HIGH-SPEED PRINTER

FUNCTION WORDS

1	TOP MARGIN	BOT MARGIN	NO. LINES/PAGE	CHAN	UNIT
29	28 23	22 17	16 08	07 04	03 00

0		LINE SPACE	000 000 000	NEXT PAGE	000000	UNIT SELECT
29	24	23 18	17 09	08	07 04	03 00

PRINT BUFFER: 26 WDS; USE STOP CODE IF < 128 CHAR/LINE

STATUS WORDS

PRINTER INTERRUPT CODE		REX STATUS CODE ⟶ A_{05-00}		CONSOLE TYPEOUT
20	SEQUENCE OPERATION ERROR	00	**OPERATION NOT COMPLETE**	NONE
30	CHARACTER COUNT ERROR	01	**OPERATION COMPLETE**	NONE
40	NORMAL COMPLETION	07	**ILLEGAL ERROR**	MSG 231 P
54	ILLEGAL UNIT SELECT	10	**ILLEGAL FUNCTION**	MSG 231 P
74	INTERLOCK ERROR	06	**INTERLOCK FAULT**	MSG 230 P
50	ILLEGAL FUNCTION	11	**ILLEGAL UNIT SELECT**	MSG 231 P
		12	**PRINT ERROR**	MSG 231 P

PUNCHED CARDS

FC		MULT. READ DESIGNATOR	
29 24	23 15	14 09	08 00

CARD READER

SPURT CODE	FUNCTION	CODE
03 **RMODE•FD**	Condition: Card to Fieldata	62/72
04 **RMODE•CBIN**	Condition: Column Binary Image	63/73
05 **RMODE•RBIN**	Condition: Row Binary Image	64/74
00 **STACK 1**	Select Stacker 1	60/70
01 **STACK 2**	Select Stacker 2	61/71
02 **READNT**	Read 1 Card; No Transfer	43/53
06 **READ**	Transfer 1 Card; Trip Fill	42/52
07 **TRANS**	Transfer 1 Card	41/51
10 **MREAD** *	Read, Transfer XX Cards	42/52

*Designate No. of Cards to be read in bits 14–09 of Function Word. (Maximum: 63 Cards)

CARD PUNCH

SPURT CODE	FUNCTION	CODE
13 **PMODE•FD**	Condition: Fieldata to Card	04/14
14 **PMODE•CBIN**	Condition: Column Binary Image	05/15
15 **PMODE•RBIN**	Condition: Row Binary Image	06/16
11 **PUNCH**	Punch Card, Stacker 0	02/12
12 **PUNCHS**	Punch Card, Stacker 1	03/13

CARD SUBSYSTEM INTERRUPT CODES	
20	SEQUENCE OPERATION ERROR
30	CHARACTER COUNT ERROR
40	NORMAL FUNCTION TERMINATION
50	ILLEGAL FUNCTION
54	CHARACTER VERIFY ERROR
60	INAPPROPRIATE FUNCTION
70	ILLEGAL CHARACTER
74	INTERLOCK ERROR

REX STATUS CODE ⟶ A_{05}–A_{00}		CONSOLE TYPEOUT
00	**OPERATION NOT COMPLETE**	NONE
01	**OPERATION COMPLETE**	NONE
06	**INTERLOCK FAULT**	MSG 260/270 P
07	**VERIFY ERROR**	MSG 263/273 P
10	**ILLEGAL CHARACTER**	NONE
11	**ILLEGAL FUNCTION**	MSG 261/271 P
12	**INAPPROPRIATE OPERATION**	MSG 264 PSS *
13	**INCORRECT PARAMETER**	MSG 262/272 P

*Typeout occurs only on "Late Stacker Selection Error"

MAGNETIC TAPE FUNCTION CODES

UNISERVO IIA

FUNCTION WORD

FC		CM		UNIT SELECT BIT
29	24	14		11 00

SPURT CODE			FUNC CODE
OC	MNEM	FUNCTION	
0	**REWIND**	Rewind Tape	20/30
1	**RWI**	Rewind and Interlock	21/31
2	**SEARCHB**	Search-Read Backward	66/76
3	**SEARCHF**	Search-Read Forward	46/56
4	**PWRITE**	Write At 125 CPI	01/11
6	**WRITE**	Write At 250 CPI	02/12
10	**READF**	Read Forward 1 Block	42/52
12	**READB**	Read Backward 1 Block	62/72
13	**MOVEF**	Move Forward N Blocks	42/52
14	**MOVEB**	Move Backward N Blocks	62/72
2	***LOCATEB**	Search Backward, No Read	66/76
3	***LOCATEF**	Search Forward, No Read	46/56
		Terminate	23/33
		Bootstrap	40/50

