

Computer Architecture and Engineering

CS152 Quiz #1

February 19th, 2008

Professor Krste Asanovic

Name: _____

This is a closed book, closed notes exam.

80 Minutes

10 Pages

Notes:

- Not all questions are of equal difficulty, so look over the entire exam and budget your time carefully.
- Please carefully state any assumptions you make.
- Please write your name on every page in the quiz.
- You must not discuss a quiz's contents with students who have not yet taken the quiz. If you have inadvertently been exposed to the quiz prior to taking it, you must tell the instructor or TA.
- You will get no credit for selecting multiple choice answers without giving explanations if the instructions ask you to explain your choice.

Writing name on each sheet	_____	1 Point
Question 1	_____	28 Points
Question 2	_____	33 Points
Question 3	_____	10 Points
Question 4	_____	8 Points
TOTAL	_____	80 Points

NAME: _____

Problem Q.1: Microprogramming Bus-Based Architectures

[28 points]

In this problem, we explore microprogramming by writing microcode for the bus-based implementation of the MIPS machine described in Handout #1 (Bus-Based MIPS Implementation), which we have included at the end of this quiz for your reference. In order to further simplify this problem, *ignore* the busy signal, and assume that the memory is as fast as the register file. The final solution should be elegant and efficient.

You are to implement in microcode a double indirect addressing mode, as described below. In this addressing mode, the source register contains a pointer to a location in memory whose value is a pointer to the location in memory whose value is to be loaded. The instruction has the following format:

LWmm r_d, r_s

LWmm performs the following operation:

$r_d \leftarrow M[M[r_s]]$

Fill in Worksheet Q1-1 with the microcode for LWmm. Use *don't cares* (*) for fields where it is safe to use don't cares. Study the hardware description well, and make sure all your microinstructions are legal.

Please comment your code clearly. If the pseudo-code for a line does not fit in the space provided, or if you have additional comments, you may write in the margins as long as you do it neatly. Your code should exhibit “clean” behavior and not modify any registers (except r_d) in the course of executing the instruction.

Finally, make sure that the instruction fetches the next instruction (i.e., by doing a microbranch to FETCH0 as discussed in the Handout).

NAME: _____

State	PseudoCode	ld IR	Reg Sel	Reg W	en Reg	ld A	ld B	ALUOp	en ALU	ld MA	Mem W	en Mem	Ex Sel	en Imm	μ B r	Next State
FETCH0:	MA <- PC; A <- PC	0	PC	0	1	1	*	*	0	1	*	0	*	0	N	*
	IR <- Mem	1	*	*	0	0	*	*	0	0	0	1	*	0	N	*
	PC <- A+4	0	PC	1	1	0	*	INC_A_4	1	*	*	0	*	0	D	*
...																
NOPO:	microbranch back to FETCH0	0	*	*	0	*	*	*	0	*	*	0	*	0	J	FETCH0
LWMM0:																

Problem Q2: Dual ALU Pipeline

[33 points]

In this problem we consider further improvements to the fully bypassed 5-stage MIPS processor pipelines presented in Lecture 3 and Problem Set 1. The pipeline introduced in Problem Set 1 had a modified 3rd stage (address calculation) and a modified 4th stage (parallel data memory and ALU). In this new pipeline we essentially replace the Adder introduced in Problem Set 1 with a proper ALU, with the goal of eliminating all hazards (see Figure 2-A).

The Dual ALU Pipeline has two ALUs: ALU1 is in the 3rd pipeline stage (EX1) and ALU2 is in the 4th pipeline stage (EX2/MEM). A memory instruction always uses ALU1 to compute its address. An ALU instruction uses either ALU1 or ALU2, but never both. *If an ALU instruction's operands are available (either from the register file or the bypass network) by the end of the ID stage, the instruction uses ALU1; otherwise, the instruction uses ALU2.*

In this problem, assume that the control logic is optimized to stall only when necessary, and that the pipeline is fully bypassed. *You may ignore branch and jump instructions in this problem.*

IF	ID	EX1	EX2/MEM	WB
Instruction fetch	Instruction decode and register read	ALU1 execution and address calculation	ALU2 execution and memory access	Writeback to register file

