

# “Arquitetura e Organização de Computadores I – Aula\_08 – *Pipeline* – Parte 2”

Prof. Dr. Emerson Carlos Pedrino  
DC/UFSCar  
São Carlos

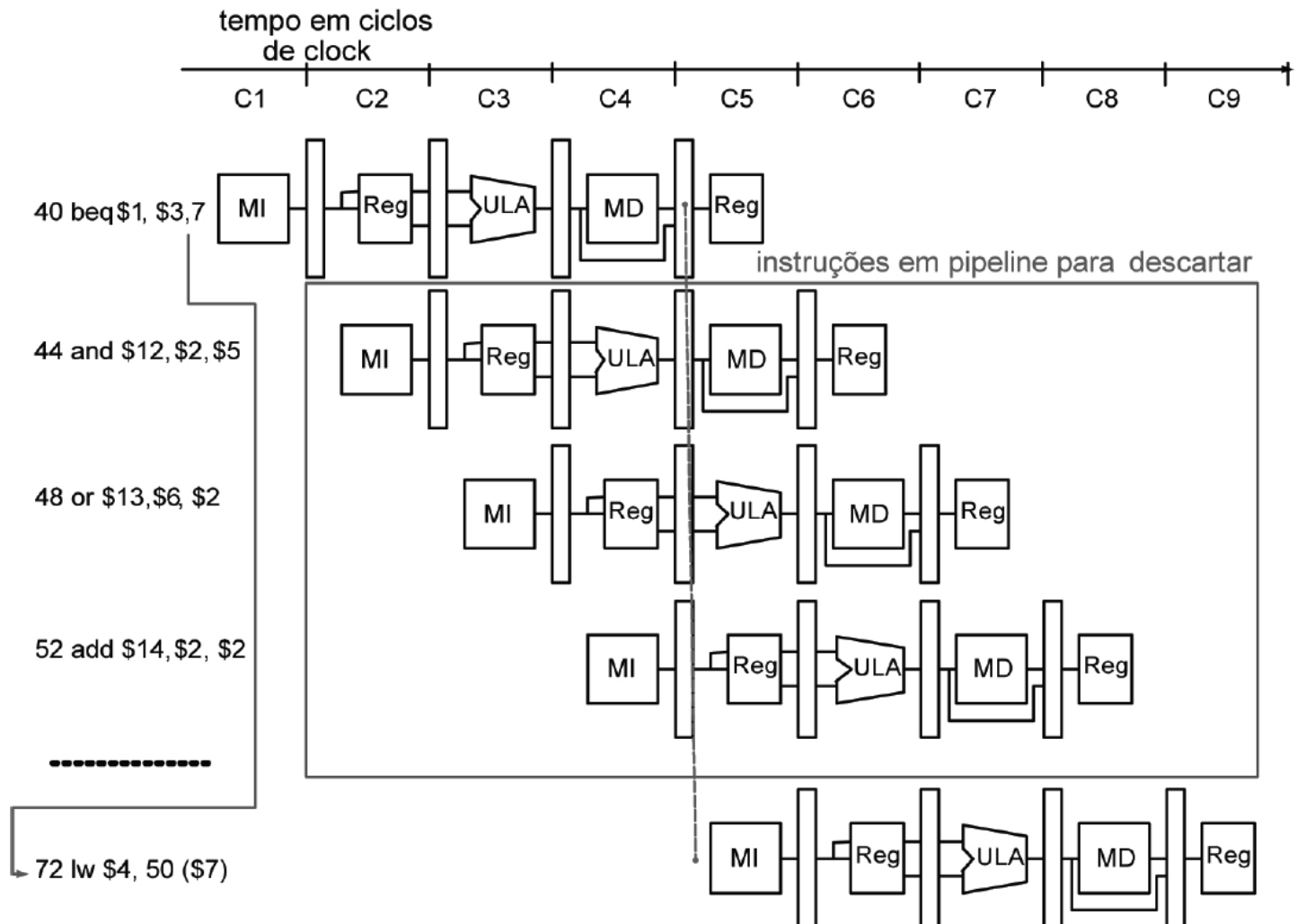




# Problema da Instrução de Desvio

- Confirmação da instrução de desvio condicional: ocorre no estágio M.
- Quando decidido: outras instruções estão *em pipeline*.
- Se não ocorrer o desvio, as instruções subsequentes são necessárias, caso contrário devem ser descartadas.

