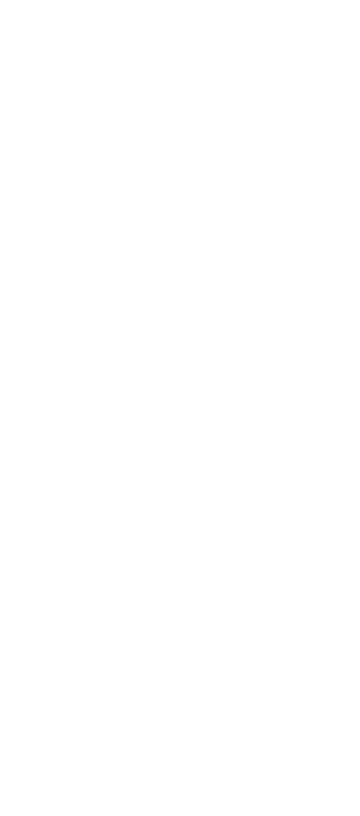


Reference Summary





Reference Summary

Tenth Edition (September, 2017)

This revision differs from the previous edition by containing instructions related to the facilities marked by a bar under "Facility" in "Preface" and minor corrections and clarifications. Changes are indicated by a bar in the margin.

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

Additional copies of this and other IBM publications may be ordered or downloaded from the IBM publications web site at http://www.ibm.com/support/documentation.

Please direct any comments on the contents of this publication to:

IBM Corporation Department E57A 2455 South Road Poughkeepsie, NY 12601-5400

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

© Copyright International Business Machines Corporation 2001-2017. All rights reserved.

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

Preface

This publication is intended primarily for use by z/Architecture™ assembler-language application programmers. It contains basic machine information summarized from the IBM z/Architecture Principles of Operation (SA22-7832), about the IBM z Systems™ 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), The Load-Program-Parameter and the CPU-Measurement Facilities (SC23-2260), and IBM High Level Assembler for z/OS, z/VM & z/VSE Language Reference (SC26-4940). This publication will be updated from time to time. However, the above publications and others cited in this publication are the authoritative reference sources and will be first to reflect changes.

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

Facility	Instruction
ASN-and-LX-reuse	EPAIR, ESAIR, PTI, SSAIR
Compare-and-swap-and-store	CSST
Configuration-topology	PTF
Constrained-transactional-execution	TBEGINC
DAT-enhancement 1	CSPG, IDTE
DAT-enhancement 2	LPTEA

Decimal-floating-point

ADTR, AXTR, CDGTR, CDSTR, CDTR, CDUTR, CEDTR, CEXTR, CGDTR, CGXTR, CSDTR, CSXTR, CUDTR, CUXTR, CXGTR, CXGTR, CXSTR, CXTR, CUDTR, CUXTR, CXGTR, CXSTR, CXTR, CXTR, DTR, EEDTR, EEXTR, ESDTR, ESXTR, FIDTR, FIXTR, IEDTR, IEXTR, KDTR, KXTR, LDETR, LDXTR, LEDTR, LTDTR, LTXTR, LXDTR, MDTR, MXTR, QADTR, QAXTR, RRDTR, RRXTR, SDTR, SLDT, SLXT, SRDT, SRXT, RRDTR, RRXTR, SDTR, SLDT, SLXT, SRDT, SRXT,

RRDTR, RRXTR, SDTR, SLDT, SLXT, SRDT, SRXT, SXTR, TDCDT, TDCET, TDCXT, TDGDT, TDGET, TDGXT SRNMT

DFP-rounding SRNMT

DFP-nacked-conversion CDPT CPDT CF

DFP-packed-conversion CDPT, CPDT, CPXT, CXPT
DFP-zoned-conversion CDZT, CXZT, CZDT, CZXT

Distinct-operands AGHIK, AGRK, AHIK, ALGHSIK, ALGRK, ALHSIK, ALRK,

ARK, NGRK, NRK, OGRK, ORK, SGRK, SLAK, SLGRK, SLLK, SLRK, SRAK, SRK, SRLK, XGRK, XRK

 Enhanced-DAT 1
 PFMF

 Enhanced-DAT 2
 CRDTE

 Execute-extensions
 EXRL

 Execution-hint
 BPP, BPRP, NIAI

Execution-nint BPP, BPHP, NIAI

Expanded-storage PGIN, PGOUT

Extended-immediate AFI, AGFI, ALFI, ALGFI, CFI, CGFI, CLFI, CLGFI,

FLOGR, IİHF, IILF, LBR, LGBR, LGHR, LGFI, LHR, LLC, LLCR, LLGCR, LLGHR, LLH, LLHR, LLIHF, LLILF, LT, LTG, NIHF, NILF, OIHF, OILF, SLFI, SLGFI, XIHF, XILF

CLCLU, MVCLU, PKA, PKU, TP, TROO, TROT, TRTO,

TRTT, UNPKA, UNPKU

Extended-translation 3 CU14, CU24, CU41, CU42, SRSTU, TRTR

Extract-CPU-time ECTG

Extended-translation 2

Floating-point-extension ADTRA, AXTRA, CDFBRA, CDFTR, CDGBRA, CDG-

TRA, CDLFBR, CDLFTR, CDLGBR, CDLGTR, CEFBRA, CEGBRA, CELFBR, CELGBR, CFDBRA, CFDTR, CFEBRA, CFXBRA, CFXBRA, CFXTR, CGDBRA, CGDTRA, CGEBRA, CGXBRA, CGXTRA, CLFDBR, CLFDTR, CLFEBR, CLFXBR, CLFXTR, CLGDBR, CLGDTR, CLGEBR, CLGXBR, CLGXTR, CXFBRA, CXFTR, CXGBRA, CXGTRA, CXLFTR, CXLGBR, CXLGTR, DDTRA, DXTRA, FIDBRA, FIEBRA, FIXBRA, LDXBRA, LEDBRA, LEXBRA, MDTRA, MTRA, SDTRA, SRNMB, SXTRA

Floating-point-support-sign-handling CPSDR, LCDFR, LNDFR, LPDFR

FPR-GR-transfer LDGR, LGDR

Facility Instruction

General-instructions-extension ASI, AGSI, ALSI, ALGSI, CRB, CGRB, CRJ, CGRJ, CRT, CGRT, CGH, CHHSI, CHSI, CGHSI, CHRL, CGHRL, CIB,

CGIB, CIJ, CGIJ, CIT, CGIT, CLRB, CLGRB, CLRJ, CLGRJ, CLRT, CLGRT, CLHHSI, CLFHSI, CLGHSI, CLIB, CLGIB, CLIJ, CLGIJ, CLFIT, CLGIT, CLRL, CLHRL, CLGRL, CLGHRL, CLGFRL, CRL, CGRL, CGFRL, ECAG, LAEY, LTGF, LHRL, LGHRL, LLHRL, LLGHRL LLGFRL, LRL, LGRL, LGFRL, MVHHI, MVHI, MVGHI, MFY, MHY, MSFI, MSGFI, PFD, PFDRL, RNSBG, RXSBG, RISBG, ROSBG, STHRL, STRL, STGRL

Guarded storage LGG, LGSC, LLGFSG, STGSC

MAD, MADR, MAE, MAER, MSD, MSDR, MSE, MSER HFP-multiply-and-add/subtract MAY, MAYR, MAYH, MAYHR, MAYL, MAYLR, MY, MYH, HFP-unnormalized extensions

MYL, MYR, MYHR, MYLR

AHHHR, AHHLR, AIH, ALHHHR, ALHHLR, ALSIH, High-word

ALSIHN, BRCTH, CHF, CHHR, CHLR, CIH, CLHF, CLHHR, CLHLR, CLIH, LBH, LHH, LFH, LLCH, LLHH, RISBHG, RISBLG, SHHHR, SHHLR, SLHHHR, SLHHLR, STCH, STHH, STFH

LEAS SEASE IEEE-exception-simulation

Insert-reference-bits-multiple facility IRRM

LAA, LAAG, LAAL, LAALG, LAN, LANG, LAO, LAOG, Interlocked-access

LAX, LAXG, LPD, LPDG

LAT, LFHAT, LGAT, LLGFAT, LLGTAT Load-and-trap

Load-and-zero-rightmost-byte LLZRGF, LZRF, LZRG

LOC, LOCG, LOCGR, LOCR, STOC, STOCG Load/store-on-condition facility 1

LOCFH, LOCFHR, LOCGHI, LOCHHI, LOCHI, STOCFH Load/store-on-condition facility 2 Long displacement

AHY, ALY, AY, CDSY, CHY, CLIY, CLMY, CLY, CSY, CVBY, CVDY, CY, ICMY, ICY, LAMY, LAY, LB, LDY, LEY, LGB, LHY, LMY, LRAY, LY, MSY, MVIY, NIY, NY, OIY, OY, SHY SLY, STAMY, STCMY, STCY, STDY, STEY, STHY, STMY,

STY, SY, TMY, XIY, XY

Message-security-assist KM, KMC, KIMD, KLMD, KMAC

Message-security-assist extension 3 **PCKMO**

KMCTR, KMF, KMO, PCC Message-security-assist extension 4

PRNO Message-security-assist extension 5 ΚΜΔ Message-security-assist extension 8

Miscellaneous-instruction-extensions 1 CLT CLGT RISBGN

AGH, BIC, MG, MGH, MGRK, MSC, MSGC, MSGRKC, Miscellaneous-instruction-extensions 2

MSRKC, SGH

MVCOS Move-with-optional-specifications TRTE, TRTRE Parsing-enhancement

Perform-floating-point-operation PFPO POPCNT Population-count PPA Processor-assist Reset-reference-bits-multiple RRBM Store-clock fast STCKF Store-facility-list extended STFLE

PTFF Transactional-execution ETND, NTSTG, TABORT, TBEGIN, TEND

TPFI Test-pending-external-interruption

TOD-clock steering

structio

П

Vector-facility-for-z/Architecture LCBB, VA, VAC, VACC, VACC, VAVG, VAVGL, VCDG, VCDLG, VCEQ, VCGD, VCH, VCHL, VCKSM, VCLGD,

VCLZ, VCTZ, VEC, VECI, VERIM, VERILL, VERILV, VESIL, VESILV, VESARA, VESRAV, VESRIL, VERILV, VFA, VFAE, VFGE, VFCH, VFCHE, VFD, VFEE, VFENE, VFI, VFLL, VFLR, VFM, VFMS, VFPSO, VFS, VFSO, VFTCI, VGBM, VGFMA, VGM, VGFMA, VGM, VISTR, VL, VLBB, VLCF, VGEG, VGFM, VGFMA, VGM, VISTR, VL, VLBG, VLCH, VLEF, VLEG, VLCH, VLLEZ, VLM, VLP, VLR, VLREP, VLVG, VLVGP, VMAE, VMAH, VMAL, VMALD, VMAO, VME, VMH, VML, VMM, VMML, VMO, VMRH, VMM, VMXL, VN, VNC, VNO, VO, VPDI, VPERM, VPK, VPKLS, VPSO, VPOPCT, VREP, VREPI, VS, VSBCBI, VSBI, VSCBI, VSCB, VSTEP, VSTEP, VSTEP, VSTEP, VSTEP, VSTEP, VSTEP, VSTEM, VSTM, VSTMC, VSUMG, VSTM, VUPL, VUPLH, VUPL, VUPLH,

VUPLL, VX, WFC, WFK

Vector-extensions facility 1 VBPERM, VFMAX, VFMMA, VFNMS, VMSL,

VNN, VNX, VOC

Vector-packed-decimal VAP, VCP, VCVB, VCVBG, VCVD, VCVDG, VDP, VLIP,
VMP, VMSP, VPKZ, VPSOP, VRP, VSDP, VSPP, VSP, VTP.

VUPKZ, VLRLR, VLRL, VSTRLR, VSTRL

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

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

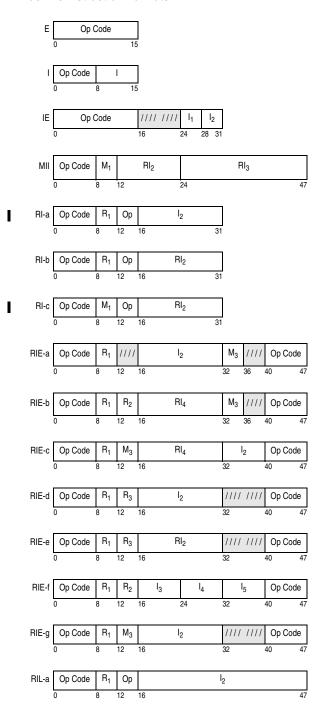
Contents

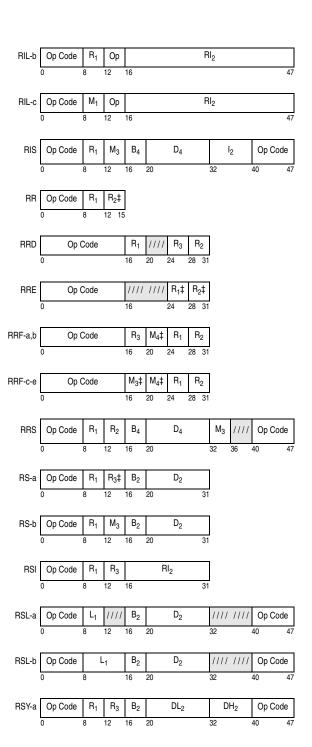
Pretace
Contents vi
Machine Instruction Formats
Machine Instructions by Mnemonic
Machine Instructions by Operation Code
Condition Codes
Assembler Instructions
CNOP Alignment
Extended-Mnemonic Instructions for Branch on Condition and
Branch Indirect on Condition
Extended-Mnemonic Instructions for Relative-Branch Instructions 41
Extended-Mnemonic Suffixes for Compare-and-Branch, and
Compare-and-Trap Instructions
Extended-Mnemonic Suffixes for Load/Store-on-Condition
Instructions
Extended-Mnemonic Suffixes for Rotate-Then-Insert / AND / OR /
Exclusive OR-Selected-Bits Instructions
Extended-Mnemonics for Vector-Facility Instructions 43
Summary of Constants
Assigned Storage Locations
External-Interruption Codes45
Program-Interruption Codes
Data-Exception Code (DXC)
Vector-Exception Code (VXC)47
PER Code, ATMID, and Al47
Translation-Exception Identification47
Machine-Check Interruption Code
External-Damage Code
Facility Indications
Control Registers51
Floating-Point-Control (FPC) Register 53
Program-Status Word (PSW)
z/Architecture PSW
Short-Format PSW54
Dynamic Address Translation
Virtual-Address Format
Address-Space-Control Element (ASCE)
Region-Table or Segment-Table Designation (RTD or STD) 55
Real-Space Designation (RSD)
Table Values
Region-Table Entry (RTE)
Segment-Table Entry (STE, FC=0)
Segment-Table Entry (STE, FC=1)
Page-Table Entry (PTE)
ASN Translation
Address-Space Number (ASN)
ASN-First-Table Entry
ASN-Second-Table Entry (ASTE)
PC-Number Translation
Program-Call Number (20-Bit) 58 Program-Call Number (32-Bit, Bit 44=0) 58
Program-Call Number (32-Bit, Bit 44=0)
riogram-call number (32-bit, Bit 44=1)

Linkage-Table Entry (LTE)	
Linkage-First-Table Entry (LFTE)	
Linkage-Second-Table Entry (LSTE)	59
Entry-Table Entry (ETE)	59
Access-Register Translation	60
Access-List-Entry Token (ALET)	
Dispatchable-Unit-Control Table (DUCT)	
Access-List Entry (ALE)	
Linkage-Stack Entries	61
Entry Descriptor	
Header Entry (Entry Type 0001001)	
Trailer Entry (Entry Type 0001010)	62
Branch State Entry (Entry Type 0001100) and	
Program-Call State Entry (Entry Type 0001101)	
Trapping	
Trap Control Block	
Trap Save Area	
Trace-Entry Formats	
Identification of Trace Entries	
Branch	
Branch in Subspace Group (if ASN Tracing on)	66
Mode Switch	
Mode-Switching Branch	
Program Call	
Program Return	
Program Transfer	
Set Secondary ASN	70
Trace	
Operand of Store Clock and Store Clock Fast	71
Operand of Store Clock Extended	
Transaction Diagnostic Block (TDB)	
Guarded-Storage Facility Registers and Parameters	
Guarded-Storage-Designation (GSD) Register	73
Guarded-Storage Control Block	
Guarded-Storage-Event Parameter List	73
Operation-Request Block (ORB)	
Command-Mode ORB	
Transport-Mode ORB	
Channel-Command Word (CCW)	75
Format-0 CCW	75
Format-1 CCW	
Indirect-Data-Address Word (IDAW)	75
Format-1 IDAW	
Format-2 IDAW	
Modified-CCW-Indirect-Data-Address Word (MIDAW)	
Transport Control Word (TCW)	
Transport-Indirect-Data-Address Word (TIDAW)	77
Transport Command Control Block (TCCB)	
Transport Command Area Header (TCAH)	
Device-Command Word (DCW)	78
Transport Command Area Trailer (TCAT)	
CBC-Offset Block (COB)	
Transport Status Block (TSB)	
Transport Status Header (TSH)	80

I/O-Status TSA 80
Device-Detected-Program-Check TSA
Interrogate TSA82
Subchannel-Information Block (SCHIB)
Path-Management-Control Word (PMCW) 83
Interruption-Response Block (IRB)
Command-Mode Subchannel-Status Word (SCSW)84
Transport-Mode Subchannel-Status Word (SCSW) 85
Extended-Status Word (ESW)86
Format-0 ESW
Format-0 ESW Word 0 (Subchannel Logout) 87
Format-0 ESW Word 1 (Extended-Report Word) 87
Format-1 ESW Word 0
Format-2 ESW Word 0 ¹
Information Stored in ESW
Extended-Control Word (ECW)
Extended-Measurement Word
Format 0 Measurement Block
Format 1 Measurement Block
Channel-Report Word (CRW)
Error-Recovery Codes
Reporting Source91
I/O Command Codes
Standard Command-Code Assignments
(CCW and DCW Bits 0-7)91
Standard Meanings of Bits of First Sense Byte
Hexadecimal and Decimal Conversion
Powers of 2 and 16
Character Assignments96
Control Character Representations
Additional ISO-8 Control Character Representations 98
Formatting Character Representations
Two-Character BSC Data Link Controls
Commonly Used Editing Pattern Characters
ANSI-Defined Printer Control Characters

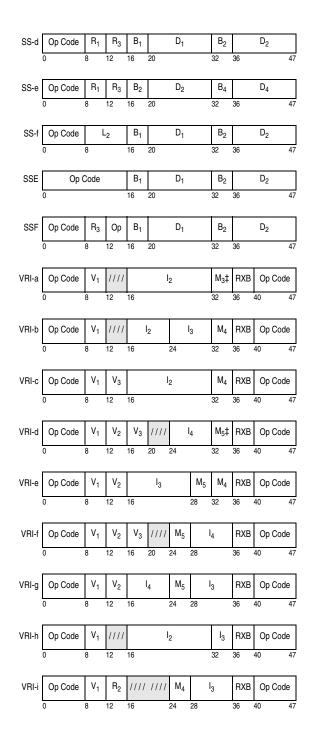
Machine Instruction Formats





RSY-b	Op Code	R ₁	M ₃	B ₂		DL ₂	32	H ₂	Op Co	de
	0	8	12	16	20		32		40	47
RX-a	Op Code	R ₁	X_2	B ₂		D_2				
	0	8	12	16	20		31			
					,					
RX-b	Op Code	M ₁	X ₂	B ₂		D_2				
	0	8	12	16	20		31			
		1	1		1		-		ı	
RXE	Op Code	R ₁	X ₂	B ₂		D_2	M ₃ ‡	////	Op Co	de
	0	8	12	16	20		32	36	40	47
RXF	Op Code	R_3	X ₂	B ₂		D ₂	R ₁	////	Op Co	de
	0	8	12	16	20		32	36	40	47
5).0.4			Lv	_	1	DI	1 -			. 1
нх ү-а	Op Code	Н1	λ ₂	В2		DL ₂		т2	Op Co	de
	0	8	12	16	20		32		40	47
DVVh	On Codo	М.	Υ.	R.	1	DI.	Г	H.	On Co	do
חאוים	Op Code	0 1111	10	16	20	DL ₂	20	'1'2	Op Co	47
	U	0	12	10	20		32		40	47
s	Op	Code		B ₂ ‡	I	Do‡				
	Op 0			16	20	21	31			
SI	Op Code	ŀ	2‡	B ₁		D ₁				
	0	8		16	20		31			
SIL	Op 0	Code		B ₁		D ₁			l ₂	
	0			16	20		32			47
									1	
SIY	Op Code		l ₂	B ₁		DL ₁		H ₁	Op Co	de
	0	8		16	20		32		40	47
		ı			1					_
SMI	Op Code	M ₁	////	B ₃		D_3		F	RI ₂	
	0	8	12	16	20		32			47
	Г						-1-			
SS-a	Op Code	Lo	or L ₁	В ₁		D ₁	B ₂		D ₂	
	0	8	_	16	20		32	36		47
00 '	0-0-1	1.	1 ,	В	1		I P	1		
SS-b	Op Code	L1	L ₂	В1		D ₁	B ₂		D ₂	
	U	8	12	16	20		32	36		47
99.	On Codo	1.	l.	R.		D ₁	R.		D ₂	
35-C	Op Code	<u> </u>	13	16	20	۲1	22	26	D ₂	17
	U	O	12	10	20		32	30		4/

I



VRR-a	Op Code	V ₁	V ₂	////	////	M ₅ ‡	M ₄ ‡	M ₃ ‡	RXB	Ор	Code
	0	8	12	16		24	28	32	36	40	47
VRR-b	Op Code	V ₁	V ₂	V ₃	////	M ₅ ‡	////	M ₄ ‡	RXB	Ор	Code
	0	8	12	16	20	24	28	32	36	40	47
VRR-c	Op Code	V ₁	V ₂	V ₃	////	M ₆ ‡	M ₅ ‡	M ₄ ‡	RXB	Ор	Code
	0	8	12	16	20	24	28	32	36	40	47
VRR-d	Op Code	V ₁	V ₂	V ₃	M ₅ ‡	M ₆ ‡	////	V ₄	RXB	Ор	Code
VRR-e	Op Code	V ₁	V ₂	V ₃	M ₆ ‡	////	M ₅ ‡	V ₄	RXB	Ор	Code
VRR-f	Op Code	V ₁	R ₂	R ₃	20	////	////	////	RXB 36	Op (Code 47
VRR-g	Op Code	8	V ₁	16	////	////	////	////	RXB	Op (Code
•	•	•							00		
		1	l		i	l			1		
VRR-h	Op Code	////	V ₁	V ₂	////	M ₃	////	////	RXB	Op (Code 47
	Op Code										
	Op Code Op Code 0										
VRR-i	Op Code	R ₁	V ₂	1///	////	M ₃	//// 28	////	RXB 36	Op (Code 47
VRR-i		R ₁	V ₂	//// 16 B ₂	////	M ₃	28	////	RXB 36 RXB	Op (Code 47
VRR-i VRS-a	Op Code Op Code	8 V ₁ 8	V ₂ 12 V ₃ 12	//// 16 B ₂	20	M ₃	28	//// M ₄ ‡	RXB 36 RXB 36	Op (Code 47 Code 47
VRR-i VRS-a VRS-b	Op Code	8 V ₁ 8	V ₂ 12 V ₃ 12	16 B ₂ 16 B ₂	20	M ₃	28	//// M ₄ ‡	RXB 36 RXB 36	Op (Code 47 Code 47
VRR-i VRS-a VRS-b	Op Code Op Code Op Code Op Code	8 V ₁ 8	V ₂ 12 V ₃ 12 R ₃	16 B ₂ 16 B ₂ 16	20 20	M ₃ 24 D ₂	28	//// M ₄ ‡ 32 M ₄ ‡ 32	RXB 36 RXB 36 RXB 36	Op (40 Op (40 Op (40) 40	Code 47 Code 47
VRR-i VRS-a VRS-b	Op Code Op Code Op Code Op Code	8 V ₁ 8	V ₂ 12 V ₃ 12 R ₃ 12 V ₃	16 B ₂ 16 B ₂ 16	20 20	M ₃	228	//// M ₄ ‡ 32 M ₄ ‡ 32	RXB 36 RXB 36 RXB RXB	Op (Op (Op (40) Op (40) Op (Op	Code 47 Code 47 Code
VRR-i VRS-a VRS-b VRS-c	Op Code Op Code Op Code Op Code	R ₁ 8 V ₁ 8 R ₁ 8	V ₂ 12 V ₃ 12 R ₃ 12 V ₃ 12	16 B ₂ 16 B ₂ 16	20 20	M ₃ 24 D ₂	228	M ₄ ‡ 32 M ₄ ‡ 32 M ₄ ‡	RXB 36 RXB 36 RXB RXB	Op (Op (40 Op (40 Op (40 Op (40)	Code 47 Code 47 Code 47 Code 47
VRS-a VRS-b VRS-c	Op Code Op Code Op Code Op Code Op Code Op Code	R ₁ 8 V ₁ 8 R ₁ 8	V ₂ 12 V ₃ 12 R ₃ 12 V ₃ 12	//// 16 B ₂ B ₂	20 20	M ₃ 224 D ₂ D ₂	228	M ₄ ‡ 32 M ₄ ‡ 32 M ₄ ‡ 32	RXB 36 RXB 36 RXB 36 RXB RXB	Op (Op (40 Op (40 Op (40 Op (40)	Code 47 Code 47 Code 47 Code 47
VRS-a VRS-b VRS-c	Op Code Op Code Op Code Op Code Op Code Op Code	R ₁ 8 V ₁ 8 V ₁ 8 V ₁ 8	V ₂ 112 V ₃ 112 R ₃ 112 R ₃ 112	//// 16 B ₂ B ₂	220	M ₃ 224 D ₂ D ₂	228	M ₄ ‡ 32 M ₄ ‡ 32 M ₄ ‡ 32 V ₁ 32	RXB 36 RXB 36 RXB 36 RXB RXB	Op (40	Code 47 Code 47 Code 47 Code 47 Code 47 47

VRX	Op Code	V ₁	X ₂	B ₂	D ₂	M ₃ ‡	RXB	Op Code
	0	8	12	16	20	32	36	40 47

VSI	Op Code	l ₃	B ₂	D ₂	V ₁	RXB	Op Code	
	0	8	16	20	32	36	40	47

1, 2, 3, 4, 5, 6 Denotes association with first, second, third, fourth, fifth, or sixth operand

a, b, c, d, e, f Distinguishes among instances of the same basic instruction format

B₁, B₂, B₃, B₄ Base register designation field

D₁, D₂, D₃, D₄ Displacement field (including DH and DL for long-displacement

forms)

 $\mathsf{I},\,\mathsf{I}_2,\,\mathsf{I}_3,\,\mathsf{I}_4,\,\mathsf{I}_5 \qquad \qquad \mathsf{Immediate operand field}$

 L, L_1, L_2 Length field M_1, M_3, M_4, M_5, M_6 Mask field

R₁, R₂, R₃ Register designation field
RI₂, RI₃, RI₄ Relative-immediate operand field

RXB Most significant bits of vector registers designated by the $V_1, V_2,$

V₃, V₄ fields, respectively Index register designation field

X₂ Index register designation field ‡ For certain instructions, this operand is not defined

Machine Instructions by Mnemonic

Mne- monic	Onorande	Name	For- mat	Op- code	Clas & Note
A	Operands	Add (32)	RX-a	5A	C
AD	R ₁ ,D ₂ (X ₂ ,B ₂)	Add Normalized (LH)	RX-a	6A	пc
ADB	$R_1, D_2(X_2, B_2)$	Add (LB)	RXE	ED1A	
ADBR	R ₁ ,D ₂ (X ₂ ,B ₂)	Add (LB)	RRE	B31A	
ADBR	R ₁ ,R ₂	Add Normalized (LH)	RR	2A	пc
ADTR	R ₁ ,R ₂	, ,		B3D2	
ADTRA	R ₁ ,R ₂ ,R ₃	Add (LD) Add (LD)		B3D2	
AD I NA AE	R ₁ ,R ₂ ,R ₃ ,M ₄	Add Normalized (SH)	RX-a		D C
AEB	$R_1, D_2(X_2, B_2)$	Add (SB)	RXE	ED0A	
AEBR	$R_1, D_2(X_2, B_2)$	• •	RRE	B30A	
	R ₁ ,R ₂	Add (SB)	RR	3A	
AER	R ₁ ,R ₂	Add Normalized (SH)	RIL-a		¤ C
AFI AG	R ₁ ,l ₂	Add Immediate (32)			c El
	$R_1, D_2(X_2, B_2)$	Add (64)		E308	
AGF	$R_1, D_2(X_2, B_2)$	Add (64←32)		E318	
AGFI	R ₁ ,l ₂	Add Immediate (64←32)	RIL-a		c El
AGFR	R ₁ ,R ₂	Add (64←32)	RRE	B918	
AGH	$R_1,D_2(X_2,B_2)$	Add Halfword (64←16)		E338	
AGHI	R ₁ ,l ₂	Add Halfword Immediate (64←16)	RI-a	A7B	c N
AGHIK	R ₁ ,R ₃ ,I ₂	Add Immediate (64←16)		ECD9	
AGR	R ₁ ,R ₂	Add (64)	RRE	B908	
AGRK	R ₁ ,R ₂ ,R ₃	Add (64)		B9E8	
AGSI	$D_1(B_1), I_2$	Add Immediate (64←8)	SIY	EB7A	
AH	$R_1, D_2(X_2, B_2)$	Add Halfword (32←16)	RX-a	4A	С
AHHHR	R_1,R_2,R_3	Add High (32)		B9C8	
AHHLR	R_1,R_2,R_3	Add High (32)		B9D8	c HW
AHI	R_1,I_2	Add Halfword Immediate (32←16)	RI-a	A7A	С
AHIK	R_1,R_3,I_2	Add Immediate (32←16)		ECD8	
AHY	$R_1, D_2(X_2, B_2)$	Add Halfword (32←16)		E37A	c LD
AIH	R_1,I_2	Add Immediate High (32)	RIL-a		c HW
AL	$R_1,D_2(X_2,B_2)$	Add Logical (32)	RX-a	5E	С
ALC	$R_1,D_2(X_2,B_2)$	Add Logical with Carry (32)	RXY-a	E398	c N3
ALCG	$R_1,D_2(X_2,B_2)$	Add Logical with Carry (64)	RXY-a	E388	сN
ALCGR	R ₁ ,R ₂	Add Logical with Carry (64)	RRE	B988	сN
ALCR	R ₁ ,R ₂	Add Logical with Carry (32)	RRE	B998	c N3
ALFI	R_1,I_2	Add Logical Immediate (32)	RIL-a	C2B	c El
ALG	$R_1,D_2(X_2,B_2)$	Add Logical (64)	RXY-a	E30A	сN
ALGF	$R_1,D_2(X_2,B_2)$	Add Logical (64←32)	RXY-a	E31A	сN
ALGFI	R_1,I_2	Add Logical Immediate (64←32)	RIL-a	C2A	c El
ALGFR	R ₁ ,R ₂	Add Logical (64←32)	RRE	B91A	сN
ALGHSIK	R_1,R_3,I_2	Add Logical with Signed Immediate (64←16)	RIE-d	ECDB	c DO
ALGR	R ₁ ,R ₂	Add Logical (64)	RRE	B90A	сN
ALGRK	R ₁ ,R ₂ ,R ₃	Add Logical (64)	RRF-a	B9EA	c DO
ALGSI	$D_1(B_1), I_2$	Add Logical with Signed Immediate (64←8)	SIY	EB7E	c GE
ALHHHR		Add Logical High (32)		B9CA	c HV
ALHHLR		Add Logical High (32)	RRF-a	B9DA	c HV
ALHSIK	R ₁ ,R ₃ ,I ₂	Add Logical with Signed Immediate (32←16)	RIE-d	ECDA	c DC
ALR	R ₁ ,R ₂	Add Logical (32)	RR	1E	С
ALRK	R ₁ ,R ₂ ,R ₃	Add Logical (32)	RRF-a	B9FA	c DO
ALSI	D ₁ (B ₁),l ₂	Add Logical with Signed Immediate (32←8)		EB6E	
ALSIH	R ₁ ,I ₂	Add Logical with Signed Immediate High (32)	RIL-a	CCA	c HW
ALSIHN	R_1,I_2	Add Logical with Signed Immediate High (32)	RIL-a	ССВ	HW
ALY	$R_1,D_2(X_2,B_2)$	Add Logical (32)	RXY-a	E35E	c LD
AP	$D_1(L_1,B_1),D_2(L_2,B_2)$		SS-b	FA	пc
AR	R ₁ ,R ₂	Add (32)	RR	1A	С
ARK	R ₁ ,R ₂ ,R ₃	Add (32)		B9F8	c DO

Mne-			For-	Ор-	Clas &
monic	Operands	Name	mat	code	Note
ASI	$D_1(B_1),I_2$	Add Immediate (32←8)	SIY	EB6A	c GE
AU	$R_1,D_2(X_2,B_2)$	Add Unnormalized (SH)	RX-a	7E	¤С
AUR	R ₁ ,R ₂	Add Unnormalized (SH)	RR	3E	¤ C
AW	$R_1,D_2(X_2,B_2)$	Add Unnormalized (LH)	RX-a	6E	Ω C
AWR	R ₁ ,R ₂	Add Unnormalized (LH)	RR	2E	ΩC
AXBR	R ₁ ,R ₂	Add (EB)	RRE	B34A	
AXR	R ₁ ,R ₂	Add Normalized (EH)	RR	36	¤ C
AXTR	R ₁ ,R ₂ ,R ₃	Add (ED)	RRF-a		
AXTRA	R ₁ ,R ₂ ,R ₃ ,M ₄	ADD (ED)	RRF-a		
AY	R ₁ ,D ₂ (X ₂ ,B ₂)	Add (32)	RXY-a		
BAKR	R ₁ ,R ₂	Branch and Stack	RRE	B240	a a
BAL	$R_1,D_2(X_2,B_2)$	Branch and Link	RX-a	45	Ω D
BALR	R ₁ ,R ₂	Branch and Link	RR	05 4D	
BAS	$R_1,D_2(X_2,B_2)$	Branch and Save	RX-a	4D	0
BASR	R ₁ ,R ₂	Branch and Save	RR	0D	0
BASSM	R ₁ ,R ₂	Branch and Save and Set Mode	RR	0C	0
BC BCB	M ₁ ,D ₂ (X ₂ ,B ₂)	Branch on Condition	RX-b	47 07	a a
BCR BCT	M ₁ ,R ₂	Branch on Count (32)	RR RX-a	07 46	Ω Ω
	$R_1,D_2(X_2,B_2)$	Branch on Count (32)			
BCTG	R ₁ ,D ₂ (X ₂ ,B ₂)	Branch on Count (64)	RXY-a RRE		пN
BCTGR BCTR	R ₁ ,R ₂	Branch on Count (64)	RR	B946	
BIC	R ₁ ,R ₂	Branch on Count (32)	RXY-b	06 E247	D n M
BPP	M ₁ ,D ₂ (X ₂ ,B ₂)	Branch Indirect on Condition Branch Prediction Preload	SMI	C7	¤ MI
	M ₁ ,Rl ₂ ,D ₃ (B ₃)				
BPRP	M ₁ ,Rl ₂ ,Rl ₃	Branch Prediction Relative Preload	MII	C5	¤ EH
BRAS	R ₁ ,Rl ₂	Branch Relative and Save	RI-b RIL-b	A75	¤ N3
BRASL BRC	R ₁ ,RI ₂	Branch Relative and Save Long Branch Relative on Condition	RI-c	A74	n M3
BRCL	M ₁ ,Rl ₂ M ₁ ,Rl ₂	Branch Relative on Condition Long	RIL-c	C04	¤ N3
BRCT		Branch Relative on Count (32)	RI-b	A76	α α
BRCTG	R ₁ ,Rl ₂ R ₁ ,Rl ₂	Branch Relative on Count (64)	RI-b	A77	¤Ν
BRCTH	R ₁ ,RI ₂	Branch Relative on Count High (32)	RIL-b	CC6	¤ HV
BRXH	R ₁ ,R ₃ ,Rl ₂	Branch Relative on Index High (32)	RSI	84	מ
BRXHG	R ₁ ,R ₃ ,Rl ₂	Branch Relative on Index High (64)	RIE-e		
BRXLE	R ₁ ,R ₃ ,Rl ₂	Branch Relative on Index Low or Equal (32)		85	D IN
BRXLG	R ₁ ,R ₃ ,Rl ₂	Branch Relative on Index Low or Equal (64)			
BSA	R ₁ ,R ₂	Branch and Set Authority	RRE	B25A	q
BSG	R ₁ ,R ₂	Branch in Subspace Group	RRE	B258	ч D
BSM	R ₁ ,R ₂	Branch and Set Mode	RR	0B	D D
BXH	R ₁ ,R ₃ ,D ₂ (B ₂)	Branch on Index High (32)	RS-a	86	¤
BXHG	R ₁ ,R ₃ ,D ₂ (B ₂)	Branch on Index High (64)	RSY-a		
BXLE	$R_1, R_3, D_2(B_2)$ $R_1, R_3, D_2(B_2)$	Branch on Index Low or Equal (32)	RS-a	87	מ
BXLEG	$R_1, R_3, D_2(B_2)$ $R_1, R_3, D_2(B_2)$	Branch on Index Low or Equal (64)	RSY-a		
C	$R_1, D_2(X_2, B_2)$	Compare (32)	RX-a	59	C
CD	$R_1,D_2(X_2,B_2),M_3$	Compare (LH)	RX-a	69	С
CDB	$R_1,D_2(X_2,B_2)$	Compare (LB)	RXE	ED19	
CDBR	R ₁ ,R ₂	Compare (LB)	RRE	B319	пc
CDFBR	R ₁ ,R ₂	Convert from Fixed (LB←32)	RRE	B395	D C
	R ₁ ,M ₃ ,R ₂ ,M ₄	Convert from Fixed (LB←32)	RRF-e		¤Ε
CDFR	R ₁ ,R ₂	Convert from Fixed (LH←32)	RRE	B3B5	
CDFTR	R ₁ ,M ₃ ,R ₂ ,M ₄	Convert from Fixed (LD←32)	RRE	B951	
CDGBR	R ₁ ,R ₂	Convert from Fixed (LB←64)	RRE	B3A5	
	R ₁ ,M ₃ ,R ₂ ,M ₄	Convert from Fixed (LB←64)	RRF-e		
CDGR	R ₁ ,R ₂	Convert from Fixed (LH←64)	RRE	B3C5	
CDGTR	R ₁ ,R ₂	Convert from Fixed (LD←64)	RRE	B3F1	
	R ₁ ,M ₃ ,R ₂ ,M ₄	Convert from Fixed (LD ← 64)	RRF-e		
	R ₁ ,M ₃ ,R ₂ ,M ₄	Convert from Logical (LB←32)	RRF-e		¤F
	R ₁ ,W ₃ ,R ₂ ,W ₄	Convert from Logical (LD←32)	RRF-e		
	R ₁ ,M ₃ ,R ₂ ,M ₄ R ₁ ,M ₃ ,R ₂ ,M ₄	Convert from Logical (LB←64)		B3A1	
	R ₁ ,W ₃ ,R ₂ ,W ₄	Convert from Logical (LD←64)	RRF-e		
OPPOIN	1 17,1413,1 12,1414	Convert Horri Logical (LD v 04)	e	عرون	~ 1

CDR R₁-R₂ Compare (LH) RR CDS R₁-R₃,D₂(B₂) Compare Double and Swap (32) RS-a CDSG R₁-R₃,D₂(B₂) Compare Double and Swap (64) RSY-i CDSTR R₁-R₂ Convert from Signed Packed (LD←64) RSY-i CDSY R₁-R₃,D₂(B₂) Compare Double and Swap (32) RSY-i CDTR R₁-R₂ Compare Double and Swap (32) RSY-i CDTR R₁-R₂ Compare (LD) RRE CDUTR R₁-R₂ Convert from Unsigned Packed (LD←64) RRE CDZT R₁-D₂(L₂-B₂),M₃ Convert from Daved (to long DFP) RSL-i CE R₁-D₂(L₂-B₂) Compare (SH) RX-a CEB R₁-D₂(L₂-B₂) Compare (SH) RX-a CEB R₁-D₂(X₂-B₂) Compare (SH) RX-a CEB R₁-D₂(X₂-B₂) Compare (SH) RX-a CEB R₁-R₂ Compare (SH) RRE CEDTR R₁-R₂ Compare Biased Exponent (LD) RRE CEFBR R₁-R₂ Convert from Fixed (SH←32)	B3F2 D EDAA 79 ED09 B309 B3F4 B394 B384 B384 B3A4 B3A4 B3C4 B390 B3FC B21A	0 C 0 C C C C C C C C C C C C C C C C C
CDPT R ₁ ,D ₂ (L ₂ ,B ₂),M ₃ Convert from Packed (To Long DFP) RSL-I CDR R ₁ ,R ₃ ,D ₂ (B ₂) Compare (LH) RR CDS R ₁ ,R ₃ ,D ₂ (B ₂) Compare Double and Swap (32) RS-a CDSG R ₁ ,R ₃ ,D ₂ (B ₂) Compare Double and Swap (64) RSY-G CDSTR R ₁ ,R ₂ Convert from Signed Packed (LD-64) RRE CDSY R ₁ ,R ₃ ,D ₂ (B ₂) Compare Double and Swap (32) RSY-G CDTR R ₁ ,R ₂ Compare Double and Swap (32) RSY-G CDTR R ₁ ,R ₂ Compare Double and Swap (32) RSY-G CDTR R ₁ ,R ₂ Compare Double and Swap (32) RSY-G CDTR R ₁ ,R ₂ Compare Double and Swap (32) RSY-G CDTR R ₁ ,R ₂ Compare Glath RSY-G CDTR R ₁ ,R ₂ Compare (LD) RRE CDTR R ₁ ,R ₂ Convert from Unsigned Packed (LD-64) RRE CEB R ₁ ,D ₂ (L ₂ ,B ₂),M ₃ Convert (SB) RXE CEB R ₁ ,D ₂ (L ₂ ,B ₂),M ₃ Compare (SB)	DEDAE 29 BB 1 EB3E B3F3 1 B3E4 B3F2 DEDAE 79 ED09 B309 B3F4 B394 PC CC CC CC CC CC CC CC CC CC	
$ \begin{array}{llllllllllllllllllllllllllllllllllll$	29 BB BBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB	0 C 0 C C C C C C C C C C C C C C C C C
$ \begin{array}{llllllllllllllllllllllllllllllllllll$	BB	OCNOTE OCN
$\begin{array}{llll} \text{CDSG} & R_1, R_3, D_2(D_2) & \text{Compare Double and Swap } (64) & \text{RSY-CDSTR} \\ R_1, R_2 & \text{Convert from Signed Packed } (LD \leftarrow 64) & \text{RRE} \\ \text{CDSY} & R_1, R_3, D_2(B_2) & \text{Compare Double and Swap } (32) & \text{RSY-CDTR} \\ R_1, R_2 & \text{Compare } (LD) & \text{RRE} \\ \text{CDUTR} & R_1, R_2 & \text{Convert from Unsigned Packed } (LD \leftarrow 64) & \text{RRE} \\ \text{CDUTR} & R_1, D_2(L_2, B_2), M_3 & \text{Convert from Zoned (to long DFP)} & \text{RSL-IDSTR} \\ \text{CE} & R_1, D_2(X_2, B_2) & \text{Compare } (SH) & \text{RX-RE} \\ \text{CEBR} & R_1, R_2 & \text{Compare } (SH) & \text{RX-RE} \\ \text{CEDTR} & R_1, R_2 & \text{Compare } (SH) & \text{RRE} \\ \text{CEDTR} & R_1, R_2 & \text{Compare } (SH) & \text{RRE} \\ \text{CEDTR} & R_1, R_2 & \text{Compare } (SH) & \text{RRE} \\ \text{CEFBR} & R_1, R_2 & \text{Compare } (SH) & \text{RRE} \\ \text{CEFBR} & R_1, R_2 & \text{Convert from Fixed } (SH \leftarrow 32) & \text{RRE} \\ \text{CEFBRA} & R_1, M_3, R_2, M_4 & \text{Convert from Fixed } (SH \leftarrow 32) & \text{RRE} \\ \text{CEGBR} & R_1, R_2 & \text{Convert from Fixed } (SH \leftarrow 32) & \text{RRE} \\ \text{CEGBRA} & R_1, M_3, R_2, M_4 & \text{Convert from Fixed } (SH \leftarrow 64) & \text{RRE} \\ \text{CEGBRA} & R_1, R_2 & \text{Convert from Fixed } (SH \leftarrow 64) & \text{RRE} \\ \text{CEGBRA} & R_1, M_3, R_2, M_4 & \text{Convert from Logical } (SH \leftarrow 64) & \text{RRE} \\ \text{CELGBR} & R_1, R_2 & \text{Convert from Logical } (SH \leftarrow 64) & \text{RRE} \\ \text{CELGBR} & R_1, R_2 & \text{Convert from Logical } (SH \leftarrow 64) & \text{RRE} \\ \text{CELGBR} & R_1, R_2 & \text{Convert from Logical } (SH \leftarrow 64) & \text{RRE} \\ \text{CELGBR} & R_1, R_2 & \text{Convert from Logical } (SH \leftarrow 64) & \text{RRE} \\ \text{CELGBR} & R_1, R_2 & \text{Convert from Logical } (SH \leftarrow 64) & \text{RRE} \\ \text{CECTATR} & R_1, R_2 & \text{Compare } SH & \text{Convert from Logical } (SH \leftarrow 64) & \text{RRE} \\ \text{CFDR} & R_1, R_2 & \text{Compare } SH & \text{Convert } SH \\ \text{CFDR} & R_1, R_2 & \text{Compare } SH & \text{Convert } SH \\ \text{CFDR} & R_1, R_2 & \text{Compare } SH & \text{Convert } SH \\ \text{CFDBR} & R_1, M_3, R_2 & \text{Convert } SH & \text{Convert } SH \\ \text{CFDBR} & R_1, R_2 & \text{Compare } SH & \text{Convert } SH \\ \text{CFDBR} & R_1, R_3, R_2 & \text{Convert } SH & \text{CONVERT } SH \\ \text{CFDBR} & R_1, R_3, R_2 & \text{Convert } SH & \text{CONVERT } SH \\ \text{CFDBR} & R_1, R_3, R_2 & $	B3F3 B3F4 B3F4 B3F4 B3F4 B3F4 B3F4 B3F4	a c N a TF a c LD a c TF a ZF a c a c a c a c a c a c a F a N a F a F a N a F a F a C a C a C a C TF a C TF
CDSTR R_1,R_2 Convert from Signed Packed (LD←64) RRE CDSY $R_1,R_3,D_2(B_2)$ Compare Double and Swap (32) RSY-CDTR R_1,R_2 Compare (LD) RRE CDUTR R_1,R_2 Convert from Unsigned Packed (LD←64) RRE CDZT $R_1,D_2(L_2,B_2)$ M3 Convert from Zoned (to long DFP) RSL-CE $R_1,D_2(X_2,B_2)$ Compare (SH) RX-a CDZT $R_1,D_2(L_2,B_2)$ Compare (SH) RX-a CDZT $R_1,D_2(X_2,B_2)$ Compare (SB) RXE CEBR R_1,R_2 Compare (SB) RXE CEBR R_1,R_2 Compare (SB) RRE CEDTR R_1,R_2 Compare (SB) RRE CEFBR R_1,R_2 Convert from Fixed (SB←32) RRE CEFBR R_1,R_2 Convert from Fixed (SH←32) RRE CEGBR R_1,R_2 Convert from Fixed (SH←64) RRE CEGBR R_1,R_2 Convert from Fixed (SH←64) RRE CELFBR R_1,R_2 Convert from Fixed (SH←64) RRE CELFBR R_1,R_2 Convert from Logical (SH←64) RRE CEXTR R_1,R_2 Compare SIAD RAPE CEXTR R_1,R_2 Compare Biased Exponent (ED) RRE CEXTR R_1,R_2 Compare Biased Exponent (ED) RRE CFDBR R_1,M_3,R_2 Convert to Fixed (32←LB) RFF-CFDBRA R_1,M_3,R_2,M_4 Convert to Fixed (32←LB) RRF-CFDBRA R_1,M_3,R_2,M_4 Convert to Fixed (32←LB) RRF-CFDBRA R_1,M_3,R_2,M_4 Convert to Fixed (32←LB) RRF-	B3F3 B3E4 B3F2 D EDAA 79 ED09 B309 B3F4 B394 B394 B394 B384 B384 B384 B384 B384 B384 B384 B38	o TF o c LD o c TF o TF o ZF o c o c o c o c TF o o N o F o F o c o c TF
$ \begin{array}{llllllllllllllllllllllllllllllllllll$	B314 B352 B364 B364 B364 B364 B364 B364 B364 B364	a c LD a c TF a TF a ZF a c a c a c a c a c TF a B B B B B B B B B B B B B B B B B B
$ \begin{array}{llllllllllllllllllllllllllllllllllll$	B3E4 B3F2 D EDAA 79 ED09 B309 B3F4 B394 B384 B3A4 B3A4 B3A4 B3C4 B390 B3FC B21A	acTF TF TF CZF CC
$ \begin{array}{llllllllllllllllllllllllllllllllllll$	B3F2 D EDAA 79 ED09 B309 B3F4 B394 B384 B384 B3A4 B3A4 B3C4 B390 B3FC B21A	a TF a ZF a c a c a c a c a c b c TF a b F a N a F a F a F a F a C a c TF
$ \begin{array}{llllllllllllllllllllllllllllllllllll$	D EDAA 79 ED09 B309 B3F4 B394 B394 B384 B3A4 B3A4 B3C4 B390 B3FC B21A	a ZF a c a c a c TF a F a N a F a N a F a N a F a C a C TF
$ \begin{array}{cccccccccccccccccccccccccccccccccccc$	79 ED09 B309 B3F4 B394 B384 B3A4 B3C4 B3C4 B390 B3A0 39 B3FC B21A	a c a c TF a a F a N a F a F a C a C TF a c TF
$ \begin{array}{llllllllllllllllllllllllllllllllllll$	B309 B374 B394 B394 B384 B3A4 B3A4 B3C4 B390 B3A0 39 B3FC B21A	a c a c TF a a F a N a F a F a C a c TF
$ \begin{array}{cccccccccccccccccccccccccccccccccccc$	B309 B3F4 B394 B394 B3B4 B3A4 B3C4 B390 B3A0 39 B3FC B21A	ac ac TF
$ \begin{array}{llllllllllllllllllllllllllllllllllll$	B3F4 B394 B394 B3B4 B3A4 B3C4 B390 B3A0 39 B3FC B21A	a c TF a a F a a N a F a N a F a C a C TF
$ \begin{array}{llllllllllllllllllllllllllllllllllll$	B394 B384 B3A4 B3A4 B3C4 B390 B3A0 39 B3FC B21A	a F a N a F a F a C a C TF
$ \begin{array}{llllllllllllllllllllllllllllllllllll$	B394 B3B4 B3A4 B3A4 B3C4 B390 B3A0 39 B3FC B21A	a F a N a F a N a F a F a C
$ \begin{array}{llllllllllllllllllllllllllllllllllll$	B3B4 B3A4 B3C4 B3C9 B3A0 B3A0 B3FC B21A	a N a F a F a C a c TF
DEGBR R_1, R_2 Convert from Fixed (SB \leftarrow 64) RRE DEGBRA R_1, M_3, R_2, M_4 Convert from Fixed (SB \leftarrow 64) RRF DEGR R_1, R_2 Convert from Fixed (SH \leftarrow 64) RRE DELFBR R_1, M_3, R_2, M_4 Convert from Logical (SB \leftarrow 32) RRF DELGBR R_1, M_3, R_2, M_4 Convert from Logical (SB \leftarrow 64) RRF DECR R_1, R_2 Compare (SH) RR DEXTR R_1, R_2 Compare Biased Exponent (ED) RRE DECTC D_2 (B2) Compare and Form Codeword S DEFDBR R_1, M_3, R_2 Convert to Fixed (32 \leftarrow LB) RRF DEFDBRA R_1, M_3, R_2, M_4 Convert to Fixed (32 \leftarrow LB) RRF	B3A4 B3C4 B390 B3A0 39 B3FC B21A	a N a F a N a F a F a c a c TF
$ \begin{array}{llllllllllllllllllllllllllllllllllll$	B3A4 B3C4 B390 B3A0 39 B3FC B21A	¤ F ¤ N ¤ F ¤ F ¤ c ¤ c TF
$ \begin{array}{llllllllllllllllllllllllllllllllllll$	B3C4 B390 B3A0 39 B3FC B21A	¤ N ¤ F ¤ F ¤ c ¤ c TF
$ \begin{array}{llll} \text{CELFBR} & R_1 M_3 R_2 M_4 & \text{Convert from Logical (SB} \leftarrow 32) & \text{RRF-} \\ \text{CELGBR} & R_1 M_3 R_2 M_4 & \text{Convert from Logical (SB} \leftarrow 64) & \text{RRF-} \\ \text{CER} & R_1 R_2 & \text{Compare (SH)} & \text{RR} \\ \text{CEXTR} & R_1 R_2 & \text{Compare Biased Exponent (ED)} & \text{RRE} \\ \text{CFC} & D_2 (B_2) & \text{Compare and Form Codeword} & S \\ \text{CFDBR} & R_1 M_3 R_2 & \text{Convert to Fixed (32} \leftarrow \text{LB)} & \text{RRF-} \\ \text{CFDBRA} & R_1 M_3 R_2 M_4 & \text{Convert to Fixed (32} \leftarrow \text{LB)} & \text{RRF-} \\ \end{array} $	B390 B3A0 39 B3FC B21A	¤F ¤F ¤c ¤cTF
$ \begin{array}{llllllllllllllllllllllllllllllllllll$	B3A0 39 B3FC B21A	¤F ¤c ¤cTF
$ \begin{array}{llllllllllllllllllllllllllllllllllll$	39 B3FC B21A	¤c ¤cTF
	B3FC B21A	¤ c TF
$ \begin{array}{llllllllllllllllllllllllllllllllllll$	B21A	
CFDBR R_1 , M_3 , R_2 Convert to Fixed (32 \leftarrow LB) RRF-CFDBRA R_1 , M_3 , R_2 , M_4 Convert to Fixed (32 \leftarrow LB) RRF-		
CFDBRA R ₁ ,M ₃ ,R ₂ ,M ₄ Convert to Fixed (32←LB) RRF-		
1. 0. 2. 4		
FDR R ₁ ,M ₃ ,R ₂ Convert to Fixed (32←LH) RRF-	B399	
1. 0. 2. 4	B941	
17 07 2	B398	
1. 0. 2. 4	B398	
1, 3, 2	B3B8	
1/2	C2D	c El
	B39A	
1, 0, 5, 4	B39A	
	B3BA	
1. 0. 2. 4	B949	
1. 2. 2. 2	E320	
· · · · ·	B3A9	
	B3A9	
1, 0, 5	B3C9	
. 0 2	B3E1	
17 07 27 4	B3E1	
1. 0. 2	B3A8	
1, 3, 2, 4	B3A8	
1, 0, 5	B3C8	
1, 2, 2, 2,,	E330	
	C2C	c El
CGFR R ₁ ,R ₂ Compare (64←32) RRE	B930	
1, 5	C6C	c GE
	E334	
CGHI R ₁ ,I ₂ Compare Halfword Immediate (64←16) RI-a	A7F	c N
CGHRL R ₁ ,RI ₂ Compare Halfword Relative Long (64←16) RIL-b		c GE
CGHSI D ₁ (B ₁),I ₂ Compare Halfword Immediate (64←16) SIL	E558	
CGIB R ₁ ,I ₂ ,M ₃ ,D ₄ (B ₄) Compare Immediate and Branch (64←8) RIS		¤ GE
CGIJ R ₁ ,I ₂ ,M ₃ ,RI ₄ Compare Immediate and Branch Relative RIE-c (64←8)	EC7C	¤ GE
, ,	EC70	GE
CGR R ₁ ,R ₂ Compare (64) RRE	B920	
CGRB R ₁ ,R ₂ ,M ₃ ,D ₄ (B ₄) Compare and Branch (64) RRS	ECE4	
CGRJ R ₁ ,R ₂ ,M ₃ ,RI ₄ Compare and Branch Relative (64) RIE-b		

