

1218 COMPUTER INSTRUCTION REPERTORY

INSTRUCTION WORD FORMAT: I 17-f-12 11-m-6 5-k-0 Y = y or y+(B)ICR
II 17-f-12 11-m-6 5-k-0 Y = u or u_p or SR

ENTER	STORE	ARITHMETIC	MODIFY	LOGICAL	SHIFT	SKIP	INPUT/OUTPUT	MISC	UNCOND.	JUMP	CONDITIONAL
#10*	ENTAU	Enter AU with (Y)	(Y) → AU	8							
#12*	ENTAL	Enter AL with (Y)	(Y) → AL	8							
#32*	ENTB	Enter B with (Y)	(Y) → B	12							
#36*	ENTBK	Enter B with constant	Y → B (See Note 1)	8							
70	ENTALK	Enter AL with constant	Y → AL (See Note 1)	4,67							
50	72 ENTICR	Enter Index Control Reg	k _{3,0} → ICR	4							
50	73 ENTSR	Enter Special Register	k _{3,0} → SR	4							
#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 _{5,0} ; Y _{17,6} unchanged; if (ICR) = 0, 000 → Y _{5,3} ; if (ICR) = 0, 001 → Y _{5,3}	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} ; 0 → Y _{5,4} ; (Y)17-6 unchanged; 0 → SR ₃	8							
#14*	ADDAL	Add (Y) to (AL)	(AL)+(Y) → AL	8							
#16*	SUBAL	Subtract (Y) from (AL)	(AL)-(Y) → AL	8							
#20*	ADDBN	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); Quot → AL, Rem → AU	48							
71	ADDAKAL	Add constant to (AL)	(AL)+Y → AL, (See Note 1)	4,67							
50	60 RND	Round AU	If (A)20, (AU)(AL)17 → AL, (AU) ₇ = (AU) ₁ If (A)0, (AU)-Comp (AL)17 → AL, (AU) ₇ = (AU) ₁	5,33							
37	ENTBKB	Modify (B) with constant	(B)+Y → B (See Note 1)	12							
65	BSK	B skip	If (B)=(Y), skip NI	16							
57	ISK	Index skip	If (B)≠(Y), (B)-1 → B, read NI If (Y)=0, skip NI	12							
73	BJP	B jump	If (Y)≠0, (Y)-1 → Y, read NI If (B)≠0, (B)-1 → B, Y → P If (B)=0, read NI	12							
#02*	CMAL	Compare and set designator	(AL):(Y); (AL)≠(AL) ₁	8							
#06*	CMSK	Comp. with mask and set des.	L(AL)(AU):L(Y)(AU) ₇ (A) ₇ ≠(A) ₇	8							
#04*	SLSU	Selective substitute	L(AU) ₇ (AL)+L(AU)(Y) → AL; (AU) ₇ ≠(AU) ₁	8							
51	SLEST	Selective set (incl. or)	(AL)(Y) → AL; set AL ₁ for (Y) ₁ = 1	8							
53	SLECL	Selective clear (log. prod)	(AL)(Y) → AL; clear AL ₁ for (Y) ₁ = 0	8							
53	SLCP	Selective compl (excl. or)	(AL) ⊕ (Y) → AL; complement (AL) _n for (Y) _n = 1	8							
50	61 CPAL	Complement (AL)	(AL) ₇ → AL	5,33							
50	62 CPAU	Complement (AU)	(AU) ₇ → AU	5,33							
50	63 CPA	Complement (A)	(A) ₇ → A	5,33							
50	41 RSHAU	Right shift (AU)	Shift (AU) right by k; sign fill	5,33							
50	42 RSHAL	Right shift (AL)	Shift (AL) right by k; sign fill	plus							
50	43 RSHA	Right shift (A)	Shift (A) right by k; sign fill	2k/3							
50	44 SF	Scale factor shift	Shift (A) left, with sign fill, until (A)35 ≠ (A)34 or [k-shiftcount] = 0; [k-shiftcount] → 00017	9,33, 2k/3							
50	45 LSHAU	Left shift (AU)	Cyclic shift (AU) left by k	5,33							
50	46 LSHAL	Left shift (AL)	Cyclic shift (AL) left by k	plus							
50	47 LSHA	Left shift (A)	Cyclic shift (A) left by k	2k/3							
50	51 SKPNBO	Skip on no borrow	Skip NI if borrow designator not set								
50	52 SKPOV	Skip on overflow	Skip NI if overflow designator set								
50	53 SKPNOV	Skip on no overflow	Skip NI if overflow designator not set								
50	54 SKPODD	Skip on odd parity	Skip NI if sum of ones in L(AU)(AL) is odd	Skip							
50	55 SKPEVN	Skip on even parity	Skip NI if sum of ones in L(AU)(AL) is even	Skip							
50	21 SKPIN	Skip on input inactive	Skip NI if chan. k input is inactive	No							
50	22 SKPON	Skip on output inactive	Skip NI if chan. k output is inactive	Skip							
50	23 SKPIN	Skip on ext. funct. inactive	Skip NI if chan. k ext. funct. is inactive	4,67							
50	50 SKP	Skip on key setting	Skip NI if k = console key setting								
50	57 SKPNR	Skip on no resume	Skip NI if resume designator is not set								
50	11 IN	Input transfer	(P)-1 → 60-2k; (P)+2 → 61+2k; set input active	20							
50	12 OUT	Output transfer	(P)+1 → 40-2k; (P)+2 → 41+2k; set output active	20							
50	13 EXF	Ext. funct. transfer	(P)-1 → 20-2k; (P)+2 → 21+2k; set ext. func. active	20							
50	15 INSTP	Terminate input	Clear input active, channel k	4							
50	16 OUTSTP	Terminate output	Clear output active, channel k	4							
50	17 EXFSTP	Terminate ext. funct.	Clear function active, channel k	4							
50	20 SRSM	Set resume	Set the resume designator (intercomputer)	4							
50	26 OUTOV	Output override	Force one word out channel k with output ack.	4,67							
50	27 EXFOV	External function override	Force one word out channel k with ext. funct.	4,67							
50	30 RIL	Remove interrupt lockout	Enable all interrupts	4							
50	32 RXL	Remove ext. int. lockout	Enable external interrupts	4							
50	34 SIL	Set interrupt lockout	Disable all interrupts	4							
50	36 SKL	Set ext. int. lockout	Disable external interrupts	4							
50	24 WFI	Wait for interrupt	Stop, then interrupt entrance reg. for NI	4							
50	56 STOP	Stop on key setting	Stop if k = Console key setting	4,67							
30*	IRJP	Indirect return jump	(P)+1 → (Y); (Y)+1 → P (See Note 2)	8							
34*	JP	Unconditional jump	Y → P	8							
54	IJPEI	Ind. jump and enable int.	(Y) → P; enable interrupts	8							
55	IJP	Indirect jump	(Y) → P	8							
76	RJP	Return jump	(P)-1 → Y; Y+1 → P (See Note 2)	8							

