

UNIVAC 1218 COMPUTER

Repertoire of Instructions

	f m	SYMBOL	INSTRUCTION	DESCRIPTION	EXEC TIME
LOGICAL	#02*	CMAL	COMPARE SET DESIGNATOR	(AL):(Y); (AL) _f = (AL) _i	8
	#06*	CMSK	COMPARE WITH MASK SET DES.	L(AL)(AU):L(Y)(AU); (A) _f =(A) _i	8
	#04*	SLSU	SELECTIVE SUBSTITUTE	L(AU) [*] (AL)+L(AU)(Y)→AL; (AU) _f =(AU) _i	8
	51	SLSET	SELECTIVE SET (INCLUSIVE OR)	(AL)v(Y)→AL; SET (AL) _n FOR (Y) _n = 1	8
	52	SLCL	SELECTIVE CLEAR (LOGICAL PROD)	L(AL)(Y)→AL; CLEAR (AL) _n FOR (Y) _n = 0	8
	53	SLCP	SELECTIVE COMPLEMENT (EXCL. OR)	(AL)⊕(Y)→AL; COMPLEMENT (AL) _n FOR (Y) _n =1	8
	50 61	CPAL	COMPLEMENT AL	(AL) [*] →AL IGNORE k	5.33
	50 62	CPAU	COMPLEMENT AD	(AU) [*] →AU IGNORE k	5.33
	50 63	CPA	COMPLEMENT A	(A) [*] →A IGNORE k	5.33
	SHIFT	50 41	RSHAU	RIGHT SHIFT AU	} SHIFT (REG.) RIGHT k BIT POSITIONS END OFF & FILL THE UPPER k BITS WITH INITIAL SIGN
50 42		RSHAL	RIGHT SHIFT AL		
50 43		RSHA	RIGHT SHIFT A		
50 44		SF	SCALE FACTOR SHIFT	LEFT CIRCULAR SHIFT A UNTIL A ₃₅ ≠A ₃₄ OR k - SHIFTCOUNT = 0 THEN k - SHIFTCOUNT → 00017	9.33+2k/3 (k≠0) 8 (k=0)
50 45		LSHAU	LEFT SHIFT AU	} SHIFT (REG.) LEFT k BIT POSITIONS CIRCULARLY	4 μsec (k=0); 5.33 +2k/3 μsec (k≠0)
50 46		LSHAL	LEFT SHIFT AL		
50 47	LSHA	LEFT SHIFT A			
SKIP ARITHMETIC I/O	50 51	SKPNO	SKIP ON NO BORROW	SKIP NI IF BORROW DESIGNATOR NOT SET	6 4.67
	50 52	SKPOV	SKIP ON OVERFLOW	SKIP NI IF OVERFLOW DESIGNATOR SET	6 4.67
	50 53	SKPNO	SKIP ON NO OVERFLOW	SKIP NI IF OVERFLOW DESIGNATOR NOT SET	6 4.67
	50 54	SKPODD	SKIP ON ODD PARITY	SKIP NI IF SUM OF ONES IN L(AU)(AL) IS ODD	6 4.67
