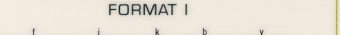
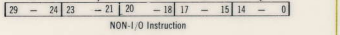
INSTRUCTION WORD FORMATS

FORMAT I

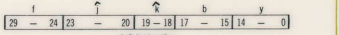


NON-I/O Instruction

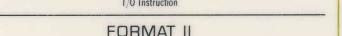


I/O Instruction

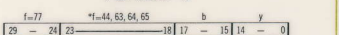
FORMAT II



General Instruction**



Special Instruction



Direct Addressing Instruction**

*f=Format II Function Code **Forced k=3

MEMORY ADDRESS ASSIGNMENT

OCTAL ADDRESS RANGE	USE
00000	Program Fault Interrupt Entrance Address
00001	Count-down Clock Interrupt Entrance Address
00002	Memory Protect Interrupt Entrance Address
00003	Input Power Failure Interrupt Entrance Address
00004	Power On Entrance Address
00005-00017	Unassigned
00020-00037	External Interrupt Entrance Addresses
00040-00057	Input Monitor Interrupt Entrance Addresses
00060-00077	Output Monitor Interrupt Entrance Addresses
00100-00117	Input Buffer Control Words
00120-00137	Output Buffer Control Words
00140-00157	External Function Buffer Control Words
00160	Real Time Clock
00161	Count-down Clock
00162-00177	Unassigned
00200-00217	ESI Input Buffer Termination Words
00220-00237	ESI Output Buffer Termination Words
00240-00257	ESI External Function Buffer Termination Words
00260-00477	Unassigned
00500-00517	External Function Buffer Monitor Interrupt Entrance Addresses
00520-00537	Interrupt Word Storage Addresses
00540-00577	Unassigned
00600-00617	Intercomputer Time-Out Entrance Address
00620 and over	Unassigned
40000-40777	NDRO Memory when enabled

UNIVAC 1830A COMPUTER (CP-901)

AN/ASQ-114(V)

