60458100

CONTROL DATA

CYBER 170 STATE
VIRTUAL STATE

CODES BOOKLET

REVISION RECORD

REV	DESCRIPTION
A (12-21-82)	Manual released.
B (07-15-84)	Revised per PCO 45823 to add support of CYBER 180 Models 810, 830, 835, 845, 850, 855, 860, and 990 to existing Codes Booklet. Because extensive changes are made, change bars and dots are not used and all pages reflect the latest revision level. This edition obsoletes all previous editions.
C (06-26-86)	Manual revised; includes Engineering Change Order 47774. Front cover, inside front cover, and pages 2 through 4, and 7 are revised. Inside back cover information and page 4.1/4.2 are added.
D (12-01-86	Manual revised; includes Engineering Change Order 48390. Front cover, inside front cover, 2 through 4.2, 7, and inside back cover are revised. Pages 28.1 and 28.2 are added.
E (04-24-87)	Manual revised; includes Engineering Change Order 48575. Front cover, 2 through 4.1, 7, and inside back cover are revised.
F (08-01-88)	Manual revised; includes Publications Change Order 49674. Because extensive changes are made, change bars and dots are not used and all pages reflect the latest revision level. This edition obsoletes all previous editions.

Publication No. 60458100

Revision letters I, O, Q, S, X, and Z are not used.

Address comments to:

Control Data
Technical Publications
4201 North Lexington Avenue
St. Paul, Minnesota 55126-6198

©1982, 1984, 1986, 1987, 1988
by Control Data Corporation
All Rights Reserved
Printed in USA

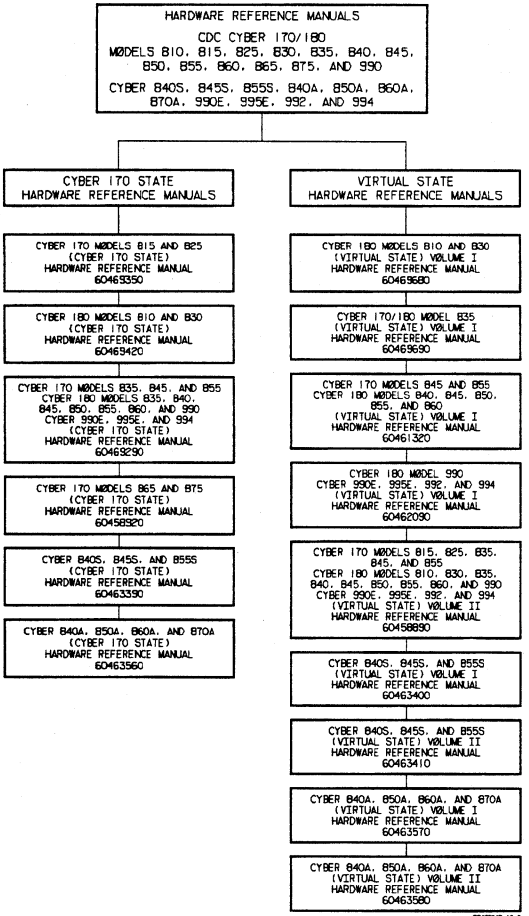
PREFACE

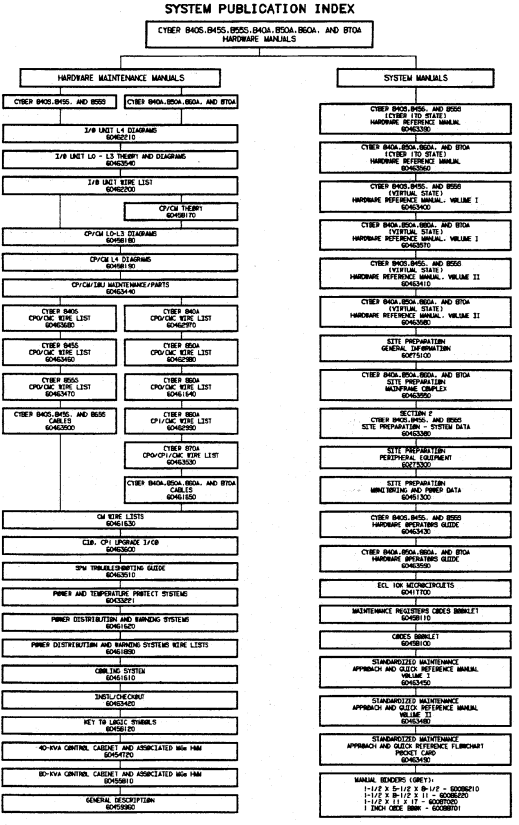
This codes booklet summarizes all instructions and codes for the CONTROL DATA® CYBER 170 Computer Systems, Models 815, 825, 835, 845, and 855, the CDC® CYBER 180 Computer Systems, Models 810, 830, 835, 840, 845, 850, 855, 860, and 990, and the CDC CYBER 840S, 845S, 855S, 840A, 850A, 860A, 870A, 960, 962, 992, 994, 990E, and 995E Computer Systems.

The information in this booklet is from the hardware reference manuals listed in the system publication indexes which follow. This information does not supplant nor is it as complete as the information contained in the reference manuals. The reference manuals are, at all times, more authoritative, reliable, and current than the codes booklet.

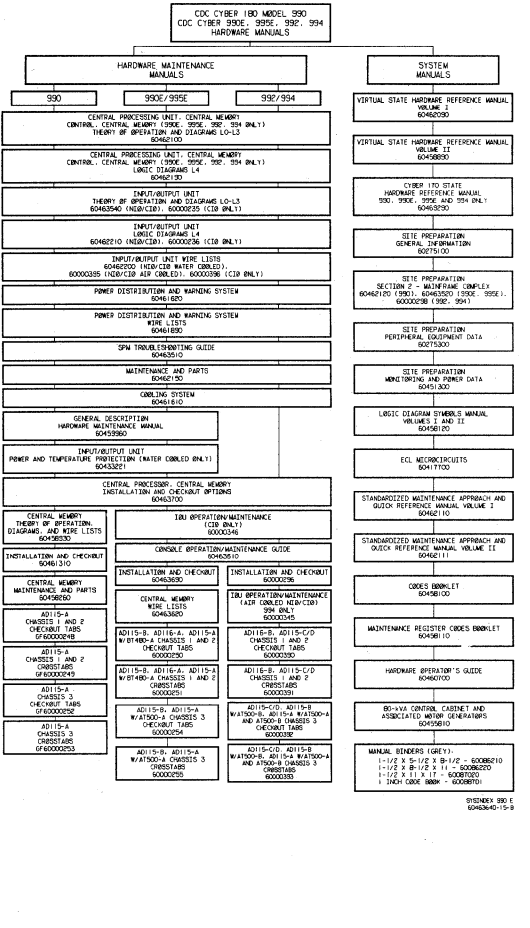
Refer to the Literature and Distribution Services Catalog for the latest manual revision levels and literature ordering procedures.

SYSTEM PUBLICATION INDEX

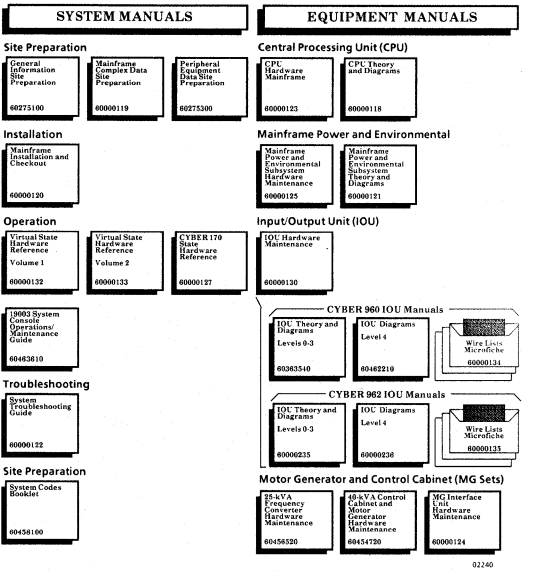



SYSTEM PUBLICATION INDEX
CYBER 840S,845S,855S,840A,850A,860A, AND 870A HARDWARE MANUALS
HARDWARE MAINTENANCE MANUALS
SYSTEM MANUALS
CYBER 840S,845S, AND 855S
CYBER 840A,850A,860A, AND 870A
I/O UNIT L4 DIAGRAMS 60462210
I/O UNIT L0 - L3 THERMY AND DIAGRAMS 60463540
I/O UNIT WIRE LIST 60462200
CP/CM THERMY 60458170
CP/CM L0-L3 DIAGRAMS 60458180
CP/CM L4 DIAGRAMS 60458190
CP/CM/DBJ MAINTENANCE/PARTS 60463440
CYBER 840S CPU/CMC WIRE LIST 60463680
CYBER 840A CPU/CMC WIRE LIST 60462970
CYBER 845S CPU/CMC WIRE LIST 60463460
CYBER 850A CPU/CMC WIRE LIST 60462980
CYBER 855S CPU/CMC WIRE LIST 60463470
CYBER 860A CPU/CMC WIRE LIST 60461640
CYBER 840S,845S, AND 855S CABLES 60463500
CYBER 860A CPI/CMC WIRE LIST 60462990
CYBER 870A CPU/CPI/CMC WIRE LIST 60463530
CYBER 840A,850A,860A, AND 870A CABLES 60461650
CM WIRE LISTS 60461630
CIB, CPI UPGRADE I/CB 60462600
SPM TROUBLESHOOTING GUIDE 60463510
PRIMER AND TEMPERATURE PROTECT SYSTEMS 60433291
PRIMER DISTRIBUTION AND WARNING SYSTEMS 60461620
PRIMER DISTRIBUTION AND WARNING SYSTEMS WIRE LISTS 60461630
COOLING SYSTEM 60461610
INSTL/CHECKOUT 60463420
KEY TO LOGIC SYMBOLS 60458120
40-KVA CONTROL CABINET AND ASSOCIATED MGe M&M 60454720
80-KVA CONTROL CABINET AND ASSOCIATED MGe M&M 60455810
GENERAL DESCRIPTION 60459360
CYBER 840S,845S, AND 855S (CYBER 170 STATE) HARDWARE REFERENCE MANUAL 60463330
CYBER 840A,850A,860A, AND 870A (CYBER 170 STATE) HARDWARE REFERENCE MANUAL 60463560
CYBER 840S,845S, AND 855S (VIRTUAL STATE) HARDWARE REFERENCE MANUAL, VOLUME I 60463400
CYBER 840A,850A,860A, AND 870A (VIRTUAL STATE) HARDWARE REFERENCE MANUAL, VOLUME I 60463510
CYBER 840S,845S, AND 855S (VIRTUAL STATE) HARDWARE REFERENCE MANUAL, VOLUME II 60463410
CYBER 840A,850A,860A, AND 870A (VIRTUAL STATE) HARDWARE REFERENCE MANUAL, VOLUME II 60463580
SITE PREPARATION GENERAL INFORMATION 60475100
CYBER 840A,850A,860A, AND 870A SITE PREPARATION MAINFRAME/PLEX 60463850
SECTION F CYBER 840S,845S, AND 855S SITE PREPARATION - SYSTEM DATA 60463380
SITE PREPARATION PERIPHERAL EQUIPMENT 60475300
SITE PREPARATION WIRE/WIRING AND POWER DATA 60461300
CYBER 840S,845S, AND 855S HARDWARE OPERATORS GUIDE 60463430
CYBER 840A,850A,860A, AND 870A HARDWARE OPERATORS GUIDE 60463580
ECL 10K MICROCIRCUITS 60417700
MAINTENANCE REGISTERS CODES BOOKLET 60458110
CODES BOOKLET 60458100
STANDARDIZED MAINTENANCE APPROACH AND QUICK REFERENCE MANUAL VOLUME I 60463450
STANDARDIZED MAINTENANCE APPROACH AND QUICK REFERENCE MANUAL VOLUME II 60463460
STANDARDIZED MAINTENANCE APPROACH AND QUICK REFERENCE FLOWCHART POCKET CARD 60463430
MANUAL RENDERING (GREY)= 1-1/2 X 5-1/2 X 8-1/2 - 60008210 1-1/2 X 8-1/2 X 11 - 60056220 1-1/2 X 11 X 17 - 60007020 1 INCH CODE BOOK - 60008101
60463540-14-A

SYSTEM PUBLICATION INDEX



CYBER 960/962 HARDWARE MANUALS



CONTENTS

Introduction 11

Central Processor Instruction Formats (CYBER 170 State) 11

CYBER 170 State Instructions 15

Central Processor Instruction Formats (Virtual State) 18

Virtual State System Instructions 21

Virtual State General Instructions 22

Virtual State Floating-Point Instructions 25

Virtual State BDP Instructions 26

BDP Descriptor 26

Virtual State Vector Instructions 27

Interrupt Conditions 29

Peripheral Processor Instruction Formats (Virtual State and CYBER 170 State) 44

PP Instructions (Virtual State and CYBER 170 State) 45

Virtual State Exchange Package 48

Interstate Exchange Package 49

Detail for Virtual State and Interstate Exchange Packages 50

CYBER 170 State Exchange Package 50

Stack Frame Save Area 51

Stack Frame Save Area Descriptor 51

Code Base Pointer - CBP 51

Program Address Register - P Register 52

Segment Descriptor Table Entry - SDE 52

Page Descriptor 52

Processor Register Definitions and Accesses 53

Purge Buffer (05) 53

Monitor Condition Register (42) 54

User Condition Register (43) 54

BDP Data Types 56

Floating Point Representation 57

Character Set 58

Glossary 60

FIGURES

1 CP CYBER 170 State INstruction Formats 12

TABLES

1 CP CYBER 170 State Instruction Designators 13

2 PP Instruction Designators 43

INTRODUCTION

This codes booklet is a concise summary of all instructions and codes related to the CYBER 170 Series 800 Computer Systems, the CYBER 180 Series 800 and 900 Computer Systems, and the CYBER 840S, 845S, 855S, 840A, 850A, 860A, 870A, 960, 962, 992, 994, 990E, and 995E Computer Systems. This booklet is an abstract of the computer system hardware reference manual and is not intended for use as a stand alone document. Refer to the appropriate hardware reference manual listed in the preface for detailed information. All mnemonics listed in this booklet apply to the COMPASS assembly language.

