

CONTROL STATEMENT

COMPASS (p ₁ , p ₂ , ..., p _n)		or	COMPASS.	
A omitted	Do not abort	ML omitted or ML	MODLEVEL returns JDATE	
A	Abort on assembly errors	ML=string	MODLEVEL returns 9-character string	
B omitted or B	Binary on LGO	N omitted	Normal ejects	
B=0	No binary	N	No ejects	
B=lfm	Binary on lfm	O omitted or O	Short list on OUTPUT	
D omitted	No debug mode	O=lfm	Short list on lfm	
D	Debug mode	O=0	No short list	
F	*F returns 0	PC omitted or PC	PCOMMENT is 30 blanks	
F=n	*F returns n (decimal)	PC=string	PCOMMENT is 30-character string	
F=name	*F returns as follows:	P omitted	New pagination on END	
	COMPASS 0	P	Continue pagination	
	RUN 1			
	FTN 2			
G† omitted or G	No system text	S† omitted	SYSTEXT overlay	
G=lfm	Overlay on lfm	S=0	No system text	
G=lfm/ovl	Named overlay on lfm	S=ovl	Named overlay on library	
I omitted	Source on INPUT	S=lib/ovl	Named overlay on named library	
I	Source on COMPILE	X omitted	XTEXT on OLDPL	
I=lfm	Source on lfm	X=lfm	XTEXT on lfm	
L omitted or L	Full list on OUTPUT	X=0	XTEXT on OPL	
L=lfm	List on lfm			
L=0	No full list			
LO omitted or	Selects B,L,N, and R			
LO=0				
LO	Selects C,F,G,X; deselects R			
LO=c ₁ c ₂ ...c _n	Deselects if c _i is B,L,N, or R; selects if c _i is other.			

MODEL 72, 73, 74 EXIT MODES

Bit 50	Bit 49	Bit 48
INDEFINITE OPERAND	OPERAND OUT OF RANGE	ADDRESS OUT OF RANGE

MODEL 76 PSD REGISTER

MODE					CONDITION				
Exit	Mon	Step	Ind	Ovf	Undf	LPar	SPar	LBik	SBik
						LDir	SDir	Prog	Bkp
						Step	Ind	Ovf	Und

FATAL ERRORS

A	ADDRESS FIELD BAD
B	DOUBLY DEFINED SYMBOL. THE FIRST DEFINITION HOLDS.
E	ECHO, DUP, RMT, OR MACRO ILLEGALLY NESTED.
F	NUMBER OF ENTRIES EXCEEDS PERMISSIBLE AMOUNT.
L	LOCATION FIELD BAD.
N	NEGATIVE RELOCATION ON ENTRY POINT.
O	OPERATION FIELD BAD.
P	CONSULT LISTING FOR REASON BEHIND P ERROR.
R	DATA ORIGIN OUTSIDE BLOCK OR IN BLANK COMMON BLOCK.
U	UNDEFINED SYMBOL. VALUE ASSUMED 0.
V	BIT COUNT ERROR ON VFD (MUST BE 0 ≤ COUNT ≤ 60).

INFORMATIVE ERRORS

1	LOCATION SYMBOL BAD. SYMBOL NOT DEFINED.
2	ADDRESS ERROR ON SYMBOL DEFINITION.
3	DUPLICATE MACRO DEFINITION. NEW ONE OVERRIDES.
4	BAD FORMAL PARAMETER NAME IGNORED.
5	CPU OPERATION SYNTAX INCORRECTLY SPECIFIED.
6	LOCATION FIELD MEANINGLESS.
7	ADDRESS VALUE EXCEEDS FIELD SIZE, RESULT TRUNCATED.
8	MISSING OR EXTRA ADDRESS SUBFIELD.
9	MICRO SUBSTITUTION ERROR. NO SUBSTITUTION.