CM: COMPATIBLE MODE (CM=1)
FC: FUNCTION CODE
OC: SPURT OPERATION CODE
*BUFFER WD=0

UNISERVO IIIA

FUNCTION WORD

FC		UNIT SELECT
29	24	3 0

SPURT CODE			FUNC CODE
OC	MNEM	FUNCTION	
0	**REWIND**	Rewind Tape	20/30
1	**RWI**	Rewind and Interlock	21/31
13	**MOVEF**	Move Forward N Blocks	41/51
14	**MOVEB**	Move Backward N Blocks	61/71
10	**READF**	Read Forward 1 Block	41/51
12	**READB**	Read Backward 1 Block	61/71
2	**SEARCHF**	Search-Read Forward	45/55
3	**SEARCHB**	Search-Read Backward	65/75
2	**LOCATEF**	Search Forward, No Read	45/55
3	**LOCATEB**	Search Backward, No Read	65/75
2	***SEARCHF**	Mask Search Forward, Read	46/56
3	***SEARCHB**	Mask Search Backward, Read	66/76
6	**WRITE**	Write 1 Block	01/11
7	**ENDFILE**	Write End-Of-File	03/13
5	**WRITEO**	Contingency Write	02/12
		Reposition Read Forward	42/52
		Reposition Read Backward	62/72
		Terminate	23/33
		Bootstrap	40/50
		Request Control	26/36
		Demand Control	25/35
		Release Control	24/34

* If mask word is supplied a masked search will be performed.

UNISERVO IIIC

FUNCTION WORD

FC			UNIT SELECT
29	21		11 00

SPURT CODE				FUNC CODE
OC	MNEM	FUNCTION		
10	**REWIND**	Rewind Tape		201/301
11	**RWI**	Rewind and Interlock		211/311
13	**MOVEB**	Move Backward N Blocks		203/303
14	**MOVEF**	Move Forward N Blocks		
12	**BACKFILE**	Backspace File		213/313
0	**LOBIN**	Condition: Binary, Low		
1	**HIBIN**	Condition: Binary, High		
2	**LOBCD**	Condition: BCD, Low		
3	**HIBCD**	Condition: BCD, High		
4	**SEARCH**	Search: BIN, 556		460/560
		BIN, 200		470/570
		BCD, 556		462/562
		BCD, 200		472/572
5	**WRITE**	Write: BIN, 556		020/120
		BIN, 200		030/130
		BCD, 556		022/122
		BCD, 200		032/132
6	**READ**	Read: BIN, 556		420/520
		BIN, 200		430/530
		BCD, 556		422/522
		BCD, 200		432/532
07	**ENDFILE**	Write End-Of-File	556	023/123
			200	033/133
		Backspace 1 Block		203/303
		Erase		003/103
		Terminate		230/330
		Bootstrap		400/500

MAGNETIC TAPE SUBSYSTEM INTERRUPT CODES

UNISERVO IIA		UNISERVO IIIA		UNISERVO IIIC	
30	CHAR. COUNT, CHAN. SYNCH.	64	FUNCTION TERMINATED (ABN)	34	ILLEGAL FUNCTION
74	INTERLOCK	50	ILLEGAL FUNCTION	24	REQUESTED UNIT REWINDING
50	ILLEGAL FUNCTION	74	INTERLOCK	54	END OF FILE
54	UNIT SELECT ERROR	14	OTHER CU HAS CONTROL	50	SEQUENCE ERROR, CONTROL
10	REQUESTED UNIT REWINDING	10	REQUESTED UNIT REWINDING	74	INTERLOCK
24	SEQ. ERROR, CONTROL UNIT	30	TAPE UNIT FAULT	44	REPEAT OPERATION
34	END OF FILE	34	END OF FILE	70	ABNORMAL FRAME COUNT
70	CHAR. COUNT, CONTROL UNIT	60	REPEAT OPERATION (WRITE)	60	END OF TAPE
60	PARITY ERROR	54	IMPROPERLY WRITTEN BLOCK	30	CHAR. COUNT, CHAN. SYNCH.
44	END OF TAPE	60	REPEAT OPERATION	20	SEQ. ERROR, CHAN. SYNCH.
20	SEQ. ERROR, CHAN. SYNCH.	70	ABNORMAL FRAME COUNT	40	NORMAL COMPLETION
40	NORMAL COMPLETION	44	TAPE MARKER DETECTED		
		40	NORMAL COMPLETION		

REX TRANSLATED STATUS CODES A_{05-00}		CONSOLE TYPE OUT IIA	IIIA	IIIC
0	OPERATION NOT COMPLETE	----	----	----
1	OPERATION COMPLETE	----	----	----
2	NOT USED	----	----	----
3	UNISERVO REWINDING	----	----	----
4	END OF FILE	----	----	----
5	END OF TAPE	----	----	----
6	INTERLOCK	MSG 200	MSG 210	MSG 220
7	SUBSYSTEM FAULT	MSG 201	MSG 211	MSG 221
10	INCORRECT PARAMETER		MSG 212	MSG 222
10	READ/WRITE ERROR	MSG 201		
11	SEQUENCE ERROR	MSG 201		
12	INCORRECT PARAMETER	MSG 202		

