z/Architecture

Reference Summary

SA22-7871-04

z/Architecture

Reference Summary

SA22-7871-04

Fourth Edition (February, 2008)

This revision differs from the previous edition by containing instructions related to the facilities marked by a bar under “Facility” in “Preface” and minor corrections and clarifications. 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 2001-2008. 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**
ASN-and-LX reuse	EPAIR, ESAIR, PTI, SSAIR
Compare-and-swap-and-store	CSST
Configuration Topology	PTF
DAT enhancement	CSPG, IDTE
DAT enhancement 2	LPTEA
Decimal-floating-point	ADTR, AXTR, CDGTR, CDSTR, CDTR, CDUTR, CEDTR, CEXTR, CGDTR, CGXTR, CSDTR, CSXTR, CUDTR, CUXTR, CXGTR, CXSTR, CXTR, CXUTR, DDTR, DXTR, EEDTR, EEXTR, ESDTR, ESXTR, FIDTR, FIXTR, IEDTR, IEXTR, KDTR, KXTR, LDETR, LDXTR, LEDTR, LTDTR, LTXTR, LXDTR, MDTR, MXTR, QADTR, QAXTR, RRDTR, RRXTR, SDTR, SLDT, SLXT, SRDT, SRXT, SXTR, TDCDT, TDCET, TDCXT, TDGDT, TDGET, TDGXT
Decimal-floating-point-rounding	SRNMT
Enhanced DAT	PFMF
Execute extensions	EXRL
Expanded storage	PGIN, PGOUT
Extended immediate	AFI, AGFI, ALFI, ALGFI, CFI, CGFI, CLFI, CLGFI, FLOGR, IIHF, IILF, LBR, LGBR, LGHR, LGFI, LHR, LLC, LLCR, LLGCR, LLGHR, LLH, LLHR, LLIHF, LLILF, LT, LTG, NIHF, NILF, OIHF, OILF, SLFI, SLGFI, XIHF, XILF
Extended translation 2	CLCLU, MVCLU, PKA, PKU, TP, TROO, TROT, TRTO, TRTT, UNPKA, UNPKU
Extended translation 3	CU14, CU24, CU41, CU42, SRSTU, TRTR
Extract CPU time	ECTG
Floating-point-support-sign-handling	CPSDR, LCDFR, LNDFR, LPDFR
FPR-GR-transfer	LDGR, LGDR
General-instructions-extension	ASI, AGSI, ALSI, ALGSI, CRB, CGRB, CRJ, CGRJ, CRT, CGRT, CGH, CHHSI, CHSI, CGHSI, CHRL, CGHRL, CIB, CGIB, CIJ, CGIJ, CIT, CGIT, CLRB, CLGRB, CLRJ, CLGRJ, CLRT, CLGRT, CLHHSI, CLFHSI, CLGHSI, CLIB, CLGIB, CLIJ, CLGIJ, CLFIT, CLGIT, CLRL, CLHRL, CLGRL, CLGHRL, CLGFRL, CRL, CGRL, CGFRL, ECAG, LAEY, LTGF, LHRL, LGHRL, LLHRL, LLGHRL, LLGFRL, LRL, LGRL, LGFRL, MVHHI, MVHI, MVGHI, MFY, MHY, MSFI, MSGFI, PFD, PFDRL, RNSBG, RXSBG, RISBG, ROSBG, STHRL, STRL, STGRL
HFP multiply-and-add/subtract	MAD, MADR, MAE, MAER, MSD, MSDR, MSE, MSER
HFP unnormalized extensions	MAY, MAYR, MAYH, MAYHR, MAYL, MAYLR, MY, MYH, MYL, MYR, MYHR, MYLR
IEEE-Exception-Simulation	LFAS, SFASR
Long displacement	AHY, ALY, AY, CDSY, CHY, CLIY, 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
Move-with-optional-specifications	MVCOS
Parsing enhancement	TRTE, TRTRE

Facility	Instruction
Perform-floating-point-operation	PFPO
Store-clock fast	STCKF
Store-facility-list extended	STFLE
TOD-clock steering	PTFF

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.

Note: IBM, z/Architecture, zSeries, Enterprise Systems Architecture/390, and ESA/390 are trademarks of the International Business Machines Corporation in the United States, other countries, or both.

Contents

Preface iii

Contents v

Machine Instruction Formats 1

Machine Instructions by Mnemonic 4

Machine Instructions by Operation Code 19

Condition Codes 23

Assembler Instructions 27

- Extended-Mnemonic Instructions for Branch on Condition 28
- Extended-Mnemonic Instructions for Relative-Branch Instructions 28
- Extended-Mnemonic Suffixes for Compare-and-Branch and Compare-and-Trap Instructions 29
- CNOP Alignment 29
- Summary of Constants 29

Operand of Store Clock 30

Operand of Store Clock Extended 30

Fixed Storage Locations 30

External-Interruption Codes 31

Program-Interruption Codes 31

Data-Exception Code (DXC) 32

Translation-Exception Identification 33

Facility Indications 34

Control Registers 35

Floating-Point-Control (FPC) Register 36

Program-Status Word (PSW) 37

- z/Architecture PSW 37
- ESA/390 PSW 37

Dynamic Address Translation 38

- Virtual-Address Format 38
- Address-Space-Control Element (ASCE) 38
- Region-Table or Segment-Table Designation (RTD or STD) 38
- Real-Space Designation (RSD) 38
- Table Values 38
- Region-Table Entry (RTE) 39
- Segment-Table Entry (STE, FC=0) 39
- Segment-Table Entry (STE, FC=1) 39
- Page-Table Entry (PTE) 39

ASN Translation 40

- Address-Space Number (ASN) 40
- ASN-First-Table Entry 40
- ASN-Second-Table Entry (ASTE) 40

PC-Number Translation 41

- Program-Call Number (20-Bit) 41
- Program-Call Number (32-Bit, Bit 44=0) 41
- Program-Call Number (32-Bit, Bit 44=1) 41
- Linkage-Table Entry (LTE) 41
- Linkage-First-Table Entry (LFTE) 41
- Linkage-Second-Table Entry (LSTE) 42
- Entry-Table Entry (ETE) 42

Access-Register Translation 43

- Access-List-Entry Token (ALET) 43
- Dispatchable-Unit-Control Table (DUCT) 43
- Access-List Entry (ALE) 44

Linkage-Stack Entries 44
- Entry Descriptor 44
- Header Entry (Entry Type 0001001) 45
- Trailer Entry (Entry Type 0001010) 45
- Branch State Entry (Entry Type 0001100) and Program-Call State Entry (Entry Type 0001101) 45

Trapping 46
- Trap Control Block 46
- Trap Save Area 47

Trace-Entry Formats 48
- Identification of Trace Entries 48
- Branch 49
- Branch in Subspace Group (if ASN Tracing on) 49
- Mode Switch 49
- Mode-Switching Branch. 50
- Program Call 50
- Program Return 51
- Program Transfer. 53
- Set Secondary ASN. 53
- Trace 53

Machine-Check Interruption Code. 54
- External-Damage Code 54

Operation-Request Block (ORB) 55

Channel-Command Word (CCW) 55
- Format-0 CCW 55
- Format-1 CCW 56

Indirect-Data-Address Word (IDAW) 56
- Format-1 IDAW 56
- Format-2 IDAW 56

Modified-CCW-Indirect-Data-Address Word (MIDAW) 56

Subchannel-Information Block (SCHIB) 57
- Path-Management-Control Word (PMCW) 57

Interruption-Response Block (IRB) 58
- Subchannel-Status Word (SCSW) 58
- Extended-Status Word (ESW). 59
- Format-0 ESW 59
- Format-0 ESW Word 0 (Subchannel Logout) 59
- Format-0 ESW Word 1 (Extended-Report Word) 60
- Format-1 ESW Word 0 60
- Format-2 ESW Word 0[1] 60
- Format-3 ESW Word 0[1] 60
- Information Stored in ESW 60
- Extended-Control Word (ECW) 61
- Extended-Measurement Word 62

Format 0 Measurement Block 62

Format 1 Measurement Block 62

Channel-Report Word (CRW) 63
- Error-Recovery Codes 63
- Reporting Source. 63

I/O Command Codes. 63
- Standard Command-Code Assignments (CCW Bits 0-7) 63
- Standard Meanings of Bits of First Sense Byte 64

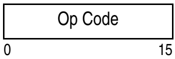
Character Assignments 65
- Control Character Representations 67

- Additional ISO-8 Control Character Representations 67
- Formatting Character Representations 67
- Two-Character BSC Data Link Controls 67
- Commonly Used Editing Pattern Characters 67
- ANSI-Defined Printer Control Characters 68

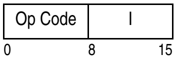
Hexadecimal and Decimal Conversion 68
- Powers of 2 and 16 70

Machine Instruction Formats

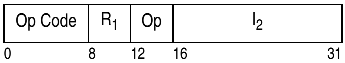
E



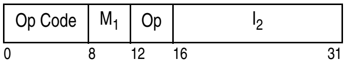
I


Op Code
I
0
8
15

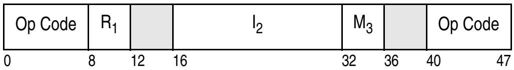
RI$_1$


Op Code
R_1
Op
I_2
0
8
12
16
31

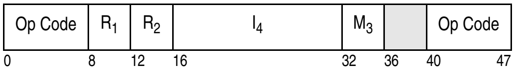
RI$_2$


Op Code
M_1
Op
I_2
0
8
12
16
31

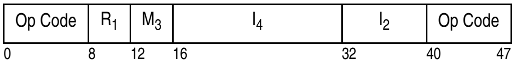
RIE$_1$


Op Code
R_1
I_2
M_3
Op Code
0
8
12
16
32
36
40
47

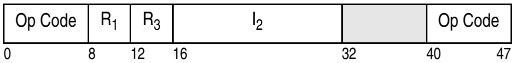
RIE$_2$


Op Code
R_1
R_2
I_4
M_3
Op Code
0
8
12
16
32
36
40
47

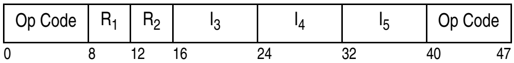
RIE$_3$


Op Code
R_1
M_3
I_4
I_2
Op Code
0
8
12
16
32
40
47

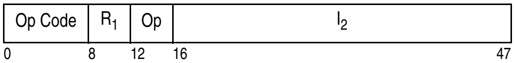
RIE$_4$


Op Code
R_1
R_3
I_2
Op Code
0
8
12
16
32
40
47

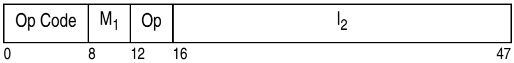
RIE$_5$


Op Code
R_1
R_2
I_3
I_4
I_5
Op Code
0
8
12
16
24
32
40
47

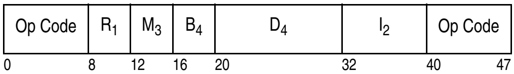
RIL$_1$


Op Code
R_1
Op
I_2
0
8
12
16
47

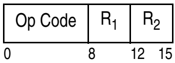
RIL$_2$


Op Code
M_1
Op
I_2
0
8
12
16
47

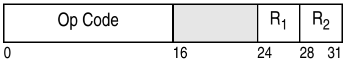
RIS


Op Code
R_1
M_3
B_4
D_4
I_2
Op Code
0
8
12
16
20
32
40
47

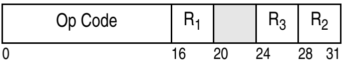
RR


Op Code
R_1
R_2
0
8
12
15

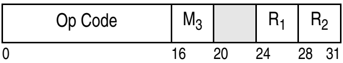
RRE


Op Code
R_1
R_2
0
16
24
28
31

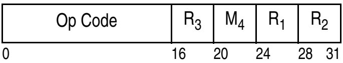
RRF$_1$


Op Code
R_1
R_3
R_2
0
16
20
24
28
31

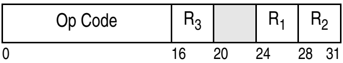
RRF$_2$


Op Code
M_3
R_1
R_2
0
16
20
24
28
31

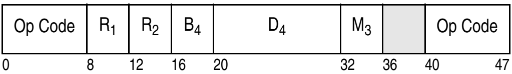
RRF$_3$


Op Code
R_3
M_4
R_1
R_2
0
16
20
24
28
31

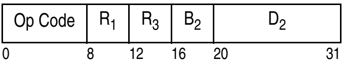
RRR


Op Code
R_3
R_1
R_2
0
16
20
24
28
31

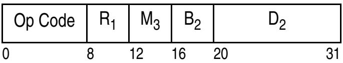
RRS


Op Code
R_1
R_2
B_4
D_4
M_3
Op Code
0
8
12
16
20
32
36
40
47

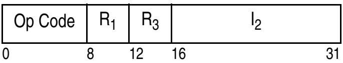
RS$_1$


Op Code
R_1
R_3
B_2
D_2
0
8
12
16
20
31

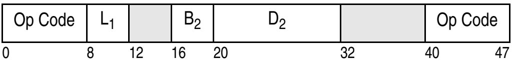
RS$_2$


Op Code
R_1
M_3
B_2
D_2
0
8
12
16
20
31

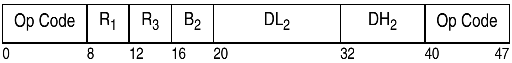
RSI


Op Code
R_1
R_3
I_2
0
8
12
16
31

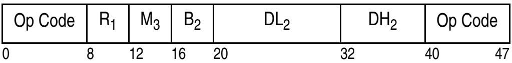
RSL


Op Code
L_1
B_2
D_2
Op Code
0
8
12
16
20
32
40
47

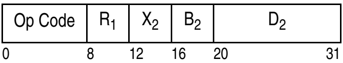
RSY$_1$


Op Code
R_1
R_3
B_2
DL_2
DH_2
Op Code
0
8
12
16
20
32
40
47

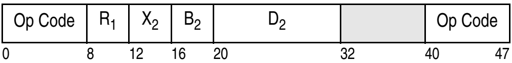
RSY$_2$


Op Code
R_1
M_3
B_2
DL_2
DH_2
Op Code
0
8
12
16
20
32
40
47

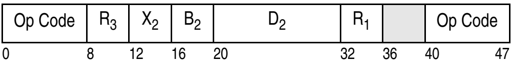
RX


Op Code
R_1
X_2
B_2
D_2
0
8
12
16
20
31

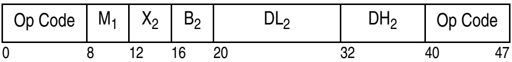
RXE


Op Code
R_1
X_2
B_2
D_2
Op Code
0
8
12
16
20
32
40
47

RXF


Op Code
R_3
X_2
B_2
D_2
R_1
Op Code
0
8
12
16
20
32
36
40
47

RXY$_1$


Op Code
M_1
X_2
B_2
DL_2
DH_2
Op Code
0
8
12
16
20
32
40
47

RXY_2

Op Code	R_1	X_2	B_2	DL_2	DH_2	Op Code
0	8	12	16	20	32	40 47

S

Op Code	B_1	D_1
0	16	20 31

SI

Op Code	I_2	B_1	D_1
0	8	16	20 31

SIL

Op Code	B_1	D_1	I_2
0	16	20	32 47

SIY

Op Code	I_2	B_1	DL_1	DH_1	Op Code
0	8	16	20	32	40 47

SS_1

Op Code	L	B_1	D_1	B_2	D_2
0	8	16	20	32	36 47

SS_2

Op Code	L_1	L_2	B_1	D_1	B_2	D_2
0	8	12	16	20	32	36 47

SS_3

Op Code	L_1	I_3	B_1	D_1	B_2	D_2
0	8	12	16	20	32	36 47

SS_4

Op Code	R_1	R_3	B_1	D_1	B_2	D_2
0	8	12	16	20	32	36 47

SS_5

Op Code	R_1	R_3	B_2	D_2	B_4	D_4
0	8	12	16	20	32	36 47

SSE

Op Code	B_1	D_1	B_2	D_2
0	16	20	32	36 47

SSF

Op Code	R_3	Op	B_1	D_1	B_2	D_2
0	8	12	16	20	32	36 47

1, 2, 3, 4, 5	Denotes association with first, second, third, fourth, or fifth operand; distinguishes among instances of the same basic instruction format
B_1, B_2, B_4	Base register designation field
D_1, D_2, D_4	Displacement field
I, I_2, I_3, I_4, I_5	Immediate operand field
L, L_1, L_2	Length field
M_1, M_3, M_4	Mask field
R_1, R_2, R_3	Register designation field
X_2	Index register designation field

Machine Instructions by Mnemonic

Mnemonic	Operands	Name	Format	Opcode	Class & Notes
A	$R_1,D_2(X_2,B_2)$	Add (32)	RX	5A	c
AD	$R_1,D_2(X_2,B_2)$	Add Normalized (LH)	RX	6A	c
ADB	$R_1,D_2(X_2,B_2)$	Add (LB)	RXE	ED1A	c
ADBR	R_1,R_2	Add (LB)	RRE	B31A	c
ADR	R_1,R_2	Add Normalized (LH)	RR	2A	c
ADTR	R_1,R_2,R_3	Add (LD)	RRR	B3D2	c TF
AE	$R_1,D_2(X_2,B_2)$	Add Normalized (SH)	RX	7A	c
AEB	$R_1,D_2(X_2,B_2)$	Add (SB)	RXE	ED0A	c
AEBR	R_1,R_2	Add (SB)	RRE	B30A	c
AER	R_1,R_2	Add Normalized (SH)	RR	3A	c
AFI	R_1,I_2	Add Immediate (32)	RIL	C29	c EI
AG	$R_1,D_2(X_2,B_2)$	Add (64)	RXY$_2$	E308	c N
AGF	$R_1,D_2(X_2,B_2)$	Add (64←32)	RXY$_2$	E318	c N
AGFI	R_1,I_2	Add Immediate (64←32)	RIL	C28	c EI
AGFR	R_1,R_2	Add (64←32)	RRE	B918	c N
AGHI	R_1,I_2	Add Halfword Immediate (64←16)	RI$_1$	A7B	c N
AGR	R_1,R_2	Add (64)	RRE	B908	c N
AGSI	$D_1(B_1),I_2$	Add Immediate (64←8)	SIY	EB7A	c GE
AH	$R_1,D_2(X_2,B_2)$	Add Halfword (32←16)	RX	4A	c
AHI	R_1,I_2	Add Halfword Immediate (32←16)	RI$_1$	A7A	c
AHY	$R_1,D_2(X_2,B_2)$	Add Halfword (32←16)	RXY$_2$	E37A	c LD
AL	$R_1,D_2(X_2,B_2)$	Add Logical (32)	RX	5E	c
ALC	$R_1,D_2(X_2,B_2)$	Add Logical with Carry (32)	RXY$_2$	E398	c N3
ALCG	$R_1,D_2(X_2,B_2)$	Add Logical with Carry (64)	RXY$_2$	E388	c N
ALCGR	R_1,R_2	Add Logical with Carry (64)	RRE	B988	c N
ALCR	R_1,R_2	Add Logical with Carry (32)	RRE	B998	c N3
ALFI	R_1,I_2	Add Logical Immediate (32)	RIL	C2B	c EI
ALG	$R_1,D_2(X_2,B_2)$	Add Logical (64)	RXY$_2$	E30A	c N
ALGF	$R_1,D_2(X_2,B_2)$	Add Logical (64←32)	RXY$_2$	E31A	c N
ALGFI	R_1,I_2	Add Logical Immediate (64←32)	RIL	C2A	c EI
ALGFR	R_1,R_2	Add Logical (64←32)	RRE	B91A	c N
ALGR	R_1,R_2	Add Logical (64)	RRE	B90A	c N
ALGSI	$D_1(B_1),I_2$	Add Logical with Signed Immediate (64←8)	SIY	EB7E	c GE
ALR	R_1,R_2	Add Logical (32)	RR	1E	c
ALSI	$D_1(B_1),I_2$	Add Logical with Signed Immediate (32←8)	SIY	EB6E	c GE
ALY	$R_1,D_2(X_2,B_2)$	Add Logical (32)	RXY$_2$	E35E	c LD
AP	$D_1(L_1,B_1),D_2(L_2,B_2)$	Add Decimal	SS$_2$	FA	c
AR	R_1,R_2	Add (32)	RR	1A	c
ASI	$D_1(B_1),I_2$	Add Immediate (32←8)	SIY	EB6A	c GE
AU	$R_1,D_2(X_2,B_2)$	Add Unnormalized (SH)	RX	7E	c
AUR	R_1,R_2	Add Unnormalized (SH)	RR	3E	c
AW	$R_1,D_2(X_2,B_2)$	Add Unnormalized (LH)	RX	6E	c
AWR	R_1,R_2	Add Unnormalized (LH)	RR	2E	c
AXBR	R_1,R_2	Add (EB)	RRE	B34A	c
AXR	R_1,R_2	Add Normalized (EH)	RR	36	c
AXTR	R_1,R_2,R_3	Add (ED)	RRR	B3DA	c TF
AY	$R_1,D_2(X_2,B_2)$	Add (32)	RXY$_2$	E35A	c LD
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$_2$	E346	N
BCTGR	R_1,R_2	Branch on Count (64)	RRE	B946	N

