THE JOHNS HOPKINS UNIVERSITY
APPLIED PHYSICS LABORATORY

AN/UYK-20 COMPUTER REPERTOIRE OF INSTRUCTIONS

OCTAL CODE			DESCRIPTION	MNEMONIC/ULTRA FORMAT								CARRY/ OVERFLOW/ COND. CODE			SEE NOTE
				RR (f_1 = 0)		RI (f_1 = 1)		RK (f_1 = 2)		RX (f_1 = 3)					
f	a	m		MN	UF	MN	UF	MN	UF	MN	UF	C	OF	CC	
00	–	–	BYTE LOAD (Y_b) ← R_a	–	–	–	–	–	–	BL	3	0	0	X	3
01	–	–	LOAD R_a (Y) ← R_a	LR	2	LI	2	LK	3	L	3	0	0	X	–
02	–	0	MAKE POSITIVE R_a	PR	1	–	–	–	–	–	–	X	0	X	–
02	–	1	MAKE NEGATIVE R_a	NR	1	–	–	–	–	–	–	X	0	X	–
02	–	2	ROUND R_a, R_{a+1} bit 15 set	RR	1	–	–	–	–	–	–	X	X	X	–
02	–	4	TWO'S COMPLEMENT SINGLE R_a	TCR	1	–	–	–	–	–	–	X	X	X	–
02	–	5	TWO'S COMPLEMENT DOUBLE R_{a-a+1} TCDR		1	–	–	–	–	–	–	X	X	X	1
02	–	6	ONE'S COMPLEMENT SINGLE R_a	OCR	1	–	–	–	–	–	–	0	0	X	–
02	–	10	INCREASE R_a BY 1	IROR	1	–	–	–	–	–	–	X	X	X	–
02	–	11	DECREASE R_a BY 1	DROR	1	–	–	–	–	–	–	X	X	X	–
02	–	12	INCREASE R_a BY 2	IRTR	1	–	–	–	–	–	–	X	X	X	–
02	–	13	DECREASE R_a BY 2	DRTR	1	–	–	–	–	–	–	X	X	X	–
02	–	–	LOAD DOUBLE (Y, Y + 1) ← R_{a-a+1}	–	–	–	–	LDI	2	–	–	LD	3	0	X	X	1,2
03	–	0	EXEC RETURN, P + 1 ← R_a, CI II INT	ER	1	–	–	–	–	–	–	–	0	0	X	6
03	–	1	STORE STATUS REG 1 IN R_a	SSOR	1	–	–	–	–	–	–	–	0	0	X	–
03	–	2	STORE STATUS REG 2 IN R_a	SSTR	1	–	–	–	–	–	–	–	0	0	X	–
03	–	3	STORE RTC LOWER IN R_a	SCR	1	–	–	–	–	–	–	–	0	0	X	–
03	–	4	LOAD P WITH R_a	LPR	1	–	–	–	–	–	–	–	NC	NC	NC	–
03	–	5	LOAD STATUS REG 1 WITH R_a	LSOR	1	–	–	–	–	–	–	–	NC	NC	NC	–
