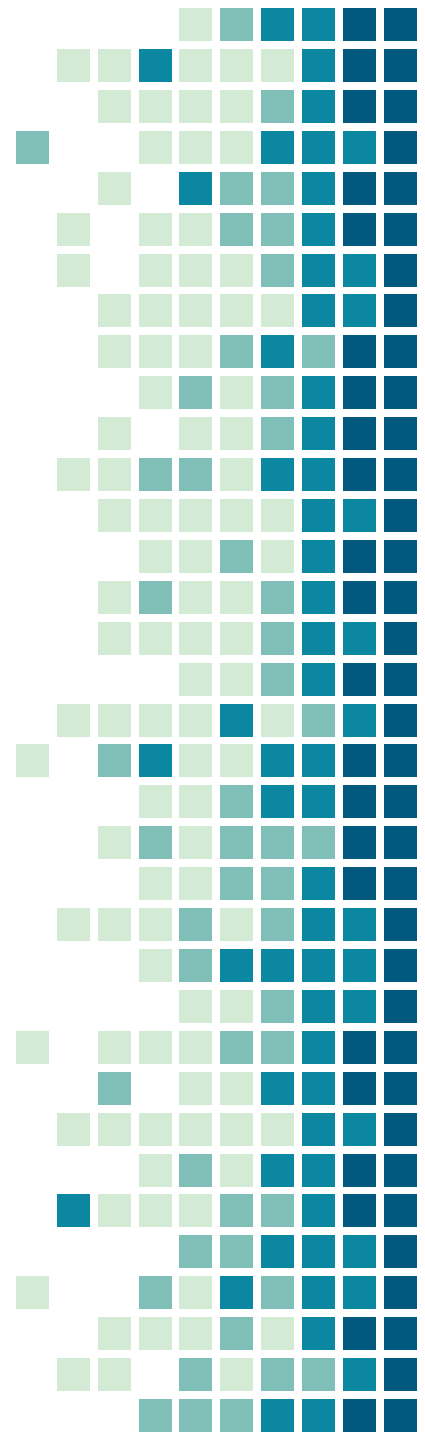


Algoritmo Tomasulo



Agenda

- **Algoritmo Tomasulo.**
- **Arquitectura del algoritmo.**
- **Etapas.**
- **Ejemplos.**
- **Resumen: Tomasulo vs. Marcador**

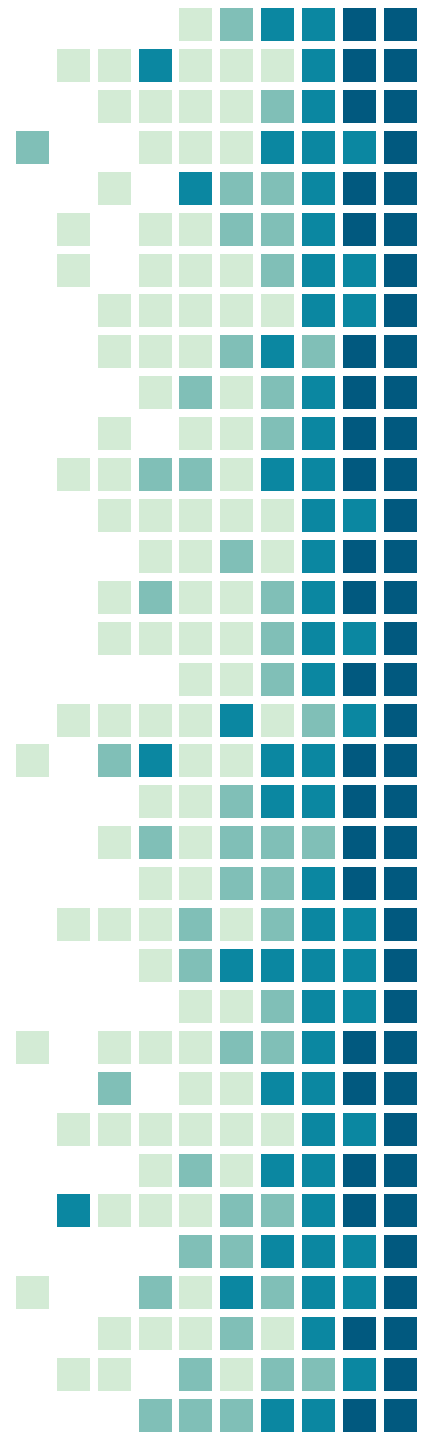


Algoritmo Tomasulo



Algoritmo Tomasulo

- ✓ Algoritmo de calendarización dinámica.
- ✓ Issue en orden, ejecución y terminación fuera de orden.
- ✓ Ejecución continúa aunque hayan dependencias
- ✓ Objetivo: lograr alto rendimiento sin compiladores especiales.
- ✓ Utilizado en: Alpha 21264, MIPS 10000, Pentium II, PowerPC 604.



Tomasulo vs. Marcador

Algoritmo Tomasulo:

Buffers y control distribuido en las FU.

- ❑ Existe renombramiento de registros, para evitar dependencias de nombre.



Algoritmo Marcador:

Control centralizado ("marcador"). No existe renombramiento de registros.

Las estaciones de reserva mantienen los resultados/operandos para el proceso de renombramiento de registros.



