Enterprise Systems
Architecture/390

Reference Summary

SA22-7209-02

Enterprise Systems
Architecture/390

Reference Summary

SA22-7209-02

Third Edition (November 1999)

This revision differs from the previous edition by containing minor corrections and also instructions, control blocks, and fields related to the new facilities marked by a bar under “Facility” in “Preface.” Additions to the assembler also are included. Changes are indicated by a bar in the margin.

References in this publication to IBM products, programs, or services do not imply that IBM intends to make these available in all countries in which IBM operates. Any reference to an IBM program product in this publication is not intended to state or imply that only IBM's program product may be used. Any functionally equivalent program may be used instead.

Requests for copies of this and other IBM publications should be made to your IBM representative or to the IBM branch office serving your locality.

Please direct any comments on the contents of this publication to

IBM Corporation
Department E57
2455 South Road
Poughkeepsie, NY
12601-5400
USA

IBM may use or distribute whatever information you supply in any way it believes appropriate without incurring any obligation to you.

Preface

This publication is intended primarily for use by Enterprise Systems Architecture/390™ (ESA/390™) assembler-language application programmers. It contains basic machine information summarized from the *IBM ESA/390 Principles of Operation*, SA22-7201, about the S/390™ processors. It also contains frequently used information from *IBM ESA/390 Vector Operations*, SA22-7207, *IBM ESA/390 Data Compression*, SA22-7208, *IBM ESA/390 Common I/O-Device Commands and Self Description*, SA22-7204, *IBM System/370 Extended Architecture Interpretive Execution*, SA22-7095, and *IBM High Level Assembler/MVS & VM Language Reference*, SC26-4940. This publication will be updated from time to time. However, the above publications and others cited in this publication are the authoritative reference sources and will be first to reflect changes.

As a possible convenience to the reader, this publication carries forward most of the device information in the previous edition. This information has not been updated.

The following instructions may be uninstalled or not available on a particular model:

Facility	Instruction
Additional floating point	(All instructions marked with "a" in "Class & Notes" column)
Basic vector	(All instructions with mnemonics that start with "V" except as listed below)
Branch and set authority	BSA
Checksum	CKSM
Compare and move extended	CLCLE, MVCLE
Compression	CMPSC
Extended TOD clock	SCKPF, STCKE
Extended translation	CUTFU, CUUTF, TRE
Immediate and relative	AHI, BRAS, BRC, BRCT, BRXH, BRXLE, CHI, LHI, MHI, MS, MSR, TMH, TML
Load VIX	VLVXA
Move inverse	MVCIN
Multiply then add/subtract	VTAD, VTAE, VTSD, VTSE
Perform locked operation	PLO
Program call fast	PCF
Resume program	RP
Set address space control fast	SACF
Store system information	STSI
String	CLST, MVST, SRST
Square root	SQDR, SQER
Subspace group	BSG
Trap	TRAP2, TRAP4
Vector square root	VSQD, VSQDR, VSQE, VSQER

For information about System/370™ architecture, refer to *IBM System/370 Principles of Operation*, GA22-7000, and *IBM System/370 Reference Summary*, GX20-1850.

Enterprise Systems Architecture/390, ESA/390, S/390, System/370, and Enterprise Systems Architecture/370 are trademarks of the International Business Machines Corporation.

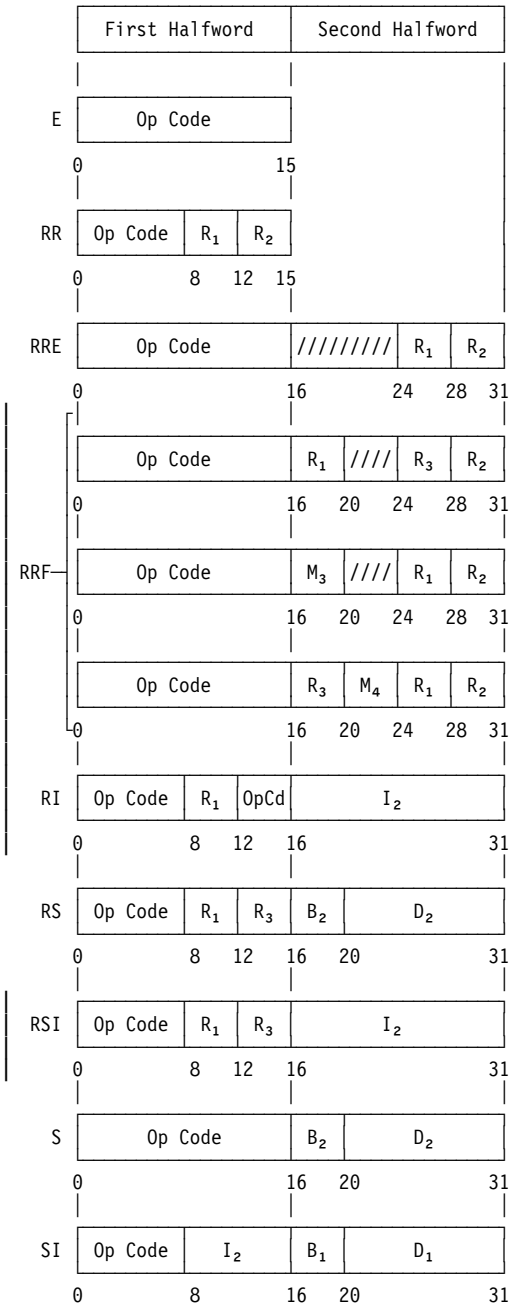
For information about System/370 extended architecture, refer to *IBM System/370 Extended Architecture Principles of Operation*, SA22-7085, and *IBM System/370 Extended Architecture Reference Summary*, GX20-0157.

For information about Enterprise Systems Architecture/370™ architecture, refer to *IBM Enterprise Systems Architecture/370 Principles of Operation*, SA22-7200, *IBM Enterprise Systems Architecture/370 and System/370 Vector Operations*, SA22-7125, and *IBM Enterprise Systems Architecture/370 Reference Summary,* GX20-0406.

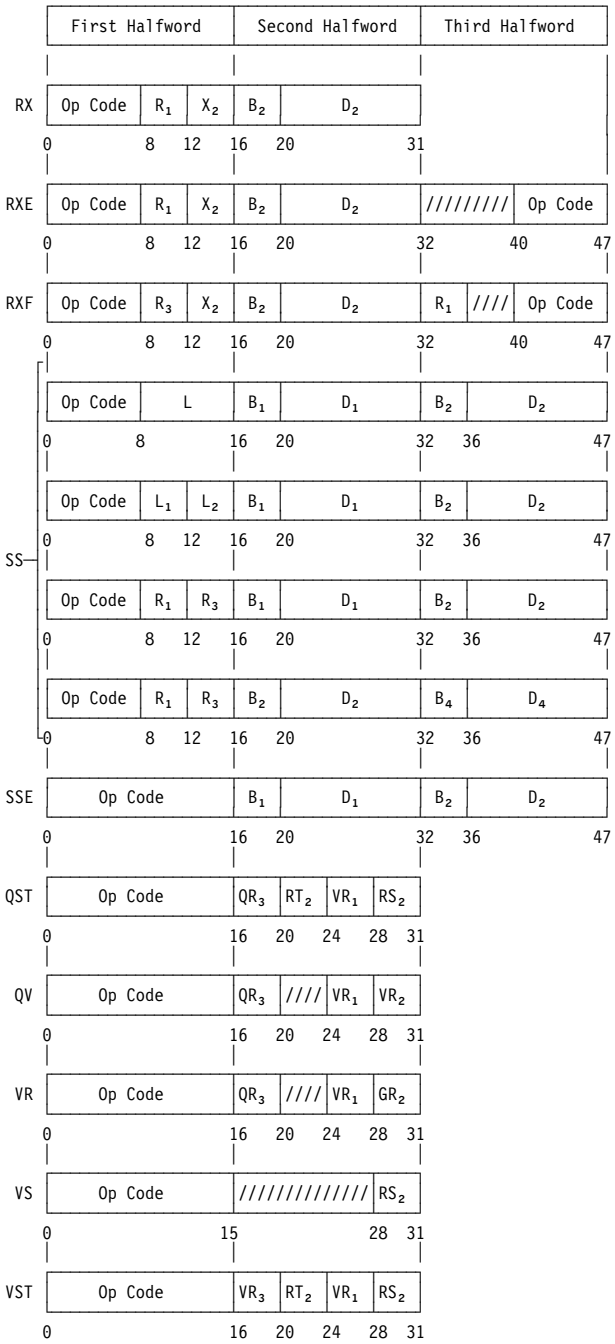
Contents

NOTES

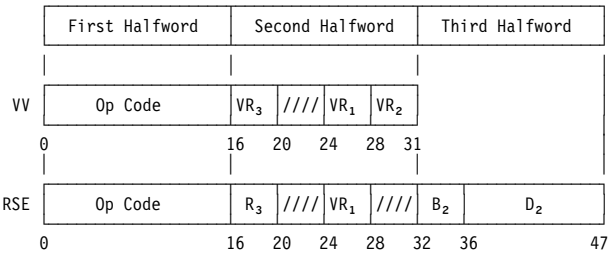
Machine Instruction Formats



Machine Instruction Formats (Cont'd)



Machine Instruction Formats (Cont'd)



$_{1}$, $_{2}$, $_{3}$, $_{4}$:	Denotes association with first, second, third, or fourth operand
B_1, B_2, B_4:	Base register designation field
D_1, D_2, D_4:	Displacement Field
GR_2:	Register designation field (general register)
I_2:	Immediate operand field
L, L_1, L_2:	Length field
M_3, M_4:	Mask field
QR_3:	Register designation field (equivalent to GR_3 if general register, or FR_3 if floating-point register)
R_1, R_2, R_3:	Register designation field
RS_2:	Register designation field (starting address of vector)
RT_2:	Register designation field (stride of vector)
VR_1, VR_2, VR_3:	Register designation field (vector register)
X_2:	Index register designation field

Machine Instructions by Mnemonic

Mnemonic	Operands	Name	Format	Op Code	Class & Notes
A	$R_1,D_2(X_2,B_2)$	Add	RX	5A	c
AD	$R_1,D_2(X_2,B_2)$	Add Normalized (LH)	RX	6A	c
ADB	$R_1,D_2(X_2,B_2)$	Add (LB)	RXE	ED1A	ac
ADBR	R_1,R_2	Add (LB)	RRE	B31A	ac
ADR	R_1,R_2	Add Normalized (LH)	RR	2A	c
AE	$R_1,D_2(X_2,B_2)$	Add Normalized (SH)	RX	7A	c
AEB	$R_1,D_2(X_2,B_2)$	Add (SB)	RXE	ED0A	ac
AEBR	R_1,R_2	Add (SB)	RRE	B30A	ac
AER	R_1,R_2	Add Normalized (SH)	RR	3A	c
AH	$R_1,D_2(X_2,B_2)$	Add Halfword	RX	4A	c
AHI	R_1,I_2	Add Halfword Immediate	RI	A7A	c
AL	$R_1,D_2(X_2,B_2)$	Add Logical	RX	5E	c
ALR	R_1,R_2	Add Logical	RR	1E	c
AP	$D_1(L_1,B_1),D_2(L_2,B_2)$,	Add Decimal	SS	FA	c
AR	R_1,R_2	Add	RR	1A	c
AU	$R_1,D_2(X_2,B_2)$	Add Unnormalized (SH)	RX	7E	c
AUR	R_1,R_2	Add Unnormalized (SH)	RR	3E	c
AW	$R_1,D_2(X_2,B_2)$	Add Unnormalized (LH)	RX	6E	c
AWR	R_1,R_2	Add Unnormalized (LH)	RR	2E	c
AXBR	R_1,R_2	Add (EB)	RRE	B34A	ac
AXR	R_1,R_2	Add Normalized (EH)	RR	36	c
BAKR	R_1,R_2	Branch and Stack	RRE	B240	q
BAL	$R_1,D_2(X_2,B_2)$	Branch and Link	RX	45	
BALR	R_1,R_2	Branch and Link	RR	05	
BAS	$R_1,D_2(X_2,B_2)$	Branch and Save	RX	4D	
BASR	R_1,R_2	Branch and Save	RR	0D	
BASSM	R_1,R_2	Branch and Save and Set Mode	RR	0C	
BC	$M_1,D_2(X_2,B_2)$	Branch on Condition	RX	47	
BCR	M_1,R_2	Branch on Condition	RR	07	
BCT	$R_1,D_2(X_2,B_2)$	Branch on Count	RX	46	
BCTR	R_1,R_2	Branch on Count	RR	06	
BRAS	R_1,I_2	Branch Relative and Save	RI	A75	
BRC	M_1,I_2	Branch Relative on Condition	RI	A74	
BRCT	R_1,I_2	Branch Relative on Count	RI	A76	
BRXH	R_1,R_3,I_2	Branch Relative on Index High	RSI	84	
BRXLE	R_1,R_3,I_2	Branch Relative on Index Low or Equal	RSI	85	
BSA	R_1,R_2	Branch and Set Authority	RRE	B25A	q
BSG	R_1,R_2	Branch in Subspace Group	RRE	B258	
BSM	R_1,R_2	Branch and Set Mode	RR	0B	
BXH	$R_1,R_3,D_2(B_2)$	Branch on Index High	RS	86	
BXLE	$R_1,R_3,D_2(B_2)$	Branch on Index Low or Equal	RS	87	
C	$R_1,D_2(X_2,B_2)$	Compare	RX	59	c
CD	$R_1,D_2(X_2,B_2)$	Compare (LH)	RX	69	c
CDB	$R_1,D_2(X_2,B_2)$	Compare (LB)	RXE	ED19	ac
CDBR	R_1,R_2	Compare (LB)	RRE	B319	ac

Mnemonic	Operands	Name	Format	Op Code	Class & Notes
CDFBR	R_1,R_2	Convert from Fixed (32/LB)	RRE	B395	a
CDFR	R_1,R_2	Convert from Fixed (32/LH)	RRE	B3B5	a
CDR	R_1,R_2	Compare (LH)	RR	29	c
CDS	$R_1,R_3,D_2(B_2)$	Compare Double and Swap	RS	BB	c
CEB	$R_1,D_2(X_2,B_2)$	Compare (SB)	RXE	ED09	ac
CE	$R_1,D_2(X_2,B_2)$	Compare (SH)	RX	79	c
CEBR	R_1,R_2	Compare (SB)	RRE	B309	ac
CEFBR	R_1,R_2	Convert from Fixed (32/SB)	RRE	B394	a
CEFR	R_1,R_2	Convert from Fixed (32/SH)	RRE	B3B4	a
CER	R_1,R_2	Compare (SH)	RR	39	c
CFC	$D_2(B_2)$	Compare and Form Codeword	S	B21A	ic
CFDBR	R_1,M_3,R_2	Convert to Fixed (LB/32)	RRF	B399	ac
CFDR	R_1,M_3,R_2	Convert to Fixed (LH/32)	RRF	B3B9	ac
CFEBR	R_1,M_3,R_2	Convert to Fixed (SB/32)	RRF	B398	ac
CFER	R_1,M_3,R_2	Convert to Fixed (SH/32)	RRF	B3B8	ac
CFXBR	R_1,M_3,R_2	Convert to Fixed (EB/32)	RRF	B39A	ac
CFXR	R_1,M_3,R_2	Convert to Fixed (EH/32)	RRF	B3BA	ac
CH	$R_1,D_2(X_2,B_2)$	Compare Halfword	RX	49	c
CHI	R_1,I_2	Compare Halfword Immediate	RI	A7E	c
CKSM	R_1,R_2	Checksum	RRE	B241	c
CL	$R_1,D_2(X_2,B_2)$	Compare Logical	RX	55	c
CLC	$D_1(L,B_1),D_2(B_2)$	Compare Logical	SS	D5	c
CLCL	R_1,R_2	Compare Logical Long	RR	0F	ic
CLCLE	$R_1,R_3,D_2(B_2)$	Compare Logical Long Extended	RS	A9	c
CLI	$D_1(B_1),I_2$	Compare Logical	SI	95	c
CLM	$R_1,M_3,D_2(B_2)$	Compare Logical Characters under Mask	RS	BD	c
CLR	R_1,R_2	Compare Logical	RR	15	c
CLST	R_1,R_2	Compare Logical String	RRE	B25D	c
CMPSC	R_1,R_2	Compression Call	RRE	B263	ic
CP	$D_1(L_1,B_1),D_2(L_2,B_2)$	Compare Decimal	SS	F9	c
CPYA	R_1,R_2	Copy Access	RRE	B24D	
CR	R_1,R_2	Compare	RR	19	c
CS	$R_1,R_3,D_2(B_2)$	Compare and Swap	RS	BA	c
CSCH		Clear Subchannel	S	B230	pc
CUSE	R_1,R_2	Compare until Substring Equal	RRE	B257	ic
CUTFU	R_1,R_2	Convert UTF-8 to Unicode	RRE	B2A7	c
CUUTF	R_1,R_2	Convert Unicode to UTF-8	RRE	B2A6	c
CVB	$R_1,D_2(X_2,B_2)$	Convert to Binary	RX	4F	
CVD	$R_1,D_2(X_2,B_2)$	Convert to Decimal	RX	4E	
CXBR	R_1,R_2	Compare (EB)	RRE	B349	ac
CXFBR	R_1,R_2	Convert from Fixed (32/EB)	RRE	B396	a
CXFR	R_1,R_2	Convert from Fixed (32/EH)	RRE	B3B6	a
CXR	R_1,R_2	Compare (EH)	RRE	B369	ac
D	$R_1,D_2(X_2,B_2)$	Divide	RX	5D	
DD	$R_1,D_2(X_2,B_2)$	Divide (LH)	RX	6D	
DDB	$R_1,D_2(X_2,B_2)$	Divide (LB)	RXE	ED1D	a
DDBR	R_1,R_2	Divide (LB)	RRE	B31D	a
DDR	R_1,R_2	Divide (LH)	RR	2D	
DE	$R_1,D_2(X_2,B_2)$	Divide (SH)	RX	7D	
DEB	$R_1,D_2(X_2,B_2)$	Divide (SB)	RXE	ED0D	a
DEBR	R_1,R_2	Divide (SB)	RRE	B30D	a
DER	R_1,R_2	Divide (SH)	RR	3D	
DIDBR	R_1,R_3,R_2,M_4	Divide to Integer (LB)	RRF	B35B	ac
DIEBR	R_1,R_3,R_2,M_4	Divide to Integer (SB)	RRF	B353	ac
DP	$D_1(L_1,B_1),D_2(L_2,B_2)$	Divide Decimal	SS	FD	
DR	R_1,R_2	Divide	RR	1D	
DXBR	R_1,R_2	Divide (EB)	RRE	B34D	a
DXR	R_1,R_2	Divide (EH)	RRE	B22D	
EAR	R_1,R_2	Extract Access	RRE	B24F	
ED	$D_1(L,B_1),D_2(B_2)$	Edit	SS	DE	c
EDMK	$D_1(L,B_1),D_2(B_2)$	Edit and Mark	SS	DF	c
EFPC	R_1	Extract FPC	RRE	B38C	a
EPAR	R_1	Extract Primary ASN	RRE	B226	q
EREG	R_1,R_2	Extract Stacked Registers	RRE	B249	
ESAR	R_1	Extract Secondary ASN	RRE	B227	q
ESTA	R_1,R_2	Extract Stacked State	RRE	B24A	c
EX	$R_1,D_2(X_2,B_2)$	Execute	RX	44	
FIDBR	R_1,M_3,R_2	Load FP Integer (LB)	RRF	B35F	a
FIDR	R_1,R_2	Load FP Integer (LH)	RRE	B37F	a
FIEBR	R_1,M_3,R_2	Load FP Integer (SB)	RRF	B357	a
FIER	R_1,R_2	Load FP Integer (SH)	RRE	B377	a
FIXBR	R_1,M_3,R_2	Load FP Integer (EB)	RRF	B347	a
FIXR	R_1,R_2	Load FP Integer (EH)	RRE	B367	a
HDR	R_1,R_2	Halve (LH)	RR	24	
HER	R_1,R_2	Halve (SH)	RR	34	
HSCH		Halt Subchannel	S	B231	pc
IAC	R_1	Insert Address Space Control	RRE	B224	qc
IC	$R_1,D_2(X_2,B_2)$	Insert Character	RX	43	
ICM	$R_1,M_3,D_2(B_2)$	Insert Characters under Mask	RS	BF	c
IPK		Insert PSW Key	S	B20B	q
IPM	R_1	Insert Program Mask	RRE	B222	
IPTE	R_1,R_2	Invalidate Page Table Entry	RRE	B221	p

Machine Instructions by Mnemonic (Cont'd)