03	–	6	LOAD STATUS REG 2 WITH R_a	LSTR	1	–	–	–	–	–	–	–	NC	NC	NC	–
03	–	7	LOAD RTC LOWER WITH R_a	LCR	1	–	–	–	–	–	–	–	NC	NC	NC	–
03	–	10	ENABLE RTC (O. F. INT ENAB)	ECR	1	–	–	–	–	–	–	–	NC	NC	NC	–
03	–	11	DISABLE RTC	DCR	1	–	–	–	–	–	–	–	NC	NC	NC	–
03	–	12	LOAD AND ENABLE MON CLOCK	LEM	1	–	–	–	–	–	–	–	NC	NC	NC	–
03	–	13	DISABLE MON CLOCK	DM	1	–	–	–	–	–	–	–	NC	NC	NC	–
03	–	14	LOAD RTC DOUBLE WITH R_{a-a+1}	LCRD	1	–	–	–	–	–	–	–	NC	NC	NC	–
03	–	15	STORE RTC DOUBLE IN R_{a-a+1}	SCRD	1	–	–	–	–	–	–	–	NC	NC	NC	–
03	–	16	ENABLE RTC INTERRUPT	ECIR	1	–	–	–	–	–	–	–	NC	NC	NC	–
03	–	17	DISABLE RTC INTERRUPT	DCIR	–	–	–	–	–	–	–	–	NC	NC	NC	–
03	–	–	LOAD MULTIPLE (Y_j) ← R_a THRU R_m	–	–	–	–	–	–	LM	3	NC	NC	NC	4
04	–	1	REVERSE REGISTER BITS OF R_a	RVR	1	–	–	–	–	–	–	0	0	X	–
04	–	2	COUNT ONES OF R_a, CNT ← R_{a+1}	CNT	1	–	–	–	–	–	–	NC	NC	NC	–
04	–	3	SCALE FACTOR SHIFT $R_{a,a+1}$ CNT R_{a-2}	SFR	1	–	–	–	–	–	–	NC	NC	NC	–
04	–	–	BYTE LOAD AND INDEX R_m BY 1	–	–	–	–	–	–	BLX	3	NC	NC	X	3
05	–	–	SET BIT m IN R_a	SBR	2	–	–	–	–	–	–	0	0	X	–
05	–	–	LOAD AND INDEX R_m BY 1	–	–	LXI	2	–	–	LX	3	0	0	X	–
06	–	–	ZERO BIT m IN R_a	ZBR	2	–	–	–	–	–	–	0	0	X	–
06	–	–	LOAD DOUBLE AND INDEX R_m BY 2	–	–	LDXI	2	–	–	LDX	3	0	0	X	1, 2
07	–	–	COMPARE BIT m OF R_a : 0	CBR	2	–	–	–	–	–	–	0	0	X	–
07	–	–	LOAD 3 PSWs WITH (Y_j)	–	–	LPI	5	–	–	LP	4	NC	NC	NC	4
10	–	–	LOGICAL RIGHT SINGLE SHIFT R_a	LRSR	2	–	–	LRS	3	–	–	0	0	X	–
10	–	–	BYTE STORE R_a ← (Y_b)	–	–	–	–	–	–	BS	3	NC	NC	NC	3
