

z/Architecture



Reference Summary

z/Architecture



Reference Summary

Second Edition (June, 2003)

This revision differs from the previous edition by containing instructions related to the facilities marked by a bar under "Facility" in "Preface," fields related to new I/O facilities, and minor corrections and clarifications. Additionally, in the sections "Machine Instruction Formats" and "Machine Instructions by Operation Code," multiple instances of the same instruction format are identified by a subscript. 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.

© Copyright International Business Machines Corporation
2001-2003. All rights reserved.

US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Preface

This publication is intended primarily for use by z/Architecture™ assembler-language application programmers. It contains basic machine information summarized from the *IBM z/Architecture Principles of Operation*, SA22-7832, about the zSeries™ processors. It also contains frequently used information from *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 for MVS & VM & VSE 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.

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

Facility	Instruction
DAT enhancement	CSPG, IDTE
Expanded storage	PGIN, PGOUT
Extended translation 2	CLCLU, MVCLU, PKA, PKU, TP, TROO, TROT, TRTO, TRTT, UNPKA, UNPKU
HFP multiply-and-add/subtract	MAD, MADR, MAE, MAER, MSD, MSDR, MSE, MSER
Long displacement	AHY, ALY, AY, CDSY, CHY, CLLY, CLMY, CLY, CSY, CVBY, CVDY, CY, ICMY, ICY, LAMY, LAY, LB, LDY, LEY, LGB, LHY, LMY, LRAY, LY, MSY, MVIY, NIY, NY, OIY, OY, SHY, SLY, STAMY, STCMY, STCY, STDY, STEY, STHY, STMY, STY, SY, TMY, XIY, XY
Message-security assist	KM, KMC, KIMD, KLMD, KMAC

For information about Enterprise Systems Architecture/390™ (ESA/390™) architecture, refer to *IBM Enterprise Systems Architecture/390 Principles of Operation*, SA22-7201, and *IBM Enterprise Systems Architecture/390 Reference Summary*, SA22-7209.

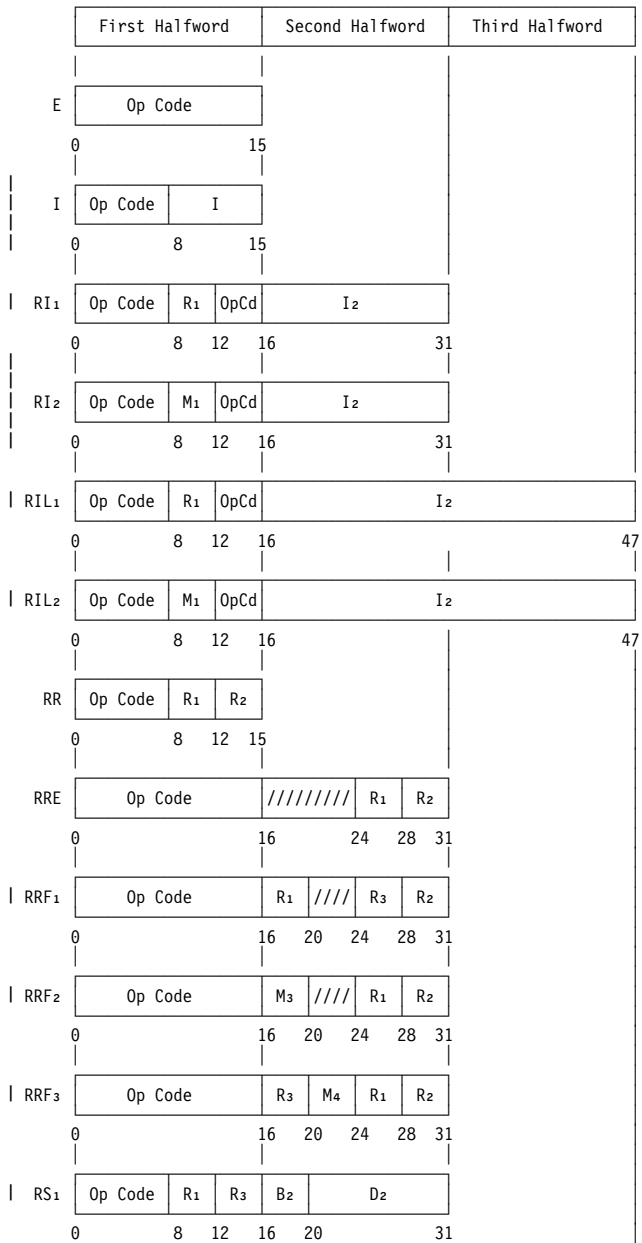
Contents

Preface	iii
Machine Instruction Formats	2
Machine Instructions by Mnemonic	4
Machine Instructions by Operation Code	12
Condition Codes	15
Operand of Store Clock	19
Operand of Store Clock Extended	19
Assembler Instructions	19
Extended-Mnemonic Instructions for Branch on Condition	21
Extended-Mnemonic Instructions for Relative-Branch Instructions	21
CNOP Alignment	22
Summary of Constants	22
Fixed Storage Locations	22
External-Interruption Codes	24
Program-Interruption Codes	24
Translation-Exception Identification	25
Data-Exception Code (DXC)	25
Control Registers	26
Floating-Point-Control (FPC) Register	27
Program-Status Word (PSW)	28
z/Architecture PSW	28
ESA/390 PSW	28
Dynamic Address Translation	29
Virtual-Address Format	29
Address-Space-Control Element (ASCE)	29
Table Values	29
Region-Table Entry (RTE)	30
Segment-Table Entry (STE)	30
Page-Table Entry (PTE)	30
ASN Translation	30
Address-Space Number (ASN)	30
ASN-First-Table Entry	31
ASN-Second-Table Entry (ASTE)	31
PC-Number Translation	32
Program-Call Number	32
Linkage-Table Entry (LTE)	32
Entry-Table Entry (ETE)	32
Access-Register Translation	33
Access-List-Entry Token (ALET)	33
Dispatchable-Unit-Control Table (DUCT)	33
Access-List Entry (ALE)	34
Linkage-Stack Entries	35
Entry Descriptor	35
Header Entry (Entry Type 0001001)	35
Trailer Entry (Entry Type 0001010)	35
Branch State Entry (Entry Type 0001100) and Program-Call State Entry (Entry Type 0001101)	36
Trapping	37
Trap Control Block	37
Trap Save Area	38
Trace-Entry Formats	39
Identification of Trace Entries	39
Branch	39
Branch in Subspace Group (if ASN Tracing on)	40
Mode Switch	40
Mode-Switching Branch	40
Program Call	40
Program Return	41

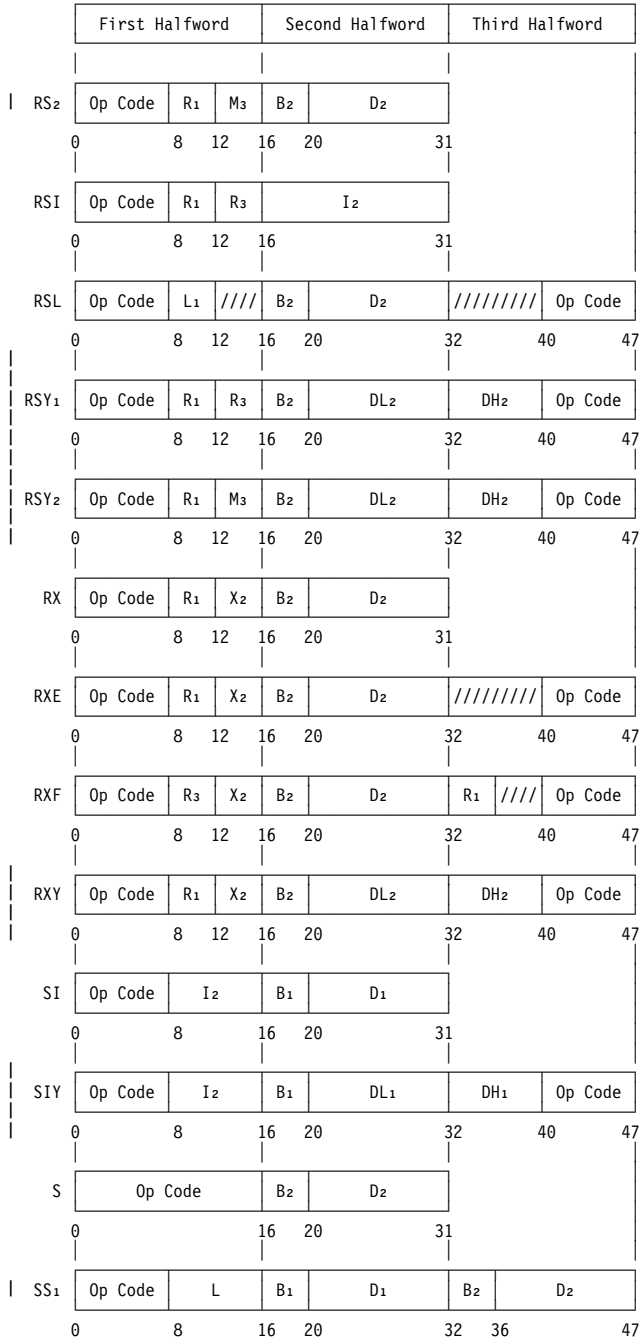
Program Transfer	42
Set Secondary ASN	42
Trace	42
Machine-Check Interruption Code	43
External-Damage Code	43
Operation-Request Block (ORB)	44
Channel-Command Word (CCW)	44
Format-0 CCW	44
Format-1 CCW	45
Indirect-Data-Address Word (IDAW)	45
Format-1 IDAW	45
Format-2 IDAW	45
Subchannel-Information Block (SCHIB)	46
Path-Management-Control Word (PMCW)	46
Interruption-Response Block (IRB)	47
Subchannel-Status Word (SCSW)	47
Extended-Status Word (ESW)	48
Information Stored in ESW	49
Extended-Control Word (ECW)	50
Extended-Measurement Word	50
Format 0 Measurement Block	50
Format 1 Measurement Block	51
Channel-Report Word (CRW)	51
Error-Recovery Codes	51
Reporting Source	51
I/O Command Codes	52
Standard Command-Code Assignments (CCW Bits 0-7)	52
Standard Meanings of Bits of First Sense Byte	52
Code Assignments	53
Code Table	53
Control Character Representations	59
Additional ISO-8 Control Character Representations	60
Formatting Character Representations	60
Two-Character BSC Data Link Controls	60
Commonly Used Editing Pattern Characters	60
ANSI-Defined Printer Control Characters	60
Hexadecimal and Decimal Conversion	60
Powers of 2 and 16	62

NOTES

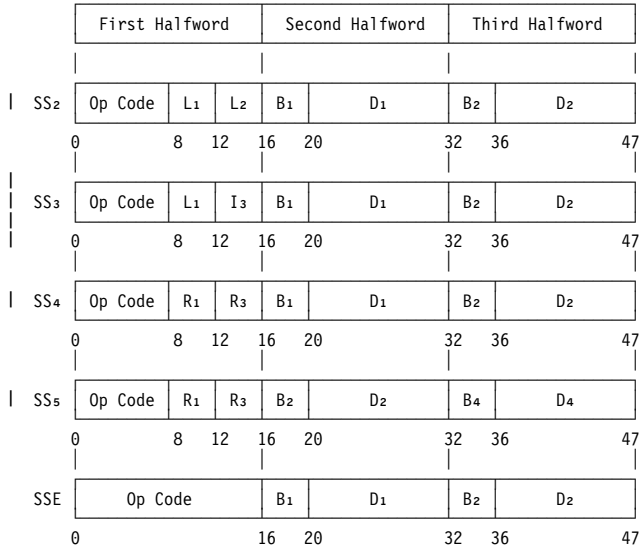
Machine Instruction Formats



Machine Instruction Formats (Cont'd)



Machine Instruction Formats (Cont'd)



- 1, 2, 3, 4: Denotes association with first, second, third, or fourth operand; distinguishes among multiple instances of the same instruction format
- B1, B2, B4: Base register designation field
- D1, D2, D4: Displacement field
- I, I2: Immediate operand field
- L, L1, L2: Length field
- M1, M3, M4: Mask field
- R1, R2, R3: Register designation field
- X2: Index register designation field

Machine Instructions by Mnemonic

Mnemonic	Operands	Name	Format	Op Code	Notes
A	R1, D2(X2, B2)	Add (32)	RX	5A	c
AD	R1, D2(X2, B2)	Add Normalized (LH)	RX	6A	c
ADB	R1, D2(X2, B2)	Add (LB)	RXE	ED1A	c
ADBR	R1, R2	Add (LB)	RRE	B31A	c
ADR	R1, R2	Add Normalized (LH)	RR	2A	c
AE	R1, D2(X2, B2)	Add Normalized (SH)	RX	7A	c
AEB	R1, D2(X2, B2)	Add (SB)	RXE	ED0A	c
AEBR	R1, R2	Add (SB)	RRE	B30A	c
AER	R1, R2	Add Normalized (SH)	RR	3A	c
AG	R1, D2(X2, B2)	Add (64)	RXY	E308	c N
AGF	R1, D2(X2, B2)	Add (64-32)	RXY	E318	c N
AGFR	R1, R2	Add (64-32)	RRE	B918	c N
AGHI	R1, I2	Add Halfword Immediate	RI1	A7B	c N
AGR	R1, R2	Add (64)	RRE	B908	c N
AH	R1, D2(X2, B2)	Add Halfword	RX	4A	c
AHI	R1, I2	Add Halfword Immediate (32)	RI1	A7A	c
AHY	R1, D2(X2, B2)	Add Halfword	RXY	E37A	c LD
AL	R1, D2(X2, B2)	Add Logical (32)	RX	5E	c
ALC	R1, D2(X2, B2)	Add Logical with Carry (32)	RXY	E398	c N3
ALCG	R1, D2(X2, B2)	Add Logical with Carry (64)	RXY	E388	c N
ALCGR	R1, R2	Add Logical with Carry (64)	RRE	B988	c N
ALCR	R1, R2	Add Logical with Carry (32)	RRE	B998	c N3
ALG	R1, D2(X2, B2)	Add Logical (64)	RXY	E30A	c N
ALGF	R1, D2(X2, B2)	Add Logical (64-32)	RXY	E31A	c N
ALGFR	R1, R2	Add Logical (64-32)	RRE	B91A	c N
ALGR	R1, R2	Add Logical (64)	RRE	B90A	c N
ALR	R1, R2	Add Logical (32)	RR	1E	c
ALY	R1, D2(X2, B2)	Add Logical (32)	RXY	E35E	c LD
AP	D1(L1, B1), D2(L2, B2)	Add Decimal	SS2	FA	c
AR	R1, R2	Add (32)	RR	1A	c
AU	R1, D2(X2, B2)	Add Unnormalized (SH)	RX	7E	c
AUR	R1, R2	Add Unnormalized (SH)	RR	3E	c
AW	R1, D2(X2, B2)	Add Unnormalized (LH)	RX	6E	c
AWR	R1, R2	Add Unnormalized (LH)	RR	2E	c
AXBR	R1, R2	Add (EB)	RRE	B34A	c
AXR	R1, R2	Add Normalized (EH)	RR	36	c
AY	R1, D2(X2, B2)	Add (32)	RXY	E35A	c LD

Machine Instructions by Mnemonic (Cont'd)

Mnemonic	Operands	Name	Format	Op Code	Notes
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 (32)	RX	46	
BCTG	R ₁ , D ₂ (X ₂ , B ₂)	Branch on Count (64)	RXY	E346	N
BCTGR	R ₁ , R ₂	Branch on Count (64)	RRE	B946	N
BCTR	R ₁ , R ₂	Branch on Count (32)	RR	06	
BRAS	R ₁ , I ₂	Branch Relative and Save	RI ₁	A75	
BRASL	R ₁ , I ₂	Branch Relative and Save Long	RIL ₁	C05	N3
BRC	M ₁ , I ₂	Branch Relative on Condition	RI ₂	A74	
BRCL	M ₁ , I ₂	Branch Relative on Condition Long	RIL ₂	C04	N3
BRCT	R ₁ , I ₂	Branch Relative on Count (32)	RI ₁	A76	
BRCTG	R ₁ , I ₂	Branch Relative on Count (64)	RI ₁	A77	N
BRXH	R ₁ , R ₃ , I ₂	Branch Relative on Index High (32)	RSI	84	
BRXHG	R ₁ , R ₃ , I ₂	Branch Relative on Index High (64)	RIE	EC44	N
BRXLE	R ₁ , R ₃ , I ₂	Branch Relative on Index Low or Equal (32)	RSI	85	
BRXLG	R ₁ , R ₃ , I ₂	Branch Relative on Index Low or Equal (64)	RIE	EC45	N
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 (32)	RS ₁	86	
BXHG	R ₁ , R ₃ , D ₂ (B ₂)	Branch on Index High (64)	RSY ₁	EB44	N
BXLE	R ₁ , R ₃ , D ₂ (B ₂)	Branch on Index Low or Equal (32)	RS ₁	87	
BXLEG	R ₁ , R ₃ , D ₂ (B ₂)	Branch on Index Low or Equal (64)	RSY ₁	EB45	N
C	R ₁ , D ₂ (X ₂ , B ₂)	Compare (32)	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	c
CDBR	R ₁ , R ₂	Compare (LB)	RRE	B319	c
CDFBR	R ₁ , R ₂	Convert from Fixed (LB<32)	RRE	B395	
CDFR	R ₁ , R ₂	Convert from Fixed (LH<32)	RRE	B3B5	
CDGBR	R ₁ , R ₂	Convert from Fixed (LB<64)	RRE	B3A5	N
CDGR	R ₁ , R ₂	Convert from Fixed (LB<64)	RRE	B3C5	N
CDR	R ₁ , R ₂	Compare (LH)	RR	29	c
CDS	R ₁ , R ₃ , D ₂ (B ₂)	Compare Double and Swap (32)	RS ₁	BB	c
CDSG	R ₁ , R ₃ , D ₂ (B ₂)	Compare Double and Swap (64)	RSY ₁	EB3E	c N
CDSY	R ₁ , R ₃ , D ₂ (B ₂)	Compare Double and Swap (32)	RSY ₁	EB31	c LD
CE	R ₁ , D ₂ (X ₂ , B ₂)	Compare (SH)	RX	79	c
CEB	R ₁ , D ₂ (X ₂ , B ₂)	Compare (SB)	RXE	ED09	c
CEBR	R ₁ , R ₂	Compare (SB)	RRE	B309	c
CEFBR	R ₁ , R ₂	Convert from Fixed (SB<32)	RRE	B394	
CEFR	R ₁ , R ₂	Convert from Fixed (SH<32)	RRE	B3B4	
CEGBR	R ₁ , R ₂	Convert from Fixed (SB<64)	RRE	B3A4	N
CEGR	R ₁ , R ₂	Convert from Fixed (SH<64)	RRE	B3C4	N
CER	R ₁ , R ₂	Compare (SH)	RR	39	c
CFC	D ₂ (B ₂)	Compare and Form Codeword	S	B21A	ic
CFDDBR	R ₁ , M ₃ , R ₂	Convert to Fixed (32<LB)	RRF ₂	B399	c
CFDR	R ₁ , M ₃ , R ₂	Convert to Fixed (32<LH)	RRF ₂	B3B9	c
CFEBR	R ₁ , M ₃ , R ₂	Convert to Fixed (32<SB)	RRF ₂	B398	c
CFER	R ₁ , M ₃ , R ₂	Convert to Fixed (32<SH)	RRF ₂	B3B8	c
CFXBR	R ₁ , M ₃ , R ₂	Convert to Fixed (32<EB)	RRF ₂	B39A	c
CFXR	R ₁ , M ₃ , R ₂	Convert to Fixed (32<EH)	RRF ₂	B3BA	c
CG	R ₁ , D ₂ (X ₂ , B ₂)	Compare (64)	RXY	E320	c N
CGDBR	R ₁ , M ₃ , R ₂	Convert to Fixed (64<LB)	RRF ₂	B3A9	c N
CGDR	R ₁ , M ₃ , R ₂	Convert to Fixed (64<LH)	RRF ₂	B3C9	c N
CGEBR	R ₁ , M ₃ , R ₂	Convert to Fixed (64<SB)	RRF ₂	B3A8	c N
CGER	R ₁ , M ₃ , R ₂	Convert to Fixed (64<SH)	RRF ₂	B3C8	c N
CGF	R ₁ , D ₂ (X ₂ , B ₂)	Compare (64<32)	RXY	E330	c N
CGFR	R ₁ , R ₂	Compare (64<32)	RRE	B930	c N
CGHI	R ₁ , I ₂	Compare Halfword Immediate (64)	RI ₁	A7F	c N
CGR	R ₁ , R ₂	Compare (64)	RRE	B920	c N
CGXBR	R ₁ , M ₃ , R ₂	Convert to Fixed (64<EB)	RRF ₂	B3AA	c N
CGXR	R ₁ , M ₃ , R ₂	Convert to Fixed (64<EH)	RRF ₂	B3CA	c N
CH	R ₁ , D ₂ (X ₂ , B ₂)	Compare Halfword	RX	49	c
CHI	R ₁ , I ₂	Compare Halfword Immediate (32)	RI ₁	A7E	c
CHY	R ₁ , D ₂ (X ₂ , B ₂)	Compare Halfword	RXY	E379	c LD
CKSM	R ₁ , R ₂	Checksum	RRE	B241	c
CL	R ₁ , D ₂ (X ₂ , B ₂)	Compare Logical (32)	RX	55	c
CLC	D ₁ (L, B ₁), D ₂ (B ₂)	Compare Logical (character)	SS ₁	D5	c
CLCL	R ₁ , R ₂	Compare Logical Long	RR	0F	ic

