

Ejemplo Pág. 53

Ingeniería de Computadores II

Tutor: Antonio Rivero Cuesta

Ejemplo página 53 libro

Mostrar la evolución de los Registros en coma flotante (FR) y las estaciones de Reserva (RS) para todos los ciclos que sean necesarios en la ejecución del siguiente fragmento de código utilizando el algoritmo de Tomasulo.

i1: ADDD F4,F0,F6

i2: MULTD F2,F0,F4

i3: ADDD F4,F4,F6

i4: MULTD F6,F4,F2

- Considere las siguientes hipótesis de partida:
- Para reducir el número de ciclos máquina se permite que la FLOS distribuya hasta dos instrucciones en cada ciclo según el orden del programa.
- Una instrucción puede comenzar su ejecución en el mismo ciclo en que se distribuye a una estación de reserva.

- La operación **suma** tiene una latencia de **dos ciclos** y la de **multiplicación** de **tres ciclos**.
- Se permite que una instrucción reenvíe su resultado a instrucciones dependientes durante su último ciclo de ejecución. De esta forma una instrucción a la espera de un resultado puede comenzar su ejecución en el siguiente ciclo si se detecta una coincidencia.

- Los valores de etiqueta 01, 02 y 03 se utilizan para identificar las tres estaciones de reserva de la unidad funcional de suma, mientras que 04 y 05 se utilizan para identificar las dos estaciones de reserva de la unidad funcional de multiplicación/división. Estos valores de etiqueta son los ID de las estaciones de reserva.
- Inicialmente, el valor de los registros es $F0=6.0$, $F2=3.5$, $F4=10.0$ y $F6=7.8$.

i1: ADDD F4,F0,F6
i2: MULTD F2,F0,F4
i3: ADDD F4,F4,F6
i4: MULTD F6,F4,F2

FR

	bitOc.	etiqueta	dato
F0			6.0
F2			3.5
F4			10.0
F6			7.8

RS

ID	eti_1	oper_1	eti_2	oper_2
01				
02				
03				

SUMA

RS

ID	eti_1	oper_1	eti_2	oper_2
04				
05				

MULT/DIV

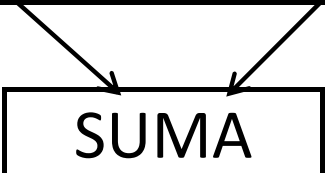
Ciclo 1: Se distribuye i1 e i2
 i1: ADDD F4,F0,F6
 i2: MULTD F2,F0,F4
 Se ejecuta RS 01 1/2
 No se ejecuta RS 04

FR

	bitOc.	etiqueta	dato
F0			6.0
F2	Si	04	3.5
F4	Si	01	10.0
F6			7.8

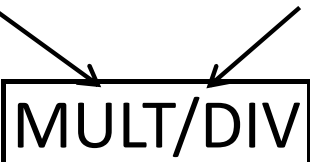
RS

ID	etiq_1	oper_1	etiq_2	oper_2
i1:01	00	6.0	00	7.8
02				
03				



RS

ID	etiq_1	oper_1	etiq_2	oper_2
i2:04	00	6.0	01	xx
05				



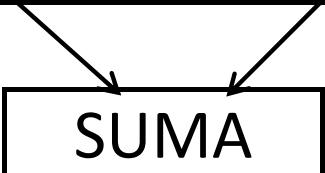
Ciclo 2: Se distribuye i3 e i4
 I3: ADDD F4,F4,F6
 i4: MULTD F6,F4,F2
 Se ejecuta RS 01 2/2
 NO se ejecuta RS 02, 04, 05
 Se envía **RS 01: 13.8** al CDB

