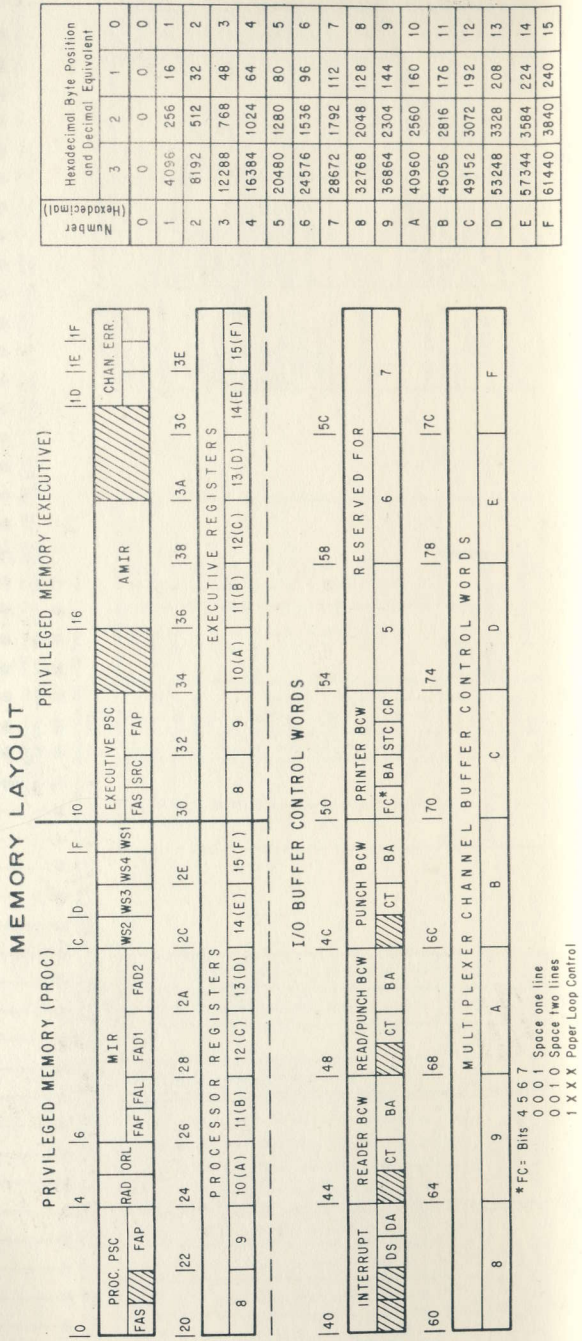
MEMORY LAYOUT

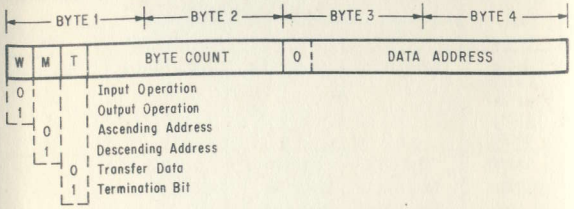


*FC= Bits 4 5 6 7
0 0 0 1 Space one line
0 0 1 0 Space two lines
1 X X X Paper Loop Control

Number (Hexadecimal)	Hexadecimal Byte Position and Decimal Equivalent 3	2	1	0
0	0	0	0	0
1	4096	256	16	1
2	8192	512	32	2
3	12288	768	48	3
4	16384	1024	64	4
5	20480	1280	80	5
6	24576	1536	96	6
7	28672	1792	112	7
8	32768	2048	128	8
9	36864	2304	144	9
A	40960	2560	160	10
B	45056	2816	176	11
C	49152	3072	192	12
D	53248	3328	208	13
E	57344	3584	224	14
F	61440	3840	240	15

BASIC MULTIPLEXER CHANNEL

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

COMMAND DETAILS

X= Variable to Control Units
P= Parity Bit (Odd)
*= Used with A4 Instruction Only