	50 55	SKPEVN	SKIP ON EVEN PARITY	SKIP NI IF SUM OF ONES IN L(AU)(AL) IS EVEN	6 4.67
	50 21	SKPIIN	SKIP ON INPUT INACTIVE	SKIP NI IF CHAN. k INPUT IS INACTIVE	6 4.67
	50 22	SKPOIN	SKIP ON OUTPUT INACTIVE	SKIP NI IF CHAN. k OUTPUT IS INACTIVE	6 4.67
	50 23	SKPFIN	SKIP ON EXT. FUNCT. INACTIVE	SKIP NI IF CHAN. k EXT. FUNCT. IS INACTIVE	6 4.67
	50 50	SKP	SKIP ON KEY SETTING k	SKIP NI IF k = CONSOLE KEY SETTING	6 4.67
	50 57	SKPNR	SKIP ON NO RESUME	SKIP NI IF RESUME DESIGNATOR NOT SET	6 4.67
INPUT / OUTPUT	50 11	IN	INPUT TRANSFER	(P+1)→60+2k; (P+2)→61+2k; SET IN. ACTIVE	20
	50 12	OUT	OUTPUT TRANSFER	(P+1)→40+2k; (P+2)→41+2k; SET OUT. ACTIVE	20
	50 13	EXF	EXT. FUNCT. TRANSFER	(P+1)→20+2k; (P+2)→21+2k; SET EXT. F. ACTIVE	20
	50 15	INSTP	TERMINATE INPUT	CLEAR INPUT ACTIVE CHAN. k	4
	50 16	OUTSTP	TERMINATE OUTPUT	CLEAR OUTPUT ACTIVE CHAN. k	4
	50 17	EXFSTP	TERMINATE EXT. FUNCT.	CLEAR FUNCTION ACTIVE CHAN. k	4
	50 20	SRSM	SET RESUME	SET THE RESUME DESIGNATOR (INTERCOMPUTER)	4
	50 26	OUTOV	OUTPUT OVERRIDE	FORCE ONE WORD OUT CHAN. k WITH OUTPUT ACK.	4.67
	50 27	EXFOV	FUNCTION OVERRIDE	FORCE ONE WORD OUT CHAN. k WITH EXT. FUNCT.	4.67
	50 30	RIL	REMOVE INTERRUPT LOCKOUT	ENABLE ALL INTERRUPTS	4
	50 32	RIL	REMOVE EXT. INT. LOCKOUT	ENABLE EXTERNAL INTERRUPTS	4
	50 34	SIL	SET INTERRUPT LOCKOUT	DISABLE ALL INTERRUPTS	4
	50 36	SIL	SET EXT. INT. LOCKOUT	DISABLE EXTERNAL INTERRUPTS	4
	50 24	WIFI	WAIT FOR INTERRUPT	STOP; THEN INTERRUPT ENTRANCE REG. FOR NI	4
50 56	STOP	STOP ON KEY SETTING	STOP IF k = KEY SETTING	4.67	
FAULT	00, 01, 77	FAULT	FAULT = 4 usec		
	50 00, 50 01, 50 77	FAULT	FAULT = 4 usec		