Mno			Fo-	٥٣	Class
Mne- monic	Operands	Name	For- mat	Op- code	& Notes
CGRL	R ₁ ,RI ₂	Compare Relative Long (64)	RIL-b	C68	c GE
CGRT	R_1,R_2,M_3	Compare and Trap (64)	RRF-c	B960	GE
CGXBR	R_1, M_3, R_2	Convert to Fixed (64←EB)	RRF-e	B3AA	¤сN
CGXBRA	R_1,M_3,R_2,M_4	Convert to Fixed (64←EB)	RRF-e	ВЗАА	¤сF
CGXR	R_1, M_3, R_2	Convert to Fixed (64←EH)	RRF-e	B3CA	¤сN
CGXTR	R_1, M_3, R_2	Convert to Fixed (64←ED)	RRF-e	B3E9	¤ c TF
CGXTRA	R_1, M_3, R_2, M_4	Convert to Fixed (64←ED)	RRF-e	B3E9	¤сF
CH	$R_1,D_2(X_2,B_2)$	Compare Halfword (32←16)	RX-a	49	С
CHF	$R_1,D_2(X_2,B_2)$	Compare High (32)	RXY-a	E3CD	
CHHR	R ₁ ,R ₂	Compare High (32)	RRE	B9CD	
CHHSI	$D_1(B_1),I_2$	Compare Halfword Immediate (16←16)	SIL	E554	c GE
CHI	R_1,I_2	Compare Halfword Immediate (32←16)	RI-a	A7E	С
CHLR	R ₁ ,R ₂	Compare High (32)	RRE	B9DD	c HW
CHRL	R ₁ ,RI ₂	Compare Halfword Relative Long (32←16)		C65	c GE
CHSI	$D_1(B_1),I_2$	Compare Halfword Immediate (32←16)	SIL	E55C	
CHY	$R_1,D_2(X_2,B_2)$	Compare Halfword (32←16)	RXY-a	E379	c LD
CIB	$R_1, I_2, M_3, D_4(B_4)$	Compare Immediate and Branch (32←8)	RIS	ECFE	¤ GE
CIH	R_1,I_2	Compare Immediate High (32)	RIL-a		
CIJ	R ₁ ,I ₂ ,M ₃ ,RI ₄	Compare Immediate and Branch Relative (32←8)		EC7E	
CIT	R_1,I_2,M_3	Compare Immediate and Trap (32←16)		EC72	GE
CKSM	R ₁ ,R ₂	Checksum	RRE	B241	шС
CL	$R_1,D_2(X_2,B_2)$	Compare Logical (32)	RX-a	55	С
CLC	$D_1(L,B_1),D_2(B_2)$	Compare Logical (character)	SS-a	D5	пc
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-a	A9	шС
CLCLU	$R_1, R_3, D_2(B_2)$	Compare Logical Long Unicode	RSY-a	EB8F	□ c E2
CLFDBR	R_1,M_3,R_2,M_4	Convert to Logical (32←LB)	RRF-e	B39D	¤сF
CLFDTR	R_1, M_3, R_2, M_4	Convert to Logical (32←LD)	RRF-e		
CLFEBR	R_1, M_3, R_2, M_4	Convert to Logical (32←SB)	RRF-e	B39C	¤cF
CLFHSI	$D_1(B_1), I_2$	Compare Logical Immediate (32←16)	SIL	E55D	c GE
CLFI	R_1,I_2	Compare Logical Immediate (32)	RIL-a	C2F	c El
CLFIT	R ₁ ,I ₂ ,M ₃	Compare Logical Immediate and Trap (32←16)		EC73	
	R_1, M_3, R_2, M_4	Convert to Logical (32←EB)	RRF-e		
CLFXTR	R_1, M_3, R_2, M_4	Convert to Logical (32←ED)	RRF-e	B94B	¤сF
CLG	$R_1,D_2(X_2,B_2)$	Compare Logical (64)	RXY-a	E321	сN
CLGDBR	R_1, M_3, R_2, M_4	Convert to Logical (64←LB)	RRF-e	B3AD	¤cF
CLGDTR	R_1, M_3, R_2, M_4	Convert to Logical (64←LD)	RRF-e	B942	¤cF
CLGEBR	R_1, M_3, R_2, M_4	Convert to Logical (64←SB)	RRF-e	B3AC	¤cF
CLGF	$R_1,D_2(X_2,B_2)$	Compare Logical (64←32)	RXY-a	E331	сN
CLGFI	R_1,I_2	Compare Logical Immediate (64←32)	RIL-a	C2E	c El
CLGFR	R ₁ ,R ₂	Compare Logical (64←32)	RRE	B931	сN
CLGFRL	R ₁ ,RI ₂	Compare Logical Relative Long (64←32)	RIL-b	C6E	c GE
CLGHRL	R ₁ ,RI ₂	Compare Logical Relative Long (64←16)	RIL-b	C66	c GE
CLGHSI	$D_1(B_1), I_2$	Compare Logical Immediate (64←16)	SIL	E559	c GE
CLGIB	$R_1, I_2, M_3, D_4(B_4)$	Compare Logical Immediate and Branch (64←8)	RIS	ECFD	¤ GE
CLGIJ	R ₁ ,I ₂ ,M ₃ ,RI ₄	Compare Logical Immediate and Branch Relative (64←8)		EC7D	
CLGIT	R ₁ ,I ₂ ,M ₃	Compare Logical Immediate and Trap (64←16)		EC71	
CLGR	R ₁ ,R ₂	Compare Logical (64)	RRE	B921	
CLGRB	$R_1, R_2, M_3, D_4(B_4)$	Compare Logical and Branch (64)	RRS	ECE5	
CLGRJ	R_1,R_2,M_3,RI_4	Compare Logical and Branch Relative (64)	RIE-b		
CLGRL	R ₁ ,RI ₂	Compare Logical Relative Long (64)	RIL-b		c GE
CLGRT	R_1,R_2,M_3	Compare Logical and Trap (64)	RRF-c		GE
CLGT	$R_1,M_3,D_2(B_2)$	Compare Logical and Trap (64)		EB2B	
	R_1,M_3,R_2,M_4	Convert to Logical (64←EB)		B3AE	
	R_1,M_3,R_2,M_4	Convert to Logical (64←ED)		B94A	
CLHF	$R_1,D_2(X_2,B_2)$	Compare Logical High (32)	RXY-a	E3CF	c HW

Mne- monic	Operands	Name	For- mat	Op- code	Class & Notes
CLHHR	R ₁ ,R ₂	Compare Logical High (32)	RRE	B9CF	c HW
CLHHSI	$D_1(B_1),I_2$	Compare Logical Immediate (16←16)	SIL	E555	c GE
CLHLR	R ₁ ,R ₂	Compare Logical High (32)	RRE	B9DF	c HW
CLHRL	R ₁ ,RI ₂	Compare Logical Relative Long (32←16)	RIL-b	C67	c GE
CLI	D ₁ (B ₁),l ₂	Compare Logical Immediate	SI	95	С
CLIB	$R_1, I_2, M_3, D_4(B_4)$	Compare Logical Immediate and Branch (32←8)	RIS	ECFF	¤ GE
CLIH	R_1, I_2	Compare Logical Immediate High (32)	RIL-a	CCF	c HW
CLIJ	R ₁ ,I ₂ ,M ₃ ,RI ₄	Compare Logical Immediate and Branch Relative (32←8)	RIE-c	EC7F	¤ GE
CLIY	$D_1(B_1), I_2$	Compare Logical Immediate	SIY	EB55	c LD
CLM	$R_1, M_3, D_2(B_2)$	Compare Logical Char. under Mask (low)	RS-b	BD	С
CLMH	R ₁ ,M ₃ ,D ₂ (B ₂)	Compare Logical Char. under Mask (high)	RSY-b	EB20	сN
CLMY	$R_1, M_3, D_2(B_2)$	Compare Logical Char. under Mask (low)	RSY-b	EB21	c LD
CLR	R ₁ ,R ₂	Compare Logical (32)	RR	15	С
CLRB	R ₁ ,R ₂ ,M ₃ ,D ₄ (B ₄)	Compare Logical and Branch (32)	RRS	ECF7	
CLRJ	R ₁ ,R ₂ ,M ₃ ,RI ₄	Compare Logical and Branch Relative (32)	RIE-b		
CLRL		Compare Logical Relative Long (32)	RIL-b		c GE
	R ₁ ,RI ₂				
CLRT	R ₁ ,R ₂ ,M ₃	Compare Logical and Trap (32)	RRF-c		GE
CLST	R ₁ ,R ₂	Compare Logical String	RRE	B25D	
CLT	R ₁ ,M ₃ ,D ₂ (B ₂)	Compare Logical and Trap (32)	RSY-b		
CLY	$R_1,D_2(X_2,B_2)$	Compare Logical (32)	RXY-a		c LD
CMPSC	R_1,R_2	Compression Call	RRE	B263	Ι¤С
CP	$D_1(L_1,B_1),D_2(L_2,B_2)$	Compare Decimal	SS-b	F9	шС
CPDT	$R_1,D_2(L_2,B_2),M_3$	Convert to Packed (From Long DFP)	RSL-b	EDAC	c PC
CPSDR	R ₁ ,R ₃ ,R ₂	Copy Sign (L)	RRF-b	B372	¤ FS
CPXT	R ₁ ,D ₂ (L ₂ ,B ₂),M ₃	Convert to Packed (From Extended DFP)	RSL-b	EDAD	c PC
CPYA	R ₁ ,R ₂	Copy Access	RRE	B24D	
CR	R ₁ ,R ₂	Compare (32)	RR	19	С
CRB	R ₁ ,R ₂ ,M ₃ ,D ₄ (B ₄)	Compare and Branch (32)	RRS	ECF6	
CRDTE	R ₁ ,R ₃ ,R ₂ [,M ₄]	Compare and Replace DAT Table Entry	RRF-b		
CRJ	R ₁ ,R ₂ ,M ₃ ,RI ₄	Compare and Branch Relative (32)	RIE-b	EC76	
CRL	R ₁ ,RI ₂	Compare Relative Long (32)	RIL-b		c GE
CRT	R ₁ ,R ₂ ,M ₃	Compare and Trap (32)	RRF-c		GE
CS		Compare and Swap (32)	RS-a	BA	¤ C
CSCH	$R_1, R_3, D_2(B_2)$				
CSDTR	R. R. M.	Clear Subchannel Convert to Signed Packed (64←LD)	S RRF-d	B230	p C
CSG	R ₁ ,R ₂ ,M ₄		RSY-a		
CSP	R ₁ ,R ₃ ,D ₂ (B ₂)	Compare and Swap (64)			
	R ₁ ,R ₂	Compare and Swap and Purge (32)	RRE	B250	p c
CSPG	R ₁ ,R ₂	Compare and Swap and Purge (64)	RRE		p c DE
CSST	D ₁ (B ₁),D ₂ (B ₂),R ₃	Compare and Swap and Store	SSF	C82	Ω C
CSXTR	R ₁ ,R ₂ ,M ₄	Convert to Signed Packed (128←ED)	RRF-d		
CSY	$R_1, R_3, D_2(B_2)$	Compare and Swap (32)			¤ c LD
CU12	$R_1, R_2[, M_3]$	Convert UTF-8 to UTF-16	RRF-c		
CU14	$R_1, R_2[, M_3]$	Convert UTF-8 to UTF-32	RRF-c	B9B0	¤cE3
CU21	$R_1,R_2[,M_3]$	Convert UTF-16 to UTF-8	RRF-c	B2A6	пc
CU24	$R_1, R_2[, M_3]$	Convert UTF-16 to UTF-32	RRF-c	B9B1	¤cE3
CU41	R ₁ ,R ₂	Convert UTF-32 to UTF-8	RRE	B9B2	¤ c E3
CU42	R ₁ ,R ₂	Convert UTF-32 to UTF-16	RRE	B9B3	¤ c E3
CUDTR	R ₁ ,R ₂	Convert to Unsigned Packed (64←LD)	RRE	B3E2	¤ TF
CUSE	R ₁ ,R ₂	Compare until Substring Equal	RRE		
CUTFU	R ₁ ,R ₂ [,M ₃]	Convert UTF-8 to Unicode	RRF-c		
		Convert Unicode to UTF-8			
CUUTF	R ₁ ,R ₂ [,M ₃]		RRF-c		
CUXTR	R ₁ ,R ₂	Convert to Unsigned Packed (128←ED)	RRE	B3EA	
CVB	$R_1, D_2(X_2, B_2)$	Convert to Binary (32)	RX-a	4F	σ
CVBG	$R_1, D_2(X_2, B_2)$	Convert to Binary (64)	RXY-a		
CVBY	$R_1, D_2(X_2, B_2)$	Convert to Binary (32)	RXY-a		
CVD	$R_1,D_2(X_2,B_2)$	Convert to Decimal (32)	RX-a	4E	۵
OVD					
CVDG	$R_1, D_2(X_2, B_2)$	Convert to Decimal (64)	RXY-a	E32E	¤Ν

Mne-			For-	Ор-	Clas &
monic	Operands	Name	mat	code	Note
CXBR	R ₁ ,R ₂	Compare (EB)	RRE	B349	αс
CXFBR	R ₁ ,R ₂	Convert from Fixed (EB←32)	RRE	B396	۵
CXFBRA	R_1, M_3, R_2, M_4	Convert from Fixed (EB←32)	RRF-e	B396	¤F
CXFR	R ₁ ,R ₂	Convert from Fixed (EH←32)	RRE	B3B6	۵
CXFTR	R_1,M_3,R_2,M_4	Convert from Fixed (ED←32)	RRE	B959	¤F
CXGBR	R ₁ ,R ₂	Convert from Fixed (EB←64)	RRE	B3A6	
	R_1, M_3, R_2, M_4	Convert from Fixed (EB←64)		B3A6	
CXGR	R ₁ ,R ₂	Convert from Fixed (EH←64)	RRE	B3C6	
CXGTR	R ₁ ,R ₂	Convert from Fixed (ED←64)	RRE	B3F9	
	R ₁ ,M ₃ ,R ₂ ,M ₄	Convert from Fixed (ED←64)		B3F9	
	R ₁ ,M ₃ ,R ₂ ,M ₄	Convert from Logical (EB←32)	RRF-e		
CXLFTR	R ₁ ,M ₃ ,R ₂ ,M ₄	Convert from Logical (ED←32)		B95B	
	R ₁ ,M ₃ ,R ₂ ,M ₄	Convert from Logical (EB←64)		B3A2	
	R ₁ ,M ₃ ,R ₂ ,M ₄	Convert from Logical (ED←64)		B95A	
CXPT	$R_1,D_2(L_2,B_2),M_3$	Convert from Packed (To Extended DFP)		EDAF	
CXR	R ₁ ,R ₂	Compare (EH)	RRE	B369	
CXSTR	R ₁ ,R ₂	Convert from Signed Packed (ED←128)	RRE	B3FB	
CXTR	R ₁ ,R ₂	Compare (ED)	RRE	B3EC	
CXUTR	R ₁ ,R ₂	Convert from Unsigned Packed (ED←128)		B3FA	
CXZT	$R_1,D_2(L_2,B_2),M_3$	Convert from Zoned (to extended DFP)		EDAB	
CY	R ₁ ,D ₂ (X ₂ ,B ₂)	Compare (32)		E359	
CZDT	R ₁ ,D ₂ (L ₂ ,B ₂),M ₃	Convert to Zoned (from long DFP) Convert to Zoned (from extended DFP)		EDA8 EDA9	
CZXT D	$R_1,D_2(L_2,B_2),M_3$,	RX-a	5D	u ZF
DD	$R_1, D_2(X_2, B_2)$	Divide (32←64) Divide (LH)	RX-a	6D	۵
DDB	$R_1,D_2(X_2,B_2)$ $R_1,D_2(X_2,B_2)$	Divide (LB)	RXE	ED1D	
DDBR	R ₁ ,R ₂	Divide (LB)	RRE	B31D	
DDR	R ₁ ,R ₂	Divide (LH)	RR	2D	۵
DDTR	R ₁ ,R ₂ ,R ₃	Divide (LD)	RRF-a		
DDTRA	R ₁ ,R ₂ ,R ₃ ,M ₄	Divide (LD)		B3D1	
DE	R ₁ ,D ₂ (X ₂ ,B ₂)	Divide (SH)	RX-a	7D	۵
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	۵
DIDBR	R ₁ ,R ₃ ,R ₂ ,M ₄	Divide to Integer (LB)		B35B	
DIEBR	R ₁ ,R ₃ ,R ₂ ,M ₄	Divide to Integer (SB)		B353	
DL	R ₁ ,D ₂ (X ₂ ,B ₂)	Divide Logical (32←64)		E397	
DLG	$R_1, D_2(X_2, B_2)$	Divide Logical (64←128)	RXY-a	E387	¤Ν
DLGR	R ₁ ,R ₂	Divide Logical (64←128)	RRE	B987	¤Ν
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)		SS-b	FD	۵
DR	R ₁ ,R ₂	Divide (32←64)	RR	1D	۵
DSG	$R_1, D_2(X_2, B_2)$	Divide Single (64)	RXY-a	E30D	¤Ν
DSGF	$R_1,D_2(X_2,B_2)$	Divide Single (64←32)	RXY-a	E31D	пN
DSGFR	R_1,R_2	Divide Single (64←32)	RRE	B91D	пN
DSGR	R ₁ ,R ₂	Divide Single (64)	RRE	B90D	¤Ν
DXBR	R_1,R_2	Divide (EB)	RRE	B34D	۵
DXR	R_1,R_2	Divide (EH)	RRE	B22D	۵
DXTR	R_1,R_2,R_3	Divide (ED)	RRF-a	B3D9	¤ TF
DXTRA	R_1, R_2, R_3, M_4	Divide (ED)		B3D9	۵F
EAR	R_1,R_2	Extract Access	RRE	B24F	
ECAG	$R_1, R_3, D_2(B_2)$	Extract CPU Attribute		EB4C	
ECTG	$D_1(B_1), D_2(B_2), R_3$	Extract CPU Time	SSF	C81	¤ ET
ED	$D_1(L,B_1),D_2(B_2)$	Edit	SS-a	DE	ΩС
EDMK	$D_1(L,B_1),D_2(B_2)$	Edit and Mark	SS-a	DF	αс
EEDTR	R_1,R_2	Extract Biased Exponent (64←LD)	RRE	B3E5	
EEXTR	R_1,R_2	Extract Biased Exponent (64←ED)	RRE	B3ED	
EFPC	R ₁	Extract FPC	RRE	B38C	۵
EPAIR	R ₁	Extract Primary ASN and Instance	RRE	B99A	q RA
EPAR	R ₁	Extract Primary ASN	RRE	B226	q

Mne- monic	Operands	Name	For- mat	Op- code	CI:
EPSW	R ₁ ,R ₂	Extract PSW	RRE	B98D	¤Ν
EREG	R ₁ ,R ₂	Extract Stacked Registers (32)	RRE		¤
EREGG	R ₁ ,R ₂	Extract Stacked Registers (64)	RRE	B90E	۵N
ESAIR	R ₁	Extract Secondary ASN and Instance	RRE	B99B	
ESAR	R ₁	Extract Secondary ASN	RRE	B227	
ESDTR	R ₁ ,R ₂	Extract Significance (64←LD)	RRE	B3E7	
ESEA	R ₁ ,R ₂	Extract and Set Extended Authority	RRE	B99D	
ESTA	R ₁ ,R ₂	Extract Stacked State	RRE	B24A	
ESXTR	R ₁ ,R ₂	Extract Significance (64←ED)	RRE	B3EF	
ETND	R ₁	Extract Transaction Nesting Depth	RRE	B2EC	-
EX	$R_1,D_2(X_2,B_2)$	Execute	RX-a	44	σ,
EXRL	R ₁ ,RI ₂	Execute Relative Long	RIL-b		۵)
FIDBR	R ₁ ,M ₃ ,R ₂	Load FP Integer (LB)	RRF-e		
FIDBRA			RRF-e		
FIDR	R ₁ ,M ₃ ,R ₂ ,M ₄	Load FP Integer (LB)	RRE	B37F	
	R ₁ ,R ₂	Load FP Integer (LH)			
FIDTR	R ₁ ,M ₃ ,R ₂ ,M ₄	Load FP Integer (LD)	RRF-e		
FIEBR	R ₁ ,M ₃ ,R ₂	Load FP Integer (SB)	RRF-e		
FIEBRA	R ₁ ,M ₃ ,R ₂ ,M ₄	Load FP Integer (SB)	RRF-e		
FIER	R ₁ ,R ₂	Load FP Integer (SH)	RRE	B377	
FIXBR	R ₁ ,M ₃ ,R ₂	Load FP Integer (EB)	RRF-e		0
FIXBRA	R ₁ ,M ₃ ,R ₂ ,M ₄	Load FP Integer (EB)	RRF-e		
FIXR	R ₁ ,R ₂	Load FP Integer (EH)	RRE	B367	
FIXTR	R ₁ ,M ₃ ,R ₂ ,M ₄	Load FP Integer (ED)	RRF-e		
FLOGR	R ₁ ,R ₂	Find Leftmost One	RRE	B983	
HDR	R ₁ ,R ₂	Halve (LH)	RR	24	۵
HER	R_1,R_2	Halve (SH)	RR	34	۵
HSCH	_	Halt Subchannel	S	B231	ро
IAC	R ₁	Insert Address Space Control	RRE	B224	qc
IC	$R_1,D_2(X_2,B_2)$	Insert Character	RX-a	43	
ICM	$R_1, M_3, D_2(B_2)$	Insert Characters under Mask (low)	RS-b	BF	С
ICMH	$R_1, M_3, D_2(B_2)$	Insert Characters under Mask (high)	RSY-b	EB80	сN
ICMY	$R_1, M_3, D_2(B_2)$	Insert Characters under Mask (low)	RSY-b	EB81	сL
ICY	$R_1,D_2(X_2,B_2)$	Insert Character	RXY-a		LD
IDTE	R_1, R_3, R_2	Invalidate DAT Table Entry	RRF-b	B98E	pι
IEDTR	R_1, R_3, R_2	Insert Biased Exponent (LD←64&LD)	RRF-b	B3F6	٦α
IEXTR	R_1, R_3, R_2	Insert Biased Exponent (ED←64&ED)	RRF-b	B3FE	٦α
IIHF	R_1,I_2	Insert Immediate (high)	RIL-a	C08	ΕI
IIHH	R_1,I_2	Insert Immediate (high high)	RI-a	A50	Ν
IIHL	R_1, I_2	Insert Immediate (high low)	RI-a	A51	Ν
IILF	R_1,I_2	Insert Immediate (low)	RIL-a	C09	El
IILH	R_1,I_2	Insert Immediate (low high)	RI-a	A52	Ν
IILL	R_1,I_2	Insert Immediate (low low)	RI-a	A53	Ν
IPK		Insert PSW Key	S	B20B	q
IPM	R ₁	Insert Program Mask	RRE	B222	
IPTE	R ₁ ,R ₂	Invalidate Page Table Entry	RRF-a	B221	р
IRBM	R ₁ ,R ₂	Insert Reference Bits Multiple	RRE	B2AC	pΙ
ISKE	R_1,R_2	Insert Storage Key Extended	RRE	B229	р
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	۵۵
KDBR	R ₁ ,R ₂	Compare and Signal (LB)	RRE	B318	۵۵
KDTR	R ₁ ,R ₂	Compare and Signal (LD)	RRE	B3E0	
KEB	$R_1,D_2(X_2,B_2)$	Compare and Signal (SB)	RXE	ED08	۵۵
KEBR	R ₁ ,R ₂	Compare and Signal (SB)	RRE	B308	۵۵
KIMD	R ₁ ,R ₂	Compute Intermediate Message Digest	RRE	B93E	
KLMD	R ₁ ,R ₂	Compute Last Message Digest	RRE	B93F	
KM	R ₁ ,R ₂	Cipher Message	RRE	B92E	
KMA	R ₁ ,R ₃ ,R ₂	Cipher Message with Authentication	RRF-b		
KMAC	R ₁ ,R ₂	Compute Message Authentication Code	RRE	B91E	
		•			
KMC	R_1,R_2	Cipher Message with Chaining	RRE	B92F	D C

					Class
Mne- monic	Operands	Name	For- mat	Op- code	& Notes
KMF	R ₁ ,R ₂	Cipher Message with Cipher Feedback	RRE	B92A	¤ c M4
KMO	R ₁ ,R ₂	Cipher Message with Output Feedback	RRE		¤ c M4
KXBR	R ₁ ,R ₂	Compare and Signal (EB)	RRE	B348	
KXTR L	R ₁ ,R ₂	Compare and Signal (ED)	RRE	B3E8 58	¤ cTF
LA	$R_1, D_2(X_2, B_2)$	Load (32) Load Address	RX-a RX-a	56 41	
LAA	$R_1,D_2(X_2,B_2)$ $R_1,R_3,D_2(B_2)$	Load and Add (32)		EBF8	ποIΔ
LAAG	R ₁ ,R ₃ ,D ₂ (B ₂)	Load and Add (64)		EBE8	
LAAL	R ₁ ,R ₃ ,D ₂ (B ₂)	Load and Add Logical (32)		EBFA	
LAALG	R ₁ ,R ₃ ,D ₂ (B ₂)	Load and Add Logical (64)	RSY-a	EBEA	¤сIА
LAE	$R_1,D_2(X_2,B_2)$	Load Address Extended	RX-a	51	۵
LAEY	$R_1,D_2(X_2,B_2)$	Load Address Extended	RXY-a	E375	¤ GE
LAM	$R_1, R_3, D_2(B_2)$	Load Access Multiple	RS-a	9A	۵
LAMY	$R_1, R_3, D_2(B_2)$	Load Access Multiple	RSY-a	EB9A	¤LD
LAN	$R_1, R_3, D_2(B_2)$	Load and AND (32)		EBF4	
LANG	$R_1, R_3, D_2(B_2)$	Load and AND (64)		EBE4	
LAO	R ₁ ,R ₃ ,D ₂ (B ₂)	Load and OR (32)		EBF6	
LAOG	R ₁ ,R ₃ ,D ₂ (B ₂)	Load and OR (64)		EBE6	
LARL	R ₁ ,Rl ₂	Load Address Relative Long	RIL-b		N3
LASP	D ₁ (B ₁),D ₂ (B ₂)	Load Address Space Parameters	SSE	E500	
LAT LAX	R ₁ ,D ₂ (X ₂ ,B ₂)	Load and Trap (32)		E39F	
LAXG	R ₁ ,R ₃ ,D ₂ (B ₂)	Load and Exclusive OR (32) Load and Exclusive OR (64)		EBF7 EBE7	
LAXG	$R_1, R_3, D_2(B_2)$ $R_1, D_2(X_2, B_2)$	Load Address		E371	
LB	$R_1,D_2(X_2,B_2)$ $R_1,D_2(X_2,B_2)$	Load Byte (32←8)	RXY-a		LD
LBH	$R_1,D_2(X_2,B_2)$ $R_1,D_2(X_2,B_2)$	Load Byte High (32←8)		E3C0	
LBR	R ₁ ,R ₂	Load Byte (32←8)	RRE	B926	
LCBB	$R_1, D_2(X_2, B_2), M_3$	Load Count to Block Boundary	RXE	E727	
LCDBR	R ₁ ,R ₂	Load Complement (LB)	RRE	B313	
LCDFR	R ₁ ,R ₂	Load Complement (L)	RRE	B373	¤ FS
LCDR	R ₁ ,R ₂	Load Complement (LH)	RR	23	ΩС
LCEBR	R_1,R_2	Load Complement (SB)	RRE	B303	пс
LCER	R_1,R_2	Load Complement (SH)	RR	33	пС
LCGFR	R_1,R_2	Load Complement (64←32)	RRE	B913	
LCGR	R ₁ ,R ₂	Load Complement (64)	RRE	B903	c N
LCR	R ₁ ,R ₂	Load Complement (32)	RR	13	С
LCTL	R ₁ ,R ₃ ,D ₂ (B ₂)	Load Control (32)	RS-a	B7	p
LCTLG	R ₁ ,R ₃ ,D ₂ (B ₂)	Load Control (64)		EB2F	
LCXBR	R ₁ ,R ₂	Load Complement (EB)	RRE	B343	пС
LCXR LD	R ₁ ,R ₂	Load Complement (EH) Load (L)	RRE RX-a	B363 68	a C
LDE	$R_1,D_2(X_2,B_2)$ $R_1,D_2(X_2,B_2)$	Load (L) 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	D
LDER	R ₁ ,R ₂	Load Lengthened (LH←SH)	RRE	B324	
LDETR	R ₁ ,R ₂ ,M ₄	Load Lengthened (LD←SD)	RRF-d	B3D4	¤ TF
LDGR	R ₁ ,R ₂	Load FPR from GR (L←64)	RRE	B3C1	¤ FG
LDR	R ₁ ,R ₂	Load (L)	RR	28	۵
LDXBR	R ₁ ,R ₂	Load Rounded (LB←EB)	RRE	B345	۵
LDXBRA	R_1,M_3,R_2,M_4	Load Rounded (LB←EB)	RRF-e	B345	۵F
LDXR	R_1,R_2	Load Rounded (LH←EH)	RR	25	۵
LDXTR	R_1, M_3, R_2, M_4	Load Rounded (LD←ED)		B3DD	
LDY	$R_1, D_2(X_2, B_2)$	Load (L)		ED65	
LE	$R_1,D_2(X_2,B_2)$	Load (S)	RX-a	78	۵
LEDBR	R ₁ ,R ₂	Load Rounded (SB←LB)	RRE	B344	σ_
LEDBRA	17 U7 E7 T	Load Rounded (SB←LB)		B344	
LEDR	R ₁ ,R ₂	Load Rounded (SH←LH)	RR	35	¤ ~ TF
LEDTR	R ₁ ,M ₃ ,R ₂ ,M ₄	Load Rounded (SD←LD)		B3D5	
LER	R ₁ ,R ₂	Load (S)	RR	38	0
LEXBR	R ₁ ,R ₂	Load Rounded (SB←EB)	RRE	B346	۵

Mne- monic	Operande	Name	For- mat	Op-	Class & Notes
LEXBRA	Operands	Load Rounded (SB←EB)	RRF-e	Code	¤ F
LEXR	R ₁ ,M ₃ ,R ₂ ,M ₄		RRE	B366	0
LEY	R_1, R_2 $R_1, D_2(X_2, B_2)$	Load Rounded (SH←EH) Load (S)		ED64	
LFAS	$D_2(B_2)$	Load FPC and Signal	S	B2BD	
LFH	$R_1, D_2(X_2, B_2)$	Load High (32)		E3CA	
LFHAT	$R_1,D_2(X_2,B_2)$	Load and Trap (32H←32)		E3C8	
LFPC	$D_2(B_2)$	Load FPC	S	B29D	
LG	$R_1, D_2(X_2, B_2)$	Load (64)		E304	
LGAT	$R_1,D_2(X_2,B_2)$	Load and Trap (64)	RXY-a		LT
LGB	$R_1,D_2(X_2,B_2)$	Load Byte (64←8)		E377	
LGBR	R ₁ ,R ₂	Load Byte (64←8)	RRE	B906	
LGDR	R ₁ ,R ₂	Load GR from FPR (64←L)	RRE	B3CD	
LGF	$R_1, D_2(X_2, B_2)$	Load (64←32)	RXY-a		N.
LGFI	R ₁ ,l ₂	Load Immediate (64←32)	RIL-a		El
LGFR	R ₁ ,R ₂	Load (64←32)	RRE	B914	
LGFRL	R ₁ ,Rl ₂	Load Relative Long (64←32)	RIL-b		GE
LGG	$R_1, D_2(X_2, B_2)$	Load guarded (64)		E34C	
LGH	$R_1,D_2(X_2,B_2)$	Load Halfword (64←16)		E315	
LGHI	R ₁ ,I ₂	Load Halfword Immediate (64←16)	RI-a	A79	N
LGHR	R ₁ ,R ₂	Load Halfword (64←16)	RRE	B907	
LGHRL	R ₁ ,Rl ₂	Load Halfword Relative Long (64←16)	RIL-b		GE
LGR	R ₁ ,R ₂	Load (64)	RRE	B904	N
LGRL	R ₁ ,Rl ₂	Load Relative Long (64)	RIL-b		GE
LGSC	$R_1, D_2(X_2, B_2)$	Load guarded storage controls		E34D	
LH	$R_1, D_2(X_2, B_2)$	Load Halfword (32←16)	RX-a	48	٠.
LHH	$R_1,D_2(X_2,B_2)$	Load Halfword High (32←16)		E3C4	HW
LHI	R ₁ ,l ₂	Load Halfword Immediate (32←16)	RI-a	A78	
LHR	R ₁ ,R ₂	Load Halfword (32←16)	RRE	B927	ΕI
LHRL	R ₁ ,Rl ₂	Load Halfword Relative Long (32←16)	RIL-b		GE
LHY	$R_1, D_2(X_2, B_2)$	Load Halfword (32←16)		E378	LD
LLC	$R_1,D_2(X_2,B_2)$	Load Logical Character (32←8)		E394	
LLCH	$R_1,D_2(X_2,B_2)$	Load Logical Character High (32←8)		E3C2	
LLCR	R ₁ ,R ₂	Load Logical Character (32←8)	RRE	B994	El
LLGC	$R_1, D_2(X_2, B_2)$	Load Logical Character (64←8)	RXY-a		N.
LLGCR	R ₁ ,R ₂	Load Logical Character (64←8)	RRE	B984	El
LLGF	$R_1, D_2(X_2, B_2)$	Load Logical (64←32)	RXY-a		N.
LLGFAT	$R_1,D_2(X_2,B_2)$	Load and Trap (64←32)		E39D	
LLGFR	R ₁ ,R ₂	Load Logical (64←32)	RRE	B916	
LLGFRL	R ₁ ,Rl ₂	Load Logical Relative Long (64←32)	RIL-b		GE
	$R_1, D_2(X_2, B_2)$	Load logical and shift guarded (64←32)		E348	
LLGH	$R_1,D_2(X_2,B_2)$	Load Logical Halfword (64←16)	RXY-a		N
LLGHR	R ₁ ,R ₂	Load Logical Halfword (64←16)	RRE	B985	El
LLGHRL		Load Logical Halfword Relative Long	RIL-b		GE
LEGITILE	11,1112	(64←16)		040	۵L
LLGT	$R_1,D_2(X_2,B_2)$	Load Logical Thirty One Bits (64←31)	RXY-a	E317	N
LLGTAT	$R_1,D_2(X_2,B_2)$	Load Logical Thirty One Bits and Trap	RXY-a	E39C	LT
		(64←31)			
LLGTR	R ₁ ,R ₂	Load Logical Thirty One Bits (64←31)	RRE	B917	N
LLH	$R_1,D_2(X_2,B_2)$	Load Logical Halfword (32←16)	RXY-a		El
LLHH	$R_1,D_2(X_2,B_2)$	Load Logical Halfword High (32←16)		E3C6	
LLHR	R_1,R_2	Load Logical Halfword (32←16)	RRE	B995	El
LLHRL	R ₁ ,Rl ₂	Load Logical Halfword Relative Long (32←16)	RIL-b	C42	GE
LLIHF	R ₁ ,l ₂	Load Logical Immediate (high)	RIL-a	C0E	EI
LLIHH	R ₁ ,l ₂	Load Logical Immediate (high high)	RI-a	A5C	N
LLIHL	R ₁ ,l ₂	Load Logical Immediate (high low)	RI-a	A5D	N
LLILF	R ₁ ,l ₂	Load Logical Immediate (low)	RIL-a	C0F	EI
LLILH	R ₁ ,l ₂	Load Logical Immediate (low high)	RI-a	A5E	N
LLILL	R ₁ ,l ₂	Load Logical Immediate (low low)	RI-a	A5F	N
LLZRGF	R ₁ ,D ₂ (X ₂ ,B ₂)	Load Logical and Zero Rightmost Byte (32)			LZ
LM	$R_1, R_3, D_2(B_2)$	Load Multiple (32)	RS-a	98	

					Class
Mne- monic	Operands	Name	For- mat	Op- code	& Notes
LMD	R ₁ ,R ₃ ,D ₂ (B ₂),D ₄ (B ₄)	Load Multiple Disjoint (64←32&32)	SS-e	EF	¤Ν
LMG	$R_1, R_3, D_2(B_2)$	Load Multiple (64)	RSY-a	EB04	N
LMH	$R_1, R_3, D_2(B_2)$	Load Multiple High	RSY-a	EB96	N
LMY	$R_1, R_3, D_2(B_2)$	Load Multiple (32)	RSY-a	EB98	LD
LNDBR	R_1,R_2	Load Negative (LB)	RRE	B311	αС
LNDFR	R_1,R_2	Load Negative (L)	RRE	B371	¤ FS
LNDR	R_1,R_2	Load Negative (LH)	RR	21	ΩС
LNEBR	R_1,R_2	Load Negative (SB)	RRE	B301	αС
LNER	R_1,R_2	Load Negative (SH)	RR	31	ΩС
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	С
LNXBR	R_1,R_2	Load Negative (EB)	RRE	B341	шC
LNXR	R ₁ ,R ₂	Load Negative (EH)	RRE	B361	αС
LOC	$R_1,D_2(B_2),M_3$	Load on Condition (32)	RSY-b	EBF2	L1
LOCFH	$R_1,D_2(B_2),M_3$	Load High on Condition (32)	RSY-b	EBE0	L2
LOCFHR	R ₁ ,R ₂ ,M ₃	Load High on Condition (32)	RRF-c	B9E0	L2
LOCG	$R_1, D_2(B_2), M_3$	Load on Condition (64)	RSY-b	EBE2	L1
LOCGHI	R_1,I_2,M_3	Load Halfword Immediate on Condition (64←16)	RIE-g	EC46	L2
LOCGR	R_1, R_2, M_3	Load on Condition (64)	RRF-c	B9E2	L1
LOCHHI	R ₁ ,I ₂ ,M ₃	Load Halfword High Immediate on Condition (32←16)	RIE-g	EC4E	L2
LOCHI	R_1, I_2, M_3	Load Halfword Immediate on Condition (32←16)	RIE-g	EC42	L2
LOCR	R_1, R_2, M_3	Load on Condition (32)	RRF-c	B9F2	L1
LPD	$R_3, D_1(B_1), D_2(B_2)$	Load Pair Disjoint (32)	SSF	C84	c IA
LPDBR	R ₁ ,R ₂	Load Positive (LB)	RRE	B310	αс
LPDFR	R ₁ ,R ₂	Load Positive (L)	RRE	B370	¤ FS
LPDG	R ₃ ,D ₁ (B ₁),D ₂ (B ₂)	Load Pair Disjoint (64)	SSF	C85	c IA
LPDR	R ₁ ,R ₂	Load Positive (LH)	RR	20	αс
LPEBR	R ₁ ,R ₂	Load Positive (SB)	RRE	B300	αс
LPER	R ₁ ,R ₂	Load Positive (SH)	RR	30	αс
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 (64&64←128)	RXY-a	E38F	¤Ν
LPR	R ₁ ,R ₂	Load Positive (32)	RR	10	С
LPSW	D ₂ (B ₂)	Load PSW	SI	82	p n
LPSWE	D ₂ (B ₂)	Load PSW Extended	S	B2B2	p n N
LPTEA	R ₁ ,R ₃ ,R ₂ ,M ₄	Load Page-Table-Entry Address	RRF-b	B9AA	p c D2
LPXBR	R ₁ ,R ₂	Load Positive (EB)	RRE	B340	
LPXR	R ₁ ,R ₂	Load Positive (EH)	RRE	B360	αс
LR	R ₁ ,R ₂	Load (32)	RR	18	
LRA	$R_1, D_2(X_2, B_2)$	Load Real Address (32)	RX-a	B1	рс
LRAG	R ₁ ,D ₂ (X ₂ ,B ₂)	Load Real Address (64)	RXY-a	E303	pcN
LRAY	R ₁ ,D ₂ (X ₂ ,B ₂)	Load Real Address (32)	RXY-a		pcLD
LRDR	R ₁ ,R ₂	Load Rounded (LH←EH)	RR	25	D D
LRER	R ₁ ,R ₂	Load Rounded (SH←LH)	RR	35	a
LRL	R ₁ ,RI ₂	Load Relative Long (32)	RIL-b	C4D	GE
LRV	R ₁ ,D ₂ (X ₂ ,B ₂)	Load Reversed (32)		E31E	
LRVG	$R_1,D_2(X_2,B_2)$	Load Reversed (64)		E30F	
LRVGR	R ₁ ,R ₂	Load Reversed (64)	RRE	B90F	
LRVH	R ₁ ,D ₂ (X ₂ ,B ₂)	Load Reversed (04)		E31F	
LRVR		Load Reversed (32)	RRE	B91F	
	R ₁ ,R ₂	. ,			
LTDRD	R ₁ ,D ₂ (X ₂ ,B ₂)	Load and Test (32)	RRE	E312	
LTDBR	R ₁ ,R ₂	Load and Test (LB)		B312	
LTDTD	R ₁ ,R ₂	Load and Test (LH)	RR	22	¤ C
LTDTR	R ₁ ,R ₂	Load and Test (CD)	RRE		¤ c TF
LTEBR	R ₁ ,R ₂	Load and Test (SB)	RRE	B302	
LTER	R ₁ ,R ₂	Load and Test (SH)	RR	32	α C
LTG	$R_1,D_2(X_2,B_2)$	Load and Test (64)	RXY-a	E302	c El

					Class
Mne- monic	Operands	Name	For- mat	Op- code	& Notes
LTGF	R ₁ ,D ₂ (X ₂ ,B ₂)	Load And Test (64←32)	RXY-a		c GE
LTGFR	R ₁ ,R ₂	Load and Test (64←32)	RRE	B912	c N
LTGR	R ₁ ,R ₂	Load and Test (64)	RRE	B902	c N
LTR	R ₁ ,R ₂	Load and Test (32)	RR	12	С
LTXBR	R ₁ ,R ₂	Load and Test (EB)	RRE	B342	
LTXR	R ₁ ,R ₂	Load and Test (EH)	RRE	B362	αс
LTXTR	R ₁ ,R ₂	Load and Test (ED)	RRE	B3DE	¤ c TF
LURA	R ₁ ,R ₂	Load Using Real Address (32)	RRE	B24B	p
LURAG	R ₁ ,R ₂	Load Using Real Address (64)	RRE	B905	pΝ
LXD	$R_1, D_2(X_2, B_2)$	Load Lengthened (EH←LH)	RXE	ED25	۵
LXDB	$R_1,D_2(X_2,B_2)$	Load Lengthened (EB←LB)	RXE	ED05	۵
LXDBR	R ₁ ,R ₂	Load Lengthened (EB←LB)	RRE	B305	۵
LXDR	R ₁ ,R ₂	Load Lengthened (EH←LH)	RRE	B325	۵
LXDTR	R_1, R_2, M_4	Load Lengthened (ED←LD)	RRF-d	B3DC	¤ TF
LXE	$R_1,D_2(X_2,B_2)$	Load Lengthened (EH←SH)	RXE	ED26	۵
LXEB	$R_1,D_2(X_2,B_2)$	Load Lengthened (EB←SB)	RXE	ED06	۵
LXEBR	R ₁ ,R ₂	Load Lengthened (EB←SB)	RRE	B306	۵
LXER	R ₁ ,R ₂	Load Lengthened (EH←SH)	RRE	B326	۵
LXR	R ₁ ,R ₂	Load (E)	RRE	B365	۵
LY	$R_1,D_2(X_2,B_2)$	Load (32)	RXY-a	E358	LD
LZDR	R ₁	Load Zero (L)	RRE	B375	۵
LZER	R ₁	Load Zero (S)	RRE	B374	۵
LZRF	$R_1,D_2(X_2,B_2)$	Load and Zero Rightmost Byte (32)	RXY-a	E33B	LZ
LZRG	$R_1,D_2(X_2,B_2)$	Load and Zero Rightmost Byte (64)	RXY-a	E32A	LZ
LZXR	R ₁	Load Zero (E)	RRE	B376	۵
M	$R_1,D_2(X_2,B_2)$	Multiply (64←32)	RX-a	5C	
MAD	$R_1, R_3, D_2(X_2, B_2)$	Multiply and Add (LH)	RXF	ED3E	¤ HM
MADB	$R_1, R_3, D_2(X_2, B_2)$	Multiply and Add (LB)	RXF	ED1E	۵
MADBR	R_1,R_3,R_2	Multiply and Add (LB)	RRD	B31E	۵
MADR	R_1,R_3,R_2	Multiply and Add (LH)	RRD	B33E	¤ HM
MAE	$R_1, R_3, D_2(X_2, B_2)$	Multiply and Add (SH)	RXF	ED2E	¤ HM
MAEB	$R_1, R_3, D_2(X_2, B_2)$	Multiply and Add (SB)	RXF	ED0E	
MAEBR	R_1,R_3,R_2	Multiply and Add (SB)	RRD	B30E	
MAER	R ₁ ,R ₃ ,R ₂	Multiply and Add (SH)	RRD	B32E	
MAY	R ₁ ,R ₃ ,D ₂ (X ₂ ,B ₂)	Multiply and Add Unnormalized (EH←LH)	RXF	ED3A	
MAYH	R ₁ ,R ₃ ,D ₂ (X ₂ ,B ₂)	Multiply and Add Unnormalized (EH _H ←LH)		ED3C	
MAYHR	R ₁ ,R ₃ ,R ₂	Multiply and Add Unnormalized (EH _H ←LH)		B33C	
MAYL	R ₁ ,R ₃ ,D ₂ (X ₂ ,B ₂)	Multiply and Add Unnormalized (EH _L ←LH)		ED38	
MAYLR	R ₁ ,R ₃ ,R ₂	Multiply and Add Unnormalized (EH _L ←LH)		B338	¤ UE
MAYR	R ₁ ,R ₃ ,R ₂	Multiply and Add Unnormalized (EH←LH)	RRD	B33A	
MC	D ₁ (B ₁),l ₂	Monitor Call	SI	AF	0
MD	R ₁ ,D ₂ (X ₂ ,B ₂)	Multiply (LH)	RX-a	6C	0
MDB	R ₁ ,D ₂ (X ₂ ,B ₂)	Multiply (LB)	RXE	ED1C	
MDBR	R ₁ ,R ₂	Multiply (LB)	RRE	B31C	0
MDE	R ₁ ,D ₂ (X ₂ ,B ₂)	Multiply (LH←SH)	RX-a	7C	0
MDEB	R ₁ ,D ₂ (X ₂ ,B ₂)	Multiply (LB←SB)	RXE	ED0C B30C	
MDEBR MDER	R ₁ ,R ₂	Multiply (LB←SB)	RRE RR	3C	<u>а</u>
MDR	R ₁ ,R ₂	Multiply (LH←SH) Multiply (LH)	RR	2C	D D
	R ₁ ,R ₂	1, 1, 1	RRF-a		
MDTR MDTRA	R ₁ ,R ₂ ,R ₃ R ₁ ,R ₂ ,R ₃ ,M ₄	Multiply (LD) Multiply (LD)	RRF-a		
ME	R ₁ ,D ₂ (X ₂ ,B ₂)	Multiply (LH←SH)	RX-a	7C	۵
MEE		Multiply (SH)	RXE	ED37	
MEEB	$R_1,D_2(X_2,B_2)$ $R_1,D_2(X_2,B_2)$	Multiply (SB)	RXE	ED37	
MEEBR	R_1, R_2	Multiply (SB)	RRE	B317	۵
MEER	R ₁ ,R ₂	Multiply (SH)	RRE	B337	۵
MER	R ₁ ,R ₂	Multiply (LH←SH)	RR	3C	۵
MFY	$R_1, D_2(X_2, B_2)$	Multiply (64←32)		E35C	
MG	$R_1,D_2(X_2,B_2)$ $R_1,D_2(X_2,B_2)$	Multiply (128←64)	RXY-a		MI2
MGH	$R_1,D_2(X_2,B_2)$ $R_1,D_2(X_2,B_2)$	Multiply Halfword(64←16)	RXY-a		MI2
	,	manaprij i idii mordi(o-r · 10)	. 17.1-a	2000	.4112

Mne- monic	Operands	Name	For- mat	Op- code	Class & Notes
MGHI	R ₁ ,l ₂	Multiply Halfword Immediate (64←16)	RI-a	A7D	N
MGRK	R ₁ ,R ₂ ,R ₃	Multiply (128←64)	RRF-a	B9EC	MI2
MH	R ₁ ,D ₂ (X ₂ ,B ₂)	Multiply Halfword (32←16)	RX-a	4C	
MHI	R ₁ ,l ₂	Multiply Halfword Immediate (32←16)	RI-a	A7C	
MHY	$R_1,D_2(X_2,B_2)$	Multiply Halfword (64←16)	RXY-a		GE
ML	R ₁ ,D ₂ (X ₂ ,B ₂)	Multiply Logical (64←32)	RXY-a	E396	N3
MLG	$R_1, D_2(X_2, B_2)$	Multiply Logical (128←64)	RXY-a		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	$D_1(L_1,B_1),D_2(L_2,B_2)$		SS-b	FC	а
MR	R ₁ ,R ₂	Multiply (64←32)	RR	1C	
MS	$R_1, D_2(X_2, B_2)$	Multiply Single (32)	RX-a	71	
MSC	$R_1,D_2(X_2,B_2)$	Multiply Single (32)	RXY-a		c MI2
MSCH	D ₂ (B ₂)	Modify Subchannel	S	B232	рс
MSD	$R_1, R_3, D_2(X_2, B_2)$	Multiply and Subtract (LH)	RXF	ED3F	
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)	RRD	B31F	
MSDR	R ₁ ,R ₃ ,R ₂	Multiply and Subtract (LH)	RRD	B33F	
MSE	R_1, R_3, R_2 $R_1, R_3, D_2(X_2, B_2)$	Multiply and Subtract (SH)	RXF	ED2F	
MSEB	$R_1, R_3, D_2(X_2, B_2)$ $R_1, R_3, D_2(X_2, B_2)$	Multiply and Subtract (SB)	RXF	ED0F	
MSEBR		Multiply and Subtract (SB)	RRD	B30F	
MSER	R ₁ ,R ₃ ,R ₂ R ₁ ,R ₃ ,R ₂	Multiply and Subtract (SH)	RRD	B32F	
MSFI	n ₁ ,n ₃ ,n ₂ R ₁ ,l ₂	Multiply Single Immediate (32)	RIL-a		GE
MSG	1. 2		RXY-a		
	$R_1, D_2(X_2, B_2)$	Multiply Single (64)	RXY-a		
MSGC	$R_1, D_2(X_2, B_2)$	Multiply Single (64)			
MSGF	R ₁ ,D ₂ (X ₂ ,B ₂)	Multiply Single (64←32)	RXY-a		
MSGFI MSGFR	R ₁ ,l ₂	Multiply Single Immediate (64←32)	RIL-a		GE
	R ₁ ,R ₂	Multiply Single (64←32)	RRE	B91C	
MSGR	R ₁ ,R ₂	Multiply Single (64)	RRE	B90C	
MSGRKC		Multiply Single (64)	RRF-a		C IVII2
MSR	R ₁ ,R ₂	Multiply Single (32)	RRE	B252	- 1410
MSRKC	R ₁ ,R ₂ ,R ₃	Multiply Single (32)	RRF-a RRE	B247	C IVII2
MSTA	R ₁	Modify Stacked State			
MSY	R ₁ ,D ₂ (X ₂ ,B ₂)	Multiply Single (32)	RXY-a SS-a	E351	LD ¤
MVCDK	D ₁ (L,B ₁),D ₂ (B ₂)	Move (character)			
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-a	E8	۵
MVCK	$D_1(R_1,B_1),D_2(B_2),R_3$		SS-d	D9	q c
MVCL	R ₁ ,R ₂	Move Long	RR	0E	i¤c
MVCLE	R ₁ ,R ₃ ,D ₂ (B ₂)	Move Long Extended	RS-a	A8	¤ C
MVCLU	R ₁ ,R ₃ ,D ₂ (B ₂)	Move Long Unicode	RSY-a		
MVCOS	D ₁ (B ₁),D ₂ (B ₂),R ₃	Move with Optional Specifications	SSF	C80	q c M
MVCP	D ₁ (R ₁ ,B ₁),D ₂ (B ₂),R ₃		SS-d	DA	q c
MVCS		Move to Secondary	SS-d	DB	q c
MVCSK	D ₁ (B ₁),D ₂ (B ₂)	Move with Source Key	SSE	E50E	
MVGHI	D ₁ (B ₁),l ₂	Move (64←16)	SIL	E548	GE
MVHHI	D ₁ (B ₁),l ₂	Move (16←16)	SIL	E544	GE
MVHI	D ₁ (B ₁),l ₂	Move (32←16)	SIL	E54C	GE
MVI	D ₁ (B ₁),l ₂	Move Immediate	SI	92	
MVIY	D ₁ (B ₁),l ₂	Move Immediate	SIY	EB52	
MVN	D ₁ (L,B ₁),D ₂ (B ₂)	Move Numerics	SS-a	D1	۵
MVO	$D_1(L_1,B_1),D_2(L_2,B_2)$		SS-b	F1	۵
MVPG	R ₁ ,R ₂	Move Page	RRE	B254	q c
MVST	R ₁ ,R ₂	Move String	RRE	B255	шС
MVZ	$D_1(L,B_1),D_2(B_2)$	Move Zones	SS-a	D3	۵
MXBR	R_1,R_2	Multiply (EB)	RRE	B34C	
MXD	$R_1, D_2(X_2, B_2)$	Multiply (EH←LH)	RX-a	67	۵
MXDB	$R_1, D_2(X_2, B_2)$	Multiply (EB←LB)	RXE	ED07	۵
MXDBR	R ₁ ,R ₂	Multiply (EB←LB)	RRE	B307	۵
MXDR	R_1,R_2	Multiply (EH←LH)	RR	27	ø

