CSEE 3827: Fundamentals of Computer Systems

Final Exam

May 11, 2009

Name: ______________________________ UNI: __________________

Read all of the following information before starting the exam:

- Be sure to write your name on each page of the exam.
- Use the exam itself for your solutions (no blue books or spare sheets of paper). You may use the backside of pages if you need more space.
- Show your work in order to earn partial credit.
- You may use your textbook and class notes, but *absolutely no electronic devices (laptops, cell phones, etc.)*
- Good luck!

Problem	Point Value	Points Earned
1	6	
2	6	
3	10	
4	10	
5	9	
6	15	
Total	56	

Name: ______________________________

1. Suppose you have a machine which executes a program consisting of 50% multiply instructions, 20% divides, and the remaining 30% are other instructions.

   (a) Management wants the machine to execute these programs 4 times faster. You can make the divide instructions at most 3 times faster and the multiplies at most 8 times faster. Can you meet managements goal by making only one improvement? If so, which one?

   (b) New management takes over the company and decides to make both improvements: to multiplies and divides. What is the speed of the improved machine relative to the original machine?

2. The design team for a processor is choosing between a pipelined or non-pipelined implementation. Here are some design parameters for the two possibilities:

	Pipelined Version	**Non-Pipelined**
Clock Rate	500 MHz	350MHz
CPI for ALU instructions	1	1
CPI for Control instructions	2	1
CPI for Memory instructions	2.7	1

(a) For a program with 20% ALU instructions, 10% control instructions, which design will be faster? Give a quantitative CPI average for each case.

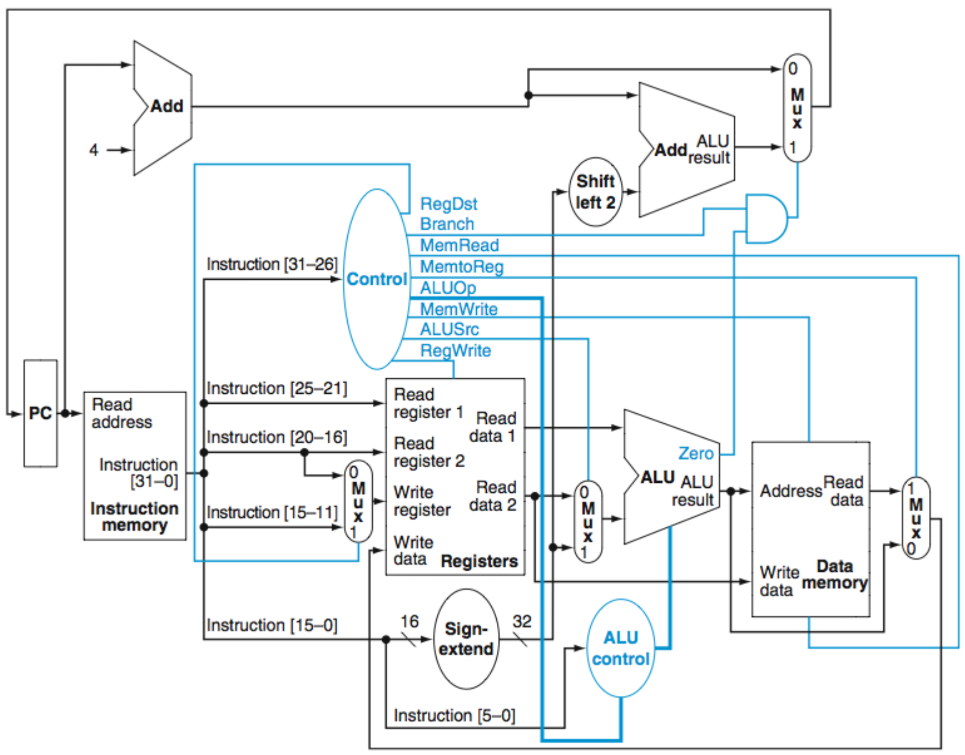
(b) For a program with 80% ALU instructions, 10% control instructions and 10% memory instructions, which design will be faster? Give a quantitative CPI average for each case.

Name: ____________________

3. Write the MIPS instructions to implement a function: `oddzeros`. This function takes a single register argument (*arg* below) and returns a result (*ret* below) in which every other bit of the argument has been set to zero. Be sure your implementation of `oddzeros` respects MIPS procedure calling conventions.

