OpenPOWER ISA

H. Peter Hofstee, Ph.D.
IBM & TU Delft

First things first …

If you are interested in the OpenPOWER architecture …
first thing to do is join the OpenPOWER foundation!

Free for associate/academic members

https://openpowerfoundation.org/

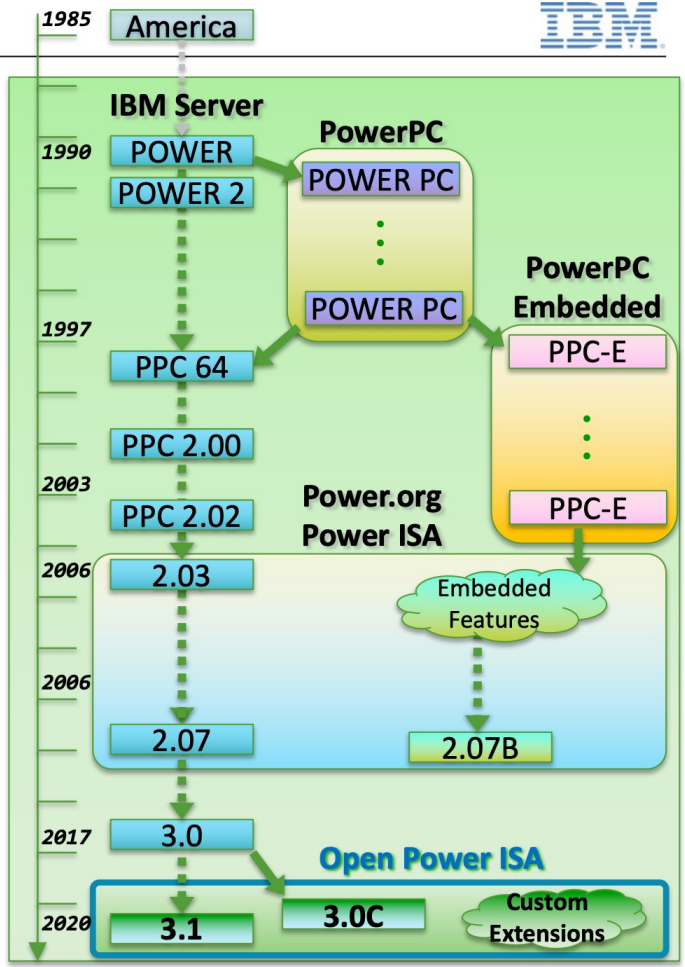
Early History

- Telephone switch (from 1974)
- Project ‘801’ (from 1975)
- ‘Cheetah’ (from 1982)
- ‘America’ (from 1985)
- RS/6000 (from 1986)

Power ISA Timeline

- Abbreviated Lineage of the Power ISA
  - Greater than 30 years of innovation and a developed ecosystem
  - Instruction heritage shown for ***Power ISA 3.1***

Instruction Heritage	Note	# Instr.	Cum Instr.	Open ISA
POWER (P1)	Base	218	218	*Contributing*
POWER (P2)		6	224	*Contributing*
PowerPC (P3)	64b	119	343	*Contributing*
PowerPC 2.00 (P4)		7	350	*Contributing*
PowerPC 2.01		2	352	*Contributing*
PowerPC 2.02 (P5)		14	366	*Contributing*
Power ISA 2.03	SIMD-VMX	171	537	*Contributing*
Power ISA 2.05 (P6)		105	642	*Contributing*
Power ISA 2.06 (P7)	SIMD-VSX	189	831	*Contributing*
Power ISA 2.07 (P8)		111	942	*Contributing*
Power ISA 3.0 (P9)		231	1173	***Compliance***
Power ISA 3.1 (P10)	Prefix	246	1419	***Compliance***



A rich history – just some highlights

- POWER2 (P2SC + ASIC) - Deep Blue chess champion
- RAD750 (derived from ppc750 iMac G3) - Mars Rover Perseverance
- Playstation3, Xbox360, Nintendo Wii - Games trifecta
- POWER7 - Jeopardy champion
- POWER9 - Summit and Sierra