Mne- monic	Operands	Name	For- mat	Op- code	Class & Notes
MXR	R ₁ ,R ₂	Multiply (EH)	RR	26	۵
MXTR	R ₁ ,R ₂ ,R ₃	Multiply (ED)	RRF-a	B3D8	¤ TF
MXTRA	R ₁ ,R ₂ ,R ₃ ,M ₄	Multiply (ED)	RRF-a		
MY	$R_1, R_3, D_2(X_2, B_2)$	Multiply Unnormalized (EH←LH)	RXF	ED3B	
MYH	$R_1, R_3, D_2(X_2, B_2)$	Multiply Unnormalized (EH _H ←LH)	RXF	ED3D	
MYHR	R ₁ ,R ₃ ,R ₂	Multiply Unnormalized (EH _H ←LH)	RRD	B33D	
MYL		Multiply Unnormalized (EH _I ←LH)	RXF	ED39	
MYLR	R ₁ ,R ₃ ,D ₂ (X ₂ ,B ₂)		RRD	B339	
MYR	R ₁ ,R ₃ ,R ₂	Multiply Unnormalized (EH _L ←LH)			
	R ₁ ,R ₃ ,R ₂	Multiply Unnormalized (EH←LH)	RRD	B33B	
N	$R_1,D_2(X_2,B_2)$	AND (32)	RX-a	54	С
NC	$D_1(L,B_1),D_2(B_2)$	AND (character)	SS-a	D4	шС
NG	$R_1, D_2(X_2, B_2)$	AND (64)	RXY-a		c N
NGR	R ₁ ,R ₂	AND (64)	RRE	B980	сN
NGRK	R ₁ ,R ₂ ,R ₃	AND (64)	RRF-a	B9E4	c DO
NI	$D_1(B_1),I_2$	AND Immediate	SI	94	С
NIAI	l ₁ ,l ₂	Next Instruction Access Intent	ΙE	B2FA	EH
NIHF	R ₁ ,I ₂	AND Immediate (high)	RIL-a	C0A	c El
NIHH	R ₁ ,I ₂	AND Immediate (high high)	RI-a	A54	сN
NIHL	R ₁ ,l ₂	AND Immediate (high low)	RI-a	A55	сN
NILF	R ₁ ,l ₂	AND Immediate (low)	RIL-a		c El
NILH	R ₁ ,l ₂	AND Immediate (low high)	RI-a	A56	сN
NILL	R ₁ ,l ₂	AND Immediate (low low)	RI-a	A57	c N
NIY	D ₁ (B ₁),I ₂	AND Immediate	SIY	EB54	
NR			RR	14	
	R ₁ ,R ₂	AND (32)			C . DO
NRK	R ₁ ,R ₂ ,R ₃	AND (32)	RRF-a		
NTSTG	$R_1,D_2(X_2,B_2)$	Nontransactional Store (64)	RXY-a		
NY	$R_1, D_2(X_2, B_2)$	AND (32)	RXY-a		c LD
0	$R_1,D_2(X_2,B_2)$	OR (32)	RX-a	56	С
OC	$D_1(L,B_1),D_2(B_2)$	OR (character)	SS-a	D6	ΩC
OG	$R_1,D_2(X_2,B_2)$	OR (64)	RXY-a	E381	сN
OGR	R ₁ ,R ₂	OR (64)	RRE	B981	сN
OGRK	R ₁ ,R ₂ ,R ₃	OR (64)	RRF-a	B9E6	c DO
OI	$D_1(B_1),I_2$	OR Immediate	SI	96	С
OIHF	R_1, I_2	OR Immediate (high)	RIL-a	COC	c El
OIHH	R ₁ ,l ₂	OR Immediate (high high)	RI-a	A58	сN
OIHL	R ₁ ,l ₂	OR Immediate (high low)	RI-a	A59	сN
OILF	R ₁ ,l ₂	OR Immediate (low)	RIL-a		c El
OILH	R ₁ ,I ₂	OR Immediate (low high)	RI-a	A5A	c N
OILL		OR Immediate (low low)	RI-a	A5B	c N
OILL	R ₁ ,l ₂	, ,	SIY	EB56	
	D ₁ (B ₁),l ₂	OR Immediate			
OR	R ₁ ,R ₂	OR (32)	RR	16	C
ORK	R ₁ ,R ₂ ,R ₃	OR (32)	RRF-a		
OY	$R_1, D_2(X_2, B_2)$	OR (32)	RXY-a		c LD
PACK	$D_1(L_1,B_1),D_2(L_2,B_2)$		SS-b	F2	۵
PALB		Purge ALB	RRE	B248	p
PC	$D_2(B_2)$	Program Call	S	B218	q
PCC		Perform Cryptographic Computation	RRE	B92C	
PCKMO		Perform Crypto. Key Mgmt. Operations	RRE	B928	M3
PFD	$M_1,D_2(X_2,B_2)$	Prefetch Data	RXY-b	E336	¤ GE
PFDRL	M ₁ ,RI ₂	Prefetch Data Relative Long	RIL-c	C62	
PFMF	R ₁ ,R ₂	Perform Frame Management Function	RRE	B9AF	p ED1
PFPO		Perform Floating-Point Operation	E	010A	¤PF
PGIN	R ₁ ,R ₂	Page In	RRE	B22E	p c ES
PGOUT	R ₁ ,R ₂	Page Out	RRE	B22F	p c ES
PKA	D ₁ (B ₁),D ₂ (L ₂ ,B ₂)	Pack ASCII	SS-f	E9	¤ E2
PKU	D ₁ (B ₁),D ₂ (L ₂ ,B ₂)	Pack Unicode	SS-f	E1	¤ E2
PLO		Perform Locked Operation	SS-e	EE	пС
POPCNT		Population Count	RRE	B9E1	
		· operation count	100		0111
PPA	R ₁ ,R ₂ ,M ₃	Perform Processor Assist	RRF-c	R2E9	PA

Mne-			For-	Op-	Clas &
monic	Operands	Name	mat	code	Note
PRNO	R ₁ ,R ₂	Perform Random Number Operation	RRE	B93C	M5
PT	R ₁ ,R ₂	Program Transfer	RRE	B228	q
PTF	R ₁	Perform Topology Function	RRE	B9A2	c p C
PTFF		Perform Timing-Facility Function	Е	0104	q c
PTI	R ₁ ,R ₂	Program Transfer with Instance	RRE	B99E	
PTLB		Purge TLB	S	B20D	
QADTR	R_1, R_3, R_2, M_4	Quantize (LD)	RRF-b		
QAXTR	R_1, R_3, R_2, M_4	Quantize (ED)	RRF-b	B3FD	¤ TF
RCHP		Reset Channel Path	S	B23B	
RISBG	$R_1, R_2, I_3, I_4[, I_5]$	Rotate then Insert Selected Bits (64)	RIE-f		
RISBGN	$R_1, R_2, I_3, I_4[, I_5]$	Rotate then Insert Selected Bits (64)	RIE-f	EC59	MI1
RISBHG	$R_1,I_2,I_3,I_4[,I_5]$	Rotate then Insert Selected Bits High (32)	RIE-f	EC5D	HW
RISBLG	$R_1, I_2, I_3, I_4[, I_5]$	Rotate then Insert Selected Bits Low (32)	RIE-f	EC51	HW
RLL	$R_1, R_3, D_2(B_2)$	Rotate Left Single Logical (32)	RSY-a	EB1D	N3
RLLG	$R_1, R_3, D_2(B_2)$	Rotate Left Single Logical (64)	RSY-a	EB1C	N
RNSBG	R ₁ ,R ₂ ,I ₃ ,I ₄ [,I ₅]	Rotate then AND Selected Bits (64)	RIE-f	EC54	c GE
ROSBG	R ₁ ,R ₂ ,I ₃ ,I ₄ [,I ₅]	Rotate then OR Selected Bits (64)	RIE-f		
RP	D ₂ (B ₂)	Resume Program	S	B277	
RRBE	R ₁ ,R ₂	Reset Reference Bit Extended	RRE	B22A	
RRBM	R ₁ ,R ₂	Reset Reference Bits Multiple	RRE	B9AE	
RRDTR	R ₁ ,R ₃ ,R ₂ ,M ₄	Reround (LD)	RRF-b		
RRXTR		Reround (ED)		B3FF	
RSCH	R_1,R_3,R_2,M_4	Resume Subchannel	S		
RXSBG	$R_1, R_2, I_3, I_4[, I_5]$	Rotate then Exclusive OR Selected Bits (64)	RIE-f	B238 EC57	
S	$R_1,D_2(X_2,B_2)$	Subtract (32)	RX-a	5B	С
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	2(2)	Set Address Limit	S	B237	p
SAM24		Set Addressing Mode (24)	Ē	010C	
SAM31		Set Addressing Mode (31)	Е	010D	
SAM64		Set Addressing Mode (64)	Е	010E	¤Ν
SAR	R_1,R_2	Set Access	RRE	B24E	۵
SCHM		Set Channel Monitor	S	B23C	р
SCK	$D_2(B_2)$	Set Clock	S	B204	рс
SCKC	D ₂ (B ₂)	Set Clock Comparator	S	B206	p
SCKPF		Set Clock Programmable Field	Е	0107	p
SD	$R_1,D_2(X_2,B_2)$	Subtract Normalized (LH)	RX-a	6B	пc
SDB	$R_1,D_2(X_2,B_2)$	Subtract (LB)	RXE	ED1B	αс
SDBR	R ₁ ,R ₂	Subtract (LB)	RRE	B31B	αс
SDR	R ₁ ,R ₂	Subtract Normalized (LH)	RR	2B	αс
SDTR	R ₁ ,R ₂ ,R ₃	Subtract (LD)	RRF-a	B3D3	¤ c T
SDTRA	R ₁ ,R ₂ ,R ₃ ,M ₄	Subtract (LD)	RRF-a		
SE	$R_1, D_2(X_2, B_2)$	Subtract Normalized (SH)	RX-a		пς
SEB	$R_1,D_2(X_2,B_2)$	Subtract (SB)	RXE	ED0B	
SEBR		, ,	RRE	B30B	
	R ₁ ,R ₂	Subtract Normalized (SU)			
SER	R ₁ ,R ₂	Subtract Normalized (SH)	RR	3B	¤ C
SFASR	R ₁	Set FPC and Signal	RRE	B385	
SFPC	R ₁	Set FPC	RRE	B384	α
SG	$R_1, D_2(X_2, B_2)$	Subtract (64)	RXY-a		
SGF	$R_1, D_2(X_2, B_2)$	Subtract (64←32)	RXY-a	E319	сN
SGFR	R ₁ ,R ₂	Subtract (64←32)	RRE		
SGH	$R_1, D_2(X_2, B_2)$	Subtract Halfword (64←16)	RXY-a	E339	c MI2
SGR	R ₁ ,R ₂	Subtract (64)	RRE	B909	сN
SGRK	R ₁ ,R ₂ ,R ₃	Subtract (64)	RRF-a	B9E9	c DO
SH	R ₁ ,D ₂ (X ₂ ,B ₂)	Subtract Halfword (32←16)	RX-a		С
SHHHR	R ₁ ,R ₂ ,R ₃	Subtract High (32)		B9C9	
SHHLR	R ₁ ,R ₂ ,R ₃	Subtract High (32)		B9D9	
SHY	R ₁ ,D ₂ (X ₂ ,B ₂)	Subtract Halfword (32←16)		E37B	
SIE	$D_2(B_2)$	Start Interpretive Execution	S	B214	
	-21-21	Signal Processor	RS-a	AE	٠,٢

Mne-			For-	Op-	Class &
monic	Operands	Name	mat	code	Notes
SL	$R_1,D_2(X_2,B_2)$	Subtract Logical (32)	RX-a	5F	С
SLA	R ₁ ,D ₂ (B ₂)	Shift Left Single (32)	RS-a	8B	C
SLAG	R ₁ ,R ₃ ,D ₂ (B ₂)	Shift Left Single (64)	RSY-a		
SLAK	R ₁ ,R ₃ ,D ₂ (B ₂)	Shift Left Single (32)	RSY-a		
SLB	$R_1, D_2(X_2, B_2)$	Subtract Logical with Borrow (32)	RXY-a		
SLBG SLBGR	R ₁ ,D ₂ (X ₂ ,B ₂)	Subtract Logical with Borrow (64)	RXY-a RRE		c N c N
SLBR	R ₁ ,R ₂ R ₁ ,R ₂	Subtract Logical with Borrow (64) Subtract Logical with Borrow (32)	RRE	B989 B999	c N3
SLDA	R ₁ ,D ₂ (B ₂)	Shift Left Double (64)	RS-a	8F	C
SLDL	R ₁ ,D ₂ (B ₂)	Shift Left Double Logical (64)	RS-a	8D	•
SLDT	R ₁ ,R ₃ ,D ₂ (X ₂ ,B ₂)	Shift Significand Left (LD)	RXF	ED40	¤ TF
SLFI	R ₁ ,I ₂	Subtract Logical Immediate (32)	RIL-a		c El
SLG	$R_1, D_2(X_2, B_2)$	Subtract Logical (64)	RXY-a	E30B	
SLGF	$R_1, D_2(X_2, B_2)$	Subtract Logical (64←32)	RXY-a	E31B	c N
SLGFI	R ₁ ,l ₂	Subtract Logical Immediate (64←32)	RIL-a	C24	c El
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
SLGRK	R ₁ ,R ₂ ,R ₃	Subtract Logical (64)	RRF-a	B9EB	c DO
SLHHHR	R ₁ ,R ₂ ,R ₃	Subtract Logical High (32)	RRF-a	B9CB	c HW
SLHHLR	R_1,R_2,R_3	Subtract Logical High (32)	RRF-a	B9DB	c HW
SLL	$R_1,D_2(B_2)$	Shift Left Single Logical (32)	RS-a	89	
SLLG	$R_1, R_3, D_2(B_2)$	Shift Left Single Logical (64)	RSY-a	EB0D	N
SLLK	$R_1, R_3, D_2(B_2)$	Shift Left Single Logical (32)	RSY-a		DO
SLR	R ₁ ,R ₂	Subtract Logical (32)	RR	1F	C
SLRK	R_1, R_2, R_3	Subtract Logical (32)	RRF-a		
SLXT	R ₁ ,R ₃ ,D ₂ (X ₂ ,B ₂)	Shift Significand Left (ED)	RXF	ED48	
SLY	$R_1,D_2(X_2,B_2)$	Subtract Logical (32)	RXY-a		c LD
SP	$D_1(L_1,B_1),D_2(L_2,B_2)$		SS-b	FB	шС
SPKA	D ₂ (B ₂)	Set PSW Key from Address	S	B20A	
SPM	R ₁	Set Program Mask	RR	04	n
SPT SPX	D ₂ (B ₂)	Set CPU Timer Set Prefix	S S	B208 B210	p
SQD	$D_2(B_2)$ $R_1, D_2(X_2, B_2)$	Square Root (LH)	RXE		p ¤
SQDB	$R_1,D_2(X_2,B_2)$ $R_1,D_2(X_2,B_2)$	Square Root (LB)	RXE	ED15	
SQDBR	R ₁ ,R ₂	Square Root (LB)	RRE	B315	۵
SQDR	R ₁ ,R ₂	Square Root (LH)	RRE	B244	۵
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	۵
SQER	R ₁ ,R ₂	Square Root (SH)	RRE	B245	۵
SQXBR	R ₁ ,R ₂	Square Root (EB)	RRE	B316	۵
SQXR	R ₁ ,R ₂	Square Root (EH)	RRE	B336	۵
SR	R_1,R_2	Subtract (32)	RR	1B	С
SRA	$R_1,D_2(B_2)$	Shift Right Single (32)	RS-a	8A	С
SRAG	$R_1, R_3, D_2(B_2)$	Shift Right Single (64)	RSY-a	EB0A	c N
SRAK	$R_1, R_3, D_2(B_2)$	Shift Right Single (32)	RSY-a	EBDC	c DO
SRDA	$R_1,D_2(B_2)$	Shift Right Double (64)	RS-a	8E	С
SRDL	$R_1,D_2(B_2)$	Shift Right Double Logical (64)	RS-a	8C	
SRDT	$R_1, R_3, D_2(X_2, B_2)$	Shift Significand Right (LD)	RXF	ED41	¤ TF
SRK	R ₁ ,R ₂ ,R ₃	Subtract (32)		B9F9	c DO
SRL	$R_1, D_2(B_2)$	Shift Right Single Logical (32)	RS-a		
SRLG	$R_1, R_3, D_2(B_2)$	Shift Right Single Logical (64)		EB0C	
SRLK	R ₁ ,R ₃ ,D ₂ (B ₂)	Shift Right Single Logical (32)		EBDE	
SRNM	D ₂ (B ₂)	Set BFP Rounding Mode (2 bit)	S	B299	
SRNMB	D ₂ (B ₂)	Set BFP Rounding Mode (3 bit)	S	B2B8	
SRNMT	D ₂ (B ₂)	Set DFP Rounding Mode	S	B2B9	
SRP		Shift and Round Decimal	SS-c	F0	шС
SRST	R ₁ ,R ₂	Search String	RRE	B25E	
SRSTU	R ₁ ,R ₂	Search String Unicode	RRE		¤ c E3
SRXT	$R_1, R_3, D_2(X_2, B_2)$	Shift Significand Right (ED)	RXF	ED49	u IF

Mne-			For-	Ор-	Class &	
monic	Operands	Name	mat	code	Note	
SSAIR	R ₁	Set Secondary ASN with Instance	RRE	B99F	¤ RA	
SSAR	R ₁	Set Secondary ASN	RRE	B225	۵	
SSCH	D ₂ (B ₂)	Start Subchannel	S	B233	рс	
SSKE	R ₁ ,R ₂ [,M ₃]	Set Storage Key Extended	RRF-d	B22B	рс	
SSM	D ₂ (B ₂)	Set System Mask	SI	80	p	
ST	$R_1,D_2(X_2,B_2)$	Store (32)	RX-a	50		
STAM	$R_1, R_3, D_2(B_2)$	Store Access Multiple	RS-a	9B		
STAMY	$R_1, R_3, D_2(B_2)$	Store Access Multiple	RSY-a	EB9B	LD	
STAP	D ₂ (B ₂)	Store CPU Address	S	B212	p	
STC	$R_1,D_2(X_2,B_2)$	Store Character	RX-a	42		
STCH	$R_1, D_2(X_2, B_2)$	Store Character High (8)	RXY-a	E3C3	HW	
STCK	D ₂ (B ₂)	Store Clock	S	B205	αс	
STCKC	D ₂ (B ₂)	Store Clock Comparator	S	B207	p	
STCKE	D ₂ (B ₂)	Store Clock Extended	S	B278	пc	
STCKF	D ₂ (B ₂)	Store Clock Fast	S	B27C	¤ c S	
STCM	$R_1,M_3,D_2(B_2)$	Store Characters under Mask (low)	RS-b	BE		
STCMH	$R_1,M_3,D_2(B_2)$	Store Characters under Mask (high)	RSY-b	EB2C	пΝ	
STCMY	$R_1,M_3,D_2(B_2)$	Store Characters under Mask (low)	RSY-b	EB2D	LD	
STCPS	D ₂ (B ₂)	Store Channel Path Status	S	B23A	p	
STCRW	D ₂ (B ₂)	Store Channel Report Word	S	B239	рс	
STCTG	$R_1, R_3, D_2(B_2)$	Store Control (64)	RSY-a	EB25	pΝ	
STCTL	$R_1, R_3, D_2(B_2)$	Store Control (32)	RS-a	B6	p	
STCY	$R_1,D_2(X_2,B_2)$	Store Character	RXY-a	E372	LD	
STD	$R_1,D_2(X_2,B_2)$	Store (L)	RX-a	60	۵	
STDY	$R_1,D_2(X_2,B_2)$	Store (L)	RXY-a	ED67	¤ LD	
STE	$R_1,D_2(X_2,B_2)$	Store (S)	RX-a	70	۵	
STEY	$R_1,D_2(X_2,B_2)$	Store (S)	RXY-a	ED66	¤ LD	
STFH	$R_1,D_2(X_2,B_2)$	Store High (32)	RXY-a	E3CB	HW	
STFL	D ₂ (B ₂)	Store Facility List	S	B2B1	pN3	
STFLE	D ₂ (B ₂)	Store Facility List Extended	S	B2B0	¤сF	
STFPC	D ₂ (B ₂)	Store FPC	S	B29C	۵	
STG	$R_1,D_2(X_2,B_2)$	Store (64)	RXY-a	E324	N	
STGRL	R ₁ ,RI ₂	Store Relative Long (64)	RIL-b	C4B	GE	
STGSC	$R_1,D_2(X_2,B_2)$	Store guarded storage controls	RXY-a	E349	¤ GF	
STH	$R_1,D_2(X_2,B_2)$	Store Halfword (16)	RX-a	40		
STHH	$R_1,D_2(X_2,B_2)$	Store Halfword High (16)	RXY-a	E3C7	HW	
STHRL	R ₁ ,RI ₂	Store Halfword Relative Long (16)	RIL-b	C47	GE	
STHY	$R_1, D_2(X_2, B_2)$	Store Halfword (16)	RXY-a	E370	LD	
STIDP	D ₂ (B ₂)	Store CPU ID	S	B202	p	
STM	$R_1, R_3, D_2(B_2)$	Store Multiple (32)	RS-a	90		
STMG	R ₁ ,R ₃ ,D ₂ (B ₂)	Store Multiple (64)	RSY-a	EB24	N	
STMH	R ₁ ,R ₃ ,D ₂ (B ₂)	Store Multiple High (32)	RSY-a	EB26	N	
STMY	R ₁ ,R ₃ ,D ₂ (B ₂)	Store Multiple (32)	RSY-a	EB90	LD	
STNSM	D ₁ (B ₁),I ₂	Store Then And System Mask	SI	AC	р	
STOC	$R_1,D_2(B_2),M_3$	Store on Condition (32)	RSY-b	EBF3	L1	
STOCFH	$R_1,D_2(B_2),M_3$	Store High on Condition (32)	RSY-b	EBE1	L2	
STOCG	$R_1,D_2(B_2),M_3$	Store on Condition (64)	RSY-b	EBE3	L1	
STOSM	D ₁ (B ₁),l ₂	Store Then Or System Mask	SI	AD	р	
STPQ	$R_1,D_2(X_2,B_2)$	Store Pair to Quadword (64,64→128)	RXY-a	E38E	¤Ν	
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 (64)	SSE	E502	pΝ	
STRL	R ₁ ,RI ₂	Store Relative Long (32)	RIL-b		GE	
STRV	R ₁ ,D ₂ (X ₂ ,B ₂)	Store Reversed (32)	RXY-a			
STRVG	$R_1,D_2(X_2,B_2)$	Store Reversed (64)	RXY-a		N	
STRVH	$R_1,D_2(X_2,B_2)$	Store Reversed (16)	RXY-a			
STSCH	$D_2(B_2)$	Store Subchannel	S	B234	рс	
STSI	$D_2(B_2)$ $D_2(B_2)$	Store System Information	S	B27D		
STURA	R ₁ ,R ₂	Store Using Real Address (32)	RRE	B246	р	
STURG	R ₁ ,R ₂	Store Using Real Address (64)	RRE	B925	рN	

Mne- monic	Operands	Name	For- mat	Op- code	Cla & Not
STY	R ₁ ,D ₂ (X ₂ ,B ₂)	Store (32)	RXY-a	E350	LD
SU	R ₁ ,D ₂ (X ₂ ,B ₂)	Subtract Unnormalized (SH)	RX-a	7F	αс
SUR	R ₁ ,R ₂	Subtract Unnormalized (SH)	RR	3F	αс
SVC	1, 2	Supervisor Call	1	0A	D .
SW	R ₁ ,D ₂ (X ₂ ,B ₂)	Subtract Unnormalized (LH)	RX-a	6F	αс
SWR	R ₁ ,R ₂	Subtract Unnormalized (LH)	RR	2F	αс
SXBR	R ₁ ,D ₂	Subtract (EB)	RRE	B34B	пc
SXR	R ₁ ,D ₂	Subtract Normalized (EH)	RR	37	пс
SXTR	R ₁ ,R ₂ ,R ₃	Subtract (ED)	RRF-a		
SXTRA	R ₁ ,R ₂ ,R ₃ ,M ₄	Subtract (ED)	RRF-a		
SY	$R_1, D_2(X_2, B_2)$	Subtract (32)	RXY-a		
TABORT		Transaction Abort	S	B2FC	
TAM	D ₂ (B ₂)		E	010B	
TAR	R ₁ ,R ₂	Test Addressing Mode Test Access	RRE	B24C	
TB	R ₁ ,R ₂	Test Block	RRE	B22C	
TBDR	R ₁ ,M ₃ ,R ₂	Convert HFP to BFP (LB←LH)	RRF-e		шC
TBEDR	R ₁ ,M ₃ ,R ₂	Convert HFP to BFP (SB←LH)		B350	
TBEGIN	$D_1(B_1), I_2$	Transaction Begin (nonconstrained)	SIL	E560	
TBEGINC		Transaction Begin (constrained)	SIL	E561	
TCDB	$R_1,D_2(X_2,B_2)$	Test Data Class (LB)	RXE	ED11	
TCEB	$R_1,D_2(X_2,B_2)$	Test Data Class (SB)	RXE	ED10	αС
TCXB	$R_1,D_2(X_2,B_2)$	Test Data Class (EB)	RXE	ED12	αС
TDCDT	$R_1,D_2(X_2,B_2)$	Test Data Class (LD)	RXE	ED54	¤ TI
TDCET	$R_1,D_2(X_2,B_2)$	Test Data Class (SD)	RXE	ED50	¤ TI
TDCXT	$R_1,D_2(X_2,B_2)$	Test Data Class (ED)	RXE	ED58	¤Τ
TDGDT	$R_1,D_2(X_2,B_2)$	Test Data Group (LD)	RXE	ED55	¤ TI
TDGET	$R_1,D_2(X_2,B_2)$	Test Data Group (SD)	RXE	ED51	¤ TI
TDGXT	R ₁ ,D ₂ (X ₂ ,B ₂)	Test Data Group (ED)	RXE	ED59	¤ TI
TEND	1, 2(2, 2)	Transaction End	S	B2F8	
THDER	R ₁ ,R ₂	Convert BFP to HFP (LH←SB)	RRE	B358	
THDR	R ₁ ,R ₂	Convert BFP to HFP (LH←LB)	RRE	B359	αс
TM	D ₁ (B ₁),l ₂	Test under Mask	SI	91	С
TMH	R ₁ ,I ₂	Test under Mask High	RI-a	A70	С
TMHH	R ₁ ,l ₂	Test under Mask (high high)	RI-a	A72	c N
TMHL		Test under Mask (high low)	RI-a	A73	c N
TML	R ₁ ,I ₂	Test under Mask (nightiow)	RI-a	A73	
	R ₁ ,I ₂				C
TMLH	R ₁ ,l ₂	Test under Mask (low high)	RI-a	A70	c N
TMLL	R ₁ ,l ₂	Test under Mask (low low)	RI-a	A71	c N
TMY	D ₁ (B ₁),l ₂	Test under Mask	SIY	EB51	
TP	$D_1(L_1,B_1)$	Test Decimal	RSL	EBC0	
TPEI	R ₁ ,R ₂	Test Pending External Interruption	RRE	B9A1	рс
TPI	$D_2(B_2)$	Test Pending Interruption	S	B236	рс
TPROT	$D_1(B_1), D_2(B_2)$	Test Protection	SSE	E501	рс
TR	$D_1(L,B_1),D_2(B_2)$	Translate	SS-a	DC	۵
TRACE	$R_1,R_3,D_2(B_2)$	Trace (32)	RS-a	99	p
TRACG	$R_1,R_3,D_2(B_2)$	Trace (64)	RSY-a	EB0F	pΝ
TRAP2		Trap	E	01FF	۵
TRAP4	D ₂ (B ₂)	Trap	S	B2FF	۵
TRE	R ₁ ,R ₂	Translate Extended	RRE	B2A5	αс
TROO	R ₁ ,R ₂ [,M ₃]	Translate One to One	RRF-c	B993	αс
TROT	R ₁ ,R ₂ [,M ₃]	Translate One to Two	RRF-c		
TRT	D ₁ (L,B ₁),D ₂ (B ₂)	Translate and Test	SS-a		αс
TRTE	R ₁ ,R ₂ [,M ₃]	Translate and Test Extended		B9BF	
TRTO	R ₁ ,R ₂ [,M ₃]	Translate Two to One	RRF-c		
TRTR	D ₁ (L,B ₁),D ₂ (B ₂)	Translate and Test Reverse	SS-a	D0	пC
TRTRE		Translate and Test Reverse Extended	RRF	B9BD	
	R ₁ ,R ₂ [,M ₃]		RRF-c		
TRTT	R ₁ ,R ₂ [,M ₃]	Translate Two to Two			
TS	D ₂ (B ₂)	Test and Set	SI	93	пC
TSCH	D ₂ (B ₂)	Test Subchannel	S	B235	рс
UNPK	$D_1(L_1,B_1),D_2(L_2,B_2)$	Unnack	SS-b	F3	Ø

Mne-			For-	Op-	Class &	
monic	Operands	Name	mat	code	Note	
UNPKA	D ₁ (L ₁ ,B ₁),D ₂ (B ₂)	Unpack ASCII	SS-a	EA	¤сЕ	
UNPKU	$D_1(L_1,B_1),D_2(B_2)$	Unpack Unicode	SS-a	E2	¤сЕ	
JPT		Update Tree	E	0102	i¤c	
/A	V_1, V_2, V_3, M_4	Vector Add	VRR-c	E7F3	¤ VF	
/AC	V_1, V_2, V_3, V_4, M_5	Vector Add With Carry	VRR-d	E7BB	¤ VF	
/ACC	V ₁ ,V ₂ ,V ₃ ,M ₄	Vector Add Compute Carry	VRR-c	E7F1	¤ VF	
/ACCC	V ₁ ,V ₂ ,V ₃ ,V ₄ ,M ₅	Vector Add With Carry Compute Carry	VRR-d	E7B9	¤ VF	
/AP	V ₁ ,V ₂ ,V ₃ ,I ₄ ,M ₅	Vector Add Decimal	VRI-f	E671	¤ c* \	
VAVG	V ₁ ,V ₂ ,V ₃ ,M ₄	Vector Average	VRR-c			
/AVGL	V ₁ ,V ₂ ,V ₃ ,M ₄	Vector Average Logical	VRR-c			
/BPERM		Vector Bit Permute	VRR-c			
VCDG	V ₁ ,V ₂ ,M ₃ ,M ₄ ,M ₅	Vector FP Convert from Fixed 64-bit	VRR-a			
VCDLG	V ₁ ,V ₂ ,M ₃ ,M ₄ ,M ₅	Vector FP Convert from Logical 64-bit	VRR-a			
VCEQ		Vector Compare Equal	VRR-b			
VCEQ	V ₁ ,V ₂ ,V ₃ ,M ₄ ,M ₅		VRR-a			
	V ₁ ,V ₂ ,M ₃ ,M ₄ ,M ₅	Vector FP Convert to Fixed 64-bit				
VCH	V ₁ ,V ₂ ,V ₃ ,M ₄ ,M ₅	Vector Compare High	VRR-b			
VCHL	V ₁ ,V ₂ ,V ₃ ,M ₄ ,M ₅	Vector Compare High Logical	VRR-b			
VCKSM	V ₁ ,V ₂ ,V ₃	Vector Checksum	VRR-c			
VCLGD	V_1, V_2, M_3, M_4, M_5	Vector FP Convert to Logical 64-bit	VRR-a			
VCLZ	V_1, V_2, M_3	Vector Count Leading Zeros	VRR-a			
VCP	V_1, V_2, M_3	Vector Compare Decimal	VRR-h			
VCTZ	V_1, V_2, M_3	Vector Count Trailing Zeros	VRR-a	E752	¤ VF	
VCVB	R_1,V_2,M_3	Vector Convert to Binary	VRR-i	E650	¤ C* /	
VCVBG	R_1, V_2, M_3	Vector Convert to Binary	VRR-i	E652	¤ C* \	
VCVD	V_1,R_2,I_3,M_4	Vector Convert to Decimal	VRI-i	E658	¤ C* \	
VCVDG	V_1,R_2,I_3,M_4	Vector Convert to Decimal	VRI-i	E65A	¤ c* \	
VDP	V ₁ ,V ₂ ,V ₃ ,I ₄ ,M ₅	Vector Divide Decimal	VRI-f	E67A	¤ c* \	
VEC	V_1, V_2, M_3	Vector Element Compare	VRR-a	E7DB	¤сV	
VECL	V ₁ ,V ₂ ,M ₃	Vector Element Compare Logical	VRR-a	E7D9	псV	
VERIM	V ₁ ,V ₂ ,V ₃ ,I ₄ ,M ₅	Vector Element Rotate and Insert Under Mask	VRI-d	E772	¤ VF	
VERLL	$V_1, V_3, D_2(B_2), M_4$	Vector Element Rotate Left Logical	VRS-a	E733	¤ VF	
VERLLV	V_1, V_2, V_3, M_4	Vector Element Rotate Left Logical	VRR-c	E773	¤ VF	
VESL	$V_1, V_3, D_2(B_2), M_4$	Vector Element Shift Left	VRS-a	E730	¤ VF	
VESLV	V_1, V_2, V_3, M_4	Vector Element Shift Left	VRR-c	E770	¤ VF	
VESRA	V ₁ ,V ₃ ,D ₂ (B ₂),M ₄	Vector Element Shift Right Arithmetic	VRS-a	E73A	¤ VF	
VESRAV	V ₁ ,V ₂ ,V ₃ ,M ₄	Vector Element Shift Right Arithmetic	VRR-c	E77A	¤ VF	
VESRL	V ₁ ,V ₃ ,D ₂ (B ₂),M ₄	Vector Element Shift Right Logical	VRS-a	E738	¤ VF	
VESRLV	V ₁ ,V ₂ ,V ₃ ,M ₄	Vector Element Shift Right Logical	VRR-c			
VFA	V ₁ ,V ₂ ,V ₃ ,M ₄ ,M ₅	Vector FP Add	VRR-c			
VFAE	V ₁ ,V ₂ ,V ₃ ,M ₄ [,M ₅]	Vector Find Any Element Equal	VRR-b			
VFCE	V ₁ ,V ₂ ,V ₃ ,M ₄ ,M ₅ ,M ₆	Vector FP Compare Equal	VRR-c			
VFCH		Vector FP Compare High	VRR-c			
VFCHE	V ₁ ,V ₂ ,V ₃ ,M ₄ ,M ₅ ,M ₆ V ₁ ,V ₂ ,V ₃ ,M ₄ ,M ₅ ,M ₆	, ,	VRR-c			
VFORE		Vector FP Compare High or Equal Vector FP Divide	VRR-c			
VFEE	V ₁ ,V ₂ ,V ₃ ,M ₄ ,M ₅		VRR-b			
	V ₁ ,V ₂ ,V ₃ ,M ₄ [,M ₅]	Vector Find Element Equal				
VFENE	V ₁ ,V ₂ ,V ₃ ,M ₄ [,M ₅]	Vector Find Element Not Equal	VRR-b			
VFI	V ₁ ,V ₂ ,M ₃ ,M ₄ ,M ₅	Vector Load FP Integer	VRR-a			
VFLL	V ₁ ,V ₂ ,M ₃ ,M ₄	Vector FP Load Lengthened	VRR-a			
VFLR	V_1, V_2, M_3, M_4, M_5	Vector FP Load Rounded	VRR-a			
VFM	V_1, V_2, V_3, M_4, M_5	Vector FP Multiply	VRR-c			
VFMA	$V_1, V_2, V_3, V_4, M_5, M_6$		VRR-e			
VFMAX	$V_1, V_2, V_3, M_4, M_5, M_6$	Vector FP Maximum	VRR-c			
VFMIN	$V_1, V_2, V_3, M_4, M_5, M_6$	Vector FP Minimum	VRR-c	E7EE	¤ V1	
VFMS	$V_1, V_2, V_3, V_4, M_5, M_6$	Vector FP Multiply and Subtract	VRR-e	E78E	¤۷F	
VFNMA		Vector FP Negative Multiply and Add	VRR-e	E79F	¤V1	
VFNMS		Vector FP Negative Multiply and Subtract	VRR-e			
VFPSO	V ₁ ,V ₂ ,M ₃ ,M ₄ ,M ₅	Vector FP Perform Sign Operation	VRR-a			
VFS	V ₁ ,V ₂ ,V ₃ ,M ₄ ,M ₅	Vector FP Subtract		E7E2		

Mne-			For-	Ор-	Class &
monic	Operands	Name	mat	code	Notes
VFTCI	V ₁ ,V ₂ ,I ₃ ,M ₄ ,M ₅	Vector FP Test Data Class Immediate	VRI-e	E74A	¤ VF
VGBM	V ₁ ,I ₂	Vector Generate Byte Mask	VRI-a		
VGEF	V ₁ ,D ₂ (V ₂ ,B ₂),M ₃	Vector Gather Element (32)	VRV	E713	
VGEG	V ₁ ,D ₂ (V ₂ ,B ₂),M ₃	Vector Gather Element (64)	VRV	E712	
VGFM	V ₁ ,V ₂ ,V ₃ ,M ₄	Vector Galois Field Multiply Sum	VRR-c		
VGFMA	V ₁ ,V ₂ ,V ₃ ,W ₄ ,M ₅	Vector Galois Field Multiply Sum and Accu-			
· OI IVII (*1,*2,*3,*4,**5	mulate	VIIII	2,00	- • • •
VGM	V_1, I_2, I_3, V_4	Vector Generate Mask	VRI-b	E746	¤VF
VISTR	V ₁ ,V ₂ ,M ₃ [,M ₅]	Vector Isolate String	VRR-a	E75C	□ c* VF
VL	$V_1,D_2(X_2,B_2)$	Vector Load	VRX	E706	¤VF
VLBB	V ₁ ,D ₂ (X ₂ ,B ₂),M ₃	Vector Load to Block Boundary	VRX	E707	
VLC	V ₁ ,V ₂ ,M ₃	Vector Load Complement	VRR-a	E7DE	¤ VF
VLEB	V ₁ ,D ₂ (X ₂ ,B ₂),M ₃	Vector Load Element (8)	VRX	E700	
VLEF	$V_1,D_2(X_2,B_2),M_3$	Vector Load Element (32)	VRX	E703	
VLEG	$V_1,D_2(X_2,B_2),M_3$ $V_1,D_2(X_2,B_2),M_3$	Vector Load Element (64)	VRX	E702	
VLEH		Vector Load Element (16)	VRX	E701	
VLEIB	V ₁ ,D ₂ (X ₂ ,B ₂),M ₃	Vector Load Element Immediate (8)	VRI-a		
	V ₁ ,I ₂ ,M ₃	* *			
VLEIF	V ₁ ,I ₂ ,M ₃	Vector Load Element Immediate (32)	VRI-a		
/LEIG	V ₁ ,I ₂ ,M ₃	Vector Load Element Immediate (64)	VRI-a		
VLEIH	V ₁ ,I ₂ ,M ₃	Vector Load Element Immediate (16)	VRI-a		
VLGV	$R_1, V_3, D_2(B_2), M_4$	Vector Load GR from VR Element	VRS-c		
VLIP	V_{1},I_{2},I_{3}	Vector Load Immediate Decimal	VRI-h		
VLL	$V_1,R_3,D_2(B_2)$	Vector Load With Length	VRS-b	E737	¤ VF
VLLEZ	$V_1,D_2(X_2,B_2),M_3$	Vector Load Logical Element and Zero	VRX	E704	¤ VF
VLM	$V_1, V_3, D_2(B_2)$	Vector Load Multiple	VRS-a	E736	¤VF
VLP	V_1, V_2, M_3	Vector Load Positive	VRR-a	E7DF	¤ VF
VLR	V_1, V_2	Vector Load	VRR-a	E756	¤VF
VLREP	$V_1,D_2(X_2,B_2),M_3$	Vector Load and Replicate	VRX	E705	¤VF
VLRL	$V_1,D_2(B_2),I_3$	Vector Load Rightmost with Length	VSI	E635	¤ VD
VLRLR	$V_1, R_3, D_2(B_2)$	Vector Load Rightmost with Length	VRS-d	E637	□ VD
VLVG	V ₁ ,R ₃ ,D ₂ (B ₂),M ₄	Vector Load VR Element from GR	VRS-b	E722	¤ VF
VLVGP	V ₁ ,R ₂ ,R ₃	Vector Load VR from GRs Disjoint	VRR-f	E762	¤ VF
VMAE	V ₁ ,V ₂ ,V ₃ ,V ₄ ,M ₅	Vector Multiply and Add Even	VRR-d	E7AE	¤VF
VMAH	V ₁ ,V ₂ ,V ₃ ,V ₄ ,M ₅	Vector Multiply and Add High	VRR-d	E7AB	¤ VF
VMAL	V ₁ ,V ₂ ,V ₃ ,V ₄ ,M ₅	Vector Multiply and Add Low	VRR-d		
VMALE	V ₁ ,V ₂ ,V ₃ ,V ₄ ,M ₅	Vector Multiply and Add Logical Even	VRR-d		
VMALH	V ₁ ,V ₂ ,V ₃ ,V ₄ ,M ₅	Vector Multiply and Add Logical High	VRR-d		
VMALO		Vector Multiply and Add Logical Odd	VRR-d		
VMAO	V ₁ ,V ₂ ,V ₃ ,V ₄ ,M ₅	Vector Multiply and Add Odd	VRR-d		
VIVIAO	V ₁ ,V ₂ ,V ₃ ,V ₄ ,M ₅	• • •	VRR-c		
VMH	V ₁ ,V ₂ ,V ₃ ,M ₄	Vector Multiply Even			
	V ₁ ,V ₂ ,V ₃ ,M ₄	Vector Multiply High	VRR-c		
VML	V ₁ ,V ₂ ,V ₃ ,M ₄	Vector Multiply Low	VRR-c		
VMLE	V ₁ ,V ₂ ,V ₃ ,M ₄	Vector Multiply Logical Even	VRR-c		
VMLH	V ₁ ,V ₂ ,V ₃ ,M ₄	Vector Multiply Logical High	VRR-c		
VMLO	V_1, V_2, V_3, M_4	Vector Multiply Logical Odd	VRR-c		
VMN	V_1, V_2, V_3, M_4	Vector Minimum	VRR-c		
VMNL	V_1, V_2, V_3, M_4	Vector Minimum Logical	VRR-c	E7FC	¤ VF
VMO	V_1, V_2, V_3, M_4	Vector Multiply Odd	VRR-c		
VMP	V_1, V_2, V_3, I_4, M_5	Vector Multiply Decimal	VRI-f	E678	a C _* AE
VMRH	V_1, V_2, V_3, M_4	Vector Merge High	VRR-c	E761	¤ VF
VMRL	V_1, V_2, V_3, M_4	Vector Merge Low	VRR-c	E760	¤VF
VMSL	$V_1, V_2, V_3, V_4, M_5, M_6$	Vector Multiply Sum Logical	VRR-d	E6B8	¤ V1
VMSP	V_1, V_2, V_3, I_4, M_5	Vector Multiply and Shift Decimal	VRI-f	E679	¤c* VD
VMX	V ₁ ,V ₂ ,V ₃ ,M ₄	Vector Maximum	VRR-c		
VMXL	V ₁ ,V ₂ ,V ₃ ,M ₄	Vector Maximum Logical	VRR-c		
VN	V ₁ ,V ₂ ,V ₃	Vector AND	VRR-c		
VNC	V ₁ ,V ₂ ,V ₃	Vector AND with Complement	VRR-c		
VNN	V ₁ ,V ₂ ,V ₃ V ₁ ,V ₂ ,V ₃	Vector NAND	VRR-c		
VNO	V ₁ ,V ₂ ,V ₃ V ₁ ,V ₂ ,V ₃	Vector NOR	VRR-c		
	V ₁ ,V ₂ ,V ₃ V ₁ ,V ₂ ,V ₃	Vector Not Exclusive OR	VRR-c		
VNX					

Mne-			For-	Op-	Class &
monic	Operands	Name	mat	code	Notes
VO	V_1, V_2, V_3	Vector OR	VRR-c		¤ VF
VOC	V ₁ ,V ₂ ,V ₃	Vector OR with Complement	VRR-c		
VPDI	V_1, V_2, V_3, M_4	Vector Permute Doubleword Immediate	VRR-c		
VPERM	V ₁ ,V ₂ ,V ₃ ,V ₄	Vector Permute	VRR-e		
VPK	V ₁ ,V ₂ ,V ₃ ,M ₄	Vector Pack	VRR-c		
VPKLS	V ₁ ,V ₂ ,V ₃ ,M ₄ ,M ₅	Vector Pack Logical Saturate			¤ c* VF
VPKS VPKZ	V ₁ ,V ₂ ,V ₃ ,M ₄ ,M ₅	Vector Pack Saturate			¤ c* VF
	V ₁ ,D ₂ (B ₂),I ₃	Vector Pack Zoned	VSI	E634	
VPSOP	V ₁ ,V ₂ ,M ₃	Vector Population Count Vector Perform Sign Operation Decimal	VRR-a		¤c*VD
VREP	V ₁ ,V ₂ ,I ₃ ,I ₄ ,M ₅ V ₁ ,V ₃ ,I ₂ ,M ₄	Vector Replicate	VRI-c		
VREPI	V ₁ , V ₃ , I ₂ , IVI ₄ V ₁ , I ₂ , M ₃	Vector Replicate Immediate	VRI-a		
VRP	V ₁ , I ₂ , IVI ₃ V ₁ , V ₂ , V ₃ , I ₄ , M ₅	Vector Remainder Decimal			¤c* VD
VS	V ₁ ,V ₂ ,V ₃ ,I ₄ ,IVI ₅ V ₁ ,V ₂ ,V ₃ ,M ₄	Vector Subtract	VRR-c		
VSBCBI	V ₁ , V ₂ , V ₃ , W ₄ V ₁ , V ₂ , V ₃ , V ₄ , M ₅	Vector Subtract With Borrow Compute	VRR-d		
		Borrow Indication			
VSBI	V ₁ ,V ₂ ,V ₃ ,V ₄ ,M ₅	Vector Subtract With Borrow Indication	VRR-d		
VSCBI	V ₁ ,V ₂ ,V ₃ ,M ₄	Vector Subtract Compute Borrow Indication			
VSCEF	V ₁ ,D ₂ (V ₂ ,B ₂),M ₃	Vector Scatter Element (32)	VRV	E71B	
VSCEG	V ₁ ,D ₂ (V ₂ ,B ₂),M ₃	Vector Scatter Element (64)	VRV	E71A	
VSDP	V ₁ ,V ₂ ,V ₃ ,I ₄ ,M ₅	Vector Shift and Divide Decimal	VRI-f		¤c* VD
VSEG	V ₁ ,V ₂ ,M ₃	Vector Sign Extend to Doubleword	VRR-a		
VSEL	V ₁ ,V ₂ ,V ₃ ,V ₄	Vector Select	VRR-e		
VSL	V ₁ ,V ₂ ,V ₃	Vector Shift Left	VRR-c		
VSLB VSLDB	V ₁ ,V ₂ ,V ₃	Vector Shift Left By Byte	VRR-c		
	V ₁ ,V ₂ ,V ₃ ,I ₄	Vector Shift Left Double By Byte	VRI-d		¤c*VD
VSP VSRA	V ₁ ,V ₂ ,V ₃ ,I ₄ ,M ₅	Vector Subtract Decimal	VRI-I		
VSRAB	V ₁ ,V ₂ ,V ₃	Vector Shift Right Arithmetic Vector Shift Right Arithmetic By Byte	VRR-c		
VSRL	V_1, V_2, V_3 V_1, V_2, V_3	Vector Shift Right Logical	VRR-c		
VSRLB	V ₁ , V ₂ , V ₃ V ₁ , V ₂ , V ₃	Vector Shift Right Logical By Byte	VRR-c		
VSRP	V ₁ , V ₂ , I ₃ , I ₄ , M ₅	Vector Shift and Round Decimal			¤c* VD
VST	V ₁ ,D ₂ (X ₂ ,B ₂)	Vector Store	VRX	E70E	
VSTEB	$V_1,D_2(X_2,B_2),M_3$	Vector Store Element (8)	VRX	E708	
VSTEF	$V_1,D_2(X_2,B_2),M_3$	Vector Store Element (32)	VRX	E70B	
VSTEG	V ₁ ,D ₂ (X ₂ ,B ₂),M ₃	Vector Store Element (64)	VRX	E70A	¤ VF
VSTEH	V ₁ ,D ₂ (X ₂ ,B ₂),M ₃	Vector Store Element (16)	VRX	E709	¤ VF
VSTL	V ₁ ,R ₃ ,D ₂ (B ₂)	Vector Store With Length	VRS-b	E73F	¤ VF
VSTM	V ₁ ,V ₃ ,D ₂ (B ₂)	Vector Store Multiple	VRS-a	E73E	۵VF
VSTRC		Vector String Range Compare	VRR-d	E78A	¤ c* VF
VSTRL	V ₁ ,D ₂ (B ₂),I ₃	Vector Store Rightmost with Length	VSI	E63D	αVD
VSTRLR	$V_1,R_3,D_2(B_2)$	Vector Store Rightmost with Length	VRS-d	E63F	αVD
VSUM	V_1, V_2, V_3, M_4	Vector Sum Across Word	VRR-c	E764	۵VF
VSUMG	V_1, V_2, V_3, M_4	Vector Sum Across Doubleword	VRR-c	E765	۵VF
VSUMQ	V_1, V_2, V_3, M_4	Vector Sum Across Quadword	VRR-c	E767	¤VF
VTM	V_1,V_2	Vector Test Under Mask	VRR-a	E7D8	¤ VF
VTP	V ₁	Vector Test Decimal	VRR-g	E65F	¤c* VD
VUPH	V_1, V_2, M_3	Vector Unpack High	VRR-a	E7D7	¤ VF
VUPKZ	$V_1,D_2(B_2),I_3$	Vector Unpack Zoned	VSI	E63C	¤VD
VUPL	V_1, V_2, M_3	Vector Unpack Low	VRR-a	E7D6	۵VF
VUPLH	V_1, V_2, M_3	Vector Unpack Logical High	VRR-a	E7D5	۵VF
VUPLL	V_1, V_2, M_3	Vector Unpack Logical Low	VRR-a	E7D4	¤ VF
VX	V_1, V_2, V_3	Vector Exclusive OR	VRR-c		
WFC	V_1, V_2, M_3, M_4	Vector FP Compare Scalar	VRR-a		
WFK	V_1, V_2, M_3, M_4	Vector FP Compare and Signal Scalar	VRR-a	E7CA	۵VF
Χ	$R_1, D_2(X_2, B_2)$	Exclusive OR (32)	RX-a	57	С
XC	$D_1(L,B_1), D_2(B_2)$	Exclusive OR (character)	SS-a	D7	пС
XG	$R_1, D_2(X_2, B_2)$	Exclusive OR (64)	RXY-a		
XGR	R_1,R_2	Exclusive OR (64)	RRE	B982	
XGRK	R ₁ ,R ₂ ,R ₃	Exclusive OR (64)	RRF-a	B9E7	c DO

Mne- monic	Operands	Name	For- mat	Op- code	Class & Notes
XI	$D_1(B_1), I_2$	Exclusive OR Immediate	SI	97	С
XIHF	R_1,I_2	Exclusive OR Immediate (high)	RIL-a	C06	c El
XILF	R_1,I_2	Exclusive OR Immediate (low)	RIL-a	C07	c El
XIY	$D_1(B_1), I_2$	Exclusive OR Immediate	SIY	EB57	c LD
XR	R_1,R_2	Exclusive OR (32)	RR	17	С
XRK	R ₁ ,R ₂ ,R ₃	Exclusive OR (32)	RRF-a	B9F7	c DO
XSCH		Cancel Subchannel	S	B276	рс
XY	$R_1,D_2(X_2,B_2)$	Exclusive Or (32)	RXY-a	E357	c LD
ZAP	D ₁ (L ₁ ,B ₁),D ₂ (L ₂ ,B ₂)	Zero and Add	SS-b	F8	пС

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

Point Operand Lengths and Types:		
Extended (binary, decimal or hex)	LB	Long binary
Extended binary	LD	Long decimal
Extended decimal	LH	Long hex
Extended hex	S	Short (binary, decimal or hex)
Extended hex (low-order part)	SB	Short binary
Extended hex (high-order part)	SD	Short decimal
Long (binary, decimal or hex)	SH	Short hex
	Extended (binary, decimal or hex) Extended binary Extended decimal Extended hex Extended hex Extended hex (low-order part) Extended hex (high-order part)	Extended (binary, decimal or hex) Extended binary Extended decimal Extended hex S Extended hex (low-order part) SB Extended hex (high-order part)

No	tes:			
	&	Combination of fields	L1	Load/store-on-condition facility 1
	Ø	One or more restrictions apply in the	L2	Load/store-on-condition facility 2
		transactional-execution mode	LD	Long-displacement facility
	С	Condition code set	LT	Load-and-trap facility
	C*	Condition code may be set based on	LZ	Load-and-zero-rightmost-byte facility
		control in the instruction	M3	Message-security assist extension 3
	i	Interruptible instruction	M4	Message-security assist extension 4
	n	New condition code loaded	M5	Message-security assist extension 5
	р	Privileged instruction; restricted in the	M8	Message-security assist extension 8
		transactional-execution mode	MI1	Miscellaneous-instructions facility 1
	q	Semiprivileged instruction; restricted in	MI2	Miscellaneous-instructions facility 2
		the transactional-execution mode	MO	Move-with-optional-specifications
	u	Condition code is unpredictable		facility
	CS	Compare-and-swap-and-store facility	MS	Message-security assist
	CT	Configuration topology facility	N	New in z/Architecture
	CX	Constrained-transactional-execution	N3	New in z/Architecture and added to
		facility		ESA/390
	D2	DAT-enhancement facility 2	PA	Processor-assist facility
	DE	DAT-enhancement facility	PC	DFP-packed-conversion facility
	DO	Distinct-operands facility	PE	Parsing-enhancement facility
	E2	Extended-translation facility 2	PF	PFPO facility
	E3	Extended-translation facility 3	PK	Population-count facility
	ED1	Enhanced-DAT facility 1	RA	ASN-and-LX-reuse facility
	ED2		RB	Reset-reference-bits multiple facility
	EH	Execution-hint facility	SC	Store-clock-fast facility
	El	Extended-immediate facility	TE	Test-pending-external-interruption
	ES	Expanded-storage facility		facility
	ET	Extract-CPU-time facility	TF	Decimal-floating-point facility
	F	Floating-point-extension facility	TR	Decimal-floating-point-rounding facility
	FG	FPR-GPR-transfer facility	TS	TOD-clock-steering facility
	FL	Store-facility-list-extended facility	TX	Transactional-execution facility
	FS	Floating-point-support-sign-handling	UE	HFP unnormalized-extension facility
		facility	VD	Vector-packed-decimal facility
_	GE	General-instructions-extension facility	VF	Vector facility for z/Architecture
	GF	Guarded-storage facility	V1	Vector-enhancements facility 1
	HM	HFP multiply-and-add/subtract facility	XF	IEEE-exception-support facility
	HW	High-word facility	XX	Execute-extension facility
	IA	Interlocked-access facility	ZF	DFP zoned-conversion facility
ı	IM	Insert-reference-bits-multiple facility		