   $arg = a_{31}a_{30}a_{29}a_{28}a_{27}a_{26}...a_2a_1a_0 \Rightarrow ret = 0a_{30}0a_{28}0a_{26}...a_20a_0$

4. In this problem you will enhance the single-cycle MIPS processor that we have developed to support the `jr` instruction. To do this you should extend the baseline datapath and control shown below by adding to the diagram or tables as necessary. Here is some information about the instruction. `jr` is an R-type instruction (`opcode=000000,funct=001000`) which jumps to the address specified by register rs ($PC \leftarrow Reg[rs]$).

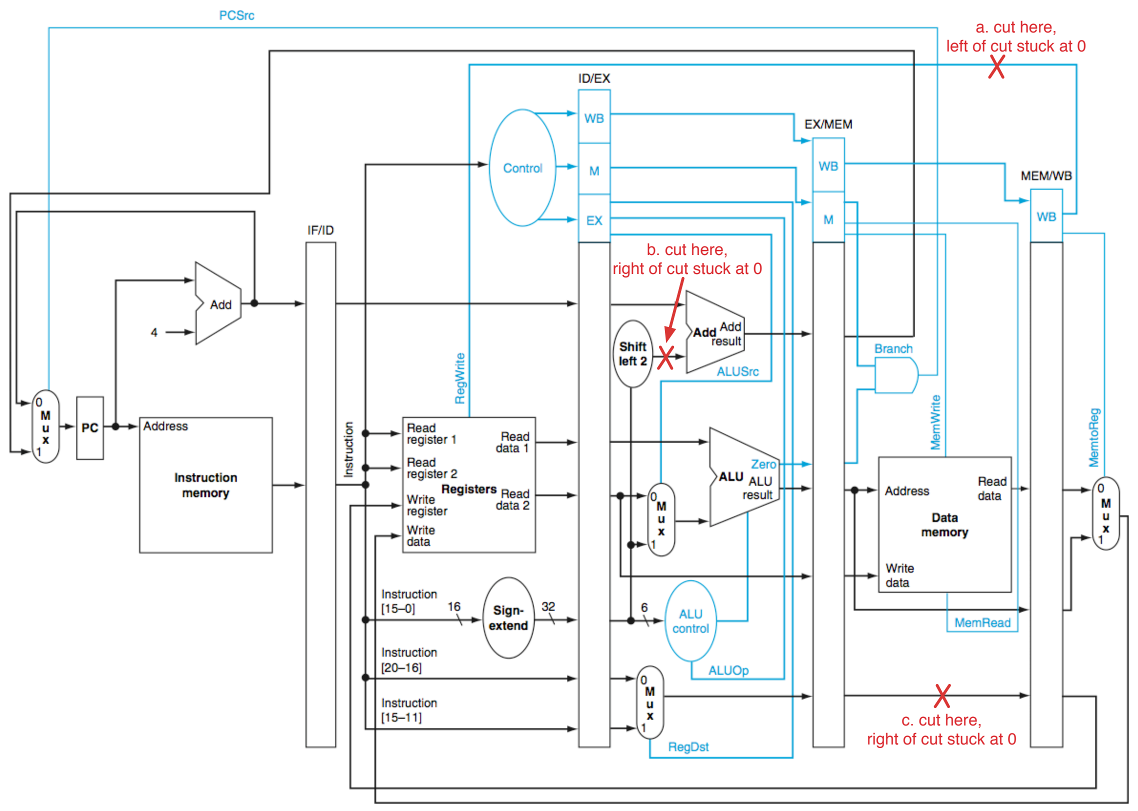


Instruction	RegDst	ALUSrc	Memto-Reg	Reg-Write	Mem-Read	Mem-Write	Branch	ALUOp1	ALUOp0
R-format	1	0	0	1	0	0	0	1	0
lw	0	1	1	1	1	0	0	0	0
sw	X	1	X	0	0	1	0	0	0
beq	X	0	X	0	0	0	1	0	1

