2-273. SUBPROGRAM G41 (SUSIC). SUSIC performs sustainer initialization. The FORTRAN II reference statement is CALL SUSIC.

a. Inputs. The inputs are as follows

COMMON TAG	SYMBOL	UNITS
XS(205), XS(206)	S_{103}	
. .	.	
. .	.	
XS(221), XS(222)	S_{111}	
XC(28)	C_{14}	
XS(224)	S_{112}	quanta/cy
XS(226)	S_{113}	quanta/cy
XM(72)	M_{36}	ft/sec

b. Outputs. The outputs are as follows:

COMMON TAG	SYMBOL	UNITS
XDEW(69), XDEW(70)	$\bar{\dot{\epsilon}}_z^{k}$	ft/sec
XDEW(71), XDEW(72)	$\bar{\dot{\epsilon}}_z^{k-1}$	ft/sec
XDEW(73), XDEW(74)	$\bar{\dot{\epsilon}}_z^{k-2}$	ft/sec
XDEW(75), XDEW(76)	$\bar{\dot{\epsilon}}_z^{k-3}$	ft/sec
XDEW(77), XDEW(78)	$\bar{\dot{\epsilon}}_c^{k}$	ft/sec
XDEW(79), XDEW(80)	$\bar{\dot{\epsilon}}_c^{k-1}$	ft/sec
XDEW(81), XDEW(82)	$\bar{\dot{\epsilon}}_c^{k-2}$	ft/sec
XDEW(83), XDEW(84)	$\bar{\dot{\epsilon}}_c^{k-3}$	ft/sec
XDEW(493), XDEW(494)	$\bar{\dot{\epsilon}}_c^{k}$	ft/sec
XDEW(513), XDEW(514)	$\bar{\dot{\epsilon}}_z^{k}$	ft/sec

COMMON TAG	SYMBOL	UNITS
XDEW(533), XDEW(534)	$\dot{\theta}_o^{\,k}$	$\frac{\text{quanta ft}}{\text{sec}^2\text{ cy}}$
XDEW(535), XDEW(536)	$\dot{\theta}_o^{\,k-1}$	$\frac{\text{quanta ft}}{\text{sec}^2\text{ cy}}$
XDEW(537), XDEW(538)	$\dot{\theta}_o^{\,k-2}$	$\frac{\text{quanta ft}}{\text{sec}^2\text{ cy}}$
XDEW(539), XDEW(540)	$\dot{\theta}_o^{\,k-3}$	$\frac{\text{quanta ft}}{\text{sec}^2\text{ cy}}$
XDEW(541), XDEW(542)	$\dot{\Psi}_o^{\,k}$	$\frac{\text{quanta ft}}{\text{sec}^2\text{ cy}}$
XDEW(543), XDEW(544)	$\dot{\Psi}_o^{\,k-1}$	$\frac{\text{quanta ft}}{\text{sec}^2\text{ cy}}$
XDEW(545), XDEW(546)	$\dot{\Psi}_o^{\,k-2}$	$\frac{\text{quanta ft}}{\text{sec}^2\text{ cy}}$
XDEW(547), XDEW(548)	$\dot{\Psi}_o^{\,k-3}$	$\frac{\text{quanta ft}}{\text{sec}^2\text{ cy}}$
XDEW(667), XDEW(668)	$\theta_A^{\,k}$	quanta/cy
XDEW(671), XDEW(672)	$\theta_B^{\,k}$	quanta/cy
XDEW(681), XDEW(682)	$\Psi_A^{\,k}$	quanta/cy
XDEW(685), XDEW(686)	$\Psi_B^{\,k}$	quanta/cy
XDEW(636)	$\dot{C}_W^{\,k}$	
XDEW(638)	$\dot{C}_W^{\,k-1}$	
XDEW(642)	$\dot{Z}_W^{\,k}$	
XDEW(644)	$\dot{Z}_W^{\,k-1}$	
NFLAG(4)	q (substage cycle counter)	
XC(61), XC(62)	C_{31}	
. .	.	
XC(77), XC(78)	C_{39}	
XC(26)	C_{13}	
XC(82)	C_{41}	
.	.	
XC(88)	C_{44}	

c. <u>Program Logic</u>. IFLAG is set to identification integer 741. SUSIC performs the following expressions for sustainer initialization.

$$C_{31} = S_{103},\ C_{32} = S_{104},\ C_{33} = S_{105},\ C_{34} = S_{106},$$
$$C_{35} = S_{107},\ C_{36} = S_{108},\ C_{37} = S_{109},\ C_{38} = S_{110},$$
$$C_{39} = S_{111},\ C_{13} = C_{14},\ C_{41} = 0,\ C_{42} = 0,\ C_{43} = S_{112}$$
$$C_{44} = S_{113}$$
$$\theta_A^{\,k} = \theta_B^{\,k} = \psi_A^{\,k} = \psi_B^{\,k} = \dot{Z}_W^{\,k} = \dot{Z}_W^{\,k-1} = 0$$
$$\epsilon_{\dot{Z}}^{k-1} = \dot{\theta}_o^{k-1} = \epsilon_{\dot{c}}^{k-1} = 0 \text{ for } i = 1,2,3$$
$$\dot{C}_W^{\,k-1} = \dot{C}_W^{\,k} = 0$$

Substage cycle counter q is set to zero. CUTIE is stepped by one and control is returned to the user subprogram.