CENTRAL PROCESSOR INSTRUCTION FORMATS (CYBER 170 STATE)

CYBER 170 State central processor (CP) instructions use the rightmost 60 bits in the 64-bit word. The leftmost 4 bits are undefined. For these instructions, the most significant bit is bit 59 and the least significant bit is bit 0.

Program instruction words are divided into 15-bit fields called parcels. The first parcel (parcel 0) is the highest-order 15 bits of the 60-bit word. The second, third, and fourth parcels (parcels 1, 2, and 3) follow in order. Figure 1 shows possible parcel arrangements for instructions within a program instruction word. Table 1 lists and defines the instruction designators. The complete CYBER 170 State instruction set is listed following figure 1 and table 1.

An instruction may occupy one, two, or four parcels. This arrangement depends upon the instruction format. When an instruction occupies two parcels, it must occupy two parcels within the same program word. A program word may be filled with a one-parcel pass instruction or an instruction acting as a two-parcel pass instruction. These instructions are used to fill a program word when it is necessary to place a particular instruction in the first parcel of a program word or to avoid starting a two-parcel instruction in the fourth parcel of a program word. Pass instructions may also be used for branch entry points because a branch instruction destination address must begin with a new word. One-parcel pass instructions are 460xx through 463xx. Instructions 60xxx through 62xxx may be used as two-parcel pass instructions by setting the i instruction designator to zero.

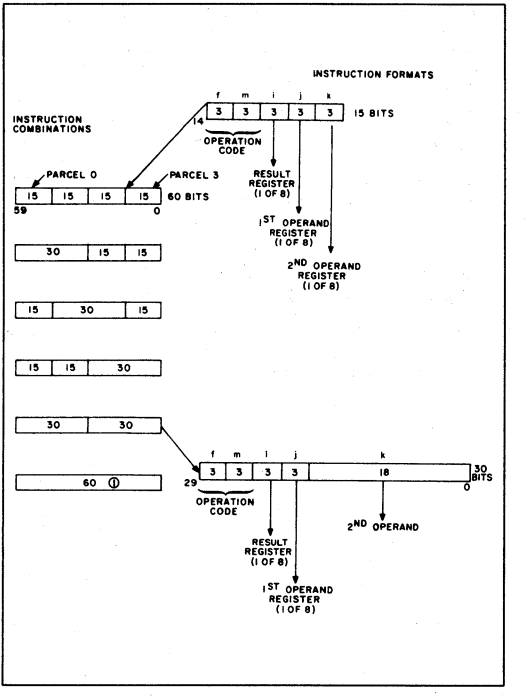


Figure 1. CP CYBER 170 State Instruction Formats

Table 1. CP CYBER 170 State
Instruction Designators (Sheet 1 of 2)

Designator	Description
Opcode	6-bit/9-bit field specifying instruction operation code.
i,j,k	3-bit code specifying one of eight registers.
jk	6-bit code specifying amount of shift or mask.
K	18-bit operand or address.
x	Unused designator.
A	One of eight 18-bit address registers.
B	One of eight 18-bit index registers; B0 is fixed and equal to zero.
X	One of eight 60-bit operand registers.
()	Content of the word at a central memory address.

Table 1. CP CYBER 170 State
Instruction Designators (Sheet 2 of 2)

Designator	Description
C1*	Offset (character address) of the first character in the first word of the source field.
C2*	Character address of the first character in the first word of the result field.
K1*	18-bit address indicating the central memory location of the first (leftmost) character of the source field.
K2*	18-bit address indicating the central memory location of the first (leftmost) character of the source field.
LL*	Lower 4 bits of the field length (character count) for a move or compare instruction; used with LU to specify field length.
LU*	Upper 9 bits of the field length; (character count) for indirect move instruction or the upper 3 bits for direct instructions; used with LL to specify field length.

* Applicable to compare/move instructions only.

CYBER 170 STATE INSTRUCTIONS (Sheet 1 OF 3)

OP CODE	MNEMONIC	NAME	OP CODE	MNEMONIC	NAME
00xxx	PS	ERROR EXIT TO MA OR PROGRAM STOP	036jK	DF	BRANCH TO K IF (Xj) DEFINITE
			037jK	ID	BRANCH TO K IF (Xj) INDEFINITE
010xK	RJ	RETURN JUMP TO K	04ijK	EQ	BRANCH TO K IF (Bi)=(Bj)
011jK	REC Bj+K	BLOCK COPY (Bj)+K WORDS FROM (X0)+(RAe) TO (A0)+(RAc)	05ijK	NE	BRANCH TO K IF (Bi)≠(Bj)
			06ijK	GE	BRANCH TO K IF (Bi)≥(Bj)
012jK	WEC Bj+K	BLOCK COPY (Bj)+K WORDS FROM (A0)+(RAc) TO (X0)+(RAe)	07ijK	LT	BRANCH TO K IF (Bi)<(Bj)
013jK	XJ Bj+K	CENTRAL EXCHANGE JUMP TO (Bj)+K (MONITOR FLAG SET)	10ijx	BXi Xj	TRANSMIT (Xj) TO Xi
			11ijk	BXi Xj*Xk	LOGICAL PRODUCT OF (Xj) AND (Xk) TO Xi
013xx	XJ	CENTRAL EXCHANGE JUMP TO MA (MONITOR FLAG NOT SET)	12ijk	BXi Xj+Xk	LOGICAL SUM OF (Xj) AND (Xk) TO Xi
014jk	RXj Xk	READ ONE WORD FROM ((Xk)+RAe) TO Xj	13ijk	BXi Xj-Xk	LOGICAL DIFFERENCE OF (Xj) AND (Xk) TO Xi
015jk	WXj Xk	WRITE ONE WORD FROM Xj TO ((Xk)+RAe)	14ixk	BXi -Xk	TRANSMIT COMPLEMENT OF (Xk) TO Xi
016jk	RC	READ FREE RUNNING COUNTER	15ijk	BXi -Xk*Xj	LOGICAL PRODUCT OF (Xj) AND COMPLEMENT OF (Xk) TO Xi
017jk	RT	TRAP 180	16ijk	BXi -Xk+Xj	LOGICAL SUM OF (Xj) AND COMPLEMENT OF (Xk) TO Xi
02ixK	JP	JUMP TO (Bi)+K	17ijk	BXi -Xk-Xj	LOGICAL DIFFERENCE OF (Xj) AND COMPLEMENT OF (Xk) TO Xi
030jK	ZR	BRANCH TO K IF (Xj)=0	20ijk	LXi jk	LEFT SHIFT (Xi) BY jk
031jK	NZ	BRANCH TO K IF (Xj)≠0	21ijk	AXi jk	RIGHT SHIFT (Xi) BY jk
032jK	PL	BRANCH TO K IF (Xj) POSITIVE	22ijk	LXi Bj Xk	LEFT SHIFT (Xk) NOMINALLY (Bj) PLACES TO Xi
033jK	NG	BRANCH TO K IF (Xj) NEGATIVE			
034jK	IR	BRANCH TO K IF (Xj) IN RANGE			
035jK	OR	BRANCH TO K IF (Xj) OUT OF RANGE			

CYBER 170 STATE INSTRUCTIONS (Sheet 2 OF 3)

OP CODE	MNEMONIC	NAME	OP CODE	MNEMONIC	NAME
23ijk	AXi Bj Xk	RIGHT SHIFT (Xk) NOMINALLY (Bj) PLACES TO Xi	50ijK	SAi Aj+K	SET Ai TO (Aj)+K
			51ijK	SAi Bj+K	SET Ai TO (Bj)+K
24ijk	NXi Bj Xk	NORMALIZE (Xk) TO Xi AND Bj	52ijK	SAi Xj+K	SET Ai TO (Xj)+K
25ijk	ZXi Bj Xk	ROUND NORMALIZE (Xk) TO Xi AND Bj	53ijk	SAi Xj+Bk	SET Ai TO (Xj)+(Bk)
			54ijk	SAi Aj+Bk	SET Ai TO (Aj)+(Bk)
26ijk	UXi Bj Xk	UNPACK (Xk) TO Xi AND Bj	55ijk	SAi Aj-Bk	SET Ai TO (Aj)-(Bk)
27ijk	PXi Bj Xk	PACK (Xk) AND (Bj) TO Xi	56ijk	SAi Bj+Bk	SET Ai TO (Bj)+(Bk)
30ijk	FXi Xj+Xk	FLOATING SUM OF (Xj) AND (Xk) TO Xi	57ijk	SAi Bj-Bk	SET Ai TO (Bj)-(Bk)
31ijk	FXi Xj-Xk	FLOATING DIFFERENCE OF (Xj) AND (Xk) TO Xi	60ijK	SBi Aj+K	SET Bi TO (Aj)+K
			61ijK	SBi Bj+K	SET Bi TO (Bj)+K
32ijk	DXi Xj+Xk	FLOATING DOUBLE-PRECISION SUM OF (Xj) AND (Xk) TO Xi	62ijK	SBi Xj+K	SET Bi TO (Xj)+K
			63ijk	SBi Xj+Bk	SET Bi TO (Xj)+(Bk)
33ijk	DXi Xj-Xk	FLOATING DOUBLE-PRECISION DIFFERENCE OF (Xj) AND (Xk) TO Xi	64ijk	SBi Aj+Bk	SET Bi TO (Aj)+(Bk)
			65ijk	SBi Aj-Bk	SET Bi TO (Aj)-(Bk)
34ijk	RXi Xj+Xk	ROUND FLOATING SUM OF (Xj) AND (Xk) TO Xi	66ijk	SBi Bj+Bk	SET Bi TO (Bj)+(Bk)
			660jk	CRXj Xk	READ CM AT (Xk) TO Xj
35ijk	RXi Xj-Xk	ROUND FLOATING DIFFERENCE OF (Xj) AND (Xk) TO Xi	67ijk	SBi Bj-Bk	SET Bi TO (Bj)-(Bk)
			670jk	CWXj Xk	WRITE Xj INTO CM AT (Xk)
36ijk	IXi Xj+Xk	INTEGER SUM OF (Xj) AND (Xk) TO Xi	70ijK	SXi Aj+K	SET Xi TO (Aj)+K
			71ijK	SXi Bj+K	SET Xi TO (Bj)+K
37ijk	IXi Xj-Xk	INTEGER DIFFERENCE OF (Xj) AND (Xk) TO Xi	72ijK	SXi Xj+K	SET Xi TO (Xj)+K
			73ijk	SXi Xj+Bk	SET Xi TO (Xj)+(Bk)
40ijk	FXi Xj*Xk	FLOATING PRODUCT OF (Xj) AND (Xk) TO Xi			

CYBER 170 STATE INSTRUCTIONS (Sheet 3 OF 3)

OP CODE	MNEMONIC	NAME	OP CODE	MNEMONIC	NAME
41ijk	RXi Xj*Xk	ROUND FLOATING PRODUCT OF (Xj) AND (Xk) TO Xi	74ijk	SXi Aj+Bk	SET Xi TO (Aj)+(Bk)
			75ijk	SXi Aj-Bk	SET Xi TO (Aj)-(Bk)
42ijk	DXi Xj*Xk	FLOATING DOUBLE-PRECISION PRODUCT OF (Xj) AND (Xk) TO Xi	76ijk	SXi Bj+Bk	SET Xi TO (Bj)+(Bk)
			77ijk	SXi Bj-Bk	SET Xi TO (Bj)-(Bk)
43ijk	MXi jk	FORM MASK OF jk BITS TO Xi			
44ijk	FXi Xj/Xk	FLOATING DIVIDE (Xj) BY (Xk) TO Xi			
45ijk	RXi Xj/Xk	ROUND FLOATING DIVIDE (Xj) BY (Xk) TO Xi			
460xx	NO	PASS			
464jK	IM	MOVE INDIRECT*			
465	DM	MOVE DIRECT*			
466	CC	COMPARE COLLATED*			
467	CU	COMPARE UNCOLLATED*			
47ixk	CXi Xk	POPULATION COUNT OF (Xk) TO Xi			

*INSTRUCTIONS 464 THROUGH 467 ARE DETECTED AS UNIMPLEMENTED INSTRUCTIONS IN THE C170 STATE.

CENTRAL PROCESSOR INSTRUCTION FORMATS (VIRTUAL STATE)

Virtual State central processor (CP) instructions use the entire 64 bits in the 64-bit word. For these instructions, the most significant bit is bit 59 and the least significant bit is bit 0.

Program instruction words are divided into 16-bit fields called parcels. The first parcel (parcel 0) is the lowest-order 16 bits of the 64 bit word. The second, third, and fourth parcels (parcels 1, 2, and 3) follow in order.

An instruction may occupy one or two parcels. Since Virtual State instructions are byte-addressable rather than word-addressable (CYBER 170 State), two-parcel instructions can span instruction-word boundaries without the insertion of pass (no-op) instructions.

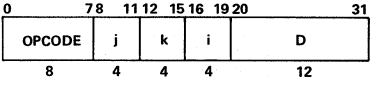
Virtual State instructions are noninterruptible. The CP always prevalidates an instruction before executing it.

