COMMON TAG	ITEM	SYMBOL	UNITS
XDEW(136)	X component of missile velocity relating to local air mass		ft/sec
XDEW(137)	Y component of missile velocity relative to local air mass		ft/sec
XDEW(138)	Z component of missile velocity relative to local air mass		ft/sec
XDEW(139)	X component of drag vector	D_x	lbs
XDEW(140)	Y component of drag vector	D_y	lbs
XDEW(141)	Z component of drag vector	D_z	lbs
XDEW(142)	Cosine of angle of attack	α	
XDEW(143)	Dynamic pressure of missile	q	lbs/ft^2
XDEW(144)		q (α)	$lbs\text{-}deg/ft^2$
XDEW(145)	Mach number		

(d) Present during powered flight:

COMMON TAG	ITEM	SYMBOL	UNITS
XDEW(147)	P-10 X, XDEW(492)		ft/sec
XDEW(148)	P-10 Y, XDEW(380)		ft/sec
XDEW(149)	P-10 Z, XDEW(464)		ft/sec
XDEW(150)	P-10 V_R/V, XDEW(624)		
XDEW(151)	P-10 $\epsilon_{\dot{c}}$, XDEW(494)		ft/sec
XDEW(152)	P-10 t_g, XDEW(658)		seconds
XDEW(153)	P-10 V, XDEW(628)		ft/sec
XDEW(154)	P-10 $\ddot{D}$ smoothed, XDEW(586)		ft/sec^2

(e) On last record of file:

COMMON TAG	ITEM	UNITS
XDEW(156)	Cross range offset	feet
XDEW(157)	Down range offset	feet

RSDIME handles three sets of range safety data simultaneously. The corresponding times t_1, t_{12}, and t_2 where $t_1 < t_{12} < t_2$ are associated with each set, XDEW(101)-XDEW(157), XDEW(201)-XDEW(257), and XDEW(301)-XDEW(357).

b. Outputs. The outputs are as follows:

COMMON TAG	ITEM	SYMBOL	UNITS
SW(14)	If ØN, bypass interpolation		
SW(13)	If ØN, bypass to process interrupt point		
IFLAG	Identification integer		integer
XDEW(698)	Re-entry vehicle separation switch		
XDEW(1)	Current category A printouts ACURR		seconds
XDEW(5)	Time difference between successive category A printout times	ΔtA	seconds

The following printed and written statements are also outputs:

a. BOOSTER ENGINE CUTOFF

b. SUSTAINER ENGINE CUTOFF

c. VERNIER ENGINE CUTOFF

d. END OF RANGE SAFETY PRINTOUT FOR THIS TARGET

e. RSDIME ERROR IN ACTION VARIABLE NAL= ______

c. Variables. The logical and timing variables (not defined in Common) are as follows:

VARIABLE	DESCRIPTION
NEØ	End-of-file indicator
NINT	Interrupt interval indicator
NACUR or NBCUR	Indicates if current category A or B printout time belongs to the current RSD interval
NASTP or NBSTP	Indicates if current category A or B printout time is still in the desired printing range
NGØTØ	Number associated with the occurrence of a derived event
ASTEP or BSTEP	Category A or Category B printout step count test

d. Program Logic. FD P38

(1) Steps 1-3. SAVE4 saves the return path to the user subprogram and IFLAG is set to identification integer 1638. Logical and timing variables are initialized to zero and SW(14) is set ØFF.

(2) Step 4. If re-entry vehicle separation has not occurred (LSEQ < 10) the subprogram continues at step 17. Otherwise the subprogram continues at step 5.

(3) Steps 5-8. If the current category A printout time ACURR is greater than or equal to the time of flight since liftoff in the past interval t_{12}, or greater than the time to stop category A printouts ASTØP; or if re-entry vehicle separation has not occurred in this interval

(LSEQ $\leq$ 9), the subprogram continues at step 9. Otherwise RSDA computes and writes category A data. The time ACURR is stepped up by the time difference between successive category A printout times ΔtA. The subprogram continues at step 5.

(4) Steps 9-16. If re-entry vehicle separation has occurred in this interval (LSEQ $>$ 9), initialization is performed for a special run without category B data and the subprogram continues at step 15. Otherwise the interrupt code is tested for two. If this is an interrupt point (code for past interval is two) and if the missile is at re-entry (LSEQ = 10), the time ACURR is set equal to the time of flight t_{12}, ΔtA is set to one, and the subprogram continues at step 17. If the missile is not at re-entry, the subprogram continues at step 17. If this is a simulation interval (code for past interval is one) and if re-entry has occurred, ΔtA is set to one; otherwise ΔtA is set to 30. The subprogram continues at step 17.

