

UNIVAC CP-890 COMPUTER

REPERTOIRE OF INSTRUCTIONS

# OCTAL	INSTRUCTION	DESCRIPTION	** TIME USEC
01	RIGHT SHIFT .Q	SHIFT (Q) RIGHT BY Y	1.8
02	RIGHT SHIFT .A	SHIFT (A) RIGHT BY Y	1.8
03	RIGHT SHIFT .AQ	SHIFT (AQ) RIGHT BY Y	1.8
04	COMPARE .A, .Q, .AQ	COMPARE Y WITH A, Q or A & Q; A1=A1	1.8
05	LEFT SHIFT .Q	SHIFT Q LEFT BY Y	1.8
06	LEFT SHIFT .A	SHIFT A LEFT BY Y	1.8
07	LEFT SHIFT .AQ	SHIFT (AQ) LEFT BY Y	1.8
10	ENTER .Q	Y → Q	1.8
11	ENTER .A	Y → A	1.8
12	ENTER .B ⁿ	Y → B ⁿ	3.6*
12bo	NO OPERATION	0 → B ⁰ (DO NOTHING OPERATION)	3.6*
13ko	EF ON C [↑] W MONITOR	(Y) → C [↑]	3.6
13k1	EF ON C [↑] W MONITOR·FORCE	(Y) → C [↑]	3.6
13k2	EF ON C [↑]	(Y) → C [↑]	3.6
13k3	EF ON C [↑] W FORCE	(Y) → C [↑]	3.6
14	STORE .Q	(Q) → Y	1.8
14ko	COMPLEMENT .Q	(Q') → Q	1.8
15	STORE A	(A) → Y	1.8
15k4	COMPLEMENT A	(A') → A	1.8
16	STORE B ⁿ	(B) → Y	1.8
17k	JUMP .Y, C [↑] , COMACTIVE	JUMP TO Y IF EXTERNAL FUNCTION BUFFERACTIVE	3.6*
17k1	JUMP .L(Y), C [↑] COMACTIVE	JUMP TO L(Y) IF EXTERNAL FUNCTION BUFFERACTIVE	3.6*
17k2	STORE .C [↑] W(Y)·FORCE	FORCE C [↑] → (Y)	3.6
17k3	STORE .C [↑] W(Y)	(000520 + ↑) → (Y)	3.6
20	ADD .A	(A) + Y → A	1.8
21	SUBTRACT .A	(A) - Y → A	1.8
22	MULTIPLY	(Q) Y → AQ	7.2*
23	DIVIDE	(AQ) / Y → Q ; R → A _f	2.6*
23k7	SQUARE ROOT	√(Q) → Q ; R → A	7.2*
24	REPLACE .A+Y	(A) + (Y) → Y & A	3.6
25	REPLACE .A-Y	(A) - (Y) → Y & A	3.6
26	ADD .Q	(Q) + Y → Q	1.8
27	SUBTRACT .Q	(Q) - Y → Q	1.8
30	ENTER .Y + Q	Y + Q → A	1.8
31	ENTER .Y - Q	Y - Q → A	1.8
32	STORE .A + Q	(A) + (Q) → Y & A	3.6
33	STORE .A - Q	(A) - (Q) → Y & A	3.6
34	REPLACE .Y + Q	(Y) + (Q) → Y & A	3.6
35	REPLACE .Y - Q	(Y) - (Q) → Y & A	3.6
36	REPLACE .Y + I	(Y) + I → Y & A	3.6
37	REPLACE .Y - I	(Y) - I → Y & A	3.6
40	ENTER .LP	L [Y (Q)] → A	1.8

* - EXECUTIVE INSTRUCTIONS

* - EXECUTION TIME IS CONSTANT
 ** - FOR NONOVERLAPPED ADD 1.8 USEC
 Y - THE OPERAND; Y OR (Y)