Machine Instructions by Operation Code

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

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

OpCode Mnemonic OpCode Mnemonic B22A RRBE B307 MXDBR B370 LPDFR B22B SSKE B308 KEBR B371 LINDFR B22D DXR B308 KEBR B372 CPSDR B22D PGNI B30B SEBR B373 LCDFR B22F PGOUT B30C MDEBR B375 LZDR B231 HSCH B30D DEBR B376 LZXR B231 HSCH B30D DEBR B376 LZXR B232 MSCH B30F MSEBR B377 FIER B232 MSCH B311 LVDBR B384 SFPC B234 STSCH B311 LVDBR B385 SFASR B235 TSCH B312 LTDBR B386 EFPC B236 TSCH B311 LVDBR B389 CELFBR B233 STCHW B315 SQDBR
B22B SSKE B308 KEBR B371 LNDFR B22C TB B309 CEBR B372 CPSDR B22E PGIN B30A AEBR B373 LOFR B22F PGOUT B30C MDEBR B375 LZDR B230 CSCH B30D DEBR B376 LZXR B231 HSCH B30E MAEBR B377 FIER B232 MSCH B30F MSEBR B377 FIER B233 SSCH B310 LPDBR B384 SFPC B234 STSCH B311 LNDBR B385 SFASR B235 TSCH B312 LCDBR B390 CELFBR B236 TSCH B312 LCDBR B390 CELFBR B237 SAL B314 SQEBR B391 CDLFBR B237 SAL B315 SQUBR B392 CXLFBR B238 RSCH <t< th=""></t<>
B22C TB B309 CEBR B372 CPSDR B22D DXR B30A AEBR B373 LCDFR B22E PGOUT B30C MDEBR B375 LZDR B230 CSCH B30D DEBR B376 LZXR B231 HSCH B30E MAEBR B377 FIER B232 MSCH B30F MSEBR B377 FIER B233 SSCH B310 LPDBR B384 SFPC B234 STSCH B311 LNDBR B385 SFASR B235 TSCH B312 LTDBR B386 EFPC B236 TPI B313 LODBR B390 CELFBR B237 SAL B314 SOEBR B391 CDLFBR B238 RSCH B315 SQDBR B392 CXLFBR B238 RSCH B315 SQDBR B392 CXLFBR B238 RCHP <t< td=""></t<>
B22D DXR B30A AEBR B373 LCDFR B22E PGIN B30B SEBR B374 LZER B22F PGOUT B30C MDEBR B375 LZDR B231 HSCH B30E MAEBR B377 FIER B231 HSCH B30E MAEBR B377 FIER B232 MSCH B301 LPDBR B384 SFPC B233 SSCH B310 LPDBR B384 SFPC B233 SSCH B311 LNDBR B385 SFASR B234 STSCH B311 LNDBR B385 SFASR B235 TSCH B312 LTDBR B386 SFPC B236 TSCH B311 LNDBR B385 SFASR B237 SAL B314 SOEBR B391 CDLFBR B238 STCFW B316 SQXBR B394 CEFBRA B239 STCFW
B22E PGIN B30B SEBR B374 LZER B22F PGOUT B30C MDEBR B375 LZDR B230 CSCH B30D DEBR B377 FIER B231 HSCH B30E MAEBR B377 FIER B232 MSCH B30F MSEBR B377 FIER B233 SSCH B310 LPDBR B384 SFPC B234 STSCH B311 LNDBR B385 SFASR B235 TSCH B312 LCDBR B390 CELFBR B236 TSCH B313 LCDBR B390 CELFBR B237 SAL B314 SQEBR B391 CDLFBR B238 RSCH B315 SQWBR B394 CEFBRA B238 STCPS B317 MEEBR B394 CEFBRA B238 STCPS B317 MEEBR B394 CEFBRA B230 SCHM
B22F PGOUT B30C MDEBR B375 LZDR B230 CSCH B30D DEBR B376 LZXR B231 HSCH B30E MAEBR B377 FIER B232 MSCH B30F MSEBR B377 FIDR B233 SSCH B310 LPDBR B384 SFPC B234 STSCH B311 LNDBR B385 SFASR B235 TSCH B312 LTDBR B380 CEFPC B236 TPI B313 LODBR B390 CELFBR B236 TPI B314 SOEBR B391 CDLFBR B238 RSCH B315 SQDBR B392 CXLFBR B238 TSCRW B316 SOXBR B394 CEFBR B238 RCHP B318 KDBR B395 CDFBR B230 SCHM B319 CDBR B395 CDFBR B231 CKSM
B30E MAEBR B377 FIER B232 MSCH B30F MSEBR B377 FIDR B323 SSCH B310 LPDBR B384 SFPC B234 STSCH B311 LNDBR B385 SFASR B235 TSCH B312 LTDBR B380C EFPC B236 TSCH B313 LCDBR B390 CELFBR B237 SAL B314 SQEBR B391 CDLFBR B237 SAL B314 SQEBR B391 CDLFBR B238 STCRW B316 SQWBR B392 CXLFBR B239 STCRW B316 SQWBR B394 CEFBR B230 STCRW B316 SQWBR B394 CEFBR B230 CELFBR B317 MEEBR B394 CEFBR B230 CEFBR B318 KDBR B395 CDFBR B230 CXFBR B314 ADBR B395 CDFBR B240 BAKR B314 ADBR B395 CDFBR B241 CKSM B318 SDBR B396 CXFBR B244 SQLF B310 DDBR B398 CFEBR B245 SQLF B310 DDBR B398 CFEBR B246 STURA B315 MADBR B399 CFDBR B246 STURA B315 MADBR B399 CFDBR B248 PALB B324 LDER B394 CFEBR B249 EREG B325 LXDR B394 CFEBR B249 CFEBR B326 LXER B390 CLFDBR B246 CFEBR B326 LXER B390 CLFDBR B247 B348 CEFBR B341 CDLFDBR B248 LIPA B325 MAER B330 CLFDBR B246 CAFBR B346 CLFDBR B247 CAFBR B336 CAFBR B346 CLFDBR B250 CSP B330 MYIR B344 CEGBR B250 CSP B330 MYIR B345 CDGBR B255 MSST B331 LNXBR B346 CXGBR B255 SRST B341 LNXBR B346 CXGBR B256 SRST B341 LNXBR B348 CGEBR B257 CUSE B330 MYIR B346 CXGBR B257 CUSE B331 LNXBR B348 CGEBR B257 CUSE B331 LNXBR B346 CXGBR B257 CUSE B331 MADR B346 CXGBR B257 CUSE B345 LDXBR B340 CGBBR B257 CUSE B345 LDXBR B346 CXGBR B257 STCKF B344 LEDBR B346 CXGBR B257 STCKF B345 LDXBR B346 CXGBR B345 LDXBR B346 CAGBR B346 CXGBR B346 CXGB
B232 MSCH B30F MSEBR B37F FIDR B233 SSCH B310 LPDBR B384 SFPC B234 STSCH B311 LINDBR B385 SFASR B235 TSCH B312 LTDBR B386 EFPC B236 TPI B313 LODBR B390 CELFBR B238 RSCH B315 SQDBR B391 CDLFBR B238 RSCH B315 SQDBR B392 CXLFBR B239 STCRW B316 SQXBR B392 CXLFBR B238 RSCH B315 SQDBR B392 CXLFBR B238 RCHP B318 KDBR B394 CEFBRA B238 RCHP B318 KDBR B395 CDFBRA B230 SCHMM B319 CDBR B395 CDFBRA B241 CKSM B318 SDBR B396 CXFBRA B241 CKSM
B233 SSCH B310 LPDBR B384 SFPC B234 STSCH B311 LNDBR B385 SFASR B235 TSCH B312 LTDBR B38C EFPC B236 TPI B313 LCDBR B390 CELFBR B237 SAL B314 SOEBR B391 CDLFBR B238 RSCH B315 SQDBR B392 CXLFBR B238 STCPS B316 SQXBR B394 CEFBR B238 STCPS B317 MEEBR B394 CEFBR B238 STCPS B317 MEEBR B394 CEFBR B238 STCPS B317 MEEBR B394 CEFBR B230 SCHM B318 KDBR B395 CDFBRA B240 BAKR B31A ADBR B395 CDFBRA B241 CKSM B31B SDBR B396 CXFBRA B244 SQER
B234 STSCH B311 LNDBR B385 SFASR B235 TSCH B312 LITDBR B38C EFPC B236 TPI B313 LCDBR B390 CELFBR B237 SAL B314 SQEBR B391 CDLFBR B238 RSCH B315 SQDBR B392 CXLFBR B239 STCPS B317 MEEBR B394 CEFBRA B230 SCCHM B318 KOBR B395 CDFBR B230 SCHM B319 CDBR B395 CDFBR B230 SCHM B319 CDBR B396 CXFBR B241 CKSM B318 SDBR B396 CXFBR B241 CKSM B318 SDBR B396 CXFBR B244 SQER B310 DDBR B398 CFEBRA B245 SQER B31D DDBR B398 CFEBRA B246 STURA
B235 TSCH B312 LTDBR B38C EFPC B236 TPI B313 LCDBR B390 CELFBR B237 SAL B314 SOEBR B391 CDLFBR B238 RSCH B315 SQDBR B392 CXLFBR B238 RSCH B315 SQDBR B392 CXLFBR B239 STCRW B316 SQXBR B394 CEFBRA B238 RCHP B318 KDBR B395 CDFBRA B230 SCHM B319 CDBR B395 CDFBRA B230 SCHM B318 ADBR B395 CDFBRA B240 BAKR B314 ADBR B395 CDFBRA B241 CKSM B318 SDBR B396 CXFBRA B241 CKSM B318 SDBR B396 CXFBRA B244 SODR B31C MDBR B399 CFDBR B245 SQER
B236 TPI B313 LCDBR B390 CELFBR B237 SAL B314 SQEBR B391 CDLFBR B238 RSCH B315 SQDBR B392 CXLFBR B239 STCRW B316 SQXBR B394 CEFBR B23A STCPS B317 MEEBR B394 CEFBR B23B RCHP B318 KDBR B395 CDFBR B23B SCHM B319 CDBR B395 CDFBRA B240 BAKR B31A ADBR B396 CXFBRA B241 CKSM B31B SDBR B396 CXFBRA B244 SQDR B31C MDBR B398 CFEBR B244 SQDR B31D DDBR B398 CFEBR B245 SQER B31D DDBR B398 CFEBR B245 SQER B31F MSDBR B399 CFDBR B248 PALB
B237 SAL B314 SQEBR B391 CDLFBR B238 RSCH B315 SQDBR B392 CXLFBR B239 STCRW B316 SQXBR B394 CEFBRA B230 STCPS B317 MEEBR B394 CEFBRA B230 SCHM B318 KDBR B395 CDFBRA B240 BAKR B318 CDBR B396 CXFBR B241 CKSM B318 SDBR B396 CXFBR B241 CKSM B318 SDBR B396 CXFBR B244 SQDR B310 MDBR B398 CFEBRA B245 SQER B31D DDBR B398 CFEBRA B246 STURA B31E MADBR B399 CFDBRA B247 MSTA B31E MADBR B399 CFDBRA B248 PALB B324 LDER B398 CFEBRA B248 BABA
B238 RSCH B315 SQDBR B392 CXLFBR B239 STCRW B316 SQXBR B394 CEFBRA B23A STCPS B317 MEEBR B394 CEFBRA B23B RCHP B318 KDBR B395 CDFBRA B23C SCHM B319 CDBR B395 CDFBRA B240 BAKR B31A ADBR B396 CXFBRA B241 CKSM B31B SDBR B396 CXFBRA B241 CKSM B31B SDBR B396 CXFBRA B244 SQDR B31C MDBR B396 CXFBRA B244 SQDR B31D DDBR B398 CFBBRA B245 SQER B31B MDBR B399 CFDBR B246 STURA B31E MADBR B399 CFDBR B246 STURA B31F MSDBR B399 CFDBR B249 EREG </td
B239 STCRW B316 SQXBR B394 CEFBRA B23A STCPS B317 MEEBR B394 CEFBRA B23B RCHP B318 KDBR B395 CDFBRA B23C SCHM B319 CDBR B395 CDFBRA B240 BAKR B31A ADBR B396 CXFBRA B241 CKSM B31B SDBR B396 CXFBRA B244 SQDR B31C MDBR B398 CFEBRA B245 SQER B31D DDBR B398 CFEBRA B244 STURA B31E MADBR B399 CFDBRA B246 STURA B31F MSDBR B399 CFDBRA B248 PALB B324 LDER B399 CFDBRA B248 PALB B325 LXDR B399 CFDBRA B249 EREG B325 LXDR B399 CFDBRA B24B LURA
B23B RCHP B318 KDBR B395 CDFBR B23C SCHM B319 CDBR B395 CDFBRA B240 BAKR B31A ADBR B396 CXFBR B241 CKSM B31B SDBR B396 CXFBRA B244 SODR B31C MDBR B398 CFEBRA B245 SQER B31D DDBR B398 CFEBRA B246 STURA B31E MADBR B399 CFDBR B247 MSTA B31E MADBR B399 CFDBR B248 PALB B324 LDER B399 CFDBR B249 EREG B325 LXDR B39A CFXBR B249 EREG B325 LXDR B39A CFXBR B249 EREG B32E MAER B39C CLFEBR B249 EREG B32E MAER B39A CFXBRA B24B LUFA
B23C SCHM B319 CDBR B395 CDFBR B240 BAKR B31A ADBR B396 CXFBR B241 SQDR B31B SDBR B396 CXFBRA B244 SQDR B31C MDBR B398 CFEBR B245 SQER B31D DDBR B398 CFEBR B246 STURA B31E MADBR B399 CFDBRA B247 MSTA B31F MSDBR B399 CFDBRA B248 PALB B324 LDER B39A CFXBRA B249 EREG B325 LXDR B39A CFXBRA B24A ESTA B326 LXER B39C CLFBBR B24B LURA B32E MAER B39C CLFBBR B24B LURA B32F MSER B39E CLFSBR B24B SAR B337 MSER B341 CDLGBR B24E SAR
B240 BAKR B31A ADBR B396 CXFBRA B241 CKSM B31B SDBR B396 CXFBRA B244 SQDR B31C MDBR B398 CFEBRA B245 SQER B31D DDBR B398 CFEBRA B246 STURA B31F MSDBR B399 CFDBRA B247 MSTA B31F MSDBR B399 CFDBRA B248 PALB B324 LDER B39A CFXBRA B249 EREG B325 LXDR B39A CFXBRA B249 EREG B325 LXDR B39A CFXBRA B249 EREG B325 LXER B39C CLFBBR B249 EREG B325 LXER B39C CLFBBR B240 LIRA B32E MAER B39D CLFBBR B240 CPYA B336 SQXR B34O CLFBBR B241 CPYA
B241 CKSM B31B SDBR B396 CXFBRA B244 SQDR B31C MDBR B398 CFEBRA B245 SQER B31D DDBR B398 CFEBRA B246 STURA B31E MADBR B399 CFDBRA B248 PALB B324 LDER B399 CFDBRA B248 PALB B324 LDER B39A CFXBRA B249 EREG B325 LXDR B39A CFXBRA B248 LURA B32E MAER B39C CLFBR B24B LURA B32E MAER B39D CLFDBR B24C TAR B32F MSER B39E CLFSBR B24C TAR B32F MSER B39E CLFSBR B24D CPYA B336 SQXR B3AO CELGBR B24F EAR B337 MEER B3A1 CDLGBR B250 CSP
B244 SQDR B31C MDBR B398 CFEBRA B245 SQER B31D DDBR B398 CFEBRA B246 STURA B31E MADBR B399 CFDBRA B247 MSTA B31F MSDBR B399 CFDBRA B248 PALB B324 LDER B39A CFXBRA B249 EREG B325 LXDR B39A CFXBRA B24A ESTA B326 LXER B39C CLFBBR B24B LURA B32E MAER B39D CLFBBR B24B LURA B32E MAER B39C CLFBBR B24D CPYA B336 SQXR B34A CPLFBR B24D CPYA B336 SQXR B34A CPLGBR B24E SAR B337 MEER B34A CDLGBR B24F EAR B338 MYLR B3A4 CEGBR B252 MSR
B245 SQER B31D DDBR B398 CFEBRA B246 STURA B31E MADBR B399 CFDBRA B247 MSTA B31F MSDDR B399 CFDBRA B248 PALB B324 LDER B39A CFXBRA B249 EREG B325 LXDR B39A CFXBRA B24A ESTA B326 LXER B39C CLFEBR B24B LURA B32E MAER B39D CLFDBR B24C TAR B32F MSER B39D CLFDBR B24D CPYA B336 SQXR B34D CLFBBR B24D CPYA B336 SQXR B34O CLGBR B24E SAR B337 MEER B3A1 CDLGBR B24F EAR B338 MAYLR B3A2 CXLGBR B250 CSP B339 MYLR B3A4 CEGBRA B254 MVPG
B246 STURA B31E MADBR B399 CFDBR B247 MSTA B31F MSDBR B399 CFDBRA B248 PALB B324 LDER B399 CFDBRA B249 EREG B325 LXDR B39A CFXBRA B24B LURA B32E LXER B39C CLFBBR B24B LURA B32E MAER B39D CLFDBR B24C TAR B32F MSER B39E CLFSBR B24C TAR B32F MSER B39E CLFSBR B24D CPYA B336 SQXR B3AO CELGBR B24F EAR B337 MEER B3A1 CDLGBR B24F EAR B338 MAYLR B3A2 CXLGBR B250 CSP B339 MYR B3A4 CEGBR B254 MVPG B338 MYR B3A5 CDGBR B255 MST <
B248 PÅLB B324 LDER B39A CFXBR B249 EREG B325 LXDR B39A CFXBRA B24A ESTA B326 LXER B39C CLFEBR B24B LURA B32E MAER B39D CLFDBR B24C TAR B32F MSER B39E CLFDBR B24D CPYA B336 SQXR B3A0 CELGBR B24E SAR B337 MEER B3A1 CDLGBR B24E EAR B338 MAYLR B3A2 CXLGBR B250 CSP B339 MYLR B3A4 CEGBRA B252 MSR B33A MAYR B3A4 CEGBRA B254 MVPG B33B MYR B3A5 CDGBR B255 MVST B33C MAYHR B3A5 CDGBR B255 MVST B33D MYHR B3A6 CXGBR B257 CUSE <t< td=""></t<>
B249 EREG B325 LXDR B39A CFXBRA B24A ESTA B326 LXER B39C CLFEBR B24B LURA B32E MAER B39D CLFDBR B24C TAR B32F MSER B39E CLFDBR B24C TAR B33F MSER B39E CLFBR B24D CPYA B336 SOXR B3A0 CELGBR B24F EAR B337 MEER B3A1 CDLGBR B24F EAR B338 MAYLR B3A2 CXLGBR B250 CSP B339 MYLR B3A4 CEGBR B255 MSR B33A MAYR B3A4 CEGBR B254 MVPG B33B MYR B3A5 CDGBR B255 MVST B33D MYHR B3A6 CXGBR B255 MSA B33F MSDR B3A6 CXGBR B258 BSA B33F
B24A ESTA B326 LXER B39C CLFEBR B24B LURA B32E MAER B39D CLFDBR B24C TAR B32F MSER B39E CLFDBR B24D CPYA B336 SOXR B3A0 CELGBR B24E SAR B337 MEER B3A1 CDLGBR B24F EAR B338 MAYLR B3A2 CXLGBR B250 CSP B339 MYLR B3A4 CEGBR B252 MSR B33A MAYR B3A4 CEGBR B255 MVST B33C MYHR B3A5 CDGBR B255 MVST B33C MYHR B3A6 CXGBR B257 CUSE B33D MYHR B3A6 CXGBR B258 BSG B33E MSDR B3A8 CGEBR B250 CLST B340 LPXBR B3A8 CGEBR B250 CLST B
B24B LURA B32E MAER B39D CLFDBR B24C TAR B32F MSER B39E CLFDBR B24D CPYA B336 SQXR B3A0 CELGBR B24E SAR B337 MEER B3A1 CDLGBR B24F EAR B338 MYLR B3A2 CXLGBR B250 CSP B339 MYLR B3A4 CEGBRA B252 MSR B33A MAYR B3A4 CEGBRA B254 MVPG B33B MYR B3A5 CDGBR B255 MVST B33C MAYHR B3A5 CDGBR B257 CUSE B33D MYHR B3A6 CXGBR B258 BSG B33E MADR B3A6 CXGBR B255 ABA B33F MSDR B3A8 CGEBR B250 CLST B340 LPXBR B3A8 CGEBRA B25E SRST B
B24C TAR B32F MSER B39E CLFXBR B24D CPYA B336 SQXR B3A0 CELGBR B24F SAR B337 MEER B3A1 CDLGBR B24F EAR B338 MAYLR B3A2 CXLGBR B250 CSP B338 MYR B3A4 CEGBR B252 MSR B33A MAYR B3A4 CEGBR B254 MVPG B33B MYR B3A5 CDGBR B255 MVST B33C MAYHR B3A5 CDGBR B255 MVST B33D MYHR B3A6 CXGBR B258 BSG B33F MADR B3A6 CXGBR B258 BSA B33F MSDR B3A6 CXGBR B250 CLST B340 LPXBR B3A8 CGEBR B25E SRST B341 LNXBR B3A9 CGDBR B263 CMPSC B34
B24D CPYA B336 SQXR B3A0 CELGBR B24E SAR B337 MEER B3A1 CDLGBR B24F EAR B338 MAYLR B3A2 CXLGBR B250 CSP B339 MYLR B3A4 CEGBR B252 MSR B3A3 MAYR B3A4 CEGBRA B254 MVPG B338 MYR B3A5 CDGBRA B255 MVST B33C MAYHR B3A5 CDGBRA B257 CUSE B33D MYHR B3A6 CXGBRA B258 BSG B33E MADR B3A6 CXGBRA B258 BSG B33F MSDR B3A8 CGEBR B250 CLST B340 LPXBR B3A8 CGEBRA B251 CLST B340 LPXBR B3A9 CGDBRA B252 SRST B341 LIXBR B3A9 CGDBRA B276 XSCH
B24E SAR B337 MEER B341 CDLGBR B24F EAR B338 MAYLR B3A2 CXLGBR B250 CSP B339 MYLR B3A4 CEGBR B252 MSR B33A MAYR B3A4 CEGBRA B255 MVST B33C MAYHR B3A5 CDGBR B255 MVST B33C MAYHR B3A5 CDGBRA B257 CUSE B33D MYHR B3A6 CXGBRA B258 BSG B33E MADR B3A6 CXGBRA B25A BSA B33F MSDR B3A8 CGEBRA B25D CLST B340 LPXBR B3A8 CGEBRA B25E SRST B341 LIXBR B3A9 CGDBRA B263 CMPSC B342 LIXBR B3A9 CGDBRA B276 XSCH B343 LOXBR B3AA CGXBR B277 RP
B24F EAR B338 MAYLR B3A2 CXLGBR B250 CSP B339 MYLR B3A4 CEGBR B252 MSR B33A MAYR B3A4 CEGBRA B254 MVPG B33B MYR B3A5 CDGBRA B255 MVST B33C MAYHR B3A5 CDGBRA B257 CUSE B33D MYHR B3A6 CXGBRA B258 BSG B33F MADR B3A6 CXGBRA B25A BSA B33F MSDR B3A6 CXGBRA B25D CLST B340 LPXBR B3A8 CGEBRA B25E SRST B341 LNXBR B3A9 CGDBRA B263 CMPSC B342 LCXBR B3AA CGXBR B265 SRST B341 LNXBR B3A9 CGDBRA B276 XSCH B343 LCXBR B3AA CGXBR B277 RP
B252 MSR B33A MAYR B3A4 CEGBRA B254 MVPG B33B MYR B3A5 CDGBR B255 MVST B33C MAYHR B3A5 CDGBR B257 CUSE B33D MYHR B3A6 CXGBR B258 BSG B33E MADR B3A6 CXGBR B25A BSA B33F MSDR B3A8 CGEBR B25D CLST B340 LPXBR B3A8 CGEBRA B25E SRST B341 LIXBR B3A9 CGDBR B263 CMPSC B342 LTXBR B3A9 CGDBR B276 XSCH B343 LOXBR B3AA CGXBR B277 RP B344 LEDBR B3AA CGXBR B278 SACF B344 LEDBRA B3AC CLGBR B279 SACF B345 LDXBR B3AE CLGDSR B27C STCKF
B254 MVPG B33B MYR B3A5 CDGBR B255 MVST B33C MAYHR B3A5 CDGBRA B257 CUSE B33D MYHR B3A6 CXGBR B258 BSG B33E MADR B3A6 CXGBRA B25A B3A B33F MSDR B3A8 CGEBRA B25D CLST B340 LPXBR B3A8 CGEBRA B25E SRST B341 LNXBR B3A9 CGDBRA B263 CMPSC B342 LCXBR B3AA CGXBR B276 XSCH B343 LCXBR B3AA CGXBR B277 RP B344 LEDBR B3AA CGXBR B279 SACF B345 LDXBR B3AD CLGEBR B270 STCKE B345 LDXBR B3AD CLGEBR B270 STCKF B345 LDXBRA B3AE CLGEBR B270 STSI
B255 MVST B33C MAYHR B3A5 CDGBRA
B257 CUSE B33D MYHR B3A6 CXGBRA B258 BSG B33E MADR B3A6 CXGBRA B25A BSA B33F MSDR B3A8 CGEBR B25D CLST B340 LPXBR B3A8 CGEBRA B26S SRST B341 LIXBR B3A9 CGDBRA B263 CMPSC B342 LTXBR B3A9 CGDBRA B276 XSCH B343 LCXBR B3AA CGXBRA B277 RP B344 LEDBR B3AA CGXBRA B278 STCKE B344 LEDBRA B3AC CLGEBR B279 SACF B345 LDXBR B3AD CLGDBR B27C STCKF B345 LDXBRA B3AE CLGSBR B27D STSI B346 LEXBRA B3B4 CEFR B299 SRNM B346 LEXBRA B3B5 CDFR B290 STFP
B258 BSG B33E MADR B3A6 CXGBRA B25A BSA B33F MSDR B3A8 CGEBRA B25D CLST B340 LPXBR B3A8 CGEBRA B25E SRST B341 LNXBR B3A9 CGDBRA B263 CMPSC B342 LCXBR B3A9 CGDBRA B276 XSCH B343 LCXBR B3AA CGXBR B277 RP B344 LEDBR B3AA CGXBRA B278 STCKE B344 LEDBRA B3AC CLGEBR B279 SACF B345 LDXBR B3AD CLGDBR B27C STCKF B345 LDXBRA B3AE CLGSBR B27D STSI B346 LEXBRA B3B4 CEFR B299 SRNM B346 LEXBRA B3B5 CDFR B290 STFPC B347 FIXBR B3B6 CXFR
B25A BSA B33F MSDR B3A8 CGEBR B25D CLST B340 LPXBR B3A8 CGEBRA B25E SRST B341 LNXBR B3A9 CGDBRA B263 CMPSC B342 LTXBR B3A9 CGDBRA B276 XSCH B343 LCXBR B3AA CGXBRA B277 RP B344 LEDBRA B3AA CGXBRA B278 STCKE B344 LEDBRA B3AC CLGEBR B279 SACF B345 LDXBR B3AD CLGDBR B27C STCKF B345 LDXBRA B3AE CLGXBR B27D STSI B346 LEXBR B3B4 CEFR B299 SRNM B346 LEXBRA B3B5 CDFR B29C STFPC B347 FIXBR B3B6 CXFR
B25D CLST B340 LPXBR B348 CGEBRA B25E SRST B341 LIXBR B3A9 CGDBRA B263 CMPSC B342 LTXBR B3A9 CGDBRA B276 XSCH B343 LCXBR B3AA CGXBRA B277 RP B344 LEDBR B3AA CGXBRA B278 STCKE B344 LEDBRA B3AC CLGEBR B279 SACF B345 LDXBR B3AD CLGDBR B27C STCKF B345 LDXBRA B3AE CLGXBR B27D STSI B346 LEXBRA B3B4 CEFR B299 SRNM B346 LEXBRA B3B5 CDFR B290 STFPC B347 FIXBR B3B6 CXFR
B263 CMPSC B342 LTXBR B3A9 CGDBRA B276 XSCH B343 LCXBR B3AA CGXBR B277 RP B344 LEDBR B3AA CGXBRA B278 STCKE B344 LEDBRA B3AC CLGEBR B279 SACF B345 LDXBR B3AD CLGDBR B27C STCKF B345 LDXBRA B3AE CLGXBR B27D STSI B346 LEXBR B3B4 CEFR B299 SRNM B346 LEXBRA B3B5 CDFR B29C STFPC B347 FIXBR B3B6 CXFR
B276 XSCH B343 LCXBR B3AA CGXBR B277 RP B344 LEDBR B3AA CGXBRA B278 STCKE B344 LEDBRA B3AC CLGEBR B279 SACF B345 LDXBR B3AD CLGDBR B27C STCKF B345 LDXBRA B3AE CLGXBR B27D STSI B346 LEXBR B3B4 CEFR B299 SRNM B346 LEXBRA B3B5 CDFR B29C STFPC B347 FIXBR B3B6 CXFR
B277 RP B344 LEDBR B3AA CGXBRA B278 STCKE B344 LEDBRA B3AC CLGEBR B279 SACF B345 LDXBR B3AD CLGDBR B27C STCKF B345 LDXBRA B3AE CLGXBR B27D STSI B346 LEXBRA B3B4 CEFR B299 SRNM B346 LEXBRA B3B5 CDFR B29C STFPC B347 FIXBR B3B6 CXFR
B278 STCKE B344 LEDBRA B3AC CLGEBR B279 SACF B345 LDXBR B3AD CLGDBR B27C STCKF B345 LDXBRA B3AE CLGXBR B27D STSI B346 LEXBR B3B4 CEFR B299 SRNM B346 LEXBRA B3B5 CDFR B29C STFPC B347 FIXBR B3B6 CXFR
B279 SACF B345 LDXBR B3AD CLGDBR B27C STCKF B345 LDXBRA B3AE CLGXBR B27D STSI B346 LEXBR B3B4 CEFR B299 SRNM B346 LEXBRA B3B5 CDFR B29C STFPC B347 FIXBR B3B6 CXFR
B27C STCKF B345 LDXBRA B3AE CLGXBR B27D STSI B346 LEXBR B384 CEFR B299 SRNM B346 LEXBRA B3B5 CDFR B29C STFPC B347 FIXBR B3B6 CXFR
B27D STSI B346 LEXBR B384 CEFR B299 SRNM B346 LEXBRA B385 CDFR B29C STFPC B347 FIXBR B3B6 CXFR
B29C STFPC B347 FIXBR B3B6 CXFR
B29D LFPC B347 FIXBRA B3B8 CFER
B2A5 TRE B348 KXBR B3B9 CFDR B2A6 CU21 B349 CXBR B3BA CFXR
B2A6 CUUTF B34A AXBR B3C1 LDGR
B2A7 CU12 B34B SXBR B3C4 CEGR
B2A7 CUTFU B34C MXBR B3C5 CDGR
B2B0 STFLE B34D DXBR B3C6 CXGR
B2B1 STFL B350 TBEDR B3C8 CGER
B2B2 LPSWE B351 TBDR B3C9 CGDR B2B8 SRNMB B353 DIEBR B3CA CGXR
B2B8 SRNMB B353 DIEBR B3CA CGXR B2B9 SRNMT B357 FIEBR B3CD LGDR
B2BD LFAS B357 FIEBRA B3D0 MDTR
B2E8 PPA B358 THDER B3D0 MDTRA
B2EC ETND B359 THDR B3D1 DDTR
B2F8 TEND B35B DIDBR B3D1 DDTRA
B2FA NIAI B35F FIDBR B3D2 ADTR
B2FC TABORT B35F FIDBRA B3D2 ADTRA B2FF TRAP4 B360 LPXR B3D3 SDTR
B2FF TRAP4 B360 LPXR B3D3 SDTR B300 LPEBR B361 LNXR B3D3 SDTRA
B301 LNEBR B362 LTXR B3D4 LDETR
B302 LTEBR B363 LCXR B3D5 LEDTR
B303 LCEBR B365 LXR B3D6 LTDTR
B304 LDEBR B366 LEXR B3D7 FIDTR
B305 LXDBR B367 FIXR B3D8 MXTR
B306 LXEBR B369 CXR B3D8 MXTRA

B309	OpCode	Mnemonic		OpCode	Mnemonic		OpCode	Mnemonic
B3DA AXTR B3DA AXTRA B3DB SXTR B3DD LDXTR B3DD LDXTR B3DB B3D2 KMG B3DD CLHR B3DD LDXTR B3DD LDXTR B3DD KMGTR B3DD SHHLR B3DB SAHHLR	B3D9	DXTR		B921	CLGR		B9C8	AHHHR
B30B	B3D9	DXTRA		B925	STURG		B9C9	SHHHR
B30B	B3DA	AXTR		B926	LBR		B9CA	ALHHHR
B30B	B3DA	AXTRA		B927	LHR		B9CB	SLHHHR
B3DD	B3DB						B9CD	
B3DD								
B3DD LDXTR B82B KMO B8909 SHHLR B87D SHHLR B89D ALHHLR B87D ALHHLR B89D CHR B87D FIXTR B82E KMCTR B89D ALHHLR B89D CHR B87D CHLR B87D CCHLR B87D CCHR B87D CCGR B87D CCGTR B87D CCGTR			•					
B3DE								
B35D	B3DE							
B35E1	-				KMCTB			
B35E1 CGDTRA B35E CGDTRA B35E2 CUDTR B35E3 CDTR B35E3 CDTR B35E4 CDTR B35E5 EEDTR B35E5 EEDTR B35E6 CGXTR B35E9 CGXTR B34E1 CFDTR B35E9 CGXTR B35E9 CGXTR B35E9 CGXTRA B35E0 CGXTRA B35E1 CDFTRA B35E1 CDFTRA B35E2 CDFTRA B35E2 CDFTRA B35E3 CDFTRA B35E4 CDFTRA B35E5 CDFTRA B35E1 CDFTRA B35E1 CDFTRA B35E1 CDFTRA B35E3 CDFTA CDFTA CDFT								
B35E1								
B3E2								
B3E3								
B354 CDTR B935 KIMD B946 OGRK B357 ESDTR B941 CFDTR B956 OGRK B957 CGNTR B942 CLGDTR B958 AGRK B959 CGNTR B949 CFNTR B958 AGRK B958 AGRK B359 CGNTR B949 CFNTR B958 AGRK B958 AGRK B359 CGNTR B949 CFNTR B958 AGRK B958 AGRK B958 AGRK B356 CONTR B949 CFNTR B958 AGRK B958 AGRK B958 AGRK B958 AGRK B958 AGRK B958 AGRK B959 CFNTR B958 AGRK B958 AGRK B958 AGRK B958 AGRK B958 AGRK B959 AGRK B959 AGRK B959 AGRK B959 AGRK B959 AGRK AGRK AGRK B959 AGRK								
B35E EEDTR B94F KLMD B96F CGRK B36E KXTR B941 CFDTR B967 XGRK B36E KXTR B943 CLFDTR B968 AGRK B968 AGRK B369 CGXTR B949 CFXTR B968 ALGRK B969 ALGRK B960 AL			•					
B3E7								
B3E8								
B3E9								-
B3E9								
B3EA CUXTR B949 CFXTR B3EB CSXTR B3EC CXTR B394A CLGXTR B3EC MGRK B3EC CXTR B3ED EEXTR B3E5 CDFTR B3EF EXTR B3E5 CDFTR B3E7 CDGTR B953 CDLFTR B9F6 ORK B3F1 CDGTR B953 CDLFTR B9F6 ORK B3F1 CDGTR B953 CXFTR B9F7 XFK B3F2 CDUTR B954 CXLGTR B9F8 ARK B3F3 CDSTR B958 CXLFTR B9F8 ARK B3F3 CDSTR B959 CXFTR B9F8 ARK B3F3 CDSTR B960 CGRT B9F8 ARK B3F6 EDTR B9F7 CLRT B9F8 SLFK B3F7 RRDTR B973 CLRT B9F8 SLFK B3F9 CXGTR B9F8 CXGTR B9F8 CXGTR B9F8 CXGTR B9F8 CXGTR B9F8 CXGTR B9F9 CXGTR								
B3EB								
B3EC						١.		
B3ED								
BBSEF ESXTR B952 CDLGTR B956 ORK B3F1 CDGTR B959 CXFTR B9F6 ORK B3F1 CDGTRA B959 CXFTR B9F6 CRK B3F2 CDUTR B950 CXLGTR B9F8 ARK B3F2 CDSTR B958 CXLFTR B9F9 SRK B3F3 CDSTR B960 CGRT B9F8 ALRK B3F5 QADTR B961 CLGRT B9FB SLRK B3F5 QADTR B961 CLGRT B9FB SLRK B3F7 RADTR B973 CLRT B3FD MSRKC B3F9 CXGTR B980 NGR BB CDS B3F9 CXGTR B980 NGR BB CDS B3F9 CXGTR B981 OGR BD CLM B3FB CXSTR B982 XGR BE STCM B3FB CXSTR B983 FLOGR BF ICM EARLY COL LGF CAL LGF CAL						•		
B3F1								
B3F1 CDGTRA B959 CXFTR B958 ARK B3F2 CDUTR B958 CXLGTR B958 ARK B3F3 CDSTR B958 CXLGTR B958 ARK B3F3 ARK B3F5 ARK B								
B3F2								
B3F3								
B3F4								
B3F5								
B3F6								
B3F7								
B3F9						•		
B3F9								
B3FA CXUTR B982 XGR BF CM B3FB CXSTR B983 FLOGR BF CM B3FC CEXTR B984 LLGCR C00 LARL C01 LGFI C02 BRASL C06 XIHF C01 LGFI C02 MIHF C02 XIHF								
B3FB CXSTR B983								-
B3FC CEXTR B984								
B3FD								
BBFE EXTR B986 MLGR C04 BRCL BBFF RRXTR B987 DLGR C05 BRASL B6 STCTL B988 ALCGR C06 XIHF B990 LCTL B989 SLBGR C07 XILF B990 LTGR B980 EPSW C09 IILF B992 LTGR B98E IDTE C0A MIHF B902 LTGR B98F CDTE C0B MILF B995 LURAG B991 TRTO C0D OILF B905 LURAG B991 TRTO C0D OILF B905 LURAG B991 TRTO C0D OILF B906 LGBR B992 TROT C0E LLIHF B908 AGR B994 LLCR C20 MSGFI B908 AGR B994 LLCR C20 MSGFI B909 SGR B995 LLHR C21 MSFI B908 AGR B995 LLHR C21 MSFI B908 AGR B996 MIRR C24 SLGFI B908 SLGR B997 DLR C25 SLFI B908 SLGR B997 DLR C25 SLFI B908 B998 ALCR C28 AGFI B909 B909 BSAR B999 ESAR C29 AFI B906 EREGG B990 ESAR C29 AFI B901 LNGFR B991 ESAR C22 CGFI B911 LNGFR B992 FTF C22 CLFI B914 LGFR B994 ESAR C25 CLFI B915 LLGFR B994 ESAR C25 CLFI B991 SGFR B994 ERBM C45 LHRL C46 LLGHRL B916 LLGFR B946 RRBM C45 LHRL B916 LLGFR B946 RRBM C46 LLGHRL B916 SGFR B946 CU14 C48 LGRL STGRL B916 SGFR B946 STGRL C46 LLGFRL B916 SGFR B958 CU24 C48 STGRL C46 LLGFRL B916 SGFR B996 SRSTU C47 STRL C47 STRL C46 LGFRL C47 STRL C47								
B3FF RRXTR B997 DLGR C05 BRASL B6 STCTL B989 SLBGR C06 XIHF B7 LCTL B999 SLBGR C07 XILF B991 LNGR B991 ENW C09 IILF B992 LTGR B998 EDTE C08 IIHF B902 LTGR B998 EDTE C08 IIHF B903 LCGR B996 EDTE C08 NILF B903 LCGR B996 TRTT C0C C0HF B995 LURAG B991 TRTO C0D C0HF B906 LGBR B992 TROT C0E LLIHF B907 LGHR B993 TROO C0F LILIF B908 AGR B994 LLCR C20 MSGF B998 LLHR C21 MSGF B909 SGR B995 LLHR C21 MSGF B909 SGR B995 LLHR C24 SLGF B900 SGR B996 MLR C24 SLGF B900 DSGR B997 DLR C25 SLF B906 B996 MLR C24 SLGF B906 B996 B998 ALCR C28 AGF B906 B997 DLR C25 SLF B906 B998 ESAIR C29 AF B906 B996 B998 ESAIR C29 AF B906 B996 B998 ESAIR C28 ALF B906 B996 B998 ESAIR C28 ALF B906 B991 ESA C20 CGF B913 LCGFR B995 B996 PT C2D CF B912 LTGFR B996 B997 DLR C25 CLF B914 LGFR B996 B996 PT C2D CF B915 LURGR B996 PT C2D CF B916 LLGFR B996 RB996 RB996 C44 LGHRL B916 LLGFR B996 RB996 RB996 RB996 C45 LHRL B916 LLGFR B996 RB996 C46 LLGHRL B916 C46 LGHRL B916 C47 STHRL B916 C48 LGFR B996 C44 C48 LGFR B916 C46 LGFR B996 C47 C48 LGFR B916 C47 STHRL C48 LGFR B916 C47 C48 LGFR B916 C47 C48 LGFR B916 C47 STHRL C48 LGFR B916 C47 C48 LGF								
B7	B3FF	RRXTR						BRASL
B900 LPGR B98A CSPG C08 IIHF B901 LNGR B98D EPSW C09 IILF B902 LTGR B98E IDTE C0A NIHF B903 LCGR B98F CRDTE C0B NILF B904 LGR B991 TRTO C0C OIHF B906 LGBR B991 TRTO C0D OILF B906 LGBR B992 TROT C0E LLIHF B907 LGHR B993 TROO C0F LLILF B908 AGR B994 LLCR C20 MSGFI B909 SGR B995 LLHR C21 MSGFI B909 SGR B995 LLHR C24 SLGFI B900 SGR B995 LLHR C24 SLGFI B900 MSGR B996 MLR C22 SSEFI B900 MSGR B998 ALCR	B6	STCTL		B988	ALCGR		C06	XIHF
B901	B7	LCTL		B989	SLBGR		C07	XILF
B902	B900	LPGR		B98A	CSPG		C08	IIHF
B903 LCGR B98F CRDTE COB NILF B904 LGR B990 TRTT COC OIHF B905 LURAG B991 TRTO COD OILF B906 LGBR B992 TROT COE LLIHF B906 LGHR B993 TROO COF LLILF B908 AGR B994 LLCR C20 MSGFI B909 SGR B995 LLHR C21 MSGFI B900 SGR B996 MLR C24 SLGFI B900 MSGR B996 MLR C25 SLFI B900 MSGR B998 ALCR C28 AGFI B900 DSGR B999 SLBR C29 AFI B90F LRVGR B998 ESAIR C2A ALGFI B910 LPGFR B990 ESEA C2C CGFI B912 LTGFR B991 T	B901	LNGR		B98D	EPSW		C09	IILF
B904	B902	LTGR		B98E	IDTE		C0A	NIHF
B905								
B906 LGBR B992 TROT COE LLIHF B907 LGHR B993 TROO COF LLILF B908 AGR B994 LLCR C20 MSGFI B909 SGR B995 LLHR C21 MSFI B908 ALGR B996 MLR C24 SLGFI B900 MSGR B998 ALCR C28 AGFI B900 DSGR B999 SLBR C29 AFI B90F LRVGR B998 ESAIR C28 ALFI B90F LRVGR B99B ESAIR C2B ALFI B910 LPGFR B99B ESAIR C2C CGFI B911 LNGFR B99B PTI C2D CFI B912 LTGFR B99F SSAIR C2E CLGFI B914 LGFR B941 TPEI C2F CLFI B916 LLGFR B9AE <								
B907 LGHR B993 TROO COF LLILF B908 AGR B994 LLCR C20 MSGFI B909 SGR B995 LLHR C21 MSFI B90A ALGR B996 MLR C24 SLGFI B90C MSGR B997 DLR C25 SLFI B90D DSGR B998 ALCR C28 AGFI B90F LRVGR B998 SLBR C29 AFI B90F LRVGR B998 SEAIR C28 ALFI B90F LPGFR B99B ESAIR C28 ALFI B911 LNGFR B99B ESEA C2C CGFI B911 LNGFR B99F PTI C2D CFI B914 LGFR B99A PTF C42 LLHRL B916 LLGFR B9AA LPTEA C44 LGHRL B917 LLGTR B9AE <td< td=""><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></td<>								
B908								
B909 SGR B995 LLHR C21 MSFI B90A ALGR B996 MLR C24 SLGFI B90C MSGR B997 DLR C25 SLFI B90C MSGR B998 ALCR C28 AGFI B90D DSGR B999 SLBR C29 AFI B90F LRVGR B998 EPAIR C2A ALGFI B90F LRVGR B99B ESAIR C2B ALFI B910 LPGFR B99B ESAIR C2C CGFI B911 LNGFR B99B PTI C2D CFI B912 LTGFR B99F SSAIR C2E CLGFI B914 LGFR B941 TPEI C2F CLFI B916 LLGFR B9AA LPTEA C44 LGHRL B917 LLGTR B9AC RBM C45 LHRL B919 SGFR B9AF <								
B90A ALGR B996 MLR C24 SLGFI B90B SLGR B997 DLR C25 SLFI B90C MSGR B998 ALCR C28 AGFI B90D DSGR B999 SLBR C29 AFI B90F LRVGR B998 ESAIR C2A ALGFI B910 LPGFR B99B ESAIR C2C CGFI B911 LNGFR B99B ESAIR C2C CGFI B911 LNGFR B99F PTI C2D CFI B913 LCGFR B99F SSAIR C2E CLGFI B914 LGFR B9A1 TPEI C2F CLFI B914 LGFR B9A2 PTF C42 LLHRL B917 LLGTR B9A2 RRBM C45 LHRL B918 AGFR B9AF PRMF C47 STHRL B918 AGFR B9AF <								
B90B SLGR B997 DLR C25 SLFI B90C MSGR B998 B999 SLBR C28 AGFI B90E EREGG B999 SLBR C29 AFI B90E EREGG B99A EPAIR C2A ALGFI B90F ESAIR C2B ALFI B991 LCB C2B ALFI B910 LPGFR B99B ESAIR C2C CGFI CGFI B991 ESEA C2C CGFI CGFI B99F SSAIR C2E CLGFI C2E CLGFI C2E CLGFI C2E CLGFI C2F CLFI C2F CLFI C2F CLFI C2F CLFI C2F CLFI C2F CLFI C2E CLGFI C2F CLFI								
B90C MSGR B998 ALCR C28 AGFI B90D DSGR B999 SLBR C29 AFI B90F EREGG B99A EPAIR C2A ALGFI B90F LRVGR B99B ESAIR C2B ALFI B910 LPGFR B99B ESAIR C2C CGFI B911 LNGFR B99E PTI C2D CFI B912 LTGFR B99F SSAIR C2E CLGFI B913 LCGFR B99A PTF C42 CLHR B914 LGFR B9A2 PTF C42 LLHRL B916 LLGFR B9A2 PTF C42 LLHRL B917 LLGTR B9AC IBBM C45 LHRL B918 AGFR B9AC IBBM C46 LLGHRL B919 SGFR B9BF CV14 C48 LGRL B910 CV24 C4B								
B90D DSGR B999 SLBR C29 AFI B90E EREGG B99A EPAIR C2A ALGFI B90F LRVGR B99B ESAIR C2B ALFI B910 LPGFR B99D ESEA C2C CGFI B911 LNGFR B99E PTI C2D CFI B912 LTGFR B99F SSAIR C2E CLGFI B913 LCGFR B99F SSAIR C2E CLGFI B914 LGFR B9A2 PTF C42 LLHRL B916 LLGFR B9A2 PTF C42 LLHRL B918 AGFR B9AC IBBM C45 LHRL B918 AGFR B9AF PFMF C47 STHRL B91A ALGFR B9B0 CU14 C48 LGRL B91B BLGFR B9B1 CU24 C4B STGRL B91C MSGFR B9B2								
B90E EREGG B99A EPAIR C2A ALGFI B90F LRVGR B99B ESAIR C2B ALFI B910 LPGFR B99B ESAIR C2C CGFI B911 LNGFR B99E PTI C2D CFI B913 LCGFR B99F SSAIR C2E CLGFI B913 LCGFR B9A1 TPEI C2F CLFI B914 LGFR B9A2 PTF C42 LLHRL B917 LLGTR B9A2 LPTEA C44 LGHRL B917 LLGTR B9AC IRBM C45 LHRL B918 AGFR B9AF RRBM C46 LIGHRL B918 BJGFR B9B0 CU14 C48 LGRL B910 MSGFR B9B1 CU24 C4B STGRL B910 DSGFR B9B2 CU41 C4C LGFRL B91F KMAC B9BD <td></td> <td></td> <td></td> <td></td> <td>-</td> <td></td> <td></td> <td></td>					-			
B90F LRVGR B99B ESAIR C2B ALFI B910 LPGFR B99D ESEA C2C CGFI B911 LNGFR B99E PTI C2D CFI B912 LTGFR B99F SSAIR C2E CLGFI B913 LGFR B9A1 TPEI C2F CLFI B916 LLGFR B9A2 PTF C42 LLHRL B916 LLGFR B9A2 RTBM C45 LHRL B917 LLGTR B9AE RRBM C45 LHRL B918 SGFR B9AE RRBM C46 LLGHRL B919 SGFR B9B0 CU14 C48 LGRL B916 LU24 C4B STGRL B9BC CU41 C4C LGFRL B910 DSGFR B9B2 CU41 C4C LGFRL B9BC CU41 C4C LGFRL B91F KMAC B9BD TTTRE								
B910 LPGFR B99D ESEA C2C CGFI B911 LNGFR B99E PTI C2D CFI B912 LTGFR B99F SSAIR C2E CLGFI B913 LCGFR B99F SSAIR C2E CLFI B914 LGFR B9A2 PTF C42 LLHRL B916 LLGFR B9AC IBBM C45 LHRL B918 AGFR B9AC IBBM C45 LHRL B918 AGFR B9AF PFMF C47 STHRL B91A ALGFR B9BO CU14 C48 LGRL B91A ALGFR B9BO CU14 C48 LGRL B91C MSGFR B9B1 CU24 C4B STGRL B91C MSGFR B9B2 CU41 C4C LGFRL B91E KMAC B9BD TRTRE C4E LLGFRL B91F LRVR B9BE								
B911					-			
B912								
B913	D040	LTOED		D00F	00410		005	
B914								
B916 LLGFR B9AA LPTEA C44 LGHRL B917 LLGTR B9AC IRBM C45 LHRL B918 AGFR B9AE RRBM C46 LLGHRL B919 SGFR B9AF PFMF C47 STHRL B918 SLGFR B9B0 CU14 C48 LGRL B91C MSGFR B9B1 CU24 C4B STGRL B91C MSGFR B9B2 CU41 C4C LGFRL B91D DSGFR B9B3 CU42 C4D LRL B91E KMAC B9BD TRTRE C4E LLGFRL B91F LRVR B9BE SRSTU C4F STRL			•					
B917 LLGTR ■ B9AC IRBM C45 LHRL B918 AGFR B9AE RRBM C46 LLGHRL B919 SGFR B9AF PFMF C47 STHRL B91A ALGFR B9B0 CU14 C48 LGRL B91B SLGFR B9B1 CU24 C4B STGRL B91D DSGFR B9B2 CU41 C4C LGFRL B91E KMAC B9BD TRTRE C4E LLGFRL B91F LRVR B9BE SRSTU C4F STRL								
B918 AGFR B9AE RRBM C46 LLGHRL B919 SGFR B9AF PFMF C47 STHRL B91A ALGFR B9B0 CU14 C48 LGRL B91B SLGFR B9B1 CU24 C4B STGRL B91C MSGFR B9B2 CU41 C4C LGFRL B91D DSGFR B9B3 CU42 C4D LRL B91E KMAC B9BD TRTRE C4E LLGFRL B91F LRVR B9BE SRSTU C4F STRL							-	
B919 SGFR B9AF PFMF C47 STHRL B91A ALGFR B9B0 CU14 C48 LGRL B91B SLGFR B9B1 CU24 C4B STGRL B91C MSGFR B9B2 CU41 C4C LGFRL B91D DSGFR B983 CU42 C4D LRL B91E KMAC B9BD TRTRE C4E LLGFRL B91F LRVR B9BE SRSTU C4F STRL			•					
B91A ALGFR B9B0 CU14 C48 LGRL B91B SLGFR B9B1 CU24 C4B STGRL B91C MSGFR B9B2 CU41 C4C LGFRL B91D DSGFR B9B3 CU42 C4D LRL B91E KMAC B9BD TRTRE C4E LLGFRL B91F LRVR B9BE SRSTU C4F STRL								
B91B SLGFR B9B1 CU24 C4B STGRL B91C MSGFR B9B2 CU41 C4C LGFRL B91D DSGFR B9B3 CU42 C4D LRL B91E KMAC B9BD TRTRE C4E LLGFRL B91F LRVR B9BE SRSTU C4F STRL								
B91C MSGFR B9B2 CU41 C4C LGFRL B91D DSGFR B9B3 CU42 C4D LRL B91E KMAC B9BD TRTRE C4E LLGFRL B91F LRVR B9BE SRSTU C4F STRL								
B91D DSGFR B9B3 CU42 C4D LRL B91E KMAC B9BD TRTRE C4E LLGFRL B91F LRVR B9BE SRSTU C4F STRL				B9B2				
B91F LRVR B9BE SRSTU C4F STRL	B91D			B9B3				
B920 CGR B9BF TRTE C5 BPRP				-			-	-
	B920	CGR		B9BF	TRTE		C5	BPRP

