

1. Record Nr.	UNINA9910784549403321
Autore	Sweetman Dominic
Titolo	See MIPS run [[electronic resource] /] / Dominic Sweetman
Pubbl/distr/stampa	San Francisco, Calif., : Morgan Kaufmann Publishers/Elsevier, c2007
ISBN	1-281-02320-5 9786611023201 0-08-052523-7
Edizione	[2nd ed.]
Descrizione fisica	1 online resource (513 p.)
Collana	The Morgan Kaufmann Series in Computer Architecture and Design
Disciplina	004.165
Soggetti	MIPS (Computer architecture) RISC microprocessors Embedded computer systems - Programming
Lingua di pubblicazione	Inglese
Formato	Materiale a stampa
Livello bibliografico	Monografia
Note generali	Description based upon print version of record.
Nota di bibliografia	Includes bibliographical references (p. 477-479) and index.
Nota di contenuto	Front Cover; See MIPS® Run; Copyright Page; Foreword; Contents; Preface; Style and Limits; Conventions; Acknowledgments; Chapter 1. RISCs and MIPS Architectures; 1.1 Pipelines; 1.2 The MIPS Five-Stage Pipeline; 1.3 RISC and CISC; 1.4 Great MIPS Chips of the Past and Present; 1.5 MIPS Compared with CISC Architectures; Chapter 2. MIPS Architecture; 2.1 A Flavor of MIPS Assembly Language; 2.2 Registers; 2.3 Integer Multiply Unit and Registers; 2.4 Loading and Storing: Addressing Modes; 2.5 Data Types in Memory and Registers; 2.6 Synthesized Instructions in Assembly Language 2.7 MIPS I to MIPS64 ISAs: 64-Bit (and Other) Extensions 2.8 Basic Address Space; 2.9 Pipeline Visibility; Chapter 3. Coprocessor 0: MIPS Processor Control; 3.1 CPU Control Instructions; 3.2 Which Registers Are RelevantWhen?; 3.3 CPU Control Registers and Their Encoding; 3.4 CP0 Hazards-A Trap for the Unwary; Chapter 4. How CachesWork on MIPS Processors; 4.1 Caches and Cache Management; 4.2 How CachesWork; 4.3 Write-Through Caches in Early MIPS CPUs; 4.4 Write-Back Caches in MIPS CPUs; 4.5 Other Choices in Cache Design; 4.6 Managing Caches; 4.7 L2 and L3 Caches 4.8 Cache Configurations for MIPS CPUs 4.9 Programming MIPS32/64 Caches; 4.10 Cache Efficiency; 4.11 Reorganizing Software to Influence

Cache Efficiency; 4.12 Cache Aliases; Chapter 5. Exceptions, Interrupts, and Initialization; 5.1 Precise Exceptions; 5.2 When Exceptions Happen; 5.3 Exception Vectors: Where Exception Handling Starts; 5.4 Exception Handling: Basics; 5.5 Returning from an Exception; 5.6 Nesting Exceptions; 5.7 An Exception Routine; 5.8 Interrupts; 5.9 Starting Up; 5.10 Emulating Instructions; Chapter 6. Low-level Memory Management and the TLB
6.1 The TLB/MMU Hardware and What It Does; 6.2 TLB/MMU Registers Described; 6.3 TLB/MMU Control Instructions; 6.4 Programming the TLB; 6.5 Hardware-Friendly Page Tables and Refill Mechanism; 6.6 Everyday Use of the MIPS TLB; 6.7 Memory Management in a Simpler OS; Chapter 7. Floating-Point Support; 7.1 A Basic Description of Floating Point; 7.2 The IEEE 754 Standard and Its Background; 7.3 How IEEE Floating-Point Numbers Are Stored; 7.4 MIPS Implementation of IEEE 754; 7.5 Floating-Point Registers; 7.6 Floating-Point Exceptions/Interrupts; 7.7 Floating-Point Control: The Control/Status Register
7.8 Floating-Point Implementation Register; 7.9 Guide to FP Instructions; 7.10 Paired-Single Floating-Point Instructions and the MIPS-3D ASE; 7.11 Instruction Timing Requirements; 7.12 Instruction Timing for Speed; 7.13 Initialization and Enabling on Demand; 7.14 Floating-Point Emulation; Chapter 8. Complete Guide to the MIPS Instruction Set; 8.1 A Simple Example; 8.2 Assembly Instructions and What They Mean; 8.3 Floating-Point Instructions; 8.4 Differences in MIPS32/64 Release 1; 8.5 Peculiar Instructions and Their Purposes; 8.6 Instruction Encodings; 8.7 Instructions by Functional Group
Chapter 9. Reading MIPS Assembly Language

Sommario/riassunto

This second edition is not only a thorough update of the first edition, it is also a marriage of the best-known RISC architecture--MIPS--with the best-known open-source OS--Linux. The first part of the book begins with MIPS design principles and then describes the MIPS instruction set and programmers' resources. It uses the MIPS32 standard as a baseline (the 1st edition used the R3000) from which to compare all other versions of the architecture and assumes that MIPS64 is the main option. The second part is a significant change from the first edition. It provides concrete examples of operating
