

INTERNATIONAL
EDITION



Structured Computer Organization

SIXTH EDITION

Andrew S. Tanenbaum

Todd Austin

ALWAYS LEARNING

PEARSON

STRUCTURED COMPUTER ORGANIZATION

SIXTH EDITION

Structured Computer Organization

Table of Contents

Cover

CONTENTS

PREFACE

1 INTRODUCTION

1.1 STRUCTURED COMPUTER ORGANIZATION

1.1.1 Languages, Levels, and Virtual Machines

1.1.2 Contemporary Multilevel Machines

1.1.3 Evolution of Multilevel Machines

1.2 MILESTONES IN COMPUTER ARCHITECTURE

1.2.1 The Zeroth Generation Mechanical Computers (1642-1945)

1.2.2 The First Generation Vacuum Tubes (1945-1955)

1.2.3 The Second Generation Transistors (1955-1965)

1.2.4 The Third Generation Integrated Circuits (1965-1980)

1.2.5 The Fourth Generation Very Large Scale Integration (1980?)

1.2.6 The Fifth Generation Low-Power and Invisible Computers

1.3 THE COMPUTER ZOO

1.3.1 Technological and Economic Forces

1.3.2 The Computer Spectrum

1.3.3 Disposable Computers

1.3.4 Microcontrollers

1.3.5 Mobile and Game Computers

1.3.6 Personal Computers

1.3.7 Servers

1.3.8 Mainframes

Table of Contents

1.4 EXAMPLE COMPUTER FAMILIES

- 1.4.1 Introduction to the x86 Architecture
- 1.4.2 Introduction to the ARM Architecture
- 1.4.3 Introduction to the AVR Architecture

1.5 METRIC UNITS

1.6 OUTLINE OF THIS BOOK

2 COMPUTER SYSTEMS

2.1 PROCESSORS

- 2.1.1 CPU Organization
- 2.1.2 Instruction Execution
- 2.1.3 RISC versus CISC
- 2.1.4 Design Principles for Modern Computers
- 2.1.5 Instruction-Level Parallelism
- 2.1.6 Processor-Level Parallelism

2.2 PRIMARYMEMORY

- 2.2.1 Bits
- 2.2.2 Memory Addresses
- 2.2.3 Byte Ordering
- 2.2.4 Error-Correcting Codes
- 2.2.5 Cache Memory
- 2.2.6 Memory Packaging and Types

2.3 SECONDARYMEMORY

- 2.3.1 Memory Hierarchies
- 2.3.2 Magnetic Disks
- 2.3.3 IDE Disks
- 2.3.4 SCSI Disks
- 2.3.5 RAID
- 2.3.6 Solid-State Disks