The Linux Foundation Announces New Open Hardware Technologies and Collaboration

THE LINUX FOUNDATION | 21 AUGUST 2019

OpenPOWER to become Linux Foundation project to advance development of data-driven architectures required for hybrid cloud environments and increasingly intensive workloads

SAN DIEGO, CA August 21, 2019 – The Linux Foundation, the nonprofit organization enabling mass innovation through open source, today announced that the OpenPOWER Foundation will become a project hosted at The Linux Foundation. The project includes IBM's open POWER Instruction Set Architecture (ISA) and contributed Source Design Implementations required to support data-driven hardware for intensive workloads like Artificial Intelligence (AI).

OpenPOWER is the open steward for the Power Architecture and has the support of 350 members, including IBM, Google, Inspur Power Systems, Yadro, Hitachi, Wistron, Mellanox, NVIDIA, and Red Hat.

The governance model within the Linux Foundation gives software developers assurance of compatibility while developing AI and hybrid cloud native applications that take advantage of POWER's rich feature set and open compute hardware and software ecosystems.

OpenPOWER

POWER ISA: Open Contribution Timeline

- **August 20, 2019** – Open ISA Announcement at NA OpenPOWER Summit
- **February 13, 2020** – Final Draft of the End User License Released by OPF:
  - https://openpowerfoundation.org/final-draft-of-the-power-isa-eula-released/
- **April 2020** – POWER ISA 3.0c contribution to OPF
  - Same as POWER ISA 3.0b except for
    - Compliancy Subsets
    - Custom Extension Space (Sandbox)
    - SMF Feature
- **May 2020** – POWER ISA 3.1 contribution to OPF (latest is 3.1c)
- **May 2020** – POWER ISA Workgroup Chartered in OPF

Technical Specifications

Search...

Matching items

- 64-bit ELF ABI Specification for OpenPOWER Architecture
- Advanced Accelerator Adapter Electro-Mechanical Specification
- Coherent Accelerator Interface Architecture
- Field Replaceable Unit Service Interface Specification
- Instruction Set Architecture
- IO Design Architecture (IODA) Specification
- Linux on POWER Architecture Reference
- Memory Bus Specification
- OpenCAPI POWER Platform Architecture Guide
- PSL / AFU Interface Specification
- Vector Intrinsics Porting Guide
- Vector Intrinsics Programming Reference Specification

Instruction Set Architecture

Download version 3.1c 2024-05-26

Owned by : Instruction Set Architecture TWG

Compliance : Test Suite & Harness

The Power Instruction Set Architecture (ISA) Version is a specification that describes the architecture used by POWER processors.

It defines the instructions the processor executes.

It is comprised of three books and a set of appendices.

- Book I
  - Power ISA User Instruction Set Architecture, covers the base instruction set and related facilities available to the application programmer.
- Book II
  - Power ISA Virtual Environment Architecture, defines the storage model and related instructions and facilities available to the application programmer.
- Book III
  - Power ISA Operating Environment Architecture, defines the supervisor instructions and related facilities.

Request For Change Feedback for "non ISA TWG members"

This feedback process uses the RFC method, which we divide in 3 types : *minor changes*, *formal proposal*, **or** *RFC*.

version 3.1c 2024-05-26

Data cleanup

download

version 3.1b 2021-09-14

Incorporate errata.

download

version 3.1 2020-05-02

Initial release.

