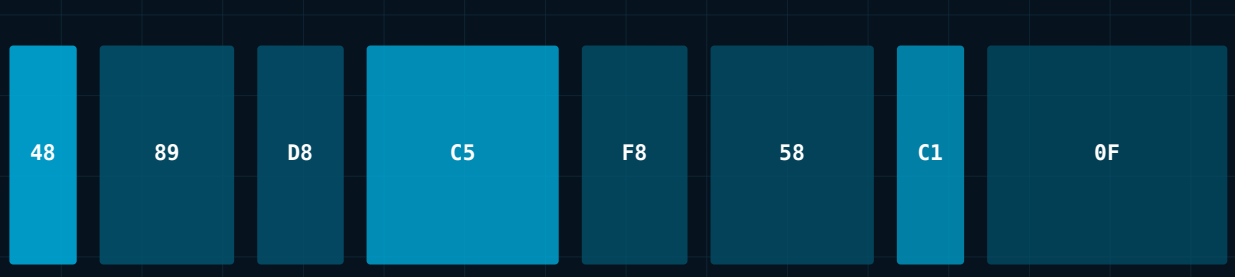
THE PROCESSOR ARCHITECTURE ATLAS SERIES · VOLUME 01

The x86-64 Architecture Atlas

Registers, Instruction Encoding, System Programming, Memory, Interrupts, SIMD, and ABI

X64



GPR · RFLAGS · ENC · PAGING · IDT · AVX · ABI

A3 PORTRAIT · VISUAL SYSTEMS REFERENCE · FIRST EDITION

REFERENCE SCOPE

- **Variable-length encoding**
- Legacy-to-modern register state
- Paging and privilege
- SSE / AVX / AVX-512 / AMX

Prepared by Ayman Alheraki

Sponsored by https://ForgeVM.org Project

64 interior pages · English edition · Official-reference baseline

VOLUME 01

The x86-64 Architecture Atlas

Registers, Instruction Encoding, System Programming, Memory, Interrupts, SIMD, and ABI

X64

Prepared by Ayman Alheraki

Sponsored by https://ForgeVM.org Project

EDITION

First Edition · 2026

FORMAT

A3 Portrait · 297 × 420 mm

SCOPE STATEMENT

This volume is a dense visual reference for software-visible x86-64 architecture. It is designed for operating-system developers, compiler and JIT engineers, debugger authors, reverse engineers, firmware developers, and low-level programmers.

PUBLICATION NOTE

Official architecture specifications remain normative. This atlas compresses programming-visible concepts into traceable diagrams, tables, and cross-references; it does not replace formal instruction pseudocode or vendor-specific implementation manuals.

NAVIGATION SYSTEM

How to Read This Volume

A consistent editorial language for dense technical information

REG — ARCHITECTURE STATE

Registers, flags, system state, and privilege-visible control.

ENC — ENCODING FIELD

Instruction bits, prefixes, opcode maps, and immediate reconstruction.

MEM — MEMORY EFFECT

Address formation, ordering, cache/TLB effect, and translation behavior.

EXC — EXCEPTION PATH

Fault, trap, interrupt, privilege transition, and return mechanism.

STS — STATUS MARKER

Required, optional, legacy, deprecated, ratified, or implementation-defined.

SRC — SOURCE CODE

Compact official-reference key printed at the foot of each technical page.

STATUS LANGUAGE

- REQUIRED
- OPTIONAL
- LEGACY
- DEPRECATED
- RATIFIED
- IMPLEMENTATION-DEFINED

TABLE READING ORDER

1 **Name and purpose**
2 Operands and width
3 Architectural state affected
4 Privilege and memory ordering
5 Exceptions, caveats, and source

PAGE ARCHITECTURE

Detailed Contents I

64 interior pages · covers excluded

PAGE ARCHITECTURE

Detailed Contents II

64 interior pages · covers excluded

TECHNICAL PAGE PROOF

x86-64 Programmer-Visible Architecture Map

Editorial sample: architecture state, privilege, memory, and extension layers

ARCHITECTURAL STATE

- RAX-R15
- RIP
- RFLAGS
- CR0 / CR2 / CR3 / CR4
- XCR0 / EFER
- XMM-YMM-ZMM
- K0-K7 / TILECFG
- Descriptor and interrupt state

MODE AND PRIVILEGE

Long mode · Rings 0-3 · GDT/IDT/TSS

INSTRUCTION MODEL

1-15 bytes · prefixes · opcode maps · ModR/M · SIB

MEMORY SYSTEM

Canonical VA · 4/5-level paging · PCID · TLB

EVENT DELIVERY

Exceptions · IDT gates · APIC · IRETQ

MODERN COMPUTE

SSE · AVX2 · AVX-512 · AMX · crypto

Every full-volume technical page maps back to one or more layers shown here.

