

UNIVAC® CP-667 COMPUTER

REPertoire OF INSTRUCTIONS

f001	Right SHift•Q	Shift (Q) Right by Y
f002	Right SHift•A	Shift (A) Right by Y
f003	Right SHift•AQ	Shift (AQ) Right by Y
*f004	COMpare•A•Q•AQ	Sense (j); $A_i = A_f$
f005	Left SHift•Q	Shift (Q) Left by Y
f006	Left SHift•A	Shift (A) Left by Y
f007	Left SHift•AQ	Shift (AQ) Left by Y
f010	ENTer•Q	$\bar{Y} \rightarrow Q$
f010	CLear•Q	$\bar{Y}=0, \bar{Y} \rightarrow Q$
f011	ENTer•A	$\bar{Y} \rightarrow A$
f011	CLear•A	$\bar{Y}=0, \bar{Y} \rightarrow A$
^f012	ENTer•B ⁿ	$\bar{Y} \rightarrow B_j$
^f012	CLear•B ⁿ	$\bar{Y}=0, \bar{Y} \rightarrow B_j$
f012	NO-Operation	Enter B ₀ with 0 (do nothing operation)
^f013k0	EXternal-COMmand•C ⁿ •W(Y)•MONITOR	(Y) → C _j ; (interrupt at 000500 _j)
^f013k1	EXternal-COMmand•C ⁿ •W(Y)•MONFORCE	Y → C _j ; (interrupt at 000500 _j for use on CP642A equipment)
^f013k2	EXternal-COMmand•C ⁿ •W(Y)	(Y) → C _j
^f013k3	EXternal-COMmand•C ⁿ •W(Y)•FORCE	(Y) → C _j ; (to be used on all CP642A/USQ-20 peripheral equipment)
f014	SToRe•Q	(Q) → Y
f014	Complement•Q	When Y is Q; then Q' → Q
f015	SToRe•A	(A) → Y
^f016	SToRe•B ⁿ	(B _j) → Y
f016	CLear•Y	Zero → Y
^f017k0	JumP•Y•C ⁿ •COMACTIVE	Jump to Y if external command buffer active
^f017k1	JumP•L(Y)•C ⁿ •COMACTIVE	Jump to L(Y) if external command buffer active
^f017k3	SToRe•C ⁿ •W(Y)	000520 _j → (Y)
^f017k2	SToRe•C ⁿ •W(Y)•FORCE	Force C _j → (Y) (abnormal test mode)
f020	ADD•A	(A)+Y → A
f021	SUBtract•A	(A)-Y → A
f022	MULTiply•Q	(Q)Y → AQ
f023	DIVide•AQ	(AQ)/Y → Q; R → A _f
*f023k7	SQware RooT•Q	$\sqrt{ Q } \rightarrow Q$; remainder → A
f024	RePLace•A•Y	(A)+(Y) → Y&A
f025	RePLace•A•Y	(A)-(Y) → Y&A
*f026	ADD•Q	(Q)+Y → Q
*f027	SUBtract•Q	(Q)-Y → Q
f030	ENTer•Y+Q	Y+Q → A
f031	ENTer•Y-Q	Y-Q → A
f032	SToRe•A+Q	(A)+(Q) → Y&A
f033	SToRe•A-Q	(A)-(Q) → Y&A
f034	RePLace•Y+Q	(Y)+(Q) → Y&A
f035	RePLace•Y-Q	(Y)-(Q) → Y&A
f036	RePLace•Y+1	(Y)+1 → Y&A
f037	RePLace•Y-1	(Y)-1 → Y&A
f040	ENTer•LP	L [Y(Q)] → A
f041	ADD•LP	L [Y(Q)] + (A) → A
f042	SUBtract•LP	(A) - L [Y(Q)] → A
f043	COMpare•MASK	A - L [Y(Q)] sense (j), A+L [Y(Q)]; (A) ₁ = (A) _f
f044	RePLace•LP	L(Y)(Q) → Y&A
f045	RePLace•A+LP	L(Y)(Q)+(A) → Y&A
f046	RePLace•A-LP	(A)-L(Y)(Q) → Y&A