# Exemplo



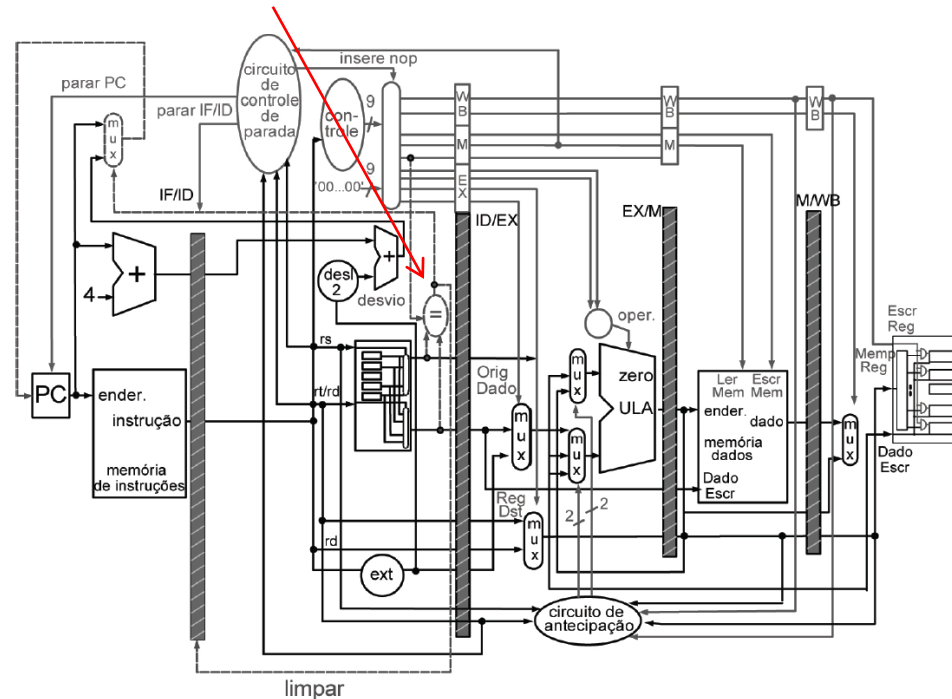


# Solução

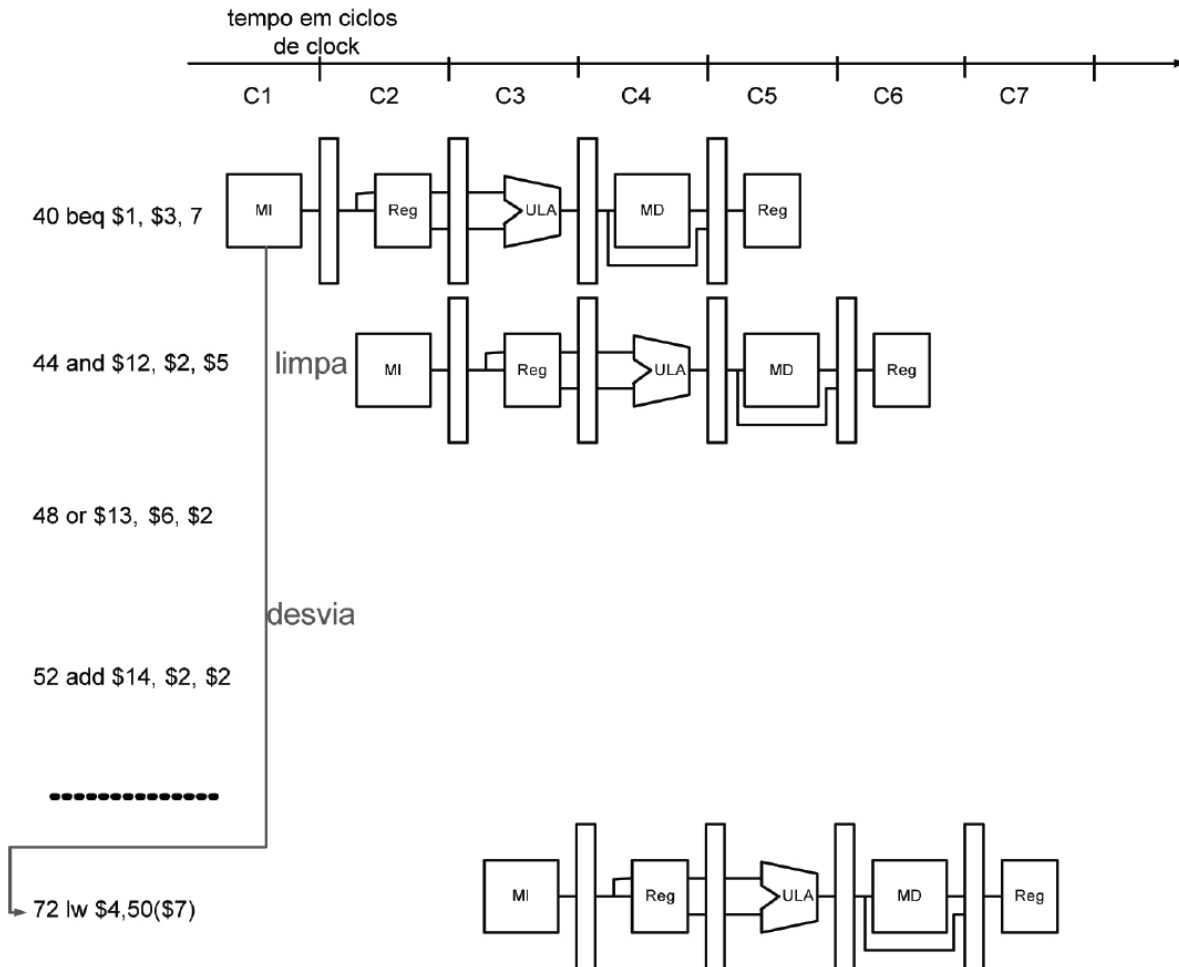
- Circuito para descartar as instruções posteriores caso se confirme o desvio.
- Quanto mais cedo ocorrer a confirmação, melhor. Assim, menos instruções precisarão ser descartadas.
- Tal circuito é inserido no estágio ID para antecipar a verificação da condição de desvio e o cálculo do endereço.

# Exemplo

- Ao detectar a condição, o circuito limpa a instrução lida em IF e atualiza o PC com o endereço de desvio. Assim, apenas uma instrução é descartada.



# Ex: Estágios das instruções após *beq* usando o circuito de detecção





# Observações

- Existem outras formas para melhorar o desempenho dessa arquitetura, como, por exemplo, inserindo uma instrução após *beq*, a qual sempre deve ser executada, haja ou não o desvio.
- Também nesse caso, não é necessário limpar a instrução subsequente.
- Quem deve inserir essa instrução para o *branch delay slot* é o compilador.



# Outras Arquiteturas

- **Processador Superescalar:** a arquitetura superescalar está presente na maioria dos processadores de uso geral, e caracteriza-se pela **capacidade de iniciar e executar mais de uma instrução no mesmo ciclo.**
- **No MIPS** essa **capacidade** pode ser derivada da memória de instruções, permitindo a leitura simultânea de duas instruções.

