RAND H 10.2	RAND0 STD2	"Read and AND H" AND's c(A) and c(H) and stores logical product in A.
READ H 10.0	READ0 STD2	"Read H" Enters c(H) into A.
RELINT 00.0003	STD2	"Release Inhibit Interrupt" Resets inhibit interrupt switch to allow program interruption in favor of a programmed operation of higher priority.
RESUME 05.0017	NDX0 RSM3	"Resume Interrupted Program" Takes next instruction from location 0017 and enters content of location 0015 into Z. Thus, execution of the interrupted program section is resumed.
ROR H 10.4	ROR0 STD2	"Read and OR H" OR's c(A) and c(H), and stores logical sum in A.
RUPT 10.7	RUPT0 RUPT1 STD2	"Interrupt Program Execution" Takes next instruction from address supplied by Interrupt Priority Control. Stores c(B) in location 0017 and c(Z) in location 0015.
RXOR H 10.6	RXOR0 STD2	"Read and Exclusive OR H" Forms exclusive OR from c(A) and c(H), and stores result in A.
SHANC C	SHANC	"Shift and Add Increment C" Shifts c(C) one place to the left and enters a ONE into bit position 0 of C.
SHINC C	SHINC	"Shift Increment C" Shifts c(C) one place to the left and enters a ZERO into bit position 0 of C.
SR .0021		"Shift Right" Shifts quantity, which is entered into location 0021, one place to the right.

STORE E	STORE0 STORE1	"Store E"; data supplied by GSE is entered into E by GSE. Address E is also supplied by GSE.
SU E 16.0	SU0 STD2	"Subtract E" Subtracts c(E) from c(A) and stores the difference in A.
TC K 00.	TC0	"Transfer Control to K" Takes next instruction from K and stores return address (I+1) in Q.
TCF F 01.2 .4 .6	TCF0	"Transfer Control to Fixed F" Takes next instruction from F without changing c(Q).
TCSAJ K 00.	TCSAJ3 STD2	"Transfer control to specified address K" Takes next instruction from address which is supplied by GSE.
TS E 05.4	TS0 STD2	"Transfer to Storage E" If A does not contain an overflow quantity, instruction enters c(A) into E and takes next instruction from I+1. If A contains a positive overflow, instruction enters c(A) without overflow bit into E, enters plus one into A, and takes next instruction from I+2. If A contains a negative overflow, instruction enters c(A) without overflow bit into E, enters minus one into A, and takes next instruction from I+2.
WAND H 10.3	WAND0 STD2	"Write and AND H" AND's c(A) and c(H), and stores logical product in A and H.
WOR H 10.5	WOR0 STD2	"Write and OR H" OR's c(A) and c(H), and stores logical sum in A and H.
WRITE H 10.1	WRITE0 STD2	"Write H" Enters c(A) into H.
XCH E 05.6	XCH0 STD2	"Exchange A and E" Exchanges c(A) with c(E).

RAYTHEON

G. L. Silver

SPACE AND INFORMATION SYSTEMS DIVISION

SYSTEMS SUPPORT TRAINING

ADS E 02.6	ADS0 STD2	"Add to Storage E" Adds c(A) and c(E), stores sum with overflow bit in A and sum without overflow bit in E.
AD K 06.	AD0 STD2	"Add K" Adds c(K) to c(A) and stores sum in A.
AUG E 12.4	AUG0 STD2	"Augment E" Increases the magnitude of the quantity contained in E by one and stores the augmented quantity in E.
BZF F 11.2 11.4 11.6	BZF0 STD2	"Branch on Zero to Fixed F" Branches according to c(A). c(A) — Transfers to plus or minus zero — F (subinstruction STD2 is not executed) non zero — I+1 (subinstruction STD2 is executed)
BZMF F 16.2 .4 .6	BZMF0 STD2	"Branch on Zero or Minus to Fixed F" Branches according to c(A). c(A) — Transfers to zero or negative nonzero — F (subinstruction STD2 is not executed) positive nonzero — I+1 (subinstruction STD2 is executed)

