

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

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)

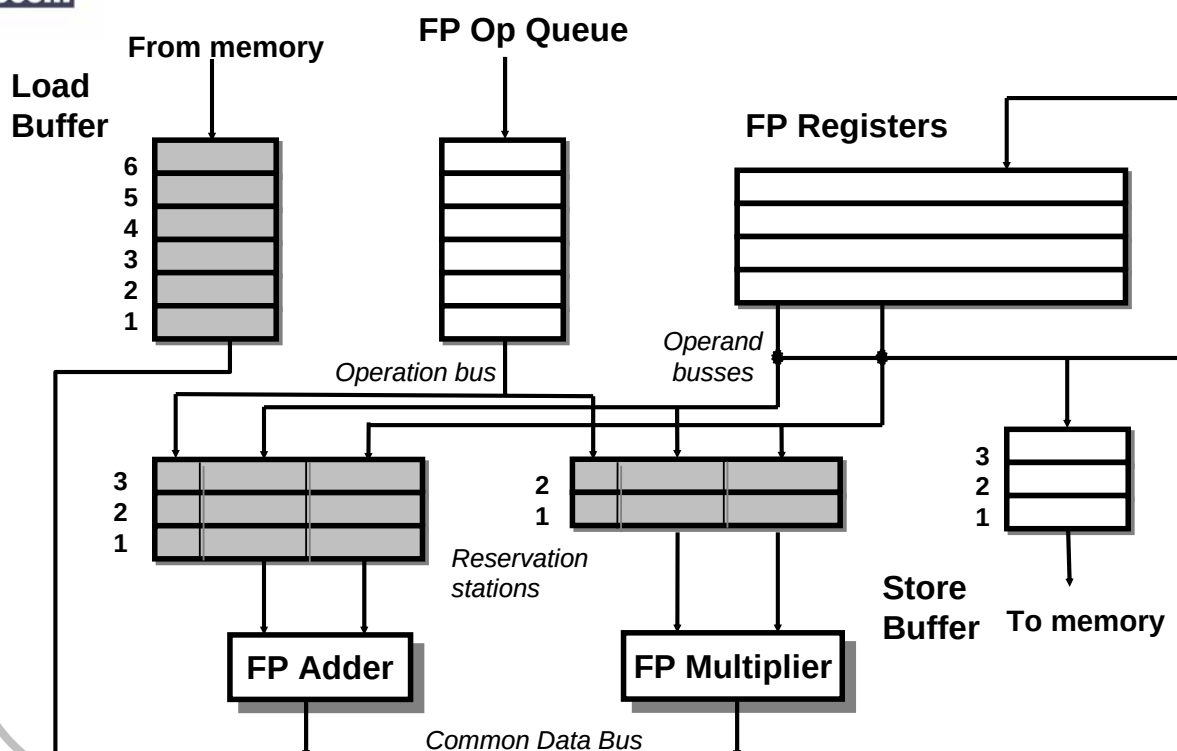
Algoritmo de Tomasulo

- 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 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

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

Tag

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 Q_j (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 V_j e V_k 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 Q_j , Q_k e Q_i , escreva o valor e desmarque o tag



Tomasulo Example Cycle 0

Instruction status

Instruction	<i>j</i>	<i>k</i>	Issue	Execution complete	Write Result	Load1	Busy	Address
LD F6 34+ R2							No	
LD F2 45+ R3							No	
MULT F0 F2 F4							No	
SUBD F8 F6 F2								
DIVD F10 F0 F6								
ADDD F6 F8 F2								

Reservation Stations

Time	Name	Busy	Op	<i>S1</i> <i>Vj</i>	<i>S2</i> <i>Vk</i>	<i>RS for j</i> <i>Qj</i>	<i>RS for k</i> <i>Qk</i>
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	<i>j</i>	<i>k</i>	Issue	Execution complete	Write Result	Load1	Busy	Address
LD F6 34+ R2			1				Yes	34+R2
LD F2 45+ R3							No	
MULT F0 F2 F4							No	
SUBD F8 F6 F2								
DIVD F10 F0 F6								
ADDD F6 F8 F2								