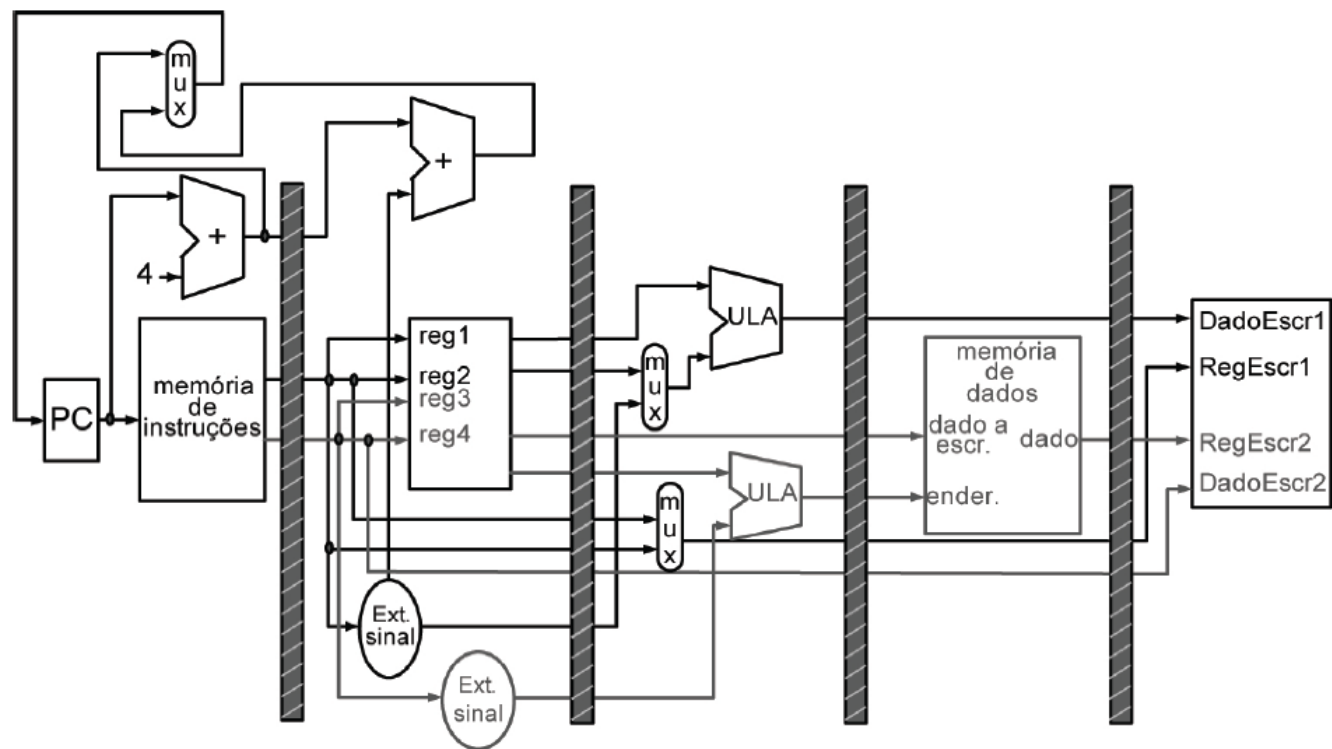




# MIPS Superescalar

- Novo banco de registradores: permissão para 4 leituras de registradores simultaneamente.
- 2 ULAs: 1 para instruções de referência à memória e outra para as demais instruções.
- Assim, no último estágio a escrita de registrador seria duplicada. Logo, as instruções *lw* e *sw* podem ser executadas paralelamente às demais instruções.

# Diagrama Simplificado de um MIPS Superescalar



# Diagrama de Execução de Instruções na Arquitetura Superscalar

Tipo de instrução		Estágios do <i>pipeline</i>							
R ou desvio	}	IF	ID	EX	MEM	WB			
Load/store		IF	ID	EX	MEM	WB			
R ou desvio	}		IF	ID	EX	MEM	WB		
Load/store			IF	ID	EX	MEM	WB		
R ou desvio	}			IF	ID	EX	MEM	WB	
Load/store				IF	ID	EX	MEM	WB	
R ou desvio	}				IF	ID	EX	MEM	WB
Load/store					IF	ID	EX	MEM	WB



# Escalação Dinâmica de Instruções

- Realizada pelo *hardware*. Este tenta encontrar instruções para executar.
- É possível realizar a execução fora da ordem original do programa.
- É possível também a execução especulativa (por exemplo: previsão dinâmica de desvio).



# Previsão Dinâmica de Desvios

- Previsão de desvios durante a **execução**, usando informações em tempo de execução.
- Um **esquema bastante simples** de predição consiste em **utilizar o resultado da última execução da instrução de desvio**.

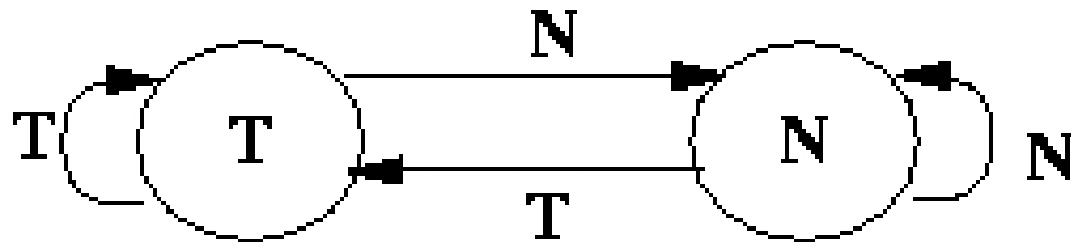


# Previsão Dinâmica de Desvios

- Nesse caso, **um bit** seria suficiente **para** armazenar o resultado anterior da instrução de desvio.
- Se a predição indicar que o desvio deve ser tomado e se o estágio de execução indicar o contrário, a tabela BHT (*Branch History Table*) é atualizada, as instruções nos estágios precedentes são descartadas e o estágio de busca inicia a transferência de instruções pertencentes ao fluxo apropriado.

# Previsão Dinâmica de Desvios

- Autômato para previsão com 1 bit de história:



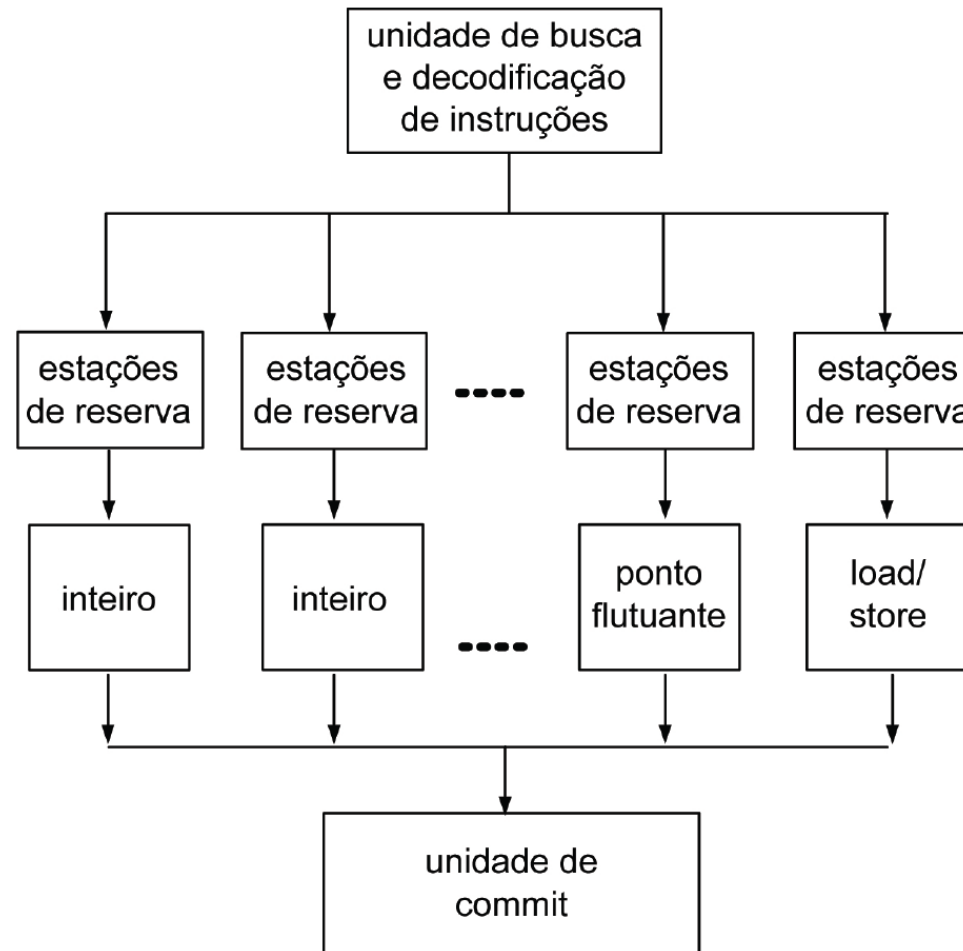


## Exercício\*

- Mostre o problema do Autômato de 1 bit de História Anterior para o caso de *loops*.
- Mostre como deveria ser o novo autômato para resolver o problema.
- Pesquisar outras técnicas de previsão.



