Planificación dinámica - Concepto

Emisión y ejecución en orden: una limitación.

Stall también detiene a operaciones independientes.

DIVD F0,F2,F4 (1)

ADDD F10,F0,F8 (2)

SUBD F12,F8,F14 (3)

No siempre ejecutar en el orden del programa.

Dividir la etapa ID en dos:

Emisión: decodificación y hazards estructurales.

Leer operandos: si no hay hazard, lee los operandos.

Hazards de datos

Se produce cuando por la segmentación, el orden de LECTURA de los operandos y la ESCRITURA de resultados se modifica respecto a lo especificado en el programa.

Se produce un riesgo si existe dependencia entre instrucciones que se ejecutan concurrentemente.

Si	a = b ⊙ c
	.....
Sj	c = a ⊗ e
	.....
Sk	a = d ⊜ e

Dominio: operandos de la instrucción
Rango: resultado de la instrucción

Situaciones: i precede a j

$D(i) \cap D(j) \neq \phi$ Sin RIESGO
$D(i) \cap R(j) \neq \phi$ riesgo WAR
$R(i) \cap D(j) \neq \phi$ riesgo RAW
$R(i) \cap R(k) \neq \phi$ riesgo WAW

Planificación dinámica - Riesgos WAR y WAW

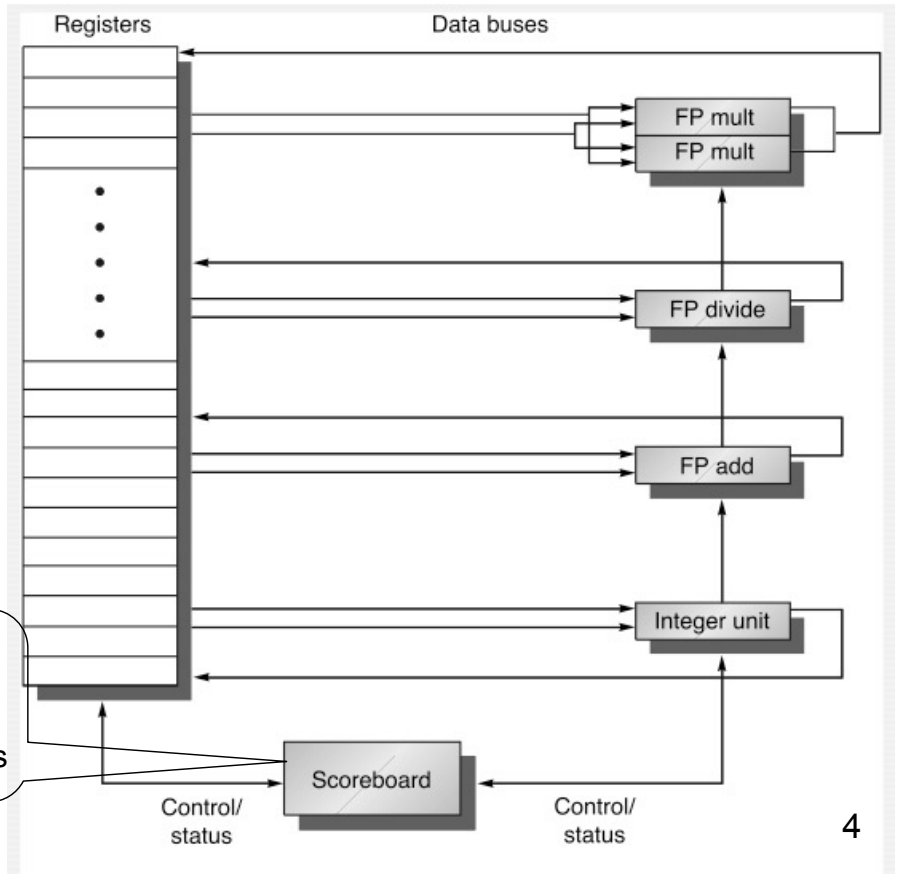
```
DIVD F0,F2,F4
ADDD F10,F0,F8
SUBD F8,F8,F14
```