FR

	bitOc.	etiqueta	dato
F0			6.0
F2	Si	04	3.5
F4	Si	02	10.0
F6	Si	05	7.8

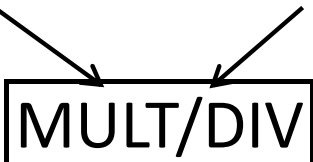
RS

ID	etiq_1	oper_1	etiq_2	oper_2
i1:01	00	6.0	00	7.8
i3:02	01	xx	00	7.8
03				



RS

ID	etiq_1	oper_1	etiq_2	oper_2
i2:04	00	6.0	01	xx
i4:05	02	xx	04	xx



Ciclo 3: Se actualiza el valor de **RS 01: 13.8**

Se vacía RS 01

Se ejecuta RS 04 1/3

Se ejecuta RS 02 1/2

NO se ejecuta RS 05

FR

	bitOc.	etiqueta	dato
F0			6.0
F2	Si	04	3.5
F4	Si	02	10.0
F6	Si	05	7.8

RS

ID	etiq_1	oper_1	etiq_2	oper_2
01				
i3: 02	00	13.8	00	7.8
03				

SUMA

RS

ID	etiq_1	oper_1	etiq_2	oper_2
i2: 04	00	6.0	00	13.8
i4: 05	02	xx	04	xx

MULT/DIV

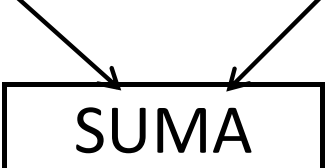
Ciclo 4: Se ejecuta RS 04 2/3
 Se ejecuta RS 02 2/2
 Se envía RS 02: 21.6 al CDB
 NO se ejecuta RS 05

FR

	bitOc.	etiqueta	dato
F0			6.0
F2	Si	04	3.5
F4	Si	02	10.0
F6	Si	05	7.8

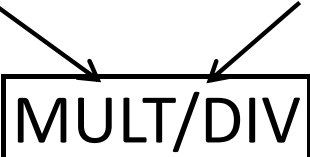
RS

ID	etiq_1	oper_1	etiq_2	oper_2
01				
i3: 02	00	13.8	00	7.8
03				



RS

ID	etiq_1	oper_1	etiq_2	oper_2
i2: 04	00	6.0	00	13.8
i4: 05	02	xx	04	xx



Ciclo 5: Se actualiza el valor de RS 02: 21.6
Se vacía RS 02
Se ejecuta RS 04 3/3
Se envía RS 04: 82.8 al CDB
NO se ejecuta RS 05

FR

	bitOc.	etiqueta	dato
F0			6.0
F2	Si	04	3.5
F4			21.6
F6	Si	05	7.8

RS

ID	eti_1	oper_1	eti_2	oper_2
01				
02				
03				

SUMA

RS

ID	eti_1	oper_1	eti_2	oper_2
i2: 04	00	6.0	00	13.8
i4: 05	00	21.6	04	xx

MULT/DIV

Ciclo 6: Se actualiza el valor de RS 04: 82.8
Se vacía RS 04
Se ejecuta RS 05 1/3