CODE (Octal) f	INSTRUCTION	DESCRIPTION	Time μsec.
01	Right SHift-Q	Shift (Q) Right by Y	2-4-6
02	Right SHift-A	Shift (A) Right by Y	2-4-6
03	Right SHift-AQ	Shift (AQ) Right by Y	2-4-6
*04	COMpare-A,Q-AQ	Sense (j): A=A$_T$	4
05	Left SHift-Q	Shift (Q) Left by Y	2-4-6
06	Left SHift-A	Shift (A) Left by Y	2-4-6
07	Left SHift-AQ	Shift (AQ) Left by Y	2-4-6
10	ENTer-Q	Y → Q	2-4
10k0	CLear-Q	0 → Q	2
11	ENTer-A	Y → A	2-4
11k0	CLear-A	0 → A	2
12	ENTer-Bj	Y → Bj	2-4
12k0	CLear-Bj	0 → Bj	2
12j0	NO-OPeration	Enter B^0 with 0 (do nothing operation)	2-4
^13k0	EXternal-COMmand-C$_j$-W(Y)-MONITOR	(Y) → C$_j$, Y → 140+j ; ¿monitor interrupt	6 min.
^13k1	EXternal-COMmand-C$_j$-W (Y)-MONFORCE	Force (Y) → C$_j$, Y → 140+j ¿address is 500+j	6 min.
^13k2	EXternal-COMmand-C$_j$-W (Y)	Initiate external function Y → 140+j	6 min.
^13k3	EXternal-COMmand-C$_j$-W (Y)-FORCE	Force (Y) → C$_j$, Y → 140+j	6 min.
14k≠0	STOReQ	(Q) → Y	2-4
14k0	ComPlementQ	(Q)' → Q	2
15k≠4	STOReA	(A) → Y	2-4
15k4	ComPlementA	(A)' → A	2
16	STOReB	(Bj) → Y	2-4
^17k0	JumP-Y-C$_j$-COMACTIVE	Jump to Y if external function buffer active on C$_j$	4
^17k1	JumP-L(Y)-C$_j$-COMACTIVE	Jump to (Y)$_L$ if external function buffer active on C$_j$	4
^17k2	STOReC$_j$-W (Y)-FORCE	Input data word C$_j$ → Y, force IDA	6 min.
^17k3	STOReC$_j$-W (Y)	(00520+j) → Y	4-6
20	ADD-A	(A)+Y → A	2-4
21	SUBtract-A	(A)–Y → A	2-4
*22	MULtiply	(Q) Y → AQ	18
*23	DIVide	(AQ)/Y → Q; R → A	32
*23k7	SQuare RooT	$\sqrt{Q}$ → Q; residue → A	32
24	RePLace-A+Y	(A)+(Y) → Y & A	4-6
25	RePLace-A–Y	(A)–(Y) → Y & A	4-6
*26	ADD-Q	(Q)+Y → Q	2-4
*27	SUBtract-Q	(Q)–Y → Q	2-4
30	ENTer-Y–Q	Y+(Q) → A	2-4
31	ENTer-Y–Q	Y–(Q) → A	2-4
32	STOReA+Q	(A)+(Q) → A & Y	2-4
33	STOReA–Q	(A)–(Q) → A & Y	2-4
34	RePLace-Y+Q	(Y)+(Q) → Y & A	4-6
35	RePLace-Y–Q	(Y)–(Q) → Y & A	4-6
36	RePLace-Y+1	(Y)+1 → Y & A	4-6
37	RePLace-Y–1	(Y)–1 → Y & A	4-6
*40	ENTer-LP	L[Y(Q)] → A	2-4
41	ADD-LP	(A)+L[Y(Q)] → A	2-4
42	SUBtract-LP	(A)–L[Y(Q)] → A	2-4
43	COMpare-MASK	(A)–L[Y(Q)] sense (j), A+L(Y(Q)); (A)$_L$–(A)r	2-4
*44	RePLace-LP	L(Y)(Q)) → Y & A	4-6
45	RePLace-A+LP	(A)+L(Y)(Q)] → Y & A	4-6
46	RePLace-A–LP	(A)–L(Y)(Q)] → Y & A	4-6
47	STOReLP	L(A) (Q)] → Y	2-4
50	SELective-SET	Set A$_n$ for Y$_{n=1}$	2-4
51	SELective-ComPlement	Complement A$_n$ for Y$_{n=1}$	2-4
52	SELective-CLear	Clear A$_n$ for Y$_{n=1}$	2-4
53	SELective-SUbstitute	Y$_n$ → A$_n$ for Q$_{n=1}$	2-4

Y–The operand designator modified by B^k

x | Special j and k designators

Y–The operand; Y or (Y)

