Computer Science 605.611

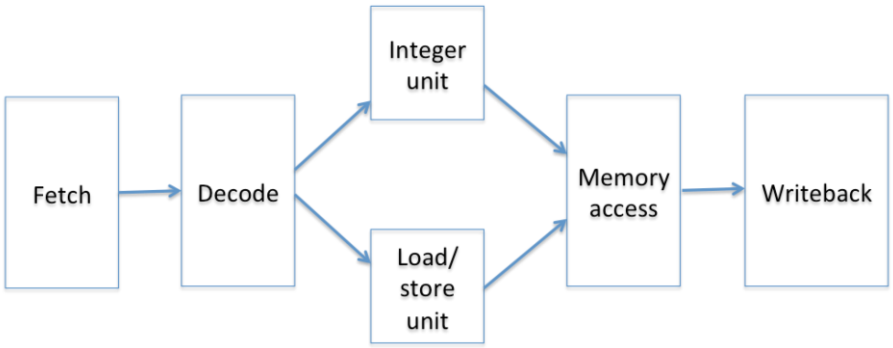
Problem Set 8

1. (5) Recall that our 5-stage MIPS pipeline is an example of a "scalar" pipeline since at most one instruction can occupy any stage at one time. Which of the following types of hazards does this pipeline never experience and why?

a) Data hazard
b) resource hazard
c) branch hazard
d) control hazard

b) resource hazard – there is only one ALU, therefore only one instruction can occupy at once. Systems with multiple ALU's may experience resource hazards if multiple of the same instruction are trying to occupy too few of a resource.

2. The diagram below represents a degree 2 superscalar version of our 5-stage MIPS pipelined system. The decode stage includes a buffer to hold instructions that have been decoded but cannot yet execute. The execute stage includes an integer unit and a load/store unit. The system allows both out-of-order execution and out-of-order completion. Each stage requires 1 clock cycle.



a) (5) How many clock cycles does the following short instruction sequence consume on this superscalar system?

1. add $5,$4,$3
2. lw $2,16($6)
3. sw $9,8($7)
4. lw $10,48($12)

int	Load/Store	cycle
add $5,$4,$3	lw $2,16($6)	1
	sw $9,8($7)	2

	lw $10,48($12)	3

3 cycles + 4 cycles to advance from fetch to writeback = 7 cycles

b) (3) How many clock cycles does the same instruction sequence consume on the original 5-stage scalar system?

cycle	Fetch	Decode	Execute	Memory	Write-Back
1	add $5,$4,$3				
2	lw $2,16($6)	add $5,$4,$3			
3	sw $9,8($7)	lw $2,16($6)	add $5,$4,$3		
4	lw $10,48($12)	sw $9,8($7)	lw $2,16($6)	add $5,$4,$3	
5		lw $10,48($12)	sw $9,8($7)	lw $2,16($6)	add $5,$4,$3
6			lw $10,48($12)	sw $9,8($7)	lw $2,16($6)
7				lw $10,48($12)	sw $9,8($7)
8					lw $10,48($12)

8 cycles

c) (3) How many read ports are required for the register file to support the superscalar system? Recall that a separate read port is required for each register input operand.

Integer operations such as add require 2 read ports, while load/store operations require just one read port. To implement both as parallel ALU's, there must be a total of 3 read ports.

d) (3) How many write ports are required for the register file to support the superscalar system? Write ports receive the register results produced by instructions.

Both integer operations and load/store operations write to one register, so we need a total of 2 write ports for this system.

3. Consider the following instruction sequence that executes on our 5-stage scalar pipelined system with no code rearrangement and no branch prediction, but with a hazard unit as well as a forwarding unit:

```
        ori   $4,$0,2
loop:   add    $6,$7,$8
        and    $9,$4,$7
        bne   $4,$0,loop          ; delayed branch
        srl    $4,$4,1
        addi   $3,$0,1
        sll    $3,$3,1
```

a) (5) The ori instruction is fetched in cycle 1. During which clock cycle does the sll instruction complete the write-back stage if no early branch condition evaluation or branch prediction is used?

cycle	Fetch	Deccode	Execute	Memory	Write-Back
1	ori $4,$0,2				
2	add $6,$7,$8	ori $4,$0,2			
3	and $9,$4,$7	add $6,$7,$8	ori $4,$0,2		
4	bne $4,$0,loop	and $9,$4,$7	add $6,$7,$8	ori $4,$0,2	
5		bne $4,$0,loop	and $9,$4,$7	add $6,$7,$8	ori $4,$0,2
6			bne $4,$0,loop	and $9,$4,$7	add $6,$7,$8
7				bne $4,$0,loop	and $9,$4,$7
8	srl $4,$4,1				bne $4,$0,loop
9	addi $3,$0,1	srl $4,$4,1			
10	sll $3,$3,1	addi $3,$0,1	srl $4,$4,1		
11		sll $3,$3,1	addi $3,$0,1	srl $4,$4,1	
12			sll $3,$3,1	addi $3,$0,1	srl $4,$4,1
13				sll $3,$3,1	addi $3,$0,1
14					sll $3,$3,1