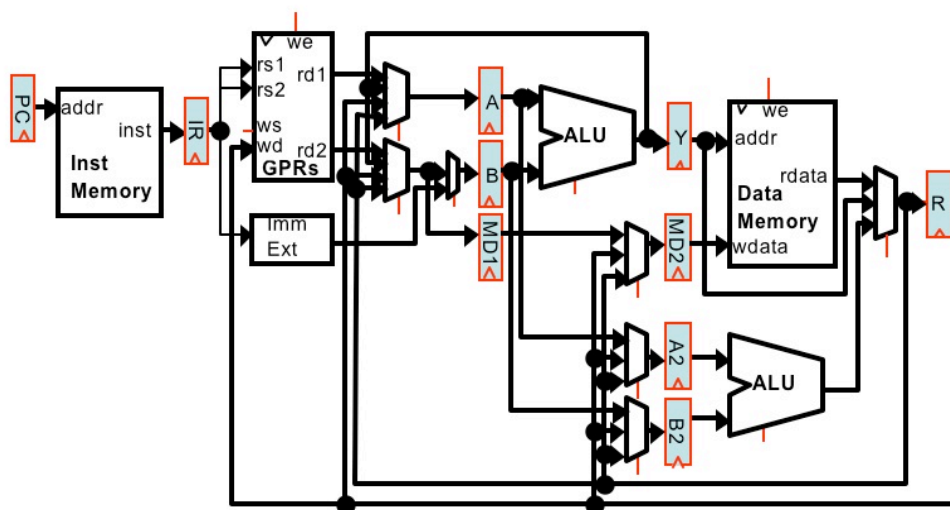


Figure 2-A. Dual ALU Pipeline

NAME: _____

Problem Q.2.A

ALU Usage
[15 points]

For the following instruction sequence, **indicate which ALU each add instruction uses**. Assume that the pipeline is initially idle (for example, it has been executing nothing but nop instructions). Registers involved in inter-instruction dependencies are highlighted in bold for your convenience.

	ALU1 or ALU2?
add r1 , r2, r3	
lw r4 , 0(r1)	-----
add r5 , r4 , r6	
add r7, r5 , r8	
add r1 , r2, r3	
lw r4, 0(r1)	-----
add r5, r1 , r6	

NAME: _____

Problem Q.2.B

Instruction Sequences Causing Stalls

[18 points]

Indicate whether each of the following instruction sequences causes a stall in the pipeline and a short summary of the reason why. Consider each sequence *separately* and assume that the pipeline is initially idle (for example, it has been executing nothing but nop instructions). Registers involved in inter-instruction dependencies are highlighted in bold for your convenience.

Stall? Reason?
(yes/no)

add r1 , r2, r3 lw r4, 0(r1)		
lw r1 , 0(r2) add r3, r1 , r4 lw r5, 0(r1)		
lw r1 , 0(r2) lw r3, 0(r1)		
lw r1 , 0(r2) sw r1 , 0(r3)		
lw r1 , 0(r2) add r3 , r1 , r4 sw r5, 0(r3)		
lw r1 , 0(r2) add r3, r1 , r4		

NAME: _____

Problem Q3: ISA Compatibility (Short Yes/No Questions)

[10 points]

The following questions describe two variants of a processor which are otherwise identical. In each case, circle "Yes" if the variants might generate different results from the same compiled program, and circle "No" otherwise. You must also **briefly explain** your reasoning and any assumptions you are making. *Ignore differences in the time taken by each machine to execute the program.*

Problem Q3.A

Interlock vs. Bypassing

Pipelined processor A uses interlocks to resolve data hazards while pipelined processor B has full bypassing.

Yes / No

Problem Q3.B

Delay Slot

Pipelined processor A uses branch delay slots to resolve control hazards while pipelined processor B kills instructions following a taken branch.

Yes / No

Problem Q3.C

Structural Hazard

Pipelined processor A has a single memory port used to fetch instructions and data, while pipelined processor B has no structural hazards.

Yes / No

Problem Q3.D

Microcode size

Microcoded machine A uses 32-bit microcode instructions, while microcoded machine B uses 64-bit microcode instructions.

Yes / No

Problem Q3.E

Register size

Microcoded machine A has 32-bit data registers, while microcoded machine B has 64-bit data registers.

Yes / No

NAME: _____

Problem Q.4: Iron Law of Processor Performance (Short Answer)

[8 points]

Mark whether the following modifications will cause each of the categories to **increase**, **decrease**, or whether the modification will have **no effect**. **Explain your reasoning** to receive credit.

	Instructions / Program	Cycles / Instruction	Seconds / Cycle	Reasoning?
Combining two pipeline stages				
Removing a complex instruction				
Running the machine at a higher clock frequency				
Using a better compiler				

NAME: _____

END OF QUIZ MATERIAL

This page intentionally left blank.
The following pages are a replica of Handout #1 (Bus-Based MIPS Implementation)