2-274. SUBPROGRAM G47 (TEST). TEST checks certain duplexed quantities for agreement or values of P as determined by the setting of SW(60). The FORTRAN II reference statement is CALL TEST.

a. Inputs. The inputs are the setting of SW(60) and the registers to be checked for agreement.

b. Outputs. No outputs are defined since TEST performs no computations.

c. Program Logic.

(1) IFLAG is set to identification integer 1610. SW(60) is interrogated for an ØFF or ØN condition.

(2) SW(60) ØFF. Three values of P are checked. If any two values agree, TEST sets the third equal to this common value. If all three disagree, TEST sets IFLAG to the identification integer 1610 and calls RLLBCK for return to the previous checkpoint.

(3) SW(60) ØN. Certain portions of the XSTØR array are checked. If any values do not agree within a specified tolerance, IFLAG is set to identification integer 1610 and RLLBCK is called.

(4) If the subprogram is completed successfully, CUT is stepped by one and control is returned to the user subprogram.

2-275. SUBPROGRAM G06 (TFLYT). TFLYT computes time of flight. The FORTRAN II reference statement is CALL TFLYT.

a. Inputs. The inputs are as follows:

COMMON TAG	SYMBOL		UNITS
XDEW(4)	b^{k-1}	Aged upon entry.	$feet^{-1/2}$
XDEW(14)	f^{k-1}		pure no.
XDEW(24)	t_f^{k-1}		seconds
XDEW(254)	R_M^{k-1}		feet
XDEW(612)	$(R_M \cdot V_N)^{k-1}$		ft^2/sec
XDEW(618)	$(V^{k-1})^2$		ft^2/sec^2
XDEW(622)	$(V_R)^{k-1\ 2}$		ft^2/sec^2
XDEW(624)	$(V_R)^{k-1}$		ft/sec
XDEW(628)	V^k		ft/sec
XC(2)	C_1		feet
XS(78)	S_{39}		$sec/feet^{3/2}$
XS(80)	S_{40}		$sec/feet^{3/2}$
XS(82)	S_{41}		sec/cy
XS(148)	S_{74}		sec^2/ft^3
FPI(2)	π		radians
NFLAG(10)	P		integer

b. Outputs. The outputs are as follows:

COMMON TAG	SYMBOL	UNITS
XDEW(2)	a^k	pure no.
XDEW(4)	b^k	$feet^{1/2}$

COMMON TAG	SYMBOL	UNITS
XDEW(6)	b^{k-1}	$feet^{1/2}$
XDEW(8)	c^k	pure no.
XDEW(10)	d^k	pure no.
XDEW(12)	e^k	pure no.
XDEW(14)	f^k	pure no.
XDEW(16)	f^{k-1}	pure no.
XDEW(18)	h^k	pure no.
XDEW(20)	i^k	pure no.
XDEW(22)	g^k	radians
XDEW(24)	t^k_{fk-1}	seconds
XDEW(26)	t_f	seconds

c. <u>Program Logic</u>. IFLAG is set to identification integer 706. If $P \geq 7$, the values b, f, and t_f are aged. The following expressions are performed to determine a new time of flight:

$$a^k = (R_M^{k-1} V_R^{k-1} S_{74}) - 1$$

CALL SQDEW(B, b^{k-1}, B_1)

$$B = (1 - a^k)/R_M^{k-1}$$

$$b^k = B_1$$

$$c^k = (R_M \cdot V_N)^{k-1} S_{39} (V_R)^{k-1} b^k / V^k$$

$$d^k = 1 - C_1 (b^k)^2$$

$$e^k = (a^k)^2 + (c^k)^2$$

CALL SQDEW (B, f^{k-1}, B_1)

$f^k = -B_1$

$h^k = (a^k d^k + c^k f^k)/e^k$

$i^k = (a^k f^k - c^k d^k)/e^k$

$g^k = 0$	if $i^k = 0$ and $h^k \geq 0$
$g^k = \pi$	if $i^k = 0$ and $h^k < 0$
$g^k = 3/2\ \pi$	if i^k 0 and $h^k = 0$
$g^k = \pi/2$	if i^k 0 and $h^k = 0$
$g^k = \arctan(i^k/h^k)$ + constant	if i^k 0 and $h^k \neq 0$
constant = 0	if $\arctan(i^k/h^k) > 0$
constant = π	if $\arctan(i^k/h^k) \leq 0$

$t_f{}^k = \left[(g^k - f^k + c^k)/(b^k)3\right] S_{40} - S_{41}$

In the preceding computations SQDEW performs the square root function of B. ARCTAN performs the inverse tangent function. CUTIE is stepped by one and control is returned to the user subprogram.

