z/Architecture

# Reference Summary

SA22-7871-00

z/Architecture

# Reference Summary

SA22-7871-00

**First Edition (September 2001)**

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

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

Please direct any comments on the contents of this publication to

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

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

# Preface

This publication is intended primarily for use by 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:

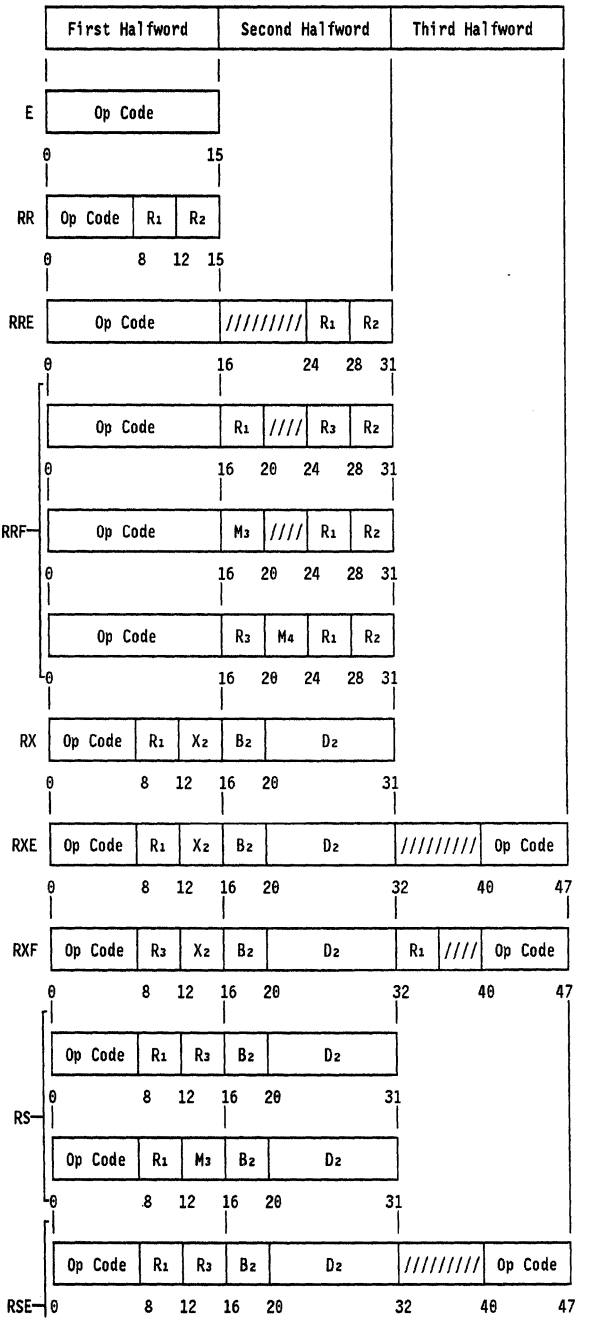
| **Facility** | **Instruction** |
|---|---|
| Expanded storage | PGIN, PGOUT |
| Extended translation 2 | CLCLU, MVCLU, PKA, PKU, TP, TROO, TROT, TRTO, TRTT, UNPKA, UNPKU |

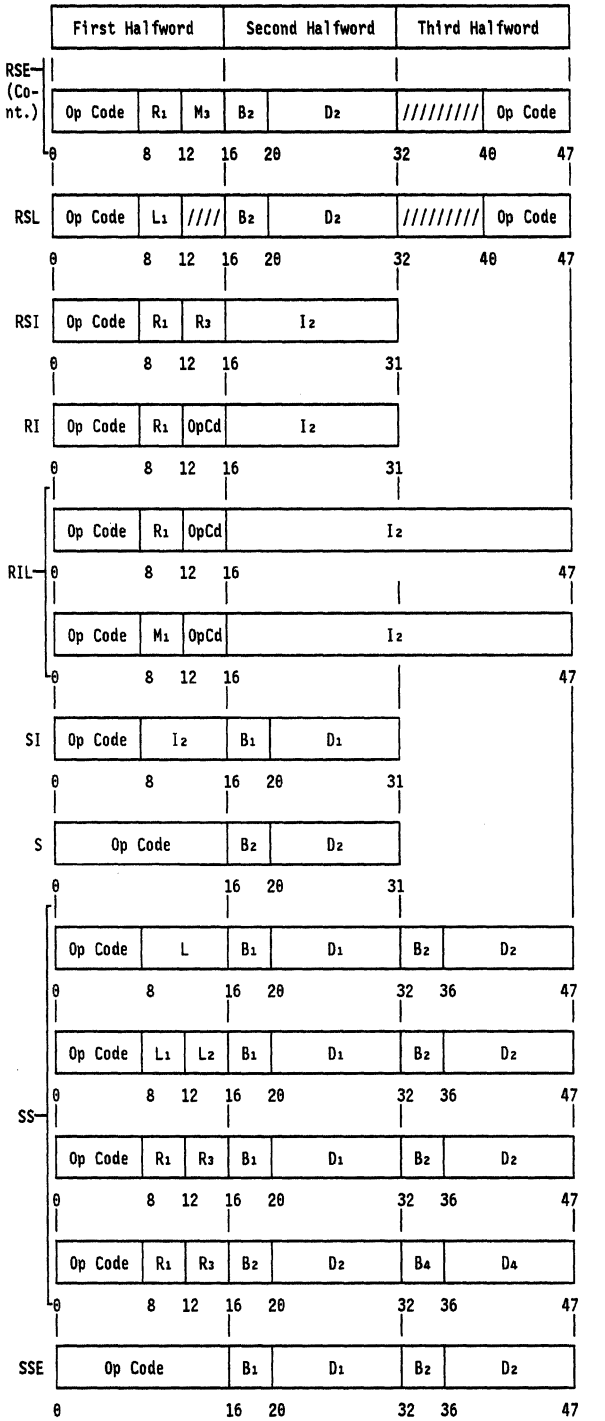
For information about Enterprise Systems Architecture/390™ (ESA/390™) architecture, refer to *IBM Enterprise Systems Architecture/390 Principles of Operation*, SA22-7201, and *IBM Enterprise Systems Architecture/390 Reference Summary*, SA22-7209.

# Contents

## NOTES

# Machine Instruction Formats




First Halfword
Second Halfword
Third Halfword
RSE (Cont.)
Op Code
R1
M3
B2
D2
////////
Op Code
0
8
12
16
20
32
40
47
RSL
Op Code
L1
////
B2
D2
////////
Op Code
RSI
Op Code
R1
R3
I2
31
RI
Op Code
R1
OpCd
I2
RIL
Op Code
R1
OpCd
I2
Op Code
M1
OpCd
I2
SI
Op Code
I2
B1
D1
S
Op Code
B2
D2
SS
Op Code
L
B1
D1
B2
D2
Op Code
L1
L2
B1
D1
B2
D2
Op Code
R1
R3
B1
D1
B2
D2
Op Code
R1
R3
B2
D2
B4
D4
SSE
Op Code
B1
D1
B2
D2
36

| | |
|---|---|
| 1, 2, 3, 4: | Denotes association with first, second, third, or fourth operand |
| $B_1$, $B_2$, $B_4$: | Base register designation field |
| $D_1$, $D_2$, $D_4$: | Displacement field |
| $I_2$: | 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 | Op Code | 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 |
| 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 |
| AG | $R_1,D_2(X_2,B_2)$ | Add (64) | RXE | E308 | c N |
| AGF | $R_1,D_2(X_2,B_2)$ | Add (64 < 32) | RXE | E318 | c N |
| AGFR | $R_1,R_2$ | Add (64 < 32) | RRE | B918 | c N |
| AGHI | $R_1,I_2$ | Add Halfword Immediate | RI | A7B | c N |
| AGR | $R_1,R_2$ | Add (64) | RRE | B908 | c N |
| AH | $R_1,D_2(X_2,B_2)$ | Add Halfword | RX | 4A | c |
| AHI | $R_1,I_2$ | Add Halfword Immediate (32) | RI | A7A | c |
| 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) | RXE | E398 | c N3 |
| ALCG | $R_1,D_2(X_2,B_2)$ | Add Logical with Carry (64) | RXE | 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 |
| ALG | $R_1,D_2(X_2,B_2)$ | Add Logical (64) | RXE | E30A | c N |
| ALGF | $R_1,D_2(X_2,B_2)$ | Add Logical (64 < 32) | RXE | E31A | c N |
| 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 |
| ALR | $R_1,R_2$ | Add Logical (32) | RR | 1E | c |
| AP | $D_1(L_1,B_1),D_2(L_2,B_2)$ | Add Decimal | SS | FA | c |
| AR | $R_1,R_2$ | Add (32) | RR | 1A | c |
| AU | $R_1,D_2(X_2,B_2)$ | Add Unnormalized (SH) | RX | 7E | c |
| AUR | $R_1,R_2$ | Add Unnormalized (SH) | RR | 3E | c |
| AW | $R_1,D_2(X_2,B_2)$ | Add Unnormalized (LH) | RX | 6E | c |
| AWR | $R_1,R_2$ | Add Unnormalized (LH) | RR | 2E | c |
| AXBR | $R_1,R_2$ | Add (EB) | RRE | B34A | c |
| AXR | $R_1,R_2$ | Add Normalized (EH) | RR | 36 | c |
| BAKR | $R_1,R_2$ | Branch and Stack | RRE | B240 | q |
| BAL | $R_1,D_2(X_2,B_2)$ | Branch and Link | RX | 45 | |
| BALR | $R_1,R_2$ | Branch and Link | RR | 05 | |
| BAS | $R_1,D_2(X_2,B_2)$ | Branch and Save | RX | 4D | |
| BASR | $R_1,R_2$ | Branch and Save | RR | 0D | |
| BASSM | $R_1,R_2$ | Branch and Save and Set Mode | RR | 0C | |
| BC | $M_1,D_2(X_2,B_2)$ | Branch on Condition | RX | 47 | |
| BCR | $M_1,R_2$ | Branch on Condition | RR | 07 | |
| BCT | $R_1,D_2(X_2,B_2)$ | Branch on Count (32) | RX | 46 | |
| BCTG | $R_1,D_2(X_2,B_2)$ | Branch on Count (64) | RXE | E346 | N |
| BCTGR | $R_1,R_2$ | Branch on Count (64) | RRE | B946 | N |
| BCTR | $R_1,R_2$ | Branch on Count (32) | RR | 06 | |
| BRAS | $R_1,I_2$ | Branch Relative and Save | RI | A75 | |
| BRASL | $R_1,I_2$ | Branch Relative and Save Long | RIL | C05 | N3 |
| BRC | $M_1,I_2$ | Branch Relative on Condition | RI | A74 | |
| BRCL | $M_1,I_2$ | Branch Relative on Condition Long | RIL | C04 | N3 |
| BRCT | $R_1,I_2$ | Branch Relative on Count (32) | RI | A76 | |
| BRCTG | $R_1,I_2$ | Branch Relative on Count (64) | RI | A77 | N |
| BRXH | $R_1,R_3,I_2$ | Branch Relative on Index High (32) | RSI | 84 | |
| BRXHG | $R_1,R_3,I_2$ | Branch Relative on Index High (64) | RIE | EC44 | N |
| BRXLE | $R_1,R_3,I_2$ | Branch Relative on Index Low or Equal (32) | RSI | 85 | |
| BRXLG | $R_1,R_3,I_2$ | Branch Relative on Index Low or Equal (64) | RIE | EC45 | N |
| BSA | $R_1,R_2$ | Branch and Set Authority | RRE | B25A | q |
| BSG | $R_1,R_2$ | Branch in Subspace Group | RRE | B258 | |
| BSM | $R_1,R_2$ | Branch and Set Mode | RR | 0B | |
| BXH | $R_1,R_3,D_2(B_2)$ | Branch on Index High (32) | RS | 86 | |
| BXHG | $R_1,R_3,D_2(B_2)$ | Branch on Index High (64) | RSE | EB44 | N |
| BXLE | $R_1,R_3,D_2(B_2)$ | Branch on Index Low or Equal (32) | RS | 87 | |

| Mnemonic | Operands | Name | Format | Op Code | Notes |
|---|---|---|---|---|---|
| BXLEG | $R_1,R_3,D_2(B_2)$ | Branch on Index Low or Equal (64) | RSE | EB45 | N |
| C | $R_1,D_2(X_2,B_2)$ | Compare (32) | RX | 59 | c |
| CD | $R_1,D_2(X_2,B_2)$ | Compare (LH) | RX | 69 | c |
| CDB | $R_1,D_2(X_2,B_2)$ | Compare (LB) | RXE | ED19 | c |
| CDBR | $R_1,R_2$ | Compare (LB) | RRE | B319 | c |
| CDFBR | $R_1,R_2$ | Convert from Fixed (LB < 32) | RRE | B395 | |
| CDFR | $R_1,R_2$ | Convert from Fixed (LH < 32) | RRE | B3B5 | |
| CDGBR | $R_1,R_2$ | Convert from Fixed (LB < 64) | RRE | B3A5 | N |
| CDGR | $R_1,R_2$ | Convert from Fixed (LB < 64) | RRE | B3C5 | N |
| CDR | $R_1,R_2$ | Compare (LH) | RR | 29 | c |
| CDS | $R_1,R_3,D_2(B_2)$ | Compare Double and Swap (32) | RS | BB | c |
| CDSG | $R_1,R_3,D_2(B_2)$ | Compare Double and Swap (64) | RSE | EB3E | c N |
| CEB | $R_1,D_2(X_2,B_2)$ | Compare (SB) | RXE | ED09 | c |
| CE | $R_1,D_2(X_2,B_2)$ | Compare (SH) | RX | 79 | c |
| CEBR | $R_1,R_2$ | Compare (SB) | RRE | B309 | c |
| CEFBR | $R_1,R_2$ | Convert from Fixed (SB < 32) | RRE | B394 | |
| CEFR | $R_1,R_2$ | Convert from Fixed (SH < 32) | RRE | B3B4 | |
| CEGBR | $R_1,R_2$ | Convert from Fixed (SB < 64) | RRE | B3A4 | N |
| CEGR | $R_1,R_2$ | Convert from Fixed (SH < 64) | RRE | B3C4 | N |
| CER | $R_1,R_2$ | Compare (SH) | RR | 39 | c |
| CFC | $D_2(B_2)$ | Compare and Form Codeword | S | B21A | ic |
| CFDBR | $R_1,M_3,R_2$ | Convert to Fixed (32 < LB) | RRF | B399 | c |
| CFDR | $R_1,M_3,R_2$ | Convert to Fixed (32 < LH) | RRF | B3B9 | c |
| CFEBR | $R_1,M_3,R_2$ | Convert to Fixed (32 < SB) | RRF | B398 | c |
| CFER | $R_1,M_3,R_2$ | Convert to Fixed (32 < SH) | RRF | B3B8 | c |
| CFXBR | $R_1,M_3,R_2$ | Convert to Fixed (32 < EB) | RRF | B39A | c |
| CFXR | $R_1,M_3,R_2$ | Convert to Fixed (32 < EH) | RRF | B3BA | c |
| CG | $R_1,D_2(X_2,B_2)$ | Compare (64) | RXE | E320 | c N |
| CGDBR | $R_1,M_3,R_2$ | Convert to Fixed (64 < LB) | RRF | B3A9 | c N |
| CGDR | $R_1,M_3,R_2$ | Convert to Fixed (64 < LH) | RRF | B3C9 | c N |
| CGEBR | $R_1,M_3,R_2$ | Convert to Fixed (64 < SB) | RRF | B3A8 | c N |
| CGER | $R_1,M_3,R_2$ | Convert to Fixed (64 < SH) | RRF | B3C8 | c N |
| CGF | $R_1,D_2(X_2,B_2)$ | Compare (64 < 32) | RXE | E330 | c N |
| CGFR | $R_1,R_2$ | Compare (64 < 32) | RRE | B930 | c N |
| CGHI | $R_1,I_2$ | Compare Halfword Immediate (64) | RI | A7F | c N |
| CGR | $R_1,R_2$ | Compare (64) | RRE | B920 | c N |
| CGXBR | $R_1,M_3,R_2$ | Convert to Fixed (64 < EB) | RRF | B3AA | c N |
| CGXR | $R_1,M_3,R_2$ | Convert to Fixed (64 < EH) | RRF | B3CA | c N |
| CH | $R_1,D_2(X_2,B_2)$ | Compare Halfword | RX | 49 | c |
| CHI | $R_1,I_2$ | Compare Halfword Immediate (32) | RI | A7E | c |
| 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 | D5 | c |
| CLCL | $R_1,R_2$ | Compare Logical Long | RR | 0F | ic |
| CLCLE | $R_1,R_3,D_2(B_2)$ | Compare Logical Long Extended | RS | A9 | c |
| CLCLU | $R_1,R_3,D_2(B_2)$ | Compare Logical Long Unicode | RSE | EB8F | c E2 |
| CLG | $R_1,D_2(X_2,B_2)$ | Compare Logical (64) | RXE | E321 | c N |
| CLGF | $R_1,D_2(X_2,B_2)$ | Compare Logical (64 < 32) | RXE | E331 | c N |
| CLGFR | $R_1,R_2$ | Compare Logical (64 < 32) | RRE | B931 | c N |
| CLGR | $R_1,R_2$ | Compare Logical (64) | RRE | B921 | c N |
| CLI | $D_1(B_1),I_2$ | Compare Logical (immediate) | SI | 95 | c |
| CLM | $R_1,M_3,D_2(B_2)$ | Compare Logical Characters under Mask | RS | BD | c |
| CLMH | $R_1,M_3,D_2(B_2)$ | Compare Logical Characters under Mask | RSE | EB20 | c N |
| CLR | $R_1,R_2$ | Compare Logical (32) | RR | 15 | c |
| CLST | $R_1,R_2$ | Compare Logical String | RRE | B25D | c |
| CMPSC | $R_1,R_2$ | Compression Call | RRE | B263 | ic |
| CP | $D_1(L_1,B_1),D_2(L_2,B_2)$ | Compare Decimal | SS | F9 | c |
| CPYA | $R_1,R_2$ | Copy Access | RRE | B24D | |
| CR | $R_1,R_2$ | Compare (32) | RR | 19 | c |
| CS | $R_1,R_3,D_2(B_2)$ | Compare and Swap (32) | RS | BA | c |
| CSCH | | Clear Subchannel | S | B230 | pc |
| CSG | $R_1,R_3,D_2(B_2)$ | Compare and Swap (64) | RSE | EB30 | c N |
| CSP | $R_1,R_2$ | Compare and Swap and Purge | RRE | B250 | pc |
| CUSE | $R_1,R_2$ | Compare until Substring Equal | RRE | B257 | ic |
| CUTFU | $R_1,R_2$ | Convert UTF-8 to Unicode | RRE | B2A7 | c |
| CUUTF | $R_1,R_2$ | Convert Unicode to UTF-8 | RRE | B2A6 | c |
| CVB | $R_1,D_2(X_2,B_2)$ | Convert to Binary (32) | RX | 4F | |
| CVBG | $R_1,D_2(X_2,B_2)$ | Convert to Binary (64) | RXE | E30E | N |
| CVD | $R_1,D_2(X_2,B_2)$ | Convert to Decimal (32) | RX | 4E | |

