


WAGNER

 SPERRY RAND

UNIVAC

9200

9200 II

9300

9300 II

SYSTEMS

**REFERENCE  
CARD**

# INSTRUCTIONS

TYPE	MNEMONIC	FUNCTION	HEXADECMAL OPERATION CODE	FORMAT	
Arithmetic	AH	ADD HALF-WORD	AA	RX	
	AI	ADD IMMEDIATE	A6	SI	
	AP	ADD (PACKED) DECIMAL	FA	SS2	
	DP	DIVIDE (PACKED) DECIMAL	FD	SS2	
	MP	MULTIPLY (PACKED) DECIMAL	FC	SS2	
	SH	SUBTRACT HALF-WORD	AB	RX	
	SP	SUBTRACT (PACKED) DECIMAL	FB	SS2	
	ZAP	ZERO ADD (PACKED) DECIMAL	F8	SS2	
	Branch	BAL	BRANCH AND LINK	45	RX
		BC	BRANCH ON CONDITION	47	RX
Comparison	CH	COMPARE HALF-WORD	49	RX	
	CLC	COMPARE LOGICAL CHARACTER	D5	SS1	
	CLI	COMPARE LOGICAL IMMEDIATE	95	SI	
	CP	COMPARE (PACKED) DECIMAL	F9	SS2	
	TM	TEST UNDER MASK	91	SI	
Data Manipulation	ED	EDIT	DE	SS1	
	PACK	PACK	F2	SS2	
	TR	TRANSLATE	DC	SS1	
	UNPK	UNPACK	F3	SS2	
Data Transfer	LH	LOAD HALF-WORD	48	RX	
	MVC	MOVE CHARACTERS	D2	SS1	
	MVI	MOVE IMMEDIATE DATA	92	SI	
	MVN	MOVE NUMERICS	D1	SS1	
	MVO	MOVE WITH OFFSET	F1	SS2	
	STH	STORE HALF-WORD	40	RX	
	Display	HPR	HALT AND PROCEED	A9	SI
I/O	TIO	TEST I/O	A5	SI	
	XIOF	EXECUTE I/O FUNCTION	A4	SI	
Logical	NC	AND CHARACTERS	D4	SS1	
	NI	AND IMMEDIATE DATA	94	SI	
	OC	OR CHARACTERS	D6	SS1	
	OI	OR IMMEDIATE DATA	96	SI	
Supervisor	LPSC	LOAD PROGRAM STATE CONTROL	A8	SI	
	SPSC	STORE PROGRAM STATE CONTROL	A0	SI	
	SRC	SUPERVISOR REQUEST	A1	SI	