CCS E 01.0	CCS0 STD2	"Count, Compare, and Skip on E" Branches according to c(E) and stores in A the |c(E)| diminished by one. c(E) | Transfers to positive nonzero | I+1 plus zero | I+2 negative nonzero | I+3 minus zero | I+4
CA K 03.	CA0 STD2	"Clear and Add K" Enters c(K) into A
CYL .0022		"Cycle Left" Cycles quantity, which is entered into location 0022, one place to the left.
CS K 04.	CS0 STD2	"Clear and Subtract K" Enters the complemented c(K) into A.
CYR .0020		"Cycle Right" Cycles quantity, which is entered into location 0020, one place to the right.
DAS E 02.0	DAS0 DAS1 STD2	"Double Add to Storage E" Adds c(A, L) and c(E, E+1) and stores sum without overflow bit in E and E+1. Enters plus one into A in case of positive overflow, minus one in case of negative overflow, and plus zero in case of no overflow. Enters plus zero into L.
DCA K 13.	DCA0 DCA1 STD2	"Double Clear and Add K" Enters c(K, K+1) into A and L.
DCS K 14.	DCS0 DCS1 STD2	"Double Clear and Subtract K" Enters the complemented c(K, K+1) into A and L.
DIM E 12.6	DIM0 STD2	"Diminish E" Decreases the magnitude of the quantity contained in E by one and stores diminished quantity in E.

DINC C	DINC	"Diminish Increment C" Decreases the magnitude of the quantity contained in C by one and stores diminished quantity in C.
DV E 11.0	DV0 DV1 DV3 DV7 DV6 DV4 STD2	"Divide by E" Divides double precision quantity c(A, L) by c(E), stores quotient in A and remainder in L. Signs of b(A) and b(L) need not agree. Sign of remainder equals sign of dividend.
DXCH E 05.2	DXCH0 DXCH1 STD2	"Double Exchange A and E" Exchanges c(A, L) with c(E, E+1).
EDOP .0023		"Edit Operator" Shifts quantity, which is entered into location 0023, seven places to the right.
EXTEND 00.0006	STD2	"Extend" Enters a ONE into bit position SQ-EXT. The next instruction, taken from I+1, is an Extra-Code Instruction
FETCH K	FETCH0 FETCH1	"Fetch K" Displays c(K) on GSE. Address K is supplied by GSE.
GO 00.	GOJ1 TC0	"Go" Takes next instruction from location 04000 in E Memory.
INCR E 02.4	INCR0 STD2	"Increment E" Adds plus one to c(E) and stores incremented quantity in E.
INHINT 00.0004	STD2	"Inhibit Interrupt" Sets inhibit interrupt switch in Interrupt Priority Control to prevent interruption of program execution.
INOTLD H	INOTLD	"In Out Load H" Data supplied by GSE is entered into H by GSE. Channel address H is supplied by GSE.
INOTRD H	INOTRD	"In Out Read H" Displays c(H) on GSE. Channel address H is supplied by GSE.
LXCH E 02.2	LXCH0 STD2	"Exchange L and E" Exchanges c(L) with c(E).

MCDU C	MCDU	"Minus CDU C" Subtracts one from cyclic TWO's complement number in C and stores decremented quantity in C.
MINC C	MINC	"Minus Increment C" Subtracts one from c(C) and stores decremented quantity in C.
MP K 17.	MP0 MP1 MP3	"Multiply K" Multiplies c(K) by c(A) and stores double precision product in A and L (signs in A and L agree).
MSK K 07.	MSK0 STD2	"Mask with K" AND's c(A) with c(K) and stores logical product in A.
MSU E 12.0	MSU0 STD2	"Modular Subtract E" Subtracts cyclic TWO's complement number in E from cyclic TWO's complement number in A and stores difference expressed in ONE's complement number in A.
NDX E 05.0	NDX0 NDX1	"Index Next Basic Instruction with E" Adds c(E) to c(I+1) and takes sum as next instruction.
NDX K 15.	NDXX0 NDXX1	"Index Next Extra-Code Instruction with K" Adds c(K) to c(I+2) and takes sum as next instruction. Retains the ONE in bit position SQ-EXT.
PCDU C	PCDU	"Plus CDU C" Adds one to cyclic TWO's complement number in C and stores incremented quantity in C.
PINC C	PINC	"Plus Increment C" Adds one to c(C) and stores incremented quantity in C.
QXCH E 12.2	QXCH0 STD2	"Exchange Q and E" Exchanges c(Q) with c(E). Takes next instruction from I+1.