

Contents

<i>PREFACE</i>	<i>xi</i>
1 SYSTEM CONCEPTS AND MICROPROCESSOR ARCHITECTURE	1.1
1.1 Advances and Microprocessor Performance	1.1
1.1.1 <i>Advances in Processors, 1.2</i>	
1.1.2 <i>Overall System Performance, 1.3</i>	
1.1.3 <i>Processor Performance, 1.5</i>	
1.1.4 <i>Benchmarks, 1.7</i>	
1.1.5 <i>The RISC Approach, 1.8</i>	
1.2 The Processor's External View	1.11
1.2.1 <i>Processor-Bus: Lines and Signals, 1.13</i>	
1.2.2 <i>Processor Interface Components, 1.22</i>	
1.3 Microprocessor-Based System Configurations	1.26
1.3.1 <i>Introduction, 1.26</i>	
1.3.2 <i>System Components, 1.26</i>	
1.3.3 <i>Hierarchy of Buses, 1.31</i>	
1.3.4 <i>Multiprocessor Systems, 1.34</i>	
1.4 Microprocessor Internal Organization	1.35
1.4.1 <i>The Bus Interface Unit, 1.35</i>	
1.4.2 <i>The Instruction Fetch and Decode Unit, 1.36</i>	
1.4.3 <i>The Execution Unit, 1.38</i>	
1.4.4 <i>On-Chip Caches, 1.38</i>	
1.4.5 <i>Example CISC Scalar Processor, 1.39</i>	

1.5	How the Computer Executes Program Instructions	1.40
1.5.1	<i>Instruction Formats,</i>	1.40
1.5.2	<i>Bus Cycles, Clock Cycles, and "States",</i>	1.43
1.5.3	<i>Microoperations and State Transition Diagrams,</i>	1.49
1.6	Pipelined RISC Processors	1.55
1.6.1	<i>Instruction Pipelines,</i>	1.57
1.6.2	<i>Advanced Microprocessor Operation,</i>	1.61
1.6.3	<i>Problems with Pipelining,</i>	1.63
1.7	High-Performance Superscalar Processors	1.66
1.7.1	<i>Superpipelined (Scalar) Processors,</i>	1.66
1.7.2	<i>Superscalar Processors,</i>	1.69
1.7.3	<i>Superpipelined-Superscalar Processors,</i>	1.77
1.7.4	<i>The VLIW Approach,</i>	1.78
1.8	Computer System Design Methodology	1.81
	References	1.84
	Exercises	1.85

2 DATA TRANSFERS ON THE MICROCOMPUTER BUSES 51

2.1	Introduction	51
2.2	Data Ordering and Alignment in Main Memory	52
2.2.1	<i>Big- and Little-Edian Ordering,</i>	52
2.2.2	<i>16- and 32-Bit Processors,</i>	54
2.2.3	<i>RISC Processors with 64-Bit External Data Buses,</i>	61
2.2.4	<i>Loads and Stores in Configurable RISC Processors,</i>	61
2.2.5	<i>Operand and Instruction Alignment,</i>	62
2.3	Aligned Data Bus Transfers	65
2.3.1	<i>Synchronous/Asynchronous Buses,</i>	65
2.3.2	<i>Intel Bus Transfers,</i>	69
2.3.3	<i>Motorola Bus Transfers,</i>	73
2.4	More Advanced Data Bus Transfers	80
2.4.1	<i>Address Pipelining,</i>	80
2.4.2	<i>Burst Transfers,</i>	84

2.5	Transferring over the Bus Misaligned Data	92
2.5.1	<i>Intel Misaligned Transfers</i>	95
2.5.2	<i>Motorola Misaligned Transfers</i>	97
2.6	Microprocessors that Dynamically Adjust their Bus Size	99
2.6.1	<i>Intel 80386</i>	100
2.6.2	<i>Intel 80486</i>	104
2.6.3	<i>Motorola 68020/30</i>	110
2.7	Local Bus Arbitration	116
2.7.1	<i>Processor Arbitration Pins</i>	116
2.7.2	<i>Serial Arbitration Techniques</i>	118
	Exercises	121
	References	124