# INSTRUCTION LOGIC AND TIMING

HEXADECMAL OPERATION CODE	MNEMONIC	LOGIC	TIMES (MICROSECONDS) <sup>①</sup>
40	STH	(R1) → OP2	20.4
45	BAL	Branch to OP2; store address at R1	18
47	BC	if match → I2	15.6 if no branch 18 if branch
48	LH	(OP2) → R1	20.4
49	CH	(R1) : (OP2)	20.4
91	TM	(OP1) : I2	16.8 if no match or match on zero 19.2 if partial or full match
92	MVI	I2 → OP1	16.8
94	NI	(OP1) AND I2 → OP1	16.8
95	CLI	(OP1) : I2	16.8
96	OI	(OP1) OR I2 → OP1	16.8
A0	SPSC	PSC → OP1	24
A1	SRC	OP2 → SRC	12
A4	XIOF	initiate I/O operations	18 for integrated I/O units; variable for multiplexer
A5	TIO	status → OP1	18 for integrated I/O units; variable for multiplexer
A6	AI	(OP1) + I2 → OP1	19.2
A8	LPSC	(OP1) → PSC	24 to load entire PSC word; 18 otherwise
A9	HPR	Display OP1	14.4
AA	AH	(R1) + (OP2) → R1	20.4
AB	SH	(R1) - (OP2) → R1	20.4
D1	MVN	(OP2) → OP1	16.8 + 8.4(N) <sup>②</sup>
D2	MVC	(OP2) → OP1	16.8 + 8.4(N)
D4	NC	(OP1) AND (OP2) → OP1	16.8 + 8.4(N)
D5	CLC	(OP1) - (OP2)	25.2 + 8.4(N) <sup>③</sup>
D6	OC	(OP1) OR (OP2) → OP1	16.8 + 8.4(N)
DC	TR	translate (OP1) using (OP2) → OP1	16.8 + 14.4(N)
DE	ED	(OP2) → OP1	See ④
F1	MVO	(OP2) → OP1	25.2 + 3.6(N <sub>2</sub> ) + 6(N <sub>1</sub> )
F2	PACK	(OP2) → OP1	25.2 + 3.6(N <sub>2</sub> ) + 4.8(N <sub>1</sub> )
F3	UNPK	(OP2) → OP1	21.6 + 7.2(N <sub>2</sub> ) + 4.8(N <sub>1</sub> )
F8	ZAP	0 → OP1; (OP2) → OP1	26.4 + 3.6(N <sub>2</sub> ) + 4.8(N <sub>1</sub> )
F9	CF	(OP1) - (OP2)	26.4 + 3.6(N <sub>2</sub> ) + 4.8(N <sub>1</sub> )
FA	AF	(OP1) + (OP2) → OP1	26.4 + 3.6(N <sub>2</sub> ) + 4.8(N <sub>1</sub> )
FB	SP	(OP1) - (OP2) → OP1	26.4 + 3.6(N <sub>2</sub> ) + 4.8(N <sub>1</sub> )
FC	MP	(OP2) × (OP1) → OP1	See ④
FD	DP	(OP1) ÷ (OP2) → OP1	See ④

① Timing for all instructions assumes no indexing. Add 3.6 microseconds for each indexing operation. Timing is given for UNIVAC 9300 Systems; for UNIVAC 9200 Systems, multiply by two.

② N, N<sub>1</sub>, and N<sub>2</sub> equal the number of bytes specified in the length of the operand.

③ N<sub>2</sub> equals the number of most significant bytes that will compare identically between OP1 and OP2 in the Compare Logical Character instruction.

④ See UNIVAC 9200/9200 II/9300/9300 II Systems Card Assembler Programmers Reference, UP-4092 (current version); or UNIVAC 9200/9200 II/9300/9300 II Systems Tape/Disc Assembler Programmers Reference, UP-7508 (current version).

# INSTRUCTION FORMATS

INSTRUCTION TYPE	SPECIFICATION TYPE	OPERAND	
		OP1	OP2
RX	Complete	R1	,D2(B2)
	Relative address	R1	,tag
SI	Complete	D1(B1)	,I2
	Relative address	symbol	,I2
SS1	Complete	D1(L1, B1)	,D2(B2)
	Relative address	symbol(L1)	,tag
	Implied length	D1(B1)	,D2(B2)
	Relative address and length	symbol	,tag
SS2	Complete	D1(L1, B1)	,D2(L2, B2)
	Relative address	symbol(L1)	,tag(L2)
	Implied length	D1(B1)	,D2(B2)
	Relative address and length	symbol	,tag

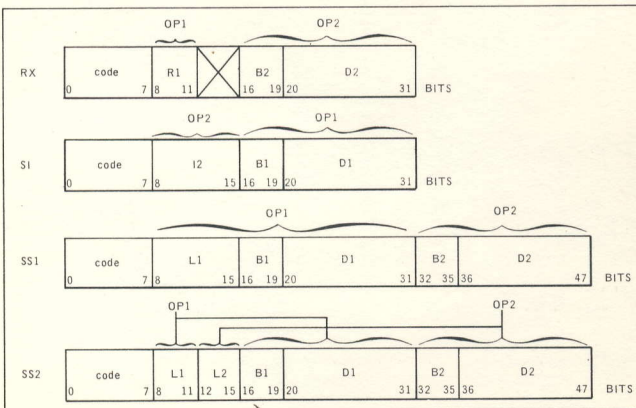
If the length is not specified in an implied operand, it is determined by the assembler. If the length is not specified but the address is specified, the assembler assumes a length of one.

