

**UNIVAC**  
**CP-823/U COMPUTER**  
**REPERTOIRE OF INSTRUCTIONS**

CODE (Octal)	INSTRUCTION	DESCRIPTION	Time = Sec
1			
01	Right Shift-Q	Shift (Q) Right by $\underline{Y}$	8-12
02	Right Shift-A	Shift (A) Right by $\underline{Y}$	8-12
03	Right Shift-AQ	Shift (AQ) Right by $\underline{Y}$	8-12
*04	COMpare-A,Q,-AQ	Sense (j); A, -A <sub>i</sub>	8
05	Left Shift-Q	Shift (Q) Left by $\underline{Y}$	8-12
06	Left Shift-A	Shift (A) Left by $\underline{Y}$	8-12
07	Left Shift-AQ	Shift (AQ) Left by $\underline{Y}$	8-12
10	ENTer-Q	$\underline{Y} \rightarrow Q$	8
10	CLear-Q	$\underline{Y}-0, Y \rightarrow Q$	8
11	ENTer-A	$\underline{Y} \rightarrow A$	8
11	CLear-A	$\underline{Y}-0, \underline{Y} \rightarrow A$	8
12	ENTer-B <sup>n</sup>	$\underline{Y} \rightarrow B^i$	12
12	CLear-B <sup>n</sup>	$\underline{Y}-0, \underline{Y} \rightarrow B^i$	12
12	NO-Operation	Enter B <sup>n</sup> with 0 (do nothing operation)	12
^13k0	EXternal-COMmand-C <sup>n</sup> -W(Y)-MONITOR	(Y) $\rightarrow C^i$ ; (interrupt at 00500+ <sub>i</sub> )	8
^13k1	EXternal-COMmand-C <sup>n</sup> -W(Y)-MONFORCE	Force (Y) $\rightarrow C^i$ ; (interrupt at 00500+ <sub>i</sub> )	8
^13k2	EXternal-COMmand-C <sup>n</sup> -W(Y)	(Y) $\rightarrow C^i$	8
^13k3	EXternal-COMmand-C <sup>n</sup> -W(Y)-FORCE	Force (Y) $\rightarrow C^i$	16
14	SToRe-Q	(Q) $\rightarrow Y$	8
14k≠0	ComPlement-Q	When $\underline{Y}$ is Q; then(Q) <sup>'</sup> $\rightarrow Q$	8
15	SToRe-A	(A) $\rightarrow Y$	8
16	SToRe-B <sup>n</sup>	(B <sup>i</sup> ) $\rightarrow Y$	12
^17k0	JumP- $\underline{Y}$ -C <sup>n</sup> -COMACTIVE	Jump to $\underline{Y}$ if external function buffer active	8