Mnemonic	Operands	Name	Format	Opcode	Class & Notes
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_4	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_4	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
CDGTR	R_1,R_2	Convert from Fixed (LD←64)	RRE	B3F1	TF
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
CDSTR	R_1,R_2	Convert from Signed Packed (LD←64)	RRE	B3F3	TF
CDSY	$R_1,R_3,D_2(B_2)$	Compare Double and Swap (32)	RSY_1	EB31	c LD
CDTR	R_1,R_2	Compare (LD)	RRE	B3E4	c TF
CDUTR	R_1,R_2	Convert from Unsigned Packed (LD←64)	RRE	B3F2	TF
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
CEDTR	R_1,R_2	Compare Biased Exponent (LD)	RRE	B3F4	c TF
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
CEXTR	R_1,R_2	Compare Biased Exponent (ED)	RRE	B3FC	c TF
CFC	$D_2(B_2)$	Compare and Form Codeword	S	B21A	ic
CFDBR	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
CFI	R_1,I_2	Compare Immediate (32)	RIL	C2D	c EI
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_2	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
CGDTR	R_1,M_3,R_2	Convert to Fixed (64←LD)	RRF_2	B3E1	c TF
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_2	E330	c N
CGFI	R_1,I_2	Compare Immediate (64←32)	RIL	C2C	c EI

Mnemonic	Operands	Name	Format	Op-code	Class & Notes
CGFR	R_1,R_2	Compare (64←32)	RRE	B930	c N
CGFRL	R_1,I_2	Compare Relative Long (64←32)	RIL	C6C	c GE
CGH	$R_1,D_2(X_2,B_2)$	Compare Halfword (64←16)	RXY$_2$	E334	c GE
CGHI	R_1,I_2	Compare Halfword Immediate (64←16)	RI$_1$	A7F	c N
CGHRL	R_1,I_2	Compare Halfword Relative Long (64←16)	RIL	C64	c GE
CGHSI	$D_1(B_1),I_2$	Compare Halfword Immediate (64←16)	SIL	E558	c GE
CGIB	$R_1,I_2,M_3,D_4(B_4)$	Compare Immediate and Branch (64←8)	RIS	ECFC	GE
CGIJ	R_1,I_2,M_3,I_4	Compare Immediate and Branch Relative (64←8)	RIE$_3$	EC7C	GE
CGIT	R_1,I_2,M_3	Compare Immediate and Trap (64←16)	RIE$_1$	EC70	GE
CGR	R_1,R_2	Compare (64)	RRE	B920	c N
CGRB	$R_1,R_2,M_3,D_4(B_4)$	Compare and Branch (64)	RRS	ECE4	GE
CGRJ	R_1,R_2,M_3,I_4	Compare and Branch Relative (64)	RIE$_2$	EC64	GE
CGRL	R_1,I_2	Compare Relative Long (64)	RIL	C68	c GE
CGRT	R_1,R_2,M_3	Compare and Trap (64)	RRF	B960	GE
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
CGXTR	R_1,M_3,R_2	Convert to Fixed (64←ED)	RRF$_2$	B3E9	c TF
CH	$R_1,D_2(X_2,B_2)$	Compare Halfword (32←16)	RX	49	c
CHHSI	$D_1(B_1),I_2$	Compare Halfword Immediate (16←16)	SIL	E554	c GE
CHI	R_1,I_2	Compare Halfword Immediate (32←16)	RI$_1$	A7E	c
CHRL	R_1,I_2	Compare Halfword Relative Long (32←8)	RIL	C65	c GE
CHSI	$D_1(B_1),I_2$	Compare Halfword Immediate (32←16)	SIL	E55C	c GE
CHY	$R_1,D_2(X_2,B_2)$	Compare Halfword (32←16)	RXY$_2$	E379	c LD
CIB	$R_1,I_2,M_3,D_4(B_4)$	Compare Immediate and Branch (32←8)	RIS	ECFE	GE
CIJ	R_1,I_2,M_3,I_4	Compare Immediate and Branch Relative (32←8)	RIE$_3$	EC7E	GE
CIT	R_1,I_2,M_3	Compare Immediate and Trap (32←16)	RIE$_1$	EC72	GE
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
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
CLFHSI	$D_1(B_1),I_2$	Compare Logical Immediate (32←16)	SIL	E55D	c GE
CLFI	R_1,I_2	Compare Logical Immediate (32)	RIL	C2F	c EI
CLFIT	R_1,I_2,M_3	Compare Logical Immediate and Trap (32←16)	RIE$_1$	EC73	GE
CLG	$R_1,D_2(X_2,B_2)$	Compare Logical (64)	RXY$_2$	E321	c N
CLGF	$R_1,D_2(X_2,B_2)$	Compare Logical (64←32)	RXY$_2$	E331	c N
CLGFI	R_1,I_2	Compare Logical Immediate (64←32)	RIL	C2E	c EI
CLGFR	R_1,R_2	Compare Logical (64←32)	RRE	B931	c N
CLGFRL	R_1,I_2	Compare Logical Relative Long (64←32)	RIL	C6E	c GE
CLGHRL	R_1,I_2	Compare Logical Relative Long (64←16)	RIL	C66	c GE
CLGHSI	$D_1(B_1),I_2$	Compare Logical Immediate (64←16)	SIL	E559	c GE
CLGIB	$R_1,I_2,M_3,D_4(B_4)$	Compare Logical Immediate and Branch (64←8)	RIS	ECFD	GE
CLGIJ	R_1,I_2,M_3,I_4	Compare Logical Immediate and Branch Relative (64←8)	RIE$_3$	EC7D	GE
CLGIT	R_1,I_2,M_3	Compare Logical Immedical and Trap (64←16)	RIE$_1$	EC71	GE
CLGR	R_1,R_2	Compare Logical (64)	RRE	B921	c N
CLGRB	$R_1,R_2,M_3,D_4(B_4)$	Compare Logical and Branch (64)	RRS	ECE5	GE
CLGRJ	R_1,R_2,M_3,I_4	Compare Logical and Branch Relative (64)	RIE$_2$	EC65	GE
CLGRL	R_1,I_2	Compare Logical Relative Long (64)	RIL	C6A	c GE
CLGRT	R_1,R_2,M_3	Compare Logical and Trap (64)	RRF	B961	GE
CLHHSI	$D_1(B_1),I_2$	Compare Logical Immediate (16←16)	SIL	E555	c GE
CLHRL	R_1,I_2	Compare Logical Relative Long (32←16)	RIL	C67	c GE
CLI	$D_1(B_1),I_2$	Compare Logical Immediate	SI	95	c
CLIB	$R_1,I_2,M_3,D_4(B_4)$	Compare Logical Immediate and Branch (32←8)	RIS	ECFF	GE

Mnemonic	Operands	Name	Format	Op-code	Class & Notes
CLIJ	R_1,I_2,M_3,I_4	Compare Logical Immediate and Branch Relative (32←8)	RIE$_3$	EC7F	GE
CLIY	$D_1(B_1),I_2$	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
CLRB	$R_1,R_2,M_3,D_4(B_4)$	Compare Logical and Branch (32)	RRS	ECF7	GE
CLRJ	R_1,R_2,M_3,I_4	Compare Logical and Branch Relative (32)	RIE$_2$	EC77	GE
CLRL	R_1,I_2	Compare Logical Relative Long (32)	RIL	C6F	c GE
CLRT	R_1,R_2,M_3	Compare Logical and Trap (32)	RRF	B973	GE
CLST	R_1,R_2	Compare Logical String	RRE	B25D	c
CLY	$R_1,D_2(X_2,B_2)$	Compare Logical (32)	RXY$_2$	E355	c LD
CMPSC	R_1,R_2	Compression Call	RRE	B263	ic
CP	$D_1(L_1,B_1),D_2(L_2,B_2)$	Compare Decimal	SS$_2$	F9	c
CPSDR	R_1,R_3,R_2	Copy Sign	RRF$_1$	B372	FS
CPYA	R_1,R_2	Copy Access	RRE	B24D	
CR	R_1,R_2	Compare (32)	RR	19	c
CRB	$R_1,R_2,M_3,D_4(B_4)$	Compare and Branch (32)	RRS	ECF6	GE
CRJ	R_1,R_2,M_3,I_4	Compare and Branch Relative (32)	RIE$_2$	EC76	GE
CRL	R_1,I_2	Compare Relative Long (32)	RIL	C6D	c GE
CRT	R_1,R_2,M_3	Compare and Trap (32)	RRF	B972	GE
CS	$R_1,R_3,D_2(B_2)$	Compare and Swap (32)	RS$_1$	BA	c
CSCH		Clear Subchannel	S	B230	pc
CSDTR	R_1,R_2,M_4	Convert to Signed Packed (64←LD)	RRF$_3$	B3E3	TF
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
CSST	$D_1(B_1),D_2(B_2),R_3$	Compare and Swap and Store	SSF	C82	c
CSXTR	R_1,R_2,M_4	Convert to Signed Packed (128←ED)	RRF$_3$	B3EB	TF
CSY	$R_1,R_3,D_2(B_2)$	Compare and Swap (32)	RSY$_1$	EB14	c LD
CU12	$R_1,R_2[,M_3]$	Convert UTF-8 to UTF-16	RRF$_2$	B2A7	c
CU14	$R_1,R_2[,M_3]$	Convert UTF-8 to UTF-32	RRF$_2$	B9B0	c E3
CU21	$R_1,R_2[,M_3]$	Convert UTF-16 to UTF-8	RRF$_2$	B2A6	c
CU24	$R_1,R_2[,M_3]$	Convert UTF-16 to UTF-32	RRF$_2$	B9B1	c E3
CU41	R_1,R_2	Convert UTF-32 to UTF-8	RRE	B9B2	c E3
CU42	R_1,R_2	Convert UTF-32 to UTF-16	RRE	B9B3	c E3
CUDTR	R_1,R_2	Convert to Unsigned Packed (64←LD)	RRE	B3E2	TF
CUSE	R_1,R_2	Compare until Substring Equal	RRE	B257	ic
CUTFU	$R_1,R_2[,M_3]$	Convert UTF-8 to Unicode	RRF$_2$	B2A7	c
CUUTF	$R_1,R_2[,M_3]$	Convert Unicode to UTF-8	RRF$_2$	B2A6	c
CUXTR	R_1,R_2	Convert to Unsigned Packed (128←ED)	RRE	B3EA	TF
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$_2$	E30E	N
CVBY	$R_1,D_2(X_2,B_2)$	Convert to Binary (32)	RXY$_2$	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$_2$	E32E	N
CVDY	$R_1,D_2(X_2,B_2)$	Convert to Decimal (32)	RXY$_2$	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
CXGTR	R_1,R_2	Convert from Fixed (ED←64)	RRE	B3F9	TF
CXR	R_1,R_2	Compare (EH)	RRE	B369	c
CXSTR	R_1,R_2	Convert from Signed Packed (ED←128)	RRE	B3FB	TF
CXTR	R_1,R_2	Compare (ED)	RRE	B3EC	c TF
CXUTR	R_1,R_2	Convert from Unsigned Packed (ED←128)	RRE	B3FA	TF
CY	$R_1,D_2(X_2,B_2)$	Compare (32)	RXY$_2$	E359	c LD
D	$R_1,D_2(X_2,B_2)$	Divide (32←64)	RX	5D	

Mnemonic	Operands	Name	Format	Op-code	Class & Notes
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	
DDTR	R_1,R_2,R_3	Divide (LD)	RRR	B3D1	TF
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$_2$	E397	N3
DLG	$R_1,D_2(X_2,B_2)$	Divide Logical (64←128)	RXY$_2$	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 (32←64)	RR	1D	
DSG	$R_1,D_2(X_2,B_2)$	Divide Single (64)	RXY$_2$	E30D	N
DSGF	$R_1,D_2(X_2,B_2)$	Divide Single (64←32)	RXY$_2$	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	
DXTR	R_1,R_2,R_3	Divide (ED)	RRR	B3D9	TF
EAR	R_1,R_2	Extract Access	RRE	B24F	
ECAG	$R_1,R_3,D_2(B_2)$	Extract Cache Attribute	RSY	EB4C	GE
ECTG	$D_1(B_1),D_2(B_2),R_3$	Extract CPU Time	SSF	C81	ET
ED	$D_1(L,B_1),D_2(B_2)$	Edit	SS$_1$	DE	c
EDMK	$D_1(L,B_1),D_2(B_2)$	Edit and Mark	SS$_1$	DF	c
EEDTR	R_1,R_2	Extract Biased Exponent (64←LD)	RRE	B3E5	TF
EEXTR	R_1,R_2	Extract Biased Exponent (64←ED)	RRE	B3ED	TF
EFPC	R_1	Extract FPC	RRE	B38C	
EPAIR	R_1	Extract Primary ASN and Instance	RRE	B99A	q RA
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
ESAIR	R_1	Extract Secondary ASN and Instance	RRE	B99B	q RA
ESAR	R_1	Extract Secondary ASN	RRE	B227	q
ESDTR	R_1,R_2	Extract Significance (64←LD)	RRE	B3E7	TF
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
ESXTR	R_1,R_2	Extract Significance (64←ED)	RRE	B3EF	TF
EX	$R_1,D_2(X_2,B_2)$	Execute	RX	44	
EXRL	R_1,I_2	Execute Relative Long	RIL	C60	XX
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	
FIDTR	R_1,M_3,R_2,M_4	Load FP Integer (LD)	RRF$_3$	B3D7	TF
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	
FIXTR	R_1,M_3,R_2,M_4	Load FP Integer (ED)	RRF$_3$	B3DF	TF
FLOGR	R_1,R_2	Find Leftmost One	RRE	B983	c EI
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

Mnemonic	Operands	Name	Format	Opcode	Class & Notes
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$_2$	E373	LD
IDTE	R_1,R_3,R_2	Invalidate DAT Table Entry	RRF$_3$	B98E	pu DE
IEDTR	R_1,R_3,R_2	Insert Biased Exponent (LD←64|LD)	RRF$_3$	B3F6	TF
IEXTR	R_1,R_3,R_2	Insert Biased Exponent (ED←64|ED)	RRF$_3$	B3FE	TF
IIHF	R_1,I_2	Insert Immediate (high)	RIL	C08	EI
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
IILF	R_1,I_2	Insert Immediate (low)	RIL	C09	EI
IILH	R_1,I_2	Insert Immediate (low high)	RI$_1$	A52	N
IILL	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
KDTR	R_1,R_2	Compare and Signal (LD)	RRE	B3E0	c TF
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
KXTR	R_1,R_2	Compare and Signal (ED)	RRE	B3E8	c TF
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	
LAEY	$R_1,D_2(X_2,B_2)$	Load Address Extended	RXY$_2$	E375	GE
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$_2$	E371	LD
LB	$R_1,D_2(X_2,B_2)$	Load Byte (32)	RXY$_2$	E376	LD
LBR	R_1,R_2	Load Byte (32)	RRE	B926	EI
LCDBR	R_1,R_2	Load Complement (LB)	RRE	B313	c
LCDFR	R_1,R_2	Load Complement (L)	RRE	B373	FS
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 (SH)	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	
LDETR	R_1,R_2,M_4	Load Lengthened (LD←SD)	RRF$_3$	B3D4	TF
LDGR	R_1,R_2	Load FPR from GR (L←64)	RRE	B3C1	FG
LDR	R_1,R_2	Load (L)	RR	28	

Mnemonic	Operands	Name	Format	Op-code	Class & Notes
LDXBR	R_1,R_2	Load Rounded (LB←EB)	RRE	B345	
LDXR	R_1,R_2	Load Rounded (LH←EH)	RR	25	
LDXTR	R_1,M_3,R_2,M_4	Load Rounded (LD←ED)	RRF_3	B3DD	TF
LDY	$R_1,D_2(X_2,B_2)$	Load (L)	RXY_2	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	
LEDTR	R_1,M_3,R_2,M_4	Load Rounded (SD←LD)	RRF_3	B3D5	TF
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_2	ED64	LD
LFAS	$D_2(B_2)$	Load FPC and Signal	S	B2BD	XF
LFPC	$D_2(B_2)$	Load FPC	S	B29D	
LG	$R_1,D_2(X_2,B_2)$	Load (64)	RXY_2	E304	N
LGB	$R_1,D_2(X_2,B_2)$	Load Byte (64←8)	RXY_2	E377	LD
LGBR	R_1,R_2	Load Byte (64←8)	RRE	B906	EI
LGDR	R_1,R_2	Load GR from FPR (64←L)	RRE	B3CD	FG
LGF	$R_1,D_2(X_2,B_2)$	Load (64←32)	RXY_2	E314	N
LGFI	R_1,I_2	Load Immediate (64←32)	RIL	C01	EI
LGFR	R_1,R_2	Load (64←32)	RRE	B914	N
LGFRL	R_1,I_2	Load Relative Long (64←32)	RIL	C4C	GE
LGH	$R_1,D_2(X_2,B_2)$	Load Halfword (64←16)	RXY_2	E315	N
LGHI	R_1,I_2	Load Halfword Immediate (64←16)	RI_1	A79	N
LGHR	R_1,R_2	Load Halfword (64←16)	RRE	B907	EI
LGHRL	R_1,I_2	Load Halfword Relative Long (64←16)	RIL	C44	GE
LGR	R_1,R_2	Load (64)	RRE	B904	N
LGRL	R_1,I_2	Load Relative Long (64)	RIL	C48	GE
LH	$R_1,D_2(X_2,B_2)$	Load Halfword (32←16)	RX	48	
LHI	R_1,I_2	Load Halfword Immediate (32←16)	RI_1	A78	
LHR	R_1,R_2	Load Halfword (32←16)	RRE	B927	EI
LHRL	R_1,I_2	Load Halfword Relative Long (32←16)	RIL	C45	GE
LHY	$R_1,D_2(X_2,B_2)$	Load Halfword (32←16)	RXY_2	E378	LD
LLC	$R_1,D_2(X_2,B_2)$	Load Logical Character (32←8)	RXY_2	E394	EI
LLCR	R_1,R_2	Load Logical Character (32←8)	RRE	B994	EI
LLGC	$R_1,D_2(X_2,B_2)$	Load Logical Character (64←8)	RXY_2	E390	N
LLGCR	R_1,R_2	Load Logical Character (64←8)	RRE	B984	EI
LLGF	$R_1,D_2(X_2,B_2)$	Load Logical (64←32)	RXY_2	E316	N
LLGFR	R_1,R_2	Load Logical (64←32)	RRE	B916	N
LLGFRL	R_1,I_2	Load Logical Relative Long (64←32)	RIL	C4E	GE
LLGH	$R_1,D_2(X_2,B_2)$	Load Logical Halfword (64←16)	RXY_2	E391	N
LLGHR	R_1,R_2	Load Logical Halfword (64←16)	RRE	B985	EI
LLGHRL	R_1,I_2	Load Logical Halfword Relative Long (64←16)	RIL	C46	GE
LLGT	$R_1,D_2(X_2,B_2)$	Load Logical Thirty One Bits (64←31)	RXY_2	E317	N
LLGTR	R_1,R_2	Load Logical Thirty One Bits (64←31)	RRE	B917	N
LLH	$R_1,D_2(X_2,B_2)$	Load Logical Halfword (32←16)	RXY_2	E395	EI
LLHR	R_1,R_2	Load Logical Halfword (32←16)	RRE	B995	EI
LLHRL	R_1,I_2	Load Logical Halfword Relative Long (32←16)	RIL	C42	GE
LLIHF	R_1,I_2	Load Logical Immediate (high)	RIL	C0E	EI
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
LLILF	R_1,I_2	Load Logical Immediate (low)	RIL	C0F	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

