SPERRY UNIVAC
DEFENSE SYSTEMS

AN/UYK-7 COMPUTER
REPERTOIRE OF INSTRUCTIONS

CENTRAL PROCESSOR COMMANDS

Code	Mnemonic	NAME	DESCRIPTION	F	CA	R	UF	Time‡ µS
00	ILLEGAL							
01 0	OR	Inclusive OR (Selective Set A)	$(Y) \oplus (A_a) \rightarrow A_a$	II	Y	Y	2	1.5
01 1	SC	Selective Clear A	$(A_a) \odot (Y)' \rightarrow A_a$	II	Y	Y	2	1.5
01 2	MS	Selective Substitute	$(Y)_n \rightarrow (A_{a+1})_n$ for all $(A_a)_n = 1$; $(A_a)_i = (A_a)_f$	II	Y	Y	2	1.5
01 3	XOR	Exclusive OR (Sel. Comp. A)	$(Y) \overline{\oplus} (A_a) \rightarrow A_a$; $(A_a)_n' \rightarrow (A_a)$ for $(Y)_n = 1$	II	Y	Y	2	1.5
01 4	ALP	Add Logical Product	$(A_{a+1}) + (Y) \odot (A_a) \rightarrow A_{a+1}$; $(A_a)_i = (A_a)_f$	II	Y	Y	2	1.5
01 5	LLP	Load Logical Product	$(Y) \odot (A_a) \rightarrow A_a$	II	Y	Y	2	1.5
01 6	NLP	Subtract Logical Product	$(A_{a+1}) - (Y) \odot (A_a) \rightarrow A_{a+1}$; $(A_a)_i = (A_a)_f$	II	Y	Y	2	1.5
01 7	LLPN	Load Logical Product Next	$(Y) \odot (A_a) \rightarrow A_{a+1}$; $(A_a)_i = (A_a)_f$	II	Y	Y	2	1.5
02 0	CNT	Count Ones	No. of Bits Set in $(Y) \rightarrow A_a$	II	Y	Y	2	7.5†
02 1	ILLEGAL							
02 2	XR	Execute Remote	$(Y) \rightarrow U$. Execute $(Y)_U$ only of two half words.	II	N	N	8	1.5
02 3	XRL	Execute Remote Lower	$(Y)_L \rightarrow U$	II	N	N	8	1.5
02 4	SLP	Store Logical Product	$(A_{a+1}) \odot (A_a) \rightarrow Y$; $(A_a)_i = (A_a)_f$; $(A_{a+1})_i = (A_{a+1})_f$	II	Y	Y	2	1.5
02 5	SSUM	Store Sum	$(A_a) + (A_{a+1}) \rightarrow A_{a+1}$&Y; $(A_a)_i = (A_a)_f$	II	Y	Y	2	2.0
02 6	SDIF	Store Difference	$(A_{a+1}) - (A_a) \rightarrow A_{a+1}$&Y; $(A_a)_i = (A_a)_f$	II	Y	Y	2	2.0
02 7††	DS	Double Store A	$(A_{a+1}, A_a) \rightarrow Y+1, Y$	II	N	N	2	3.0
03 0	ROR	Replace Inclusive OR	$(Y) \oplus (A_a) \rightarrow A_a$ & Y	II	Y	Y	2	2.5
03 1	RSC	Replace Selective Clear	$(A_a) \odot (Y)' \rightarrow A_a$ & Y	II	Y	Y	2	2.5
03 2	RMS	Replace Selective Substitute	$(Y)_n \rightarrow (A_{a+1})_n$ for all $(A_a)_n = 1$; Then $(A_{a+1}) \rightarrow Y$; $(A_a)_i = (A_a)_f$	II	Y	Y	2	2.5
03 3	RXOR	Replace Exclusive OR	$(Y) \overline{\oplus} (A_a) \rightarrow A_a$ & Y; $(A_a)_n' \rightarrow A_a$ & Y for $Y_n = 1$	II	N	Y	2	2.5
03 4	RALP	Replace A + Logical Product	$(A_{a+1}) + (Y) \odot (A_a) \rightarrow A_{a+1}$ & Y; $(A_a)_i = (A_a)_f$	II	Y	Y	2	2.5
03 5	RLP	Replace Logical Product	$(Y) \odot (A_a) \rightarrow Y$ & A_{a+1}; $(A_a)_i = (A_a)_f$	II	Y	Y	2	2.5
03 6	RNLP	Replace A — Logical Product	$(A_{a+1}) - (Y) \odot (A_a) \rightarrow A_{a+1}$ & Y; $(A_a)_i = (A_a)_f$	II	Y	Y	2	2.5
03 7	TSF	Test and Set Flag	If $(Y)_{31} = 0$, CD Set EQUAL. $1 \rightarrow Y_{31}$ If $(Y)_{31} = 1$, CD Set UNEQUAL. This instruction cannot use indirect addressing.	II	N	Y	8	2.5
04 X	ILLEGAL							
05 0††	DL	Double Load A	$(Y+1, Y) \rightarrow A_{a+1}, A_a$	II	N	N	2	3.0
05 1††	DA	Double Add A	$(A_{a+1}, A_a) + (Y+1, Y) \rightarrow A_{a+1}, A_a$	II	N	N	2	3.0
05 2††	DAN	Double Subtract A	$(A_{a+1}, A_a) - (Y+1, Y) \rightarrow A_{a+1}, A_a$	II	N	N	2	3.0
05 3††	DC	Double Compare	Compare (A_{a+1}, A_a) to $(Y+1, Y)$, Set CD	II	N	N	2	3.0
05 4	LBMP	Load Base and Memory Protection	$(Y)_{17\text{-}0} \rightarrow S_a$; $(Y+1)_{20\text{-}0} \rightarrow SPR_a$; $Y \rightarrow SIR_a$ Privileged if: ASR bit 8 = 0, s≠7 or a = 7. Illegal if y + (B_b) = odd.	II	N	N	2	5.75
05 5	ILLEGAL							
05 6	ILLEGAL							
05 7	ILLEGAL							
06 0††	FA	Floating-point Add	Shift (A_{a+1}) or $(Y+1)$ Right such that $(A_a) = (Y)$ $(A_{a+1}) + (Y+1) \rightarrow A_{a+1}$; Normalize	II	N	N	2	6.25†
06 1††	FAN	Floating-point Subtract	Shift (A_{a+1}) or $(Y+1)$ Right such that $(A_a) = (Y)$ $(A_{a+1}) - (Y+1) \rightarrow A_{a+1}$; Normalize	II	N	N	2	6.25†
06 2††	FM	Floating-point Multiply	$(A_a) + (Y) \rightarrow (A_a)$ $(A_{a+1}) \bullet (Y+1) \rightarrow A_{a+1}$; Normalize	II	N	N	2	10.0†
06 3††	FD	Floating-point Divide	$(A_a) - (Y) \rightarrow (A_a)$ $(A_{a+1}) \div (Y+1) \rightarrow A_{a+1}$; Normalize	II	N	N	2	17.0†
06 4††	FAR	Floating-point Add with Round	Same as FA with (A_{a+1}) rounded	II	N	N	2	6.25†
06 5††	FANR	Floating-point Subtract w/Rd.	Same as FAN with (A_{a+1}) rounded	II	N	N	2	6.25†
06 6††	FMR	Floating-point Multiply w/Rd.	Same as FM with (A_{a+1}) rounded	II	N	N	2	10.0†
06 7††	FDR	Floating-point Divide w/Rd.	Same as FD with (A_{a+1}) rounded	II	N	N	2	17.0
07 0 a = 0	XS	Enter Executive State	sy + $(B_b) \rightarrow$ CMR 156; Enter class IV(Executive)	II	N	N	11	4.0
07 0* a = 1	IPI	Interprocessor Interrupt	Send Class II interrupt to processors n (0-7) IF bit n of sy + (B_b) = 1. Prevent self-interrupt if sy + (B_b) bit 15 = 1.	II	N	N	11	4.0
07 1**	AEI	Allow Enable Interrupt	Allow Monitor interrupts from IOC a on Channels n; IF bit n of sy + (B_b) = 1: Bit 25 is ignored	II	N	N	6	2.0
07 2**	PEI	Prevent Enable Interrupt	Prevent Monitor interrupts from IOC a on Channels n; IF bit n of sy + (B_b) = 1: Bit 25 is ignored	II	N	N	6	2.0
07 3**	LIM	Load IOC Monitor Clock	sy + $(B_b) \rightarrow$ IOC a MON CLK	II	N	N	6	3.0
07 4**	IO	Initiate I/O	Initiate IOC a at address Y	II	N	N	2	3.5
07 5*	IR	Interrupt Return	Return from highest State Specified by ASR bits 19-16.	II	N	N	9	3.0
07 6	RP	Repeat	Repeat N.I.B_7 Times; sy sign extended of Repeat added to B_b of N.I. after each cycle. See Repeat Conditions Illegal if in N.I. i = 1 and c = 00.	II	N	N	6	1.5
07 7	ILLEGAL							
10	LA	Load A	$\underline{Y} \rightarrow A_a$	I	Y	Y	1	1.5
11††	LXB	Load A and Index B	$\underline{Y} \rightarrow A_a$;$(B_b + 1 \rightarrow B_b$. Illegal if i = 1 and cc = 00.	I	Y	N	1	1.5
12	LDIF	Load Difference	$\underline{Y} - (A_a) \rightarrow A_{a+1}$;$(A_a)_i = (A_a)_f$	I	Y	Y	1	1.5
13	ANA	Subtract A	$(A_a) - \underline{Y} \rightarrow A_a$	I	Y	Y	1	1.5
14	AA	Add A	$(A_a) + \underline{Y} \rightarrow A_a$	I	Y	Y	1	1.5
15	LSUM	Load Sum	$(A_a) + \underline{Y} \rightarrow A_{a+1}$; $(A_a)_i = (A_a)_f$	I	Y	Y	1	1.5
16	LNA	Load Negative	$\underline{Y}' \rightarrow A_a$	I	Y	Y	1	1.5
17	LM	Load Magnitude	$\lvert\underline{Y}\rvert \rightarrow A_a$	I	Y	Y	1	1.5
20	LB	Load B	$\underline{Y} \rightarrow B_a$	I	Y	Y	1	2.0
21	AB	Add B	$(B_a) + \underline{Y} \rightarrow B_a$; B_a zero extended	I	Y	Y	1	2.0
22	ANB	Subtract B	$(B_a) - \underline{Y} \rightarrow B_a$; B_a zero extended	I	Y	Y	1	2.0
23	SB	Store B	$(B_a) \rightarrow \underline{Y}$	I	Y	Y	1	1.5
24	SA	Store A	$(A_a) \rightarrow \underline{Y}$	I	Y	Y	1	1.5
25††	SXB	Store A and Index B	$(A_a) \rightarrow \underline{Y}$;$(B_b) + 1 \rightarrow B_b$. Illegal if i = 1 and cc = 00.	I	Y	N	1	1.5
26	SNA	Store Negative	$(A_a)' \rightarrow \underline{Y}$	I	N	Y	1	1.5
27	SM	Store Magnitude	$\lvert(A_a)\rvert \rightarrow \underline{Y}$	I	N	Y	1	1.5
30	ILLEGAL							

