AN/UYK-7 COMPUTER
REPERTOIRE OF INSTRUCTIONS

Code	Mnemonic	NAME	DESCRIPTION	F	CA	R	UF	Time‡ μs
01 0	OR	Inclusive OR (Selective Set A)	(Y) ∨ (A_B)→A_B	II	Y	Y	2	1.5
01 1	SC	Selective Clear A	(A_B) ○ (Y)→A_B	II	Y	Y	2	1.5
01 2	MS	Selective Substitute	(Y)$_m$→$(A_{A+1})_m$ for all $(A_A)_{n=1}$; $(A_A)_{n=}(A_B)_t$	II	Y	Y	2	1.5
01 3	XOR	Exclusive OR (Sel. Comp. A)	(Y) ≢ (A_B)→A_A; $(A_A)_{n'}$→$(A_B)_n$ for (Y)$_{n=1}$	II	Y	Y	2	1.5
01 4	ALP	Add Logical Product	(A_{A+1}) + (Y) ○ (A_B)→A_{A+1}; $(A_B)_t$=$(A_A)_t$	II	Y	Y	2	1.5
01 5	LLP	Load Logical Product	(Y) ○ (A_B)→A_B	II	Y	Y	2	1.5
01 6	NLP	Subtract Logical Product	(A_{A+1})−(Y) ○ (A_B)→A_{A+1}; $(A_B)_t$=$(A_A)_t$	II	Y	Y	2	1.5
01 7	LLPN	Load Logical Product Next	(Y) ○ (A_B)→A_{A+1}; $(A_B)_t$=$(A_A)_t$	II	Y	Y	2	1.5
02 0	CNT	Count Ones	No. of Bits Set in (Y)→A_B	II	Y	Y	2	7.5†
02 2	XR	Execute Remote	(Y)→U	II	N	N	8	1.5
02 3	XRL	Execute Remote Lower	(Y)$_L$→U	II	N	N	8	1.5
02 4	SLP	Store Logical Product	(A_{A+1}) ○ (A_B)→Y; $(A_B)_t$=$(A_A)_t$; $(A_{A+1})_t$=$(A_{A+1})_t$	II	Y	Y	2	1.5
02 5	SSUM	Store Sum	(A_B)+(A_{A+1})→A_{A+1} & Y; $(A_B)_t$=$(A_B)_t$	II	Y	Y	2	2.0
02 6	SDIF	Store Difference	(A_{A+1})−(A_B)→A_{A+1} & Y; $(A_B)_t$=$(A_B)_t$	II	Y	Y	2	2.0
02 7	DS	Double Store A	(A_{A+1}, A_B)→Y+1, Y	II	N	N	2	3.0
03 0	ROR	Replace Inclusive OR	(Y) ∨ (A_B)→A_B & Y	II	Y	Y	2	2.5
03 1	RSC	Replace Selective Clear	(A_B) ○ (Y)′→A_B & Y	II	Y	Y	2	2.5
03 2	RMS	Replace Selective Substitute	(Y)$_m$→$(A_{A+1})_m$ for all $(A_A)_{n=1}$; Then $(A_A)_{n=1}$→Y; $(A_B)_t$=$(A_B)_t$	II	Y	Y	2	2.5
03 3	RXOR	Replace Exclusive OR	(Y) ≢ (A_B)→A_B & Y; $(A_B)_{n'}$→A_B & Y for $Y_{n=1}$	II	N	Y	2	2.5
03 4	RALP	Replace A→Logical Product	(A_{A+1})+(Y) ○ (A_B)→A_{A+1} & Y; $(A_B)_t$=$(A_B)_t$	II	Y	Y	2	2.5
03 5	RLP	Replace A→Logical Product	(Y) ○ (A_B)→A_{A+1} & Y; $(A_B)_t$=$(A_B)_t$	II	Y	Y	2	2.5
