COMMON TAG	DIMENSION	ITEM	UNITS
JPØNT	1	Control number	integer
XDEW	2,400	Storage block	

The following printed statement is also an output:

CHANGE TAPE B2, PROGRAM IS HALTING - PUSH START TO GO ON.

c. Program Logic. FD U65

(1) Steps 1-4. The contents of the index registers are saved. The record counter is initialized to zero and IFLAG is set to identification integer 2165. The control number for the target kit and the target tape production data are saved and the subprogram continues at step 5.

(2) Steps 5-6. The physical number representation of the first (next) digit of the year is stored in the record block. If both digits of the year have not been stored control is transferred to step 5. Otherwise the subprogram continues at step 7.

(3) Steps 7-9. The month of the production date is compared with the first (next) month in the list. If both months agree, the physical representation of the month is stored in the record block and the subprogram continues at step 10. Otherwise control is transferred to step 7.

(4) Steps 10-11. The physical representation of the first (next) digit of the day is stored in the record block.

If both digits of the day have not been stored control is transferred to step 10. Otherwise the subprogram continues at step 12.

(5) Steps 12-14. The physical number representation of the first (next) digit of the target tape control number is stored in the record block. If all six digits have not been stored control is transferred to step 12. Otherwise binary tape B2 is rewound, the tape check indicator is set ØFF, and the subprogram continues at step 15.

(6) Steps 15-20. The record block information is written on tape B2 as the 17 binary leader words. If the redundancy indicator is ØFF control is transferred to step 23. If two attempts to write the record have not been made the record is backspaced and control is transferred to step 15. If three areas on the tape have not been tried the written record is erased, initialization is made to read twice, and control is transferred to step 15. Otherwise the subprogram continues at step 21.

WWW.CHROMEHOOVES.NET

(7) Steps 21-22. SW(120) is set ØN and U08 prints the output statement. The subprogram halts for manual intervention and restarts at step 5.

(8) Steps 23-32. The record counter is stepped up by one and eleven 18-word records of zeros are written on tape B2. The block number is stepped up by one and drum slot

WWW.CHROMEHOOVES.NET

addresses are picked up for the first (next) target section. If SW(91), SW(92) . . ., or SW(100) is neither ØN nor ØFF, ITYER is set to six, SW(70) is set ØN, and ERRPRT prints a notification of the error. If ØN, ICØNT is set to T16 and JPØNT to one. If ØFF, T16 is set to zero and JPØNT to one. The subprogram continues at step 33.

(9) Steps 33-46. The RSLTS word is set up from the first (next) value in the RSLTS table, and the RDRUM word is set up from the first (next) value in the RDRUM table. If the RDRUM word is negative it is set plus. The scaling constant is picked up from the RDRUM word and is stored in the RRU scaling word. If JPØNT is one, the scaling is checked and TARKI processes the T constants. Otherwise the constant scaled for RRU computations is saved and SUB1 sets up the RRU drum address and instruction. If constants T1-T14 have not been processed, control is transferred to step 33. Otherwise JPØNT is set to one and the target data inventory number and desired ground zero are saved. The RSLTS word is set up from the RSLTS table and the subprogram continues at step 47.

(10) Steps 47-58. RDRUM words are set up from values in the RDRUM table. The RRU drum address and scaling constants are saved. SUB1 sets up the RRU drum address and instruction. If constants T16 through T18 have not been processed control is transferred to step 47. Otherwise

the scaling constant from the RDRUM table is checked and TARKI processes constant T19. If the target being processed is the first target, scaling constants are picked up from the RDRUM table and SUB1 sets up the RRU drum address and instruction. Control number T21 is set up and the target section control is stepped up by one and tested. If all ten target sectors have not been processed control is transferred to step 23. The M constants sector is set up in the XM table and the subprogram continues at step 59.

(11) Steps 59-66. The first (next) value of XM is saved in the RSLTS table. The RRU drum address and tag are saved and the sign of the RDRUM word is reversed. The XM scaling constant is checked and if the RDRUM tag is zero the RDRUM tag constants are checked. If all the M constants have not been processed control is transferred to step 59. Otherwise the G constants sector is set up and the subprogram continues at step 67.