WWW.CHROMEHOOVES.NET

(5) Steps 17-19. If t_{12} is an interrupt time (interrupt code for this interval equals 2 and LSEQ for past interval equals 4, 7, or 9) logic indicator NINT is set to 16. Otherwise NINT is set to zero. The subprogram continues at step 20.

(6) Steps 20-67. If variable ASTEP is 511, time ACURR is stepped by ΔtA and ASTEP is set to zero. If variable BSTEP is 511, time BCURR is stepped up by ΔtB and BSTEP is

WWW.CHROMEHOOVES.NET

set to zero. The logic variables are set depending on the following conditions:

CONDITION	LOGIC VARIABLE
ACURR $\leq$ ASTØP	NASTP = 2
ACURR > ASTØP	NASTP = 0
BCURR $\leq$ BSTØP	NBSTP = 1
BCURR > BSTØP	NBSTP = 0
$1/2(t_1 + t_{12}) \leq$ ACURR $< 1/2(t_2 + t_{12})$	NACUR = 8
$1/2(t_2 + t_{12}) \leq$ ACURR $< 1/2(t_1 + t_{12})$	NACUR = 0
$1/2(t_1 + t_{12}) \leq$ BCURR $< 1/2(t_2 + t_{12})$	NBCUR = 4
$1/2(t_2 + t_{12}) \leq$ BCURR $< 1/2(t_1 + t_{12})$	NBCUR = 0

The logic varaible NGØTØ is computed as a function of the derived events E by use of expression (1). The action then depends on the value of NGØTØ as follows:

NGØTØ	ACTION
1,2,3,4,5,7,9,10, or 13	Cont at step 104
6,8, or 14	Cont at step 74
11,12, or 15	Cont at step 70
17,18,19,20,21,23,25,26, or 29	Cont at step 86
16	Cont at step 79

NGØTØ	ACTION
22,24, or 30	Cont at step 72
27,28, or 31	Cont at step 68

For any other values of NGØTØ, logic variable NAL is set to zero. The action then depends on the condition as follows:

CONDITION	RESULT
ACCUR > BCURR	NAL stepped by 2
ACURR = BCURR	NAL stepped by 1
ACURR > t_{12}	NAL stepped by 16
ACURR = t_{12}	NAL stepped by 8
ACURR < t_{12}	NAL stepped by 4
BCURR > t_{12}	NAL stepped by 128
BCURR = t_{12}	NAL stepped by 64
BCURR < t_{12}	NAL stepped by 32

The course of action depends on the value of NAL as follows:

NAL	ACTION
36,68, or 132	Cont at step 70
38,42, or 50	Cont at step 74
37	Cont at step 79
73	Set BSTEP = ASTEP = 511 Cont at step 86
82	Set BSTEP = 511 and cont at step 86

NAL	ACTION
136	Set ASTEP = 511 and cont at step 83
144,145, or 146	Cont at step 86

For any other values of NAL the subprogram continues at step 112.

(7) Steps 68-69. If the time of flight t_{12} is less than the time ACURR, the subprogram continues at step 86; if equal to ACURR, the subprogram continues at step 103. Otherwise the subprogram continues at step 70.

(8) Steps 70-71. RSDA computes and writes category A data. Variable ASTEP is set to 511 and the subprogram continues at step 72.

(9) Steps 72-73. If the time of flight t_{12} is less than the time BCURR, the subprogram continues at step 86; if equal to BCURR, variable BSTEP is set to 511 and the subprogram continues at step 86. Otherwise the subprogram continues at step 74.

(10) Steps 74-78. The value of ACURR is saved and ACURR is set equal to BCURR. RSDA computes and writes category A data. Logic and test variables are saved and IIP computes and writes category B data. The logic variables, test variables, and ACURR are restored. Variable BSTEP is set to 511 and the subprogram continues at step 20.

(11) Steps 79-82. If category A printout time is less

than category B printout time, the subprogram continues at step 70. If ACURR is greater than BCURR, the subprogram continues at step 74. Otherwise RSDA computes and writes category A data. Variable ASTEP is set to 511.

(12) Steps 83-85. The logic and test variables are saved. IIP computes and writes categor B data. Logic variables and test variables are restored, BSTEP is set to 511, and the subprogram continues at step 20.