3 MAIN MEMORY DESIGN AND INTERFACING 125

3.1	Definitions and Types of Memory Chips	125
3.1.1	<i>Access/Cycle Times and Memory Bandwidth</i>	126
3.1.2	<i>Read-Only Memories</i>	128
3.1.3	<i>Static and Dynamic RAMs</i>	129
3.1.4	<i>Memory Organization</i>	137
3.2	Design of the Memory Subsystem	140
3.2.1	<i>Wordlength and Capacity</i>	140
3.2.2	<i>SRAM Memories</i>	144
3.2.3	<i>DRAM Memories</i>	151
3.3	Advanced (DRAM) Memory Access Modes	161
3.3.1	<i>Page Mode</i>	162
3.3.2	<i>Static Column Mode</i>	162
3.3.3	<i>Nibble Mode</i>	166
3.3.4	<i>Comparing the Three Modes</i>	166
3.3.5	<i>Example of 32-Bit Page/Static Column Design</i>	166
3.4	Interleaved Memory Designs	169
3.4.1	<i>Basic Concepts</i>	169
3.4.2	<i>Examples of Designing Interleaved Memories</i>	175
3.4.3	<i>Page-Interleaved Designs</i>	182

3.5	Calculating the Memory Access Time Requirements	185
3.5.1	<i>Matching Processor and Memory Speeds, 185</i>	
3.5.2	<i>Calculating Memory Latency and Memory Access Times, 186</i>	
	Exercises	191
	References	193
4	INDUSTRIAL "STANDARD" SYSTEM BUSES	194
4.1	Introduction	194
4.2	System Buses	194
4.3	Multibus	197
4.3.1	<i>Bus Signals, 197</i>	
4.3.2	<i>Data Transfers on the Multibus I, 201</i>	
4.3.3	<i>Interfacing to the Multibus I, 207</i>	
4.4	VMEbus	207
4.4.1	<i>General Concepts, 207</i>	
4.4.2	<i>VMEbus Signals, 207</i>	
4.4.3	<i>Data Transfers on the VMEbus, 209</i>	
4.4.4	<i>Sequential Access on the VMEbus, 209</i>	
4.4.5	<i>Interfacing to the VMEbus, 212</i>	
4.5	Futurebus	215
4.5.1	<i>Futurebus Signals, 217</i>	
4.5.2	<i>Hierarchical Protocol, 219</i>	
4.5.3	<i>Single-Data-Transfer Cycles, 220</i>	
4.5.4	<i>Block Transfers on the Futurebus, 222</i>	
4.5.5	<i>Broadcasting Mode, 223</i>	
4.6	Mad-Endian and Sad-Endian System Buses	224
4.6.1	<i>Definitions, 224</i>	
4.6.2	<i>Interfacing Processors to Mad- and Sad-Edian System Buses, 225</i>	
4.7	System Bus Arbitration	232
4.7.1	<i>Arbitration Lines, 232</i>	
4.7.2	<i>Bus Arbitration, 234</i>	