Mne-monic	Operands	Name	For-mat	Op-code	Class & Notes
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
LNDFR	R_1,R_2	Load Negative (L)	RRE	B371	FS
LNDR	R_1,R_2	Load Negative (LH)	RR	21	c
LNEBR	R_1,R_2	Load Negative (SB)	RRE	B301	c
LNER	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
LNXBR	R_1,R_2	Load Negative (EB)	RRE	B341	c
LNXR	R_1,R_2	Load Negative (EH)	RRE	B361	c
LPDBR	R_1,R_2	Load Positive (LB)	RRE	B310	c
LPDFR	R_1,R_2	Load Positive (L)	RRE	B370	FS
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
LPGR	R_1,R_2	Load Positive (64)	RRE	B900	c N
LPQ	$R_1,D_2(X_2,B_2)$	Load Pair from Quadword	RXY_2	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
LPTEA	R_1,R_3,R_2,M_4	Load Page-Table-Entry Address	RRF_3	B9AA	c D2
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_2	E303	pc N
LRAY	$R_1,D_2(X_2,B_2)$	Load Real Address (32)	RXY_2	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	
LRL	R_1,I_2	Load Relative Long (32)	RIL	C4D	GE
LRV	$R_1,D_2(X_2,B_2)$	Load Reversed (32)	RXY_2	E31E	N3
LRVG	$R_1,D_2(X_2,B_2)$	Load Reversed (64)	RXY_2	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_2	E31F	N3
LRVR	R_1,R_2	Load Reversed (32)	RRE	B91F	N3
LT	$R_1,D_2(X_2,B_2)$	Load and Test (32)	RXY_2	E312	c EI
LTDBR	R_1,R_2	Load and Test (LB)	RRE	B312	c
LTDR	R_1,R_2	Load and Test (LH)	RR	22	c
LTDTR	R_1,R_2	Load and Test (LD)	RRE	B3D6	c TF
LTEBR	R_1,R_2	Load and Test (SB)	RRE	B302	c
LTER	R_1,R_2	Load and Test (SH)	RR	32	c
LTG	$R_1,D_2(X_2,B_2)$	Load and Test (64)	RXY_2	E302	c EI
LTGF	$R_1,D_2(X_2,B_2)$	Load And Test (64←32)	RXY_2	E332	c GE
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
LTXTR	R_1,R_2	Load and Test (ED)	RRE	B3DE	c TF
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	p N
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	
LXDTR	R_1,R_2,M_4	Load Lengthened (ED←LD)	RRF_3	B3DC	TF
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	

Mnemonic	Operands	Name	Format	Op-code	Class & Notes
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$_2$	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
MAY	$R_1,R_3,D_2(X_2,B_2)$	Multiply and Add Unnormalized (EH←LH)	RXF	ED3A	UE
MAYH	$R_1,R_3,D_2(X_2,B_2)$	Multiply and Add Unnormalized (EH_H←LH)	RXF	ED3C	UE
MAYHR	R_1,R_3,R_2	Multiply and Add Unnormalized (EH_H←LH)	RRF$_1$	B33C	UE
MAYL	$R_1,R_3,D_2(X_2,B_2)$	Multiply and Add Unnormalized (EH_L←LH)	RXF	ED38	UE
MAYLR	R_1,R_3,R_2	Multiply and Add Unnormalized (EH_L←LH)	RRF$_1$	B338	UE
MAYR	R_1,R_3,R_2	Multiply and Add Unnormalized (EH←LH)	RRF$_1$	B33A	UE
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	
MDTR	R_1,R_2,R_3	Multiply (LD)	RRR	B3D0	TF
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	
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	
MFY	$R_1,D_2(X_2,B_2)$	Multiply	RXY$_2$	E35C	GE
MGHI	R_1,I_2	Multiply Halfword Immediate (64←16)	RI$_1$	A7D	N
MH	$R_1,D_2(X_2,B_2)$	Multiply Halfword (32←16)	RX	4C	
MHI	R_1,I_2	Multiply Halfword Immediate (32←16)	RI$_1$	A7C	
MHY	$R_1,D_2(X_2,B_2)$	Multiply Halfword	RXY$_2$	E37C	GE
ML	$R_1,D_2(X_2,B_2)$	Multiply Logical (64←32)	RXY$_2$	E396	N3
MLG	$R_1,D_2(X_2,B_2)$	Multiply Logical (128←64)	RXY$_2$	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	
MSER	R_1,R_3,R_2	Multiply and Subtract (SH)	RRF$_1$	B32F	HM
MSFI	R_1,I_2	Multiply Single Immediate	RIL	C21	GE

Mne-monic	Operands	Name	For-mat	Op-code	Class & Notes
MSG	$R_1,D_2(X_2,B_2)$	Multiply Single (64)	RXY_2	E30C	N
MSGF	$R_1,D_2(X_2,B_2)$	Multiply Single (64←32)	RXY_2	E31C	N
MSGFI	R_1,I_2	Multiply Single Immediate	RIL	C20	GE
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_2	E351	LD
MVC	$D_1(L,B_1),D_2(B_2)$	Move (character)	SS_1	D2	
MVCDK	$D_1(B_1),D_2(B_2)$	Move with Destination key	SSE	E50F	q
MVCIN	$D_1(L,B_1),D_2(B_2)$	Move Inverse	SS_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
MVCLE	$R_1,R_3,D_2(B_2)$	Move Long Extended	RS_1	A8	c
MVCLU	$R_1,R_3,D_2(B_2)$	Move Long Unicode	RSY_1	EB8E	c E2
MVCOS	$D_1(B_1),D_2(B_2),R_3$	Move with Optional Specifications	SSF	C80	c q MO
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
MVGHI	$D_1(B_1),I_2$	Move (64←16)	SIL	E548	GE
MVHHI	$D_1(B_1),I_2$	Move (16←16)	SIL	E544	GE
MVHI	$D_1(B_1),I_2$	Move (32←16)	SIL	E54C	GE
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	
MXTR	R_1,R_2,R_3	Multiply (ED)	RRR	B3D8	TF
MY	$R_1,R_3,D_2(X_2,B_2)$	Multiply Unnormalized (EH←LH)	RXF	ED3B	UE
MYH	$R_1,R_3,D_2(X_2,B_2)$	Multiply Unnormalized (EH_H←LH)	RXF	ED3D	UE
MYHR	R_1,R_3,R_2	Multiply Unnormalized (EH_H←LH)	RRF_1	B33D	UE
MYL	$R_1,R_3,D_2(X_2,B_2)$	Multiply Unnormalized (EH_L←LH)	RXF	ED39	UE
MYLR	R_1,R_3,R_2	Multiply Unnormalized (EH_L←LH)	RRF_1	B339	UE
MYR	R_1,R_3,R_2	Multiply Unnormalized (EH←LH)	RRF_1	B33B	UE
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_2	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
NIHF	R_1,I_2	And Immediate (high)	RIL	C0A	c EI
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
NILF	R_1,I_2	And Immediate (low)	RIL	C0B	c EI
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_2	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

Mnemonic	Operands	Name	Format	Op-code	Class & Notes
OG	$R_1,D_2(X_2,B_2)$	Or (64)	RXY$_2$	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
OIHF	R_1,I_2	Or Immediate (high)	RIL	C0C	c EI
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
OILF	R_1,I_2	Or Immediate (low)	RIL	C0D	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_1R_2	Or (32)	RR	16	c
OY	$R_1,D_2(X_2,B_2)$	Or (32)	RXY$_2$	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
PFD	$M_1,D_2(X_2,B_2)$	Prefetch Data	RXY$_1$	E336	GE
PFDRL	M_1,I_2	Prefetch Data Relative Long	RIL	C62	GE
PFMF	R_1,R_2	Perform Frame Management Function	RRE	B9AF	p ED
PFPO		Perform Floating-Point Operation	E	010A	PF
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
PTF	R_1	Perform Topology Function	RRE	B9A2	c p CT
PTFF		Perform Timing-Facility Function	E	0104	qc
PTI	R_1,R_2	Program Transfer with Instance	RRE	B99E	q RA
PTLB		Purge TLB	S	B20D	p
QADTR	R_1,R_3,R_2,M_4	Quantize (LD)	RRF$_3$	B3F5	TF
QAXTR	R_1,R_3,R_2,M_4	Quantize (ED)	RRF$_3$	B3FD	TF
RCHP		Reset Channel Path	S	B23B	pc
RISBG	$R_1,R_2,I_3,I_4[,I_5]$	Rotate Then Insert Selected Bits	RIE$_5$	EC55	c GE
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
RNSBG	$R_1,R_2,I_3,I_4[,I_5]$	Rotate Then AND Selected Bits	RIE$_5$	EC54	c GE
ROSBG	$R_1,R_2,I_3,I_4[,I_5]$	Rotate Then ORSelected Bits	RIE$_5$	EC56	c GE
RP	$D_2(B_2)$	Resume Program	S	B277	qn
RRBE	R_1,R_2	Reset Reference Bit Extended	RRE	B22A	pc
RRDTR	R_1,R_3,R_2,M_4	Reround (LD)	RRF$_3$	B3F7	TF
RRXTR	R_1,R_3,R_2,M_4	Reround (ED)	RRF$_3$	B3FF	TF
RSCH		Resume Subchannel	S	B238	pc
RXSBG	$R_1,R_2,I_3,I_4[,I_5]$	Rotate Then EXCLUSIVE OR Selected Bits	RIE$_5$	EC57	c GE
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
SDTR	R_1,R_2,R_3	Subtract (LD)	RRR	B3D3	c TF

Mnemonic	Operands	Name	Format	Op-code	Class & Notes
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
SFASR	R_1	Set FPC and Signal	RRE	B385	XF
SFPC	R_1	Set FPC	RRE	B384	
SG	$R_1,D_2(X_2,B_2)$	Subtract (64)	RXY$_2$	E309	c N
SGF	$R_1,D_2(X_2,B_2)$	Subtract (64←32)	RXY$_2$	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 (32←16)	RX	4B	c
SHY	$R_1,D_2(X_2,B_2)$	Subtract Halfword (32←16)	RXY$_2$	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$_2$	E399	c N3
SLBG	$R_1,D_2(X_2,B_2)$	Subtract Logical with Borrow (64)	RXY$_2$	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 (64)	RS$_1$	8F	c
SLDL	$R_1,D_2(B_2)$	Shift Left Double Logical (64)	RS$_1$	8D	
SLDT	$R_1,R_3,D_2(X_2,B_2)$	Shift Significand Left (LD)	RXF	ED40	TF
SLFI	R_1,I_2	Subtract Logical Immediate (32)	RIL	C25	c EI
SLG	$R_1,D_2(X_2,B_2)$	Subtract Logical (64)	RXY$_2$	E30B	c N
SLGF	$R_1,D_2(X_2,B_2)$	Subtract Logical (64←32)	RXY$_2$	E31B	c N
SLGFI	R_1,I_2	Subtract Logical Immediate (64←32)	RIL	C24	c EI
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
SLXT	$R_1,R_3,D_2(X_2,B_2)$	Shift Significand Left (ED)	RXF	ED48	TF
SLY	$R_1,D_2(X_2,B_2)$	Subtract Logical (32)	RXY$_2$	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 (64)	RS$_1$	8E	c
SRDL	$R_1,D_2(B_2)$	Shift Right Double Logical (64)	RS$_1$	8C	
SRDT	$R_1,R_3,D_2(X_2,B_2)$	Shift Significand Right (LD)	RXF	ED41	TF
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 BFP Rounding Mode	S	B299	
SRNMT	$D_2(B_2)$	Set DFP Rounding Mode	S	B2B9	TR

Mnemonic	Operands	Name	Format	Opcode	Class & Notes
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
SRSTU	R_1,R_2	Search String Unicode	RRE	B2BE	c E3
SRXT	$R_1,R_3,D_2(X_2,B_2)$	Shift Significand Right (ED)	RXF	ED49	TF
SSAIR	R_1	Set Secondary ASN with Instance	RRE	B99F	RA
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	RRF	B22B	pc
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	
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
STCKF	$D_2(B_2)$	Store Clock Fast	S	B27C	c SC
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$_2$	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$_2$	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$_2$	ED66	LD
STFL	$D_2(B_2)$	Store Facility List	S	B2B1	p N3
STFLE	$D_2(B_2)$	Store Facility List Extended	S	B2B0	c FL
STFPC	$D_2(B_2)$	Store FPC	S	B29C	
STG	$R_1,D_2(X_2,B_2)$	Store (64)	RXY$_2$	E324	N
STGRL	R_1,I_2	Store Relative Long (64)	RIL	C4B	GE
STH	$R_1,D_2(X_2,B_2)$	Store Halfword	RX	40	
STHRL	R_1,I_2	Store Halfword Relative Long	RIL	C47	GE
STHY	$R_1,D_2(X_2,B_2)$	Store Halfword	RXY$_2$	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$_2$	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
STRL	R_1,I_2	Store Relative Long (32)	RIL	C4F	GE
STRV	$R_1,D_2(X_2,B_2)$	Store Reversed (32)	RXY$_2$	E33E	N3
STRVG	$R_1,D_2(X_2,B_2)$	Store Reversed (64)	RXY$_2$	E32F	N
STRVH	$R_1,D_2(X_2,B_2)$	Store Reversed (16)	RXY$_2$	E33F	N3
STSCH	$D_2(B_2)$	Store Subchannel	S	B234	pc
STSI	$D_2(B_2)$	Store System Information	S	B27D	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$_2$	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

Mnemonic	Operands	Name	Format	Op-code	Class & Notes
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
SXTR	R_1,R_2,R_3	Subtract (ED)	RRR	B3DB	c TF
SY	$R_1,D_2(X_2,B_2)$	Subtract (32)	RXY$_2$	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
TDCDT	$R_1,D_2(X_2,B_2)$	Test Data Class (LD)	RXE	ED54	TF
TDCET	$R_1,D_2(X_2,B_2)$	Test Data Class (SD)	RXE	ED50	TF
TDCXT	$R_1,D_2(X_2,B_2)$	Test Data Class (ED)	RXE	ED58	TF
TDGDT	$R_1,D_2(X_2,B_2)$	Test Data Group (LD)	RXE	ED55	TF
TDGET	$R_1,D_2(X_2,B_2)$	Test Data Group (SD)	RXE	ED51	TF
TDGXT	$R_1,D_2(X_2,B_2)$	Test Data Group (ED)	RXE	ED59	TF
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	EBC0	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	$D_1(L,B_1),D_2(B_2)$	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	$R_1,R_2[,M_3]$	Translate One to One	RRF$_2$	B993	c E2
TROT	$R_1,R_2[,M_3]$	Translate One to Two	RRF$_2$	B992	c E2
TRT	$D_1(L,B_1),D_2(B_2)$	Translate and Test	SS$_1$	DD	c
TRTE	$R_1,R_2[,M_3]$	Translate and Test Extended	RRF	B9BF	PE
TRTO	$R_1,R_2[,M_3]$	Translate Two to One	RRF$_2$	B991	c E2
TRTR	$D_1(L,B_1),D_2(B_2)$	Translate and Test Reverse	SS$_1$	D0	c E3
TRTRE	$R_1,R_2[,M_3]$	Translate and Test Reverse Extended	RRF	B9BD	PE
TRTT	$R_1,R_2[,M_3]$	Translate Two to Two	RRF$_2$	B990	c E2
TS	$D_2(B_2)$	Test and Set	S	93	c
TSCH	$D_2(B_2)$	Test Subchannel	S	B235	pc
UNPK	$D_1(L_1,B_1),D_2(L_2,B_2)$	Unpack	SS$_2$	F3	
UNPKA	$D_1(L_1,B_1),D_2(B_2)$	Unpack ASCII	SS$_1$	EA	c E2
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	$D_1(L,B_1),D_2(B_2)$	Exclusive Or (character)	SS$_1$	D7	c
XG	$R_1,D_2(X_2,B_2)$	Exclusive Or (64)	RXY$_2$	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
XIHF	R_1,I_2	Exclusive Or Immediate (high)	RIL	C06	c N

Mnemonic	Operands	Name	Format	Op-code	Class & Notes
XILF	R_1,I_2	Exclusive Or Immediate (low)	RIL	C07	c N
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_2	E357	c LD
ZAP	$D_1(L_1,B_1),D_2(L_2,B_2)$	Zero and Add	SS_2	F8	c

Floating-Point Operand Lengths and Types:

E	Extended (binary, decimal or hex)
EB	Extended binary
ED	Extended decimal
EH	Extended hex
EHL	Extended hex (low-order part)
EHH	Extended hex (high-order part)
L	Long (binary, decimal or hex)
LB	Long binary
LD	Long decimal
LH	Long hex
S	Short (binary, decimal or hex)
SB	Short binary
SD	Short decimal
SH	Short hex

Notes:

c	Condition code set
i	Interruptible instruction
n	New condition code loaded
p	Privileged instruction
q	Semiprivileged instruction
u	Condition code is unpredictable
CS	Compare-and-swap-and-store facility
CT	Configuration topology facility
DE	DAT-enhancement facility
D2	DAT-enhancement facility 2
ED	Enhanced-DAT facility
EI	Extended-immediate facility
E2	Extended-translation facility 2
E3	Extended-translation facility 3
ES	Expanded-storage facility
ET	Extract-CPU-time facility
FG	FPR-GPR-transfer facility
FL	Store-facility-list-extended facility
FS	Floating-point-support-sign-handling facility
GE	General-instructions-extension facility
HM	HFP multiply-and-add/subtract facility
LD	Long-displacement facility
N	New in z/Architecture
MO	Move-with-optional-specifications facility
MS	Message-security assist
N3	New in z/Architecture and added to ESA/390
PE	Parsing-enhancement facility
PF	PFPO facility
RA	ASN-and-LX-reuse facility
SC	Store-clock-fast facility
TF	Decimal-floating-point facility
TR	Decimal-floating-point-rounding facility
UE	HFP unnormalized-extension facility
XF	IEEE-exception-support facility
XX	Execute-extension facility