OpCode Mnemonic OpCode Mnemonic OpC C60 EXRL E331 CLGF E3C C62 PFDRL E332 LTGF E50 C64 CCULD E324 CCULD E50	
C62 PFDRL E332 LTGF E500	E CLHE
C64 CGHRL E334 CGH E50	
C65 CHRL E336 PFD E502	
C66 CLGHRL E338 AGH E501 C67 CLHRL E339 SGH E501	
C68 CGRL E33A LLZRGF E544	
C6A CLGRL E33B LZRF E544	
C6C CGFRL ■ E33C MGH E540	
C6D CRL E33E STRV E554	
C6E CLGFRL E33F STRVH E555	
C6F CLRL E346 BCTG E558	
C7 BPP ■ E347 BIC E555	
C80 MVCOS E348 LLGFSG E550	C CHSI
C81 ECTG E349 STGSC E55I	CLFHSI
C82 CSST E34C LGG E560	
C84 LPD E34D LGSC E56	
C85 LPDG E350 STY E634	
CC6 BRCTH E351 MSY E638	
CC8 AIH E353 MSC E637	
CCA ALSIH E354 NY E630	
CCB ALSIHN E355 CLY E631 CCD CIH E356 OY E631	
CCF CLIH E357 XY E649	
D0 TRTR E358 LY E650	
D1 MVN E359 CY E652	
D2 MVC E35A AY E658	
D3 MVZ E35B SY E655	
D4 NC E35C MFY E65A	A VCVDG
D5 CLC E35E ALY E65I	3 VPSOP
D6 OC E35F SLY E65I	
D7 XC E370 STHY E67	
D9 MVCK E371 LAY E673	
DA MVCP E372 STCY E677	
DB MVCS E373 ICY E678	
DC TR E375 LAEY E679	
DD TRT	
DE ED E377 LGB E671 DF EDMK E378 LHY E671	
E1 PKU E379 CHY E700	
E2 UNPKU E37A AHY E70	
E302 LTG E37B SHY E702	
E303 LRAG E37C MHY E700	
E304 LG E380 NG E704	
E306 CVBY E381 OG E709	VLREP VLREP
E308 AG E382 XG E706	S VL
E309 SG E383 MSGC E703	
E30A ALG E384 MG E708	
E30B SLG E385 LGAT E709	
E30C MSG E386 MLG E70/	
E30D DSG E387 DLG E701 E30E CVBG E388 ALCG E701	
E30F LRVG E389 SLBG E712	
E312 LT E38E STPQ E713	
E313 LRAY E38F LPQ E71/	
E314 LGF E390 LLGC E718	
E315 LGH E391 LLGH E72:	
E316 LLGF E394 LLC E722	
E317 LLGT E395 LLH E72	
E318 AGF E396 ML E730	
E319 SGF E397 DL E730	
E31A ALGF E398 ALC E736	
E31B SLGF E399 SLB E731	
E31C MSGF E39C LLGTAT E736 E31D DSGF ■ E39D LLGFAT E736	
E31D DSGF E39D LLGFAI E73F	
E31F LRVH E3C0 LBH E73I	
E320 CG E3C2 LLCH E740	
E321 CLG E3C3 STCH E74:	
E324 STG E3C4 LHH E742	
E325 NTSTG E3C6 LLHH E745	
E326 CVDY E3C7 STHH E744	
E32A LZRG E3C8 LFHAT E748	
E32E CVDG E3CA LFH E746	
E32F STRVG E3CB STFH E74/	
E330 CGF E3CD CHF E74	O VREP

E752 VCTZ E768 WFC E896 STMY E756 VLR E7CC VFSO E888 LMH E756 VSEG E7D4 VUPLL E898 LMH E760 VMRL E7D6 VUPL E888 LMH E760 VMRH E7D6 VUPL E800 STAMY E761 VMRH E7D70 VUPL EBD0 STAMY E762 VSUMG E7D8 VTM EBD0 STAMY E764 VSUM E7D8 VTM EBD0 STAK E765 VCKSM E7D8 VCC EBDF SLLK E766 VCKSM E7DF VLC EBE0 LOCH E768 WN E7E7 VFD EBE1 STOCG E768 WNO E7E7 VFM EBE6 LAOG E766 VX E7EA VFCH EBE7 LAX E760 VX E7EA VFC	OpCode	Mnemonic		OpCode	Mnemonic		OpCode	Mnemonic
E756 V.R E7CC VFSO E896 LMH E756 V.STR E7CE VFSO E898 LMY E761 WRL E7D6 VUPL EB8A LAMY E761 WMRL E7D6 VUPL EB0C TP E762 LYGP E7D8 VTM EBDD SLAK E766 VSUMG E7D9 VECL EBDF SLLK E766 VSUMG E7D8 VEC EBDF SLLK E766 VSUMG E7DE VLC EBE0 LOCFH E767 VSUMG E7DE VLC EBE0 LOCG E768 VN E7E3 VFA EBE1 STOCFH E768 VN E7E3 VFA EBE3 STOCG E760 VX E7E8 VFCH EBEA LANG E760 VX E7E8 VFCH EBEA LANG E770 VSELU E7E7 VFM	E750	VPOPCT		E7CA	WFK		EB8F	CLCLU
E756 V.R E7CC VFSO E896 LMH E756 V.STR E7CE VFSO E898 LMY E761 WRL E7D6 VUPL EB8A LAMY E761 WMRL E7D6 VUPL EB0C TP E762 LYGP E7D8 VTM EBDD SLAK E766 VSUMG E7D9 VECL EBDF SLLK E766 VSUMG E7D8 VEC EBDF SLLK E766 VSUMG E7DE VLC EBE0 LOCFH E767 VSUMG E7DE VLC EBE0 LOCG E768 VN E7E3 VFA EBE1 STOCFH E768 VN E7E3 VFA EBE3 STOCG E760 VX E7E8 VFCH EBEA LANG E760 VX E7E8 VFCH EBEA LANG E770 VSELU E7E7 VFM	E752	VCTZ		E7CB	WFC		EB90	STMY
E756 V.ISTR E7CE VSEG E7D4 VUPLL EB8A LAMY E756 V.WSEG E7D6 VUPLH EB8A STAMY E760 VMRIL E7D6 VUPLH EB8D STAMY E762 V.UMGP E7D8 VTM EBBD STAMY E764 VSUM E7D8 VTM EBBD STAMY E766 VCKSM E7D8 VCC EBBC SCILK E766 VCKSM E7D8 VCC EBBD SSILK E768 VN E7DF VLC EB60 LOCPH E768 VN E7E2 VFS EBE2 LOG6 E768 VN E7E7 VPD EBE4 LANG E768 VN E7E8 VFD EBE4 LANG E768 VX E7EA VFCHE EBE3 LAOG E769 VX E7EA VFCHE EBE8 LAA E760								
E75C VISTR E7D4 VUPLL E898 LAMY E761 VMRL E7D5 VUPL E800 TP E761 VMRH E7D6 VUPL EBDC STAMY E762 VLVGP E7D8 VTM EBDD SLAK E766 VSUMG E7D8 VEC EBDF SLLK E766 VSUMG E7DE VLC EBED SLLK E767 VSUMG E7DE VLC EBED LOCFH E768 VN E7E2 VFA EBE1 STOCFH E768 VN E7E3 VFA EBE3 STOCG E768 VN E7E8 VFCD EBE7 LANG E760 VIX E7E8 VFCH EBE8 LAAG E760 VIX E7E8 VFCH EBE8 LAAG E774 VSL E7E9 VFMA EBF2 LAC E7773 VSERLU E7F6 <								
E756 VSEG E7D5 VUPLH E800 TP E761 VMRH E7D6 VUPH EBO0 TP E761 VMRH E7D6 VUPH EBD0 SRAK E762 VSUMG E7D8 VTM EBDD SLAK E765 VSUMG E7D8 VEC EBDD SLAK E766 VSKSM E7D8 VEC EBDD SLLK E768 VN E7D8 VLC EBD0 SLLK E768 VN E7E2 VFD EBE1 STOCG E768 VN E7E3 VFD EBE4 LANG E768 VN E7E8 VFCD EBE7 LAXG E768 VN E7E8 VFCD EBE3 LANG E769 VX E7E8 VFCH EBE8 LANG E772 VSEIJW E7F8 VFCH EBE8 LAX E773 VSLDB E7F8 VFMAX <td></td> <td></td> <td></td> <td></td> <td></td> <td>1</td> <td></td> <td></td>						1		
E760 WMRH E761 WMRH E762 WUNGP E7D6 YUPH EBDC SRAK E762 WLVGP E7D8 VTM EBDC SRAK E765 YUSUMG E7D8 VCC EBDF SILK E766 YUSUMG E7D8 VCC EBGD SILK E767 YSUMG E7E6 VCC EBE1 STOCCH E768 VN E7E6 YER EBE2 LOCG E76B VN E7E5 VFD EBE3 LOCG E76C VNX E7E8 VFC EBE3 LAGG E76E VNN E7E8 VFCHE EBE8 LAAG E76E VNN E7E8 VFCHE EBE8 LAAG E776 VSLIV E7E7 VYFM EBE6 LAOG E773 VERILU E7FE VYFOHE EBE7 LAX E773 VERLU E7FE VAVG EBF7 LAX								
E761 VMRH E7762 VLVGP E708 VTM EBDD SRAK E764 VSUMG E708 VTM EBDD SRAK E766 VSUMG E708 VCC EBDF SLLK E7676 VSUMG E70F VLC EBDD SLK E768 VN E72E VFD EBE1 LOCFH E768 VN E72E VFD EBE2 LOCG E760 VX E76B VX E7E8 VFCE EBE7 LANG E76F VX E76B VX E7E8 VFCH EBE8 LANG E76F VX E76F VCC E7E7 VFM EBE8 LANG E772 VERIM E772 VFM EBE8 LAAG E7E8 VFCH EBE8 LAAU E7E7 VSL E7FE VFM EBF6 LAX E7FE VFM EBF6 LAX E7FE VFM EBF6 LAX<	-						-	
E768								
E766	E761	VMRH		E7D7	VUPH		EBDC	SRAK
Ē756 VSUMG E7DB VEC EBDF SLLK Ē767 VSUMQ E7DF VLC EBBO LOCFH Ē768 VN E7EZ VFS EBEZ LOCG Ē768 VN E7EZ VFS EBEZ LOCG Ē768 VN E7ES VFA EBE3 STOCCH Ē768 VN E7ES VFA EBE3 LOCG Ē760 VX E7ES VFD EBE4 LANG Ē760 VN E7ER VFCH EBE8 LAAG Ē776 VSL E7ER VFCH EBE8 LAAG Ē773 VERILU E7FE VFHMN EBF2 LOC Ē7774 VSL E7FT VAVG EBF8 LAA Ē7775 VSLB E7FT VAC EBF8 LAA Ē7770 VSRL E7FT VSCBI EFF5 VSCBI EBF7 LAX Ē7770 VSRLB<	E762	VLVGP		E7D8	VTM		EBDD	SLAK
Ē756 VSUMG E7DB VEC EBDF SLLK Ē767 VSUMQ E7DF VLC EBBO LOCFH Ē768 VN E7EZ VFS EBEZ LOCG Ē768 VN E7EZ VFS EBEZ LOCG Ē768 VN E7ES VFA EBE3 STOCCH Ē768 VN E7ES VFA EBE3 LOCG Ē760 VX E7ES VFD EBE4 LANG Ē760 VN E7ER VFCH EBE8 LAAG Ē776 VSL E7ER VFCH EBE8 LAAG Ē773 VERILU E7FE VFHMN EBF2 LOC Ē7774 VSL E7FT VAVG EBF8 LAA Ē7775 VSLB E7FT VAC EBF8 LAA Ē7770 VSRL E7FT VSCBI EFF5 VSCBI EBF7 LAX Ē7770 VSRLB<	E764	VSUM		E7D9	VECL		EBDE	SRLK
E766								
E768								
E768 VN E7E2 VFS EBE2 LOCG E768 VO E7E3 VFA EBE3 STOCG E760 VN E7E5 VFD EBE4 LANG E760 VX E7E8 VFOH EBE6 LAOG E766 VIX E7E8 VFCH EBE7 LAXG E766 VIX E7E8 VFCH EBE8 LAALG E776 VCO VESILV E7E8 VFCH EBE8 LAALG E772 VERIM E9F8 TO CE7F1 VAVG EBF3 STOC E773 VSLB E7F9 VAVG EBF8 LAA LAA E775 VSLB E7F9 VAVG EBF8 LAA E776 VSLB E7F9 VAVG EBF8 LAA E770 VSRLB E7F7 VSCBI EBFA LAA E771 VSRLB E7F9 VCHQ EC44 BRXLG								
E768 VNC F76A VO E76B F76A VO E76B VO E76F VFD EBE4 LANG EP6B LAOG EP76D VX E76B VYC EBE7 VFCH EBE8 LAAG EP6B LAXG EP76F VX E76B VYCHE EBE8 LAAG EP6F VFCH EBE8 LAAG EP76F VX E77B VYCHH EBE8 LAAG EP76F VYCH EBE7 LAX EP76F VYCH EP6F LAA LBF7 LOC EP76 VYMQL EBF7 LAX EP778 VSL EP779 VAVGL EBF7 LAX EP778 VSL EP779 VAVGL EBF7 LAX EP777 VSL EP778 VSC EP779 VAVGL EBF7 LAX EP777 VSR EC42 LOCHI EVAG EP777 VSR EC42 LOCHI EVAG EP779 VCH EC45 BRXLG EP779 VML EC46 LOC								
E76A VO F76B VNO E76C F76D E8E4 LANG E76C F76D VN E76C VNX E76E VNX E76E VNN E76E VNN E76F VOC E772 VENIM E772 VERIM E772 VERIM E772 VERIM E774 VSL E776 VOX E771 VACC E8F6 LAO LAO E777 VSLDB E777 VSRD E778 VCFQ E244 BRXHG E779 VCHL E045 BRXHG E779 VCHL E046 LOCHII								
Ē76B VNO F7E7F VFM ĒBĒĞ LAOĞ Ē76C VNX F7E8 VFCE ĒBĒĞ LAVĞ Ē76F VN F7EA VFCH ĒBĒBA LAAĞ Ē770 VESLV F7EB VFCH ĒBĒBA LAALĞ Ē7770 VESLU F7FF VFMAX ĒBF3 STOC Ē773 VSLB Ē7F7 VAQC ĒBF6 LAO Ē7775 VSLB Ē7F7 VSLB Ē7F7 VAQC ĒBFA LAA Ē7778 VSLBB Ē7F7 VSCBI ĒBFA LAA Ē7778 VSLB Ē7F7 VS CBI ĒBFA LAA Ē7770 VSRL Ē7F7 VS ĒC42 LOCHI Ē7770 VSRL Ē7F7 VS CBI ĒBFA LAA Ē7770 VSRL Ē7F8 VCHL ĒC45 BRXLG Ē7772 VSRLB Ē7F8 VCHL ĒC46 LOCHII Ē								
E76C VNX F76D VX E76E E76E VYC E76E E76E VYC E76E VYCHE EBE8 LAXG E76E VFCHE EBE8 LAXG E76E VFCHE EBE8 LAXG E76E VFCHE EBE8 LAXG EF6F VFCHE EBE8 LAXG EF7E VFMMX EBF3 LAXG EF7E VFMMX EBF3 LAXG EF7E VFMMX EBF3 LAXG EF7E VAYGL EBF4 LAX EF7E VAYGL EBF4 LAX EF7E VAYGL EBF5 LAXG EBF4 LAXG EBF4 LAX EF7E VAYGL EBF6 LAXG EBF4 LAXG EBF4	E76A			E7E5				
E76D VX L76E VRN E76E C76E VRN E77E VFCH EBEA LAAG EAG LAAG E77E VFCH EBF2 LOC C77C VERILV E77E VFMIN EBF3 STOC EFFE LAA EBF3 STOC EFFT VAVGL EBF6 LAO LAO EFFT VAVG EBF7 LAX EFF7 VAVG EBF7 LAX EFF7 VAVG EBF7 LAX EFF8 LAA LAA LAA EFF8 LAA LAA LAA LAA LAA LAA LAA LAA LAA	E76B	VNO		E7E7			EBE6	LAOG
E76D VX L76E VRN E76E C76E VRN E77E VFCH EBEA LAAG EAG LAAG E77E VFCH EBF2 LOC C77C VERILV E77E VFMIN EBF3 STOC EFFE LAA EBF3 STOC EFFT VAVGL EBF6 LAO LAO EFFT VAVG EBF7 LAX EFF7 VAVG EBF7 LAX EFF7 VAVG EBF7 LAX EFF8 LAA LAA LAA EFF8 LAA LAA LAA LAA LAA LAA LAA LAA LAA	E76C	VNX		E7E8	VFCE		EBE7	LAXG
F76E							FRF8	
F76F								
E770			_					
E772						1		
E773 VERLLV E7F1 VACC EBF6 LAO E774 VSL E7F2 VAVG EBF7 LAX E775 VSLB E7F3 VA EBF8 LAA E778 VESRIV E7F VSCBI EBFA LAA E778 VESRIV E7F VSCQ EC42 LOCHI E777 VSRL E7F9 VCHL EC45 BRXLG E77D VSRL E7F9 VCHL EC45 BRXLG E77D VSRA E7FC VMNL EC46 LOCGHI E77F VSRAB E7FD VMXL EC51 RISBLG E77F VSRAB E7FD VMXL EC51 RISBLG E780 VFEE E7FF VMX EC55 RISBG E781 VFENE E7FF VMX EC56 RISBG E781 VFENE E7FF VMX EC56 RISBG E781 VFENE E8 </td <td></td> <td></td> <td></td> <td></td> <td></td> <td>1</td> <td></td> <td></td>						1		
E774 VSL E7F2 VAVG EBF8 LAX E775 VSLB E7F3 VA EBF8 LAA E777 VSLDB E7F5 VSCBI EBFA LAAL E778 VESRIV E7F7 VS EC42 LOCHI E77A VESRAV E7F9 VCHL EC45 BRXLG E77D VSRLB E7FB VCH EC45 BRXLG E77D VSRLB E7FB VCH EC46 LOCGHI E77E VSRAB E7FC VMNL EC51 RISBLG E77E VSRAB E7FC VMNL EC51 RISBLG E781 VFEE E7FE VMN EC54 RNSBG E781 VFENE E7FF VMN EC55 RISBG E781 VFENE E7FF VMN EC56 ROSBG E782 VFAE E8 MVCIN EC56 ROSBG E785 VPSERM E8						1		
E775 VSLB E775 VA EBFA LAA E777 VSLDB E7F5 VSCBI EBFA LAAL E778 VESRLV E7F7 VS EC42 LOCHI E77A VESRAV E7F8 VCEQ EC44 BRXHG E77D VSRLB E7F9 VCH EC45 BRXLG E77D VSRAB E7FC VMNL EC46 LOCGHII E77E VSRAB E7FD VMXL EC51 RISBLG E77E VSRAB E7FD VMXL EC54 LOCGHII E77E VSRAB E7FD VMXL EC54 RISBLG E781 VFENE E7FE VMX EC54 RISBLG E781 VFENE E7FF VMX EC54 RISBLG E781 VFENE E8 MVCIN EC54 RISBLG E782 VFAE E8 MVCIN EC54 RISBLG E784 VPERM	E773	VERLLV		E7F1	VACC	1	EBF6	
E777 VSLDB E7F5 VSCBI EBFA LAAL E778 VESRIV E7F7 VS EC42 LOCHI E77C VSRL E7F8 VCEQ EC44 BRXHG E7TC VSRL E7F9 VCHL EC45 BRXLG E7TD VSRLB E7FB VCHL EC45 BRXLG E77T VSRAB E7FD VMNL EC4E LOCHII E77F VSRAB E7FD VMXL EC51 RISBLG E780 VFEE E7FE VMX EC54 RISBLG E781 VFENE E7FF VMX EC55 RISBG E782 VFAE E8 MVCIN EC56 ROSBG E784 VPDI E9 PKA EC57 RXSBG E784 VPDI E9 PKA EC57 RXSBG E785 VSEL E80A LMG EC59 RISBHG E780 VFMS E80A	E774				VAVG	1		
E777 VSLDB E7F5 VSCBI EBFA LAAL E778 VESRIV E7F7 VS EC42 LOCHI E77C VSRL E7F8 VCEQ EC44 BRXHG E7TC VSRL E7F9 VCHL EC45 BRXLG E7TD VSRLB E7FB VCHL EC45 BRXLG E77T VSRAB E7FD VMNL EC4E LOCHII E77F VSRAB E7FD VMXL EC51 RISBLG E780 VFEE E7FE VMX EC54 RISBLG E781 VFENE E7FF VMX EC55 RISBG E782 VFAE E8 MVCIN EC56 ROSBG E784 VPDI E9 PKA EC57 RXSBG E784 VPDI E9 PKA EC57 RXSBG E785 VSEL E80A LMG EC59 RISBHG E780 VFMS E80A		VSLB			VA	1		
E778 VESRIV E7F7 VS EC42 LOCHI E77A VESRAV E7F8 VCCQ EC44 BRXHG E77D VSRLB E7FB VCH EC45 BRXLG E77D VSRAB E7FC VMNL EC46 LOCGHI E77F VSRAB E7FC VMNL EC4E LOCHHI E780 VFEE E7FD VMXL EC51 RISBLG E780 VFENE E7FF VMX EC55 RISBG E781 VFENE E7FF VMX EC55 RISBG E781 VFENE E7FF VMX EC56 ROSBG E782 VFAE E8 MVCIN EC56 ROSBG E784 VPDI E9 PKA EC56 ROSBG E784 VPDI E9 PKA EC59 RISBGN E784 VPERM EB0A SRAG EC66 CGRJ E786 VPERM E						1		
E77A VESRAV E7F8 VCEQ EC44 BRXHG E77C VSRIL E7F9 VCHL EC45 BRXLG E77D VSRAB E7FC VMNL EC4E LOCHHI E77F VSRAB E7FC VMNL EC4E LOCHHI E77F VSRAB E7FD VMNL EC51 RISBLG E780 VFEE E7FE VMN EC54 RNSBG E781 VFENE E7FF VMX EC55 RISBG E781 VFENE E8 MVCIN EC56 ROSBG E784 VPENE E8 MVCIN EC56 RYSBG E785 VBPERM EA UNPKA EC57 RXSBG E786 VPERM EB04 LMG EC57 RXSBG E786 VFBERM EB04 SRAG EC64 CGRJ E786 VFBS EB0C SRLG EC70 CGIT CLGIT E787				-				
E77C VSRL E7FB VCHL EC46 BRXLG E77D VSRAB E7FC VMNL EC46 LOCGHI E77F VSRAB E7FC VMNL EC4E LOCHHI E77F VSRAB E7FD VMXL EC51 RISBLG E780 VFEE E7FE VMX EC55 RISBG E781 VFENE E7FF VMX EC55 RISBG E782 VFAE E8 MVCIN EC56 ROSBG E784 VPDI E9 PKA EC57 RXSBG E785 VBPERM EA UNPKA EC59 RISBGN E78A VSTRC EB04 LMG EC59 RISBG E78A VSTRC EB0A SRAG EC66 CGRJ E78B VFMS EB0A SRLG EC70 CGIT E78E VFMS EB0C SRLG EC70 CGIT E79E VFMLS E						1		
E77D VSRAB E7FB VCH EC46 LOCGHI E77E VSRAB E7FC VMNL EC4E LOCHII E77B VSRAB E7FD VMXL EC51 RISBLG E780 VFEE E7FE VMN EC54 RNSBG E781 VFENE E7FF VMX EC55 RISBG E782 VFAE E8 MYCIN EC56 ROSBG E784 VPDI E9 PKA EC57 RXSBG E784 VPDI E9 PKA EC59 RISBGN E788 VBPERM EA UNPKA EC59 RISBGN E784 VPDI EB04 LMG EC50 RISBGN E786 VPERM EB04 LMG EC50 RISBGN E786 VPERM EB0C SRAG EC64 CGRJ E787 VPKA EB0F TRACG EC70 CGIT E795 VPKLS E								
E77E VSRAB E7FC VMNL EC4E LOCHHI E77F VSRAB E7FD VMXL EC51 RISBLG E780 VFEE E7FE VMN EC53 RISBG E781 VFENE E7FF VMX EC55 RISBG E782 VFAE E8 MVCIN EC56 ROSBG E784 VPDI E9 PKA EC57 RXSBG E785 VBPERM EA UNPKA EC57 RXSBG E785 VSTRC EB04 LMG EC53 RISBHG E786 VPERM EB04 LMG EC50 RISBHG E786 VFBRM EB0C SRLG EC66 CLGRJ E786 VFMS EB0C SRLG EC71 CLGIT E787 VPKLS EB14 CSY EC73 CLFIT E798 VPKLS EB14 CSY EC73 CLFIT E799 VPKMS <td< td=""><td>-</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></td<>	-							
E77F VSRAB E7FD VMXL EC51 RISBLG E780 VFEE E7FE VMX EC54 RNSBG E781 VFENE E7FF VMX EC55 RISBG E782 VFAE E8 MVCIN EC56 ROSBG E784 VPDI E9 PKA EC57 RXSBG E785 VBPERM EA UNPKA EC59 RISBLG E78A VSTRC EB04 LMG EC59 RISBLG E78A VSEL EB0A SRAG EC64 CGRJ E78B VFMS EB0C SRLG EC65 CLGTJ E78E VFMS EB0C SRLG EC70 CGIT E798 VFMS EB0F TRACG EC72 CIT E795 VPKLS EB14 CSY EC73 CLFIT E797 VPKS EB1C RLL EC77 CLRJ E797 VFNMA EB20 <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td>								
E780 VFEE E7FE VMN EC54 RNSBG E781 VFENE E7FF VMX EC55 RISBG E782 VFAE E8 MVCIN EC656 ROSBG E784 VPDI E9 PKA EC57 RXSBG E788 VBPERM EA UNPKA EC59 RISBGN E780 VSTRC EB04 LMG EC59 RISBGN E780 VSTRC EB04 LMG EC59 RISBGN E780 VSEL EB08 SLAG EC64 CGRJ E780 VSEL EB00 SLLG EC70 CGIT E781 VFMS EB0C SRLG EC70 CGIT E784 VPK EB0F TRACG EC72 CIT E795 VPKLS EB14 CS2Y EC73 CLFIT E797 VPKS EB1C RLLG EC76 CRJ E799 VFNMS EB1D <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td>								
E781 VFENE E7FF VMX EC55 RISBG E782 VFAE E8 MVCIN EC56 ROSBG E784 VPDI E9 PKA EC57 RXSBG E785 VBPERM EA UNPKA EC59 RISBGN E786 VSTRC EB04 LMG EC50 RISBHG E786 VFERM EB0A SRAG EC64 CGRJ E780 VSEL EB0B SLAG EC65 CLGRJ E787 VFMS EB0C SRLG EC70 CGIT E787 VFMA EB0D SLLG EC71 CLGIT E787 VPKLS EB14 CSY EC72 CIT E795 VPKLS EB14 CSY EC73 CLFIT E797 VPKS EB10 RLL EC77 CLRJ E798 VFNMA EB20 CLMY EC77 CLRJ E799 VFNMA EB20 <td>E77F</td> <td>VSRAB</td> <td></td> <td>E7FD</td> <td>VMXL</td> <td></td> <td>EC51</td> <td>RISBLG</td>	E77F	VSRAB		E7FD	VMXL		EC51	RISBLG
E782 VFAE E8 MVCIN EC56 ROSBG E784 VPDI E9 PKA EC57 RXSBG E785 VBPERM EA UNPKA EC59 RISBHG E78A VSTRC EB04 LMG EC5D RISBHG E78B VSEL EB0A SRAG EC64 CGRJ E78E VFMS EB0C SRLG EC70 CGIT E78F VFMS EB0D SLLG EC71 CLGIT E798F VFMA EB0D SLLG EC72 CIT E794 VPK EB0F TRACG EC72 CIT E795 VPKLS EB14 CSY EC73 CLFIT E795 VPKLS EB1C RLL EC76 CRJ E797 VPKS EB1C RLL EC76 CRJ E797 VPKMS EB1C LMH EC76 CLBJ E794 VML EB21	E780	VFEE		E7FE	VMN		EC54	RNSBG
E782 VFAE E8 MYCIN EC56 ROSBG E784 VPDI E9 PKA EC57 RXSBG E785 VBPERM EA UNPKA EC59 RISBGN E78A VSTRC EB04 LMG EC59 RISBGN E78B VSEL EB0A SRAG EC64 CGRJ E78D VSEL EB0A SRAG EC65 CLGRJ E78E VFMS EB0C SRLG EC70 CGIT E78F VFMA EB0D SLLG EC71 CLGIT E794 VPK EB0F TRACG EC72 CIT E795 VPKLS EB14 CSY EC73 CLFIT E795 VPKLS EB1C RLL EC76 CRJ E797 VPKS EB1C RLL EC76 CRJ E797 VPNMA EB20 CLMH EC7C CGIJ E794 VML EB23	E781	VFENE		E7FF	VMX		EC55	RISBG
E784 VPDI E9 PKA EC57 RXSBG E785 VBPERM EA UNPKA EC59 RISBGN E78A VSTRC EB04 LMG EC59 RISBGN E78C VPERM EB04 SRAG EC64 CGRJ E78C VPERM EB06 SRLG EC670 CGIT E78E VFMS EB0C SRLG EC70 CGIT E78F VFMA EB0D SLLG EC71 CLGIT E78F VFMA EB0F TRACG EC72 CIT E79F VFK EB0F TRACG EC72 CLT E79F VFNMS EB1C RLL EC76 CRJ E79F VFNMA EB20 CLMH EC77 CLRJ E749F VFNMA EB20 CLMH EC77 CLGJ E7A9 VML EB23 CLT EC72 CUJ E7A2 VML EB23	E782	VFAE		E8	MVCIN			ROSBG
E785 VBPERM EA UNPKA EC59 RISBGN E78A VSTRC EB04 LMG EC5D RISBHG E78C VPERM EB0A SRAG EC64 CGRJ E78D VSEL EB0B SLAG EC65 CLGRJ E78E VFMS EB0C SRLG EC70 CGIT E78F VFMA EB0D SLLG EC71 CLGIT E798 VPKLS EB14 CSY EC72 CIT E799 VPKLS EB1C RLLG EC76 CRJ E79E VFNMA EB20 CLMH EC77 CLRJ E79F VFNMA EB20 CLMH EC77 CLRJ E79F VFNMA EB20 CLMY EC7D CLGIJ E7A1 VMLH EB23 CLT EC7E CLJ E7A2 VML EB23 STCTG EC7E CLJ E7A3 VML EB25								
E78A VSTRC EB04 LMG EC5D RISBHG E78C VPERM EB0A SRAG EC64 CGRJ E78D VSEL EB0B SLAG EC64 CGRJ E78E VFMS EB0C SRLG EC70 CGIT E78F VFMA EB0D SLLG EC71 CLGIT E798 VPKLS EB14 CSY EC73 CLFIT E795 VPKLS EB1C RLLG EC76 CRJ E795 VPKUS EB1C RLLG EC76 CRJ E797 VPKS EB1C RLLG EC76 CRJ E798 VFNMS EB1C RLLG EC77 CLRJ E799F VFNMA EB20 CLMH EC77 CLRJ E741 VMLH EB21 CLT EC77 CLRJ E7A2 VML EB23 STCTG EC7F CLIJ E7A3 VMLD EB25								
E78C VPERM EBOA SRAG EC64 CGRJ E78E VSEL EBOB SLAG EC65 CLGRJ E78E VFMS EBOC SRLG EC70 CGIT E78F VFMA EBOD SLLG EC71 CLGIT E79F VPKL EBOF TRACG EC72 CIT E79F VPKLS EB14 CSY EC76 CRJ E797 VPKS EB1C RLLG EC76 CRJ E79F VFNMS EB1D RLL EC77 CLFI E79F VFNMA EB20 CLMH EC7C CLRJ E79F VFNMA EB20 CLMH EC7C CLGJ E7A1 VML EB23 CLT EC7D CLGIJ E7A2 VML EB23 STCTG EC7F CLJ E7A3 VMLD EB26 STMH EC09 AGHIK E7A5 VMLO EB26								
E78D VSEL EB0B SLAG EC65 CLGRJ E78E VFMS EB0C SRLG EC70 CGIT E78F VFMA EB0D SLLG EC71 CLGIT E794 VPK EB0F TRACG EC72 CIT E795 VPKLS EB14 CSY EC73 CLFIT E797 VPKS EB1D RLL EC76 CRJ E79E VFNMA EB20 CLMH EC77 CLRJ E79F VFNMA EB20 CLMH EC7C CGJ E7A1 VMLH EB23 CLT EC7E CLJ E7A2 VML EB24 STMG EC7F CLU E7A3 VML EB24 STCTG ECD8 AHIK E7A4 VMLE EB25 STCTG ECD8 AHIK E7A5 VMLO EB26 STMH ECD9 AGHIK E7A5 VMC EB2B								
E78E VFMS EB0C SRLG EC70 CGIT E78F VFMA EB0D SLLG EC71 CLGIT E794 VPK EB0F TRACG EC72 CIT E795 VPKLS EB14 CSY EC73 CLFIT E795 VPKLS EB1C RLL EC76 CRJ E79F VFNMS EB1D RLL EC77 CLRJ E79F VFNMA EB20 CLMH EC7C CGIJ E7A1 VMLH EB21 CLMY EC7D CLGIJ E7A2 VML EB23 CLT EC7E CLJ E7A3 VML EB24 STMG EC7F CLIJ E7A4 VMLE EB25 STCTG ECD8 AHIK E7A5 VMLO EB26 STMH ECD9 AGHIK E7A6 VME EB2B CLGT ECDA ALGHSIK E7A7 VMO EB2C								
E78F VFMA EB0D SLLG EC71 CLGIT E794 VPK EB0F TRACG EC72 CIT E795 VPKLS EB14 CSY EC73 CLFIT E797 VPKS EB1C RLLG EC76 CRJ E797 VPNMS EB1D RLL EC77 CLRJ E798 VFNMM EB20 CLMH EC77 CLRJ E797 VPNMA EB20 CLMH EC77 CLRJ E741 VMLH EB23 CLT EC7D CLGIJ E7A2 VML EB23 STCTG ECD8 AHIK E7A3 VMLE EB25 STCTG ECD8 AHIK E7A4 VMLE EB25 STCTG ECD8 AHIK E7A5 VMLO EB26 STMH ECD9 AGHIK E7A7 VMO EB26 STCMY EC4 CGRB E7A7 VMA EB27								
E794 VPK EB0F TRACG EC72 CIT E795 VPKLS EB14 CSY EC73 CLFIT E797 VPKS EB1C RLLG EC76 CRJ E79E VFNMA EB20 CLMH EC77 CLRJ E79F VFNMA EB20 CLMH EC7D CLGJ E7A2 VML EB23 CLT EC7E CLJ E7A2 VML EB23 CLT EC7E CLJ E7A3 VMH EB24 STMG EC7F CLU E7A4 VMLE EB25 STCTG ECD8 AHIK E7A5 VMLO EB26 STMH ECD9 AGHIK E7A5 VMC EB2B CLGT ECDA ALHSIK E7A7 VMO EB2B CLGT ECDA ALHSIK E7A7 VMA EB2C STCMH ECDB ALGHSIK E7A8 VMAH EB27	E78E	VFMS		EB0C	SRLG		EC70	CGIT
E795 VPKLS EB14 CSY EC73 CLFIT E797 VPKS EB1C RLLG EC76 CRJ E79F VFNMS EB1D RLL EC77 CLRJ E79F VFNMA EB20 CLMH EC77 CLRJ E7A2 VML EB23 CLT EC7E CLJ E7A3 VMH EB24 STMG EC7F CLJ E7A4 VMLE EB25 STCTG ECD8 AHIK E7A5 VMLO EB26 STMH ECD9 AGHIK E7A6 VME EB28 CLGT ECDA ALHSIK E7A6 VMC EB2C STCMH ECDB ALGHSIK E7A9 VMALH EB2D STCMY ECE4 CGRB E7AA VMAL EB2F LCTLG EC5 CLGRB E7AA VMAL EB31 CDSY EC7C CGIB E7AF VMAC EB34	E78F	VFMA		EB0D	SLLG		EC71	CLGIT
E795 VPKLS EB14 CSY EC73 CLFIT E797 VPKS EB1C RLLG EC76 CRJ E79F VFNMS EB1D RLL EC77 CLRJ E79F VFNMA EB20 CLMH EC77 CLRJ E7A2 VML EB23 CLT EC7E CLJ E7A3 VMH EB24 STMG EC7F CLJ E7A4 VMLE EB25 STCTG ECD8 AHIK E7A5 VMLO EB26 STMH ECD9 AGHIK E7A6 VME EB28 CLGT ECDA ALHSIK E7A6 VMC EB2C STCMH ECDB ALGHSIK E7A9 VMALH EB2D STCMY ECE4 CGRB E7AA VMAL EB2F LCTLG EC5 CLGRB E7AA VMAL EB31 CDSY EC7C CGIB E7AF VMAC EB34	E794	VPK		EB0F	TRACG		EC72	CIT
E797 VPKS EB1C RLLG EC76 CRJ E798 VFNMS EB1D RLL EC77 CLRJ E799F VFNMA EB20 CLMH EC77 CLRJ E7A1 VMLH EB21 CLMY EC7D CLGIJ E7A2 VML EB23 CLT EC7E CIJ E7A3 VML EB24 STMG EC7F CLIJ E7A4 VMLE EB25 STCTG ECD8 AHIK E7A5 VMLO EB26 STMH ECD9 AGHIK E7A6 VME EB28 CLGT ECDA ALHSIK E7A7 VMO EB2C STCMH ECDB ALGHSIK E7A7 VMAL EB2D STCMY EC64 CGRB E7A7 VMAL EB2D STCMY EC64 CGRB E7A8 VMAL EB2F LCTLG EC65 CLGRB E7A6 VMAL EB31 <td>F795</td> <td>VPKLS</td> <td></td> <td>FB14</td> <td></td> <td></td> <td></td> <td>CLEIT</td>	F795	VPKLS		FB14				CLEIT
E79E VFNMS EB1D RLL EC77 CLRJ E79F VFNMA EB20 CLMH EC7C CGIJ E7A1 WLH EB21 CLMY EC7D CLGIJ E7A2 VML EB23 CLT EC7E CLJ E7A3 VMH EB24 STMG EC7F CLJ E7A4 VMLE EB25 STCTG ECD8 AHIK E7A5 VMLO EB26 STMH ECD9 AGHIK E7A6 VME EB28 CLGT ECDA ALHSIK E7A7 VMO EB2C STCMH ECDB ALGHSIK E7A9 VMALH EB2D STCMY EC6E CGRB E7A9 VMAL EB2F LCTLG EC65 CLGRB E7AB VMAL EB31 CDSY EC7F CLRB E7AC VMALE EB31 CDSY EC7F CLRB E7AF VMAC EB44								
E79F VFNMA EB20 CLMH EC7C CGIJ E7A1 VMLH EB21 CLMY EC7D CLGIJ E7A2 VML EB23 CLT EC7E CLJ E7A3 VMH EB24 STMG EC7F CLJ E7A3 VMLE EB25 STCTG ECD8 AHIK E7A5 VMLO EB26 STMH ECD9 AGHIK E7A6 VME EB2B CLGT ECDA ALHSIK E7A6 VMC EB2C STCMH ECDB ALGHSIK E7A7 VMO EB2C STCMH ECDB ALGHSIK E7A9 VMAL EB2P LCTLG EC6E4 CGRB E7AA VMAL EB2P LCTLG EC6E5 CLGRB E7AA VMAL EB31 CDSY EC7C CGIB E7AD VMALO EB3E CDSG EC7C CGIB E7AF VMAC EB44								
E7A1 VMLH EB21 CLMY EC7D CLGIJ E7A2 VML EB23 CLT EC7E CLJ E7A3 VMH EB24 STMG EC7F CLJ E7A4 VMLE EB25 STCTG ECD8 AHIK E7A5 VMLO EB26 STMH ECD9 AGHIK E7A6 VME EB2B CLGT ECDA ALHSIK E7A7 VMO EB2C STCMH ECDB ALGHSIK E7A7 VMAL EB2D STCMY EC64 CGRB E7A8 VMAL EB2D STCMY EC64 CGRB E7A8 VMAL EB2D LCTLG EC55 CLGRB E7A8 VMAL EB31 CDSY EC76 CRB E7AC VMALE EB31 CDSY EC77 CLRB E7AF VMAO EB3E CDSG EC76 CGIB E7AF VMAO EB45								
E7A2 VML EB23 CLT EC7E CJJ E7A3 VMH EB24 STMG EC7F CLIJ E7A4 VMLE EB25 STCTG ECD8 AHIK E7A5 VMLO EB26 STMH ECD9 AGHIK E7A5 VME EB28 CLGT ECDA ALHSIK E7A7 VMO EB2C STCMH ECDB ALGHSIK E7A9 VMALH EB2D STCMY ECE4 CGRB E7A9 VMALH EB2D STCMY ECE5 CLGRB E7A8 VMAL EB2F LCTLG ECE5 CLGRB E7A0 VMALE EB31 CDSY ECF6 CRB E7AC VMALE EB31 CDSY ECF7 CLRB E7AC VMALE EB31 CDSY ECF6 CRB E7AC VMALE EB32 CDSG ECF6 CRB E7AF VMALE EB44<								
E7A3 VMH EB24 STMG EC7F CLIJ E7A4 VMLE EB25 STCTG ECD8 AHIK E7A5 VMLO EB26 STMH ECD9 AGHIK E7A6 VME EB28 CLGT ECDA ALHSIK E7A6 VMC EB2C STCMH ECDB ALGHSIK E7A9 VMALH EB2D STCMY ECE4 CGRB E7AA VMAL EB2F LCTLG ECE5 CLGRB E7AA VMAL EB30 CSG ECF6 CRB E7AC VMALE EB31 CDSY ECF7 CLRB E7AD VMALO EB3E CDSG ECFC CGIB E7AF VMAC EB44 BXHEG ECFC CGIB E7AF VMAC EB45 BXLEG ECFF CLIB E7B8 VMSL EB51 TMY ED04 LDEB E7B9 VACC EB52 </td <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td>								
E7A4 VMLE EB25 STCTG ECD8 AHIK E7A5 VMLO EB26 STMH ECD9 AGHIK E7A6 VME EB28 CLGT ECDA ALHSIK E7A7 VMO EB2C STCMH ECDB ALGHSIK E7A7 VMALH EB2D STCMY ECE4 CGRB E7AA VMAL EB2F LCTLG ECE5 CLGRB E7AB VMAH EB30 CSG ECF6 CRB E7AB VMALD EB3E CDSG ECF7 CLRB E7AD VMALO EB3E CDSG ECFC CGIB E7AF VMAD EB45 BXLEG ECFE CIB E7BF VGFM EB4C ECAG ECFF CLIB E7B8 VMS EB51 TMY ED04 LDEB E7B9 VACC EB52 MVIY ED05 LXDB E7BD VSBCBI EB56 </td <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td>								
E7A5 VMLO EB26 STMH ECD9 AGHIK E7A6 VME EB2B CLGT ECDA ALHSIK E7A7 VMO EB2C STCMH ECDB ALGHSIK E7A9 VMALH EB2D STCMY ECE4 CGRB E7AB VMAL EB2F LCTLG ECE5 CLGRB E7AB VMALE EB31 CDSY ECF6 CRB E7AC VMALE EB31 CDSY ECF7 CLRB E7AC VMALE EB31 CDSY ECF7 CLRB E7AC VMALE EB31 CDSY ECF7 CLRB E7AC VMALE EB44 BXHG ECFD CLGIB E7AE VMAE EB44 BXHG ECFD CLGIB E7AF VMAD EB45 BXLEG ECFE CLIB E7B8 VGFM EB4C ECAG ECFF CLIB E7B8 VAC E						1		
E7A6 VME EB2B CLGT ECDA ALHSIK E7A7 VMO EB2C STCMH ECDB ALGHSIK E7A9 VMALH EB2D STCMY ECE4 CGRB E7AA VMAL EB2F LCTLG ECE5 CLGRB E7AA VMALE EB30 CSG ECF6 CRB E7AC VMALE EB31 CDSY ECF7 CLRB E7AD VMALO EB3E CDSG ECFC CGIB E7AF VMAC EB44 BXHG ECFD CLGIB E7AF VMAO EB45 BXLEG ECFF CLIB E7B4 VGFM EB4C ECAG ECFF CLIB E7B8 VMSL EB51 TMY ED04 LDEB E7B9 VACC EB52 MYIY ED05 LXDB E7BB VAC EB54 NIY ED06 LXEB E7BD VGFMA EB55 <td></td> <td></td> <td></td> <td></td> <td></td> <td>1</td> <td></td> <td></td>						1		
E7A7 VMO EB2C STCMH ECDB ALGHSIK E7A9 VMALH EB2D STCMY ECE4 CGRB E7AA VMAL EB2D LCTLG ECE5 CLGRB E7AB VMAH EB30 CSG ECF6 CRB E7AC VMALE EB31 CDSY ECF7 CLRB E7AD VMALO EB3E CDSG ECFC CGIB E7AE VMAE EB44 BXHG ECFD CLGIB E7AF VMAO EB45 BXLEG ECFE CIB E7B8 VMSL EB51 TMY ED04 LDEB E7B8 VMSL EB51 TMY ED05 LXDB E7B9 VACC EB54 MIY ED05 LXDB E7BB VAC EB54 NIY ED06 LXEB E7BC VSBCBI EB56 CIY ED07 MXDB E7BF VSBI EB57	E7A5	VMLO		EB26	STMH	1	ECD9	AGHIK
E7A7 VMO EB2C STCMH ECDB ALGHSIK E7A9 VMALH EB2D STCMY ECE4 CGRB E7AA VMAL EB2D LCTLG ECE5 CLGRB E7AB VMAH EB30 CSG ECF6 CRB E7AC VMALE EB31 CDSY ECF7 CLRB E7AD VMALO EB3E CDSG ECFC CGIB E7AE VMAE EB44 BXHG ECFD CLGIB E7AF VMAO EB45 BXLEG ECFE CIB E7B8 VMSL EB51 TMY ED04 LDEB E7B8 VMSL EB51 TMY ED05 LXDB E7B9 VACC EB54 MIY ED05 LXDB E7BB VAC EB54 NIY ED06 LXEB E7BC VSBCBI EB56 CIY ED07 MXDB E7BF VSBI EB57	E7A6	VME		EB2B	CLGT	1	ECDA	ALHSIK
E7A9 VMALH EB2D STCMY ECE4 CGRB E7AA VMAL EB2F LCTLG ECE5 CLGRB E7AB VMAL EB30 CSG ECF6 CRB E7AC VMALE EB31 CDSY ECF7 CLRB E7AC VMALO EB3E CDSG ECF7 CLRB E7AE VMAL EB44 BXHG ECFD CLGIB E7AF VMAO EB45 BXLEG ECFE CIB E7BF VMSL EB51 TMY ED04 LDEB E7B8 VAC EB51 TMY ED05 LXDB E7BB VAC EB54 NIY ED06 LXEB E7BC VGFMA EB55 CLIY ED07 MXDB E7BB VSBCBI EB56 CIY ED08 KEB E7BF VSBI EB67 XIY ED09 CEB E7C1 VCDLG EB6A		VMO		EB2C		1		
E7AA VMAL EB2F LCTLG ECE5 CLGRB E7AB VMAH EB30 CSG ECF6 CRB E7AC VMALE EB31 CDSY ECF7 CLRB E7AE VMALO EB3E CDSG ECFC CGIB E7AE VMAE EB44 BXHG ECFD CLGIB E7AF VMAO EB45 BXLEG ECFE CIB E7B4 VGFM EB4C ECAG ECFF CLIB E7B8 VMSL EB51 TMY ED04 LDEB E7B9 VACC EB52 MVIY ED05 LXDB E7BB VAC EB54 NIY ED06 LXEB E7BD VGFMA EB55 CLIY ED07 MXDB E7BD VSBCBI EB56 CIY ED08 KEB E7BF VSBI EB67 XIY ED09 CEB E7C1 VCDLG EB6A						1		
E7AB VMAH EB30 CSG ECF6 CRB E7AC VMALE EB31 CDSY ECF7 CLRB E7AD VMALO EB3E CDSG ECFC CGIB E7AE VMAE EB44 BXHG ECFD CLGIB E7AF VMAO EB45 BXLEG ECFE CIB E7B8 VMSL EB51 TMY ED04 LDEB E7B8 VACC EB52 MVIY ED05 LXDB E7BB VAC EB54 NIY ED06 LVEB E7BC VGFMA EB55 CLIY ED07 MXDB E7BD VSBCBI EB56 OIY ED08 KEB E7BF VSBI EB57 XIY ED09 CEB E7C0 VCLGD EB6A ALSI ED0A AEB E7C1 VGDLG EB6E ALSI ED0B SEB E7C2 VCGD EB7A <								
E7AC VMALE EB31 CDSY ECF7 CLRB E7AD VMALO EBSE CDSG ECFC CGIB E7AE VMAC EB44 BXHG ECFD CLGIB E7AF VMAO EB45 BXLEG ECFE CIB E7B4 VGFM EB4C ECAG ECFF CLIB E7B8 VMSL EB51 TMY ED04 LDEB E7B9 VACC EB52 MVIY ED05 LXDB E7BB VAC EB54 NIY ED06 LXEB E7BC VGFMA EB55 CLIY ED07 MXDB E7BD VSBCBI EB56 CIY ED08 KEB E7BF VSBI EB57 XIY ED09 CEB E7C0 VCLGD EB6A ASI ED0A AEB E7C1 VCDLG EB6E ALSI ED0B SEB E7C2 VCGD EB7A						1		
E7AD VMALO EB3E CDSG ECFC CGIB E7AE VMAE EB44 BXHG ECFD CLGIB E7AF VMAO EB45 BXLEG ECFE CIB E7B4 VGFM EB4C ECAG ECFF CLIB E7B8 VMSL EB51 TMY ED04 LDEB E7B9 VACC EB52 MYIY ED05 LXDB E7BB VAC EB54 NIY ED06 LXEB E7BC VGFMA EB55 CLIY ED07 MXDB E7BD VSBCBI EB56 OIY ED08 KEB E7BF VSBI EB57 XIY ED09 CEB E7C1 VCDLG EB6A ASI ED0A AEB E7C2 VCGD EB7A AGSI ED0C MDEB E7C3 VCDG EB7E ALGSI ED0C MDEB E7C3 VCDG EB7E						1		
E7AE VMAE EB44 BXHG ECFD CLGIB E7AF VMAO EB45 BXLEG ECFE CIB E7B4 VGFM EB4C ECAG ECFF CLIB E7B8 VMSL EB51 TMY ED04 LDEB E7B9 VACC EB52 MVIY ED05 LXDB E7BB VACC EB54 NIY ED06 LXEB E7BC VGFMA EB55 CLIY ED07 MXDB E7BD VSBCBI EB56 OIY ED08 KEB E7BF VSBI EB57 XIY ED09 CEB E7C0 VCLGD EB6A ASI ED0A AEB E7C1 VCDLG EB6E ALSI ED0B SEB E7C2 VGGD EB7A AGSI ED0C MDEB E7C3 VCDG EB7E ALGSI ED0D DEB E7C4 VFLR EB80 <								
E7AF VMAO EB45 BXLEG ECFE CIB E7B4 VGFM EB4C ECAG ECFF CLIB E7B8 VMSL EB51 TMY ED04 LDEB E7B9 VACC EB52 MVIY ED05 LXDB E7BB VAC EB54 NIY ED06 LXEB E7BC VGFMA EB55 CLIY ED07 MXDB E7BD VSBCBI EB56 OIY ED08 KEB E7BF VSBI EB57 XIY ED09 CEB E7C0 VCLGD EB6A ASI ED0A AEB E7C1 VCDLG EB6E ALSI ED0B SEB E7C2 VCGD EB7A AGSI ED0C MDEB E7C3 VCDG EB7E ALGSI ED0D DEB E7C4 VFLR EB80 ICMH ED0E MAEB E7C5 VFLR EB81 <td< td=""><td></td><td></td><td></td><td>-</td><td></td><td>1</td><td></td><td></td></td<>				-		1		
E7B4 VGFM EB4C ECAG ECFF CLIB E7B8 VMSL EB51 TMY ED04 LDEB E7B9 VACC EB52 MYIY ED05 LXDB E7BB VAC EB54 NIY ED06 LXEB E7BC VGFMA EB55 CLIY ED07 MXDB E7BD VSBCBI EB56 OIY ED08 KEB E7BF VSBI EB57 XIY ED09 CEB E7C0 VCLGD EB6A ASI ED0A AEB E7C1 VCDLG EB6E ALSI ED0B SEB E7C2 VCGD EB7A AGSI ED0C MDEB E7C3 VCDG EB7E ALGSI ED0D DEB E7C4 VFLI EB80 ICMH ED0E MAEB E7C5 VFLR EB81 ICMY ED0F MSEB						1		
E7B8 VMSL EB51 TMY ED04 LDEB E7B9 VACCC EB52 MVIY ED05 LXDB E7BB VAC EB54 NIY ED06 LXEB E7BC VGFMA EB55 CLIY ED07 MXDB E7BD VSBCBI EB56 OIY ED08 KEB E7BF VSBI EB57 XIY ED09 CEB E7C0 VCLGD EB6A ASI ED0A AEB E7C1 VCDLG EB6E ALSI ED0B SEB E7C2 VGGD EB7A AGSI ED0C MDEB E7C3 VCDG EB7E ALGSI ED0D DEB E7C4 VFLR EB80 ICMH ED0E MAEB E7C5 VFLR EB81 ICMY ED0F MSEB	E7AF	VMAO		EB45	BXLEG	1	ECFE	CIB
E7B8 VMSL EB51 TMY ED04 LDEB E7B9 VACCC EB52 MVIY ED05 LXDB E7BB VAC EB54 NIY ED06 LXEB E7BC VGFMA EB55 CLIY ED07 MXDB E7BD VSBCBI EB56 OIY ED08 KEB E7BF VSBI EB57 XIY ED09 CEB E7C0 VCLGD EB6A ASI ED0A AEB E7C1 VCDLG EB6E ALSI ED0B SEB E7C2 VGGD EB7A AGSI ED0C MDEB E7C3 VCDG EB7E ALGSI ED0D DEB E7C4 VFLR EB80 ICMH ED0E MAEB E7C5 VFLR EB81 ICMY ED0F MSEB	E7B4	VGFM		EB4C	ECAG	1	ECFF	CLIB
E7B9 VACCC EBS2 MVIY ED05 LXDB E7BB VAC EBS4 MIY ED06 LXEB E7BC VGFMA EBS5 CLIY ED07 MXDB E7BD VSBCBI EBS6 OIY ED08 KEB E7BF VSBI EBS7 XIY ED09 CEB E7C0 VCLGD EB6A ASI ED0A AEB E7C1 VCDLG EB6E ALSI ED0B SEB E7C2 VGD EB7A AGSI ED0C MDEB E7C3 VCDG EB7E ALGSI ED0D DEB E7C4 VFLL EB80 ICMH ED0E MAEB E7C5 VFLR EB81 ICMY ED0F MSEB	E7B8	VMSL		EB51	TMY	1	ED04	
E7BB VAC EB54 NIY ED06 LXEB E7BC VGFMA EB55 CLIY ED07 MXDB E7BD VSBCBI EB56 OIY ED08 KEB E7BF VSBI EB57 XIY ED09 CEB E7C0 VCDLGD EB6A ASI ED0A AEB E7C1 VCDLG EB6E ALSI ED0B SEB E7C2 VCGD EB7A AGSI ED0C MDEB E7C3 VCDG EB7E ALGSI ED0D DEB E7C4 VFLL EB80 ICMH ED0E MAEB E7C5 VFLR EB81 ICMY ED0F MSEB		VACCC						
E7BC VGFMA EB55 CLIY ED07 MXDB E7BD VSBCBI EB56 OIY ED08 KEB E7BF VSBI EB57 XIY ED09 CEB E7C0 VCLGD EB6A ASI ED0A AEB E7C1 VCDLG EB6E ALSI ED0B SEB E7C2 VGGD EB7A AGSI ED0C MDEB E7C3 VCDG EB7E ALGSI ED0D DEB E7C4 VFLL EB80 ICMH ED0E MAEB E7C5 VFLR EB81 ICMY ED0F MSEB								
E7BD VSBCBI EB56 OIY ED08 KEB E7BF VSBI EB57 XIY ED09 CEB E7C0 VCLGD EB6A ASI ED0A AEB E7C1 VCDLG EB6E ALSI ED0B SEB E7C2 VGGD EB7A AGSI ED0C MDEB E7C3 VCDG EB7E ALGSI ED0D DEB E7C4 VFLL EB80 ICMH ED0E MAEB E7C5 VFLR EB81 ICMY ED0F MSEB		-				1		
E7BF VSBI EB57 XIY ED09 CEB E7C0 VCLGD EB6A ASI ED0A AEB E7C1 VCDLG EB6E ALSI ED0B SEB E7C2 VCGD EB7A AGSI ED0C MDEB E7C3 VCDG EB7E ALGSI ED0D DEB E7C4 VFLL EB80 ICMH ED0E MAEB E7C5 VFLR EB81 ICMY ED0F MSEB								
E7CO VCLGD EB6A ASI ED0A AEB E7C1 VCDLG EB6E ALSI ED0B SEB E7C2 VCGD EB7A AGSI ED0C MDEB E7C3 VCDG EB7E ALGSI ED0D DEB E7C4 VFLL EB80 ICMH ED0E MAEB E7C5 VFLR EB81 ICMY ED0F MSEB						1		
E7C1 VCDLG EB6E ALSI ED0B SEB E7C2 VCGD EB7A AGSI ED0C MDEB E7C3 VCDG EB7E ALGSI ED0D DEB E7C4 VFLL EB80 ICMH ED0E MAEB E7C5 VFLR EB81 ICMY ED0F MSEB						1		
E7C2 VCGD EB7A AGSI ED0C MDEB E7C3 VCDG EB7E ALGSI ED0D DEB E7C4 VFLL EB80 ICMH ED0E MAEB E7C5 VFLR EB81 ICMY ED0F MSEB						1	-	
E7C3 VCDG EB7E ALGSI ED0D DEB E7C4 VFLL EB80 ICMH ED0E MAEB E7C5 VFLR EB81 ICMY ED0F MSEB						1		-
E7C4 VFLL EB80 ICMH ED0E MAEB E7C5 VFLR EB81 ICMY ED0F MSEB	E7C2	VCGD		EB7A	AGSI	1	ED0C	MDEB
E7C4 VFLL EB80 ICMH ED0E MAEB E7C5 VFLR EB81 ICMY ED0F MSEB	E7C3	VCDG		EB7E	ALGSI	1	ED0D	DEB
E7C5 VFLR EB81 ICMY ED0F MSEB						1		
						1		
	E7C7	VFI		EB8E	MVCLU	1	ED10	TCEB