4.8 Criteria for Selecting and Evaluating System Buses 236

Exercises 243

References 244

5 CACHES 246

5.1 Introduction 246

5.2 System Configurations and Policies 248

5.2.1 Configurations, 248

5.2.2 Write Policies, 256

5.3 Cache Organizations 258

5.3.1 Fully Associative, 260

5.3.2 Direct Mapped, 265

5.3.3 Set-Associative, 267

5.3.4 General Cache Structure, 268

5.3.5 Cache Line Status Bits, 268

5.4 Cache Update Policies 275

5.4.1 Cache Allocate, 276

5.4.2 LRU Replacement Algorithm, 276

5.4.3 Cache Line Fill Techniques, 276

5.5 Example Processor Caches 279

5.5.1 Fully Associative: Zilog Z80000, 279

5.5.2 Direct Mapped: Motorola 68030, 281

5.5.3 Set Associative: Motorola 68040, 286

5.6 Cache Coherency Issues 289

5.6.1 The Stale Data Problem, 289

5.6.2 Software Solutions, 291

5.6.3 Hardware Solutions, 291

5.7 Cache Design and Performance 289

5.7.1 Parameters That Affect Cache Performance, 299

5.7.2 Measuring Cache Performance, 308

5.7.3 Formulas for ICaches, DCaches, and Unified Caches, 311

Exercises 313
References 314

6. MICROPROCESSOR MAIN MEMORY MANAGEMENT 316

6.1 Introduction 316

6.2 Logical (Virtual) and Physical Address Spaces 319

6.2.1 Main Memory Allocation, 319

6.2.2 Structure of the Logical Address Space, 322

6.3 Mapping the Virtual Address Space 325

6.4 Processors with Paging 328

6.4.1 General Concepts, 328

6.4.2 Page Descriptors, 330

6.4.3 Paged Address Translation, 331

6.4.4 MMUs with ATC or TLB Caches, 331

6.4.5 Table-Walkthrough or Table-Search Mechanism, 334

6.4.6 Page Descriptor Formats and ATC/TLB Structures, 340

6.4.7 External Paged MMU Chips, 343

6.4.8 Concluding Remarks on Paging, 346

6.5 Processors with Segmentation 348

6.5.1 General Concepts, 348

6.5.2 Segmentation in the 8086, 349

6.5.3 Intel On-Chip Segmentation Registers, 353

6.5.4 Intel Segment Descriptor Formats, 354

6.5.5 Intel Segmented Address Translation, 357

6.5.6 External Segmented MMU Chips, 357

6.5.7 Concluding Remarks on Segmentation, 362

6.6 Combined Segmentation and Paging 363

6.7 Comparing with Segmentation 365

Exercises 366
References 367

7	INTERRUPTS, MULTITASKING AND PROTECTION	369
7.1	Introduction	369
7.2	Protection Mechanisms	370
7.2.1	<i>User, Supervisor, and Protected Modes, 370</i>	
7.2.2	<i>Intel Protected Mode, 371</i>	
7.3	External Interrupts	374
7.3.1	<i>Interrupt Lines, 374</i>	
7.3.2	<i>Vectored Interrupts, 377</i>	
7.3.3	<i>Interrupt Latency, 391</i>	
7.4	Exception Processing	393
7.4.1	<i>Intel Protected-Mode Exceptions, 393</i>	
7.4.2	<i>Motorola Exception Processing, 397</i>	
7.5	Pipelined Processors	403
7.5.1	<i>Survey of Processor Pipelining, 403</i>	
7.5.2	<i>Pipelining Problems, 411</i>	
7.6	Making a Segment "Known"	413
7.6.1	<i>Translating Segment Names to Physical Base Addresses, 413</i>	
7.6.2	<i>Making a Segment "Known, 414</i>	
	Exercises	416
	References	418

Appendices

A EXAMPLE PROCESSORS

A.1	A 16-BIT CISC PROCESSOR: THE INTEL 8086	A.1
a)	<i>External View, A.1</i>	
b)	<i>8086 Internal Organization, A.2</i>	
c)	<i>8086 Address Conversion Mechanism, A.4</i>	
d)	<i>8086 State Transition Diagram, A.8</i>	
e)	<i>8086 Basic Timing Diagram, A.8</i>	

A.2	A 32-BIT CISC PROCESSOR: THE INTEL 486	A.11
	<i>a) 486 I/O Pins and Signals,</i>	<i>A.11</i>
	<i>b) 486 Internal Organization and Register Structure,</i>	<i>A.12</i>
	<i>c) 486 State Transition Diagram,</i>	<i>A.16</i>
A.3	A SUPERPIPELINED (SCALAR) RISC PROCESSOR: THE MIPS R4000	A.18
A.4	SUPERSCALAR PROCESSORS	A.23
	A.4.1 The 32-bit CISC/RISC Intel Pentium,	A.23
	<i>a) Pentium System Interfacing,</i>	<i>A.22</i>
	<i>b) Pentium Internal Organization,</i>	<i>A.27</i>
	<i>c) Pentium State Transition Diagram,</i>	<i>A.28</i>
	<i>d) Pentium Basic Bus Timing Diagram,</i>	<i>A.28</i>
	A.4.2 The 64-bit Superscalar MIPS R10000,	A.30
A.5	SUPERPIPELINED-SUPERSCALAR PROCESSORS	A.33
	A.5.1 Intel Pentium Pro,	A.33
	A.5.2 DEC Alpha 21164,	A.35

B MOTOROLA 680x0 CISC PROCESSORS

B.1	MOTOROLA 16-BIT 68000 MICROPROCESSOR,	B-1
	<i>B.1.1 68000 I/O Pins and Signals,</i>	<i>B-1</i>
	<i>B.1.2 68000 State Transition and Bus Timing Diagrams,</i>	<i>B-3</i>
B.2	MOTOROLA 32-BIT 680X0 MICROPROCESSORS,	B-6
	<i>B.2.1 680x0 I/O Pins and Signals,</i>	<i>B-7</i>
	<i>B.2.2 680x0 Internal Organization and Register Structure,</i>	<i>B-7</i>
	<i>B.2.3 680x0 State Transition and Timing Diagrams,</i>	<i>B-15</i>
	<i>B.2.4 68020/30 Asynchronous Data Transfers,</i>	<i>B-17</i>
	<i>B.2.5 68020/30 Synchronous Data Transfers,</i>	<i>B-17</i>
	<i>B.2.6 68030/40 Descriptor Formats,</i>	<i>B-17</i>

Preface

This textbook presents processor-based system design concepts and illustrates them through specific examples, using current technology processors and other very large scale integration (VLSI) components. It is intended for more advanced undergraduate or beginning graduate students in electrical and computer engineering, computer science, and other related fields. The reader is assumed to have had a course in the design of digital systems (or a first course in microprocessors) and basic knowledge of computer organization and architecture.