# Ex: Diagrama de fluxo de execução de instruções com escalação dinâmica





# Observações

- As instruções decodificadas são despachadas para as unidades funcionais por meio das estações de reserva, onde as instruções aguardam os operandos ficarem disponíveis, momento no qual as instruções começam a execução.
- Após a execução, os resultados são disponibilizados para as estações de reserva que estão no aguardo dos mesmos.
- A unidade de *commit* reordena as ações e libera, quando julgar seguro, os resultados para a memória ou registradores.

# Exemplos de *Hazards* de dados

- **RAW** (*Read After Write*)

- Ex:

- i1.  $R2 \leftarrow R1 + R3$

- i2.  $R4 \leftarrow R2 + R3$

# Exemplos de *Hazards* de dados

- **WAR** (Write After Read)

- Ex:

- i1.  $R4 \leftarrow R1 + R3$

- i2.  $R3 \leftarrow R1 + R2$

# Exemplos de *Hazards* de dados

- **WAW** (Write After Write)

- Ex:

- i1.  $R2 \leftarrow R4 + R7$

- i2.  $R2 \leftarrow R1 + R2$



# Algoritmo de Tomasulo

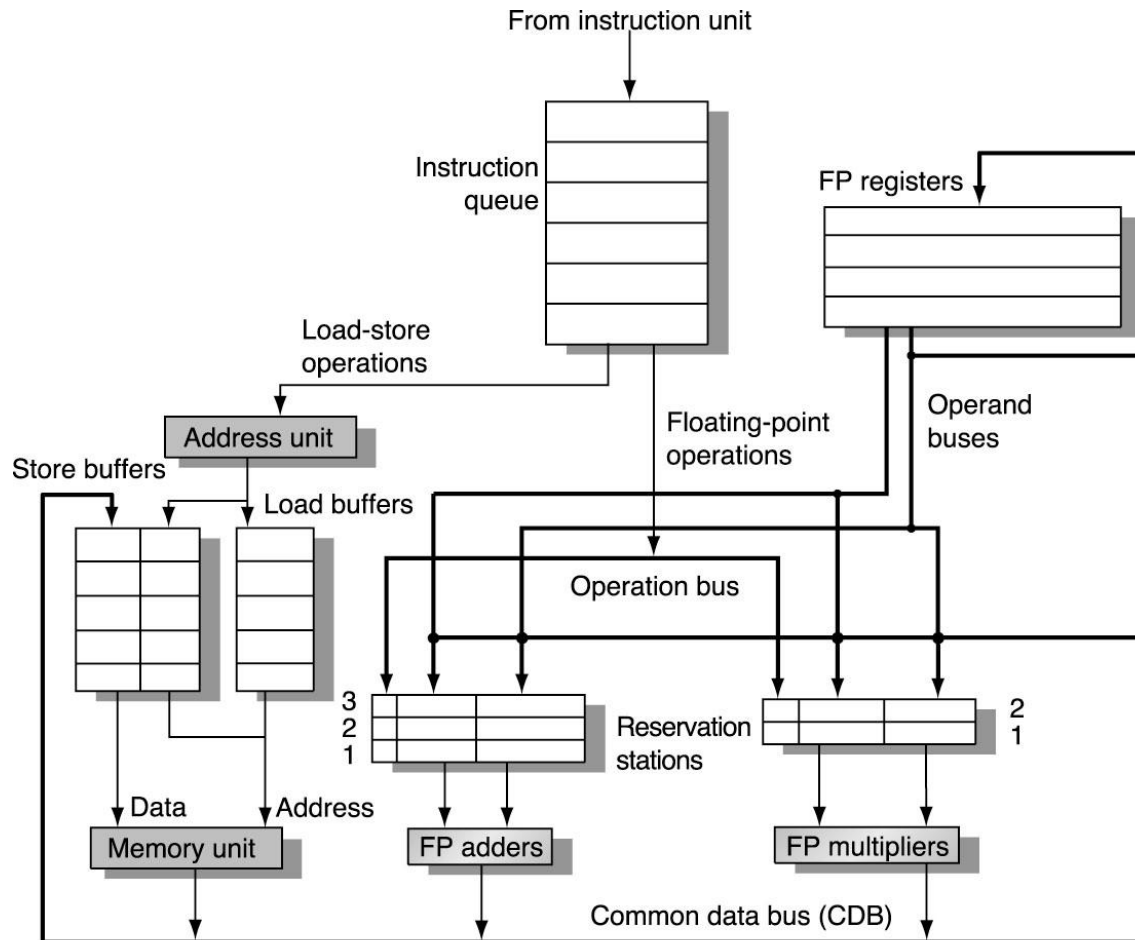
- Algoritmo de despacho de instruções em arquiteturas com múltiplas unidades funcionais.
- Objetivo: efetuar o despacho de instruções de forma a otimizar o uso das unidades funcionais, respeitando as dependências entre as mesmas.



# Algoritmo Dinâmico de Tomasulo

- Para: IBM 360/91
- Objetivo: Alto desempenho sem a necessidade de compiladores especiais
- Por que estudar? Resp.: Alpha 21264, HP 8000, MIPS 10000, Pentium II, Power PC 604 ...
- Observações: Permite a execução fora de ordem, minimiza riscos: RAW (verifica quando os operandos estão disponíveis), WAR e WAW (renomeação de registrador).

# Unidades Funcionais





# Estágios

- 1. Emissão** — a instrução da fila de instruções é emitida com os nomes dos operandos renomeados, se tiver estação de reserva disponível
- 2. Execução** — realiza a execução da operação (EX) se ambos os operandos estão prontos; senão, aguarda o Bus de Dados Comum pelo resultado
- 3. Escrita do resultado** — conclui a instrução (WB)
  - escreve no Bus de Dados Comum para que todas as unidades à espera possam ler o resultado
  - marca a estação de reserva da instrução concluída como disponível

# Componentes

**Op**—operação na unidade (ex., add ou sub)

**Vj, Vk**—**valores** dos operandos fonte

- Os buffers de armazenamento (store) tem um campo do dado a ser armazenado e um campo de endereço
- Os buffers de carga (load) tem apenas um campo de endereço

**Qj, Qk**— unidades funcionais/estação de reserva produzindo os valores dos registradores fonte

- Os buffers de armazenamento tem somente Qi para indicar a unidade funcional/estação de reserva produzindo o dado

