PRINTABLE CHARACTERS

Zone Bits/Char. 63 Char. Bar, Bits 2 and 3				Numeric Bits 63 and 48 Char. Bar				Zone Bits/Char. 48 Char. Bar, Bits 2 and 3			
00	01	10	11	Bits				00	01	10	11
Character				4	5	6	7	Character			
A	B	-	0	0	0	0	0	A		Q	0
B	J	/	1	0	0	0	1	B		R	1
C	K	S	2	0	0	1	0	C		S	2
D	L	T	3	0	0	1	1	D		T	3
E	M	U	4	0	1	0	0	E		U	4
F	N	V	5	0	1	0	1	F		V	5
G	O	W	6	0	1	1	0	G		W	6
H	P	X	7	0	1	1	1	H		X	7
I	Q	Y	8	1	0	0	0	I		Y	8
¢	R	Z	9	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		-	,

COMPRESSED CARD CODE

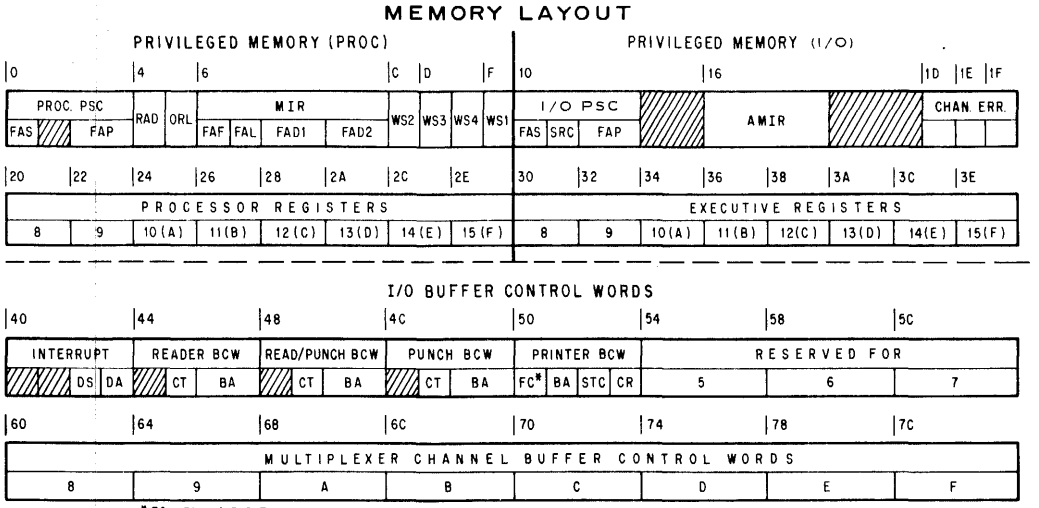
Hexadecimal	Bit Positions 4 5 6 7	Bit Positions 0 1 2 3	Punch Positions	
0	0 0 0 0			
1	0 0 0 1		3	12
2	0 0 1 0		4	11
3	0 0 1 1		1	12,11
4	0 1 0 0		5	0
5	0 1 0 1		2	12,0
6	0 1 1 0		7	11,0
7	0 1 1 1		6	12,11,0
8	1 0 0 0		9	8
9	1 0 0 1		9,3	8,12
A	1 0 1 0		9,4	8,11
B	1 0 1 1		9,1	8,12,11
C	1 1 0 0		9,5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

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 Ø's ⟶ function performed as specified

MEMORY LAYOUT



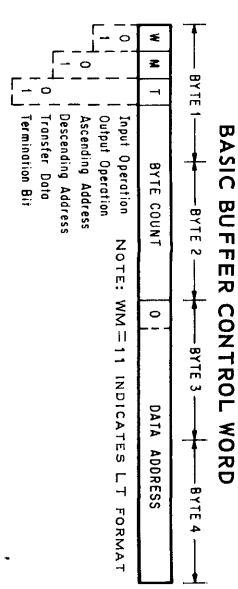
I/O BUFFER CONTROL WORDS

*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

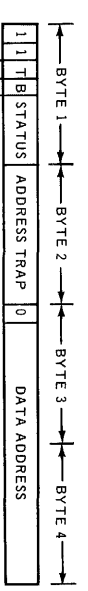
Number (Hexadecimal)	Hexadecimal Byte Position and Decimal Equivalent 3	2	1	0
0	0	0	0	0
1	4096	256	16	1
2	8192	512	32	2
3	12288	768	48	3
4	16384	1024	64	4
5	20480	1280	80	5
6	24576	1536	96	6
7	28672	1792	112	7
8	32768	2048	128	8
9	36864	2304	144	9
A	40960	2560	160	10
B	45056	2816	176	11
C	49152	3072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

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

MULTIPLEXER CHANNEL COMMANDS

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

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	COMMAND DETAILS						0	1
Read	P							1	0
Control	P							1	1
Read Backward	P							1	0
Reserved for Chan. Cont.	P	X	X	X	X	1	0	0	0

