DAVID E LO

CONTROL DATA

INSTANT

3100/3200/3300/3500

COMPASS

DAVID E. LEI
12/12/68

3100/3200/3300/3500 COMPASS

COMPASS is the upward compatible assembly system for the CONTROL DATA® 3100/3200/3300/3500 computers. COMPASS provides convenient mnemonics for the complete repertoire of machine instructions. Information may be referenced by word or by character.

In addition, COMPASS offers a variety of pseudo instructions to expedite programming. Pseudo instructions provide for:

Storage allocation

Storage reservation

Subprogram communication and linkage

Definition of various modes of constants

Variable field definitions

Macros

Conditional assembly

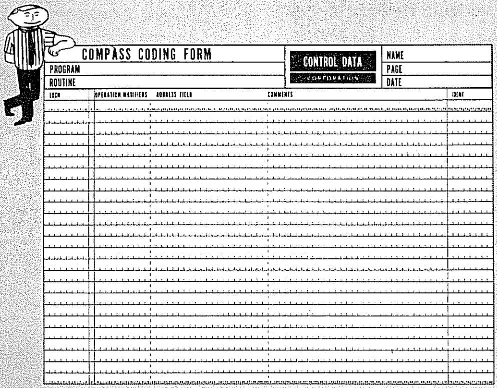
Output listing control

COMPASS source programs can be assembled with a variety of hardware configurations running under a Control Data operating system. The operating systems provide convenient input/output and data handling macros which programmers may reference within COMPASS programs.

COMPASS CODING FORM

Each line of a COMPASS coding form represents the four fields of a punched card.

Field	Position
Location field	Columns 1-8; column 9 blank
Operation field	Begins in column 10; terminates with first blank column
Address field	Subfields begin before column 41; terminate with first blank column or column 72
Identification field	Columns 73-80



Location Field

Program locations, data, and common area information.

Operation Field

Mnemonic machine instruction, pseudo instruction, macro name, $00\text{-}77_8$ in the first subfield, and operation modifiers as applicable.

Address Field

Relocatable or fixed	Subfield	Number of bits
m	First or second operand or jump address	15
n	Second operand address	15
r	First character address	17
s	Second character address	17
y	Operand	15
z	Operand	17
Fixed only		
k	Shift count	15
b	Index register	3
x	Connect code or interrupt mask	12
i	Increment or decrement	3
v	Address in register file	6
c	Character code or field	6
ch	Channel designator	3
l	Field length of block	7
Fixed only, 3300/3500 only		
B_m	Index register flag for M-field of BDP instructions	
B_r	Index register flag for R-field of BDP instructions	
B_s	Index register flag for S-field of BDP instructions	
l_m	Number of characters in M-field for BDP instructions	
l_r	Number of characters in R-field for BDP instructions	
l_s	Number of characters in S-field for BDP instructions	
sc	Scan character	
w	Page index file address	
cm	Channel mask	

Expression

An address field expression may be a symbol, a constant, *, or a combination of these, joined by the operators

+ addition

- subtraction

Constants

Decimal unless suffixed with B to signify octal.

* Current value of location counter; if the instruction occupies two words, the asterisk signifies the address of the first word.

** Associated subfield in assembled instruction is filled with ones.

Literals

The digit 2 may be inserted between the equal sign and D, O, H, or I to indicate double precision

=Dv	Decimal value v in DEC or DECD pseudo instruction format
=Ov	Octal value v in OCT pseudo instruction format
=Hv	Four-character Hollerith value v; eight character for = 2Hv
=Iv	Two-character BCD (ASCII) value v; four characters for = 2Iv (3300/3500 only)

Comments

Begin with first column after the first blank column in the address field and end with column 72

Identification Field

Printed with program listing

Address Modification

The contents of the address field plus the contents of a specified index register may be combined to form a modified address:

$$m + (B^b) = M$$
$$r + (B^b) = R$$
$$y + (B^b) = Y$$

COMPASS Assembly Error Flags

A	Format error in address field
C	Attempt to assemble information into common
D	Multiply defined symbol
F	Full symbol table
L	Location of field error
M	Operation modifier error
O	Operation error
U	Undefined symbol
T	Truncation error

