Computer Organization 2 Spring 2019

Midterm 2 Study Guide

April 4, 2019

The test consists of

- 1. Multiple choice questions $15 \ge 2 = 30$ points
- 2. Pipeline simulation $2 \ge 10 = 20$ points
- 3. Performance Analysis Problem 15 points
- 4. Fill out a branch prediction table 10 points
- 5. Reorder instructions 10 points
- 6. Short answer questions $5 \ge 6 = 30$ points
- You will have an opportunity to earn 15 extra credit points.
- Please try and attempt all questions. You get points for trying.
- Anything from the slides/homework is fair game.
- Making me laugh might gain you points (depends on the quality of the joke).

Topics to study

• Performance

- Latency and throughput
 - * Definitions
 - * Know how simple changes to the computer affect these metrics.
- Performance equation. Be able to compare the performance of two machines.
- Processor concepts: clock cycles, clock period (clock rate), CPI.
- Classic CPU performance equation.
- Amdahl's Law.
- The power wall and the switch from single to multi-core processors.

• Multi-Cycle Datapath and Control

- General steps performed by all instructions.
- Datapath changes: single memory unit, single ALU unit, multiple temporary registers (IR, MDR, A, B, ALUout).
- New control signals: IorD, ALUSrcA, ALUSrcB, IRWrite, PCWrite, PCWriteCond, PC-Source

- What are the actions taken on each of the following steps (be able to trace datapath for any instruction during any step and set control signals):
- Instruction Fetch
- Instruction Decode + Reg Fetch
- Execution (for all instruction types)
- Mem Access/ R-type completion
- Read Completion
- How many cycles does each instruction require?
- Advantages and Disadvantages of multi-cycle. Be able to explain how it differs from single-cycle.

• Pipelining

- Laundry analogy
- Effect on throughput and instruction latency.
- What is the ideal speedup? What factors prevent us from achieving ideal speedup?
- What happens in each of the following stages? IF, ID, EX, MEM, WB
- What are the datapath elements of a 5-stage pipeline? Note the addition of IF/ID, ID/EX, EX/MEM, and MEM/WB pipeline registers.
- Be able to calculate speedup for a particular program vs single- or multi-cycle.
- Define dependencies, hazards, and stalls.
- Be able to explain and give an example of structural, data, and control hazards.
- Solutions to data hazards: forwarding and stalls. When can forwarding be done without a stall? When is a stall required?
- Solutions to control hazards: stalling and prediction methods.
- Be able to identify dependencies and hazards within MIPS code snippets. Be able to identify what the solution should be to resolve the issue and practice reordering instructions to avoid hazards.

• Pipelining Datapath and Control

- Know datapath elements, control signals, and pipeline registers for pipeline which does not handle hazards. Be able to take any instruction and chart its path through the pipeline across 5 cycles as well as indicate control signal values.
- What information needs to be carried with the instruction? What are the fields of each pipeline register?
- Know how control lines are grouped into EX, MEM, and WB and passed into pipeline registers.
- If you understand all aspects of example walkthrough at the end of the slides, you are good. Make sure you understand everything happening during each cycle!

• Pipelining Hazards

- Data hazards: forwarding and stalling
- Know the data hazard conditions listed on slide 12, as well as how to classify any data hazards into one of these 4 conditions. Does this make sense to you intuitively?
- Know full data hazard conditions listed on slides 17 and 18 (revised on slide 29). Be able to explain in words each part of these equations and why/how these equations identify data hazards.

- Forwarding unit: what are input and outputs? Understand conditions for values of ForwardA and ForwardB (outlined on slide 31). Do NOT memorize these you should be able to reconstruct this table with your understanding rather than memorization.
- Know to stall immediately after load word with a dependent subsequent instruction (equation on 34).
- How do we implement a stall?
- Control hazards: be able to describe the problems in words.
- Solutions: assume not taken, reduce delay by moving branch into ID stage, or prediction. Explain each approach. What are advantages/disadvantages of each approach?
- How does dynamic branch prediction work?
- 1 and 2 bit branch prediction buffers how are they implemented? How do they work? Be able to explain behavior of code using these prediction buffers.
- Branch target buffer how does it work? How does it improve performance when used with branch prediction buffer?
- Be comfortable with final datapath and control. Given an instruction and a stage, be able to explain what is happening for that instruction in that stage as well as any hazard control activities that are happening.
- Project 2

• Advanced ILP

- What are exceptions?
- Difficulties with handling exceptions in pipeline: can occur out of order + multiple exceptions can occur in same clock cycle (be able to give examples).
- What does it mean to support precise exceptions?
- What are the basic steps to handle an exception in a pipeline? Know use of EPC and Cause registers.
- Multiple cycle operations: why are many arithmetic operations (mult, FP ops, etc) not performed in one cycle?
- Definitions: use latency and initiation interval. Be able to calculate these for instruction type.
- Understand modified pipeline of slide 13 using pipelined mult and FP add, but unpipelined division.
- Properties of multiple cycle operations (slide 15).
- Short answer: What is superpipelining? Multiple issue? Dynamic scheduling? Out of order execution processors?

