

AD-A038 893

COMMAND AND CONTROL TECHNICAL CENTER WASHINGTON D C  
PROGRAM MAINTENANCE MANUAL. VOLUME II. WEAPON/TARGET IDENTIFICA--ETC(U)  
NOV 76

F/G 15/6

UNCLASSIFIED

CCTC-CSM-MM-9-74-VOL-2-4

NL

↓ OF 1  
AD  
A038893



END

DATE  
FILMED

5-77

AD A 038893



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

11

IN REPLY  
REFER TO: C314

11 Nov 76

12 36p.

TO: RECIPIENTS (6)

SUBJECT: Change 4 to Program Maintenance Manual, CSM MM 9-74,  
Volume II, Weapon/Target Identification Subsystem  
Change # 4.

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

*J. Douglas Potter*  
J. DOUGLAS POTTER  
Assistant to the Director  
for Administration

35 Enclosures  
Change 4 pages

14  
CCTC - CSM-MM-9-74-Vol-2-4

DDC  
APR 25 1971  
C

AD NO. \_\_\_\_\_  
DDC FILE COPY,

RTIS	White Section	<input checked="" type="checkbox"/>
SEC	Buff Section	<input type="checkbox"/>
UNANNOUNCED		<input type="checkbox"/>
JUSTIFICATION		<input type="checkbox"/>
BY	file 56 on	
DISTRIBUTION/AVAILABILITY CODES		
Dist.	AVAIL. and/or SPECIAL	
A		

DISTRIBUTION STATEMENT A  
Approved for public release;  
Distribution Unlimited

409658



EFFECTIVE PAGES - NOVEMBER 1976

This list is used to verify the accuracy of CSM MM 9-74 Volume II after change 4 pages have been inserted. Original pages are indicated by the letter 0, change 1 pages by the number 1, change 2 pages by the number 2, etc.

<u>Page No.</u>	<u>Change No.</u>	<u>Page No.</u>	<u>Change No.</u>
Front Cover	1	366-367	4
Title Page	1	368	1
ii-iv	1	369-371	4
v	0	372-376	0
vi-vii	1	377	4
viii-xii	0	378-397	0
xiii	1	398-399	1
xiv	0	400-401	0
1-12	0	402-405	4
13	2	406	1
14-21	0	407-410	0
22-23	3	411	4
24-28	0	412-498	0
29-31	2	498.1	1
32-114	0	498.2	0
115-117	2	498.3-498.4	1
118-306	0		
307-308	4		
309	3		
310-315	0		
316	1		
317	0		
318	4		
319	0		
320	4		
321-322	0		
323	4		
324-330	0		
331	4		
332	0		
333	1		
334-337	0		
338-339	1		
340	4		
341	4		
342	4		
343-344	3		
345-358	0		
359	4		
360-365	0		

## SECTION 7. PROGRAM PLANSET

### 7.1 Purpose

PLANSET prepares the data files required by the Plan Generator to develop a plan for one side. It forms weapon groups, prepares the target list, computes and normalizes the class value factors, calculates the representative attributes for complex targets, and creates the WINFILE (weapon input file) and TINFILE (target input file) required by program PREPALOC. PLANSET also creates the TARFILE (target designation input file) to be used by program PLANOUT. Note that program PLANSET must be processed by program DECLARES before being executed. Program PLANSET calls overlay programs SHUFFLE, PLANSA, SHUFF1 and TARLST.

### 7.2 Input Files

The principal input to PLANSET consists of the indexed data base, INDEXDB or INMODDB, generated by program INDEXER or program DBMOD of the Data Assembly subsystem. Several user-option data cards are also accepted, which specify:

- a. The command and control reliability factor for each region in the plan
- b. A request that missile retargeting be used for all missiles with reprogramming capability
- c. A range multiplier RANGEMOD to be used when determining whether a weapon is sufficiently within range of a weapon group to be added as a group member
- d. The SIDE for which a plan is to be generated
- e. Names of each attacking weapon type
- f. The maximum absolute difference in DBL probability MAXDBL that is allowed between the first and last weapons of a weapon group
- g. The class name and value of an exemplar target for each class in the current plan
- h. The values of the attribute TASK to be given priority when assigning the lead target of a complex
- i. The alphabetic portion of the values of the attribute DESIG to be given priority when assigning the lead target of a complex
- j. Print options allowing a print of the target list weapon group list, and/or complex target list.

### 7.3 Output Files

Program PLANSET prepares the TINFILE (target input file) and WINFILE (weapon input file) to be used in program PREPALOC. PLANSET also creates the TARFILE (target designator input file) to be used in program PLANOUT. The TINFILE (see table 16) contains a 39-word block of descriptive information for each target to be considered in the current plan. The targets are placed on the TINFILE in a random order which facilitates evaluation of the allocation process used in program ALOC. The target input file (TINFILE) includes three types of targets:

- a. Simple target: One target element
- b. Complex target: several target elements either exactly collocated or within the lethal radius of a single weapon (a one-megaton weapon is considered) 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 far enough apart that each target element must be treated as an independent aim point.

Each complex target is represented on the TINFILE in an aggregated form representing the total value of the complex as required by ALOC. This aggregated representation on TINFILE is paralleled by auxiliary detailed target data on WINFILE which includes a specific representation of each target element as a separate simple target. Similarly each multiple target is represented on TINFILE by a single representative target (of the appropriate multiplicity) as required by ALOC. This representative target is also paralleled by a list of specific coordinates for each target element in the auxiliary target data on WINFILE.

Subroutine VLRADP will be called twice for each hardness component. The two calls will each obtain the radius for either the air or ground burst. For each hardness component the two radii will be packed into one word of the ITD array. The upper 18 bits of the word will contain the air burst lethal radius expressed in hundred thousandths of a nautical mile. The lower 18 bits will contain the ground burst radius expressed in the same units. The air burst lethal radius of each complex element will be recalculated in subroutine CALCOMP. The recalculation will be based on the optimal height of burst associated with the hardest element of the complex.

Table 17 shows the final format of the WINFILE. It lists 17 separate blocks of information. WINFILE is also used to temporarily store the DESIG and FLAG of each target processed. (The column "BLOCK" includes both a general descriptor which specifies the type data being output and a common block designator (shown parenthetically) which indicates the associated PLANSET common blocks. The variable/array names shown are those associated with this data in subsequent programs of plan generation. The first block

Table 17. (Part 7 of 7)

<u>BLOCK TYPE</u>	<u>MAXIMUM LENGTH/ ACTUAL LENGTH</u>	<u>VARIABLE OR ARRAY</u>	<u>DESCRIPTION</u>
		IBASE(2)	Index of second base
		BASELAT(2)	⋮
		BASELONG(2)	⋮
		IPAYLOAD(2)	
		VONBASE(2)	
		IBASE(3)	
		⋮	
		(NBASE)	
Tanker /C3/	600/(NTANKBAS X 12)	INDEXTK	Tanker index number
		TKLAT	Tanker base latitude
		TKLONG	Tanker base longitude
		IREFTK	Refuel area assigned to tanker base
		NPSQNTK	Number of tankers per squadron or base
		NALRTK	Number of alert tankers per base
		SPEEDTK	Tanker speed in knots
		DLYALTK	Alert delay
		DLYNTK	Nonalert delay
		TTOS	Total time on station
		ITYPETK	Tanker type index
		RANGE	Tanker range

contains header information which includes the run identification together with information on the size of the succeeding blocks. The second block contains the breakpoint tables which reflect the indexed structure of the game data base. The next six blocks contain point information which describes bomber routes, corridors, depenetration corridors, recovery bases, and directed refuel areas. The next block contains the descriptions of the air defense zone boundaries. The tenth block contains information on complex and multiple targets which augments the target data contained on the TINFILE: it contains information on each complex and multiple target element. The next six blocks provide data which define the offensive force. This includes the warhead, ASM (air-to-surface missile), and payload tables and weapon system information by region, type, group, and base. The final block contains information on the tanker units which support the bomber force.

The TARFILE is first generated by subroutine CALCOMP and contains three words consisting of DESIG, ICOMPLEX, and VULN for each representative target of those complexes where a non-representative element has a VULN harder than that of the representative target. Overlay four (program TARLST) reads this data and, then, sets TARFILE for a new definition.

The final TARFILE writes (see table 18) a three-word header block of descriptive information regarding the contents of the file, a 7771-word block of "hash" ordered "page" and target designator string pointers, and a 3512-word block for each "page" of the TARGET LIST (TARLIST). That is to say the TARLIST is divided into pages. Each page contains strings of target designators (DESIGs) which have the same "hashed" address. The strings of DESIGs, as well as the list of available space in the page, are maintained in a "Last-In-First-Out" (LIFO) manner. Each target DESIG item in a string has associated with it the following data: FLAG, TARDEFHI, CNTRYLOC, TASK, INDEXNO, LAT, LONG, H1, and CNTRYOWN. Each list item in a string is contained in four words.