2-276. SUBPROGRAM G42 (VRNIC). VRNIC performs vernier initialization. The FORTRAN II reference statement is CALL VRNIC.

a. Inputs. The inputs are as follows:

COMMON TAG	ITEM	UNITS
XDEW(250)	$\ddot{Z}_g^k$	ft/sec-cy
XDEW(380)	$\dot{Y}_1^k$	ft/sec
XDEW(554)	U_y^k	pure no.
XDEW(558)	U_z^k	pure no.
XDEW(592)	$\dot{Z}_{WN}^k$	ft/sec
XDEW(642)	$\dot{Z}_W^k$	ft/sec
XDEW(648)	a_V^k	ft/sec-cy
XC(32)	C_{16}	rad/sec
XC(34)	C_{17}	rad/sec
XC(36)	C_{18}	rad/sec
XS(228)	S_{114}	
.	.	
.	.	
XS(238)	S_{119}	
XS(239), XS(240)	S_{120}	
.	.	
.	.	
XS(259), XS(260)	S_{130}	

b. Outputs. The outputs are as follows:

COMMON TAG	ITEM	UNITS
XDEW(69), XDEW(70)	$\bar{\dot{\epsilon}}_Z^{k}$	ft/sec
XDEW(71), XDEW(72)	$\bar{\dot{\epsilon}}_Z^{k-1}$	ft/sec
XDEW(73), XDEW(74)	$\bar{\dot{\epsilon}}_Z^{k-2}$	ft/sec
XDEW(75), XDEW(76)	$\bar{\dot{\epsilon}}_Z^{k-3}$	ft/sec
XDEW(77), XDEW(78)	$\bar{\dot{\epsilon}}_C^{k}$	ft/sec
XDEW(79), XDEW(80)	$\bar{\dot{\epsilon}}_C^{k-1}$	ft/sec
XDEW(81), XDEW(82)	$\bar{\dot{\epsilon}}_C^{k-2}$	ft/sec
XDEW(83), XDEW(84)	$\bar{\dot{\epsilon}}_C^{k-3}$	ft/sec
XDEW(376)	$\ddot{Y}_1^{k}$	ft/sec-cy
XDEW(378)	$\ddot{Y}_1^{k-1}$	ft/sec-cy
XDEW(382)	$\dot{Z}_2^{k}$	ft/sec
XDEW(384)	$\dot{Z}_2^{k-1}$	ft/sec
XDEW(386)	$\dot{Z}_{2a}^{k}$	ft/sec
XDEW(388)	$\dot{Z}_{2A}^{k-1}$	ft/sec
XDEW(390)	$\ddot{Y}_{1A}^{k}$	ft/sec
XDEW(392)	$\ddot{Y}_{1A}^{k-1}$	ft/sec-cy
XDEW(394)	$\ddot{Y}_{1B}^{k}$	ft/sec-cy

COMMON TAG	ITEM	UNITS
XDEW(396)	$\ddot{Y}_{1B}^{k-1}$	ft/sec-cy
XDEW(406)	$\dot{Y}_{DR}^{k}$	ft/sec
XDEW(460)	$\dot{Z}_{g}^{k}$	ft/sec
XDEW(462)	$\dot{Z}_{g}^{k-1}$	ft/sec
XDEW(464)	$\dot{Z}_{1}^{k}$	ft/sec
XDEW(466)	$\dot{Z}_{1}^{k-1}$	ft/sec
XDEW(493), XDEW(494)	$\dot{\epsilon}_{C}^{k}$	ft/sec
XDEW(513), XDEW(514)	$\dot{\epsilon}_{Z}^{k}$	ft/sec
XDEW(533), XDEW(534)	$\dot{\theta}_{o}^{k}$	quanta ft/sec^2 cy
XDEW(535), XDEW(536)	$\dot{\theta}_{o}^{k-1}$	quanta ft/sec^2 cy
XDEW(537), XDEW(538)	$\dot{\theta}_{o}^{k-2}$	quanta ft/sec^2 cy
XDEW(539), XDEW(540)	$\dot{\theta}_{o}^{k-3}$	quanta ft/sec^2 cy
XDEW(541), XDEW(542)	$\dot{\psi}_{o}^{k}$	quanta ft/sec^2 cy
XDEW(543), XDEW(544)	$\dot{\psi}_{o}^{k-1}$	quanta ft/sec^2 cy
XDEW(545), XDEW(546)	$\dot{\psi}_{o}^{k-2}$	quanta ft/sec^2 cy
XDEW(547), XDEW(548)	$\dot{\psi}_{o}^{k-3}$	quanta ft/sec^2 cy
XDEW(667), XDEW(668)	θ_{A}^{k}	quanta/cy
XDEW(671), XDEW(672)	θ_{B}^{k}	quanta/cy
XDEW(681), XDEW(682)	ψ_{A}^{k}	quanta/cy
XDEW(685), XDEW(686)	ψ_{B}^{k}	quanta/cy
XC(56)	C_{28}	ft/sec-cy

