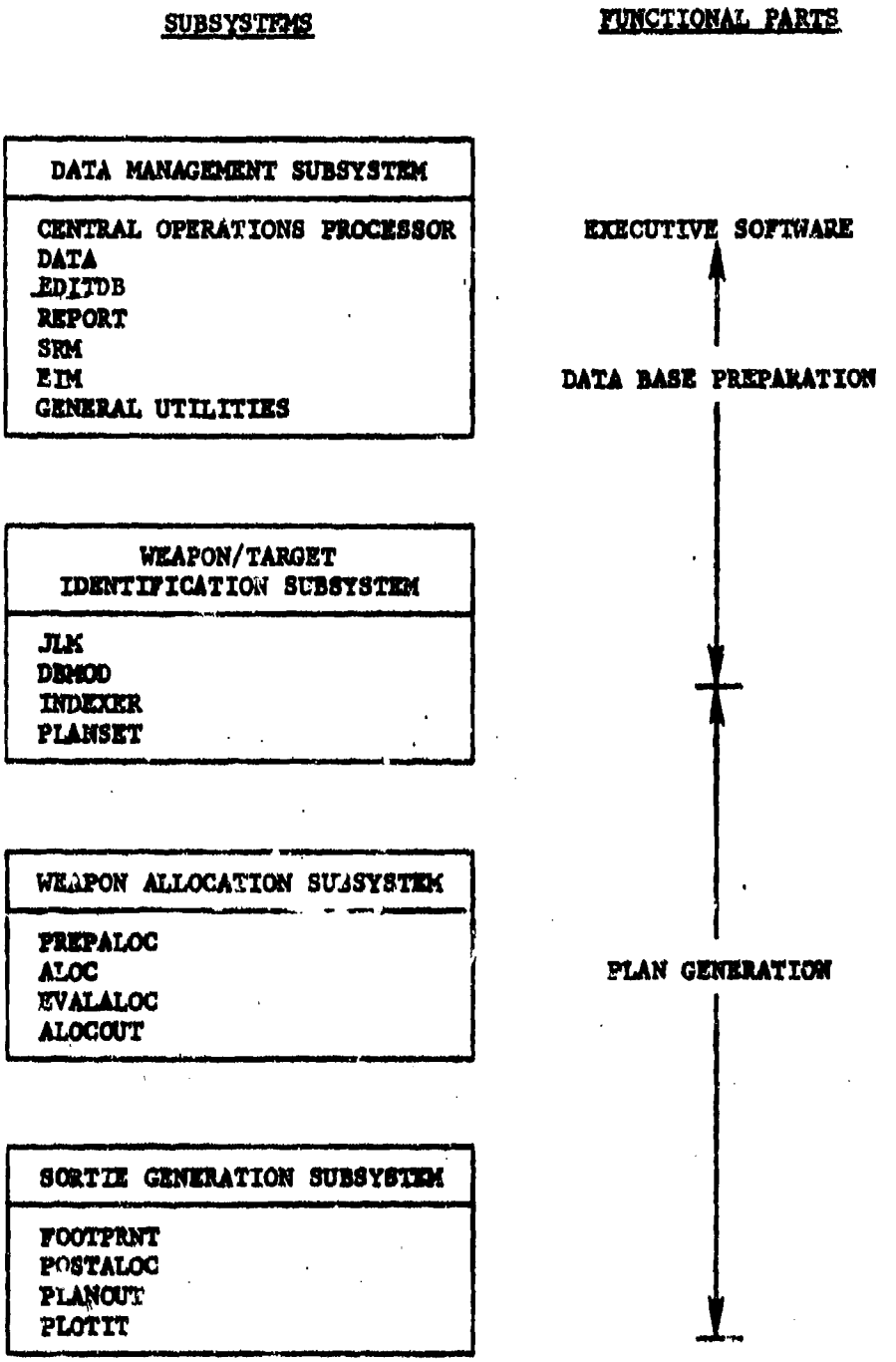


Figure 1. Major Subsystems of the QUICK System

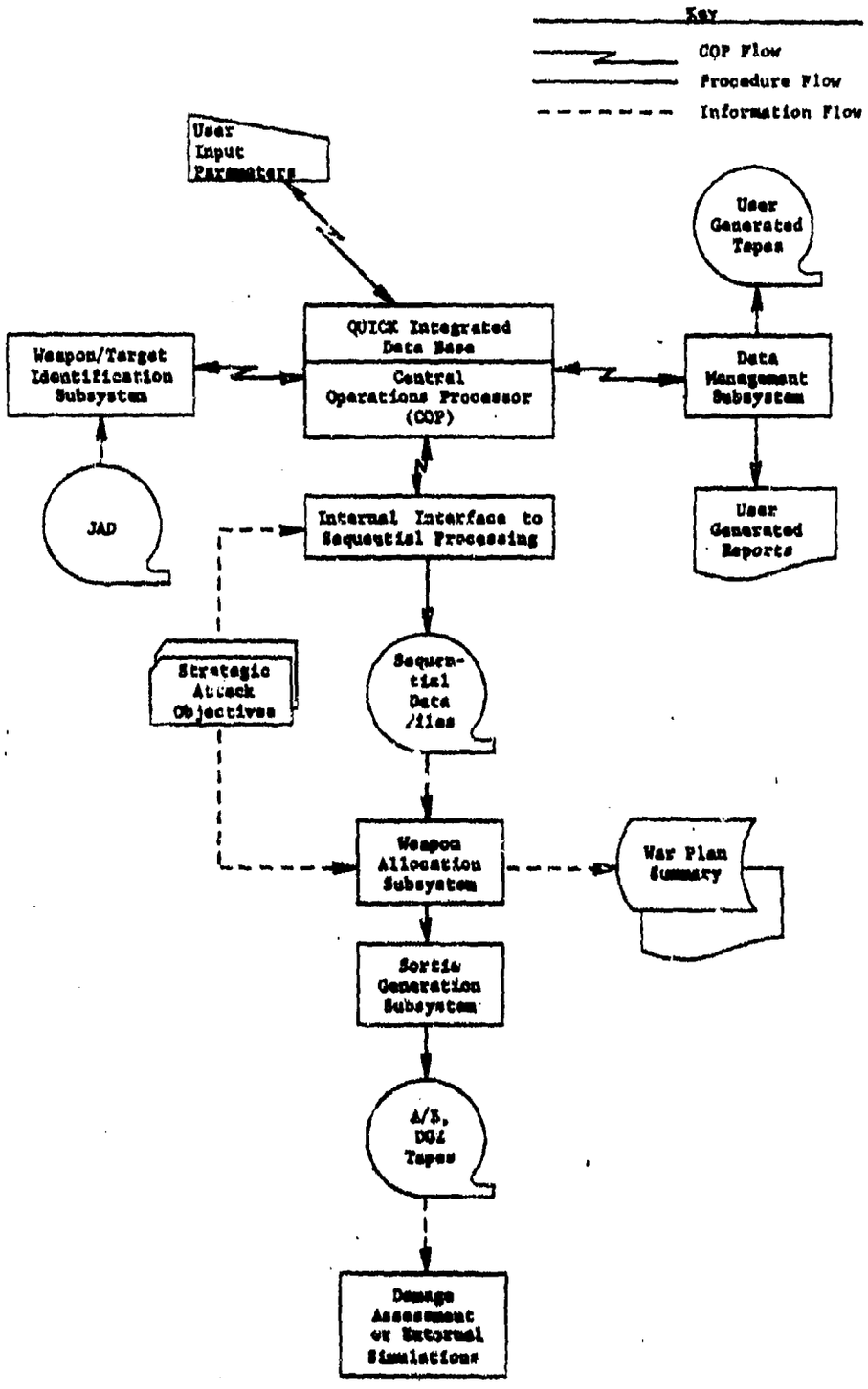


Figure 2. Procedure and Information Flow in QUICK/HIS 6000

Module PLANSET forms weapon groups, prepares the target list for the allocator, computes and normalizes the class value factors and calculates the representative attributes for complex targets.

1.3 Organization of Maintenance Manual, Volume II

Each major section of this manual details a module along with the subroutines and functions which comprise the module. Major subsections are:

a. Module input - details what chains must be created prior to module execution

b. Module output - details what chains will be updated by each module

c. Functional description - details the macro function of the module and the associated major subroutines

d. Common blocks - detail the contents of all internal common blocks. All common blocks used to communicate with the COP are given in Program Maintenance Manual, Volume I, appendix A. These are: C10, C15, C20, C30, C40, C50, ERRCOM, INS, UPQT, OOPS, STRING

Within the QUICK system the COP is viewed as the operating program. Based on user direction, the COP will execute overlay links or modules which perform the objectives of the user requests. Each overlay link is called through knowledge of the command verb and within each link the first subroutine is called ENTMOD (for entry module). That is, there are as many subroutines called ENTMOD as there are modules. Confusion is avoided by executing the correct overlay link. Subroutine discussion, then, is initiated with ENTMOD whose meaning, or function, varies according to the overlay link.

Comments on the QUICK integrated data base can be found in Program Maintenance Manual, Volume I, section 2. It will be assumed within this manual that the reader has an understanding of QUICK's data base.

SECTION 2. JAD LOADING MODULE (JLM)

2.1 Purpose

JLM and its associated subroutines assemble data from the CCTC JAD files and manipulate and reformat the data in a structure which is acceptable to the QUICK system.

2.2 Input

JLM creates target records as directed by user card image input. This module normally is the first phase in the total creation of the integrated data base. For proper execution, the organizational portion of the data base must have been finalized.

Target records are obtained from input JAD files according to the format given in figure 3. Not all JAD entries are used by JLM, only those listed under the column labeled 'USED'.

A JAD format has a maximum of 336 characters of which only the first 258 have defined inputs. In those cases when JLM generates a JAD format tape, entries are placed within characters 289 through 336 (the third column of figure 3).

2.3 Output

JLM builds that portion of the organizational structure of the data base called the Assignment Table. This table shows the kind of target to be added to the data base and how it will be included. The table includes:

- The valid country code and what region and side the country is;
- The target classes for each side;
- The selection criteria for each target type based on category code, owner, location, and capacity or name;
- The TASK that corresponds to the target types; and
- The list of DESIGs (alphabetic portion) that are to be used.

From the Assignment Table, the user selects targets from the JAD input file and creates targets (record TARGET) within the gaming portion of the integrated data base. For each created target, all linkage is properly updated.

COLS.	ITEM	USED*	CREATED BY JLM
1-5	Category Code	JCATCODE	
6-9	WAC No.	JWACNO	
10-15	BE No.	JBENO	
16-20	Blank		
21-58	Name	JNAME(21-26)	
59-64	Major number	JMAJOR	
65-88	Complex Name		
89-94	Minor number	JMINER	
95-118	Concentration Name		
119-125	Latitude DDMMSS N/S	JLAT	
126-133	Longitude DDDMMSS E/W	JLONG	
134-135	World Division		
136-137	Sub Div		
138-139	Country Location	JLOC	
140-141	Special Region	JAD14	
142-143	Region		
144-147	Blank		
148-149	Owner Country	JOWN	
150-151	Agency or Service owner		
152-153	U & S Cmd or Supreme All		
154-155	Component or All Regn CMD		
156-159	Severe VN		
160-163	Moderate VN	JVN	
164-167	Light VN		
168-171	Review Data yy mm		
172-175	ICOD yy mm		
176-190	Significance		
191-198	Capacity	JCAP	
199-200	Data Source		
201-204	Units of measure		

*Variables are named as used in common block JADREC.

Figure 3. JAD Format (Part 1 of 2)

COLS.	ITEM	USED#	CREATED BY JLM
205	Scaling Factor		
206-208	Radius	JRADIUS	
209-212	Percent Capacity		
213-224	Dimensions		
225-236	Fiscal Year Projections		
237	File ID Code		
238	Phase Code		
239-245	Security Class		
246-247	Remark		
248-253	Owner UIC		
254-255	Serv Spcl Code		
256-258	READY Code		
259-267	Blank		
268-288	Not Used		
289-293	DESIG		DESIG
294	Flag if in the Data Base		* or blank
295-300	Type		TYPE
301-303	Not Used		
304-306	Subset of Class Index		*
307-308	Not Used		
309-312	Sequential Count Within Subset		*
313-314	Task		TASK
315-318	Not Used		
319	QUICK Region		IREG
320	SAGA Region		*
321-324	Not Used		
325-330	SAGA Flag		
331-336	BLANK		

#Variables are named as used in common block JADREC

Figure 3. (Part 2 of 2)

An optional output of JLM is a JAD format file from the selected targets for use by damage assessment systems external to QUICK. The format is as shown in figure 3 with the added entries created by JLM.

2.4 Concept of Operation

The function of the JLM is to build portions (targets) of the integrated data base by selecting records from a file that is in a JAD format. JLM operates in three modes. First, a section of the integrated data base called the Assignment Table is built through user inputs. This table describes what sort of target is to be added to the data base and how it will be included. Second, given a completed Assignment Table, the selection of JAD records is executed and a Damage Assessment tape is prepared for use in processors external to the QUICK system. Third, after record selection, provisions are included for deleting individual records not required for QUICK processing. In the text English sense verbs ASSIGN, SELECT, and ASTERISK initiate the three JLM functions.

2.5 Identification of Subroutine Functions

2.5.1 Subroutine ASSIGN. This subroutine is the first subroutine within overlay link ASSI executed upon the appearance of verb ASSIGN. Depending upon the adverb; subroutines ALPHAS and/or PLAYERS are executed.

2.5.1.1 Subroutine ALPHAS. The ALPHAS clause builds the bulk of the Assignment Table and is performed by this subroutine. The main portion of subroutine ALPHAS involves the reading and correct definition of the generalized input clause. This clause is fully detailed within Section 2 of Users Manual, Volume II. Major points are repeated here for purposes of outlining the major thrust of the code written for subroutine ALPHAS. The input clause has the form:

ASSIGN ALPHAS side // class - type $\geq$ [minimum-quantity / blank]

/ low-catcode [- high-catcode] [* task] ,

design-alphabetic [, alternate-design . . .]

[[NOT] { OWNED BY / LOCATED IN } country-code [, country-code . . .]]

General comments are:

- o The side must be first
- o The target class must be preceded by two slashes
- o The target type must be preceded by a dash
- o If minimum capacity or name is used, it must be preceded by a greater than symbol
- o The lowest catcode must be preceded by one slash and if a range of catcodes are used the highest catcode is preceded by a dash
- o TASK is preceded by an asterisk and DESIG by a comma
- o Country codes are preceded by either OWNED or IN if the assignment is restricted

2.5.1.2 Subroutine PLAYERS. Similar to ALPHAS, subroutine PLAYERS reads the input clause and updates the Assignment Table. The generalized input command is:

ASSIGN PLAYERS side // region / country-code

[, country-code . . .] [side // region . . .]

The side must be first, the region must be preceded by two slashes and the list of country codes must be preceded by a single slash.

2.5.2 Subroutine SELECT. The use of the SELECT verb (and hence the execution of overlay link (SELE)) instructs the JLM to select records from the JAD format input file according to the developed Assignment Table. The SELECT command has a maximum of six optional adverbs and are:

- o WHERE - normal WHERE clause without OF or LIKE
- o UNIT - used to define input unit if it is not 20
- o ONPRINTS - causes the print of the output JAD format
- o REPLACING or OMITTING - used to replace existing targets or to ignore duplicates
- o ORDER - allows the user to specify the arrangement that the classes will be added to the integrated data base
- o SETTING - used to set the value of TARDEF to allow for automatic assignment of values for attributes TARDEFNI and TARDEFLO

After user input definition, SELECT reads a JAD input record and queries the Assignment table (subroutine FNDTAR) to ascertain if the target should be added to QUICKs data base. If it is not to be added, the next JAD input record is read. Otherwise investigations are made for exclusion of the input record due to a WHERE clause. For each selected target record, data is written onto files 25 and 21.

After all target records have been selected, data files are sorted and read. For each record, subroutine ADTOBASE is called for the definition of the selected target record onto QUICKs data base.

Finally tests and code are made to guarantee the JAD records are in proper sort and additionally JAD selected records are printed if user directed.

2.5.2.1 Subroutine ADTOBASE. This subroutine adds the data from the selected JAD record onto QUICKs data base. In addition to inserting attribute values, this subroutine places the target record on the proper chains. This includes linkage under the sector, vulnerability, class, type, and other headings as directed by the nature of the target.

2.5.3 Subroutine ASTERISK. This overlay removes targets from the integrated data base and flags all target records on the output JAD format file. If identical target records reside both within the integrated data base and the output JAD file, an asterisk (character position 294) is placed on that record within the JAD file.

The list of target record to be retained are defined within a KEEPING clause and has the form:

KEEPING lowdesig [- highdesig]

[, lowdesig [- highdesig] . . .]

This clause consists of a list of DESIG ranges that are to be kept in the data base and flagged on the output file.

2.6 Common Block Definition

Common blocks used by JLM are outlined in table 1. Common blocks that communicate with the COP are given in Appendix A of Program Maintenance Manual, Volume I.

Table 1. Module JLM Internal Common Blocks

BLOCK	VARIABLE OR ARRAY	DESCRIPTION
ASNKEY	ASNKEY	Reference code of the ASNTYP record describing target
CLASSES	CLASSES(60)	Valid class names from the Assignment table
JADREC		LIST of words as read from the JAD format (see figure 3)
OPTION	OMITTING	Start of the OMITTING clause
	REPLACIN	Start of the REPLACING clause
	SETTING	Start of the SETTING clause
	UNIT	Start of the UNIT clause
	WHRSTRT	Start of the WHERE clause
	ONPRINTS	Start of the ONPRINTS clause (0 if option used)
	ORDER	Start of the ORDER clause
PAIRS		Commonly used pairs of words (character*12) in input
	ALFOS	Alphabetic follows
	FLOFO	Numbers follow
	SLASH	/
	COMMA	,
	DASH	-
	STAR	*
	NOT	NOT
	GREATER	>
	PAIR	Two word input to compare with the above operators
PRINSP	PRINON	Logical flag to print optional prints
SIDES	SIDES(5)	Values of SIDE found in the data base
TARDEF	STDLO	Level of local bomber defense at low altitude
	STDHI	Level of local bomber defense at high altitude

2.7 Subroutine ENTMOD

PURPOSE: Read the command verb and control the flow accordingly

ENTRY POINTS: ENTMOD (first subroutine called when overlay link JLM is executed)

FORMAL PARAMETERS: None

COMMON BLOCKS: ERRCOM

SUBROUTINES CALLED: ASSIGN, ASTERISK, INSCNT, LLINK, SELECT

CALLED BY: COP

Method:

This subroutine calls utility subroutine INSCNT in order to define the command verb that caused the execution of this overlay. Based on the verb, the proper overlay is executed and a RETURN made. If the verb is not recognized an error message is printed.

Subsections to follow present each overlay that may be executed within JLM.

ENTMOD is illustrated within figure 4.

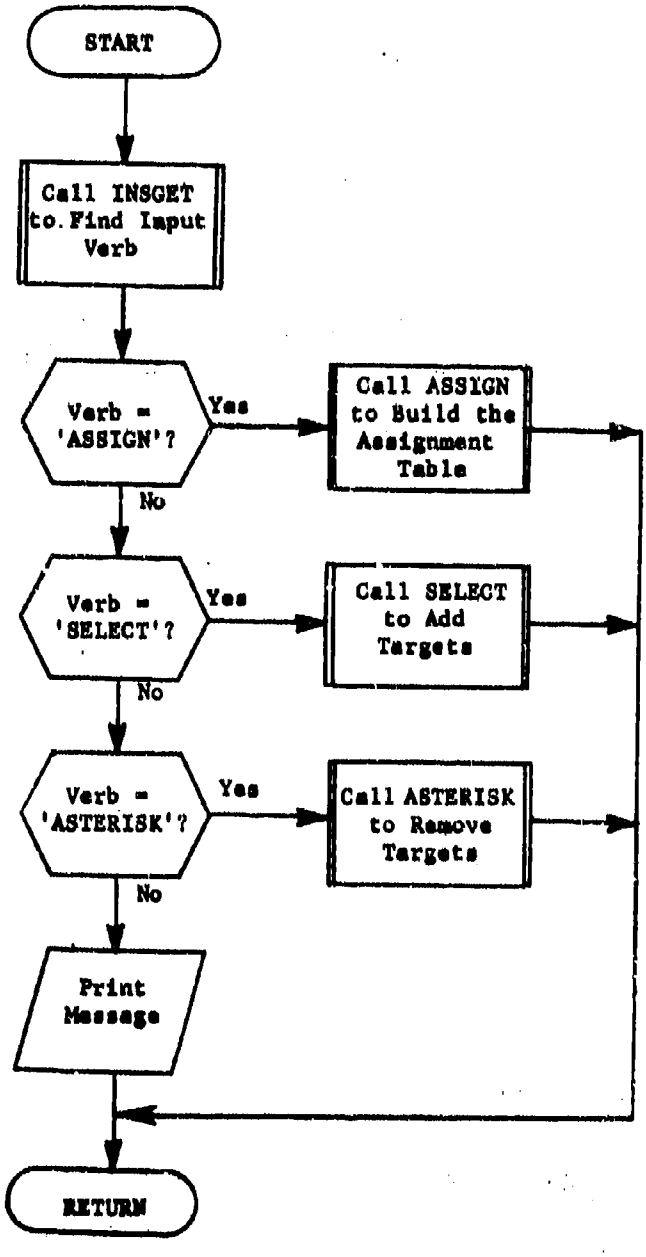


Figure 4. JAD Loading Module

2.8 Subroutine ASSIGN*

PURPOSE: Search for all adverbs associated with verb ASSIGN

ENTRY POINTS: ASSIGN

FORMAL PARAMETERS: None

COMMON BLOCKS: None

SUBROUTINES CALLED: ALPHAS, INSGET, PLAYERS, TOPRINT

CALLED BY: ENTMOD (of JLM)

Method:

Each clause associated with verb ASSIGN is unique. Therefore subroutine ASSIGN simply determines each included clause and executes the necessary subroutine for processing. Subroutines ALPHAS and PLAYERS conduct the actual investigation of input clauses. Subroutine TOPRINT prints results.

Subroutine ASSIGN is illustrated within figure 5.

*First subroutine of overlay link ASSI

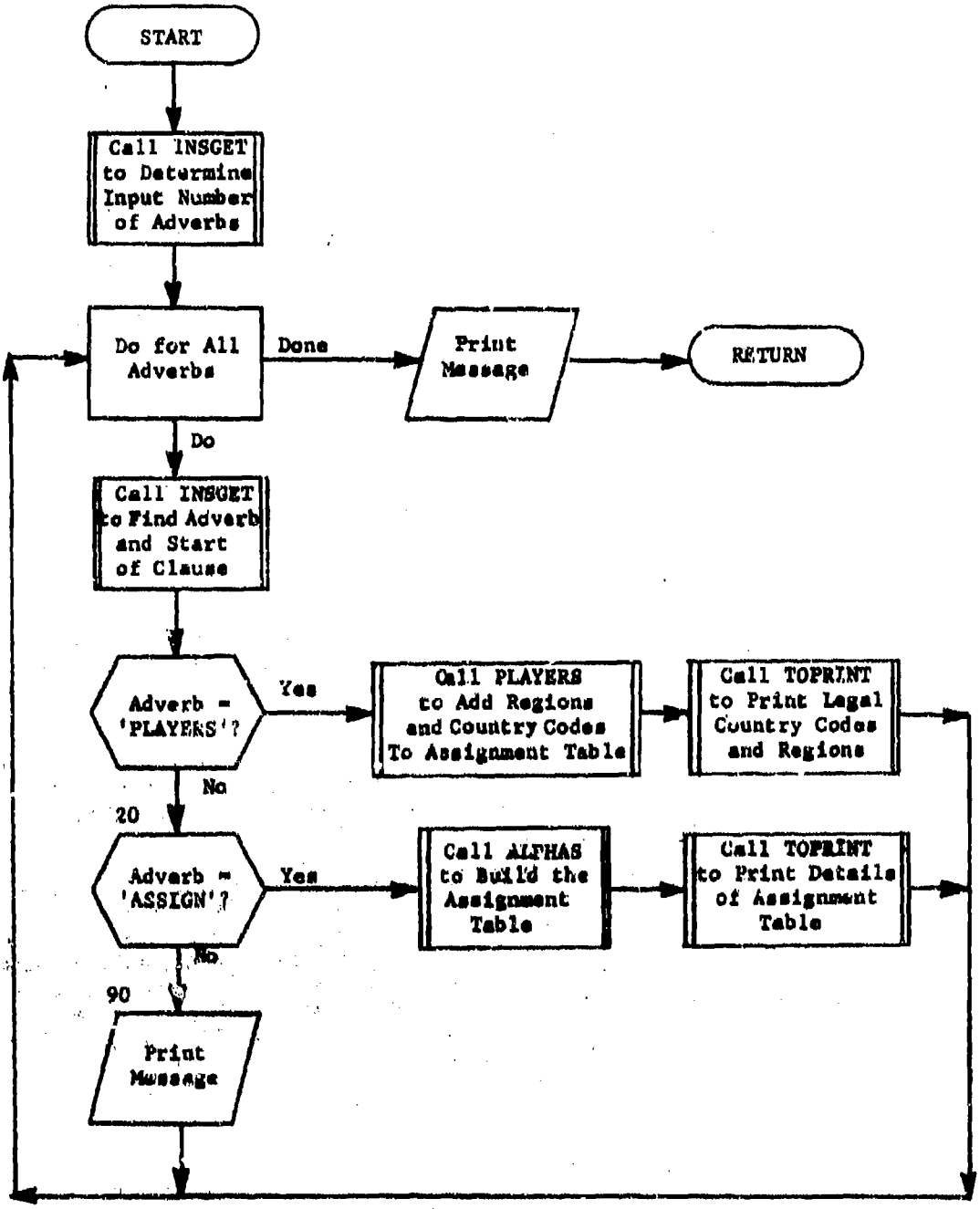


Figure 5. Subroutine ASSIGN

2.8.1 Subroutine ALPHAS

PURPOSE: Build the Assignment table as directed by the input clause ALPHAS

ENTRY POINTS: ALPHAS

FORMAL PARAMETERS: IPTR: Starting location into INSGET's arrays for ALPHAS clause

COMMON BLOCKS: CLASSES, C10, C15, C30, OOPS, PAIRS, SIDES

SUBROUTINES CALLED: DIRECT, FINDCLASS, FINDSIDE, NDFND, HEAD, INSGET, MODFY, NEXTTT, RETRV, STORE

CALLED BY: ASSIGN

Method:

ALPHAS begins by reading and storing parameters within the input clause (see subsection on subroutine functions) and, then, builds the Assignment table as directed.

Input Translation

The initial thrust of ALPHAS involves the reading and storage of parameters describing the input clause. Code may be readily understood through knowledge of the various combinations permitted within the input clause.

Local parameter IPTR is the pointer to the current position within the input clause and parameters CCF and CCL are the values of IPTR when the first and last countries of a list are found. PAIRS and LOCOWN are collections of pairs of values corresponding to key values in the input clause. LOCOWN contains the values generated by LOCATED IN and OWNED BY. The key values in PAIRS are ALFOS (alphabetic value follows), FLOPO (numeric value follows), SLASH ('/'), DASH ('-'), GREATER ('>'), STAR ('*'), COMMA (','), and NOT ('NOT'). These key values determine the function of the values that follow. For example, an alphabetic follows, not directly preceded by an operator indicates a new value for SIDE. '//' indicates a new ACLASS, '-' a new ATYPE or CATNI, '>' a name or minimum capacity, '/' CATLO, '*' ASNTASK, ',' DESIGAI or country code, 'LOCATED IN' or 'OWNED BY' a country code.

ALPHAS scans through the input clause until it encounters a redefinition of either SIDE, ACLASS, ATYPE or CATLO. When this occurs the information collected up to this point is added to the Assignment Table. It should be noted that when a key value like ATYPE is encountered all higher values, SIDE and ACLASS, will not be redefined.

As a point of clarification, be aware that attributes ACLASS and ATYPE are defined within the organizational structure and their values will be set to attributes CLASS and TYPE upon proper target selection.

Assignment Table Structure

Subroutine ALPHAS will construct the bulk of the Assignment Table and will link the position constructed with the country codes added by subroutines PLAYERS. The structure, or shape, of this table follows.

Chained to the header (ASNTAB) for each side are records containing the valid country codes and the region they are in (ASNCTY). These records are sorted on region and country code. Also under each header are records (ASNCLS) containing the values for CLASS for the side. Under each of these records are all the ATYPE names that belong to this class (ASNTYP). The countries and types are connected via common ASNREC records. These records contain restrictions based on Category Code, the location or owner of the target, and either its name or size. It also contains the TASK that will be assigned to a target meeting these restrictions. On the other chain under ATYPE are the alphabetic portions of DESIG to be used in assigning a DESIG to the target (TYPDES). Subsequent DESIGs are used if the first values are already used. All of these records with the same alphabetic portion of DESIG are chained together under a common record (ASNDES) containing the DESIG and the number in each region.

Assignment Table Construction

After the input values have been collected, the Assignment Table is searched to find where new information is to be inserted. First, there is a search for ACLASS (under the proper SIDE). If the ASNCLS record of value ACLASS does not exist, the record will be created. Similarly, the ATYPE value record is searched for and the record created if necessary. It should be added that the first ASNREC record is also checked so that every ASNTYP record has a unique MINCAP associated with it. Now that the key record in the table has been found or created, the CONTRY chain is traversed looking for country codes that match those in the input clause between CCF and CCL. When a desired ASNCTY record is found, the ASNRNG chain is processed to see if there already exists an ASNREC record corresponding to the input, or one that only needs to be modified. If not, the ASNREC will be created. This process of searching the ASNRNG chain is done for each desired ASNCTY record. When completed, the DESIGA2

values collected in array IDESIG are placed at the end of the ALTDES chain. They are also linked under the proper ASNDES record, again creating a record if necessary.

This process is continued until the ALPHAS clause is exhausted.

Subroutine ALPHAS is illustrated in figure 6.

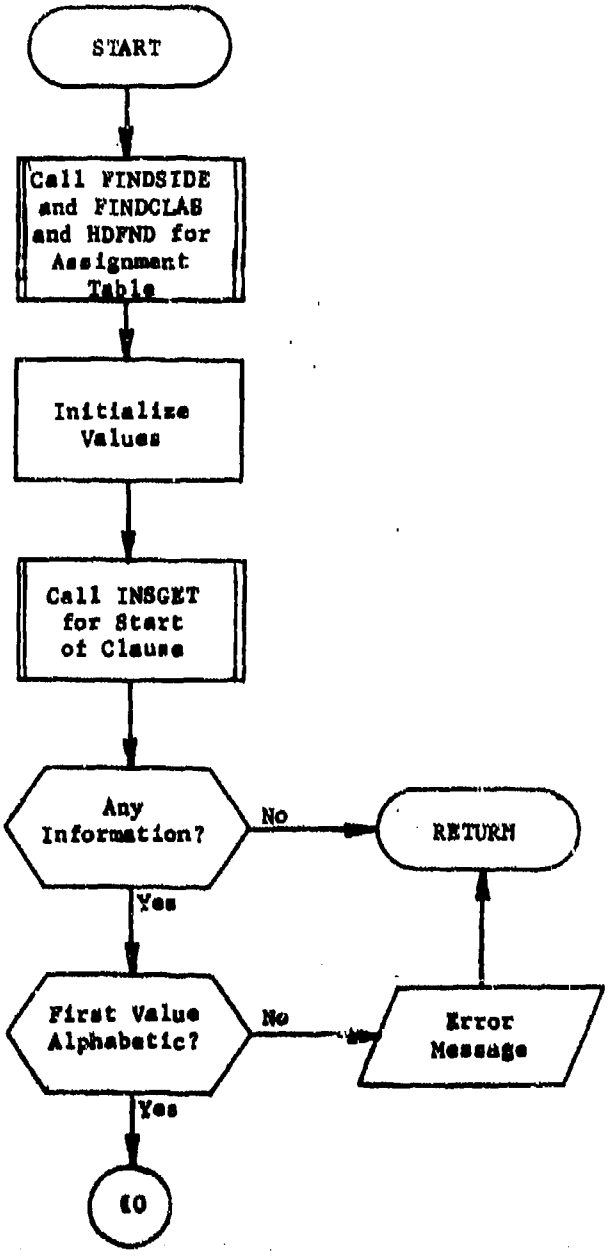


Figure 6. Subroutine ALPHAS (Part 1 of 17)

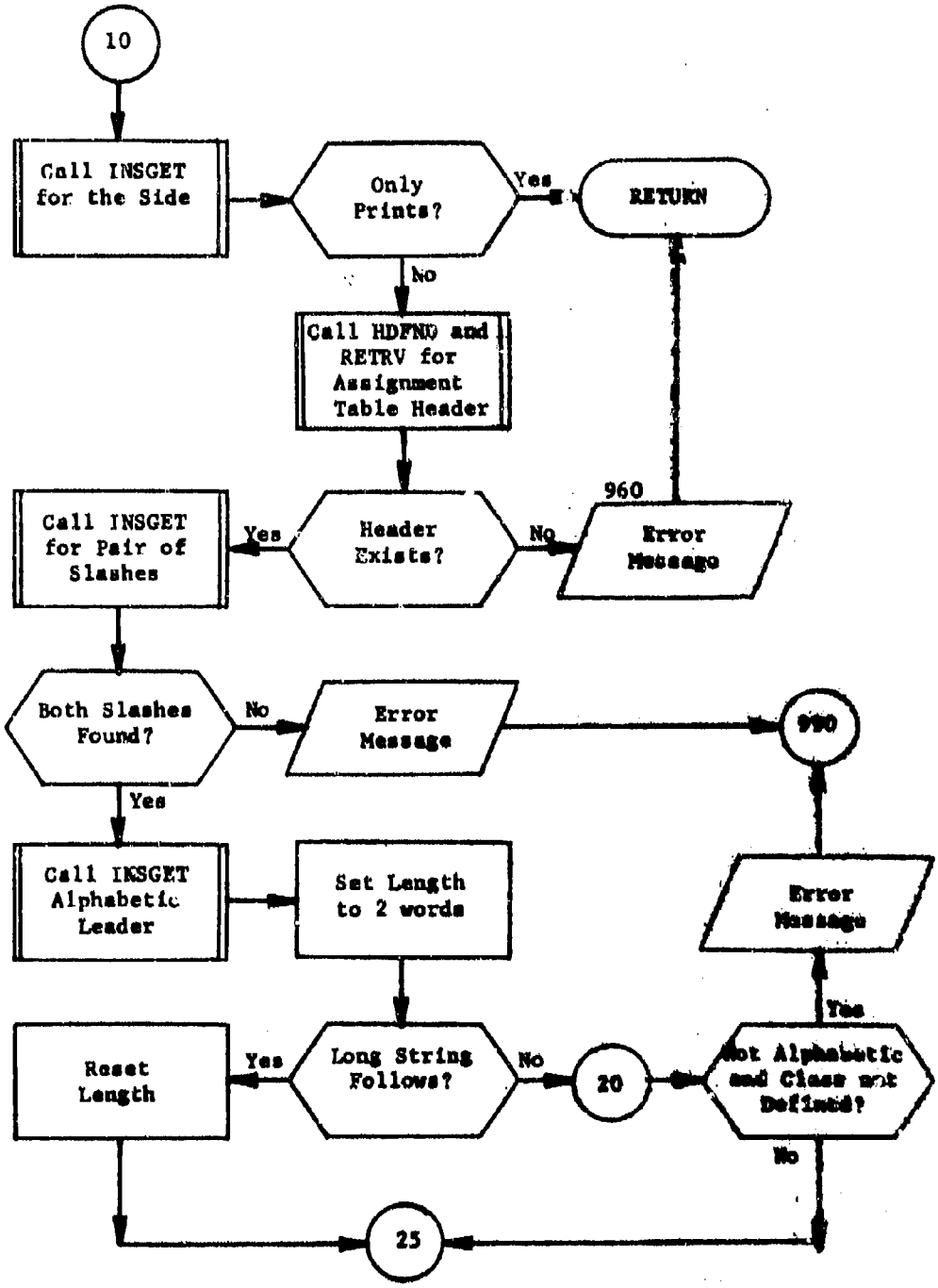


Figure 6. (Part 2 of 17)

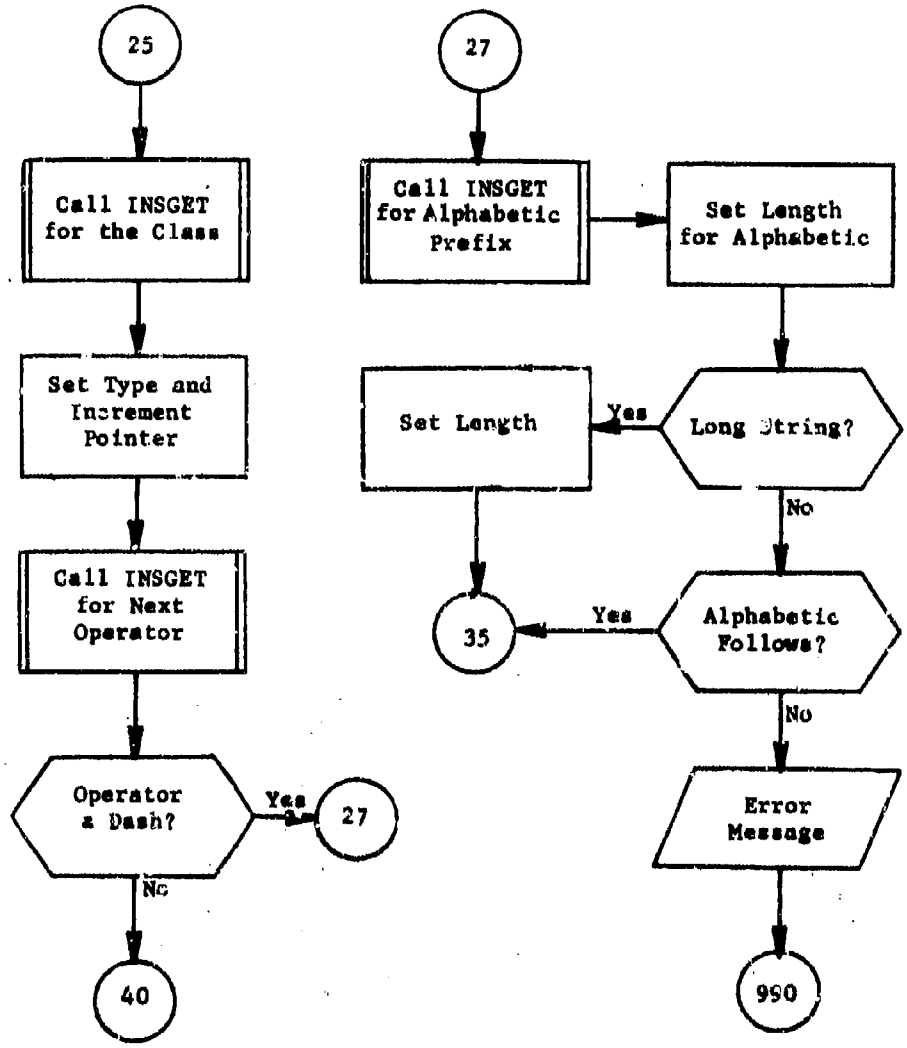


Figure 6. (Part 3 of 17)

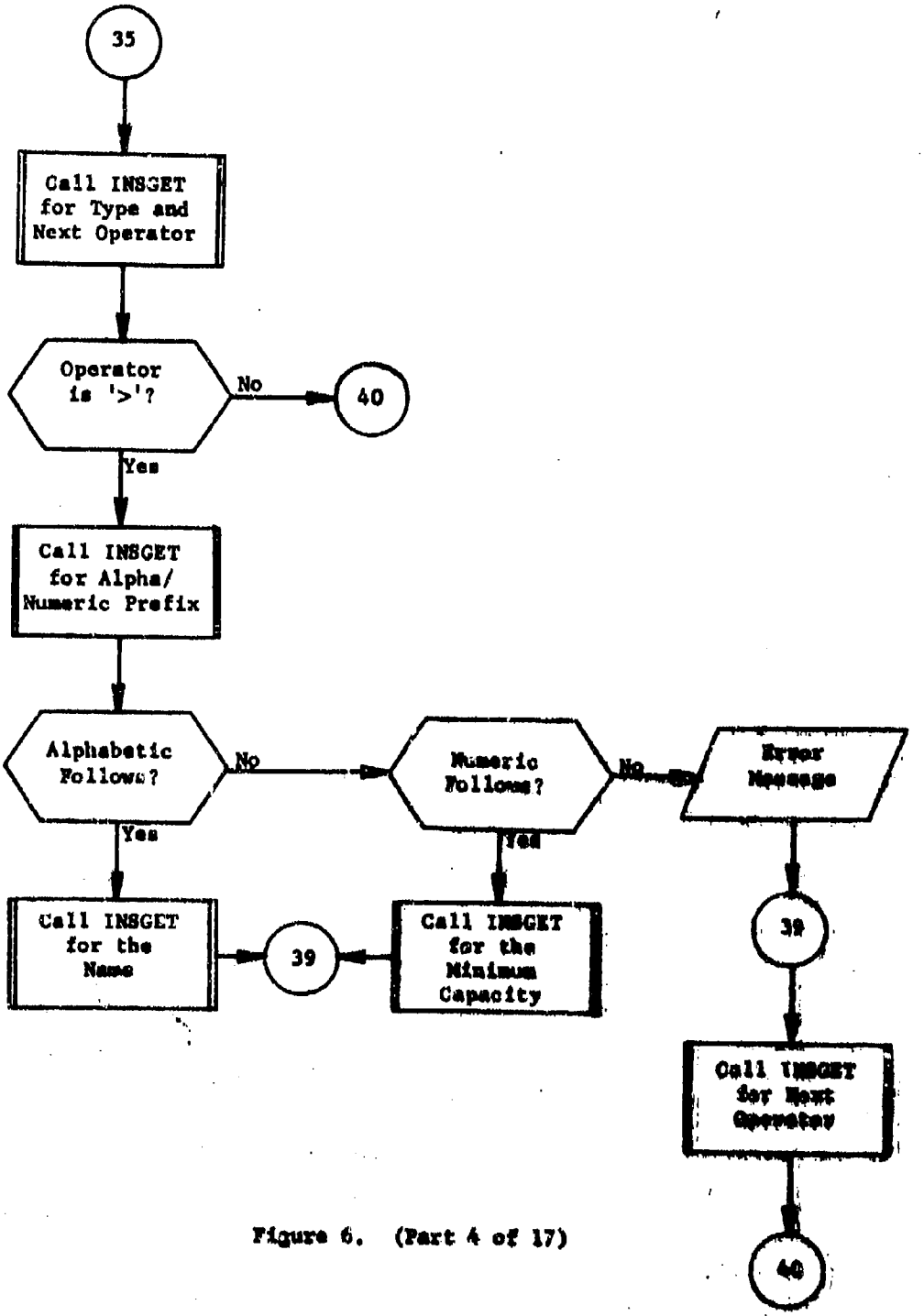


Figure 6. (Part 4 of 17)

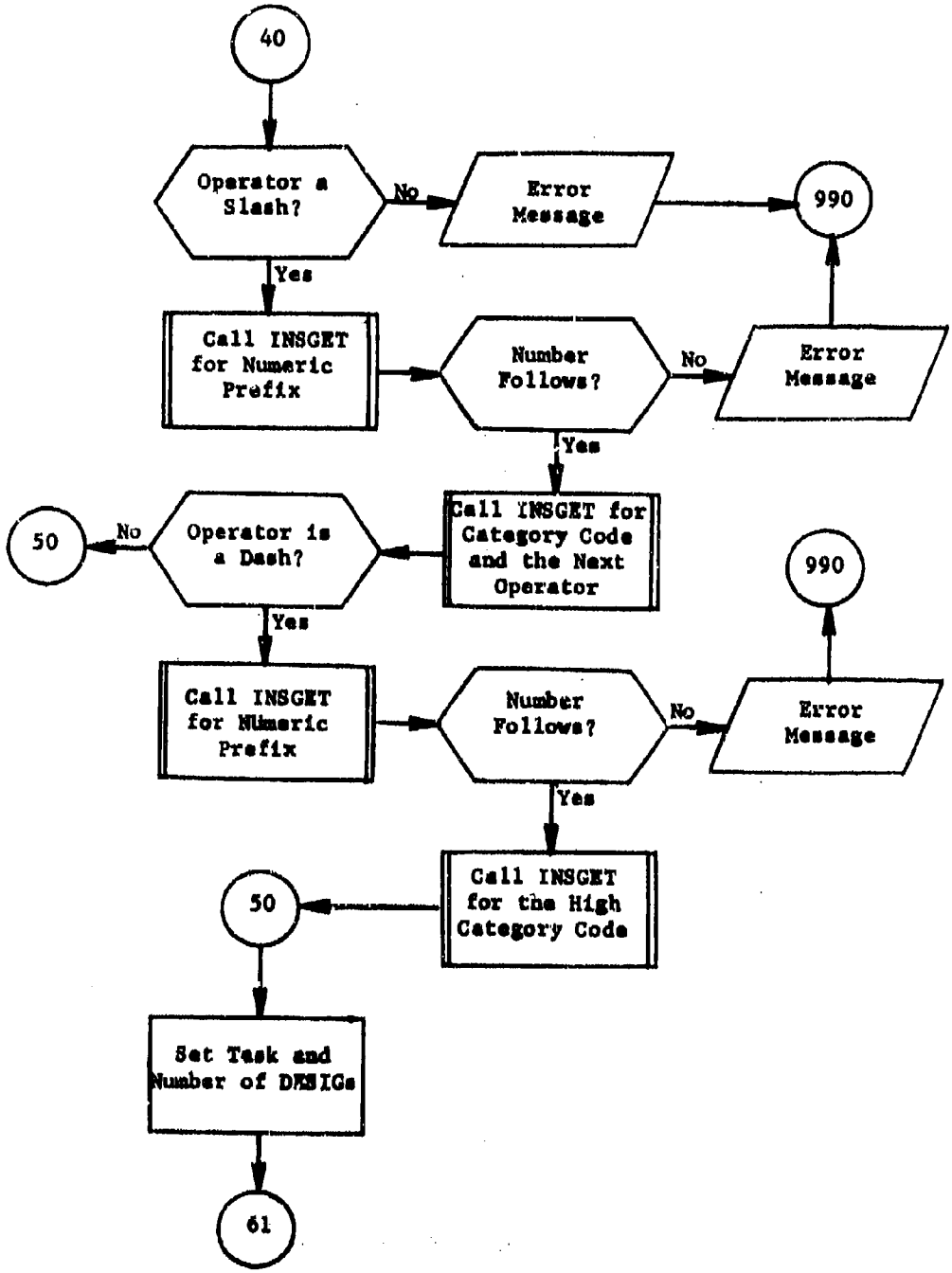


Figure 6. (Part 5 of 17)

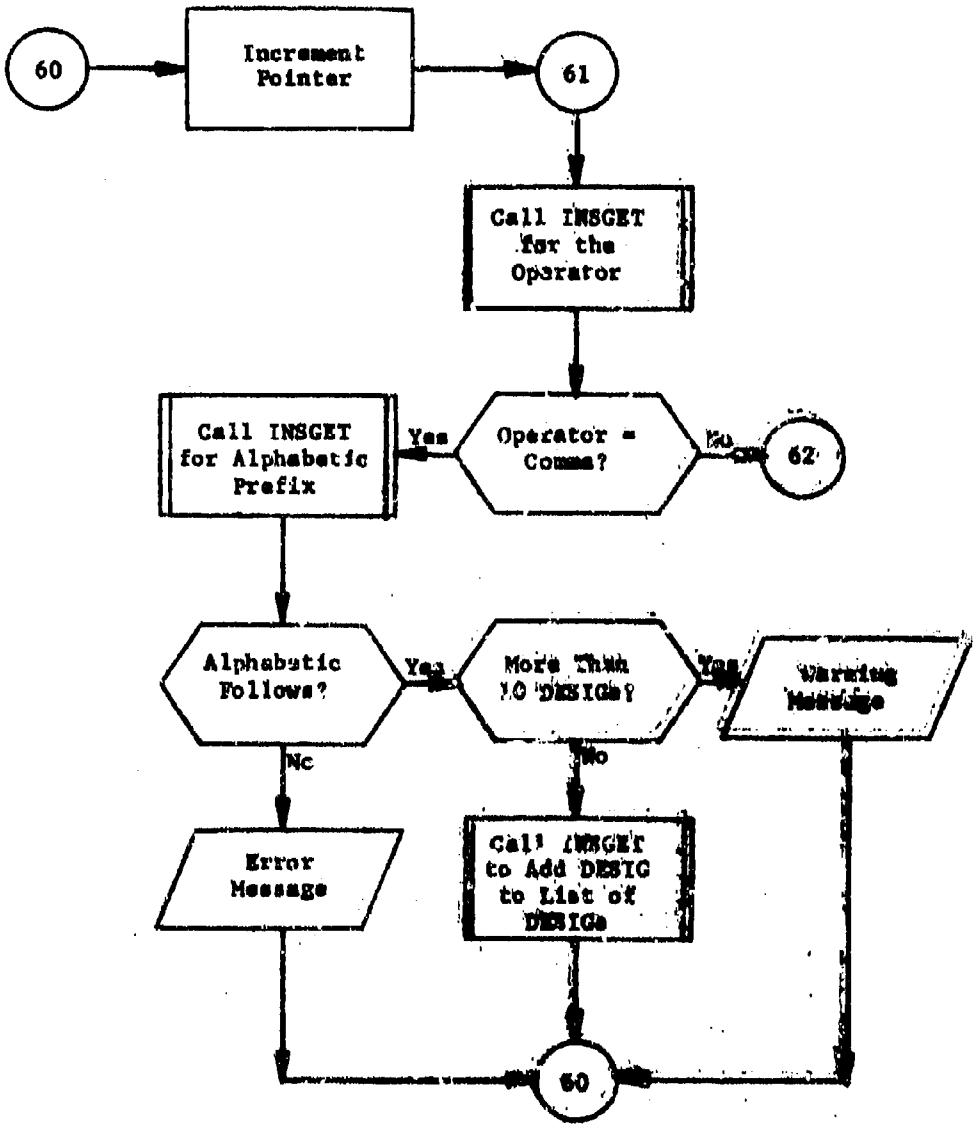


Figure 6. (Part 6 of 17)

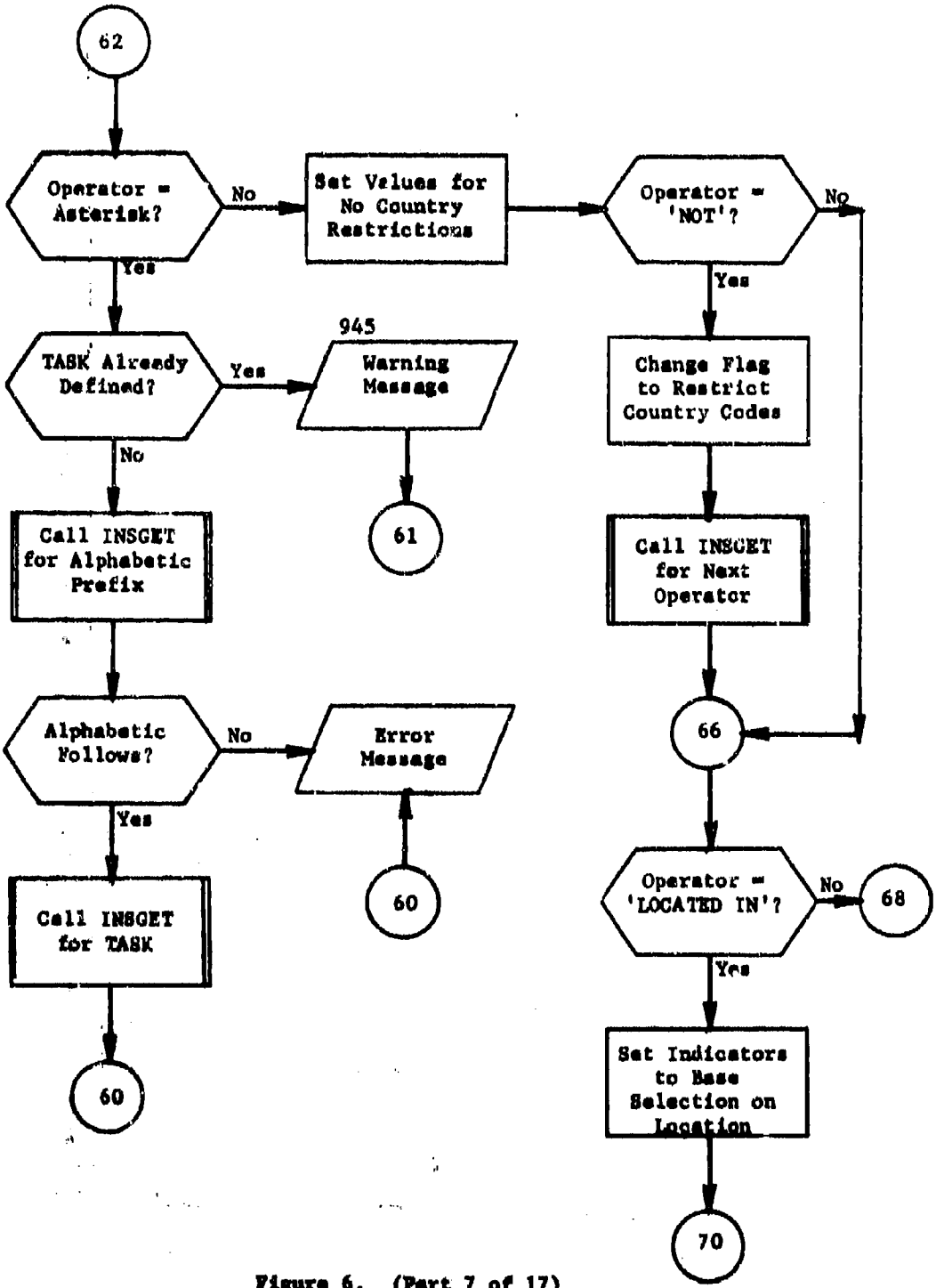


Figure 6. (Part 7 of 17)

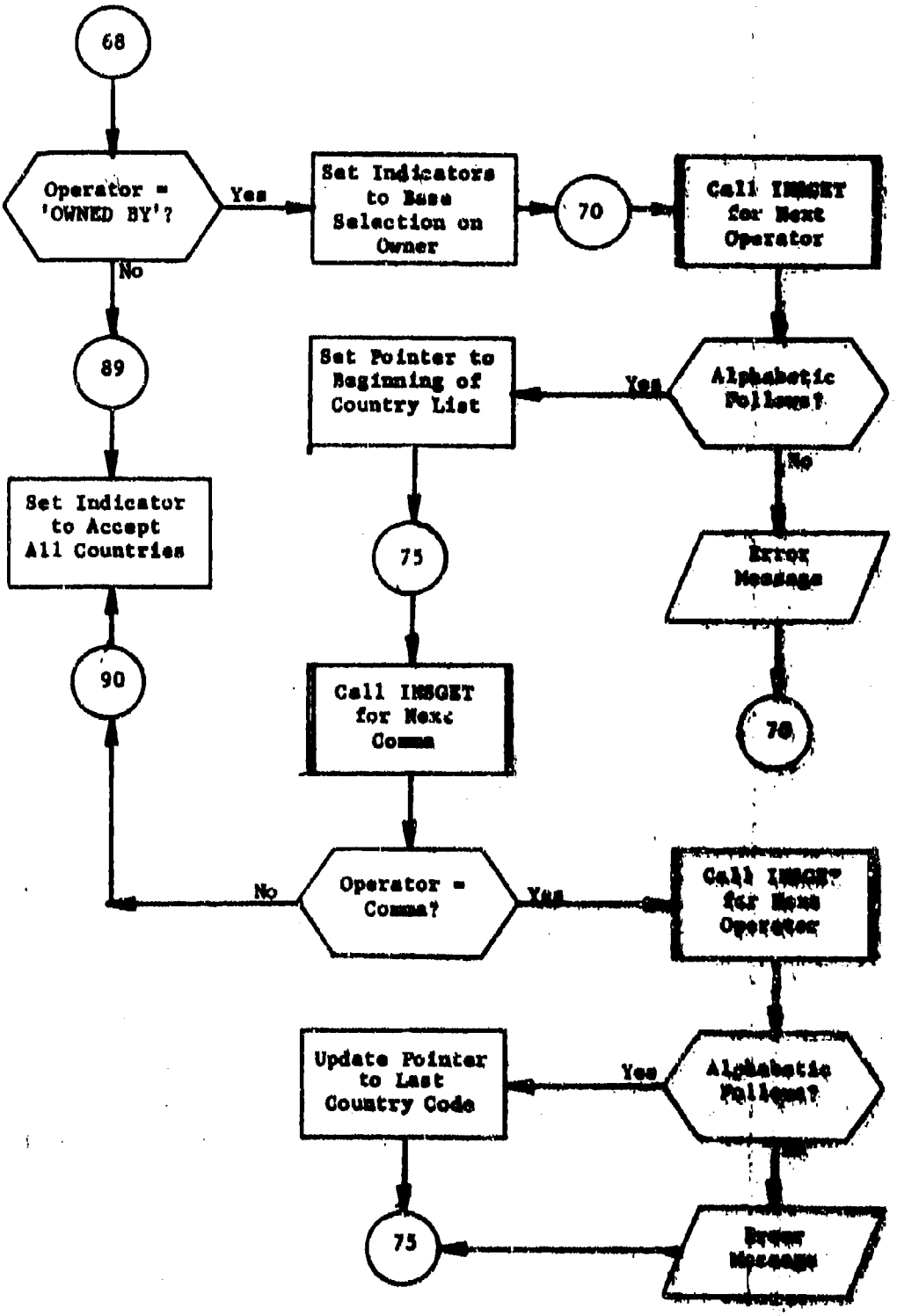


Figure 6. (Part 8 of 17)

