

CP-642B COMPUTER REPERTOIRE OF INSTRUCTIONS

UNIVAC

DIVISION OF SPERRY RAND CORPORATION

f01	Right Shift•Q	Shift (Q) Right by \bar{Y}
f02	Right Shift•A	Shift (A) Right by \bar{Y}
f03	Right Shift•AQ	Shift (AQ) Right by \bar{Y}
*f04	COMpare•A•Q•AQ	Sense (j); $A_j = A_j$
f05	Left Shift•Q	Shift (Q) Left by \bar{Y}
f06	Left Shift•A	Shift (A) Left by \bar{Y}
f07	Left Shift•AQ	Shift (AQ) Left by \bar{Y}
f10	ENTer•Q	$\bar{Y} \rightarrow Q$
f10	CLear•Q	$\bar{Y}=0, \bar{Y} \rightarrow Q$
f11	ENTer•A	$\bar{Y} \rightarrow A$
f11	CLear•A	$\bar{Y}=0, \bar{Y} \rightarrow A$
f12	ENTer•B ⁿ	$\bar{Y} \rightarrow B^j$
f12	CLear•B ⁿ	$\bar{Y}=0, \bar{Y} \rightarrow B^j$
f12	NO-OPERation	Enter B ⁰ with 0 (do nothing operation)
^f13k0	EXternal-COMmand•C ⁿ •W(Y)•MONITOR	(Y) \rightarrow C ^j ; (interrupt at 00500•j)
^f13k2	EXternal-COMmand•C ⁿ •W(Y)	(Y) \rightarrow C ^j
^f13k3	EXternal-COMmand•C ⁿ •W(Y)•FORCE	(Y) \rightarrow C ^j ; (to be used on all CP642A/USQ-20 peripheral equipment)
f14	SToRe•Q	(Q) \rightarrow Y
f14k0	ComPlement•Q	When Y is Q; then Q' \rightarrow Q
f15	SToRe•A	(A) \rightarrow Y
f16	SToRe•B ⁿ	(B) ^j \rightarrow Y
^f17k0	JumP•Y•C ⁿ •COMACTIVE	Jump to Y if external function buffer active
^f17k1	JumP•L(Y)•C ⁿ •COMACTIVE	Jump to L(Y) if external function buffer active
^f17k3	SToRe•C ⁿ •W(Y)	00520•j \rightarrow (Y)
^f17k2	SToRe•C ⁿ •W(Y)•FORCE	Force C ^j \rightarrow (Y) - (abnormal test mode)
f20	ADD•A	(A)+Y \rightarrow A
f21	SUBtract•A	(A)-Y \rightarrow A
f22	MULTIply	(Q)Y \rightarrow AQ
*f23	DIVide	(AQ)/Y \rightarrow Q; R \rightarrow A _f
*f23k7	SQuare Root•Q	\sqrt{Q} \rightarrow Q; residue \rightarrow A
f24	RePLace•A+Y	(A)+(Y) \rightarrow Y&A
f25	RePLace•A-Y	(A)-(Y) \rightarrow Y&A
*f26	ADD•Q	(Q)+Y \rightarrow Q
*f27	SUBtract•Q	(Q)-Y \rightarrow Q
f30	ENTer•Y+Q	Y+Q \rightarrow A
f31	ENTer•Y-Q	Y-Q \rightarrow A
f32	SToRe•A+Q	(A)+(Q) \rightarrow Y&A
f33	SToRe•A-Q	(A)-(Q) \rightarrow Y&A
f34	RePLace•Y+Q	(Y)+(Q) \rightarrow Y&A
f35	RePLace•Y-Q	(Y)-(Q) \rightarrow Y&A
f36	RePLace•Y+1	(Y)+1 \rightarrow Y&A
f37	RePLace•Y-1	(Y)-1 \rightarrow Y&A
*f40	ENTer•LP	L [Y(Q)] \rightarrow A
f41	ADD•LP	L [Y(Q)]+(A) \rightarrow A
f42	SUBtract•LP	(A)-L [Y(Q)] \rightarrow A
f43	COMpare•MASK	(A)-L [Y(Q)] sense (j), A+L [Y(Q)]; (A) _i =(A) _i
*f44	RePLace•LP	L(Y)(Q) \rightarrow Y&A
f45	RePLace•A+LP	L(Y)(Q)+(A) \rightarrow Y&A
f46	RePLace•A-LP	(A)-L(Y)(Q) \rightarrow Y&A

