

Structured Computer Organization



[Structured Computer Organization_ 下载链接1](#)

著者:Andrew S. Tanenbaum

出版者:Prentice Hall

出版时间:2012-8-4

装帧:Hardcover

isbn:9780132916523

Structured Computer Organization, specifically written for undergraduate students, is a best-selling guide that provides an accessible introduction to computer hardware and architecture. This text will also serve as a useful resource for all computer professionals and engineers who need an overview or introduction to computer architecture. This book takes a modern structured, layered approach to understanding computer systems. It's highly accessible - and it's been thoroughly updated to reflect today's most critical new technologies and the latest developments in computer organization and architecture. Tanenbaum's renowned writing style and painstaking research make this one of the most accessible and accurate books available, maintaining the author's popular method of presenting a computer as a series of layers, each one built upon the ones below it, and understandable as a separate entity.

作者介绍:

Andrew S. Tanenbaum has a B.S. Degree from M.I.T. and a Ph.D. from the University of California at Berkeley. He is currently a Professor of Computer Science at the Vrije Universiteit in Amsterdam, The Netherlands, where he heads the Computer Systems

Group. Until 2005, he was the Dean of the Advanced School for Computing and Imaging, an inter-university graduate school doing research on advanced parallel, distributed, and imaging systems.

In the past, he has done research on compilers, operating systems, networking, and local-area distributed systems. His current research focuses primarily on the design of wide-area distributed systems that scale to a billion users. These research projects have led to five books and over 85 referred papers in journals and conference proceedings.

Prof. Tanenbaum has also produced a considerable volume of software. He was the principal architect of the Amsterdam Compiler Kit, a widely-used toolkit for writing portable compilers, as well as of MINIX, a small UNIX clone intended for use in student programming labs. Together with his Ph.D. students and programmers, he helped design the Amoeba distributed operating system, a high-performance microkernel-based distributed operating system. The MINIX and Amoeba systems are now available for free via the Internet..

Prof. Tanenbaum is a Fellow of the ACM, a Fellow of the IEEE, a member of the Royal Netherlands Academy of Arts and Sciences, winner of the 1994 ACM Karl V. Karlstrom Outstanding Educator Award, and winner of the 1997 ACM/SIGCSE Award for Outstanding Contributions to Computer Science Education. He is also listed in Who's Who in the World.

Todd Austin is a Professor of Electrical Engineering and Computer Science at the University of Michigan in Ann Arbor. His research interests include computer architecture, reliable system design, hardware and software verification, and performance analysis tools and techniques. Prior to joining academia, Todd was a Senior Computer Architect in Intel's Microcomputer Research Labs, a product-oriented research laboratory in Hillsboro, Oregon. Todd is the first to take credit (but the last to accept blame) for creating the SimpleScalar Tool Set, a popular collection of computer architecture performance analysis tools. In addition to his work in academia, Todd is co-founder of SimpleScalar LLC and InTempo Design LLC. In 2002, Todd was a Sloan Research Fellow, and in 2007 he received the ACM Maurice Wilkes Award for "for innovative contributions in Computer Architecture including the SimpleScalar Toolkit and the DIVA and Razor architectures." Todd received his Ph.D. in Computer Science from the University of Wisconsin in 1996.

目录:	1.1 STRUCTURED COMPUTER ORGANIZATION	2
1.1.1	Languages, Levels, and Virtual Machines	2
1.1.2	Contemporary Multilevel Machines	5
1.1.3	Evolution of Multilevel Machines	8
1.2	MILESTONES IN COMPUTER ARCHITECTURE	13
1.2.1	The Zeroth Generation—Mechanical Computers (1642—1945)	13
1.2.2	The First Generation—Vacuum Tubes (1945—1955)	16
1.2.3	The Second Generation—Transistors (1955—1965)	19
1.2.4	The Third Generation—Integrated Circuits (1965—1980)	21
1.2.5	The Fourth Generation—Very Large Scale Integration (1980—?)	23
1.2.6	The Fifth Generation—Low-Power and Invisible Computers	26
1.3	THE COMPUTER ZOO	28
1.3.1	Technological and Economic Forces	28
1.3.2	The Computer Spectrum	30
1.3.3	Disposable Computers	31