03 6	RNLP	Replace A−Logical Product	(A_{A+1})−(Y) ○ (A_B)→A_{A+1} & Y; $(A_B)_t$=$(A_B)_t$	II	Y	Y	2	2.5
03 7	TSF	Test and Set Flag	If (Y)$_{31}$=0, CD Set EQUAL. 1→Y$_{31}$ If (Y)$_{31}$=1, CD Set UNEQUAL.	II	N	Y	8	2.5
05 0	DL	Double Load A	(Y+1, Y)→A_{B+1}, A_B	II	N	N	2	3.0
05 1	DA	Double Add A	(A_{A+1}, A_B)+(Y+1, Y)→A_{B+1}, A_B	II	N	N	2	3.0
05 2	DAN	Double Subtract A	(A_{A+1}, A_B)−(Y+1, Y)→A_{B+1}, A_B	II	N	N	2	3.0
05 3	DC	Double Compare	Compare (A_{A+1}, A_B) to (Y+1, Y), Set CD	II	N	N	2	3.0
05 4	LBMP	Load Base and Memory Protection	(Y)$_{17}$→S_B; (Y+1)$_{20}$→S_{PR_B}; Y→SIR_B Privileged if; ASR bit 8 =0, s ≠ 7 or a=7	II	N	N	2	5.75
06 0	FA	Floating-point Add	Shift (A_{A+1}) or (Y+1) Right such that (A_B)−(Y) (A_{A+1})−(Y+1)→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_B)−(Y) (A_{A+1})−(Y+1)→A_{A+1}; Normalize	II	N	N	2	6.25†
06 2	FM	Floating-point Multiply	(A_B)+(Y)→(A_B) (A_{A+1}) + (Y+1)→A_{A+1}; Normalize	II	N	N	2	10.0†
06 3	FD	Floating-point Divide	(A_B)−(Y)→(A_B) (A_{A+1})−(Y+1)→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	$sy+(B_B)$→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	II	N	N	11	4.0
07 1**	AEI	Allow Enable Interrupt	Allow Monitor interrupts from IOCa on Channels n; IF bit n of $sy+(B_B)$=1	II	N	N	6	2.0
07 2**	PEI	Prevent Enable Interrupt	Prevent Monitor interrupts from IOCa on Channels n; IF bit n of $sy+(B_B)$=1	II	N	N	6	2.0
07 3**	LIM	Load IOC Monitor Clock	$sy+(B_B)$→IOCa MON CLK	II	N	N	6	3.0
07 4**	IO	Initiate I/O	Initiate IOCa at address Y	II	N	N	2	3.5
07 5*	IR	Interrupt Return	Return to State Specified by ASR storage DSW	II	N	N	9	3.0
07 6	RP	Repeat	Repeat N.I.B; Times; sy of Repeat added to B_B of N.I. after each cycle. See Repeat Conditions	II	N	N	6	1.5
10	LA	Load A	Y→A_B	I	Y	Y	1	1.5
11	LXB	Load A and Index B	Y→A_B; (B_B)+1→B_B	I	Y	N	1	1.5
12	LDIF	Load Difference	Y−(A_B)→A_{A+1}; $(A_B)_t$=$(A_A)_t$	I	Y	Y	1	1.5
13	ANA	Subtract A	(A_B)−Y→A_B	I	Y	Y	1	1.5