(12) Steps 67-71. The first (next) value of the G constants in the XGI table is saved in the RSLTS table. The RRU drum address is saved and the sign of the RDRUM word is reversed. The XGI scaling constants are checked and if constants G1-G8 have not been processed control is transferred to step 67. Otherwise the subprogram continues at step 72.

(13) Steps 72-87. The next value in the XGI table is

saved in the RSLTS table. The RDRUM address and tag are saved and SUB1 sets up the RRU drum address and instruction. If constant G11 is not being processed and the RDRUM tag is not zero, the RDRUM tag constants are checked. Otherwise constant G11 is bypassed. If constants G9-G15 have not been processed control is transferred to step 72. Otherwise the last value in the XGI table is saved in the RSLTS table. The address and scaling factor of the RDRUM is checked and saved, the RRU word is set up, and the G constant is scaled. If G16-G19 have not been processed step 83 is repeated for the next G constant. Constant T19 is stored for sector zero and SUB1 sets up the RRU drum address for constant T19. The end-of-block and end-of-tape codes are stored as the last word of the block and the subprogram continues at step 88.

(14) Steps 88-101. An 18 word record is written on tape B2. If the tape check indicator is ØN and three attempts have been made to write the record, control is transferred to step 21, otherwise the record is backspaced and control is transferred to step 88. If the tape check indicator is ØFF the record counter is stepped up by one. If all blocks have not been written control is transferred to step 88. One record is written on tape B2 from the record block, with an end-of-file code in the last word of the record. If an even number of records have been written two end-of-files are written on tape B2. Otherwise one record is written on tape B2 from the record block, the redundancy indicator if ØN, it is turned ØFF and two end-of-

files are written on tape B2. Tape B2 is rewound, the index registers are restored, and the subprogram returns to the user subprogram.

2-180. SUBPROGRAM U58 (TARKI). TARKI performs scaling of the T, M, and G constants for the RRU computer. The FORTRAN II reference statement is CALL TARKI.

a. Inputs. The inputs are as follows:

COMMON TAG	ITEM	UNITS
GRASE(10)	RDRUM word	
GRASE(11)	Scaling constant	integer
GRASE(15)	RSLTS word	

b. Outputs. The outputs are as follows:

COMMON TAG	ITEM	UNITS
GRASE(14)	Constant scaled for RRU computer	

c. Program Logic. FD U58. The contents of the index registers are saved. The sign of the RDRUM word is tested, and if negative, the sign of the RSLTS word is made positive. If positive, the sign of RSLTS is made negative. The characteristic of the floating point RSLTS word is tested. If the characteristic is greater than 128, the bits are shifted for scaling and then shifted again to obtain the characteristic. If the characteristic is less than 128, the characteristic is subtracted from 128, the bits are shifted for scaling, and then shifted to obtain the characteristic. If the characteristic is 128, no shift is required for scaling, and the bits are shifted to obtain the characteristic. The scaling constant is tested with 35. If the scaling constant is less than or equal to 35, the scaling constant is subtracted from 35. The scaling constant is positioned for storage in the address of the RRU word. The sign of the RSLTS word and the scaled constant are masked with the RRU word (bits S, 1-35). The RRU word is positioned and rounded at bit 24 such that the RRU word is now contained in bits S, 1-23. If the RRU word is negative and not new, the magnitude is complemented, if negative and zero it is set to positive zero. SUB1 stores the RRU word into a block for processing. The contents of the index registers are restored and the subprogram returns to the user subprogram.

2-181. B5 SUBPROGRAMS.

2-182. The subprograms described in this area are the entry and control subprograms of the Target Area Accessibility determination function.

2-183. Subprogram LO4 (BENTRY) enables the loader to establish linkage between the B1 and B5 subprograms. This version of BENTRY will be in core only when the TAA mode of operation is requested. The return path of the user subprogram is saved by SAVE4, and TAANT is called. After TAA has been completed, the subprogram exits to DØCNT through RTRN4.

a.	BENTRY	*LO4	Establish B5 Control Area
b.	TAANT	P53	TAA Control

* Subprogram description is in the introductory paragraph of this area.

