ECE552 – Lab 3 Prelab
Bing Li – 1004191910

1. Tomasulo's algorithm extracts more ILP for a given program compared to an in-order processor because it allows for out of order execution – whereas an in-order processor will stall all the later instructions due to a RAW hazard, Tomasulo's algorithm will only force the instruction that has the RAW hazard to wait, while the subsequent independent instructions can still issue, providing more opportunities for parallel execution.

2. Reservation stations are used to allocate hardware resources to instructions. The fields of a reservation station entry are:

RS entry	Functionality
FU (Functional Unit)	Specifies which hardware functional unit is being reserved/used
Busy	Specifies whether the FU is reserved
Op	Specifies which operation the FU is performing
Qi, Qk	Specifies source tags (RS#/FU of the RS entry that will produce an input value)
Vj, Vk	Specifies source values (the actual values being used in the operation)

3. Tomasulo's algorithm deals with false dependencies (WAR and WAW hazards) by using register renaming. There are 2 storage locations for registers, the RegFile and the reservation stations; the RegFile holds the most recent values of the register, while the RS's hold the working values. In this way, a register's value is decoupled from its name in the RS.
For example, given the instructions:

```
1: DIV r2, r3 -> r1
2: ADD r1, r2 -> r3 # RAW hazard for r1
3: SUB r4, r5 -> r2 # WAR hazard for r2
4: ADD r4, r5 -> r1 # WAW hazard for r1
```

For simplicity we will only show the RS and Map tables,
Inst 1: R1 is mapped to output from MULT1, and the values of r2 and r3 are copied into Vj & Vk

FU	Busy	Op	Qj	Qk	Vj	Vk		Map Table	
MULT1	Y	DIV			[r2]	[r3]		R1	MULT1
ADD1	N							R2	
ADD2	N							R3	
ADD3	N							…	

Inst 2: R3 is mapped to output from ADD1, and the value of r2 is copied into Vk, and MULT1 is copied into Qj (since R1 maps to a RS entry)

FU	Busy	Op	Qj	Qk	Vj	Vk		Map Table	
MULT1	Y	DIV			[r2]	[r3]		R1	MULT1
ADD1	Y	ADD	MULT1			[r2]		R2	
ADD2	N							R3	ADD1
ADD3	N							…	

Inst 1: R2 is mapped to output from ADD2, and the values of r4 and r5 are copied into Vj & Vk. Since the value of r2 was copied into the RS entries of the instructions before it, the WAR hazard is resolved.

FU	Busy	Op	Qj	Qk	Vj	Vk		Map Table	
MULT1	Y	DIV			[r2]	[r3]		R1	MULT1
ADD1	Y	ADD	MULT1			[r2]		R2	ADD2
ADD2	Y	SUB			[r4]	[r5]		R3	ADD1
ADD3	N							…	

Inst 1: R1 is mapped to output from ADD1, and the values of r4 and r5 are copied into Vj & Vk. Since the previous value of R1 (output from MULT1) is sent directly to the FU using it (ADD1), R1 can be renamed to receive the latest value and therefore avoiding a WAW hazard.

FU	Busy	Op	Qj	Qk	Vj	Vk		Map Table	
MULT1	Y	DIV			[r2]	[r3]		R1	ADD3
ADD1	Y	ADD	MULT1			[r2]		R2	ADD2
ADD2	Y	SUB			[r4]	[r5]		R3	ADD1
ADD3	Y	ADD			[r4]	[r5]		…	