*Privileged **CPU→IOC Instr. — Privileged ✓Privileged when ak = 2X, 6X, or 7X or Repeated.
†Execution time independent of overlap operation ††Privileged if i = 1 & $(SPR)_s$ bit 16 = 1.
‡Times shown assume 1.5µs memory with operands not in same bank as instructions (overlapped).

Code	Mnemonic	NAME	DESCRIPTION	F	CA	R	UF	Time‡ µS
31	ILLEGAL							
32	BZ	Clear Bit	$0 \rightarrow Y_{ak}$	I	N	Y	3	2.5
33	BS	Set Bit	$1 \rightarrow Y_{ak}$	I	N	Y	3	2.5
34	RA	Replace Add	$(A_a) + \underline{Y} \rightarrow A_{a+1}$ & $\underline{Y}$; $(A_a)_i = (A_a)_f$	I	Y	Y	1	2.5
35	RI	Replace Increment	$\underline{Y} + 1 \rightarrow \overline{A}_a$ & $\underline{Y}$	I	Y	Y	1	2.5
36	RAN	Replace Subtract	$\underline{Y} - (A_a) \rightarrow A_{a+1}$ & $\underline{Y}$; $(A_a)_i = (A_a)_t$	I	Y	Y	1	2.5
37	RD	Replace Decrement	$\underline{Y} - 1 \rightarrow A_a$ & $\underline{Y}$	I	Y	Y	1	2.5
40	M	Multiply A	$(A_a) \bullet \underline{Y} \rightarrow A_{a+1}, A_a$	I	Y	Y	1	7.5†
41	D	Divide A	$(A_{a+1}, A_a) \div \underline{Y} \rightarrow A_a$; remainder $\rightarrow A_{a+1}$	I	Y	Y	1	14.5†
42	BC	Compare Bit to Zero	If $(Y)_{ak} = 0$, CD Set EQUAL If $(Y)_{ak} = 1$, CD Set UNEQUAL Bit 25 is ignored	I	N	Y	3	1.5
43	CXI	Compare Index Increment	If $(B_a) \geq \underline{Y}$, CD Set OUTSIDE, $0 \rightarrow B_a$ If $(B_a) < \underline{Y}$, CD Set WITHIN, $(B_a) + 1 \rightarrow B_a$	I	Y	N	1	2.0
44	C	Compare	Compare (A_a) to $\underline{Y}$, Set the CD	I	Y	Y	1	1.5
45	CL	Compare Limits	If $(A_{a+1}) > \underline{Y} \geq (A_a)$, Set CD WITHIN	I	Y	Y	1	1.5
46	CM	Compare Masked	Compare (A_{a+1}) to $(A_a) \odot \underline{Y}$, Set the CD	I	Y	Y	1	1.5
47	CG	Compare Gated	Compare $\lvert\underline{Y} - (A_a)\rvert$ to (A_{a+1}), Set the CD	I	Y	Y	1	1.5
50 0	JEP	Jump on Even Parity	If $(A_{a+1}) \odot (A_a)$ is Even Parity, jump to Y	III	N	N	1	2.0
50 1	JOP	Jump on Odd Parity	If $(A_{a+1}) \odot (A_a)$ is Odd Parity, jump to Y	III	N	N	1	2.0
50 2	DJZ	Jump Double Precision Zero	If $(A_{a+1}, A_a) = 0$, jump to Y	III	N	N	1	2.0
50 3	DJNZ	Jump Double Precision Not Zero	If $(A_{a+1}, A_a) \neq 0$, jump to Y	III	N	N	1	2.0
51 0	JP	Jump A Positive	If $(A_a) \geq 0$, jump to Y	III	N	N	1	1.5
51 1	JN	Jump A Negative	If $(A_a) < 0$, jump to Y	III	N	N	1	1.5
51 2	JZ	Jump A Zero	If $(A_a) = 0$, jump to Y	III	N	N	1	1.5
51 3	JNZ	Jump A Not Zero	If $(A_a) \neq 0$, jump to Y	III	N	N	1	1.5
52 0	LBJ	Load B and Jump	$(P) + 1 \rightarrow B_a$, jump to Y	III	N	N	1	1.8
52 1	JBNZ	Index Jump B	If $(B_a) \neq 0$, then $(B_a) - 1 \rightarrow B_a$, jump to Y	III	N	N	1	1.8
52 2	JS	Jump sy + B	Jump to sy + (B_b)	III	N	N	13	1.5
52 3	JL	Unconditional Jump Lower	Jump to the Lower of Y	III	N	N	12	1.5
53 0 a = 0	JNF	Jump on No Overflow	If OD is not Set, Jump to Y; Clear OD	III	N	N	12	1.5
53 0 a = 1	JOF	Jump on Overflow	If OD is Set, jump to Y; Clear OD	III	N	N	12	1.5
53 1 a = 0	JNE	Jump on Not Equal	If CD ≠ , jump to Y	III	N	N	12	1.5
53 1 a = 1	JE	Jump on Equal	If CD = , jump to Y	III	N	N	12	1.5
53 1 a = 2	JG	Jump on Greater Than	If CD >, jump to Y	III	N	N	12	1.5
53 1 a = 3	JGE	Jump on Greater Than or Equal	If CD ≥, jump to Y	III	N	N	12	1.5
53 1 a = 4	JLT	Jump on Less Than	If CD <, jump to Y	III	N	N	12	1.5
53 1 a = 5	JLE	Jump on Less Than or Equal	If CD ≤, jump to Y	III	N	N	12	1.5
53 1 a = 6	JNW	Jump Outside Limits	If CD Outside Limits, jump to Y	III	N	N	12	1.5
53 1 a = 7	JW	Jump Within Limits	If CD Within Limits, jump to Y	III	N	N	12	1.5
53 2	RJ	Return Jump a = o	$(P) + 1 \rightarrow Y$, jump to Y + 1	III	N	N	12	3.0
53 2	RJC	Return Jump a = 1, 2, 3	If switch a is Set, $(P) + 1 \rightarrow Y$, jump to Y + 1; otherwise N.I.	III	N	N	1	3.0
53 2*	RJSC	Return Jump a = 4, 5, 6, 7	If switch a is Set, Stop;$(P) + 1 \rightarrow Y$, jump to Y + 1 at restart	III	N	N	1	3.75
53 3	J	Manual Jump a = o	Jump to Y	III	N	N	12	1.5
53 3	JC	Manual Jump a = 1, 2, 3	If switch a is Set, jump to Y; otherwise N.I.	III	N	N	1	1.5
53 3*	JSC	Manual Jump a = 4, 5, 6, 7	If switch a is Set, Stop; Jump to Y at restart	III	N	N	1	2.25