#Y₁₄₋₁₂ from SR_{2,0} if SR active, i.e., (SR)₃ = 1; otherwise from P₁₄₋₁₂.

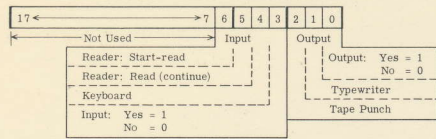
*In the listed instruction, Y = y; for the equivalent B-indexed instruction, i.e., Y = y+(B)ICR; suffix trim symbols with the letter "B", add "1" to the function code, I, and add 4 usec to the execution time.

- NOTES: 1. For f = 36, 37, 70, or 71, y = u extended with sign to 18 bits.
2. Store (P) rather than (P)+1 if return jump is executed from Interrupt Entrance Registers.
3. Comparison designator is set by a "compare" instruction, is unaffected by a "conditional jump" instruction, and is cleared by any other instruction.

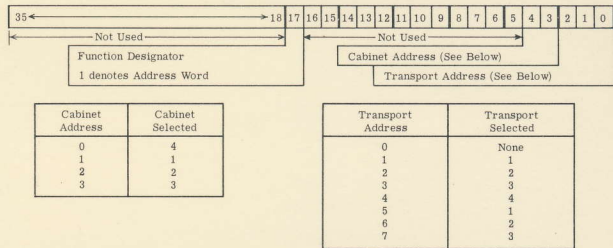
SPECIAL ADDRESSES

000-010	Fault entrance register-Index registers	060-077	Input buffer control
001-009		100-117	External interrupt entrance

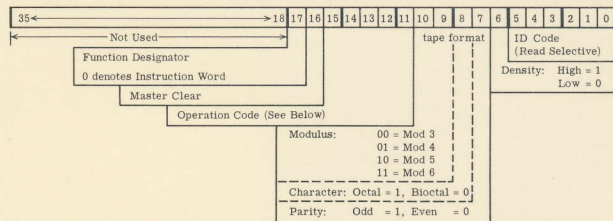
1218 COMPUTER I/O CONSOLE FUNCTION WORD FORMAT