Mnemonic	Operands	Name	Format	Op Code	Class & Notes
ISKE	R_1,R_2	Insert Storage Key Extended	RRE	B229	p
IVSK	R_1,R_2	Insert Virtual Storage Key	RRE	B223	q
KDB	$R_1,D_2(X_2,B_2)$	Compare and Signal (LB)	RXE	ED18	ac
KDBR	R_1,R_2	Compare and Signal (LB)	RRE	B318	ac
KEB	$R_1,D_2(X_2,B_2)$	Compare and Signal (SB)	RXE	ED08	ac
KEBR	R_1,R_2	Compare and Signal (SB)	RRE	B308	ac
KXBR	R_1,R_2	Compare and Signal (EB)	RRE	B348	ac
L	$R_1,D_2(X_2,B_2)$	Load	RX	58	
LA	$R_1,D_2(X_2,B_2)$	Load Address	RX	41	
LAE	$R_1,D_2(X_2,B_2)$	Load Address Extended	RX	51	
LAM	$R_1,R_3,D_2(B_2)$	Load Access Multiple	RS	9A	
LASP	$D_1,(B_1),D_2(B_2)$	Load Address Space Parameters	SSE	E500	pc
LCDBR	R_1,R_2	Load Complement (LB)	RRE	B313	ac
LCDR	R_1,R_2	Load Complement (LH)	RR	23	c
LCEBR	R_1,R_2	Load Complement (SB)	RRE	B303	ac
LCER	R_1,R_2	Load Complement (S)	RR	33	c
LCR	R_1,R_2	Load Complement	RR	13	c
LCTL	$R_1,R_3,D_2(B_2)$	Load Control	RS	B7	p
LCXBR	R_1,R_2	Load Complement (EB)	RRE	B343	ac
LCXR	R_1,R_2	Load Complement (EH)	RRE	B363	ac
LD	$R_1,D_2(X_2,B_2)$	Load (L)	RX	68	
LDE	$R_1,D_2(X_2,B_2)$	Load Lengthened (SH/LH)	RXE	ED24	a
LDEB	$R_1,D_2(X_2,B_2)$	Load Lengthened (SB/LB)	RXE	ED04	a
LDEBR	R_1,R_2	Load Lengthened (SB/LB)	RRE	B304	a
LDER	R_1,R_2	Load Lengthened (SH/LH)	RRE	B324	a
LDR	R_1,R_2	Load (L)	RR	28	
LDXBR	R_1,R_2	Load Rounded (EB/LB)	RRE	B345	a
LDXR	R_1,R_2	Load Rounded (EH/LH)	RR	25	
LE	$R_1,D_2(X_2,B_2)$	Load (S)	RX	78	
LEDBR	R_1,R_2	Load Rounded (LB/SB)	RRE	B344	a
LEDR	R_1,R_2	Load Rounded (LH/SH)	RR	35	
LER	R_1,R_2	Load (S)	RR	38	
LEXBR	R_1,R_2	Load Rounded (EB/SB)	RRE	B346	a
LEXR	R_1,R_2	Load Rounded (EH/SH)	RRE	B366	a
LFPC	$D_2(B_2)$	Load FPC	S	B29D	a
LH	$R_1,D_2(X_2,B_2)$	Load Halfword	RX	48	
LHI	R_1,I_2	Load Halfword Immediate	RI	A78	
LM	$R_1,R_3D_2(B_2)$	Load Multiple	RS	98	
LNDBR	R_1,R_2	Load Negative (LB)	RRE	B311	ac
LNDR	R_1,R_2	Load Negative (LH)	RR	21	c
LNEBR	R_1,R_2	Load Negative (SB)	RRE	B301	ac
LNER	R_1,R_2	Load Negative (SH)	RR	31	c
LNR	R_1,R_2	Load Negative	RR	11	c
LNXBR	R_1,R_2	Load Negative (EB)	RRE	B341	ac
LNXR	R_1,R_2	Load Negative (EH)	RRE	B361	ac
LPDBR	R_1,R_2	Load Positive (LB)	RRE	B310	ac
LPDR	R_1,R_2	Load Positive (LH)	RR	20	c
LPEBR	R_1,R_2	Load Positive (SB)	RRE	B300	ac
LPER	R_1,R_2	Load Positive (SH)	RR	30	c
LPR	R_1,R_2	Load Positive	RR	10	c
LPSW	$D_2(B_2)$	Load PSW	S	82	pn
LPXBR	R_1,R_2	Load Positive (EB)	RRE	B340	ac
LPXR	R_1,R_2	Load Positive (EH)	RRE	B360	ac
LR	R_1,R_2	Load	RR	18	
LRA	$R_1,D_2(X_2,B_2)$	Load Real Address	RX	B1	pc
LRDR	R_1,R_2	Load Rounded (EH/LH)	RR	25	
LRER	R_1,R_2	Load Rounded (LH/SH)	RR	35	
LTDBR	R_1,R_2	Load and Test (LB)	RRE	B312	ac
LTDR	R_1,R_2	Load and Test (LH)	RR	22	c
LTEBR	R_1,R_2	Load and Test (SB)	RRE	B302	ac
LTER	R_1,R_2	Load and Test (SH)	RR	32	c
LTR	R_1,R_2	Load and Test	RR	12	c
LTXBR	R_1,R_2	Load and Test (EB)	RRE	B342	ac
LTXR	R_1,R_2	Load and Test (EH)	RRE	B362	ac
LURA	R_1,R_2	Load Using Real Address	RRE	B24B	p
LXD	$R_1,D_2(X_2,B_2)$	Load Lengthened (LH/EH)	RXE	ED25	a
LXDB	$R_1,D_2(X_2,B_2)$	Load Lengthened (LB/EB)	RXE	ED05	a
LXDBR	R_1,R_2	Load Lengthened (LB/EB)	RRE	B305	a
LXDR	R_1,R_2	Load Lengthened (LH/EH)	RRE	B325	a
LXE	$R_1,D_2(X_2,B_2)$	Load Lengthened (SH/EH)	RXE	ED26	a
LXEB	$R_1,D_2(X_2,B_2)$	Load Lengthened (SB/EB)	RXE	ED06	a
LXEBR	R_1,R_2	Load Lengthened (SB/EB)	RRE	B306	a
LXER	R_1,R_2	Load Lengthened (SH/EH)	RRE	B326	a
LXR	R_1,R_2	Load (E)	RRE	B365	a
LZDR	R_1,R_2	Load Zero (L)	RRE	B375	a
LZER	R_1,R_2	Load Zero (S)	RRE	B374	a
LZXR	R_1,R_2	Load Zero (E)	RRE	B376	a
M	$R_1,D_2(X_2,B_2)$	Multiply	RX	5C	
MADB	$R_1,R_3,D_2(X_2,B_2)$	Multiply and Add (LB)	RXF	ED1E	a
MADBR	R_1,R_3,R_2	Multiply and Add (LB)	RRF	B31E	a
MAEB	$R_1,R_3,D_2(X_2,B_2)$	Multiply and Add (SB)	RXF	ED0E	a
MAEBR	R_1,R_3,R_2	Multiply and Add (SB)	RRF	B30E	a
MC	$D_1(B_1),I_2$	Monitor Call	SI	AF	
MD	$R_1,D_2(X_2,B_2)$	Multiply (LH)	RX	6C	
MDB	$R_1,D_2(X_2,B_2)$	Multiply (LB)	RXE	ED1C	a
MDBR	R_1,R_2	Multiply (LB)	RRE	B31C	a
MDE	$R_1,D_2(X_2,B_2)$	Multiply (SH/LH)	RX	7C	

Mnemonic	Operands	Name	Format	Op Code	Class & Notes
MDEB	$R_1,D_2(X_2,B_2)$	Multiply (SB/LB)	RXE	ED0C	a
MDEBR	R_1,R_2	Multiply (SB/LB)	RRE	B30C	a
MDER	R_1,R_2	Multiply (SH/LH)	RR	3C	
MDR	R_1,R_2	Multiply (LH)	RR	2C	
ME	$R_1,D_2(X_2,B_2)$	Multiply (SH/LH)	RX	7C	
MEE	$R_1,D_2(X_2,B_2)$	Multiply (SH)	RXE	ED37	a
MEEB	$R_1,D_2(X_2,B_2)$	Multiply (SB)	RXE	ED17	a
MEEBR	R_1,R_2	Multiply (SB)	RRE	B317	a
MEER	R_1,R_2	Multiply (SH)	RRE	B337	a
MER	R_1,R_2	Multiply (SH/LH)	RR	3C	
MH	$R_1,D_2(X_2,B_2)$	Multiply Halfword	RX	4C	
MHI	R_1,I_2	Multiply Halfword Immediate	RI	A7C	
MP	$D_1(L_1,B_1),D_2(L_2,B_2)$	Multiply Decimal	SS	FC	
MR	R_1,R_2	Multiply	RR	1C	
MS	$R_1,D_2(X_2,B_2)$	Multiply Single	RX	71	
MSDB	$R_1,R_3,D_2(X_2,B_2)$	Multiply and Subtract (LB)	RXF	ED1F	a
MSDBR	R_1,R_3,R_2	Multiply and Subtract (LB)	RRF	B31F	a
MSEB	$R_1,R_3,D_2(X_2,B_2)$	Multiply and Subtract (SB)	RXF	ED0F	a
MSEBR	R_1,R_3,R_2	Multiply and Subtract (SB)	RRF	B30F	a
MSR	R_1,R_2	Multiply Single	RRE	B252	
MSCH	$D_2(B_2)$	Modify Subchannel	S	B232	pc
MSTA	R_1	Modify Stacked State	RRE	B247	
MVC	$D_1(L,B_1),D_2(B_2)$	Move	SS	D2	
MVCDK	$D_1(B_1),D_2(B_2)$	Move with Destination key	SSE	E50F	q
MVCIN	$D_1(L,B_1),D_2(B_2)$	Move Inverse	SS	E8	
MVCK	$D_1(R_1,B_1),D_2(B_2),R_3$	Move with Key	SS	D9	qc
MVCL	R_1,R_2	Move Long	RR	0E	ic
MVCLE	$R_1,R_3,D_2(B_2)$	Move Long Extended	RS	A8	c
MVCP	$D_1(R_1B_1),D_2(B_2),R_3$	Move to Primary	SS	DA	qc
MVCS	$D_1(R_1B_1),D_2(B_2),R_3$	Move to Secondary	SS	DB	qc
MVCSK	$D_1(B_1),D_2(B_2)$	Move with Source Key	SSE	E50E	q
MVI	$D_1(B_1),I_2$	Move	SI	92	
MVN	$D_1(L,B_1),D_2(B_2)$	Move Numerics	SS	D1	
MVO	$D_1(L_1,B_1),D_2(L_2,B_2)$	Move with Offset	SS	F1	
MVPG	R_1,R_2	Move Page	RRE	B254	qc
MVST	R_1,R_2	Move String	RRE	B255	c
MVZ	$D_1(L,B_1),D_2(B_2)$	Move Zones	SS	D3	
MXBR	R_1,R_2	Multiply (EB)	RRE	B34C	a
MXD	$R_1,D_2(X_2,B_2)$	Multiply (LH/EH)	RX	67	
MXDB	$R_1,D_2(X_2,B_2)$	Multiply (LB/EB)	RXE	ED07	a
MXDBR	R_1,R_2	Multiply (LB/EB)	RRE	B307	a
MXDR	R_1,R_2	Multiply (LH/EH)	RR	27	
MXR	R_1,R_2	Multiply (EH)	RR	26	
N	$R_1,D_2(X_2,B_2)$	AND	RX	54	c
NC	$D_1(L,B_1),D_2(B_2)$	AND	SS	D4	c
NI	$D_1(B_1),I_2$	AND	SI	94	c
NR	R_1,R_2	AND	RR	14	c
O	$R_1,D_2(X_2,B_2)$	OR	RX	56	c
OC	$D_1(L,B_1),D_2(B_2)$	OR	SS	D6	c
OI	$D_1(B_1),I_2$	OR	SI	96	c
OR	R_1,R_2	OR	RR	16	c
PACK	$D_1(L_1,B_1),D_2(L_2,B_2)$	Pack	SS	F2	
PALB		Purge ALB	RRE	B248	p
PC	$D_2(B_2)$	Program Call	S	B218	q
PCF	$D_2(B_2)$	Program Call Fast	S	B218	q
PLO	$R_1,D_2(B_2),R_3,D_4(B_4)$	Perform Locked Operation	SS	EE	c
PR		Program Return	E	0101	u
PT	R_1,R_2	Program Transfer	RRE	B228	q
PTLB		Purge TLB	S	B20D	p
RCHP		Reset Channel Path	S	B23B	pc
RP	$D_2(B_2)$	Resume Program	S	B277	qn
RRBE	R_1,R_2	Reset Reference Bit Extended	RRE	B22A	pc
RSCH		Resume Subchannel	S	B238	pc
S	$R_1,D_2(X_2,B_2)$	Subtract	RX	5B	c
SAC	$D_2(B_2)$	Set Address Space Control	S	B219	q
SACF	$D_2(B_2)$	Set Address Space Control Fast	S	B279	q
SAL		Set Address Limit	S	B237	p
SAR	R_1,R_2	Set Access	RRE	B24E	
SCHM		Set Channel Monitor	S	B23C	p
SCK	$D_2(B_2)$	Set Clock	S	B204	pc
SCKC	$D_2(B_2)$	Set Clock Comparator	S	B206	p
SCKPF		Set Clock Programmable Field	E	0107	p
SD	$R_1,D_2(X_2,B_2)$	Subtract Normalized (LH)	RX	6B	c
SDB	$R_1,D_2(X_2,B_2)$	Subtract (LB)	RXE	ED1B	ac
SDBR	R_1,R_2	Subtract (LB)	RRE	B31B	ac
SDR	R_1,R_2	Subtract Normalized (LH)	RR	2B	c
SE	$R_1,D_2(X_2,B_2)$	Subtract Normalized (SH)	RX	7B	c
SEB	$R_1,D_2(X_2,B_2)$	Subtract (SB)	RXE	ED0B	ac
SEBR	R_1,R_2	Subtract (SB)	RRE	B30B	ac
SER	R_1,R_2	Subtract Normalized (SH)	RR	3B	c
SFPC	R_1	Set FPC	RRE	B384	a
SH	$R_1,D_2(X_2,B_2)$	Subtract Halfword	RX	4B	c
SIE	$D_2(B_2)$	Start Interpretive Execution	S	B214	ip
SIGP	$R_1,R_3,D_2(B_2)$	Signal Processor	RS	AE	pc
SL	$R_1,D_2(X_2,B_2)$	Subtract Logical	RX	5F	c

Mnemonic	Operands	Name	Format	Op Code	Class & Notes
SLA	$R_1,D_2(B_2)$	Shift Left Single	RS	8B	c
SLDA	$R_1,D_2(B_2)$	Shift Left Double	RS	8F	c
SLDL	$R_1,D_2(B_2)$	Shift Left Double Logical	RS	8D	
SLL	$R_1,D_2(B_2)$	Shift Left Single Logical	RS	89	
SLR	R_1,R_2	Subtract Logical	RR	1F	c
SP	$D_1(L_1,B_1),D_2(L_2,B_2)$	Subtract Decimal	SS	FB	c
SPKA	$D_2(B_2)$	Set PSW Key from Address	S	B20A	q
SPM	R_1	Set Program Mask	RR	04	n
SPT	$D_2(B_2)$	Set CPU Timer	S	B208	p
SPX	$D_2(B_2)$	Set Prefix	S	B210	p
SQD	$R_1,D_2(X_2,B_2)$	Square Root (LH)	RXE	ED35	a
SQDB	$R_1,D_2(X_2,B_2)$	Square Root (LB)	RXE	ED15	a
SQDBR	R_1,R_2	Square Root (LB)	RRE	B315	a
SQDR	R_1,R_2	Square Root (LH)	RRE	B244	
SQE	$R_1,D_2(X_2,B_2)$	Square Root (SH)	RXE	ED34	a
SQEB	$R_1,D_2(X_2,B_2)$	Square Root (SB)	RXE	ED14	a
SQEBR	R_1,R_2	Square Root (SB)	RRE	B314	a
SQER	R_1,R_2	Square Root (SH)	RRE	B245	
SQXR	R_1,R_2	Square Root (EH)	RRE	B336	a
SQXBR	R_1,R_2	Square Root (EB)	RRE	B316	a
SR	R_1,R_2	Subtract	RR	1B	c
SRA	$R_1,D_2(B_2)$	Shift Right Single	RS	8A	c
SRDA	$R_1,D_2(B_2)$	Shift Right Double	RS	8E	c
SRDL	$R_1,D_2(B_2)$	Shift Right Double Logical	RS	8C	
SRL	$R_1,D_2(B_2)$	Shift Right Single Logical	RS	88	
SRNM	$D_2(B_2)$	Set Rounding Mode	S	B299	a
SRP	$D_1(L_1,B_1),D_2(B_2),I_3$	Shift and Round Decimal	SS	F0	c
SRST	R_1,R_2	Search String	RRE	B25E	c
SSAR	R_1	Set Secondary ASN	RRE	B225	
SSCH	$D_2(B_2)$	Start Subchannel	S	B233	pc
SSKE	R_1,R_2	Set Storage Key Extended	RRE	B22B	p
SSM	$D_2(B_2)$	Set System Mask	S	80	p
ST	$R_1,D_2(X_2,B_2)$	Store	RX	50	
STAM	$R_1,R_3,D_2(B_2)$	Store Access Multiple	RS	9B	
STAP	$D_2(B_2)$	Store CPU Address	S	B212	p
STC	$R_1,D_2(X_2,B_2)$	Store Character	RX	42	
STCK	$D_2(B_2)$	Store Clock	S	B205	c
STCKC	$D_2(B_2)$	Store Clock Comparator	S	B207	p
STCKE	$D_2(B_2)$	Store Clock Extended	S	B278	c
STCM	$R_1,M_3,D_2(B_2)$	Store Characters under Mask	RS	BE	
STCPS	$D_2(B_2)$	Store Channel Path Status	S	B23A	p
STCRW	$D_2(B_2)$	Store Channel Report Word	S	B239	pc
STCTL	$R_1,R_3,D_2(B_2)$	Store Control	RS	B6	p
STD	$R_1,D_2(X_2,B_2)$	Store (L)	RX	60	
STE	$R_1,D_2(X_2,B_2)$	Store (S)	RX	70	
STFPC	$D_2(B_2)$	Store FPC	S	B29C	a
STH	$R_1,D_2(X_2,B_2)$	Store Halfword	RX	40	
STIDP	$D_2(B_2)$	Store CPU ID	S	B202	p
STM	$R_1,R_3,D_2(B_2)$	Store Multiple	RS	90	
STNSM	$D_1(B_1),I_2$	Store Then AND System Mask	SI	AC	p
STOSM	$D_1(B_1),I_2$	Store Then OR System Mask	SI	AD	p
STPT	$D_2(B_2)$	Store CPU Timer	S	B209	p
STPX	$D_2(B_2)$	Store Prefix	S	B211	p
STSI	$D_2(B_2)$	Store System Information	S	B27D	pc
STSCH	$D_2(B_2)$	Store Subchannel	S	B234	pc
STURA	R_1,R_2	Store Using Real Address	RRE	B246	p
SU	$R_1,D_2(X_2,B_2)$	Subtract Unnormalized (SH)	RX	7F	c
SUR	R_1,R_2	Subtract Unnormalized (SH)	RR	3F	c
SVC	I	Supervisor Call	RR	0A	
SW	$R_1,D_2(X_2,B_2)$	Subtract Unnormalized (LH)	RX	6F	c
SWR	R_1,R_2	Subtract Unnormalized (LH)	RR	2F	c
SXBR	R_1,D_2	Subtract (EB)	RRE	B34B	ac
SXR	R_1,D_2	Subtract Normalized (EH)	RR	37	c
TAR	R_1,R_2	Test Access	RRE	B24C	c
TB	R_1,R_2	Test Block	RRE	B22C	ipc
TBDR	R_1,M_3,R_2	Convert HFP to BFP (LH/LB)	RRF	B351	ac
TBEDR	R_1,M_3,R_2	Convert HFP to BFP (LH/SB)	RRF	B350	ac
TCDB	$R_1,D_2(X_2,B_2)$	Test Data Class (LB)	RXE	ED11	ac
TCEB	$R_1,D_2(X_2,B_2)$	Test Data Class (SB)	RXE	ED10	ac
TCXB	$R_1,D_2(X_2,B_2)$	Test Data Class (EB)	RXE	ED12	ac
THDER	R_1,R_2	Convert BFP to HFP (SB/LH)	RRE	B358	ac
THDR	R_1,R_2	Convert BFP to HFP (LB/LH)	RRE	B359	ac
TM	$D_1(B_1),I_2$	Test under Mask	SI	91	c
TMH	R_1,I_2	Test under Mask High	RI	A70	c
TML	R_1,I_2	Test under Mask Low	RI	A71	c
TPI	$D_2(B_2)$	Test Pending Interruption	S	B236	pc
TPROT	$D_1(B_1),D_2(B_2)$	Test Protection	SSE	E501	pc
TR	$D_1(L,B_1),D_2(B_2)$	Translate	SS	DC	
TRACE	$R_1,R_3,D_2(B_2)$	Trace	RS	99	p
TRAP2		Trap	E	01FF	

Mnemonic	Operands	Name	Format	Op Code	Class & Notes
TRAP4	$D_2(B_2)$	Trap	S	B2FF	
TRE	R&sub1,R_2	Translate Extended	RRE	B2A5	c
TRT	$D_1(L,B_1),D_2(B_2)$	Translate and Test	SS	DD	c
TS	$D_2(B_2)$	Test and Set	S	93	c
TSCH	$D_2(B_2)$	Test Subchannel	S	B235	pc
UNPK	$D_1(L_1,B_1),D_2(L_2,B_2)$	Unpack	SS	F3	
UPT		Update Tree	E	0102	ic
VA	$VR_1,VR_3,RS_2(RT_2)$	Add	VST	A420	IM
VACD	$VR_1,RS_2(RT_2)$	Accumulate (LH)	VST	A417	IM
VACDR	VR_1,VR_2	Accumulate (LH)	VV	A517	IM
VACE	$VR_1,RS_2(RT_2)$	Accumulate (SH/LH)	VST	A407	IM
VACER	VR_1,VR_2	Accumulate (SH/LH)	VV	A507	IM
VACRS	$D_2(B_2)$	Restore VAC	S	A6CB	NO p
VACSV	$D_2(B_2)$	Save VAC	S	A6CA	NO p
VAD	$VR_1,VR_3,RS_2(RT_2)$	Add (LH)	VST	A410	IM
VADQ	VR_1,FR_3,VR_2	Add (LH)	QV	A590	IM
VADR	VR_1,VR_3,VR_2	Add (LH)	VV	A510	IM
VADS	$VR_1,FR_3,RS_2(RT_2)$	Add (LH)	QST	A490	IM
VAE	$VR_1,VR_3,RS_2(RT_2)$	Add (SH)	VST	A400	IM
VAEQ	VR_1,FR_3,VR_2	Add (SH)	QV	A580	IM
VAER	VR_1,VR_3,VR_2	Add (SH)	VV	A500	IM
VAES	$VR_1,FR_3,RS_2(RT_2)$	Add (SH)	QST	A480	IM
VAQ	VR_1,GR_3,VR_2	Add	QV	A5A0	IM
VAR	VR_1,VR_3,VR_2	Add	VV	A520	IM
VAS	$VR_1,GR_3,RS_2(RT_2)$	Add	QST	A4A0	IM
VC	$M_1,VR_3,RS_2(RT_2)$	Compare	VST	A428	IC
VCD	$M_1,VR_3,RS_2(RT_2)$	Compare (LH)	VST	A418	IC
VCDQ	M_1,FR_3,VR_2	Compare (LH)	QV	A598	IC
VCDR	M_1,VR_3,VR_2	Compare (LH)	VV	A518	IC
VCDS	$M_1,FR_3,RS_2(RT_2)$	Compare (LH)	QST	A498	IC
VCE	$M_1,VR_3,RS_2(RT_2)$	Compare (SH)	VST	A408	IC
VCEQ	M_1,FR_3,VR_2	Compare (SH)	QV	A588	IC
VCER	M_1,VR_3,VR_2	Compare (SH)	VV	A508	IC
VCES	$M_1,FR_3,RS_2(RT_2)$	Compare (SH)	QST	A488	IC
VCOVM	GR_1	Count Ones in VMR	RRE	A643	NC c
VCQ	M_1,GR_3,VR_2	Compare	QV	A5A8	IC
VCR	M_1,VR_3,VR_2	Compare	VV	A528	IC
VCS	$M_1,GR_3,RS_2(RT_2)$	Compare	QST	A4A8	IC
VCVM		Complement VMR	RRE	A641	NC
VCZVM	GR_1	Count Left Zeros in VMR	RRE	A642	NC c
VDD	$VR_1,VR_3,RS_2(RT_2)$	Divide (LH)	VST	A413	IM
VDDQ	VR_1,FR_3,VR_2	Divide (LH)	QV	A593	IM
VDDR	VR_1,VR_3,VR_2	Divide (LH)	VV	A513	IM
VDDS	$VR_1,FR_3,RS_2(RT_2)$	Divide (LH)	QST	A493	IM
VDE	$VR_1,VR_3,RS_2(RT_2)$	Divide (SH)	VST	A403	IM
VDEQ	VR_1,FR_3,VR_2	Divide (SH)	QV	A583	IM
VDER	VR_1,VR_3,VR_2	Divide (SH)	VV	A503	IM
VDES	$VR_1,FR_3,RS_2(RT_2)$	Divide (SH)	QST	A483	IM
VL	$VR_1,RS_2,(RT_2)$	Load	VST	A409	IC
VLBIX	$VR_1,GR_3,D_2(B_2)$	Load Bit Index	RSE	E428	IG c
VLCDR	VR_1,VR_2	Load Complement (LH)	VV	A552	IM
VLCER	VR_1,VR_2	Load Complement (SH)	VV	A542	IM
VLCR	VR_1,VR_2	Load Complement	VV	A562	IM
VLCVM	RS_2	Load VMR Complement	VS	A681	NC
VLD	$VR_1,RS_2(RT_2)$	Load (LH)	VST	A419	IC
VLDQ	VR_1,FR_2	Load (LH)	QV	A599	IC
VLDR	VR_1,VR_2	Load (LH)	VV	A519	IC
VLE	$VR_1,RS_2(RT_2)$	Load (SH)	VST	A409	IC
VLEL	VR_1,GR_3,GR_2	Load Element	VR	A628	N1
VLELD	VR_1,FR_3,GR_2	Load Element (LH)	VR	A618	N1
VLELD	VR_1,FR_3,GR_2	Load Element (SH)	VR	A608	N1
VLEQ	VR_1,FR_2	Load (SH)	QV	A589	IC
VLER	VR_1,FR_2	Load (SH)	VV	A509	IC
VLH	$VR_1,RS_2(RT_2)$	Load Halfword	VST	A429	IC
VLI	$VR_1,VR_3,D_2(B_2)$	Load Indirect	RSE	E400	IC
VLID	$VR_1,VR_3,D_2(B_2)$	Load Indirect (LH)	RSE	E410	IC
VLIE	$VR_1,VR_3,D_2(B_2)$	Load Indirect (SH)	RSE	E400	IC
VLINT	$VR_1,RS_2(RT_2)$	Load Integer Vector	VST	A42A	IC
VLM	$VR_1,RS_2(RT_2)$	Load Matched	VST	A40A	IC
VLMD	$VR_1,RS_2(RT_2)$	Load Matched (LH)	VST	A41A	IC
VLMDQ	VR_1,FR_2	Load Matched (LH)	QV	A59A	IC
VLMDR	VR_1,VR_2	Load Matched (LH)	VV	A51A	IC
VLME	$VR_1,RS_2(RT_2)$	Load Matched (SH)	VST	A40A	IC
VLMEQ	VR_1,FR_2	Load Matched (SH)	QV	A58A	IC
VLMER	VR_1,VR_2	Load Matched (SH)	VV	A50A	IC
VLMQ	VR_1,GR_2	Load Matched	QV	A5AA	IC
VLMR	VR_1,VR_2	Load Matched	VV	A50A	IC
VLNDR	VR_1,VR_2	Load Negative (LH)	VV	A551	IM
VLNER	VR_1,VR_2	Load Negative (SH)	VV	A541	IM
VLNR	VR_1,VR_2	Load Negative	VV	A561	IM
VLPDR	VR_1,VR_2	Load Positive (LH)	VV	A550	IM
VLPER	VR_1,VR_2	Load Positive (SH)	VV	A540	IM
VLPR	VR_1,VR_2	Load Positive (SH)	VV	A560	IM
VLQ	VR_1,GR_2	Load	QV	A5A9	IC
VLR	VR_1,VR_2	Load	VV	A509	IC
VLVCA	$D_2(B_2)$	Load VCT from Address	S	A6C4	NO c
VLVCU	GR_1	Load VCT and Update	RRE	A645	NO c
VLVM	RS_2	Load VMR	VS	A680	NO
VLVXA	$D_2(B_2)$	Load VIX from Address	S	A6C7	NO c

