Aula 10: Algoritmo de Tomasulo

Revisão: Scoreboarding

- Limitações do scoreboarding
  - Sem forwarding
  - Limitado a instruções em um bloco básico
  - Paralelismo limitado pelo número de unidades funcionais (*structural hazards*)
  - Aguardar escrita em registradores para evitar hazards WAR
  - Prevenir execução até resolução de hazards WAW

Reduzindo Hazards de WAR e WAW:*Register Renaming* pelo Compilador

- Considere o código:

```
DIVD    F2,F4,F6        DIVD F2,F4,F6
SUBD    F4,F6,F8   →    SUBD 10,F12,F16
STD     F8,0(R1)        STD  F10,0(R1)
```

- Existe uma dependência do tipo WAR em F2 entre DIVD e SUBD
- Compilador consegue eliminar dependência a um custo adicional no número de registradores

Register Renaming em Hardware

- Toda vez que formos escrever em um registrador, aloque um novo registrador físico (número do registrador == apontador)
- Implicações para implementação
  - Precisa de mais registradores físicos do que endereços para registradores
  - Precisa alocação e *garbage collection* para mapearmos registradores físicos a endereços de registradores
  - Uso de registrador é remapeado para uso do apontador para registrador físico
- Principal vantagem sobre *scoreboarding*: não existem stalls para hazards WAR e WAW.

Register Renaming Dinâmico

- Código original

```
MULTD F0,F2,F4     MULTD F0a,F2a,F4a
SD    F0,0(R4)  ⇒  SD    F0a,0(R4a)
ADDU  R4,R2,#8     ADDU  R4b,R2a,#8
ADDD  F0,F6,F8     ADDD  F0b,F6a,F8a
```

- Dois hazards WAR (R4 e F0)

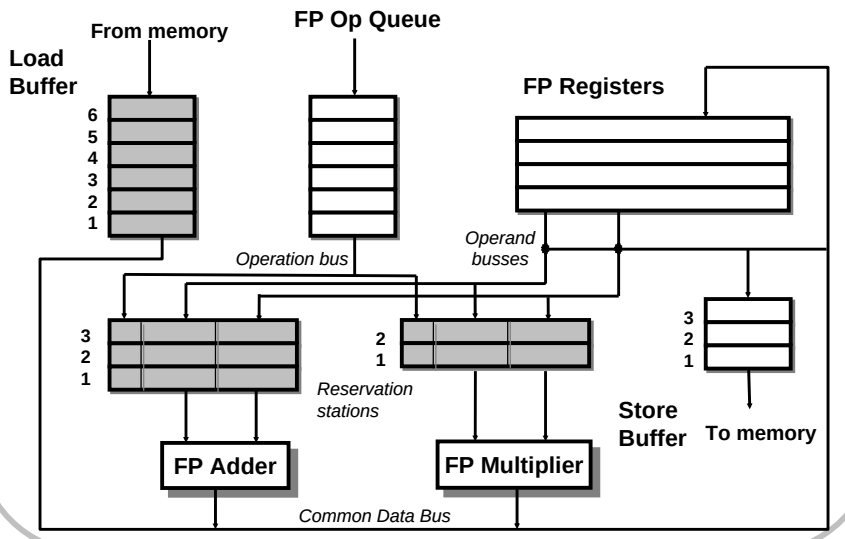
Algoritmo de Tomasulo

- Troca tabela de status de unidade funcional por *reservation station*
  - Cada FU possui conjunto de *reservation stations* para instruções aguardando execução
  - *Reservation station* armazena valores assim que eles ficarem disponíveis assim como apontadores para as *reservation stations* que proverão os operandos
  - Número original do registrador não é mais importante, uma vez que o número do registrador corresponderá a um registrador ou a uma *reservation station*
- Ao final da execução, *reservation station* faz um *broadcast* do resultado para todas as Fus que dependem dele
- Instruções podem ser *issued* na presença de hazards estruturais, do tipo WAW ou WAR

Algoritmo de Tomasulo

- Feito para IBM 360/91 3 anos depois do CDC 6600
- Objetivo: Alta performance sem compiladores especiais
- Diferenças entre ISA do IBM 360 e ISA do CDC 6600
  - IBM possui dois operandos/instr vs. 3 no CDC 6600
  - IBM possui 4 registradores de FP vs. 8 no CDC 6600
- Diferenças entre Algoritmo de Tomasulo & Scoreboard
  - Controle e buffers são distribuídos nas unidades funcionais vs. centralizados no scoreboard; chamados de *reservation stations*
  - Registradores nas instruções são trocados por apontadores nas *reservation stations* (pode conter dado ou apontador para unidade funcional gerando dado)
  - *Register renaming* para evitar hazards WAR e WAW
  - FU não precisa aguardar escrita em registrador para ler operando (*Common Data Bus)*
  - Loads e Stores são tratados como FU

Organização da Máquina



Três Estágios do Algoritmo de Tomasulo

1. Issue — busca instrução de *FP Op Queue*

   Dispara operação FP se *reservation table* está livre. Se operandos estão nos registradores, leia-os, caso contrário, renomeie os registradores. Se for um load, então aloque uma entrada no buffer de Loads, se não houverem entradas livres, houve um hazard estrutural.

2. Execução — execute instrução (EX)

   Quando operandos estiverem disponíveis, execute instrução; se não estiver pronto, observe *Common Data Bus* (CDB) até operandos estarem prontos. Nesse estágio, são checados hazards do tipo RAW.

3. Escrita de resultados — termine execução (WB)

   Escreva resultado no *Common Data Bus* para todas as unidades aguardando o resultado; marque a *reservation station* como disponível.

Componentes da *Reservation Station*

Tag

Value

Tag

Op—Operação executada pela FU (e.g., + or –)

Qj, Qk—*Reservation stations* produzindo operandos

Vj, Vk—Valores dos operandos

Busy— Indica que a *reservation station* e FU estão ocupadas

Qi — Para stores, indica o número da *reservation station* gerando resultado que será gravado na memória (o store também possui Vi contendo valor)

Status do registrador resultado — Indica qual FU escreverá cada registrador, se ele existir. Em branco quando nenhuma instrução pendente escreverá no registrador.

