Lecture 33 [01.05.2020]

Design Concepts in Static and Dynamic Scheduling

John Jose

Assistant Professor

Department of Computer Science & Engineering

Indian Institute of Technology Guwahati, Assam.

True/False

- Which of the following statements is/are FALSE?

(I) In a dynamic scheduled processor every execution unit writes the result and the name of the reservation station waiting for the result on to the CDB.

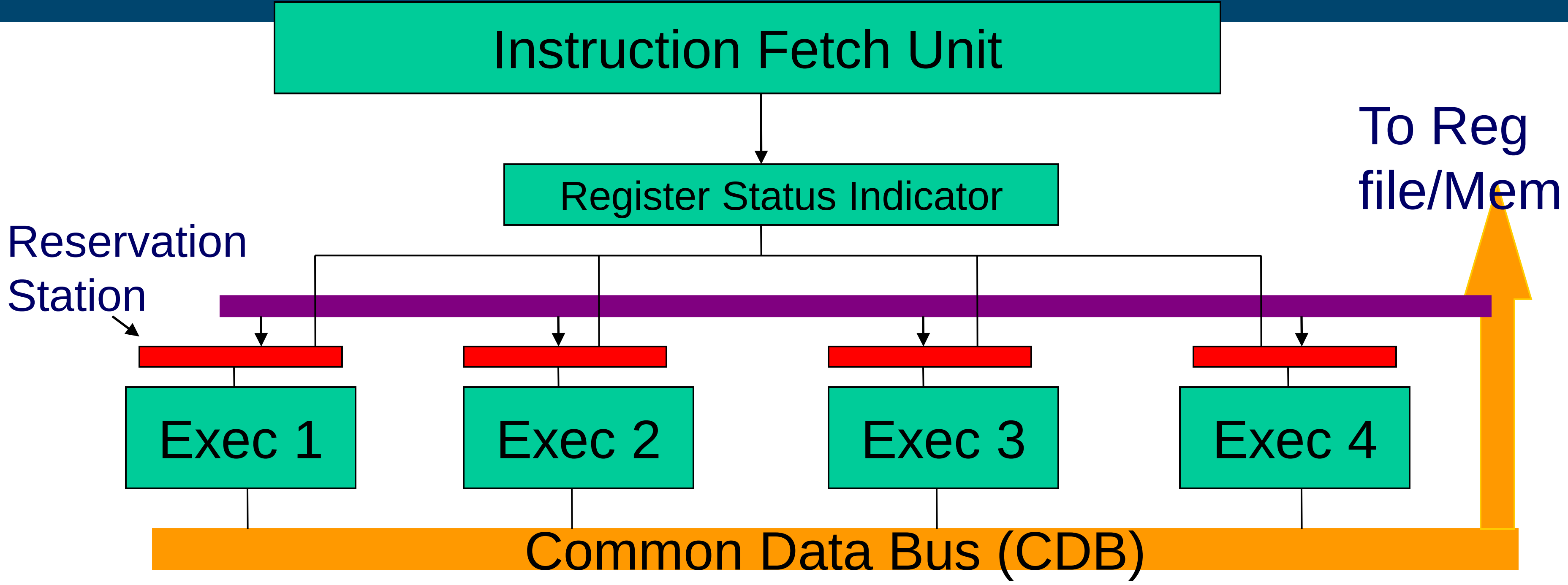
(II) In a speculative dynamic scheduled processor we can issue an instruction if there is an empty reservation station even though operands are nit available.

(III) If the register status indicator value of a the register is 0, then it means the operand is available in Register File

(IV) In a dynamic scheduled pipeline instructions are issued in order, executed out of order, completed out of order and committed in order.

(A) II & III only (B) I & II only (C) I & IV only (D) III & IV only

Dynamic Scheduling - Tomasulo



Every Execution unit writes the result along with the unit number on to the CDB which is forwarded to all reservation stations, Reg-file and Memory

True/False

- Which of the following statements is/are FALSE?

