

1206 COMPUTER

Repertoire of Instructions

JP & RJP j-DESIGNATORS

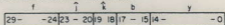
j	JP	RJP	JP	RJP
0	60	64	61	65
0	[No Jump]*		(Uncond. Jump)	
1	[Uncond. Jump]*		KEY 1	
2	Q POS		KEY 2	
3	Q NEG		KEY 3	
4	A ZERO		STOP	
5	A NOT Zero		STOP 5	
6	A POS		STOP 6	
7	A NEG		STOP 7	
↑	62 ↑		63 ↑	
0-15 _h	C ⁿ ACTIVE IN		C ⁿ ACTIVE OUT	

*60 Clears interrupt B bootstrap modes.

↑-DESIGNATORS

(4 bits)

↑ Occupies 4 bit positions and represents Cⁿ where n may be 0—15_h.
The instruction word assumes the format:



↑-DESIGNATORS

(2 bits)

↑	EX- FCT	STR+C ⁿ	JP	IN+C ⁿ , OUT+C ⁿ
	13	17	62 63	73 75 74 76
0	'not used'	'not used'	'blank'	'blank'
1	'not used'	'not used'	L	L
2	'not used'	'not used'	U	'not used'
3	W	W	W	W

*j-DESIGNATORS

j	COM * A, * Q, * AQ	DIV	ADD * Q, SUB * Q	ENT * LP, RPL * LP	RPT
	04	23	26 27	40 44	70
0	(no skip)	(no skip)	(no skip)	(no skip)	(no mod.) Y of NE = Y
1	(unconditional skip)	SKIP	SKIP	SKIP	ADV Y of NE = Y + 1
2	Y LESS Y ≤ (Q)	NO Over Flow	A POS	EVEN parity	BACK Y of NE = Y - 1
3	Y MORE Y > (Q)	Over Flow	A NEG	ODD parity	ADD B Y of NE = Y + B ⁶
4	Y IN (Q) ≥ Y and Y > (A)	A ZERO	Q ZERO	A ZERO	Rpl. inc. Y of NE = Y + B ⁶ ✓
5	Y OUT (Q) < Y or Y ≤ (A)	A NOT Zero	Q NOT Zero	A NOT Zero	ADV R Y of NE = Y + 1 + B ⁶ ✓
6	Y LESS Y ≤ (A)	A POS	Q POS	A POS	BACK R Y of NE = Y - 1 + B ⁶ ✓
7	Y MORE Y > (A)	A NEG	Q NEG	A NEG	ADD BR Y of NE = Y + B ⁶ + B ⁶ ✓

✓ B⁶ Increment if NI is RPL class, increments Y address for the store portion of the replace.

NE - Next execution

NORMAL j-DESIG.

j	(Not applicable on * or ~) Skip Code
0	(no skip)
1	SKIP
2	Q POS
3	Q NEG
4	A ZERO
5	A NOT Zero
6	A POS
7	A NEG

NORMAL k-DESIGNATORS

k	READ		STORE		REPLACE	
	Code	Origin	Code	Dest.	Code	Origin Dest.
0	'blank'	U _L	Q	Q	not used	— —
1	L	M _L	L	M _L	L	M _L M _L
2	U	M _U	U	M _U	U	M _U M _U
3	W	M	W	M	W	M M
4	X	XU _L	A	A	not used	— —
5	LX	XM _L	CPL	Cpl M _L	LX	XM _L M _L
6	UX	XM _U	CPU	Cpl M _U	UX	XM _U M _U
7	A	A	CPW	Cpl M	not used	— —

LEGEND

- M - Memory word (30 bits)
- M_L - Lower half memory word
- M_U - Upper half memory word
- X - Sign bit extended
- Cpl - Complement
- A - A-register
- Q - Q-register
- U - U-register