REPERTOIRE OF INSTRUCTIONS

f OCTAL	INSTRUCTION	DESCRIPTION	** TIME USEC
41	ADD.LP	$(A) + L [Y(Q)] \rightarrow A$	1.8
42	SUBTRACT.LP	$(A) - L [Y(Q)] \rightarrow A$	1.8
43	COMPARE.MASK	$(A) - L [Y(Q)]$ SENSE (j); $A + L [Y(Q)] A_i = A_i$	1.8
44	REPLACE.LP	$L(Y)(Q) \rightarrow Y \& A$	3.6
45	REPLACE.A + LP	$A + L(Y)(Q) \rightarrow Y \& A$	3.6
46	REPLACE.A - LP	$A - L(Y)(Q) \rightarrow Y \& A$	3.6
47	STORE .LP	$L(A)(Q) \rightarrow Y; (A)_j = (A)_j$	1.8
50	SELECTIVE.SET	SET $(A)_n$ FOR $\underline{Y}_n = 1$	1.8
51	SELECTIVE.COMPLEMENT	COMP. $(A)_n$ FOR $\underline{Y}_n = 1$	1.8
51k4	COMPLEMENT.A	IF Y IS 77777, $A \rightarrow \bar{A}$	1.8
52	SELECTIVE.CLEAR	CLEAR $(A)_n$ FOR $\underline{Y}_n = 1$	1.8
53	SELECTIVE.SUBSTITUTE	$\underline{Y}_n \rightarrow (A)$ FOR $(Q)_n = 1$	1.8
54	REPLACE SELECTIVE.SET	SET $(A)_n$ FOR $(Y)_n = 1 \rightarrow Y \& A$	3.6
55	REPLACE SELECTIVE.COMP.	COMPLEMENT $(A)_n$ FOR $(Y)_n = 1 \rightarrow Y \& A$	3.6
56	REPLACE SELECTIVE.CLEAR	CLEAR $(A)_n$ FOR $(Y)_n = 1 \rightarrow Y \& A$	3.6
57	REPLACE SELECTIVE.SUBSTITUTE	$(Y)_n \rightarrow (A)_n$ FOR $(Q)_n = 1 \rightarrow Y \& A$	3.6
60	JUMP (ARITHMETIC)	JUMP TO \underline{Y} IF J CONDITION SATISFIED	3.6
60jo	REMOVE INTERRUPT LOCKOUT	ENABLE ALL INT. NOT LOCKED OUT BY SIL-EX	3.6*
60j1	REMOVE INT. LOCKOUT JUMP.Y	ENABLE INT. AND JUMP TO \underline{Y}	3.6*
61	JUMP (MANUAL)	JUMP TO \underline{Y} IF J CONDITION SATISFIED	3.6*
62	JUMP ON C [↑] ACTIVE INPUT BUFFER	JUMP TO \underline{Y} IF C [↑] INPUT BUFFER ACTIVE	3.6*
63	JUMP ON C [↑] ACTIVE OUTPUT BUFFER	JUMP TO \underline{Y} IF C [↑] OUTPUT BUFFER ACTIVE	3.6*
64	RETURN JUMP (ARITHMETIC)	JUMP TO $\underline{Y} + 1$ AND (P) + 1 $\rightarrow \underline{Y}_L$ IF J	5.4*
