ASMA Instruction Status

Table of Contents

Notices

IBM and z/Architecture are registered trademarks of International Business Machines Corporation.

Introduction

ASMA is current with all defined instruction and extended mnemonics. This document describes implementation plans completed or yet to be completed going forward.

Over time this document will be updated with the implementation status and next set of planned changes.

ASMA as of change 11 supports all machine instructions and extended mnemonics for all mainframe architectures starting with S/360 models through the systems defined by the SA22-7832-13 version of the *IBM® z/Architecture® Principles of Operation* manual released in May, 2022.

Machine instructions are supported by ASMA for instructions defined by versions -11, -12, and -13, as described in the following three sections devoted to each version. All previous PoO manual versions and systems are also supported but not included in this document which is focused on bringing ASMA machine instruction support current with the latest PoO manual.

See the section "Extended Mnemonics" for details on extended mnemonics requiring ASMA enhancements for support.

Because ASMA instructions are defined by the MSL files (see the `asma/msl` directory), this is largely a plan for MSL. The last features implemented in `s390x-inst.msl` when this document was created, are those instructions added by the PoO manual SA22-7832-11, released in September, 2017.

The next version of the PoO manual beyond -11 was -12, released in September, 2019. The latest PoO manual, -13, was released in May, 2022. Adding these instructions makes ASMA current. Most, but not all, new instructions are in the area of new vector instructions. Of most interest by the users of ASMA are support for new instructions that are of more general use. "General use" includes privileged instructions for bare-metal programs. All machine instructions matching this description were implemented and described by "Change 4" of this document's change history.

Addition of the remaining instructions, bringing ASMA current with the latest PoO manual, -13, are described by "Change 7" of this document's change history.

ASMA Instruction Status

This table documents the general status of ASMA instruction development for the -11, -12, and -13 PoO manuals by chapter. The "Type" column identifies whether the instructions defined by the chapter are available to all programs (general) or only programs executing in privileged state (privileged). The three status columns identify the status of the chapter contents within ASMA:

- Implemented – new instructions introduced by the chapter and implemented by ASMA,

- **Partial** – new instructions introduced by the chapter and only partially implemented by ASMA,

- **Implement** – new instructions introduced and ASMA does not yet support the new instructions, and

- No change – no new instructions were introduced by the chapter in the PoO manual version.

Instructions	Type	Chapter	-11 Status	-12 Status	-13 Status
General	general	7	Implemented	Implemented	Implemented
Decimal	general	8	Implemented	No change	No change
FP Overview & Support	general	9	Implemented	No change	No change
Control	privileged	10	Implemented	Implemented	Implemented
I/O	privileged	14	Implemented	No change	No change
Hexadecimal FP	general	18	Implemented	No change	No change
Binary FP	general	19	Implemented	No change	No change
Decimal FP	general	20	Implemented	No change	No change
Vector Overview & Support	general	21	Implemented	Implemented	No change
Vector Integer	general	22	Implemented	Implemented	No change
Vector String	general	23	Implemented	Implemented	No change
Vector FP	general	24	Implemented	Implemented	No change
Vector Decimal	general	25	Implemented	No change	Implemented
Specialized Function Assist	general	26	--	--	Implemented

Report on the SATK `github` repository web site: https://github.com/s390guy/SATK Issues page any instruction unsupported or incorrectly supported by ASMA that is a

- machine instruction introduced **prior to and including the SA22-7832-13 PoO** version, May, 2022, or

- extended mnemonic **prior to and including SA22-7832-13 PoO**, May, 2022.

Based upon the analysis of the remaining machine instructions requiring implementation, bringing ASMA current with the latest PoO version will proceed to completion. All chapters will eventually be shown as "Implemented" or "No change". As reflected in the previous table, all of the required work to bring ASMA machine instructions current with the latest PoO

manual, -13, relates to vector instructions and the nine new instructions in Chapter 26. This work was completed with "Change 7" of this document.

Following implementation of all new instructions, the Machine Specification Language enhancements required for the new extended mnemonics using bits beyond 15, introduced by z/Architecture, will be made. See the "Extended Mnemonics" section for base mnemonics present in Chapter 7 (-13) having defined extended mnemonics. Extended mnemonics introduced in other chapters have now been implemented.

In the tables in the following sections, instructions, MSL formats, and programming notes in **bold** text require implementation. As implementation occurs, the bold text font will be changed to normal text font.

Change History

Change	Date	Description
1	4 Sep 2022	Initial release of this status.
2	5 Sep 2022	-11 PoO fully supported with addition of two instructions that rename instructions released in the -10 PoO, chapter 24.
3	6 Sep 2022	Completed research for -12 and -13 PoO versions of chapters 7, 8, 10, and 14 instruction changes. Ready for implementation.
4	8 Sep 2022	Provided BIC extended mnemonic analysis. Implemented all instruction to current level (-13) for chapters 7, 8, 10, and 14. Added appendix describing issues and enhancements required for support of all extended mnemonics.
5	13 Sep 2022	Completed analysis of **all** instruction defining chapters required to bring ASMA **machine** instruction support current with the latest PoO (-13).
6	14 Sep 2022	Completed implementation of vector instructions missing from original release by PoO -10. All machine instructions prior to and including the -11 PoO are now supported by ASMA. Only some vector instructions and Chapter 26 still require work.
7	17 Sep 2022	Completed implementation of all machine instructions introduced prior to the -13 PoO manual, consisting of machine introductions introduced by the -12 and -13 manuals. These instructions are all vector instructions or part of the new facility assist, chapter 26 of the -13 manual.
8	14 Oct 2022	Added new z/Architecture extended mnemonics for general instructions (Chapter 7) introduced in PoO's -00 through -06, inclusive.
9	17 Oct 2022	Added remaining z/Architecture extended mnemonics for general instructions (Chapter 7) introduced in PoO-'s -07 through -13, inclusive.
10	21 Oct 2022	Completed planning for implementation of extended mnemonics in

ASMA Instruction Status

Change	Date	Description
11	3 Nov 2022	Chapters 21-25 of versions -10 through -13 of the PoO manual. Add changes triggered by github Issue #31 research and implementation including discovered format errors for PoO -02 and -03.