An option output tape (TGTSORT) may be requested for saving the sorted target or complexes for processing by other user programs.

#### 7.4 Concept of Operation

Seven intermediate files are used by program PLANSET in processing the required weapon and target data.

- a. Intermediate group file (LTGRP): Information pertaining to each weapon in a group and to each tanker is written onto the intermediate group file, LTGRP. For each weapon group element, LTGRP contains a six-word record in the format of array GRPX (IGRPX) in common block /C3/. For each tanker squadron (item entry in ICLASS three) a 12-word record is written in the format of array TANK (LTANK) in common block /C3/, preceded by a one-word code. The code is -1 for tankers which have been preassigned to refuel

Table 18. TARFILE Format

<u>BLOCK TYPE</u>	<u>MAXIMUM LENGTH/ ACTUAL LENGTH</u>	<u>VARIABLE OR ARRAY</u>	<u>DESCRIPTION</u>
Header /C5/	1 (each) 3 total	MXSCAT	Maximum length of the SCAT array
		MAXPPG	Maximum length of a TARLST page
		NIOPGS	Number of TARLST pages on the TARFILE
Hash Table /C5/	7771	SCAT(7771)	Target designator "hash" ordered "page" and string pointer table
TARLST Pages /C5/	Variable in 3516-word segments	TARLST(3516)	TARLST page of target designator strings.

areas in the data base, and -4 for those which are to be automatically allocated. The end-of-file is signalled by a single word, ENDGRO

- b. Intermediate target file (LTTGT): All target data are stored on the intermediate target file LTTGT. For each individual target (except elements of multiple targets), LTTGT contains a 36-word record corresponding to common block /TD/. For each multiple target, there is a record consisting of the 36 words corresponding to common block /TD/ followed by, for each member target, eight words in the format of array MLTX in common block /MLTX/. The target data end with a dummy target record containing XXXXXX in word 2. Following the target data, the GRP (IGRP) array from common block /GROUP/ is written to enable the core storage it occupied to be reused. It is read back into core after the target data has been processed (in subroutine TGTSORT).
- c. Sort files LSRTA and LSRTB: When the complex target print is requested (optional) two sort files, LSRTA and LSRTB, are used to sort the data pertaining to complex targets. Each record is a 35-word copy of one column (J constant) of the array ICPLX(I,J) contained in common block /C12/. The records are sequenced by complex number. The end of the file is signified by a dummy record containing 999999 in word 35.
- d. WTCL Data File WTCLDATA: This random access file contains the weapon type characteristics list data for all types requested in the game. One record for each type is output at the appearance of each WTCL item. The file is read and printed at the end of the processing of the third overlay.
- e. TARLST Scratch Files IN and IOUT: These two scratch files are used for intermediate processing of the TARLIST pages by program TARLST, the fourth overlay of program PLANSET.

PLANSET (flowchart shown in figure 65) begins processing the indexed data base by calling subroutine FILFSP and reading in the breakpoint tables from the end of INDEXDB (or INMODDB). The input data cards then are read and their information stored. The TASK and DESIG priority cards are read by overlay 1 of PLANSET, program SHUFFLE. Program PLANSA, overlay 2 of program PLANSET, continues the normal target and weapon processing. Program SHUFF1, overlay 3 of program PLANSET, initializes TINFILE and WINFILE, writes the point data to WINFILE, and adds target records to both output files. Program TARLST, overlay 4 of program PLANSET, is then called to create the TARFILE.

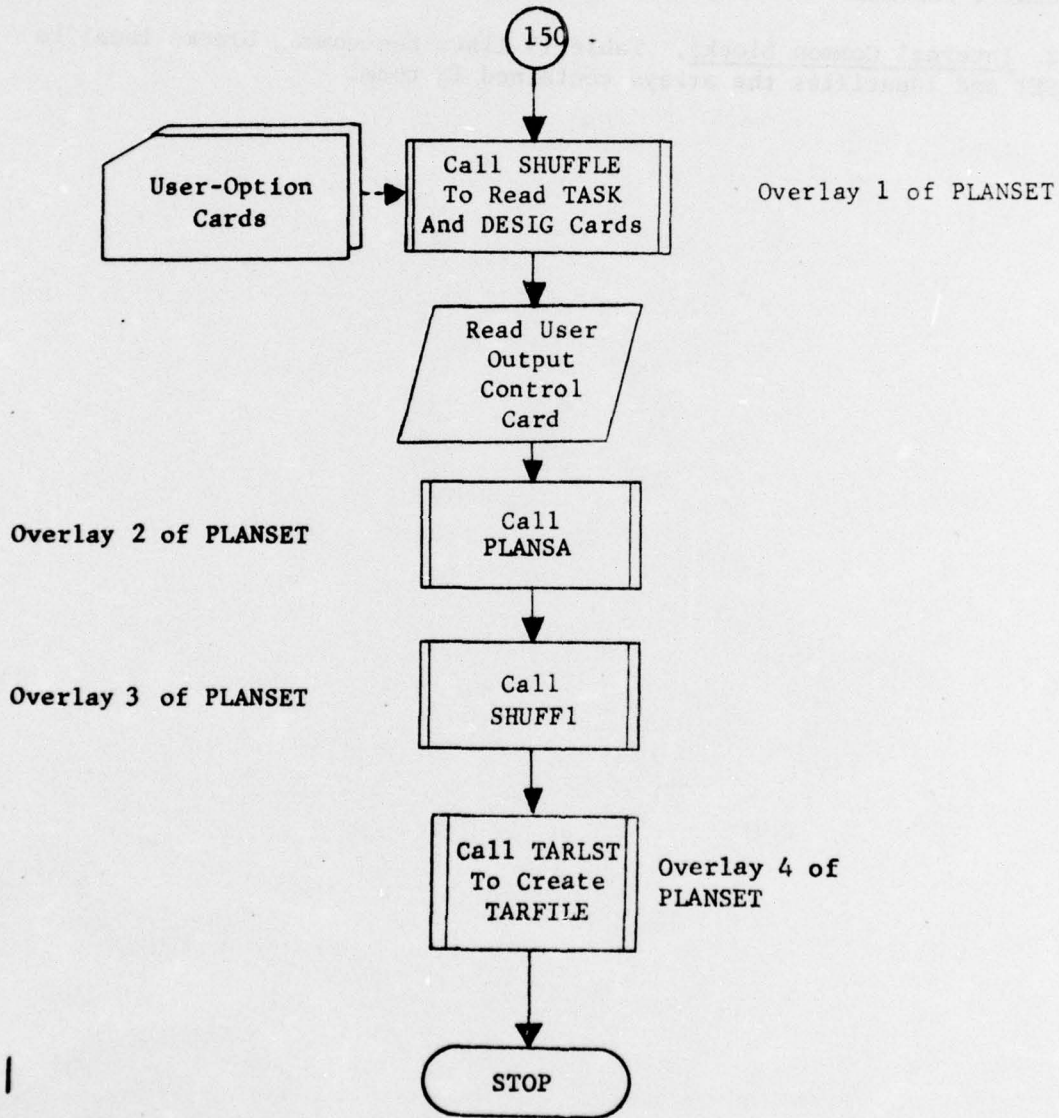


Figure 65. (Part 3 of 3)

## 7.5 Common Block Definition

7.5.1 External Common Blocks. PLANSET references the following utility routine common blocks, which are described in appendix A of the Program Maintenance Manual, Volume I: /EDITAPE/, /EDITERM/, /FILABEL/, /ITP/, /MYIDENT/, /NOPRINT/, /PROCESS/, and /WAROUT/.

7.5.2 Internal Common Blocks. Table 19 lists the common blocks local to PLANSET and identifies the arrays contained in them.