FR

	bitOc.	etiqueta	dato
F0			6.0
F2			82.8
F4			21.6
F6	Si	05	7.8

RS

ID	eti_1	oper_1	eti_2	oper_2
01				
02				
03				

SUMA

RS

ID	eti_1	oper_1	eti_2	oper_2
04				
i4: 05	02	21.6	00	82.8

MULT/DIV

Ciclo 7: Se ejecuta RS 05 2/3

FR

	bitOc.	etiqueta	dato
F0			4.0
F2			82.8
F4			21.6
F6	Si	05	3.5

RS

ID	eti_1	oper_1	eti_2	oper_2
01				
02				
03				

SUMA

RS

ID	eti_1	oper_1	eti_2	oper_2
04				
i4: 05	02	11.0	00	30.0

MULT/DIV

Ciclo 8: Se ejecuta RS 05 3/3
Se envía RS 05: 330.0 al CDB

FR

	bitOc.	etiqueta	dato
F0			6.0
F2			82.8
F4			21.6
F6	Si	05	330

RS

ID	eti_1	oper_1	eti_2	oper_2
01				
02				
03				

SUMA

RS

ID	eti_1	oper_1	eti_2	oper_2
04				
i4: 05	02	11.0	00	30.0

MULT/DIV

Ciclo 9: Se vacía RS 05

FR

	bitOc.	etiqueta	dato
F0			6.0
F2			82.8
F4			21.6
F6			330

RS

ID	eti_1	oper_1	eti_2	oper_2
01				
02				
03				

SUMA

RS

ID	eti_1	oper_1	eti_2	oper_2
04				
i4: 05				

MULT/DIV

Actividad 1.14

Ingeniería de Computadores II

Tutor: Antonio Rivero Cuesta

Actividad 1.14

Mostrar la evolución de los Registros en coma flotante (FR) y las estaciones de Reserva (RS) para todos los ciclos que sean necesarios en la ejecución del siguiente fragmento de código utilizando el algoritmo de Tomasulo.

i1:	ADDD	F2,F0,F6
i2:	MULTD	F4,F0,F2
i3:	ADDD	F2,F2,F6
i4:	MULTD	F6,F2,F4
i5:	ADDD	F4,F4,F6
i6:	ADDD	F6,F2,F4

- Considere las siguientes hipótesis de partida:
- Para reducir el número de ciclos máquina se permite que la FLOS distribuya hasta dos instrucciones en cada ciclo según el orden del programa.
- Una instrucción puede comenzar su ejecución en el mismo ciclo en que se distribuye a una estación de reserva.

- La operación **suma** tiene una latencia de **dos ciclos** y la de **multiplicación** de **tres ciclos**.
- Se permite que una instrucción reenvíe su resultado a instrucciones dependientes durante su último ciclo de ejecución. De esta forma una instrucción a la espera de un resultado puede comenzar su ejecución en el siguiente ciclo si se detecta una coincidencia.

- Los valores de etiqueta 01, 02 y 03 se utilizan para identificar las tres estaciones de reserva de la unidad funcional de suma, mientras que 04 y 05 se utilizan para identificar las dos estaciones de reserva de la unidad funcional de multiplicación/división. Estos valores de etiqueta son los ID de las estaciones de reserva.
- Inicialmente, el valor de los registros es $F0=4.0$, $F2=2.5$, $F4=10.0$ y $F6=3.5$.

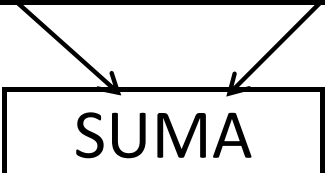
Ciclo 1: Se distribuye i1 e i2
 i1: ADDD F2,F0,F6
 i2: MULTD F4,F0,F2
 Se ejecuta RS 01 1/2
 No se ejecuta RS 04

FR

	bitOc.	etiqueta	dato
F0			4.0
F2	Si	01	2.5
F4	Si	04	10.0
F6			3.5

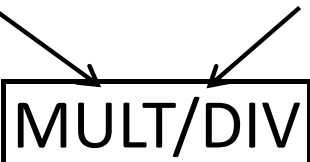
RS

ID	etiq_1	oper_1	etiq_2	oper_2
i1:01	00	4.0	00	3.5
02				
03				



RS

ID	etiq_1	oper_1	etiq_2	oper_2
i2:04	00	4.0	01	xx
05				



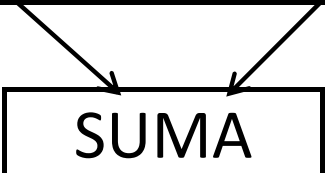
Ciclo 2: Se distribuye i3 e i4
 I3: ADDD F2,F2,F6
 i4: MULTD F6,F2,F4
 Se ejecuta RS 01 2/2
 NO se ejecuta RS 02
 NO se ejecuta RS 05
 Se envía **RS 01: 7.5** al CDB