Machine Instructions by Mnemonic (Cont'd)

Mnemonic	Operands	Name	Format	Op Code	Notes
CLCLE	R ₁ ,R ₃ ,D ₂ (B ₂)	Compare Logical Long Extended	RS ₁	A9	c
CLCLU	R ₁ ,R ₃ ,D ₂ (B ₂)	Compare Logical Long Unicode	RSY ₁	EB8F	c E2
CLG	R ₁ ,D ₂ (X ₂ ,B ₂)	Compare Logical (64)	RXY	E321	c N
CLGF	R ₁ ,D ₂ (X ₂ ,B ₂)	Compare Logical (64<32)	RXY	E331	c N
CLGFR	R ₁ ,R ₂	Compare Logical (64<32)	RRE	B931	c N
CLGR	R ₁ ,R ₂	Compare Logical (64)	RRE	B921	c N
CLI	D ₁ (B ₁),I ₂	Compare Logical (immediate)	SI	95	c
CLII	D ₁ (B ₁),I ₂	Compare Logical (immediate)	SIY	EB55	c LD
CLM	R ₁ ,M ₃ ,D ₂ (B ₂)	Compare Logical Characters under Mask	RS ₂	BD	c
CLMH	R ₁ ,M ₃ ,D ₂ (B ₂)	Compare Logical Characters under Mask	RSY ₂	EB20	c N
CLMY	R ₁ ,M ₃ ,D ₂ (B ₂)	Compare Logical Characters under Mask	RSY ₂	EB21	c LD
CLR	R ₁ ,R ₂	Compare Logical (32)	RR	15	c
CLST	R ₁ ,R ₂	Compare Logical String	RRE	B25D	c
CLY	R ₁ ,D ₂ (X ₂ ,B ₂)	Compare Logical (32)	RXY	E355	c LD
CMPS	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 ₂	F9	c
CPYA	R ₁ ,R ₂	Copy Access	RRE	B24D	
CR	R ₁ ,R ₂	Compare (32)	RR	19	c
CS	R ₁ ,R ₃ ,D ₂ (B ₂)	Compare and Swap (32)	RS ₁	BA	c
CSCH		Clear Subchannel	S	B230	pc
CSG	R ₁ ,R ₃ ,D ₂ (B ₂)	Compare and Swap (64)	RSY ₁	EB30	c N
CSP	R ₁ ,R ₂	Compare and Swap and Purge (32)	RRE	B250	pc
CSPG	R ₁ ,R ₂	Compare and Swap and Purge (64)	RRE	B98A	pc DE
CSY	R ₁ ,R ₃ ,D ₂ (B ₂)	Compare and Swap (32)	RSY ₁	EB14	c LD
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 (32)	RX	4F	
CVBG	R ₁ ,D ₂ (X ₂ ,B ₂)	Convert to Binary (64)	RXY	E30E	N
CVBY	R ₁ ,D ₂ (X ₂ ,B ₂)	Convert to Binary (32)	RXY	EB06	LD
CVD	R ₁ ,D ₂ (X ₂ ,B ₂)	Convert to Decimal (32)	RX	4E	
CVDG	R ₁ ,D ₂ (X ₂ ,B ₂)	Convert to Decimal (64)	RXY	E32E	N
CVDY	R ₁ ,D ₂ (X ₂ ,B ₂)	Convert to Decimal (32)	RXY	E326	LD
CXBR	R ₁ ,R ₂	Compare (EB)	RRE	B349	c
CXFBR	R ₁ ,R ₂	Convert from Fixed (EB<32)	RRE	B396	
CXFR	R ₁ ,R ₂	Convert from Fixed (EH<32)	RRE	B3B6	
CXGBR	R ₁ ,R ₂	Convert from Fixed (EB<64)	RRE	B3A6	N
CXGR	R ₁ ,R ₂	Convert from Fixed (EH<64)	RRE	B3C6	N
CXR	R ₁ ,R ₂	Compare (EH)	RRE	B369	c
CY	R ₁ ,D ₂ (X ₂ ,B ₂)	Compare (32)	RXY	E359	c LD
D	R ₁ ,D ₂ (X ₂ ,B ₂)	Divide (32<64)	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	
DDBR	R ₁ ,R ₂	Divide (LB)	RRE	B31D	
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	
DEBR	R ₁ ,R ₂	Divide (SB)	RRE	B30D	
DER	R ₁ ,R ₂	Divide (SH)	RR	3D	
DIDBR	R ₁ ,R ₃ ,R ₂ ,M ₄	Divide to Integer (LB)	RRF ₃	B35B	c
DIEBR	R ₁ ,R ₃ ,R ₂ ,M ₄	Divide to Integer (SB)	RRF ₃	B353	c
DL	R ₁ ,D ₂ (X ₂ ,B ₂)	Divide Logical (32<64)	RXY	E397	N3
DLG	R ₁ ,D ₂ (X ₂ ,B ₂)	Divide Logical (64<128)	RXY	E387	N
DLGR	R ₁ ,R ₂	Divide Logical (64<128)	RRE	B987	N
DLR	R ₁ ,R ₂	Divide Logical (32<64)	RRE	B997	N3
DP	D ₁ (L ₁ ,B ₁),D ₂ (L ₂ ,B ₂)	Divide Decimal	SS ₂	FD	
DR	R ₁ ,R ₂	Divide	RR	1D	
DSG	R ₁ ,D ₂ (X ₂ ,B ₂)	Divide Single (64)	RXY	E30D	N
DSGF	R ₁ ,D ₂ (X ₂ ,B ₂)	Divide Single (64<32)	RXY	E31D	N
DSGFR	R ₁ ,R ₂	Divide Single (64<32)	RRE	B91D	N
DSGR	R ₁ ,R ₂	Divide Single (64)	RRE	B90D	N
DXBR	R ₁ ,R ₂	Divide (EB)	RRE	B34D	
DXR	R ₁ ,R ₂	Divide (EH)	RRE	B22D	
EAR	R ₁ ,R ₂	Extract Access	RRE	B24F	
ED	D ₁ (L ₁ ,B ₁),D ₂ (B ₂)	Edit	SS ₁	DE	c
EDMK	D ₁ (L ₁ ,B ₁),D ₂ (B ₂)	Edit and Mark	SS ₁	DF	c
EFPC	R ₁	Extract FPC	RRE	B38C	
EPAR	R ₁	Extract Primary ASN	RRE	B226	q
EPSW	R ₁ ,R ₂	Extract PSW	RRE	B98D	N3
EREG	R ₁ ,R ₂	Extract Stacked Registers (32)	RRE	B249	
EREGG	R ₁ ,R ₂	Extract Stacked Registers (64)	RRE	B90E	N
ESAR	R ₁	Extract Secondary ASN	RRE	B227	q
ESEA	R ₁ ,R ₂	Extract and Set Extended Authority	RRE	B99D	p N
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 ₂	B35F	
FIDR	R ₁ ,R ₂	Load FP Integer (LH)	RRE	B37F	
FIEBR	R ₁ ,M ₃ ,R ₂	Load FP Integer (SB)	RRF ₂	B357	
FIER	R ₁ ,R ₂	Load FP Integer (SH)	RRE	B377	
FIXBR	R ₁ ,M ₃ ,R ₂	Load FP Integer (EB)	RRF ₂	B347	
FIXR	R ₁ ,R ₂	Load FP Integer (EH)	RRE	B367	

Machine Instructions by Mnemonic (Cont'd)

Mnemonic	Operands	Name	Format	Op Code	Notes
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 (low)	RS ₂	BF	c
ICMH	R ₁ ,M ₃ ,D ₂ (B ₂)	Insert Characters under Mask (high)	RSY ₂	EB80	c N
ICMY	R ₁ ,M ₃ ,D ₂ (B ₂)	Insert Characters under Mask (low)	RSY ₂	EB81	c LD
ICY	R ₁ ,D ₂ (X ₂ ,B ₂)	Insert Character	RXY	E373	LD
IDTE	R ₁ ,R ₃ ,R ₂	Invalidate DAT Table Entry	RRF ₁	B98E	pu DE
IIHH	R ₁ ,I ₂	Insert Immediate (high high)	RI ₁	A50	N
IIHL	R ₁ ,I ₂	Insert Immediate (high low)	RI ₁	A51	N
IIHL	R ₁ ,I ₂	Insert Immediate (low high)	RI ₁	A52	N
IIHL	R ₁ ,I ₂	Insert Immediate (low low)	RI ₁	A53	N
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
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	c
KDBR	R ₁ ,R ₂	Compare and Signal (LB)	RRE	B318	c
KEB	R ₁ ,D ₂ (X ₂ ,B ₂)	Compare and Signal (SB)	RXE	ED08	c
KEBR	R ₁ ,R ₂	Compare and Signal (SB)	RRE	B308	c
KIMD	R ₁ ,R ₂	Compute Intermediate Message Digest	RRE	B93E	c MS
KLMD	R ₁ ,R ₂	Compute Last Message Digest	RRE	B93F	c MS
KM	R ₁ ,R ₂	Cipher Message	RRE	B92E	c MS
KMAC	R ₁ ,R ₂	Compute Message Authentication Code	RRE	B91E	c MS
KMC	R ₁ ,R ₂	Cipher Message with Chaining	RRE	B92F	c MS
KXBR	R ₁ ,R ₂	Compare and Signal (EB)	RRE	B348	c
L	R ₁ ,D ₂ (X ₂ ,B ₂)	Load (32)	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 ₁	9A	
LAMY	R ₁ ,R ₃ ,D ₂ (B ₂)	Load Access Multiple	RSY ₁	EB9A	LD
LARL	R ₁ ,I ₂	Load Address Relative Long	RIL ₁	C00	N3
LASP	D ₁ (B ₁),D ₂ (B ₂)	Load Address Space Parameters	SSE	E500	pc
LAY	R ₁ ,D ₂ (X ₂ ,B ₂)	Load Address	RXY	E371	LD
LB	R ₁ ,D ₂ (X ₂ ,B ₂)	Load Byte (32)	RXY	E376	LD
LCDBR	R ₁ ,R ₂	Load Complement (LB)	RRE	B313	c
LCDR	R ₁ ,R ₂	Load Complement (LH)	RR	23	c
LCEBR	R ₁ ,R ₂	Load Complement (SB)	RRE	B303	c
LCER	R ₁ ,R ₂	Load Complement (S)	RR	33	c
LCGFR	R ₁ ,R ₂	Load Complement (64<32)	RRE	B913	c N
LCGR	R ₁ ,R ₂	Load Complement (64)	RRE	B903	c N
LCR	R ₁ ,R ₂	Load Complement (32)	RR	13	c
LCTL	R ₁ ,R ₃ ,D ₂ (B ₂)	Load Control (32)	RS ₁	B7	p
LCTLG	R ₁ ,R ₃ ,D ₂ (B ₂)	Load Control (64)	RSY ₁	EB2F	p N
LCXBR	R ₁ ,R ₂	Load Complement (EB)	RRE	B343	c
LCXR	R ₁ ,R ₂	Load Complement (EH)	RRE	B363	c
LD	R ₁ ,D ₂ (X ₂ ,B ₂)	Load (L)	RX	68	
LDE	R ₁ ,D ₂ (X ₂ ,B ₂)	Load Lengthened (LH<SH)	RXE	ED24	
LDEB	R ₁ ,D ₂ (X ₂ ,B ₂)	Load Lengthened (LB<SB)	RXE	ED04	
LDEBR	R ₁ ,R ₂	Load Lengthened (LB<SB)	RRE	B304	
LDER	R ₁ ,R ₂	Load Lengthened (LH<SH)	RRE	B324	
LDR	R ₁ ,R ₂	Load (L)	RR	28	
LDXBR	R ₁ ,R ₂	Load Rounded (LB<EB)	RRE	B345	
LDXR	R ₁ ,R ₂	Load Rounded (LH<EH)	RR	25	
LDY	R ₁ ,D ₂ (X ₂ ,B ₂)	Load (L)	RXY	ED65	LD
LE	R ₁ ,D ₂ (X ₂ ,B ₂)	Load (S)	RX	78	
LEDBR	R ₁ ,R ₂	Load Rounded (SB<LB)	RRE	B344	
LEDR	R ₁ ,R ₂	Load Rounded (SH<LH)	RR	35	
LER	R ₁ ,R ₂	Load (S)	RR	38	
LEXBR	R ₁ ,R ₂	Load Rounded (SB<EB)	RRE	B346	
LEXR	R ₁ ,R ₂	Load Rounded (SH<EH)	RRE	B366	
LEY	R ₁ ,D ₂ (X ₂ ,B ₂)	Load (S)	RXY	ED64	LD
LFPC	D ₂ (B ₂)	Load FPC	S	B29D	
LG	R ₁ ,D ₂ (X ₂ ,B ₂)	Load (64)	RXY	E304	N
LGB	R ₁ ,D ₂ (X ₂ ,B ₂)	Load Byte (64)	RXY	E377	LD
LGF	R ₁ ,D ₂ (X ₂ ,B ₂)	Load (64<32)	RXY	E314	N
LGFR	R ₁ ,R ₂	Load (64<32)	RRE	B914	N
LGH	R ₁ ,D ₂ (X ₂ ,B ₂)	Load Halfword (64)	RXY	E315	N
LGHI	R ₁ ,I ₂	Load Halfword Immediate (64)	RI ₁	A79	N
LGR	R ₁ ,R ₂	Load (64)	RRE	B904	N
LH	R ₁ ,D ₂ (X ₂ ,B ₂)	Load Halfword (32)	RX	48	
LHI	R ₁ ,I ₂	Load Halfword Immediate (32)	RI ₁	A78	N
LHY	R ₁ ,D ₂ (X ₂ ,B ₂)	Load Halfword (32)	RXY	E378	LD
LLGC	R ₁ ,D ₂ (X ₂ ,B ₂)	Load Logical Character	RXY	E390	N
LLGF	R ₁ ,D ₂ (X ₂ ,B ₂)	Load Logical (64<32)	RXY	E316	N
LLGFR	R ₁ ,R ₂	Load Logical (64<32)	RRE	B916	N
LLGH	R ₁ ,D ₂ (X ₂ ,B ₂)	Load Logical Halfword	RXY	E391	N
LLGT	R ₁ ,D ₂ (X ₂ ,B ₂)	Load Logical Thirty One Bits	RXY	E317	N
LLGTR	R ₁ ,R ₂	Load Logical Thirty One Bits	RRE	B917	N

Machine Instructions by Mnemonic (Cont'd)