SPURT I/O PACKET FORMAT

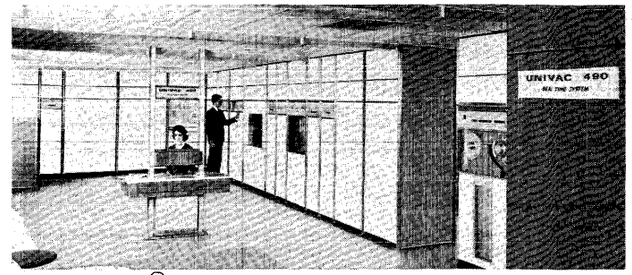
		WORD FORMAT
ENTRY		SILRJP • U (140) 64120 00140
PARAMETER WORDS	1	FUNCTION WORD (VARIABLE FORMAT) *
	2	N (29–24) C (23–20) P (19–15) 00000 (14–00)
	3	1103 (29–18) b (17–15) XXXXX (14–00)
	4	110 (29–21) k (20–18) b (17–15) YYYYY (14–00)
	5	110 (29–21) k (20–18) b (17–15) ZZZZZ (14–00)

N: NUMBER OF WORDS IN PACKET
C: CHANNEL NUMBER
P: PERIPHERAL CODE
b: ANY B-REGISTER FROM B_1–B_6 (B_7 EXCLUDED)
k: DETERMINES Y
XXXXX: ADDRESS OF BUFFER CONTROL WORD SUBJECT TO MODIFICATION (XXXXX + b_1)
YYYYY: ADDRESS OF IDENTIFIER WORD, IF APPLICABLE, SUBJECT TO MODIFICATION (YYYYY + b_n) & K
ZZZZZ: ADDRESS OF MASK WORD ON APPLICABLE FUNCTIONS

P	PERIPHERAL
1	DRUM
2	DISC FILE
3	UNISERVO IIA
4	PT READER
5	PRINTER
6	CARD READER
7	CARD PUNCH
10	UNISERVO IIIC
11	PT PUNCH
12	UNISERVO IIIA
13	COMMUNICATION
14	FASTRAND

ERROR RECOVERY	
REGISTER	CONTENTS
A	STATUS WORD
Q_U	I/O REQUEST ADDRESS
Q_L	CKSTAT + 1 ADDRESS
B_7	aaaaa = B_1 & B_2
	aaaaa + 1 = B_3 & B_4
	aaaaa + 2 = B_5 & B_6

TAPE	29	OC	20 15	14		CHAN 07 04	UNIT 03 00
CARD	29 24	OC			MULTI-READ 14 09		00
DRUM DISC FAST	29 24	OC	k 20 18	b 17 15	14		00



UNIVAC® 490 REAL-TIME SYSTEM

Reference Card

FIELDATA	CHARACTER CODE: CONSOLE T/W	CHARACTER CODE: HSP	CHARACTER CODE: 80 COL. CARD	BAUDOT
000000	(NO ACTION)	NP	7-8	00000
000001	\	NP	12-5-8	11011
000010	%	NP	11-5-8	11111
000011	LINE FEED	≠	12-7-8	00010
000100	CAR RET	NP	11-7-8	01000
000101	SPACE	NP	BLANK	00100
000110	A	A	12-1	L00011
000111	B	B	12-2	L11001
001000	C	C	12-3	L01110
001001	D	D	12-4	L01001
001010	E	E	12-5	L00001
001011	F	F	12-6	L01101
001100	G	G	12-7	L11010
001101	H	H	12-8	L10100
001110	I	I	12-9	L00110
001111	J	J	11-1	L01011
010000	K	K	11-2	L01111
010001	L	L	11-3	L10010
010010	M	M	11-4	L11100
010011	N	N	11-5	L01100
010100	O	O	11-6	L11000
010101	P	P	11-7	L10110
010110	Q	Q	11-8	L10111
010111	R	R	11-9	L01010
011000	S	S	0-2	L00101
011001	T	T	0-3	L10000
011010	U	U	0-4	L00111
011011	V	V	0-5	L11110
011100	W	W	0-6	L10011
011101	X	X	0-7	L11101
011110	Y	Y	0-8	L10101
011111	Z	Z	0-9	L10001
100000	)	)	12-4-8	U10010
100001	–	–	11	U00011
100010	+	+	12	U11010
100011	<	NP	12-6-8	NO CODE
100100	=	NP	3-8	NO CODE
100101	>	NP	6-8	NO CODE
100110	¬	&	2-8	NO CODE
100111	$	$	11-3-8	U01001
101000	*	*	11-4-8	NO CODE
101001	(	(	0-4-8	U01111
101010	'	%	0-5-8	U10001
101011	:	:	5-8	U01110
101100	?	NP	12-0	U11001
101101	!	NP	11-0	U01101
101110	COMMA	COMMA	0-3-8	U01100
101111	STOP	NP	0-6-8	NO CODE
110000	0	0	0	U10110
110001	1	1	1	U10111
110010	2	2	2	U10011
110011	3	3	3	U00001
110100	4	4	4	U01010
110101	5	5	5	U10000
110110	6	6	6	U10101
110111	7	7	7	U00111
111000	8	8	8	U00110
111001	9	9	9	U11000
111010	APOSTROPHE	APOSTROPHE	4-8	U01011
111011	;	;	11-6-8	U11110
111100	/	/	0-1	U11101
111101	.	.	12-3-8	U11100
111110	□	NP	0-7-8	NO CODE
111111	^	STOP	0-2-8	NO CODE