Machine Instructions by Mnemonic (Cont'd)

Mnemonic	Operands	Name	Format	Op Code	Class & Notes
VLY	VR_1,$RS_2(RT_2)$	Load Expanded	VST	A40B	IC
VLYD	VR_1,$RS_2(RT_2)$	Load Expanded (LH)	VST	A41B	IC
VLYE	VR_1,$RS_2(RT_2)$	Load Expanded (SH)	VST	A40B	IC
VLZDR	VR_1	Load Zero (LH)	VV	A51B	IC
VLZER	VR_1	Load Zero (SH)	VV	A50B	IC
VLZR	VR_1	Load Zero	VV	A50B	IC
VM	VR_1,VR_3,$RS_2(RT_2)$	Multiply	VST	A422	IM
VMAD	VR_1,VR_3,$RS_2(RT_2)$	Multiply and Add (LH)	VST	A414	IM
VMADQ	VR_1,FR_3,VR_2	Multiply and Add (LH)	QV	A594	IM
VMADS	VR_1,FR_3,$RS_2(RT_2)$	Multiply and Add (LH)	QST	A494	IM
VMAE	VR_1,VR_3,$RS_2(RT_2)$	Multiply and Add (SH/LH)	VST	A404	IM
VMAEQ	VR_1,FR_3,VR_2	Multiply and Add (SH/LH)	QV	A584	IM
VMAES	VR_1,FR_3,$RS_2(RT_2)$	Multiply and Add (SH/LH)	QST	A484	IM
VMCD	VR_1,VR_3,$RS_2(RT_2)$	Multiply and Accumulate (LH)	VST	A416	IM
VMCDR	VR_1,VR_3,VR_2	Multiply and Accumulate (LH)	VV	A516	IM
VMCE	VR_1,VR_3,$RS_2(RT_2)$	Multiply and Accumulate (SH/LH)	VST	A406	IM
VMCER	VR_1,VR_3,VR_2	Multiply and Accumulate (SH/LH)	VV	A506	IM
VMD	VR_1,VR_3,$RS_2(RT_2)$	Multiply (LH)	VST	A412	IM
VMDQ	VR_1,FR_3,VR_2	Multiply (LH)	QV	A592	IM
VMDR	VR_1,VR_3,VR_2	Multiply (LH)	VV	A512	IM
VMDS	VR_1,FR_3,$RS_2(RT_2)$	Multiply (LH)	QST	A492	IM
VME	VR_1,VR_3,$RS_2(RT_2)$	Multiply (SH/LH)	VST	A402	IM
VMEQ	VR_1,FR_3,VR_2	Multiply (SH/LH)	QV	A582	IM
VMER	VR_1,VR_3,VR_2	Multiply (SH/LH)	VV	A502	IM
VMES	VR_1,FR_3,$RS_2(RT_2)$	Multiply (SH/LH)	QST	A482	IM
VMNSD	VR_1,FR_3,GR_2	Minimum Signed (LH)	VR	A611	IM
VMNSE	VR_1,FR_3,GR_2	Minimum Signed (SH)	VR	A601	IM
VMQ	VR_1,GR_3,VR_2	Multiply	QV	A5A2	IM
VMR	VR_1,VR_3,VR_2	Multiply	VV	A522	IM
VMRRS	$D_2(B_2)$	Restore VMR	S	A6C3	NZ
VMRSV	$D_2(B_2)$	Save VMR	S	A6C1	NZ
VMS	VR_1,GR_3,$RS_2(RT_2)$	Multiply	QST	A4A2	IM
VMSD	VR_1,VR_3,$RS_2(RT_2)$	Multiply and Subtract (LH)	VST	A415	IM
VMSDQ	VR_1,FR_3,VR_2	Multiply and Subtract (LH)	QV	A595	IM
VMSDS	VR_1,FR_3,$RS_2(RT_2)$	Multiply and Subtract (LH)	QST	A495	IM
VMSE	VR_1,VR_3,$RS_2(RT_2)$	Multiply and Subtract (SH/LH)	VST	A405	IM
VMSEQ	VR_1,FR_3,VR_2	Multiply and Subtract (SH/LH)	QV	A585	IM
VMSES	VR_1,FR_3,$RS_2(RT_2)$	Multiply and Subtract (SH/LH)	QST	A485	IM
VMXAD	VR_1,FR_3,GR_2	Maximum Absolute (LH)	VR	A612	IM
VMXAE	VR_1,FR_3,GR_2	Maximum Absolute (SH)	VR	A602	IM
VMXSD	VR_1,FR_3,GR_2	Maximum Signed (LH)	VR	A610	IM
VMXSE	VR_1,FR_3,GR_2	Maximum Signed (SH)	VR	A600	IM
VN	VR_1,VR_3,$RS_2(RT_2)$	AND	VST	A424	IM
VNQ	VR_1,GR_3,VR_2	AND	QV	A5A4	IM
VNR	VR_1,VR_3,VR_2	AND	VV	A524	IM
VNS	VR_1,GR_3,$RS_2(RT_2)$	AND	QST	A4A4	IM
VNVM	RS_2	AND to VMR	VS	A684	NC
VO	VR_1,VR_3,$RS_2(RT_2)$	OR	VST	A425	IM
VOQ	VR_1,VR_3,VR_2	OR	QV	A5A5	IM
VOR	VR_1,VR_3,VR_2	OR	VV	A525	IM
VOS	VR_1,GR_3,$RS_2(RT_2)$	OR	QST	A4A5	IM
VOVM	RS_2	OR to VMR	VS	A685	NC
VRCL	$D_2(B_2)$	Clear VR	S	A6C5	IZ
VRRS	GR_1	Restore VR	RRE	A648	IZ xc
VRSV	GR_1	Save VR	RRE	A64A	IZ c
VRSVC	GR_1	Save Changed VR	RRE	A649	IZ pc
VS	VR_1,VR_3,$RS_2(RT_2)$	Subtract	VST	A421	IM
VSD	VR_1,VR_3,$RS_2(RT_2)$	Subtract (LH)	VST	A411	IM
VSDQ	VR_1,FR_3,VR_2	Subtract (LH)	QV	A591	IM
VSDR	VR_1,VR_3,VR_2	Subtract (LH)	VV	A511	IM
VSDS	VR_1,FR_3,$RS_2(RT_2)$	Subtract (LH)	QST	A491	IM
VSE	VR_1,VR_3,$RS_2(RT_2)$	Subtract (SH)	VST	A401	IM
VSEQ	VR_1,FR_3,VR_2	Subtract (SH)	QV	A581	IM
VSER	VR_1,VR_3,VR_2	Subtract (SH)	VV	A501	IM
VSES	VR_1,FR_3,$RS_2(RT_2)$	Subtract (SH)	QST	A481	IM
VSLL	VR_1,VR_3,$D_2(B_2)$	Shift Left Single Logical	RSE	E425	IM
VSPSD	VR_1,FR_2	Sum Partial Sums (LH)	VR	A61A	IP
VSQ	VR_1,GR_3,VR_2	Subtract	QV	A5A1	IM
VSQD	VR_1,$RS_2(RT_2)$	Square Root (LH)	VST	A453	IM
VSQDR	VR_1,VR_2	Square Root (SH)	VV	A553	IM
VSQE	VR_1,$RS_2(RT_2)$	Square Root (LH)	VST	A443	IM
VSQER	VR_1,VR_2	Square Root (SH)	VV	A543	IM
VSR	VR_1,VR_3,VR_2	Subtract	VV	A521	IM
VSRL	VR_1,VR_3,$D_2(B_2)$	Shift Right Single Logical	RSE	E424	IM
VSRRS	$D_2(B_2)$	Restore VSR	S	A6C2	IZ x
VSRSV	$D_2(B_2)$	Save VSR	S	A6C0	NO x
VSS	VR_1,GR_3,$RS_2(RT_2)$	Subtract	QST	A4A1	IM
VST	VR_1,$RS_2(RT_2)$	Store	VST	A40D	IC
VSTD	VR_1,$RS_2(RT_2)$	Store (LH)	VST	A41D	IC
VSTE	VR_1,$RS_2(RT_2)$	Store (SH)	VST	A40D	IC
VSTH	VR_1,$RS_2(RT_2)$	Store Halword	VST	A42D	IC
VSTI	VR_1,$VR_3D_2(B_2)$	Store Indirect	RSE	E401	IC
VSTID	VR_1,$VR_3D_2(B_2)$	Store Indirect (LH)	RSE	E411	IC
VSTIE	VR_1,$VR_3D_2(B_2)$	Store Indirect (SH)	RSE	E401	IC
VSTK	VR_1,$RS_2(RT_2)$	Store Compressed	VST	A40F	IC
VSTKD	VR_1,$RS_2(RT_2)$	Store Compressed (LH)	VST	A41F	IC
VSTKE	VR_1,$RS_2(RT_2)$	Store Compressed (SH)	VST	A40F	IC

Mnemonic	Operands	Name	Format	Op Code	Class & Notes
VSTM	$VR_1,RS_2(RT_2)$	Store Matched	VST	A40E	IC
VSTMD	$VR_1,RS_2(RT_2)$	Store Matched (LH)	VST	A41E	IC
VSTME	$VR_1,RS_2(RT_2)$	Store Matched (SH)	VST	A40E	IC
VSTVM	RS_2	Store VMR	VS	A682	NC
VSTVP	$D_2(B_2)$	Store Vector Parameters	S	A6C8	NO
VSVMM	$D_2(B_2)$	Set Vector Mask Mode	S	A6C6	NO
VTAD	$VR_1,VR_3,RS_2(RT_2)$	Multiply then ADD (LH)	VST	A454	IM
VTAE	$VR_1,VR_3,RS_2(RT_2)$	Multiply then ADD (SH/LH)	VST	A444	IM
VTSD	$VR_1,VR_3,RS_2(RT_2)$	Multiply then Subtract (LH)	VST	A455	IM
VTSE	$VR_1,VR_3,RS_2(RT_2)$	Multiply then Subtract (SH/LH)	VST	A445	IM
VTVM		Test VMR	RRE	A640	NC c
VX	$VR_1,VR_3,RS_2(RT_2)$	Exclusive OR	VST	A426	IM
VXEL	VR_1,GR_3,GR_2	Extract Element	VR	A629	N1
VXELD	VR_1,FR_3,GR_2	Extract Element (LH)	VR	A619	N1
VXELE	VR_1,FR_3,GR_2	Extract Element (SH)	VR	A609	N1
VXQ	VR_1,GR_3,VR_2	Exclusive OR	QV	A5A6	IM
VXR	VR_1,VR_3,VR_2	Exclusive OR	VV	A526	IM
VXS	$VR_1,GR_3,RS_2(RT_2)$	Exclusive OR	QST	A4A6	IM
VXVC	GR_1	Extract VCT	RRE	A644	NO
VXVM	RS_2	Exclusive OR to VMR	VS	A686	NC
VXVMM	GR_1	Extract Vector Mask Mode	RRE	A646	NO
VZPSD	VR_1	Zero Partial Sums (LH)	VR	A61B	IP
X	$R_1,D_2(X_2,B_2)$	Exclusive OR	RX	57	c
XC	$D_1(L,B_1)D_2(B_2)$	Exclusive OR	SS	D7	c
XI	$D_1(B_1)I_2$	Exclusive OR	SI	97	c
XR	R_1,R_2	Exclusive OR	RR	17	c
ZAP	$D_1(L_1,B_1),D_2(L_2,B_2)$	Zero and Add	SS	F8	c
---	Model-dependent	Diagnose	--	83	pu

Floating-point operand lengths and types:

(x)	Source and result
(x/y)	Source (x) and result (y)
E	Extended (binary or hex)
EB	Extended binary
EH	Extended hex
L	Long (binary or hex)
LB	Long binary
LH	Long hex
S	Short (binary or hex)
SB	Short binary
SH	Short hex
32	32-bit integer

Notes:

a	Additional-floating-point instruction
c	Condition code set
i	Interruptible instruction
n	New condition code loaded
p	Privileged instruction
q	Semiprivileged instruction
u	Condition code is unpredictable or may be set

Class (for instructions subject to vector-control bit, CR 0 bit 14)

IC	Interruptible; (VCT – VIX) elements processed
IG	Interruptible; either (bit count in a general register) elements or (section-size – VIX) elements processed, whichever is fewer
IM	Interruptible; (VCT – VIX) elements processed, vector-mask mode
IP	Interruptible; (partial-sum-number – VIX) elements processed
IZ	Interruptible; (section-size) elements processed
NC	Not interruptible; (VCT) elements processed
NZ	Not interruptible; (section-size) elements processed
N0	Not interruptible; no elements processed (VSR/VAC housekeeping)
N1	Not interruptible; one element processed

Machine Instructions by Operation Code

Op Code	Mne-monic
0101	PR
0102	UPT
0107	SCKPF
01FF	TRAP2
04	SPM
05	BALR
06	BCTR
07	BCR
0A	SVC
0B	BSM
0C	BASSM
0D	BASR
0E	MVCL
0F	CLCL
10	LPR
11	LNR
12	LTR
13	LCR
14	NR
15	CLR
16	OR
17	XR
18	LR
19	CR
1A	AR
1B	SR
1C	MR
1D	DR
1E	ALR
1F	SLR
20	LPDR
21	LNDR
22	LTDR
23	LCDR
24	HDR
25	LDXR
25	LRDR
26	MXR
27	MXDR
28	LDR
29	CDR
2A	ADR
2B	SDR
2C	MDR
2D	DDR
2E	AWR
2F	SWR
30	LPER
31	LNER
32	LTER
33	LCER
34	HER
25	LEDR
35	LRER
36	AXR
37	SXR
38	LER
39	CER
3A	AER
3B	SER
3C	MDER
3C	MER
3D	DER
3E	AUR
3F	SUR
40	STH
41	LA
42	STC
43	IC
44	EX
45	BAL
46	BCT
47	BC
48	LH
49	CH
4A	AH
4B	SH
4C	MH
4D	BAS
4E	CVD
4F	CVB
50	ST
51	LAE
54	N
55	CL
56	O
57	X
58	L
59	C
5A	A
5B	S
5C	M
5D	D
5E	AL
5F	SL
60	STD
67	MXD
68	LD
69	CD
6A	AD
6B	SD
6C	MD
6D	DD
6E	AW

Op Code	Mne-monic
6F	SW
70	STE
71	MS
78	LE
79	CE
7A	AE
7B	SE
7C	MDE
7C	ME
7D	DE
7E	AU
7F	SU
80	SSM
82	LPSW
83	Diagnose
84	BRXH
85	BRXLE
86	BXH
87	BXLE
88	SRL
89	SLL
8A	SRA
8B	SLA
8C	SRDL
8D	SLDL
8E	SRDA
8F	SLDA
90	STM
91	TM
92	MVI
93	TS
94	NI
95	CLI
96	OI
97	XI
98	LM
99	TRACE
9A	LAM
9B	STAM
A400	VAE
A401	VSE
A402	VME
A403	VDE
A404	VMAE
A405	VMSE
A406	VMCE
A407	VACE
A408	VCE
A409	VL
A409	VLE
A40A	VLM
A40A	VLME
A40B	VLY
A40B	VLYE
A40D	VST
A40D	VSTE
A40E	VSTM
A40E	VSTME
A40F	VSTK
A40F	VSTKE
A410	VAD
A411	VSD
A412	VMD
A413	VDD
A414	VMAD
A415	VMSD
A416	VMCD
A417	VACD
A418	VCD
A419	VLD
A41A	VLMD
A41B	VLYD
A41D	VSTD
A41E	VSTMD
A41F	VSTKD
A420	VA
A421	VS
A422	VM
A424	VN
A425	VO
A426	VX
A428	VC
A429	VLH
A42A	VLINT
A42D	VSTH
A443	VSQE
A444	VTAE
A445	VTSE
A453	VSQD
A454	VTAD
A455	VTSD
A480	VAES
A481	VSES
A482	VMES
A483	VDES
A484	VMAES
A485	VMSES
A488	VCES
A490	VADS
A491	VSDS
A492	VMDS
A493	VDDS
A494	VMADS
A495	VMSDS

Op Code	Mne-monic
A498	VCDS
A4A0	VAS
A4A1	VSS
A4A2	VMS
A4A4	VNS
A4A5	VOS
A4A6	VXS
A4A8	VCS
A500	VAER
A501	VSER
A502	VMER
A503	VDER
A506	VMCER
A507	VACER
A508	VCER
A509	VLER
A509	VLR
A50A	VLMER
A50A	VLMR
A50B	VLZER
A50B	VLZR
A510	VADR
A511	VSDR
A512	VMDR
A513	VDDR
A516	VMCDR
A517	VACDR
A518	VCDR
A519	VLDR
A51A	VLMDR
A51B	VLZDR
A520	VAR
A521	VSR
A522	VMR
A524	VNR
A525	VOR
A526	VXR
A528	VCR
A540	VLPER
A541	VLNER
A542	VLCER
A543	VSQER
A550	VLPDR
A551	VLNDR
A552	VLCDR
A553	VSQDR
A560	VLPR
A561	VLNR
A562	VLCR
A580	VAFQ
A581	VSEQ
A582	VMEQ
A583	VDEQ
A584	VMAEQ
A585	VMSEQ
A588	VCEQ
A589	VLEQ
A58A	VLMEQ
A590	VADQ
A591	VSDQ
A592	VMDQ
A593	VDDQ

Machine Instructions by Operation Code (Cont'd)

Op Code	Mne-monic
A594	VMADQ
A595	VMSDQ
A598	VCDQ
A599	VLDQ
A59A	VLMDQ
A5A0	VAQ
A5A1	VSQ
A5A2	VMQ
A5A4	VNQ
A5A5	VOQ
A5A6	VXQ
A5A8	VCQ
A5A9	VLQ
A5AA	VLMQ
A600	VMXSE
A601	VMNSE
A602	VMXAE
A608	VLELE
A609	VXELE
A610	VMXSD
A611	VMNSD
A612	VMXAD
A618	VLELD
A619	VXELD
A61A	VSPSD
A61B	VZPSD
A628	VLEL
A629	VXEL
A640	VTVM
A641	VCVM
A642	VCZVM
A643	VCOVM
A644	VXVC
A645	VLVCU
A646	VXVMM
A648	VRRS
A649	VRSVC
A64A	VRSV
A680	VLVM
A681	VLCVM
A682	VSTVM
A684	VNVM
A685	VOVM
A686	VXVM
A6C0	VSRSV
A6C1	VMRSV
A6C2	VSRRS
A6C3	VMRRS
A6C4	VLVCA
A6C5	VRCL
A6C6	VSVMM
A6C7	VLVXA
A6C8	VSTVP
A6CA	VACSV
A6CB	VACRS
A70	TMH
A71	TML
A74	BRC
A75	BRAS
A76	BRCT
A78	LHI
A7A	AHI
A7C	MHI
A7E	CHI
A8	MVCLE
A9	CLCLE
AC	STNSM
AD	STOSM
AE	SIGP
AF	MC
B1	LRA
B202	STIDP
B204	SCK
B205	STCK
B206	SCKC
B207	STCKC
B208	SPT
B209	STPT
B20A	SPKA
B20B	IPK
B20D	PTLB
B210	SPX
B211	STPX
B212	STAP
B214	SIE
B218	PC
B218	PCF
B219	SAC
B21A	CFC
B221	IPTE
B222	IPM
B223	IVSK
B224	IAC
B225	SSAR
B226	EPAR
B227	ESAR
B228	PT
B229	ISKE
B22A	RRBE
B22B	SSKE
B22C	TB
B22D	DXR
B230	CSCH
B231	HSCH
B232	MSCH
B233	SSCH
B234	STSCH
B235	TSCH
B236	TPI

Op Code	Mne-monic
B237	SAL
B238	RSCH
B239	STCRW
B23A	STCPS
B23B	RCHP
B23C	SCHM
B240	BAKR
B241	CKSM
B244	SQDR
B245	SQER
B246	STURA
B247	MSTA
B248	PALB
B249	EREG
B24A	ESTA
B24B	LURA
B24C	TAR
B24D	CPYA
B24E	SAR
B24F	EAR
B252	MSR
B254	MVPG
B255	MVST
B257	CUSE
B258	BSG
B25A	BSA
B25D	CLST
B25E	SRST
B263	CMPSC
B277	RP
B278	STCKE
B279	SACF
B27D	STSI
B299	SRNM
B29C	STFPC
B29D	LFPC
B2A5	TRE
B2A6	CUUTF
B2A7	CUTFU
B2A6	CUUTF
B2FF	TRAP4
B300	LPEBR
B301	LNEBR
B302	LTEBR
B303	LCEBR
B304	LDEBR
B305	LXDBR
B306	LXEBR
B307	MXDBR
B308	KEBR
B309	CEBR
B30A	AEBR
B30B	SEBR
B30C	MDEBR
B30D	DEBR
B30E	MAEBR
B30F	MSEBR
B310	LPDBR
B311	LNDBR
B312	LTDBR
B313	LCDBR
B314	SQEBR
B315	SQDBR
B316	SQXBR
B317	MEEBR
B318	KDBR
B319	CDBR
B31A	ADBR
B31B	SDBR
B31C	MDBR
B31D	DDBR
B31E	MADBR
B31F	MSDBR
B324	LDER
B325	LXDR
B326	LXER
B336	SQXR
B337	MEER
B340	LPXBR
B341	LNXBR
B342	LTXBR
B343	LCXBR
B344	LEDBR
B345	LDXBR
B346	LEXBR
B347	FIXBR
B348	KXBR
B349	CXBR
B34A	AXBR
B34B	SXBR
B34C	MXBR
B34D	DXBR
B350	TBEDR
B351	TBDR
B353	DIEBR
B357	FIEBR
B358	THDER
B359	THDR
B35B	DIDBR
B35F	FIDBR
B360	LPXR
B361	LNXR
B362	LTXR
B363	LCXR
B365	LXR
B366	LEXR
B367	FIXR
B369	CXR
B374	LZER