FR

	bitOc.	etiqueta	dato
F0			4.0
F2	Si	02	2.5
F4	Si	04	10.0
F6	Si	05	3.5

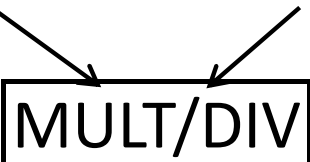
RS

ID	eti_1	oper_1	eti_2	oper_2
i1:01	00	4.0	00	3.5
i3:02	01	xx	00	3.5
03				



RS

ID	eti_1	oper_1	eti_2	oper_2
i2:04	00	4.0	01	xx
i4:05	02	xx	04	xx



Ciclo 3: Se actualiza el valor de **RS 01: 7.5**

Se vacía RS 01

Se ejecuta RS 04 1/3

Se ejecuta RS 02 1/2

Se distribuye i5 e i6

i5: ADDD F4,F4,F6

i6: ADDD F6,F2,F4

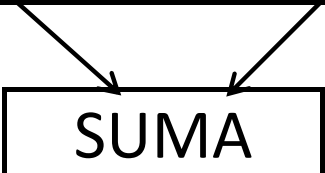
NO se ejecuta RS 05, 03, 01

FR

	bitOc.	etiqueta	dato
F0			4.0
F2	Si	02	2.5
F4	Si	03	10.0
F6	Si	01	3.5

RS

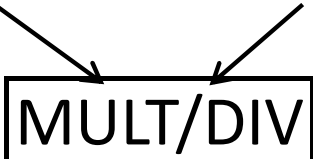
ID	etiq_1	oper_1	etiq_2	oper_2
i6: 01	02	xx	03	xx
i3: 02	00	7.5	00	3.5
i5: 03	04	xx	05	xx



SUMA

RS

ID	etiq_1	oper_1	etiq_2	oper_2
i2: 04	00	4.0	00	7.5
i4: 05	02	xx	04	xx



MULT/DIV

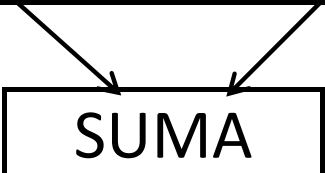
Ciclo 4: Se ejecuta RS 04 2/3
 Se ejecuta RS 02 2/2
 Se envía RS 02: 11.0 al CDB
 NO se ejecuta RS 05, 03, 01

FR

	bitOc.	etiqueta	dato
F0			4.0
F2	Si	02	2.5
F4	Si	03	10.0
F6	Si	01	3.5

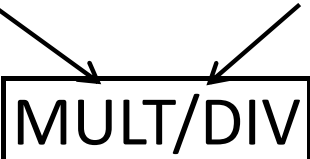
RS

ID	etiq_1	oper_1	etiq_2	oper_2
i6:01	02	xx	03	xx
i3:02	00	7.5	00	3.5
i5:03	04	xx	05	xx



RS

ID	etiq_1	oper_1	etiq_2	oper_2
i2:04	00	4.0	00	7.5
i4:05	02	xx	04	xx



Ciclo 5: Se actualiza el valor de RS 02: 11.0

Se vacía RS 02

Se ejecuta RS 04 3/3

Se envía RS 04: 30.0 al CDB

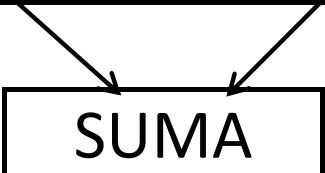
NO se ejecuta RS 05, 03, 01

FR

	bitOc.	etiqueta	dato
F0			4.0
F2			11.0
F4	Si	03	10.0
F6	Si	01	3.5

RS

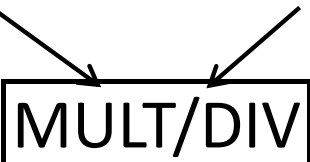
ID	etiq_1	oper_1	etiq_2	oper_2
i6:01	00	11.0	03	xx
02				
i5:03	04	xx	05	xx