REGISTERS

Mnemonic	Register	Bits
A	A register	24
B^b	Index register b	15
E	E register	48, 52
P	P register	15
Q	Q register	24

INSTRUCTION MODIFIERS

EQ	Equal
NE	Not equal
GE	Greater than or equal
LT	Less than
I	Indirect addressing
S	Sign extension
INT	Interrupt on completion
A	Conversion
B	Backward read or write
H	Half assembly or disassembly
N	No assembly or disassembly
C	Assign character address
NC	No conversion
dc	Delimiting character option in BDP instructions

3100/3200/3300/3500 MACHINE INSTRUCTIONS

Stops and Jumps

HLT[†]	m	Halt; next instruction from m
SJ1	m	Read next instruction at m if key 1 is set
SJ2	m	key 2
SJ3	m	key 3
SJ4	m	key 4
SJ5	m	key 5
SJ6	m	key 6
RTJ	m	(P) + 1 → m_{14-00} and read next instruction at m + 1
UJP, I	m,b	Unconditional jump to M
IJI	m,b	If $(B^b) \neq 0$, $(B^b) + 1 \rightarrow (B^b)$ and read next instruction at m; if $(B^b) = 0$, read next instruction at P + 1
IJD	m,b	If $(B^b) \neq 0$, $(B^b) - 1 \rightarrow (B^b)$ and read next instruction at m; if $(B^b) = 0$, read next instruction at P + 1
AZJ, EQ	m	Read next instruction at m if $(A) = 0$
NE	m	$(A) \neq 0$
GE	m	$(A) \geq 0$
LT	m	$(A) < 0$
AQJ, EQ	m	Read next instruction at m if $(A) = (Q)$
NE	m	$(A) \neq (Q)$
GE	m	$(A) \geq (Q)$
LT	m	$(A) < (Q)$

[†] When the 3300/3500 computer is operating in the program state of executive mode, an attempt to execute this instruction generates an executive interrupt and the processor reverts to the monitor state.

Register Operations, No Storage Referenc

ASE,S	y	Read next instruction at P + 2 if	$(A) = y$
QSE,S	y		$(Q) = y$
ISE	y,b		$(B^b) = y$
ISE	y		$y = 0$
ASG,S	y		$(A) \geq y$
QSG,S	y		$(Q) \geq y$
ISG	y,b		$(B^b) \geq y$
ISG	y		$y \geq 0$
ENA,S	y	Enter y into A	
ENQ,S	y	Enter y into Q	
ENI	y,b	Enter y into B^b	
ENI	y	No operation	
INA,S	y	$(A) + y \rightarrow A$	
INQ,S	y	$(Q) + y \rightarrow Q$	
INI	y,b	$(B^b) + y \rightarrow B^b$	
INI	y	No operation	
XOA,S	y	Exclusive OR	$(A) \forall y \rightarrow A$
XOQ,S	y		$(Q) \forall y \rightarrow A$
XOI	y,b		$(B^b) \forall y \rightarrow B^b$
XOI	y		No operation
ANA,S	y	AND	$(A) \wedge y \rightarrow A$
ANQ,S	y		$(Q) \wedge y \rightarrow Q$
ANI	y,b		$(B^b) \wedge y \rightarrow B^b$
ANI	y		No operation
ISI	y,b	If $(B^b) = y$, $0 \rightarrow B^b$ and read next instruction at P + 2, otherwise, $(B^b) + 1 \rightarrow B^b$ and read next instruction at P + 1.	
ISD	y,b	If $(B^b) = y$, $0 \rightarrow B^b$ and read next instruction at P + 2, otherwise, $(B^b) - 1 \rightarrow B^b$ and read next instruction at P + 1.	
ECHA,S	r	Enter 17-bit character address into A	
SHA	k,b	Shift A ± K positions to the right (-K) or end-around left (+K)	
SHQ	k,b	Shift Q ± K positions to the right (-K) or end-around left (+K)	
SHAQ	k,b	Shift AQ ± K positions to the right (-K) or end-around left (+K)	
SCAQ	y,b	Scale AQ; shift end-around left; if b = 1,2 or 3,y - shift count $\rightarrow B^b$	