Instruction opcode	ALUOp	Instruction operation	Funct field	Desired ALU action	ALU control input
LW	00	load word	XXXXXX	add	0010
SW	00	store word	XXXXXX	add	0010
Branch equal	01	branch equal	XXXXXX	subtract	0110
R-type	10	add	100000	add	0010
R-type	10	subtract	100010	subtract	0110
R-type	10	AND	100100	AND	0000
R-type	10	OR	100101	OR	0001
R-type	10	set on less than	101010	set on less than	0111

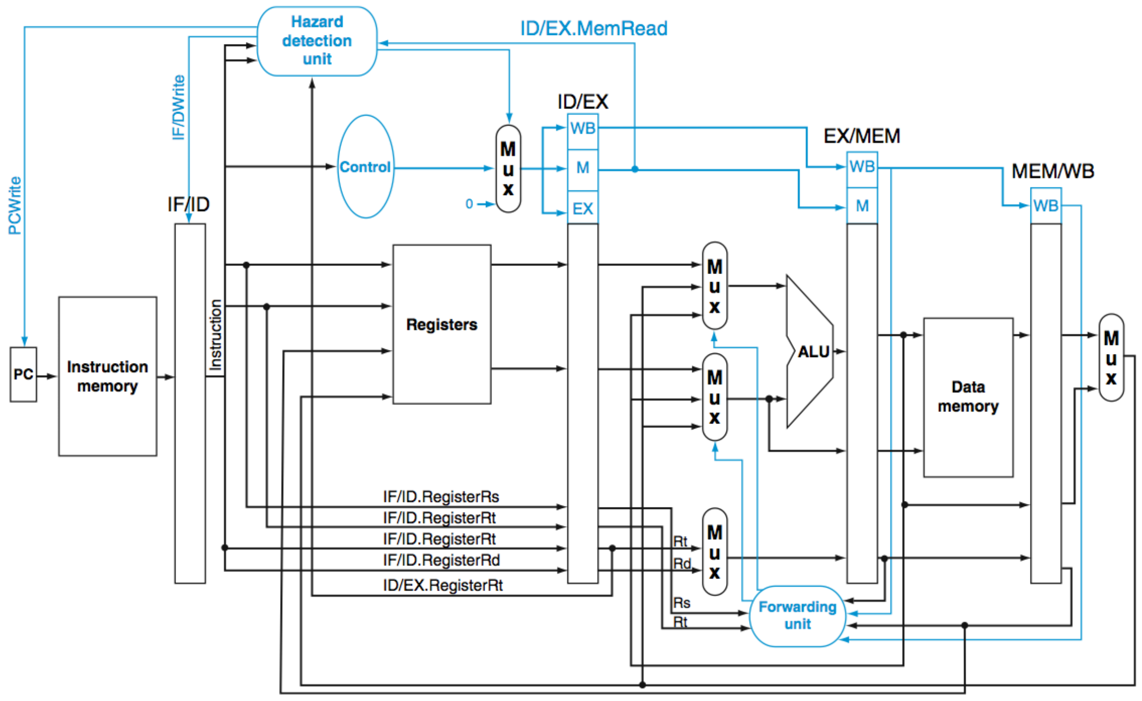
5. In the datapath shown below, three wires are marked with X. For each one:

- Briefly describe in words the negative consequence of cutting this line relative to the working, unmodified processor.
- Provide a snippet of code that will fail.
- Provide a snippet of code that will still work.



Name: ____________________________

6. Consider the MIPS 5-stage pipeline shown below.



(a) What is the load use latency for this MIPS pipeline?

(b) Assuming the pipeline above identify whether the value for each register operand in the code below is coming from the bypass or from the register file. For clarity, please write REG or BYPASS in each box.

Instruction	Source Operand 1	Source Operand 2
`add $t2, $s1, $sp`		
`lw $t1, $t1, 0`		N/A
`addi $t2, $t1, 7`		N/A
`add $t1, $s2, $sp`		
`lw $t1, $t1, 0`		N/A
`addi $t1, $t1, 9`		N/A
`sub $t1, $t1, $t2`		

(c) How many cycles will the program take to execute?

(d) Assume, due to circuit constraints, that the bypass wire from the memory stage back to the execute stage is omitted from the pipeline. What is the load-use latency for this modified pipeline?

(e) How long does the program take to execute on the modified pipeline?