Table 19. (Part 7 of 15)

<u>BLOCK</u>	<u>VARIABLE OR ARRAY</u>	<u>DESCRIPTION</u>
MLTX	MULT (36)	Array containing multiple target data. Entries correspond to those for common block /TD/
	MLTX/FMLTX (I,J) (I = 8, J = 5)	I words of data for the Jth multiple target element I = 1: target name 2: target index number 3: target designator code 4: target task code 5: target country location code 6: target flag code 7: target latitude 8: target longitude
	NMULT	Number of elements in the current multiple target
NDES	NDES	The number of words written on WINFILE by program PLANSA.
PLSTCL		Hollerith literal data pool (see appendix A.)
PRCNTL	JJTGTS	Contents of target entry on user output option card
	JJJGP	Contents of group entry on user output option card
	JJJCPX	Contents of complex entry on user output option card
	JJTYPE	Contents of TGTSORT entry on user output option card
	JTGTPE	Contents of TGTSORT tape entry on user output option card
PRIOR	IPTASK(48)	Array into which TASK priority option cards are read
	IPDES(200)	Array into which DESIG priority option cards are read
	MPTASK	Number of TASKs in IPTASK array

Table 19. (Part 8 of 15)

<u>BLOCK</u>	<u>VARIABLE OR ARRAY</u>	<u>DESCRIPTION</u>
PRIOR (cont.)	MPDES	Number of DESIGs in IPDES array
	ISUBT	Task flag: If 0, only 1 character in TASK
RETARG	REDUCE(40)	The factor to be applied to NOPERSQN for missile type I when retargeting is considered
	RDBL (40)	The new ALERTDBL for missile type I when retargeting is considered
TAPES	LTWIN	Logical unit for WINFILE
	LTTIN	Logical unit for TINFILE
	LTTGT	Logical unit for intermediate target file
	LTGRP	Logical unit for intermediate group file
	LTDB	Logical unit for INDEXDB input
	LSRTA LSRTB	Logical units for sorting complex target information for sequenced printout
TAU	TAU (200)	Time components of all elements of a complex
	HC (80)	Hardness components of all elements of a complex
	V (200)	Values corresponding to TAU or HC
	INDEXT(90)	Array containing ordered indices
	FV (5)	Fractions of total value corresponding to V
	T(5)	Array containing time components for single target
	TBOX(5)	Not used
	VBOX(5)	Not used

Table 19. (Part 15 of 15)

<u>BLOCK</u>	<u>VARIABLE OR ARRAY</u>	<u>DESCRIPTION</u>
C12 (cont.)	CPLX/ICPLX(I,J) (I = 30, j = 210) (cont.)	29 correspond to the same variables in common block /TD/. ICPLX(30, J) is a copy of the complex number.
	LOOK(30)	Hold area for LSRTA and LSRTB data
	GRPCOMP/IGRPCOMP(I,J) (I = 5, J = 2500)	Array containing data for each weapon in a group. For each group element J, I = 1: base country location and index 2: base latitude 3: base longitude 4: base payload index 5: number of first vehicle/ number per base
C4	NDEXG(210)	Hold array containing the ordered group indexer
	IORDER(210)	Reordered group array
C5	TARLST/CURPGE(3516)	In core "page" of target list (TARLST)
	SRCPAD(3516)	Scratch pad of DESIG strings and items
	SCAT(7771)	Hash ordered array of "page" and target designator string pointers
	MAXPPG	Maximum number of words in a TARLST page (3516)
	MXSCAT	Maximum number of entries in the SCAT array (7771)
	MAXSRC	Maximum number of words in the SRCPAD array (3516)
	LAVSPG(15)	Pointers to lists of available space for TARLST pages 1 through 15
MXPGS	Maximum number of TARLST pages (15)	
	INCR	Number of words in target designator string item (6 words)

## 7.6 Overlay TARLST

PURPOSE: Generate TARFILE

ENTRY POINTS: TARLST

FORMAL PARAMETERS: None

COMMON BLOCKS: C5, EDITAP, EDITER, FILABEL, ITP, MYLABEL, MYIDENT, NOPRINT, PLSTCL, PROCESS, TAPES, TWORD, WT

SUBROUTINES CALLED: ABORT, CHANGE, IHASH, INITAR, INITED, IT, INITPG, INPITEM, NEXTITEM, RDARRAY, SETREAD, SETWRITE, TERMTAPE, WRARRAY, WRWORD

### Method:

TARLST is the last overlay within PLANSET and is called to create the final version of TARFILE. In the original design of PLANSET, TARLST was the first overlay executed, hence, the reason for its being presented in this document prior to the other overlays. The flowchart for TARLST is shown in figure 66. TARLST first reads the temporary version of TARFILE as written by subroutine CALCOMP and stores the information read. This data contains the hardest vulnerability for each complex where the representative target is not the hardest element. TARLST, then, reads one item at a time from the INDEXDB (or INMODDB) tape. If the item is an appropriate data base item, the target designator code (DESIG) is used to calculate a storage address in the hash ordered array SCAT. If the SCAT array entry at the calculated address is empty, the current TARLIST page number (KCPG) and the pointer index (INDX) to the storage area in the current page (CURPGE) are packed into the SCAT array entry. The item information LAT, LONG, DESIG, ISIX, FLAG, TARDEFHI, CNTRYLOC, TASK, CNTRYOWN, H1, and INDEXNO are packed and stored in six consecutive words of CURPGE beginning at the address INDX.\* The target designator string pointer (ISIX) is set to zero to indicate that this is the first item in the DESIG string. If a subsequent item DESIG calculates the same hashed SCAT address (and the indicated page (IPG) for the previous DESIG(s) is currently in core) the pointer in the SCAT entry (INDX) is changed to point to the new item and the new item string pointer (ISIX) is set to point to the predecessor item. If the indicated page (IPG) is not currently in core the item information is stored in a "scratch pad" in the same packed format as that of the TARLIST entries. The string pointers (ISIX) for these scratch pad items are set to '9999' as a flag for later reprocessing of the TARLIST pages. As pages of the TARLIST are filled in this manner they are output on the intermediate scratch file (IOUT). If a string of DESIGs were to overflow a page as it is filled, the string is "pulled" and stored in the SRCPAD. The pointer in the SCAT entry is changed to point to the SRCPAD string and the SCAT entry is set

\*See figure 67 for a description of the packing of this data.

negative to indicate that the string is in the SRCPAD. When the SRCPAD is 3/4 full, the intermediate scratch files IN and OUT are interchanged and the pages previously output in IOUT are reprocessed on IN. The reprocessing is accomplished by scanning the SRCPAD for '9999' flagged item entries. If the SCAT entry IPG is the same as that of the current page (input from IN) the string is pulled and stored in the SRCPAD in the same manner as described for overflow strings (negative SCAT entry). All pages are reprocessed in this manner until the SRCPAD is full or all strings are pulled. The SCAT array is then scanned for negative entries. The strings for these negative entries are then moved from the SRCPAD to the current page. This process continues until all SCAT entries are positive. Processing of the INDEXDB tapes then continues until all items have been processed.

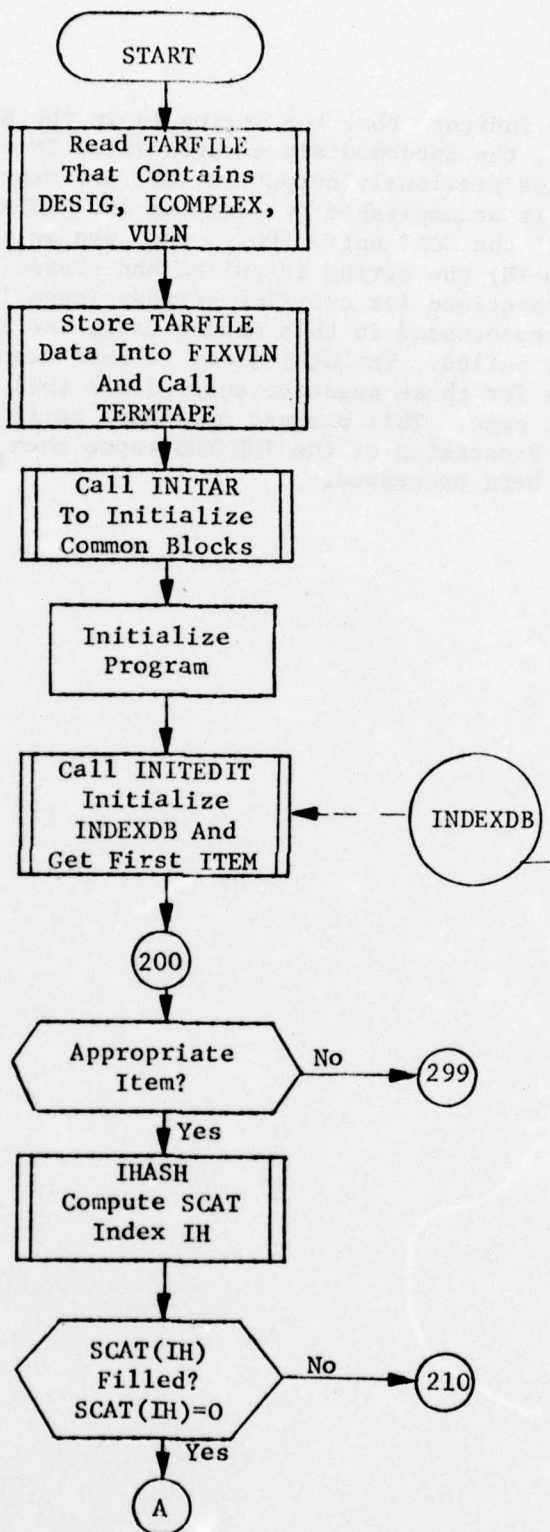


Figure 66. Overlay TARLST (Part 1 of 17)



### 7.6.1 Subroutine INITAR

PURPOSE: Initialize common block /C5/ used by Program TARLST

ENTRY POINTS: INITAR

FORMAL PARAMETERS: None

COMMON BLOCKS: C5

SUBROUTINES CALLED: None

CALLED BY: TARLST

Method:

INITAR (see figure 68 ) is called at the beginning of TARLST to initialize the above common block. The TARLST program limits (i.e., MAXPPG, MXSCAT, and MXPGS) are initialized; the SCAT and LAVSPG arrays are set to zero; and the array SRCPAD is initialized to be a list of available scratch space.