CHARACTER SETS

Char.	Display	Hollerith	BCD		ASCII SUBSET		
			Ext.	Int.	Char.	Code	Punch
:		8-2	00	12	:	3A	8-2
A	01	12-1	61	21	A	41	12-1
B	02	12-2	62	22	B	42	12-2
C	03	12-3	63	23	C	43	12-3
D	04	12-4	64	24	D	44	12-4
E	05	12-5	65	25	E	45	12-5
F	06	12-6	66	26	F	46	12-6
G	07	12-7	67	27	G	47	12-7
H	10	12-8	70	30	H	48	12-8
I	11	12-9	71	31	I	49	12-9
J	12	11-1	41	41	J	4A	11-1
K	13	11-2	42	42	K	4B	11-2
L	14	11-3	43	43	L	4C	11-3
M	15	11-4	44	44	M	4D	11-4
N	16	11-5	45	45	N	4E	11-5
O	17	11-6	46	46	O	4F	11-6
P	20	11-7	47	47	P	50	11-7
Q	21	11-8	50	50	Q	51	11-8
R	22	11-9	51	51	R	52	11-9
S	23	0-2	22	62	S	53	0-2
T	24	0-3	23	63	T	54	0-3
U	25	0-4	24	64	U	55	0-4
V	26	0-5	25	65	V	56	0-5
W	27	0-6	26	66	W	57	0-6
X	30	0-7	27	67	X	58	0-7
Y	31	0-8	30	70	Y	59	0-8
Z	32	0-9	31	71	Z	5A	0-9
0	33	0	12	00	0	30	0
1	34	1	01	01	1	31	1
2	35	2	02	02	2	32	2
3	36	3	03	03	3	33	3
4	37	4	04	04	4	34	4
5	40	5	05	05	5	35	5
6	41	6	06	06	6	36	6
7	42	7	07	07	7	37	7
8	43	8	10	10	8	38	8
9	44	9	11	11	9	39	9
+	45	12	60	20	+	2B	12-8-6
-	46	11	40	40	-	2D	11
*	47	11-8-4	54	54	*	2A	11-8-4
/	50	0-1	21	61	/	2F	0-1
(	51	0-8-4	34	74	(	28	12-8-5
)	52	12-8-4	74	34	)	29	11-8-5
\$	53	11-8-3	53	53	\$	24	11-8-3
=	54	8-3	13	13	=	30	8-6
space	55	space	20	60	space	20	space
.	56	0-8-3	33	73	.	2C	0-8-3
,	57	12-8-3	73	33	,	2E	12-8-3
!	60	0-8-6	36	76	!	23	8-3
	61	8-7	17	17		5B	12-8-2
%	62	0-8-2	32	72	%	5D	11-8-2
^	63	8-6	16	16	^	25	0-8-4
&	64	8-4	14	14	&	22	8-7
∇	65	0-8-5	35	75	∇	5F	0-8-5
∧	66	11-0①	52	52	∧	21	12-8-7 ①
∨	67	0-8-7	37	77	∨	26	12
↑	70	11-8-5	55	55	↑	27	8-5
↓	71	11-8-6	56	56	↓	3F	0-8-7
↖	72	12-0②	72	32	↖	3C	12-8-4 ②
↗	73	11-8-7	57	57	↗	3E	0-8-6
↘	74	8-5	15	15	↘	40	8-4
↙	75	12-8-5	75	35	↙	5C	0-8-2
↚	76	12-8-6	76	36	↚	5E	11-8-7
↛	77	12-8-7	77	37	↛	5B	11-8-6

- ① 11-0 and 11-8-2 are equivalent
② 12-0 and 12-8-2 are equivalent
③ 11-0 and 12-8-7 are equivalent
④ 12-0 and 12-8-4 are equivalent