65	RETURN JUMP (MANUAL)	CONDITION IS SATISFIED	5.4*
*66	TERMINATE.C [↑] INPUT	TERMINATE INPUT BUFFER ON C [↑]	1.8
*66r1	REMOVE INT. LOCKOUT.ALL	ENABLE ALL INT. NOT LOCKED OUT BY SIL-EX	1.8
*66r2	REMOVE INT. LOCKOUT-EXT.ALL	ENABLE EXTERNAL INT.; ALL CHANNELS	1.8
*66r3	REMOVE INT. LOCKOUT-EXT.C [↑]	ENABLE EXTERNAL INT. ON C [↑]	1.8
*66r1b1	SET INT. LOCKOUT . ALL	LOCKOUT ALL INT. ON ALL CHANNELS	1.8
*66r2b1	SET INT. LOCKOUT - EXT..ALL	LOCKOUT EXTERNAL INT.; ALL CHANNELS	1.8
*66r3b1	SET INT. LOCKOUT - EXT..C [↑]	LOCKOUT EXTERNAL INT. ON C [↑]	1.8
*67	TERMINATE.C [↑] OUTPUT	TERMINATE OUTPUT BUFFER ON C [↑]	1.8
*67r1	TERMINATE.C [↑] COMMAND	TERMINATE E F BUFFER ON C [↑]	1.8
*67r2	TERMINATE.ALL	TERMINATE ALL BUFFERS	1.8
70	REPEAT	EXECUTE NI \underline{Y} TIMES	3.6*
71	B SKIP.B ^j	$(B)^j = \underline{Y}$, SKIP NI & CLEAR $(B)^j$; $(B)^j \neq Y$	3.6*
72	B JUMP.B ^j	ADVANCE B ^j & READ NI	
73	INPUT.C [↑] (WITHOUT MONITOR)	$(B)^j = 0$, READ NI; $(B)^j \neq 0$, $(B)^j - 1 \rightarrow B^j$ & JUMP TO \underline{Y}	3.6
74	OUTPUT.C [↑] (WITHOUT MONITOR)	BUFFER IN ON C [↑] ; $\underline{Y} \rightarrow 000100 + \uparrow$	3.6
74r2	EXT. COMMAND-MULTI WORD. C [↑] .W(Y)	BUFFER OUT ON C [↑] ; $\underline{Y} \rightarrow 000120 + \uparrow$	3.6
75	INPUT.C [↑] (WITH MONITOR)	BUFFER COMMANDS OUT ON C [↑] ; $(Y) \rightarrow 000140 + \uparrow$	3.6
76	OUTPUT.C [↑] (WITH MONITOR)	BUFFER IN ON C [↑] (MON); $\underline{Y} \rightarrow 000100 + \uparrow$	3.6
76r2	EX-COM-MULTI WORK.C [↑] .W(Y).MONITOR	BUFFER OUT ON C [↑] (MON); $\underline{Y} \rightarrow 000120 + \uparrow$	3.6
* - EXECUTIVE INSTRUCTIONS		* - EXECUTION TIME IS CONSTANT	
		** - FOR NONOVERLAPPED ADD 1.8 USEC.	
		Y - THE OPERAND; Y OR (Y)	

