Computer Architecture: A Quantitative Approach, Second Edition by John L. Hennessy and David A. Patterson

Foreword

Preface

Acknowledgments

1 Fundamentals of Computer Design

1.1 Introduction

- 1.2 The Task of a Computer Designer
- 1.3 Technology and Computer Usage Trends
- 1.4 Cost and Trends in Cost

1.5 Measuring and Reporting Performance

1.6 Quantitative Principles of Computer Design

1.7 Putting It All Together: The Concept of Memory Hierarchy

1.8 Fallacies and Pitfalls

1.9 Concluding Remarks

1.10 Historical Perspective and References Exercises

2 Instruction Set Principles and Examples

2.1 Introduction

2.2 Classifying Instruction Set Architectures

2.3 Memory Addressing

2.4 Operations in the Instruction Set

2.5 Type and Size of Operands

2.6 Encoding and Instruction Set

2.7 Crosscutting Issues: The Role of Compilers

2.8 Putting It All together: The DLX Architecture

2.9 Fallacies and Pitfalls

2.10 Concluding Remarks

Historical Perspective and References

Exercises

3 Pipelining

- 3.1 What is Pipelining?
- 3.2 The Basic Pipeline for DLX
- 3.3 The Major Hurdle of Pipelining-Pipeline Hazards

3.4 Data Hazards

3.5 Control Hazards

3.6 What Makes Pipelining Hard to Implement?

3.7 Extending the DLX Pipeline to Handle Multicycle Operations

3.8 Crosscutting Issues: Instruction Set Design and Pipelining

3.9 Putting It All Together: The MIPS R4000 Pipeline

3.10 Fallacies and Pitfalls

3.11 Concluding Remarks

3.12 Historical Perspective and References

Exercises

4 Advanced Pipelining and Instruction-Level Parallelism

4.1 Instruction-Level Parallelism: Concepts and Challenges

4.2 Overcoming Data Hazards with Dynamic Scheduling

4.3 Reducing Branch Penalties with Dynamic Hardware Prediction

4.4 Taking Advantage of More ILP with Multiple Issue

4.5 Compiler Support for Exploiting ILP

4.6 Hardware Support for Extracting More Parallelism

4.7 Studies of ILP

4.8 Putting It All Together: The PowerPC 620

4.9 Fallacies and Pitfalls

4.10 Concluding Remarks

4. 11 Historical Perspective and References

Exercises

5 Memory-Hierarchy Design

5.1 Introduction

5.2 The ABCs of Caches

5.3 Reducing Cache Misses

5.4 Reducing Cache Miss Penalty

5.5 Reducing Hit Time

5.6 Main Memory

5.7 Virtual Memory

5.8 Protection and Examples of Virtual Memory

5.9 Crosscutting Issues in the Design of Memory Hierarchies

5.10 Putting It All Together: The PowerPC 620

5.11 Fallacies and Pitfalls

5.12 Concluding Remarks

5.13 Historical Perspective and References

Exercises

6 Storage Systems

6.1 Introduction

6.2 Types of Storage Devices

6.3 Buses-Connecting I/O Devices to CPU/Memory

6.4 I/O Performance Measures

6.5 Reliability, Availability, and RAID

6.6 Crosscutting Issues: Interfacing to an Operating System

6.7 Designing an I/O System

6.8 Putting It All Together: UNIX File System Performance

6.9 Fallacies and Pitfalls

6.10 Concluding Remarks

6.11 Historical Perspective and References

Exercises

7 Interconnectoin Networks

7.1 Introduction

7.2 A Simple Network

7.3 Connecting the Interconnection Network to the Computer

7.4 Interconnection network Media

7.5 Connecting More Than Two Computers

7.6 Practical Issues for Commercial Interconnection Networks

7.7 Examples of Interconnection Networks

7.9 Internetworking

7.10 Putting it All Together: An ATM Network of Workstations

7.11 Fallacies and Pitfalls

7.12 Concluding Remarks

7.13 Historical Perspective and References

Exercises

8 Multiprocessors

8.1 Introduction

8.2 Characteristics of Application Domains

8.3 Centralized Shared-Memory Architectures

8.4 Distributed Shared-Memory Architectures

8.5 Synchronization

8.6 Models of Memory Consistency

8.7 Crosscutting Issues

8.8 Putting It All Together: The SGI Challenge Multiprocessor

8.9 Fallacies and Pitfalls

8.10 Concluding Remarks

8.11 Historical Perspective and References

Exercises

Appendix A: Computer Arithmetic by David Goldberg Xerox Palo Alto Research Center

A.1 Introduction

A.2 Basic Techniques of Integer Arithmetic

A.3 Floating Point

A.4 Floating Point Multiplication

A.5 Floating Point-Addition

A.6 Division and Remainder

A.7 More on Floating-Point Arithmetic

A.8 Speeding Up integer Addition

A.9 Speeding Up integer Multiplication and Division

A.10 Putting It All together

A.11 Fallacies and Pitfalls

A.12 Historical Perspective and References

Exercises

Appendix B: Vector Processors

B.1 Why Vector Processors?
B.2 Basic Vector Architecture
B.3 Two Real World Issues: Vector Length and Stride
B.4 Effectiveness of Compiler Vectorization
B.5 Enhancing Vector Performance
B.6 Putting It All Together: Performance of Vector Processors
B.7 Fallacies and Pitfalls
B.8 Concluding Remarks
B.9 Historical Perspective and References
Exercises

Appendix C: Survey of RISC Architectures

C.1 Introduction C.2 Addressing Modes and Instruction Formats C.3 Instructions: The DLX Subset C.4 Instructions: Common Extensions to DLX C.5 Instructions Unique to MIPS C.6 Instructions Unique to SPARC C.7 Instructions Unique to PowerPC C.8 Instructions Unique to PA-RISC C.9 Concluding Remarks C.10 References

Appendix D: An Alternative to RISC: the Intel 80x86

D.1 Introduction

D.2 80x86 Registers and Data Addressing Modes

D.3 80x86 Integer Operations

D.4 80x86 Floating Point Operations

D.5 80x86 Instruction Encoding

D.6 Putting It All together: Measurements of Instruction Set Usage

D.7 Concluding Remarks

D.8 Historical Perspective and References

Appendix E: Implementing Coherence Protocols

E.1 Implementation Issues for the Snooping Coherence Protocol E.2 Implementation Issues in the Distributed Directory Protocol References Index