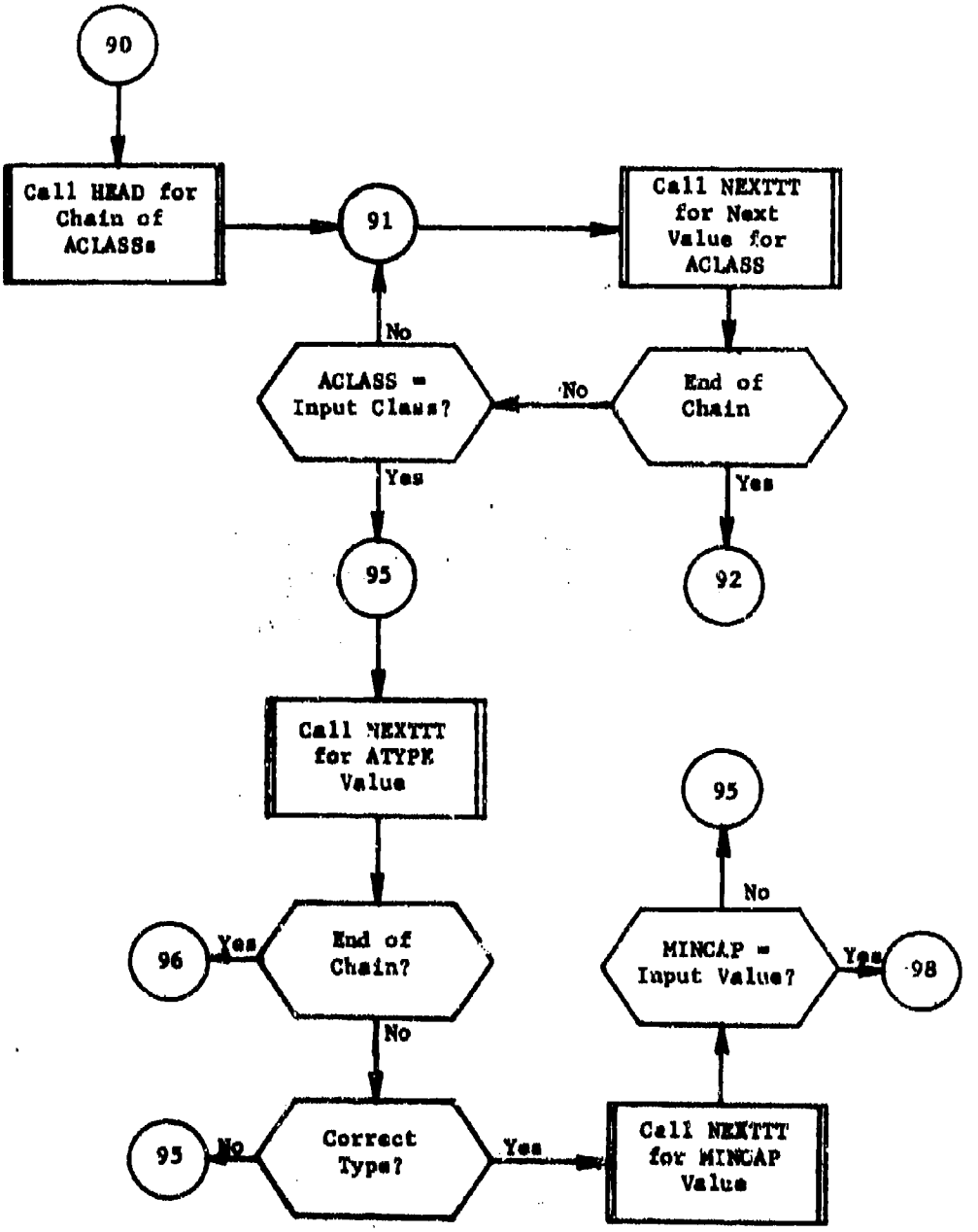


Figure 6. (Part 9 of 17)

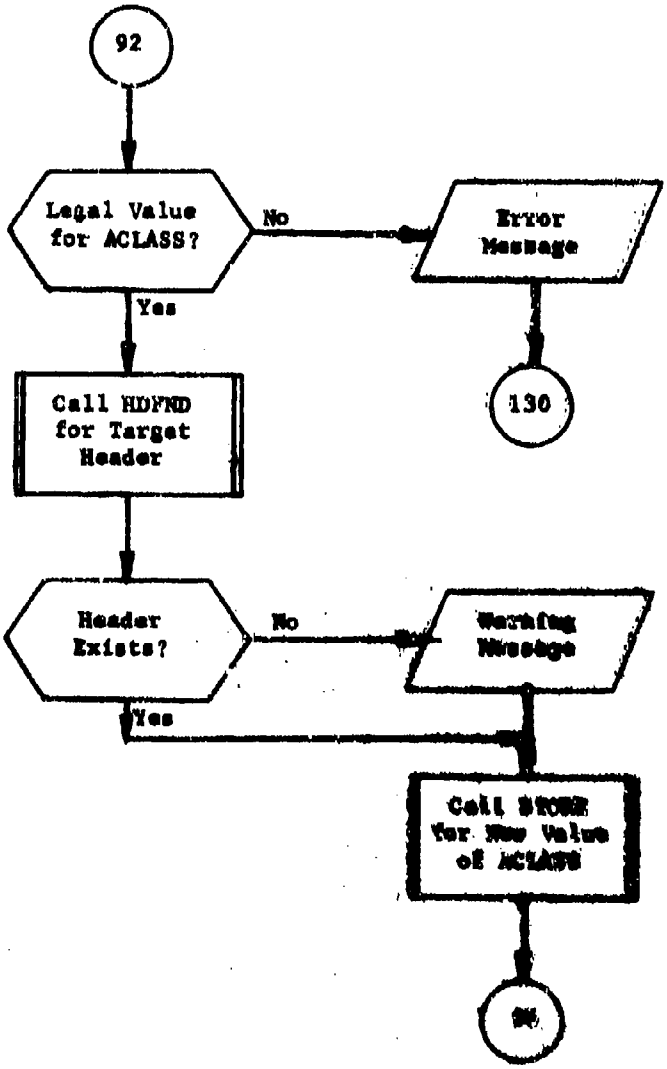


Figure 6. (Part 10 of 17)

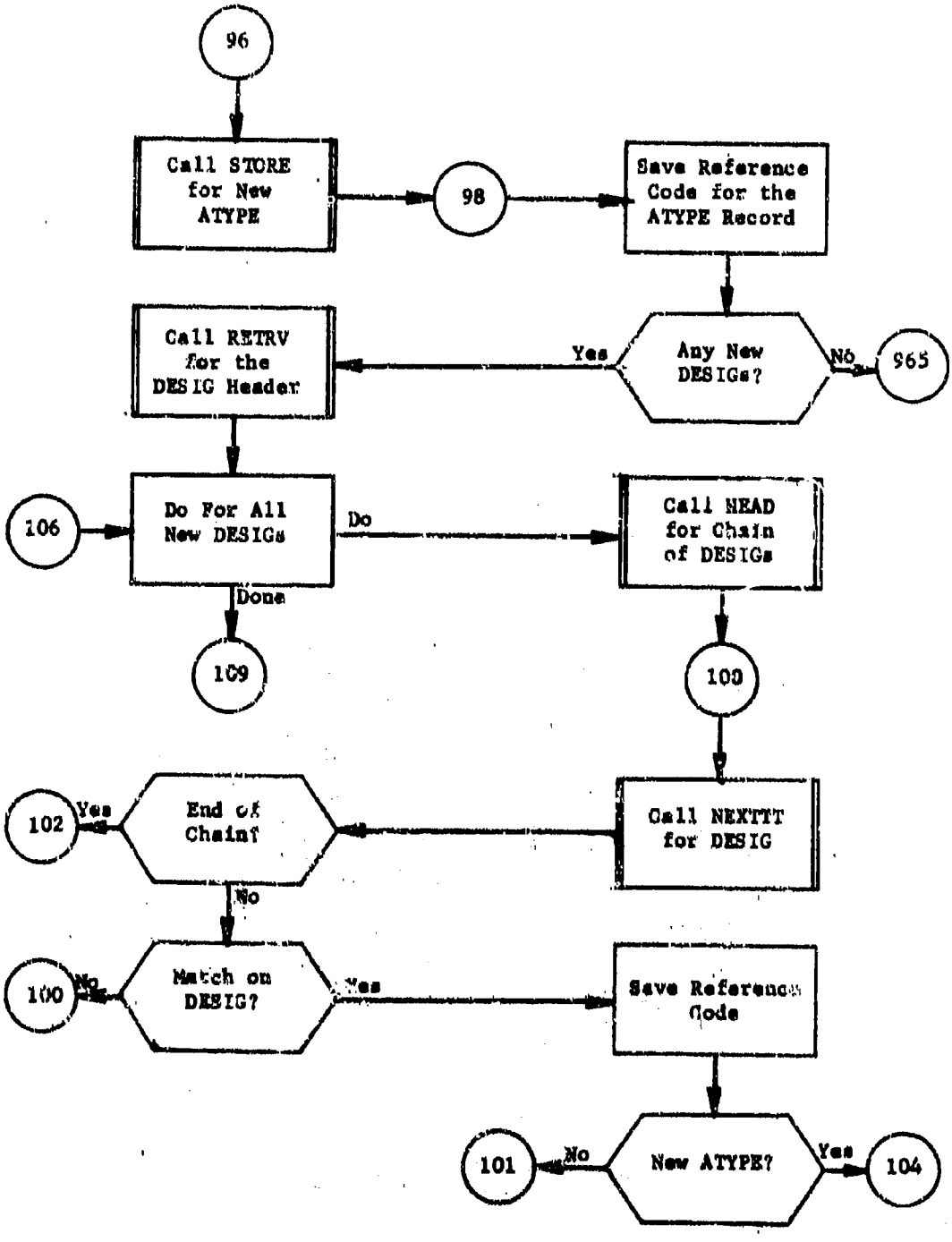


Figure 6. (Part 11 of 17)

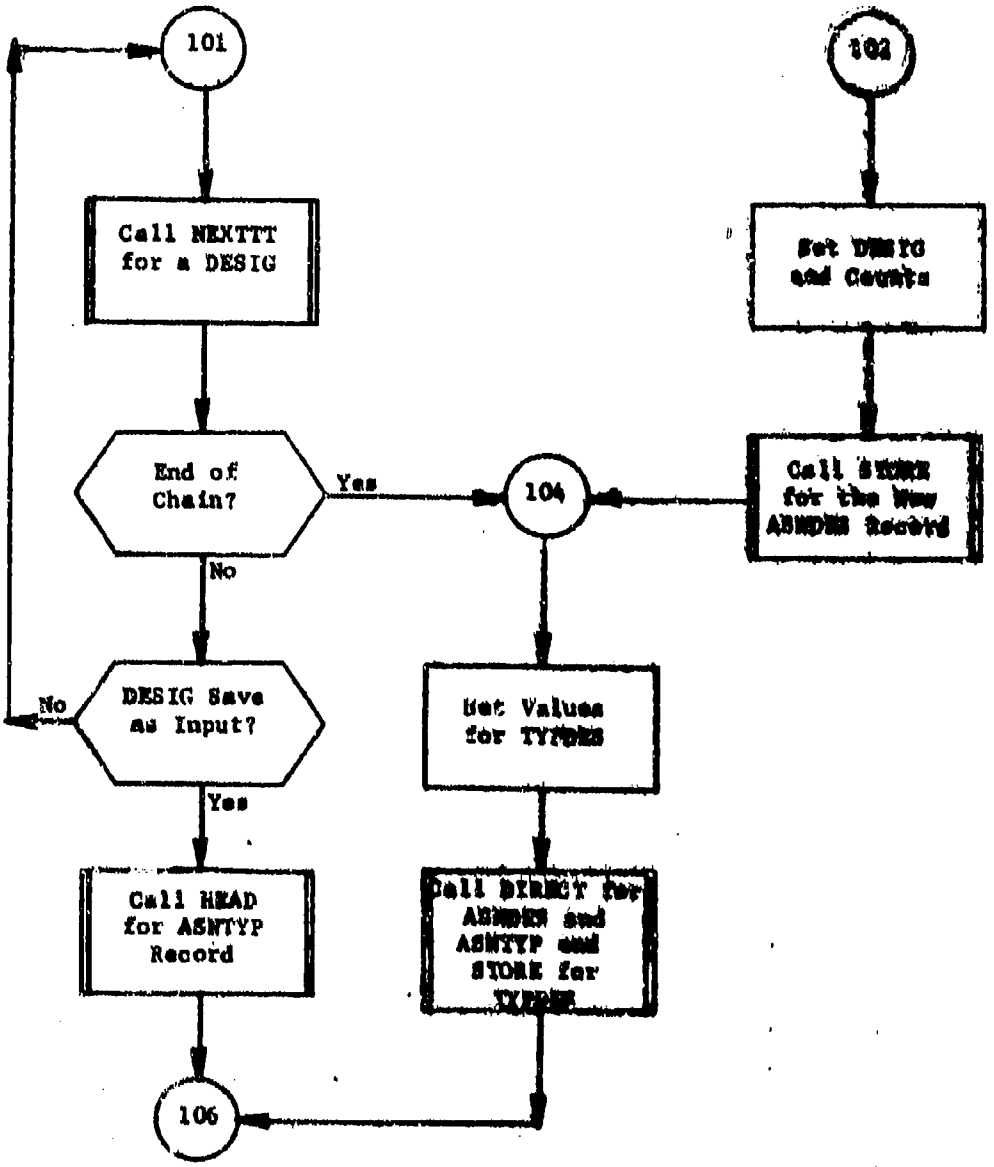


Figure 6. (Part 12 of 17)

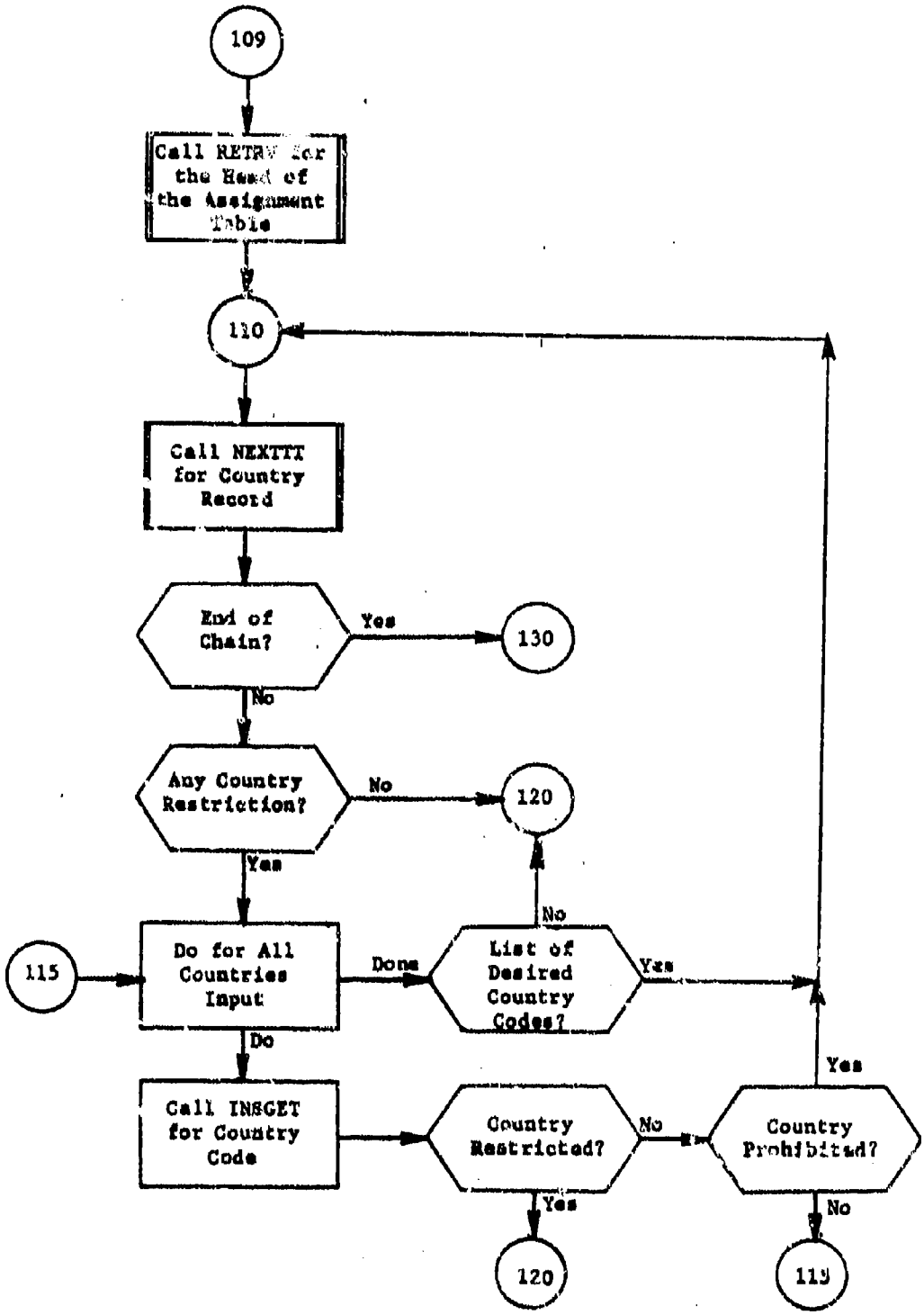


Figure 6. (Part 9 of 17)

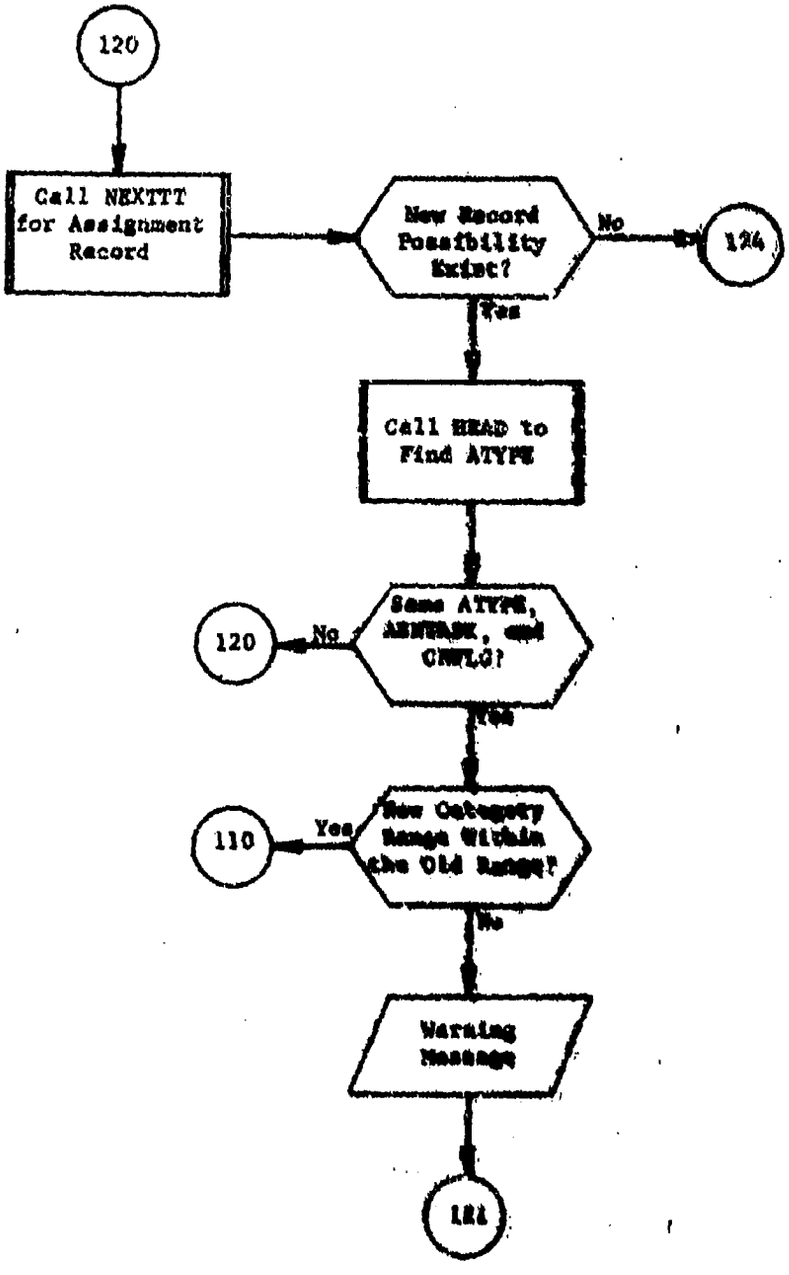


Figure 6. (Part 14 of 17).

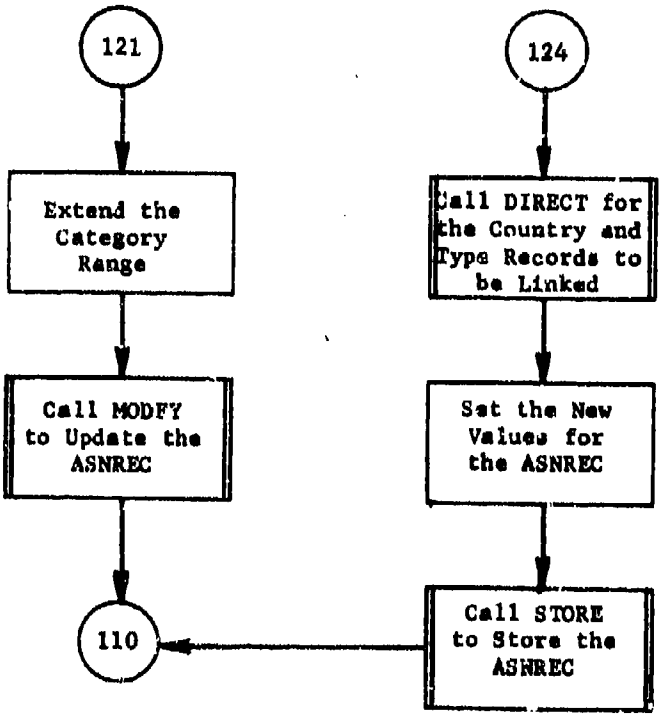


Figure 6. (Part 15 of 17)

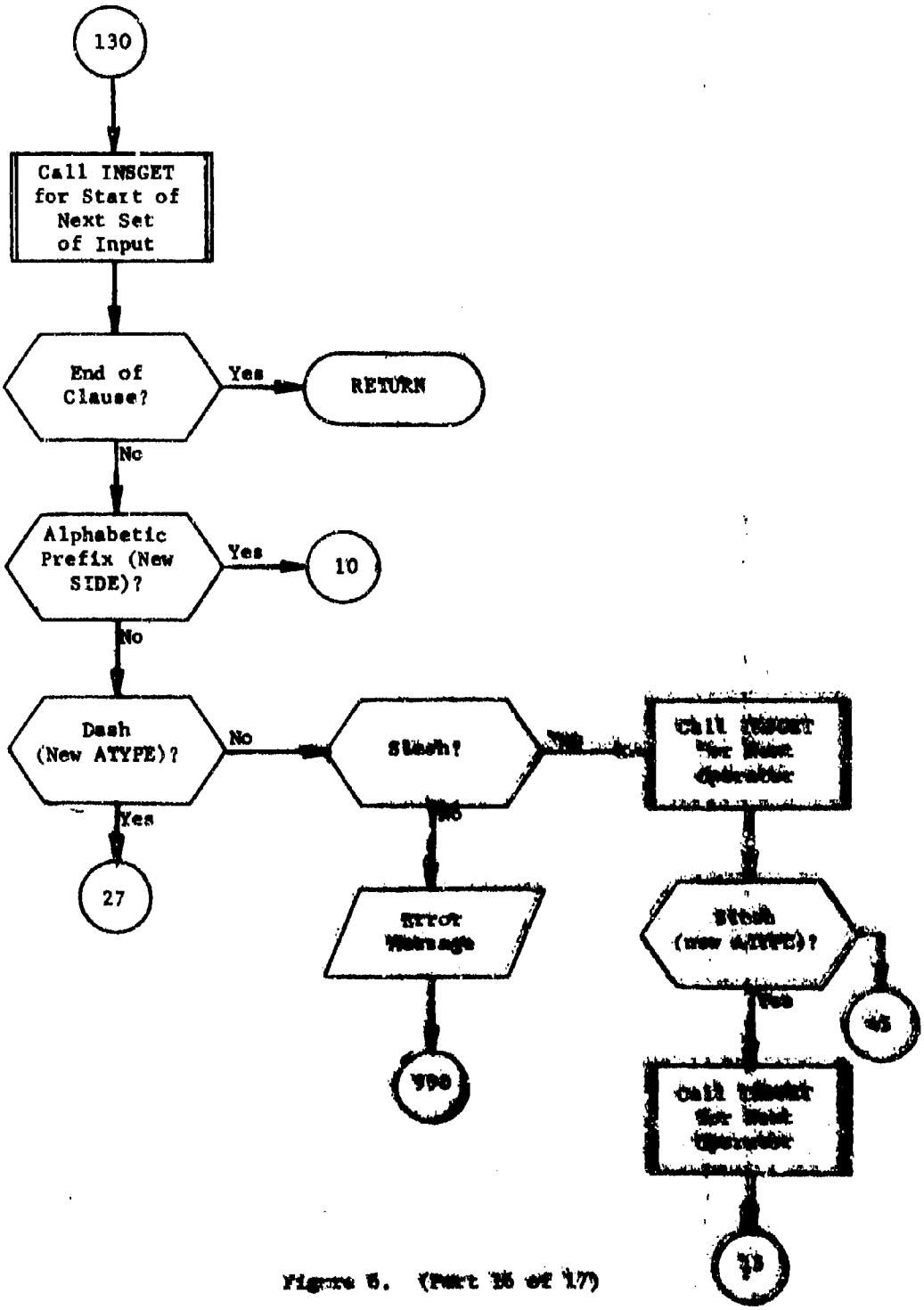


Figure 5. (Part 16 of 17)

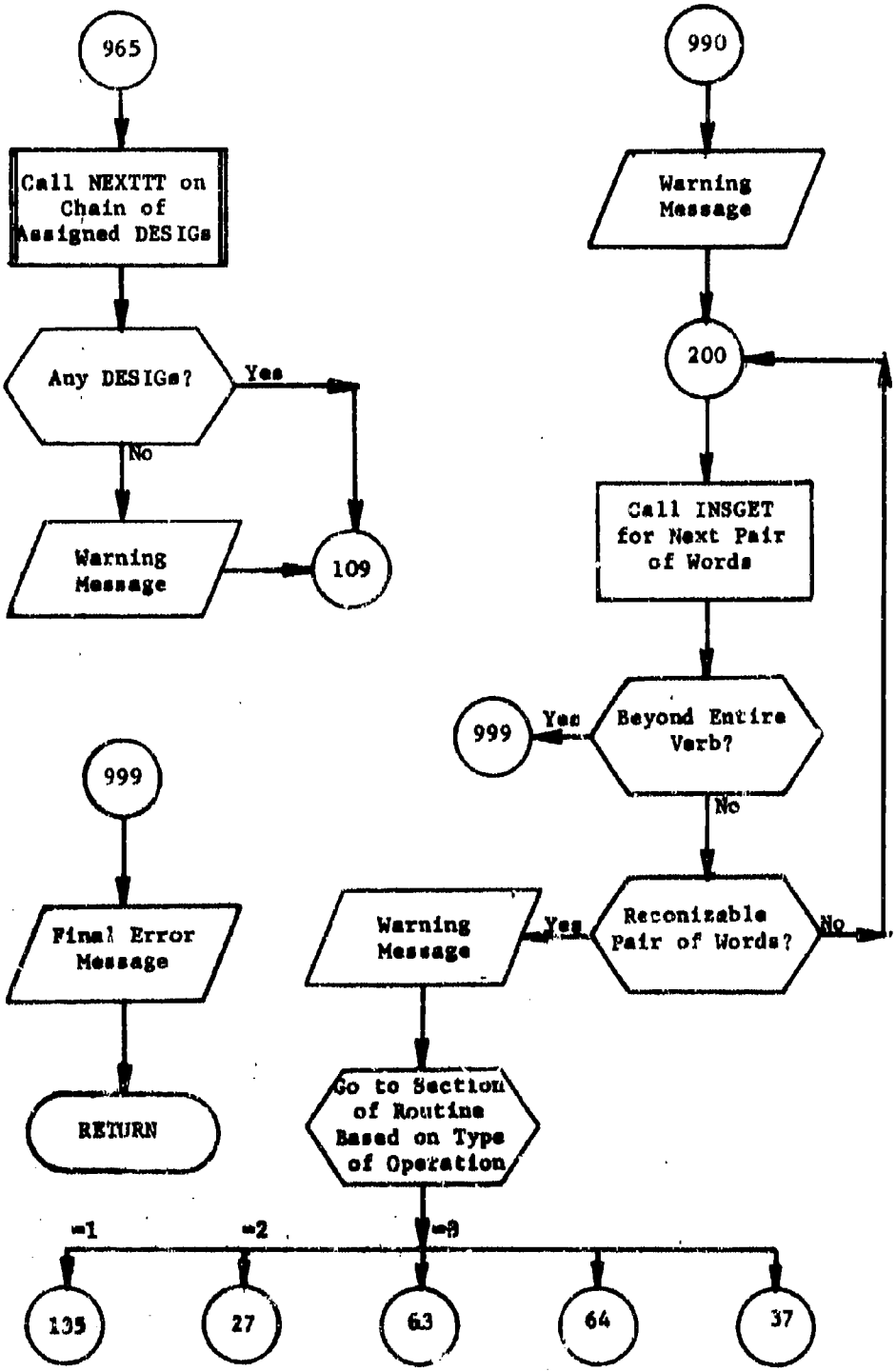


Figure 6. (Part 17 of 17)

2.8.2 Subroutine PLAYERS

PURPOSE: Build the Assignment table as directed by the input clause PLAYERS.

ENTRY POINTS: PLAYERS

FORMAL PARAMETERS: IPTR: Starting location into INSETS for PLAYERS clause

COMMON BLOCKS: C10, C15, C30, OOPS, PAIRS

SUBROUTINES CALLED: HDFND, HEAD, INSERT, MODIFY, SKIPTY, ENTRY, STORE

CALLED BY: ASSIGN

Method:

PLAYERS is used to build the portion of the Assignment table that relates a country code to a specific region and side. This consists of the CONTRY chain of records sorted on region and country code under the header for each side.

Subroutines GETTAR/FNDTAR enter the Assignment table via the CONTRY chain. The country codes for the owner and location of the target are found, then the valid category codes (created by ALPHAS) are checked to see if the target is acceptable.

The processing in PLAYERS is largely controlled by what input operator is encountered.

When an alphabetic follows (ALFOS) is encountered with an appropriately preceding operator then the alphabetic value following it will be interpretted as a SIDE and the Assignment table header for this side will be retrieved.

If a pair of slashes are encountered then the floating point number following will be the region for all subsequent country codes until another pair of slashes (or a new side) is encountered. If a single slash (SLASH), or comma (',') is encountered, then the alphabetic value following it will be a country code to be added to the chain.

Before the country code is added, however, control is exercised to see if the country already exists on the chain. If it doesn't, the record for the input country code and region is created. If the country exists but under a different region, the region is changed and a message is printed.

This processing continues until the input via INSERT has been exhausted.

Subroutine PLAYERS is illustrated in figure 7.

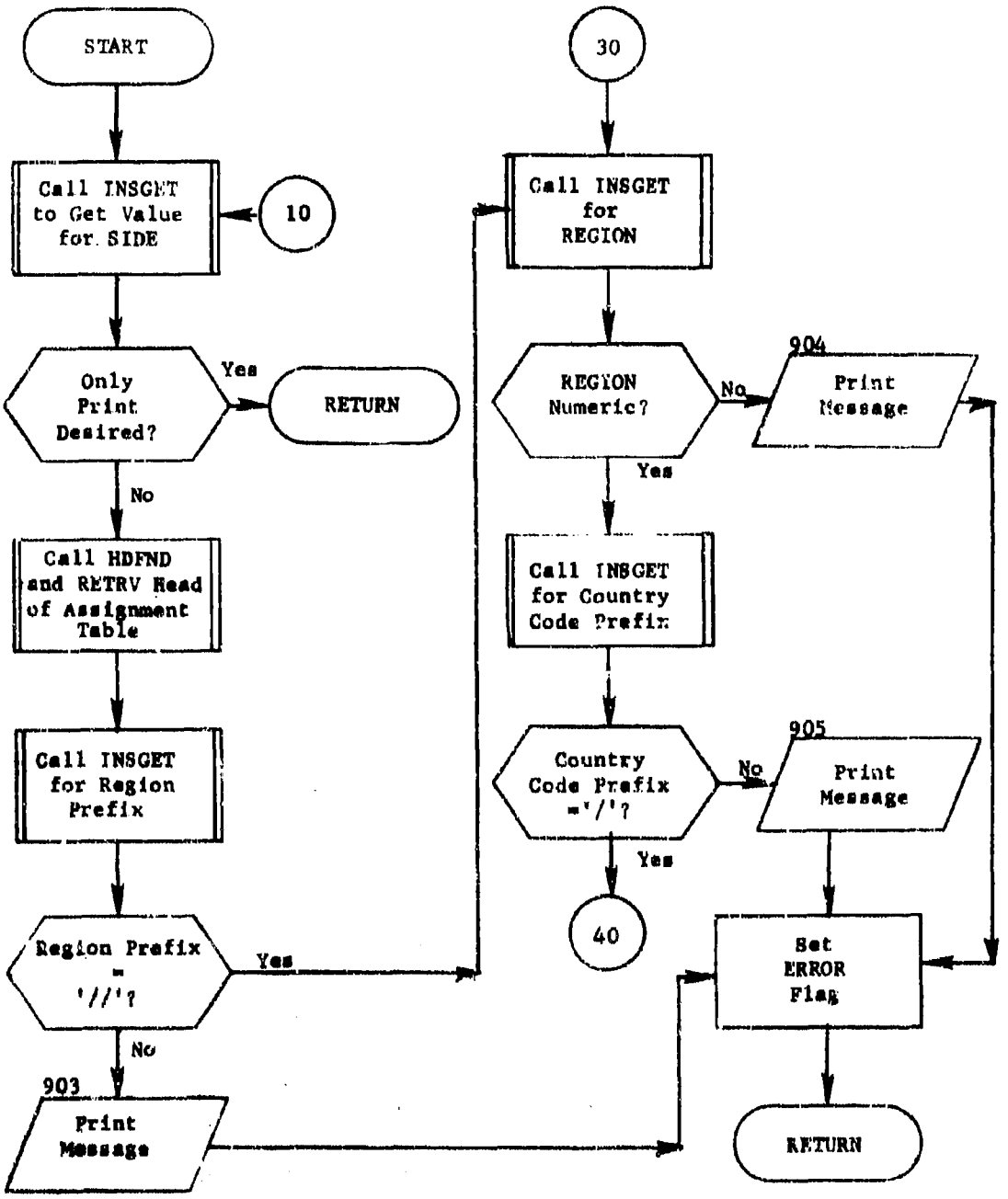


Figure 7. Subroutine PLAYERS (Part 1 of 3)

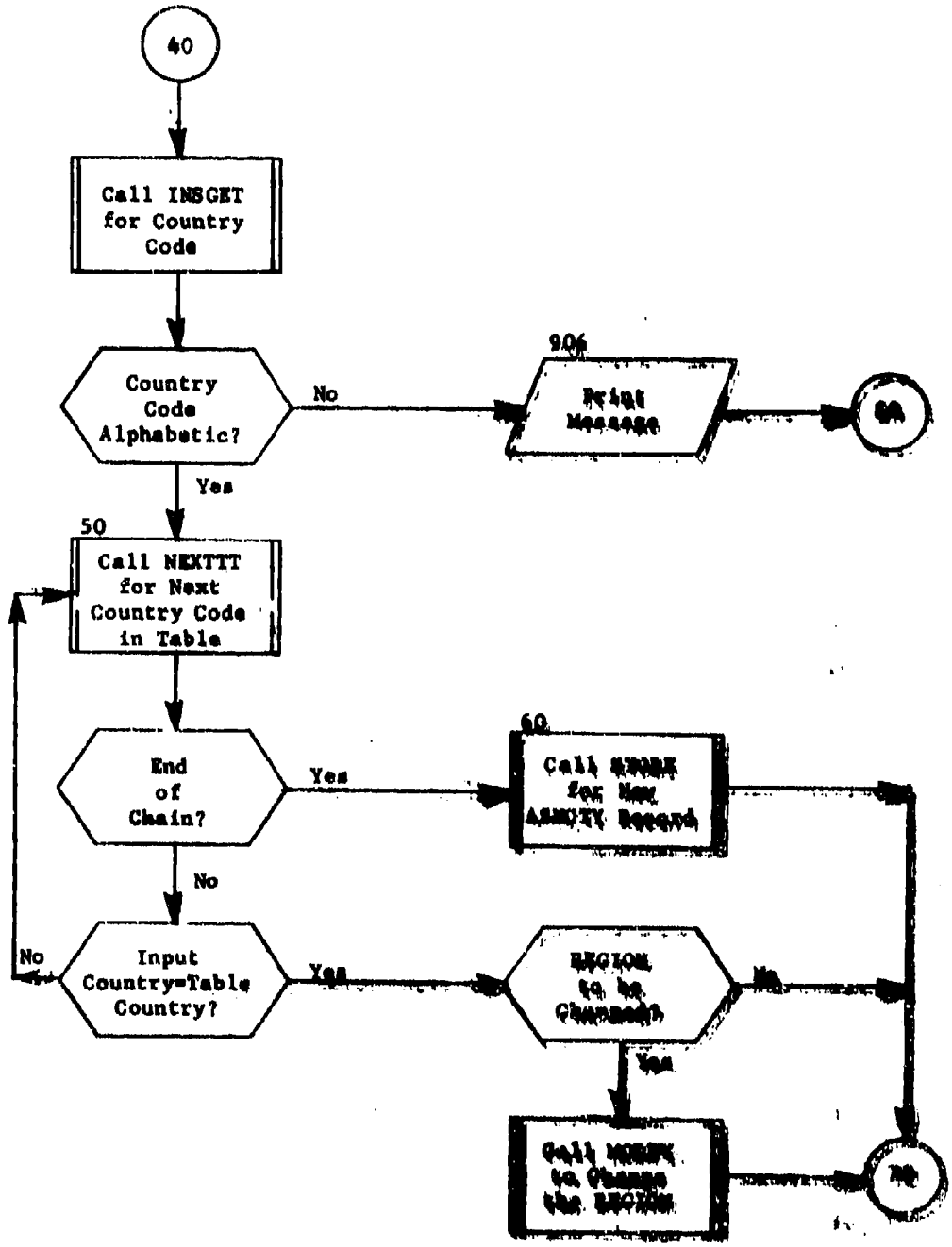


Figure 7. (Page 2 of 3)

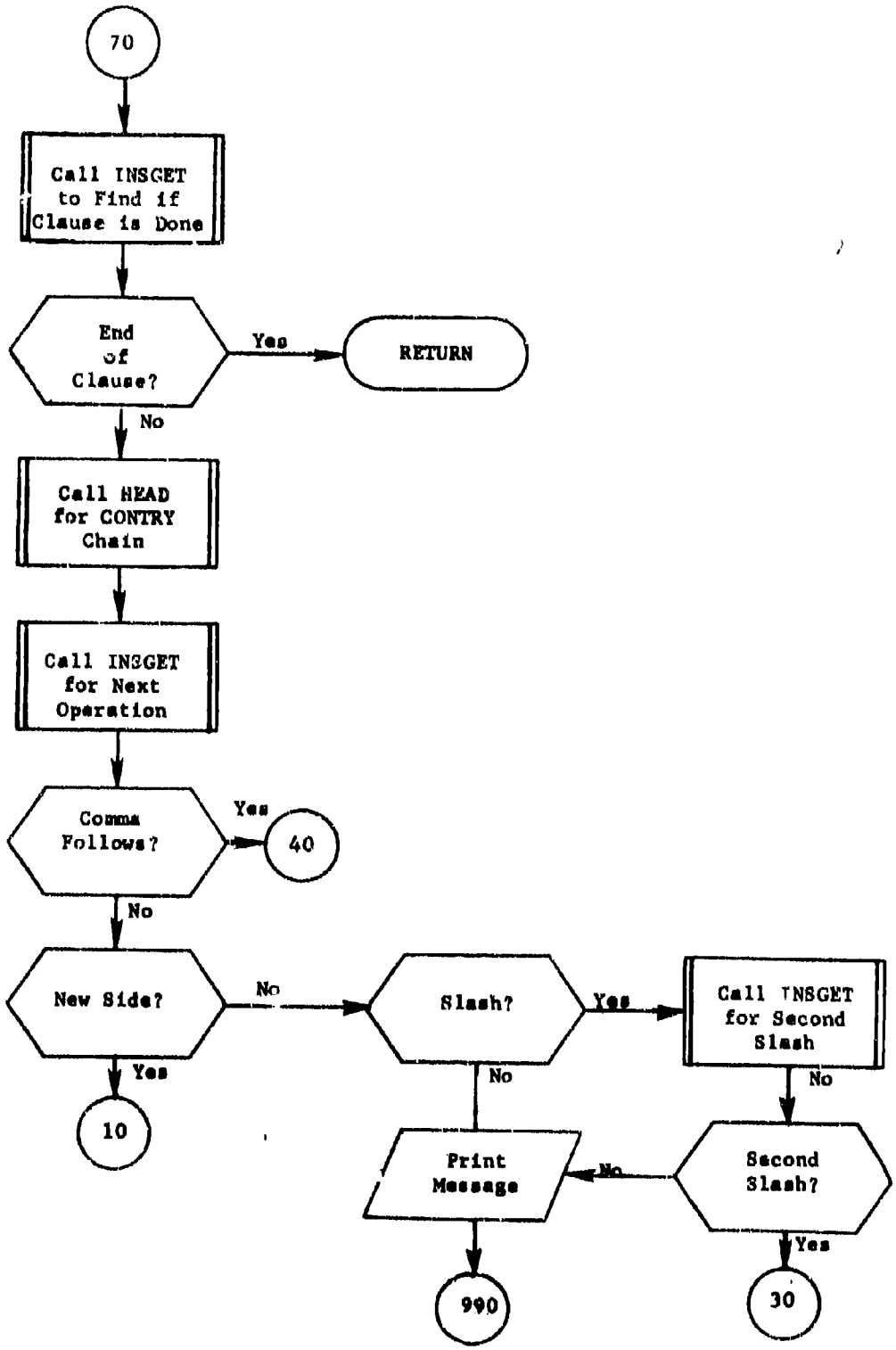


Figure 7. (Part 3 of 3)

2.8.3 Subroutine TOPRINT

PURPOSE: Print the Assignment table as built by the input clause.

ENTRY POINTS: TOPRINT

FORMAL PARAMETERS: WHAT = 'ALPHAS' or 'PLAYERS'

COMMON BLOCKS: C10, C15, C13

SUBROUTINES CALLED: HDFND, HEAD, FINDSIDE, NEXTYT, RETRY

CALLED BY: ASSIGN

Method:

The generation of print reports is the only function of TOPRINT (figure 8). The flow is subdivided into sections according to the two possible input clauses. In either situation, the Assignment table is chained, summarizing values collected and reports produced. The outline of each report is given in Users Manual, Volume II, section 2.

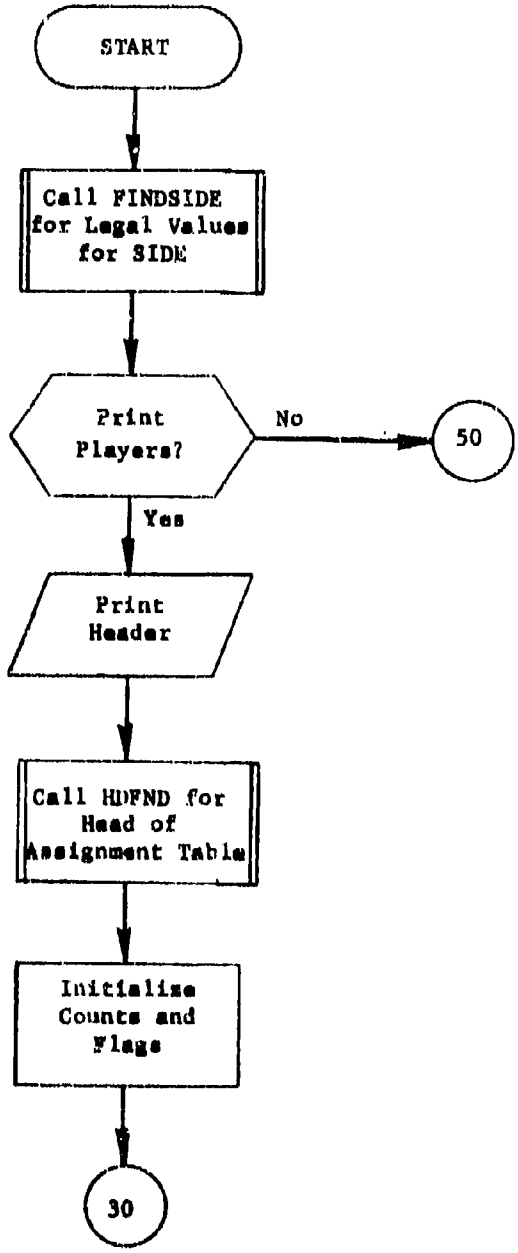


Figure 8. Subroutine TOPRINT (Part 1 of 6)

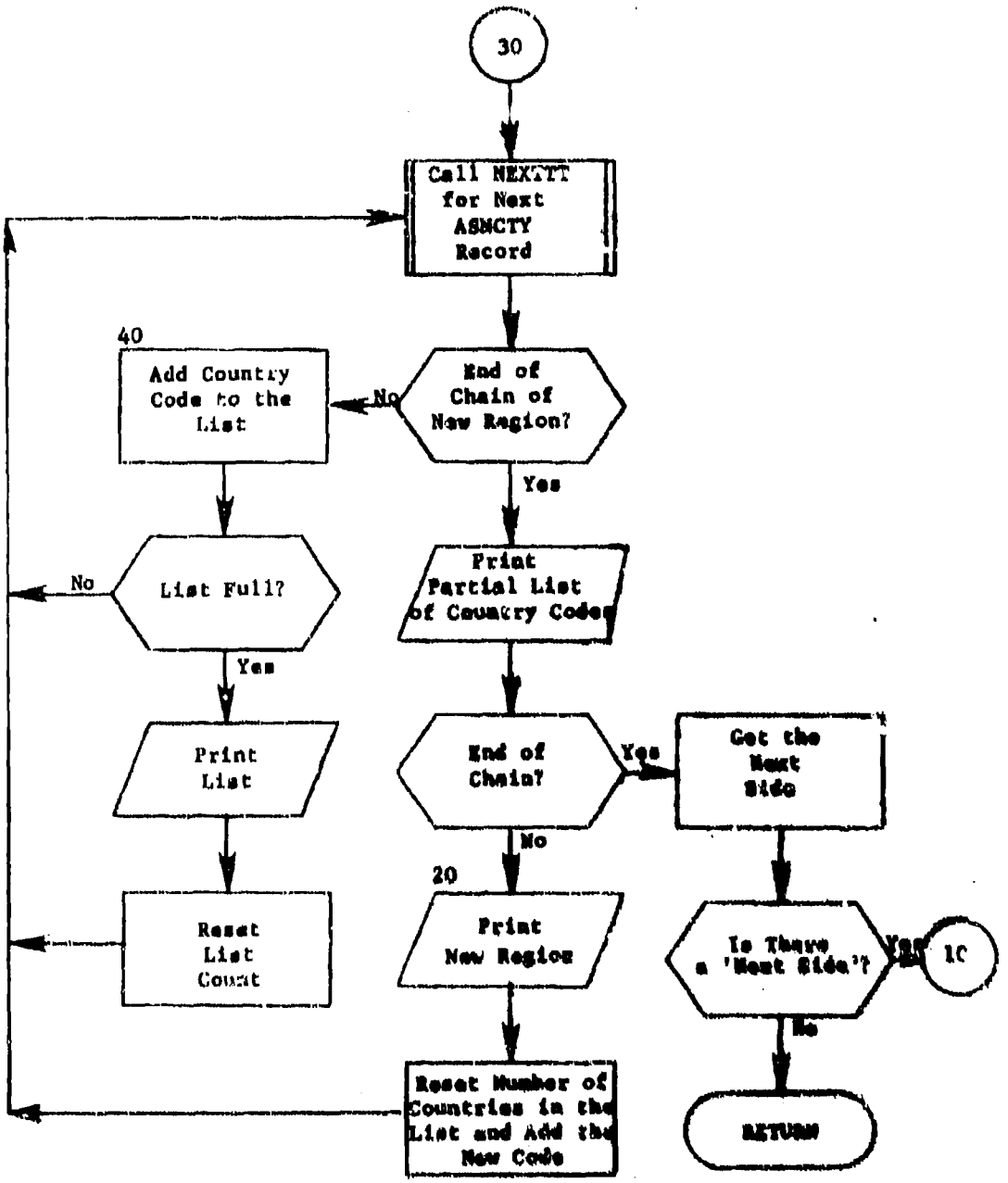


Figure 8. (Part 2 of 6)

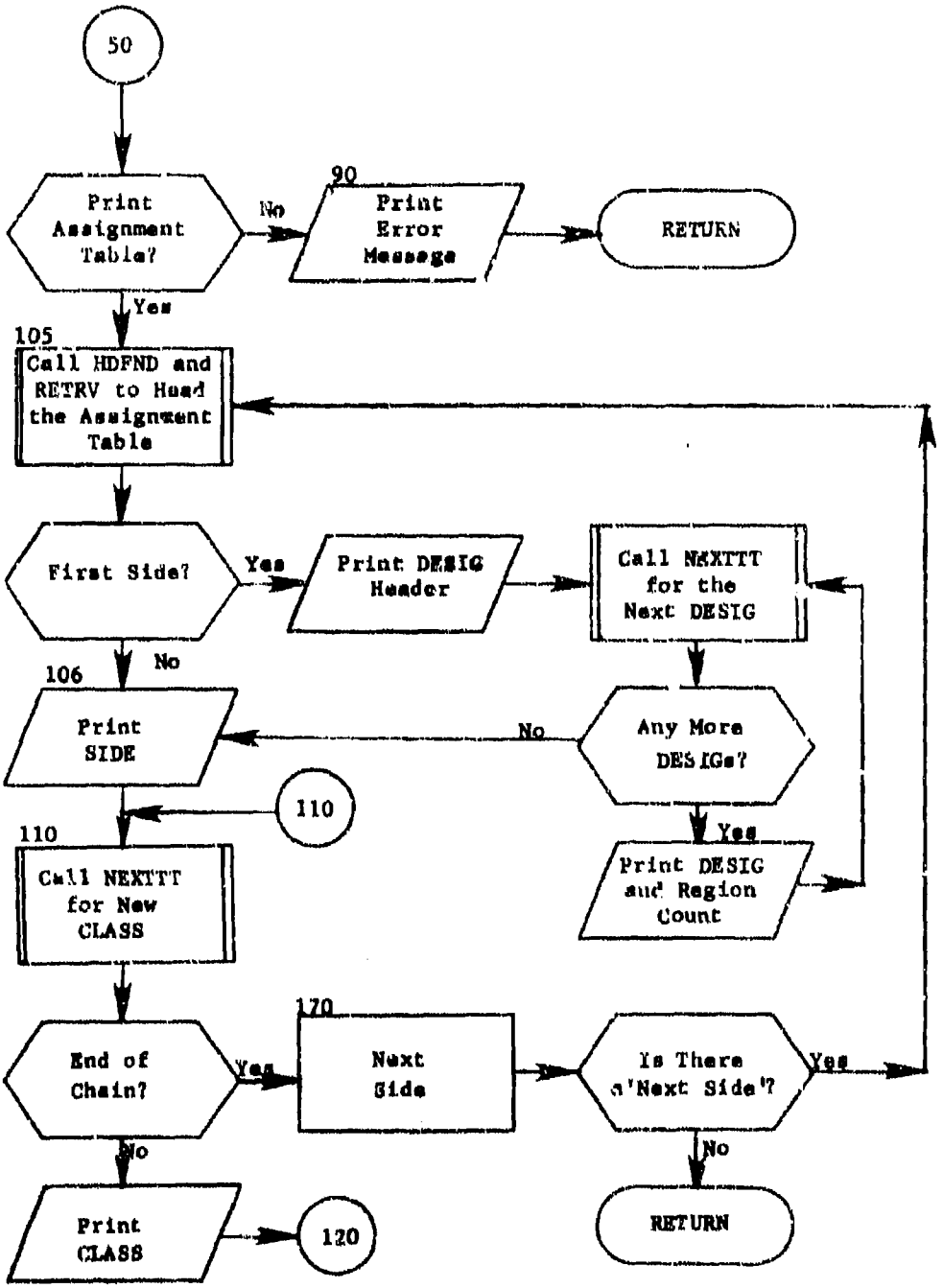


Figure 8. (Part 3 of 6)

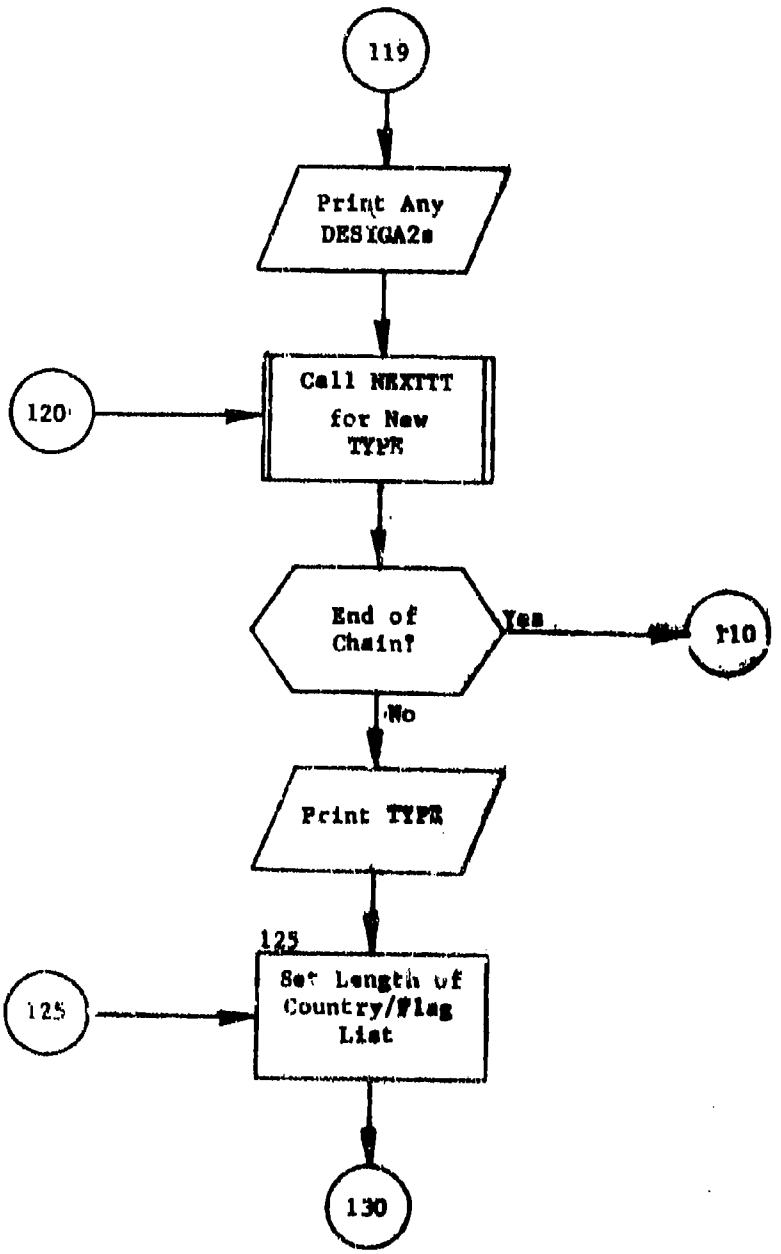


Figure 8. (Part 4 of 6)

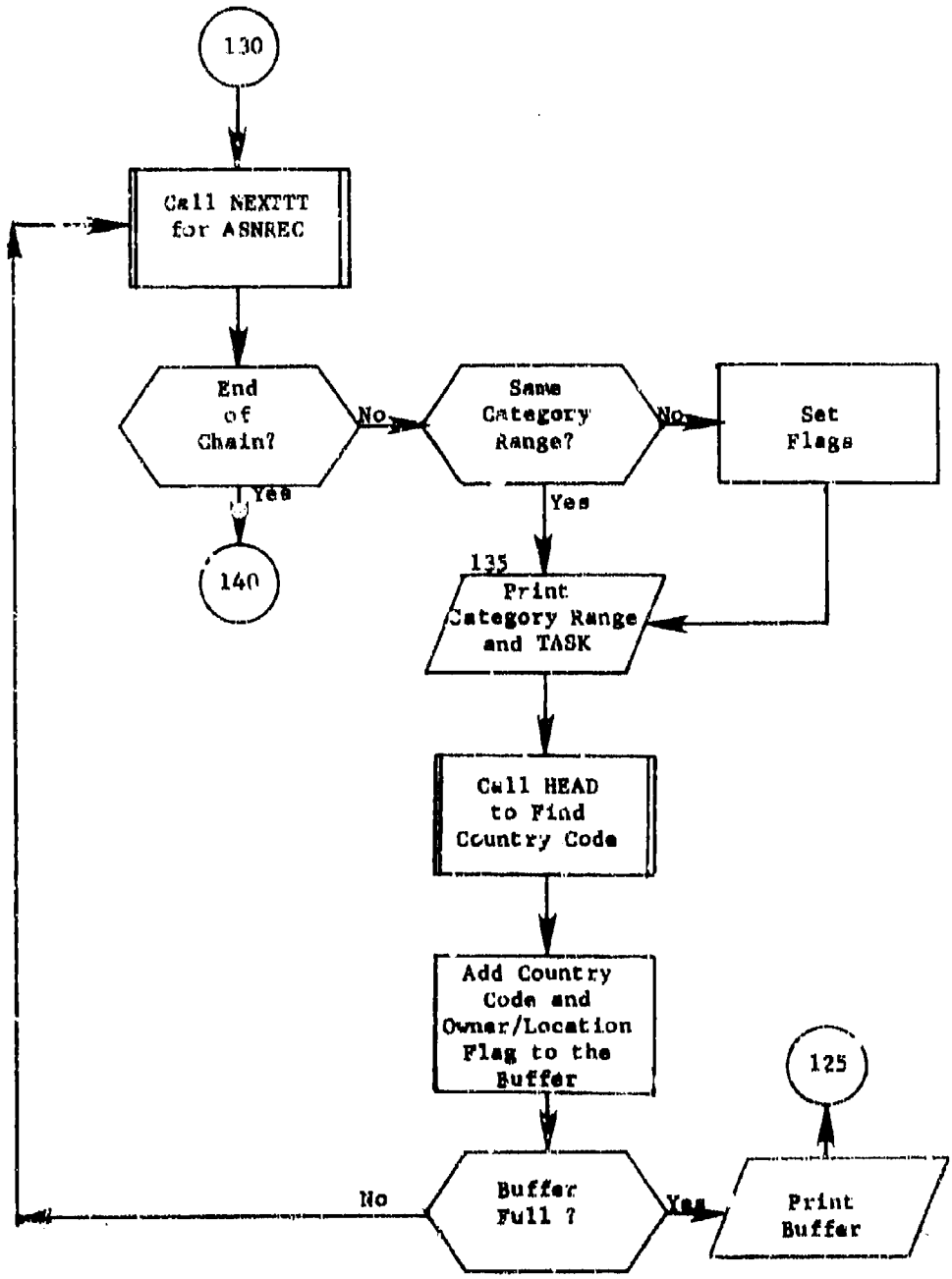


Figure 8. (Part 5 of 6)

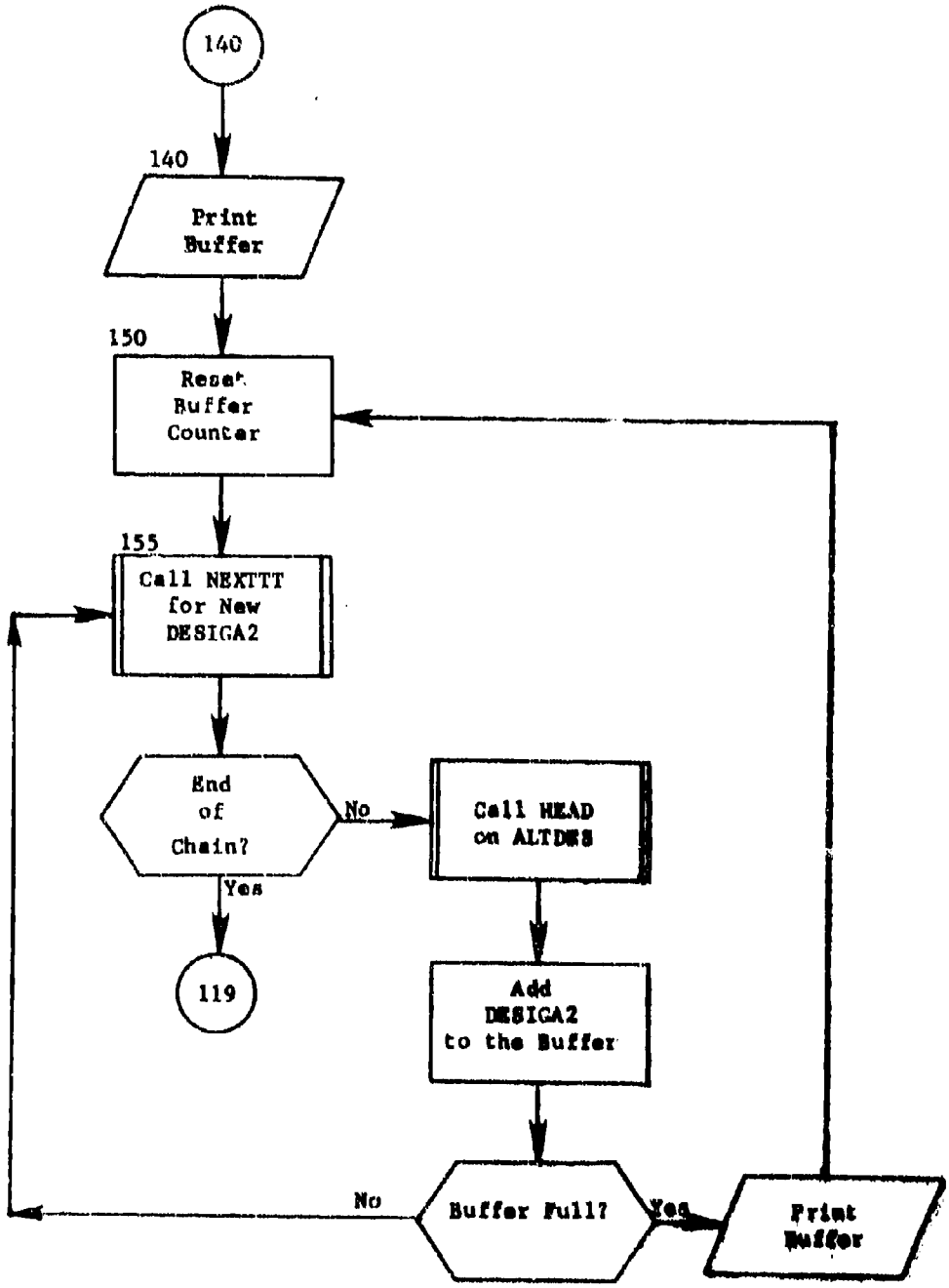


Figure 8. (Part 6 of 6)

2.9 Subroutine SELECT*

PURPOSE:	Select targets records from input JAD file
ENTRY POINTS:	SELECT
FORMAL PARAMETERS:	None
COMMON BLOCKS:	ASNKEY, CLASSES, C10, C30, JADREC, OPTION, PRINSP, SIDES, TARDEF
SUBROUTINES CALLED:	ADTOBASE, FINDCLASS, FINDSIDE, FMEDIA, FNDTAR, INSGET, KOMPCH, KRUNCH, NEXTTT, RANSIZE, SAMSET, SETDEF, SORTIT, XWHERE
CALLED BY:	ENTMOD (of JLM)

Method:

Target records are selected from the input JAD files, added to QUICKs integrated data base (subroutine ADTOBASE) and a new output JAD file generated for use outside of the QUICK system (see figure 9). The selection process is user directed and outlined in subsection 2.5.