Antidependencia (WAR)

```
DIVD F0,F2,F4
ADDD F10,F0,F8
SUBD F10,F8,F14
```

Riesgo WAW

MIPS usando Scoreboard



Estado de:
*Instrucciones
*Unidades funcionales
*Registros de resultados

Etapas en Scoreboarding (1)

Etapa de despacho: controla que la unidad funcional esté disponible y que no haya otra instrucción que tenga el mismo registro destino. Con esto se evitan hazards estructurales y hazard WAW. Se despacha la instrucción y se actualizan las estructuras de datos. Si hay WAW o hazard estructural se genera un stall.

Etapas en Scoreboarding (2)

Etapa de emisión: Controla si los operandos están disponibles.

Resuelve RAW dinámicamente.

Si es posible se leen los operandos desde los registros para comenzar la ejecución.

Etapas en Scoreboarding (3)

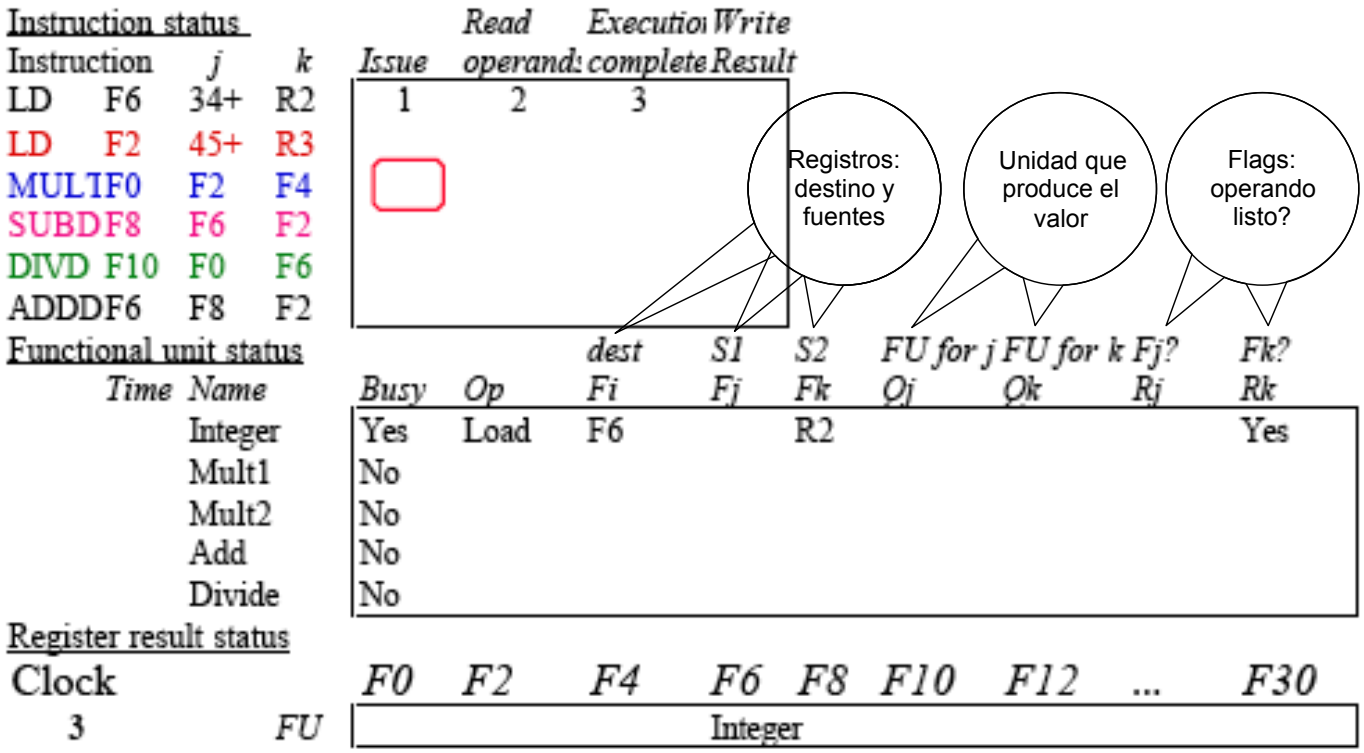
Etapa de ejecución : Hace la ejecución y notifica al scoreboard que se ha completado la ejecución.