COMMON TAG	ITEM	UNITS
XC(58)	C_{29}	ft/sec-cy
XC(60)	C_{30}	ft/sec-cy
XC(61), XC(62)	C_{31}	
. .	.	
XC(77), XC(78)	C_{39}	
XC(85), XC(86)	C_{43}	quanta/cy
XC(87), XC(88)	C_{44}	quanta/cy
NFLAG(4)	q	cycle

c. <u>Program Logic</u>. IFLAG is set to identification integer 742. The following computations are performed for vernier initialization:

$$C_{28} = a_V{}^k \left(U_y{}^k S_{114} - U_z{}^k S_{115}\right)$$

$$C_{29} = a_V{}^k \left(U_z{}^k S_{114} + U_y{}^k S_{115}\right) - 2C_{16}\dot{Y}_1{}^k S_{116}$$

$$C_{30} = S_{117} \Big[S_{118}\, a_V{}^k C_{28} + 2C_{16}\left(C_{29} + \ddot{Z}_g\right) S_{119} + 3\dot{Y}_1{}^k \left[(C_{16})^2 + (C_{18})^2\right] - 3\left(\dot{Z}_{WN}{}^k + \dot{Z}_W{}^k\right)(C_{17})(C_{18})\Big]$$

$$\dot{Z}_g{}^k = \dot{Z}_g{}^{k-1} = 0$$

$$q = 0$$

$$\ddot{Y}_{1A}^{k} = \ddot{Y}_{1A}^{k-1} = 0$$

$$\ddot{Y}_{1b}^{k} = \ddot{Y}_{1b}^{k-1} = 0$$

$$\ddot{Y}_{1}^{k} = \ddot{Y}_{1}^{k-1} = 0$$

$$\dot{Z}_{2A}^{k} = \dot{Z}_{2A}^{k-1} = 0$$

$$\dot{Z}_{2}^{k} = \dot{Z}_{2}^{k-1} = 0$$

$$\dot{Z}_{1}^{k} = \dot{Z}_{1}^{k-1} = \dot{Z}_{WN}^{k}$$

$$\dot{Y}_{DR}^{k} = 0$$

$$\bar{\epsilon}_{\dot{Z}}^{k-i} = \bar{\epsilon}_{\dot{C}}^{k-i} = \dot{\theta}_{o}^{k-i} = \dot{\Psi}_{o}^{k-i} = 0 \text{ for } i \doteq 1,2,3$$

$$\theta_{A}^{k} = \theta_{B}^{k} = \Psi_{A}^{k} = \Psi_{B}^{k} = 0$$

$$C_{31} = S_{120},\ C_{32} = S_{121},\ C_{33} = S_{122},\ C_{34} = S_{123},$$

$$C_{35} = S_{124},\ C_{36} = S_{125},\ C_{37} = S_{126},\ C_{38} = S_{127},$$

$$C_{39} = S_{128},\ C_{43} = S_{129},\ C_{44} = S_{130}$$

CUTIE is stepped by one and control is returned to the user subprogram.

2-277. SUBPROGRAM G27 (WIRES). WIRES initializes the constant attitude wire components from saved measurements of pitch and yaw plane velocity errors. The FORTRAN II reference statement is CALL WIRES.

a. Inputs. The inputs are as follows:

COMMON TAG	SYMBOL	UNITS
XDEW(236)	$\bar{\bar{\dot{\varepsilon}}}_{c}^{k}$	ft/sec
XDEW(240)	$\bar{\bar{\dot{\varepsilon}}}_{z}^{k}$	ft/sec
XDEW(632)	V_G^{k-1}	ft/sec
XC(26)	C_{13}	ft/sec
XS(158)	S_{79}	ft/sec
XDEW(234)	q	cycles
NFLAG(10)	p	integer

b. Outputs. The outputs are as follows:

COMMON TAG	SYMBOL	UNITS
XDEW(510)	$\bar{\bar{\dot{\varepsilon}}}_{cB}^{k}$	ft/sec
XDEW(512)	$\bar{\bar{\dot{\varepsilon}}}_{cS}^{k}$	ft/sec
XC(82)	C_{41}	pure no.
XC(84)	C_{42}	pure no.