SA22-7832-02 – June, 2003

As reported in the SATK github Issue #31, eleven instructions whose formats used 12-bit displacements were changed to use 20-bit displacements with the introduction of the long-displacement facility.

This section focuses on an analysis of the long-displacement facility and the status as of this writing, 3 November, 2022, within ASMA, in particular looking at the instruction formats.

The review of the ASMA implementation of the long-displacement facility found five instruction formats incorrectly defined within the MSL files.

General Instructions – Chapter 7

Page numbers refer to PoO -02.

Mnemonic	Page #	Prog. Notes	MSL Format	Implemented	Other Notes
AG	7-18	unnumbered	RXYA	yes	ADD (64)
AGF	7-18	unnumbered	RXYA	yes	ADD (64<-32)
AHY	7-18	unnumbered	RXYA	yes	ADD HALFWORD
ALG	7-19	unnumbered	RXYA	yes	ADD LOGICAL (64)
ALGF	7-19	unnumbered	RXYA	yes	ADD LOGICAL (64<-32)
ALY	7-19	unnumbered	RXYA	yes	ADD LOGICAL (32)
AY	7-18	unnumbered	RXYA	yes	ADD (32)
CDSG	7-53	unnumbered	RSYA	yes	COMPARE DOUBLE AND SWAP (64)
CDSY	7-53	unnumbered	RSYA	yes	COMPARE DOUBLE AND SWAP (32)
CG	7-46	unnumbered	RXYA	yes	COMPARE (64)
CGF	7-46	unnumbered	RXYA	yes	COMPARE (64<-32)
CHY	7-55	unnumbered	RXYA	yes	COMPARE HALFWORD
CLG	7-56	unnumbered	RXYA	yes	COMPARE LOGICAL (64)
CLGF	7-56	unnumbered	RXYA	yes	COMPARE LOGICAL (64<-32)
CLIY	7-56	unnumbered	SIY	yes	COMPARE LOGICAL
CLMH	7-57	unnumbered	RSYB	yes	COMPARE LOGICAL CHARACTERS UNDER MASK (HIGH)
CLMY	7-57	unnumbered	RSYB	yes	COMPARE LOGICAL CHARACTERS UNDER MASK (LOW)
CLY	7-56	unnumbered	RXYA	yes	COMPARE LOGICAL (32)
CSG	7-53	unnumbered	RSYA	yes	COMPARE AND SWAP (64)
CSY	7-53	unnumbered	RSYA	yes	COMPARE AND SWAP (32)
CVBG	7-97	unnumbered	RXYA	yes	CONVERT TO BINARY (64)
CVBY	7-97	unnumbered	RXYA	yes	CONVERT TO BINARY (32)
CVDG	7-98	unnumbered	RXYA	yes	CONVERT TO DECIMAL (64)
CVDY	7-98	unnumbered	RXYA	yes	CONVERT TO DECIMAL (32)
CY	7-46	unnumbered	RXYA	yes	COMPARE (32)
ICMH	7-109	unnumbered	RSYB	yes	INSERT CHARACTER UNDER MASK
ICMY	7-109	unnumbered	RSYB	yes	INSERT CHARACTER UNDER MASK
ICY	7-109	unnumbered	RXYA	yes	INSERT CHARACTER
LAMY	7-111	unnumbered	RSYA	yes	LOAD ACCESS MULTIPLE

ASMA Instruction Status

Mnemonic	Page #	Prog. Notes	MSL Format	Implemented	Other Notes
LAY	7-112	unnumbered	RXYA	yes	LOAD ADDRESS
LB	7-114	unnumbered	RXYA	yes	LOAD BYTE (32)
LG	7-111	unnumbered	RXYA	yes	LOAD (64)
LGB	7-114	unnumbered	RXYA	yes	LOAD BYTE (64)
LGF	7-111	unnumbered	RXYA	yes	LOAD (64<-32)
LHY	7-115	unnumbered	RXYA	yes	LOAD HALFWORD (32)
LY	7-111	unnumbered	RXYA	yes	LOAD (32)
MSG	7-139	unnumbered	RXYA	yes	MULTIPLY SINGLE (64)
MSGF	7-139	unnumbered	RXYA	yes	MULTIPLY SINGLE (64<-32)
MSY	7-138	unnumbered	RXYA	yes	MULTIPLY SINGLE (32)
MVIY	7-121	unnumbered	SIY	yes	MOVE
NG	7-20	unnumbered	RXYA	yes	AND (64)
NIY	7-21	unnumbered	SIY	yes	AND
NY	7-20	unnumbered	RXYA	yes	AND (32)
OG	7-106	unnumbered	RXYA	yes	OR (64)
OIY	7-140	unnumbered	SIY	yes	OR
OY	7-140	unnumbered	RXYA	yes	OR (32)
SG	7-174	unnumbered	RXYA	yes	SUBTRACT (64)
SGF	7-174	unnumbered	RXYA	yes	SUBTRACT (64<-32)
SHY	7-174	unnumbered	RXYA	yes	SUBTRACT HALFWORD (32)
SLG	7-175	unnumbered	RXYA	yes	SUBTRACT LOGICAL (64)
SLGF	7-175	unnumbered	RXYA	yes	SUBTRACT LOGICAL (64<-32)
SLY	7-175	unnumbered	RXYA	yes	SUBTRACT LOGICAL (32)
STCMH	7-168	unnumbered	RSYB	yes	STORE CHARACTERS UNDER MASK
STCMY	7-168	unnumbered	RSYB	yes	STORE CHARACTERS UNDER MASK
STCY	7-168	unnumbered	RXYA	yes	STORE CHARACTER
STG	7-167	unnumbered	RXYA	yes	STORE (64)
STHY	7-172	unnumbered	RXYA	yes	STORE HALFWORD
STMY	7-172	unnumbered	RSYA	yes	STORE MULTIPLE
STY	7-167	unnumbered	RXYA	yes	STORE (32)
SY	7-174	unnumbered	RXYA	yes	SUBTRACT (32)
TMY	7-178	unnumbered	SIY	yes	TEST UNDER MASK
XG	7-107	unnumbered	RXYA	yes	EXCLUSIVE OR (64)
XIY	7-107	unnumbered	SIY	yes	EXCLUSIVE OR
XY	7-106	unnumbered	RXYA	yes	EXCLUSIVE OR (32)

Programming Notes

Note #	Description
unnumbered	Long-Displacement Facility

SA22-7832-03 – May, 2004

As reported in SATK github Issue #31, the instruction format changed with the PoO -03 for fifteen instructions without any notice. The MSL format requires changing from RSA2 to RSYA.