LANGUAGE ELEMENTS

SPECIAL CHARACTERS

Any Field

≠ Micro substitution
Concatenation

First Column

* Comments line
* Continuation

Location Field

+ Force Upper
- Negate Force Upper

Variable Field

=item Literal
=Sym Deferred Symbol

=Xsym External Symbol
=O Origin Counter

*or *L Location Counter
*P Position Counter

\$ Position Counter - 1
\$F Caller: COMPASS

RUN 0
FTN 2

/name/ Subfield Delimiter
Symbol Qualifier

CHARACTER NOTATION

Data Item sign n t string

or
sign t d string d

Constant n t string

Literal = sign n t string

or
= sign t d string d

t = C Left, 12 zero bits
H Left, blank fill

A Right, blank fill
R Right, zero fill

L Left, zero fill
Z Left, 6 zero bits

d = delimiter character
n = no. of characters

string = Code characters

NUMERIC NOTATION

Data Item sign prefx value mods

Constant value mods

Literal = sign prefx value mods

Mods Spec Default

Sign + or - +

Radix O or B BASE

(pre/post) or D Pseudo

Integer n 0

Fraction n.or none

Pwr 10 sgl E or En

Pwr 10 dbl EE or EEn

Pwr 2 S or Sn

Binary pt S or Ph

DATE QUAL
JDATE SEQUENCE
TIME MODLEVEL
BASE PCOMMENT
CODE

PSEUDO INSTRUCTIONS

PROGRAM DEFINITION

IDENT name, fwa, eptsym
sym END trasyim

BINARY CONTROL

ABS MACHINE type, hf₁, hf₂, ..., hf_n
type = 6 or 7
hf_i = C, D, I, L, X

PPU J
PERIPH J

IDENT name, org, enty, ℓ₁, ℓ₂
IDENT name, org, enty, ppu

name SEGMENT fwa, eptsym
SET

record STEXT
LOC directive
COMMENT string
NOLABEL I

MODE CONTROL

mname BASE O or D or M or *
CODE A or D or E or I or *
QUAL name or *

B7=1
B7=1
COL n

USE *or name or // or /name/
USELCM *or name or // or /name/
ORG exp
ORGC exp

sym BSS exp
LOC exp
POS aexp

SYMBOL DEFINITION

sym = exp
sym EQU exp
sym SET exp

sym MAX exp₁, exp₂, ..., exp_n
sym MIN exp₁, exp₂, ..., exp_n
sym MIOCNT mname
sym SST sym...

COUNTER CONTROL

USE *or name or // or /name/
USELCM *or name or // or /name/
ORG exp
ORGC exp

sym BSS exp
LOC exp
POS aexp

SYMBOL DEFINITION

sym = exp
sym EQU exp
sym SET exp

sym MAX exp₁, exp₂, ..., exp_n
sym MIN exp₁, exp₂, ..., exp_n
sym MIOCNT mname
sym SST sym...

NUMERIC NOTATION

Data Item sign prefx value mods

Constant value mods

Literal = sign prefx value mods

Mods Spec Default

Sign + or - +

Radix O or B BASE

(pre/post) or D Pseudo

Integer n 0

Fraction n.or none

Pwr 10 sgl E or En

Pwr 10 dbl EE or EEn

Pwr 2 S or Sn

Binary pt S or Ph

DATE QUAL
JDATE SEQUENCE
TIME MODLEVEL
BASE PCOMMENT
CODE

LINKAGE CONTROL

ENTRY sym₁, sym₂, ..., sym_n
ENTRYC sym₁, sym₂, ..., sym_n
EXT sym₁, sym₂, ..., sym_n

DATA GENERATION

sym BSS exp
sym DATA item₁, ..., item_n
sym DIS string
sym DIS dstring
sym LIT item₁, ..., item_n
sym VFD item₁, exp₁, ..., exp_n
sym CON item₁, exp₁, ..., exp_n
sym R reg exp
sym REP Saddr, Daddr, C/rep, B/bsz, l/inc
sym REPC Saddr, Daddr, C/rep, B/bsz, l/inc
sym REPI Saddr, Daddr, C/rep, B/bsz, l/inc