| Mnemonic | Operands | Name | Format | Op Code | Notes |
|---|---|---|---|---|---|
| CVDG | $R_1,D_2(X_2,B_2)$ | Convert to Decimal (64) | RXE | E32E | N |
| CXBR | $R_1,R_2$ | Compare (EB) | RRE | B349 | c |
| CXFBR | $R_1,R_2$ | Convert from Fixed (EB < 32) | RRE | B396 | |
| CXFR | $R_1,R_2$ | Convert from Fixed (EH < 32) | RRE | B3B6 | |
| CXGBR | $R_1,R_2$ | Convert from Fixed (EB < 64) | RRE | B3A6 | N |
| CXGR | $R_1,R_2$ | Convert from Fixed (EH < 64) | RRE | B3C6 | N |
| CXR | $R_1,R_2$ | Compare (EH) | RRE | B369 | c |
| D | $R_1,D_2(X_2,B_2)$ | Divide (32 < 64) | RX | 5D | |
| DD | $R_1,D_2(X_2,B_2)$ | Divide (LH) | RX | 6D | |
| DDB | $R_1,D_2(X_2,B_2)$ | Divide (LB) | RXE | ED1D | |
| DDBR | $R_1,R_2$ | Divide (LB) | RRE | B31D | |
| DDR | $R_1,R_2$ | Divide (LH) | RR | 2D | |
| DE | $R_1,D_2(X_2,B_2)$ | Divide (SH) | RX | 7D | |
| DEB | $R_1,D_2(X_2,B_2)$ | Divide (SB) | RXE | ED0D | |
| DEBR | $R_1,R_2$ | Divide (SB) | RRE | B30D | |
| DER | $R_1,R_2$ | Divide (SH) | RR | 3D | |
| DIDBR | $R_1,R_3,R_2,M_4$ | Divide to Integer (LB) | RRF | B35B | c |
| DIEBR | $R_1,R_3,R_2,M_4$ | Divide to Integer (SB) | RRF | B353 | c |
| DL | $R_1,D_2(X_2,B_2)$ | Divide Logical (32 < 64) | RXE | E397 | N3 |
| DLG | $R_1,D_2(X_2,B_2)$ | Divide Logical (64 < 128) | RXE | 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 | FD | |
| DR | $R_1,R_2$ | Divide | RR | 1D | |
| DSG | $R_1,D_2(X_2,B_2)$ | Divide Single (64) | RXE | E30D | N |
| DSGF | $R_1,D_2(X_2,B_2)$ | Divide Single (64 < 32) | RXE | E31D | N |
| DSGFR | $R_1,R_2$ | Divide Single (64 < 32) | RRE | B91D | N |
| DSGR | $R_1,R_2$ | Divide Single (64) | RRE | B90D | N |
| DXBR | $R_1,R_2$ | Divide (EB) | RRE | B34D | |
| DXR | $R_1,R_2$ | Divide (EH) | RRE | B22D | |
| EAR | $R_1,R_2$ | Extract Access | RRE | B24F | |
| ED | $D_1(L,B_1),D_2(B_2)$ | Edit | SS | DE | c |
| EDMK | $D_1(L,B_1),D_2(B_2)$ | Edit and Mark | SS | DF | c |
| EFPC | $R_1$ | Extract FPC | RRE | B38C | |
| EPAR | $R_1$ | Extract Primary ASN | RRE | B226 | q |
| EPSW | $R_1,R_2$ | Extract PSW | RRE | B98D | N3 |
| EREG | $R_1,R_2$ | Extract Stacked Registers (32) | RRE | B249 | |
| EREGG | $R_1,R_2$ | Extract Stacked Registers (64) | RRE | B90E | N |
| ESAR | $R_1$ | Extract Secondary ASN | RRE | B227 | q |
| ESEA | $R_1,R_2$ | Extract and Set Extended Authority | RRE | B99D | p N |
| ESTA | $R_1,R_2$ | Extract Stacked State | RRE | B24A | c |
| EX | $R_1,D_2(X_2,B_2)$ | Execute | RX | 44 | |
| FIDBR | $R_1,M_3,R_2$ | Load FP Integer (LB) | RRF | B35F | |
| FIDR | $R_1,R_2$ | Load FP Integer (LH) | RRE | B37F | |
| FIEBR | $R_1,M_3,R_2$ | Load FP Integer (SB) | RRF | B357 | |
| FIER | $R_1,R_2$ | Load FP Integer (SH) | RRE | B377 | |
| FIXBR | $R_1,M_3,R_2$ | Load FP Integer (EB) | RRF | B347 | |
| FIXR | $R_1,R_2$ | Load FP Integer (EH) | RRE | B367 | |
| 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 | BF | c |
| ICMH | $R_1,M_3,D_2(B_2)$ | Insert Characters under Mask (high) | RSE | EB80 | c N |
| IIHH | $R_1,I_2$ | Insert Immediate (high high) | RI | A50 | N |
| IIHL | $R_1,I_2$ | Insert Immediate (high low) | RI | A51 | N |
| IILH | $R_1,I_2$ | Insert Immediate (low high) | RI | A52 | N |
| IILL | $R_1,I_2$ | Insert Immediate (low low) | RI | A53 | N |
| IPK | | Insert PSW Key | S | B20B | q |
| IPM | $R_1$ | Insert Program Mask | RRE | B222 | |
| IPTE | $R_1,R_2$ | Invalidate Page Table Entry | RRE | B221 | p |
| ISKE | $R_1,R_2$ | Insert Storage Key Extended | RRE | B229 | p |
| IVSK | $R_1,R_2$ | Insert Virtual Storage Key | RRE | B223 | q |
| KDB | $R_1,D_2(X_2,B_2)$ | Compare and Signal (LB) | RXE | ED18 | c |
| KDBR | $R_1,R_2$ | Compare and Signal (LB) | RRE | B318 | c |
| KEB | $R_1,D_2(X_2,B_2)$ | Compare and Signal (SB) | RXE | ED08 | c |
| KEBR | $R_1,R_2$ | Compare and Signal (SB) | RRE | B308 | c |
| KXBR | $R_1,R_2$ | Compare and Signal (EB) | RRE | B348 | c |
| L | $R_1,D_2(X_2,B_2)$ | Load (32) | RX | 58 | |
| LA | $R_1,D_2(X_2,B_2)$ | Load Address | RX | 41 | |
| LAE | $R_1,D_2(X_2,B_2)$ | Load Address Extended | RX | 51 | |
| LAM | $R_1,R_3,D_2(B_2)$ | Load Access Multiple | RS | 9A | |
| LARL | $R_1,I_2$ | Load Address Relative Long | RIL | C00 | N3 |
| LASP | $D_1(B_1),D_2(B_2)$ | Load Address Space Parameters | SSE | E500 | pc |
| LCDBR | $R_1,R_2$ | Load Complement (LB) | RRE | B313 | c |
| LCDR | $R_1,R_2$ | Load Complement (LH) | RR | 23 | c |

| Mnemonic | Operands | Name | Format | Op Code | Notes |
|---|---|---|---|---|---|
| LCEBR | $R_1,R_2$ | Load Complement (SB) | RRE | B303 | c |
| LCER | $R_1,R_2$ | Load Complement (S) | RR | 33 | c |
| LCGFR | $R_1,R_2$ | Load Complement (64<32) | RRE | B913 | c N |
| LCGR | $R_1,R_2$ | Load Complement (64) | RRE | B903 | c N |
| LCR | $R_1,R_2$ | Load Complement (32) | RR | 13 | c |
| LCTL | $R_1,R_3,D_2(B_2)$ | Load Control (32) | RS | B7 | p |
| LCTLG | $R_1,R_3,D_2(B_2)$ | Load Control (64) | RSE | EB2F | p N |
| LCXBR | $R_1,R_2$ | Load Complement (EB) | RRE | B343 | c |
| LCXR | $R_1,R_2$ | Load Complement (EH) | RRE | B363 | c |
| LD | $R_1,D_2(X_2,B_2)$ | Load (L) | RX | 68 | |
| LDE | $R_1,D_2(X_2,B_2)$ | Load Lengthened (LH<SH) | RXE | ED24 | |
| LDEB | $R_1,D_2(X_2,B_2)$ | Load Lengthened (LB<SB) | RXE | ED04 | |
| LDEBR | $R_1,R_2$ | Load Lengthened (LB<SB) | RRE | B304 | |
| LDER | $R_1,R_2$ | Load Lengthened (LH<SH) | RRE | B324 | |
| LDR | $R_1,R_2$ | Load (L) | RR | 28 | |
| LDXBR | $R_1,R_2$ | Load Rounded (LB<EB) | RRE | B345 | |
| LDXR | $R_1,R_2$ | Load Rounded (LH<EH) | RR | 25 | |
| LE | $R_1,D_2(X_2,B_2)$ | Load (S) | RX | 78 | |
| LEDBR | $R_1,R_2$ | Load Rounded (SB<LB) | RRE | B344 | |
| LEDR | $R_1,R_2$ | Load Rounded (SH<LH) | RR | 35 | |
| LER | $R_1,R_2$ | Load (S) | RR | 38 | |
| LEXBR | $R_1,R_2$ | Load Rounded (SB<EB) | RRE | B346 | |
| LEXR | $R_1,R_2$ | Load Rounded (SH<EH) | RRE | B366 | |
| LFPC | $D_2(B_2)$ | Load FPC | S | B29D | |
| LG | $R_1,D_2(X_2,B_2)$ | Load (64) | RXE | E304 | N |
| LGF | $R_1,D_2(X_2,B_2)$ | Load (64<32) | RXE | E314 | N |
| LGFR | $R_1,R_2$ | Load (64<32) | RRE | B914 | N |
| LGH | $R_1,D_2(X_2,B_2)$ | Load Halfword | RXE | E315 | N |
| LGHI | $R_1,I_2$ | Load Halfword Immediate | RI | A79 | N |
| LGR | $R_1,R_2$ | Load (64) | RRE | B904 | N |
| LH | $R_1,D_2(X_2,B_2)$ | Load Halfword (32) | RX | 48 | |
| LHI | $R_1,I_2$ | Load Halfword Immediate (32) | RI | A78 | |
| LLGC | $R_1,D_2(X_2,B_2)$ | Load Logical Character | RXE | E390 | N |
| LLGF | $R_1,D_2(X_2,B_2)$ | Load Logical (64<32) | RXE | E316 | N |
| LLGFR | $R_1,R_2$ | Load Logical (64<32) | RRE | B916 | N |
| LLGH | $R_1,D_2(X_2,B_2)$ | Load Logical Halfword | RXE | E391 | N |
| LLGT | $R_1,D_2(X_2,B_2)$ | Load Logical Thirty One Bits | RXE | E317 | N |
| LLGTR | $R_1,R_2$ | Load Logical Thirty One Bits | RRE | B917 | N |
| LLIHH | $R_1,I_2$ | Load Logical Immediate (high high) | RI | A5C | N |
| LLIHL | $R_1,I_2$ | Load Logical Immediate (high low) | RI | A5D | N |
| LLILH | $R_1,I_2$ | Load Logical Immediate (low high) | RI | A5E | N |
| LLILL | $R_1,I_2$ | Load Logical Immediate (low low) | RI | A5F | N |
| LM | $R_1,R_3D_2(B_2)$ | Load Multiple (32) | RS | 98 | |
| LMD | $R_1,R_3,D_2(B_2),D_4(B_4)$ | Load Multiple Disjoint | SS | EF | N |
| LMG | $R_1,R_3D_2(B_2)$ | Load Multiple (64) | RSE | EB04 | N |
| LMH | $R_1,R_3D_2(B_2)$ | Load Multiple High | RSE | EB96 | N |
| LNDBR | $R_1,R_2$ | Load Negative (LB) | RRE | B311 | c |
| LNDR | $R_1,R_2$ | Load Negative (LH) | RR | 21 | c |
| LNEBR | $R_1,R_2$ | Load Negative (SB) | RRE | B301 | c |
| 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 |
| 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 | RXE | E38F | N |
| LPR | $R_1,R_2$ | Load Positive (32) | RR | 10 | c |
| LPSW | $D_2(B_2)$ | Load PSW | S | 82 | pn |
| LPSWE | $D_2(B_2)$ | Load PSW Extended | S | B2B2 | pn N |
| LPXBR | $R_1,R_2$ | Load Positive (EB) | RRE | B340 | c |
| LPXR | $R_1,R_2$ | Load Positive (EH) | RRE | B360 | c |
| LR | $R_1,R_2$ | Load (32) | RR | 18 | |
| LRA | $R_1,D_2(X_2,B_2)$ | Load Real Address (32) | RX | B1 | pc |
| LRAG | $R_1,D_2(X_2,B_2)$ | Load Real Address (64) | RXE | E303 | pc N |
| LRDR | $R_1,R_2$ | Load Rounded (LH<EH) | RR | 25 | |
| LRER | $R_1,R_2$ | Load Rounded (SH<LH) | RR | 35 | |
| LRV | $R_1,D_2(X_2,B_2)$ | Load Reversed (32) | RXE | E31E | N3 |
| LRVG | $R_1,D_2(X_2,B_2)$ | Load Reversed (64) | RXE | 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) | RXE | E31F | N3 |
| LRVR | $R_1,R_2$ | Load Reversed (32) | RRE | B91F | N3 |
| LTDBR | $R_1,R_2$ | Load and Test (LB) | RRE | B312 | c |
| LTDR | $R_1,R_2$ | Load and Test (LH) | RR | 22 | c |
| LTEBR | $R_1,R_2$ | Load and Test (SB) | RRE | B302 | c |
| LTER | $R_1,R_2$ | Load and Test (SH) | RR | 32 | c |
| LTGFR | $R_1,R_2$ | Load and Test (64<32) | RRE | B912 | c N |

| Mnemonic | Operands | Name | Format | Op Code | Notes |
|---|---|---|---|---|---|
| LTGR | $R_1,R_2$ | Load and Test (64) | RRE | B902 | c N |
| LTR | $R_1,R_2$ | Load and Test (32) | RR | 12 | c |
| LTXBR | $R_1,R_2$ | Load and Test (EB) | RRE | B342 | c |
| LTXR | $R_1,R_2$ | Load and Test (EH) | RRE | B362 | c |
| LURA | $R_1,R_2$ | Load Using Real Address (32) | RRE | B24B | p |
| LURAG | $R_1,R_2$ | Load Using Real Address (64) | RRE | B905 | 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 | |
| LXE | $R_1,D_2(X_2,B_2)$ | Load Lengthened (EH < SH) | RXE | ED26 | |
| LXEB | $R_1,D_2(X_2,B_2)$ | Load Lengthened (EB < SB) | RXE | ED06 | |
| LXEBR | $R_1,R_2$ | Load Lengthened (EB < SB) | RRE | B306 | |
| LXER | $R_1,R_2$ | Load Lengthened (EH < SH) | RRE | B326 | |
| LXR | $R_1,R_2$ | Load (E) | RRE | B365 | |
| 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 | |
| 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 | B31E | |
| 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 | B30E | |
| MC | $D_1(B_1),I_2$ | Monitor Call | SI | AF | |
| MD | $R_1,D_2(X_2,B_2)$ | Multiply (LH) | RX | 6C | |
| MDB | $R_1,D_2(X_2,B_2)$ | Multiply (LB) | RXE | ED1C | |
| MDBR | $R_1,R_2$ | Multiply (LB) | RRE | B31C | |
| MDE | $R_1,D_2(X_2,B_2)$ | Multiply (LH < SH) | RX | 7C | |
| MDEB | $R_1,D_2(X_2,B_2)$ | Multiply (LB < SB) | RXE | ED0C | |
| MDEBR | $R_1,R_2$ | Multiply (LB < SB) | RRE | B30C | |
| MDER | $R_1,R_2$ | Multiply (LH < SH) | RR | 3C | |
| MDR | $R_1,R_2$ | Multiply (LH) | RR | 2C | |
| ME | $R_1,D_2(X_2,B_2)$ | Multiply (LH < SH) | RX | 7C | |
| MEE | $R_1,D_2(X_2,B_2)$ | Multiply (SH) | RXE | ED37 | |
| MEEB | $R_1,D_2(X_2,B_2)$ | Multiply (SB) | RXE | ED17 | |
| MEEBR | $R_1,R_2$ | Multiply (SB) | RRE | B317 | |
| MEER | $R_1,R_2$ | Multiply (SH) | RRE | B337 | |
| MER | $R_1,R_2$ | Multiply (LH < SH) | RR | 3C | |
| MGHI | $R_1,I_2$ | Multiply Halfword Immediate (64) | RI | A7D | N |
| MH | $R_1,D_2(X_2,B_2)$ | Multiply Halfword (32) | RX | 4C | |
| MHI | $R_1,I_2$ | Multiply Halfword Immediate (32) | RI | A7C | |
| ML | $R_1,D_2(X_2,B_2)$ | Multiply Logical (64 < 32) | RXE | E396 | N3 |
| MLG | $R_1,D_2(X_2,B_2)$ | Multiply Logical (128 < 64) | RXE | 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 | 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 |
| 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 | B31F | |
| 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 | B30F | |
| MSG | $R_1,D_2(X_2,B_2)$ | Multiply Single (64) | RXE | E30C | N |
| MSGF | $R_1,D_2(X_2,B_2)$ | Multiply Single (64 < 32) | RXE | E31C | N |
| MSGFR | $R_1,R_2$ | Multiply Single (64 < 32) | RRE | B91C | N |
| MSGR | $R_1,R_2$ | Multiply Single (64) | RRE | B90C | N |
| MSR | $R_1,R_2$ | Multiply Single (32) | RRE | B252 | |
| MSTA | $R_1$ | Modify Stacked State | RRE | B247 | |
| MVC | $D_1(L,B_1),D_2(B_2)$ | Move (character) | SS | D2 | |
| MVCDK | $D_1(B_1),D_2(B_2)$ | Move with Destination key | SSE | E50F | q |
| MVCIN | $D_1(L,B_1),D_2(B_2)$ | Move Inverse | SS | E8 | |
| MVCK | $D_1(R_1,B_1),D_2(B_2),R_3$ | Move with Key | SS | D9 | qc |
| MVCL | $R_1,R_2$ | Move Long | RR | 0E | ic |
| MVCLE | $R_1,R_3,D_2(B_2)$ | Move Long Extended | RS | A8 | c |
| MVCLU | $R_1,R_3,D_2(B_2)$ | Move Long Unicode | RSE | EB8E | c E2 |
| MVCP | $D_1(R_1,B_1),D_2(B_2),R_3$ | Move to Primary | SS | DA | qc |
| MVCS | $D_1(R_1,B_1),D_2(B_2),R_3$ | Move to Secondary | SS | DB | qc |
| MVCSK | $D_1(B_1),D_2(B_2)$ | Move with Source Key | SSE | E50E | q |
| MVI | $D_1(B_1),I_2$ | Move (immediate) | SI | 92 | |
| MVN | $D_1(L,B_1),D_2(B_2)$ | Move Numerics | SS | D1 | |
| MVO | $D_1(L_1,B_1),D_2(L_2,B_2)$ | Move with Offset | SS | F1 | |
| MVPG | $R_1,R_2$ | Move Page | RRE | B254 | qc |
| MVST | $R_1,R_2$ | Move String | RRE | B255 | c |
| MVZ | $D_1(L,B_1),D_2(B_2)$ | Move Zones | SS | D3 | |
| MXBR | $R_1,R_2$ | Multiply (EB) | RRE | B34C | |
| MXD | $R_1,D_2(X_2,B_2)$ | Multiply (EH < LH) | RX | 67 | |
| MXDB | $R_1,D_2(X_2,B_2)$ | Multiply (EB < LB) | RXE | ED07 | |
| MXDBR | $R_1,R_2$ | Multiply (EB < LB) | RRE | B307 | |
| MXDR | $R_1,R_2$ | Multiply (EH < LH) | RR | 27 | |
| MXR | $R_1,R_2$ | Multiply (EH) | RR | 26 | |
| N | $R_1,D_2(X_2,B_2)$ | And (32) | RX | 54 | c |
| NC | $D_1(L,B_1),D_2(B_2)$ | And (character) | SS | D4 | c |
| NG | $R_1,D_2(X_2,B_2)$ | And (64) | RX | 54 | c N |
| NGR | $R_1,R_2$ | And (64) | RRE | B980 | c N |