Twelve of the instructions are general instructions defined in Chapter 7. The remaining three are defined in Chapter 10, Control Instructions.

Page numbers in this section refer to PoO -03.

General Instructions – Chapter 7

Mnemonic	Page #	Prog. Notes	MSL Format	Implemented	Other Notes
CLCLU	7-64		RSYA	yes	COMPARE LOGICAL LONG UNICODE
LMG	7-129		RSYA	yes	LOAD MULTIPLE (64)
LMH	7-130		RSYA	yes	LOAD MULTIPLE HIGH
MVCLU	7-142		RSYA	yes	MOVE LONG UNICODE
RLL	7-171		RSYA	yes	ROTATE LEFT SINGLE LOGICAL (32)
RLLG	7-171		RSYA	yes	ROTATE LEFT SINGLE LOGICAL (64)
SLAG	7-177		RSYA	yes	SHIFT LEFT SINGLE (64)
SLLG	7-178		RSYA	yes	SHIFT LEFT SINGLE LOGICAL (64)
SRAG	7-179		RSYA	yes	SHIFT RIGHT SINGLE (64)
SRLG	7-180		RSYA	yes	SHIFT RIGHT SINGLE LOGICAL (64)
STMG	7-185		RSYA	yes	STORE MULTIPLE (64)
STMH	7-186		RSYA	yes	STORE MULTIPLE HIGH

Control Instructions – Chapter 10

Mnemonic	Page #	Prog. Notes	MSL Format	Implemented	Other Notes
LCTLG	10-45		RSYA	yes	LOAD CONTROL (64)
STCTG	10-107		RSYA	yes	STORE CONTROL (64)
TRACG	10-132		RSYA	yes	TRACE (64)

SA22-7832-11 – September, 2017

No new decimal instructions, Chapter 8, nor new I/O Instructions, Chapter 14, were added. Instructions that operate upon decimal floating point data, Chapter 20, and decimal data acted upon by vector instructions, Chapter 25 *were* added. Those additions have been implemented in ASMA.

Upon inspection of the actual MSL files, nearly all new instructions **WERE** already added to the MSL files for the -11 version of the PoO manual except for the mnemonic change for two instructions in Chapter 24.

During the analysis of floating point and vector instructions, ten vector instructions were found to be only partially implemented within the MSL. These ten were actually introduced in the -10 version of the PoO. By adding these ten instructions' mnemonics to the MSL `s390x-vector iset` statement (the statement defining the VECTOR FACILITY), the instructions were made visible to the assembler for z/Architecture target assemblies and those targets including all instructions.

By incorporating the mnemonic name changes, ASMA is current with the -11 version of the PoO manual for all -11 and previous machine instructions.

At this point ASMA is now requiring enhancements for -12, released in September, 2019 and -13, released in May, 2022. ASMA is now three years behind.

General Instructions – Chapter 7

Mnemonic	Page #	Prog. Notes	MSL Format	Implemented	Other Notes
AGH	7-28	31	RXYA	yes	
BIC	7-39	31	RXYB	yes	See BIC EM table below
KMA	7-78	33	RRFB3	yes	
CLT	7-155	26	RSYB	yes	
CLGT	7-155	26	RSYB	yes	
LGG	7-274	32	RXYA	yes	
LGSC	7-275	32	RXYA	yes	
LLGFSG	7-274	32	RXYA	yes	
MG	7-304	31	RXYA	yes	
MGRK	7-304	31	RRFA1	yes	
MGH	7-305	31	RXYA	yes	
MSC	7-307	31	RXYA	yes	
MSRKC	7-307	31	RRFA1	yes	
MSGC	7-307	31	RXYA	yes	
MSGRKC	7-307	31	RRFA1	yes	
PRNO	7-346	29	RRE	yes	
RISBGN	7-363	26	RIEF	yes	
STGSC	7-383	32	RXYA	yes	
SGH	7-388	31	RXYA	yes	

BIC Extended Mnemonics (EM)

The `BIC` instruction utilizes an extended operation code in bits 40-47. Because current extended mnemonic support utilizes the extended operation in bits 8-11, it is not possible to define the `BIC` extended mnemonics by overloading the extended operation field with the fixed mask. The new support for extended mnemonics is required for the `BIC` instruction.

Mnemonic	Page #	Prog. Notes	MSL Format	Implemented	Notes
BIO	7-39		ERXYB	yes	M1 = 1
BIP	7-39		ERXYB	yes	M1 = 2
BIH	7-39		ERXYB	yes	M1 = 2

Mnemonic	Page #	Prog. Notes	MSL Format	Implemented		Notes
BIM	7-39		ERXYB	yes	M1 = 4	
BIL	7-39		ERXYB	yes	M1 = 4	
BINZ	7-39		ERXYB	yes	M1 = 7	
BINE	7-39		ERXYB	yes	M1 = 7	
BIZ	7-39		ERXYB	yes	M1 = 8	
BIE	7-39		ERXYB	yes	M1 = 8	
BINM	7-39		ERXYB	yes	M1 = 11 or B	
BINL	7-39		ERXYB	yes	M1 = 11 or B	
BINP	7-39		ERXYB	yes	M1 = 13 or D	
BINH	7-39		ERXYB	yes	M1 = 13 or D	
BINO	7-39		ERXYB	yes	M1 = 14 or E	
BI	7-39		ERXYB	yes	M1 = 15 or F	

Change 9 implements BIC's extended mnemonics using the enhancements to MSL.

