



DEFENSE COMMUNICATIONS AGENCY  
 COMMAND AND CONTROL TECHNICAL CENTER  
 WASHINGTON, D. C. 20301

12

IN REPLY  
 REFER TO: C314

**LEVEL III**  
 A050 041

30 April 1980

TO: RECIPIENTS

SUBJECT: Change 3 to Program Maintenance Manual CSM MM 9-77, Volume II,  
 Weapon/Target Identification Subsystem

1. Insert the enclosed change pages and destroy the replaced pages according to applicable security regulations.
2. A list of Effective Pages to verify the accuracy of this manual is enclosed. This list should be inserted before the title page.
3. When this change has been posted, make an entry in the Record of Changes.

FOR THE DIRECTOR:

57 Enclosures  
 Change 3 Pages

*[Signature]*  
 J. DOUGLAS BOTTLE  
 Assistant to the Director  
 for Administration

ADA 085814

*cc7c*  
 (QUICK) Program Maintenance Manual,  
 Volume II, Weapon/Target Identifica-  
 tion Subsystem, Change 3

**DTIC**  
**ELECT**  
**S**  
 JUN 12 1980  
**C**

31 30 APR 80

12 180

This document has been approved  
 for public release and sale; its  
 distribution is unlimited.

14) C314 - CSM - MM - 9 - 77 - V2 - CHG - 3

80 6 9 114

409658

500

DDC FILE COPY

EFFECTIVE PAGES - APRIL 1980

This list is used to verify the accuracy of GSM MM 9-77, Volume II after change 3 pages have been inserted. Original pages are indicated by the letter O, change 1 pages by the numeral 1, change 2 pages by the numeral 2, and change 3 pages by numeral 3.

<u>Page No.</u>	<u>Change No.</u>	<u>Page No.</u>	<u>Change No.</u>
Title Page	0	82-83	0
ii	3	84-86	1
iii	0	87	3
iv-vi	3	88-89	0
1	0	90	1
2	3	91-96	0
3	1	97	2
4	0	98	0
5	1	99-102.2	3
6	0	103-108	3
7	2	109	0
8-9	0	110-111.2	3
10	2	112-113	3
10.1-10.2	1	114-114.2	2
11-33	0	115-120	3
34	3	121-123	1
35-38	0	124-127	3
39	3	128	0
40-46	0	129	3
47	1	130	2
48-49	0	131	3
50	1	132	2
51	0	133	0
52-53	1	134-135	2
54	0	136-138	0
55-56	1	139-140	2
57-58	0	141	3
59-62	1	142-146	2
63	3	147	3
64-65	0	148	0
66	3	149-152	2
67-69	0	153	0
70	3	154-156	3
71-76	0	157	2
77	2	158-160	3
78-79	0	161-161.1	3
80	2	161.2-167.1	3
81	3	167.2-172	3

<u>Page No.</u>	<u>Change No.</u>
173-175	0
176-178	3
179	3
180-182	1
183	3
184	1
185-186	3
187	1
188	3
189	1
190-190.2	3
191-191.2	3
192-199	3
200	1
201-202	3

Accession For	
NHS GMAI	
D.C. TAB	
Unannounced	
Justification <i>per file</i>	
<i>50 on file</i>	
By _____	
Distribution/ _____	
Availability Codes	
Dist	Avail and/or special
<b>A</b>	

#### ACKNOWLEDGMENT

This document was prepared under the direction of the Chief for Military Studies and Analyses, CGTC, 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 DCA 100-75-C-0019. Change set two was prepared under Contract Number DCA 100-78-C-0035. Computer Sciences Corporation prepared change set three under Contract Number DCA 100-78-C-0042.

Section	Page
3.5 Identification of Subroutine Function.....	77
3.5.1 Subroutine DESTAB.....	77
3.6 Common Block Definition.....	78
3.7 Subroutine ENTMOD.....	80
3.8 Subroutine DESTAB.....	92
 4	
INDEXER MODULE.....	97
4.1 Purpose.....	97
4.2 Input.....	97
4.3 Output.....	97
4.4 Concept of Operation.....	98
4.5 Identification of Subroutine Functions.....	98
4.5.1 Subroutine COMPLEX.....	98
4.5.2 Subroutine CRTBLE.....	98
4.5.3 Subroutine SETVAL.....	98
4.6 Common Block Definition.....	98
4.7 Subroutine ENTMOD.....	100
4.8 Subroutine COMPLEX.....	110
4.9 Function-CRTBLE.....	118
4.10 Subroutine SETVAL.....	120
 5	
PLANSET MODULE.....	126
5.1 Purpose.....	126
5.2 Input.....	126
5.3 Output.....	126
5.4 Concept of Operation.....	127
5.5 Identification of Subroutine Functions.....	128
5.5.1 Subroutine ADJUSTGP.....	128
5.5.2 Subroutine CALCOMP.....	128
5.5.3 Subroutine GRPEM.....	128
5.5.4 Subroutine SRTTGT.....	128
5.6 Internal Common Blocks.....	129
5.7 Subroutine ENTMOD.....	132
5.8 Subroutine ADJUSTGP.....	135
5.9 Subroutine CALCOMP.....	140
5.10 Subroutine GRPEM.....	147
5.11 Subroutine PRINTGP.....	158
(Entry PRINTW).....	
5.12 Subroutine SRTTGT.....	160
5.13 Subroutine TANKER.....	173
 APPENDIX A.....	179
 DISTRIBUTION.....	199
 DD Form 1473.....	201

## ILLUSTRATIONS

Figure		Page
1	Major Subsystems of the QUICK System.....	2
2	Procedure and Information Flow in QUICK/HIS 6000..	3
3	JAD Format.....	6
4	JAD Loading Module.....	13
5	Subroutine ASSIGN.....	15
6	Subroutine ALPHAS.....	19
7	Subroutine PLAYERS.....	37
8	Subroutine TOPRINT.....	41
9	Subroutine SELECT.....	49
10	Subroutine ADTOBASE.....	57
11	Subroutine DEFAULT.....	65
12	Subroutine KRUNCH.....	68
13	Subroutine SAMSET.....	72
14	Subroutine ASTERISK.....	74
15	DEMOM Module.....	81
16	Subroutine DESTAB.....	93
17	INDEXER Module.....	101
18	Subroutine COMPLEX.....	112
19	Function CRTBLE.....	119
20	Subroutine SETVAL.....	121
22	PLANSET Module.....	133
23	Subroutine ADJUSTGP.....	136
24	Subroutine CALCOMP.....	142
25	Subroutine GRPEM.....	149
26	Subroutine PRINTGP.....	159
27	Subroutine SRTTGT.....	161.2
28	Subroutine TANKER.....	174
29	Intentionally Deleted.....	177
30	Time Dependent Target Value Curve.....	187

## TABLES

Table		
1	Module JLM Internal Common Blocks.....	11
2	Module DEMOM Internal Common Blocks.....	79
3	Module INDEXER Internal Common Blocks.....	99
4	Module PLANSET Internal Common Blocks.....	130
5	Computations for Reprogrammable Missiles.....	184
6	Sample Exemplar Target Value Calculation.....	189

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 tapes to simulator subsystems external to QUICK. QUICK has been programmed in FORTRAN for use on the CCTC HIS 6000 computer system.

The QUICK 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 Maintenance Manual complements the other QUICK Computer System Manuals to facilitate application of the war gaming system. This volume, Volume I, in two parts, provides the programmer/analyst with a technical description of the purpose, functions, general procedures, and programming techniques applicable to the modules (programs) and subroutines of the Data Management 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 9-77, Volume III  
Computer System Manual CSM UM 9-77, 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 QUICK's 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.

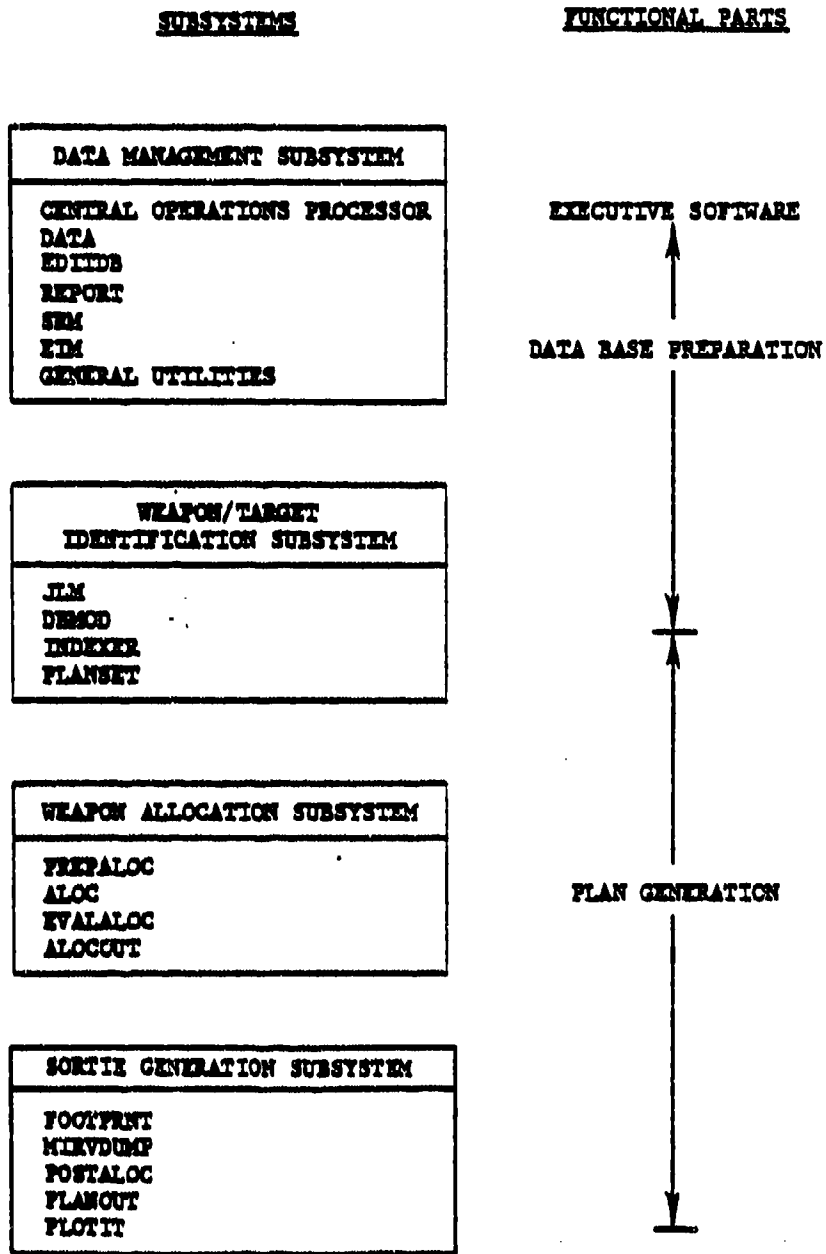


Figure 1. Major Subsystems of the QUICK System

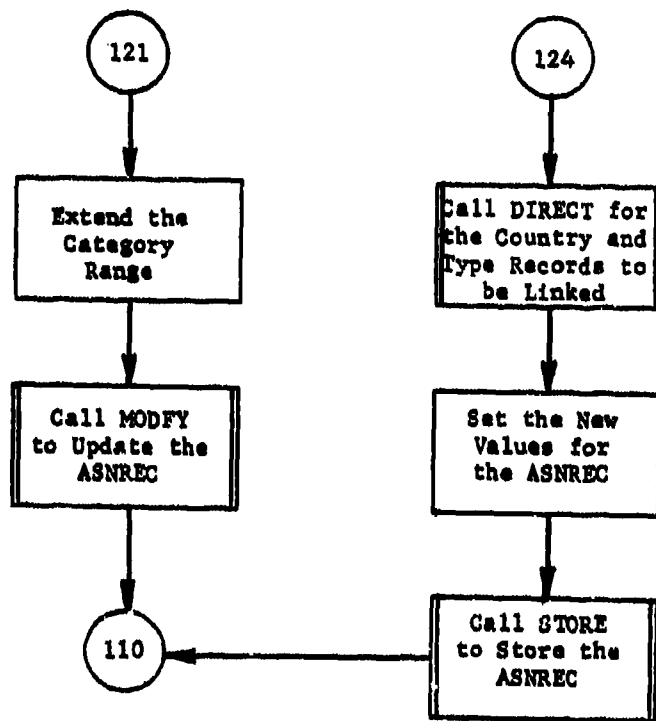


Figure 6. (Part 15 of 17)

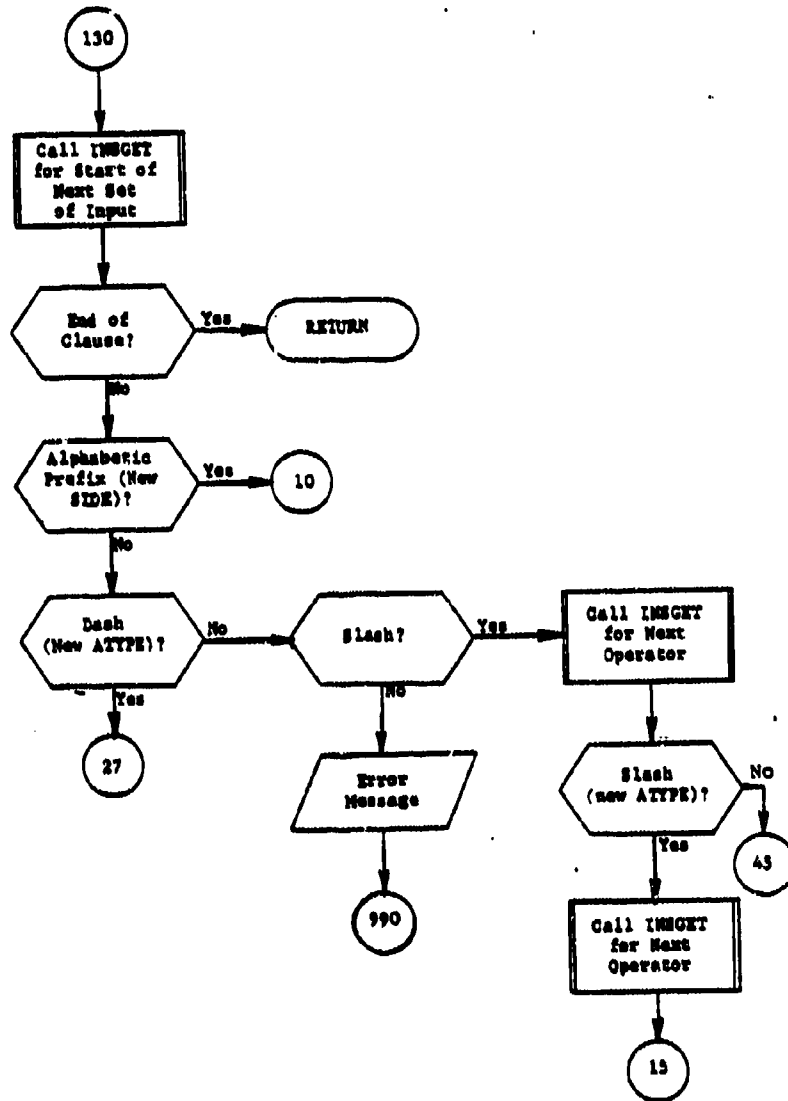


Figure 6. (Part 16 of 17)

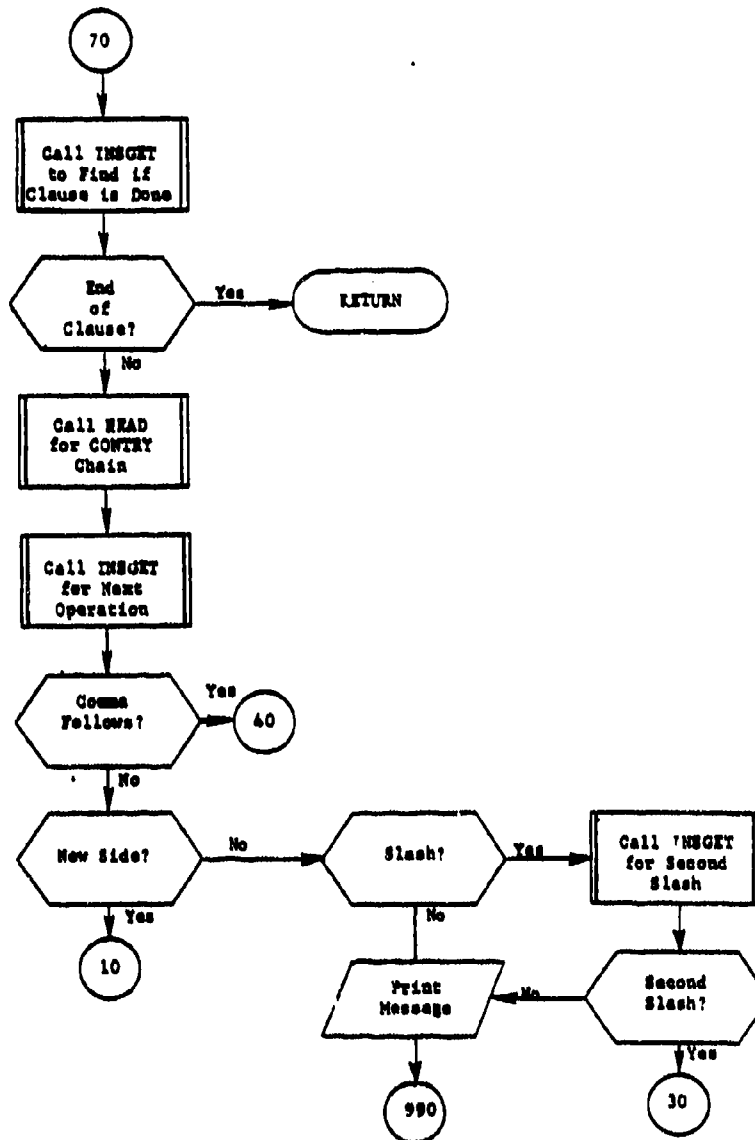


Figure 7. (Part 3 of 3)

### 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, NEXTTT, 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.

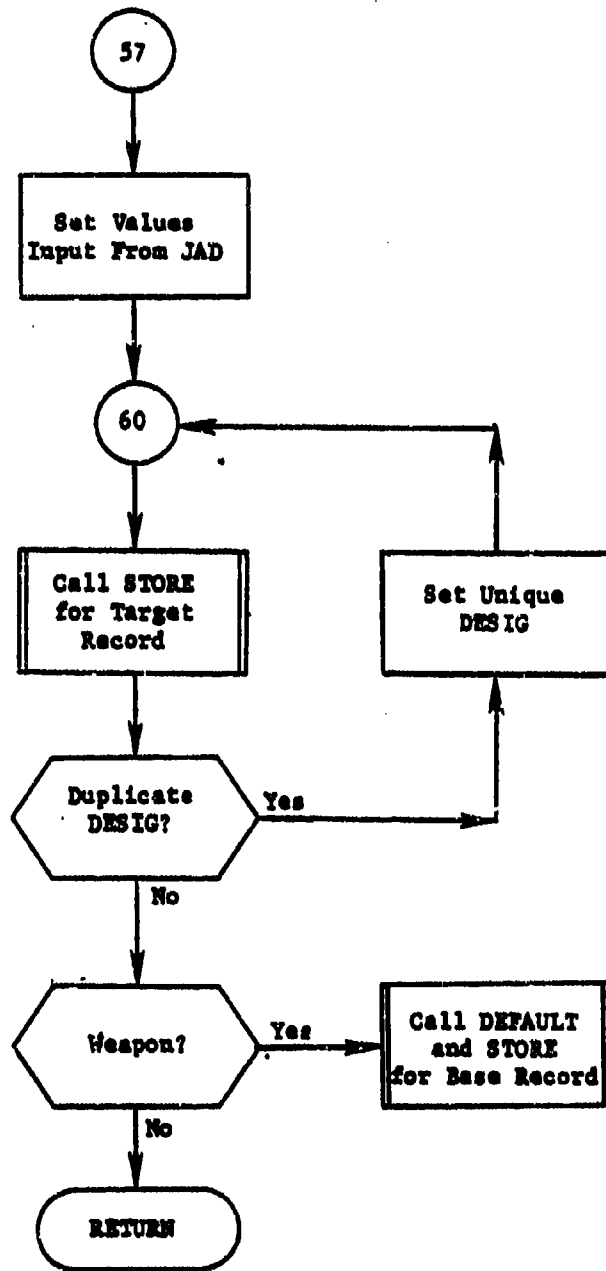


Figure 10. (Part 7 of 7)

### 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.

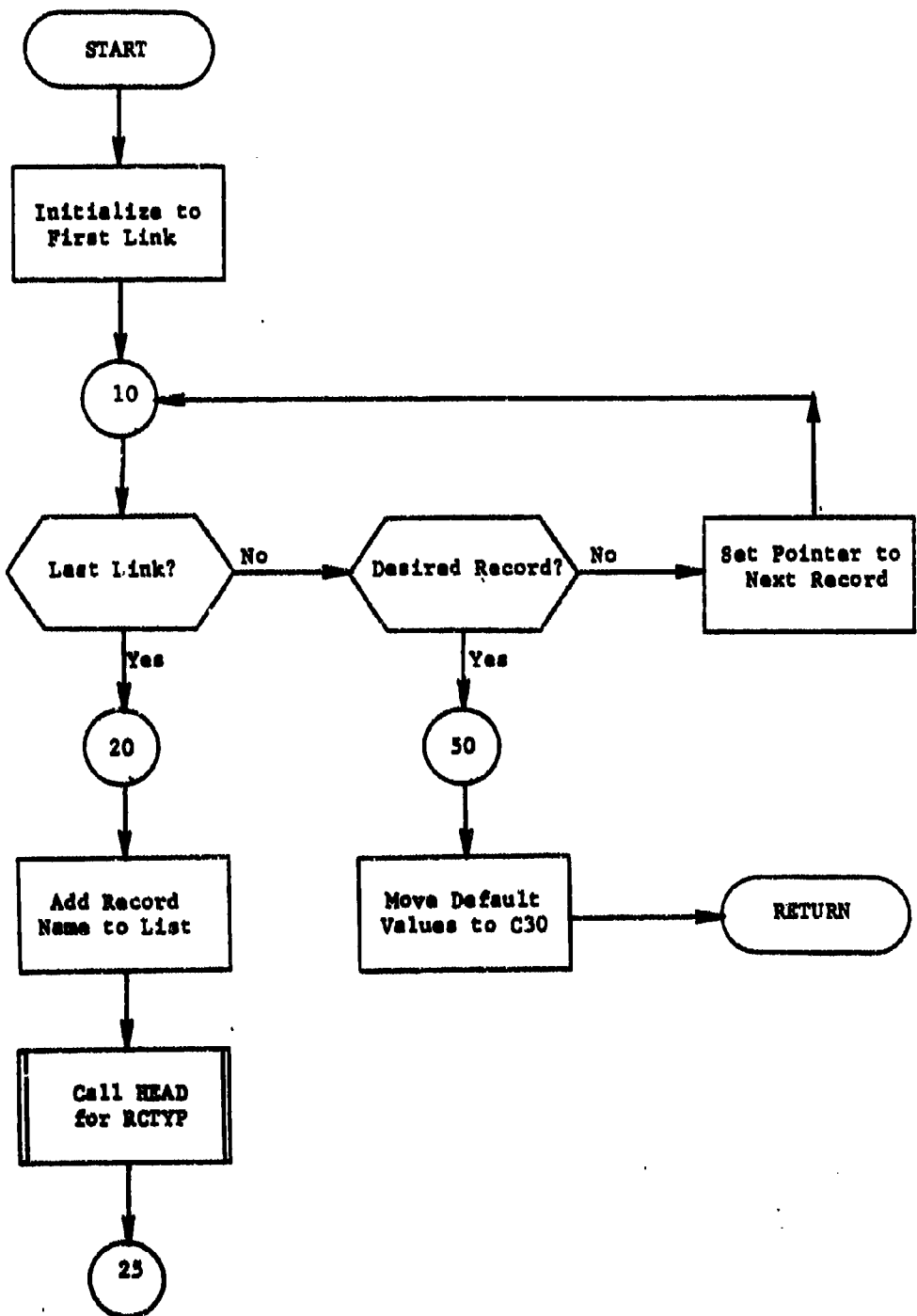


Figure 11. Subroutine DEFAULT (Part 1 of 2)