Subroutine INITAR is illustrated in figure 68.

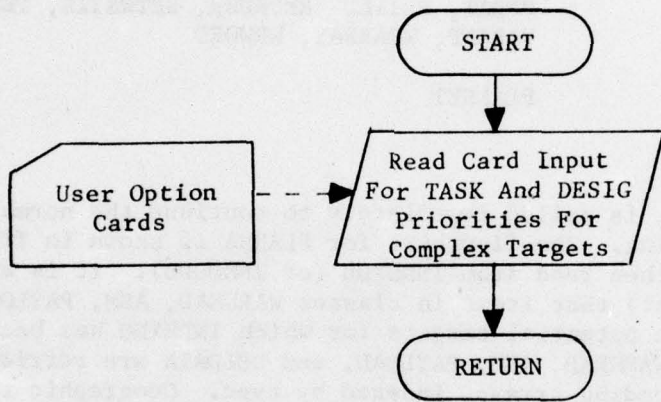


Figure 70. Overlay SHUFFLE

## 7.8 Overlay PLANS

PURPOSE: Processing of targets and weapons.

ENTRY POINTS: PLANS

COMMON BLOCKS: CLASNA, C1, C2, C3, C4, DPOOL, EDITAPE, EDITER, FILABEL, GROUP, ITP, KEYHOB, MAX, MISC, MLTX MYIDENT, NDES, NOPRINT, OVL, PAYDAT, PLSTCL, PRCNTL, RETARG, TAPES, TD, TWORD, WAROUT, WT

SUBROUTINES CALLED: ABORT, AROVRFLW, BUFOU, CHANGE, DBLCAL, IFUNIT, INITEDIT, INPITEM, IPUT, KEYMAKE, NEXTITEM, ORDER, PPDAT, PRITEM, REORDER, SETWRITE, TERMTAPE, VLRADP, WRARRAY, WRWORD

CALLED BY: PLANSET

### Method:

Overlay PLANS, is called immediately to continue the normal target and weapon processing. The flowchart for PLANS is shown in figure 71. The first item is then read from INDEXDB (or INMODDB). It is assumed (a data base requirement) that items in classes WARHEAD, ASM, PAYLOAD, and DBLDATA precede potential targets for which INDEXNO has been defined. Type data for WARHEAD, ASM, PAYLOAD, and DBLDATA are retrieved and stored in the corresponding arrays, indexed by type. Geographic type data also is retrieved and stored in memory for items in classes POINT, CORRIDOR, and LEGS for the attacking side, and for items in classes POINT and BOUNDARY on the defending side. Attributes that are similar for weapon types are retrieved for WTCL and stored in proper arrays.

Target Processing: 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. When the exemplar target specified by the data card for each class is encountered on the data base, a message is printed and 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 values (VALs), together with the value factors for each class, are used to compute the final normalized class value factors.

Multiple targets are not 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. Whenever an eligible missile site is encountered, a check is made to see if a multiple target is currently being processed. If not, one is started by retrieving the target data and storing it in the array MULT. The target counter (NTAR) then is incremented by one, as is the counter for the number of sites in the multiple target (NMULT), and the attributes NAME, INDEXNO, DESIG, TASK, CNTRYOWN, CNTRYLOC, FLAG, LAT and LONG are stored in the array MLTX. If a multiple target was being processed, the new site is added by incrementing the site counter NMULT and storing the appropriate attributes in array MLTX. Subroutine WRMULT is called whenever a multiple target is to be terminated; this occurs if any of the multiple criteria fail.

Targets which are not missile sites are separated into individual targets and targets which belong to a complex. For individual targets, the target data are retrieved and immediately written on an intermediate target file (LTTGT). When a complex target is encountered, the number of targets in the complex is incremented and stored by a complex in array NCPX. The maximum complex index (ICOMPLEX) is found and stored under MAXICOMP. Target data then are retrieved and stored in the target array, ITD. The first target in each complex is indicated by placing "1" under ITD(30); for all other targets in the complex a "2" is placed in this word after counters for the number of complexes (NCOMPLEX) and number of targets (NTAR) have been incremented. ICOMPLEX is stored under ITD(7) to distinguish the target as belonging to a complex, and the target data array is written into the target file (LTTGT). For all elements of complexes vulnerabilities are temporarily stored in ITD(12) and ITD(16) for use in subroutine CALCOMP. For each target processed a single word is written onto WINFILE for print by overlay SHUFF1. This word contains the FLAG and the DESIG of the target.

Weapons Processing: If the item being processed is the first site of a missile squadron (ISITE=1) or a bomber on the attacking side, and if auxiliary class WTCL has been defined (CHK(ITYPE) = 1), weapon type data which are the same for all weapons of the given type are defined from array WTP. This array was loaded from the WTCL entry. In the case of a missile, however, before PLANSET fills the WTP array, it checks the re-targeting flag (IRETARG). If on, the user has requested that the data base attribute IREP (reprogramming index) be considered for all missiles. PLANSET 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 that type. The new values will reflect the type of reprogramming capability indicated by IREP. If the IRETARG flag is off, no modifications for re-targeting capabilities are made (these calculations are described in the Analytical Manual, Volume II, Chapter 2, Missile Reprogramming).

At each appearance of a WTCL data item, the type characteristics are output to the WTCLDATA file for printing later in the program. The data for the Ith type (i.e., plan generation type LTYPE=1) is placed in the Ith sector of the WTCLDATA file.

After type data has been 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 (IALERT), type (ITYPE), region (IREG), naval data (IDBL, PKNAV), and payload. Bombers must also have the same refueling index (IREFUEL). 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.

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, this 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 200 groups can be formed for use in plan generation. However, PLANSET processes and prints information for up to 210 weapon groups to enable planners to adjust their data base should more than 200 groups be formed.

In addition, if the weapons have a time-dependent destruction before launch probability (DBL), then the spread in DBL between the first and last weapons must be less than the input parameter DMAXDBL. If the number of weapons on a base is so large that this criterion is not met, these weapons are split up. After the weapons are split, the program checks to determine if the weapons can be added to an existing group or if they must begin a new group.

When a new group is started group data are retrieved and stored in array GRP. The corresponding index to GRP and the attributes CNTYLOC, INDEXNO, LAT, LONG, and PAYLOAD are placed in the first five words of the array GRPX as each new base is added to the group. An index to vehicles on the base (ISTART) and the number of vehicles either on alert or in commission (NX) are packed into the sixth word of GRPX. The array is written immediately onto the intermediate group file (LTGRP). 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  $2 \times$  RANGEMOD.

Tanker bases are not included in the group assignment. As each tanker base is encountered, tanker base data are retrieved and stored in the array TANK. If the tankers on the base are to be automatically allocated (IREFUEL = 0 or  $\leq -4$ ), ITWORD is set to "-4"; for preassigned tankers (IREFUEL < 0), ITWORD is set to "-1". If IREFUEL = -3, an error message is printed and IREFUEL is reset to automatically allocate the tankers. For bases with IREFUEL = -1 or -2, the same error message is printed and the next data item is read. Otherwise the number of tanker bases (NTANKBAS) is incremented and array TANK, preceded by ITWORD, is written onto the group tape.

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 refuel once (IREFUEL = -4) if they originally refueled twice, or to nonrefuel (IREFUEL = 0) otherwise; 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, along with the index to the corresponding weapon array (WTP) and both are stored in the appropriate group array. The groups are sorted to place missile groups before bomber groups with time dependent missile groups appearing first. The intermediate target and group files (LTTGT and LTGRP) now are terminated as well as file WINFILE and a check is made to insure that all complexes have at least two targets. Subroutine PPDAT is called to print finalized arrays and subroutine AROVFLW is called to print messages indicating any array overflows which occurred during processing.

Overlay PLANSA is illustrated in Figure 71.