Regras para o Algoritmo de Tomasulo

- *Issue*:
  - SE: *Reservation station* está disponível
  - ENTÃO: Aloque *reservation station* para instrução
    - *Se registrador de operando não tiver resultado pendente, então seu valor está pronto; caso contrário, marque o campo Qj (Tag) com a reservation station que produzirá o resultado*
    - *Marque o status do registrador de resultado para a reservation station alocada (note que poderemos estar escrevendo sobre alguma outra reservation station)*

Regras para o Algoritmo de Tomasulo

- *Execução:*
  - SE: Valores dos operandos Vj e Vk estão presentes
  - ENTÃO: comece execução
- *Writeback:*
  - SE: slot no CDB está disponível
  - ENTÃO:
    - *Escreva todos os registradores para os quais somos o destino, e limpe o status do registrador*
    - *Para todas as reservation stations ou buffers de store que nos listam como pendências em Qj, Qk e Qi, escreva o valor e desmarque o tag*

Tomasulo Example Cycle 0

Instruction status

Instruction		j	k	Issue	Execution complete	Write Result
LD	F6	34+	R2			
LD	F2	45+	R3			
MULT	F0	F2	F4			
SUBD	F8	F6	F2			
DIVD	F10	F0	F6			
ADDD	F6	F8	F2			

	Busy	Address
Load1	No	
Load2	No	
Load3	No	

Reservation Stations

Time	Name	Busy	Op	S1 Vj	S2 Vk	RS for j Qj	RS for k Qk
0	Add1	No					
0	Add2	No					
	Add3	No					
0	Mult1	No					
0	Mult2	No					

Register result status

Clock	F0	F2	F4	F6	F8	F10	F12	...	F30
0	FU								

Tomasulo Example Cycle 1

Instruction status

Instruction		j	k	Issue	Execution complete	Write Result
LD	F6	34+	R2	1		
LD	F2	45+	R3			
MULT	F0	F2	F4			
SUBD	F8	F6	F2			
DIVD	F10	F0	F6			
ADDD	F6	F8	F2			

	Busy	Address
Load1	Yes	34+R2
Load2	No	
Load3	No	

Reservation Stations

Time	Name	Busy	Op	S1 Vj	S2 Vk	RS for j Qj	RS for k Qk
0	Add1	No					
0	Add2	No					
	Add3	No					
0	Mult1	No					
0	Mult2	No					

Register result status

Clock	F0	F2	F4	F6	F8	F10	F12	...	F30
1	FU			Load1					

Tomasulo Example Cycle 2

Instruction status

Instruction		j	k	Issue	Execution complete	Write Result
LD	F6	34+	R2	1		
LD	F2	45+	R3	2		
MULT	F0	F2	F4			
SUBD	F8	F6	F2			
DIVD	F10	F0	F6			
ADDD	F6	F8	F2			

	Busy	Address
Load1	Yes	34+R2
Load2	Yes	45+R3
Load3	No	

Reservation Stations

Time	Name	Busy	Op	S1 Vj	S2 Vk	RS for j Qj	RS for k Qk
0	Add1	No					
0	Add2	No					
	Add3	No					
0	Mult1	No					
0	Mult2	No					

Register result status

Clock	F0	F2	F4	F6	F8	F10	F12	...	F30
2	FU	Load2		Load1					

Tomasulo Example Cycle 3

Instruction status

Instruction		j	k	Issue	Execution complete	Write Result
LD	F6	34+	R2	1	3	
LD	F2	45+	R3	2		
MULT	F0	F2	F4	3		
SUBD	F8	F6	F2			
DIVD	F10	F0	F6			
ADDD	F6	F8	F2			

	Busy	Address
Load1	Yes	34+R2
Load2	Yes	45+R3
Load3	No	

Reservation Stations

Time	Name	Busy	Op	S1 Vj	S2 Vk	RS for j Qj	RS for k Qk
0	Add1	No					
0	Add2	No					
	Add3	No					
0	Mult1	Yes	MULTD		R(F4)	Load2	
0	Mult2	No					

Register result status

Clock	F0	F2	F4	F6	F8	F10	F12	...	F30
3	FU	Mult1	Load2		Load1				

Tomasulo Example Cycle 4

Instruction status

Instruction	j	k	Issue	Execution complete	Write Result
LD F6	34+	R2	1	3	4
LD F2	45+	R3	2	4	
MULTI F0	F2	F4	3		
SUBD F8	F6	F2	4		
DIVD F10	F0	F6			
ADDD F6	F8	F2			

	Busy	Address
Load1	No	
Load2	Yes	45+R3
Load3	No	

Reservation Stations

Time	Name	Busy	Op	S1 Vj	S2 Vk	RS for j Qj	RS for k Qk
0	Add1	Yes	SUBD	M(34+R2)			Load2
0	Add2	No					
	Add3	No					
0	Mult1	Yes	MULTD		R(F4)	Load2	
0	Mult2	No					

Register result status

Clock		F0	F2	F4	F6	F8	F10	F12	...	F30
4	FU	Mult1	Load2		M(34+R2)	Add1				

Tomasulo Example Cycle 5

Instruction status

Instruction	j	k	Issue	Execution complete	Write Result
LD F6	34+	R2	1	3	4
LD F2	45+	R3	2	5	
MULT F0	F2	F4	3		
SUBD F8	F6	F2	4		
DIVD F10	F0	F6	5		
ADDD F6	F8	F2			

	Busy	Address
Load1	No	
Load2	Yes	45+R3
Load3	No	

Reservation Stations

Time	Name	Busy	Op	S1 Vj	S2 Vk	RS for j Qj	RS for k Qk
0	Add1	Yes	SUBD	M(34+R2)			Load2
0	Add2	No					
	Add3	No					
0	Mult1	Yes	MULTD		R(F4)	Load2	
0	Mult2	Yes	DIVD		M(34+R2)	Mult1	

Register result status

Clock		F0	F2	F4	F6	F8	F10	F12	...	F30
5	FU	Mult1	Load2		M(34+R2)	Add1	Mult2			

Tomasulo Example Cycle 6

Instruction status

Instruction	j	k	Issue	Execution complete	Write Result
LD F6	34+	R2	1	3	4
LD F2	45+	R3	2	5	6
MULT F0	F2	F4	3		
SUBD F8	F6	F2	4		
DIVD F10	F0	F6	5		
ADDD F6	F8	F2	6		

	Busy	Address
Load1	No	
Load2	No	
Load3	No	

Reservation Stations

Time	Name	Busy	Op	S1 Vj	S2 Vk	RS for j Qj	RS for k Qk
2	Add1	Yes	SUBD	M(34+R2)	M(45+R3)		
0	Add2	Yes	ADDD		M(45+R3)	Add1	
	Add3	No					
10	Mult1	Yes	MULTD	M(45+R3)	R(F4)		
0	Mult2	Yes	DIVD		M(34+R2)	Mult1	

Register result status

Clock		F0	F2	F4	F6	F8	F10	F12	...	F30
6	FU	Mult1	M(45+R3)		Add2	Add1	Mult2			

Tomasulo Example Cycle 7

Instruction status

Instruction	j	k	Issue	Execution complete	Write Result
LD F6	34+	R2	1	3	4
LD F2	45+	R3	2	5	6
MULT F0	F2	F4	3		
SUBD F8	F6	F2	4		
DIVD F10	F0	F6	5		
ADDD F6	F8	F2	6		