Mnemonic	Operands	Name	Format	Op Code	Notes
LLIHH	R ₁ ,I ₂	Load Logical Immediate (high high)	RI ₁	A5C	N
LLIHL	R ₁ ,I ₂	Load Logical Immediate (high low)	RI ₁	A5D	N
LLILH	R ₁ ,I ₂	Load Logical Immediate (low high)	RI ₁	A5E	N
LLILL	R ₁ ,I ₂	Load Logical Immediate (low low)	RI ₁	A5F	N
LM	R ₁ ,R ₃ ,D ₂ (B ₂)	Load Multiple (32)	RS ₁	98	
LMD	R ₁ ,R ₃ ,D ₂ (B ₂),D ₄ (B ₄)	Load Multiple Disjoint	SS ₅	EF	N
LMG	R ₁ ,R ₃ ,D ₂ (B ₂)	Load Multiple (64)	RSY ₁	EB04	N
LMH	R ₁ ,R ₃ ,D ₂ (B ₂)	Load Multiple High	RSY ₁	EB96	N
LMY	R ₁ ,R ₃ ,D ₂ (B ₂)	Load Multiple (32)	RSY ₁	EB98	LD
LNDBR	R ₁ ,R ₂	Load Negative (LB)	RRE	B311	c
LNDR	R ₁ ,R ₂	Load Negative (LH)	RR	21	c
LNEBR	R ₁ ,R ₂	Load Negative (SB)	RRE	B301	c
LNDR	R ₁ ,R ₂	Load Negative (SH)	RR	31	c
LNGFR	R ₁ ,R ₂	Load Negative (64<32)	RRE	B911	c N
LNGR	R ₁ ,R ₂	Load Negative (64)	RRE	B901	c N
LNR	R ₁ ,R ₂	Load Negative (32)	RR	11	c
LNDBR	R ₁ ,R ₂	Load Negative (EB)	RRE	B341	c
LNDR	R ₁ ,R ₂	Load Negative (EH)	RRE	B361	c
LPDBR	R ₁ ,R ₂	Load Positive (LB)	RRE	B310	c
LPDR	R ₁ ,R ₂	Load Positive (LH)	RR	20	c
LPEBR	R ₁ ,R ₂	Load Positive (SB)	RRE	B300	c
LPER	R ₁ ,R ₂	Load Positive (SH)	RR	30	c
LPGFR	R ₁ ,R ₂	Load Positive (64<32)	RRE	B910	c N
LPR	R ₁ ,R ₂	Load Positive (64)	RRE	B900	c N
LQ	R ₁ ,D ₂ (X ₂ ,B ₂)	Load Pair from Quadword	RXY	E38F	N
LPR	R ₁ ,R ₂	Load Positive (32)	RR	10	c
LPSW	D ₂ (B ₂)	Load PSW	S	82	pn
LPSWE	D ₂ (B ₂)	Load PSW Extended	S	B2B2	pn N
LPXBR	R ₁ ,R ₂	Load Positive (EB)	RRE	B340	c
LPXR	R ₁ ,R ₂	Load Positive (EH)	RRE	B360	c
LR	R ₁ ,R ₂	Load (32)	RR	18	
LRA	R ₁ ,D ₂ (X ₂ ,B ₂)	Load Real Address (32)	RX	B1	pc
LRAG	R ₁ ,D ₂ (X ₂ ,B ₂)	Load Real Address (64)	RXY	E303	pc N
LRAY	R ₁ ,D ₂ (X ₂ ,B ₂)	Load Real Address (32)	RXY	E313	pc LD
LRDR	R ₁ ,R ₂	Load Rounded (LH<EH)	RR	25	
LRER	R ₁ ,R ₂	Load Rounded (SH<LH)	RR	35	
LRV	R ₁ ,D ₂ (X ₂ ,B ₂)	Load Reversed (32)	RXY	E31E	N3
LRVG	R ₁ ,D ₂ (X ₂ ,B ₂)	Load Reversed (64)	RXY	E30F	N
LRVGR	R ₁ ,R ₂	Load Reversed (64)	RRE	B90F	N
LRVH	R ₁ ,D ₂ (X ₂ ,B ₂)	Load Reversed (16)	RXY	E31F	N3
LRVR	R ₁ ,R ₂	Load Reversed (32)	RRE	B91F	N3
LTDBR	R ₁ ,R ₂	Load and Test (LB)	RRE	B312	c
LTDR	R ₁ ,R ₂	Load and Test (LH)	RR	22	c
LTEBR	R ₁ ,R ₂	Load and Test (SB)	RRE	B302	c
LTER	R ₁ ,R ₂	Load and Test (SH)	RR	32	c
LTGFR	R ₁ ,R ₂	Load and Test (64<32)	RRE	B912	c N
LTGR	R ₁ ,R ₂	Load and Test (64)	RRE	B902	c N
LTR	R ₁ ,R ₂	Load and Test (32)	RR	12	c
LTXBR	R ₁ ,R ₂	Load and Test (EB)	RRE	B342	c
LTXR	R ₁ ,R ₂	Load and Test (EH)	RRE	B362	c
LURA	R ₁ ,R ₂	Load Using Real Address (32)	RRE	B24B	p
LURAG	R ₁ ,R ₂	Load Using Real Address (64)	RRE	B905	pn
LXD	R ₁ ,D ₂ (X ₂ ,B ₂)	Load Lengthened (EH<LH)	RXE	ED25	
LXDB	R ₁ ,D ₂ (X ₂ ,B ₂)	Load Lengthened (EB<LB)	RXE	ED05	
LXDBR	R ₁ ,R ₂	Load Lengthened (EB<LB)	RRE	B305	
LXDR	R ₁ ,R ₂	Load Lengthened (EH<LH)	RRE	B325	
LXE	R ₁ ,D ₂ (X ₂ ,B ₂)	Load Lengthened (EH<SH)	RXE	ED26	
LXEB	R ₁ ,D ₂ (X ₂ ,B ₂)	Load Lengthened (EB<SB)	RXE	ED06	
LXEBR	R ₁ ,R ₂	Load Lengthened (EB<SB)	RRE	B306	
LXER	R ₁ ,R ₂	Load Lengthened (EH<SH)	RRE	B326	
LXR	R ₁ ,R ₂	Load (E)	RRE	B365	
LY	R ₁ ,D ₂ (X ₂ ,B ₂)	Load (32)	RXY	E358	LD
LZDR	R ₁	Load Zero (L)	RRE	B375	
LZER	R ₁	Load Zero (S)	RRE	B374	
LZXR	R ₁	Load Zero (E)	RRE	B376	
M	R ₁ ,D ₂ (X ₂ ,B ₂)	Multiply (64<32)	RX	5C	
MAD	R ₁ ,R ₃ ,D ₂ (X ₂ ,B ₂)	Multiply and Add (LH)	RXF	ED3E	HM
MADB	R ₁ ,R ₃ ,D ₂ (X ₂ ,B ₂)	Multiply and Add (LB)	RXF	ED1E	
MADBR	R ₁ ,R ₃ ,R ₂	Multiply and Add (LB)	RRF ₁	B31E	
MADR	R ₁ ,R ₃ ,R ₂	Multiply and Add (LH)	RRF ₁	B33E	HM
MAE	R ₁ ,R ₃ ,D ₂ (X ₂ ,B ₂)	Multiply and Add (SH)	RXF	ED2E	HM
MAEB	R ₁ ,R ₃ ,D ₂ (X ₂ ,B ₂)	Multiply and Add (SB)	RXF	ED0E	
MAEBR	R ₁ ,R ₃ ,R ₂	Multiply and Add (SB)	RRF ₁	B30E	
MAER	R ₁ ,R ₃ ,R ₂	Multiply and Add (SH)	RRF ₁	B32E	HM
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	
MDBR	R ₁ ,R ₂	Multiply (LB)	RRE	B31C	
MDE	R ₁ ,D ₂ (X ₂ ,B ₂)	Multiply (LH<SH)	RX	7C	
MDEB	R ₁ ,D ₂ (X ₂ ,B ₂)	Multiply (LB<SB)	RXE	ED0C	
MDEBR	R ₁ ,R ₂	Multiply (LB<SB)	RRE	B30C	
MDER	R ₁ ,R ₂	Multiply (LH<SH)	RR	3C	
MDR	R ₁ ,R ₂	Multiply (LH)	RR	2C	
ME	R ₁ ,D ₂ (X ₂ ,B ₂)	Multiply (LH<SH)	RX	7C	
MEE	R ₁ ,D ₂ (X ₂ ,B ₂)	Multiply (SH)	RXE	ED37	

Machine Instructions by Mnemonic (Cont'd)

Mnemonic	Operands	Name	Format	Op Code	Notes
MEEB	R ₁ ,D ₂ (X ₂ ,B ₂)	Multiply (SB)	RXE	ED17	
MEEBR	R ₁ ,R ₂	Multiply (SB)	RRE	B317	
MEER	R ₁ ,R ₂	Multiply (SH)	RRE	B337	
MER	R ₁ ,R ₂	Multiply (LH<SH)	RR	3C	
MGHI	R ₁ ,I ₂	Multiply Halfword Immediate (64)	RI ₁	A7D	N
MH	R ₁ ,D ₂ (X ₂ ,B ₂)	Multiply Halfword (32)	RX	4C	
MHI	R ₁ ,I ₂	Multiply Halfword Immediate (32)	RI ₁	A7C	
ML	R ₁ ,D ₂ (X ₂ ,B ₂)	Multiply Logical (64<32)	RXY	E396	N3
MLG	R ₁ ,D ₂ (X ₂ ,B ₂)	Multiply Logical (128<64)	RXY	E386	N
MLGR	R ₁ ,R ₂	Multiply Logical (128<64)	RRE	B986	N
MLR	R ₁ ,R ₂	Multiply Logical (64<32)	RRE	B996	N3
MP	D ₁ (L ₁ ,B ₁),D ₂ (L ₂ ,B ₂)	Multiply Decimal	SS ₂	FC	
MR	R ₁ ,R ₂	Multiply (64<32)	RR	1C	
MS	R ₁ ,D ₂ (X ₂ ,B ₂)	Multiply Single (32)	RX	71	
MSCH	D ₂ (B ₂)	Modify Subchannel	S	B232	pc
MSD	R ₁ ,R ₃ ,D ₂ (X ₂ ,B ₂)	Multiply and Subtract (LH)	RXF	ED3F	HM
MSDB	R ₁ ,R ₃ ,D ₂ (X ₂ ,B ₂)	Multiply and Subtract (LB)	RXF	ED1F	
MSDBR	R ₁ ,R ₃ ,R ₂	Multiply and Subtract (LB)	RRF ₁	B31F	
MSDR	R ₁ ,R ₃ ,R ₂	Multiply and Subtract (LH)	RRF ₁	B33F	HM
MSE	R ₁ ,R ₃ ,D ₂ (X ₂ ,B ₂)	Multiply and Subtract (SH)	RXF	ED2F	HM
MSEB	R ₁ ,R ₃ ,D ₂ (X ₂ ,B ₂)	Multiply and Subtract (SB)	RXF	ED0F	
MSEBR	R ₁ ,R ₃ ,R ₂	Multiply and Subtract (SB)	RRF ₁	B30F	
MSEDR	R ₁ ,R ₃ ,R ₂	Multiply and Subtract (SH)	RRF ₁	B32F	HM
MSG	R ₁ ,D ₂ (X ₂ ,B ₂)	Multiply Single (64)	RXY	E30C	N
MSGF	R ₁ ,D ₂ (X ₂ ,B ₂)	Multiply Single (64<32)	RXY	E31C	N
MSGFR	R ₁ ,R ₂	Multiply Single (64<32)	RRE	B91C	N
MSGR	R ₁ ,R ₂	Multiply Single (64)	RRE	B90C	N
MSR	R ₁ ,R ₂	Multiply Single (32)	RRE	B252	
MSTA	R ₁	Modify Stacked State	RRE	B247	
MSY	R ₁ ,D ₂ (X ₂ ,B ₂)	Multiply Single (32)	RXY	E351	LD
MVC	D ₁ (L,B ₁),D ₂ (B ₂)	Move (character)	SS ₁	D2	
MVCCK	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 ₁	E8	
MVCK	D ₁ (R ₁ ,B ₁),D ₂ (B ₂),R ₃	Move with Key	SS ₄	D9	qc
MVCL	R ₁ ,R ₂	Move Long	RR	0E	ic
MVCLC	R ₁ ,R ₃ ,D ₂ (B ₂)	Move Long Extended	RS ₁	A8	c
MVCLU	R ₁ ,R ₃ ,D ₂ (B ₂)	Move Long Unicode	RSY ₁	EB8E	c E2
MVCP	D ₁ (R ₁ ,B ₁),D ₂ (B ₂),R ₃	Move to Primary	SS ₄	DA	qc
MVCS	D ₁ (R ₁ ,B ₁),D ₂ (B ₂),R ₃	Move to Secondary	SS ₄	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 (immediate)	SI	92	
MVIY	D ₁ (B ₁),I ₂	Move (immediate)	SIY	EB52	LD
MVN	D ₁ (L,B ₁),D ₂ (B ₂)	Move Numerics	SS ₁	D1	
MVO	D ₁ (L ₁ ,B ₁),D ₂ (L ₂ ,B ₂)	Move with Offset	SS ₂	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 ₁	D3	
MXBR	R ₁ ,R ₂	Multiply (EB)	RRE	B34C	
MXD	R ₁ ,D ₂ (X ₂ ,B ₂)	Multiply (EH<LH)	RX	67	
MXDB	R ₁ ,D ₂ (X ₂ ,B ₂)	Multiply (EB<LB)	RXE	ED07	
MXDBR	R ₁ ,R ₂	Multiply (EB<LB)	RRE	B307	
MXDR	R ₁ ,R ₂	Multiply (EH<LH)	RR	27	
MXR	R ₁ ,R ₂	Multiply (EH)	RR	26	
N	R ₁ ,D ₂ (X ₂ ,B ₂)	And (32)	RX	54	c
NC	D ₁ (L,B ₁),D ₂ (B ₂)	And (character)	SS ₁	D4	c
NG	R ₁ ,D ₂ (X ₂ ,B ₂)	And (64)	RXY	E380	c N
NGR	R ₁ ,R ₂	And (64)	RRE	B980	c N
NI	D ₁ (B ₁),I ₂	And (immediate)	SI	94	c
NIHH	R ₁ ,I ₂	And Immediate (high high)	RI ₁	A54	c N
NIHL	R ₁ ,I ₂	And Immediate (high low)	RI ₁	A55	c N
NILH	R ₁ ,I ₂	And Immediate (low high)	RI ₁	A56	c N
NILL	R ₁ ,I ₂	And Immediate (low low)	RI ₁	A57	c N
NIY	D ₁ (B ₁),I ₂	And (immediate)	SIY	EB54	c LD
NR	R ₁ ,R ₂	And (32)	RR	14	c
NY	R ₁ ,D ₂ (X ₂ ,B ₂)	And (32)	RXY	E354	c LD
O	R ₁ ,D ₂ (X ₂ ,B ₂)	Or (32)	RX	56	c
OC	D ₁ (L,B ₁),D ₂ (B ₂)	Or (character)	SS ₁	D6	c
OG	R ₁ ,D ₂ (X ₂ ,B ₂)	Or (64)	RXY	E381	c N
OGR	R ₁ ,R ₂	Or (64)	RRE	B981	c N
OI	D ₁ (B ₁),I ₂	Or (immediate)	SI	96	c
OIHH	R ₁ ,I ₂	Or Immediate (high high)	RI ₁	A58	c N
OIHL	R ₁ ,I ₂	Or Immediate (high low)	RI ₁	A59	c N
OILH	R ₁ ,I ₂	Or Immediate (low high)	RI ₁	A5A	c N
OILL	R ₁ ,I ₂	Or Immediate (low low)	RI ₁	A5B	c N
OIY	D ₁ (B ₁),I ₂	Or (immediate)	SIY	EB56	c LD
OR	R ₁ ,R ₂	Or (32)	RR	16	c
OY	R ₁ ,D ₂ (X ₂ ,B ₂)	Or (32)	RXY	E356	c LD
PACK	D ₁ (L ₁ ,B ₁),D ₂ (L ₂ ,B ₂)	Pack	SS ₂	F2	
PALB		Purge ALB	RRE	B248	p
PC	D ₂ (B ₂)	Program Call	S	B218	q
PGIN	R ₁ ,R ₂	Page In	RRE	B22E	pc ES
PGOUT	R ₁ ,R ₂	Page Out	RRE	B22F	pc ES
PKA	D ₁ (B ₁),D ₂ (L ₂ ,B ₂)	Pack ASCII	SS ₁	E9	E2
PKU	D ₁ (B ₁),D ₂ (L ₂ ,B ₂)	Pack Unicode	SS ₁	E1	E2
PLO	R ₁ ,D ₂ (B ₂),R ₃ ,D ₄ (B ₄)	Perform Locked Operation	SS ₅	EE	c
PR		Program Return	E	0101	qn
PT	R ₁ ,R ₂	Program Transfer	RRE	B228	q

Machine Instructions by Mnemonic (Cont'd)

Mnemonic	Operands	Name	Format	Op Code	Notes
PTLB		Purge TLB	S	B20D	p
RCHP		Reset Channel Path	S	B23B	pc
RLL	R ₁ , R ₃ , D ₂ (B ₂)	Rotate Left Single Logical (32)	RSY ₁	EB1D	N3
RLLG	R ₁ , R ₃ , D ₂ (B ₂)	Rotate Left Single Logical (64)	RSY ₁	EB1C	N
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 (32)	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
SAM24		Set Addressing Mode (24)	E	010C	N3
SAM31		Set Addressing Mode (31)	E	010D	N3
SAM64		Set Addressing Mode (64)	E	010E	N
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	c
SDBR	R ₁ , R ₂	Subtract (LB)	RRE	B31B	c
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	c
SEBR	R ₁ , R ₂	Subtract (SB)	RRE	B30B	c
SER	R ₁ , R ₂	Subtract Normalized (SH)	RR	3B	c
SFPC	R ₁	Set FPC	RRE	B384	
SG	R ₁ , D ₂ (X ₂ , B ₂)	Subtract (64)	RXY	E309	c N
SGF	R ₁ , D ₂ (X ₂ , B ₂)	Subtract (64<32)	RXY	E319	c N
SGFR	R ₁ , R ₂	Subtract (64<32)	RRE	B919	c N
SGR	R ₁ , R ₂	Subtract (64)	RRE	B909	c N
SH	R ₁ , D ₂ (X ₂ , B ₂)	Subtract Halfword	RX	4B	c
SHY	R ₁ , D ₂ (X ₂ , B ₂)	Subtract Halfword	RXY	E37B	c LD
SIE	D ₂ (B ₂)	Start Interpretive Execution	S	B214	ip
SIGP	R ₁ , R ₃ , D ₂ (B ₂)	Signal Processor	RS ₁	AE	pc
SL	R ₁ , D ₂ (X ₂ , B ₂)	Subtract Logical (32)	RX	5F	c
SLA	R ₁ , D ₂ (B ₂)	Shift Left Single (32)	RS ₁	8B	c
SLAG	R ₁ , R ₃ , D ₂ (B ₂)	Shift Left Single (64)	RSY ₁	EB0B	c N
SLB	R ₁ , D ₂ (X ₂ , B ₂)	Subtract Logical with Borrow (32)	RXY	E399	c N3
SLBG	R ₁ , D ₂ (X ₂ , B ₂)	Subtract Logical with Borrow (64)	RXY	E389	c N
SLBGR	R ₁ , R ₂	Subtract Logical with Borrow (64)	RRE	B989	c N
SLBR	R ₁ , R ₂	Subtract Logical with Borrow (32)	RRE	B999	c N3
SLDA	R ₁ , D ₂ (B ₂)	Shift Left Double	RS ₁	8F	c
SLDL	R ₁ , D ₂ (B ₂)	Shift Left Double Logical	RS ₁	8D	
SLG	R ₁ , D ₂ (X ₂ , B ₂)	Subtract Logical (64)	RXY	E30B	c N
SLGF	R ₁ , D ₂ (X ₂ , B ₂)	Subtract Logical (64<32)	RXY	E31B	c N
SLGFR	R ₁ , R ₂	Subtract Logical (64<32)	RRE	B91B	c N
SLGR	R ₁ , R ₂	Subtract Logical (64)	RRE	B90B	c N
SLL	R ₁ , D ₂ (B ₂)	Shift Left Single Logical (32)	RS ₁	89	
SLLG	R ₁ , R ₃ , D ₂ (B ₂)	Shift Left Single Logical (64)	RSY ₁	EB0D	N
SLR	R ₁ , R ₂	Subtract Logical (32)	RR	1F	c
SLY	R ₁ , D ₂ (X ₂ , B ₂)	Subtract Logical (32)	RXY	E35F	c LD
SP	D ₁ (L ₁ , B ₁), D ₂ (L ₂ , B ₂)	Subtract Decimal	SS ₂	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	
SQDB	R ₁ , D ₂ (X ₂ , B ₂)	Square Root (LB)	RXE	ED15	
SQDBR	R ₁ , R ₂	Square Root (LB)	RRE	B315	
SQDR	R ₁ , R ₂	Square Root (LH)	RRE	B244	
SQE	R ₁ , D ₂ (X ₂ , B ₂)	Square Root (SH)	RXE	ED34	
SQEB	R ₁ , D ₂ (X ₂ , B ₂)	Square Root (SB)	RXE	ED14	
SQEBR	R ₁ , R ₂	Square Root (SB)	RRE	B314	
SQER	R ₁ , R ₂	Square Root (SH)	RRE	B245	
SQXBR	R ₁ , R ₂	Square Root (EB)	RRE	B316	
SQXR	R ₁ , R ₂	Square Root (EH)	RRE	B336	
SR	R ₁ , R ₂	Subtract (32)	RR	1B	c
SRA	R ₁ , D ₂ (B ₂)	Shift Right Single (32)	RS ₁	8A	c
SRAG	R ₁ , R ₃ , D ₂ (B ₂)	Shift Right Single (64)	RSY ₁	EB0A	c N
SRDA	R ₁ , D ₂ (B ₂)	Shift Right Double	RS ₁	8E	c
SRDL	R ₁ , D ₂ (B ₂)	Shift Right Double Logical	RS ₁	8C	
SRL	R ₁ , D ₂ (B ₂)	Shift Right Single Logical (32)	RS ₁	88	
SRLG	R ₁ , R ₃ , D ₂ (B ₂)	Shift Right Single Logical (64)	RSY ₁	EB0C	N
SRNM	D ₂ (B ₂)	Set Rounding Mode	S	B299	
SRP	D ₁ (L ₁ , B ₁), D ₂ (B ₂), I ₃	Shift and Round Decimal	SS ₃	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 (32)	RX	50	
STAM	R ₁ , R ₃ , D ₂ (B ₂)	Store Access Multiple	RS ₁	9B	