Los resultados van a las unidades funcionales (que los necesiten) desde las estaciones de reserva y no a través de los registros, por medio de un bus común (common bus).



Loads y Stores se tratan como unidades funcionales con respectivas estaciones de reserva.



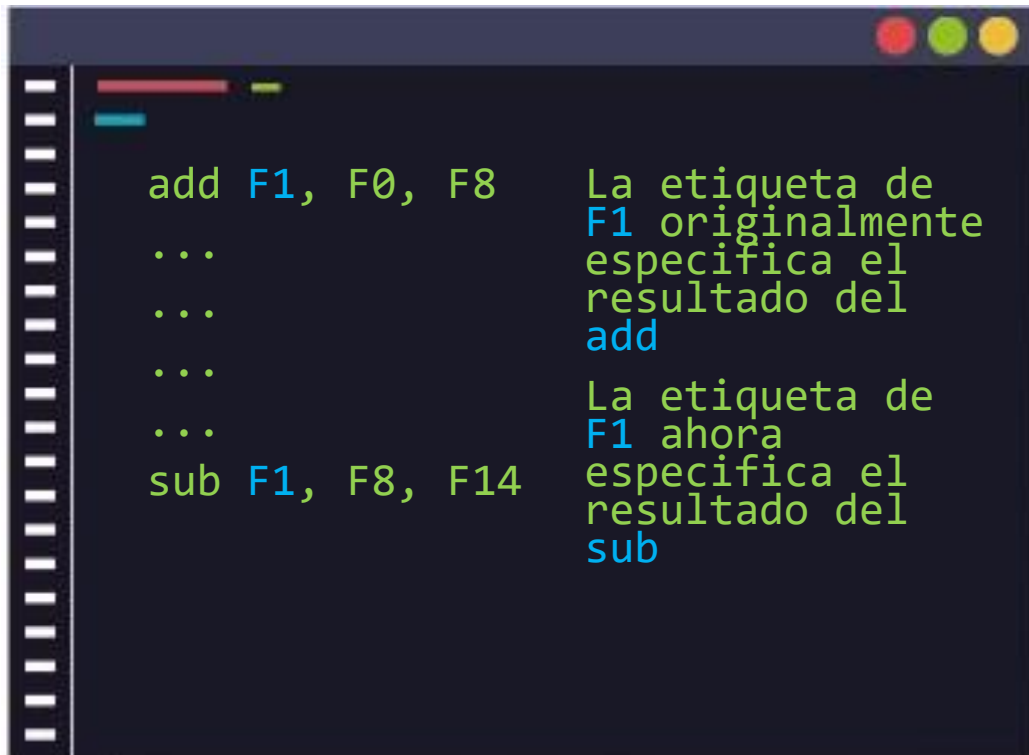
Los operandos son reemplazados por valores o punteros (direcciones efectivas)



Estaciones de reserva: WAW y WAR

Renombramiento de registros elimina los riesgos WAW y WAR.

 **Etiqueta:** en la estación de reserva indica de dónde proviene el resultado.

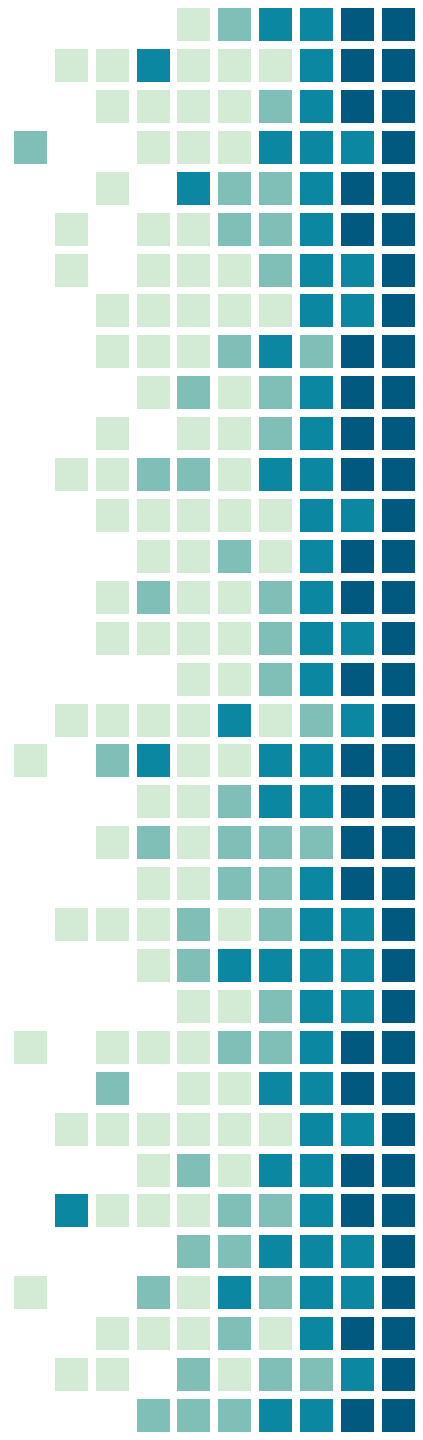


```
add F1, F0, F8
...
...
...
sub F1, F8, F14
```

La etiqueta de F1 originalmente especifica el resultado del add

La etiqueta de F1 ahora especifica el resultado del sub

En caso de que add termine de último, ningún registro tendrá este resultado, ya que la etiqueta representa las dependencias verdaderas y no de nombre (renombramiento).