Besides the commons described in table 1 there are some special arrays used by SELECT. VADV holds the value for all legal adverbs. ADVPTR holds the adverbs and pointers from the input clauses. START holds the pointer to the beginning of the input clauses in the order described in VADV. PUNCT holds the pair of value pairs that correspond to the alphabetic prefix and the operator comma as they appear in the input. RECORD is a 336-character array which contains the JAD record before it is decoded into JADREC. LINES contain an incremented count of the items in 'subsets', based on CLASS.

First a JAD format record is read. If the automatic generation of bomber defenses is desired and the record is a SAM site, SAMSET is called to save it on a special file (unit 19). The target then must meet the criterion defined by the Assignment table as tested by FNDTAR. The record must then pass any restrictions in the optional WHERE clause before it is decoded into JADREC. At this time the key reference code (ASNKEY) from FNDTAR is saved on the record which is put on a random file. The sortkey for this record is written onto file 21. This process continues until the input file is exhausted.

*First subroutine of overlay link SELE

The next step is to sort the random file and optionally create SAM complexes. The sort is done by subroutine SORTIT with the output defined on unit MIDLUN (21 or 23). The sort is primarily on DESIGA2 (found after FNDTAR was called by retrieving the first DESIGA2 for this type).

The DESIGA2 is retrieved so that the output tape will be in DESIG sort without necessarily doing a second sort. The file of selected JAD records is also sorted on region for the same reason. The SIZE/CLASS index (JSETNO) is used to keep the classes together within region for the output file and also to improve the efficiency of ADTOBASE. Attribute TYPE is included in the sort primarily for ADTOBASE efficiency but also for the output print. The NAME of the target is primarily there for the INDEXER module. After the sort is accomplished, KRUNCH is called to form SAM complexes if the option is on. The sorted file is then read and SETDEF is called, if the option is on, and ADTOBASE called to add the record to the integrated data base. The value for DESIG is now defined and the JAD record is output noting if the DESIG remains in sort (DESIG overflow could cause it to be out of sort). This is done for the entire sorted file. If yet out of sort, the JAD output file is resorted and put into the output file. If the optional prints are desired the JAD output file is reread and printed.

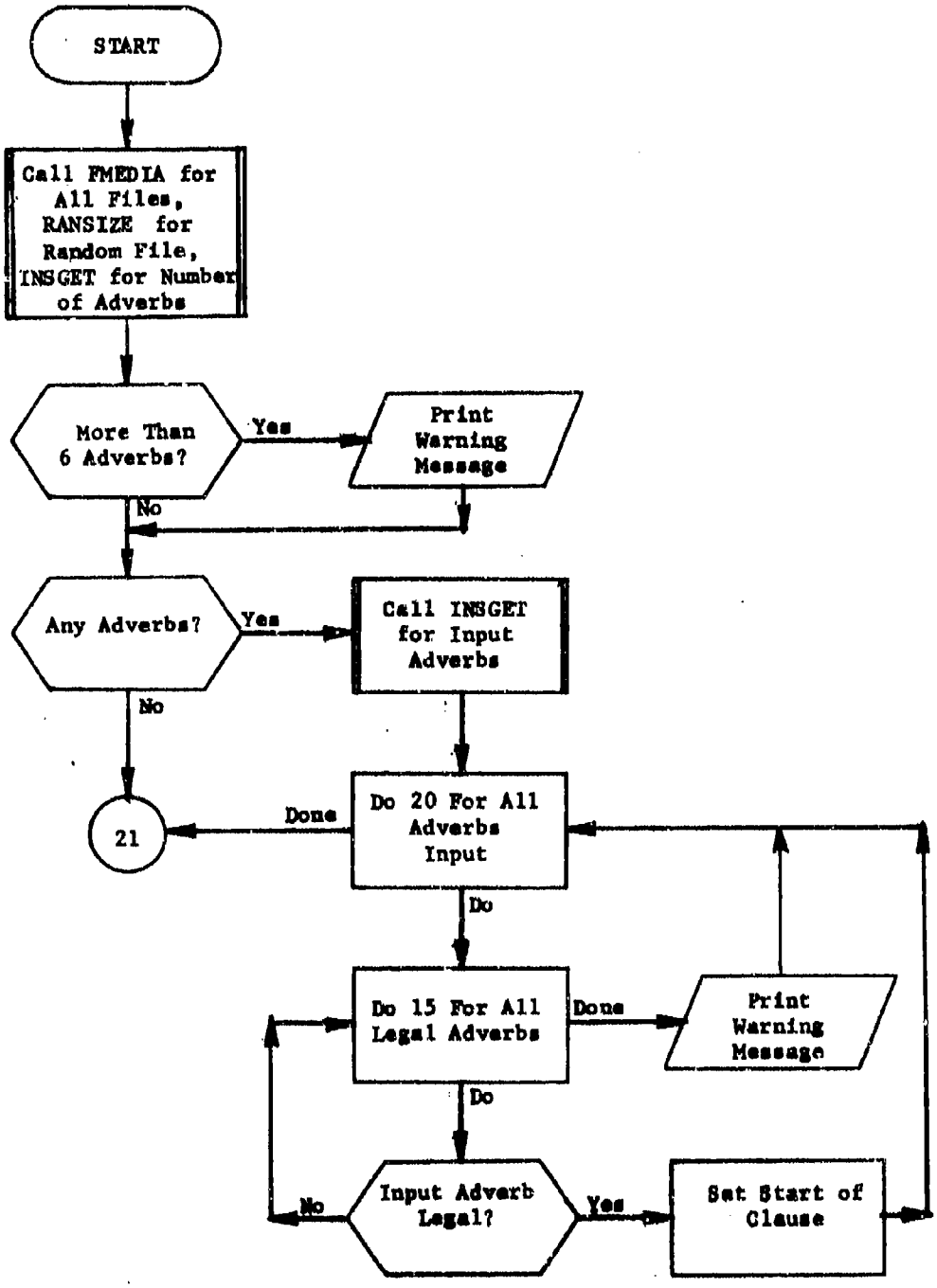


Figure 9. Subroutine SELECT (Part 1 of 6)

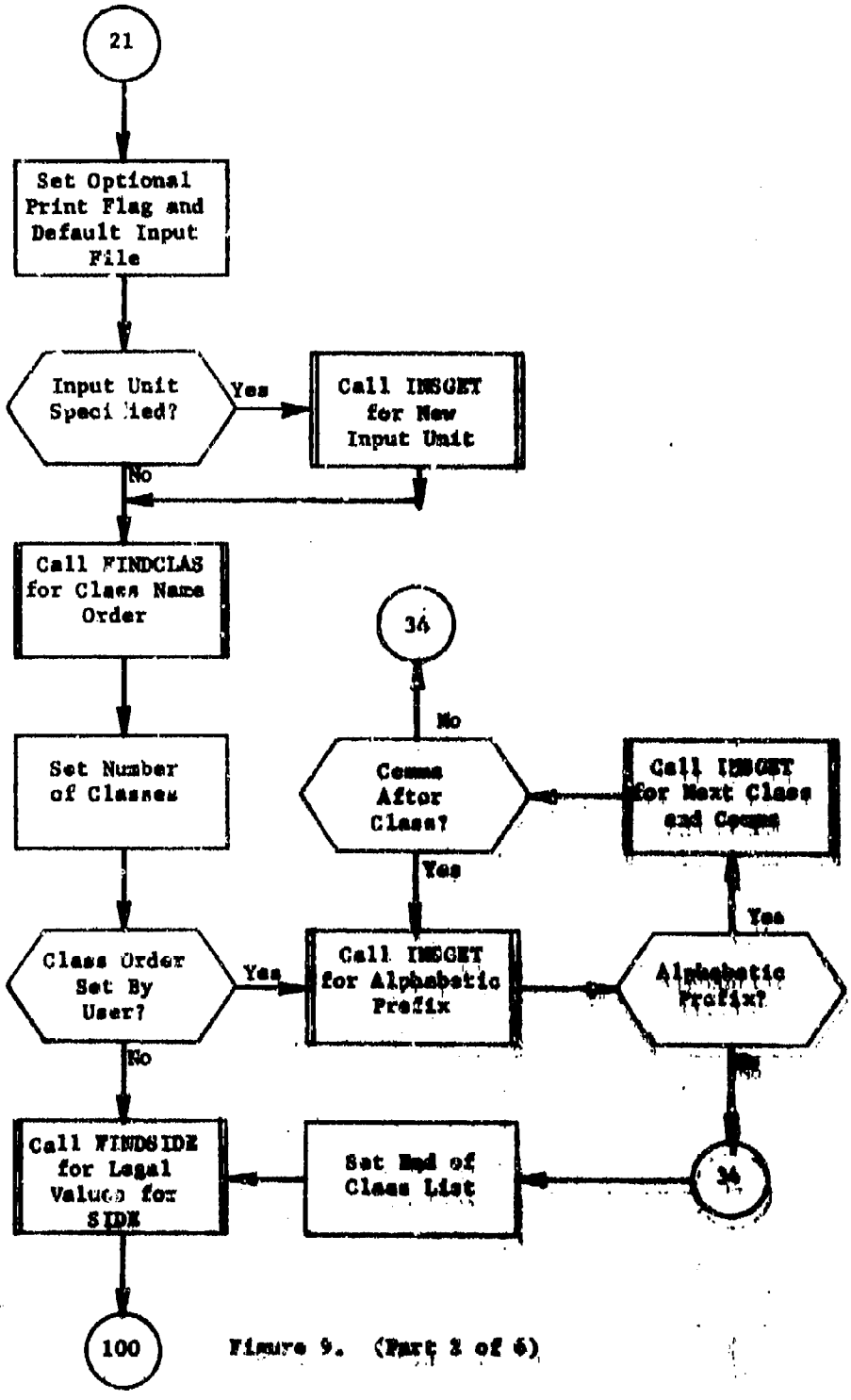


Figure 9. (Part 2 of 6)

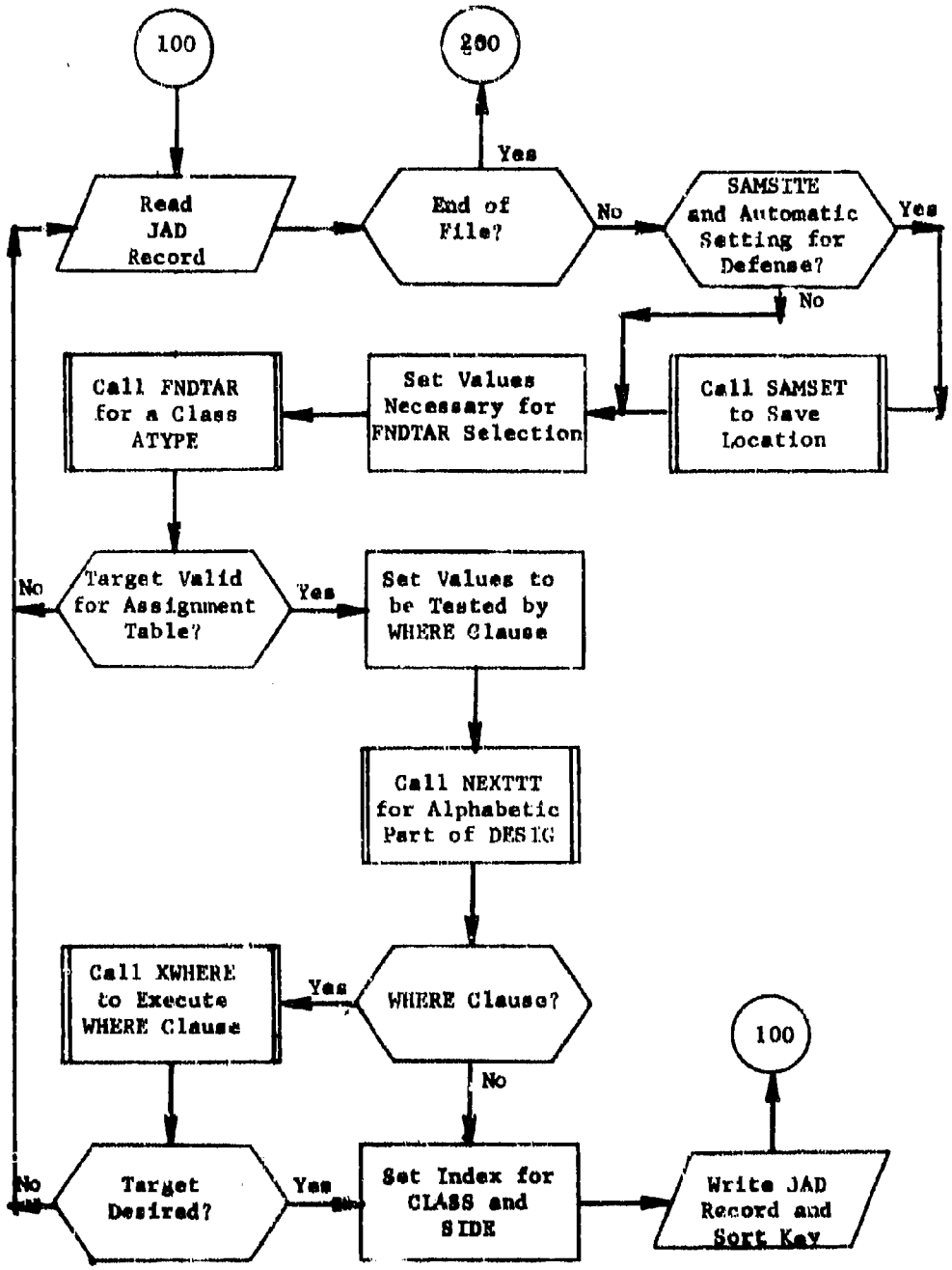


Figure 9. (Part 3 of 6)

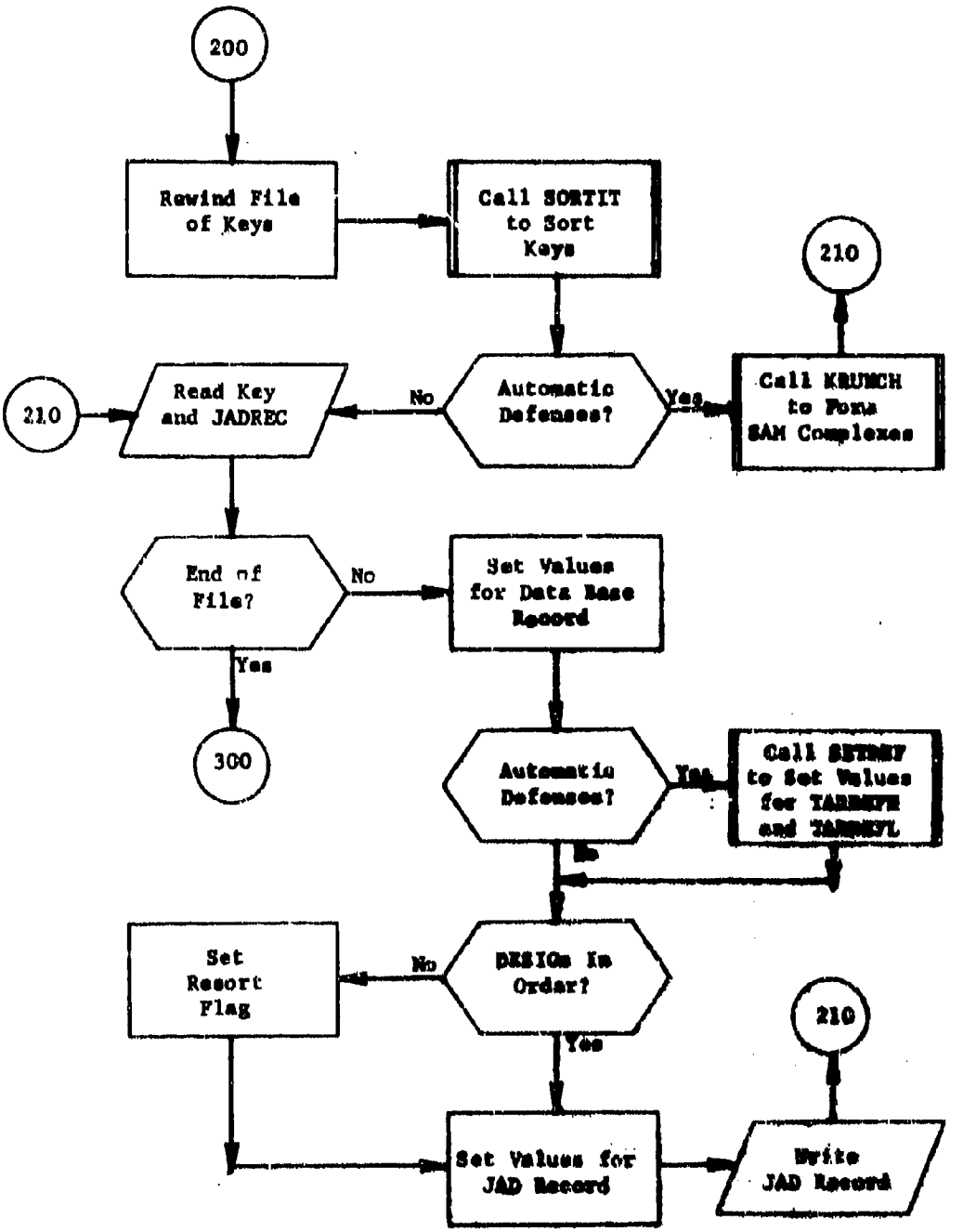


Figure 9. (Part 4 of 6)

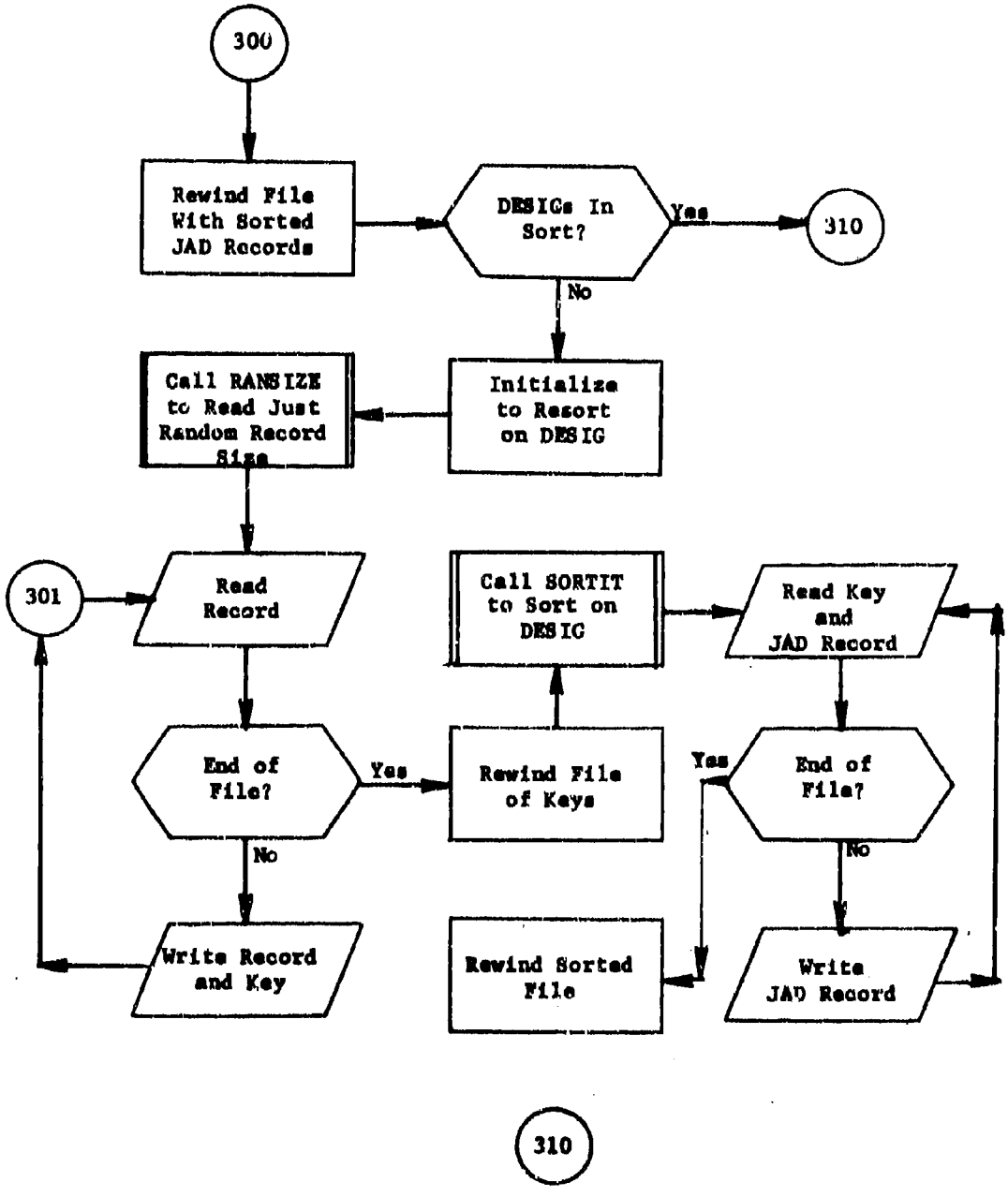


Figure 9. (Part 5 of 6)

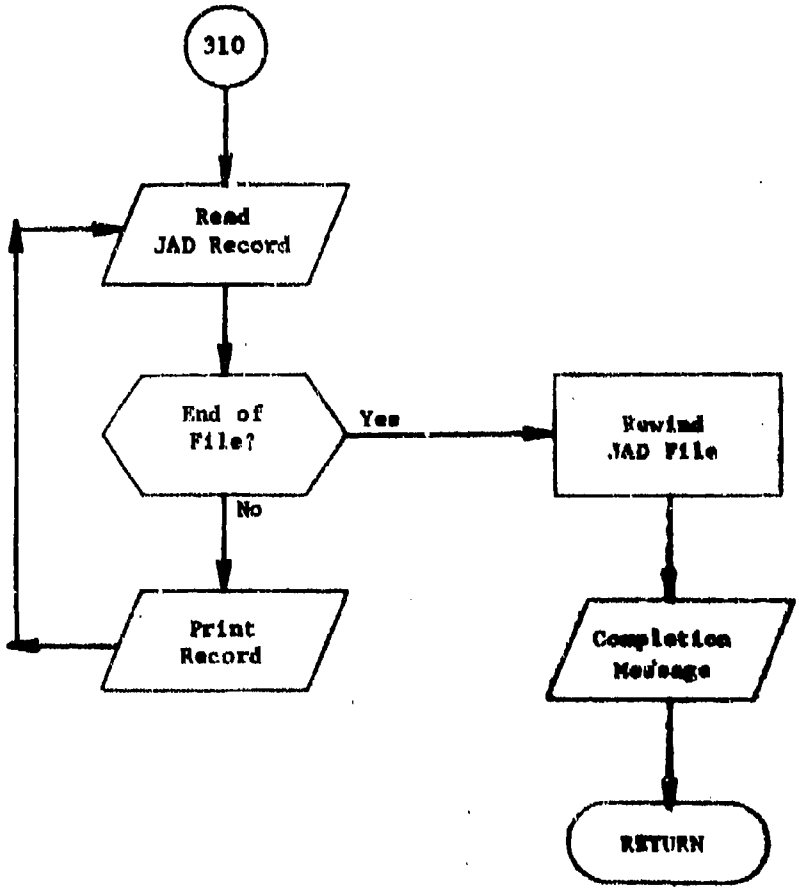


Figure 9. (Part 6 of 6)

2.9.1 Subroutine ADTOBASE

PURPOSE:	Add the selected JAD record to QUICKs integrated data base
ENTRY POINTS:	ADTOBASE
FORMAL PARAMETERS:	None
COMMON BLOCKS:	C10, C15, C30, JADREC, OPTION, TARDEF
SUBROUTINES CALLED:	DEFAULT, DLETE, GETDES, HDFND, HEAD, MODFY, NEXTTT, RETRV, STORE
CALLED BY:	SELECT

Method:

ADTOBASE is called to put the target described in common JADREC into QUICKs Integrated Data Base. In order to do this, the master of all the chains of which the TARGET Record is a detail of, must be found or in many cases created. Since the processing is very similar for most of the chains, the process has been generalized as much as possible. The general approach is to start with the headers and query the chains until the proper masters have been found or created for this TARGET record.

The first time that ADTOBASE is called the records on the ATRIB chain are checked in order to define the common C30 locations of the key attributes describing the master records. This is only done on the first call.

The first action normally performed in ADTOBASE is to define the array of values that describe the location in the data base where the TARGET record is to be linked. The linkage must be placed under the chains that contain the matching values for attributes VULN1, TYPE, IREG, and CLASS. If the TARGET is a weapon, chains for weapon group, payload number and the weapon TYPE (defined under the weapon CLASS definition) must be searched for matching values.

The first step in retrieving the master records is to retrieve any headers that have not been retrieved already by the previous target.

Then the four chains (seven for weapons) are searched for a record that has a key value matching that of the JAD record. If the record agrees with the one previously found, no searching is necessary. If the record does not exist, then it must be created.

Before any record is created in ADTOBASE the subroutine DEFAULT is called for that record type. DEFAULT will set all the default values for that record type. After this is done, the key value from JADREC are set and the record is created.

The SECTOR chain requires special processing since the comparison here is not a match but longitude being between two values. The chain is simply searched for the record. This is not necessary if the record is not already there from the previous call to ADTOBASE.

The TARGTY record must also be checked since it must be unique to more values than just TYPE. The values of country location and owner and vulnerability are also checked. If a match is not found on the entire chain, a new record is created. Note also that if the record had been created earlier it must now be modified to account for the additional unique values (VULN1, etc.).

For weapons there is one additional record that must be defined now, WEPSUB, the weapon subtype. As usual, if it does not already exist it is created. Note that DEFAULT is not called since the entire record consists of the one name 'DUMMY'.

Now if either the REPLACING or OMITTING options are used the TGTTGT chain is searched for targets with the same values for WACNO and BENO. Note that for some types of targets, this can result in an excessive amount of searching and that the options should be used sparingly. If a match is found OMITTING will cause a return to SELECT while REPLACING will delete the old record.

Now that the masters for the TARGET record have been defined the actual record can be added to the data base.

First subroutine GETDES completes the work that FNDTAR started using the reference code mentioned in ALPHAS, GETTAR, and SELECT.

With the DESIG now defined, and the default values defined, the values in JADREC are moved to common C30.

For Urban/Industrial targets the radius is defined and the value for either POP or IGIW is set (POP for population centers).

The target record is now created.

If the target is also a weapon then a MSBMTG record is also created. At this point ADTOBASE returns.

Subroutine ADTOBASE is illustrated in figure 10.

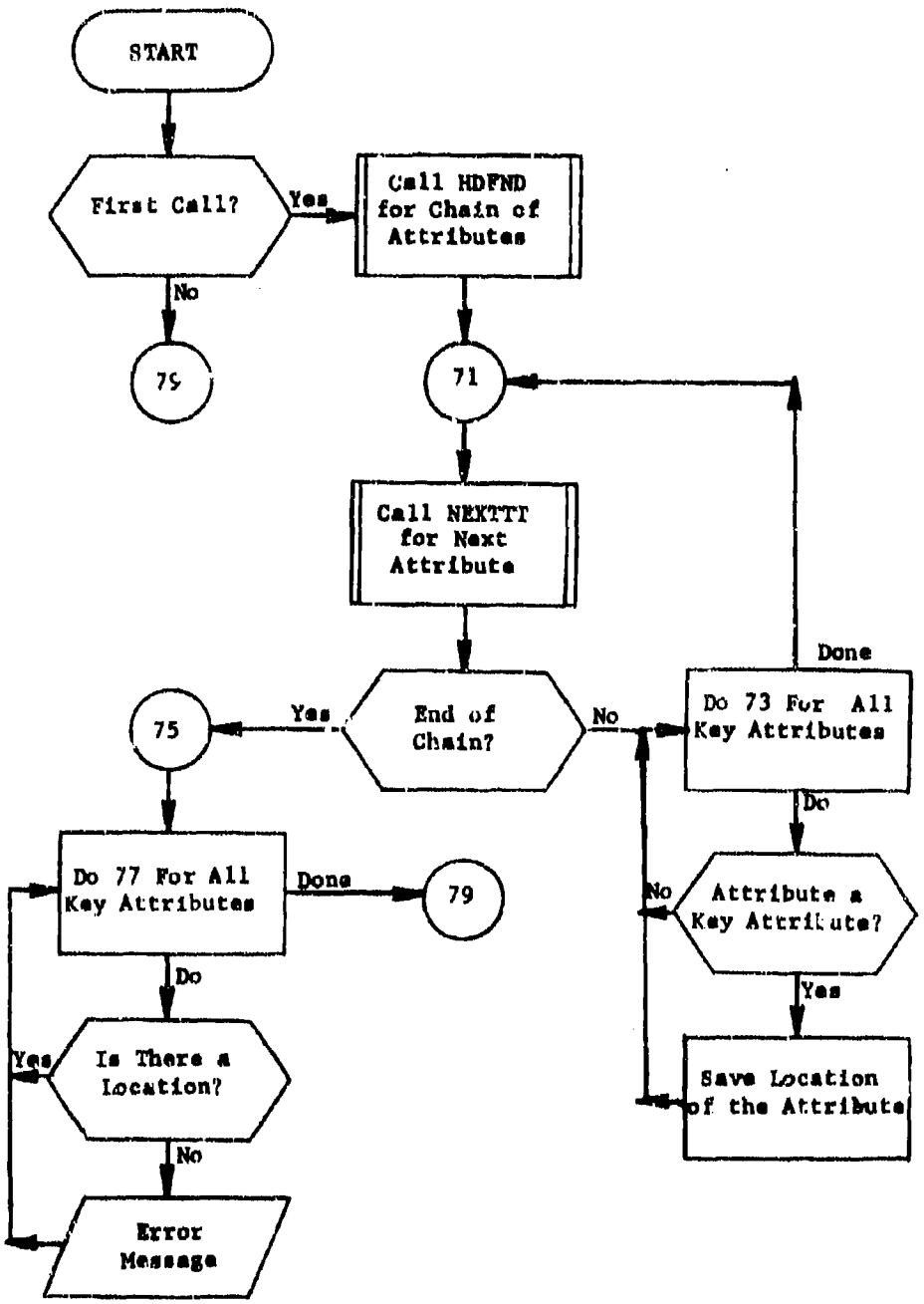


Figure 10. Subroutine ADTOBASE (Part 1 of 7)

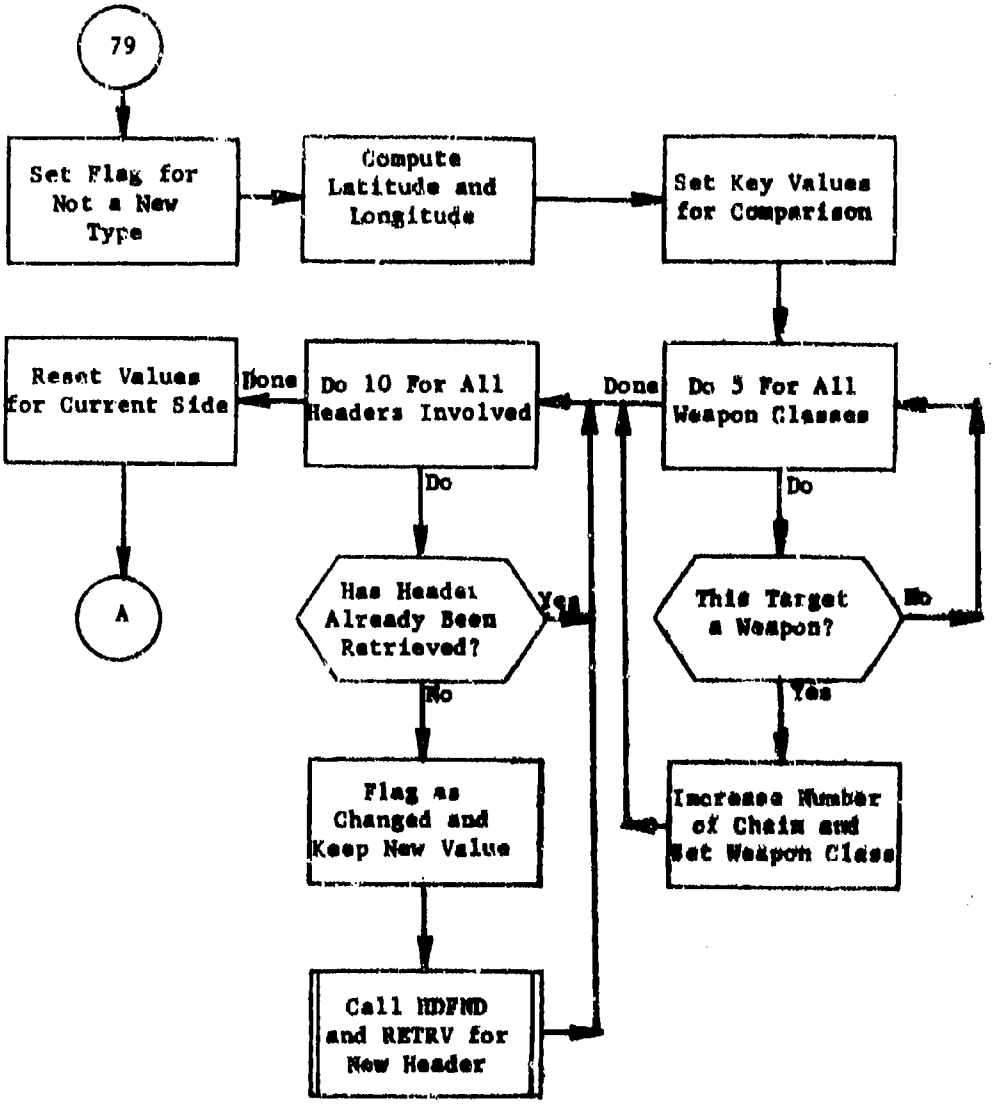


Figure 10. (Part 2 of 7)

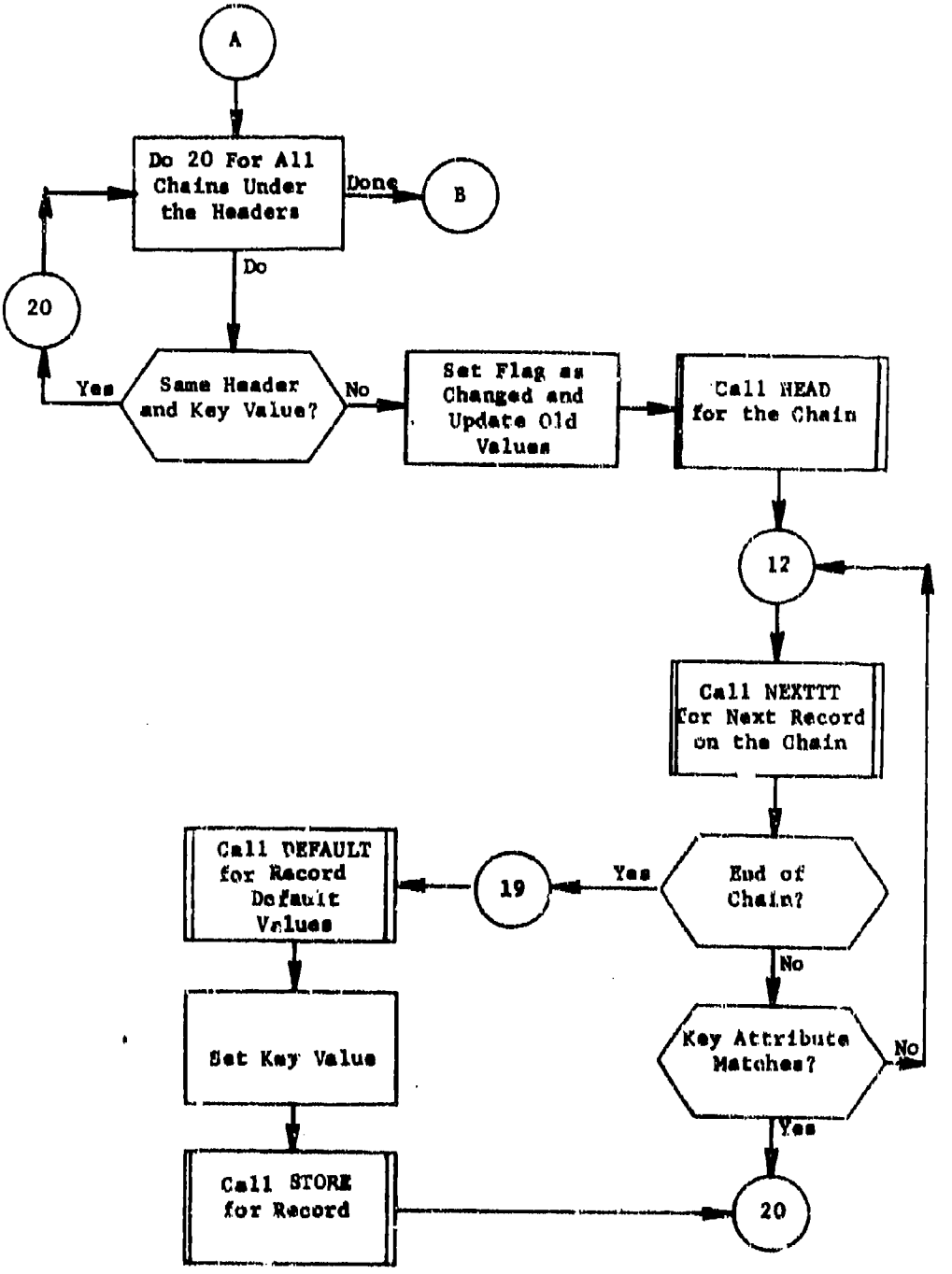


Figure 10. (Part 3 of 7)

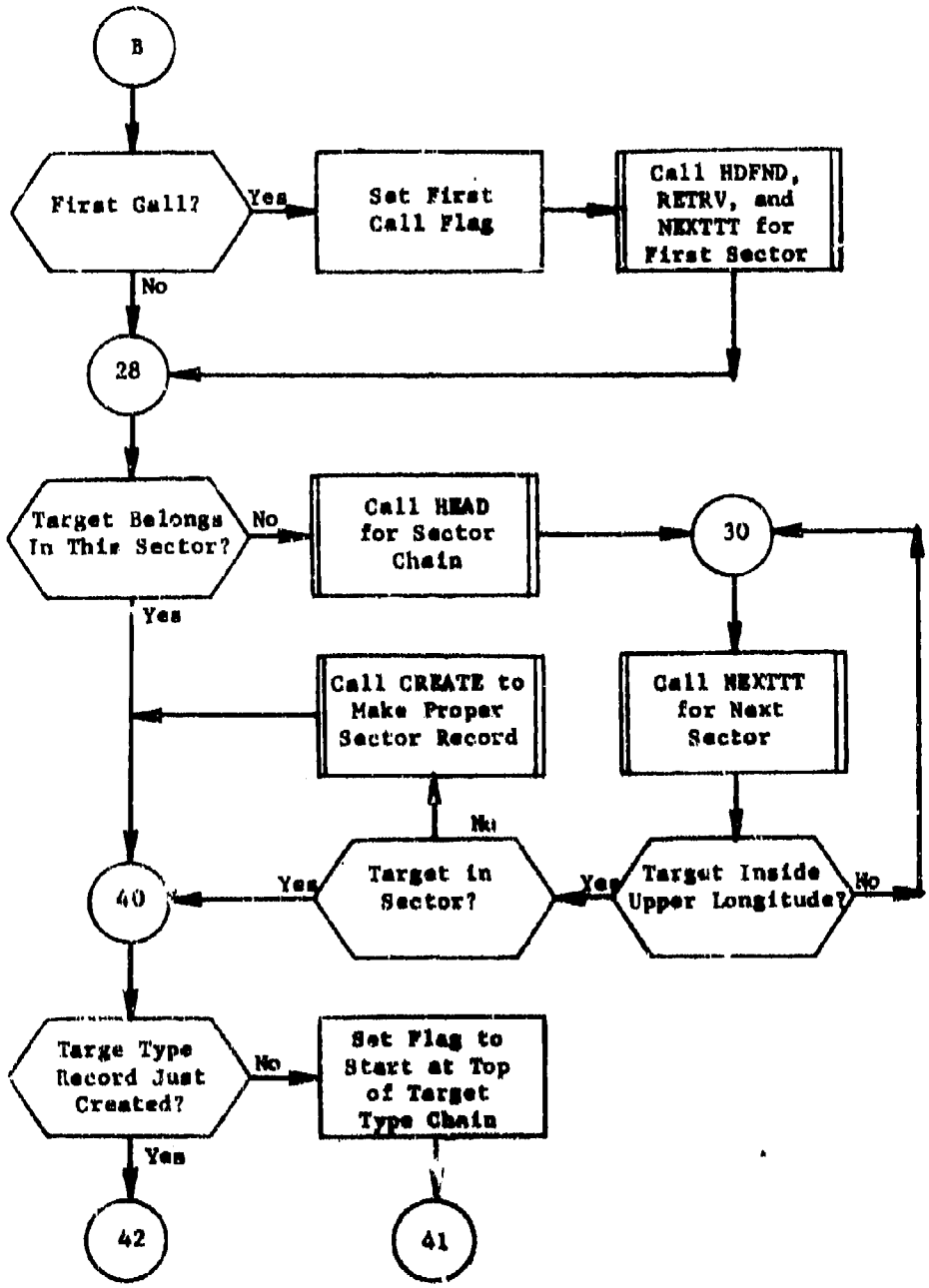


Figure 10. (Part 4 of 7)

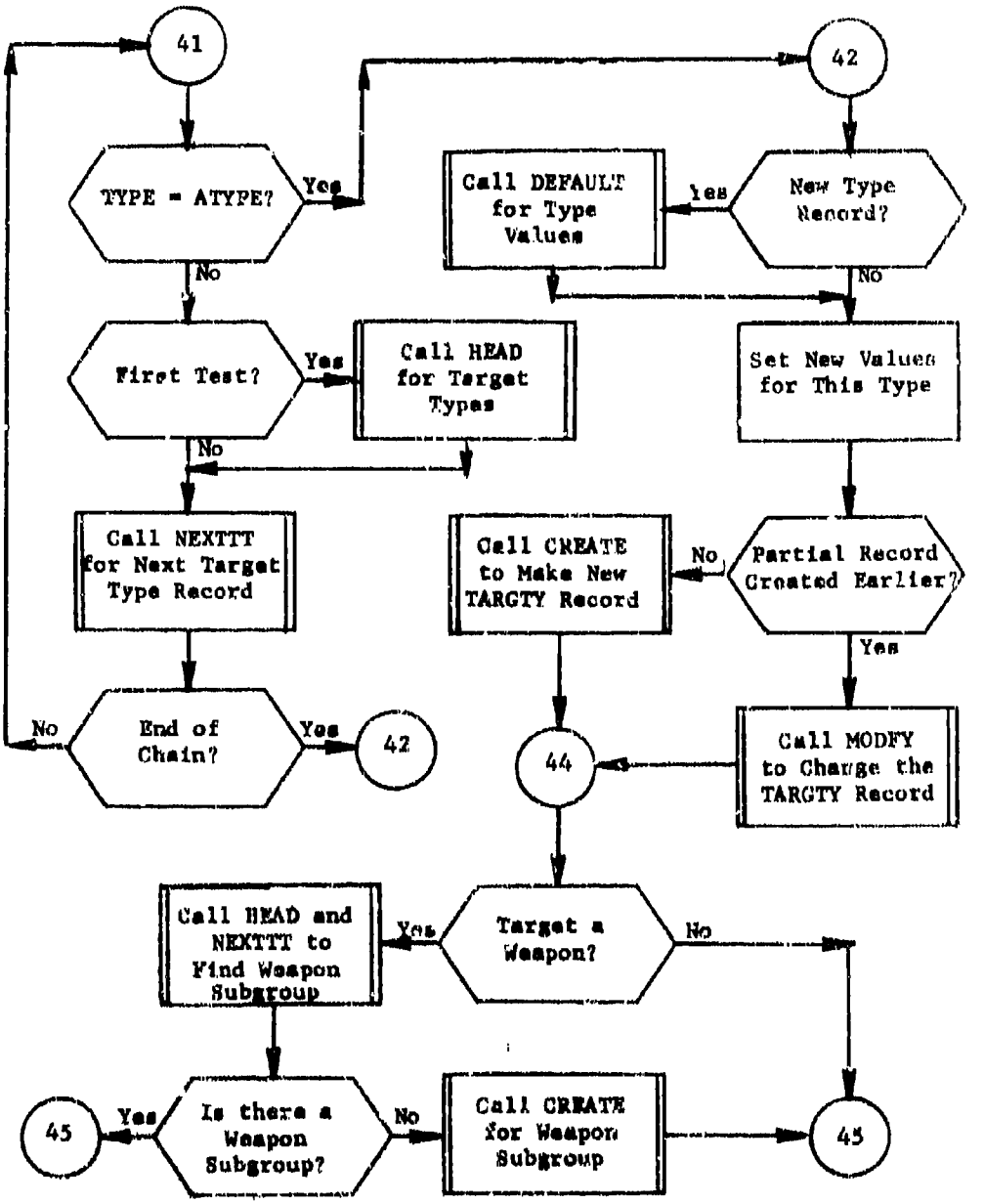


Figure 10. (Part 5 of 7)

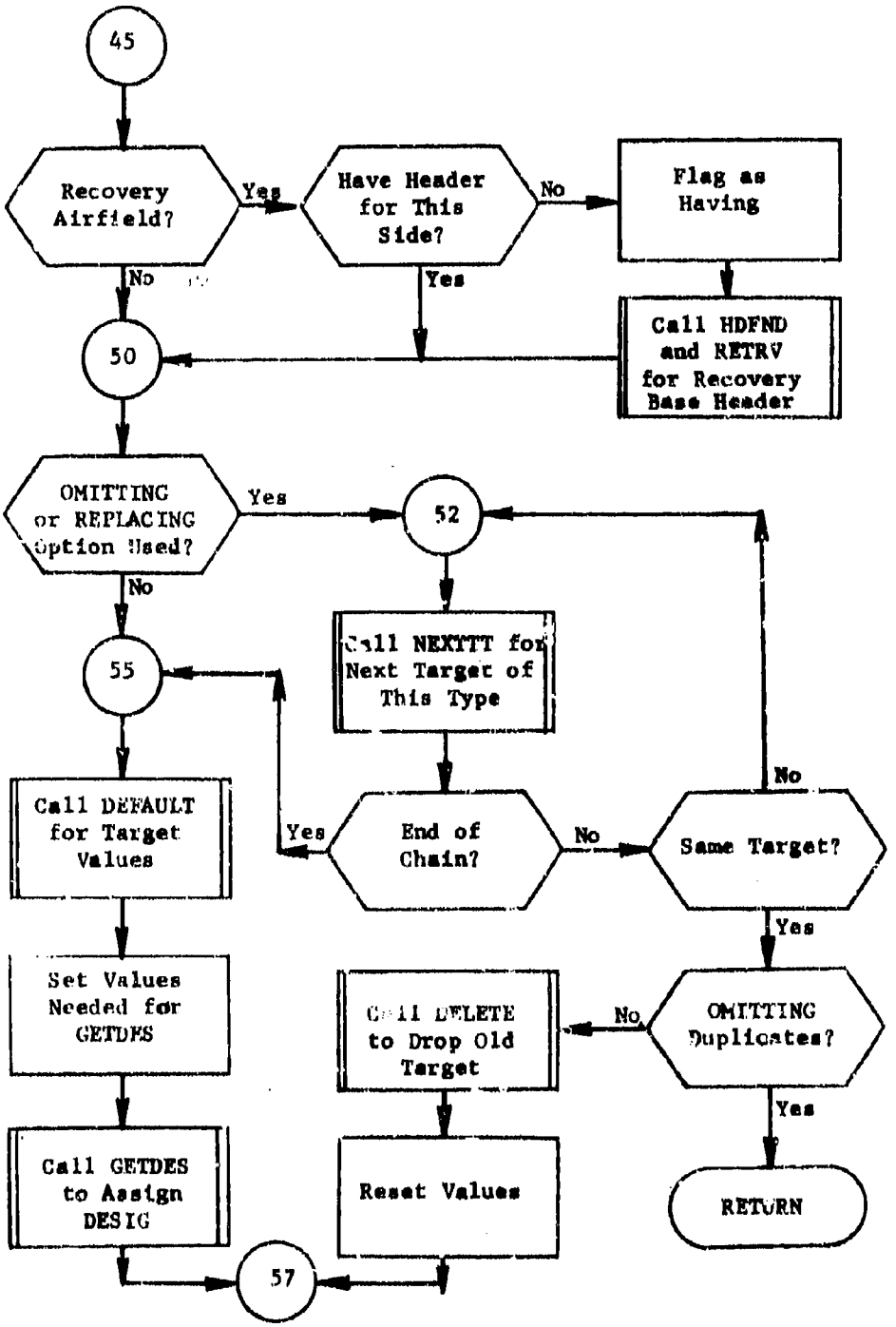


Figure 10. (Part 6 of 7)

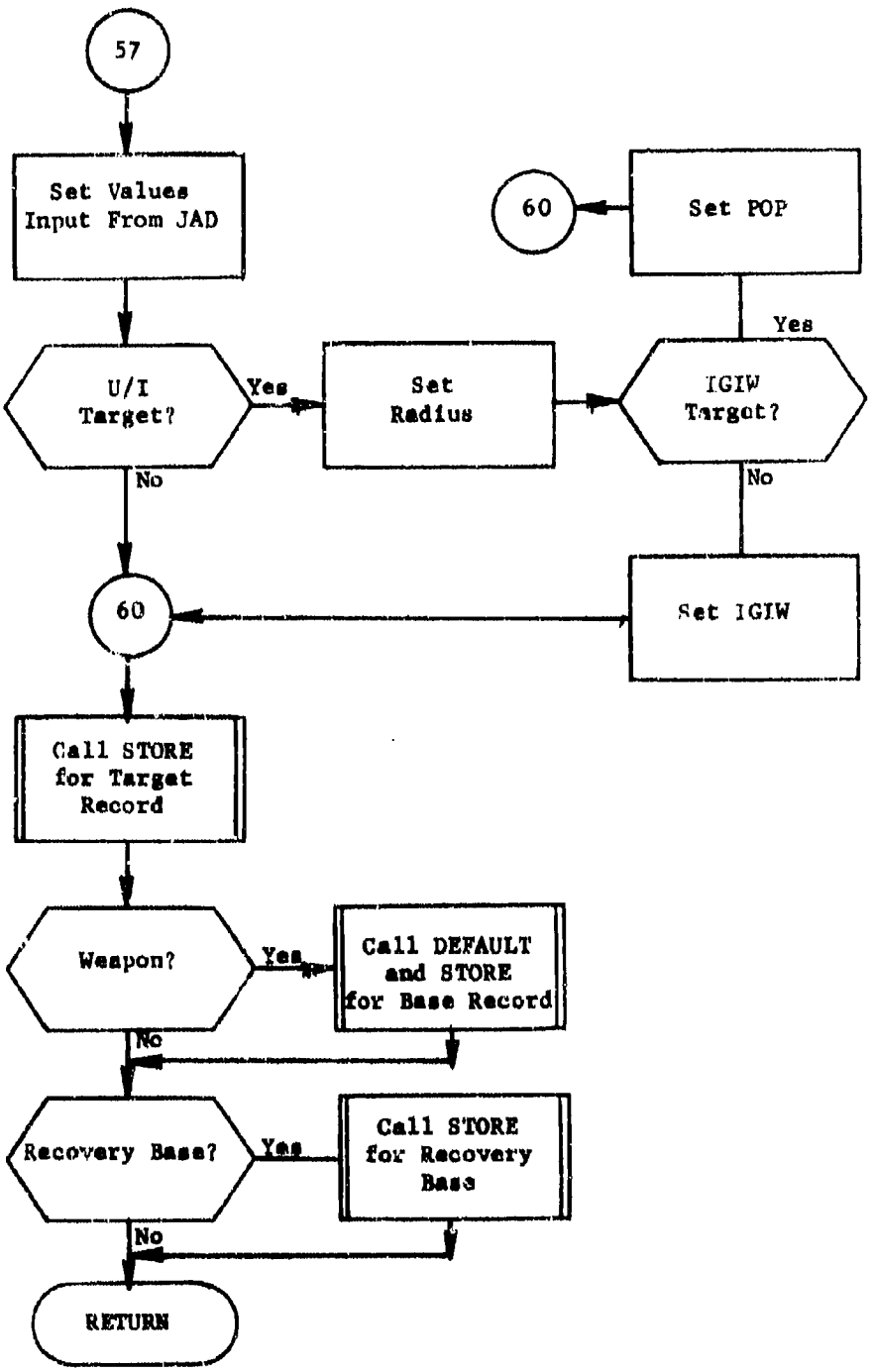


Figure 10. (Part 7 of 7)

2.9.2 Subroutine DEFAULT

PURPOSE: Set default values into common block C30

ENTRY POINTS: DEFAULT

FORMAL PARAMETERS: RECORD: record name

COMMON BLOCKS: C10, C20, C30

SUBROUTINES CALLED: HEAD, NEXTTT

CALLED BY: ADTOBASE

Method:

Local arrays RECNAME and VALUE hold the record name and the default value for the attribute. Also arrays LINK and LOC contain either a link to the start of the record description (0, if no more records are described) or the location of the attribute in common C30.