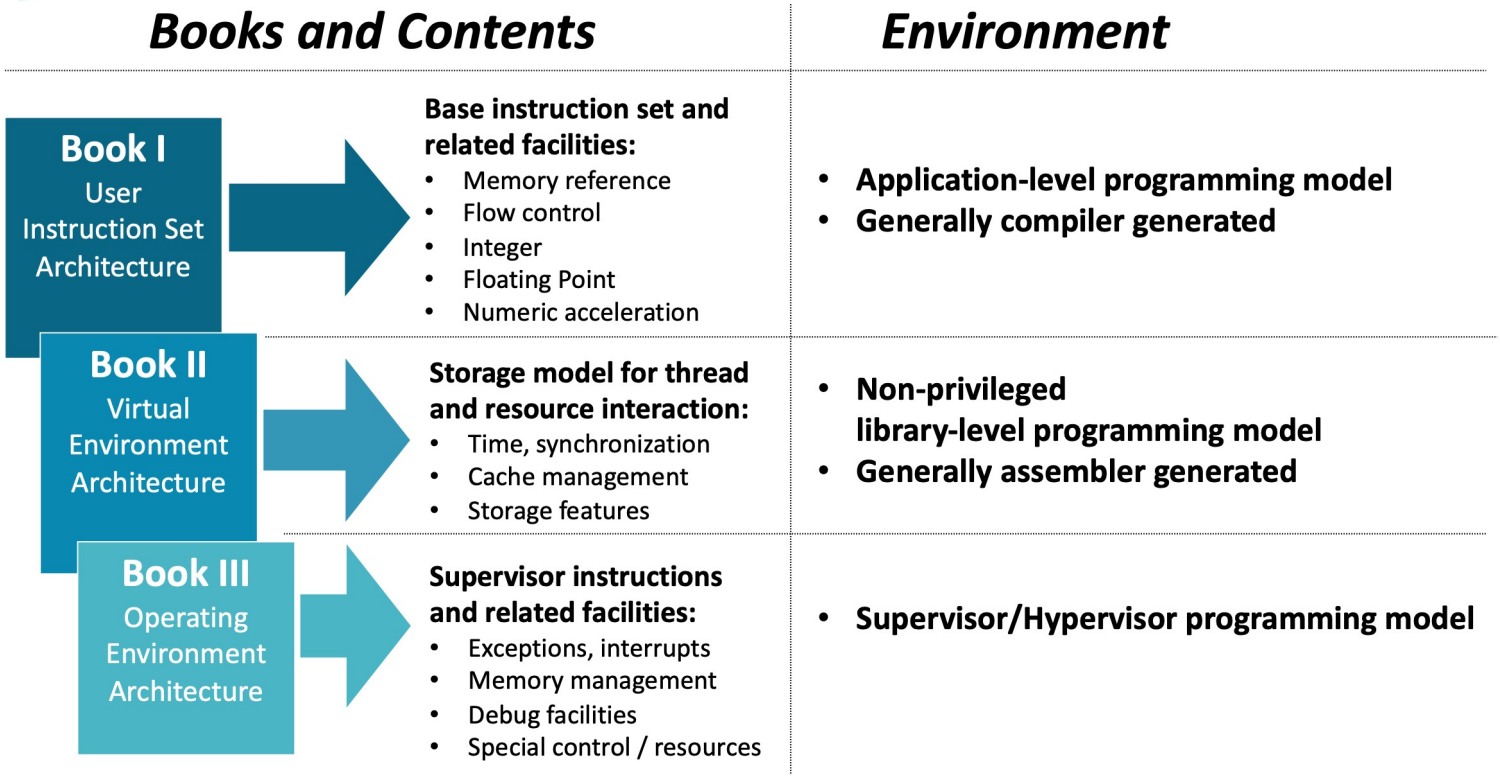
download

version 3.0c 2020-05-01

Incorporate errata.