11	–	–	ALGEBRAIC RIGHT SINGLE SHIFT R_a	ARSR	2	–	–	ARS	3	–	–	0	0	X	–
11	–	–	STORE R_a ← Y	–	–	SI	2	–	–	S	3	NC	NC	NC	1
12	–	–	LOGICAL RIGHT DOUBLE SHIFT	LRDR	2	–	–	LRD	3	–	–	0	0	X	1
12	–	–	STORE DOUBLE R_{a-a+1} ← Y, Y + 1	–	–	SDI	2	–	–	SD	3	NC	NC	NC	1, 2
12	–	–	ALGEBRAIC RIGHT DOUBLE SHIFT	ARDR	2	–	–	ARD	3	–	–	0	0	X	1
13	–	–	STORE MULTIPLE R_a THRU R_m ← Y_j	–	–	–	–	–	–	SM	3	NC	NC	NC	4
14	–	–	ALGEBRAIC LEFT SINGLE SHIFT	ALSR	2	–	–	ALS	3	–	–	0	X	X	–
14	–	–	BYTE STORE AND INDEX R_m BY 1	–	–	–	–	–	–	BSX	3	NC	NC	NC	3
15	–	–	CIRCULAR LEFT SINGLE SHIFT	CLSR	2	–	–	CLS	3	–	–	0	0	X	–
15	–	–	STORE AND INDEX R_m BY 1	–	–	SXI	2	–	–	SX	3	NC	NC	NC	–
16	–	–	ALGEBRAIC LEFT DOUBLE SHIFT	ALDR	2	–	–	ALD	3	–	–	0	X	X	1
16	–	–	STORE DOUBLE AND INDEX R_m BY 2	–	–	SDXI	2	–	–	SDX	3	NC	NC	NC	1, 2
17	–	–	CIRCULAR LEFT DOUBLE SHIFT	CLDR	2	–	–	CLD	3	–	–	0	0	X	1
17	–	–	STORE ZEROS (Y)	–	–	SZI	5	–	–	SZ	4	NC	NC	NC	–
20	–	–	SUBTRACT R_a - (Y) ← R_a	SUR	2	SUI	3	SUK	3	SU	3	X	X	X	–
21	–	–	SUB DOUBLE R_{a-a+1} - (Y, Y + 1) ← R_{a-a+1}	SUDR	2	SUDI	2	–	–	SUD	3	X	X	X	1
22	–	–	ADD R_a + (Y) ← R_a	AR	2	AI	2	AK	3	A	3	X	X	X	–
23	–	–	ADD DOUBLE R_{a-a+1} + (Y, Y + 1) ← R_{a-a+1}	ADR	2	ADI	2	–	–	AD	3	X	X	X	1
24	–	–	COMPARE R_a : (Y)	CR	2	CI	2	CK	3	C	3	X	X	X	–
25	–	–	COMPARE DOUBLE R_{a-a+1} : (Y, Y + 1)	CDR	2	CDI	2	–	–	CD	3	X	X	X	1, 2
26	–	–	MULTIPLY R_a + (Y) ← R_{a-a+1}	MR	2	MI	2	MK	3	M	3	0	0	X	1
27	–	–	DIVIDE R_{a-a+1} ÷ (Y) ← R_{a+1}	DR	2	DI	2	DK	3	D	3	0	X	X	1, 5