CONDITIONAL ASSEMBLY

name IFop exp₁, exp₂, ¬ op att
name IFM1 exp, ¬ EQ ABS
name IFPL exp, ¬ GE COM
name IFCP ¬ LT DEF
name IFCP6 ¬ GT EXT
name IFCP7 ¬ LT LCM
name IFPP ¬ LT LOC
name IFPP6 ¬ LT MIC
name IFPP7 ¬ LT REG
name IF att, exp, ¬ REL
name IF -att, exp, ¬ SET
name IF -op, dstring, dstring₁ d, ¬ SST
name IF -op, dstring, dstring₁ d, ¬ SST

name ELSE ¬
name SKIP ¬

LIST CONTROL

LIST p₁, p₂, ..., p_n or *

p = A Assembly
B Binary control
C Control statements
D Detail
E Echoed lines
F IF - Skipped lines
G Code generation
H Reference table only
L User macros
M Referenced symbols only
N No references
R System macros
S SST symbols
T XTEXT lines

name EJECT
name SPACE aexp₁, aexp₂
name TITLE string
name NOREF sym₁, ..., sym_n
name TTL string
name CTEXT string
name ENDX
name XREF A or B

DEFINITION OPERATIONS

file XTEXT record
name DUP rep, ¬
name ECHO ¬, p₁ = {list₁}, p₂ = {list₂}, ...
name ENDD
name STOPDUP

name RMT
name HERE
name MACRO p₁, p₂, ..., p_n
name MACRO name, p₁, p₂, ..., p_n
name MACRO p₁, p₂, ..., p_n
name MACROE name, p₁, p₂, ..., p_n
name OPDEF p₁, p₂, ..., p_n
name ENDM
name LOCAL sym₁, ..., sym_n
name IRP p