OpCode ED11 ED12	Mnemonic TCDB
ED12	TCDB
	TCXB
ED14 ED15	SQEB SQDB
ED13	MEEB
ED18	KDB
ED19	CDB
ED1A	ADB
ED1B	SDB
ED1C	MDB
ED1D	DDB
ED1E ED1F	MADB MSDB
ED1F ED24	LDE
ED25	LXD
ED26	LXE
ED2E	MAE
ED2F	MSE
ED34	SQE
ED35	SQD
ED37 ED38	MEE MAYL
ED39	MYL
ED3A	MAY
ED3B	MY
ED3C	MAYH
ED3D	MYH
ED3E	MAD
ED3F ED40	MSD SLDT
ED40 ED41	SRDT
ED41	SLXT
ED49	SRXT
ED50	TDCET
ED51	TDGET
ED54	TDCDT
ED55	TDGDT
ED58	TDCXT
ED59 ED64	TDGXT LEY
ED64 ED65	LDY
ED66	STEY
ED67	STDY
EDA8	CZDT
EDA9	CZXT
EDAA	CDZT
EDAB EDAC	CXZT
EDAC	CPXT
EDAE	CDPT
EDAF	CXPT
EE	PLO
EF	LMD
F0	SRP
F1	MVO
F2	PACK
F3 F8	UNPK ZAP
F8 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	
	0	7			
General Instructions Add	Zero	< Zero	> Zero	Overflow	
Add Halfword	Zero	< Zero	> Zero	Overflow	
Add Halfword Immediate	Zero	< Zero	> Zero	Overflow	
	Zero	< Zero	> Zero	Overflow	
Add Immediate	Zero	< Zero	> Zero	Overflow	
Add Immediate	Zero	< Zero	> Zero	Overflow	
Add Immediate High					
Add Logical	Zero, no carry	Not zero, no	Zero, carry	Not zero, carry	
Add Logical High	Zero, no carry	Not zero, no	Zero, carry	Not zero, carry	
Add Logical Immediate	Zero, no carry	Not zero, no	Zero, carry	Not zero, carry	
Add Logical with Carry	Zero, no carry	Not zero, no	Zero, carry	Not zero, carry	
Add Logical with Signed Immediate	Zero, no carry	Not zero, no	Zero, carry	Not zero, carry	
Add Logical with Signed Imme-	Zero, no carry	Not zero, no	Zero, carry	Not zero, carry	
diate High	7	carry			
AND	Zero	Not zero	_	_	
AND Immediate	ANDed bits	ANDed bits	_	_	
Observen	zero	not zero		ODLI deter	
Checksum	Checksum	_	_	CPU-deter-	
	complete			mined com-	
0:1	l	., .,		pletion	
Cipher Message	Normal com-	Verification	_	Partial com-	
	pletion	mismatch		pletion	
Cipher Message with Authenti-	Normal com-	Verification	Partial com-	Partial com-	
cation	pletion	mismatch	pletion (LAAD	pletion (time	
			or LPC zero)	out)	
Cipher Message with Chaining	Normal com-	Verification	—	Partial com-	
	pletion	mismatch		pletion	
Cipher Message with Cipher	Normal com-	Verification	 -	Partial com-	
Feedback	pletion	mismatch		pletion	
Cipher Message with Counter	Normal com-	Verification	 -	Partial com-	
	pletion	mismatch		pletion	
Cipher Message with Output	Normal com-	Verification	_	Partial com-	
Feedback	pletion	mismatch		pletion	
Compare	Equal	First op low	First op high	_	
Compare and Form Codeword	Equal	First op low	First op high	_	
		and ctl = 0, or	and ctl = 0, or		
		first op high	first op low		
		and ctl = 1	and ctl = 1		
Compare and Swap	Equal	Not equal	 -	_	
Compare and Swap and Store	Equal	Not equal	_	_	
Compare Double and Swap	Equal	Not equal	_	_	
Compare Halfword	Equal	First op low	First op high	_	
Compare Halfword Immediate	Equal	First op low	First op high	_	
Compare Halfword Relative Long	Equal	First op low	First op high	_	
Compare High	Equal	First op low	First op high	_	
Compare Immediate	Equal	First op low	First op high	_	
Compare Immediate High	Equal	First op low	First op high	_	
Compare Logical	Equal	First op low	First op high	_	
Compare Logical Characters	Equal, or	First op low	First op high	_	
under Mask	Mask is zero	1	' "		
Compare Logical High	Equal	First op low	First op high	_	
Compare Logical Immediate	Equal	First op low	First op high	_	
Compare Logical Immediate High	Equal	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-deter- mined com-	
Compare Logical Long Uni-	Equal	First op low	First op high	pletion CPU-deter-	
code				mined com- pletion	
Compare Logical Relative Long	Equal	First op low	First op high	— —	
Compare Logical String	Equal	First op low	First op high	CPU-deter- mined com- pletion	
Compare Relative Long	Equal	First op low	First op high	— —	

Condition Code →	0	1	2	3
Mask Bit Value →	8	4	2	1
Compare until Substring Equal	Equal sub-	Last bytes	Last bytes	CPU-deter-
	string	equal	unequal	mined com-
				pletion
Compression Call	Second op	First op end,	_	CPU-deter-
	end	not second op		mined com-
		end		pletion
Compute Intermediate Mes-	Normal com-	-	_	Partial com-
sage Digest	pletion			pletion
Compute Last Message Digest	Normal com-	-	_	Partial com-
	pletion			pletion
Compute Message Authentica-	Normal com-	Verification	_	Partial com-
tion Code	pletion	mismatch		pletion
Convert UTF-16 to UTF-32	Data pro-	First op full	Invalid low	CPU-deter-
	cessed		surrogate	mined com-
				pletion
Convert UTF-16 to UTF-8	Data pro-	First op full	Invalid low	CPU-deter-
	cessed		surrogate	mined com-
				pletion
Convert UTF-32 to UTF-16	Data pro-	First op full	Invalid UTF-32	CPU-deter-
	cessed		character	mined com-
				pletion
Convert UTF-32 to UTF-8	Data pro-	First op full	Invalid UTF-32	CPU-deter-
	cessed		character	mined com-
				pletion
Convert UTF-8 to UTF-16	Data pro-	First op full	Invalid UTF-8	CPU-deter-
	cessed		character	mined com-
				pletion
Convert UTF-8 to UTF-32	Data pro-	First op full	Invalid UTF-8	CPU-deter-
	cessed		character	mined com-
				pletion
Exclusive OR	Zero	Not zero	_	_
Exclusive OR Immediate	XORed bits	XORed bits	_	_
	zero	not zero		
Find Leftmost One	No one bit	_	One bit found	_
	found			
Insert Characters under Mask	All zero, or	Leftmost bit =	Not zero, but	_
	mask is zero	1	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	First op	First op longer	Overlap
•	lengths equal	shorter		· ·
Move Long Extended	Operand	First op	First op longer	CPU-deter-
•	lengths equal	shorter		mined com-
				pletion
Move Long Unicode	Operand	First op	First op longer	CPU-deter-
•	lengths equal	shorter		mined com-
				pletion
Move String	_	Second op	_	CPU-deter-
Ť		moved		mined com-
				pletion
Multiply Single (MSC, MSGC,	Zero, no over-	< Zero, no	> Zero, no	Overflow
MSGRKC, MRRKC)	flow	overflow	overflow	
OR	Zero	Not zero	_	_
OR Immediate	ORed bits	ORed bits not	_	_
	zero	zero		
Perform Cryptographic Compu-	Normal com-	Verification	_	Partial com-
tation	pletion	mismatch		pletion
Perform Locked Operation (test	Equal	First op not	First op equal,	_
bit zero)	4	equal	third op not	
			equal	
Perform Locked Operation (test	Code valid	l_		Code invalid
bit one)				
Perform Random Number	Normal com-	 _	_	Partial com-
Operation	pletion			pletion
	Zero	Not zero	_	_
Population Count		Selected bits	_	_
Population Count Rotate Then AND Selected				
Rotate Then AND Selected	Selected bits	not zero		
Rotate Then AND Selected Bits	zero	not zero	_	_
Rotate Then AND Selected Bits Rotate Then Exclusive OR	zero Selected bits	Selected bits	_	_
Rotate Then AND Selected Bits	zero			_

Condition Code →	0	1	2	3
Mask Bit Value →	8	4	2	1
Rotate Then OR Selected Bits	Selected bits	Selected bits	_	_
	zero	not zero		
Search String, Search String	 -	Character	Character not	CPU-deter-
Unicode		found	found	mined com-
				pletion
Set Program Mask ⁴	See Note	See Note	See Note	See Note
Shift Left (Double / Single)	Zero	< Zero	> Zero	Overflow
Shift Right (Double / Single)	Zero	< Zero	> Zero	
Store Clock (STCK, STCKE or STCKF)	Set state	Not-set state	Error state	Stopped state or not opera-
oroid)				tional
Store Facility List Extended	Complete list	_	_	Incomplete list
Ctoro i domiy ziot zixtoridod	stored			stored
Subtract	Zero	< Zero	> Zero	Overflow
Subtract Halfword	Zero	< Zero	> Zero	Overflow
Subtract High	Zero	< Zero	> Zero	Overflow
Subtract Logical	_	Not zero, bor-	Zero, no bor-	Not zero, no
Subtract Logical High		row Not zero, bor-	row Zero, no bor-	borrow Not zero, no
Subtract Logical Flight		row	row	borrow
Subtract Logical Immediate	_	Not zero, bor-	Zero, no bor-	Not zero, no
		row	row	borrow
Subtract Logical with Borrow	Zero, borrow	Not zero, bor-	Zero, no bor-	Not zero, no
		row	row	borrow
Test Addressing Mode	24-bit mode	31-bit mode	_	64-bit mode
Test and Set	Leftmost bit	Leftmost bit	_	_
Test under Meek (TM)	zero All zeros, or	one Mixed 0's and		All ones
Test under Mask (TM)	mask is zero	1's	_	All Offes
Test under Mask (TMH, TMHH,	All zeros or	Mixed 0's and	Mixed 0's and	All ones
TMHL, TML, TMLH, TMLL)	mask is zero	1's and left-	1's and left-	7 111 01100
,		most bit zero	most bit one	
Test under Mask High, Low	All zeros or	Mixed 0's and	Mixed 0's and	All ones
	mask is zero	1's and left-	1's and left-	
Transaction Design	Cusasastul	most bit zero	most bit one	
Transaction Begin Transaction End	Successful In TX mode		Not in TX	
nansaction End	III IX IIIOGE		mode	
Translate and Test, Translate	All zeros	Not zero, scan	Not zero, scan	_
and Test Reverse		incomplete	complete	
Translate and Test Extended,	All selected	Nonzero func-	_	CPU-deter-
Translate and Test Reverse	function codes	tion code		mined com-
Extended Translate Extended	zero	selected		pletion
Translate Extended	Data pro- cessed	First op byte equal test byte		CPU-deter- mined com-
	cesseu	equal lest byte		pletion
Translate One to One, One to	Character not	Character	_	CPU deter-
Two, Two to One, Two to	found	found		mined com-
Two				pletion
Unpack ASCII	Sign plus	Sign minus	_	Sign invalid
Unpack Unicode	Sign plus	Sign minus	_	Sign invalid
Update Tree	Compare equal at cur-	Path com- plete, no		Path not com- plete and
	rent node on	nodes com-		compared req-
	path	pared equal		ister negative
	ľ			•
Decimal Instructions	l_	l _	_	
Add Decimal	Zero	< Zero	> Zero	Overflow
Compare Decimal	Equal	First op low	First op high	_
Edit Edit and Mark	Zero Zero	< Zero < Zero	> Zero > Zero	
Shift and Round Decimal	Zero	< Zero	> Zero	Overflow
Subtract Decimal	Zero	< Zero	> Zero	Overflow
Test Decimal	Digits and sign	Sign invalid	Digit invalid	Sign and digit
	valid			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 Biased Exponent Convert HFP to HFP Zero Corvert to Fixed Convert to Fixed Convert to Logical Convert to Logical Convert to Packed Zero Convert to Packed Zero Convert to Packed Zero Convert to Packed Zero Convert to Dacked Zero Zero Zero Zero Zero Zero Zero Zero	pare and Signal pare blassed Exponent pare blassed Exponent (Equal First op low First op high First	Condition Code →	0	1	2	3
Comyert BFP to HFP Convert BFP to HFP Convert tBFP to BFP Zero	pare Blased Exponent yetr HFP to BFP yetr HFP to BFP Zero vert to Fixed yetr HFP to BFP Zero vert to Fixed Zero vert to Packed Zero vert to Zero vert to Packed Zero vert to Zero vert vert vert vert vert vert vert vert	Mask Bit Value →	8	4	2	1
Convert BFP to HFP Zero Zero Zero Special cand	Zero Zero Zero Zero Special case Speci	Compare and Signal	Equal	First op low	First op high	Unordered
Convert to Fixed Zero Zero Zero Zero Special corower to Logical Zero Zero Zero Zero Zero Special corower to Logical Zero Zero Zero Zero Zero Special corower to Zero Zero Zero Zero Zero Special corower to Zero Zero Zero Zero Zero Special corower to Zero	Vert HFP to BFP	Compare Biased Exponent	Equal	First op low	First op high	Unordered
Convert to Fixed Zero Zero Zero Zero Zero Special corover to Logical Zero Zero Zero Zero Zero Zero Special corover to Packed Zero Zero Zero Zero Zero Special corover to Zoned Zero	vert to Discola	Convert BFP to HFP	Zero	< Zero	> Zero	Special case
Convert to Logical Convert to Packed Convert to Packed Convert to Coned Divide to Integer Divide to Integer Complete, quotient normal Cad and Test (BFP) Load and Test (BFP) Load Complement (BFP) Load Complement (BFP) Load Complement (BFP) Load Negative (BFP) Load Negative (BFP) Load Positive (BFP) Load Positive (BFP) Cond Perform Floating-Point Operation (T=1) Subtract Normalized Subtract Subtract Unnormalized Subtract Unnormalized Subtract Compare Equals Load Count to Block Boundary Vector Compare Equals Vector Compare High Logicals Vector Convert to Binarys Vector Convert to Decimals Vector Convert to Decimals Vector Convert to Decimals Vector Divide Decimals Vector Find Any Element Equal Vector Find Element Compare Vector Find Element Equals Vector Find Element Equals Vector Find Element Equals Vector Find Element Not Equal Vector Find Element Not Equal Vector For Compare And Signal Vecto	vert to Logical vert to Packed Zero Zero Zero Special case vert to Zoned Zero Zero Zero Special case server to Zoned Zero Remainder complete, quotient normal complete, quotient normal complete, quotient normal complete, quotient normal complete, quotient very flow or NaN 2 Zero Zero Zero Zero Zero Zero Zero Zero	Convert HFP to BFP	Zero	< Zero	> Zero	Special case
Convert to Packed Convert to Packed Convert to Zoned Divide to Integer Remainder complete, quotient over- mal Load and Test (BFP) Load Canghement (BFP) Load Complement (BFP) Load Complement (BFP) Load Complement (BFP) Load Negative (HFP) Load Negative (HFP) Load Positive (HFP) Load Po	vert to Zoned le to intéger le to invenir le complete, quoitent over mail le udeitent rocmplete, quoitent over mail le valorient rocmplete, quoitent over mail le udeitent server le to re verte vere par vere pour le varie	Convert to Fixed	Zero	< Zero	> Zero	
Convert to Zoned Divide to Integer Remainder complete, quotient normal complete, quotient over man complete, quotient normal complete, quotient or han complete, quotient or han complete, quotient normal complete, quotient or han complete, quotient or han complete, quotient or han complete, quotient or han complete, quotient normal complete, quotient or han complete, complete	Zero Remainder Complete, quotient normal Complete Quotient normal Complement (BFP) Zero Zero Zero Zero Zero NaN Zero Zero Zero Zero Zero NaN Zero Zero Zero Zero Zero Zero NaN Zero	Convert to Logical	Zero			Special case
Divide to Integer Remainder complete, quotient normal complete, quotient normal complete, quotient normal complete, quotient overmal flow or NaN	Remainder complete, quotient normal at and Test (BFP) at and Test (BFP) Zero	Convert to Packed		< Zero	> Zero	Special case
Complete, quotient normal vertifier or NaN Zero	d and Test (BFP) at and Test (HFP) at and Test (HFP) at and Test (HFP) but (Complement (BFP) cor (Complement (HFP) but (Begative (BFP) cor (Complement (HFP) but (Begative (BFP) cor (T=0) brink (BFP) cor (T=0) corn (T=0)	Convert to Zoned	_0.0			
Quotient normal Quotient ormal Quotient normal Quotient normal Gero Sero Subtract (Incorrell) Subtract Normalized Sero Ser	di and Test (BFP) di and Test (HFP) de l'and Test (Here) d	Divide to Integer				
mal flow or NaN 2ero 2	d and Test (BFP) Jand Test (HFP) Jand Test (H					
Load and Test (BFP) Load and Test (HFP) Load Complement (BFP) Load Complement (BFP) Load Complement (HFP) Load Negative (BFP) Load Negative (BFP) Load Negative (BFP) Load Negative (HFP) Load Positive (HFP) Load Negative (HFP) Load Positive (HFP) Load Positive (HFP) Load Negative (HFP) Load Positive (HFP) Load Positive (HFP) Load Negative (HFP) Load Positive (HFP) Load Negative (Hepa Noral Negative (Hall Nega	d and Test (HFP) Zero XaN					
Load and Test (HFP) Load Complement (HFP) Load Complement (HFP) Load Negative (BFP) Load Negative (BFP) Load Negative (BFP) Load Positive (BFP) Load Rogative (BFP) Load Positive (BFP) Load Care	2 and Test (HFP) 2 cero					
Load Complement (BFP) Load Complement (HFP) Load Omplement (HFP) Load Negative (HFP) Load Ostitive (BFP) Load Ostitive (BFP) Load Ostitive (HFP) Load Ostitive (HFP) Load Ostitive (HFP) Load Positive (HFP) Perform Floating-Point Operation (T=0) Perform Floating-Point Operation (T=1) Subtract Subtract Normalized Zero	Complement (BFP) Complement					NaN
Load Complement (HFP) Load Negative (BFP) Load Positive (HFP) Load Positive (Heptonic Invalid Invalid Invalid Invalid Invalid Invalide Positive (Positive Positive Positive Positive Posit	Complement (HFP) Zero Z					_
Load Negative (BFP) Load Negative (BFP) Load Positive (Broton Invalid Positi	Negative (HFP) Zero Zero Zero Zero NaN					NaN
Load Negative (HFP) Load Positive (HFP) Perform Floating-Point Operation (T=0) Perform Floating-Point Operation (T=0) Perform Floating-Point Operation (T=0) Perform Floating-Point Operation (T=0) Subtract Subtract Normalized Zero	Negative (HFP) Positive (BFP) Positive (HFP) Positive (HFP) Zero Trap exception — Function valid NaN N				> Zero	
Load Positive (HFP) Load Positive (HFP) Perform Floating-Point Operation (T=0) Perform Floating-Point Operation (T=1) Subtract Subtract Normalized Subtract Unnormalized Zero	I Positive (HFP) I Positive (Positive (Positive A) I Positive (Positive A)				_	NaN
Load Positive (HFP) Perform Floating-Point Operation (T=0) Perform Floating-Point Operation (T=1) Subtract Normalized Subtract Unnormalized Test Data Class Test Data Group Vector-Facility Instructions Load Count to Block Boundary Vector Compare Equal Vector Compare High Logical Vector Compare High Subtract Vector Convert to Binary Vector Convert to Binary Vector Divide Decimal Vector Flement Compare Vector Flement Compare Vector Find Any Element Equal Vector Find Any Element Equal Vector Find Element Equal Vector Find Element Not Equal Vector Find Element Not Equal Vector Formpare And Signal Vector Formpare	If Positive (HFP) If If If Positive (HFP) If If Positive (HFP) If If If If Positive (HFP) If I			< Zero	I	
Perform Floating-Point Operation (T=0) Perform Floating-Point Operation (T=0) Perform Floating-Point Operation (T=1) Subtract Subtract Vormalized Zero	Normal result Normal resu			_		NaN
Perform Floating-Point Operation (T=1) Subtract Normalized Zero < Zero > Zero > Zero — — — — — — — — — — — — — — — — — — —	on (T=0) orm Floating-Point Opera- orn (T=1) orm Floating-Point Opera- orn (T=1) orn Floating-Point Opera- orn (T=1) orn Floating-Point Opera- orn (T=1) orn Floating-Point Opera- oract Normalized Zero			l 		_
Perform Floating-Point Operation (T=1) Subtract Normalized Subtract Normalized Subtract Unnormalized Test Data Class Test Data Group Vector-Facility Instructions Load Count to Block Boundary Vector Compare Decimal Vector Compare High Logical ⁵ Vector Compare High Logical ⁵ Vector Convert to Binary ⁵ Vector Convert to Binary ⁵ Vector Convert to Boicmal ⁵ Vector Element Compare Vector Flement Compare Vector Flind Any Element Equal Vector Find Any Element Equal Vector Find Element Equal ⁵ Vector Found Vector Flind Element Equal ⁵ Vector Found Vector Flind Element Equal ⁵ Vector Flind Element Not Equal Vector Flind Element Equal ⁵ Vector Flind Element	rorr Floating-Point Opera- on (T=1) ract ract Zero Ze		Normal result		Irap exception	_
tion (T=1) Subtract Normalized Subtract Unnormalized Test Data Class Test Data Class Test Data Group Vector-Facility Instructions Load Count to Block Boundary Vector Add Decimal Vector Compare Decimal Vector Compare High Logical Vector Compare High Logical Vector Compare High S Vector Find Any Element Equal Vector Find Any Element Equal Vector Find Any Element Equal S Vector Find Element Equal Vector Find Element Equal Vector Find Element Rot Equal element found, no zeros if ZS=1 Zero found Vector Find Element Not Equal element found, no zeros Vector Find Element Not Equal element found, and zero Vector Find Element Not Equal element found, less than Vector Frompare And Signal Vector Frompare And Signal Vector Frompare And Signal Vector Frompare And Signal	ract Tract Normalized Zero			exception		
Subtract Vormalized Zero < Zero > Zero > Zero	ract Vormalized Zero		Function valid	_	<u> </u>	
Subtract Normalized Subtract Unnormalized Test Data Class Test Data Group Vector-Facility Instructions Load Count to Block Boundary Vector Compare Decimal Vector Compare High Logical ⁵ Vector Compare High Logical ⁵ Vector Convert to Binary ⁵ Vector Convert to Binary ⁵ Vector Convert to Decimal ⁵ Vector Element Compare Vector Element Compare Vector Flind Any Element Equal Vector Find Any Element Equal Vector Find Element Equal ⁵ Vector Find	ract Normalized Zero Zero (Zero		_	_	_	
Subtract Unnormalized Test Data Class Test Data Class Test Data Group Vector-Facility Instructions Load Count to Block Boundary Vector Add Decimal Vector Compare Decimal Vector Compare Equal Vector Compare High Logical Vector Compare High S Vector Compare Highs All elements high No overflow Vector Convert to Binary Vector Convert to Decimal Vector Convert to Decimal Vector Element Compare Vector Flind Any Element Equal Vector Find Any Element Equal Vector Find Any Element Equal Vector Find Element Equal Vector Find Element Equal Vector Find Element Not Equal Equal Low High High	Zero					NaN
Test Data Class Zero (no match) — — — — — — — — — — — — —	Data Class Data Group Zero (no match)) Zero (no match)) One (match) One (matc					_
Test Data Group Match Zero (no match) Zer	Data Group Da				> Zero	_
Test Data Group Vector-Facility Instructions Load Count to Block Boundary Vector Add Decimal ⁵ Vector Compare Decimal Vector Compare Equal ⁵ Vector Compare High Logical ⁵ Vector Compare High Logical ⁵ Vector Compare High Logical ⁵ Vector Compare High S Vector Compare High S Vector Compare High S Vector Compare High S Vector Compare High Logical ⁵ Vector Compare High S Vector Flivide Decimal S Vector Flivide Decimal S Vector Flind Any Element Equal Vector Find Any Element Equal S Vector Find Any Element Equal S Vector Find Element Rota S Some elements high No overflow No overflow Some elements high No overflow No overflow Some elements high No overflow No overflow No overflow Some elements high No overflow No overflow No overflow Some elements high No overflow No overflow No overflow Some elements high No overflow No overflow No overflow Some elements high No over	Data Group Or-Facility Instructions I Count to Block Boundary or Add Decimal ⁵ or Compare Decimal or Compare Equal ⁵ or Compare High Logical ⁵ or Corvert to Binary ⁵ or Corvert to Binary ⁵ or Corvert to Decimal ⁵ or Corvert to Decimal ⁵ or Divide Decimal ⁵ or Element Compare or Element Compare or Element Compare Logi- or Find Any Element al ⁵ I Zero found Zero found Zero found Equal Low High High No equal element found, no zeros if 2S=1 Equal element found, no zeros if ound or Find Element Not al ⁵ or FP Compare Equal ⁵ All elements equal All elements or FP Compare High Or al ⁶ All elements ≥ Mix of ≥ and < All elements < (or unordered) All elements ≤ (or unordered)	lest Data Class		One (match)	_	_
Match Matc	match) or-Facility Instructions of Count to Block Boundary or Add Decimal ⁵ or Compare Equal ⁵ or Compare High Logical ⁵ or Compare High ⁵ or Corport to Binary ⁵ or Convert to Binary ⁵ or Corvert to Decimal ⁵ or Convert to Decimal ⁵ or Convert to Decimal ⁵ or Find Apy Element al ⁵ or Find Element Equal ⁵ Zero found Zero found Zero found Elements ar or FP Compare And Signal ar or FP Compare High Or al ⁶ or FP Compare High Or al ⁶ or FP Compare High Or al ⁶ All elements > All elements > All elements > Mix of > and ≤ Mix of > and ≤ All elements < Overflow	T . D . O				
Vector-Facility Instructions = 16 — — < 16	or Facility Instructions of Count to Block Boundary or Add Decimal ⁵ or Compare Decimal or Compare Equal ⁵ or Compare High Logical ⁵ or Compare High ⁵ or For Divide Decimal ⁵ or Plant Compare or Find Any Element ar or FP Compare And Signal ar or FP Compare Equal ⁵ or FP Compare High Or or FP Compare High ⁵ All elements ≥ Mix of > and <	lest Data Group		One (match)	_	_
Load Count to Block Boundary Vector Add Decimal ⁵ Vector Compare Equal ⁵ Vector Compare High Logical ⁵ Vector Compare High S Vector Convert to Binary S Vector Convert to Decimal S Vector Convert to Decimal S Vector Divide Decimal S Vector Element Compare Vector Element Compare Vector Element Compare Logical Vector Find Any Element Equal S Vector Find Element Equal S Vector Find Element Not Equal S Zero found Vector Find Element Not Equal Element found, no zeros Not equal element found, less than First element Vector First element Vector First element Vector Find Element Not Equal S Zero found Vector Find Element Not Equal Element First element Vector First element Vector Find Element Not Equal Element First element Vector Find Element Not Equal Element First element Vector First element	Count to Block Boundary or Add Decimals or Compare Equals or Compare High Logicals or Compare High Logicals or Compare High Decimals or Compare High Decimals or Compare High Decimals or Compare High Decimals high or Compare Highs or Compare Highs or Compare Highs or Convert to Binarys All elements high No overflow or Convert to Binarys No overflow or Divide Decimals No overflow or Element Compare Logicals or Find April Element Equal Service found Service found or Find Element Equals or Find Element Not alis or FP Compare Equal Service for FP Compare High Or alis or FP Compare Highs All elements > Mix of > and ≤ — All elements ≤ (or unordered) All elements ≤ (or unord		match)			
Load Count to Block Boundary Vector Add Decimal ⁵ Vector Compare Decimal Vector Compare High Logical ⁵ Vector Compare Highs All elements high All elements high No overflow Vector Corvert to Binary ⁵ Vector Corvert to Becimal ⁵ Vector Corvert to Decimal ⁵ Vector Corvert to Decimal ⁵ Vector Element Compare Vector Element Compare Vector Find Any Element Equal Vector Find Element Equal ⁵ Vector Find Element Equal ⁵ Vector Find Element Not Equal Vector Find Element Not Equal Equal Vector Find Element Not Equal Equal element found, no zeros Vector Find Element Not Equal Equal element found, no zeros Vector Find Element Not Equal Vector Find Element Not Equal Equal element found, no zeros Vector Find Element Not Equal Equal element found, no zeros Vector Find Element Not Equal Equal element found, no zeros Vector Find Element Not Equal Equal element found, less than Vector FP Compare And Signal Vector FP Compare And Signal Elements First element Fi	Count to Block Boundary or Add Decimal or Compare Equal or Compare High Logical or Compare High Logical or Compare High Logical or Compare High Some elements equal high or Compare High Some elements high or Compare High Some elements high or Convert to Binary or Convert to Binary or Convert to Binary or Convert to Decimal or Element Compare or Element Compare Logical or Find Apr Element Requal and Some elements high No overflow or Divide Decimal or Element Compare Logical or Find Element Equal or Find Element Equal Some element found, no zeros in Some element Compare Logical Equal Low High — Overflow Overflow or Divide Decimal or Find Element Equal Some element found, no zeros in Some element found, no zeros in Some element Compare Logical Low High — Overflow Overflow or Element Compare Logical Equal Equal element found, no zeros in Some element found, no zeros in Some element Low High — Overflow Overflow — Overflow Overflow or Element Compare Logical Equal Element found, no zeros in Some element found, no zeros in Some element Low High — No element high — No element high — Overflow Overflow — Overflow Overflow — Overflow — No equal element found, no zeros in Some element found, no zeros in S	Vester Feeilitz Instructions				
Vector Add Decimal ⁵ No overflow — Overflow Vector Compare Decimal Equal — First op low Vector Compare Equal ⁵ All elements equal — No element equal Vector Compare High Logical ⁵ All elements high — No element high Vector Compare High ⁵ All elements high — No element high Vector Convert to Binary ⁵ No overflow — — Vector Divide Decimal ⁵ No overflow — — Vector Element Compare Equal Low High — Vector Find Any Element Equal Low High — Vector Find Element Equal ⁵ Zero found Equal element found, no zeros if zeros Equal element found, and zero Not equal element found, and zero Vector Find Element Not Zero found Equal element found, less than Equal element f	or Add Decimal ⁵ or Compare Decimal or Compare Equal ⁵ or Compare High Logical ⁵ or Compare High ⁵ or Find Element Equal ⁵ or Find Element Round or Find Element Round or Find Element Round or Find Element Equal ⁵ or FP Compare Equal ⁵ All elements equal All elements equal Elements equal Elements equal Elements equal All elements equal Elements equal Elements equal All elements equal Elements equal All elements equal Elements equal All elements equal All elements equal All elements equal All elements equal Elements equal All		10			. 10
Vector Compare Decimal Vector Compare Equal ⁵ Vector Compare High Logical ⁵ Vector Compare High S Vector Compare High ⁵ Vector Divide Decimal ⁵ Vector Pivide Decimal ⁵ Vector Flind Any Element Equal Vector Find Any Element Equal Vector Find Any Element Equal Vector Find Element Equal ⁵ Zero found	or Compare Decimal or Compare Decimal or Compare Equal ⁵ or Compare High Logical ⁵ or Compare High Sort Convert to Binary ⁵ or Convert to Binary ⁵ or Convert to Decimal ⁵ or Corvert to Decimal ⁵ or Corvert to Decimal ⁵ or Divide Decimal ⁵ or Element Compare Corpiane or Element Compare Logi- or Find Any Element al ⁵ or Find Element Equal ⁵ or Find Element Equal ⁵ or FP Compare Equal ⁵ All elements ≥ Mix of ≥ and < Signal and uncordered) or FP Compare High Or al ⁶ or FP Compare High Or al ⁶ All elements ≥ Mix of > and ≤ — Equal Equal element bigh — No element equal No element high No e			_	_	
Vector Compare Equal ⁵ Vector Compare High Logical ⁵ Vector Compare High ⁵ Vector Compare High ⁵ Vector Convert to Binary ⁵ Vector Convert to Decimal ⁵ Vector Divide Decimal ⁵ Vector Element Compare Vector Flement Compare Vector Flement Compare Vector Find Any Element Equal Vector Find Any Element Equal Zero found Zero found Vector Find Element Not Equal element found, no zeros if ZS=1 Zero found Zero found Zero found Zero found Zero found Vector Find Element Not Equal element found, no zeros Not equal element found, and zero First element First element than	or Compare Equal ⁵ or Compare High Logical ⁵ or Compare High Degical ⁵ or Convert to Binary ⁵ or Convert to Decimal ⁵ or Corvert to Decimal ⁵ or Divide Decimal ⁵ or Pilment Compare or Element Compare or Element Compare or Find Any Element al ⁵ or Find Element Equal ⁵ or FP Compare And Signal ar or FP Compare Equal ⁵ or FP Compare High Or al ⁶ or FP Compare High Or al ⁶ All elements ≥ All elements ≥ Mix of ≥ and < Mix of > and ≤ Mix of > an			I	<u> </u>	Overnow
equal ments equal Some ele- Mall elements high Vector Compare High Logical ⁵ All elements high Vector Convert to Binary ⁵ All elements high No overflow Pector Convert to Decimal ⁵ Vector Convert to Decimal ⁵ Vector Element Compare Vector Element Compare Vector Element Compare Vector Element Compare Vector Find Any Element Pequal Pequal Pequal Pequal Periodo Pe	equal Mal elements high or Compare High Logicals or Compare High Logicals or Compare High Some elements high All elements high All elements high No overflow or Convert to Binarys No overflow No ove				First op high	
Vector Compare High Logical ⁵ Vector Compare High ⁵ Vector Convert to Binary ⁵ Vector Convert to Becimal ⁵ Vector Divide Decimal ⁵ Vector Element Compare Vector Element Compare Vector Flind Any Element Equal Vector Find Element Equal ⁵ Vector Find Element Equal ⁵ Vector Find Element Not Equal Vector Find Element Not Equal Zero found Zero found Zero found Vector Find Element Not Equal element found, no zeros Not equal element found, no zeros Not equal element found, and zero Not equal element found, less than Vector Find Element Not Equal element found, no zeros Not equal element found, less than Vector FP Compare And Signal Elements Some elements high No element igh No element high No element high No element high No eleme	or Compare High Logical ⁵ or Compare High ⁵ All elements high All elements high Or Convert to Binary ⁵ Or Convert to Decimal ⁵ Or Film Any Element All Element Compare Or Find Any Element All Element Equal Or Find Element Round Or Find Element Or Find Element Or Find Element Or Find Eleme	Vector Compare Equal ⁵			_	
high All elements high Some elements high No elements high No elements high No elements high No overflow Pector Divide Decimal No overflow Pector Element Compare Logical Vector Find Any Element Equal Equal Low High Pequal Equal Equal element found, no zeros if ZS=1 Vector Find Element Equal Zero found Equal element found, no zeros Not equal element found, less than Vector FP Compare And Signal Elements First element First element Elements Elements	high All elements high Some element bigh No element high No element high Overflow or Convert to Binary⁵ No overflow — — — — Overflow Overflow — — — Overflow Or Element Compare Logi- or Find Any Element Equal Sor Find Element Equal⁵ Some element Compare Logi- or Find Any Element Republic Some element Compare Logi- or Find Element Equal⁵ Zero found Some element Compare Logi- or Find Element Equal⁵ Zero found Some element Compare Logi- or Find Element Equal⁵ Zero found Some element Compare Logi- or Find Element Equal⁵ Zero found Some element Compare Logi- — — — — — — — — — — — — — — — — — — —	_				
Vector Compare High ⁵ Vector Convert to Binary ⁵ Vector Convert to Decimal ⁵ Vector Divide Decimal ⁵ Vector Element Compare Vector Element Compare Vector Element Compare Vector Find Any Element Equal Vector Find Element Equal ⁵ Vector Find Element Equal ⁵ Vector Find Element Not Equal Zero found Zero found Equal element found, no zeros Vector Find Element Not Equal element found, no zeros Vector Find Element Not Equal element found, no zeros Vector Find Element Not Equal element found, no zeros Vector Find Element Not Equal element found, no zeros Vector Find Element Not Equal element found, no zeros Vector Find Element Not Equal element found, no zeros Vector Find Element Not Equal element found, no zeros Vector Find Element Not Equal element found, no zeros Vector Find Element Not Equal element found, no zeros Vector Find Element Not Equal element found, less than First element First element Elements Elements	or Compare High5 or Convert to Binary5 or Convert to Decimal5 or Convert to Decimal5 or Divide Decimal6 or Element Compare or Element Compare or Find Any Element al6 or Find Element Equal5 or Find Element Equal5 or Find Element Round or Find Element Round or Find Element Equal5 or Find Element Round or Equal element found, no zeros Not equal element found, no zeros Not equal element found, find zero Not equal element foun	Vector Compare High Logical ⁵			_	
Vector Convert to Binarys No overflow No overflow Dector Divide Decimals No overflow Pector Element Compare Vector Find Any Element Non Equal Sequal Sector Find Element Equals Sector Find Element Not Equal element found, no zeros if ZS=1 found Sector Find Element Equals Sector Find Element Not Equal element found, no zeros Not equal element found, no zeros Not equal element found, less than Sector First element First element First element Elements Elements Elements	high No overflow No overflow Userflow Overflow Overflow Overflow Or Chovert to Decimal⁵ No overflow User Compare Logi- or Find Element Equal⁵ Sor FP Compare Equal Sor FP Compare Equal Sor FP Compare High Or alf⁵ No overflow User Compare Logi- or FP Compare High Or alf⁵ No overflow User Compare Logi- or FP Compare High Or alf⁵ No overflow User Compare Logi- or FP Compare High Or alf⁵ No overflow User Compare Logi- or FP Compare High Or alf⁵ No overflow User Compare Logi- No overflow User User Compare Logi- User Compare Logi- None equal, Equal element found, no zeros if and zero Sor FP Compare Equal⁵ Sor FP Compare Equal⁵ Not equal element found, and zero Not equal element found, no zeros Not equal element found, seros Not equal element low Mix of equal element sequal (or unor-dered) element sequal (or unor-dered) elements Not equal (or unordered) el	_				
Vector Convert to Binary ⁵ Vector Convert to Decimal ⁵ Vector Divide Decimal ⁵ Vector Element Compare Vector Element Compare Logical Vector Find Any Element Equal Vector Find Element Equal ⁵ Vector Find Element Equal ⁵ Vector Find Element Not Equal Zero found Z	or Convert to Binary ⁵ or Convert to Decimal ⁵ or Divide Decimal ⁵ or Divide Decimal ⁵ or Element Compare or Element Compare Logi- or Find Any Element al ⁵ No overflow — — — — — — — — — — — — — — — — — — —	Vector Compare High ⁵			_	
Vector Decimals Vector Divide Decimals Vector Element Compare Vector Element Compare Vector Element Compare Vector Find Any Element Equal Vector Find Element Equals Vector Find Element Not Equal element found, no zeros Vector Find Element Not Equal element found, no zeros Vector Find Element Not Equal element found, no zeros Vector Find Element Not Equal element found, no zeros Vector Find Element Not Equal Equal element found, no zeros Vector Find Element Not Equal Equal element found, no zeros Vector Find Element Not Equal Equal element found, no zeros Vector Find Element Not Equal Equal element found, no zeros Vector Find Element Not Equal Equal element found, no zeros Vector Find Element Not Equal Equal element found, no zeros Vector Find Element Not Equal Equal element found, no zeros Vector Find Element Not Equal Equal element found, no zeros Vector Find Element Not Equal Equal element found, no zeros Vector Find Element Not Equal Equal element found, no zeros Vector Find Element Not Equal Equal element found, no zeros Vector Find Element Not Equal Equal element found, no zeros Vector Find Element Not Equal Element First element	or Convert to Decimal ⁵ or Divide Decimal ⁵ or Element Compare or Equal Equal Low High High — No equal element found, no zeros if ZS=1 Zero found Zero found Figual element found, no zeros or Find Element Equal ⁵ Zero found Zero found Zero found Figual element found, no zeros or Find Element Not als Zero found Zero found Zero found Zero found Zero found Equal element found, no zeros Not equal element found, no zeros or FP Compare And Signal ar or FP Compare Equal ⁵ All elements equal All elements or FP Compare High Or als All elements ≥ Mix of > and < All elements ≤ (or unordered)	_		ments high		
Vector Divide Decimal Vector Element Compare Vector Element Compare Vector Element Compare Vector Find Any Element Equal Vector Find Any Element Equal Vector Find Element Equal Vector Find Element Equal Vector Find Element Not Equal Zero found Zero found Zero found Zero found Equal element found, no zerosi Tound, no zerosi Vector Find Element Not Equal Vector Find Element Not Equal Equal element found, no zerosi Not equal element found, no zerosi Vector Find Element Not Equal Equal element found, no zerosi Not equal element found, less than First element First element Equal element First element Equal element found, no zerosi First element Equal element First element Equal element found, no zerosi Equal element found, n	or Divide Decimals or Divide Decimals or Divide Decimals or Element Compare or Element Compare Logi- or Element Compare Logi- or Find Any Element Low Equal Low High — High — High — None equal, zero found Non zeros if ZS=1 found SZS=1 found SZS=1 found Equal element found, no zeros if ZS=1 found Element Equals or FP Compare And Signal ar or FP Compare Equals All elements equal (or unor-dered) elements or FP Compare High Or also or FP Compare High Or	Vector Convert to Binary ⁵		_	_	
Vector Divide Decimal ⁵ Vector Element Compare Vector Element Compare Logical Vector Find Any Element Equal Vector Find Element Equal ⁵ Vector Find Element Equal ⁵ Vector Find Element Equal ⁵ Vector Find Element Not Equal Eq	or Divide Decimals or Element Compare or Equal Equal Low High — Overflow Equal Equa	Vector Convert to Decimal ⁵	No overflow	 -	 -	Overflow
Vector Element Compare Vector Element Compare Vector Element Compare Vector Find Any Element Equal Vector Find Element Equal Vector Find Element Equal Vector Find Element Not Equal Equal element found, no zeros Not equal element found, less than Interval Not equal found, greater than Interval Not equal element found, no zeros Not equal element found, less than Interval Not equal found, greater than Interval Not equal element found, no zeros Interval Not e	or Element Compare or Element Compare or Element Compare or Element Compare or Find Any Element als None equal, zero found		No overflow	 	 _	Overflow
Vector Element Compare Logical None equal, vector Find Any Element Equal S Vector Find Element Equal S Vector Find Element Equal S Vector Find Element Not Equal	or FInd Any Element Louse or Find Equal Server found and zero seros if server found and zero seros or Find Element Not server found and zero seros or Find Element Not server found and zero seros or FP Compare And Signal ar or FP Compare Equal Server found and zero server found, server		Foual	Low	High	_
cal Vector Find Any Element Equal ⁵ Vector Find Element Equal ⁵ Vector Find Element Not Equal ⁶ Vector Find Element Not Equal ⁹ Vector FP Compare And Signal Elements Vector FP Compare And Signal None equal, ment found, no zeros if and zero found Equal element found, no zeros Not equal element found, less than found, greater than First element Equal element found No equal ment found and zero Not equal element found, no zeros and zero Not equal found, greater than First element Equal element found no zeros if No equal ment found no zeros First element Equal element found no zeros First element First element First element First element	or Find Any Element als None equal, zero found None equal, zero found Tero found ZS=1 Zero found ZS=1 Zero found Not equal element found, no zeros Not equal element found, less than First element high First element Sero found Zero found Zero found Zero found Zero found Not equal element found, no zeros Not equal element found, less than First element high — All elements element with fired found, preater found, prea					_
Vector Find Any Element Equals None equal, zero found no zeros if ZS=1 found Equal element found, no zeros if ZS=1 found Equal element found, no zeros Not equal element found, no zeros Vector Find Element Not Equals Vector Find Element Not Equals Vector FP Compare And Signal Elements First element First element Elements	zero found ment found, no zeros if zS=1 found zero if zS=1 found zeros if zS=1 found, greater than zero if zS=1 found zeros i		Lquui	Low	1 ligit	
Equal ⁵ zero found ment found, no zeros if $ZS=1$ found found, no zeros if $ZS=1$ found Equal element found, no zeros solvector Find Element Not Equal ⁵ Zero found Equal element found, no zeros Not equal element found, less than found, greater than Vector FP Compare And Signal Elements First element First element Elements Elements	zero found ment found, no zeros if zS=1 found zero if zS=1 found zeros if zS=1 found, greater than zero if zS=1 found zeros i		None equal	Foual ele-	Faual ele-	No equal ele-
Vector Find Element Equal ⁵ Vector Find Element Not Equal ⁵ Vector Find Element Not Equal ⁵ Zero found Zero found Zero found Zero found Zero found Zero found Not equal element found, and zero Not equal element found, and zero Not equal element found, and zero Not equal found, ess than Equal ⁵ Vector FP Compare And Signal Elements Not equal First element First element First element Elements	or Find Element Equal ⁵ Zero found Zero found Zero found Zero found Equal element found, no zeros Not equal element found, less than First element sequal or FP Compare And Signal ar or FP Compare Equal ⁵ All elements ≥ Mix of ≥ and < All elements < (or unordered) All elements > Mix of > and ≤ Or FP Compare High ⁵ All elements > Mix of > and ≤ All elements < (or unordered)					
Vector Find Element Equal ⁵ Zero found Equal element found, no zeros Vector Find Element Not Equal ⁵ Vector FP Compare And Signal Elements Zero found Zero found Equal element found, no zeros Not equal element found, less than First element First element First element Elements First element Elements	or Find Element Equal ⁵ Zero found Zero found Zero found Definite Element Not apail Elements and unequal (or unordered) Definite Equal element found, and zero Not equal element found, and zero Not equal or first element than First element First element Inligh Mix of equal and unequal (or unordered) Definite Equal First element First element and unequal (or unordered) All elements All element	Lquai				
Vector Find Element Equal ^S Zero found Equal element found, no zeros Vector Find Element Not Equal ^S Zero found Zero found Not equal element found, no zeros Not equal element found, no zeros Not equal element found, less than First element First element Equal element found, greater than Equal element found, greater than Elements Elements	or Find Element Equal ⁵ Zero found Not equal element found, and zero Not equal element found, greater than First element low All elements ard or FP Compare Equal ⁵ All elements or FP Compare High Or or FP Compare High Or All elements ≥ Mix of > and ≤ Mix of > and ≤ Equal element found, and zero Not equal or low dequal element found, greater than First element high — All elements or equal (or unor-dered) All elements ≥ Mix of > and ≤ — All elements cor unordered) All elements ≤ (or unordered)					
Vector Find Element Not Zero found Not equal element found, no zeros Not equal element found, and zero Not equal element found, less than than Vector FP Compare And Signal Elements First element First element Elements	ment found, no zeros Not equal element Not and zero Not equal element found, less than First element low All elements ar or FP Compare Equal All elements equal or FP Compare High Or also or FP Compare High S All elements > Mix of > and < — All elements < (or unordered) All elements > Mix of > and ≤ — All elements < (or unordered)	Vector Find Floment Faual ⁵	Zero found	-		Not equal, no
Vector Find Element Not Equal Square And Signal Elements Not Equal First element First element First element Elements Not equal element found, less than than than than than than than than	or FP Compare High Or or FP Compare High 5 All elements 5	vector Find Element Equal	2010 100110			
Vector Find Element Not	or Find Element Not also provided by the prov					
Equal ⁵ ment found, found, greater than Vector FP Compare And Signal Elements First element First element Elements	ment found, less than first element high or FP Compare And Signal ar or FP Compare Equal ⁵ All elements equal or FP Compare High Or also or FP Compare High S All elements ≥ Mix of > and < — All elements < (or unordered) All elements > Mix of > and ≤ — All elements < (or unordered) All elements < (or unordered) All elements < (or unordered)	Vector Find Element Not	Zero found			Egual, no zero
Vector FP Compare And Signal Elements less than than First element First element Elements	or FP Compare And Signal ar or FP Compare Equal 5 Elements equal 6 All elements 6 Or FP Compare High Or 6 All elements 6 All elements 6 Or FP Compare High 6 All elements 6 All elements 6 Mix of 6 All elements 6 Or FP Compare High 6 All elements 6 Mix of 6 All elements 6 Or FP Compare High 6 All elements 6 All elements 6 Or IP Compare High 6 All elements 6 All elements 6 Or IP Compare High 6					
Vector FP Compare And Signal Elements First element First element Elements	or FP Compare And Signal ar equal or FP Compare Equal or FP Compare Equal or FP Compare Equal and I elements equal (or unordered) and unequal (or unordered) elements or FP Compare High Or all elements ≥ Mix of ≥ and < I elements → All elements ≥ or FP Compare High⁵ All elements > Mix of > and ≤ I elements ≤ (or unordered) elements → All elements ≤ (or unordered) elements ✓ (or u	Lquai				
	ar equal low high mordered All elements equal low hix of equal mix of equal mordered All elements equal (or unordered) elements ments ments ments ments mordered elements Mix of \geq and $<$ All elements $<$ or FP Compare High 5 All elements $>$ Mix of $>$ and $<$ All elements \leq (or unordered) All elements \leq (or unordered) All elements \leq (or unordered)	Vector FP Compare And Signal	Elements			Elements
Ocaiai I Tulio	or FP Compare Equal 5 All elements and unequal (or unor-dered) dered) elements or FP Compare High Or 6 All elements $>$ Mix of $>$ and $<$ All elements $<$ All elements $<$ All elements $<$ Cor unordered) $<$ Cor unordered) $<$ All elements $<$ Cor unordered)	Scalar				
	equal and unequal (or unor-dered) dered) elements or FP Compare High Or All elements \geq Mix of $>$ and \leq — All elements \leq (or unordered) alements \leq (or unordered) All elements \leq (or unordered) alements \leq (or unordered) alements \leq (or unordered)	Vector EP Compare Equal ⁵		Mix of equal	_	All elements
	or FP Compare High Or $_{10}^{\circ}$ All elements $>$ Mix of $>$ and $<$ — $_{10}^{\circ}$ All elements $>$ Or FP Compare High ⁵ All elements $>$ Mix of $>$ and $<$ — $_{10}^{\circ}$ All elements $>$ Or unordered)	vootoi i i Oompare Equal	egual			not equal (or
	or FP Compare High Or $_{2l}^{\text{S}}$ All elements \geq $_{2l}^{\text{Mix}}$ of $>$ and $<$ — All elements $<$ (or unordered) or FP Compare High ⁵ All elements $>$ $_{2l}^{\text{Mix}}$ of $>$ and $<$ — All elements $<$ (or unordered) or unordered)		l '	(or unor-		
	or FP Compare High Or $_{2l}^{5}$ All elements \geq $_{mix}^{5}$ or FP Compare High ⁵ All elements $>$ $_{mix}^{6}$ Mix of $>$ and \leq $_{mix}^{6}$ All elements \leq $_{mix}^{6}$ (or unordered) All elements \leq (or unordered)					,
	or FP Compare High Or $_{11}^{\circ}$ All elements \geq Mix of \geq and $<$ All elements $<$ (or unordered) or FP Compare High ⁵ All elements $>$ Mix of $>$ and \leq — All elements \leq (or unordered) curvordered)		Ì			
	al^5 or FP Compare High ⁵ All elements > all Mix of > and all - all elements all (or unordered) All elements all (or unordered)	Vector FP Compare High Or	All elements ≥		 _	All elements <
	or FP Compare High ⁵ All elements $>$ Mix of $>$ and \le — All elements \le (or unordered)					
_400.	(or unordered)				1	
		Equal ⁵	All elements >	Mix of > and <	l —	All elements <
	or FP Compare Scalar Elements First element First element Flements	Equal ⁵ Vector FP Compare High ⁵	All elements >	$\text{Mix of} > \text{and} \leq$	_	
		Equal ⁵ Vector FP Compare High ⁵	All elements >		— First element	
	egual low high unordered	Equal ⁵	Elements	First element		(or unordered) Elements

Condition Code →	0	1	2	3
Mask Bit Value →	8	4	2	1
Vector FP Test Data Class	Match	Selected bit 1	_	No match
Immediate		for some (but not all) ele- ments		
Vector Isolate String ⁵	Zero element found		_	All elements nonzero
Vector Multiply and Shift Deci- mal ⁵	No overflow	_	_	Overflow
Vector Multiply Decimal ⁵	No overflow	_	_	Overflow
Vector Pack Logical Saturate ⁵	No saturation	Some satu- rated		All saturated
Vector Pack Saturate ⁵	No saturation	Some satu- rated		All saturated
Vector Perform Sign Operation Decimal ⁵	No overflow	_	_	Overflow
Vector Remainder Decimal ⁵	No overflow	_	_	Overflow
Vector Shift and Divide Deci- mal ⁵	No overflow	_	_	Overflow
Vector Shift and Round Deci- mal ⁵	No overflow	_	_	Overflow
Vector String Range Compare ⁵	Zero found	At least one in ranges, no	At least one in ranges, zero	No ranges match, no
		zero	found	zeros
Vector Subtract Decimal ⁵	No overflow	-	-	Overflow
Vector Test Decimal ⁵	Digits and sign valid	Sign invalid	Digit invalid	Sign and digit invalid
Vector Test Under Mask	All zeros or mask zero	Mixed	_	All ones
Control Instructions				
Compare and Replace DAT Table Entry	Equal	Not equal	_	_
Compare and Swap and Purge	Equal	Not equal	_	_
Diagnose ¹	See note	See note	See note	See note
Extract Stacked State	Branch state	Program call	_	_
Insert Address Space Control	entry Primary-space	state entry Secondary-	Access regis-	Home-space
·	mode	space mode	ter mode	mode
Load Address Space Parameters	Parameters loaded	Primary not available	Secondary not authorized or	Space-switch event
Load Page-Table-Entry	Address	Address	not available Invalid bit on	Exception
Address	returned; STE.P=0	returned; STE.P=1	in RTE or STE.	condition exists.
Load PSW ³	See note	See note	See note	See note
Load PSW Extended ³	See note	See note	See note	See note
Load Real Address ²	Translation	Segment table	Page table	See note
	available	entry invalid	entry invalid	
Move Page	Data moved	First op invalid, both valid in ES,	Second op invalid	_
		locked, or ES		
Move to Primary	Length ≤ 256	error —	_	Length > 256
Move to Secondary	Length ≤ 256	_	_	Length > 256
Move with Key	Length ≤ 256	_	_	Length > 256
Move with Optional Specifications	Length ≤ 4096	_	_	Length > 4096
Page In	Operation completed	ES data error	_	ES block not available
Page Out	Operation completed	ES data error	_	ES block not available
Perform Timing Facility Function	Function per- formed	_	_	Function not available
Perform Topology Function	Initiated	_	Rejected	_
Program Return	See note	See note	See note	See note
Reset Reference Bit Extended	Ref = 0, Chg = 0	Ref = 0, Chg =	Ref = 1, Chg = 0	Ref = 1, Chg = 1
Resume Program ³	See note	See note	See note	See note
Set Clock	Set	Secure	_	Not opera- tional
	·	·	·	- 31.00

Condition Code →	0	1	2	3
Mask Bit Value →	8	4	2	1
Signal Processor	Accepted	Status stored	Busy	Not opera-
Store System Information	Info provided	_	_	tional Info not avail- able
Test Access	ALET = 0	ALET uses DUALD	ALET uses PSALD	ALET = 1 or causes ART exception
Test Pending External Interruption	None pending	One or more pending	_	_
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			Not opera-
Cancel Subchannel	started	_		tional
Clear Subchannel	Function	_	_	Not opera-
	started			tional
Halt Subchannel	Function started	Non-interme- diate status pending	Busy	Not opera- tional
Modify Subchannel	Function exe- cuted	Status pend- ing	Busy	Not opera- tional
Reset Channel Path	Function started	_	Busy	Not opera- tional
Resume Subchannel	Function started	Status pend- ing	Not applicable	Not opera- tional
Start Subchannel	Function started	Status pend- ing	Busy	Not opera- tional
Store Channel Report Word	CRW stored	Zeros stored	-	-
Store Subchannel	SCHIB stored	_		Not opera- tional
Test Pending Interruption	Interruption not pending	Interruption code stored	_	_
Test Subchannel	IRB stored;	IRB stored;	 _	Not opera-
	status pending	not status pending		tional

Notes:

ı

- ¹ 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.
- For various vector-facility instructions, the condition code is optionally set based on the CS control in the M⁵ field of the instruction.