The Virtual State CP instructions comprise the following five groups. (The instructions are listed by group after the Virtual State instruction formats described in the following paragraphs.)

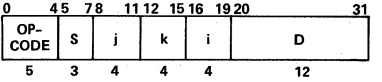
- System
- General
- Floating point (FP)
- Business data processing (BDP)
- Vector

The Virtual State CP Virtual State instructions are 16 or 32 bits long and have four basic formats.

Format jkiD (32 Bits)



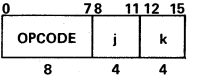
Format SjkiD (32 Bits)



Field	Description
Opcode	Operation code.
j,k,i	Register designators.
D	Signed shift count, positive displacement, or bit string descriptor.
S	Suboperation code.

Business data processing (BDP) instructions using these formats also have one or two 64-bit data descriptor words which are stored in CM immediately after the instruction.

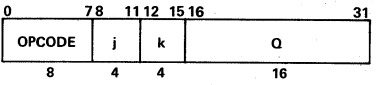
Format jk (16 Bits)



Field	Description
Opcode	Operation Code.
j	Register designator, suboperation code, or immediate operand.
k	Register designator or immediate operand.

BDP instructions using this format have two data descriptor words which are stored in CM immediately after the instruction. (Refer to BDP Data Descriptors later in this manual.)

Format jkQ (32 Bits)



Field	Description
Opcode	Operation code.
j,k	Register designators, suboperation codes, or immediate operand value.
Q	Signed displacement or immediate operand value.

VIRTUAL STATE SYSTEM INSTRUCTIONS

OP CODE	FORMAT	MNEMONIC	NAME	ADDRESS FIELD
B5	jkQ	CALLSEG	CALL INDIRECT	LABEL, Aj, Ak
B0	jkQ	CALLREL	CALL RELATIVE	
04	jk	RETURN	RETURN	
06	jk	POP	POP	(BLANK)
02	jk	EXCHANGE	EXCHANGE	
03	jk	INTRUPT	PROCESSOR INTERRUPT	Xk
9F	jkQ	BRCR	BRANCH ON CONDITION REGISTER	j, k, LABEL
14	jk	LBSET	TEST AND SET BIT	Xk, Aj, X0
B4	jkQ	CMPXA	COMPARE SWAP	Xk, Aj, X0, Q
16	jk	TPAGE	TEST AND SET PAGE	Xk, Aj
17	jk	LPAGE	LOAD PAGE TABLE INDEX	Xk, Xj, X1
0E	jk	CPYSX	COPY FROM STATE REGISTER	Xk, Xj
0F	jk	CPYXS	COPY TO STATE REGISTER	
08	jk	CPYTX	COPY FREE RUNNING COUNTER	
05	jk	PURGE	PURGE BUFFER	Xj, k
B1	KEYPOINT	KEYPOINT	RESERVED	j, Xk, Q
C0–C7	SjkiD	EXECUTE,S (S=0-7)	EXECUTE ALGORITHM	j, k, i, D
00	jk	HALT	PROGRAM ERROR	(BLANK)
01	jk	SYNC	SCOPE LOOP SYNC	(BLANK)

VIRTUAL STATE GENERAL INSTRUCTIONS (Sheet 1 OF 3)

OP CODE	FORMAT	MNEMONIC	NAME	ADDRESS FIELD
A2	jkiD	LXI	LOAD WORD INDEXED	Xk, Aj, Xi, D
82	jkQ	LX	LOAD WORD	Xk, Aj, Q
A3	jkiD	SXI	STORE WORD INDEXED	Xk, Aj, Xi, D
83	jkQ	SX	STORE WORD	Xk, Aj, Q
D0–D7	SjkiD	LBYTS,S (S=1-8)	LOAD BYTES, IMMEDIATE	Xk, Aj, Xi, D
D8–DF	SjkiD	SBYTS,S (S=1-8)	STORE BYTES, IMMEDIATE	↓ ↓ ↓ ↓
A4	jkiD	LBYT,X0	LOAD BYTES	
A5	jkiD	SBYT,X0	STORE BYTES	
88	jkQ	LBIT	LOAD BIT	Xk, Aj, Q, X0
89	jkQ	SBIT	STORE BIT	↓ ↓ ↓ ↓
A0	jkiD	LAI	LOAD ADDRESS, INDEXED	Ak, Aj, Xi, D
84	jkQ	LA	LOAD ADDRESS	Ak, Aj, Q
A1	jkiD	SAI	STORE ADDRESS, INDEXED	Ak, Aj, Xi, D
85	jkQ	SA	STORE ADDRESS	Ak, Aj, Q
80	jkQ	LMULT	LOAD MULTIPLE	Xk, Aj, Q
81	jkQ	SMULT	STORE MULTIPLE	↓ ↓ ↓
24	jk	ADDX	INTEGER SUM	Xk, Xj
25	jk	SUBX	INTEGER DIFFERENCE	↓ ↓
26	jk	MULX	INTEGER PRODUCT	
27	jk	DIVX	INTEGER QUOTIENT	
2D	jk	CMPX	INTEGER COMPARE*	X1, Xj, Xk
8B	jkQ	ADDXQ	INTEGER SUM SIGNED IMMEDIATE	Xk, Xj, Q
10	jk	INCX	INTEGER SUM IMMEDIATE	Xk, j
11	jk	DECX	INTEGER DIFFERENCE IMMEDIATE	↓ ↓
B2	jkQ	MULXQ	INTEGER PRODUCT SIGNED IMMEDIATE	Xk, Xj, Q

* COMPARE RESULTS — BITS 32, 33

Xj = Xk	00	
Xj > Xk	01	
INDEFINITE	10	(FP ONLY)
Xj < Xk	11	

VIRTUAL STATE GENERAL INSTRUCTIONS (Sheet 2 OF 3)

OP CODE	FORMAT	MNEMONIC	NAME	ADDRESS FIELD
94	jkQ	BRXEQ	BRANCH ON EQUAL	Xj, Xk, LABEL
95	jkQ	BRXNE	BRANCH ON NOT EQUAL	
96	jkQ	BRXGT	BRANCH ON GREATER THAN	
97	jkQ	BRXGE	BRANCH ON GREATER THAN OR EQUAL	
9C	jkQ	BRINC	BRANCH AND INCREMENT	
20	jk	ADDR	HALF WORD INTEGER SUM	Xk, Xj
21	jk	SUBR	HALF WORD INTEGER DIFFERENCE	
22	jk	MULR	HALF WORD INTEGER PRODUCT	
23	jk	DIVR	HALF WORD INTEGER QUOTIENT	
2C	jk	CMPR	HALF WORD INTEGER COMPARE	X1, Xj, Xk
8A	jkQ	ADDRQ	HALF WORD INTEGER SUM SIGNED IMMEDIATE	Xk, Xj, Q
28	jk	INCR	HALF WORD INTEGER SUM IMMEDIATE	Xk, j
29	jk	DECR	HALF WORD INTEGER DIFFERENCE IMMEDIATE	
8C	jkQ	MULRQ	HALF WORD INTEGER PRODUCT SIGNED IMMEDIATE	Xk, Xj, Q
90	jkQ	BRREQ	BRANCH ON HALF WORD EQUAL	Xj, Xk, LABEL
91	jkQ	BRRNE	BRANCH ON HALF WORD NOT EQUAL	
92	jkQ	BRRGT	BRANCH ON HALF WORD GREATER THAN	
93	jkQ	BRRGE	BRANCH ON HALF WORD GREATER THAN OR EQUAL	
0D	jk	CPYXX	COPY FULL WORD	Xk, Xj
0B	jk	CPYAX	COPY ADDRESS A TO X	Xk, Aj
09	jk	CPYAA	COPY ADDRESS A TO A	Ak, Aj
0A	jk	CPYXA	COPY ADDRESS X TO A	Ak, Xj
0C	jk	CPYRR	COPY HALF WORD	Xk, Xj
8E	jkQ	ADDAQ	ADDRESS INCREMENT SIGNED IMMEDIATE	Ak, Aj, Q
2A	jk	ADDAX	ADDRESS INCREMENT, INDEXED	Ak, Xj
8F	jkQ	ADDPXQ	ADDRESS RELATIVE	Ak, Xj, Q

VIRTUAL STATE GENERAL INSTRUCTIONS (Sheet 3 OF 3)

OP CODE	FORMAT	MNEMONIC	NAME	ADDRESS FIELD
3D	jk	ENTP	ENTER IMMEDIATE POSITIVE	Xk, j
3E	jk	ENTN	ENTER IMMEDIATE NEGATIVE	
8D	jkQ	ENTE	ENTER SIGNED IMMEDIATE	Xk, Q
3F	jk	ENTL	ENTER X0 IMMEDIATE LOGICAL	X0, jk
1F	jk	ENTZ	ENTER ZEROS	Xk
		ENTO	ENTER ONES	Xk
		ENTS	ENTER SIGNS	Xk
39	jk	ENTX	ENTER X1 IMMEDIATE LOGICAL	X1, jk
87	jkQ	ENTC	ENTER X1 SIGNED IMMEDIATE	X1, jkQ
B3	jkQ	ENTA	ENTER X0 SIGNED IMMEDIATE	X0, jkQ
A8	jkiD	SHFC	SHIFT CIRCULAR	Xk, Xj, Xi, D
A9	jkiD	SHFX	SHIFT FULL WORD	
AA	jkiD	SHFR	SHIFT HALF WORD	
18	jk	IORX	LOGICAL SUM	Xk, Xj
19	jk	XORX	LOGICAL DIFFERENCE	
1A	jk	ANDX	LOGICAL PRODUCT	
1B	jk	NOTX	LOGICAL COMPLEMENT	
1C	jk	INHX	LOGICAL INHIBIT	
AC	jkiD	ISOM	ISOLATE BIT MASK	Xk, Xi, D
AD	jkiD	ISOB	ISOLATE	Xk, Xj, Xi, D
AE	jkiD	INSB	INSERT	
1E	jk	MARK	MARK TO BOOLEAN	Xk, X1, j
9D	jkQ	BRSEG	BRANCH ON SEGMENTS UNEQUAL	X1, Aj, Ak, LABEL
2E	jk	BRREL	BRANCH RELATIVE	Xk
2F	jk	BRDIR	INTER-SEGMENT BRANCH	Aj, Xk

VIRTUAL STATE FLOATING-POINT INSTRUCTIONS

OP CODE	FORMAT	MNEMONIC	NAME	ADDRESS FIELD
3A	jk	CNIF	CONVERT FROM INTEGER TO FLOATING POINT	Xk, Xj
3B	jk	CNFI	CONVERT FROM FLOATING POINT TO INTEGER	
30	jk	ADDF	SINGLE PRECISION FLOATING POINT SUM	Xk, Xj
31	jk	SUBF	SINGLE PRECISION FLOATING POINT DIFFERENCE	
32	jk	MULF	SINGLE PRECISION FLOATING POINT PRODUCT	
33	jk	DIVF	SINGLE PRECISION FLOATING POINT QUOTIENT	
98	jkQ	BRFEQ	BRANCH ON EQUAL	Xj, Xk, LABEL
99	jkQ	BRFNE	BRANCH ON NOT EQUAL	
9A	jkQ	BRFGT	BRANCH ON GREATER THAN	
9B	jkQ	BRFGE	BRANCH ON GREATER THAN OR EQUAL	
9E	jkQ	BR ---	FP BRANCH ON EXCEPTION	Xk, LABEL
3C	jk	CMPF	FLOATING POINT COMPARE *	X1, Xj, Xk
34	jk	ADDD	DOUBLE PRECISION FLOATING POINT SUM	XXk, XXj
35	jk	SUBD	DOUBLE PRECISION FLOATING POINT DIFFERENCE	
36	jk	MULD	DOUBLE PRECISION FLOATING POINT PRODUCT	
37	jk	DIVD	DOUBLE PRECISION FLOATING POINT QUOTIENT	

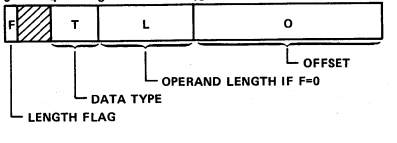
* COMPARE RESULTS — BITS 32, 33

Xj = Xk	00	
Xj > Xk	01	
INDEFINITE	10	(FP ONLY)
Xj < Xk	11	

VIRTUAL STATE BDP INSTRUCTIONS

OP CODE	FORMAT	MNEMONIC	NAME	ADDRESS FIELD
70	jk(2)	ADDN,Aj,X0	DECIMAL SUM	Ak, X1
71	jk(2)	SUBN,Aj,X0	DECIMAL DIFFERENCE	↓ ↓
72	jk(2)	MULN,Aj,X0	DECIMAL PRODUCT	
73	jk(2)	DIVN,Aj,X0	DECIMAL QUOTIENT	
E4	jkiD(2)	SCLN,Aj,X0	DECIMAL SCALE	Ak, X1, Xi, D
E5	jkiD(2)	SCLR,Aj,X0	DECIMAL SCALE ROUNDED	↓ ↓ ↓ ↓
74	jk(2)	CMPN,Aj,X0	DECIMAL COMPARE	Ak, X1
75	jk(2)	MOVN,Aj,X0	DECIMAL MOVE	↓ ↓
77	jk(2)	CMPB,Aj,X0	BYTE COMPARE	Ak, X1
E9	jkiD(2)	CMPC,Aj,X0	BYTE COMPARE COLLATED	Ak, X1, Ai, D
F3	jkiD(1)	SCNB,X0	BYTE SCAN WHILE NON-MEMBER	↓ ↓ ↓ ↓
EB	jkiD(2)	TRANB,Aj,X0	BYTE TRANSLATE	
76	jk(2)	MOVB,Aj,X0	MOVE BYTES	Ak, X1
ED	jkiD(2)	EDIT,Aj,X0	EDIT	Ak, X1, Ai, D
F9	jkiD(1)	MOVI,Xi,D	MOVE IMMEDIATE DATA	Ak, X1, j
FA	jkiD(1)	CMPI,Xi,D	COMPARE IMMEDIATE DATA	↓ ↓ ↓
FB	jkiD(1)	ADDI,Xi,D	ADD IMMEDIATE DATA	