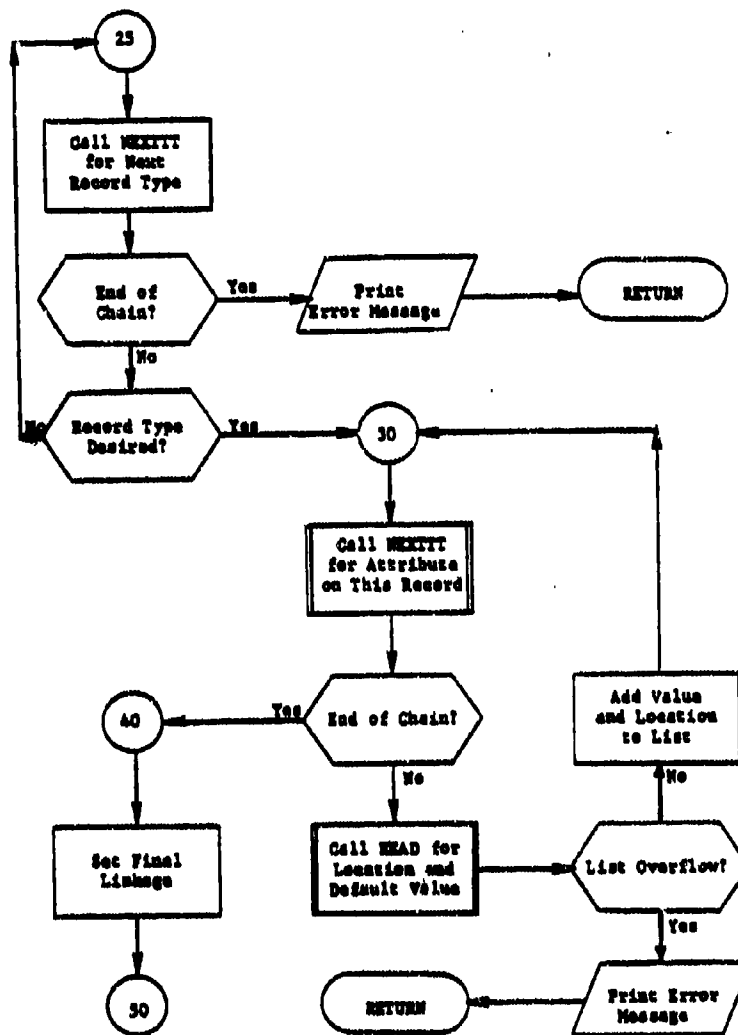


Figure 11. (Part 2 of 2)

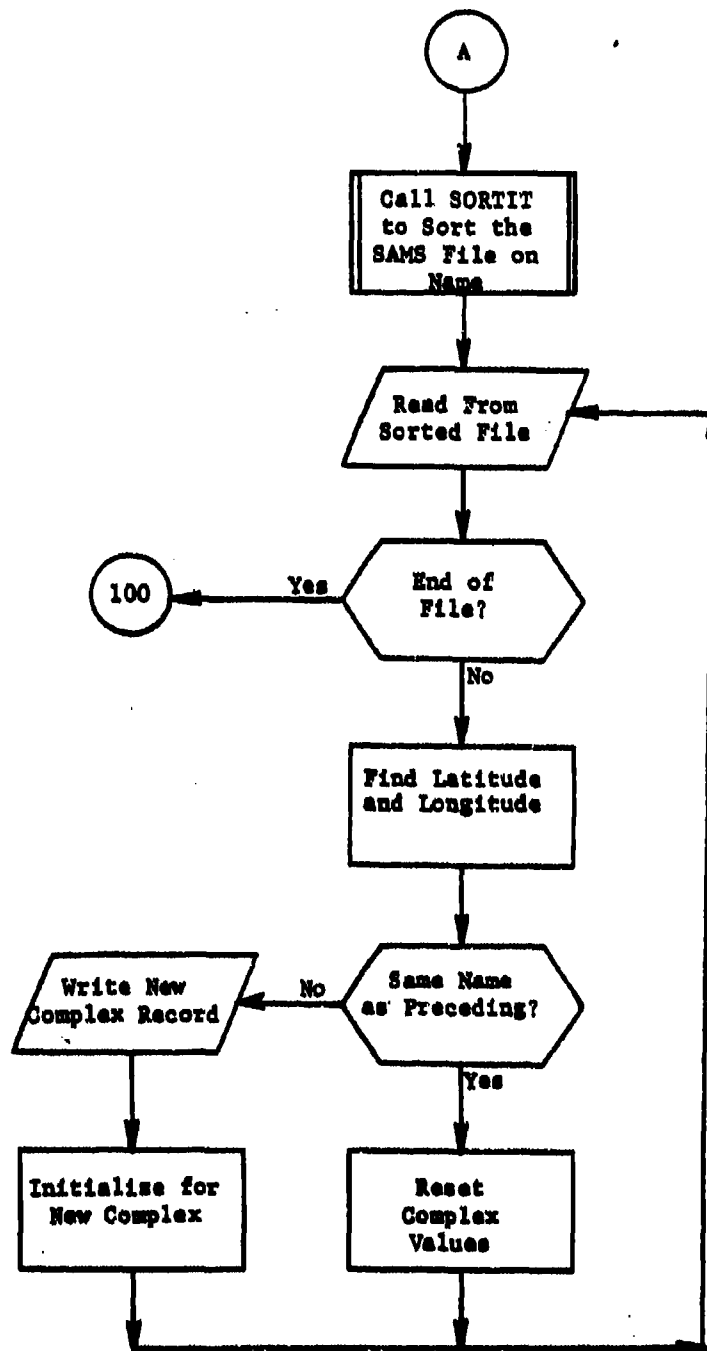


Figure 12. (Part 2 of 3)

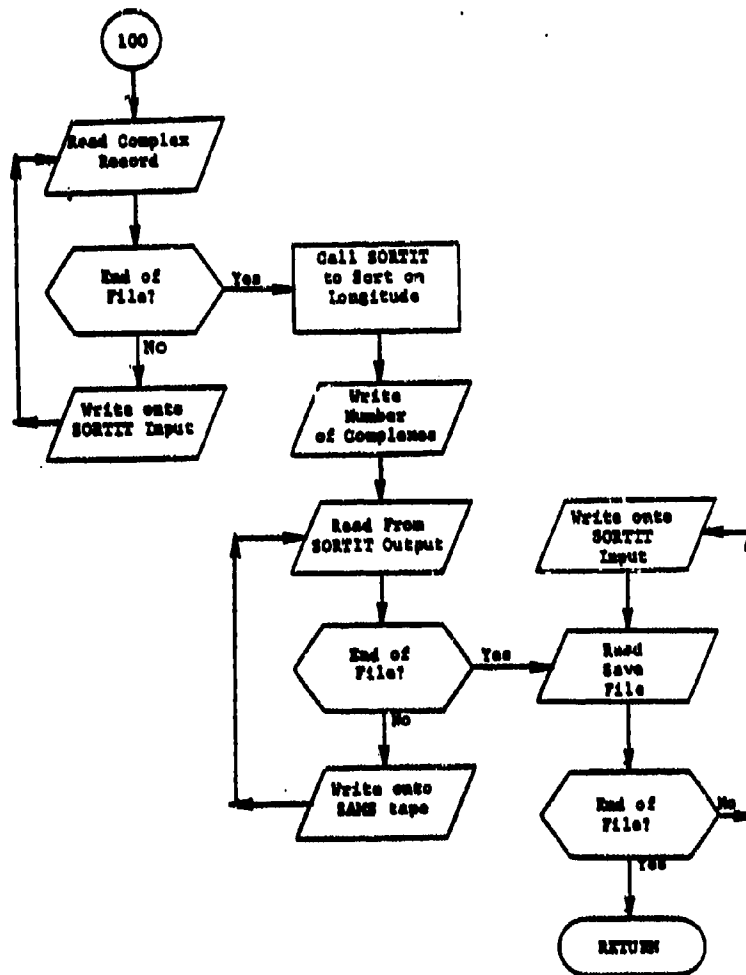


Figure 12. (Part 3 of 3)

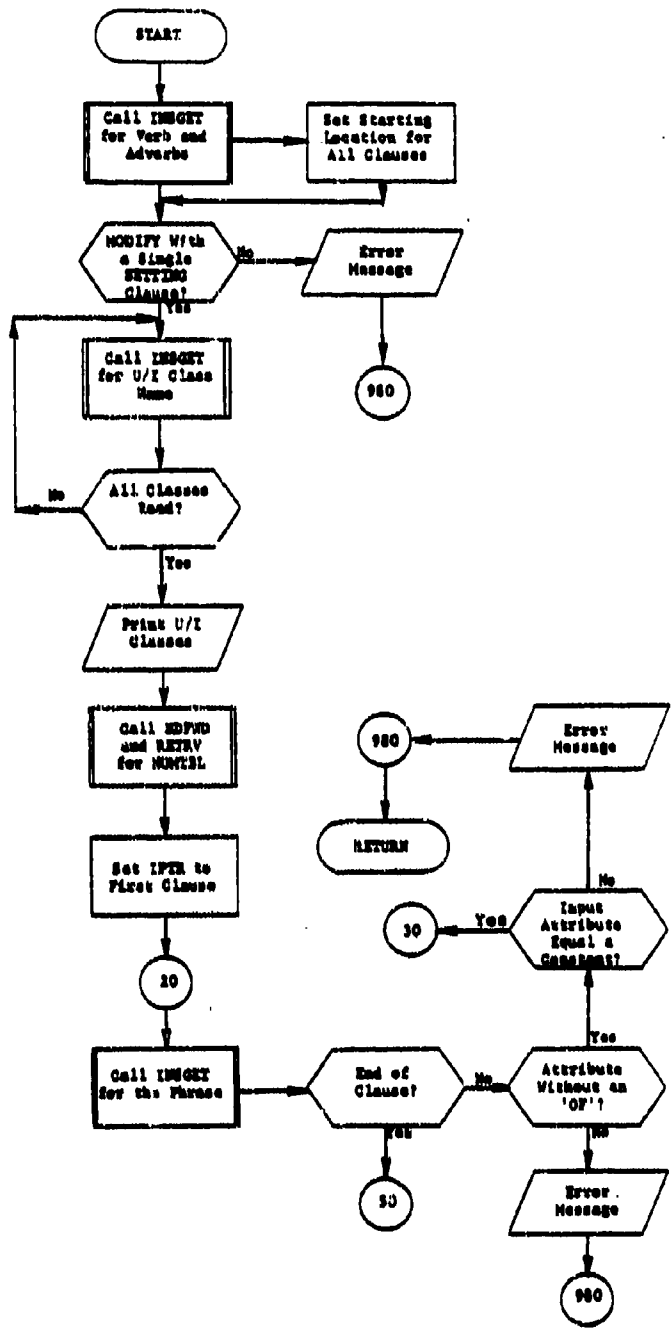


Figure 15. DBMOD Module (Part 1 of 10)

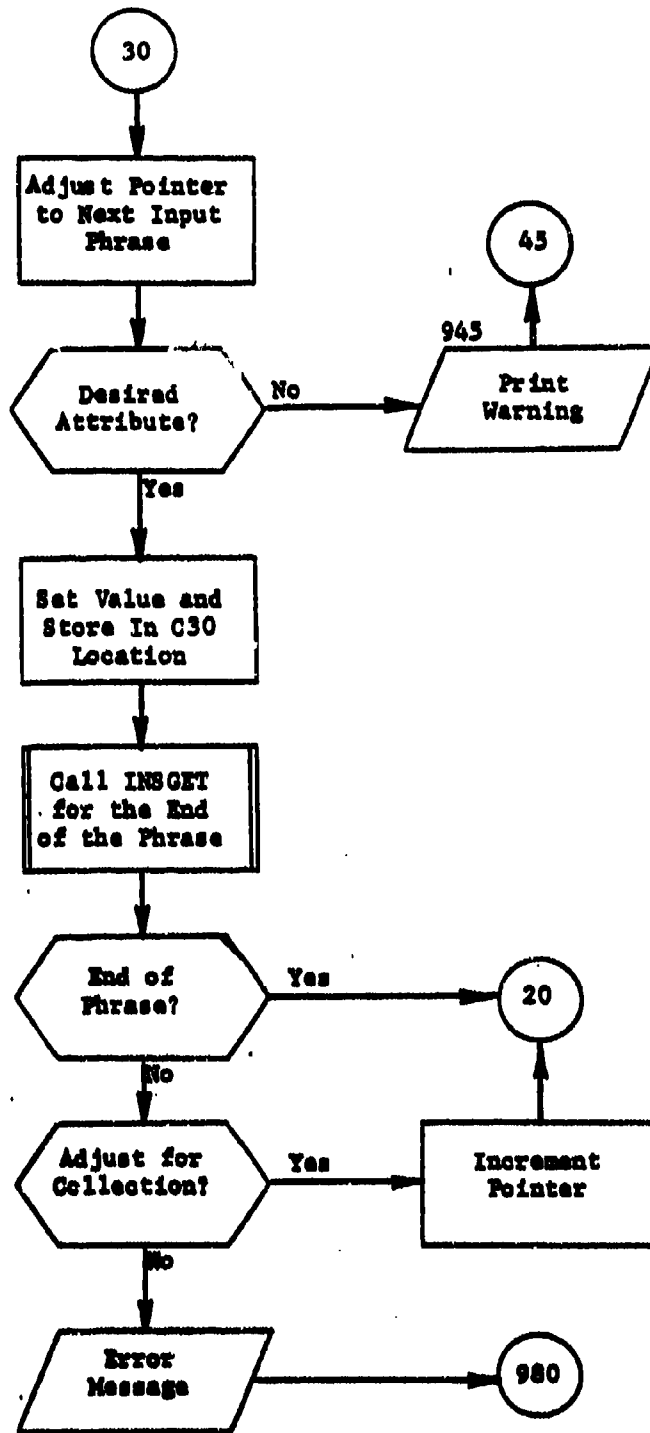


Figure 15. (Part 2 of 10)

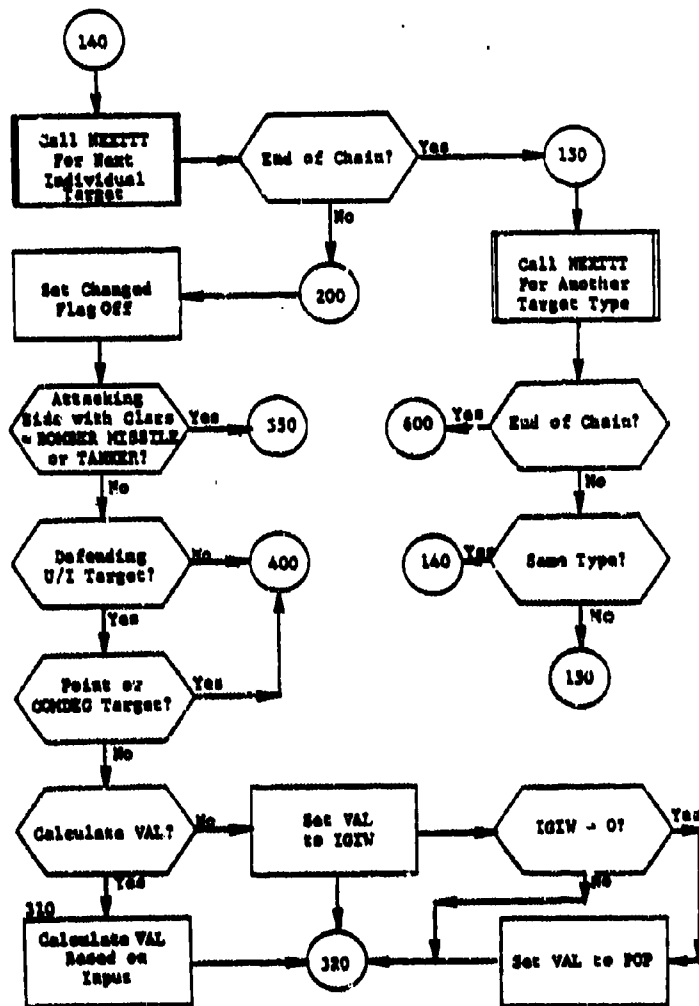


Figure 15. (Part 7 of 10)

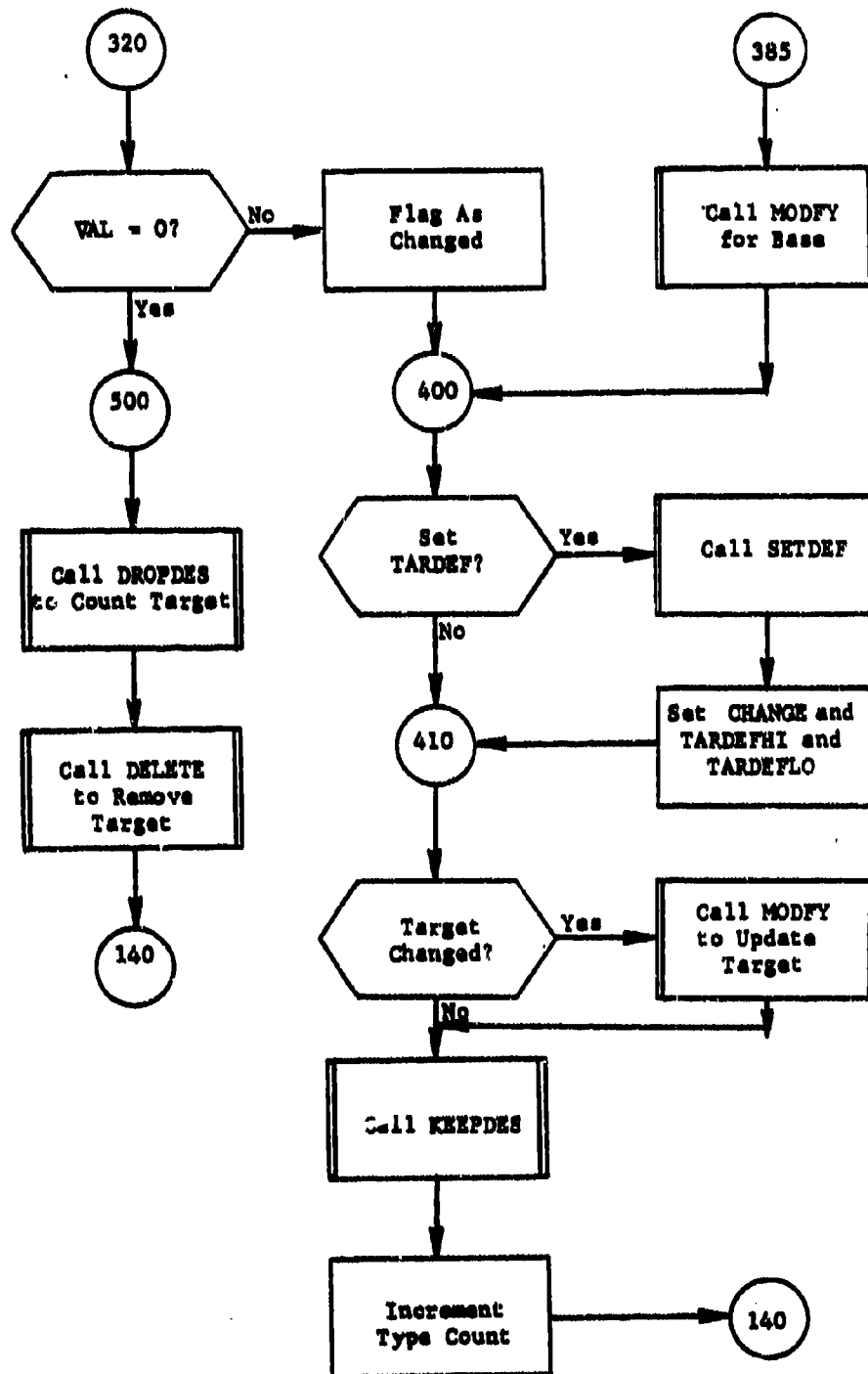


Figure 15. (Part 8 of 10)

Table 3. Module INDEXER Internal Common Blocks

<u>BLOCK</u>	<u>VARIABLE OR ARRAY</u>	<u>DESCRIPTION</u>
CYIELD	SYIELD(2)	Yield, in megatons, used in forming complexes. Value is user determined.
IOPRT	IOPRT	If zero, nonstandard prints are suppressed. User determined.
	FIRST	If true, causes initialization complexing logic to be performed.
	NOBASE	If true, implies that as complexes are formed, they will not be saved on the data base.
KEEP		As used within ENTMOD, contains local controlling parameters (used to reduce core space). As used within subroutine COMPLEX, contains varying definitions of target data.
NMCLAS		Contains the class name for the NUMTBL record.

#### 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 INDEXER is executed)

FORMAL PARAMETERS: None

COMMON BLOCKS: C10, C13, C25, C20, C30, CYIELD, IOPRT, KEEP

SUBROUTINES CALLED: COMPLEX, CRTBLE, DIRECT, HDFND, HEAD, INSGET, ITLE, MODFY, NEXITT, RETRV, SETVAL, VLRADP

CALLED BY: COP

#### Method:

Module INDEXER begins (figure 17) by reading (through utility subroutine INSGET) and storing user input parameters. 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. Then for each unique vulnerability contained within the data base, a complexing lethal radius is calculated and stored. 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 DEMOD. 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, ONPRINTS, and RECALC are checked for. Use of adverb ONPRINTS with a value of 1 (ONPRINTS 1), implies nonstandard prints are to be produced; ONPRINTS 2 implies that initialization logic in COMPLEX is to be executed, ONPRINTS 3 will make NOBASE .TRUE., implying that as complexes are formed, the match key ICOMPL will not be modified. Adverb VNOPTION implies that complexing lethal radius is to be obtained from hardcoded 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. Use of the RECALC adverb will cause target complexing by a call to COMPLEX.

