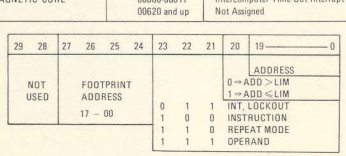
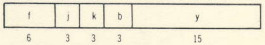
EAM CP-642B COMPUTER MEMORY ASSIGNMENTS

MEMORY TYPE	MEMORY ADDRESS	FUNCTION
MAGNETIC CORE	00000	Fault Entrance
	00001-00015	Not Assigned
	00016	Address Out Of Range/ Parity Error Interrupt Entrance
	00017	Memory Resume Time-Out Interrupt Entrance
	00020-00037	External Interrupt Entrance
	00040-00057	Input Monitor Interrupt Entrance
	00060-00077	Output Monitor Interrupt Entrance
IC CONTROL MEMORY	00100-00117	Input Buffer Control Registers
	00120-00137	Output Buffer Control Registers
	00140-00157	External Function Buffer Control Registers
	00160	Real-Time Clock
	00161-00167	B1 Through B7 Index Registers
	00170-00177	Not Assigned
MAGNETIC CORE	00200-00277	Not Assigned
IC IN EAM MODE MAGNETIC CORE OTHERWISE	00300-00317	I/O Base Registers ($IR_0 - IR_{17}$)
	00320-00327	Operand & Instr. Base Registers ($R_0 - R_7$)
	00330	Common Access Register (CAR)
	00331	Limit Register (LIM)
	00332-00336	Not Used
	00337	Footprint Control/Access
MAGNETIC CORE	00340-00477	Not Assigned
	00500-00517	External Function Monitor Interrupt Entrance
	00520-00537	External Interrupt Code Storage
ROM	00540-00577	NDRO Bootstrap Prog. I and II
MAGNETIC CORE	00600-00617	Intercomputer Time-Out Interrupt Entrance
	00620 and up	Not Assigned



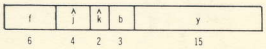
FOOTPRINT REGISTER – ADDRESS 337

Two forms of instruction words are intergreted by the computer and contain either the f (6 bits), j (3 bits), k (3 bits), b (3 bits) and y (15 bits) or the f (6 bits), $\hat{j}$ (4 bits), $\hat{k}$ (2 bits), b (3 bits) and y (15 bits) designators as shown below.



General instruction format

No. of bits 6 3 3 3 15



I/O instruction format

No. of bits 6 4 2 3 15

SPERRY UNIVAC DEFENSE SYSTEMS

UNIVAC CP-642B COMPUTER
REPERTOIRE OF INSTRUCTIONS

NORMAL j-DESIGNATORS

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_L—U-register, lower half

SPECIAL *j-DESIGNATORS

j	SKIP CONDITIONS: COM.A, .Q, .AQ f04	DIV f23	ADD.Q, SUB.Q f26 – f27	ENT.LP, RPL.LP f40 – f44	SQRT f23 – k7	ADDRESS MODIFICATIONS: RPT f70
0	(no skip)	(no skip)	(no skip)	(no skip)	(no skip)	(no mod.) : Y of NE = Y
1	(unconditional skip)	SKIP	SKIP	SKIP	SKIP	ADV : Y of NE = Y + 1
2	Y LESS : Y ≥ (Q)	NO Over Flow	A POS	EVEN parity	'not used'	BACK : Y of NE = Y − 1
3	Y MORE : Y > (Q)	Over Flow	A NEG	ODD parity	'not used'	ADD B : Y of NE = Y + B^b
4	Y IN (Q) ≥ Y and Y > (A)	A ZERO	Q ZERO	A ZERO	NO REM	Rpl. inc. : Y of NE = Y|+B^k| ↘
5	Y OUT (Q) < Y or Y < (A)	A NOT Zero	Q NOT Zero	A NOT Zero	REM	ADV R : Y of NE = Y+1|+B^k| ↘
6	Y LESS : Y ≥ (A)	SKIP	Q POS	A POS	'not used'	BACK R : Y of NE = Y−1|+B^k| ↘
7	Y MORE : Y > (A)	no skip	Q NEG	A NEG	'not used'	ADD BR : Y of NE = Y+B^k|+B^k| ↘

If NI is RPL class, B^k increments Y address for the store portion of the replace.
NE—Next execution.