Assembler Instructions

Function	Mnemonic	Meaning
Option control	*PROCESS	Specify assembler options
·	ACONTROL	Dynamically modify options
Data definition	CCW	Define channel command word
	CCW0	Define format-0 channel command word
	CCW1	Define format-1 channel command word
	DC	Define constant
	DS	Define storage
Program	ALIAS	Rename external symbol
sectioning	AMODE	Specify addressing mode
and linking	CATTR	Define class/part name and attributes
-	COM	Identify common control section
	CSECT	Identify control section
	CXD	Cumulative length of external dummy section
	DSECT	Identify dummy section
	DXD	Define external dummy section

Function	Mnemonic	Meaning
	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	DROP	Drop base address register
assignment	USING	Use base address and register
Control of	AEJECT	Start new page in macro definition
listing	ASPACE	Space lines in macro definition
· ·	CEJECT	Conditional start new page
	EJECT	Start new page
	PRINT	Control listing contents
	SPACE	Space listing
	TITLE	Identify assembly output
Program control	ADATA	Provide data for SYSADATA file
i rogiam control	CNOP	Conditional no operation
	COPY	Copy predefined source coding
	END	End assembly
	EOU	Equate symbol
	EXITCTL	
	ICTL	Program control data for I/O exits
	ISEQ	Input format control
	LTORG	Input sequence checking
	OPSYN	Begin literal pool
		Equate operation code
	ORG POP	Set location counter
	PUNCH	Restore ACONTROL, PRINT, or USING status Punch a record
	PUNCH	
	REPRO	Save current ACONTROL, PRINT, or USING status Reproduce following record
0	ACTO	On distance of the bounds of t
Conditional	ACTR	Conditional assembly branch counter
assembly	AGO	Unconditional branch
	AIF	Conditional branch
	AINSERT	Create input record
	ANOP	Assembly no operation
	AREAD	Assign input record to SETC symbol
	GBLA	Define global SETA symbol
	GBLB	Define global SETB symbol
	GBLC	Define global SETC symbol
	LCLA	Define local SETA symbol
	LCLB	Define local SETB symbol
	LCLC	Define local SETC symbol
	MHELP	Trace macro flow
	MNOTE	Generate message
	SETA	Set arithmetic variable symbol
	SETAF	Set arithmetic variable symbol from external function
	SETB	Set binary variable symbol
	SETC	Set character variable symbol
	SETCF	Set character variable symbol from external function
Macro definition	MACRO	Macro definition header
Macro definition	MACRO MEND	Macro definition header Macro definition trailer

Source: SC26-4940

CNOP Alignment

	Quadword														
			Doubl	eword							Doubl	eword			
Fullword Fullword					Full	word			Full	word					
Half	word	Half	word	Half	word	Half	word	Half	word	Half	word	Half	word	Half	word
Byte	Byte	Byte	Byte	Byte	Byte	Byte	Byte	Byte	Byte	Byte	Byte	Byte	Byte	Byte	Byte
0,4 0,8		2,4 2.8		0,4 4.8		2,4 6.8		0,4 0.8		2,4 2.8		0,4 4.8		2,4 6.8	
0,16		2,16		4,16		6,16		8,16		10,16		12,16		14,16	

For byte offset and boundary values greater than 16, see IBM High Level Assembler for z/OS, z/VM & z/VSE Language Reference (SC26-4940).

Extended-Mnemonic Instructions for Branch on Condition and Branch Indirect on Condition

	Extende	ed Mnemoni	ic for:		
Use	BC	BCR	BIC	Meaning	M ₁ Value*
Control	В	BR	BI	Unconditional branch	15
	NOP	NOPR	_	No operation	0
After	ВН	BHR	BIH	Branch on A High	2
Compare	BL	BLR	BIL	Branch on A Low	4
Instructions	BE	BER	BIE	Branch on A Equal B	8
(A:B)	BNH	BNHR	BINH	Branch on A Not High	13
	BNL	BNLR	BINL	Branch on A Not Low	11
	BNE	BNER	BINE	Branch on A Not Equal B	7
After	BP	BPR	BIP	Branch on Plus	2
Arithmetic	BM	BMR	BIM	Branch on Minus	4
Instructions	BZ	BZR	BIZ	Branch on Zero	8
	ВО	BOR	BIO	Branch on Overflow	1
	BNP	BNPR	BINP	Branch on Not Plus	13
	BNM	BNMR	BINM	Branch on Not Minus	11
	BNZ	BNZR	BINZ	Branch on Not Zero	7
	BNO	BNOR	BINO	Branch on No Overflow	14
After Test	ВО	BOR	BIO	Branch if Ones	1
under Mask	BM	BMR	BIM	Branch if Mixed	4
Instruction	BZ	BZR	BIZ	Branch if Zeros	8
	BNO	BNOR	BINO	Branch if Not Ones	14
	BNM	BNMR	BINM	Branch if Not Mixed	11
	BNZ	BNZR	BINZ	Branch if Not Zeros	7

Source: SC26-4940.

Not applicable for BIC

Extended-Mnemonic Instructions for Relative-Branch Instructions

	Extended		Machine
Use	Mnemonic	Meaning	Instr.
General	BRU or J	Unconditional Branch Relative	BRC 15,I ₂
Branch Rel.	BRUL or JLU	Unconditional Branch Relative	BRCL 15,I ₂
on Condition	JNOP*	No Operation	BRC 0,I ₂
After	BRH or JH*	Branch Relative on A High	BRC 2,I ₂
Compare	BRL or JL*	Branch Relative on A Low	BRC 4,I ₂
Instructions	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	BRP or JP*	Branch Relative on Plus	BRC 2,I ₂
Arithmetic	BRM or JM*	Branch Relative on Minus	BRC 4,I ₂
Instructions	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,l 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

Extended mnemonic replaces the M_1 field; second operand, not shown, is $D_2(X_2,B_2)$ for RX and RXY formats and R_2 for RR format.

	Extended		Machine
Use	Mnemonic	Meaning	Instr.
After Test	BRO or JO*	Branch Relative if Ones	BRC 1,I ₂
under Mask	BRM or JM*	Branch Relative if Mixed	BRC 4,I ₂
Instruction	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,l ₂
	BRNZ or JNZ*	Branch Relative if Not Zeros	BRC 7,I ₂
Other Branch	JAS	Branch Relative and Save	BRAS R ₁ ,I ₂
Relative	JASL	Branch Relative and Save Long	BRASL R ₁ ,I ₂
Instructions	JCT	Branch Relative on Count (32)	BRCT R ₁ ,I ₂
	JCTG	Branch Relative on Count (64)	BRCTG R ₁ ,I ₂
	JXH	Branch Relative on Index High (32)	BRXH R ₁ ,R ₃ ,I ₂
	JXHG	Branch Relative on Index High (64)	BRXHG R ₁ ,R ₃ ,I ₂
	JXLE	Br. Rel. on Index Low or Equal (32)	BRXLE R ₁ ,R ₃ ,I ₂
	JXLEG	Br. Rel. on Index Low or Equal (64)	BRXLG R ₁ ,R ₃ ,I ₂

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

Suffix	Meaning	M ₃ Value	Suffix	Meaning	M ₃ Value
Н	High	2	NH	Not High	13
L	Low	4	NL	Not Low	11
E	Equal	8	NE	Not Equal	7

Explanation:

These suffixes may be appended to the following mnemonics: CGIB, CGIJ, CGIT, CGRB, CGRJ, CGRT, CIB, CIJ, CIT, CLFIT, CLGIB, CLGIJ, CLGIT, CLGRB, CLGRJ, CLGRT, CLGT, CLIB, CLIJ, CLRB, CLRJ, CLRT, CLT, CRB, CRJ, CRT. When the suffix is coded, the M₃ operand must be omitted.

Extended-Mnemonic Suffixes for Load/Store-on-Condition Instructions

Suffix	Meaning	M ₃ Value	Suffix	Meaning	M ₃ Value
0 *	One / Overflow	1	NO *	Not one / Not overflow	14
H	High	2	NH	Not High	13
P *	Plus	2	NP *	Not Plus	13
L	Low	4	NL	Not Low	11
M *	Minus / Mixed	4	NM *	Not Minus / Mixed	11
E	Equal	8	NE	Not Equal	7
Z *	Zero	8	NZ *	Not Zero	7

Explanation:

These suffixes may be appended to the following mnemonics: LOC, LOCG, LOCGHI, LOCGR, LOCHHI, LOCHI, LOCR, STOC, STOCFH, STOCG. Suffixes marked with an asterisk (*) may not be available on earlier versions of the High Level Assembler.

Extended-Mnemonic Suffixes for Rotate-Then-Insert / AND / OR / **Exclusive OR-Selected-Bits Instructions**

Extended-Mnemonic		Basic-Mne	monic	
Syntax		Equivalent		Meaning
LHHR	R_1,R_2	RISBHGZ	R ₁ ,R ₂ ,0,31	LOAD (HIGH ← HIGH)
LHLR	R_1,R_2	RISBHGZ	R ₁ ,R ₂ ,0,31,32	LOAD (HIGH ← LOW)
LLCHHR	R_1,R_2	RISBHGZ	R ₁ ,R ₂ ,24,31	LOAD LOG. CH. (HIGH ← HIGH)
LLCHLR	R_1,R_2	RISBHGZ	R ₁ ,R ₂ ,24,31,32	LOAD LOG. CH. (HIGH ← LOW)
LLCLHR	R_1,R_2	RISBLGZ	R ₁ ,R ₂ ,24,31,32	LOAD LOG. CH. (LOW ← HIGH)
LLHFR	R_1,R_2	RISBLGZ	R ₁ ,R ₂ ,0,31,32	LOAD (LOW ← HIGH)
LLHHHR	R_1,R_2	RISBHGZ	R ₁ ,R ₂ ,16,31	LOAD LOG. HW. (HIGH ← HIGH)
LLHHLR	R_1,R_2	RISBHGZ	R ₁ ,R ₂ ,16,31,32	LOAD LOG. HW. (HIGH ← LOW)
LLHLHR	R_1,R_2	RISBLGZ	R ₁ ,R ₂ ,16,31,32	LOAD LOG. HW. (LOW ← HIGH)
NHHR	R_1,R_2	RNSBG	R ₁ ,R ₂ ,0,31	AND HIGH (HIGH ← HIGH)
NHLR	R_1,R_2	RNSBG	R ₁ ,R ₂ ,0,31,32	AND HIGH (HIGH ← LOW)
NLHR	R_1,R_2	RNSBG	R ₁ ,R ₂ ,32,63,32	AND HIGH (LOW ← HIGH)
OHHR	R_1,R_2	ROSBG	R ₁ ,R ₂ ,0,31	OR HIGH (HIGH ← HIGH)
OHLR	R_1,R_2	ROSBG	R ₁ ,R ₂ ,0,31,32	OR HIGH (HIGH ← LOW)
OLHR	R_1,R_2	ROSBG	R ₁ ,R ₂ ,32,63,32	OR HIGH (LOW ← HIGH)
OHHR OHLR	R ₁ ,R ₂ R ₁ ,R ₂ R ₁ ,R ₂	ROSBG ROSBG	R ₁ ,R ₂ ,0,31 R ₁ ,R ₂ ,0,31,32	OR HIGH (HIGH ← HIGH) OR HIGH (HIGH ← LOW)

Source: SC26-4940.
* To obtain BRCL instead of BRC, add L at the end of the B mnemonic or insert L after the J of the J mnemonic. For example, change BRNZ or JNZ to BRNZL or JLNZ.

Extended-Mnemonic		Basic-Mne		
Syntax		Equivalen	t	Meaning
RISBGNZ	R ₁ ,R ₂ ,I ₃ ,I ₄ ,I ₅	RISBGN	R ₁ ,R ₂ ,I ₃ ,I ₄ +128,I ₅	Set zero-remaining-bits control to 1.
RISBGZ	R_1, R_2, I_3, I_4, I_5	RISBG	R ₁ ,R ₂ ,I ₃ ,I ₄ +128,I ₅	Set zero-remaining-bits control to 1.
RISBHGZ	R_1, R_2, I_3, I_4, I_5	RISBHG	R ₁ ,R ₂ ,I ₃ ,I ₄ +128,I ₅	Set zero-remaining-bits control to 1.
RISBLGZ	R_1, R_2, I_3, I_4, I_5	RISBLG	R ₁ ,R ₂ ,I ₃ ,I ₄ +128,I ₅	Set zero-remaining-bits control to 1.
RNSBGT	R_1, R_2, I_3, I_4, I_5	RNSBG	R ₁ ,R ₂ ,I ₃ +128,I ₄ ,I ₅	Set test-results control to 1.
ROSBGT	R_1, R_2, I_3, I_4, I_5	ROSBG	R ₁ ,R ₂ ,I ₃ +128,I ₄ ,I ₅	Set test-results control to 1.
RXSBGT	R_1, R_2, I_3, I_4, I_5	RXSBG	R ₁ ,R ₂ ,I ₃ +128,I ₄ ,I ₅	Set test-results control to 1.
XHHR	R_1,R_2	RXSBG	R ₁ ,R ₂ ,0,31	EXCL. OR HIGH (HIGH ← HIGH)
XHLR	R_1,R_2	RXSBG	R ₁ ,R ₂ ,0,31,32	EXCL. OR HIGH (HIGH ← LOW)
XLHR	R_1,R_2	RXSBG	R ₁ ,R ₂ ,32,63,32	EXCL. OR HIGH (LOW ← HIGH)
Source: SA	22-7832			

Extended-Mnemonics for Vector-Facility Instructions

See z/Architecture Principles of Operation (SA22-7832) Chapters 21-24

Summary of Constants

	Implied			Trunca-
	Length,	Default Align-		tion/
Type	Bytes	ment	Format	Padding
Α	4	Word	Value of address or expression	Left
AD	8	Doubleword	Value of address or expression	Left
В	-	Byte	Binary digits	Left
С	-	Byte	Characters	Right
CA	-	Byte	Characters (ASCII)	Right
CE	-	Byte	Characters (EBCDIC)	Right
CU	Even	Byte	Characters, translated to Unicode	Right
D	8	Doubleword	Long hex floating point	Right
DB	8	Doubleword	Long binary floating point	Right
DD	8	Doubleword	Long decimal floating point	Right
DH	8	Doubleword	Long hex floating point	Right
E	4	Word	Short hex floating point	Right
EB	4	Word	Short binary floating point	Right
ED	4	Word	Short decimal floating point	Right
EH	4	Word	Short hex floating point	Right
F	4	Word	Fixed-point binary	Left
FD	8	Doubleword	Fixed-point binary	Left
G	Even	Byte	Graphic (double-byte) characters	
Н	2	Halfword	Fixed-point binary	
J	4	Word	Symbol naming a DXD, DSECT, or class	Left
JD	8	Doubleword	Symbol naming a DXD, DSECT, or class	Left
L	16	Doubleword	Extended hex floating point	Right
LB	16	Doubleword	Extended binary floating point	Right
LD	16	Doubleword	Extended decimal floating point	Right
LH	16	Doubleword	Extended hex floating point	Right
LQ	16	Quadword	Extended hex floating point	Right
Р	-	Byte	Packed decimal	Left
Q	4	Word	Symbol naming a DXD, DSECT, or part	Left
QD	8	Doubleword	Symbol naming a DXD, DSECT, or part	Left
QY	3	Halfword	Symbol naming a DXD, DSECT, or part in long- displacement form	_
R	4	Word	PSECT address value	Left
RD	8	Doubleword	PSECT address value	Left
S	2	Halfword	Address in base-displacement form	_
SY	3	Halfword	Address in base-and-long-displacement form	_
V	4	Word	Externally defined address value	_
VD	8	Doubleword	Externally defined address value	_
X	-	Byte	Hexadecimal digits	Left
Υ	2	Halfword	Value of address or expression	Left
Z	-	Byte	Zoned decimal	Left
	SC26-494			L

Source: SC26-4940.

Assigned Storage Locations

Hex Addr	Dec Addr	Addr Type	Function
0-7	0-7	A	IPL PSW [†]
8-F	8-15	Α	CCW-type IPL: IPL CCW1 [†]
10-17	16-23	Α	
			CCW-type IPL: IPL CCW2 [†]
10-13	16-19	A	LD-IPL: Machine-loader execution-space size ^T
14-17	20-23	Α	LD-IPL: System-IPL parameter-list pointer [†]
80-83	128-131	R	External-interruption parameter
84-85	132-133	R	CPU address associated with external interruption, or zeros
86-87	134-135	R	External-interruption code (see table on page 45)
88-8B 8C-8F	136-139 140-143	R R	SVC-interruption identification: 0-12 zeros, 13-14 ILC, 15 zero, 16-31 code
			Program-interruption identification: 0-12 zeros, 13-14 ILC, 15 zero, 16-31 code (see table on page 45)
90-93	144-147	R	Data-exception code or vector-exception code: 0-23 zeros, 24- 31 code (for DXC, see table on page 46; for VXC, see table on page 47)
94-95	148-149	R	Monitor-class number: 0-7 zeros, 8-15 number
96-97	150-151	R	PER code, ATMID, AI (see table on page 48)
98-9F	152-159	R	PER address
A0	160	R	Exception access identification: 0-3 zeros, 4-7 access-register number
A1	161	R	PER access identification: 0-3 zeros, 4-7 access-register number
A2	162	R	Operand access identification (if page-translation exception recognized by MOVE PAGE): 0-3 R ₁ , 4-7 R ₂
A3	163	A/R	Store-status/machine-check architectural-mode identification: (6 zeros, 7 one
A8-AF	168-175	R	Translation-exception identification (see table on page 47)
B0-B7	176-183	R	Monitor code
B8-BB	184-187 188-191	R	Subsystem-identification word: 0-12 zeros, 13-14 SSID,15 one 16-31 subchannel number
BC-BF C0-C3	192-195	R R	I/O-interruption parameter I/O-interruption-identification word: 0-1 zeros, 2-4 I/O-interrup-
00-03	132-133	n	tion subclass, 5-31 zeros
C8-CB	200-203	R	STFL facility list (see "Facility Indications" on page 49)
E8-EF	232-239	R	Machine-check-interruption code (see diagram on page 48)
F4-F7	244-247	R	External-damage code (see diagram on page 49)
F8-FF	248-255	R	Failing-storage address
100-107	256-263	R	Enhanced-Monitor Counter-Array Origin
108-10B	264-267	R	Enhanced-Monitor Counter-Array Size
10C-10F	268-271	R	Enhanced-Monitor Exception Count
110-117	272-279	R	Breaking-event address
120-12F	288-303	R	Restart old PSW
130-13F	304-319	R	External old PSW
140-14F	320-335	R R	Supervisor-call old PSW
150-15F 160-16F	336-351 352-367	R R	Program old PSW Machine-check old PSW
170-101 170-17F	368-383	R	Input/output old PSW
1A0-1AF	416-431	R	Restart new PSW
1B0-1BF	432-447	R	External new PSW
1C0-1CF	448-463	R	Supervisor-call new PSW
1D0-1DF	464-479	R	Program new PSW
1E0-1EF	480-495	R	Machine-check new PSW
1F0-1FF	496-511	R	Input/output new PSW
11B0-11B7	4528-4535	R	Machine-check-extended-save-area address
11C0-11FF	4544-4607	R	Available for programming
1200-127F	4608-4735		Store-status/machine-check floating-point-register save area
1280-12FF 1300-130F	4736-4863 4864-4879	A/R A/R	Store-status/machine-check general-register save area Store-status PSW save area or machine-check fixed-logout
1010 1010	4000 4004	,	area‡
1318-131B 131C-131F	4888-4891 4892-4895	A A/R	Store-status prefix save area Store-status/machine-check floating-point-control-register save area
	4900-4903	A/R	Store-status/machine-check TOD-programmable-register save

R Real address.

A/R A if store status; R if machine check.

† When the configuration-z/Architecture-architectural-mode (CZAM) facility is installed.

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
1007	Warning-track interruption
1200	Malfunction alert
1201	Emergency signal
1202	External call
1406	Timing alert
1407	Measurement alert
2401	Service signal

Program-Interruption Codes

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

Code (Hex)	Condition	IL	c s	Set			str. nding	j	
0001	Operation exception		1	2	3		5	3	
0002	Privileged-operation exception			2	3		9	3	
0003	Execute exception			2	3		9	3	
0004	Protection exception		1	2	3		5	S T	
0005	Addressing exception		1	2	3		5	S T	
0006	Specification exception	0	1	2	3	С	9	3	
0007	Data exception		1	2	3	С	5	S T	
8000	Fixed-point-overflow exception		1	2	3	С			
0009	Fixed-point-divide exception		1	2	3	С	9	3	
000A	Decimal-overflow exception			2	3	С			
000B	Decimal-divide exception			2	3		9	3	
000C	HFP-exponent-overflow exception		1	2	3	С			
000D	HFP-exponent-underflow exception		1	2	3	С			
000E	HFP-significance exception		1	2		С			
000F	HFP-floating-point-divide exception		1	2			5	3	
0010	Segment-translation exception		1	2	3		N		
0011	Page-translation exception		1	2	3		N		
0012	Translation-specification exception		1	2	3		9	3	
0013	Special-operation exception		1	2	3		9	3	
0015	Operand exception			2			9	3	
0016	Trace-table exception		1	2			N		
0018	Transaction constraint		1	2	3		9	3	
001B	Vector-processing				3		9	3	
001C	Space-switch event	0	1	2		С			
001D	HFP-square-root exception			2			9		
001F	PC-translation-specification exception			2			5	3	
0020	AFX-translation exception		1	2			N		
0021	ASX-translation exception		1	2			N		
0022	LX-translation exception			2			N		
0023	EX-translation exception			2			N		

Code (Hex)	Condition	ILC Set	Instr. Ending
0024	Primary-authority exception	2	N
0025	Secondary-authority exception	1 2	N
0026	LFX-translation exception	2	N
0027	LSX-translation exception	2	N
0028	ALET-specification exception	1 2 3	S
0029	ALEN-translation exception	1 2 3	N
002A	ALE-sequence exception	1 2 3	N
002B	ASTE-validity exception	1 2 3	N
002C	ASTE-sequence exception	1 2 3	N
002D	Extended-authority exception	1 2 3	N
002E	LSTE sequence	2	N
002F	ASTE instance	1 2 3	N
0030	Stack-full exception	2	N
0031	Stack-empty exception	1 2	N
0032	Stack-specification exception	1 2	N
0033	Stack-type exception	1 2	N
0034	Stack-operation exception	1 2	N
0038	ASCE-type exception	1 2 3	N
0039	Region-first-translation exception	1 2 3	N
003A	Region-second-translation exception	1 2 3	N
003B	Region-third-translation exception	1 2 3	N
0040	Monitor event	2	С
0800	PER basic event (code may be combined with another code)	0 1 2 3	С
0800	PER nullification event	0	N
0119	Crypto-operation exception	2	N
0200	Transactional-execution-aborted event	1 2 3	С

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	General operand
01	AFP register
02	BFP instruction
03	DFP instruction
04	Quantum exception
07	Simulated quantum exception
08	IEEE inexact and truncated
0B	IXS inexact
0C	IEEE inexact and incremented
10	IEEE underflow, exact
13	IXS underflow, exact
18	IEEE underflow, inexact and truncated
1B	IXS underflow, inexact
1C	IEEE underflow, inexact and incremented
20	IEEE overflow, exact
23	IXS overflow, exact
28	IEEE overflow, inexact and truncated
2B	IXS overflow, inexact
2C	IEEE overflow, inexact and incremented
40	IEEE division by zero
43	IXS division by zero
80	IEEE invalid operation
83	IXS invalid operation
FE	Vector instruction
FF	Compare-and-trap or load-and-trap instruction

C Completed
ILC Instruction-length code
N Nullified

S

Suppressed Terminated

Vector-Exception Code (VXC)

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

١	VIX	VXC		Vector Interrupt Code (VIC)							
	VIX	VXC		Value	Meaning						
ı)	4 7		0001	IEEE invalid operation						
				0010	IEEE division by zero						
	Vector Inde	x (VIX)		0011	IEEE overflow						
	ndex to the	leftmost elen	nent that recognized the	0100	IEEE underflow						
	exception			0101	IEEE inexact						

PER Code, ATMID, and AI

At real-storage locations 150-151

	PER Code	ATMID	Al
- 1	0	8	14 15

Program	-Event-Recording (PER Code)	Addressing and-Translation-Mode ID (ATM							
Bit	<u>Meaning</u>	Bit	Meaning						
0	Successful-branching event	8	PSW bit 31						
1	Instruction-fetching event	9	ATMID-validity bit						
2	Storage-alteration event	10	PSW bit 32						
3	Reserved	11	PSW bit 5						
4	Store-using-real-address event	12-13	PSW bits 16-17						
5	Zero-address-detection event	PER ASC	CE Identification (AI)						
6	Transaction-end event	14-15	0 - primary; 1 - AR-specified;						
7	Instruction-fetch-nullification event		2 - secondary; 3 - home						

Translation-Exception Identification

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

Interrup- tion Code (Hex)	Exception or Event	Format of Information Stored*
0004	Protection	If 61 zero: rest unpredictable, If 61 one: suppression, 0-51 address; 52-53 access-exception fetch/store indication; bits 56, 60, and 61 form a 3-bit protection code, 62-63 ASCE identification, rest unpredictable, location 160 valid; if DAT was off, rest unpredictable
0010	Segment translation	0-51 address; 52-53 access-exception fetch/store indication; 54-61 unpredictable, 62-63 ASCE identification
0011	Page translation	0-51 address; 52-53 access-exception fetch/store indication; 54-60 unpredictable, if 61, zero, not MOVE PAGE; if 61 one, MOVE PAGE (see location 162); 62-63 ASCE identification
001C	Space switch	From primary-space mode: 32 old primary-space- switch-event control, 33-47 zeros, 48-63 old PASN From home-space mode: 32 home-space-switch- event control, 33-63 zeros
0020	AFX translation	32-47 zeros, 48-63 address-space number
0021	ASX translation	32-47 zeros, 48-63 address-space number
0022	LX translation	32-43 zeros, 44-63 program-call number
0023	EX translation	32-43 zeros, 44-63 program-call number
0024	Primary authority	32-47 zeros, 48-63 address-space number
0025	Secondary authority	32-47 zeros, 48-63 address-space number
0026 0027	LFX translation LSX translation	When bit 44 is 0: 32-43 zeros, 44-63 program-call number. When bit 44 is 1, 32-63 program-call num- ber

Interrup- tion Code (Hex)	Exception or Event	Format of Information Stored*
0038	ASCE type	0-51 address; 52-53 access-exception fetch/store indication; 54-61 unpredictable, 62-63 ASCE identification
0039	Region-first translation	0-51 address; 52-53 access-exception fetch/store indication; 54-61 unpredictable, 62-63 ASCE identification
003A	Region-second translation	0-51 address; 52-53 access-exception fetch/store indication; 54-61 unpredictable, 62-63 ASCE identification
003B	Region-third translation	0-51 address; 52-53 access exception fetch/store indication; 54-61 unpredictable, 62-63 ASCE identification

14

V E F G C R C P R R

00000000

Machine-Check Interruption Code

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

C E D D

	U				4				0						14		10							
	I E	A R	D A	0	G S	0	0	0	0	0	P R	F C	A P	0	C	C	0	0	0	0	0	0	0	0
ı	32				36				40		42				46		48							
	Bit				Ме	an	ing																	
	0				(SE	0) 8	Syst	em	da	ıma	age													
	1				(PE	D) I	nstr	uct	ion	-pr	OCE	essi	ng	daı	ma	ge								
	2				(SF	3 (8	Syst	em	re	COV	/ery	1												
	4				(CE	D) ⁻	Γimi	ng	fac	ilit	y da	ama	age											
	5				(EI	D) E	Exte	rna	al d	am	age	Э												
	7				(DC	G) I	Deg	rac	lati	on														
	8				(W) W	<i>l</i> arn	ing																
	9				(CF	2) (Cha	nne	el re	epc	ort p	en	din	g										
	10				(SF	9) (Serv	rice	-pr	006	ess	or c	lan	nag	е									
	11				(Cł	() (Cha	nne	el-s	ub	sys	tem	n da	ama	age									
	14				(B)	Ва	icke	dι	ıр															
	16				(SE	E) S	Stor	age	e er	ror	un	cor	rec	ted										
	17				(SC	C) S	Stor	age	e ei	101	CO	rre	cte	b										
	18				(KE	Ξ) ξ	Stor	age	e-ke	еу е	erro	r u	ncc	rre	cte	d								
	19				(DS	S) S	Stor	age	e de	egr	ada	tio	n											
	20				(W	P)	PS۱	N-N	ЛW	P١	/alio	dity												
	21				,	,	PSV								•									
	22				(PN	I (N	PSV	۷p	rog	rar	n-n	nas	k a	nd	cor	dit	ion-	-co	de	vali	dity	1		
	23				(IA) P	SW	-ins	stru	ctio	on-a	add	lres	s v	alic	lity								
	24				(FA	() F	ailir	ng-	sto	ag	e-a	ddı	ess	s va	lidi	ty								
	25				(VF	۲) ۱	/ect	or-	reg	iste	er v	alic	lity											
	26				(EC	C) E	Exte	rna	al-d	am	nag	e-c	ode	va	lidi	ty								
	27				(FF	P) F	loa	tinç	g-po	oin	t-re	gist	er	vali	dity	1								
	28				(GF	R) (Gen	era	al-re	egi	ster	va	lidi	ty										

(CR) Control-register validity

(RI) Reserved for IBM use

(ST) Storage logical validity (IE) Indirect storage error

(AR) Access-register validity

(AP) Ancillary report

(CT) CPU-timer validity

(CC) Clock-comparator validity

(DA) Delayed-access exception

(GS) Guarded-storage-registers validity

(PR) TOD-programmable-register validity

(FC) Floating-point-control-register validity

29

30

31

32 33

34

36

42

43

44

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

External-Damage Code

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

0)	0	0	0	0	0	0	0	X N	X F	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0)								8	9	10						16								24							31

Bit Meaning

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

Facility Indications

Bit Meaning when Bit is One

The first 32 facility indications are stored at real-storage locations 200-203 (C8-CB hex) by STFL; the specified number of doublewords of facility indications are stored at second-operand location by STFLE.

	Dit	meaning when bit is one
-	0	The instructions marked "N3" in the instruction-summary figures in Chapters 7 and 10 are installed.
	1	The z/Architecture architectural mode is installed.
	2	The z/Architecture architectural mode is active. When bits 2 and 168 are both zero, the ESA/390 architectural mode is active; when bit 2 is zero and bit 168 is one, the ESA/390 compatibility mode is active.
	3	The DAT-enhancement facility is installed in the z/Architecture architectural mode. The DAT-enhancement facility includes the INVALIDATE DAT TABLE ENTRY (IDTE) and COMPARE AND SWAP AND PURGE (CSPG) instructions.
	4	INVALIDATE DAT TABLE ENTRY (IDTE) performs the invalidation-and-clearing operation by selectively clearing TLB segment-table entries when a segment-table entry or entries are invalidated. IDTE also performs the clearing-by-ASCE operation. Unless bit 4 is one, IDTE simply purges all TLBs. Bit 3 is one if bit 4 is one.
	5	INVALIDATE DAT TABLE ENTRY (IDTE) performs the invalidation-and-clearing operation by selectively clearing TLB region-table entries when a region-table entry or entries are invalidated. Bits 3 and 4 are ones if bit 5 is one.
	6	The ASN-and-LX reuse facility is installed in the z/Architecture architectural mode.
	7	The store-facility-list-extended facility is installed.
	8	The enhanced-DAT facility 1 is installed in the z/Architecture architectural mode.
	9	The sense-running-status facility is installed in the z/Architecture architectural mode.
	10	The conditional-SSKE facility is installed in the z/Architecture architectural mode.
	11	The configuration-topology facility is installed in the z/Architecture architectural mode.
	13	The IPTE-range facility is installed in the z/Architecture architectural mode.
	14	The nonquiescing key-setting facility is installed in the z/Architecture architectural mode.
	16	The extended-translation facility 2 is installed.
	17	The message-security assist is installed.
	18	The long-displacement facility is installed in the z/Architecture architectural mode.
	19	The long-displacement facility has high performance. Bit 18 is one if bit 19 is one.
	20	The HFP-multiply-add/subtract facility is installed.
	21	The extended-immediate facility is installed in the z/Architecture architectural mode.
	22	The extended-translation facility 3 is installed in the z/Architecture architectural mode.
	23	The HFP-unnormalized-extension facility is installed in the z/Architecture architectural mode.
	24	The ETF2-enhancement facility is installed.
	25	The store-clock-fast facility is installed in the z/Architecture architectural mode.
	26	The parsing-enhancement facility is installed in the z/Architecture architectural mode.
	27	The move-with-optional-specifications facility is installed in the z/Architecture architectural mode.
	28	The TOD-clock-steering facility is installed in the z/Architecture architectural mode.
	30	The ETF3-enhancement facility is installed in the z/Architecture architectural mode.
	31	The extract-CPU-time facility is installed in the z/Architecture architectural mode.
	32	The compare-and-swap-and-store facility is installed in the z/Architecture architectural mode.
	33	The compare-and-swap-and-store facility 2 is installed in the z/Architecture architectural mode.
	34	The general-instructions-extension facility is installed in the z/Architecture architectural mode.
	35	The execute-extensions facility is installed in the z/Architecture architectural mode.
		The subsected as a facility is in a final in the subsection of the standard and a subsection of the standard in the subsection of the subsection of the standard in the subsection of the standard in the subsection of th

The enhanced-monitor facility is installed in the z/Architecture architectural mode.

The floating-point extension facility is installed in the z/Architecture architectural mode.

Rit Meaning when Bit is One

49

- 39 Assigned to IBM internal use.
- 4١ The set-program-parameters facility is installed in the z/Architecture architectural mode.
- 41 The floating-point-support-enhancement facilities (FPR-GR-loading, FPS-sign-handling, and DFP-rounding) are installed in the z/Architecture architectural mode.
- 12 The DFP (decimal-floating-point) facility is installed in the z/Architecture architectural mode.
- 43 The DFP (decimal-floating-point) facility has high performance. Bit 42 is one if bit 43 is
- 11 The PFPO instruction is installed in the z/Architecture architectural mode.
- 45 The distinct-operands, fast-BCR-serialization, high-word, and population-count facilities, the interlocked-access facility 1, and the load/store-on-condition facility 1 are installed in the z/Architecture architectural mode.
- 47 The CMPSC-enhancement facility is installed in the z/Architecture architectural mode.
- The decimal-floating-point zoned-conversion facility is installed in the z/Architecture archi-48 tectural mode
- The execution-hint, load-and-trap, and processor-assist facilities and the miscellaneousinstruction-extensions facility 1, are installed in the z/Architecture architectural mode. 50 The constrained transactional-execution facility is installed in the z/Architecture architec-
- tural mode. This bit is meaningful only when bit 73 is one.
- 51 The local-TLB-clearing facility is installed in the z/Architecture architectural mode.
- The interlocked-access facility 2 is installed. 52
- 53 The load/store-on-condition facility 2 and load-and-zero-rightmost-byte facility are installed in the z/Architecture architectural mode.
- 57 The message-security-assist-extension 5 is installed in the z/Architecture architectural mode.
- 58 The miscellaneous-instruction-extensions facility 2 is installed in the z/Architecture architectural mode.
- 66 The reset-reference-bits-multiple facility is installed in the z/Architecture architectural mode
- 67 The CPU-measurement counter facility is installed in the z/Architecture architectural
- 68 The CPU-measurement sampling facility is installed in the z/Architecture architectural mode.
- 73 The transactional-execution facility is installed in the z/Architecture architectural mode. Bit 49 is one when bit 73 is one
- 7/ The store-hypervisor-information facility is installed in the z/Architecture architectural mode (see z/VM CP Programming Services [SC24-6179]).
- 75 The access-exception-fetch/store-indication facility is installed in the z/Architecture architectural mode. 76 The message-security-assist-extension 3 is installed in the z/Architecture architectural
- mode. 77 The message-security-assist-extension 4 is installed in the z/Architecture architectural
- 78 The enhanced-DAT facility 2 is installed in the z/Architecture architectural mode.
- 80 The decimal-floating-point packed-conversion facility is installed in the z/Architecture architectural mode.
- 129 The vector facility for z/Architecture is installed in the z/Architecture architectural mode.
- The instruction-execution-protection facility is installed in the z/Architecture architectural 130 mode.
- The side-effect-access facility and the enhanced-suppression-on-protection facility 2 are 131 installed in the z/Architecture architectural mode.
- 133 The guarded-storage facility is installed in the z/Architecture architectural mode.
- 134 The vector-packed-decimal facility is installed in the z/Architecture architectural mode. 135
- The vector-enhancements facility 1 is installed in the z/Architecture architectural mode.
- 138 The configuration z/Architecture architectural mode facility is installed.
- 139 The multiple-epoch facility is installed in the z/Architecture architectural mode.
- 142 The store-CPU-counter-multiple facility is installed.
- 144 The test-pending-external-interruption facility is installed in the z/Architecture architectural mode.
- 145 The insert-reference-bits-multiple facility is installed in the z/Architecture architectural
- 146 The message-security-assist-extension 8 is installed in the z/Architecture architectural
 - The ESA/390-compatibility-mode facility is installed in the configuration.

Control Registers

CR	Bits	Name of Field	Associated with	lni
0	8	Transactional-execution control	Transactional-execution	0
	9	Program-interruption filtering override	Transactional-execution	0
	10	Clock-comparator sign control	TOD clock	0
	30	Warning-track-interruption enable- ment	Virtual machines	0
	32	Trace TOD-clock control	TOD clock	0
	33	SSM-suppression control	SSM instruction	0
	34	TOD-clock-sync control	TOD clock	0
	35	Low-address-protection control	Low-address protection	(
	36	Extraction-authority control	Instruction authorization	(
	37	Secondary-space control	Instruction authorization	(
	38	Fetch-protection-override control	Key-controlled protection	(
	39	Storage-protection-override control	Key-controlled protection	(
	40	Enhanced-DAT-enablement control	Dynamic address translation	(
	43	Instruction-execution-protection- enablement control	Instruction-execution protection	(
	44	ASN-and-LX-reuse control	Instruction authorization	(
	45	AFP-register control	Floating point	(
	46	Vector enablement control	Vector facility for z/Architecture	
	48	Malfunction-alert subclass mask	External interruptions	
	49	Emergency-signal subclass mask	External interruptions	
	50	External-call subclass mask	External interruptions	Ι,
	52	Clock-comparator subclass mask	External interruptions	Ι,
	53	CPU-timer subclass mask	External interruptions	1
	54	Service-signal subclass mask	External interruptions	Ι,
	56	Unused (See note)		
	57	Interrupt-key subclass mask	External interruptions	
	58	Unused (See note)		
	59	ETR subclass mask	External interruptions	Ι,
	61	Crypto control	Cryptography	1
1	0-63	Primary address-space-control ele-	Dynamic address translation	
		ment		
	0-51	Primary region-table or segment- table origin or real-space token origin	Dynamic address translation	(
	54	Primary subspace-group control	Subspace groups	(
	55	Primary private-space control	Dynamic address translation	'
	56	Primary storage-alteration-event control	Program-event recording	'
	57	Primary space-switch-event control	Program interruptions	1
	58	Primary real-space control	Dynamic address translation	
	60-61	Primary designation-type control	Dynamic address translation	
	62-63	Primary table length	Dynamic address translation	
2	33-57	Dispatchable-unit-control-table origin	Access-register translation	
	59	Guarded-storage facility enablement	Guarded-storage facility	
	61	Transaction diagnostic scope	Transactional execution	
	62-63	Transaction diagnostic control	Transactional execution	
3	0-31	Secondary ASTE Instance Number	Instruction authorization	T
	32-47	PSW-key mask	Instruction authorization	
	48-63	Secondary ASN	Address spaces	
4	0-31	Primary ASTE Instance Number	Instruction authorization	T
	32-47	Authorization index	Instruction authorization	1
	48-63	Primary ASN	Address spaces	
5	33-57	Primary-ASTE origin	Access-register translation	
6	32-39	I/O-interruption subclass mask	I/O interruptions	
7	0-63	Secondary address-space-control element	Dynamic address translation	
	0-51	Secondary region-table or segment- table origin or real-space token origin	Dynamic address translation	'
	54	Secondary subspace-group control	Subspace groups	
	55	Secondary private-space control	Dynamic address translation	
	56	Secondary storage-alteration-event control	Program-event recording	'
	58	Secondary real-space control	Dynamic address translation	
	60-61	Secondary designation-type control	Dynamic address translation	(

CR	Bits	Name of Field	Associated with	Init*
8	16-31	Enhanced-monitor masks	MONITOR CALL instruction	0
	32-47	Extended authorization index	Access-register translation	0
	48-63	Monitor masks	MONITOR CALL 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
	37	Zero-address-detection-event mask	Program-event recording	0
	38	Transaction-end-event mask	Program-event recording	0
	39	Instruction-fetching-nullification-event mask	Program-event recording	0
	40	Branch-address control	Program-event recording	0
	41	Event-suppression control	Program-event recording	0
	42	Storage-alteration-space control	Program-event recording	0
10	0-63	PER starting address	Program-event recording	0
11	0-63	PER ending address	Program-event recording	0
12	0	Branch-trace control	Tracing	0
	1	Mode-trace control	Tracing	0
	2-61	Trace-entry address	Tracing	0
	62	ASN-trace control	Tracing	0
	63	Explicit-trace control	Tracing	0
13	0-63	Home address-space-control element	Dynamic address translation	0
	0-51	Home region-table or segment-table origin or real-space token origin	Dynamic address translation	0
	54	Home subspace-group control	Subspace groups	0
	55	Home private-space control	Dynamic address translation	0
	56	Home storage-alteration-event control	Program-event recording	0
	57	Home space-switch-event control	Program interruptions	0
	58	Home real-space control	Dynamic address translation	0
	60-61	Home designation-type control	Dynamic address translation	0
	62-63	Home table length	Dynamic address translation	0
14	32	Unused (See note)		1
	33	Unused (See note)		1
	35	Channel-report-pending subclass mask	I/O machine-check handling	0
	36	Recovery subclass mask	Machine-check handling	0
	37	Degradation subclass mask	Machine-check handling	0
	38	External-damage subclass mask	Machine-check handling	1
	39	Warning subclass mask	Machine-check handling	0
	42	TOD-clock-control-override control	TOD clock	0
	44	ASN-translation control	Instruction authorization	0
	45-63	ASN-first-table origin	ASN translation	0
15	0-60	Linkage-stack-entry address	Linkage-stack operations	0

* Value after initial CPU reset.

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

Floating-Point-Control (FPC) Register

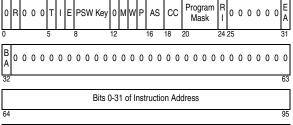
Masks	Masks Flags								DXC (see page 46)				
I I I I I M M M M M i z o u x	I M 0 0 q	S S F F i z	S F o	SFu	S F x	S F q	0	0	or VCX (see page 47)	0	DRM	0	BRM
0		8							16	24			31

Bit	<u>Meaning</u>
0	(IMi) IEEE-invalid-operation mask

- (IMz) IEEE-division-by-zero mask 2 (IMo) IEEE-overflow mask
- 3 (IMu) IEEE-underflow mask 4 (IMx) IEEE-inexact mask
- 5 (IMg) Quantum-exception mask (SFi) IEEE-invalid-operation flag
- (SFz) IEEE-division-by-zero flag 9 (SFo) IEEE-overflow flag 10 (SFu) IEEE-underflow flag 11
- (SFx) IEEE-inexact flag 13 (SFq) Quantum-exception flag
- 16-23 (DXC) Data-exception code (see table on page 46)
- 25-27 (DRM) DFP Rounding mode
- 000 Round to nearest with ties to even
 - 001 Round toward 0 010 Round toward +∞
 - 011 Round toward -∞
 - 100 Round to nearest with ties away from 0 101 Round to nearest with ties toward 0
 - 110 Round away from 0
 - 111 Round to prepare for shorter precision
- (BRM) BFP Rounding mode 29-31
 - 000 Round to nearest 001 Round toward 0
 - 010 Round toward +∞
 - 011 Round toward -∞
 - 111 Round to prepare for shorter precision

Program-Status Word (PSW)

z/Architecture PSW



96			
Rit	Meaning		

_	
1	(R) Program-event-recording mask
5	(T = 1) DAT mode

(I) Input/output mask 6 (E) External mask 12 Zero indicates a 16-byte PSW

16-17

(M) Machine-check mask 13 (W = 1) Wait state 15 (P = 1) Problem state

> xx Real mode (T = 0) 00 - Primary-space mode (T = 1) 01 - Access-register mode (T = 1) 10 - Secondary-space mode (T = 1)

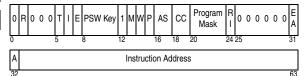
11 - Home-space mode (T = 1) (CC) Condition code 18-19 Fixed-point-overflow mask 20

21 Decimal-overflow mask 22 HFP-exponent-underflow mask

24 Reserved for IBM use HFP-significance mask 23 31/32 Extended/basic addressing mode

00 - 24-bit mode 01 - 31-bit mode 10 - Invalid 11 - 64-bit mode

Short-Format PSW



Meaning 12

One indicates a short-format PSW

Dynamic Address Translation

Virtual-Address Format

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

RX Region index (region = 2G bytes)

RFX Region first index RSX Region second index

RTX Region third index

SX Segment index (segment = 1M bytes)

PX Page index (page = 4K bytes)

BX Byte index

Address-Space-Control Element (ASCE)

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

	Region-Table or Segment-Table Origin		GI	S	ΧR	DT	DL
0		52	54		58	60	63
Bit	<u>Meaning</u>						
54	(G) Subspace-group control						
55	(P) Private-space control						
56	56 (S) Storage-alteration-event control						
57	57 (X) Space-switch-event control						
58	(R) Real-space control (R = 0)						
60-61	(DT) Designation-type control						
	11 Region-first-table						
	10 Region-second-table						
	01 Region-third-table						
	00 Segment-table						
62-63	(DL) Designation length (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

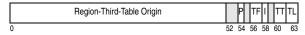
	laava	lass	Incr. En-	Max.	Max. En-	Max Table Maps		
Table	Incre- ment	Incr. Size	tries	Size	tries	Regions	Bytes	
Region First	1-4	4KB	512	16KB	2K	8G	16E =16×2 ⁶⁰	
Region Second	1-4	4KB	512	16KB	2K	4M	$8P = 8 \times 2^{50}$	
Region Third	1-4	4KB	512	16KB	2K	2K	$4T = 4 \times 2^{40}$	
Segment	1-4	4KB	512	16KB	2K	1	$2G = 2 \times 2^{30}$	
Page	1	2KB	256	2KB	256	_	$1M = 2^{20}$	

Region-Table Entry (RTE)

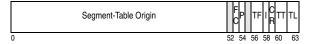
Region-First-Table Entry (RFTE)

Region-Second-Table Origin		Р	TF	ı	TT	TL
0	52	54	56	58	60	63

Region-Second-Table Entry (RSTE)



Region-Third-Table Entry (RTTE, FC=0)



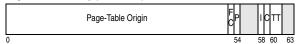
Region-Third-Table Entry (RTTE, FC-1)



Bit Meaning

- 47 (AV) Access-control (ACC) and fetch-protection (F) validity bit
- 48-51 (ACC) Access-control bits
- 52 (F) Fetch-protection bit
- 53 (FC) Format control
- 54 (P) DAT protection bit
- 55 (IEP) Instruction-execution-protection bit (format-1 RTTE only)
- 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
 - 59 (CR) Common-region bit
 - 60-61 (TT) Table-type bits (for this table) 11=Region first table
 - 10=Region second table
 - 01=Region third table
 - 62-63 (TL) Table length (for next-lower-level table) (x 4K bytes)

Segment-Table Entry (STE, FC=0)



Segment-Table Entry (STE, FC=1)



Bit Meaning

- 47 (AV) Access-control (ACC) and fetch-protection (F) validity bit
- 48-51 (ACC) Access-control bits
- 52 (F) Fetch-protection bit
- 53 (FC) Format control
- 54 (P) DAT-protection bit 55 (IEP) Instruction-execution-protection bit (format-1 STE only)
- 58 (I) Segment-invalid bit
- 59 (CS) Common-segment bit
 - 60-61 (TT) Table-type bits (for this table): 00=Segment table

Page-Table Entry (PTE)



Bit Meaning

- 53 (I) Page-invalid bit
- 54 (P) DAT-protection bit
- 55 (IEP) Instruction-execution-protection bit (format-1 STE only)

ASN Translation

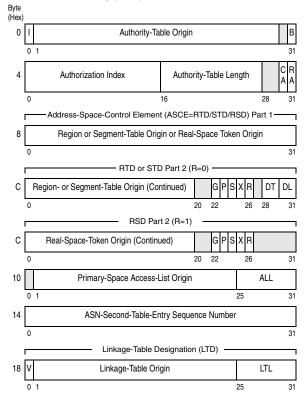
Address-Space Number (ASN)

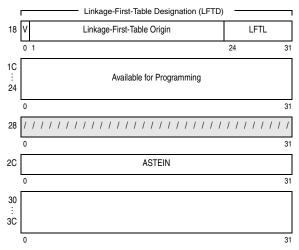
ASN-First- Table Index	ASN-Se Table I	
0	10	15

ASN-First-Table Entry

I	ASN-Second-Table Origin	0 0 0 0 0 0
0 1		26 31
Bit	<u>Meaning</u>	
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 4.30 (CA) Controlled-ASN bit

4.31 (RA) Reusable-ASN 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)

18.24-31 (LFTL) Linkage-first-table length (x 256 bytes)

PC-Number Translation

Program-Call Number (20-Bit)

	Linkage Index	Entry Index	
32	44	56	63

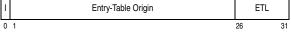
Program-Call Number (32-Bit, Bit 44=0)

	0	LFX	LSX	Entry Index	
32	44		51	56	63

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

	LFX1	1	LFX2	LSX	Entry Index	
32		44		51	56	63

Linkage-Table Entry (LTE)



Bit Meaning

0 (I) LX-invalid bit

26-31 (ETL) Entry-table length (x 128 bytes)

Linkage-First-Table Entry (LFTE)

(I) LFX-invalid bit

1	Linkage-Second-Table Origin		
0 1		24	31
Bit	Meaning		

Linkage-Second-Table Entry (LSTE)

(ETL) Entry-table length (x 128 bytes)

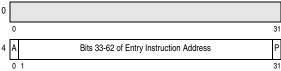
	• • • •		
1	Entry-Table Origin		ETL
0 1		:	26 31
	LSTESN		
32			63
<u>Bit</u> 0	Meaning (I) LSX-invalid bit		

Entry-Table Entry (ETE)

Byte (Hex)

0

If Bit 10.1 (G) Is Zero



If Bit 10.1 (G) Is One

Р
31

Bits 0-31 of Entry Instruction Address

8	Authorization Key Mask		Address-Space Number		
	0		16	31	
С		Entry Key Mask			

	U					'	0			3
10	Т	GRIKME	cs	EK			E	Entry Ext. Authori	ty Index	
	0	3	8		12	1	6			31
14			ASN-S	Second	-Table	-Entr	v Addre	ess		