When DEFAULT is called, the RECNAME array is queried using the corresponding LINK value until the record name passed is found or the entire list queried.

If the record name is not found the RCTYP chain is traversed in an effort to find the record name passed. When found the IALINK and AILINK chains are traversed to find the default value for the attribute and its location in C30. These values are added to VALUE and LOC following the record name. In any case the values in VALUE and LOC between the record and the next record are used to move the value in VALUE into the location in common C30 specified by LOC.

Subroutine DEFAULT is illustrated in figure 11.

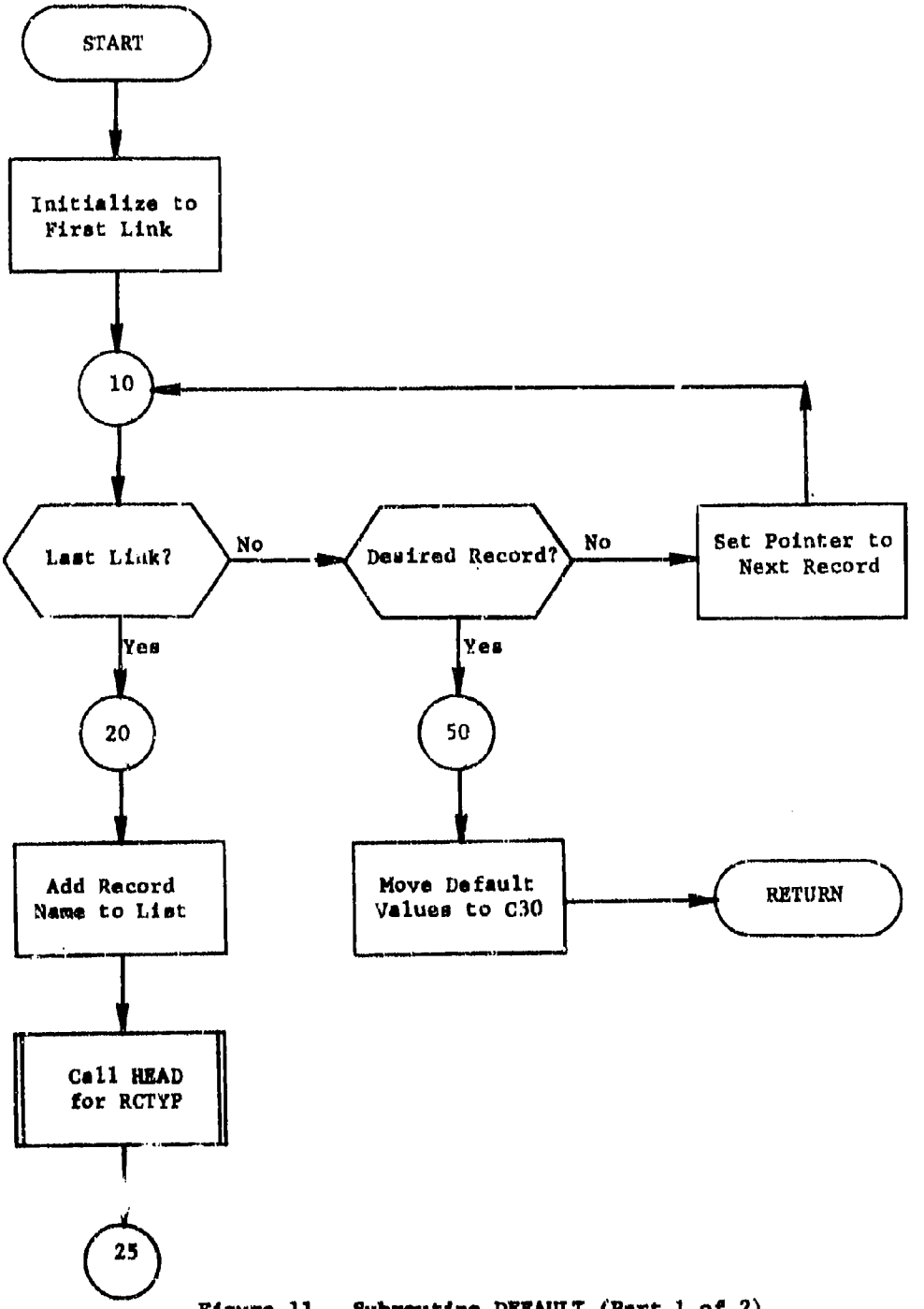


Figure 11. Subroutine DEFAULT (Part 1 of 2)

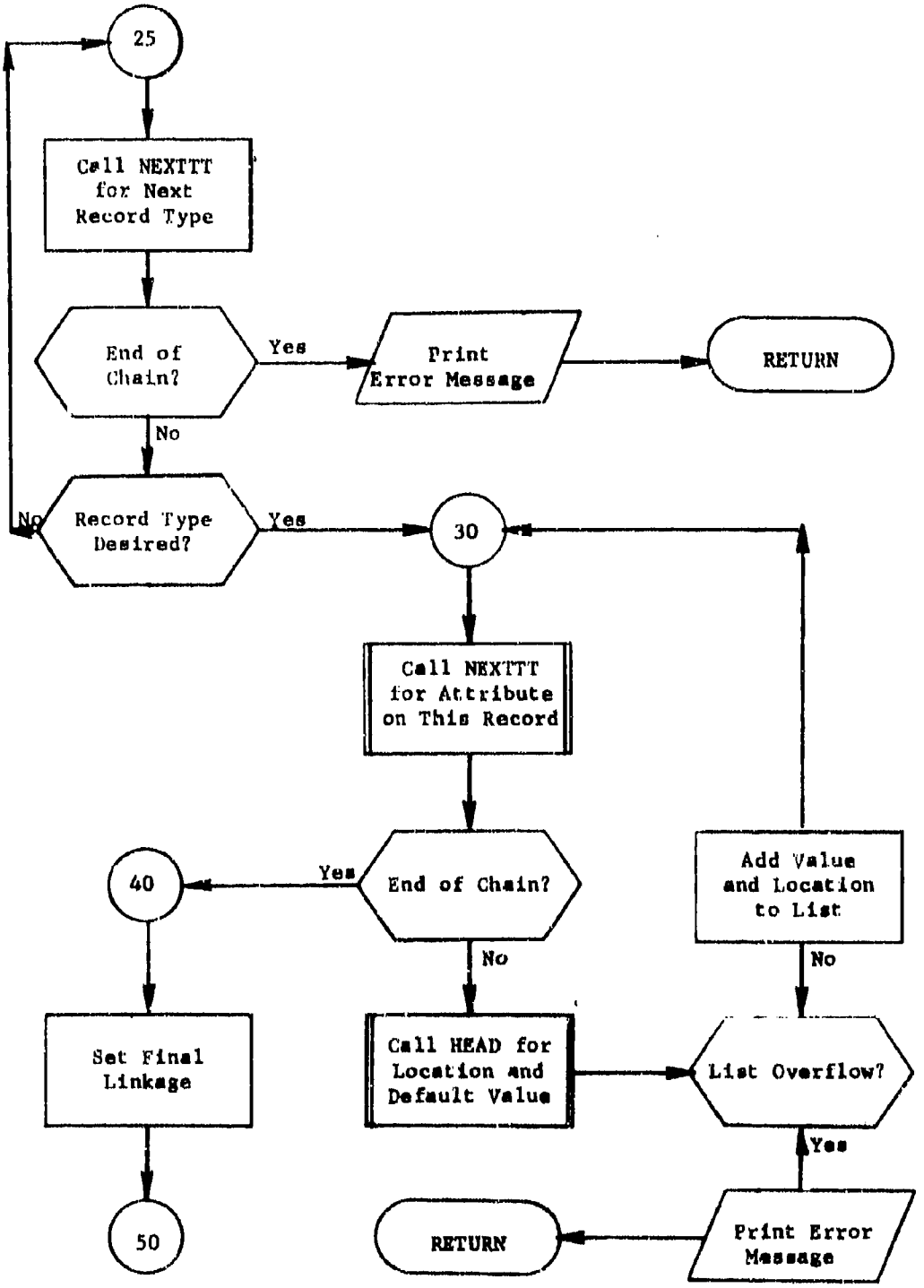


Figure 11. (Part 2 of 2)

2.9.3 Subroutine KRUNCH

PURPOSE: Form SAM complexes based on name with adjusted radius and range

ENTRY POINTS: KRUNCH

FORMAL PARAMETERS: SORT: the file containing sorted records from SELECT

COMMON BLOCKS: None

SUBROUTINES CALLED: SORTIT, DISTF

CALLED BY: SELECT

Method:

The information on file SORT is saved on a temporary file. The output from SAMSET is then sorted on NAME. Complexes with identical names are formed into complexes with an effective radius and range. The complexes are then resorted on longitude and an output file for SETDEF is created, the first record on this file is the number of complexes. Finally, the data that was on SORT file is returned to SORT.

Subroutine KRUNCH is illustrated in figure 12.

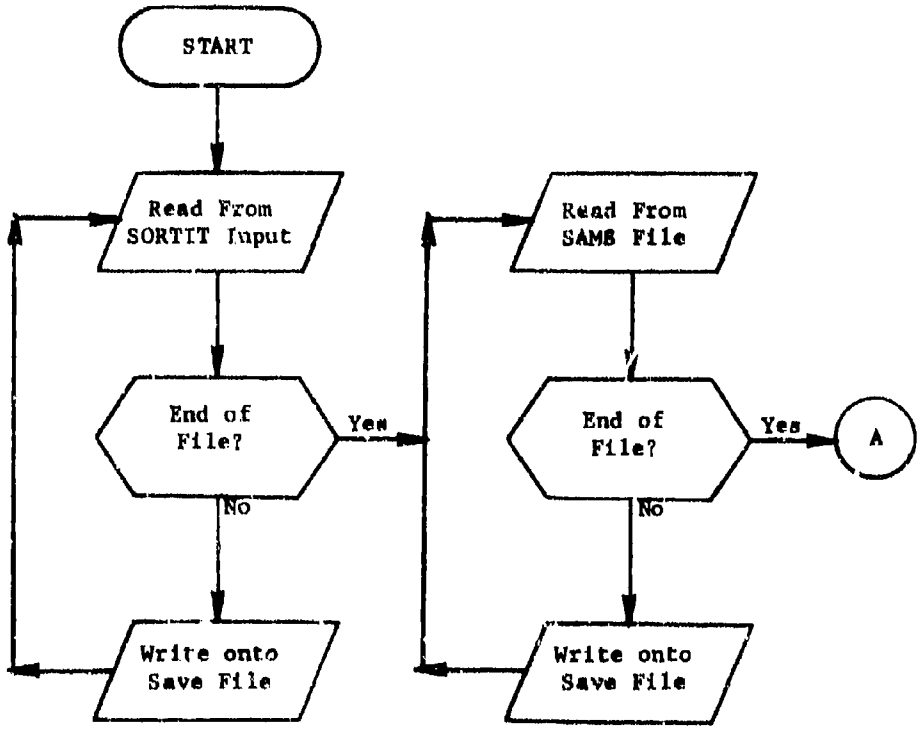


Figure 12. Subroutine KRUNCH (Part 1 of 3)

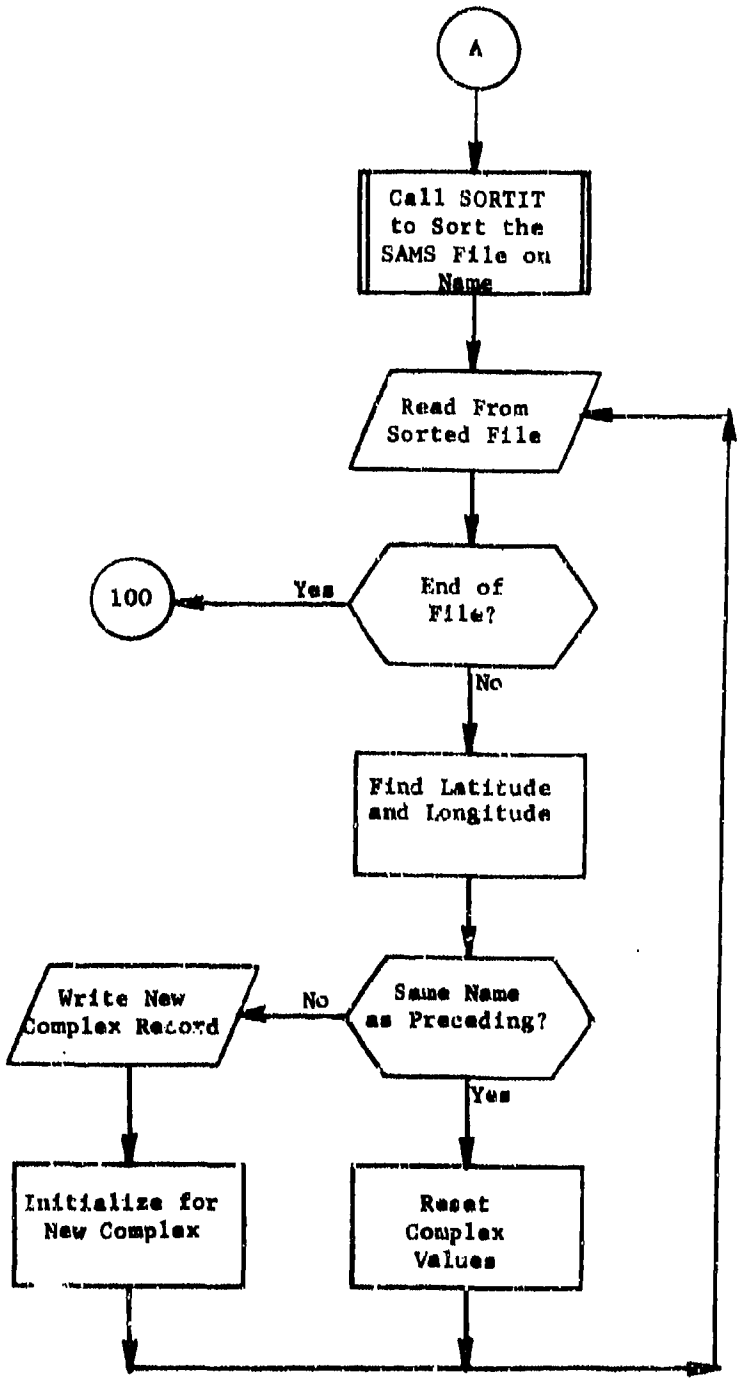


Figure 12. (Part 2 of 3)

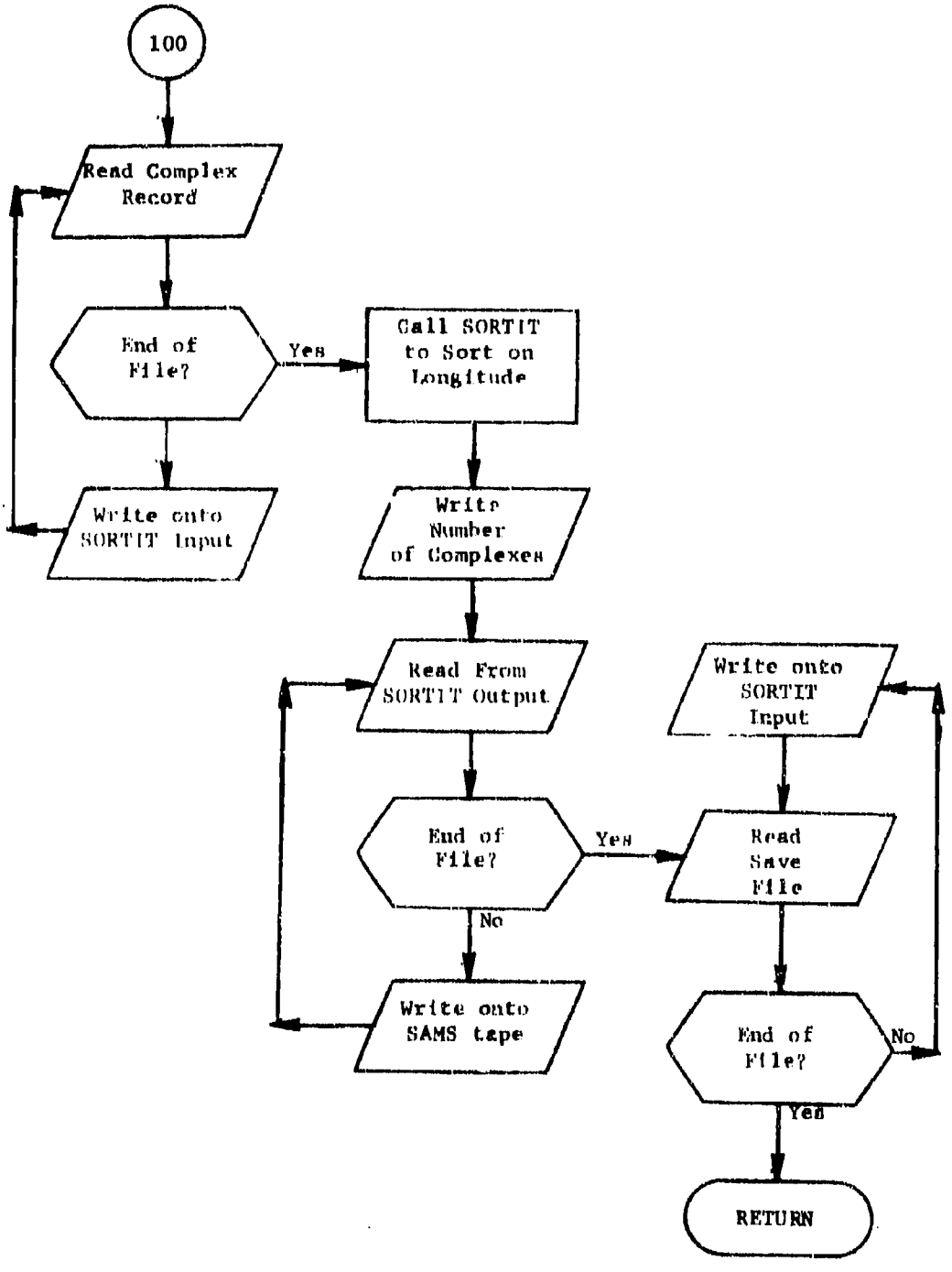


Figure 12. (Part 3 of 3)

2.9.4 Subroutine SAMSET

PURPOSE: Collect location and type of SAM site

ENTRY POINTS: SAMSET

FORMAL PARAMETERS: RECORD: 336-character JAD format record

COMMON BLOCKS: None

SUBROUTINES CALLED: None

CALLED BY: SELECT

Method:

SAMSET is passed the JAD format record of a SAM site. The name, latitude, longitude and type code is then put on a scratch file for later processing by subroutine KRUNCH.

Subroutine SAMSET is illustrated in figure 13.

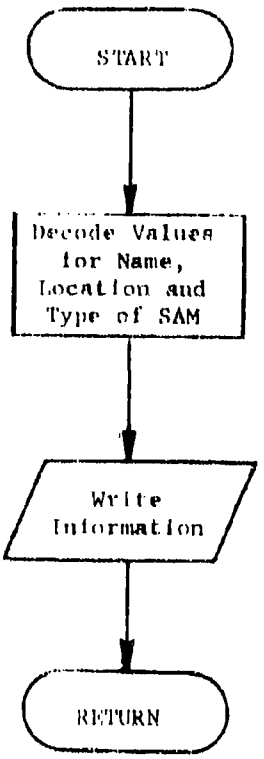


Figure 13. Subroutine SAMSET

2.10 Subroutine ASTERISK*

PURPOSE: To delete target records from QUICKs integrated data base

ENTRY POINTS: ASTERISK

FORMAL PARAMETERS: None

COMMON BLOCKS: C10, C30

SUBROUTINES CALLED: DLETE, HEAD, INSGET, RETRY

CALLED BY: ENTMOD (of JLM)

Method:

ASTERISK begins by locating the KEEPING clause in the INSGET input and retrieving the first range of DESIGs to be retained in the data base. The input tape (JAD format) created by an earlier execution of SELECT is then read. If the DESIG on the tape is beyond the DESIG input range, then INSGET is used to retrieve additional DESIG ranges until this condition is no longer true. The TARGET record is then retrieved and the TGTTGT chain is headed to find the value of TYPE which is used on the output tape. If the tape DESIG is below the DESIG input range, the TARGET record is removed from the data base and the JAD format record is output as is. But, if this DESIG is between the DESIG input range, then an asterisk is appended to the DESIG before the record is output. In either case a new tape record is read and the DESIG is tested as before.

Subroutine ASTERISK is illustrated in figure 14.

*First subroutine of overlay link ASTE

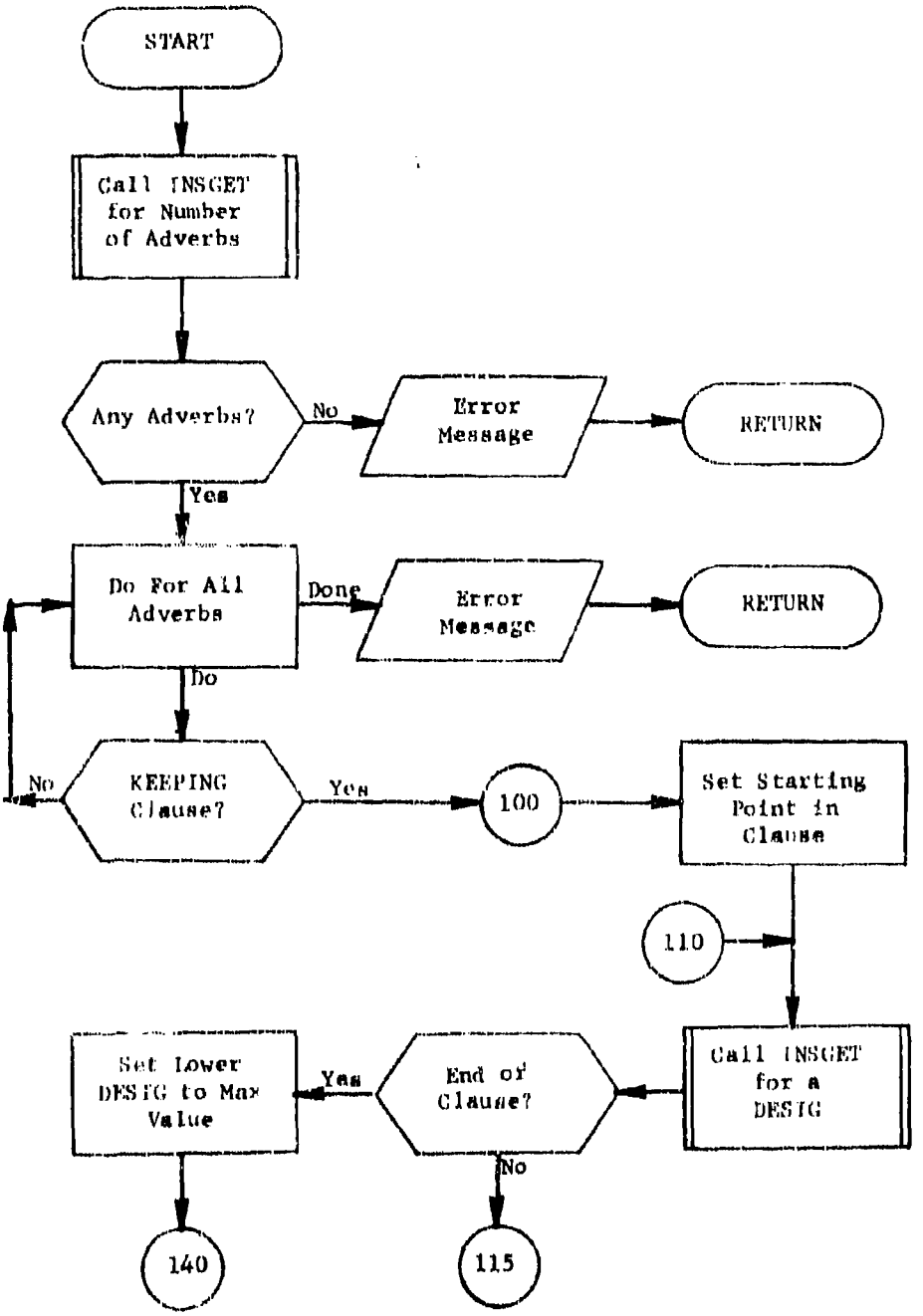


Figure 14. Subroutine ASTERI3K (Part 1 of 3)

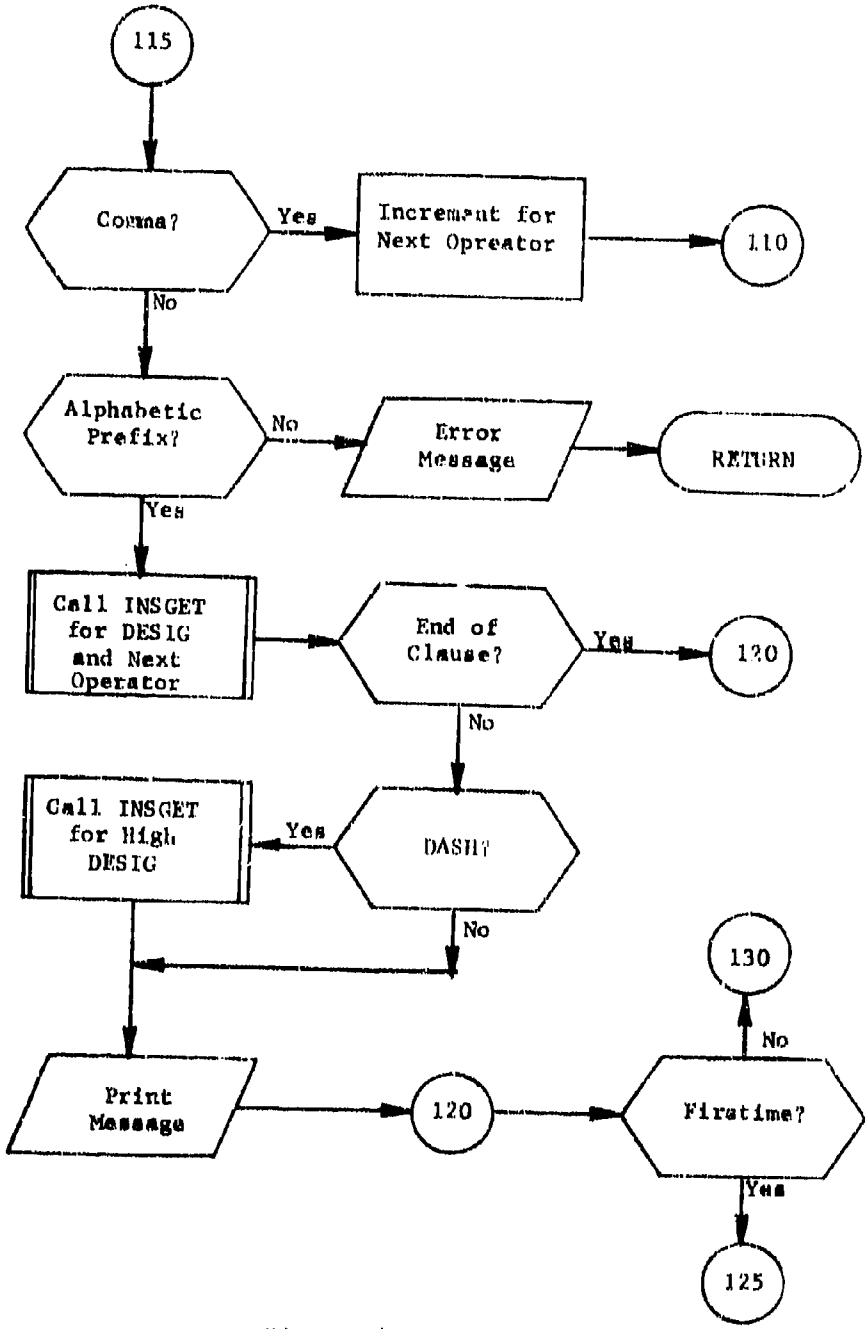


Figure 14. (Part 2 of 3)

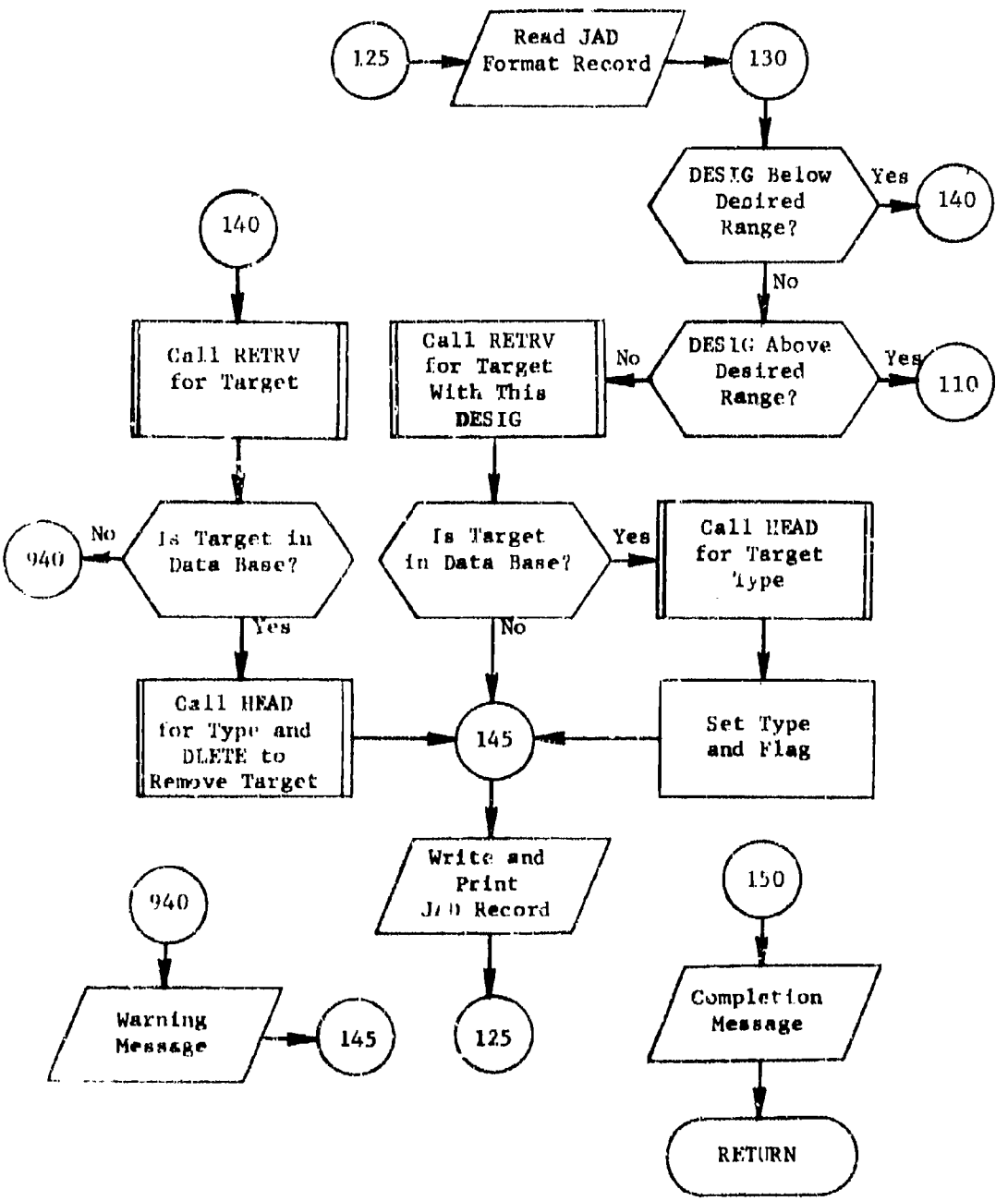


Figure 14. (Part 3 of 3)

SECTION 3. DBMOD MODULE

3.1 Purpose

The purpose of DBMOD is to alter the content or characteristics of a data base in order to adapt the data base to the specific scenario for which the plan is being developed. Because of its highly specialized nature, module DBMOD should be examined for possible revision each time a new plan is to be generated.

3.2 Input

User commands plus the integrated data base are necessary inputs to DBMOD. User inputs define the scenario, attacking and defending sides, plus optional inputs whereby nondefault scaling factors may be set.

All targets to be processed by the QUICK system must have been defined prior to DBMOD execution. This also includes a definition for each target's value (attributes VAL, IGIW or POP). For the attacking side attributes ADBLI, NADBLI, or ADBLR and NADBLR, and NPRSQ1, NPRSQ2, or NPRSQ3 must also have been defined.

3.3 Output

DBMOD generates printed reports and modifies the integrated data base for all U/I class targets for the defending side and for all missile and bomber class targets for the attacking side. U/I targets modify attribute VAL. Missile and bomber class targets modify attributes NOINCO, NALERT, NOPERSQ, ALRTDB and NLRTDB. If user requested, attributes TARDEFHI and TARDEFLO are modified.

3.4 Concept of Operation

DBMOD begins by reading input user commands and stores values that define the scenario to be constructed, the attacking and defending sides, and the nondefault scaling factors used for U/I class value calculations. DBMOD, then determines the attributes for NOINCO (number in commission) and NALERT (number on alert) for bombers and missiles. The user also has the option of scaling the value (VAL) given to an U/I target based on the values for population (POP) and IGIW. The option also exists to calculate local bomber defenses (attributes TARDEFHI and TARDEFLO). For given collections of targets records, parameters are summed and properly printed.

3.5 Identification of Subroutine Function

3.5.1 Subroutine DESTAB. With the exception of utility routines, DESTAB is the only subroutine included under DBMOD. DESTAB keeps track of the number of target records within the data base, the number of target records deleted, and produces summary prints.

3.6 Common Block Definition

Common blocks used by DBMOD are outlined in table 2. Common blocks that communicate with the COP are given in appendix A of Maintenance Manual, Volume I.

Table 2. Module DBMOD Internal Common Blocks

BLOCK	VARIABLE OR ARRAY	DESCRIPTION
CLASSES	CLASSES(60)	List of all legal class values from FINDCLAS
LOCAL	LOCAL(1)	Attacking side
	LOCAL(2)	Defending side
	SCENARIO	SIERRA, INDIA, or ROMEO input
	AUTARD	Flag to automaticaly generate TARDEFHI and TARDEFLO
	PCTTIW, PFIW, PCTPOP, PFPOP	Constants used in U/I scaling equations. Default values are: PCTIW=3.06, PFIW=.81, PFPOP=PCTPOP=0
SIDES	SIDES(5)	Common with FINDSIDE and stores all sides with headers in the data base.
TARDEF	NTARHI	Level of local bomber defense at high altitude
	NTARLO	Level of local bomber defense at low altitude

3.7 Subroutine ENTMOD

PURPOSE:	Alter data base to specified scenario
ENTRY POINTS:	ENTMOD (first subroutine called when overlay link DBMOD is executed)
FORMAL PARAMETERS:	None
COMMON BLOCKS:	CLASSES, C10, C15, OOPS, LOCAL, SIDES, TARDEF
SUBROUTINES CALLED:	DIRECT, DROPDES, FINDCLAS, FINDSIDE, HDFND, HEADRF, UNSGET, KEEPDES, MODFY, NEXTTT, PRNTDES, RETRV, SETDEF

Method:

DBMOD initially reads user's inputs and stores needed parameters. The verb for this module is MODIFY and the only recognized adverb is SETTING. Within the clause parameters SCENARIO, ASIDE and DSIDE must be defined. SCENARIO defines how the data base is to shaped and recognizes values of SIERRA, INDIA, or ROMEO. Parameter ASIDE function is to set the attacking side and DSIDE to set the defending side. Sides are set within record 'NUMTBL' (see figure 15).

If desired the user may request calculations for local bomber defenses through an input within the clause of setting TARDEF=YES. The remaining allowable inputs to the clause are the setting of variables PCTIW, PFIW, PCTPOP, or PFPOP to non-default values for use in U/I calculations.

After input definition, the individual tasks performed are:

a. The appropriate number of bombers or tankers for each squadron (NOPERSQN) is selected depending upon the particular plan being developed (Initiative, Surprise, or Retaliatory)

b. The number of bombers or tankers in commission (NOINCOM) for each squadron is calculated by specifying that attribute NOINCOM is equal to NOPERSQN

c. The number of bombers or tankers which are on alert (NALERT) for each squadron is also set to NOPERSQN

d. The relative value (attribute VAL) of urban/industrial targets is calculated as a function of general industrial worth (IGIW) and population (POP)

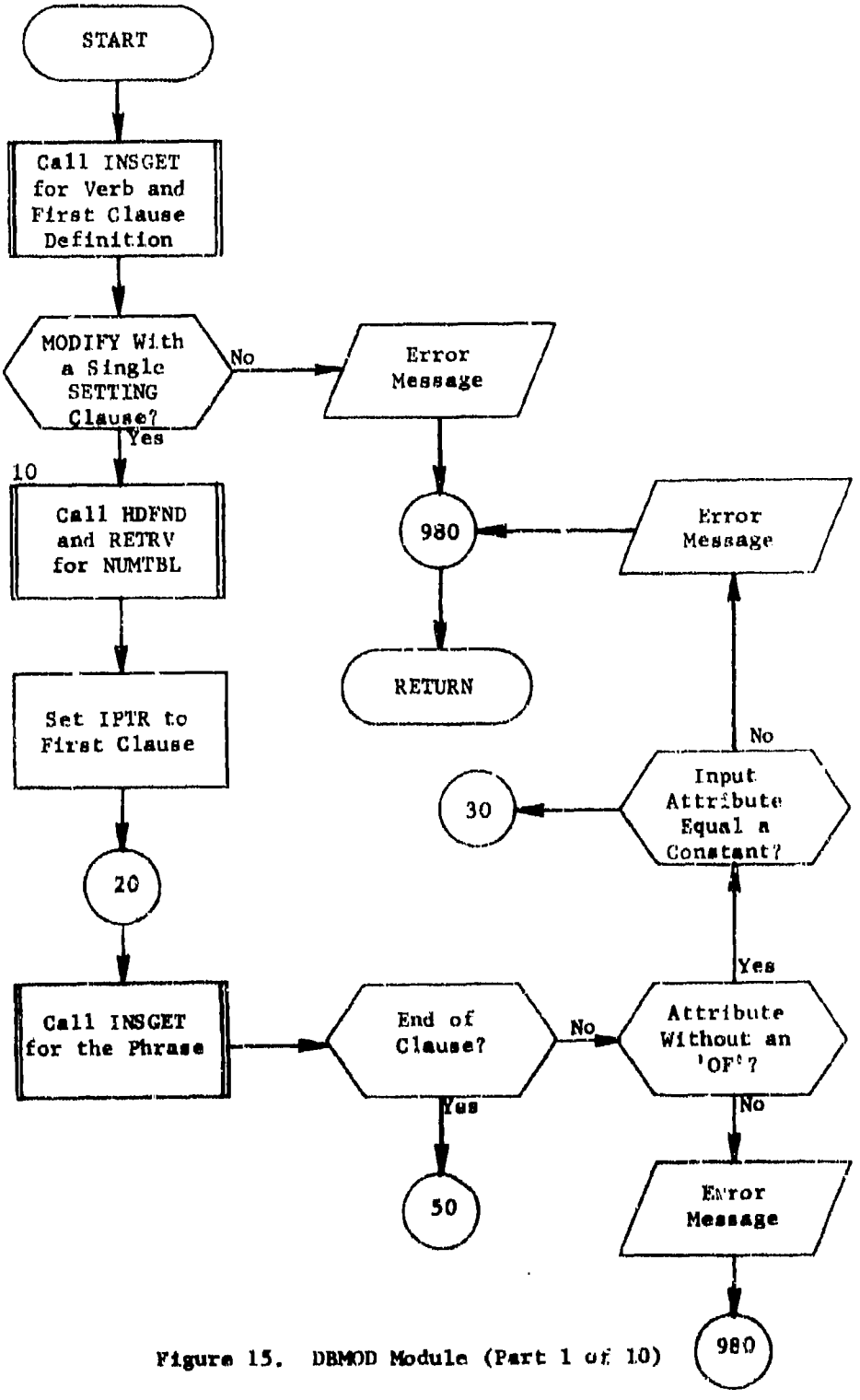


Figure 15. DBMOD Module (Part 1 of 10)

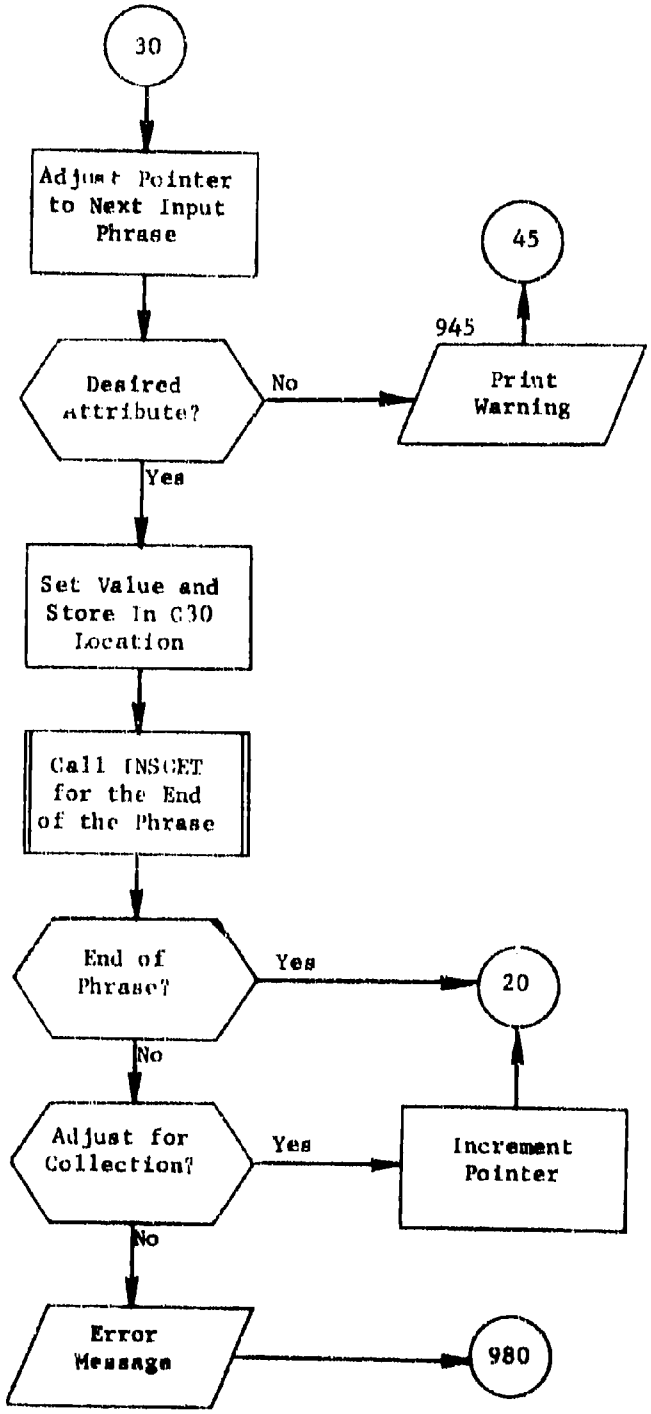


Figure 15. (Part 2 of 10)

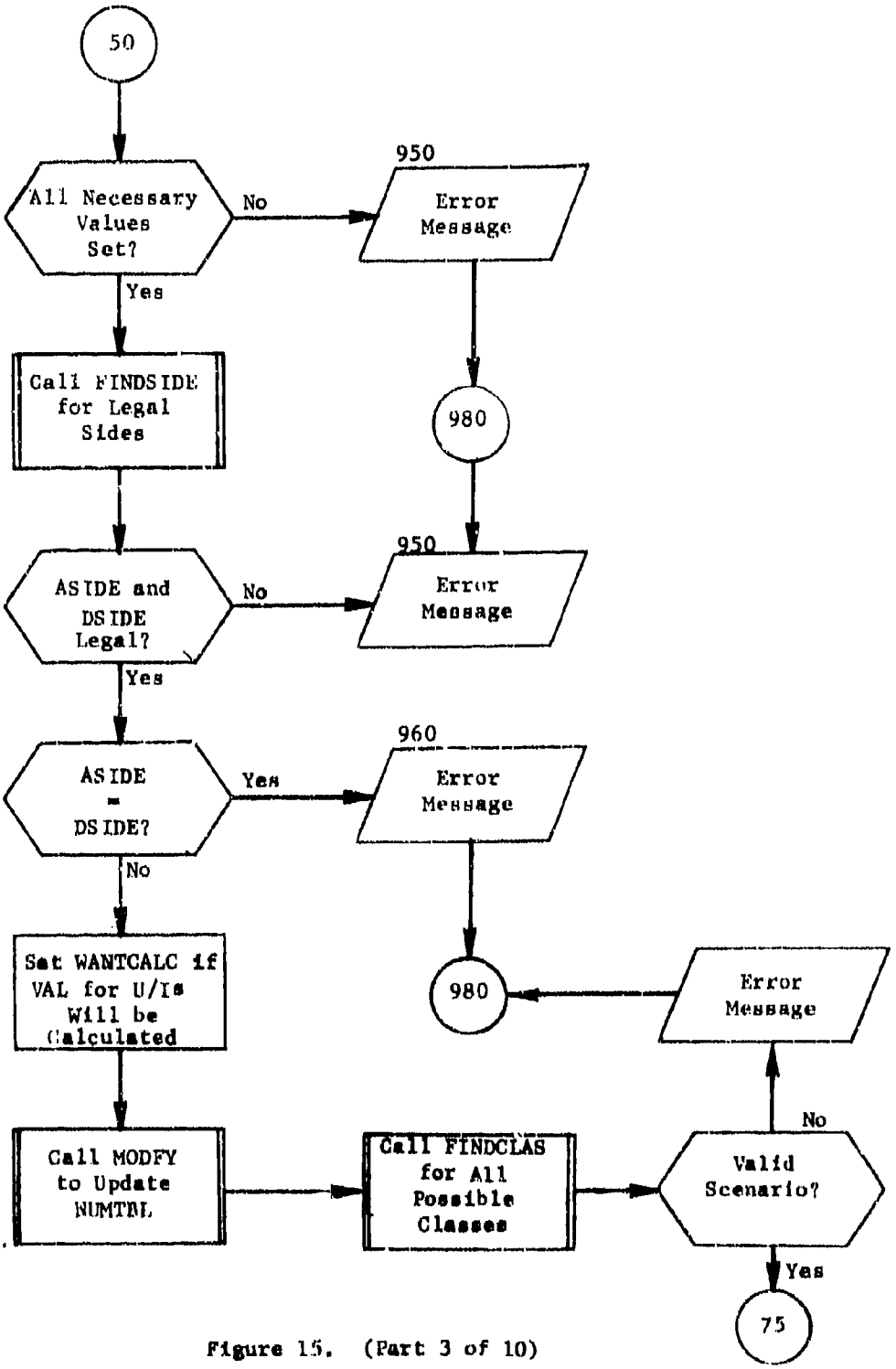


Figure 15. (Part 3 of 10)

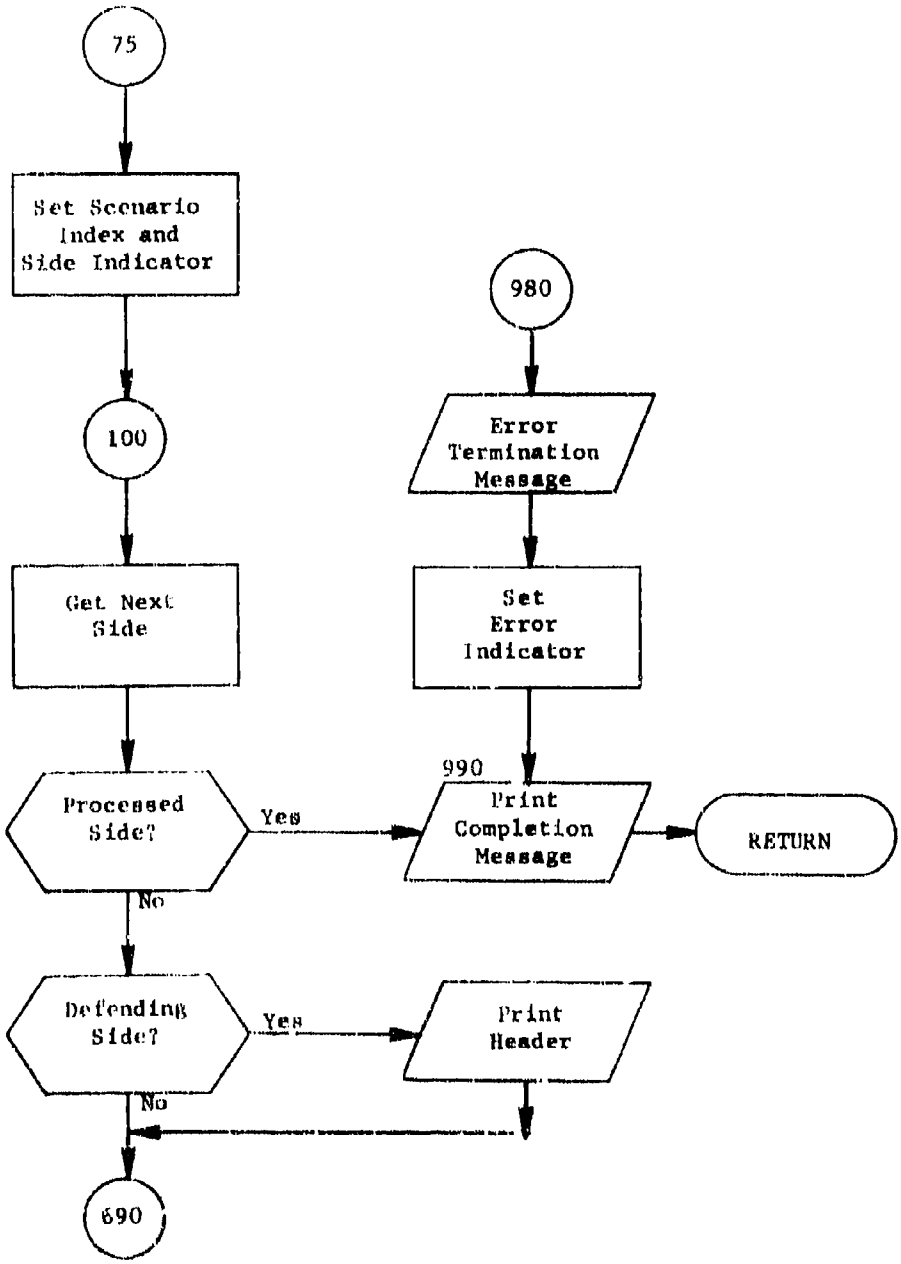


Figure 15. (Part 4 of 10)

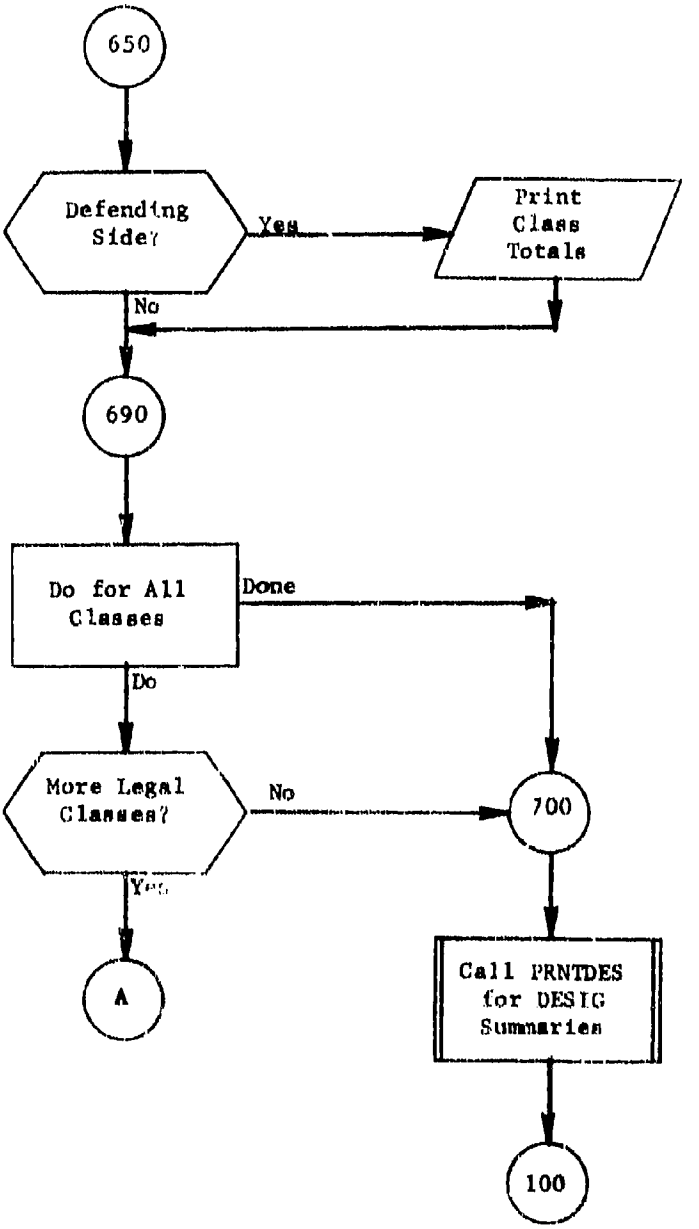


Figure 15. (Part 5 of 10)

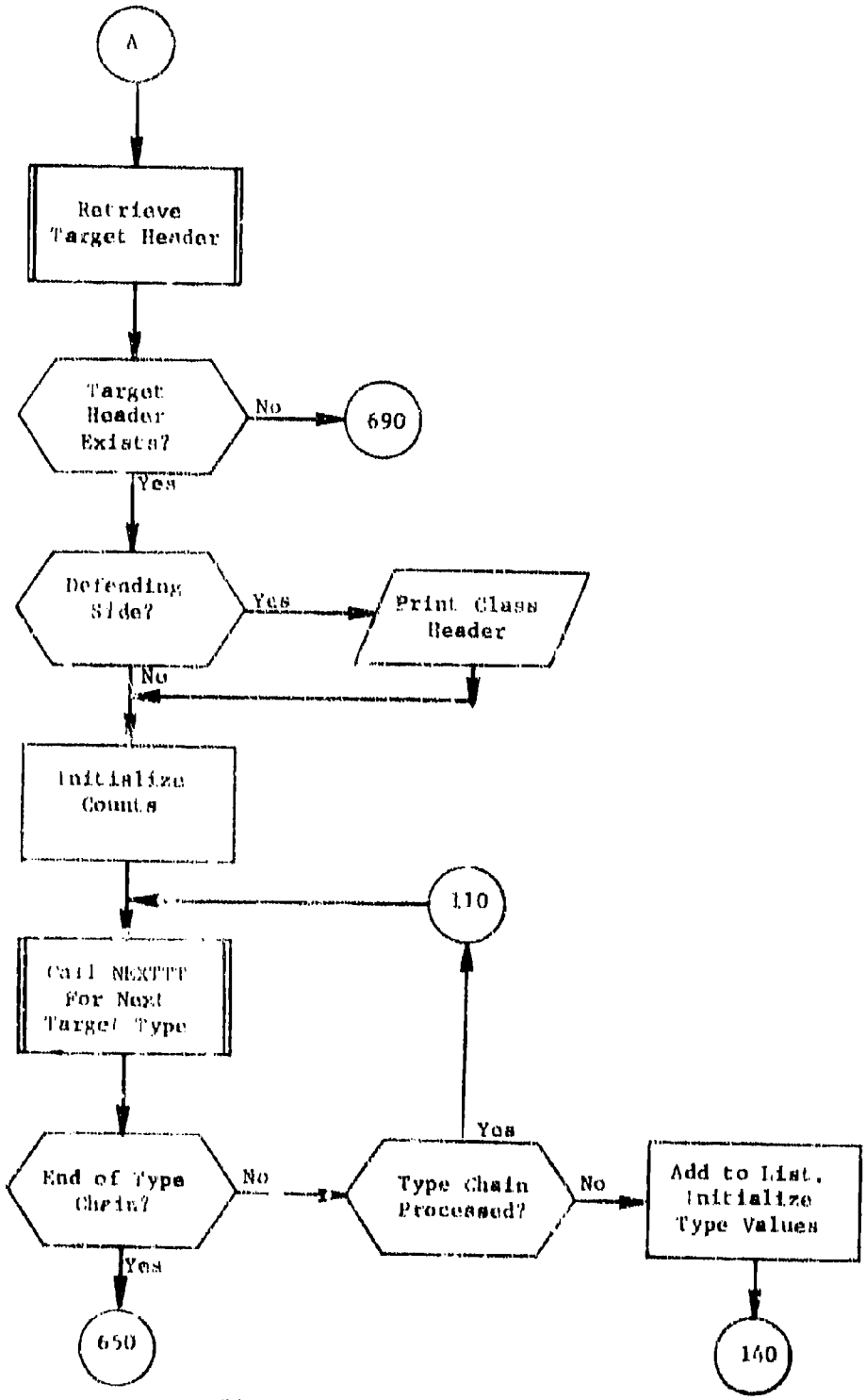


Figure 15. (Part 6 of 10)