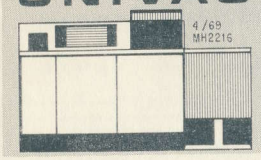
FIRST SENSE BYTE

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

BACKBOARD WIRING

Pin No.	Usage
1 and 55	Ground
4	-3V
16 and 34	+6V
46	-12V

UNIVAC 9200/9300 systems



MAINTENANCE CARD

INSTRUCTION REPERTOIRE

FORMAT	OP CODE	MNEMONIC	INSTRUCTION	OPERATION
BINARY				
	40	STH	Store Halfword	$(R_1) \rightarrow B2D2$
	48	LH	Load Halfword	$(B2D2) \rightarrow R_1$
RX	49	CH	Compare Halfword	(R_1) : (B2D2); set CC
SI	A6	AI	Add Immediate	I_2 (Sign Ext) + (B1D1) → B1D1(2 Bytes); set CC
	AA	AH	Add Halfword	$(R_1) + (B2D2) \rightarrow R_1$; set CC
RX	AB	SH	Subtract Halfword	$(R_1) - (B2D2) \rightarrow R_1$; set CC
LOGICAL				
	91	TM	Test Under Mask	I_2 (Mask) : B1D1; set CC
	92	MVI	Move Immediate	$I_2 \rightarrow$ B1D1
	94	NI	AND Immediate	$I_2 \odot$ (B1D1) → B1D1 ; set CC
SI	95	CLI	Compare Immediate	(B1 D1); I2; set CC
	96	OI	OR Immediate	$I_2 \oplus$ (B1D1) → B1D1; set CC
	A9	HPR	Halt & Proceed	Display B1D1
	D1	MVN	Move Numeric	(B2D2) Lower → B1D1 Lower (Zones Unchanged)
	D2	MVC	Move Character	(B2D2) → B1D1
	D4	NC	AND Character	(B2D2) ⊙ (B1D1) → B1D1; set CC
SS	D5	CLC	Compare Logical	(B1D1): (B2D2); set CC
	D6	OC	OR Character	(B2D2) ⊕ (B1D1) → B1D1; set CC
	DC	TR	Translate	[(B1 D1) + B2 D2] → B1 D1
	DE	ED	Edit	Unpack & Expand OP_2 (L_1+1) → OP_1 ; Control Pat in OP_1 MSB of OP_1 = Fill Char, DSB=20_{16}, SSB=21_{16}, FSB=22_{16}
DECIMAL				
	F1	MVO	Move with Offset	(B2D2) → B1D1 (Shift left four bits)*
	F2	PACK	Pack	(B2D2) Packed → B1D1*
	F3	UNPK	Unpack	(B2 D2) Unpk → B1D1 (Generate zones)*
	F8	ZAP	Zero and Add	(B2D2) → B1D1 Dec. ($L_1 \geq L_2$)*; set CC
	F9	CP	Compare Decimal	(B2D2): (B1D1) Dec.($L_1 \geq L_2$)*; set CC
SS	FA	AP	Add Decimal	(B1D1) + (B2D2) → B1D1 Dec. ($L_1 \geq L_2$)* ; set CC
	FB	SP	Subtract Decimal	(B1D1) − (B2D2) → B1D1 Dec. ($L_1 \geq L_2$)* ; set CC
	FC	MP	Multiply Decimal	OP_2 x OP_1 → OP_1 ; L_1 +1 = Size of product; L_2+1 = Size of Mult.; OP_1 must have L_2 + 1 leading dec. zero*
	FD	DP	Divide Decimal	$OP_1 \div OP_2 \rightarrow OP_1$; $L_1 > L_2$; Quotient in $L_1 - L_2$ MSB of OP_1
BRANCH				
RX	45	BAL	Branch and Link	(FAP) → R_1, B2D2 → FAP
	47	BC	Branch on Condition	If match, (B2D2) → FAP; see chart
PRIVILEGED				
SI	A0	SPSC	Store State	(PSC) → B1D1
	A8	LPSC	Load State	(B1D1) → PSC
SPECIAL				
SI	A1	SRC	Supervisor Call	$I_2 \rightarrow 11_{(16)}$; set interrupt
INPUT/OUTPUT				
SI	A4	XIOF	Execute I/O	Function → Device; set CC
	A5	TIO	Test I/O	Test Device Status → B1D1; set CC