OCTAL CODE			DESCRIPTION	MNEMONIC/ULTRA FORMAT								CARRY/ OVERFLOW/ COND. CODE			SEE NOTE
				RR (f_1 = 0)		RI (f_1 = 1)		RK (f_1 = 2)		RX (f_1 = 3)					
f	a	m		MN	UF	MN	UF	MN	UF	MN	UF	C	OF	CC	
30	–	–	AND R_a ⊙ (Y) ← R_a	ANDR	2	ANDI	2	ANDK	3	AND	3	0	0	X	–
31	–	–	OR R_a ⊕ (Y) ← R_a	ORR	2	ORI	2	ORK	3	OR	3	0	0	X	–
32	–	–	EXCLUSIVE OR R_a ⊕ (Y) ← R_a	XORR	2	XORI	2	XORK	3	XOR	3	0	0	X	–
33	–	–	MASKED SUBSTITUTE R_{a+1} ⊙ Y ⊕ R_{a+1} ⊙ R_a ← R_a	MSR	2	MSI	2	MSK	3	MS	3	0	0	X	1
34	–	–	COMPARE MASKED R_a ⊙ R_{a+1} : (Y) ⊙ R_{a+1}	CMR	2	CMI	2	CMK	2	CM	3	0	0	X	–
35	–	–	EXEC IO CMD AT ADR 140, 141 AND 0 ← $2^{14, 15}$ OF ADDR 140	IOCR	8	–	–	–	–	–	–	NC	NC	NC	–
35	–	–	BIASED FETCH, SET $2^{14, 15}$ OF Y	–	–	BFI	5	–	–	BF	4	0	0	X	–
35	–	–	EXECUTE REMOTE (Y), P + 2 ← P	–	–	–	–	REX	4	–	–	NC	NC	NC	–
37	–	0	VECTOR FN TRIG	VF	2	–	–	–	–	–	–	NC	NC	NC	–
37	–	1	ROTATE FN TRIG	RF	2	–	–	–	–	–	–	NC	NC	NC	–
37	–	2	VECTOR FN TRIG W/PRESCALE	VFP	2	–	–	–	–	–	–	NC	NC	NC	–
37	–	3	ROTATE FN TRIG W/PRESCALE	RFP	2	–	–	–	–	–	–	NC	NC	NC	–
37	–	4	VECTOR FN HYP	VH	2	–	–	–	–	–	–	NC	NC	NC	–
37	–	5	ROTATE FN HYP	RH	2	–	–	–	–	–	–	NC	NC	NC	–
37	–	6	VECTOR FN HYP W/POSTSCALE	VHP	2	–	–	–	–	–	–	NC	NC	NC	–
37	–	7	ROTATE FN HYP W/POSTSCALE	RHP	2	–	–	–	–	–	–	NC	NC	NC	–
40	–	–	JUMP CC ZERO/EQUAL	JER	5	–	–	JE	4	JE	6	NC	NC	NC	–
40	–	1	JUMP CC NOT ZERO/NOT EQUAL	JNER	5	–	–	JNE	4	JNE	6	NC	NC	NC	–
40	–	2	JUMP CC POS/GREATER THAN OR EQUAL	JLSR	5	–	–	MLS	4	JLS	6	NC	NC	NC	–
40	–	3	JUMP CC NEG/LESS THAN	JGER	5	–	–	JGE	4	JGE	6	NC	NC	NC	–
40	–	4	JUMP CON OVERFLOW SET	JOR	5	–	–	JO	4	JO	6	NC	NC	NC	–
40	–	5	JUMP ON CARRY SET	JCR	5	–	–	JC	4	JC	6	NC	NC	NC	–
40	–	6	JUMP POWER OUT OF TOL.	JPTR	5	–	–	JPT	4	JPT	6	NC	NC	NC	–
40	–	7	JUMP BOOTSTRAP 2 SELECTED	JBR	5	–	–	JB	4	JB	6	NC	NC	NC	–
40	–	10	JUMP	JR	5	–	–	J	4	J	6	NC	NC	NC	–
40	–	11	JUMP, STOP	JSR	5	–	–	JS	4	JS	6	NC	NC	NC	11
40	–	12	JUMP, STOP IF KEY 1 SET	JKSR	9	–	–	JKS	11	JKS	13	NC	NC	NC	11
40	–	13	JUMP, STOP IF KEY 2 SET	JKSR	10	–	–	JKS	12	JKS	14	NC	NC	NC	11
40	–	–	LOCAL JUMP TO Y	–	–	LJ	–	–	–	–	–	NC	NC	NC	7
41	–	–	INDEX JUMP, IF R_a ≠ 0 THEN R_a − 1 ← R_a AND JUMP, ELSE N. I.	XJR	2	–	–	XJ	3	XJ	7	NC	NC	NC	7
41	–	–	LOCAL JUMP INDIRECT TO (Y)	–	–	LJI	17	–	–	–	–	NC	NC	NC	7