	Busy	Address
Load1	No	
Load2	No	
Load3	No	

Reservation Stations

Time	Name	Busy	Op	S1 Vj	S2 Vk	RS for j Qj	RS for k Qk
1	Add1	Yes	SUBD	M(34+R2)	M(45+R3)		
0	Add2	Yes	ADDD		M(45+R3)	Add1	
	Add3	No					
9	Mult1	Yes	MULTD	M(45+R3)	R(F4)		
0	Mult2	Yes	DIVD		M(34+R2)	Mult1	

Register result status

Clock		F0	F2	F4	F6	F8	F10	F12	...	F30
7	FU	Mult1	M(45+R3)		Add2	Add1	Mult2			

Tomasulo Example Cycle 8

Instruction status

Instruction	j	k	Issue	Execution complete	Write Result
LD F6	34+	R2	1	3	4
LD F2	45+	R3	2	5	6
MULT F0	F2	F4	3		
SUBD F8	F6	F2	4	8	
DIVD F10	F0	F6	5		
ADDD F6	F8	F2	6		

	Busy	Address
Load1	No	
Load2	No	
Load3	No	

Reservation Stations

Time	Name	Busy	Op	S1 Vj	S2 Vk	RS for j Qj	RS for k Qk
0	Add1	Yes	SUBD	M(34+R2)	M(45+R3)		
0	Add2	Yes	ADDD		M(45+R3)	Add1	
	Add3	No					
8	Mult1	Yes	MULTD	M(45+R3)	R(F4)		
0	Mult2	Yes	DIVD		M(34+R2)	Mult1	

Register result status

Clock	F0	F2	F4	F6	F8	F10	F12	...	F30
8	Mult1	M(45+R3)		Add2	Add1	Mult2			

FU

Tomasulo Example Cycle 9

Instruction status

Instruction	j	k	Issue	Execution complete	Write Result
LD F6	34+	R2	1	3	4
LD F2	45+	R3	2	5	6
MULT F0	F2	F4	3		
SUBD F8	F6	F2	4	8	9
DIVD F10	F0	F6	5		
ADDD F6	F8	F2	6		

	Busy	Address
Load1	No	
Load2	No	
Load3	No	

Reservation Stations

Time	Name	Busy	Op	S1 Vj	S2 Vk	RS for j Qj	RS for k Qk
0	Add1	No					
0	Add2	Yes	ADDD	M()–M()	M(45+R3)		
	Add3	No					
7	Mult1	Yes	MULTD	M(45+R3)	R(F4)		
0	Mult2	Yes	DIVD		M(34+R2)	Mult1	

Register result status

Clock	F0	F2	F4	F6	F8	F10	F12	...	F30
9	Mult1	M(45+R3)		Add2	M()–M()	Mult2			

FU

Tomasulo Example Cycle 10

Instruction status

Instruction	j	k	Issue	Execution complete	Write Result
LD F6	34+	R2	1	3	4
LD F2	45+	R3	2	5	6
MULT F0	F2	F4	3		
SUBD F8	F6	F2	4	8	9
DIVD F10	F0	F6	5		
ADDD F6	F8	F2	6		

	Busy	Address
Load1	No	
Load2	No	
Load3	No	

Reservation Stations

Time	Name	Busy	Op	S1 Vj	S2 Vk	RS for j Qj	RS for k Qk
0	Add1	No					
2	Add2	Yes	ADDD	M()–M()	M(45+R3)		
	Add3	No					
7	Mult1	Yes	MULTD	M(45+R3)	R(F4)		
0	Mult2	Yes	DIVD		M(34+R2)	Mult1	

Register result status

Clock	F0	F2	F4	F6	F8	F10	F12	...	F30
10	Mult1	M(45+R3)		Add2	M()–M()	Mult2			

FU

Tomasulo Example Cycle 11

Instruction status

Instruction	j	k	Issue	Execution complete	Write Result
LD F6	34+	R2	1	3	4
LD F2	45+	R3	2	5	6
MULT F0	F2	F4	3		
SUBD F8	F6	F2	4	8	9
DIVD F10	F0	F6	5		
ADDD F6	F8	F2	6		

	Busy	Address
Load1	No	
Load2	No	
Load3	No	

Reservation Stations

Time	Name	Busy	Op	S1 Vj	S2 Vk	RS for j Qj	RS for k Qk
0	Add1	No					
1	Add2	Yes	ADDD	M()–M()	M(45+R3)		
	Add3	No					
5	Mult1	Yes	MULTD	M(45+R3)	R(F4)		
0	Mult2	Yes	DIVD		M(34+R2)	Mult1	

Register result status

Clock	F0	F2	F4	F6	F8	F10	F12	...	F30
11	Mult1	M(45+R3)		Add2	M()–M()	Mult2			

FU

Tomasulo Example Cycle 12

Instruction status

Instruction	j	k	Issue	Execution complete	Write Result
LD F6	34+	R2	1	3	4
LD F2	45+	R3	2	5	6
MULT F0	F2	F4	3		
SUBD F8	F6	F2	4	8	9
DIVD F10	F0	F6	5		
ADDD F6	F8	F2	6	12	

	Busy	Address
Load1	No	
Load2	No	
Load3	No	

Reservation Stations

Time	Name	Busy	Op	S1 Vj	S2 Vk	RS for j Qj	RS for k Qk
0	Add1	No					
0	Add2	Yes	ADDD	M()–M()	M(45+R3)		
	Add3	No					
4	Mult1	Yes	MULTD	M(45+R3)	R(F4)		
0	Mult2	Yes	DIVD		M(34+R2)	Mult1	

Register result status

Clock	F0	F2	F4	F6	F8	F10	F12	...	F30
12 FU	Mult1	M(45+R3)		Add2	M()–M()	Mult2			

Tomasulo Example Cycle 13

Instruction status

Instruction	j	k	Issue	Execution complete	Write Result
LD F6	34+	R2	1	3	4
LD F2	45+	R3	2	5	6
MULT F0	F2	F4	3		
SUBD F8	F6	F2	4	8	9
DIVD F10	F0	F6	5		
ADDD F6	F8	F2	6	12	13

	Busy	Address
Load1	No	
Load2	No	
Load3	No	

Reservation Stations

Time	Name	Busy	Op	S1 Vj	S2 Vk	RS for j Qj	RS for k Qk
0	Add1	No					
0	Add2	No					
	Add3	No					
3	Mult1	Yes	MULTD	M(45+R3)	R(F4)		
0	Mult2	Yes	DIVD		M(34+R2)	Mult1	