**Busy**—indica se a estação de reserva está ocupada

**Status de resultado do registrador** — Indica qual unidade funcional irá escrever em cada registrador.

# Exemplo: Ciclo 0

<u>Instruction status</u>					<i>Execution</i>	<i>Write</i>							
Instruction		<i>j</i>	<i>k</i>	<i>Issue</i>	<i>complete</i>	<i>Result</i>			Busy	Address			
LD	F6	34+	R2					Load1	No				
LD	F2	45+	R3					Load2	No				
MULTD	F0	F2	F4					Load3	No				
SUBD	F8	F6	F2										
DIVD	F10	F0	F6										
ADDD	F6	F8	F2										
<u>Reservation Stations</u>					<i>S1</i>	<i>S2</i>	<i>RS for j</i>	<i>RS for k</i>					
	<i>Time</i>	<i>Name</i>	<i>Busy</i>	<i>Op</i>	<i>Vj</i>	<i>Vk</i>	<i>Qj</i>	<i>Qk</i>					
	0	Add1	No										
	0	Add2	No										
		Add3	No										
	0	Mult1	No										
	0	Mult2	No										
<u>Register result status</u>													
Clock					<i>F0</i>	<i>F2</i>	<i>F4</i>	<i>F6</i>	<i>F8</i>	<i>F10</i>	<i>F12</i>	<i>...</i>	<i>F30</i>
0			<i>FU</i>										

# Ciclo 1

<u>Instruction status</u>					<i>Execution</i>	<i>Write</i>						
Instruction		<i>j</i>	<i>k</i>	<i>Issue</i>	<i>complete</i>	<i>Result</i>			Busy	Address		
LD	F6	34+	R2	1				Load1	Yes	34+R2		
LD	F2	45+	R3					Load2	No			
MULTD	F0	F2	F4					Load3	No			
SUBD	F8	F6	F2									
DIVD	F10	F0	F6									
ADDD	F6	F8	F2									
<u>Reservation Stations</u>					<i>S1</i>	<i>S2</i>	<i>RS for j</i>	<i>RS for k</i>				
	<i>Time</i>	<i>Name</i>	<i>Busy</i>	<i>Op</i>	<i>Vj</i>	<i>Vk</i>	<i>Qj</i>	<i>Qk</i>				
	0	Add1	No									
	0	Add2	No									
		Add3	No									
	0	Mult1	No									
	0	Mult2	No									
<u>Register result status</u>												
Clock				<i>F0</i>	<i>F2</i>	<i>F4</i>	<i>F6</i>	<i>F8</i>	<i>F10</i>	<i>F12</i>	...	<i>F30</i>
1			FU				Load1					

# Ciclo 2

<u>Instruction status</u>					<i>Execution</i>	<i>Write</i>						
Instruction		<i>j</i>	<i>k</i>	<i>Issue</i>	<i>complete</i>	<i>Result</i>			Busy	Address		
LD	F6	34+	R2	1	2-			Load1	Yes	34+R2		
LD	F2	45+	R3	2				Load2	Yes	45+R3		
MULTD	F0	F2	F4					Load3	No			
SUBD	F8	F6	F2			Assume Load takes 2 cycles						
DIVD	F10	F0	F6									
ADDD	F6	F8	F2									
<u>Reservation Stations</u>					<i>S1</i>	<i>S2</i>	<i>RS for j</i>	<i>RS for k</i>				
	<i>Time</i>	<i>Name</i>	<i>Busy</i>	<i>Op</i>	<i>Vj</i>	<i>Vk</i>	<i>Qj</i>	<i>Qk</i>				
	0	Add1	No									
	0	Add2	No									
		Add3	No									
	0	Mult1	No									
	0	Mult2	No									
<u>Register result status</u>												
Clock				<i>F0</i>	<i>F2</i>	<i>F4</i>	<i>F6</i>	<i>F8</i>	<i>F10</i>	<i>F12</i>	<i>...</i>	<i>F30</i>
2			<i>FU</i>		Load2		Load1					

# Ciclo 3

<u>Instruction status</u>					<i>Execution</i>	<i>Write</i>						
Instruction		<i>j</i>	<i>k</i>	<i>Issue</i>	<i>complete</i>	<i>Result</i>			Busy	Address		
LD	F6	34+	R2	1	2--3			Load1	Yes	34+R2		
LD	F2	45+	R3	2	3-			Load2	Yes	45+R3		
MULTD	F0	F2	F4	3				Load3	No			
SUBD	F8	F6	F2									
DIVD	F10	F0	F6									
ADDD	F6	F8	F2									
<u>Reservation Stations</u>					<i>S1</i>	<i>S2</i>	<i>RS for j</i>	<i>RS for k</i>				
	<i>Time</i>	<i>Name</i>	<i>Busy</i>	<i>Op</i>	<i>Vj</i>	<i>Vk</i>	<i>Qj</i>	<i>Qk</i>				
	0	Add1	No									
	0	Add2	No									
		Add3	No									
	0	Mult1	Yes	Mult								
	0	Mult2	No									
<u>Register result status</u>												
Clock				<i>F0</i>	<i>F2</i>	<i>F4</i>	<i>F6</i>	<i>F8</i>	<i>F10</i>	<i>F12</i>	...	<i>F30</i>
3			FU	Mult1	Load2		Load1					

read value

R(F4) Load2

# Ciclo 4

<u>Instruction status</u>						Execution	Write						
Instruction		<i>j</i>	<i>k</i>	Issue	complete	Result			Busy	Address			
LD	F6	34+	R2	1	2--3	4		Load1	No				
LD	F2	45+	R3	2	3--4			Load2	Yes	45+R3			
MULTD	F0	F2	F4	3				Load3	No				
SUBD	F8	F6	F2	4									
DIVD	F10	F0	F6										
ADDD	F6	F8	F2										
<u>Reservation Stations</u>						S1	S2	RS for j	RS for k				
	Time	Name	Busy	Op	Vj	Vk	Qj	Qk					
	0	Add1	Yes	Sub	M(A1)			Load2					
	0	Add2	No										
		Add3	No										
	0	Mult1	Yes	Mult		R(F4)	Load2						
	0	Mult2	No										
<u>Register result status</u>													
Clock				F0	F2	F4	F6	F8	F10	F12	...	F30	
4			FU	Mult1	Load2		M(A1)	Add1					

# Ciclo 5

<u>Instruction status</u>					<i>Execution</i>	<i>Write</i>						
Instruction		<i>j</i>	<i>k</i>	<i>Issue</i>	<i>complete</i>	<i>Result</i>			Busy	Address		
LD	F6	34+	R2	1	2--3	4		Load1	No			
LD	F2	45+	R3	2	3--4	5		Load2	No			
MULTD	F0	F2	F4	3				Load3	No			
SUBD	F8	F6	F2	4								
DIVD	F10	F0	F6	5								
ADDD	F6	F8	F2									
<u>Reservation Stations</u>					<i>S1</i>	<i>S2</i>	<i>RS for j</i>	<i>RS for k</i>				
	<i>Time</i>	<i>Name</i>	<i>Busy</i>	<i>Op</i>	<i>Vj</i>	<i>Vk</i>	<i>Qj</i>	<i>Qk</i>				
	2	Add1	Yes	Sub	M(A1)	M(A2)						
	0	Add2	No									
		Add3	No									
	10	Mult1	Yes	Mult	M(A2)	R(F4)						
	0	Mult2	Yes	Div		M(A1)	Mult1					
<u>Register result status</u>												
Clock				<i>F0</i>	<i>F2</i>	<i>F4</i>	<i>F6</i>	<i>F8</i>	<i>F10</i>	<i>F12</i>	...	<i>F30</i>
5			FU	Mult1	M(A2)		M(A1)	Add1	Mult2			