download

Power Architecture Format – The Books

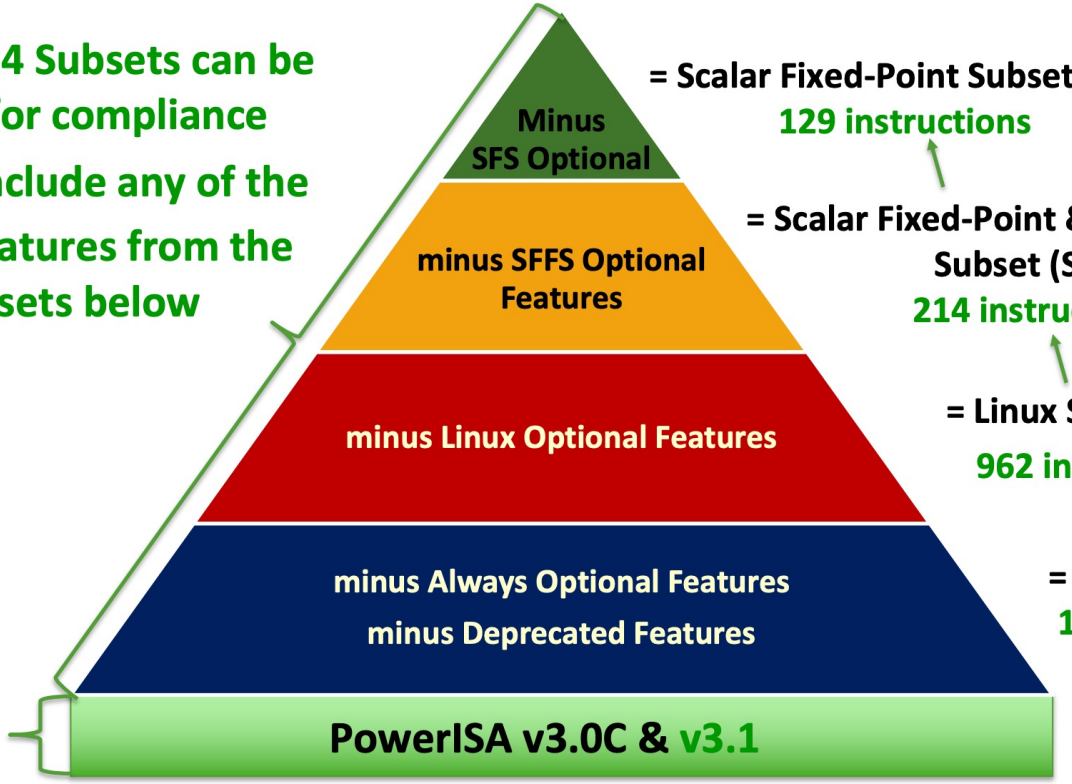


Open Power ISA: Compliancy

- **A supporting / compliant design must:**
  - Support the ***Base*** architecture (never optional)
  - ***And*** support at least one of the compliancy sub-sets:
    - ***ACS***: **AIX** Compliancy Subset
    - ***LCS***: **Linux** Compliancy Subset
    - ***SFFS***: **Scalar Fixed-Point + Floating-Point** Compliancy Subset
    - ***SFS***: **Scalar Fixed-Point** Compliancy Subset
  - ***And may*** support any **optional features** for the compliancy subset
    - ***All optional features must be supported in their entirety per compliancy specification***
  - ***And may*** support **custom extensions using the architecture sandbox**
  - ***And may*** utilize a combination of hardware and firmware
    - Any firmware must rely on on base architecture, the included optional features and the included custom extensions; must honor restrictions specified in the architecture for use of storage; and must honor explicitly prohibited uses.

Open Power ISA: Compliancy Subsets

- Any of the 4 Subsets can be chosen for compliance
- Can also include any of the optional features from the other sets below



= Scalar Fixed-Point Subset (SFS)
129 instructions

= Scalar Fixed-Point & Floating- Point Subset (SFFS)
214 instructions

= Linux Subset (LCS)
962 instructions

= AIX Subset (ACS)
1099 instructions

Full v3.1 ISA
1419 instructions

- Full Arch Compliance

Power ISA: Compliancy Subsets

Refer to official PowerISA document for compliancy subset definition and additional details

Copy-paste-accel (CPA)
Secure-mem-facility (SMF)
Data-stream-prefetch (STM)
Non-coherent-mem (M=0)
Wr-tru-req-mem (W=1)
Power-management (PM)
Matrix-math-assist (MMA)

Always Optional Features *<320i>*

AIL-HAIL-programmability
Atomic-mem-ops (AMO)
Big-endian (BE)
Branch-history-buf (BHRB)
Event-based-branch (EBB)
EVIRT-programmability
SLB / HPT xlate
Proc-compat-reg (PCR)
Broadcast-TLBIE
Control-reg (CTRL)
Quad-prec-float (QFP)
Decimal-float (DFP)
Load-store-multiple (LM)
Load-store-string (LS)
SMT