Machine Instructions by Operation Code

OpCode	Mnemonic
0101	PR
0102	UPT
0104	PTFF
0107	SCKPF
010A	PFPO
010B	TAM
010C	SAM24
010D	SAM31
010E	SAM64
01FF	TRAP2
04	SPM
05	BALR
06	BCTR
07	BCR
0A	SVC
0B	BSM
0C	BASSM
0D	BASR
0E	MVCL
0F	CLCL
10	LPR
11	LNR
12	LTR
13	LCR
14	NR
15	CLR
16	OR
17	XR
18	LR
19	CR
1A	AR
1B	SR
1C	MR
1D	DR
1E	ALR
1F	SLR
20	LPDR
21	LNDR
22	LTDR
23	LCDR
24	HDR
25	LDXR
25	LRDR
26	MXR
27	MXDR
28	LDR
29	CDR
2A	ADR
2B	SDR
2C	MDR
2D	DDR
2E	AWR
2F	SWR
30	LPER
31	LNER
32	LTER
33	LCER
34	HER
35	LEDR
35	LRER
36	AXR
37	SXR
38	LER
39	CER
3A	AER
3B	SER
3C	MDER
3C	MER
3D	DER
3E	AUR
3F	SUR
40	STH

OpCode	Mnemonic
41	LA
42	STC
43	IC
44	EX
45	BAL
46	BCT
47	BC
48	LH
49	CH
4A	AH
4B	SH
4C	MH
4D	BAS
4E	CVD
4F	CVB
50	ST
51	LAE
54	N
55	CL
56	O
57	X
58	L
59	C
5A	A
5B	S
5C	M
5D	D
5E	AL
5F	SL
60	STD
67	MXD
68	LD
69	CD
6A	AD
6B	SD
6C	MD
6D	DD
6E	AW
6F	SW
70	STE
71	MS
78	LE
79	CE
7A	AE
7B	SE
7C	MDE
7C	ME
7D	DE
7E	AU
7F	SU
80	SSM
82	LPSW
83	Diagnose
84	BRXH
85	BRXLE
86	BXH
87	BXLE
88	SRL
89	SLL
8A	SRA
8B	SLA
8C	SRDL
8D	SLDL
8E	SRDA
8F	SLDA
90	STM
91	TM
92	MVI
93	TS
94	NI
95	CLI
96	OI

OpCode	Mnemonic
97	XI
98	LM
99	TRACE
9A	LAM
9B	STAM
A50	IIHH
A51	IIHL
A52	IILH
A53	IILL
A54	NIHH
A55	NIHL
A56	NILH
A57	NILL
A58	OIHH
A59	OIHL
A5A	OILH
A5B	OILL
A5C	LLIHH
A5D	LLIHL
A5E	LLILH
A5F	LLILL
A70	TMLH
A70	TMH
A71	TMLL
A71	TML
A72	TMHH
A73	TMHL
A74	BRC
A75	BRAS
A76	BRCT
A77	BRCTG
A78	LHI
A79	LGHI
A7A	AHI
A7B	AGHI
A7C	MHI
A7D	MGHI
A7E	CHI
A7F	CGHI
A8	MVCLE
A9	CLCLE
AC	STNSM
AD	STOSM
AE	SIGP
AF	MC
B1	LRA
B202	STIDP
B204	SCK
B205	STCK
B206	SCKC
B207	STCKC
B208	SPT
B209	STPT
B20A	SPKA
B20B	IPK
B20D	PTLB
B210	SPX
B211	STPX
B212	STAP
B214	SIE
B218	PC
B219	SAC
B21A	CFC
B221	IPTE
B222	IPM
B223	IVSK
B224	IAC
B225	SSAR
B226	EPAR
B227	ESAR
B228	PT
B229	ISKE

OpCode	Mnemonic
B22A	RRBE
B22B	SSKE
B22C	TB
B22D	DXR
B22E	PGIN
B22F	PGOUT
B230	CSCH
B231	HSCH
B232	MSCH
B233	SSCH
B234	STSCH
B235	TSCH
B236	TPI
B237	SAL
B238	RSCH
B239	STCRW
B23A	STCPS
B23B	RCHP
B23C	SCHM
B240	BAKR
B241	CKSM
B244	SQDR
B245	SQER
B246	STURA
B247	MSTA
B248	PALB
B249	EREG
B24A	ESTA
B24B	LURA
B24C	TAR
B24D	CPYA
B24E	SAR
B24F	EAR
B250	CSP
B252	MSR
B254	MVPG
B255	MVST
B257	CUSE
B258	BSG
B25A	BSA
B25D	CLST
B25E	SRST
B263	CMPSC
B276	XSCH
B277	RP
B278	STCKE
B279	SACF
B27C	STCKF
B27D	STSI
B299	SRNM
B29C	STFPC
B29D	LFPC
B2A5	TRE
B2A6	CU21
B2A6	CUUTF
B2A7	CU12
B2A7	CUTFU
B2B0	STFLE
B2B1	STFL
B2B2	LPSWE
B2B9	SRNMT
B2BD	LFAS
B2FF	TRAP4
B300	LPEBR
B301	LNEBR
B302	LTEBR
B303	LCEBR
B304	LDEBR
B305	LXDBR
B306	LXEBR
B307	MXDBR
B308	KEBR
B309	CEBR
B30A	AEBR
B30B	SEBR
B30C	MDEBR

OpCode	Mnemonic
B30D	DEBR
B30E	MAEBR
B30F	MSEBR
B310	LPDBR
B311	LNDBR
B312	LTDBR
B313	LCDBR
B314	SQEBR
B315	SQDBR
B316	SQXBR
B317	MEEBR
B318	KDBR
B319	CDBR
B31A	ADBR
B31B	SDBR
B31C	MDBR
B31D	DDBR
B31E	MADBR
B31F	MSDBR
B324	LDER
B325	LXDR
B326	LXER
B32E	MAER
B32F	MSER
B336	SQXR
B337	MEER
B338	MAYLR
B339	MYLR
B33A	MAYR
B33B	MYR
B33C	MAYHR
B33D	MYHR
B33E	MADR
B33F	MSDR
B340	LPXBR
B341	LNXBR
B342	LTXBR
B343	LCXBR
B344	LEDBR
B345	LDXBR
B346	LEXBR
B347	FIXBR
B348	KXBR
B349	CXBR
B34A	AXBR
B34B	SXBR
B34C	MXBR
B34D	DXBR
B350	TBEDR
B351	TBDR
B353	DIEBR
B357	FIEBR
B358	THDER
B359	THDR
B35B	DIDBR
B35F	FIDBR
B360	LPXR
B361	LNXR
B362	LTXR
B363	LCXR
B365	LXR
B366	LEXR
B367	FIXR
B369	CXR
B370	LPDFR
B371	LNDFR
B372	CPSDR
B373	LCDFR
B374	LZER
B375	LZDR
B376	LZXR
B377	FIER
B37F	FIDR
B384	SFPC
B385	SFASR
B38C	EFPC

OpCode	Mnemonic
B394	CEFBR
B395	CDFBR
B396	CXFBR
B398	CFEBR
B399	CFDBR
B39A	CFXBR
B3A4	CEGBR
B3A5	CDGBR
B3A6	CXGBR
B3A8	CGEBR
B3A9	CGDBR
B3AA	CGXBR
B3B4	CEFR
B3B5	CDFR
B3B6	CXFR
B3B8	CFER
B3B9	CFDR
B3BA	CFXR
B3C1	LDGR
B3C4	CEGR
B3C5	CDGR
B3C6	CXGR
B3C8	CGER
B3C9	CGDR
B3CA	CGXR
B3CD	LGDR
B3D0	MDTR
B3D1	DDTR
B3D2	ADTR
B3D3	SDTR
B3D4	LDETR
B3D5	LEDTR
B3D6	LTDTR
B3D7	FIDTR
B3D8	MXTR
B3D9	DXTR
B3DA	AXTR
B3DB	SXTR
B3DC	LXDTR
B3DD	LDXTR
B3DE	LTXTR
B3DF	FIXTR
B3E0	KDTR
B3E1	CGDTR
B3E2	CUDTR
B3E3	CSDTR
B3E4	CDTR
B3E5	EEDTR
B3E7	ESDTR
B3E8	KXTR
B3E9	CGXTR
B3EA	CUXTR
B3EB	CSXTR
B3EC	CXTR
B3ED	EEXTR
B3EF	ESXTR
B3F1	CDGTR
B3F2	CDUTR
B3F3	CDSTR
B3F4	CEDTR
B3F5	QADTR
B3F6	IEDTR
B3F7	RRDTR
B3F9	CXGTR
B3FA	CXUTR
B3FB	CXSTR
B3FC	CEXTR
B3FD	QAXTR
B3FE	IEXTR
B3FF	RRXTR
B6	STCTL
B7	LCTL
B900	LPGR
B901	LNGR
B902	LTGR
B903	LCGR

OpCode	Mnemonic
B904	LGR
B905	LURAG
B906	LGBR
B907	LGHR
B908	AGR
B909	SGR
B90A	ALGR
B90B	SLGR
B90C	MSGR
B90D	DSGR
B90E	EREGG
B90F	LRVGR
B910	LPGFR
B911	LNGFR
B912	LTGFR
B913	LCGFR
B914	LGFR
B916	LLGFR
B917	LLGTR
B918	AGFR
B919	SGFR
B91A	ALGFR
B91B	SLGFR
B91C	MSGFR
B91D	DSGFR
B91E	KMAC
B91F	LRVR
B920	CGR
B921	CLGR
B925	STURG
B926	LBR
B927	LHR
B92E	KM
B92F	KMC
B930	CGFR
B931	CLGFR
B93E	KIMD
B93F	KLMD
B946	BCTGR
B960	CGRT
B961	CLGRT
B972	CRT
B973	CLRT
B980	NGR
B981	OGR
B982	XGR
B983	FLOGR
B984	LLGCR
B985	LLGHR
B986	MLGR
B987	DLGR
B988	ALCGR
B989	SLBGR
B98A	CSPG
B98D	EPSW
B98E	IDTE
B990	TRTT
B991	TRTO
B992	TROT
B993	TROO
B994	LLCR
B995	LLHR
B996	MLR
B997	DLR
B998	ALCR
B999	SLBR
B99A	EPAIR
B99B	ESAIR
B99D	ESEA
B99E	PTI
B99F	SSAIR
B9A2	PTF
B9AA	LPTEA
B9AF	PFMF
B9B0	CU14
B9B1	CU24

OpCode	Mnemonic
B9B2	CU41
B9B3	CU42
B9BD	TRTRE
B9BE	SRSTU
B9BF	TRTE
BA	CS
BB	CDS
BD	CLM
BE	STCM
BF	ICM
C00	LARL
C01	LGFI
C04	BRCL
C05	BRASL
C06	XIHF
C07	XILF
C08	IIHF
C09	IILF
C0A	NIHF
C0B	NILF
C0C	OIHF
C0D	OILF
C0E	LLIHF
C0F	LLILF
C20	MSGFI
C21	MSFI
C24	SLGFI
C25	SLFI
C28	AGFI
C29	AFI
C2A	ALGFI
C2B	ALFI
C2C	CGFI
C2D	CFI
C2E	CLGFI
C2F	CLFI
C42	LLHRL
C44	LGHRL
C45	LHRL
C46	LLGHRL
C47	STHRL
C48	LGRL
C4B	STGRL
C4C	LGFRL
C4D	LRL
C4E	LLGFRL
C4F	STRL
C60	EXRL
C62	PFDRL
C64	CGHRL
C65	CHRL
C66	CLGHRL
C67	CLHRL
C68	CGRL
C6A	CLGRL
C6C	CGFRL
C6D	CRL
C6E	CLGFRL
C6F	CLRL
C80	MVCOS
C81	ECTG
C82	CSST
D0	TRTR
D1	MVN
D2	MVC
D3	MVZ
D4	NC
D5	CLC
D6	OC
D7	XC
D9	MVCK
DA	MVCP
DB	MVCS
DC	TR
DD	TRT
DE	ED

OpCode	Mnemonic
DF	EDMK
E1	PKU
E2	UNPKU
E302	LTG
E303	LRAG
E304	LG
E306	CVBY
E308	AG
E309	SG
E30A	ALG
E30B	SLG
E30C	MSG
E30D	DSG
E30E	CVBG
E30F	LRVG
E312	LT
E313	LRAY
E314	LGF
E315	LGH
E316	LLGF
E317	LLGT
E318	AGF
E319	SGF
E31A	ALGF
E31B	SLGF
E31C	MSGF
E31D	DSGF
E31E	LRV
E31F	LRVH
E320	CG
E321	CLG
E324	STG
E326	CVDY
E32E	CVDG
E32F	STRVG
E330	CGF
E331	CLGF
E332	LTGF
E334	CGH
E336	PFD
E33E	STRV
E33F	STRVH
E346	BCTG
E350	STY
E351	MSY
E354	NY
E355	CLY
E356	OY
E357	XY
E358	LY
E359	CY
E35A	AY
E35B	SY
E35C	MFY
E35E	ALY
E35F	SLY
E370	STHY
E371	LAY
E372	STCY
E373	ICY
E375	LAEY
E376	LB
E377	LGB
E378	LHY
E379	CHY
E37A	AHY
E37B	SHY
E37C	MHY
E380	NG
E381	OG
E382	XG
E386	MLG
E387	DLG
E388	ALCG
E389	SLBG
E38E	STPQ

OpCode	Mnemonic
E38F	LPQ
E390	LLGC
E391	LLGH
E394	LLC
E395	LLH
E396	ML
E397	DL
E398	ALC
E399	SLB
E500	LASP
E501	TPROT
E502	STRAG
E50E	MVCSK
E50F	MVCDK
E544	MVHHI
E548	MVGHI
E54C	MVHI
E554	CHHSI
E555	CLHHSI
E558	CGHSI
E559	CLGHSI
E55C	CHSI
E55D	CLFHSI
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	STCMH
EB2D	STCMY
EB2F	LCTLG
EB30	CSG
EB31	CDSY
EB3E	CDSG
EB44	BXHG
EB45	BXLEG
EB4C	ECAG
EB51	TMY
EB52	MVIY
EB54	NIY
EB55	CLIY
EB56	OIY
EB57	XIY
EB6A	ASI
EB6E	ALSI
EB7A	AGSI
EB7E	ALGSI
EB80	ICMH
EB81	ICMY
EB8E	MVCLU
EB8F	CLCLU
EB90	STMY
EB96	LMH
EB98	LMY
EB9A	LAMY
EB9B	STAMY
EBC0	TP
EC44	BRXHG
EC45	BRXLG
EC54	RNSBG
EC55	RISBG
EC56	ROSBG
EC57	RXSBG
EC64	CGRJ

OpCode	Mnemonic
EC65	CLGRJ
EC70	CGIT
EC71	CLGIT
EC72	CIT
EC73	CLFIT
EC76	CRJ
EC77	CLRJ
EC7C	CGIJ
EC7D	CLGIJ
EC7E	CIJ
EC7F	CLIJ
ECE4	CGRB
ECE5	CLGRB
ECF6	CRB
ECF7	CLRB
ECFC	CGIB
ECFD	CLGIB
ECFE	CIB
ECFF	CLIB
ED04	LDEB
ED05	LXDB
ED06	LXEB
ED07	MXDB
ED08	KEB
ED09	CEB
ED0A	AEB
ED0B	SEB
ED0C	MDEB
ED0D	DEB
ED0E	MAEB
ED0F	MSEB
ED10	TCEB
ED11	TCDB
ED12	TCXB
ED14	SQEB
ED15	SQDB
ED17	MEEB
ED18	KDB
ED19	CDB
ED1A	ADB
ED1B	SDB
ED1C	MDB
ED1D	DDB
ED1E	MADB
ED1F	MSDB
ED24	LDE
ED25	LXD
ED26	LXE
ED2E	MAE
ED2F	MSE
ED34	SQE
ED35	SQD
ED37	MEE
ED38	MAYL
ED39	MYL
ED3A	MAY
ED3B	MY
ED3C	MAYH
ED3D	MYH
ED3E	MAD
ED3F	MSD
ED40	SLDT
ED41	SRDT
ED48	SLXT
ED49	SRXT
ED50	TDCET
ED51	TDGET
ED54	TDCDT
ED55	TDGDT
ED58	TDCXT
ED59	TDGXT
ED64	LEY
ED65	LDY
ED66	STEY
ED67	STDY
EE	PLO

OpCode	Mnemonic
EF	LMD
F0	SRP
F1	MVO
F2	PACK
F3	UNPK
F8	ZAP
F9	CP
FA	AP
FB	SP
FC	MP
FD	DP

Condition Codes

Condition Code →	**0**	**1**	**2**	**3**
Mask Bit Value →	**8**	**4**	**2**	**1**
General Instructions				
Add	Zero	< Zero	> Zero	Overflow
Add Halfword	Zero	< Zero	> Zero	Overflow
Add Halfword Immediate	Zero	< Zero	> Zero	Overflow
Add 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
Add Logical with Signed Immediate	Zero, no carry	Not zero, no carry	Zero, carry	Not zero, carry
And	Zero	Not zero	—	—
And Immediate	ANDed bits zero	ANDed 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 and Swap and Store	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 Halfword Relative Long	Equal	First op low	First op high	—
Compare 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 Relative Long	Equal	First op low	First op high	—
Compare Logical String	Equal	First op low	First op high	CPU-determined completion
Compare Relative Long	Equal	First op low	First op high	—
Compare until Substring Equal	Equal substring	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 Intermediate 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	—	—
Exclusive Or Immediate	XORed bits zero	XORed bits not zero	—	—
Find Leftmost One	No one bit found	—	One bit found	—

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
Load Page-Table-Entry Address	Address returned; STE.P=0	Address returned; STE.P=1	Invalid bit on in RTE or STE.	Exception condition exists.
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	ORed bits zero	ORed 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
Perform Timing-Facility Function	Function performed	—	—	Function not avail.
Rotate Then And Selected Bits	Selected bits zero	Selected bits not zero	—	—
Rotate Then Exclusive Or Selected Bits	Selected bits zero	Selected bits not zero	—	—
Rotate Then Insert Selected Bits	Zero	< zero	> zero	—
Rotate Then Or Selected Bits	Selected bits zero	Selected bits not zero	—	—
Search String, Search String Unicode	—	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 (STCK, STCKE or STCKF)	Set state	Not-set state	Error state	Stopped state or not operational
Store Facility List Extended	Complete list stored	—	—	Incomplete list stored
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 (TMH, TMHH, TMHL, TML, 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 Code →	**0**	**1**	**2**	**3**
Mask Bit Value →	**8**	**4**	**2**	**1**
Translate and Test, Translate and Test Reverse	All zeros	Not zero, scan incomplete	Not zero, scan complete	—
Translate and Test Extended, Translate and Test Reverse Extended	All selected function codes zero	Nonzero function code selected	—	CPU-determined completion
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
Compare Biased Exponent	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
Load Positive (HFP)	Zero	—	> Zero	—
Perform Floating-Point Operation (T=0)	Normal result	Nontrap exception	Trap exception	—
Perform Floating-Point Operation (T=1)	Function valid	—	—	Function invalid
Subtract	Zero	< Zero	> Zero	NaN
Subtract Normalized	Zero	< Zero	> Zero	—
Subtract Unnormalized	Zero	< Zero	> Zero	—
Test Data Class	Zero (no match)	One (match)	—	—
Test Data Group	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	Primaryspace mode	Secondaryspace mode	Accessregister mode	Homespace mode

Condition Code →	**0**	**1**	**2**	**3**
Mask Bit Value →	**8**	**4**	**2**	**1**
Load Address Space Parameters	Parameters loaded	Primary not available	Secondary not authorized or not available	Spaceswitch 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	Segmenttable entry invalid	Pagetable 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
Move with Optional Specifications	Length ≤ 4096	—	—	Length > 4096
Page In	Operation completed	ES data error	—	ES block not available
Page Out	Operation completed	ES data error	—	ES block not available
Perform Timing Facility Function	Function performed	—	—	Function not available
Perform Topology Function	Initiated	—	Rejected	—
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
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	—	—

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.

