

**CS/ECE 333 - Fall 2006
Final Exam Important Concepts Guide**

Summary of Readings and Textbook Questions

1. Textbook Chapters 1-5 (Entire chapters)
 - a. Chapter 1 - Introduction
 - i. Problem 1.1 – Given a memory size and a word size, be able to figure out how many words of memory can be stored
 - ii. Problem 1.13 – What is an ISA? What are its components?
 - iii. Problem 1.12 – What is the stored program concept? Why is this important?
 - b. Chapter 2 – Instruction Set Architecture
 - i. Probs. 2.5 – Given a program, calculate the total memory traffic in bytes for both instruction fetch and instruction execute.
 - ii. Probs. 2.7, 2.16 – Write SRC code for some high-level expression/program
 - iii. Probs. 2.23, 2.25 – Describe the difference in addressing modes, given an example in RTN
 - iv. Procedure calling conventions –
 1. What is a procedure calling convention? Why do we need it? (Functions need to agree on how to pass parameters, return values, where to store local variables, etc.)
 2. What are the basic steps in a calling convention? (Class slides discuss a suggested calling convention implementation for SRC)
 3. Terminology – Caller, Callee, Activation record (What is it, what information does it contain?)
 - c. Chapter 3 – Real Machines and Performance Measurement
 - i. Probs like 3.1, 3.6, 3.24
 - ii. Terminology – Speedup (and how to calculate it), RISC design philosophy, CISC characteristics
 - d. Chapter 4 – Processor Design
 - i. Know the difference between concrete and abstract RTN
 - ii. (datapath labs) Given some datapath diagram and an ISA specification, identify control signals required by the datapath for instruction XYZ
 - iii. (datapath labs) Given some datapath be able to explain how a particular instruction gets executed in the datapath.
 - e. Chapter 5 – Pipelining, Pipeline Hazards, Instruction Level Parallelism, and Microprogramming
 - i. Pipelining and Pipeline Hazards
 1. Probs like 5.3, 5.4, 5.8, 5.9 – Be able to diagram execution of instructions through a pipeline, identify hazards
 2. Terminology - bandwidth, latency, types of hazards, hardware solutions for hazards
 - a. data hazards - data forwarding/bypassing or stalling

CS/ECE 333 – Fall 2006 – Final Exam Concept Guide

- b. control hazards – branch prediction
 - c. structural hazards – add more hardware
 - ii. Instruction Level Parallelism
 1. Goals (increase instruction execution rate)
 2. Techniques
 - a. pipelining
 - b. VLIW
 - c. superscalar
 - i. multiple instructions issued in the same clock tick
 - ii. multiple pipelines
 - iii. Microprogramming a control unit
 1. What is it? How is it different from hardwiring a control unit? What are some benefits of microprogramming? What are some disadvantages?
 2. What are the components?
 - a. Microcode memory, sequencer
 3. What is the difference between vertical and horizontal microcode?
 4. (microprogrammed control unit lab) Approach/process – Be familiar with how a microcoded control unit is designed.
- 2. Textbook Chapter 6
 - a. Sections 6.1 and 6.4
 - b. Problems like 6.1, 6.2, 6.3 (Given some binary representation and a representation scheme, what number is representation?)
 - c. Terminology – sign-magnitude, 2's complement, 1's complement
- 3. Textbook Chapter 7
 - a. Sections 7.1, 7.4-7.7
 - b. Problems like: 7.1, 7.2, 7.19-7.21
 - c. Terminology:
 - i. Principle of locality (spatial and temporal)
 - ii. Memory hierarchy – what it is, why do modern processors have a memory hierarchy?
 - iii. Caches
 1. block/line
 2. Block placement strategies
 - a. set associative, direct-mapped, fully associative
 3. cache miss, cache hit
 4. miss ratio, hit ratio
 5. Block replacement strategies (LRU, random)
 6. Read and write policies
 - a. Write Hit policies
 - i. Write-through – writes to cache and to memory (Consistency)
 - ii. Write-back – writes to cache only. Updates memory on eviction (Speed)

