

**PRINTABLE CHARACTERS**

Zone Bits/Char. 63 Char. Bar		Zone Bits/Char. 48 Char. Bar					
Bits 2 and 3		Bits 2 and 3					
00	01	10	11	00	01	10	11
Character		Character		Character		Character	
0	0	0	0	A	Q	0	0
1	1	0	0	B	R	1	1
2	2	0	0	C	S	2	2
3	3	0	0	D	T	3	3
4	4	0	0	E	U	4	4
5	5	0	0	F	V	5	5
6	6	0	0	G	W	6	6
7	7	0	0	H	X	7	7
8	8	0	0	I	Y	8	8
9	9	1	0	J	Z	9	9
10	10	1	0	K	+	10	10
11	11	1	0	L	⋄	11	11
12	12	1	0	M	%	12	12
13	13	1	0	N	#	13	13
14	14	1	1	O	⊕	14	14
15	15	1	1	P	.	15	15

**COMPRESSED CARD CODE**

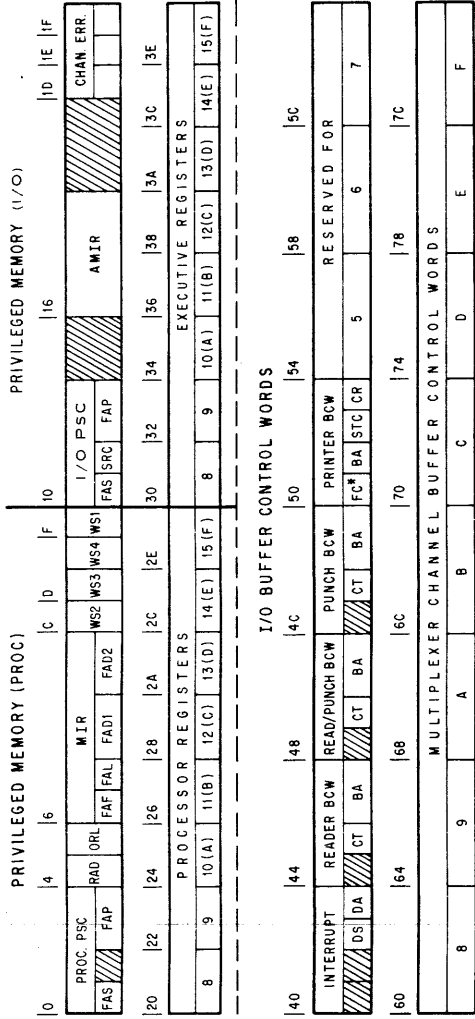
Hexadecimal	Bit Positions				Punch Positions			
	4	5	6	7	1	2	3	4
0	0	0	0	0	3	12	11	12
1	0	0	0	1	4	11	12	11
2	0	0	1	0	5	0	12	11
3	0	0	1	1	6	12	11	0
4	0	1	0	0	7	12	11	0
5	0	1	0	1	8	12	11	0
6	0	1	1	0	9	9,3	8,12	9,4
7	0	1	1	1	10	9,4	8,11	9,1
8	1	0	0	0	11	9,1	8,12,11	9,5
9	1	0	0	1	12	9,2	8,12,0	9,2
A	1	0	1	0	13	9,7	8,11,0	9,7
B	1	0	1	1	14	9,6	8,12,11,0	9,6
C	1	1	0	0				
D	1	1	0	1				
E	1	1	1	0				
F	1	1	1	1				

**I/O STATUS BYTE**

Printer (DA=3)		Card Punch or Read Punch (DA=2)		Card Reader (DA=1)		Status Byte Bit*
0	1	0	1	0	1	0
2	3	4	5	6	7	1
4	5	6	7	8	9	2
6	7	8	9	10	11	3
8	9	10	11	12	13	4
10	11	12	13	14	15	5
12	13	14	15	16	17	6
14	15	16	17	18	19	7

\*All 0's → function performed as specified

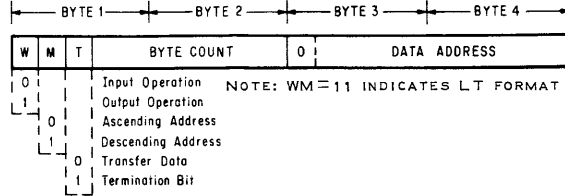
**MEMORY LAYOUT**



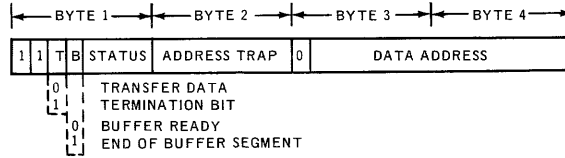
\*FC: Bits 4 5 6 7  
 0 0 0 1 Space one line  
 0 0 1 0 Space two lines  
 1 X X X Paper Loop Control