*Data processed right to left ⊕ = OR − = Minus ⊙ = AND
: = Compare R = Prog. Reg. ( ) = Contents of CC = Condition Code + = Plus

PROGRAM STATE CONTROL STORAGE

	BYTE 1								BYTE 2								BYTE 3,4																
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	→													31
PROC. PSC	C	C	A	X	X	X	X	X	0	0	0	0	0	0	0	0	0	PROGRAM ADDRESS														
I/O PSC	C	C	A	X	X	X	X	X	SRC								0	PROGRAM ADDRESS														

CC = Condition Code
A = ASCII Control Code
A = 1 = ASCII
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 Restriction
1	0	ASCII Off							
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. Int.		Image	Read	
Read/Punch	2				↓	Stkr Sel.	↓	↓	Punch
Printer	3	48 Char. bar **	Numeric **		↓			*	*

* 30 and 31 ⟹ Space; 31 only ⟹ Print and Space
** If Bit 25 = 1, then Bit 24 must = 1

PRINT/MEMORY LOCATIONS

Col.	Decimal	0 1 2 3 4 5 6 7 8 9	1 0 1 2 3 4 5 6 7 8 9	2 0 1 2 3 4 5 6 7 8 9	3 0 1 2 3
Mem. Loc.	Hexadecimal	8 0 1 2 3 4 5 6 7 8 9 A B C D E F	9 0 1 2 3 4 5 6 7 8 9 A B C D E F		A 0

Col.	Decimal	3 4 5 6 7 8 9	4 0 1 2 3 4 5 6 7 8 9	5 0 1 2 3 4 5 6 7 8 9	6 0 1 2 3 4 5 6
Mem. Loc.	Hexadecimal	A 1 2 3 4 5 6 7 8 9 A B C D E F	B 0 1 2 3 4 5 6 7 8 9 A B C D E F		C 0 1

Col.	Decimal	6 7 8 9	7 0 1 2 3 4 5 6 7 8 9	8 0 1 2 3 4 5 6 7 8 9	9 0 1 2 3 4 5 6 7 8 9
Mem. Loc.	Hexadecimal	C 2 3 4 5 6 7 8 9 A B C D E F	D 0 1 2 3 4 5 6 7 8 9 A B C D E F		E 0 1 2

Col.	Decimal	10 0 1 2 3 4 5 6 7 8 9	11 0 1 2 3 4 5 6 7 8 9	12 0 1 2 3 4 5 6 7 8 9	13 0 1 2
Mem. Loc.	Hexadecimal	E 3 4 5 6 7 8 9 A B C D E F	F 0 1 2 3 4 5 6 7 8 9 A B C D E F		10 0 1 2 3

CONDITION CODE (CC) SETTINGS

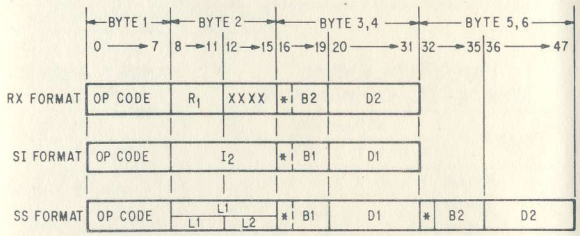
Instr.	Condition Codes/Conditions			
	0(00)	1(01)	2(10)	3(11)
SH(AB) AH(AA) AI(A6) AP(FA) SP(FB)	Result=Zero ↓	Result=Neg. ↓	Result Positive ↓	Overflow ↓
CH(49)	$(R_1)=(OP_2)$	$(R_1)<(OP_2)$	$(R_1)>(OP_2)$	
CLI(95)	$(OP_1)=I_2$	$(OP_1)<I_2$	$(OP_1)>I_2$	
CLC(D5)	$(OP_1)=(OP_2)$	$(OP_1)<(OP_2)$	$(OP_1)>(OP_2)$	
CP(F9)	$(OP_1)=(OP_2)$	$(OP_1)<(OP_2)$	$(OP_1)>(OP_2)$	
ZAP(F8)	$(OP_2)=Ø$	(OP_2) Neg.	(OP_2) Pos.	
NI(94) NC(D4) OI(96) OC(D6)	Result=Zero ↓	Result≠Zero ↓		
TM(91)	No match or mask=Ø	Partial match		Full match
XIOF(A4)	Accepted	Status Pending	Busy	Rejected
TIO(A5)	Available	Valid Status	Busy	Rejected