Storage Test

MEQ	m,i	$(B^1) - i \rightarrow B^1$; if (B^1) negative, read next instructions at P + 1; if (B^1) positive, test $(A) = (Q) \wedge (M)$, if true, read next instruction at P + 2; if false, repeat sequence.
MTH	m,i	$(B^2) - i \rightarrow B^2$; if (B^2) negative, read next instruction at P + 1; if (B^2) positive, test $(A) \geq (Q) \wedge (M)$, if true, read next instruction at P + 2; if false, repeat sequence.
SSH	m	Test sign of (M), shift (M) end-around left one place. If sign is negative, read next instruction at P + 2; otherwise, read next instruction at P + 1.
CPR,I	m,b	$(M) > (A)$, read next instruction at P + 1 $(Q) > (M)$, read next instruction at P + 2 $(A) \geq (M) \geq (Q)$, read next instruction at P + 3.

Logical Instructions, Storage Reference

SSA,I	m,b	SET $(A_n) = 1$ where $(M_n) = 1$
SCA,I	m,b	Complement (A_n) where $(M_n) = 1$
LPA,I	m,b	$(A) \wedge (M) \rightarrow A$

Load

LDA,I	m,b	$(M) \rightarrow A$
LDQ,I	m,b	$(M) \rightarrow Q$
LACH	r,1	$0 \rightarrow A$, $(R) \rightarrow A_{05-00}$
LQCH	r,2	$0 \rightarrow Q$, $(R) \rightarrow Q_{05-00}$
LCA,I	m,b	$(\bar{M}) \rightarrow A$
LDAQ,I	m,b	$(M, M+1) \rightarrow AQ$
LCAQ,I	m,b	$(\bar{M}, \overline{M+1}) \rightarrow AQ$
LDL,I	m,b	$(Q) \wedge (M) \rightarrow A$
LDI,I	m,b	$(M_{14-00}) \rightarrow B^b$

Store

STA,I	m,b	$(A) \rightarrow M$
STQ,I	m,b	$(Q) \rightarrow M$
SACH	r,2	$(A_{05-00}) \rightarrow R$
SQCH	r,1	$(Q_{05-00}) \rightarrow R$

SWA,I	m,b	$(A_{14-00}) \rightarrow M_{14-00}$
STAQ,I	m,b	$(A) \rightarrow M$, $(Q) \rightarrow M + 1$
SCHA,I	m,b	$(A_{16-00}) \rightarrow M_{16-00}$
STI,I	m,b	$(B^b) \rightarrow M_{14-00}$

Inter-Register Transfer, 24-Bit Precisio

AQA		$(A) + (Q) \rightarrow A$
AIA	b	$(A) + (B^b) \rightarrow A$
IAI	b	$(B^b) + (A) \rightarrow B^b$
TIA	b	$(B^b) \rightarrow A$
TAI	b	$(A_{14-00}) \rightarrow B^b$; no operation if $b = 0$
TMQ†	v	(Register v) → Q
TQM†	v	(Q) → Register v
TMA	v	(Register v) → A
TAM†	v	(A) → Register v
TMI	v,b	(Register v_{14-00}) $\rightarrow B^b$
TIM†	v,b	$(B^b) \rightarrow$ Register v_{14-00}

Inter-Register Transfer, 48-Bit Precisio

ELQ		$(E_{lower}) \rightarrow Q$
QEL		$(Q) \rightarrow E_{lower}$
EUA		$(E_{upper}) \rightarrow A$
AEU		$(A) \rightarrow E_{upper}$
EAQ		$(E_u) \rightarrow A$, $(E_{lower}) \rightarrow Q$
AQE		$(A) \rightarrow E_{upper}$,$(Q) \rightarrow E_{lower}$

†When the 3300/3500 computer is operating in the program state of executive mode, an attempt to execute this instruction generates an executive interrupt and the processor reverts to the monitor state.