Assembler Instructions

Function	Mnemonic	Meaning
Option control	*PROCESS	Specify assembler options
	ACONTROL	Dynamically modify options
Data definition	CCW	Define channel command word
	CCW0	Define format-0 channel command word
	CCW1	Define format-1 channel command word
	DC	Define constant
	DS	Define storage
Program sectioning and linking	ALIAS	Rename external symbol
	AMODE	Specify addressing mode
	CATTR	Define class/part 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 listing	AEJECT	Start new page in macro definition
	ASPACE	Space lines in macro definition
	CEJECT	Conditional start new page
	EJECT	Start new page
	PRINT	Print optional data
	SPACE	Space listing
	TITLE	Identify assembly output
Program control	ADATA	Provide data for SYSADATA file
	CNOP	Conditional no operation
	COPY	Copy predefined source coding
	END	End assembly
	EQU	Equate symbol
	EXITCTL	Program control data for I/O exits
	ICTL	Input format control
	ISEQ	Input sequence checking
	LTORG	Begin literal pool
	OPSYN	Equate operation code
	ORG	Set location counter
	POP	Restore ACONTROL, PRINT, or USING status
	PUNCH	Punch a 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 message
	SETA	Set arithmetic variable symbol
	SETAF	Set arithmetic variable symbol from external function
	SETB	Set binary variable symbol
	SETC	Set character variable symbol
	SETCF	Set character variable symbol from external function

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

Source: SC26-4940.

Extended-Mnemonic Instructions for Branch on Condition

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

Source: SC26-4940.

* Second operand, not shown, is D_2 (X_2, B_2) for RX format and R_2 for RR format.

Extended-Mnemonic Instructions for Relative-Branch Instructions

Use	Extended Mnemonic	Meaning	Machine Instr.
General Branch Rel. on Condition	BRU or J	Unconditional Branch Relative	BRC 15,I_2
	BRUL or JLU	Unconditional Branch Relative	BRCL 15,I_2
	JNOP*	No Operation	BRC 0,I_2
After Compare Instructions	BRH or JH*	Branch Relative on A High	BRC 2,I_2
	BRL or JL*	Branch Relative on A Low	BRC 4,I_2
	BRE or JE*	Branch Relative on A Equal B	BRC 8,I_2
	BRNH or JNH*	Branch Relative on A Not High	BRC 13,I_2
	BRNL or JNL*	Branch Relative on A Not Low	BRC 11,I_2
	BRNE or JNE*	Branch Relative on A Not Equal B	BRC 7,I_2
After Arithmetic Instructions	BRP or JP*	Branch Relative on Plus	BRC 2,I_2
	BRM or JM*	Branch Relative on Minus	BRC 4,I_2
	BRZ or JZ*	Branch Relative on Zero	BRC 8,I_2
	BRO or JO*	Branch Relative on Overflow	BRC 1,I_2
	BRNP or JNP*	Branch Relative on Not Plus	BRC 13,I_2
	BRNM or JNM*	Branch Relative on Not Minus	BRC 11,I_2
	BRNZ or JNZ*	Branch Relative on Not Zero	BRC 7,I_2
	BRNO or JNO*	Branch Relative on No Overflow	BRC 14,I_2
After Test under Mask Instruction	BRO or JO*	Branch Relative if Ones	BRC 1,I_2
	BRM or JM*	Branch Relative if Mixed	BRC 4,I_2
	BRZ or JZ*	Branch Relative if Zeros	BRC 8,I_2
	BRNO or JNO*	Branch Relative if Not Ones	BRC 14,I_2
	BRNM or JNM*	Branch Relative if Not Mixed	BRC 11,I_2
	BRNZ or JNZ*	Branch Relative if Not Zeros	BRC 7,I_2
Other Branch Relative Instructions	JAS	Branch Relative and Save	BRAS R_1,I_2
	JASL	Branch Relative and Save Long	BRASL R_1,I_2
	JCT	Branch Relative on Count (32)	BRCT R_1,I_2
	JCTG	Branch Relative on Count (64)	BRCTG R_1,I_2

Use	Extended Mnemonic	Meaning	Machine Instr.
	JXH	Branch Relative on Index High (32)	BRXH R_1,R_3,I_2
	JXHG	Branch Relative on Index High (64)	BRXHG R_1,R_3,I_2
	JXLE	Br. Rel. on Index Low or Equal (32)	BRXLE R_1,R_3,I_2
	JXLEG	Br. Rel. on Index Low or Equal (64)	BRXLG R_1,R_3,I_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 JLNZ.

Extended-Mnemonic Suffixes for Compare-and-Branch and Compare-and-Trap Instructions

Suffix	Meaning	M_3 Value	Suffix	Meaning	M_3 Value
H	High	2	NH	Not High	13
L	Low	4	NL	Not Low	11
E	Equal	8	NE	Not Equal	7

Explanation:

These suffixes may be appended to the following mnemonics: CGIB, CGIJ, CGIT, CGRB, CGRJ, CGRT, CIB, CIJ, CIT, CLFIT, CLGIB, CLGIJ, CLGIT, CLGRB, CLGRJ, CLGRT, CLIB, CLIJ, CLRB, CLRJ, CLRT, CRB, CRJ, CRT. When the suffix is coded, the M_3 operand must be omitted.

CNOP Alignment

<table>
<tr><td colspan="16">Quadword</td></tr>
<tr><td colspan="8">Doubleword</td><td colspan="8">Doubleword</td></tr>
<tr><td colspan="4">Fullword</td><td colspan="4">Fullword</td><td colspan="4">Fullword</td><td colspan="4">Fullword</td></tr>
<tr><td colspan="2">Halfword</td><td colspan="2">Halfword</td><td colspan="2">Halfword</td><td colspan="2">Halfword</td><td colspan="2">Halfword</td><td colspan="2">Halfword</td><td colspan="2">Halfword</td><td colspan="2">Halfword</td></tr>
<tr><td>Byte</td><td>Byte</td><td>Byte</td><td>Byte</td><td>Byte</td><td>Byte</td><td>Byte</td><td>Byte</td><td>Byte</td><td>Byte</td><td>Byte</td><td>Byte</td><td>Byte</td><td>Byte</td><td>Byte</td><td>Byte</td></tr>
<tr><td colspan="2">0,4</td><td colspan="2">2,4</td><td colspan="2">0,4</td><td colspan="2">2,4</td><td colspan="2">0,4</td><td colspan="2">2,4</td><td colspan="2">0,4</td><td colspan="2">2,4</td></tr>
<tr><td colspan="2">0,8</td><td colspan="2">2,8</td><td colspan="2">4,8</td><td colspan="2">6,8</td><td colspan="2">0,8</td><td colspan="2">2,8</td><td colspan="2">4,8</td><td colspan="2">6,8</td></tr>
<tr><td colspan="2">0,16</td><td colspan="2">2,16</td><td colspan="2">4,16</td><td colspan="2">6,16</td><td colspan="2">8,16</td><td colspan="2">10,16</td><td colspan="2">12,16</td><td colspan="2">14,16</td></tr>
</table>

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
CA	-	Byte	Characters (ASCII)	RIght
CE	-	Byte	Characters (EBCDIC)	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
DD	8	Doubleword	Long decimal 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
ED	4	Word	Short decimal 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
JD	8	Doubleword	Symbol naming a DXD, DSECT, or class	Left
L	16	Doubleword	Extended hex floating point	Right
LB	16	Doubleword	Extended binary floating point	Right
LD	16	Doubleword	Extended decimal floating point	Right
LH	16	Doubleword	Extended hex floating point	Right
LQ	16	Quadword	Extended hex floating point	Right
P	-	Byte	Packed decimal	Left
Q	4	Word	Symbol naming a DXD, DSECT, or part	Left
QD	8	Doubleword	Symbol naming a DXD, DSECT, or part	Left

Type	Implied Length, Bytes	Default Alignment	Format	Truncation/ Padding
QY	3	Halfword	Symbol naming a DXD, DSECT, or part in long-displacement form	–
R	4	Word	PSECT address value	Left
RD	8	Doubleword	PSECT address value	Left
S	2	Halfword	Address in base-displacement form	–
SY	3	Halfword	Address in base-and-long-displacement form	–
V	4	Word	Externally defined address value	–
VD	8	Doubleword	Externally defined address value	–
X	-	Byte	Hexadecimal digits	Left
Y	2	Halfword	Value of address	Left
Z	-	Byte	Zoned decimal	Left

Source: SC26-4940.

Operand of Store Clock

Bits 0-63 of Time-of-Day (TOD) Clock
0 … 63

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

Operand of Store Clock Extended

Zeros	Time-of-Day (TOD) Clock	Programmable Field
0	8	112 … 127

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

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 31)
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 (see table on page 31)
144-147	R	90	Data-exception code: 0-23 zeros, 24-31 code (see table on page 32)
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, 4 STURA (with 2), 3 and 5-6 zeros, 7 instruction-fetching nullification (with 1), 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
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 33)
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

Area (Dec)	Addr Type	Hex Addr	Function
200-203	R	C8	STFL facility list (see “Facility Indications” on page 34 for the first 32 facility bits)
232-239	R	E8	Machine-check-interruption code (see diagram on page 54)
244-247	R	F4	External-damage code (see diagram on page 54)
248-255	R	F8	Failing-storage address
272-279	R	110	Breaking-event 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	ILC Set	Instr. Ending
0001	Operation exception	1 2 3	S
0002	Privileged-operation exception	2 3	S
0003	Execute exception	2	S
0004	Protection exception	1 2 3	S T
0005	Addressing exception	1 2 3	S T
0006	Specification exception	0 1 2 3	C S
0007	Data exception	1 2 3	C S T
0008	Fixed-point-overflow exception	1 2 3	C
0009	Fixed-point-divide exception	1 2 3	C S

Code (Hex)	Condition	ILC Set	Instr. Ending
000A	Decimal-overflow exception	2 3	C
000B	Decimal-divide exception	2 3	S
000C	HFP-exponent-overflow exception	1 2 3	C
000D	HFP-exponent-underflow exception	1 2 3	C
000E	HFP-significance exception	1 2	C
000F	HFP-floating-point-divide exception	1 2	S
0010	Segment-translation exception	1 2 3	N
0011	Page-translation exception	1 2 3	N
0012	Translation-specification exception	1 2 3	S
0013	Special-operation exception	1 2 3	S
0015	Operand exception	2	S
0016	Trace-table exception	1 2	N
001C	Space-switch event	0 1 2	C
001D	HFP-square-root exception	2	S
001F	PC-translation-specification exception	2	S
0020	AFX-translation exception	1 2	N
0021	ASX-translation exception	1 2	N
0022	LX-translation exception	2	N
0023	EX-translation exception	2	N
0024	Primary-authority exception	2	N
0025	Secondary-authority exception	1 2	N
0026	LFX-translation exception	2	N
0027	LSX-translation exception	2	N
0028	ALET-specification exception	1 2 3	S
0029	ALEN-translation exception	1 2 3	N
002A	ALE-sequence exception	1 2 3	N
002B	ASTE-validity exception	1 2 3	N
002C	ASTE-sequence exception	1 2 3	N
002D	Extended-authority exception	1 2 3	N
002E	LSTE sequence	2	N
002F	ASTE instance	1 2 3	N
0030	Stack-full exception	2	N
0031	Stack-empty exception	1 2	N
0032	Stack-specification exception	1 2	N
0033	Stack-type exception	1 2	N
0034	Stack-operation exception	1 2	N
0038	ASCE-type exception	1 2 3	N
0039	Region-first-translation exception	1 2 3	N
003A	Region-second-translation exception	1 2 3	N
003B	Region-third-translation exception	1 2 3	N
0040	Monitor event	2	C
0080	PER basic event (code may be combined with another code)	0 1 2 3	C
0080	PER nullification event	0	N
0119	Crypto-operation exception	2	N

C	Completed
ILC	Instruction-length code
N	Nullified
S	Suppressed
T	Terminated

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
03	DFP instruction
08	IEEE inexact and truncated
0B	IXS inexact
0C	IEEE inexact and incremented
10	IEEE underflow, exact
13	IXS underflow, exact
18	IEEE underflow, inexact and truncated
1B	IXS underflow, inexact

Code (Hex)	Data Exception
1C	IEEE underflow, inexact and incremented
20	IEEE overflow, exact
23	IXS overflow, exact
28	IEEE overflow, inexact and truncated
2B	IXS overflow, inexact
2C	IEEE overflow, inexact and incremented
40	IEEE division by zero
43	IXS division by zero
80	IEEE invalid operation
83	IXS invalid operation

Translation-Exception Identification

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

Interruption 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-conrolled protection, 62-63 ASCE identification, rest unpredictable, location 160 valid; if DAT was off, rest unpredictable
0010	Segment translation	0-51 address, 52-61 unpredictable, 62-63 ASCE identification
0011	Page translation	0-51 address, 52-60 unpredictable, if 61, zero, not MOVE PAGE; if 61 one, MOVE PAGE (see location 162); 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	AFX translation	32-47 zeros, 48-63 address-space number
0021	ASX translation	32-47 zeros, 48-63 address-space number
0022	LX translation	32-43 zeros, 44-63 program-call number
0023	EX translation	32-43 zeros, 44-63 program-call number
0024	Primary authority	32-47 zeros, 48-63 address-space number
0025	Secondary authority	32-47 zeros, 48-63 address-space number
0026 0027	LFX translation LSX translation	When bit 44 is 0: 32-43 zeros, 44-63 program-call number. When bit 44 is 1, 32-63 program-call number
0038	ASCE type	0-51 address, 52-61 unpredictable, 62-63 ASCE identification
0039	Region-first translation	0-51 address, 52-61 unpredictable, 62-63 ASCE identification
003A	Region-second translation	0-51 address, 52-61 unpredictable, 62-63 ASCE identification
003B	Region-third translation	0-51 address, 52-61 unpredictable, 62-63 ASCE identification

* Bits 0-31 (bytes 168-171) unchanged if not described.

Facility Indications

Stored at real-storage locations 200-203 (C8-CB hex) by STFL; stored at second-operand location by STFLE.

Bit	Meaning when Bit is One
0	The instructions marked "N3" in the instruction-summary figures in Chapters 7 and 10 are installed.
1	The z/Architecture architectural mode is installed.
2	The z/Architecture architectural mode is active. When this bit is zero, the ESA/390 architectural mode is active.
3	The DAT-enhancement facility is installed in the z/Architecture architectural mode. The DAT-enhancement facility includes the INVALIDATE DAT TABLE ENTRY (IDTE) and COMPARE AND SWAP AND PURGE (CSPG) instructions.
4	INVALIDATE DAT TABLE ENTRY (IDTE) performs the invalidation-and-clearing operation by selectively clearing combined region-and-segment-table entries when a segment-table entry or entries are invalidated. IDTE also performs the clearing-by-ASCE operation. Unless bit 4 is one, IDTE simply purges all TLBs. Bit 3 is one if bit 4 is one.
5	INVALIDATE DAT TABLE ENTRY (IDTE) performs the invalidation-and-clearing operation by selectively clearing combined region-and-segment-table entries when a region-table entry or entries are invalidated. Bits 3 and 4 are ones if bit 5 is one.
6	The ASN-and-LX reuse facility is installed in the z/Architecture architectural mode.
7	The store-facility-list-extended facility is installed.
8	The enhanced-DAT facility is installed in the z/Architecture architectural mode.
9	The sense-running-status facility is installed in the z/Architecture architectural mode.
10	The conditional-SSKE facility is installed in the z/Architecture architectural mode.
11	The configuration-topology facility is installed in the z/Architecture architectural mode.
16	The extended-translation facility 2 is installed.
17	The message-security assist is installed.
18	The long-displacement facility is installed in the z/Architecture architectural mode.
19	The long-displacement facility has high performance. Bit 18 is one if bit 19 is one.
20	The HFP-multiply-add/subtract facility is installed.
21	The extended-immediate facility is installed in the z/Architecture architectural mode.
22	The extended-translation facility 3 is installed in the z/Architecture architectural mode.
23	The HFP-unnormalized-extension facility is installed in the z/Architecture architectural mode.
24	The ETF2-enhancement facility is installed.
25	The store-clock-fast facility is installed in the z/Architecture architectural mode.
26	The parsing-enhancement facility is installed in the z/Architecture architectural mode.
27	The move-with-optional-specifications facility is installed in the z/Architecture architectural mode.
28	The TOD-clock-steering facility is installed in the z/Architecture architectural mode.
30	The ETF3-enhancement facility is installed in the z/Architecture architectural mode.
31	The extract-CPU-time facility is installed in the z/Architecture architectural mode.
32	The compare-and-swap-and-store facility is installed in the z/Architecture architectural mode.
33	The compare-and-swap-and-store facility 2 is installed in the z/Architecture architectural mode.
34	The general-instructions-extension facility is installed in the z/Architecture architectural mode.
35	The execute-extensions facility is installed in the z/Architecture architectural mode.
39	Assigned to IBM internal use.
41	The floating-point-support-enhancement facilities (FPR-GR-loading, FPS-sign-handling, and DFP-rounding) are installed in the z/Architecture architectural mode.
42	The DFP (decimal-floating-point) facility is installed in the z/Architecture architectural mode.
43	The DFP (decimal-floating-point) facility has high performance. Bit 42 is one if bit 43 is one.
44	The PFPO instruction is installed in the z/Architecture architectural mode.