c. Program Logic. FD G27. IFLAG is set to identification integer 727 and SW(61) is set ØN. The wire coefficients are computed as follows:

$$C_{41} = \bar{\bar{\dot{\varepsilon}}}_c^{\,k} / V_G^{\,k-1} \quad \text{if } \bar{\bar{\dot{\varepsilon}}}_c^{\,k} - C_{13} < S_{79} \text{ and } q > 0$$

or SW(43) = ØN

$$C_{41} = C_{13} / V_G^{\,k-1} \quad \text{if otherwise}$$

$$C_{42} = \bar{\bar{\dot{\varepsilon}}}_z^{\,k} / V_G^{\,k-1} \quad \text{if } \left| \bar{\bar{\dot{\varepsilon}}}_z^{\,k} \right| < S_{79} \text{ and } q > 0$$

$$C_{42} = 0 \quad \text{if otherwise}$$

$$\bar{\bar{\dot{\varepsilon}}}_{cB}^{\,k} = \bar{\bar{\dot{\varepsilon}}}_c^{\,k} \quad \text{if } P \leq 10$$

$$\bar{\bar{\dot{\varepsilon}}}_{cS}^{\,k} = \bar{\bar{\dot{\varepsilon}}}_c^{\,k} \quad \text{if } P > 10$$

CUTIE is stepped by one and control is returned to the user subprogram.

2-278. SUBPROGRAM G50 (YAWCØ). YAWCØ computes $\delta\dot{\epsilon}_o$ to correct an initial value of launch azimuth. The FORTRAN II reference statement is CALL YAWCØ.

a. Inputs. The inputs are as follows:

COMMON TAG	ITEM	UNITS
XDEW(260)	ΔX_1	ft/sec
XDEW(262)	ΔY_1	ft/sec
XDEW(264)	ΔZ_1	ft/sec
XDEW(266)	$\Delta_2 X_1$	ft/sec-cy
XDEW(268)	$\Delta_2 Y_1$	ft/sec-cy
XDEW(270)	$\Delta_2 Z_1$	ft/sec-cy
XDEW(632)	V_Q^k	ft/sec
XDEW(676)	$\dot{\Psi}_2^k$	quanta/cy
XDEW(696)	[illegible]$_s^k$	cycles
XC(48)	C_{24}	pure no.
XC(50)	C_{25}	pure no.
XC(62)	C_{31}	
.	.	
.	.	
XC(78)	C_{39}	
XS(154)	S_{77}	cycles
NFLAG(2)	M	cycles
NFLAG(10)	P	pure no.
SW(61)	Switch 61	
FRTØD	$180/\pi$	deg/rad

b. Outputs. The outputs are as follows:

COMMON TAG	ITEM	UNITS
XDEW(702)	$\delta\epsilon_{\dot{c}}$	degrees
XDEW(704)	$(\delta\epsilon_{\dot{c}})_1$	degrees
XDEW(706)	$(\delta\epsilon_{\dot{c}})_2$	degrees
XDEW(708)	$(V_R - V)_o = \alpha$	degrees

c. Program Logic.

(1) IFLAG is set to identification integer 750. If SW(61) is ØN, it is set ØFF and this section is continued.

$$\left.\begin{aligned}\hat{X}_1 &= \Delta X_1 + S_{77}\Delta_2 X_1\\ \hat{Y}_1 &= \Delta Y_1 + S_{77}\Delta_2 Y_1\\ \hat{Z}_1 &= \Delta Z_1 + S_{77}\Delta_2 Z_1\\ (DE)^2 &= \hat{X}_1^{\,2} + (C_{24}\hat{Y}_1 + C_{25}\hat{Z}_1)^2\\ DE &= \sqrt{(DE)^2}\end{aligned}\right\} \text{ if } P = 7$$

$$\left.\begin{aligned}DF &= V_G\\ DA &= 0\\ DB &= 0\end{aligned}\right\} \text{ for all values of } P$$

$$DC = \frac{180}{\pi}\,\frac{(1 - C_{35} - C_{36} - C_{37})DF}{(C_{31} + C_{32} + C_{33} + C_{34})DE}$$