Machine Instructions by Mnemonic (Cont'd)

Mnemonic	Operands	Name	Format	Op Code	Notes
STAMY	R ₁ , R ₃ , D ₂ (B ₂)	Store Access Multiple	RSY ₁	EB9B	LD
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 (low)	RS ₂	BE	
STCMH	R ₁ , M ₃ , D ₂ (B ₂)	Store Characters under Mask (high)	RSY ₁	EB2C	N
STCMY	R ₁ , M ₃ , D ₂ (B ₂)	Store Characters under Mask (low)	RSY ₂	EB2D	LD
STCPS	D ₂ (B ₂)	Store Channel Path Status	S	B23A	p
STCRW	D ₂ (B ₂)	Store Channel Report Word	S	B239	pc
STCTG	R ₁ , R ₃ , D ₂ (B ₂)	Store Control (64)	RSY ₁	EB25	p N
STCTL	R ₁ , R ₃ , D ₂ (B ₂)	Store Control (32)	RS ₁	B6	p
STCY	R ₁ , D ₂ (X ₂ , B ₂)	Store Character	RXY	E372	LD
STD	R ₁ , D ₂ (X ₂ , B ₂)	Store (L)	RX	60	
STDY	R ₁ , D ₂ (X ₂ , B ₂)	Store (L)	RXY	ED67	LD
STE	R ₁ , D ₂ (X ₂ , B ₂)	Store (S)	RX	70	
STEY	R ₁ , D ₂ (X ₂ , B ₂)	Store (S)	RXY	ED66	LD
STFL	D ₂ (B ₂)	Store Facility List	S	B2B1	p N3
STFPC	D ₂ (B ₂)	Store FPC	S	B29C	
STG	R ₁ , D ₂ (X ₂ , B ₂)	Store (64)	RXY	E324	N
STH	R ₁ , D ₂ (X ₂ , B ₂)	Store Halfword	RX	40	
STHY	R ₁ , D ₂ (X ₂ , B ₂)	Store Halfword	RXY	E370	LD
STIDP	D ₂ (B ₂)	Store CPU ID	S	B202	p
STM	R ₁ , R ₃ , D ₂ (B ₂)	Store Multiple (32)	RS ₁	90	
STMG	R ₁ , R ₃ , D ₂ (B ₂)	Store Multiple (64)	RSY ₁	EB24	N
STMH	R ₁ , R ₃ , D ₂ (B ₂)	Store Multiple High	RSY ₁	EB26	N
STMY	R ₁ , R ₃ , D ₂ (B ₂)	Store Multiple (32)	RSY ₁	EB90	LD
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
STPQ	R ₁ , D ₂ (X ₂ , B ₂)	Store Pair to Quadword	RXY	E38E	N
STPT	D ₂ (B ₂)	Store CPU Timer	S	B209	p
STPX	D ₂ (B ₂)	Store Prefix	S	B211	p
STRAG	D ₁ (B ₁), D ₂ (B ₂)	Store Real Address	SSE	E502	p N
STRV	R ₁ , D ₂ (X ₂ , B ₂)	Store Reversed (32)	RXY	E33E	N3
STRVG	R ₁ , D ₂ (X ₂ , B ₂)	Store Reversed (64)	RXY	E32F	N
STRVH	R ₁ , D ₂ (X ₂ , B ₂)	Store Reversed (16)	RXY	E33F	N3
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 (32)	RRE	B246	p
STURG	R ₁ , R ₂	Store Using Real Address (64)	RRE	B925	p N
STY	R ₁ , D ₂ (X ₂ , B ₂)	Store (32)	RXY	E350	LD
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	I	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	c
SXR	R ₁ , D ₂	Subtract Normalized (EH)	RR	37	c
SY	R ₁ , D ₂ (X ₂ , B ₂)	Subtract (32)	RXY	E35B	c LD
TAM		Test Addressing Mode	E	010B	c N3
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 (LB<LH)	RRF ₂	B351	c
TBEDR	R ₁ , M ₃ , R ₂	Convert HFP to BFP (SB<LH)	RRF ₂	B350	c
TCDB	R ₁ , D ₂ (X ₂ , B ₂)	Test Data Class (LB)	RXE	ED11	c
TCEB	R ₁ , D ₂ (X ₂ , B ₂)	Test Data Class (SB)	RXE	ED10	c
TCXB	R ₁ , D ₂ (X ₂ , B ₂)	Test Data Class (EB)	RXE	ED12	c
THDER	R ₁ , R ₂	Convert BFP to HFP (LH<SB)	RRE	B358	c
THDR	R ₁ , R ₂	Convert BFP to HFP (LH<LB)	RRE	B359	c
TM	D ₁ (B ₁), I ₂	Test under Mask	SI	91	c
TMH	R ₁ , I ₂	Test under Mask High	RI ₁	A70	c
TMHH	R ₁ , I ₂	Test under Mask (high high)	RI ₁	A72	c N
TMHL	R ₁ , I ₂	Test under Mask (high low)	RI ₁	A73	c N
TML	R ₁ , I ₂	Test under Mask Low	RI ₁	A71	c
TMLH	R ₁ , I ₂	Test under Mask (low high)	RI ₁	A70	c N
TMLL	R ₁ , I ₂	Test under Mask (low low)	RI ₁	A71	c N
TMY	D ₁ (B ₁), I ₂	Test under Mask	SIY	EB51	c LD
TP	D ₁ (L ₁ , B ₁)	Test Decimal	RSL	EB0C	c E2
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 ₁ , B ₁), D ₂ (B ₂)	Translate	SS ₁	DC	
TRACE	R ₁ , R ₃ , D ₂ (B ₂)	Trace (32)	RS ₁	99	p
TRACG	R ₁ , R ₃ , D ₂ (B ₂)	Trace (64)	RSY ₁	EB0F	p N
TRAP2		Trap	E	01FF	
TRAP4	D ₂ (B ₂)	Trap	S	B2FF	
TRE	R ₁ , R ₂	Translate Extended	RRE	B2A5	c
TROO	R ₁ , R ₂	Translate One to One	RRE	B993	c E2
TROT	R ₁ , R ₂	Translate One to Two	RRE	B992	c E2
TRT	D ₁ (L ₁ , B ₁), D ₂ (B ₂)	Translate and Test	SS ₁	DD	c
TRTO	R ₁ , R ₂	Translate Two to One	RRE	B991	c E2
TRTT	R ₁ , R ₂	Translate Two to Two	RRE	B990	c E2
TS	D ₂ (B ₂)	Test and Set	S	93	c
TSCB	D ₂ (B ₂)	Test Subchannel	S	B235	pc
UNPK	D ₁ (L ₁ , B ₁), D ₂ (L ₂ , B ₂)	Unpack	SS ₂	F3	
UNPKA	D ₁ (L ₁ , B ₁), D ₂ (B ₂)	Unpack ASCII	SS ₁	EA	c E2

Machine Instructions by Operation Code (Cont'd)

Mnemonic	Operands	Name	Format	Op Code	Notes
UNPKU	D ₁ (L ₁ ,B ₁),D ₂ (B ₂)	Unpack Unicode	SS ₁	E2	c E2
UPT		Update Tree	E	0102	ic
X	R ₁ ,D ₂ (X ₂ ,B ₂)	Exclusive Or (32)	RX	57	c
XC	D ₁ (L ₁ ,B ₁),D ₂ (B ₂)	Exclusive Or (character)	SS ₁	D7	c
XG	R ₁ ,D ₂ (X ₂ ,B ₂)	Exclusive Or (64)	RXY	E382	c N
XGR	R ₁ ,R ₂	Exclusive Or (64)	RRE	B982	c N
XI	D ₁ (B ₁),I ₂	Exclusive Or (immediate)	SI	97	c
XIY	D ₁ (B ₁),I ₂	Exclusive Or (immediate)	SIY	EB57	c LD
XR	R ₁ ,R ₂	Exclusive Or (32)	RR	17	c
XSCH		Cancel Subchannel	S	B276	pc
XY	R ₁ ,D ₂ (X ₂ ,B ₂)	Exclusive Or (32)	RXY	E357	c LD
ZAP	D ₁ (L ₁ ,B ₁),D ₂ (L ₂ ,B ₂)	Zero and Add	SS ₂	F8	c
---	Model-dependent	Diagnose	--	83	pu

Floating-Point Operand Lengths and Types:

		Notes:
E	Extended (binary or hex)	c Condition code set
EB	Extended binary	i Interruptible instruction
EH	Extended hex	n New condition code loaded
L	Long (binary or hex)	p Privileged instruction
LB	Long binary	q Semiprivileged instruction
LH	Long hex	u Condition code is unpredictable
S	Short (binary or hex)	DE DAT-enhancement facility
SB	Short binary	E2 Extended-translation facility 2
SH	Short hex	ES Expanded-storage facility
		HM Hexadecimal multiply-and-add/subtract facility
		LD Long displacement facility
		MS Message-security assist facility
		N Instruction new in z/Architecture compared to ESA/390
		N3 Instruction new in z/Architecture compared to ESA/390 and added to ESA/390

Machine Instructions by Operation Code

Op Code	Mnemonic	Op Code	Mnemonic	Op Code	Mnemonic
0101	PR	2F	SWR	67	MXD
0102	UPT	30	LPER	68	LD
0107	SCKPF	31	LNER	69	CD
010B	TAM	32	LTER	6A	AD
010C	SAM24	33	LCER	6B	SD
010D	SAM31	34	HER	6C	MD
010E	SAM64	35	LEDR	6D	DD
01FF	TRAP2	35	LRER	6E	AW
04	SPM	36	AXR	6F	SW
05	BALR	37	SXR	70	STE
06	BCTR	38	LER	71	MS
07	BCR	39	CER	78	LE
0A	SVC	3A	AER	79	CE
0B	BSM	3B	SER	7A	AE
0C	BASSM	3C	MDER	7B	SE
0D	BASR	3C	MER	7C	MDE
0E	MVCL	3D	DER	7C	ME
0F	CLCL	3E	AUR	7D	DE
10	LPR	3F	SUR	7E	AU
11	LNR	40	STH	7F	SU
12	LTR	41	LA	80	SSM
13	LCR	42	STC	82	LPSW
14	NR	43	IC	83	Diagnose
15	CLR	44	EX	84	BRXH
16	OR	45	BAL	85	BRXLE
17	XR	46	BCT	86	BXH
18	LR	47	BC	87	BXLE
19	CR	48	LH	88	SRL
1A	AR	49	CH	89	SLL
1B	SR	4A	AH	8A	SRA
1C	MR	4B	SH	8B	SLA
1D	DR	4C	MH	8C	SRDL
1E	ALR	4D	BAS	8D	SLDL
1F	SLR	4E	CVD	8E	SRDA
20	LPDR	4F	CVB	8F	SLDA
21	LNDR	50	ST	90	STM
22	LTDR	51	LAE	91	TM
23	LCDR	54	N	92	MVI
24	HDR	55	CL	93	TS
25	LDXR	56	O	94	NI
25	LRDR	57	X	95	CLI
26	MXR	58	L	96	OI
27	MXDR	59	C	97	XI
28	LDR	5A	A	98	LM
29	CDR	5B	S	99	TRACE
2A	ADR	5C	M	9A	LAM
2B	SDR	5D	D	9B	STAM
2C	MDR	5E	AL	A50	IIHH
2D	DDR	5F	SL	A51	IIHL
2E	AWR	60	STD	A52	ILH

Machine Instructions by Operation Code (Cont'd)

Op Code	Mnemonic	Op Code	Mnemonic	Op Code	Mnemonic
A53	ILL	B276	XSCH	B3B8	CFER
A54	NIHH	B277	RP	B3B9	CFDR
A55	NIHL	B278	STCKE	B3BA	CFXR
A56	NILH	B279	SACF	B3CA	CEGR
A57	NILL	B27D	STSI	B3C5	CDGR
A58	OIHH	B299	SRNM	B3C6	CXGR
A59	OIHL	B29C	STFPC	B3C8	CGER
A5A	OILL	B29D	LFPC	B3C9	CGDR
A5B	OILL	B2A5	TRE	B3CA	CGXR
A5C	LLIHH	B2A6	CUUTF	B6	STCTL
A5D	LLIHL	B2A7	CUTFU	B7	LCTL
A5E	LLILH	B2B1	STFL	B900	LPGR
A5F	LLILL	B2B2	LPSWE	B901	LNGR
A70	TMLH	B2FF	TRAP4	B902	LTGR
A70	TMH	B300	LPFBR	B903	LCGR
A71	TMLL	B301	LNBR	B904	B904
A71	TML	B302	LTEBR	B905	LURAG
A72	TMHH	B303	LCEBR	B908	AGR
A73	TMHL	B304	LDEBR	B909	SGR
A74	BRC	B305	LXDBR	B90A	ALGR
A75	BRAS	B306	LXEBR	B90B	SLGR
A76	BRCT	B307	MXDBR	B90C	MSGR
A77	BRCTG	B308	KEBR	B90D	DSGR
A78	LHI	B309	CEBR	B90E	EREGG
A79	LGHI	B30A	AEBR	B90F	LRVGR
A7A	AHI	B30B	SEBR	B910	LPGFR
A7B	AGHI	B30C	MDEBR	B911	LNFR
A7C	MHI	B30D	DEBR	B912	LTGFR
A7D	MGHI	B30E	MAEBR	B913	LCGFR
A7E	CHI	B30F	MSEBR	B914	LGFR
A7F	CGHI	B310	LPDBR	B916	LLGFR
A8	MVCLE	B311	LNDBR	B917	LLGTR
A9	CLCLE	B312	LTDBR	B918	AGFR
AC	STNSM	B313	LCDBR	B919	SGFR
AD	STOSM	B314	SOEBR	B91A	ALGFR
AE	SIGP	B315	SQDBR	B91B	SLGFR
AF	MC	B316	SQXBR	B91C	MSGFR
B1	LRA	B317	MEEBR	B91D	DSGFR
B202	STIDP	B318	KDBR	B91E	KMAC
B204	SCKA	B319	QDBR	B91F	RVFR
B205	STCK	B31A	ADBR	B920	CGR
B206	SCKC	B31B	SDBR	B921	CLGR
B207	STCKC	B31C	MDBR	B925	STURG
B208	SPT	B31D	DBBR	B92E	KM
B209	STPT	B31E	MADBR	B92F	KMC
B20A	SPKA	B31F	MSDBR	B930	CGFR
B20B	IPK	B324	LDER	B931	CLGFR
B20D	PTLB	B325	LXDR	B93E	KIMD
B210	SPX	B326	LXER	B93F	KLMD
B211	STPX	B32E	MAER	B946	BCTGR
B212	STAP	B32F	MSEBR	B980	NGR
B214	SIE	B336	SQXBR	B981	NGR
B218	PC	B337	MEER	B982	XGR
B219	SAC	B33E	MADR	B986	MLGR
B21A	CFC	B33F	MSDR	B987	DLGR
B221	IPTE	B340	LPXBR	B988	ALCGR
B222	IPM	B341	LNGBR	B989	SLBGR
B223	IVSK	B342	LTXBR	B98A	CSFG
B224	IAC	B343	LCXBR	B98D	EPSW
B225	SSAR	B344	LEDBR	B98E	IDTE
B226	EPAR	B345	LDXBR	B990	TRTT
B227	ESAR	B346	LEXBR	B991	TRTO
B228	PT	B347	FIXBR	B992	TRPT
B229	ISKE	B348	KXBR	B993	TRTO
B22A	RRBE	B349	CXBR	B996	MLR
B22B	SSKE	B34A	AXBR	B997	DLR
B22C	TB	B34B	SXBR	B998	ALCR
B22D	DXR	B34C	MXBR	B999	SLBR
B22E	PGIN	B34D	DXBR	B99D	ESBEA
B22F	PGOUT	B350	TBEDR	BA	CS
B230	CSCH	B351	TBDR	BB	CDS
B231	HSCH	B353	DIEBR	BD	CLM
B232	MSCH	B357	FIEBR	BE	STCM
B233	SSCH	B358	THDER	BF	ICM
B234	STSCH	B359	THDR	C00	LARL
B235	TSCH	B35B	DIDBR	C04	BRCL
B236	TPI	B35F	FIDBR	C05	BRASL
B237	SAL	B360	LPXR	D1	MVN
B238	RSCH	B361	LNXR	D2	MVC
B239	STCRW	B362	LTXR	D3	MVZ
B23A	STOPS	B363	LCXR	D4	NC
B23B	RCHP	B365	LXR	D5	CLC
B23C	SCHM	B366	LEXR	D6	OC
B240	BAKR	B367	FIXR	D7	XC
B241	CKSM	B369	CXR	D9	MVCK
B244	SQDR	B374	LZER	DA	MVCP
B245	SQER	B375	LZDR	DB	MVCS
B246	STURA	B376	LZXR	DC	TR
B247	MSTA	B377	FIER	DD	TRT
B248	PALB	B37F	FIDR	DE	ED
B249	EREG	B384	SFPC	DF	EDMK
B24A	ESTA	B38C	EPFC	E1	PKU
B24B	LJRA	B394	QFBR	E2	UNPKU
B24C	TAR	B395	QDFBR	E303	LRAQ
B24D	CPYA	B396	CXFBR	E304	LG
B24E	SAR	B398	CFEBR	E306	CVBY
B24F	EAR	B399	CFDBR	E308	AG
B250	CSP	B39A	CFXBR	E309	SG
B252	MSR	B3A4	CFGBR	E30A	ALG
B254	MVPG	B3A5	CFGBR	E30B	SLG
B255	MVST	B3A6	CXGBR	E30C	MSG
B257	CUSE	B3A8	CGEBR	E30D	DSG
B258	BSG	B3A9	CGDBR	E30E	CVBG
B25A	BSA	B3AA	CGXBR	E30F	LRVG
B25D	CLST	B3B4	CFR	E313	LRAV
B25E	SRST	B3B5	CFR	E314	TR
B263	CMPSC	B3B6	CXFR	E315	LGH

