NOTE: this is the SOLUTION to Quiz 5.

The correct answers are indicated for each question, with explanations as needed.

Dr. Manikas

Your Answers:

1 4 / 4 points

We are given the following MIPS code with the following latencies. What is the **baseline performance** (number of cycles) for this code?

Label	Opcode	Operands	Comment	Latency
Loop:	L.D	F2,0(Rx)	# F2 <= Mem[Rx+0]	3
I0:	MULT.D	F2,F0,F2	# F2 <= F0 * F2	4
I1:	L.D	F4,0(Ry)	# F4 <= Mem[Ry+0]	3
I2:	ADD.D	F4,F0,F4	# F4 <= F0 + F4	2
I3:	S.D	F4,0(Ry)	# Mem[Ry+0] <= F4	1
I4:	ADDI	Rx,Ry,#8	# Rx <= Ry + 8	0
I5:	SUB	R20,R4,Rx	# R20 <= R4 – Rx	0
I6:	BNZ	R20, Loop	# If R20 ≠ 0, goto Loop	1

✓ 22

Feedback

General Feedback

Examine the code and add latencies. Recall that latencies are **extra clock cycles needed**, so add 1 to each latency. Sum all values to get the baseline performance.

Label	Opcode	Operands	Comment	Latency	Cycles
Loop:	L.D	F2,0(Rx)	# F2 <= Mem[Rx+0]	3	4
I0:	MULT.D	F2,F0,F2	# F2 <= F0 * F2	4	5

I1:	L.D	F4,0(Ry)	# F4 <= Mem[Ry+0]	3	4
I2:	ADD.D	F4,F0,F4	# F4 <= F0 + F4	2	3
I3:	S.D	F4,0(Ry)	# Mem[Ry+0] <= F4	1	2
I4:	ADDI	Rx,Ry,#8	# Rx <= Ry + 8	0	1
I5:	SUB	R20,R4,Rx	# R20 <= R4 – Rx	0	1
I6:	BNZ	R20, Loop	# If R20 ≠ 0, goto Loop	1	2
					22

Thus, the total baseline performance of this code is 22 cycles.

2 4 / 4 points

We are given the following MIPS code. Registers that have **data dependencies** are (check all that apply):

Label	Opcode	Operands	Comment	Latency
Loop:	L.D	F8,0(Rx)	# F8 <= Mem[Rx+0]	2
I0:	MULT.D	F10,F0,F8	# F10 <= F0 * F8	4
I1:	L.D	F6,0(Ry)	# F6 <= Mem[Ry+0]	2
I2:	SUB.D	F4,F0,F8	# F4 <= F0 - F8	1
I3:	S.D	F4,0(Ry)	# Mem[Ry+0] <= F4	2
I4:	ADDI	Rx,Ry,#8	# Rx <= Ry + 8	0

- [ ] Ry
- [x] F4
- [ ] Rx
- [ ] F0
- [x] F8
- [ ] F10

Feedback

General Feedback

F8 value updated in Loop: L.D, used as input in I0: MULT.D

F4 value updated in I2: SUB.D, used as input in I3: S.D

3 4 / 4 points

We are given the following partial MIPS code with the following latencies. How many **stalls** should be added after the first line of code?

Opcode	Operands	Latency
L.D	F8,0(Rx)	2
MULT.D	F10,F0,F8	4

- 3
- ✓ 2
- 1
- 4
- 0

Feedback

General Feedback

MULT.D has a data dependency in register F8 with L.D.

L.D has a latency of 2, so we need to add 2 stalls after instruction L.D.

4 4 / 4 points

We are given the following MIPS code with the following latencies:

		Latency
L.D	F0, 10(R2)	2
ADD.D	F10, F8, F0	1
DIV.D	F2, F10, F6	5
L.D	F4, 0(R3)	2
ADD.D	F12, F4, F2	1

The code has started executing using the following **Tomasulo's** architecture, up through clock cycle **2** as shown in the table below. Items updated during cycle 2 are shown in ***red*** font.

Instruction status table:

Instruction		j	k	Issue	Exec comp	Write result
L.D	F0	10	R2	1		
ADD.D	F10	F8	F0	2		
DIV.D	F2	F10	F6			
L.D	F4	0	R3			
ADD.D	F12	F4	F2			

Load buffers:

	Busy	Addres
Load1	Yes	10+R2
Load2	No	
Load3	No	

Reservation Stations:

				S1	S2	RS	RS
Time	Name	Busy	Op	Vj	Vk	Qj	Qk
	Add1	Yes	ADD.D	F8			Load1
	Add2	No					
	Add3	No					
	Mult1	No					
	Mult2	No					

Register result status:

Clock		F0	F2	F4	F6	F8	F10	F12	F14
2	FU	Load1					Add1		

During clock cycle **3**, which **Load buffers** get updated (select all that are updated during this cycle)?

- [ ] Load2
- [ ] Load1
- [ ] Load3
- [x] No load buffers get updated

Feedback

General Feedback

DIV.D instruction issued with latency 5. Assign to reservation station Mult1. Value of F6 known, but value of F10 waiting on result of Add1. Load1 completing.

Load1 is completing, but will not write result until next cycle. Therefore, no load buffers get updated.