Table of Contents

2.3.7 CD-ROMs

2.3.8 CD-Recordables

2.3.9 CD-Rewritables

2.3.10 DVD

2.3.11 Blu-ray

2.4 INPUT/OUTPUT

2.4.1 Buses

2.4.2 Terminals

2.4.3 Mice

2.4.4 Game Controllers

2.4.5 Printers

2.4.6 Telecommunications Equipment

2.4.7 Digital Cameras

2.4.8 Character Codes

2.5 SUMMARY

3 THE DIGITAL LOGIC LEVEL

3.1 GATES AND BOOLEAN ALGEBRA

3.1.1 Gates

3.1.2 Boolean Algebra

3.1.3 Implementation of Boolean Functions

3.1.4 Circuit Equivalence

3.2 BASIC DIGITAL LOGIC CIRCUITS

3.2.1 Integrated Circuits

3.2.2 Combinational Circuits

3.2.3 Arithmetic Circuits

3.2.4 Clocks

3.3 MEMORY

3.3.1 Latches

Table of Contents

3.3.2 Flip-Flops

3.3.3 Registers

3.3.4 Memory Organization

3.3.5 Memory Chips

3.3.6 RAMs and ROMs

3.4 CPU CHIPS AND BUSES

3.4.1 CPU Chips

3.4.2 Computer Buses

3.4.3 Bus Width

3.4.4 Bus Clocking

3.4.5 Bus Arbitration

3.4.6 Bus Operations

3.5 EXAMPLE CPU CHIPS

3.5.1 The Intel Core i7

3.5.2 The Texas Instruments OMAP4430 System-on-a-Chip

3.5.3 The Atmel ATmega168 Microcontroller

3.6 EXAMPLE BUSES

3.6.1 The PCI Bus

3.6.2 PCI Express

3.6.3 The Universal Serial Bus

3.7 INTERFACING

3.7.1 I/O Interfaces

3.7.2 Address Decoding

3.8 SUMMARY

4 THE MICROARCHITECTURE LEVEL

4.1 AN EXAMPLE MICROARCHITECTURE

4.1.1 The Data Path

4.1.2 Microinstructions

4.1.3 Microinstruction Control: The Mic-1

Table of Contents

4.2 AN EXAMPLE ISA: IJVM

4.2.1 Stacks

4.2.2 The IJVM Memory Model

4.2.3 The IJVM Instruction Set

4.2.4 Compiling Java to IJVM

4.3 AN EXAMPLE IMPLEMENTATION

4.3.1 Microinstructions and Notation

4.3.2 Implementation of IJVM Using the Mic-1

4.4 DESIGN OF THE MICROARCHITECTURE LEVEL

4.4.1 Speed versus Cost

4.4.2 Reducing the Execution Path Length

4.4.3 A Design with Prefetching: The Mic-2

4.4.4 A Pipelined Design: The Mic-3

4.4.5 A Seven-Stage Pipeline: The Mic-4

4.5 IMPROVING PERFORMANCE

4.5.1 Cache Memory

4.5.2 Branch Prediction

4.5.3 Out-of-Order Execution and Register Renaming

4.5.4 Speculative Execution

4.6 EXAMPLES OF THE MICROARCHITECTURE LEVEL

4.6.1 The Microarchitecture of the Core i7 CPU

4.6.2 The Microarchitecture of the OMAP4430 CPU

4.6.3 The Microarchitecture of the ATmega168 Microcontroller

4.7 COMPARISON OF THE I7, OMAP4430, AND ATMEGA168

4.8 SUMMARY

5 THE INSTRUCTION SET

5.1 OVERVIEW OF THE ISA LEVEL

5.1.1 Properties of the ISA Level

Table of Contents

5.1.2 Memory Models

5.1.3 Registers

5.1.4 Instructions

5.1.5 Overview of the Core i7 ISA Level

5.1.6 Overview of the OMAP4430 ARM ISA Level

5.1.7 Overview of the ATmega168 AVR ISA Level

5.2 DATA TYPES

5.2.1 Numeric Data Types

5.2.2 Nonnumeric Data Types

5.2.3 Data Types on the Core i7

5.2.4 Data Types on the OMAP4430 ARM CPU

5.2.5 Data Types on the ATmega168 AVR CPU

5.3 INSTRUCTION FORMATS

5.3.1 Design Criteria for Instruction Formats

5.3.2 Expanding Opcodes

5.3.3 The Core i7 Instruction Formats

5.3.4 The OMAP4430 ARM CPU Instruction Formats

5.3.5 The ATmega168 AVR Instruction Formats

5.4 ADDRESSING

5.4.1 Addressing Modes

5.4.2 Immediate Addressing

5.4.3 Direct Addressing

5.4.4 Register Addressing

5.4.5 Register Indirect Addressing

5.4.6 Indexed Addressing

5.4.7 Based-Indexed Addressing

5.4.8 Stack Addressing

5.4.9 Addressing Modes for Branch Instructions

5.4.10 Orthogonality of Opcodes and Addressing Modes

Table of Contents

- 5.4.11 The Core i7 Addressing Modes
- 5.4.12 The OMAP4440 ARM CPU Addressing Modes
- 5.4.13 The ATmega168 AVR Addressing Modes
- 5.4.14 Discussion of Addressing Modes

5.5 INSTRUCTION TYPES

- 5.5.1 Data Movement Instructions
- 5.5.2 Dyadic Operations
- 5.5.3 Monadic Operations
- 5.5.4 Comparisons and Conditional Branches
- 5.5.5 Procedure Call Instructions
- 5.5.6 Loop Control
- 5.5.7 Input/Output
- 5.5.8 The Core i7 Instructions
- 5.5.9 The OMAP4430 ARM CPU Instructions
- 5.5.10 The ATmega168 AVR Instructions
- 5.5.11 Comparison of Instruction Sets

5.6 FLOW OF CONTROL

- 5.6.1 Sequential Flow of Control and Branches
- 5.6.2 Procedures
- 5.6.3 Coroutines
- 5.6.4 Traps
- 5.6.5 Interrupts

5.7 A DETAILED EXAMPLE: THE TOWERS OF HANOI

- 5.7.1 The Towers of Hanoi in Core i7 Assembly Language
- 5.7.2 The Towers of Hanoi in OMAP4430 ARM Assembly Language

5.8 THE IA-64 ARCHITECTURE AND THE ITANIUM 2

- 5.8.1 The Problem with the IA-32 ISA
- 5.8.2 The IA-64 Model: Explicitly Parallel Instruction Computing
- 5.8.3 Reducing Memory References