• Memory Heirarchy

- Desirable properties of memory: quick access and large size.
- Temporal and Spatial Locality
- General Memory Hierarchy Concepts
- Four technologies used in memory.
- Memory Hierarchy Terms

Some Sample Questions

1. At the end of which pipeline stage is the data loaded from memory AVAILABLE for use?

- (a) EX
- (b) WB
- (c) MEM
- (d) ID

2. Which of the following is not a part of pipelined datapath?

- (a) IorD control signal
- (b) Left shift logic for branches
- (c) PC+1 adder hardware
- (d) Forwarding unit
- 3. Look at the following MIPS code and fill out the Branch Prediction Table. Initially, populate them with the PC address of the branches, PC+4 in the branch target and WNT for the state. Then, show the changes to the table as we run through the code. At the end, specify the number of mispredicted branches.

	.text			
	addi	\$s1,\$0,3		
	addi	\$t0,\$0,0		
	ori	\$s0,\$0,44		
	beq	<pre>\$t0,\$s1,DONE</pre>		
LOOP:	lw	\$t2,0(\$s0)		
	addi	\$t2,\$t2,1		
	SW	\$t2,0(\$s0)		
	addi	\$s0,\$s0,4		
	addi	\$s1,\$s1,-1		
	bne	\$t0,\$s1,LOOP		
DONE:	halt			
	.data			
DATA1:	.word	1 20		
DATA1:	.word	l 13		
DATA1:	.word	l -8		

PC	Branch Target	State

PC	Prediction	Action

4. Reorder the following MIPS instructions to mitigate the data hazards. Please make sure not to introduce new hazards or modify the original flow of control. You do not have to worry about actual values in the register for the first instruction. Assume they have all been 0'd out. Also, you are unaware about the availability of forwarding hardware, so you would have to work on the assumption that it is not available. If it is not possible to eliminate all hazards, try and reduce the number of stalls.

```
addi $s0,$0,10
addi $s1,$0,15
slt $t0, $s1, $s0
sub $t3,$t1,$t2
beq $t0, $zero, L1
sub $s0, $s0, $s1
j L3
L1: addi, $t1,$t1,1
slt $t0 $s0, $s1
beq $t0, $zero, L2
sub $s1, $s1, $s0
j L3
L2: add $s2, $zero, $s0
L3:
```

- 5. Why do we need sign-extension logic? How does it work?
- 6. What is a structural hazard? Suggest some ways of dealing with structural hazards?
- 7. The following table shows the number of instructions for a program.

Artihmetic/Logic	Load	Store	Branch	Jump	Total
200	120	50	50	80	500

- (a). Assuming that arithmetic and logic instructions take 1 cycle, load and store take 5 cycles each, and branches and jumps 3 cycles each, what is the execution time of this program in a 2.5 GHz processor?
- (b). What is the CPI for the program?
- (c). If the number of load instructions can be reduced by one third, what is the speedup and the new CPI?
- (d). If the number of jump instructions is doubled, is the resulting reduction in speed and the new CPI?
- 8. Consider the following instructions. Draw out the pipeline diagram, indicating the cycle in which each instructions stages are executed. Assume that forwarding is used to avoid stalls, when possible. Draw lines between stages to indicate forwarded values. If necessary, stalls may be indicated with an S or a shaded box. You only need to show one run through these lines of code. Assume the branch is NOT TAKEN at the bne. This is a loop, but you don't have to go on until the program halts. You can also assume the registers contain some valid data.

.text LOOP add \$t0,\$t1,\$t2 addi \$t3,\$t0,-100 nor \$t5,\$t4,\$t3 lw \$s1,4(\$t0)

ori	\$s0,\$s1,10
sll	\$s2,\$s0,2
lw	\$s4,4(\$s2)
bne	\$s4,\$s2,LOOP
SW	\$s2,0(\$s0)
.data	
.word	4
	ori sll lw bne sw .data .word