# ABBREVIATIONS

SYMBOL	MEANING
B1	The number of the general register that holds the base address of operand 1.
B2	The number of the general register that holds the base address of operand 2.
code	The mnemonic operation code of the instruction.
D1	The displacement from the base address of operand 1.
D2	The displacement from the base address of operand 2.
I2	The immediate data used as operand 2 in SI format instructions.
L1	The length of operand 1 as stated in source code.*
L2	The length of operand 2 as stated in source code.*
OP1	Operand 1.
OP2	Operand 2.
R1	The number of the general register that holds operand 1 in RX format instructions.
symbol	The expression or symbolic label used as operand 1 or the label of an instruction.
tag	The expression or symbolic label used as operand 2.

\* The length is coded as the true length of the operand, not the length less one as required by the object code. The assembler makes the appropriate reduction by one when converting source code to object code.

# INSTRUCTION OBJECT CODE FORMATS



Using abbreviations described in the abbreviations chart, found elsewhere on this code card.

**EXTENDED OPERATION CODES**  
(Not available with card assembler)

MNEMONIC	FUNCTION	HEXADECIMAL OPERATION CODE, R1	FORMAT
B NOP	BRANCH NO OPERATION	47 F 47 O	RX RX
USED AFTER COMPARISON INSTRUCTIONS			
BH BL BE	BRANCH IF HIGH BRANCH IF LOW BRANCH IF EQUAL	47 2 47 4 47 8	RX RX RX
BNH BNL BNE	BRANCH IF NOT HIGH BRANCH IF NOT LOW BRANCH IF NOT EQUAL	47 D 47 B 47 7	RX RX RX
USED AFTER TEST UNDER MASK INSTRUCTIONS			
BO BZ BM	BRANCH IF ALL ONES BRANCH IF ALL ZEROS BRANCH IF MIXED	47 1 47 8 47 4	RX RX RX
BNO BNZ BNM	BRANCH IF NOT ALL ONES BRANCH IF NOT ALL ZEROS BRANCH IF NOT MIXED	47 E 47 7 47 B	RX RX RX
USED AFTER ARITHMETIC INSTRUCTIONS			
BO BZ BM	BRANCH IF OVERFLOW BRANCH IF ZERO BRANCH IF MINUS	47 1 47 8 47 4	RX RX RX
BP BNO BNZ	BRANCH IF POSITIVE BRANCH IF NO OVERFLOW BRANCH IF NOT ZERO	47 2 47 E 47 7	RX RX RX
BNM BNP	BRANCH IF NOT MINUS BRANCH IF NOT POSITIVE	47 B 47 D	RX RX

**CONDITION CODE SETTINGS**

Instr.	Condition Codes/Conditions			
	0(00)	1(01)	2(10)	3(11)
SH (AB) AH (AA) AI (A6) AP (FA) SP (FB)	Result=Zero	Result=Neg.	Result Positive	Overflow
CH (49) CLI (95) CLC (D5) CP (F9)	(R <sub>1</sub> ) = (OP <sub>2</sub> ) (OP <sub>1</sub> ) = I <sub>2</sub> (OP <sub>1</sub> ) = (OP <sub>2</sub> ) (OP <sub>1</sub> ) = (OP <sub>2</sub> )	(R <sub>1</sub> ) < (OP <sub>2</sub> ) (OP <sub>1</sub> ) I <sub>2</sub> (OP <sub>1</sub> ) < (OP <sub>2</sub> ) (OP <sub>1</sub> ) < (OP <sub>2</sub> )	(R <sub>1</sub> ) > (OP <sub>2</sub> ) (OP <sub>1</sub> ) > I <sub>2</sub> (OP <sub>1</sub> ) > (OP <sub>2</sub> ) (OP <sub>1</sub> ) > (OP <sub>2</sub> )	
ZAP (F8)	(OP <sub>2</sub> ) = β	(OP <sub>2</sub> ) Neg.	(OP <sub>2</sub> ) Pos.	
NI (94) NC (D4) OI (96) OC (D6)	Result=Zero	Result≠Zero		
TM (91)	No match or mask = β	Partial match		Full match
XIOF (A4)	Accepted	Status Pending	Busy	Rejected
TIO (A5)	Available	Valid Status	Busy	Rejected