min–minimum

LP or L[ ]–Logical Product

CODE (Octal) f	INSTRUCTION	DESCRIPTION	Time μsec.
54	Replace SElective•SET	Set A_n for $(Y)_{n-1}$, (A) → Y	4-6
55	Replace SElective•ComPlement	Complement A_n for $(Y)_{n-1}$, (A) → Y	4-6
56	Replace SElective•CLear	Clear A_n for $(Y)_{n-1}$, (A) → Y	4-6
57	Replace SElective•SUbstitute	$(Y)_n \rightarrow A_n$ for Q_{n-1}, (A) → Y	4-6
‡*60	JumP (arithmetic)	Jump to Y if jump j condition satisfied	2-4
60j0	Remove Interrupt Lockout	Release master I/O interrupt lockout	2
‡60j1	Remove Interrupt Lockout, JumP	RIL same as above, jump to Y	2-4
161	JumP (maguaal)	Jump to Y if $\overline{j}$ condition is satisfied	2-4
^62	JumP•Y•C_I•ACTIVE INput buffer	Jump to Y if C_I input buffer active	4
^63	JumP•Y•C_O•ACTIVE OUTput buffer	Jump to Y if C_O output buffer active	4
‡*64	Return JumP (arithmetic)	j If j condition is satisfied, P → Y_L, Addressing mode → Y_U, jump to Y_{+1}	4-6
‡*65	Return JumP (manual)		4-6
^66k_0	TERMinate•C_I•INPUT	Terminate input buffer on C_I	2
^66k_1b0	Remove Interrupt Lockout•ALL	Release master I/O interrupt lockout	2
^66k_1b≠0	Set Interrupt Lockout•ALL	Set master I/O interrupt lockout	2
^66k_2b0	Remove Interrupt Lockout-EXternal•ALL	Release all external channel interrupt lockouts	2
^66k_2b≠0	Set Interrupt Lockout-EXternal•ALL	Set all external channel interrupt lockouts	2
^66k_3b0	Remove Interrupt Lockout-EXternal•C_I	Release external channel interrupt lockout on C_I	2
^66k_3b≠0	Set Interrupt Lockout-EXternal•C_I	Set external channel interrupt lockout on C_I	2
	Note: Master Clear sets all external channel interrupt lockouts and releases master I/O interrupt lockout.		
^6$\overline{7}$0	TERMinate•C_O•OUTPUT	Terminate output buffer on C_O	2
^6$\overline{7}$1	TERMinate•C_O•COMmand	Terminate external function buffer on C_O	2
^6$\overline{7}$2	TERMinate•ALL	Terminate all buffers	2
^70	RePeaT	Execute NI Y times	2-4
71	B SKIP•Bj	(Bj)≠$\underline{Y}_j$ (Bj)+1 → Bj and read NI; (Bj)−$\underline{Y}_j$ skip NI and clear Bj	4-6
72	B JumP•Bj	(Bj)≠0, (Bj)−1 → Bj and jump to $\underline{Y}$; (Bj)=0, read NI	4
^73	INput•C_I (without monitor mode)	Initiate input buffer on C_I, (Y) → 00100+$\overline{7}$	6 min.
^74$\underline{Y}$	OUTput•C_O (without monitor mode)	Initiate output buffer on C_O; (Y) → 00120+$\overline{7}$	6 min.
^74k2	EXternal-COMmand-MultiWord•C_O•W(Y)	Initiate EF buffer on C_O; (Y) → 00140+$\overline{7}$	6 min.
^75	INput•C_I (with MONITOR mode)	Initiate input buffer on C_I with monitor; (Y) → 00100+$\overline{7}$, monitor interrupt address is 00040+$\overline{7}$	6 min.
^76	OUTput•C_O (with MONITOR mode)	Initiate output buffer on C_O with monitor; (Y) → 00120+$\overline{7}$, monitor interrupt address is 00060+$\overline{7}$	6 min.
^76k2	EXternal-COMmand-MultiWord•C_O•W(Y)•MONITOR	Initiate EF buffer on C_O with monitor; monitor interrupt address is 00500+$\overline{7}$; (Y) → 00140+$\overline{7}$	6 min.
7711	Enter A With Y	Enter A using address in U; (Y) → A $Y=U_{16\,0}$	2-4
7715	Store A With Y	Store A using address in U; (A) → Y $Y=U_{16\,0}$	2-4
7744	Test and Set Flag	(Y) ≠ 0, read NI; (Y)=0, skip NI; always set Y=1s	4-6
^7750	Enter Absolute Page Register	$\underline{Y}_{5-0}$ → APR$\overline{j}$	4
^7754	Store Absolute Page Register	APR$\overline{j}$ → Y	2-4
7763	Load B and JumP	(P) → B7, jump to (Y)	4
7764	Direct Load B and JumP	(P) → B7, set direct addressing mode, jump to (Y)	4
7765	Page Load B and JumP	(P) → B7, set page addressing mode, jump to (Y)	4

Y—The operand designator modified by B
k, j Special j and k designators
$\underline{Y}$—The operand; Y or (Y)
‡See Address Mode Selection k designators

NI—Next Instruction
BCW—Buffer Control Word.
APR—Absolute page register
min—minimum

JP & RJP J-DESIGNATORS

j	JP f60	RJP f64	JP f61	RJP f65
0	(No Jump)*		Uncond. Jump	
1	Uncond. Jump*		KEY1	
2	QPOS		KEY2	
3	QNEG		KEY3	
4	AZERO		STOP	
5	ANOT zero		STOP5	
6	APOS		STOP6	
7	ANEG		STOP7	
$\overline{7}$	f62j		f63j	
0-17$_8$	C_I•ACTIVEIN		C_I•ACTIVEOUT	

*f60 Bootstrap or Spec L/O set, clear L/O;
Bootstrap or Spec L/O clear, clear I/O L/O and sets C_I EIE line after external interrupt.