(13) Steps 86-103. Categor A printout time and categor B printout time are saved. Variable ACURR and BCURR are set equal to t_{12} and SW(14) is set ØN. Output statements a, b, or c are written, depending on whether LSEQ equals 4, 7, or 9 respectively. If LSEQ ~~equals 9, RSDA computes and writes category A data.~~ is not 4, 7, or 9, BSTØP is compared with time of flight since lift off t_f, and if greater, is set equal to t_f. RSDA computes and writes category A data. Logic variables and test variables are saved. If booster cutoff time t_{BECO} is equal to t_{12}, or if t_{BECO} is less than t_{12} and BSTØP is greater than or equal to t_{12}, IIP computes and writes categor B data; logic variables, test variables, ACURR, and BCURR are restored; and NINT is set to zero. Variable ASTEP is set to 511 and the subprogram continues at step 83.

(14) Steps 104-107. If the end-of-file has not been met (NEØ ≠ 511), the subprogram continues at step 108. Otherwise output statement d is written. URSD1 counts 51 lines and restores a page, if necessar . RTRN4 returns the subprogram for RSDNT control.

(15) Steps 108-111. The RSD data blocks are aged by replacing in core memory those blocks associated with t_1 and t_{12} by the blocks associated with t_{12} and t_{21} such that the next RSD data block would then overlay, in memory, the original location of the block associated with time t_2. URSD and RSDE read in a new block of data associated with time t_2 and, if an end-of-file has been encountered, NEØ is set to 511. The subprogram continues at step 4.

(16) Steps 112-113. Statement e is written. URSD1 restores a page and counts the number of lines written and the subprogram continues at step 104.

e. Expressions.

$$\text{NGØTØ} = 1 + f(E) \qquad (1)$$

$$= 1 + f\left(\bigcap_{i=1}^{5} X_i\right)$$

$$= 1 + \sum_{i=1}^{5} X_i$$

E = derived event

$f(X_1)$ = NINT

$f(X_2)$ = NACUR

$f(X_3)$ = NBCUR

$f(X_4)$ = NASTP

$f(X_5)$ = NBSTP

NGØTØ = NINT + NACUR + NBCUR + NASTP + NBSTP + 1

2-207. SUBPROGRAM P52 (RSDNT). RSDNT is the control subprogram for the RSD package. It decides which file on tape B3 to process as a function of parameters from a file control card. It also sets up timing requirements for the entire tape based on information culled from the RSD timing card. RSDNT also diagnoses input data and records error printouts, taking the appropriate action as required. Finally RSDNT initializes parameters necessary for each target processing. This is done in one pass through a target file. After this pass, RSDNT backspaces one file, completes initializations for the current target under consideration, and then transfers control to RSDIME for a second data processing pass. When RSDIME returns control to RSDNT, RSDNT prepares for the next target, if it exists. The FORTRAN II reference statement is CALL RSDNT.

a. Inputs. The inputs are from the on-line card reader and are as follows:

COMMON TAG	ITEM	SYMBOL	UNITS
ISATA	Time to start category A printouts ASTART		seconds
IDLTA	Time difference between successive category A printouts	ΔtA	seconds
IKSTB	Time to start category B printouts BSTART		integer
IHLTA	Time to stop category A printouts ASTØP		seconds

COMMON TAG	ITEM	SYMBOL	UNITS
IHLTB	Time to stop category B printouts BSTØP		seconds
IDLTB	Time difference between successive category B printouts	ΔtB	integer
ZPST2(1)	Temporary storage for SW(13) code number		pure no.
ZPST2(2)	Temporary storage for SW(15) code number		pure no.
ZPST2(3)	Temporary storage for SW(16) code number		pure no.
ZPST2(4)	Temporary storage for SW(82) code number		pure no.
XDEW(49)	ABC flip switch		
ZPST1(1,1) ZPST1(2,10)	Radar IR Launch pad IL } number		
NFLAG(1)- NFLAG(10)	Input target data counter		integer

b. Outputs. The outputs are as follows:

COMMON TAG	ITEM	SYMBOL	UNITS
XDEW(1)	Category A printout time ACURR		seconds
XDEW(2)	Category B printout time BCURR		seconds
XDEW(3)	Time to stop category A printouts ASTØP		seconds
XDEW(4)	Time to stop category B printouts BSTØP		
XDEW(5)	Category A print time interval	ΔtA	seconds