Reservation Stations

Time	Name	Busy	Op	<i>S1</i> <i>Vj</i>	<i>S2</i> <i>Vk</i>	<i>RS for j</i> <i>Qj</i>	<i>RS for k</i> <i>Qk</i>
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	Load	Busy	Address
LD F6 34+ R2	1					Load1	Yes	34+R2
LD F2 45+ R3	2					Load2	Yes	45+R3
MULT F0 F2 F4						Load3	No	
SUBD F8 F6 F2								
DIVD F10 F0 F6								
ADDD F6 F8 F2								

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	Load	Busy	Address
LD F6 34+ R2	1			3		Load1	Yes	34+R2
LD F2 45+ R3	2					Load2	Yes	45+R3
MULT F0 F2 F4	3					Load3	No	
SUBD F8 F6 F2								
DIVD F10 F0 F6								
ADDD F6 F8 F2								

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				Execution			Write			
Instruction	j	k	Issue	complete	Result	Load1	Load2	Load3	Busy	Address
LD F6 34+	R2		1	3	4	No			No	
LD F2 45+	R3		2	4		Yes	45+R3		Yes	
MULT F0 F2	F4		3			No			No	
SUBD F8 F6	F2		4							
DIVD F10 F0	F6									
ADDD F6 F8	F2									

Reservation Stations				S1	S2	RS for j	RS for k
Time	Name	Busy	Op	Vj	Vk	Qj	Qk
0	Add1	Yes	SUBD	M(34+R2)			Load2
0	Add2	No					
0	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				Execution			Write			
Instruction	j	k	Issue	complete	Result	Load1	Load2	Load3	Busy	Address
LD F6 34+	R2		1	3	4	No			No	
LD F2 45+	R3		2	5		Yes	45+R3		Yes	
MULT F0 F2	F4		3			No			No	
SUBD F8 F6	F2		4							
DIVD F10 F0	F6		5							
ADDD F6 F8	F2									

Reservation Stations				S1	S2	RS for j	RS for k
Time	Name	Busy	Op	Vj	Vk	Qj	Qk
0	Add1	Yes	SUBD	M(34+R2)			Load2
0	Add2	No					
0	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	<i>j</i>	<i>k</i>	Issue	Execution complete	Write Result	Busy	Address
LD F6 34+ R2	1	3	4			Load1	No
LD F2 45+ R3	2	5	6			Load2	No
MULT F0 F2 F4	3					Load3	No
SUBD F8 F6 F2	4						
DIVD F10 F0 F6	5						
ADDD F6 F8 F2	6						

Reservation Stations

Time	Name	Busy	Op	S1 <i>V_j</i>	S2 <i>V_k</i>	RS for <i>j</i> <i>Q_j</i>	RS for <i>k</i> <i>Q_k</i>
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	<i>j</i>	<i>k</i>	Issue	Execution complete	Write Result	Busy	Address
LD F6 34+ R2	1	3	4			Load1	No
LD F2 45+ R3	2	5	6			Load2	No
MULT F0 F2 F4	3					Load3	No
SUBD F8 F6 F2	4						
DIVD F10 F0 F6	5						
ADDD F6 F8 F2	6						

Reservation Stations

Time	Name	Busy	Op	S1 <i>V_j</i>	S2 <i>V_k</i>	RS for <i>j</i> <i>Q_j</i>	RS for <i>k</i> <i>Q_k</i>
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	<i>j</i>	<i>k</i>	Issue	Execution complete	Write Result	Busy	Address
LD F6 34+ R2			1	3	4	Load1	No
LD F2 45+ R3			2	5	6	Load2	No
MULT F0 F2 F4			3			Load3	No
SUBD F8 F6 F2			4	8			
DIVD F10 F0 F6			5				
ADDD F6 F8 F2			6				

Reservation Stations

Time	Name	Busy	Op	S1 <i>V_j</i>	S2 <i>V_k</i>	RS for <i>j</i> <i>Q_j</i>	RS for <i>k</i> <i>Q_k</i>
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	FU	Mult1	M(45+R3)		Add2	Add1	Mult2			