2-184. SUBPROGRAM P53 (TAANT). TAANT determines the area of possible targets accessible from a given ground guidance complex. The target accessibility area of each launch pad-antenna combination is defined by four points which outline a spherical quadrangle on the real-earth. The two sides of the spherical quadrangle are obtained by simulating maximum and minimum burnout times along the left-most and right-most launch azimuth directions. The FORTRAN II reference statement is CALL TAANT.

a. Inputs. The inputs are as follows:

COMMON TAG	DIMENSION	ITEM	SYMBOL	UNITS
GTAA	2,10	TAA storage block		
	I,1	Maximum burning time to SECO	t_{MAXB}	seconds
	I,2	Minimum burning time to SECO	t_{MINB}	seconds
	I,5	Vernier burning time	t_{BV}	seconds
	I,6	Maximum positive (clockwise) launch azimuth relative to base reference azimuth	A_{LMAX}	degrees
	I,7	Maximum negative (counterclockwise) launch azimuth relative to base reference azimuth	A'_{LMAX}	degrees
	I,8	Burning time to SECO this run	t_{BT}	seconds
	I,9	Maximum range value-input to TRJPAR	R_{MAX}	degrees
	I,10	Minimum range value-input to TRJPAR	R_{MIN}	degrees

COMMON TAG	DIMENSION	ITEM	SYMBOL	UNITS
PLWR	2	Reference azimuth bearing of launch pads	A_{ref}	degrees
FRTØD	2	Conversion constant: radians to degrees (180.0/π=57.295780)		
GMDST	2	Distance from point to center of earth	r_p	
GMLØN	2	Longitude of point west of Greenwich	λ'_p	
GMILE	2	Conversion constant: degrees to nautical miles (=60.042499)		
GMLAT	2	Geocentric latitude of point	L_{CP}	
FTFSP	2	Current time of flight since liftoff	t_f	
GTRNG	2	Inertial range to target-general	ϕ'	
GKICK	2	Current value of kick angle delta	δ	
SW(82)	1	If ØN, print on line		

b. Outputs. The outputs are as follows:

COMMON TAG	DIMENSION	ITEM	SYMBOL	UNITS
IPØNT	1	Internal counter		
GCLAT	2	Geocentric latitude	L_C	
GGLAT	1	Geographic latitude	L_g	
GMLØN	2	Longitude of point west of Greenwich	λ'_p	
GDPSM	2,15	Final detonation point		

COMMON TAG	DIMENSION	ITEM	SYMBOL	UNITS
GZMTH	2	Current estimate of earth-fixed target bearing	A_T	
GTRNG	2	Inertial range to target	ϕ'	
GTBRG	2	Inertial target bearing	B_T	
GLAZM	2	Launch azimuth	A_L	degrees
GEFXR	2	Current value of earth-fixed range	R_T	
KØUNT	1	Internal counter	K_{TAA}	
GTMFL	2	Total time of flight since liftoff at which point position is valid		
TWSCL	2	Current target - sine of geocentric latitude	$\sin L_{CT}$	
TWCCL	2	Current target - cosine of geocentric latitude	$\cos L_{CT}$	
TWDRV	2	Current target - distance to center of earth	r_T	feet
TWLN	2	Current target - longitude west of Greenwich	λ_T	degrees
TWCLT	2	Current target - geocentric latitude	L_{CT}	degrees
SW(8)	1	If ØN, monitor radar slew rate A during early portion of booster flight		
SW(9)	1	If ØN, RSDØRE requested to perform data recordings		
SW(20)	1	If ØFF, omit D term in gravity computations		