54✓	LCT	Load CMR Task	$(Y) \rightarrow CMR_{ak}$	I	N	Y	3	1.5
55*	LCI	Load CMR Interrupt	$(Y) \rightarrow CMR_{ak+100}$	I	N	Y	3	1.5
56✓	SCT	Store CMR Task	$(CMR_{ak}) \rightarrow Y$	I	N	Y	3	1.5
57*	SCI	Store CMR Interrupt	$(CMR_{ak+100}) \rightarrow Y$	I	N	Y	3	1.5
60✓i = 0	HSCT	Store CMR in A	$(CMR_{af4}) \rightarrow A_b$ (load/store only bits 15–0 of B)	IV A	N	N	4	1.75
60*i = 1	HSCI	Store CMR in A	$(CMR_{af+100}) \rightarrow A_b$ (load/store only bits 15–0 of B)	IV A	N	N	4	1.75
61✓i = 0	HLCT	Load CMR in A	$(A_b) \rightarrow CMR_{af4}$ (load/store only bits 15–0 of B)	IV A	N	N	4	1.7
61*i = 1	HLCI	Load CMR from A	$(A_b) \rightarrow CMR_{af4+100}$ (load/store only bits 15–0 of B)	IV A	N	N	4	1.75
62	HLC	Shift Left Circularly	(A_a) Left Shifted End Around $\rightarrow A_a$	IV B	N	N	10	1.75
63	HDLC	Shift Left Circularly Double	(A_{a+1}, A_a) Left Shifted End Around $\rightarrow A_{a+1}, A_a$	IV B	N	N	10	1.75
64	HRZ	Shift Right Fill Zeros	(A_a) Right Shifted, Zero Fill $\rightarrow A_a$	IV B	N	N	10	1.75
65	HDRZ	Shift Right Double, Fill Zeros	(A_{a+1}, A_a) Right Shifted, Zero Fill $\rightarrow A_{a+1}, A_a$	IV B	N	N	10	1.75
66	HRS	Shift Right Fill Sign	(A_a) Right Shifted, Sign Fill $\rightarrow A_a$	IV B	N	N	10	1.75
67	HDRS	Shift Right Double, Fill Sign	(A_{a+1}, A_a) Right Shifted Sign Fill $\rightarrow A_{a+1}, A_a$	IV B	N	N	10	1.75
70 0	HSF	Scale Factor	Normalize (A_a) Shift Count $\rightarrow A_b$	IV A	N	N	5	2.25
70 1	HDSF	Double Scale Factor	Normalize (A_{a+1}, A_a) Shift Count $\rightarrow A_b$	IV A	N	N	5	2.25
70 2	HCP	Complement A	$(A_a)' \rightarrow A_a$	IV A	N	N	7	1.1
70 3	HDCP	Double Complement A	$(A_{a+1}, A_a)' \rightarrow A_{a+1}, A_a$	IV A	N	N	7	1.1
70 4	ILLEGAL							
70 5	ILLEGAL							
70 6	ILLEGAL							
70 7	ILLEGAL							
71 0	HOR	Logical Sum	$(A_a) \oplus (A_b) \rightarrow A_a$; $(A_b)_i = (A_b)_f$ if a ≠ b	IV A	N	N	5	1.0
71 1	HA	Sum	$(A_a) + (A_b) \rightarrow A_a$; $(A_b)_i = (A_b)_f$ if a ≠ b	IV A	N	N	5	1.0
71 2	HAN	Difference	$(A_a) - (A_b) \rightarrow A_a$; $(A_b)_i = (A_b)_f$ if a ≠ b	IV A	N	N	5	1.0
71 3	HXOR	Logical Difference	$(A_a) \overline{\oplus} (A_b) \rightarrow A_a$; $(A_b)_i = (A_b)_f$ if a ≠ b	IV A	N	N	5	1.0
71 5	HAND	AND	$(A_a) \odot (A_b) \rightarrow A_a$; $(A_b)_i = (A_b)_f$ if a ≠ b	IV A	N	N	5	1.0
71 6	ILLEGAL							
71 7	ILLEGAL							
72 X	ILLEGAL							
73 X	ILLEGAL							
74 0	HM	Multiply Register	$(A_a) \bullet (A_b) \rightarrow A_{a+1}, A_a$	IV A	N	N	5	7.75†
74 1	HD	Divide Register	$(A_{a+1}, A_a) \div (A_b) \rightarrow A_a$; Remainder $\rightarrow A_{a+1}$	IV A	N	N	5	15.0†
74 2	HRT	Square Root	$\sqrt{(A_{a+1}, A_a)} \rightarrow A_b$; Residue $\rightarrow A_{b+1}$	IV A	N	N	5	15.0†
74 3	HLB	Load B_a with B_b	$(B_b) \rightarrow B_a$	IV A	N	N	5	1.75
74 4	HC	Compare, Register	Compare (A_a) to (A_b), Set CD	IV A	N	N	5	1.1
74 5	HCL	Compare Limits, Register	If $(A_{a+1}) > (A_b) \geqslant (A_a)$, Set CD WITHIN	IV A	N	N	5	1.75
74 6	HCM	Compare Masked, Register	Compare $(A_{a+1}) \odot (A_a)$ to (A_b), Set the CD	IV A	N	N	5	1.1
74 7	HCB	Compare B_b with B_a	Compare (B_b) to (B_a), Set the CD	IV A	N	N	5	2.0
75 X	ILLEGAL							
76 X	ILLEGAL							
77 0**	HSIM	Store IOC Monitor Clock in A	$(IOC_a$ MON CLK$) \rightarrow A_b$	IV A	N	N	5	3.0
77 1	HSTC	Store Real-Time Clock in A	$(IOC_a$ RTC$) \rightarrow A_b$	IV A	N	N	5	3.5
77 2	ILLEGAL							
77 3	ILLEGAL							
77 4*	HPI	Prevent Class III Interrupts	Set Class III Interrupt Lockout in the ASR	IV A	N	N	9	2.25
77 5*	HAI	Allow Class III Interrupts	Clear Class III Interrupt Lockout in the ASR	IV A	N	N	9	2.25
77 6*i = 0	HALT	Stop Processor	Stop CPU (4-Stop); Continue at Restart	IV A	N	N	9	2.25
77 6*i = 1	HWFI	Wait for Interrupt	Cease Memory References until Interrupted	IV A	N	N	9	2.25
77 7	ILLEGAL							

*Privileged **CPU→IOC Instr. — Privileged ✓Privileged when ak = 2X, 6X, or 7X or Repeated.
†Execution time independent of overlap operation ††Privileged if i = 1 & $(SPR)_s$ bit 16 = 1.
‡Times shown assume 1.5µs memory with operands not in same bank as instructions (overlapped).