1240 MAGNETIC TAPE HANDLER FUNCTION WORD FORMATS Address Word Format



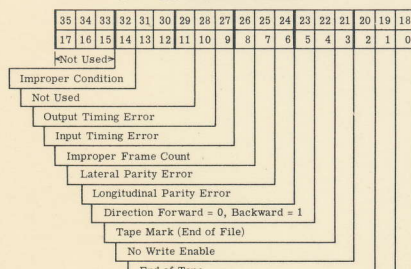
Instruction Word Format



OPERATION CODE	OPERATION
0000	Read
0001	Read Selective
0010	Read; Ignore Error Halt
0011	Space File Forward
0100	Search Forward Type I
0101	Search Forward Type II
0110	Search File Forward Type I
0111	Search File Forward Type II
1000	Back Space
1001	Back Space = Read
1011	Space File Backward
1100	Search Backward Type I
1101	Search Backward Type II
1110	Search File Backward Type I
1111	Search File Backward Type II
1100X	Write
1101	Write; XIRG
1110	Write; Ignore Error Halt
1111	Write; XIRG, Ignore Error Halt
0110X	Write Tape Mark
0111X	Write Tape Mark, XIRG
1000X	Back Space
1001X	Back Space = Read
1011X	Space File Backward
1100X	Search Backward Type I
1101X	Search Backward Type II
1110X	Search File Backward Type I
1111X	Search File Backward Type II
110X1	Rewind
110X1	Rewind, Clear Write Enable
111X0	Rewind-Read
111X1	Rewind-Read, Clear Write Enable

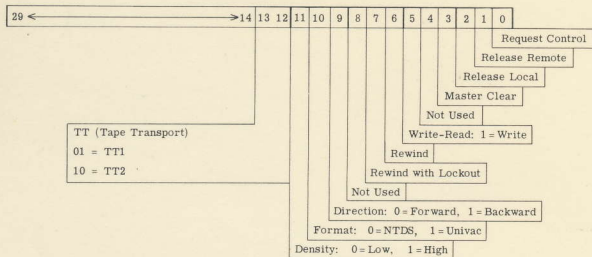
X can be 0 or 1

STATUS WORD FORMAT

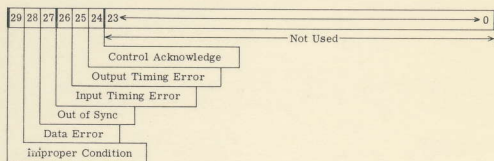


1243 MAGNETIC TAPE HANDLER

FUNCTION WORD FORMAT

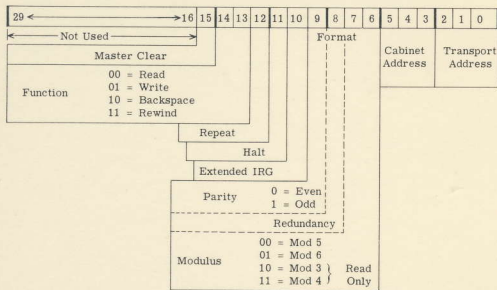


STATUS WORD FORMAT



1245 MAGNETIC TAPE HANDLER

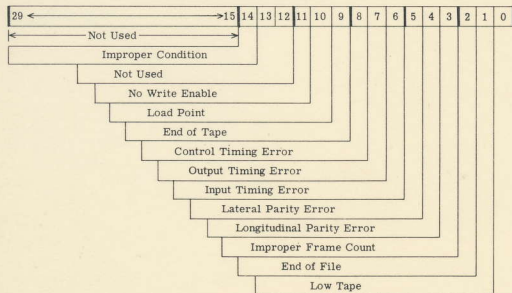
FUNCTION WORD FORMAT



Function Modifiers

BIT	READ (00)	WRITE (01)	BACKSPACE (10)	REWIND (11)
0	Stop at EOR	Stop at EOR	Backspace One Record	Normal
1	Continuous Motion	Continuous Motion	Backspace To Beginning of File	Rewind - Read
0	Ignore Error Halt	Ignore Error Halt	No Meaning	Normal
1	Halt On Error	Halt On Error		Disable Write
0	Even Parity Check	Even Parity	Even Parity Check	Even Parity Check
1	Odd Parity Check	Odd Parity	Odd Parity Check	Odd Parity Check

