

---

# CS152 – Exam 2 Review

2004-5-2

Jack and Kurt and John

[www-inst.eecs.berkeley.edu/~cs152/](http://www-inst.eecs.berkeley.edu/~cs152/)



## Question 3A: 1c

---

- What is the bottleneck to move from 1 to 2-way issue ... not counting the issue unit?



## Question 3a: 1c Answer

---

- What is the bottleneck to move from 1 to 2-way issue ... not counting the issue unit?
- Well, just about anything in ALL the 2-way commit paths ... cdb, rob, etc.
  - A worse answer is any one functional unit, as the code can be compiled to issue packets of insts.



## Question 3a: 1d

---

- RAW?
- RAR?
- WAW?
- WAR?



## Question 3a: 1d

---

- RAW – Register Status + in-order issue  
→ reads will get correct value because they always wait until the providing FU (gotten from reg status) issues the value



## Question 3a: 1d

---

- RAR – NOT A HAZARD
  - Ha haha hahahahahahahah



## Question 3a: 1d

---

- WAW – Reg status + in-order issue →
  - All subsequent reads will get the last write (by looking in reg status) if the result is still in flight for both writes. Otherwise, it's just like a RAW.



## Question 3a: 1d

---

- WAR – Reg status + in-order issue →
  - The read will get its pointer from the reg status fields, the write will change the value in the status fields, but not the pointer that the read will use.



## Question 3a: 1e

---

- How many stations?
- Did you say 2a/4a/10a/16a?



## Question 3a: 1e

---

- How many stations?
- Did you say  $2a/4a/10a/16a$ ?
  - You recognized that dual-issue will double the number of entries we need for the fus.
  - Logic: Issue a stream of 2 adds per cycle first two will finish after a cycles, therefore the queue will be  $2 * a$  cycles.



## Question 3a: 1e

---

- How many stations?
- Did you say inf/inf/inf/inf?
  - You recognized that for dual-issue, we need to double the number of functional units in order to empty the q as fast as we (may) fill it.
  - Logic: issue a continuous stream of 2 adds per cycle. If we have only 1 add FU, we can broadcast at most 1 add per cycle.
    - Issue rate > commit rate! BAD BAD BAD



## Question 3a: 1e

---

- How many stations?
- Did you say inf/inf/inf/inf?



## Question 3a: 1f

---

- How many physical regs if ...
  - 8 programmer-visible
  - up to 3 reads per inst
  - Up to 32 in flight at a time



## Question 3a: 1f

---

- How many physical regs if ...
  - 8 programmer-visible
  - up to 3 reads per inst
  - Up to 32 in flight at a time
- Did you say 32?
  - Logic: stream of adds like this:
    - Add \$t0 \$t0 \$t0 \$t1
    - Add \$t0 \$t0 \$t0 \$t2
    - Add \$t0 \$t0 \$t0 \$t2
  - If 32 of these are in flight, we need 32 dest regs



## Question 3a: 1f

---

- BUT ...



## Question 3a: 1f

---

- BUT ...
  - These 32 insts are all writing reg \$t0 ...
  - if we need 32 regs to hold the intermediates ...
  - then we will need 8 more to hold the values for the other regs (including the original \$t0)!
  - Total: 40 (Or 39 if you are \*really\* sneaky!)



## Question 3a: 1f

---

- Interesting notes:
  - 1: It's not a function of the number of reg reads in an inst – it is a function of the number of insts in flight at a time!