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
			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	(B1D1) : I_2 ; set
	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	[(B1D1) + B2D2] → B1D1
	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	(B2D2) Unpk → B1D1 (Generate zones)*
	F8	ZAP	Zero and Add	(B2D2) → B1D1 Dec. ($L1 \geq L2$)* ; set CC
SS	F9	CP	Compare Decimal	(B1D1) : (B2D2) Dec. ($L1 \geq L2$)* ; set CC
	FA	AP	Add Decimal	(B1D1) + (B2D2) → B1D1 Dec. ($L1 \geq L2$)* ; set CC
	FB	SP	Subtract Decimal	(B1D1) − (B2D2) → B1D1 Dec. ($L1 \geq L2$)* ; 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

9300 EXTENDED OPERATION CODES

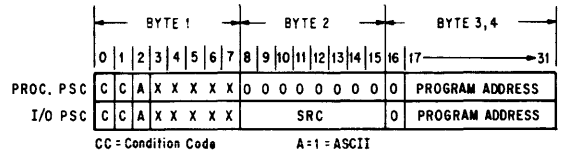
MNEMONIC	FUNCTION	HEXADECIMAL OPERATION CODE, R1	FORMAT
B	BRANCH	47 F	RX
NOP	NO OPERATION	47 0	RX
	USED AFTER COMPARISON INSTRUCTIONS		
BH	BRANCH IF HIGH	47 2	RX
BL	BRANCH IF LOW	47 4	RX
BE	BRANCH IF EQUAL	47 8	RX
BNH	BRANCH IF NOT HIGH	47 D	RX
BNL	BRANCH IF NOT LOW	47 B	RX
BNE	BRANCH IF NOT EQUAL	47 7	RX
	USED AFTER TEST UNDER MASK INSTRUCTIONS		
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
	USED AFTER ARITHMETIC INSTRUCTIONS		
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



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 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 (24=1)	Numeric					*	*

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

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

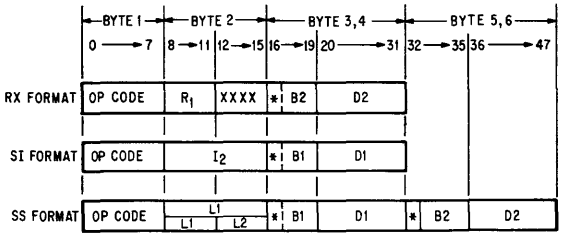
Instr.	Condition Codes/Conditions			
	0(00)	1(01)	2(10)	3(11)
SH (AB) AH (AA) AI (A6) AP (FA) SP (FB)	Result=Zero	Result=Neg.	Result Positive	Overflow
CH (49)	$(R_1) = (OP_2)$	$(R_1) < (OP_2)$	$(R_1) > (OP_2)$	
CLI (95)	$(OP_1) = I_2$	$(OP_1) < I_2$	$(OP_1) > I_2$	
CLC (D5)	$(OP_1) = (OP_2)$	$(OP_1) < (OP_2)$	$(OP_1) > (OP_2)$	
CP (F9)	$(OP_1) = (OP_2)$	$(OP_1) < (OP_2)$	$(OP_1) > (OP_2)$	
ZAP (F8)	$(OP_2) = \emptyset$	(OP_2) Neg.	(OP_2) Pos.	
NI (94) NC (D4) OI (96) OC (D6)	Result=Zero	Result≠Zero		
TM (91)	No match or mask = ∅	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

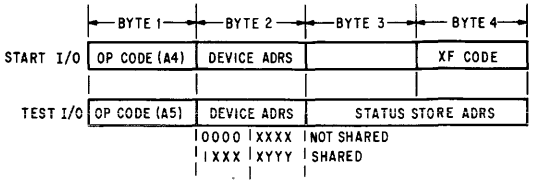
	SIGN +	SIGN −	ZONE
ASCII	1010	1011	0101
EBCDIC	1100	1101	1111

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

ASCII CODES

DCT 2000 ASCII CHARACTER CODE

Bits b_4	b_3	b_2	b_1	Row \ Col.	0		1		2		3		4		5		6		7	
				b_7	0		0		0		0		1		1		1		1	
				b_6	0		0		1		1		0		0		1		1	
				b_5	0		1		0		1		0		1		0		1	
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	*	,	0-3-8	<	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.

MH2221 8/67

MAINTENANCE CARD

UNIVAC DCT 2000

DATA COMMUNICATION TERMINAL

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 SYN	SEL
7	0001	SYN	STX
8	1001	SYN	TEXT
9	1000	SYN	ETX
10	1010	ACK/NAK/BEL	BP
11	1110	ACK/NAK/BEL	DC/EOT
12	1111	One Bit Time	DC/EOT
13	1011	(Not Used)	One Bit Time

Note: Information enclosed in the blocks involves block parity.

ASCII AND DLT1/3 FORM CONTROL PAPER-LOOP

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

ASCII MESSAGE CONTROL SELECT CODE

Select Character	b_7	b_6	b_5	b_4	b_3	b_2	b_1
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 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	8	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	$ £	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	△	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, £, 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 SYN/SOM	Rec SYN/SOM
1	0001	MC/DC/BEL/EOT	ACK/SPA/NAK/BEL
2	0011	TEXT /EOM	EOM
3	0010	BP	BP
4	0110	Xmit SYN	Xmit SYN
5	0100	SYN	SYN
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	b_7	b_6	b_5	b_4	b_3	b_2	b_1
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
Acknowledge 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
N-b_5 and b_4 cannot both be zero
P - Odd Parity
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.