INSTRUCTION WORD FORMATS

Format I

31 26	25 23	22 20	19 17	16	15 13	12 0
f	a	k	b	i	s	y

Format II

31 26	25 23	22 20	19 17	16	15 13	12 0
f	a	f_2	b	i	s	y

Format III

31 26	25 23	22 21	20	19 17	16	15 13	12 0
f	a	f_3	z	b	i	s	y

Format IV A

31 26	25 23	22 20	19 17	16
15 10	9 7	6 4	3 1	0
f	a	f_4	b	i

Format IV B

31 26	25 23	22 16
15 10	9 7	6 0
f	a	m

f — Function Code
$f_2 f_3 f_4$ — Subfunction Codes
a — Accumulator Register
k — Operand Interpretation
b — Index Register
i — Indirect Bit
s — Base Register
m — Shift Designator

Bit 2^6	Function
0	Shift by count $2^5 - 2^0$
1	Shift by B_b if $2^5 = 0$
1	Shift by A_b if $2^5 = 1$
b is specified by bits $2^3 - 2^1$	

FORMAT I INSTRUCTION k—FIELD INTERPRETATION

k	Memory to Arithmetic (Read)	Arithmetic to Memory (Store)	
0	sy SE + $(B_b) \rightarrow A_{15\text{-}0}$SE	Not Used	
1	$(Y_{15\text{-}0}) \rightarrow A_{15\text{-}0}$SE	$(A_{15\text{-}0}) \rightarrow Y_{15\text{-}0}$;	$Y_{31\text{-}16}$ — UN
2	$(Y_{31\text{-}16}) \rightarrow A_{15\text{-}0}$SE	$(A_{15\text{-}0}) \rightarrow Y_{31\text{-}16}$;	$Y_{15\text{-}0}$ — UN
3	$(Y_{31\text{-}0}) \rightarrow A_{31\text{-}0}$	$(A_{31\text{-}0}) \rightarrow Y_{31\text{-}0}$	
4	$(Y_{7\text{-}0}) \rightarrow A_{7\text{-}0}$ ZE	$(A_{7\text{-}0}) \rightarrow Y_{7\text{-}0}$;	$Y_{31\text{-}8}$ — UN
5	$(Y_{15\text{-}8}) \rightarrow A_{7\text{-}0}$ ZE	$(A_{7\text{-}0}) \rightarrow Y_{15\text{-}8}$;	$Y_{31\text{-}16}$ — UN $Y_{7\text{-}0}$ — UN
6	$(Y_{23\text{-}16}) \rightarrow A_{7\text{-}0}$ ZE	$(A_{7\text{-}0}) \rightarrow Y_{23\text{-}16}$;	$Y_{31\text{-}24}$ — UN $Y_{15\text{-}0}$ — UN
7	$(Y_{31\text{-}24}) \rightarrow A_{7\text{-}0}$ ZE	$(A_{7\text{-}0}) \rightarrow Y_{31\text{-}24}$;	$Y_{23\text{-}0}$ — UN

k — Field Interpretation for Replace Instructions:
Read Cycle — Same as memory to arithmetic.
Store Cycle — Same as arithmetic to memory. For Repeat, with b of repeat instruction not zero, Y will be modified by S_6 and not S_s for store cycle.

SE — Sign Extended; ZE — Zero Extended; UN — Unchanged

CENTRAL PROCESSOR CONTROL MEMORY ADDRESS ASSIGNMENT

Task Mode

Address	Use	Bits
0-7	Accumulator (A) registers 0-7	32
10	Unassigned	19
11-17	Index (B) registers 1-7	19†
20-27	Base (S) registers 0-7**	18
30-57	Unassigned (not usable)	—
6x	Breakpoint register**	20
7x	Active status register**	23

Interrupt Mode

Address	Use	Bits
100-107	Accumulator (A) registers 0-7	32
110	CP monitor clock register	19*
111-117	Index (B) registers 1-7	19†
120-127	Base (S) registers 0-7	18
130-137	Unassigned (not usable)	—
140	ICW — Class I	20
141	DSW — Class I ASR storage	20
142	DSW — Class I interrupt status code	20
143	DSW — Class I P — storage	20
144	ICW — Class II	20
145	DSW — Class II ASR storage	20
146	DSW — Class II interrupt status code	20
147	DSW — Class II P — storage	20
150	ICW — Class III	20
151	DSW — Class III ASR storage	20
152	DSW — Class III interrupt status code	20
153	DSW — Class III P — storage	20
154	ICW — Class IV	20
155	DSW — Class IV ASR storage	20
156	DSW — Class IV interrupt status code	20
157	DSW — Class IV P — storage	20
160-167	Storage Protection Registers (SPR) 0-7	21
170-177	Segment Identification Registers (SIR) 0-7	21

*Clock is low order 16 bits
**Not Addressable in the Task Mode. (Privileged instruction error will occur)
†Lower 16 bits used for index and arithmetic functions. Upper three bits used **only** as a base-register designation.

LBMP (05 4) CONSIDERATIONS

1) The LBMP instruction is privileged when bit 8 of the ASR = 0, or if bit 8 of the ASR = 1 and [s ≠ 7 or a = 7)

2) All function codes except the 05 4 (LBMP) are privileged when bit 8 of the ASR = 1 and s = 7 in the instruction.

FLOATING POINT FORMAT (each word is one's complement)

±	30 0	Sign Fill ±	14 0
Mantissa in A_{a+1} or Y + 1		Characteristic (exponent) in A_a or Y	

DOUBLE PRECISION (DOUBLE LENGTH) FORMAT

±	30 Most Significant Half 0	31 Least Significant Half 0
Y + 1 or A_{a+1}		Y or A

ULTRA/32 PSEUDO INSTRUCTIONS				F	CA	R	UF	Time ‡ µS
10	ZA	Clear A	$0 \rightarrow A_a$	I	N	Y	7	1.5
20	ZB	Clear B	$0 \rightarrow B_a$	I	N	Y	7	2.0
20	NOOP	No Operation	$0 \rightarrow B_0$	I	N	Y	9	2.0
23	SZ	Store Zeros	$0 \rightarrow \underline{Y}$	I	Y	Y	12	1.5
74 3	HNO	Half Word No Operation	$(B_0) \rightarrow B_0$	IV A	N	N	9	1.75

ULTRA/32 FORMATING MNEMONICS

—	HK	Half Word Constant (Variable field becomes next halfword)	—	—	—	16	—
—	IW	Indirect Word (c = 10)	—	—	—	8	—
—	IWS	Indirect Word, Special Base (c = 00, $c_1 = 0$)	—	—	—	11	—
—	IWB	Indirect Word, Special Index (c = 00, $c_1 = 1$)	—	—	—	11	—
—	IWC	Indirect Word, Character (c = 01)	—	—	—	14	—
—	IWCI	Indirect Word, Character Increment (c = 11)	—	—	—	14	—
—	MP	Memory Protection (see SPR format)	—	—	—	15	—

ULTRA/32 CODING FORMATS (UF)

