

CP-642B MEMORY ADDRESS ASSIGNMENT

DECIMAL 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 - 00160 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, 888-8491

UNIVAC CP-642B COMPUTER REPERTOIRE OF INSTRUCTIONS

SPERRY RAND CORPORATION
UNIVAC
Defense Systems Division

101	Right Shift•Q	Shift (Q) Right by Y
102	Right Shift•A	Shift (A) Right by Y
103	Right Shift•AQ	Shift (AQ) Right by Y
104	Compare•A•Q•AQ	Sense (J); A ₁ = A ₂
105	Left Shift•Q	Shift (Q) Left by Y
106	Left Shift•A	Shift (A) Left by Y
107	Left Shift•AQ	Shift (AQ) Left by Y
110	ENTER•Q	Y → Q
110	CLEAR•Q	Y=0, Y → Q
111	ENTER•A	Y → A
111	CLEAR•A	Y=0, Y → A
112	ENTER•B ⁰	Y → B ⁰
112	CLEAR•B ⁰	Y=0, Y → B ⁰
112	NO-Operation	Enter B ⁰ with 0 (do nothing operation)
^ 113k2	EXTERNAL-COMMAND•C ⁰ •W(Y)•MONITOR	(Y) → C ⁰ ; (interrupt at 00500•J)
^ 113k1	EXCOM-COMMAND•C ⁰ •W(Y)•MONFORCE	(Y) → C ⁰ ; (to be used on all CP-642A/USQ-20 peripheral equipment; (Interrupt at 00500 •J)
		(Y) → C ⁰
^ 113k2	EXTERNAL-COMMAND•C ⁰ •W(Y)	(Y) → C ⁰ ; (to be used on all CP-642A/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 ⁰	(B ⁰) → Y
^ 117k0	JUmP•Y•C ⁰ •COMACTIVE	Jump to Y if external function buffer active
^ 117k1	JUmP•L(Y)•C ⁰ •COMACTIVE	Jump to L(Y) if external function buffer active
^ 117k3	STORe•C ⁰ •W(Y)	00520•J → (Y)
^ 117k2	STORe•C ⁰ •W(Y)•FORCE	Force C ⁰ → (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 ₁
*123k7	SQuare ROOT	√(Q) → Q; remainder → 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•Y•Q	Y•Q → A
131	ENTER•Y•Q	Y•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•L•P	L [Y(Q)] → A
141	ADD•L•P	L [Y(Q)]•(A) → A
142	SUBtract•L•P	(A)•L [Y(Q)] → A
143	COMpare•MASK	(A)•L [Y(Q)] sense (J); A=L [Y(Q)]; (A) ₁ = (A) ₂
144	RePlAce•L•P	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