(I) In a dynamic scheduled processor every execution unit writes the result and the name of the reservation station waiting for the result on to the CDB. ✗

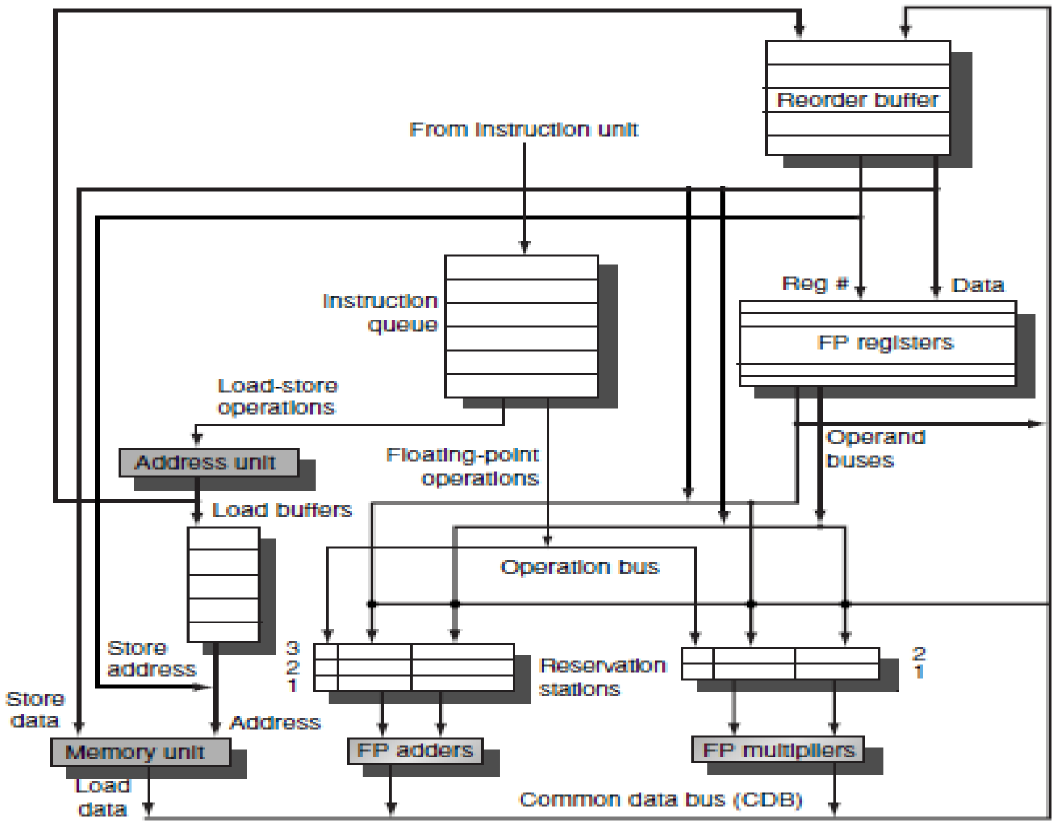
(II) In a speculative dynamic scheduled processor we can issue an instruction if there is an empty reservation station even though operands are not available.

(III) If the register status indicator value of a the register is 0, then it means the operand is available in Register File

(IV) In a dynamic scheduled pipeline instructions are issued in order, executed out of order, completed out of order and committed in order.

(A) II & III only (B) I & II only (C) I & IV only (D) III & IV only

Tomasulo's Algorithm with ROB



True/False

- Which of the following statements is/are FALSE?

(I) In a dynamic scheduled processor every execution unit writes the result and the name of the reservation station waiting for the result on to the CDB. ✖