(An Asterisk (*) Preceding y Indicates Indirect Addressing)

No.	Variable Field	No.	Variable Field	No.	Variable Field	No.	Variable Field	No.	Variable Field	No.	Variable Field
1	a, y, k, b, s	4	af_4, b	7	a	10	a, m (shift by m)	11	sy, b	14	y, w, p, b, s
2	a, y, b, s	5	a, b	8	y, b, s		a, b, 1 (shift by B_b)	12	y, k, b, s	15	r, i, or, ow, ia, ir
3	ak, y, b, s	6	a, sy, b	9	None		a, b, 2 (shift by A_b)	13	sy, k, b	16	e

SYMBOL DEFINITIONS

CMR — Control Memory Register
F — Format
CA — Character Addressable
R — Repeatable
DSW — Designator Storage Word

UF — Ultra Format
$(A)_n$ — Contents of A, bit n
CD — Compare Designator
Y — Address formed by y + (B_b) + (S_s)
ICW — Initial Condition Word

$\underline{Y}$ — Operand (Y) (Whole word or partial word) or Y, depending on k
⊙ — Logical product (AND)
⊕ — Logical sum (Inclusive OR)
$\overline{\oplus}$ — Logical difference (Exclusive OR)

I/O CONTROLLER COMMANDS

(All Unused Function Codes are Illegal)

Code	Mnemonic	NAME	DESCRIPTION	UF**	Time µS
10	IB	Initiate Input Buffer on Cj	(y)→CMA* 0 + j; Activate Input	1	3.25
11	OB	Initiate Output Buffer on Cj	(y)→CMA* 20 + j; Activate Output	1	3.25
12	FB	Initiate External Function Buffer on Cj	(y)→CMA* 40 + j; Activate EF	1	3.25
13	XB	Initiate External Interrupt Buffer on Cj	(y)→CMA* 60 + j; Activate EI	1	3.25
14 k = 0	TIB†	Terminate Input Buffer on Cj	Terminate Input } m = 0 Suppress	2	3.0
14 k = 1	TOB†	Terminate Output Buffer on Cj	Terminate Output } Queued Interrupt;	2	3.0
14 k = 2	TFB†	Terminate External Function Buffer on Cj	Terminate EF } m = 1 Allow Queued	2	3.0
14 k = 3	TXB†	Terminate External Interrupt Buffer on Cj	Terminate EI } Interrupt	2	3.0
15 k = 0	IMIR	Set Input Monitor Interrupt Request on Cj	Set Input Monitor Interrupt on Chan j	3	2.5
15 k = 1	OMIR	Set Output Monitor Interrupt Request on Cj	Set Output Monitor Interrupt on Chan j	3	2.5
15 k = 2	FMIR	Set EF Monitor Interrupt Request on Cj	Set EF Monitor Interrupt on Chan j	3	2.5
15 k = 3	XMIR	Set EI Monitor Interrupt Request on Cj	Set EI Monitor Interrupt on Chan j	3	2.5
16 k = 0	AIC	Set Input Chain Active on Cj	y→Command Address Pointer Field	4	2.5
16 k = 1	AOC	Set Output Chain Active on Cj	(bits 55-38) of CMA* 20k + j;	4	2.5
16 k = 2	AFC	Set External Function Chain Active on Cj	Activate Chain	4	2.5
16 k = 3	AXC	Set External Interrupt Chain Active on Cj		4	2.5
17 m = 0	TBZ	Test Bit Zero	If $(y)_{kj} = 0$, SKIP; Else NI	7	4.0
17 m = 1	TBS	Test Bit Set	If $(y)_{kj} \neq 0$, SKIP; Else NI	7	4.0
20	JIO	Jump to y	y→Command Address Pointer or CAR‡	6	2.5
22	LICM	Load IOC Control Memory	(y)→IOC Control Memory Address kj	5	3.25
23	ILTC	Load Real-Time Clock	(y)→Real Time Clock	6	4.0
24	SICM	Store IOC Control Memory	$(\text{IOC Control Memory})_{kj} \rightarrow y$	5	2.75
25	IBS	Set Bit	$1 \rightarrow y_{kj}$	5	3.25
26	IBZ	Clear Bit	$0 \rightarrow y_{kj}$	5	3.25
27	ITSF	Test and Set Flag	$1 \rightarrow y_{31}$; If $(y)_{31}$ was Originally Cleared, Skip; Else NI	6	3.25

FORMATING MNEMONICS

Code	Mnemonic	NAME	DESCRIPTION	UF**	Time µS
—	BCW	Buffer Control Word		8	—
—	BCWE	Buffer Control Word ESI		9	—

‡ Command Address Register
* Control Memory Address

**ULTRA FORMAT

1—j, y, k, c, m	4—j, y, c	7—kj, y	(l = buffer length)
2—j, c, m	5—kj, y, c	8—y, l	
3—j, c	6—y, c	9—y, l, k	

k—DESIGNATOR DEFINITIONS

	k = 0	k = 1	k = 2	k = 3
f = 10, 11, 13	Suppress data	Pack Quarter word	Pack Half word	Whole word
f = 12	Force One Word (y) is EF	One Word Buffer (y) is EF	Multi Word Buffer	Not Used

† The terminate buffer commands terminate only active buffers. They have no effect on active chains. Terminating an active buffer also terminates the chain since the buffer never completed normally. To terminate an active chain, it is recommended that a JIO instruction with no chaining be initiated on the channel & function to be terminated (y may be any valid address). However, attempts to terminate a chain on a channel and function with an active buffer will result in the CAP being overlayed but no change to the chain bit in IOCM. In this case, the buffer will complete normally and chaining will commence with the JIO instruction which then terminates the chain.

Note: Clearing the IOC enables all monitor interrupts to all CPU's (i.e., all bits set in all ILR's) and clears all requests.

IOC COMMAND WORD FORMAT

31 26	25 24	23 20	19	18	17 0
Function Code f	Partial Word Desig. k	Channel Number (0-17) j	Monitor Flag m	Chain Flag c	Operand Address y

NORMAL BUFFER CONTROL WORD FORMAT

31 18	17 0
Final Address Compare Bits	Initial Address

IOC CONTROL MEMORY WORD FORMAT

55 38	37 36	35	34	33 32	31 18	17 0
Command Address Pointer	Partial Word Desig.	Chain Flag	Monitor Interrupt Flag	Byte Pointer	Final Buffer	Current Address

ESI BUFFER CONTROL WORD FORMAT

31 29	28 18	17 0
Partial Word Designator	Final Address Compare Bits	Current Address

Partial Word Designator Definitions

31	30	29		
X	X	1	Quarter Word	XX = 00 next byte 31-24 01 next byte 23-16 10 next byte 15- 8 11 next byte 7- 0

X	1	0	Half Word
			X = 0 next word 31-16
			X = 1 next word 15- 0	

1	0	0	Full Word	
0	0	0	Suppress Data*	

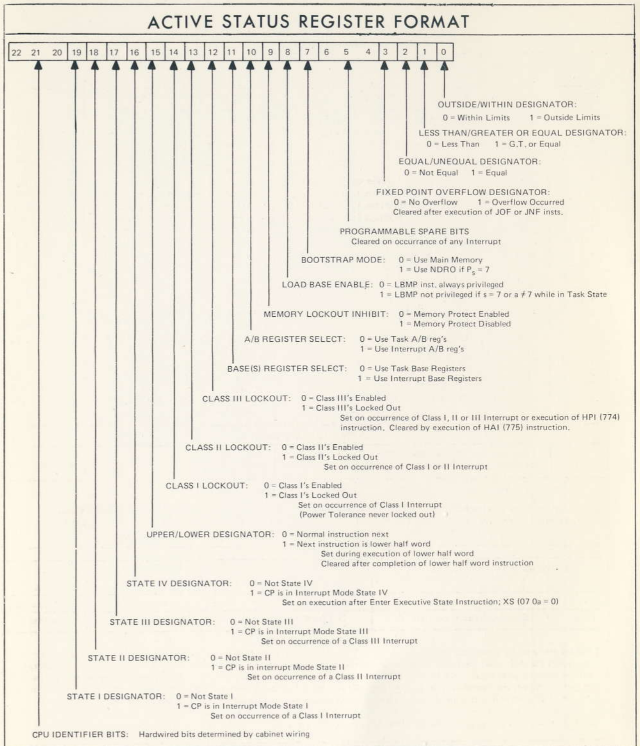
Maximum ESI Buffer is 2048 Words

* Suppress Data stores zeros at the specified address on input (ESI only)