Register result status

Clock	F0	F2	F4	F6	F8	F10	F12	...	F30
13 FU	Mult1	M(45+R3)		(M–M)+M()	M()–M()	Mult2			

Tomasulo Example Cycle 14

Instruction status

Instruction	j	k	Issue	Execution complete	Write Result
LD F6	34+	R2	1	3	4
LD F2	45+	R3	2	5	6
MULT F0	F2	F4	3		
SUBD F8	F6	F2	4	8	9
DIVD F10	F0	F6	5		
ADDD F6	F8	F2	6	12	13

	Busy	Address
Load1	No	
Load2	No	
Load3	No	

Reservation Stations

Time	Name	Busy	Op	S1 Vj	S2 Vk	RS for j Qj	RS for k Qk
0	Add1	No					
0	Add2	No					
	Add3	No					
2	Mult1	Yes	MULTD	M(45+R3)	R(F4)		
0	Mult2	Yes	DIVD		M(34+R2)	Mult1	

Register result status

Clock	F0	F2	F4	F6	F8	F10	F12	...	F30
14 FU	Mult1	M(45+R3)		(M–M)+M()	M()–M()	Mult2			

Tomasulo Example Cycle 15

Instruction status

Instruction	j	k	Issue	Execution complete	Write Result
LD F6	34+	R2	1	3	4
LD F2	45+	R3	2	5	6
MULT F0	F2	F4	3		
SUBD F8	F6	F2	4	8	9
DIVD F10	F0	F6	5		
ADDD F6	F8	F2	6	12	13

	Busy	Address
Load1	No	
Load2	No	
Load3	No	

Reservation Stations

Time	Name	Busy	Op	S1 Vj	S2 Vk	RS for j Qj	RS for k Qk
0	Add1	No					
0	Add2	No					
	Add3	No					
1	Mult1	Yes	MULTD	M(45+R3)	R(F4)		
0	Mult2	Yes	DIVD		M(34+R2)	Mult1	

Register result status

Clock	F0	F2	F4	F6	F8	F10	F12	...	F30
15 FU	Mult1	M(45+R3)		(M–M)+M()	M()–M()	Mult2			

Tomasulo Example Cycle 16

Instruction status

Instruction	j	k	Issue	Execution complete	Write Result
LD F6	34+	R2	1	3	4
LD F2	45+	R3	2	5	6
MULT F0	F2	F4	3	16	
SUBD F8	F6	F2	4	8	9
DIVD F10	F0	F6	5		
ADDD F6	F8	F2	6	12	13

	Busy	Address
Load1	No	
Load2	No	
Load3	No	

Reservation Stations

Time	Name	Busy	Op	S1 Vj	S2 Vk	RS for j Qj	RS for k Qk
0	Add1	No					
0	Add2	No					
	Add3	No					
0	Mult1	Yes	MULTD	M(45+R3)	R(F4)		
0	Mult2	Yes	DIVD		M(34+R2)	Mult1	

Register result status

Clock	F0	F2	F4	F6	F8	F10	F12	...	F30
16 FU	Mult1	M(45+R3)		(M–M)+M()	M()–M()	Mult2			

Tomasulo Example Cycle 17

Instruction status

Instruction	j	k	Issue	Execution complete	Write Result
LD F6	34+	R2	1	3	4
LD F2	45+	R3	2	5	6
MULT F0	F2	F4	3	16	17
SUBD F8	F6	F2	4	8	9
DIVD F10	F0	F6	5		
ADDD F6	F8	F2	6	12	13

	Busy	Address
Load1	No	
Load2	No	
Load3	No	

Reservation Stations

Time	Name	Busy	Op	S1 Vj	S2 Vk	RS for j Qj	RS for k Qk
0	Add1	No					
0	Add2	No					
	Add3	No					
0	Mult1	No					
0	Mult2	Yes	DIVD	M*F4	M(34+R2)		

Register result status

Clock	F0	F2	F4	F6	F8	F10	F12	...	F30
17 FU	M*F4	M(45+R3)		(M–M)+M()	M()–M()	Mult2			

Tomasulo Example Cycle 18

Instruction status

Instruction	j	k	Issue	Execution complete	Write Result
LD F6	34+	R2	1	3	4
LD F2	45+	R3	2	5	6
MULT F0	F2	F4	3	16	17
SUBD F8	F6	F2	4	8	9
DIVD F10	F0	F6	5		
ADDD F6	F8	F2	6	12	13

	Busy	Address
Load1	No	
Load2	No	
Load3	No	

Reservation Stations

Time	Name	Busy	Op	S1 Vj	S2 Vk	RS for j Qj	RS for k Qk
0	Add1	No					
0	Add2	No					
	Add3	No					
0	Mult1	No					
40	Mult2	Yes	DIVD	M*F4	M(34+R2)		

Register result status

Clock	F0	F2	F4	F6	F8	F10	F12	...	F30
18 FU	M*F4	M(45+R3)		(M–M)+M()	M()–M()	Mult2			

Tomasulo Example Cycle 19…56

Faster than light computation

Tomasulo Example Cycle 57

Instruction status

Instruction	j	k	Issue	Execution complete	Write Result
LD F6	34+	R2	1	3	4
LD F2	45+	R3	2	5	6
MULT F0	F2	F4	3	16	17
SUBD F8	F6	F2	4	8	9
DIVD F10	F0	F6	5		
ADDD F6	F8	F2	6	12	13

	Busy	Address
Load1	No	
Load2	No	
Load3	No	

Reservation Stations

Time	Name	Busy	Op	S1 Vj	S2 Vk	RS for j Qj	RS for k Qk
0	Add1	No					
0	Add2	No					
	Add3	No					
0	Mult1	No					
1	Mult2	Yes	DIVD	M*F4	M(34+R2)		

Register result status

Clock	F0	F2	F4	F6	F8	F10	F12	...	F30
57 FU	M*F4	M(45+R3)		(M–M)+M()	M()–M()	Mult2			

Tomasulo Example Cycle 58

Instruction status

Instruction	j	k	Issue	Execution complete	Write Result
LD F6	34+	R2	1	3	4
LD F2	45+	R3	2	5	6
MULT F0	F2	F4	3	16	17
SUBD F8	F6	F2	4	8	9
DIVD F10	F0	F6	5	58	
ADDD F6	F8	F2	6	12	13

	Busy	Address
Load1	No	
Load2	No	
Load3	No	

Reservation Stations

Time	Name	Busy	Op	S1 Vj	S2 Vk	RS for j Qj	RS for k Qk
0	Add1	No					
0	Add2	No					
	Add3	No					
0	Mult1	No					
0	Mult2	Yes	DIVD	M*F4	M(34+R2)		