WWW.CHROMEHOOVES.NET

WWW.CHROMEHOOVES.NET

COMMON TAG	DIMENSION	ITEM	SYMBOL	UNITS
SW(21)	1	If ØFF, omit J term in gravity computations		
SW(32)	1	If ØN, ØLØØP on re-entry to stop at air burst time; if ØFF, stop at detonation altitude		
SW(41)	1	If ØN, input meteorological data specifying pressure and density deviations from standard atmosphere are to be used		
SW(43)	1	If ØN, suppress $\epsilon_{\dot{c}o}$ maximum initial value gate logic		
SW(44)	1	If ØN, suppress yaw steering		
SW(47)	1	If ØN, suppress noise in RADSZM, PRCSØ		
SW(55)	1	If ØN, IIP or fuel exhaustion impact point to be determined		
SW(54)	1	If ØN, booster shell impact point to be determined		
SW(58)		If ØN, RSDØRE is to record data from GGDSIM		
SW(64)	1	If ØFF, do not compute time to go in GGDSIM		
SW(131)	1	If ØN, SWAP to call CLØØP		
SW(132)	1	If ØN, SWAP to call ØLØØP		
SW(133)	1	If ØN, open loop guidance to be used (TAA control)		
SW(159)	1	If ØN, ØLØØP does ballistic simulation		
SW(160)	1	If ØN, ØLØØP does re-entry simulation		

Pages 2-465 through 2-468 are missing

2-185. B6 SUBPROGRAMS.

2-186. The subprograms described in this area make decimal corrections to Common location only for the MSS subprograms, and generates or updates the missile trajectory tape.

2-187. Subprogram L05 (BENTRY) enables the loader to establish linkage between the B1 and B6 subprograms. This version of BENTRY will be in core only when the DEC mode of operation is requested. The return path of the user subprogram is saved by SAVE4, and subprogram DECNT is called. After DEC has been completed, the subprogram exits to DØCNT through RTRN4. The subprograms are as follows:

a.	BENTRY	**L05	Establish B6 Control Area
b.	CMPUT	U47	Establish Relative Address
c.	DADDR	U46	Determine Common Address
d.	DCARD	U49	Process DOC-DEC Card
e.	DECNT	U45	Control Processing of Decimal Correction Cards
f.	MTTAPE	U10	Generate or Update Missile Trajectory Tape
g.	STØRE	U48	Store Converted Decimal Correction

** Subprogram description is in the introductory paragraph of this area.

2-188. SUBPROGRAM U47 (CMPUT). CMPUT computes the relative address for one, two, and three dimensional arrays. The FORTRAN II reference statement is CALL CMPUT (NØ,N1,N2,N3, L1,L2).

a. Inputs. The inputs are as follows: NØ = 1,2,3 for dimension of array; N1, N2, N3 the integers to request address in a dimensional array; L1, L2 length of second and third index, respectively.

b. Outputs. The output is as follows:

COMMON TAG	DIMENSION	ITEM	UNITS
IACTG	1	Address of desired tag	integer

c. Program Logic. IFLAG is set to identification integer 2147. The relative address for a one, two, or three dimensional array is computed by use of expressions (1), (2), or (3) and is stored in IACTG. CUTIE is stepped by one and control is returned to the user subprogram.

d. Expressions.

$$IX = N1 - 1 \qquad (1)$$

$$IX = N1 - 1 + L1\,(N2 - 1) \qquad (2)$$

$$IX = N1 + L1\,(N2 - 1) + L1\,L2\,(N3 - 1) \qquad (3)$$

2-189. SUBPROGRAM U46 (DADDR). DADDR determines the address of a Common tag for the input array and sets up the relative address constants according to a binary coded decimal correction card. The FORTRAN II reference statement is CALL DADDR.

a. Inputs. The inputs are columns 31-54 of a BCD correction card, which are stored in ADRES.

b. Outputs. The outputs are as follows:

COMMON TAG	DIMENSION	ITEM
ISATA	1	Number of dimensions in the array
ISATA-1	1	First subscript of the array.
IDLTA	1	Second subscript of the array
IDLTA-1	1	Third subscript of the array
IHLTA	1	Binary length of the first subscript of the array
IHLTA-1	1	Binary length of the second subscript of the array
UDCAD	1	Common address of the tag

c. Program Logic. FD U46

(1) Steps 1-6. The contents of the index registers are saved. IFLAG is set to the identification integer 2146. The data and output registers are initialized to zero. Step 18 of the error routine is initially set to store the error bits in CLEL for columns 1-36.