Tomasulo Example Cycle 9

Instruction status

Instruction	<i>j</i>	<i>k</i>	Issue	Execution complete	Write Result	Busy	Address
LD F6 34+ R2			1	3	4	Load1	No
LD F2 45+ R3			2	5	6	Load2	No
MULT F0 F2 F4			3			Load3	No
SUBD F8 F6 F2			4	8	9		
DIVD F10 F0 F6			5				
ADDD F6 F8 F2			6				

Reservation Stations

Time	Name	Busy	Op	S1 <i>V_j</i>	S2 <i>V_k</i>	RS for <i>j</i> <i>Q_j</i>	RS for <i>k</i> <i>Q_k</i>
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	FU	Mult1	M(45+R3)		Add2	M()–M()	Mult2			



Tomasulo Example Cycle 10

Instruction status

Instruction	<i>j</i>	<i>k</i>	Issue	Execution complete	Write Result	Busy	Address
LD F6 34+ R2			1	3	4	Load1	No
LD F2 45+ R3			2	5	6	Load2	No
MULT F0 F2 F4			3			Load3	No
SUBD F8 F6 F2			4	8	9		
DIVD F10 F0 F6			5				
ADDD F6 F8 F2			6				

Reservation Stations

Time	Name	Busy	Op	S1 <i>V_j</i>	S2 <i>V_k</i>	RS for <i>j</i> <i>Q_j</i>	RS for <i>k</i> <i>Q_k</i>
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	FU	Mult1	M(45+R3)		Add2	M()–M()	Mult2		



Tomasulo Example Cycle 11

Instruction status

Instruction	<i>j</i>	<i>k</i>	Issue	Execution complete	Write Result	Busy	Address
LD F6 34+ R2			1	3	4	Load1	No
LD F2 45+ R3			2	5	6	Load2	No
MULT F0 F2 F4			3			Load3	No
SUBD F8 F6 F2			4	8	9		
DIVD F10 F0 F6			5				
ADDD F6 F8 F2			6				

Reservation Stations

Time	Name	Busy	Op	S1 <i>V_j</i>	S2 <i>V_k</i>	RS for <i>j</i> <i>Q_j</i>	RS for <i>k</i> <i>Q_k</i>
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	FU	Mult1	M(45+R3)		Add2	M()–M()	Mult2		



Tomasulo Example Cycle 12

Instruction status

Instruction	<i>j</i>	<i>k</i>	Issue	Execution complete	Write Result	Busy	Address
LD F6 34+ R2			1	3	4	Load1	No
LD F2 45+ R3			2	5	6	Load2	No
MULT F0 F2 F4			3			Load3	No
SUBD F8 F6 F2			4	8	9		
DIVD F10 F0 F6			5				
ADDD F6 F8 F2			6	12			

Reservation Stations

Time	Name	Busy	Op	S1 <i>V_j</i>	S2 <i>V_k</i>	RS for <i>j</i> <i>Q_j</i>	RS for <i>k</i> <i>Q_k</i>
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	<i>j</i>	<i>k</i>	Issue	Execution complete	Write Result	Busy	Address
LD F6 34+ R2			1	3	4	Load1	No
LD F2 45+ R3			2	5	6	Load2	No
MULT F0 F2 F4			3			Load3	No
SUBD F8 F6 F2			4	8	9		
DIVD F10 F0 F6			5				
ADDD F6 F8 F2			6	12	13		

Reservation Stations

Time	Name	Busy	Op	S1 <i>V_j</i>	S2 <i>V_k</i>	RS for <i>j</i> <i>Q_j</i>	RS for <i>k</i> <i>Q_k</i>
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	<i>j</i>	<i>k</i>	Issue	Execution complete	Write Result	Busy	Address
LD F6 34+ R2			1	3	4	Load1	No
LD F2 45+ R3			2	5	6	Load2	No
MULT F0 F2 F4			3			Load3	No
SUBD F8 F6 F2			4	8	9		
DIVD F10 F0 F6			5				
ADDD F6 F8 F2			6	12	13		

Reservation Stations

Time	Name	Busy	Op	S1 <i>V_j</i>	S2 <i>V_k</i>	RS for <i>j</i> <i>Q_j</i>	RS for <i>k</i> <i>Q_k</i>
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	<i>j</i>	<i>k</i>	Issue	Execution complete	Write Result	Busy	Address
LD F6 34+ R2			1	3	4	Load1	No
LD F2 45+ R3			2	5	6	Load2	No
MULT F0 F2 F4			3			Load3	No
SUBD F8 F6 F2			4	8	9		
DIVD F10 F0 F6			5				
ADDD F6 F8 F2			6	12	13		