COMMON TAG	ITEM	SYMBOL	UNITS
XDEW(6)	Category B print time interval	ΔtB	seconds
XDEW(7)	$\dot{X}_{LP}$		feet
XDEW(8)	$\dot{Y}_{LP}$		feet
XDEW(9)	$\dot{Z}_{LP}$		feet
XDEW(37)	Time to start category A printouts ASTART-per tape		seconds
XDEW(38)	Time to start category B printouts BSTART-per tape		seconds
XDEW(39)	Time to stop category A printouts ASTØP-per tape		seconds
XDEW(40)	Time to stop category B printouts BSTØP-per tape		seconds
XDEW(41)	Time difference between successive category A printouts	ΔtA-per tape	second
XDEW(42)	Time difference between successive category B printouts	ΔtB per tape	seconds
XDEW(43)			
XDEW(44)	t_{BECO} (booster cutoff time)		seconds
XDEW(45)	t_{SECO} (sustainer cutoff time)		seconds
XDEW(46)	t_{VECO} (vernier cutoff time)		secondr
XDEW(47)	t_{NCS} (re-entry vehicle separation)		seconds
XDEW(48)	SW(13)-per tape		
XDEW(49)	ACB-flip switch		

COMMON TAG	ITEM	UNITS
XDEW(57)	$t_{RE-ENTRY}$	seconds
XDEW(58)	$X_{RE-ENTRY}$	feet
XDEW(59)	$Y_{RE-ENTRY}$	feet
XDEW(60)	$Z_{RE-ENTRY}$	feet
XDEW(61)	$\dot{X}_{RE-ENTRY}$	ft/sec
XDEW(62)	$\dot{Y}_{RE-ENTRY}$	ft/sec
XDEW(63)	$\dot{Z}_{RE-ENTRY}$	ft/sec
XDEW(64)	E-error and end-of-file indicator	ft/sec
XDEW(66)	Radius R_T to target detonation point	feet
XDEW(67)	X_{RADAR}	feet
XDEW(68)	Y_{RADAR}	feet
XDEW(69)	Z_{RADAR}	feet
XDEW(70)	Longitude of radar	degrees
XDEW(73)	sin α	degrees
XDEW(75)	cos α	degrees
XDEW(84)	X_{VECO}	feet
XDEW(85)	Y_{VECO}	feet
XDEW(86)	Z_{VECO}	feet
XDEW(87)	$\dot{X}_{VECO}$	ft/sec
XDEW(88)	$\dot{Y}_{VECO}$	ft/sec
XDEW(89)	$\dot{Z}_{VECO}$	ft/sec
XDEW(91)	h_{VECO}	feet
XDEW(93)	k_1 adjustment constant	
XDEW(94)	k_2 adjustment constant	

COMMON TAG	ITEM	UNITS
XDEW(162)	Longitude of current launcher	degrees
XDEW(400)	Geographic latitude L_{GL} at launch pad	degrees
XDEW(401)	Geocentric latitude L'_{GL} at launch pad	degrees
XDEW(402)	Earth radius RL	feet
XDEW(403)	RSD flag	
XDEW(404)	SGNL to initialize IIP	
XDEW(407)	Convert feet to nautical miles	naut mi
XDEW(427)	XT	
XDEW(428)	YT	
XDEW(429)	ZT	
ZPST3(29) ZPST3(30) ZPST3(31)	X_{BECO} Y_{BECO} Z_{BECO} } Missile position at booster cutoff	feet
ZPST3(32) ZPST3(33) ZPST3(34)	$\dot{X}_{BECO}$ $\dot{Y}_{BECO}$ $\dot{Z}_{BECO}$ } Missile velocity at booster cutoff	ft/sec
ZPST3(35)	t_{BECO}	seconds
RSLTS(1)-RSLTS(4)	Output data matrix	

The following printed and written statements are also outputs:

a. WRONG CARD TYPE RSD CANNOT PROCEED

b. ERROR IN CARD RSD CANNOT PROCEED

c. INPUT CARD ERROR - ASTART EXCEEDS ASTOP

d. INPUT CARD ERROR - BSTART EXCEEDS BSTOP

e. CARD ERROR - DLTB/DLTA NOT INTEGRAL

f. INPUT ERROR - (ASTART-BSTART)/DLTA - NOT INTEGRAL

g. ASTART OR BSTART, IN ERROR, HAVE BEEN ADJUSTED

c. Program Logic. FD P52

(1) Steps 1-7. SAVE4 saves the return path to the user subprogram. IFLAG is set to identification integer 1652 and the pseudo end-of-file register is set to five. The input data table number three of past values (ZPST3 array) is initialized to zero. Arrays ZNØW, ZPST1, and ZPST2 are initialized to zero and U20 reads the RSD timing card. INTRØG interrogates SW(70) to determine if an error occurred in reading the RSD timing card. If ØN, the subprogram continues at step 8. If ØFF, the subprogram continues at step 9.

(2) Step 8. ERRPRT prints a notification of the error and the subprogram halts for manual intervention.