NORMAL b-DESIGNATOR

b=0: Do not modify y
b=1: Add (B1) to y
b=2: Add (B2) to y
b=3: Add (B3) to y
b=4: Add (B4) to y
b=5: Add (B5) to y
b=6: Add (B6) to y
b=7: Add (B7) to y

$\hat{k}$-DESIGNATORS

f60, 61, 64, 65
k=0, 4 Jump to Y
k=1, 3, 5 Jump to (Y)L
k=2, 6 Jump to (Y)U
k=7 Jump to the Address in A.

JP & RJP $\hat{j}$ & *j-DESIGNATORS

$\hat{j}$	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	
$\hat{k}$	62 $\hat{j}$		63 $\hat{j}$	
0-17$_8$	C^a ACTIVE IN		C^a ACTIVE OUT	

*60 Clears interrupt & bootstrap modes.

CP-642B COMPUTER
REPERTOIRE OF INSTRUCTIONS

Code (Octal)	Instruction	Description
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); $A_i=A_r$
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
10	CLear•Q	$\overline{Y}=0$, Y→Q
11	ENTer•A	Y→A
11	CLear•A	$\overline{Y}=0$, Y→A
12	ENTer•B^n	Y→B^i
12	CLear•B^n	$\overline{Y}=0$, Y→B^i
12	NO-OPeration	Enter B^n with 0 (do nothing operation)
^13k0	EXternal-COMmand•C^n•W(Y)•MONITOR	(Y)→C^i; (interrupt at 00500+j)
^13k1	EXcom-COMmand•C^n•W(Y)•MONFORCE	(Y)→C^i; (interrupt at 00500+j)
^13k2	EXternal-COMmand•C^n•W(Y)	(Y)→C^i
^13k3	EXternal-COMmand•C^n•W(Y)•FORCE	(Y)→C^i
14	SToRe•Q	(Q)→Y
14k0	ComPlement•Q	When Y is Q; then Q^i→Q
15	SToRe•A	(A)→Y
16	SToRe•B^n	(B) →Y
^17k0	JumP•Y•C^n•COMACTIVE	Jump to Y if external function buffer active
^17k1	JumP•L(Y)•C^n•COMACTIVE	Jump to L(Y) if external function buffer active
^17k2	SToRe•C^n•W(Y)•FORCE	Force (C)→Y—(abnormal test mode)
^17k3	SToRe•C^n•W(Y)	(00520+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
*23k7	SQuare RooT	⌊|Q|→Q; remainder→A
24	RePLace•A+Y	(A)+(Y)→Y&A
25	RePLace•A−Y	(A)−(Y)→Y&A
*26	ADD•Q	(Q)+Y→Q
*27	SUBtract•Q	(Q)−Y→Q
30	ENTer•Y+Q	Y+Q→A
31	ENTer•Y−Q	Y−Q→A
32	SToRe•A+Q	(A)+(Q)→Y&A
33	SToRe•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
41	ADD•LP	L(Y(Q))+(A)→A
42	SUBtract•LP	(A)−L(Y(Q))→A