**MULTIPLEXER CHANNEL**

**BASIC BUFFER CONTROL WORD**



**LT BUFFER CONTROL WORD**



**STATUS BYTE**

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

**CHANNEL ERROR STATUS**

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

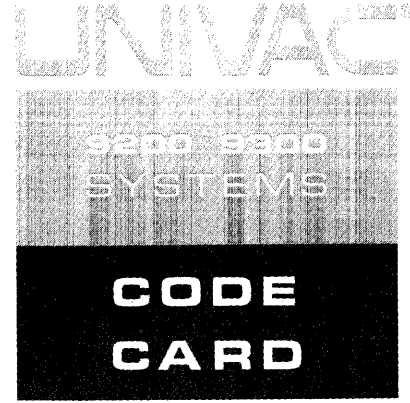
**MULTIPLEXER CHANNEL COMMANDS**

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	0	0 0 0
Write	P	0	0	0	1	0	0 1
Read	P	0	0	1	0	0	1 0
Control	P	0	1	1	0	0	1 1
Read Backward	P	1	1	0	0	0	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



# INSTRUCTION REPERTOIRE

FORMAT	OP CODE	MNEMONIC	INSTRUCTION	OPERATION
<b>BINARY</b>				
RX	40	STH	Store Halfword	(R <sub>1</sub> ) → B2D2
	48	LH	Load Halfword	(B2D2) → R <sub>1</sub>
	49	CH	Compare Halfword	(R <sub>1</sub> ): (B2D2); set CC
SI	A6	AI	Add Immediate	I <sub>2</sub> (Sign Ext) + (B1D1) → B1D1(2 Bytes); set CC
	AA	AH	Add Halfword	(R <sub>1</sub> ) + (B2D2) → R <sub>1</sub> ; set CC
RX	AB	SH	Subtract Halfword	(R <sub>1</sub> ) - (B2D2) → R <sub>1</sub> ; set CC
<b>LOGICAL</b>				
	91	TM	Test Under Mask	I <sub>2</sub> (Mask): B1D1; set CC
	92	MVI	Move Immediate	I <sub>2</sub> → B1D1
	94	NI	AND Immediate	I <sub>2</sub> ⊙ (B1D1) → B1D1; set CC
SI	95	CLI	Compare Immediate	(B1D1): I <sub>2</sub> ; set
	96	OI	OR Immediate	I <sub>2</sub> ⊕ (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	[(B1D1) + B2D2] → B1D1
	DE	ED	Edit	Unpack & Expand OP <sub>2</sub> (L <sub>1</sub> +1) → OP <sub>1</sub> ; Control Pat in OP <sub>1</sub> MSB of OP <sub>1</sub> = Fill Char, DSB=20 <sub>16</sub> , SSB=21 <sub>16</sub> , FSB=22 <sub>16</sub>
<b>DECIMAL</b>				
	F1	MVO	Move with Offset	(B2D2) → B1D1 (Shift left four bits)*
	F2	PACK	Pack	(B2D2) Packed → B1D1*
	F3	UNPK	Unpack	(B2D2) Unpk → B1D1 (Generate zones)*
SS	F8	ZAP	Zero and Add	(B2D2) → B1D1 Dec. (L <sub>1</sub> ≥ L <sub>2</sub> )*; set CC
	F9	CP	Compare Decimal	(B1D1): (B2D2) Dec. (L <sub>1</sub> ≥ L <sub>2</sub> )*; set CC
	FA	AP	Add Decimal	(B1D1) + (B2D2) → B1D1 Dec. (L <sub>1</sub> ≥ L <sub>2</sub> )*; set CC
	FB	SP	Subtract Decimal	(B1D1) - (B2D2) → B1D1 Dec. (L <sub>1</sub> ≥ L <sub>2</sub> )*; set CC
	FC	MP	Multiply Decimal	OP <sub>2</sub> × OP <sub>1</sub> → OP <sub>1</sub> ; L <sub>1</sub> +1: Size of product; L <sub>2</sub> +1: Size of Mult.; OP <sub>1</sub> must have L <sub>2</sub> +1 leading dec. zero*
	FD	DP	Divide Decimal	OP <sub>1</sub> ÷ OP <sub>2</sub> → OP <sub>1</sub> ; L <sub>1</sub> > L <sub>2</sub> ; Quotient in L <sub>1</sub> -L <sub>2</sub> MSB of OP <sub>1</sub>
<b>BRANCH</b>				
RX	45	BAL	Branch and Link	(FAP) → R <sub>1</sub> , B2D2 → FAP
	47	BC	Branch on Condition	If match, (B2D2) → FAP; see chart
<b>PRIVILEGED</b>				
SI	A0	SPSC	Store State	(PSC) → B1D1
	AB	LPSC	Load State	(B1D1) → PSC
<b>SPECIAL</b>				
SI	A1	SRC	Supervisor Call	I <sub>2</sub> → I <sub>1</sub> (16); set interrupt
<b>INPUT/OUTPUT</b>				
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