INSTRUCTION REPERTOIRE

TYPE	MNEM CODE	OCTAL CODE	INSTRUCTION	OPERATION	CLASS	EXECUTION TIMES* k:0,4	k:7	k:1,2,3,5,6
TRANSFER	ENT•Q	10	Enter Q	Y ⟶ Q	Rd	8.4 / 6.0	7.2 / 4.8	12
	ENT•A	11	Enter Accumulator	Y ⟶ A	Rd			
	ENT•B_n	12	Enter B-Register	Y ⟶ B_n; B_0: NoOp	Rd	6.0 / 3.6		
	STR•Q	14	Store Q	(Q) ⟶ Y	St	8.4 / 4.8	12 /	7.2
	STR•A	15	Store Accumulator	(A) ⟶ Y	St			
	STR•B_n	16	Store B-Register	(B) ⟶ Y	St			
SHIFTS	RSH•Q	01	Shift Q Right	Shift Q right by Y_{05-00}; Sign Fill	Rd	8.4 /	9.6 /	13.2 / .....
	RSH•A	02	Shift A Right	Shift A right by Y_{05-00}, Sign Fill	Rd			
	RSH•AQ	03	Shift AQ Right	Shift AQ right by Y_{05-00}, Sign fill	Rd	8.4 /	9.6 /	13.2 / .....
	LSH•Q	05	Shift Q Left	Shift Q left by Y_{05-00}, circularly	Rd	8.4 /	9.6 /	13.2 / .....
	LSH•A	06	Shift A Left	Shift A left by Y_{05-00}, circularly	Rd			
	LSH•AQ	07	Shift AQ Left	Shift AQ left by Y_{05-00}, circularly	Rd	12 /	13.2 /	16.8 / .....
COM	COM•A•Q•A•Q	04	Compare	A:Y or Q:Y or AQ:Y; and sense j	Rd	9.6 / 7.2	8.4 / 6.0	12 / 8.4
	COM•MASK	43	Mask Compare	(A)–L (Y (Q)); sense j A and Q Unchanged	Rd	8.4 / 6.0	7.2 / 4.8	12 / 7.2
JUMPS	JP	60	Arithmetic Jump (Normal)	Y ⟶ P, per j; Y: 15 bits maximum	Rd	6.0 /	7.2 /	12 / .....
	JP	61	Manual Jump (Normal)	Y ⟶ P, per j; Y: 15 bits maximum	Rd			
	RJP	64	Arithmetic Return Jump	P+1 ⟶ Y_{14-00}; Y+1 ⟶ P if j is met	Rd	13.2 /	14.4 /	18 / .....
	RJP	65	Manual Return Jump	P+1 ⟶ Y_{14-00}; Y+1 ⟶ P if j is met	Rd			
	RIL	$60_{j=0}$	Release Interrupt Lockout	Release interrupt lockout	Rd			
	RILJP	$60_{j=1}$	Release Interrupt Lckt, Jump	Release interrupt lockout and Y-jump	Rd			
	SIL	$64_{j=0}$	Set Interrupt Lockout	Set interrupt lockout	Rd			
	SILRJP JP•C_n	$64_{j=1}$	Set Interrupt Lockout, Jump	Set interrupt lockout and Y-return jump	Rd			
	ACTIVE IN JP•C_n	62	Input Active Buffer Jump	If C_n active, jump to Y	Rd	6.0 /	..... /	12 / .....
	ACTIVE OUT	63	Output Active Buffer Jump	If C active, jump to Y	Rd			
MODIFY	RPT	70	Repeat	NI (Y) times per j; B7=NEn	Rd	..... /	7.2 /	..... / .....
	BSK•B_n	71	Index Skip	If (B_n)≠(Y), skip NI, and CL (B_n) If (B_n)=(Y), take NI, B_n=(B+1)	Rd	7.2 /	8.4 /	..... / .....
	BJP•B_n	72	Index Jump	If (B_n)≥0, take NI If (B_n)≠0, jump to Y; B_n=(B-1)	Rd	6.0 /	7.2 /	..... / .....
INPUT/OUTPUT	STR•C_n	17	Store Input Channel	C_n ⟶ Y	St	..... /		15.6 / .....
	IN•C_n	73	Initiate Input	Activate buffer on channel n	Rd	12 /		18 / .....
	IN•C_n MONITOR	75	Initiate Input, Monitor	Activate buffer on channel n	Rd	12 /		18 / .....
	TERM•C_n INPUT	66	Terminate Input	Terminate buffer on channel n	Rd	6.0 /		12 / .....
	EX-FCT•C_n	13	Enter External Function	Y ⟶ C_n if j ≠ 0 or 1	Rd	..... /		18 / .....