↑Q47	SToRe•LP	L(A)Q → Y (A) ₁ = (A) ₁
↑Q50	SELEctive•SET	Set (A) _n for Y _n = 1
↑Q51	SELEctive•Complement	COMPLEMENT (A) _n for Y _n =1
↑Q51k4	Complement•A	When Y is 77777, then A' → A
↑Q52	SELEctive•CLear	Clear (A) _n for Y _n =1
↑Q53	SELEctive•Substitute	Y _n → (A) _n for (Q) _n =1
↑Q54	Replace SELEctive•SET	Set (A) _n for (Y) _n =1, → Y&A
↑Q55	Replace SELEctive•CP	COMPLEMENT (A) _n for (Y) _n =1 → Y&A
↑Q56	Replace SELEctive•CL	Clear (A) _n for (Y) _n =1 → Y&A
↑Q57	Replace SELEctive•SU	(Y) _n → (A) _n for (Q) _n =1, → Y
*↑Q60	Jump (arithmetic)	Jump to Y if special j-condition is satisfied
^↑Q60j1	Remove Interrupt Lockout Jump•Y	RIL; jump to Y
*↑Q61	Jump (manual)	Jump to Y if special j-condition is satisfied
^↑Q62	Jump (if C ⁿ has•ACTIVE Input buffer)	Jump to Y if C ⁿ input buffer active
^↑Q63	Jump (if C ⁿ has•ACTIVE OUTPut buffer)	Jump to Y if C ⁿ output buffer active
*↑Q64	Return JumpP (arithmetic)	Jump to Y+1 and P+1 → Y ₁ if j-condition is satisfied (see JP&RJP j-Designators)
*↑Q65	Return JumpP (manual)	Terminates input buffer on C ⁿ
^↑Q66	TERMiNate•C ⁿ •INPut	RIL on all internal channels and all external channels not locked out by SIL-EX
^↑Q66k1	Remove Interrupt Lockout•ALL	RIL for external interrupts on all channels
^↑Q66k2	Remove Interrupt Lockout-EXtErnal•ALL	Settings external and internal lockout on all channels
^↑Q66k3	Remove Interrupt Lockout-EXtErnal•C ⁿ	Settings external interrupt lockout on all channels
^↑Q66k1b1	Set Interrupt Lockout•ALL	Settings external interrupt lockout on C ⁿ
^↑Q66k2b1	Set Interrupt Lockout-EXtErnal•ALL	Terminates output buffer on C ⁿ
^↑Q66k3b1	Set Interrupt Lockout-EXtErnal•C ⁿ	Terminates external command buffer on C ⁿ
^↑Q67	TERMiNate•C ⁿ •OUTPut	Terminates ALL buffers
^↑Q67k1	TERMiNate•C ⁿ •COMmAnd	Execute NI Y times
^↑Q67k2	TERMiNate•ALL	(B) ^j =Y, skip NI and clear (B) ^j ; (B) ^j ≠Y, advance B ^j and read NI
*↑Q70	RePeAT	(B) ^j =0, read NI; (B) ^j ≠0, (B) ^j -1 and jump to address Y
^↑Q71	BSKip•B ⁿ	Buffer IN on C ⁿ ; buffer control word → 000100 [†]
^↑Q72	BJump•B ⁿ	Buffer OUT on C ⁿ ; buffer control address → 000120 [†]
^↑Q73	INPut•C ⁿ (without monitor mode)	Buffer OUT on C ⁿ ; (Y) → (000140 [†])
^↑Q74	OUTPut•C ⁿ (without monitor mode)	Buffer IN on C ⁿ with monitor; buffer control word → 00100 [†] ; monitor interrupt address → (000040 [†])
^↑Q74k2	EXtErnal-COMmAnd-MultiWord•C ⁿ •W(Y)	Buffer OUT on C ⁿ with monitor; buffer control address → 000120 [†] ; monitor interrupt address → (000060 [†])
^↑Q75	INPut•C ⁿ (with•MONITOR mode)	Buffer OUT on C ⁿ ; interrupt at → (000500 [†])
^↑Q76	OUTPut•C ⁿ (with•MONITOR mode)	MONITOR
^↑Q76k2	EXtErnal-COMmAnd-MultiWord•C ⁿ •W(Y)•MONITOR	