OP CODE MANAGEMENT

CPU INSTRUCTIONS				① Model 76 only ② Models 72, 73, and 74 only ③ Privileged to Monitor			
Instruction	Operation	Variable	Description				
00000	ES ① ③		Error exit to EEA				
0000K	PS ②	K	Program stop				
0100K	RJ	K	Return jump to K				
011JK	RL ①	Bj+K	Block-copy K plus (Bj) words from LCM to SCM				
011JK	RE ②	Bj+K	Read extended core storage				
012JK	WL ①	Bj+K	Block-copy K plus (Bj) words from SCM to LCM				
012JK	WE ②	Bj+K	Write extended core storage				
01300	MJ ①		Exchange-exit to NEA if exit flag clear				
013JK	MJ ① ③	Bj+K	Exchange-exit to K + (Bj) if exit flag set				
013JK	MJ ② ③	Bj+K	Central exchange jump to Bj+K				
014JK	RXj ①	Xk	Read LCM at (Xk) to Xj				
015JK	WXj ①	Xk	Write (Xj) into LCM at (Xk)				
0160K	RI ① ③	Bk	Reset channel (Bk) input buffer				
016JK	IBj ①	Bk	Read channel (Bk) input status to Bj if j ≠ 0; otherwise, same as RI				
016JK	OBj ① ③	Bk	Read channel (Bk) output status to Bj if j ≠ 0; otherwise, same as RO				
020JK	JP	Bj+K	Jump to K plus (Bj)				
030JK	ZR	Xj,K	Branch to K if (Xj) = 0				
031JK	NZ	Xj,K	Branch to K if (Xj) ≠ 0				
032JK	PL	Xj,K	Branch to K if (Xj) sign is plus				
033JK	MI	Xj,K	Branch to K if (Xj) sign is minus				
034JK	NG	Xj,K	Branch to K if (Xj) sign is minus				
034JK	IR	Xj,K	Branch to K if (Xj) in range				
035JK	OR	Xj,K	Branch to K if (Xj) not in range				
036JK	DF	Xj,K	Branch to K if (Xj) definite				
037JK	ID	Xj,K	Branch to K if (Xj) indefinite				
0400K	EQ	K	Branch to K				
0400K	ZR	K	Branch to K				
041JK	EQ	Bj, Bj, K	Branch to K if (Bi) = (Bj)				
041JK	ZR	Bj, K	Branch to K if (Bi) = 0				
051JK	NE	Bj, Bj, K	Branch to K if (Bi) ≠ (Bj)				
051JK	GE	Bj, K	Branch to K if (Bi) ≥ (Bj)				
061JK	NZ	Bj, K	Branch to K if (Bi) > (Bj)				
061JK	GE	Bj, K	Branch to K if (Bi) ≥ (Bj)				
061JK	LE	Bj, Bj, K	Branch to K if (Bi) ≤ (Bj)				
061JK	LE	Bj, K	Branch to K if (Bi) < (Bj)				
061JK	LT	Bj, K	Branch to K if (Bi) > (Bj)				
071JK	LT	Bj, K	Branch to K if (Bi) < (Bj)				