SUMA

RS

ID	etiq_1	oper_1	etiq_2	oper_2
i2:04	00	4.0	00	7.5
i4:05	00	11.0	04	xx



MULT/DIV

Ciclo 6: Se actualiza el valor de RS 04: 30.0

Se vacía RS 04

Se ejecuta RS 05 1/3

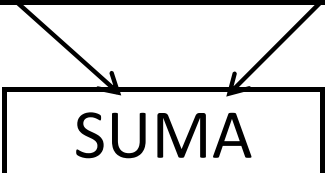
NO se ejecuta RS 03, 01

FR

	bitOc.	etiqueta	dato
F0			4.0
F2			11.0
F4	Si	03	10.0
F6	Si	01	3.5

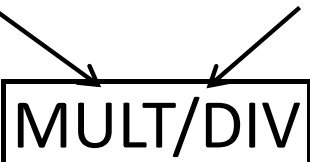
RS

ID	eti_1	oper_1	eti_2	oper_2
i6: 01	02	11.0	03	xx
02				
i5: 03	00	30.0	05	xx



RS

ID	eti_1	oper_1	eti_2	oper_2
04				
i4: 05	02	11.0	00	30.0



Ciclo 7: Se ejecuta RS 05 2/3
NO se ejecuta RS 03, 01

FR

	bitOc.	etiqueta	dato
F0			4.0
F2			11.0
F4	Si	03	10.0
F6	Si	01	3.5

RS

ID	eti_1	oper_1	eti_2	oper_2
i6: 01	02	11.0	03	xx
02				
i5: 03	00	30.0	05	xx

SUMA

RS

ID	eti_1	oper_1	eti_2	oper_2
04				
i4: 05	02	11.0	00	30.0

MULT/DIV

Ciclo 8: Se ejecuta RS 05 3/3
Se envía RS 05: 330.0 al CDB
NO se ejecuta RS 03, 01

FR

	bitOc.	etiqueta	dato
F0			4.0
F2			11.0
F4	Si	03	10.0
F6	Si	01	3.5

RS

ID	etiq_1	oper_1	etiq_2	oper_2
i6: 01	02	11.0	03	xx
02				
i5: 03	00	30.0	05	xx

SUMA

RS

ID	etiq_1	oper_1	etiq_2	oper_2
04				
i4: 05	02	11.0	00	30.0

MULT/DIV

Ciclo 9: Se actualiza el valor de RS 05: 330.0

Se vacía RS 05

Se ejecuta RS 03 1/2

NO se ejecuta RS 01

FR

	bitOc.	etiqueta	dato
F0			4.0
F2			11.0
F4	Si	03	10.0
F6	Si	01	3.5

RS

ID	eti_1	oper_1	eti_2	oper_2
i6:01	02	11.0	03	xx
02				
i5:03	00	30.0	00	330.0

SUMA

RS

ID	eti_1	oper_1	eti_2	oper_2
04				
05				

MULT/DIV

Ciclo 10: Se ejecuta RS 03 2/2
Se envía RS 03: 360.0 al CDB
NO se ejecuta RS 01

FR

	bitOc.	etiqueta	dato
F0			4.0
F2			11.0
F4	Si	03	10.0
F6	Si	01	3.5

RS

ID	etiq_1	oper_1	etiq_2	oper_2
i6: 01	02	11.0	03	xx
02				
i5: 03	00	30.0	00	330.0

SUMA

RS

ID	etiq_1	oper_1	etiq_2	oper_2
04				
05				

MULT/DIV

Ciclo 11: Se actualiza el valor de RS 03: 360.0
Se vacía RS 03
Se ejecuta RS 01 1/2

