



**CPU INSTRUCTIONS**

- ① CYBER 70 Model 76, and 7600
- ② CYBER 170 Models 171, 172, 173, 174, 175, 720, 730, 740, 750, and 760;
- ③ CYBER 70 Models 71, 72, 73, and 74; and 6000 Series
- ④ CYBER 170 Model 176
- ⑤ CYBER 170 Models 815, 825, 835, 855, 865, and 875
- ⑥ Privileged to monitor

Instruction	Operation	Variable	Description
00000	ES ①⑤		Error exit to EEA
0000K	PS ②③④	K	Return jump to K
0100K	RJ	K	Return jump to K
011JK	RL ①③	Bj±K	Block-copy K plus (Bj) words from LCM to SCM
011JK	RE ②④	Bj±K	Read extended core storage
012JK	WL ③④	Bj±K	Block-copy K plus (Bj) words from SCM to LCM
012JK	WE ②④	Bj±K	Write extended core storage
01300	MJ ①③⑤		Exchange-exit to NEA if exit flag clear
013JK	MJ ①③⑤⑥	Bj±K	Exchange-exit to K + (Bj) if exit flag set
013JK	XJ ①③④⑤	Bj±K	Central exchange jump to Bj±K
014JK	RXJ ①③④	Xj	Read LCM at (Xj) to Xj
015JK	WXJ ①③④⑤	Xj	Write (Xj) into LCM at (Xj)
0160K	RI ①③⑤	Bk	Reset channel (Bk) input buffer
016JK	IBJ ①③⑤	Bk	Read channel (Bk) input status to Bj if j ≠ 0; otherwise, same as RI
016J0	TBJ ①③④⑤		Set Bj to current clock time
0170K	RO ①③⑤⑥	Bk	Reset channel (Bk) output buffer
017JK	OBJ ①③⑤⑥	Bk	Read channel (Bk) output status to Bj if j ≠ 0; otherwise, same as RO
0210K	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
033JK	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 (Bj) = (Bj)
0410K	ZR	Bj, K	Branch to K if (Bj) = 0
041JK	NE	Bj, Bj, K	Branch to K if (Bj) ≠ (Bj)
041JK	NZ	Bj, K	Branch to K if (Bj) ≠ 0
0510K	GE	Bj, K	Branch to K if (Bj) ≥ (Bj)
0510K	GE	Bj, K	Branch to K if (Bj) ≥ (Bj)
0610K	LE	Bj, Bj, K	Branch to K if (Bj) ≤ (Bj)
0610K	LE	Bj, K	Branch to K if (Bj) ≤ 0
0610K	PL	Bj, K	Branch to K if (Bj) > 0
0710K	LT	Bj, K	Branch to K if (Bj) < 0
071JK	LT	Bj, Bj, K	Branch to K if (Bj) < (Bj)
0710K	NG	Bj, K	Branch to K if (Bj) < 0
071JK	GT	Bj, Bj, K	Branch to K if (Bj) > (Bj)
071JK	GT	Bj, K	Branch to K if (Bj) > 0
0710K	MI	Bj, K	Branch to K if (Bj) < 0
101J	BXJ	Xj	Copy (Xj) to (Xj)
111JK	BXJ	Xj±Xk	Logical product of (Xj) and (Xk) to Xj
121JK	BXJ	Xj±Xk	Logical sum of (Xj) plus (Xk) to Xj
131JK	BXJ	Xj±Xk	Logical difference of (Xj) minus (Xk) to Xj
141kk	BXJ	-Xk	Copy complement of (Xk) to Xj
151JK	BXJ	-Xk±Xj	Logical product of (Xj) and complement of (Xk) to Xj
161JK	BXJ	-Xk±Xj	Logical sum of (Xj) plus complement of (Xk) to Xj
171JK	BXJ	-Xk±Xj	Logical difference of (Xj) minus complement of (Xk) to Xj
201JK	LXJ	Xj	Logical-shift (Xj) by ±jk
211JK	AXJ	jk	Arithmetic-shift (Xj) by ±jk
221JK	LXJ	Bj, Xk	Logical-shift (Xk) by (Bj) to Xj
221JK	LXJ	Bj	Logical-shift (Xj) by (Bj) to Xj
2210K	LXJ	Xj	Transmit (Xk) to Xj
2210K	LXJ	Xk, Bj	Logical-shift (Xk) by (Bj) to Xj
231JK	AXJ	-j, Xk	Arithmetic-shift (Xk) by (Bj) to Xj
231JK	AXJ	Bj	Arithmetic-shift (Xj) by (Bj) to Xj
2310K	AXJ	.k	Transmit (Xk) to Xj
231JK	AXJ	Xk, Bj	Arithmetic-shift (Xk) by (Bj) to Xj
241JK	NXJ, Bj	Xk	Normalize (Xk) to Xj and Bj
2410K	NXJ	Xj	Normalize (Xj) to Xj and Bj
241JK	NXJ	Xk	Normalize (Xk) to Xj and Bj
241JK	NXJ	Xk, Bj	Normalize (Xk) to Xj and Bj
251JK	ZXJ, Bj	Xk	Round and normalize (Xk) to Xj and Bj
2510K	ZXJ	Xj	Round and normalize (Xj) to Xj and Bj
251JK	ZXJ	Xk, Bj	Round and normalize (Xk) to Xj and Bj
251JK	ZXJ	Xk, Bj	Round and normalize (Xk) to Xj and Bj
2610K	UXJ, Bj	Xk	Unpack (Xk) to Xj and Bj
2610K	UXJ	Xj	Unpack (Xj) to Xj and Bj
2610K	UXJ	Xk, Bj	Unpack (Xk) to Xj and Bj
2610K	UXJ	Xj, Bj	Unpack (Xj) to Xj and Bj