(3) Steps 9-31. SW(120) is set ØFF and U08 writes the contents of the RSD timing card. INTRØG interrogates SW(70) to determine if an error occurred in writing. If ØN, the subprogram continues at step 8. If ØFF, CDTYPE determines the type of control card present. INTRØG interrogates SW(118) to determine if the control card is an RSD timing card (SW(118) = ØN). If ØFF, statement a is written and printed and the subprogram continues at step 45. If SW(118) is ØN, RSDTIM converts the RSD timing card information to

floating point form and INTRØG interrogates SW(70). If ØN, statement b is written and printed and the subprogram continues at step 8. If SW(70) is ØFF, SW(54) and SW(55) are saved, ITAPE is initialized to 13 and IWHAT is initialized to one. Tape parameters and radar parameters are computed for the entire tape. If the time to stop category A printouts ASTØP is less than the time to start category A printouts ASTART, statement c is written and printed and the subprogram continues at step 45. If BSTØP is less than BSTART statement d is written and printed and the subprogram continues at step 45. If the category B print interval ΔtB is not an integral of the category A print interval ΔtA, statement e is written and printed. If the starting time difference between category A printouts and category B printouts is not an integral of the category A interval ΔtA, statement f is written and printed. The subprogram continues at step 32.

(4) Step 32. ILØØP, I, IJINIT, and IDEW are initialized to zero.

(5) Steps 33-36. If the RSD mode of operation is complete ($E \geq 5$), the subprogram continues at step 44. If not, ILØØP is stepped by one. If ILØØP is greater than three, the subprogram continues at step 44. If the error indicator is zero, the subprogram continues at step 46. Otherwise, the subprogram continues at step 37.

(6) Steps 37-43. URSD reads the first or next record

of high density input tape B3. The error and end-of-file indicator and XDEW(101) are initialized and RSDE splits the input information from the GRASE array into five distinct groups for RSD processing. If XDEW(101) is equal to one, BSREC backspaces tape ITAPE one record and the subprogram continues at step 46. If the error and end-of-file indicator is less than five, the subprogram continues at step 37. Otherwise, the subprogram continues at step 44.

(7) Step 44. Tape B3 is rewound.

(8) Step 45. RTRN4 causes return to the user subprogram along the path established by SAVE4.

(9) Step 46. URSD1 counts the number of lines written and restores the page when necessary.

(10) Steps 47-67. URSD and RSDE read a record from tape B3 into block one, splitting the information from the GRASE array into five distinct groups for RSD processing. The values from the past integration interval, block two, are set to the values from the present integration interval, block one. URSD and RSDE read a record from tape B3 into block one, splitting the information from the GRASE array into five distinct groups for RSD processing. Values for the second past integration interval, block three, are set up by extrapolating backwards, and the end-of-file indicator is initialized. If ASTART or BSTART is less than the time in the extrapolated integration interval, block three,

statement g is written and printed. Parameters are initialized for RSDIME, and range safety parameters are set up. LGTØLC and ELLRAD compute the geocentric latitude as a function of the geographic latitude and then compute the earth ellipsoid radius as a function of the geocentric latitude. RSDBAA, SINE, and CØSINE are used in computing the approximate target coordinate XT, YT, ZT for time zero. If the radar or launch pad geocentric longitude is less than zero, an adjustment is made. The difference in radar and launch pad longitude is computed and converted to radians, θ. RSDBAA, SINE, and CØSINE compute sin θ, cos θ, sin L'_{GL}, and cos L'_{GL} in double precision. Expressions (1)-(3) compute the inertial launch coordinate X'_{LP}, Y'_{LP}, Z'_{LP}. Adjustments are made for ACURR, BCURR, or BSTØP is necessary, and time flags are reset to zero. The contents of the ZPST1 registers are saved and RSDIME performs the second pass for the current target run and interpolates category A and B independently, in accordance with the RSD timing card. The subprogram continues at step 33.

d. Expressions.

$$X'_{LP} = RP \cos L'_{GL} \cos \theta \quad (1)$$

$$Y'_{LP} = RP \cos L'_{GL} \sin \theta \quad (2)$$

$$Z'_{LP} = RP \sin L'_{GL} \quad (3)$$

2-209. SUBPROGRAM U29(RSDTIM). RSDTIM in conjunction with U41, converts the RSD time card from BCD to floating point form and stores it in the Common Area. The FORTRAN II reference statement is CALL RSDTIM.

a. Inputs. The input is an RSD time card in BCD in the card image area, registers CDIO-CDI9. Figure 2-1 illustrates the format of the RSD time card. The bit configuration of these columns in the card image area CDIO-CDI13 is as follows:

	S 5	6 11	12 17	18 23	24 29	30 35
CDIO	1	2	3	4	5	6
CDI1	7	8	9	10	11	12
CDI2	13	14	15	16	17	18
CDI3	19	20	21	22	23	24
CDI4	25	26	27	28	29	30
CDI5	31	32	33	34	35	36
CDI6	37	38	39	40	41	42
CDI7	43	44	45	46	47	48
CDI8	49	50	51	52	53	54
CDI9	55	56	57	58	59	60
CDI10	61	62	63	64	65	66
CDI11	67	68	69	70	71	72
CDI12	73	74	75	76	77	78
CDI13	79	80				

b. Outputs. The output of RSDTIM is the RSD timing information stored in floating binary form in the following Common registers:

REGISTER	DATA
ISATA and ISATA-1	Start A time
IDLTA and IDLTA-1	A printout interval
IHLTA and IHLTA-1	Stop A time
IKSTB and IKSTB-1	Start B time
IKDTB and IKDTB-1	B printout interval
IHLTB and IHLTB-1	Stop B time

c. Program Logic. FD U29

(1) Steps 1-2. The contents of index registers 1, 2, and 4 are saved. Zeros are stored in CLEL and CLER, and the error indicator SW(70) is set ØFF.

(2) Steps 3-8. Columns 19, 21, 26, and 31 are examined one at a time for an A, S, D, and H respectively. If the letters do not appear, the column or columns in error are indicated by setting to one the corresponding bits in CLEL, and SW(70) is set ØN. After the above four columns have been examined, columns 25, 30, and 35 are examined one at a time for blanks. If the columns are correct, the subprogram continues at the next step. Otherwise the column or columns in error are indicated in CLEL, SW(70) is set ØN, and the subprogram continues at the next step.

(3) Steps 9-14. Columns 48, 54, and 59 are examined for blanks. Columns 41, 43 and 49, 50, and 55 are examined for a B, K, D, and H respectively. If the columns are correct,

the subprogram continues at the next step. Otherwise the column or columns in error are indicated in CLER, SW(70) is set ØN, and the subprogram continues at the next step.

(4) Steps 15-21. U41 converts columns 22-24, 27-29, 32-34, 51-53, and 56-58 from BCD to floating binary form. The converted data is stored in the GRASE area. After each conversion, SENSE light 4 is examined for any error in U41. If ON an error occurred, and SW(70) is set ØN. U41 converts columns 44-47 to floating point binary last. SENSE light 4 is interrogated and, if ON, SW(70) is set ØN.

(5) Steps 22-24. SW(70) is interrogated for any error in U41 or in the information on the RSD time card. If ØFF no errors have occurred and control is transferred to step 26. If ØN, a four is stored in ITYER, and IFLAG is set to identification 2129.

(6) Step 25. The contents of the index registers are restored and the subprogram exits to the user subprogram.

(7) Steps 26-27. The converted data is stored in the following registers:

DATA	REGISTERS
B start time (col 44-47)	IKSTB and IKSTB-1
A start time (col 22-24)	ISATA and ISATA-1
Interval of A printout (col 27-29)	IDLTA and IDLTA-1
A stop time (col 32-34)	IHLTA and IHLTA-1
Interval of B printout (col 51-53)	IKDTB and IKDTB-1

DATA	REGISTERS
B stop time (col 56-58)	IHLTB and IHLTB-1

The RSD time card input is logged in UCARD, and control is transferred to step 25.

2-210. SUBPROGRAM P95 (URSD). URSD reads the high density input tape B3 into core a record at a time. The FORTRAN II reference statement is CALL URSD.

a. Inputs. No inputs are defined.

b. Outputs. The outputs are as follows:

COMMON TAG	DIMENSION	ITEM	UNITS
GRASE	100	Storage block containing words of record	
FRASE	319	Temporary storage used for transmission of words from tape	
IRECR	5	Tape record number	integer
IFILE	4300	Current tape file number	integer
NFLAG	16	Error and end-of-file indicator	integer

c. Program Logic. FD P95

(1) Steps 1-4. The contents of the index registers are saved and storage blocks used for transmission of information are cleared to zero. The end-of-file indicator is set ØFF and binary tape B3 is selected. The subprogram continues at step 5.

(2) Steps 5-6. The first (next) word is read into FRASE from tape B3 and the end-of-file indicator is tested. If the end-of-file indicator is ØFF, the subprogram continues at step 8, otherwise the subprogram continues at the next step.