| Mnemonic | Operands | Name | Format | Op Code | Notes |
|---|---|---|---|---|---|
| NI | $D_1(B_1),I_2$ | And (immediate) | SI | 94 | c |
| NIHH | $R_1,I_2$ | And Immediate (high high) | RI | A54 | c N |
| NIHL | $R_1,I_2$ | And Immediate (high low) | RI | A55 | c N |
| NILH | $R_1,I_2$ | And Immediate (low high) | RI | A56 | c N |
| NILL | $R_1,I_2$ | And Immediate (low low) | RI | A57 | c N |
| NR | $R_1,R_2$ | And (32) | RR | 14 | c |
| 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 | D6 | c |
| OG | $R_1,D_2(X_2,B_2)$ | Or (64) | RXE | E381 | c N |
| OGR | $R_1,R_2$ | Or (64) | RRE | B981 | c N |
| OI | $D_1(B_1),I_2$ | Or (immediate) | SI | 96 | c |
| OIHH | $R_1,I_2$ | Or Immediate (high high) | RI | A58 | c N |
| OIHL | $R_1,I_2$ | Or Immediate (high low) | RI | A59 | c N |
| OILH | $R_1,I_2$ | Or Immediate (low high) | RI | A5A | c N |
| OILL | $R_1,I_2$ | Or Immediate (low low) | RI | A5B | c N |
| OR | $R_1R_2$ | Or (32) | RR | 16 | c |
| PACK | $D_1(L_1,B_1),D_2(L_2,B_2)$ | Pack | SS | F2 | |
| PALB | | Purge ALB | RRE | B248 | p |
| PC | $D_2(B_2)$ | Program Call | S | B218 | q |
| PGIN | $R_1R_2$ | Page In | RRE | B22E | pc ES |
| PGOUT | $R_1R_2$ | Page In | RRE | B22F | pc ES |
| PKA | $D_1(B_1),D_2(L_2,B_2)$ | Pack ASCII | SS | E9 | E2 |
| PKU | $D_1(B_1),D_2(L_2,B_2)$ | Pack Unicode | SS | E1 | E2 |
| PLO | $R_1,D_2(B_2),R_3,D_4(B_4)$ | Perform Locked Operation | SS | EE | c |
| PR | | Program Return | E | 0101 | qn |
| PT | $R_1,R_2$ | Program Transfer | RRE | B228 | q |
| PTLB | | Purge TLB | S | B20D | p |
| RCHP | | Reset Channel Path | S | B23B | pc |
| RLL | $R_1,R_3,D_2(B_2)$ | Rotate Left Single Logical (32) | RSE | EB1D | N3 |
| RLLG | $R_1,R_3,D_2(B_2)$ | Rotate Left Single Logical (64) | RSE | EB1C | N |
| RP | $D_2(B_2)$ | Resume Program | S | B277 | qn |
| RRBE | $R_1,R_2$ | Reset Reference Bit Extended | RRE | B22A | pc |
| RSCH | | Resume Subchannel | S | B238 | pc |
| S | $R_1,D_2(X_2,B_2)$ | Subtract (32) | RX | 5B | c |
| SAC | $D_2(B_2)$ | Set Address Space Control | S | B219 | q |
| SACF | $D_2(B_2)$ | Set Address Space Control Fast | S | B279 | q |
| SAL | | Set Address Limit | S | B237 | p |
| SAM24 | | Set Addressing Mode (24) | E | 010C | N3 |
| SAM31 | | Set Addressing Mode (31) | E | 010D | N3 |
| SAM64 | | Set Addressing Mode (64) | E | 010E | N |
| SAR | $R_1,R_2$ | Set Access | RRE | B24E | |
| SCHM | | Set Channel Monitor | S | B23C | p |
| SCK | $D_2(B_2)$ | Set Clock | S | B204 | pc |
| SCKC | $D_2(B_2)$ | Set Clock Comparator | S | B206 | p |
| SCKPF | | Set Clock Programmable Field | E | 0107 | p |
| SD | $R_1,D_2(X_2,B_2)$ | Subtract Normalized (LH) | RX | 6B | c |
| SDB | $R_1,D_2(X_2,B_2)$ | Subtract (LB) | RXE | ED1B | c |
| SDBR | $R_1,R_2$ | Subtract (LB) | RRE | B31B | c |
| SDR | $R_1,R_2$ | Subtract Normalized (LH) | RR | 2B | c |
| SE | $R_1,D_2(X_2,B_2)$ | Subtract Normalized (SH) | RX | 7B | c |
| SEB | $R_1,D_2(X_2,B_2)$ | Subtract (SB) | RXE | ED0B | c |
| SEBR | $R_1,R_2$ | Subtract (SB) | RRE | B30B | c |
| SER | $R_1,R_2$ | Subtract Normalized (SH) | RR | 3B | c |
| SFPC | $R_1$ | Set FPC | RRE | B384 | |
| SG | $R_1,D_2(X_2,B_2)$ | Subtract (64) | RXE | E309 | c N |
| SGF | $R_1,D_2(X_2,B_2)$ | Subtract (64 < 32) | RXE | E319 | c N |
| SGFR | $R_1,R_2$ | Subtract (64 < 32) | RRE | B919 | c N |
| SGR | $R_1,R_2$ | Subtract (64) | RRE | B909 | c N |
| SH | $R_1,D_2(X_2,B_2)$ | Subtract Halfword | RX | 4B | c |
| SIE | $D_2(B_2)$ | Start Interpretive Execution | S | B214 | ip |
| SIGP | $R_1,R_3,D_2(B_2)$ | Signal Processor | RS | AE | pc |
| SL | $R_1,D_2(X_2,B_2)$ | Subtract Logical (32) | RX | 5F | c |
| SLA | $R_1,D_2(B_2)$ | Shift Left Single (32) | RS | 8B | c |
| SLAG | $R_1,D_2(B_2)$ | Shift Left Single (64) | RSE | EB0B | c N |
| SLB | $R_1,D_2(X_2,B_2)$ | Subtract Logical with Borrow (32) | RXE | E399 | c N3 |
| SLBG | $R_1,D_2(X_2,B_2)$ | Subtract Logical with Borrow (64) | RXE | E389 | c N |
| SLBGR | $R_1,R_2$ | Subtract Logical with Borrow (64) | RRE | B989 | c N |
| SLBR | $R_1,R_2$ | Subtract Logical with Borrow (32) | RRE | B999 | c N3 |
| SLDA | $R_1,D_2(B_2)$ | Shift Left Double | RS | 8F | c |
| SLG | $R_1,D_2(X_2,B_2)$ | Subtract Logical (64) | RXE | E30B | c N |
| SLGF | $R_1,D_2(X_2,B_2)$ | Subtract Logical (64 < 32) | RXE | E31B | c N |
| SLGFR | $R_1,R_2$ | Subtract Logical (64 < 32) | RRE | B91B | c N |
| SLGR | $R_1,R_2$ | Subtract Logical (64) | RRE | B90B | c N |
| SLDL | $R_1,D_2(B_2)$ | Shift Left Double Logical | RS | 8D | |
| SLL | $R_1,D_2(B_2)$ | Shift Left Single Logical (32) | RS | 89 | |
| SLLG | $R_1,D_2(B_2)$ | Shift Left Single Logical (64) | RSE | EB0D | N |
| SLR | $R_1,R_2$ | Subtract Logical (32) | RR | 1F | c |
| SP | $D_1(L_1,B_1),D_2(L_2,B_2)$ | Subtract Decimal | SS | FB | c |
| SPKA | $D_2(B_2)$ | Set PSW Key from Address | S | B20A | q |

| Mnemonic | Operands | Name | Format | Op Code | Notes |
|---|---|---|---|---|---|
| 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 | |
| SQXR | $R_1,R_2$ | Square Root (EH) | RRE | B336 | |
| SQXBR | $R_1,R_2$ | Square Root (EB) | RRE | B316 | |
| SR | $R_1,R_2$ | Subtract (32) | RR | 1B | c |
| SRA | $R_1,D_2(B_2)$ | Shift Right Single (32) | RS | 8A | c |
| SRAG | $R_1,D_2(B_2)$ | Shift Right Single (64) | RSE | EB0A | c N |
| SRDA | $R_1,D_2(B_2)$ | Shift Right Double | RS | 8E | c |
| SRDL | $R_1,D_2(B_2)$ | Shift Right Double Logical | RS | 8C | |
| SRL | $R_1,D_2(B_2)$ | Shift Right Single Logical (32) | RS | 88 | |
| SRLG | $R_1,D_2(B_2)$ | Shift Right Single Logical (64) | RSE | EB0C | N |
| SRNM | $D_2(B_2)$ | Set Rounding Mode | S | B299 | |
| SRP | $D_1(L_1,B_1),D_2(B_2),I_3$ | Shift and Round Decimal | SS | F0 | c |
| SRST | $R_1,R_2$ | Search String | RRE | B25E | c |
| SSAR | $R_1$ | Set Secondary ASN | RRE | B225 | |
| SSCH | $D_2(B_2)$ | Start Subchannel | S | B233 | pc |
| SSKE | $R_1,R_2$ | Set Storage Key Extended | RRE | B22B | p |
| SSM | $D_2(B_2)$ | Set System Mask | S | 80 | p |
| ST | $R_1,D_2(X_2,B_2)$ | Store (32) | RX | 50 | |
| STAM | $R_1,R_3,D_2(B_2)$ | Store Access Multiple | RS | 9B | |
| STAP | $D_2(B_2)$ | Store CPU Address | S | B212 | p |
| STC | $R_1,D_2(X_2,B_2)$ | Store Character | RX | 42 | |
| STCK | $D_2(B_2)$ | Store Clock | S | B205 | c |
| STCKC | $D_2(B_2)$ | Store Clock Comparator | S | B207 | p |
| STCKE | $D_2(B_2)$ | Store Clock Extended | S | B278 | c |
| STCM | $R_1,M_3,D_2(B_2)$ | Store Characters under Mask (low) | RS | BE | |
| STCMH | $R_1,M_3,D_2(B_2)$ | Store Characters under Mask (high) | RSE | EB2C | N |
| 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) | RSE | EB25 | p N |
| STCTL | $R_1,R_3,D_2(B_2)$ | Store Control (32) | RS | B6 | p |
| STD | $R_1,D_2(X_2,B_2)$ | Store (L) | RX | 60 | |
| STE | $R_1,D_2(X_2,B_2)$ | Store (S) | RX | 70 | |
| STFL | $D_2(B_2)$ | Store Facility List | S | B2B1 | p N3 |
| STFPC | $D_2(B_2)$ | Store FPC | S | B29C | |
| STG | $R_1,D_2(X_2,B_2)$ | Store (64) | RXE | E324 | N |
| STH | $R_1,D_2(X_2,B_2)$ | Store Halfword | RX | 40 | |
| STIDP | $D_2(B_2)$ | Store CPU ID | S | B202 | p |
| STM | $R_1,R_3,D_2(B_2)$ | Store Multiple (32) | RS | 90 | |
| STMG | $R_1,R_3,D_2(B_2)$ | Store Multiple (64) | RSE | EB24 | N |
| STMH | $R_1,R_3,D_2(B_2)$ | Store Multiple High | RSE | EB26 | N |
| 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 | RXE | E38E | N |
| STPT | $D_2(B_2)$ | Store CPU Timer | S | B209 | p |
| STPX | $D_2(B_2)$ | Store Prefix | S | B211 | p |
| STRAG | $D_1(B_1),D_2(B_2)$ | Store Real Address | SSE | E502 | p N |
| STRV | $R_1,D_2(X_2,B_2)$ | Store Reversed (32) | RXE | E33E | N3 |
| STRVG | $R_1,D_2(X_2,B_2)$ | Store Reversed (64) | RXE | E32F | N |
| STRVH | $R_1,D_2(X_2,B_2)$ | Store Reversed (16) | RXE | E33F | N3 |
| STSI | $D_2(B_2)$ | Store System Information | S | B27D | pc |
| STSCH | $D_2(B_2)$ | Store Subchannel | S | B234 | pc |
| STURA | $R_1,R_2$ | Store Using Real Address (32) | RRE | B246 | p |
| STURG | $R_1,R_2$ | Store Using Real Address (64) | RRE | B925 | p N |
| SU | $R_1,D_2(X_2,B_2)$ | Subtract Unnormalized (SH) | RX | 7F | c |
| SUR | $R_1,R_2$ | Subtract Unnormalized (SH) | RR | 3F | c |
| SVC | I | Supervisor Call | RR | 0A | |
| SW | $R_1,D_2(X_2,B_2)$ | Subtract Unnormalized (LH) | RX | 6F | c |
| SWR | $R_1,R_2$ | Subtract Unnormalized (LH) | RR | 2F | c |
| SXBR | $R_1,D_2$ | Subtract (EB) | RRE | B34B | c |
| SXR | $R_1,D_2$ | Subtract Normalized (EH) | RR | 37 | c |
| 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 | B351 | c |
| TBEDR | $R_1,M_3,R_2$ | Convert HFP to BFP (SB < LH) | RRF | B350 | c |
| TCDB | $R_1,D_2(X_2,B_2)$ | Test Data Class (LB) | RXE | ED11 | c |
| TCEB | $R_1,D_2(X_2,B_2)$ | Test Data Class (SB) | RXE | ED10 | c |
| TCXB | $R_1,D_2(X_2,B_2)$ | Test Data Class (EB) | RXE | ED12 | c |
| THDER | $R_1,R_2$ | Convert BFP to HFP (LH < SB) | RRE | B358 | c |

| Mnemonic | Operands | Name | Format | Op Code | Notes |
|---|---|---|---|---|---|
| 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 | A70 | c |
| TMHH | $R_1,I_2$ | Test under Mask (high high) | RI | A72 | c N |
| TMHL | $R_1,I_2$ | Test under Mask (high low) | RI | A73 | c N |
| TML | $R_1,I_2$ | Test under Mask Low | RI | A71 | c |
| TMLH | $R_1,I_2$ | Test under Mask (low high) | RI | A70 | c N |
| TMLL | $R_1,I_2$ | Test under Mask (low low) | RI | A71 | c N |
| 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 | DC | |
| TRACE | $R_1,R_3,D_2(B_2)$ | Trace (32) | RS | 99 | p |
| TRACG | $R_1,R_3,D_2(B_2)$ | Trace (64) | RSE | EB0F | p N |
| TRAP2 | | Trap | E | 01FF | |
| TRAP4 | $D_2(B_2)$ | Trap | S | B2FF | |
| TRE | $R_1,R_2$ | Translate Extended | RRE | B2A5 | c |
| TROO | $R_1,R_2$ | Translate One to One | RRE | B993 | c E2 |
| TROT | $R_1,R_2$ | Translate One to Two | RRE | B992 | c E2 |
| TRT | $D_1(L,B_1),D_2(B_2)$ | Translate and Test | SS | DD | c |
| TRTO | $R_1,R_2$ | Translate Two to One | RRE | B991 | c E2 |
| TRTT | $R_1,R_2$ | Translate Two to Two | RRE | B990 | c E2 |
| TS | $D_2(B_2)$ | Test and Set | S | 93 | c |
| TSCH | $D_2(B_2)$ | Test Subchannel | S | B235 | pc |
| UNPK | $D_1(L_1,B_1),D_2(L_2,B_2)$ | Unpack | SS | F3 | |
| UNPKA | $D_1(L_1,B_1),D_2(B_2)$ | Unpack ASCII | SS | EA | c E2 |
| UNPKU | $D_1(L_1,B_1),D_2(B_2)$ | Unpack Unicode | SS | 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 | D7 | c |
| XG | $R_1,D_2(X_2,B_2)$ | Exclusive Or (64) | RXE | 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 |
| XR | $R_1,R_2$ | Exclusive Or (32) | RR | 17 | c |
| XSCH | | Cancel Subchannel | S | B276 | pc |
| ZAP | $D_1(L_1,B_1),D_2(L_2,B_2)$ | Zero and Add | SS | F8 | c |
| --- | Model-dependent | Diagnose | -- | 83 | pu |