IOC CONTROL MEMORY ASSIGNMENT

Address	Assignment
0-17	Input
20-37	Output
40-57	External Function
60-77	External Interrupt

ACTIVE STATUS REGISTER FORMAT



OUTSIDE/WITHIN DESIGNATOR:
0 = Within Limits 1 = Outside Limits

LESS THAN/GREATER OR EQUAL DESIGNATOR:
0 = Less Than 1 = G.T. or Equal

EQUAL/UNEQUAL DESIGNATOR:
0 = Not Equal 1 = Equal

FIXED POINT OVERFLOW DESIGNATOR:
0 = No Overflow 1 = Overflow Occurred
Cleared after execution of JOF or JNF insts.

PROGRAMMABLE SPARE BITS
Cleared on occurrance of any Interrupt

BOOTSTRAP MODE: 0 = Use Main Memory
1 = Use NDRO if $P_s = 7$

LOAD BASE ENABLE: 0 = LBMP inst. always privileged
1 = LBMP not privileged if s = 7 or a ≠ 7 while in Task State

MEMORY LOCKOUT INHIBIT: 0 = Memory Protect Enabled
1 = Memory Protect Disabled

A/B REGISTER SELECT: 0 = Use Task A/B reg's
1 = Use Interrupt A/B reg's

BASE(S) REGISTER SELECT: 0 = Use Task Base Registers
1 = Use Interrupt Base Registers

CLASS III LOCKOUT: 0 = Class III's Enabled
1 = Class III's Locked Out
Set on occurrence of Class I, II or III Interrupt or execution of HPI (774) instruction. Cleared by execution of HAI (775) instruction.

CLASS II LOCKOUT: 0 = Class II's Enabled
1 = Class II's Locked Out
Set on occurrence of Class I or II Interrupt

CLASS I LOCKOUT: 0 = Class I's Enabled
1 = Class I's Locked Out
Set on occurrence of Class I Interrupt
(Power Tolerance never locked out)

UPPER/LOWER DESIGNATOR: 0 = Normal instruction next
1 = Next instruction is lower half word
Set during execution of lower half word
Cleared after completion of lower half word instruction

STATE IV DESIGNATOR: 0 = Not State IV
1 = CP is in Interrupt Mode State IV
Set on execution after Enter Executive State Instruction; XS (07 0a = 0)

STATE III DESIGNATOR: 0 = Not State III
1 = CP is in Interrupt Mode State III
Set on occurrence of a Class III Interrupt

STATE II DESIGNATOR: 0 = Not State II
1 = CP is in interrupt Mode State II
Set on occurrence of a Class II Interrupt

STATE I DESIGNATOR: 0 = Not State I
1 = CP is in Interrupt Mode State I
Set on occurrence of a Class I Interrupt

CPU IDENTIFIER BITS: Hardwired bits determined by cabinet wiring

Note: The State bits (19–16) are mutually exclusive. Operation with multiple State bits set is "UNDEFINED."

The Task State is defined as bits 19–16 = 0

The Interrupt State is defined as any bit 19--16 ≠ 0.

Bits 11-9 are normally cleared for Task and set for the Interrupt States. However, this is not a requirement.

The SCT, HSCT instructions store all 23 bits of the ASR. However, the LCT or HLCT load only bits 14-0.
To load all 23 bits of the ASR, an IR must be performed.

MEMORY PROTECTION REGISTERS

Storage Protection Register (SPR)

20	19	18	17	16	15 0
I	OR	OW	IA	IR	R
Allow Instruction Execution*	Allow Operand Reading*	Allow Operand Writing*	Allow Indirect Addressing*	Use Interrupt B&S Registers during Indirect Addressing*†	Displacement Value

*Operation Allowed if Bit is Set

Segment Identification Register (SIR)

20	19 17	16	15 0
	SIR_s		SIR_d
	Base Register Designator		16 Bit Displacement

† For complete description of all possible uses, consult the equipment specification NAVSHIPS 0967-051-6291, also, see notes under repeat and indirect addressing tables.

BREAKPOINT REGISTER

19	18	17 Comparison Address Bits 0
0	0	–Disabled
0	1	–Instruction address*
1	0	–Operand address*
1	1	–Instruction and operand addresses*

* An instruction breakpoint match is obtainable on the operand address of a conditional jump instruction, satisfied or unsatisfied. The breakpoint compare is done on the address as it is requested. When a jump instruction is executed the jump address will be requested: (and the breakpoint match will occur) whether the jump condition is met or not.

The P-storage of a satisfied instruction breakpoint interrupt on the operand address of a jump instruction will be the P address of the jump instruction: which did not complete due to the interrupt.

An Instruction or Operand Breakpoint Interrupt occurring on a remotely executed instruction will store the Address of the Execute Remote instruction at CMR 144 (P-storage DSW).

INTERRUPT STATUS CODES

(in descending priority)

Class	INTERRUPT	Status Code Bits** 9 8 7 6 5 4 3 2 1 0	Action Taken
I*	Power Tolerance (never locked out)	0 0 0 0 0 0 1 1 1 1	(ASR)→CMR141 ISC→CMR142 (P)→CMR143 (CMR140)→P Set ASR bits 19, 14-8. Clear bits 6-0. Bit 7 is unchanged.
I	CP—Operand Memory Resume✓	0 0 M M M M 0 0 0 0	(ASR)→CMR141 ISC→CMR142 (P)→CMR143 NDRO Address 000_8→P Set ASR bits 19, 14-8, 7. Clear bits 6-0.
I	CP—IOC Command Resume✓	K K 0 0 0 0 0 0 0 1	
I	CP—Instruction Memory Resume✓	0 0 M M M M 0 0 1 0	
I	CP—IOC Interrupt Code Resume✓	K K 0 0 0 0 0 0 1 1	
I*	IOC Memory Resume	K K M M M M 1 0 1 0	
I*	Intercomputer Timeout	K K C C C C 1 0 1 1	
II*	Interprocessor Interrupt	0 0 0 0	(ASR)→CMR145 ISC→CMR146 (P)→CMR147 (CMR144)→P Set ASR bits 18, 13-8. Clear bits 6-0. ASR bit 7 set only if performing AUTO REC. Otherwise bit 7 is unchanged.
II	Floating Point Error✓	0 0 0 1	
II	CP Illegal Instruction Error‡✓††	0 0 1 0	
II	Privileged Instruction Error✓††	0 0 1 1	
	Not Assigned	0 1 0 0	
II	Operand Breakpoint Match✓†	0 1 0 1	
II	Operand Read or Indirect Addressing✓	0 1 1 0	
	Not Assigned	0 1 1 1	
	Not Assigned	1 0 0 0	
II	Operand Write✓	1 0 0 1	
II	Operand Limit✓	1 0 1 0	
II	Instruction Breakpoint Match✓†	1 0 1 1	
	Not Assigned	1 1 0 0	
II	Instruction Execute ✓††	1 1 0 1	
II	Instruction Limit✓	1 1 1 0	
II*	CP Monitor Clock	1 1 1 1	
III*	IOC Illegal CAR Instruction	K K 0 0 P P 0 0 0 0	(ASR)→CMR151 ISC→CMR152 (P)→CMR153 (CMR150)→P Set ASR bits 17, 12-8. Clear bits 6-0. Bit 7 is unchanged.
III*	IOC Illegal Chain Instruction	K K C C C C 0 1 F F	
III*	IOC CP Interrupt	K K 0 0 0 0 1 0 1 1	
III*	IOC Monitor Clock	K K 0 0 0 0 1 0 1 0	
III*	IOC External Interrupt Monitor	K K C C C C 1 1 0 0	
III*	IOC External Function Monitor	K K C C C C 1 1 0 1	
III*	IOC Output Data Monitor	K K C C C C 1 1 1 0	
III*	IOC Input Data Monitor	K K C C C C 1 1 1 1	
IV	Executive Return	ISC = sy + $(B)_b$ Z.E. 16 bit ISC assigned thru software	(ASR)→CMR155 ISC→CMR156 (P)→CMR157 (CMR154)→P Set ASR bits 16, 11-8. Clear bits 6-0. Bit 7 is unchanged.