**Logical Operators**

*	0100
	1101
	0100
+	0100
	1101
	1101
-	0100
	1101
	1001

Instruction	Operation	Variable	Description
271JK	PXJ	Bj, Xk	Pack (Xk) and (Bj) to Xj
2710i	PXJ		Pack (Xj) to Xj
271ji	PXJ	Bj	Pack (Xj) and (Bj) to Xj
2710K	PXJ	Xk	Pack (Xk) to Xj
271JK	PXJ	Xj, Bj	Pack (Xj) and (Bj) to Xj
301JK	FXJ	Xj+Xk	Sum of (Xj) plus (Xk) to Xj
311JK	FXJ	Xj-Xk	Difference of (Xj) minus (Xk) to Xj
321JK	DXJ	Xj+Xk	Double-precision sum of (Xj) plus (Xk) to Xj
331JK	DXJ	Xj-Xk	Double-precision difference of (Xj) minus (Xk) to Xj
341JK	RXJ	Xj+Xk	Rounded sum of (Xj) plus (Xk) to Xj
351JK	RXJ	Xj-Xk	Rounded difference of (Xj) minus (Xk) to Xj
361JK	IXJ	Xj+Xk	Integer sum of (Xj) plus (Xk) to Xj
371JK	IXJ	Xj-Xk	Integer difference of (Xj) minus (Xk) to Xj
401JK	FXJ	Xj * Xk	Product of (Xj) times (Xk) to Xj
411JK	RXJ	Xj * Xk	Rounded product of (Xj) times (Xk) to Xj
421JK	IXJ	Xj * Xk	Integer product of (Xj) times (Xk) to Xj
421JK	DXJ	Xj * Xk	Double-precision product of (Xj) times (Xk) to Xj
431JK	MXJ	±jk	Form mask of ±jk bits in Xj
441JK	FXJ	Xj/Xk	Divide (Xj) by (Xk) to Xj
451JK	RXJ	Xj/Xk	Rounded divide (Xj) by (Xk) to Xj
46000	NO		Pass (do-nothing)
471JK	CXJ	Xk	Population count of (Xk) to Xj
501JK	SAJ	Aj±K	(A)j plus K to Aj
511JK	SAJ	Bj±K	(B)j plus K to Aj
510JK	SAJ	K	K plus 0 to Aj
521JK	SAJ	Xj±K	(Xj) plus K to Aj
531JK	SAJ	Xj+Bk	(Xj) plus (Bk) to Aj
531JK	SAJ	Bk+Xj	(Bk) plus (Xj) to Aj
5310K	SAJ	Xj	(Xj) plus 0 to Aj
541JK	SAJ	Aj+Bk	(Aj) plus (Bk) to Aj
541JK	SAJ	Bk+Aj	(Bk) plus (Aj) to Aj
5410K	SAJ	Aj	(Aj) plus 0 to Aj
551JK	SAJ	Aj-Bk	(Aj) minus (Bk) to Aj
551JK	SAJ	Bk-Aj	(Bk) minus (Aj) to Aj
5510K	SAJ	Bj	(Bj) plus 0 to Aj
561JK	SAJ	Bj-Bk	(Bj) minus (Bk) to Aj
5710K	SAJ	-Bk	0 minus (Bk) to Aj
571JK	SAJ	-Bk+Bj	(Bj) minus (Bk) to Aj
601JK	SBJ	Aj±K	(Aj) plus K to Bj
611JK	SBJ	Bj±K	(Bj) plus K to Bj
610JK	SBJ	K	K plus 0 to Bj
621JK	SBJ	Xj±K	(Xj) plus K to Bj
631JK	SBJ	Bk+Xj	(Bk) plus (Xj) to Bj
6310K	SBJ	Xj	(Xj) plus 0 to Bj
641JK	SBJ	Aj+Bk	(Aj) plus (Bk) to Bj
641JK	SBJ	Bk+Aj	(Bk) plus (Aj) to Bj
6410K	SBJ	Aj	(Aj) plus 0 to Bj
651JK	SBJ	Aj-Bk	(Aj) minus (Bk) to Bj
6510K	SBJ	-Bk+Aj	(Aj) minus (Bk) to Bj
661JK	CF ④	Xj, Xk	Read central memory
661JK	SBJ	Bj+Bk	(Bj) plus (Bk) to Bj
6610K	SBJ	Bj	(Bj) plus 0 to Bj
6610K	CW ④	Xj, Xk	Write central memory
671JK	SBJ	Bj-Bk	(Bj) minus (Bk) to Bj
6710K	SBJ	-Bk	0 minus (Bk) to Bj
671JK	SBJ	-Bk+Bj	(Bj) minus (Bk) to Bj
701JK	SXJ	Aj±K	(Aj) plus K to Xj
711JK	SXJ	Bj±K	(Bj) plus K to Xj
7110K	SXJ	K	K plus 0 to Xj
721JK	SXJ	Xj+K	(Xj) plus K to Xj
731JK	SXJ	Xj+Bk	(Xj) plus (Bk) to Xj
7310K	SXJ	Bk+Xj	(Bk) plus (Xj) to Xj
741JK	SXJ	Xj	(Xj) plus 0 to Xj
741JK	SXJ	Aj+Bk	(Aj) plus (Bk) to Xj
7410K	SXJ	Bk+Aj	(Bk) plus (Aj) to Xj
751JK	SXJ	Aj	(Aj) plus 0 to Xj
751JK	SXJ	Aj-Bk	(Aj) minus (Bk) to Xj
7510K	SXJ	-Bk+Aj	(Aj) minus (Bk) to Xj
761JK	SXJ	Bj+Bk	(Bj) plus (Bk) to Xj
7610K	SXJ	Bj	(Bj) plus 0 to Xj
771JK	SXJ	Bj-Bk	(Bj) minus (Bk) to Xj
7710K	SXJ	-Bk	0 minus (Bk) to Xj
771JK	SXJ	-Bk+Bj	(Bj) minus (Bk) to Xj