**Floating-Point Operand Lengths and Types:**

| | |
|---|---|
| E | Extended (binary or hex) |
| EB | Extended binary |
| EH | Extended hex |
| L | Long (binary or hex) |
| LB | Long binary |
| LH | Long hex |
| S | Short (binary or hex) |
| SB | Short binary |
| SH | Short hex |

**Notes:**

| | |
|---|---|
| c | Condition code set |
| i | Interruptible instruction |
| n | New condition code loaded |
| p | Privileged instruction |
| q | Semiprivileged instruction |
| E2 | Extended-translation facility 2 |
| ES | Expanded-storage facility |
| N | Instruction new in z/Architecture compared to ESA/390 |
| N3 | Instruction new in z/Architecture compared to ESA/390 and added to ESA/390 |

# Machine Instructions by Operation Code

| Op Code | Mnemonic |
|---|---|
| 0101 | PR |
| 0102 | UPT |
| 0107 | SCKPF |
| 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 |

| Op Code | Mnemonic |
|---|---|
| 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 |

| Op Code | Mnemonic |
|---|---|
| 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 |
| 41 | LA |
| 42 | STC |
| 43 | IC |
| 44 | EX |
| 45 | BAL |

| Op Code | Mne-monic |
|---|---|
| 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 |
| 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 |

| Op Code | Mne-monic |
|---|---|
| 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 |
| 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 |
| B27D | STSI |
| B299 | SRNM |

| Op Code | Mne-monic |
|---|---|
| B29C | STFPC |
| B29D | LFPC |
| B2A5 | TRE |
| B2A6 | CUUTF |
| B2A7 | CUTFU |
| B2B1 | STFL |
| B2B2 | LPSWE |
| B2FF | TRAP4 |
| B300 | LPEBR |
| B301 | LNEBR |
| B302 | LTEBR |
| B303 | LCEBR |
| B304 | LDEBR |
| B305 | LXDBR |
| B306 | LXEBR |
| B307 | MXDBR |
| B308 | KEBR |
| B309 | CEBR |
| B30A | AEBR |
| B30B | SEBR |
| B30C | MDEBR |
| B30D | DEBR |
| B30E | MAEBR |
| B30F | MSEBR |
| B310 | LPDBR |
| B311 | LNDBR |
| B312 | LTDBR |
| B313 | LCDBR |
| B314 | SQEBR |
| B315 | SQDBR |
| B316 | SQXBR |
| B317 | MEEBR |
| B318 | KDBR |
| B319 | CDBR |
| B31A | ADBR |
| B31B | SDBR |
| B31C | MDBR |
| B31D | DDBR |
| B31E | MADBR |
| B31F | MSDBR |
| B324 | LDER |
| B325 | LXDR |
| B326 | LXER |
| B336 | SQXR |
| B337 | MEER |
| B340 | LPXBR |
| B341 | LNXBR |
| B342 | LTXBR |
| B343 | LCXBR |
| B344 | LEDBR |
| B345 | LDXBR |
| B346 | LEXBR |
| B347 | FIXBR |
| B348 | KXBR |
| B349 | CXBR |
| B34A | AXBR |
| B34B | SXBR |
| B34C | MXBR |
| B34D | DXBR |
| B350 | TBEDR |
| B351 | TBDR |
| B353 | DIEBR |
| B357 | FIEBR |
| B358 | THDER |
| B359 | THDR |
| B35B | DIDBR |
| B35F | FIDBR |
| B360 | LPXR |
| B361 | LNXR |
| B362 | LTXR |
| B363 | LCXR |
| B365 | LXR |
| B366 | LEXR |
| B367 | FIXR |
| B369 | CXR |
| B374 | LZER |
| B375 | LZDR |
| B376 | LZXR |
| B377 | FIER |
| B37F | FIDR |
| B384 | SFPC |
| B38C | EFPC |
| B394 | CEFBR |
| B395 | CDFBR |
| B396 | CXFBR |
| B398 | CFEBR |
| B399 | CFDBR |
| B39A | CFXBR |
| B3A4 | CEGBR |
| B3A5 | CDGBR |
| B3A6 | CXGBR |
| B3A8 | CGEBR |
| B3A9 | CGDBR |
| B3AA | CGXBR |

| Op Code | Mnemonic |
|---|---|
| B3B4 | CEFR |
| B3B5 | CDFR |
| B3B6 | CXFR |
| B3B8 | CFER |
| B3B9 | CFDR |
| B3BA | CFXR |
| B3C4 | CEGR |
| B3C5 | CDGR |
| B3C6 | CXGR |
| B3C8 | CGER |
| B3C9 | CGDR |
| B3CA | CGXR |
| B6 | STCTL |
| B7 | LCTL |
| B900 | LPGR |
| B901 | LNGR |
| B902 | LTGR |
| B903 | LCGR |
| B904 | LGR |
| B905 | LURAG |
| 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 | LLGFR |
| B916 | LCGR |
| B917 | LLGTR |
| B918 | AGFR |
| B919 | SGFR |
| B91A | ALGFR |
| B91B | SLGFR |
| B91C | MSGFR |
| B91D | DSGFR |
| B91F | LRVR |
| B920 | CGR |
| B921 | CLGR |
| B925 | STURG |
| B930 | CGFR |
| B931 | CLGFR |
| B946 | BCTGR |
| B980 | NGR |
| B981 | OGR |
| B982 | XGR |
| B986 | MLGR |
| B987 | DLGR |
| B988 | ALCGR |
| B989 | SLBGR |
| B98D | EPSW |
| B990 | TRTT |
| B991 | TRTO |
| B992 | TROT |
| B993 | TROO |
| B996 | MLR |
| B997 | DLR |
| B998 | ALCR |
| B999 | SLBR |
| B99D | ESEA |
| BA | CS |
| BB | CDS |
| BD | CLM |
| BE | STCM |
| BF | ICM |
| C00 | LARL |
| C04 | BRCL |
| C05 | BRASL |
| D1 | MVN |
| D2 | MVC |
| D3 | MVZ |
| D4 | NC |
| D5 | CLC |
| D6 | OC |
| D7 | XC |
| D9 | MVCK |
| DA | MVCP |
| DB | MVCS |
| DC | TR |
| DD | TRT |
| DE | ED |
| DF | EDMK |
| E1 | PKU |
| E2 | UNPKU |
| E303 | LRAG |
| E304 | LG |
| E308 | AG |
| E309 | SG |
| E30A | ALG |

| Op Code | Mnemonic |
|---|---|
| E30B | SLG |
| E30C | MSG |
| E30D | DSG |
| E30E | CVBG |
| E30F | LRVG |
| 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 |
| E32E | CVDG |
| E32F | STRVG |
| E330 | CGF |
| E331 | CLGF |
| E33E | STRV |
| E33F | STRVH |
| E346 | BCTG |
| E380 | NG |
| E381 | OG |
| E382 | XG |
| E386 | MLG |
| E387 | DLG |
| E388 | ALCG |
| E389 | SLBG |
| E38E | STPQ |
| E38F | LPQ |
| E390 | LLGC |
| E391 | LLGH |
| E396 | ML |
| E397 | DL |
| E398 | ALC |
| E399 | SLB |
| E500 | LASP |
| E501 | TPROT |
| E502 | STRAG |
| E50E | MVCSK |
| E50F | MVCDK |
| E8 | MVCIN |
| E9 | PKA |
| EA | UNPKA |
| EB04 | LMG |
| EB0A | SRAG |
| EB0B | SLAG |
| EB0C | SRLG |
| EB0D | SLLG |
| EB0F | TRACG |
| EB1C | RLLG |
| EB1D | RLL |
| EB20 | CLMH |
| EB24 | STMG |
| EB25 | STCTG |
| EB26 | STMH |
| EB2C | STCMH |
| EB2F | LCTLG |
| EB30 | CSG |
| EB3E | CDSG |
| EB44 | BXHG |
| EB45 | BXLEG |
| EB80 | ICMH |
| EB8E | MVCLU |
| EB8F | CLCLU |
| EB96 | LMH |
| EBC0 | TP |
| EC44 | BRXHG |
| EC45 | BRXLG |
| 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 |

| Op Code | Mnemonic |
|---|---|
| ED19 | CDB |
| ED1A | ADB |
| ED1B | SDB |
| ED1C | MDB |
| ED1D | DDB |
| ED1E | MADB |
| ED1F | MSDB |
| ED24 | LDE |
| ED25 | LXD |
| ED26 | LXE |
| ED34 | SQE |
| ED35 | SQD |
| ED37 | MEE |
| EE | PLO |
| 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 Logical | Zero, no carry | Not zero, no carry | Zero, carry | Not zero, carry |
| Add Logical with Carry | Zero, no carry | Not zero, no carry | Zero, carry | Not zero, carry |
| And | Zero | Not zero | ---- | ---- |
| And Immediate | Sixteen bits zero | Sixteen bits not zero | ---- | ---- |
| Checksum | Checksum complete | ---- | ---- | CPU-determined completion |
| Compare | Equal | First op low | First op high | ---- |
| Compare and Form Codeword | Equal | First op low and ctl = 0, or first op high and ctl = 1 | First op high and ctl = 0, or first op low and ctl = 1 | ---- |
| Compare and Swap | Equal | Not equal | ---- | ---- |
| Compare Double and Swap | Equal | Not equal | ---- | ---- |
| Compare Halfword | Equal | First op low | First op high | ---- |
| Compare Halfword Immediate | Equal | First op low | First op high | ---- |
| Compare Logical | Equal | First op low | First op high | ---- |
| Compare Logical Characters under Mask | Equal, or Mask is zero | First op low | First op high | ---- |
| Compare Logical Long | Equal | First op low | First op high | ---- |
| Compare Logical Long Extended | Equal | First op low | First op high | CPU-determined completion |
| Compare Logical Long Unicode | Equal | First op low | First op high | CPU-determined completion |
| Compare Logical String | Equal | First op low | First op high | CPU-determined completion |
| Compare until 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 |
| Convert Unicode to UTF-8 | Data processed | First op full | ---- | CPU-determined completion |
| Convert UTF-8 to Unicode | Data processed | First op full | ---- | CPU-determined completion |
| Exclusive Or | Zero | Not zero | ---- | ---- |
| Insert Characters under Mask | All zero, or mask is zero | Leftmost bit = 1 | Not zero, but with leftmost bit = 0 | ---- |
| Load and Test | Zero | < Zero | > Zero | ---- |
| Load Complement | Zero | < Zero | > Zero | Overflow |
| Load Negative | Zero | < Zero | ---- | ---- |
| Load Positive | Zero | ---- | > Zero | Overflow |
| Move Long | Operand lengths equal | First op shorter | First op longer | Overlap |
| Move Long Extended | Operand lengths equal | First op shorter | First op longer | CPU-determined completion |
| Move Long Unicode | Operand lengths equal | First op shorter | First op longer | CPU-determined completion |
| Move Page | Data moved | First op invalid, both valid in ES, locked, or ES error | Second op invalid | ---- |
| Move String | ---- | Second op moved | ---- | CPU-determined completion |
| Or | Zero | Not zero | ---- | ---- |
| Or Immediate | Sixteen bits zero | Sixteen bits not zero | ---- | ---- |

| Condition Code → | 0 | 1 | 2 | 3 |
|---|---|---|---|---|
| Mask Bit Value → | 8 | 4 | 2 | 1 |
| Perform Locked Operation (test bit zero) | Equal | First op not equal | First op equal, third op not equal | ---- |
| Perform Locked Operation (test bit one) | Code valid | ---- | ---- | Code invalid |
| Search String | ---- | Character found | Character not found | CPU-determined completion |
| Set Program Mask | See Note | See Note | See Note | See Note |
| Shift Left Double | Zero | < Zero | > Zero | Overflow |
| Shift Left Single | Zero | < Zero | > Zero | Overflow |
| Shift Right Double | Zero | < Zero | > Zero | ---- |
| Shift Right Single | Zero | < Zero | > Zero | ---- |
| Store Clock | Set state | Not-set state | Error state | Stopped state or not operational |
| Store Clock | Set state | Not-set state | Error state | Stopped state or not operational |
| Subtract | Zero | < Zero | > Zero | Overflow |
| Subtract Halfword | Zero | < Zero | > Zero | Overflow |
| Subtract Logical | ---- | Not zero, borrow | Zero, no borrow | Not zero, no borrow |
| Subtract Logical with Borrow | Zero, borrow | Not zero, borrow | Zero, no borrow | Not zero, no borrow |
| Test Addressing Mode | 24-bit mode | 31-bit mode | ---- | ---- |
| Test and Set | Leftmost bit zero | Leftmost bit one | ---- | ---- |
| Test under Mask (TM) | All zeros, or mask is zero | Mixed 0's and 1's | --- | All ones |
| Test under Mask (TMHH, TMHL, TMLH, TMLL) | All zeros or mask is zero | Mixed 0's and 1's and leftmost bit zero | Mixed 0's and 1's and leftmost bit one | All ones |
| Test under Mask High, Low | All zeros or mask is zero | Mixed 0's and 1's and leftmost bit zero | Mixed 0's and 1's and leftmost bit one | All ones |
| Translate and Test | All zeros | Not zero, scan incomplete | Not zero, scan complete | ---- |
| Translate Extended | Data processed | First op byte equal test byte | ---- | CPU-determined completion |
| Translate One to One, One to Two, Two to One, Two to Two | Character not found | Character found | ---- | CPU determinded completion |
| Unpack ASCII | Sign plus | Sign minus | ---- | Sign invalid |
| Unpack Unicode | Sign plus | Sign minus | ---- | Sign invalid |
| Update Tree | Compare equal at current node on path | Path complete, no nodes compared equal | ---- | Path not complete and compared register negative |
| **Decimal Instructions** | | | | |
| Add Decimal | Zero | < Zero | > Zero | Overflow |
| Compare Decimal | Equal | First op low | First op high | ---- |
| Edit | Zero | < Zero | > Zero | ---- |
| Edit and Mark | Zero | < Zero | > Zero | ---- |
| Shift and Round Decimal | Zero | < Zero | > Zero | Overflow |
| Subtract Decimal | Zero | < Zero | > Zero | Overflow |
| Test Decimal | Digits and sign valid | Sign invalid | Digit invalid | Sign and digit invalid |
| Zero and Add | Zero | < Zero | > Zero | Overflow |
| **Floating-Point Instructions** | | | | |
| Add | Zero | < Zero | > Zero | NaN |
| Add Normalized | Zero | < Zero | > Zero | ---- |
| Add Unnormalized | Zero | < Zero | > Zero | ---- |
| Compare (BFP) | Equal | First op low | First op high | Unordered |
| Compare (HFP) | Equal | First op low | First op high | ---- |
| Compare and Signal | Equal | First op low | First op high | Unordered |
| Convert BFP to HFP | Zero | < Zero | > Zero | Special case |
| Convert HFP to BFP | Zero | < Zero | > Zero | Special case |

| Condition Code → | 0 | 1 | 2 | 3 |
|---|---|---|---|---|
| Mask Bit Value → | 8 | 4 | 2 | 1 |
| 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 | ---- |
| Subtract | Zero | < Zero | > Zero | NaN |
| Subtract Normalized | Zero | < Zero | > Zero | ---- |
| Subtract Unnormalized | Zero | < Zero | > Zero | ---- |
| Test Data Class | Zero (no match) | One (match) | ---- | ---- |
| **Control Instructions** | | | | |
| Compare and Swap and Purge | Equal | Not equal | ---- | ---- |
| Diagnose | See note | See note | See note | See note |
| Extract Stacked State | Branch state entry | Program call state entry | ---- | ---- |
| Insert Address Space Control | Primary-space mode | Secondary-space mode | Access-register mode | Home-space mode |
| Load Address Space Parameters | Parameters loaded | Primary not available | Secondary not authorized or not available | Space-switch event |
| Load PSW | See note | See note | See note | See note |
| Load PSW Extended | See note | See note | See note | See note |
| Load Real Address | Translation available | Segment-table entry invalid | Page-table entry invalid | See note |
| Move to Primary | Length ≤ 256 | ---- | ---- | Length > 256 |
| Move to Secondary | Length ≤ 256 | ---- | ---- | Length > 256 |
| Move with Key | Length ≤ 256 | ---- | ---- | Length > 256 |
| Page In | Operation completed | ES data error | ---- | ES block not available |
| Page Out | Operation completed | ES data error | ---- | ES block not available |
| Program Return | See note | See note | See note | See note |
| Reset Reference Bit Extended | Ref = 0, Chg = 0 | Ref = 0, Chg = 1 | Ref = 1, Chg = 0 | Ref = 1, Chg = 1 |
| Resume Program | See note | See note | See note | See note |
| Set Clock | Set | Secure | ---- | Not operational |
| Signal Processor | Accepted | Status stored | Busy | Not operational |
| Store System Information | Info provided | ---- | ---- | Info not available |
| Test Access | ALET = 0 | ALET uses DUALD | ALET uses PSALD | ALET = 1 or causes ART exception |
| Test Block | Usable | Unusable | ---- | ---- |
| Test Protection | Fetch and store allowed | Fetch allowed; no store allowed | No fetch or store allowed | Translation not available |
| **Input/Output Instructions** | | | | |
| Cancel Subchannel | Function started | ---- | ---- | Not operational |
| Clear Subchannel | Function started | ---- | ---- | Not operational |

| Condition Code → | 0 | 1 | 2 | 3 |
|---|---|---|---|---|
| Mask Bit Value → | 8 | 4 | 2 | 1 |
| Halt Subchannel | Function started | Noninter-mediate status pending | Busy | Not opera-tional |
| Modify Subchannel | Function executed | Status pending | Busy | Not opera-tional |
| Reset Channel Path | Function started | ---- | Busy | Not opera-tional |
| Resume Sub-channel | Function started | Status pending | Not applicable | Not opera-tional |
| Start Subchannel | Function started | Status pending | Busy | Not opera-tional |
| Store Channel Report Word | CRW stored | Zeros stored | ---- | ---- |
| Store Subchannel | SCHIB stored | ---- | ---- | Not opera-tional |
| Test Pending Inter-ruption | Inter-ruption not pend-ing | Inter-ruption code stored | ---- | ---- |
| Test Subchannel | Status was pending | Status was not pending | ---- | Not opera-tional |

**Notes:**

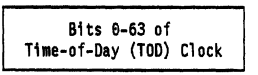
For Diagnose, the resulting condition code is model-dependent.