* Queued

** Definitions: PP—CPU NO. (0-2)
MMMM—Memory Bank (0-17)
CCCC—IOC Channel (0-17)
KK—IOC NO. (0-3)

FF = 00—EXT. INT.
01—EXT. FCT.
10—OUTPUT
11—INPUT

‡ If in Interrupt Mode and AUTO REC switch selected, then jump to NDRO address:
01 if bootstrap 0 selected
02 if bootstrap 1 selected
03 if bootstrap 2 selected

† Maintenance Console Breakpoint Program/Manual switch must be in the PROGRAM position.

✓ Stored P value is the address of the instruction causing the interrupt. (Exception - If the processor is executing an instruction while in the repeat mode, the stored P value will be the address of the repeat instruction.)

†† Fault conditions which illuminate program fault light.

For all Class IV, Class III and Class I or II not denoted above, the Stored P value is the address of the next instruction in the interrupted program. (Exception - if the processor is executing an instruction while in the repeat mode, the stored P value will be the address of the repeat instruction.)

FIXED POINT OVERFLOW CONDITIONS

a) Addition: Addend and augend have like signs and the sum has a different sign.

b) Subtraction: Minuend and subtrahend have different signs and the difference has a sign different from the minuend.

c) Division: Attempt to divide by zero or if the magnitude of divisor times 2^{31} is less than the magnitude of the dividend.

d) Square Root: Attempt to take square root of a negative number or a number greater than or equal to 2^{62}.

INDIRECT ADDRESSING FORMATS

NORMAL (IW) $Y = y + (B_b) + (S_s)$

31 30	29 20	19 17	16	15 13	12 0
C = 10	Not Used	Index Register Designator (b)	Indirect Addressing Designator (i)	Base Register Designator (s)	Relative Address (y)

SPECIAL BASE (IWS) $Y = sy + (S_b)$

31 30	29	28 20	19 17	16	15 0
C = 00	$C_1 = 0$	Not Used	Base Register Designator (s)	Indirect Addressing Designator (i)	16-bit Relative Address (sy)

SPECIAL INDEX (IWB) $Y = sy + (B_b)_{15-0} + (S_{(B_b)_{19-17}})$

31 30	29	28 20	19 17	16	15 0
C = 00	$C_1 = 1$	Not Used	Index Register Designator (b)	Indirect Addressing Designator (i)	16-bit Relative Address (sy)

CHARACTER (IWC) $Y = y + (B_b) + (S_s)$

31 30	29 25	24 20	19 17	16	15 13	12 0
C = 01	Character Length Designator (w)	Bit Position Designator (p) [Specifies the LSB of character]	Index Register Designator (b)	Indirect Addressing Designator (i)	Base Register Designator (s)	Relative Address (y)

CHARACTER INCREMENT (IWCI) $Y = y + (B_b) + (S_s)$ Each reference: (p) – (w)→p. If (p) – (w)<0, then 32 – (w)→p and y + 1→y.

31 30	29 25	24 20	19 17	16	15 13	12 0
C = 11	Character Length Designator (w)	Bit Position Designator (p) [Specifies LSB of character]	Index Register Designator (b)	Indirect Addressing Designator (i)	Base Register Designator (s)	Relative Address (y)

INDIRECT WORD ADDRESS GENERATION

If Bits 31, 30 & 29 Equal	and *i* Equals	Designators in current indirect control word used as follows:
000 (IWS)	1	The next indirect word address $Y = sy + (S_b)$
001 (IWB)	1	The next indirect word address $Y = sy + (B_b) + (S)$ as designated by $(B_b)_{19-17}$
000 (IWS)	0	The operand* address $Y = sy + (S_b)$
001 (IWB)	0	The operand* address $Y = sy + (B_b) + (S)$ as designated by $(B_b)_{19-17}$
10X (IW)	1	The next indirect word address is $Y = y + (B_b) + (S_s)$
01X (IWC)	1	The next indirect word address is $Y = y + (B_b) + (S_s)$
11X (IWCI)	1	The next indirect word address is $Y = y + (B_b) + (S_s)$
10X (IW)	0	The operand* address $Y = y + (B_b) + (S_s)$
01X (IWC)	0	The address of the single character operand defined by *w* and *p* is $Y = y + (B_b) + (S_s)$
11X (IWCI)	0	The address of the sequential character operand defined by *w* and *p* is $Y = y + (B_b) + (S_s)$. Then if $p - w \geqslant 0$, $p - w \rightarrow p$ and $y \rightarrow y$ if $p - w < 0$, $32 - w \rightarrow p$ and $y + 1 \rightarrow y$ The updated indirect control word is stored back into main memory for the next execution.

* The operand is defined by the function code and in Format I instructions the *k* designator.

OPERAND INTERPRETATIONS FOR JUMP INSTRUCTIONS (FORMAT III)

k is not used
When i = 0 the jump address $Y = y + (B_b) + (S_s)$
When i = 1 the indirect control address $Y = y + (B_b) + (S_s)$.

Indirect addressing continues through all indirect control words until i = 0 is encountered. Depending on the c-field in the indirect control word the jump address will be $Y = y + (B_b) + (S_s)$, $Y = sy + (S_b)$ or $Y = sy + (B_b)_{15-0} + (S)$ as specified by $(B_b)_{19-17}$ as designated by those respective fields in the indirect control word. A request for character addressing in the indirect control word for a Format III instruction is not allowed. These are jump instructions.

Note: Any jump instruction with i = 1 and $(SPR)_S$ bit 16 = 1 is privileged.

REPEAT CONDITIONS (X For Usable)*

Repeated Instruction	a Field of Repeat Inst.								Terminate on	Repeated Instruction	a Field of Repeated Inst.								Terminate on
	0	1	2	3	4	5	6	7			0	1	2	3	4	5	6	7	
01 0	X	X	X	X	X			X	A_a	17	X	X	X	X	X			X	A_a
01 1	X	X	X	X	X			X	A_a	20					X			X	
01 2	X	X	X	X	X			X	A_{a+1}	21					X			X	
01 3	X	X	X	X	X			X	A_a	22					X			X	
01 4	X	X	X	X	X			X	A_{a+1}	23					X			X	
01 5	X	X	X	X	X			X	A_a	24	X	X	X	X	X	X	X	X	A_a
01 6	X	X	X	X	X			X	A_{a+1}	26					X	X	X	X	OP*
01 7	X	X	X	X	X			X	A_{a+1}	27					X	X	X	X	OP*
02 0	X	X	X	X	X			X	A_a	32‡					X			X	
02 4	X	X	X	X	X	X	X	X	OP**	33‡					X			X	
02 5	X	X	X	X	X	X	X	X	A_{a+1}	34‡	X	X	X	X	X	X	X	X	A_{a+1}
02 6	X	X	X	X	X	X	X	X	A_{a+1}	35‡	X	X	X	X	X	X	X	X	A_a
03 0‡	X	X	X	X	X	X	X	X	A_a	36‡	X	X	X	X	X	X	X	X	A_{a+1}
03 1‡	X	X	X	X	X	X	X	X	A_a	37‡	X	X	X	X	X	X	X	X	A_a
03 2‡	X	X	X	X	X	X	X	X	A_{a+1}	40					X			X	
03 3‡	X	X	X	X	X			X	A_a	41					X			X	
03 4‡	X	X	X	X	X	X	X	X	A_{a+1}	42	X	X							CD
03 5‡	X	X	X	X	X	X	X	X	A_{a+1}	44	X	X	X	X	X	X			CD
03 6‡	X	X	X	X	X	X	X	X	A_{a+1}	45							X	X	CD
03 7‡	X	X							CD	46	X	X	X	X	X	X			CD
10	X	X	X	X	X			X	A_a	47	X	X	X	X	X	X			CD
12	X	X	X	X	X			X	A_{a+1}	54†					X			X	
13	X	X	X	X	X			X	A_a	55†					X			X	
14	X	X	X	X	X			X	A_a	56†					X			X	
15	X	X	X	X	X			X	A_{a+1}	57†					X			X	
16	X	X	X	X	X			X	A_a										

* Unpredictable operation will occur for unusable conditions.