REPERTOIRE OF INSTRUCTIONS (CONT.)

Code	Mnemonic	NAME	DESCRIPTION	F	CA	R	UF	Time‡ μs
14	AA	Add A	(A_B)+Y→A_B	I	Y	Y	1	1.5
15	LSUM	Load Sum	(A_B)+Y→A_{B+1}; $(A_B)_t$=$(A_B)_t$	I	Y	Y	1	1.5
16	LNA	Load Negative	Y′→A_B	I	Y	Y	1	1.5
17	LM	Load Magnitude	|Y|→A_B	I	Y	Y	1	2.0
20	LB	Load B	Y→B_B	I	Y	Y	1	2.0
21	AB	Add B	(B_B)+Y→B_B; B_B zero extended	I	Y	Y	1	2.0
22	ANB	Subtract B	(B_B)−Y→B_B; B_B zero extended	I	Y	Y	1	2.0
23	SB	Store B	(B_B)→Y	I	Y	Y	1	1.5
24	SA	Store A	(A_B)→Y	I	Y	Y	1	1.5
25	SXB	Store A and Index B	(A_B)→Y; (B_B)+1→B_B	I	Y	N	1	1.5
26	SNA	Store Negative	(A_B)′→Y	I	Y	Y	1	1.5
27	SM	Store Magnitude	|(A_B)|→Y	I	Y	Y	1	1.5
32	BZ	Clear Bit	0→Y_{sk}	I	N	Y	3	2.5
33	BS	Set Bit	1→Y_{sk}	I	Y	Y	3	2.5
34	RA	Replace Add	(A_B)+Y→A_{B+1} & Y; $(A_B)_t$=$(A_B)_t$	I	Y	Y	1	2.5
35	RI	Replace Increment	Y+1→A_B & Y	I	Y	Y	1	2.5
36	RAN	Replace Subtract	Y−(A_B)→A_{B+1} & Y; $(A_B)_t$=$(A_B)_t$	I	Y	Y	1	2.5
37	RD	Replace Decrement	Y−1→A_B & Y	I	Y	Y	1	2.5
40	M	Multiply A	(A_B) • Y→A_{B+1}, A_B	I	Y	Y	1	7.5†
41	D	Divide A	(A_{B+1}, A_B)÷Y→A_B; remainder→A_{B+1}	I	Y	Y	1	14.5†
42	BC	Compare Bit to Zero	If (Y)$_{sk}$=0, CD Set EQUAL If (Y)$_{sk}$=1, CD Set UNEQUAL	I	N	Y	3	1.5
43	CXI	Compare Index Increment	If (B_B) ≥ Y, CD Set OUTSIDE, 0→B_B If (B_B) < Y, CD Set WITHIN, (B_B)+1→B_B	I	Y	Y	1	2.0
44	C	Compare	Compare (A_B) to Y, Set the CD	I	Y	Y	1	1.5
45	CL	Compare Limits	If (A_{A+1}) > Y ≥ (A_B), Set CD within	I	Y	Y	1	1.5
46	CM	Compare Masked	Compare (A_{A+1}) to (A_B) ○ Y, Set the CD	I	Y	Y	1	1.5
47	CG	Compare Gated	Compare |Y−(A_B)| to (A_{B+1}), Set the CD	I	Y	Y	1	1.5
50 0	JEP	Jump on Even Parity	If (A_{B+1}) ○ (A_B) is Even Parity, jump to Y	III	N	N	1	2.0
50 1	JOP	Jump on Odd Parity	If (A_{B+1}) ○ (A_B) is Odd Parity, jump to Y	III	N	N	1	2.0
50 2	DJZ	Jump Double Precision Zero	If (A_{B+1}, A_B)=0, jump to Y	III	N	N	1	1.8
50 3	DJNZ	Jump Double Precision Not Zero	If (A_{B+1}, A_B)≠0, jump to Y	III	N	N	1	2.0
51 0	JP	Jump A Positive	If (A_B) ≥ 0, jump to Y	III	N	N	1	1.5
51 1	JN	Jump A Negative	If (A_B) < 0, jump to Y	III	N	N	1	1.5
51 2	JZ	Jump A Zero	If (A_B)=0, jump to Y	III	N	N	1	1.5
51 3	JNZ	Jump A Not Zero	If (A_B) ≠ 0, jump to Y	III	N	N	1	1.5
52 0	LBJ	Load B and Jump	P + 1→B_B, jump to Y	III	N	N	1	1.8
52 1	JBNZ	Index Jump B	If (B_B) ≠ 0, then (B_B)−1→B_B, 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=0	P+1→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→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→Y, jump to Y+1 at restart	III	N	N	1	3.75
53 3	J	Manual Jump a=0	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_k√	LCT	Load CMR Task	(Y)→CMR_{ak}	I	N	Y	3	1.5
55*	LCI	Load CMR Interrupt	(Y)→CMR_{ak+100}	I	N	Y	3	1.5

REPERTOIRE OF INSTRUCTIONS (CONT.)