Sources: Intel SDM Vol. 1 & 3 · AMD APM Vol. 2

TECHNICAL PAGE PROOF

Register State and Instruction Encoding

High-density A3 portrait layout with vector tables and concise explanations

REGISTER / STATE MAP

RAX	RBX	RCX	RDX
RSI	RDI	RBP	RSP
R8	R9	R10	R11
R12	R13	R14	R15

ENCODING / FIELD MODEL

Prefixes	0-4+ bytes	Legacy / REX / VEX / EVEX
Opcode	1-3+ bytes	Primary and extended maps
ModR/M	1 byte	mod · reg/opcode · r/m
SIB	0 or 1 byte	scale · index · base
Displacement	0/1/2/4/8 bytes	Address contribution
Immediate	0/1/2/4/8 bytes	Constant or relative target

Variable length requires ordered prefix and field decoding; architecturally valid length is at most 15 bytes.

Sources: Intel SDM Vol. 1-2 · AMD APM Vol. 1 & 3

SOURCE INTEGRITY

Official English References

The complete volume uses section-level source keys on every technical page

PRIMARY NORMATIVE SOURCES

01 **Intel® 64 and IA-32 Architectures Software Developer’s Manual, Volumes 1-4.**

02 AMD64 Architecture Programmer’s Manual, Volumes 1-5, Revision 4.10.

03 System V Application Binary Interface: AMD64 Architecture Processor Supplement.

04 Microsoft x64 Calling Convention and PE/COFF Specification.

05 DWARF Debugging Information Format, Version 5.

REFERENCE-KEY EXAMPLE

INTEL-SDM-V1 §3.4 · INTEL-SDM-V3A §4 · AMD-APM-V2 §5

Revision and access-date metadata are stored in the bibliography and frozen for each published edition. Corrections are tracked through a public errata register.

EDITORIAL COMMITMENT

- **No unsourced register maps or bit positions.**
- No mnemonic equivalence without behavior verification.
- Architecture revision shown for optional and recent features.
- Official manuals remain normative when ambiguity exists.

VOLUME 01 · X86-64

A reference built for the moment when a diagram must be faster than a manual search — without sacrificing traceability.

DESIGNED FOR

— Operating-system and kernel developers

— Compiler, assembler, linker, and JIT engineers

— Firmware and hypervisor developers

— Debugger, profiler, and reverse-engineering specialists

— Advanced low-level programmers and architecture educators

SERIES PRINCIPLE

One architecture per volume. One dedicated comparison volume. A unified visual grammar across registers, instruction encodings, privilege, virtual memory, exceptions, atomics, vectors, ABI, and binary interfaces.

Prepared by Ayman Alheraki

Sponsored by https://ForgeVM.org Project

THE PROCESSOR ARCHITECTURE ATLAS SERIES

Independent technical reference · English edition · A3 portrait

ISBN / BARCODE AREA

The Arm AArch64 Architecture Atlas

Registers, A64 Encoding, Exception Levels, Memory Translation, SVE/SME, Barriers, and ABI

A64

X0-X30 · PSTATE · A64 · EL0-3 · MMU · SVE · ABI

A3 PORTRAIT · VISUAL SYSTEMS REFERENCE · FIRST EDITION

REFERENCE SCOPE

- **Fixed 32-bit A64 encoding**
- Exception levels and security states
- Stage 1 / Stage 2 translation
- NEON / SVE / SVE2 / SME

Prepared by Ayman Alheraki

Sponsored by https://ForgeVM.org Project

56 interior pages · English edition · Official-reference baseline

VOLUME 02

The Arm AArch64 Architecture Atlas

Registers, A64 Encoding, Exception Levels, Memory Translation, SVE/SME, Barriers, and ABI

A64

Prepared by Ayman Alheraki

Sponsored by https://ForgeVM.org Project

EDITION

First Edition · 2026

FORMAT

A3 Portrait · 297 × 420 mm

SCOPE STATEMENT

This volume is a dense visual reference for software-visible Arm AArch64 architecture. It is designed for operating-system developers, compiler and JIT engineers, debugger authors, reverse engineers, firmware developers, and low-level programmers.

