CS 351 Exam 2, Fall 2013

Your name: _____

Rules

- You may use one handwritten 8.5 x 11" cheat sheet (front and back). This is the only resource you may consult during this exam.
- Include explanations and comments in your answers in order to maximize your partial credit. However, you will be penalized for giving extraneous incorrect information.
- You may use the backs of these pages if you need more space, but make it clear where to find your answer to each question.
- Unless otherwise specified, you do not need to work out the arithmetic on math problems. Just do enough algebra to set up an answer of the form: Answer = [arithmetic expression] [units]

Grade (instructor use only)

	Your Score	**Max Score**
Problem 1: Single-cycle datapath		30
Problem 2: Extending the datapath		20
Problem 3: Pipeline performance		20
Problem 4: Data hazards		15
Problem 5: Control hazards		15
Total		100

Problem 1: Single-cycle datapath (30 points)

Answer the following questions about the single-cycle datapath. This is the datapath that executes a single instruction, from start to finish, on every clock cycle – the diagram is attached. Be sure to explain your answers if you want to receive partial credit.

Consider the execution of the instruction

Assembly code: sw $t0, 8($t1)
Instruction format: opcode / rs / rt / imm
Machine code: 101011 01001 01000 0000 0000 0000 1000

Provide the exact numerical answer to each question if possible. If not, describe the value without stating an exact number.

---

a) [2 points] What is the input to the control unit for this instruction?

b) [4 points] What values does this instruction send to the *Read Register 1* and *Read Register 2* ports?

c) [4 points] What values end up on the *Read data 1* and *Read data 2* ports?

d) [6 points] What operation does the ALU do this cycle? What are its exact operands?

e) [4 points] What gets put on the register file's *Write register* port? What data gets put on the register file's *Write data* port?

f) [4 points] What address goes to the memory's address port? What data goes to the memory's input data port?

g) [2 points] How is the next PC computed?

h) [4 points] Which of your answers to parts (a)-(g) are eventually discarded? That is, which of these answers do NOT have an impact on the state of the processor?

Problem 2: Extending the datapath (20 points)

You are modifying the single-cycle datapath to support a new instruction called PUSH. Here is what PUSH will do:

- Subtracts 4 from $sp to adjust the stack pointer.
- Store the data in some register to the top of the stack (the current value in $sp minus 4)

PUSH will be an R-format instruction, encoded as follows:

- Opcode = 111111
- $rt = the register whose value will be pushed to the top of the stack

---

A. [10 points] State the value of each control signal for the PUSH instruction, and briefly explain.

   If the value of a control signal does not matter, use an X (don't care). You will lose points if you use a 1 or a 0 for a signal whose value does not matter.

MemWrite:

MemRead:

RegWrite:

Branch:

Jump:

RegDst:

MemtoReg:

ALUSrc:

B. [10 points] Describe *in detail* any hardware you would need to add to the existing datapath to implement this instruction, including any new control signals.

   If you are adding any muxes, please draw them below with their inputs and outputs clearly labeled. It should be 100% clear where each mux links up with the attached datapath diagram.

Problem 3: Pipelining and performance (20 points)

Assume that the stages of a MIPS instruction's execution take the following amounts of time:

Instruction fetch: t_A ps
Instruction decode / register read: t_B ps
Execution: t_C ps
Memory: t_D ps
Write back: t_E ps

A. [5 points] In terms of the times given above, what is the minimum clock cycle time of a ***single-cycle*** implementation of this processor? How long (in seconds, not cycles) will a single instruction take to execute on this implementation?

B. [5 points] What is the minimum clock cycle time of a version of this processor with a 5-stage pipeline? How long (in seconds, not cycles) will a single instruction take to execute on this implementation?

C. [5 points] In order to maximize the performance of the pipelined implementation, what should be true about the relationships between t_A, t_B, t_C, t_D, and t_E, and why?

D. [5 points] A snippet of code takes 1,000,000 cycles to run on the single-cycle implementation and 1,000,010 cycles to run on the pipelined implementation. Can we conclude that the single-cycle implementation is slightly faster? Explain.

Problem 4: Data hazards (15 points)

[5 points]
Show two snippets of code: one that has a data hazard, and one that doesn't. Explain.

[5 points]
Consider your snippet of code with the data hazard. If the processor does not allow data forwarding, show a pipeline diagram of its execution.