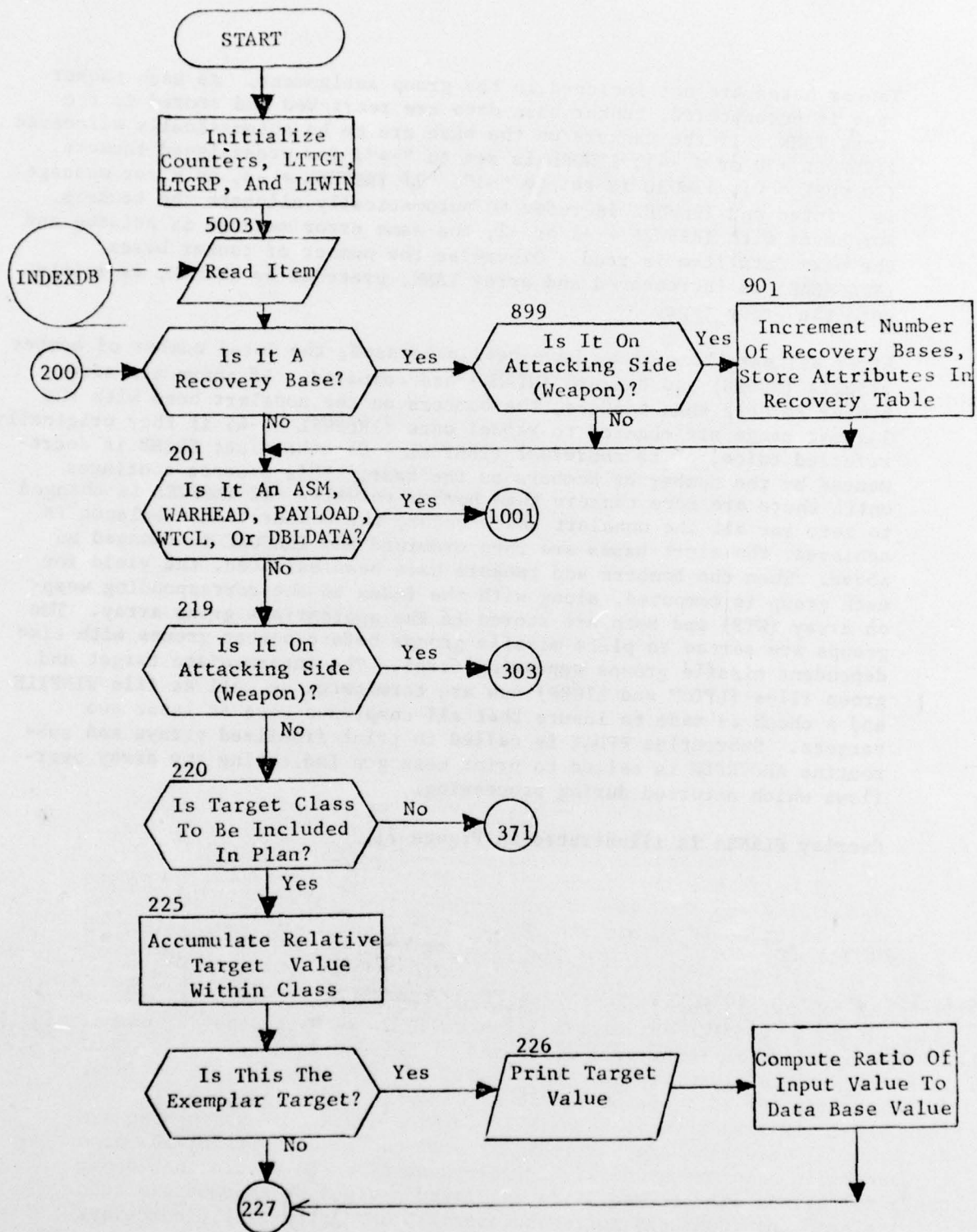


Figure 71. Overlay PLANSA (Part 1 of 22)

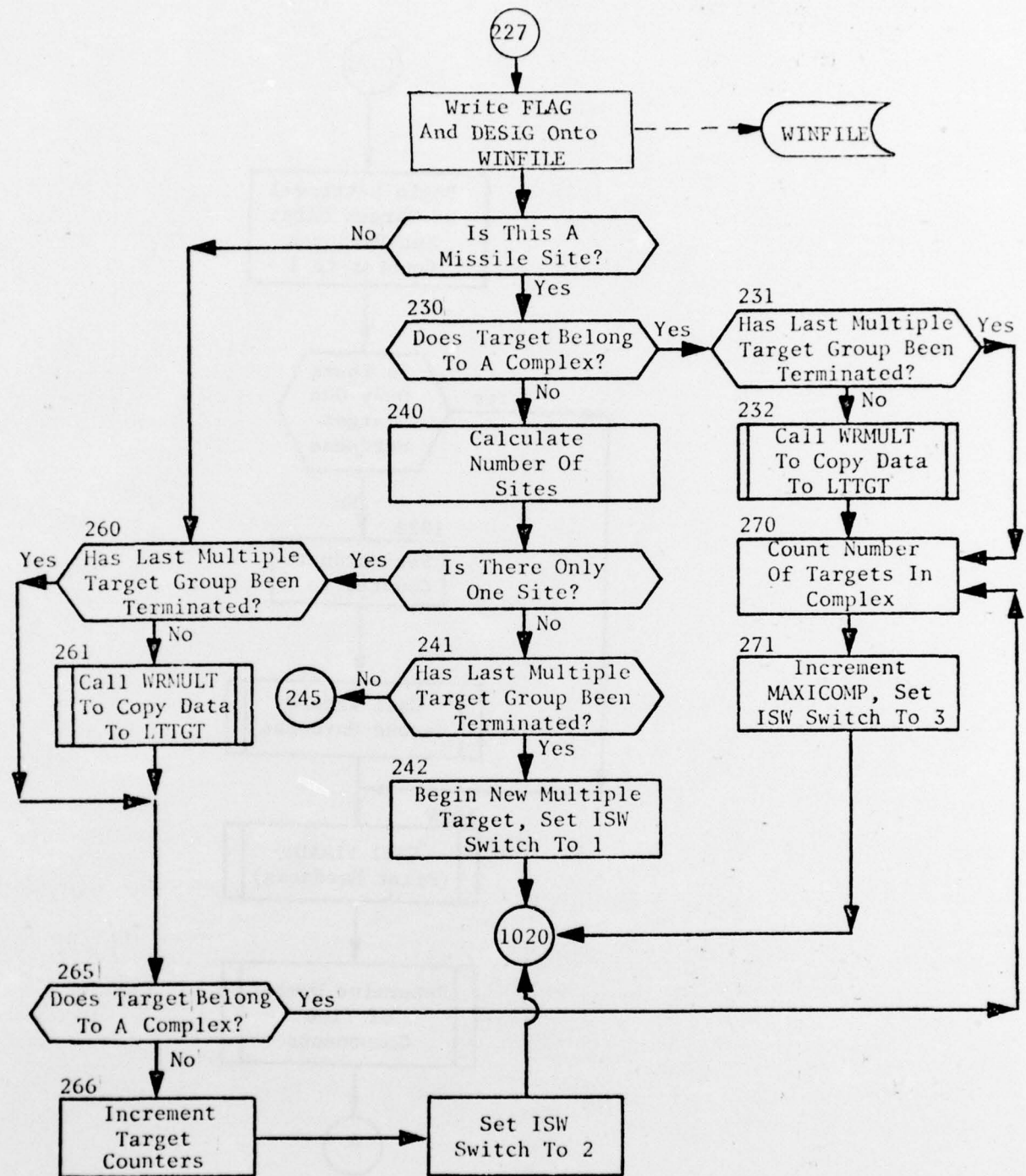


Figure 71. (Part 2 of 22)

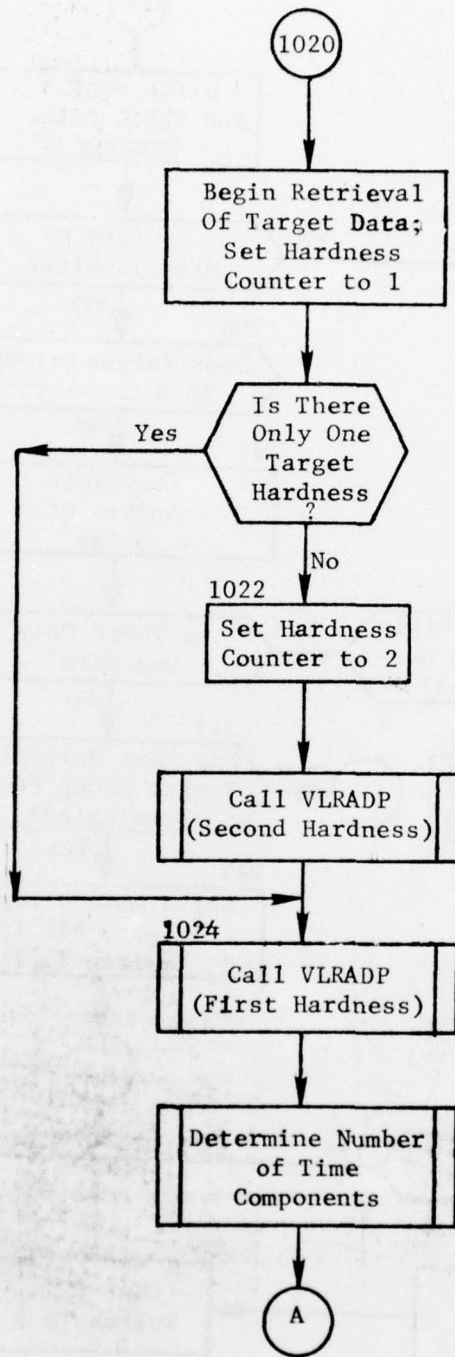


Figure 71. (Part 3 of 22)