**BC INSTRUCTION**

R1	TESTS
8	CC 0
4	CC 1
2	CC 2
1	CC 3
15	unconditional branch skip
0	

**CHARACTER CODES**

63 CHARACTER BAR GRAPHIC	EBCDIC	HEX	HOLLERITH	COMPRESSED
A	11000001	C1	12-1	00110001 31
B	11000010	C2	12-2	00110001 51
C	11000011	C3	12-3	00110001 11
D	11000100	C4	12-4	00110001 21
E	11000101	C5	12-5	01000001 41
F	11000110	C6	12-6	01110001 71
G	11000111	C7	12-7	01100001 81
H	11001000	C8	12-8	00001001 01
I	11001001	C9	12-9	10000001 81
J	11010001	D1	11-1	00110010 31
K	11010010	D2	11-2	01010010 52
L	11010011	D3	11-3	00100010 12
M	11010100	D4	11-4	00100010 22
N	11010101	D5	11-5	01000010 42
O	11010110	D6	11-6	01110010 72
P	11010111	D7	11-7	01100010 82
Q	11011000	D8	11-8	00001010 02
R	11011001	D9	11-9	10000010 82
S	11100010	E2	0-2	01010100 54
T	11100011	E3	0-3	00010100 14
U	11100100	E4	0-4	00100100 24
V	11100101	E5	0-5	01000100 44
W	11100110	E6	0-6	01110100 74
X	11100111	E7	0-7	01100100 84
Y	11101000	E8	0-8	00001100 0C
Z	11101001	E9	0-9	10000100 94
1	11110001	F1	1	00110000 30
2	11110010	F2	2	01010000 50
3	11110011	F3	3	00010000 10
4	11110100	F4	4	00100000 20
5	11110101	F5	5	01000000 40
6	11110110	F6	6	01110000 70
7	11110111	F7	7	01100000 80
8	11111000	F8	8	00001000 00
9	11111001	F9	9	10000000 80
0	11110000	F0	0	00000100 04
b	01000000	40	b	00000000 00
&	01010000	50	12	00000001 00
-	01100000	60	11	00000010 00
.	01100001	61	0-1	00110100 34
^	01001010	4A	12-8-2	01011001 59
!	01011010	5A	11-8-2	01011010 5A
:	01111010	7A	8-2	01011000 58
;	01001011	4B	12-8-3	00011001 19
?	01011011	5B	11-8-3	00011010 1A
=	01101011	6B	0-8-3	00011100 1C
>	01110111	7B	8-3	00011000 18
@	01001100	4C	12-8-4	00101001 2A
*	01011100	5C	11-8-4	00101010 29
%	01101100	6C	0-8-4	00101100 2C
~	01111100	7C	8-4	00101000 28
~	01001101	4D	12-8-5	01001001 49
~	01011101	5D	11-8-5	01001010 4A
~	01101101	6D	0-8-5	01001100 4C
~	01111101	7D	8-5	01001000 48
~	01001110	4E	12-8-6	01111001 79
~	01011110	5E	11-8-6	01111010 7A
~	01101110	6E	0-8-6	01111100 7E
~	01111110	7E	8-6	01111000 78
~	01001111	4F	12-8-7	01101001 69
~	01011111	5F	11-8-7	01101010 6A
~	01101111	6F	0-8-7	01101100 6C
~	01111111	7F	8-7	01101000 68