STATUS WORD FORMAT



Reference Card

for

UNIVAC

DIVISION OF SPERRY RAND CORPORATION

Defense Systems Programming

St. Paul, Minnesota

I/O CODES

OCTAL CODE	FLEXOWRITER		1232 I-O CONSOLE	MONITORED 1206 OUTPUT FOR SS-80 PRINTER	TELETYPE	80 COLUMN CARDS	
	UPPER CASE	LOWER CASE	MOD. FIELD DATA		LETTERS	FIGURES	CHAR. PUNCH
00	—	—	BLANK (1)	i (7)	—	—	0 0
01	T	t	UPPER CASE (2)	SPACE	T	5	1 1
02	COLOR	SHIFT	LOWER CASE (3)	—	CARRIAGE RET.	—	2 2
03	0	o	LINE FEED	—	0	9	3 3
04	SPACE	—	CARRIAGE RET.	1	SPACE	—	4 4
05	H	h	SPACE	2	H	# (8)	5 5
06	N	n	A	3	N	.	6 6
07	M	m	B	4	M	.	7 7
10	—	—	C	5	—	—	8 8
11	L	l	D	6	L)	9 9
12	R	r	E	7	R	4	A Y-1
13	G	g	F	8	G	6	B Y-2
14	I	i	G	9	I	8	C Y-3
15	P	p	H	.	P	0	D Y-4
16	C	c	I	&	C	:	E Y-5
17	V	v	J	(V	;	F Y-6
20	E	e	K	i	E	3	G Y-7
21	Z	z	L	.	Z	"	H Y-8
22	D	d	M	.	D	\$	I Y-9
23	B	b	N	:	B	? (9)	J X-1
24	S	s	O	.	S	0 (8)	K X-2
25	Y	y	P	A	Y	6	L X-3
26	F	f	Q	B	F	!	M X-4
27	X	x	R	C	X	/	N X-5
30	A	a	S	D	A	-	O X-6
31	W	w	T	E	W	2	P X-7
32	J	j	U	F	J	'	Q X-8
33	9	9	V	G	9	,	R X-9
34	U	u	W	H	U	7	S 0-2
35	Q	q	X	I	Q	1	T 0-3
36	K	k	Y	#	K	(U 0-4
37	o	o	Z	i	—	—	V 0-5
40	—	—)	i	—	—	W 0-6
41	—	—	-	i	STOP THE FLEXOWRITER (1)	—	X 0-7
42)	—	+ <	i	—	—	Y 0-8
43	—	—	=	—	MINUS (2)	—	Z 0-9
44	—	—	=	J	—	—	Y
45	CARRIAGE RET.	—	>	K	UPPER CASE K ON KEYBOARD "R" IS TYPED (3)	—	X 0-4-8
46	(—	\$	L	"R" IS TYPED (3)	—	Y 4-8
47	SHIFT UP	—	*	M	—	—	0-1
50	TAB	—	*	N	"I" IS TYPED (4)	—	3-8
51	1	—	"	O	"%" IS TYPED (5)	—	0-3-8
52	1	—	"	P	—	—	X-3-8
53	—	—	:	Q	—	—	Y-3-8
54	/	—	?	R	—	—	4-8
55	—	—	!	S	—	—	—
56	(2)	(2)	!	S	—	—	—
57	SHIFT DOWN	—	0	i	SEE MONITOR SECTION FOR MEANING OF FIRST CODE IN EACH BLOCK (6)	—	—
60	8	8	0	i	—	—	—
61	BACK SPACE	—	1	i	IGNORE (SPACE) (7)	—	—
62	5	5	2	:	PRINTED ON TAPE BUT NOT ON PAGE (8)	—	—
63	—	—	3	+	DASH (9)	—	—
64	4	4	4	/	—	—	—
65	—	—	5	S	—	—	—
66	6	6	6	T	—	—	—
67	—	—	7	U	—	—	—
70	3	3	8	V	—	—	—
71	—	—	9	W	—	—	—
72	7	7	:	X	—	—	—
73	—	—	;	Y	—	—	—
74	2	2	/	Z	—	—	—
75	—	—	.	%	—	—	—
76	—	—	□	i	—	—	—
77	CODE DELETE	—	↑	i	—	—	—

SPECIAL CODING SHEET NOTATIONS

ACTION	SYMBOL	FLEXOWRITER	80 COL. CARD
1st Major Separator	→	Tab	Tab
2nd Major Separator	→	Tab	--- (3 Dashes) Holes 4-8
Minor Separator	.	(UC 44)	*
End of Operation)	Car. Ret.	Card Release
Continuation	<	—	Continue next line as if same line

INSTRUCTION REPERTORY

COMPUTER INSTRUCTION WORD FORMAT

29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0														
f										j										k								b								y							
Function Code										Jump Designator										Operand Interpret.								Index Register								Operand or Operand Address							

NORMAL j-DESIGNATOR

0	No skip	4	AZERO	Skip NI if (A) = 0
1	SKIP Skip NI*	5	ANOT	Skip NI if (A) ≠ 0
2	QPOS Skip NI if (Q) is positive	6	APOS	Skip NI if (A) is positive
3	QNEG Skip NI if (Q) is negative	7	ANEG	Skip NI if (A) is negative