(2) If SW(61) is ØFF, and P is 9, 13, 14, 20, or 21,

the procedure is as follows:

$$DA = DA + \left[(\dot{\Psi}_2^{\,k}) / (C_{38} - C_{39}\, t_s^{\,k}) \right]$$

$$DB = DB + V_G$$

$$(\delta\epsilon_{\dot{c}})_1 = \left[(DC)(DA)\right] / \left[DB\right] \qquad \text{if } P = 9, \text{ and } DB \neq 0$$

$$(\delta\epsilon_{\dot{c}})_2 = 0 \qquad \text{if } P = 9 \text{ and } DB = 0$$

$$(\delta\epsilon_{\dot{c}})_2 = \left[(DC)(DA)\right] / \left[DB\right] \qquad \text{if } P = 13 \text{ or } 14 \text{ and } DB \neq 0$$

$$(\delta\epsilon_{\dot{c}})_2 = 0 \qquad \text{if } P = 13 \text{ or } 14 \text{ and } DB = 0$$

$$\alpha = \left[(DC)(DA)\right] / \left[DB\right] \qquad \text{if } P = 20 \text{ or } 21 \text{ and } DB \neq 0$$

$$\alpha = 0 \qquad \text{if } P = 20 \text{ or } 21 \text{ and } DB = 0$$

If SW(61) is ØFF, and P is not 9, 13, 14, 20, or 21, the procedure is as follows:

$$DC = \frac{180}{\pi}\left[\frac{1 - C_{35} - C_{36} - C_{37}}{(C_{31} + C_{32} + C_{33} + C_{34})\, DE}\right] \qquad \text{if } P = 19 \text{ and } M = 1$$

$$\delta\epsilon_{\dot{c}} = (\delta\epsilon_{\dot{c}})_1 + (\delta\epsilon_{\dot{c}})_2 + \alpha \qquad \text{if } P \geq 22$$

(3) After the above computations have been completed, or if P is none of the values specified, [redacted] control is returned to the user subprogram.

2-279. MANUFACTURING SUBPROGRAMS.

2-280. The subprograms described in this area are in two categories. The first category contains the utility subprograms which are operated independently of the TTP.

a.	CØMPTP	U55	Tape Compare
b.	DUPTP	U53	Tape Duplicate
c.	MERGE	U16	Merge COMMON and DIMENSION Statements into Hollerith decks
d.	PRINT	U36	Write Decimal Information on Binary RSD Tape B3
e.	PRTCØN	U50	Control Decimal Dump for Binary RSD Tape

2-281. The second category generates the Common Area record.

a.	CØMLØD	M00	Common Initialization-FORTRAN
b.	CTLNMB	M01	Common Initialization-FAP

2-282. SUBPROGRAM U55 (CØMPTP). CØMPTP compares the input on tape A4 with the input on tape A7. The FORTRAN II reference statement is CALL CØMPTP.

a. Inputs. The inputs are binary records on tape A4 and A7. The last word on each record is a checksum. The first word contains the file number, the record number, and the first address of the information plus the number of words. The second word contains the number of words and the starting address of the subprogram. The end-of-file is indicated by three tape marks.

b. Outputs. The output is the U08 output mode indicator SW(120) set ØN. If a tape redundancy occurs, SW(123) is set ØN. For all errors SW(70) is set ØN.

a. ERROR READING REC ______ FILE ____

b. REC ______ FILE ____ DOES NOT COMPARE ON TAPE 7

c. TAPE 4 AND 7 COMPARISON COMPLETE

d. EOF MISSING ON TAPE 7

c. Program Logic. FD U55

(1) Steps 1-3. Tapes 4 and 7 are rewound. The contents of index registers 1, 2, and 4 are saved. Work register ERRØR and SW(123) are set ØFF and SW(120) is set ØN.

(2) Steps 4-13. The first and second words are read into core from tape A4. If the end-of-file has been reached, control is transferred to step 32. If this is an

end-of-tape record (all zeros) on a hash record (all sevens), control is transferred to step 17. The remainder of the record is read from tape A4. If a tape redundancy error occurs and three attempts have been made to read the tape, SW(123) is set ØN, an eight is stored in ITYER and control is transferred to step 14. Otherwise the tape is backspaced one record and control is transferred to step 4. If no redundancy occurs, the logical checksum is computed and compared with the tape checksum. If the checksums agree and three attempts have been made to read the record, ITYER is set to one and the subprogram continues at step 14. Otherwise the record is backspaced and control is transferred to step 4. If the checksums are not in agreement control is transferred to step 17.

(3) Steps 14-15. Work register ERRØR is set ØN and U08 prints statement a. The subprogram continues at step 16.

(4) Step 16. The first and second words are read into core from tape A7.

(5) Steps 17-21. If the first two words from tape A7 do not compare with tape A4, control is transferred to step 30. If this is an end-of-tape record control is transferred to step 27. If this is a hash record control is transferred to step 4. If not, the remainder of the record is read from tape A7. If the record from tape A7 compares with tape A4, control is transferred to step 4. Otherwise the subprogram continues at step 22.

(6) Steps 22-26. If three attempts have not been made to read tape A7, tape A7 is backspaced one record and control is transferred to step 16. Otherwise ITYER is set to eight, ERRØR is set ØN and U08 prints statement b. If the record on tape A4 is an end-of-tape record the subprogram continues at step 27. Otherwise control is transferred to step 4.

