

# Computer Systems: An Integrated Approach to Architecture and Operating Systems

---

## Table of Contents

<b>Preface</b> .....	<b>i</b>
Why a New Book on Computer Systems? .....	i
The structure of the book .....	ii
Where Does This Textbook Fit into the Continuum of CS Curriculum? .....	iii
Supplementary Material for Teaching an Integrated Course in Systems .....	v
Example project ideas included in the supplementary material .....	vi
<b>Chapter 1 Introduction</b> .....	<b>1-1</b>
1.1 What is Inside a Box? .....	1-1
1.2 Levels of Abstraction in a Computer System .....	1-2
1.3 The Role of the Operating System .....	1-5
1.4 What is happening inside the box? .....	1-7
1.4.1 Launching an application on the computer .....	1-9
1.5 Evolution of Computer Hardware .....	1-9
1.6 Evolution of Operating Systems.....	1-11
1.7 Roadmap of the rest of the book.....	1-12
1.8 Review Questions.....	1-13
<b>Chapter 2 Processor Architecture</b> .....	<b>2-1</b>
2.1 What is involved in processor design?.....	2-2
2.2 How do we design an instruction set?.....	2-2
2.3 A Common High-Level Language Feature Set.....	2-3

2.4 Expressions and Assignment Statements .....	2-4
2.4.1 Where to keep the operands? .....	2-4
2.4.2 How do we specify a memory address in an instruction? .....	2-9
2.4.3 How wide should each operand be?.....	2-10
2.4.4 Endianness .....	2-12
2.4.5 Packing of operands and Alignment of word operands.....	2-15
2.5 High-level data abstractions .....	2-17
2.5.1 Structures.....	2-17
2.5.2 Arrays .....	2-18
2.6 Conditional statements and loops .....	2-20
2.6.1 If-then-else statement .....	2-20
2.6.2 Switch statement .....	2-22
2.6.3 Loop statement.....	2-24
2.7 Checkpoint .....	2-24
2.8 Compiling Function calls .....	2-24
2.8.1 State of the Caller .....	2-25
2.8.2 Remaining chores with procedure calling.....	2-28
2.8.3 Software Convention .....	2-30
2.8.4 Activation Record.....	2-36
2.8.5 Recursion .....	2-37
2.8.6 Frame Pointer .....	2-37
2.9 Instruction-Set Architecture Choices .....	2-40
2.9.1 Additional Instructions.....	2-40
2.9.2 Additional addressing modes.....	2-41
2.9.3 Architecture styles .....	2-41
2.9.4 Instruction Format .....	2-42
2.10 LC-2200 Instruction Set.....	2-45
2.10.1 Instruction Format .....	2-45
2.10.2 LC-2200 Register Set .....	2-48
2.11 Issues influencing processor design.....	2-48
2.11.1 Instruction-set.....	2-48
2.11.2 Influence of applications on instruction-set design.....	2-50

2.11.3 Other issues driving processor design .....	2-51
2.12 Summary .....	2-52
2.13 Review Questions.....	2-53

## **Chapter 3 Processor Implementation ..... 3-1**

3.1 Architecture versus Implementation .....	3-1
3.2 What is involved in Processor Implementation? .....	3-2
3.3 Key hardware concepts.....	3-3
3.3.1 Circuits .....	3-3
3.3.2 Hardware resources of the datapath.....	3-3
3.3.3 Edge Triggered Logic .....	3-5
3.3.4 Connecting the datapath elements .....	3-7
3.3.5 Towards bus-based Design .....	3-10
3.3.6 Finite State Machine (FSM).....	3-13
3.4 Datapath Design.....	3-15
3.4.1 ISA and datapath width.....	3-17
3.4.2 Width of the Clock Pulse .....	3-18
3.4.3 Checkpoint .....	3-18
3.5 Control Unit Design.....	3-18
3.5.1 ROM plus state register .....	3-19
3.5.2 FETCH macro state .....	3-23
3.5.3 DECODE macro state.....	3-25
3.5.4 EXECUTE macro state: ADD instruction (part of R-Type).....	3-26
3.5.5 EXECUTE macro state: NAND instruction (part of R-Type) .....	3-28
3.5.6 EXECUTE macro state: JALR instruction (part of J-Type) .....	3-28
3.5.7 EXECUTE macro state: LW instruction (part of I-Type) .....	3-29
3.5.8 EXECUTE macro state: SW and ADDI instructions (part of I-Type) .....	3-30
3.5.9 EXECUTE macro state: BEQ instruction (part of I-Type) .....	3-31
3.5.10 Engineering a conditional branch in the microprogram .....	3-32
3.5.11 DECODE macro state revisited .....	3-34
3.6 Alternative Style of Control Unit Design .....	3-35
3.6.1 Microprogrammed Control.....	3-35