# Ciclo 6

<u>Instruction status</u>					Execution	Write						
Instruction		<i>j</i>	<i>k</i>	Issue	complete	Result			Busy	Address		
LD	F6	34+	R2	1	2--3	4		Load1	No			
LD	F2	45+	R3	2	3--4	5		Load2	No			
MULTD	F0	F2	F4	3	6 --			Load3	No			
SUBD	F8	F6	F2	4	6 --							
DIVD	F10	F0	F6	5								
ADDD	F6	F8	F2	6								
<u>Reservation Stations</u>					S1	S2	RS for <i>j</i>	RS for <i>k</i>				
	Time	Name	Busy	Op	V <sub>j</sub>	V <sub>k</sub>	Q <sub>j</sub>	Q <sub>k</sub>				
	1	Add1	Yes	Sub	M(A1)	M(A2)						
	0	Add2	Yes	Add		M(A2)	Add1					
		Add3	No									
	9	Mult1	Yes	Mult	M(A2)	R(F4)						
	0	Mult2	Yes	Div		M(A1)	Mult1					
<u>Register result status</u>												
Clock				F0	F2	F4	F6	F8	F10	F12	...	F30
6			FU	Mult1	M(A2)		Add2	Add1	Mult2			

# Ciclo 7

<u>Instruction status</u>					<i>Execution</i>	<i>Write</i>						
Instruction		<i>j</i>	<i>k</i>	<i>Issue</i>	<i>complete</i>	<i>Result</i>			Busy	Address		
LD	F6	34+	R2	1	2--3	4		Load1	No			
LD	F2	45+	R3	2	3--4	5		Load2	No			
MULTD	F0	F2	F4	3	6 --			Load3	No			
SUBD	F8	F6	F2	4	6 -- 7							
DIVD	F10	F0	F6	5								
ADDD	F6	F8	F2	6								
<u>Reservation Stations</u>					<i>S1</i>	<i>S2</i>	<i>RS for j</i>	<i>RS for k</i>				
	<i>Time</i>	<i>Name</i>	<i>Busy</i>	<i>Op</i>	<i>Vj</i>	<i>Vk</i>	<i>Qj</i>	<i>Qk</i>				
	0	Add1	Yes	Sub	M(A1)	M(A2)						
	0	Add2	Yes	Add		M(A2)	Add1					
		Add3	No									
	8	Mult1	Yes	Mult	M(A2)	R(F4)						
	0	Mult2	Yes	Div		M(A1)	Mult1					
<u>Register result status</u>												
Clock				<i>F0</i>	<i>F2</i>	<i>F4</i>	<i>F6</i>	<i>F8</i>	<i>F10</i>	<i>F12</i>	<i>...</i>	<i>F30</i>
7			FU	Mult1	M(A2)		Add2	Add1	Mult2			

# Ciclo 8

<u>Instruction status</u>					<i>Execution</i>	<i>Write</i>						
Instruction		<i>j</i>	<i>k</i>	<i>Issue</i>	<i>complete</i>	<i>Result</i>			Busy	Address		
LD	F6	34+	R2	1	2--3	4		Load1	No			
LD	F2	45+	R3	2	3--4	5		Load2	No			
MULTD	F0	F2	F4	3	6 --			Load3	No			
SUBD	F8	F6	F2	4	6 -- 7	8						
DIVD	F10	F0	F6	5								
ADDD	F6	F8	F2	6								
<u>Reservation Stations</u>					<i>S1</i>	<i>S2</i>	<i>RS for j</i>	<i>RS for k</i>				
	<i>Time</i>	<i>Name</i>	<i>Busy</i>	<i>Op</i>	<i>Vj</i>	<i>Vk</i>	<i>Qj</i>	<i>Qk</i>				
	0	Add1	No									
	2	Add2	Yes	Add	M1-M2	M(A2)						
		Add3	No									
	7	Mult1	Yes	Mult	M(A2)	R(F4)						
	0	Mult2	Yes	Div		M(A1)	Mult1					
<u>Register result status</u>												
Clock				<i>F0</i>	<i>F2</i>	<i>F4</i>	<i>F6</i>	<i>F8</i>	<i>F10</i>	<i>F12</i>	<i>...</i>	<i>F30</i>
8			FU	Mult1	M(A2)		Add2	M1-M2	Mult2			

# Ciclo 9

<u>Instruction status</u>					<i>Execution</i>	<i>Write</i>						
Instruction		<i>j</i>	<i>k</i>	<i>Issue</i>	<i>complete</i>	<i>Result</i>			Busy	Address		
LD	F6	34+	R2	1	2--3	4		Load1	No			
LD	F2	45+	R3	2	3--4	5		Load2	No			
MULTD	F0	F2	F4	3	6 --			Load3	No			
SUBD	F8	F6	F2	4	6 -- 7	8						
DIVD	F10	F0	F6	5								
ADDD	F6	F8	F2	6	9 --							
<u>Reservation Stations</u>					<i>S1</i>	<i>S2</i>	<i>RS for j</i> <i>RS for k</i>					
	<i>Time</i>	<i>Name</i>	<i>Busy</i>	<i>Op</i>	<i>Vj</i>	<i>Vk</i>	<i>Qj</i>	<i>Qk</i>				
	0	Add1	No									
	1	Add2	Yes	Add	M1-M2	M(A2)						
		Add3	No									
	6	Mult1	Yes	Mult	M(A2)	R(F4)						
	0	Mult2	Yes	Div		M(A1)	Mult1					
<u>Register result status</u>												
Clock				<i>F0</i>	<i>F2</i>	<i>F4</i>	<i>F6</i>	<i>F8</i>	<i>F10</i>	<i>F12</i>	...	<i>F30</i>
9			FU	Mult1	M(A2)		Add2	M1-M2	Mult2			

# Ciclo 10

<u>Instruction status</u>					<i>Execution</i>	<i>Write</i>						
Instruction		<i>j</i>	<i>k</i>	<i>Issue</i>	<i>complete</i>	<i>Result</i>			Busy	Address		
LD	F6	34+	R2	1	2--3	4		Load1	No			
LD	F2	45+	R3	2	3--4	5		Load2	No			
MULTD	F0	F2	F4	3	6 --			Load3	No			
SUBD	F8	F6	F2	4	6 -- 7	8						
DIVD	F10	F0	F6	5								
ADDD	F6	F8	F2	6	9 -- 10							
<u>Reservation Stations</u>					<i>S1</i>	<i>S2</i>	<i>RS for j</i> <i>RS for k</i>					
	<i>Time</i>	<i>Name</i>	<i>Busy</i>	<i>Op</i>	<i>Vj</i>	<i>Vk</i>	<i>Qj</i>	<i>Qk</i>				
	0	Add1	No									
	0	Add2	Yes	Add	M1-M2	M(A2)						
		Add3	No									
	5	Mult1	Yes	Mult	M(A2)	R(F4)						
	0	Mult2	Yes	Div		M(A1)	Mult1					
<u>Register result status</u>												
Clock				<i>F0</i>	<i>F2</i>	<i>F4</i>	<i>F6</i>	<i>F8</i>	<i>F10</i>	<i>F12</i>	...	<i>F30</i>
10			FU	Mult1	M(A2)		Add2	M1-M2	Mult2			