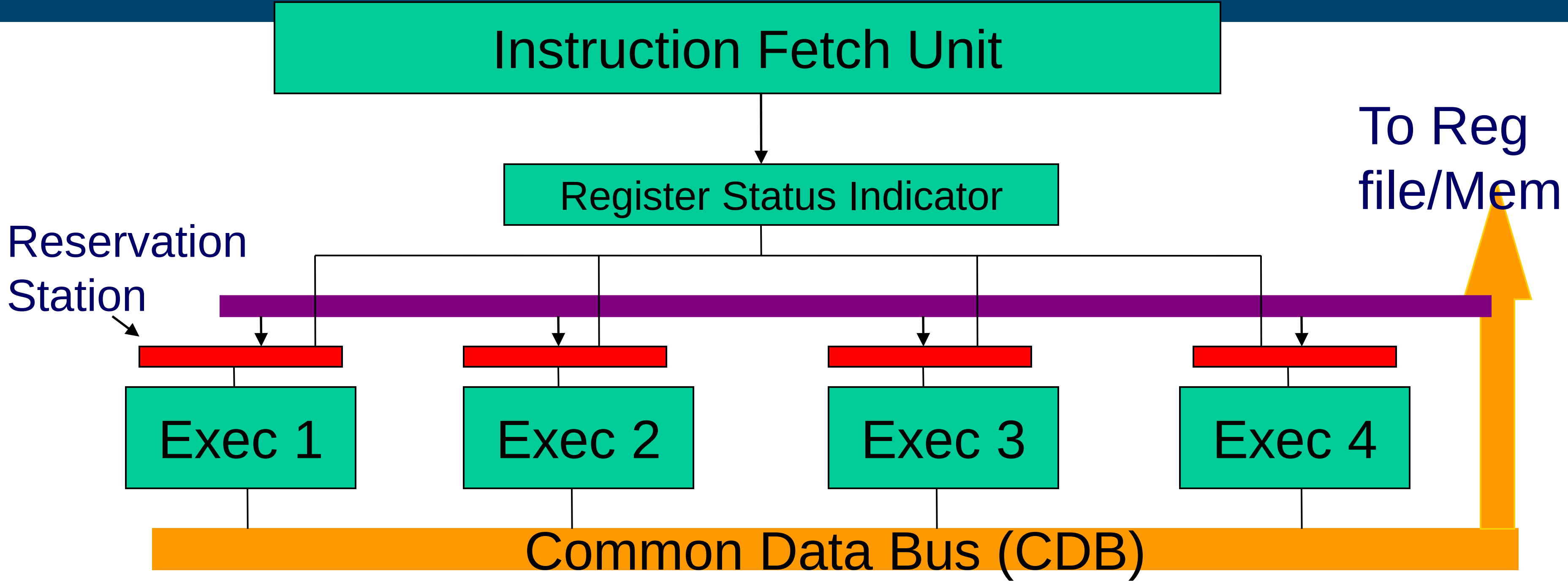
(II) In a speculative dynamic scheduled processor we can issue an instruction if there is an empty reservation station even though operands are not available. ✖

(III) If the register status indicator value of a the register is 0, then it means the operand is available in Register File

(IV) In a dynamic scheduled pipeline instructions are issued in order, executed out of order, completed out of order and committed in order.

(A) II & III only (B) I & II only (C) I & IV only (D) III & IV only

Dynamic Scheduling - Tomasulo



Register Status Indicator indicates whether the latest value of the register is in the reg file or currently being computed by some execution unit and if so, it states the execution unit number

True/False

❖ Which of the following statements is/are FALSE?

(I) In a dynamic scheduled processor every execution unit writes the result and the name of the reservation station waiting for the result on to the CDB. ✗

(II) In a speculative dynamic scheduled processor we can issue an instruction if there is an empty reservation station even though operands are not available. ✗

(III) If the register status indicator value of a the register is 0, then it means the operand is available in Register File ✓

(IV) In a dynamic scheduled pipeline instructions are issued in order, executed out of order, completed out of order and committed in order.

(A) II & III only (B) I & II only (C) I & IV only (D) III & IV only

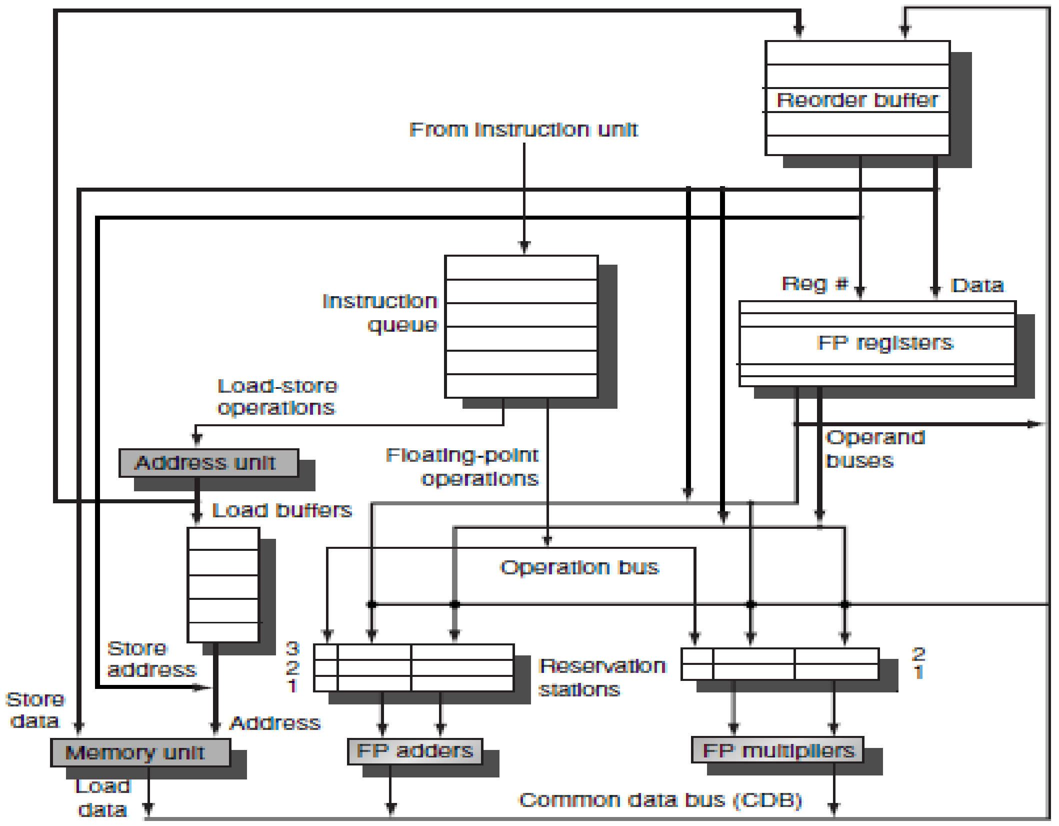
How dynamic scheduling works ?