42	–	–	JUMP AND LINK REG P + i ← R_a	JLRR	2	–	–	JLR	3	JLR	7	NC	NC	NC	8
43	–	–	LOCAL JUMP AND LINK MEMORY	–	–	LJLM	17	–	–	–	–	NC	NC	NC	7
43	–	–	JUMP AND LINK MEMORY P + i ← R_a	–	–	–	–	JLM	3	JLM	7	NC	NC	NC	8
44	–	–	JUMP REG R_a ZERO	JZR	2	–	–	JZ	3	JZ	7	NC	NC	NC	–
44	–	–	LOCAL JUMP EQUAL	–	–	LJE	17	–	–	–	–	NC	NC	NC	7
45	–	–	JUMP REG R_a NOT ZERO	JNZR	2	–	–	JNZ	3	JNZ	7	NC	NC	NC	–
45	–	–	LOCAL JUMP NOT EQUAL	–	–	LJNE	17	–	–	–	–	NC	NC	NC	7
46	–	–	JUMP REG R_a POSITIVE	JPR	2	–	–	JP	3	JP	7	NC	NC	NC	–
46	–	–	LOCAL JUMP GRT THAN OR EQUAL	–	–	LJGE	17	–	–	–	–	NC	NC	NC	7
47	–	–	JUMP REG R_a NEGATIVE	JNR	2	–	–	JN	3	JN	7	NC	NC	NC	–
47	–	–	LOCAL JUMP LESS THAN	–	–	LJLS	17	–	–	–	–	NC	NC	NC	7
54	–	–	LOAD ADDR REG, MULT (Y_j) ← PAGE [R_a]	LARR	2	LARI	2	–	–	–	–	NC	NC	NC	–
54	–	–	LOAD ADDR REG MULT (Y_j) ← PAGE [i]	–	–	–	–	–	–	LARM	3	NC	NC	NC	9
55	–	–	STORE ADDRESS REG, PAGE [R_a] ← Y	SARR	2	SARI	2	–	–	–	–	NC	NC	NC	–
55	–	–	STORE ADDR REG, MULT PAGE i ← Y_j	–	–	–	–	–	–	SARM	3	NC	NC	NC	–
60	–	–	LOGICAL RIGHT SINGLE SHIFT R_a	LLRS	2	(RL FORMAT) f_1 = 0						0	0	X	–
60	–	–	ALGEBRAIC RIGHT SINGLE SHIFT	LARS	2	(RL FORMAT) f_1 = 1						0	0	X	–
60	–	–	LOGICAL RIGHT DOUBLE SHIFT	LLRD	2	(RL FORMAT) f_1 = 2						0	0	X	1
60	–	–	ALGEBRAIC RIGHT DOUBLE SHIFT	LARD	2	(RL FORMAT) f_1 = 3						0	0	X	1
61	–	–	LOGICAL LEFT SINGLE SHIFT	LLLS	2	(RL FORMAT) f_1 = 0						0	X	X	–
61	–	–	CIRCULAR LEFT SINGLE SHIFT	LCLS	2	(RL FORMAT) f_1 = 1						0	0	X	–
61	–	–	ALGEBRAIC LEFT DOUBLE SHIFT	LALD	2	(RL FORMAT) f_1 = 2						0	X	X	1
61	–	–	CIRCULAR LEFT DOUBLE SHIFT	LCLD	2	(RL FORMAT) f_1 = 3						0	0	X	1
62	–	–	SUBTRACT R_a − m ← R_a	LSU	2	(RL FORMAT) f_1 = 0						X	X	X	–
62	–	–	SUBTRACT DOUBLE $R_{a,a+1}$ − m ← $R_{a,a+1}$	LSUD	2	(RL FORMAT) f_1 = 1						X	X	X	1
62	–	–	ADD R_a + m	LA	2	(RL FORMAT) f_1 = 2						X	X	X	–
62	–	–	ADD DOUBLE R_{a-a+1} + m R_{a-a+1}	LAD	2	(RL FORMAT) f_1 = 3						X	X	X	1
63	–	–	LOAD m ← R_a	LL	2	(RL FORMAT) f_1 = 0						0	0	0	–
63	–	–	COMPARE m : R_a	LC	2	(RL FORMAT) f_1 = 1						X	X	X	–
63	–	–	MULTIPLY m + R_a + 1 ← R_{a-a+1}	LMUL	2	(RL FORMAT) f_1 = 2						0	X	X	1
63	–	–	DIVIDE R_{a-a+1} ÷ m ← R_{a+1}	LDIV	2	(RL FORMAT) f_1 = 3						0	X	X	1, 5