Fixed-Point Arithmetic, 24-Bit Precision

ADA,I	m,b	(A) + (M) → A
SBA,I	m,b	(A) - (M) → A
RAD,I	m,b	(A) + (M) → M
MUA,I	m,b	(A) * (M) → QA
DVA,I	m,b	(AQ)/(M) → A remainder in Q register

Fixed-Point Arithmetic, 48-Bit Precision

ADAQ,I	m,b	(AQ) + (M,M + 1) → AQ
SBAQ,I	m,b	(AQ) - (M,M + 1) → AQ
MUAQ,I	m,b	(AQ) * (M,M + 1) → AQE
DVAQ,I	m,b	(AQE)/(M,M + 1) → AQ remainder in E register

Floating-Point Arithmetic

FAD,I	m,b	(AQ) + (M,M + 1) → AQ
FSB,I	m,b	(AQ) - (M,M + 1) → AQ
FMU,I	m,b	(AQ) * (M,M + 1) → AQ
FDV,I	m,b	(AQ)/(M,M + 1) → AQ remainder in E register

Block Operations

SRCE,INT	c,r,s	Search from r to s for character = c
SRCN,INT	c,r,s	Search from r to s for character ≠ c
MOVE,INT[†]	l,r,s	Move characters from r to s, $1 \leq l \leq 177_8$ $l = 0$ means 200_8 characters
INAC,INT[†]	ch	A cleared, character from peripheral device → A_{05-00}
INAW,INT[†]	ch	A cleared, word from peripheral device → A
OTAC,INT[†]	ch	A_{05-00} → peripheral device
OTAW,INT[†]	ch	A_{word} → peripheral device

[†] When the 3300/3500 computer is operating in the program state of executive mode, an attempt to execute this instruction generates an executive interrupt and the processor reverts to the monitorstate.

INPC,INT,B,H[†]	ch,r,s	6 or 12-bit input characters r to s
INPW,INT,B,N[†]	ch,m,n	12 or 24-bit input words m to n
OUTC,INT,B,H[†]	ch,r,s	6 or 12-bit output characters r to s
OUTW,INT,B,N[†]	ch,m,n	12 or 24-bit output words m to n

Sensing, Selecting, Interrupt and Control Functions

CON[†]	x,ch	If channel not busy, connect code x is sent to I/O equipment on channel ch and read next instruction at P + 2; if channel busy, read next instruction at P + 1
SEL[†]	x,ch	If channel not busy, function code x is sent on channel ch, read next instruction at P + 2; if channel busy, read next instruction at P + 1
COPY[†]	ch	External status code → A_{11-00} (Interrupt Mask register) → A_{23-12}
EXS[†]	x,ch	Compare external status line bits with x_i, $0 \le i \le 11$; read next instruction at P + 1 if comparison, P + 2 if no comparison
INS[†]	x,ch	Compare internal status lines with bits x_i, $0 \le i \le 11$; read next instruction at P + 1 if comparison, P + 2 if no comparison
CINS[†]		Interrupt mask and internal status → A
INTS[†]	x,ch	Compare interrupt bits with bits x_i, $0 \le i \le 11$; read next instruction at P + 1 if comparison, P + 2 if no comparison
INCL[†]	x	Clears interrupt faults defined by bits x_i, $0 \le i \le 11$
IOCL[†]	x	Clear I/O channel, or search/move control defined by bits 00-08 and 11 of x
SSIM[†]	x	Set interrupt mask register with bits x_i, $0 \le i \le 11$
SCIM[†]	x	Clear interrupt mask register where bits of x are ones
IAPR[†]		Send interrupt to associated processor on left

[†] When the 3300/3500 computer is operating in the program state of executive mode, an attempt to execute this instruction generates an executive interrupt and the processor reverts to the monitor state.