14 cycles

b) (5) The ori instruction is fetched in cycle 1. There is no early branch condition evaluation. During which clock cycle does the sll instruction complete the write-back stage if branch prediction based on a decode history table (DHT) is used? Assume that the DHT always predicts that the branch will be taken.

cycle	Fetch	Deccode	Execute	Memory	Write-Back
1	ori $4,$0,2				
2	add $6,$7,$8	ori $4,$0,2			
3	and $9,$4,$7	add $6,$7,$8	ori $4,$0,2		
4	bne $4,$0,loop	and $9,$4,$7	add $6,$7,$8	ori $4,$0,2	
5	add $6,$7,$8	bne $4,$0,loop	and $9,$4,$7	add $6,$7,$8	ori $4,$0,2
6	and $9,$4,$7	add $6,$7,$8	bne $4,$0,loop	and $9,$4,$7	add $6,$7,$8
7	bne $4,$0,loop	and $9,$4,$7	add $6,$7,$8	bne $4,$0,loop	and $9,$4,$7
8	srl $4,$4,1	bubble	bubble	bubble	bne $4,$0,loop
9	addi $3,$0,1	srl $4,$4,1	bubble	bubble	bubble
10	sll $3,$3,1	addi $3,$0,1	srl $4,$4,1	bubble	bubble
11		sll $3,$3,1	addi $3,$0,1	srl $4,$4,1	bubble
12			sll $3,$3,1	addi $3,$0,1	srl $4,$4,1
13				sll $3,$3,1	addi $3,$0,1
14					sll $3,$3,1

14 cycles

4. (5) Our MIPS system treats all branch instructions as delayed branches. Which one of the following is the main reason for using delayed branches?

a) to reduce the number of data hazards
b) to provide time to compute the branch target address
c) to reduce the number of pipeline bubbles
d) to provide time to predict the branch behavior

Employing delayed branches is the practice of filling the delay slots with instructions that need to execute in any case. The compiler can be used to fill the delay slots, or use NOPs if instructions cannot be found. The main reason for using delayed branches would be to C) reduce the number of pipeline bubbles.

5. a) (3) Once a conditional branch instruction is brought into the pipeline, what is the earliest stage within the pipeline in which a branch prediction can be made if the prediction is based on a branch history table?

By using the incremented PC to access the BHT we can check for a hit as the instruction is fetched by the fetch stage. Therefore the prediction can be made immediately, in the first cycle.

b) (3) When the prediction for a conditional branch instruction is found in a decode history table, what is the maximum branch penalty (i.e., the maximum number of pipeline bubbles that must be inserted) if the prediction turns out to be incorrect and the instruction is a delayed branch?

There may be a maximum of 3 bubbles produced by a delayed branch.

6. Consider the following code containing nested loops:

```
Loop1:  …
        …
Loop2:  …
        …
        …
        bne   $t2,Loop2
        …
        …
        beq   $t3,Loop1
        …
```

The bne and beq instructions are the only branch instructions within the code sequence. The outer loop (Loop1) contains an inner loop (Loop2). Assume that the beq at the end of the outer loop transfers control 120 consecutive times before the loop is exited. Also assume that for each iteration of the outer loop, the bne instruction at the end of the inner loop transfers control 100 consecutive times before the inner loop exits. That is, the inner loop iterates 100 times for each of the 120 iterations of the outer loop. Branch prediction is used for both branch instructions. What is the total number of mispredictions for each branch instruction, if:

a) the prediction is based on a separate single branch prediction bit for each branch instruction?

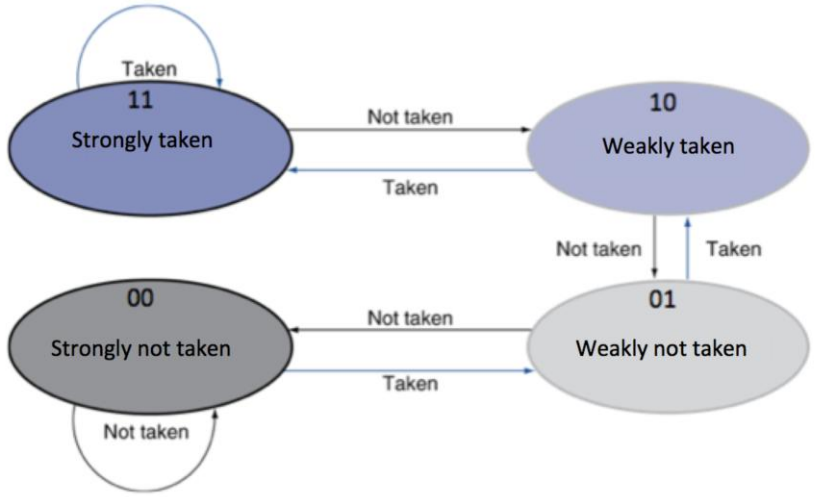
For each 100 iterations of the inner loop, it mispredicts 2 times.
For each 120 iterations of the outer loop, it mispredicts 2 times.