Control Registers

CR	Bits	Name of Field	Associated with	Init*
0	32	Trace TOD-clock control	TOD clock	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
	40	Enhanced-DAT-enablement control	Dynamic address translation	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	
3	32-47	PSW-key mask	Instruction authorization	
	48-63	Secondary ASN	Address spaces	
4	32-47	Authorization index	Instruction authorization	
	48-63	Primary ASN	Address spaces	
5	33-57	Primary-ASTE origin	Access-register translation	
6	32-39	I/O-interruption subclass mask	I/O interruptions	
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
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
	39	Instruction-fetching-nullification-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

CR	Bits	Name of Field	Associated with	Init*
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

Masks						Flags										
IMi	IMz	IMo	IMu	IMx	0 0 0	SMi	SMz	SMo	SMu	SMx	0 0 0	DXC (see page 32)	0	DRM	0 0	BRM
0						8						16	24			31

Bit	**Meaning**
0	(IMi) IEEE-invalid-operation mask
1	(IMz) IEEE-division-by-zero mask
2	(IMo) IEEE-overflow mask
3	(IMu) IEEE-underflow mask
4	(IMx) IEEE-inexact mask
8	(SFi) IEEE-invalid-operation flag
9	(SFz) IEEE-division-by-zero flag
10	(SFo) IEEE-overflow flag
11	(SFu) IEEE-underflow flag
12	(SFx) IEEE-inexact flag
16-23	(DXC) Data-exception code (see table on page 32)
25-27	(DRM) DFP Rounding mode 000 Round to nearest with ties to even 001 Round toward 0 010 Round toward +∞ 011 Round toward -∞ 100 Round to nearest with ties away from 0 101 Round to nearest with ties toward 0 110 Round away from 0 111 Round to prepare for shorter precision

30-31	(BRM) BFP Rounding mode 00 Round to nearest 01 Round toward 0 10 Round toward +∞ 11 Round toward -∞

Program-Status Word (PSW)

z/Architecture PSW

0	R	0 0 0	T	I	E	PSW Key	0	M	W	P	AS	CC	Program Mask	0 0 0 0 0 0 0	EA
0			5			8	12				16	18	20	24	31

BA	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
32	63

Bits 0-31 of Instruction Address	
64	95

Bits 32-63 of Instruction Address	
96	127

Bit	**Meaning**
1	(R) Program-event-recording mask
5	(T = 1) DAT mode
6	(I) Input/output mask
7	(E) External mask
12	Zero indicates z/Architecture
13	(M) Machine-check mask
14	(W = 1) Wait state
15	(P = 1) Problem state
16-17	xx Real mode (T = 0) 00 Primary-space mode (T = 1) 01 Access-register mode (T = 1) 10 Secondary-space mode (T = 1) 11 Home-space mode (T = 1)
18-19	(CC) Condition code
20	Fixed-point-overflow mask
21	Decimal-overflow mask
22	HFP-exponent-underflow mask
23	HFP-significance mask
31/32	Extended/basic addressing mode 00 24-bit mode 01 31-bit mode 10 Invalid 11 64-bit mode

ESA/390 PSW

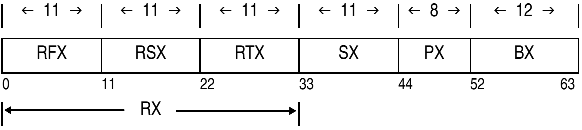
0	R	0 0 0	T	I	E	PSW Key	1	M	W	P	AS	CC	Program Mask	0 0 0 0 0 0 0 0
0			5			8	12				16	18	20	24 31

A	Instruction Address
32	63

Bit	**Meaning**
12	One indicates ESA/390
32	(A = 1) 31-bit addressing mode

Dynamic Address Translation

Virtual-Address Format

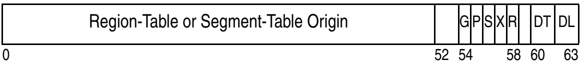


← 11 →
← 11 →
← 11 →
← 11 →
← 8 →
← 12 →
RFX
RSX
RTX
SX
PX
BX
0
11
22
33
44
52
63
RX

Field	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)

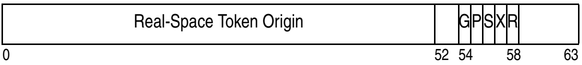
Region-Table or Segment-Table Designation (RTD or STD)



Region-Table or Segment-Table Origin
G
P
S
X
R
DT
DL
0
52
54
58
60
63

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	(DL) Designation length (× 4K bytes)

Real-Space Designation (RSD)



Real-Space Token Origin
G
P
S
X
R
0
52
54
58
63

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

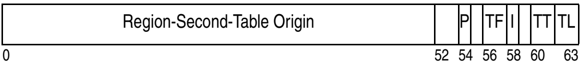
Note: Other bits are as in RTD or STD.

Table Values

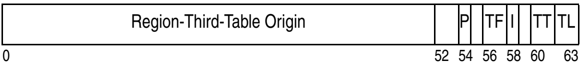
Table	Incre-ment	Incr. Size	Incr. Ent-ries	Max. Size	Max. Ent-ries	Max Table Maps	
						Regions	Bytes
Region First	1-4	4KB	512	16KB	2K	8G	16E =16×2^{60}
Region Second	1-4	4KB	512	16KB	2K	4M	8P = 8×2^{50}
Region Third	1-4	4KB	512	16KB	2K	2K	4T = 4×2^{40}
Segment	1-4	4KB	512	16KB	2K	1	2G = 2×2^{30}
Page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

Region-Table Entry (RTE)

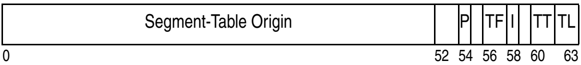
Region-First-Table Entry (RFTE)


Region-Second-Table Origin
P
TF
I
TT
TL
0
52
54
56
58
60
63

Region-Second-Table Entry (RSTE)

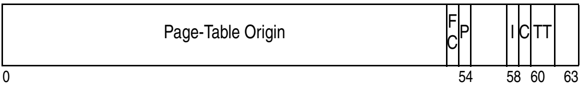

Region-Third-Table Origin
P
TF
I
TT
TL
0
52
54
56
58
60
63

Region-Third-Table Entry (RTTE)

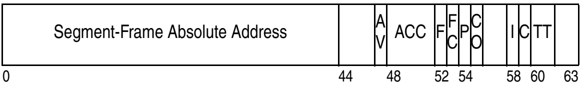

Segment-Table Origin
P
TF
I
TT
TL
0
52
54
56
58
60
63

Bit	Meaning
54	DAT protection bit
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, FC=0)

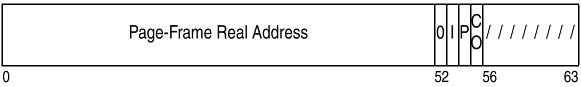

Page-Table Origin
FC
P
I
C
TT
0
54
58
60
63

Segment-Table Entry (STE, FC=1)


Segment-Frame Absolute Address
AV
ACC
F
FC
P
CO
I
C
TT
0
44
48
52
54
58
60
63

Bit	Meaning
47	(AV) Access-control (ACC) and fetch-protection (F) validity bit
48-51	(ACC) Access-control bits
52	(F) Fetch-protection bit
53	(FC) Format control
54	(P) DAT-protection bit
55	(CO) Change-bit override
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)


Page-Frame Real Address
0
I
P
CO
/ / / / / / / /
0
52
56
63

Bit	Meaning
53	(I) Page-invalid bit
54	(P) Page-protection bit
55	(CO) Change-bit override

ASN Translation

Address-Space Number (ASN)

ASN-First-Table Index	ASN-Second-Table Index
0	10 15

ASN-First-Table Entry

I	ASN-Second-Table Origin	0 0 0 0 0 0
0	1	26 31

Bit	Meaning
0	(I) AFX-invalid bit

ASN-Second-Table Entry (ASTE)

Byte (Hex)										
0	I	Authority-Table Origin								B
	0	1								31

Byte (Hex)					
4	Authorization Index	Authority-Table Length		CA	RA
	0	16	28		31

Address-Space-Control Element (ASCE=RTD/STD/RSD) Part 1

Byte (Hex)	
8	Region or Segment-Table Origin or Real-Space Token Origin
	0 31

RTD or STD Part 2 (R=0)

Byte (Hex)										
C	Region or Seg.-Table Origin (Continued)		G	P	S	X	R		DT	DL
	0	20	22				26		28	31

RSD Part 2 (R=1)

Byte (Hex)								
C	Real-Space Token Origin (Continued)		G	P	S	X	R	
	0	20	22				26	31

Byte (Hex)			
10		Primary-Space Access-List Origin	ALL
	0	1	25 31

Byte (Hex)	
14	ASN-Second-Table-Entry Sequence Number
	0 31

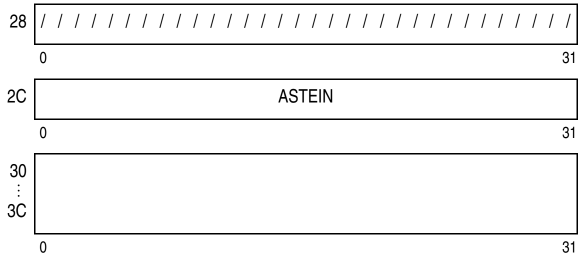
Linkage-Table Designation (LTD)

Byte (Hex)			
18	V	Linkage-Table Origin	LTL
	0	1	25 31

Linkage-First-Table Designation (LFTD)

Byte (Hex)			
18	V	Linkage-First-Table Origin	LFTL
	0	1	24 31

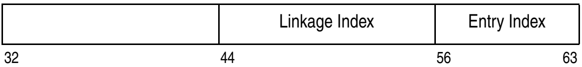
Byte (Hex)	
1C ⋮ 24	Available for Programming
	0 31


28
/ /
0
31
2C
ASTEIN
0
31
30
⋮
3C
0
31

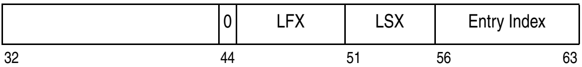
Byte.Bit	Meaning
0.0	(I) ASX-invalid bit
0.31	(B) Base-space bit
4.30	(CA) Controlled-ASN bit
4.31	(RA) Reusable-ASN bit
10.25-31	(ALL) Access-list length (× 128 bytes)
18.0	(V) Subsystem-linkage control
18.25-31	(LTL) Linkage-table length (× 128 bytes)
18.24-31	(LFTL) Linkage-first-table length (× 256 bytes)

PC-Number Translation

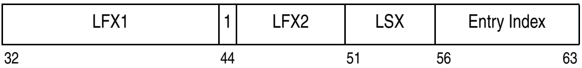
Program-Call Number (20-Bit)


Linkage Index
Entry Index
32
44
56
63

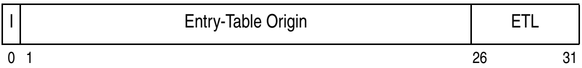
Program-Call Number (32-Bit, Bit 44=0)


0
LFX
LSX
Entry Index
32
44
51
56
63

Program-Call Number (32-Bit, Bit 44=1)

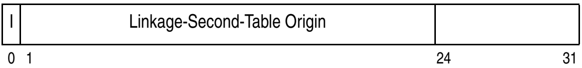

LFX1
1
LFX2
LSX
Entry Index
32
44
51
56
63

Linkage-Table Entry (LTE)


I
Entry-Table Origin
ETL
0
1
26
31

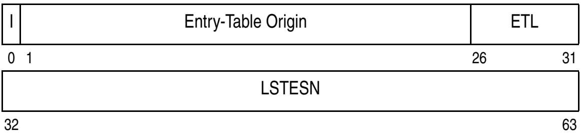
Bit	Meaning
0	(I) LX-invalid bit
26-31	(ETL) Entry-table length (× 128 bytes)

Linkage-First-Table Entry (LFTE)


I
Linkage-Second-Table Origin
0
1
24
31

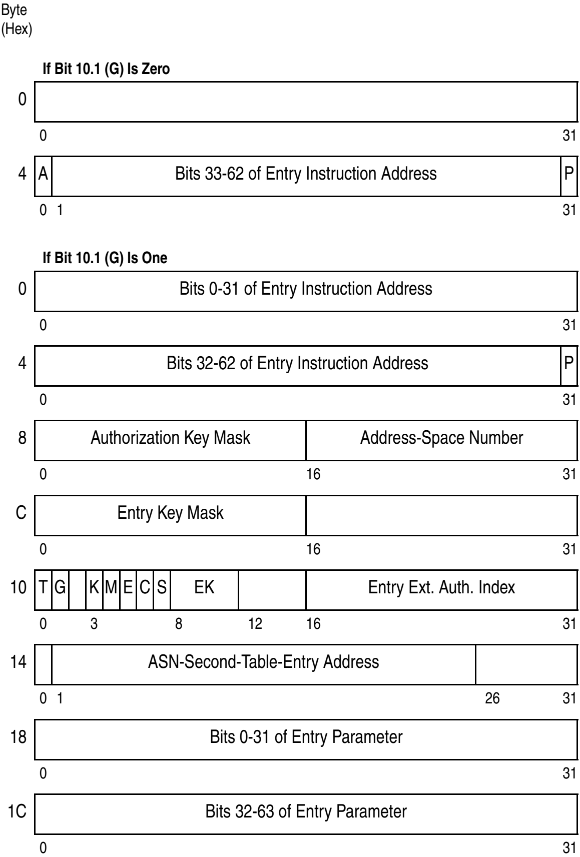
Bit	Meaning
0	(I) LFX-invalid bit

Linkage-Second-Table Entry (LSTE)


I
Entry-Table Origin
ETL
0
1
26
31
LSTESN
32
63

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

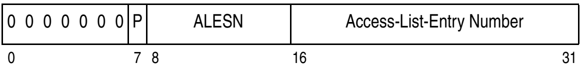
Entry-Table Entry (ETE)


Byte
(Hex)
If Bit 10.1 (G) Is Zero
0
0
31
4
A
Bits 33-62 of Entry Instruction Address
P
0
1
31
If Bit 10.1 (G) Is One
0
Bits 0-31 of Entry Instruction Address
0
31
4
Bits 32-62 of Entry Instruction Address
P
0
31
8
Authorization Key Mask
Address-Space Number
0
16
31
C
Entry Key Mask
0
16
31
10
T
G
K
M
E
C
S
EK
Entry Ext. Auth. Index
0
3
8
12
16
31
14
ASN-Second-Table-Entry Address
0
1
26
31
18
Bits 0-31 of Entry Parameter
0
31
1C
Bits 32-63 of Entry Parameter
0
31

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

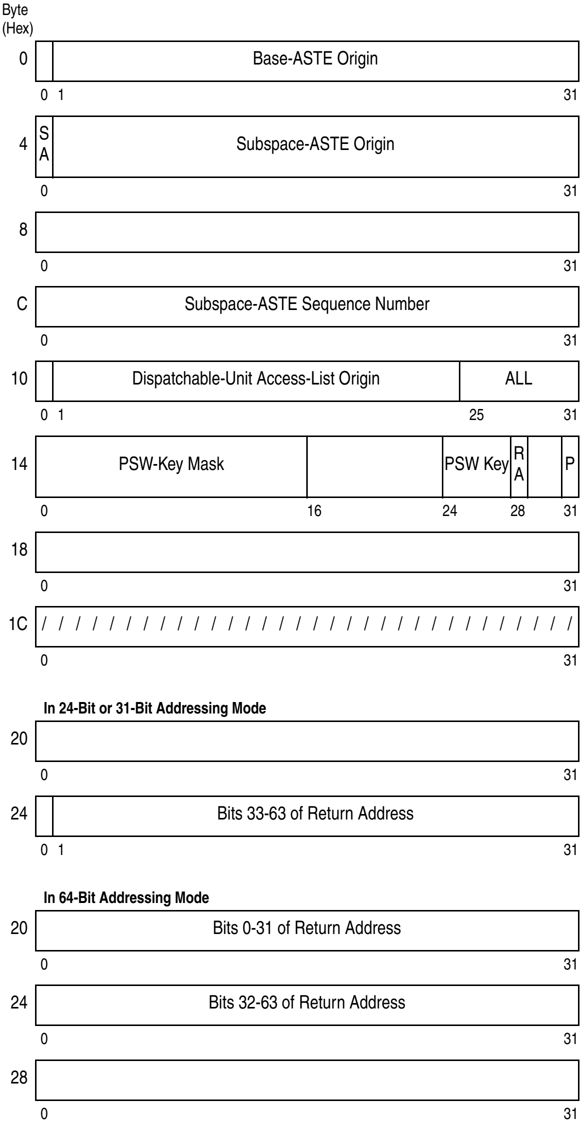
Access-Register Translation

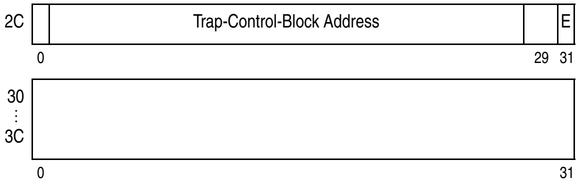
Access-List-Entry Token (ALET)


0 0 0 0 0 0 0
P
ALESN
Access-List-Entry Number
0
7
8
16
31

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

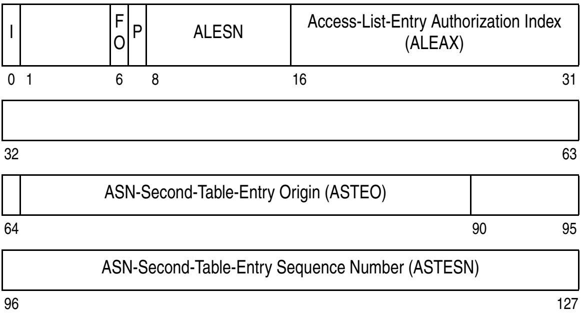
Dispatchable-Unit-Control Table (DUCT)


Byte (Hex)
0
Base-ASTE Origin
0 1
31
4
S A
Subspace-ASTE Origin
0
31
8
0
31
C
Subspace-ASTE Sequence Number
0
31
10
Dispatchable-Unit Access-List Origin
ALL
0 1
25
31
14
PSW-Key Mask
PSW Key
R A
P
0
16
24
28
31
18
0
31
1C
0
31
In 24-Bit or 31-Bit Addressing Mode
20
0
31
24
Bits 33-63 of Return Address
0 1
31
In 64-Bit Addressing Mode
20
Bits 0-31 of Return Address
0
31
24
Bits 32-63 of Return Address
0
31
28
0
31


2C
Trap-Control-Block Address
E
0
29
31
30
⋮
3C
0
31

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

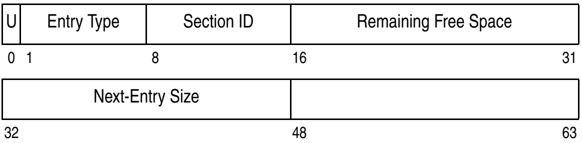
Access-List Entry (ALE)


I
F O
P
ALESN
Access-List-Entry Authorization Index (ALEAX)
0
1
6
8
16
31
32
63
ASN-Second-Table-Entry Origin (ASTEO)
64
90
95
ASN-Second-Table-Entry Sequence Number (ASTESN)
96
127

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

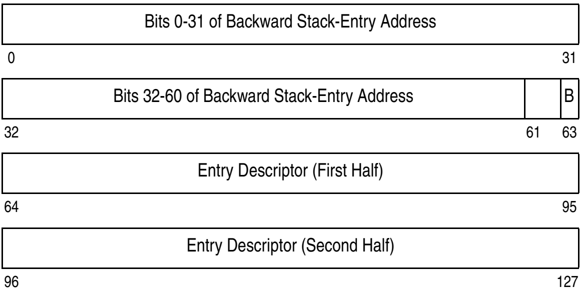
Linkage-Stack Entries

Entry Descriptor


U
Entry Type
Section ID
Remaining Free Space
0
1
8
16
31
Next-Entry Size
32
48
63

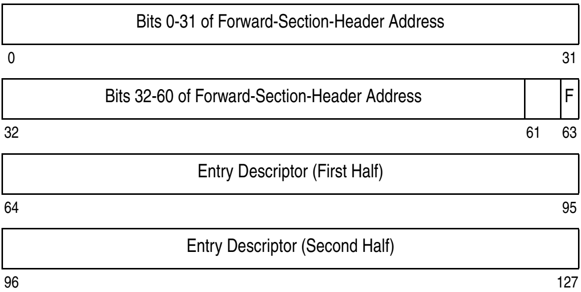
Bit	Meaning
0	(U) Unstack-suppression bit
1-7	Entry type: Header entry = 0001001 binary Trailer entry = 0001010 binary Branch state entry = 0001100 binary Program-call state entry = 0001101 binary Available for program use = 1xxxxxx binary

Header Entry (Entry Type 0001001)


Bits 0-31 of Backward Stack-Entry Address
0
31
Bits 32-60 of Backward Stack-Entry Address
B
32
61
63
Entry Descriptor (First Half)
64
95
Entry Descriptor (Second Half)
96
127