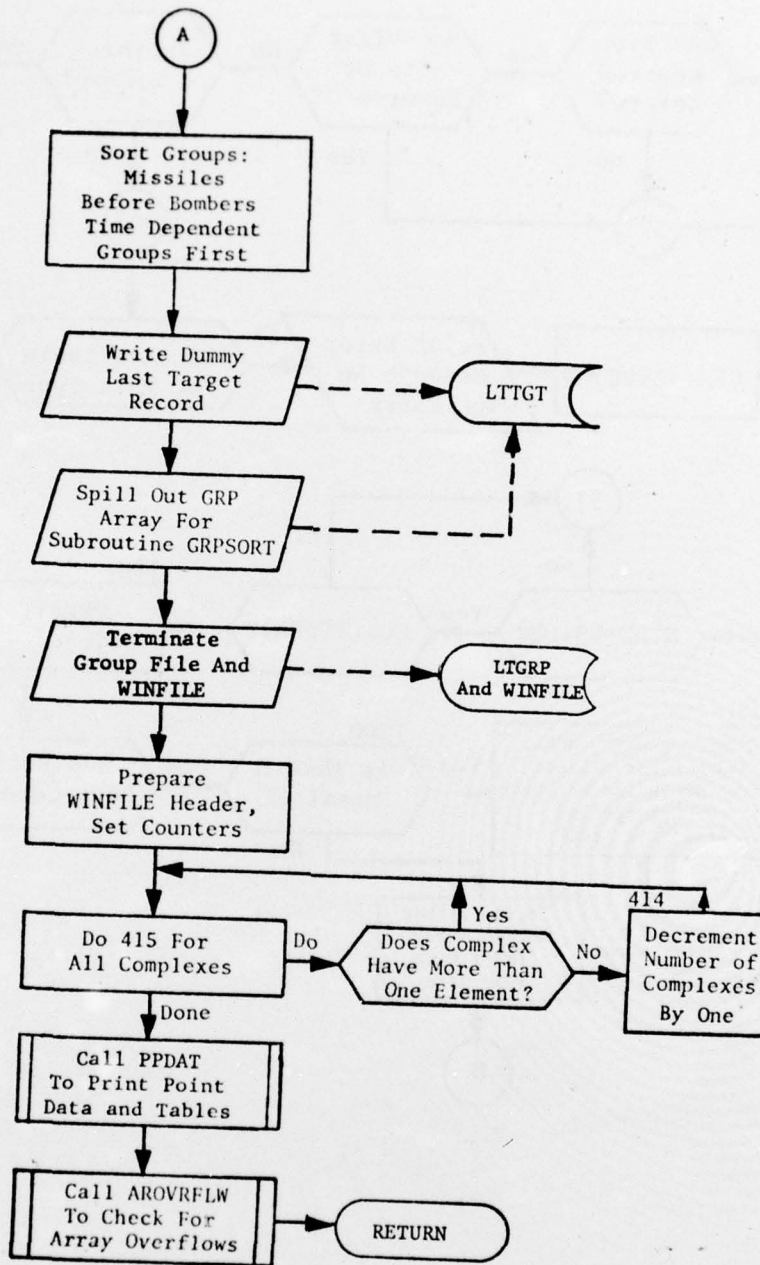


Figure 71. (Part 8 of 22)

# BEST AVAILABLE COPY

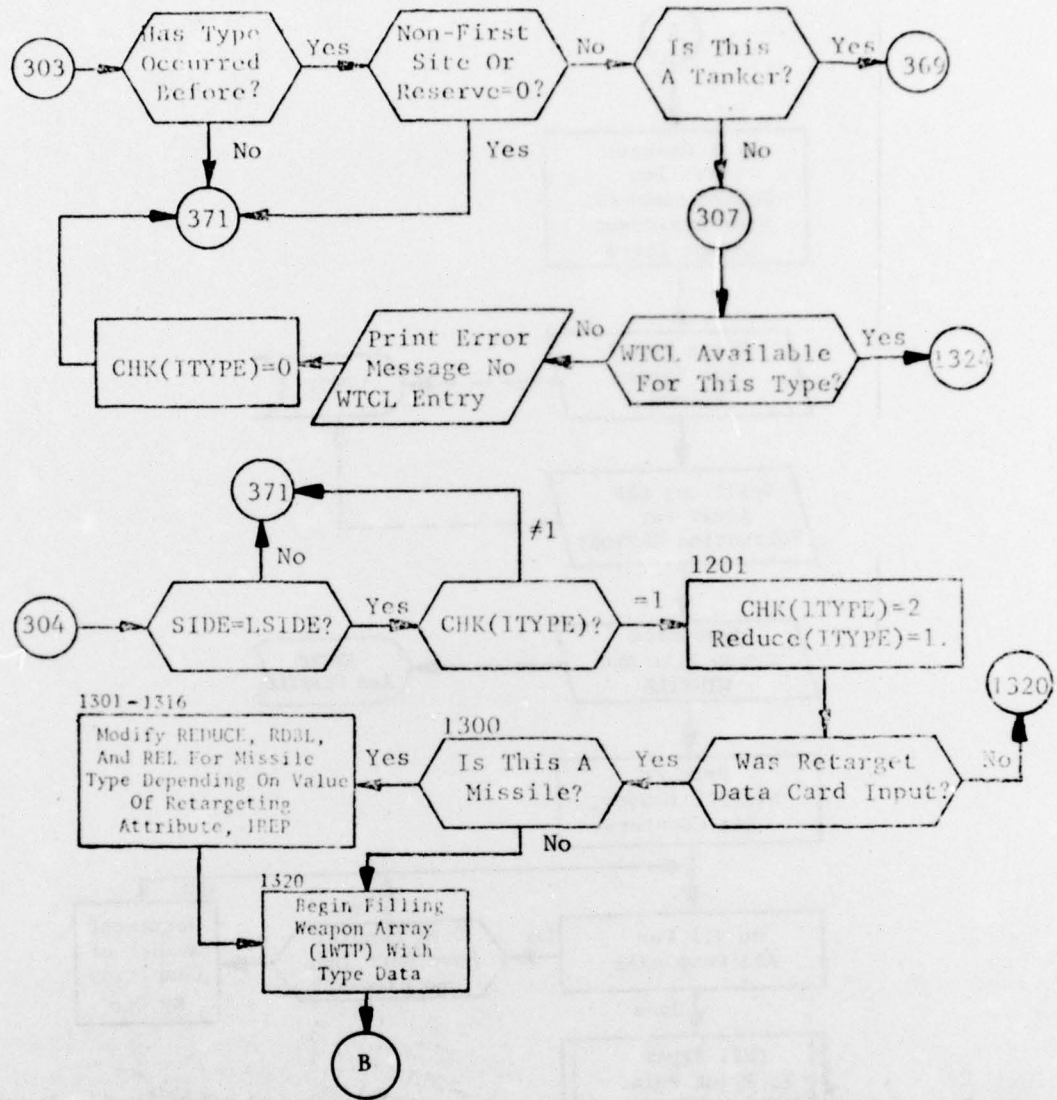


Figure 71. (Part 9 of 22)