OCTAL CODE			DESCRIPTION	MNEMONIC/ULTRA FORMAT								CARRY/ OVERFLOW/ COND. CODE			SEE NOTE
				RR (f_1 = 0)		RI (f_1 = 1)		RK (f_1 = 2)		RX (f_1 = 3)					
f	a	m		MN	UF	MN	UF	MN	UF	MN	UF	C	OF	CC	
64	–	–	BYTE SUBTRACT R_a − (Y_b) ← R_a	–	–	–	–	–	–	BSU	3	X	X	X	3
65	–	–	BYTE ADD R_a + (Y_b) ← R_a	–	–	–	–	–	–	BA	3	X	X	X	3
66	–	–	BYTE COMPARE R_a : Y_b	–	–	–	–	–	–	BC	3	X	X	X	3
67	–	–	RESERVED FOR USER MACRO	UM1	2	–	–	–	–	–	–	–	–	–	–
67	–	–	BYTE COMPARE AND INDEX R_m BY 1	–	–	–	–	–	–	BCX	3	X	X	X	3
70	–	–	IO MASTER CLEAR (CHN, BFR, EI, & EIE)	ACR	4	–	–	–	–	–	–	NA	NA	NA	12
70	–	4	ENABLE ALL EXT INT DATA, SET EIE	ACR	4	–	–	–	–	–	–	NA	NA	NA	12
70	–	5	DISABLE ALL EXT INT, CLEAR EIE	ACR	4	–	–	–	–	–	–	NA	NA	NA	12
70	–	6	ENABLE ALL IO MONITORS	ACR	4	–	–	–	–	–	–	NA	NA	NA	12
70	–	7	DISABLE ALL IO MONITORS	ACR	4	–	–	–	–	–	–	NA	NA	NA	12
70	–	10	MASTER CLEAR CHAN. a	CCP	2	–	–	–	–	–	–	NA	NA	NA	12
70	–	14	ENABLE CHAN. a EXT. INT DATA	CCR	2	–	–	–	–	–	–	NA	NA	NA	12
70	–	15	DISABLE CHAN. a EXT. INT DATA	CCR	2	–	–	–	–	–	–	NA	NA	NA	12
70	–	16	ENABLE CHAN a IO MON.	CCR	2	–	–	–	–	–	–	NA	NA	NA	12
70	–	17	DISABLE CHAN a IO MON.	CCR	2	–	–	–	–	–	–	NA	NA	NA	12
70	0	NA	INITIATE INPUT TRANSFER	–	–	–	–	–	–	IO	18	NA	NA	NA	2, 13
	1	NA	INITIATE OUTPUT TRANSFER	–	–	–	–	–	–	IO	18	NA	NA	NA	2, 13
	2	NA	INITIATE EXF W/O FORCE	–	–	–	–	–	–	IO	18	NA	NA	NA	2, 13
	3	NA	INITIATE EXF W/FORCE	–	–	–	–	–	–	IO	18	NA	NA	NA	2, 13
71	–	2	INITIATE INPUT CHAIN Y ← CAP (COMM)	–	–	–	–	ICK	18	–	–	NA	NA	NA	–
71	–	6	INITIATE OUTPUT CHAIN (COMM) Y ← CAP	–	–	–	–	OCK	18	–	–	NA	NA	NA	–
71	–	–	LOAD CONT MEM, Y ← CM [m] (CHAIN)	–	–	–	–	LCMK	19	LCM	19	NA	NA	NA	–
71	–	–	WRITE (LOAD) CONTROL MEMORY Y ← CM [am] (COMM)	–	–	–	–	–	–	WIM or WCM	3	NA	NA	NA	–
72	–	–	READ (STORE) CONTROL MEMORY CM [am] ← Y	–	–	–	–	–	–	RIM or RCM	3	NA	NA	NA	–
72	–	–	STORE CONTROL MEMORY CM [m] ← Y CHAIN	–	–	–	–	–	–	SCM	19	NA	NA	NA	–
73	–	0	HALT (CHAIN)	HCR	8	–	–	–	–	–	–	–	NA	NA	–
73	–	1	INTERRUPT (CHAIN)	IPR	8	–	–	–	–	–	–	–	NA	NA	–
73	–	0	CLEAR FLAG (CHAIN) $2^{14, 15}$ OF Y	–	–	–	–	–	–	ZF	17	–	NA	NA	–
73	–	1	SET FLAG (CHAIN) $2^{14, 15}$ OF Y	–	–	–	–	–	–	SF	17	–	NA	NA	–
74	–	–	CONDITIONAL JUMP (CHAIN) TO Y	–	–	–	–	SJMC	18	–	–	–	NA	NA	–
75	–	–	SEARCH FOR SERIAL SYNC/SET MON/ SET SUPP (CHAIN)	SFSC	5	–	–	–	–	–	–	–	NA	NA	–
76	–	–	SET/CLEAR DISCRETES (COMM)	SICR	2	–	–	–	–	–	–	NA	NA	NA	–
76	–	–	SET/CLEAR DISCRETES (CHAIN)	CSIR	5	–	–	–	–	–	–	NA	NA	NA	–
77	–	–	STORE SERIAL STATUS (COMM)	–	–	–	–	–	–	SST	18	NA	NA	NA	–
77	–	–	STORE SERIAL STATUS (CHAIN)	–	–	–	–	–	–	CSST	17	NA	NA	NA	–