**CONSTANT CHARACTERISTICS**

CONSTANT TYPE	EXPLICIT LENGTH	IMPLICIT LENGTH	TRUNCATION OR PADDING	VALUE PADDED	ALIGNMENT	CONSTANT FORM
C	variable 1-256	maximum 256	on right side	blanks	none	character (EBCDIC)
X	variable 1-256	maximum 256	on left side	hexadecimal 0	none	hexadecimal digits
P*	variable 1-16	maximum 16	on left side	hexadecimal 0	none	packed decimal
Z*	variable 1-16	maximum 16	on left side	EBCDIC 0	none	unpacked decimal
H*	variable 1-2	2	on left side	hexadecimal 0	half-word**	binary
Y	variable 1-2	2	on left side	hexadecimal 0	half-word**	binary address
S*	2	2	none	none	half-word**	base and displacement

\* Not available with card assembler.  
\*\* Half-word alignment takes place only if implicit lengths are used.

**SIGN CONVENTIONS**

HEXADECIMAL DIGIT	BINARY REPRESENTATION	SIGN VALUE
0 through 8	0000 through 1000	POSITIVE
9	1001	NEGATIVE
A	1010	POSITIVE
B	1011	NEGATIVE
C	1100	POSITIVE
D	1101	NEGATIVE
E	1110	POSITIVE
F	1111	POSITIVE

\*Alternate sign convention.

**OPERATING SYSTEM ADDRESS TABLE**

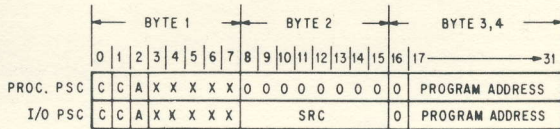
LOCATION	CONTENTS
(Decimal)	
260-261	PU Table Base Address
262-263	LU Table Base Address
264-265	Exec Activity Sum
266-267	Display Subroutine Address
268-269	Boundary Table Base Address
270-271	Interrupt Table Base Address
272-273	SRC Table Base Address
274-275	Re-entry Routine Address
276-277	Keyin Table Base Address
278	Version Number
279	UPS1 Byte
280-285	Date
289	Two times the channel number of an 8410 disc unit
291	Two times the channel number of a magnetic tape unit
334-335	Address of 256-byte translation table used to translate cards in the control stream

**LOGICAL UNIT TABLE FORMAT**

LOGICAL UNIT NUMBER*	0							1							
	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6
00	Swap Indicator	Alternate LU Table Entry							PU Table Pointer						
01															
02															
.															
3F															

\*Hexadecimal Code

# PROGRAM STATE CONTROL STORAGE



CC = Condition Code      A-1 = ASCII  
 A = ASCII Control Code      A=0 = EBCDIC

## PROGRAM STATE CONTROL

Load Action*		PSC Selection		Next Instr. Control*		Alter/Display Action*	
Instr. Bit	Action	Instr. Bit	PSC	Instr. Bit	Control PSC	Instr. Bit	Action
8 9		10		11		12 13	
0 0	None	0	Proc.	0	Proc.	1 0	Restrict
0 1	Fullword	1	I/O	1	I/O	0 1	Remove
1 0	ASCII Off						Restriction
1 1	ASCII On						

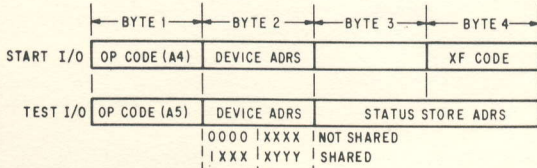
\*Load State Instruction only

## INTERNAL I/O COMMANDS

I/O Device	DA	Instruction Bit/Function								
		24	25	26	27	28	29	30	31	
Card Reader	1				Inh. Inf.		Image	Read		
Read/Punch	2					Stkr Sel.				Punch
Printer	3	48 Char. bar (24-1)	Numeric						*	*