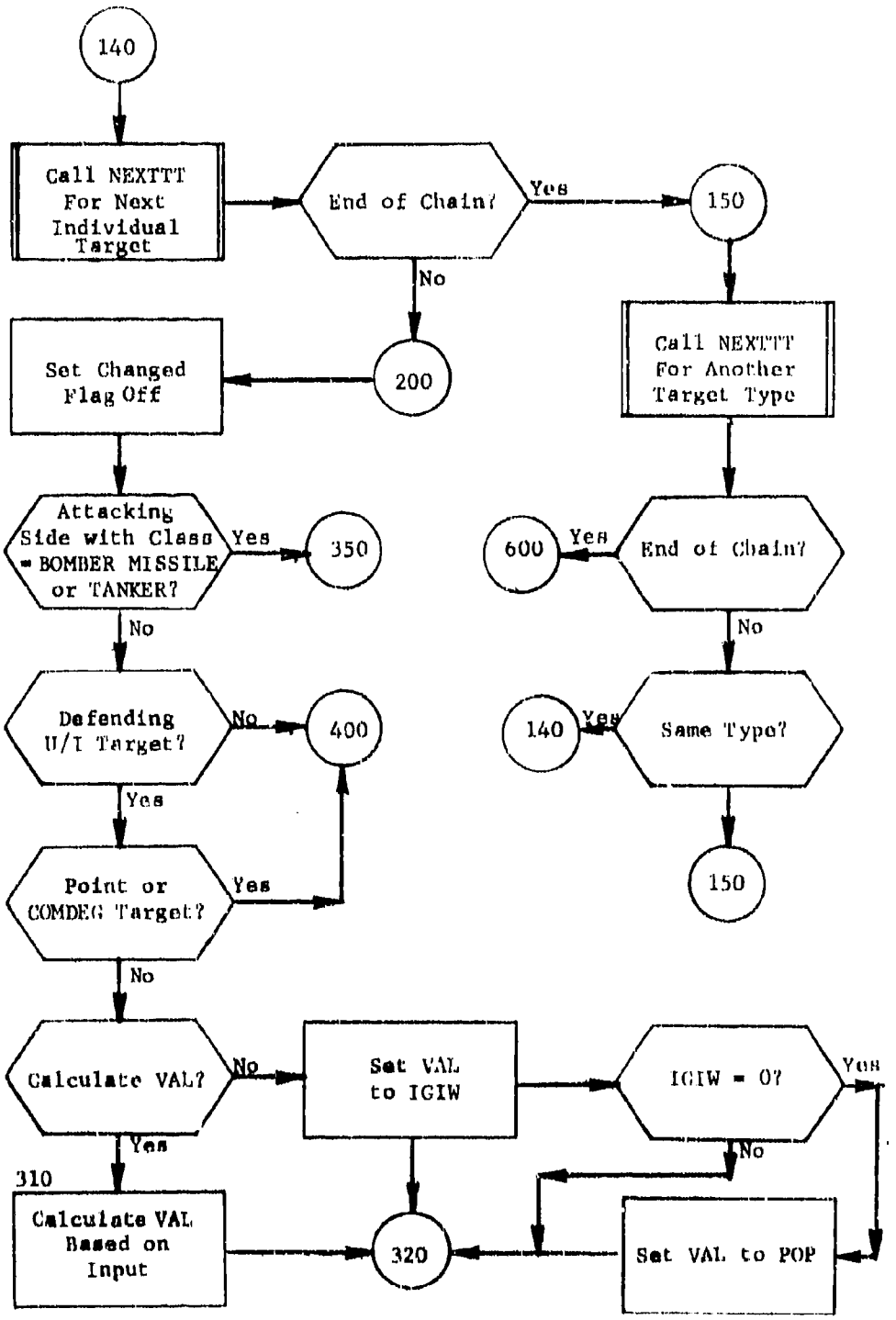


Figure 15. (Part 7 of 10)

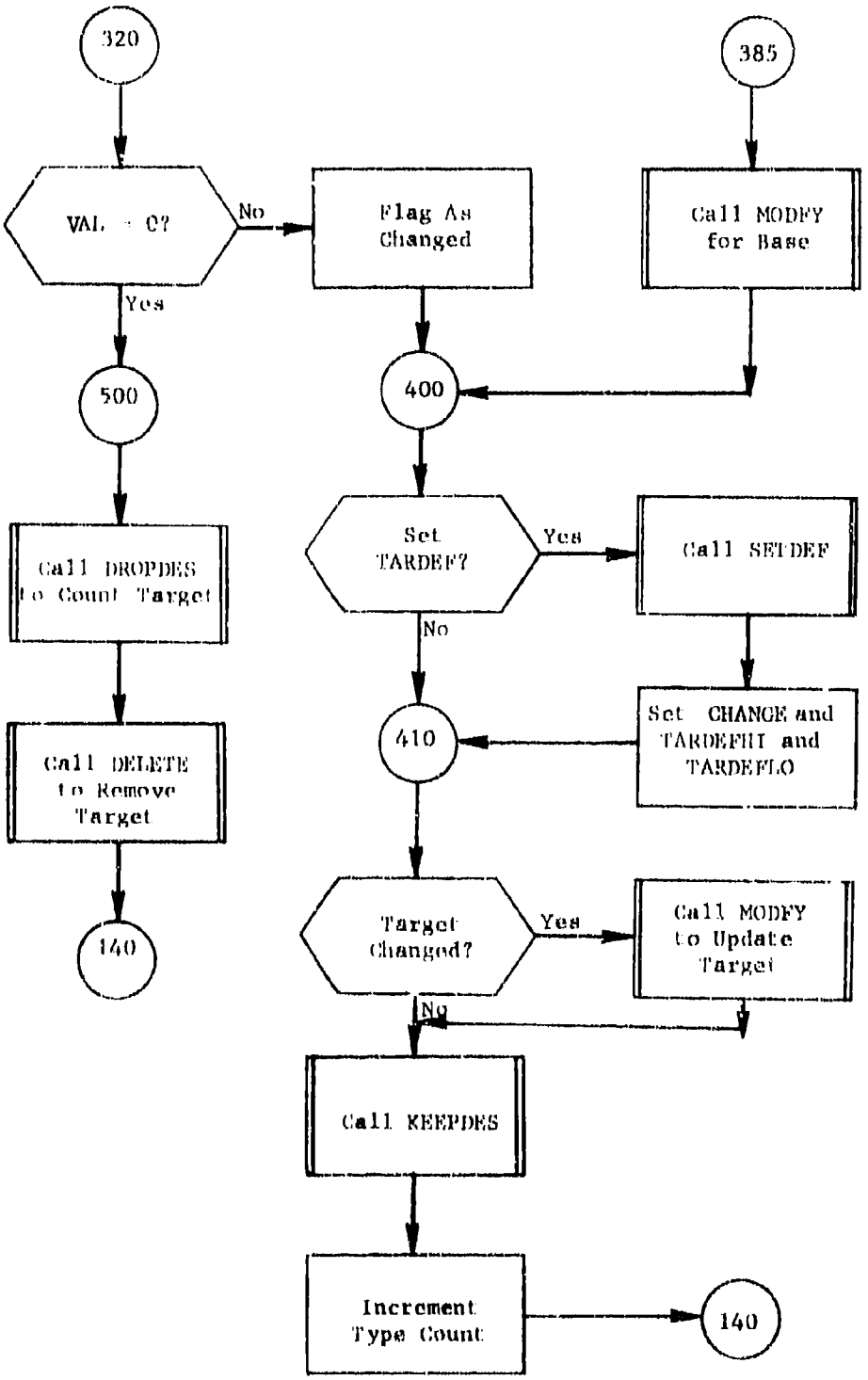


Figure 15. (Part 8 of 10)

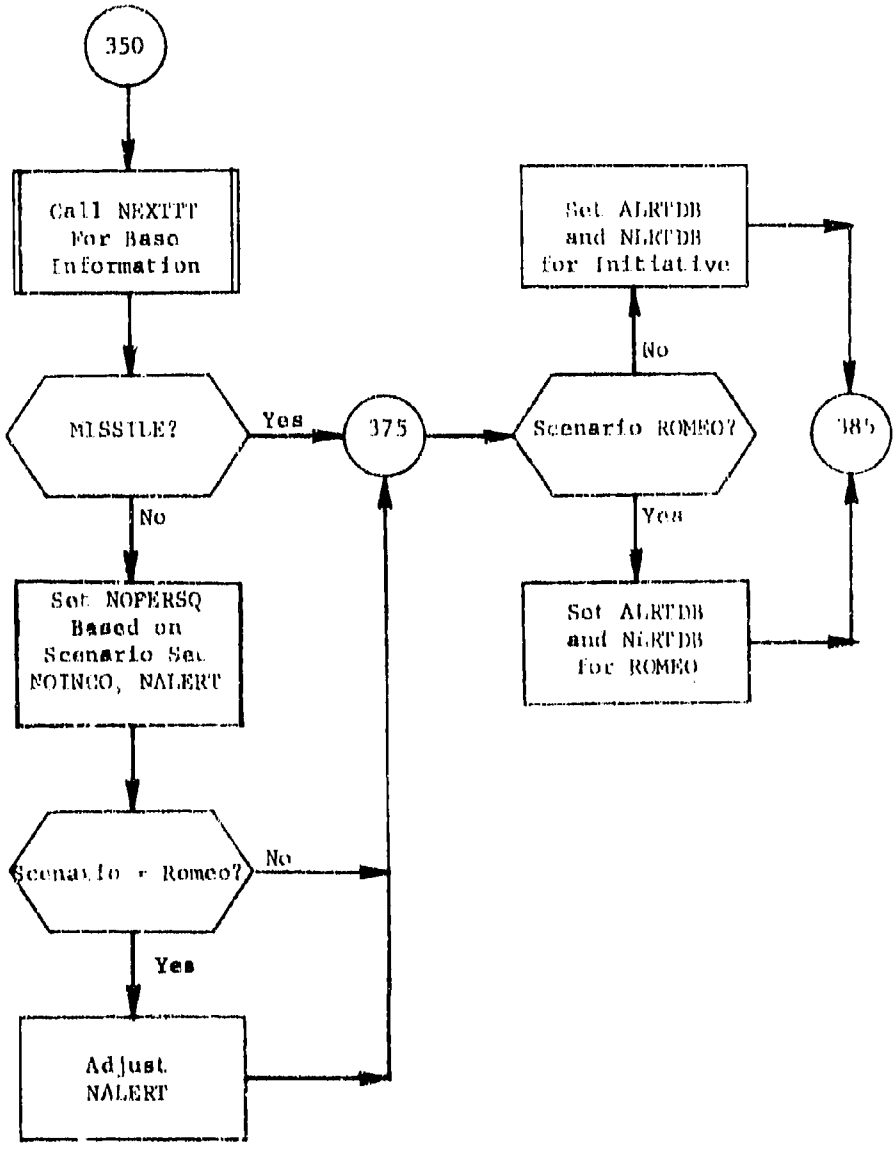


Figure 15. (Part 9 of 10)

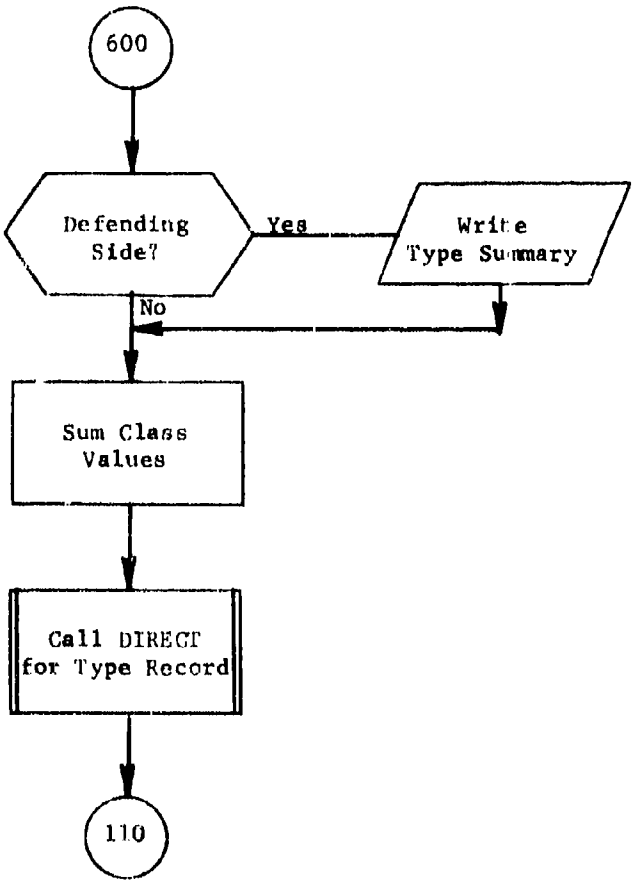


Figure 15. (Part 10 of 10)

e. If the TARDEF option is exercised, each target (on the defending side) is processed and the level of local bomber defense available at the target is calculated.

f. If any target value equal zero (VAL=0), that record will be deleted from the data base.

If addition to the above tasks, the following prints are produced:

a. The target value summary that reflects the count and cumulative value of the targets by SIDE, CLASS, and TYPE.

b. The target count by SIDE, REGION, and the alphabetic portion of the DESIG

c. The target count deleted by SIDE, REGION, and the alphabetic portion of the DESIG

The order of target chaining is conducted in such a way that the target value print summary may be produced within having to save various combinations of SIDE, CLASS, TYPE, and VAL values. Namely, for a given side a target class is chosen, a TYPE is chained for that class and all individual targets processed for the CLASS and TYPE intersection. All TYPEs are chained for a given CLASS entry and then a second CLASS entry is picked and processing continues.

Note that attribute TYPE and the record type chain called 'TGTTYP' are not necessarily in one-to-one correspondence. That is, for a value of TYPE there may exist more than 'TGTTYP' record. Code is implemented to check for this occurrence.

3.8 Subroutine DESTAB

PURPOSE: Builds and prints counts of DESIGs by region

ENTRY POINTS: DROPDES, KEEPDES, and PRNTDES

FORMAL PARAMETERS:

DESIG, IREG	Target DESIG and region for entries DROPDES and KEEPDES
SIDE, IREG	Side to be printed and region for entry PRNTDES

COMMON BLOCKS: None

SUBROUTINES CALLED: None

CALLED BY: ENTMOD (of overlay link DBMOD)

Method:

Entries DROPDES and KEEPDES

Based on formal parameters DESIG and IREG maintain a count of records to be deleted or records to be retained within the data base.

Entry PRNTDES

Prints the counts as built by DBMOD.

Subroutine DESTAB is illustrated in figure 16.

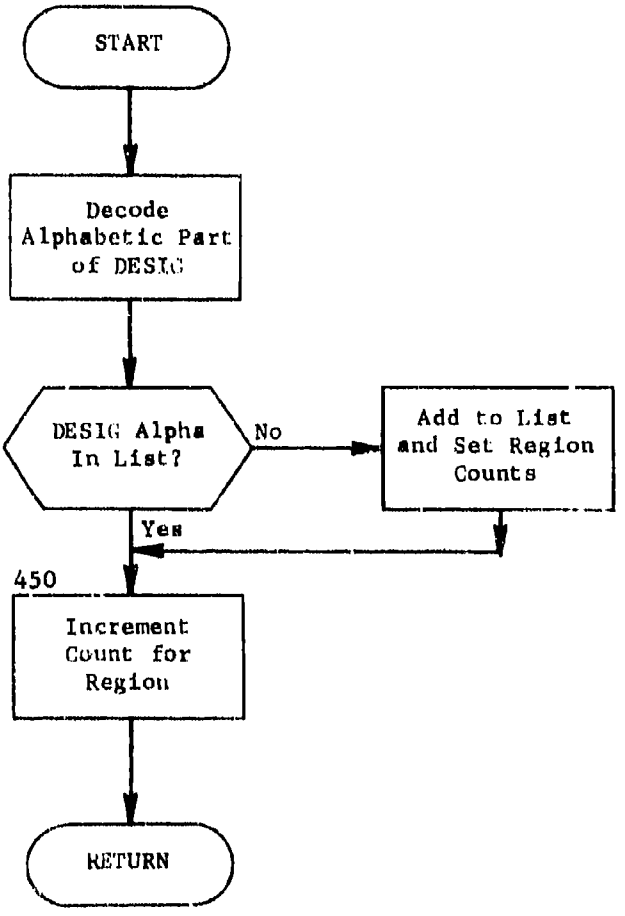


Figure 16. Subroutine DESTAB: Entry KEEPDES (Part 1 of 3)

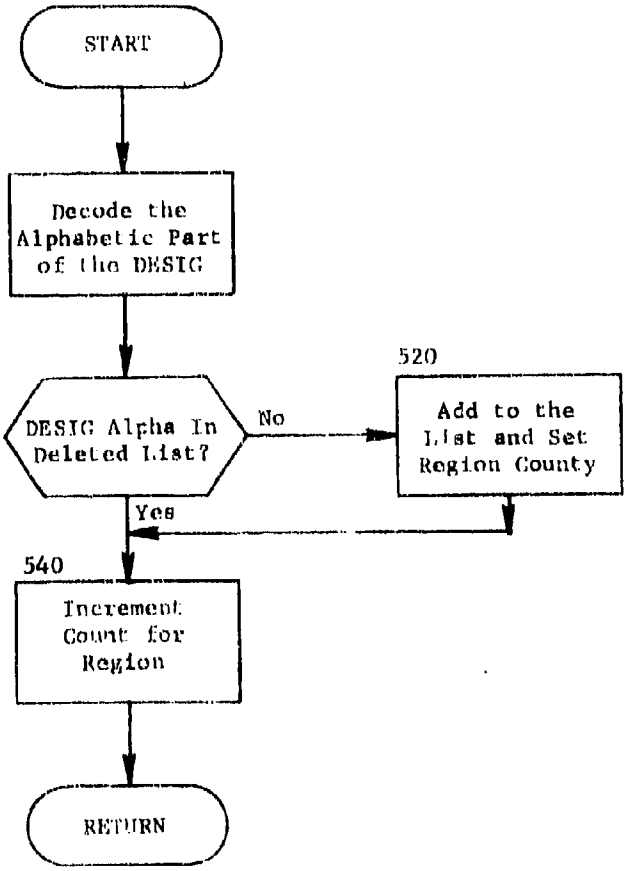


Figure 16. Entry DROPDES
(Part 2 of 3)

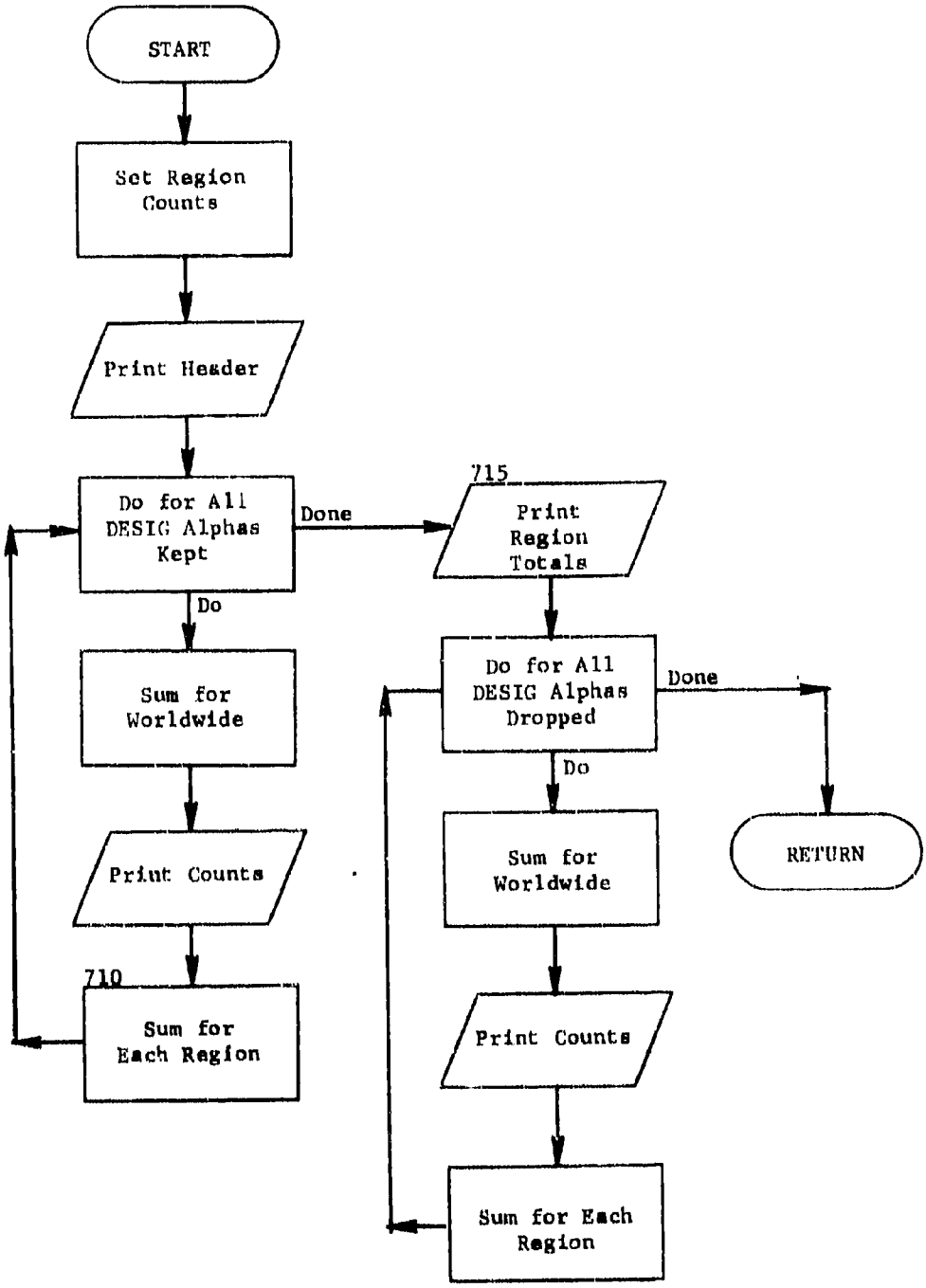


Figure 16. Entry PRNTDES
(Part 3 of 3)

THIS PAGE INTENTIONALLY LEFT BLANK

SECTION 4. INDEXER MODULE

4.1 Purpose

To provide for economical handling of data and to facilitate communications between QUICK modules, it is necessary to assign indices to various data contained in the data base. Module INDEXER is designed to perform this task. In addition, INDEXER processes all potential targets and, where appropriate, forms them into complex targets.

4.2 Input

The input to module INDEXER consists of the user-input command and the integrated data base. The user-input command identifies a complexing option or a weapon yield used for complexing and print option requests.

INDEXER accesses the integrated data base in three ways. First, it accesses the vulnerability table header VNTKHD and the vulnerability number records (VULNUM) on the VNTKS chain. Second, it accesses the target data by retrieving the target headers (TGTHD), the target type records (TARGTY) on the TGTTYP chain, the TARGET records on the TGTTGT chain, and finally the MSBMTG records on the TARGXX chain. Third, it accesses the targets within sector using the sector header (SECHD), the sector records (SECTR) on the SECTOR chain, and the targets on the TGTSEC chain.

4.3 Output

INDEXER modifies the integrated data base in two ways. First, it alters the contents of the TARGET record, adding an index number and, for missile and bomber targets, calculates the time decay values. Also, as a result of the complex formation process, it changes the linkage of targets on the CMPTGT chain. When originally stored, all TARGET records are placed on the CMPTGT chain as details of a single master: This master record (type COMPTG) may be thought of as the master of a chain of all simple targets. Following the complex formation process, each complex causes the creation of a COMPTG record. This record is used as the master of a COPTGT chain whose details are those TARGET records which make up the complex. After the creation of this record, all TARGET records which are a part of this complex are modified so that they are now chained to the new record rather than the simple target master. Thus when all processing is completed, all targets will be differentiated as to whether they are simple, or complex targets by the COMPTG record to which they are chained.

4.4 Concept of Operation

After a scenario has been selected, module INDEXER performs necessary calculations and additions to the refined data base. The major objectives of INDEXER are to: (a) assign unique indices to all targetable records (referred to as index number, attribute INDEXNO); (b) automatically calculate time decaying value points for all target bomber and missile bases; (c) calculate for each unique target vulnerability a complexing lethal radius based on user selected yields; (d) complex individual targets based on selected algorithm; and (e) define the target complex classes.

4.5 Identification of Subroutine Functions

4.5.1 Subroutine COMPLEX. This subroutine queries all potential targets on an earth sector (boundaries of longitude) basis and forms complexes. Simply, elements of targets are defined as being in a complex if they are geographically within a defined destruct radius of each other. The destruct radius is the lethal radius calculated and stored in earlier processing.

4.5.2 Subroutine CRTBLE. This subroutine calculates complexing lethal radius based on hard coded tables and is executed only when user directed.

4.5.3 Subroutine SETVAL. If a target belongs to a missile or bomber class and is salvoed, this routine will calculate time value decay curves based on the rate at which sorties leave a launch base.

4.6 Common Block Definition

Common blocks used by INDEXER are outlined in table 3. Common blocks that communicate with the COP are given in appendix A of Program Maintenance Manual, Volume I.

Table 3. Module INDEXER Internal Common Blocks

BLOCK	VARIABLE OR ARRAY	DESCRIPTION
CYIELD	SYIELD(2)	Yield, in megatons, used in forming complexes. Value is user determined.
DIFFLAT	DIFFLAT	Maximum difference in latitude (DEG) in forming complexes. Computed in INDEXER based on the softest target in the data base.
IOPRT	IOPRT	If zero, nonstandard prints are suppressed. User determined.

4.7 Subroutine ENTMOD

PURPOSE:	Read user inputs, calculate and store complexing lethal radius, determine attribute INDEXNO and control flow of supporting subroutines
ENTRY POINTS:	ENTMOD (first subroutine called when overlay link INDXER is executed)
FORMAL PARAMETERS:	None
COMMON BLOCKS:	C10, C15, C25, C20, C30, CYIELD, DIFFLAT, IOPRT
SUBROUTINES CALLED:	COMPLEX, CRTBLE, DIRECT, HDFND, HEAD, INSGET, ITLE, MODFY, NEXTTT, RETRV, SETVAL, VLRADI
CALLED BY:	COP

Method:

Module INDEXER begins (figure 17) by reading (through utility subroutine INSGET) and storing user input parameters, then for each unique vulnerability contained within the data base, a complexing lethal radius is calculated and stored. Following this, individual targets are chained in a specified manner and modified to include attribute INDEXNO. As individual targets are chained, subroutine SETVAL is called for all missile and bomber classes for possible time value decay calculations. After querying targets, subroutine COMPLEX is called in order to form target complexes and upon completion processing is terminated.

User Input Definition

INDEXER initially retrieves record type 'NUMTBL' in order to define the attacking (ASIDE) and defending (DSIDE) side. These attributes were stored in the data base by module DBMOD. Following side definition, the user inputs are retrieved and needed values stored.

If the verb is correct (comparison to local parameter IND) processing continues; otherwise an error message is printed and processing stops.

Existence of adverbs WITH, VNOPTION, or ONPRINTS are checked for. Use of adverb ONPRINTS, implies nonstandard prints are to be produced; adverb VNOPTION implies that complexing lethal radius is to be obtained from hard coded tables. In absence of the VNOPTION clause, complexing is performed with an assumed weapon yield of one megaton. The user may override this yield through a clause introduced by the adverb WITH. Both attributes YIELD and SIDE are included within the WITH clause.

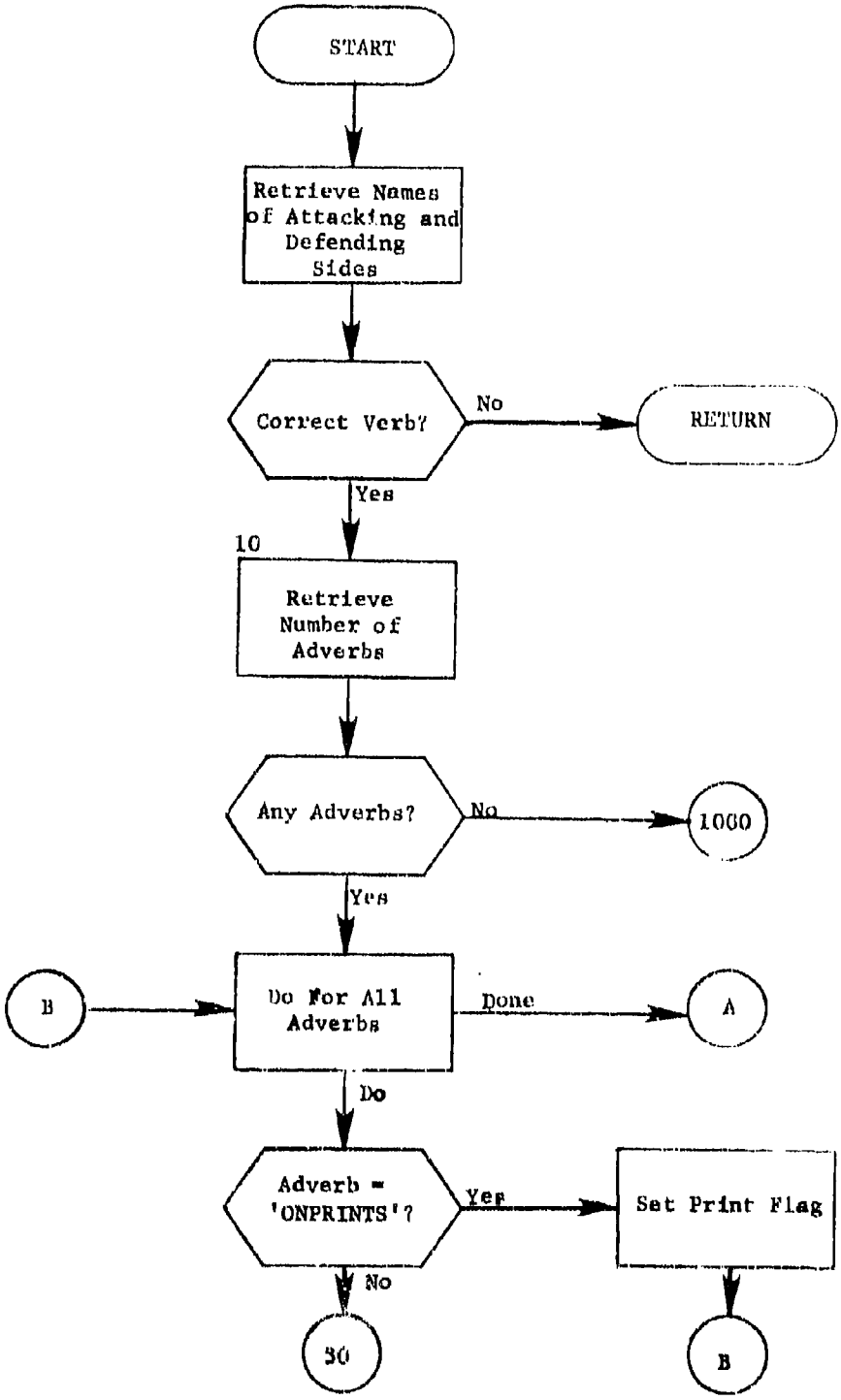


Figure 17. INDEXER Module (Part 1 of 7)

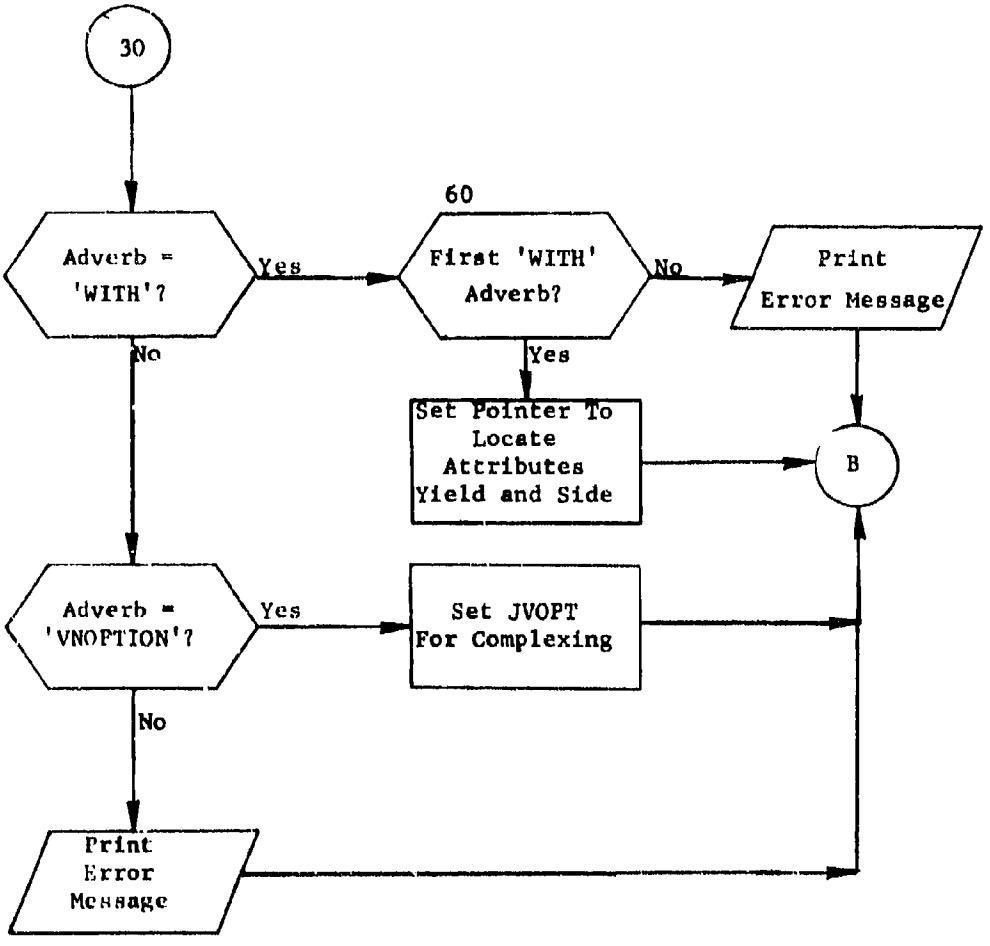


Figure 17. (Part 2 of 7)

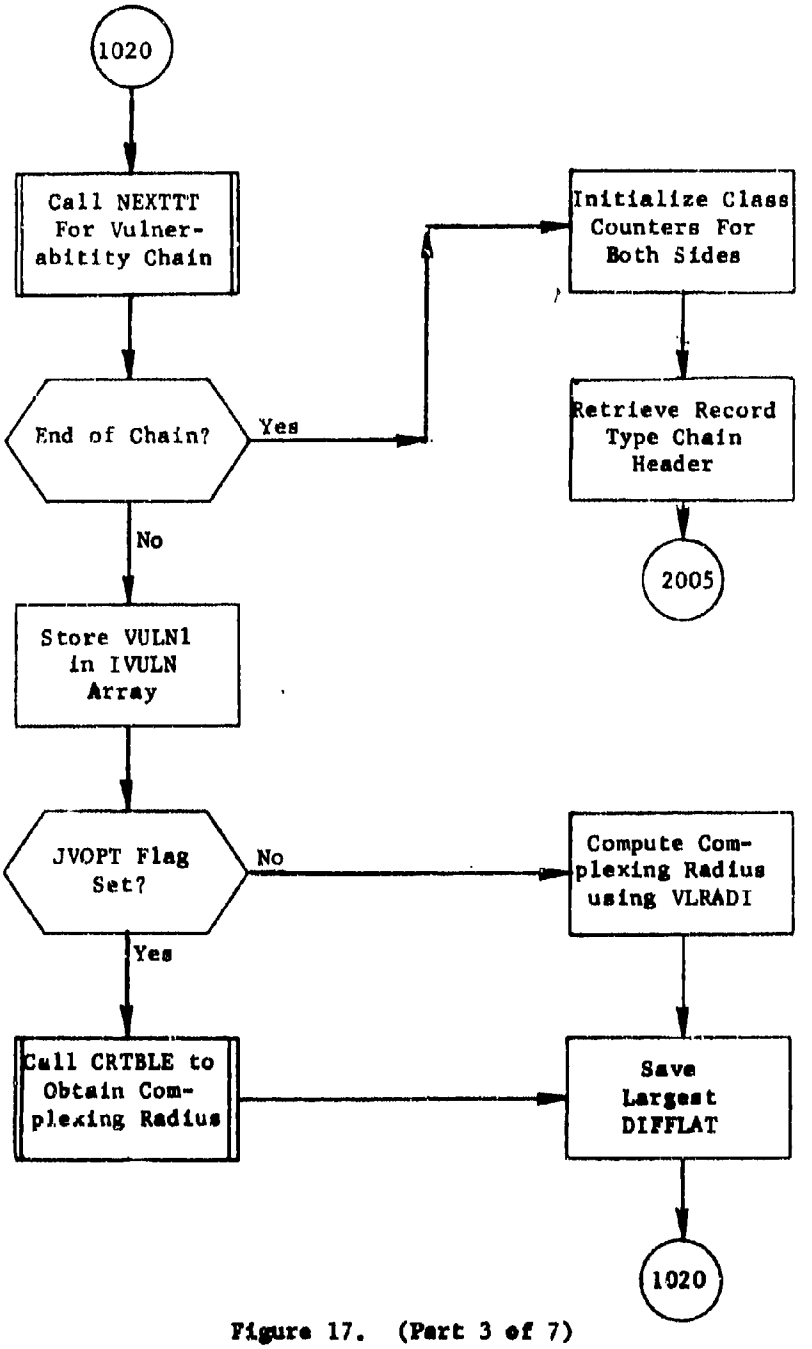


Figure 17. (Part 3 of 7)

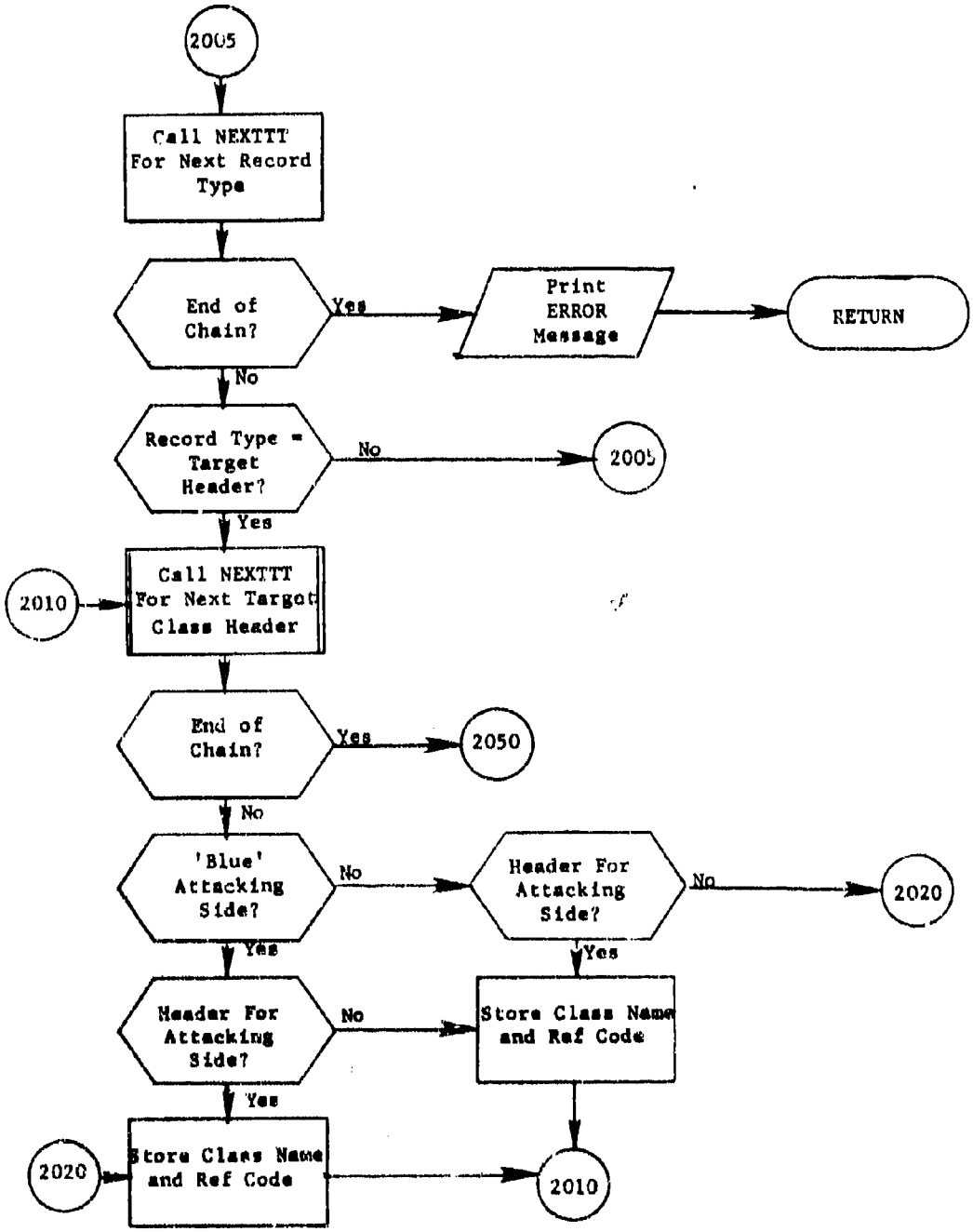
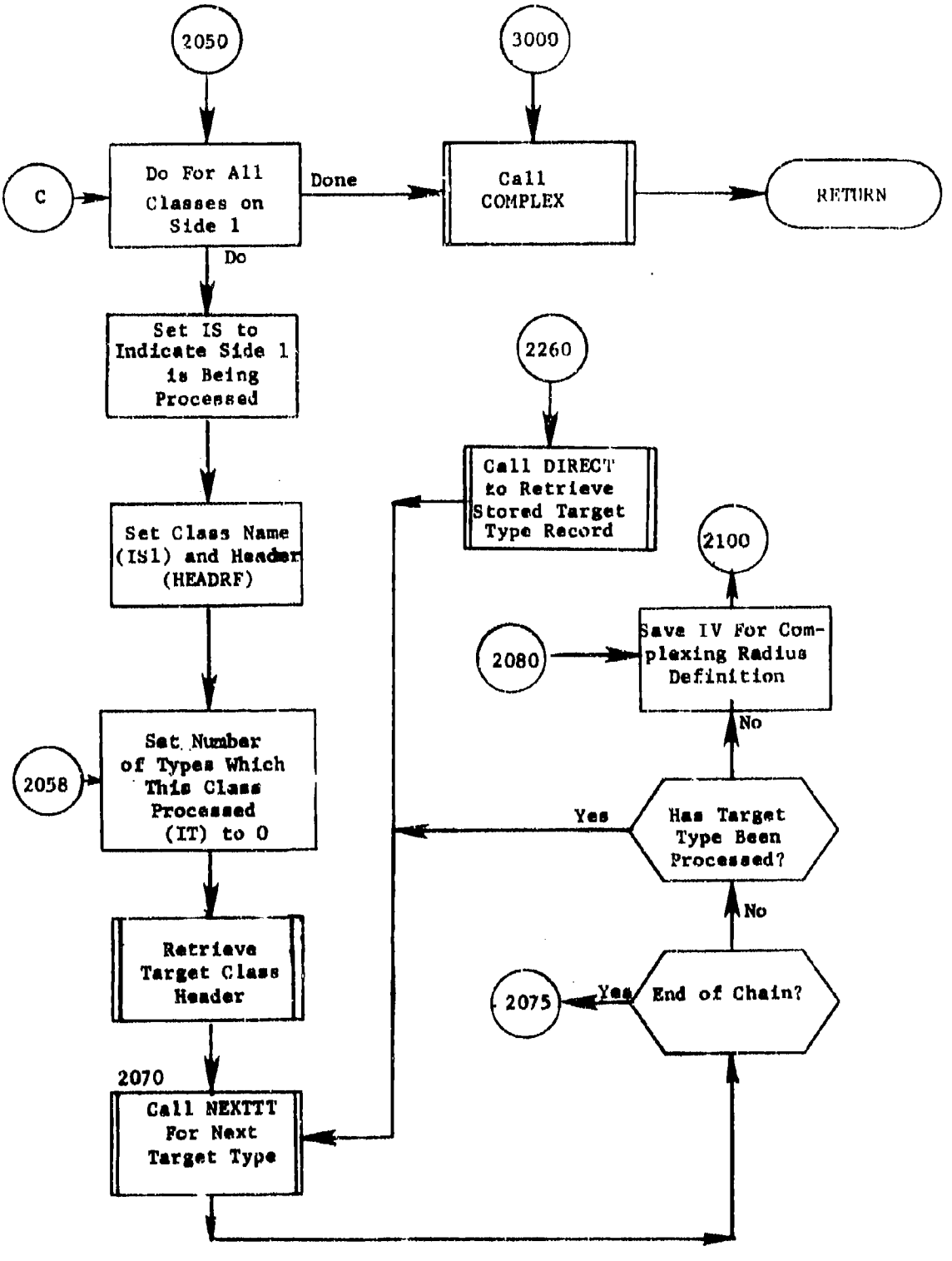


Figure 17. (Part 4 of 7)


2050
3000
C
Do For All Classes on Side 1
Done
Call COMPLEX
RETURN
Do
Set IS to Indicate Side 1 is Being Processed
2260
Call DIRECT to Retrieve Stored Target Type Record
Set Class Name (IS1) and Header (HEADRF)
2100
2080
Save IV For Complexing Radius Definition
2058
Set Number of Types Which This Class Processed (IT) to 0
No
Yes
Has Target Type Been Processed?
Retrieve Target Class Header
No
2075
Yes
End of Chain?
2070
Call NEXTTT For Next Target Type

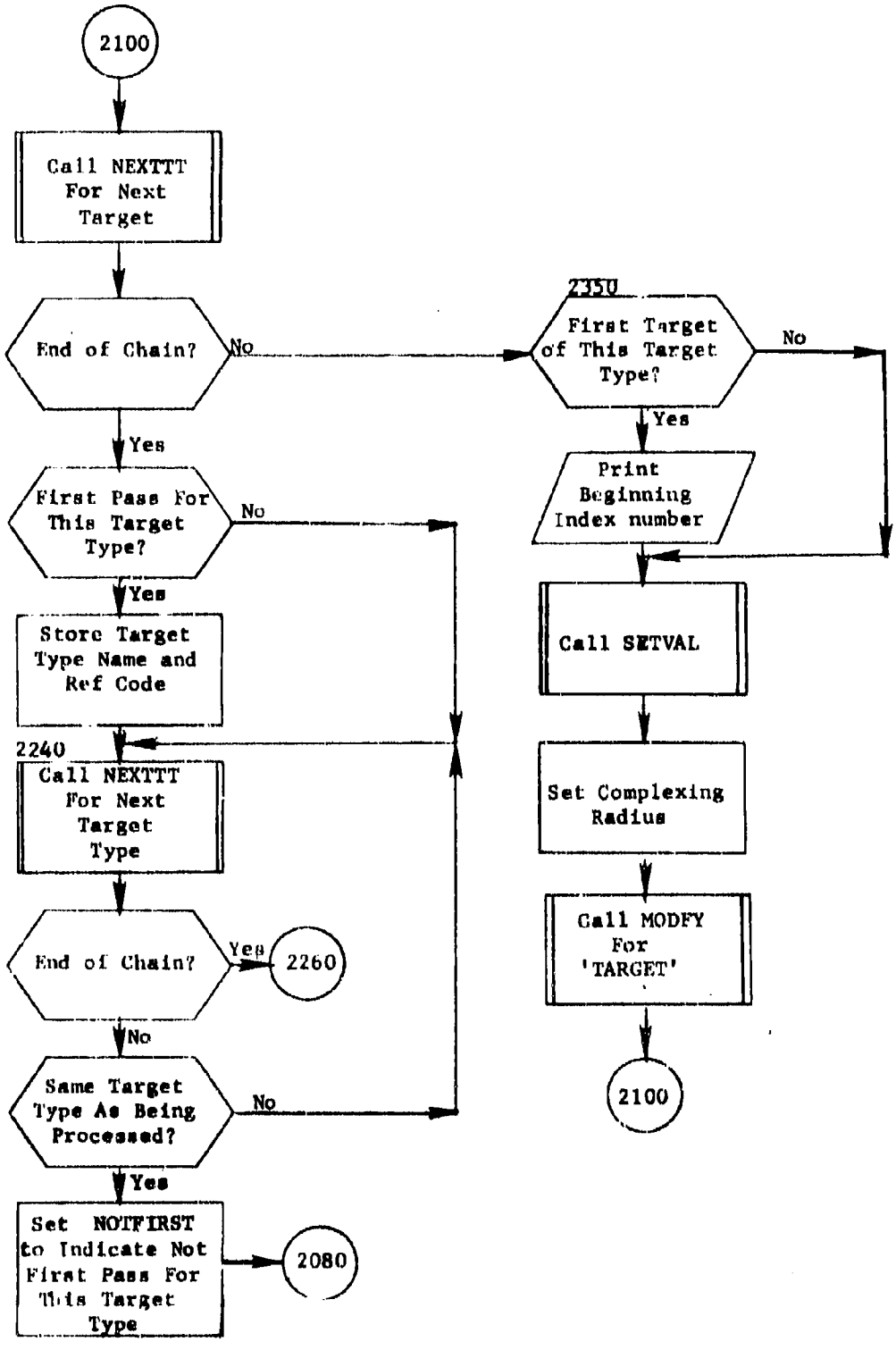


Figure 17. (Part 6 of 7)

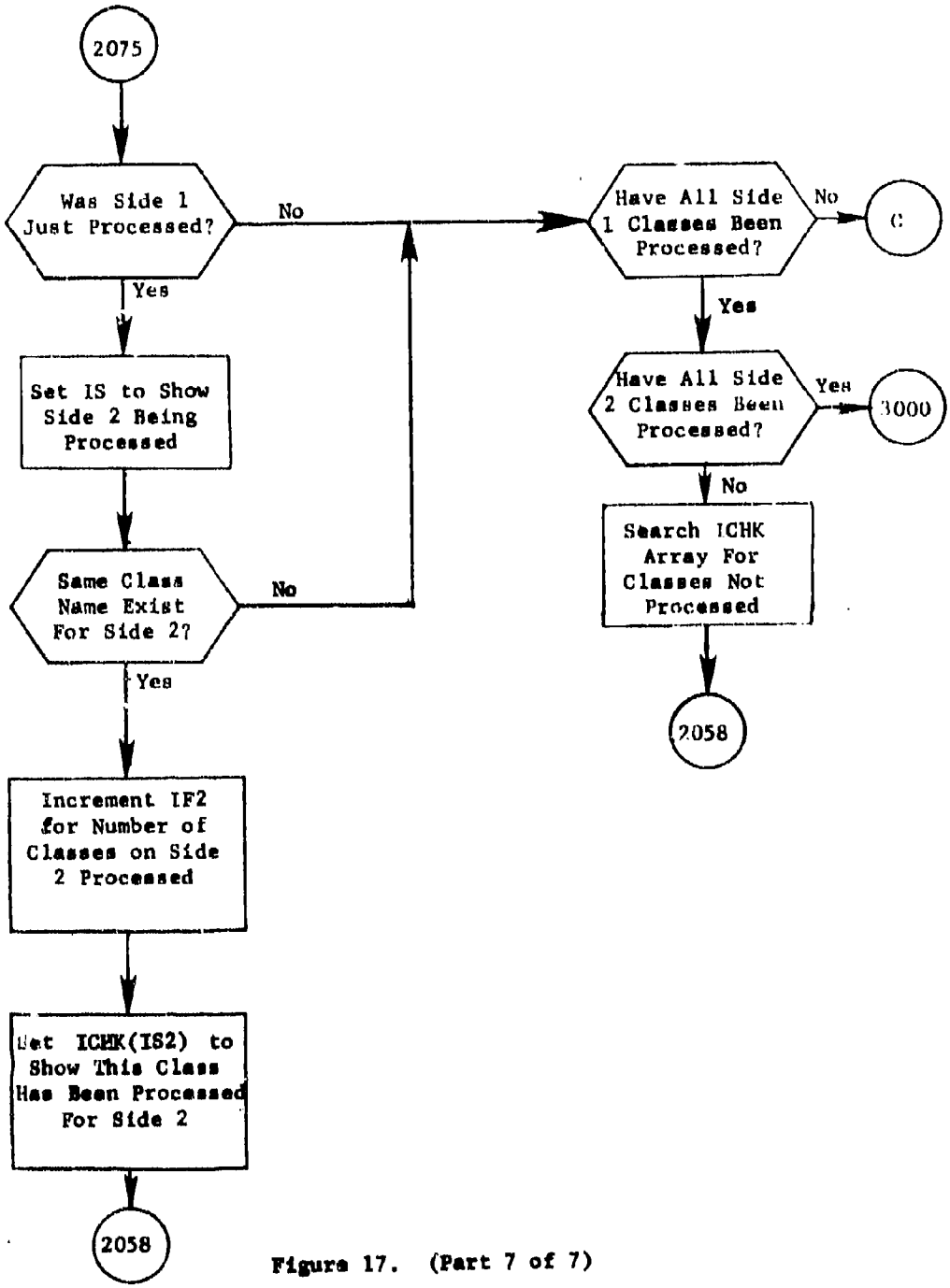


Figure 17. (Part 7 of 7)

Complexing Lethal Radius Calculation

For each unique vulnerability defined within the data base for the defending side, a complexing lethal radius is calculated and stored within local array CLR which contains a maximum of 255 entries. Calculations are performed either in subroutines VLRADI or CRTBLE depending upon user requests. After each calculation, results are printed and DIFFLAT is rechecked for definition of the softest target among targets to be queried.

Vulnerabilities (attribute VULN) are collected under header 'VNTKHD' and are contained within chain 'VNTKS'.

Attribute INDEXNO and Time Value Assignments

Individual targets are now processed in a defined manner in order to assign index numbers. For target classes called missiles or bombers, time value factors are generated. Also, the stored complexing lethal radius (array CLR) is defined within each individual target record based on attribute VULN1.

Index numbers will be assigned sequentially for all target records that have a similar value for attribute TYPE for a given class and side combination. For a given target class individual target records are collected for one side followed by all records for the remaining side, if TYPEs exist for the second side. Within a collection of records for a given TYPE value, items are queried according to the order in which they appear in the data base.

Target class names that are stored within the data base may be found by querying record chain 'RCTYP' of the organizational data. Each class name encountered under chain 'RCTYP' is locally stored in array KREF. Storage is on a side basis. The existence of a class name for one side does not guarantee an entry for the other side.

INDEXNO assignments begins by picking a target class (header TGTHD) for side 1 (local parameter IS=1) and further picking a target type record. Now for a target class, side, target type combination, all of the individual targets are chained and INDEXNO, complexing lethal radius, and, if necessary, time value factors are stored.

After a given TYPE record is processed, the next TYPEs are processed for the same class and side. Upon exhausting a class and side combination, side 2 (IS=2) is investigated for all TYPE records. Finally a new target class for side 1 is chosen and cited processing is repeated.

Attribute TYPE and the record type chain called 'TGTTYP' are not necessarily in one-to-one correspondence. That is, for a value of TYPE there

may exist more than one 'TGTTYP' record. This is possible since the attributes defined in the 'TGTTYP' records may have different values for a TYPE value. For instance, for TYPE=MMIII there could be two entries for attribute CNTRYL (country location), say US and CA. For this condition two 'TGTTYP' records will exist and both have attribute TYPE=MMIII. Therefore upon chaining 'TGTTYP', the entire list must be checked for multiple occurrences of the same TYPE value.

After the last class entry for side 1 is processed, checks are made to ensure all side 2 entries have been processed. This is necessary since the major processing is for all side 1 entries and the fact exists that class names may be defined for side 2 but not side 1. Local array ICHK is set to nonzero as each side 2 class name is processed.

4.8 Subroutine COMPLEX

PURPOSE: To form complex targets

ENTRY POINTS: COMPLEX

FORMAL PARAMETERS: None

COMMONB BLOCKS: C10, C15, C30, DIFFLAT, IOPRT

SUBROUTINES CALLED: DIRECT, DLETE, HDFND, MODFY, NEXTTT, RETRV, SORTIT, STORE

CALLED BY: ENTMOD (of overlay link INDEXER)

Method:

Individual target records are queried in order to form target complexes. If any two targets are geographically located within one half the sum of the complexing lethal radius of each target, they belong to the same complex. For each new complex formed, a complex number (parameter ICOMPL) is sequentially updated, stored under record 'COMPTG' and each individual target belonging to the complex is stored on the 'CMPTGT' chain.

Complexes are formed within earth segments which are simply divisions of longitude. The earth sector chain is called 'SECTOR'. The initial action of COMPLEX, as shown in figure 18, is to query a 'SECTOR' chain and sort all records within that chain by increasing latitude. Subroutine SORTIT performs the sort and final results placed on file unit ISORTLUN. Each ISORTLUN record contains latitude and the Reference Code of the target record associated with the latitude. File ISORTLUN is read, and an indexed random file (ISAMLUN) is written which will be interrogated within COMPLEX. Each ISAMLUN record contains target latitude, longitude, reference code, complex number, and complexing lethal radius (CC0I).

The search for complex targets begins by comparing differences in latitude for consecutive targets in the sorted file, beginning with the first noncomplex target. When a distance between the first selected target (associated with latitude CLATI) and any other individual (latitude CLATJ) that has not as yet been complexed is less than one half the sum of the lethal radius of each target, the target associated with latitude CLATJ is said to belong to a complex associated with latitude CLATI. Array LCOMP is updated to record this occurrence.

Target CLATI continues to be tested against subsequent targets in the sorted file until a difference in latitude greater than parameter

DIFFLAT is encountered. DIFFLAT is the maximum complexing lethal radius defined within the game. Targets included in the list LCOMP are now compared in the same way to find additional members of the complex. The process is repeated until all targets in the list LCOMP have been investigated, and the complex is completed. The complex then is assigned the next value of ICOMPL and each member within the complex are properly chained.