BDP DESCRIPTOR



VIRTUAL STATE VECTOR INSTRUCTIONS (Sheet 1 OF 2)

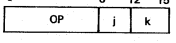
REF NO.	OP CODE	FORMAT	MNEMONIC	NAME	ADDRESS FIELD
172	44	jkiD	ADDXV	INTEGER VECTOR SUM	Ak, Aj, Ai, D or Ak, Xj, Ai, D
173	45	jkiD	SUBXV	INTEGER VECTOR DIFFERENCE	
176	50	jkiD	CMPEQV	INTEGER VECTOR COMPARE – EQUAL	
177	51	jkiD	CMPLEV	INTEGER VECTOR COMPARE – LESS	
178	52	jkiD	CMPGEV	INTEGER VECTOR COMPARE – GREATER, EQUAL	
179	53	jkiD	CMPNEV	INTEGER VECTOR COMPARE – NOT EQUAL	
180	4D	jkiD	SHFV	SHIFT VECTOR CIRCULAR	
181	48	jkiD	IORV	LOGICAL VECTOR SUM	
182	49	jkiD	XORV	LOGICAL VECTOR DIFFERENCE	
183	4A	jkiD	ANDV	LOGICAL VECTOR PRODUCT	

VIRTUAL STATE VECTOR INSTRUCTIONS (Sheet 2 OF 2)

REF NO.	OP CODE	FORMAT	MNEMONIC	NAME	ADDRESS FIELD
184 185	4B 4C	jkiD jkiD	CNIFV CNFIV	CONVERT VECTOR FROM INTEGER TO FLOATING POINT CONVERT VECTOR FROM FLOATING POINT TO INTEGER	Ak, Aj, D or Ak, Xj, D
186 187 188 189	40 41 42 43	jkiD jkiD jkiD jkiD	ADDFV SUBFV MULFV DIVFV	FLOATING POINT VECTOR SUM FLOATING POINT VECTOR DIFFERENCE FLOATING POINT VECTOR PRODUCT FLOATING POINT VECTOR QUOTIENT	Ak, Aj, Ai, D or Ak, Xi, Ai, D
190	57	jkiD	SUMFV	FLOATING POINT VECTOR SUMMATION	Xk, Ai, D
191	54	jkiD	MRGV	MERGE VECTOR	Ak, Aj, Ai, D or Ak, Xj, Ai, D
192 193	55 56	jkiD jkiD	GTHV SCTV	GATHER VECTOR SCATTER VECTOR	Ak, Aj, Xi, D or Ak, Xj, Xi, D

INTERRUPT CONDITIONS - OX

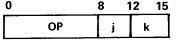
FORMAT: jk



OP	MNEMONIC	INSTRUCTION NAME	ADDRESS	INTERRUPT CONDITION: MCR: ADDR. SPEC. INST. SPEC. ERR. DET. UNCORR. ERR. (48)	MCR: PT SEARCH/ACCESS VIOLATION (51)	MCR: SYSTEM CALL/IND EMA SPEC. ERR. (52)	UCR: OUTPV. CALL/IN O PRIV. INST. RET. (54)	UCR: UNIMP. INST./ILL. INTERRING POP (55)	UCR: DIVIDE FAULT RESERVED FLAG CRIT. FAULT (57)	UCR: EXP. UNDERFLOW EXP. OVERFLOW ARITH. DEBUG (58)	UCR: INV. BDP DATA ARITH. LOSS/NITE FP LOSS OF SIG. FP INDEFINITE SIG. (59)	60	61	62	63
00	HALT	PROGRAM ERROR	(BLANK)	X	X										
01	SYNC	SCOPE LOOP SYNC	(BLANK)	X											
02	EXCHANGE	EXCHANGE	(BLANK)	X		X	X								
03	INTRUPT	PROCESSOR INTERRUPT	Xk	X				X							
04	RETURN	RETURN	(BLANK)	X	X	X X X		X X		X	X				
05	PURGE	PURGE BUFFER	Xj, k	X	X			X X							
06	POP	POP	(BLANK)	X	X	X X X		X		X X	X				
07		PURGE SFSA PUSHDOWN	(BLANK)	X					X						
08	CPYTX	COPY FREE RUNNING COUNTER	Xk, Xj	X											
09	CPYAA	COPY ADDRESS A TO A	Ak, Aj	X											
0A	CPYXA	COPY ADDRESS X TO A	Ak, Xj	X											
0B	CPYAX	COPY ADDRESS A TO X	Xk, Aj	X											
0C	CPYRR	COPY HALF WORD	Xk, Xj	X											
0D	CPYXX	COPY FULL WORD	Xk, Xj	X											
0E	CPYSX	COPY FROM STATE REGISTER	Xk, Xj	X											
0F	CPYXS	COPY TO STATE REGISTER	Xk, Xj	X	X			X							

INTERRUPT CONDITIONS - 1X

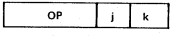
FORMAT: jk



OP	MNEMONIC	INSTRUCTION NAME	ADDRESS	INTERRUPT CONDITION: MCR: DET. UNCORR. ERR. 48	MCR: ADDR. SPEC. INST. SPEC. ERR. 51	MCR: SYSTEM CALL/PT ENV. SPEC. ACCESS VIOLATION 52	MCR: SEARCH NO FIND ERR. 54	MCR: OUT. CALL/RN INV. SEG/RN O 55	MCR: UNIMPLD. INST./PRIV. INST. FAULT 57	MCR: RET. CRIT. FRAME INTERRING 58	UCR: DIVIDE FAULT 60	UCR: FLAG PRESENT/ROP FAULT 61	UCR: EXP. OVERFLOW ARITH. DEBUG 48	UCR: EXP. UNDERFLOW ARITH. OVERFLOW 49	UCR: INV. BDP DATA ARITH. LOSS OF SIG. FP INDEFINITE FP LOSS OF SIG. 55	52	53	54	56	57	58	59	60	63	62	61	60
10	INCX	INTEGER SUM IMMEDIATE	Xk, j	X																X						
11	DECX	INTEGER DIFFERENCE IMM.	Xk, j	X																X						
12				X										X												
13				X										X												
14	LBSET	TEST AND SET BIT	Xk, Aj, X0	X		X	X	X	X		X									X						
15				X										X												
16	TPAGE	TEST AND SET PAGE	Xk, Aj	X		X					X															
17	LPAGE	LOAD PAGE TABLE INDEX	Xk, Xj, X1	X		X								X												
18	IORX	LOGICAL SUM	Xk, Xj	X																						
19	XORX	LOGICAL DIFFERENCE	Xk, Xj	X																						
1A	ANDX	LOGICAL PRODUCT	Xk, Xj	X																						
1B	NOTX	LOGICAL COMPLEMENT	Xk, Xj	X																						
1C	INHX	LOGICAL INHIBIT	Xk, Xj	X																						
1D				X										X												
1E	MARK	MARK TO BOOLEAN	Xk, X1, j	X																						
1F	ENTZ/O/S	ENTER ZEROS/ONES/SIGNS	Xk	X																						

INTERRUPT CONDITIONS - 2X

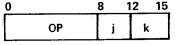
FORMAT: jk



OP	MNEMONIC	INSTRUCTION NAME	ADDRESS	INTERRUPT CONDITION: MCR: DET. UNCORR. ERR. (48)	MCR: ADDR. SPEC. ERR. INST. SPEC. ERR. (51)	MCR: PT SEARCH WORD ERR. ACCESS VIOLATION INV. SPEC. ERR. (52)	MCR: SYSTEM CALL/FIND OUT-CALLING INV. ASSIGN (54)	MCR: UNIMPL'D INST. FAULT PRIV. INST. RET. O. (55)	MCR: RESERVED FLAG CRIT. FRAMING INTERFRAING POP (57)	UCR: DIVIDE FAULT (58)	UCR: EXP. UNDERFLOW EXP. OVERFLOW ARITH. OVERFLOW DEBUG (59)	UCR: FP INDEFINITE FP LOSS OF SIG. (60)	UCR: INV. BDP DATA ARITH. LOSS OF SIG. (61)	(62)	(63)
20	ADDR	HALF WORD INTEGER SUM	Xk, Xj	X							X				
21	SUBR	HALF WORD INTEGER DIFF.	Xk, Xj	X							X				
22	MULR	HALF WORD INTEGER PROD.	Xk, Xj	X							X				
23	DIVR	HALF WORD INTEGER QUOT.	Xk, Xj	X							X				
24	ADDX	INTEGER SUM	Xk, Xj	X							X				
25	SUBX	INTEGER DIFFERENCE	Xk, Xj	X							X				
26	MULX	INTEGER PRODUCT	Xk, Xj	X							X				
27	DIVX	INTEGER QUOTIENT	Xk, Xj	X							X				
28	INCR	HALF WORD INT. SUM IMM.	Xk, j	X							X				
29	DECR	HALF WORD INT. DIFF. IMM.	Xk, j	X							X				
2A	ADDAX	ADDRESS INCREMENT INDEXED	Ak, Xj	X											
2B				X					X						
2C	CMPR	HALF WORD INT. COMPARE	X1, Xj, Xk	X											
2D	CMPX	INTEGER COMPARE	X1, Xj, Xk	X											
2E	BRREL	BRANCH RELATIVE	Xk	X		X			X			X			
2F	BRDIR	INTER-SEGMENT BRANCH	Aj, Xk	X		X	X		X	X		X			

INTERRUPT CONDITIONS - 3X

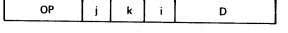
FORMAT: jk



OP	MNEMONIC	INSTRUCTION NAME	ADDRESS	INTERRUPT CONDITION: MCR: DET. UNCORR. ERR. (48)	MCR: ADDR. SPEC. INST. SPEC. ERR. (51)	MCR: SYSTEM PT BNDRY SPEC. ACCESS VIOLATION (52)	UCR: OUT. CALL/RN INV. SEG. WO FIND (54)	UCR: UNIMPL.ID. PRIV. INST. RET. (57)	UCR: INSTR. FAULT CRIT. FRAME INTER RING POP (60)	UCR: DIVIDE FAULT RESERVED FLAG FRAME POP (48)	UCR: EXP. UNDERFLOW ARITH. DEBUG (49)	UCR: EXP. UNDERFLOW OVERFLOW (52)	UCR: INV. BDP DATA FTH. LOSS OF SIG. FP LOSS OF SIG. (53)	UCR: INV. BDP DATA ARITH. LOSS OF SIG. FP LOSS OF SIG. (54)	(55)	(56)	(57)	(58)	(59)	(60)	(61)	(62)	(63)
30	ADDF	SINGLE PRECISION FP SUM	Xk, Xj	X															X	X	X	X	
31	SUBF	SINGLE PRECISION FP DIFF.	Xk, Xj	X															X	X	X	X	
32	MULF	SINGLE PRECISION FP PROD.	Xk, Xj	X															X	X		X	
33	DIVF	SINGLE PRECISION FP QUOT.	Xk, Xj	X												X			X	X		X	
34	ADDD	DOUBLE PRECISION FP SUM	XXk, XXj	X															X	X	X	X	
35	SUBD	DOUBLE PRECISION FP DIFF.	XXk, XXj	X															X	X	X	X	
36	MULD	DOUBLE PRECISION FP PROD.	XXk, XXj	X															X	X		X	
37	DIVD	DOUBLE PRECISION FP QUOT.	XXk, XXj	X												X			X	X		X	
38				X						X													
39	ENTX	ENTER X1 IMM. LOGICAL	X1, jk	X																			
3A	CNIF	CONVERT FROM INT. TO FP	Xk, Xj	X																			
3B	CNFI	CONVERT FROM FP TO INT.	Xk, Xj	X																		X	X
3C	CMPF	FLOATING POINT COMPARE	X1, Xj, Xk	X																		X	
3D	ENTP	ENTER IMMEDIATE POSITIVE	Xk, j	X																			
3E	ENTN	ENTER IMMEDIATE NEGATIVE	Xk, j	X																			
3F	ENTL	ENTER X0 IMM. LOGICAL	X0, jk	X																			

INTERRUPT CONDITIONS — 4X

FORMAT: jkiD



OP	MNEMONIC	INSTRUCTION NAME	ADDRESS	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63
40	ADDFV	FP VECTOR SUM	Ak, Aj, Ai, D or Ak, Xj, Ai, D	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
41	SUBFV	FP VECTOR DIFFERENCE		X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
42	MULFV	FP VECTOR PRODUCT		X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
43	DIVFV	FP VECTOR QUOTIENT		X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
44	ADDXV	INTEGER VECTOR SUM	Ak, Aj, Ai, D or Ak, Xj, Ai, D	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
45	SUBXV	INTEGER VECTOR DIFFERENCE		X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
46																			
47																			
48	IORV	LOGICAL VECTOR SUM	Ak, Aj, Ai, D or Ak, Xj, Ai, D	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
49	XORV	LOGICAL VECTOR DIFFERENCE		X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
4A	ANDV	LOGICAL VECTOR PRODUCT		X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
4B	CNIFV	CONVERT VECTOR FROM INT. TO FP	Ak, Aj, D or Ak, Xj, D	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
4C	CNFIV	CONVERT VECTOR FROM FP TO INT.	Ak, Aj, D or Ak, Xj, D	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
4D	SHFV	SHIFT VECTOR CIRCULAR	Ak, Aj, Ai, D or Ak, Xj, Ai, D	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
4E																			
4F																			