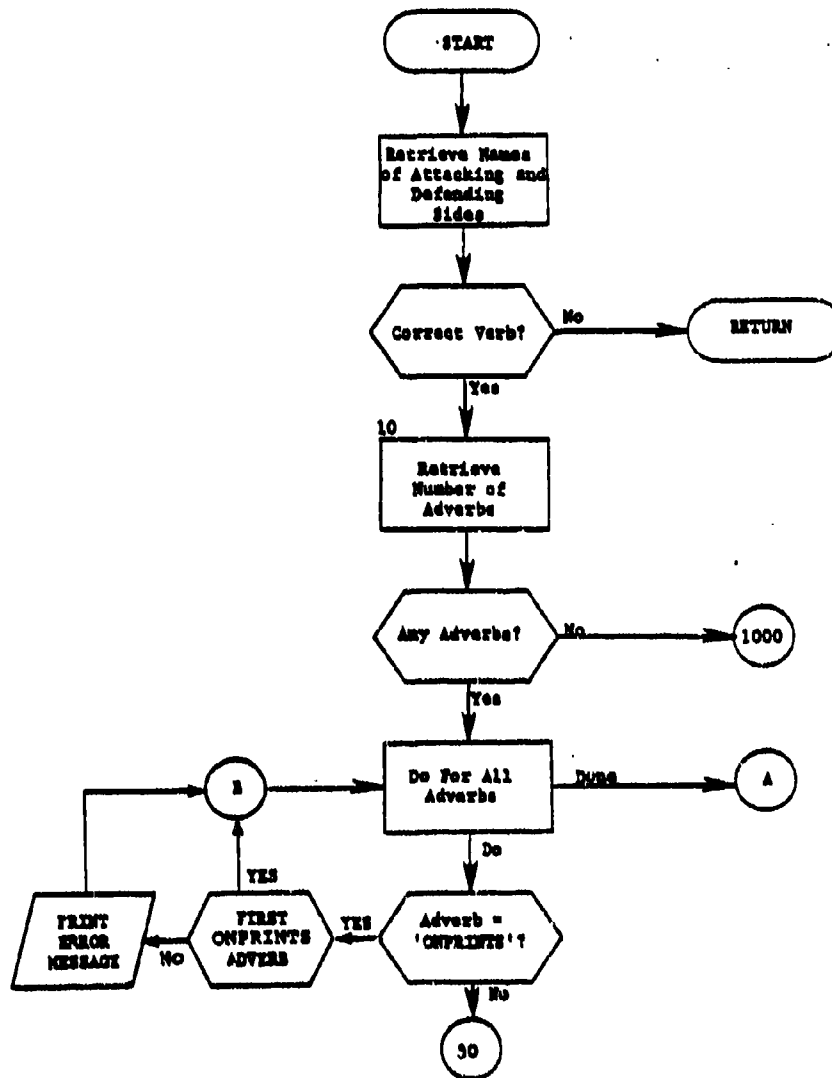


Figure 17. INDEKER Module (Part 1 of 8)

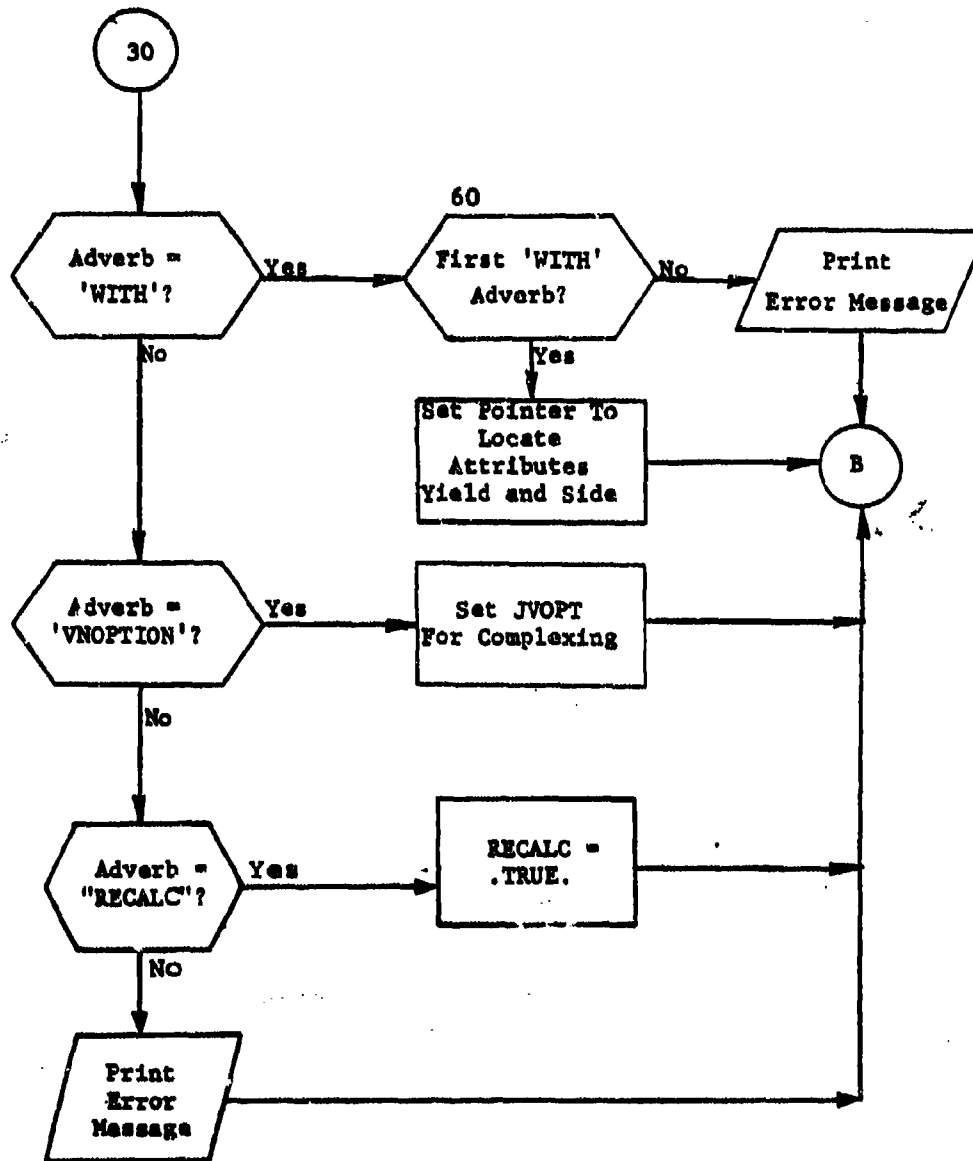


Figure 17. (Part 2 of 8)

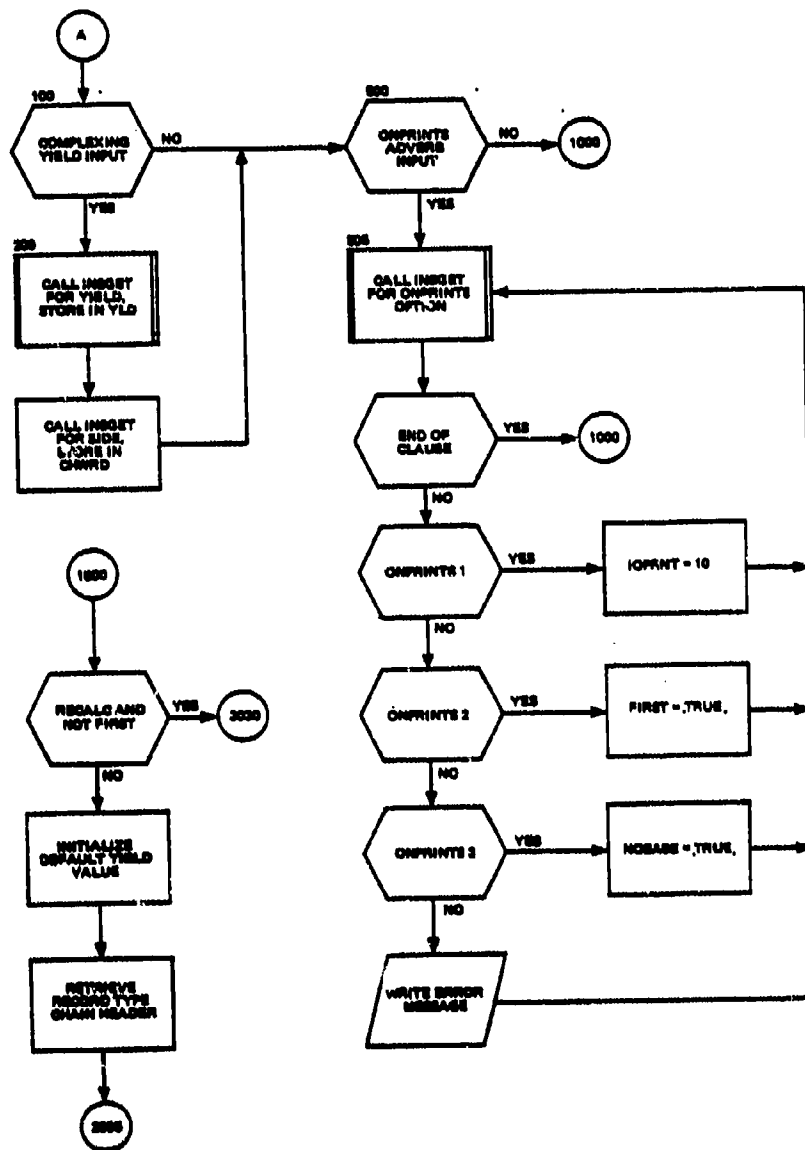


Figure 17. (Part 3 of 8)

THIS PAGE INTENTIONALLY LEFT BLANK

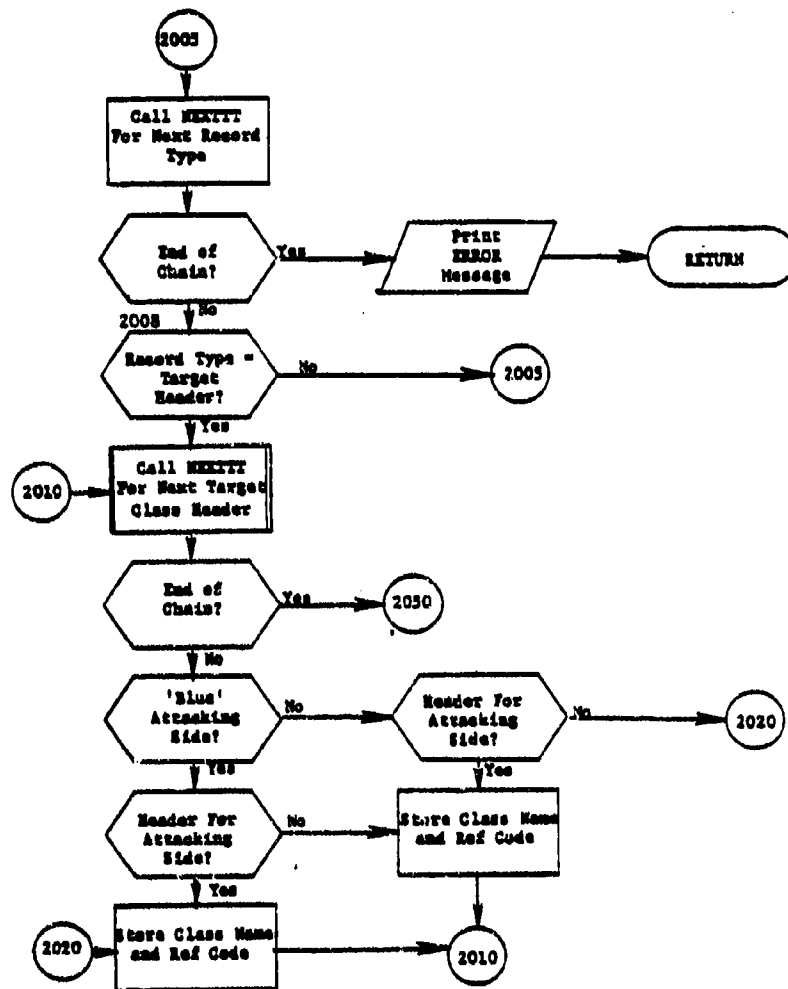


Figure 17. (Part 4 of 8)

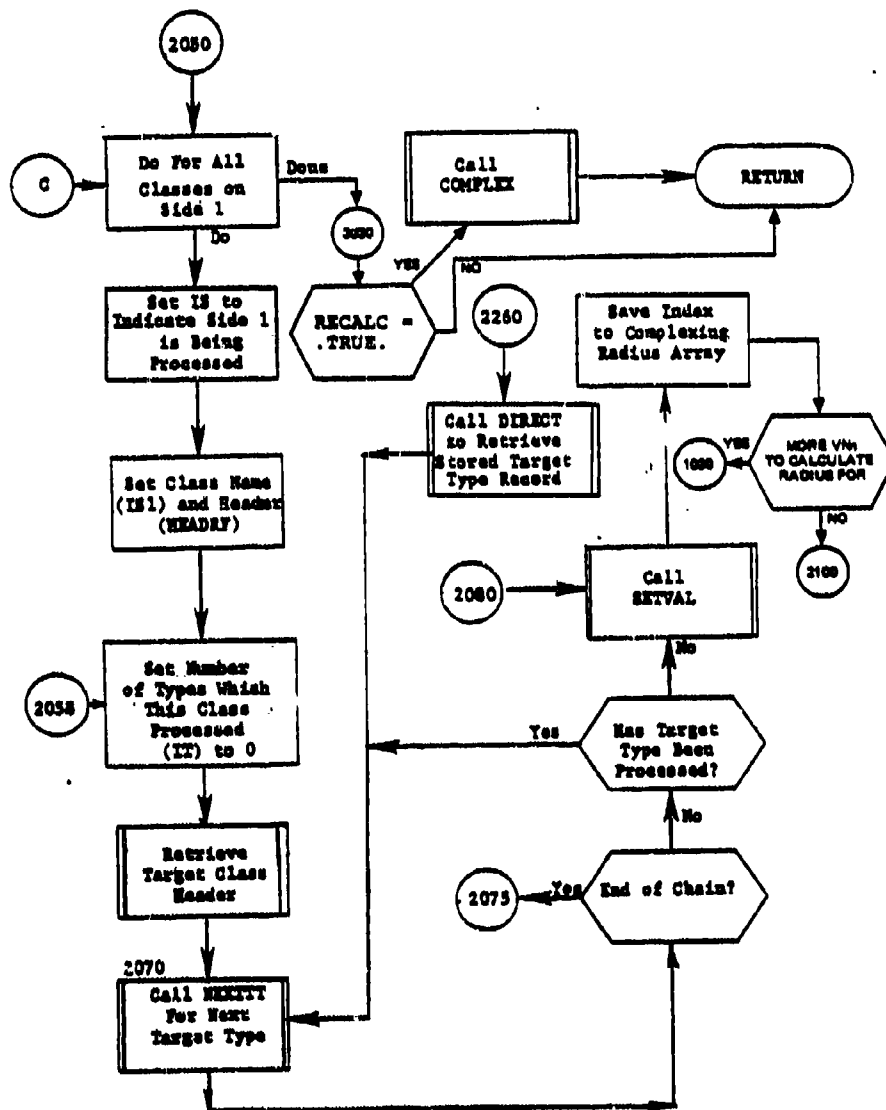


Figure 17. (Part 5 of 8)

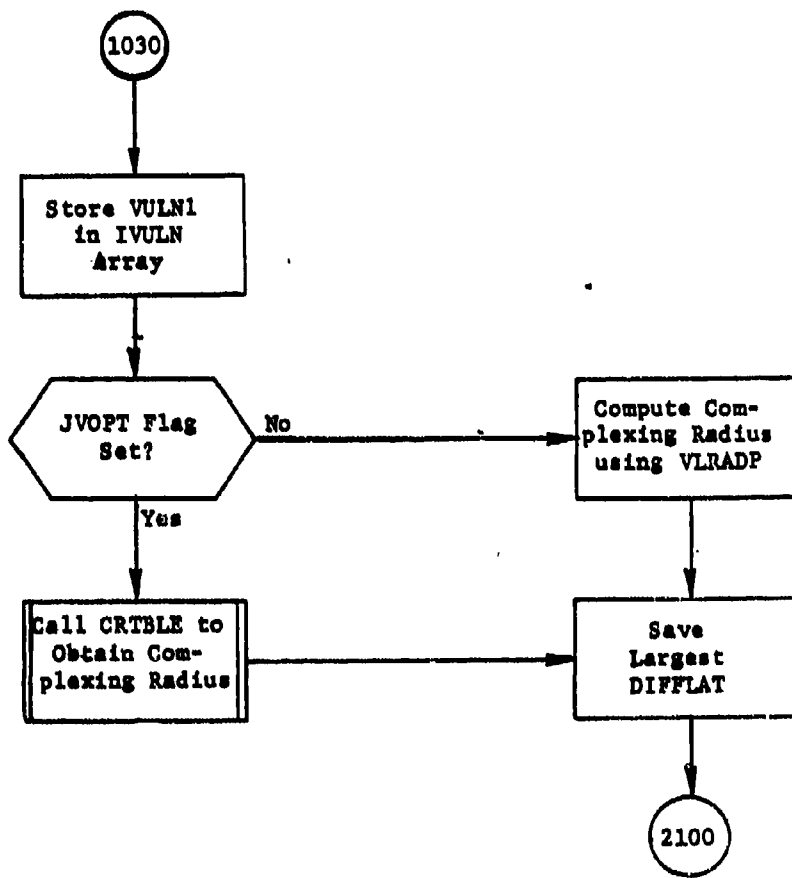


Figure 17. (Part 6 of 8)

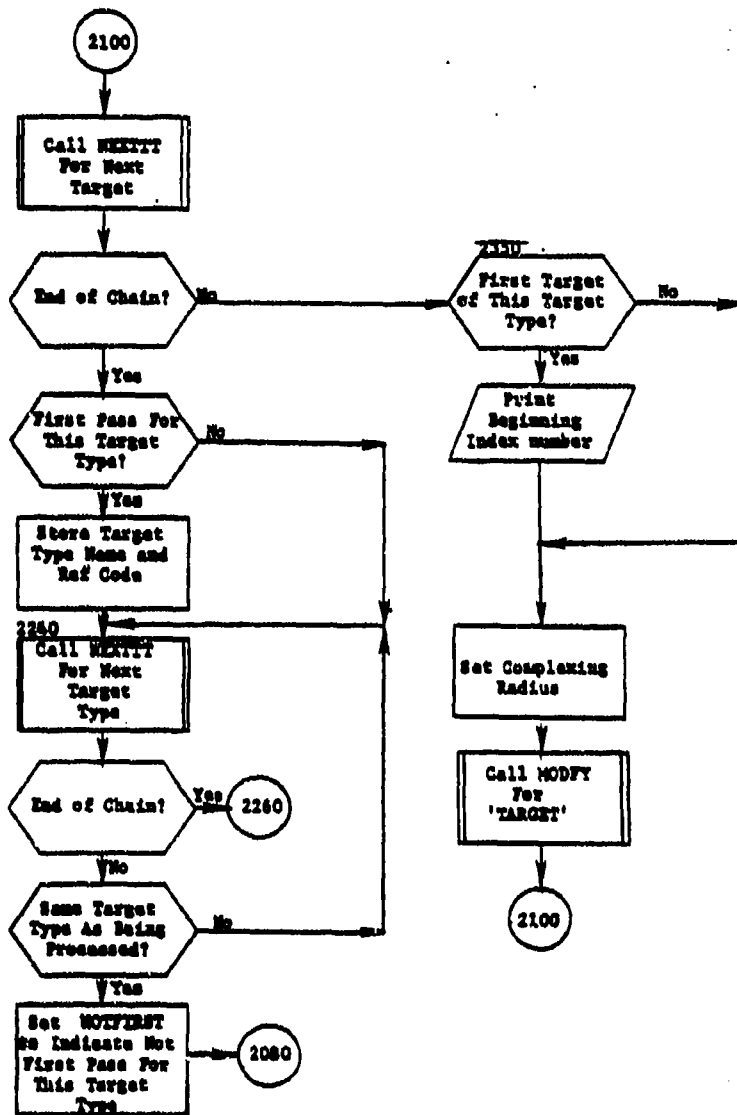


Figure 17. (Part 7 of 8)

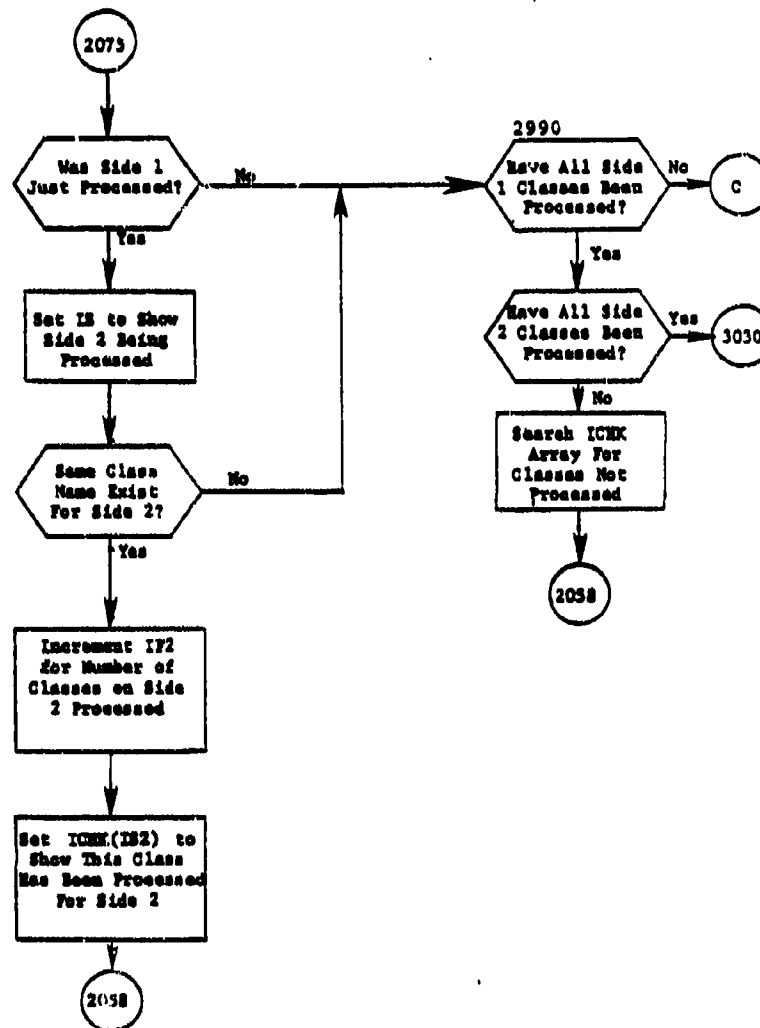


Figure 17. (Part 8 of 8)

### 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. DIFFLAT is stored on the data base by being equivalenced to EXNBOMB in C30 to facilitate restart.

Vulnerabilities (attribute VULN) are collected under header 'VNTRKD' 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

COMMON BLOCKS: C10, C15, C30, IOPRT, KEEP, NMCLAS

SUBROUTINES CALLED: DIRECT, DELETE, HDFND, KEYMAKE, GLOG, MODFY, NEXTTT, ORDER, RETRV, SLOG, STORE, TIMEME, TIMEOUT

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.

Each target stored within the data base is retrieved, sorted on increasing latitude (methodology outlined in next subsection), and final results written onto data file unit number IR. Data file IR is then read, target data stored in working arrays /KEEP/, and complexes formed.

#### Complexing Algorithm

The search for complex targets begins by comparing differences in latitude for consecutive targets sorted by increasing latitude, 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.