*NI = Next Instruction

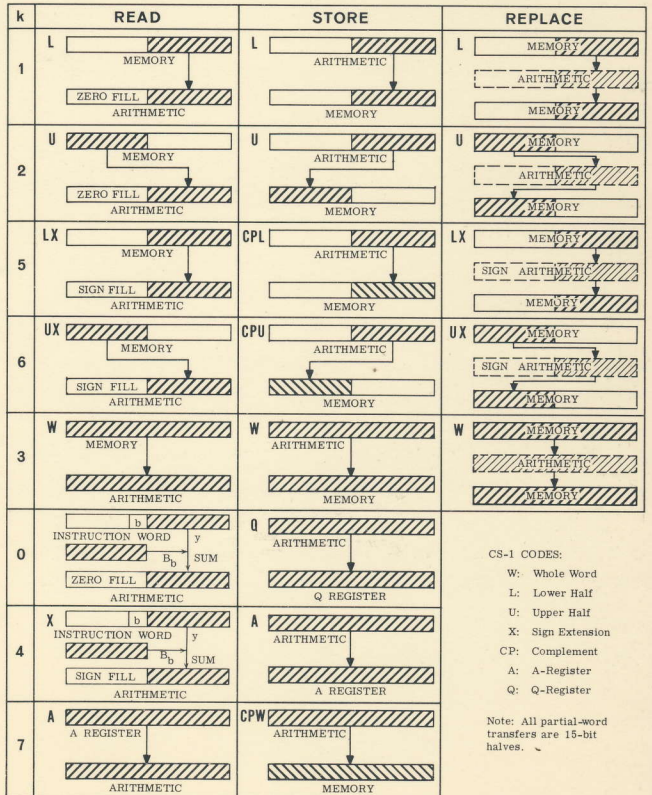
SPECIAL j-DESIGNATOR ①

j	COM A Q AQ	ADD/SUB Q	ENT/RPL LP	DIV.	RPT ②	JP/RJP	JP(60)/RJP(64)
0	No skip	No skip	No skip	No skip	NE: y=y	Always jump	RIL Release interlock
1	SKIP Always skip	SKIP Always skip	SKIP Always skip	SKIP Always skip	ADV NE: y=y+1	KEY 1 Jump Key 1	RILJP Release interlock, jump
2	YLESS Y Q	APOS Skip, (A) pos.	EVEN Skip, (A) even	NOOF Skip, no overflow	BACK NE: y=y-1	KEY 2 Stop Key 2	QPOS Jump, (Q) pos.
3	YMORE Y Q	ANEG Skip, (A) neg.	ODD Skip, (A) odd	OF Skip, overflow	ADDB NE: y=y+(B ₀)	KEY 3 Stop Key 3	QNEG Jump, (Q) neg.
4	YIN Q Y A	QZERO Skip, (Q)=0	AZERO Skip, (A)=0	AZERO Skip, (A)=0	R ³ NE: y=y+(B ₀) ⁴	STOP Always stop	AZERO Jump, (A)=0
5	YOUT Q Y or Y A	QNOT Skip, (Q)≠0	ANOT Skip, (A)≠0	ANOT Skip, (A)≠0	ADVR ³ NE: y=y+1+(B ₀) ⁵	STOP 5 Stop Key 5	ANOT Jump, (A)≠0
6	YLESS Y A	QPOS Skip, (Q) pos.	APOS Skip, (A) pos.	APOS Skip, (A) pos.	BACR ³ NE: y=y-1+(B ₀) ⁶	STOP 6 Stop Key 6	APOS Jump, (A) Pos.
7	YMORE Y A	QNEG Skip, (Q) neg.	ANEG Skip, (A) neg.	ANEG Skip, (A) neg.	ADDBR ³ NE: y=y+(B ₀) ⁷	STOP 7 Stop Key 7	ANEG Jump, (A) neg.

- ① j-designator used to designate B register in the following functions: ENT•B₁ STR•B₁ BSK BJP
- ② NE = next execution
- ③ Only for RPL class instructions
- ④ (B₀) included only in store portion; (B₀) not included in read portion.

JUMP ADDRESS
 k=0,4 Y=(Y₁)
 k=1,3,5 Y=(Y₁)
 k=2,6 Y=(Y₁)
 k=7 Y=(A₁)

OPERAND INTERPRETATION - NORMAL - k DESIGNATOR



- CS-1 CODES:
 W: Whole Word
 L: Lower Half
 U: Upper Half
 X: Sign Extension
 CP: Complement
 A: A-Register
 Q: Q-Register

Note: All partial-word transfers are 15-bit halves.