AIX Subset (ACS) *<1099i>*

SIMD-VMX-VSX
Nested radix xlate (ROR)
OV modifying ops
SF=1 (64-bit mode)
LE-mode
LPAR

Linux Subset (LCS) *<962i>*

Scalar-Binary-FP

Scalar Fixed-Point & Floating- Point Subset (SFFS) *<214i>*

Base Architecture

Scalar Fixed-Point Subset (SFS) *<129i>*

Power ISA: Compliancy Subsets + Sandbox

Architecture Sandbox

For customization with *limited* applicability

Opcode-22	[H]FSCR(8:9)	FPSCR(14:15)	Interrupt 0x0FE0
SPR-704:719,720:735	XER(54:55)	VSCR 96,112	

Copy-paste-accel (CPA)	Non-coherent-mem (M=0)	Matrix-math-assist (MMA)	
Secure-mem-facility (SMF)	Wr-tru-req-mem (W=1)	**Always Optional Features** *<320i>*	
Data-stream-prefetch (STM)	Power-management (PM)		

AIL-HAIL-programmability	EVIRT-programmability	Quad-prec-float (QFP)	**AIX Subset (ACS)** *<1099i>*
Atomic-mem-ops (AMO)	SLB / HPT xlate	Decimal-float (DFP)	
Big-endian (BE)	Proc-compat-reg (PCR)	Load-store-multiple (LM)	
Branch-history-buf (BHRB)	Broadcast-TLBIE	Load-store-string (LS)	
Event-based-branch (EBB)	Control-reg (CTRL)	SMT	

SIMD-VMX-VSX	SF=1 (64-bit mode)	**Linux Subset (LCS)** *<962i>*
Nested radix xlate (ROR)	LE-mode	
OV modifying ops	LPAR	

Scalar-Binary-FP — **Scalar Fixed-Point & Floating- Point Subset (SFFS)** *<214i>*

Base Architecture — **Scalar Fixed-Point Subset (SFS)** *<129i>*

The Open Power ISA: Architecture Sandbox

- **The compliancy subset permits "custom extensions"**
  - The architecture sandbox features are intended to allow implementation dependent customization

- **For facilities with broad applicability:**
  - Developers are ***strongly*** encouraged to submit a proposal for adoption into the architecture.
  - Adopted proposals will become optional or required features of the architecture
    - These will be assigned resources that are not in the architecture sandbox to avoid fragmentation of the architecture.
  - Facilities described in proposals that are not adopted into the architecture may be implemented as Custom Extensions using the architecture sandbox.

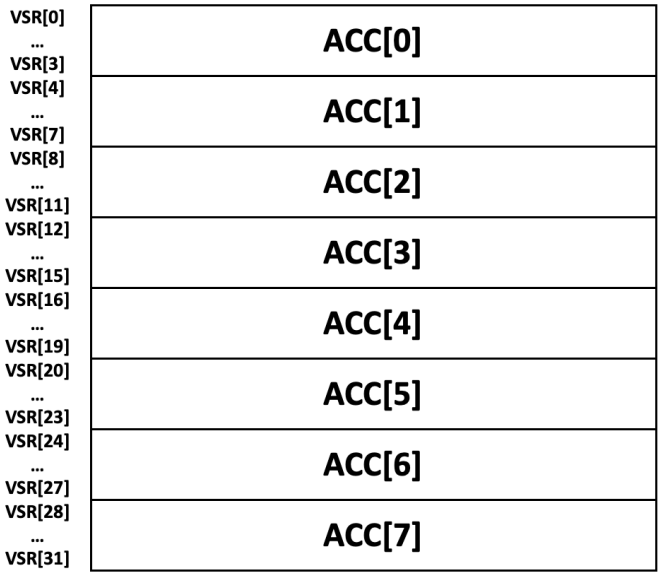
- **System software and toolchain support of Custom Extensions is not guaranteed.**
  - Developers are encouraged to provide a means to disable custom extensions to present an architecture that is supported by standard system software and toolchain.