^17k1	JumP-L( $\underline{Y}$ )-C <sup>n</sup> -COMACTIVE	Jump to L( $\underline{Y}$ ) if external function buffer active	8
^17k3	SToRe-C <sup>n</sup> -W(Y)	00520+ <sub>i</sub> $\rightarrow Y$	12
^17k2	SToRe-C <sup>n</sup> -W(Y)-FORCE	Force $C^i \rightarrow Y$ -(abnormal test mode)	16-29
20	ADD-A	(A)+ $\underline{Y} \rightarrow A$	8
21	SUBtract-A	(A)- $\underline{Y} \rightarrow A$	8
22	MULTiPLY	(Q) $\underline{Y} \rightarrow AQ$	32-48
*23	DIVide	(AQ)/ $\underline{Y} \rightarrow Q$ ; R $\rightarrow A$ <sub>i</sub>	48
*23k≠71	SQuare Root-Q	$\sqrt{Q} \rightarrow Q$ ; residue $\rightarrow A$	48
24	RePLace-A+ $\underline{Y}$	(A)+(Y) $\rightarrow Y\&A$	12
25	RePLace-A- $\underline{Y}$	(A)-(Y) $\rightarrow Y\&A$	12
*26	ADD-Q	(Q)+ $\underline{Y} \rightarrow Q$	8
*27	SUBtract-Q	(Q)- $\underline{Y} \rightarrow Q$	8
30	ENTer- $\underline{Y}$ +Q	$\underline{Y}+Q \rightarrow A$	8
31	ENTer- $\underline{Y}$ -Q	$\underline{Y}-Q \rightarrow A$	8
32	SToRe-A+Q	(A)+(Q) $\rightarrow Y\&A$	12
33	SToRe-A-Q	(A)-(Q) $\rightarrow Y\&A$	12
34	RePLace- $\underline{Y}$ +Q	(Y)+(Q) $\rightarrow Y\&A$	12
35	RePLace- $\underline{Y}$ -Q	(Y)-(Q) $\rightarrow Y\&A$	12
36	RePLace- $\underline{Y}$ +1	(Y)+1 $\rightarrow Y\&A$	12
37	RePLace- $\underline{Y}$ -1	(Y)-1 $\rightarrow Y\&A$	12
*40	ENTer-LP	L [ $\underline{Y}$ (Q)] $\rightarrow A$	8
41	ADD-LP	L [ $\underline{Y}$ (Q)]+(A) $\rightarrow A$	8
42	SUBtract-LP	(A)-L [ $\underline{Y}$ (Q)] $\rightarrow A$	8
43	COMpare-MASK	(A)-L [ $\underline{Y}$ (Q)] sense (j), A+L [ $\underline{Y}$ (Q)]; (A)-(A) <sub>i</sub>	8
*44	RePLace-LP	L(Y)(Q) $\rightarrow Y\&A$	12
45	RePLace-A-LP	L(Y)(Q)+(A) $\rightarrow Y\&A$	12
46	RePLace-A-LP	(A)-L(Y)(Q) $\rightarrow Y\&A$	12