For Load Real Address, condition code 3 is set if address-space-control element not available, region-table entry outside table or invalid, segment-table entry outside table, or, for LRA in 24- or 31-bit mode when bits 0-32 of entry address not all zeros, segment- or page-table entry invalid.

For Load PSW, Load PSW Extended, and Resume Program, the condition code is loaded from the condition-code field of the second operand.

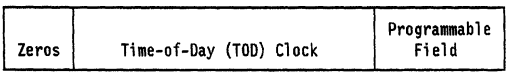
For Set Program Mask, the condition code is loaded from bit positions 2 and 3 of the first operand.

## Operand of Store Clock




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

## Operand of Store Clock Extended




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

## Assembler Instructions

| Function | Mnemonic | Meaning |
|---|---|---|
| Option control | *PROCESS | 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 |

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

**Source:** SC26-4940.

# Extended-Mnemonic Instructions for Branch on Condition

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

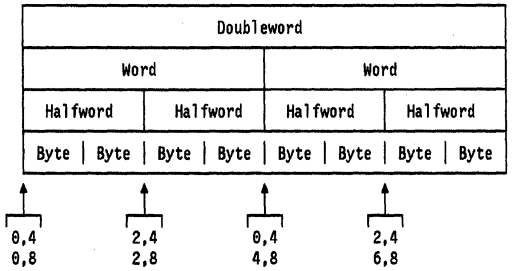
Source: SC26-4940.
*Second operand, not shown, is $D_2$ ($X_2$, $B_2$) for RX format and $R_2$ for RR format.

# Extended-Mnemonic Instructions for 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 (A:B) | BRH or JH* | Branch Relative on A High | BRC 2,$I_2$ |
| | BRL or JL* | Branch Relative on A Low | BRC 4,$I_2$ |
| | BRE or JE* | Branch Relative on A Equal B | BRC 8,$I_2$ |
| | BRNH or JNH* | Branch Relative on A Not High | BRC 13,$I_2$ |
| | BRNL or JNL* | Branch Relative on A Not Low | BRC 11,$I_2$ |
| | BRNE or JNE* | Branch Relative on A Not Equal B | BRC 7,$I_2$ |
| After Arithmetic Instructions | BRP or JP* | Branch Relative on Plus | BRC 2,$I_2$ |
| | BRM or JM* | Branch Relative on Minus | BRC 4,$I_2$ |
| | BRZ or JZ* | Branch Relative on Zero | BRC 8,$I_2$ |
| | BRO or JO* | Branch Relative on Overflow | BRC 1,$I_2$ |
| | BRNP or JNP* | Branch Relative on Not Plus | BRC 13,$I_2$ |
| | BRNM or JNM* | Branch Relative on Not Minus | BRC 11,$I_2$ |
| | BRNZ or JNZ* | Branch Relative on Not Zero | BRC 7,$I_2$ |
| | BRNO or JNO* | Branch Relative on No Overflow | BRC 14,$I_2$ |
| After Test under Mask instruction | BRO or JO* | Branch Relative if Ones | BRC 1,$I_2$ |
| | BRM or JM* | Branch Relative if Mixed | BRC 4,$I_2$ |
| | BRZ or JZ* | Branch Relative if Zeros | BRC 8,$I_2$ |
| | BRNO or JNO* | Branch Relative if Not Ones | BRC 14,$I_2$ |
| | BRNM or JNM* | Branch Relative if Not Mixed | BRC 11,$I_2$ |
| | BRNZ or JNZ* | Branch Relative if Not Zeros | BRC 7,$I_2$ |
| Non-Branch Relative on Condition | 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 | BRCT $R_1$,$I_2$ |
| | JCTG | Branch Relative on Count | BRCTG $R_1$,$I_2$ |
| | JXH | Branch Relative on Index High | BRXH $R_1$,$R_3$,$I_2$ |
| | JXHG | Branch Relative on Index High | BRXHG $R_1$,$R_3$,$I_2$ |
| | JXLE | Br. Rel. on Index Low or Equal | BRXLE $R_1$,$R_3$,$I_2$ |
| | JXLEG | Br. Rel. on Index Low or Equal | 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.

## CNOP Alignment




Source: SC26-4840.

## Summary of Constants

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

Source: SC26-4840.

## 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 22) |
| 136-139 | R | 88 | SVC-interruption identification (0-12 zeros, 13-14 ILC, 15 zero, 16-31 code) |
| 140-143 | R | 8C | Program-interruption identification (0-12 zeros, 13-14 ILC, 15 zero, 16-31 code) |
| 144-147 | R | 90 | Data-exception code (0-23 zeros, 24-31 code (see table on page 23)) |
| 148-149 | R | 94 | Monitor-class number (0-7 zeros, 8-15 number) |

| Area (Dec) | Addr Type | Hex Addr | Function |
|---|---|---|---|
| 150-151 | R | 96 | PER code (0 successful branching, 1 instruction fetching, 2 storage alteration, 2 and 4 stura, 3 and 5-8 zeros, 8-13 ATMID, 14-15 AI) |
| 152-159 | R | 98 | PER address |
| 160 | R | A0 | Exception access identification (0-3 zeros, 4-7 access-register number) |
| 161 | R | A1 | PER access identification (0-3 zeros, 4-7 access-register number) |
| 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 23) |
| 176-183 | R | B0 | Monitor code |
| 184-187 | R | B8 | Subsystem-identification word (0-14 zeros, 15 one, 16-31 subchannel number) |
| 188-191 | R | BC | I/O-interruption parameter |
| 192-195 | R | C0 | I/O-interruption-identification word (0-1 zeros, 2-4 I/O-interruption subclass, 5-31 zeros) |
| 200-203 | R | C8 | STFL facility list (0 certain z/Architecture instructions available, 1 z/Architecture installed, 2 z/Architecture active, 16 extended-translation facility 2 installed) |
| 232-239 | R | E8 | Machine-check-interruption code (see diagram on page 41) |
| 244-247 | R | F4 | External-damage code (see diagram on page 41) |
| 248-255 | R | F8 | Failing-storage address |
| 288-303 | R | 120 | Restart old PSW |
| 304-319 | R | 130 | External old PSW |
| 320-335 | R | 140 | Supervisor-call old PSW |
| 336-351 | R | 150 | Program old PSW |
| 352-367 | R | 160 | Machine-check old PSW |
| 368-383 | R | 170 | Input/output old PSW |
| 416-431 | R | 1A0 | Restart new PSW |
| 432-447 | R | 1B0 | External new PSW |
| 448-463 | R | 1C0 | Supervisor-call new PSW |
| 464-479 | R | 1D0 | Program new PSW |
| 480-495 | R | 1E0 | Machine-check new PSW |
| 496-511 | R | 1F0 | Input/output new PSW |
| 4544-4607 | R | 11C0 | Available for programming |
| 4608-4735 | A/R | 1200 | Store-status/machine-check floating-point-register save area |
| 4736-4863 | A/R | 1280 | Store-status/machine-check general-register save area |
| 4864-4879 | A/R | 1300 | Store-status PSW save area or machine-check fixed-logout area* |
| 4888-4891 | A | 1318 | Store-status prefix save area |
| 4892-4895 | A/R | 131C | Store-status/machine-check floating-point-control-register save area |
| 4900-4903 | A/R | 1324 | Store-status/machine-check TOD-programmable-register save area |
| 4904-4911 | A/R | 1328 | Store-status/machine-check CPU-timer save area |
| 4913-4919 | A/R | 1331 | Store-status/machine-check clock-comparator bits 0-55 save area (zeros at 4912) |
| 4928-4991 | A/R | 1340 | Store-status/machine-check access-register save area |
| 4992-5119 | A/R | 1380 | Store-status/machine-check control-register save area |

A=Absolute address. R=Real address.
A/R=A if store status, R if machine check.
* Contents may vary among models; see System Library manuals.

## External-Interruption Codes

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

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

## Program-Interruption Codes

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

| Code (Hex) | Condition |
|---|---|
| 0001 | Operation exception |
| 0002 | Privileged-operation exception |
| 0003 | Execute exception |
| 0004 | Protection exception |
| 0005 | Addressing exception |
| 0006 | Specification exception |
| 0007 | Data exception |
| 0008 | Fixed-point-overflow exception |
| 0009 | Fixed-point-divide exception |
| 000A | Decimal-overflow exception |
| 000B | Decimal-divide exception |
| 000C | HFP-exponent-overflow exception |
| 000D | HFP-exponent-underflow exception |
| 000E | HFP-significance exception |
| 000F | HFP-floating-point-divide exception |
| 0010 | Segment-translation exception |
| 0011 | Page-translation exception |
| 0012 | Translation-specification exception |
| 0013 | Special-operation exception |
| 0015 | Operand exception |
| 0016 | Trace-table exception |
| 001C | Space-switch event |
| 001D | HFP-square-root exception |
| 001F | PC-translation-specification exception |
| 0020 | AFX-translation exception |
| 0021 | ASX-translation exception |
| 0022 | LX-translation exception |
| 0023 | EX-translation exception |
| 0024 | Primary-authority exception |
| 0025 | Secondary-authority exception |
| 0028 | ALET-specification exception |
| 0029 | ALEN-translation exception |
| 002A | ALE-sequence exception |
| 002B | ASTE-validity exception |
| 002C | ASTE-sequence exception |
| 002D | Extended-authority exception |
| 0030 | Stack-full exception |
| 0031 | Stack-empty exception |
| 0032 | Stack-specification exception |
| 0033 | Stack-type exception |
| 0034 | Stack-operation exception |
| 0038 | ASCE-type exception |
| 0039 | Region-first-translation exception |
| 003A | Region-second-translation exception |
| 003B | Region-third-translation exception |
| 0040 | Monitor event |
| 0080 | PER event (code may be combined with another code) |
| 0119 | Crypto-operation exception |

## Translation-Exception Identification

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

| Interruption Code (Hex) | Exception or Event | Format of Information Stored* |
|---|---|---|
| 0004 | Protection | If 61 zero: rest unpredictable<br>If 61 one: suppression, 0-51 address; if DAT was on, 60 one if access-list-controlled protection, 62-63 ASCE identification, rest unpredictable, location 160 valid; if DAT was off, rest unpredictable |
| 0010, 0038, 0039, 003A, 003B | Segment translation; ASCE type; region-first, second, third 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<br>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 |

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

## Data-Exception Code (DXC)

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

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

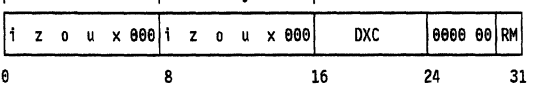
# Control Registers

| CR | Bits | Name of Field | Associated with | Init* |
|---|---|---|---|---|
| 0 | 33 | SSM-suppression control | SSM instruction | 0 |
| | 34 | TOD-clock-sync control | TOD clock | 0 |
| | 35 | Low-address-protection control | Low-address protection | 0 |
| | 36 | Extraction-authority control | Instruction authorization | 0 |
| | 37 | Secondary-space control | Instruction authorization | 0 |
| | 38 | Fetch-protection-override control | Key-controlled protection | 0 |
| | 39 | Storage-protection-override control | Key-controlled protection | 0 |
| | 45 | AFP-register control | Floating point | 0 |
| | 48 | Malfunction-alert subclass mask | External interruptions | 0 |
| | 49 | Emergency-signal subclass mask | External interruptions | 0 |
| | 50 | External-call subclass mask | External interruptions | 0 |
| | 52 | Clock-comparator subclass mask | External interruptions | 0 |
| | 53 | CPU-timer subclass mask | External interruptions | 0 |
| | 54 | Service-signal subclass mask | External interruptions | 0 |
| | 56 | Unused (See note) | | 1 |
| | 57 | Interrupt-key subclass mask | External interruptions | 1 |
| | 58 | Unused (See note) | | 1 |
| | 59 | ETR subclass mask | External interruptions | 0 |
| | 61 | Crypto control | Cryptography | 0 |
| 1 | 0-63 | Primary address-space-control element | Dynamic address translation | 0 |
| | 0-51 | Primary region-table or segment-table origin or real-space token origin | Dynamic address translation | 0 |
| | 54 | Primary subspace-group control | Subspace groups | 0 |
| | 55 | Primary private-space control | Dynamic address translation | 0 |
| | 56 | Primary storage-alteration-event control | Program-event recording | 0 |
| | 57 | Primary space-switch-event control | Program interruptions | 0 |
| | 58 | Primary real-space control | Dynamic address translation | 0 |
| | 60-61 | Primary designation-type control | Dynamic address translation | 0 |
| | 62-63 | Primary table length | Dynamic address translation | 0 |
| 2 | 33-57 | Dispatchable-unit-control-table origin | Access-register translation | 0 |
| 3 | 32-47 | PSW-key mask | Instruction authorization | 0 |
| | 48-63 | Secondary ASN | Address spaces | 0 |
| 4 | 32-47 | Authorization index | Instruction authorization | 0 |
| | 48-63 | Primary ASN | Address spaces | 0 |
| 5 | 33-57 | Primary-ASTE origin | Access-register translation | 0 |
| 6 | 32-39 | I/O-interruption subclass mask | I/O interruptions | 0 |
| 7 | 0-63 | Secondary address-space-control element | Dynamic address translation | 0 |
| | 0-51 | Secondary region-table or segment-table origin or real-space token origin | Dynamic address translation | 0 |
| | 54 | Secondary subspace-group control | Subspace groups | 0 |
| | 55 | Secondary private-space control | Dynamic address translation | 0 |
| | 56 | Secondary storage-alteration-event control | Program-event recording | 0 |
| | 58 | Secondary real-space control | Dynamic address translation | 0 |
| | 60-61 | Secondary designation-type control | Dynamic address translation | 0 |
| | 62-63 | Secondary table length | Dynamic address translation | 0 |
| 8 | 32-47 | Extended authorization index | Access-register translation | 0 |
| | 48-63 | Monitor masks | MC instruction | 0 |
| 9 | 32 | Successful-branching-event mask | Program-event recording | 0 |
| | 33 | Instruction-fetching-event mask | Program-event recording | 0 |
| | 34 | Storage-alteration-event mask | Program-event recording | 0 |
| | 36 | Store-using-real-address-event mask | Program-event recording | 0 |
| | 40 | Branch-address control | Program-event recording | 0 |
| | 42 | Storage-alteration-space control | Program-event recording | 0 |
| 10 | 0-63 | PER starting address | Program-event recording | 0 |

| 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.

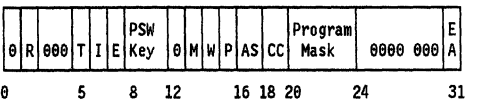
## Floating-Point-Control (FPC) Register

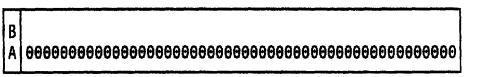


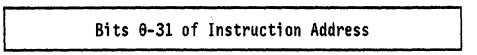

| Bit | Meaning |
|---|---|
| 0 | (IMi) IEEE-invalid-operation mask |
| 1 | (IMz) IEEE-division-by-zero mask |
| 2 | (IMo) IEEE-overflow mask |
| 3 | (IMu) IEEE-underflow mask |
| 4 | (IMx) IEEE-inexact mask |
| 8 | (SFi) IEEE-invalid-operation flag |
| 9 | (SFz) IEEE-division-by-zero flag |
| 10 | (SFo)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

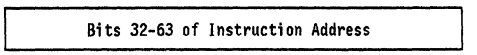
# Program-Status Word (PSW)

## z/Architecture PSW



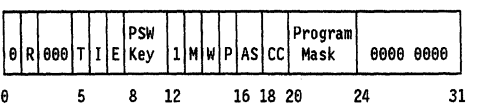



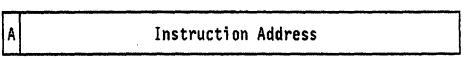




| Bit | Meaning |
|---|---|
| 1 | (R) Program-event-recording mask |
| 5 | (T = 1) DAT mode |
| 6 | (I) Input/output mask |
| 7 | (E) External mask |
| 12 | Zero indicates z/Architecture |
| 13 | (M) Machine-check mask |
| 14 | (W = 1) Wait state |
| 15 | (P = 1) Problem state |
| 16-17 | (AS) Address-space control<br>xx Real mode (T = 0)<br>00 Primary-space mode (T = 1)<br>01 Access-register mode (T = 1)<br>10 Secondary-space mode (T = 1)<br>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<br>00 24-bit mode<br>01 31-bit mode<br>10 Invalid<br>11 64-bit mode |

## ESA/390 PSW

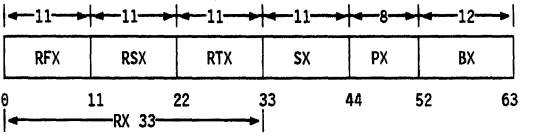



| Bit | Meaning |
|---|---|
| 12 | One indicates ESA/390 |
| 32 | (A = 1) 31-bit addressing mode |

# Dynamic Address Translation

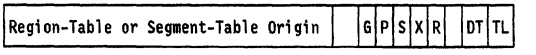
## Virtual-Address Format




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

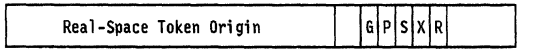
## Address-Space-Control Element (ASCE)

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




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

## Real-Space Designation (RSD)




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

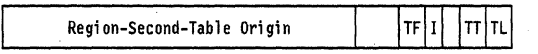
**Note:** Other bits are as in RTD or STD.