TYPE	MNEMONIC CODE	INSTRUCTION	OPERATION	EXECUTION TIMES* For k-Designators
TRANSFER	ENT•Q	10	Enter Q register	Rd 11.2/9.0
	ENT•A	11	Enter A register	Rd 8.0/6.4
	ENT•B ₁	12	ENT•B register	Rd 8.0/6.4
	STR•Q	14	Store Q register	St 12.8/9.6
	STR•A	15	Store A register	St 9.6/7.2
	STR•B ₁	16	Store B register	St 6.4/4.8
SHIFTS	RSR•Q	01	Shift Q right by Y ₀₅₋₀₀ Sign Fill	Rd 8.0/6.4
	RSR•A	02	Shift A right by Y ₀₅₋₀₀ Sign Fill	Rd 11.2/9.6
	RSR•AQ	03	Shift AQ right by Y ₀₅₋₀₀ Sign Fill	Rd 11.2/9.6
	LSR•Q	05	Shift Q Left by Y ₀₅₋₀₀ cyclic	Rd 16.0/12.8
	LSR•A	06	Shift A Left by Y ₀₅₋₀₀ cyclic	Rd 16.0/12.8
	LSR•AQ	07	Shift AQ Left by Y ₀₅₋₀₀ cyclic	Rd 16.0/12.8
COM.	COM•A•Q•AQ	04	Compare	Rd 12.8/9.6
	COM•MASK	43	Mask Compare	Rd 11.2/9.6
JUMPS	JP	40	Arithmetic Jump	Rd 8.0/6.4
	RJP	64	Arithmetic Return Jump	Rd 12.8/9.6
	RIL	60 ₀₁	Release Interrupt Lockout	Rd 16.0/12.8
	RILJP	60 ₀₂	Release Interrupt Lockout, Jump	Rd 8.0/6.4
	C ₀ ACTIVE IN	62	Input Active Buffer Jump	Rd 8.0/6.4
	C ₀ ACTIVE OUT	63	Output Active Buffer Jump	Rd 8.0/6.4
MODEY	RPT	70	Repeat	Rd 8.0/6.4
	BSK•B ₁	71	Index Skip	Rd 9.6/7.2
	BJP•B ₁	72	Index Jump	Rd 8.0/6.4
INPUT OUTPUT	STR•C ₀	17	Store Input Channel	St 16.0/12.8
	IN•C ₀	73	Initiate Input	Rd 16.0/12.8
	IN•C ₀ •[y]•MONITOR	75	Initiate Input, Monitor	Rd 8.0/6.4
	TERM•C ₀ •INPUT	66	Terminate Input	Rd 8.0/6.4
	EX•FCT•C ₀	13	Enter External Function	Rd 8.0/6.4
	OUT•C ₀	74	Initiate Output	Rd 16.0/12.8
COMP. CLR.	NO•OP	12	No Operation (15000 00000)	Rd 8.0/6.4
	CP•A	31	Complement A (51040 71777)	Rd 12.8/9.6
	CP•Q	14	Complement Q (14000 00000)	Rd 6.4/4.8
	CL•A	11	Clear A (11000 00000)	Rd 12.8/9.6
	CL•Q	10	Clear Q (10000 00000)	Rd 6.4/4.8
	CL•B ₁	12	Clear B ₁ (12010 00000)	Rd 16.0/12.8
ADDITION	ADD•A	20	Add	Rd 12.8/9.6
	ADD•Q	26	Q Add	Rd 11.2/9.6
	ADD•LP	41	Add Logical Product	Rd 11.2/9.6
	ENT•Y•Q	30	Add Q and Load A	Rd 8.0/6.4
	STR•A•Q	32	Add Q and Store	Rd 12.8/9.6
	RPL•A•Y	24	Replace Add	Rd 24.0/18.0
SUBTRACTION	SUB•A	21	Subtract	Rd 11.2/9.6
	SUB•Q	27	Q Subtract	Rd 12.8/9.6
	SUB•LP	42	Subtract Logical Product	Rd 11.2/9.6
	ENT•Y•Q	31	Subtract Q and Load A	Rd 8.0/6.4
	STR•A•Q	33	Subtract Q and Store	Rd 12.8/9.6
	RPL•A•Y	25	Replace Subtract	Rd 24.0/18.0
MULT. DIV.	MUL	22	Multiply	Rd 35.2/26.4
	ENT•LP	40	Enter Logical Product	Rd 11.2/9.6
	STR•LP	44	Store Logical Product	Rd 12.8/9.6
	RPL•LP	47	Replace Logical Product	Rd 12.8/9.6
	DIV	23	Divide	Rd 11.2/9.6
	SEL•SET	50	Selective Set	Rd 12.8/9.6
SEL•CP	51	Selective Complement	Rd 9.6/7.2	
SEL•CL	52	Selective Clear	Rd 8.0/6.4	
SEL•SU	53	Selective Substitute	Rd 11.2/9.6	
RSE•SET	54	Replace Selective Set	Rd 24.0/18.0	
RSE•CP	55	Replace Selective Compl.	Rd 11.2/9.6	
RSE•CL	56	Replace Selective Clear	Rd 8.0/6.4	
RSE•SU	57	Replace Selective Substitute	Rd 11.2/9.6	