Estaciones de reserva: WAW y WAR

Riesgos WAR

```
ld F1,-
```

La etiqueta de **F1**
originalmente
especifica el
resultado del **ld**

```
Add-, F1,-
```

La RS del **add**
especifica la
dependencia de la
etiqueta proveniente
del **ld**

```
sub F1,-
```

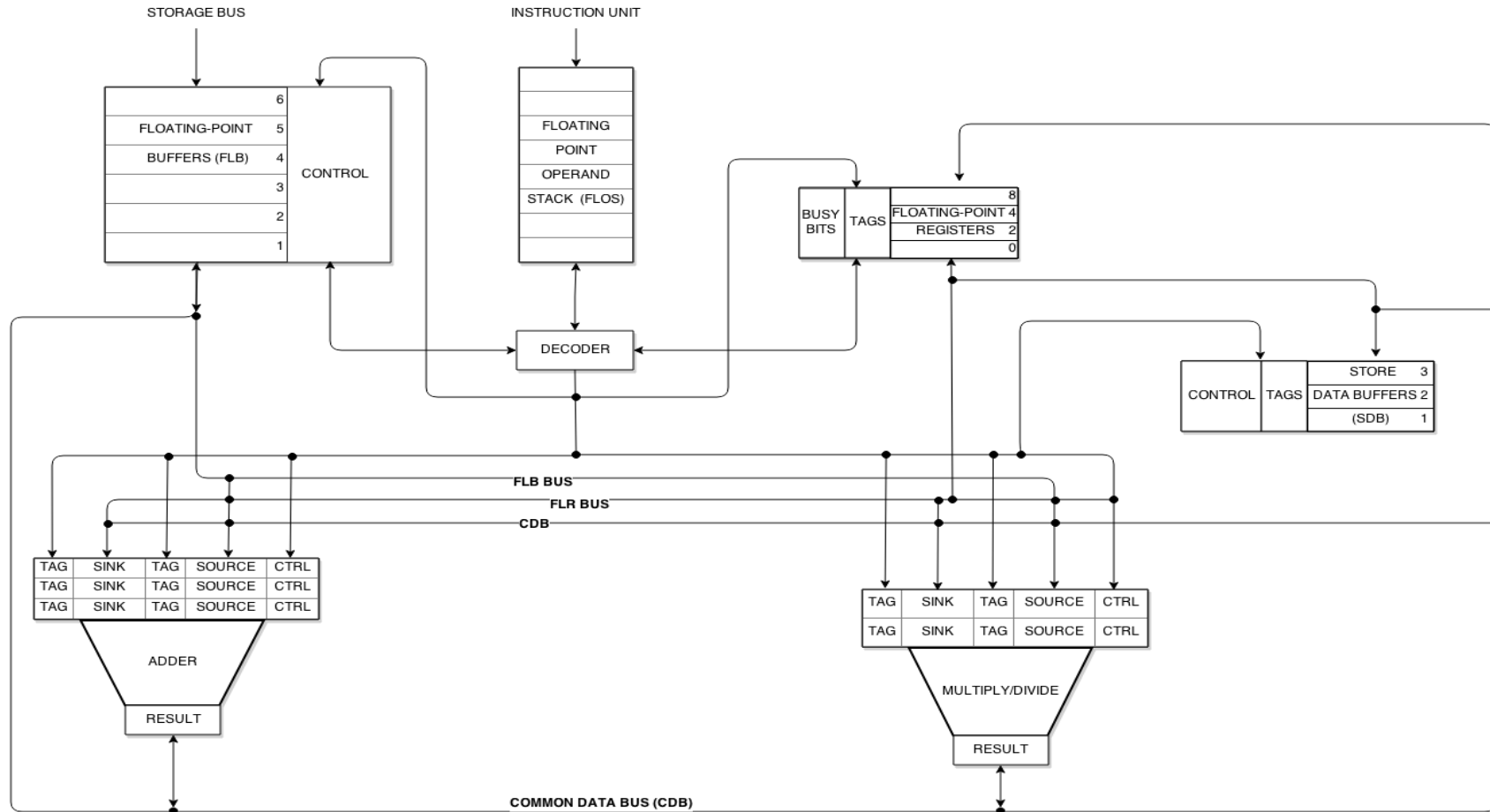
La etiqueta de **F1**
ahora especifica el
resultado del **sub**

No importa si **ld** termina de ejecutarse después del **sub** ya que **F1** ya no depende realmente de esta etiqueta, porque no hay una dependencia real. Add no se ve afectado.