CP-642B COMPUTER REPERTOIRE OF INSTRUCTIONS

UNIVAC

DIVISION OF SPERRY RAND CORPORATION

101	Right Shift•Q	Shift (Q) Right by \underline{Y}
102	Right Shift•A	Shift (A) Right by \underline{Y}
103	Right Shift•AQ	Shift (AQ) Right by \underline{Y}
*104	COMpare•A•Q•AQ	Sense (j); $A_j = A_j$
105	Left Shift•Q	Shift (Q) Left by \underline{Y}
106	Left Shift•A	Shift (A) Left by \underline{Y}
107	Left Shift•AQ	Shift (AQ) Left by \underline{Y}
110	ENTer•Q	$\underline{Y} \rightarrow Q$
110	CLear•Q	$\underline{Y}=0, \underline{Y} \rightarrow Q$
111	ENTer•A	$\underline{Y} \rightarrow A$
111	CLear•A	$\underline{Y}=0, \underline{Y} \rightarrow A$
112	ENTer•B ⁿ	$\underline{Y} \rightarrow B^j$
112	CLear•B ⁿ	$\underline{Y}=0, \underline{Y} \rightarrow B^j$
112	NO-Operation	Enter B ⁰ with 0 (do nothing operation)
^113k0	EXternal-COMmand•C ⁿ •W(Y)•MONITOR	(Y) → C ^j ; (interrupt at 00500•j)
^113k2	EXternal-COMmand•C ⁿ •W(Y)	(Y) → C ^j
^113k3	EXternal-COMmand•C ⁿ •W(Y)•FORCE	(Y) → C ^j ; (to be used on all CP642A/USQ-20 peripheral equipment)
114	SToRe•Q	(Q) → Y
114k0	Complement•Q	When Y is Q; then Q' → Q
115	SToRe•A	(A) → Y
116	SToRe•B ⁿ	(B) ^j → Y
^117k0	JumP•Y•C ⁿ •COMACTIVE	Jump to Y if external function buffer active
^117k1	JumP•L(Y)•C ⁿ •COMACTIVE	Jump to L(Y) if external function buffer active
^117k3	SToRe•C ⁿ •W(Y)	00520•j → (Y)
^117k2	SToRe•C ⁿ •W(Y)•FORCE	Force C ^j → (Y) - (abnormal test mode)
120	ADD•A	(A)+Y → A
121	SUBtract•A	(A)-Y → A
122	MULTiply	(Q)Y → AQ
*123	DIVide	(AQ)/Y → Q; R → A _j
*123k7	SQuare RooT•Q	\sqrt{Q} → Q; residue → A
124	RePLace•A+Y	(A)+(Y) → Y&A
125	RePLace•A-Y	(A)-(Y) → Y&A
*126	ADD•Q	(Q)+Y → Q
*127	SUBtract•Q	(Q)-Y → Q
130	ENTer•X+Q	X+Q → A
131	ENTer•X-Q	X-Q → A
132	SToRe•A+Q	(A)+(Q) → Y&A
133	SToRe•A-Q	(A)-(Q) → Y&A
134	RePLace•Y+Q	(Y)+(Q) → Y&A
135	RePLace•Y-Q	(Y)-(Q) → Y&A
136	RePLace•Y+1	(Y)+1 → Y&A
137	RePLace•Y-1	(Y)-1 → Y&A
*140	ENTer•LP	L [X(Q)] → A
141	ADD•LP	L [Y(Q)]+(A) → A
142	SUBtract•LP	(A)-L [X(Q)] → A
143	COMpare•MASK	(A)-L [Y(Q)] sense (j), A+L [Y(Q)]; (A) ₁ =(A) ₁
*144	RePLace•LP	L(Y)(Q) → Y&A
145	RePLace•A+LP	L(Y)(Q)+(A) → Y&A
146	RePLace•A-LP	(A)-L(Y)(Q) → Y&A

