Study Guide

For each module, you should be familiar with and able to explain all of the terms in the glossary provided at the end of each modules power point presentation. In addition, pay attention to the following.

Introduction

- Explain Moore's Law: what is it and what are the power, energy, and performance consequences?
- What is Dennard scaling?
- What are the major elements of the cost of producing modern processors?
- Power wall and memory wall – what are they?
- What are the major elements of a modern processing system?
- Role of the instruction set architecture.

Instruction Set Architecture

- Understand the MIPS memory map and the register structure, i.e., the conventions for using the general purpose register file.
- Instruction encodings
    - Given a MIPS program and the instruction set, be able to encode the data segment and all of the program instructions
    - Big and little endian storage
- Procedure calls
    - State of a stack frame after a sequence of nested calls and returns
    - State saving conventions
    - Computation of the encodings for `jal` instructions
- Program Assembly
    - Address computations for labels
    - Encoding of branch and jump instructions
    - Structure of an executable
        - Relocatable code
        - Symbol table
- Be able to read SPIM code and understand what is being computed

Arithmetic

- Operation of a 32-bit ALU
- Understand how to add instructions to the ALU
- Understand the ISA interface and implementation of unsigned integer multiplication/division.
- Understand the basic energy/power behavior of an ALU
    - What are the sources of energy dissipation?
    - Understand the difference between power and energy
- Basic elements of power dissipation
    - Basic RC behavior and leakage currents

- Expressions for dynamic power dissipation, static power dissipation, and delay
- Difference between energy and power
- Basic understanding of relationships between voltage, frequency, threshold voltage, and delay and the consequences for power dissipation.

Single Cycle Datapath

- Operation of a single cycle datapath
  - The state of all signals in the datapath for every instruction
  - The ability to add new instructions to the data path
    - Add the data flow paths
    - Modifying the controller
  - Energy behavior
    - Assessing the energy behavior of different instructions
    - Instruction level model of energy behavior
    - Component model of energy behavior

Multi-Cycle Datapath

- Operation of a multi-cycle datapath
  - The state of all signals in the datapath for any clock cycle of any instruction
  - The ability to add new instructions to the datapath
    - Add the data flow paths
    - Modifying the ROM controller
    - Modify the sequencer to implement the new instruction
  - Energy behavior
    - Assessing the energy behavior of different instructions
    - Instruction level model of energy behavior
    - Component model of energy behavior
- Exceptions
  - Adding support for new exceptions
  - Modifying the sequencer to add support for new exceptions
- Performance
  - Assessing performance using the CPI model
  - Amdal's Law and its applications
    - What is Amdahl's Law
    - Use in assessing performance limitations and identifying bottlenecks

Pipelined Datapath and Performance

- Hazards:
  - Data hazards, structural hazards and control hazards
  - Be able to distinguish between them and be able to identify all hazards in a sequence of SPIM code
- Schedule code to avoid or minimize pipeline stalls

- Given a pipeline structure and code sequence, determine the state of every pipeline register – this is equivalently the value of every signal propagating through the pipeline
- Set the value of all control signals for an instruction
- Write the forwarding condition
- Modify the forwarding condition for a modified pipeline, e.g., one that has added a stage
- Modify the pipeline and control to be able to add new instructions
- Modify the pipeline for exceptions
- Show the sequence of instructions that flow through the pipeline on a page fault
- Compute the number of cycles to execute a code block
- Given instruction statistics as well as the probabilities of various hazards, compute the CPI and execution time
- Re-computation of the CPI when the pipeline is modified
- Be able to apply Amdahl's Law

Memory and Virtual Memory

- Describe (with examples) spatial and temporal locality
- Describe how a set associative cache works, and by extension fully associative and direct mapped caches
- Address breakdown for all caches
- DRAM organization and computation of miss penalties
- Computation of the impact of cache misses on CPI
- Multilevel cache organization and operation
- Write-through vs. writeback design – how are they different and how does each one impact the CPI?
- Apply an address sequence to a multilevel cache hierarchy
- Compute the size of the page table
- Operation of the TLB on a miss
    - When the referenced page is in memory
    - When the referenced page is not in memory
- Operation of the complete memory hierarchy – from TLB down to DRAM through a multilevel cache hierarchy

I/O

- Distinguish between different types of storage devices
- Operation of RAID devices
    - Different types of RAID devices: basic operation
    - Different types of RAID devices: relative advantages
- Basic operating principles of buses

Energy

Demonstrate an understanding of the following concepts in terms of being able to describe them, or distinguish between them.

- Different types of power dissipation
- Power vs. energy
- Different ways to control power dissipation
    - Clock gating
    - Power gating
    - Dynamic voltage frequency scaling
- What Dennard Scaling?
- What is Moore's Law?
- Distinguish between P-states and C-states
- What is TDP and how is it calculated?

Parallelism

Demonstrate an understanding of the following concepts in terms of being able to describe them, or distinguish between them.

- Strong vs. weak scaling
- Amdahl's Law
- ILP, TLP, and DLP
- SIMD vs. MIMD vs. Vector
- Forms of multithreading: fine grained, coarse grained, simultaneous
- Cache coherence
    - Show an example of the cache coherence problem
    - Given a sequence of operations on a shared cache, determine if this will lead to incorrect operation. This is really a matter of understanding how caches and buses work.
- Shared memory vs. message passing machines