Reservation Stations

Time	Name	Busy	Op	S1 <i>V_j</i>	S2 <i>V_k</i>	RS for <i>j</i> <i>Q_j</i>	RS for <i>k</i> <i>Q_k</i>
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	<i>j</i>	<i>k</i>	Issue	Execution complete	Write Result	Busy	Address
LD F6 34+ R2			1	3	4	Load1	No
LD F2 45+ R3			2	5	6	Load2	No
MULT F0 F2 F4			3	16		Load3	No
SUBD F8 F6 F2			4	8	9		
DIVD F10 F0 F6			5				
ADDD F6 F8 F2			6	12	13		

Reservation Stations

Time	Name	Busy	Op	S1 <i>V_j</i>	S2 <i>V_k</i>	RS for <i>j</i> <i>Q_j</i>	RS for <i>k</i> <i>Q_k</i>
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	<i>j</i>	<i>k</i>	Issue	Execution complete	Write Result	Busy	Address
LD F6 34+ R2			1	3	4	Load1	No
LD F2 45+ R3			2	5	6	Load2	No
MULT F0 F2 F4			3	16	17	Load3	No
SUBD F8 F6 F2			4	8	9		
DIVD F10 F0 F6			5				
ADDD F6 F8 F2			6	12	13		

Reservation Stations

Time	Name	Busy	Op	S1 <i>V_j</i>	S2 <i>V_k</i>	RS for <i>j</i> <i>Q_j</i>	RS for <i>k</i> <i>Q_k</i>
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	<i>j</i>	<i>k</i>	Issue	Execution complete	Write Result	Busy	Address
LD F6 34+ R2			1	3	4	Load1	No
LD F2 45+ R3			2	5	6	Load2	No
MULT F0 F2 F4			3	16	17	Load3	No
SUBD F8 F6 F2			4	8	9		
DIVD F10 F0 F6			5				
ADDD F6 F8 F2			6	12	13		

Reservation Stations

Time	Name	Busy	Op	S1 <i>V_j</i>	S2 <i>V_k</i>	RS for <i>j</i> <i>Q_j</i>	RS for <i>k</i> <i>Q_k</i>
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 57

Instruction status

Instruction	<i>j</i>	<i>k</i>	Issue	Execution complete	Write Result	Busy	Address
LD F6 34+ R2			1	3	4	Load1	No
LD F2 45+ R3			2	5	6	Load2	No
MULT F0 F2 F4			3	16	17	Load3	No
SUBD F8 F6 F2			4	8	9		
DIVD F10 F0 F6			5				
ADDD F6 F8 F2			6	12	13		

Reservation Stations

Time	Name	Busy	Op	S1 <i>V_j</i>	S2 <i>V_k</i>	RS for <i>j</i> <i>Q_j</i>	RS for <i>k</i> <i>Q_k</i>
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	<i>j</i>	<i>k</i>	Issue	Execution complete	Write Result	Busy	Address
LD F6 34+ R2			1	3	4	Load1	No
LD F2 45+ R3			2	5	6	Load2	No
MULT F0 F2 F4			3	16	17	Load3	No
SUBD F8 F6 F2			4	8	9		
DIVD F10 F0 F6			5	58			
ADDD F6 F8 F2			6	12	13		

Reservation Stations

Time	Name	Busy	Op	S1 <i>V_j</i>	S2 <i>V_k</i>	RS for <i>j</i> <i>Q_j</i>	RS for <i>k</i> <i>Q_k</i>
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	<i>j</i>	<i>k</i>	Issue	Execution complete	Write Result	Busy	Address
LD F6 34+ R2			1	3	4	Load1	No
LD F2 45+ R3			2	5	6	Load2	No
MULT F0 F2 F4			3	16	17	Load3	No
SUBD F8 F6 F2			4	8	9		
DIVD F10 F0 F6			5	58	59		
ADDD F6 F8 F2			6	12	13		