Programming Notes

Note #	Description
26	Miscellaneous-Instruction Extensions Facility 1
29	Message-Security-Assist Extension 5
31	Miscellaneous-Instruction Extensions Facility 2
32	Guarded Storage Facility
33	Message-Security-Assist Extension 8

Control Instructions – Chapter 10

Mnemonic	Page #	Prog. Notes	MSL Format	Implemented	Notes
IRBM	10-30	11	RRE	yes	
TPEI	10-169	12	RRE	yes	

Programming Notes

Note #	Description
11	Insert-Reference-Bits-Multiple Facility
12	Test Pending External Interruption Facility

Vector Overview and Support Instructions – Chapter 21

Mnemonic	Page #	Prog. Notes	MSL Format	Implemented	Notes
VLEF	21-7	A	VRX	yes	
VLEG	21-7	A	VRX	yes	
VLBB	21-10	A	VRX	yes	
VPKLS	21-14	A	VRRB	yes	
VSEL	21-17	A	VRRE1	yes	

Programming Notes

Note #	Description
A	Vector Facility introduced in PoO version -10

Vector Integer Instruction – Chapter 22

Mnemonic	Page #	Prog. Notes	MSL Format	Implemented	Notes
VNC	22-5	A	VRRC2	yes	
VCH	22-8	A	VRRB	yes	
VCHL	22-9	A	VRRB	yes	

Programming Notes

Note #	Description
A	Vector Facility introduced in PoO version -10

Vector Floating Point Instructions – Chapter 24

Mnemonic	Page #	Prog. Notes	MSL Format	Implemented	Notes
VFLL	24-25		VRRA4	yes	Changed Mnemonic from VLDE
VFLR	24-26		VRRA	yes	Changed Mnemonic from VLED

SA22-7832-12 – September, 2019

General Instructions – Chapter 7

Mnemonic	Page #	Prog. Notes	MSL Format	Implemented	Notes
NCRK	7-34	34	RRFA1	yes	
NCGRK	7-34	34	RRFA1	yes	
MVCRL	7-300	34	SSE2	yes	
NNRK	7-308	34	RRFA1	yes	
NNGRK	7-308	34	RRFA1	yes	
NORK	7-311	34	RRFA1	yes	
NOGRK	7-311	34	RRFA1	yes	
NXRK	7-311	34	RRFA1	yes	
NXGRK	7-311	34	RRFA1	yes	
OCRK	7-314	34	RRFA1	yes	
OCGRK	7-314	34	RRFA1	yes	
POPCNT	7-365	34	RRFC	yes	
SELR	7-376	34	RRFA2	yes	
SELGR	7-376	34	RRFA2	yes	
SELFHR	7-376	34	RRFA2	yes	

Programming Notes

Note #	Description
34	Miscellaneous-Instruction-Extension Facility 3

Vector Overview and Support Instructions – Chapter 21

Mnemonic	Page #	Prog. Notes	MSL Format	Implemented	Notes
VLEBRH	21-7	unnumbered	VRX	yes	
VLEBRF	21-7	unnumbered	VRX	yes	
VLEBRG	21-7	unnumbered	VRX	yes	
VLBRREP	21-8	unnumbered	VRX	yes	
VLLEBRZ	21-8	unnumbered	VRX	yes	
VLBR	21-9	unnumbered	VRX	yes	
VLER	21-10	unnumbered	VRX	yes	
VSTEBRH	21-22	unnumbered	VRX	yes	
VSTEBRF	21-22	unnumbered	VRX	yes	
VSTEBRG	21-22	unnumbered	VRX	yes	
VSTBR	21-22	unnumbered	VRX	yes	
VSTER	21-24	unnumbered	VRX	yes	

Programming Notes

Note #	Description
unnumbered	Vector-Enhancements Facility 2

Vector Integer Instructions – Chapter 22

Mnemonic	Page #	Prog. Notes	MSL Format	Implemented	Notes
VSLD	22-25	unnumbered	VRID1	yes	
VSRD	22-26	unnumbered	VRID1	yes	

Programming Notes

Note #	Description
unnumbered	Vector-Enhancements Facility 2

Vector String Instructions – Chapter 23

Mnemonic	Page #	Prog. Notes	MSL Format	Implemented	Notes
VSTRS	23-8	2	VRRD	yes	

Programming Notes

Note #	Description
2	Vector-Enhancements Facility 2

Vector Floating Point Instructions – Chapter 24

Mnemonic	Page #	Prog. Notes	MSL Format	Implemented	Notes
VCFPS	24-15	unnumbered	VRRA	yes	Rename VCDG
VCFPL	24-17	unnumbered	VRRA	yes	Rename VCDLG
VCSFP	24-18	unnumbered	VRRA	yes	Rename VCGD
VCLFP	24-20	unnumbered	VRRA	yes	Rename VCLDG

Programming Notes

Note #	Description
unnumbered	Vector-Enhancements Facility 2

SA22-7832-13 – May, 2022

Control Instructions – Chapter 10

Mnemonic	Page #	Prog. Notes	MSL Format	Implemented	Notes
LBEAR	10-51	13	S0 (zero)	yes	
LPSWEY	10-57	13	SIY0 (zero)	yes	
QPACI	10-123	15	S0 (zero)	yes	
RDP	10-124	14	RRFB2	yes	
STBEAR	10-145	13	S0 (zero)	yes	

Programming Notes

Note #	Description
13	BEAR-Enhancement Facility
14	Reset DAT-Protection Facility
15	Processor-Activity-Instrumentation Facility

Vector Decimal Instructions – Chapter 25

Mnemonic	Page #	Prog. Notes	MSL Format	Implemented	Notes
VSCSHP	25-4	2	VRRC	implemented	
VSCHP	25-5	2	VRRB	implemented	
VCSPH	25-11	2	VRRJ	implemented	
VCLZDP	25-11	2	VRRK	implemented	
VPKZR	25-18	2	VRIF	implemented	
VSRPR	25-26	2	VRIF	implemented	
VUPKZH	25-30	2	VRRK	implemented	
VUPKZL	25-31	2	VRRK	implemented	