PUBLICATION NOTE

Official architecture specifications remain normative. This atlas compresses programming-visible concepts into traceable diagrams, tables, and cross-references; it does not replace formal instruction pseudocode or vendor-specific implementation manuals.

NAVIGATION SYSTEM

How to Read This Volume

A consistent editorial language for dense technical information

REG — ARCHITECTURE STATE

Registers, flags, system state, and privilege-visible control.

ENC — ENCODING FIELD

Instruction bits, prefixes, opcode maps, and immediate reconstruction.

MEM — MEMORY EFFECT

Address formation, ordering, cache/TLB effect, and translation behavior.

EXC — EXCEPTION PATH

Fault, trap, interrupt, privilege transition, and return mechanism.

STS — STATUS MARKER

Required, optional, legacy, deprecated, ratified, or implementation-defined.

SRC — SOURCE CODE

Compact official-reference key printed at the foot of each technical page.

STATUS LANGUAGE

- REQUIRED
- OPTIONAL
- LEGACY
- DEPRECATED
- RATIFIED
- IMPLEMENTATION-DEFINED

TABLE READING ORDER

1 **Name and purpose**
2 Operands and width
3 Architectural state affected
4 Privilege and memory ordering
5 Exceptions, caveats, and source

PAGE ARCHITECTURE

Detailed Contents I

56 interior pages · covers excluded

PAGE ARCHITECTURE

Detailed Contents II

56 interior pages · covers excluded

TECHNICAL PAGE PROOF

Arm AArch64 Programmer-Visible Architecture Map

Editorial sample: architecture state, privilege, memory, and extension layers

ARCHITECTURAL STATE

- X0-X30
- SP / PC
- PSTATE / NZCV
- SCTLR_ELx
- TTBR / TCR / MAIR
- V0-V31
- Z / P / ZA state
- Exception and vector state

MODE AND PRIVILEGE

AArch64 · EL0-EL3 · Security states

INSTRUCTION MODEL

Fixed 32-bit A64 · major decode groups

MEMORY SYSTEM

Stage 1/2 · 4K/16K/64K granules · ASID

EVENT DELIVERY

Sync · IRQ · FIQ · SError · ERET

MODERN COMPUTE

NEON · SVE · SVE2 · SME · crypto

Every full-volume technical page maps back to one or more layers shown here.

Source: Arm Architecture Reference Manual for A-profile Architecture

TECHNICAL PAGE PROOF

Register State and A64 Encoding

High-density A3 portrait layout with vector tables and concise explanations

REGISTER / STATE MAP

X0	X1	X2	X3	X4	X5	X6	X7
X8	X9	X10	X11	X12	X13	X14	X15
X16	X17	X18	X19	X20	X21	X22	X23
X24	X25	X26	X27	X28	X29	X30	SP

ENCODING / FIELD MODEL

Major op	Bits 31:25	Top-level A64 decode group
sf / size	Variant field	Operation width or element size
Rm / Rn / Rd	5 bits each	Source and destination registers
imm fields	Variable width	Immediate value reconstruction
shift / extend	Subfields	Operand transformation
cond / op	Control fields	Condition or operation selector

A64 instructions are 32 bits; operation classes are recognized from high-order encoding fields.

Source: Arm A64 Instruction Set Architecture and Arm ARM

SOURCE INTEGRITY

Official English References

The complete volume uses section-level source keys on every technical page

PRIMARY NORMATIVE SOURCES

01 **Arm® Architecture Reference Manual for A-profile Architecture.**

02 Procedure Call Standard for the Arm 64-bit Architecture (AAPCS64).

03 ELF for the Arm 64-bit Architecture (AAELF64).

04 Arm® System Registers Reference.

05 DWARF for the Arm 64-bit Architecture.

REFERENCE-KEY EXAMPLE

ARM-ARM D1.3 · AAPCS64 §6 · AAELF64 §4

Revision and access-date metadata are stored in the bibliography and frozen for each published edition. Corrections are tracked through a public errata register.