PAUS[†]	x	Compare busy lines with bits x_i, $0 \le i \le 11$; if positive, do not advance. If advancement inhibited for more than 40 ms, read next instruction at P + 1; if no comparison, read next instruction at P + 2
SLS[†]		Stop if Selective Stop switch is set; read next instruction at P + 1 if restarted
SFPF		Set floating-point fault
SBCD		Set BCD fault
DINT		Disable interrupt
EINT[†]		Enable interrupt after next instruction
CTI[†]		Set Type in
CTO[†]		Set Type out
UCS[†]		Unconditional stop; restarts at P + 1
NOP		No operation

3300/3500 MACHINE INSTRUCTIONS, EXECUTIVE MODE

ACI		$(A_{02-00}) \rightarrow$ Channel Index register
ACR		$A \rightarrow$ Condition register
AIS		$(A_{02-00}) \rightarrow$ Instruction State register
AOS		$(A_{02-00}) \rightarrow$ Operand State register
APF	w,2	$(A_{11-00}) \rightarrow$ Page File Index Address W
CIA		Clear A; (Channel Index register) $\rightarrow A_{02-00}$
CILO	cm	Lockout external interrupt on masked channels, cm, until channels not busy
CLCA	cm	Clear channels, cm, but not external equipment; clear channels activity
CRA		Condition register $\rightarrow A$
ISA		Clear A; (Instruction State register) $\rightarrow A_{02-00}$
JAA		Last executed jump address $\rightarrow A_{14-00}$
LBR	m	Load m with BDP conditions

[†] When the 3300/3500 computer is operating in the program state of executive mode, an attempt to execute this instruction generates an executive interrupt and the processor reverts to the monitor state.

OSA		Clear A; (Operand State register) → A_{02-00}
PFA	w,2	Clear A; (Page Index File) → A_{11-00}
PRP	x	Same as PAUS, but halt real-time clock incrementing
RCR		(Subcondition register) → Condition register
RIS		Relocate to instruction state
ROS		Relocate to operand state
SBJP		Monitor state to Program state when next jump occurs
SBR	m	BDP conditions → m
SDL		When next LDA instruction encountered (M) → A 77777777 → M
SRA		Clear A; (Subcondition register) → A_{12-00}
TMAV		Initiate memory request. If reply occurs within 5 usec., read next instruction at P + 2; if not read next instruction at P + 1. Storage address in (B^b) with (operand state register) or zero appended

WITH BDP HARDWARE

ADM	r,B_r,I_r,s,B_r,I_s	(R) + (S) → S
ATD,dc	m,B_m,I_m,s,B_s	(M_{ASCII}) → S_{BCD}
CMP	r,B_r,I_r,s,B_s,I_s	Compare (R) to (S); exit upon encountering unequal characters
CMP,dc	r,B_r,s,B_s,I_s	Compare (R) to (S); exit upon encountering unequal characters
CVBD	m,B_m,n,B_n	(M_{binary}) → N_{BCD}
CVDB	r,B_r,I_r,m,B_m	(R_{BCD}) → M_{binary}
DTA,dc	r,B_r,I_r,m,B_m	(R_{BCD}) → M_{ASCII}
EDIT	r,B_r,I_r,s,B_s,I_s	(R) → S_{COBOL} picture editing
FRMT	r,B_r,I_r,s,B_s,I_s	(R) → S, comma insertions
HI		
JMP,LOW	m	Read next instruction at m if > 0
ZRO		(BDP condition register) < 0
		= 0

MVBF	r,B_r,I_r,s,B_s,I_s	(R) → S with blank fill
MVE	r,B_r,I_r,s,B_s,I_s	(R) → S
MVE,dc	r,B_r,s,B_s,I_s	(R) → S
MVZF	r,B_r,I_r,s,B_s,I_s	(R) → S with zero fill
MVZS	r,B_r,I_r,s,B_s,I_s	(R) → S with leading zeros suppressed
MVZS,dc	r,B_r,s,B_s,I_s	(R) → S with leading zeros suppressed
PAK	r,B_r,I_r,m,B_m	$(R_{BCD\ 6\text{-bit numeric}}) \to M_{BCD\ 4\text{-bit numeric}}$
SBM	r,B_r,I_r,s,B_s,I_s	(S) - (R) → S
SCAN,LR,$^{EQ}_{NE}$,dc	r,B_r,I_r,sc	Scan (R) from left to right for character equality/inequality
SCAN,RL,$^{EQ}_{NE}$,dc	r,B_r,I_r,sc	Scan (R) from right to left for character equality/inequality
TST	r,B_r,I_r	Test (R); -,0, or + → BCD condition register
UPAK	m,B_m,s,B_s,I_s	$(M_{BCD\ 4\text{-bit}}) \to S_{BCD\ 6\text{-bit}}$
ZADM	r,B_r,I_r,s,B_s,I_s	(R) → S right justified