BRANCH/CONDITION

Condition Code	0	1	2	3
BC Instr. Bit	8	9	10	11

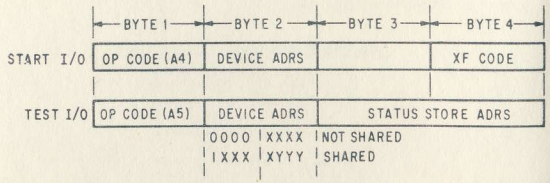
	SIGN +	SIGN −	ZONE
ASCII	1010	1011	0101
EBCDIC	1100	1101	1111

INSTRUCTION FORMAT



R1 = Adrs of program register
B1D1 = Adrs of operand one
B2D2 = Adrs of operand two
I = Immediate operand
L1 = One less than length of operand one
L2 = One less than length of operand two
* = Most sig. bit of B1 or B2 Field = 1 indicates indexing of OP_1 or OP_2

I/O INSTRUCTION FORMAT



XXXX = Subchannel Address
YYY = Device Number

I/O STATUS BYTE

I/O Device/Status Indication*				Status Byte Bit*
Printer (DA=3)	Card Punch or Read Punch (DA=2)	Card Reader (DA=1)		
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 Ø's → function performed as specified

COMPRESSED CARD CODE

Hexadecimal	Bit Positions 4 5 6 7 / 0 1 2 3	Punch Positions
0	0 0 0 0	
1	0 0 0 1	3
2	0 0 1 0	4
3	0 0 1 1	1
4	0 1 0 0	5
5	0 1 0 1	2
6	0 1 1 0	7
7	0 1 1 1	6
8	1 0 0 0	9
9	1 0 0 1	9,3
A	1 0 1 0	9,4
B	1 0 1 1	9,1
C	1 1 0 0	9,5
D	1 1 0 1	9,2
E	1 1 1 0	9,7
F	1 1 1 1	9,6
		12
		11
		12,11
		0
		12,0
		11,0
		12,11,0
		8
		8,12
		8,11
		8,12,11
		8,0
		8,12,0
		8,11,0
		8,12,11,0

PRINTABLE CHARACTERS

Zone Bits/Char. 63 Char. Bar			Numeric Bits 63 and 48 Char. Bar					Zone Bits/Char. 48 Char. Bar		
Bits 2 and 3			Bits					Bits 2 and 3		
00	01	10	11	4	5	6	7	Character	00	01	10	11	Character
				0	0	0	0		A	Q	0
				0	0	0	1		B	R	1
				0	0	1	0		C	S	2
				0	0	1	1		D	T	3
				0	1	0	0		E	U	4
				0	1	0	1		F	V	5
				0	1	1	0		G	W	6
				0	1	1	1		H	X	7
				1	0	0	0		I	Y	8
				1	0	0	1		J	Z	9
				1	0	1	0		K	+	.
				1	0	1	1		L	&	$
				1	1	0	0		M	%	*
				1	1	0	1		N	#	-
				1	1	1	0		O	@	/
				1	1	1	1		P	-	,
8	-	0											
A	J	/	1										
B	K	S	2										
C	L	T	3										
D	M	U	4										
E	N	V	5										
F	O	W	6										
G	P	X	7										
H	Q	Y	8										
I	R	Z	9										
¢	!	:	#										
.	$	*	@										
<	)	%	-										
(	]	_	>										
+	;	?	=										
	¬		"										