Table of Contents

5.8.4 Instruction Scheduling

5.8.5 Reducing Conditional Branches: Predication

5.8.6 Speculative Loads

5.9 SUMMARY

6 THE OPERATING SYSTEM

6.1 VIRTUAL MEMORY

6.1.1 Paging

6.1.2 Implementation of Paging

6.1.3 Demand Paging and the Working-Set Model

6.1.4 Page-Replacement Policy

6.1.5 Page Size and Fragmentation

6.1.6 Segmentation

6.1.7 Implementation of Segmentation

6.1.8 Virtual Memory on the Core i7

6.1.9 Virtual Memory on the OMAP4430 ARM CPU

6.1.10 Virtual Memory and Caching

6.2 HARDWARE VIRTUALIZATION

6.2.1 Hardware Virtualization on the Core i7

6.3 OSM-LEVEL I/O INSTRUCTIONS

6.3.1 Files

6.3.2 Implementation of OSM-Level I/O Instructions

6.3.3 Directory Management Instructions

6.4 OSM-LEVEL INSTRUCTIONS FOR PARALLEL PROCESSING

6.4.1 Process Creation

6.4.2 Race Conditions

6.4.3 Process Synchronization Using Semaphores

6.5 EXAMPLE OPERATING SYSTEMS

6.5.1 Introduction

Table of Contents

- 6.5.2 Examples of Virtual Memory
- 6.5.3 Examples of OS-Level I/O
- 6.5.4 Examples of Process Management

6.6 SUMMARY

7 THE ASSEMBLY LANGUAGE LEVEL

7.1 INTRODUCTION TO ASSEMBLY LANGUAGE

- 7.1.1 What Is an Assembly Language?
- 7.1.2 Why Use Assembly Language?
- 7.1.3 Format of an Assembly Language Statement
- 7.1.4 Pseudoinstructions

7.2 MACROS

- 7.2.1 Macro Definition, Call, and Expansion
- 7.2.2 Macros with Parameters
- 7.2.3 Advanced Features
- 7.2.4 Implementation of a Macro Facility in an Assembler

7.3 THE ASSEMBLY PROCESS

- 7.3.1 Two-Pass Assemblers
- 7.3.2 Pass One
- 7.3.3 Pass Two
- 7.3.4 The Symbol Table

7.4 LINKING AND LOADING

- 7.4.1 Tasks Performed by the Linker
- 7.4.2 Structure of an Object Module
- 7.4.3 Binding Time and Dynamic Relocation
- 7.4.4 Dynamic Linking

7.5 SUMMARY

8 PARALLEL COMPUTER ARCHITECTURES

8.1 ON-CHIP PARALELLISM

Table of Contents

8.1.1 Instruction-Level Parallelism

8.1.2 On-Chip Multithreading

8.1.3 Single-Chip Multiprocessors

8.2 COPROCESSORS

8.2.1 Network Processors

8.2.2 Graphics Processors

8.2.3 Cryptoprocessors

8.3 SHARED-MEMORY MULTIPROCESSORS

8.3.1 Multiprocessors vs. Multicomputers

8.3.2 Memory Semantics

8.3.3 UMA Symmetric Multiprocessor Architectures

8.3.4 NUMA Multiprocessors

8.3.5 COMA Multiprocessors

8.4 MESSAGE-PASSING MULTICOMPUTERS

8.4.1 Interconnection Networks

8.4.2 MPPs Massively Parallel Processors

8.4.3 Cluster Computing

8.4.4 Communication Software for Multicomputers

8.4.5 Scheduling

8.4.6 Application-Level Shared Memory

8.4.7 Performance

8.5 GRID COMPUTING

8.6 SUMMARY

9 BIBLIOGRAPHY

A: BINARY NUMBERS

A.1 FINITE-PRECISION NUMBERS

A.2 RADIX NUMBER SYSTEMS

A.3 CONVERSION FROM ONE RADIX TO ANOTHER

Table of Contents

A.4 NEGATIVE BINARY NUMBERS

A.5 BINARY ARITHMETIC

B: FLOATING-POINT NUMBERS

B.1 PRINCIPLES OF FLOATING POINT

B.2 IEEE FLOATING-POINT STANDARD 754

C: ASSEMBLY LANGUAGE PROGRAMMING

C.1 OVERVIEW

C.1.1 Assembly Language

C.1.2 A Small Assembly Language Program

C.2 THE 8088 PROCESSOR

C.2.1 The Processor Cycle

C.2.2 The General Registers

C.2.3 Pointer Registers

C.3 MEMORY AND ADDRESSING

C.3.1 Memory Organization and Segments

C.3.2 Addressing

C.4 THE 8088 INSTRUCTION SET

C.4.1 Move, Copy and Arithmetic

C.4.2 Logical, Bit and Shift Operations

C.4.3 Loop and Repetitive String Operations

C.4.4 Jump and Call Instructions

C.4.5 Subroutine Calls

C.4.6 System Calls and System Subroutines

C.4.7 Final Remarks on the Instruction Set

C.5 THE ASSEMBLER

C.5.1 Introduction

C.5.2 The ACK-Based Tutorial Assembler as88

C.5.3 Some Differences with Other 8088 Assemblers

Table of Contents

C.6 THE TRACER

C.6.1 Tracer Commands

C.7 GETTING STARTED

C.8 EXAMPLES

C.8.1 Hello World Example

C.8.2 General Registers Example

C.8.3 Call Command and Pointer Registers

C.8.4 Debugging an Array Print Program

C.8.5 String Manipulation and String Instructions

C.8.6 Dispatch Tables

C.8.7 Buffered and Random File Access

INDEX