3.6.2 Hardwired control .....	3-36
3.6.3 Choosing between the two control design styles .....	3-37
3.7 Summary .....	3-38
3.8 Historical Perspective.....	3-38
3.9 Review Questions.....	3-41
<b>Chapter 4 Interrupts, Traps and Exceptions .....</b>	<b>4-1</b>
4.1 Discontinuities in program execution .....	4-2
4.2 Dealing with program discontinuities .....	4-4
4.3 Architectural enhancements to handle program discontinuities .....	4-7
4.3.1 Modifications to FSM .....	4-8
4.3.2 A simple interrupt handler.....	4-9
4.3.3 Handling cascaded interrupts .....	4-10
4.3.4 Returning from the handler .....	4-13
4.3.5 Checkpoint .....	4-14
4.4 Hardware details for handling program discontinuities .....	4-14
4.4.1 Datapath details for interrupts .....	4-14
4.4.2 Details of receiving the address of the handler .....	4-16
4.4.3 Stack for saving/restoring .....	4-18
4.5 Putting it all together .....	4-20
4.5.1 Summary of Architectural/hardware enhancements .....	4-20
4.5.2 Interrupt mechanism at work .....	4-20
4.6 Summary .....	4-23
4.7 Review Questions.....	4-25
<b>Chapter 5 Processor Performance and Rudiments of Pipelined Processor Design .....</b>	<b>5-1</b>
5.1 Space and Time Metrics .....	5-1
5.2 Instruction Frequency .....	5-4
5.3 Benchmarks.....	5-5
5.4 Increasing the Processor Performance .....	5-9
5.5 Speedup .....	5-10
5.6 Increasing the Throughput of the Processor .....	5-14

5.7 Introduction to Pipelining .....	5-14
5.8 Towards an instruction processing assembly line .....	5-15
5.9 Problems with a simple-minded instruction pipeline .....	5-17
5.10 Fixing the problems with the instruction pipeline .....	5-18
5.11 Datapath elements for the instruction pipeline .....	5-20
5.12 Pipeline-conscious architecture and implementation .....	5-22
5.12.1 Anatomy of an instruction passage through the pipeline .....	5-23
5.12.2 Design of the Pipeline Registers.....	5-26
5.12.3 Implementation of the stages.....	5-27
5.13 Hazards.....	5-27
5.13.1 Structural hazard.....	5-28
5.13.2 Data Hazard.....	5-30
5.13.3 Control Hazard .....	5-41
5.13.4 Summary of Hazards .....	5-51
5.14 Dealing with program discontinuities in a pipelined processor .....	5-52
5.15 Advanced topics in processor design .....	5-55
5.15.1 Instruction Level Parallelism .....	5-55
5.15.2 Deeper pipelines .....	5-56
5.15.3 Revisiting program discontinuities in the presence of out-of-order processing .....	5-59
5.15.4 Managing shared resources.....	5-60
5.15.5 Power Consumption .....	5-62
5.15.6 Multi-core Processor Design .....	5-63
5.15.7 Intel Core Microarchitecture: An example pipeline .....	5-64
5.16 Summary .....	5-67
5.17 Historical Perspective.....	5-67
5.18 Review Questions.....	5-68

## **Chapter 6 Processor Scheduling..... 6-1**

6.1 Introduction .....	6-1
6.2 Programs and Processes .....	6-2
6.3 Scheduling Environments.....	6-7
6.4 Scheduling Basics .....	6-9

6.5 Performance Metrics .....	6-12
6.6 Non-preemptive Scheduling Algorithms.....	6-15
6.6.1 First-Come First-Served (FCFS).....	6-15
6.6.2 Shortest Job First (SJF) .....	6-19
6.6.3 Priority.....	6-21
6.7 Preemptive Scheduling Algorithms.....	6-23
6.7.1 Round Robin Scheduler.....	6-26
6.8 Combining Priority and Preemption .....	6-31
6.9 Meta Schedulers .....	6-31
6.10 Evaluation .....	6-32
6.11 Impact of Scheduling on Processor Architecture.....	6-34
6.12 Summary and a Look ahead.....	6-36
6.13 Linux Scheduler – A case study .....	6-36
6.14 Historical Perspective.....	6-39
6.15 Review Questions.....	6-41