(2) Steps 7-13. The contents of columns 31-36 are

examined in order. If a column contains an illegal character, control is transferred to the error routine at step 16. If it contains a left parenthesis or a blank, the end of the array tag is indicated, and control is transferred to step 14 or 15, respectively, to locate the address of this tag in Common. Otherwise, the character is inserted in a register accumulating the characters of the tag. If all six columns (31-36) have been examined without reaching a blank or a left parenthesis, control is transferred to the error routine at step 16. Otherwise, control is transferred back to step 7 to check the next column.

(3) Steps 14-15. If a left parenthesis has been found, step 21 is modified to continue at step 22 to examine the succeeding characters. The address in Common of the assembled tag is searched for, and control is transferred to step 20 when found. If none is found, the program continues to the error routine at step 16.

(4) Steps 16-19. Error routine. SW(70) is set ØN and ITYER is set to four. The columns in error are stored in CLEL for columns 1-36 or CLER for columns 37-72. The contents of the index registers are restored and the subprogram returns to the user subprogram.

(5) Steps 20-28. The Common address of the tag is stored in UDCAD. If a left parenthesis has not been found, control is then transferred to step 122. Otherwise, the count of the number of dimensions is set to one and the

next column is made available for examination.

(6) Steps 29-39. If the column being examined contains a comma or a right parenthesis without at least one previous digit having been found, control is transferred to the error routine at step 16. If it contains a comma following a digit, the count of the number of dimensions is set to two, the next column is made available for examination, and control is transferred to step 48 to process the second dimension. If it contains a right parenthesis following a digit, control is transferred to step 79 to convert the dimension found to binary. If it contains neither a comma nor a right parenthesis, control is transferred to step 40 to examine it for a digit from 0-9.

(7) Steps 40-47. If examination at this point does not reveal a digit from 0-9, control is transferred to the error routine. Otherwise, the digit is saved, the digit count for the first dimension is increased by one, the contents of the next column are made available, and control is transferred back to step 28.

(8) Steps 48-67. These steps repeat steps 28-47, examining the second dimension characters. On finding a properly located comma, the count of the number of dimensions is set to three and control is transferred to step 68 to examinine the third dimension characters.

(9) Steps 68-78. These steps repeat steps 28-47, ex-

amining the third dimension characters, except that no examination is made for a comma, only a digit from 0-9, or a right parenthesis following such a digit, being acceptable at this point.

(10) Steps 79-86. The digits, if any, found in dimensions one, two, and three are converted from BCD to binary. If only one dimension has been found, control is transferred to step 122, bypassing the examination of columns 49-54. Otherwise, those columns are made available for serial examination and step 18 of the error routine is modified to store the error bits in CLER for columns 37-72. Columns 49-54 contain the lengths of the first, or first and second, subscripts in columns 31-44.

(11) Steps 87-96. These steps, and steps 97-108, examine the first set of characters in columns 49-54. If the first column examined does not contain a left parenthesis, or if the next column does not contain a digit from 0-9, control is transferred to the error routine. Otherwise, the digit is saved, the digit count for the first subscript length is set to one and the contents of the next column are made available.

(12) Steps 97-108. If this character is a comma, and either three dimensions were not found or all six characters from columns 49-54 have already been examined, control is transferred to the error routine. Otherwise, on finding a comma, control is transferred to step 109 to check the sec-

ond set of characters. If a right parenthesis is found, control is transferred to step 118 to process the digits just assembled. If a digit from 0-9 is found it is saved, the digit count for the first subscript length is increased by one, and, if all six columns have not been examined, control is transferred back to step 97 to examine the next character. If the digit found is the last of the six columns being examined, control is transferred to the error routine.