- To allow out-of-order execution, **split the ID stage into two**
  - **Issue**—Decode instructions, check for structural hazards.
  - **Read operands**—Wait until no data hazards, then read operands.
- In a dynamically scheduled pipeline, all instructions pass through the **issue stage in order** (in-order issue); however, they can be stalled or bypass each other in the second stage (read operands) and thus enter execution out of order.

Hardware-Based Speculation

- Speculate – **Fetch, Issue and Execute** as if branch predictions are always correct; **commit** the results only if prediction was correct
- **Instruction commit:** allowing an instruction to update the register file/memory ONLY when instruction is no longer speculative
- Reorder buffer (ROB) – holds the result of instruction between completion and commit

Tomasulo’s Algorithm with ROB



True/False

❖ Which of the following statements is/are FALSE?

(I) In a dynamic scheduled processor every execution unit writes the result and the name of the reservation station waiting for the result on to the CDB. ✖

(II) In a speculative dynamic scheduled processor we can issue an instruction if there is an empty reservation station even though operands are not available. ✖

(III) If the register status indicator value of a the register is 0, then it means the operand is available in Register File ✓

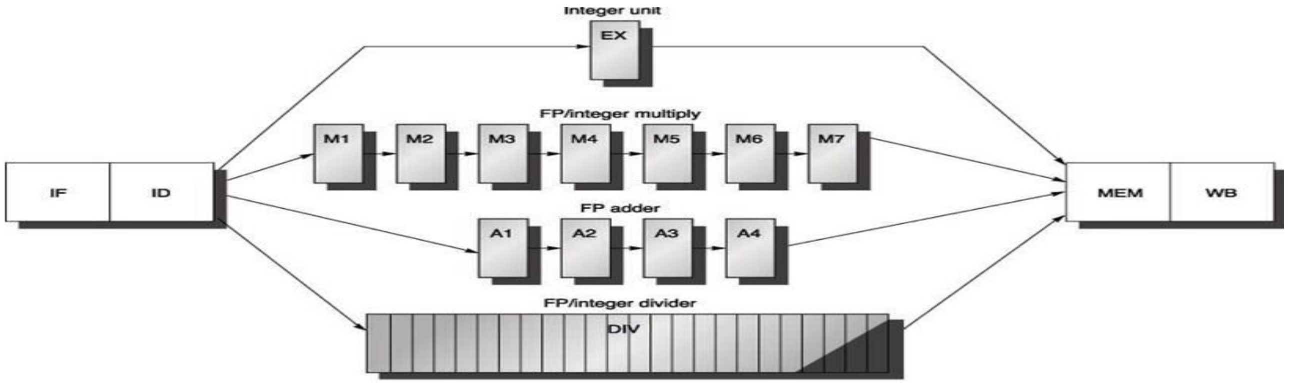
(IV) In a dynamic scheduled pipeline instructions are issued in order, executed out of order, completed out of order and committed in order. ✓

(A) II & III only **(B) I & II only** (C) I & IV only (D) III & IV only

Multicycle Pipeline

Consider the following instruction sequence executed on a MIPS floating point pipeline. Operand forwarding is implemented. [R indicates integer registers and F indicates floating point registers]. Find the clock cycle in which STOR instruction reaches MEM stage. If 8(R2) contains value 'X' and F2 contains value 'A', then what is stored in 16(R3)