### Target Latitude Sort

Data based stored targets are sorted on increasing latitude in order to simplify the complexing burden. This reordering is performed by a basic sort/merge technique and is as follows:

- o Retrieve a data base target record
- o Store target latitude in array KEY1 and target reference code in array KEY3.
- o When working arrays are filled branch to label 9000 to sort the arrays and merge with any previous sorts. Arrays are sorted via the standard QUICK subroutine ORDER. If there were multiple calls to label 9000, results were written onto file unit IR. This file is read and results merged with contents in working arrays.
- o After sort/merge has been completed, flip data read and write files (IR and IW) reset pointers to initial conditions and continue IDS query.

After this process is complete data unit IR contains all target records in latitude sort. Each record contains two words: target latitude and target reference code.

### Target Data Retrieval

After targets are sorted, complexes are formed by querying data file IR and retrieving additional data base information. The size of QUICK's data base mandates that IDS interface be minimal. Therefore needed target data is stored in working arrays KEY1, KEY2, KEY3. Latitude and the fraction portion of lethal radius is packed in KEY1; Longitude, complex flag indicator, and the integer portion of the lethal radius is packed in KEY2; and the reference code is stored in KEY3. Up to 3,800 individual targets may be contained in the working arrays.

As complexes are formed information is obtained from the working arrays as directed by the controlling pointer INDEX. When INDEX exceeds array limits, complexing must temporarily halt and working arrays must be redefined. At this juncture, processing has been completed for all records starting with the first entry into the working arrays up to location LCOMP(1) minus one (recall that LCOMP contains the indexes of potential complexes). With this knowledge, working arrays are compressed, file IR read, and working arrays filled with additional target data. This process begins at label 565.

Code must honor the condition where arrays can no longer be compressed (LCOMP(1)=1). If this occurs, target data is written onto an indexed random file (ISAMLUN) and query, then, is from that file.

### Restart and NOBASE Options

A restart capability is available in INDEXER to allow a user to save the data base just prior to an abort caused by insufficient processing time. A series of small jobs can be run with a save after each. The complexing can be continued from any of the saved data bases. This capability is implemented by the ONPRINTS 2 switch (FIRST). If the switch is on all initialization is accomplished. If the switch is off the old complexes will remain and any uncomplexed targets will be complexed. The variable VOZ is used to store complexing status in the data base between jobs. A value of -1 indicates the target is complexed and a value of -2 indicates the target will not complex with another target. Complexing can also be accomplished without storing the final complexes on the IDS data base. This option greatly reduces run time and should be used when experimentation with complexing parameters is being done. This capability is implemented by the ONPRINT 3 switch (NOBASE).

Subroutine COMPLEX is illustrated in figure 18.

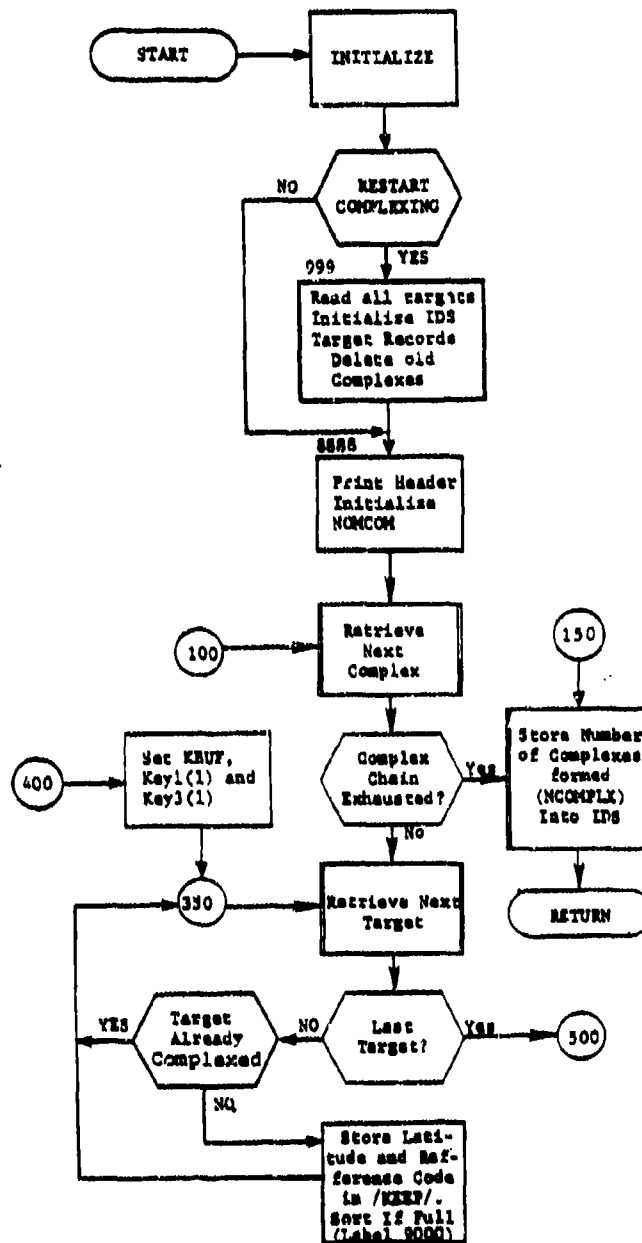


Figure 18. Subroutine COMPLEX (Part 1 of 8)

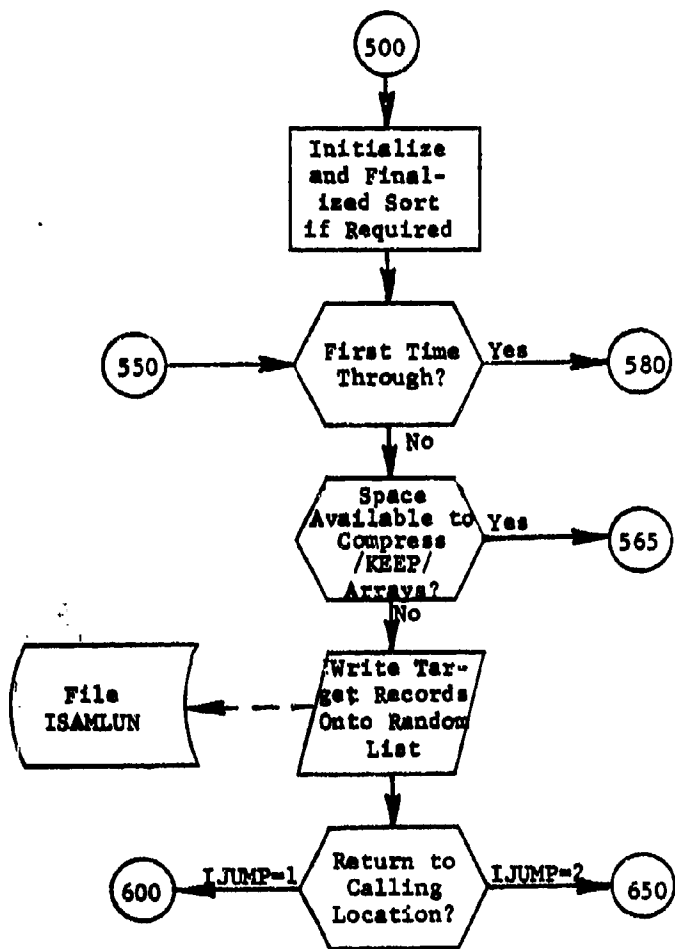


Figure 18. (Part 2 of 8)

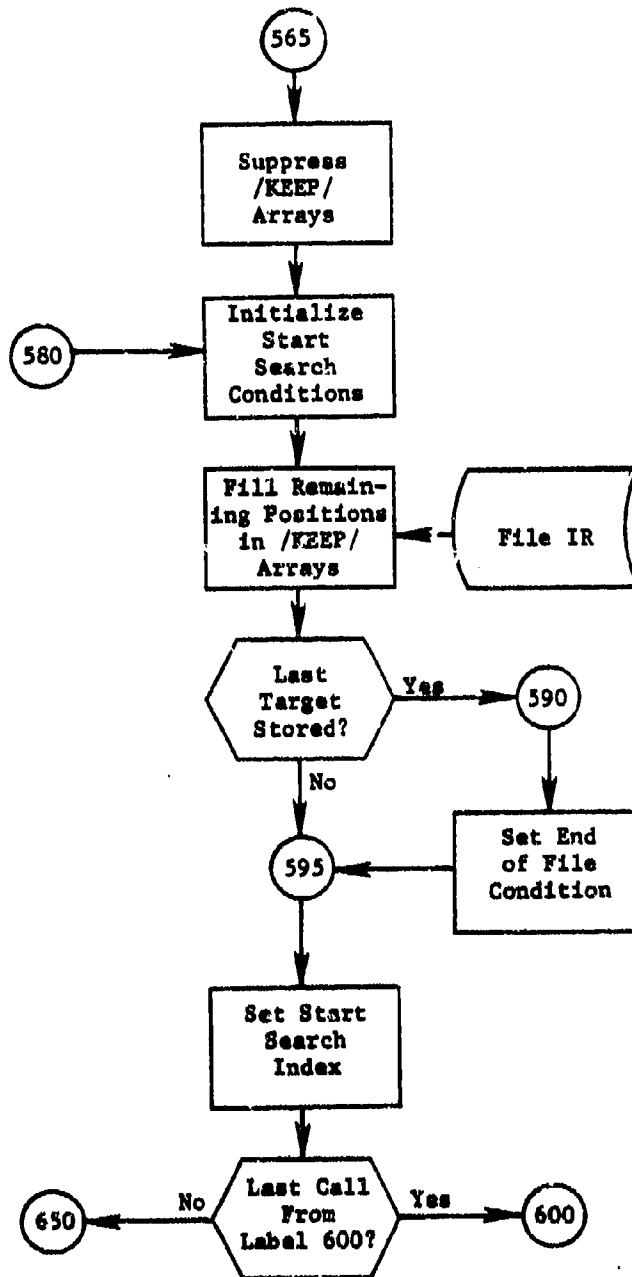


Figure 18. (Part 3 of 8)

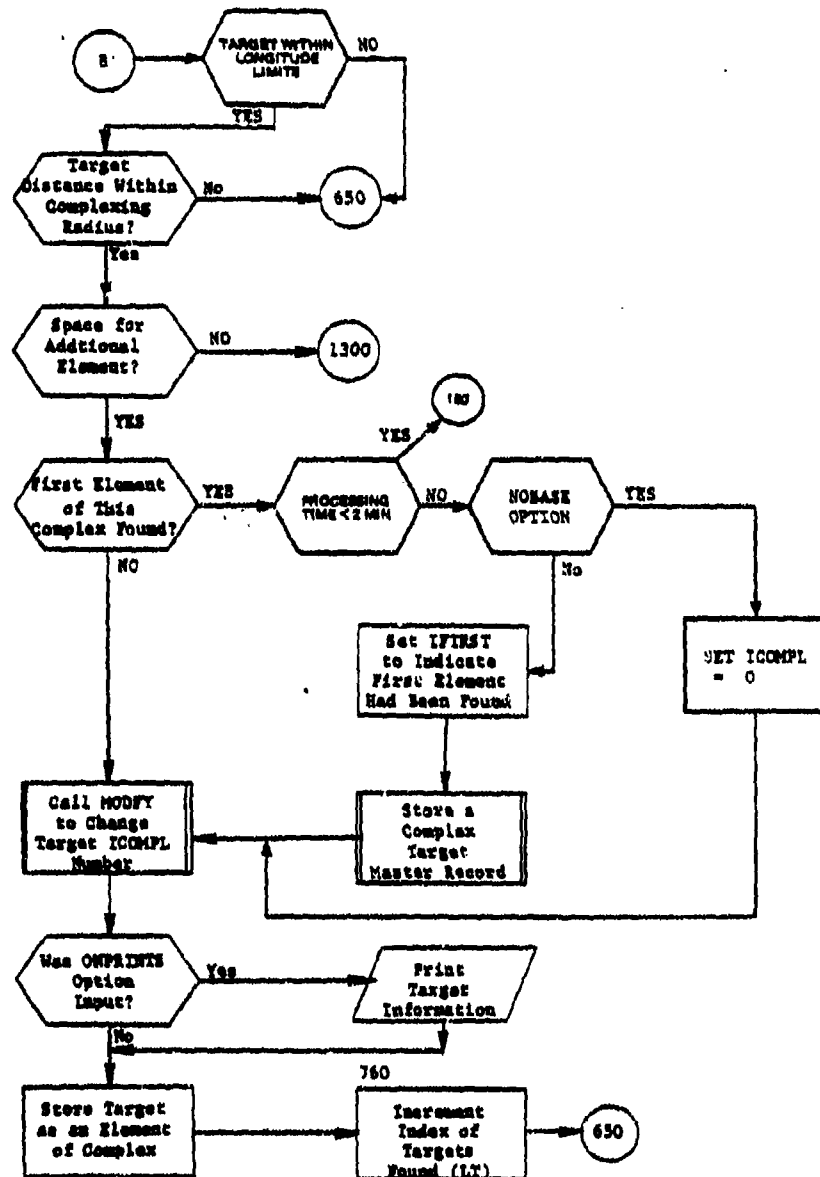


Figure 18. (Part 6 of 8)

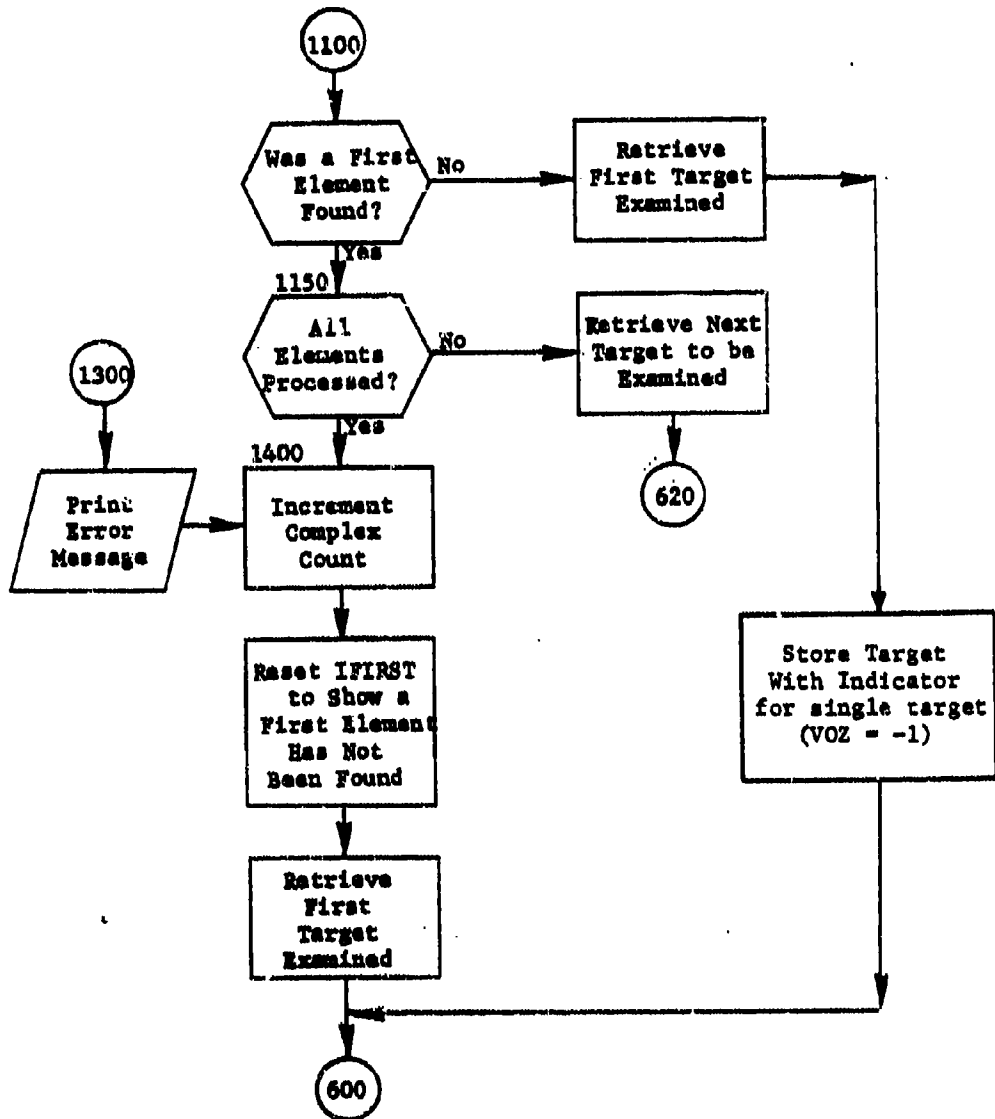


Figure 18. (Part 7 of 8)

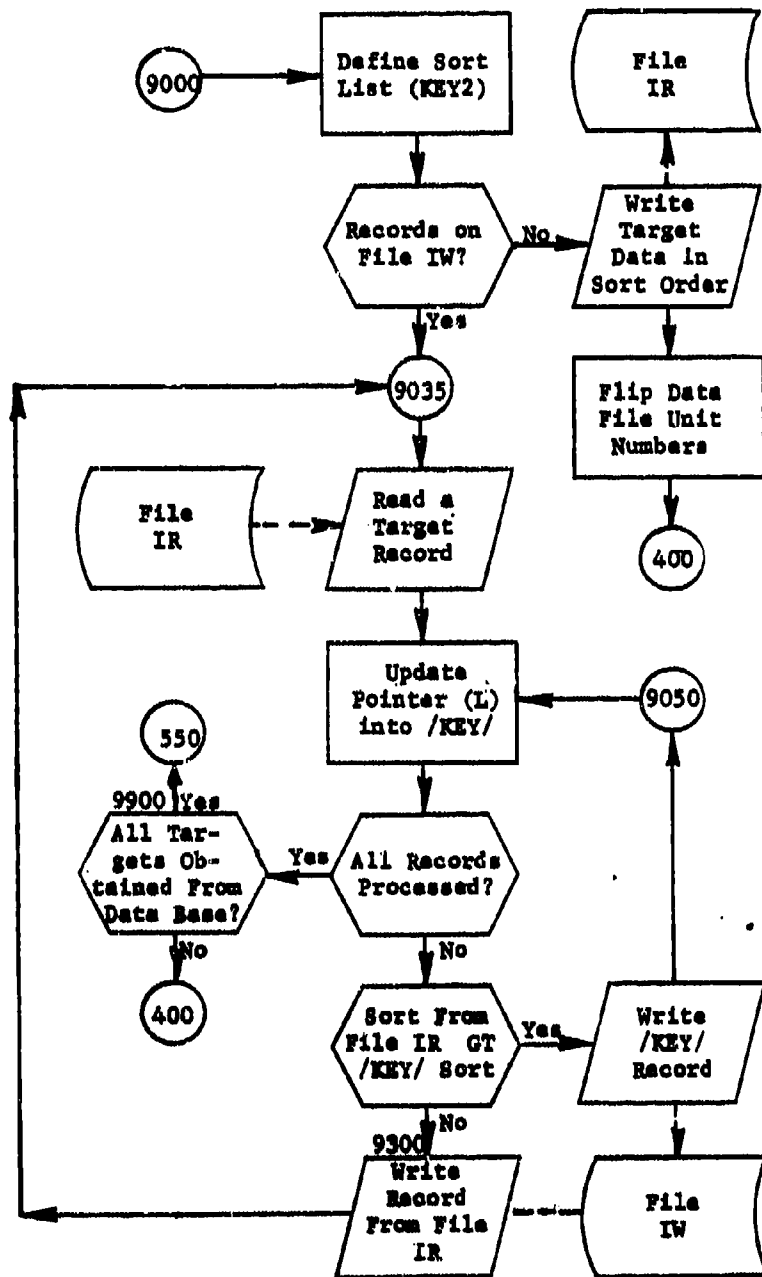


Figure 18. (Part 8 of 8)

#### 4.9 Function CRTBLE

PURPOSE: To calculate complexing lethal radius

ENTRY POINTS: CRTBLE

FORMAL PARAMETERS: IVN, vulnerability

COMMON BLOCKS: None

SUBROUTINES CALLED: VLRADP

CALLED BY: ENTMOD (of overlay link INDEXER)

#### 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 VLRADP is called to find an adjusted VN which in turn is used as the index into the P and Q arrays.

Function CRTBLE is illustrated in figure 19.