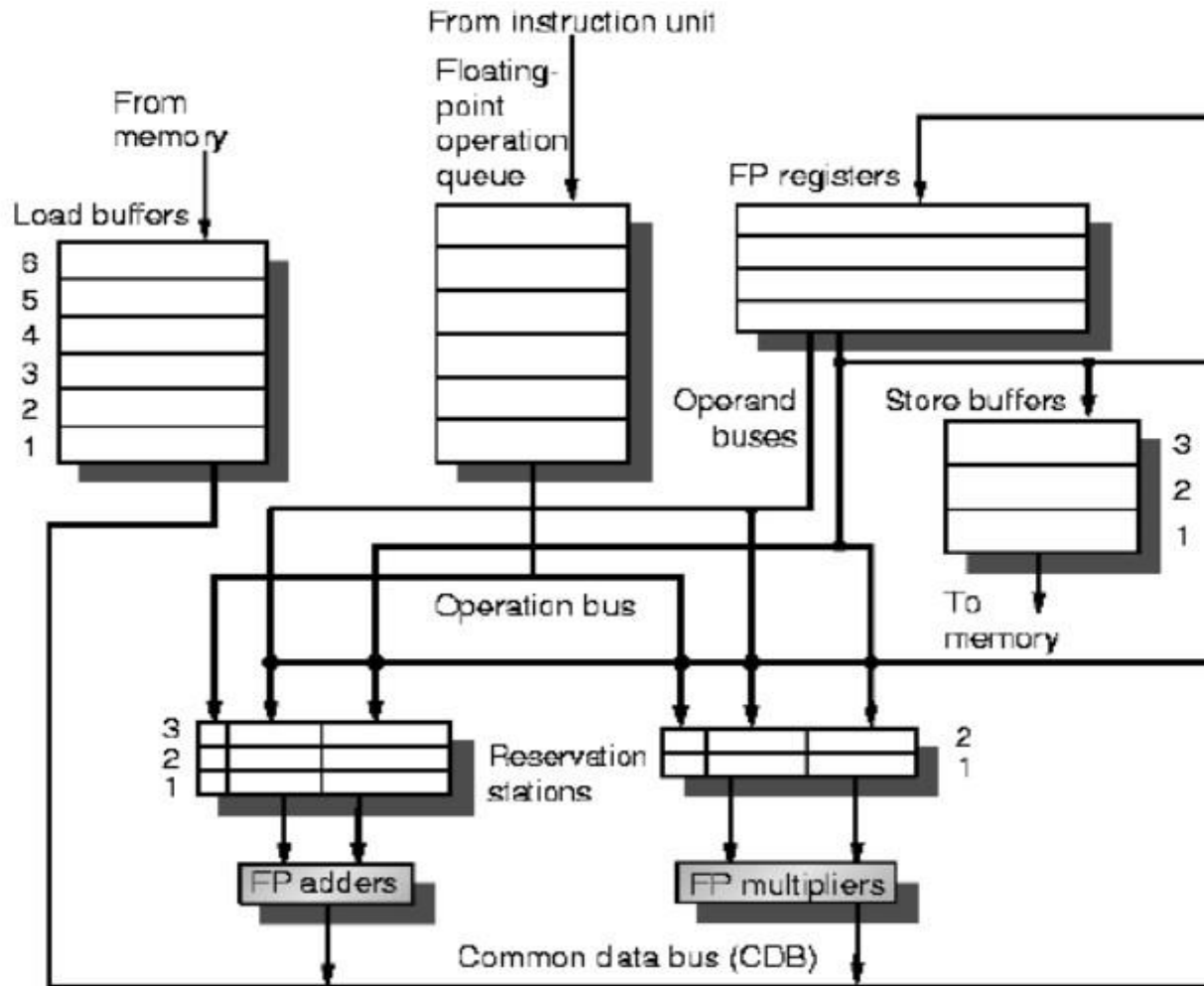
Arquitectura del algoritmo



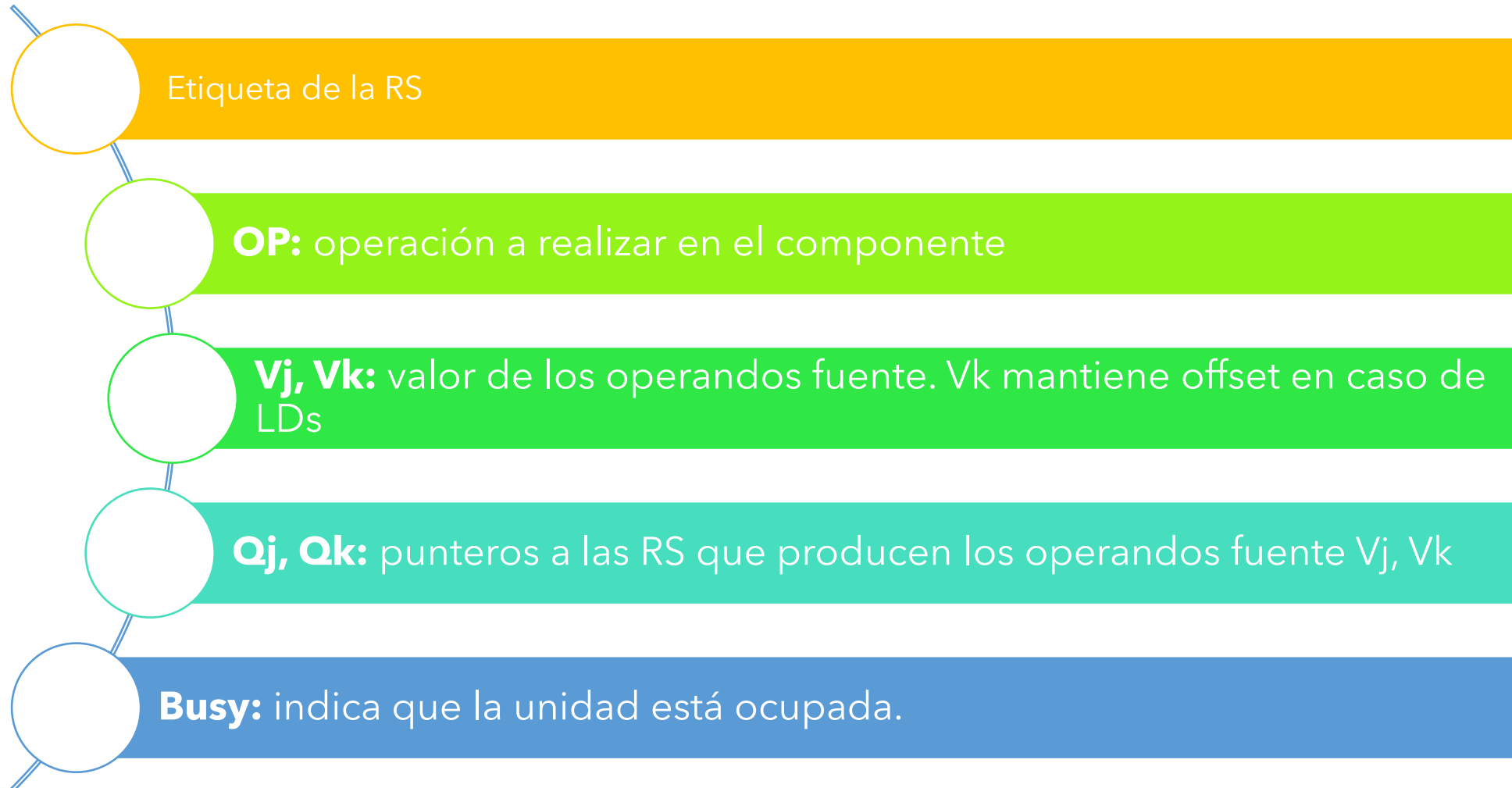
Arquitectura Tomasulo [1]



Arquitectura Tomasulo [2]



Esquema

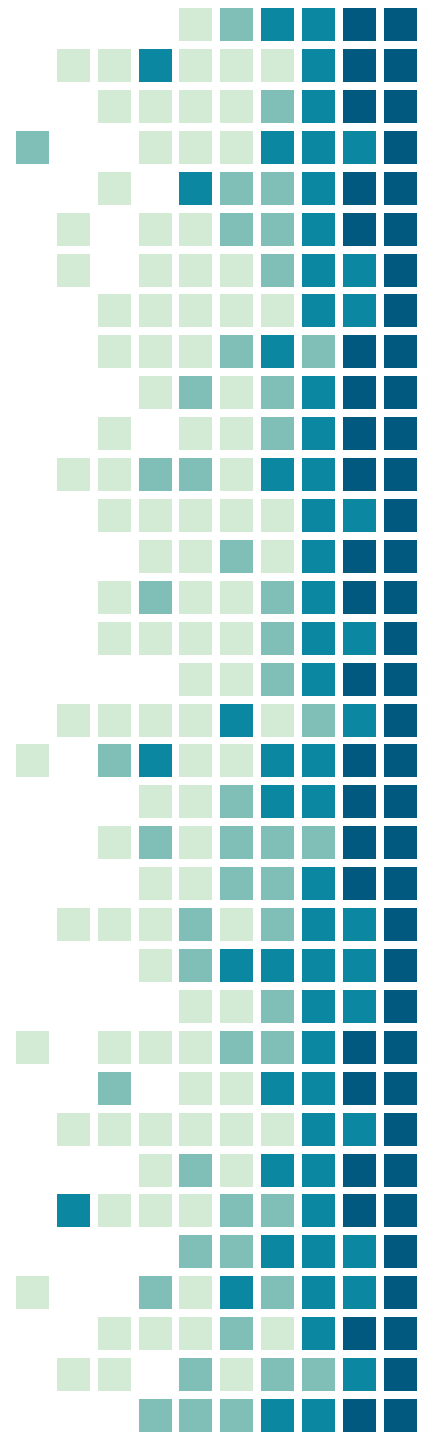


Etapas



1: Issue

- ☑ Si no hay RS vacías para el tipo de operación entonces se detiene la ejecución (Riesgos estructurales).
- ☑ Si los operandos no están disponibles (aún), mantiene información de qué unidad funcional va a producirlos. No se detiene.
- ☑ Renombramiento de registros:
 - WAR: Si **I** escribe Rx, leído por instrucción **K** que ya ha sido lanzada, **K** ya conoce el valor de Rx o conoce la instrucción que lo va a escribir, por lo que registro fuente puede ligarse con **I**
 - WAW: Ya que Issue es en orden, el RF puede ser ligado con **I**



2: Ejecución

Instrucciones LD/SW: Dos pasos

Cuando los operandos estén listos, se ejecuta la instrucción. Sino, se monitorea el bus común hasta obtener los resultados. (Evitar riesgos RAW).