LOAD F4, 8(R2);
FMUL F0, F4, F2;
FADD F3, F0, F2;
STOR F3, 16(R3);



Multicycle Pipeline

```
LOAD F4, 8(R2);
FMUL F0, F4, F2;
FADD F3, F0, F2;
STOR F3, 16(R3);
```

```
LOAD F4, 8(R2);      - -- F4=X
FMUL F0, F4, F2;     ---- F0=AX
FADD F3, F0, F2;     ---- F3=AX+A
STOR F3, 16(R3);
```

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
F	D	X	M	W													
	F	D	*	M1	M2	M3	M4	M5	M6	M7	M	W					
		F	*	D	*	*	*	*	*	*	A1	A2	A3	A4	M	W	
				F	*	*	*	*	*	*	D	*	*	*	X	M	W

Tomasulo’s Algorithm

Consider an instruction pipeline of an issue width of 1 that uses Tomasulo's algorithm with two reservation station per functional unit. There is one Integer Mul Unit, one Integer Div unit and one Integer Add unit all connected to a single CDB. There is an arbitration mechanism for resolving CDB entry collision. Preference is given in round robin fashion. The functional units are not pipelined. An instruction waiting for data on CDB can move to its EX stage in the cycle after the CDB broadcast.

Assume the following information about functional units.

Functional unit type	Cycles in Execution stage
Integer Mul	4
Integer Div	8
Integer Add	1

Complete the following table using Tomasulo's algorithm with the above specifications. Fill in the cycle numbers in each pipeline stage for each instruction, and indicate where its source operand's are read from (use RF for register file, ROB for reorder buffer and CDB for common data bus).

Tomasulo’s Algorithm

#	Instruction	Issue	Src_Op1	Src_ Op2	EX	WB	Commit
1	DIVI R4, R4, #12	1	RF	Imm	2	10	11
2	MUL R2, R6, R12	2					
3	DIV R1, R1, R2	3					
4	ADD R5, R1, R3						
5	ADDI R7, R2, #4						
6	ADD R5, R6, R7						
7	ADDI R8,R8, #24						
8	ADD R9, R6, R8						
9	MUL R5, R5, R10						
10	ADD R6, R8, R5						

Tomasulo's Algorithm

#	Instruction	Issue	Src_Op1	Src_ Op2	EX	WB	Commit
1	DIVI R4, R4, #12	1	RF	Imm	2	10	11
2	MUL R2, R6, R12	2	RF	RF	3	7	12
3	DIV R1, R1, R2	3					
4	ADD R5, R1, R3						
5	ADDI R7, R2, #4						
6	ADD R5, R6, R7						
7	ADDI R8,R8, #24						
8	ADD R9, R6, R8						
9	MUL R5, R5, R10						
10	ADD R6, R8, R5						

Tomasulo’s Algorithm

#	Instruction	Issue	Src_Op1	Src_ Op2	EX	WB	Commit
1	**DIVI** R4, R4, #12	1	RF	Imm	2	10	11
2	MUL R2, R6, R12	2	RF	RF	3	7	12
3	**DIV** R1, R1, R2	3	RF	CDB	10	18	19
4	ADD R5, R1, R3						
5	ADDI R7, R2, #4						
6	ADD R5, R6, R7						
7	ADDI R8,R8, #24						
8	ADD R9, R6, R8						
9	MUL R5, R5, R10						
10	ADD R6, R8, R5						

Tomasulo’s Algorithm

#	Instruction	Issue	Src_Op1	Src_ Op2	EX	WB	Commit
1	DIVI R4, R4, #12	1	RF	Imm	2	10	11
2	MUL R2, R6, R12	2	RF	RF	3	7	12
3	DIV R1, R1, R2	3	RF	CDB	10	18	19
4	ADD R5, R1, R3	4	CDB	RF	19	20	21
5	ADDI R7, R2, #4						
6	ADD R5, R6, R7						
7	ADDI R8,R8, #24						
8	ADD R9, R6, R8						
9	MUL R5, R5, R10						
10	ADD R6, R8, R5						