Programming Notes

Note #	Description
2	Vector-Packed-Decimal-Enhancement Facility 2

Specialized-Function-Assist Instructions – Chapter 26

Mnemonic	Page #	Prog. Notes	MSL Format	Implemented	Notes
KDSA	26-2	1	RRE	implemented	
DFLTCC	26-17	2	RRFA1	implemented	
NNPA	26-61	4	RRE1	implemented	
SORTL	26-96	3	RRE	implemented	
VCLFNH	26-121	4	VRRA4	implemented	
VCLFNL	26-122	4	VRRA4	implemented	
VCRNF	26-123	4	VRRA4	implemented	

ASMA Instruction Status

Mnemonic	Page #	Prog. Notes	MSL Format	Implemented	Notes
VCFN	26-123	4	VRRA4	implemented	
VCNF	26-124	4	VRRA4	implemented	

Programming Notes

Note #	Description
1	Message-Security-Assist Extension 9
2	DEFLATE-Conversion Facility
3	Enhanced-Sort Facility
4	Neural-Network-Processing-Assist Facility

Extended Mnemonics

This section documents the current status of extended mnemonics in ASMA. Support for the base machine instruction mnemonics are described in the preceding sections.

Refer to SA22-7832-13, Appendix J, for details. The following table only reflects the base mnemonic for which extended mnemonics are defined. Page numbers are for the -13 version.

All extended mnemonics up to and including ESA/390 either separate from z/Architecture or ESA/390 on a z/Architecture system *are* supported by ASMA. While ESA/390 continued to be available on z/Architecture until the release of the CZAM facility, ESA/390 ceased being enhanced on z/Architecture beyond PoO -02.

All extended mnemonics defined for general instructions, Chapter 7, *are* supported.

The following table is similar to the one provided for machine instructions. The individual status columns reflect the status for extended mnemonics by PoO chapter.   Hyphens indicate the Chapter was not supplied in the PoO version. Status reflects the following:

- none – no extended mnemonics defined by this PoO version,

- **no** – planned for implementation, or

- yes – implemented.

The goal is for the entry to be either "none" or "yes".

Instructions	Type	Chapter	-00 - -04	-05 - -09	-10	-11	-12	-13
General	general	7	yes	yes	yes	yes	yes	yes
Decimal	general	8	none	none	none	none	none	none
FP Overview & Support	general	9	none	none	none	none	none	none
Control	privileged	10	none	none	none	none	none	none
I/O	privileged	14	none	none	none	none	none	none
Hexadecimal FP	general	18	none	none	none	none	none	none
Binary FP	general	19	none	none	none	none	none	none
Decimal FP	general	20	--	none	none	none	none	none
Vector Overview & Support	general	21	--	--	yes	none	yes	none
Vector Integer	general	22	--	--	yes	yes	none	none
Vector String	general	23	--	--	yes	none	yes	none
Vector FP	general	24	--	--	yes	yes	yes	none
Vector Decimal	general	25	--	--	none	none	none	none
Specialized Function Assist	general	26	--	--	--	--	--	none

General Instructions – Chapter 7

This table describes the base mnemonic for which extend mnemonics are defined.

ASMA Instruction Status

"Page #" is the page on which the base machine instruction is described in the -13 PoO version.

"Base ASMA Target" is the ASMA target instruction set architecture in which the mnemonic was introduced. The number preceded by a dash "-" refers to the z/Architecture PoO version in which the base mnemonic was introduced. The dash applies only to SA22-7832 manual versions.

"EM MSL Format" is the MSL format used when the extended mnemonic is defined.

"Opcode" is the hexadecimal machine instruction operation code of the base mnemonic and is the operation code used by all extended mnemonics of that machine instruction.

Base Mnemonic	Page #	Base ASMA Target	EM MSL Format	Opcode	Notes
BCR	7-40	s360	ER	07	
BC	7-40	s360	ERXB	47	
BRAS	7-45	e390	RIB	A75	JAS mnemonic
BRC	7-46	e390	ERIC	A74	
BRCT	7-47	e390	RIB	A76	JCT mnemonic
BRXH	7-48	e390	RSI	84	JXH mnemonic
BRXLE	7-48	e390	RSI	85	JXLE mnemonic
BRASL	7-45	s390	RILB	C05	JASL mnemonic
BRCL	7-46	s390	ERILC	C04	
BRCTG	7-47	S390x -00	RIB	A77	JCTG mnemonic
BRCTH	7-47	S390x -00	RILB	CC6	JCTH mnemonic
BRXHG	7-48	S390x -00	RIEE	EC44	JXHG mnemonic
BRXLG	7-48	S390x -00	RIEE	EC45	JXLEG mnemonic
IILF	7-266	S390x -04	RILD	C09	LFI listed in Appendix J, but not on 7-266. Renames IILF. See also Note 1.
LLILF	7-284	S390x -04	RILD	C0F	LLGFI Listed in Appendix J, but not on 7-284. Renames LLILF. See also Note 1.
CRB	7-137	S390x -06	ERRS	ECF6	Compare branch
CGRB	7-137	S390x -06	ERRS	ECE4	Compare branch
CRJ	7-137	S390x -06	ERIEB	EC76	Compare branch
CGRJ	7-137	S390x -06	ERIEB	EC64	Compare branch
CRT	7-150	S390x -06	ERRFC	B972	Compare branch; listed in Appendix J, but no extended mnemonics on 7-150
CGRT	7-150	S390x -06	ERRFC	B960	Compare branch; listed in Appendix J, but no extended mnemonics on 7-150
CIB	7-137	S390x -06	ERIS	ECFE	Compare branch
CGIB	7-137	S390x -06	ERIS	ECFC	Compare branch
CIJ	7-137	S390x -06	ERIEC	EC7E	Compare branch
CGIJ	7-137	S390x -06	ERIEC	EC7C	Compare branch
CIT	7-150	S390x -06	ERIEA	EC72	Compare branch; listed in Appendix J, but no extended mnemonics on 7-150
CGIT	7-150	S390x -06	ERIEA	EC70	Compare branch; listed in Appendix J, but no extended mnemonics on 7-150
CLRB	7-155	S390x -06	ERRS	ECF7	Compare branch; listed in Appendix J, but no extended mnemonics on 7-155