\*30 and 31 ⇒ Space; 31 only ⇒ Print and Space

## I/O INSTRUCTION FORMAT



XXXX - Subchannel Address  
 YYY - Device Number

## I/O STATUS BYTE

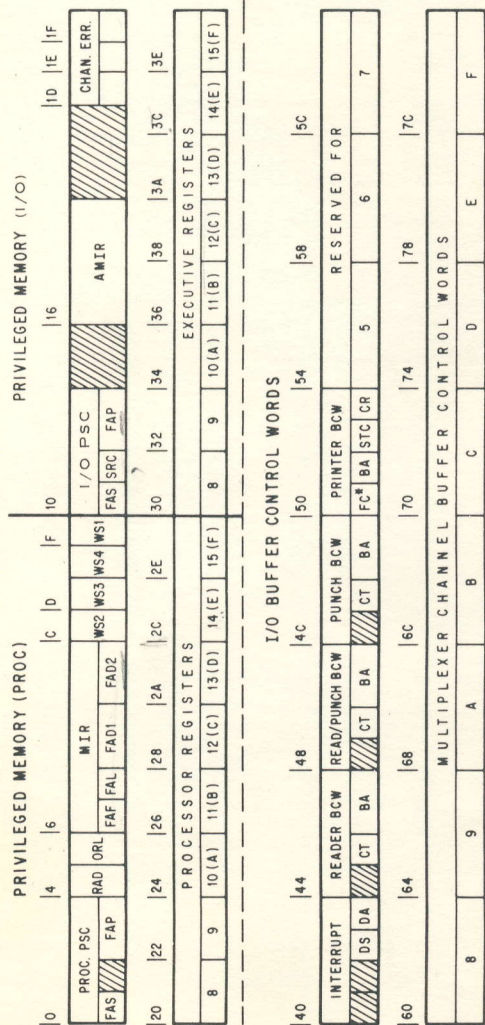
I/O Device/Status Indication*			
Printer (DA=3)	Card Punch or Read Punch (DA=2)	Card Reader (DA=1)	Status Byte Bit*
Abnormal or Not Ready	Stkr Jam, Intlk, Punch Entry, or Exit Chk Error	Stkr Jam, Cont. Par. Err., or Photocell Chk Err.	0
Paper Runaway		Misfeed, Not Ready, Hopper Empty, or Stacker Full	1
Memory Overload	Punch Chk Err.		2
Data Par. or Cont. Par. Err.	Data Par. or Cont. Par. Err.		3
Bar Switch in Err.	Photocell Chk Err.		4
Interrupt Pending	Interrupt Pending	Interrupt Pending	5
Form Overflow	Hopper Empty or Stkr Full		6
Paper Low			7

\*All #'s → function performed as specified

## HEXADECIMAL VALUES

HEX DIGIT	VALUE BY DIGIT POSITION																			
	4	3	2	1	0	4	3	2	1	0	4	3	2	1	0					
0	0	0	0	0	0	65,536	4096	256	16	1	131,072	8192	512	32	2	196,608	12,288	768	48	3
1	0	0	0	0	0	65,536	4096	256	16	1	131,072	8192	512	32	2	196,608	12,288	768	48	3
2	0	0	0	0	0	65,536	4096	256	16	1	131,072	8192	512	32	2	196,608	12,288	768	48	3
3	0	0	0	0	0	65,536	4096	256	16	1	131,072	8192	512	32	2	196,608	12,288	768	48	3
4	0	0	0	0	0	65,536	4096	256	16	1	131,072	8192	512	32	2	196,608	12,288	768	48	3
5	0	0	0	0	0	65,536	4096	256	16	1	131,072	8192	512	32	2	196,608	12,288	768	48	3
6	0	0	0	0	0	65,536	4096	256	16	1	131,072	8192	512	32	2	196,608	12,288	768	48	3
7	0	0	0	0	0	65,536	4096	256	16	1	131,072	8192	512	32	2	196,608	12,288	768	48	3
8	0	0	0	0	0	65,536	4096	256	16	1	131,072	8192	512	32	2	196,608	12,288	768	48	3
9	0	0	0	0	0	65,536	4096	256	16	1	131,072	8192	512	32	2	196,608	12,288	768	48	3
A	0	0	0	0	0	65,536	4096	256	16	1	131,072	8192	512	32	2	196,608	12,288	768	48	3
B	0	0	0	0	0	65,536	4096	256	16	1	131,072	8192	512	32	2	196,608	12,288	768	48	3
C	0	0	0	0	0	65,536	4096	256	16	1	131,072	8192	512	32	2	196,608	12,288	768	48	3
D	0	0	0	0	0	65,536	4096	256	16	1	131,072	8192	512	32	2	196,608	12,288	768	48	3
E	0	0	0	0	0	65,536	4096	256	16	1	131,072	8192	512	32	2	196,608	12,288	768	48	3
F	0	0	0	0	0	65,536	4096	256	16	1	131,072	8192	512	32	2	196,608	12,288	768	48	3