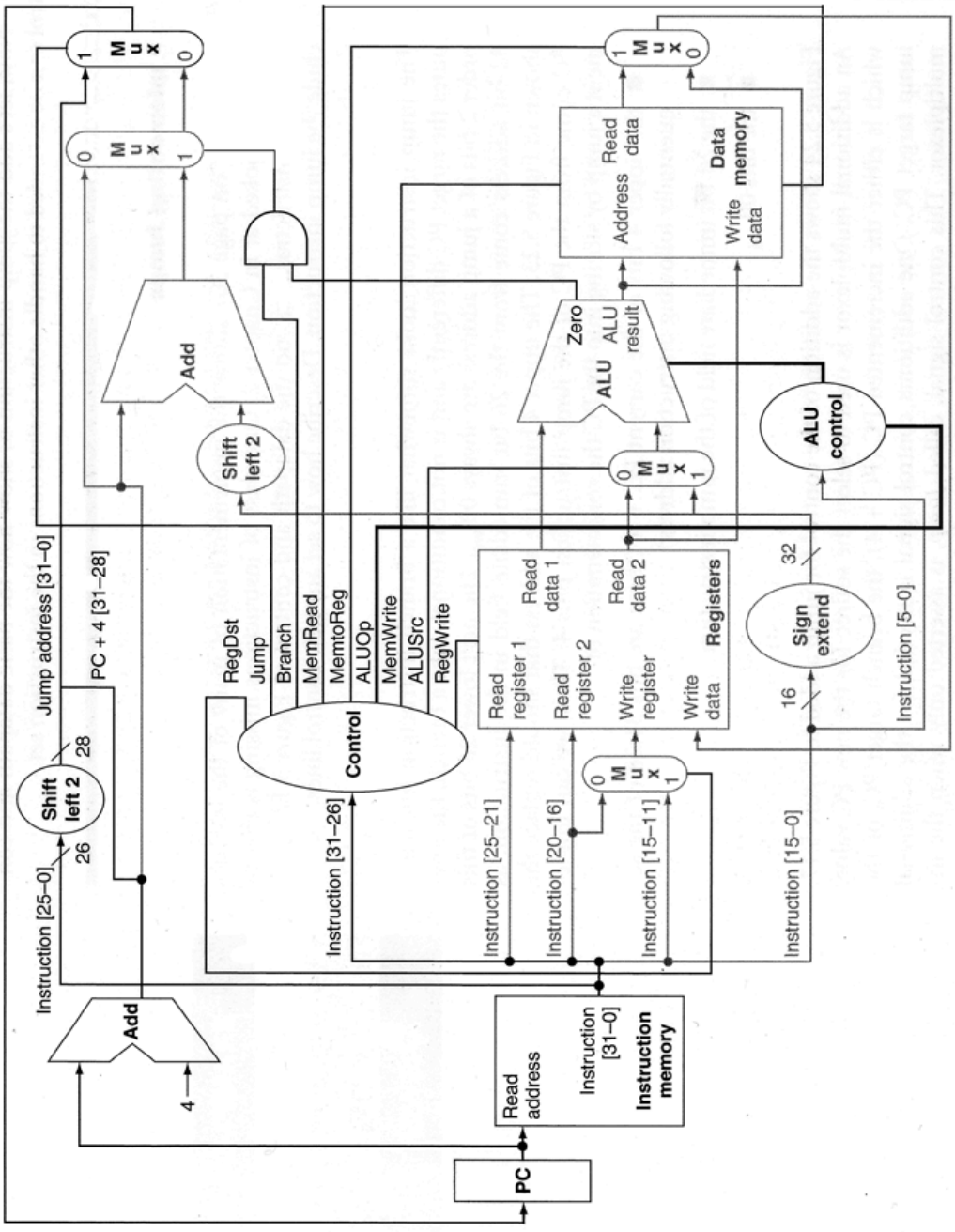
[5 points]
If the processor does have data forwarding, which forwarding path (from which stage to which stage) will your snippet use? Draw a revised pipeline diagram.

Problem 5: Control hazards and branch prediction (15 points)

[5 points]
In what pipeline stage does branch prediction take place? What is the input to the branch prediction unit?

[10 points]
Trace the ***long-term*** accuracy of the 1-bit and 2-bit predictors for the following repeated branch sequence:
T, T, NT, NT, T, T, NT, NT

If the initial state of either predictor affects the long-term accuracy, explain. (Hint: start by picking some initial state for the first iteration of this inner loop and seeing what happens going into the second iteration.)


M u x
Add
Shift left 2
Jump address [31–0]
PC + 4 [31–28]
Instruction [25–0]
Shift left 2
28
26
Add
4
PC
Read address
Instruction [31–0]
Instruction memory
RegDst
Jump
Branch
MemRead
MemtoReg
ALUOp
MemWrite
ALUSrc
RegWrite
Control
Instruction [31–26]
Instruction [25–21]
Instruction [20–16]
Instruction [15–11]
Instruction [15–0]
Read register 1
Read register 2
Write register
Write data
Read data 1
Read data 2
Registers
16
32
Sign extend
Instruction [5–0]
ALU
Zero
ALU result
ALU control
Address
Read data
Write data
Data memory