UNIVAC CP-642B COMPUTER
REPERTOIRE OF INSTRUCTIONS

Code (Octal)	Instruction	Description
43	COMpare•MASK	(A)−L(Y(Q)) sense (j); A+L(Y(Q)); $(A)_i=(A)_r$
44	RePLace•LP	L(Y(Q))→Y&A
45	RePLace•A+LP	(A)+L(Y(Q))→Y&A
46	RePLace•A−LP	(A)−L(Y(Q))→Y&A
47	SToRe•LP	L(A)(Q)→Y; (A)→(A)$_r$
50	SELective•SET	Set $(A)_r$ for $Y_n=1$
51	SELective•ComPlement	Complement $(A)_r$ for $Y_n=1$
51k4	ComPlement•A	When Y is 77777, then A^i→A
52	SELective•CLear	Clear $(A)_r$ for $Y_n=1$
53	SELective•SUbstitute	Y_n→$(A)_r$ for $(Q)_n=1$
54	Replace SElective•SET	Set $(A)_r$ for $(Y)_n=1$→Y&A
55	Replace SElective•CP	Complement $(A)_r$ for $(Y)_n=1$→Y&A
56	Replace SElective•CL	Clear $(A)_r$ for $(Y)_n=1$→Y&A
57	Replace SElective•SU	$(Y)_n$→$(A)_r$ for $(Q)_n=1$→Y
*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
60j1	Remove Interrupt Lockout JumP•Y	RIL; jump to Y
*61	JumP (manual)	Jump to Y if jump j-condition is satisfied
^62	JumP (if•C^n has•ACTIVE INput buffer)	Jump to Y if C^i input buffer active
^63	JumP (if•C^n has•ACTIVE OUTput buffer)	Jump to Y if C^i output buffer active
*64	Return JumP (arithmetic)	Jump to Y+1 and P+1→Y if j-condition
*65	Return JumP (manual)	is satisfied (see JP and RJP j-designators)
^66	TERMinate•C^n•INPUT	Terminates input buffer on C^i
^66k1b0	Remove Interrupt Lockout•ALL	RIL on all internal channels and all external channels not locked out by SIL-EX
^66k2b0	Remove Interrupt Lockout-EXternal•ALL	RIL for external interrupts on all channels
^66k3b0	Remove Interrupt Lockout-EXternal•C^n	RIL for external interrupts on C^i
^66k1b$\overline{0}$	Set Interrupt Lockout•ALL	Sets external and internal lockout on all channels
^66k2b$\overline{0}$	Set Interrupt Lockout-EXternal•ALL	Sets external interrupt lockout on all channels
^66k3b$\overline{0}$	Set Interrupt Lockout-EXternal•C^n	Sets external interrupt lockout on C^i
^67	TERMinate•C^n•OUTPUT	Terminates output buffer on C^i
^67k1	TERMinate•C^n•COMmand	Terminates external function buffer on C^i
^67k2	TERMinate•ALL	Terminates all buffers
*70	RePeaT	Execute NI Y times
71	BSKip•B^n	$(B)^i=\overline{Y}$, skip NI and clear $(B)^i$; $(B)^i\neq\overline{Y}$, advance B^i and read NI
72	BJumP•B^n	$(B)^i=0$, read NI; $(B)^i\neq0$, $(B)^i-1$ and jump to address Y
^73	INput•C^n (without monitor mode)	Buffer in on C^i; buffer control word→00100+j
^74	OUTput•C^n (without monitor mode)	Buffer out on C^i; buffer control word→00120+j
^74k2	EXternal-COMmand-MultiWord•C^n•W(Y)	Buffer out on C^i; (Y)→(00140+j)
^75	Input•C^n (with•MONITOR mode)	Buffer in on C^i with monitor; buffer control word→00100+j; (interrupt address 00040+j)
^76	OUTput•C^n (with MONITOR mode)	Buffer out on C^i with monitor; buffer control address→00120+j; (interrupt address 00060+j)
^76k2	EXternal-COMmand-MultiWord•C^n•W(Y)•MONITOR	Buffer out on C^i; (interrupt at 00500+j)
7744	Test and Set Flag	$(Y)_{29}\neq0$, Execute NI; $(Y)_{29}=0$, skip NI and set $(Y)_{29}$

j, k Special j and k designators (* j, $\hat{j}$ or $\hat{k}$)

Y—The operand; Y, (Y) or (A)