## Question 3:

---

### Extra Credit (Problem 3X):

Assume that you have a Tomasulo architecture with functional units of the same execution latency (number of cycles) as our deeply pipelined processor (*be careful to adjust use latencies to get number of execution cycles!*). Assume that it issues one instruction per cycle and has an unpipelined divider with a small number of reservation stations. Suppose the other functional units are duplicated with many reservation stations and that there are many CDBs. . What is the minimum number of divide reservation stations to achieve one instruction per cycle with the optimized code of (3b)? Show your work. [*hint: assume that the maximum issue rate is sustained and look at the scheduling of a single iteration*]

*Load: 3 cycles, Add: 2 cycles, Multiply: 4 cycles, Divide: 9 cycles (careful here!)*

```
loop: ldf    $F20, 0($r10)
      ldf    $F10, 8($r10) ←
      multf  $F6, $F20, $F1 ←
      addf   $F12, $F6, $F2 ←
      addi   $r10, $r10, #16
      divf   $F13, $F12, $F10
      addi   $r20, $r20, #8
      subi   $r1, $r1, #1
      bne    $r1, $zero, loop ←
      stf    -8($r20), $F13
```



## Question 3:

---

*Load: 3 cycles, Add: 2 cycles, Multiply: 4 cycles, Divide: 9 cycles (careful here!)*

```
loop: ldf    $F20, 0($r10)
      ldf    $F10, 8($r10) ←
      multf  $F6, $F20, $F1 ←
      addf   $F12, $F6, $F2 ←
      addi   $r10, $r10, #16
      divf   $F13, $F12, $F10
      addi   $r20, $r20, #8
      subi   $r1, $r1, #1
      bne    $r1, $zero, loop ←
      stf    -8($r20), $F13
```

### Keys to Problem:

- 1) # of station entries needed = # of div instructions in flight at same time
- 2)



## Question 3:

---

*Load: 3 cycles, Add: 2 cycles, Multiply: 4 cycles, Divide: 9 cycles (careful here!)*

```
loop: ldf    $F20, 0($r10)
      ldf    $F10, 8($r10) ←
      multf  $F6, $F20, $F1 ←
      addf   $F12, $F6, $F2 ←
      addi   $r10, $r10, #16
      divf   $F13, $F12, $F10
      addi   $r20, $r20, #8
      subi   $r1, $r1, #1
      bne    $r1, $zero, loop ←
      stf    -8($r20), $F13
```

### Keys to Problem:

- 1) # of station entries needed = # of div instructions in flight at same time
- 2) We can trace through a few iterations of the loop to see how many divs are active at any given time



## Question 3:

---

*Load: 3 cycles, Add: 2 cycles, Multiply: 4 cycles, Divide: 9 cycles (careful here!)*

```
loop: ldf    $F20, 0($r10)
      ldf    $F10, 8($r10) ←
      multf  $F6, $F20, $F1 ←
      addf   $F12, $F6, $F2 ←
      addi   $r10, $r10, #16
      divf   $F13, $F12, $F10
      addi   $r20, $r20, #8
      subi   $r1, $r1, #1
      bne    $r1, $zero, loop ←
      stf    -8($r20), $F13
```

### Keys to Problem:

- 1) # of station entries needed = # of div instructions in flight at same time
- 2) We can trace through a few iterations of the loop to see how many divs are active at any given time
- 3) We need a table to trace the tomasulo!













# Question 3:

Tomasulo Trace:

**CC 6:** First Few instructions

N	rd	rs	rt	I	E1	EF	WB	N	rd	rs	rt	I	E1	EF	WB
ldf	F20	R10		1	2	4	5								
Ldf	<b>F10</b>	R10		2	3	5	<b>6</b>								
multf	<b>F6</b>	F20	F1	3	<b>6</b>	<b>9*</b>									
addf	<b>F12</b>	<b>F6</b>	F2	4											
addi	R10	R10		5	<b>6</b>	<b>6</b>									
divf	<b>F13</b>	<b>F12</b>	<b>F10</b>	<b>6</b>											

# Question 3:

Tomasulo Trace:

CC 7: First->Second Divf

N	rd	rs	rt	I	E1	EF	WB
ldf	F20	R10		1	2	4	5
Ldf	F10	R10		2	3	5	6
multf	F6	F20	F1	3	6	9*	
addf	F12	F6	F2	4			
addi	R10	R10		5	6	6	7
divf	F13	F12	F10	6			
<b>addi</b>	<b>R20</b>	<b>R20</b>		<b>7</b>			

N	rd	rs	rt	I	E1	EF	WB

# Question 3:

Tomasulo Trace:

**CC 8: First->Second Divf**

N	rd	rs	rt	I	E1	EF	WB	N	rd	rs	rt	I	E1	EF	WB
ldf	F20	R10		1	2	4	5								
Ldf	F10	R10		2	3	5	6								
multf	F6	F20	F1	3	6	9*									
addf	F12	F6	F2	4											
addi	R10	R10		5	6	6	7								
divf	F13	F12	F10	6											
addi	R20	R20		7	8	8									
<b>Subi</b>	<b>R1</b>	<b>R1</b>		<b>8</b>											

# Question 3:

Tomasulo Trace:

**CC 9: First->Second Divf**

N	rd	rs	rt	I	E1	EF	WB
ldf	F20	R10		1	2	4	5
Ldf	F10	R10		2	3	5	6
multf	F6	F20	F1	3	6	9	
addf	F12	F6	F2	4			
addi	R10	R10		5	6	6	7
divf	F13	F12	F10	6			
addi	R20	R20		7	8	8	9
Subi	R1	R1		8	9	9	
bne	--	R1		9			

N	rd	rs	rt	I	E1	EF	WB



# Question 3:

Tomasulo Trace:

CC 10: First->Second Divf

N	rd	rs	rt	I	E1	EF	WB	N	rd	rs	rt	I	E1	EF	WB
ldf	f20	r10		1	2	4	5								
ldf	f10	r10		2	3	5	6								
multf	f6	f20	f1	3	6	9	10								
addf	f12	f6	f2	4											
addi	r10	r10		5	6	6	7								
divf	f13	f12	f10	6											
addi	r20	r20		7	8	8	9								
subi	r1	r1		8	9	9	10								
bne	--	r1		9	--	--	--								
stf	--	f13	r20	10											



# Question 3:

Tomasulo Trace:

**CC 11: First->Second Divf**

N	rd	rs	rt	I	E1	EF	WB	N	rd	rs	rt	I	E1	EF	WB
ldf	f20	r10		1	2	4	5								
ldf	f10	r10		2	3	5	6								
multf	f6	f20	f1	3	6	9	10								
addf	f12	f6	f2	4	11	12*									
addi	r10	r10		5	6	6	7								
divf	f13	f12	f10	6											
addi	r20	r20		7	8	8	9								
subi	r1	r1		8	9	9	10								
bne	--	r1		9	--	--	--								
stf	--	f13	r20	10											
<b>ldf</b>	<b>f20</b>	<b>r10</b>		<b>11</b>											

# Question 3:

Tomasulo Trace:

CC 12: First->Second Divf

N	rd	rs	rt	I	E1	EF	WB	N	rd	rs	rt	I	E1	EF	WB
ldf	f20	r10		1	2	4	5								
ldf	f10	r10		2	3	5	6								
multf	f6	f20	f1	3	6	9	10								
addf	f12	f6	f2	4	11	12									
addi	r10	r10		5	6	6	7								
divf	f13	f12	f10	6											
addi	r20	r20		7	8	8	9								
subi	r1	r1		8	9	9	10								
bne	--	r1		9	--	--	--								
stf	--	f13	r20	10											
ldf	f20	r10		11	12	14*									
<b>ldf</b>	<b>f10</b>	<b>r10</b>		<b>12</b>											



# Question 3:

Tomasulo Trace:

**CC 13: First->Second Divf**

N	rd	rs	rt	I	E1	EF	WB	N	rd	rs	rt	I	E1	EF	WB
ldf	f20	r10		1	2	4	5								
ldf	f10	r10		2	3	5	6								
multf	f6	f20	f1	3	6	9	10								
addf	<b>f12</b>	f6	f2	4	11	12	<b>13</b>								
addi	r10	r10		5	6	6	7								
divf	<b>f13</b>	<b>f12</b>	f10	6											
addi	r20	r20		7	8	8	9								
subi	r1	r1		8	9	9	10								
bne	--	r1		9	--	--	--								
stf	--	<b>f13</b>	r20	10											
ldf	<b>f20</b>	r10		11	12	14*									
ldf	f10	r10		12	<b>13</b>	<b>15*</b>									
multf	<b>f6</b>	<b>f20</b>	<b>f1</b>	<b>13</b>											