Code	Mnemonic	NAME	DESCRIPTION	F	CA	R	UF	Time‡ μs
56_k√	SCT	Store CMR Task	(CMR_{ak})→Y	I	N	Y	3	1.5
57*	SCI	Store CMR Interrupt	(CMR_{ak+100})→Y	I	N	Y	3	1.5
60_k√$_{i=0}$	HSCT	Store CMR in A	(CMR_{ak})→A_b	IV A	N	N	4	1.75
60^i*$_{i=1}$	HSCI	Store CMR in A	(CMR_{ak+100})→A_b	IV A	N	N	4	1.75
61_k√$_{i=0}$	HLCT	Load CMR from A	(A_b)→CMR_{ak}	IV A	N	N	4	1.75
61^i*$_{i=1}$	HLCI	Load CMR from A	(A_b)→CMR_{ak+100}	IV A	N	N	4	1.75
62	HLC	Shift Left Circularly	(A_A) Left Shifted End Around→A_A	IV B	N	N	10	1.75
63	HDLC	Shift Left Circularly Double	(A_{A+1}, A_A) Left Shifted End Around→A_{A+1}, A_A	IV B	N	N	10	1.75
64	HRZ	Shift Right Fill Zeros	(A_A) Right Shifted, Zero Fill→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→A_{A+1}, A_A	IV B	N	N	10	1.75
66	HRS	Shift Right Fill Sign	(A_A) Right Shifted, Sign Fill→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→A_{A+1}, A_A	IV B	N	N	10	1.75
70 0	HSF	Scale Factor	Normalize (A_A) Shift Count→A_b	IV A	N	N	5	2.25
70 1	HDSF	Double Scale Factor	Normalize (A_{A+1}, A_A) Shift Count→A_b	IV A	N	N	5	2.25
70 2	HCP	Complement A	$(A_A)'$→A_A	IV A	N	N	7	1.1
70 3	HDCP	Double Complement A	$(A_{A+1}, A_A)'$→A_{A+1}, A_A	IV A	N	N	7	1.1
71 0	HOR	Logical Sum	(A_A) ∘ (A_b)→A_A; $(A_b)_t$=$(A_b)_t$	IV A	N	N	5	1.0
71 1	HA	Sum	(A_A) + (A_b)→A_A	IV A	N	N	5	1.0
71 2	HAN	Difference	(A_A) − (A_b)→A_A	IV A	N	N	5	1.0
71 3	HXOR	Logical Difference	(A_A) ≢ (A_b)→A_A	IV A	N	N	5	1.0
71 5	HAND	AND	(A_A) ○ (A_b)→A_A; $(A_b)_t$=$(A_b)_t$	IV A	N	N	5	1.0
74 0	HM	Multiply Register	(A_A) • (A_b)→A_{A+1}, A_A	IV A	N	N	5	7.75†
74 1	HD	Divide Register	(A_{A+1}, A_A)÷(A_b)→A_A; Remainder→A_{A+1}	IV A	N	N	5	15.0†
74 2	HRT	Square Root	√(A_{A+1}, A_A)→A_b; Residue→A_{b+1}	IV A	N	N	5	15.0†
74 3	HLB	Load B_B with B_b	(B_b)→B_B	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) ≥ (A_A), Set CD in Limit	IV A	N	N	5	1.75
74 6	HCM	Compare Masked, Register	Compare (A_{A+1}) ○ (A_A) to (A_b), Set the CD	IV A	N	N	5	1.1
74 7	HCB	Compare B_b with B_B	Compare (B_b) to (B_B), Set the CD	IV A	N	N	5	2.0
77 0**	HSIM	Store IOC Monitor Clock in A	(IOC$_a$ MON CLK)→A_b	IV A	N	N	5	3.0
77 1	HSTC	Store Real-Time Clock in A	(IOC$_a$ RTC)→A_b	IV A	N	N	5	3.5
77 4*	HPI	Prevent Class III Interrupts	Set Class III Interrupt Lockout	IV A	N	N	9	2.25
77 5*	HAI	Allow Class III Interrupts	Clear Class III Interrupt Lockout	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

