Lab Report 8

Score Board And Tomasulo's Loop Algorithm

201651015 Dipansh Khandelwal

201651008 Aman Singh

1st April, 2019

Aim

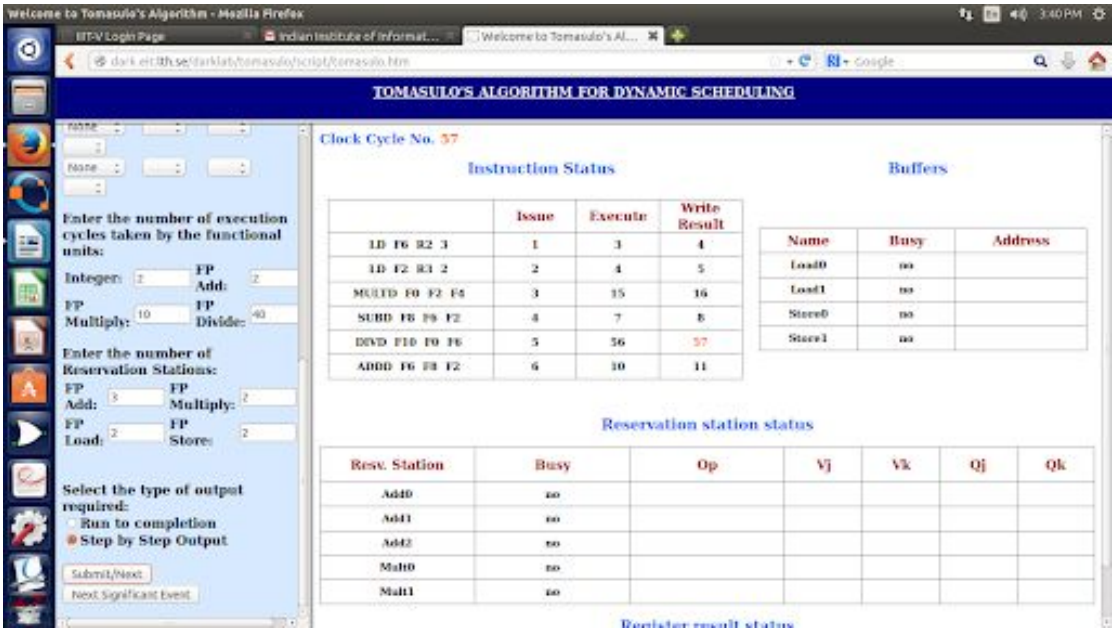
Scheduling the given codes using Tomasulo's algorithm and Score Board algorithms.

