CMPE 110 Computer Architecture

Fall 2016, Homework #2

Computer Engineering

UC Santa Cruz

October 18, 2016

Name: _____

Email: _____

Submission Guidelines:

- This homework is due on **Wednesday, November 2nd, 2016 11:59 pm.**
- The homework must be submitted to eCommons by 11:59pm.
    - Anything later is a late submission
- Please write your name and your UCSC email address
- Provide details on how to reach a solution. An answer without explanation gets no credit. Clearly state all assumptions.
- The homework should be "readable" without too much effort
    - The homework must be typed and submitted as a single file in PDF format
    - Please name your homework file cmpe110-hw2-yourcruzid.pdf
    - Please keep your responses coherent and organized or you may lose points.
- Points: 64 = 16 + 16 + 16 +16

Question	part 1	part 2	part 3	part 4	part 5
1			-	-	-
2					
3					-
4				-	-
Total					

By default, the pipeline stalls at decode stage with data hazards.

Question 1. Datapath (16 Points)

In this problem, you will modify the single-cycle datapath we built up in class to implement JAL instruction. The MIPS jump and link instruction, JAL, is used to support procedure calls by jumping to jump address (similar to j) and saving the address of the following instruction PC+4 in register $ra ($31), i.e.,

JAL: R[31] ← PC + 4
PC ← Jump Address

JAL uses the j instruction format:

JAL *address*
op (6 bits) | Target address (26 bits)

1. Add to the provided datapath the necessary data and control signals to implement the JAL instruction. Draw and label all components and wires very clearly (give control signals meaningful names; if selecting a subset of bits from many, specify exactly which bits are selected; and so on). Justify the need for the modifications, if any.

2. Specify control line values for this instruction.

Question 2. Pipelining (16 Points)

Given the following code:

```
        xor r0, r0, r0
        addiu r1, r0, 10
        j L1
loop:   lw r3, 0(r2)
        mul r4, r3, r3
        mul r3, r3, r1
        addiu r0, r0, 1
        div r3, r4, r3
        sw r3, 0(r2)
        addiu r2, r2, 4
L1:     bne r0, r1, -8
```

Calculate the number of cycles it takes to execute one iteration of the loop (including the bne instruction) on the following models:

2.1 A non-pipelined machine

2.2 A pipelined machine with scoreboarding and five adders and five multipliers without data forwarding

2.3 A pipelined machine with scoreboarding and five adders and five multipliers with data forwarding.

2.4 A pipelined machine with scoreboarding and one adder and one multiplier without data forwarding

2.5 A pipelined machine with scoreboarding and one adder and one multiplier with data forwarding

Note: For all machine models, use the basic instruction cycle as follows:

- Fetch (one clock cycle)
- Decode (one clock cycle)
- Execute (MUL takes 6, ADD takes 4 clock cycles, other instructions 1 clock cycle). The multiplier and the adder are NOT pipelined.
- Memory access (one clock cycle)
- Write-back (one clock cycle)

Do not forget to list any assumptions you make about the pipeline structure (e.g., how is data forwarding done between pipeline stages).

Question 3. Branch prediction (16 Points)

Assume a machine with a 7-stage pipeline. Assume that branches are resolved in the sixth stage. Assume that 30% of instructions are branches.

3.1 How many instructions of wasted work are there per branch mis-prediction on this machine?

3.2 Assume 1000 instructions are on the correct path of a program and assume a branch predictor accuracy of 10%. How many instructions are fetched on this machine? (Please show your work for full credit.)

3.3 Let's say we modified the machine so that it used dual path execution (where an equal number of instructions are fetched from each of the two branch paths). Assume branches are resolved before new branches are fetched. Write how many instructions would be fetched in this case, as a function of N. (Please show your work for full credit.)