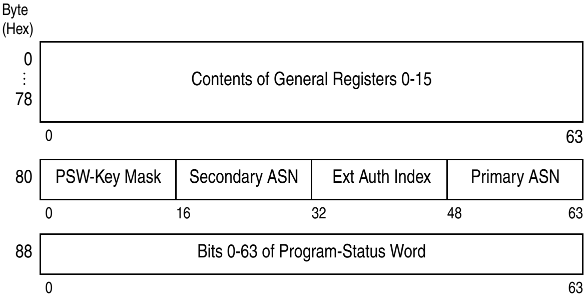
Bit	Meaning
63	(B) Backward stack-entry validity bit

Trailer Entry (Entry Type 0001010)

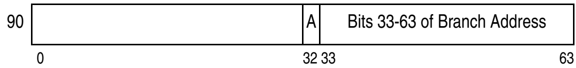

Bits 0-31 of Forward-Section-Header Address
0
31
Bits 32-60 of Forward-Section-Header Address
F
32
61
63
Entry Descriptor (First Half)
64
95
Entry Descriptor (Second Half)
96
127

Bit	Meaning
63	(F) Forward-section validity bit

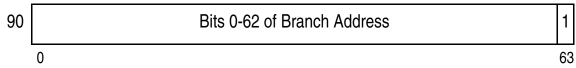
Branch State Entry (Entry Type 0001100) and Program-Call State Entry (Entry Type 0001101)

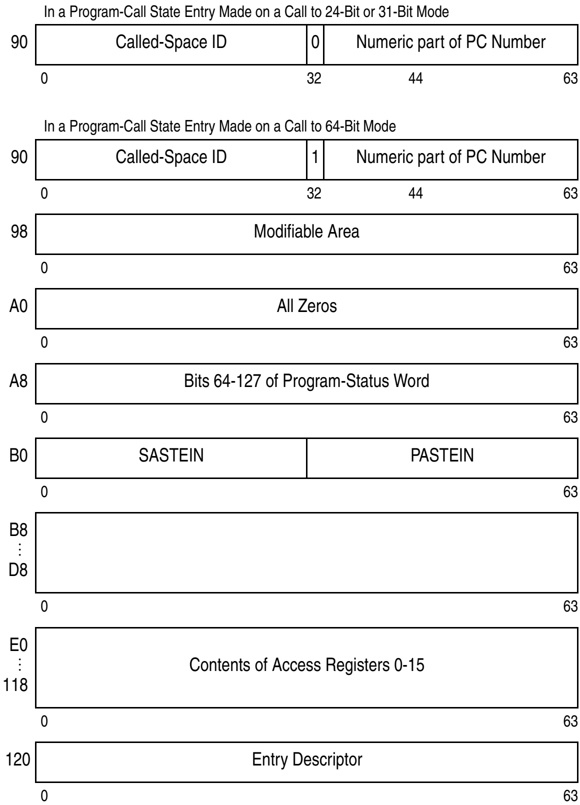

Byte (Hex)
0
⋮
78
Contents of General Registers 0-15
0
63
80
PSW-Key Mask
Secondary ASN
Ext Auth Index
Primary ASN
0
16
32
48
63
88
Bits 0-63 of Program-Status Word
0
63

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


90
A
Bits 33-63 of Branch Address
0
32 33
63

In a Branch State Entry Made in 64-Bit Mode

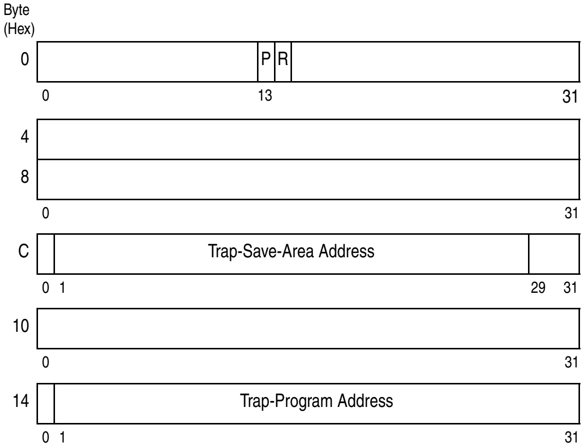

90
Bits 0-62 of Branch Address
1
0
63

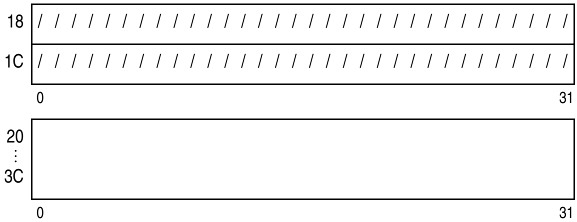

In a Program-Call State Entry Made on a Call to 24-Bit or 31-Bit Mode
90
Called-Space ID
0
Numeric part of PC Number
0
32
44
63
In a Program-Call State Entry Made on a Call to 64-Bit Mode
90
Called-Space ID
1
Numeric part of PC Number
0
32
44
63
98
Modifiable Area
0
63
A0
All Zeros
0
63
A8
Bits 64-127 of Program-Status Word
0
63
B0
SASTEIN
PASTEIN
0
63
B8
⋮
D8
0
63
E0
⋮
118
Contents of Access Registers 0-15
0
63
120
Entry Descriptor
0
63

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

Trapping

Trap Control Block

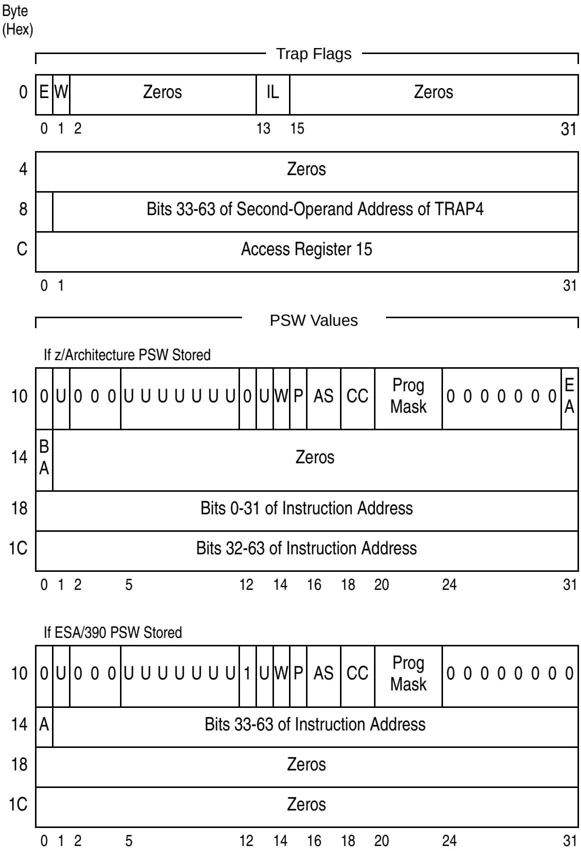

Byte
(Hex)
0
P
R
0
13
31
4
8
0
31
C
Trap-Save-Area Address
0
1
29
31
10
0
31
14
Trap-Program Address
0
1
31



18
1C
20
⋮
3C
/ /
0
31

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

Trap Save Area



Byte (Hex)
Trap Flags
0
E
W
Zeros
IL
Zeros
0 1 2 13 15 31
4
Zeros
8
Bits 33-63 of Second-Operand Address of TRAP4
C
Access Register 15
0 1 31
PSW Values
If z/Architecture PSW Stored
10
0 U 0 0 0 U U U U U U U 0 U W P AS CC
Prog Mask
0 0 0 0 0 0 0
EA
14
BA
Zeros
18
Bits 0-31 of Instruction Address
1C
Bits 32-63 of Instruction Address
0 1 2 5 12 14 16 18 20 24 31
If ESA/390 PSW Stored
10
0 U 0 0 0 U U U U U U U 1 U W P AS CC
Prog Mask
0 0 0 0 0 0 0 0
14
A
Bits 33-63 of Instruction Address
18
Zeros
1C
Zeros
0 1 2 5 12 14 16 18 20 24 31

Offset	Bits 0–31
20 ⋮ 9C	General Registers 0-15
A0	/ /
A4	/ /
A8 ⋮ FC	Unchanged
	0 … 31

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 37)
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		000N	Set Secondary ASN	1
00100001			Program Call	1[1]
00100010			Program Call	2[1]
00100001		0	Program Call	3[1]
00100010		0	Program Call	4[1]
00100010		100E	Program Call	5[1]
00100010		101E	Program Call	6[1]
00100011		111E	Program Call	7[1]
00110001		000N	Program Transfer	1
00110001		100N	Program Transfer	2
00110010		0000	Program Return	1
00110010		0010	Program Return	2
00110010		1000	Program Return	4
00110010		1010	Program Return	5
00110010		110N	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

Trace-Entry Bits			Trace Entry	
0-7	8-11	12-15	Type	Format
0111	0		Trace	1
0111	1		Trace	2
1			Branch	

[1] Format-1 and -2 entries are made when the ASN-and-LX-reuse facility (ALRF) is not enabled. Entries of formats 3-7 are made when the facility is enabled.

E Indicates, when one, that the extended-addressing-mode bit, PSW bit 31, was set to one.

N Indicates, when one, that an entry was made because of PTI or SSAIR.

Branch

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

00000000	Bits 40-63 of Branch Address	
0	8	31

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

1	Bits 33-63 of Branch Address	
0		31

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

01010010	1100	All Zeros	Bits 0-31 of Branch Address	
0	8	12	32	63

Bits 32-63 of Branch Address	
64	95

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

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	
0	8		32		63

F2 (in 64-Bit Mode)

01000010	P	Bits 9-31 of ALET	Bits 33-63 of Branch Address	
0	8		32	63

Bits 32-63 of Branch Address	
64	95

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	
0	8	12	32		63

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. Address	
0	8	12	32	63

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. Address
0	8	12	32 63

Bits 32-63 of Updated Inst. Address
64 95

Mode-Switching Branch

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

01010001	1010	All Zeros	A	Branch Address
0	8	12	32	63

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

01010001	1011	All Zeros	Bits 32-63 of Branch Address
0	8	12	32 63

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

01010010	1111	All Zeros	Bits 0-31 of Branch Address
0	8	12	32 63

Bits 32-63 of Branch Address
64 95

Program Call

F1 (in 24/31-Bit Mode, ALRF Not Enabled)

00100001	PSW Key	All Zeros	A	Bits 33-62 of Return Address	P
0	8	12	32		63

F2 (in 64-Bit Mode, ALRF Not Enabled)

00100010	PSW Key	All Zeros	Bits 0-31 of Return Address
0	8	12	32 63

Bits 32-62 of Return Address	P
64	95

F3 (in 24/31-Bit Mode, ALRF Enabled, 20-Bit PC Number)

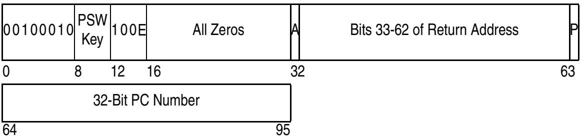
00100001	PSW Key	0	Bits 1-19 of 20-Bit PC Number	A	Bits 33-62 of Return Address	P
0	8	12		32		63

F4 (in 64-Bit Mode, ALRF Enabled, 20-Bit PC Number)

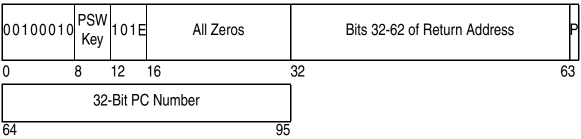
00100010	PSW Key	0	Bits 1-19 of 20-Bit PC Number	Bits 0-31 of Return Address
0	8	12		32 63

Bits 32-62 of Return Address	P
64	95

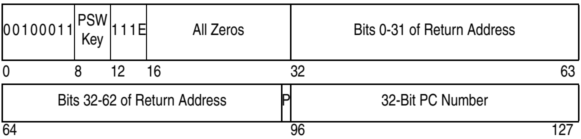
F5 (in 24/31-Bit Mode, ALRF Enabled, 32-Bit PC Number)


00100010
PSW Key
100E
All Zeros
A
Bits 33-62 of Return Address
P
0
8
12
16
32
63
32-Bit PC Number
64
95

F6 (in 64-Bit Mode, ALRF Enabled, 32-Bit PC Number, Bits 0-31 of Return Address All Zeros)

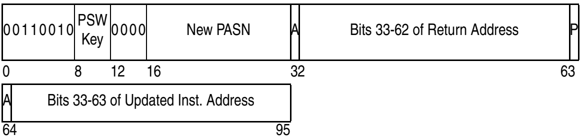

00100010
PSW Key
101E
All Zeros
Bits 32-62 of Return Address
P
0
8
12
16
32
63
32-Bit PC Number
64
95

F7 (in 64-Bit Mode, ALRF Enabled, 32-Bit PC Number, Bits 0-31 of Return Address Not All Zeros)

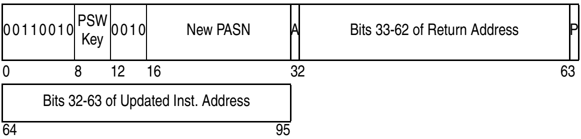

00100011
PSW Key
111E
All Zeros
Bits 0-31 of Return Address
0
8
12
16
32
63
Bits 32-62 of Return Address
P
32-Bit PC Number
64
96
127

Program Return

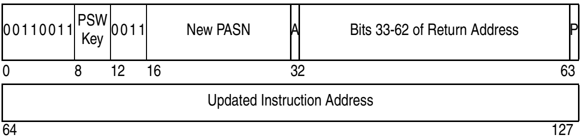
F1 (in 24/31-Bit to 24/31-Bit Mode)


00110010
PSW Key
0000
New PASN
A
Bits 33-62 of Return Address
P
0
8
12
16
32
63
A
Bits 33-63 of Updated Inst. Address
64
95

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


00110010
PSW Key
0010
New PASN
A
Bits 33-62 of Return Address
P
0
8
12
16
32
63
Bits 32-63 of Updated Inst. Address
64
95

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


00110011
PSW Key
0011
New PASN
A
Bits 33-62 of Return Address
P
0
8
12
16
32
63
Updated Instruction Address
64
127

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

00110010	PSW Key	1000	New PASN	Bits 32-62 of Return Address	P
0	8	12	16	32	63

A	Bits 33-63 of Updated Inst. Address
64	95

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

00110010	PSW Key	1010	New PASN	Bits 32-62 of Return Address	P
0	8	12	16	32	63

Bits 32-63 of Updated Inst. Address
64 95

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

00110011	PSW Key	1011	New PASN	Bits 32-62 of Return Address	P
0	8	12	16	32	63

Updated Instruction Address
64 127

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

00110011	PSW Key	1100	New PASN	Bits 0-31 of Return Address
0	8	12	16	32 63

Bits 32-62 of Return Address	P	A	Updated Instruction Address
64		96	127

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

00110011	PSW Key	1110	New PASN	Bits 0-31 of Return Address
0	8	12	16	32 63

Bits 32-62 of Return Address	P	Bits 32-63 of Updated Inst. Address
64		96 127

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

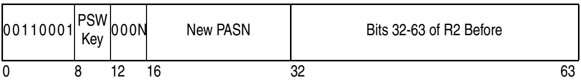
00110100	PSW Key	1111	New PASN	Bits 0-31 of Return Address
0	8	12	16	32 63

Bits 32-62 of Return Address	P	Bits 0-31 of Updated Inst. Address
64		96 127

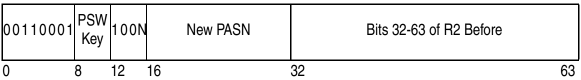
Bits 32-63 of Updated Inst. Address
128 159

Program Transfer

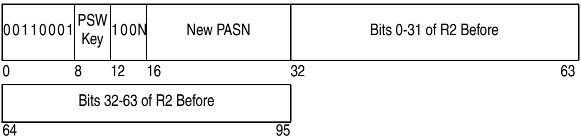
F1 (in 24/31-Bit Mode)


00110001
PSW Key
000N
New PASN
Bits 32-63 of R2 Before
0
8
12
16
32
63

F2 (in 64-Bit Mode, Bits 0-31 of R_2 All Zeros)

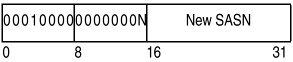

00110001
PSW Key
100N
New PASN
Bits 32-63 of R2 Before
0
8
12
16
32
63

F3 (in 64-Bit Mode, Bits 0-31 of R_2 Not All Zeros)


00110001
PSW Key
100N
New PASN
Bits 0-31 of R2 Before
0
8
12
16
32
63
Bits 32-63 of R2 Before
64
95

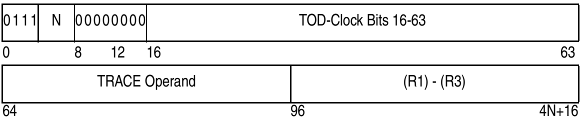
Set Secondary ASN

F1

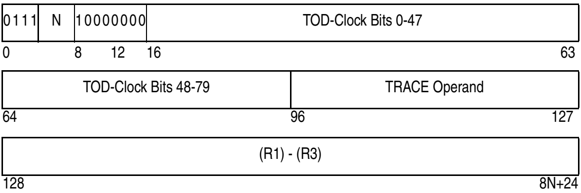

00010000
0000000N
New SASN
0
8
16
31

Trace

F1 (TRACE)


0111
N
00000000
TOD-Clock Bits 16-63
0
8
12
16
63
TRACE Operand
(R1) - (R3)
64
96
4N+16

F2 (TRACG)


0111
N
10000000
TOD-Clock Bits 0-47
0
8
12
16
63
TOD-Clock Bits 48-79
TRACE Operand
64
96
127
(R1) - (R3)
128
8N+24

Bit	**Meaning**
4-7	(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)

SD	PD	SR	0	CD	ED	0	DG	W	CP	SP	CK	0	0	B	0	SE	SC	KE	DS	WP	MS	PM	IA	FA	0	EC	FP	GR	CR	0	ST
0				4				8						14		16								24		26					31

IE	AR	DA	0	0	0	0	0	0	0	PR	FC	AP	0	CT	CC	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
32			35					40		42				46		48								56							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
7	(DG) Degradation
8	(W) Warning
9	(CP) Channel report pending
10	(SP) Service-processor damage
11	(CK) Channel-subsystem damage
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
42	(PR) TOD-programmable-register validity
43	(FC) Floating-point-control-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	XN	XF	0	0	0	0	0	0	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)

<table>
<tr><th>Word</th><th colspan="4"></th></tr>
<tr><td>0</td><td colspan="4">Interruption Parameter</td></tr>
<tr><td>1</td><td>Key S C M Y</td><td>F P I A U 0 H T</td><td>LPM</td><td>L D 0 0 0 0 0 X</td></tr>
<tr><td>2</td><td colspan="4">0 Channel-Program Address</td></tr>
<tr><td>3</td><td>CSS Priority</td><td>Reserved</td><td>CU Priority</td><td>Reserved</td></tr>
<tr><td>4</td><td colspan="4">Reserved</td></tr>
<tr><td>5</td><td colspan="4">Reserved</td></tr>
<tr><td>6</td><td colspan="4">Reserved</td></tr>
<tr><td>7</td><td colspan="4">Reserved</td></tr>
<tr><td></td><td>0</td><td>8</td><td>16</td><td>24 31</td></tr>
</table>

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
1.25	(D) Modified-CCW-indirect-data-addressing control
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

<table>
<tr><td>Command Code</td><td colspan="3">Data Address</td></tr>
<tr><td>0</td><td>8</td><td></td><td>31</td></tr>
<tr><td>Flags</td><td></td><td colspan="2">Byte Count</td></tr>
<tr><td>32</td><td>40</td><td>48</td><td>63</td></tr>
</table>

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
39	(MIDA) Causes bits 8-31 of CCW to specify location of first MIDAW

Format-1 CCW

Command Code	Flags	Byte Count	
0	8	16	31

0	Data Address	
32		63

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 33-63 of CCW to specify location of first IDAW
14	(Suspend) Causes suspension before execution of this CCW
15	(MIDA) Causes bits 33-63 of CCW to specify location of first MIDAW

Indirect-Data-Address Word (IDAW)

Format-1 IDAW

0	Data Address	
0	1	31

Format-2 IDAW

Bits 0-31 of Data Address	
0	31

Bits 32-63 of Data Address	
32	63

Modified-CCW-Indirect-Data-Address Word (MIDAW)

Reserved	
0	31

Reserved	Flags	Count	
32	40	48	63

Bits 0-31 of Data Address	
64	95

Bits 32-63 of Data Address	
96	127

Bit	**Meaning**
40	Last MIDAW
41	Skip
42	Data-transfer-interruption control
43-47	Reserved

Subchannel-Information Block (SCHIB)

Word	
0	Path-Management-Control Word
1	
2	
3	
4	
5	
6	
7	Subchannel-Status Word (See “Subchannel-Status Word (SCSW)” on page 58.)
8	
9	
10	Model-Dependent Area / Measurement-Block Address
11	
12	Model-Dependent Area
	0 … 31

Path-Management-Control Word (PMCW)

Word				
0	Interruption Parameter			
1	0 0 ISC 0 0 0	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	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 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)

Word	
0 1 2	Subchannel-Status Word
3 4 5 6 7	Extended-Status Word
8 ⋮ 15	Extended-Control Word
16 ⋮ 23	Extended-Measurement Word
	0 … 31

Subchannel-Status Word (SCSW)

Word																
														Subchannel Control		
0	Key	S	L	CC	F	P	I	A	U	Z	E	N	0	FC	AC	SC
	0	4	5	6	8					13				17	20	27 31

Word			
1	CCW Address		
2	Device Status	Subchan. Status	Byte Count
	0	8	16 31

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 21 (04) Start pending 22 (02) Halt pending 23 (01) Clear pending 24 (80) Subchannel active 25 (40) Device active 26 (20) Suspended
0.27-31	(SC) Status control 27 (10) Alert 28 (08) Intermediate 29 (04) Primary 30 (02) Secondary 31 (01) Status pending
2.0-15	Device status (0-7) 0 (80) Attention 1 (40) Status modifier 2 (20) Control-unit end 3 (10) Busy 4 (08) Channel end 5 (04) Device end 6 (02) Unit check 7 (01) Unit exception Subchannel status (8-15) 8 (80) Program-controlled int. 9 (40) Incorrect length 10 (20) Program check 11 (10) Protection check 12 (08) Channel-data check 13 (04) Channel-control check 14 (02) Interface-control check 15 (01) Chaining check

Extended-Status Word (ESW)

See chart on page 60 to determine the appropriate ESW format.