REPERTOIRE OF INSTRUCTIONS

f OCTAL	INSTRUCTION	DESCRIPTION	** TIME USEC
7700	ENTER EXECUTIVE MODE	JUMP TO 000010 & SET CLASS III LOCKOUT	3.6
7701	EXIT EXECUTIVE MODE	Y P & SET 2 ⁹ OF STATUS REG.	3.6
7702	LOAD B AND JUMP	ADDRESS OF NI → B & JUMP TO Y	3.6
7703	TEST OVERFLOW DESIGNATOR	IF SET, JUMP TO Y; IF NOT SET, DO NI	3.6
7705	EXECUTE REMOTE	EXECUTE INSTRUCTION LOCATED AT Y	1.8
7706	PROBE ALARM	ALARM WILL LIGHT IF INST. IS NOT EXECUTED EVERY 10-40M.SEC. AFTER FIRST EXECUTION	1.8
7707	NORMALIZE AQ	SHIFT AQ LEFT UNTIL A ₂₈ ≠ A ₂₉ ; COUNT Y	3.6
*7710	ENTER BREAKPOINT REGISTER	Y(2 ¹⁹ -2 ⁰) → BREAKPOINT REGISTER	1.8
7711	DOUBLE LENGTH ENTER	(Y) → Q & (Y + 1) → A	3.6
*7712	ENTER DISPLAY REGISTER	Y → DISPLAY REGISTER	1.8
7713	ENTER INDIRECT ADDRESS DES.	Y (2 ⁶ -2 ⁰) → STATUS REG. (2 ⁶ -2 ⁰)	1.8
7714	STORE BREAKPOINT REG.	BREAKPOINT → Y (2 ¹⁹ -2 ⁰)	1.8
7715	DOUBLE LENGTH STORE	Q → Y & A → (Y + 1)	3.6
7716	STORE DISPLAY REGISTER	DISPLAY → Y	1.8
7720	FLOATING POINT ADD	A + (Y + 1) → A(MANTISSA) & Q + Y → Q (CHARAC.)	7.2*
7721	FLOATING POINT SUBTRACT	A - (Y + 1) → A (MANTISSA) & Q - Y → Q (CHARAC.)	7.2*
7722	FLOATING POINT MULTIPLY	Ax (Y + 1) → A (MANTISSA) & QxY → Q (CHARAC.)	10.8*
7723	FLOATING POINT DIVIDE	A/(Y + 1) → A (MANTISSA) & Q/Y → Q (CHARAC.)	16.2*
7724	DOUBLE LENGTH ADD	AQ + (Y + 1, Y) AQ	3.6
7725	DOUBLE LENGTH SUBTRACT	AQ - (Y + 1, Y) AQ	3.6
*7726	SET & ENABLE MONITOR CLOCK	Y (2 ¹⁹ -2 ⁰) → MON. CLOCK. ENABLE CLOCK	1.8
7727	TEST & SET FLAG	LOOKS AT Y _n & SETS Y _n IF NOT ALREADY SET	3.6
7740	SAMPLE KEYBOARD & KEYSET	(KEYBOARD, KEYSET, REMOTE LOCAL(SW.) → DISPLAY	1.8
7741	F/P ROUND, ENABLE/DISABLE	Ĥ = 0 ⇒ DISABLE ROUND; Ĥ = 1 ⇒ ENABLE ROUND	1.8
*7742	ENTER STATUS REGISTER	Y → STATUS REGISTER	1.8
*7743	ENTER MEMORY LOCKOUT REGISTER	Y (2 ¹⁵ -2 ⁰) → MLO	1.8
*7744	STORE MEMORY LOCKOUT REGISTER	MLO → Y (2 ¹⁵ -2 ⁰)	1.8
7745	STORE STATUS REGISTER	Ĥ = 0 ⇒ STATUS → Q; Ĥ = 1 ⇒ STATUS → Y	1.8
7760	ENTER SPECIAL REGISTER	Y (2 ⁶ -2 ⁰) SR ↑	1.8
*7761	ENTER INPUT EXTENSION REG.	Y (2 ² -2 ⁰) INPUT SR ON CHANN. ↑	1.8
*7762	ENTER OUTPUT EXTENSION REG.	Y (2 ² -2 ⁰) OUTPUT SR ON CHANN. ↓	1.8
*7763	ENTER EF EXTENSION REG.	Y (2 ² -2 ⁰) EF SR ON CHANNEL ↑	1.8
*7764	ENABLE INPUT CDM	ENABLE INPUT CHANNEL ↑ CDM	1.8
*7765	ENABLE OUTPUT CDM	ENABLE OUTPUT CHANNEL ↑ CDM	1.8
*7766	DISABLE INPUT CDM	DISABLE INPUT CHANNEL ↑ CDM	1.8
*7767	DISABLE OUTPUT CDM	DISABLE OUTPUT CHANNEL ↑ CDM	1.8
7770	STORE SPECIAL REGISTER	Ĥ = 0 ⇒ SR ↑ → Q; Ĥ = 1 ⇒ SR ↑ → Y	1.8
*7771	STORE INPUT EXTENSION REG.	SR ↑ (2 ² -2 ⁰) → Y (2 ² -2 ⁰) FOR INPUT CHANN. ↑	1.8
*7772	STORE OUTPUT EXTENSION REG.	SR ↑ → Y (2 ² -2 ⁰) FOR OUTPUT CHANNEL ↑	1.8
*7773	STORE EF EXTENSION REG.	SR ↑ → Y (2 ² -2 ⁰) FOR EF CHANNEL ↑	1.8
7774	DISABLE EXPANDED ADDR. MODE	DISABLE 18 BIT ADDRESSING MODE	1.8
7775	ENABLE EXPANDED ADDR. MODE	ENABLE 18 BIT ADDRESSING MODE	1.8
7776	ENABLE - DISABLE DISCRETE INT.	Ĥ = 0 ⇒ ENABLE DISCRETE INT.; Ĥ = 1 ⇒ DISABLE DISCRETE ↑	1.8