3100/3200/3300/3500 PSEUDO INSTRUCTIONS

BCD	$n,c_1,c_2,\ldots,c_{4n}$	Define $c_1,c_2,\ldots,c_{4n}$ as BCD values stored four characters per word in n words. Symbol in location field is assigned the first word address
BCD,C	$n,c_1,c_2,\ldots,c_n$	Define the n BCD characters $c_1,c_2,\ldots,c_n$; symbol in the location field is assigned the first 17-bit character address
BSS	m	Reserve m words; symbol in location field is assigned the first word address
BSS,C	m	Reserve m character locations; symbol in location field is assigned the first character address
COMMON		Assign to common storage counter
DATA		Assign to data storage counter
DEC	$m_1,m_2,\ldots,m_n$	Define decimal integer values $m_1,m_2,\ldots,m_n$; symbol in location field is assigned the first word address

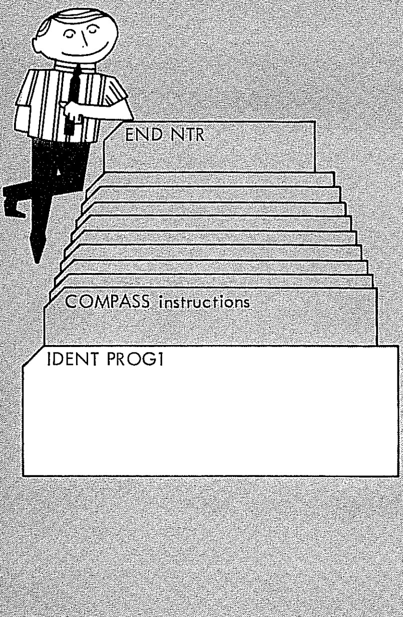
DECD	$d_1,d_2,\ldots,d_n$	Define double precision values $d_1,d_2,\ldots,d_n$ as floating point if a decimal is included, or as integers if a decimal point does not appear. Symbol in location field is assigned the first word address. Floating-point values may also include decimal and binary scaling factors
EJECT		Begin new page for listing
END	m	Last statement of subprogram; m is the transfer address or blank
ENDM		Terminate the macro definition
ENTRY	$m_1,m_2,\ldots,m_n$	Define $m_1,m_2,\ldots,m_n$, for reference as addresses by other subprograms
EQU	m	Equate symbol in location field to the 15-bit contents of the address field m
EQU,C	r	Equate symbol in location field to the 17-bit contents of the address field r
EXT	$m_1,m_2,\ldots,m_n$	Define $m_1,m_2,\ldots,m_n$ for reference as addresses in other subprograms
FINIS		Terminate assembly process
IDENT	m	First statement of subprogram m
IFF	m,p,n	Assemble following n lines in a macro definition if symbol p is not identical to symbol m
IFN	m,n	Assemble following n lines if the value of the expression m is non-zero
IFT	m,p,n	Assemble following n lines in a macro definition if symbol p is identical to symbol m
IFZ	m,n	Assemble following n lines if the value of the expression m is zero
LIBM	$\text{name}_1,\text{name}_2,\ldots$	Library macros called in program
LIST		Resume listing of source program