Subroutine COMPLEX is illustrated in figure 18.

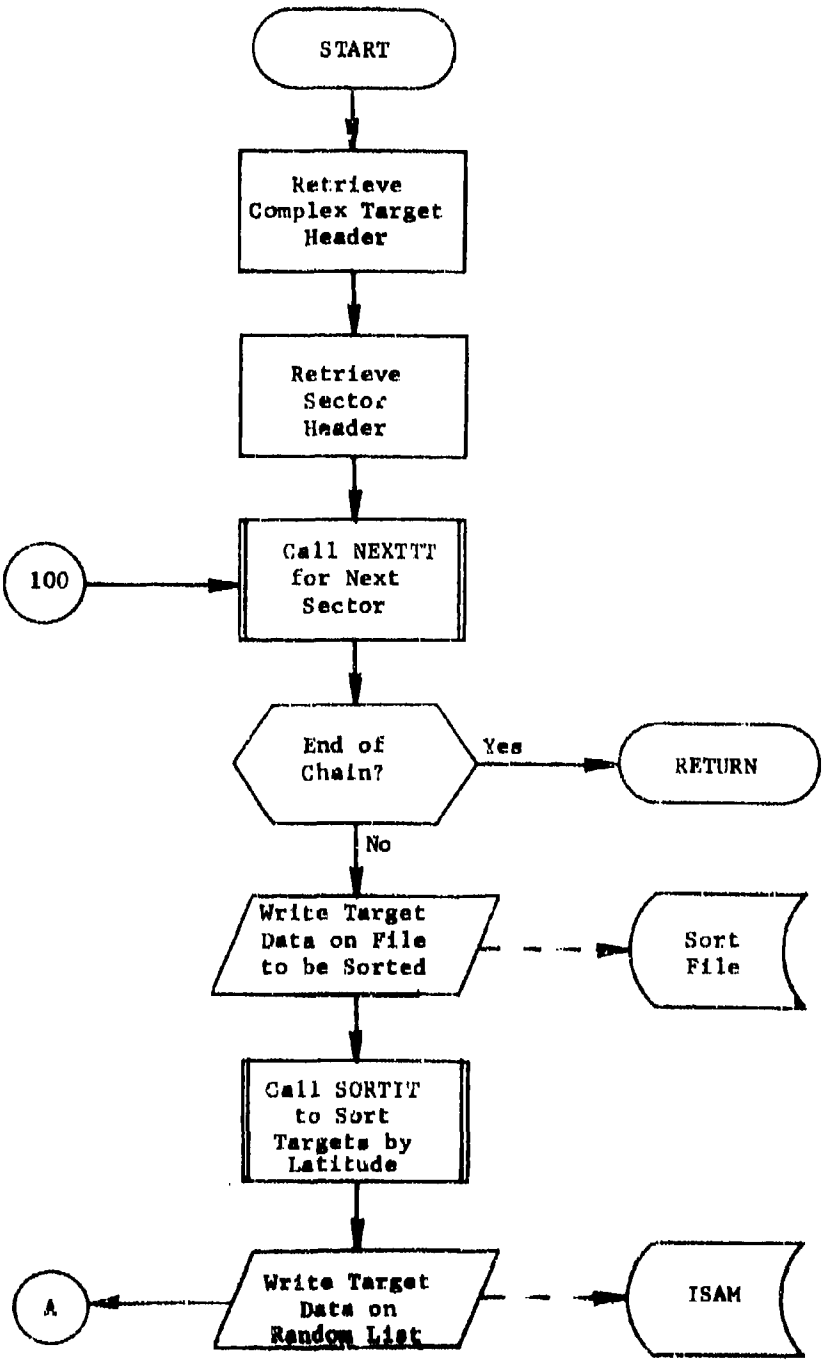


Figure 18. Subroutine COMPLEX (Part 1 of 6)

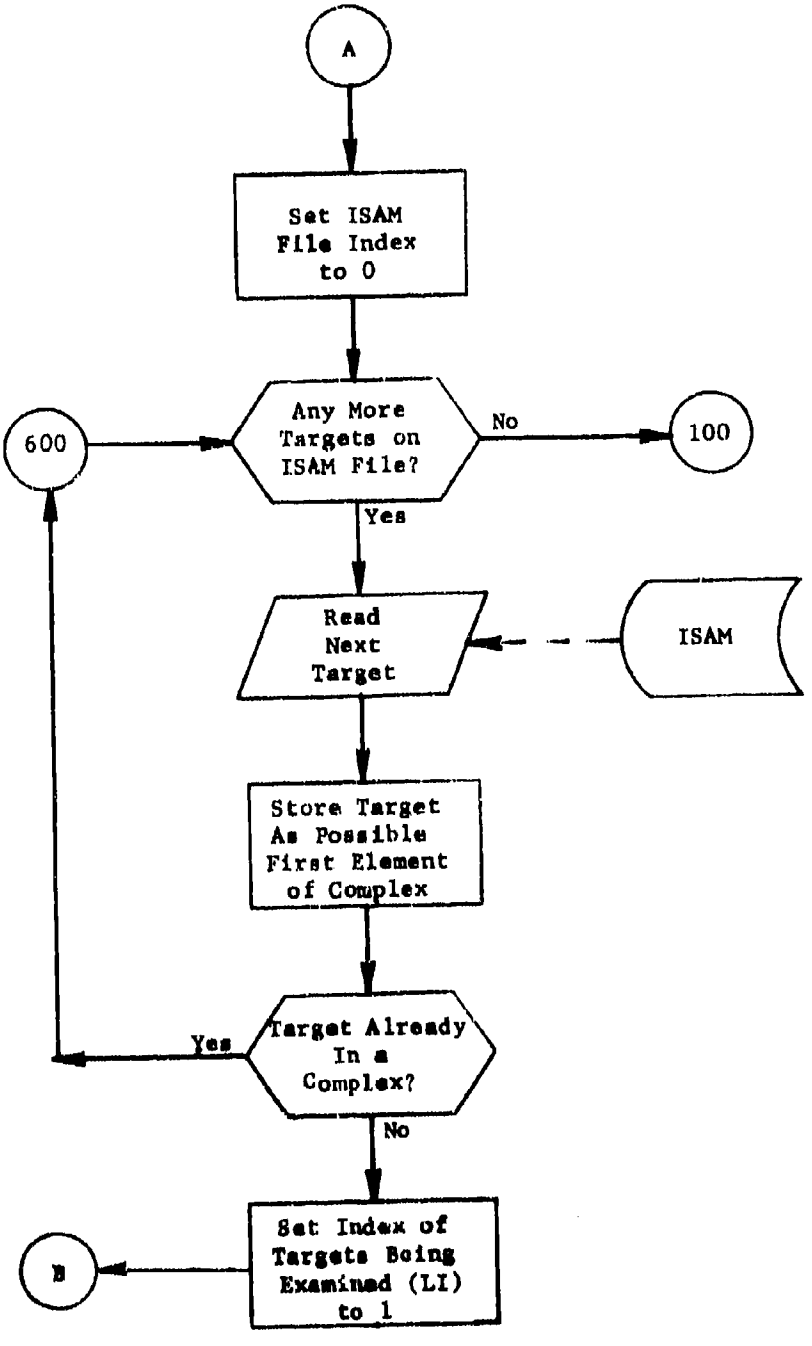


Figure 18. (Part 2 of 6)

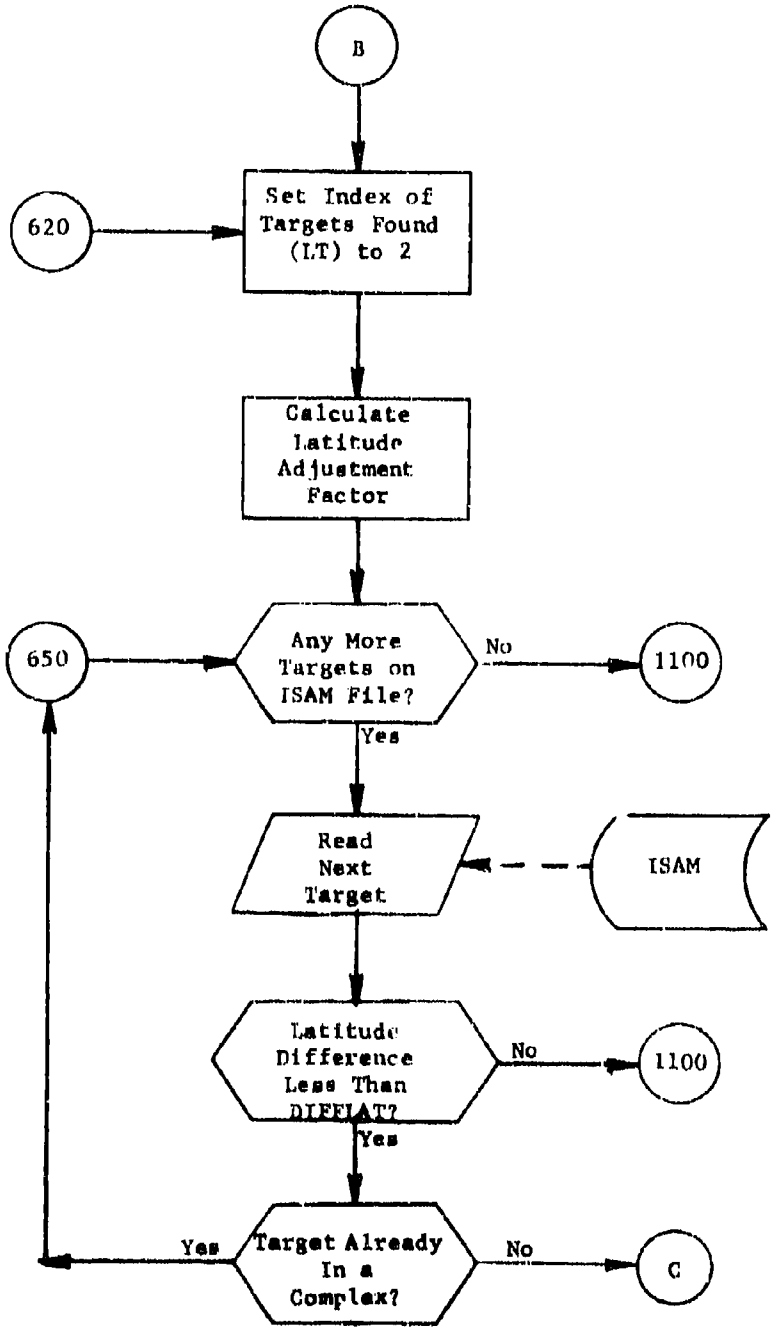


Figure 18. (Part 3 of 6)

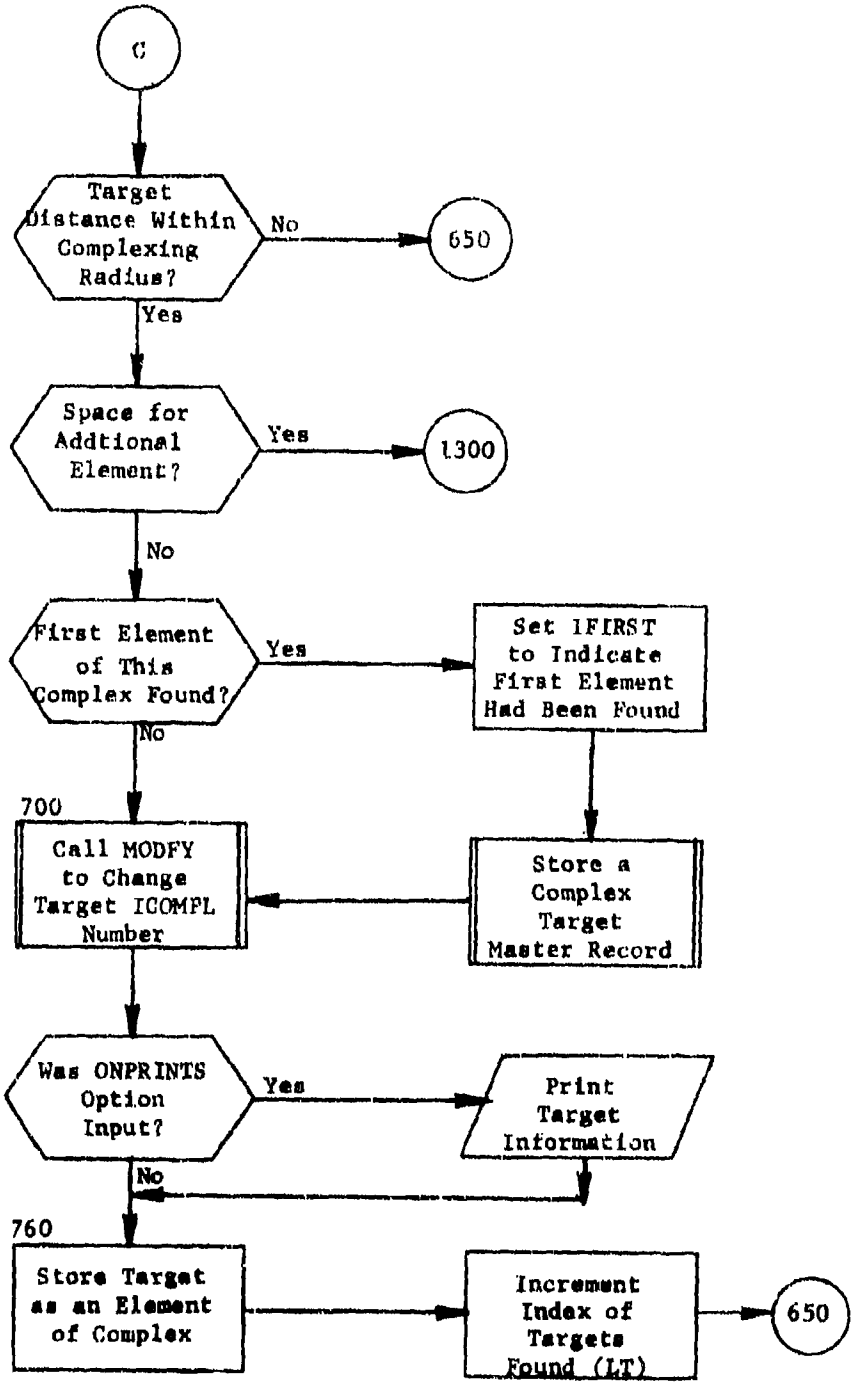


Figure 18. (Part 4 of 6)

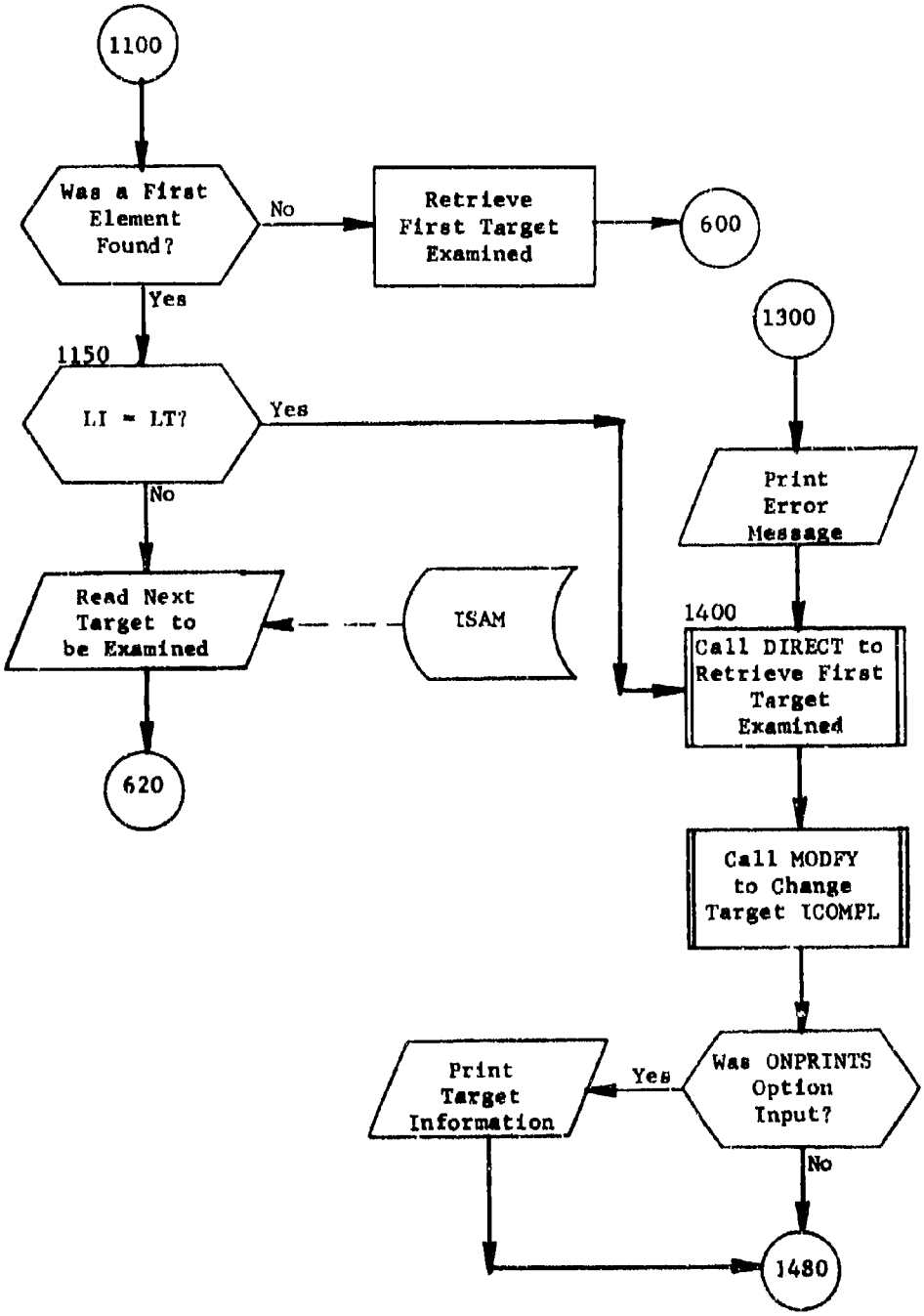


Figure 18. (Part 5 of 6)

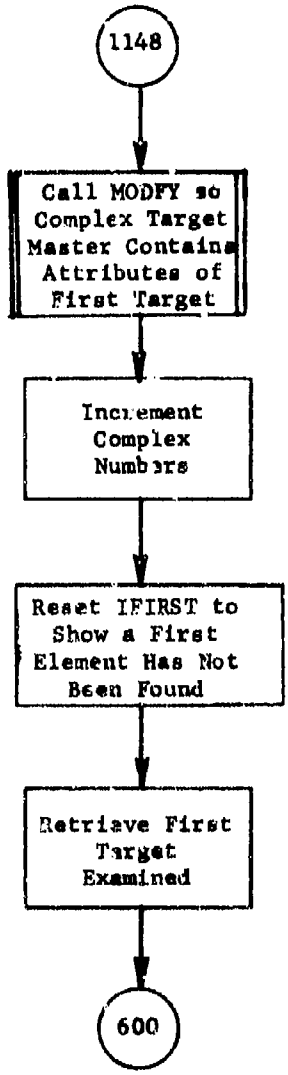


Figure 18. (Part 6 of 6)

4.9 Function CRTBLE

PURPOSE: To calculate complexing lethal radius

ENTRY POINTS: CRTBLE

FORMAL PARAMETERS: IVN, vulnerability

COMMON BLOCKS: None

SUBROUTINES CALLED: VLRADI

CALLED BY: ENTMOD (of overlay link INDXER)

Method:

Based on input parameter (IVN) complexing lethal radius is obtained by proper indexing into hard coded arrays (Q and P) which is the letter portion of vulnerabilities. Defining a yield as equaling the first two integers (in megatons) of IVN, subroutine VLRADI is called to find an adjusted VN which is turn is used as the index into the P and Q arrays.

Function CRTBLE is illustrated in figure 19.

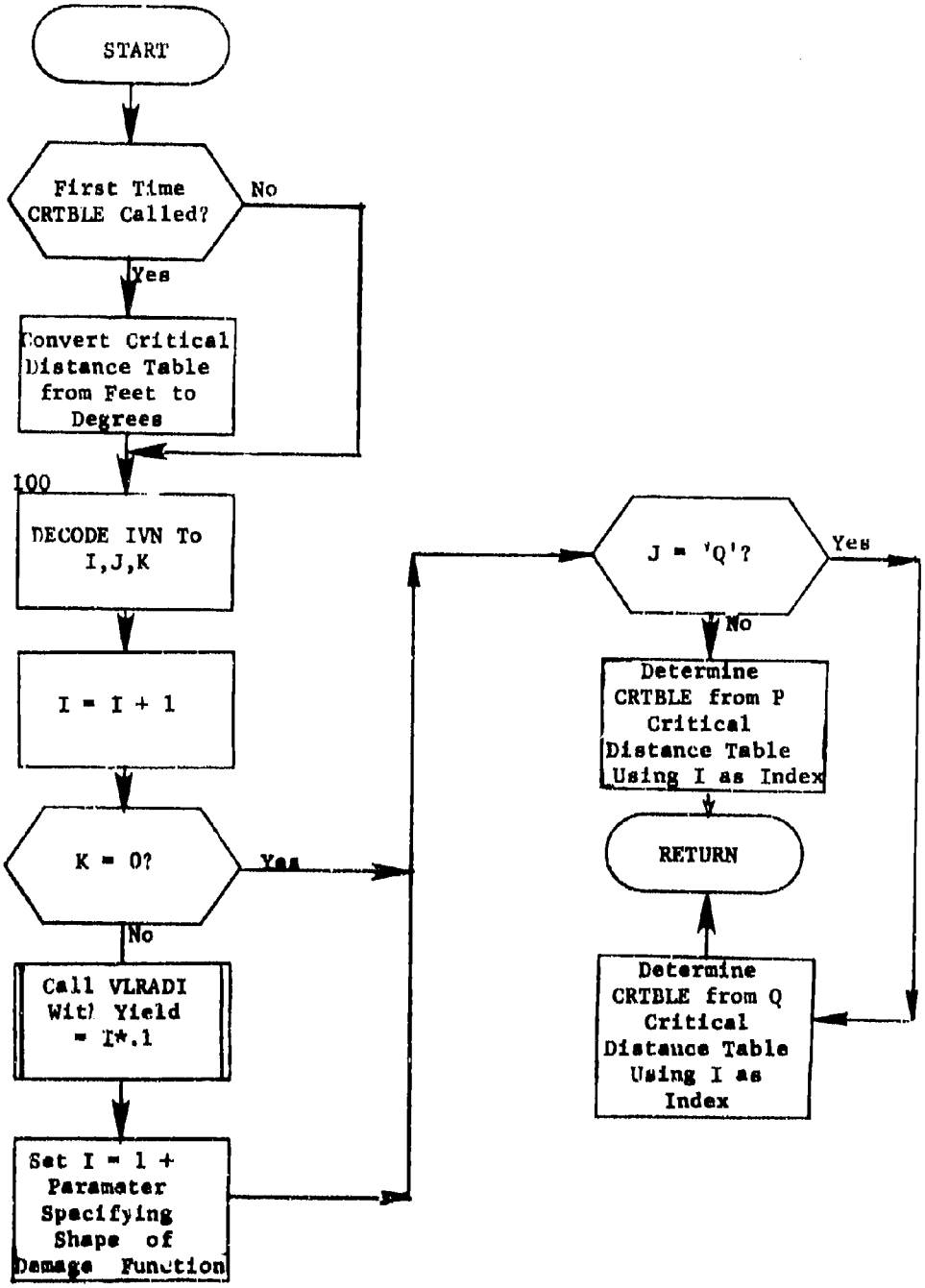


Figure 19. Function CRTBLE

4.10 Subroutine SETVAL

PURPOSE: To add time dependent value curves to missile and bomber target bases

ENTRY POINTS: SETVAL

FORMAL PARAMETERS: None

COMMON BLOCKS: C10, C15, C30

SUBROUTINES CALLED: DIRECT, HDFND, ITLE, NEXTTT, RETRY

CALLED BY: ENTMOD (of overlay link INDXER)

Method:

SETVAL (figure 20) sets the values of FVALTn and Tn attributes according to the attributes NOPERSQ, NALERT, ALRTDL, NLRTDL, LCHINT, and SIMLUN for missile and bomber bases. The value curve is calculated so that the value of the item is equal to the fraction of the vehicles remaining on base at any time.

The generated value curve considers the following time points:

T1 = Alert delay (ALRTDL)
T2 = Time of last launch of alert vehicle
T3 = Nonalert delay (NLRTDL)
T4 = Time of last launch of nonalert vehicle

On the first call to SETVAL all weapon types are chained and their associated Reference Codes stored along with attribute TYPE (arrays ITARTP and ITARRF). Following this storage, SETVAL begins by retrieving total vehicles (NOPERSQ), number of vehicles on alert (NALERT), alert delay time (ALRTDL) and nonalert delay time (NLRTDL). If there are no vehicles available (NOPERSQ), attribute VAL (target value) is set to zero and a RETURN is executed. For this situation, the value curve need not be generated. Attribute T1 is set to alert delay and FVALT1 is set to 1.0. If there is no launch interval (LCHINT), a RETURN is executed; otherwise T2 and FVALT2 are calculated. T3, T4, FVALT3 and FVALT4 are calculated only if T2 is less than the nonalert delay.

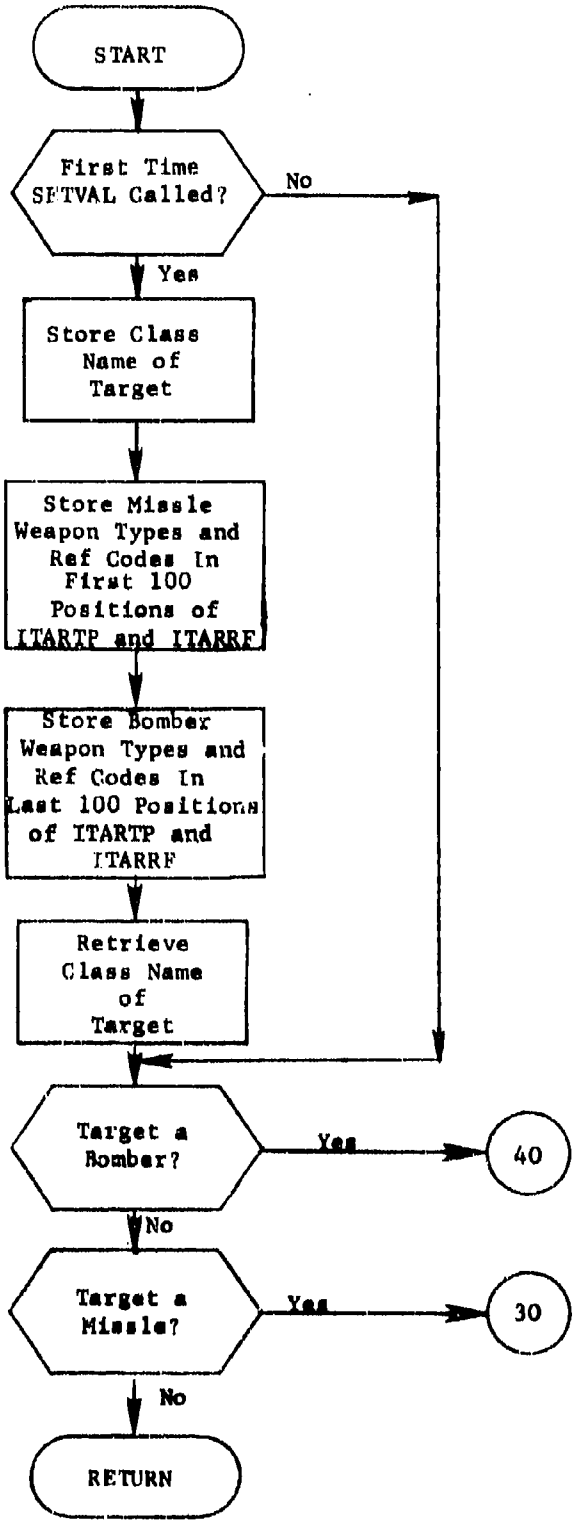


Figure 20. Subroutine SETVAL (Part 1 of 3)

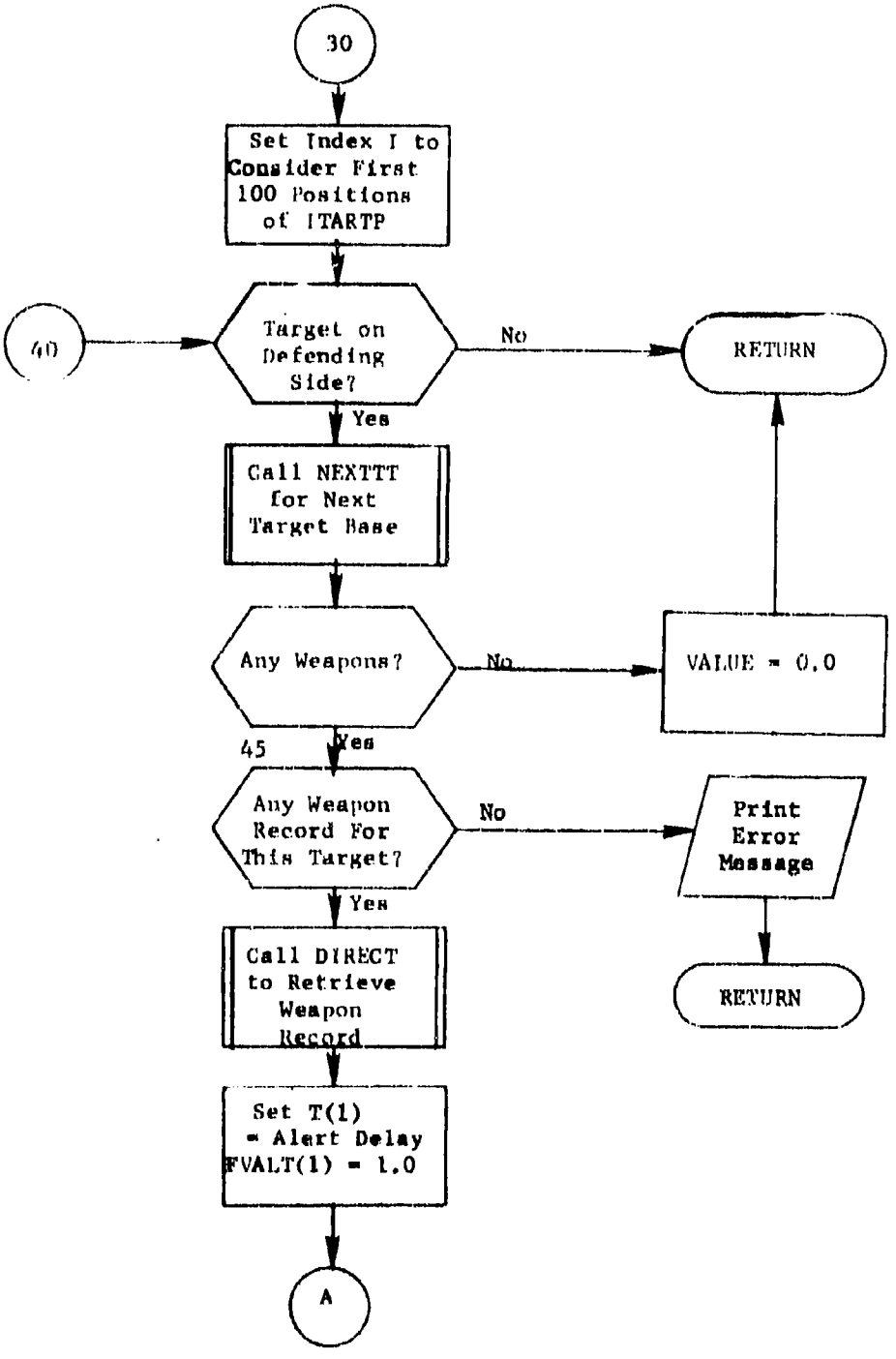


Figure 20. (Part 2 of 3)

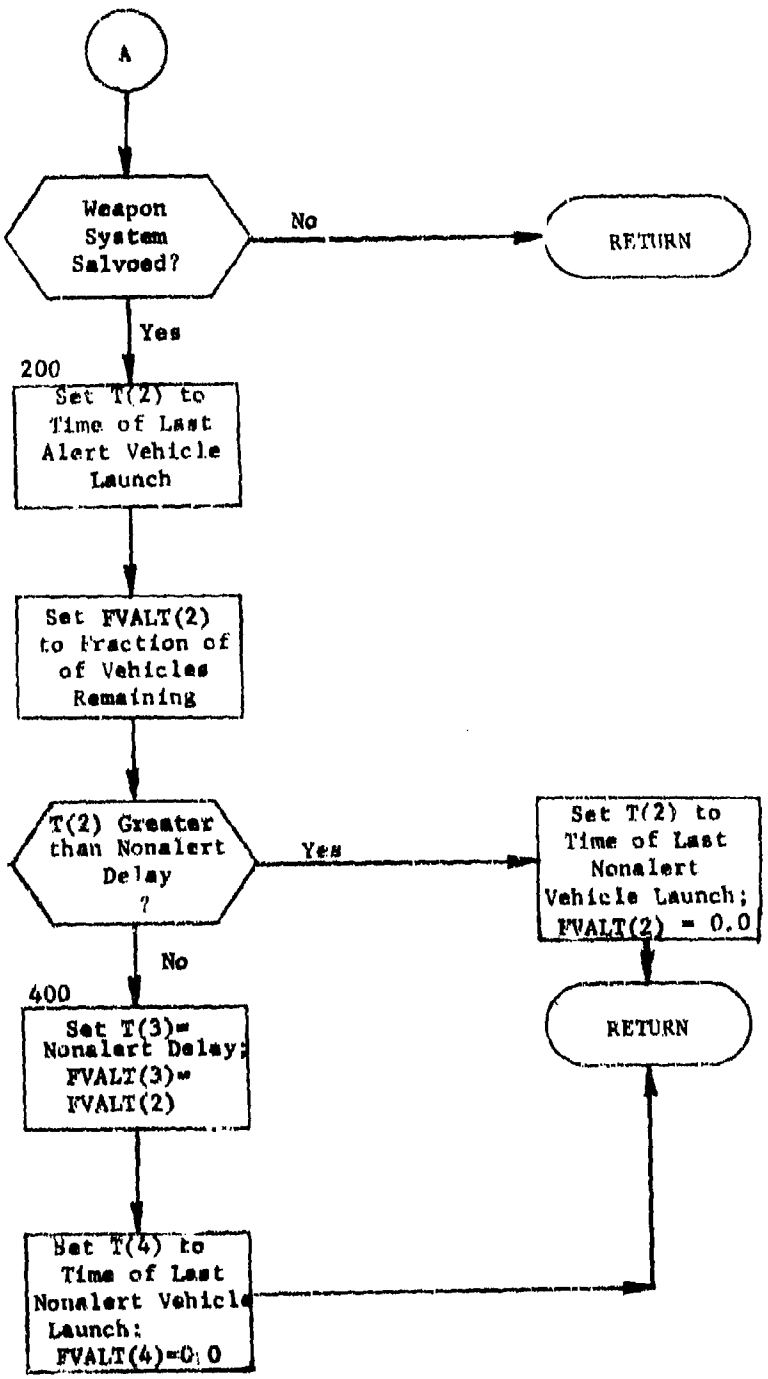


Figure 20. (Part 3 of 3)

4.11 Function VLRADI

PURPOSE: To find the lethal radius of a weapon delivered against a target of a specified vulnerability, and to set FN for use by the calling subroutine.

ENTRY POINTS: VLRADI

FORMAL PARAMETERS:
- YIELD - Yield of weapon in megatons
- NVN - Vulnerability parameter of target
- HOB - Weapon height of burst
- FN - Parameter specifying shape of damage function

COMMON BLOCKS: INDXRHL, RADATA

SUBROUTINES CALLED: EXP

CALLED BY: ENTMOD (of overlay link INDXER)

Method:

NVN is decoded into the appropriate vulnerability number VN, the letter (P or Q), and the K-factor XK. The cube root of the yield is extracted. Then the adjusted vulnerability number AVN is determined by methods described in "Computer Computation of Weapon Radius," B-139-61, Air Force Intelligence Center. FN is set to six or three of P and Q type targets, respectively.

Common block /RADATA/ contains four arrays (for the four combinations of P or Q vulnerability and air-to-surface burst) each of which contains the natural logarithm of the lethal radius (in nautical miles) of a 1-megaton burst. The data are at intervals of five vulnerability numbers. Function VLRADI interpolates in the appropriate array to find the logarithm of the 1-megaton lethal radius for AVN. The lethal radius of the weapon is then determined by exponentiating and multiplying by the cube root of the yield.

A flowchart for VLRADI is shown in figure 21.

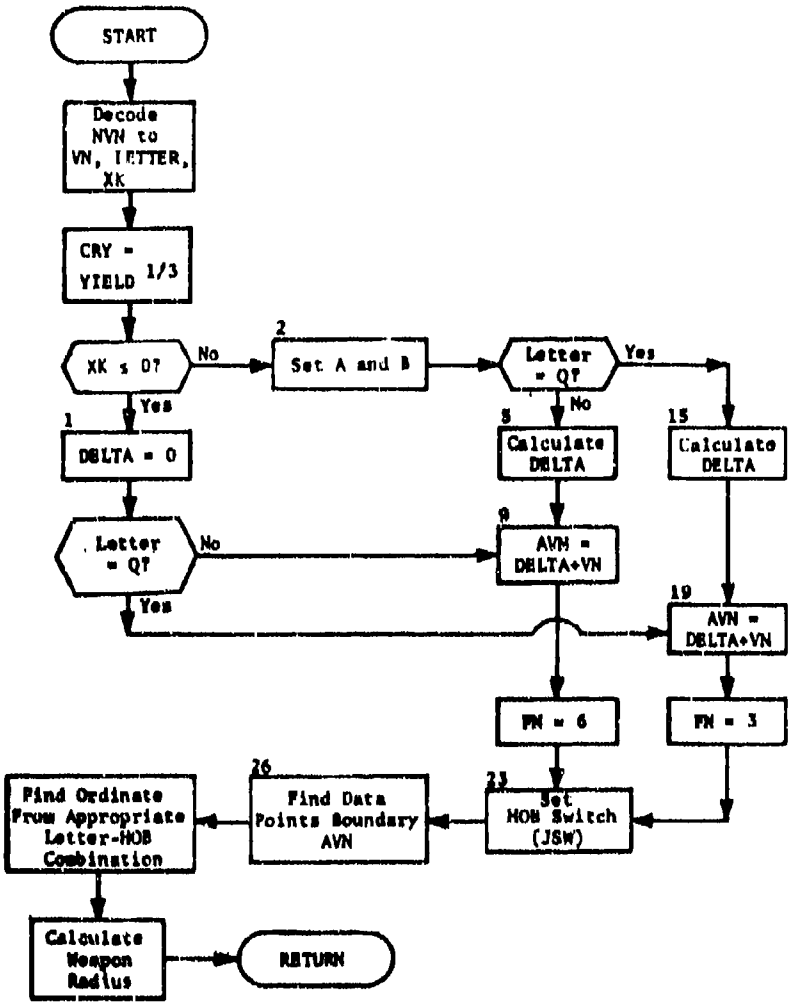


Figure 21. Function VLRADI

SECTION 5. PLANSET MODULE

5.1 Purpose

PLANSET prepares the target list for ALOC, computes and normalizes the class value factors, calculates the representative attributes for complex targets and forms weapon groups.

5.2 Input

With the exception of geographic related records, the entire remaining structure within the integrated data base will be queried. This includes records and chains to define the targets, complexes, weapon characteristics, payload data, and weapon base locations.

User commands select targets and weapons to be used within the allocation definition. In addition, information is supplied for value scaling calculation plus data for representative complex target setting.

5.3 Output

The major objective of PLANSET involves the formation of weapon groups and the definition of a list of target numbers as demanded by the allocation system. Also, individual target records are modified to reflect values as scaled through user inputs.

Each weapon group is a WEPNGP record on the WEPGRP chain. Each weapon group has a chain, MYSQDN, that links together the MSBMTG records for each base in the group. With regard to processing the data base it should be noted that the individual launch sites assigned to each missile squadron are grouped together; and that the value of the attribute ISITE (site number) is set positive for the record representing the site. Corresponding records that are missile squadrons within the site have ISITE set to a negative value. When input in this manner, the missile squadrons are viewed as one launch base during plan generation.

Generated within PLANSET is the chain, LISTXX of target number records, TARCDE, which is headed by record TARNUM. This chain contains a list of reference codes that points to target records in a sort order proper for ALOC processing. There are three types of targets:

a. Simple target: one target element

b. Complex target: several target elements within the lethal radius of a single weapon so that they must be treated as a single target complex

c. Multiple targets: Actually several independent identical targets such as separate missile silos in a Minuteman squadron that are close together (relative to the range of the weapon) but for enough apart that each target element must be treated as an independent aim point.

The list of reference codes, then, contained within TARCDE records stores simple target references which point to one target element (called representative targets) for each complex and multiple target formation. Individual elements of complexes or multiples are not referenced by chain LISTXX; therefore they are viewed by the allocator.

Prior to PLANSET termination, each complex and multiple target has its elements chained under a COMPTG record. The maximum number of complexes has been defined by INDEXER. Multiple targets reside on the CMPHD header and have complex number (attribute ICOMPL) values that are greater than INDEXER defined.

Each TARGET record selected must be modified to include the correct normalized target value as well as the lethal radius for both air and ground bursts. If necessary there are lethal radius calculations for two hardness components.

5.4 Concept of Operation

PLANSET performs the following:

- store user input parameters;
- calculate the weighting value of each selected target class;
- calculate the lethal radius of all selected targets;
- chain collected multiple targets;
- for each complex, choose the representative target and calculate the necessary attributes;
- normalize the value of each target so that the game total will be 1,000 points;
- randomly assign target numbers (and eventually sort) to selected target records;
- form weapon groups from weapon launch bases;
- adjust the number of alert bomber refuels based on the number of tankers available;
- finally, supply prints.

The controlling subroutine (called ENTMOD) reads and stores user requests and then executes subroutines GRPEM (for grouping), SRTTGT (for target number definition), and PRINTGP (for printing). Each of these three subroutines performs the desired objectives.

5.5 Identification of Subroutine Functions

5.5.1 Subroutine ADJUSTGP. This subroutine is called by GRPEM after weapon groups are formed. ADJUSTGP alters the number of bomber refuels so that it is less than or equal to the number of tankers and resets yields for each weapon group.

5.5.2 Subroutine CALCOMP. Whenever subroutine SRTTGT encounters a complex target, CALCOMP will define the representative target element for that complex as well as determine necessary attributes for the complex as a whole.

5.5.3 Subroutine GRPEM. The user selected weapon system types are chained along with their associated missile and bomber bases and weapon groups are formed. A weapon group is simply a collection of individual weapons (or reentry vehicles) that have similar destruct capability and are located within defined proximity of each other. Therefore, a weapon group consists of a TYPE (an attribute) of weapon that has the same alert status, payload indicator, region code, and for bombers, refuel capacity.

5.5.4 Subroutine SRTTGT. In order for the allocation process to function properly it is best for the target list to be arranged in a randomized fashion. This necessity is dictated by the fact that if targets were arranged with like characteristics, the allocation algorithms in sensing how processing is proceeding would very likely interject biases This subroutine, then, sorts the target list and for each complex, calls subroutine CALCOMP for proper chaining.

A second major function is the modification of the target value for individual records. Targets for which the input class value is zero (i.e., an exemplar target is not defined for the class or the exemplar target is assigned a value of zero) are not to be included in the plan and hence are ignored in the processing. Otherwise, the data base attribute VAL (relative value within class) for each item is accumulated within its class. For each exemplar target specified by the input data, the value factor

$$\frac{\text{data card value for exemplar target}}{\text{data base VAL for exemplar target}}$$

is calculated. After the entire data base has been read, the accumulated value (VALs), together with the value factors for each class, are used to

compute the final normalized class value factors. Also, the summation of the VALs for each record is so scaled such that its result equals 1000. This scaling allows for allocation evaluations between various data bases.

Multiple targets are made up for missile sites which do not belong to a complex. A multiple target consists of at least two and not more than five consecutively indexed sites from the same squadron.

5.6 Internal Common Blocks

All of the common blocks used by module PLANSET are given in table 4. Common blocks which communicate with the COP are given in appendix A of Maintenance Manual, Volume I.

Table 4. Module PLANSET Internal Common Blocks (Part 1 of 2)

BLOCK	VARIABLE OR ARRAY	DESCRIPTION
CPRIOR	MAXDSG	Maximum number of input DESIGs alpha portions. Used for representative target definition
	MAXTASK	Maximum number of input TASKs. Used for representative target definition
	ITSK	Number of entries in array IPTSK
	IDSG	Number of entries in array IPDSG
	IPTSK(48)	Stores TASK inputs for defining representative target of complexes
	IPDSG(200)	Stores DESIG alpha portion input for defining representative target of complexes
	ISUBT	Flag indicating whether TASK inputs are 1 or 2 characters long
EXCLAS	UCLASS(15)	Stores names of target classes for defending side
	UCREF(15)	Stores raference codes for target class headers for defending side
	VALFAC(15)	Stores scaling factors for each target class
MASK	MASK1	Testing parameter for attribute TASK
	MASK2	Testing parameter for attribute TASK
PBLOK	PBLOK(28,10)	Used in SRTTGT for target designator, number print and in PRINTGP for weapon and group prints
SET	RANGEMOD	Fraction of weapon system range for grouping use
	RETARGET	Nonzero for missile retargeting capability

Table 4. (Part 2 of 2)

BLOCK	VARIABLE OR ARRAY	DESCRIPTION
SET (cont.)	CCREB(20)	Command and control reliability for regions
	NCCREL	Number of input values for CCREB
TARCLAS	MAXCLAS	Maximum number of target classes
	INCLAS	Number of user selected target classes
	INDESIG(15)	DESIG of exemplar target for selected class
	EXPVAL(15)	Value of selected exemplar target
WEAPON	MAXW	Maximum number of weapon systems permitted for processing
	IWEAP	Number of user selected weapon systems
	INWEAP(100)	Names of user selected weapon systems
	IREFW(100)	Reference codes of user selected weapon systems

5.7 Subroutine ENTMOD

PURPOSE:	Read and store user inputs and ontrol flow of supporting subroutines
ENTRY POINTS:	ENTMOD (first subroutine called when overlay link PLANS is executed)
FORMAL PARAMETERS:	None
COMMON BLOCKS:	CPRIOR, SET, TARCLA, WEAPON
SUBROUTINES CALLED:	CINSGET, GRPSM, INSGET, PRINTGP, SRTTGT
CALLED BY:	COF

Method:

In addition to controlling the flow of supporting subroutines, ENTMOD mainly reads and stores user input data (figure 22). The verb and adverbs recognized by this module are:

- PLANSET - the verb that causes execution
- SETTING - the adverb which introduces a clause to set parameters RANGEMOD, RETARGET or CCREL
- PRIORITY - the adverb which introduces a clause to set criteria for choosing representative targets of complexes. Criteria is ordered lists of TASK and alpha-portion of the DESIG
- ATTACKERS - the adverb which introduces a clause to select weapon system inventory. Input is a list of values for attribute TYPE
- DEFENDERS - The adverb which introduces a clause to select target classes. Inputs are a pair of words for DESIG (and hence the target class that the DESIG defines) and exemplar value of each DESIG.

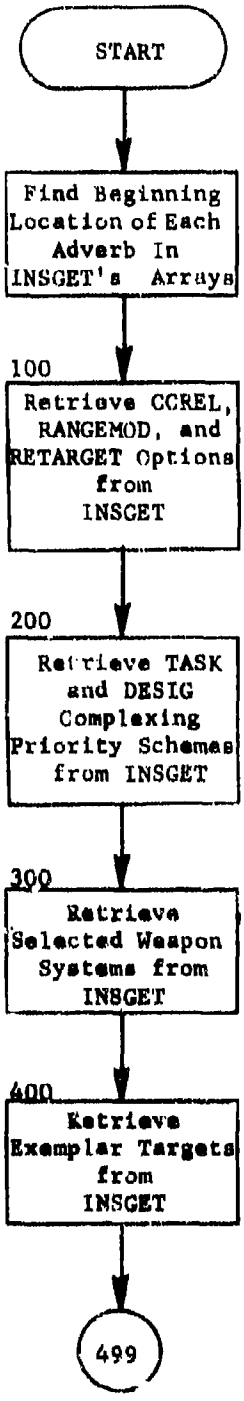


Figure 22. PLANSET Module (Part 1 of 2)

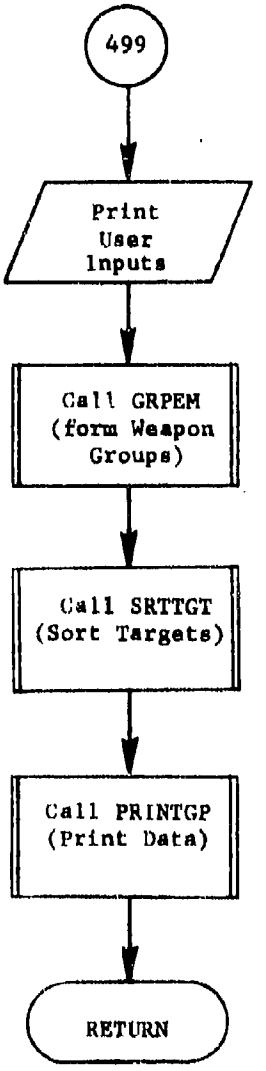


Figure 22. (Part 2 of 2)

5.8 Subroutine ADJUSTGP

PURPOSE: Adjust number of bomber refuels so that it is less than or equal to the number of tankers and, further, reset yield for each group

ENTRY POINTS: ADJUSTGP

FORMAL PARAMETERS: None

COMMON BLOCKS: C10, C15, C30

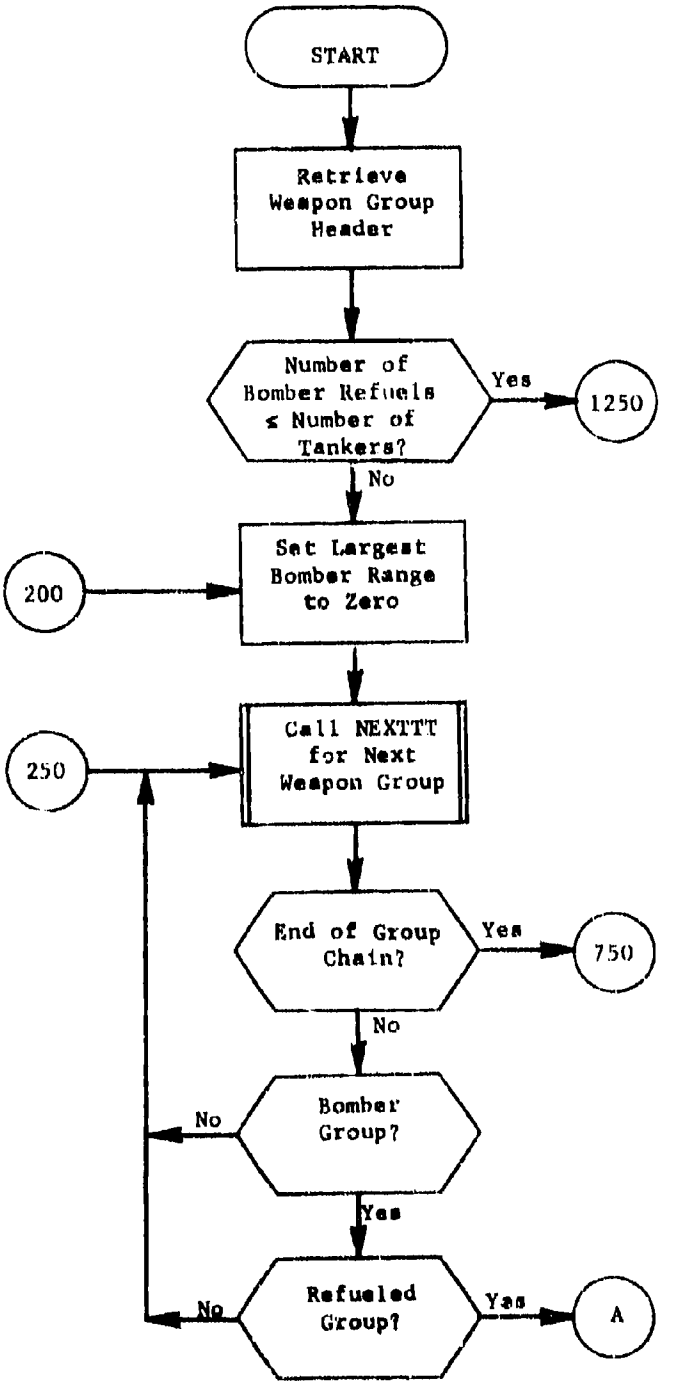
SUBROUTINES CALLED: DIRECT, HDFND, HEAD, MODFY, NEXTTT, RETRV

CALLED BY: GRPEM

Method:

After all data base items have been processed, the total number of bomber refuels (NBOMB) and tankers (NTANK) are compared. If there are more bomber refuels than tankers, the bombers on the nonalert base with the largest range are changed to nonrefuel (IREFUEL=0); NBOMB is decremented by the number of bombers on the base. This process continues until there are more tankers than bomber refuels. If IREFUEL is changed to zero for all the nonalert bases before the bomber-tanker balance is achieved, the alert bases are then examined and IREFUEL is changed as above. When the bombers and tankers have been balanced, the yield for each group is computed. During grouping, yields were accumulated for each weapon base record encountered. Therefore, these accumulated yields must be scaled according to the number of weapons within each group.

Subroutine ADJUSTGP is illustrated in figure 23.