# 9300 EXTENDED OPERATION CODES

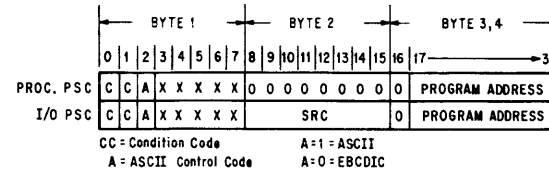
MNEMONIC	FUNCTION	HEXADECIMAL OPERATION CODE, R <sub>1</sub>	FORMAT
B	BRANCH	47 F	RX
NOP	NO OPERATION	47 0	RX
<i>USED AFTER COMPARISON INSTRUCTIONS</i>			
BH	BRANCH IF HIGH	47 2	RX
BL	BRANCH IF LOW	47 4	RX
BE	BRANCH IF EQUAL	47 8	RX
BNH	BRANCH IF NOT HIGH	47 D	RX
BNL	BRANCH IF NOT LOW	47 B	RX
BNE	BRANCH IF NOT EQUAL	47 7	RX
<i>USED AFTER TEST UNDER MASK INSTRUCTIONS</i>			
BO	BRANCH IF ALL ONES	47 1	RX
BZ	BRANCH IF ALL ZEROS	47 8	RX
BM	BRANCH IF MIXED	47 4	RX
BNO	BRANCH IF NOT ALL ONES	47 E	RX
BNZ	BRANCH IF NOT ALL ZEROS	47 7	RX
BNM	BRANCH IF NOT MIXED	47 B	RX
<i>USED AFTER ARITHMETIC INSTRUCTIONS</i>			
BO	BRANCH IF OVERFLOW	47 1	RX
BZ	BRANCH IF ZERO	47 8	RX
BM	BRANCH IF MINUS	47 4	RX
BP	BRANCH IF POSITIVE	47 2	RX
BNO	BRANCH IF NO OVERFLOW	47 E	RX
BNZ	BRANCH IF NOT ZERO	47 7	RX
BNM	BRANCH IF NOT MINUS	47 B	RX
BNP	BRANCH IF NOT POSITIVE	47 D	RX

# CONSTANT CHARACTERISTICS

CONSTANT TYPE	EXPLICIT LENGTH	IMPLICIT LENGTH	TRUNCATION OR PADDING	VALUE PADDED
C	variable 1 - 256	maximum 256	on right side	blanks
X	variable 1 - 256	maximum 256	on left side	hexadecimal 0
P*	variable 1 - 16	maximum 16	on left side	hexadecimal 0
Z*	variable 1 - 16	maximum 16	on left side	EBCDIC 0
H*	variable 1 - 2	2	on left side	hexadecimal 0
Y	variable 1 - 2	2	on left side	hexadecimal 0
S*	2	2	none	none

\*9300 only.