Register result status

Clock	F0	F2	F4	F6	F8	F10	F12	...	F30
58 FU	M*F4	M(45+R3)		(M–M)+M()	M()–M()	Mult2			

Tomasulo Example Cycle 59

Instruction status

Instruction	j	k	Issue	Execution complete	Write Result
LD F6	34+	R2	1	3	4
LD F2	45+	R3	2	5	6
MULT F0	F2	F4	3	16	17
SUBD F8	F6	F2	4	8	9
DIVD F10	F0	F6	5	58	59
ADDD F6	F8	F2	6	12	13

	Busy	Address
Load1	No	
Load2	No	
Load3	No	

Reservation Stations

Time	Name	Busy	Op	S1 Vj	S2 Vk	RS for j Qj	RS for k Qk
0	Add1	No					
0	Add2	No					
	Add3	No					
0	Mult1	No					
0	Mult2	No					

Register result status

Clock	F0	F2	F4	F6	F8	F10	F12	...	F30
59 FU	M*F4	M(45+R3)		(M–M)+M()	M()–M()	M*F4/M			

Tomasulo Loop Example

```
Loop: LD     F0   0   R1
      MULTD  F4   F0  F2
      SD     F4   0   R1
      SUBI   R1   R1  #8
      BNEZ       R1  Loop
```

- Multiply takes 4 clocks
- Load have cache misses

Loop Example Cycle 0

Instruction status

Instruction	j	k	iteration	Issue	Execution complete	Write Result
LD F0	0	R1	1			
MULT F4	F0	F2	1			
SD F4	0	R1	1			
LD F0	0	R1	2			
MULT F4	F0	F2	2			
SD F4	0	R1	2			

	Busy	Address	Qi
Load1	No		
Load2	No		
Load3	No		
Store1	No		
Store2	No		
Store3	No		

Reservation Stations

Time	Name	Busy	Op	S1 Vj	S2 Vk	RS for j Qj	RS for k Qk	Code:
0	Add1	No						LD F0 0 R1
0	Add2	No						MULT F4 F0 F2
0	Add3	No						SD F4 0 R1
0	Mult1	No						SUBI R1 R1 #8
0	Mult2	No						BNEZ R1 Loop

Register result status

Clock	R1		F0	F2	F4	F6	F8	F10	F12 ...	F30
0	80	Qi								

Loop Example Cycle 1

Instruction status

Instruction	j	k	iteration	Issue	Execution complete	Write Result
LD F0	0	R1	1	1		
MULT F4	F0	F2	1			
SD F4	0	R1	1			
LD F0	0	R1	2			
MULT F4	F0	F2	2			
SD F4	0	R1	2			

	Busy	Address	Qi
Load1	Yes	80	
Load2	No		
Load3	No		
Store1	No		
Store2	No		
Store3	No		

Reservation Stations

Time	Name	Busy	Op	S1 Vj	S2 Vk	RS for j Qj	RS for k Qk	Code:
0	Add1	No						LD F0 0 R1
0	Add2	No						MULT F4 F0 F2
0	Add3	No						SD F4 0 R1
0	Mult1	No						SUBI R1 R1 #8
0	Mult2	No						BNEZ R1 Loop

Register result status

Clock	R1		F0	F2	F4	F6	F8	F10	F12 ...	F30
1	80	Qi	Load1							

Loop Example Cycle 2

Instruction status

Instruction	j	k	iteration	Issue	Execution complete	Write Result
LD F0	0	R1	1	1		
MULT F4	F0	F2	1	2		
SD F4	0	R1	1			
LD F0	0	R1	2			
MULT F4	F0	F2	2			
SD F4	0	R1	2			

	Busy	Address	Qi
Load1	Yes	80	
Load2	No		
Load3	No		
Store1	No		
Store2	No		
Store3	No		

Reservation Stations

Time	Name	Busy	Op	S1 Vj	S2 Vk	RS for j Qj	RS for k Qk	Code:
0	Add1	No						LD F0 0 R1
0	Add2	No						MULT F4 F0 F2
0	Add3	No						SD F4 0 R1
0	Mult1	Yes	MULTD		R(F2)	Load1		SUBI R1 R1 #8
0	Mult2	No						BNEZ R1 Loop

Register result status

Clock	R1		F0	F2	F4	F6	F8	F10	F12 ...	F30
2	80	Qi	Load1		Mult1					

Loop Example Cycle 3

Instruction status

Instruction	j	k	iteration	Issue	Execution complete	Write Result
LD F0	0	R1	1	1		
MULT F4	F0	F2	1	2		
SD F4	0	R1	1	3		
LD F0	0	R1	2			
MULT F4	F0	F2	2			
SD F4	0	R1	2			

	Busy	Address	Qi
Load1	Yes	80	
Load2	No		
Load3	No		
Store1	Yes	80	Mult1
Store2	No		
Store3	No		

Reservation Stations

Time	Name	Busy	Op	S1 Vj	S2 Vk	RS for j Qj	RS for k Qk	Code:
0	Add1	No						LD F0 0 R1
0	Add2	No						MULT F4 F0 F2
0	Add3	No						SD F4 0 R1
0	Mult1	Yes	MULTD		R(F2)	Load1		SUBI R1 R1 #8
0	Mult2	No						BNEZ R1 Loop

Register result status

Clock	R1		F0	F2	F4	F6	F8	F10	F12 ...	F30
3	80	Qi	Load1		Mult1					

Loop Example Cycle 4

Instruction status

Instruction	j	k	iteration	Issue	Execution complete	Write Result
LD F0	0	R1	1	1		
MULT F4	F0	F2	1	2		
SD F4	0	R1	1	3		
LD F0	0	R1	2			
MULT F4	F0	F2	2			
SD F4	0	R1	2			

	Busy	Address	Qi
Load1	Yes	80	
Load2	No		
Load3	No		
Store1	Yes	80	Mult1
Store2	No		
Store3	No		

Reservation Stations

Time	Name	Busy	Op	S1 Vj	S2 Vk	RS for j Qj	RS for k Qk	Code:
0	Add1	No						LD F0 0 R1
0	Add2	No						MULT F4 F0 F2
0	Add3	No						SD F4 0 R1
0	Mult1	Yes	MULTD		R(F2)	Load1		SUBI R1 R1 #8
0	Mult2	No						BNEZ R1 Loop

Register result status

Clock	R1		F0	F2	F4	F6	F8	F10	F12 ...	F30
4	72	Qi	Load1		Mult1					

Loop Example Cycle 5

Instruction status

Instruction	j	k	iteration	Issue	Execution complete	Write Result
LD F0	0	R1	1	1		
MULT F4	F0	F2	1	2		
SD F4	0	R1	1	3		
LD F0	0	R1	2			
MULT F4	F0	F2	2			
SD F4	0	R1	2			