EDITORIAL COMMITMENT

- **No unsourced register maps or bit positions.**
- No mnemonic equivalence without behavior verification.
- Architecture revision shown for optional and recent features.
- Official manuals remain normative when ambiguity exists.

VOLUME 02 · ARM AARCH64

A reference built for the moment when a diagram must be faster than a manual search — without sacrificing traceability.

DESIGNED FOR

— Operating-system and kernel developers

— Compiler, assembler, linker, and JIT engineers

— Firmware and hypervisor developers

— Debugger, profiler, and reverse-engineering specialists

— Advanced low-level programmers and architecture educators

SERIES PRINCIPLE

One architecture per volume. One dedicated comparison volume. A unified visual grammar across registers, instruction encodings, privilege, virtual memory, exceptions, atomics, vectors, ABI, and binary interfaces.

Prepared by Ayman Alheraki

Sponsored by https://ForgeVM.org Project

THE PROCESSOR ARCHITECTURE ATLAS SERIES

Independent technical reference · English edition · A3 portrait

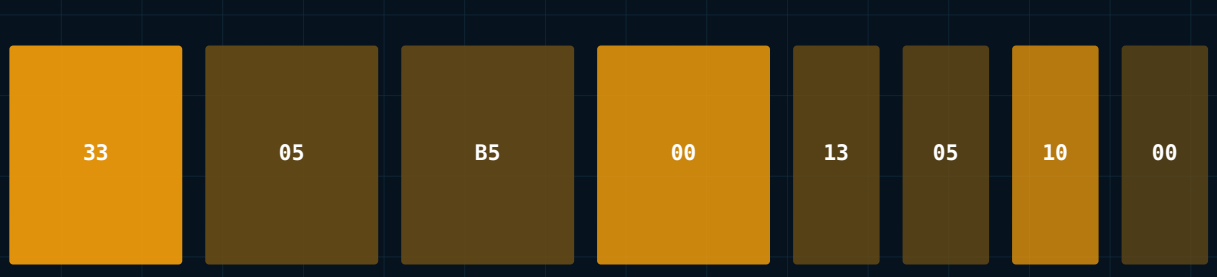
ISBN / BARCODE AREA

THE PROCESSOR ARCHITECTURE ATLAS SERIES · VOLUME 03

The RISC-V Architecture Atlas

Base ISAs, Extensions, Encoding, Privilege, CSRs, Virtual Memory, Vectors, Atomics, and psABI

RV



x0-x31 | CSR | FMT | U/S/M | SV48 | RVV | psABI

A3 PORTRAIT · VISUAL SYSTEMS REFERENCE · FIRST EDITION

REFERENCE SCOPE

- **Modular ISA and extension naming**
- R/I/S/B/U/J instruction formats
- Privilege modes and CSRs
- RVV vector-length-agnostic design

Prepared by Ayman Alheraki

Sponsored by https://ForgeVM.org Project

56 interior pages · English edition · Official-reference baseline

VOLUME 03

The RISC-V Architecture Atlas

Base ISAs, Extensions, Encoding, Privilege, CSRs, Virtual Memory, Vectors, Atomics, and psABI

RV

Prepared by Ayman Alheraki

Sponsored by https://ForgeVM.org Project

EDITION

First Edition · 2026

FORMAT

A3 Portrait · 297 × 420 mm

SCOPE STATEMENT

This volume is a dense visual reference for software-visible RISC-V architecture. It is designed for operating-system developers, compiler and JIT engineers, debugger authors, reverse engineers, firmware developers, and low-level programmers.

PUBLICATION NOTE

Official architecture specifications remain normative. This atlas compresses programming-visible concepts into traceable diagrams, tables, and cross-references; it does not replace formal instruction pseudocode or vendor-specific implementation manuals.

NAVIGATION SYSTEM

How to Read This Volume

A consistent editorial language for dense technical information

REG — ARCHITECTURE STATE

Registers, flags, system state, and privilege-visible control.