## Table Values

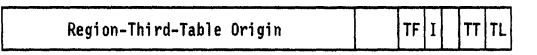
| Table | Incre-ments | Incr. Size | Incr. Entries | Max Size | Max Entries | Max Table Maps | |
|---|---|---|---|---|---|---|---|
| | | | | | | Regions | Bytes |
| Region First | 1-4 | 4KB | 512 | 16KB | 2K | 8G | $16E=16\times2^{60}$ |
| Region Second | 1-4 | 4KB | 512 | 16KB | 2K | 4M | $8P=8\times2^{50}$ |
| Region Third | 1-4 | 4KB | 512 | 16KB | 2K | 2K | $4T=4\times2^{40}$ |
| Segment | 1-4 | 4KB | 512 | 16KB | 2K | 1 | $2G=2\times2^{30}$ |
| Page | 1 | 2KB | 256 | 2KB | 256 | - | $1M=2^{20}$ |

## Region-Table Entry (RTE)

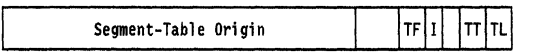
Region-First-Table Entry (RFTE)




Region-Second-Table Entry (RSTE)

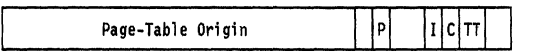



Region-Third-Table Entry (RTTE)



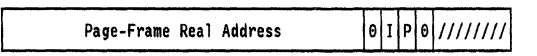

| Bit | Meaning |
|---|---|
| 56-57 | (TF) Table offset (for next-lower-level table) |
| 58 | (I) Invalid bit (for set of regions in RFTE or RSTE, or for region in RTTE) |
| 60-61 | (TT) Table-type bits (for this table)<br>11 Region first table<br>10 Region second table<br>01 Region third table |
| 62-63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

## Segment-Table Entry (STE)




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

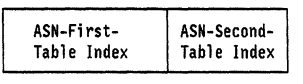
## Page-Table Entry (PTE)



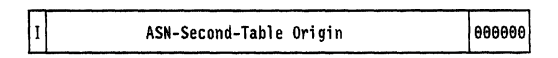

| Bit | Meaning |
|---|---|
| 53 | (I) Page-invalid bit |
| 53 | (P) Page-protection bit |

# ASN Translation

## Address-Space Number (ASN)

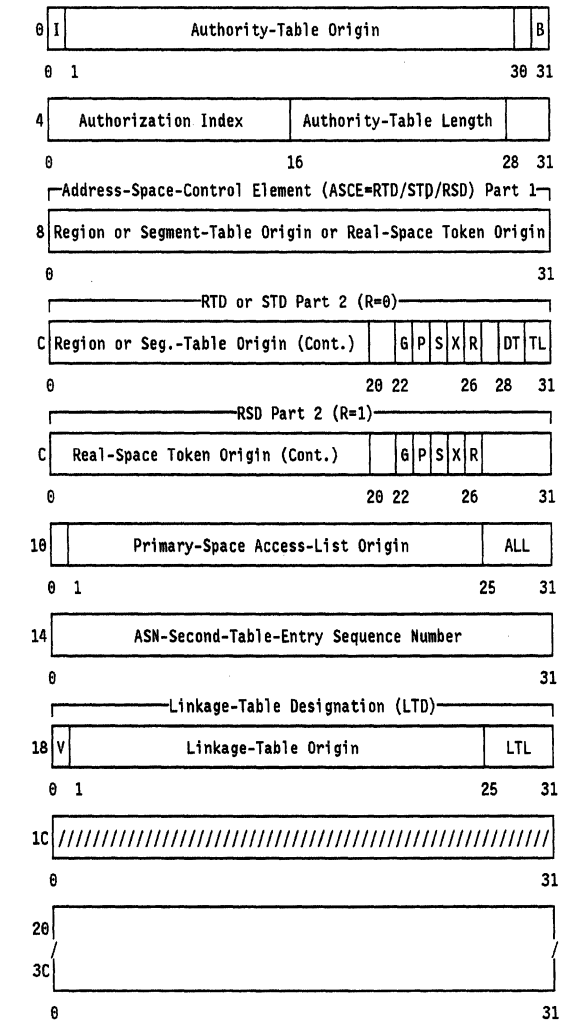


## ASN-First-Table Entry




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

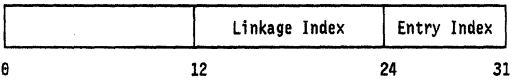
## ASN-Second-Table Entry (ASTE)



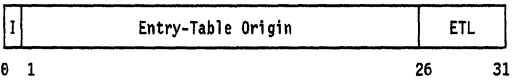

| Byte.Bit | Meaning |
|---|---|
| 0.0 | (I) ASX-invalid bit |
| 0.31 | (B) Base-space bit |
| 10.25-31 | (ALL) Access-list length (x 128 bytes) |
| 18.0 | (V) Subsystem-linkage control |
| 18.25-31 | (LTL) Linkage-table length (x 128 bytes) |

# PC-Number Translation

## Program-Call Number




## Linkage-Table Entry (LTE)

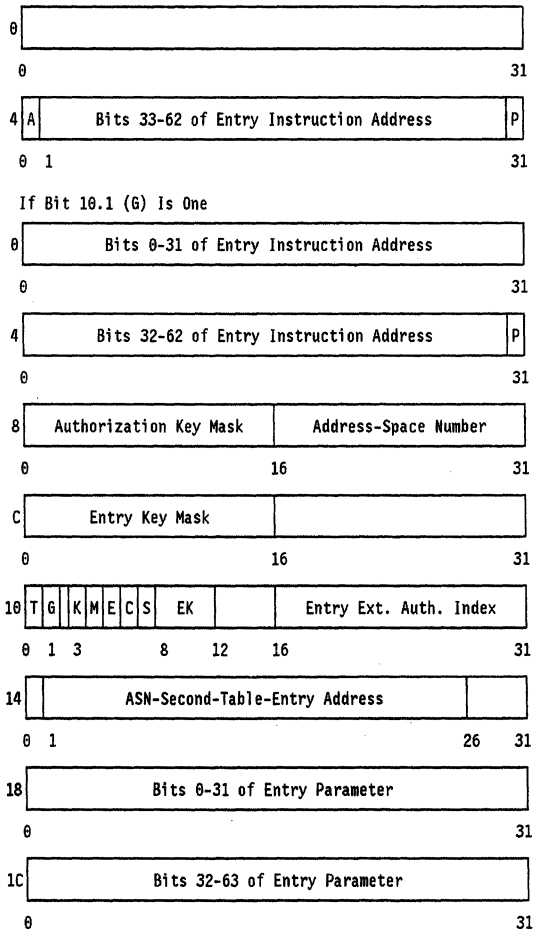



| Bit | Meaning |
|---|---|
| 0 | (I) LX-invalid bit |
| 26-31 | (ETL) Entry-table length (x 128 bytes) |

## Entry-Table Entry (ETE)

Byte (Hex)

If Bit 10.1 (G) Is Zero



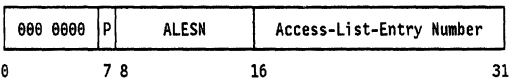

If Bit 10.1 (G) Is One

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

| Byte.Bit | Meaning |
|---|---|
| 4.0 | (A) Entry addressing mode |
| 4.31 | (P) Entry problem state |
| 10.0 | (T) PC-type bit (zero: basic; one: stacking) |
| 10.1 | (G) Entry extended addressing mode |
| 10.3 | (K) PSW-key control (zero: unchanged; one: replace if stacking |
| 10.4 | (M) PSW-key-mask control (zero: Or; one: replace if stacking) |
| 10.5 | (E) EAX control (zero: unchanged; one: replace if stacking) |
| 10.6 | (C) Address-space-control control |
| 10.7 | (S) Secondary-ASN control |
| 10.8-11 | (EK) Entry key |

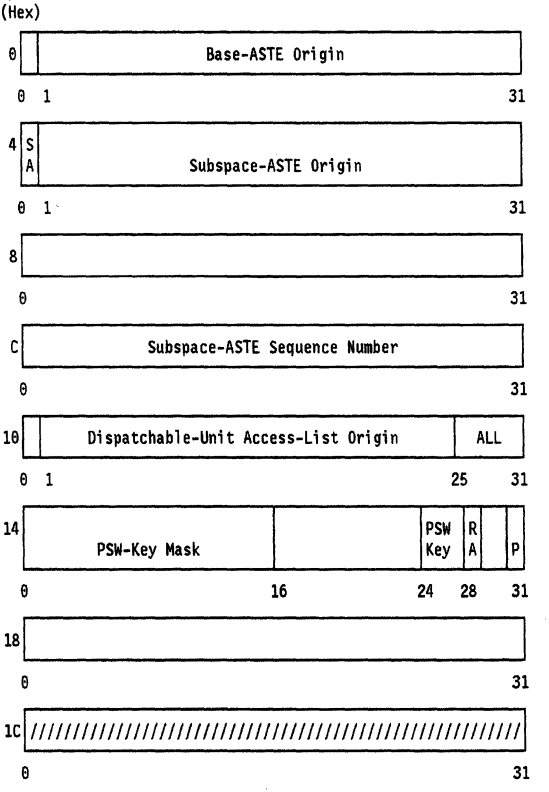
# Access-Register Translation

## Access-List-Entry Token (ALET)



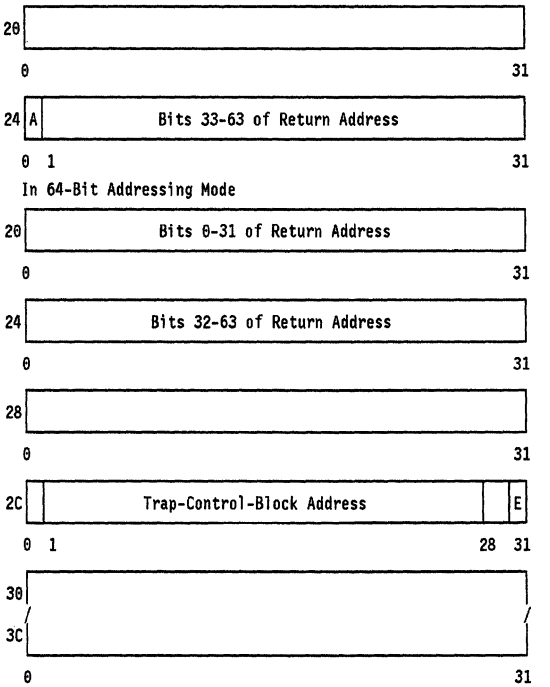

| Bit | Meaning |
|---|---|
| 7 | (P) Primary-list bit (zero: use DUCT; one: use primary ASTE) |
| 8-15 | (ALESN) Access-list-entry sequence number |

## Dispatchable-Unit-Control Table (DUCT)



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

In 24-Bit or 31-Bit Addressing Mode

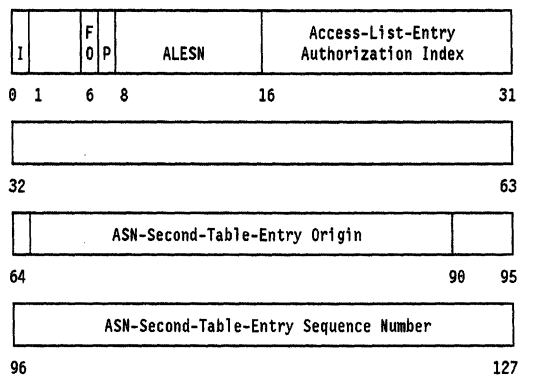



In 64-Bit Addressing Mode




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

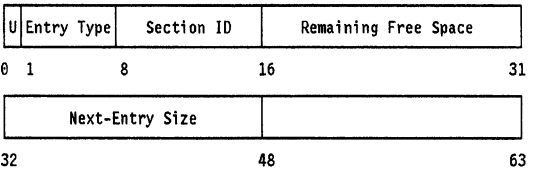
## Access-List Entry (ALE)




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

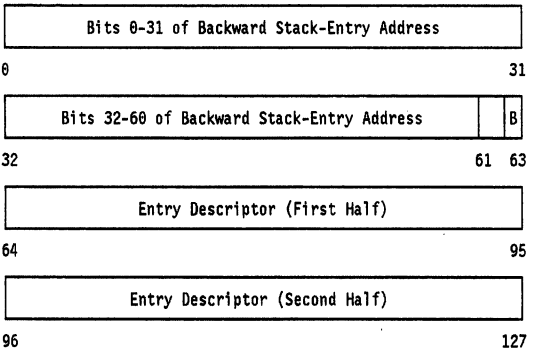
# Linkage-Stack Entries

## Entry Descriptor



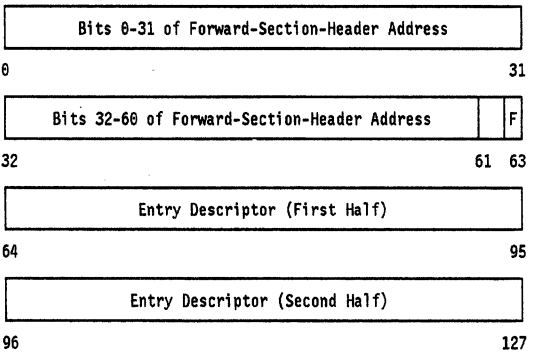

| Bit | Meaning |
|---|---|
| 0 | (U) Unstack-suppression bit |
| 1-7 | Entry type:<br>Header entry = 0001001 binary<br>Trailer entry = 0001010 binary<br>Branch state entry = 0001100 binary<br>Program-call state entry = 0001101 binary<br>Available for program use = 1xxxxxx binary |

## Header Entry (Entry Type 0001001)



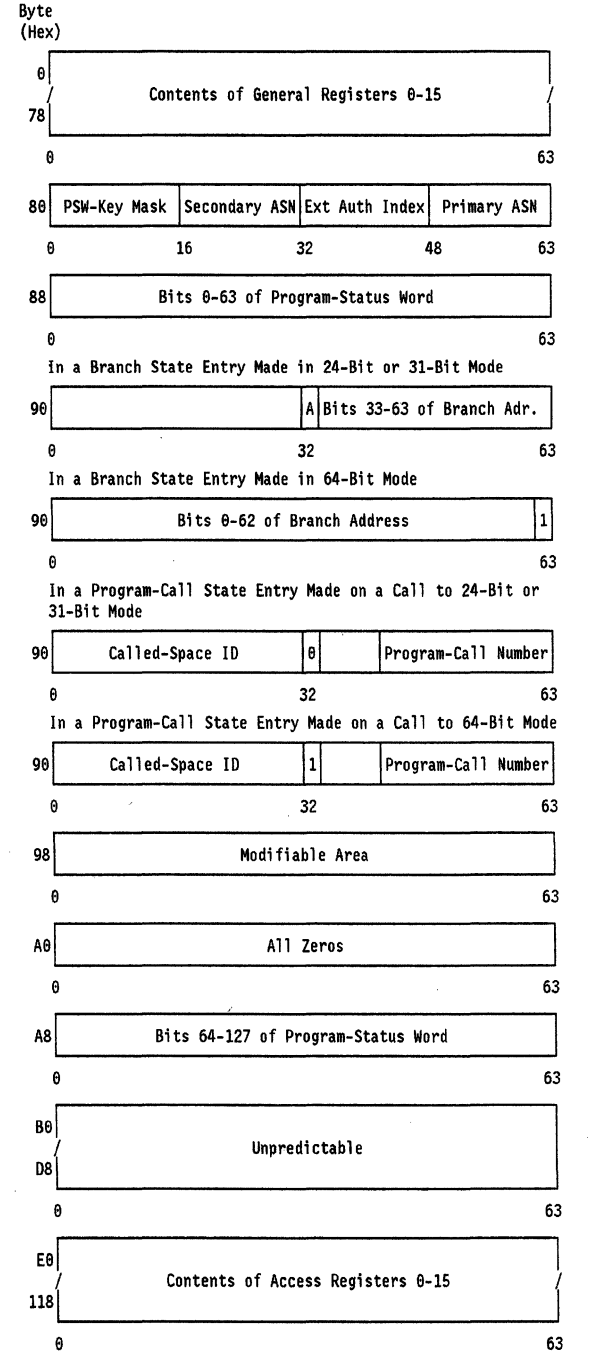

| Bit | Meaning |
|---|---|
| 63 | (B) Backward stack-entry validity bit |

## Trailer Entry (Entry Type 0001010)

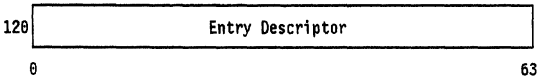



| Bit | Meaning |
|---|---|
| 63 | (F) Forward-section validity bit |

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



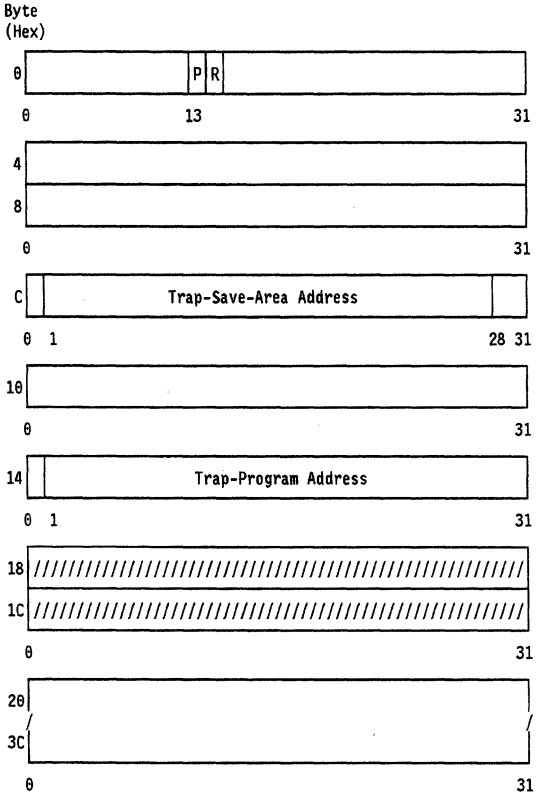
Branch State Entry (Entry Type 0001100) and
Program-Call State Entry (Entry Type 0001101) (Cont'd)