(7) Steps 27-29. U08 prints statement c and SW(70) is set to ERRØR. The contents of the index registers are restored and the subprogram returns to the user subprogram.

(8) Steps 30-31. If the record from tape A4 is a hash record, control is transferred to step 13. If the record from tape A7 is a hash record, control is transferred to step 16. Otherwise control is transferred to step 22.

(9) Steps 32-34. The first and second words are read from tape A7. If an end-of-file is reached, and three tape marks are read, control is transferred to step 27. Otherwise control is transferred to step 4. If no end-of-file is reached, an eight is stored in ITYER, ERRØR is set ØN, and U08 prints statement d. Control is transferred to step 34.

2-283. SUBPROGRAM U53 (DUPTP). DUPTP duplicates the contents of a magnetic tape by writing a new magnetic tape. The FORTRAN II reference statement is CALL DUPTP. The FAP reference instruction is TSX DUPTP, 4.

a. Inputs. The input is magnetic tape containing binary records and mounted on tape A4. The first word on the tape contains the file number, record number, and the first address of information plus the number of words. The second word contains the transfer instruction, the number of words, and the starting address of the program. The last word is a checksum. The end-of-file is indicated by three tape marks.

b. Outputs. The output is a duplicate of the input tape on tape A7. An on-line printout indicates either that the tape has been duplicated successfully or, if an error occurs, that the error was made during a read or a write routine. The following printed statements are also outputs:

a. ERROR READING REC_____ FILE____ .

b. TAPE 4 DUPLICATED ON TAPE 7.

c. ERROR WRITING REC_____ FILE____ .

c. Program Logic. FD U53.

(1) Steps 1-3. Tapes A4 and A7 are rewound. The contents of index registers 1,2, and 4 are saved. SW(70) and SW(123) are set ØFF and SW(120) is set ØN.

(2) Steps 4-8. Each word is read into core from tape A4 and a checksum is accumulated until the end of the record is reached. The last word on the record is a checksum which is stored in a work area for later comparison. If the end-of-file is reached, control is transferred to step 32. If there is no tape redundancy, control is transferred to step 20.

(3) Steps 9-11. Index register 2, which is used as a counter, is interrogated to determine if three attempts have been made to read the tape. Until three attempts have been made, the tape is backspaced and control is transferred to step 4. If the redundancy tape error still occurs after the third attempt, an eight is stored in ITYER.

(4) Steps 12-14. U08 prints statement a. SW(70) is set ØN.

(5) Steps 15-16. The record that was read in from tape A4 is written on tape A7. If the record was not written correctly, control is transferred to step 25.

(6) Step 17. If the end-of-file has not been reached, control is transferred to step 4.

(7) Steps 18-19. U08 prints statement b. The contents of the index registers are restored and the subprogram exits to the user subprogram.

(8) Steps 20-24. If the second word read in was a hash record (all sevens) or an end-of-tape record (all zeros),

control is transferred to step 15. The subprogram accumulated checksum is compared with the checksum read from the tape. If the checksums agree, control is transferred to step 15. Index register 2, which is used as a counter, is interrogated to determine if three attempts have been made to arrive at an equal checksum. Until three attempts have been made, the tape is backspaced and control is transferred to step 4. If the checksums do not agree after the third attempt, a one is stored in ITYER and control is transferred to step 12.

(9) Steps 25-31. If the second word was a hash record, control is transferred to step 4. Index register 2, which is used as a counter, is interrogated to determine if three attempts have been made to write the tape. Until three attempts have been made, the tape is backspaced and control is transferred to step 15. If the record still has not been written correctly after the third attempt, a one is stored in ITYER. UO8 prints error statement c. SW(70) is set ØN and control is transferred to step 17.

(10) Steps 32-33. A tape mark is written on tape A7. If three tape marks have been written, control is transferred to step 18; otherwise control is transferred to step 4.

2-284. SUBPROGRAM U16 (MERGE). MERGE inserts the FORTRAN II COMMON and DIMENSION statement deck into the FORTRAN Hollerith decks before compilation. This subprogram is compiled as a main program.

a. Inputs. The input is a tape on unit B10 which contains Hollerith decks to be compiled.

b. Outputs. The output is a tape on unit A7 which contains Hollerith decks ready for compilation. The following statements are also outputs.

a. TAPE B10 REDUNDANT

b. MERGE COMPLETED

c. MERGE OUTPUT INCOMPLETE DO OVER

c. Program Logic. FD U16

(1) Steps 1-3. Common tags ØN, ØFF, and ITYER are initialized and the redundancy and end-of-tape indicators are turned ØFF. Index register 1 is initialized for reading a maximum of 6400 BCD words from COMMON and DIMENSION statement cards and card counter (index register 2) is initialized to zero.