	OUT•C_n	74	Initiate Output	Activate buffer on channel n	Rd	12 /		18 / .....
	OUT•C_n MONITOR	76	Initiate Output, Monitor	Activate buffer on channel n	Rd			
	TERM•C_n OUTPUT	67	Terminate Output	Terminate buffer on channel n	Rd	6.0 /		12 / .....
	TEST•C_0/C_1	13	Test Input Buffer	Test C_0/C_1; if active, skip NI	Rd	..... /		18 / .....
COMP./CLR.	NO-OP	-	No Operation	12 000 00000	Rd			
	CP•A	-	Complement A	15070 00000	St			
	CP•Q	-	Complement Q	14000 00000	St			
	CL•A•Q	-	Clear A or Q	11000 00000 10000 00000	Rd			
	CL•B_n	-	Clear B_n	12010 00000	Rd			
	CL•Y	-	Clear Y	160k0$_{8y}$	St			
ADDITION	ADD•A	20	Add	(A) + Y ⟶ A	Rd	8.4 / 6.0	7.2 / 4.8	12 / 7.2
	ADD•Q	26	Q Add	(Q) + Y ⟶ Q	Rd	9.6 / 7.2	8.4 / 6.0	12 / 8.4
	ADD•LP	41	Add Logical Product	L (Y (Q)) + (A) ⟶ A	Rd	8.4 / 6.0	7.2 / 4.8	12 / .....
	ENT•Y+Q	30	Add Q and Load A	(Q) + Y ⟶ A	Rd			
	STR•A+Q	32	Add Q and Store	(A) + (Q) ⟶ Y and A	St	8.4 / 4.8	12 /	..... / 7.2
	RPL•A+Y	24	Replace Add	Y + (A) ⟶ Y and A	Rp			
	RPL•Y+Q	34	Replace Add Q	Y + (Q) ⟶ Y and A	Rp			18 / .....
	RPL•Y+1	36	Replace Add 1	Y + 1 ⟶ Y and A	Rp			
	RPL•A+LP	45	Replace Add Logical Product	L (Y (Q)) + (A) ⟶ Y and A	Rp			12
SUBTRACTION	SUB•A	21	Subtract	(A) – Y ⟶ A	Rd	8.4 / 6.0	7.2 / 4.8	12 / 7.2
	SUB•Q	27	Q Subtract	(Q) – Y ⟶ Q	Rd	9.6 / 7.2	8.4 / 6.0	12 / 8.4
	SUB•LP	42	Subtract Logical Product	(A) – L (Y (Q)) ⟶ A	Rd	8.4 / 6.0	7.2 / 4.8	12 / 7.2
	ENT•Y-Q	31	Subtract Q and Load A	Y – (Q) ⟶ A	Rd			12 / 9.6
	STR•A-Q	33	Subtract Q and Store	(A) – (Q) ⟶ Y and A	St	8.4 / 4.8	12 /	12 / 7.2
	RPL•A-Y	25	Replace Subtract	(A) – Y ⟶ Y and A	Rp			
	RPL•Y-Q	35	Replace Subtract Q	Y – (Q) ⟶ Y and A	Rp			18 / .....
	RPL•Y-1	37	Replace Subtract 1	Y – 1 ⟶ Y and A	Rp			
	RPL•A-LP	46	Replace Subtract Log. Prod.	(A) – L (Y (Q)) ⟶ Y and A	Rp			12
MULT./DIV.	MUL	22	Multiply	(Q)•Y ⟶ AQ	Rd	(37.2 — 85.2) VARIABLE		
	ENT•LP	40	Enter Logical Product	L (Y•(Q)) ⟶ A	Rd	8.4 / 6.0	7.2 / 4.8	12 / .....
	STR•LP	47	Store Logical Product	L (A)•(Q) ⟶ Y	St	8.4 / 4.8	12 /	..... / 7.2
	RPL•LP	44	Replace Logical Product	L (Y (Q)) ⟶ Y and A	Rp	..... /	..... /	18 / 12
	DIV	23	Divide	(AQ) ÷ Y; Quot ⟶ Q, Rem ⟶ A	Rd	86.4	.....	
SELECTIVE	SEL•SET	50	Selective Set	Set (A_n) for (Y_n) = 1	Rd	9.6 / 7.2	8.4 / 6.0	12 / 8.4
	SEL•CP	51	Selective Complement	Complement (A_n) for (Y_n) = 1	Rd			
	SEL•CL	52	Selective Clear	Clear (A_n) for (Y_n) = 1	Rd			
	SEL•SU	53	Substitute	(Y_n) ⟶ (A_n) for (Q_n) = 1	Rd			
	RSE•SET	54	Replace Selective Set	Set (A_n) for (Y_n) = 1 ⟶ Y and A	Rp			18 / .....
	RSE•CP	55	Replace Selective Compl	Complement (A_n) for (Y_n)=1 ⟶ Y and A	Rp			
	RSE•CL	56	Replace Selective Clear	Clear (A_n) for (Y_n) = 1 ⟶ Y and A	Rp			
	RSE•SU	57	Replace Substitute	(Y_n) ⟶ (A_n) for (Q_n)=1 ⟶ Y and A	Rp			12