INTERRUPT CONDITION

MCR

- DET. UNCORR. ERR.
- ADDR. SPEC. INST. SPEC. ERR.
- PT SEARCH AND ENV. SPEC. ERR.
- SYSTEM CALL-IN ACCESS VIOLATION
- OUTPUT CALL-IN INV. SEG/RN
- PRIV. INST. FAULT
- UNIMPLD. INST. FAULT
- CRIT. INTERFERING POP
- DIVIDE FAULT RESERVED FLAG
- EXP. OVERFLOW ARITH OVERFLOW
- EXP. UNDERFLOW FP LOSS OF SIG.
- INV. BDP DATA ARITH. LOSS OF SIG. FP INDEFINITE

UCR

INTERRUPT CONDITIONS — 5X, 6X

FORMAT: jkiD

0–7	8–11	12–15	16–19	20–31
OP	j	k	i	D

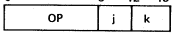
INTERRUPT CONDITION

OP	MNEMONIC	INSTRUCTION NAME	ADDRESS	MCR 48 DET. UNCORR. ERR.	MCR 51 INST. SPEC. ERR.	MCR 52 ADDR. SPEC. ERR.	MCR 54 ACCESS VIOLATION	MCR 55 ENV. SPEC. ERR.	MCR 57 PT SEARCH WO FIND	MCR 58 SYSTEM CALL	MCR 60 INV. SEG/RN O	MCR 61 OUT. CALL/IN. RET.	UCR 48 PRIV. INST. FAULT	UCR 49 UNIMPL'D. INST.	UCR 52 INTER-RING POP	UCR 53 CRIT. FRAME FLAG	UCR 54 RESERVED	UCR 55 DIVIDE FAULT	UCR 56 DEBUG	UCR 57 ARITH. OVERFLOW	UCR 58 EXP. OVERFLOW	UCR 59 EXP. UNDERFLOW	UCR 60 FP LOSS OF SIG.	UCR 61 FP INDEFINITE	UCR 62 ARITH. LOSS OF SIG.	UCR 63 INV. BDP DATA
50	CMPEQV	INTEGER VECTOR COMPARE =	Ak, Aj, Ai, D or Ak, Xj, Ai, D	X	X	X	X	X	X		X								X							
51	CMPLTV	INTEGER VECTOR COMPARE <		X	X	X	X	X	X		X								X							
52	CMPGEV	INTEGER VECTOR COMPARE ≥		X	X	X	X	X	X		X								X							
53	CMPNEV	INTEGER VECTOR COMPARE ≠		X	X	X	X	X	X		X								X							
54	MRGV	MERGE VECTOR	Ak, Aj, Ai, D or Ak, Xj, Ai, D	X	X	X	X	X	X		X								X							
55	GTHV	GATHER VECTOR	Ak, Aj, Xi, D or	X	X	X	X	X	X		X								X							
56	SCTV	SCATTER VECTOR	Ak, Xj, Xi, D	X	X	X	X	X	X		X								X							
57	SUMFV	FP VECTOR SUMMATION	Xk, Ai, D	X	X	X	X	X	X										X		X	X	X	X		
58	TPSFV	FP VECTOR TRIAD, * +	Ak, Aj, Ai, X0, D or Ak, Xj, Ai, X0, D	X	X	X	X	X	X		X															
59	TPDFV	FP VECTOR TRIAD, * -		X	X	X	X	X	X		X															
5A	TSPFV	FP VECTOR TRIAD, + *		X	X	X	X	X	X		X															
5B	TDPFV	FP VECTOR TRIAD, - *		X	X	X	X	X	X		X															
5C	SUMPFV	FP VECTOR DOT PRODUCT	Xk, Aj, Ai, D or Xk, Xj, Ai, D	X	X	X	X	X	X		X															
5D	GTHJIV	GATHER VECTOR, INDEX	Ak, Aj, Ai, D or	X	X	X	X	X	X		X															
5E	SCTIV	SCATTER VECTOR, INDEX	Ak, Xj, Ai, D	X	X	X	X	X	X		X															
5F														X												
60 ↓ 6F														X ↓ X												

1573

INTERRUPT CONDITIONS - 7X

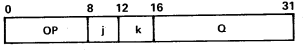
FORMAT: jk(2)



OP	MNEMONIC	INSTRUCTION NAME	ADDRESS	INTERRUPT CONDITION: MCR 48 DET. UNCORR. ERR.	MCR 51 ADDR. SPEC. INST. SPEC. ERR.	MCR 52 PT SEARCH EV. SPEC. ACCESS VIOLATION	MCR 54 SYSTEM CALL IND OUT. CALL/INV. INV. USE/RAN O	MCR 55 WO ERR.	MCR 57	MCR 58 UNIMPL. I.D. PRIV. INST. RET.	MCR 60 FAULT INST. FAULT	MCR 61	UCR 48 DIVIDE FAULT RESERVED FLAG CRIT. INTER-FRAME POP	UCR 49	UCR 52 EXP. EXPONENT ARITH. DEBUG	UCR 53	UCR 54	UCR 55 EXP. UNDERFLOW EXP. OVERFLOW	UCR 56	UCR 57	UCR 58	UCR 59 INV. BDP DATA ARITH. LOSS OF SIG. FP INDEFINITE FP LOSS OF SIG.	UCR 60	UCR 61	UCR 62	UCR 63
70	ADDN, Aj, X0	DECIMAL SUM	Ak, X1	X	X	X	X		X										X	X					X	
71	SUBN, Aj, X0	DECIMAL DIFFERENCE	Ak, X1	X	X	X	X		X										X	X					X	
72	MULN, Aj, X0	DECIMAL PRODUCT	Ak, X1	X	X	X	X		X										X	X					X	
73	DIVN, Aj, X0	DECIMAL QUOTIENT	Ak, X1	X	X	X	X		X									X	X	X					X	
74	CMPN, Aj, X0	DECIMAL COMPARE	Ak, X1	X	X	X	X		X										X						X	
75	MOVN, Aj, X0	DECIMAL MOVE	Ak, X1	X	X	X	X		X										X					X	X	
76	MOVB, Aj, X0	MOVE BYTES	Ak, X1	X	X	X	X		X										X							
77	CMPB, Aj, X0	BYTE COMPARE	Ak, X1	X	X	X	X		X										X							
78				X										X												
79				X										X												
7A				X										X												
7B				X										X												
7C				X										X												
7D				X										X												
7E				X										X												
7F				X										X												

INTERRUPT CONDITIONS - 8X

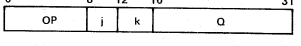
FORMAT: jkQ



OP	MNEMONIC	INSTRUCTION NAME	ADDRESS	INTERRUPT CONDITION: MCR: ADDR. SPEC. ERR. INST. SPEC. ERR. DET. UNCORR. ERR. (48)	MCR: PT ENV. SPEC. ERR. SEARCH ACCESS VIOLATION (51)	MCR: SYSTEM CALL INV. C-SEG/RN (52)	MCR: INV. C-SEG/RN FIND W/ ERR IND (54)	MCR: OUT. C-SEG/RN RET. (55)	MCR: UNIMPL'D. INST. PRIV. INST. (57)	MCR: CRIT. INST. INTERRING FAULT (58)	UCR: DIVIDE FAULT RESERVED LAG FRAME POP (60)	UCR: EXP. UNDERFLOW ARITH. OVERFLOW DEBIG (61)	UCR: EXP. UNDERFLOW ARITH. OVERFLOW (48)	UCR: INV. BDP DATA ARITH. LOSS OF SIG. FP INDEFINITE FP LOSS OF SIG. (49)	UCR: (52)	UCR: (53)	UCR: (54)	UCR: (55)	UCR: (56)	UCR: (57)	UCR: (58)	UCR: (59)	UCR: (60)	UCR: (61)	UCR: (62)	UCR: (63)
80	LMULT	LOAD MULTIPLE	Xk, Aj, Q	X	X	X	X		X											X						
81	SMULT	STORE MULTIPLE	Xk, Aj, Q	X	X	X	X		X											X						
82	LX	LOAD WORD	Xk, Aj, Q	X	X	X	X		X											X						
83	SX	STORE WORD	Xk, Aj, Q	X	X	X	X		X											X						
84	LA	LOAD ADDRESS	Ak, Aj, Q	X	X	X	X		X											X						
85	SA	STORE ADDRESS	Ak, Aj, Q	X	X	X	X		X											X						
86	LBYTP, j	LOAD BYTES, RELATIVE	Xk, Q	X	X	X	X		X											X						
87	ENTC	ENTER X1 SIGNED IMM.	X1, jkQ	X																						
88	LBIT	LOAD BIT	Xk, Aj, Q, X0	X	X	X	X		X											X						
89	SBIT	STORE BIT	Xk, Aj, Q, X0	X	X	X	X		X											X						
8A	ADDRQ	HALF WD INTEGER SUM SIGNED IMM.	Xk, Xj, Q	X																	X					
8B	ADDXQ	INTEGER SUM SIGNED IMM.	Xk, Xj, Q	X																	X					
8C	MULRQ	HALF WD INTEGER PROD. SIGNED IMM.	Xk, Xj, Q	X																	X					
8D	ENTE	ENTER Xk SIGNED IMMEDIATE	Xk, Q	X																						
8E	ADDAQ	ADDRESS INCR. SIGNED IMM.	Ak, Aj, Q	X																						
8F	ADDPXQ	ADDRESS RELATIVE	Ak, Xj, Q	X																						

INTERRUPT CONDITIONS - 9X

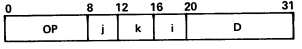
FORMAT: jkQ



OP	MNEMONIC	INSTRUCTION NAME	ADDRESS	INTERRUPT CONDITION: MCR: DET. UNCORR. ERR. (48)	MCR: ADDR. SPEC. ERR. INST. SPEC. ERR. (51)	MCR: PT SEARCH ENV. SPEC. ERR. ACCESS VIOLATION (52)	UCR: SYSTEM CALL OUT. CALL/N RET. INV. SEG/RN O (54)	UCR: UNIMPL. INST. PRIV. INST. (57)	UCR: CRIT. INSTR. INTERRUPT (60)	UCR: DIVIDE FAULT RESERVED FLAG FRAME POP (48)	UCR: EXP. UNDERFLOW EXP. OVERFLOW ARITH. OVERFLOW DEBUG (49)	UCR: INV. BDP DATA ARITH. LOSS OF E FP INDEFINITE FP LOSS OF SIG. (52)	UCR: (53)	UCR: (54)	UCR: (55)	UCR: (56)	UCR: (57)	UCR: (58)	UCR: (59)	UCR: (60)	UCR: (61)	UCR: (62)	UCR: (63)
90	BRREQ	BRANCH ON HALF WORD =	Xj, Xk, LABEL	X	X	X										X							
91	BRRNE	BRANCH ON HALF WORD ≠	Xj, Xk, LABEL	X	X	X										X							
92	BRRGT	BRANCH ON HALF WORD >	Xj, Xk, LABEL	X	X	X										X							
93	BRRGE	BRANCH ON HALF WORD ≥	Xj, Xk, LABEL	X	X	X										X							
94	BRXEQ	BRANCH ON =	Xj, Xk, LABEL	X	X	X										X							
95	BRXNE	BRANCH ON ≠	Xj, Xk, LABEL	X	X	X										X							
96	BRXGT	BRANCH ON >	Xj, Xk, LABEL	X	X	X										X							
97	BRXGE	BRANCH ON ≥	Xj, Xk, LABEL	X	X	X										X							
98	BRFEQ	FP BRANCH ON =	Xj, Xk, LABEL	X	X	X										X						X	
99	BRFNE	FP BRANCH ON ≠	Xj, Xk, LABEL	X	X	X										X						X	
9A	BRFGT	FP BRANCH ON >	Xj, Xk, LABEL	X	X	X										X						X	
9B	BRFGE	FP BRANCH ON ≥	Xj, Xk, LABEL	X	X	X										X						X	
9C	BRINC	BRANCH AND INCREMENT	Xj, Xk, LABEL	X	X	X										X							
9D	BRSEG	BRANCH ON SEGMENTS UNEQUAL	X1,Aj,Ak, LABEL	X	X	X										X							
9E	BR · · ·	FP BRANCH ON EXCEPTION	Xk, LABEL	X	X	X										X							
9F	BRCR	BRANCH ON CONDITION REG.	j, k, LABEL	X	X	X	X									X							