ASMA Instruction Status

Base Mnemonic	Page #	Base ASMA Target	EM MSL Format	Opcode	Notes
CLGRB	7-155	S390x -06	ERRS	ECE5	Compare branch; listed in Appendix J, but no extended mnemonics on 7-155
CLRJ	7-155	S390x -06	ERIEB	EC77	Compare branch; listed in Appendix J, but no extended mnemonics on 7-155
CLGRJ	7-155	S390x -06	ERIEB	EC65	Compare branch; listed in Appendix J, but no extended mnemonics on 7-155
CLRT	7-156	S390x -06	ERRFC	B973	Compare branch; listed in Appendix J, but no extended mnemonics on 7-156
CLGRT	7-156	S390x -06	ERRFC	B961	Compare branch; listed in Appendix J, but no extended mnemonics on 7-156
CLIB	7-155	S390x -06	ERIS	ECFF	Compare branch; listed in Appendix J, but no extended mnemonics on 7-155
CLGIB	7-155	S390x -06	ERIS	ECFD	Compare branch; listed in Appendix J, but no extended mnemonics on 7-155
CLIJ	7-155	S390x -06	ERIEC	EC7F	Compare branch; listed in Appendix J, but no extended mnemonics on 7-155
CLGIJ	7-155	S390x -06	ERIEC	EC7D	Compare branch; listed in Appendix J, but no extended mnemonics on 7-155
CLFIT	7-157	S390x -06	ERIEA	EC73	Compare branch; listed in Appendix J, but no extended mnemonics on 7-157
CLGIT	7-157	S390x -06	ERIEA	EC71	Compare branch; listed in Appendix J, but no extended mnemonics on 7-157
RNSBG	7-372	S390x -06	ERIEF	EC54	
RNSBGT	7-373	S390x -06	ERIEF-A	EC54	Treat this base as an extended mnemonic
RXSGB	7-372	S390x -06	ERIEF	EC57	
RXSGBT	7-372	S390x -06	ERIEF-A	EC57	
RISBG	7-374	S390x -06	ERIEF-A	EC55	
RISBGN	7-374	S390x -06	ERIEF-A	EC59	
ROSBG	7-372	S390x -06	ERIEF	EC56	
ROSBGT	7-372	S390x -06	ERIEF-A	EC56	
BRCTH	7-47	S390x -08	RILB	CC6	
LOCR	7-287	S390x -08	ERRFC	B9F2	
LOCGR	7-287	S390x -08	ERRFC	B9E2	
LOC	7-287	S390x -08	ERSYB1	EBF2	
LOCG	7-287	S390x -08	ERSYB1	EBE2	
CLT	7-156	S390x -08	ERSYB1	EB23	
RISBHG	7-375	S390x -08	ERIEF	EC5D	
RISBLG	7-375	S390x -08	ERIEF	EC51	
STOC	7-397	S390x -08	ERSYB1	EBF3	
STOCG	7-397	S390x -08	ERSYB1	EBE3	
CLGT	7-156	S390x -09	ERSYB1	EB2B	
LOCHHI	7-280	S390x -10	ERIEG	EC4E	
LOCHI	7-280	S390x -10	ERIEG	EC42	
LOCGHI	7-280	S390x -10	ERIEG	EC46	
LOCFHR	7-287	S390x -10	ERRFC	B9E0	
LOCFH	7-287	S390x -10	ERSYB1	EBE0	
STOCFH	7-397	S390x -10	ERSYB1	EBE1	
BIC	7-39	S390x -11	ERXYB	E347	
NORK	7-314	S390x -12	ERRFA1	B976	

ASMA Instruction Status

Base Mnemonic	Page #	Base ASMA Target	EM MSL Format	Opcode	Notes
NOGRK	7-314	S390x -12	ERRFA1	B966	
SELR	7-380	S390x -12	ERRFA2	B9F0	
SELGR	7-380	S390x -12	ERRFA2	B9E3	
SELFHR	7-380	S390x -12	ERRFA2	B9C0	

Note 1. These two extended mnemonics rename their respective base mnemonics. The two extended mnemonics, LFI and LLGFI, appear to have been introduced with the -13 version of the PoO. The entries for the base mnemonics, IILF and LLILF, have change bars next to them in the -13 PoO J appendix. The base mnemonics for the machine instructions, IILF and LLILF, were introduced with the -04 version of the PoO in which its Appendix B, List of Instructions, has change bars next to these instructions. It can only be speculated why there are 17 years between the introduction of the machine instructions with the -04 PoO manual (September, 2005) and the introduction of the extended mnemonics with the -13 PoO manual, (May, 2022). The extended mnemonic implementation plan documented here is based upon when the base instruction was introduced. So these two extended mnemonics introduced in 2022 are implemented as if they were introduced by the -04 PoO version.

Floating Point Instructions – Chapters 9, 18-20

No extended mnemonics are defined for floating point instructions.

Vector Instructions – Chapters 21-25

Vector Overview and Support Instructions – Chapter 21

Base Mnemonic	Page #	Base ASMA Target	EM MSL Format	Opcode	Notes
VGBM	21-5	s390x -10	EVRIA	E744	
VGM	21-6	s390x -10	EVRIB	E746	
VLGV	21-11	s390x -10	EVRSC	E7C7	
VLLEZ	21-12	s390x -10	EVRX	E704	
VLREP	21-7	s390x -10	EVRX	E705	
VLVG	21-14	s390x -10	EVRSB	E722	
VMRH	21-15	s390x -10	EVRRC1	E761	
VMRL	21-16	s390x -10	EVRRC1	E760	
VPK	21-16	s390x -10	EVRRC1	E794	
VPKLS	21-18	s390x -10	EVRRB	E795	
VPKS	21-17	s390x -10	EVRRB	E797	
VREP	21-19	s390x -10	EVRIC	E74D	
VREPI	21-20	s390x -10	EVRIA1	E745	
VSEG	21-21	s390x -10	EVRRA1	E75F	
VUPH	21-26	s390x -10	EVRRA1	E7D7	
VUPL	21-27	s390x -10	EVRRA1	E7D6	
VUPLH	21-26	s390x -10	EVRRA1	E7D5	

Base Mnemonic	Page #	Base ASMA Target	EM MSL Format	Opcode	Notes
VUPLL	21-27	s390x -10	EVRRA1	E7D4	
VLBR	21-9	s390x -12	EVRX	E606	
VLBRREP	21-8	s390x -12	EVRX	E605	
VLER	21-10	s390x -12	EVRX	E607	
VLLEBRZ	21-8	s390x -12	EVRX	E604	
VSTBR	21-22	s390x -12	EVRX	E60A	
VSTER	21-24	s390x -12	EVRX	E60F	
VSTEBRF	21-22	s390x -12	EVRX	E609	
VSTEBRG	21-22	s390x -12	EVRX	E60A	

Vector Integer Instructions – Chapter 22

Column headings are the same as used by the "General Instructions – Chapter 7" section.