CS/ECE 333 – Fall 2006 – Final Exam Concept Guide

- b. Read Miss policies –
 - i. Bring in block, wait for whole block to arrive in cache before allowing the desired word to be used
 - ii. Bring in desired word first and serve it, rest of block loads in background
- c. Write Miss policies (allocate refers to allocating space in the CACHE)
 - i. Write allocate
 - ii. Write no-allocate
- iv. Virtual Memory
 - 1. What is it? Why do modern processors have it?
 - 2. How does it work?
 - 3. Terminology
 - a. Memory Management Unit (MMU)
 - b. virtual address
 - c. physical address
 - d. Address translation
 - e. Segmentation
 - f. Paging
 - i. Single-level paging
 - ii. Multilevel paging
 - iii. Inverted page tables (why?)
 - 1. Linear inverted page table
 - 2. Hashed inverted page table
 - g. Translation Lookaside Buffers (TLBs)
 - i. What are they? How do they work? What do they do?
- 4. Class slides/notes (NOTE: only those that differ significantly from textbook are noted below)
 - a. Cache example (class slides)
 - b. “Microprogramming 2” slides gives a presentation of the process for microprogramming a control unit
 - c. “Pipelining 2” notes are a summary of class describing pipelining
 - d. “Pipelining in Real Processors” summarizes class presentation of
 - i. static vs. dynamic multiple issue
 - ii. reservation stations
 - iii. simple branch prediction schemes
 - 1. static prediction
 - 2. dynamic prediction (simple bimodal predictor)
 - e. Virtual memory example exercise
 - f. Inverted page table notes
- 5. Virtual memory reference reading (Supplemental, not presented in class. Only gives high-level overview) http://en.wikipedia.org/wiki/Virtual_memory

Chapter 1

CS/ECE 333 – Fall 2006 – Final Exam Concept Guide

1. Basic elements of a computer and their functions
 - Input, Output, Arithmetic and logic, Control, Memory, Bus
2. Basic terminology
 - Numerical prefixes and their meanings: Kilo, Mega, Giga, Tera, etc.
3. Programmer's view
 - Compiler
 - Assembler
 - High level language
 - Assembly language
 - Machine language
 - Program
 - Stored program concept
 - Instruction
 - Instruction set architecture (ISA)
 - Operation code (Opcode)
 - Operands
 - Word length (byte, nibble, word)
 - Fetch-execute cycle
 - CISC (Complex Instruction Set Computer)
 - RISC (Reduced Instruction Set Computer)
4. Basic registers and their functions within the computer
 - Program counter
 - Instruction register
 - General-purpose registers
5. Computer logic design
 - Data path
 - Control path
6. Instruction classes
 - Data movement
 - Arithmetic and logic
 - Control flow
9. Memory hierarchy
 - CPU registers
 - Cache memory
 - Main memory
 - Disk memory
 - Tape memory

Chapter 2

1. Basic concepts of memory
 - Memory as a collection of storage devices
 - Word length

CS/ECE 333 – Fall 2006 – Final Exam Concept Guide

- Concept of a memory address
 - Memory address space
 - Basic memory operations
 - Memory read
 - Memory write
 - Program memory
 - Data memory
2. Basic instruction cycle
 - Instruction address calculation
 - Instruction fetch
 - Operand address calculation
 - Operand fetch
 - Instruction execution
 - Result operand address calculation
 - Operand store
 3. Details of the role of special-purpose registers
 - Program counter
 - Stack pointer
 - Memory address register
 - Memory data register
 - Instruction register
 4. Instruction format
 - Operation code
 - Source operand
 - Result operand
 5. Instruction classes
 - Data transfer
 - Arithmetic and logic
 - Logical shift instructions
 - Arithmetic shift instructions
 - Rotate instructions
 - Program sequencing and control
 - Input/output
 6. Instruction representation and formats
 - Instruction mnemonics
 - Four address
 - Three address
 - Two address
 - One address
 - Zero address
 7. Addressing modes
 - Immediate
 - Absolute
 - Indirect

- Register Direct
 - Register Indirect
 - Indexed
8. Branching
 - Conditional branching
 - Unconditional branching
 - Condition code bits
 9. Subroutines
 - Subroutine call
 - Subroutine return
 - Return address
 - Subroutine nesting
 10. Register transfer notation