Machine Instructions by Operation Code (Cont'd)

Op Code	Mne-monic	Op Code	Mne-monic
E316	LLGF	ED05	LXDB
E317	LLGT	ED06	LXEB
E318	AGF	ED07	MXDB
E319	SGF	ED08	KEB
E31A	ALGF	ED09	CEB
E31B	SLGF	ED0A	AEB
E31C	MSGF	ED0B	SEB
E31D	DSGF	ED0C	MDEB
E31E	LRV	ED0D	DEB
E31F	LRVH	ED0E	MAEB
E320	CG	ED0F	MSEB
E321	CLG	ED10	TCEB
E324	STG	ED11	TCDB
E326	CVDY	ED12	TCXB
E32E	CVDG	ED14	SOEB
E32F	STRVG	ED15	SODB
E330	CGF	ED17	MEEB
E331	CLGF	ED18	KDB
E33E	STRV	ED19	CDB
E33F	STRVH	ED1A	ADB
E346	BCTG	ED1B	SDB
E350	STY	ED1C	MDB
E351	MSY	ED1D	DDB
E354	NY	ED1E	MADB
E355	CLY	ED1F	MSDB
E356	OY	ED24	LDE
E357	XY	ED25	LXD
E358	LY	ED26	LXE
E359	CY	ED2E	MAE
E35A	AY	ED2F	MSE
E35B	SY	ED34	SOE
E35E	ALY	ED35	SQD
E35F	SLY	ED37	MEE
E370	STHY	ED3E	MAD
E371	LAY	ED3F	MSD
E372	STCY	ED64	LEY
E373	ICY	ED65	LDY
E376	LB	ED66	STEY
E377	LGB	ED67	STDY
E378	LHY	EE	PLO
E379	CHY	EF	LMD
E37A	AHY	F0	SRP
E37B	SHY	F1	MVO
E380	NG	F2	PACK
E381	OG	F3	UNPK
E382	XG	F8	ZAP
E386	MLG	F9	CP
E387	DLG	FA	AP
E388	ALCG	FB	SP
E389	SLBG	FC	MP
E38E	STPQ	FD	DP
E38F	LPO		
E390	LLGC		
E391	LLGH		
E396	ML		
E397	DL		
E398	ALC		
E399	SLB		
E500	LASP		
E501	TPROT		
E502	STRAG		
E50E	MVCSK		
E50F	MVCDK		
E8	MVCIN		
E9	PKA		
EA	UNPKA		
EB04	LMG		
EB0A	SRAG		
EB0B	SLAG		
EB0C	SRLG		
EB0D	SLLG		
EB0F	TRACG		
EB14	CSY		
EB1C	RLLG		
EB1D	RLL		
EB20	CLMH		
EB21	CLMY		
EB24	STMG		
EB25	STCTG		
EB26	STMH		
EB2C	STOMH		
EB2D	STCMY		
EB2F	LCTLG		
EB30	CSG		
EB31	CDSY		
EB3E	CDSG		
EB44	BXHG		
EB45	BXLEG		
EB51	TMY		
EB52	MVIY		
EB54	NIY		
EB55	CLY		
EB56	OY		
EB57	XIY		
EB80	ICMH		
EB81	ICMY		
EB8E	MVCLU		
EB8F	CLCLU		
EB90	STMY		
EB96	LMH		
EB98	LMY		
EB9A	LAMY		
EB9B	STAMY		
EBCO	TP		
EC44	BRXHG		
EC45	BRXLG		
ED04	LDEB		

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
Add Logical with Carry	Zero, no carry	Not zero, no carry	Zero, carry	Not zero, carry
And	Zero	Not zero	----	----
And Immediate	Sixteen bits zero	Sixteen bits not zero	----	----
Checksum	Checksum complete	----	----	CPU-determined completion
Cipher Message	Normal completion	----	----	Partial completion
Cipher Message with Chaining	Normal completion	----	----	Partial 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 Long Unicode	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 Sub-string 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
Compute Intermed. Message Digest	Normal completion	----	----	Partial completion
Compute Last Message Digest	Normal completion	----	----	Partial completion
Compute Message Authen. Code	Normal completion	----	----	Partial 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	----	----

Condition Codes (Cont'd)

Condition Code →	0	1	2	3
Mask Bit Value →	8	4	2	1
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 Long Unicode	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	----	----
Or Immediate	Sixteen bits zero	Sixteen bits 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
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 operational
Store Clock	Set state	Not-set state	Error state	Stopped state or not operational
Subtract	Zero	< Zero	> Zero	Overflow
Subtract Halfword	Zero	< Zero	> Zero	Overflow
Subtract Logical	----	Not zero, borrow	Zero, no borrow	Not zero, no borrow
Subtract Logical with Borrow	Zero, borrow	Not zero, borrow	Zero, no borrow	Not zero, no borrow
Test Addressing Mode	24-bit mode	31-bit mode	----	----
Test and Set	Leftmost bit zero	Leftmost bit one	----	----
Test under Mask (TM)	All zeros, or mask is zero	Mixed 0's and 1's	---	All ones
Test under Mask (TMHH, TMHL, TMLH, TMLL)	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, Low	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

Condition Codes (Cont'd)

Condition Code →	0	1	2	3
Mask Bit Value →	8	4	2	1
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
Translate One to One, One to Two, Two to One, Two to Two	Character not found	Character found	----	CPU determined completion
Unpack ASCII	Sign plus	Sign minus	----	Sign invalid
Unpack Unicode	Sign plus	Sign minus	----	Sign invalid
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
Test Decimal	Digits and sign valid	Sign invalid	Digit invalid	Sign and digit invalid
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)	----	----
Control Instructions				
Compare and Swap and Purge	Equal	Not equal	----	----
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 PSW Extended	See note	See note	See note	See note
Load Real Address	Translation available	Segment-table entry invalid	Page-table entry invalid	See note
Move to Primary	Length ≤ 256	----	----	Length > 256
Move to Secondary	Length ≤ 256	----	----	Length > 256
Move with Key	Length ≤ 256	----	----	Length > 256
Page In	Operation completed	ES data error	----	ES block not available
Page Out	Operation completed	ES data error	----	ES block not available
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
Resume Program	See note	See note	See note	See note
Set Clock	Set	Secure	----	Not operational
Signal Processor	Accepted	Status stored	Busy	Not operational
Store System Information	Info provided	----	----	Info not available
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	Translation not available
Input/Output Instructions				
Cancel Subchannel	Function started	----	----	Not operational
Clear Subchannel	Function started	----	----	Not operational

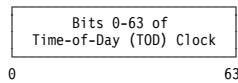
Condition Codes (Cont'd)

Condition Code →	0	1	2	3
Mask Bit Value →	8	4	2	1
Halt Subchannel	Function started	Nonintermediate status pending	Busy	Not operational
Modify Subchannel	Function executed	Status pending	Busy	Not operational
Reset Channel Path	Function started	----	Busy	Not operational
Resume Subchannel	Function started	Status pending	Not applicable	Not operational
Start Subchannel	Function started	Status pending	Busy	Not operational
Store Channel Report Word	CRW stored	Zeros stored	----	----
Store Subchannel	SCHIB stored	----	----	Not operational
Test Pending Interruption	Interruption not pending	Interruption code stored	----	----
Test Subchannel	Status was pending	Status was not pending	----	Not operational

Notes:

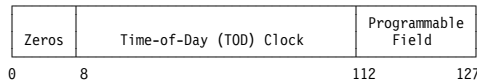
For Diagnose, the resulting condition code is model-dependent.
 For Load Real Address, condition code 3 is set if address-space-control element not available, region-table entry outside table or invalid, segment-table entry outside table, or, for LRA in 24- or 31-bit mode when bits 0-32 of entry address not all zeros, segment- or page-table entry invalid.
 For Load PSW, Load PSW Extended, and Resume Program, the condition code is loaded from the condition-code field of the second operand.
 For Set Program Mask, the condition code is loaded from bit positions 2 and 3 of the first operand.

Operand of Store Clock



Note: Bit 51 of the TOD clock corresponds to one microsecond.

Operand of Store Clock Extended



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

Assembler Instructions

Function	Mnemonic	Meaning
Option control	*PROCESS ACONTROL	Specify assembler options Dynamically modify options
Data definition	CCW CCW0 CCW1 DC DS	Define channel command word Define format-0 channel command word Define format-1 channel command word Define constant Define storage

Assembler Instructions (Cont'd)

Function	Mnemonic	Meaning
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
	XATTR	Declare external symbol attributes
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
Program control	TITLE	Identify assembly output
	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 record
	PUSH	Save current ACONTROL, PRINT, or USING status
	REPRO	Reproduce following record
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	
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 NOP or NOPR	Unconditional Branch No Operation	BC or BCR 15, BC or BCR 0,
After Compare Instructions (A:B)	BH or BHR BL or BLR BE or BER BNH or BNHR BNL or BNLR BNE or BNER	Branch on A High Branch on A Low Branch on A Equal B Branch on A Not High Branch on A Not Low Branch on A Not Equal B	BC or BCR 2, BC or BCR 4, BC or BCR 8, BC or BCR 13, BC or BCR 11, BC or BCR 7,
After Arithmetic Instructions	BP or BPR BM or BMR BZ or BZR BO or BOR BNP or BNPR BNM or BNMR BNZ or BNZR BNO or BNOR	Branch on Plus Branch on Minus Branch on Zero Branch on Overflow Branch on Not Plus Branch on Not Minus Branch on Not Zero Branch on No Overflow	BC or BCR 2, BC or BCR 4, BC or BCR 8, BC or BCR 1, BC or BCR 13, BC or BCR 11, BC or BCR 7, BC or BCR 14,
After Test under Mask instruction	BO or BOR BM or BMR BZ or BZR BNO or BNOR BNM or BNMR BNZ or BNZR	Branch if Ones Branch if Mixed Branch if Zeros Branch if Not Ones Branch if Not Mixed Branch if Not Zeros	BC or BCR 1, BC or BCR 4, BC or BCR 8, BC or BCR 14, BC or BCR 11, BC or BCR 7,

Source: SC26-4940.

*Second operand, not shown, is D₂ (X₂, B₂) for RX format and R₂ for RR format.

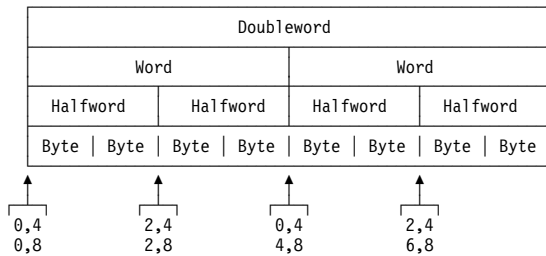
Extended-Mnemonic Instructions for Relative-Branch Instructions

Use	Extended Mnemonic	Meaning	Machine Instr.
General Branch Rel. on Condition	BRU or J BRUL or JLU JNOP*	Unconditional Branch Relative Unconditional Branch Relative No Operation	BRC 15, _{1,2} BRCL 15, _{1,2} BRC 0, _{1,2}
After Compare Instructions (A:B)	BRH or JH* BRL or JL* BRE or JE* BRNH or JNH* BRNL or JNL* BRNE or JNE*	Branch Relative on A High Branch Relative on A Low Branch Relative on A Equal B Branch Relative on A Not High Branch Relative on A Not Low Branch Relative on A Not Equal B	BRC 2, _{1,2} BRC 4, _{1,2} BRC 8, _{1,2} BRC 13, _{1,2} BRC 11, _{1,2} BRC 7, _{1,2}
After Arithmetic Instructions	BRP or JP* BRM or JM* BRZ or JZ* BRO or JO* BRNP or JNP* BRNM or JNM* BRNZ or JNZ* BRNO or JNO*	Branch Relative on Plus Branch Relative on Minus Branch Relative on Zero Branch Relative on Overflow Branch Relative on Not Plus Branch Relative on Not Minus Branch Relative on Not Zero Branch Relative on No Overflow	BRC 2, _{1,2} BRC 4, _{1,2} BRC 8, _{1,2} BRC 1, _{1,2} BRC 13, _{1,2} BRC 11, _{1,2} BRC 7, _{1,2} BRC 14, _{1,2}
After Test under Mask instruction	BRO or JO* BRM or JM* BRZ or JZ* BRNO or JNO* BRNM or JNM* BRNZ or JNZ*	Branch Relative if Ones Branch Relative if Mixed Branch Relative if Zeros Branch Relative if Not Ones Branch Relative if Not Mixed Branch Relative if Not Zeros	BRC 1, _{1,2} BRC 4, _{1,2} BRC 8, _{1,2} BRC 14, _{1,2} BRC 11, _{1,2} BRC 7, _{1,2}
Non-Branch Relative on Condition	JAS JASL JCT JCTG JXH JXHG JXLE JXLEG	Branch Relative and Save Branch Relative and Save Long Branch Relative on Count Branch Relative on Count Branch Relative on Index High Branch Relative on Index High Br. Rel. on Index Low or Equal Br. Rel. on Index Low or Equal	BRAS R ₁ , _{1,2} BRASL R ₁ , _{1,2} BRCT R ₁ , _{1,2} BRCTG R ₁ , _{1,2} BRXH R ₁ ,R ₃ , _{1,2} BRXHG R ₁ ,R ₃ , _{1,2} BRXLE R ₁ ,R ₃ , _{1,2} BRXLG R ₁ ,R ₃ , _{1,2}

Source: SC26-4940.

*To obtain BRCL instead of BRC, add L at the end of the B mnemonic or insert L after the J of the J mnemonic. For example, change BRNZ or JNZ to BRNZL or JNLZ.

CNOP Alignment



Source: SC26-4940.

Summary of Constants

Type	Implied Length, Bytes	Default Alignment	Format	Truncation/ Padding
A	4	Word	Value of address	Left
AD	8	Doubleword	Value of address	Left
B	-	Byte	Binary digits	Left
C	-	Byte	Characters	Right
CU	Even	Byte	Characters, translated to Unicode	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
FD	8	Doubleword	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
LB	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
R	4	Word	PSECT address value	Left
S	2	Halfword	Address in base-displacement form	-
V	-	Word	Externally defined address value	-
X	4	Word	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
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 24)
136-139	R	88	SVC-interruption identification (0-12 zeros, 13-14 ILC, 15 zero, 16-31 code)
140-143	R	8C	Program-interruption identification (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 25))
148-149	R	94	Monitor-class number (0-7 zeros, 8-15 number)
150-151	R	96	PER code (0 successful branching, 1 instruction fetching, 2 storage alteration, 2 and 4 stura, 3 and 5-8 zeros, 8-13 ATMid, 14-15 AI)
152-159	R	98	PER address
160	R	A0	Exception access identification (0-3 zeros, 4-7 access-register number)
161	R	A1	PER access identification (0-3 zeros, 4-7 access-register number)

Fixed Storage Locations (Cont'd)

Area (Dec)	Addr Type	Hex Addr	Function
162	R	A2	Operand access identification (if page-translation exception recognized by Move Page: 0-3 R ₁ , 4-7 R ₂)
163	A/R	A3	Store-status/machine-check architectural-mode identification (0-6 zeros, 7 one)
168-175	R	A8	Translation-exception identification (see table on page 25)
176-183	R	B0	Monitor code
184-187	R	B8	Subsystem-identification word (0-14 zeros, 15 one, 16-31 subchannel number)
188-191	R	BC	I/O-interruption parameter
192-195	R	C0	I/O-interruption-identification word (0-1 zeros, 2-4 I/O-interruption subclass, 5-31 zeros)
200-203	R	C8	STFL facility list (0 certain z/Architecture instructions available, 1 z/Architecture installed, 2 z/Architecture active, 16 extended-translation facility 2 installed)
232-239	R	E8	Machine-check-interruption code (see diagram on page 43)
244-247	R	F4	External-damage code (see diagram on page 43)
248-255	R	F8	Failing-storage address
288-303	R	120	Restart old PSW
304-319	R	130	External old PSW
320-335	R	140	Supervisor-call old PSW
336-351	R	150	Program old PSW
352-367	R	160	Machine-check old PSW
368-383	R	170	Input/output old PSW
416-431	R	1A0	Restart new PSW
432-447	R	1B0	External new PSW
448-463	R	1C0	Supervisor-call new PSW
464-479	R	1D0	Program new PSW
480-495	R	1E0	Machine-check new PSW
496-511	R	1F0	Input/output new PSW
4544-4607	R	11C0	Available for programming
4608-4735	A/R	1200	Store-status/machine-check floating-point-register save area
4736-4863	A/R	1280	Store-status/machine-check general-register save area
4864-4879	A/R	1300	Store-status PSW save area or machine-check fixed-logout area*
4888-4891	A	1318	Store-status prefix save area
4892-4895	A/R	131C	Store-status/machine-check floating-point-control-register save area
4900-4903	A/R	1324	Store-status/machine-check TOD-programmable-register save area
4904-4911	A/R	1328	Store-status/machine-check CPU-timer save area
4913-4919	A/R	1331	Store-status/machine-check clock-comparator bits 0-55 save area (zeros at 4912)
4928-4991	A/R	1340	Store-status/machine-check access-register save area
4992-5119	A/R	1380	Store-status/machine-check control-register save area

A=Absolute address. R=Real address.

A/R=A if store status, R if machine check.

* Contents may vary among models; see System Library manuals.

External-Interruption Codes

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

Code (Hex)	Condition
0040	Interrupt key
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 locations 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
0008	Fixed-point-overflow exception
0009	Fixed-point-divide exception
000A	Decimal-overflow exception
000B	Decimal-divide exception
000C	HFP-exponent-overflow exception
000D	HFP-exponent-underflow exception
000E	HFP-significance exception
000F	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
001C	Space-switch event
001D	HFP-square-root 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
0038	ASCE-type exception
0039	Region-first-translation exception
003A	Region-second-translation exception
003B	Region-third-translation exception
0040	Monitor event
0080	PER event (code may be combined with another code)
0119	Crypto-operation exception

Translation-Exception Identification

At real-storage locations 168-175 (A8-AF hex)

Inter- ruption Code (Hex)	Exception or Event	Format of Information Stored*
0004	Protection	If 61 zero: rest unpredictable If 61 one: suppression, 0-51 address; if DAT was on, 60 one if access-list-controlled protection, 62-63 ASCE identification, rest unpredictable, location 160 valid; if DAT was off, rest unpredictable
0010, 0038, 0039, 003A, 003B 0011	Segment trans- lation; ASCE type; region-first, second, third trans- lation Page translation	0-51 address, 52-61 unpredictable, 62-63 ASCE identification
001C	Space switch	From primary-space mode: 32 old primary-space-switch-event control, 33-47 zeros, 48-63 old PASN From home-space mode: 32 home-space-switch-event control, 33-63 zeros
0020 0021 0022 0023 0024 0025	AFX translation ASX translation LX translation EX translation Primary authority Secondary authority	32-47 zeros, 48-63 address-space number 32-47 zeros, 48-63 address-space number 32-43 zeros, 44-63 program-call number 32-43 zeros, 44-63 program-call number 32-47 zeros, 48-63 address-space number 32-47 zeros, 48-63 address-space number

* Bits 0-31 (bytes 68-71) unchanged if not described.