(3) Step 7. The error and end-of-file indicator E is set to five or six depending on whether the end-of-file was good or bad. The subprogram continues at step 16.

(4) Steps 8-15. If all words of the current record have not been copied, the subprogram continues at step 5. Otherwise the record number, file number, and accumulated checksum are saved. The computed checksum is compared with the stored checksum for agreement. If the checksums do not agree, the error and end-of-file indicator is set to two to indicate checksum error and the subprogram continues at step 7. The subprogram initializes for storage of record words and the words are stored in GRASE. The error and end-of-file indicator E is set to one to indicate a good record and the subprogram continues at step 16.

(5) Step 16. The index registers are restored to their original contents and the subprogram returns to the user subprogram.

2-211. SUBPROGRAM P96 (URSD1). URSD1 controls output spacing and ejects to a new page when necessary. The FORTRAN II reference statement is CALL URSD1 (STEP).

a. Inputs. The input is the control variable STEP which determines the number of lines to be counted. XDEW(10) is defined as the total number of lines already printed.

b. Outputs. No outputs are defined.

c. Program Logic. If the total number of lines printed so far is less than or equal to 50, the number is stepped by STEP and the subprogram returns to the user subprogram. Otherwise the total number of lines is reset to the value STEP, a new page is restored, and the subprogram returns to the user subprogram.

2-212. SUBPROGRAM D26 (XYZRSD). XYZRSD converts position and velocity components in the flight simulation inertial system to the range safety coordinate system. The FORTRAN II reference statement is CALL XYZRSD.

a. Inputs. The inputs are as follows:

COMMON TAG	ITEM	SYMBOL	UNITS
XDEW(1)	Time of flight since liftoff	t_f	seconds
XDEW(7)	X coordinate of launch pad	X'_{LP}	feet
XDEW(8)	Y coordinate of launch pad	Y'_{LP}	feet
XDEW(9)	Z coordinate of launch pad	Z'_{LP}	feet
XDEW(107)	Launch azimuth	A_L	degrees
XDEW(400)	Geographic latitude of launch pad	L_{GL}	degrees
ZNØW(1)	X inertial position coordinate	X	feet
ZNØW(2)	Y inertial position coordinate	Y	feet
ZNØW(3)	Z inertial position coordinate	Z	feet
ZNØW(4)	X inertial velocity coordinate	$\dot{X}$	ft/sec
ZNØW(5)	Y inertial velocity coordinate	$\dot{Y}$	ft/sec
ZNØW(6)	Z inertial velocity coordinate	$\dot{Z}$	ft/sec
FPI(1)		π	
GØMGA(1)	Rate of earth rotation	Ω	rad/sec

b. Outputs. The outputs are as follows:

COMMON TAG	ITEM	SYMBOL	UNITS
XDEW(50)	X range safety position coordinate	X_{RSD}	feet
XDEW(51)	Y range safety position coordinate	Y_{RSD}	feet
XDEW(52)	Z range safety position coordinate	Z_{RSD}	feet
XDEW(53)	X range safety velocity coordinate	$\dot{X}_{RSD}$	ft/sec
XDEW(54)	Y range safety velocity coordinate	$\dot{Y}_{RSD}$	ft/sec
XDEW(55)	Z range safety velocity coordinate	$\dot{Z}_{RSD}$	ft/sec

c. Program Logic. FD D26

(1) Steps 1-9. IFLAG is set to identification integer 426. SINE, CØSINE, and RØUND compute sin Ωt_f and cos Ωt_f in single precision. The earth-centered inertial coordinates X, Y, Z are converted to earth-centered, earth-fixed coordinates X', Y', Z' with associated velocities $\dot{X}'$, $\dot{Y}'$, $\dot{Z}'$ by use of expressions (1)-(6). Expressions (7)-(12) translate the new coordinates to the launch site to form X", Y", Z" and $\dot{X}''$, $\dot{Y}''$, $\dot{Z}''$. SINE, CØSINE and RØUND compute sin L_{GL} and cos L_{GL} in single precision. Expressions (13)-(18) determine rectangular earth-fixed coordinates X''', Y''', Z''', and $\dot{X}'''$, $\dot{Y}'''$, $\dot{Z}'''$ at the launch site. Expressions (19)-(24) determine X'''', Y'''', Z'''' and $\dot{X}''''$, $\dot{Y}''''$, $\dot{Z}''''$ with Z''' perpendicular to the local horizontal at

the launch site. CØSINE and SINE compute cosine and sine of A_L and RØUND rounds these values to single precision. Expressions (25)-(30) determine the position X_{RSD}, Y_{RSD}, Z_{RSD} and velocity $\dot{X}_{RSD}$, $\dot{Y}_{RSD}$, $\dot{Z}_{RSD}$ components of the missile in the range safety coordinate system. This system is right-handed, earth-fixed, and rectangular with center at the launch site. The Z_{RSD} coordinate is perpendicular to the horizontal at the launch site, and the X_{RSD} coordinate is in the direction of the launch azimuth. The subprogram returns to the user subprogram.