**FUNCTIONAL UNITS**

Model 74 Octal Codes	Model 76 Octal Codes
Branch	Boolean
00-07	10-17
Boolean	26-27
10-17	Shift
20-27	20-23
43	43
FP Add	24, 25
30-35	FP Add
Long Add	30-35
36-37	Long Add
FP Multiply	36, 37
40-42	FP Multiply
FP Divide	40-42
44, 45, 47	FP Divide
Increment	44, 45
50-77	Population
	47
	Increment
	50-77

**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 Uj k <sub>a</sub>		MD R <sub>k</sub> , k <sub>a</sub> , k <sub>d</sub> , c <sub>d</sub>	Indirect move descriptor word
465 Uj k <sub>a</sub>		DM R <sub>k</sub> , k <sub>a</sub> , k <sub>d</sub> , c <sub>d</sub>	Direct move
466 Uj k <sub>a</sub>		CC R <sub>k</sub> , k <sub>a</sub> , k <sub>b</sub> , c <sub>b</sub>	Compare collated
467 Uj k <sub>a</sub>		CU R <sub>k</sub> , k <sub>a</sub> , k <sub>b</sub> , c <sub>b</sub>	Compare uncollated

**PP INSTRUCTIONS**

- ① CYBER 70 Model 76, and 7600
- ② CYBER 170 Models 171, 172, 173, 174, 175, 720, 730, 740, 750, and 760;
- ③ CYBER 70 Models 71, 72, 73, and 74; and 6000 Series
- ④ CYBER 170 Model 176
- ⑤ CYBER 170 Models 815, 825, 835, 855, 865, and 875
- ⑥ j is required