- ➔ Calcular la dirección efectiva cuando la base esté disponible. Colocarla en el buffer store o load.
- ➔ **Loads:** se ejecutan tan pronto la unidad de memoria esté disponible.
- ➔ **Stores:** se espera en el buffer a que el valor sea almacenado antes de escribirlo



3: Escritura de resultado

Cuando el resultado está disponible, se escribe en el bus común (CB), de ahí a todas las Estaciones de Reserva que esperan resultado y luego al archivo de registros.

- Stores escriben datos a memoria durante esta etapa.
- Se marcan estaciones de reserva como disponibles



Ejemplos



Algoritmo Tomasulo

[2], [3]

Instruction status									
Instruction		j	k	Issue	Execution	Write Result			
LD	F6	34+	R2				Load1	Busy	Address
LD	F2	45+	R3				Load2	No	
MULTD	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		
	Add1	No							
	Add2	No							
	Add3	No							
	Mult1	No							
	Mult2	No							
Register result status									
Clock		F0	F2	F4	F6	F8	F10	F12	...
0	FU								

Algoritmo Tomasulo

Ciclo 1

Instruction status										
Instruction		j	k	Issue	Execution	Write Result			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							
Reservation stations										
				S1	S2	RS for j	RS for k			
Time	Name	Busy	Op	Vj	Vk	Qj	Qk			
	Add1	No								
	Add2	No								
	Add3	No								
	Mult1	No								
	Mult2	No								
Register result status										
Clock		F0	F2	F4	F6	F8	F10	F12	...	F30
1	FU				Load1					

Algoritmo Tomasulo

Ciclo 2

Instruction status										
Instruction		j	k	Issue	Execution	Write Result			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							
DIVD	F10	F0	F6							
ADD	F6	F8	F2							
Asumir que el load toma 2 ciclos										
Reservation stations										
				S1	S2	RS for j	RS for k			
Time	Name	Busy	Op	Vj	Vk	Qj	Qk			
	Add1	No								
	Add2	No								
	Add3	No								
	Mult1	No								
	Mult2	No								
Register result status										
Clock		F0	F2	F4	F6	F8	F10	F12	...	F30
2	FU		Load2		Load1					

Algoritmo Tomasulo

Ciclo 3

Instruction status								
Instruction		j	k	Issue	Execution	Write Result		
LD	F6	34+	R2	1	2-3		Load1	Yes
LD	F2	45+	R3	2	3-		Load2	Yes
MULTD	F0	F2	F4	3			Load3	No
SUBD	F8	F6	F2					
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
	Add1	No					
	Add2	No					
	Add3	No					
	Mult1	Yes	Mult		R(F4)	Load2	
	Mult2	No					

Leer el valor

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

Algoritmo Tomasulo

Ciclo 4

Instruction status										
Instruction		j	k	Issue	Execution	Write 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							
ADD	F6	F8	F2							
Reservation stations										
				S1	S2	RS for j	RS for k			
Time	Name	Busy	Op	Vj	Vk	Qj	Qk			
	Add1	Yes	Sub	M(A1)			Load2			
	Add2	No								
	Add3	No								
	Mult1	Yes	Mult		R(F4)	Load2				
	Mult2	No								
Register result status										
Clock		F0	F2	F4	F6	F8	F10	F12	...	F30
4	FU	Mult1	Load2		M(A1)	Add1				

Algoritmo Tomasulo

Ciclo 5

Instruction status										
Instruction		j	k	Issue	Execution	Write 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				Load3	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			
2	Add1	Yes	Sub	M(A1)	M(A2)					
	Add2	No								
	Add3	No								
10	Mult1	Yes	Mult	M(A2)	R(F4)					
	Mult2	Yes	Div		M(A1)	Mult1				
Register result status										
Clock		F0	F2	F4	F6	F8	F10	F12	...	F30
5	FU	Mult1	M(A2)		M(A1)	Add1	Mult2			

Algoritmo Tomasulo

Ciclo 6

Instruction status										
Instruction		j	k	Issue	Execution	Write 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						
Reservation stations										
				S1	S2	RS for j	RS for k			
Time	Name	Busy	Op	Vj	Vk	Qj	Qk			
1	Add1	Yes	Sub	M(A1)	M(A2)					
	Add2	Yes	Add		M(A2)	Add1				
	Add3	No								
9	Mult1	Yes	Mult	M(A2)	R(F4)					
	Mult2	Yes	Div		M(A1)	Mult1				
Register result status										
Clock		F0	F2	F4	F6	F8	F10	F12	...	F30
6	FU	Mult1	M(A2)		Add2	Add1	Mult2			

Algoritmo Tomasulo

Ciclo 7

Instruction status										
Instruction		j	k	Issue	Execution	Write 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	6-7					
DIVD	F10	F0	F6	5						
ADDD	F6	F8	F2	6						
Reservation stations										
				S1	S2	RS for j	RS for k			
Time	Name	Busy	Op	Vj	Vk	Qj	Qk			
0	Add1	Yes	Sub	M(A1)	M(A2)					
	Add2	Yes	Add		M(A2)	Add1				
	Add3	No								
8	Mult1	Yes	Mult	M(A2)	R(F4)					
	Mult2	Yes	Div		M(A1)	Mult1				
Register result status										
Clock		F0	F2	F4	F6	F8	F10	F12	...	F30
7	FU	Mult1	M(A2)		Add2	Add1	Mult2			