d. Expressions.

$$X' = X \cos \Omega t_f + Y \sin \Omega t_f \quad (1)$$

$$Y' = Y \cos \Omega t_f - X \sin \Omega t_f \quad (2)$$

$$Z' = Z \quad (3)$$

$$\dot{X}' = \dot{X} \cos \Omega t_f + \dot{Y} \sin \Omega t_f - \Omega X \sin \Omega t_f + \Omega Y \cos \Omega t_f \quad (4)$$

$$\dot{Y}' = \dot{Y} \cos \Omega t_f - \dot{X} \sin \Omega t_f - \Omega Y \sin \Omega t_f - \Omega X \cos \Omega t_f \quad (5)$$

$$\dot{Z}' = \dot{Z} \quad (6)$$

$$X'' = X' - X'_{LP} \quad (7)$$

$$Y'' = Y' - Y'_{LP} \quad (8)$$

$$Z'' = Z' - Z'_{LP} \quad (9)$$

$$\dot{X}'' = \dot{X}' \quad (10)$$

$$\dot{Y}'' = \dot{Y}' \quad (11)$$

$$\dot{Z}'' = \dot{Z}' \quad (12)$$

$$X''' = X'' \sin L_{GL} - Z'' \cos L_{GL} \quad (13)$$

$$Y''' = Y'' \quad (14)$$

$$Z''' = Z'' \sin L_{GL} + X'' \cos L_{GL} \quad (15)$$

$$\dot{X}''' = \dot{X}'' \sin L_{GL} - \dot{Z}'' \cos L_{GL} \quad (16)$$

$$\dot{Y}''' = \dot{Y}'' \quad (17)$$

$$\dot{Z}''' = \dot{Z}'' \sin L_{GL} + \dot{X}'' \cos L_{GL} \quad (18)$$

$$X'''' = X''' \quad (19)$$

$$Y'''' = Y''' \quad (20)$$

$$Z'''' = Z''' \quad (21)$$

$$\dot{X}'''' = \dot{X}''' \quad (22)$$

$$\dot{Y}'''' = \dot{Y}''' \quad (23)$$

$$\dot{Z}'''' = \dot{Z}''' \quad (24)$$

$$X_{RSD} = -X'''' \cos A_L + Y'''' \sin A_L \quad (25)$$

$$Y_{RSD} = -Y'''' \cos A_L - X'''' \sin A_L \quad (26)$$

$$Z_{RSD} = Z'''' \quad (27)$$

$$\dot{X}_{RSD} = -\dot{X}'''' \cos A_L + \dot{Y}'''' \sin A_L \quad (28)$$

$$\dot{Y}_{RSD} = -\dot{Y}'''' \cos A_L - \dot{X}'''' \sin A_L \quad (29)$$

$$\dot{Z}_{RSD} = \dot{Z}'''' \quad (30)$$

2-213. B8 SUBPROGRAMS.

2-214. The subprograms described in this area are required for data modification and missile system simulation control.

2-215. Subprogram L08 (BENTRY) enables the loader to establish linkage between the B1 and B8 subprograms. This version of BENTRY will be in core only when the SIM mode of operation is requested. The return path of the user subprogram is saved by SAVE4 and SIMNT is called. After SIM has been completed, the subprogram exits to DØCNT through RTRN4.

a.	ADDØUT	P02	Reads Additional Outputs Requested
b.	BENTRY	*L08	Establish B8 Control Area
c.	MET	P74	Interpret Meteorological Data Cards
d.	RENTRY	D03	Read Re-entry Data
e.	SBALM	P72	Ballistic and Re-entry Phase of Simulation
f.	SCLØM	P75	Closed Loop Simulation
g.	SIDRM	P70	Ideal-Earth Herget Solution
h.	SIMNT	P71	Missile System Simulation Control
i.	SØPNM	P73	Open Loop Simulation
j.	SPERM	P77	Perturbation Phase of Simulation
k.	TCØNST	P79	Read Switch Indications and T Constants
l.	WRITE	P76	Write Simulation Output

WWW.CHROMEHOOVES.NET

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

WWW.CHROMEHOOVES.NET