# Ciclo 11

<u>Instruction status</u>					<i>Execution</i>	<i>Write</i>						
Instruction		<i>j</i>	<i>k</i>	<i>Issue</i>	<i>complete</i>	<i>Result</i>			Busy	Address		
LD	F6	34+	R2	1	2--3	4		Load1	No			
LD	F2	45+	R3	2	3--4	5		Load2	No			
MULTD	F0	F2	F4	3	6 --			Load3	No			
SUBD	F8	F6	F2	4	6 -- 7	8						
DIVD	F10	F0	F6	5								
ADDD	F6	F8	F2	6	9 -- 10	11						
<u>Reservation Stations</u>					<i>S1</i>	<i>S2</i>	<i>RS for j</i>	<i>RS for k</i>				
	<i>Time</i>	<i>Name</i>	<i>Busy</i>	<i>Op</i>	<i>Vj</i>	<i>Vk</i>	<i>Qj</i>	<i>Qk</i>				
	0	Add1	No									
		Add2	No									
		Add3	No									
	4	Mult1	Yes	Mult	M(A2)	R(F4)						
	0	Mult2	Yes	Div		M(A1)	Mult1					
<u>Register result status</u>												
Clock				<i>F0</i>	<i>F2</i>	<i>F4</i>	<i>F6</i>	<i>F8</i>	<i>F10</i>	<i>F12</i>	<i>...</i>	<i>F30</i>
11			FU	Mult1	M(A2)	M1-M2+M(	M1-M2+M(	M1-M2	Mult2			

# Ciclo 12

<u>Instruction status</u>					<i>Execution</i>	<i>Write</i>							
Instruction		<i>j</i>	<i>k</i>	<i>Issue</i>	<i>complete</i>	<i>Result</i>			Busy	Address			
LD	F6	34+	R2	1	2--3	4		Load1	No				
LD	F2	45+	R3	2	3--4	5		Load2	No				
MULTD	F0	F2	F4	3	6 --			Load3	No				
SUBD	F8	F6	F2	4	6 -- 7	8							
DIVD	F10	F0	F6	5									
ADDD	F6	F8	F2	6	9 -- 10	11							
<u>Reservation Stations</u>					<i>S1</i>	<i>S2</i>	<i>RS for j</i>	<i>RS for k</i>					
	<i>Time</i>	<i>Name</i>	<i>Busy</i>	<i>Op</i>	<i>Vj</i>	<i>Vk</i>	<i>Qj</i>	<i>Qk</i>					
	0	Add1	No										
		Add2	No										
		Add3	No										
	4	Mult1	Yes	Mult	M(A2)	R(F4)							
	0	Mult2	Yes	Div		M(A1)	Mult1						
<u>Register result status</u>													
Clock					<i>F0</i>	<i>F2</i>	<i>F4</i>	<i>F6</i>	<i>F8</i>	<i>F10</i>	<i>F12</i>	<i>...</i>	<i>F30</i>
12				FU	Mult1	M(A2)	M1-M2+M(j)		M1-M2	Mult2			

# Ciclo 15

<u>Instruction status</u>					<i>Execution</i>	<i>Write</i>						
Instruction		<i>j</i>	<i>k</i>	<i>Issue</i>	<i>complete</i>	<i>Result</i>			Busy	Address		
LD	F6	34+	R2	1	2--3	4		Load1	No			
LD	F2	45+	R3	2	3--4	5		Load2	No			
MULTD	F0	F2	F4	3	6 -- 15			Load3	No			
SUBD	F8	F6	F2	4	6 -- 7	8						
DIVD	F10	F0	F6	5								
ADDD	F6	F8	F2	6	9 -- 10	11						
<u>Reservation Stations</u>					<i>S1</i>	<i>S2</i>	<i>RS for j</i>	<i>RS for k</i>				
	<i>Time</i>	<i>Name</i>	<i>Busy</i>	<i>Op</i>	<i>Vj</i>	<i>Vk</i>	<i>Qj</i>	<i>Qk</i>				
	0	Add1	No									
		Add2	No									
		Add3	No									
	0	Mult1	Yes	Mult	M(A2)	R(F4)						
	0	Mult2	Yes	Div		M(A1)	Mult1					
<u>Register result status</u>												
Clock				<i>F0</i>	<i>F2</i>	<i>F4</i>	<i>F6</i>	<i>F8</i>	<i>F10</i>	<i>F12</i>	...	<i>F30</i>
15			FU	Mult1	M(A2)	M1-M2+M(		M1-M2	Mult2			



# Ciclo 16

<u>Instruction status</u>					<i>Execution</i>	<i>Write</i>						
Instruction		<i>j</i>	<i>k</i>	<i>Issue</i>	<i>complete</i>	<i>Result</i>			Busy	Address		
LD	F6	34+	R2	1	2--3	4		Load1	No			
LD	F2	45+	R3	2	3--4	5		Load2	No			
MULTD	F0	F2	F4	3	6 -- 15	16		Load3	No			
SUBD	F8	F6	F2	4	6 -- 7	8						
DIVD	F10	F0	F6	5								
ADDD	F6	F8	F2	6	9 -- 10	11						
<u>Reservation Stations</u>					<i>S1</i>	<i>S2</i>	<i>RS for j</i>	<i>RS for k</i>				
	<i>Time</i>	<i>Name</i>	<i>Busy</i>	<i>Op</i>	<i>Vj</i>	<i>Vk</i>	<i>Qj</i>	<i>Qk</i>				
	0	Add1	No									
		Add2	No									
		Add3	No									
		Mult1	No									
	40	Mult2	Yes	Div	M*F4	M(A1)						
<u>Register result status</u>												
Clock				<i>F0</i>	<i>F2</i>	<i>F4</i>	<i>F6</i>	<i>F8</i>	<i>F10</i>	<i>F12</i>	...	<i>F30</i>
16			FU	M*F4	M(A2)	M1-M2+M(		M1-M2	Mult2			