CODE (Octal)	INSTRUCTION	DESCRIPTION	Time μ Sec
47	SToRe-LP	L(A)(Q) → Y; (A), -(A);	8
50	SElective-SET	Set A <sub>n</sub> for Y <sub>n</sub> - 1	8
51	SElective-ComPlement	Complement A <sub>n</sub> for Y <sub>n</sub> - 1	8
51k4	ComPlement-A	When Y is 77777, then A <sup>1</sup> → A	8
52	SElective-CLear	Clear A <sub>n</sub> for Y <sub>n</sub> - 1	8
53	SElective-SUBstitute	Y <sub>n</sub> → A <sub>n</sub> for Q <sub>n</sub> - 1	8
54	ReplacE SElective-SET	Set A <sub>n</sub> for (Y) <sub>n</sub> - 1, → Y&A	12
55	ReplacE SElective-CP	Complement (A) <sub>n</sub> for (Y) <sub>n</sub> - 1 → Y&A	12
56	ReplacE SElective-CL	Clear A <sub>n</sub> for (Y) <sub>n</sub> - 1 → Y&A	12
57	ReplacE SElective-SU	(Y) <sub>n</sub> → A <sub>n</sub> for Q <sub>n</sub> - 1, → Y	12
*60	JumP (arithmetic)	Jump to Y if jump j-condition is satisfied	
60j0	Remove Interrupt Lockout	RIL on all internal channels and all external channels not locked out by SIL-EX (f66k2, 3)	12
60j1	Remove Interrupt Lockout JumP-Y	RIL; jump to Y	8
*61	JumP (manual)	Jump to Y if jump j-condition is satisfied	8
^62	JumP (if-C* has-ACTIVE INput buffer)	Jump to Y if C <sup>1</sup> input buffer active	8
^63	JumP (if-C* has-ACTIVE OUTput buffer)	Jump to Y if C <sup>2</sup> output buffer active	8
*64	Return JumP (arithmetic)	Jump to Y+1 and P-1 → Y <sub>j</sub> if j-condition	12
*65	Return JumP (manual)	is satisfied (see JP and RJP j-designators)	12
^66	TERMINate-C*INPUt	Terminates input buffer on C <sup>1</sup>	4
^66k1	Remove Interrupt Lockout-ALL	RIL on all internal channels and all external channels not locked out by SIL-EX (f66k2, 3)	4
^66k2	Remove Interrupt Lockout-EXternal-ALL	RIL for external interrupts on all channels	4
^66k3	Remove Interrupt Lockout-EXternal-C*	RIL for external interrupts on C <sup>1</sup>	4
^66k1b1	Set Interrupt Lockout-ALL	Sets external and internal lockout on all channels	4
^66k2b1	Set Interrupt Lockout-EXternal-ALL	Sets external interrupt lockout on all channels	4
^66k3b1	Set Interrupt Lockout-EXternal-C*	Sets external interrupt lockout on C <sup>1</sup>	4
^67	TERMINate-C*OUTPUt	Terminates output buffer on C <sup>2</sup>	4
^67k1	TERMINate-C*COMManD	Terminates external function buffer on C <sup>1</sup>	4
^67k2	TERMINate-ALL	Terminates ALL buffers	4
*70	RePeaT	Execute NI Y times	12
71	BSkip-B*	(B) <sub>j</sub> = Y, skip NI and clear B; (B) <sub>j</sub> ≠ Y, (B) <sub>j</sub> + 1 → B <sub>j</sub> and read NI	12
72	BJumP-B*	(B) <sub>j</sub> = 0, read NI; (B) <sub>j</sub> ≠ 0, (B) <sub>j</sub> - 1 → B <sub>j</sub> and jump to address Y	12
^73	INPUt-C* (without monitor mode)	Buffer IN on C <sup>1</sup> ; buffer control word → 00100+ <sup>1</sup>	12
^74	OUTPUt-C* (without monitor mode)	Buffer OUT on C <sup>2</sup> ; buffer control word → 00120+ <sup>1</sup>	12
^74k2	EXternal-COMManD-MultiWord-C*W(Y)	Buffer out on C <sup>1</sup> ; (Y) → 00140+ <sup>1</sup>	12
^75	INPUt-C* (with-MONITOR mode)	Buffer IN on C <sup>1</sup> with monitor; buffer control word → 00100+ <sup>1</sup> ; monitor interrupt address is 00040+ <sup>1</sup>	12
^76	OUTPUt-C* (with MONITOR mode)	Buffer OUT on C <sup>2</sup> with monitor; buffer control word → 00120+ <sup>1</sup> ; monitor interrupt address is 00060+ <sup>1</sup>	12
^76k2	EXternal-COMManD-MultiWord-C*W(Y)-MONITOR	Buffer OUT on C <sup>1</sup> ; (Y) → 00140+ <sup>1</sup> ; monitor interrupt address is 00050+ <sup>1</sup>	12