1206 COMPUTER

Repertoire of Instructions

01	Right Shift * Q	Shift (Q) Right by Y	
02	Right Shift * A	Shift (A) Right by Y	
03	Right Shift * AQ	Shift (AQ) Right by Y	
04	Compare * A, * Q, * AQ	Sense (j), (A), (A) _j	
05	Left Shift * Q	Shift (Q) Left by Y	
06	Left Shift * A	Shift (A) Left by Y	
07	Left Shift * AQ	Shift (AQ) Left by Y	
10	ENTER * Q	Y → Q	
11	ENTER * A	Y → A	
12	ENTER * B ⁿ	Y → B ⁿ	
13	EXternal-Function * C ⁿ	$\bar{1} \neq 0$ or $1, (Y) \rightarrow Cn = 0$ or 1	
14	SToRe * AQ	(Q) → Y, x=0, Q' → Q	
15	SToRe * A	(A) → Y, k=4, A' → A	
16	SToRe * B ⁿ	(B) _j → Y	
17	SToRe * C ⁿ	(C) _j → Y	
20	ADD * A	(A) + Y → A	
21	SUBtract * A	(A) - Y → A	
22	MULTiply	(Q) Y → AQ	
23	DIVide	(AQ) / Y → Q, R → A _f	
24	RePLace * A + Y	(A) + (Y) → YB A	
25	RePLace * A - Y	(A) - (Y) → YB A	
26	ADD * Q	(Q) + Y → Q, (A) _j = (A) _j } j interpretation	
27	SUBtract * Q	(Q) - Y → Q, (A) _j = (A) _j } reversed for ABQ	
30	ENTER * Y + Q	Y + (Q) → A	
31	ENTER * Y - Q	Y - (Q) → A	
32	SToRe * A + Q	(A) + (Q) → YB A	
33	SToRe * A - Q	(A) - (Q) → YB A	
34	RePLace * Y + Q	(Y) + (Q) → YB A	
35	RePLace * Y - Q	(Y) - (Q) → YB A	
36	RePLace * Y + 1	(Y) + 1 → YB A	
37	RePLace * Y - 1	(Y) - 1 → YB A	
40	ENTER * LP**	L(Y)(Q) → A _j , j=2, even parity, j=3, odd parity	
41	ADD * LP	L(Y)(Q) + (A) → A	
42	SUBtract * LP	(A) - L(Y)(Q) → A	
43	COMpare * MASK	(A) - L(Y)(Q) SENSE (j), (A) + L(Y)(Q), (A) _j = (A) _j	
44	RePLace * LP	L(Y)(Q) → YB A, j=2, even parity, j=3, odd parity	
45	RePLace * A + LP	L(Y)(Q) + (A) → YB A	
46	RePLace * A - LP	(A) - L(Y)(Q) → YB A	
47	SToRe * LP	L(A)(Q) → Y, (A) _j = (A) _j	
50	SElective * SET	SET (A) _n FOR (Y) _n =1	
51	SElective * CP**	COMPLEMENT (A) _n FOR (Y) _n =1	
52	SElective * CL**	CLEAR (A) _n FOR (Y) _n =1	
53	SElective * SU**	Y _n → (A) _n FOR (Q) _n =1	
54	Replace SElective * SET	SET (A) _n FOR (Y) _n =1, → Y B A	
55	Replace SElective * CP	COMPLEMENT (A) _n FOR (Y) _n =1, → Y B A	
56	Replace SElective * CL	CLEAR (A) _n FOR (Y) _n =1, → Y B A	
57	Replace SElective * SU	(Y) _n → (A) _n FOR (Q) _n =1, → Y	
60	Jump (arithmetic)	Jump to Y if j-condition is satisfied.	
61	Jump (manual)	{see JP B RJP j-Designators}	
62	Jump (if * C ⁿ has ACTIVE INPUT buffer)	Jump to Y if C ⁿ input buffer active	(see JP B RJP j-Designators)
63	Jump (if * C ⁿ has ACTIVE OUTPUT buffer)	Jump to Y if C ⁿ output buffer active	
64	Return Jump (arithmetic)	Jump to Y+1 and P+1 → Y _L if j condition is satisfied (see JP B RJP j-Designators)	
65	Return Jump (manual)		
66	TERMINate * C ⁿ * INPUT	Terminate input buffer on channel j	
67	TERMINate * C ⁿ * OUTPUT	Terminate output buffer on channel j	
70	RePeat	Execute N _i Y times	
71	BSKip * B ⁿ	(B) _j = Y, skip N _i and clear (B) _j , (B) _j ≠ Y, Advance B _j and read N _i	
72	BJump * B ⁿ	(B) _j = 0, read N _i , (B) _j ≠ 0, (B) _j - 1 and jump to address Y	
73	INput * C ⁿ (without monitor mode)	Buffer IN on C ⁿ , k=3, (Y) → (00100 + j), k=1, (Y) _i → (00100 + j) _i , k=0, Y → (00100 + j)	
74	OUTput * C ⁿ (without monitor mode)	Buffer OUT on C ⁿ , k=3, (Y) → (00120 + j), k=1, (Y) _i → (00120 + j) _i , k=0, Y → (00120 + j)	
75	INput * C ⁿ (with MONITOR mode)	Buffer IN on C ⁿ with mon. k=3, (Y) → (00100 + j), k=1, (Y) _i → (00100 + j) _i , k=0, Y → (00100 + j), mon inter at 00040 ^j	
76	OUTput * C ⁿ (with MONITOR mode)	Buffer OUT on C ⁿ with mon. k=3, (Y) → (00120 + j), k=1, (Y) _i → (00120 + j) _i , k=0, Y → (00120 + j), mon inter at 00060 ^j	
	- NO - OPeration		} CS-1 Mono - codes
	- ComPlement * A or * Q		
	- CLear * A, * Q, * B ⁿ , or Y		
	- Remove Interrupt Lockout		
	- Remove Interrupt Lockout and Jump * Y		
	- TEST * CO or * CI		

**LP - Logical Product CP - Complement SU - Substitute CL - Clear

} Special j & k Designators (see opposite side of card) Y - The operand, Y or (Y)