071JK	NG	Bj, K	Branch to K if (Bi) < 0				
071JK	GT	Bj, Bj, K	Branch to K if (Bj) > (Bi)				
070JK	GT	Bj, K	Branch to K if (Bj) > 0				
070JK	MI	Bj, K	Branch to K if (Bj) < 0				
101JK	BXi	Xj	Copy (Xj) to Xi				
101JK	BXi	Xj+Xk	Logical product of (Xj) and (Xk) to Xi				
121JK	BXi	Xj+Xk	Logical sum of (Xj) plus (Xk) to Xi				
131JK	BXi	Xj-Xk	Logical difference of (Xj) minus (Xk) to Xi				
141KK	BXi	-Xk	Copy complement of (Xk) to Xi				
151JK	BXi	-Xk+Xj	Logical product of (Xj) and complement of (Xk) to Xi				
171JK	BXi	-Xk+Xj	Logical sum of (Xj) plus complement of (Xk) to Xi				
171JK	BXi	-Xk-Xj	Logical difference of (Xj) minus complement of (Xk) to Xi				
201JK	LXi		Logical-shift (Xi) by ± jk				
211JK	AXi	jk	Arithmetic-shift (Xi) by ± jk				
221JK	LXi	Bj, Xk	Logical-shift (Xk) by (Bj) to Xi				
221JK	LXi	Bj	Logical-shift (Xi) by (Bj) to Xi				
221JK	LXi	Xk	Transmit (Xk) to Xi				
221JK	LXi	Xk, Bj	Logical-shift (Xk) by (Bj) to Xi				
231JK	AXi	Bj, Xk	Arithmetic-shift (Xk) by (Bj) to Xi				
231JK	AXi	Bj	Arithmetic-shift (Xi) by (Bj) to Xi				
231JK	AXi	Xk	Transmit (Xk) to Xi				
231JK	AXi	Xk, Bj	Arithmetic-shift (Xk) by (Bj) to Xi				
241JK	NXi, Bj	Xk	Normalize (Xk) to Xi and Bj				
241JK	NXi		Normalize (Xi) to Xi				
241JK	NXi, Bj	Xk	Normalize (Xi) to Xi and Bj				
241JK	NXi	Bj, Xk	Normalize (Xk) to Xi and Bj				
241JK	NXi	Xk, Bj	Normalize (Xk) to Xi and Bj				
251JK	ZXi, Bj	Xk	Round and normalize (Xk) to Xi and Bj				
251JK	ZXi		Round and normalize (Xi) to Xi				
251JK	ZXi, Bj	Xk	Round and normalize (Xk) to Xi and Bj				
251JK	ZXi	Xk	Round and normalize (Xi) to Xi				
251JK	ZXi	Bj, Xk	Round and normalize (Xk) to Xi and Bj				
251JK	ZXi	Xk, Bj	Round and normalize (Xk) to Xi and Bj				
261JK	UXi, Bj	Xk	Unpack (Xk) to Xi and Bj				
261JK	UXi		Unpack (Xi) to Xi				
261JK	UXi, Bj	Xk	Unpack (Xi) to Xi and Bj				
261JK	UXi	Xk	Unpack (Xk) to Xi				
261JK	UXi	Bj, Xk	Unpack (Xk) to Xi and Bj				
261JK	UXi	Xk, Bj	Unpack (Xk) to Xi and Bj				