Op Code	Mne-monic
B375	LZDR
B376	LZXR
B377	FIER
B37F	FIDR
B384	SFPC
B38C	EFPC
B394	CEFBR
B395	CDFBR
B396	CXFBR
B398	CFEBR
B399	CFDBR
B39A	CFXBR
B3B4	CEFR
B3B5	CDFR
B3B6	CXFR
B3B8	CFER
B3B9	CFDR
B3BA	CFXR
B6	STCTL
B7	LCTL
BA	CS
BB	CDS
BD	CLM
BE	STCM
BF	ICM
D1	MVN
D2	MVC
D3	MVZ
D4	NC
D5	CLC
D6	OC
D7	XC
D9	MVCK
DA	MVCP
DB	MVCS
DC	TR
DD	TRT
DE	ED
DF	EDMK
E400	VLI
E400	VLIE
E401	VSTI
E401	VSTIE
E410	VLID
E411	VSTID
E424	VSRL
E425	VSLL
E428	VLBIX
E500	LASP
E501	TPROT
E50E	MVCSK
E50F	MVCDK
E8	MVCIN
ED04	LDEB
ED05	LXDB
ED06	LXEB
ED07	MXDB
ED08	KEB
ED09	CEB
ED0A	AEB
ED0B	SEB
ED0C	MDEB
ED0D	DEB
ED0E	MAEB
ED0F	MSEB
ED10	TCEB
ED11	TCDB
ED12	TCXB
ED14	SQEB
ED15	SQDB
ED17	MEEB
ED18	KDB
ED19	CDB
ED1A	ADB
ED1B	SDB
ED1C	MDB
ED1D	DDB
ED1E	MADB
ED1F	MSDB
ED24	LDE
ED25	LXD
ED26	LXE
ED34	SQE
ED35	SQD
ED37	MEE
EE	PLO
F0	SRP
F1	MVO
F2	PACK
F3	UNPK
F8	ZAP
F9	CP
FA	AP
FB	SP
FC	MP
FD	DP

Condition Codes

Condition Code →	0	1	2	3
Mask Bit Value →	8	4	2	1
General Instructions				
Add	Zero	< Zero	> Zero	Overflow
Add Halfword	Zero	< Zero	> Zero	Overflow
Add Halfword Immediate	Zero	< Zero	> Zero	Overflow
Add Logical	Zero, no carry	Not zero, no carry	Zero, carry	Not zero, carry
AND	Zero	Not zero	----	----
Checksum	Checksum complete	----	----	CPU-determined completion
Compare	Equal	First op low	First op high	----
Compare and Form Codeword	Equal	First op low and ctl = 0, or first op high and ctl = 1	First op high and ctl = 0, or first op low and ctl = 1	----
Compare and Swap	Equal	Not equal	----	----
Compare Double and Swap	Equal	Not equal	----	----
Compare Halfword	Equal	First op low	First op high	----
Compare Halfword Immediate	Equal	First op low	First op high	----
Compare Logical	Equal	First op low	First op high	----
Compare Logical Characters under Mask	Equal, or Mask is zero	First op low	First op high	----
Compare Logical Long	Equal	First op low	First op high	----
Compare Logical Long Extended	Equal	First op low	First op high	CPU-determined completion
Compare Logical String	Equal	First op low	First op high	CPU-determined completion
Compare until Substring Equal	Equal sub-string	Last bytes equal	Last bytes unequal	CPU-determined completion
Compression Call	Second op end	First op end, not second op end	----	CPU-determined completion
Convert Unicode to UTF-8	Data processed	First op full	----	CPU-determined completion
Convert UTF-8 to Unicode	Data processed	First op full	----	CPU-determined completion
Exclusive OR	Zero	Not zero	----	----
Insert Characters under Mask	All zero, or mask is zero	Leftmost bit = 1	Not zero, but with leftmost bit = 0	----
Load and Test	Zero	< Zero	> Zero	----
Load Complement	Zero	< Zero	> Zero	Overflow
Load Negative	Zero	< Zero	----	----
Load Positive	Zero	----	> Zero	Overflow
Move Long	Operand lengths equal	First op shorter	First op longer	Overlap
Move Long Extended	Operand lengths equal	First op shorter	First op longer	CPU-determined completion
Move Page	Data moved	First op invalid, both valid in ES, locked, or ES error	Second op invalid	----
Move String	----	Second op moved	----	CPU-determined completion
OR	Zero	Not zero	----	----
Perform Locked Operation (test bit zero)	Equal	First op not equal	First op equal, third op not equal	----
Perform Locked Operation (test bit one)	Code valid	----	----	Code invalid

Condition Codes (Cont'd)

Condition Code →	0	1	2	3
Mask Bit Value →	8	4	2	1
Search String	----	Character found	Character not found	CPU-determined completion
Set Program Mask	See Note	See Note	See Note	See Note
Shift Left Double	Zero	< Zero	> Zero	Overflow
Shift Left Single	Zero	< Zero	> Zero	Overflow
Shift Right Double	Zero	< Zero	> Zero	----
Shift Right Single	Zero	< Zero	> Zero	----
Store Clock	Set state	Not-set state	Error state	Stopped state or not oper
Store Clock	Set state	Not-set state	Error state	Stopped state or not oper
Subtract	Zero	< Zero	> Zero	Overflow
Subtract Halfword	Zero	< Zero	> Zero	Overflow
Subtract Logical	----	Not zero, no carry	Zero, carry	Not zero, carry
Test and Set	Leftmost bit zero	Leftmost bit one	----	----
Test under Mask	All zeros, or mask is zero	Mixed 0's and 1's	---	All ones
Test under Mask High	All zeros or mask is zero	Mixed 0's and 1's and leftmost bit zero	Mixed 0's and 1's and leftmost bit one	All ones
Test under Mask High	All zeros or mask is zero	Mixed 0's and 1's and leftmost bit zero	Mixed 0's and 1's and leftmost bit one	All ones
Translate and Test	All zeros	Not zero, scan incomplete	Not zero, scan complete	----
Translate Extended	Data processed	First op byte equal test byte	----	CPU-determined completion
Update Tree	Compare equal at current node on path	Path complete, no nodes compared equal	----	Path not complete and compared register negative
Decimal Instructions				
Add Decimal	Zero	< Zero	> Zero	Overflow
Compare Decimal	Equal	First op low	First op high	----
Edit	Zero	< Zero	> Zero	----
Edit and Mark	Zero	< Zero	> Zero	----
Shift and Round Decimal	Zero	< Zero	> Zero	Overflow
Subtract Decimal	Zero	< Zero	> Zero	Overflow
Zero and Add	Zero	< Zero	> Zero	Overflow
Floating-Point Instructions				
Add	Zero	< Zero	> Zero	NaN
Add Normalized	Zero	< Zero	> Zero	----
Add Unnormalized	Zero	< Zero	> Zero	----
Compare (BFP)	Equal	First op low	First op high	Unordered
Compare (HFP)	Equal	First op low	First op high	----
Compare and Signal	Equal	First op low	First op high	Unordered
Convert BFP to HFP	Zero	< Zero	> Zero	Special case
Convert HFP to BFP	Zero	< Zero	> Zero	Special case
Convert to Fixed	Zero	< Zero	> Zero	Special case
Divide to Integer	Remainder complete, quotient normal	Remainder complete, quotient overflow or NaN	Remainder incomplete, quotient normal	Remainder incomplete, quotient overflow or NaN
Load and Test (BFP)	Zero	< Zero	> Zero	NaN
Load and Test (HFP)	Zero	< Zero	> Zero	----
Load Complement (BFP)	Zero	< Zero	> Zero	NaN
Load Complement (HFP)	Zero	< Zero	> Zero	----
Load Negative (BFP)	Zero	< Zero	----	NaN
Load Negative (HFP)	Zero	< Zero	----	----
Load Positive (BFP)	Zero	----	> Zero	NaN

Condition Codes (Cont'd)

Condition Code →	0	1	2	3
Mask Bit Value →	8	4	2	1
Load Positive (HFP)	Zero	----	> Zero	----
Subtract	Zero	< Zero	> Zero	NaN
Subtract Normalized	Zero	< Zero	> Zero	----
Subtract Unnormalized	Zero	< Zero	> Zero	----
Test Data Class	Zero (no match)	One (match)	----	----
Vector Instructions				
Count Left Zeros in VMR	Active bits all zeros	Active bits 0's and 1's	----	Active bits all ones
Count Ones in VMR	Active bits all zeros	Active bits 0's and 1's	----	Active bits all ones
Load Bit Index	VCT = 0, bit count = 0	VCT = 0, bit count < 0	VCT = section size, bit count > 0	VCT > 0, bit count ≤0
Load VCT and Update	VCT = 0, result = 0	VCT = 0, result < 0	VCT = section size, result > 0	VCT > 0, result = 0
Load VCT from Address	VCT = 0, addr = 0	VCT = 0, addr < 0	VCT = section size, addr > section size	VCT > 0, addr > 0, addr ≤ section size
Load VIX from Address	VIX = 0 and eff addr = 0	VIX = 0 and eff addr < 0	VIX > 0 and < VCT	VIX > 0 and ≥ VCT
Restore VR	VR pair 14 examined and not restores	VR pair Other than 14 examined and not restored	VR pair 14 restored	VR pair other than 14 restored
Save Changed VR	VR pair 14 examined and not saved	VR pair other than 14 examined and not saved	VR pair 14 saved	VR pair other than 14 saved
Save VR	VR pair 14 examined and not saved	VR pair other than 14 examined and not saved	VR pair 14 saved	VR pair other than 14 saved
Test VMR	Active bits all zeros	Active bits 0's and 1's	----	Active bits all ones
Control Instructions				
Diagnose	See Note	See Note	See Note	See Note
Extract Stacked State	Branch state entry	Program call state entry	----	----
Insert Address Space Control	Primary-space mode	Secondary-space mode	Access-register mode	Home-space mode
Load Address Space Parameters	Parameters loaded	Primary not available	Secondary not authorized or not available	Space-switch event
Load PSW	See Note	See Note	See Note	See Note
Load Real Address	Translation available	Segment-table entry invalid	Page-table entry invalid	Table length exceeded or ART exception
Move to Primary	Length ≤ 256	----	----	Length > 256
Move to Secondary	Length ≤ 256	----	----	Length > 256
Move with Key	Length ≤ 256	----	----	Length > 256
Program Return	See Note	See Note	See Note	See Note
Reset Reference Bit Extended	Ref = 0, Chg = 0	Ref = 0, Chg = 1	Ref = 1, Chg = 0	Ref = 1, Chg = 1
Set Clock	Set	Secure	----	Not oper
Signal Processor	Accepted	Status stored	Busy	Not oper
Store System Information	Info provided	----	----	Info not available

Condition Codes (Cont'd)

Condition Code →	0	1	2	3
Mask Bit Value →	8	4	2	1
Test Access	ALET = 0	ALET uses DUALD	ALET uses PSALD	ALET = 1 or causes ART exception
Test Block	Usable	Unusable	----	----
Test Protection	Fetch and store allowed	Fetch allowed; no store allowed	No fetch or store allowed	Trans-lation not avail-able
Input/Output Instructions				
Clear Subchannel	Function started	----	----	Not oper
Halt Subchannel	Function started	Noninter-mediate status pending	Busy	Not oper
Modify Subchannel	Function executed	Status pending	Busy	Not oper
Reset Channel Path	Function started	----	Busy	Not oper
Resume Sub-channel	Function started	Status pending	Not applicable	Not oper
Start Subchannel	Function started	Status pending	Busy	Not oper
Store Channel Report Word	CRW stored	Zeros stored	----	----
Store Subchannel	SCHIB stored	----	----	Not oper
Test Pending Inter-ruption	Inter-ruption not pend-ing	Inter-ruption code stored	----	----
Test Subchannel	Status was pending	Status was not pending	----	Not oper

Notes:

For Diagnose, the resulting condition code is model-dependent.

For Load PSW, the condition code is loaded from a field of the second operand (the new PSW's condition code field).

For Program Return, the resulting condition code is unpredictable.

For Set Program Mask, the condition code is loaded from bits 2 and 3 of the first operand.

Assembler Instructions

Function	Mnemonic	Meaning
Option control	*PROCESS	Specify assembler options
	ACONTROL	Dynamically modify options
Data definition	CCW	Define channel command word
	CCW0	Define format-0 channel command word
	CCW1	Define format-1 channel command word
	DC	Define constant
	DS	Define storage
Program sectioning and linking	ALIAS	Rename external symbol
	AMODE	Specify addressing mode
	CATTR	Define class name and attributes
	COM	Identify common control section
	CSECT	Identify control section
	CXD	Cumulative length of external dummy section
	DSECT	Identify dummy section
	DXD	Define external dummy section
	ENTRY	Identify entry-point symbol
	EXTRN	Identify external symbol
	LOCTR	Specify multiple location counters
	RMODE	Specify residence mode
	RSECT	Identify read-only control section
	START	Start assembly
	WXTRN	Identify weak external symbol
Base register assignment	DROP	Drop base address register
	USING	Use base address and register
Control of listings	AEJECT	Start new page in macro definition
	ASPACE	Space lines in macro definition
	CEJECT	Conditional start new page
	EJECT	Start new page
	PRINT	Print optional data
	SPACE	Space listing
	TITLE	Identify assembly output
Program control	ADATA	Provide data for SYSADATA file
	CNOP	Conditional no operation
	COPY	Copy predefined source coding
	END	End assembly
	EQU	Equate symbol
	EXITCTL	Program control data for I/O exits
	ICTL	Input format control
	ISEQ	Input sequence checking
	LTORG	Begin literal pool
	OPSYN	Equate operation code
	ORG	Set location counter
	POP	Restore ACONTROL, PRINT, or USING status
	PUNCH	Punch a card
	PUSH	Save current ACONTROL, PRINT, or USING status
	REPRO	Reproduce following card
Conditional assembly	ACTR	Conditional assembly branch counter
	AGO	Unconditional branch
	AIF	Conditional branch
	AINSERT	Create input record
	ANOP	Assembly no operation
	AREAD	Assign input record to SETC symbol
	GBLA	Define global SETA symbol
	GBLB	Define global SETB symbol
	GBLC	Define global SETC symbol
	LCLA	Define local SETA symbol
	LCLB	Define local SETB symbol
	LCLC	Define local SETC symbol
	MHELP	Trace macro flow
	MNOTE	Generate error message
	SETA	Set arithmetic variable symbol
	SETAF	Set arithmetic variable symbol from external function
	SETB	Set binary variable symbol
	SETC	Set character variable symbol
	SETCF	Set character variable symbol from external function

Function	Mnemonic	Meaning
Macro definition	MACRO	Macro definition header
	MEND	Macro definition trailer
	MEXIT	Macro definition exit

Source: SC26-4940.

Extended-Mnemonic Instructions for Branch on Condition

Use	Extended Mnemonic* (RX or RR)	Meaning	Machine Instr.* (RX or RR)
General	B or BR	Unconditional Branch	BC or BCR 15,
	NOP or NOPR	No Operation	BC or BCR 0,
After Compare Instructions (A:B)	BH or BHR	Branch on A High	BC or BCR 2,
	BL or BLR	Branch on A Low	BC or BCR 4,
	BE or BER	Branch on A Equal B	BC or BCR 8,
	BNH or BNHR	Branch on A Not High	BC or BCR 13,
	BNL or BNLR	Branch on A Not Low	BC or BCR 11,
	BNE or BNER	Branch on A Not Equal B	BC or BCR 7,
After Arithmetic Instructions	BP or BPR	Branch on Plus	BC or BCR 2,
	BM or BMR	Branch on Minus	BC or BCR 4,
	BZ or BZR	Branch on Zero	BC or BCR 8,
	BO or BOR	Branch on Overflow	BC or BCR 1,
	BNP or BNPR	Branch on Not Plus	BC or BCR 13,
	BNM or BNMR	Branch on Not Minus	BC or BCR 11,
	BNZ or BNZR	Branch on Not Zero	BC or BCR 7,
	BNO or BNOR	Branch on No Overflow	BC or BCR 14,
After Test under Mask instruction	BO or BOR	Branch if Ones	BC or BCR 1,
	BM or BMR	Branch if Mixed	BC or BCR 4,
	BZ or BZR	Branch if Zeros	BC or BCR 8,
	BNO or BNOR	Branch if Not Ones	BC or BCR 14,
	BNM or BNMR	Branch if Not Mixed	BC or BCR 11,
	BNZ or BNZR	Branch if Not Zeros	BC or BCR 7,

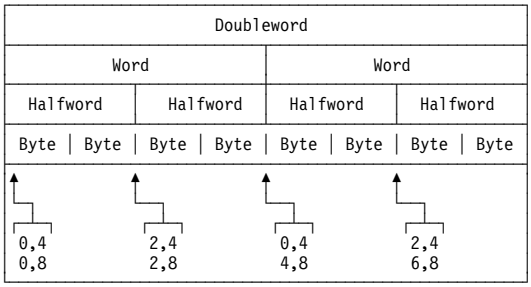
Source: SC26-4940.
*Second operand, not shown, is D_2 (X_2, B_2) for RX format and R_2 for RR format.

Extended-Mnemonic Instructions for Branch Relative on Condition

Use	Extended Mnemonic	Meaning	Machine Instr.
General	BRU or J	Unconditional Branch Relative	BRC 15,I_2
	JNOP	No Operation	BRC 0,I_2
After Compare Instructions (A:B)	BRH or JH	Branch Relative on A High	BRC 2,I_2
	BRL or JL	Branch Relative on A Low	BRC 4,I_2
	BRE or JE	Branch Relative on A Equal B	BRC 8,I_2
	BRNH or JNH	Branch Relative on A Not High	BRC 13,I_2
	BRNL or JNL	Branch Relative on A Not Low	BRC 11,I_2
	BRNE or JNE	Branch Relative on A Not Equal B	BRC 7,I_2
After Arithmetic Instructions	BRP or JP	Branch Relative on Plus	BRC 2,I_2
	BRM or JM	Branch Relative on Minus	BRC 4,I_2
	BRZ or JZ	Branch Relative on Zero	BRC 8,I_2
	BRO or JO	Branch Relative on Overflow	BRC 1,I_2
	BRNP or JNP	Branch Relative on Not Plus	BRC 13,I_2
	BRNM or JNM	Branch Relative on Not Minus	BRC 11,I_2
	BRNZ or JNZ	Branch Relative on Not Zero	BRC 7,I_2
	BRNO or JNO	Branch Relative on No Overflow	BRC 14,I_2
After Test under Mask instruction	BRO or JO	Branch Relative if Ones	BRC 1,I_2
	BRM or JM	Branch Relative if Mixed	BRC 4,I_2
	BRZ or JZ	Branch Relative if Zeros	BRC 8,I_2
	BRNO or JNO	Branch Relative if Not Ones	BRC 14,I_2
	BRNM or JNM	Branch Relative if Not Mixed	BRC 11,I_2
	BRNZ or JNZ	Branch Relative if Not Zeros	BRC 7,I_2

Source: SC26-4940.

CNOP Alignment



Source: SC26-4940.

Summary of Constants

Type	Implied Length, Bytes	Alignment	Format	Truncation/ Padding
A	4	Word	Value of address	Left
B	-	Byte	Binary digits	Left
C	-	Byte	Characters	Right
D	8	Doubleword	Long hex floating point	Right
DB	8	Doubleword	Long binary floating point	Right
DH	8	Doubleword	Long hex floating point	Right
E	4	Word	Short hex floating point	Right
EB	4	Word	Short binary floating point	Right
EH	4	Word	Short hex floating point	Right
F	4	Word	Fixed-point binary	Left
G	Even	Byte	Graphic (double-byte) characters	Right
H	2	Halfword	Fixed-point binary	Left
J	4	Word	Symbol naming a DXD, DSECT, or class	Left
L	16	Doubleword	Extended hex floating point	Right
LB	16	Doubleword	Extended binary floating point	Right
LH	16	Doubleword	Extended hex floating point	Right
P	-	Byte	Packed decimal	Left
Q	4	Word	Symbol naming a DXD or DSECT	Left
S	2	Halfword	Address in base-displacement form	-
V	4	Word	Externally defined address value	-
X	-	Byte	Hexadecimal digits	Left
Y	2	Halfword	Value of address	Left
Z	-	Byte	Zoned decimal	Left

Source: SC26-4940.

Fixed Storage Locations

Area (Dec)	Addr Type	Hex Addr	Function
0-7	A	0	Initial-program-loading PSW
0-7	R	0	Restart new PSW
8-15	A	8	Initial-program-loading CCW1
8-15	R	8	Restart old PSW
16-23	A	10	Initial-program-loading CCW2
24-31	R	18	External old PSW
32-39	R	20	Supervisor-call old PSW
40-47	R	28	Program old PSW
48-55	R	30	Machine-check old PSW
56-63	R	38	Input/output old PSW
88-95	R	58	External new PSW
96-103	R	60	Supervisor-call new PSW
104-111	R	68	Program new PSW
112-119	R	70	Machine-check new PSW
120-127	R	78	Input/output new PSW
128-131	R	80	External-interruption parameter
132-133	R	84	CPU address associated with external interruption, or zeros
134-135	R	86	External-interruption code (see table on page 22)
136-139	R	88	SVC-interruption ID (0-12 zeros, 13-14 ILC, 15 zero, 16-31 code)

Area (Dec)	Addr Type	Hex Addr	Function
140-143	R	8C	Program-interruption ID (0-12 zeros, 13-14 ILC, 15 zero, 16-31 code)
144-147	R	90	Data-exception code (0-23 zeros, 24-31 code (see table on page 23))
144-147	R	90	Translation-exception ID (see table on page 23)
148-149	R	94	Monitor-class number (0-7 zeros, 8-15 number)
150-151	R	96	PER-1 code (0-3 code, 4-15 zeros)
150-151	R	96	PER-2 code (0-2 and 4 code, 3 and 5-8 zeros, 9-13 ATMID, 14-15 SI)
152-155	R	98	PER address (0 zero, 1-31 address)
156-159	R	9C	Monitor code
160	R	A0	Exception access ID (0-3 zeros, 4-7 access-register number)
161	R	A1	PER access ID (0-3 zeros, 4-7 access-register number)
184-187	R	B8	Subsystem-identification word (0-14 zeros, 15 one, 16-31 subchannel number)
188-191	R	BC	I/O-interruption parameter
196-199	R	C4	PCF-entry-table origin (0 and 20-31 zeros, 1-19 origin)
212-215	A	D4	Store-status extended-save-area address (0 and 20-31 zeros, 1-19 address or zeros)**
212-215	R	D4	Machine-check extended-save-area address (0 and 20-31 zeros, 1-19 address or zeros)**
216-223	A	D8	Store-status CPU-timer save area
216-223	R	D8	Machine-check CPU-timer save area
224-231	A	E0	Store-status clock-comparator save area
224-231	R	E0	Machine-check clock-comparator save area
232-239	R	E8	Machine-check-interruption code (see diagram on page 37)
244-247	R	F4	External-damage code (see diagram on page 37)
248-251	R	F8	Failing-storage address (0 zero, 1-31 address)
256-263	A	100	Store-status PSW save area
256-271	R	100	Fixed-logout area*
264-267	A	108	Store-status prefix save area
288-351	A	120	Store-status access-register save area
288-351	R	120	Machine-check access-register save area
352-383	A	160	Store-status floating-point-register save area (registers 0, 2, 4, 6 only)
352-383	R	160	Machine-check floating-point-register save area (registers 0, 2, 4, 6 only)
384-447	A	180	Store-status general-register save area
384-447	R	180	Machine-check general-register save area
448-511	A	1C0	Store-status control-register save area
448-511	R	1C0	Machine-check control-register save area

A=Absolute address. R=Real address.
* Contents may vary among models; see System Library manuals.
** Extended-save-area contents: 0-127 floating-point registers 0-15, 128-131 floating-point-control register.