Data-Exception Code (DXC)

At real-storage location 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	33	SSM-suppression control	SSM instruction	0
	34	TOD-clock-sync control	TOD clock	0
	35	Low-address-protection control	Low-address protection	0
	36	Extraction-authority control	Instruction authorization	0
	37	Secondary-space control	Instruction authorization	0
	38	Fetch-protection-override control	Key-controlled protection	0
	39	Storage-protection-override control	Key-controlled protection	0
	45	AFP-register control	Floating point	0
	48	Malfunction-alert subclass mask	External interruptions	0
	49	Emergency-signal subclass mask	External interruptions	0
	50	External-call subclass mask	External interruptions	0
	52	Clock-comparator subclass mask	External interruptions	0
	53	CPU-timer subclass mask	External interruptions	0
	54	Service-signal subclass mask	External interruptions	0
	56	Unused (See note)		1
	57	Interrupt-key subclass mask	External interruptions	1
	58	Unused (See note)		1
59	ETR subclass mask	External interruptions	0	
61	Crypto control	Cryptography	0	
1	0-63	Primary address-space-control element	Dynamic address translation	0
	0-51	Primary region-table or segment-table origin or real-space token origin	Dynamic address translation	0
	54	Primary subspace-group control	Subspace groups	0
	55	Primary private-space control	Dynamic address translation	0
	56	Primary storage-alteration-event control	Program-event recording	0
	57	Primary space-switch-event control	Program interruptions	0
	58	Primary real-space control	Dynamic address translation	0
60-61	Primary designation-type control	Dynamic address translation	0	
62-63	Primary table length	Dynamic address translation	0	
2	33-57	Dispatchable-unit-control-table origin	Access-register translation	0
3	32-47	PSW-key mask	Instruction authorization	0
	48-63	Secondary ASN	Address spaces	0
4	32-47	Authorization index	Instruction authorization	0
	48-63	Primary ASN	Address spaces	0
5	33-57	Primary-ASTE origin	Access-register translation	0
6	32-39	I/O-interruption subclass mask	I/O interruptions	0
7	0-63	Secondary address-space-control element	Dynamic address translation	0
	0-51	Secondary region-table or segment-table origin or real-space token origin	Dynamic address translation	0
	54	Secondary subspace-group control	Subspace groups	0
	55	Secondary private-space control	Dynamic address translation	0
	56	Secondary storage-alteration-event control	Program-event recording	0
	58	Secondary real-space control	Dynamic address translation	0
	60-61	Secondary designation-type control	Dynamic address translation	0
62-63	Secondary table length	Dynamic address translation	0	
8	32-47	Extended authorization index	Access-register translation	0
	48-63	Monitor masks	MC instruction	0

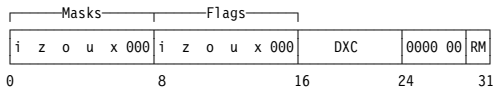
Control Registers (Cont'd)

CR	Bits	Name of Field	Associated with	Init*
9	32	Successful-branching-event mask	Program-event recording	0
	33	Instruction-fetching-event mask	Program-event recording	0
	34	Storage-alteration-event mask	Program-event recording	0
	36	Store-using-real-address-event mask	Program-event recording	0
	40	Branch-address control	Program-event recording	0
	42	Storage-alteration-space control	Program-event recording	0
10	0-63	PER starting address	Program-event recording	0
11	0-63	PER ending address	Program-event recording	0
12	0	Branch-trace control	Tracing	0
	1	Mode-trace control	Tracing	0
	2-61	Trace-entry address	Tracing	0
	62	ASN-trace control	Tracing	0
	63	Explicit-trace control	Tracing	0
13	0-63	Home address-space-control element	Dynamic address translation	0
	0-51	Home region-table or segment-table origin or real-space token origin	Dynamic address translation	0
	54	Home subspace-group control	Subspace groups	0
	55	Home private-space control	Dynamic address translation	0
	56	Home storage-alteration-event control	Program-event recording	0
	57	Home space-switch-event control	Program interruptions	0
	58	Home real-space control	Dynamic address translation	0
	60-61	Home designation-type control	Dynamic address translation	0
62-63	Home table length	Dynamic address translation	0	
14	32	Unused (See note)		1
	33	Unused (See note)		1
	35	Channel-report-pending subclass mask	I/O machine-check handling	0
	36	Recovery subclass mask	Machine-check handling	0
	37	Degradation subclass mask	Machine-check handling	0
	38	External-damage subclass mask	Machine-check handling	1
	39	Warning subclass mask	Machine-check handling	0
	42	TOD-clock-control-override control	TOD clock	0
	44	ASN-translation control	Instruction authorization	0
	45-63	ASN-first-table origin	ASN translation	0
15	0-60	Linkage-stack-entry address	Linkage-stack operations	0

* Value after initial CPU reset.

Note: This bit is not used but is initialized to one for consistency with the System/370 definition.

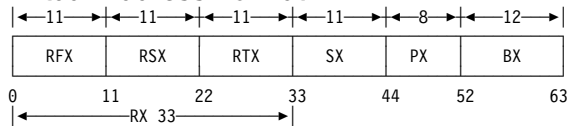
Floating-Point-Control (FPC) Register



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	(SF _i) IEEE-invalid-operation flag
9	(SF _z) IEEE-division-by-zero flag
10	(SF _o) IEEE-overflow flag
11	(SF _u) IEEE-underflow flag
12	(SF _x) IEEE-inexact flag
16-23	(DXC) Data-exception code (see table on page 25)
30-31	(RM) Rounding mode
	00 Round to nearest
	01 Round toward 0
	10 Round toward +∞
	11 Round toward -∞

Dynamic Address Translation

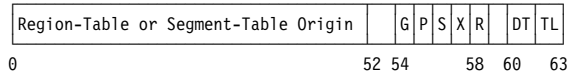
Virtual-Address Format



Bit	Meaning
RX	Region index (region = 2G bytes)
RFX	Region first index
RSX	Region second index
RTX	Region third index
SX	Segment index (segment = 1M bytes)
PX	Page index (page = 4K bytes)
BX	Byte index

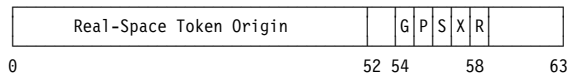
Address-Space-Control Element (ASCE)

Region-Table or Segment-Table Designation (RTD or STD)



Bit	Meaning
54	(G) Subspace-group control
55	(P) Private-space control
56	(S) Storage-alteration-event control
57	(X) Space-switch-event control
58	(R) Real-space control (R = 0)
60-61	(DT) Designation-type control
11	Region-first-table
10	Region-second-table
01	Region-third-table
00	Segment-table
62-63	(TL) Table length (× 4K bytes)

Real-Space Designation (RSD)



Bit	Meaning
58	(R) Real-space control (R = 1)

Note: Other bits are as in RTD or STD.

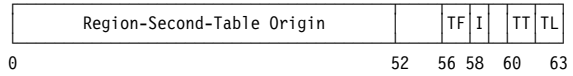
Table Values

Table	Incr. re-ments	Incr. Size	Incr. Ent-ries	Max Size	Max Ent-ries	Max Table Maps	
						Regions	Bytes
Region First	1-4	4KB	512	16KB	2K	8G	16E=16×2 ⁶⁰
Region Second	1-4	4KB	512	16KB	2K	4M	8P= 8×2 ⁵⁰
Region Third	1-4	4KB	512	16KB	2K	2K	4T= 4×2 ⁴⁰
Segment	1-4	4KB	512	16KB	2K	1	2G= 2×2 ³⁰
Page	1	2KB	256	2KB	256	-	1M= 2 ²⁰

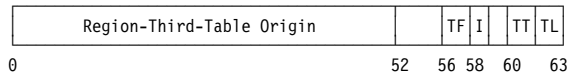
Dynamic Address Translation (Cont'd)

Region-Table Entry (RTE)

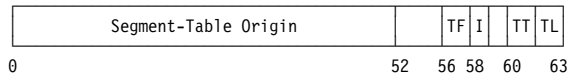
Region-First-Table Entry (RFTE)



Region-Second-Table Entry (RSTE)

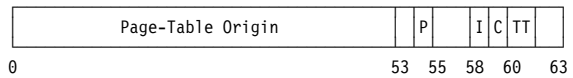


Region-Third-Table Entry (RTTE)



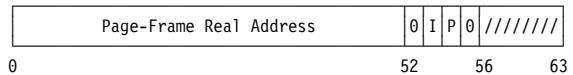
Bit	Meaning
56-57	(TF) Table offset (for next-lower-level table)
58	(I) Invalid bit (for set of regions in RFTE or RSTE, or for region in RTTE)
60-61	(TT) Table-type bits (for this table)
	11 Region first table
	10 Region second table
	01 Region third table
62-63	(TL) Table length (for next-lower-level table) (× 4K bytes)

Segment-Table Entry (STE)



Bit	Meaning
54	(P) Page-protection bit
58	(I) Segment-invalid bit
59	(C) Common-segment bit
60-61	(TT) Table-type bits (for this table)
	00 Segment table

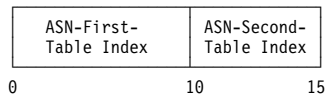
Page-Table Entry (PTE)



Bit	Meaning
53	(I) Page-invalid bit
53	(P) Page-protection bit

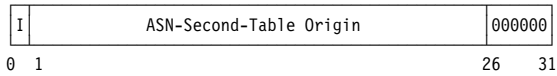
ASN Translation

Address-Space Number (ASN)



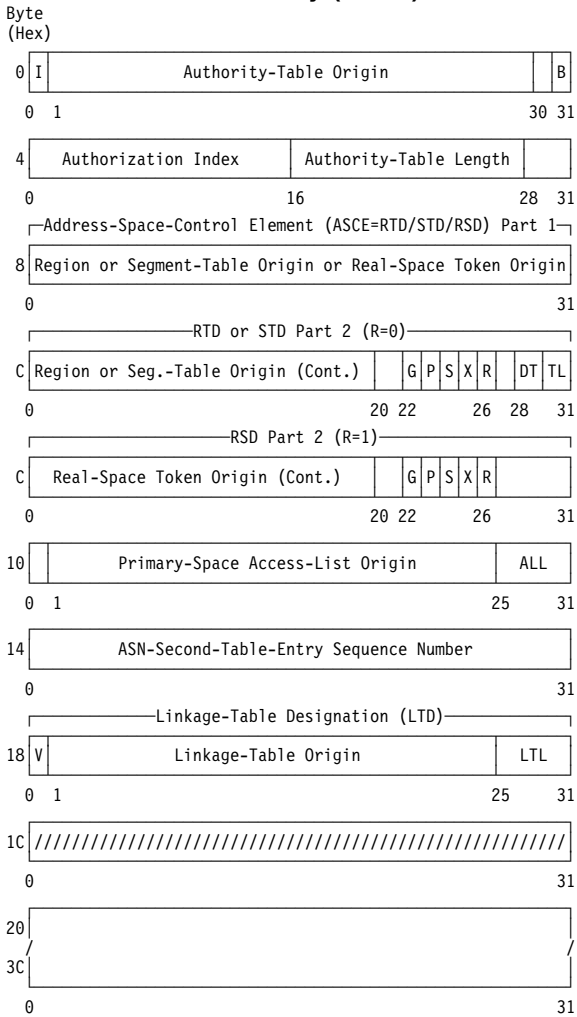
ASN Translation (Cont'd)

ASN-First-Table Entry



Bit **Meaning**
 0 (I) AFX-invalid bit

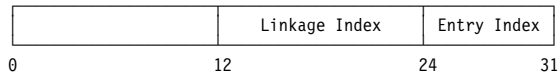
ASN-Second-Table Entry (ASTE)



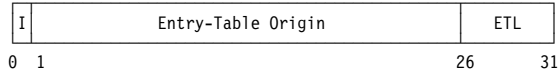
Byte.Bit **Meaning**
 0.0 (I) ASX-invalid bit
 0.31 (B) Base-space bit
 10.25-31 (ALL) Access-list length (x 128 bytes)
 18.0 (V) Subsystem-linkage control
 18.25-31 (LTL) Linkage-table length (x 128 bytes)

PC-Number Translation

Program-Call Number



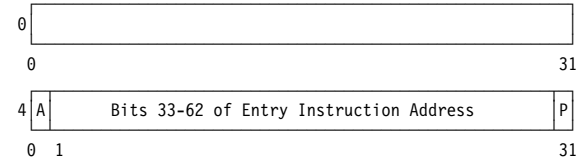
Linkage-Table Entry (LTE)



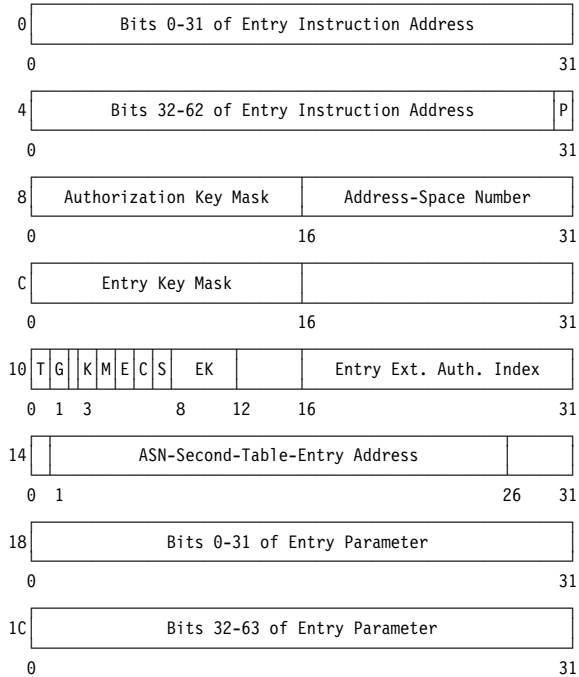
Bit	Meaning
0	(I) LX-invalid bit
26-31	(ETL) Entry-table length (× 128 bytes)

Entry-Table Entry (ETE)

Byte
(Hex)
If Bit 10.1 (G) Is Zero



If Bit 10.1 (G) Is One



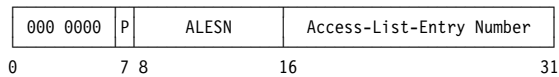
PC-Number Translation (Cont'd)

Entry-Table Entry (ETE) (Cont'd)

Byte.Bit	Meaning
4.0	(A) Entry addressing mode
4.31	(P) Entry problem state
10.0	(T) PC-type bit (zero: basic; one: stacking)
10.1	(G) Entry extended addressing mode
10.3	(K) PSW-key control (zero: unchanged; one: replace if stacking)
10.4	(M) PSW-key-mask control (zero: Or; one: replace if stacking)
10.5	(E) EAX control (zero: unchanged; one: 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

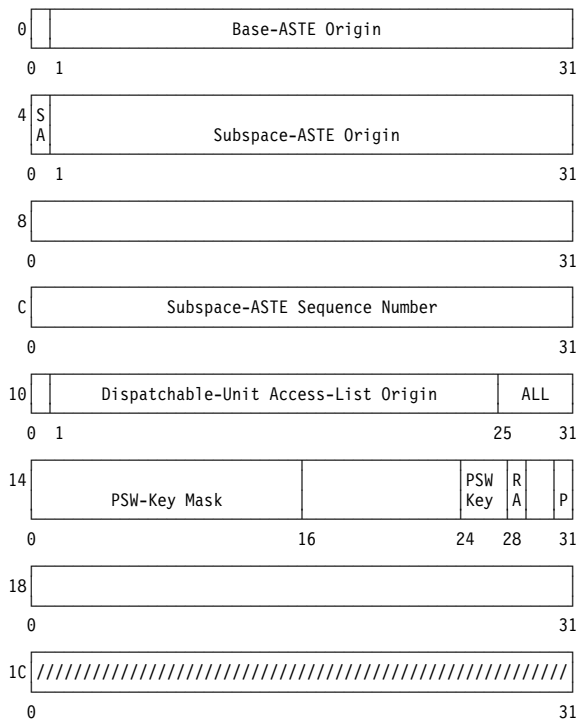
Access-List-Entry Token (ALET)



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

Dispatchable-Unit-Control Table (DUCT)

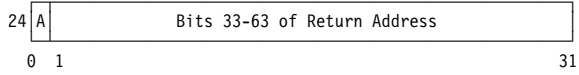
Byte
(Hex)



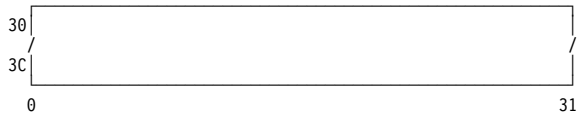
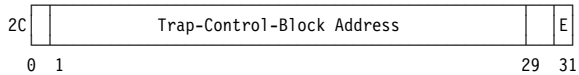
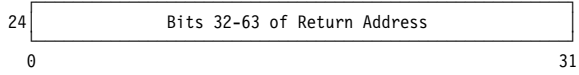
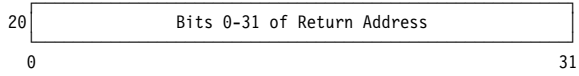
Access-Register Translation (Cont'd)

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

In 24-Bit or 31-Bit Addressing Mode

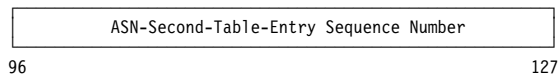
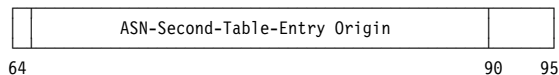
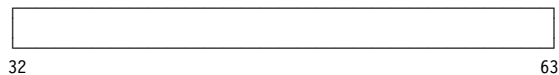


In 64-Bit Addressing Mode



Byte.Bit	Meaning
4.0	(SA) Subspace-active bit
10.25-31	(ALL) Access-list length (× 128 bytes)
14.28	(RA) Reduced-authority bit
14.31	(P) Problem-state bit
2C.31	(E) TRAP-enabled bit
///	Available for programming

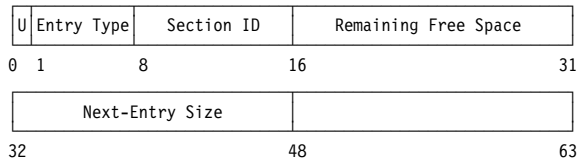
Access-List Entry (ALE)