** OP is the 32-bit result of the execution.

† In the repeat mode, ak + 1→ak for each execution. These instructions are not interruptable in the repeat mode. These instructions are privileged if repeat is attempted in the Task mode (Privileged Instruction Error).

‡ For replace class instructions, use S6 on store cycle; if in repeat instruction, b ≠ 0.

Note: Any repeated instruction with i = 1 and $(SPR)_S$ bit 16 = 1 is privileged.
If B7 = Ø skip next instruction.
At termination, sy sign extended will have been added to (B_b).

REPEAT TERMINATE CONDITIONS

a	Non-Compare Instructions	a	Compare Instructions
0	Terminate if A ≠ 0	0	Terminate if CD set to ≠
1	Terminate if A = 0	1	Terminate if CD set to =
2	Terminate if A > 0	2	Terminate if CD set to >
3	Terminate if A < 0	3	Terminate if CD set to >
4	Do not terminate	4	Terminate if CD set to <
5	Terminate if (A) is even parity on write into memory	5	Terminate if CD set to <
6	Terminate if (A) is odd parity on write into memory	6	Terminate if CD set to outside limit
7	Do not terminate	7	Terminate if CD set to within limit

ROUNDING OF FLOATING POINT RESULTS

Mantissa rounding is performed (A_{a+1}) according to the status of the intermediate double-length result in the arithmetic section for add, subtract and multiply; and according to the value of the remainder in divide operations. The final sum or difference mantissa in (A_{a+1}) is rounded as follows:

1. If bit 31 of the 64 bit intermediate sum or difference equals 1 and (A_{a+1}) are positive, 1 is added to (A_{a+1}).
2. If bit 31 of the 64 bit intermediate sum or difference equals 0 and (A_{a+1}) are negative, 1 is subtracted from (A_{a+1}).
3. If not 1 or 2 above, (A_{a+1}) are not changed.
4. If overflow results in 1 or 2 above (A_{a+1}) are shifted right one place, 1 is added to the characteristic exponent in A_a and the mantissa sign bit in A_{a+1} is restored.

Rounding of a product mantissa is done before final sign correction.

1 is added to (A_{a+1}) if bit 31 of the 64 bit intermediate product equals 1; otherwise (A_{a+1}) are not changed.

Rounding of a quotient mantissa is done before final sign correction.

1. If the remainder is equal to or greater than one-half the divisor and there is no overflow, 1 is added to (A_{a+1}).
2. If bit 31 of the quotient in A_{a+1} equals 1, (A_{a+1}) are shifted right one place, $(A_{a+1})_0$ before shifting, is added to the shifted (A_{a+1}) and 1 is added to the characteristic exponent in A_a.

PROGRAMMER NOTES

USE A PENCIL FOR ENTRIES AND CHANGES MAY BE MADE WITH AN ERASER.

IOC BUFFERED REQUEST PRIORITY

REQUEST PRIORITY	REQUEST TITLE	ACTION WHEN PROCESSED
Channel dependent	Buffer request (includes EI, EF, outputs and input)	Performs transfer based on buffer request priority first by channel (17 highest, 0 lowest) then as specified below.
1a	External interrupt request (occurs when an external device sets the external interrupt request line)	Performs a one word external interrupt word transfer using the control memory word at CMR address for channel.
1b	External function request (occurs when an external device sets the external function request line)	Performs a one word external function code word transfer using the control memory word at CMR address for channel.
1c	Output data request (occurs when an external device sets the output data request line)	Performs a one word output data word transfer using the control memory word at CMR address for channel.
1d	Input data request (occurs when an external device sets the input data request line)	Performs a one word input data word transfer using the control memory word at CMR address for channel.

IOC REQUEST (NONBUFFERED) PRIORITY

PRIORITY REQUEST	REQUEST TITLE	ACTION WHEN PROCESSED
1	Intercomputer Terminate Sequence	Performs the termination functions when an intercomputer channel terminates.
2	Clock Request	Decrement the IOC monitor clock by 1 and increment the real-time clock by 1.
3	Central Processor Instruction for IOC and Interrupt Status Code Requests.	Performs the function as commanded according to priority below.
3a	CP No. 0 Request*	
3b	CP No. 1 Request*	
3c	CP No. 2 Request*	
4	Central Processor Command Address Request	Performs the function as commanded according to priority below.
4a	CP No. 0 Request * The numbers 1 & 2 are IOC port numbers and not necessarily the same as CPU I.D.	
4b	CP No. 1 Request	
4c	CP No. 2 Request	
5	Chain Commands (Note 1) (channel associated)	Performs the function as commanded according to normal channel priority.
Note 1:	For buffer terminations with the chain bit (IOCM 35) set, normal priority is bypassed and the next sequential command in that chain is executed.	

IOC MAINTENANCE CONSOLE DISPLAY

REGISTER SELECT	CM ADDRESS SELECT/SELECT 2	I/O CONTROLLER DISPLAY (IOC must be in SEQ mode)	MON/ CHAIN
CMP	0-77	Bits 55-38 of IOCM (CAP) specified by CM ADDRESS SELECT	N.U.
CMU	0-77	Bits 37, 36 and 33-18 of IOCM specified by CM ADDRESS SELECT	bit 35 (chain)
CML	0-77	Bits 17-0 of IOCM specified by CM ADDRESS SELECT	bit 34 (monitor)
DIRU	N.U.	Bits 31-18 of DIR	N.U.
DIRL	N.U.	Bits 17-0 of DIR	N.U.
SEL 2	CAR + 0	(CAR 0) bits 17-0‡	CAR ACT.
	CAR + 1	(CAR 1) bits 17-0‡	CAR ACT.
	CAR + 2	(CAR 2) bits 17-0‡	CAR ACT.
	ILR + 0	(ILR 0) channels 15-0†	N.U.
	ILR + 1	(ILR 1) channels 15-0†	N.U.
	ILR + 2	(ILR 2) channels 15-0†	N.U.
	CHAN + 0	Buffer actives by type on channels 3-0†	N.U.
	CHAN + 1	Buffer actives by type on channels 7-4 †	N.U.
	CHAN + 2	Buffer actives by type on channels 10-13†	N.U.
	CHAN + 3	Buffer actives by type on channels 17-14†	N.U.
	CHAIN + 0	Chain actives by type on channels 3-0 †	N.U.
	CHAIN + 1	Chain actives by type on channels 7-4 †	N.U.
	CHAIN + 2	Chain actives by type on channels 13-10†	N.U.
	CHAIN + 3	Chain actives by type on channels 17-14†	N.U.
	60	(RTC) bits 17-0	CAR 0 ACTIVE
	61	(RTC) bits 31-18	CAR 1 ACTIVE
	62	(IOC MONITOR CLK) 15-0	N.U.

N.U. Not Used

† These displays are indicate only and are available in both RUN and SEQ mode.

‡ These displays are available in both RUN and SEQ mode.