## **Chapter 7 Memory Management Techniques..... 7-1**

7.1 Functionalities provided by a memory manager .....	7-1
7.2 Simple Schemes for Memory Management .....	7-4
7.3 Memory Allocation Schemes .....	7-8
7.3.1 Fixed Size Partitions .....	7-9
7.3.2 Variable Size Partitions .....	7-10
7.3.3 Compaction.....	7-13
7.4 Paged Virtual Memory .....	7-14
7.4.1 Page Table .....	7-17
7.4.2 Hardware for Paging .....	7-19
7.4.3 Page Table Set up.....	7-20
7.4.4 Relative sizes of virtual and physical memories .....	7-20
7.5 Segmented Virtual Memory.....	7-21
7.5.1 Hardware for Segmentation .....	7-26
7.6 Paging versus Segmentation .....	7-27
7.6.1 Interpreting the CPU generated address.....	7-29

7.7 Summary .....	7-30
7.8 Historical Perspective.....	7-32
7.8.1 MULTICS.....	7-33
7.8.2 Intel’s Memory Architecture.....	7-35
7.9 Review Questions.....	7-36

## **Chapter 8 Details of Page-based Memory Management.... 8-1**

8.1 Demand Paging .....	8-1
8.1.1 Hardware for demand paging.....	8-1
8.1.2 Page fault handler.....	8-2
8.1.3 Data structures for Demand-paged Memory Management.....	8-3
8.1.4 Anatomy of a Page Fault .....	8-5
8.2 Interaction between the Process Scheduler and Memory Manager.....	8-8
8.3 Page Replacement Policies .....	8-9
8.3.1 Belady’s Min.....	8-10
8.3.2 Random Replacement.....	8-10
8.3.3 First In First Out (FIFO).....	8-11
8.3.4 Least Recently Used (LRU) .....	8-13
8.3.5 Second chance page replacement algorithm.....	8-17
8.3.6 Review of page replacement algorithms .....	8-20
8.4 Optimizing Memory Management.....	8-22
8.4.1 Pool of free page frames.....	8-22
8.4.2 Thrashing.....	8-23
8.4.3 Working set.....	8-25
8.4.4 Controlling thrashing .....	8-26
8.5 Other considerations .....	8-28
8.6 Translation Lookaside Buffer (TLB) .....	8-28
8.6.1 Address Translation with TLB.....	8-29
8.7 Advanced topics in memory management .....	8-31
8.7.1 Multi-level page tables.....	8-31
8.7.2 Access rights as part of the page table entry.....	8-34
8.7.3 Inverted page tables .....	8-34

8.8 Summary .....	8-34
8.9 Review Questions.....	8-35
<b>Chapter 9 Memory Hierarchy .....</b>	<b>9-1</b>
9.1 The Concept of a Cache .....	9-2
9.2 Principle of Locality .....	9-3
9.3 Basic terminologies .....	9-4
9.4 Multilevel Memory Hierarchy.....	9-5
9.5 Cache organization.....	9-8
9.6 Direct-mapped cache organization.....	9-9
9.6.1 Cache Lookup .....	9-11
9.6.2 Fields of a Cache Entry .....	9-13
9.6.3 Hardware for direct mapped cache.....	9-14
9.7 Repercussion on pipelined processor design.....	9-16
9.8 Cache read/write algorithms .....	9-17
9.8.1 Read access to the cache from the CPU .....	9-18
9.8.2 Write access to the cache from the CPU .....	9-19
9.9 Dealing with cache misses in the processor pipeline .....	9-22
9.9.1 Effect of memory stalls due to cache misses on pipeline performance .....	9-23
9.10 Exploiting spatial locality to improve cache performance.....	9-25
9.10.1 Performance implications of increased blocksize.....	9-30
9.11 Flexible placement .....	9-31
9.11.1 Fully associative cache .....	9-32
9.11.2 Set associative cache .....	9-34
9.11.3 Extremes of set associativity.....	9-37
9.12 Instruction and Data caches.....	9-39
9.13 Reducing miss penalty .....	9-40
9.14 Cache replacement policy.....	9-41
9.15 Recapping Types of Misses .....	9-43
9.16 Integrating TLB and Caches.....	9-46
9.17 Cache controller .....	9-48
9.18 Virtually indexed physically tagged cache .....	9-49