# Question 3:

Tomasulo Trace:

CC 19: First->Second- >Third Divf

N	rd	rs	rt	I	E1	EF	WB
ldf	f20	r10		1	2	4	5
ldf	f10	r10		2	3	5	6
multf	f6	f20	f1	3	6	9	10
addf	f12	f6	f2	4	11	12	13
addi	r10	r10		5	6	6	7
divf	f13	f12	f10	6	14	22	
addi	r20	r20		7	8	8	9
subi	r1	r1		8	9	9	10
bne	--	r1		9	--	--	--
stf	--	f13	r20	10			
ldf	f20	r10		11	12	14	15
ldf	f10	r10		12	13	15	16
multf	f6	f20	f1	13	16	19	

N	rd	rs	rt	I	E1	EF	WB
addf	f12	f6	f2	14			
addi	r10	r10		15	16	16	17
divf	f13	f12	f10	16			
addi	r20	r20		17	18	18	19
subi	r1	r1		18	19	19	
bne	--	r1		19	--	--	--



# Question 3:

Tomasulo Trace:

CC 20: First->Second- >Third Divf

N	rd	rs	rt	I	E1	EF	WB
ldf	f20	r10		1	2	4	5
ldf	f10	r10		2	3	5	6
multf	f6	f20	f1	3	6	9	10
addf	f12	f6	f2	4	11	12	13
addi	r10	r10		5	6	6	7
divf	f13	f12	f10	6	14	22	
addi	r20	r20		7	8	8	9
subi	r1	r1		8	9	9	10
bne	--	r1		9	--	--	--
stf	--	f13	r20	10			
ldf	f20	r10		11	12	14	15
ldf	f10	r10		12	13	15	16
multf	f6	f20	f1	13	16	19	20

N	rd	rs	rt	I	E1	EF	WB
addf	f12	f6	f2	14			
addi	r10	r10		15	16	16	17
divf	f13	f12	f10	16			
addi	r20	r20		17	18	18	19
subi	r1	r1		18	19	19	20
bne	--	r1		19	--	--	--
stf	--	f13	r20	20			



# Question 3:

Tomasulo Trace:

**CC 21: First->Second- >Third Divf**

N	rd	rs	rt	I	E1	EF	WB	N	rd	rs	rt	I	E1	EF	WB
ldf	f20	r10		1	2	4	5	addf	f12	f6	f2	14	21	22	
ldf	f10	r10		2	3	5	6	addi	r10	r10		15	16	16	17
multf	f6	f20	f1	3	6	9	10	divf	f13	f12	f10	16			
addf	f12	f6	f2	4	11	12	13	addi	r20	r20		17	18	18	19
addi	r10	r10		5	6	6	7	subi	r1	r1		18	19	19	20
divf	f13	f12	f10	6	14	22		bne	--	r1		19	--	--	--
addi	r20	r20		7	8	8	9	stf	--	f13	r20	20			
subi	r1	r1		8	9	9	10	<b>ldf</b>	<b>f20</b>	<b>r10</b>		<b>21</b>			
bne	--	r1		9	--	--	--								
stf	--	f13	r20	10											
ldf	f20	r10		11	12	14	15								
ldf	f10	r10		12	13	15	16								
multf	f6	f20	f1	13	16	19	20								



# Question 3:

Tomasulo Trace:

CC 22: First->Second- >Third Divf

N	rd	rs	rt	I	E1	EF	WB	N	rd	rs	rt	I	E1	EF	WB
ldf	f20	r10		1	2	4	5	addf	f12	f6	f2	14	21	22	
ldf	f10	r10		2	3	5	6	addi	r10	r10		15	16	16	17
multf	f6	f20	f1	3	6	9	10	divf	f13	f12	f10	16			
addf	f12	f6	f2	4	11	12	13	addi	r20	r20		17	18	18	19
addi	r10	r10		5	6	6	7	subi	r1	r1		18	19	19	20
divf	f13	f12	f10	6	14	22		bne	--	r1		19	--	--	--
addi	r20	r20		7	8	8	9	stf	--	f13	r20	20			
subi	r1	r1		8	9	9	10	ldf	f20	r10		21	22	24*	
bne	--	r1		9	--	--	--	<b>ldf</b>	<b>f10</b>	<b>r10</b>		<b>22</b>			
stf	--	f13	r20	10											
ldf	f20	r10		11	12	14	15								
ldf	f10	r10		12	13	15	16								
multf	f6	f20	f1	13	16	19	20								



# Question 3:

Tomasulo Trace:

CC 23: First->Second- >Third Divf

N	rd	rs	rt	I	E1	EF	WB
ldf	f20	r10		1	2	4	5
ldf	f10	r10		2	3	5	6
multf	f6	f20	f1	3	6	9	10
addf	f12	f6	f2	4	11	12	13
addi	r10	r10		5	6	6	7
divf	f13	f12	f10	6	14	22	23
addi	r20	r20		7	8	8	9
subi	r1	r1		8	9	9	10
bne	--	r1		9	--	--	--
stf	--	f13	r20	10			
ldf	f20	r10		11	12	14	15
ldf	f10	r10		12	13	15	16
multf	f6	f20	f1	13	16	19	20

N	rd	rs	rt	I	E1	EF	WB
addf	f12	f6	f2	14	21	22	23
addi	r10	r10		15	16	16	17
divf	f13	f12	f10	16			
addi	r20	r20		17	18	18	19
subi	r1	r1		18	19	19	20
bne	--	r1		19	--	--	--
stf	--	f13	r20	20			
ldf	f20	r10		21	22	24*	
ldf	f10	r10		22	23	25*	
multf	f6	f20	f1	23			



# Question 3:

Tomasulo Trace:

CC 24: First->Second- >Third Divf

N	rd	rs	rt	I	E1	EF	WB
ldf	f20	r10		1	2	4	5
ldf	f10	r10		2	3	5	6
multf	f6	f20	f1	3	6	9	10
addf	f12	f6	f2	4	11	12	13
addi	r10	r10		5	6	6	7
divf	f13	f12	f10	6	14	22	23
addi	r20	r20		7	8	8	9
subi	r1	r1		8	9	9	10
bne	--	r1		9	--	--	--
stf	--	f13	r20	10	<b>24</b>	<b>26*</b>	
ldf	f20	r10		11	12	14	15
ldf	f10	r10		12	13	15	16
multf	f6	f20	f1	13	16	19	20

N	rd	rs	rt	I	E1	EF	WB
addf	f12	f6	f2	14	21	22	23
addi	r10	r10		15	16	16	17
divf	f13	f12	f10	16	24	32*	
addi	r20	r20		17	18	18	19
subi	r1	r1		18	19	19	20
bne	--	r1		19	--	--	--
stf	--	f13	r20	20			
ldf	f20	r10		21	22	24	
ldf	f10	r10		22			
multf	f6	f20	f1	23			
...							



## Question 3:

---

Tomasulo Trace:

**CC 23:** First->Second- >Third Divf

<b>Divf1:</b>	<b>Issued 6</b>	<b>Finished 23</b>
<b>Divf2:</b>	<b>Issued 16</b>	<b>Finished 33</b>
<b>Divf3:</b>	<b>Issued ??</b>	<b>Finished ??</b>

**We're Done!**

## Question 3:

---

Tomasulo Trace:

**CC 23:** First->Second- >Third Divf

<b>Divf1:</b>	<b>Issued 6</b>	<b>Finished 23</b>
<b>Divf2:</b>	<b>Issued 16</b>	<b>Finished 33</b>
<b>Divf3:</b>	<b>Issued 26</b>	<b>Finished 43</b>

**We're Done!**

**The second divf issues before the first finished, so we will need at least 2 entries.**

**The first finishes before the third issues, so we will need at most 2 entries.**

**Therefore, we need 2 entries.**