	Busy	Address	Qi
Load1	Yes	80	
Load2	No		
Load3	No		
Store1	Yes	80	Mult1
Store2	No		
Store3	No		

Reservation Stations

Time	Name	Busy	Op	S1 Vj	S2 Vk	RS for j Qj	RS for k Qk	Code:
0	Add1	No						LD F0 0 R1
0	Add2	No						MULT F4 F0 F2
0	Add3	No						SD F4 0 R1
0	Mult1	Yes	MULTD		R(F2)	Load1		SUBI R1 R1 #8
0	Mult2	No						BNEZ R1 Loop

Register result status

Clock	R1		F0	F2	F4	F6	F8	F10	F12 ...	F30
5	72	Qi	Load1		Mult1					

Loop Example Cycle 6

Instruction status

Instruction	j	k	iteration	Issue	Execution complete	Write Result
LD F0	0	R1	1	1		
MULT F4	F0	F2	1	2		
SD F4	0	R1	1	3		
LD F0	0	R1	2	6		
MULT F4	F0	F2	2			
SD F4	0	R1	2			

	Busy	Address	Qi
Load1	Yes	80	
Load2	Yes	72	
Load3	No		
Store1	Yes	80	Mult1
Store2	No		
Store3	No		

Reservation Stations

Time	Name	Busy	Op	S1 Vj	S2 Vk	RS for j Qj	RS for k Qk	Code:
0	Add1	No						LD F0 0 R1
0	Add2	No						MULT F4 F0 F2
0	Add3	No						SD F4 0 R1
0	Mult1	Yes	MULTD		R(F2)	Load1		SUBI R1 R1 #8
0	Mult2	No						BNEZ R1 Loop

Register result status

Clock	R1		F0	F2	F4	F6	F8	F10	F12 ...	F30
6	72	Qi	Load2		Mult1					

Loop Example Cycle 7

Instruction status

Instruction	j	k	iteration	Issue	Execution complete	Write Result
LD F0	0	R1	1	1		
MULT F4	F0	F2	1	2		
SD F4	0	R1	1	3		
LD F0	0	R1	2	6		
MULT F4	F0	F2	2	7		
SD F4	0	R1	2			

	Busy	Address	Qi
Load1	Yes	80	
Load2	Yes	72	
Load3	No		
Store1	Yes	80	Mult1
Store2	No		
Store3	No		

Reservation Stations

Time	Name	Busy	Op	S1 Vj	S2 Vk	RS for j Qj	RS for k Qk	Code:
0	Add1	No						LD F0 0 R1
0	Add2	No						MULT F4 F0 F2
0	Add3	No						SD F4 0 R1
0	Mult1	Yes	MULTD		R(F2)	Load1		SUBI R1 R1 #8
0	Mult2	Yes	MULTD		R(F2)	Load2		BNEZ R1 Loop

Register result status

Clock	R1		F0	F2	F4	F6	F8	F10	F12 ...	F30
7	72	Qi	Load2		Mult2					

Loop Example Cycle 8

Instruction status

Instruction	j	k	iteration	Issue	Execution complete	Write Result
LD F0	0	R1	1	1		
MULT F4	F0	F2	1	2		
SD F4	0	R1	1	3		
LD F0	0	R1	2	6		
MULT F4	F0	F2	2	7		
SD F4	0	R1	2	8		

	Busy	Address	Qi
Load1	Yes	80	
Load2	Yes	72	
Load3	No		
Store1	Yes	80	Mult1
Store2	Yes	72	Mult2
Store3	No		

Reservation Stations

Time	Name	Busy	Op	S1 Vj	S2 Vk	RS for j Qj	RS for k Qk	Code:
0	Add1	No						LD F0 0 R1
0	Add2	No						MULT F4 F0 F2
0	Add3	No						SD F4 0 R1
0	Mult1	Yes	MULTD		R(F2)	Load1		SUBI R1 R1 #8
0	Mult2	Yes	MULTD		R(F2)	Load2		BNEZ R1 Loop

Register result status

Clock	R1		F0	F2	F4	F6	F8	F10	F12 ...	F30
8	72	Qi	Load2		Mult2					

Loop Example Cycle 9

Instruction status

Instruction	j	k	iteration	Issue	Execution complete	Write Result
LD F0	0	R1	1	1	9	
MULT F4	F0	F2	1	2		
SD F4	0	R1	1	3		
LD F0	0	R1	2	6		
MULT F4	F0	F2	2	7		
SD F4	0	R1	2	8		

	Busy	Address	Qi
Load1	Yes	80	
Load2	Yes	72	
Load3	No		
Store1	Yes	80	Mult1
Store2	Yes	72	Mult2
Store3	No		

Reservation Stations

Time	Name	Busy	Op	S1 Vj	S2 Vk	RS for j Qj	RS for k Qk	Code:
0	Add1	No						LD F0 0 R1
0	Add2	No						MULT F4 F0 F2
0	Add3	No						SD F4 0 R1
0	Mult1	Yes	MULTD		R(F2)	Load1		SUBI R1 R1 #8
0	Mult2	Yes	MULTD		R(F2)	Load2		BNEZ R1 Loop

Register result status

Clock	R1		F0	F2	F4	F6	F8	F10	F12 ...	F30
9	64	Qi	Load2		Mult2					

Loop Example Cycle 10

Instruction status

Instruction	j	k	iteration	Issue	Execution complete	Write Result
LD F0	0	R1	1	1	9	10
MULT F4	F0	F2	1	2		
SD F4	0	R1	1	3		
LD F0	0	R1	2	6	10	
MULT F4	F0	F2	2	7		
SD F4	0	R1	2	8		

	Busy	Address	Qi
Load1	No		
Load2	Yes	72	
Load3	No		
Store1	Yes	80	Mult1
Store2	Yes	72	Mult2
Store3	No		

Reservation Stations

Time	Name	Busy	Op	S1 Vj	S2 Vk	RS for j Qj	RS for k Qk	Code:
0	Add1	No						LD F0 0 R1
0	Add2	No						MULT F4 F0 F2
0	Add3	No						SD F4 0 R1
4	Mult1	Yes	MULTD	M(80)	R(F2)			SUBI R1 R1 #8
0	Mult2	Yes	MULTD		R(F2)	Load2		BNEZ R1 Loop