(13) Steps 109-116. These steps repeat steps 97-108, examining the second set of characters in columns 49-54, except that the check for a comma is omitted, only a digit from 0-9, or a right parenthesis is following such a digit, being acceptable at this point.

(14) Steps 117-118. The first or second of these two steps is entered on finding a right parenthesis after the second or first digit set, respectively, in columns 49-54. Therefore, if the digit count for that dimension is not at least one at this point, control is transferred to the error routine at step 16.

(15) Steps 119-122. If three dimensions were found in columns 31-44, the second, as well as the first, set of digits in columns 49-54 is converted from BCD to binary. If only two dimensions were found in columns 31-44, only the first set of digits from columns 49-54 is converted. INTRØG then determines if any errors have occurred by exam-

ining SW(70). If it is on, an error has occurred and control is transferred to step 126, bypassing the storage in Common of the address constants determined by this subprogram.

(16) Steps 123-125. The number of dimensions in the array are stored in ISATA. The first, second, and third subscripts of the array are stored in ISATA-1, IDLTA, and IDLTA-1. The binary length of the first and second subscripts is stored in IHLTA and IHLTA-1.

(17) Step 126. The contents of the index registers are restored, and the subprogram returns to the user subprogram.

(18) Steps 127-132. Except for the very first time, this is the open subroutine used to step the column indicator. If the column indicator is already set to the last column of the present set of six columns being examined, the error bits for this set are stored in CLEL, and step 16 of the error routine is modified to store error bits in CLER. If the column indicator is not set to the last column of this set, it is stepped one. Control is then transferred to the step following the transfer to step 127.

2-190. SUBPROGRAM U49 (DCARD). DCARD interprets the DOC-DEC card. The FORTRAN II reference statement is CALL DCARD.

a. Input. The input is the DOC-DEC card located in the card image areas CDI1 to CDI11.

b. Outputs. If an error occurs in U41, or if any car column is in error, SW(70) is set ØN. The following regi - ters are also outputs:

COMMON TAG	DIMENSION	ITEM
NØEES	1	Number of E conversions
NØIIS	1	Number of I conversions
NØØØS	1	Number of Ø conversions

c. Program Logic. FD U49

(1) Steps 1-3. The E value represents the number of floating or fixed point corrections, the I value represents the number of integer corrections, and the Ø value represents the number of octal corrections. The contents of index registers 1, 2 and 4 are saved. IFLAG is set to identification integer 2149. Error switch SW(70) is set ØFF.

(2) Steps 4-9. Columns 19-21, 25-27, and 31-33 are examined to determine if they contain bEb, bIb, and bØb, respectively (b denotes blank). If any columns are in error, they are indicated by setting to one the proper bits in CLEL and SW(70) is set ØN.

(3) Steps 10-18. U41 converts the number of E, I, and Ø corrections from BCD to binary. If an error occurred in U41 (SENSE light 4 ON), SW(70) is set ØN. Otherwise the converted E, I, and Ø values are stored in an internal work area.

(4) Steps 19-21. INTRØG interrogates SW(70) to determine if any errors have occurred (SW(70) = ON). If ØN, ITYER is set to four. Otherwise the converted E, I, and Ø values are stored in NØEES, NØIIS, and NØØØS, respectively.

(5) Step 22. The contents of index registers 1, 2, and 4 are restored, and the subprogram exits to the user program.

2-191. SUBPROGRAM U45 (DECNT). DECNT is a control program for processing decimal correction cards. The FORTRAN II reference statement is CALL DECNT.

a. Inputs. The inputs are SENSE switch 5 set ON if input is on cards, or OFF if input is on tape, and the following Common registers:

COMMON TAG	DIMENSION	ITEM	UNITS
CDTYP	1	Card type	BCD
UDCVL	1	Value from decimal correction card	
ADRES	6	Tag and relative address of correction cards	BCD
JPØNT	1	Counter used for TAANT	
ISATA	2,1,1	Array dimension	integer
IDLTA	2,1,1	Array address	integer
IHLTA	2,1,1	Length of second and third index	integer
SW(74)		If ØN, DOC card indicated	
SW(80)		If ØN, DOC-DEC card indicated	
SW(82)		If ØN, direct print requested	
SW(116)		If ØN, REM card indicated	
SW(189)		If ØN, DECNT calls MTTAPE	

b. Outputs. The output is the printed and written statement:

DECNT EXPECTED A DOC-DEC CARD

c. Program Logic. FD U45

(1) Steps 1-3. SAVE4 saves the return path to the user subprogram. If DECNT is not to call MTTAPE (SW(189) = ØFF), control is transferred to step 4, otherwise MTTAPE generates the missile and trajectory (M/T) tape.