NI: NEXT INSTRUCTION

CLASS: Rd=READ, St=STORE, Rp=REPLACE

*NOTE: LOWER TIME IN EACH BRACKET IS FOR REPEAT MODE TIMES

NORMAL j–DESIGNATOR		
	j = 0	NEXT INSTRUCTION
SKIP	j = 1	Skip NI
QPOS	j = 2	Skip NI if (Q) Pos.
QNEG	j = 3	Skip NI if (Q) Neg.
AZERO	j = 4	Skip NI if (A) = 0
ANOT	j = 5	Skip NI if (A) ≠ 0
APOS	j = 6	Skip NI if (A) Pos.
ANEG	j = 7	Skip NI if (A) Neg.

SPECIAL ADDRESSES	
00000	FAULT, MS TIMEOUT. INTERRUPT
00017	Δ CLOCK INCREMENT
00020–00035	EXTERNAL INTERRUPT
00040–00055	INPUT MONITOR INTERRUPT
00060–00075	OUTPUT MONITOR INTERRUPT
00100–00115	INPUT BUFFER CONTROL
00120–00135	OUTPUT BUFFER CONTROL
00036	Δ CLOCK INTERRUPT

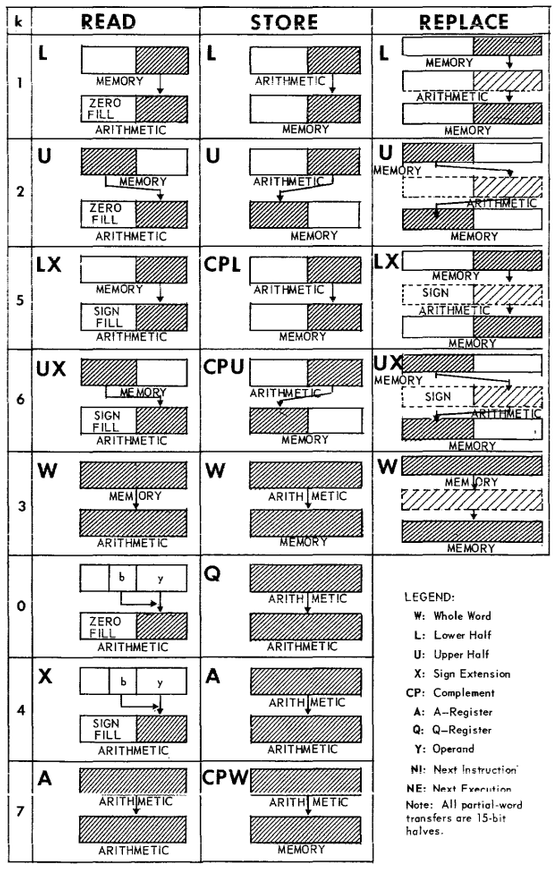
SPECIAL j-DESIGNATOR INTERPRETATION

j	COM•A•Q•A•Q	ADD•Q/SUB•Q	ENT•LP,RPL•LP	DIV.	RPT	JP/RJP	JP(60)/RJP(64)	
	No skip	No skip	No skip	No skip	NE: y=y	Always jump	RIL/SIL	Release/Set Interlock Jump
1	SKIP Always skip	SKIP Always skip	SKIP Always skip	SKIP Always skip	ADV NE: y=y+1	KEY 1 Jump Key 1	RILJP/SILRJP	Release/Set 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 Jump 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_b	KEY 3 Jump Key 3	QNEG	Jump (Q) Neg.
4	YIN Q ≥ Y > A	QZERO Skip (Q) =0	AZERO Skip (A) =0	AZERO Skip (A) =0	R NE: y=y+(B_R)*	STOP Always stop	AZERO	Jump (A)=0
5	YOUT Q< Y≠Y ≤ A	QNOT Skip (Q) ≠0	ANOT Skip (A) ≠ 0	ANOT Skip (A) ≠ 0	ADVR NE: y=y+1+(B_R)*	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.	BACKRNE: y=y–1+(B_R)*	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 NE: y=y+B_R+(B_R)*	STOP 7 Stop Key 7	ANEG	Jump (A) Neg.