External-Interruption Codes

At real-storage address 134-135 (86-87 hex)

Code (Hex)	Condition
0040	Interrupt key
1003	TOD-clock-sync check
1004	Clock comparator
1005	CPU timer
1200	Malfunction alert
1201	Emergency signal
1202	External call
1406	ETR
2401	Service signal

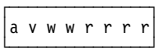
Program-Interruption Codes

At real-storage address 142-143 (8E-8F hex)

Code (Hex)	Condition
0001	Operation exception
0002	Privileged-operation exception
0003	Execute exception
0004	Protection exception
0005	Addressing exception
0006	Specification exception
0007	Data exception
nn08*	Fixed-point-overflow exception
0009	Fixed-point-divide exception
000A	Decimal-overflow exception
000B	Decimal-divide exception
nn0C*	HFP-exponent-overflow exception
nn0D*	HFP-exponent-underflow exception
nn0E*	HFP-significance exception
nn0F*	HFP-floating-point-divide exception
0010	Segment-translation exception
0011	Page-translation exception
0012	Translation-specification exception
0013	Special-operation exception
0015	Operand exception
0016	Trace-table exception
0017	ASN-translation-specification exception
0019	Vector-operation exception
001C	Space-switch event
001D*	HFP-square-root exception
nn1E*	Unnormalized-operand exception
001F	PC-translation-specification exception
0020	AFX-translation exception
0021	ASX-translation exception
0022	LX-translation exception
0023	EX-translation exception
0024	Primary-authority exception
0025	Secondary-authority exception
0028	ALET-specification exception
0029	ALEN-translation exception
002A	ALE-sequence exception
002B	ASTE-validity exception
002C	ASTE-sequence exception
002D	Extended-authority exception
0030	Stack-full exception
0031	Stack-empty exception
0032	Stack-specification exception
0033	Stack-type exception
0034	Stack-operation exception
0040	Monitor event
0080	PER event (code may be combined with another code)
0119	Crypto-operation exception

* Use the exception-extension-code table on page 23 for bits 0-7 (nn) of the program-interruption code.

Exception-Extension Code



Bit	Meaning
0	(a) Arithmetic-partial-completion bit 0 Completion or suppression of instruction, and bits 1-7 of the exception-extension code are also zero 1 Partial completion of vector instruction
1	(v) Arithmetic-result location 0 Scalar register 1 Vector register
2-3	(ww) Arithmetic-result width 01 4-byte result 10 8-byte result
4-7	(rrrr) Register number of result register designated by the interrupted instruction

Translation-Exception Identification

At real-storage address 144-147 (90-93 hex)

Interruption Code (Hex)	Exception or Event	Format of Information Stored
0004	Protection	If 29 zero: rest unpredictable If 29 one and DAT used: 1-19 address, 30-31 STD identification, rest unpredictable If 29 one and DAT not used: 1-19 address, rest unpredictable
0010	Segment translation	0 secondary address, 1-19 address, 20-29 unpredictable, 30-31 STD identification
0011	Page translation	0 secondary address, 1-19 address, 20-29 unpredictable, 30-31 STD identification
001C	Space switch	From primary-space mode: 0 old primary-space-switch-event control, 1-15 zeros, 16-31 old PASN From home-space mode: 0 home-space-switch-event control, 1-31 zeros
0020	AFX translation	0-15 zeros, 16-31 address-space number
0021	ASX translation	0-15 zeros, 16-31 address-space number
0022	LX translation	0-11 zeros, 12-31 program-call number
0023	EX translation	0-11 zeros, 12-31 program-call number
0024	Primary authority	0-15 zeros, 16-31 address-space number
0025	Secondary authority	0-15 zeros, 16-31 address-space number

Data-Exception Code (DXC)

At real-storage address 147 (93 hex) and in byte 2 of floating-point-control register

Code (Hex)	Data Exception
00	Decimal operand
01	AFP register
02	BFP instruction
08	IEEE inexact and truncated
0C	IEEE inexact and incremented
10	IEEE underflow, exact
18	IEEE underflow, inexact and truncated
1C	IEEE underflow, inexact and incremented
20	IEEE overflow, exact
28	IEEE overflow, inexact and truncated
2C	IEEE overflow, inexact and incremented
40	IEEE division by zero
80	IEEE invalid operation

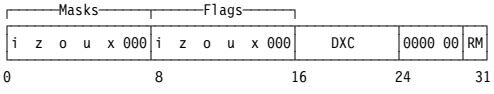
Control Registers

CR	Bits	Name of Field	Associated with	Init*
0	1	SSM-suppression control	SSM instruction	0
	2	TOD-clock-sync control	Multiprocessing	0
	3	Low-address-protection control	Low-address protection	0
	4	Extraction-authority control	Instruction authorization	0
	5	Secondary-space control	Instruction authorization	0
	6	Fetch-protection-override control	Key-controlled protection	0
	7	Storage-protection-override control	Key-controlled protection	0
	8-12	Translation format (10110 binary)	Dynamic address translation	0
	13	AFP-register control	Floating point	0
	14	Vector control	Vector operations	0
	15	Address-space-function control	Instruction authorization	0
	16	Malfunction-alert subclass mask	Multiprocessing	0
	17	Emergency-signal subclass mask	Multiprocessing	0
	18	External-call subclass mask	Multiprocessing	0
	19	TOD-clock sync-check subclass mask	Multiprocessing	0
	20	Clock-comparator subclass mask	Clock comparator	0
	21	CPU-timer subclass mask	CPU timer	0
	22	Service-signal subclass mask	Service signal	0
	24	Unused (See note)		1
	25	Interrupt-key subclass mask	Interrupt key	1
	26	Unused (See note)		1
	27	ETR subclass mask	External interruptions	0
	28	Program-call-fast control	PROGRAM CALL FAST	0
	29	Crypto control	Cryptography	0
1	0	Primary space-switch-event control	Program interruptions	0
	1-19	Primary segment-table origin	Dynamic address translation	0
	22	Primary subspace-group control	Subspace groups	0
	23	Primary private-space control	Dynamic address translation	0
	24	Primary storage-alteration-event control	Program-event rec. 2 only	0
	25-31	Primary segment-table length	Dynamic address translation	
2	1-25	Dispatchable-unit-control-table origin	Access-register translation	0
3	0-15	PSW-key mask	Instruction authorization	0
	16-31	Secondary ASN	Address spaces	0
4	0-15	Authorization index	Instruction authorization	0
	16-31	Primary ASN	Address spaces	0
5	0	Subsystem-linkage control	Instruction authorization	0
	1-24	Linkage-table origin	PC-number translation	0
	25-31	Linkage-table length	PC-number translation	0
&	-----	----------	----------	---
5	1-25	Primary-ASTE origin	Access-register translation	0
6	0-7	I/O-interruption subclass mask	I/O interruptions	0
7	1-19	Secondary segment-table origin	Dynamic address translation	0
	22	Secondary subspace-group control	Subspace groups	0
	23	Secondary private-space control	Dynamic address translation	0
	24	Secondary-storage-alteration-event control	Program-event rec. 2 only	0
	25-31	Secondary segment-table length	Dynamic address translation	0
8	0-15	Extended authorization index	Access-register translation	0
	16-31	Monitor masks	MC instruction	0

CR	Bits	Name of Field	Associated with	Init*
9	0	Successful-branching-event mask	Program-event recording	0
	1	Instruction-fetching-event mask	Program-event recording	0
	2	Storage-alteration-event mask	Program-event recording	0
	3	GR-alteration-event mask	Program-event rec. 1 only	0
	4	Store-using-real-address-event mask	Program-event recording	0
	8	Branch-address control	Program-event rec. 2 only	0
	10	Storage-alteration-space control	Program-event rec. 2 only	0
	16-31	PER general-register masks	Program-event rec. 1 only	0
10	1-31	PER starting address	Program-event recording	0
11	1-31	PER ending address	Program-event recording	0
12	0	Branch-trace control	Tracing	0
	1-29	Trace-entry address	Tracing	0
	30	ASN-trace control	Tracing	0
	31	Explicit-trace control	Tracing	0
13	0	Home space-switch-event control	Program interruptions	0
	1-19	Home segment-table origin	Dynamic address translation	0
	23	Home private-space control	Dynamic address translation	0
	24	Home storage-alteration-event control	Program-event rec. 2 only	0
	25-31	Home segment-table length	Dynamic address translation	
14	0	Unused (See note)		1
	1	Unused (See note)		1
	2	Extended-save-area control	Floating point	0
	3	Channel-report-pending subclass mask	I/O machine-check handling	0
	4	Recovery subclass mask	Machine-check handling	0
	5	Degradation subclass mask	Machine-check handling	0
	6	External-damage subclass mask	Machine-check handling	1
	7	Warning subclass mask	Machine-check handling	0
	10	TOD-clock-control-override control	TOD clock Instruction authorization	0
	12	ASN-translation control	ASN translation	0
	13-31	ASN-first-table origin		0
15	1-28	Linkage-stack-entry address	Linkage-stack operations	0

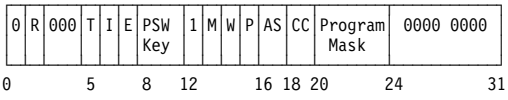
* Value after initial CPU reset.
& The interpretation of control register 5 depends on the state of bit 15 of control register 0, the address-space-function control.
Note: This bit is not used but is initialized to one for consistency with the System/370 definition.

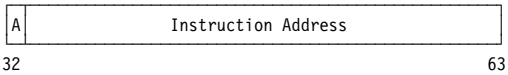
Floating-Point-Control (FPC) Register



Bit	Meaning
0	(IMi) IEEE-invalid-operation mask
1	(IMz) IEEE-division-by-zero mask
2	(IMo) IEEE-overflow mask
3	(IMu) IEEE-underflow mask
4	(IMx) IEEE-inexact mask
8	(SFi) IEEE-invalid-operation flag
9	(SFz) IEEE-division-by-zero flag
10	(SFo) IEEE-overflow flag
11	(SFu) IEEE-underflow flag
12	(SFx) IEEE-inexact flag
16-23	(DXC) Data-exception code (see table on page 23)
30-31	(RM) Rounding mode 00 Round to nearest 01 Round toward 0 10 Round toward +∞ 11 Round toward −∞

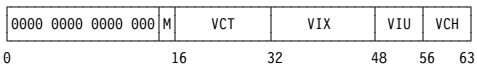
Program-Status Word (PSW)





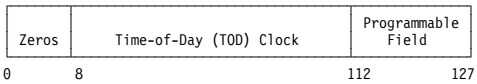
Bit	Meaning
1	(R) Program-event-recording mask
5	(T = 1) DAT mode
6	(I) Input/output mask
7	(E) External mask
13	(M) Machine-check mask
14	(W = 1) Wait state
15	(P = 1) Problem state
16-17	(AS) Address-space control xx Real mode (T = 0) 00 Primary-space mode (T = 1) 01 Access-register mode (T = 1) 10 Secondary-space mode (T = 1) 11 Home-space mode (T = 1)
18-19	(CC) Condition code
20	Fixed-point-overflow mask
21	Decimal-overflow mask
22	HFP-exponent-underflow mask
23	HFP-significance mask
32	(A = 1) 31-bit addressing mode

Vector-Status Register



Bit	Meaning
15	(M) Vector-mask-mode bit
16-31	(VCT) Vector count
32-47	(VIX) Vector interruption index
48-55	(VIU) Vector in-use bits
56-63	(VCH) Vector change bits

Operand of STORE CLOCK EXTENDED



Note: Bit 51 of the TOD clock (bit 59 of the operand) corresponds to one microsecond.

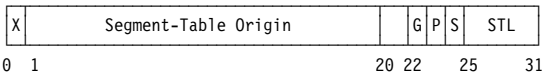
Dynamic Address Translation

Dynamic-Address-Translation Format

Addr Mode	Segment Size	Page Size	Virtual Address Fields			
			Ignored	Segment Index	Page Index	Byte Index
24	1M	4K	0-7	8-11	12-19	20-31
31	1M	4K	0	1-11	12-19	20-31

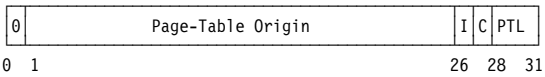
Note: Control register 0 bits 8-12 must contain 10110 (binary); any other combination of bits 8-12 is invalid for translation.

Segment-Table Designation (STD)



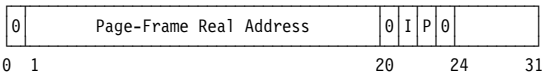
Bit	Meaning
0	(X) Space-switch-event control
22	(G) Subspace-group control
23	(P) Private-space control
24	(S) Storage-alteration-event control
25-31	(STL) Segment-table length

Segment-Table Entry (STE)



Bit	Meaning
26	(I) Segment-invalid bit
27	(C) Common-segment bit
28-31	(PTL) Page-table length

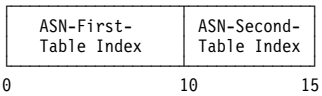
Page-Table Entry (PTE)



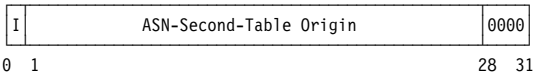
Bit	Meaning
21	(I) Page-invalid bit
22	(P) Page-protection bit

ASN Translation

Address-Space Number (ASN)

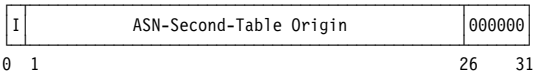


ASN-First-Table Entry (when CR0 Bit 15 Is Zero)



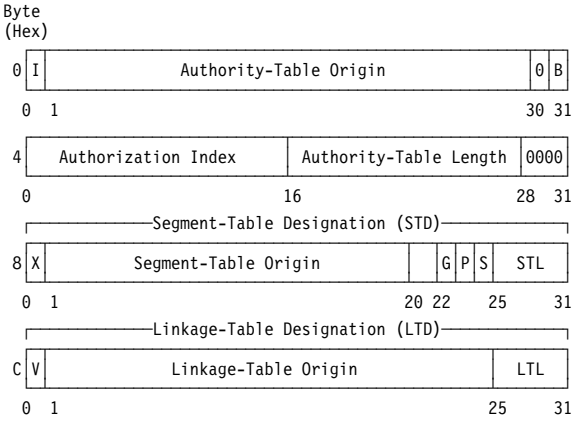
Bit	Meaning
0	(I) AFX-invalid bit

ASN-First-Table Entry (when CR0 Bit 15 Is One)

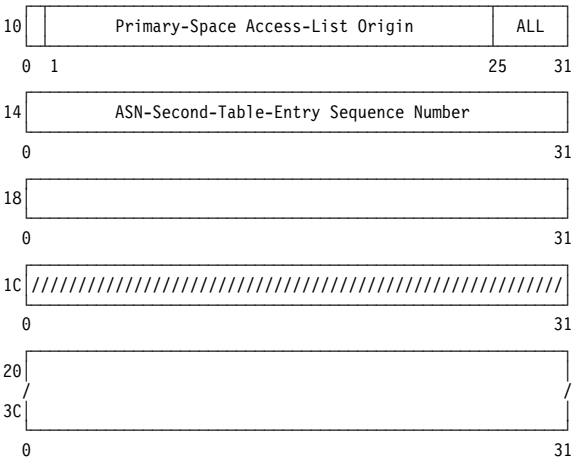


Bit	Meaning
0	(I) AFX-invalid bit

ASN-Second-Table Entry (ASTE)



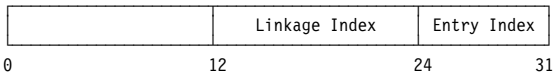
The following extension exists if CR0 bit 15 is one:



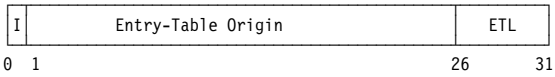
Byte.Bit	Meaning
0.0	(I) ASX-invalid bit
0.31	(B) Base-space bit
8.0-31	See STD on page 27
C.0	(V) Subsystem-linkage control
C.25-31	(LTL) Linkage-table length
10.25-31	(ALL) Access-list length, format 0 (in format 1, bits 24-31)
///	Available for programming

PC-Number Translation

Program-Call Number

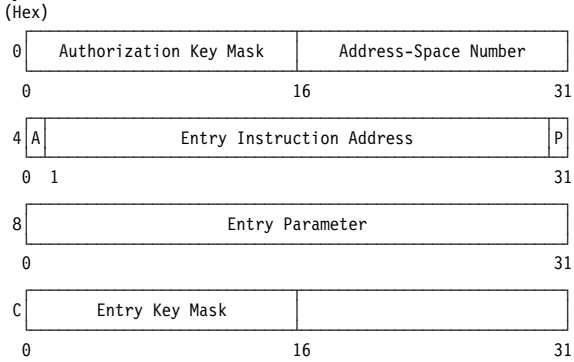


Linkage-Table Entry (LTE)

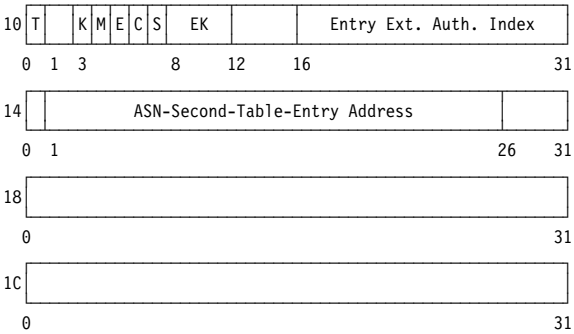


Bit	Meaning
0	(I) LX-invalid bit
26-31	(ETL) Entry-table length

Entry-Table Entry (ETE)



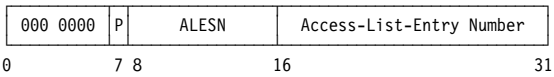
The following extension exists if CR0 bit 15 is one:



Byte.Bit	Meaning
4.0	(A) Entry addressing mode
4.31	(P) Entry problem state
10.0	(T) PC-type bit (0: basic; 1: stacking)
10.3	(K) PSW-key control (0: unchanged; 1: replace if stacking)
10.4	(M) PSW-key-mask control (0: OR, 1: replace if stacking)
10.5	(E) EAX control (0: unchanged; 1: replace if stacking)
10.6	(C) Address-space-control control
10.7	(S) Secondary-ASN control
10.8-11	(EK) Entry key

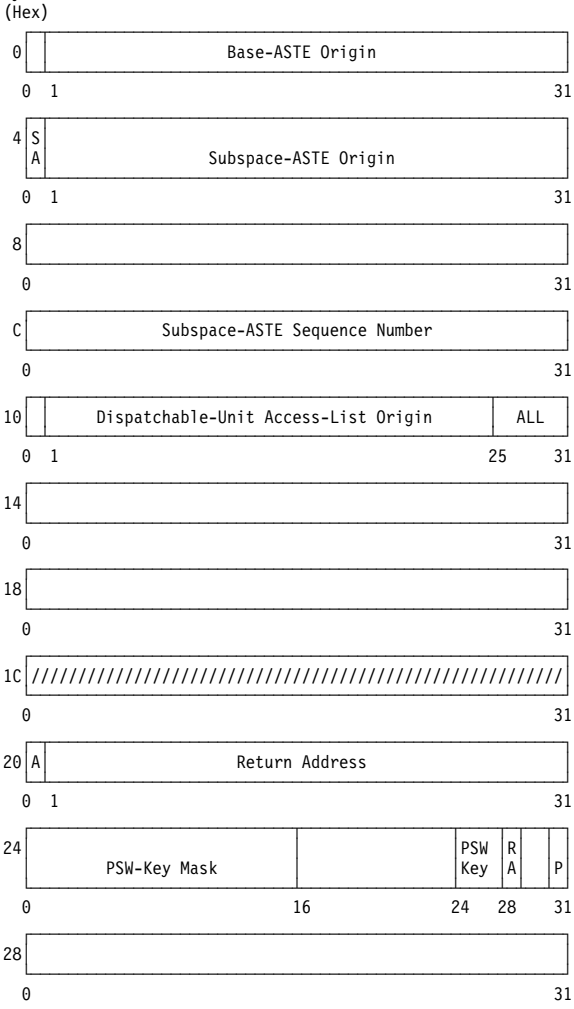
Access-Register Translation

Access-List-Entry Token (ALET)

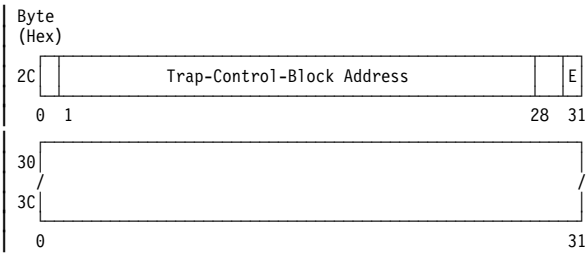


Bit	Meaning
7	(P) Primary-list bit (0: use DUCT; 1: use primary ASTE)
8-15	(ALESN) Access-list-entry sequence number

Dispatchable-Unit-Control Table (DUCT)

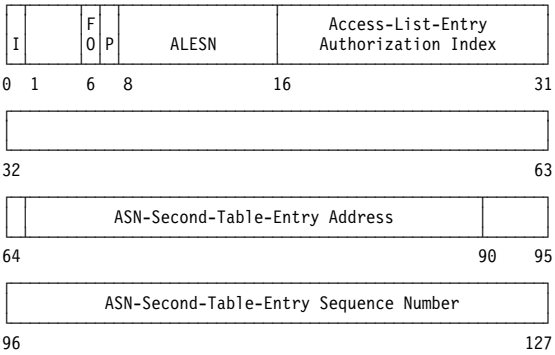


Dispatchable-Unit-Control Table (DUCT) (Cont'd)



Byte.Bit	Meaning
4.0	(SA) Subspace-active bit
10.25-31	(ALL) Access-list length, format 0 (in format 1, bits 24-31)
24.28	(RA) Reduced-authority bit
24.31	(P) Problem-state bit
2C.31	(E) TRAP-enabled bit
///	Available for programming

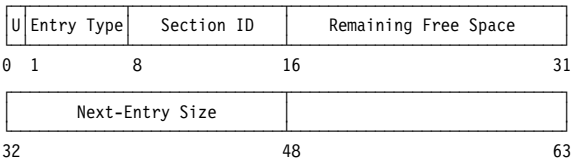
Access-List Entry (ALE)