9.19 Recap of Cache Design Considerations .....	9-52
9.20 Main memory design considerations.....	9-52
9.20.1 Simple main memory .....	9-53
9.20.2 Main memory and bus to match cache block size.....	9-54
9.20.3 Interleaved memory .....	9-55
9.21 Elements of a modern main memory systems .....	9-56
9.21.1 Page mode DRAM .....	9-61
9.22 Performance implications of memory hierarchy .....	9-62
9.23 Summary .....	9-63
9.24 Memory hierarchy of modern processors – An example .....	9-65
9.25 Review Questions.....	9-66

## **Chapter 10 Input/Output and Stable Storage ..... 10-1**

10.1 Communication between the CPU and the I/O devices .....	10-1
10.1.1 Device controller .....	10-2
10.1.2 Memory Mapped I/O .....	10-3
10.2 Programmed I/O .....	10-5
10.3 DMA .....	10-6
10.4 Buses .....	10-9
10.5 I/O Processor.....	10-10
10.6 Device Driver .....	10-11
10.6.1 An Example .....	10-12
10.7 Peripheral Devices .....	10-15
10.8 Disk Storage .....	10-17
10.8.1 Saga of Disk Technology.....	10-24
10.9 Disk Scheduling Algorithms.....	10-27
10.9.1 First-Come First Served .....	10-30
10.9.2 Shortest Seek Time First .....	10-30
10.9.3 Scan (elevator algorithm).....	10-31
10.9.4 C-Scan (Circular Scan) .....	10-32
10.9.5 Look and C-Look.....	10-33
10.9.6 Disk Scheduling Summary .....	10-33

10.9.7 Comparison of the Algorithms .....	10-34
10.10 Solid State Drive .....	10-36
10.11 Evolution of I/O Buses and Device Drivers .....	10-38
10.11.1 Dynamic Loading of Device Drivers.....	10-39
10.11.2 Putting it all Together .....	10-39
10.12 Summary .....	10-42
10.13 Review Questions.....	10-42

## **Chapter 11 File System..... 11-1**

11.1 Attributes .....	11-2
11.2 Design Choices in implementing a File System on a Disk Subsystem .....	11-8
11.2.1 Contiguous Allocation .....	11-9
11.2.2 Contiguous Allocation with Overflow Area.....	11-12
11.2.3 Linked Allocation.....	11-12
11.2.4 File Allocation Table (FAT) .....	11-13
11.2.5 Indexed Allocation .....	11-15
11.2.6 Multilevel Indexed Allocation .....	11-17
11.2.7 Hybrid Indexed Allocation.....	11-18
11.2.8 Comparison of the allocation strategies.....	11-21
11.3 Putting it all together .....	11-22
11.3.1 i-node .....	11-28
11.4 Components of the File System .....	11-29
11.4.1 Anatomy of creating and writing files.....	11-30
11.5 Interaction among the various subsystems .....	11-31
11.6 Layout of the file system on the physical media.....	11-34
11.6.1 In memory data structures .....	11-37
11.7 Dealing with System Crashes .....	11-38
11.8 File systems for other physical media.....	11-39
11.9 A summary of modern file systems .....	11-39
11.9.1 Linux .....	11-39
11.9.2 Microsoft Windows.....	11-45
11.10 Summary .....	11-47

11.11 Review Questions.....	11-48
-----------------------------	-------

## **Chapter 12 Multithreaded Programming and Multiprocessors ..... 12-1**

12.1 Why Multithreading? .....	12-1
12.2 Programming support for threads .....	12-3
12.2.1 Thread creation and termination.....	12-3
12.2.2 Communication among threads .....	12-6
12.2.3 Read-write conflict, Race condition, and Non-determinism .....	12-7
12.2.4 Synchronization among threads .....	12-12
12.2.5 Internal representation of data types provided by the threads library.....	12-19
12.2.6 Simple programming examples .....	12-20
12.2.7 Deadlocks and livelocks .....	12-25
12.2.8 Condition variables .....	12-27
12.2.9 A complete solution for the video processing example .....	12-30
12.2.10 Rechecking the predicate.....	12-33
12.3 Summary of thread function calls and threaded programming concepts.....	12-36
12.4 Points to remember in programming with threads.....	12-38
12.5 Using threads as software structuring abstraction.....	12-39
12.6 POSIX pthreads library calls summary .....	12-40
12.7 OS support for threads.....	12-42
12.7.1 User level threads .....	12-45
12.7.2 Kernel level threads .....	12-47
12.7.3 Solaris threads: An example of kernel level threads .....	12-49
12.7.4 Threads and libraries.....	12-50
12.8 Hardware support for multithreading in a uniprocessor.....	12-51
12.8.1 Thread creation, termination, and communication among threads .....	12-51
12.8.2 Inter-thread synchronization .....	12-51
12.8.3 An atomic test-and-set instruction .....	12-52
12.8.4 Lock algorithm with test-and-set instruction.....	12-54
12.9 Multiprocessors .....	12-55
12.9.1 Page tables .....	12-56