Reservation Stations

Time	Name	Busy	Op	S1 <i>V_j</i>	S2 <i>V_k</i>	RS for <i>j</i> <i>Q_j</i>	RS for <i>k</i> <i>Q_k</i>
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				Execution Write					
Instruction	j	k	iteration	Issue	complete	Result	Busy	Address	
LD F0	0	R1	1				No		
MULT F4	F0	F2	1				No		
SD F4	0	R1	1				No	Qi	
LD F0	0	R1	2				No		
MULT F4	F0	F2	2				No		
SD F4	0	R1	2				No		

Reservation Stations				S1	S2	RS for j	RS for k		
Time	Name	Busy	Op	Vj	Vk	Qj	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		F0	F2	F4	F6	F8	F10	F12...	F30
Clock	R1								
0	80 Qi								

Loop Example Cycle 1

Instruction status				Execution Write			Busy	Address
Instruction	j	k	iteration	Issue	complete	Result		
LD F0	0	R1	1	1		Load1	Yes	80
MULT F4	F0	F2	1			Load2	No	
SD F4	0	R1	1			Load3	No	Qi
LD F0	0	R1	2			Store1	No	
MULT F4	F0	F2	2			Store2	No	
SD F4	0	R1	2			Store3	No	
Reservation Stations				S1	S2	RS for j	RS for k	Code:
Time	Name	Busy	Op	Vj	Vk	Qj	Qk	
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				Execution Write			Busy	Address
Instruction	j	k	iteration	Issue	complete	Result		
LD F0	0	R1	1	1		Load1	Yes	80
MULT F4	F0	F2	1	2		Load2	No	Qi
SD F4	0	R1	1			Load3	No	
LD F0	0	R1	2			Store1	No	
MULT F4	F0	F2	2			Store2	No	
SD F4	0	R1	2			Store3	No	
Reservation Stations				S1	S2	RS for j	RS for k	Code:
Time	Name	Busy	Op	Vj	Vk	Qj	Qk	
0	Add1	No			R(F2)	Load1	LD F0	0 R1
0	Add2	No					MULT F4	F0 F2
0	Add3	No					SD F4	0 R1
0	Mult1	Yes					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				Execution Write				
Instruction	j	k	iteration	Issue	complete	Result	Busy	Address
LD F0	0	R1	1	1		Load1	Yes	80
MULT F4	F0	F2	1	2		Load2	No	
SD F4	0	R1	1	3		Load3	No	Qi
LD F0	0	R1	2			Store1	Yes	80
MULT F4	F0	F2	2			Store2	No	Mult1
SD F4	0	R1	2			Store3	No	

Reservation Stations				S1	S2	RS for j	RS for k	Code:
Time	Name	Busy	Op	Vj	Vk	Qj	Qk	
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		F0	F2	F4	F6	F8	F10	F12...	F30
Clock	R1								
3	80	Qi	Load1	Mult1					

Loop Example Cycle 4

Instruction status				Execution Write				
Instruction	j	k	iteration	Issue	complete	Result	Busy	Address
LD F0	0	R1	1	1		Load1	Yes	80
MULT F4	F0	F2	1	2		Load2	No	
SD F4	0	R1	1	3		Load3	No	Qi
LD F0	0	R1	2			Store1	Yes	80
MULT F4	F0	F2	2			Store2	No	Mult1
SD F4	0	R1	2			Store3	No	

Reservation Stations				S1	S2	RS for j	RS for k	Code:
Time	Name	Busy	Op	Vj	Vk	Qj	Qk	
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		F0	F2	F4	F6	F8	F10	F12...	F30
Clock	R1								
4	72	Qi	Load1	Mult1					

Loop Example Cycle 5

Instruction status				Execution Write			Busy	Address
Instruction	j	k	iteration	Issue	complete	Result		
LD F0	0	R1	1	1		Load1	Yes	80
MULT F4	F0	F2	1	2		Load2	No	
SD F4	0	R1	1	3		Load3	No	Qi
LD F0	0	R1	2			Store1	Yes	80
MULT F4	F0	F2	2			Store2	No	
SD F4	0	R1	2			Store3	No	