Questions and Answers

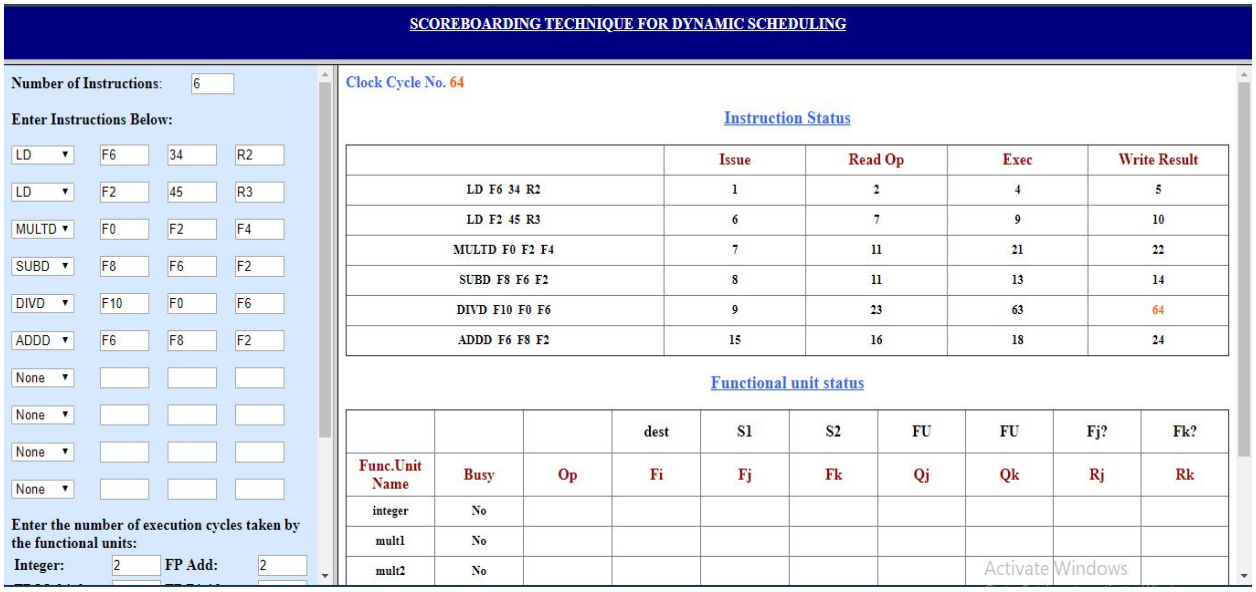
Problem - 1

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

Using Tomasulo's algorithm



Using Scoreboard's algorithm



SCOREBOARDING TECHNIQUE FOR DYNAMIC SCHEDULING

Number of Instructions: 6

Enter Instructions Below:

LD	F6	34	R2
LD	F2	45	R3
MULTD	F0	F2	F4
SUBD	F8	F6	F2
DIVD	F10	F0	F6
ADDD	F6	F8	F2
None			
None			
None			
None			

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

Integer: 2 FP Add: 2

Clock Cycle No. 64

Instruction Status

	Issue	Read Op	Exec	Write Result
LD F6 34 R2	1	2	4	5
LD F2 45 R3	6	7	9	10
MULTD F0 F2 F4	7	11	21	22
SUBD F8 F6 F2	8	11	13	14
DIVD F10 F0 F6	9	23	63	64
ADDD F6 F8 F2	15	16	18	24

Functional unit status

			dest	S1	S2	FU	FU	Fj?	Fk?
Func.Unit Name	Busy	Op	Fi	Fj	Fk	Qj	Qk	Rj	Rk
integer	No								
mult1	No								
mult2	No								

Answer the following questions:-

1. In which clock cycle (numbered 0,1,2,...) does the second LD instruction complete?

5

2. In which clock cycle does the MULTD instruction complete?

16

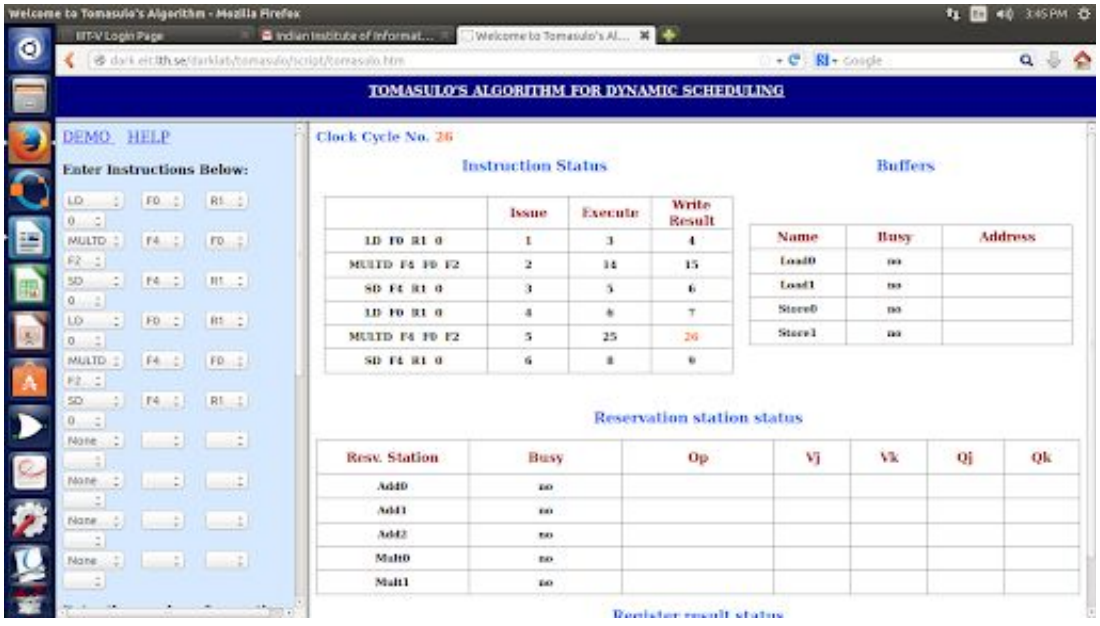
3. In which clock cycle does the ADDD instruction complete?