12.9.2 Memory hierarchy .....	12-56
12.9.3 Ensuring atomicity .....	12-59
12.10 Advanced Topics .....	12-59
12.10.1 OS topics .....	12-60
12.10.2 Architecture topics.....	12-76
12.10.3 The Road Ahead: Multi- and Many-core Architectures.....	12-87
12.11 Summary.....	12-89
12.12 Historical Perspective.....	12-90
12.13 Review Questions.....	12-92

## **Chapter 13 Fundamentals of Networking and Network Protocols..... 13-1**

13.1 Preliminaries .....	13-1
13.2 Basic Terminologies .....	13-2
13.3 Networking Software .....	13-6
13.4 Protocol Stack .....	13-8
13.4.1 Internet Protocol Stack .....	13-9
13.4.2 OSI Model.....	13-12
13.4.3 Practical issues with layering .....	13-13
13.5 Application Layer .....	13-14
13.6 Transport Layer .....	13-15
13.6.1 Stop and wait protocols.....	13-17
13.6.2 Pipelined protocols .....	13-20
13.6.3 Reliable Pipelined Protocol .....	13-22
13.6.4 Dealing with transmission errors.....	13-28
13.6.5 Transport protocols on the Internet.....	13-28
13.6.6 Transport Layer Summary.....	13-32
13.7 Network Layer.....	13-32
13.7.1 Routing Algorithms .....	13-33
13.7.2 Internet Addressing .....	13-40
13.7.3 Network Service Model.....	13-43
13.7.4 Network Routing Vs. Forwarding.....	13-47

13.7.5 Network Layer Summary .....	13-48
13.8 Link Layer and Local Area Networks .....	13-50
13.8.1 Ethernet .....	13-50
13.8.2 CSMA/CD.....	13-51
13.8.3 IEEE 802.3.....	13-53
13.8.4 Wireless LAN and IEEE 802.11 .....	13-54
13.8.5 Token Ring.....	13-55
13.8.6 Other link layer protocols .....	13-57
13.9 Networking Hardware.....	13-58
13.10 Relationship between the Layers of the Protocol Stack .....	13-63
13.11 Data structures for packet transmission.....	13-63
13.11.1 TCP/IP Header .....	13-65
13.12 Message transmission time .....	13-66
13.13 Summary of Protocol Layer Functionalities .....	13-72
13.14 Networking Software and the Operating System .....	13-73
13.14.1 Socket Library.....	13-73
13.14.2 Implementation of the Protocol Stack in the Operating System.....	13-75
13.14.3 Network Device Driver.....	13-76
13.15 Network Programming using Unix Sockets.....	13-77
13.16 Network Services and Higher Level Protocols .....	13-85
13.17 Summary .....	13-86
13.18 Historical Perspective.....	13-87
13.18.1 From Telephony to Computer Networking.....	13-87
13.18.2 Evolution of the Internet .....	13-90
13.18.3 PC and the arrival of LAN .....	13-91
13.18.4 Evolution of LAN .....	13-91
13.19 Review Questions.....	13-94

## **Chapter 14 Epilogue: A Look Back at the Journey..... 14-1**

14.1 Processor Design .....	14-1
14.2 Process .....	14-1
14.3 Virtual Memory System and Memory Management.....	14-2

14.4 Memory Hierarchy .....	14-2
14.5 Parallel System .....	14-3
14.6 Input/Output Systems.....	14-3
14.7 Persistent Storage .....	14-3
14.8 Network .....	14-4
14.9 Concluding Remarks.....	14-4

## **Appendix A Network Programming with Unix Sockets ... A-1**

A.1 The problem.....	A-1
A.2 Source files provided .....	A-1
A.3 Makefile .....	A-1
A.4 Common header file .....	A-3
A.5 Client source code .....	A-3
A.6 Server source code .....	A-7
A.7 Instantiating the client/server programs.....	A-12