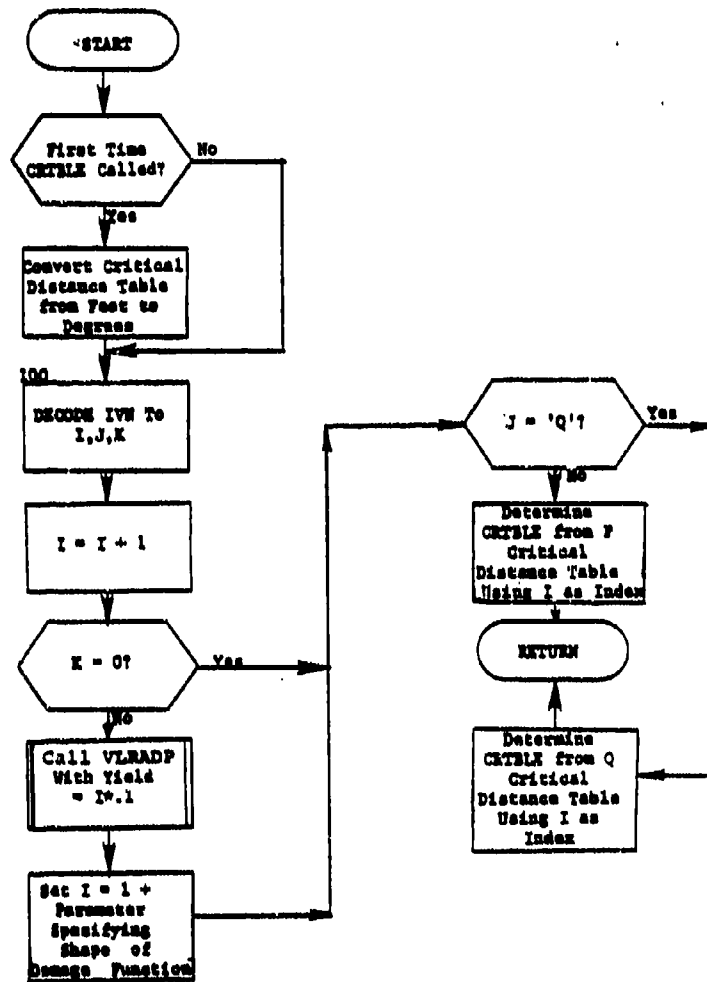


Figure 19. Function CRTBLE

#### 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 INDEXER)

#### 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.

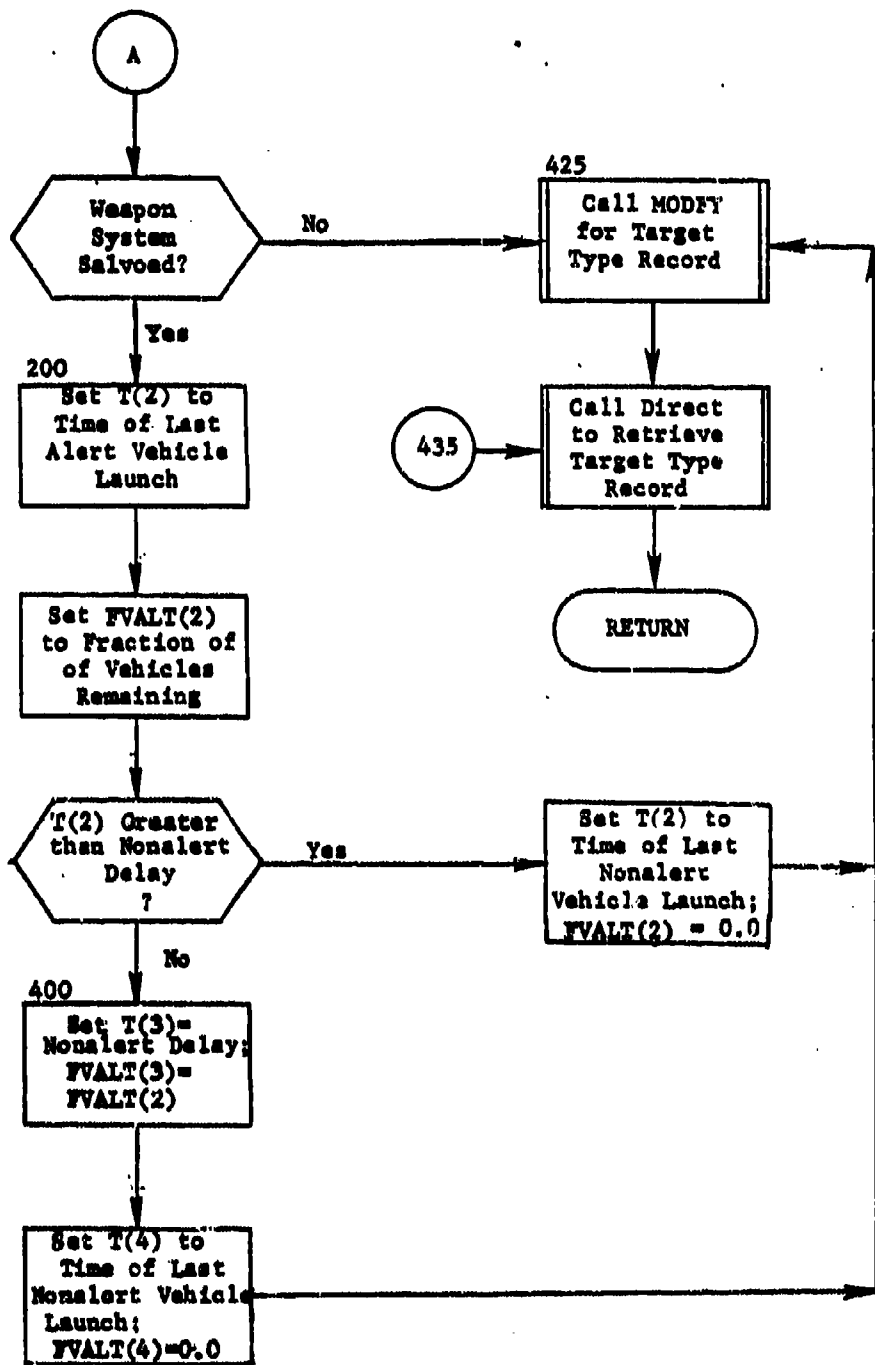


Figure 20. (Part 3 of 3)

CONTENTS OF PAGES 124 AND 125 INTENTIONALLY DELETED

## 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, TARGDE, 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 two 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

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. Individual elements of complexes are not referenced by chain LISTXX; therefore only the representative targets are viewed by the allocator.

Prior to PLANSET termination, each complex has its elements chained under a COMPTG record. The maximum number of complexes has been defined by INDEXER.

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:

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

data card value for exemplar target  
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.

#### 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)

<u>BLOCK</u>	<u>VARIABLE OR ARRAY</u>	<u>DESCRIPTION</u>
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
	YPTSK(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 reference codes for target class headers for defending side
	VALFAC(15)	Stores scaling factors for each target class
INZ	I1, I2, I3	Local storage for INSGET's processing
KEEP		Local storage whose meaning varied during the computation process.
MASK	MASK1	Testing parameter for attribute TASK
	MASK2	Testing parameter for attribute TASK
PBLOK	PBLOK(31,10)	Used in SRTTGT for target designator, number print and in PRINTGP for weapon and group prints
	ARRAY(750)	Temp storage

Table 4. (Part 2 of 2)

<u>BLOCK</u>	<u>VARIABLE OR ARRAY</u>	<u>DESCRIPTION</u>	
PRTZ	LPRTZ(6)	Contains user directed computational options. If,  LPRTZ(1) > 0; Suppress DESIG/Number Print LPRTZ(2) > 0; Suppress TARGET/COMPLEX Print LPRTZ(3) > 0; Ignore grouping LPRTZ(4) > 0; Renormalize target value, nothing else LPRTZ(5) > 0; Suppress FLAG/DESIG Print LPRTZ(6) > 0; Group only	
	SET	RANGEMOD	Fraction of weapon system range for grouping use
		RETARGET	Nonzero for missile retargeting capability
		GCREB(20)	Command and control reliability for regions
	TARCLAS	NCCREL	Number of input values for GCREB
		MAXCLAS	Maximum number of target classes
INCLAS		Number of user selected target classes	
INDESIG(15)		DESIG of exemplar target for selected class	
WEAPON	EXPVAL(15)	Value of selected exemplar target	
	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	
	ICLAS(100)	Zero if TYPE = MISSILE	

### 5.7 Subroutine ENIMOD

**PURPOSE:** Read and store user inputs and control flow of supporting subroutines

**ENTRY POINTS:** ENIMOD (first subroutine called when overlay link PLANS is executed)

**FORMAL PARAMETERS:** None

**COMMON BLOCKS:** CPRIOR, INZ, PRTZ, SET, TARCLA, WEAPON

**SUBROUTINES CALLED:** CINSGET, GRPEM, INSGET, PRINTW, SRTTGT

**CALLED BY:** COP

**Method:**

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

- o PLANSET - the verb that causes execution
- o SETTING - the adverb which introduces a clause to set parameters RANGEMOD, RETARGET or CCREL
- o 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
- o ATTACKERS - the adverb which introduces a clause to select weapon system inventory. Input is a list of values for attribute TYPE
- o 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.
- o ONPRINTS - the clause which permits the user to control the computational flow. Since PLANSET is normally executed many times for one scenario, many of the calculations and prints are redundant. This clause provides a means of suppressing output and, hence, increasing throughput time.

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 ITAR.

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 PLANSET) 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 VSOF) 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 'COMITG'.

See figure 24 for the logic flow within CALCOMP.

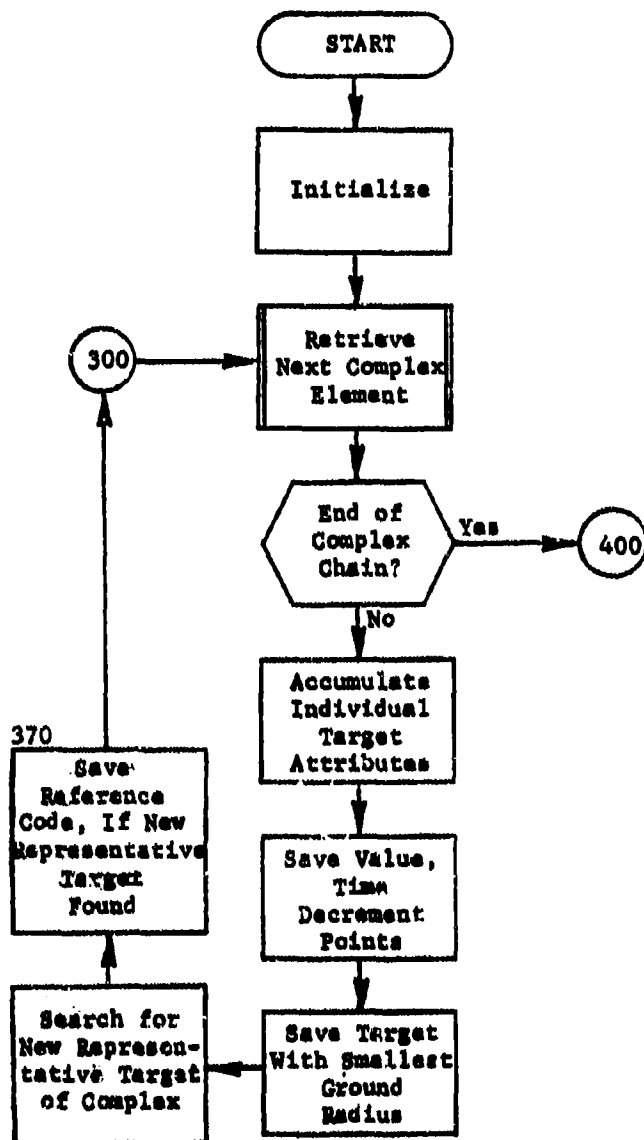


Figure 24. Subroutine CALCOMP (Part 1 of 4)

### 5.10 Subroutine GRPEM

PURPOSE: To form weapon groups

ENTRY POINTS: GRPEM

FORMAL PARAMETERS: None

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

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

CALLED BY: ENTMOD (of overlay link PLANS)

#### Method:

GRPEM forms weapon groups by chaining the user selected weapon type records ('WEPTYT') 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. The grouping order will be performed in the same sequence as the user selected weapon types are.

The attribute IDHOB (which occupies the same position in common C30 as GRPFLG) will control the weapon grouping as follows:

- IDHOB = 0 - Group according to PLANSET algorithms
- 1 - Group alone
- 2-99 - Group with weapons having similar IDHOB.

Tankers are not grouped but are collected and reformatted within sub-routines 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 ('MSBMTG') 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 non-alert bombers then are tested using the criterion  $2 \times \text{RANGEMOD}$ .

Subroutine GRPEM is illustrated in figure 25.

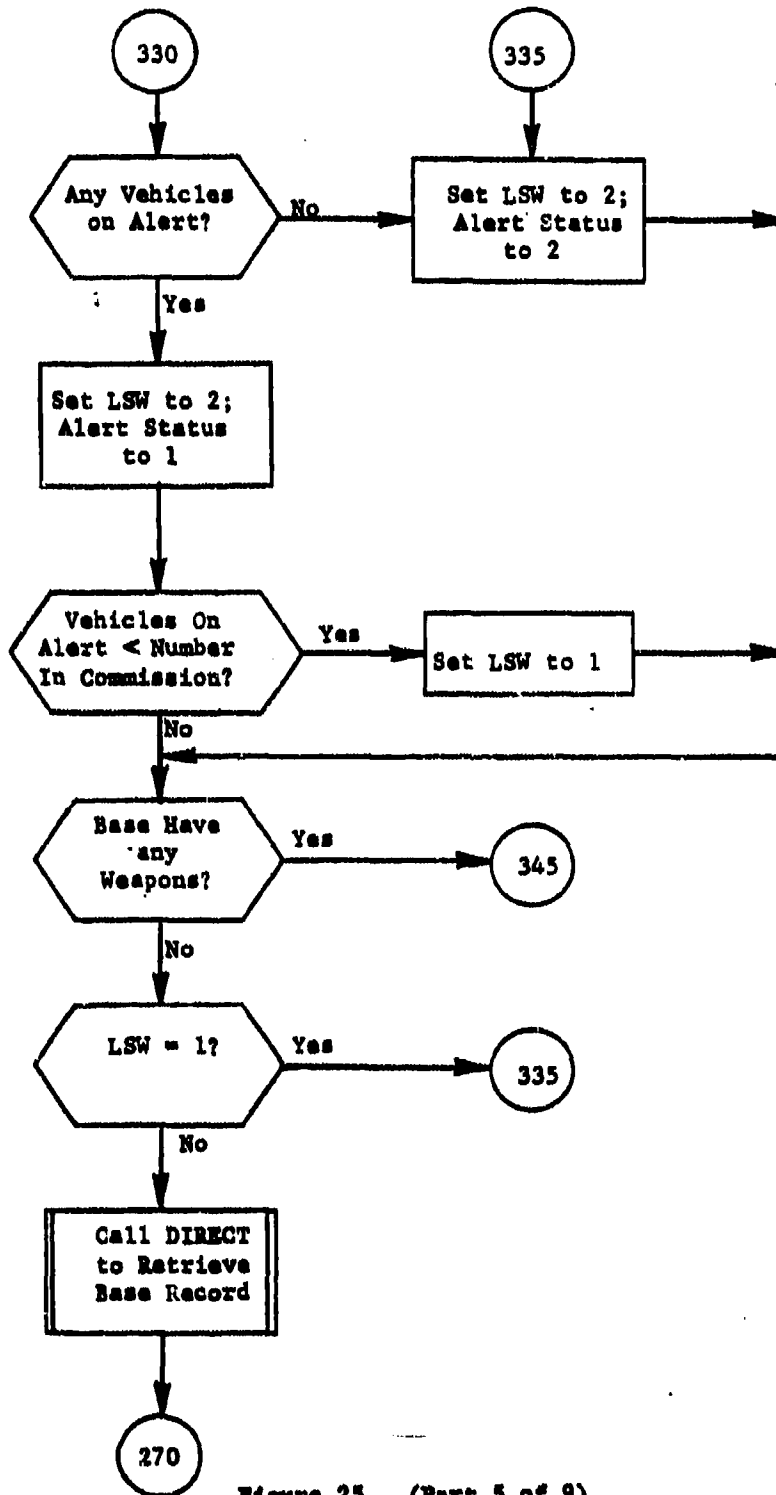


Figure 25. (Part 5 of 9)

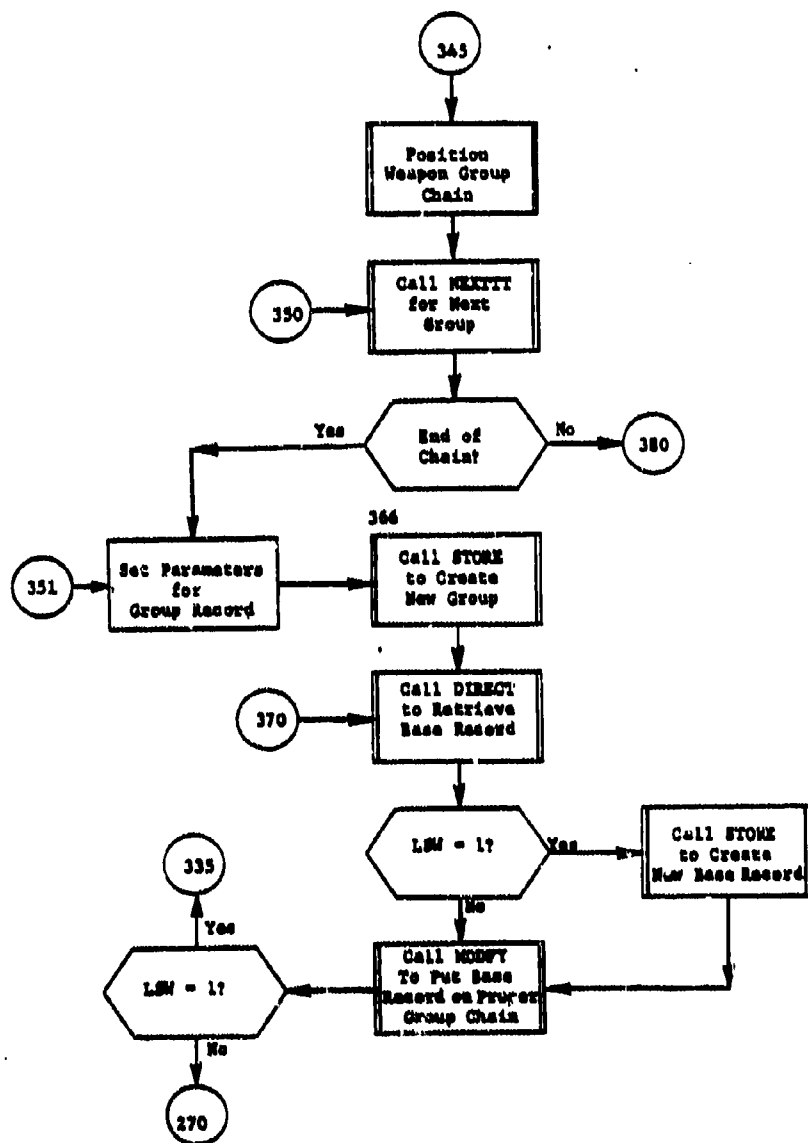


Figure 25. (Part 6 of 9)

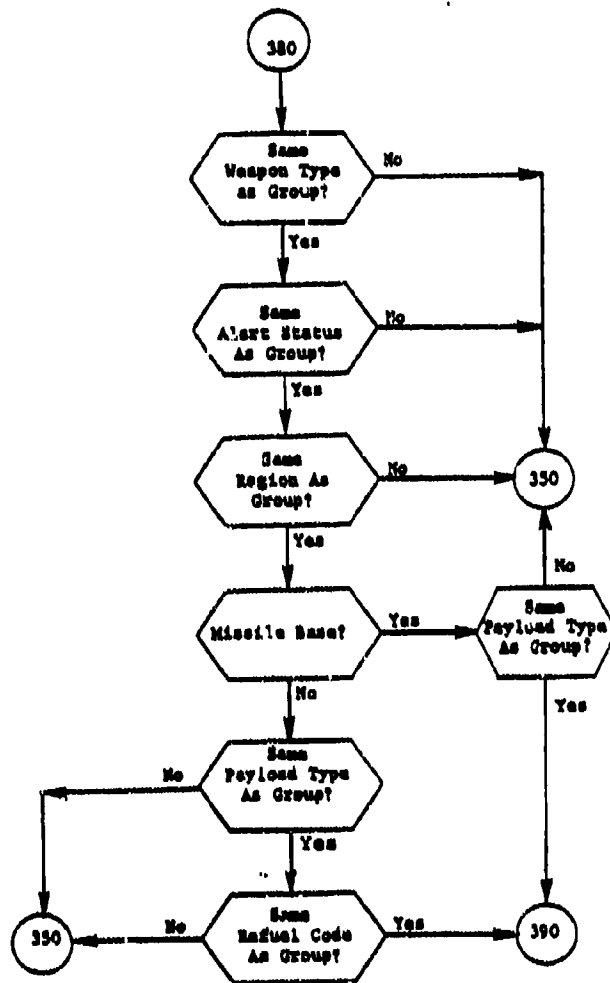


Figure 25. (Part 7 of 9)

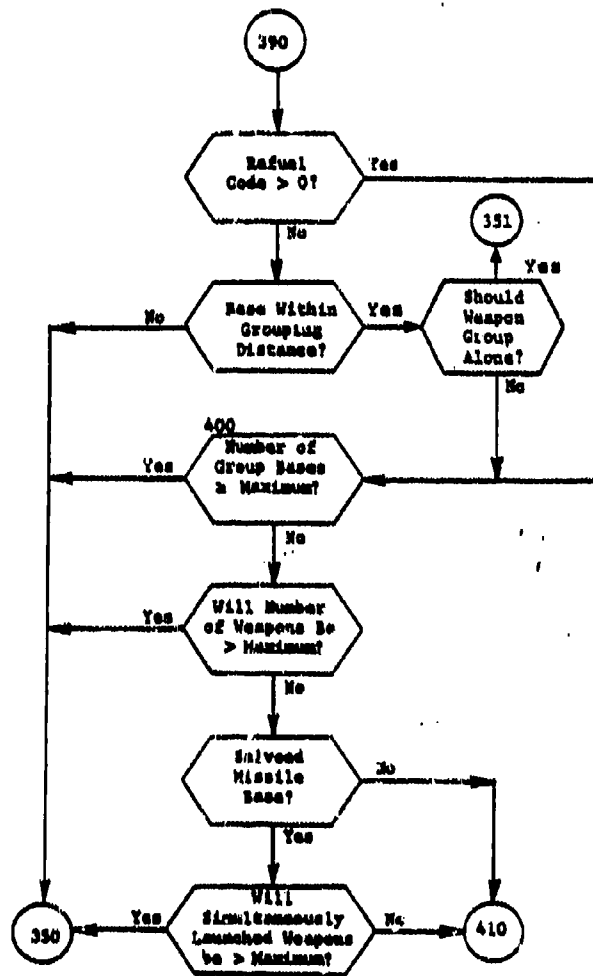


Figure 25. (Part 8 of 9)

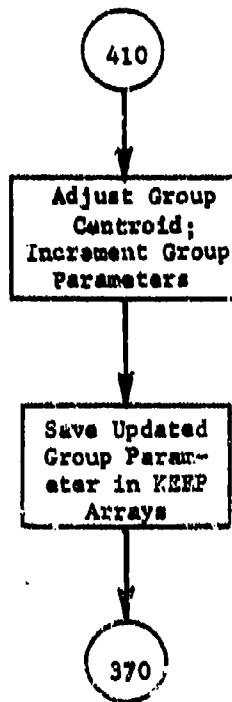


Figure 25. (Part 9 of 9)

### 5.11 Subroutine PRINTGP

PURPOSE: Print standard tables

ENTRY POINTS: PRINTGP, PRINTW

FORMAL PARAMETERS: None

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

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

CALLED BY: ENTMOD (of overlay link PLANS)

#### Method:

This subroutine is called twice: once prior to weapon group and target: processing (Entry PRINTGP) and one after all processing has been completed (Entry PRINTW). Execution through Entry PRINTGP permits the collection and print of the Warhead, ASM, Payload, Weapon Type Characteristics Tables, and weapon systems characteristics for missiles and bombers. Weapon grouping requires payload parameters. Therefore, during the collection process these parameters in addition to being printed will be saved (in /KEEP/ arrays) for use within subroutine GRPEM. After processing, Entry PRINTW causes the printing of weapon groups and complex lists.

Subroutine PRINTGP is illustrated in figure 26.