11

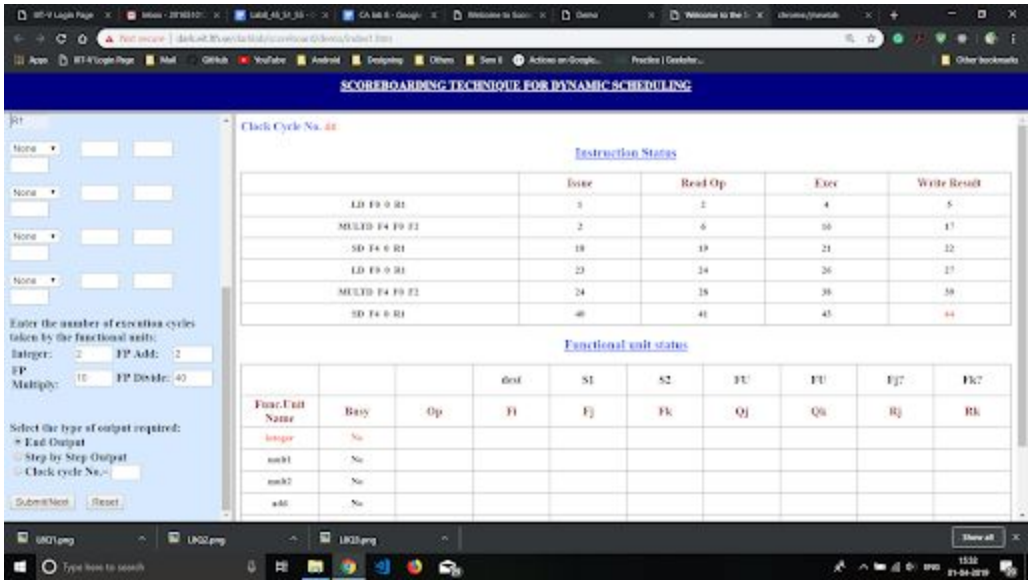
Problem - 2

```
LOOP: LD F0 0 R1
MULTD F4 F0 F2
SD F4 0 R1
LD F0 0 R1
MULTD F4 F0 F2
SD F4 0 R1 LOOP
```

Using Tomasulo's algorithm



Using Scoreboard's algorithm



Answer the following questions:-

1. In which clock cycle does the second SD instruction complete?

9

2. In which clock cycle does the first MULTD instruction complete?

15

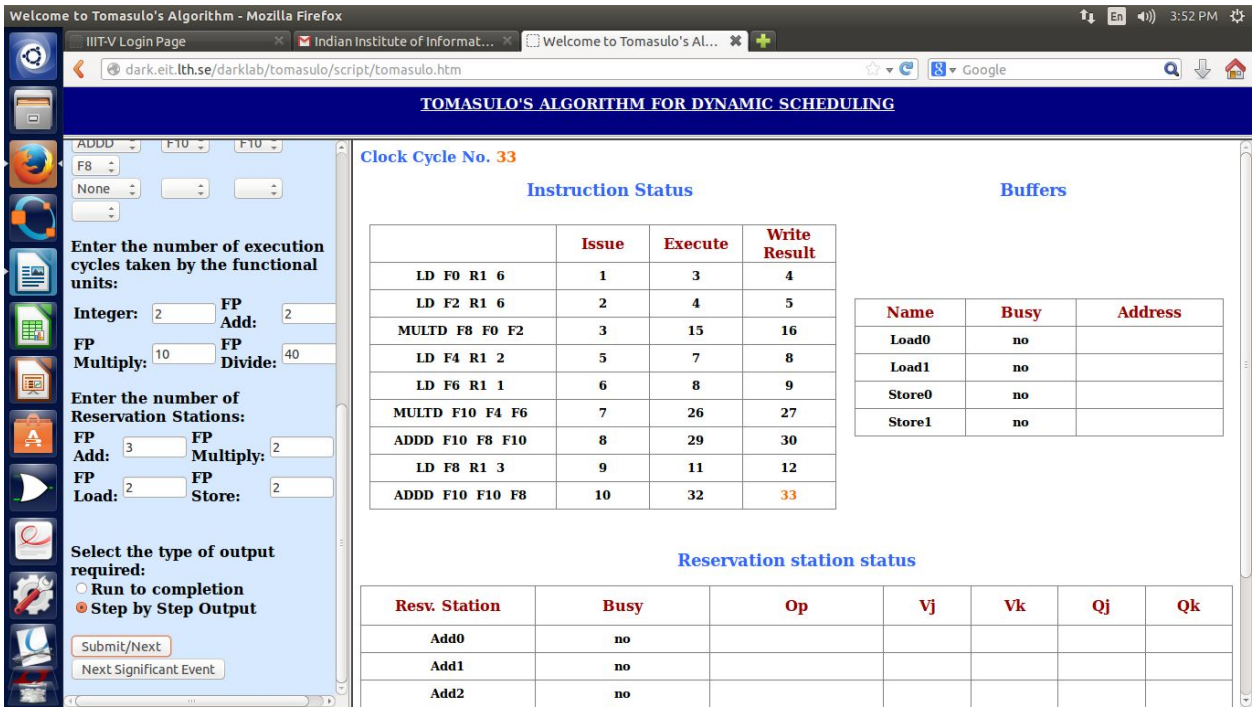
3. In which clock cycle does the second MULTD instruction complete?