Format-0 ESW

Word	0 — 31
0	Subchannel Logout
1	Extended-Report Word
2 3	Failing-Storage Address
4	Secondary-CCW Address

Format-0 ESW Word 0 (Subchannel Logout)

0	ESF	LPUM	R	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
16	(R) Ancillary Report
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)

0	L	E	A	P	T	F	S	C	R	SCNT	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0			3					8		10	16 31

Bit **Meaning**

Bit	Meaning
1	(L) Request logging only
2	(E) Extended-subchannel-logout pending
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[1]

0 0 0 0 0 0 0 0	LPUM	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0	8	16 31

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

Format-2 ESW Word 0[1]

0 0 0 0 0 0 0 0	LPUM	DCTI
0	8	16 31

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

Format-3 ESW Word 0[1]

0 0 0 0 0 0 0 0	LPUM	Unpredictable
0	8	16 31

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		Device-Connect-Time Measurement-Mode Active					
Status-Control Field					L Bit	Suspended Bit	Timing-Facility Bit	Device-Connect-Time Measurement-Mode Enable Bit		Format	Contents Word 0 Byte			
A	I	P	S	X							0	1	2	3
-	-	-	-	0	-	*	*	*	No / Yes	U	*	*	*	*
*	*	0	0	1	1	*	*	*	No / Yes	0	R	R	R	R
*	*	1	*	1	1	*	*	*	No / Yes	0	R	R	R	R
1	0	0	1	1	1	*	*	*	No / Yes	0	R	R	R	R
0	0	0	0	1	0	*	*	*	No / Yes	U	*	*	*	*
0	0	0	1	1	0	*	*	*	No / Yes	3	Z	M	*	*
1	0	0	*	1	0	*	*	*	No / Yes	3	Z	M	*	*
*	*	1	*	1	0	*	0	*	No / Yes	1	Z	M	Z	Z
*	*	1	*	1	0	*	1	0	No / Yes	1	Z	M	Z	Z

---

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

Subchannel Conditions under which ESW Is Stored by Test Subchannel Instruction										Extended-Status Word (ESW)				
Subchannel-Status Word							Path-Management-Control Word							
Status-Control Field											Contents Word 0 Byte			
A	I	P	S	X	L Bit	Suspended Bit	Timing-Facility Bit	Device-Connect-Time Measurement-Mode Enable Bit	Device-Connect-Time Measurement-Mode Active	Format	0	1	2	3
*	*	1	*	1	0	*	1	1	No	1	Z	M	Z	Z
*	*	1	*	1	0	*	1	1	Yes	2	Z	M	D	D
0	1	0	0	1	0	0	*	*	No / Yes	U	*	*	*	*
0	1	0	0	1	0	1	0	*	No / Yes	1	Z	M	Z	Z
0	1	0	0	1	0	1	1	0	No / Yes	1	Z	M	Z	Z
0	1	0	0	1	0	1	1	1	No	1	Z	M	Z	Z
0	1	0	0	1	0	1	1	1	Yes	2	Z	M	D	D
0	0	0	1	1	1	These combinations do not occur.								
1	1	0	0	1	0									
*	1	0	1	1	*									

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.

Extended-Control Word (ECW)

SCSW Bits		**ERW**		
5	**14**	**Bit 7**	**ERW Bits 10-15**	**ECW Words 0-7**
0	0	0	Zeros	Unpredictable
0	1	1	Number of concurrent-sense bytes[a]	Concurrent-sense information[a]
1	0	0	Zeros	Unpredictable
1	1	0	Zeros	Model-dependent information
1	1	1	Number of concurrent-sense bytes	Concurrent-sense information

a. 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 ⋮ 15	Reserved
	0 31

Channel-Report Word (CRW)

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

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**
0 0 0 0 0 1	Available
0 0 0 0 1 0	Initialized
0 0 0 0 1 1	Temporary error
0 0 0 1 0 0	Installed parameters initialized
0 0 0 1 0 1	Terminal
0 0 0 1 1 0	Permanent error with facility not initialized
0 0 0 1 1 1	Permanent error with facility initialized
0 0 1 0 0 0	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**	
0 0 1 0	Monitoring facility	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0
0 0 1 1	Subchannel (first or only CRW)	X X X X X X X X	X X X X X X X X
0 0 1 1	Subchannel (chained CRW)	0 0 0 0 0 0 0 0	0 0 S S 0 0 0 0
0 1 0 0	Channel path	0 0 0 0 0 0 0 0	Y Y Y Y Y Y Y Y
1 0 0 1	Configuration-alert facility	0 0 0 0 0 0 0 0	Y Y Y Y Y Y Y Y
1 0 1 1	Channel subsystem	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0

S = Subchannel-set identifier (SSID) when the MSS facility is installed and the CRW is chained immediately following a CRW for a subchannel.
X = Subchannel number
Y = Channel-path ID (CHPID)

I/O Command Codes

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

x x x x	0 0 0 0	Invalid Command
mmmm	mm0 1	Write
mmmm	mm1 0	Read
0 0 0 0	0 0 1 0	— Read IPL
mmmm	mm1 1	Control
0 0 0 0	0 0 1 1	— Control no operation

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

mmmm	0 1 0 0	Sense
0 0 0 0	0 1 0 0	— Basic Sense
1 1 1 0	0 1 0 0	— Sense ID
x x x x	1 0 0 0	Transfer in channel (a)
0 0 0 0	1 0 0 0	Transfer in channel (b)
mmmm	1 0 0 0	Invalid command (c)
mmmm	1 1 0 0	Read backwards

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

Standard Meanings of Bits of First Sense Byte

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

Character Assignments

Dec	Hex	EBCDIC[1]	ISO-8[2]
0	00	NUL	NUL
1	01	SOH	SOH
2	02	STX	STX
3	03	ETX	ETX
4	04	SEL	EOT
5	05	HT	ENQ
6	06	RNL	ACK
7	07	DEL	BEL
8	08	GE	BS
9	09	SPS	HT
10	0A	RPT	LF
11	0B	VT	VT
12	0C	FF	FF
13	0D	CR	CR
14	0E	SO	SO
15	0F	SI	SI
16	10	DLE	DLE
17	11	DC1	DC1
18	12	DC2	DC2
19	13	DC3	DC3
20	14	RES/ENP	DC4
21	15	NL	NAK
22	16	BS	SYN
23	17	POC	ETB
24	18	CAN	CAN
25	19	EM	EM
26	1A	UBS	SUB
27	1B	CU1	ESC
28	1C	IFS	IFS
29	1D	IGS	IGS
30	1E	IRS	IRS
31	1F	ITB/IUS	IUS
32	20	DS	SP
33	21	SOS	!
34	22	FS	"
35	23	WUS	#
36	24	BYP/INP	$
37	25	LF	%
38	26	ETB	&
39	27	ESC	'
40	28	SA	(
41	29	SFE	)
42	2A	SM/SW	*
43	2B	CSP	+
44	2C	MFA	,
45	2D	ENQ	-
46	2E	ACK	.
47	2F	BEL	/
48	30		0
49	31		1
50	32	SYN	2
51	33	IR	3
52	24	PP	4
53	35	TRN	5
54	36	NBS	6
55	37	EOT	7
56	38	SBS	8
57	39	IT	9
58	3A	RFF	:
59	3B	CU3	;
60	3C	DC4	<
61	3D	NAK	=
62	3E		>
63	3F	SUB	?

Dec	Hex	EBCDIC[1]	ISO-8[2]
64	40	SP	@
65	41	RSP	A
66	42	â	B
67	43	ä	C
68	44	à	D
69	45	á	E
70	46	ã	F
71	47	å	G
72	48	ç	H
73	49	ñ	I
74	4A	¢	J
75	4B	.	K
76	4C	<	L
77	4D	(	M
78	4E	+	N
79	4F	|	O
80	50	&	P
81	51	é	Q
82	52	ê	R
83	53	ë	S
84	54	è	T
85	55	í	U
86	56	î	V
87	57	ï	W
88	58	ì	X
89	59	ß	Y
90	5A	!	Z
91	5B	$	[
92	5C	*	\
93	5D	)	]
94	5E	;	^
95	5F	¬	_
96	60	-	`
97	61	/	a
98	62	Â	b
99	63	Ä	c
100	64	À	d
101	65	Á	e
102	66	Ã	f
103	67	Å	g
104	68	Ç	h
105	69	Ñ	i
106	6A	¦	j
107	6B	,	k
108	6C	%	l
109	6D	_	m
110	6E	>	n
111	6F	?	o
112	70	ø	p
113	71	É	q
114	72	Ê	r
115	73	Ë	s
116	74	È	t
117	75	Í	u
118	76	Î	v
119	77	Ï	w
120	78	Ì	x
121	79	`	y
122	7A	:	z
123	7B	#	{
124	7C	@	|
125	7D	'	}
126	7E	=	~
127	7F	"	•

Dec	Hex	EBCDIC[1]	ISO-8[2]
128	80	Ø	
129	81	a	
130	82	b	BPH
131	83	c	NBH
132	84	d	IND
133	85	e	NEL
134	86	f	SSA
135	87	g	ESA
136	88	h	HTS
137	89	i	HTJ
138	8A	«	VTS
139	8B	»	PLD
140	8C	ð	PLU
141	8D	ý	RI
142	8E	þ	SS2
143	8F	±	SS3
144	90	°	DCS
145	91	j	PU1
146	92	k	PU2
147	93	l	STS
148	94	m	CCH
149	95	n	MW
150	96	o	SPA
151	97	p	EPA
152	98	q	SOS
153	99	r	
154	9A	ª	SCI
155	9B	º	CSI
156	9C	æ	ST
157	9D	¸	OSC
158	9E	Æ	PM
159	9F	¤	APC
160	A0	µ	RSP
161	A1	~	¡
162	A2	s	¢
163	A3	t	£
164	A4	u	¤
165	A5	v	¥
166	A6	w	¦
167	A7	x	§
168	A8	y	¨
169	A9	z	©
170	AA	¡	ª
171	AB	¿	«
172	AC	Ð	¬
173	AD	Ý	SHY
174	AE	þ	®
175	AF	®	¯
176	B0	^	°
177	B1	£	±
178	B2	¥	²
179	B3	·	³
180	B4	©	´
181	B5	§	µ
182	B6	¶	¶
183	B7	¼	·
184	B8	½	¸
185	B9	¾	¹
186	BA	[	º
187	BB	]	»
188	BC	ä	¼
189	BD	¨	½
190	BE	´	¾
191	BF	×	¿

Dec	Hex	EBCDIC[1]	ISO-8[2]
192	C0	{	À
193	C1	A	Á
194	C2	B	Â
195	C3	C	Ã
196	C4	D	Ä
197	C5	E	Å
198	C6	F	Æ
199	C7	G	Ç
200	C8	H	È
201	C9	I	É
202	CA	SHY	Ê
203	CB	ô	Ë
204	CC	ö	Ì
205	CD	ò	Í
206	CE	ó	Î
207	CF	õ	Ï
208	D0	}	Ð
209	D1	J	Ñ
210	D2	K	Ò
211	D3	L	Ó
212	D4	M	Ô
213	D5	N	Õ
214	D6	O	Ö
215	D7	P	×
216	D8	Q	Ø
217	D9	R	Ù
218	DA	¹	Ú
219	DB	û	Û
220	DC	ü	Ü
221	DD	ù	Ý
222	DE	ú	Þ
223	DF	ÿ	ß
224	E0	\	à
225	E1	÷	á
226	E2	S	â
227	E3	T	ã
228	E4	U	ä
229	E5	V	å
230	E6	W	æ
231	E7	X	ç
232	E8	Y	è
233	E9	Z	é
234	EA	²	ê
235	EB	Ô	ë
236	EC	Ö	ì
237	ED	Ò	í
238	EE	Ó	î
239	EF	Õ	ï
240	F0	0	ð
241	F1	1	ñ
242	F2	2	ò
243	F3	3	ó
244	F4	4	ô
245	F5	5	õ
246	F6	6	ö
247	F7	7	÷
248	F8	8	ø
249	F9	9	ù
250	FA	³	ú
251	FB	Û	û
252	FC	Ü	ü
253	FD	Ù	ý
254	FE	Ú	þ
255	FF	EO	ÿ

Notes:

1 The EBCDIC characters are based on code page 037.

2 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.

Control Character Representations

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

Additional ISO-8 Control Character Representations

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

Formatting Character Representations

NSP	Numeric Space
RSP	Required Space
SP	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
20	Digit selector
21	Start of significance
22	Field separator
40	Blank
4B	Period

Code (Hex)	Meaning
5B	Dollar sign
5C	Asterisk
6B	Comma
C3D9	CR (credit)
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.

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

Powers of 2 and 16

m	*n*	2^m and 16^n	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	

m	*n*	2^m and 16^n	Symbol
64	16	18 446 744 073 709 551 616	
65		36 893 488 147 419 103 232	
66		73 786 976 294 838 206 464	
67		147 573 952 589 676 412 928	
68	17	295 147 905 179 352 825 856	
69		590 295 810 358 705 651 712	
70		1 180 591 620 717 411 303 424	Z (zetta)
71		2 361 183 241 434 822 606 848	
72	18	4 722 366 482 869 645 213 696	
73		9 444 732 965 739 290 427 392	
74		18 889 465 931 478 580 854 784	
75		37 778 931 862 957 161 709 568	
76	19	75 557 863 725 914 323 419 136	
77		151 115 727 451 828 646 838 272	
78		302 231 454 903 657 293 676 544	
79		604 462 909 807 314 587 353 088	
80	20	1 208 925 819 614 629 174 706 176	Y (yotta)
81		2 417 851 639 229 258 349 412 352	
82		4 835 703 278 458 516 698 824 704	
83		9 671 406 556 917 033 397 649 408	
84	21	19 342 813 113 834 066 795 298 816	
85		38 685 626 227 668 133 590 597 632	
86		77 371 252 455 336 267 181 195 264	
87		154 742 504 910 672 534 362 390 528	
88	22	309 485 009 821 345 068 724 781 056	
89		618 970 019 642 690 137 449 562 112	
90		1 237 940 039 285 380 274 899 124 224	(see note)
91		2 475 880 078 570 760 549 798 248 448	
92	23	4 951 760 157 141 521 099 596 496 896	
93		9 903 520 314 283 042 199 192 993 792	
94		19 807 040 628 566 084 398 385 987 584	
95		39 614 081 257 132 168 796 771 975 168	
96	24	79 228 162 514 264 337 593 543 950 336	
97		158 456 325 028 528 675 187 087 900 672	
98		316 912 650 057 057 350 374 175 801 344	
99		633 825 300 114 114 700 748 351 602 688	
100	25	1 267 650 600 228 229 401 496 703 205 376	(see note)
101		2 535 301 200 456 458 802 993 406 410 752	
102		5 070 602 400 912 917 605 986 812 821 504	
103		10 141 204 801 825 835 211 973 625 643 008	
104	26	20 282 409 603 651 670 423 947 251 286 016	
105		40 564 819 207 303 340 847 894 502 572 032	
106		81 129 638 414 606 681 695 789 005 144 064	
107		162 259 276 829 213 363 391 578 010 288 128	
108	27	324 518 553 658 426 726 783 156 020 576 256	
109		649 037 107 316 853 453 566 312 041 152 512	
110		1 298 074 214 633 706 907 132 624 082 305 024	(see note)
111		2 596 148 429 267 413 814 265 248 164 610 048	
112	28	5 192 296 858 534 827 628 530 496 329 220 096	
113		10 384 593 717 069 655 257 060 992 658 440 192	
114		20 769 187 434 139 310 514 121 985 316 880 384	
115		41 538 374 868 278 621 028 243 970 633 760 768	
116	29	83 076 749 736 557 242 056 487 941 267 521 536	
117		166 153 499 473 114 484 112 975 882 535 043 072	
118		332 306 998 946 228 968 225 951 765 070 086 144	
119		664 613 997 892 457 936 451 903 530 140 172 288	
120	30	1 329 227 995 784 915 872 903 807 060 280 344 576	(see note)
121		2 658 455 991 569 831 745 807 614 120 560 689 152	
122		5 316 911 983 139 663 491 615 228 241 121 378 304	
123		10 633 823 966 279 326 983 230 456 482 242 756 608	
124	31	21 267 647 932 558 653 966 460 912 964 485 513 216	
125		42 535 295 865 117 307 932 921 825 928 971 026 432	
126		85 070 591 730 234 615 865 843 651 857 942 052 864	
127		170 141 183 460 469 231 731 687 303 715 884 105 728	
128	32	340 282 366 920 938 463 463 374 607 431 768 211 456	

Note: No Système international d'unités (SI) symbols greater than Y (yotta) are defined.

File Number: S-390-00

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

SA22-7871-04