Base Mnemonic	Page #	Base ASMA Target	EM MSL Format	Opcode	Notes
VA	22-3	s390x -10	EVRRC1	E7F3	
VACC	22-4	s390x -10	EVRRC1	E7F1	
VAC	22-4	s390x -10	EVRRD1	E7BB	
VACCC	22-5	s390x -10	EVRRD1	E7B9	
VAVG	22-6	s390x -10	EVRRC1	E7F2	
VAVGL	22-6	s390x -10	EVRRC1	E7F0	
VCEQ	22-7	s390x -10	EVRRB	E7F8	
VCH	22-8	s390x -10	EVRRB	E7FB	
VCHL	22-9	s390x -10	EVRRB	E7F9	
VCLZ	22-10	s390x -10	EVRRA1	E753	
VCTZ	22-10	s390x -10	EVRRA1	E752	
VEC	22-7	s390x -10	EVRRA1	E7DB	
VECL	22-7	s390x -10	EVRRA1	E7D9	
VERIM	22-22	s390x -10	EVRID	E772	
VERLLV	22-21	s390x -10	EVRRC1	E773	
VERLL	22-21	s390x -10	EVRSA	E733	
VESLV	22-23	s390x -10	EVRRC1	E770	
VESL	22-23	s390x -10	EVRSA	E730	
VESRAV	22-23	s390x -10	EVRRC1	E77A	
VESRA	22-23	s390x -10	EVRSA	E73A	
VESRLV	22-24	s390x -10	EVRRC1	E778	
VESRL	22-24	s390x -10	EVRSA	E738	
VGFM	22-11	s390x -10	EVRRC1	E7B4	
VGFMA	22-12	s390x -10	EVRRD1	E7BC	
VLC	22-12	s390x -10	EVRRA1	E7DE	
VLP	22-12	s390x -10	EVRRA1	E7DF	
VMAE	22-15	s390x -10	EVRRD1	E7AE	
VMAH	22-15	s390x -10	EVRRD1	E7AB	
VMAL	22-14	s390x -10	EVRRD1	E7AA	
VMALE	22-15	s390x -10	EVRRD1	E7AC	
VMALH	22-15	s390x -10	EVRRD1	E7A9	
VMALO	22-16	s390x -10	EVRRD1	E7AD	
VMAO	22-16	s390x -10	EVRRD1	E7AF	
VML	22-17	s390x -10	EVRRC1	E7A2	

ASMA Instruction Status

Base Mnemonic	Page #	Base ASMA Target	EM MSL Format	Opcode	Notes
VMLE	22-18	s390x -10	EVRRC1	E7A4	
VMLH	22-17	s390x -10	EVRRC1	E7A1	
VMLO	22-18	s390x -10	EVRRC1	E7A5	
VME	22-18	s390x -10	EVRRC1	E7A6	
VMH	22-16	s390x -10	EVRRC1	E7A3	
VMN	22-13	s390x -10	EVRRC1	E7FE	
VMNL	22-14	s390x -10	EVRRC1	E7FC	
VMO	22-18	s390x -10	EVRRC1	E7A7	
VMX	22-13	s390x -10	EVRRC1	E7FF	
VMXL	22-13	s390x -10	EVRRC1	E7FD	
VNO	22-20	s390x -10	EVRRC2	E76B	
VPOPCT	22-21	s390x -10	EVRRA1	E750	
VS	22-27	s390x -10	EVRRC1	E7F7	
VSBI	22-28	s390x -10	EVRRD1	E7BF	
VSBCBI	22-29	s390x -10	EVRRD1	E7BD	
VSCBI	22-28	s390x -10	EVRRC1	E7F5	
VSUM	22-30	s390x -10	EVRRC1	E764	
VSUMG	22-29	s390x -10	EVRRC1	E765	
VSUMQ	22-30	s390x -10	EVRRC1	E767	
VMSL	22-19	s390x -11	EVRRD	E7B8	

Vector String Instructions – Chapter 23

Column headings are the same as used by the "General Instructions – Chapter 7" section.

Base Mnemonic	Page #	Base ASMA Target	EM MSL Format	Opcode	Notes
VFAE	23-2	s390x -10	EVRRB2	E782	
VFEE	23-3	s390x -10	EVRRB2 EVRRB2A	E780	
VFENE	23-4	s390x -10	EVRRB2 EVRRB2A	E781	
VSTRC	23-6	s390x -10	EVRRD EVRRD2	E78A	
VSTRS	23-8	s390x -12	EVRRD2 EVRRD3	E78B	

Vector Floating-Point Instructions – Chapter 24

Column headings are the same as used by the "General Instructions – Chapter 7" section.

