Tomasulo Implementation

Problem :- Calculating Determinant of a Square matrix.

Example :-

A is a 3x3 matrix.

$$A = \begin{matrix} 1 & 1 & 7 \\ 4 & 1 & 1 \\ 3 & 1 & 2 \end{matrix}$$

Det(A) = 1(1x2 - 1x1) - 1(4x2 - 1x3) + 7(4x1 - 1x3)
Det(A) = 1 - 5 + 7
Det(A) = 3

Three Stages of Tomasulo Algorithm

1. **Issue**—get instruction from FP Op Queue
–If reservation station free (no structural hazard),
– control issues instr & sends operands (renames registers).
2. **Execution**—operate on operands (EX)
– When both operands ready then execute;
– if not ready, watch Common Data Bus for result
3. **Write result**—finish execution (WB)
– Write on Common Data Bus to all awaiting units;
– reservation station available.

•Normal data bus: data + destination (“go to” bus)

•Common data bus: data + source (“come from” bus)

– Write if matches expected Functional Unit (produces result)

– Does the broadcast.

•16 fp registers used.

S. No.	Instruction	Denotation	No. of clock cycles
1	Load	LDR r_name addr	2
2	Store	STR addr r_name	2
3	Add	ADD r_result r_val1 r_val2	4
4	Subtract	SUB r_result r_val1 r_val2	4
5	Multiply	MUL r_result r_val1 r_val2	8
6	Divide	DIV r_result r_num r_deno	12

Instructions for computing determinant using Tomasulo Algorithm:-

```
LDR F0 100
LDR F1 104
MUL F2 F0 F1
LDR F3 108
LDR F4 112
MUL F5 F3 F4
SUB F6 F2 F5
```

```
LDR F7 116
MUL F0 F6 F7
LDR F8 120
LDR F9 124
MUL F10 F8 F9
LDR F11 128
LDR F12 132
MUL F13 F11 F12
SUB F14 F10 F13
LDR F15 136
MUL F1 F14 F15
LDR F2 140
LDR F3 144
MUL F4 F2 F3
LDR F5 148
LDR F6 152
MUL F7 F5 F6
SUB F8 F4 F7
LDR F9
MUL F2 F8 F9
SUB F3 F0 F1
ADD F0 F2 F3
STR 200 F0
```

No	Instruction	Issue	Start	End	Write	Explanation
1	LDR F0 100	0	0	2		F0 => free
2	LDR F1 104	1	1	3		F1 => free
3	MUL F2 F0 F1	2	3	11		F2 => free but waits for the output in

						F1
4	LDR F3 108	3	3	5		F3 => free, out of order execution
5	LDR F4 112	4	4	6		F4 => free
6	MUL F5 F3 F4	5	6	14		F5 => free but waits for the output in F1
7	SUB F6 F2 F5	6	14	18		F6 => free but waits for the output in F1
8	LDR F7 116	7	7	9		F7 => free, out of order execution
9	MUL F0 F6 F7	8	18	26		F0 => free but waits for the output in F1
10	LDR F8 120	9	9	11		F8 => free, out of order execution
11	LDR F9 124	10	10	12		F9 => free
12	MUL F10 F8 F9	11	12	20		F10 => free but waits for the output in

						F1
13	LDR F11 128	12	12	14		F11 => free, out of order execution
14	LDR F12 132	13	13	15		F12 => free
15	MUL F13 F11 F12	14	15	23		F13 => free but waits for the output in F1
16	SUB F14 F10 F13	15	23	27		F14 => free but waits for the output in F1
17	LDR F15 136	16	16	18		F15 => free, out of order execution
18	MUL F1 F14 F15	17	27	35		F1 => free but waits for the output in F1
19	LDR F2 140	18	18	20		F2 => free, out of order execution
20	LDR F3 144	19	19	21		F3 => free
21	MUL F4 F2 F3	20	21	29		F4 => free but waits for

						the output in F1
22	LDR F5 148	21	21	23		F5 => free, out of order execution
23	LDR F6 152	22	22	24		F6 => free
24	MUL F7 F5 F6	23	24	32		F7 => free but waits for the output in F1
25	SUB F8 F4 F7	24	32	36		F8 => free but waits for the output in F1
26	LDR F9	25	25	27		F9 => free, out of order execution
27	MUL F2 F8 F9	26	36	44		F2 => free but waits for the output in F1
28	SUB F3 F0 F1	27	44	48		F3 => free but waits for the output in F1
29	ADD F0 F2 F3	28	48	52		F0 => not free but waits for the

						output in F1
30	STR 200 F0	29	52	54	54	Final answer (Determinant of a square matrix) is written to memory

Number of clock cycles using Tomasulo Algorithm for calculating determinant	No of clock cycles using normal execution for calculating Determinant
54	116

- Decentralized control
- Use of reservation stations to buffer and/or rename registers (hence gets rid of WAW and WAR hazards)
- Results -and their names-are broadcast to reservations stations and register file •Instructions are issued in order but can be dispatched, executed and completed out-of-order
- No issue on structural hazards
- WAR :- renaming avoids
- WAW :- renaming avoids
- Control :- reservation stations

•THE END•

THANK YOU