* - EXECUTIVE INSTRUCTIONS

* - EXECUTION TIME IS CONSTANT

** - FOR NON OVERLAPPED ADD 1.8 USEC.

Y - THE OPERAND; Y OR (Y)

NORMAL
K - DESIGNATORS

K	READ	STORE	REPLACE	
	ORIGIN	DEST.	ORIGIN	DEST.
0	U _L	Q	NOT	USED
1	M _L	M _L	M _L	M _L
2	M _u	M _u	M _u	M _u
3	M	M	M	M
4	XU _L	A	NOT	USED
5	XM _L	COMP→M _L	XM _L	M _L
6	XM _u	COMP→M _u	XM _u	M _u
7	A	COMP→M	NOT	USED

NORMAL
j DESIGNATORS

j	ALL BUT ^ AND j
0	NO SKIP
1	SKIP
2	Q POS
3	Q NEG
4	A=0
5	A≠0
6	A POS
7	A NEG

LEGEND

M - WHOLE WORD
OF MEMORY
M_L - LOWER HALF
OF MEMORY
WORD
M_u - UPPER HALF
OF MEMORY
WORD
X - SIGN BIT
EXTENDED
COMP. - COMPLEMENT

SPECIAL j DESIGNATORS

j	f = 04	f = 22, 23	f = 23 K = 7	f = 26, 27
0	NO SKIP	NO SKIP	NO SKIP	NO SKIP
1	SKIP	SKIP	SKIP	SKIP
2	Y ≤ Q	NO OVERFLOW	REMAINDER	A POS
3	Y > Q	OVERFLOW	NO REMAINDER	A NEG
4	Q ≥ Y AND Y > A	A = 0	NOT USED	Q = +0
5	Y > Q OR Y ≤ A	A ≠ 0	NOT USED	Q ≠ 0
6	Y ≤ A	A POS	NOT USED	Q POS
7	Y > A	A NEG	NOT USED	Q NEG

SPECIAL j DESIGNATORS

j	f = 40, 44	f = 60, 64	f = 61, 65	f = 70
0	NO SKIP	NO JUMP	JUMP	NO MODIFICATION
1	SKIP	JUMP	KEY 1	Y OF NE = Y + 1
2	A _f = EVEN PARITY	Q POS	KEY 2	Y OF NE = Y - 1
3	A _f = ODD PARITY	Q NEG	KEY 3	Y OF NE = Y + B ⁶
4	A = +0	A = 0	STOP 4	**Y OF NE = Y + [B ⁶]
5	A ≠ 0	A ≠ 0	STOP 5	**Y OF NE = Y + 1 + [B ⁶]
6	A POS	A POS	STOP 6	**Y OF NE = Y - 1 + B ⁶
7	A NEG	A NEG	STOP 7	**Y OF NE = Y + B ⁶ + [B ⁶]