(3) Number of mispredictions for the bne = 2 * 120 = 240

(3) Number of mispredictions for the beq = 2

The initial value for the branch prediction bit for each branch instruction is 0 (not taken) and the bit is inverted each time that the prediction is incorrect (i.e., mispredicted).

b) the prediction is instead based on a separate pair of branch prediction bits for each branch instruction. The initial value of the branch prediction bits for each branch instruction = 00 (strongly not taken) and the bits are updated based on the diagram below:



Prediction of inner loop is wrong three times for 1st iteration of outer loop, but only wrong once for each remaining outer loop iteration.
Prediction of outer loop is wrong three times.

(5) Number of mispredictions for the bne = 3 + 119 = 122

(5) Number of mispredictions for the beq = 3

7. a) (3) What type of data hazard (RAW, WAR, WAW, or NONE), requiring a stall, is caused by the following pair of instructions on a scalar pipeline?

```
sw    $5,44($2)
add   $10,$5,$2
```

RAW

b) (3) What type of data hazard (RAW, WAR, WAW, or NONE), requiring a stall, is caused by the following pair of instructions on a scalar pipeline?

```
sw    $5,44($2)
lw    $5,44($2)
```

NONE

c) (3) What type of data hazard (RAW, WAR, WAW, or NONE) can be caused by the following two instructions if instead of the scalar system they are executed on superscalar pipeline with multiple load/store units that allows out-of-order execution?

```
sw    $5,44($2)
lw    $5,44($2)
```

WAW

8. (10) On a certain VLIW system, each long instruction word is a packet or bundle of six individual machine instructions. The hardware system contains three integer units and three floating point units. Therefore, up to 3 integer operations and 3 floating point operations can be performed in parallel in the same clock cycle. However, due to the mix of available instructions and possible dependencies, the system may not be able to make use of all the execution units in every cycle. In that case, one or more nop instructions must be inserted into the instruction bundle (i.e., into the long instruction word) to fill any unused slots.
Using the following format to show the instructions within each long instruction word:

Int unit1	Int unit2	Int unit3	Flt unit1	Flt unit2	Flt unit3

List the contents of the all long instruction words or bundles that are required for the following group of instructions using the minimum number of nop instructions:

```
add     $11,$2,$3
add     $4,$5,$11
mtc1    $4,$f12
cvt.s.w $f12,$f12
sub.s   $f14,$f16,$f18
add.s   $f8,$f14,$f12
sll     $6,$9,5
sub.s   $f8,$f14,$f8
```

cycle	Int unit 1	Int unit 2	Int unit 3	Flt unit 1	Flt unit 2	Flt unit 3
1	add $11,$2,$3	sll $6,$9,5	nop	mtc1 $4,$f12	sub.s $f14,$f16,$f18	nop
2	add $4,$5,$11	nop	nop	cvt.s.w $f12,$f12	nop	nop
3	nop	nop	nop	add.s $f8,$f14,$f12	nop	nop
4	nop	nop	nop	sub.s $f8,$f14,$f8	nop	nop

9. (5) What is the main advantage of using separate reservation stations versus using a centralized instruction window to supply instructions to the execution units within a superscalar system?

Using separate reservation stations versus using a centralized instruction window prevents one common type of instruction from monopolizing the instruction window and ensures there is space for all the relevant instructions. Further, reservation stations can have their sizes optimized based on the expected frequency of the various types of instructions.

10. (5) Which type of data hazard (RAW, WAR or WAW) is best handled by using a reorder buffer (ROB)? Explain why.

The job of the reorder buffer is to ensure the registers are retired in the proper order. Therefore, using a ROB would best handle WAW dependencies. WAW dependencies can be introduced by instructions later in the program completing earlier than instructions earlier in the program. Ensuring the registers are retired in the proper order would mitigate an issue such as this.

11. (5) For each of the 3 types of data hazard: RAW, WAR and WAW, indicate whether register renaming can or can not be used to avoid stalling the pipeline and explain why.

RAW – Renaming the registers cannot be used to avoid stalling the pipeline. RAW dependencies are typically between related registers, and cannot be renamed.

WAR – Renaming registers can be used to avoid stalling the pipeline by eliminating false dependencies in separate uses of the same register.

WAW – Renaming registers can be used to avoid stalling the pipeline by eliminating false dependencies in separate uses of the same register.

12. a) (5) What is the minimum width (in bits) required for the CPU-to-instruction memory bus in a MIPS degree 4 superscalar system?

4 * 32 = 128 bits

b) (5) Does hardware or does software select the instructions that execute together in the same clock cycle on a:

i. VLIW system
ii. superscalar system.

VLIW – software

superscalar system - hardware