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

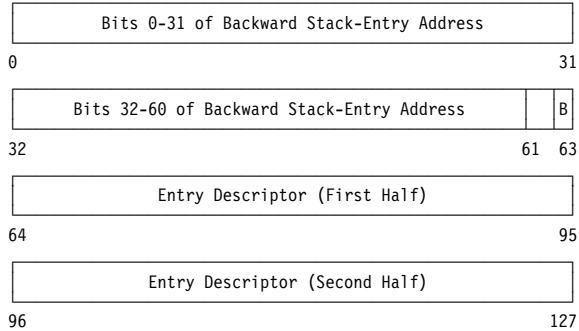
Linkage-Stack Entries

Entry Descriptor



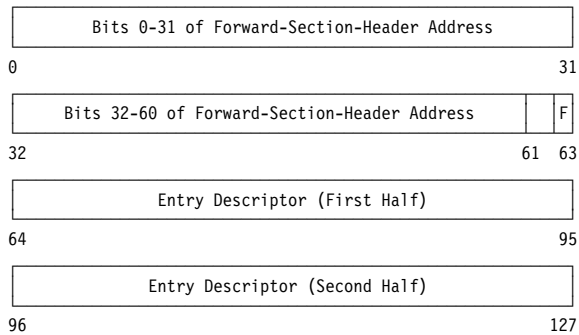
Bit **Meaning**
 0 (U) Unstack-suppression bit
 1-7 Entry type:
 Header entry = 0001001 binary
 Trailer entry = 0001010 binary
 Branch state entry = 0001100 binary
 Program-call state entry = 0001101 binary
 Available for program use = 1xxxxxx binary

Header Entry (Entry Type 0001001)



Bit **Meaning**
 63 (B) Backward stack-entry validity bit

Trailer Entry (Entry Type 0001010)

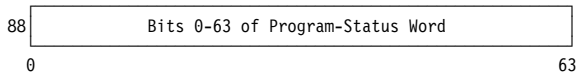
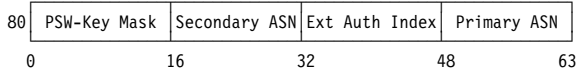
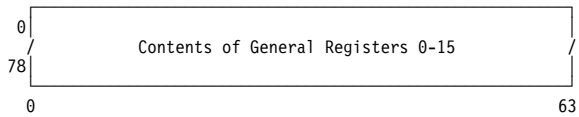


Bit **Meaning**
 63 (F) Forward-section validity bit

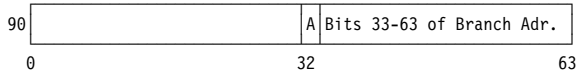
Linkage-Stack Entries (Cont'd)

**Branch State Entry (Entry Type 0001100) and
Program-Call State Entry (Entry Type 0001101)**

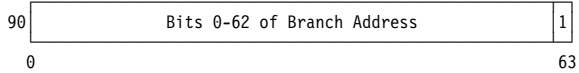
Byte
(Hex)



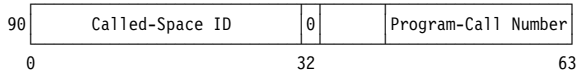
In a Branch State Entry Made in 24-Bit or 31-Bit Mode



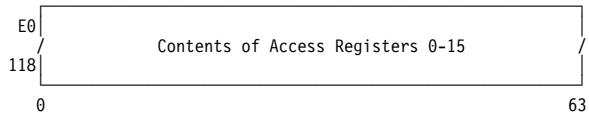
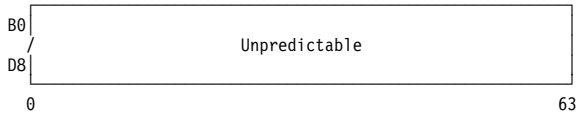
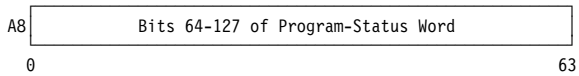
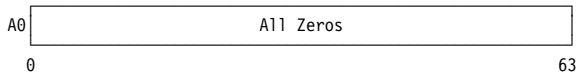
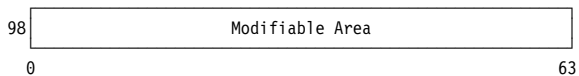
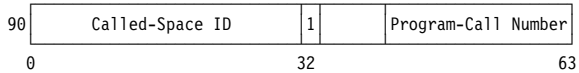
In a Branch State Entry Made in 64-Bit Mode



In a Program-Call State Entry Made on a Call to 24-Bit or 31-Bit Mode

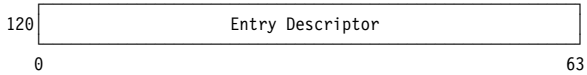


In a Program-Call State Entry Made on a Call to 64-Bit Mode



Linkage-Stack Entries (Cont'd)

Branch State Entry (Entry Type 0001100) and
Program-Call State Entry (Entry Type 0001101) (Cont'd)

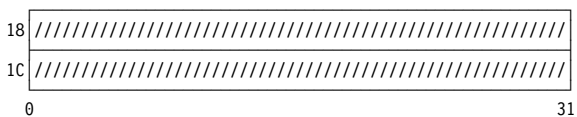
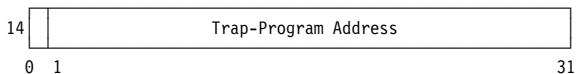
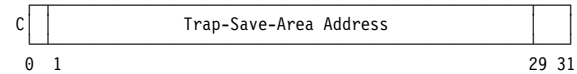
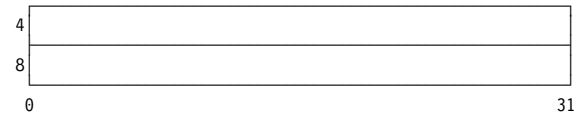
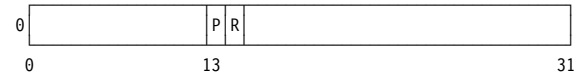


Byte.Bit	Meaning
90.32	(A) Addressing mode (in branch state entry)

Trapping

Trap Control Block

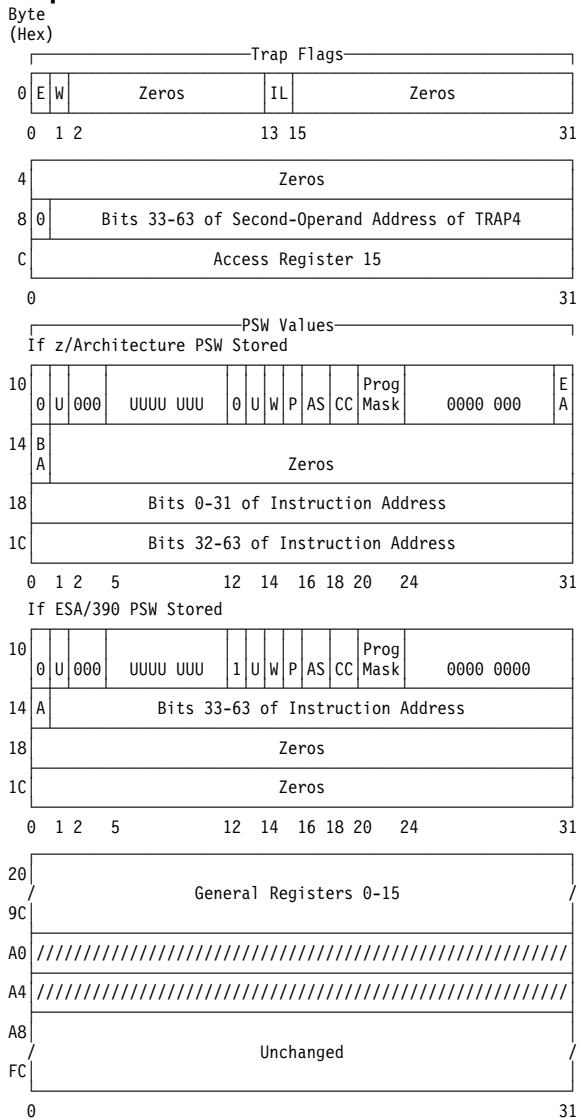
Byte
(Hex)



Byte.Bit	Meaning
0.13	(P) PSW control (zero: PSW.31 must be zero, ESA/390 PSW stored; one: z/Architecture PSW stored)
0.14	(R) General-register control (zero: bits 32-63 stored; one: bits 0-63 stored)
///	Available for programming

Trapping (Cont'd)

Trap Save Area



Byte.Bit	Meaning
0.0	(E) TRAP was target of EXECUTE
0.1	(W) TRAP is TRAP4 (not TRAP2)
0.13-14	(IL) Instruction-length code
10-1F	PSW values (see PSW on page 28)
U	Unpredictable
///	Available for programming

Trace-Entry Formats

Identification of Trace Entries

Trace-Entry Bits			Trace Entry	
0-7	8-11	12-15	Type	Format
00000000			Branch	1
00010000			Set Secondary ASN	1
00100001			Program Call	1
00100010			Program Call	2
00110001		0000	Program Transfer	1
00110001		1000	Program Transfer	2
00110010		0000	Program Return	1
00110010		0010	Program Return	2
00110010		1000	Program Return	4
00110010		1010	Program Return	5
00110010		1100	Program Transfer	3
00110011		0011	Program Return	3
00110011		1011	Program Return	6
00110011		1100	Program Return	7
00110011		1110	Program Return	8
00110100		1111	Program Return	9
01000001			Branch in Subspace Group	1
01000010			Branch in Subspace Group	2
01010001	0010		Mode Switch	2
01010001	0011		Mode Switch	1
01010001	1010		Mode-Switching Branch	1
01010001	1011		Mode-Switching Branch	2
01010010	0110		Mode Switch	3
01010010	1100		Branch	3
01010010	1111		Mode-Switching Branch	3
0111	0		Trace	1
0111	1		Trace	2
1			Branch	2

Branch

F1 (Branch, RP, or TRAP2/4 to 24-Bit Mode)

00000000	Bits 40-63 of Branch Adr.
----------	---------------------------

F2 (Branch, RP, or TRAP2/4 to 31/64-Bit Mode)

1	Bits 33-63 of Branch Address
---	------------------------------

F3 (Branch, RP, or TRAP2/4 to 64-Bit Mode)

01010010	1100	All Zeros	Bits 0-31 of Branch Address
		Bits 32-63 of Branch Address	

Note: "Branch" is BAKR, BALR, BASR, BASSM, BSA, or BSG.

0 8 12 32 63
64 95

Trace-Entry Formats (Cont'd)

Branch in Subspace Group (if ASN Tracing on)

F1 (in 24/31-Bit Mode)

01000001	P	Bits 9-31 of ALET	A	Bits 33-63 of Branch Address
----------	---	-------------------	---	------------------------------

F2 (in 64-Bit Mode)

01000010	P	Bits 9-31 of ALET	Bits 0-31 of Branch Address	
		Bits 32-63 of Branch Address		

Mode Switch

F1 (BASSM, BSM, PC, PR, RP, or SAM64 from 24/31-Bit to 64-Bit Mode)

01010001	0011	All Zeros	A	Updated Instruction Address
----------	------	-----------	---	-----------------------------

F2 (BASSM, BSM, PC, PR, RP, SAM24/31 from 64-Bit to 24/31-Bit Mode)

01010001	0010	All Zeros	Bits 32-63 of Updated Inst. Adr.	
----------	------	-----------	----------------------------------	--

F3 (BASSM, BSM, PC, PR, RP, SAM24/31 from 64-Bit to 24/31-Bit Mode)

01010010	0110	All Zeros	Bits 0-31 of Updated Inst. Adr.	
		Bits 32-63 of Updated Inst. Adr.		

Mode-Switching Branch

F1 (BASSM or RP from 64-Bit to 24/31-Bit Mode)

01010001	1010	All Zeros	A	Branch Address
----------	------	-----------	---	----------------

F2 (BASSM or RP from 24/31-Bit to 64-Bit Mode)

01010001	1011	All Zeros	Bits 32-63 of Branch Address	
----------	------	-----------	------------------------------	--

F3 (BASSM or RP from 24/31-Bit to 64-Bit Mode)

01010010	1111	All Zeros	Bits 0-31 of Branch Address	
		Bits 32-63 of Branch Address		

Program Call

F1 (in 24/31-Bit Mode)

00100001	PSW Key	PC Number	A	Bits 33-62 of Return Address	P
----------	---------	-----------	---	------------------------------	---

F2 (in 64-Bit Mode)

00100010	PSW Key	PC Number	Bits 0-31 of Return Address		
		Bits 32-62 of Return Address			P

0 8 12 32 63
64 95

Trace-Entry Formats (Cont'd)

Program Return

F1 (in 24/31-Bit to 24/31-Bit Mode)

00110010	PSW Key	0000	New PASN	A	Bits 33-62 of Return Address	P
A Bits 33-63 of Updated Inst. Adr.						

F2 (in 64-Bit to 24/31-Bit Mode)

00110010	PSW Key	0010	New PASN	A	Bits 33-62 of Return Address	P
Bits 32-63 of Updated Inst. Adr.						

F3 (in 64-Bit to 24/31-Bit Mode)

00110011	PSW Key	0011	New PASN	A	Bits 33-62 of Return Address	P
Updated Instruction Address						

F4 (in 24/31-Bit to 64-Bit Mode)

00110010	PSW Key	1000	New PASN		Bits 32-62 of Return Address	P
A Bits 33-63 of Updated Inst. Adr.						

F5 (in 64-Bit to 64-Bit Mode)

00110010	PSW Key	1010	New PASN		Bits 32-62 of Return Address	P
Bits 32-63 of Updated Inst. Adr.						

F6 (in 64-Bit to 64-Bit Mode)

00110011	PSW Key	1011	New PASN		Bits 32-62 of Return Address	P
Updated Instruction Address						

F7 (in 24/31-Bit to 64-Bit Mode)

00110011	PSW Key	1100	New PASN		Bits 0-31 of Return Address	
Bits 32-62 of Return Address	P	A	Updated Instruction Address			

F8 (in 64-Bit to 64-Bit Mode)

00110011	PSW Key	1110	New PASN		Bits 0-31 of Return Address	
Bits 32-62 of Return Address	P	Bits 32-63 of Updated Inst. Adr.				

0 8 12 32 63
64 96 127

Trace-Entry Formats (Cont'd)

Program Return (Cont'd)

F9 (in 64-Bit to 64-Bit Mode)

00110100	PSW Key	1111	New PASN	Bits 0-31 of Return Address
Bits 32-62 of Return Address			P	Bits 0-31 of Updated Inst. Adr.
Bits 32-63 of Updated Inst. Adr.				

Program Transfer

F1 (in 24/31-Bit Mode)

00110001	PSW Key	0000	New PASN	Bits 32-63 of R ₂ Before
----------	---------	------	----------	-------------------------------------

F2 (in 64-Bit Mode)

00110001	PSW Key	1000	New PASN	Bits 32-63 of R ₂ Before
----------	---------	------	----------	-------------------------------------

F3 (in 64-Bit Mode)

00110010	PSW Key	1100	New PASN	Bits 0-31 of R ₂ Before
Bits 32-63 of R ₂ Before				

Set Secondary ASN

F1

00010000	00000000	New SASN
----------	----------	----------

Trace

F1 (TRACE)

0111	N	00000000	TOD-Clock Bits 16-63
TRACE Operand		(R ₁) - (R ₃)	

F2 (TRACG)

0111	N	10000000	TOD-Clock Bits 0-47
TOD-Clock Bits 48-79		TRACE Operand	
(R ₁) - (R ₃)			

0	8	12	32	63
64			96	127
128			159	

Bit
4-7

Meaning

(N) One less than the number of registers in the trace entry.

Machine-Check Interruption Code

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

S	P	S	C	E	V	D	C	S	C	V	S	S	K	D	W	M	P	I	F	E	F	G	C	S		
D	D	R	D	D	F	G	W	P	P	K	S	B	E	C	E	S	P	S	M	A	A	C	P	R	R	T
0	4	8	13	16	24	26	31																			

I	A	D						X	A	C	C															
E	R	A	0	0	0	0	0	0	F	P	0	T	C	0	0	0	0	0	0	0	0	0	0	0	0	0
32	35	40	43	46	48																					63

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

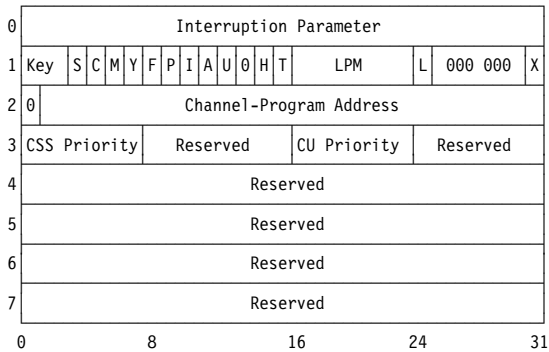
At real-storage address 244-247 (F4-F7 hex)

0	0	0	0	0	0	0	0	0	X	X																
									N	F	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	8	10	16	24	31																					

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

Operation-Request Block (ORB)

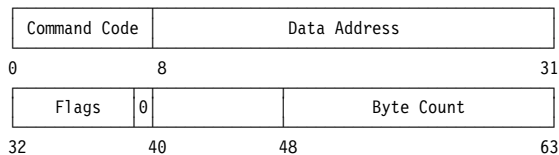
Word



Word.Bit	Meaning
1.0-3	(Key) Subchannel key
1.4	(S) Suspend control
1.5	(C) Streaming-mode control
1.6	(M) Modification control
1.7	(Y) Synchronization 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
1.14	(H) Format-2-IDAW control
1.15	(T) 2K-IDAW control
1.16-23	(LPM) Logical-path mask
1.24	(L) Incorrect-length-suppression mode
1.31	(X) ORB-extension control
3.0-7	Channel-subsystem priority
3.16-23	Control-unit priority

Channel-Command Word (CCW)

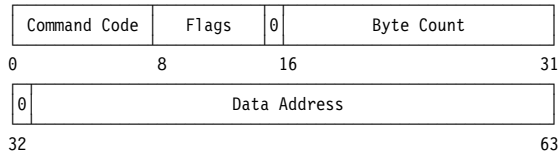
Format-0 CCW



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

Channel-Command Word (CCW) (Cont'd)

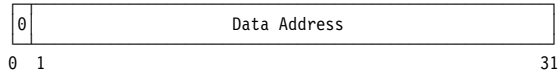
Format-1 CCW



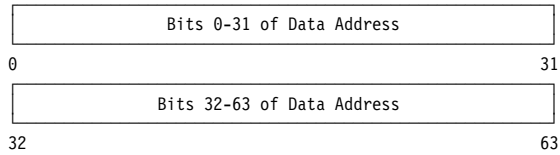
Bit	Meaning
8	(CD) Causes use of data-address portion of next CCW
9	(CC) Causes use of command code and data address of next CCW
10	(SLI) Causes suppression of possible incorrect-length indication
11	(Skip) Suppresses transfer of information to main storage
12	(PCI) Causes an intermediate-interruption 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

Indirect-Data-Address Word (IDAW)

Format-1 IDAW



Format-2 IDAW



Subchannel-Information Block (SCHIB)

Word

0	Path-Management-Control Word
1	
2	
3	
4	
5	
6	
7	Subchannel-Status Word*
8	
9	
10	Model-Dependent Area / Measurement-Block Address
11	
12	Model-Dependent Area

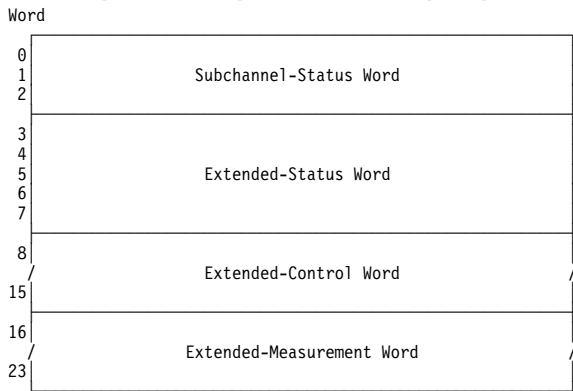
*See "Subchannel-Status Word (SCSW)" on page 47.