** IF NI IS RPL, B⁶ INCREMENTS FOR STORE PORTION ONLY.
NE - NEXT EXECUTION OF REPEATED INSTRUCTION.

PRIORITY

INTERRUPT PRIORITY	I/O REQUEST PRIORITY																																																											
<p>CLASS I - POWER</p> <p>POWER TOLERANCE ERROR (CANNOT BE LOCKED OUT)</p>	<p>1. RTC</p> <p>2. FORCED COMMAND FROM PROCESSOR</p>																																																											
<p>CLASS II - HARDWARE</p> <p>1. FIRST COME FIRST SERVE PRIORITY</p> <p>U1 PARITY ERROR</p> <p>ZP PARITY ERROR</p> <p>MEMORY RESUME ERROR</p> <p>I/O RESUME ERROR</p> <p>I/O MEMORY RESUME ERROR</p> <p>2. PROGRAM FAULT</p> <p>JUMP TO 000000 IF AUTO RECOVERY OFF</p> <p>JUMP TO LOOK IF AUTO RECOVERY ON</p>	<p>3. EIR</p> <p>4. EFR</p> <p>5. ODR</p> <p>6. IDR</p> <p>7. NORMAL COMMAND FROM PROCESSOR</p>																																																											
<p>CLASS III - PROGRAM</p> <p>1. BREAKPOINT MATCH</p> <p>2. RETURN TO EXECUTIVE PROGRAM</p> <p>3. EXECUTIVE INSTRUCTION ERROR</p> <p>4. WRITE LOCKOUT ERROR</p> <p>5. READ LOCKOUT ERROR</p> <p>6. CHARACTERISTIC OVERFLOW</p> <p>7. CHARACTERISTIC UNDERFLOW</p> <p>8. FLOATING POINT DIVIDE ERROR</p> <p>9. MONITOR</p>	<p>TRANSLATOR</p> <table border="1" data-bbox="564 476 896 579"> <tr> <th colspan="2">CODE</th> <th colspan="4">CHANNEL</th> </tr> <tr> <td>6</td> <td>5</td> <td>4</td> <td>3</td> <td>2</td> <td>1</td> <td>0</td> </tr> <tr> <td>2</td> <td>2</td> <td>2</td> <td>2</td> <td>2</td> <td>2</td> <td>2</td> </tr> </table> <p>↓ ↓ ↓ ↓ ↓ ↓</p> <table border="1" data-bbox="595 656 896 772"> <tr> <td>5</td> <td>4</td> <td>3</td> <td>2</td> <td>1</td> <td>0</td> </tr> <tr> <td>2</td> <td>2</td> <td>2</td> <td>2</td> <td>2</td> <td>2</td> </tr> <tr> <th colspan="2">CODE</th> <th colspan="4">CHANNEL</th> </tr> <tr> <th colspan="6">SI REGISTER</th> </tr> </table> <table border="1" data-bbox="585 772 802 901"> <tr> <th>PRIORITY</th> <th>5</th> <th>4</th> </tr> <tr> <td>EI</td> <td>0</td> <td>1</td> </tr> <tr> <td>EF</td> <td>0</td> <td>0</td> </tr> <tr> <td>OD</td> <td>1</td> <td>1</td> </tr> <tr> <td>ID</td> <td>1</td> <td>0</td> </tr> </table>	CODE		CHANNEL				6	5	4	3	2	1	0	2	2	2	2	2	2	2	5	4	3	2	1	0	2	2	2	2	2	2	CODE		CHANNEL				SI REGISTER						PRIORITY	5	4	EI	0	1	EF	0	0	OD	1	1	ID	1	0
CODE		CHANNEL																																																										
6	5	4	3	2	1	0																																																						
2	2	2	2	2	2	2																																																						
5	4	3	2	1	0																																																							
2	2	2	2	2	2																																																							
CODE		CHANNEL																																																										
SI REGISTER																																																												
PRIORITY	5	4																																																										
EI	0	1																																																										
EF	0	0																																																										
OD	1	1																																																										
ID	1	0																																																										
<p>CLASS IV - I/O</p> <p>1. KEYBOARD INTERRUPT</p> <p>2. EXTERNAL DISCRETE #1</p> <p>3. EXTERNAL DISCRETE #2</p> <p>4. EXTERNAL DISCRETE #3</p> <p>5. EXTERNAL DISCRETE #4</p> <p>6. EXTERNAL DISCRETE #5</p> <p>7. INTERCOMPUTER TIME OUT</p> <p>8. EI INT.</p> <p>9. EF INT. (MONITOR)</p> <p>10. OD INT. (MONITOR)</p> <p>11. ID INT. (MONITOR)</p>	<p>EXPANDED ADDRESSING</p> <p>"S" INTERPRETED FOR:</p> <p>f=01-12, 20-31, 34-46, 50-57 & K#0, 4, 7</p> <p>f=14-16, 32-33, 47 & K#0, 4</p> <p>f=13, 17, 60-65, 72</p> <p>f=73-76, 7742-7776 & K#0</p> <p>f=7700-7737</p>																																																											
<p>INTERRUPT LOCKOUTS ARE RELEASED IN ORDER OF PRIORITY BY EXECUTING Q 60, j=0, 1 INSTRUCTIONS SEQUENTIALLY</p>																																																												