Bit	Meaning
0	(I) ALEN-invalid bit
6	(FO) Fetch-only bit
7	(P) Private bit
8-15	(ALESN) Access-list-entry sequence number

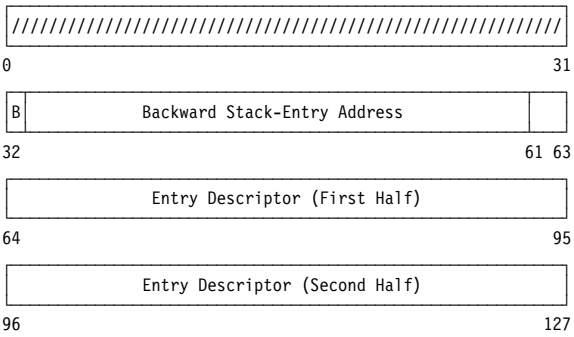
Linkage-Stack Entries

Entry Descriptor



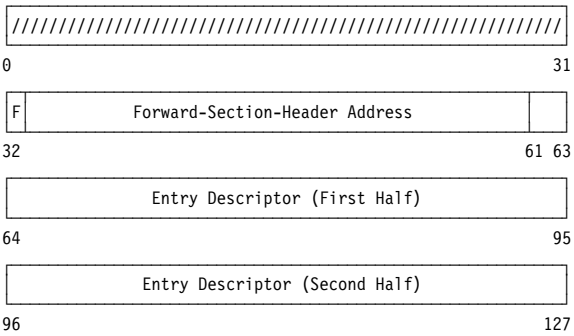
Bit	Meaning
0	(U) Unstack-suppression bit
1-7	Entry type: Header entry = 0000001 binary Trailer entry = 0000010 binary Branch state entry = 0000100 binary Program-call state entry = 000010 binary Available for program use = 1xxxxxx binary

Header Entry (Entry Type 0000001)



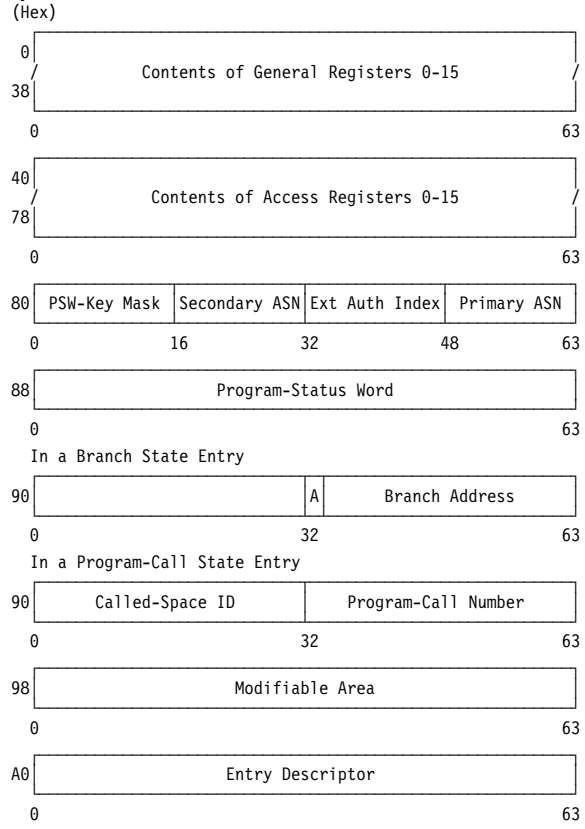
Bit	Meaning
32	(B) Backward stack-entry validity bit
///	Available for programming

Trailer Entry (Entry Type 0000010)



Bit	Meaning
32	(F) Forward-section validity bit
///	Available for programming

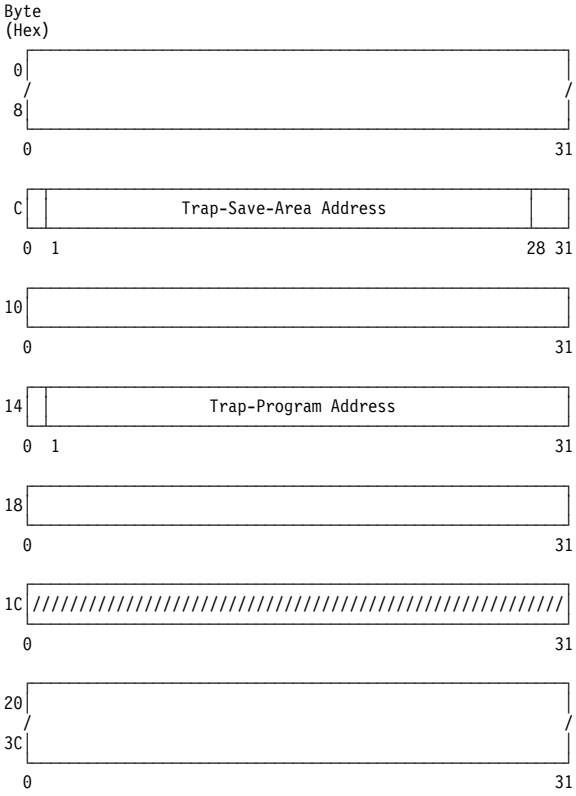
Branch State Entry (Entry Type 0000100) and Program-Call State Entry (Entry Type 0000101)



Byte.Bit	Meaning
90.32	(A) Addressing mode (in branch state entry)

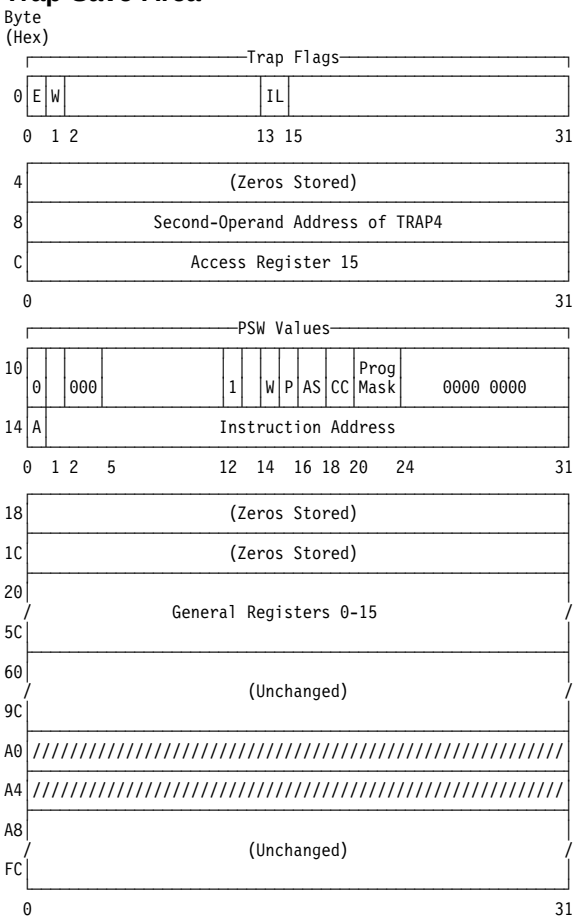
Trapping

Trap Control Block



Byte.Bit	Meaning
///	Available for programming

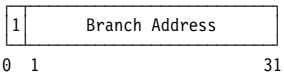
Trap Save Area



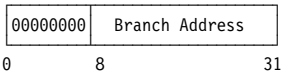
Byte.Bit	Meaning
0.0	(E) TRAP was target of EXECUTE
0.31	(W) TRAP is TRAP4 (not TRAP2)
0.13-14	(IL) Instruction-length code
10-17	PSW values (see PSW on page 26)
///	Available for programming

Trace-Entry Formats

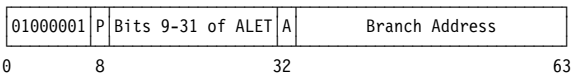
31-Bit Branch



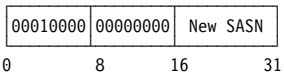
24-Bit Branch



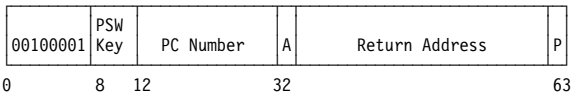
Branch in Subspace Group (if ASN Tracing On)



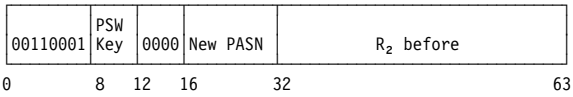
Set Secondary ASN



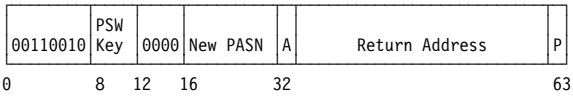
Program Call

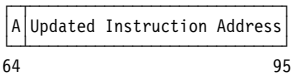


Program Transfer

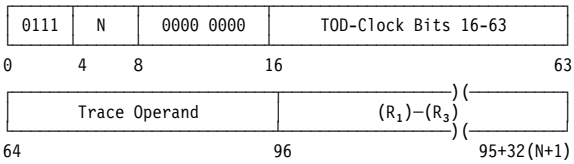


Program Return





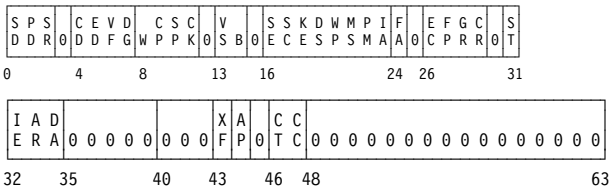
Trace



Bit	Meaning
4-7	(N) One less than the number of registers in the trace entry.

Machine-Check Interruption Code

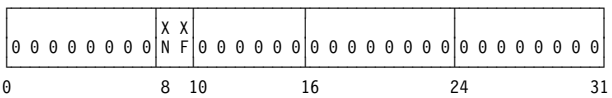
At real-storage address 232-239 (E8-EF hex)



Bit	Meaning
0	(SD) System damage
1	(PD) Instruction-processing damage
2	(SR) System recovery
4	(CD) Timing-facility damage
5	(ED) External damage
6	(VF) Vector-facility failure
7	(DG) Degradation
8	(W) Warning
9	(CP) Channel report pending
10	(SP) Service-processor damage
11	(CK) Channel-subsystem damage
13	(VS) Vector-facility source
14	(B) Backed up
16	(SE) Storage error uncorrected
17	(SC) Storage error corrected
18	(KE) Storage-key error uncorrected
19	(DS) Storage degradation
20	(WP) PSW-MWP validity
21	(MS) PSW mask and key validity
22	(PM) PSW program-mask and condition-code validity
23	(IA) PSW-instruction-address validity
24	(FA) Failing-storage-address validity
26	(EC) External-damage-code validity
27	(FP) Floating-point-register validity
28	(GR) General-register validity
29	(CR) Control-register validity
31	(ST) Storage logical validity
32	(IE) Indirect storage error
33	(AR) Access-register validity
34	(DA) Delayed-access exception
43	(XF) Extended-floating-point-register validity
44	(AP) Ancillary report
46	(CT) CPU-timer validity
47	(CC) Clock-comparator validity

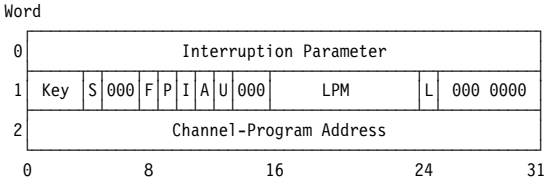
External-Damage Code

At real-storage address 244-247 (F4-F7 hex)



Bit	Meaning
8	(XN) Expanded storage not operational
9	(XF) Expanded-storage control failure

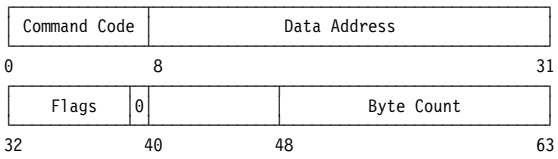
Operation-Request Block (ORB)



Word.Bit	Meaning
1.0-3	(Key) Subchannel key
1.4	(S) Suspend control
1.8	(F) CCW-format control
1.9	(P) Prefetch control
1.10	(I) Initial-status-interruption control
1.11	(A) Address-limit-checking control
1.12	(U) Suppress-suspended-interruption control
1.16-23	(LPM) Logical-path mask
1.24	(L) Incorrect-length-suppression mode

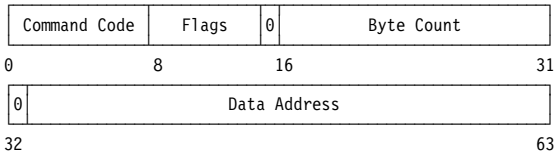
Channel-Command Word (CCW)

Format-0 CCW



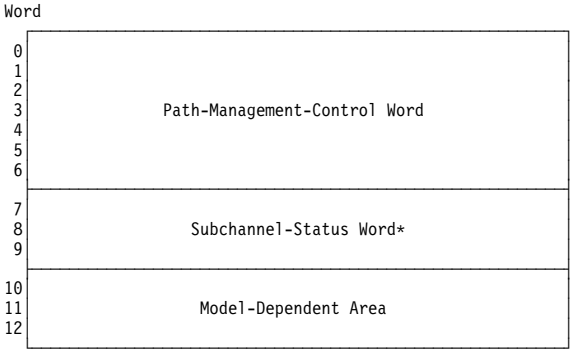
Bit	Meaning
32	(CD) Causes use of data-address portion of next CCW
33	(CC) Causes use of command code and data address of next CCW
34	(SLI) Causes suppression of possible incorrect-length indication
35	(Skip) Suppresses transfer of information to main storage
36	(PCI) Causes an intermediate-interruption condition to occur
37	(IDA) Causes bits 8-31 of CCW to specify location of first IDAW
38	(Suspend) Causes suspension before execution of this CCW

Format-1 CCW



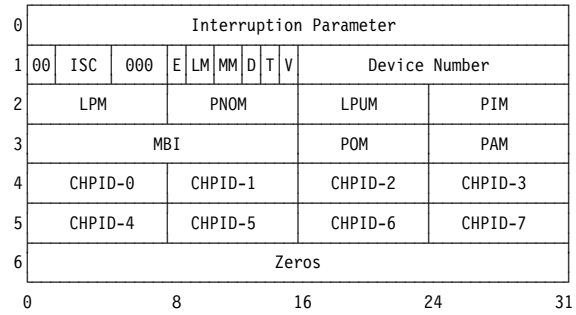
Bit	Meaning
8	(CD) Causes use of data-address portion of next CCW
9	(CC) Causes use of command code and data address of next CCW
10	(SLI) Causes suppression of possible incorrect-length indication
11	(Skip) Suppresses transfer of information to main storage
12	(PCI) Causes an intermediate-interruption condition to occur
13	(IDA) Causes bits 8-31 of CCW to specify location of first IDAW
14	(Suspend) Causes suspension before execution of this CCW

Subchannel-Information Block (SCHIB)



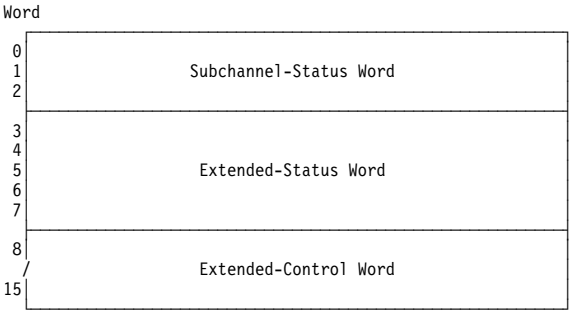
*See “Subchannel-Status Word (SCSW)” on page 40.

Path-Management-Control Word (PMCW)

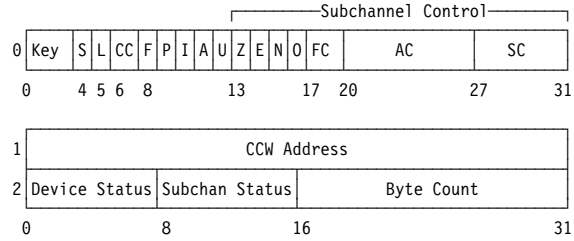


Word.Bit	Meaning
1.2-4	(ISC) Interruption-subclass code
1.8	(E) Subchannel enabled
1.9-10	(LM) limit mode 00 No Checking 01 Data address must be ≥ limit 10 Data address must be < limit 11 Reserved
1.11-12	(MM) Measurement-mode enable 00 Neither mode enabled 01 Device-connect-time-measurement enabled 10 Measurement-block-update enabled 11 Both modes enabled
1.13	(D) Multipath mode
1.14	(T) Timing facility available
1.15	(V) Device number valid
2.0-7	(LPM) Logical-path mask
2.8-15	(PNOM) Path-not-operational mask
2.16-23	(LPUM) Last-path-used mask
2.24-31	(PIM) Path-installed mask
3.0-15	(MBI) Measurement-block index
3.16-23	(POM) Path-operational mask
3.24-31	(PAM) Path-available mask
4.0-7	(CHPID-0) Channel-path ID for logical path 0 (typical)

Interruption-Response Block (IRB)



Subchannel-Status Word (SCSW)



Word. Bit **Meaning**

0.0-3 (Key) Subchannel key
0.4 (S) Suspend control
0.5 (L) Extended-status-word format (logout stored)
0.6-7 CC) Deferred condition code
00 Normal I/O interruption
01 Status in SCSW
10 Reserved
11 Path not operational
0.8 (F) CCW-format control
0.9 (P) Prefetch control
0.10 (I) Initial-status-interruption control
0.11 (A) Address-limit-checking control
0.12 (U) Suppress-suspended-interruption control
0.13 (Z) Zero condition code
0.14 (E) Extended control (information stored in ECW of IRB)
0.15 (N) Path not operational (PNOM nonzero)
0.17-19 (FC) Function control
17 (40) start, 18 (20) halt, 19 (10) clear
0.20-26 (AC) Activity control

20 (08) resume pending	24 (80) subchannel active
21 (04) start pending	25 (40) device active
22 (02) halt pending	26 (20) suspended
23 (01) clear pending	

0.27-31 (SC) Status control

27 (10) alert	30 (02) secondary
28 (08) intermediate	31 (01) status pending
29 (04) primary	

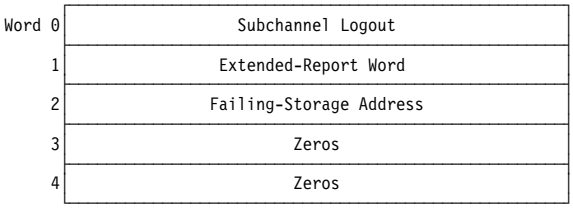
2.0-15 Device status (0-7), subchannel status (8-15)

0 (80) Attention	8 (80) Prog.-cont. int.
1 (40) Status modifier	9 (40) Incorrect length
2 (20) Control-unit end	10 (20) Program check
3 (10) Busy	11 (10) Protection check
4 (08) Channel end	12 (08) Channel-data check
5 (04) Device end	13 (04) Channel-control check
6 (02) Unit check	14 (02) Interface-control check
7 (01) Unit exception	15 (01) Chaining check

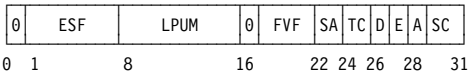
Extended-Status Word (ESW)

See chart on page 42 to determine the appropriate ESW format.

Format-0 ESW

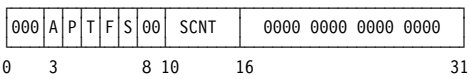


Format-0 ESW (Word 0)



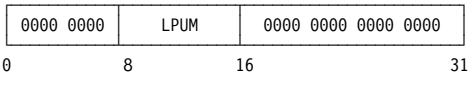
Bit	Meaning
1-7	(ESF) Extended-status flags (1 key check, 2 measurement-block program check, 3 measurement-block data check, 4 measurement -block protection check, 5 CCW check, 6 IDAW check, 7:0)
8-15	(LPUM) Last-path-used mask
17-21	(FVF) Field-validity flags (17 LPUM, 18 TC, 19 SC, 20 device status, 21 CCW address)
22-23	(SA) Storage-access code (00 access type unknown, 01 read, 10 write, 11 read backward)
24-25	(TC) Termination code (00 halt signal issued, 01 stop, stack, or normal termination, 10 clear signal issued)
26	(D) Device status check
27	(E) Secondary error
28	(A) I/O-error alert
29-31	(SC) Sequence code

Format-0 ESW (Word 1)



Bit	Meaning
3	(A) Authorization check
4	(P) Path-verification-required
5	(T) Channel-path timeout
6	(F) Failing-storage-address validity
7	(S) Concurrent sense
10-15	(SCNT) Concurrent-sense count

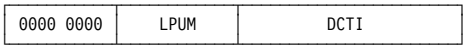
Format-1 ESW (Word 0)*



Bit	Meaning
8-15	(LPUM) Last-path-used mask

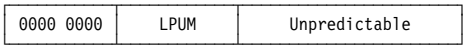
*Word 1 is stored as word 1 of a Format-0 ESW. Words 2, 3, and 4 are zeros.

Format-2 ESW (Word 0)*



Bit	Meaning
8-15	(LPUM) Last-path-used mask
16-31	(DCTI) Device-connect-time interval

Format-3 ESW (Word 0)*



Bit	Meaning
8-15	(LPUM) Last-path-used mask

Information Stored in ESW

Subchannel Conditions under which ESW Is Stored by Test Subchannel Instruction						Extended-Status Word (ESW)	
Subchannel-Status Word			Path-Management-Control Word				
Status-Control Field AIPSX	L Bit	Suspended Bit	Timing-Facility Bit	Device-Connect-Time Measurment-Mode-Enable Bit	Device-Connect-Time Measurment-Mode Active	Format	Contents Word 0 Byte 0123
----0	-	*	*	*	No/Yes	U	****
**001	1	*	*	*	No/Yes	0	RRRR
**1*1	1	*	*	*	No/Yes	0	RRRR
10011	1	*	*	*	No/Yes	0	RRRR
00001	0	*	*	*	No/Yes	U	****
00011	0	*	*	*	No/Yes	3	ZM**
100*1	0	*	*	*	No/Yes	3	ZM**
**1*1	0	*	0	*	No/Yes	1	ZMZZ
**1*1	0	*	1	0	No/Yes	1	ZMZZ
**1*1	0	*	1	1	No	1	ZMZZ
**1*1	0	*	1	1	Yes	2	ZMDD
01001	0	0	*	*	No/Yes	U	****
01001	0	1	0	*	No/Yes	1	ZMZZ
01001	0	1	1	0	No/Yes	1	ZMZZ
01001	0	1	1	1	No	1	ZMZZ
01001	0	1	1	1	Yes	2	ZMDD
00011	1	These combinations do not occur.					
11001	0						
*1011	*						

Bit	Meaning
-	Not meaningful.
*	Bits may be 0's or 1's.
A	Alert status.
D	Accumulated device-connect-time-interval (DCTI) value stored in bytes 2 and 3.
I	Intermediate status.
L	Extended-status-word format.
M	Last-path-used mask (LPUM) stored in byte 1.
P	Primary status.
R	Subchannel-logout information stored in bytes 0-3.
S	Secondary status.
U	No format defined.
X	Status pending.
Z	Bits are stored as 0's.

*Word 1 is stored as word 1 of a Format-0 ESW. Words 2, 3, and 4 are zeros.

