

Lab Report 7

Computer Architecture Lab

201651015 Dipansh Khandelwal

201651008 Aman Singh

25th March, 2019

Aim

To apply Tomasulo's algorithm for instruction level parallelism and compare the performance with Scoreboarding.

Questions and Answers

Problem - 1

```
LD    F6    34+    R2
LD    F2    45+    R3
MULTD F0     F2     F4
SUBD  F8     F6     F2
DIVD  F10    F0     F6
ADDD  F6     F8     F2
```

TOMASULO'S ALGORITHM FOR DYNAMIC SCHEDULING																																																				
DEMO HELP		Clock Cycle No. 56																																																		
Enter Instructions Below:		Instruction Status				Buffers																																														
LD ▾ F6 ▾ R2 ▾ 3 ▾		<table border="1"><thead><tr><th></th><th>Issue</th><th>Execute</th><th>Write Result</th></tr></thead><tbody><tr><td>LD F6 R2 3</td><td>1</td><td>2</td><td>3</td></tr><tr><td>LD F2 R3 4</td><td>2</td><td>3</td><td>4</td></tr><tr><td>MULTD F0 F2 F4</td><td>3</td><td>14</td><td>15</td></tr><tr><td>SUBD F8 F6 F2</td><td>4</td><td>6</td><td>7</td></tr><tr><td>DIVD F10 F0 F6</td><td>5</td><td>55</td><td>56</td></tr><tr><td>ADDD F6 F8 F2</td><td>6</td><td>9</td><td>10</td></tr></tbody></table>					Issue	Execute	Write Result	LD F6 R2 3	1	2	3	LD F2 R3 4	2	3	4	MULTD F0 F2 F4	3	14	15	SUBD F8 F6 F2	4	6	7	DIVD F10 F0 F6	5	55	56	ADDD F6 F8 F2	6	9	10	<table border="1"><thead><tr><th>Name</th><th>Busy</th><th>Address</th></tr></thead><tbody><tr><td>Load0</td><td>no</td><td></td></tr><tr><td>Load1</td><td>no</td><td></td></tr><tr><td>Store0</td><td>no</td><td></td></tr><tr><td>Store1</td><td>no</td><td></td></tr></tbody></table>				Name	Busy	Address	Load0	no		Load1	no		Store0	no		Store1	no	
	Issue	Execute	Write Result																																																	
LD F6 R2 3	1	2	3																																																	
LD F2 R3 4	2	3	4																																																	
MULTD F0 F2 F4	3	14	15																																																	
SUBD F8 F6 F2	4	6	7																																																	
DIVD F10 F0 F6	5	55	56																																																	
ADDD F6 F8 F2	6	9	10																																																	
Name	Busy	Address																																																		
Load0	no																																																			
Load1	no																																																			
Store0	no																																																			
Store1	no																																																			
LD ▾ F2 ▾ R3 ▾ 4 ▾		Reservation station status																																																		
MULTD ▾ F0 ▾ F2 ▾ F4 ▾		<table border="1"><thead><tr><th>Resv. Station</th><th>Busy</th><th>Op</th><th>Vj</th><th>Vk</th><th>Qj</th><th>Qk</th></tr></thead><tbody><tr><td>Add0</td><td>no</td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>Add1</td><td>no</td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>Add2</td><td>no</td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>Mult0</td><td>no</td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>Mult1</td><td>no</td><td></td><td></td><td></td><td></td><td></td></tr></tbody></table>								Resv. Station	Busy	Op	Vj	Vk	Qj	Qk	Add0	no						Add1	no						Add2	no						Mult0	no						Mult1	no						
Resv. Station	Busy	Op	Vj	Vk	Qj	Qk																																														
Add0	no																																																			
Add1	no																																																			
Add2	no																																																			
Mult0	no																																																			
Mult1	no																																																			
SUBD ▾ F8 ▾ F6 ▾ F2 ▾		Register result status																																																		
DIVD ▾ F10 ▾ F0 ▾ F6 ▾																																																				
ADDD ▾ F6 ▾ F8 ▾ F2 ▾																																																				
None ▾ ▾ ▾ ▾																																																				
None ▾ ▾ ▾ ▾																																																				
None ▾ ▾ ▾ ▾																																																				
None ▾ ▾ ▾ ▾																																																				
Enter the number of execution cycles taken by the functional units:																																																				
Integer: 1 ▾ FP Add: 2 ▾																																																				
FP Multiply: 10 ▾ FP Divide: 40 ▾																																																				
Enter the number of Reservation Stations:																																																				

Answer the following questions:-

1. After how many clock cycles can this program branch back to the beginning?

56 (62 in case of scoreboarding)

2. Does re-ordering influence the execution time of this program and how?

No.