Chapter 3

1. Measures of machine performance
 - Execution time
 - Clock speed
 - Clocks per instruction (CPI)
 - MIPS (Millions of Instructions Per Second)
 - FLOPS (FLoating point Operations Per Second)
 - MFLOPS (Millions of FLOPS)
 - Whetstone benchmarks
 - Dhrystone benchmarks
 - SPEC (System Performance Evaluation Cooperative) benchmarks
2. Characteristics of Reduced Instruction Set Computer (RISC)
 - One instruction per cycle
 - Fixed instruction length
 - Only load and store instructions access memory
 - Simplified addressing modes
 - Fewer and simpler instructions
 - Delayed loads and branches
 - Prefetch and speculative execution
 - Let the compiler do it
3. Characteristics of Complex Instruction Set Computer (CISC)
 - Many complex instructions and addressing modes
 - Some instructions take many steps to execute
 - Not always easy to find the best instruction for a task
4. Motorola 68000 processor
 - Understand as an example of a CISC architecture
 - Instruction set
 - Instruction formats
 - Register set

- Addressing modes
 - Memory organization
 - Condition code bits
 - Assembly language programming
 - Input/Output (I/O)
 - Memory-mapped I/O
 - Isolated I/O
5. Sun Microsystems SPARC (Scalable Processor ARChitecture)
- Understand as an example of a RISC architecture
 - Instruction set
 - Instruction format
 - Register set
 - Concept of pipelining

Chapters 4 and 5

1. Instruction Cycle
 - Machine cycles
 - Instruction fetch
 - Memory read
 - Memory write
 - Execute
 - Interrupt acknowledge
 - Others
 - Micro-operations
2. Detailed consideration of the role of the special-purpose registers
 - Program counter
 - Stack pointer
 - Memory address register
 - Memory data register
 - Instruction register
3. Organization of CPU hardware
 - Internal bus
 - Concept of control signals
4. Basic operations within CPU
 - Instruction fetch
 - Reading data from memory
 - Writing data to memory
 - Register to register transfers
 - Instruction execution
5. Timing issues
 - T states
 - Effect of propagation delay

- Effect of setup and hold times
- 6. Execution of a complete instruction
 - Sequencing of control actions
 - Instruction fetch
 - Incrementing the PC
 - Data fetch
 - Instruction execution
 - Data store
- 7. Control unit design choices
 - Hardwired control
 - Microprogrammed control
- 8. Hardwired control unit organization
 - Control step counter
 - Step decoder
 - Instruction decoder
 - Control signal generation
- 9. Microprogrammed control units
 - Microprogram counter (control address register - CAR)
 - Microprogram memory (control store)
 - Control buffer register
 - Organization of a microprogrammed control unit (hardware block diagram)
 - Sequencing of control words
 - Organization of control memory (what it contains)
 - Basic operations performed by a microprogrammed control unit
- 10. Microinstruction formats
 - Horizontal microprogramming
 - Vertical microprogramming
 - Direct encoding
 - Indirect encoding
 - Advantages and disadvantages of each approach
- 11. Next microinstruction address generation
 - Two address fields in the microinstruction
 - One address field in the microinstruction
 - A variable field in the microinstruction
 - Incrementation of the current address
- 12. Pipelining
 - Concept of an n-stage pipeline
 - Pipeline bandwidth
 - Pipeline latency

Chapter 6 – Arithmetic for Computers

1. Number representations
 - Two's complement

- One's complement
 - Sign and magnitude
2. Signed arithmetic using each number representation
 - Be able to do it
 - Know why it works
 3. Range of integers that can be represented with each technique
 4. Floating point representations

Chapter 7

1. Basic memory concepts
 - Address
 - Word size
 - Block size
 - Word addressable
 - Byte addressable
 - Memory address register (MAR)
 - Memory data register (MDR)
 - Little endian and big endian ordering
2. Memory hierarchy
 - Primary memory (registers, cache, main memory)
 - Secondary memory (disks)
 - Tertiary memory (tapes)
 - Principle of locality
3. Types of memory
 - RAM (Random Access Memory)
 - SRAM (Static RAM)
 - DRAM (Dynamic RAM)
4. Concept of virtual memory
 - Pages
 - Virtual addressing
 - Translation
 - Memory management unit
5. Cache memory
 - Mapping functions (direct, associative, set-associative)
 - Replacement algorithms (least recently used, random)
 - Cache hit
 - Cache miss