NI: NEXT INSTRUCTION CLASS: Rd=READ, St=STORE, Rp=REPLACE *Note: Lower time is for repeat mode.

1206 COMPUTER I/O INSTRUCTION WORD FORMAT

29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
f							↑	↑	b							y													
Function Code							Channel Designator			Oper. Inter.			Index Register			Operand or Operand Address													

OPERAND INTERPRETATION - \hat{k} DESIGNATOR

\hat{k}	FUNCTION CODE = 13(j≠0 or 1)	FUNCTION CODE = 17	FUNCTION CODE = 62 or 63	FUNCTION CODE = 73 - 76
0	NOT USED	NOT USED	INSTRUCTION WORD \hat{y} b ₁ b ₂ b ₃ b ₄ b ₅ b ₆ b ₇ b ₈ b ₉ b ₁₀ b ₁₁ b ₁₂ b ₁₃ b ₁₄ b ₁₅ b ₁₆ b ₁₇ b ₁₈ b ₁₉ b ₂₀ b ₂₁ b ₂₂ b ₂₃ b ₂₄ b ₂₅ b ₂₆ b ₂₇ b ₂₈ b ₂₉ P REGISTER	INSTRUCTION WORD \hat{y} b ₁ b ₂ b ₃ b ₄ b ₅ b ₆ b ₇ b ₈ b ₉ b ₁₀ b ₁₁ b ₁₂ b ₁₃ b ₁₄ b ₁₅ b ₁₆ b ₁₇ b ₁₈ b ₁₉ b ₂₀ b ₂₁ b ₂₂ b ₂₃ b ₂₄ b ₂₅ b ₂₆ b ₂₇ b ₂₈ b ₂₉ BUFFER CONTROL REG.
1	NOT USED	NOT USED	MEMORY P REGISTER	MEMORY BUFFER CONTROL REG.
2	NOT USED	NOT USED	MEMORY P REGISTER	NOT USED
3	MEMORY EXT. FUNCTION REG.	CHANNEL MEMORY	MEMORY P REGISTER	MEMORY BUFFER CONTROL REG.

SPECIAL ADDRESSES

00000	Fault	00040-00055	Input monitor interrupt
00020-00035	External interrupt	00060-00075	Output monitor interrupt
00036	Real-time clock	00100-00115	Input buffer control
		00120-00135	Output buffer control

FH-880 MAGNETIC DRUM FUNCTION WORD FORMAT

29	28	27	26	25	24	23	22	←	→	0	
Function Code							Not Used	Beginning Drum Address			

FUNCTION CODES

Function Code	Description
02	Continuous write: start at specified address
42	Continuous read: start at specified address
52	Block read: read until EOB
45	Search: search until find
55	Block search: search until find or EOB
46	Search-read: search until find, then continue read
56	Block search-read: search, find, read until EOB
40	Bootstrap: without monitor
50	Bootstrap: with monitor
23	Terminate: without monitor
33	Terminate: with monitor

STATUS WORD FORMAT

29	28	27	26	25	24	23	22	←	→	0	
Function Code							Not Used	Drum Address			

STATUS CODES

Status Code	Description
05	Search find
40	Normal completion
34	End of file
04	End of block
74	Interlock fault
50	Illegal function
54	Invalid address
64	Parity error, continue read
30	Character-count error
60	Sequence error
14	Write fault
06	Overflow parity error
07	Parity error, non-continue read
70	Write character count error