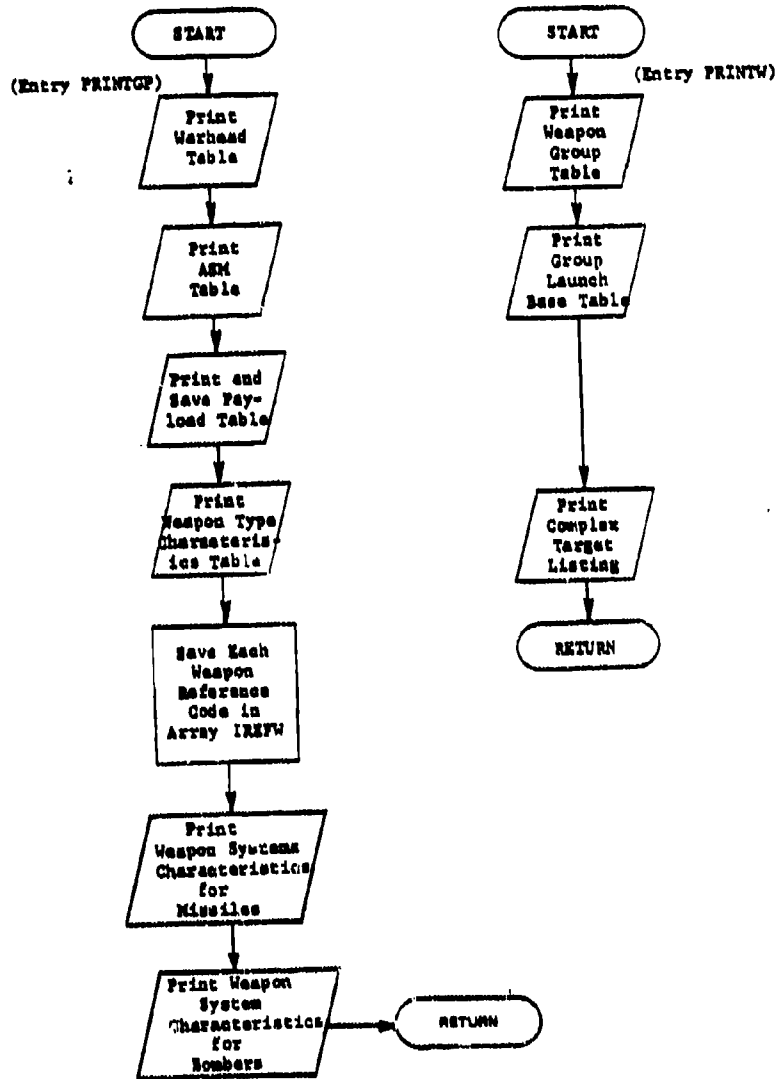


Figure 26. Subroutine PRINTGP

### 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, KEEP, MASK, OOPS, PBLOK, PRTZ, TARCLAS

SUBROUTINES CALLED: CALCOMP, DIRECT, DLETE, HDFND, HEAD, IGET, IPUT, ITLE, KEYMAKE, MODFY, NEXTTT, ORDER, RETRV, 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 0, 1, or 2 if the task inputs are one, two, or three characters long, respectively.

Next the exemplar target DESIG 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 entire target list. During the first pass target values are accumulated for each target class, lethal radius stored for individual targets, and counts of individual targets and number of elements in each complex collected. The number of elements in each complex is stored in array NELEM indexed by attribute ICOMPL. The second pass over the target list will use these counts. Parameters related to optional prints are also stored during this pass. Attribute DESIG is temporarily stored onto scratch file ITEMP if the Target DESIG/Target Number print was selected. If directed (Print option six equals zero), attributes FLAG and DESIG are packed into one word stored in working array KEY1, and eventually sorted using a generalized sort/merge beginning at label 9000.

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

After the first pass over the target list is completed, elements of complexes not selected by the user (through the exemplar target technique)

are transferred from the complex chains they reside on to the simple target chain. When all such targets are transferred, each complex that has one element remaining is also transferred to the single target chain.

The second pass over the target list assigns the final target value, the randomized target number, and forms representative elements of complexes by calling subroutine CALCOMP.

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 (TGTNUMB) is assigned to each target as it is read. TGTNUMB along with the target reference code (TGTREFCO) is temporarily written onto scratch file IF4 for eventual sort and final storage.

Each complex must be processed (by CALCOMP) once after each element in a given complex has its target value assigned. The first pass updates array NELEM (later transferred to KEY1) to define the number of elements in each complex. The second pass updates a second array (KEYC) in the same fashion. By comparing the counts between KEY1 and KEYC, CALCOMP may be executed at the appropriate time.

During the second pass, each target record is written to a random file. The record number and corresponding DESIG are written to a scratch file and the resulting elements are sorted in alphabetical order according to DESIG. The Scratch file is rewound and as each DESIG (now in alphabetical order) is read, the corresponding random record number is obtained and that random file record is printed to produce an alphabetized target sort list.

After the second pass, optional prints are provided by reading files IR and ITEMP. Also, file IF4 is read, sorted on TGTNUMB and the TARCDE records created.

Current data base sizes demand that all interfaces with IDS be kept at a minimum. To accomplish this, common block KEEP is constantly being redefined as working storage arrays that holds pertinent data. Since sizing is open-ended, arrays contained in /KEEP/ must provide for spilling

its contents onto scratch files in the desired sort order. Logic beginning at label 9000 initiates code that performs a basic sort/merge process. Working arrays, are sorted via the standard ORDER subroutine and are merged in sorted order with the information stored on scratch files. This sort/merge process is open-ended and processing is efficient.

Subroutine SRTGT is illustrated in figure 27.

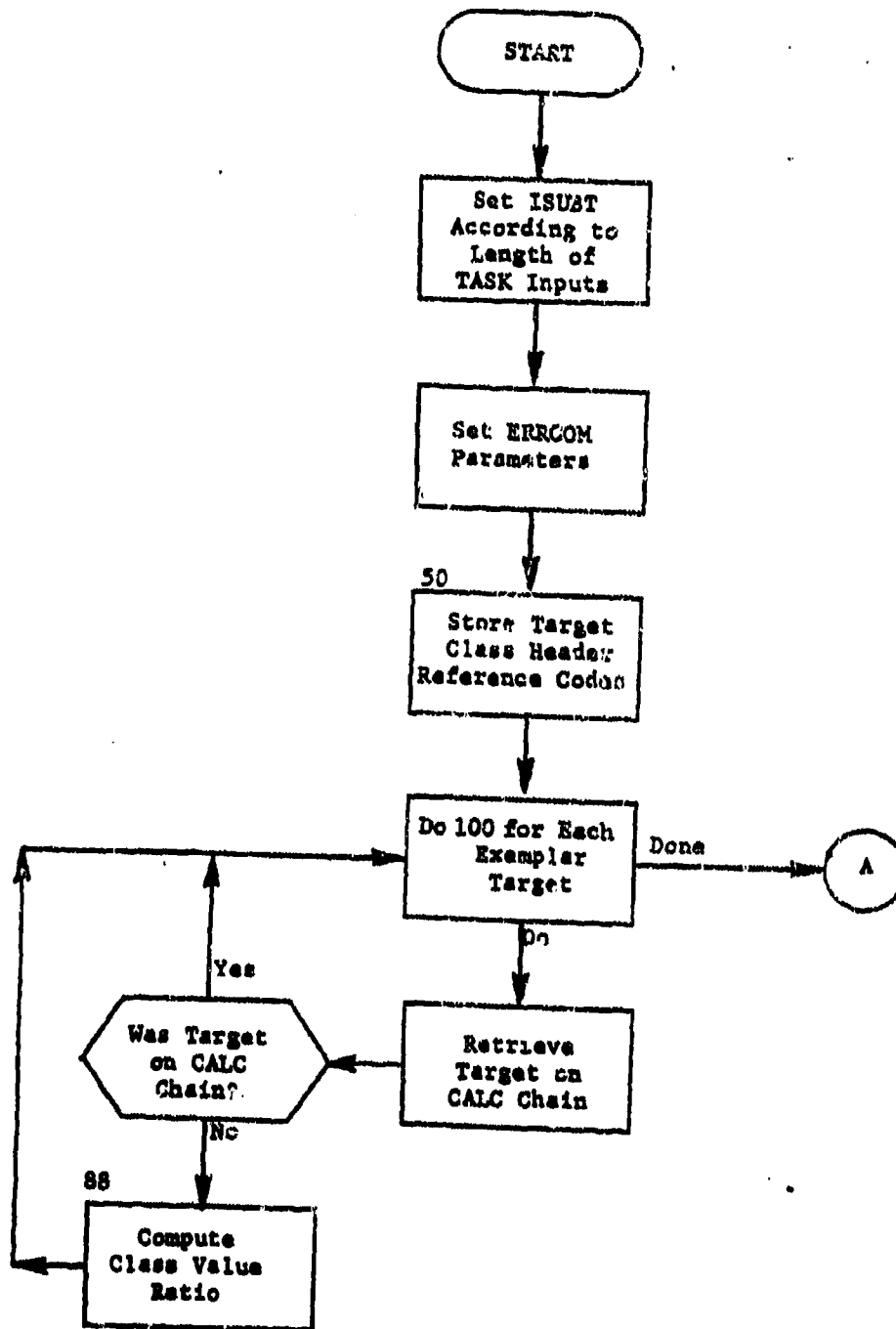


Figure 27. Subroutine SRTTGT (Part 1 of 12)

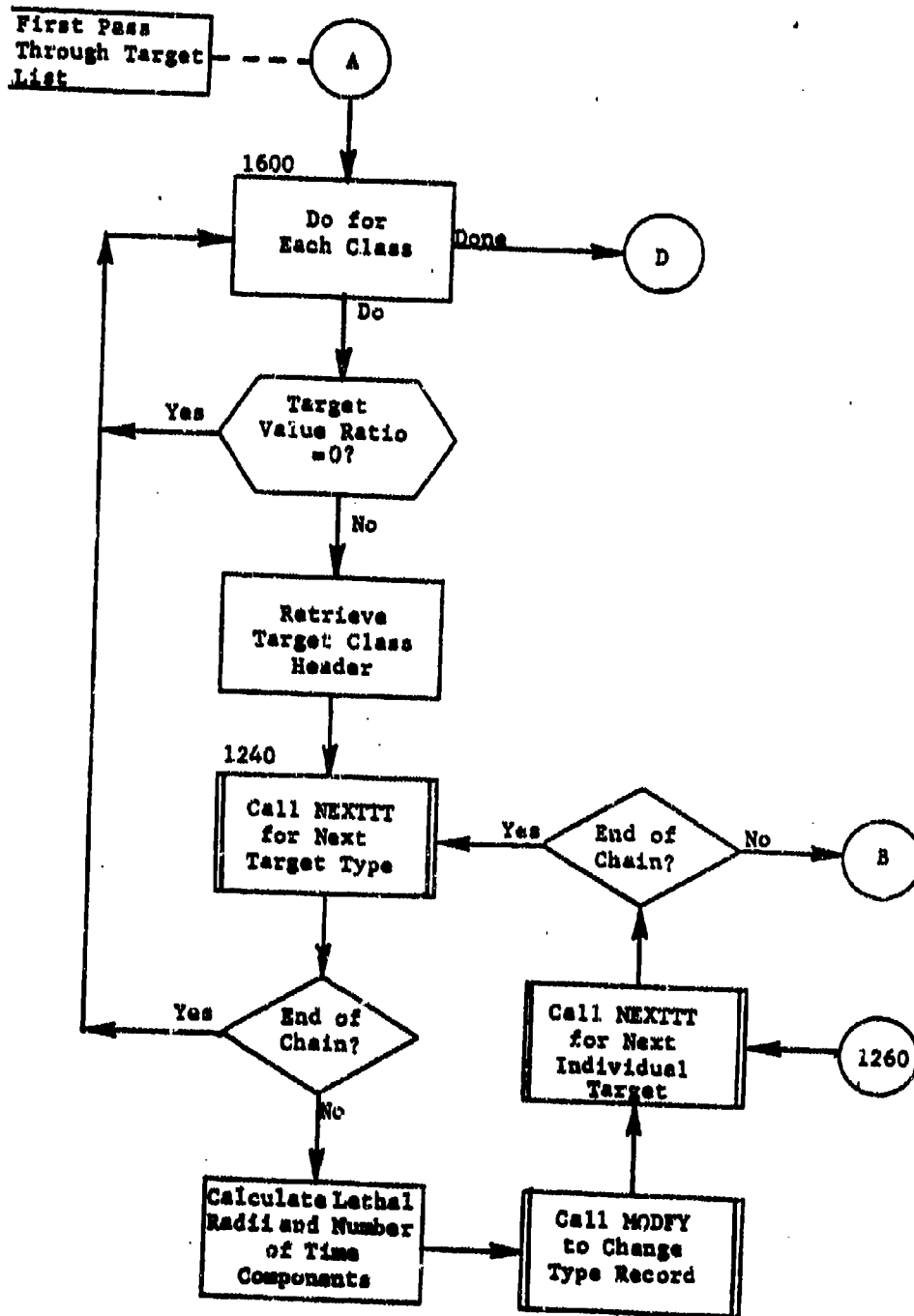


Figure 27. (Part 2 of 12)

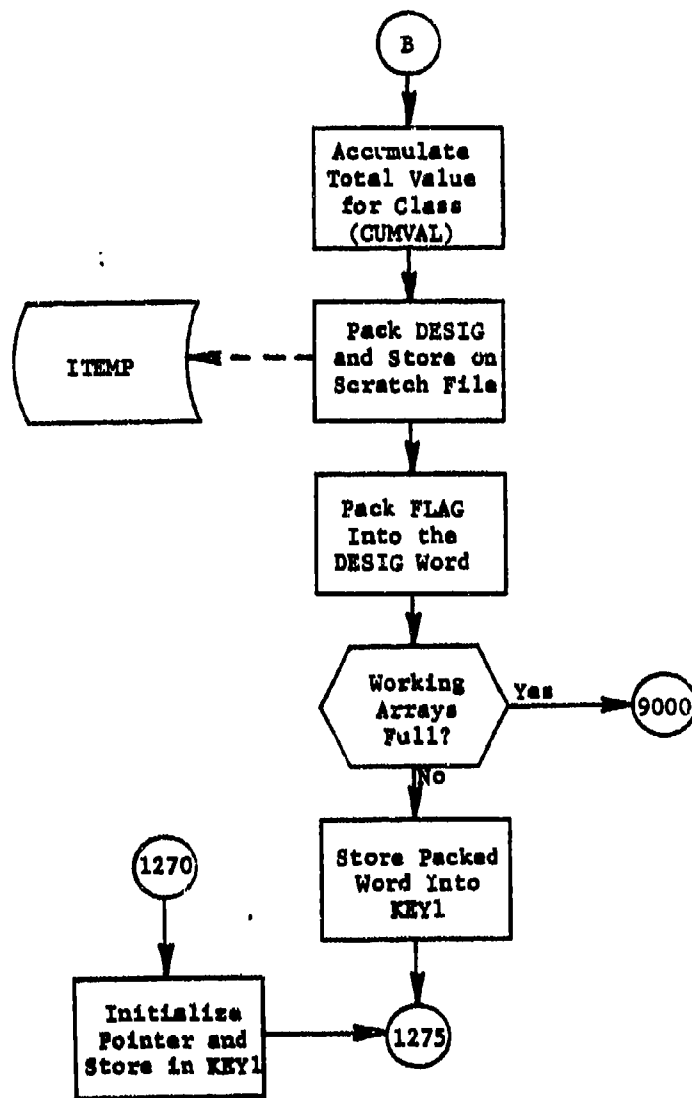


Figure 27. (Part 3 of 12)

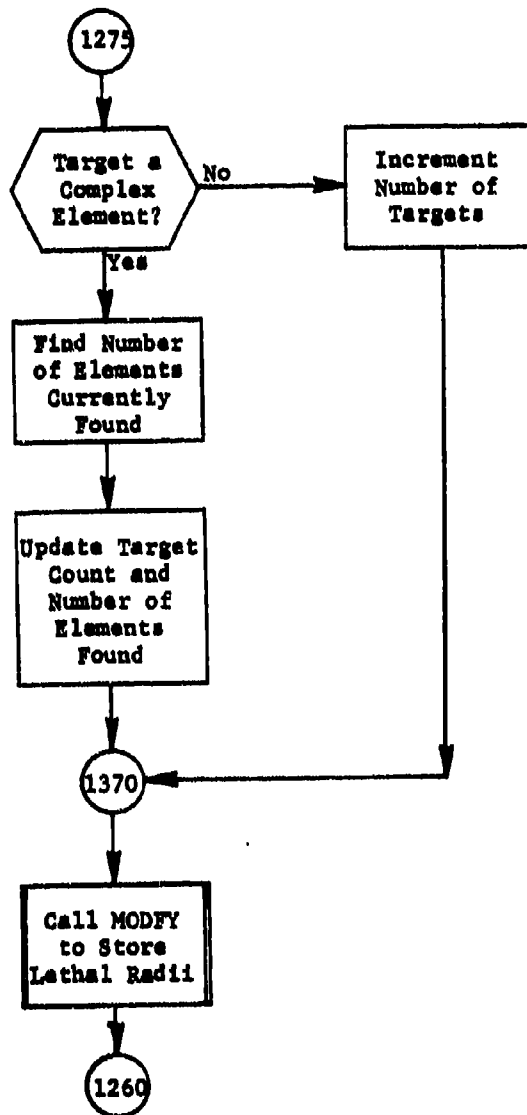


Figure 27. (Part 4 of 12)

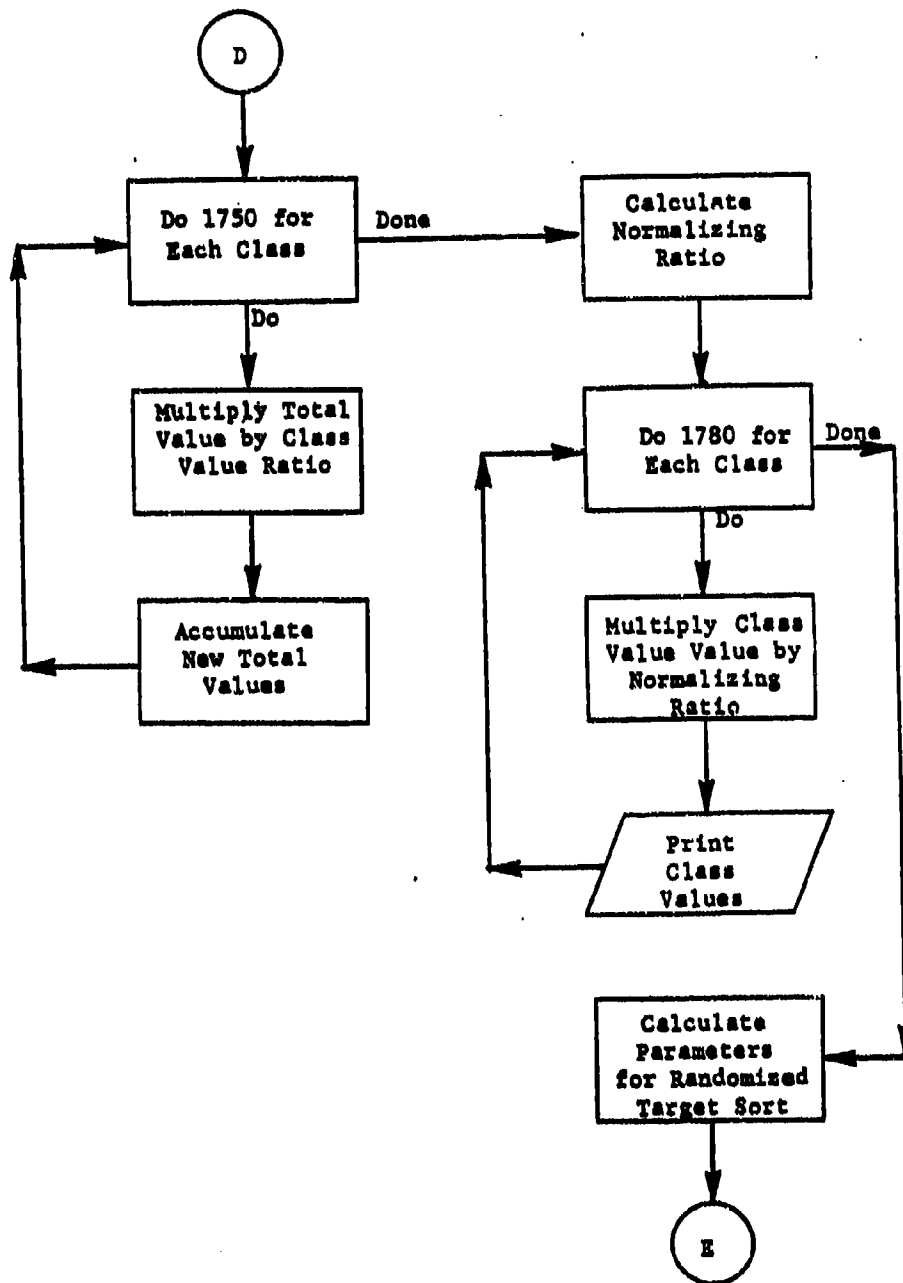


Figure 27. (Part 5 of 12)

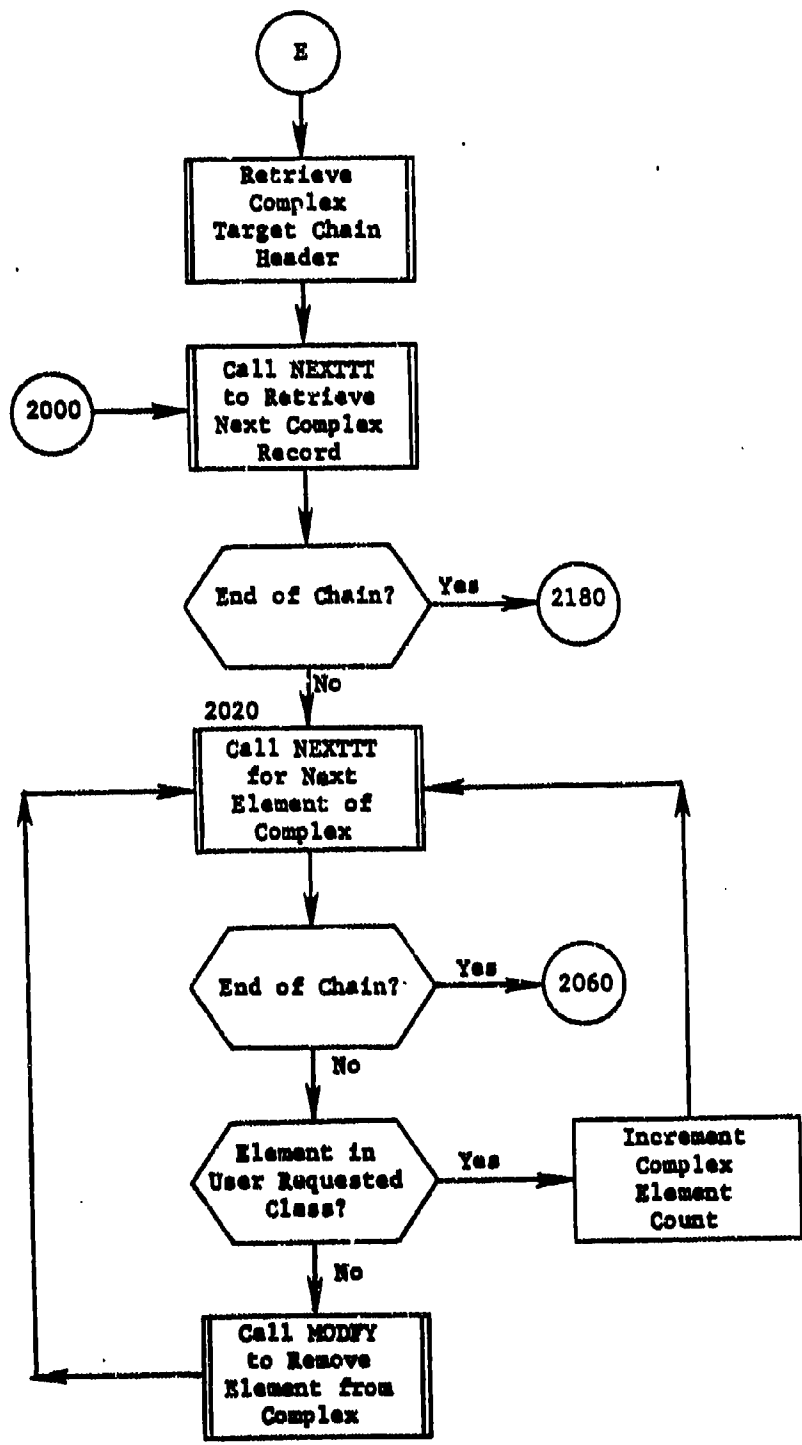


Figure 27. (Part 6 of 12)