ENC — ENCODING FIELD

Instruction bits, prefixes, opcode maps, and immediate reconstruction.

MEM — MEMORY EFFECT

Address formation, ordering, cache/TLB effect, and translation behavior.

EXC — EXCEPTION PATH

Fault, trap, interrupt, privilege transition, and return mechanism.

STS — STATUS MARKER

Required, optional, legacy, deprecated, ratified, or implementation-defined.

SRC — SOURCE CODE

Compact official-reference key printed at the foot of each technical page.

STATUS LANGUAGE

- REQUIRED
- OPTIONAL
- LEGACY
- DEPRECATED
- RATIFIED
- IMPLEMENTATION-DEFINED

TABLE READING ORDER

1 **Name and purpose**

2 Operands and width

3 Architectural state affected

4 Privilege and memory ordering

5 Exceptions, caveats, and source

PAGE ARCHITECTURE

Detailed Contents I

56 interior pages · covers excluded

PAGE ARCHITECTURE

Detailed Contents II

56 interior pages · covers excluded

TECHNICAL PAGE PROOF

RISC-V Programmer-Visible Architecture Map

Editorial sample: architecture state, privilege, memory, and extension layers

ARCHITECTURAL STATE

- x0-x31
- PC
- f0-f31
- v0-v31
- mstatus / sstatus
- satp / hgatp
- Trap CSRs
- Debug and counter state

MODE AND PRIVILEGE

U · S · M · optional H extension

INSTRUCTION MODEL

R/I/S/B/U/J · compressed · modular extensions

MEMORY SYSTEM

Sv39/Sv48/Sv57 · ASID · PMP/PMA

EVENT DELIVERY

Traps · delegation · direct/vectored entry

MODERN COMPUTE

M/A/F/D/C/B/V · scalar/vector crypto

Every full-volume technical page maps back to one or more layers shown here.

Sources: RISC-V Unprivileged and Privileged Architecture specifications

TECHNICAL PAGE PROOF

Register ABI Map and Base Encoding Formats

High-density A3 portrait layout with vector tables and concise explanations

REGISTER / STATE MAP

x0/zero	x1/ra	x2/sp	x3/gp	x4/tp	x5/t0	x6/t1	x7/t2
x8/s0	x9/s1	x10/a0	x11/a1	x12/a2	x13/a3	x14/a4	x15/a5
x16/a6	x17/a7	x18/s2	x19/s3	x20/s4	x21/s5	x22/s6	x23/s7
x24/s8	x25/s9	x26/s10	x27/s11	x28/t3	x29/t4	x30/t5	x31/t6

ENCODING / FIELD MODEL

opcode	Bits 6:0	Base operation class
rd	Bits 11:7	Destination register
funct3	Bits 14:12	Minor operation selector
rs1	Bits 19:15	Source register 1
rs2 / imm	Bits 24:20	Source register or immediate fragment
funct7 / imm	Bits 31:25	Major selector or immediate fragment

Base instructions are 32 bits; the low-order opcode bits distinguish standard-length encoding classes.

Source: RISC-V Unprivileged ISA, base instruction formats

SOURCE INTEGRITY

Official English References

The complete volume uses section-level source keys on every technical page

PRIMARY NORMATIVE SOURCES

01 **The RISC-V Instruction Set Manual, Volume I: Unprivileged Architecture, Version 20260120.**

02 The RISC-V Instruction Set Manual, Volume II: Privileged Architecture, Version 20260120.

03 RISC-V ABIs Specification.

04 RISC-V Debug Specification.

05 RISC-V Advanced Interrupt Architecture Specification.

REFERENCE-KEY EXAMPLE

RISCV-UNPRIV §2 · RISCV-PRIV §3 · RISCV-PSABI §1

Revision and access-date metadata are stored in the bibliography and frozen for each published edition. Corrections are tracked through a public errata register.

EDITORIAL COMMITMENT

- **No unsourced register maps or bit positions.**
- No mnemonic equivalence without behavior verification.
- Architecture revision shown for optional and recent features.
- Official manuals remain normative when ambiguity exists.