Path-Management-Control Word (PMCW)

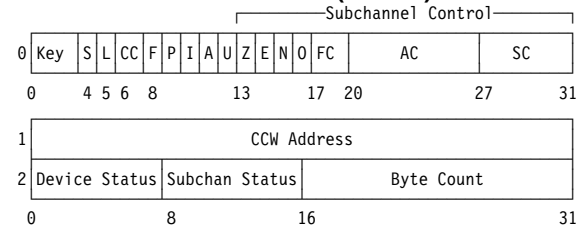
0	Interruption Parameter											
1	00	ISC	000	E	LM	MM	D	T	V	Device Number		
2	LPM			PNOM			LPUM			PIM		
3	MBI						POM			PAM		
4	CHPID-0			CHPID-1			CHPID-2			CHPID-3		
5	CHPID-4			CHPID-5			CHPID-6			CHPID-7		
6	0000 0000			0000 0000			0000 0000			00000 F X S		
0	8			16			24			31		

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)
6.29	(F) Measurement-block-format control
6.30	(X) Extended-measurement-word-mode enable
6.31	(S) Concurrent sense

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

Interruption-Response Block (IRB) (Cont'd)

Extended-Status Word (ESW)

See chart on page 49 to determine the appropriate ESW format.

Format-0 ESW

Word 0	Subchannel Logout
1	Extended-Report Word
2	Failing-Storage Address
3	
4	Secondary-CCW Address

Format-0 ESW Word 0 (Subchannel Logout)

0	ESF	LPUM	0	FVF	SA	TC	D	E	A	SC
0	1	8	16	22	24	26	28	31		

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 (Extended-Report Word)

000	A	P	T	F	S	C	R	SCNT	0000	0000	0000	0000
0	3	8	10	16								31

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
8	(C) Secondary-CCW-address validity
9	(R) Failing-storage-address format (zero: 1-31 of word 2; one: words 2 and 3)
10-15	(SCNT) Concurrent-sense count

Format-1 ESW Word 0*

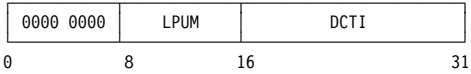
0000	0000	LPUM	0000	0000	0000	0000
0	8	16				31

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

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

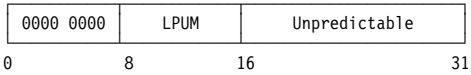
Interruption-Response Block (IRB) (Cont'd)

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			Format	Contents Word 0 Byte 0123
Status-Control Field	L Bit	Suspended Bit	Timing-Facility Bit	Device-Connect-Time Measurement-Mode-Enable Bit	Device-Connect-Time Measurement-Mode Active		
----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						
11001	0						
*1011	*						

These combinations do not occur.

Bit Meaning
 - Not meaningful.
 * Bits may be zeros or ones.
 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 zeros.

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

Interruption-Response Block (IRB) (Cont'd)

Extended-Control Word (ECW)

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

*The contents of the ECW are specified by bits 5 and 14 of word 0 of the SCSW. The combination of SCSW bit 5 zero, SCSW bit 14 one, and ERW bit 7 zero does not occur.

Extended-Measurement Word

Word	
0	Device-Connect Time
1	Function-Pending Time
2	Device-Disconnect Time
3	Control-Unit-Queuing Time
4	Device-Active-Only Time
5	Device-Busy Time
6	Initial-Command-Response Time
7	Reserved

0 31

Format 0 Measurement Block

Word	
0	SSCH + RSCH Count Sample Count
1	Device-Connect Time
2	Function-Pending Time
3	Device-Disconnect Time
4	Control-Unit-Queuing Time
5	Device-Active-Only Time
6	Device-Busy Time
7	Initial-Command-Response Time

0 16 31

Format 1 Measurement Block

Word	
0	SSCH+RSCH Count
1	Sample Count
2	Device-Connect Time
3	Function-Pending Time
4	Device-Disconnect Time
5	Control-Unit-Queuing Time
6	Device-Active-Only Time
7	Device-Busy Time
8	Initial-Command-Response Time
9	Reserved
10	
11	
12	
13	
14	
15	
	0 31

Channel-Report Word (CRW)

0	S	R	C	RSC	A	0	ERC	Reporting-Source ID	31
0	4	8	10	16					

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)
8	(A) Ancillary report
10-15	(ERC) Error-recovery code (see Error-Recovery-Code table)
16-31	Reporting-source ID (see Reporting-Source table)

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

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)

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 a Format-0 CCW
 m -- Modifier bit for specific b Format-1 CCW
 type of I/O device 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)

Code Assignments

Code Table

Dec	Hex	EBCDIC	AS- CII	ISO -8	(1) 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	dblxc1am
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	!	!	!	xc1am
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.

Code Assignments (Cont'd)

Dec	Hex	EBCDIC	AS- CII	ISO -8	(1) 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	SP	SP	SP	SP	@	@	@	atsign
65	41	RSP	RSP	RSP	RSP	RSP	A	A	A	
66	42		â	â	â		B	B	B	ac
67	43		ä	ä	ä		C	C	C	ae
68	44		ã	ã	ã		D	D	D	ag
69	45		ä	ä	ä		E	E	E	aa
70	46		å	å	å		F	F	F	at
71	47		ä	ä	ä		G	G	G	ao
72	48		ç	ç	ç		H	H	H	cc
73	49		ñ	ñ	ñ		I	I	I	nt
74	4A		ç	ç	[ç		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		é	é	é		Q	Q	Q	ea
82	52		ê	ê	ê		R	R	R	ec
83	53		ë	ë	ë		S	S	S	ee

Code Assignments (Cont'd)

Dec	Hex	EBCDIC(4)					AS- CII	ISO -8	IBM -PC	BookMaster Symbol Names(2)
		81C	94C	037	500	1047				
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			ï	ï	ï	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			Ä	Ä	Ä	b	b	b	Ac
99	63			Ä	Ä	Ä	c	c	c	Ae
100	64			Å	Å	Å	d	d	d	Ag
101	65			Å	Å	Å	e	e	e	Aa
102	66			Å	Å	Å	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			É	É	É	r	r	r	Ec
115	73			É	É	É	s	s	s	Ee
116	74			È	È	È	t	t	t	Eg
117	75			Í	Í	Í	u	u	u	Ia
118	76			Í	Í	Í	v	v	v	Ic
119	77			Í	Í	Í	w	w	w	Ie
120	78			ï	ï	ï	x	x	x	Ig
121	79			~	~	~	y	y	y	grave
122	7A	:	:	:	:	:	z	z	z	colon
123	7B	#	#	#	#	#	{	{	{	numsign, lbrk
124	7C	@	@	@	@	@				atsign, vbar
125	7D	'	'	'	'	'	}	}	}	ssq(3), rbrk
126	7E	=	=	=	=	=	~	~	~	eq, eqv
127	7F	"	"	"	"	"	DEL	△	△	sdq, house

Code Assignments (Cont'd)

Dec	Hex	EBCDIC(4)					ISO IBM-PC			BookMaster	
		81C	94C	037	500	1047	-8	437	850	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	ä	ä	ag	
134	86	f	f	f	f	f	SSA	â	â	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	ï	ï	cdqf, ie	
140	8C		ð	ð	ð		PLU	î	î	eth, ic	
141	8D		ý	ý	ý		RI	ï	ï	ya, ig	
142	8E		þ	þ	þ		SS2	Ä	Ä	thorn, Ae	
143	8F		±	±	±		SS3	Ä	Ä	pm, Ao	
144	90		°	°	°		DCS	É	É	degree, Ea	
145	91	j	j	j	j	j	PUI	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	ö	ö	og	
150	96	o	o	o	o	o	SPA	û	û	uc	
151	97	p	p	p	p	p	EPA	û	û	ug	
152	98	q	q	q	q	q	SOS	ÿ	ÿ	ye	
153	99	r	r	r	r	r		Ö	Ö	Oe	
154	9A		ä	ä	ä		SCI	Û	Û	aus, Ue	
155	9B		ø	ø	ø		CSI	ø	ø	ous, cent, os	
156	9C		æ	æ	æ		ST	£	£	aelig, Lsterling	
157	9D		¸	¸	¸		OSC	¥	¥	cedilla, yen, Os	
158	9E		¸	¸	¸		PM	₣	₣	AElig, peseta, mult	
159	9F		¸	¸	¸		ACP	f	f	currency, fnof(5)	
160	A0		μ	μ	μ		RSP	ā	ā	mu(6), aa	
161	A1		˘	˘	˘			ı	ı	tilde, inve, ia	
162	A2	s	s	s	s	s		ç	ç	cent, oa	
163	A3	t	t	t	t	t		£	£	Lsterling, ua	
164	A4	u	u	u	u	u		¸	¸	currency, nt	
165	A5	v	v	v	v	v		¥	¥	yen, Nt	
166	A6	w	w	w	w	w		ı	ı	splitvbar, aus	
167	A7	x	x	x	x	x		§	§	section, ous	
168	A8	y	y	y	y	y		¨	¨	umlaut, invq	
169	A9	z	z	z	z	z		©	©	copyr, lnotrev, regtm	
170	AA		ı	ı	ı			¸	¸	inve, aus, lnot	
171	AB		½	½	½			«	½	invq, odqf, frac12	

Code Assignments (Cont'd)

Dec	Hex	EBCDIC(4)					ISO IBM-PC			BookMaster Symbol Names(2)
		81C	94C	037	500	1047	-8	437	850	
172	AC		Ð	Ð	Ð	¬	¼	¼	Dstroke or Eth, lnot, frac14	
173	AD		Ÿ	Ÿ	[SHY	i	i	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		©	©	©	˘	ı	ı	copyr, acute, bxrj	
181	B5		§	§	§	μ	ı	Ä	section, mu(6), bx1012, Aa	
182	B6		¶	¶	¶	¶	ı	Ä	par, bx2021, Ac	
183	B7		¼	¼	¼	·	ı	Ä	frac14, smultdot, bx0021, Ag	
184	B8		½	½	½	¸	ı	©	frac12, cedilla, bx0012, copyr	
185	B9		¾	¾	¾	¼	ı	ı	frac34, sup1, bx2022	
186	BA		[¬	Ÿ	º	ı	ı	lbrk, lnot, Ya, ous, bx2020	
187	BB]		¨	»	ı	ı	rbrk, vbar, umlaut, cdqf, bx0022	
188	BC		-	-	-	¼	ı	ı	overline, frac14, bx2002	
189	BD		¨	¨]	½	ı	¢	umlaut, rbrk, frac12, bx2001, cent	
190	BE		˘	˘	˘	¾	ı	¥	acute, frac34, bx1002, yen	
191	BF		×	×	×	ı	ı	ı	mult, invq, bxur	
192	C0		{	{	{	Ä	ı	ı	lbrc, Ag, bx11	
193	C1	A	A	A	A	Ä	ı	ı	Aa, bxbj	
194	C2	B	B	B	B	Ä	ı	ı	Ac, bxtj	
195	C3	C	C	C	C	Ä	ı	ı	At, bxlj	
196	C4	D	D	D	D	Ä	-	-	Ae, bxh	
197	C5	E	E	E	E	Ä	ı	ı	Ao, bxcj	
198	C6	F	F	F	F	Æ	ı	ä	AElig, bx1210, at	
199	C7	G	G	G	G	Ç	ı	Ä	Cc, bx2120, At	
200	C8	H	H	H	H	È	ı	ı	Eg, bx2200	
201	C9	I	I	I	I	È	ı	ı	Ea, bx0220	
202	CA	SHY	SHY	SHY	SHY	È	ı	ı	Ec, bx2202	
203	CB		ô	ô	ô	È	ı	ı	oc, Ee, bx0222	
204	CC		ö	ö	ö	İ	ı	ı	oe, Ig, bx2220	
205	CD		ö	ö	ö	İ	=	=	og, Ia, bx0202	
206	CE		ö	ö	ö	İ	ı	ı	oa, Ic, bx2222	
207	CF		ö	ö	ö	İ	=	ı	ot, Ie, bx1202, currency	

Code Assignments (Cont'd)

Dec	Hex	EBCDIC(4)					ISO IBM-PC			BookMaster Symbol Names(2)
		81C	94C	037	500	1047	-8	437	850	
208	D0			}	}	}	Ð	⋈	ð	rbrc, Dstroke or Eth, bx2101, eth
209	D1	J	J	J	J	J	Ñ	ƚ	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	⋈	Ë	Oc, bx1200, Eg
213	D5	N	N	N	N	N	Õ	ƚ	ı	Ot, bx0210, idotless
214	D6	O	O	O	O	O	Ö	⋈	İ	Oe, bx0120, Ia
215	D7	P	P	P	P	P	×	⋈	İ	mult, bx2121, Ic
216	D8	Q	Q	Q	Q	Q	Ø	ƚ	İ	Os, bx1212, Ie
217	D9	R	R	R	R	R	Ù	ƚ	ı	Ug, bx1r
218	DA			ı	ı	ı	Ú	ƚ	ı	sup1, Ua, bxu1
219	DB			û	û	û	Û	■	■	uc, Uc, BOX
220	DC			ü	ü	ü	Ü	■	■	ue, Ue, BOXBOT
221	DD			Û	Û	Û	Ý	■	ı	ug, Ya, BOXLEFT, splitvbar
222	DE			ú	ú	ú	þ	■	İ	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	÷	÷	÷	â	β	β	div, aa, ss
226	E2	S	S	S	S	S	ã	ƚ	Œ	ac, Gamma, Oc
227	E3	T	T	T	T	T	ä	π	Œ	at, pi, Og
228	E4	U	U	U	U	U	ä	Σ	ō	ae, Sigma, ot
229	E5	V	V	V	V	V	å	σ	Œ	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	Ū	ea, Theta(5), Ua
234	EA			z	z	z	ê	Ω	Ū	sup2, ec, Omega, Uc
235	EB			ô	ô	ô	ë	δ	Ū	Oc, ee, delta, Ug
236	EC			ö	ö	ö	ï	∞	ý	Oe, ig, infinity, ya
237	ED			ø	ø	ø	í	φ	Ý	Og, ia, phi, Ya
238	EE			ó	ó	ó	î	ε	ı	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	≤	¼	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)					ISO IBM-PC			BookMaster	
		81C	94C	037	500	1047	-8	437	850	Symbol	Names (2)
248	F8	8	8	8	8	8	ø	°	°	os,	degree
249	F9	9	9	9	9	9	û	•	¨	ug,	1multidot, umlaut
250	FA			3	3	3	ú	.	.	sup3,	ua, smultdot
251	FB			0	0	0	û	√	1	Uc,	uc, sqrt, sup1
252	FC			Ü	Ü	Ü	ü	°	3	Ue,	ue, supn, sup3
253	FD			Û	Û	Û	ÿ	2	2	Ug,	ya, sup2
254	FE			Ů	Ů	Ů	þ	■	■	Ua,	thorn, sqbul
255	FF	E0	E0	E0	E0	E0	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 - CECF (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, œ, and Ø 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	IT	Indent Tab
BEL	Bell	ITB	Intermediate Transmission Block
BS	Backspace	IUS	International Unit Separator
BYP	Bypass	LF	Line Feed
CAN	Cancel	MFA	Modify Field Attribute
CR	Carriage Return	NAK	Negative Acknowledge
CSP	Control Sequence Prefix	NBS	Numeric Backspace
CU1	Customer Use 1	NL	New Line
CU3	Customer Use 3	NUL	Null
DC1	Device Control 1	POC	Program-Operator Communication
DC2	Device Control 2	PP	Presentation Position
DC3	Device Control 3	RES	Restore
DC4	Device Control 4	RFF	Required Form Feed
DEL	Delete	RNL	Required New Line
DLE	Data Link Escape	RPT	Repeat
DS	Digit Select	SA	Set Attribute
EM	End of Medium	SBS	Subscript
ENP	Enable Presentation	SEL	Select
ENQ	Enquiry	SFE	Start Field Extended
EO	Eight Ones	SI	Shift In
EOT	End of Transmission	SM	Set Mode
ESC	Escape	SO	Shift Out
ETB	End of Transmission Block	SOH	Start of Heading
ETX	End of Text	SOS	Start of Significance
FF	Form Feed	SPS	Superscript
FS	Field Separator	STX	Start of Text
GE	Graphic Escape	SUB	Substitute
HT	Horizontal Tab	SW	Switch
IFS	Interchange File Separator	SYN	Synchronous Idle
IGS	Interchange Group Separator	TRN	Transparent
INP	Inhibit Presentation	UBS	Unit Backspace
IR	Index Return	VT	Vertical Tab
IRS	Interchange Record Separator	WUS	Word Underscore

Code Assignments (Cont'd)

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
HTS	Character Tabulation Set	SOS	Start of String
IFS	Information Separator Four	SPA	Start of Guarded Area
IGS	Information Separator Three	SSA	Start of Selected Area
IND	Index	SS2	Single Shift Two
IRS	Information Separator Two	SS3	Single Shift Three
MW	Message Waiting	ST	String Terminator
NBH	No Break Here	STS	Set Transmit State
NEL	Next Line	US	Information Separator One
		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: Hexadecimal equivalents of all numbers from 0 to 255 are listed in the code tables.

Hexadecimal and Decimal Conversion (Cont'd)

Bits:		Halfword				Word				Halfword				Byte			
		0123		4567		0123		4567		0123		4567		0123		4567	
Hex	Dec	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	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	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	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	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	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	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	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	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	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	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	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	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	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	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	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	F	15
	8		7		6		5		4		3		2		1		1

Powers of 2 and 16

m	n	2^m and 16^n	Symbol
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	K (kilo)
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	M (mega)
21		2 097 152	
22		4 194 304	
23		8 388 608	
24	6	16 777 216	
25		33 554 432	
26		67 108 864	
27		134 217 728	
28	7	268 435 456	
29		536 870 912	
30		1 073 741 824	G (giga)
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	T (tera)
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	P (peta)
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		36 028 797 018 963 968	
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	E (exa)
61		2 305 843 009 213 693 952	
62		4 611 686 018 427 387 904	
63		9 223 372 036 854 775 808	
64	16	18 446 744 073 709 551 616	



File Number: S390-00



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

SA22-7871-01