20

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
ADDD F10, F6, F10
LD F8, 3(R1)
ADDD F10, F10, F8
```

Using Tomasulo's algorithm



	Issue	Execute	Write Result
LD F0 R1 6	1	3	4
LD F2 R1 6	2	4	5
MULTD F8 F0 F2	3	15	16
LD F4 R1 2	5	7	8
LD F6 R1 1	6	8	9
MULTD F10 F4 F6	7	26	27
ADDD F10 F8 F10	8	29	30
LD F8 R1 3	9	11	12
ADDD F10 F10 F8	10	32	33

Name	Busy	Address
Load0	no	
Load1	no	
Store0	no	
Store1	no	

Resv. Station	Busy	Op	Vj	Vk	Qj	Qk
Add0	no					
Add1	no					
Add2	no					

Using Scoreboard's algorithm

Answer the following questions:-

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

33

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

Yes

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

Yes

ADDD F10, F8, F10

LD F8, 10(R1)

QUESTIONS

1. How can the pipelined processor be faster than one without pipelining?

Ans: Pipelining is a process of arrangement of hardware elements of the CPU such that its overall performance is increased because simultaneous execution of more than one instruction takes place in a pipelined processor.

2. What are the special problems that appear in pipelining?

Ans: Limitations on the practical depth of a pipeline arise from:

Pipeline latency The fact that the execution time of each instruction does not decrease puts limitations on pipeline depth;

Imbalance among pipeline stages Imbalance among the pipe stages reduces performance since the clock can run no faster than the time needed for the slowest pipeline stage;

Pipeline overhead Pipeline overhead arises from the combination of pipeline register delay (setup time plus propagation delay) and clock skew.

3. Is there a difference in writing compilers for pipelined processors?

Ans: Compilers do most of the work directly relating to pipeline during low-level optimization and code generation. For example, a compiler can reorder instructions to reduce stalls as a result of data dependency between instructions so an instruction that needs the result from a previous one has it by the time the layer instruction is executed. The compiler could also reduce stalls from branch misprediction using conditional update rather than conditional execution and reduce memory stalls by prefetching. Modern processors can do some of those tasks, so whether the compiler needs to do pipeline optimizations depends on the hardware.

4. Which is faster, straight code or code with many branches?

Ans: Straight Code is faster since it lowers the complexity.

5. What does RISC mean? What does CISC mean?

Ans: The term "CISC" (complex instruction set computer or computing) refers to computers designed with a full set of computer instructions that were intended to provide needed capabilities in the most efficient way.

RISC (reduced instruction set computer) is a microprocessor that is designed to perform a smaller number of types of computer instructions so that it can operate at a higher speed (perform more millions of instructions per second, or MIPS).

6. Is the Pentium processor pipelined? AMD Phenom? Intel Core2? ARM Cortex-A?

Ans: Yes, Pentium processor, Intel Core 2, AMD Phenom, ARM Cortex-A are pipelined.