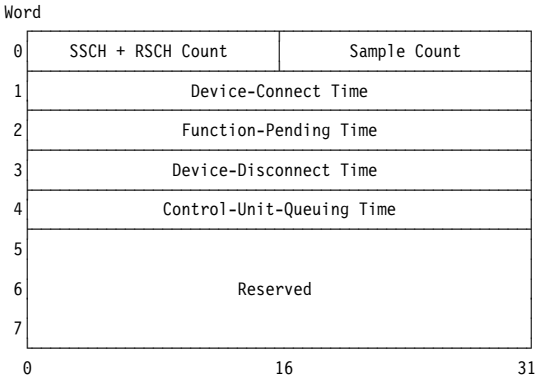
Extended-Control Word (ECW)

The contents of the extended-control word (ECW) are specified by bits 5 and 14 of word 0 of the subchannel-status word, as follows:

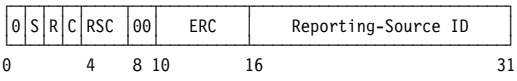
SCSW Bits 5	SCSW Bits 14	ERW Bit 7	ERW Bits 10-15	ECW Words 0-7
0	0	0	Zeros	Unpredictable
0	1	1	No. of con-sen* bytes	Concurrent-sense information*
1	0	0	Zeros	Unpredictable
1	1	0	Zeros	Model-dependent information
1	1	1	No. of con-sen bytes	Concurrent-sense information

*The combination of SCSW bit 5 as zero, SCSW bit 14 as one, and ERW bit 7 as zero does not occur.

Measurement Block



Channel-Report Word (CRW)



Bit	Meaning
1	(S) Solicited CRW
2	(R) Overflow (one or more CRWs lost)
3	(C) Chaining (meaningless if bit 2 is one)
4-7	(RSC) Reporting-source code (see Reporting-Source table)
10-15	(ERC) Error-recovery code (see Error-Recovery-Code table)
16-31	Reporting-source ID (see Reporting-Source table)

Reporting Source

The reporting-source-ID format depends on the RSC field of the channel-report word, as follows:

RSC	Reporting Source	Reporting-Source ID	
0010	Monitoring facility	00000000	00000000
0011	Subchannel	XXXXXXXX	XXXXXXXX
0100	Channel path	00000000	YYYYYYYY
1001	Configuration-alert facility	00000000	YYYYYYYY
1011	Channel subsystem	00000000	00000000

X = Subchannel number
Y = Channel-path ID (CHPID)

Error-Recovery Codes

ERC	Condition
000001	Available
000010	Initialized
000011	Temporary error
000100	Installed parameters initialized
000101	Terminal
000110	Permanent error with facility not initialized
000111	Permanent error with facility initialized
001000	Installed parameters modified

I/O Command Codes

Standard Command-Code Assignments (CCW Bits 0-7)

xxxx 0000 Invalid Command	mmmm 0100 Sense
mmmm mm01 Write	0000 0100 ——Basic Sense
mmmm mm10 Read	1110 0100 ——Sense ID
0000 0010 ——Read Ipl	xxxx 1000 Transfer in Channel (a)
mmmm mm11 Control	0000 1000 Transfer in Channel (b)
0000 0011 ——Control No operation	mmmm 1000 Invalid Command (c)
	mmmm 1100 Read Backward

x —— Bit Ignored
m —— Modifier bit for specific type of I/O device

a Format—0 CCW
b Format—1 CCW
c Format—1 CCW and nonzero m bit

Standard Meanings of Bits of First Sense Byte

Bit	Designation	Bit	Designation
0	Command reject	4	Data check
1	Intervention required	5	Overrun
2	Bus—out check	6	(Device—dependent)
3	Equipment check	7	(Device—dependent)

Console Printer Channel Commands

Write, No Carrier Return	01	Sense	04
Write, Auto Carrier Return	09	Audible Alarm	0B
Read Inquiry	0A	No Operation	03

Printer Channel Commands

COMMANDS VALID FOR ALL PRINTERS (Except 3800—3, —6 when in Page Mode)

No Operation	03
Space 1 Line Immediate	0B
Space 2 Lines Immediate	13
Space 3 Lines Immediate	1B
Block Data Check	73
Allow Data Check	7B
Skip to Channel 1 Immediate	8B
Skip to Channel 2 Immediate	93
Skip to Channel 3 Immediate	9B
Skip to Channel 4 Immediate	A3
Skip to Channel 5 Immediate	AB
Skip to Channel 6 Immediate	B3
Skip to Channel 7 Immediate	BB

IMPACT PRINTERS — ADDITIONAL COMMANDS

Printer	Column	Reference
1403—N1	A	GA24—3312
3203—5	B	GA33—1529
3211	B	GA24—3543
4248—1◄3211 mode►	B	GA24—3927
4248—2◄3211 mode►	B	GA24—3991
3262—5◄3262—1 mode►	C	GA24—3936
4245—1	C	GA33—1541
4245—12, —20	C	GA33—1579
3262—5◄4248 mode►	D	GA24—3936
4248—1◄native mode►	D	GA24—3927
4248—2◄native mode►	D	GA24—3991
6262—014	D	GA24—4234

Printer Channel Commands (Cont'd)

Skip to Channel 8 Immediate	E3
Skip to Channel 9 Immediate	CB
Skip to Channel 10 Immediate	D3
Skip to Channel 11 Immediate	DB
Skip to Channel 12 Immediate	F3
Write Without Spacing	01
Write and Space 1 Line	09
Write and Space 2 Lines	11
Write and Space 3 Lines	19
Write and Skip to Channel 1	89
Write and Skip to Channel 2	91
Write and Skip to Channel 3	99
Write and Skip to Channel 4	A1
Write and Skip to Channel 5	A9
Write and Skip to Channel 6	B1
Write and Skip to Channel 7	B9
Write and Skip to Channel 8	C1
Write and Skip to Channel 9	C9
Write and Skip to Channel 10	D1
Write and Skip to Channel 11	D9
Write and Skip to Channel 12	E1
Basic Sense	04

3800–3, –6 PAGE MODE COMMANDS
(See Note Y)

No Operation	03
Load Font Index	0F
Load Font Control	1F
Load Font	2F
Execute Order Any State	33
Load Font Equivalence	3F
Delete Font	4F
Begin Page Segment	5F
Delete Page Segment	6F
Include Page Segment	7F
Execute Order Home State	8F
Set Home State	97
Load Copy Control	9F
Begin Page	AF
End Page	BF
Load Page Description	CF
Begin Overlay	DF
Delete Overlay	EF
Write Factored Text Control	0D
Write Text	2D
Write Image Control	3D
Write Image	4D
End	5D
Load Page Position	6D
Basic Sense	04
Sense Intermediate Buffer	14
Sense Error Log	24
Sense ID	E4

3800–1 Reference: GA26–1635
3800–3, –6 Reference: GA32–0050

Note X: For 3800–3, –6 only, Set Home State (97) command will be accepted, but with command retry; the retry will succeed because Page Mode will have been sent.

Note Y: Other 3800–3, –6 commands accepted, but with command retry; the retry will succeed because Page Mode will have been reset.

Use Column A, B, C, or D.		ABCD
Unfold	23	.XXX
Execute Order	33	...X
Fold	43	.XXX
Load Forms Control Buffer	63	.XXX
Raise Cover	►6B	.12
Signal Attention	►6B	. 3
Skip to Channel 0 Immediate	83	.4.2
Clear Printer	87	..XX
UCS Gate Load	EB	X...
Load UCS Buffer and Fold	►F3	X..
Verify Band ID	►F3	..X
Load UCS Buffer (No Fold)	►FB	XXX
Verify Band ID	►FB	X
Release CU and Device	►14	5..
Sense Intermediate Buffer	►14	..X
Release CU, Reserve Device	34	5...
Reserve CU, Release Device	54	5...
Reserve CU and Device	74	5...
Release Device	94	5...
Reserve Device	B4	5...
Release CU	D4	5...
Sense ID	E4	..XX
Reserve CU	F4	5...
Read Band ID	►0A	. X
Diagnostic Read PLB	02	XX62
Diagnostic Write	05	7862
Diagnostic Check Read	06	XXX2
Diagnostic Gate	07	.XX2
Diagnostic Read UCS Buffer	►0A	.XX
Diagnostic Read FCB	12	.XXX

X = Valid . = Invalid
Blank = Not applicable.

1 = No action occurs (except 3211).
2 = No action occurs.
3 = No action occurs on 3265–5.
4 = 3211 only (no action occurs on 4248 ◄3211 mode►.
5 = Two–channel switch feature only.
6 = No action occurs (except 4245).
7 = 1403–N1 also uses command codes 0D, 15, 1D, 8D, 95, 9D, A5, AD, B5, BD, C5, CD, D5, DD, and E5.
8 = 3211 and 4248 ◄3211 mode► only.

3800 – Additional Commands
(Except 3800–3,–6 when in Page Mode; see Note X).

End of Transmission	07
Mark Form	17
Load Copy Number	23
Execute Order Any State	33
Initialize Printer	37
Load Forms Overlay Seq Control	43
Select Translate Table 0	47
Load Writable Char Gen Module	53
Select Translate Table 1	57
Load Forms Control Buffer	63
Select Translate Table 2	67
Select Translate Table 3	77
Load Translate Table	83
Clear Printer	87
Load Graphic Char Modification	25
Load Copy Modification	35
Sense Intermediate Buffer	14
Sense Error Log	24
Sense ID	E4

Magnetic-Tape Channel Commands

Channel Command	Hex Code	3420-3 3420-5 3420-7	3420-4 3420-6 3420-8	3422 3430	3480
No Operation	03	X	X	X	X
Rewind	07	X	X	X	X
Rewind Unload	0F	X	X	X	X
Modeset-1 (200/Odd/DC)	13	(a)	(b)		(b)
Erase Gap	17	X	X	X	X
Request Track-In-Error	1B	X	X	(c)	
Write Tape Mark	1F	X	X	X	X
Modeset-1 (200/Even/Normal)	23	(a)	(b)		(b)
Backspace Block	27	X	X	X	X
Modeset-1 (200/Even/TR)	2B	(a)	(b)		(b)
Backspace File	2F	X	X	X	X
Modeset-1 (200/Odd/Normal)	33	(a)	(b)		(b)
Forward Space Block	37	X	X	X	X
Modeset-1 (200/Odd/TR)	3B	(a)	(b)		(b)
Forward Space File	3F	X	X	X	X
Synchronize	43				X
Locate Block	4F				X
Modeset-1 (556/Odd/DC)	53	(d)	(b)		(b)
Suspend Multipath Reconnection	5B				(b)
Modeset-1 (556/Even/Normal)	63	(d)	(b)		(b)
Modeset-1 (556/Even/TR)	6B	(d)	(b)		(b)
Modeset-1 (556/Odd/Normal)	73	(d)	(b)		(b)
Modeset-1 (556/Odd/TR)	7B	(d)	(b)		(b)
Modeset-1 (800/Odd/DC)	93	(d)	(b)		(b)
Data Security Erase	97	X	X	X	X
Load Display	9F				X
Modeset-1 (800/Even/Normal)	A3	(d)	(b)		(b)
Modeset-1 (800/Even/TR)	AB	(d)	(b)		(b)
Set Path Group ID	AF				X
Modeset-1 (800/Odd/Normal)	B3	(d)	(b)		(b)
Assign	B7				X
Modeset-1 (800/Odd/TR)	BB	(d)	(b)		(b)
Modeset-2 (1 600 bpi PE)	┌►C3	(e)	(f)	X	-
Set Tape- Write-Immediate	└►C3	-	-	-	X
Unassign	C7				X
Modeset-2 (800 bpi NRZI)	CB	(e)	(b)		(b)
Modeset-2 (6250 bpi GCR)	D3		(f)	X	(b)
Mode Set	DB				X
Control Access	E3				X
Write	01	X	X	X	X
Read	02	X	X	X	X
Read Buffer	12				X
Read Block ID	22				X
Read Backward	OC	X	X	X	X
Basic Sense	04	X	X	X	X
Read Buffered Log	24				X
Sense Path Group ID	34				X
Release	D4	(g)	(g)	(g)	
Sense ID	E4			X	X
Reserve	F4	(g)	(g)	(g)	
Diagnostic Mode Set	OB	X	X		
Set Diagnose	4B	X	X	(c)	
Loop Write-To-Read	8B	X	X	X	

Notes:

a No action occurs unless 7-track feature is installed; if present, density set is 200 bpi by 3803-2 Tape control, 556 bpi by 3803-1.

b Valid command, but no action occurs.

c Invalid command for 3422.

d No action occurs unless 7-track feature is installed.

e No action occurs unless 800 bpi density feature is installed.

f No action occurs unless 1600 bpi density feature is installed.

g Requires two-channel switch feature, invalid for 3430.

For hex C3, the meaning depends on the machine type; hyphens signify that the alternative meaning is used.

Modeset-1 command (for 7-track drives): density (200, 556, 800 bpi)/parity (even, odd)/mode (Normal, DC=data converter, TR=translator). Modeset-2 command (for 9-track drives): density (800, 1600, 6250 bpi).

Sources:

3420-3, -5, -7(GA32-0020)
3420-4, -6, -8(GA32-0021)
3422(GA32-0089)
3430(GA32-0076)
3480(GA32-0042)

DASD Channel Commands

Channel Command	Hex Code	2305	3330 3340 3350	3350	3375 3380	3380	3370	Typical Transfer Count
		1	2	3	4	5	6	
Control								
No Operation	03	X	X	X	X	X	X	None
Seek	07	X	X	X	X	X		6
Seek Cylinder	0B	X	X	X	X	X		6
Space Count	0F	X	X		X	X		3
Recalibrate (No Op on 2305-2)	13	X	X		X	X		None
Restore (executed as No-Op)	17	X	X		X	X		None
Seek Head	1B	X	X	X	X	X		6
Set File Mask	1F	X	X		X	X		1
Set Sector (3340 RPS is optional)	23	X	X	X	X	X		1
Vary Sensing	27	X						1
Perform Subsystem Function	27					(u)		Variable
Orient (No-Op on 2305-2)	2B	X						None
Set High Performance Storage Limits	3B					(a)		10
Locate	43						X	8
Locate Record	47				(b)	(c)		16
Suspend Multipath Reconnection	5B				(d)	X		None
Define Extent	63				(b)	X	X	16
Set Subsystem Mode	87		(e)			(r)		2
Set Paging Parameters	8B			X				10
Discard Block	8F			X				2+(5xn)
Set Path Group ID	AF				(d)	X		12
Search								
Search Key Equal (*A9)	29	X	X		X	X		KL
Search ID Equal (*B1)	31	X	X	X	X	X		5
Search Home Address Equal (*B9)	39	X	X		X	X		4
Search Key High (*C9)	49	X	X		X	X		KL
Search ID High (*D1)	51	X	X		X	X		5
Search Key Equal or High (*E9)	69	X	X		X	X		KL
Search ID Equal or High (*F1)	71	X	X		X	X		5
Read								
Read Initial Program Load	02	X	X		X	X	X	DL or 512
Read Data (*86)	06	X	X	X	X	X		DL
Read Key & Data (*BE)	0E	X	X		X	X		KL+DL
Read Count (*92)	12	X	X		X	X		8
Read Record Zero (*96)	16	X	X		X	X		8+KL+DL
Read Home Address (*9A)	1A	X	X		X	X		5
Read Count Key & Data (*9E)	1E	X	X		X	X		8+KL+DL
Read Sector (3340 RPS is optional)	22	X	X		X	X		1
Read Subsystem Data	3E					(u)		Variable
Read	42						X	512xn
Read Message ID	4E					(v)		11
Read Multiple Count Key & Data	5E		(f)		X	X		nx(8+KL+DL)
Read Track	DE					(t)		Variable
Read Configuration Data	FA					(u)		256
Write								
Write Special Count Key & Data	01	X	X					8+KL+DL
Write Data	05	X	X	X	X	X		DL
Write Key & Data	0D	X	X		X	X		KL+DL
Erase	11	X	X		X	X		8+KL+DL
Write Record Zero	15	X	X		X	X		8+KL+DL
Write Home Address	19	X	X		X	X		5, 7, or 11
Write Count Key & Data	1D	X	X		X	X		8+KL+DL
Write	41						X	512xn
Write Update Data	85				(b)	(c)		DL
Write Update Key & Data	8D				(b)	(c)		KL+DL
Write Count Key & Data Next Track	9D				(b)	(c)		8+KL+DL
		1	2	3	4	5	6	

DASD Channel Commands (Cont'd)

Channel Command	Hex Code	2305	3330 3340 3350	3350	3375 3380	3380	3370	Typical Transfer Count
		1	2	3	4	5	6	
Sense								
Basic Sense	04	X	X	X	X	X	X	24 or 32
Unconditional Reserve (g)	14		(h,j)		(d,h,j)	X	(h,j)	24 or 32
Read Buffered Log	24	X						128
Sense Path Group ID	34				(d)	X		12
Reset Allegiance	44					(t)		32
Sense Subsystem Status	54		(k)			X		40
Read Device Characteristics	64					(c)	X	32
Sense Subsystem Counts	74		(k)			(s)		80
Device Release	94	(j)	(h,j)		(d,h,j)	X	(h,j)	24
Read and Reset Buffered Log	A4		X		X	X	X	24 or 32
Device Reserve	B4	(j)	(h,j)		(d,h,j)	X	(h,j)	24 or 32
Sense ID (f)	E4		X	X	X	X	X	7
Diagnostic								
Diagnostic Write Home Address	09				X	X		27 or 28
Diagnostic Read Home Address	0A				X	X		27 or 28
Diagnostic Sense #	44	X	(m)					16 or 512
Diagnostic Load	53	X	(m)					1
Diagnostic Write	73	X	(m)					8 or 512
Diagnostic Sense/Read	C4		(p)		X	X	X	Variable
Diagnostic Control	F3		(q)		(q)	X	X	4+n
		1	2	3	4	5	6	

a Valid only for 3880-13
b Speed matching buffer feature
c Not valid for 3880-13
d Dynamic path selection (only valid on 3380-AA4, -AD4, -AE4, -AJ4, AK4 strings)
e Valid only for 3880-21
f Not valid for 3330/3333 on ISC-SA; 3830-2, -3, and ISC require 3344/3350 microcode
g Not valid on ISC-SA; not valid on 3330/3333, 3340/3344
h String-switching feature
j Channel-switching feature
k Valid only for 3880-11 paging director and 3880-21
m Not valid on 3880-21
p Valid only for 3880-1, -2, -11, -21
q Valid only for 3330/3350 on 3880-1, -2, and for 3380 on 3880-2, -3 without 3380-speed-matching-buffer feature
r Valid only for 3880-13, -23, and 3990-3
s Valid only for 3880-13, -23
t Not valid for 3880-13, -23
u Valid only for 3990
v Valid only for 3990-3
* Multi-track command codes (standard)
# Also called “Read Diagnostic Status 1”

Code Assignments

Code Table

Dec	Hex	EBCDIC	AS-CII	ISO (1) -8	IBM-PC	BookMaster™ Symbol Names(2)
0	00	NUL	NUL	NUL	NUL	
1	01	SOH	SOH	SOH	SOH ☺	face
2	02	STX	STX	STX	STX ☻	FACE
3	03	ETX	ETX	ETX	ETX ♥	HEART
4	04	SEL	EOT	EOT	EOT ♦	DIAMOND
5	05	HT	ENQ	ENQ	ENQ ♣	CLUB
6	06	RNL	ACK	ACK	ACK ♠	SPADE
7	07	DEL	BEL	BEL	BEL •	bullet
8	08	GE	BS	BS	BS ◘	revbul
9	09	SPS	HT	HT	HT ○	circle
10	0A	RPT	LF	LF	LF ◙	revcir
11	0B	VT	VT	VT	VT ♂	male
12	0C	FF	FF	FF	FF ♀	female
13	0D	CR	CR	CR	CR ♪	note18
14	0E	SO	SO	SO	SO ♫	note1616
15	0F	SI	SI	SI	SI ☼	sun
16	10	DLE	DLE	DLE	DLE ►	rahead
17	11	DC1	DC1	DC1	DC1 ◄	lahead
18	12	DC2	DC2	DC2	DC2 ↕	udarrow
19	13	DC3	DC3	DC3	DC3 ‼	dblxclam
20	14	RES/ENP	DC4	DC4	DC4 ¶	par
21	15	NL	NAK	NAK	NAK §	section
22	16	BS	SYN	SYN	SYN	overline
23	17	POC	ETB	ETB	ETB ↨	udarrowus
24	18	CAN	CAN	CAN	CAN ↑	uarrow
25	19	EM	EM	EM	EM ↓	darrow
26	1A	UBS	SUB	SUB	IFS →	rarrow
27	1B	CU1	ESC	ESC	ESC ←	larrow
28	1C	IFS	FS	IFS	DEL ∟	lnotusd
29	1D	IGS	GS	IGS	GS ↔	lrarrow
30	1E	IRS	RS	IRS	RS ▲	uahead
31	1F	ITB/IUS	US	IUS	US ▼	dahead
32	20	DS	SP	SP	SP	
33	21	SOS	!	!	!	xclam
34	22	FS	"	"	"	sdq
35	23	WUS	#	#	#	numsign
36	24	BYP/INP	$	$	$	dollar
37	25	LF	%	%	%	percent
38	26	ETB	&	&	&	amp
39	27	ESC	'	'	'	ssq(3)

BookMaster is a trademark of the International Business Machines Corporation.