PowerISA 3.1: Foundation for Expansion

The PowerISA 3.1 includes a number of new features (see specification / preface for details):

- General: Byte reverse instructions, Vector Integer Multiply/Divide/Modulo Instructions, 128-bit Binary Integer Operations, Set Boolean Extension, String Operations, Test LSB by Byte Operation, VSX Scalar Minimum/Maximum/Compare Quad-Precision
- SIMD: VSX 32-byte Storage Access Operations, SIMD Permute-Class Operations, Bit-Manipulation Operations, VSX Load/Store Rightmost Element Operations, VSX Mask Manipulation Operations, VSX PCV Generate Operations for expand/compress,
- Translation Management Extensions
- Copy/Paste Extensions
- Persistent Storage / Store Sync
- Pause / Wait-reserve
- Hypervisor Interrupt Location Control
- Reduced-Precision: Outer Product Operations
  Matrix Math Assist
- Debug: BHRB Filtering updates, Multiple DEAW, New Performance Monitor SPRs, Performance Monitor Facility Sampling Security
- Instruction Prefix Support : 8B and modifying opcodes
  - Prefixed addi Instruction and Prefixed Load/Store Instructions and Addressing
  - CMODX Extension for Prefix
  - See next slide

New matrix math instruction set architecture (MMA facility)



- Accumulators (ACC) are 4×4 arrays of 32-bit elements (see below for 64-bit extension)

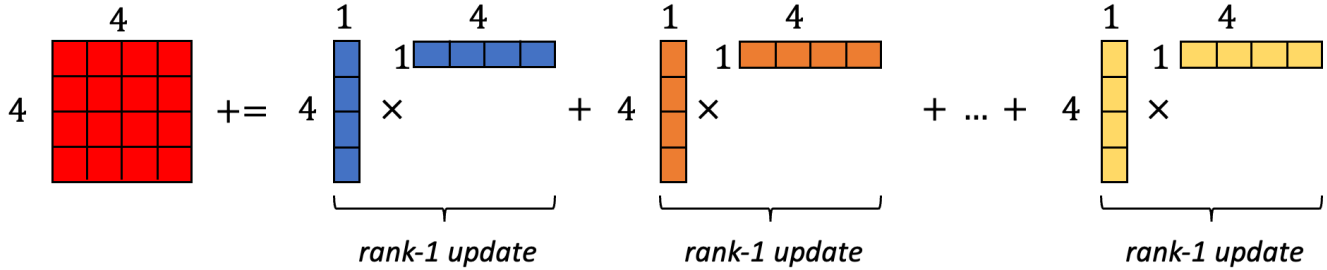
$$A = \begin{bmatrix} a_{00} & a_{01} & a_{02} & a_{03} \\ a_{10} & a_{11} & a_{12} & a_{13} \\ a_{20} & a_{21} & a_{22} & a_{23} \\ a_{30} & a_{31} & a_{32} & a_{33} \end{bmatrix}$$

- Vector-scalar registers (VSR) are 128-bit wide and can hold
  - 4 single-precision (32-bit) floating-point values
  - 8 half-precision (16-bit) floating-point values
  - 16 signed/unsigned (8-bit) integer values
  - …
- Instructions take 1 accumulator (input/output) and 2 vector-scalar registers (input)