# PROGRAM STATE CONTROL STORAGE



# 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	0	0	0	0	0	0
0	1	1	Proc.	1	Proc.	1	0
1	0	0	I/O	0	I/O	0	1
1	1	1	ASCII On	1	ASCII Off	1	1

\*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 (24=1)	Numeric					*	*

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

# PRINT/MEMORY LOCATIONS

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

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

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

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

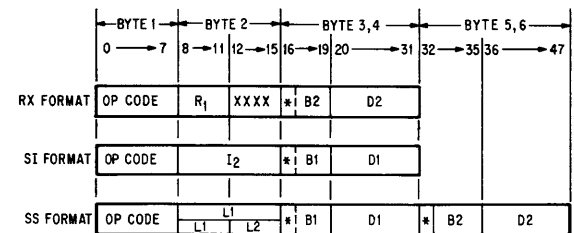
# CONDITION CODE (CC) SETTINGS

Instr.	Condition Codes/Conditions			
	0(00)	1(01)	2(10)	3(11)
SH (AB) AH (AA) AI (AG) 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>	(R <sub>1</sub> ) < (OP <sub>2</sub> ) (OP <sub>1</sub> ) < I <sub>2</sub>	(R <sub>1</sub> ) > (OP <sub>2</sub> ) (OP <sub>1</sub> ) > I <sub>2</sub>	
ZAP (FB)	(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

# BRANCH/CONDITION

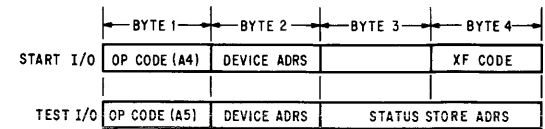
Condition Code	SIGN				ASCII	ZONE
	0	1	2	3		
BC Instr. Bit	8	9	10	11	1010	1011
					0101	
					1100	1101
					1111	

# INSTRUCTION FORMAT



R<sub>1</sub> = Adrs of program register L<sub>2</sub> = One less than length of operand two  
B<sub>1</sub>D<sub>1</sub> = Adrs of operand one \* = Most sig. bit of B<sub>1</sub> or B<sub>2</sub> Field =  
B<sub>2</sub>D<sub>2</sub> = Adrs of operand two 1 indicates indexing of OP<sub>1</sub> or OP<sub>2</sub>  
I = Immediate operand  
L<sub>1</sub> = One less than length of operand one

# I/O INSTRUCTION FORMAT



XXXX = Subchannel Address  
YYY = Device Number

# ASCII CODES

## DCT 2000 ASCII CHARACTER CODE

Bits				b <sub>7</sub>	0	0	0	0	1	1	1	1								
b <sub>6</sub>				0	0	1	1	0	0	1	1									
b <sub>5</sub>				0	1	0	1	0	1	0	1									
Col.				0	1	2	3	4	5	6	7									
b <sub>4</sub>	b <sub>3</sub>	b <sub>2</sub>	b <sub>1</sub>	Row																
0	0	0	0	0	NUL	*	DLE	*	SP	Blank	0	0	\	*	P	11-7	@	4-8	p	*
0	0	0	1	1	SOH		DC1		!	11-0	1	1	A	12-1	Q	11-8	a	*	q	*
0	0	1	0	2	STX		DC2	*	⌘	12-4-8	2	2	B	12-2	R	11-9	b	*	r	*
0	0	1	1	3	ETX		DC3	*	⌘	3-8	3	3	C	12-3	S	0-2	c	*	s	*
0	1	0	0	4	EOT		DC4	*	⌘	11-3-8	4	4	D	12-4	T	0-3	d	*	t	*
0	1	0	1	5	ENQ	*	NAK		%	0-4-8	5	5	E	12-5	U	0-4	e	*	u	*
0	1	1	0	6	ACK		SYN		⌘	12	6	6	F	12-6	V	0-5	f	*	v	*
0	1	1	1	7	BEL	12-11-0 7	ETB	*	'	7-8	7	7	G	12-7	W	0-6	g	*	w	*
1	0	0	0	8	BS	12-11-0 8	CAN	*	(	0-5-8	8	8	H	12-8	X	0-7	h	*	x	*
1	0	0	1	9	HT	12-11-0 9	EM	12-11-9	)	0-7-8	9	9	I	12-9	Y	0-8	i	*	y	*
1	0	1	0	10	LF	12-11-0 2-8	SS	*	*	11-4-8	:	5-8	J	11-1	Z	0-9	j	*	z	*
1	0	1	1	11	VT	12-11-0 3-8	ESC	*	+	2-8	;	11-6-8	K	11-2	[	12-5-8	k	*	{	*
1	1	0	0	12	FF	12-11-0 4-8	FS	*	'	⊕-3-8	<	12-6-8	L	11-3	\	0-6-8	l	*	~	*
1	1	0	1	13	CR	12-11-0 5-8	GS	*	-	11	=	12-7-8	M	11-4	]	11-5-8	m	*	}	*
1	1	1	0	14	SO	*	RS	*	.	12-3-8	>	6-8	N	11-5	Δ	11-7-8	n	*		*
1	1	1	1	15	SI	*	US	*	/	0-1	?	12-0	O	11-6	≠	0-2-8	o	*	DEL	

