NTDS UNIT COMPUTER

AN/USQ-20

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); $(A)_j = (A)_f$
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^j
13^	EXternal - FunCTion • C^n	$\hat{j} \neq 0$ or 1, (Y) → $C^{\hat{j}}$, $\hat{j} = 0$ or 1
14	STo Re • Q	(Q) → Y; k = 0, Q' → Q
15	STo Re • A	(A) → Y; k = 4, A' → A
16	STo Re • B^n	$(B)^j$ → Y
17^	STo Re • $C^{\hat{j}}$	$(C)^{\hat{j}}$ → 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) → Y & A
25	RePLace • A − Y	(A) − (Y) → Y & A
26*	ADD • Q	(Q) + Y → Q, $(A)_j = (A)_f$ } j interpretation
27*	SUBtract • Q	(Q) − Y → Q, $(A)_j = (A)_f$ } reversed for ABQ
30	ENTer • Y + Q	Y + (Q) → A
31	ENTer • Y − Q	Y − (Q) → A
32	STo Re • A + Q	(A) + (Q) → Y & A
33	STo Re • A − Q	(A) − (Q) → Y & A
34	RePLace • Y + Q	(Y) + (Q) → Y & A
35	RePLace • Y − Q	(Y) − (Q) → Y & A
36	RePLace • Y + 1	(Y) + 1 → Y & A
37	RePLace • Y − 1	(Y) − 1 → Y & A
40*	ENTer • LP**	L[Y(Q)] → A; 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)]; $(A)_j = (A)_f$
44*	RePLace • LP	L(Y)(Q) → Y & A; j = 2, even parity; j = 3, odd parity
45	RePLace • A + LP	L(Y)(Q) + (A) → Y & A
46	RePLace • A − LP	(A) − L(Y)(Q) → Y & A
47	STo Re • LP	L(A)(Q) → Y; $(A)_j = (A)_f$
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 & A
55	Replace SElective • CP	COMPLEMENT $(A)_n$ FOR $(Y)_n = 1$, → Y & A
56	Replace SElective • CL	CLEAR $(A)_n$ FOR $(Y)_n = 1$, → Y & 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 & RJP j - Designators)
62^	JumP (if • C^n has ACTIVE INput buffer)	Jump to Y if $C^{\hat{j}}$ input buffer active } (see JP & RJP j - Designators)
63^	JumP (if • C^n has ACTIVE OUTput buffer)	Jump to Y if $C^{\hat{j}}$ output buffer active } (see JP & RJP j - Designators)
64	Return JumP (arithmetic)	} Jump to Y + 1 and P + 1 → Y_L if j condition is
65	Return JumP (manual)	} satisfied (see JP & RJP j - Designators)
66^	TERMinate • C^n • INPUT	Terminate input buffer on channel $\hat{j}$
67^	TERMinate • C^n • OUTPUT	Terminate output buffer on channel $\hat{j}$
70*	RePeaT	Execute NI Y times
71	BSKip • B^n	$(B)^j = Y$, skip NI and clear $(B)^j$; $(B)^j \neq Y$, Advance B^j and read NI
72	BJumP • B^n	$(B)^j = 0$, read NI; $(B)^j \neq 0$, $(B)^j - 1$ and jump to address Y
73^	INput • C^n (without monitor mode)	Buffer IN on $C^{\hat{j}}$; $\hat{k} = 3$, (Y) → (00100 + $\hat{j}$); $\hat{k} = 1$, $(Y)_L$ → (00100 + $\hat{j}$)$_L$; $\hat{k} = 0$, Y → (00100 + $\hat{j}$)$_L$.
74^	OUTput • C^n (without monitor mode)	Buffer OUT on $C^{\hat{j}}$; $\hat{k} = 3$, (Y) → (00120 + $\hat{j}$); $\hat{k} = 1$, $(Y)_L$ → (00120 + $\hat{j}$)$_L$; $\hat{k} = 0$, Y → (00120 + $\hat{j}$)$_L$.
75^	INput • C^n (with • MONITOR mode)	Buffer IN on $C^{\hat{j}}$ with mon. $\hat{k} = 3$, (Y) → (00100 + $\hat{j}$); $\hat{k} = 1$, $(Y)_L$ → (00100 + $\hat{j}$)$_L$; $\hat{k} = 0$, Y → (00100 + $\hat{j}$)$_L$. mon. inter. at 00040 + $\hat{j}$
76^	OUTput • C^n (with • MONITOR mode)	Buffer OUT on $C^{\hat{j}}$ with mon. $\hat{k} = 3$, (Y) → (00120 + $\hat{j}$); $\hat{k} = 1$, $(Y)_L$ → (00120 + $\hat{j}$)$_L$; $\hat{k} = 0$, Y → (00120 + $\hat{j}$)$_L$. mon. inter. at 00060 + $\hat{j}$

— 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)

NTDS UNIT COMPUTER

AN/USQ-20

Repertoire of Instructions

JP & RJP j-DESIGNATORS

j	JP 60	RJP 64	JP 61	RJP 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	
$\hat{j}$	62 $\hat{j}$		63 $\hat{j}$	
$0\text{-}15_8$	C^n ACTIVE IN		C^n ACTIVE OUT	

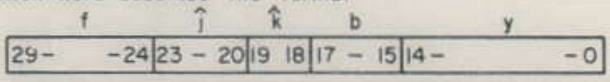
*60 Clears interrupt & bootstrap modes.

$\hat{j}$-DESIGNATORS

(4 bits)

$\hat{j}$ Occupies 4 bit positions and represents C^n where n may be $0—15_8$

The instruction word assumes the format:



$\hat{k}$-DESIGNATORS

(2 bits)

$\hat{k}$	EX–FCT 13	STR•C^n 17	JP 62 63	IN•C^n, OUT•C^n 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 04	DIV 23	ADD•Q, SUB•Q 26 27	ENT•LP, RPL•LP 40 44	RPT 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^b	
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 B R	Y of NE = Y+B^b [+B^6]	✓

✓ B^6 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 Code	READ Origin	STORE Code	STORE Dest.	REPLACE Code	REPLACE Origin	REPLACE 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