1.3.4	Microcontrollers	33
1.3.5	Mobile and Game Computers	35
1.3.6	Personal Computers	36
1.3.7	Servers	36
1.3.8	Mainframes	38
1.4	EXAMPLE COMPUTER FAMILIES	39
1.4.1	Introduction to the x86 Architecture	39
1.4.2	Introduction to the ARM Architecture	45
1.4.3	Introduction to the AVR Architecture	47
1.5	METRIC UNITS	49
1.6	OUTLINE OF THIS BOOK	50
2.1	PROCESSORS	55
2.1.1	CPU Organization	56
2.1.2	Instruction Execution	58
2.1.3	RISC versus CISC	62
2.1.4	Design Principles for Modern Computers	63
2.1.5	Instruction-Level Parallelism	65
2.1.6	Processor-Level Parallelism	69
2.2	PRIMARYMEMORY	73
2.2.1	Bits	74
2.2.2	Memory Addresses	74
2.2.3	Byte Ordering	76
2.2.4	Error-Correcting Codes	78
2.2.5	Cache Memory	82
2.2.6	Memory Packaging and Types	85
2.3	SECONDARYMEMORY	86
2.3.1	Memory Hierarchies	86
2.3.2	Magnetic Disks	87
2.3.3	IDE Disks	91
2.3.4	SCSI Disks	92
2.3.5	RAID	94
2.3.6	Solid-State Disks	97
2.3.7	CD-ROMs	99
2.3.8	CD-Recordables	103
2.3.9	CD-Rewritables	105
2.3.10	DVD	106
2.3.11	Blu-ray	108
2.4	INPUT/OUTPUT	108
2.4.1	Buses	108
2.4.2	Terminals	113
2.4.3	Mice	118
2.4.4	Game Controllers	120
2.4.5	Printers	122
2.4.6	Telecommunications Equipment	127
2.4.7	Digital Cameras	135
2.4.8	Character Codes	137
2.5	SUMMARY	142
3.1	GATES AND BOOLEAN ALGEBRA	147
3.1.1	Gates	148
3.1.2	Boolean Algebra	150
3.1.3	Implementation of Boolean Functions	152
3.1.4	Circuit Equivalence	153
3.2	BASIC DIGITAL LOGIC CIRCUITS	158
3.2.1	Integrated Circuits	158

3.2.2 Combinational Circuits	159
3.2.3 Arithmetic Circuits	163
3.2.4 Clocks	168
3.3 MEMORY	169
3.3.1 Latches	169
3.3.2 Flip-Flops	172
3.3.3 Registers	174
3.3.4 Memory Organization	174
3.3.5 Memory Chips	178
3.3.6 RAMs and ROMs	180
3.4 CPU CHIPS AND BUSES	185
3.4.1 CPU Chips	185
3.4.2 Computer Buses	187
3.4.3 Bus Width	190
3.4.4 Bus Clocking	191
3.4.5 Bus Arbitration	196
3.4.6 Bus Operations	198
3.5 EXAMPLE CPU CHIPS	201
3.5.1 The Intel Core i7	201
3.5.2 The Texas Instruments OMAP4430 System-on-a-Chip	208
3.5.3 The Atmel ATmega168 Microcontroller	212
3.6 EXAMPLE BUSES	214
3.6.1 The PCI Bus	215
3.6.2 PCI Express	223
3.6.3 The Universal Serial Bus	228
3.7 INTERFACING	232
3.7.1 I/O Interfaces	232
3.7.2 Address Decoding	233
3.8 SUMMARY	235
4.1 AN EXAMPLE MICROARCHITECTURE	243
4.1.1 The Data Path	244
4.1.2 Microinstructions	251
4.1.3 Microinstruction Control: The Mic-1	253
4.2 AN EXAMPLE ISA: IJVM	258
4.2.1 Stacks	258
4.2.2 The IJVM Memory Model	260
4.2.3 The IJVM Instruction Set	262
4.2.4 Compiling Java to IJVM	266
4.3 AN EXAMPLE IMPLEMENTATION	267
4.3.1 Microinstructions and Notation	267
4.3.2 Implementation of IJVM Using the Mic-1	272
4.4 DESIGN OF THE MICROARCHITECTURE LEVEL	283
4.4.1 Speed versus Cost	283
4.4.2 Reducing the Execution Path Length	286
4.4.3 A Design with Prefetching: The Mic-2	293
4.4.4 A Pipelined Design: The Mic-3	293
4.4.5 A Seven-Stage Pipeline: The Mic-4	301
4.5 IMPROVING PERFORMANCE	305
4.5.1 Cache Memory	306
4.5.2 Branch Prediction	312
4.5.3 Out-of-Order Execution and Register Renaming	317
4.5.4 Speculative Execution	322
4.6 EXAMPLES OF THE MICROARCHITECTURE LEVEL	324
4.6.1 The Microarchitecture of the Core i7 CPU	325