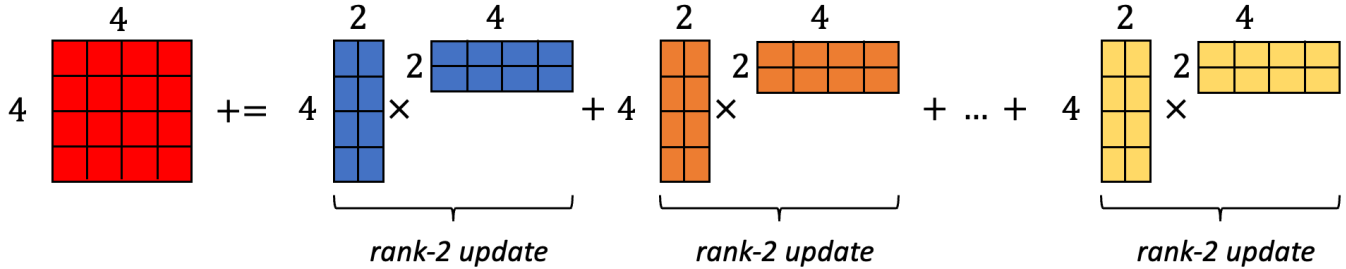
$$\langle \textbf{operation} \rangle \quad A, X, Y$$

Rank-k update with 32-bit and 16-bit elements

- 32-bit: Each VSR holds 1 column (row) of 4×32-bit elements

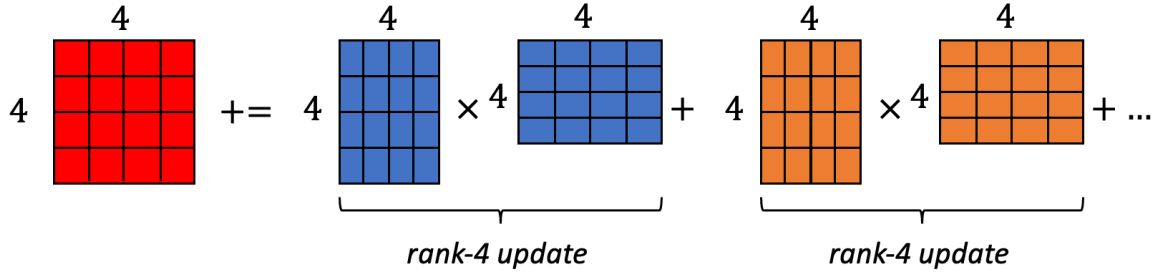


- 16-bit: Each VSR holds 2 columns (rows) of 4×16-bit elements

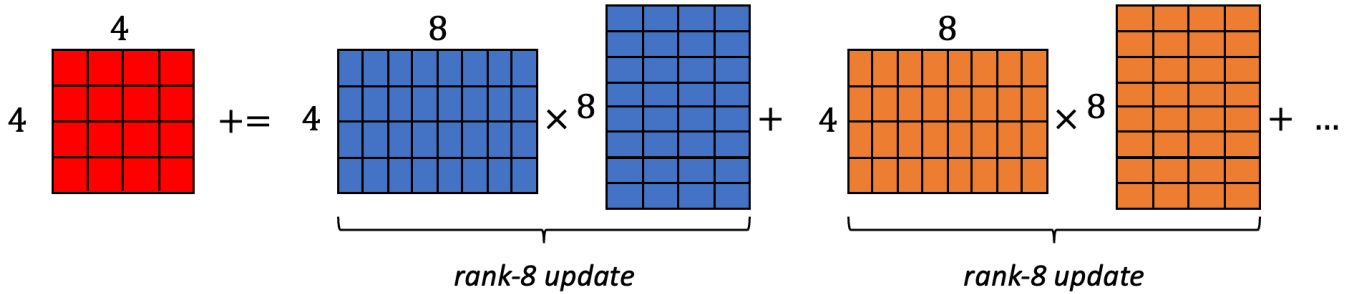


Rank-k update with 8-bit and 4-bit elements

- 8-bit: Each VSR holds 4 columns (rows) of 4×8-bit elements

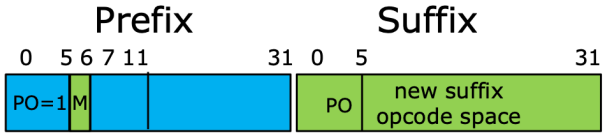


- 4-bit: Each VSR holds 8 columns (rows) of 4×4-bit elements



OpenPOWER

PowerISA 3.1 Prefix architecture



- **Prefix architecture, Primary-Opcode=1**
  - RISC-friendly variable length instructions:
    - New 8B instruction space lays the foundation for future ISA expansion
    - Always 4B instruction alignment
  - Two forms: modifying (M=1) and 8B opcode (M=0)
    - Modifying: prefix extends function of existing instructions
    - 8B opcode: provides new opcode space for expansion, multi-operand instructions, etc.
  - PC-relative addressing: reduced path-length with new Power ABI support
  - MMA lane masking : mask by lane for MMA operations
  - Additional instructions and capabilities
- **Generous room for expanded capabilities including opcode space and additional modifiers**