*(B_R) Increment if RPL CLASS

JUMP ADDRESS	
k = 0,4	Y = $\bar{Y}$
k = 1,3,5	Y = (Y_L)
k = 2,6	Y = (Y_U)
k = 7	Y = (A_L)

OPERAND INTERPRETATION-NORMAL k-DESIGNATOR



LEGEND:
- W: Whole Word
- L: Lower Half
- U: Upper Half
- X: Sign Extension
- CP: Complement
- A: A–Register
- Q: Q–Register
- Y: Operand
- NI: Next Instruction
- NE: Next Execution

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

MACRO OPERATIONS					
	CLEAR	Clear memory area	V_0: Number of Words	V_1: Start Address	
	ENTRY	Enter Subroutine	$\bar{V}_0$ ⟶ 61000 00000	V_0 ⟶ 61X00 00000	
	EXIT	Exit Subroutine	V_0: Jump condition		
	FD	Literal ⟶ FIELDATA	V_0: Number of Words occupied	V_1: Literal	
	FORM-TEXT	Form, store text ⟶ Buffer	V_0: Buffer	V_1: Init character	V_2: Text
	MOVE	Move, memory to memory	V_0: Number of Words	V_1: Origin	V_2: Destination
	PUT	Store 1 word	V_0: Word	V_1: Destination	
	RESERVE	Reserve memory block	V_0: Number of Words reserved		
	T-TAG	Tag channel and unit	V_0: Channel/Group	V_1: Unit	
	U-TAG	Tag upper word	V_0: Upper tag	V_1: Lower tag/constant	
	CKSTAT	Check status	V_0: I/O Label	V_1: EA/STOP RUN	V_2: EAS/TAKEOVER
	D-TAG	Define drum area	V_0: Drum area name + increment		
	TYPEC	Type contents	V_0: Data to be typed		
	TYPET	Type text	V_0: Text and printer commands		
DECLARATIVE	SEGMENT	Segment programs	V_0: 1st Label, Segment 1	V_1: Segment 2	
	DRUM-AREA	Reserve drum area	V_0: Drum system	V_1: Area name	V_2: Length
	EQUALS	Equate unknown-known	1: Unknown tag	V_0: Known Label/tag	
	MEANS	Sub channel number for name	V_0: Channel number		
	ASSIGN	Declare channel grouping	V_0: Unit names and numbers		
	FACIL	Declare channels, units, fixed/relocatable, number of units			
	COMMENT	Declarative function	V_0: (Notation)		
	DEF-AREA	Identify area	V_0: Area name	V_1: Initial address	V_2: Number of words
	S-TAG	Program linkage	V_0: Upper tag	V_1: Lower tag	
DEBUGGING	CORE-IMAGE	Core area mirror	V_0: Area name	V_1: Initial address	V_2: Key set
	DRUM-IMAGE	Drum area mirror	V_0: Area name	V_1: Initial address	V_2: Key set
	TEST-IMAGE	Compare image	V_0: Area name	V_1: Key set	
	DUMP-REG	Print out A, Q, B_1-B_7	V_0: Key set		
	DUMP-AREA	Print out non-zero words	V_0: Area name	V_1: Key set	
	REX	STOPRUN/TAKEOVER/TERMRUN/EXCHANGE			

INPUT-OUTPUT OPERATOR (SPURT)	
CARD•PUNCH•(BCW) CARD•PUNCHS•(BCW) CARD•PMODE•FD CARD•PMODE•CBIN CARD•PMODE•RBIN CARD•RMODE•FD CARD•RMODE•CBIN CARD•RMODE•RBIN CARD•STACK 1 CARD•STACK 2 CARD•READNT CARD•READ•(BCW) CARD•TRANS•(BCW) CARD•MREAD•(#)•(BCW)	TYPEC•(ADDRESS) TYPET•(TEXT) TYPE•(#)•(ADDRESS)
DRUM•(NAME)•WRITE•(BCW)•(ADDRESS) DRUM•(NAME)•READ•(BCW)•(ADDRESS) DRUM•(NAME)•BREAD•(BCW)(ADDRESS) DRUM•(NAME)•LOCATE•(ADDRESS)•(ID) DRUM•(NAME)•BLOCATE•(ADDRESS)•(ID) DRUM•(NAME)•SEARCH•(BCW)•(ADDRESS)•(ID) DRUM•(NAME)•BSEARCH•(BCW)•(ADDRESS)•(ID)	CTAPE•HIBCD•(NAME) CTAPE•LOBCD•(NAME) CTAPE•HIBIN•(NAME) CTAPE•LOBIN•(NAME) CTAPE•WRITE•(NAME)•(BCW) CTAPE•ENDFILE•(NAME) CTAPE•READ•(NAME)•(BCW) CTAPE•SEARCH•(NAME)•(BCW)•(ID) CTAPE•REWIND•(NAME) CTAPE•MOVEB•(NAME)•(N) CTAPE•MOVEF•(NAME)•(N) CTAPE•BACKFILE•(NAME) CTAPE•RWI•(NAME)
DISC•(NAME)•READ•(BCW)•(ADDRESS) DISC•(NAME)•WRITE•(BCW)•(ADDRESS) DISC•(NAME)•SEARCH•(BCW)•(ADDRESS)•(SL)•(ID) DISC•(NAME)•WSEARCH•(BCW)•(ADDRESS)•(SL)•(ID)	MTAPE•WRITE•(NAME)•(BCW) MTAPE•READF•(NAME)•(BCW) MTAPE•READB•(NAME)•(BCW) MTAPE•SEARCHF•(NAME)•(BCW)•(ID) MTAPE•SEARCHB•(NAME)•(BCW)•(ID) MTAPE•SEARCHF•(NAME)•(BCW)•(ID)•(MASK) MTAPE•SEARCHB•(NAME)•(BCW)•(ID)•(MASK) MTAPE•REWIND•(NAME) MTAPE•RWI•(NAME) MTAPE•ENDFILE•(NAME) MTAPE•MOVEF•(NAME)•(N) MTAPE•MOVEB•(NAME)•(N)
PIN•(NAME)•(LINES/PAGE)•(TOP)•(BOT MARGIN) PRINT•(NAME)•(BUFF BASE ADD)•(#LINES ADV)	
CONSOLE•HOLD CONSOLE•RELEASE	ACCEPT•(#CHAR)•(BUFFER) LOAD•(SEGMENT NAME)