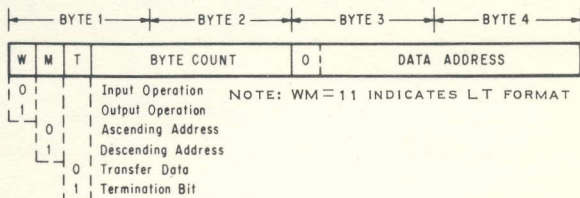
## MEMORY LAYOUT



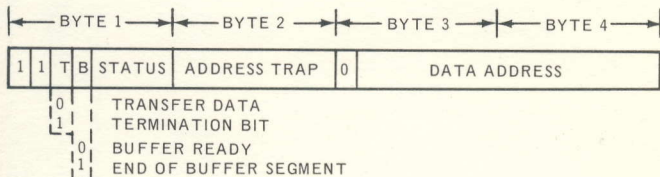
\*FC - Bits 4 5 6 7  
 0 0 0 1 Space one line  
 0 0 1 0 Space two lines  
 1 X X X Paper Loop Control

# MULTIPLEXER CHANNEL

## BASIC BUFFER CONTROL WORD



## LT BUFFER CONTROL WORD



## STATUS BYTE

Bit	0	1	2	3	4	5	6	7
Detail	Att.	Stat. Mod.	Cont. Unit End	Busy	Chan. End	Dev. End	Unit Chk.	Unit Except.

## CHANNEL ERROR STATUS

Mem. Loc.	Bit Pos.	Signal Function	Mem. Loc.	Bit Pos.	Signal Function
001D	0	Interface Error	001E	0	Status In
	1	Device Address Parity Error		1	Service Out
	2	Bus in Parity Error		2	Service In
	3	Address Out		3	TIME OUT REQUEST
	4	Select Out		4	Suppress Out
	5	Operational In		5	Select In
	6	Address In		6	Terminate/KØ FF
7	Command Out	7	Input Direction/K1 FF		
			001F	0-7	Device Address Register

## MULTIPLEXER CHANNEL COMMANDS

Function	XF Code Bits								
	P	0	1	2	3	4	5	6	7
Test	P	X	X	X	X	0	0	0	0
Sense	P					0	1	0	0
Write	P							0	1
Read	P							1	0
Control	P							1	1
Read Backward	P					1	1	0	0
Reserved for Chan. Cont.	P	X	X	X	X	1	0	0	0

X = Variable to Control Units  
 P = Parity Bit (Odd)

## FIRST SENSE BYTE

Bit	Indication
P	Parity (Odd)
0	Command Reject
1	Intervention Required
2	Bus Out Parity
3	Equipment Check
4	Data Check
5	Data Late
6	Undefined
7	Undefined

## PHYSICAL UNIT TABLE FORMAT

1		2*				3**									
0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7
0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7
Device ID:	PU	Function				Specification for magnetic tape, or when COS, 2-4 is peripheral status				Hardware device address					
00 - Mag Tape	Number	0 - up; 1 - down				000 - Unallocated				110 - Allocated to main program					
01 - Reader						111 - Allocated to system									
02 - Serial Read/Punch															
03 - Printer															
04 - Row Read/Punch															
05 - 1001															
06 - Communications Device															
07 - 8410															
08 - 8411/8414															
09 - Paper Tape															
0A - 1004 Reader															
0B - 1004 Printer															
0C - 1004 Punch															
0D - ODR															

\* For communication devices, the device's input channel address.

\*\* For communication devices, the device's output channel address.