# Ciclo 56

<u>Instruction status</u>					<i>Execution</i>	<i>Write</i>							
Instruction		<i>j</i>	<i>k</i>	<i>Issue</i>	<i>complete</i>	<i>Result</i>			Busy	Address			
LD	F6	34+	R2	1	2--3	4		Load1	No				
LD	F2	45+	R3	2	3--4	5		Load2	No				
MULTD	F0	F2	F4	3	6 -- 15	16		Load3	No				
SUBD	F8	F6	F2	4	6 -- 7	8							
DIVD	F10	F0	F6	5	17 -- 56								
ADDD	F6	F8	F2	6	9 -- 10	11							
<u>Reservation Stations</u>					<i>S1</i>	<i>S2</i>	<i>RS for j</i>	<i>RS for k</i>					
	<i>Time</i>	<i>Name</i>	<i>Busy</i>	<i>Op</i>	<i>Vj</i>	<i>Vk</i>	<i>Qj</i>	<i>Qk</i>					
	0	Add1	No										
		Add2	No										
		Add3	No										
		Mult1	No										
	0	Mult2	Yes	Div	M*F4	M(A1)							
<u>Register result status</u>													
Clock					<i>F0</i>	<i>F2</i>	<i>F4</i>	<i>F6</i>	<i>F8</i>	<i>F10</i>	<i>F12</i>	<i>...</i>	<i>F30</i>
56			<i>FU</i>	M*F4	M(A2)	M1-M2+M(		M1-M2	Mult2				

# Ciclo 57

<u>Instruction status</u>					<i>Execution</i>	<i>Write</i>						
Instruction		<i>j</i>	<i>k</i>	<i>Issue</i>	<i>complete</i>	<i>Result</i>			Busy	Address		
LD	F6	34+	R2	1	2--3	4		Load1	No			
LD	F2	45+	R3	2	3--4	5		Load2	No			
MULTD	F0	F2	F4	3	6 -- 15	16		Load3	No			
SUBD	F8	F6	F2	4	6 -- 7	8						
DIVD	F10	F0	F6	5	17 -- 56	57						
ADDD	F6	F8	F2	6	9 -- 10	11						
<u>Reservation Stations</u>					<i>S1</i>	<i>S2</i>	<i>RS for j</i>	<i>RS for k</i>				
	<i>Time</i>	<i>Name</i>	<i>Busy</i>	<i>Op</i>	<i>Vj</i>	<i>Vk</i>	<i>Qj</i>	<i>Qk</i>				
	0	Add1	No									
		Add2	No									
		Add3	No									
		Mult1	No									
	0	Mult2	No									
<u>Register result status</u>												
Clock				<i>F0</i>	<i>F2</i>	<i>F4</i>	<i>F6</i>	<i>F8</i>	<i>F10</i>	<i>F12</i>	...	<i>F30</i>
57			FU	M*F4	M(A2)	M1-M2+M(		M1-M2	result			



# Pesquisa\*

- Mostre o funcionamento do algoritmo de Tomasulo através de um exemplo baseado em *loop*.
- Como tratar exceções neste contexto?



# Arquiteturas VLIW (*Very Long Instruction Word*)

- Arquitetura na qual as operações podem ser paralelizadas.
- Detecta as operações paralelizáveis por *software* (compilador) seguida da disposição dessas operações em formato de instruções longas.
- Assim, todas essas operações podem ser executadas num mesmo ciclo.



# Observações

- Uso de **várias unidades funcionais**, uma para cada operação.
- **Exemplo**: 2 operações inteiras, 2 operações ponto flutuante, 2 referências à memória e 1 desvio.
  - Cada campo: 16 a 24 *bits*. Assim, para uma arquitetura VLIW com 7 campos:  $7 \times 16 = 112$  *bits* a  $7 \times 24 = 168$  *bits*.
  - Exemplos: Intel/HP-IA-64 (ITANIUM), *Explicitly Parallel Instruction Computer* (EPIC).

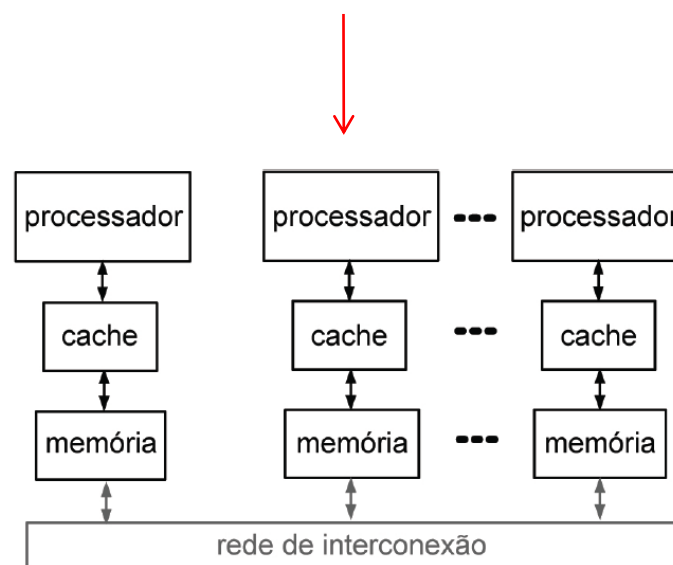
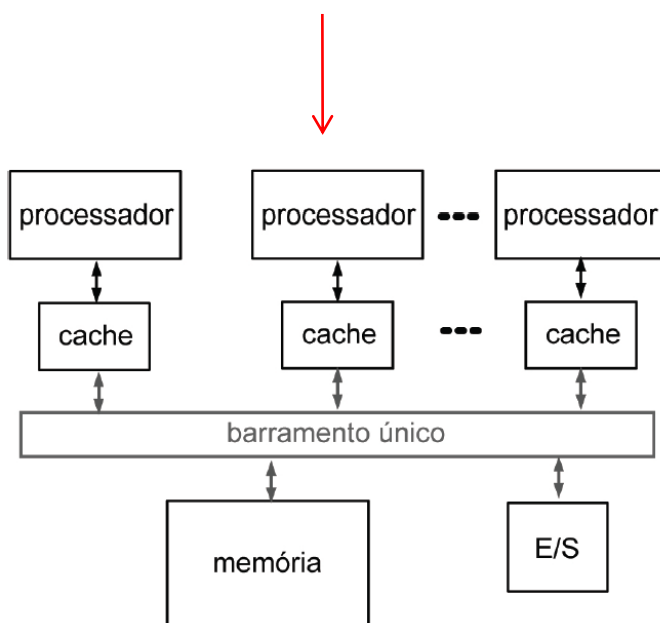


# Multiprocessadores

- Apesar dos avanços no desempenho dos processadores, a capacidade de processamento de um único processador é limitada.
- Possibilidade para melhorar o desempenho para executar uma aplicação: vários processadores em paralelo.
- Esse tipo de arquitetura é denominado: Multiprocessador.

# Multiprocessadores

- Podem ser de **memória compartilhada centralizada** ou de **memória distribuída**:







# Pesquisa\*

- Realize uma pesquisa mais detalhada sobre Multiprocessadores e suas Redes de Interconexão relacionadas. Também, avalie o desempenho de cada modelo.