NORMAL K-DESIGNATORS

k	READ Code	READ Origin	STORE Code	STORE Dest.	REPLACE Code	REPLACE Dest.
0	'blank'	U_L	Q	Q	—	—
1	L	M_L	L	M_L	L	M_L
2	U	M_U	U	M_U	U	M_U
3	W	M	W	M	W	M
4	X	XU_L	A	A	—	—
5	LX	XM_L	CPL	Cpl M_L	LX	XM_L
6	UX	XM_U	CPU	Cpl M_U	UX	XM_U
7	A	A	CPW	Cpl M	—	—

J-DESIGNATORS

j	COM•A, •Q, •AQ f04	DIV f23	SQRT f23 k7	ADD•Q, SUB•Q f26	ENT•LP, RPL•LP f27 f40 f44	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	YLESS : $\underline{Y}_{\leq}$(Q)	NOOver Flow	—	APOS	EVEN parity	BACK : Y of NE—Y−1
3	YMORE : $\underline{Y}_{>}$(Q)	Over Flow	—	ANEG	ODD parity	ADDB : Y of NE—Y+B$_8$
4	YIN : (Q)−$\underline{Y}$ and $\underline{Y}_{\neq}$(A)	AZERO	NOREM	QZERO	AZERO	R : Y of NE—Y|+B$_8$|√
5	YOUT : (Q)−Y or $\underline{Y}_{\leq}$(A)	ANOT zero	REM	QNOT zero	ANOT zero	ADVR : Y of NE—Y+|+B$_8$|√
6	YLESS : $\underline{Y}_{\leq}$(A)	skip	—	QPOS	APOS	BACKR : Y of NE—Y−|+B$_8$|√
7	YMORE : $\underline{Y}_{>}$(A)	no skip	—	QNEG	ANEG	ADDBR : Y of NE—Y+B$_8$|+B$_8$|√

√modifies Y address for the store portion by (B6) if repeated instruction is replace class.
NE—Next execution.

NORMAL J-DESIGNATORS

j	(Not applicable on * or ^)
	Skip Code
0	no skip
1	SKIP
2	QPOS
3	QNEG
4	AZERO
5	ANOT zero
6	APOS
7	ANEG

ADDRESS MODE SELECTION K-DESIGNATORS

k	f-Code	Operand	Remarks
0	f64 & f65	$\underline{Y}$—U_L	CA → Y_U, jump in CA
1	f64 & f65	$\underline{Y}$—M_L	CA → Y_U, jump in CA
2	f64 & f65	$\underline{Y}$—M_U	CA → Y_U, jump in CA
3	f60 & f61 f64 & f65	$\underline{Y}$—M	Set addressing mode contained in $(Y)_{15}$ and jump to $(Y)_L$ See note 3 CA → Y_U, jump in CA
4	f64 f65	$\underline{Y}$—U_L	Obtain instruction & operand in CA, set DA, P → Y_L, previous addressing mode → Y_U, jump in DA. Obtain instruction & operand in CA, set PA, P → Y_L, previous addressing mode → Y_U, jump in PA.
5	f64 f65	$\underline{Y}$—M_L	Obtain instruction & operand in CA, set DA, P → Y_L, previous addressing mode → Y_U, jump in DA. Obtain instruction & operand in CA, set PA, P → Y_L, previous addressing mode → Y_U, jump in PA.
6	f64 f65	$\underline{Y}$—M_U	Obtain instruction & operand in CA, set DA, P → Y_L, previous addressing mode → Y_U, jump in DA. Obtain instruction & operand in CA, set PA, P → Y_L, previous addressing mode → Y_U, jump in PA.
7	f64 & f65	$\underline{Y}$—A	CA → Y_U, jump in CA

Note: (1) CA—Current Addressing mode (DA or PA)
(2) DA—Direct Addressing mode; PA—Page Addressing mode
(3) Bit 2_{15}—mode definition bit; 0—DA, 1—PA

LEGEND

M—Memory word (30 bits)
M_L—Lower half memory word
M_U—Upper half memory word
X—Sign bit extended
Cpl—Complement
$C\overline{j}$—Channel $\overline{j}$
A—A-register
Q—Q-register