PowerISA 3.1 Prefix architecture

Table 11: Primary Opcode Map for Opcode Space 1 (64-bit instruction encoding) (suffix bits 0-5)

Primary opcodes of suffixes of 8[M](LS|RR)-form prefixed instructions are mapped to opcode space 1

	000	001	010	011	100	101	110	111	
000	1-00 {reserved}	1-01 {reserved}	1-02	1-03	1-04	1-05 {reserved}	1-06	1-07	000
001	1-08	1-09 {reserved}	1-10	1-11	1-12	1-13	1-14	1-15	001
011	1-16	1-17	1-18	1-19	1-20	1-21	1-22 {reserved}	1-23	011
010	1-24	1-25	1-26	1-27	1-28	1-29	1-30	1-31	010
110	1-32 **EXT132** {extended}	1-33 **EXT133** {extended}	1-34 **EXT134** {extended}	1-35	1-36	1-37	1-38	1-39	110
111	1-40	1-41 I **plwa** v3.1 8LS:D	1-42 I **plxsd** v3.1 8LS:D	1-43 I **plxssp** v3.1 8LS:D	1-44	1-45	1-46 I **pstxsd** v3.1 8LS:D	1-47 I **pstxssp** v3.1 8LS:D	111
101	1-48	1-49	1-50 I **plxv** v3.1 8LS:D	1-50 I **plxv** v3.1 8LS:D	1-52	1-53	1-54 I **pstxv** v3.1 8LS:D	1-54 I **pstxv** v3.1 8LS:D	101
100	1-56 I **plq** v3.1 8LS:D	1-57 I **pld** v3.1 8LS:D	1-58 I **plxvp** v3.1 8LS:D	1-59	1-60 I **pstq** v3.1 8LS:D	1-61 I **pstd** v3.1 8LS:D	1-62 I **pstxvp** v3.1 8LS:D	1-63	100
	000	001	010	011	100	101	110	111	

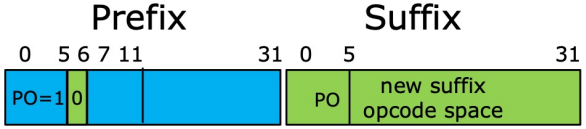


Table 12: PREFIX: Opcode Map (64-bit instruction encoding) (prefix bits 6:11)

	000	001	010	011	100	101	110	111	
000	00 0... **8LS-form** v3.1								000
001									001
010	01 0000 **8RR-form** v3.1								010
011									011
100	10 0... **MLS-form** v3.1								100
101									101
110	11 0000 **MRR-form** v3.1								110
111		11 1001 **MMIRR-form** v3.1							111
	000	001	010	011	100	101	110	111	

Generous room for expanded capabilities including opcode space and additional modifiers

Summary

- OpenPOWER is an open and license-free for compliant architectures
- Maintained by the OpenPOWER foundation working group
- Compliancy subsets allow for a very wide range of implementations
  - from very small cores that can fit on a small FPGA
  - to implementations that support all instructions and can run multiple enterprise OS's

Resources

The Evolution of RISC technology at IBM:

- https://acg.cis.upenn.edu/milom/cis501-Fall11/papers/cocke-RISC.pdf

ISA:

- https://openpowerfoundation.org/specifications/isa

Microwatt core:

- https://github.com/antonblanchard/microwatt

Other open cores:

- https://github.com/openpower-cores

Acknowledgements

This presentation draws from prior presentations by

Brian Thompto

José Moreira