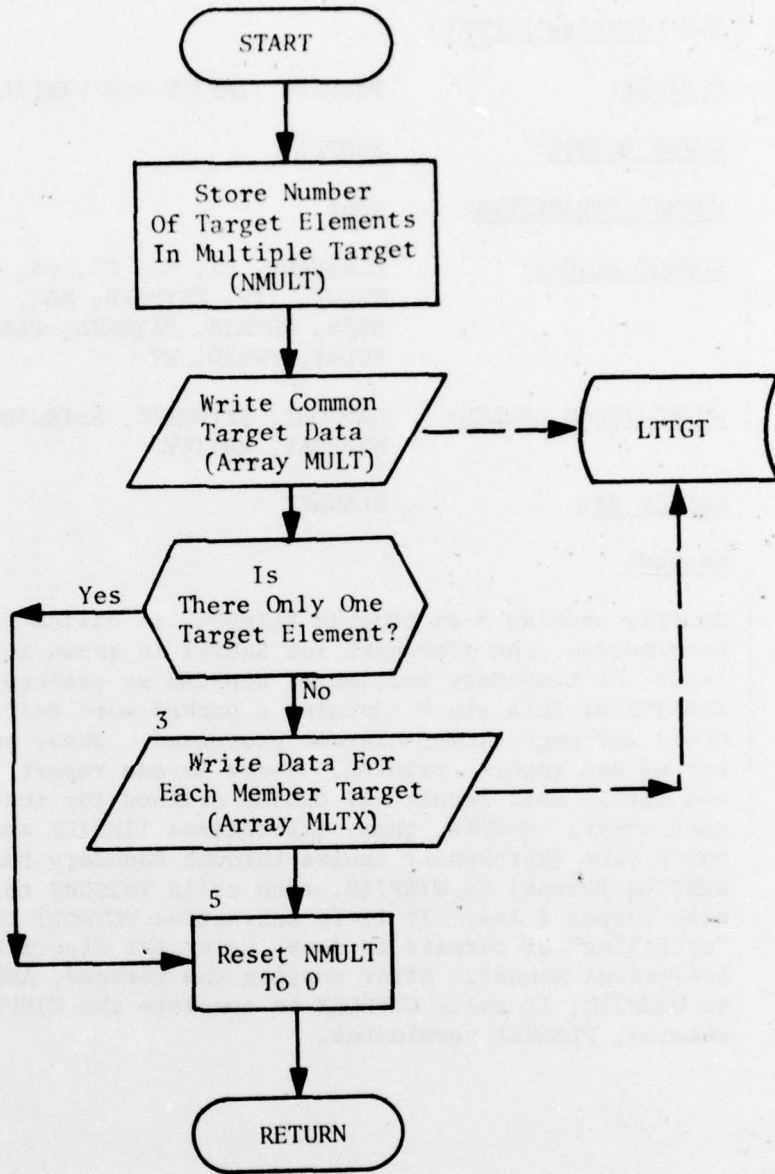


Figure 76. Subroutine WRMULT

### 7.9 Overlay SHUFF1

PURPOSE: Prepare TINFILE and WINFILE.

ENTRY POINTS: SHUFF1

FORMAL PARAMETERS: None

COMMON BLOCKS: CLASNAME, C1, C2, C3, C4, C12, DIRECTRY, DPOOL, GROUP, ITP, KEYMAKE, MAX, MISC, MYIDENT, MYLABEL, NDES, NOPRIN, PAYDATA, PLSTCL, PRIOR, TAPES, TODAY, TWORD, WT

SUBROUTINES CALLED: GRPSORT, SETWRITE, SETREAD, TERMTAPE, TGTSORT, WRARRAY, WRITER

CALLED BY: PLANSET

#### Method:

SHUFF1, overlay 3 of program PLANSET, is called just before PLANSET terminates. The flowchart for SHUFF1 is shown in figure 77. It first reads the temporary version of WINFILE as created by overlay PLANSA. WINFILE at this stage contains a packed word defining attributes FLAG and DESIG for each target element processed. These packed words are stored, sorted and reports printed. There is one report for each specified FLAG and within each report all DESIGs defined for that FLAG are printed in sort order. SHUFF1, then, initializes TINFILE and WINFILE, writes the point data (Breakpoint tables through Boundary Points; see table 17, WINFILE format) to WINFILE, then calls TGTSORT to add target records to both output files. It is in subroutine TGTSORT that the actual "shuffling" of targets is done, using the algorithm described in the Analytical Manual. After copying the warhead, ASM, payload, etc. tables to WINFILE, it calls GRPSORT to complete the WINFILE. When SHUFF1 returns, PLANSET terminates.

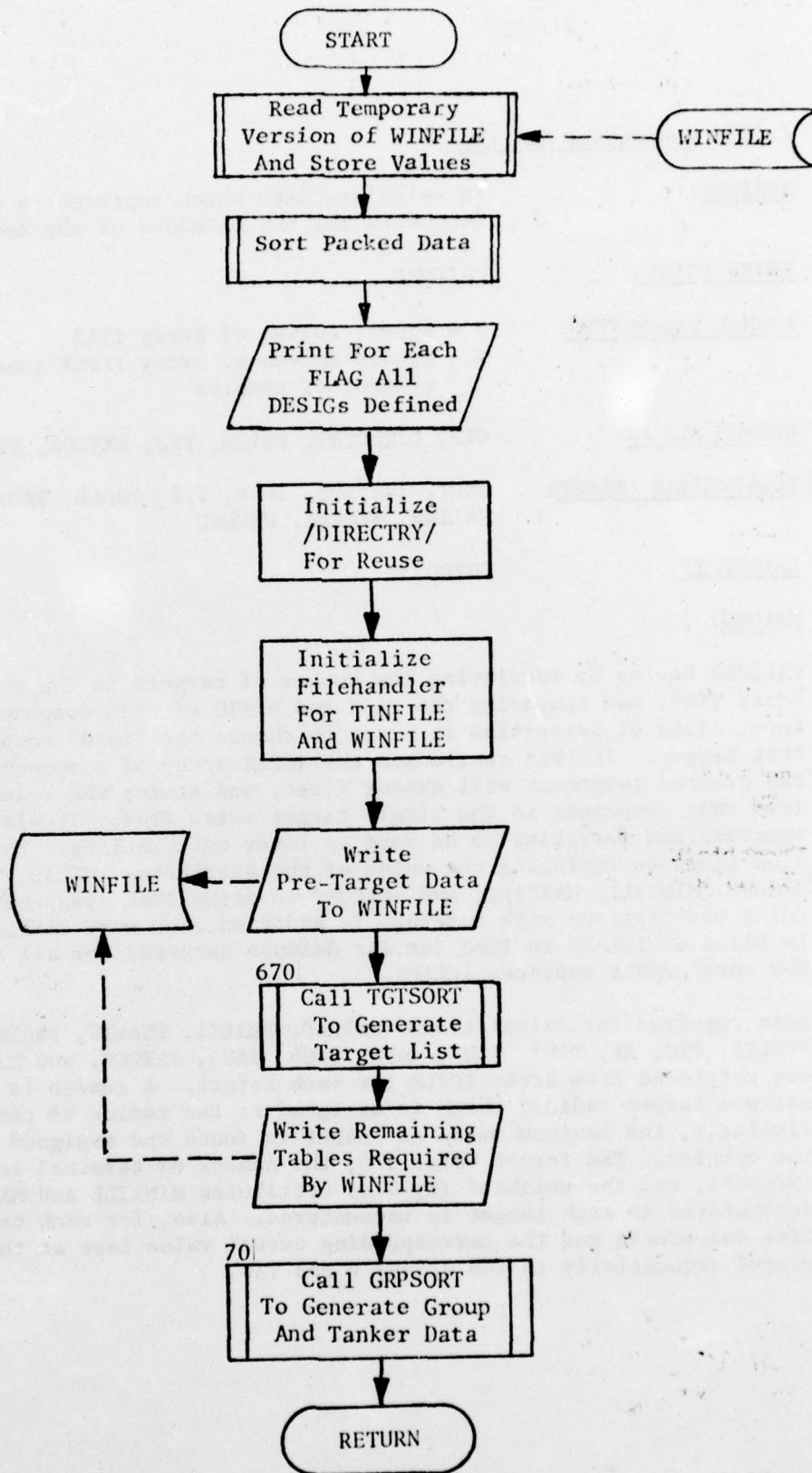


Figure 77. Overlay SHUFF1

### 7.9.1 Subroutine CALCOMP

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

ENTRY POINTS: CALCOMP

FORMAL PARAMETERS: I - Equals Column of array ITAR  
K - Equals Column of array ICPLX containing first element of complex

COMMON BLOCKS: C12, DIRECTRY, PRIOR, TAU, KEYHOB, PLSTCL

SUBROUTINES CALLED: IGET, IGETHOB, IPUT, ITP, ORDER, REORDER, TWORD, VALTAR, VLRADA, WRWORD

CALLED BY: TGTSORT

#### Method:

CALCOMP begins by retrieving the number of targets in the complex from array ITAR, and comparing the TASK and DESIG of each component to the input lists of priorities in order to choose the "lead" component for that target. It then rearranges the ICPLX array of components so that the desired component will appear first, and stores the relevant data from that component in the single target array ITAR. It also initializes counters and variables to be used in later calculations. This initialization includes replacing the value of the attributes TGTRAD, TARDEF, MISDEF, MINKILL, MAXKILL, and MAXCOST in array ITAR (see common block /C12/ description) with a zero. In addition, the word COMPLD is stored in place of ICLASS in ITAR for air defense targets; for all other targets the word COMPLE replaces ICLASS.

Data required for calculations (TGTRAD, FVALT1, FVALT2, FVALT3, FVALT4, FVALT5, VTO, NK, TAU1, TAU2, TAU3, TAU4, TAU5, TARDEF, and MISDEF) now are retrieved from array ICPLX for each target. 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 VTO, the number of terminal interceptors (MISDEF), and the weighted (by VTO) 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.

After all targets in the complex have been processed as above, the total value and the total number of terminal interceptors are stored in the target array ITAR. The average values of MINKILL and MAXKILL also are calculated and stored in ITAR. Subroutines ORDER and REORDER then 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 dependence 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 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 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. This hardest vulnerability is written onto TARFILE for use by the last PLANSET overlay.