COMPUTER INSTRUCTION WORD FORMAT

FUNCTION	SPECIAL *	OPERAND INTERPRET	MODIFIER	OPERAND/OPERAND ADDRESS
f	j	k	b	y
29 — 24	23 — 21	20 — 18	17 — 15	14 — 00

* REGISTER, SKIP/JUMP; OR REPEAT MODIFICATION DESIGNATOR

INPUT-OUTPUT INSTRUCTION WORD FORMAT

FUNCTION	CHANNEL	*	B-Reg.	CONTROL WORD ADDRESS
f	↑	↑	b	y
29 — 24	23 — 20	19 — 18	17 — 15	14 — 00

* APPLICABLE TO f=73–76 ONLY

K = 0 (00): $\bar{Y}$ — Buffer Address (001xxt$\bar{i}$)$_L$
K = 1 (01): Y_L — Buffer Address (001xxt$\bar{i}$)$_L$
K = 3 (11): Y_W — Buffer Address (001xxt$\bar{i}$)$_W$
} xx = 00 for f = 73, 75; xx = 20 for f = 74, 76

MAGNETIC DRUM

FUNCTION WORD

FC	BEGINNING j DRUM ADDRESS
29 — 24	22 — 00

DRUM FUNCTIONS

MNEM	OC	DESCRIPTION	FC
WRITE	0	CONTINUOUS WRITE: START AT SPECIFIED ADDRESS	02
READ	1	CONTINUOUS READ: START AT SPECIFIED ADDRESS	42
BREAD	2	BLOCK READ: READ TILL EOB; TERMINATE	52
LOCATE	3	SEARCH: SEARCH TILL FIND; TERMINATE	45
BLOCATE	4	BLOCK SEARCH: SEARCH TILL FIND/EOB; TERMINATE	55
SEARCH	5	SEARCH-READ: SEARCH TILL FIND, THEN CONT. READ	46
BSEARCH	6	BLOCK SEARCH-READ: SEARCH; FIND; READ TILL EOB	56
		BOOTSTRAP: REFERENCE ADDRESS 000000	40/50
		TERMINATE: TERMINATE OPERATION	23/33

STATUS WORDS

DRUM SUBSYSTEM INTERRUPT CODE		REX STATUS CODES ⟶ A_{05}–A_{00}		CONSOLE PRINTOUT
SEARCH FIND	05	00	OPERATION NOT COMPLETE	NONE
NORMAL COMPLETION	40	* 01	OPERATION COMPLETE	NONE
END OF FILE	34	02	END OF FILE WITHOUT FIND	NONE
END OF BLOCK	04	* 03	END OF BLOCK WITHOUT FIND	NONE
		04	BUFFER FILLED BEFORE EOB	NONE
		* 05	END OF FILE BEFORE COMPLETE TRANSFER	NONE
		07	ILLEGAL ERROR	MSG 251 P
ILLEGAL FUNCTION	50	10	ILLEGAL FUNCTION	MSG 251 P
ILLEGAL ADDRESS	54	11	ILLEGAL ADDRESS	MSG 251 P
PARITY ERROR CONT READ	64	* 12	READ ERROR	MSG 251 P
CHARACTER COUNT ERROR	30	* 12	READ ERROR	MSG 251 P
SEQUENCE ERROR	60	* 12	READ ERROR	MSG 251 P
WRITE FAULT	14	* 13	WRITE ERROR	MSG 251 P
OVERFLOW PARITY ERROR	06	* 14	OVERFLOW ERROR AFTER READ	MSG 251 P
PARITY ERROR,NON-CONT READ	07	* 15	OVERFLOW ERROR BEFORE READ	MSG 251 P
WRITE CHAR COUNT ERROR	70	17	DRUM DOWN	MSG 251 P
		16	ILLEGAL PARAMETER	MSG 252 P