CHARACTER  
80-COL. CARD CODE

- Notes: 1. \* Indicates that the character to the left is not used in the DCT 2000.  
 2. Characters in columns 6 and 7 print as characters shown in columns 4 and 5 (except @ which prints @ as shown).  
 3. DEL and characters in columns 0 and 1 are non-printable.  
 4. SYN is deleted from the Receive message during TEXT.

### ASCII PROGRAM COUNT

PC	CNTR	RECEIVE	TRANSMIT
0	0000	Rec SYN/SOH/DC/BEL/EOT	Rec SYN/ACK/NAK/BEL
1	0010		SEL Xmit SYN
2	0011		STX SYN
3	0111		TEXT SYN
4	0110		ETX SYN
5	0100		BP SOH
6	0101	Xmit	SEL
7	0001		STX
8	1001		TEXT
9	1000		ETX
10	1010		BP
11	1110	ACK/NAK/BEL	DC/EOT
12	1111	ACK/NAK/BEL	DC/EOT
13	1011	One Bit Time (Not Used)	One Bit Time

Note: Information enclosed in the blocks involves block parity.

MH2221 8/67

MAINTENANCE  
CARD

UNIVAC  
DCT 2000

DATA  
COMMUNICATION  
TERMINAL

### ASCII AND DLT1/3 FORM CONTROL PAPER-LOOP

		4 2 1			NOTES
		4	2	1	
Start	Auto Tab	0	1	0	X = Either 1 or 0 1 = Hole 0 = No hole FF and VT override Auto and Overflow Tab. FF must be reset before VT can reset.
Stop	Auto Tab	X	1	0	
Start	Overflow Tab	1	0	0	
Stop	Overflow Tab	1	0	1	
Stop	Vertical Tab (VT)	1	1	0	
Stop	Form Feed (FF) and Home Paper	1	1	1	

### ASCII MESSAGE CONTROL SELECT CODE

Select Character	b <sub>7</sub>	b <sub>6</sub>	b <sub>5</sub>	b <sub>4</sub>	b <sub>3</sub>	b <sub>2</sub>	b <sub>1</sub>
Integral Printer	0	1	0/1*	0	0	0	0
Integral Punch	0	1	0/1*	0	0	0	1
Peripheral 1	0	1	0/1*	0	0	1	0
Peripheral 2	0	1	0/1*	0	0	1	1
Peripheral 3	0	1	0/1*	0	1	0	0
Peripheral 4	0	1	0/1*	0	1	0	1

\*Duplicate Block Protection

# DLT 1/3 CODES

# DLT 1/3 CODES

## DLT 1/3 CHARACTER CODE

80-Col. Card Code	Printable Characters	XS-3 Code	80-Col. Card Code	Printable Characters	XS-3 Code
12-1	A	01 0100	7	7	00 1010
12-2	B	01 0101	8	8	00 1011
12-3	C	01 0110	9	9	00 1100
12-4	D	01 0111	12	␣	01 0000
12-5	E	01 1000	11	-(Minus)	00 0010
12-6	F	01 1001	12-0	?	01 0011
12-7	G	01 1010	11-0	!(Exclam)	10 0011
12-8	H	01 1011	0-1	/	11 0100
12-9	I	01 1100	2-8	+	11 0011
11-1	J	10 0100	3-8	#	01 1101
11-2	K	10 0101	4-8	@	10 1110
11-3	L	10 0110	5-8	: (Colon)	01 0001
11-4	M	10 0111	6-8	>	11 1110
11-5	N	10 1000	7-8	*(Apos)	10 0000
11-6	O	10 1001	12-3-8	*(Period)	01 0010
11-7	P	10 1010	12-4-8	␣ ϕ	11 1101
11-8	Q	10 1011	12-5-8	␣	00 1111
11-9	R	10 1100	12-6-8	<	01 1110
0-2	S	11 0101	12-7-8	=	01 1111
0-3	T	11 0110	11-3-8	␣ ℒ	10 0010
0-4	U	11 0111	11-4-8	*	10 0001
0-5	V	11 1000	11-5-8	␣	00 0001
0-6	W	11 1001	11-6-8	;(Semi-col)	00 1110
0-7	X	11 1010	11-7-8	Δ	10 1111
0-8	Y	11 1011	0-2-8	≠	11 0000
0-9	Z	11 1100	0-3-8	,(Comma)	11 0010
0	0	00 0011	0-4-8	%	11 0001
1	1	00 0100	0-5-8	(	10 1101
2	2	00 0101	0-6-8	␣	00 1101
3	3	00 0110	0-7-8	)	11 1111
4	4	00 0111			
5	5	00 1000	Blank	Space N.P.	00 0000
6	6	00 1001			