- 4.6.2 The Microarchitecture of the OMAP4430 CPU 331
- 4.6.3 The Microarchitecture of the ATmega168 Microcontroller 336
- 4.7 COMPARISON OF THE i7, OMAP4430, AND ATMEGA168 338
- 4.8 SUMMARY 339
- 5.1 OVERVIEW OF THE ISA LEVEL
 - 5.1.1 Properties of the ISA Level
 - 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
 - 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
 - 5.8.4 Instruction Scheduling
 - 5.8.5 Reducing Conditional Branches: Predication
 - 5.8.6 Speculative Loads
- 5.9 SUMMARY
- 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 VIRTUAL I/O INSTRUCTIONS
 - 6.2.1 Files
 - 6.2.2 Implementation of Virtual I/O Instructions
 - 6.2.3 Directory Management Instructions
- 6.3 VIRTUAL INSTRUCTIONS FOR PARALLEL PROCESSING
 - 6.3.1 Process Creation
 - 6.3.2 Race Conditions
 - 6.3.3 Process Synchronization Using Semaphores
- 6.4 EXAMPLE OPERATING SYSTEMS
 - 6.4.1 Introduction
 - 6.4.2 Examples of Virtual Memory
 - 6.4.3 Examples of Virtual I/O
 - 6.4.4 Examples of Process Management
- 6.5 SUMMARY
- 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
- 8.1 ON-CHIP PARALELLISM
 - 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 Media Processors
 - 8.2.3 Cryptoprocessors
- 8.3 SHARED-MEMORYMULTIPROCESSORS
 - 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
- 9.1 SUGGESTIONS FOR FURTHER READING
 - 9.1.1 Introduction and General Works
 - 9.1.2 Computer Systems Organization
 - 9.1.3 The Digital Logic Level
 - 9.1.4 The Microarchitecture Level
 - 9.1.5 The Instruction Set Architecture Level
 - 9.1.6 The Operating System Machine Level
 - 9.1.7 The Assembly Language Level
 - 9.1.8 Parallel Computer Architectures
 - 9.1.9 Binary and Floating-Point Numbers
 - 9.1.10 Assembly Language Programming
- 9.2 ALPHABETICAL BIBLIOGRAPHY
- A.1 FINITE-PRECISION NUMBERS
- A.2 RADIX NUMBER SYSTEMS
- A.3 CONVERSION FROM ONE RADIX TO ANOTHER
- A.4 NEGATIVE BINARY NUMBERS
- A.5 BINARY ARITHMETIC
- B.1 PRINCIPLES OF FLOATING POINT
- B.2 IEEE FLOATING-POINT STANDARD 754
- 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 Assembler, as88
 - C.5.3 Some Differences with Other 8088 Assemblers
- 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
- • • • • [\(收起\)](#)

[Structured Computer Organization_下载链接1](#)

标签

计算机组成原理

计算机科学

computer

CS

计算机

arch

computer-science

美國

评论

结构化、也叫层次化计算机组成，让人对计算机的结构栈有一个清醒的认识

作者极爱恶意卖萌以及讲不好笑的冷笑话-_-

[Structured Computer Organization_下载链接1](#)

书评

这本书给我最大的启发不是说它的知识讲的有多高深，有多详细。
而是本书给我带来了层次化的思想和视角，我们知道，计算机是一个很复杂的系统，要想理解这一复杂系统，是比较困难的。当年学习计算机组成原理的时候就是一头雾水。
为什么，教材把关于计算机的所有层面的知识堆在...

[Structured Computer Organization_下载链接1](#)