CP-642B COMPUTER REPERTOIRE OF INSTRUCTIONS

JP & RJP J-DESIGNATORS

j	JP f80	RJP f84	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	
\hat{j}	62 \hat{j}		63 \hat{j}	
0-17 ₈	C ⁿ ACTIVE IN		C ⁿ ACTIVE OUT	

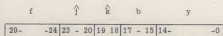
*60 Clears interrupt & bootstrap modes.

\hat{j} - DESIGNATORS & \hat{k} - DESIGNATORS

\hat{j} Occupies 4 bit positions and represents Cⁿ where n may be 0-17₈.

\hat{k} Controls operand interpretation, it is limited to 2 bit positions since the \hat{j} -designator requires 4 bits.

The instruction word assumes the format:



j - DESIGNATORS

j	COM•A,•Q,•AQ f04	DEV f23	ADD Q, SUB•Q f26 f27	ENT•LP, RPL•LP f40 f44	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 ≤ (Q)	NO OverFlow	A POS	EVEN parity	BACK : Y of NE = Y-1	REM
3	Y MORE : Y > (Q)	OverFlow	A NEG	ODD parity	ADD B : Y of NE = Y+B ^B	NO REM
4	Y IN : ((Q) > Y and Y > (A))	A ZERO	Q ZERO	A ZERO	Rpl. Inc. : Y of NE = Y + B ^B ✓	'not used'
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 ^B ✓	'not used'
6	Y LESS : Y ≤ (A)	A POS	Q POS	A POS	BACK R : Y of NE = Y-1 + B ^B ✓	'not used'
7	Y MORE : Y > (Q)	A NEG	Q NEG	A NEG	ADD B R : Y of NE = Y+B ^B + B ^B ✓	'not used'

✓ B^B Increment if NI is RPL class; increments Y address for the store portion of the replace.

NE - Next execution

NORMAL j - DESIGNATORS

j	Not applicable on * or ^)
0	Skip Code
1	(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

CP-642B MEMORY ADDRESS ASSIGNMENT

OCTAL
ADDRESS RANGE

USE

	00000 - 00000	Fault Entrance Address (With AUTOMATIC RECOVERY switch in center position) - illegal instruction code or illegal use of control memory
Core Memory	00001 - 00017 00020 - 00037 00040 - 00057 00060 - 00077	Unassigned External Interrupt Entrance Addresses Input Monitor Interrupt Entrance Addresses Output Monitor Interrupt Entrance Addresses
Control Memory	00100 - 00117 00120 - 00137 00140 - 00157 00160 - 00180 00161 - 00167 00170 - 00177	Input Buffer Control Words Output Buffer Control Words External Function Buffer Control Words Real-Time Clock B-Registers Unassigned
Core Memory	00200 - 00477 00500 - 00517 00520 - 00537	Unassigned External Function Buffer Monitor Interrupt Entrance Addresses Interrupt Word Storage Address
	00540 - 00577	Bootstrap - Wired Memory
Core Memory	00600 - 00617 00620 - 77777	Intercomputer Time Out Interrupt Entrance Address Unassigned

UNIVAC

DIVISION OF SPERRY RAND CORPORATION
DEFENSE MARKETING
UNIVAC PARK, ST. PAUL, MINN. 55116
AREA CODE 612, 696-2451