START
Retrieve Weapon Group Header
Number of Bomber Refuels ≤ Number of Tankers?
Yes
1250
No
200
Set Largest Bomber Range to Zero
250
Call NEXTTT for Next Weapon Group
End of Group Chain?
Yes
750
No
Bomber Group?
No
Yes
Refueled Group?
No
Yes
A

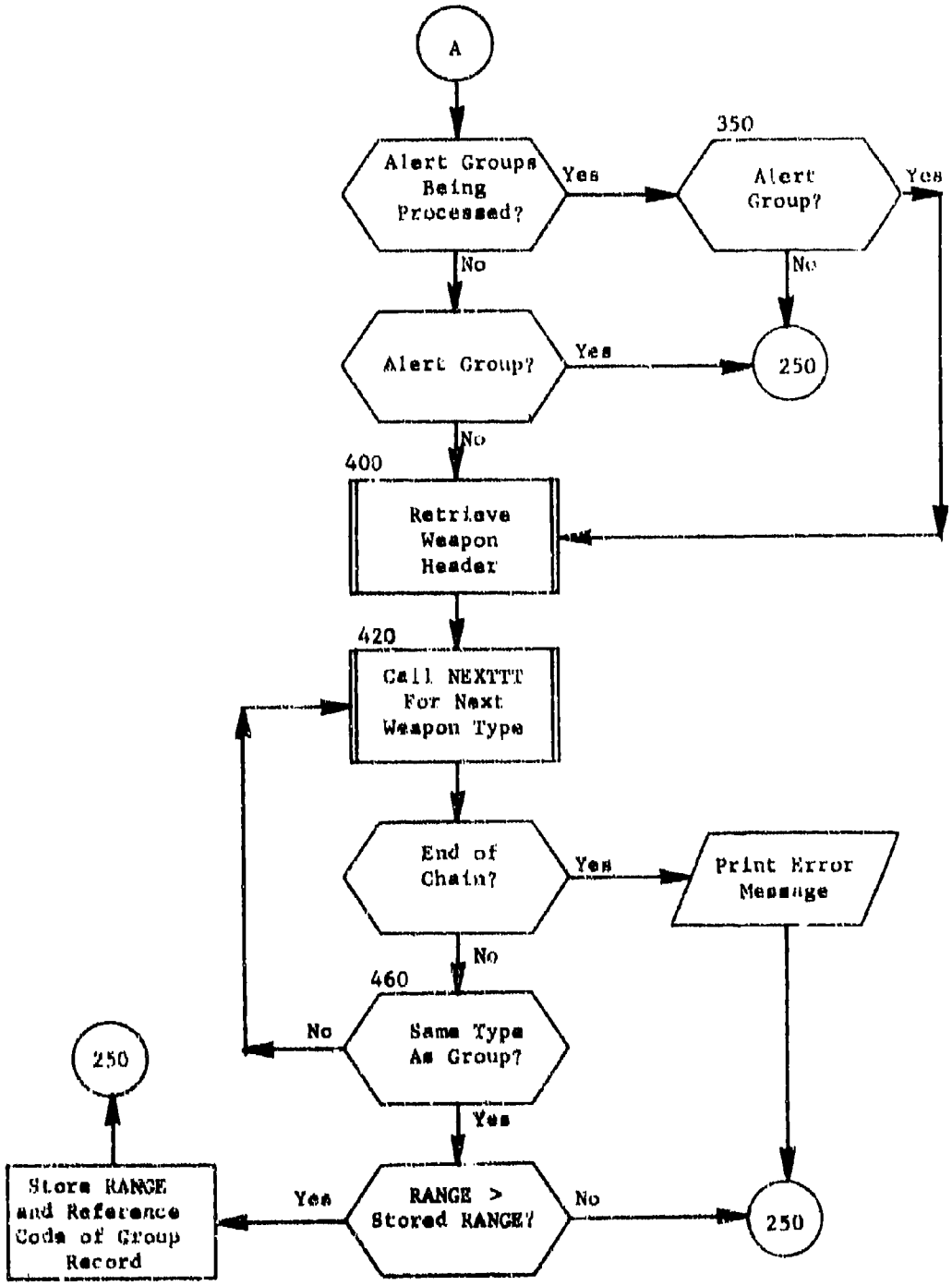


Figure 23. (Part 2 of 4)

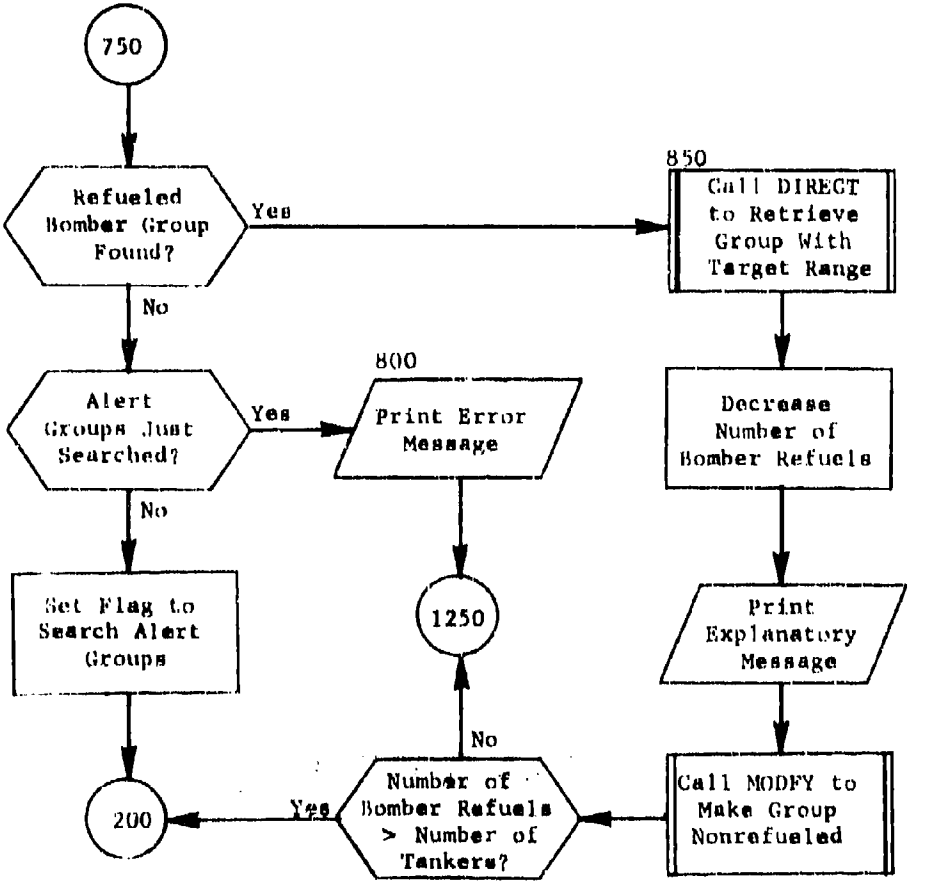


Figure 23. (Part 3 of 4)

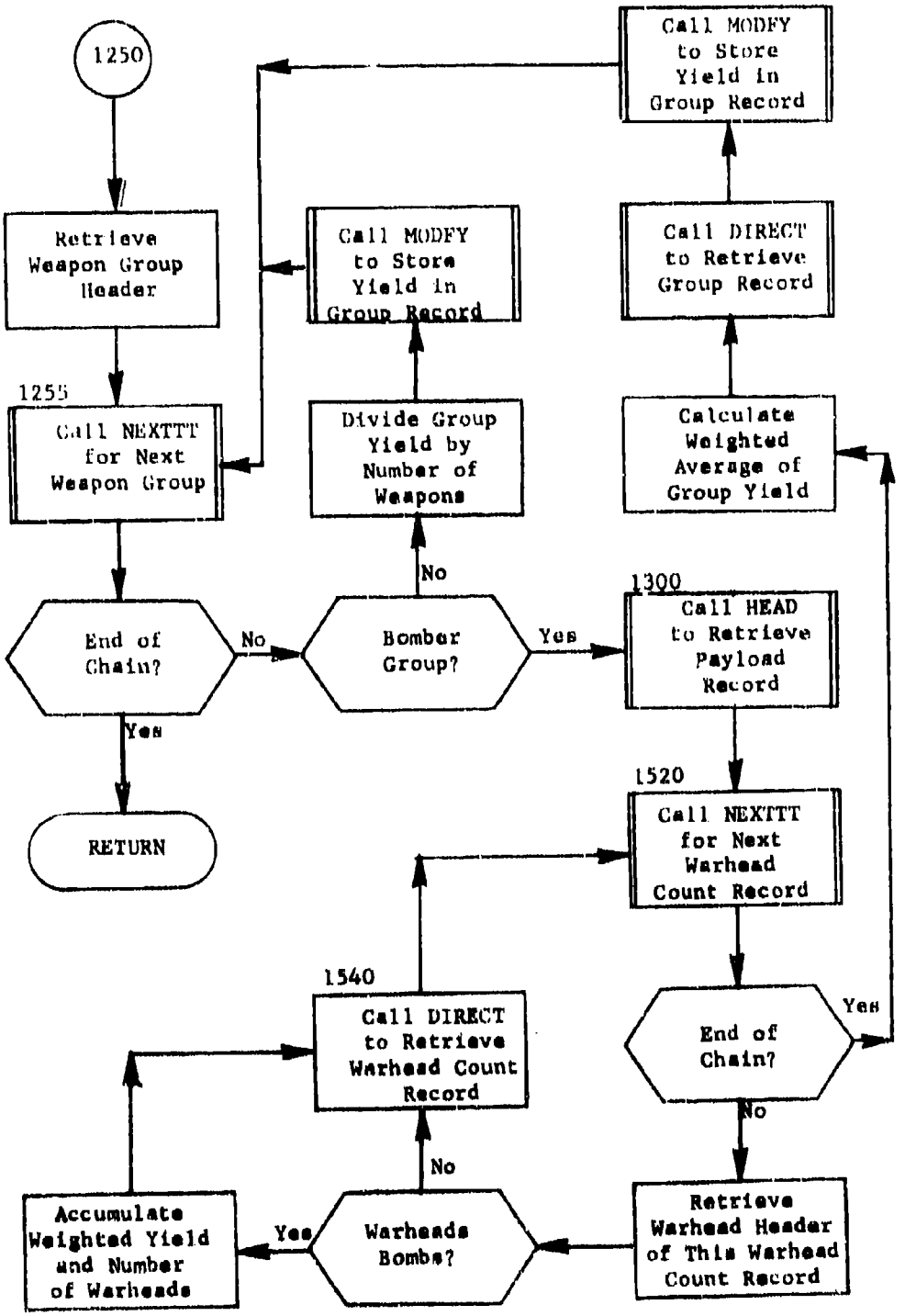


Figure 23. (Part 4 of 4)

5.9 Subroutine CALCOMP

PURPOSE: To calculate data which represent a complex target from data for the elements of the complex

ENTRY POINTS: CALCOMP

FORMAL PARAMETERS: None

COMMON BLOCKS: CPRIOR, C10, C15, C30, MASK

SUBROUTINES CALLED: DIRECT, HEAD, IGETHOB, MODFY, NEXTTT, ORDER, REORDER, VALTAR, VLRADA

CALLED BY: SRTTGT

Method:

CALCOMP begins by saving the reference code of the complex chain master record and comparing the TASK and DESIG of each component of the input lists of priorities in order to choose the 'representative' component for that target. It then stores values for the representative under the 'COMPTG' record. Values are now initialized for future processing.

A search is made for the maximum target radius, which is assigned as the radius of the complex. Similarly, the maximum value of TARDEF is found and assigned to represent the complex. The target value VOZ, the number of terminal interceptors (MISDEF), and the weighted (by VOZ) attributes MINKILL and MAXKILL are accumulated as each target is encountered. Also, for each target, the time components and the corresponding actual value lost at that time are placed sequentially in the arrays V and TAU.

Subroutines ORDER and REORDER are called to arrange the elements of TAU in numerical order and to place the elements of V in the corresponding order. Those ordered arrays are used to approximate the time dependence values for the complex (T1, T2, T3, T4, T5) in the following manner.

First the array TAU is checked for equal time components. If any are found, the corresponding values are added together, and all equal components but the last are set to zero. When the entire array has been checked it is collapsed to eliminate any zero components. If the number of remaining entries does not exceed five the time dependance of the value is approximated by these time components. Otherwise, an elimination procedure to reduce the number of entries to five is begun. To accomplish this, the slopes (change in value per change in time) are calculated for all remaining value points and the value point that

produces the smallest slope is grouped together with its neighboring value point. Hence the length of the TAU array is reduced by one. The TAU array is repetitively collapsed again, and slopes recalculated until there are five or less points remaining.

Once the elimination process is complete, the fractional value is computed for the first two components from the sums now stored in V(1) through V(5). These fractions, together with the time components in TAU and the total number of components (KK), are stored in array TTAR.

The lethal radius for air bursts must be recalculated for a uniform height of burst for all elements within the complex. This is required since the air lethal radius as calculated from VLRADP (called from PLANSA) assumed an optimal air height of burst for each target. Clearly, one height of burst is required for an air burst over a complex. That height of burst is defined in CALCOMP as the optimal scaled height of burst associated with the hardest element in the complex. The smallest ground lethal radius is defined as being the hardest element in the complex.

Calculation continues to determine the hardness components (HAZ, HGZ, HAZ2, HGZ2) and the corresponding fractional value (FVULN1) which represent the complex. VOZ, FVULN1, and the hardness number (1 or 2) are also recalculated based on the defined scaled height of burst. The complement of FVULN1 is found to represent the second hardness component. If either fractional value is nonzero, it is multiplied by VOZ to get the actual value at that hardness. After all targets have been considered, the lethal radii are separated into radii belonging to hard targets (radii less than 1.5 nautical miles) and radii belonging to soft targets. The average lethal radius, weighted by the actual value at the corresponding hardness, is calculated for both hard and soft targets for those radii. Similarly, the actual value at each hardness (VHARD or VSOFT) is accumulated. If there are no hard targets (i.e., VHARD=0), FVULN1 is set to 1; otherwise the fraction of actual value for hard targets (NHARD/VTOT) is assigned to FVULN1.

After all targets in the complex have been processed as above, the stored values are defined with the target record called 'COMPTG'.

See figure 24 for the logic flow within CALCOMP.

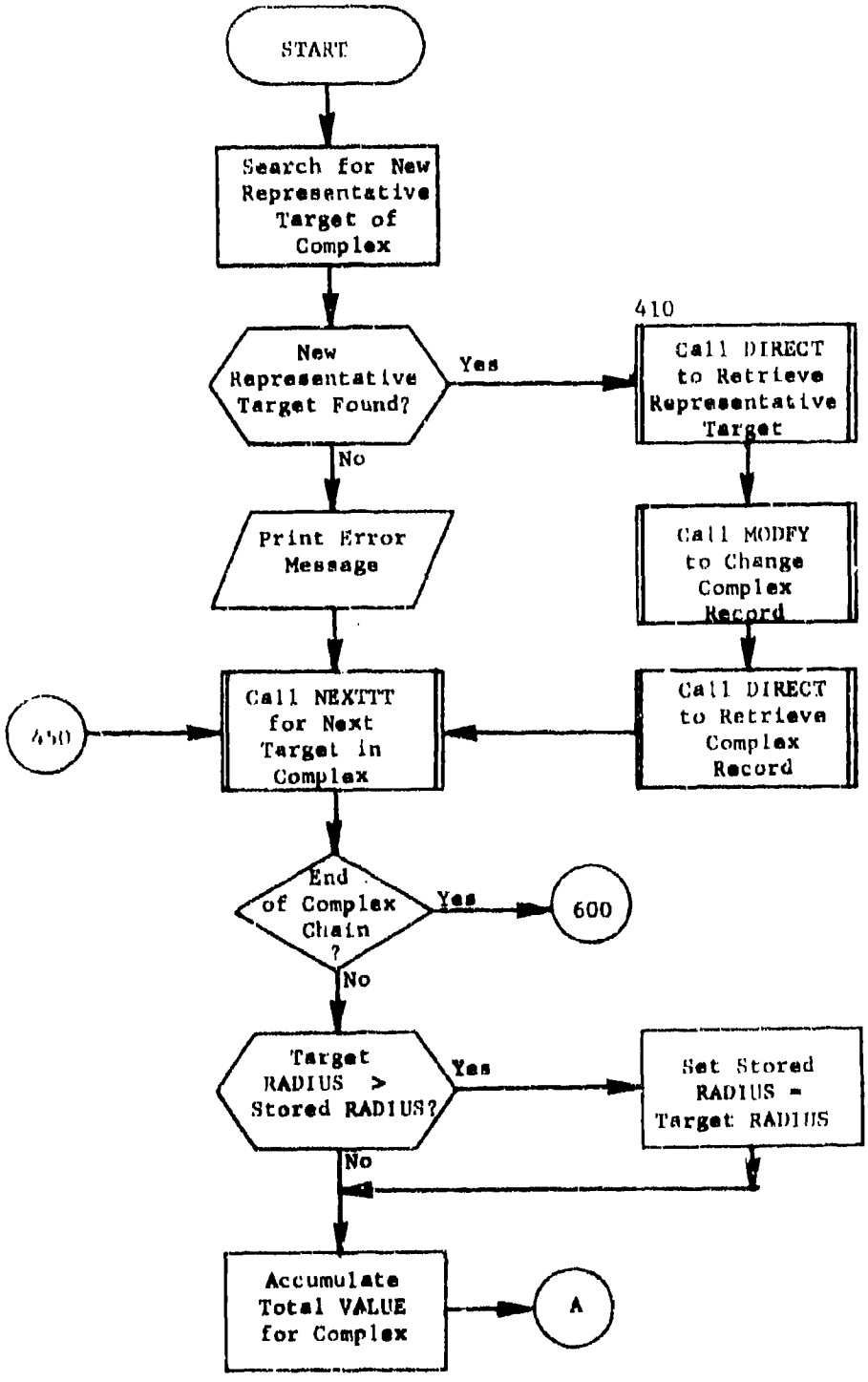


Figure 24. Subroutine CALCOMP (Part 1 of 5)

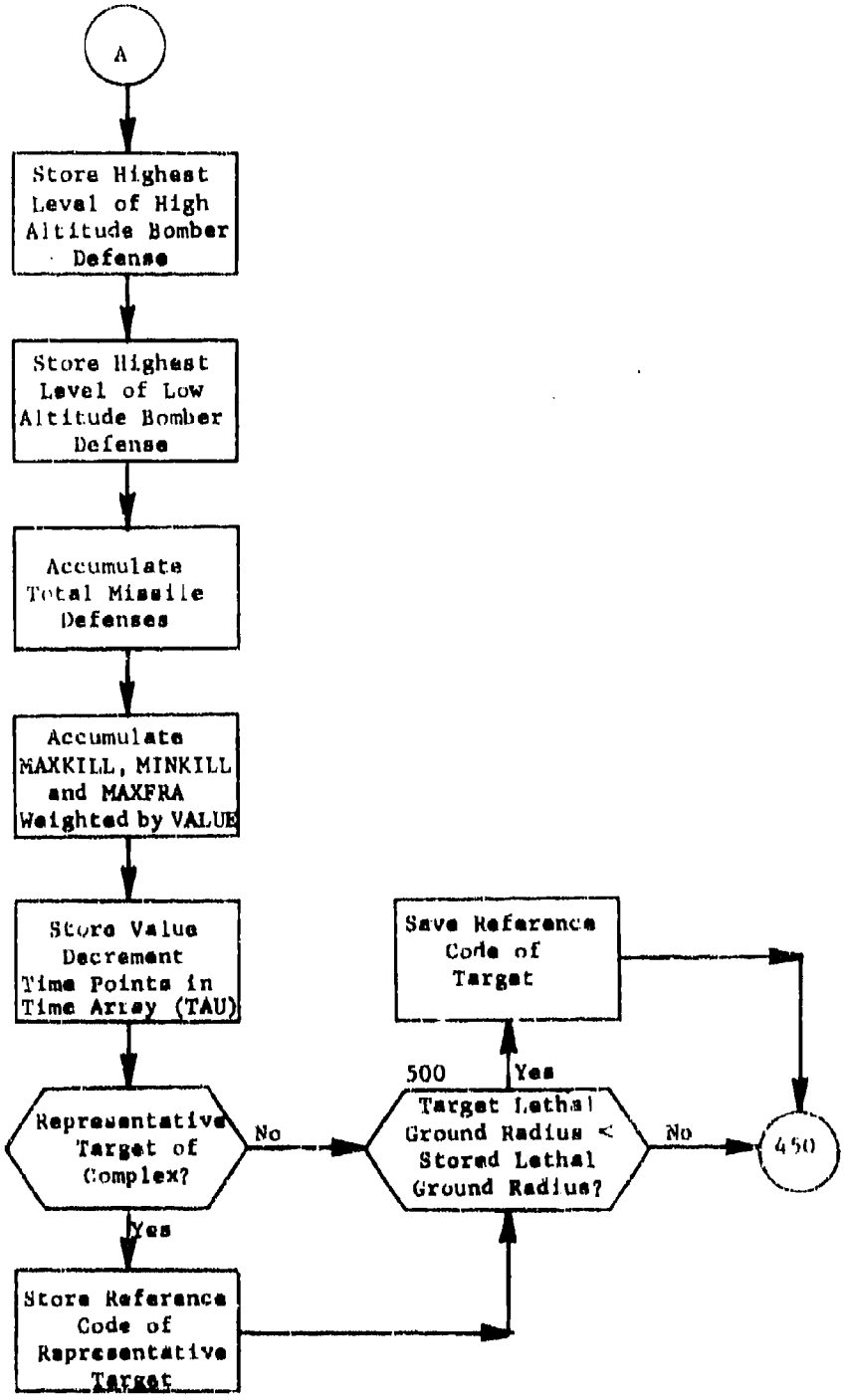


Figure 24. (Part 2 of 5)

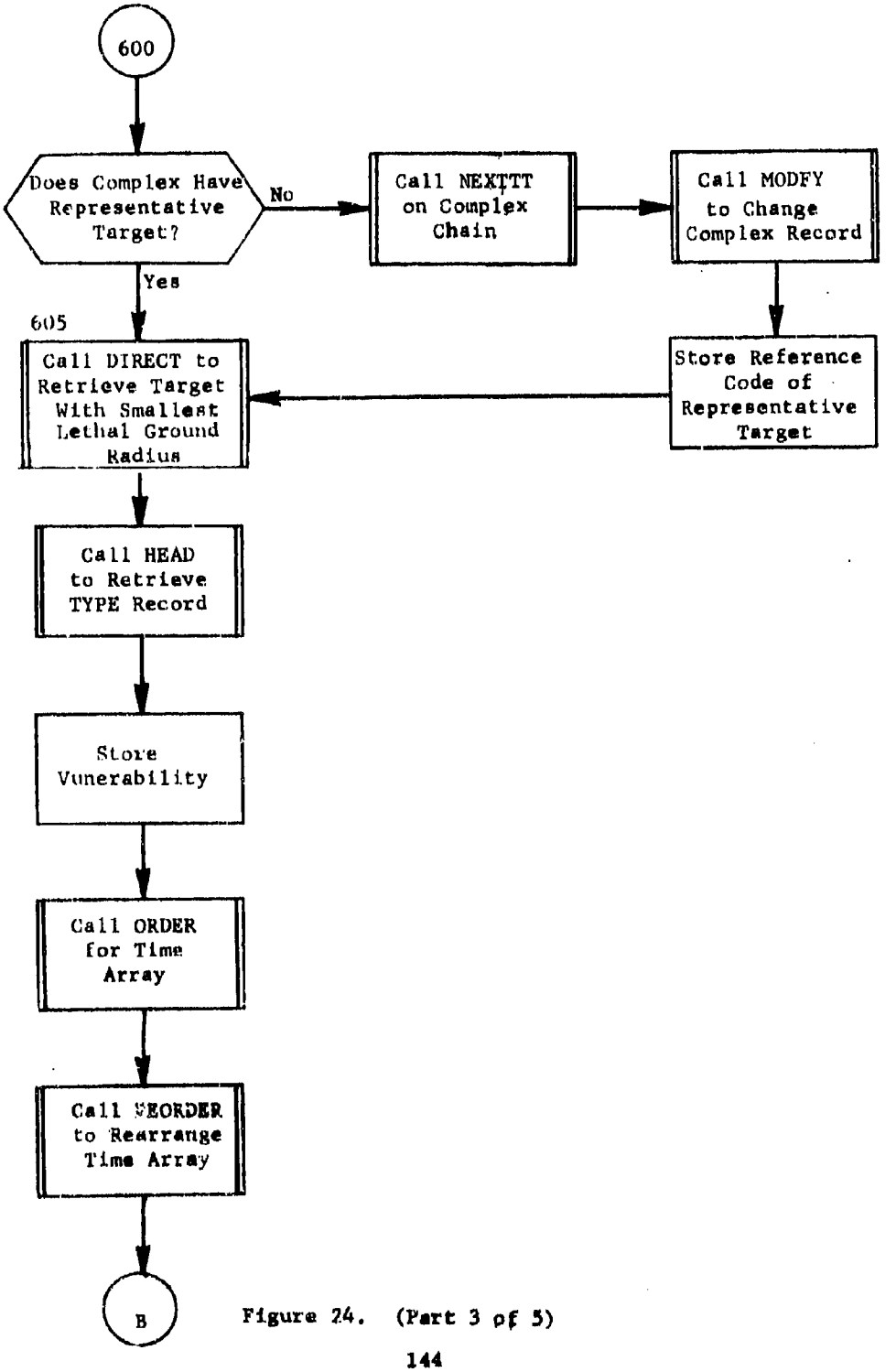


Figure 24. (Part 3 of 5)

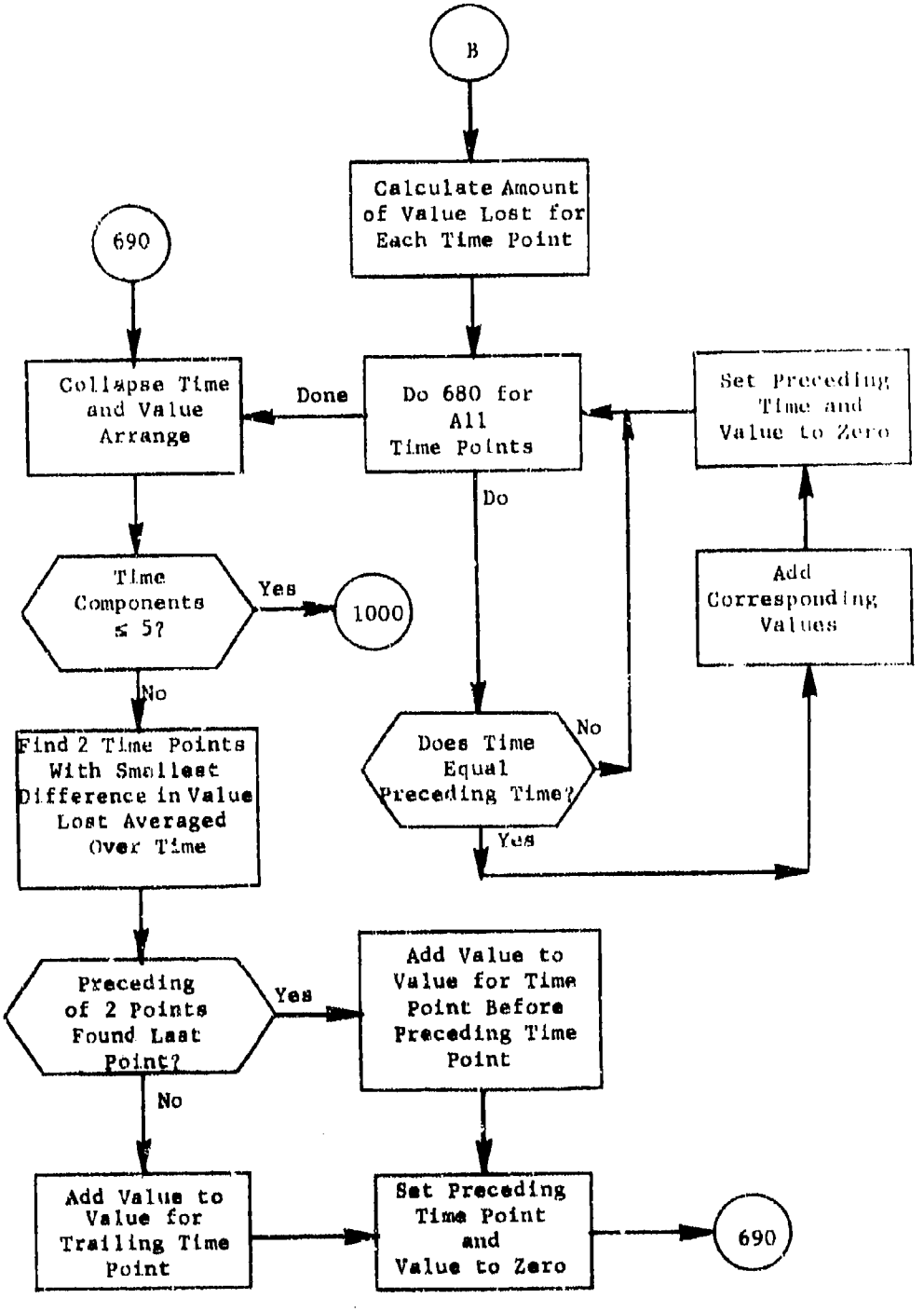


Figure 24. (Part 4 of 5)

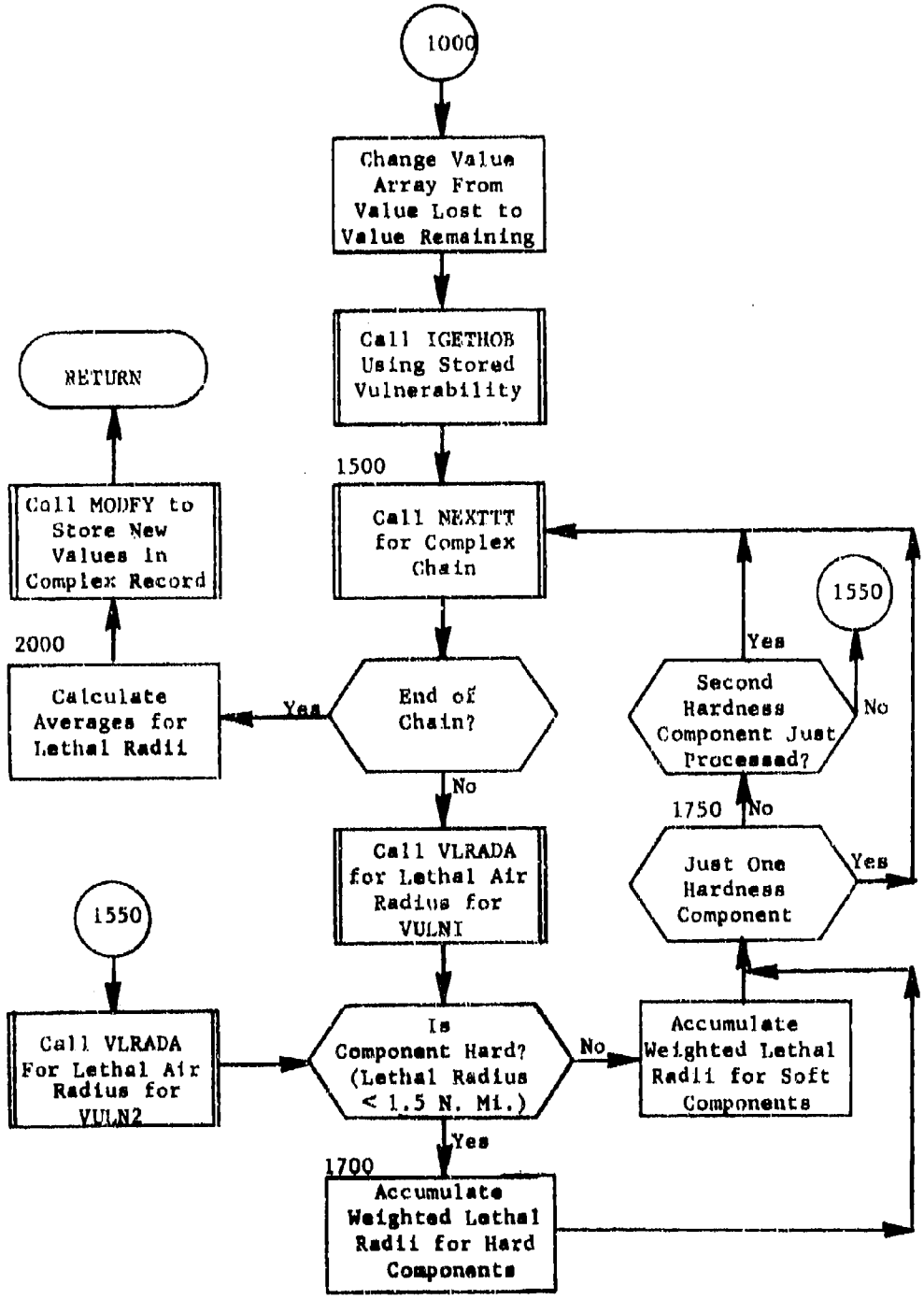


Figure 24. (Part 5 of 5)

5.10 Subroutine GRPEM

PURPOSE: To form weapon groups

ENTRY POINTS: GRPEM

FORMAL PARAMETERS: None

COMMON BLOCKS: C10, C15, C20, C25, C30, SET, WEAPON

SUBROUTINES CALLED: ADJUST, DIRECT, DLETE, HDFND, HEAD, ITLE, MODFY, NEXTTT, TANKER

CALLED BY: ENTMOD (of overlay link PLANS)

Method:

GRPEM forms weapon groups by chaining the user selected weapon type records ('WEPTYP') and for each type, the individual weapon launch bases are chained ('MSBMTIG'). Each base record then is tested for proper definition under weapon group heading.

Modules within the QUICK system requires weapon groups to be sorted in a definite order which can be easily achieved by properly chaining the weapon types. The order of chaining consists of first grouping all bases that are salvoed missiles, followed by non-salvoed missiles and then salvoed bombers are grouped followed by non-salvoed bombers. A salvoed weapon has attribute LCHINT greater than zero.

Tankers are not grouped but are collected and reformatted within subroutines TANKER and ADJUSTGP.

In the case of a missile weapon system, GRPEM checks the retargeting flag (RETARGET). If on, the user has requested that the data base attribute IREP be considered for all missiles. GRPEM then calculates and stores for the current missile type, the factors that later will be used to modify the number per squadron, number on alert, alert DBL probability, and reliability for all missiles of the type.

After weapon type data has been selected and defined, missiles and bombers are aggregated to form weapon groups. A weapon group consists of weapons from up to 150 bases. If all the weapons on a given base are nonalert, weapons of the same type are considered as one group. Otherwise, a group comprises those weapons on a base which have the same alert status, type (attribute TYPE), region, and payload. Bombers must also have the same refueling index. The maximum number of warheads allowed per group is set at 1,000. Also, for missile classes the maximum number of weapons per salvo is set at 15; if exceeded, a new missile group is formed.

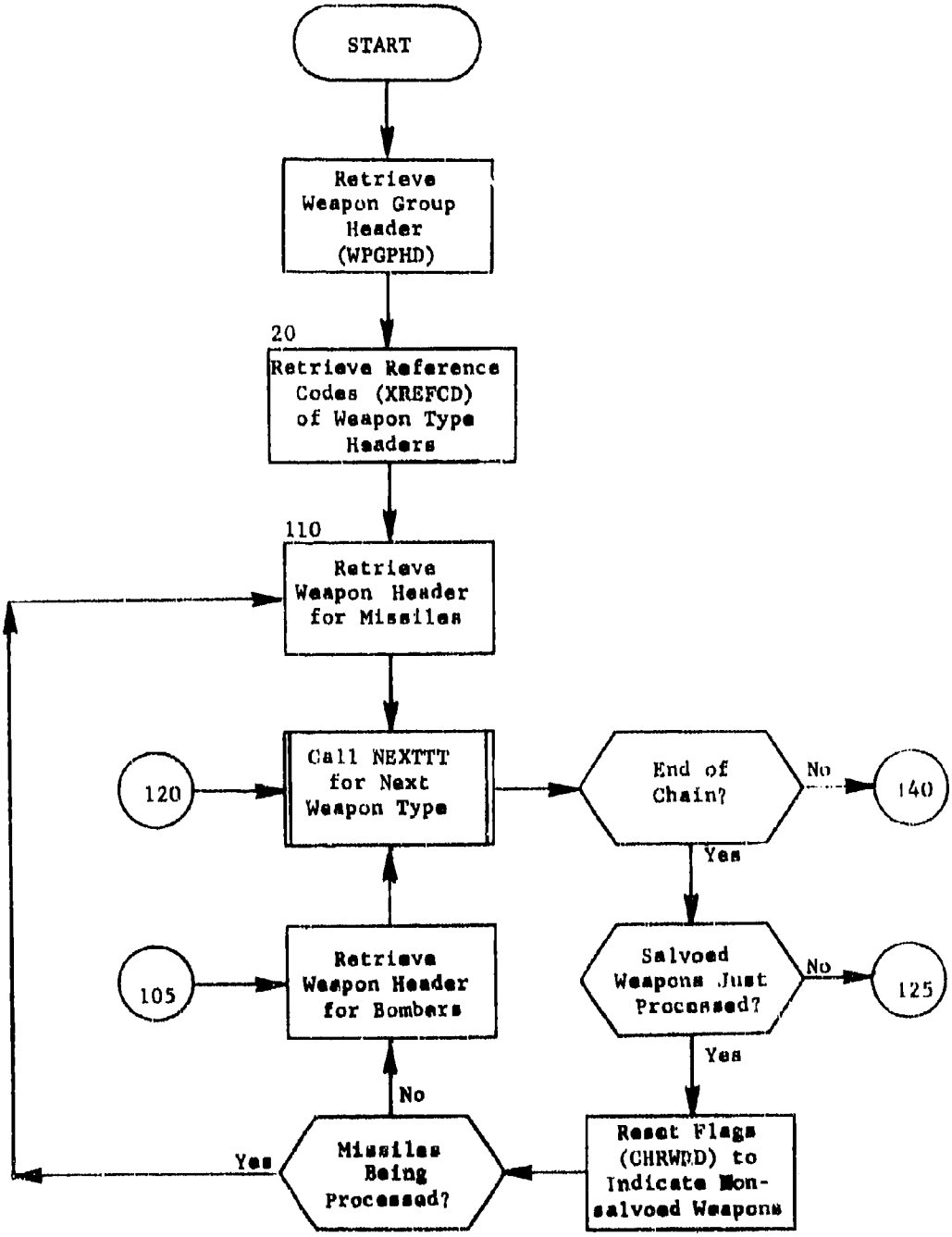
Only those records are processed that define the first site of a squadron (ISITE positive).

BOMBER units which do not refuel and missile sites must lie within a geographic region which, for alert weapons, has a radius equal to a certain percentage of the range of the weapon. This percentage is read into the variable RANGEMOD at the beginning of the program; if the percentage is not specified in the data cards, it is assumed to be 15%. For nonalert weapons, the distance criterion is automatically doubled.

In order to form a weapon group, the required radius is expressed in terms of latitude (DLAT) and longitude (DLONG), and the number of bases (NTOTBAS) is counted. If some bombers are to be used as tankers for refueling purposes (i.e., if IREFUEL=2), the number in commission and the number on alert are cut in half. The number of weapons and total yield of the warheads carried by each vehicle on the base then are computed. Up to 250 groups can be formed for use in plan generation. However, PLANSET processes and prints information for up to 260 weapon groups to enable planners to adjust their data base should more than 250 groups be formed.

When a new group is started group data are retrieved and stored under record 'WEPNGP'. For each weapon launch base, the base record ('MSBMTC') is modified and linked to the group header. As each new base is added, the group centroid is adjusted accordingly. If there are both alert and nonalert bombers on a given base, the alert bombers are tested for group assignment first using the distance criterion RANGEMOD; the nonalert bombers then are tested using the criterion 2 x RANGEMOD.

Subroutine GRPEM is illustrated in figure 25.


START
Retrieve Weapon Group Header (WPGPHD)
20
Retrieve Reference Codes (XREFCD) of Weapon Type Headers
110
Retrieve Weapon Header for Missiles
120
Call NEXTTT for Next Weapon Type
End of Chain?
No
140
Yes
105
Retrieve Weapon Header for Bombers
Salvoed Weapons Just Processed?
No
125
Yes
No
Yes
Missiles Being Processed?
Reset Flags (CHRWPD) to Indicate Non-salvoed Weapons

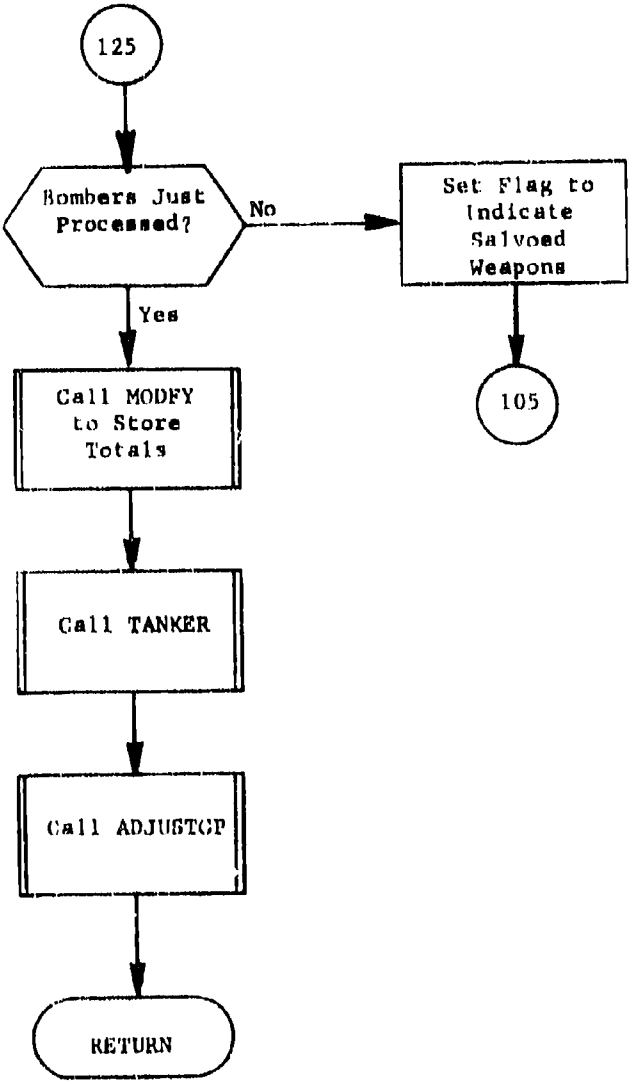
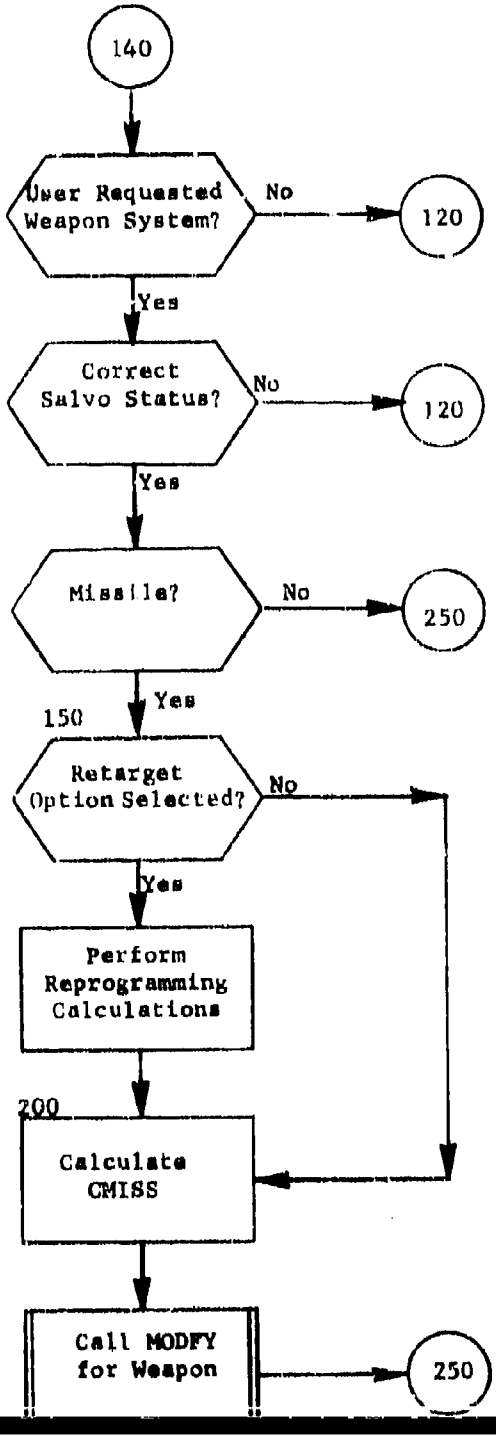


Figure 25. (Part 2 of 9)


140
User Requested Weapon System?
No
120
Yes
Correct Salvo Status?
No
120
Yes
Missile?
No
250
Yes
150
Retarget Option Selected?
No
Yes
Perform Reprogramming Calculations
200
Calculate CMISS
Call MODFY for Weapon
250

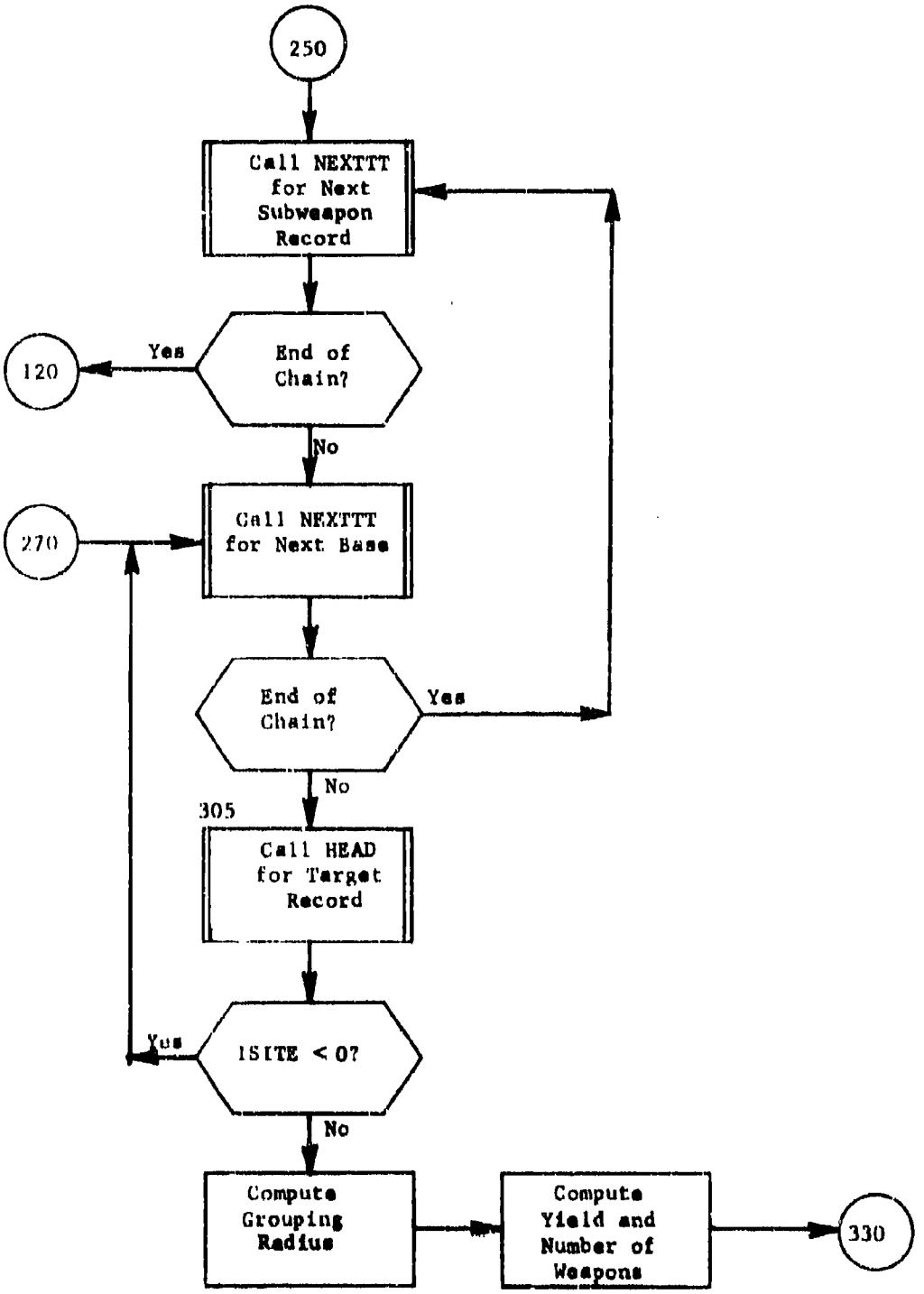
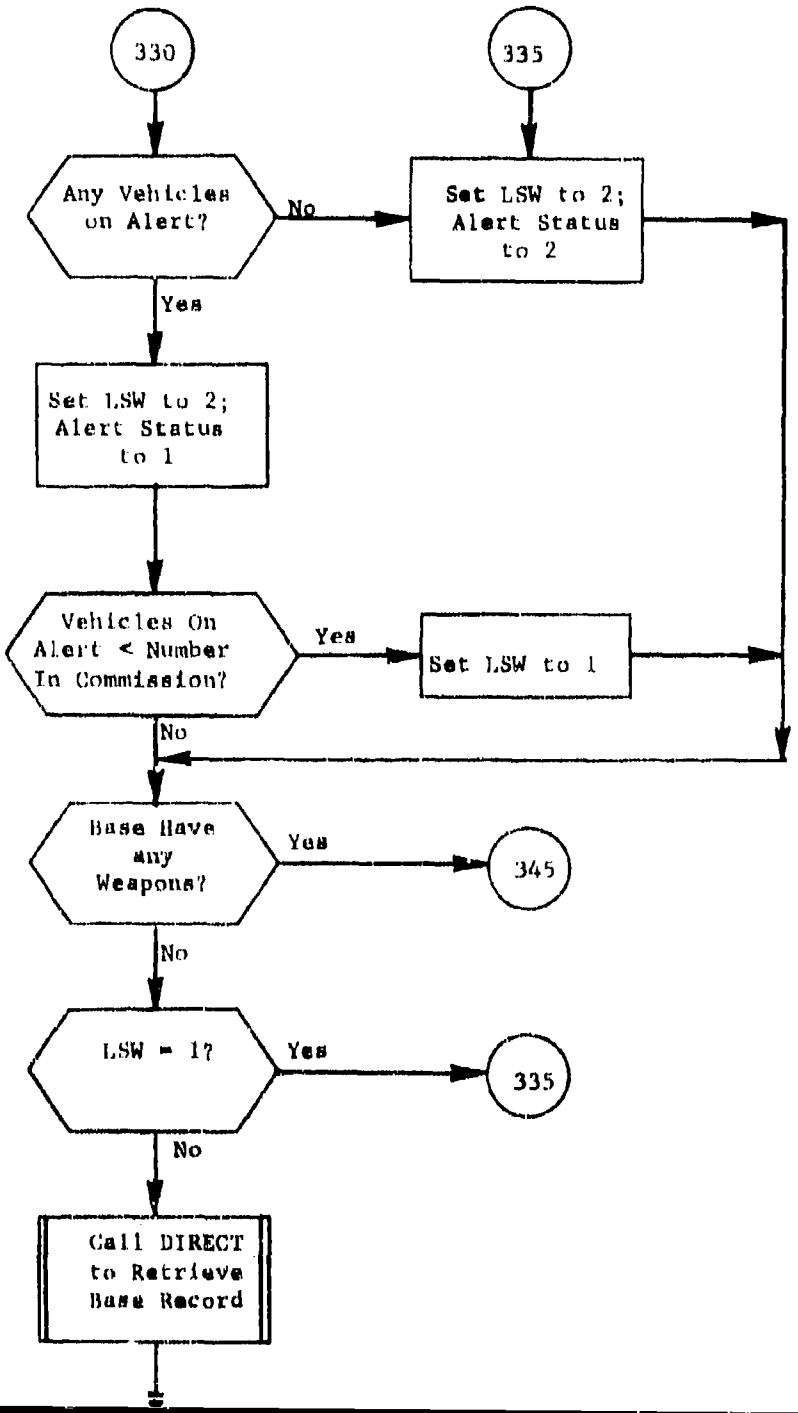
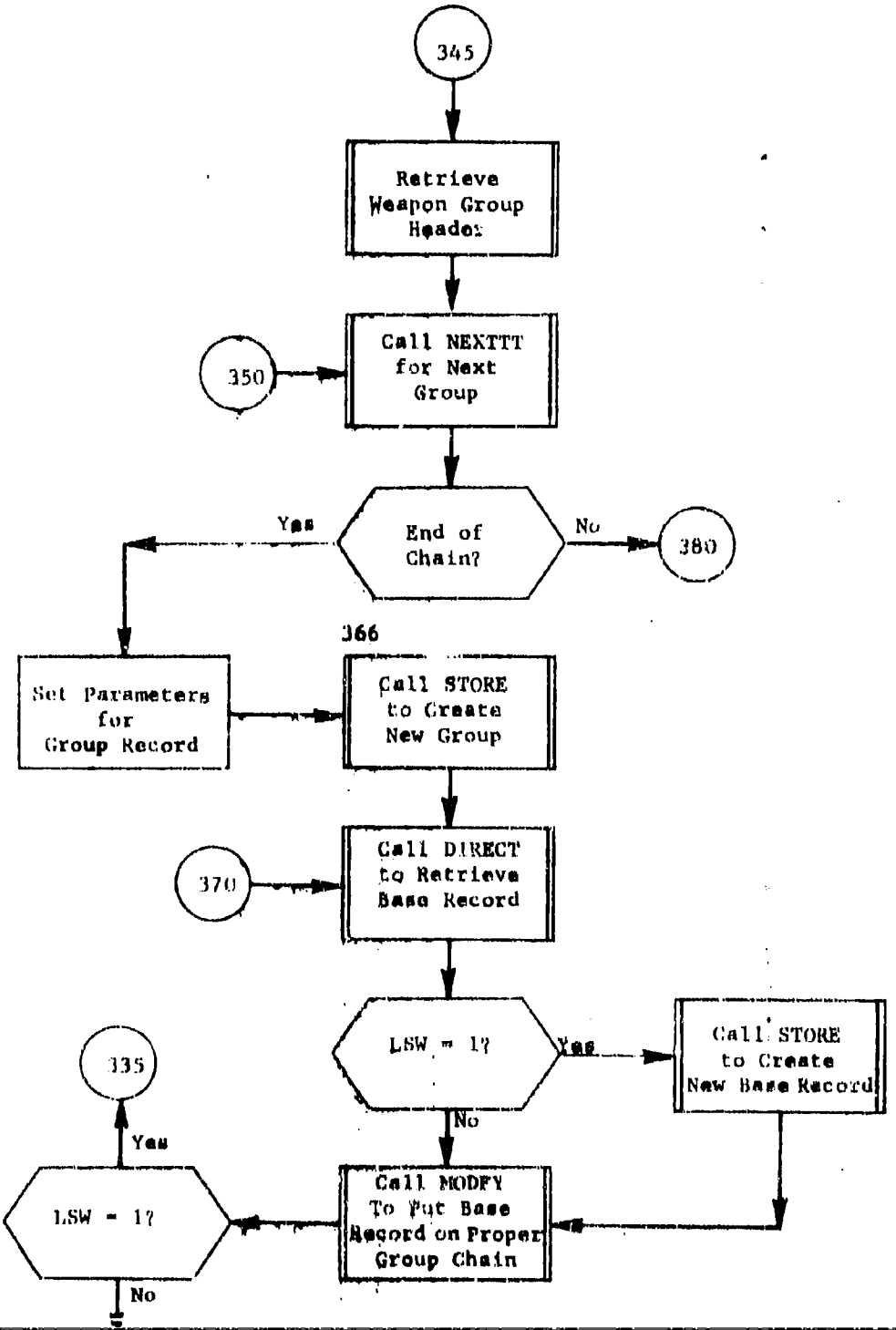


Figure 27. (Part 4 of 8)


330
335
Any Vehicles on Alert?
No
Set LSW to 2; Alert Status to 2
Yes
Set LSW to 2; Alert Status to 1
Vehicles On Alert < Number In Commission?
Yes
Set LSW to 1
No
Base Have any Weapons?
Yes
345
No
LSW = 1?
Yes
335
No
Call DIRECT to Retrieve Base Record


345
Retrieve Weapon Group Header
350
Call NEXTTT for Next Group
End of Chain?
Yes
No
380
366
Set Parameters for Group Record
Call STORE to Create New Group
370
Call DIRECT to Retrieve Base Record
LSW = 1?
Yes
Call STORE to Create New Base Record
No
335
Yes
LSW = 1?
Call MODFY To Put Base Record on Proper Group Chain
No

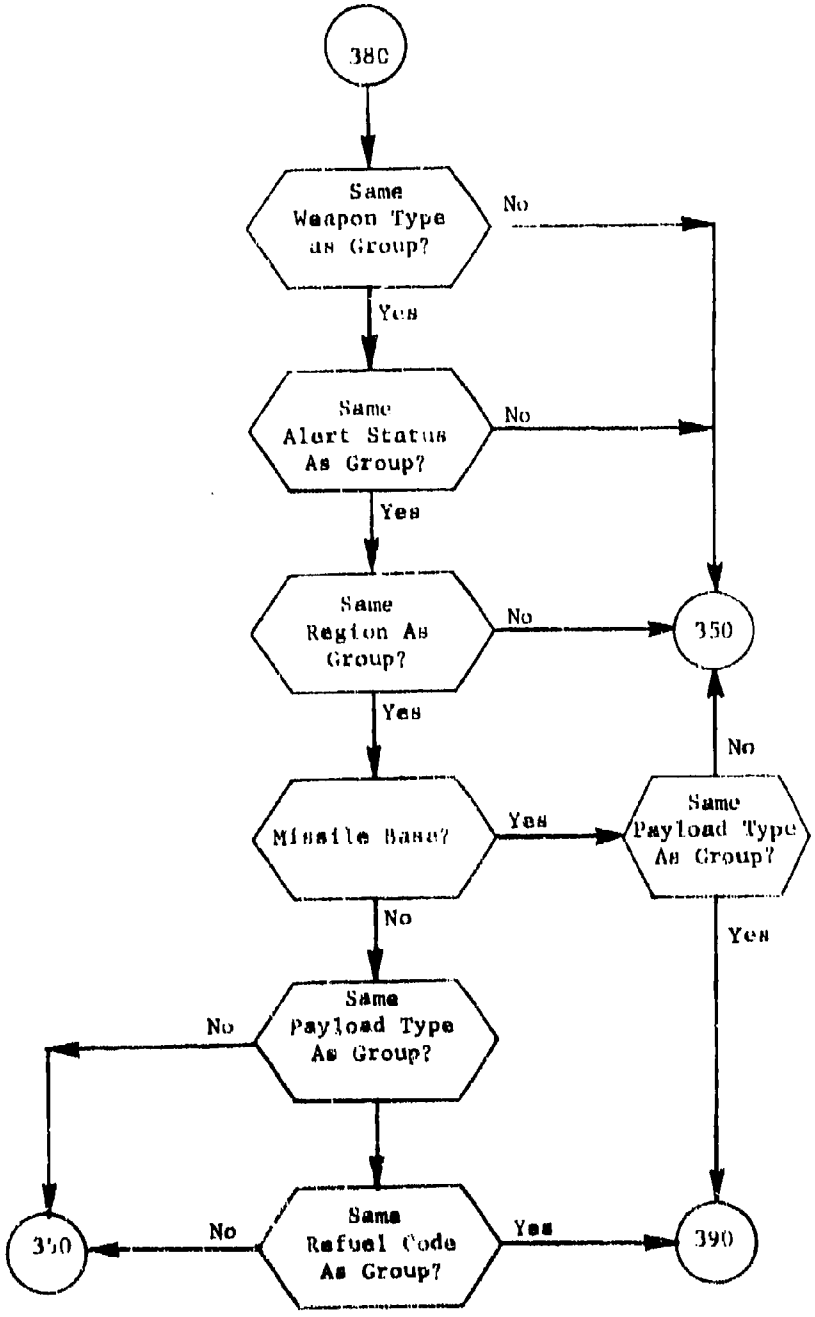


Figure 25. (Part 7 of 9)

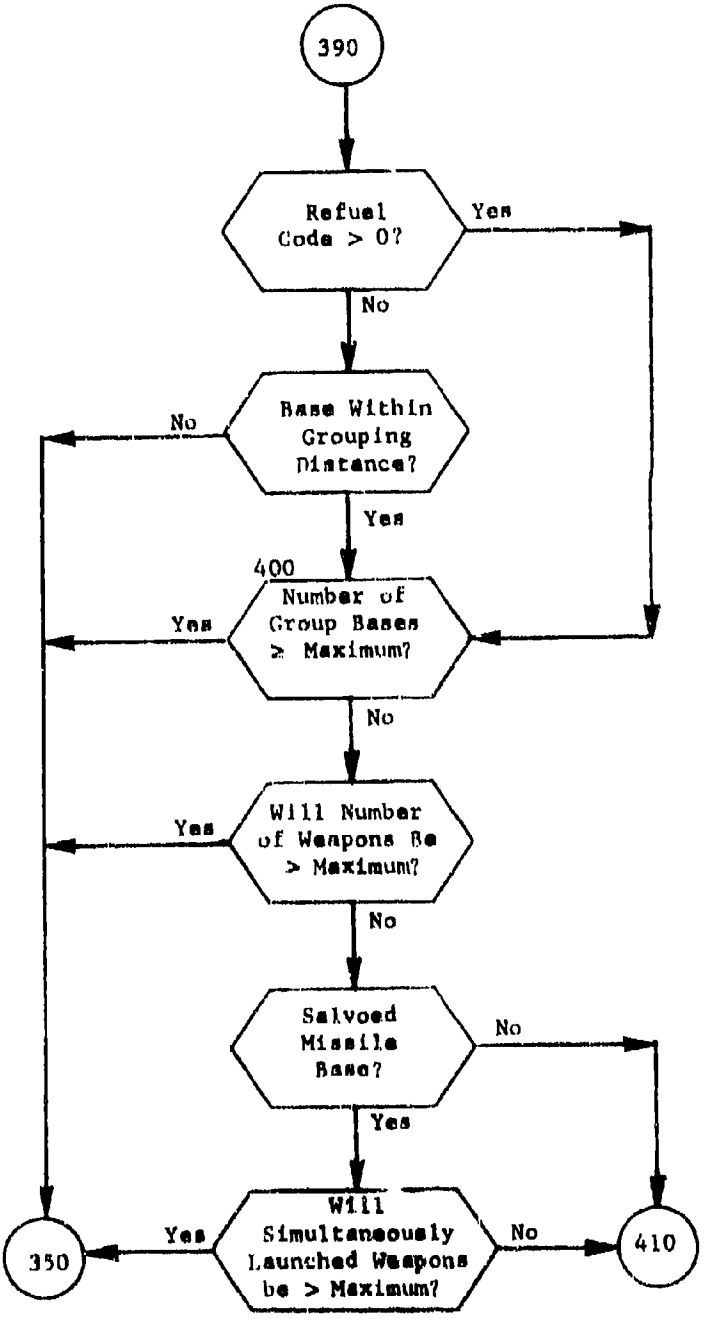


Figure 25. (Part 8 of 9)

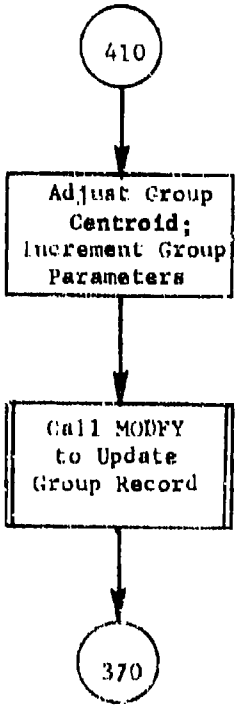


Figure 25. (Part 9 of 9)

5.11 Subroutine PRINTGP

PURPOSE: Print standard tables

ENTRY POINTS: PRINTGP

FORMAL PARAMETERS: None

COMMON BLOCKS: C10, C15, C30, EXCLAS, PBLOK, WEAPON

SUBROUTINES CALLED: DIRECT, HDFND, HEAD, ITLE, MODFY, NEXTTT, RETRV

CALLED BY: ENTMOD (of overlay link PLANS)

Method:

PRINTGPs sole purpose is to print those tables as given in figure 26.