Dec	Hex	EBCDIC	AS-CII	ISO (1) -8	IBM-PC	BookMaster Symbol Names(2)
40	28	SA	(	(	(	lpar
41	29	SFE	)	)	)	rpar
42	2A	SM/SW	*	*	*	asterisk
43	2B	CSP	+	+	+	plus
44	2C	MFA	,	,	,	comma
45	2D	ENQ	-	-	-	hyphen or minus
46	2E	ACK	.	.	.	period
47	2F	BEL	/	/	/	divslash or slash
48	30		0	0	0	
49	31		1	1	1	
50	32	SYN	2	2	2	
51	33	IR	3	3	3	
52	34	PP	4	4	4	
53	35	TRN	5	5	5	
54	36	NBS	6	6	6	
55	37	EOT	7	7	7	
56	38	SBS	8	8	8	
57	39	IT	9	9	9	
58	3A	RFF	:	:	:	colon
59	3B	CU3	;	;	;	semi
60	3C	DC4	<	<	<	lt
61	3D	NAK	=	=	=	eq
62	3E		>	>	>	gt
63	3F	SUB	?	?	?	quest
Dec	Hex	See Next Page	See Above			See Above
64	40	SP RSP	@	@	@	atsign
65	41	SP RSP	A	A	A	
66	42	SP RSP â	B	B	B	ac
67	43	SP RSP ä	C	C	C	ae
68	44	SP RSP à	D	D	D	ag
69	45	SP RSP á	E	E	E	aa
70	46	SP RSP ã	F	F	F	at
71	47	SP RSP å	G	G	G	ao
72	48	SP RSP ç	H	H	H	cc
73	49	SP RSP ñ	I	I	I	nt
74	4A	¢	J	J	J	cent, lbrk
75	4B	.	K	K	K	period
76	4C	<	L	L	L	lt
77	4D	(	M	M	M	lpar
78	4E	+	N	N	N	plus
79	4F	|	O	O	O	vbar, xclam
80	50	&	P	P	P	amp
81	51	SP RSP é	Q	Q	Q	ea
82	52	SP RSP ê	R	R	R	ec
83	53	SP RSP ë	S	S	S	ee

Code Assignments (Cont'd)

Dec	Hex	EBCDIC(4) 81C	94C	037	500	1047	AS-CII	ISO -8	IBM -PC	BookMaster Symbol Names(2)
84	54			è	è	è	T	T	T	eg
85	55			í	í	í	U	U	U	ia
86	56			î	î	î	V	V	V	ic
87	57			ï	ï	ï	W	W	W	ie
88	58			ì	ì	ì	X	X	X	ig
89	59			ß	ß	ß	Y	Y	Y	ss
90	5A		!	!	]	!	Z	Z	Z	xclam, rbrk
91	5B		$	$	$	$	[	[	[	dollar, lbrk
92	5C	*	*	*	*	*	\	\	\	asterisk, bslash
93	5D	)	)	)	)	)	]	]	]	rpar, rbrk
94	5E	;	;	;	;	;	^	^	^	semi, hat
95	5F		¬	¬	^	^	_	_	_	lnot, hat, us
96	60	-	-	-	-	-	`	`	`	hyphen or minus, grave
97	61	/	/	/	/	/	a	a	a	divslash or slash
98	62			Â	Â	Â	b	b	b	Ac
99	63			Ä	Ä	Ä	c	c	c	Ae
100	64			À	À	À	d	d	d	Ag
101	65			Á	Á	Á	e	e	e	Aa
102	66			Ã	Ã	Ã	f	f	f	At
103	67			Å	Å	Å	g	g	g	Ao
104	68			Ç	Ç	Ç	h	h	h	Cc
105	69			Ñ	Ñ	Ñ	i	i	i	Nt
106	6A		¦	¦	¦	¦	j	j	j	splitvbar
107	6B	,	,	,	,	,	k	k	k	comma
108	6C	%	%	%	%	%	l	l	l	percent
109	6D	_	_	_	_	_	m	m	m	us
110	6E	>	>	>	>	>	n	n	n	gt
111	6F	?	?	?	?	?	o	o	o	quest
112	70			ø	ø	ø	p	p	p	os
113	71			É	É	É	q	q	q	Ea
114	72			Ê	Ê	Ê	r	r	r	Ec
115	73			Ë	Ë	Ë	s	s	s	Ee
116	74			È	È	È	t	t	t	Eg
117	75			Í	Í	Í	u	u	u	Ia
118	76			Î	Î	Î	v	v	v	Ic
119	77			Ï	Ï	Ï	w	w	w	Ie
120	78			Ì	Ì	Ì	x	x	x	Ig
121	79			`	`	`	y	y	y	grave
122	7A	:	:	:	:	:	z	z	z	colon
123	7B		#	#	#	#	{	{	{	numsign, lbrc
124	7C		@	@	@	@	|	|	|	atsign, vbar
125	7D	'	'	'	'	'	}	}	}	ssq(3), rbrc
126	7E	=	=	=	=	=	~	~	~	eq, eqv
127	7F	"	"	"	"	"	DEL	⌂	⌂	sdq, house

Dec	Hex	EBCDIC(4) 81C	94C	037	500	1047	ISO -8	IBM-PC 437	850	BookMaster Symbol Names(2)
128	80			Ø	Ø	Ø		Ç	Ç	Os, Cc
129	81	a	a	a	a	a		ü	ü	ue
130	82	b	b	b	b	b	BPH	é	é	ea
131	83	c	c	c	c	c	NBH	â	â	ac
132	84	d	d	d	d	d	IND	ä	ä	ae
133	85	e	e	e	e	e	NEL	à	à	ag
134	86	f	f	f	f	f	SSA	å	å	ao
135	87	g	g	g	g	g	ESA	ç	ç	cc
136	88	h	h	h	h	h	HTS	ê	ê	ec
137	89	i	i	i	i	i	HTJ	ë	ë	ee
138	8A			«	«	«	VTS	è	è	odqf, eg
139	8B			»	»	»	PLD	ë	ë	cdqf, ee
140	8C			ð	ð	ð	PLU	î	î	eth, ic
141	8D			ý	ý	ý	RI	ì	ì	ya, ig
142	8E			þ	þ	þ	SS2	Ä	Ä	thorn, Ae
143	8F			±	±	±	SS3	Å	Å	pm, Ao
144	90			°	°	°	DCS	É	É	degree, Ea
145	91	j	j	j	j	j	PU1	æ	æ	aelig
146	92	k	k	k	k	k	PU2	Æ	Æ	AElig
147	93	l	l	l	l	l	STS	ô	ô	oc
148	94	m	m	m	m	m	CCH	ö	ö	oe
149	95	n	n	n	n	n	MW	ò	ò	og
150	96	o	o	o	o	o	SPA	û	û	uc
151	97	p	p	p	p	p	EPA	ù	ù	ug
152	98	q	q	q	q	q	SOS	ÿ	ÿ	ye
153	99	r	r	r	r	r		Ö	Ö	Oe
154	9A			a	a	a	SCI	Ü	Ü	aus, Ue
155	9B			º	º	º	CSI	¢	ø	ous, cent, os
156	9C			æ	æ	æ	ST	£	£	aelig, Lsterling
157	9D						OSC	¥	Ø	cedilla, yen, Os
158	9E			Æ	Æ	Æ	PM	Pt	×	AElig, peseta, mult
159	9F			¤	¤	¤	ACP	*f*	*f*	currency, fnof(5)
160	A0			*μ*	*μ*	*μ*	RSP	á	á	mu(6), aa
161	A1			~	~	~	¡	í	í	tilde, inve, ia
162	A2	s	s	s	s	s	¢	ó	ó	cent, oa
163	A3	t	t	t	t	t	£	ú	ú	Lsterling, ua
164	A4	u	u	u	u	u	¤	ñ	ñ	currency, nt
165	A5	v	v	v	v	v	¥	Ñ	Ñ	yen, Nt
166	A6	w	w	w	w	w	¦	a	a	splitvbar, aus
167	A7	x	x	x	x	x	§	º	º	section, ous
168	A8	y	y	y	y	y	¨	¿	¿	umlaut, invq
169	A9	z	z	z	z	z	©	⌐	®	copyr, lnotrev, regtm
170	AA			¡	¡	¡	a	¬	¬	inve, aus, lnot
171	AB			¿	¿	¿	«	½	½	invq, odqf, frac12

Code Assignments (Cont'd)

Dec	Hex	EBCDIC(4) 81C	94C	037	500	1047	ISO -8	IBM-PC 437	850	BookMaster Symbol Names(2)
172	AC			Đ	Đ	Đ	¬	¼	¼	Dstroke or Eth, lnot, frac14
173	AD			Ý	Ý	[	SHY	¡	¡	Ya, lbrk, inve
174	AE			þ	þ	þ	®	«	«	Thorn, regtm, odqf
175	AF			®	®	®	¯	»	»	regtm, overline, cdqf
176	B0			^	¢	¬	°	⁞	⁞	hat, cent, lnot, degree, box14
177	B1			£	£	£	±	▒	▒	Lsterling, pm, box12
178	B2			¥	¥	¥	²	▓	▓	yen, sup2, box34
179	B3			·	·	·	³	│	│	smultdot, sup3, bxv
180	B4			©	©	©	´	┤	┤	copyr, acute, bxrj
181	B5			§	§	§	µ	╡	Á	section, mu(6), bx1012, Aa
182	B6			¶	¶	¶	¶	╢	Â	par, bx2021, Ac
183	B7			¼	¼	¼	·	╖	À	frac14, smultdot, bx0021, Ag
184	B8			½	½	½	¸	╕	©	frac12, cedilla, bx0012, copyr
185	B9			¾	¾	¾	¹	╣	╣	frac34, sup1, bx2022
186	BA			[	¬	Ý	º	║	║	lbrk, lnot, Ya, ous, bx2020
187	BB			]	|	¨	»	╗	╗	rbrk, vbar, umlaut, cdqf, bx0022
188	BC			¯	¯	¯	¼	╝	╝	overline, frac14, bx2002
189	BD			¨	¨	]	½	╜	¢	umlaut, rbrk, frac12, bx2001, cent
190	BE			´	´	´	¾	╛	¥	acute, frac34, bx1002, yen
191	BF			×	×	×	¿	┐	┐	mult, invq, bxur
192	C0			{	{	{	À	└	└	lbrc, Ag, bxll
193	C1	A	A	A	A	A	Á	┴	┴	Aa, bxbj
194	C2	B	B	B	B	B	Â	┬	┬	Ac, bxtj
195	C3	C	C	C	C	C	Ã	├	├	At, bxlj
196	C4	D	D	D	D	D	Ä	─	─	Ae, bxh
197	C5	E	E	E	E	E	Å	┼	┼	Ao, bxcj
198	C6	F	F	F	F	F	Æ	╞	ã	AElig, bx1210, at
199	C7	G	G	G	G	G	Ç	╟	Ã	Cc, bx2120, At
200	C8	H	H	H	H	H	È	╚	Ã	Eg, bx2200, Ag
201	C9	I	I	I	I	I	É	╔	╔	Ea, bx0220
202	CA	SHY	SHY	SHY	SHY	SHY	Ê	╩	╩	Ec, bx2202
203	CB			ô	ô	ô	Ë	╤	╤	oc, Ee, bx0222
204	CC			ö	ö	ö	Ì	╠	╠	oe, Ig, bx2220
205	CD			ò	ò	ò	Í	═	═	og, Ia, bx0202
206	CE			ó	ó	ó	Î	╬	╬	oa, Ic, bx2222
207	CF			õ	õ	õ	Ï	╧	¤	ot, Ie, bx1202, currency

Dec	Hex	EBCDIC(4) 81C	94C	037	500	1047	ISO -8	IBM-PC 437	850	BookMaster Symbol Names(2)
208	D0			}	}	}	Ð	╨	ð	rbrc, Dstroke or Eth, bx2101, eth
209	D1	J	J	J	J	J	Ñ	╤	Ð	Nt, bx0212, Dstroke or Eth
210	D2	K	K	K	K	K	Ò	╥	Ê	Og, bx0121, Ec
211	D3	L	L	L	L	L	Ó	╙	Ë	Oa, bx2100, Ee
212	D4	M	M	M	M	M	Ô	╘	È	Oc, bx1200, Eg
213	D5	N	N	N	N	N	Õ	╒	ı	Ot, bx0210, idotless
214	D6	O	O	O	O	O	Ö	╓	Í	Oe, bx0120, Ia
215	D7	P	P	P	P	P	×	╫	Î	mult, bx2121, Ic
216	D8	Q	Q	Q	Q	Q	Ø	╪	Ï	Os, bx1212, Ie
217	D9	R	R	R	R	R	Ù	┘	┘	Ug, bxlr
218	DA			¹	¹	¹	Ú	┌	┌	sup1, Ua, bxul
219	DB			û	û	û	Û	█	█	uc, Uc, BOX
220	DC			ü	ü	ü	Ü	▄	▄	ue, Ue, BOXBOT
221	DD			ù	ù	ù	Ý	▌	¦	ug, Ya, BOXLEFT, splitvbar
222	DE			ú	ú	ú	þ	▐	Ì	ua, thorn, BOXRIGHT, Ig
223	DF			ÿ	ÿ	ÿ	ß	▀	▀	ye, ss, BOXTOP
224	E0			\	\	\	à	*α*	Ó	bslash, ag, alpha, Oa
225	E1		NSP	÷	÷	÷	á	ß	ß	div, aa, ss
226	E2	S	S	S	S	S	â	Γ	Ô	ac, Gamma, Oc
227	E3	T	T	T	T	T	ã	*π*	Ò	at, pi, Og
228	E4	U	U	U	U	U	ä	Σ	õ	ae, Sigma, ot
229	E5	V	V	V	V	V	å	*σ*	Õ	ao, sigma, Ot
230	E6	W	W	W	W	W	æ	*μ*	*μ*	aelig, mu(6)
231	E7	X	X	X	X	X	ç	*τ*	þ	cc, tau, thorn
232	E8	Y	Y	Y	Y	Y	è	Φ	Þ	eg, Phi, Thorn
233	E9	Z	Z	Z	Z	Z	é	Θ	Ú	ea, Theta(5), Ua
234	EA			²	²	²	ê	Ω	Û	sup2, ec, Omega, Uc
235	EB			Ô	Ô	Ô	ë	*δ*	Ù	Oc, ee, delta, Ug
236	EC			Ö	Ö	Ö	ì	∞	ý	Oe, ig, infinity, ya
237	ED			Ò	Ò	Ò	í	*ϕ*	Ý	Og, ia, phi, Ya
238	EE			Ó	Ó	Ó	î	*ϵ*	¯	Oa, ic, epsilon, overline
239	EF			Õ	Õ	Õ	ï	∩	´	Ot, ie, intersect, acute
240	F0	0	0	0	0	0	ð	≡	SHY	eth, identical
241	F1	1	1	1	1	1	ñ	±	±	nt, pm
242	F2	2	2	2	2	2	ò	≥	‗	og, ge, eq
243	F3	3	3	3	3	3	ó	≤	¾	oa, le, frac34
244	F4	4	4	4	4	4	ô	⌠	¶	oc, inttop, par
245	F5	5	5	5	5	5	õ	⌡	§	ot, intbot, section
246	F6	6	6	6	6	6	ö	÷	÷	oe, div
247	F7	7	7	7	7	7	÷	≈	¸	div, nearly(5), cedilla

Code Assignments (Cont'd)

Dec	Hex	EBCDIC(4) 81C	94C	037	500	1047	ISO -8	IBM-PC 437	850	BookMaster Symbol Names(2)
248	F8	8	8	8	8	8	ø	°	°	os, degree
249	F9	9	9	9	9	9	ù	•	¨	ug, lmultdot, umlaut
250	FA			³	³	³	ú	·	·	sup3, ua, smultdot
251	FB			Û	Û	Û	û	√	¹	Uc, uc, sqrt, sup1
252	FC			Ü	Ü	Ü	ü	ⁿ	³	Ue, ue, supn, sup3
253	FD			Ù	Ù	Ù	ý	²	²	Ug, ya, sup2
254	FE			Ú	Ú	Ú	þ	■	■	Ua, thorn, sqbul
255	FF	EO	EO	EO	EO	EO	ÿ	RSP	RSP	ye

(1) The ASCII controls and graphics are from ANSI X3.4. The ISO-8 controls are from ISO 6429, and the graphics are from ISO 8859-1. The ISO-8 graphics are code page 00819, named ISO/ANSI Multilingual. IBM-PC controls and graphics are shown. The graphics are common to code page 00437, named Personal Computer, and code page 00850, named Personal Computer - Multilingual Page. Code pages 00437 and 00850 are shown separately beginning at X'80', after which they diverge in content.
(2) The symbol names shown are to be preceded by an ampersand (&) and followed by a period (.) to form a symbol. Source: SC34-5009.
(3) ASCII, ISO-8, and IBM-PC X'27' and EBCDIC X'7D' are an apostrophe having the appearance of a straight single quote. The BookMaster “apos” produces a character having the appearance of an accent acute.
(4) Five columns of EBCDIC graphics are shown. The first is the 81-character character set 0640, called the syntactic character set, that is mapped the same on all EBCDIC code pages. The second is the standard IBM 94-character character set mapped on code page 00037. The third is code page 00037, named USA/Canada - CECP (Country Extended Code Page). The fourth is code page 00500, named International #5. The fifth is code page 01047, named Latin 1/Open Systems. Code pages 00037, 00500, 01047, and 00819 (ISO-8) all map the 189-character character set 0697. Source: SE09-8002.
(5) f, ≈, and Θ are of nonstandard width.
(6) EBCDIC X'A0' and ISO-8 X'B5' are micro but resemble mu. The BookMaster “usec” produces a character of nonstandard width.

Control Character Representations

ACK	Acknowledge
BEL	Bell
BS	Backspace
BYP	Bypass
CAN	Cancel
CR	Carriage Return
CSP	Control Sequence Prefix
CU1	Customer Use 1
CU3	Customer Use 3
DC1	Device Control 1
DC2	Device Control 2
DC3	Device Control 3
DC4	Device Control 4
DEL	Delete
DLE	Data Link Escape
DS	Digit Select
EM	End of Medium
ENP	Enable Presentation
ENQ	Enquiry
EO	Eight Ones
EOT	End of Transmission
ESC	Escape
ETB	End of Transmission Block
ETX	End of Text
FF	Form Feed
FS	Field Separator
GE	Graphic Escape
HT	Horizontal Tab
IFS	Interchange File Separator
IGS	Interchange Group Separator
INP	Inhibit Presentation
IR	Index Return
IRS	Interchange Record Separator
IT	Indent Tab
ITB	Intermediate Transmission Block
IUS	International Unit Separator
LF	Line Feed
MFA	Modify Field Attribute
NAK	Negative Acknowledge
NBS	Numeric Backspace
NL	New Line
NUL	Null
POC	Program-Operator Communication
PP	Presentation Position
RES	Restore
RFF	Required Form Feed
RNL	Required New Line
RPT	Repeat
SA	Set Attribute
SBS	Subscript
SEL	Select
SFE	Start Field Extended
SI	Shift In
SM	Set Mode
SO	Shift Out
SOH	Start of Heading
SOS	Start of Significance
SPS	Superscript
STX	Start of Text
SUB	Substitute
SW	Switch
SYN	Synchronous Idle
TRN	Transparent
UBS	Unit Backspace
VT	Vertical Tab
WUS	Word Underscore

Additional ISO-8 Control Character Representations

APC	Application Program Command	OSC	Operating System Command
BPH	Break Permitted Here	PLD	Partial Line Down
CCH	Cancel Character	PLU	Partial Line Up
CSI	Control Sequence Introducer	PM	Privacy Message
DCS	Device Control String	PU1	Private Use One
EPA	End of Guarded Area	PU2	Private Use Two
ESA	End of Selected Area	RI	Reverse Line Feed (or Index)
HTJ	Character Tabulation with Justification	SCI	Single Character Introducer
		SOS	Start of String
HTS	Character Tabulation Set	SPA	Start of Guarded Area
IFS	Information Separator Four	SSA	Start of Selected Area
IGS	Information Separator Three	SS2	Single Shift Two
IND	Index	SS3	Single Shift Three
IRS	Information Separator Two	ST	String Terminator
MW	Message Waiting	STS	Set Transmit State
NBH	No Break Here	US	Information Separator One
NEL	Next Line	VTS	Line Tabulation Set

Formatting Character Representations

NSP	Numeric Space	SP	Space
RSP	Required Space	SHY	Syllable Hyphen

Two-Character BSC Data Link Controls

Function	EBCDIC	ASCII
ACK-0	DLE,X'70'	DLE,0
ACK-1	DLE,X'61'	DLE,1
WACK	DLE,X'68'	DLE,;
RVI	DLE,X'7C'	DLE,<

Commonly Used Editing Pattern Characters

Code (Hex)	Meaning	Code (Hex)	Meaning
20	Digit selector	5B	Dollar sign
21	Start of significance	5C	Asterisk
22	Field separator	6B	Comma
40	Blank	C3D9	CR (credit)
4B	Period	C4C2	DB (debit)

ANSI-Defined Printer Control Characters

(**A** in **RECFM** field of DCB)

Code	Action before Printing Record
blank	Space 1 line
0	Space 2 lines
-	Space 3 lines
+	Suppress space
1	Skip to line 1 on new page

Hexadecimal and Decimal Conversion

From hex: locate each hex digit in its corresponding column position and note the decimal equivalents. Add these to obtain the decimal value.

From decimal: (1) locate the largest decimal value in the table that will fit into the decimal number to be converted, and (2) note its hex equivalent and hex column position. (3) Find the decimal remainder. Repeat the process on this and subsequent remainders.

Note: Decimal, hexadecimal, and binary equivalents of all numbers from 0 to 255 are listed in the code tables.

Word															
Halfword								Halfword							
Byte				Byte				Byte				Byte			
Bits: 0123		4567		0123		4567		0123		4567		0123		4567	
Hex	Dec	Hex	Dec	Hex	Dec	Hex	Dec	Hex	Dec	Hex	Dec	Hex	Dec	Hex	Dec
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	268,435,456	1	16,777,216	1	1,048,576	1	65,536	1	4,096	1	256	1	16	1	1
2	536,870,912	2	33,554,432	2	2,097,152	2	131,072	2	8,192	2	512	2	32	2	2
3	805,306,368	3	50,331,648	3	3,145,728	3	196,608	3	12,288	3	768	3	48	3	3
4	1,073,741,824	4	67,108,864	4	4,194,304	4	262,144	4	16,384	4	1,024	4	64	4	4
5	1,342,177,280	5	83,886,080	5	5,242,880	5	327,680	5	20,480	5	1,280	5	80	5	5
6	1,610,612,736	6	100,663,296	6	6,291,456	6	393,216	6	24,576	6	1,536	6	96	6	6
7	1,879,048,192	7	117,440,512	7	7,340,032	7	458,752	7	28,672	7	1,792	7	112	7	7
8	2,147,483,648	8	134,217,728	8	8,388,608	8	524,288	8	32,768	8	2,048	8	128	8	8
9	2,415,919,104	9	150,994,944	9	9,437,184	9	589,824	9	36,864	9	2,304	9	144	9	9
A	2,684,354,560	A	167,772,160	A	10,485,760	A	655,360	A	40,960	A	2,560	A	160	A	10
B	2,952,790,016	B	184,549,376	B	11,534,336	B	720,896	B	45,056	B	2,816	B	176	B	11
C	3,221,225,472	C	201,326,592	C	12,582,912	C	786,432	C	49,152	C	3,072	C	192	C	12
D	3,489,660,928	D	218,103,808	D	13,631,488	D	851,968	D	53,248	D	3,328	D	208	D	13
E	3,758,096,384	E	234,881,024	E	14,680,064	E	917,504	E	57,344	E	3,584	E	224	E	14
F	4,026,531,840	F	251,658,240	F	15,728,640	F	983,040	F	61,440	F	3,840	F	240	F	15
8		7		6		5		4		3		2		1	

Powers of 2 and 16

m	*n*	2^m and 16^n
0	0	1
1		2
2		4
3		8
4	1	16
5		32
6		64
7		128
8	2	256
9		512
10		1 024
11		2 048
12	3	4 096
13		8 192
14		16 384
15		32 768
16	4	65 536
17		131 072
18		262 144
19		524 288
20	5	1 048 576
21		2 097 152
22		4 194 304
23		8 388 608
24	6	16 777 216
25		33 554 432
26		67 217 728
27		134 217 728
28	7	268 435 456
29		536 870 912
30		1 073 741 824
31		2 147 483 648
32	8	4 294 967 296
33		8 589 934 592
34		17 179 869 184
35		34 359 738 368
36	9	68 719 476 736
37		137 438 953 472
38		274 877 906 944
39		549 755 813 888
40	10	1 099 511 627 776
41		2 199 023 255 552
42		4 398 046 511 104
43		8 796 093 022 208
44	11	17 592 186 044 416
45		35 184 372 088 832
46		70 368 744 177 664
47		140 737 488 355 328
48	12	281 474 976 710 656
49		562 949 953 421 312
50		1 125 899 906 842 624
51		2 251 799 813 685 248
52	13	4 503 599 627 370 496
53		9 007 199 254 740 992
54		18 014 398 509 481 984
55		38 028 797 018 963 968
56	14	72 057 594 037 927 936
57		144 115 188 075 855 872
58		288 230 376 151 711 744
59		576 460 752 303 423 488
60	15	1 152 921 504 606 846 976
61		2 305 843 009 213 693 952
62		4 611 686 018 427 387 904
63		9 223 372 036 854 775 808

Symbol	Value
K (kilo)	$1,024=2^{10}$
M (mega)	$1,048,576=2^{20}$
G (giga)	$1,073,741,824=2^{30}$
T (tera)	$1,099,511,627,776=2^{40}$

Hexadecimal and Decimal Conversion (Cont'd)

File Number: S390-00

Printed in the United States of America
on recycled paper containing 10%
recovered post-consumer fiber.