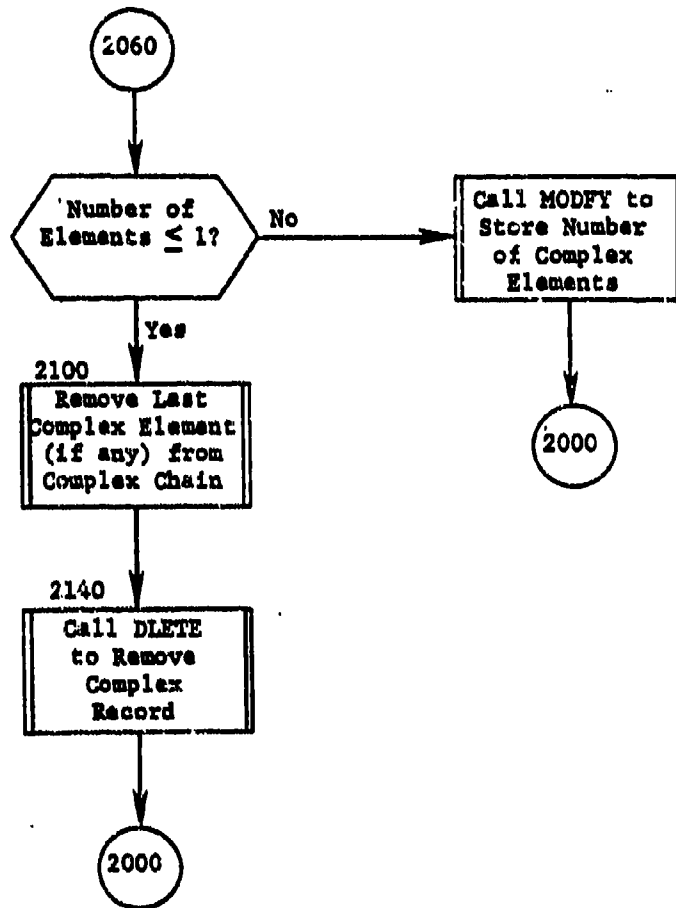


Figure 27. (Part 7 of 12)

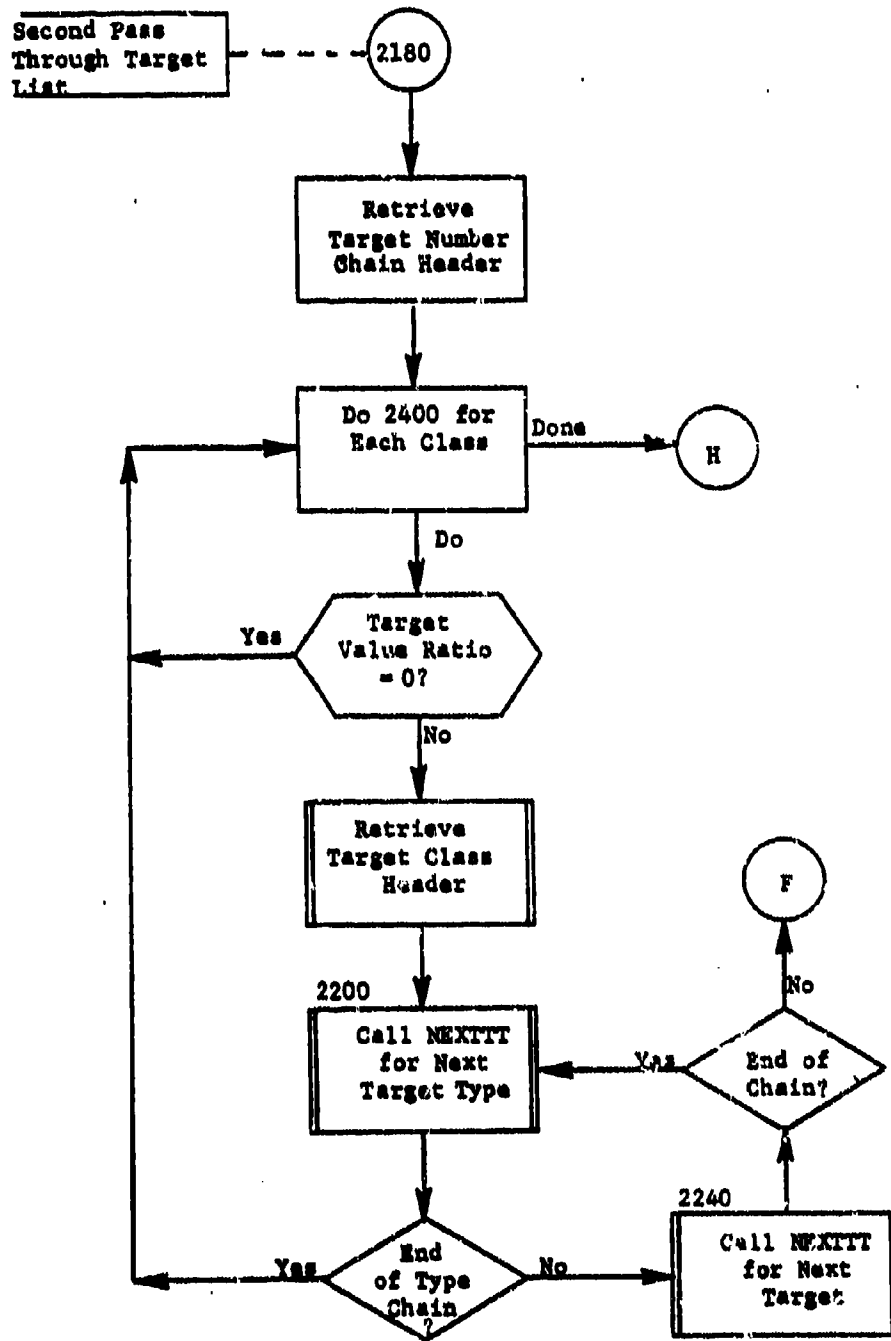


Figure 27. (Part 8 of 12)

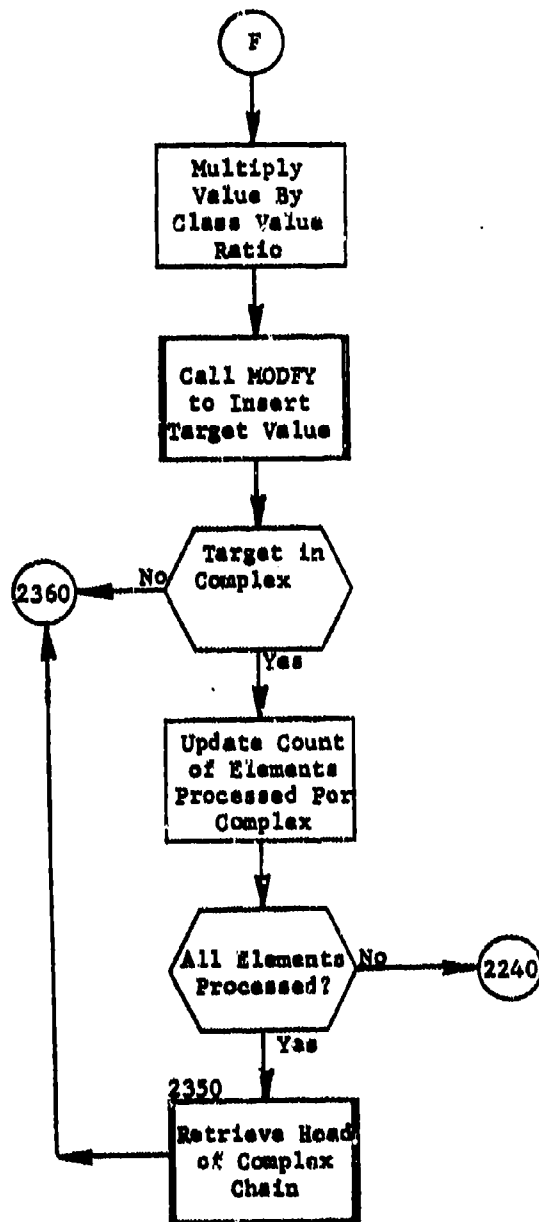


Figure 27. (Part 9 of 12)

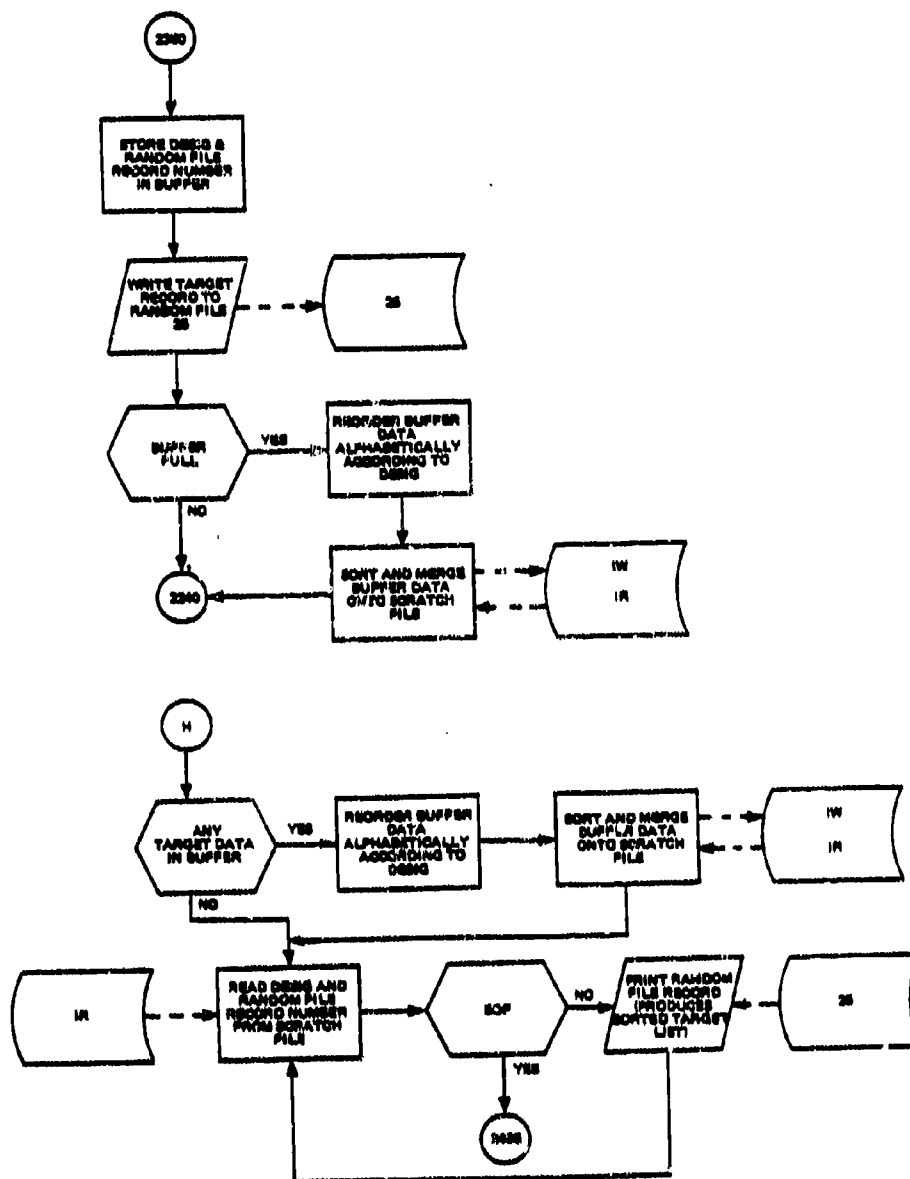


Figure 27. (Part 10 of 12)

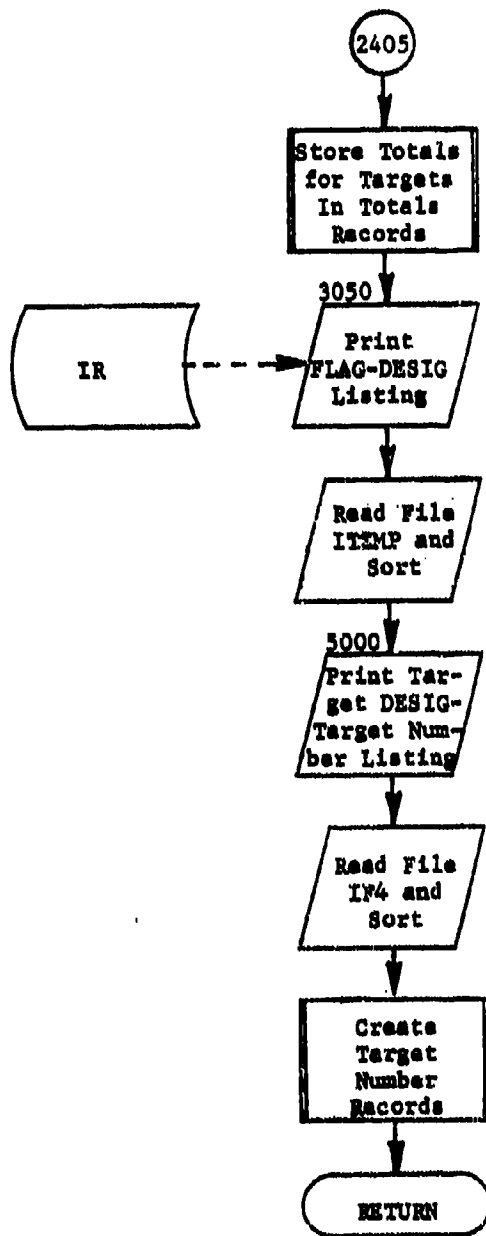


Figure 27. (Part 11 of 12)

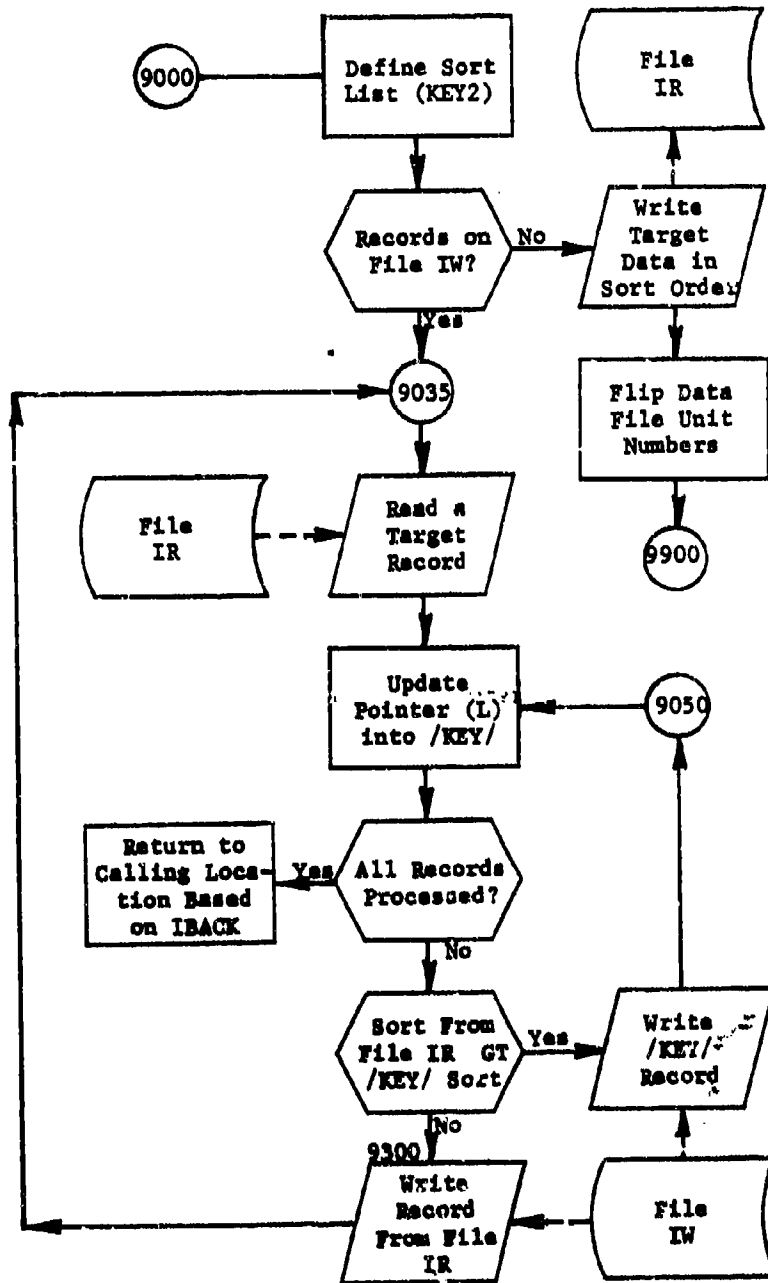


Figure 27. (Part 12 of 12)

THE CONTENTS OF PAGES 171 AND 172 INTENTIONALLY DELETED

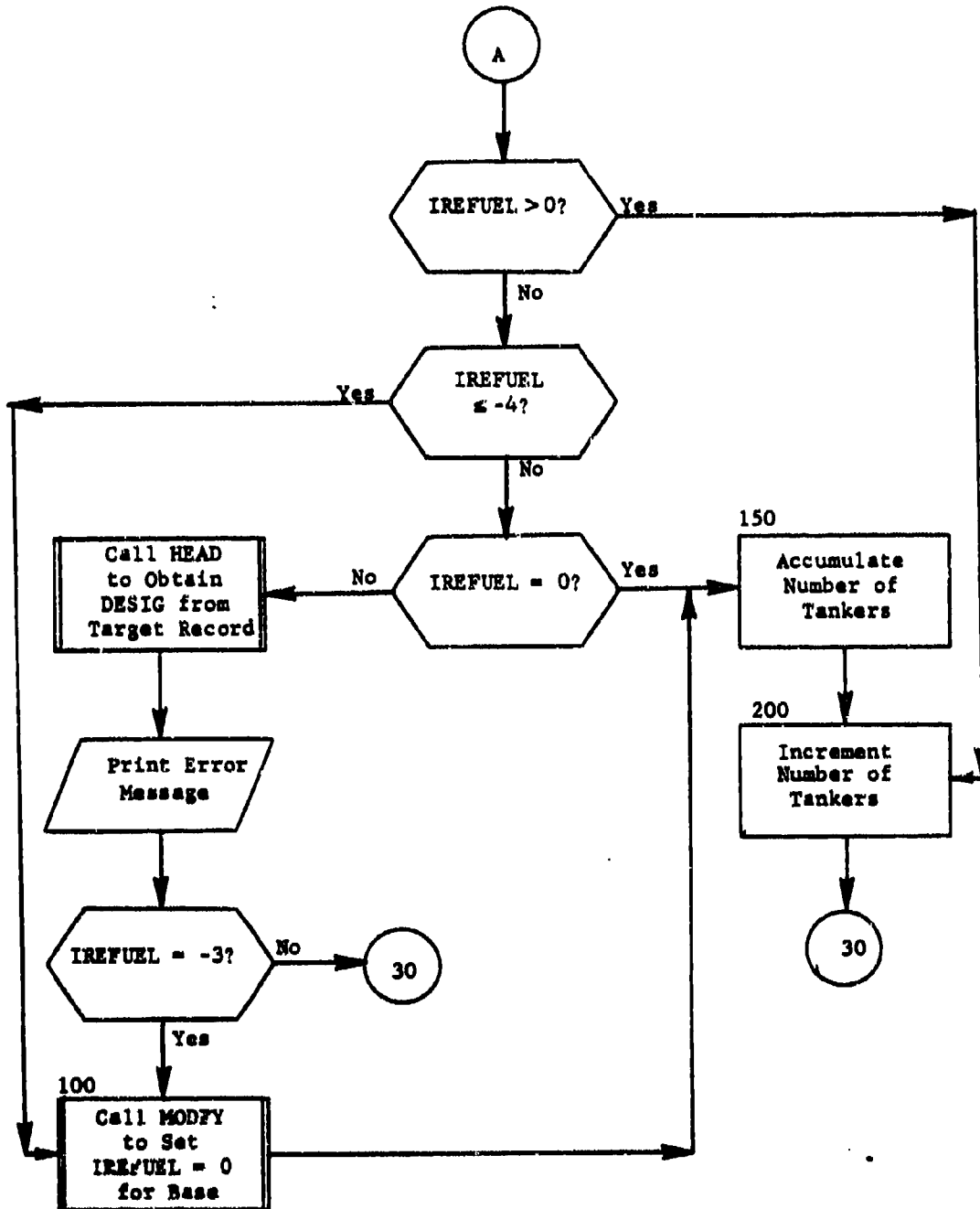


Figure 28. (Part 2 of 2)

CONTENTS OF PAGES 176-178 INTENTIONALLY DELETED

## APPENDIX A

### ANALYTICAL CONCEPTS AND TECHNIQUES

This appendix describes the major analytical concepts, techniques, and algorithms employed within modules JLM, INDEXER, and PLANSET. Topics of discussion consists of explanations of weapon grouping, missile reprogramming, target list preparation, and missile time of flight calculations.

#### A.1 Weapon Grouping

The initial phase of plan development provides an allocation of weapons<sup>\*</sup> to targets. To reduce the amount of processing required during this phase, the offensive weapons are aggregated into "weapon groups" which for the purpose of the allocation, all weapons within a group are treated identically. This phase of processing is then followed by the sortie generation phase during which the specific missile and bomber plans are developed.

Grouping Criteria: On the basis of user input, which specifies the type (TYPE) weapons to be considered, module PLANSET processes the indexed data base and assembles the individual missile and bomber units (items in classes MSLWEP and BMBWEP) into weapon groups.

A weapon group is defined as a set of weapons which are assigned to delivery vehicles that are located in the same geographic area and have like characteristics. Specifically, to be in the same group, these weapons must be of the same type; i.e., the attribute TYPE\*\* must have the same value; they must have the same alert status (alert or non-alert), originate in the same geographic region, and have the same payload. Bombers must have the same refueling index (IREFUEL). In order for missiles or nonrefueling bombers to be grouped, they must lie within a geographic area which, for alert weapons, has a radius equal to a certain percentage of the range of the weapon. This percentage is a parameter RANGEMOD specified by the user for input to program PLANSET. If RANGEMOD is not specified, it is assumed to be 15 percent. The RANGEMOD value

\* A weapon is defined here as a warhead plus the characteristics of its delivery vehicle.

\*\* A single set of delivery vehicle characteristics is associated with all weapons of a given type. These characteristics include attributes (for all weapons) ALRTDLY, CEP, FUNCTI, LCHINT, NLRDLY, PLABT, RANGE, REL, SIMLUN, SPEED (for missiles only), ALRTDB (if IREP greater than zero), IREP, NRPST, PDES, PFFP, PINC, RANGMIN, TOFMIN, IRECMO, PRABT, RANGDEC, RANGREF, and SPDLO.

used for alert weapons is automatically doubled for nonalert weapons (to reduce the proliferation of groups). Under this criterion, it is appropriate to think of the weapons of a given group as being capable of attacking the same set of targets.