Reservation Stations				S1	S2	RS for j	RS for k	Code:
Time	Name	Busy	Op	Vj	Vk	Qj	Qk	
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				Execution Write			Busy	Address
Instruction	j	k	iteration	Issue	complete	Result		
LD F0	0	R1	1	1		Load1	Yes	80
MULT F4	F0	F2	1	2		Load2	Yes	72
SD F4	0	R1	1	3		Load3	No	Qi
LD F0	0	R1	2	6		Store1	Yes	80
MULT F4	F0	F2	2			Store2	No	
SD F4	0	R1	2			Store3	No	

Reservation Stations				S1	S2	RS for j	RS for k	Code:
Time	Name	Busy	Op	Vj	Vk	Qj	Qk	
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				Execution Write				
Instruction	j	k	iteration	Issue	complete	Result	Busy	Address
LD F0	0	R1	1	1		Load1	Yes	80
MULT F4	F0	F2	1	2		Load2	Yes	72
SD F4	0	R1	1	3		Load3	No	Qi
LD F0	0	R1	2	6		Store1	Yes	80
MULT F4	F0	F2	2	7		Store2	No	
SD F4	0	R1	2			Store3	No	

Reservation Stations				S1	S2	RS for j	RS for k	Code:
Time	Name	Busy	Op	Vj	Vk	Qj	Qk	
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				Execution Write				
Instruction	j	k	iteration	Issue	complete	Result	Busy	Address
LD F0	0	R1	1	1		Load1	Yes	80
MULT F4	F0	F2	1	2		Load2	Yes	72
SD F4	0	R1	1	3		Load3	No	Qi
LD F0	0	R1	2	6		Store1	Yes	80
MULT F4	F0	F2	2	7		Store2	Yes	72
SD F4	0	R1	2	8		Store3	No	

Reservation Stations				S1	S2	RS for j	RS for k	Code:
Time	Name	Busy	Op	Vj	Vk	Qj	Qk	
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				Execution Write			Busy	Address
Instruction	j	k	iteration	Issue	complete	Result		
LD F0	0	R1	1	1	9	Load1	Yes	80
MULT F4	F0	F2	1	2		Load2	Yes	72
SD F4	0	R1	1	3		Load3	No	Qi
LD F0	0	R1	2	6		Store1	Yes	80
MULT F4	F0	F2	2	7		Store2	Yes	72
SD F4	0	R1	2	8		Store3	No	

Reservation Stations				S1	S2	RS for j	RS for k	Code:
Time	Name	Busy	Op	Vj	Vk	Qj	Qk	
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				Execution Write			Busy	Address
Instruction	j	k	iteration	Issue	complete	Result		
LD F0	0	R1	1	1	9	10	No	
MULT F4	F0	F2	1	2			Yes	72
SD F4	0	R1	1	3			No	Qi
LD F0	0	R1	2	6	10		Yes	80
MULT F4	F0	F2	2	7			Yes	72
SD F4	0	R1	2	8			No	

Reservation Stations				S1	S2	RS for j	RS for k	Code:
Time	Name	Busy	Op	Vj	Vk	Qj	Qk	
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				Execution Write				
Instruction	j	k	iteration	Issue	complete	Result	Busy	Address
LD F0	0	R1	1	1	9	10	Load1	No
MULT F4	F0	F2	1	2			Load2	No
SD F4	0	R1	1	3			Load3	Yes 64 Qi
LD F0	0	R1	2	6	10	11	Store1	Yes 80 Mult1
MULT F4	F0	F2	2	7			Store2	Yes 72 Mult2
SD F4	0	R1	2	8			Store3	No

Reservation Stations				S1	S2	RS for j	RS for k	Code:
Time	Name	Busy	Op	Vj	Vk	Qj	Qk	
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				Execution Write				
Instruction	j	k	iteration	Issue	complete	Result	Busy	Address
LD F0	0	R1	1	1	9	10	Load1	No
MULT F4	F0	F2	1	2			Load2	No
SD F4	0	R1	1	3			Load3	Yes 64 Qi
LD F0	0	R1	2	6	10	11	Store1	Yes 80 Mult1
MULT F4	F0	F2	2	7			Store2	Yes 72 Mult2
SD F4	0	R1	2	8			Store3	No