MACRO	$(p_1,p_2,\ldots,p_n)$	Assemble as the first instruction of a macro definition with the formal parameter list $(p_1,p_2,\ldots,p_n)$. The symbol in the location field is the name of the macro
macro name	$(p_1,p_2,\ldots,p_n)$	Call macro name with actual parameters $(p_1,p_2,\ldots,p_n)$; a symbol in the location field is assigned the first assembled instruction of the macro
NOLIST		Suppress listing of source program
OCT	$m_1,m_2,\ldots m_n$	Define octal values $m_1,m_2,\ldots,m_n$; symbol in location field is assigned the first word address
ORGR	m	Set the relocatable address counter with the value of the expression m in the current subprogram, data or common area
PRG		Assign to subprogram location counter
REM		Print remark appearing in columns 1-8, 14-72. A statement with an asterisk in column one will also be printed as a remark
SPACE	m	Space source program listing m lines
TITLE	name	Print name 53 characters beginning in column 20 at top of each page of listing
VFD	mn/v,...,mn/v	Define continuous fields for specified variables n Number of bits v Variable string m Mode O Octal H Hollerith A Word address arithmetic C Character address arithmetic I ASCII (3300/3500 only)

3300/3500 PSEUDO INSTRUCTIONS

ASCII,p	$n,c_1,c_2,\ldots,c_n$	Define ASCII characters n words packed p per word, 2 per word if p is omitted
BCDN	n,sdd...ddd	BCD numeric characters are converted to 4-bit characters and stored right-to-left in n consecutive words s = sign (stored in rightmost character) d = BCD numeric character

SAMPLE COMPASS DECK



Collating Sequence	Internal Code	Tape BCD Code	Printer Character	Cards	
				Character	Card
00	60	20	Δ	Δ	blank
01	15	15	≤		8,5
02	16	16	%		8,6
03	17	17	[		8,7
04	75	35	→		0,8,5
05	76	36	≡		0,8,6
06	77	37	∧		0,8,7
07	55	55	↑		11,8,5
08	56	56	↓		11,8,6
09	57	57	>		11,8,7
10	35	75	≥		12,8,5
11	36	76	┐		12,8,6
12	33	73	•	•	12,8,3
13	34	74	)	)	12,8,4
14	37	77	;		12,8,7
15	20	60	+	+	12
16	53	53	$	$	11,8,3
17	54	54	*	*	11,8,4
18	40	40	–		11
19	61	21	/	/	0,1
20	73	33	,	,	0,8,3
21	74	34	(	(	0,8,4
22	13	13	=	=	8,3
23	14	14	≠	–	8,4
24	32	72	<	+0	12,0
25	21	61	A	A	12,1
26	22	62	B	B	12,2
27	23	63	C	C	12,3
28	24	64	D	D	12,4
29	25	65	E	E	12,5
30	26	66	F	F	12,6
31	27	67	G	G	12,7
32	30	70	H	H	12,8
33	31	71	I	I	12,9
34	52	52	∀	–0	11,0
35	41	41	J	J	11,1

Collating Sequence	Internal Code	Tape BCD Code	Printer Character	Cards Character	Card
36	42	42	K	K	11,2
37	43	43	L	L	11,3
38	44	44	M	M	11,4
39	45	45	N	N	11,5
40	46	46	O	O	11,6
41	47	47	P	P	11,7
42	50	50	Q	Q	11,8
43	51	51	R	R	11,9
44	72	32	]	record-mark	0,8,2
45	62	22	S	S	0,2
46	63	23	T	T	0,3
47	64	24	U	U	0,4
48	65	25	V	V	0,5
49	66	26	W	W	0,6
50	67	27	X	X	0,7
51	70	30	Y	Y	0,8
52	71	31	Z	Z	0,9
53	00	12	0	0	0
54	01	01	1	1	1
55	02	02	2	2	2
56	03	03	3	3	3
57	04	04	4	4	4
58	05	05	5	5	5
59	06	06	6	6	6
60	07	07	7	7	7
61	10	10	8	8	8
62	11	11	9	9	9

Note: Within the collating sequence, tape codes of 00 and 12 are the same.

CONTROL DATA
CORPORATION

Documentation Department
3145 Porter Drive
Palo Alto, California 94304

60176700A ©CONTROL DATA CORPORATION Printed in U.S.A.
MARCH 1968