*J-DESIGNATORS

	COM *A,*Q,*AQ	COM *AQ I104	ADD*Q, SUB*Q I26 I27	SORT I23 K7	MANUAL JP RJP I61 I65	RPT		
						CODE	ORIGIN	DESTINATION
1	COM *A,*Q,*AQ I04	(no skip)	(no skip)	(no skip)	(no. mod.)	$Y_{n+1} = Y$		
0	(no skip)	(uncon, skip)	SKIP	SKIP	ADV	$Y_{n+1} = Y \cdot N$		
1	(uncon, skip)	(uncon, skip)	SKIP	REMAINDER	BACK	$Y_{n+1} = Y \cdot N$		
2	YLESS: $\bar{Y} < (Q)$	'not used'	APOS	REMAINDER	BACK	$Y_{n+1} = Y \cdot N$		
3	YMORE: $Y > (Q)$	'not used'	ANEG	NONREMAINDER	KEY3	$Y_{n+1} = Y \cdot N(B^6)$		
4	YIN: $(Q) \geq \bar{Y} \& \bar{Y} > A$	REQ: $\bar{Y}+1, Y=AQ$	QZERO	KEY3	STOP	$Y_{n+1} = Y$	$Y_{n+1} = Y + (B^6)$	
5	YOUT: $(Q) < \bar{Y}$ or $\bar{Y} < A$	YNOT: $\bar{Y}+1, Y=AQ$	QNOT zero	STOP5	Rpl Inc	$Y_{n+1} = Y \cdot N$	$Y_{n+1} = Y + N + (B^6)$	
6	YLESS: $\bar{Y} < (A)$	YLESS: $\bar{Y}+1, Y < AQ$	QPOS	STOP6	BACK	$Y_{n+1} = Y \cdot N$	$Y_{n+1} = Y \cdot N + (B^6)$	
7	YMORE: $\bar{Y} > (A)$	YMORE: $\bar{Y}+1, \bar{Y} > AQ$	QNEG	STOP7	ADDBR	$Y_{n+1} = Y \cdot N(B^6)$	$Y_{n+1} = Y + N(B^6) + (B^6)$	
10			AZERO	TMOVE	TMOVE	$Y_{n+1} = Z + (B^6)_{10}$	$Y_{n+1} = Z + (B^6)_{10} + N(B^6)_{10}$	
11			ANOT zero		MOVE	$Y_{n+1} = Z + (B^6)_{11}$	$Y_{n+1} = Z + (B^6)_{11} + N(B^6)_{11}$	
12			AQZERO					
13			AQNOTzero					
14			OF					
15			NOOF					

Z=Y without B-Register Modification

NORMAL

	(Not applicable on * or ^)	ARITHMETIC
J	Skip Code	JP RJP I64
0	(no skip)	No jump)
1	SKIP	(uncon, jump)
2	QPOS*	QPOS
3	QNEG**	QNEG
4	AZERO	AZERO
5	ANOT zero	ANOT zero
6	APOS	APOS
7	ANEG	ANEG
10	QZERO	QZERO
11	QNOT zero	QNOT zero
12	AQZERO	AQZERO
13	AQNOT zero	AQNOT zero
14	OF	
15	NOOF	

LEGEND

- M - Memory word (36 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

NORMAL K-DESIGNATORS

k	READ	STORE	REPLACE				
k	Code	Origin	Code	Origin	Code	Origin	Dest.
0	'blank'	UL	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	XMU	CPU	Cpl M _U	UX	XM _U	M _U
7	A	A	CPW	Cpl M	'not used'		