(2) Steps 4-10. IFLAG is set to identification integer 2145, and SW(126) is set ØFF. SENSE switch 5 is tested to determine if input is on cards or tape. Indicators are set accordingly so that no further tests have to be made. INTRØG interrogates SW(82) to determine if the output statement is both written and printed (SW(82) = ØN). Otherwise it is only written. Indicators are set accordingly so that no further tests have to be made.

(3) Steps 11-24. U20 reads an input card, and INTRØG interrogates SW(70) to determine if an error has occurred in U20. IFLAG is set to identification integer 2145. If SW(70) is ØN, the subprogram continues at step 55. If ØFF, SW(120) is set ØFF, and U08 writes the input card. INTRØG interrogates SW(70) to determine if an error has occurred in U08. IFLAG is set to identification integer 2145. If SW(70) is ØN, the subprogram continues at step 55. If ØFF, and direct print is not requested, the subprogram continues at step 25. Otherwise SW(120) is set ØN and U08 prints the input card. INTRØG interrogates SW(70) to determine if an error has occurred in U08. IFLAG is set to identification integer 2145. If SW(70) is ØN, the subprogram continues at step 55. If ØFF, the subprogram continues

at the next step.

(4) Steps 25-34. CDTYPE determines the type of control card, and INTRØG interrogates SW(70) to determine if an error has occurred in CDTYPE. IFLAG is set to identification integer 2145. If SW(70) is ØN, the subprogram continues at step 55. If ØFF, INTRØG interrogates SW(116) to determine if there are any remark cards. IFLAG is set to identification integer 2145. If SW(116) is ØN, the subprogram continues at step 11. If ØFF, the subprogram continues at the next step.

(5) Steps 35-36. The output statement is printed and written. The subprogram exits to HALT for manual intervention.

(6) Steps 37-48. DØCTYP determines which DOC cards are present, and INTRØG interrogates SW(70) to determine if an error has occurred in DØCTYP. IFLAG is set to identification integer 2145. If SW(70) is ØN, the subprogram continues at step 55. If ØFF, INTRØG interrogates SW(80) to determine if a DOC-DEC card is present. IFLAG is set to identification integer 2145. If SW(80) is ØN, DCARD processes DOC-DEC card. If ØFF, the subprogram continues at step 35. INTRØG interrogates SW(70) to determine if an error has occurred in DCARD. IFLAG is set to identification integer 2145. If SW(70) is ØN, the subprogram continues at step 55. If ØFF, the number of E decimal correction cards is tested with zero. If less than zero, the subprogram continues at

step 56; if equal, the subprogram continues at the next step.

(7) Step 49. The number of I decimal correction cards is tested with zero. If less than zero, the subprogram continues at step 53; if greater, the subprogram continues at step 75, if equal, the subprogram continues at the next step.

(8) Steps 50-52. The number of Ø decimal correction cards is tested with zero. If less than zero, the subprogram continues at step 53; if greater, the subprogram continues at step 91; if equal, the program continues at the next step. If DECNT is not to call MTTAPE (SW(189) = ØFF), RTRN4 returns control to the next subprogram, otherwise the subprogram continues at step 3.

(9) Steps 53-54. SW(70) is set ØN, and an 11 is stored in ITYER.

(10) Step 55. The subprogram exits to ERRPRT to print a notification of the error.