Logical Operators	
*	0100
	1101
	0100
+	0100
	1101
	1101
-	0100
	1101
	1001

Instruction	Operation	Variable	Description
271JK	PXi	Bj, Xk	Pack (Xk) and (Bj) to Xi
271JK	PXi		Pack (Xi) to Xi
271JK	PXi	Bj	Pack (Xi) and (Bj) to Xi
271JK	PXi	Xk	Pack (Xk) to Xi
271JK	PXi	Xk, Bj	Pack (Xk) and (Bj) to Xi
301JK	FXi	Xj+Xk	Sum of (Xj) plus (Xk) to Xi
311JK	FXi	Xj-Xk	Difference of (Xj) minus (Xk) to Xi
321JK	DXi	Xj+Xk	Double-precision sum of (Xj) plus (Xk) to Xi
321JK	DXi	Xj-Xk	Double-precision difference of (Xj) minus (Xk) to Xi
341JK	RXi	Xj+Xk	Rounded sum of (Xj) plus (Xk) to Xi
351JK	RXi	Xj-Xk	Rounded difference of (Xj) minus (Xk) to Xi
361JK	IXi	Xj+Xk	Integer sum of (Xj) plus (Xk) to Xi
371JK	IXi	Xj-Xk	Integer difference of (Xj) minus (Xk) to Xi
401JK	FXi	Xj*Xk	Product of (Xj) times (Xk) to Xi
411JK	RXi	Xj*Xk	Rounded product of (Xj) times (Xk) to Xi
421JK	IXi	Xj*Xk	Integer product of (Xj) times (Xk) to Xi
421JK	MXi	Xj*Xk	Double-precision product of (Xj) times (Xk) to Xi
431JK	MXi	±jk	Form mask of ± jk bits in Xi
441JK	FXi	Xj/Xk	Divide (Xj) by (Xk) to Xi
451JK	RXi	Xj/Xk	Rounded divide (Xj) by (Xk) to Xi
46000	NO		Pass (do-nothing)
471KK	CXi	Xk	Population count of (Xk) to Xi
501JK	SAi	Ai+K	(Ai) plus K to Ai
511JK	SAi	Bj+K	(Bj) plus K to Ai
511JK	SAi	K	K plus 0 to Ai
521JK	SAi	Xj+K	(Xj) plus K to Ai
531JK	SAi	Xj+Bk	(Xj) plus (Bk) to Ai
531JK	SAi	Bk+Xj	(Bk) plus (Xj) to Ai
531JK	SAi	Xj	(Xj) plus 0 to Ai
541JK	SAi	Aj+Bk	(Aj) plus (Bk) to Ai
541JK	SAi	Bk+Aj	(Bk) plus (Aj) to Ai
541JK	SAi	Aj	(Aj) plus 0 to Ai
551JK	SAi	(Aj) minus (Bk) to Ai	
551JK	SAi	-Bk+Aj	(Aj) minus (Bk) to Ai
561JK	SAi	Bj+Bk	(Bj) plus (Bk) to Ai
561JK	SAi	Bj	(Bj) plus 0 to Ai
571JK	SAi	Bj-Bk	(Bj) minus (Bk) to Ai
571JK	SAi	-Bk	0 minus (Bk) to Ai
571JK	SAi	-Bk+Bj	(Bj) minus (Bk) to Ai
601JK	SBi	Aj+K	(Aj) plus K to Bi
611JK	SBi	(Bj) plus K to Bi	
611JK	SBi	K	K plus 0 to Bi
621JK	SBi	Xj+K	(Xj) plus K to Bi
631JK	SBi	Bk+Xj	(Bk) plus (Xj) to Bi
631JK	SBi	Xj	(Xj) plus 0 to Bi
641JK	SBi	Aj+Bk	(Aj) plus (Bk) to Bi
641JK	SBi	Bk+Aj	(Bk) plus (Aj) to Bi
641JK	SBi	Aj	(Aj) plus 0 to Bi
651JK	SBi	Aj-Bk	(Aj) minus (Bk) to Bi
651JK	SBi	-Bk+Aj	(Aj) minus (Bk) to Bi
661JK	SBi	Bj+Bk	(Bj) plus (Bk) to Bi
661JK	SBi	Bj	(Bj) plus 0 to Bi
671JK	SBi	(Bj) minus (Bk) to Bi	
671JK	SBi	-Bk	0 minus (Bk) to Bi
671JK	SBi	-Bk+Bj	(Bj) minus (Bk) to Bi
701JK	SXi	(Xj) plus K to Xi	
711JK	SXi	(Bj) plus K to Xi	
711JK	SXi	K	K plus 0 to Xi
721JK	SXi	Xj+K	(Xj) plus K to Xi
731JK	SXi	Xj+Bk	(Xj) plus (Bk) to Xi
731JK	SXi	Bk+Xj	(Bk) plus (Xj) to Xi
731JK	SXi	Xj	(Xj) plus 0 to Xi
741JK	SXi	Aj+Bk	(Aj) plus (Bk) to Xi
741JK	SXi	Bk+Aj	(Bk) plus (Aj) to Xi
741JK	SXi	Aj	(Aj) plus 0 to Xi
751JK	SXi	Aj-Bk	(Aj) minus (Bk) to Xi
751JK	SXi	-Bk+Aj	(Aj) minus (Bk) to Xi
761JK	SXi	Bj+Bk	(Bj) plus (Bk) to Xi
761JK	SXi	Bj	(Bj) plus 0 to Xi
771JK	SXi	(Bj) minus (Bk) to Xi	
771JK	SXi	-Bk	0 minus (Bk) to Xi
771JK	SXi	-Bk+Bj	(Bj) minus (Bk) to Xi