Y - The operand designator modified by B<sup>1</sup>

^ j | Special j and k designators

Y - The operand; Y or (Y)

NI - Next Instruction

**JP & RJP  
J-DESIGNATORS**

j	JP f60	RJP f64	JP f61	RJP f65
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	
j	62j		63j	
0-17 <sub>k</sub>	C <sup>n</sup> ACTIVE IN		C <sup>n</sup> ACTIVE OUT	

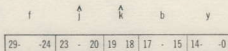
\*60 Clears interrupt mode.

**J-DESIGNATORS & K-DESIGNATORS**

<sup>^</sup> Occupies 4 bit positions and represents C<sup>n</sup> where n may be 0-17<sub>k</sub>.

<sup>k</sup> Controls operand interpretation, it is limited to 2 bit positions since the j-designator requires 4 bits.

The instruction word assumes the format:



**J-DESIGNATORS**

j	COM.A, Q, AQ f04	DIV f23	ADD Q, SUB-Q f26	ENT, LP, RPL, LP f40	RPT f70	SQRT f23 k7
0	(no skip)	(no skip)	(no skip)	(no skip)	(no mod.): Y of NE-Y	(no skip)
1	(unconditional skip)	SKIP	SKIP	SKIP	ADV : Y of NE-Y+1	SKIP
2	Y LESS : Y <sub>≤</sub> (Q)	NO OverFlow	A POS	EVEN parity	BACK : Y of NE-Y-1	REM
3	Y MORE : Y <sub>&gt;</sub> (Q)	OverFlow	A NEG	ODD parity	ADD B : Y of NE-Y+B <sup>n</sup>	NO REM
4	Y IN : (Q) <sub>≥</sub> Y and Y <sub>&gt;</sub> (A)	A ZERO	Q ZERO	A ZERO	Rpl. Inc. : Y of NE-Y[+B <sup>n</sup> ] <sub>v</sub>	'not used'
5	Y OUT : (Q)<Y or Y <sub>≤</sub> (A)	A NOT zero	Q NOT zero	A NOT zero	ADV R : Y of NE-Y+1[+B <sup>n</sup> ] <sub>v</sub>	'not used'
6	Y LESS : Y <sub>≤</sub> (A)	A POS	Q POS	A POS	BACK R : Y of NE-Y-1[+B <sup>n</sup> ] <sub>v</sub>	'not used'
7	Y MORE : Y <sub>&gt;</sub> (A)	A NEG	Q NEG	A NEG	ADD B R : Y of NE-Y+B <sup>n</sup> [+B <sup>n</sup> ] <sub>v</sub>	'not used'

<sub>v</sub>B<sup>n</sup> Increment if NI is RPL class; increments Y address for the store portion of the replace.

NE—Next execution

**NORMAL  
J-DESIGNATORS**

	(Not applicable on * or ^)
j	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

**LEGEND**

- M - Memory word (30 bits)
- M<sub>l</sub> - Lower half memory word
- M<sub>u</sub> - Upper half memory word
- X - Sign bit extended
- Cpl - Complement
- A - A-register
- Q - Q-register

**NORMAL  
K-DESIGNATORS**

k	READ		STORE		REPLACE		
	Code	Origin	Code	Dest.	Code	Origin	Dest.
0	'blank'	U <sub>L</sub>	Q	Q	'not used'	—	—
1	L	M <sub>L</sub>	L	M <sub>L</sub>	L	M <sub>L</sub>	M <sub>L</sub>
2	U	M <sub>u</sub>	U	M <sub>u</sub>	U	M <sub>u</sub>	M <sub>u</sub>
3	W	M	W	M	W	M	M
4	X	XU <sub>L</sub>	A	A	'not used'	—	—
5	LX	XM <sub>L</sub>	CPL	CplM <sub>L</sub>	LX	XM <sub>L</sub>	M <sub>L</sub>
6	UX	XM <sub>u</sub>	CPU	CplM <sub>u</sub>	UX	XM <sub>u</sub>	M <sub>u</sub>
7	A	A	CPW	CplM	'not used'	—	—

## CP-823/U MEMORY ADDRESS ASSIGNMENT

OCTAL ADDRESS RANGE	USE
Core Memory    00000-00017 00020-00037 00040-00057 00060-00077 00100-00117 00120-00137 00140-00157 00160 00161-00167 00170-00477 00500-00517 00520-00537 00540-00577 00600-00617 00620-67777	Unassigned External Interrupt Entrance Addresses Input Monitor Interrupt Entrance Addresses Output Monitor Interrupt Entrance Addresses Input Buffer Control Words Output Buffer Control Words External Function Buffer Control Words Real-Time Clock B-Registers Unassigned External Function Buffer Monitor Interrupt Entrance Addresses Interrupt Word Storage Addresses Unassigned Intercomputer Time Out Interrupt Entrance Addresses Unassigned
Film Memory    70000 70001 70002-77777	Fault Entrance Address (illegal instruction code) Power Restart Entrance Address Unassigned

**UNIVAC**  
DIVISION OF SPERRY RAND CORPORATION  
**DEFENSE MARKETING**  
UNIVAC PARK, ST. PAUL, MINN. 55118  
AREA CODE 612. 898-2461