FR

	bitOc.	etiqueta	dato
F0			4.0
F2			11.0
F4			360.0
F6	Si	01	3.5

RS

ID	etiq_1	oper_1	etiq_2	oper_2
i6:01	02	11.0	03	360.0
02				
03				

SUMA

RS

ID	etiq_1	oper_1	etiq_2	oper_2
04				
05				

MULT/DIV

Ciclo 12: Se ejecuta RS 01 2/2
Se envía RS 01: 371.0 al CDB

FR

	bitOc.	etiqueta	dato
F0			4.0
F2			11.0
F4			360.0
F6	Si	01	3.5

RS

ID	etiq_1	oper_1	etiq_2	oper_2
i6:01	02	11.0	03	360.0
02				
03				

SUMA

RS

ID	etiq_1	oper_1	etiq_2	oper_2
04				
05				

MULT/DIV

Ciclo 13: Se actualiza el valor de RS 0101: 371.0
Se vacía RS 01

FR

	bitOc.	etiqueta	dato
F0			4.0
F2			11.0
F4			360.0
F6			371.0

RS

ID	eti_1	oper_1	eti_2	oper_2
01				
02				
03				

SUMA

RS

ID	eti_1	oper_1	eti_2	oper_2
04				
05				

MULT/DIV

Actividad 1.15

Ingeniería de Computadores II

Tutor: Antonio Rivero Cuesta

Actividad 1.15

Mostrar la evolución de los Registros en coma flotante (FR) y las estaciones de Reserva (RS) para todos los ciclos que sean necesarios en la ejecución del siguiente fragmento de código utilizando el algoritmo de Tomasulo.

```
i1: MULTD  F2,F2,F6  
i2: MULTD  F4,F2,F6  
i3: ADDDD  F2,F2,F6  
i4: ADDDD  F6,F2,F6  
i5: ADDDD  F4,F4,F6
```

- Considere las siguientes hipótesis de partida:
- Para reducir el número de ciclos máquina se permite que la FLOS distribuya hasta dos instrucciones en cada ciclo según el orden del programa.
- Una instrucción puede comenzar su ejecución en el mismo ciclo en que se distribuye a una estación de reserva.

- La operación **suma** tiene una latencia de **dos ciclos** y la de **multiplicación** de **cuatro ciclos**.
- Se permite que una instrucción reenvíe su resultado a instrucciones dependientes durante su último ciclo de ejecución. De esta forma una instrucción a la espera de un resultado puede comenzar su ejecución en el siguiente ciclo si se detecta una coincidencia.

- Los valores de etiqueta 01, 02 y 03 se utilizan para identificar las tres estaciones de reserva de la unidad funcional de suma, mientras que 04 y 05 se utilizan para identificar las dos estaciones de reserva de la unidad funcional de multiplicación/división. Estos valores de etiqueta son los ID de las estaciones de reserva.
- Inicialmente, el valor de los registros es $F0=2.0$, $F2=4.5$, $F4=8.0$ y $F6=3.0$.

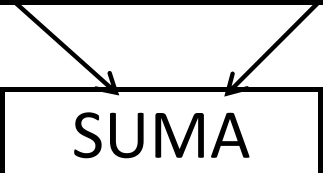
Ciclo 1: Se distribuye i1 e i2
i1: MULTD F2,F2,F6
i2: MULTD F4,F2,F6
Se ejecuta RS 04 1/4
No se ejecuta RS 05

FR

	bitOc.	etiqueta	dato
F0			2.0
F2	Si	04	4.5
F4	Si	05	8.0
F6			3.0

RS

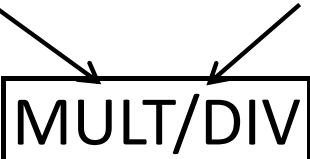
ID	etiq_1	oper_1	etiq_2	oper_2
01				
02				
03				



SUMA

RS

	ID	etiq_1	oper_1	etiq_2	oper_2
i1:	04	00	4.5	00	3.0
i2:	05	04	xx	00	3.0



MULT/DIV

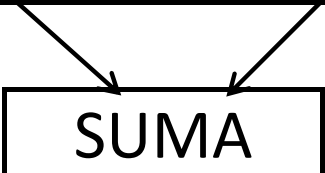
Ciclo 2: Se distribuye i3 e i4
i3: ADDD F2,F4,F6
i4: ADDD F6,F2,F6
Se ejecuta RS 04 2/4
NO se ejecuta RS 01,02,05

