

# Control Data 1604A Computer Instructions

## SYMBOLS

<p>k = Address portion of instruction          K = k + (B<sup>b</sup>), Modified shift count          m = Address portion of instruction          M = m + (B<sup>b</sup>), Modified operand address          y = Address portion of instruction          Y = y + (B<sup>b</sup>), Modified operand</p>	<p>b = Designator for index          j = Designator for 22, 23, 74-76          # = Restrict instruction to upper          ' = Complemented          ( ) = Contents of          NI = Next instruction</p>
--	--

### DATA TRANSMISSION

LDA	12	Load A	(M)→A
LAC	13	Load A complement	(M)'→A
STA	20	Store A	(A)→M
LDQ	16	Load Q	(M)→Q
LQC	17	Load Q complement	(M)'→Q
STQ	21	Store Q	(Q)→M

### SHIFTING

ARS	01	(A) right shift, by K
QRS	02	(Q) right shift, by K
LRS	03	(AQ) right shift, by K
ALS	05	(A) left shift, by K
QLS	06	(Q) left shift, by K
LLS	07	(AQ) left shift, by K

### INDEXING

ENI	50	Enter index	y→B <sup>b</sup>
INI	51	Increase index	y + (B <sup>b</sup> )→B <sup>b</sup>
LIU	52	Load index upper	(m <sub>UA</sub> )→B <sup>b</sup>
LIL	53	Load index lower	(m <sub>LA</sub> )→B <sup>b</sup>
SIU	56	Store index upper	(B <sup>b</sup> )→m <sub>UA</sub>
SIL	57	Store index lower	(B <sup>b</sup> )→m <sub>LA</sub>
ISK#	54	Index skip:	
			(B <sup>b</sup> ) ≠ y: (B <sup>b</sup> ) + 1 → B <sup>b</sup> , half exit
			(B <sup>b</sup> ) = y: 0 → B <sup>b</sup> , full exit
IJP	55	Index jump:	
			(B <sup>b</sup> ) ≠ 0: (B <sup>b</sup> ) - 1 → B <sup>b</sup> , NI = m
			(B <sup>b</sup> ) = 0: NI

## ARITHMETIC

### FIXED/FRACTIONAL

ADD	14	Add to A	[(A) + (M)]→A
SUB	15	Subtract from A	[(A) - (M)]→A
MUI	24	Multiply integer	(M)(A)→QA
DVI	25	Divide integer	(QA)/(M)→A, rem Q
MUF	26	Multiply fractional	(M)(A)→AQ
DVF	27	Divide fract.	(AQ)/(M)→A, rem Q
RAD	70	Replace add	[(M) + (A)]→M & A
RSB	71	Replace subtract	[(M) - (A)]→M & A
RAO	72	Replace add one	[(M) + 1]→M & A
RSO	73	Replace sub. one	[(M) - 1]→M & A

### FLOATING

FAD	30	Floating add.	[(A) + (M)]→A
FSB	31	Floating sub.	[(A) - (M)]→A
FMU	32	Floating mult.	(A)(M)→A
FDV	33	Floating div.	(A)/(M)→A

SCA	34	Scale A	A left until  (A)  ≥ .5 or k=0; (k=no. of shifts) →B <sup>b</sup>
SCQ	35	Scale AQ	AQ left until  (AQ)  ≥ .5 or k=0; (k=no. of shifts) →B <sup>b</sup>

**A AND Q TEST****CONSOLE TEST**

AJP\* 22 Jump to m on condition j  
 QJP\* 23 Jump to m on condition j

SLJ\* 75 Jump to m on condition j  
 SLS\* 76 Stop on j, and jump to m

**MEMORY TEST**

EQS# 64 Search (B<sub>b</sub>) words, if (M-1), or (M-2), etc. = (A) exit; ≠A, half exit  
 THS# 65 Search (B<sub>b</sub>) words, if (M-1), or (M-2), etc. > (A) exit; ≤A, half exit  
 MEQ# 66 Search (B<sub>b</sub>) words, if L(Q)(M-1), or (M-2), etc. = (A) exit; ≠A, half exit  
 MTH# 67 Search (B<sub>b</sub>) words, if L(Q)(M-1), or (M-2), etc. > (A) exit; ≤A, half exit  
 SSK# 36 Storage skip (M) neg: exit; (M) pos: half exit  
 SSH# 37 Storage shift (M) neg: exit, left 1; (M) pos: half exit, left 1

**LOGICAL**

SST 40 Selective Set  
 Set (A<sub>n</sub>) = 1 for (M<sub>n</sub>) = 1  
 SCL 41 Selective clear  
 Clear (A<sub>n</sub>) to 0 for (M<sub>n</sub>) = 1  
 SCM 42 Selective complement  
 Complement (A<sub>n</sub>) for (M<sub>n</sub>) = 1  
 SSU 43 Selective substitute  
 (M<sub>n</sub>) → (A<sub>n</sub>) for (Q<sub>n</sub>) = 1  
 LDL 44 Load logical L(Q)(M) → A  
 ADL 45 Add logical [(A) + L(Q)(M)] → A  
 SBL 46 Sub. logical [(A) - L(Q)(M)] → A  
 STL 47 Store logical L(Q)(A) → M

**GENERAL**

ENA 10 Enter A Extend sign Y, Y → A  
 ENQ 04 Enter Q Extend sign Y, Y → Q  
 INA 11 Increase A Extend sign Y,  
 [Y + (A)] → A  
 SAU 60 Substitute address upper  
 (A<sub>14-00</sub>) → M<sub>VA</sub>  
 SAL 61 Substitute address lower  
 (A<sub>14-00</sub>) → M<sub>LA</sub>  
 INT 62 Input Block trans (B<sub>b</sub>) words  
 OUT 63 Output Block trans (B<sub>b</sub>) words  
 EXP 74 External function  
 j = 0; Select  
 j = 1-6; Activate channel j  
 j = 7; Sense

**DESIGNATOR FOR \* INSTRUCTION**

22	23	75	76
0 (A) = 0: Jump	(Q) = 0: Jump	Jump	Stop: Jump
1 (A) ≠ 0: Jump	(Q) ≠ 0: Jump	Key 1: Jump	Key 1: Stop: Jump
2 (A) Pos: Jump	(Q) Pos: Jump	Key 2: Jump	Key 2: Stop: Jump
3 (A) Neg: Jump	(Q) Neg: Jump	Key 3: Jump	Key 3: Stop: Jump
4 (A) = 0: Ret. Jump	(Q) = 0: Ret. Jump	Ret. Jump	Stop: Ret. Jump
5 (A) ≠ 0: Ret. Jump	(Q) ≠ 0: Ret. Jump	Key 1: Ret. Jump	Key 1: Stop: Ret. Jump
6 (A) Pos: Ret. Jump	(Q) Pos: Ret. Jump	Key 2: Ret. Jump	Key 2: Stop: Ret. Jump
7 (A) Neg: Ret. Jump	(Q) Neg: Ret. Jump	Key 3: Ret. Jump	Key 3: Stop: Ret. Jump

1, 2, & 3 refer to Selective Jump or Stop Key Switches.

**CONTROL DATA****CORPORATION****COMPUTER DIVISION****MINNEAPOLIS, MINNESOTA**