INTERRUPT CONDITIONS - AX

FORMAT: jkiD



OP	MNEMONIC	INSTRUCTION NAME	ADDRESS	INTERRUPT CONDITION: MCR: ADDR. SPEC. INST. SPEC. ERR. DET. UNCORR. ERR. (48)	MCR: SYSTEM CALL FIND PT SEARCH WORD ERR. ENV. SPEC. ACCESS VIOLATION (51-53)	MCR: OUT. SEG./RN O.T. INV. CALL/RN (54-55)	MCR: UNIMPL'D INST. PRIV. INST. RET. (57-58)	UCR: CRIT. FRAME FAULT INTERRING. POP (60-61)	UCR: DIVIDE FAULT RESERVED FLAG (49)	UCR: EXP. OVERFLOW EXP. UNDERFLOW ARITH. DEBUG (49)	UCR: FP LOSS OF SIG. ARITH. LOSS OF SIG. FP INDEFINITE (52-55)	UCR: INV. BDP DATA ARITH. OVERFLOW (56-59)	UCR: (60-63)
A0	LAI	LOAD ADDRESS INDEXED	Ak, Aj, Xi, D	X	X	X	X	X				X	
A1	SAI	STORE ADDRESS INDEXED	Ak, Aj, Xi, D	X	X	X	X	X				X	
A2	LXI	LOAD WORD INDEXED	Xk, Aj, Xi, D	X	X	X	X	X				X	
A3	SXI	STORE WORD INDEXED	Xk, Aj, Xi, D	X	X	X	X	X				X	
A4	LBYT, X0	LOAD BYTES	Xk, Aj, Xi, D	X	X	X	X	X				X	
A5	SBYT, X0	STORE BYTES	Xk, Aj, Xi, D	X	X	X	X	X				X	
A6				X					X				
A7	ADDAD	ADDRESS INCREMENT, MODULO	Ak, Aj, D, j	X									
A8	SHFC	SHIFT WORD CIRCULAR	Xk, Xj, Xi, D	X									
A9	SHFX	SHIFT WORD END-OFF	Xk, Xj, Xi, D	X									
AA	SHFR	SHIFT HALF WORD END-OFF	Xk, Xj, Xi, D	X									
AB				X					X				
AC	ISOM	ISOLATE BIT MASK	Xk, Xi, D	X	X								
AD	ISOB	ISOLATE	Xk, Xj, Xi, D	X	X								
AE	INSB	INSERT	Xk, Xj, Xi, D	X	X								
AF				X					X				

INTERRUPT CONDITIONS - BX

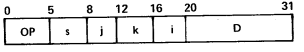
FORMAT: jkQ

0–7	8–11	12–15	16–31
OP	j	k	Q

OP	MNEMONIC	INSTRUCTION NAME	ADDRESS	MCR 48 DET. UNCORR. ERR.	MCR 51 INST. SPEC. ERR.	MCR 52 ADDR. SPEC. ERR.	MCR 54 ACCESS VIOLATION	MCR 55 ENV. SPEC. ERR.	MCR 57 PT SEARCH WO FIND	MCR 58 SYSTEM CALL	MCR 60 INV. SEG/RN O	MCR 61 OUT. CALL/IN. RET.	UCR 48 PRIV. INST. FAULT	UCR 49 UNIMPL'D. INST.	UCR 52 INTER-RING POP	UCR 53 CRIT. FRAME FLAG	UCR 54 RESERVED	UCR 55 DIVIDE FAULT	UCR 56 DEBUG	UCR 57 ARITH. OVERFLOW	UCR 58 EXP. OVERFLOW	UCR 59 EXP. UNDERFLOW	UCR 60 FP LOSS OF SIG.	UCR 61 FP INDEFINITE	UCR 62 ARITH. LOSS OF SIG.	UCR 63 INV. BDP DATA
B0	CALLREL	CALL RELATIVE	LABEL, Aj, Ak	X	X	X	X		X		X								X							
B1	KEYPOINT	KEYPOINT	j, Xk, Q	X		X	X		X		X															
B2	MULXQ	INTEGER PROD. SIGNED IMMEDIATE	Xk, Xj, Q	X																						
B3	ENTA	ENTER X0 SIGNED IMMEDIATE	X0, jkQ	X																X						
B4	CMPXA	COMPARE SWAP	Xk, Aj, X0, Q	X	X	X	X		X		X								X							
B5	CALLSEG	CALL INDIRECT	LABEL, Aj, Ak	X	X	X	X	X	X		X	X							X							
B6				X										X												
B7				X										X												
B8				X										X												
B9				X										X												
BA				X										X												
BB				X										X												
BC				X										X												
BD		(RESERVED OP CODE)		X										X												
BE		(RESERVED OP CODE)		X										X												
BF		(RESERVED OP CODE)		X										X												

INTERRUPT CONDITIONS - CX, DX

FORMAT: SjkiD

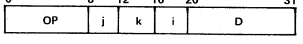


OP*	MNEMONIC	INSTRUCTION NAME	ADDRESS	INTERRUPT CONDITION: MCR: DET. UNCORR. ERR. (48)	MCR: ADDR. SPEC. ERR. INST. SPEC. ERR. (51 52)	MCR: PT SEARCH ENV. SPECIFICATION ACCESS VIOLATION (54)	MCR: SYSTEM CALL FIND OUT-CALL/RN O (55 57 58)	UCR: INV. SEG/RN (60)	UCR: UNIMPLD. INST. PRIV. INST. FAULT (61)	UCR: CRIT. INST. INTERRING (48)	UCR: DIVIDE FAULT RESERVED FLAG REPARING POP (49)	UCR: EXP. UNDERFLOW EXP. OVERFLOW ARITH. OVERFLOW DEBUG (52 53 54 55)	UCR: INV. BDP DATA OR SIG. ARITH. LOSS OF SIG. FP LOSS OF SIG. (56 57 58 59)	UCR: FP INDEFINITE (60)	UCR: (62 63)
C0 ↓ C7	EXECUTE, S	EXECUTE ALGORITHM 0 ↓ EXECUTE ALGORITHM 7	j, k, i, D	X											
C8 ↓ CF				X						X					
D0 ↓ D7	LBYTS, S	LOAD BYTES, IMMEDIATE	Xk, Aj, Xi, D	X	X	X	X		X				X		
D8 ↓ DF	SBYTS, S	STORE BYTES, IMMEDIATE	Xk, Aj, Xi, D	X	X	X	X		X				X		

* OP INCLUDES S

INTERRUPT CONDITIONS - EX

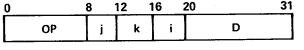
FORMAT: jkiD(2)



OP	MNEMONIC	INSTRUCTION NAME	ADDRESS	INTERRUPT CONDITION: MCR: DET. UNCORR. ERR. (48)	MCR: ADDR. SPEC. INST. SPEC. ERR. (52)	MCR: PT SEARCH INV. SPEC. ACCESS VIOLATION (55)	MCR: SYSTEM CALL FIND OUT. CALL/IN INV. SEG/IN O (57)	UCR: UNIMPL. ID. PRIV. INST. RET. (60)	UCR: INST. FAULT CRIT. INTERRING (61)	UCR: DIVIDE FAULT RESERVED FLAG FRAME POP (49)	UCR: EXP. UNDERFLOW EXP. OVERFLOW ARITH. DEBUG (52)	UCR: INV. BDP, DATA ARITH. LOSE OF SIG. FP INDEFINITE FP LOSS OF SIG. (56)	UCR: (57)	UCR: (58)	UCR: (59)	UCR: (60)	UCR: (61)	UCR: (62)	UCR: (63)
E0				X						X									
E1				X						X									
E2				X						X									
E3				X						X									
E4	SCLN, Aj, X0	DECIMAL SCALE	Ak, X1, Xi, D	X	X	X	X	X				X						X	X
E5	SCLR, Aj, X0	DECIMAL SCALE ROUNDED	Ak, X1, Xi, D	X	X	X	X	X				X						X	X
E6				X						X									
E7				X						X									
E8				X						X									
E9	CMPC, Aj, X0	BYTE COMPARE COLLATED	Ak, X1, Ai, D	X	X	X	X	X				X							
EA				X						X									
EB	TRANB, Aj, X0	BYTE TRANSLATE	Ak, X1, Ai, D	X	X	X	X	X				X							
EC				X						X									
ED	EDIT, Aj, X0	EDIT	Ak, X1, Ai, D	X	X	X	X	X				X							X
EE				X						X									
EF				X						X									

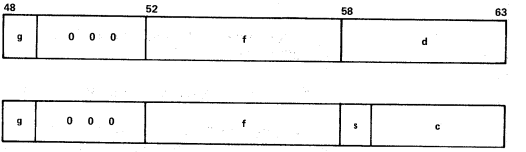
INTERRUPT CONDITIONS - FX

FORMAT: jkiD(1)

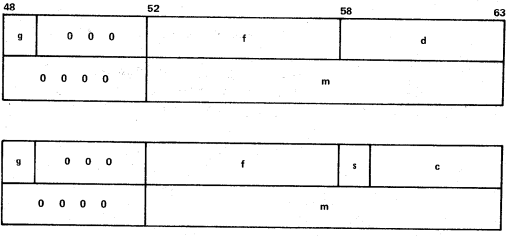


OP	MNEMONIC	INSTRUCTION NAME	ADDRESS	INTERRUPT CONDITION: MCR: ADDR. SPEC. INST. SPEC. ERR. DET. - UNCORR. ERR. (48)	MCR: SYSTEM SEARCH WORD FIND PT. ENV. SPEC. ERR. ACCESS VIOLATION (51 52)	MCR: OUT-INV.-CALL/JUN O'FLOW INV.-CALL/JUN O'FLOW (54 55 56 57 58)	UCR: UNWRT'D INST. FAULT PRIV INST. RET. (60)	UCR: CRIT. INSTR. INTERRING POP (48 49)	UCR: DIVIDE FAULT RESERVED FLAG FRAME FLAG (52)	UCR: EXP. OVERFLOW EXP. UNDERFLOW ARITH. OVERFLOW ARITH. DEBUG (53 54 55 56 57 58 59)	UCR: INV. BDP DATA ARITH. LOSS OF SIG. FP LOSS OF SIG. FP INDEFINITE (60 61 62 63)
F0				X				X			
F1				X				X			
F2				X				X			
F3	SCNB, X0	BYTE SCAN WHILE NON-MEMBER	Ak, X1, Ai, D	X	X X	X X	X			X	
F4				X				X		X	X
F5				X				X			
F6				X				X			
F7				X				X			
F8				X				X			
F9	MOVI, Xi, D	MOVE IMMEDIATE DATA	Ak, X1, j	X	X X	X X	X			X	X X
FA	CMPI, Xi, D	COMPARE IMMEDIATE DATA	Ak, X1, j	X	X X	X X	X			X	X
FB	ADDI, Xi, D	ADD IMMEDIATE DATA	Ak, X1, j	X	X X	X X	X			X X	X
FC				X				X			
FD				X				X			
FE				X				X			
FF				X				X			

PERIPHERAL PROCESSOR INSTRUCTION FORMATS (VIRTUAL STATE AND CYBER 170 STATE)



16-Bit Formats



32-Bit Formats

Table 2 lists and defines the PP instruction designators.

Table 2. PP Instruction Designators

Designator	Description
f	Specifies instruction operation code.
s	Specifies I/O instruction subcode.
c	Specifies channel number.
A	Refers to the A register (arithmetic register) or the content of the A register.
(A)	Refers to the content of the word at the CM address specified by the A register.
P	Refers to the P register or to the content of the P register (program address register).
R	Refers to the R register or to the content of the R register (relocation register).
(d)	Refers to the content of the word at the PP memory address specified by the d field (direct mode).
((d))	Refers to the content of the word at the PP memory address specified by the content of the word at the PP memory address specified by the d field (indirect mode).
m+(d)	Refers to the PP memory address specified by the m field indexed by the content of the word at the PP memory address specified by the d field.
(m+(d))	Refers to the content of the word at the PP memory address specified by the m field indexed by the content of the word at the PP memory address specified by the d field (memory mode).

PP INSTRUCTIONS (VIRTUAL STATE AND CYBER 170 STATE)
(Sheet 1 OF 3)

OP CODE	MNEMONIC	NAME	OP CODE	MNEMONIC	NAME
0000		PASS	1000d	RDSL d	CENTRAL READ AND SET LOCK
0001dm	LJM m,d	LONG JUMP	1001d	RDCL d	CENTRAL READ AND CLEAR LOCK
0002dm	RJM m,d	RETURN JUMP	1002	-	PASS
0003d	UJN d	UNCONDITIONAL JUMP	1003	-	PASS
0004d	ZJN d	ZERO JUMP	1004	-	PASS
0005d	NJN d	NON-ZERO JUMP	1005	-	PASS
0006d	PJN d	PLUS JUMP	1006	-	PASS
0007d	MJN d	MINUS JUMP	1007	-	PASS
0010d	SHN d	SHIFT	1010	-	PASS
0011d	LMN d	LOGICAL DIFFERENCE	1011	-	PASS
0012d	LPN d	LOGICAL PRODUCT	1012	-	PASS
0013d	SCN d	SELECTIVE CLEAR	1013	-	PASS
0014d	LDN d	LOAD	1014	-	PASS
0015d	LCN d	LOAD COMPLEMENT	1015	-	PASS
0016d	ADN d	ADD	1016	-	PASS
0017d	SBN d	SUBTRACT	1017	-	PASS
0020dm	LDC dm	LOAD	1020	-	PASS
0021dm	ADC dm	ADD	1021	-	PASS
0022dm	LPC dm	LOGICAL PRODUCT	1022d	LPDL d	LOGICAL PRODUCT
0023dm	LMC dm	LOGICAL DIFFERENCE	1023d	LPIL d	LOGICAL PRODUCT
002400	PSN	PASS	1024dm	LPML m,d	LOGICAL PRODUCT
0024d	LRD d	LOAD R			
002500		PASS	1025	-	PASS
0025d	SRD d	STORE R			
002600	EXN	EXCHANGE JUMP	1026d	INPN d	INTERRUPT PROCESSOR
002610	MXN	MONITOR EXCHANGE JUMP			
002620	MAN	MONITOR EXCHANGE JUMP MA			
0027d	KPT d	KEYPOINT	1027	-	PASS

