

Midterm 1 Study Guide

- Program performance
 - What aspects affect it and how
- Stored program concept, how was this revolutionary?
- Basic stored program execution flow (lec2, slide 6)
- RISC vs. CISC
 - Basic organization, how are they different?
 - Advantages and disadvantages
 - Where can operands come from? How is this advantageous for building RISC pipelines?
- MIPS
 - Registers
 - How many, how big, why are some special?
 - Instruction formats
 - What are they
 - Why only three
 - Advantages of fixed formatting/size
 - Why aren't all fields used? Advantages/disadvantages
 - Implications/side effects of changing a field size (e.g., changing the number of registers to 64 but leaving the instruction length fixed at 32-bits)
 - Support for procedure/function calls
 - How are arguments and return values handled
 - Temporary vs saved registers
 - Caller vs. callee saved registers
 - Return address?
 - Stack frame, what is in it and why is it useful
 - Frame pointer vs. stack pointer
 - Purpose of load linked and store conditional
 - Arithmetic vs. logical operations
 - Architectural structure
 - Advantages/disadvantages for speculative operations (e.g., fetching registers while the instruction is being decoded)
 - Instruction phases:
 - Instruction fetch, register fetch/instruction decode, execution/address calculation, memory access, register write back (e.g., Lec 6, slide 8)
 - What happens in each phase for each instruction type? High-level operation details based on the instruction being executed. I would not ask you to define signal values for datapath components
- Structural designs
 - Fast and big vs. small and slow
 - Implications
 - Advantages/disadvantages
 - Ex. Ripple carry vs. carry look ahead adders
 - How can this concept be applied to other structures
 - Single cycle (one long cycle) vs multi-cycle (multiple shorter cycles) designs for the same overall operation
 - Advantages/disadvantages
 - Multi-cycle vs. multiple-cycle delay path
 - Understand implications on the critical path for single vs. multi-cycle operation.
 - How does the clock cycle change? How does overall execution time of 1 instruction change? (not pipelined yet)
 - Maintaining timing constraints: different between single cycle violations and multi-cycle violations
- Processor design (note: simple designs for a small set of instructions)
 - Define datapath components based on a set of instructions

- Design an ALU based on a set of operations
 - Define a controller and datapath based on a set of instructions
 - Question 1 on the sample midterm is VERY IMPORTANT!
- Multipliers/Dividers
 - Iterative improvement purposes (what was being reduced)
 - Reasons why registers could be removed/combined, ALUs could be reduced
 - Know the basic progression of each version, but no details. I would remind you in the question of any details you needed to know
 - Work through an example for multiple/divide version 3
 - Show register values for each iteration
 - Booth's algorithm for multiply
 - What is the purpose of this algorithm
 - Basic idea of how it works
 - Do not need to work through an example
- Floating point
 - Convert a decimal number to binary single-precision floating point notation
 - Bias – what is it? What does it facilitate?
 - Single-precision vs. double precision
 - Ranges
 - Register layout
 - Exceptions
 - Underflow, overflow
 - Infinity, NaN
 - Decimal representation operations
 - Work through addition/subtract multiple/divide with and without round and guard bits
 - Purpose of special bits
 - Round, guard, sticky
 - Why aren't some FP operations associative? Give example
 - Challenges wrt to FP operations (e.g., precision, accumulated errors)
- ~~Performance~~
 - ~~○ Calculate CPI based on instruction mix~~
 - ~~○ Calculate CPI speedup based on architectural changes~~
 - ~~○ Compare CPIs of processors based on instruction mixes~~
 - ~~○ Chart on slide 16~~