INSTRUCTION	ORDER CODE	DESCRIPTION
AD K	06.	Add K Adds c(K) to c(A) and stores sum in A.
ADS E	02.6	Add to Storage E. Adds c(A) and c(E), stores sum with overflow bit in A and sum without overflow bit in E.
AUG E	12.4	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	Branch on Zero to Fixed F. Branches according to C(A). c(A) → Transfers to ±0 → F ≠0 → I + 1
BZMF F	16.2 16.4 16.6	Branch on Zero or Minus to Fixed F. Branches according to c(A). Q CODE PART OF E ADDRESS c(A) → Transfers to ±0, NNZ → F PNZ → I + 1
CA K	03.	Clear and Add K. Enters c(K) into A.
CAE E	03.	Alternate spelling of CA K when referring to E memory.
CAF F	03.	Alternate speeling of CA K when referring to F memory.
CCS E	01.0	Count, Compare, and Skip on E. Branches according to c(E) and stores in A the |c(E)| diminished by one. Leaves A=+0 if c(E) = ±0. c(E) → Transfers to PNZ → I + 1 +0 → I + 2 NNZ → I + 3 -0 → I + 4
COM	04.0000	Complement A.
CS K	04.	Clear and Subtract K. Enters the complemented c(K) into A.
CYL	.0022	Cycle Left. Cycles quantity, which is entered into location 0022, one place to the left.
CYR	.0020	Cycle Right. Cycles quantity, which is entered into location 0020, one place to the right.
DAS E	02.0	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.	Double Clear and Add K. Enters c(K,K+1) into A and L.
DCS K	14.	Double Clear and Subtract K. Enters the complemented c(K, K+1) into A and L.
DCOM	14.0000	Double precision complement; DCS A.
DDOUBL	02.0000	Double precision double; DAS A.
DIM E	12.6	Diminish E. Decreases the magnitude of the quantity contained in E by one and stores diminished quantity in E.
DOUBLE	06.0000	Double A; AD A.
DTCB	05.2005	Double precision transfer control both banks; DXCH Z. Set c(A, L) = b(Z, BB); and c(Z, BB) = b(A, L).
DV E	11.0	Divide by E. Divide 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	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	0.0006	Extend. Enters a ONE into bit position SQ-EXT. The next instruction, taken from I+1, is an Extra-Code Instruction.
INCR E	02.4	Increment E. Adds plus one to c(E) and stores incremented quantity in E.
INDEX E	05.0	Index Next Basic Instruction with E. Adds c(E) to c(I+1) and takes sum as next instruction.
INDEX K	15.	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.

INSTRUCTION	ORDER CODE	DESCRIPTION
INHINT	00.0004	Inhibit Interrupt. Sets inhibit interrupt switch in Interrupt Priority Control to prevent interruption of program execution.
LXCH E	02.2	Exchange L and E. Exchanges c(L) with c(E).
MASK K	07.	Mask with K. AND's c(A) with c(K) and stores logical product in A.
MP K	17.	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.	Alternate spelling of MASK K.
MSU E	12.0	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	Alternate spelling of INDEX E.
NDX K	15.	Alternate spelling of INDEX K.
NOOP	03.0000	No operation (Erasable); instruction is stored in E memory; CA A.
NOOP	TCF (I+1)	No operation (Fixed); where I is address of instruction TCF (I+1) stored in F memory.
OVSK	05.4000	Overflow skip; TS A.
QXCH E	12.2	Exchange Q and E. Exchanges c(Q) with c(E).
RAND H	10.2	Read and AND H. AND's c(A) and c(H) and stores logical product in A.
READ H	10.0	Read H. Enters c(H) into A.
RELINT	00.0003	Release Inhibit Interrupt. Resets inhibit interrupt switch to allow program interruption in favor of a programmed operation of higher priority.
RESUME	05.0017	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.
RETURN	00.0002	Return; TC Q.
ROR H	10.4	Read and OR H. OR's c(A) and c(H), and stores logical sum in A.
RXOR H	10.6	Read and Exclusive OR H. Forms exclusive OR from c(A) and c(H), and stores result in A.
SR	.0021	Shift Right. Shifts quantity, which is entered into location 0021, one place to the right.
SU E	16.0	Subtract E. Subtracts c(E) from c(A) and stores the difference in A.
TCAA	05.4005	Transfer control to address in A; TS Z.
TC K	00.	Transfer Control to K. Takes next instruction from K and stores return address (I+1) in Q.
TCF F	01.2 01.4 01.6	Transfer Control to Fixed F. Takes next instruction from F without changing c(Q).
TCR K	00.	Alternate spelling of TCK (Transfer control setting up return.)
TS E	05.4	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	Write and AND H. AND's c(A) and c(H), and stores logical product in A and H.
WOR H	10.5	Write and OR H. OR's c(A) and c(H), and stores logical sum in A and H.
WRITE H	10.1	Write H. Enters c(A) into H.
XCH E	05.6	Exchange A and E. Exchanges c(A) with c(E).
ZL	02.2007	Zero L; LXCH ZERO
ZQ	12.2007	Zero Q; QXCH ZERO