3.4 Now let's say that the machine combines branch prediction and dual path execution in the following way:

A branch confidence estimator, like we discussed in class, is used to gauge how confident the machine is of the prediction made for a branch. When confidence in a prediction is high, the branch predictor's prediction is used to fetch the next instruction; When confidence in a prediction is low, dual path execution is used instead.

Assume that the confidence estimator estimates a fraction C of the branch predictions have high confidence, and that the probability that the confidence estimator is wrong in its high confidence estimation is M.

Write how many instructions would be fetched in this case, as a function of N, A, C, and M. (Please show your work for full credit.)

Question 4. Out-of-order execution (16 Points)

In this problem, we will give you the state of the Register Alias Table (RAT) and Reservation Stations (RS) for an out-of-order execution engine. Your job is to determine the original sequence of the following five instructions in program order.

ADD R3, R10, R3
MUL R1, R1, R10
MUL R11, R7, R9
ADD R1, R2, R3
ADD R5, R1, R11
ADD R7, R2, R6

The out-of-order machine in this problem behaves as follows:

- The frontend of the machine has a one-cycle fetch stage and a one-cycle decode stage. The machine can fetch one instruction per cycle, and can decode one instruction per cycle.

- The machine dispatches one instruction per cycle into the reservation stations, in program order. Dispatch occurs during the decode stage.

- An instruction always allocates the first reservation station that is available (in top-to-bottom order) at the required functional unit.

- When a value is captured (at a reservation station) or written back (to a register) in this machine, the old tag that was previously at that location is not cleared; only the valid bit is set.

- When an instruction in a reservation station finishes executing, the reservation station is cleared.

- Both the adder and multiplier are fully pipelined. Add instructions take 4 cycles. Multiply instructions take 6 cycles.

- When an instruction completes execution, it broadcasts its result, and dependent instructions can begin execution in the next cycle if they have all operands available.

- When multiple instructions are ready to execute at a functional unit, the oldest ready instruction is chosen.

Initially, the machine is empty. Six instructions then are fetched, decoded, and dispatched into reservation stations, before any instruction executes. Then, one instruction completes execution. Here is the state of the machine at this point, after the single instruction completes:

Reg	V	Tag	Value
R1	0	X	2
R2	1		10
R3	0	A	17
R4	1		9
R5	1	C	1
R6	1		15
R7	0	B	7
R8	1		20
R9	1		16
R10	1		5
R11	0	Y	3

	Src 1			Src 2		
	V	Tag	Value	V	Tag	Value
A	1	-	17	1	-	5
B	1	-	10	1	-	15
C	0	Y	-	0	X	-

ADD

	Src 1			Src 2		
	V	Tag	Value	V	Tag	Value
X	1	A	27	1	-	5
Y	0	B	-	1	-	16
Z						

MUL

4.1 Give the six instructions that have been dispatched into the machine, in program order. The source registers for the first instruction can be specified in either order. Give instructions in the following format: "opcode destination ⇐ source1, source2."

4.2 Now assume that the machine flushes all instructions out of the pipeline and restarts execution from the first instruction in the sequence above. Show the full pipeline timing diagram below for the sequence of six instructions that you determined above, from the fetch of the first instruction to the writeback of the last instruction. Assume that the machine stops fetching instructions after the sixth instruction. Use "F" for fetch, "D" for decode, "E1," "E2," "E3," "E5," "E4," and "E6" to signify the first, second, third and fourth cycles of execution for an instruction (as required by the type of instruction), and "W" to signify writeback. You may or may not need all columns shown.

Instruction	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1-																
2-																
3-																
4-																
5-																
6-																

4.3 Finally, show the state of the RAT and reservation stations after 9 cycles in the blank figures below.

Reg	V	Tag	Value
R1			
R2			
R3			
R4			
R5			
R6			
R7			
R8			
R9			
R10			
R11			

	Src 1			Src 2		
	V	Tag	Value	V	Tag	Value
A						
B						
C						

ADD

	Src 1			Src 2		
	V	Tag	Value	V	Tag	Value
X						
Y						
Z						

MUL