NOTES:

1. REG # MUST BE EVEN
2. Y MUST BE EVEN
3. 2^0 = 1: LOWER BYTE
   2^0 = 0: UPPER BYTE
   Y = R_m/2 + y
4. Y_j = DENOTES CONSECUTIVE ADDR.
5. REMAINDER IN R_a, SIGN SAME AS DIVISOR
6. NOOP IF CL II INT LOCKOUT SET
7. RI TYPE 1, Y MUST BE WITHIN ± 128
8. FOR RK & RX, i = 2; FOR RI, i = 1
9. (R_a) BITS 5-0 DEFINE STARTING PAGE, (R_a) BITS 13-8 DEFINE COUNT-1
10. WITH CMS-2Y ASSEMBLER USE LDCR
11. STOP OCCURS BEFORE JUMP
12. USER MUST SUPPLY m DESIGNATOR
13. Y ← BCW, Y + 1 BAP (CHAIN)

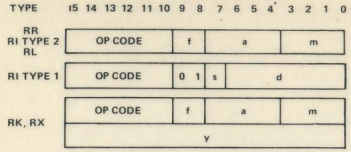
ULTRA-16 PSEUDO CODES

NOP	GENERATES NO-OP (30 CODE)
NOPD	GENERATES DOUBLE WORD NO-OP
ORIG X	SETS INST COUNTER TO X
BSS X	RESERVES X WORDS OF CORE
BCW TM,O,B,CNT, Y	DEFINE BCW & BA
LCMD R_a, Y	LOADS IO CMD LOC 140,141
DEC X,S	DECIMAL NO. X SCALED BY S
ODD	NEXT ADDRESS ODD
EVEN	NEXT ADDRESS EVEN
BAM A, S	SCALED ANGLE IN BAMS
RAD A, S	SCALED ANGLE IN RADIANS
LLT	LIST LABELS
LCR	LIST CROSS REF
NUM X	DECIMAL NUMBER

ULTRA CODING FORMATS (UF) (* y DENOTES INDIRECT)

1. a	8. None	14. 2, *y, m
2. a, m	9. 1, m	15. am, y
3. a, y, m	10. 2, m	16. j, y, x
4. y, m	11. 1, y, m	17. y
5. m	12. 2, y, m	18. a, y
6. *y, m	13. 1, *y, m	19. m, y
7. a, *y, m		

INSTRUCTION WORD FORMATS



DEFINITION OF FIELDS

OP CODE: CODE SPECIFYING THE OPERATION

f: FORMAT DESIGNATOR
- f = 00 FORMAT RR-REGISTER OR RL-1 FORMAT
- = 01 FORMAT RI-REGISTER-IMMEDIATE MEMORY OR RL-2 FORMAT
- = 10 FORMAT RK-REGISTER-LITERAL CONSTANT OR RL-3 FORMAT
- = 11 FORMAT RX-REGISTER-INDEXED ADDRESS OR CONSTANT OR RL-4 FORMAT

a: GENERAL REGISTER OR SUBFUNCTION DESIGNATOR

m: GENERAL REGISTER OR SUBFUNCTION DESIGNATOR

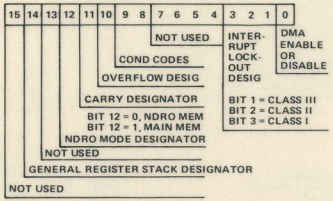
m: 4-BIT UNSIGNED LITERAL IN RL FORMAT

d: DEVIATION (2s COMPLEMENT)