Reemplaza la etapa de EX en el pipeline clásico.

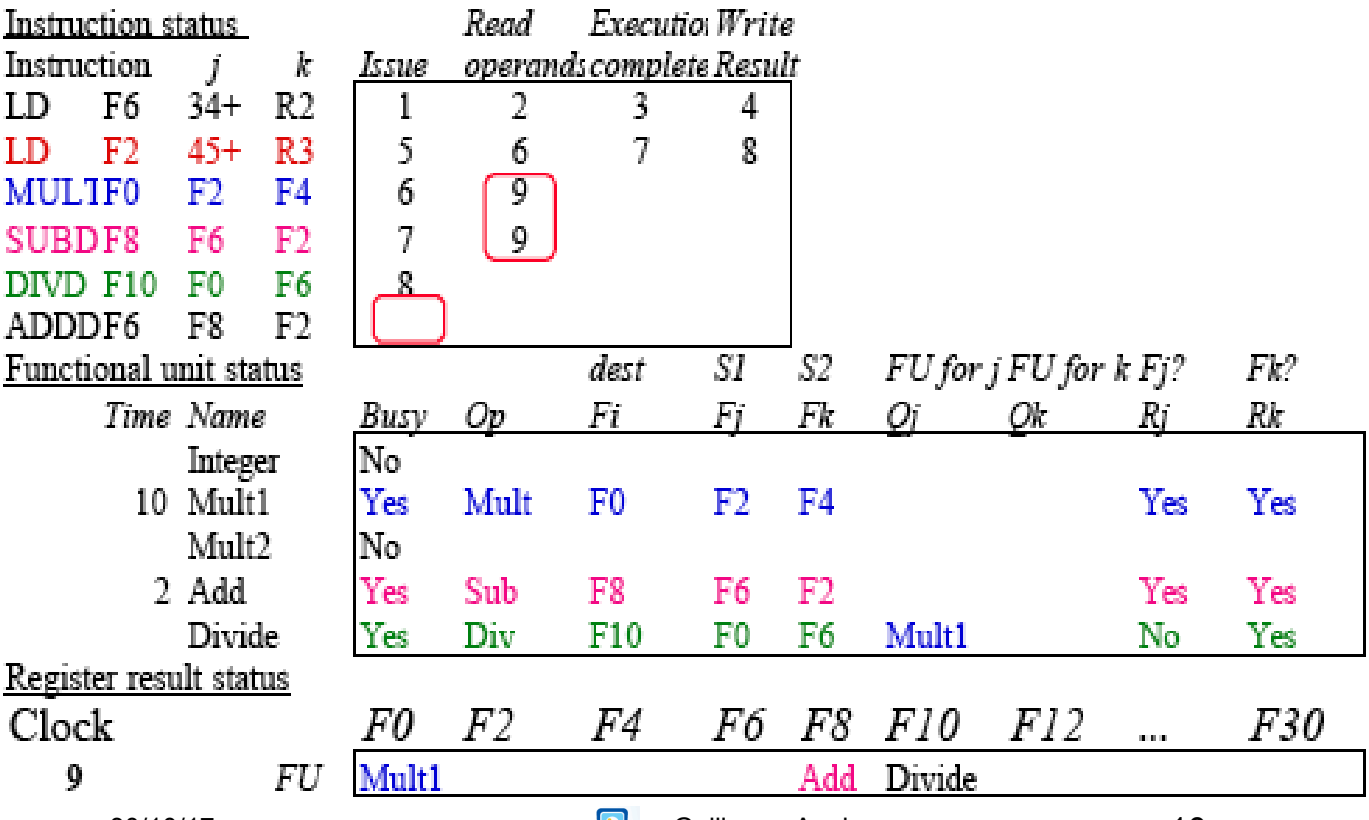
Etapas en Scoreboarding (4)

Etapa de escritura: Se chequea por hazards WAR y se hace un stall de la instrucción a punto de finalizar, si es necesario. Si el resultado se escribe en un operando fuente de alguna instrucción activa anterior, la instrucción no puede finalizar y debe esperar que la otra lea el operando.