Calculation continues to determine the hardness components (H1, H2) and the corresponding fractional value (FVALH1) which represent the complex. VTO, FVALH1, and the hardness number (1 or 2) are retrieved from array ICPLX for each target in the complex. Air lethal radii are, also, recalculated based on the defined scaled height of burst. The complement of FVALH1 is found to represent the second hardness component. If either fractional value is nonzero, it is multiplied by VTO to get the actual value at that hardness. The result is stored in array V, and the corresponding lethal radius is stored in array HC. 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 for both hard and soft targets for those radii and the result (HHARD or HSOF) is stored in array ITAR. Similarly, the actual value at each hardness (VHARD or VSOF) is accumulated. If there are no hard targets (i.e., VHARD=0), FVALH1 in array ITAR is set to 1; otherwise the fraction of actual value for hard targets (NHARD/VTOT) is assigned to FVALH1. The number of hardness components then is placed in ITAR, and control is returned to TGTSORT.

See figure 78 for the logic flow within CALCOMP.

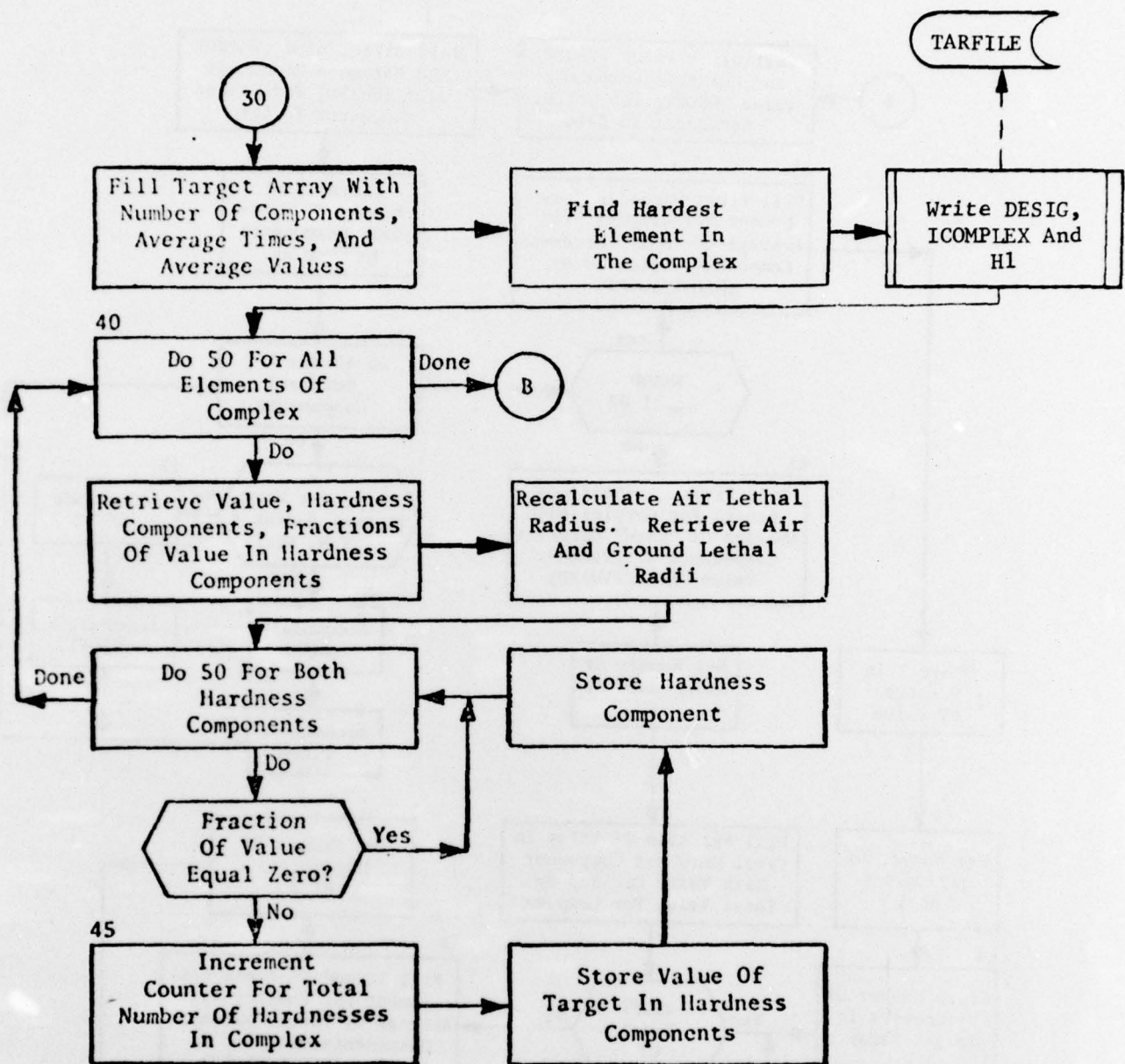


Figure 78. (Part 5 of 6)

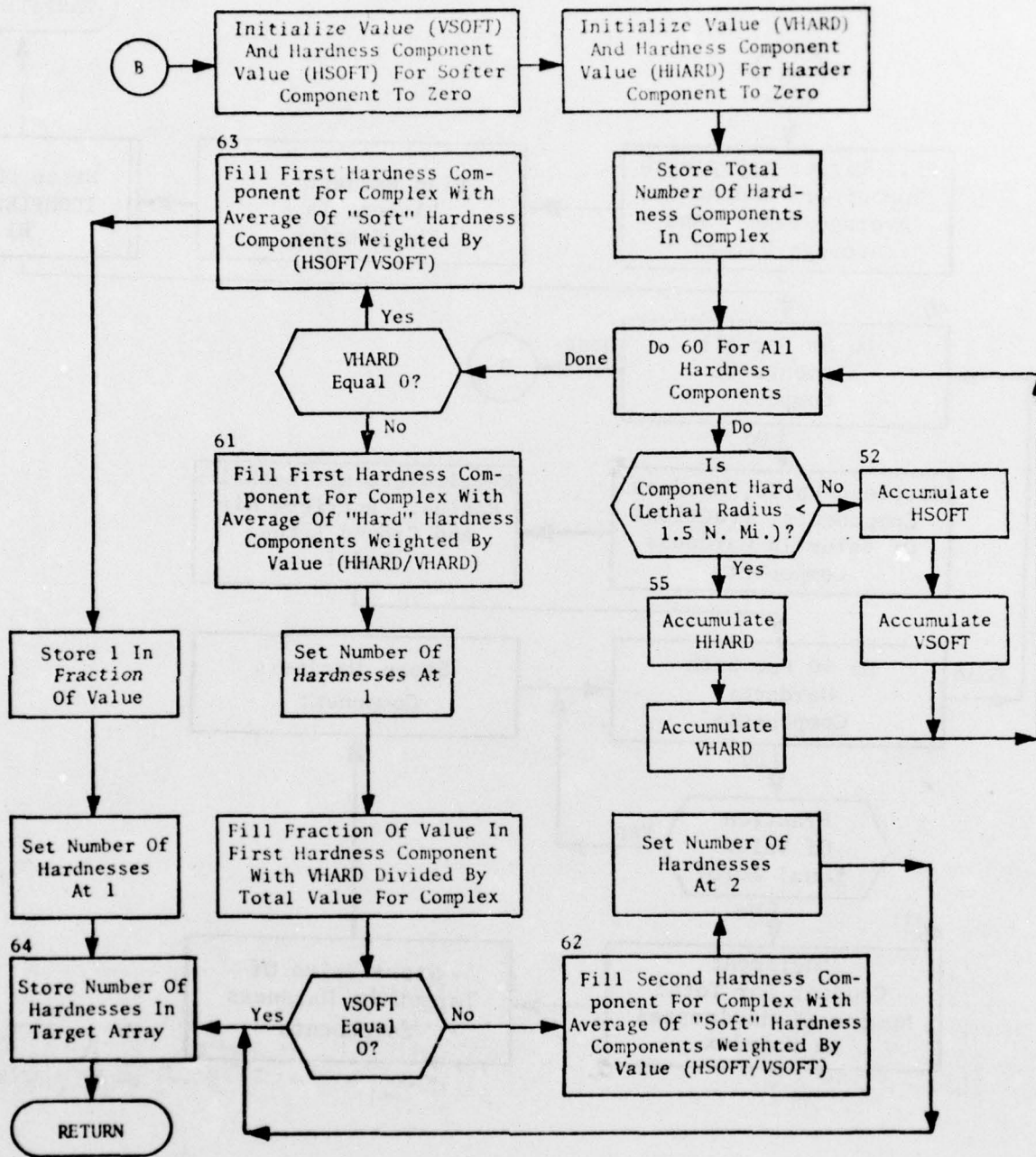


Figure 78. (Part 6 of 6)