Register result status

Clock	R1		F0	F2	F4	F6	F8	F10	F12 ...	F30
10	64	Qi	Load2		Mult2					

Loop Example Cycle 11

Instruction status

Instruction	j	k	iteration	Issue	Execution complete	Write Result
LD F0	0	R1	1	1	9	10
MULT F4	F0	F2	1	2		
SD F4	0	R1	1	3		
LD F0	0	R1	2	6	10	11
MULT F4	F0	F2	2	7		
SD F4	0	R1	2	8		

	Busy	Address	Qi
Load1	No		
Load2	No		
Load3	Yes	64	Qi
Store1	Yes	80	Mult1
Store2	Yes	72	Mult2
Store3	No		

Reservation Stations

Time	Name	Busy	Op	S1 Vj	S2 Vk	RS for j Qj	RS for k Qk	Code:
0	Add1	No						LD F0 0 R1
0	Add2	No						MULT F4 F0 F2
0	Add3	No						SD F4 0 R1
3	Mult1	Yes	MULTD	M(80)	R(F2)			SUBI R1 R1 #8
4	Mult2	Yes	MULTD	M(72)	R(F2)			BNEZ R1 Loop

Register result status

Clock	R1		F0	F2	F4	F6	F8	F10	F12 ...	F30
11	64	Qi	Load3		Mult2					

Loop Example Cycle 12

Instruction status

Instruction	j	k	iteration	Issue	Execution complete	Write Result
LD F0	0	R1	1	1	9	10
MULT F4	F0	F2	1	2		
SD F4	0	R1	1	3		
LD F0	0	R1	2	6	10	11
MULT F4	F0	F2	2	7		
SD F4	0	R1	2	8		

	Busy	Address	
Load1	No		
Load2	No		
Load3	Yes	64	Qi
Store1	Yes	80	Mult1
Store2	Yes	72	Mult2
Store3	No		

Reservation Stations

Time	Name	Busy	Op	S1 Vj	S2 Vk	RS for j Qj	RS for k Qk	Code:
0	Add1	No						LD F0 0 R1
0	Add2	No						MULT F4 F0 F2
0	Add3	No						SD F4 0 R1
2	Mult1	Yes	MULTD	M(80)	R(F2)			SUBI R1 R1 #8
3	Mult2	Yes	MULTD	M(72)	R(F2)			BNEZ R1 Loop

Register result status

Clock	R1	F0	F2	F4	F6	F8	F10	F12 ...	F30
12	64	Qi	Load3		Mult2				

Loop Example Cycle 13

Instruction status

Instruction	j	k	iteration	Issue	Execution complete	Write Result
LD F0	0	R1	1	1	9	10
MULT F4	F0	F2	1	2		
SD F4	0	R1	1	3		
LD F0	0	R1	2	6	10	11
MULT F4	F0	F2	2	7		
SD F4	0	R1	2	8		

	Busy	Address	
Load1	No		
Load2	No		
Load3	Yes	64	Qi
Store1	Yes	80	Mult1
Store2	Yes	72	Mult2
Store3	No		

Reservation Stations

Time	Name	Busy	Op	S1 Vj	S2 Vk	RS for j Qj	RS for k Qk	Code:
0	Add1	No						LD F0 0 R1
0	Add2	No						MULT F4 F0 F2
0	Add3	No						SD F4 0 R1
1	Mult1	Yes	MULTD	M(80)	R(F2)			SUBI R1 R1 #8
2	Mult2	Yes	MULTD	M(72)	R(F2)			BNEZ R1 Loop

Register result status

Clock	R1	F0	F2	F4	F6	F8	F10	F12 ...	F30
13	64	Qi	Load3		Mult2				

Loop Example Cycle 14

Instruction status

Instruction	j	k	iteration	Issue	Execution complete	Write Result
LD F0	0	R1	1	1	9	10
MULT F4	F0	F2	1	2	14	
SD F4	0	R1	1	3		
LD F0	0	R1	2	6	10	11
MULT F4	F0	F2	2	7		
SD F4	0	R1	2	8		

	Busy	Address	
Load1	No		
Load2	No		
Load3	Yes	64	Qi
Store1	Yes	80	Mult1
Store2	Yes	72	Mult2
Store3	No		

Reservation Stations

Time	Name	Busy	Op	S1 Vj	S2 Vk	RS for j Qj	RS for k Qk	Code:
0	Add1	No						LD F0 0 R1
0	Add2	No						MULT F4 F0 F2
0	Add3	No						SD F4 0 R1
0	Mult1	Yes	MULTD	M(80)	R(F2)			SUBI R1 R1 #8
1	Mult2	Yes	MULTD	M(72)	R(F2)			BNEZ R1 Loop

Register result status

Clock	R1	F0	F2	F4	F6	F8	F10	F12 ...	F30
14	64	Qi	Load3		Mult2				

Loop Example Cycle 15

Instruction status

Instruction	j	k	iteration	Issue	Execution complete	Write Result
LD F0	0	R1	1	1	9	10
MULT F4	F0	F2	1	2	14	15
SD F4	0	R1	1	3		
LD F0	0	R1	2	6	10	11
MULT F4	F0	F2	2	7	15	
SD F4	0	R1	2	8		

	Busy	Address	
Load1	No		
Load2	No		
Load3	Yes	64	Qi
Store1	Yes	80	M(80)*R(F
Store2	Yes	72	Mult2
Store3	No		

Reservation Stations

Time	Name	Busy	Op	S1 Vj	S2 Vk	RS for j Qj	RS for k Qk	Code:
0	Add1	No						LD F0 0 R1
0	Add2	No						MULT F4 F0 F2
0	Add3	No						SD F4 0 R1
0	Mult1	No						SUBI R1 R1 #8
0	Mult2	Yes	MULTD	M(72)	R(F2)			BNEZ R1 Loop

Register result status

Clock	R1	F0	F2	F4	F6	F8	F10	F12 ...	F30
15	64	Qi	Load3		Mult2				

Loop Example Cycle 16

Instruction status

Instruction	j	k	iteration	Issue	Execution complete	Write Result		Busy	Address	
LD F0	0	R1	1	1	9	10	Load1	No		
MULT F4	F0	F2	1	2	14	15	Load2	No		
SD F4	0	R1	1	3			Load3	Yes	64	Qi
LD F0	0	R1	2	6	10	11	Store1	Yes	80	M(80)*R(F
MULT F4	F0	F2	2	7	15	16	Store2	Yes	72	M(72)*R(7
SD F4	0	R1	2	8			Store3	No		

Reservation Stations

Time	Name	Busy	Op	S1 Vj	S2 Vk	RS for j Qj	RS for k Qk	Code:			
0	Add1	No						LD	F0	0	R1
0	Add2	No						MULT	F4	F0	F2
0	Add3	No						SD	F4	0	R1
0	Mult1	Yes	MULTD		R(F2)	Load3		SUBI	R1	R1	#8
0	Mult2	No						BNEZ	R1	Loop	