The single-chip processors are becoming the implementation technology of choice for an increasingly wide range of computer systems. Rather than giving a quantitative approach and a block-diagram-based view of their architecture, this textbook is more specific and gives a more hardware-oriented approach to the design of computer systems. Its purpose is to present ways of building computer systems using commercial processors; it is not about designing the processor chips themselves. (The system designer/integrator is much more interested in the speed of the CPU chip rather than on the detailed implementation of its internal hardware.) However, issues that have to do with the design of the processor itself impact significantly the performance of the overall system as well as how one goes about designing it; therefore, such related internal architectural schemes (e.g., pipelining, on-chip caching, etc.) are presented mainly from the point of view of how they effect system performance. For example, capabilities not provided on-chip will have to be implemented with external chips; the type of external signals and buses of the processor impact the expense and difficulty of interfacing it to memory and to input/output; the particular processor chip chosen for the design and the particular system bus chosen to interconnect the various boards of the computer system will influence the interface complexity of each board; other capabilities, such as paging, segmentation, or handling of the internal and external exceptions have similarly significant impact on the design of both the hardware system and the systems software.

This textbook presents a survey of the structure and capabilities of advanced processors (both CISC- and RISC-type processors are presented in a “unified

approach”), and a large number of design examples is given throughout the textbook using commercial products. An attempt is made throughout to maintain the systems-level design approach (i.e., to look at the whole picture of a computer system, made up of basic building blocks that represent complex hardware subsystems, such as processors, buses, memory chips, I/O interface modules, and specialized chips such as MMUs and caches). The text emphasizes that computer systems are structured either as a number of chips or as a number of boards interconnected through a hierarchy of several buses (processor bus, local bus, system bus, etc.). The bus is treated here as a basic component of the computer system, which permits alternative system structures and bus-based computer designs. The reader is considered to be either the designer of boards (processor boards, memory boards, interface boards, etc.) or the board-level computer system designer/integrator. When it comes to software design issues, the reader is considered to be the “systems software designer” who requires knowledge of and access to “all system resources” (i.e., system registers, status bits, and I/O ports).

In designing this textbook, I tried to observe the following guidelines:

- a. Instead of directly jumping into the details of how a particular manufacturer has implemented a feature in its processors, explain the issues and trade-offs involved in choosing among different design approaches.*
- b. Present the various design issues in a unified and generic manner, which would be applicable to all processors irrespective of their particular manufacturer, type (RISC or CISC), word length, and level of integration and then give specific examples of actual implementations.*
- c. Expose the reader to alternative ways of designing computer systems, to the trade-offs involved in putting things together or choosing system components, and to the advantages and disadvantages of the various decisions, rather than to a description of the particular method a manufacturer followed in building a system.*
- d. Cover a large number of representative processors so that the reader is exposed to various architectural features of processors and how they impact program execution and overall system performance, along with different characteristics and capabilities that impact the way a computer system is designed. For example, features and capabilities that vary significantly from one processor to another include: RISC versus CISC processors; Harvard architecture with dual external buses; MMU, cache, and floating-point capabilities; on-chip and off-chip implementation of features; processors with instruction and/or data pipelining; reconfigurable processors with advanced data bus transfers; multitasking and virtual memory capabilities based on paging, segmentation, or both; and various control functions that are an integral part of the computer system design, such as the protected mode in the Intel processors, interrupt-driven task switching, and exception handling.*

In each chapter, those generic ideas and organizational concepts that exist independent of a particular implementation are presented first, followed by extensive surveys of how specific commercial products have embodied a concept. Although basic principles are presented from a generic point of view to be applicable to the design of computer systems using a variety of products, descriptions of a large number of specific products and concrete design examples using current components are given throughout the text and are also reflected in the exercises at the end of each chapter to make the text more specific. Important issues and criteria for computer selection, as well as alternative hardware and system design choices and trade-offs are also included in each chapter.