(11) Steps 56-70. If all the E decimal correction cards have been processed the subprogram continues at step 49. Otherwise CDTYP, UDCVL, and ADRES are read in. L is set to one. The contents from cards are stored in card image areas. SW(120) is set ØFF and U08 prints the input card. Information read in step 57 is written and DADDR processes the address from DEC card. INTRØG interrogates SW(70) to determine if an error has occurred in DADDR. IFLAG is set to identification integer 2145, and IACTG is set to zero.

If SW(70) is ØFF, the subprogram continues at step 71. If ØN, a 12 is stored in ITYER. SW(120) is set ØN and U08 prints the input card.

(12) Steps 71-74. The array dimension is tested with zero. If less than zero, the subprogram continues at step 53; if greater, the subprogram continues at step 72; if equal, the subprogram continues at step 73. CMPUT computes the array dimension, integers to request address in a dimensional array, and length of second and third index; STØRE stores the computed values. IFLAG is set to identification integer 2145 and the subprogram continues at step 49.

(13) Steps 75-86. If all the I decimal correction cards have been processed, the subprogram continues at step 50. Otherwise CDTYP, UDCVL, and ADRES are read in. L is set to two. The contents from cards are stored in card image areas. SW(120) is set ØFF and U08 prints the input card. Information read in step 76 is written and DADDR processes the address from the DEC card. INTRØG interrogates SW(70) to determine if an error has occurred in DADDR. IFLAG is set to identification integer 2145, and IACTG is set to zero. If SW(70) is ØN, the subprogram continues at step 68. If ØFF, the subprogram continues at the next step.

(14) Steps 87-90. The array dimension is tested with zero. If less than zero, the subprogram continues at step 53; if greater, the subprogram continues at step 88; if equal, the subprogram continues at step 89. CMPUT computes the

array dimension, integers to request address in a dimensional array, and length of second and third index; STØRE stores the computed values. IFLAG is set to identification integer 2145, and the subprogram continues at step 50.

(15) Steps 91-102. If all the Ø decimal correction cards have been processed the subprogram continues at step 51. Otherwise CDTYP, UDCVL, and ADRES are read in. The contents from cards are stored in card image areas. SW(120) is set ØFF and U08 prints the input card. Information read in step 92 is written and DADDR processes address from DEC card. INTRØG interrogates SW(70) to determine if an error has occurred in DADDR. IFLAG is set to identification integer 2145 and IACTG is set to zero. If SW(70) is ØN, the subprogram continues at step 68. If ØFF, the subprogram continues at the next step.

(16) Steps 103-106. The array dimension is tested with zero. If less than zero, the subprogram continues at step 53; if greater, the subprogram continues at step 104; if equal the subprogram continues at step 105. CMPUT computes the array dimension, integers to request address in a dimensional array, and length of second and third index; STØRE stores the computed values. IFLAG is set to identification integer 2145, and the subprogram continues at step 51.

2-192. SUBPROGRAM U10 (MTTAPE). MTTAPE produces a binary tape with missile data, azimuth limits, M constants, and delta matrices. It is one of the data tape inputs to the TTP. The FORTRAN II reference statement is CALL MTTAPE.

a. Inputs. The inputs are binary tape A2, which contains old missile trajectory M/T information if the M/T tape is to be updated, and/or the input card deck which is used to set up a large group of Common. The following are also inputs:

COMMON TAG	DIMENSION	ITEM
FINIT	2,2	Largest positive floating point number expressible in memory
SW(183)	1	If ØN, update binary tape A2

b. Outputs. The outputs are binary tape B7, which contains M/T data to be used as input to the targeting and simulation subprograms. Figure 2-4A illustrates the M/T tape structure. Table 2-1 illustrates the record format and table 2-2 shows the Common tags which correspond to the data on tape.

COMMON TAG	DIMENSION	ITEM
SW(71)	1	If ØN, convert hundredths of seconds; if ØFF, convert tenths of seconds
SW(72)	1	If ØN, GGC or TGT latitude card indicated
SW(82)	1	If ØN, direct print
SW(120)	1	If ØN, print; if OFF, write
SW(199)	1	If ØN, limits angle less than than 360 degrees