Base Mnemonic	Page #	Base ASMA Target	EM MSL Format	Opcode	Notes
VFA	24-4	s390x -10	EVRRC3	E7E3	
VFCE	24-9	s390x -10	EVRRC4	E7E8	
VFCH	24-11	s390x -10	EVRRC4	E7EB	
VFCHE	24-13	s390x -10	EVRRC4	E7EA	
VFD	24-22	s390x -10	EVRRC3	E7E5	
VFI	24-24	s390x -10	EVRRA	E7C7	

Base Mnemonic	Page #	Base ASMA Target	EM MSL Format	Opcode	Notes
VFM	24-40	s390x -10	EVRRC3	E7E7	
VFMA	24-42	s390x -10	EVRRE	E78F	
VFMS	24-42	s390x -10	EVRRE	E78E	
VFPSO	24-44	s390x -10	EVRRA-1 EVRRA-2	E7CC	
VFS	24-46	s390x -10	EVRRC3	E7E2	
VFSQ	24-45	s390x -10	EVRRA4	E7CE	
VFTCI	24-47	s390x -10	EVRIE	E74A	
WFC	24-7	s390x -10	EVRRA4	E7CB	
WFK	24-8	s390x -10	EVRRA4	E7CA	
VFLL	24-26	s390x -11	EVRRA4-1 EVRRA4	E7C4	
VFLR	24-27	s390x -11	EVRRA-4 EVRRA-3	E7C5	
VFMAX	24-28	s390x -11	EVRRC5	E7EF	
VFMIN	24-34	s390x -11	EVRRC5	E7EE	
VFNMA	24-42	s390x -11	EVRRE	E79F	
VFNMS	24-42	s390x -11	EVRRE	E79E	
VCFPL	24-17	s390x -12	EVRRA-3	E7C1	
VCFPS	24-15	s390x -12	EVRRA-3	E7C3	
VCLFP	24-20	s390x -12	EVRRA-3	E7C0	
VCSFP	24-18	s390x -12	EVRRA-3	E7C2	

Vector Decimal Instructions – Chapter 25

No extended mnemonics are defined for vector decimal instructions.

Specialized-Function-Assist Instructions – Chapter 26

No extended mnemonics are defined for special-function-assist instructions.

Appendix A – Extended Mnemonic Limitation

This appendix explains the current (2022) situation with regards to extended mnemonics and in general terms what enhances are needed within MSL and where those enhancements affect the rest of ASMA.

Extended mnemonics when ASMA was under active development was limited to use of the mask field (bits 8-11) in various branch-type instructions. At the same time, there were a few instructions that utilized some or all of bits 8-15 for an extended operation field. Extended operation fields are used for operation codes in excess of eight bits. This influenced the design of the Machine Specification Language (MSL). MSL was from the start created to support an extended operation field, referred to as the `XOP` field in MSL formats.

Lacking multi-year imagination, the implementation decision was made to overload the XOP field as the implied mask for extended mnemonics. XOP at the time was a good decision. The contents of the XOP field is part of the instruction definition, not the format. So an extended mnemonic, while requiring a specific format for the source fields in the assembly, can be shared with all instructions using bits 8-11 for the mask.

In this paradigm, an instruction can define bits in 8-11, or 8-15 as part of an extended mnemonic (the `BC` mask field) or an extended operation code (SAMx instructions), but not both.

Fast forward a decade and a half and ASMA is faced with a different situation. For one, an instruction may have an extended operation code field in bits other than 8-15. As it turns out MSL from the beginning supported that structure. However, some instructions now have extended mnemonics that use bits 8-11 for a mask **and** an extended operation code. The BIC instruction is a case in point. The overloading of the function of XOP precludes definition of extended mnemonics for this instruction.

Additionally, for a number of complex instructions, fields outside of bits 8-11 are used for the extended mnemonic. MSL has no way to describe what is to be placed in those extended mnemonic fields. It is this enhancement that is required of the MSL for these new extended mnemonics.

This enhancement touches three main portions of ASMA:

- MSL itself (`doc/asma/MSL.odt` and `doc/asma/MSL.pdf`), the

- MSL database built by ASMA at program start (`asma/msldb.py`), and the

- instruction builder (`asma/insnbldr.py`) to fill the extended mnemonic fields for the instruction.

ASMA Instruction Status

The database creation is really conversion of the MSL text files into Python constructs. `msldb.py` is essentially the syntax analyzer of the MSL. The Python constructs are passed to the instruction builder to fill in the values of the instruction fields. It identifies the source of for the machine instruction. In the context of a language, `insnbldr.py` performs semantic processing of MSL.

Two MSL statements participate in instruction definition: the

- `inst` statement, and the

- `format` statement.

`inst` defines which format is used by the instruction and machine instruction constant content, such as the operation code or extended operation code.

`format` maps the instruction's source operands to the destination field within the machine instruction into which the operand value is inserted.

A new extended mnemonic format is required for those instructions that have extended mnemonics. This is because the instruction's source syntax changes as a result of the extended mnemonic.

However, the values used by each instruction in the extended mnemonic is instruction specific and should appear within the individual extended mnemonic's `inst` statement, as is the case for the XOP field content currently supported by MSL.

Additionally, certain extended mnemonics effect specific bits in certain ways, and some bits should be ignored. A new parameter for support of these individual cases requires addition to MSL. And example is the ROTATE THEN… instructions. Others may appear. In the general case a special routine within ASMA must be triggered by the MSL to handle these cases.

The limitation described above is eliminated by implementation of two new parameters to the `inst` statement:

- `fixed` – used to insert a specific value into an instruction's field, and

- `filter` – used to manipulate the assembled operand when placed in the instruction.

With these two enhancements, the missing extended mnemonics can be defined within the MSL files for them. Enhancements to other components involved in instruction creation in particular, `insnbldr.py`, allow these instructions to be be assembled.

These enhancements also open up the possibility of re-implementing extended mnemonics using the XOP overload approach using the new parameters.

It should be noted that a number of MSL parameters as they appear in the text file and are placed within the database are *tightly* coupled with operand syntax parsing and the instruction builder. Arbitrary changes to source operand names or machine instruction fields should not be made.

Implementation and Testing Considerations

The above is written as though the MSL language can be enhanced separate from instruction definition and implementation by ASMA. The realities of implementation and testing is that, while MSL changes must precede instruction definition, testing can not occur without all three components being enhanced. Yes, `mslrpt.py` allows testing of the MSL language. However, instructions that utilize the MSL enhancements and that can then be successfully assembled (`insnbldr.py`) is the real test. Those tests may result in changes to the MSL language or adjustments to `msldb.py` or `insnbldr.py` or other modules to achieve successful assembly.