Algoritmo Tomasulo

Ciclo 8

Instruction status										
Instruction		j	k	Issue	Execution	Write 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	6-7	8				
DIVD	F10	F0	F6	5						
ADDD	F6	F8	F2	6						
Reservation stations										
				S1	S2	RS for j	RS for k			
Time	Name	Busy	Op	Vj	Vk	Qj	Qk			
0	Add1	No								
2	Add2	Yes	Add	M(A1)-M(A2)	M(A2)					
	Add3	No								
7	Mult1	Yes	Mult	M(A2)	R(F4)					
	Mult2	Yes	Div		M(A1)	Mult1				
Register result status										
Clock		F0	F2	F4	F6	F8	F10	F12	...	F30
8	FU	Mult1	M(A2)		Add2	M(A1)-M(A2)	Mult2			

Algoritmo Tomasulo

Ciclo 9

Instruction status										
Instruction		j	k	Issue	Execution	Write 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	6-7	8				
DIVD	F10	F0	F6	5						
ADDD	F6	F8	F2	6	9-					
Reservation stations										
				S1	S2	RS for j	RS for k			
Time	Name	Busy	Op	Vj	Vk	Qj	Qk			
0	Add1	No								
1	Add2	Yes	Add	M(A1)-M(A2)	M(A2)					
	Add3	No								
6	Mult1	Yes	Mult	M(A2)	R(F4)					
	Mult2	Yes	Div		M(A1)	Mult1				
Register result status										
Clock		F0	F2	F4	F6	F8	F10	F12	...	F30
9	FU	Mult1	M(A2)		Add2	M(A1)-M(A2)	Mult2			

Algoritmo Tomasulo

Ciclo 10

Instruction status										
Instruction		j	k	Issue	Execution	Write 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	6-7	8				
DIVD	F10	F0	F6	5						
ADDD	F6	F8	F2	6	9-10					
Reservation stations										
				S1	S2	RS for j	RS for k			
Time	Name	Busy	Op	Vj	Vk	Qj	Qk			
0	Add1	No								
0	Add2	Yes	Add	M(A1)-M(A2)	M(A2)					
	Add3	No								
5	Mult1	Yes	Mult	M(A2)	R(F4)					
	Mult2	Yes	Div		M(A1)	Mult1				
Register result status										
Clock		F0	F2	F4	F6	F8	F10	F12	...	F30
10	FU	Mult1	M(A2)		Add2	M(A1)-M(A2)	Mult2			

Algoritmo Tomasulo

Ciclo 11

Instruction status										
Instruction		j	k	Issue	Execution	Write 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	6-7	8				
DIVD	F10	F0	F6	5						
ADDD	F6	F8	F2	6	9-10	11				
Reservation stations										
Time	Name	Busy	Op	S1 Vj	S2 Vk	RS for j Qj	RS for k Qk			
	Add1	No								
	Add2	No								
	Add3	No								
4	Mult1	Yes	Mult	M(A2)	R(F4)					
	Mult2	Yes	Div		M(A1)	Mult1				
Register result status										
Clock		F0	F2	F4	F6	F8	F10	F12	...	F30
11	FU	Mult1	M(A2)		M(A1)-M(A2)+M(A2)	M(A1)-M(A2)	Mult2			

Algoritmo Tomasulo

Ciclo 12

Instruction status										
Instruction		j	k	Issue	Execution	Write 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	6-7	8				
DIVD	F10	F0	F6	5						
ADD	F6	F8	F2	6	9-10	11				
Reservation stations										
				S1	S2	RS for j	RS for k			
Time	Name	Busy	Op	Vj	Vk	Qj	Qk			
	Add1	No								
	Add2	No								
	Add3	No								
3	Mult1	Yes	Mult	M(A2)	R(F4)					
	Mult2	Yes	Div		M(A1)	Mult1				
Register result status										
Clock		F0	F2	F4	F6	F8	F10	F12	...	F30
12	FU	Mult1	M(A2)		M(A1)-M(A2)+M(A2)	M(A1)-M(A2)	Mult2			

Algoritmo Tomasulo

Ciclo 15

Instruction status										
Instruction		j	k	Issue	Execution	Write Result				
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				
Reservation stations										
				S1	S2	RS for j	RS for k			
Time	Name	Busy	Op	Vj	Vk	Qj	Qk			
	Add1	No								
	Add2	No								
	Add3	No								
3	Mult1	Yes	Mult	M(A2)	R(F4)					
	Mult2	Yes	Div		M(A1)	Mult1				
Register result status										
Clock		F0	F2	F4	F6	F8	F10	F12	...	F30
12	FU	Mult1	M(A2)		M(A1)-M(A2)+M(A2)	M(A1)-M(A2)	Mult2			