Register result status

Clock	R1		F0	F2	F4	F6	F8	F10	F12 ...	F30
16	64	Qi	Load3		Mult1					

Loop Example Cycle 17

Instruction status

Instruction	j	k	iteration	Issue	Execution complete	Write Result		Busy	Address	
LD F0	0	R1	1	1	9	10	Load1	No		
MULT F4	F0	F2	1	2	14	15	Load2	No		
SD F4	0	R1	1	3			Load3	Yes	64	Qi
LD F0	0	R1	2	6	10	11	Store1	Yes	80	M(80)*R(F
MULT F4	F0	F2	2	7	15	16	Store2	Yes	72	M(72)*R(7
SD F4	0	R1	2	8			Store3	Yes	64	Mult1

Reservation Stations

Time	Name	Busy	Op	S1 Vj	S2 Vk	RS for j Qj	RS for k Qk	Code:			
0	Add1	No						LD	F0	0	R1
0	Add2	No						MULT	F4	F0	F2
0	Add3	No						SD	F4	0	R1
0	Mult1	Yes	MULTD		R(F2)	Load3		SUBI	R1	R1	#8
0	Mult2	No						BNEZ	R1	Loop	

Register result status

Clock	R1		F0	F2	F4	F6	F8	F10	F12 ...	F30
17	64	Qi	Load3		Mult1					

Loop Example Cycle 18

Instruction status

Instruction	j	k	iteration	Issue	Execution complete	Write Result		Busy	Address	
LD F0	0	R1	1	1	9	10	Load1	No		
MULT F4	F0	F2	1	2	14	15	Load2	No		
SD F4	0	R1	1	3	18		Load3	Yes	64	Qi
LD F0	0	R1	2	6	10	11	Store1	Yes	80	M(80)*R(F
MULT F4	F0	F2	2	7	15	16	Store2	Yes	72	M(72)*R(7
SD F4	0	R1	2	8			Store3	Yes	64	Mult1

Reservation Stations

Time	Name	Busy	Op	S1 Vj	S2 Vk	RS for j Qj	RS for k Qk	Code:			
0	Add1	No						LD	F0	0	R1
0	Add2	No						MULT	F4	F0	F2
0	Add3	No						SD	F4	0	R1
0	Mult1	Yes	MULTD		R(F2)	Load3		SUBI	R1	R1	#8
0	Mult2	No						BNEZ	R1	Loop	

Register result status

Clock	R1		F0	F2	F4	F6	F8	F10	F12 ...	F30
18	56	Qi	Load3		Mult1					

Loop Example Cycle 19

Instruction status

Instruction	j	k	iteration	Issue	Execution complete	Write Result		Busy	Address	
LD F0	0	R1	1	1	9	10	Load1	No		
MULT F4	F0	F2	1	2	14	15	Load2	No		
SD F4	0	R1	1	3	18	19	Load3	Yes	64	Qi
LD F0	0	R1	2	6	10	11	Store1	No		
MULT F4	F0	F2	2	7	15	16	Store2	Yes	72	M(72)*R(7
SD F4	0	R1	2	8			Store3	Yes	64	Mult1

Reservation Stations

Time	Name	Busy	Op	S1 Vj	S2 Vk	RS for j Qj	RS for k Qk	Code:			
0	Add1	No						LD	F0	0	R1
0	Add2	No						MULT	F4	F0	F2
0	Add3	No						SD	F4	0	R1
0	Mult1	Yes	MULTD		R(F2)	Load3		SUBI	R1	R1	#8
0	Mult2	No						BNEZ	R1	Loop	

Register result status

Clock	R1		F0	F2	F4	F6	F8	F10	F12 ...	F30
19	56	Qi	Load3		Mult1					

Loop Example Cycle 20

Instruction status

Instruction	j	k	iteration	Issue	Execution complete	Write Result	
LD F0	0	R1	1	1	9	10	Load1
MULT F4	F0	F2	1	2	14	15	Load2
SD F4	0	R1	1	3	18	19	Load3
LD F0	0	R1	2	6	10	11	Store1
MULT F4	F0	F2	2	7	15	16	Store2
SD F4	0	R1	2	8	20		Store3

	Busy	Address	
Load1	No		
Load2	No		
Load3	Yes	64	Qi
Store1	No		
Store2	Yes	72	M(72)*R(7
Store3	Yes	64	Mult1

Reservation Stations

Time	Name	Busy	Op	S1 Vj	S2 Vk	RS for j Qj	RS for k Qk
0	Add1	No					
0	Add2	No					
0	Add3	No					
0	Mult1	Yes	MULTD		R(F2)	Load3	
0	Mult2	No					

Code:

```
LD   F0   0  R1
MULT F4   F0 F2
SD   F4   0  R1
SUBI R1   R1 #8
BNEZ R1   Loop
```

Register result status

Clock	R1		F0	F2	F4	F6	F8	F10	F12 ...	F30
20	56	Qi	Load3		Mult1					

Loop Example Cycle 21

Instruction status

Instruction	j	k	iteration	Issue	Execution complete	Write Result	
LD F0	0	R1	1	1	9	10	Load1
MULT F4	F0	F2	1	2	14	15	Load2
SD F4	0	R1	1	3	18	19	Load3
LD F0	0	R1	2	6	10	11	Store1
MULT F4	F0	F2	2	7	15	16	Store2
SD F4	0	R1	2	8	20	21	Store3

	Busy	Address	
Load1	No		
Load2	No		
Load3	Yes	64	Qi
Store1	No		
Store2	No		
Store3	Yes	64	Mult1

Reservation Stations

Time	Name	Busy	Op	S1 Vj	S2 Vk	RS for j Qj	RS for k Qk
0	Add1	No					
0	Add2	No					
0	Add3	No					
0	Mult1	Yes	MULTD		R(F2)	Load3	
0	Mult2	No					

Code:

```
LD   F0   0  R1
MULT F4   F0 F2
SD   F4   0  R1
SUBI R1   R1 #8
BNEZ R1   Loop
```

Register result status

Clock	R1		F0	F2	F4	F6	F8	F10	F12 ...	F30
21	56	Qi	Load3		Mult1					

Resumo do Algoritmo de Tomasulo

- Evita o gargalo de registradores
- Evita hazards de WAR e WAW no Scoreboard
- Permite *loop unrolling* em HW
- Não está limitado a blocos básicos (desde que haja *branch prediction*)
- Contribuições que são usadas até hoje
  - Escalonamento dinâmico
  - *Register renaming*
  - Tratamento separado de *load/store*
- Próximo assunto: Mais *branch prediction*