ULTRA/32 PSEUDO INSTRUCTIONS

10	ZA	Clear A	0→A_a	I	N	Y	7	1.5
20	ZB	Clear B	0→B_B	I	N	Y	7	2.0
20	NOOP	No Operation	0→B_0	I	N	N	7	2.0
23	SZ	Store Zeros	0→Y	I	Y	Y	12	1.5
74 3	HNO	Half Word No Operation	(B_0)→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_{ad}=10)	—	—	8	—
—	IWS	Indirect Word, Special Base (c_{ad}=00, c_1=0)	—	—	11	—
—	IWB	Indirect Word, Special Index (c_{ad}=00, c_1=1)	—	—	11	—
—	IWC	Indirect Word, Character (c_{ad}=01)	—	—	14	—
—	IWCI	Indirect Word, Character Increment (c_{ad}=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$_a$, b	7	s	10	a, m (shift by m) a, b, 1 (shift by B_B)	11	sy, b	14	y, w, p, b, s
2	a, y, b, s	5	a, b	8	y, b, s			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

*Privileged **CPU→IOC Instr.—Privileged √Privileged when ak=2X, 6X or 7X

†Execution time independent of overlap operation

‡Times shown assume 1.5 μs memory with operands not in same bank as instructions (overlapped).

REV. 5.71

I/O CONTROLLER COMMANDS

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; m=1 Allow Queued}	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 (bits 55-38) of CMA* 20k+j; Activate Chain	4	2.5
16 k=1	AOC	Set Output Chain Active on Cj		4	2.5
16 k=2	AFC	Set External Function Chain Active on Cj		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)$_{kj}$ ≠ 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	(IOC Control Memory)$_{kj}$→y	5	2.75
25	IBS	Set Bit	1→y$_{kj}$	5	3.25
26	IBZ	Clear Bit	0→y$_{kj}$	5	3.25
27	ITSF	Test and Set Flag	1→y$_{31}$; If (y)$_{31}$ was Originally Cleared, Skip; Else NI	6	3.25
		FORMATING MNEMONICS			
—	BCW	Buffer Control Word		8	—
—	BCWE	Buffer Control Word ESI		9	—

‡Command Address Register
*Control Memory Address

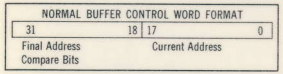
**ULTRA FORMAT

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

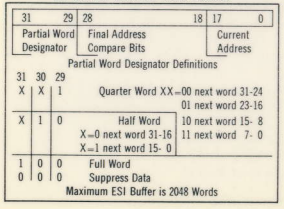
k—DESIGNATOR DEFINITIONS

	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

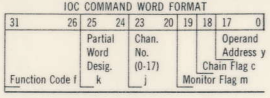
NORMAL BUFFER CONTROL WORD FORMAT



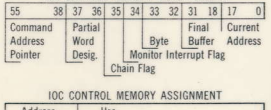
ESI BUFFER CONTROL WORD FORMAT



IOC COMMAND WORD FORMAT



IOC CONTROL MEMORY WORD FORMAT



IOC CONTROL MEMORY ASSIGNMENT

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

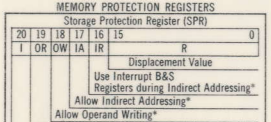
INTERRUPT STATUS CODES

Class	INTERRUPT	Status Code Bits** 9 8 7 6 5 4 3 2 1 0
I	CP—Operand Memory Resume	0 0 M M M M 0 0 0 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
I*	Power Tolerance (never locked out)	0 0 0 0 0 0 1 1 1 1
II*	Interprocessor Interrupt	0 0 0 0
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
III*	IOC Illegal Chain Instruction	K K C C C C C 0 1 F F
III*	IOC Monitor Clock	K K 0 0 0 0 1 0 1 0
III*	IOC CP Interrupt	K K 0 0 0 0 1 0 1 1
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	16 bit code assigned thru program

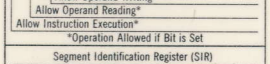
*Queued
**Definitions: PP—CPU NO. (0-2) FF—00—EXT. INT.
MMMM—Memory Bank (0-17) 01—EXT. FCT.
CCCC—IOC Channel (0-17) 10—OUTPUT
KK—IOC NO. (0-3) 11—INPUT

MEMORY PROTECTION REGISTERS

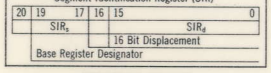
Storage Protection Register (SPR)



Segment Identification Register (SIR)



BREAKPOINT REGISTER



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

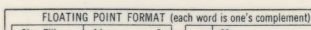
*Clock is Low order 16 bits.
**Not Addressable in the Task Mode.
(Privileged instruction error will occur)
†Lower 16 bits used for index and arithmetic functions. Upper three bits used **only** as a base-register designation.