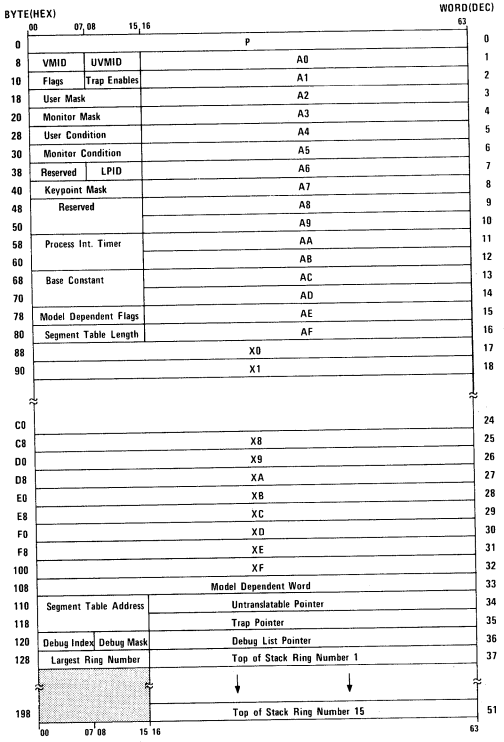
PP INSTRUCTIONS (VIRTUAL STATE AND CYBER 170 STATE)
(Sheet 2 OF 3)

OP CODE	MNEMONIC	NAME	OP CODE	MNEMONIC	NAME
0030d	LDD d	LOAD	1030d	LDDL d	LOAD
0031d	ADD d	ADD	1031d	ADDL d	ADD
0032d	SBD d	SUBTRACT	1032d	SBDL d	SUBTRACT
0033d	LMD d	LOGICAL DIFFERENCE	1033d	LMDL d	LOGICAL DIFFERENCE
0034d	STD d	STORE	1034d	STDL d	STORE
0035d	RAD d	REPLACE	1035d	RADL d	REPLACE
0036d	AOD d	REPLACE ADD ONE	1036d	AODL d	REPLACE ADD ONE
0037d	SOD d	REPLACE SUBTRACT ONE	1037d	SODL d	REPLACE SUBTRACT ONE
0040d	LDI d	LOAD	1040d	LDIL d	LOAD
0041d	ADI d	ADD	1041d	ADIL d	ADD
0042d	SBI d	SUBTRACT	1042d	SBIL d	SUBTRACT
0043d	LMI d	LOGICAL DIFFERENCES	1043d	LMIL d	LOGICAL DIFFERENCE
0044d	STI d	STORE	1044d	STIL d	STORE
0045d	RAI d	REPLACE	1045d	RAIL d	REPLACE
0046d	AOI d	REPLACE ADD ONE	1046d	AOIL d	REPLACE ADD ONE
0047d	SOI d	REPLACE SUBTRACT ONE	1047d	SOIL d	REPLACE SUBTRACT ONE
0050dm	LDM m,d	LOAD	1050dm	LDML m,d	LOAD
0051dm	ADM m,d	ADD	1051dm	ADML m,d	ADD
0052dm	SBM m,d	SUBTRACT	1052dm	SBML m,d	SUBTRACT
0053dm	LMM m,d	LOGICAL DIFFERENCE	1053dm	LMML m,d	LOGICAL DIFFERENCE
0054dm	STM m,d	STORE	1054dm	STML m,d	STORE
0055dm	RAM m,d	REPLACE	1055dm	RAML m,d	REPLACE
0056dm	AOM m,d	REPLACE ADD ONE	1056dm	AOML m,d	REPLACE ADD ONE
0057dm	SOM m,d	REPLACE SUBTRACT ONE	1057dm	SOML m,d	REPLACE SUBTRACT ONE
0060d	CRD d	CENTRAL READ TO d	1060d	CRDL d	CENTRAL READ TO d
0061dm	CRM m,d	CENTRAL READ d WORDS	1061dm	CRML m,d	CENTRAL READ d WORDS
0062d	CWD d	CENTRAL WRITE FROM d	1062d	CWDL d	CENTRAL WRITE FROM d
0063dm	CWM m,d	CENTRAL WRITE d WORDS	1063dm	CWML m,d	CENTRAL WRITE d WORDS

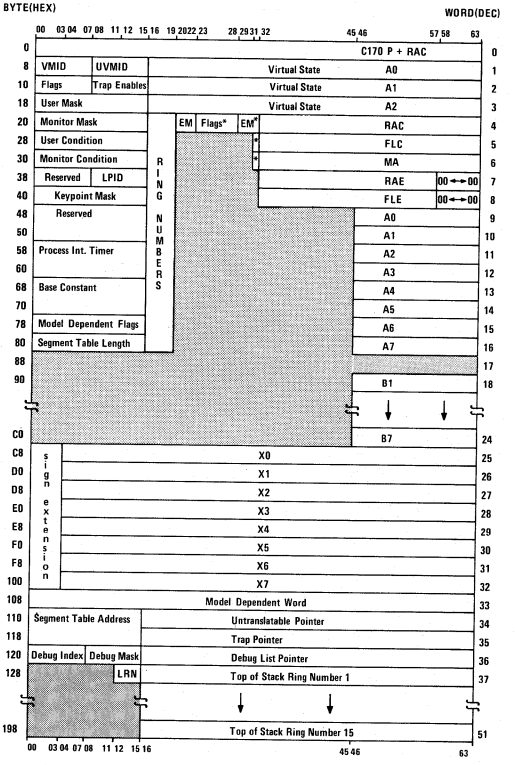
PP INSTRUCTIONS (VIRTUAL STATE AND CYBER 170 STATE)
(Sheet 3 OF 3)

OP CODE	MNEMONIC	NAME	OP CODE	MNEMONIC	NAME
00640cm	AJM m,c	JUMP IF CH. C ACTIVE	10640cm	FSJM m,c	JUMP IF CH. C FLAG SET
00641cm	SCF m,c	TEST AND SET CH. C FLAG			
00650cm	IJM m,c	JUMP IF CH. C INACTIVE	10650cm	FCJM m,c	JUMP IF CH. C FLAG CLEAR
00651cm	CCF c	CLEAR CH. C FLAG			
00660cm	FJM m,c	JUMP IF CH. C FULL	1066	-	PASS
00661cm	SFM m,c	JUMP IF CH. C ERROR FLAG SET			
00670cm	EJM m,c	JUMP IF CH. C EMPTY	1067	-	PASS
00671cm	CFM m,c	JUMP IF CH. C ERROR FLAG CLEAR			
00700c	IAN c	INPUT TO A FROM CH. C	1070	-	PASS
00701c	IAN 40B+c	INPUT TO A FROM CH. C			
00710cm	IAM m,c	INPUT A WORDS FROM CH. C	10710cm	IAPM m,c	INPUT A WORDS PACKED FROM CH. C
00720c	OAN c	OUTPUT FROM A ON CH. C	1072	-	PASS
00721c	OAN 40B+c	OUTPUT FROM A ON CH. C			
00730cm	OAM m,c	OUTPUT A WORDS ON CH. C	10730cm	OAPM m,c	OUTPUT A WORDS PACKED ON CH. C
00740c	ACN c	ACTIVATE CH. C	1074	-	PASS
00741c	ACN 40B+c	ACTIVATE CH. C			
00750c	DCN c	DEACTIVATE CH. C	1075	-	PASS
00751c	DCN 40B+c	DEACTIVATE CH. C			
00760c	FAN c	FUNCTION A ON CH. C	1076	-	PASS
00761c	FAN 40B+c	FUNCTION A ON CH. C			
00770cm	FNC m,c	FUNCTION M ON CH. C	1077	-	PASS
00771cm	FNC m,40B+c	FUNCTION M ON CH. C			

VIRTUAL STATE EXCHANGE PACKAGE



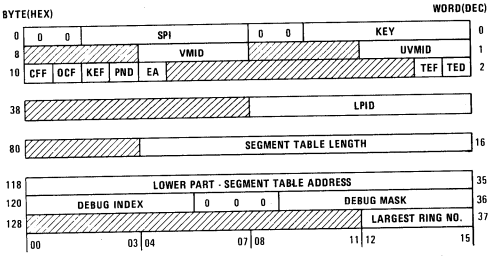
INTERSTATE EXCHANGE PACKAGE



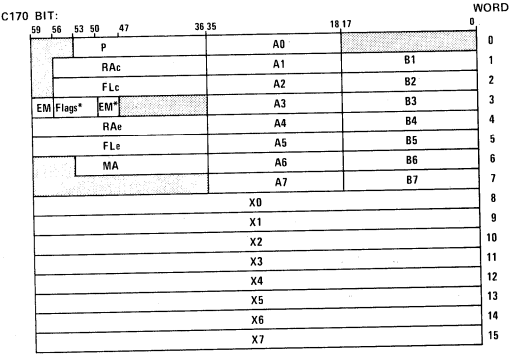
* Flags:

Bit 23, Word 4	UEM Enable Flag
Bit 24, Word 4	Expanded Addressing Select Flag (ESM Mode Flag)
Bit 25, Word 4	Extended Block Copy Flag
Bit 26, Word 4	Software Flag
Bit 27, Word 4	Instruction Stack Purge Flag
Bit 28, Word 4	Software Flag
Bit 29, Word 4	EM-Indefinite Operand
Bit 30, Word 4	EM-Infinite Operand
Bit 31, Word 4	EM-Address out of Range
Bit 31, Word 5	C170 Monitor Flag
Bit 31, Word 6	Exit Mode Halt

DETAIL FOR VIRTUAL STATE AND INTERSTATE EXCHANGE PACKAGES



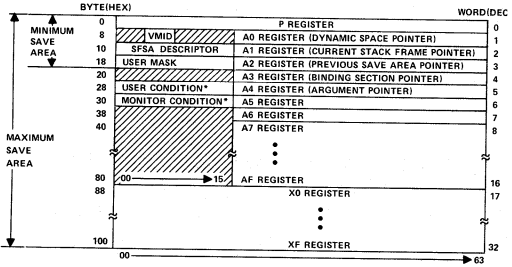
CYBER 170 STATE EXCHANGE PACKAGE



* Flags:

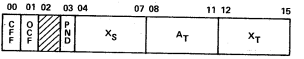
Bit 56	UEM Enable Flag
Bit 55	Expanded Addressing Select Flag (ESM Mode Flag)
Bit 54	Extended Block Copy Flag
Bit 53	Software Flag
Bit 52	Instruction Stack Purge Flag
Bit 51	Software Flag
Bit 50	EM-Indefinite Operand
Bit 49	EM-Infinite Operand
Bit 48	EM-Address out of Range

STACK FRAME SAVE AREA



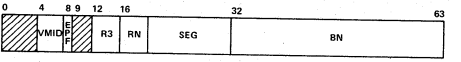
* UCR AND MCR ARE STORED ON TRAP OPERATIONS. ON CALL OPERATIONS, UCR AND MCR POSITIONS IN THE SFSA ARE UNDEFINED.

STACK FRAME SAVE AREA DESCRIPTOR



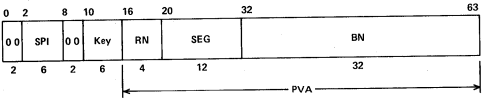
CFF	CRITICAL FRAME FLAG
OCF	ON CONDITION FLAG
PND	PROCESS NOT DAMAGED FLAG
X_S	X REGISTER, STARTING NUMBER
A_T	A REGISTER, TERMINATING NUMBER
X_T	X REGISTER, TERMINATING NUMBER

CODE BASE POINTER - CBP

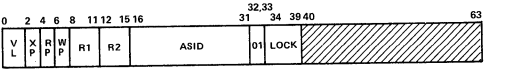


VMID	VIRTUAL MACHINE IDENTIFIER
EPF	EXTERNAL PROCEDURE FLAG
R3	HIGHEST RING NUMBER FOR CALL
RN	RING NUMBER
SEG	SEGMENT NUMBER
BN	BYTE NUMBER

PROGRAM ADDRESS REGISTER - P REGISTER



SEGMENT DESCRIPTOR TABLE ENTRY - SDE



VL
00 INVALID ENTRY
01 (RESERVED)
10 REGULAR SEGMENT
11 CACHE BY-PASS SEGMENT

XP
00 NONEXECUTABLE SEGMENT
01 NONPRIVILEGED EXECUTABLE SEGMENT
10 LOCAL PRIVILEGED EXECUTABLE SEGMENT
11 GLOBAL PRIVILEGED EXECUTABLE SEGMENT

RP
00 NONREADABLE SEGMENT
01 READ CONTROLLED BY KEY/LOCK
10 READ NOT CONTROLLED BY KEY/LOCK
11 BINDING SECTION SEGMENT –
READ NOT CONTROLLED BY KEY/LOCK

WP
00 NONWRITABLE SEGMENT
01 WRITE CONTROLLED BY KEY/LOCK
10 WRITE NOT CONTROLLED BY KEY/LOCK
11 (RESERVED)

PAGE DESCRIPTOR

Bit No.	Set	Clear	
00	Valid entry	Invalid entry	Control
01	Continue search	May stop search	
02	Used	Unused	Status
03	Modified	Unmodified	