(2) Steps 4-8. U20 reads a COMMON or DIMENSION statement card and INTRØG interrogates SW(70) to determine if there is a reading error (SW(70) = ØN). If ØFF, no error occurred and the subprogram continues at step 9. If ØN, an error occurred and U08 prints statement a. The sub-

program exits to ERRPRT to print an indication of the error.

(3) Steps 9-11. If the card just read was an END card, the subprogram continues at step 12. Otherwise, the card just read is stored in the work area TEMP which is a storage block used for storing each COMMON or DIMENSION statement card in consecutive order. The card counter is stepped by one and control is transferred to step 4.

(4) Step 12. Tape A7 is rewound and all SENSE lights are turned OFF.

(5) Steps 13-19. U20 reads a card from tape B10. These are the cards comprising the Hollerith decks. After each card is read, INTRØG interrogates SW(70) to determine if there is a read error (SW(70) = ØN). If ØFF, no error occurred and the subprogram continues at step 18. If ØN, an error occurred and a check is made for end-of-file. If the error was due to an end-of-file, control is transferred to step 37. If not an end-of-file, SENSE light 1 is turned ON and the card is examined for a FOR card. If it is a FOR card, SENSE light 2 is turned ON and control is transferred to step 26. If not a FOR card, the subprogram continues at the next step.

(6) Steps 20-24. If the card just read was a COMMON control card, control is transferred to step 28. If not a COMMON card, the contents of card columns 1 and 2 are saved. If a C is in column 1 and column 2 is not blank, column 1 is made blank and the first 12 characters on the card are

stored in a work area. Otherwise the subprogram continues at step 26.

(7) Step 25. If SENSE light 1 is ON control is transferred to step 44. Otherwise the subprogram continues at the next step.

(8) Steps 26-27. U08 writes the card contents off line. A test is made for an end-of-tape on A7. If the end-of-tape is sensed control is transferred to step 40. Otherwise, control is transferred to step 13.

(9) Step 28. The counter (index register 2) is set for copying the COMMON and DIMENSION statements onto A7 in the FORTRAN Hollerith deck image. SENSE light 4 is turned ON.

(10) Steps 29-32. The COMMON and DIMENSION statements from TEMP are copied into the output work register. U08 writes twelve BCD words from the output area onto tape A7. INTRØG interrogates SW(70) to determine if there is an output error (SW(70) = ØN). If ØN, an error occurred and the subprogram continues at step 33. If ØFF, no error occurred, and control is transferred to step 34.

(11) Step 33. The subprogram exits to ERRPRT to print an indication of the error.

(12) Step 34. If the end-of-tape is sensed, control is transferred to step 43. Otherwise the subprogram continues at step 35.

(13) Steps 35-36. If there are more COMMON and DIMENSION statements to be copied, control is transferred to step 29. If there are no more to be copied, all SENSE lights are turned OFF and control is transferred to step 13.

(14) Steps 37-39. An end-of-file is written on tape A7 and it is rewound. U08 prints statement b and ENDJØB halts the subprogram operation.

(15) Steps 40-41. Step 42 is modified to continue at step 4 to continue reading cards. If the program is to continue, U08 prints statement c.

(16) Step 42. This step is modified to continue at step 4 or step 35 if program operation is to continue.

(17) Step 43. Step 42 is modified to continue at step 35 to continue writing output on a new A7. If program operation is to continue, control is transferred to step 41.

(18) Step 44. U08 prints statement a and the subprogram exits to print an indication of the error.

2-285. SUBPROGRAM U36 (PRINT). PRINT writes in decimal form and with identifying names the binary record from tape B3 as one record on tape A7. The FORTRAN II reference statement is CALL PRINT(B,J,NAMIN,TEST).

a. Inputs. The inputs are the arguments B, J, NAMIN, and TEST. B is the starting location of the block to be written, J is the total number of words in the block, NAMIN is the starting location of the identifying names, and TEST is used to determine whether control cards have been read. The control card (figure 2-1) and the additional output cards for the SIM mode (figure 2-6) are also inputs on the first entrance to this subprogram. The card format is given in figure 2-1.

b. Output. The output is the BCD record on tape A7. The following written statement is also an output:

RSDORE TAPE

c. Program Logic. FD U36. Internal register TEST is tested with 10 to determine whether this subprogram was entered previously. If unequal, the control card and additional output cards for the SIM mode are read. If tape B3 is to be dumped without identifying words (one in column 30 of control card), the output statement and the record block are written on tape A7 and the subprogram returns to the user subprogram. Otherwise the record block is set up with additional RSD outputs and identifying names. The output statement is written as one record on tape A7, and the identifying names and the record block are written as another record. The subprogram returns to the user subprogram.