VOLUME 03 · RISC-V

A reference built for the moment when a diagram must be faster than a manual search — without sacrificing traceability.

DESIGNED FOR

— Operating-system and kernel developers

— Compiler, assembler, linker, and JIT engineers

— Firmware and hypervisor developers

— Debugger, profiler, and reverse-engineering specialists

— Advanced low-level programmers and architecture educators

SERIES PRINCIPLE

One architecture per volume. One dedicated comparison volume. A unified visual grammar across registers, instruction encodings, privilege, virtual memory, exceptions, atomics, vectors, ABI, and binary interfaces.

Prepared by Ayman Alheraki

Sponsored by https://ForgeVM.org Project

THE PROCESSOR ARCHITECTURE ATLAS SERIES

Independent technical reference · English edition · A3 portrait

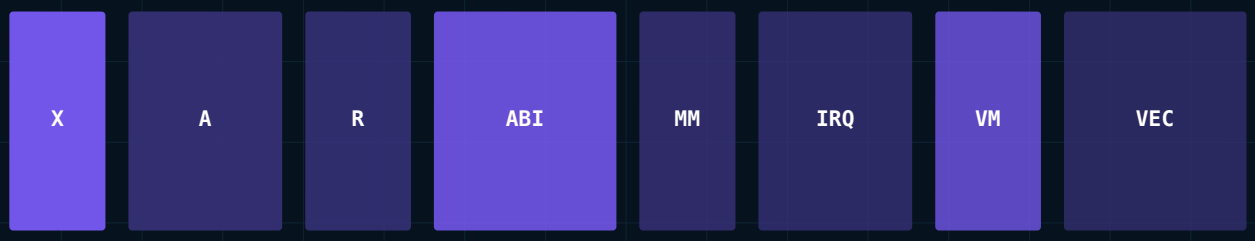
ISBN / BARCODE AREA

THE PROCESSOR ARCHITECTURE ATLAS SERIES · VOLUME 04

The Cross-Architecture Systems Atlas

x86-64 · Arm AArch64 · RISC-V — Direct Comparisons for OS, Compiler, JIT, and Low-Level Development

X · A · R



A3 PORTRAIT · VISUAL SYSTEMS REFERENCE · FIRST EDITION

REFERENCE SCOPE

- **Instruction and register equivalence**
- ABI and binary interface comparison
- Memory models and atomics
- Porting patterns and backend design

Prepared by Ayman Alheraki

Sponsored by https://ForgeVM.org Project

48 interior pages · English edition · Official-reference baseline

VOLUME 04

The Cross-Architecture Systems Atlas

x86-64 · Arm AArch64 · RISC-V — Direct Comparisons for OS, Compiler, JIT, and Low-Level Development

X·A·R

Prepared by Ayman Alheraki

Sponsored by https://ForgeVM.org Project

EDITION

First Edition · 2026

FORMAT

A3 Portrait · 297 × 420 mm

SCOPE STATEMENT

This volume is a dense visual reference for software-visible Cross-Architecture architecture. It is designed for operating-system developers, compiler and JIT engineers, debugger authors, reverse engineers, firmware developers, and low-level programmers.

PUBLICATION NOTE

Official architecture specifications remain normative. This atlas compresses programming-visible concepts into traceable diagrams, tables, and cross-references; it does not replace formal instruction pseudocode or vendor-specific implementation manuals.

NAVIGATION SYSTEM

How to Read This Volume

A consistent editorial language for dense technical information

REG — ARCHITECTURE STATE

Registers, flags, system state, and privilege-visible control.

ENC — ENCODING FIELD

Instruction bits, prefixes, opcode maps, and immediate reconstruction.

MEM — MEMORY EFFECT

Address formation, ordering, cache/TLB effect, and translation behavior.

EXC — EXCEPTION PATH

Fault, trap, interrupt, privilege transition, and return mechanism.

STS — STATUS MARKER

Required, optional, legacy, deprecated, ratified, or implementation-defined.

SRC — SOURCE CODE

Compact official-reference key printed at the foot of each technical page.