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

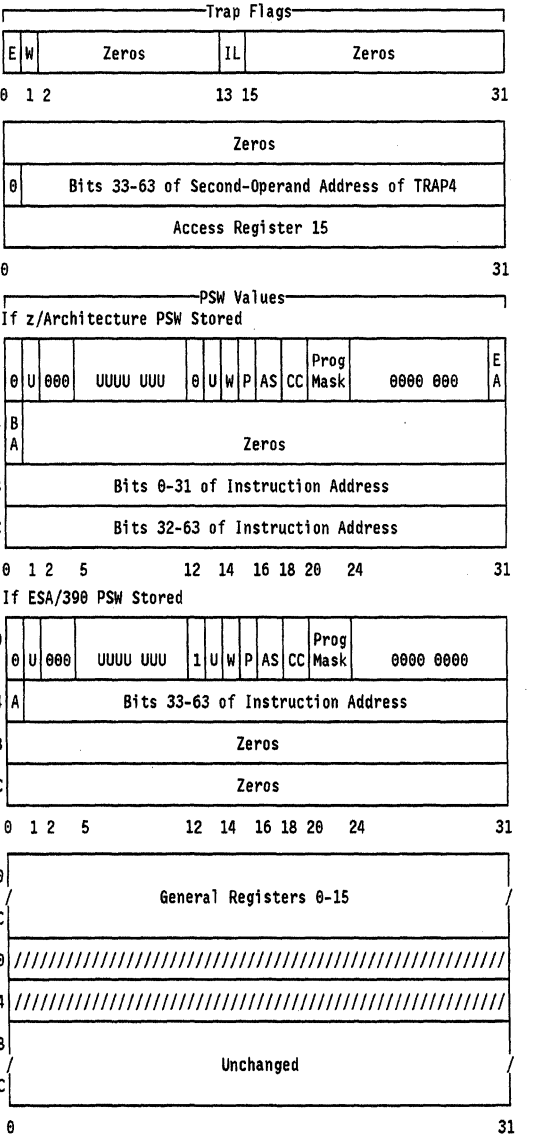
# Trapping

## Trap Control Block




| Byte.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.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 26) |
| U | Unpredictable |
| /// | Available for programming |

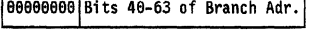
# Trace-Entry Formats

## Identification of Trace Entries

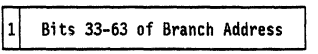
| Trace-Entry Bits | | | Trace Entry | |
|---|---|---|---|---|
| 0-7 | 8-11 | 12-15 | Type | For-mat |
| 00000000 | | | Branch | 1 |
| 00010000 | | | Set Secondary ASN | 1 |
| 00100001 | | | Program Call | 1 |
| 00100010 | | | Program Call | 2 |
| 00110001 | | 0000 | Program Transfer | 1 |
| 00110001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

## Branch

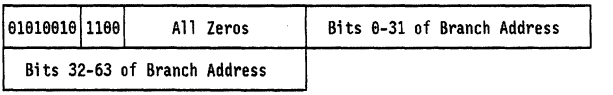
F1 (Branch, RP, or TRAP2/4 to 24-Bit Mode)



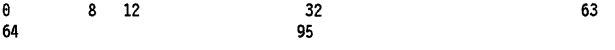

F2 (Branch, RP, or TRAP2/4 to 31/64-Bit Mode)




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




**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 |
|---|---|---|---|---|

F2 (in 64-Bit Mode)

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

## Mode Switch

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

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

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

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

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

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

## Mode-Switching Branch

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

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

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

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

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

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

## Program Call

F1 (in 24/31-Bit Mode)

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

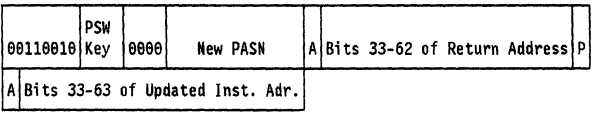
F2 (in 64-Bit Mode)

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

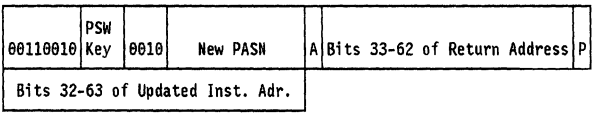
0 8 12 32 63
64 95

## Program Return

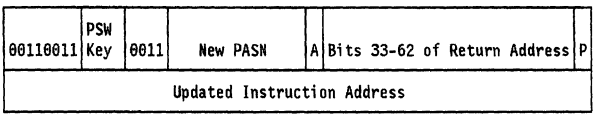
F1 (in 24/31-Bit to 24/31-Bit Mode)



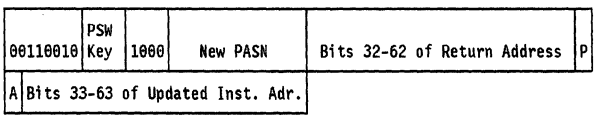

F2 (in 64-Bit to 24/31-Bit Mode)



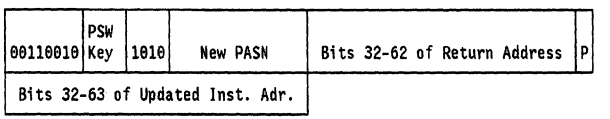

F3 (in 64-Bit to 24/31-Bit Mode)



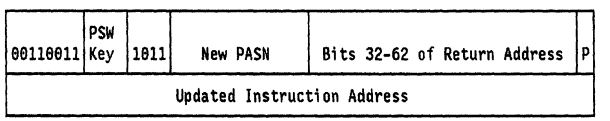

F4 (in 24/31-Bit to 64-Bit Mode)



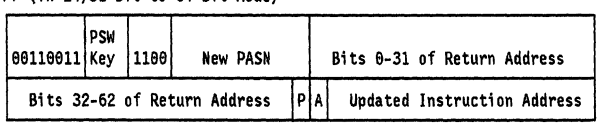

F5 (in 64-Bit to 64-Bit Mode)



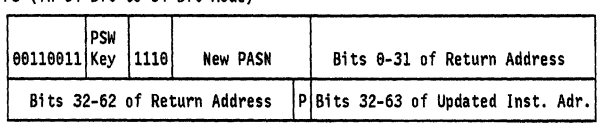

F6 (in 64-Bit to 64-Bit Mode)

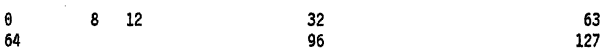



F7 (in 24/31-Bit to 64-Bit Mode)




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



Program Return (Cont'd)

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

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

## Program Transfer

F1 (in 24/31-Bit Mode)

| | | | | |
|---|---|---|---|---|
| 00110001 | PSW Key | 0000 | New PASN | Bits 32-63 of $R_2$ Before |

F2 (in 64-Bit Mode)

| | | | | |
|---|---|---|---|---|
| 00110001 | PSW Key | 1000 | New PASN | Bits 32-63 of $R_2$ Before |

F3 (in 64-Bit Mode)

| | | | | |
|---|---|---|---|---|
| 00110010 | PSW Key | 1100 | New PASN | Bits 0-31 of $R_2$ Before |
| Bits 32-63 of $R_2$ Before | | | | |

## Set Secondary ASN

F1

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

## Trace

F1 (TRACE)

| | | | | |
|---|---|---|---|---|
| 0111 | N | 00000000 | TOD-Clock Bits 16-63 | |
| TRACE Operand | | | | ($R_1$) - ($R_3$) |

F2 (TRACG)

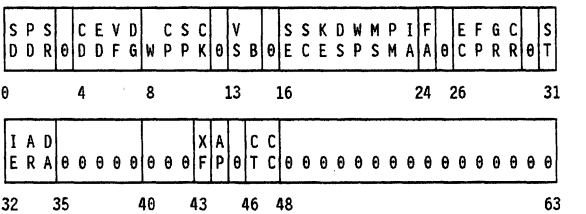
| | | | | |
|---|---|---|---|---|
| 0111 | N | 10000000 | TOD-Clock Bits 0-47 | |
| TOD-Clock Bits 48-79 | | | | TRACE Operand |
| ($R_1$) - ($R_3$) | | | | |

0 8 12 32 63
64 96 127
128 159

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

## Machine-Check Interruption Code

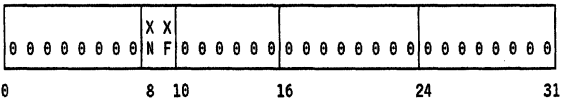
At real-storage locations 232-239 (E8-EF hex)




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

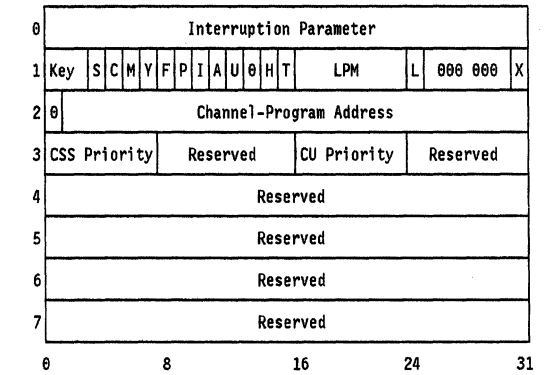
## External-Damage Code

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




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

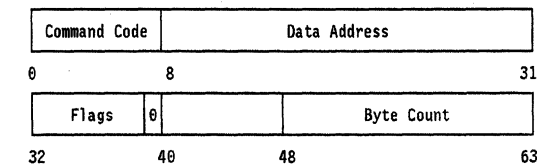
## Operation-Request Block (ORB)




| Word.Bit | Meaning |
|---|---|
| 1.0-3 | (Key) Subchannel key |
| 1.4 | (S) Suspend control |
| 1.5 | (C) Streaming-mode control |
| 1.6 | (M) Modification control |
| 1.7 | (Y) Synchronization control |
| 1.8 | (F) CCW-format control |
| 1.9 | (P) Prefetch control |
| 1.10 | (I) Initial-status-interruption control |
| 1.11 | (A) Address-limit-checking control |
| 1.12 | (U) Suppress-suspended-interruption control |
| 1.14 | (H) Format-2-IDAW control |
| 1.15 | (T) 2K-IDAW control |
| 1.16-23 | (LPM) Logical-path mask |
| 1.24 | (L) Incorrect-length-suppression mode |
| 1.31 | (X) ORB-extension control |
| 3.0-7 | Control-unit priority |
| 3.16-23 | Channel-subsystem priority |

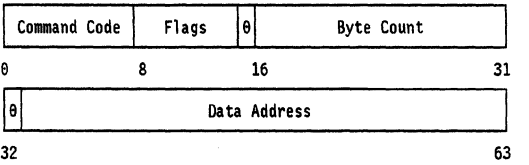
## Channel-Command Word (CCW)

### Format-0 CCW




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

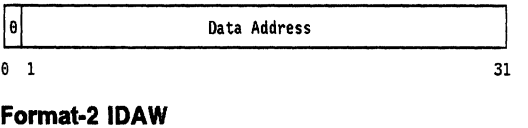
## Format-1 CCW



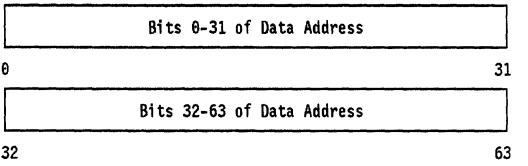

| Bit | Meaning |
|---|---|
| 8 | (CD) Causes use of data-address portion of next CCW |
| 9 | (CC) Causes use of command code and data address of next CCW |
| 10 | (SLI) Causes suppression of possible incorrect-length indication |
| 11 | (Skip) Suppresses transfer of information to main storage |
| 12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

# Indirect-Data-Address Word (IDAW)

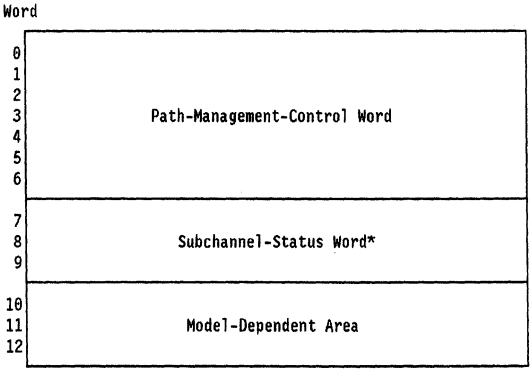
## Format-1 IDAW




## Format-2 IDAW



## Subchannel-Information Block (SCHIB)



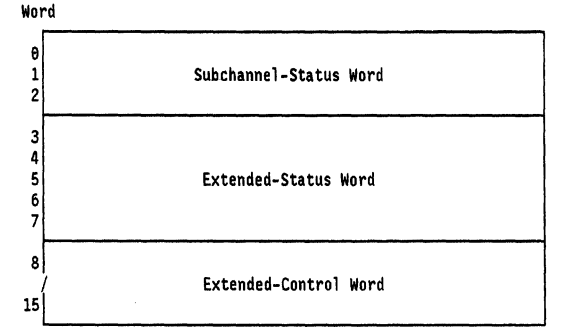

*See "Subchannel-Status Word (SCSW)" on page 45.

## Path-Management-Control Word (PMCW)

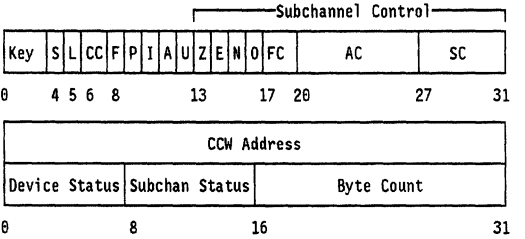
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | Interruption Parameter | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | 00 | | ISC | | | 000 | | | E | | LM | | MM | | D | | T | | V | Device Number | | | | | | | | | | |
| 2 | LPM | | | | | | | | PNOM | | | | | | | | LPUM | | | | | | | | PIM | | | | | | |
| 3 | MBI | | | | | | | | | | | | | | | | POM | | | | | | | | PAM | | | | | | |
| 4 | CHPID-0 | | | | | | | | CHPID-1 | | | | | | | | CHPID-2 | | | | | | | | CHPID-3 | | | | | | |
| 5 | CHPID-4 | | | | | | | | CHPID-5 | | | | | | | | CHPID-6 | | | | | | | | CHPID-7 | | | | | | |
| 6 | 0000 0000 | | | | | | | | 0000 0000 | | | | | | | | 0000 0000 | | | | | | | | 0000 000 | | | | | | 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<br>00 No Checking<br>01 Data address must be ≥ limit<br>10 Data address must be < limit<br>11 Reserved |
| 1.11-12 | (MM) Measurement-mode enable<br>00 Neither mode enabled<br>01 Device-connect-time-measurement enabled<br>10 Measurement-block-update enabled<br>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) |
| 6.31 | (S) Concurrent sense |

## Interruption-Response Block (IRB)




## Subchannel-Status Word (SCSW)

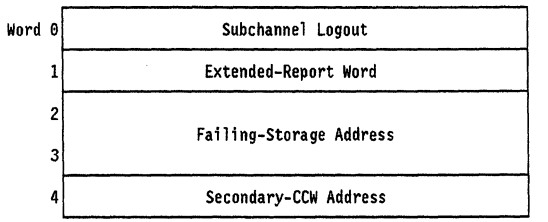



| Word. Bit | Meaning |
|---|---|
| 0.0-3 | (Key) Subchannel key |
| 0.4 | (S) Suspend control |
| 0.5 | (L) Extended-status-word format (logout stored) |
| 0.6-7 | CC) Deferred condition code<br>00 Normal I/O interruption<br>01 Status in SCSW<br>10 Reserved<br>11 Path not operational |
| 0.8 | (F) CCW-format 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<br>17 (40) start, 18 (20) halt, 19 (10) clear |
| 0.20-26 | (AC) Activity control<br>20 (08) resume pending — 24 (80) subchannel active<br>21 (04) start pending — 25 (40) device active<br>22 (02) halt pending — 26 (20) suspended<br>23 (01) clear pending |
| 0.27-31 | (SC) Status control<br>27 (10) alert — 30 (02) secondary<br>28 (08) intermediate — 31 (01) status pending<br>29 (04) primary |
| 2.0-15 | Device status (0-7), subchannel status (8-15)<br>0 (80) Attention — 8 (80) Prog.-cont. int.<br>1 (40) Status modifier — 9 (40) Incorrect length<br>2 (20) Control-unit end — 10 (20) Program check<br>3 (10) Busy — 11 (10) Protection check<br>4 (08) Channel end — 12 (08) Channel-data check<br>5 (04) Device end — 13 (04) Channel-control check<br>6 (02) Unit check — 14 (02) Interface-control check<br>7 (01) Unit exception — 15 (01) Chaining check |

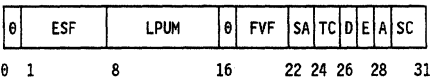
## Extended-Status Word (ESW)

See chart on page 47 to determine the appropriate ESW format.