	,		
	0 1	26 3	1
18	Bits 0-31 of Entry Parameter		

	0		31
1C		Bits 32-63 of Entry Parameter	

Byte.Bit Meaning

4.0 (A) Entry addressing mode4.31 (P) Entry problem state

10.0 (T) PC-type bit (zero: basic; one: stacking)

10.1 (G) Entry extended addressing mode

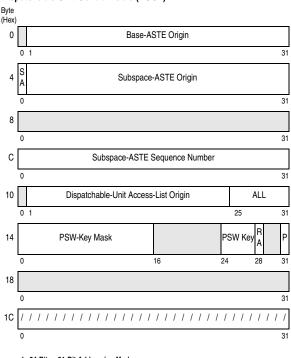
10.2	(RI) Reserved for IBM use
10.3	(K) PSW-key control (zero: unchanged; one: replace if stacking
10.4	(M) PSW-key-mask control (zero: Or; one: replace if stacking)
10.5	(E) EAX control (zero: unchanged; one: replace if stacking)
10.6	(C) Address-space-control control
10.7	(S) Secondary-ASN control
10.8-11	(EK) Entry key

Access-Register Translation

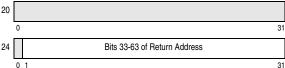
Access-List-Entry Token (ALET)

0 0 0	0 0 0 0 P	ALESN	Access-List-Entry Number	
0	7	8	16	31
Bit	Meaning			
7	(P) Primary-list bit (zero: use DUCT; one: use primary ASTE)			
8-15	(ALESN) Access-list-entry sequence number			

Dispatchable-Unit-Control Table (DUCT)







In 64-Bit Addressing Mode

	In 64-Bit Addressing Mode		
20	Bits 0-31 of Return Address		
	0		31
24	Bits 32-63 of Return Address		
	0		31
28			
	0		31
2C	Trap-Control-Block Address		Ε
	0	29	31
30 : 3C			
3C			

Byte.Bit Meaning

0

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)

Acces	ss-List Entry	(ALE)			
I	F _O P	ALESN	Access-List-Entry (ALE	Authorization EAX)	Index
0 1	6	8	16		31
32					63
	ASN-S	Second-Table-Entry	Origin (ASTEO)		
64				90	95
	ASN-S	econd-Table-Entry	Sequence Number (AS	TESN)	
96					127
<u>Bit</u>	Meaning				
0	(I) AI FNLiny	alid hit			

(I) ALEN-invalid bit (FO) Fetch-only bit

(P) Private bit

8-15 (ALESN) Access-list-entry sequence number

Linkage-Stack Entries

Entry Descriptor

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

31

Bit Meaning

0 (U) Unstack-suppression bit

1-7 Entry type:

Header entry = 0001001 binary Trailer entry = 0001010 binary Branch state entry = 0001100 binary Program-call state entry = 0001101 binary Available for program use = 1xxxxxxx binary

Header Entry (Entry Type 0001001)

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

Bit Meaning

63 (B) Backward stack-entry validity bit

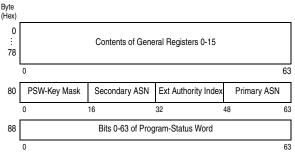
Trailer Entry (Entry Type 0001010)

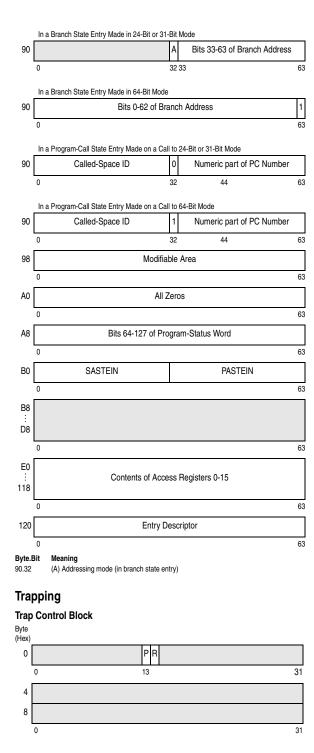
	Bits 0-31 of Forward-Section-Header Address		
0			31
	Bits 32-60 of Forward-Section-Header Address		F
32		61	63
	Entry Descriptor (First Half)		
64			95
	Entry Descriptor (Second Half)		
96			127

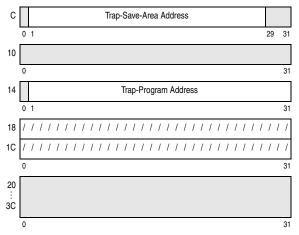
Bit Meaning

63 (F) Forward-section validity bit

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







Byte.Bit Meaning

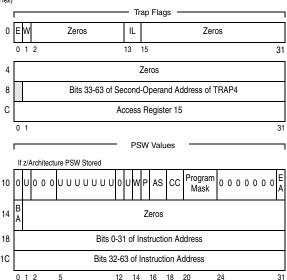
 (P) PSW control (zero: PSW.31 must be zero, ESA/390 PSW stored; one: z/Architecture PSW stored)

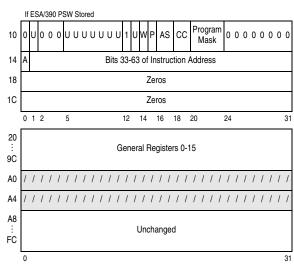
0.14 (R) General-register control (zero: bits 32-63 stored; one: bits 0-63 stored)

/// Available for programming

Trap Save Area

Byte (Hex)





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 54)

U Unpredictable

/// Available for programming

Trace-Entry Formats

Identification of Trace Entries

Trace-Entry Bits		s	Trace Entry		
0-7	8-11	12-15	Туре	Format	
00000000			Branch	1	
00010000		000N	Set Secondary ASN	1	
00100001			Program Call	11	
00100010			Program Call	21	
00100001		0	Program Call	3 ¹	
00100010		0	Program Call	41	
00100010		100E	Program Call	5 ¹	
00100010		101E	Program Call	6 ¹	
00100011		111E	Program Call	7 ¹	
00110001		000N	Program Transfer	1	
00110001		100N	Program Transfer	2	
00110010		0000	Program Return	1	
00110010		0010	Program Return	2	
00110010		1000	Program Return	4	
00110010		1010	Program Return	5	
00110010		110N	Program Transfer	3	
00110011		0011	Program Return	3	
00110011		1011	Program Return	6	
00110011		1100	Program Return	7	
00110011		1110	Program Return	8	

Trace-Entry Bits		s	Trace Entry		
0-7	8-11	12-15	Туре	Format	
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		
01010010	1100		Branch		
01010010	1111		Mode-Switching Branch	3	
0111	0		Trace	1	
0111	1		Trace	2	
1			Branch		

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

Branch

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

0000000		Bits 40-63 of Branch Address	
0	8		31

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



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

010100101100	All Zeros	Bits 0-31 of Branch Address
0 8	12	32 63
Bits 32-6	63 of Branch Address	
64	95	

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

Branch in Subspace Group (if ASN Tracing on)

F1 (in 24/31-Bit Mode)

01	000001P	Bits 9-31 of ALET	Α	Bits 33-63 of Branch Address	
0	8	•	3		63

F2 (in 64-Bit Mode)

01000010P	Bits 9-31 of ALET		Bits 33-63 of Branch Address	
0 8		32		63
Bits 32	-63 of Branch Address			
64	(95		

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

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

Mode Switch

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

01010001	0011		All Zeros	Α	Updated Instruction Address	
0	8	12		32		63

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

010100	01001	0	All Zeros	Bits 32-63 of Updated Inst. Address	
0	8	12		32	63

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

010	100100110)	All Zeros		Bits 0-31 of Updated Inst. Addres	s
0	8	12		3	32	63
	Bits 32-63	of Upd	ated Inst. Address			
64				95	į	

Mode-Switching Branch

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

0101	0001101	0	All Zeros	А	Branch Address	
0	8	12		32		63

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

01	010001	1011		All Zeros	Bits 32-63 of Branch Address	
0	8	3	12		32	63

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

010100101111	All Zeros	Bits 0-31 of Branch Address
0 8 1	2	32 63
Bits 32-63	3 of Branch Address	
64	95	•

Program Call

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

00100	001 PSW Key		All Zeros	A	Bits 33-62 of Return Address	F
0	8	12		32		63

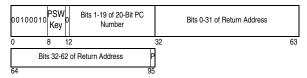
F2 (in 64-Bit Mode, ALRF Not Enabled)

00100010 PSW Key	All Zeros	Bits 0-31 of Return Address	
0 8 12	2	32	63
Bits 32-62	of Return Address	P	
64	(5	

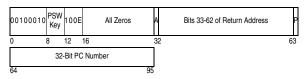
F3 (in 24/31-Bit Mode, ALRF Enabled, 20-Bit PC Number)

0010000	1 PSV Key	0	Bits 1-19 of 20-Bit PC Num- ber	Α	Bits 33-62 of Return Address	P
0	8	1	2	32		63

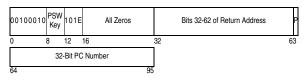
F4 (in 64-Bit Mode, ALRF Enabled, 20-Bit PC Number)



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



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

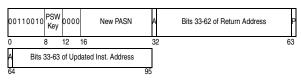


F7 (in 64-Bit Mode, ALRF Enabled, 32-Bit PC Number, Bits 0-31 of Return Address Not All Zeros)

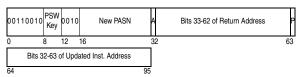
00100011	PSW Key	111E	All Zeros		Bits 0-31 of Return A	ddress
0	8	12	16		32	63
Bit	s 32-6	2 of R	eturn Address	Р	32-Bit PC Numb	er
64					96	127

Program Return

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



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



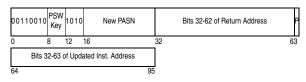
F3 (in 64-Bit to 24/31-Bit Mode)

00110	011 PSW Key	001	1	New PASN	А	Bits 33-62 of Return Address	P
0	8	12	16		32		63
				Updated Ir	struction	Address	
64							127

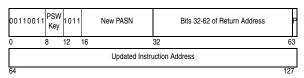
F4 (in 24/31-Bit to 64-Bit Mode)

0011	10010 PSW Key	1000	New PASN		Bits 32-62 of Return Address	P
0	8	12	16	32		63
A	Bits 33-63	of Upo	dated Inst. Address			
64			9	95		

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



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



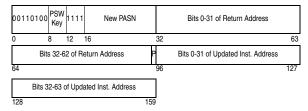
F7 (in 24/31-Bit to 64-Bit Mode)

00110011	PSW Key	1100	New PASN			Bits 0-31 of Return Address	
0	8	12	16		32		63
Bit	s 32-6	2 of R	eturn Address	Р	Α	Updated Instruction Address	
64					96		127

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

0011	0011 PSW Key	1110	New	PASN		Bits 0-31 of Return Address	
0	8	12	16		32		63
	Bits 32-6	32 of R	eturn Addre	ss F	o	Bits 32-63 of Updated Inst. Address	
64					96		127

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



Program Transfer

F1 (in 24/31-Bit Mode)

001	10001	PSW Key	000N	New PASN		Bits 32-63 of R2 Before	
0		8	12	16	32		63

F2 (in 64-Bit Mode, Bits 0-31 of R2 All Zeros)

00	0110001	PSW Key	100N		New PASN		Bits 32-63 of R2 Before	
0		8	12	16		32		63

F3 (in 64-Bit Mode, Bits 0-31 of R2 Not All Zeros)

00110001	PSW Key	100N	New PASN	Bits 0-31 of R2 Before	
0	8	12	16	32	63
	Bits 3	2-63 c	f R2 Before		
64			9	5	

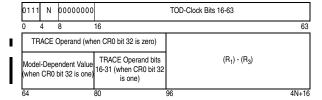
Set Secondary ASN

F1

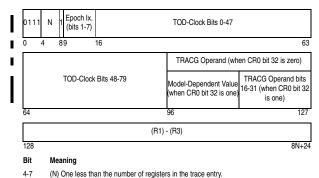
00010000	000000N	New SASN	
0	8	16	31

Trace

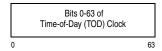
F1 (TRACE)



F2 (TRACG)



Operand of Store Clock and Store Clock Fast



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

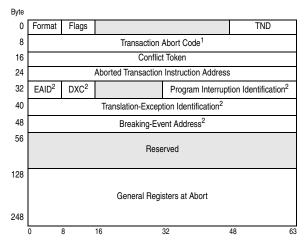
Operand of Store Clock Extended

Epoch Index	Time-of-Day (TOD) Clock	Programmable Field	
0	8	112	127

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

Transaction Diagnostic Block (TDB)

TBEGIN-specified TDB is the operand of the TBEGIN instruction when B1 \neq 0; Program-interruption TDB is at real locations 6,144 - 6,399.



Explanation:

- 1 Transaction abort codes:
 - 2 External interruption
 - 4 Non-filtered program interruption
 - 5 Machine-check interruption
 - 6 I/O interruption
 - 7 Fetch overflow
 - 8 Store overflow
 - 9 Fetch conflict 10 - Store conflict

- 11 Restricted instruction12 Filtered program inter
- 12 Filtered program interruption
- 13 Nesting depth exceeded
- 14 Cache fetch-related condition15 Cache store-related condition
- 16 Cache other condition
- 255 Miscellaneous condition
- >255 TABORT instruction
- Field is stored only in the TBEGIN-specified TDB; otherwise, the field is reserved. The program interruption identification is only stored for program-interruption conditions. The EAID and translation-exception identification are stored only for access-list-controlled or DAT protection, ASCE-type, page translation, region-first translation, region-second translation, region-third translation, and segment translation program-interruption conditions. The DXC is stored only for data program-exception conditions.

TND Transaction nesting depth

Guarded-Storage Facility Registers and Parameters

Guarded-Storage-Designation (GSD) Register

Guarded-Storage Origin (GSO)																				
0													31							
GSO (continued)	/	/	/	/	/	/	/	/	/	/	/	/	/	/	GLS	/	/	GS	SC	
32															53	56	5	58		63
Dit Maaning																				

Bit

- Meaning
 Guarded-storage origin (GSO), where J is 64-GSC 0-J
- 53-55 Guarded-load shift (GLS); valid values are 0-4
 58-63 Guarded-storage characteristic (GSC); valid va
- Guarded-storage characteristic (GSC); valid values are 25-56

Guarded-Storage Control Block

	Reserved	
	i leaci veu	
0		63
	Guarded-Storage-Designation (GSD) Register (see above)	
64		127
	Guarded-Storage-Section-Mask (GSSM) Register	
128		191
	Guarded-Storage-Event-Parameter-List-Address (GSEPLA) Register (see below)	
192		255

Guarded-Storage-Event Parameter List

			GSEAM							(GSE	ECI					GSEAI				
Reserved		0 0	0	0	0 0	Е	В	T X	C	0	0	0	0	0	N	N 0 T AS		AR			
			Reserved																		
Guarded-Storage-Event Handler Address (GSEHA)																					
	Guarded-Storage-Event Instruction Address (GSEIA)																				
	G	auarde	d-S	Stora	age-E	ver	nt C)pei	ran	d A	ddr	ess	(G	SE	O.F	١)					
	Guarded-Storage-Event Intermediate Result (GSEIR)																				
Guarded-Storage-Event Return Address (GSERA)																					
Bits Meaning																					

Bits	Meaning
0-7	Reserved
8-15	Guarded-storage-event addressing mode (GSEAM)
	14 - Extended-addressing mode (E)
	15 - Basic Addressing mode (B)
16-23	Guarded-storage-event cause indication (GSECI)
	16 - CPU was in the transactional-execution mode (TX)
	17 - CPU was in the constrained transactional-execution mode (CX)
	23 - Instruction causing the event; 0-LGG, 1=LLGFSG
24-31	Guarded-storage-event access information (GSEAI)
	25 - DAT mode (copy of PSW bit 5)
	26-27 - Address-space indication (copy of PSW bits 16-17)
	28-31 - AR number (when in the AR mode; otherwise unpredictable
32-63	Reserved
64-127	Guarded-storage-event handler address (GSEHA)
128-191	Guarded-storage-event instruction address (GSEIA)
192-255	Guarded-storage-event operand address (GSEOA)
256-319	Guarded-storage-event intermediate result (GSEIR)
320-383	Guarded-storage-event return address (GSERA)

Operation-Request Block (ORB)

Command-Mode ORB

Word

ioiu														
0		Interruption	Parameter											
1	Key SCMYFPIAUBHT LPM LD0000													
2	0 Channel-Program Address													
3	CSS Priority Reserved CU Priority Reserved													
4	Reserved													
5		Reserved												
6		Reserved												
7		Reserved												
	0 8 16 24													

Transport-Mode ORB

Word.Bit Meaning 1.0-3 (Key) Subchannel key

Word

voiu															
0				lr	nte	rru	ption	Parameter							
1	Key	0 0 0 0	0 0	0 0	0	В	0 0	LPM	0 (0 (0	0	0 (X	
2	0 Channel-Program Address														
3	CSS Priority Reserved Reserved for Pgm. Reserved														
4		Reserved													
5							Rese	erved							
6		Reserved													
7		Reserved													
	0 8 16 24												31		

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
1.13	(B) Channel-Program Type
1.14	(H) Format-2-IDAW control
1.15	(T) 2K-IDAW control
1.16-23	(LPM) Logical-path mask
1.24	(L) Incorrect-length-suppression mode
1.25	(D) Modified-CCW-indirect-data-addressing control
1.31	(X) ORB-extension control
3.0-7	Channel-subsystem priority
3.16-23	Control-unit priority

Channel-Command Word (CCW)

Format-0 CCW

Co	ommand Code		Data Address			
0		8		31		
	Flags		Byte C	ount		
32		40	48	63		
Bit	Meaning					
32	(CD) Cause	es use of data-	address portion of next CCW			
33	(CC) Cause	(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) Caus	es an interme	diate-interruption condition to occur			
37	(IDA) Caus	es bits 8-31 of	CCW to specify location of first IDAW			

(Suspend) Causes suspension before execution of this CCW

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

Format-1 CCW

38

39

Cor	nmand Code	Flags		Byte Count			
0	8		16		3		
0			Data Address				
32					6		
<u>Bit</u>	Meaning						
8	(CD) Causes u	se of data-addres	ss portion of next (CCW			
9	(CC) Causes u	se of command c	ode and data add	ress of next CCW			
10	(SLI) Causes s	(SLI) Causes suppression of possible incorrect-length indication					
11	(Skip) Suppres	ses transfer of inf	formation to main	storage			
12	(PCI) Causes a	(PCI) Causes an intermediate-interruption condition to occur					
13	(IDA) Causes b	its 33-63 of CCW	I to specify locatio	n of first IDAW			
14	(Suspend) Cau	ses suspension b	pefore execution o	f this CCW			
15	(MIDA) Causes	bits 33-63 of CC	CW to specify locat	tion of first MIDAW			

Indirect-Data-Address Word (IDAW)

Format-1 IDAW

0	Data Address
_	1 21

Format-2 IDAW

E	Bits 0-31 of Data Address
0	31
В	its 32-63 of Data Address
32	63

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

	Reserved						
0					31		
	Reserved	Flags		Count			
32		40	48		63		
	Bits 0-31 of Data Address						
64					95		
		Bits 32	2-63 of Data Addres	SS			
96					127		
Bit 40	Meaning Last MIDAW	,					

41 Skip
42 Data-transfer-interruption control
43-47 Reserved

Transport Control Word (TCW)

	•		`		•		
Word							
0	F	000000	Flags				
1		Reserved	TCCBL	RW	Re	eserved	
2			0		- A -l-l		
3			Outp	ul-Dal	a Address		
4			la a c	. D-1-	Address		
5			inpu	t-Data	Address		
6			Transport	Ctatua	Diagle Address		
7			ransport-	Status	-Block Address		
8	T 10 10 1 18 1 1 1 1						
9	Transport-Command-Control-Block Address						
10	Output Count						
11			I	nput (Count		
12							
				Rese	rved		
14							
15	Interrogate-TCW Address						
	0	2	8	14 15 1	6	24	31

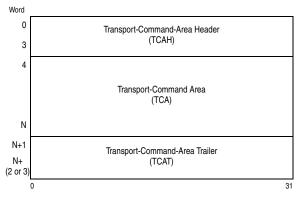
Word.Bit	Meaning
0.0-1	Format
0.13	Input transport-indirect-data addressing (TIDA)
0.14	Transport-command-control-block TIDA
0.15	Output TIDA
0.16-17	TIDAW Format
1.8-13	(TCCBL) Transport-Command-Control-Block Length
1.14	(R) Read Operations
1.15	(W) Write Operations

Transport-Indirect-Data-Address Word (TIDAW)

	Flags	Reserved	
0		8	31
		Count	
32		48	63
		Bits 0-31 of Data Address	
64			95
		Bits 32-63 of Data Address	
96			127
Bit	Meaning		
0	Last TIDA Skip		
1	σκιρ		

- Data-transfer-interruption control
- (TTIC) TIDAW Transfer In Channel 3
- Insert CBC Control
- 5-7 Reserved

Transport Command Control Block (TCCB)



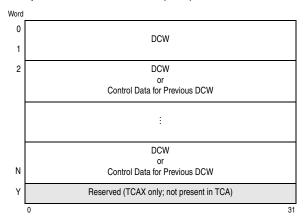
Transport Command Area Header (TCAH)

Word		•			
0	Format	Reserved			
1		TCAL			
2	Service-A	Service-Action Code Reserved			
3	Reserved				
	0	8	16	24	31

Word.Bit Meaning

1.24-31 (TCAL) Transport-Command-Area Length

Transport-Command Area (TCA) and Transport-Command-Area Extension (TCAX)



Device-Command Word (DCW)

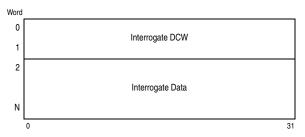
, , ,					
Command Code	Flags	Rese	rved Control-D	ata Count	
0	8	16	24	31	
		Count			
32				63	

Bit Meaning

9 (CC) Causes use of next DCW

10 (SLI) Suppresses incorrect-length indication

Interrogate TCA



Interrogate Data

Word						
0	Format	RC	RCQ	LPM		
1	PAM	PIM	PIM Timeout			
2	Flags		Reserved			
3		Res	served			
4		Т	imo			
5	Time					
6	Program Identifier					
7	Program identilier					
8						
	Program-Dependent Data					
N						
	0	8	16	24 31		

Word.Bit	Meaning
0.8-15	(RC) Reason code
	 Interrogate reason not specified
	1 Timeout
0.16-23	(RCQ) Reason-code qualifier
	 Interrogate reason qualifier not specified
	1 Primary
	2 Secondary
0.24-31	(LPM) Logical-path mask
1.0-7	(PAM) Path-available mask
1.8-15	(PIM) Path-installed mask
2.0-7	Flags
	Multipath mode
	 Program path recovery
	2 Critical

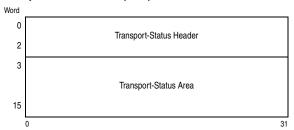
Transport Command Area Trailer (TCAT)

Word	
0	Reserved
1	Write Count or Transport Count
2	Read Count (or not present)
	0

CBC-Offset Block (COB)

Word	,
0	CBC Offset 0
1	CBC Offset 1
2	·
	÷
N	CBC Offset N
Υ	Reserved
	0 31

Transport Status Block (TSB)



Transport Status Header (TSH)

٠.			
V	V٨	ra	

vvora				
0	Length	Flags	DCW Offset	
1		Co	unt	
2		Rese	erved	
	0	8	16	31

Word.Bit Meaning

0.8 DCW-offset field valid 0.9 Count field valid 0.10 Cache miss 0.11 Time fields valid

0.13-15 Transport-Status Area (TSA) Format

TSA contents have no meaning

1 I/O-status TSA

2 Device-detected-program-check TSA

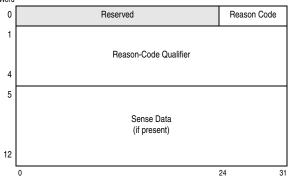
3 Interrogate TSA

I/O-Status TSA

Word	
0	Device Time
1	Defer Time
2	Queue Time
3	Device-Busy Time
4	Device-Active-Only Time
5	
	Additional Data (if present)
12	
	0 31

Device-Detected-Program-Check TSA





Word.Bit M

Meaning

0.24-31 (RC) Reason Code

- No information
 - 1 TCCB transport failure

(RCQ) Reason-code-qualifier byte 0 (1.0-7)

- 0 No additional information
- 1 TCCB transport size error
- 2 TCCB CBC error
- 2 Invalid CBC detected on output data

RCQ word 0: Offset of first output-data byte for which error was detected RCQ word 1: Offset of last output-data byte for which error was detected

3 Incorrect TCCB length specification

RCQ byte 0

- 0 No additional information
- 1 TCAL value not 8 greater than TCW TCCBL value
- 2 TCAL value is less than 20 or greater than 252

4 TCAH specification error

RCQ byte 0

- No additional information
- 1 Format field specification error
- 2 Reserved field specification error
- 3 Service-action-code field specification error
- 5 DCW specification error

RCQ byte 0

- No additional information
- 1 Reserved field specification error
- 2 Flags field command-chaining specification error
- 3 Control-data-count field specification error
- 4 TCOB location error
- 5 TCOB duplication error
- 6 TCOB multiple-count error
 - 7 TCOB direction error
- 8 TCOB chaining error
- 9 TCOB count-specification error
- 10 TTE location error
- 11 TTE duplication error
- 12 TTE CD-count specification error
- 13 TTE count specification error
- 14 TTE direction error:
- 15 TTE chaining error
- 16 TCAX specification error

6 Transfer-direction specification error

RCQ byte 0

- No additional information
- 1 Read-direction specification error
- 2 Write-direction field specification error
- 3 Read-write-conflict specification error

7 Transport-count specification error

RCQ byte 0

- 0 No additional information
- 1 Read-count specification error
- 2 Write-count specification error
- 8 Two I/O operations active

RCQ: No additional information

CBC-offset specification error

RCQ word 0: Byte offset of COB CBC-offset entry

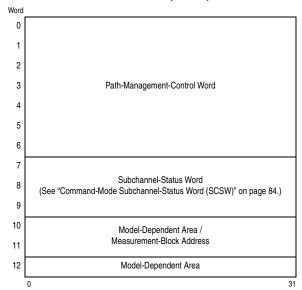
Interrogate TSA

rioru											
0	Format	Flags	Device Status								
1	Operation State	Reserved									
2											
	State-Dependent Information										
4											
5		Device-Le	vel Identifier								
6											
	Device-Dependent Information										
12											
	0	8	16 2	24 31							

Word.Bit	Meaning
0.8	Control-unit state valid
0.9	Device-state valid
0.10	Operation-state valid
0.16-23	(CS) Control-unit state
	0 Busy
	1 Recovery
	2 Interrogate maximum

- 0.24-31 (DS) Device-unit state
 - 0 Path-Group identification (in state-dependent-information field)
 - Path-Group
 Long busy
 - 2 Recovery
- 1.0-7 (OS) Operation state
 - 0 No I/O operation present.
 - 1 An I/O operation is present and executing.
 - 2 An I/O operation is present and awaiting completion of another operation initiated by another configuration.
 - 3 An I/O operation is present and awaiting completion of another operation initiated for the same device extent.
 - 4 An I/O operation is present and waiting to perform a device-dependent operation.

Subchannel-Information Block (SCHIB)



Path-Management-Control Word (PMCW)

Word																									
0	Interruption Parameter																								
1	0 0 ISC 0 0 0 E LM MM D T V Device Number																								
2	LPM PNOM											L	.PL	JM						PΙ	M				
3	MBI											POM PAM							М						
4		CHPID	-0			CI	HPID	-1			CHPID-2 CHPID								-3						
5		CHPID-4 CHPID-5									CHPID-6 CHF							ΙP	PID-7						
6	0 0	0 0 0	0	0 0	0	0 0	0 0	0	0	0	0 0	0	0	0 0	(0	0	0	0	0	0	F	Χ	S	
	0				8						16						24							31	

5		CHPID-4								CHPID-5								CHPID-6								CHPID-					
6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
	0								8							16								24							
Word.Bit Meaning																															
1.2-4	 -4 (ISC) Interrup 								ion-subclass code																						
1.8	(E) Subchann								el enabled																						
1.9-10	-10 (LM) limit mo								е																						

1.11-12

00 No Checking

01 Data address must be ≥ limit

10 Data address must be < limit

11 Reserved

(MM) Measurement-mode enable

00 Neither mode enabled

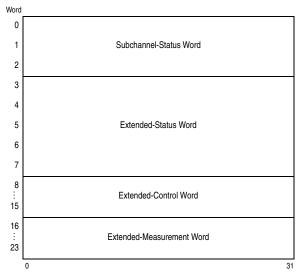
01 Device-connect-time-measurement enabled

10 Measurement-block-update enabled

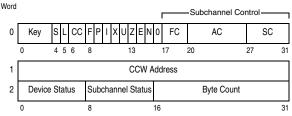
11 Both modes enabled

1.13	(D) Multipath mode
1.14	(T) Timing facility available
1.15	(V) Device number valid
2.0-7	(LPM) Logical-path mask
2.8-15	(PNOM) Path-not-operational mask
2.16-23	(LPUM) Last-path-used mask
2.24-31	(PIM) Path-installed mask
3.0-15	(MBI) Measurement-block index
3.16-23	(POM) Path-operational mask
3.24-31	(PAM) Path-available mask
4.0-7	(CHPID-0) Channel-path ID for logical path 0 (typical)
6.29	(F) Measurement-block-format control
6.30	(X) Extended-measurement-word-mode enable
6.31	(S) Concurrent sense

Interruption-Response Block (IRB)



Command-Mode Subchannel-Status Word (SCSW)



Word.Bit	Meaning
0.0-3	(Key) Subchannel key
0.4	(S) Suspend control
0.5	(L) Extended-status-word format (logout stored)
0.6-7	(CC) Deferred condition code
	00 Normal I/O interruption
	01 Status in SCSW
	10 Reserved
	11 Path not operational
0.8	(F) CCW-format control
0.9	(P) Prefetch control

0.10	(I) Initial-status-interruption control	
0.11	(X) IRB-format control	
0.12	(U) Suppress-suspended-interrupt	on control
0.13	(Z) Zero condition code	
0.14	(E) Extended control (information s	stored in ECW of IRB)
0.15	(N) Path not operational (PNOM no	onzero)
0.17-19	(FC) Function control	
	17 (40) Start, 18 (20) Halt, 19	9 (10) Clear
0.20-26	(AC) Activity control	
	20 (08) Resume pending	24 (80) Subchannel active
	21 (04) Start pending	25 (40) Device active
	22 (02) Halt pending	26 (20) Suspended
	23 (01) Clear pending	
0.27-31	(SC) Status control	
	27 (10) Alert	30 (02) Secondary
	28 (08) Intermediate	31 (01) Status pending
	29 (04) Primary	
2.0-15	Device status (0-7)	Subchannel status (8-15)
	0 (80) Attention	8 (80) Program-controlled interruption
	1 (40) Status modifier	9 (40) Incorrect length
	2 (20) Control-unit end	10 (20) Program check
	3 (10) Busy	11 (10) Protection check
	4 (08) Channel end	12 (08) Channel-data check
	5 (04) Device end	13 (04) Channel-control check
	6 (02) Unit check	14 (02) Interface-control check
	7 (01) Unit exception	15 (01) Chaining check

Transport-Mode Subchannel-Status Word (SCSW)

	isport-ivi	00	ш	Sub	cnann	lei.	-3	ld	เนร	vvc	ora (S	CSW)			
Word												-Subchar	nnel Co	ntrol	_
0	Key	0	L	CC	FMT	Χ	Q	0	ΕN	0	FC	A		SC	
	0	4	5	6	8	11		13			17	20		27	31
1								T	CW A	١dc	Iress				
2	Device	S	tat	us	Subcha	anr	nel	Si	tatus		FCX	Status	S	CHXS	
	0				8					16		·	24		31

Meaning	
(Key) Subchannel key	
(L) Extended-status-word format (logout sta	ored)
(CC) Deferred condition code	
00 Normal I/O interruption	
01 Status in SCSW	
10 Reserved	
11 Path not operational	
(FMT) Format	
(X) IRB-format control	
(Q) Interrogate complete	
(E) Extended control (information stored in	ECW of IRB)
(N) Path not operational (PNOM nonzero)	
(FC) Function control	
17 (40) Start, 18 (20) Halt, 19 (10) Cl	ear
(AC) Activity control	
21 (04) Start pending	23 (01) Clear pending
22 (02) Halt pending	25 (40) Device active
(SC) Status control	
27 (10) Alert	30 (02) Secondary
28 (08) Intermediate	31 (01) Status pending
29 (04) Primary	· · · · · · ·
	(L) Extended-status-word format (logout str (CC) Deferred condition code 00 Normal I/O interruption 01 Status in SCSW 10 Reserved 11 Path not operational (FMT) Format (X) IRB-format control (Q) Interrogate complete (E) Extended control (information stored in (N) Path not operational (PNOM nonzero) (FC) Function control 17 (40) Start, 18 (20) Halt, 19 (10) Cl (AC) Activity control 21 (04) Start pending 22 (02) Halt pending (SC) Status control 27 (10) Alert 28 (08) Intermediate

2.0-15	Device status (0-7)	Subchannel status (8-15)
	0 (80) Attention	8 (80) —
	1 (40) —	9 (40) Incorrect length
	2 (20) Control-unit end	10 (20) Program check
	3 (10) Busy	11 (10) Protection check
	4 (08) Channel end	12 (08) Channel-data check
	5 (04) Device end	13 (04) Channel-control check
	6 (02) Unit check	14 (02) Interface-control check
	7 (01) Unit exception	15 (01) Channel-subsystem retry failed
2.16-23	FCX status (16-23)	
	23 (01) TSB valid	

2.24-31

(SCHXS) Subchannel-extended status

24 (80) (F) Interrogate failed

25-31 (SESQ) SCHSX qualifier

- No status available.
- 1 Storage-request limit exceeded.
- 2 Program check when not an interrogate operation, TCW read/write data count not zero, and CE only or CE+DE only status received.
- 3 Transport mode not supported by the I/O device.
- 4 Transport mode not supported by the selected channel path.
- Program check on TCW. 6
- Device-detected program check condition due to indeterminate 7 cause.
- 8 Device-detected program check.
- Program check on TIDAW failing-storage-address (FSA) valid in ESW (see below) and contains TIDAW address.
- 32 TCW access exception - FSA field valid and contains TCW address.
- 33 TSB access exception - FSA field valid and contains TSB address.
- TCCB access exception FSA field valid and contains TCCB 34 address.
- 35 TIDAW access exception - FSA field valid and contains TIDAW address
- Data access exception FSA field valid and contains address of 36 data.
- 64 Invalid CBC error on read data.
- 66 Link protocol error condition.
- 67 Device-level recovery operation failed.
- IFCC due to failed device-level recovery operation program, pro-68 tection, or data check may also be set in subchannel status.
- Invalid CBC on status portion of transport response from device.
- Invalid CBC on TSB transported from device.
- Note: If FSA field valid for cases other than noted above, FSA field contains address of current TCW.

Extended-Status Word (ESW)

See chart on page 88 to determine the appropriate ESW format.

Format-0 ESW

Word

0	Subchannel Logout
1	Extended-Report Word
2	Failing-Storage Address
3	r alling-oldrage Address
4	Secondary-CCW Address
,	0 31

Format-0 ESW Word 0 (Subchannel Logout)

0	ESF	LPUM	R	FVF	SA	TC	D	Ε	Α	SC
0	1	8	16		22	24	26		28	31

Bit Meaning

24-25

1-7 (ESF) Extended-status flags (1 key check, 2 measurement-block program check, 3 measurement-block data check, 4 measurement-block protection check, 5 CCW check, 6 IDAW check, 7:0)

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

16 (R) Ancillary Report

17-21 (FVF) Field-validity flags (17 LPUM, 18 TC, 19 SC, 20 device status, 21 CCW address)

22-23 (SA) Storage-access code (00 access type unknown, 01 read, 10 write, 11 read backward)

(TC) Termination code (00 halt signal issued, 01 stop, stack, or normal termination, 10 clear signal issued)

26 (D) Device status check

27 (E) Secondary error 28 (A) I/O-error alert

29-31 (SC) Sequence code

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

0 L	E A P	T F S C R	SCNT	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	3	8 1	10	16															31

Bit Meaning

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

Format-1 ESW Word 01

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

Bit Meaning

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

Format-2 ESW Word 0¹

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

Bit Meaning

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

16-31 (DCTI) Device-connect-time interval

Format-3 ESW Word 01

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

Bit Meaning

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

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

Information Stored in ESW

Subchannel Conditi		nich ESW Is	Stored by Te	est Subchan-			
Subchannel-Sta	atus Word	Path-Manag trol V	ement-Con- Vord		Extended-9	Status Wo SW)	rd
Status-Control Field A I P S X L 0 * * 0 0 1 * * 1 * 1 1 0 0 0 1 1 0 0 0 0 1 1 1 0 0 0 1 1 * * 1 * 1 1 1 0 0 1 1 0 1 0 0 1 0 1 0 0 1 0 1 0 0 1 0 1 0 0 1 0 1 0 0 1 0 1 0 0 1	Sus-pended Bit - * 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1		Device- Connect- Time Mea- surement- Mode		Format U 0 0 0 U 3 3 1 1 1 2 U 1 1 1 2	Contented OF The Content of the Cont	R R R * * * Z Z Z D * Z Z Z
0 0 0 1 1	1 0	•	•	ations do not	_	ZMD	D
* 1 0 1 1	*						

Bit Meaning

- Not meaningful.
- Bits may be zeros or ones.
- Α
- D Accumulated device-connect-time-interval (DCTI) value stored in bytes 2 and 3.
- Intermediate status.
- L Extended-status-word format.
- М Last-path-used mask (LPUM) stored in byte 1.
- Р Primary status.
- R Subchannel-logout information stored in bytes 0-3.
- S Secondary status.
- U No format defined. Х
- Status pending.
- Bits are stored as zeros.

Extended-Control Word (ECW)

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

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

Extended-Measurement Word

Word	
0	Device-Connect Time
1	Function-Pending Time
2	Device-Disconnect Time
3	Control-Unit-Queuing Time
4	Device-Active-Only Time
5	Device-Busy Time
6	Initial-Command-Response Time
7	Reserved
	0 31

Format 0 Measurement Block

Word		
0	SSCH + RSCH Count	Sample Count
1	Device-Co	onnect Time
2	Function-P	ending Time
3	Device-Disc	connect Time
4	Control-Unit-	Queuing Time
5	Device-Acti	ve-Only Time
6	Device-E	Busy Time
7	Initial-Command	d-Response Time
	0	16 31

Format 1 Measurement Block

Word	
0	SSCH + RSCH Count
1	Sample Count
2	Device-Connect Time
3	Function-Pending Time
4	Device-Disconnect Time
5	Control-Unit-Queuing Time
6	Device-Active-Only Time
7	Device-Busy Time
8	Initial-Command-Response Time
9	Interrupt Delay Time
10	I/O Priority Delay Time
11	
:	Reserved
15	
I	0 31

Channel-Report Word (CRW)

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

<u>SIT</u>	Meaning

- 1 (S) Solicited CRW
- (R) Overflow (one or more CRWs lost)(C) Chaining (meaningless if bit 2 is one)

Condition

- 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

0	0	0	0	0	1	Available
0	0	0	0	1	0	Initialized
0	0	0	0	1	1	Temporary error
0	0	0	1	0	0	Installed parameters initialized
0	0	0	1	0	1	Terminal
0	0	0	1	1	0	Permanent error with facility not initialized
0	0	0	1	1	1	Permanent error with facility initialized
0	0	1	0	0	0	Installed parameters modified

Reporting Source

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

RSC	Reporting Source	Reporting-Source ID	
0010	Monitoring facility	00000000 0000000	0
0 0 1 1	Subchannel (first or only CRW)	XXXXXXXX XXXXXX	Χ
0 0 1 1	Subchannel (chained CRW)	00000000 0088000	0
0 1 0 0	Channel path	00000000 YYYYYY	Υ
1001	Configuration-alert facility	00000000 YYYYYY	Υ
1011	Channel subsystem	00000000 0000000	0

 $^{{\}sf S}={\sf Subchannel\text{-}set}$ identifier (SSID) when the MSS facility is installed and the CRW is chained immediately following a CRW for a subchannel.

I/O Command Codes

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

x x x x 0 0 0 0 Invalid Command	mmmm 0 1 0 0 Sense
mmmm mm0 1 Write (a)	0 0 0 0 0 1 0 0 — Basic Sense
mmmm mm1 0 Read (a)	1 1 1 0 0 1 0 0 — Sense ID
0 0 0 0 0 0 1 0 — Read IPL	x x x x 1 0 0 0 Transfer in channel (c)
mmmm mm1 1 Control	0 0 0 0 1 0 0 0 Transfer in channel (d)
0 0 0 0 0 0 1 1 — Control no operation	mmmm 1 0 0 0 Invalid command (e)
0 1 mm 0 0 0 0 Transport (b)	mmmm 1 1 0 0 Read backwards (f)
x - Bit Ignored m - Modifier bit for specific type of I/O device	a May designate control data in a DCW b DCW only c Format-0 CCW d Format-1 CCW e Format-1 CCW and nonzero m bit f CCW only

Standard Meanings of Bits of First Sense Byte

	۰	iara moainingo or Bito or i not	000	0 2) 10
В	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)

X = Subchannel number

Y = Channel-path ID (CHPID)

Hexadecimal and Decimal Conversion

Use the following figure to preform hexadecimal and decimal conversions.

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 "Character Assignments" on page 96.

				_	0	-	5	က	4	2	9	7	∞	၈	10	7	12	13	14	15		
			4567	Dec																		
		Byte	4	Hex	0	1	2	3	4	2	9	2	8	6	A	В	ပ	Ω	ш	ш		
		B	3	Dec	0	16	32	48	64	80	96	112	128	144	160	176	192	208	224	240		
			0123	Нех	0	-	2	3	4	2	9	7	8	6	٧	В	ပ	٥	ш	ш	2	
	Halfword		4567	Dec	0	256	512	292	1,024	1,280	1,536	1,792	2,048	2,304	2,560	2,816	3,072	3,328	3,584	3,840	3	
			45	Нех	0	-	2	3	4	2	9	7	8	6	A	В	ပ	٥	ш	ш		
		Byte	Byte	0123	Dec	0	4,096	8,192	12,288	16,384	20,480	24,576	28,672	32,768	36,864	40,960	45,056	49,152	53,248	57,344	61,440	4
			0	Hex	0	-	2	3	4	2	9	7	8	6	A	В	၁	٥	ш	ш		
Word				4567	Dec	0	65,536	131,072	196,608	262,144	327,680	393,216	458,752	524,288	589,824	092,360	720,896	786,432	851,968	917,504	983,040	5
×			4	Hex	0	-	2	3	4	2	9	7	8	6	Α	В	O	Ω	ш	ш		
		Byte	0123	Dec	0	1,048,576	2,097,152	3,145,728	4,194,304	5,242,880	6,291,456	7,340,032	8,388,608	9,437,184	10,485,760	11,534,336	12,582,912	13,631,488	14,680,064	15,728,640	9	
				Hex	0	1	2	3	4	5	9	7	8	6	Α	В	O	D	Е	F		
	Halfword		4567	Dec	0	16,777,216	33,554,432	50,331,648	67,108,864	83,886,080	100,663,296	117,440,512	134,217,728	150,994,944	167,772,160	184,549,376	201,326,592	218,103,808	234,881,024	251,658,240	7	
		_		Hex	0	-	2	3	4	2	9	7	8	6	Α	В	ပ	٥	ш	ш		
		Byte	0123	Dec	0	268,435,456	536,870,912	805,306,368	1,073,741,824	1,342,177,280	1,610,612,736	1,879,048,192	2,147,483,648	2,415,919,104	2,684,354,560	2,952,790,016	3,221,225,472	3,489,660,928	3,758,096,384	4,026,531,840	8	
			Bits:	Нех	0	-	2	က	4	2	9	7	80	6	ď	Ф	O	Ω	ш	ш		

Powers of 2 and 16

m	n	2 ^{<i>m</i>} and 16 ^{<i>n</i>}	Symbol
0	0	1	
1		2	
2		4	
3	1	8 16	
5	'	32	
6		64	
7		128	
8	2	256	
9		512	
10		1 024	K (kilo)
11	•	2 048	
12	3	4 096 8 192	
13 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	6	8 388 608 16 777 216	
25	0	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 35		17 179 869 184 34 359 738 368	
36	9	68 719 476 736	
37	Ů	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	11	8 796 093 022 208 17 592 186 044 416	
44	11	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 54		9 007 199 254 740 992 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	

т	n										2 ^m a	and	16 ⁿ	Symbol
64	16						18	446	744	073	709	551	616	
65							36	893	488	147	419	103	232	
66							73	786	976	294	838	206	464	
67							147	573	952	589	676	412	928	
68	17						295	147	905	179	352	825	856	
69							590	295	810	358	705	651	712	
70						1	180	591	620	717	411	303	424	Z (zetta)
71						2	361	183	241	434	822	606	848	
72	18					4	722	366	482	869	645	213	696	
73						9	444	732	965	739	290	427	392	
74										478				
75										957				
76	19									914				
77										828				
78										657				
79										314				
80	20									629				Y (yotta)
81										258				
82										516				
83	0.1									033				
84	21									066				
85										133				
86										267				
87	00									534				
88	22									068				
89										137				(aaa mata)
90					237									(see note)
91	00				475									
92 93	23				951 903									
94					807									
95					614									
96	24				228									
97					456									
98					912									
99					825									
100	25		1		650									(see note)
101			2	535										(
102				070										
103				141										
104	26		20	282	409	603	651	670	423	947	251	286	016	
105			40	564	819	207	303	340	847	894	502	572	032	
106			81	129	638	414	606	681	695	789	005	144	064	
107			162	259	276	829	213	363	391	578	010	288	128	
108	27		324	518	553	658	426	726	783	156	020	576	256	
109			649	037	107	316	853	453	566	312	041	152	512	
110			298											(see note)
111			596											
112	28	Ę		296										
113			384											
114			769											
115	00	41		374										
116	29	83		749										
117			153											
118			306											
119	20		613											(ann note)
120	30	1 329												(see note)
121 122		2 658 5 316												
123		10 633												
	21	21 267												
124 125	31													
126		42 535 85 070												
127	32	170 141 340 282												

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

Character Assignments

Cilaia	0.0.	Assigninen						
Dec	Hex	EBCDIC ¹	ISO-8 ²		Dec	Hex	EBCDIC ¹	ISO-8 ²
0	00	NUL	NUL		64	40	SP	@
1	01	SOH	SOH		65	41	RSP	Ā
2	02	STX	STX		66	42	â	В
3	03	ETX	ETX		67	43	ä	С
4	04	SEL	EOT		68	44	à	D
5	05	HT	ENQ		69	45	á	E
6	06	RNL	ACK		70	46	ã	F
7	07	DEL	BEL		71	47	å	G
8	08	GE	BS		72	48	ç	H
		SPS	HT			49		ï
9	09				73		ñ	
10	0A	RPT	LF		74	4A	¢	J
11	0B	VT	VT		75	4B		K
12	0C	FF	FF		76	4C	<	L
13	0D	CR	CR		77	4D	(M
14	0E	SO	SO		78	4E	+	N
15	0F	SI	SI		79	4F	1	0
16	10	DLE	DLE		80	50	&	P
17	11	DC1	DC1		81	51	é	Q
18	12	DC2	DC2		82	52	ê	R
19	13	DC3	DC3		83	53	ë	S
20	14	RES/ENP	DC4		84	54	è	T
21	15	NL	NAK		85	55	í	U
22	16	BS	SYN		86	56	î	V
23	17	POC	ETB		87	57	ï	W
24	18	CAN	CAN		88	58	ì	Х
25	19	EM	EM		89	59	В	Ϋ́
26	1A	UBS	SUB		90	5A	!	Z
27	1B	CU1	ESC		91	5B	\$	[
28	1C	IFS	IFS		92	5C	*	\
29	1D	IGS	IGS		93	5D)]
30	1E	IRS	IRS		94	5E		٨
31	1F	ITB/IUS	IUS		95	5F		
32	20	DS	SP		96	60	-	<u>, </u>
33	21	SOS					/	
			!		97	61		a
34	22	FS			98	62	Â	b
35	23	WUS	#		99	63	Ä	С
36	24	BYP/INP	\$		100	64	À	d
37	25	LF	%		101	65	Á	е
38	26	ETB	&		102	66	Ã	f
39	27	ESC	1		103	67	Å	g
40	28	SA	(104	68	Ç	h
41						69	Ñ	
	29	SFE)		105			i
42	2A	SM/SW	1		106	6A	1	j
43	2B	CSP	+	١.	107	6B	,	k
44	2C	MFA	,		108	6C	%	1
45	2D	ENQ	-		109	6D	_	m
46	2E	ACK			110	6E	>	n
47	2F	BEL	/		111	6F	?	0
48	30		0	1	112	70	Ø	р
49	31		1		113	71	É	
50	32	SYN	2	1	114	72	Ê	q
							Ë	r
51	33	IR DD	3		115	73		S
52	24	PP	4		116	74	È	t
53	35	TRN	5		117	75	ĺ	u
54	36	NBS	6		118	76	Î	v
55	37	EOT	7		119	77	Ϊ	w
56	38	SBS	8	1	120	78	ì	х
57	39	IT	9		121	79		y
58	3A	RFF	:		122	7A	:	y Z
59	3B	CU3	;		123	7B	#	{
60	3C	DC4	<		124	7C	@	1
61	3D	NAK	=		125	7D	1	}
62	3E		>		126	7E	=	~
63	3F	SUB	?		127	7F	"	•

Dec	Hex	EBCDIC ¹	ISO-8 ²
128	80	Ø	
129	81	а	
130	82	b	BPH
131	83	С	NBH
132	84	d	IND
133	85	е	NEL
134	86	f	SSA
135	87	g	ESA
136	88	h	HTS
137	89	i	HTJ
138	8A	«	VTS
139	8B	>>	PLD
140	8C	ð	PLU
141	8D	ý	RI
142	8E	þ	SS2
143	8F	±	SS3
144	90	•	DCS
145	91	j	PU1
146	92	k	PU2
147	93	i.	STS
148	94	m	CCH
149	95	n	MW
150	96	0	SPA
151	97	p	EPA
152	98	q	SOS
153	99	r	
154	9A	a	SCI
155	9B	Ω.	CSI
156	9C	æ	ST
157	9D		OSC
158	9E	Æ	PM
159	9F	π. D	APC
160	A0	μ	RSP
161	A1	μ ~	i
162	A2	s	¢
163	A3	t	£
164	A4	u u	D
165	A5	v	¥
166	A6	w	+
167	A6 A7	w X	· §
168	A8		3
169	A9	у	©
170	AA	Z :	a a
171	AB	i	~ «
	AC	j G	-
172 173	AC AD	Ý	SHY
173	AE AE		® ®
175	AF	þ ®	-
176	B0	۸ ا	0
	B1	£	
177 178	B2	¥	± 2
178		*	3
180	B3 B4	©	,
		-	
181	B5	§ ¶	μ ¶
182	B6 B7	¶ 1/4	1
183		1/4	
184	B8	1/2	1
185	B9	3/4	1 0
186	BA	[
187 188	BB]	»
188	BC	ä	1/4
			1/2
189	BD	,	21
	BE BF	, X	3∕4 ¿

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

Notes:

ACK Acknowledge

- The EBCDIC characters are based on code page 037.
- 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.

ΙT

Indent Tab

Partial Line Down

Control Character Representations

BEL	Bell	ITB	Intermediate Transmission Block
BS	Backspace	IUS	International Unit Separator
BYP	Bypass	LF	Line Feed
CAN	Cancel	MFA	Modify Field Attribute
CR	Carriage Return	NAK	Negative Acknowledge
CSP	Control Sequence Prefix	NBS	Numeric Backspace
CU1	Customer Use 1	NL	New Line .
CU3	Customer Use 3	NUL	Null
DC1	Device Control 1	POC	Program-Operator Communication

DC1 DC2 DC3 DC4 DEL POC PP RES Program-Operator Communication Device Control 1 Device Control 2 Presentation Position Device Control 3 Restore Required Form Feed Device Control 4 RFF RNL Delete Required New Line DLE Data Link Escape RPT Repeat SA SBS DS Digit Select Set Attribute EM End of Medium Subscript

ENP Enable Presentation SEL SFE Select ENQ EO EOT Start Field Extended Enquiry Eight Ones Šİ Shift In SM End of Transmission Set Mode ESC Shift Out Escape ETB End of Transmission Block SOH Start of Heading ETX End of Text SOS Start of Significance FF Form Feed SPS Superscript FS Field Separator STX Start of Text GE Graphic Escape SUB Substitute Horizontal Tab SW HΤ Switch

IFS Interchange File Separator Interchange Group Separator SYN Synchronous Idle IGS TRN Transparent INP Inhibit Presentation **UBS** Unit Backspace IR Index Return VT Vertical Tab IRS Interchange Record Separator WUS Word Underscore

Additional ISO-8 Control Character Representations APC PLD

Application Program Command Break Permitted Here Cancel Character RPH PLU Partial Line Up CCH CSI DCS Privacy Message Private Use One Private Use Two PM Control Sequence Introducer PÜ1 Device Control String PU2 Single Character Introducer ESA End of Selected Area SCI Character Tabulation w/ Justification HTJ SOS Start of String Start of Guarded Area Start of Selected Area HTS Character Tabulation Set SPA IFS Information Separator Four SSA IGS Information Separator Three SS2 Single Shift Two IND SS3 Single Shift Three Index IRS Information Separator Two ST String Terminator Set Transmit State STS MW Message Waiting No Break Here NBH US Information Separator One VTS NFI Next Line Line Tabulation Set OSC Operating System Command

Formatting Character Representations

NSP Numeric Space Space Required Space RSP 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.<	

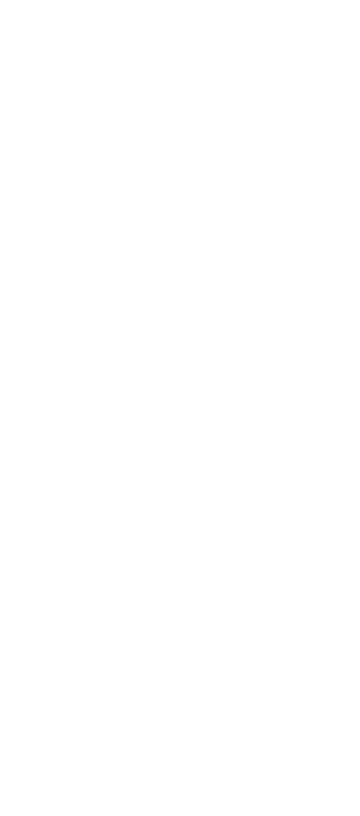
Commonly Used Editing Pattern Characters

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





File Number: S-390-00