Instruction status table:

Instruction		j	k	Issue	Exec comp	Write result
L.D	F0	10	R2	1	3	
ADD.D	F10	F8	F0	2		
DIV.D	F2	F10	F6	3		
L.D	F4	0	R3			
ADD.D	F12	F4	F2			

Load buffers:

	Busy	Address
Load1	Yes	10+R2
Load2	No	
Load3	No	

Reservation Stations:

Time	Name	Busy	Op	S1 Vj	S2 Vk	RS Qj	RS Qk
	Add1	Yes	ADD.D	F8			Load1
	Add2	No					
	Add3	No					
	Mult1	Yes	DIV.D		F6	Add1	
	Mult2	No					

Register result status:

Clock		F0	F2	F4	F6	F8	F10	F12	F14
3	FU	Load1	Mult1				Add1		

5 4 / 4 points

We are given the following MIPS code with the following latencies:

		Latency
L.D	F0, 10(R2)	2
ADD.D	F10, F8, F0	1
DIV.D	F2, F10, F6	5
L.D	F4, 0(R3)	2
ADD.D	F12, F4, F2	1

The code has started executing using the following **Tomasulo's** architecture, up through clock cycle **2** as shown in the table below. Items updated during cycle 2 are shown in **red** font.

Instruction status table:

Instruction		j	k	Issue	Exec comp	Write result
L.D	F0	10	R2	1		
ADD.D	F10	F8	F0	2		
DIV.D	F2	F10	F6			
L.D	F4	0	R3			
ADD.D	F12	F4	F2			

Load buffers:

	Busy	Addres
Load1	Yes	10+R2
Load2	No	
Load3	No	

Reservation Stations:

Time	Name	Busy	Op	S1 Vj	S2 Vk	RS Qj	RS Qk
	Add1	Yes	ADD.D	F8			Load1
	Add2	No					
	Add3	No					
	Mult1	No					
	Mult2	No					

Register result status:

Clock		F0	F2	F4	F6	F8	F10	F12	F14
2	FU	Load1					Add1		

During clock cycle **3**, which **Reservation Stations** get updated (select all that are updated during this cycle)?

- [x] Mult1
- [ ] No reservation stations get updated
- [ ] Add1
- [ ] Mult2
- [ ] Add3
- [ ] Add2

Feedback

General Feedback

DIV.D instruction issued with latency 5. Assign to reservation station Mult1. Value of F6 known, but value of F10 waiting on result of Add1. Load1 completing.

DIV.D is assigned to Mult1, so this reservation station is updated.

Instruction status table:

Instruction		j	k	Issue	Exec comp	Write result
L.D	F0	10	R2	1	3	
ADD.D	F10	F8	F0	2		
DIV.D	F2	F10	F6	3		
L.D	F4	0	R3			
ADD.D	F12	F4	F2			

Load buffers:

	Busy	Address
Load1	Yes	10+R2
Load2	No	
Load3	No	

Reservation Stations:

Time	Name	Busy	Op	S1 Vj	S2 Vk	RS Qj	RS Qk
	Add1	Yes	ADD.D	F8			Load1
	Add2	No					
	Add3	No					
	Mult1	Yes	DIV.D		F6	Add1	
	Mult2	No					

Register result status:

Clock		F0	F2	F4	F6	F8	F10	F12	F14
3	FU	Load1	Mult1				Add1		

6 4 / 4 points

We are given the following MIPS code with the following latencies:

		Latency
L.D	F2,0(R2)	2
MULT.D	F8,F6,F2	3
L.D	F4,4(R4)	2
ADD.D	F12,F0,F4	1
ADD.D	F10,F8,F2	1

The code has started executing using the following **Tomasulo's** architecture, up through clock cycle **3** as shown in the table below. Items updated during cycle 3 are shown in ***red*** font.

Instruction status table:

Instruction		j	k	Issue	Exec comp	Write result
L.D	F2	0	R2	1	3	
MULT.D	F8	F6	F2	2		
L.D	F4	4	R4	3		
ADD.D	F12	F0	F4			
ADD.D	F10	F8	F2			

Load buffers:

	Busy	Address
Load1	Yes	0+R2
Load2	Yes	4+R4
Load3	No	

Reservation Stations:

Time	Name	Busy	Op	S1 Vj	S2 Vk	RS Qj	RS Qk
	Add1	No					
	Add2	No					
	Add3	No					
	Mult1	Yes	MULT.D	F6			Load1
	Mult2	No					

Register result status:

Clock		F0	F2	F4	F6	F8	F10	F12	F14
3	FU		Load1	Load2		Mult1			

During clock cycle 4, in the **Register result status** section, which registers get updated (select all that are updated during this cycle)?

- [ ] F0
- [ ] F6
- [ ] F4
- [ ] F8
- [x] F2
- [ ] No registers get updated
- [x] F12
- [ ] F14
- [ ] F10

Feedback

General Feedback

- First L.D writes result to FU F2 and frees up Load Buffer Load1.
- MULT.D has both input register values (F6, F2) and now starts executing with latency 3.
- First ADD.D instruction is issued and occupies Reservation Station Add1.
- Register F0 is available now, but Add1 must wait for Load2 to get value for F4.

Both registers F2 and F12 are updated.

Instruction status table:

Instruction		j	k	Issue	Exec comp	Write result
L.D	F2	0	R2	1	3	4
MULT.D	F8	F6	F2	2		
L.D	F4	4	R4	3		
ADD.D	F12	F0	F4	4		
ADD.D	F10	F8	F2			

Load buffers:

	Busy	Address
Load1	No	
Load2	Yes	4+R4
Load3	No	

Reservation Stations:

Time	Name	Busy	Op	S1 Vj	S2 Vk	RS Qj	RS Qk
	Add1	Yes	ADD.D	F0			Load2
	Add2	No					
	Add3	No					
3	Mult1	Yes	MULT.D	F6	F2		
	Mult2	No					

Register result status:

Clock		F0	F2	F4	F6	F8	F10	F12	F14
4	FU		M[0+R2]	Load2		Mult1		Add1	