***Chapter 1** starts with a discussion on the advances in processors and gives an overview of their internal architecture, external I/O signals, and their operation; these are presented in a simplified, generic form here, while details for representative processors are given in the Appendices. Both RISC and CISC processors are discussed and their major differences pointed out, along with other components (such as memory, caches, special coprocessors, and buses) needed to put together a computer system. The chapter ends with an introduction to pipelined, superpipelined, and superscalar processors, and a discussion of computer system design methodology.*

***Chapter 2** concentrates on the external (local) bus of the processor CPU and how the processor uses it to perform synchronous and asynchronous data exchanges with other units of the computer system. The chapter starts with a discussion on big- and little-endian ordering of operands in memory and how recent configurable RISC processors handle them dynamically. In addition to the bus timing diagrams for the execution of aligned data transfers on 32- and 64-bit data buses, this chapter presents other advanced data transfer modes, such as “address pipelining,” “burst transfers,” and misaligned data transfers. It also explains how the recent processors implement “dynamic bus sizing,” which allows them to adjust the width of their local data bus to that of the data port they communicate with. The chapter ends with a discussion of the local bus arbitration techniques for performing DMA operations.*

***Chapter 3** covers the design of the memory subsystem and its interface to the processor CPU. It starts with some basic terms and definitions and presents the various types and characteristics of memory chips. It then gives the details of designing and interfacing memory subsystems of different wordlengths, using various types of static and dynamic RAMs. The more advanced DRAM access modes (page, static column, nibble) are also discussed and an extensive treatment of interleaved memory along with design examples is given. Finally, the chapter presents a quantitative approach to calculating the memory latency time and the memory access time requirements in order to match the processor and memory*

bandwidths.

Chapter 4 reviews the industry system buses used to interconnect boards from different manufacturers and with different wordlengths in building more powerful computer systems. The three system buses used as examples are the Multibus, the VMEbus, and the Futurebus. Their signals and timing are presented, the way they perform data transfers is explained, and examples are given for interfacing them to processor and memory boards. A section is also devoted to the mad- and sad- endian system buses and how different types of processors are interfaced to them. Both serial and parallel system bus arbitration techniques are presented and the chapter ends with a discussion of trade-offs involved in selecting the system bus.

Chapter 5 covers caches. It first discusses system issues, such as different ways of configuring external caches, multilevel caches, and different cache write policies. It then presents the three organizations (fully associative, direct mapped, and set-associative) and the cache line format, and gives examples of specific processor on-chip and off-chip cache implementations. An extensive part of the chapter is devoted to the cache coherency problem, the issues involved, and common software and hardware solutions. The chapter ends with a discussion of the parameters that affect cache design and gives approximation formulas for measuring cache performance and the cache's impact on the performance of the overall system.

Chapter 6 discusses the design and use of MMUs (Memory Management Units) in advanced multitasking; virtual memory computer systems. It presents the structure of the logical address space and ways of mapping it into the main memory space. It gives examples of how various processors implement paging, segmentation, or a combination of both, and discusses the on-chip and off-chip MMU hardware involved.

Chapter 7 contains various relevant topics, which are treated to a limited extent. The scope of this text does not permit devoting a separate chapter to each one. These topics include: protection mechanisms and task switching in various computer systems, how processors handle external interrupting devices, advanced exception processing in protected mode, a survey of pipelined processors and a discussion of the pipelining problems, and how segments become "known."

Finally, the **Appendices** survey a number of commercial processors and give their internal organization and a detailed explanation of their I/O pins and signals.

I would like to express my appreciation to a number of colleagues for their valuable criticism and comments during the early stages of the manuscript. In addition, I would like to thank the reviewers: Ken Breeding, Ohio State University; Karan Watson, Texas A&M University; Michael Chwialkowski, University of Texas, Arlington; Everett Johnson, Wichita State University; William Ohley, University of Rhode Island .

Last, but not least, I would like to express my gratitude to my wife and daughter for their support, patience and understanding; their continuous encouragement helped me tremendously in going through the various revisions and rewriting of this text.

Every attempt was made to cover the major issues in designing computer systems using recent advanced processors, to use a rather large sample of representative RISC and CISC products, and to present design decisions and trade-offs. A notable number of important and powerful processors were not covered in this text because of space limitations. I would be greatly interested in receiving your comments and criticism, suggestions for improvements to the text, corrections, or contributed exercises. I can be reached at: Dept. of Electrical Engineering and Computer Science, The George Washington University, Washington, DC 20052; tel: 202-994-0523, fax: 202-994-0227, email: alexan@seas.gwu.edu.

*Nikitas A. Alexandridis
Washington, DC*