Reservation Stations				S1	S2	RS for j	RS for k	Code:
Time	Name	Busy	Op	Vj	Vk	Qj	Qk	
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				Execution Write			Busy	Address	
Instruction	j	k	iteration	Issue	complete	Result			
LD F0	0	R1	1	1	9	10	Load1	No	64 Qi
MULT F4	F0	F2	1	2			Load2	No	
SD F4	0	R1	1	3			Load3	Yes	
LD F0	0	R1	2	6	10	11	Store1	Yes	80 Mult1
MULT F4	F0	F2	2	7			Store2	Yes	72 Mult2
SD F4	0	R1	2	8			Store3	No	

Reservation Stations				S1	S2	RS for j	RS for k	Code:
Time	Name	Busy	Op	Vj	Vk	Qj	Qk	
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				Execution Write			Busy	Address		
Instruction	j	k	iteration	Issue	complete	Result				
LD F0	0	R1	1	1	9	10	Load1	No	Qi	
MULT F4	F0	F2	1	2	14		Load2	No		
SD F4	0	R1	1	3			Load3	Yes		
LD F0	0	R1	2	6	10	11	Store1	Yes	80	Mult1
MULT F4	F0	F2	2	7			Store2	Yes	72	Mult2
SD F4	0	R1	2	8			Store3	No		
Reservation Stations				S1	S2	RS for j	RS for k			
Time	Name	Busy	Op	Vj	Vk	Qj	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	Oi	Load3		Mult2					



Loop Example Cycle 15

Instruction status				Execution Write			Busy	Address
Instruction	j	k	iteration	Issue	complete	Result		
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		Store2	Yes 72 Mult2
SD F4	0	R1	2	8			Store3	No

Reservation Stations				S1	S2	RS for j	RS for k	Code:
Time	Name	Busy	Op	Vj	Vk	Qj	Qk	
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				Execution Write				
Instruction	j	k	iteration	Issue	complete	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				S1	S2	RS for j	RS for k	
Time	Name	Busy	Op	Vj	Vk	Qj	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 Oi	Load3		Mult1						

Loop Example Cycle 17

Instruction status				Execution Write			Busy	Address
Instruction	j	k	iteration	Issue	complete	Result		
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				S1	S2	RS for j	RS for k	Code:
Time	Name	Busy	Op	Vj	Vk	Qj	Qk	
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		F0	F2	F4	F6	F8	F10	F12...	F30
Clock	R1								
17	64 Qi	Load3		Mult1					

Loop Example Cycle 18

Instruction status				Execution Write			Busy	Address
Instruction	j	k	iteration	Issue	complete	Result		
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				S1	S2	RS for j	RS for k	Code:
Time	Name	Busy	Op	Vj	Vk	Qj	Qk	
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		F0	F2	F4	F6	F8	F10	F12...	F30
Clock	R1								
18	56 Qi	Load3		Mult1					

Loop Example Cycle 19

Instruction status				Execution Write			Busy	Address
Instruction	j	k	iteration	Issue	complete	Result		
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(72)
SD F4	0	R1	2	8			Store3	Yes 64 Mult1

Reservation Stations				S1	S2	RS for j	RS for k	Code:
Time	Name	Busy	Op	Vj	Vk	Qj	Qk	
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		F0	F2	F4	F6	F8	F10	F12...	F30
Clock	R1								
19	56 Qi	Load3		Mult1					

Loop Example Cycle 20

Instruction status				Execution Write			Busy	Address
Instruction	j	k	iteration	Issue	complete	Result		
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(72)
SD F4	0	R1	2	8	20		Store3	Yes 64 Mult1

Reservation Stations				S1	S2	RS for j	RS for k	Code:
Time	Name	Busy	Op	Vj	Vk	Qj	Qk	
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		F0	F2	F4	F6	F8	F10	F12...	F30
Clock	R1								
20	56 Qi	Load3		Mult1					

Loop Example Cycle 21

Instruction status				Execution Write					
Instruction	j	k	iteration	Issue	complete	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	No	
SD F4	0	R1	2	8	20	21	Store3	Yes	64 Mult1

Reservation Stations				S1	S2	RS for j		RS for k	Code:
Time	Name	Busy	Op	Vj	Vk	Qj	Qk	Qk	
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	
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
- Próxima aula: Mais *branch prediction*