If the weapons are to be used exclusively against naval targets (a player option), all the weapons in the group must have the same value for the attribute PKNAV (the single shot kill probability for these weapons against targets of class NAVAL).

Group Centroid: As a new base is added to a group, the latitude and longitude of the group centroid are adjusted so that the final values reflect the true group centroid. That adjustment is effected as follows.

Let NG = The number of bases included in the group prior to this addition  
LSTLAT = Latitude of centroid before addition  
LSTLONG = Longitude of centroid before addition  
LAT = Latitude of the weapon being added  
LONG = Longitude of the weapon being added

Then for the new centroid latitude (NEWLAT),

$$\text{NEWLAT} = \frac{(\text{NG} * \text{LSTLAT}) + \text{LAT}}{\text{NG} + 1}$$

To determine the new centroid longitude (NEWLONG) an intermediate quantity (GLONG) is calculated. If  $\text{GLONG} < 0$ ,  $\text{NEWLONG} = \text{GLONG} + 360$ ; otherwise  $\text{NEWLONG} = \text{GLONG}$ . GLONG is calculated as follows:

1. If  $-180 \leq (\text{LSTLONG} - \text{LONG}) \leq 180$  then  $\text{GLONG} = [(\text{NG} * \text{LSTLONG}) + \text{LONG}] / [\text{NG} + 1]$
2. If  $(\text{LSTLONG} - \text{LONG}) > 180$  then  $\text{GLONG} = [(\text{NG} * [\text{LSTLONG} - 360]) + \text{LONG}] / [\text{NG} + 1]$
3. If  $(\text{LSTLONG} - \text{LONG}) < -180$  then  $\text{GLONG} = [(\text{NG} * \text{LSTLONG}) + (\text{LONG} - 360)] / [\text{NG} + 1]$

Basic Yield (Bombers): One of the composite characteristics calculated for a bomber group is its basic yield per gravity bomb. That value is obtained as follows. Define the following variables:

NOBOMB1 Number of bombs of type 1  
NOBOMB2 Number of bombs of type 2  
YIELD1 Yield of type 1 bomb  
YIELD2 Yield of type 2 bomb

## A.2 Missile Reprogramming

Each missile type in the data base has an associated attribute IREP which indicates its reprogramming capability. Missiles may be retargetable, for instance, if other weapons in the squadron have been destroyed before launch, during launch, or in powered flight. The reprogramming capabilities considered within the Plan Generator are:

- No reprogramming capability (IREP=1)
- Reprogramming for not in commission (IREP=2)
- Reprogramming for destruction before launch (IREP=3)
- Reprogramming for failure through launch (IREP=4)
- Reprogramming for failure through powered flight (IREP=5)

During QUICK plan generation, this reprogramming capability is exercised only if the user specifies a RETARGET option in program PLANSET. The effects of missile reprogramming during plan generation are to: 1) decrease the number of vehicles per squadron; 2) reduce the DBL probability for alert vehicles to zero for those missiles which reprogram for this failure mode; and 3) increase the reliability factor for reprogrammable missiles. In computing replacement values for these parameters, the data base value associated with the following attributes is considered.

- PINC: Probability that the missile is in commission
- ALRTDB: Probability of DBL for alert vehicles
- PLABT: Probability of a launch abort
- PPFF: Probability of failure during powered flight

Table 5 shows the method of calculating replacement values for each level of reprogramming capability. To illustrate the reprogramming calculations, let N be the original number per squadron, R the original reliability for any missile squadron, and S be the probability of survival before launch. If N' is the reduced number of weapons, R' the increased squadron reliability resulting from reprogramming calculations, and S' the modified survival probability, N'\*R'\*S' will still equal N\*R\*S. The new values, however, reflect the probability, with retargeting, of striking the N' highest priority targets to be assigned to the squadron. For example: for a non-SLBM (submarine-launched ballistic missile) missile squadron with attributes IREP=3, PINC=.8, ALRTDB=.1, PLABT=.2, and PFFP=.3, and a number per squadron of 30, the new attribute values assigned (see table 5) are:

$$\begin{aligned} \text{Number per squadron} &= \text{PINC}(1-\text{ALRTDB}) (N) \\ &= (.8)(1-.1)(30) = 21.6, \text{ truncated to } 21 \\ \text{New ALRTDB} &= 0 \\ &= (1-\text{PLABT})(1-\text{PFFP}) \\ \text{Reliability} &= (.8)(1-.2)(1-.3) = .56 \end{aligned}$$

Had reprogramming not been considered, the values would have been:

$$\begin{aligned} \text{Number per squadron} &= 30 \\ \text{ALRTDB} &= .1 \\ \text{Reliability} &= .8(1-.2)(1-.3) = .448 \end{aligned}$$

Table 5. Computations for Reprogrammable Missiles

Reprogramming Capability Index	New number per squadron (N = original number)	New ALRTDB for this type	Reliability for this type
IREP = 1	N	ALRTDB	PINC *(1-PLABT)*(1-PFFP)
IREP = 2	PINC * N	ALRTDB	(1-PLABT)*(1-PFFP)
Non SLBM IREP = 3	PINC *(1-ALRTDB) *N	0	(1-PLABT)*(1-PFFP)
SLBM	N	#	#
Non SLBM IREP = 4	PINC *(1-PLABT) *(1-ALRTDB)	0	1-PFFP
SLBM	PINC *(1-PLABT)*N	ALRTDB	1-PFFP
Non SLBM IREP = 5	PINC *(1-ALRTDB) *(1-PLABT)*(1-PFFP)*N	0	1
SLBM	PINC *(1-PLABT) *(1-PFFP)*N	ALRTDB	1

# Reprogramming for destruction before launch is not applicable to submarine-launched weapons since the destruction of one launch site destroys all remaining missiles in the squadron.

### A.3 Target List Preparation

The information provided to the Plan Generator consists of information on the target system which is to be attacked and on the available weapon systems which have been provided to deal with the target system. The weapon allocator (module ALOC) receives its targets as a shuffled target list; that is, a list of targets that are arranged in a random order. To be discussed are: target categories, time dependent target, target value, methods of complexing, and target shuffling (or randomizing).

Target Categories: From a computational point of view, QUICK considers two categories of targets: simple targets, and complex targets. Target numbers are assigned to all simple targets, and complex targets in classes 1-15 for both sides, one side at a time. A simple target is a single data base item with a single unique geographical location.

The second category of target, the complex target, allows the Plan Generator to deal with targets consisting of several elements and to treat them as a single simple target during the weapon allocation phase. Complex targets are formed by the Weapon/Target Identification subsystem (INDEXER) and consist of target elements (up to 99 data base items) in which each element is separated from some other element in the complex by a distance not greater than one-half the sum of the lethal radii of the two elements from a given weapon yield, considering the vulnerabilities for each of the elements. The complexing yield assumes a default value of one-megaton or any user input value. Also, at user discretion complexing yield may be obtained through query of VN and associated table look ups (see subsection under target complexes). Under either criterion, the complex target is input to program ALOC as a single element target with characteristics which are representative of the complete complex. The procedures used in identifying and describing this representative target element are discussed later in this chapter (see subsection Complex Targets).

With the above simplifications, the method of allocation used by program ALOC can be essentially the same for both types of targets.

Time-Dependent Target Value: The relative value of the targets considered during plan generation is established on the basis of two sets of input data supplied by the user. In the data base each potential target is assigned a value (VALUE) which establishes its relative worth within its assigned class. Then, the user provides data, for input to module PLANSET, which establish the target's value relative to all other potential targets in the game base (see Target Value, this appendix).

Since the relative strategic worth of a target may degrade over time (e.g., the value of a missile launch site before and after launch), the time dependence of target value must be considered in developing the attack plan. In QUICK, this relationship is established on the basis of data supplied by the user and included in the data base. The user can specify up to five separate time components which represent specified fractions of the total target value. For each of the five components, the user specifies the time (in hours) at which the value changes, T(I), and the fraction of the target value that is removed at that time, FVALT(I).

In using these data, the system (module PLANSET) automatically assumes a standard uncertainty in times specified. Figure 30 illustrates the time dependence of target value. As a result of the "value escaped" method of target damage calculation and the launch interval timing considerations, target value cannot increase at any time.

Target Value: The Plan Generator allocates weapons so as to maximize the target value destroyed. To accomplish this, the relative importance or value of the targets to be considered must be established. These target values reflect the major strategic objectives of the war plan which is to be generated. They must, therefore, be established by the user within the context of a specific game scenario.

The QUICK system uses a two-step procedure to input the user judgmental data required for target value calculations.

1. In the data base, each potential target is assigned a value calculated to reflect its relative worth within its assigned class.
2. To generate a specific plan, the user must also provide data to the Plan Generator (module PLANSET) which determine the relative value of the target classes, and hence all targets, for the current plan.

For the data base, a reasonably good judgment can be made of the relative values of the targets within each target class (such as missile, bomber, urban/industrial, or naval classes). The values may be based, for instance, on relative population or industrial importance for urban/industrial targets. For missile and bomber classes, the user will probably select target values which take into account each weapon's effective

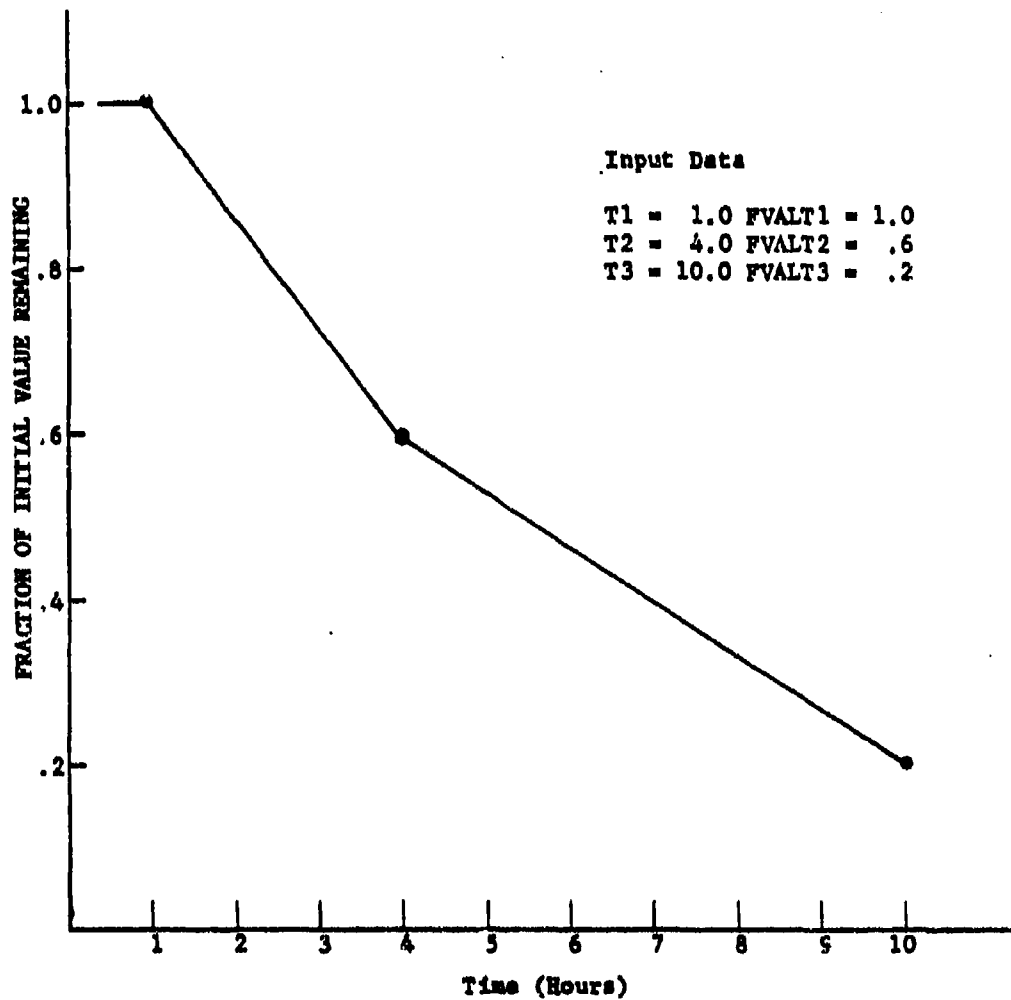


Figure 30. Time Dependent Target Value Curve

megatonnage, range, and CEP. Each potential target in the data base must be assigned the attribute VALUE, and the value associated with this attribute must establish the target's relative worth within the class to which it is assigned.

The value input is completed with data cards input to module PLANSET. Here, when generating a specific plan, the user must input his judgment as to the relative values of the target classes. This is communicated to the Plan Generator by the selection of an exemplar (or typical) target from each target class which is to be included in the plan. To that exemplar target, the user assigns a new value (NEWVAL). NEWVAL, then, is used as follows:

$$\text{Let VALCLASS}(J) = \frac{\text{NEWVAL for the exemplar target in class } J}{\text{VALUE for the exemplar target in class } J}$$

and CUMVAL(J) = the sum of the VALUES of all the targets in class J

Then the total value of the targets in class J is

$$\text{CUMVALF}(J) = \text{CUMVAL}(J) * \text{VALCLASS}(J)$$

These target class values are then scaled so that the sum of all target values is 1,000, thus facilitating comparative analyses of differing plans. This scaling is done by setting

$$\text{SUMVALX} = 1000 / \left( \sum_{j=1}^{15} \text{CUMVALF}(J) \right)$$

and establishing the final value factor for all items in class J by

$$\text{VALFAC}(J) = \text{SUMVALX} * \text{VALCLASS}(J)$$

VALFAC(J), then, is the multiplier used to derive the new value for each target in class J from its data base value, VALUE; i.e., the target's value for this plan = VALFAC(J) \* VALUE.

The QUICK value scheme allows the user to reflect a relative judgment between the worth of two specific targets in different classes, rather than to decide the total distribution of VALUE which is to be apportioned between those two classes. This judgment is much more analogous to the usual strategic decisions. It is generally easier to specify the relative worth of Moscow vs. an SS-9 missile site than it is to specify the fraction of value that will be associated with urban/industrial targets vs. missile sites. In order to better illustrate this exemplar value scheme, a simple set of four targets is shown in table 6. In this table, one exemplar target from each class is assigned a value. The final calculated values used in the allocator sum to 1,000 and maintain the original data base ratios within each class. Also, the ratio of values between the exemplar targets is the same as the ratio between the user inputs.

Table 6. Sample Exemplar Target Value Calculation

TARGET CLASS	TARGET NAME	DATA BASE VALUE (VAL)	USER INPUT EXEMPLAR VALUE (NEWVAL)	FINAL CALCULATED VALUE
U/I	Moscow	80	16	400
U/I	Kiev	60		300
Missile	Ipich	5	10	250
Missile	Aag	1		50
Total				1,000

The intermediate calculations used to derive the final calculated values above are:

	U/I CLASS	MISSILE CLASS
VALCLASS	$16/80 = .2$	$10/5 = 2$
CUMVAL	$80 + 60 = 140$	$5 + 1 = 6$
CUMVALF	$.2(140) = 28$	$2(6) = 12$
VALFAC*	$.2(25) = 5$	$2(25) = 50$

\*Where  $SUMVALX = 1000/(28 + 12) = 25$

**Complex Target:** A complex target consists of target elements in which each element is separated from some other element in the complex by a distance not greater than one-half the sum of the complexing lethal radii of the two elements. A complexing weapon yield plus the vulnerabilities of each target element determines the lethal radii. A single global user input weapon yield (a default of one-megaton is assumed) may be used for complex determination. That is, for each target element, the same weapon yield will be used for lethal radius determination. A second optional method of lethal radii calculation is permissible which employs table lookups of hard coded values. Through observation of the VN of each target element a yield is obtained from the table, then an adjusted VN is calculated and, finally, from the yield and adjusted VN tables are entered and lethal radii obtained. After lethal radii determination the collection of target elements and setting of target attributes (see below) is identical.

The potential target list input to module ALOC reflects the complex target as a single element. The target attributes for this representative target, calculated in module PLANSET, are derived from the target data associated with the individual elements of the complex. The largest target radius associated with any element of the complex is assigned as the radius\* (TGTRAD) of the representative target. Similarly, the maximum value of TARDEF (local bomber defense potential) is assigned to represent the complex. The target value (VALUE) and the number of terminal defense missile interceptors (NTINT) assigned each element are accumulated and their totals assigned to the representative element. MINKILL (the minimum kill probability required) and MAXKILL (the maximum kill probability desired) are weighted (by VALUE) averages of the element attributes. The time dependence of the value of the complex, which is due to the time components of its elements, is approximated by at most five time components. That approximation is accomplished as follows.

First, the list of time components is checked for equal values. If any are found, the corresponding values are added together, and all but one of the equal components are removed from consideration, along with any zero components. If the number of remaining entries does not exceed five, the time dependence of the complex is approximated by these time components. Otherwise, an elimination procedure to reduce the number of entries to five is performed. For 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 number of entries is reduced by one. If more than five entries still remain, the above accumulation process is repeated, until no more than five time components remain.

---

\*This technique represents an oversimplification. However, computing an adjusted radius, based on the geographic locations and dimensions of each target element, would not necessarily be an improvement. In order

to provide a significantly more accurate treatment, a much more detailed analysis would be required of each complex target, which should take into account the yield and accuracy of the available weapons as well as the number, hardness, and geographic distribution of target elements.

Air lethal radius for each target in the complex must be determined for a uniform height of burst over the complex. Air lethal radius associated with simple targets are calculated assuming an optimal height of burst based on the target's vulnerability. Since a complex may be comprised of targets with differing vulnerabilities, it's necessary to define one height of burst over the entire complex. Accordingly, the height of burst for the complex is defined as the optimal scaled height of burst associated with the hardest target in the complex. Using this defined air height of burst, the air lethal radius for each target in the complex is computed.

Similarly, the hardness components (VULN1, VULN2) and the corresponding fractional value (FVULN1) which represent the complex are determined by first taking, for each target of the complex, its VALUE, FVULN1, the number of hardness components (1 or 2), and the lethal radius corresponding to each hardness. The complement of FVULN1 is found to represent the second hardness component. If either fractional value is nonzero, it is multiplied by VALUE to obtain 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 miles) and radii belonging to soft targets. The average lethal radius, weighted by the actual value at the corresponding hardness, is calculated from both hard and soft targets for those radii, and the result (HHARD or HSOF) is assigned to the complex. Similarly, the actual value at each hardness (VHARD or VSOF) is accumulated. If there are no hard targets (i.e., VHARD = 0), FVULN1 for the complex is set to 1; otherwise the fraction of actual value for hard targets (VHARD/VTOT) is assigned to FVULN1. This FVULN1, then, and the corresponding number of hardness components are assigned to the complex.

The index number (INDEXNO) and the target designator code (DESIG) associated with the representative target will normally be the INDEXNO and DESIG assigned to the first member of the complex (i.e., the first element of the complex encountered when processing the game data base). The user may, however, establish criteria for selecting the representative INDEXNO and DESIG (a control feature used in RISOP development). The procedure for exercising this option are outlined in the Users Manual (module PLANSET).

**Target Shuffling:** During the allocation phase of plan generation, the rate of allocation for each weapon group is monitored as the targets are processed. To prevent these rates from being biased by a large number of similar targets considered consecutively, the basic target list is shuffled.

Since similar targets appear together in the data base (by class and type), target shuffling randomizes the order in which various types of targets are encountered. Thus the rate of allocation provides a good estimate of whether a group is being over-allocated or under-allocated. The algorithm used to achieve the required shuffling is accomplished in the following manner.

Consider the target indices (I) as equally spaced points on a circle, with targets in a particular class occurring consecutively. If the Ith point is displaced along the circle to the index

$$\frac{1}{2} (3 - \sqrt{5}) C * (I) \text{ mod } C$$

where C is the number of points on the circumference of the circle, the result will be the desired distribution. To accomplish the corresponding reordering of the discrete list of targets, each index must be multiplied by

$$\frac{1}{2} (3 - \sqrt{5}) N$$

where N is the number of elements in the list, and then reduced by modulo N. A direct application of this procedure, however, will result in some cases with the same final index being assigned to more than one element.

Therefore, the following algorithm is used by program PLANSET to assign new indices to the elements of the list in such a way that the above criterion is satisfied and each index from 1 to N is assigned exactly once.

Let L be the greatest integer such that  $L \leq \frac{1}{2} (3 - \sqrt{5}) N$

where N is the total number of targets.

Let P = J = L,

where P = the beginning index of the current cycle, and  
J = the index number currently being assigned.

As each list element is processed, J is replaced by J + L to obtain the next index number. If J becomes >N, J is replaced by J - N. If J becomes = P, a new cycle is to be started; 1 is added to P and to J, and the procedure continues as before.

CONTENTS OF PAGES 192 THROUGH 198 INTENTIONALLY DELETED

DISTRIBUTION

<u>Addressee</u>	<u>Copies</u>
<b>CCTC Codes</b>	
C124 (Reference and Record Set) . . . . .	3
C124 (Stock) . . . . .	6
C126 . . . . .	2
C313 . . . . .	1
C314 . . . . .	7
C630 . . . . .	1
<b>DCA Code</b>	
205 . . . . .	1
<b>EXTERNAL</b>	
Chief, Studies, Analysis and Gaming Agency, OJCS ATTN: SFD, Room 1D935, Pentagon, Washington, DC 20301 . . . . .	2
Chief of Naval Operations, ATTN: OP-654C, Room BE781 Pentagon, Washington, DC 20350 . . . . .	2
Commander-in-Chief, North American Air Defense Command ATTN: NPXYA, Ent Air Force Base, CO 80912 . . . . .	2
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 Technical Information Center, Cameron Station, Alexandria, VA 22314 . . . . .	12
	42

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (when data entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER CSM MM 9-77, Volume II	2. GOVT. ACCESSION NO. AD-A085814	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) THE CCTC QUICK-REACTING GENERAL WAR GAMING SYSTEM (QUICK), Program Maintenance Manual, Weapon/Target Identification Subsystem		5. TYPE OF REPORT & PERIOD COVERED
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR (s) Dale J. Sanders, Paul F. M. Maykrantz, Jim M. Herrin, Edward F. Berrson		8. CONTRACT OR GRANT NUMBER (s) DCA 100-75-C-0019
9. PERFORMING ORGANIZATION NAME & ADDRESS System Sciences, Incorporated 4720 Montgomery Lane Bethesda, Maryland 20014		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME & ADDRESS Command and Control Technical Center Room BE-685, The Pentagon Washington, D.C. 20301		12. REPORT DATE 1 June 1977
		13. NUMBER OF PAGES 188
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		15. SECURITY CLASS. (of this report) UNCLASSIFIED
		15a. DECLASS/DOWNGRADING SCHEDULE
16. DISTRIBUTION STATEMENT (of this report)		
DISTRIBUTION STATEMENT (of the abstract entered in block 20, if different from report) approved for public release; distribution unlimited.		
17. SUPPLEMENTARY NOTES		
18. 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.		

DD FORM 1473

EDITION OF 1 NOV 65 IS OBSOLETE

201

UNCLASSIFIED

GH-3

SECURITY CLASSIFICATION OF THIS PAGE (when data entered)

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (when data entered)

20. ABSTRACT (Continued)

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 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) are published by the Command and Control Technical Center (CCTC), Defense Communications Agency (DCA), The Pentagon, Washington, DC 20301.