* EVEN parity used for RPL•LP&ENT•LP

** ODD parity used for RPL•LP&ENT•LP

* NO Over Flow used for DIV*AQ

** Over Flow used for DIV*AQ

000j
000j
000j

f101k3	ADD•FP	(A) + Y → A, Residue → Q
f102k3	SUBtract•FP	(A) - \bar{Y} → A, Residue → Q
f103	ENter eXtended•Q	\bar{Y} → Q, sign bit extended through A
f103	CLear•AQ	\bar{Y} = 0; \bar{Y} → Q, sign Bit (0) → A
*f104k3	COMpare•AQ	(AQ)-(Y+1, Y) sense (j), (AQ+Y+1, Y), (AQ) _i =(AQ) _f
f105k3	PACK•FP	Mantissa (AQ)+ characteristic (Y) → A
f106k3	UNPACK•FP	(A) → mantissa → A & characteristic → \bar{Y}
f107k3	NORMALize•AQ	Shift (AQ) left until A34 ≠ A35, number of shifts → \bar{Y}
f110	USE•REG	Activates register group \bar{Y} , contents of A&Q of \bar{Y} register group → A&Q
f111k3	ENter•AQ	\bar{Y} + 1 → A, \bar{Y} → Q
^f112	ENter•UB ⁿ	\bar{Y} → B _u ^j
^f112	CLear•UB ⁿ	\bar{Y} =0; \bar{Y} → B _u ^j
f113	ENter•Subtractive Real Time Clock	\bar{Y} → (000240)
f114	SToRe•REG	(A) & (Q) → Active Register Group A & Q Memory Addresses
f115k3	SToRe•AQ	(A) → Y+1, Q → Y
f115k4	ComPlement•AQ	AQ ⁱ → AQ
^f116	SToRe•UB ⁿ	(B _u ^j) _i → Y
f117k0	ENABLE•FP-ROUND	Reinitiate rounding of the mantissa
f117k1	DISABLE•FP-ROUND	Suspend rounding of the mantissa
f120k3	ADD•AQ	(AQ)+(Y+1, Y) → AQ
f121k3	SUBtract•AQ	(AQ)-(Y+1, Y) → AQ
f122k3	MULTiply•FP	(A) · Y → A, Residue → Q
f123k3	DIVide•FP	(A)/ \bar{Y} → \bar{A} ; R → Q _f
^f124	TRANsfer• [Y] + B ^b • [Y] + B ^j	[Y + B ^b] → [Y + B ^j]
f130	ENABLE•CORE-LOCK	Y → memory lock out register (see manual for format)
^f160	Load B & Jump•B ⁿ	P+1 address → B _u ^j , jump to \bar{Y}
f166b0	ENABLE•EXternal Synchronizing Interrupt	Enable interrupts
f166b1	DISABLE•EXternal Synchronizing Interrupt	Disable interrupts
^f171	Multiple BSKip•B ⁿ	(B _u ^j) _i ≥ \bar{Y} , Skip NI, (B _u ^j) _i = (B _u ^j) _f ; B _u ^j < \bar{Y} , advance B _u ^j by (B _u ^j) _u and execute NI
^f172	ADD•B ⁿ	(B _u ^j) + \bar{Y} → B _u ^j
^f173	SUBtract•B ⁿ	(B _u ^j) - \bar{Y} → B _u ^j
^f125	EXecute Remote Instruction	Executes instruction located in memory as specified by \bar{Y}

* Special j designator

^ j designators specify a b register, channel, or sub-function

\bar{Y} The operand; Y or (Y)

(i) is the indirect indicator, i.e., ENT•A•W(i) (CAT)

A line under a number indicates that it is binary (i.e., 005 = 0000101 in binary)

Channel numbers and b registers are specified in octal.

The whole of a memory location (W) or a constant must be defined when used with instructions whose k designator is restricted to 3.