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 r	UJN	r	Unconditional jump to p + r
04d ZJN	r	r	Zero jump to p + r
05d NJN	r	r	Nonzero jump to p + r
06d PJN	r	r	Positive jump to p + r
07d MJN	r	r	Negative jump to p + r
10d SHN	r	r	Shift (A) left-circular (+r) or right-end off (-r)
11d LMN	d	d	Logical difference; (A) - d → A
12d LPN	d	d	Logical product; (A) * d → A
13d SCN	d	d	Selective clear; (A) at each bit set
14d LDN	d	d	Load d → A
15d LCN	d	d	Load complement d → A
16d ADN	d	d	Add d + (A) → A
17d SBN	d	d	Subtract (A) - d → A
20d m	LDC	c	Load c → A
21d m	ADC	c	Add (A) + c → A
22d m	APC	c	Logical product; (A) * c → A
23d m	LMC	c	Logical difference; (A) - c → A
2400	PSN	c	Pass
24d LRD	d	d	Load (R) from d and d+1
25d SRD	d	d	Store (R) into d and d+1
260d EXN	d	d	Exchange jump CPU d unconditionally to (A)
261d ETN	d	d	6416 Extended transfer
261d MXN	d	d	Monitor exchange jump CPU d to (A)
262d MAN	d	d	Monitor exchange jump CPU d to (MA)
270d RPN	d	d	Read program address of CPU d to A
270d ERN	d	d	6416 Extended read status
30d LDD	d	d	Load (d) → A
31d ADD	d	d	Add (A) + (d) → A
31d SGD	d	d	Subtract (A) - (d) → A
32d LMD	d	d	Logical difference; (A) and (d) → A
34d STD	d	d	Store (A) → d
35d RAD	d	d	Replace add; (d) + (A) → d and A
36d AOD	d	d	Replace add one; (d) + 1 → d and A
37d SOD	d	d	Replace subtract one; (d) - 1 → d and A
40d LDI	d	d	Load (d) → A
41d ADI	d	d	Add (A) + (d) → A
42d SBI	d	d	Subtract (A) - (d) → A
43d LMI	d	d	Logical difference; (A) - (d) → A
44d STI	d	d	Store (A) → (d)
45d RAI	d	d	Replace add; (A) + (d) → (d) and A
46d AOI	d	d	Replace add one; (d) + 1 → (d) and A
47d SOI	d	d	Replace subtract one; (d) - 1 → (d) and A
50d m	LDM	m, d	Load (m + (d)) → A
51d m	ADM	m, d	Add (m + (d)) + (A) → A
53d m	SBM	m, d	Subtract (A) - (m + (d)) → A
54d 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)) → m + (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	Jump to m if no input word flag on channel d
60d	ERM	m, d	Central read from (A) to d
61d m	ERM	m, d	Jump to m if no input word flag on channel d
61d	CRM	m, d	Central read (CM words beginning from CM address (A) to beginning PPU address m
62d m	IRM	m, d	Jump to m if no input record flag on channel d
62d	CWD	m, d	Central write from (d) to (A)
63d m	NIM	m, d	Jump to m if no input record flag on channel d
63d m	CWM	m, d	Central write (d) CM words beginning from PPU address m to beginning CM address (A)
644d m	SCF	m, d	Branch to m if d flag set
64d m	FOM	m, d	Jump to m on output word flag on channel d
64d m	AJM	m, d	Jump to m if channel d is active
654d m	OCF	m, d	Clear channel d flag
65d m	ECOM	m, d	Jump to m if no output word flag on channel d
65d m	LJM	m, d	Jump to m if channel d is inactive
664d m	SFM	m, d	Branch to m if channel d error flag on channel d
66d m	ORM	m, d	Jump to m on output record flag on channel d
66d m	FJM	m, d	Jump to m if channel d is full
674d m	CFM	m, d	Branch to m if channel d error flag clear
67d m	NOM	m, d	Jump to m if no output record flag on channel d
67d m	ECOM	m, d	Jump to m if channel d is empty
70d	IAN	d	Input to A from channel d
71d m	IAM	m, d	Input (A) words to m from channel d
72d	OAN	d	Output from A on channel d
73d m	OAM	m, d	Output (A) words from m on channel d
74d	RFN	d	Send record flag on channel d
74d	ACN	d	Activate channel d
75d	DCN	d	Deactivate channel d
76d	FAN	d	Function (A) on channel d
77d m	FNC	m, d	Function m on channel d
7700	ESN	d	Error stop