CMU INSTRUCTIONS

Instruction	Operation	Variable	Description
464 0 K	IM	K	Move data according to word at K
464 j K	IM	Bj+K	Move data according to word at Bj+K
464 j 000000	IM	Bj	Move data according to word at Bj
0 j 8U K ₈	DM	8 K ₈ C ₈ d ₈ K ₈ d ₈	Indirect move descriptor word
485 8U K ₈	MD	8 K ₈ C ₈ d ₈ K ₈ d ₈	Direct move
486 8U K ₈	CC	8 K ₈ C ₈ b ₈ K ₈ b ₈	Compare collated
467 8U K ₈	CU	8 K ₈ C ₈ b ₈ K ₈ b ₈	Compare uncollated

PPU INSTRUCTIONS

Instruction	Operation	Variable	Description
01d m	LJM	m, d	Long jump to m + (d)
02d m	RJM	m, d	Return jump to m + (d)
03d	UJN	r	Unconditional jump to p + r
04d	ZJN	r	Zero jump to p + r
05d	NJN	r	Nonzero jump to p + r
06d	PJN	r	Positive jump to p + r
07d	MJN	r	Negative jump to p + r
10r	SHN	r	Shift (A) left-circular (+r) or right-end off (-r)
11d	LMN	d	Logical difference; (A) - d → A
12d	LPN	d	Logical product; (A) * d → A
13d	SCN	d	Selective clear; (A) at each d bit set
14d	LDN	d	Load d → A
15d	LCN	d	Load complement d → A
16d	ADN	d	Add d + (A) → A
17d	SBN	d	Subtract (A) - d → A
20d m	LDC	c	Load c → A
21d m	ADC	c	Add (A) + c → A
22d m	LPC	c	Logical product; (A) * c → A
23d m	LMC	c	Logical difference; (A) - c → A
2400	PSN		Pass
260d	EXN	d ②	Exchange jump CPU d unconditionally to (A)
260d	ETN	d ②	6416 Extended transfer
261d	MXN	d ②	Monitor exchange jump CPU d to (A)
262d	MAN	d ②	Monitor exchange jump CPU d to (MA)
270d	RPN	d ②	Read program address of CPU d to A
270d	ERN	d ②	6416 Extended read status
30d	LDD	d	Load (d) → A
31d	ADD	d	Add (A) + (d) → A
32d	SDD	d	Subtract (A) - (d) → A
33d	LMD	d	Logical difference; (A) and (d) → A
34d	STD	d	Store (A) → d
35d	RAD	d	Replace add; (d) + (A) → d and A
36d	AOD	d	Replace add one; (d) + 1 → d and A
37d	SOD	d	Replace subtract one; (d) - 1 → d and A
40d	LDI	d	Load (di) → A
41d	ADI	d	Add (A) + ((d)) → A
42d	SBI	d	Subtract (A) - ((d)) → A
43d	LMI	d	Logical difference; (A) - ((d)) → A
44d	STI	d	Store (A) → (d)
45d	RAI	d	Replace add; (A) + ((d)) → (d) and A
46d	AOI	d	Replace add one; ((d)) + 1 → (d) and A
47d	SOI	d	Replace subtract one; ((d)) - 1 → (d)
50d m	LDM	m, d	Load (m + (d)) → A
51d m	ADM	m, d	Add (m + (d)) + (A) → A
52d m	SBM	m, d	Subtract (A) - (m + (d)) → A
53d m	LMM	m, d	Logical difference; (A) - (m + (d)) → A
54d m	STM	m, d	Store (A) → m + (d)
55d m	RAM	m, d	Replace add; (A) + (m + (d)) → (d) and A
56d m	AOM	m, d	Replace add one; (m + (d)) + 1 → m + (d) and A
57d m	SOM	m, d	Replace subtract one; (m + (d)) - 1 → m + (d) and A
60d m	FIM	m, d ① ③	Jump to m on input word flag on channel d
60d	CRD	m, d ② ③	Central read from (A) to d
61d m	EIM	m, d ① ③	Jump to m if no input word flag on channel d
61d	CRM	m, d ② ③	Central read (d) CM words beginning from CM address (A) to beginning PPU address m
62d m	IRM	m, d ① ③	Jump to m on input record flag on channel d
62d	CWD	d ②	Central write from d to (A)
63d m	NIM	m, d ① ③	Jump to m if no input record flag on channel d
63d m	CWM	m, d ② ③	Central write (d) CM words beginning from PPU address m to beginning CM address A
64d m	FOM	m, d ① ③	Jump to m on output word flag on channel d
64d m	AJM	m, d ② ③	Jump to m if channel d is active
65d m	EOM	m, d ① ③	Jump to m if no output word flag on channel d
65d m	IJM	m, d ② ③	Jump to m if channel d is inactive
66d m	ORM	m, d ② ③	Jump to m on output record flag on channel d
66d m	FJM	m, d ② ③	Jump to m if channel d is full
67d m	NOM	m, d ① ③	Jump to m if no output record flag on channel d
67d m	EJM	m, d ② ③	Jump to m if channel d is empty
70d	IAN	d	Input to A from channel d
71d m	IAM	m, d ②	Input (A) words to m from channel d
72d	OAN	d ③	Output (A) words from A on channel d
73d m	OM	m, d ②	Output (A) words from m on channel d
74d	RFN	d ①	Send record flag on channel d
74d	ACN	d ②	Activate channel d
75d	DCN	d ②	Disconnect channel d
76d	FAN	d ②	Function (A) on channel d
77d m	FNC	m, d ② ③	Function m on channel d
7700	ESN	d ①	Error stop

CONTROL DATA® CYBER 70 SERIES COMPUTER SYSTEMS COMPASS VERSION 3	CONTROL DATA CORPORATION Pub. No. 60361700B	CONTROL DATA CORPORATION Documentation Department 215 Moffett Park Drive Sunnyvale, California 94086	July 1972
---	--	--	-----------