Algoritmo Tomasulo

Ciclo 16

Instruction status										
Instruction		j	k	Issue	Execution	Write 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	6-15	16		Load3	No	
SUBD	F8	F6	F2	4	6-7	8				
DIVD	F10	F0	F6	5						
ADDD	F6	F8	F2	6	9-10	11				
Reservation stations										
				S1	S2	RS for j	RS for k			
Time	Name	Busy	Op	Vj	Vk	Qj	Qk			
	Add1	No								
	Add2	No								
	Add3	No								
	Mult1	No								
40	Mult2	Yes	Div	M*F4	M(A1)					
Register result status										
Clock		F0	F2	F4	F6	F8	F10	F12	...	F30
16	FU	M*F4	M(A2)		M(A1)-M(A2)+M(A2)	M(A1)-M(A2)	Mult2			

Algoritmo Tomasulo

Ciclo 56

Instruction status										
Instruction		j	k	Issue	Execution	Write 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	6-15	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				
Reservation stations										
				S1	S2	RS for j	RS for k			
Time	Name	Busy	Op	Vj	Vk	Qj	Qk			
	Add1	No								
	Add2	No								
	Add3	No								
	Mult1	No								
0	Mult2	Yes	Div	M*F4	M(A1)					
Register result status										
Clock		F0	F2	F4	F6	F8	F10	F12	...	F30
56	FU	M*F4	M(A2)		M(A1)-M(A2)+M(A2)	M(A1)-M(A2)	Mult2			

Algoritmo Tomasulo

Ciclo 57

Instruction status						
Instruction		j	k	Issue	Execution	Write Result
LD	F6	34+	R2	1	2-3	4
LD	F2	45+	R3	2	3-4	5
MULTD	F0	F2	F4	3	6-15	16
SUBD	F8	F6	F2	4	6-7	8
DIVD	F10	F0	F6	5	17-56	57
ADD	F6	F8	F2	6	9-10	11

Load1	Busy	Address
Load1	No	
Load2	No	
Load3	No	

Resumen



Resumen: Tomasulo vs. Marcador

Marcador:

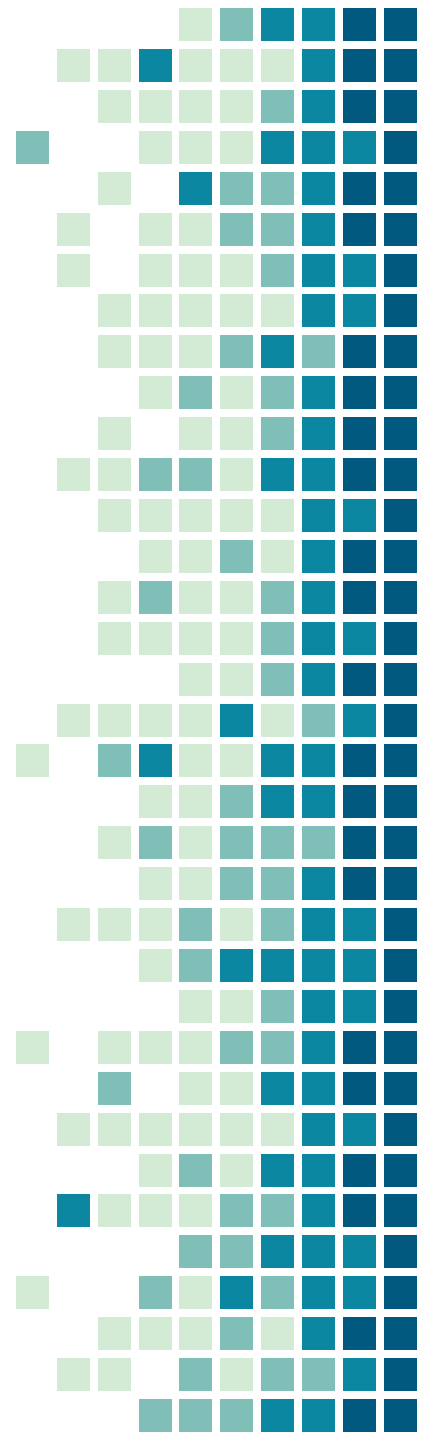
- ➡ Issue en orden. Lectura de operandos fuera de orden. Ejecución y finalización fuera de orden.
- ➡ No hay adelantamiento.
- ➡ Control centralizado en "marcador"

Tomasulo:

- ➡ Issue en orden. Ejecución y finalización fuera de orden
- ➡ Renombramiento de registros. Evita WAR y WAW
- ➡ Único bus (CB). Resultados van a las estaciones de reserva y luego al archivo de registros.
- ➡ Control distribuido.
- ➡ RS sirve como una forma de adelantamiento de datos.

Referencias

- ...► R. Tomasulo, "An Efficient Algorithm for Exploiting Multiple Arithmetic Units," IBM Journal of Research and Development, vol. 11, pp. 25-33, Jan. 1967.
- ...► C. Murillo and M. Aguilar, "Algoritmo Tomasulo," 2013.
- ...► A. González, F. Latorre, and G. Magklis, "Processor microarchitecture: An implementation perspective," Synthesis Lectures on Computer Architecture, vol. 5, no. 1, pp. 1-116, 2010.



¿Preguntas?

Realizado por: Jason Leitón Jiménez.

Tecnológico de Costa Rica

Ingeniería en Computadores

2024

