

Quick Contents

	RISC-V Reference Card	i
	List of Figures	ix
	Preface	xii
1	Why RISC-V?	2
2	RV32I: RISC-V Base Integer ISA	14
3	RISC-V Assembly Language	32
4	RV32M: Multiply and Divide	44
5	RV32FD: Single/Double Floating Point	48
6	RV32A: Atomic	60
7	RV32C: Compressed Instructions	64
8	RV32V: Vector	72
9	RV64: 64-bit Address Instructions	86
10	RV32/64 Privileged Architecture	100
11	Future RISC-V Optional Extensions	118
Appendix A	RISC-V Instruction Listings	120
Appendix B	Transliteration from RISC-V	168
	Index	174

Contents

List of Figures	x
Preface	xii
1 Why RISC-V?	2
1.1 Introduction	2
1.2 Modular vs. Incremental ISAs	4
1.3 ISA Design 101	5
1.4 An Overview of this Book	10
1.5 Concluding Remarks	11
1.6 To Learn More	12
2 RV32I: RISC-V Base Integer ISA	14
2.1 Introduction	14
2.2 RV32I Instruction formats	14
2.3 RV32I Registers	18
2.4 RV32I Integer Computation	18
2.5 RV32I Loads and Stores	20
2.6 RV32I Conditional Branch	21
2.7 RV32I Unconditional Jump	22
2.8 RV32I Miscellaneous	23
2.9 Comparing RV32I, ARM-32, MIPS-32, and x86-32	23
2.10 Concluding Remarks	24
2.11 To Learn More	26
3 RISC-V Assembly Language	32
3.1 Introduction	32
3.2 Calling convention	32
3.3 Assembly	35
3.4 Linker	40
3.5 Static vs. Dynamic Linking	41
3.6 Loader	42
3.7 Concluding Remarks	42
3.8 To Learn More	42

4	RV32M: Multiply and Divide	44
4.1	Introduction	44
4.2	Concluding Remarks	46
4.3	To Learn More	46
5	RV32FD: Single/Double Floating Point	48
5.1	Introduction	48
5.2	Floating-Point Registers	48
5.3	Floating-Point Loads, Stores, and Arithmetic	49
5.4	Floating-Point Moves and Converts	53
5.5	Miscellaneous Floating-Point Instructions	53
5.6	Comparing RV32FD, ARM-32, MIPS-32, and x86-32 using DAXPY	55
5.7	Concluding Remarks	55
5.8	To Learn More	56
6	RV32A: Atomic	60
6.1	Introduction	60
6.2	Concluding Remarks	62
6.3	To Learn More	62
7	RV32C: Compressed Instructions	64
7.1	Introduction	64
7.2	Comparing RV32GC, Thumb-2, microMIPS, and x86-32	66
7.3	Concluding Remarks	66
7.4	To Learn More	67
8	RV32V: Vector	72
8.1	Introduction	72
8.2	Vector Computation Instructions	73
8.3	Vector Registers and Dynamic Typing	74
8.4	Vector Loads and Stores	75
8.5	Parallelism During Vector Execution	76
8.6	Conditional Execution of Vector Operations	76
8.7	Miscellaneous Vector Instructions	77
8.8	Vector Example: DAXPY in RV32V	78
8.9	Comparing RV32V, MIPS-32 MSA SIMD, and x86-32 AVX SIMD	79
8.10	Concluding Remarks	81
8.11	To Learn More	82
9	RV64: 64-bit Address Instructions	86
9.1	Introduction	86
9.2	Comparison to Other 64-bit ISAs using Insertion Sort	90
9.3	Program size	92
9.4	Concluding Remarks	93
9.5	To Learn More	93

10 RV32/64 Privileged Architecture	100
10.1 Introduction	100
10.2 Machine Mode for Simple Embedded Systems	101
10.3 Machine-Mode Exception Handling	103
10.4 User Mode and Process Isolation in Embedded Systems	106
10.5 Supervisor Mode for Modern Operating Systems	108
10.6 Page-Based Virtual Memory	111
10.7 Identification and Performance CSRs	114
10.8 Concluding Remarks	115
10.9 To Learn More	117
11 Future RISC-V Optional Extensions	118
11.1 “B” Standard Extension for Bit Manipulation	118
11.2 “E” Standard Extension for Embedded	118
11.3 “H” Privileged Architecture Extension for Hypervisor Support	118
11.4 “J” Standard Extension for Dynamically Translated Languages	118
11.5 “L” Standard Extension for Decimal Floating-Point	118
11.6 “N” Standard Extension for User-Level Interrupts	119
11.7 “P” Standard Extension for Packed-SIMD Instructions	119
11.8 “Q” Standard Extension for Quad-Precision Floating-Point	119
11.9 Concluding Remarks	119
A RISC-V Instruction Listings	120
B Transliteration from RISC-V	168
B.1 Introduction	168
B.2 Comparing RV32I, ARM-32, and x86-32 using Tree Sum	170
B.3 Conclusion	171
Index	174