## Format-0 ESW

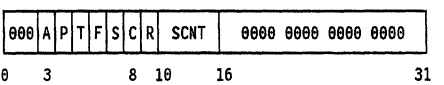



## Format-0 ESW Word 0 (Subchannel Logout)



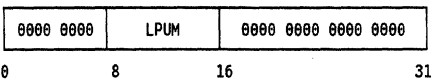

| Bit | Meaning |
|---|---|
| 1-7 | (ESF) Extended-status flags (1 key check, 2 measurement-block program check, 3 measurement-block data check, 4 measurement -block protection check, 5 CCW check, 6 IDAW check, 7:0) |
| 8-15 | (LPUM) Last-path-used mask |
| 17-21 | (FVF) Field-validity flags (17 LPUM,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

## Format-0 ESW Word 1 (Extended-Report Word)




| Bit | Meaning |
|---|---|
| 3 | (A) Authorization check |
| 4 | (P) Path-verification-required |
| 5 | (T) Channel-path timeout |
| 6 | (F) Failing-storage-address validity |
| 7 | (S) Concurrent sense |
| 8 | (C) Secondary-CCW-address validity |
| 9 | (R) Failing-storage-address format (zero: 1-31 of word 2; one: words 2 and 3) |
| 10-15 | (SCNT) Concurrent-sense count |

## Format-1 ESW Word 0*




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

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

## Format-2 ESW Word 0*

| 0000 0000 | 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*

| 0000 0000 | 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 | | | | |
| Status-Control Field AIPSX | L Bit | Suspended Bit | Timing-Facility Bit | Device-Connect-Time Measurement-Mode-Enable Bit | Device-Connect-Time Measurement-Mode Active | Format | Contents Word 0 Byte 0123 |
| ----0 | - | * | * | * | No/Yes | U | **** |
| **001 | 1 | * | * | * | No/Yes | 0 | RRRR |
| **1*1 | 1 | * | * | * | No/Yes | 0 | RRRR |
| 10011 | 1 | * | * | * | No/Yes | 0 | RRRR |
| 00001 | 0 | * | * | * | No/Yes | U | **** |
| 00011 | 0 | * | * | * | No/Yes | 3 | ZM** |
| 100*1 | 0 | * | * | * | No/Yes | 3 | ZM** |
| **1*1 | 0 | * | 0 | * | No/Yes | 1 | ZMZZ |
| **1*1 | 0 | * | 1 | 0 | No/Yes | 1 | ZMZZ |
| **1*1 | 0 | * | 1 | 1 | No | 1 | ZMZZ |
| **1*1 | 0 | * | 1 | 1 | Yes | 2 | ZMDD |
| 01001 | 0 | 0 | * | * | No/Yes | U | **** |
| 01001 | 0 | 1 | 0 | * | No/Yes | 1 | ZMZZ |
| 01001 | 0 | 1 | 1 | 0 | No/Yes | 1 | ZMZZ |
| 01001 | 0 | 1 | 1 | 1 | No | 1 | ZMZZ |
| 01001 | 0 | 1 | 1 | 1 | Yes | 2 | ZMDD |
| 00011 | 1 | These combinations do not occur. | | | | | |
| 11001 | 0 | | | | | | |
| *1011 | * | | | | | | |

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

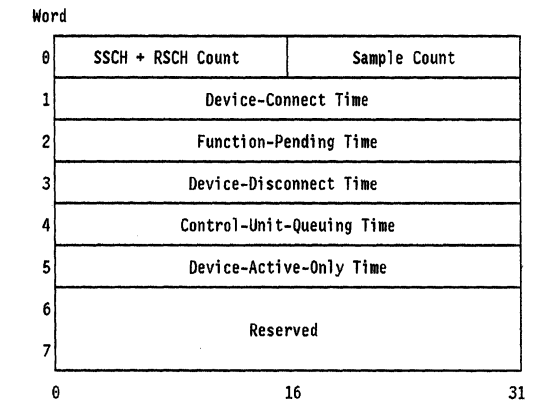
*Word 1 is the same as word 1 of a format-0 ESW. Words 2, 3, and 4 are zeros.

## Extended-Control Word (ECW)

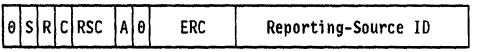
| SCSW Bits 5 | SCSW Bits 14 | ERW Bit 7 | ERW Bits 10-15 | ECW Words 0-7 |
|---|---|---|---|---|
| 0 | 0 | 0 | Zeros | Unpredictable |
| 0 | 1 | 1 | No. of con-sen* bytes | Concurrent-sense information* |
| 1 | 0 | 0 | Zeros | Unpredictable |
| 1 | 1 | 0 | Zeros | Model-dependent information |
| 1 |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

*The 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.

## Measurement Block




## Channel-Report Word (CRW)




| Bit | Meaning |
|---|---|
| 1 | (S) Solicited CRW |
| 2 | (R) Overflow (one or more CRWs lost) |
| 3 | (C) Chaining (meaningless if bit 2 is one) |
| 4-7 | (RSC) Reporting-source code (see Reporting-Source table) |
| 8 | (A) Ancillary report |
| 10-15 | (ERC) Error-recovery code (see Error-Recovery-Code table) |
| 16-31 | Reporting-source ID (see Reporting-Source table) |

## Error-Recovery Codes

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

## Reporting Source

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

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

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

# I/O Command Codes

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

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

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

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

## Standard Meanings of Bits of First Sense Byte

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

## Code Assignments

### Code Table

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

BookMaster is a trademark of the International Business Machines Corporation.

| Dec | Hex | EBCDIC | ASCII (1) | ISO -8 | IBM-PC | BookMaster Symbol Names(2) |
|---|---|---|---|---|---|---|
| 40 | 28 | SA | ( | ( | ( | lpar |
| 41 | 29 | SFE | ) | ) | ) | rpar |
| 42 | 2A | SM/SW | * | * | * | asterisk |
| 43 | 2B | CSP | + | + | + | plus |
| 44 | 2C | MFA | , | , | , | comma |
| 45 | 2D | ENQ | - | - | - | hyphen or minus |
| 46 | 2E | ACK | . | . | . | period |
| 47 | 2F | BEL | / | / | / | divslash or slash |
| 48 | 30 | | 0 | 0 | 0 | |
| 49 | 31 | | 1 | 1 | 1 | |
| 50 | 32 | SYN | 2 | 2 | 2 | |
| 51 | 33 | IR | 3 | 3 | 3 | |
| 52 | 34 | PP | 4 | 4 | 4 | |
| 53 | 35 | TRN | 5 | 5 | 5 | |
| 54 | 36 | NBS | 6 | 6 | 6 | |
| 55 | 37 | EOT | 7 | 7 | 7 | |
| 56 | 38 | SBS | 8 | 8 | 8 | |
| 57 | 39 | IT | 9 | 9 | 9 | |
| 58 | 3A | RFF | : | : | : | colon |
| 59 | 3B | CU3 | ; | ; | ; | semi |
| 60 | 3C | DC4 | < | < | < | lt |
| 61 | 3D | NAK | = | = | = | eq |
| 62 | 3E | | > | > | > | gt |
| 63 | 3F | SUB | ? | ? | ? | quest |

| Dec | Hex | See Next Page | See Above | See Above |
|---|---|---|---|---|
| 64 | 40 | SP SP SP SP SP | @ @ @ | atsign |
| 65 | 41 | RSP RSP RSP RSP RSP | A A A | |
| 66 | 42 | â â â | B B B | ac |
| 67 | 43 | ä ä ä | C C C | ae |
| 68 | 44 | à à à | D D D | ag |
| 69 | 45 | á á á | E E E | aa |
| 70 | 46 | ã ã ã | F F F | at |
| 71 | 47 | å å å | G G G | ao |
| 72 | 48 | ç ç ç | H H H | cc |
| 73 | 49 | ñ ñ ñ | I I I | nt |
| 74 | 4A | ¢ ¢ [ ¢ | J J J | cent, lbrk |
| 75 | 4B | . . . . . | K K K | period |
| 76 | 4C | < < < < < | L L L | lt |
| 77 | 4D | ( ( ( ( ( | M M M | lpar |
| 78 | 4E | + + + + + | N N N | plus |
| 79 | 4F | \| \| ! \| | O O O | vbar, xclam |
| 80 | 50 | & & & & & | P P P | amp |
| 81 | 51 | é é é | Q Q Q | ea |
| 82 | 52 | ê ê ê | R R R | ec |
| 83 | 53 | ë ë ë | S S S | ee |

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

| Dec | Hex | EBCDIC(4) 81C | 94C | 837 | 500 | 1047 | ISO IBM-PC -8 | 437 | 850 | BookMaster Symbol Names(2) |
|---|---|---|---|---|---|---|---|---|---|---|
| 128 | 80 | | | Ø | Ø | Ø | | Ç | Ç | Os, Cc |
| 129 | 81 | a | a | a | a | a | | ü | ü | ue |
| 130 | 82 | b | b | b | b | b | BPH | é | é | ea |
| 131 | 83 | c | c | c | c | c | NBH | â | â | ac |
| 132 | 84 | d | d | d | d | d | IND | ä | ä | ae |
| 133 | 85 | e | e | e | e | e | NEL | à | à | ag |
| 134 | 86 | f | f | f | f | f | SSA | å | å | ao |
| 135 | 87 | g | g | g | g | g | ESA | ç | ç | cc |
| 136 | 88 | h | h | h | h | h | HTS | ê | ê | ec |
| 137 | 89 | i | i | i | i | i | HTJ | ë | ë | ee |
| 138 | 8A | | | « | « | « | VTS | è | è | odqf, eg |
| 139 | 8B | | | » | » | » | PLD | ë | ë | cdqf, ee |
| 140 | 8C | | | ð | ð | ð | PLU | î | î | eth, ic |
| 141 | 8D | | | ý | ý | ý | RI | ì | ì | ya, ig |
| 142 | 8E | | | þ | þ | þ | SS2 | Ä | Ä | thorn, Ae |
| 143 | 8F | | | ± | ± | ± | SS3 | Å | Å | pm, Ao |
| 144 | 90 | | | ° | ° | ° | DCS | É | É | degree, Ea |
| 145 | 91 | j | j | j | j | j | PU1 | æ | æ | aelig |
| 146 | 92 | k | k | k | k | k | PU2 | Æ | Æ | AElig |
| 147 | 93 | l | l | l | l | l | STS | ô | ô | oc |
| 148 | 94 | m | m | m | m | m | CCH | ö | ö | oe |
| 149 | 95 | n | n | n | n | n | MW | ò | ò | og |
| 150 | 96 | o | o | o | o | o | SPA | û | û | uc |
| 151 | 97 | p | p | p | p | p | EPA | ù | ù | ug |
| 152 | 98 | q | q | q | q | q | SOS | ÿ | ÿ | ye |
| 153 | 99 | r | r | r | r | r | | Ö | Ö | Oe |
| 154 | 9A | | | ª | ª | ª | SCI | Ü | Ü | aus, Ue |
| 155 | 9B | | | º | º | º | CSI | ¢ | ø | ous, cent, os |
| 156 | 9C | | | æ | æ | æ | ST | £ | £ | aelig, Lsterling |
| 157 | 9D | | | | | | OSC | ¥ | Ø | cedilla, yen, Os |
| 158 | 9E | | | Æ | Æ | Æ | PM | Pt | × | AElig, peseta, mult |
| 159 | 9F | | | ¤ | ¤ | ¤ | ACP | *f* | *f* | currency, fnof(5) |
| 160 | A0 | | | μ | μ | μ | RSP | á | á | mu(6), aa |
| 161 | A1 | | | ~ | ~ | ~ | ¡ | í | í | tilde, inve, ia |
| 162 | A2 | s | s | s | s | s | ¢ | ó | ó | cent, oa |
| 163 | A3 | t | t | t | t | t | £ | ú | ú | Lsterling, ua |
| 164 | A4 | u | u | u | u | u | ¤ | ñ | ñ | currency, nt |
| 165 | A5 | v | v | v | v | v | ¥ | Ñ | Ñ | yen, Nt |
| 166 | A6 | w | w | w | w | w | ¦ | ª | ª | splitvbar, aus |
| 167 | A7 | x | x | x | x | x | § | º | º | section, ous |
| 168 | A8 | y | y | y | y | y | ¨ | ¿ | ¿ | umlaut, invq |
| 169 | A9 | z | z | z | z | z | © | ⌐ | ® | copyr, lnotrev, regtm |
| 170 | AA | | | ¡ | ¡ | ¡ | ª | ¬ | ¬ | inve, aus, lnot |
| 171 | AB | | | ¿ | ¿ | ¿ | « | ½ | ½ | invq, odqf, frac12 |

| Dec | Hex | EBCDIC(4) 81C | 94C | 037 | 500 | 1847 | ISO -8 | IBM-PC 437 | 850 | BookMaster Symbol Names(2) |
|---|---|---|---|---|---|---|---|---|---|---|
| 172 | AC | | | Ð | Ð | Ð | ¬ | ¼ | ¼ | Dstroke or Eth, lnot, frac14 |
| 173 | AD | | | Ý | Ý | [ | SHY | ¡ | ¡ | Ya, lbrk, inve |
| 174 | AE | | | þ | þ | þ | ® | « | « | Thorn, regtm, odqf |
| 175 | AF | | | ® | ® | ® | ¯ | » | » | regtm, overline, cdqf |
| 176 | B0 | | | ^ | ¢ | ¬ | ° | ⣿ | ⣿ | hat, cent, lnot, degree, box14 |
| 177 | B1 | | | £ | £ | £ | ± | ▒ | ▒ | Lsterling, pm, box12 |
| 178 | B2 | | | ¥ | ¥ | ¥ | ² | ╫ | ╫ | yen, sup2, box34 |
| 179 | B3 | | | · | · | · | ³ | │ | │ | smultdot, sup3, bxv |
| 180 | B4 | | | © | © | © | ´ | ┤ | ┤ | copyr, acute, bxrj |
| 181 | B5 | | | § | § | § | µ | ╡ | Á | section, mu(6), bx1012, Aa |
| 182 | B6 | | | ¶ | ¶ | ¶ | ¶ | ╢ | Â | par, bx2021, Ac |
| 183 | B7 | | | ¼ | ¼ | ¼ | · | ╖ | À | frac14, smultdot, bx0021, Ag |
| 184 | B8 | | | ½ | ½ | ½ | ¸ | ╕ | © | frac12, cedilla, bx0012, copyr |
| 185 | B9 | | | ¾ | ¾ | ¾ | ¹ | ╣ | ╣ | frac34, sup1, bx2022 |
| 186 | BA | | | [ | ¬ | Ý | º | ║ | ║ | lbrk, lnot, Ya, ous, bx2020 |
| 187 | BB | | | ] | \| | ¨ | » | ╗ | ╗ | rbrk, vbar, umlaut, cdqf, bx0022 |
| 188 | BC | | | | ¯ | ¯ | ¼ | ╝ | ╝ | overline, frac14, bx2002 |
| 189 | BD | | | ¨ | ¨ | ] | ½ | ╜ | ¢ | umlaut, rbrk, frac12, bx2001, cent |
| 190 | BE | | | ´ | ´ | ´ | ¾ | ╛ | ¥ | acute, frac34, bx1002, yen |
| 191 | BF | | | × | × | × | ¿ | ┐ | ┐ | mult, invq, bxur |
| 192 | C0 | | | | { | { | À | └ | └ | lbrc, Ag, bx11 |
| 193 | C1 | A | A | A | A | A | Á | ┴ | ┴ | Aa, bxbj |
| 194 | C2 | B | B | B | B | B | Â | ┬ | ┬ | Ac, bxtj |
| 195 | C3 | C | C | C | C | C | Ã | ├ | ├ | At, bxlj |
| 196 | C4 | D | D | D | D | D | Ä | ─ | ─ | Ae, bxh |
| 197 | C5 | E | E | E | E | E | Å | ┼ | ┼ | Ao, bxcj |
| 198 | C6 | F | F | F | F | F | Æ | ╞ | ã | AElig, bx1210, at |
| 199 | C7 | G | G | G | G | G | Ç | ╟ | Ã | Cc, bx2120, At |
| 200 | C8 | H | H | H | H | H | È | ╚ | À | Eg, bx2200, Ag |
| 201 | C9 | I | I | I | I | I | É | ╔ | ╔ | Ea, bx0220 |
| 202 | CA | SHY | SHY | SHY | SHY | SHY | Ê | ╩ | ╩ | Ec, bx2202 |
| 203 | CB | | | ô | ô | ô | Ë | ╦ | ╦ | oc, Ee, bx0222 |
| 204 | CC | | | ö | ö | ö | Ì | ╠ | ╠ | oe, Ig, bx2220 |
| 205 | CD | | | ò | ò | ò | Í | ═ | ═ | og, Ia, bx0202 |
| 206 | CE | | | ó | ó | ó | Î | ╬ | ╬ | oa, Ic, bx2222 |
| 207 | CF | | | õ | õ | õ | Ï | ¤ | ¤ | ot, Ie, bx1202, currency |

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

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

(1) The ASCII controls and graphics are from ANSI X3.4. The ISO-8 controls are from ISO 6429, and the graphics are from ISO 8859-1. The ISO-8 graphics are code page 00819, named ISO/ANSI Multilingual. IBM-PC controls and graphics are shown. The graphics are common to code page 00437, named Personal Computer, and code page 00850, named Personal Computer - Multilingual Page. Code pages 00437 and 00850 are shown separately beginning at X'80', after which they diverge in content.

(2) The symbol names shown are to be preceded by an ampersand (&) and followed by a period (.) to form a symbol. Source: SC34-5009.

(3) ASCII, ISO-8, and IBM-PC X'27' and EBCDIC X'7D' are an apostrophe having the appearance of a straight single quote. The BookMaster "apos" produces a character having the appearance of an accent acute.

(4) Five columns of EBCDIC graphics are shown. The first is the 81-character character set 0640, called the syntactic character set, that is mapped the same on all EBCDIC code pages. The second is the standard IBM 94-character character set mapped on code page 00037. The third is code page 00037, named USA/Canada - CECP (Country Extended Code Page). The fourth is code page 00500, named International #5. The fifth is code page 01047, named Latin 1/Open Systems. Code pages 00037, 00500, 01047, and 00819 (ISO-8) all map the 189-character character set 0697. Source: SE09-8002.

(5) $f$, ≈, and Θ are of nonstandard width.

(6) EBCDIC X'A0' and ISO-8 X'B5' are micro but resemble mu. The BookMaster "usec" produces a character of nonstandard width.

## Control Character Representations

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

## Additional ISO-8 Control Character Representations

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

## Formatting Character Representations

| | | | |
|---|---|---|---|
| NSP | Numeric Space | SP | Space |
| RSP | Required Space | SHY | Syllable Hyphen |

## Two-Character BSC Data Link Controls

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

## Commonly Used Editing Pattern Characters

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

## ANSI-Defined Printer Control Characters
(A in **RECFM** field of DCB)

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

## Hexadecimal and Decimal Conversion

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

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

**Note:** 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 | |
| 64 | 16 | 18 446 744 073 709 551 616 | |

File Number: S390-00

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

SA22-7871-00