UNIVAC CP-642B COMPUTER

REPERTOIRE OF INSTRUCTIONS

147	SToRe*LP	L(A)Q → Y; (A) ₁ = (A) ₂
150	SElective*SEt	Set (A) ₁ for Y _n = 1
151	SElective*ComPlement	COMPLEMENT (A) _n for Y _n = 1
151k4	ComPlement*A	When Y is 7777, then A' → A
152	SElective*CLear	Clear (A) ₁ for Y _n = 1
153	SElective*Substitute	Y _n → (A) ₁ for (Q) _n = 1
154	Replace SElective*SEt	Set (A) ₁ for (Y) _n = 1, → Y&A
155	Replace SElective*SU	COMPLEMENT (A) _n for (Y) _n =1 → Y&A
156	Replace SElective*CL	Clear (A) ₁ for (Y) _n = 1 → Y&A
157	Replace SElective*SU	(Y) _n → (A) ₁ for (Q) _n =1, → Y
*160	Jump*P (arithmetic)	Jump to Y if jump j-condition is satisfied
160k0	Remove Interrupt Lockout	RIL on all internal channels and all external channels not locked out by SIL-EX
160j1	Remove Interrupt Lockout Jump*Y	RIL; jump to Y
*161	Jump*P (manual)	Jump to Y if jump j-condition is satisfied
^162	Jump*P (if C ^B has ACTIVE INPUT buffer)	Jump to Y if C ^B input buffer active
^163	Jump*P (if C ^B has ACTIVE OUTPUT buffer)	Jump to Y if C ^B output buffer active
*164	Return Jump*P (arithmetic)	Jump to Y-1 and P-1 → Y _L if j-condition is satisfied (see JP and RJP j-designators)
*165	Return Jump*P (manual)	Terminates input buffer on C ^B
^166	TERMinate*C ^B *INPUt	RIL on all internal channels and all external channels not locked out by SIL-EX
^166k1	Remove Interrupt Lockout*ALL	RIL for external interrupts on all channels
^166j3	Remove Interrupt Lockout-EXternal*ALL	RIL for external interrupts on C ^B
^166k3b1	Set Interrupt Lockout*ALL	Sets external and internal lockout on all channels
^166k2b1	Set Interrupt Lockout-EXternal*ALL	Sets external interrupt lockout on all channels
^166k3b1	Set Interrupt Lockout-EXternal*ALL	Sets external interrupt lockout on C ^B
^167	TERMinate*C ^B *OUTPUt	Terminates output buffer on C ^B
^167k1	TERMinate*C ^B *COMManD	Terminates external function buffer on C ^B
^167k2	TERMinate*ALL	Terminates ALL buffers
*170	RePeaT	Execute NI Y times
171	BSkip*B ^B	(B) ₁ ≠X, skip NI and clear (B) ₁ ; (B) ₁ ≠Y, advance B ^B and read NI
172	BJump*B ^B	(B) ₁ =0, read NI; (B) ₁ ≠0, (B) ₁ -1 and jump to address Y
^173	INput*C ^B (without monitor mode)	Buffer IN on C ^B ; buffer control word → 00100 ¹
^174	OUTput*C ^B (without monitor mode)	Buffer OUT on C ^B ; buffer control address → 00120 ¹
^174k2	EXTERNAL-COMmand-MultiWords*C ^B *W(Y)	Buffer out on C ^B ; (Y) → (00140 ¹)
^175	INput*C ^B (with MONITOR mode)	Buffer IN on C ^B with monitor; buffer control word → 00100 ¹ ; monitor interrupt address → (00040 ¹)
^176	OUTput*C ^B (with MONITOR mode)	Buffer OUT on C ^B with monitor; buffer control address → 00120 ¹ ; monitor interrupt address → (00060 ¹)
^176k2	EXTERNAL-COMmand-MultiWords*C ^B *W(Y)	Buffer OUT on C ^B ; interrupt at → (00500 ¹)
MONITOR		

* } Special j and k designators
 Y - The operand; Y or (Y)

UNIVAC CP-642B COMPUTER

REPERTOIRE OF INSTRUCTIONS

JP & RJP

J-DESIGNATORS		RJP	
j	#60	#64	#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	
↑	82 ↑	63 ↑	
0-17 _g	C ^B ACTIVE IN	C ^B ACTIVE OUT	

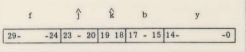
*60 Clears interrupt & bootstrap modes.

J-DESIGNATORS & K-DESIGNATORS

↑ Occupies 4 bit positions and represents C^B where n may be 0-17_g.

⌘ 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 104	DEV 123	ADD Q, SUB*Q 127	ENT*LP, RPL*LP 140	RPT 170	SQRT 223 x7
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)	NOOverFlow	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 _L 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 _L B ^B	not used
6	Y LESS : Y < (A)	A POS	Q POS	A POS	BACK R : Y of NE = Y-1 _L B ^B	not used
7	Y MORE : Y > (A)	A NEG	Q NEG	A NEG	ADD B R : Y of NE = Y+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	(no skip)
1	SKIP
2	Q POS
3	Q NEG
4	A ZERO
5	A NOT zero
6	UX XMtU CPt Cpl MtU
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	CPt	Cpl	LX	XM _L M _L
6	UX	XMtU	CPt	Cpl	UX	XMtU M _U
7	A	A	CPW	Cpl	'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