s: SIGN DESIGNATOR FOR d
- 0 = POSITIVE
- 1 = NEGATIVE

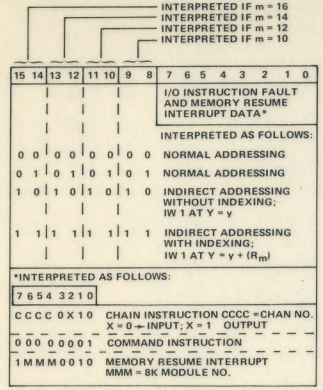
y: ADDRESS OR ARITHMETIC CONSTANT

STATUS REGISTER NO. 1 FORMAT



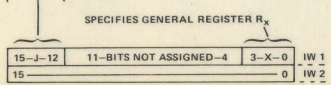
COND CODE		INDICATED RESULTS OF	
BIT 9	BIT 8	ARITH OP	COMPARE OP
0	0	ZERO	$R_a \geqslant R_m$ OR Y
0	1	NOT ZERO AND POS	$R_a > R_m$ OR Y
1	0	NOT USED	NOT USED
1	1	NOT ZERO AND NEG	$R_a < R_m$ OR Y

STATUS REGISTER NO. 2 FORMAT

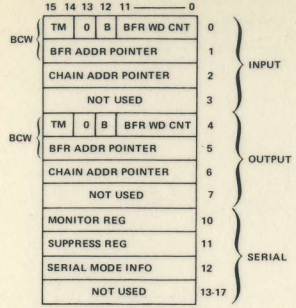


INDIRECT WORD INTERPRETATION

OCTAL J-VALUE	ADDRESS DETERMINATION
0	OPERAND AT ADDR SPEC BY (IW 2)
1	OPERAND AT ADDR SPEC BY (IW 2) + (R_x)
2	OPERAND AT ADDR SPEC BY (IW 2) + (R_m)
3	OPERAND AT ADDR SPEC BY (IW 2) + $(R_m + 1)$
4	CASCADED IW AT ADDR SPEC BY (IW 2)
5	CASCADED IW AT ADDR SPEC BY (IW 2) + (R_x)
6	CASCADED IW AT ADDR SPEC BY (IW 2) + (R_m)
7	CASCADED IW AT ADDR SPEC BY (IW 2) + $(R_m + 1)$
10-17	NOT ASSIGNED



I/O CHANNEL CONTROL MEMORY



TM = 00: ABORT TRANS (INPUT ONLY)
TM = 01: TRANS 8-BIT BYTES
TM = 10: TRANS 16-BIT WORDS
TM = 11: TRANS 32-BIT WDS
B = 0: MOST SIG 8-BITS
B = 1: LEAST SIG 8-BITS

ASSIGNED MEMORY ADDRESSES

ASSIGNMENT	ADDRESSES (OCTAL)
NDRO MEMORY	00-77 AND 300-477
FOR PROCESSING	
CLASS III INTERRUPTS	110-117
CLASS II INTERRUPTS	120-127
CLASS I INTERRUPTS	130-137
FOR IOC OPERATION	
COMMAND CELLS	140-141
EXTERNAL INTERRUPT WORD STORAGE (IOC)	200-217

FUNCTION	ADDR ASSIGNMENT TO CLASS III	II	I
STORES P AT ADDR	110	120	130
STORES SR NO. 1 AT ADDR	111	121	131
STORES SR NO. 2 AT ADDR	112	122	132
STORES RTC LOWER AT ADDR	113	123	133
RELOADS P WITH INDEX + CONTENTS OF ADDR	114	124	134
RELOADS SR NO. 1 FROM ADDR	115	125	135
RELOADS SR NO. 2 FROM ADDR	116	126	136
STORES RTC UPPER AT ADDR	117	127	137

INTERRUPT ENTRANCES

CLASS	PRIORITY WITHIN CLASS	INTERRUPT	INT ENTR ADDR INDEX (BINARY)
I	1	POWER FAULT	0
	2	MEMORY RESUME	10
II	1	CP INST FAULT	0
	2	I/O CHAIN INST FAULT	10
	3	NOT ASSIGNED	100
	4	EXEC RETURN	110
	5	RTC OVERFLOW	1000
	6	MONITOR CLOCK	1010
III	1	INTERCOM TIME-OUT	CCCC110
	2	EXT INT OR DISCRETE INT*	CCCC000
	3	OUTPUT CHAIN INT	CCCC100
	4	INPUT CHAIN INT	CCCC010

*SERIAL MIL-STD-188C OR EIA-STD-RS 232C CHANNELS