WORD FORMAT

FORMAT I GENERAL					
f	j	k	b	14 13 12	← 0
				s	y
29 ↔ 24	23 ↔ 21	20 ↔ 18	17 ↔ 15		y
				14	← 0

FORMAT I I/O					
f	\hat{j}	\hat{k}	b	14 13 12	← 0
				s	y
29 ↔ 24	23 ↔ 20	19 ↔ 18	17 ↔ 15		y
				14	← 0

FORMAT II f = 7700-7737					
f = 77	f = XX _B	b		14 13 12	← 0
				s	y
29 ↔ 24	23 ↔ 18	17 ↔ 15			y
				14	← 0

FORMAT II f = 7740-7777					
f = 77	f = XX _B	b	14 13 12 9 8 7	← 0	
			s	\hat{j}	\hat{k} y
29 ↔ 24	23 ↔ 18	17 ↔ 15	14 13	/ / / / /	7 ← 0
					y

MEMORY BUFFER					
	29	← TA	→ 15	14	← CA → 0
35 34 33 32	←	→ 18	17 ↔ 15	14	← 0
1 2 3		TA	SR I/O		CA
ZI					

KEYSET KEYBOARD DISPLAY					
29	←	→ 14	13	← 6	5 4 3 ↔ 0
16	←	KEYSET	→ 1	8 ← KEYSET → 1	NOT USED REMOTE LOCAL KEYBOARD

BREAKPOINT		
19	18	17 ← → 0
OPERAND	INSTRUCTION	ADDRESS



INTERPRETED AS 0'S FOR MEMORY ADDRESSING.