Tomasulo’s Algorithm

#	Instruction	Issue	Src_Op1	Src_ Op2	EX	WB	Commit
1	DIVI R4, R4, #12	1	RF	Imm	2	10	11
2	MUL R2, R6, R12	2	RF	RF	3	7	12
3	DIV R1, R1, R2	3	RF	CDB	10	18	19
4	ADD R5, R1, R3	4	CDB	RF	19	20	21
5	ADDI R7, R2, #4	5	CDB	Imm	8	9	22
6	ADD R5, R6, R7						
7	ADDI R8,R8, #24						
8	ADD R9, R6, R8						
9	MUL R5, R5, R10						
10	ADD R6, R8, R5						

Tomasulo's Algorithm

#	Instruction	Issue	Src_Op1	Src_ Op2	EX	WB	Commit
1	DIVI R4, R4, #12	1	RF	Imm	2	10	11
2	MUL R2, R6, R12	2	RF	RF	3	7	12
3	DIV R1, R1, R2	3	RF	CDB	10	18	19
4	**ADD** R5, R1, R3	4	CDB	RF	19	20	21
5	**ADDI** R7, R2, #4	5	CDB	Imm	8	9	22
6	ADD R5, R6, R7	10	RF	ROB	11	12	23
7	ADDI R8,R8, #24						
8	ADD R9, R6, R8						
9	MUL R5, R5, R10						
10	ADD R6, R8, R5						

Tomasulo’s Algorithm

#	Instruction	Issue	Src_Op1	Src_ Op2	EX	WB	Commit
1	DIVI R4, R4, #12	1	RF	Imm	2	10	11
2	MUL R2, R6, R12	2	RF	RF	3	7	12
3	DIV R1, R1, R2	3	RF	CDB	10	18	19
4	**ADD** R5, R1, R3	4	CDB	RF	19	20	21
5	ADDI R7, R2, #4	5	CDB	Imm	8	9	22
6	**ADD** R5, R6, R7	10	RF	ROB	11	12	23
7	ADDI R8,R8, #24	13	RF	Imm	14	15	24
8	ADD R9, R6, R8						
9	MUL R5, R5, R10						
10	ADD R6, R8, R5						

Tomasulo's Algorithm

#	Instruction	Issue	Src_Op1	Src_ Op2	EX	WB	Commit
1	DIVI R4, R4, #12	1	RF	Imm	2	10	11
2	MUL R2, R6, R12	2	RF	RF	3	7	12
3	DIV R1, R1, R2	3	RF	CDB	10	18	19
4	**ADD** R5, R1, R3	4	CDB	RF	19	20	21
5	ADDI R7, R2, #4	5	CDB	Imm	8	9	22
6	ADD R5, R6, R7	10	RF	ROB	11	12	23
7	**ADDI** R8,R8, #24	13	RF	Imm	14	15	24
8	ADD R9, R6, R8	16	RF	ROB	17	19*	25
9	MUL R5, R5, R10						
10	ADD R6, R8, R5						

Tomasulo’s Algorithm

#	Instruction	Issue	Src_Op1	Src_ Op2	EX	WB	Commit
1	DIVI R4, R4, #12	1	RF	Imm	2	10	11
2	MUL R2, R6, R12	2	RF	RF	3	7	12
3	DIV R1, R1, R2	3	RF	CDB	10	18	19
4	**ADD** R5, R1, R3	4	CDB	RF	19	20	21
5	ADDI R7, R2, #4	5	CDB	Imm	8	9	22
6	ADD R5, R6, R7	10	RF	ROB	11	12	23
7	**ADDI** R8,R8, #24	13	RF	Imm	14	15	24
8	ADD R9, R6, R8	16	RF	ROB	17	19*	25
9	MUL R5, R5, R10	17	ROB	RF	18	22	26
10	ADD R6, R8, R5						

Tomasulo’s Algorithm

#	Instruction	Issue	Src_Op1	Src_ Op2	EX	WB	Commit
1	DIVI R4, R4, #12	1	RF	Imm	2	10	11
2	MUL R2, R6, R12	2	RF	RF	3	7	12
3	DIV R1, R1, R2	3	RF	CDB	10	18	19
4	**ADD** R5, R1, R3	4	CDB	RF	19	20	21
5	ADDI R7, R2, #4	5	CDB	Imm	8	9	22
6	ADD R5, R6, R7	10	RF	ROB	11	12	23
7	ADDI R8,R8, #24	13	RF	Imm	14	15	24
8	**ADD** R9, R6, R8	16	RF	ROB	17	19*	25
9	MUL R5, R5, R10	17	ROB	RF	18	22	26
10	ADD R6, R8, R5	20	ROB	CDB	23	24	27

johnjose@iitg.ac.in

http://www.iitg.ac.in/johnjose/