

1 MIX Alphanumeric Codes							
SYMBOL	CODE						
MIX and Printer	Computer and Magnetic Tape	Paper Tape					
		Channel					
		X	0	C	8	4	2 1
							Punch Card
(Space)	00			✓			(Blank)
A	01	X	0				12 1
B	02	X	0			2	12 2
C	03	X	0	✓		2 1	12 3
D	04	X	0		4		12 4
E	05	X	0	✓	4	1	12 5
F	06	X	0	✓	4	2	12 6
G	07	X	0		4	2 1	12 7
H	10	X	0		8		12 8
I	11	X	0	✓	8	1	12 9
Δ	12	X					11
J	13	X		✓		1	11 1
K	14	X		✓		2	11 2
L	15	X				2 1	11 3
M	16	X		✓	4		11 4
N	17	X			4	1	11 5
O	20	X			4	2	11 6
P	21	X		✓	4	2 1	11 7
Q	22	X		✓	8		11 8
R	23	X			8	1	11 9
Σ	24		0		8	2	0 2-8
Π	25		0	✓		1	0 1
S	26		0	✓		2	0 2
T	27		0			2 1	0 3
U	30		0	✓	4		0 4
V	31		0		4	1	0 5
W	32		0		4	2	0 6
X	33		0	✓	4	2 1	0 7
Y	34		0	✓	8		0 8
Z	35		0		8	1	0 9
0 (Zero)	36			✓	8	2	0
1	37					1	1
2	40					2	2
3	41			✓		2 1	3
4	42				4		4
5	43			✓	4	1	5
6	44			✓	4	2	6
7	45				4	2 1	7
8	46				8		8
9	47			✓	8	1	9
.	50	X	0		8	2 1	12 3-8
,	51	X	0	✓	8	4	12 4-8
(52	X	0		8	4 1	12 5-8
)	53	X	0		8	4 2	12 6-8
+	54	X		✓	8	2 1	11 3-8
-	55	X			8	4	11 4-8
*	56	X		✓	8	4 1	11 5-8
/	57	X		✓	8	4 2	11 6-8
=	60		0	✓	8	2 1	0 3-8
\$	61		0		8	4	0 4-8
<	62		0	✓	8	4 1	0 5-8
>	63		0	✓	8	4 2	0 6-8
@	64				8	2 1	3-8
;	65			✓	8	4	4-8
:	66				8	4 1	5-8
'	67				8	4 2	6-8

INSTR.FORMAT				OPERATION		
±	AA	I	F C	ABR	DF	NAME
±	aaaa	i	L:R 00	NOP	0	NO OPERATION
±	aaaa	i	L:R 01	ADD	0:5	ADD
±	aaaa	i	06 01	FADD		FLOATING ADD
±	aaaa	i	L:R 02	SUB	0:5	SUBTRACT
±	aaaa	i	06 02	FSUB		FLOATING SUBTRACT
±	aaaa	i	L:R 03	MUL	0:5	MULTIPLY
±	aaaa	i	06 03	FMUL		FLOATING MULTIPLY
±	aaaa	i	L:R 04	DIV	0:5	DIVIDE
±	aaaa	i	06 04	FDIV		FLOATING DIVIDE
±	aaaa	i	00 05	NUM		CONVERT TO NUMERIC
±	aaaa	i	01 05	CHAR		CONVERT TO CHARACTERS
±	aaaa	i	02 05	HLT		HALT
±	aaaa	i	03 05	AND		LOGICAL PRODUCT
±	aaaa	i	04 05	OR		LOGICAL SUM
±	aaaa	i	05 05	XOR		LOGICAL DIFFERENCE
±	aaaa	i	06 05	FLOT		FIXED TO FLOAT
±	aaaa	i	07 05	FIX		FLOATING TO FIXED
±	aaaa	i	10 05	NEG		LOGICAL NEGATION
±	aaaa	i	11 05	INT		INTERRUPT
±	aaaa	i	12 05	XCH		EXCHANGE A AND X
±	aaaa	i	13 05	XEQ		EXECUTE
±	aaaa	i	00 06	SLA		SHIFT LEFT A
±	aaaa	i	01 06	SRA		SHIFT RIGHT A
±	aaaa	i	02 06	SLAX		SHIFT LEFT AX
±	aaaa	i	03 06	SRAX		SHIFT RIGHT AX
±	aaaa	i	04 06	SLC		SHIFT LEFT AX CIRCULARLY
±	aaaa	i	05 06	SRC		SHIFT RIGHT AX CIRCULARLY
±	aaaa	i	06 06	SLB		SHIFT LEFT LOGICAL AX
±	aaaa	i	07 06	SRB		SHIFT RIGHT LOGICAL AX
±	aaaa	i	N 07	MOVE	1	MOVE WORDS
±	aaaa	i	L:R 10+[r]	LD[r]	0:5	LOAD
±	aaaa	i	L:R 20+[r]	LD[r]N	0:5	LOAD r NEGATIVE
±	aaaa	i	L:R 30+[r]	ST[r]	0:5	STORE
±	aaaa	i	L:R 40	STJ	0:2	STORE J
±	aaaa	i	L:R 41	STZ	0:5	STORE ZERO
±	aaaa	i	U 42	JBUS	0	JUMP BUSY
±	aaaa	i	U 43	IOC	0	I/O CONTROL
±	aaaa	i	U 44	IN	0	INPUT
±	aaaa	i	U 45	OUT	0	OUTPUT
±	aaaa	i	U 46	JRED	0	JUMP READY
±	aaaa	i	00 47	JMP		JUMP
±	aaaa	i	01 47	JSJ		JUMP SAVE J
±	aaaa	i	02 47	JOV		JUMP ON OVERFLOW
±	aaaa	i	03 47	JNOV		JUMP ON NO OVERFLOW
±	aaaa	i	04 47	JL		JUMP ON LESS
±	aaaa	i	05 47	JE		JUMP ON EQUAL
±	aaaa	i	06 47	JG		JUMP ON GREATER
±	aaaa	i	07 47	JGE		JUMP ON GREATER-OR-EQUAL
±	aaaa	i	10 47	JNE		JUMP ON UNEQUAL
±	aaaa	i	11 47	JLE		JUMP ON LESS-OR-EQUAL
±	aaaa	i	00 50+[r]	J[r]N		JUMP r NEGATIVE
±	aaaa	i	01 50+[r]	J[r]Z		JUMP r ZERO
±	aaaa	i	02 50+[r]	J[r]P		JUMP r POSITIVE
±	aaaa	i	03 50+[r]	J[r]NN		JUMP r NONNEGATIVE
±	aaaa	i	04 50+[r]	J[r]NZ		JUMP r NONZERO
±	aaaa	i	05 50+[r]	J[r]NP		JUMP r NONPOSITIVE
±	aaaa	i	06 50+[r]	J[r]E		JUMP r EVEN
±	aaaa	i	07 50+[r]	J[r]O		JUMP r ODD
±	aaaa	i	00 60+[r]	INC[r]		INCREASE r
±	aaaa	i	01 60+[r]	DEC[r]		DECREASE r
±	aaaa	i	02 60+[r]	ENT[r]		ENTER r
±	aaaa	i	03 60+[r]	ENN[r]		ENTER NEGATIVE r
±	aaaa	i	04 60+[r]	CPM[r]		COMPARE r WITH M
±	aaaa	i	L:R 70+[r]	CMP[r]	0:5	COMPARE r
±	aaaa	i	06 70	FCMP		FLOATING COMPARE

[r]: rA=0, r1, r2, r3, r4, r5, r6, rX=7, i: i1:i2, 7 is indirect addressing