Ejemplo ScoreBoard - Ciclo 3



Ejemplo ScoreBoard - Ciclo 9



Ejemplo ScoreBoard - Ciclo 17

Instruction status

Instruction		j	k	Issue	Read operands	Execution complete	Write Result
LD	F6	34+	R2	1	2	3	4
LD	F2	45+	R3	5	6	7	8
MULT	F0	F2	F4	6	9		
SUBD	F8	F6	F2	7	9	11	12
DIVD	F10	F0	F6	8			
ADDD	F6	F8	F2	13	14	16	

La suma toma 2 ciclos

Functional unit status

Time	Name	Busy	Op	dest Fi	S1 Fj	S2 Fk	FU for j Qj	FU for k Qk	Fj? Rj	Fk? Rk
	Integer	No								
2	Mult1	Yes	Mult	F0	F2	F4			Yes	Yes
	Mult2	No								
	Add	Yes	Add	F6	F8	F2			Yes	Yes
	Divide	Yes	Div	F10	F0	F6	Mult1		No	Yes

Register result status

Clock	F0	F2	F4	F6	F8	F10	F12	...	F30
17 FU	Mult1			Add		Divide			

Ejemplo ScoreBoard - Ciclo 62

Instruction status

Instruction		j	k	Issue	Read operands	Execution complete	Write Result
LD	F6	34+	R2	1	2	3	4
LD	F2	45+	R3	5	6	7	8
MULT	F0	F2	F4	6	9	19	20
SUBD	F8	F6	F2	7	9	11	12
DIVD	F10	F0	F6	8	21	61	62
ADDD	F6	F8	F2	13	14	16	22

La multiplicación toma 10 ciclos

La división toma 40 ciclos

Functional unit status

Time	Name	Busy	Op	dest Fi	S1 Fj	S2 Fk	FU for j Qj	FU for k Qk	Fj? Rj	Fk? Rk
	Integer	No								
	Mult1	No								
	Mult2	No								
	Add	No								
0	Divide	No								

Register result status

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

Costo y beneficio

Mejora del desempeño entre 1.7 y 2.5

El scoreboard no requiere muchos más circuitos que las unidades comunes.

Requiere muchos buses, ≅ 4 veces más.

De interés al considerar :

- Despacho de múltiples instrucciones por ciclo
- Especulación

Factores limitantes

Cantidad de paralelismo disponible entre las instrucciones.

Número de entradas en el ScoreBoard.

Número y tipo de unidades funcionales.

Presencia de antidependencias y dependencias de salida

Algoritmo de Scoreboard (1) Emisión o Issue

Esperar que	Tareas
Unidad Funcional[FU] no esté ocupada y Result[D] esté en blanco	Busy[FU]←yes;Op[FU]←op; Fi[FU]←D Fj[FU]←S1; Fk[FU]←S2; Qj←Result[S1]; Qk←Result[S2]; Rj←not Qj; Rk←not Qk; result[D]←FU

Algoritmo de Scoreboard (2) Lectura de registros y ejecución

Esperar que	Tareas
Rj y Rk sean verdaderos	Rj←no; Rk←no; Qj←0; Qk←0
La FU finalice	

Algoritmo de Scoreboard (3)
Escritura de resultados

Esperar que	Tareas
∀f((Fj[f]≠Fi[FU]∨Rj[f]=no)∧(Fk[f]≠Fi[FU]∨Rk[f]=no))	∀f(Si Qj[f]=FU then Rj[f]←yes); ∀f(Si Qk[f]=FU then Rk[f]←yes); Result[Fi[FU]]←0; Busy[FU]←No