ACTIVE STATUS REGISTER

Bit	Designator	
22-20	Central Processor Identifier	Hardwired
19	State I	
18	State II	
17	State III	
16	State IV	
15	Upper—lower	
14	Class I lockout	
13	Class II lockout	
12	Class III lockout	
11	Base (s) register selector	
10	Accumulator /B register selector	
9	Memory lockout inhibit	
8	Load base enable	
7	Bootstrap mode	
6-4	Programmable spare bits	
3	Fixed point overflow indicator	
2	0—Not equal 1—Equal	
1	0—Less than 1—G.T. or equal	
0	0—Within limits 1—Outside limits	

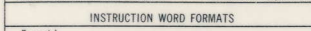
Bits 9-11 1—Interrupt mode
0—Task mode

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

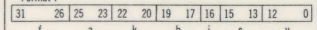


INSTRUCTION WORD FORMATS

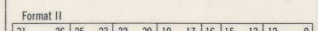
Format I



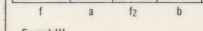
Format II



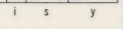
Format III



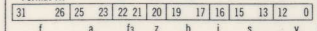
Format IV A



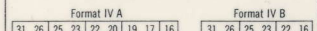
Format IV B



NORMAL INDIRECT ADDRESS WORD FORMAT



SPECIAL INDIRECT ADDRESS WORD FORMAT



f—Function Code
f$_2$ f$_3$ f$_4$—Subfunction Codes
a—Accumulator Register
k—Operand Interpretation
b—Index Register
i—Indirect Bit

c$_1$—Special Indirect Subfunction
0—Y=d+(S$_b$)
1—Y=d+(B$_b$)+(S) as specified by (B$_b$)$_{19-17}$

c—Addressing Designator
00—Indirect Special
10—Indirect Normal
01—Single Character
11—Sequential Character

s—Base Register
w—Field Width
p—Bit Position
y—Operand Address
x—Not Used
d—16 Bit Displacement
z—Not Used—Must be Zero
m—Shift Designator

Bit 26	Function
0	Shift by count 25—20
1	Shift by B$_b$ if 25=0
1	Shift by A$_8$ if 25=1

b is specified by bits 23—21

REPEAT CONDITIONS

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

a	Compare Instructions
0	Terminate if CD set to ≠
1	Terminate if CD set to =
2	Terminate if CD set to >
3	Terminate if CD set to ≥
4	Terminate if CD set to <
5	Terminate if CD set to ≤
6	Terminate if CD set to outside limit
7	Terminate if CD set to within limit

FORMAT I INSTRUCTION k—FIELD INTERPRETATION

k	Memory to Arithmetic (Read)	Arithmetic to Memory (Store)
0	sy SE+(B$_b$)→A 15-0 SE	Not Used
1	(Y$_{15-0}$)→A$_{15-0}$ SE	(A$_{15-0}$)→Y$_{15-0}$; Y31-16—Un
2	(Y$_{31-16}$)→A$_{15-0}$ SE	(A$_{15-0}$)→Y$_{31-16}$; Y15-0—Un
3	(Y$_{31-0}$)→A$_{31-0}$	(A$_{31-0}$)→Y$_{31-0}$
4	(Y$_{7-0}$)→A$_{7-0}$ ZE	(A$_{7-0}$)→Y$_{7-0}$; Y31-8—Un
5	(Y$_{15-8}$)→A$_{7-0}$ ZE	(A$_{7-0}$)→Y$_{15-8}$; Y31-16—Un Y7-0—Un
6	(Y$_{23-16}$)→A$_{7-0}$ ZE	(A$_{7-0}$)→Y$_{23-16}$; Y31-24—Un Y15-0—Un
7	(Y$_{31-24}$)→A$_{7-0}$ ZE	(A$_{7-0}$)→Y$_{31-24}$; Y23-0—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$_8$ and not S$_b$ for store cycle.

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

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
Y—Operand (Y) (Whole word or partial word) or Y, depending on k
⊙—Logical product (AND)
⊕—Logical sum (Inclusive OR)
⊕—Logical difference (Exclusive OR)