1206 MONITOR CONTROL CARD

Column 80 must have a 5 punch on all cards

CONTROL						
FIELD 0 COL (1-10)	FIELD 1 COL (11-20)	FIELD 2 COL (21-30)	FIELD 3 COL (31-40)	FIELD 4 COL (41-50)	FIELD 5 COL (51-60)	DESCRIPTION
—	JOB	Job I. D. Number	Account Number	Termination Time	Programmers Name	Job identification card
—	LOADR	Base Address	Last Address	—	—	Loads a relative deck
—	LOADA	Starting Address	Last Address	—	—	Loads an absolute deck
—	LOD40	Tape Unit	Starting Address	Last Address	—	Loads an absolute 40 format tape
—	LOADM	Tape Unit	Starting Address	Last Address	File Number	Loads scratch tape (monitor core dump)
—	LOD41	Tape Unit	Starting Address	Last Address	—	Loads a relative 41 format scratch tape
—	LOADC	Base Address	—	—	—	Load correction cards
—	CS-1	—	—	—	—	Compiling job follows
—	LIB	Program Name	Base Address	—	—	Program from system library
—	WRITEM	Tape Unit	Starting Address	Ending Address	File Number	Writes core dump on scratch tape
—	PTCNV	Source Medium	Converted Medium	—	—	Converts paper tape to mag. tape or card format
—	TRA	Transfer Address	—	—	—	Transfers control to loaded program
—	END	—	—	—	—	Indicates job is finished.
*(Descriptive Comments for Users Convenience Col 2-79)						Prints descriptive comments

*(Descriptive Comments for Users Convenience Col 2-79)

DEBUG						
Execution Address	SNAP	No. of Passes before Dump	Number of Dumps	Starting Address of Area	Ending Address of Area	Dumps upon executing an instruction
Execution Address	SNAPW	No. of Passes before Dump	Number of Dumps	Starting Address of Area	Ending Address of Area	Same as SNAP except a changed word dump
—	CORPM	Starting Address	Ending Address	—	—	Dumps an area after executing program
—	CWCOR	Starting Address	Ending Address	—	—	Same as CORPM except a changed word dump

⓪ Not applicable on compiling jobs.

PRINTER FORMAT CONTROLS

First (XS-3 Coded) Character of a 24-Word (120 Characters) Buffer Area

First Character	Function
1 thru 9	Space 1 thru 9, respectively, and print
A	Print this header line on top of next page ⓪
B	Space to top of next page and print
C	Space to last line of this page and print
D	Space 15 lines and print
E	Space 30 lines and print
F	Space 5 lines and print this header (subheader) line ⓪

⓪ This header line will print on the top of each ensuing page until replaced by a new header.

I/O DIRECTIONS (HIGH LEVEL)

LABEL	MNREAD•P0 (ERROR EXIT) (NORMAL EXIT)	Reads one 80-column card image in XS-3 code from SYSIT into an area defined by P0.
LABEL	MNDREAD•P0 (ERROR EXIT) (NORMAL EXIT)	Reads one data card image from SYSIT into an area defined by P0.
LABEL	MNPRINT•P0 (NORMAL EXIT)	Transfers a 24 word blockette to SYSOT for off-line printing. The print information must be XS-3 code in the area defined by P0.
LABEL	MNDPUNCH•P0 (NORMAL EXIT)	Transfers an area, in octal notation, to SYSOT for peripheral punching. The area defined by P0 will produce cards having 7 1206 words of data per card.
LABEL	MNBWRITE•P0•P1•P2 (NORMAL EXIT)	Writes a variable length binary record in odd parity onto a tape unit specified by P0. P1 designates the address of a word containing the starting and ending address of the block to be written. P2 is the return control designator.
LABEL	MNCWRITE•P0•P1•P2	Same as MNBWRITE except record is written in even parity.
LABEL	MNBREAD•P0•P1•P2 (END OF FILE EXIT) (ERROR EXIT) (NORMAL EXIT)	Reads a variable length binary record from the tape unit specified by P0. P1 designates the address of a word containing the starting address of the input area. P2 is the return control designator.
LABEL	MNCREAD•P0•P1•P2 (END OF FILE EXIT) (ERROR EXIT) (NORMAL EXIT)	Reads a variable length record in odd parity from the tape unit specified by P0. P1 designates the address of a word containing the starting address of the input area. P2 is the return control designator.
LABEL	REWIND•P0 (NORMAL EXIT)	Rewinds a tape unit specified by P0.
LABEL	ENDFILE•P0 (NORMAL EXIT)	Writes an END OF FILE mark on the tape specified by P0.
LABEL	FSFILE•P0•P1 (NORMAL EXIT)	Forward space the tape designated by P0 a number of files specified by P1.
LABEL	FSREC•P0•P1 (NORMAL EXIT)	Forward space the tape designated by P0 a number of records specified by P1.
LABEL	BACKSPACE•P0•P1 (NORMAL EXIT)	Backspaces the tape designated by P0 a number of records specified by P1.
LABEL	MNBSPD•P0	Backspaces a number of data images specified by P0.
LABEL	MNBSPC•P0	Backspaces a number of 80 column card images specified by P0.