STATUS LANGUAGE

- REQUIRED
- OPTIONAL
- LEGACY
- DEPRECATED
- RATIFIED
- IMPLEMENTATION-DEFINED

TABLE READING ORDER

1 **Name and purpose**
2 Operands and width
3 Architectural state affected
4 Privilege and memory ordering
5 Exceptions, caveats, and source

PAGE ARCHITECTURE

Detailed Contents I

48 interior pages · covers excluded

PAGE ARCHITECTURE

Detailed Contents II

48 interior pages · covers excluded

TECHNICAL PAGE PROOF

Cross-Architecture Programmer-Visible Architecture Map

Editorial sample: architecture state, privilege, memory, and extension layers

ARCHITECTURAL STATE

- Integer registers
- Program counter
- Condition state
- Privilege state
- Translation state
- Vector state
- Exception state
- ABI-visible state

REGISTER MODEL

Overlapping vs orthogonal vs ABI aliases

INSTRUCTION MODEL

Variable vs fixed vs modular encoding

MEMORY SYSTEM

Page-table geometry and ordering rules

EVENT DELIVERY

Vectors, causes, privilege entry, return

MODERN COMPUTE

SIMD, scalable vectors, matrices, crypto

Every full-volume technical page maps back to one or more layers shown here.

Sources: official Intel, AMD, Arm, and RISC-V architecture manuals

TECHNICAL PAGE PROOF

Direct Architecture Comparison Matrix

High-density A3 portrait layout with vector tables and concise explanations

Property	x86-64	Arm AArch64	RISC-V
Instruction length	1-15 bytes	32-bit A64	32-bit base; 16-bit C; longer encodings reserved
Integer GPRs	16 × 64-bit	31 × 64-bit + SP/ZR views	32 × XLEN; x0 fixed zero
Condition state	RFLAGS	NZCV + conditional select	Compare-and-branch; no global integer flags
Addressing	Base + index×scale + disp; RIP-relative	Base + immediate/register; pre/post-index; literal	Base + signed immediate; AUIPC sequences
Privilege	Rings 0-3	EL0-EL3	U/S/M (+ H)
Memory model	Strong / TSO-oriented	Weakly ordered	RVWMO; optional Ztso
Vector model	SSE/AVX/AVX-512 fixed widths	NEON fixed + SVE scalable	RVV scalable via VLEN/VL
Fast syscall	SYSCALL/SYSRET	SVC + exception return	ECALL + trap return

Editorial rule: equivalent behavior is compared by architectural effect, not by mnemonic similarity.

Sources: official architecture and ABI manuals

SOURCE INTEGRITY

Official English References

The complete volume uses section-level source keys on every technical page

PRIMARY NORMATIVE SOURCES

01 **Official Intel®, AMD, Arm®, and RISC-V architecture manuals.**

02 System V AMD64 ABI, AAPCS64, and RISC-V psABI.

03 Microsoft PE/COFF and x64 ABI documentation.

04 DWARF Debugging Information Format, Version 5.

05 IEEE Standard for Floating-Point Arithmetic.

REFERENCE-KEY EXAMPLE

INTEL-SDM · ARM-ARM · RISCV-UNPRIV · ABI-SPECS

Revision and access-date metadata are stored in the bibliography and frozen for each published edition. Corrections are tracked through a public errata register.

EDITORIAL COMMITMENT

- **No unsourced register maps or bit positions.**
- No mnemonic equivalence without behavior verification.
- Architecture revision shown for optional and recent features.
- Official manuals remain normative when ambiguity exists.

VOLUME 04 · CROSS-ARCHITECTURE

A reference built for the moment when a diagram must be faster than a manual search — without sacrificing traceability.

DESIGNED FOR

— Operating-system and kernel developers

— Compiler, assembler, linker, and JIT engineers

— Firmware and hypervisor developers

— Debugger, profiler, and reverse-engineering specialists

— Advanced low-level programmers and architecture educators

SERIES PRINCIPLE

One architecture per volume. One dedicated comparison volume. A unified visual grammar across registers, instruction encodings, privilege, virtual memory, exceptions, atomics, vectors, ABI, and binary interfaces.

Prepared by Ayman Alheraki

Sponsored by https://ForgeVM.org Project

THE PROCESSOR ARCHITECTURE ATLAS SERIES

Independent technical reference · English edition · A3 portrait

ISBN / BARCODE AREA