1 - NOT USED

2 - CDM

3 - MONITOR INTERRUPT

MEMORY ADDRESS ALLOCATIONS

NO. OF BITS	ADDRESS (OCTAL)	FUNCTION	TYPE
32	000000	PROGRAM FAULT INTERRUPT ENTRANCE	CORE
32	000001	POWER RESTART INTERRUPT ENTRANCE	CORE
32	000002	POWER TOLERANCE INTERRUPT ENTRANCE	CORE
32	000003	RESERVED	CORE
32	000004	BREAKPOINT INTERRUPT ENTRANCE	CORE
32	000005-000007	RESERVED	CORE
32	000010	EXECUTIVE ENTRANCE	CORE
32	000011	EXECUTIVE INST. ERROR INTERRUPT ENTRANCE	CORE
32	000012	WRITE LOCKOUT INTERRUPT ENTRANCE	CORE
32	000013	READ LOCKOUT INTERRUPT ENTRANCE	CORE
32	000014	CHARACTERISTIC OVERFLOW INTERRUPT ENTRANCE	CORE
32	000015	CHARACTERISTIC UNDERFLOW INTERRUPT ENTRANCE	CORE
32	000016	FLOATING POINT DIVIDE ERROR INTERRUPT ENTRANCE	CORE
32	000017	MONITOR CLOCK INTERRUPT ENTRANCE	CORE
32	000020-000037	EXTERNAL INTERRUPT ENTRANCE	CORE
32	000040-000057	INPUT MONITOR INTERRUPT ENTRANCE	CORE
32	000060-000077	OUTPUT MONITOR INTERRUPT ENTRANCE	CORE
36	000100-000117	INPUT BUFFER CONTROL WORDS	FF
36	000120-000137	OUTPUT BUFFER CONTROL WORDS	FF
36	000140-000157	EXTERNAL FUNCTION BUFFER CONTROL WORDS	FF
30	000160	REAL TIME CLOCK REGISTER	FF
18	000161-000167	B REGISTERS	FF
36	000170-000177	UNASSIGNED	FF
32	000200-000217	ESI INPUT BUFFER TERMINATE OR CDM RELOAD	CORE
32	000220-000237	ESI OUTPUT BUFFER TERMINATE OR CDM RELOAD	CORE
32	000240-000257	ESI EXTERNAL FUNCTION BUFFER TERMINATE	CORE
32	000260	KEYBOARD INTERRUPT ENTRANCE	CORE
32	000261	EXTERNAL DISCRETE NO. 1 INTERRUPT ENTRANCE	CORE
32	000262	EXTERNAL DISCRETE NO. 2 INTERRUPT ENTRANCE	CORE
32	000263	EXTERNAL DISCRETE NO. 3 INTERRUPT ENTRANCE	CORE
32	000264	EXTERNAL DISCRETE NO. 4 INTERRUPT ENTRANCE	CORE
32	000265	EXTERNAL DISCRETE NO. 5 INTERRUPT ENTRANCE	CORE
32	000266-000277	UNASSIGNED	CORE
32	000300-000377	NDRC HARDWARE ANALYSIS ROUTINE	NDRO
32	000400-000477	RESERVED	CORE
32	000500-000517	EXTERNAL FUNCTION MONITOR INTERRUPT ENTRANCE	CORE
32	000520-000537	EXTERNAL INTERRUPT CODE STORAGE	CORE
32	000540-000577	UNASSIGNED	CORE
32	000600-000617	INTERCOMPUTER TIME-OUT INTERRUPT ENTRANCE	CORE
32	000620-000677	RESERVED	CORE
32	000700-000777	NDRO LOAD PROGRAMS LOAD I = 000700 LOAD II = 000740	NDRO NDRO
32	001000-077777	UNASSIGNED	CORE
32	100000-777777	EMU	CORE

INDIRECT ADDRESSING

$Y + (b^n) = \text{FINAL ADDRESS}$

$(b^n) = \text{CONTENTS OF } b^n$

$b^n = \text{B BOX SPECIFIED BY INDIRECT DESIGNATOR}$

INSTRUCTIONS NOT INDIRECTLY ADDRESSED: 13, 17 $\hat{k} = 0, 1, 60-67, 72,$
 7700-7707, 7713, 7740, 7742, 7745, 7764-7767, 7774-7776,
 STORE INSTRUCTIONS WITH $k = 0, 4,$
 READ INSTRUCTIONS WITH $k = 0, 4, 7$

MEMORY LOCKOUT

BIT POSITION		MEMORY ADDRESSES LOCKED OUT IN BANK SPECIFIED BY \hat{j}
READ	WRITE	
8	0	00000 - 03777
9	1	04000 - 07777
10	2	10000 - 13777
11	3	14000 - 17777
12	4	20000 - 23777
13	5	24000 - 27777
14	6	30000 - 33777
15	7	34000 - 37777