Note: ϕ, ℒ, and ␣ are international code characters for DLT 1/3 and ASCII codes.

## DC VOLTAGE

Voltage	Terminal Board	Module Buss	Backboard Pin No.	Special Voltages
-12	A3TB1-4	W1	46	---
Gnd	A3TB1-11	W2	1 and 55	---
+6	A3TB1-10	W3	16 and 34	---
+12	A3TB1-6	---	22	Four Memory Chassis
+48	A3TB1-8	---	22	Many Pr and R/PU Chassis
+170	A4TB2-3 A4TB2-4	---	40	Printer Actuator Drivers

## INDICATOR TEST TERMINALS (TT)

Indicator	Location	TT
IND 1	C39	03
IND 2	C38	03

## DLT 1/3 PROGRAM COUNT

PC	CNTR	RECEIVE	TRANSMIT
0	0000	Rec	Rec
1	0001	SYN/SOM	SYN/SOM
2	0011	MC/DC/BEL/EOT	ACK/SPA/NAK/BEL
3	0010	TEXT/EOM	EOM
4	0110	BP	BP
5	0100	Xmit	Xmit
6	0101	SYN	SYN
7	0111	SYN	SYN
8	1111	SOM	SOM
9	1101	ACK/NAK	MC/DC/BEL/EOT
10	1100	(Not Used)	TEXT
11	1110	EOM	EOM
12	1010	BP	BP
13	1011	One Bit Time	One Bit Time

Notes: 1. Information enclosed in the blocks involves block parity.  
2. DC, BEL and EOT messages transmitted by DCT 2000 include TEXT.

## DLT 1/3 MESSAGE CONTROL CHARACTER CODES

Framing Characters	b7	b6	b5	b4	b3	b2	b1
SYNC (SYN)	0	1	1	0	1	0	1
Start of Message (SOM)	1	0	0	0	0	0	0
End of Message (EOM)	1	0	1	0	1	0	1

Message Control Characters (MC)							
Integral Printer	P	0/1*	N	N	0	0	0
Integral Punch	P	0/1*	N	N	0	0	1
Peripheral 3	P	0/1*	N	N	0	1	0
Peripheral 4	P	0/1*	N	N	0	1	1
Peripheral 5	P	0/1*	N	N	1	0	0
Peripheral 6	P	0/1*	N	N	1	0	1
Form Feed (FF)	P	0/1*	0	1	X	X	X
Vertical Tab (VT)	P	0/1*	1	0	X	X	X

Acknowledgment Characters							
Negative (NAK)	0	0	0	0	1	0	0
Positive (ACK)	1	0	0	0	1	0	1
Special Positive Acknowledge (SPA)	1	0	0	0	1	1	0

Special Characters							
Bell (BEL)	0	1	0	0	0	0	0
Device Control (DC)	0	1	0	0	0	1	1
End of Transmission (EOT)	1	1	0	0	0	0	1
Memory Fill Character	1	0	0	0	0	0	0

\*Duplicate Block Protection P - Odd Parity  
N-b5 and b4 cannot both be zero X - Either 1 or 0

## TRAP FF TEST TERMINALS (TT)

Circuit	ASCII Location and TT	DLT 1/3 Location and TT
*Pos. And Gate	B52 02, 03, 04, 05	B53 01, 02, 03, 04
Negative Set	B52 15	B53 12
*Pos. And Reset	B52 13, 14	B53 13, 14
Inverter	B52 01, 10	B53 07, 15

\*Ground input of Pos. And Gate if gate is not used.