PROCESSOR REGISTER DEFINITIONS AND ACCESSES

REGISTER GROUP	REGISTER NUMBER(S)	REGISTER NAME	COPY ACCESS PRIVILEGES: COPY FROM STATE REGISTER (READ)	COPY ACCESS PRIVILEGES: COPY TO STATE REGISTER (WRITE)	MCU ACCESS
00-0F	00	STATUS SUMMARY	NO ACCESS		
	10	ELEMENT ID			
10-1F	11	PROCESSOR ID	UNPRIV.	NO ACCESS	R
	12	OPTIONS INSTALLED			
	13	VIRTUAL MACHINE CAPABILITY LIST			
	21	RESERVED			
	22	PERFORMANCE MONITORING FACILITY			
	30	DEPENDENT ENVIRONMENT CONTROL			
20-2F	31	CONTROL STORE ADDRESS	NO ACCESS	NO ACCESS	R/W
30-3F	32	CONTROL STORE BREAKPOINT			
	40	P REGISTER			
	41	MONITOR PROCESS STATE POINTER			
	42	MONITOR CONDITION REGISTER			
	43	USER CONDITION REGISTER			
	44	UNTRANSLATABLE POINTER			
40-4F	45	SEGMENT TABLE LENGTH			
	46	SEGMENT TABLE ADDRESS			
50-5F	47	BASE CONSTANT	UNPRIV.	NO ACCESS	R/W
	48	PAGE TABLE ADDRESS			
	49	PAGE TABLE LENGTH			
	4A	PAGE SIZE MASK			
	50	MODEL DEPENDENT FLAGS			
	51	MODEL DEPENDENT WORD			
	60	MONITOR MASK REGISTER			
60-6F	61	JOB PROCESS STATE POINTER	UNPRIV.	MONITOR	R/W
70-7F	62	SYSTEM INTERVAL TIMER			
	63	KEYPOINT BUFFER POINTER			
	80-8F	PROCESSOR FAULT STATUS			
80-8F	90	RETRY CORRECTED ERROR LOG			
90-9F	91	CONTROL STORE CORRECTED ERROR LOG	UNPRIV.	GLOBAL	R/W
A0-AF	92	CACHE CORRECTED ERROR LOG			
B0-BF	93	MAP CORRECTED ERROR LOG			
	A0	PROCESSOR TEST MODE			
	C0-C3	TRAP ENABLES			
	C4	TRAP POINTER			
C0-CF	C5	DEBUG LIST POINTER			
	C6	KEYPOINT MASK	UNPRIV.	LOCAL	R/W
D0-DF	C7	RESERVED			
	C8	RESERVED			
	C9	PROCESS INTERVAL TIMER			
	CA-CB	KEYPOINT ENABLE FLAG			
	E0-E1	CRITICAL FRAME FLAG			
E0-EF	E2-E3	ON CONDITION FLAG			
	E4	DEBUG INDEX	UNPRIV.	UNPRIV.	R/W
F0-FF	E5	DEBUG MASK REGISTER			
	E6	USER MASK REGISTER			

PURGE BUFFER (05)

PURGE CACHE		PURGE MAP	
k		k	
0	512 bytes per SVA	8	PTE per SVA
1	ASID per SVA	9	PTEs in segment per SVA
2	All	A	PTE per PVA
3	512 bytes per PVA	B	SDE per PVA plus PTEs in segment per PVA
*4a.	All in Instr. Stack		
b.	No-op. Allowable per/Intersegment Branch	**C-E	No-op
*5-6	No-op	F	All
7	Seg per PVA		

* Certain processors execute as if K=7

** Certain processors execute as if K=F

MONITOR CONDITION REGISTER (42)

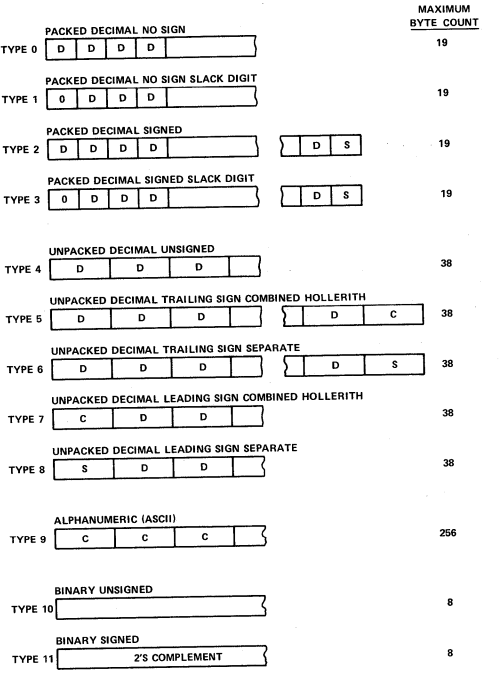
P RGTR	BIT NUMBER AND DEFINITION			ASSOCIATED MONITOR MASK REGISTER BIT SET				MASK BIT CLEAR
				TRAP ENABLED		TRAP DISABLED		TRAP ENABLED OR DISABLED
				JOB MODE	MONITOR MODE	JOB MODE	MONITOR MODE	JOB OR MONITOR MODE
–	48	Detected Uncorrectable Error	Mon	EXCH	TRAP	EXCH	HALT	HALT
–	49	Unassigned		EXCH	TRAP	EXCH	HALT	HALT
P+	50	Short Warning	Sys	EXCH	TRAP	EXCH	STACK	STACK
P	51	Instruction Specification Error	Mon	EXCH	TRAP	EXCH	HALT	HALT
P	52	Address Specification Error	Mon	EXCH	TRAP	EXCH	HALT	HALT
P+	53	170 Exchange Request	Sys	EXCH	TRAP	EXCH	STACK	STACK
P	54	Access Violation	Mon	EXCH	TRAP	EXCH	HALT	HALT
P	55	Environment Specification Error	Mon	EXCH	TRAP	EXCH	HALT	HALT
	56	External Interrupt	Sys	EXCH	TRAP	EXCH	STACK	STACK
	57	Page Table Search Without Find	Mon	EXCH	TRAP	EXCH	HALT	HALT
	58	System Call	Status - This bit is a flag only and does not cause any hardware action.					
	59	System Interval Timer	Sys	EXCH	TRAP	EXCH	STACK	STACK
P/P+*	60	Invalid Segment/Ring Number Zero	Mon	EXCH	TRAP	EXCH	HALT	HALT
P	61	Outward Call/Inward Return	Mon	EXCH	TRAP	EXCH	HALT	HALT
P+	62	Soft Error Log	Sys	EXCH	TRAP	EXCH	STACK	STACK
–	63	Trap Exception	Status - This bit is a flag only and does not cause any hardware action.					

* P, unless P+ for RNO on loads

USER CONDITION REGISTER (43)

P RGTR	BIT NUMBER AND DEFINITION			ASSOCIATED USER MASK REGISTER BIT SET				MASK BIT CLEAR
				TRAP ENABLED		TRAP DISABLED		TRAP ENABLED OR DISABLED
				JOB MODE	MONITOR MODE	JOB MODE	MONITOR MODE	JOB OR MONITOR MODE
P	48	Privileged Instruction Fault	Mon	TRAP	TRAP	EXCH	HALT	These mask bits are permanently set.
P	49	Unimplemented Instruction	Mon	TRAP	TRAP	EXCH	HALT	
P	50	Free Flag	User	TRAP	TRAP	STACK	STACK	
P+	51	Process Interval Timer	User	TRAP	TRAP	STACK	STACK	
P	52	Inter-ring Pop	Mon	TRAP	TRAP	EXCH	HALT	
P	53	Critical Frame Flag	Mon	TRAP	TRAP	EXCH	HALT	
–	54	Unassigned	User	TRAP	TRAP	STACK	STACK	
P	55	Divide Fault	User	TRAP	TRAP	STACK	STACK	STACK
P	56	Debug	User	TRAP	TRAP	STACK	STACK	STACK
P	57	Arithmetic Overflow	User	TRAP	TRAP	STACK	STACK	STACK
P+	58	Exponent Overflow	User	TRAP	TRAP	STACK	STACK	STACK
P+	59	Exponent Underflow	User	TRAP	TRAP	STACK	STACK	STACK
P+	60	F. P. Loss of Significance	User	TRAP	TRAP	STACK	STACK	STACK
P	61	F. P. Indefinite	User	TRAP	TRAP	STACK	STACK	STACK
P	62	Arithmetic Loss of Significance	User	TRAP	TRAP	STACK	STACK	STACK
P	63	Invalid BDP Data	User	TRAP	TRAP	STACK	STACK	STACK

BDP DATA TYPES



FLOATING POINT REPRESENTATION

	HEXADECIMAL EXPONENT INCLUDING COEFFICIENT SIGN	ACTUAL EXPONENT (TO THE BASE 2)	INPUT ARGUMENTS	
COEFFICIENT SIGN EQUAL TO 0 (POSITIVE NUMBERS)	7XXX	– – – –	INDEFINITE	
	6FFF ↑ 5000	$2^{12,287}$ ↑ $2^{4,096}$	INFINITE	
	4FFF ↑ 4000 3FFF ↓ 3000	2^{4095} ↑ 2^{0} 2^{-1} ↓ $2^{-4,096}$	STANDARD	NUMBERS IN THIS RANGE WITH ZERO COEFFICIENTS ARE TERMED +Z3
	2FFF ↓ 1000	$2^{-4,097}$ ↓ $2^{-12,288}$	ZERO	+Z2
	0XXX	$2^{-12,289}$ ↓ $2^{-16,384}$	ZERO	+Z1
COEFFICIENT SIGN EQUAL TO 1 (NEGATIVE NUMBERS)	8XXX	$2^{-16,384}$ ↑ $2^{-12,289}$	ZERO	–Z1
	9000 ↑ AFFF	$2^{-12,288}$ ↑ $2^{-4,097}$	ZERO	–Z2
	B000 ↑ BFFF C000 ↓ CFFF	$2^{-4,096}$ ↑ 2^{-1} 2^{0} ↓ $2^{4,095}$	STANDARD	NUMBERS IN THIS RANGE WITH ZERO COEFFICIENTS ARE TERMED –Z3
	D000 ↓ EFFF	2^{4096} ↓ $2^{12,287}$	INFINITE	
	FXXX	– – –	INDEFINITE	

CHARACTER SET

Code	Character	Code	Character
00	NUL	20	SP
01	SOH	21	!
02	STX	22	"
03	ETX	23	#
04	EOT	24	$
05	ENQ	25	%
06	ACK	26	&
07	BEL	27	'
08	BS	28	(
09	HT	29	)
0A	LF	2A	*
0B	VT	2B	+
0C	FF	2C	,
0D	CR	2D	-
0E	SO	2E	.
0F	SI	2F	/
10	DLE	30	0
11	DC1	31	1
12	DC2	32	2
13	DC3	33	3
14	DC4	34	4
15	NAK	35	5
16	SYN	36	6
17	ETB	37	7
18	CAN	38	8
19	EM	39	9
1A	SUB	3A	:
1B	ESC	3B	;
1C	FS	3C	<
1D	GS	3D	=
1E	RS	3E	>
1F	US	3F	?

Code	Character	Code	Character
40	@	60	`
41	A	61	a
42	B	62	b
43	C	63	c
44	D	64	d
45	E	65	e
46	F	66	f
47	G	67	g
48	H	68	h
49	I	69	i
4A	J	6A	j
4B	K	6B	k
4C	L	6C	l
4D	M	6D	m
4E	N	6E	n
4F	O	6F	o
50	P	70	p
51	Q	71	q
52	R	72	r
53	S	73	s
54	T	74	t
55	U	75	u
56	V	76	v
57	W	77	w
58	X	78	x
59	Y	79	y
5A	Z	7A	z
5B	[	7B	{
5C	\	7C	|
5D	]	7D	}
5E	^	7E	~
5F	_	7F	DEL

GLOSSARY

ASID	Active segment identifier
BDP	Business data processing
BN	Byte number
CBP	Code base pointer
CFF	Critical frame flag
EM	Exit mode
EPF	External procedure flag
LPID	Last processor identification
MCR	Monitor condition register
OCF	On-condition flag
P	Program address register
PEP	Physical ECS permission
PFA	Page frame address
PND	Process-not-damaged flag
PP	Peripheral processor
PTE	Page table entry
PVA	Process virtual address
RN	Ring number
RP	Read access control (SDE)
SDE	Segment descriptor table entry
SEG	Segment number
SFSA	Stack frame save area
SVA	System virtual address
TED	Trap enable delay flip-flop
TEF	Trap enable flip-flop
UCR	User condition register
UEM	Unified extended memory
UVMID	Untranslatable virtual machine identifier
VL	Segment validation (SDE)
VMID	Virtual machine identifier
WP	Write access control (SDE)
XP	Execute access control (SDE)

	MEMORY SIZE OPTIONS (MEGABYTES)											
	2	4	6	8	12	16	24	32	64	96	128	DUAL-CP OPTION
CYBER 180 MODEL 810	X	X		X	X	X						NO
CYBER 170 MODEL 815	X	X		X								NO
CYBER 170 MODEL 825	X	X		X								NO
CYBER 180 MODEL 830	X	X		X	X	X						YES
CYBER 170/180 MODEL 835		X		X	X	X						YES
CYBER 180 MODEL 840						X		X	X	X	X	NO
CYBER 170/180 MODEL 845		X		X	X	X						NO
CYBER 180 MODEL 850						X		X	X	X	X	NO
CYBER 170/180 MODEL 855		X		X	X	X						YES
CYBER 180 MODEL 860						X		X	X	X	X	YES
CYBER 180 MODEL 990				X		X	X	X				YES

(continued)

	MEMORY SIZE OPTIONS (MEGABYTES)											
	16	24	32	48	64	80	96	112	128	192	256	DUAL-CP OPTION
CYBER 840S/845S	X		X		X		X		X			NO
CYBER 855S	X		X		X		X		X			YES
CYBER 840A/850A	X		X		X		X		X			NO
CYBER 860A	X		X		X		X		X			YES
CYBER 870A	X		X		X		X		X			STD
CYBER 960 MODEL 11					X				X	X	X	NO
CYBER 960/962 MODEL 31					X				X	X	X	YES
CYBER 960/962 MODEL 32					X				X	X	X	STD
CYBER 962 MODEL 11			X		X				X	X	X	NO
CYBER 990E	X		X	X	X	X	X	X	X			NO
CYBER 995E	X		X	X	X	X	X	X	X			STD

M02489-1

'ORATE HEADQUARTERS
3OX O.
IEAPOLIS, MINNESOTA 55440

S OFFICES AND SERVICE CENTERS
AJOR CITIES
)UGHOUT THE WORLD

CONTROL DATA