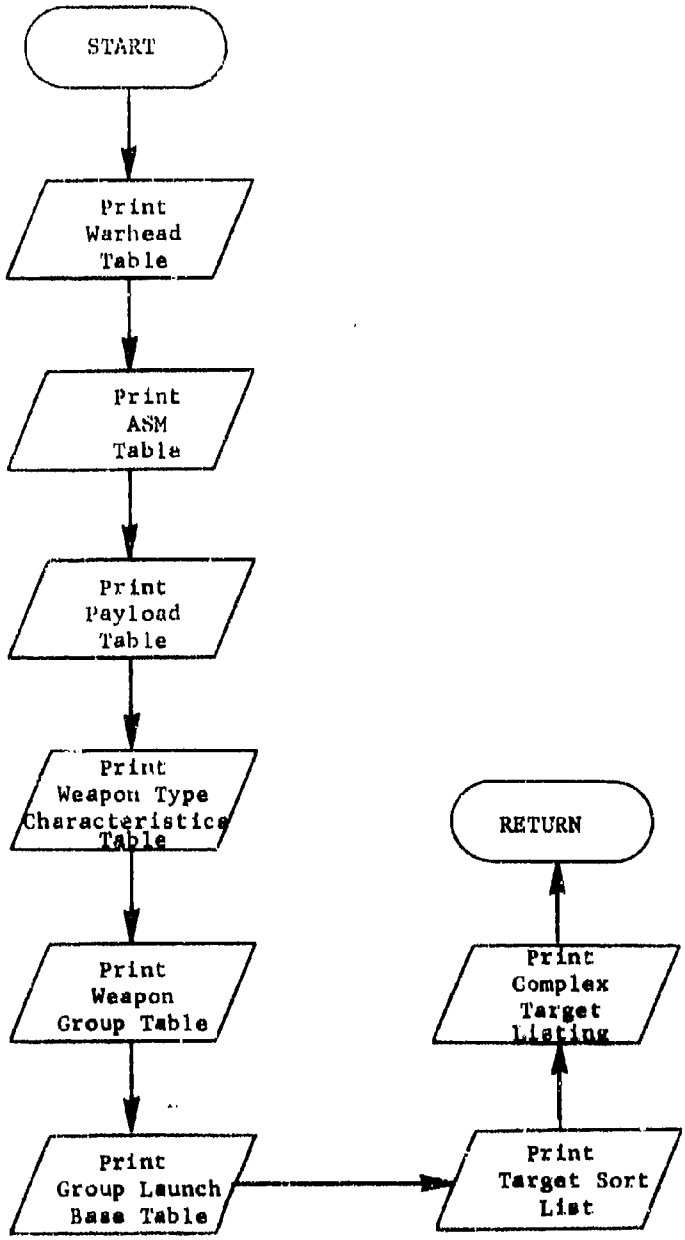


Figure 26. Subroutine PRINTGP

5.12 Subroutine SRTTGT

PURPOSE:	To process user selected targets
ENTRY POINTS:	SRTTGT
FORMAL PARAMETERS:	None
COMMON BLOCKS:	ERRCOM, CPRIOR, C10, C15, C20, C25, C30, EXCLAS, MASK, OOPS, PBLOK, TARCLAS
SUBROUTINES CALLED:	CALCOMP, DIRECT, DLETE, HDFND, HEAD, ITLE, MODFY, NEXTTT, RETRV, SORTIT, STORE, VLRADP
CALLED BY:	ENTMOD (of overlay PLANS)

Method:

SRTTGT begins by counting the number of characters in the task inputs (used for choosing representative targets of complexes in CALCOMP) and setting ISUBT to zero or one if the task inputs are one or two characters long, respectively.

Next the exemplar target DESIGs and corresponding values are processed. For each target class to be considered by the allocator, a DESIG and a corresponding value are entered. The DESIG pertains to a target within the target class and the value is its target value before the sum of target values are normalized to 1000. All targets within that class will have their data base values adjusted by a similar ratio before normalization. The exemplar targets are retrieved directly on the CALC chain.

Two passes are made through the targets in SRTTGT. During the first pass target values are accumulated for each target class, multiple targets are formed and the targets which will be considered by the allocator are counted. Also during this pass for each target two separate words are written onto the scratch file ISORTLUN for sorting by subroutine SORTIT. The first word contains attribute FLAG in the first character and DESIG in the remaining characters and is used for the FLAG-DESIG listing.

The second word, containing a blank in the first character and DESIG in the remaining characters, is used for the Target Designator-Number Directory print.

Before the second pass, scaling factors which will adjust relative target values and normalize the sum of target values are computed. Also before this pass parameters for assigning randomized target numbers are computed.

During the second pass the value of each target is normalized by multiplying its data base value by the scaling factor for its target class. Also a randomized target number is calculated for each complex target, multiple target, and simple target not in a complex or multiple target. A TARCDE record is stored for each of these targets. This record contains the target number of the target as well as the reference code of the representative target if the target is complex or multiple or if the target itself is a simple target.

Upon processing each target record, the target number can be immediately calculated and stored. The calculation is: a sorting index (LEAD), which is a function of the total number of targets (NTAR), is determined by the formula:

$$LEAD = \frac{NTAR\ (3 - \sqrt{5}}{2}$$

To start a cycle, a beginning index (IBEG) is designated and assigned to the target being read. Initially IBEG=LEAD. The index for the next target (IND) then is found by incrementing the previous index (LAST) by LEAD. If the result exceeds NTAR, the cycle is reset by subtracting NTAR from IND. When a cycle is completed (i.e., when IND=IBEG), the next cycle is begun by incrementing IBEG by one and proceeding as above. Thus, a unique nonsequential index is assigned to each target as it is read.

Also, during the record pass through the targets, elements of complexes not belonging to target classes chosen to be considered by the allocator are transferred from the complex chains they are on to the simple target chain. When all such targets are transfered if a complex target has just one element remaining, this element is also transfered to the simple target chain to be considered as a single target. For the remaining complex targets, subroutine CALCOMP is called.

After the second pass complex records to complex targets which do not have any elements belonging to target classes to be considered by the allocator are deleted. Also SORTIT is called and the FLAG-DESIG Listing and Target Designator-Number Directory are printed.

Subroutine SRTTG is illustrated in figure 27.

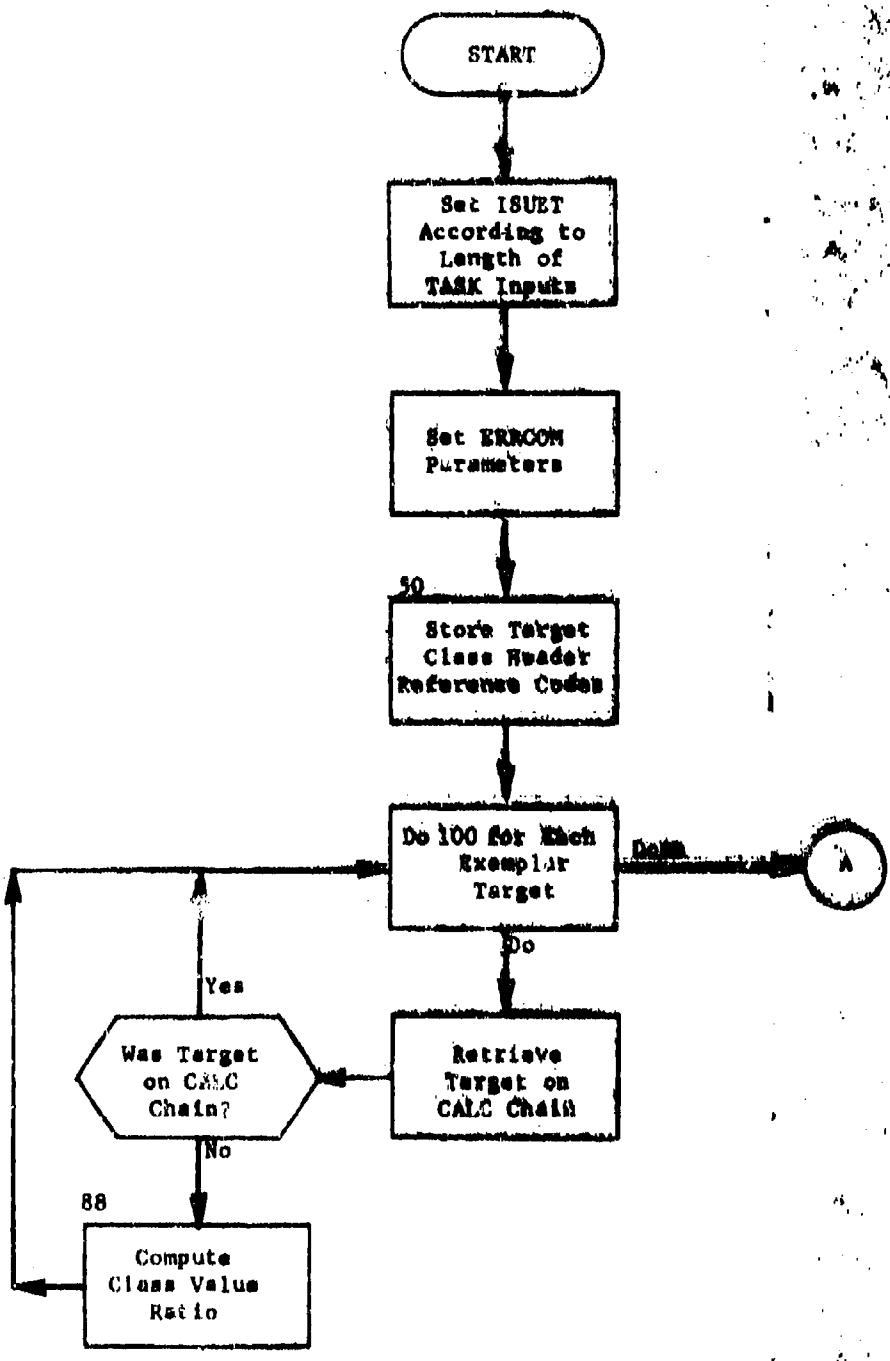


Figure 27. Subroutine SRTTGT (Part 1 of 11)

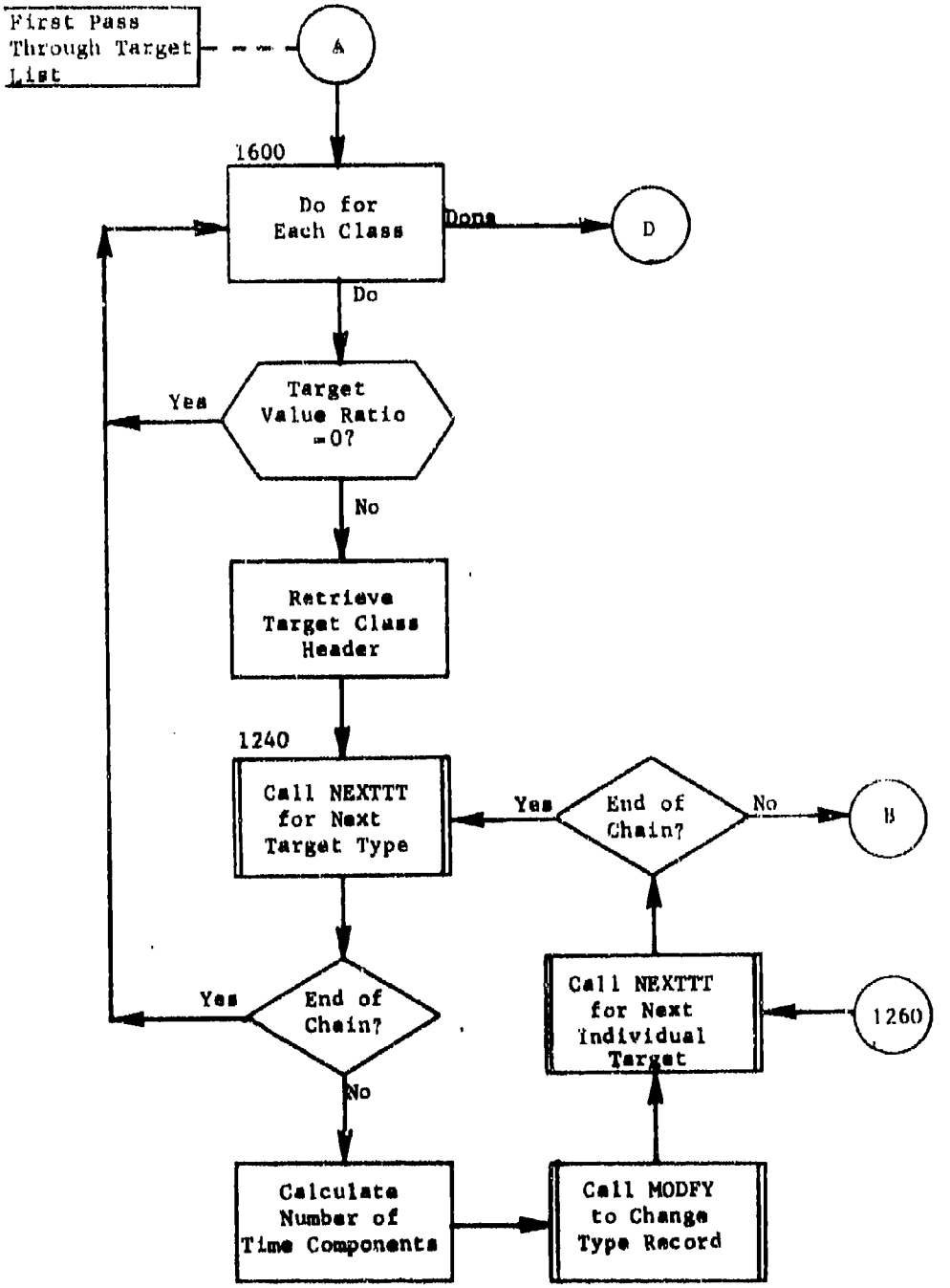


Figure 27. (Part 2 of 11)

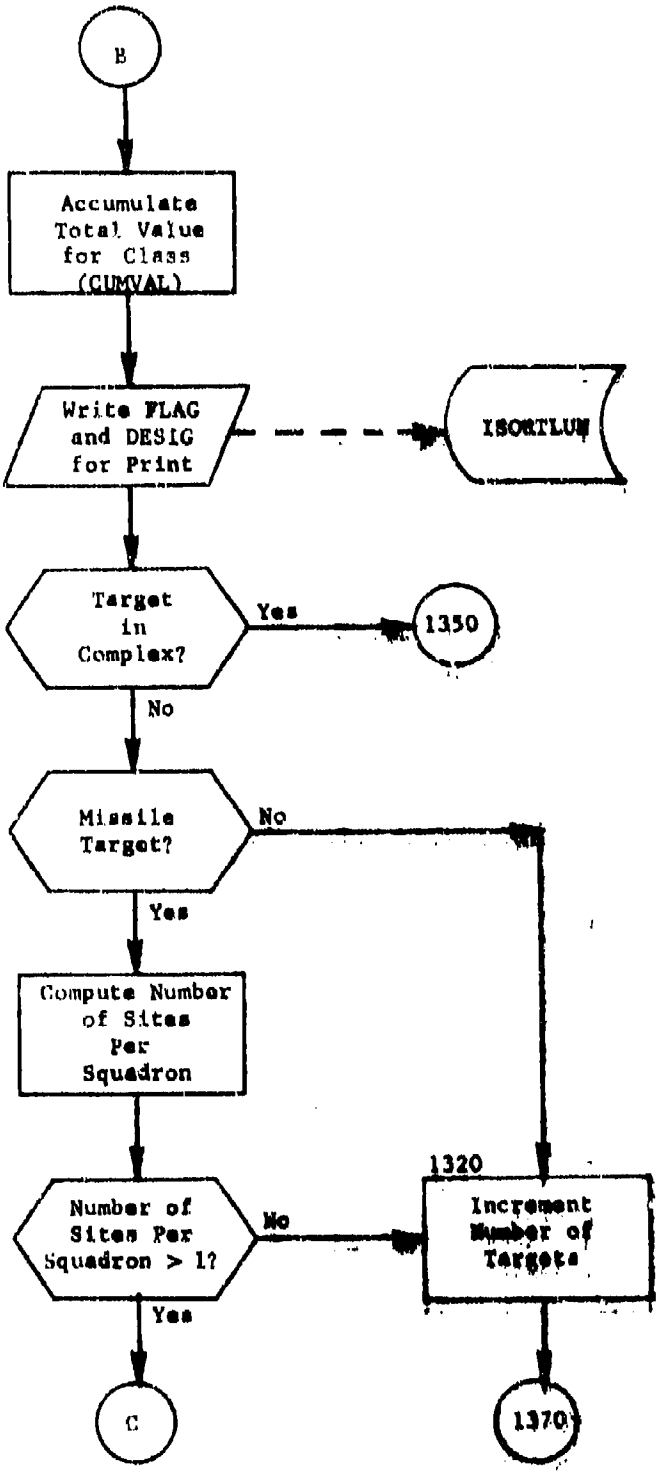


Figure 27. (Part 3 of 11)

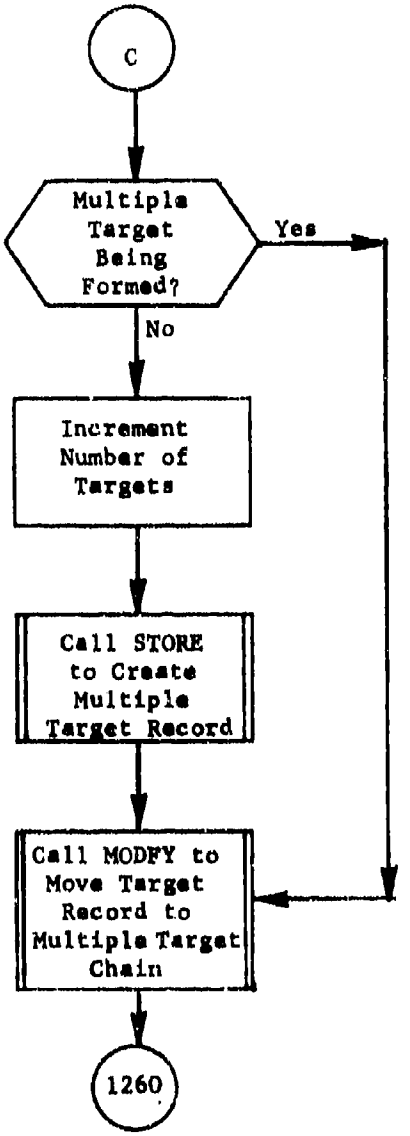


Figure 27. (Part 4 of 11)

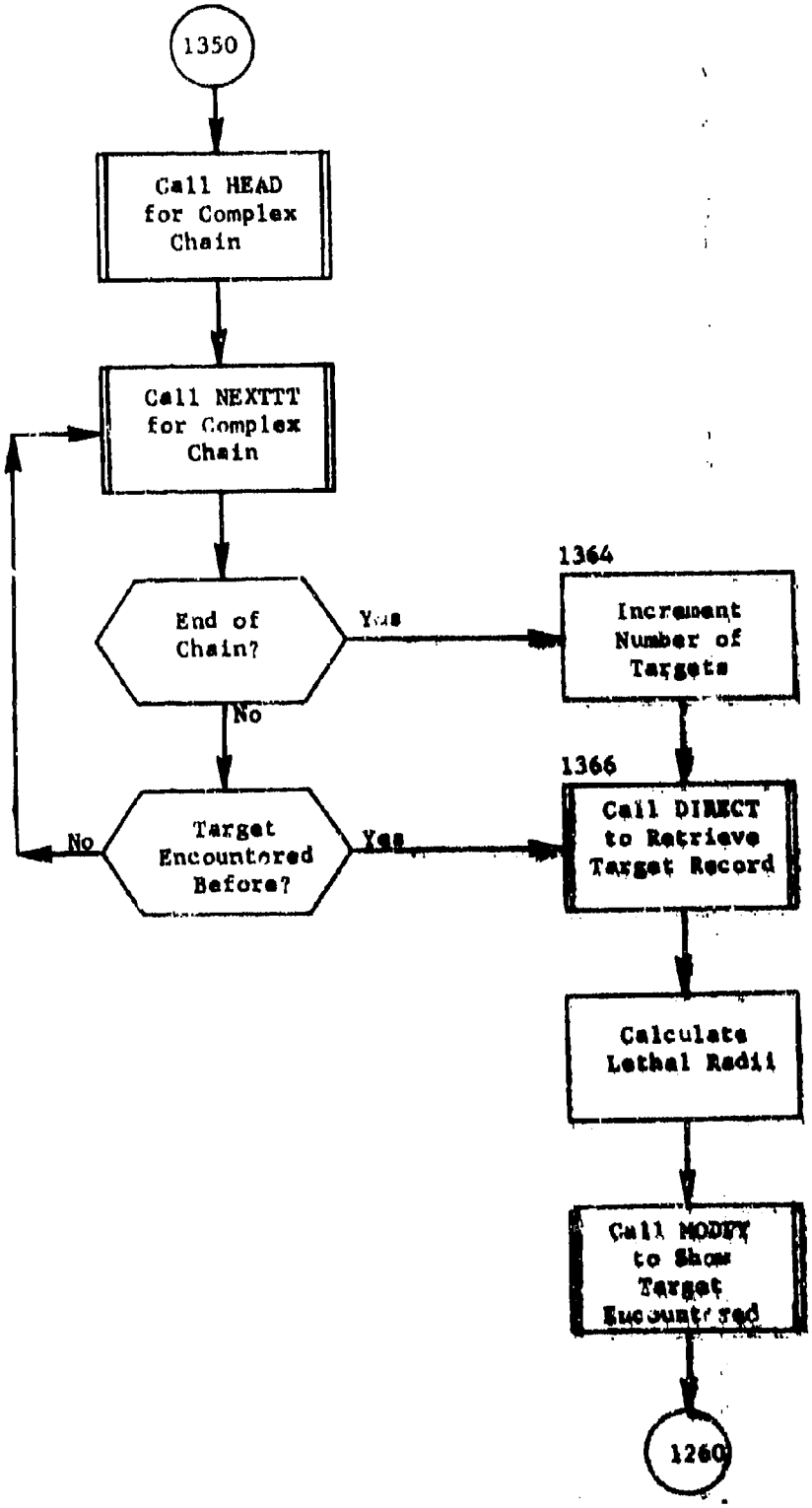


Figure 27. (Part 5 of 11)

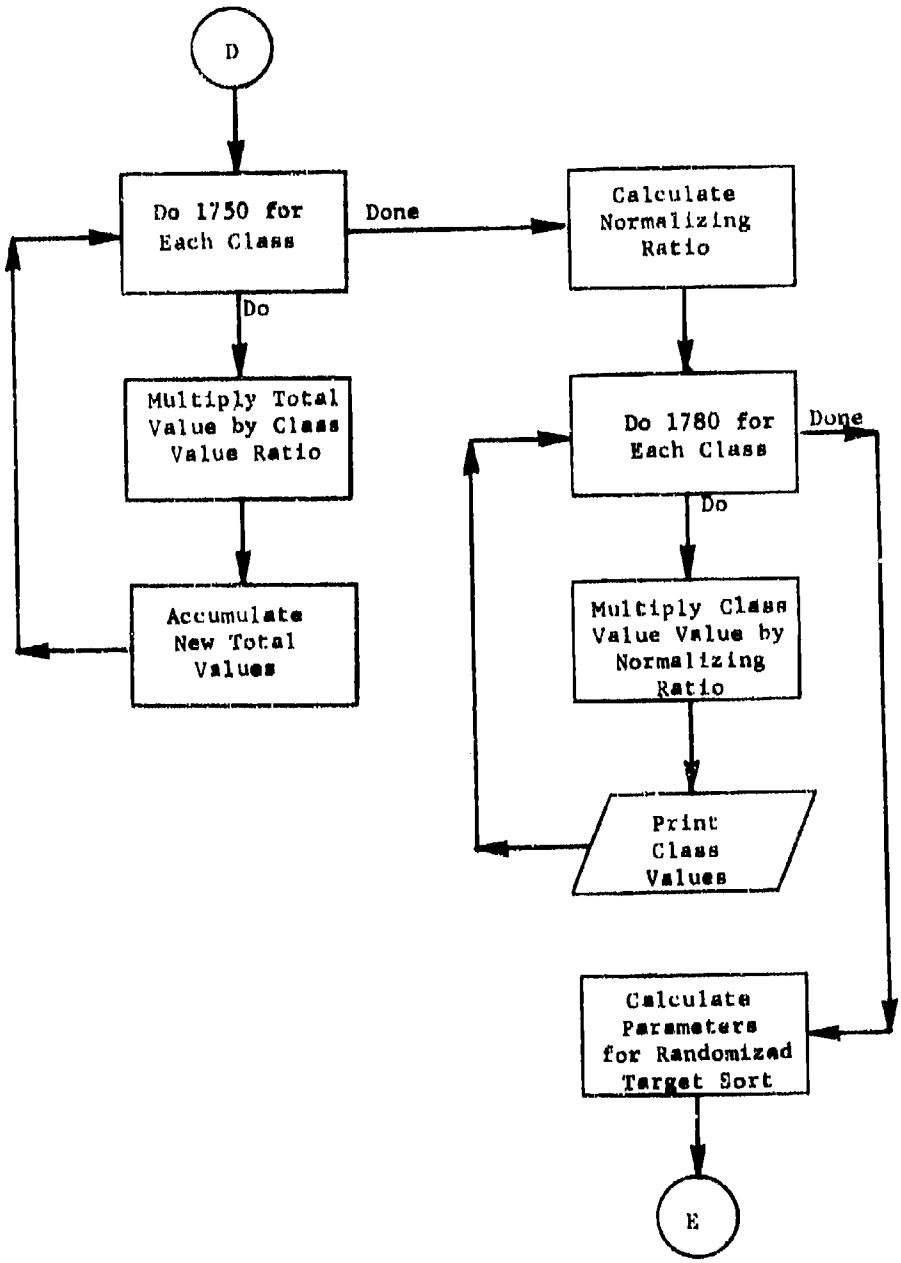


Figure 27. (Part 6 of 11)

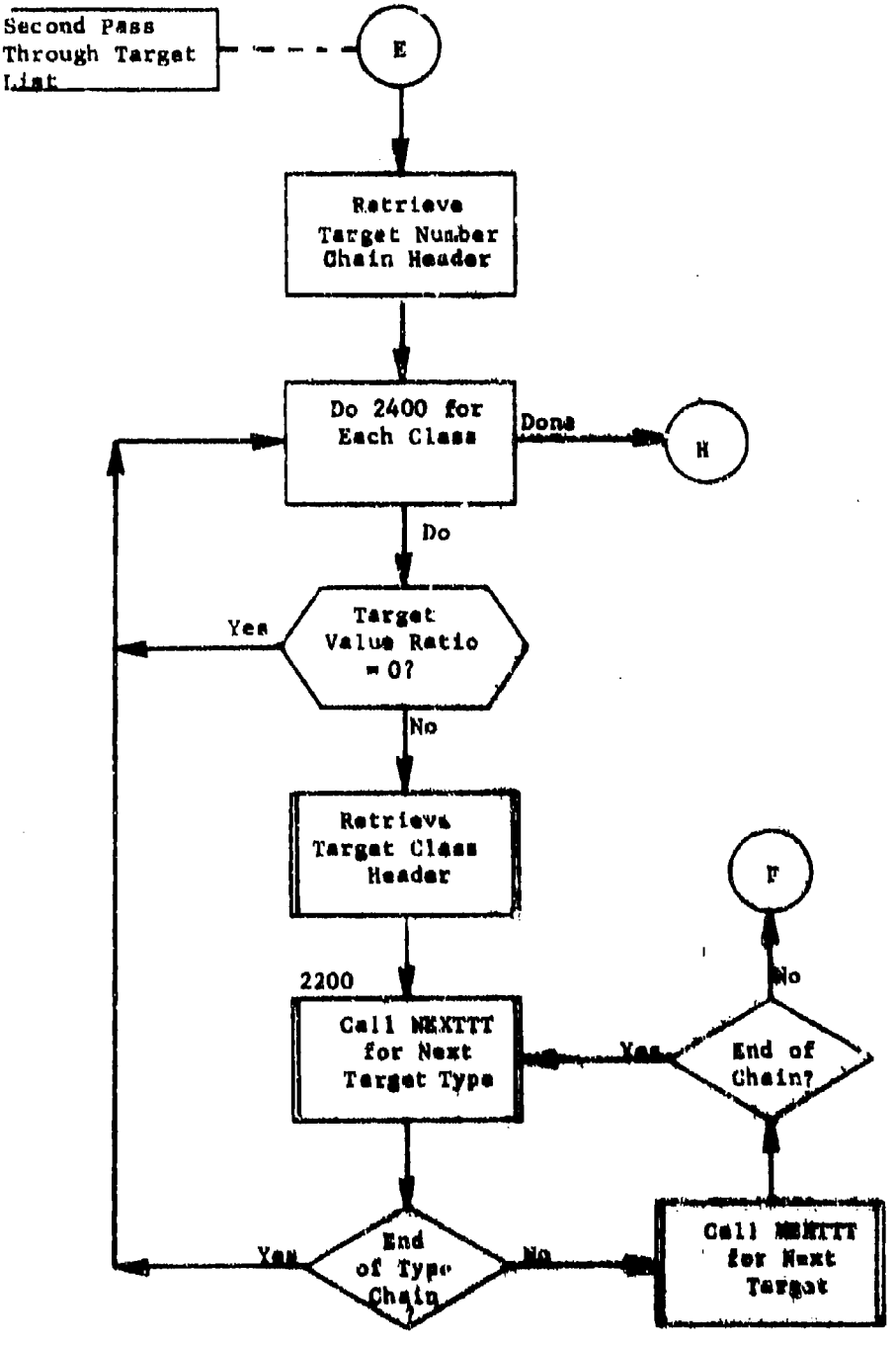


Figure 27. (Part 7 of 11)

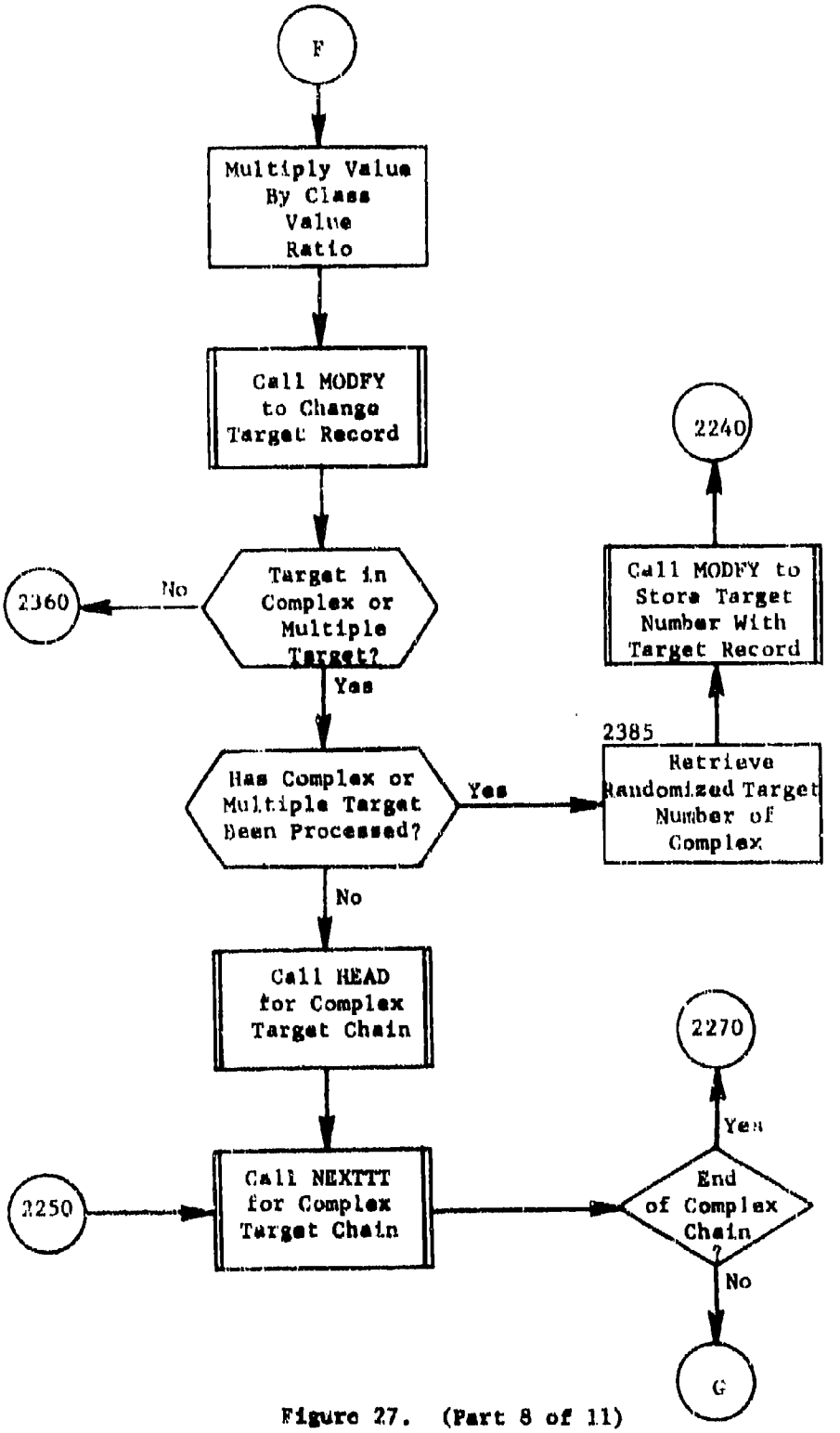


Figure 27. (Part 8 of 11)

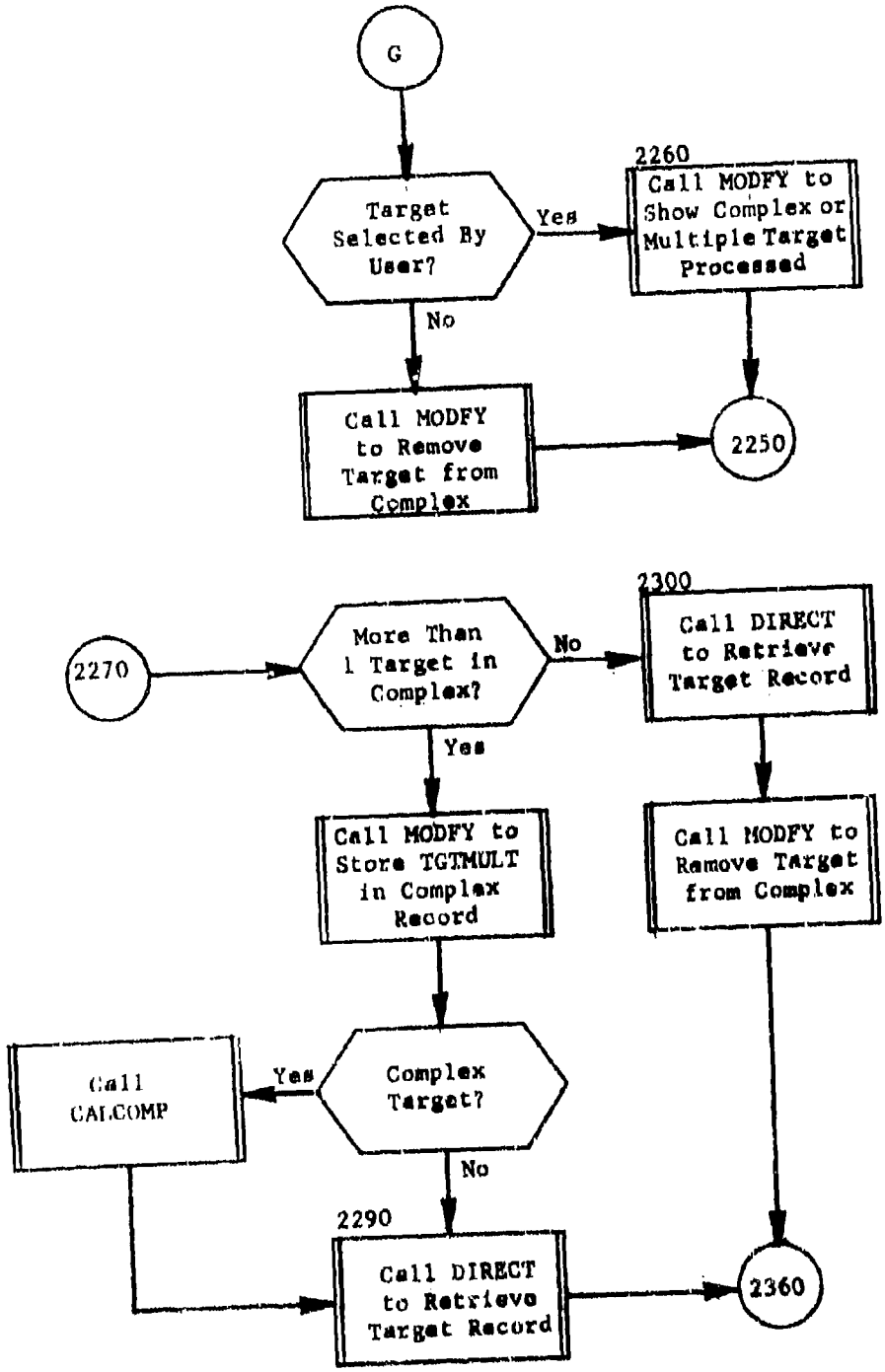


Figure 27. (Part 9 of 11)

Figure 27. (Part 10 of 11)

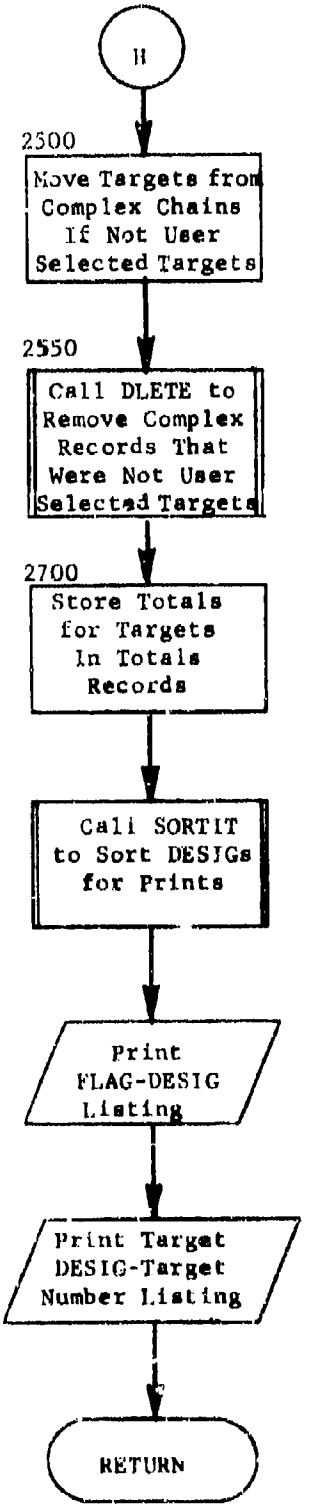


Figure 27. (Part 11 of 11)

5.13 Subroutine TANKER

PURPOSE: Count tanker records

ENTRY POINTS: TANKER

FORMAL PARAMETERS: None

COMMON BLOCKS: C10, C15, C30, WEAPON

SUBROUTINES CALLED: HDFND, HEAD, ITLE, MODFY, NEXTTT, RETRV

CALLED BY: GRPEM

Method:

Tanker bases are not included in the group assignment. For each tanker base, TANKER counts the number of data base tankers (NTANK) and the number of tanker bases (NTANKERB).

For each tanker record that has a value of IREFUEL less than 4, IREFUEL will be reset to zero.

Figure 28 illustrates TANKER.

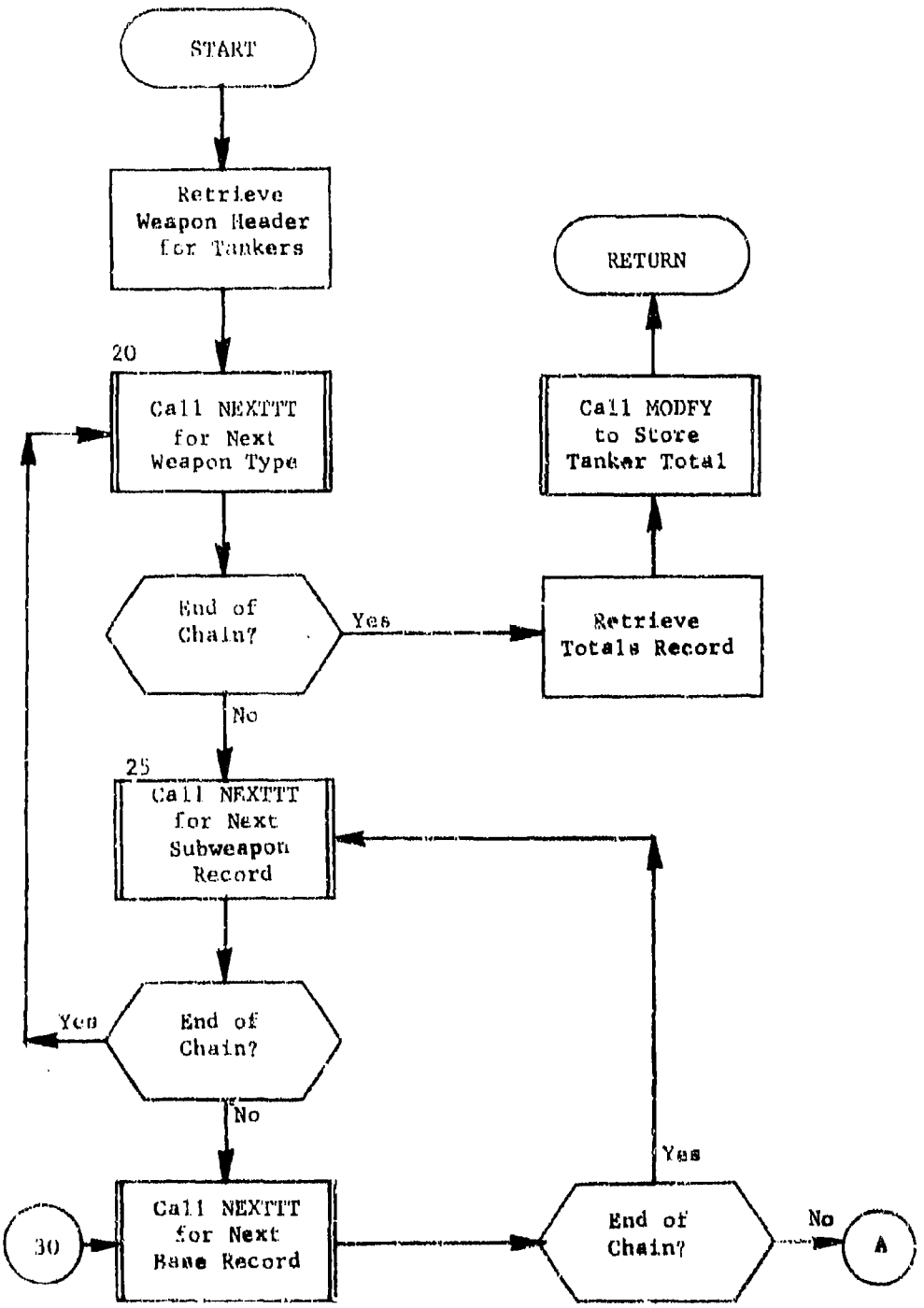


Figure 28. Subroutine TANKER (Part 1 of 2)

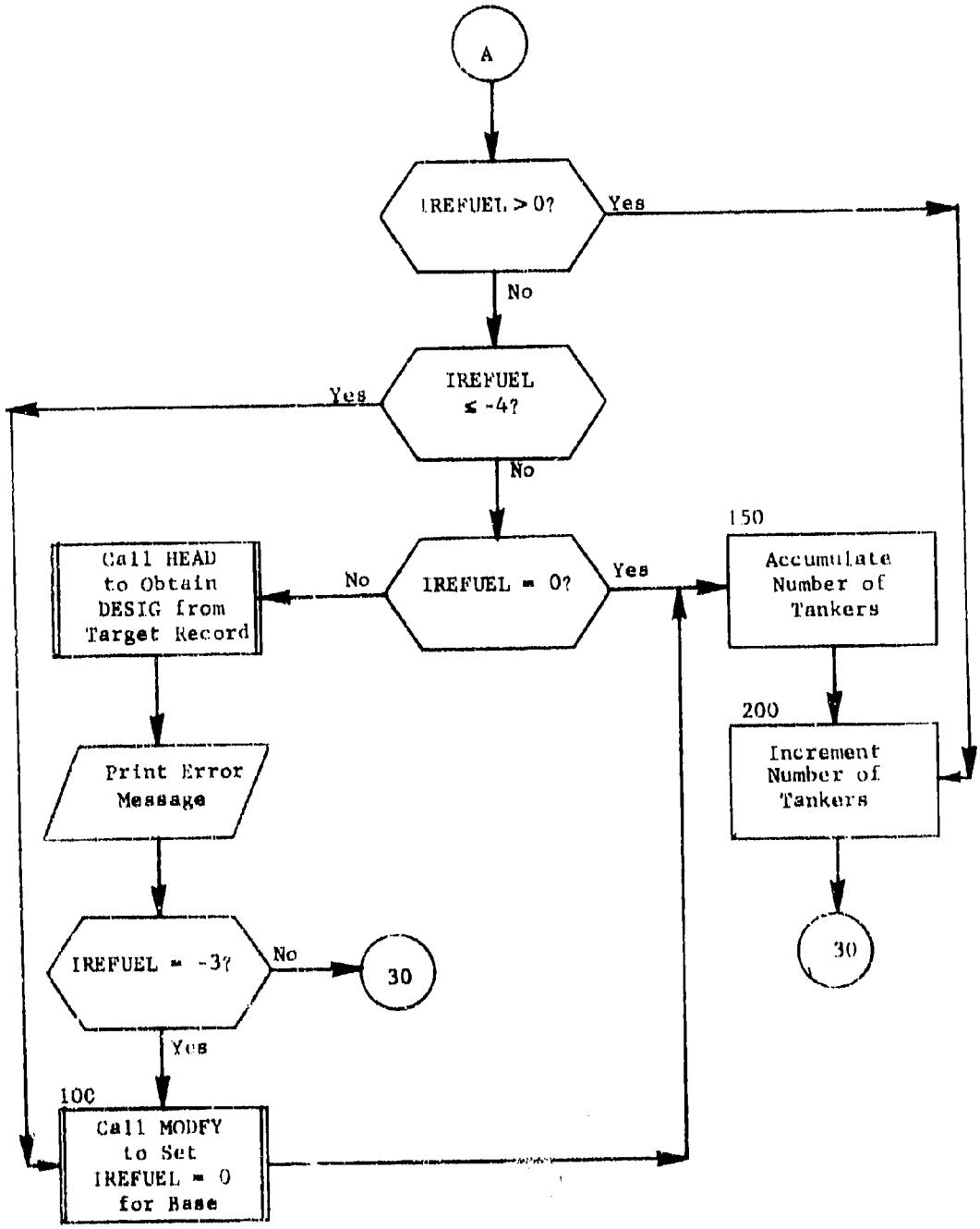


Figure 28. (Part 2 of 2)

5.14 Function VLRADP

PURPOSE: 1. Find lethal radius of weapon
2. Set FN for use by calling subroutine

ENTRY POINTS: VLRADA, VLRADP

FORMAL PARAMETERS:
YIELD - Yield of weapon in megatons
NVN - Vulnerability parameter of target
HOB - Weapon height of burst
FN - Parameter specifying shape of damage function

COMMON BLOCKS: DPOOL, FLSTCL

SUBROUTINES CALLED: None

CALLED BY: CALCOMP, SRTTG1

Method:

Entry VLRADA is executed if called from subroutine CALCOMP (local parameter IN is set to nonzero); else VLRADP is executed (local parameter is set to zero). VLRADP entry implies air burst lethal radius is to be calculated purely on target vulnerability; VLRADA entry implies air burst lethal radius is to be calculated based on target vulnerability and scaled height of burst. Ground burst lethal will be calculated the same for both entry points.

NVN is decoded into the appropriate vulnerability number VN, the latter (P or Q), and K-factor XK. The cube root of the yield is extracted. Then the adjusted vulnerability number AVN is determined by methods described in "Computer Computation of Weapon Radius," B-139-61, Air Force Intelligence Center. FN is set to six or three for P and Q type targets, respectively.

The natural logarithm of the lethal radius (in nautical miles) of a 1-megaton burst is contained in arrays PG, QG, QA, PA, QQA, and PPA. Function VLRADP interpolates in the appropriate array to find the logarithm of the 1-megaton lethal radius for AVN. Arrays PG, QG, QA, and PA are at intervals of five vulnerability numbers. The first index of arrays QQA and PPA are also at intervals of hundreds of feet for an air burst. The lethal radius of the weapon is then determined by exponentiating and multiplying by the cube root of the yield.

A flowchart for VLRADP is shown in figure 29.

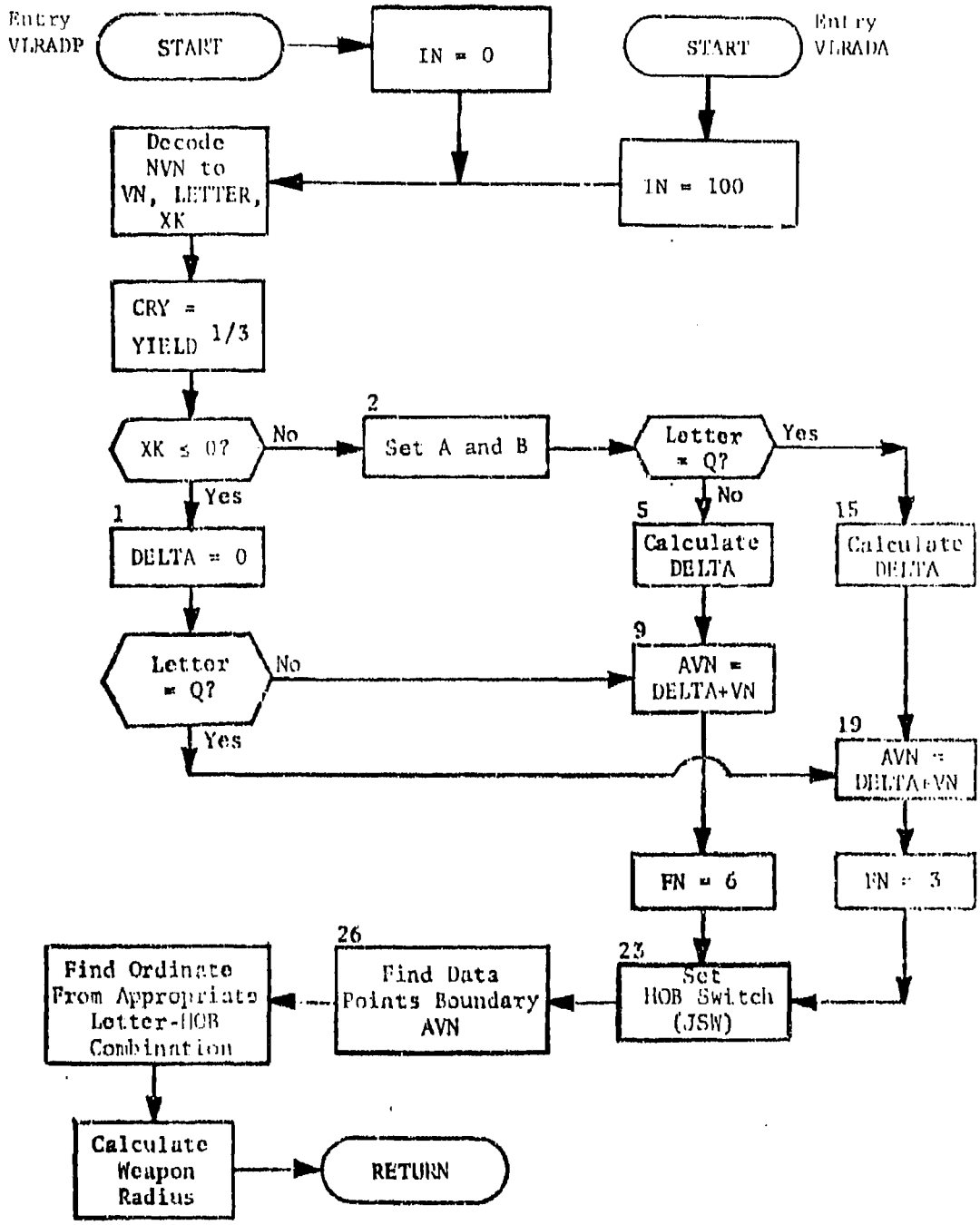


Figure 29. Function VLRADP

DISTRIBUTION

Addressee	Copies
CCTC Codes	
Technical Library (C124)	3
C124 (Stock)	6
C126	1
C313	1
C314	11
C600	1
C730	1
DCA Code	
205	1
EXTERNAL	
Chief, Studies, Analysis and Gaming Agency, OJCS ATTN: SFD, Room 1D957, Pentagon, Washington, DC 20301	5
Chief of Naval Operations, ATTN: OP-96C4, Room 4A478, Pentagon, Washington, DC 20350	2
Commander-in-Chief, North American Air Defense Command ATTN: NPXYA, Ent Air Force Base, CO 80912	2
Commander, U. S. Air Force Weapons Laboratory (AFSC) ATTN: AFWL/SAB, Kirtland Air Force Base, NM 87117	1
Commander, U. S. Air Force Weapons Laboratory (AFSC) ATTN: AFWL/SUL (Technical Library), Kirtland Air Force Base, NM 87117	1
Director, Strategic Target Planning, ATTN: (JPS), Offutt Air Force Base, NE 68113	2
Defense Documentation Center, Cameron Station, Alexandria, VA 22314	12
	50

UNCLASSIFIED (18) CCTC, SBIE

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

(19) REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER: CSM-MM-9-77-Vol II (2) AD-E105 023 (3) 2. GOVT ACCESSION NO. (4) Computer system manual, 3. RECIPIENT'S CATALOG NUMBER

4. TITLE (and Subtitle): (6) The CCTC QUICK-REACTING GENERAL WAR GAMING SYSTEM (QUICK), Program Maintenance Manual, Weapon/Target Identification Subsystem (U) (8) — 5. TYPE OF REPORT & PERIOD COVERED: Volume II.

6. PERFORMING ORG. REPORT NUMBER

7. AUTHOR(s): (10) Dale J. Sanders, Paul F. M. Maykrantz, Jim M. Herrin, Edward F. Bersson

8. CONTRACT OR GRANT NUMBER(s): (15) DCA 100-75-C-0019

9. PERFORMING ORGANIZATION NAME AND ADDRESS: System Sciences, Incorporated, 4720 Montgomery Lane, Bethesda, Maryland 20014

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS

11. CONTROLLING OFFICE NAME AND ADDRESS: Command and Control Technical Center, Room BE-685, The Pentagon, Washington, DC 20301

12. REPORT DATE: 1 June 1977

13. NUMBER OF PAGES: 188 pp.

14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)

15. SECURITY CLASS. (of this report): UNCLASSIFIED

15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

DISTRIBUTION STATEMENT A
Approved for public release;
Distribution Unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

Approved for public release; distribution unlimited.

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

War Gaming, Resource Allocation

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

The computerized Quick-Reacting General War Gaming System (QUICK) will accept input data, automatically generate global strategic nuclear war plans, provide statistical output summaries, and produce input tapes to simulator subsystems external to QUICK.

The Program Maintenance Manual consists of four volumes which facilitate maintenance of the war gaming system. This volume, Volume II, provides the programmer/analyst with a technical description of the purpose, functions, general procedures, and programming techniques applicable to the modules and

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

388286 181

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

20. ABSTRACT (CONT'D)
subroutines of the Weapon/Target Identification Subsystem.

The Program Maintenance Manual complements the other QUICK Computer Manuals to facilitate application of the war gaming system. These manuals (Series 9-77 for Volumes I & II, Series 9-74 for Volumes III & IV) are published by the Command and Control Technical Center (CCTC), Defense Communications Agency (DCA), The Pentagon, Washington, DC 20301.