ASSIGNED CORE MEMORY LOCATIONS (OCTAL)

000	FAULT ENTRANCE REGISTER	060-077	INPUT BUFFER CONTROL
001-010	INDEX REGISTERS	100-117	EXTERNAL INTERRUPT ENTRANCE
016	SYNCHRONIZATION INTERRUPT ENTRANCE	120-137	EXT. FUNCT. TERMINATION INTERRUPT
017	SCALE FACTOR SHIFT COUNT	140-157	OUTPUT TERMINATION INTERRUPT
020-037	EXTERNAL FUNCTION BUFFER CONTROL	160-177	INPUT TERMINATION INTERRUPT
040-057	OUTPUT BUFFER CONTROL	200-237	BOOTSTRAP PROGRAM

UNIVAC 1218 COMPUTER

Repertoire of Instructions

INSTRUCTION WORD FORMAT I 17-f-12 11—u—0 FORMAT II 17-f-12 11-m-6 5—k—0

	f m	SYMBOL	INSTRUCTION	DESCRIPTION	EXEC TIME
ENTER	#10*	ENTAU	ENTER AU WITH (Y)	(Y) → AU	8
	#12*	ENTAL	ENTER AL WITH (Y)	(Y) → AL	8
	#32*	ENTB	ENTER B REGISTER WITH (Y)	(Y) → B REGISTER	12
	36*	ENTBK	ENTER B REGISTER WITH CONSTANT	Y → B REGISTER (Note 1)	8
	70	ENTALK	ENTER AL WITH CONSTANT	Y → AL (Note 1)	4, 67
	50 72	ENTICR	ENTER INDEX CONTROL REGISTER	k ₂₋₀ → ICR	4
50 73	ENTSR	ENTER SR	k ₃₋₀ → SR	4	
STORE	#40*	CL	STORE ZERO (CLEAR Y)	0 → Y	8
	#42*	STRB	STORE B IN Y	B → Y	12
	#44*	STRAL	STORE (AL) IN Y	(AL) → Y	8
	#46*	STRAU	STORE (AU) IN Y	(AU) → Y	8
	72	STRICR	STORE (ICR) IN Y _L	(ICR) → Y ₃₋₀ ; (Y) ₁₇₋₆ UNCHANGED	8
	74	STRADR	STORE ADDRESS IN Y _L	(AL) ₁₁₋₀ → Y ₁₁₋₀ ; (Y) ₁₇₋₁₂ UNCHANGED	8
75	STRSR	STORE (SR) IN Y _L	(SR) → Y ₃₋₀ ; (Y) ₁₇₋₆ UNCHGD. 0 → SR ₃	8	
ARITHMETIC	#14*	ADDAL	ADD (Y) TO AL	(AL) + (Y) → AL	8
	#16*	SUBAL	SUBTRACT (Y) FROM AL	(AL) - (Y) → AL	8
	#20*	ADDA	ADD (Y+1, Y) TO A	(A)+(Y+1, Y) → A	12
	#22*	SUBA	SUBTRACT (Y+1, Y) FROM A	(A)-(Y+1, Y) → A	12
	#24*	MULAL	MULTIPLY AL BY (Y)	(AL) · (Y) → A	26-48, 67
	#26*	DIVA	DIVIDE A BY (Y)	(A) ÷ (Y) → AL; Remainder → AU	48
	71	ADDALK	ADD CONSTANT TO AL	(AL)+Y → AL, Note 1	4, 67
	50 60	RND	ROUND AU	{ (AU)+AL ₁₇ → AL; OR } IGNORE k { -(AU)-AL ₁₇ → AL; } AU _f = AU _i	5, 33 5, 33
MODIFYING	37	ENTBKB	MODIFY B WITH CONSTANT	B+Y → B REGISTER	12
	56	BSK	B SKIP	{ IF B = (Y), SKIP NI } { IF B ≠ (Y), B+1 → B REGISTER & DO NI }	16
	57	ISK	INDEX SKIP	{ IF (Y)=0, SKIP NI } { IF (Y) ≠ 0, (Y)-1 → Y & DO NI }	12
	73	BJP	B JUMP	{ IF B ≠ 0, B-1 → B REGISTER AND Y → P } { IF B=0, DO NI }	12
UNCOND.	30*	IRJP	INDIRECT RETURN JUMP	(P)+1 → (Y); (Y)+1 → P (Note 2)	12
	34*	JP	UNCONDITIONAL JUMP	Y → P	4
	54	IJPEI	IND JUMP & ENABLE INT.	(Y) → P & ENABLE INTERRUPTS	8
	55	IJP	INDIRECT JUMP	(Y) → P	8
	76	RJP	DIRECT RETURN JUMP	(P)+1 → Y; Y+1 → P (Note 2)	8
JUMP CONDITIONAL	JUMP TO Y (Y → P) IF COMPARISON DESIGNATOR IS;			EXEC TIME	LEGEND Y = y ⊕ y+B y = u _p ⊕ u _{SR} ⊕ u B = CONTENTS OF INDEX REG. NI = NEXT INST.
	NOT SET AND		SET AND		
	60	JPAUZ (AU) = 0	JPEQ M = (AL)	4	
	61	JPALZ (AL) = 0		4	
	62	JPAUNZ (AU) ≠ 0	JPNOT M ≠ (AL)	4	
	63	JPALNZ (AL) ≠ 0		4	
	64	JPAUP (AU) POSITIVE	JPMLEQ M ≤ (AL)	4	
	65	JPALP (AL) POSITIVE		4	
	66	JPAUNG (AU) NEGATIVE	JPMGR M > (AL)	4	
	67	JPALNG (AL) NEGATIVE		4	

* SR SENSITIVE

* B MODIFICATION of "y" REQUESTED: ADD SUFFIX "B" TO SYMBOL and "1" to f-CODE;
ADD 4 usec to EXECUTION TIME

NOTES: 1. For f = 36, 37, 70, 71 y=u extended with sign to 18 bits.
2. RETURN JUMPS, executed from Interrupt Entrance Registers, Store (P)