3. Is there a Write-after-Read hazard present and how is it solved?

Yes, there is write-after-read hazard in last two instructions which are resolved by Tomasulo Algo.

Problem - 2

```
LD F0, 0(R1)
ADDD F4, F0, F2
SD F4, 0, R1
LD F0, -8(R1)
ADDD F4, F0, F2
SD F4, -8(R1)
```

TOMASULO'S ALGORITHM FOR DYNAMIC SCHEDULING

[DEMO](#) [HELP](#)

Enter Instructions Below:

LD	R0	R0	0
ADD	R4	R0	R2
SD	R4	R0	0
LD	R0	R0	4
ADD	R0	R0	R2
SD	R4	R0	4
None			
None			
None			
None			

Enter the number of execution cycles taken by the functional units:

Integer: FP Add:

FP Multiply: FP Divide:

Enter the number of Reservation Stations:

FP Add: FP Multiply:

FP Load: FP Store:

Select the type of output required:

☒ Run to completion

☐ Step by Step Output

Clock Cycle No: 9

Instruction Status

	Issue	Execute	Write Result
LD R0 R0 0	1	2	3
ADD R4 R0 R2	2	5	6
SD R4 R0 0	3	4	5
LD R0 R0 4	4	5	6
ADD R4 R0 R2	6	8	9
SD R4 R0 4	6	7	8

Buffers

Name	Busy	Address
Load0	no	
Load1	no	
Store0	no	
Store1	no	

Reservation station status

Res. Station	Busy	Op	Vj	Vk	Qj	Qk
Add0	no					
Add1	no					
Add2	no					
Mult0	no					
Mult1	no					

Register result status

Answer the following questions:-

1. After how many clock cycles can this program branch back to the beginning?

9 (24 in case of scoreboarding)

2. Does re-ordering influence the execution time of this program and how?

No

3. Is there a Write-after-Read hazard present and how is it solved?

No, there isn't any Write-after-Read Hazard.

Problem - 3

```
LD F0, 0(R1)
LD F4, 4(R1)
MULTD F8, F0, F2
LD F4, 6(R1)
LD F6, 5(R1)
MULTD F10, F4, F6
ADD F10, F6, F10
LD F8, 3(R1)
ADD F10, F10, F8
```

TOMAR'S ALGORITHM FOR DYNAMIC SCHEDULING

DEMO HELP

Enter Instruction Below:

LD F0, 0(R1)
LD F4, 4(R1)
MULTD F8, F0, F2
LD F4, 6(R1)
LD F6, 5(R1)
MULTD F10, F4, F6
ADD F10, F6, F10
LD F8, 3(R1)
ADD F10, F10, F8

Enter the number of reservation cycles taken by the functional unit:
Integer: 1 FP Add: 2
FP Multiply: 10 FP Divide: 40

Enter the number of Reservation Stations:
FP Add: 5 FP Multiply: 2
FP Load: 2 FP Store: 2

Click Cycle No. 32

Instruction Status

	Issue	Execute	Write Result
LD F0 R1 0	1	2	3
LD F4 R1 4	2	3	4
MULTD F8 F0 F2	3	34	35
LD F4 R1 4	4	5	6
LD F6 R1 5	5	6	7
MULTD F10 F4 F6	6	36	37
ADD F10 F6 F10	7	38	39
LD F8 R1 3	8	9	10
ADD F10 F10 F8	9	40	41

Buffer

Name	Busy	Address
Load0	no	
Load1	no	
Store0	no	
Store1	no	

Reservation station status

Res. Station	Busy	Op	Vj	Vk	Qj	Qk
RA00	no					
RA01	no					
RA02	no					
RA03	no					
RA04	no					

Register result status



Answer the following questions:-

1. After how many clock cycles can this program branch back to the beginning?

32 (65 in case of scoreboarding)

2. Does re-ordering influence the execution time of this program and how?

No

3. Is there a Write-after-Read hazard present and how is it solved?

Yes, in second last and third last instruction there is write-after-read hazard which is resolved by register renaming.