FR

	bitOc.	etiqueta	dato
F0			2.0
F2	Si	01	4.5
F4	Si	05	8.0
F6	Si	02	3.0

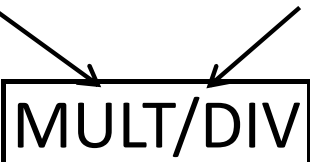
RS

ID	etiq_1	oper_1	etiq_2	oper_2
i3:01	05	xx	00	3.0
i4:02	01	xx	00	3.0
03				



RS

ID	etiq_1	oper_1	etiq_2	oper_2
i1:04	00	4.5	00	3.0
i2:05	04	xx	00	3.0



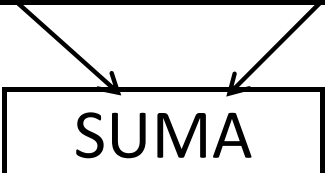
Ciclo 3: Se distribuye i5
i5: ADDD F4,F4,F6
Se ejecuta RS 04 3/4
NO se ejecuta RS 01,02,03,05

FR

	bitOc.	etiqueta	dato
F0			2.0
F2	Si	01	4.5
F4	Si	03	8.0
F6	Si	02	3.0

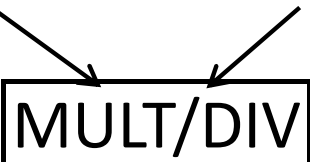
RS

ID	etiq_1	oper_1	etiq_2	oper_2
i3:01	05	xx	00	3.0
i4:02	01	xx	00	3.0
i5:03	05	xx	02	xx



RS

ID	etiq_1	oper_1	etiq_2	oper_2
i1:04	00	4.5	00	3.0
i2:05	04	xx	00	3.0



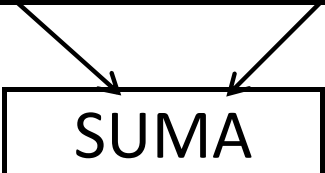
Ciclo 4: Se ejecuta RS 04 4/4
Se envía **RS 04: 13.5** al CDB
NO se ejecuta RS 01,02,03,05

FR

	bitOc.	etiqueta	dato
F0			2.0
F2	Si	01	4.5
F4	Si	03	8.0
F6	Si	02	3.0

RS

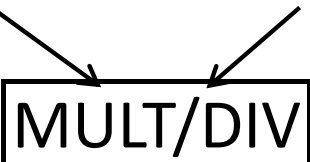
ID	etiq_1	oper_1	etiq_2	oper_2
i3:01	05	xx	00	3.0
i4:02	01	xx	00	3.0
i5:03	05	xx	02	xx



SUMA

RS

ID	etiq_1	oper_1	etiq_2	oper_2
i1:04	00	4.5	00	3.0
i2:05	04	xx	00	3.0



MULT/DIV

Ciclo 5: Se actualiza el valor de **RS 04: 13.5**

Se vacía RS 04

Se ejecuta RS 05 1/4

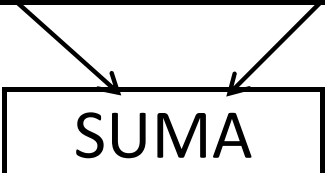
NO se ejecuta RS 01,02,03

FR

	bitOc.	etiqueta	dato
F0			2.0
F2	Si	01	4.5
F4	Si	03	8.0
F6	Si	02	3.0

RS

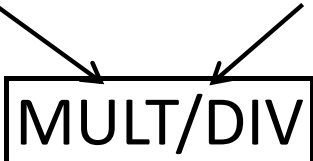
ID	eti_1	oper_1	eti_2	oper_2
i3: 01	05	xx	00	3.0
i4: 02	01	xx	00	3.0
i5: 03	05	xx	02	xx



SUMA

RS

ID	eti_1	oper_1	eti_2	oper_2
04				
i2: 05	00	13.5	00	3.0



MULT/DIV

Ciclo 6: Se ejecuta RS RS 05 2/4
NO se ejecuta RS 01,02,03

FR

	bitOc.	etiqueta	dato
F0			2.0
F2	Si	01	4.5
F4	Si	03	8.0
F6	Si	02	3.0

RS

ID	etiq_1	oper_1	etiq_2	oper_2
i3: 01	05	xx	00	3.0
i4: 02	01	xx	00	3.0
i5: 03	05	xx	02	xx

SUMA

RS

ID	etiq_1	oper_1	etiq_2	oper_2
04				
i2: 05	00	13.5	00	3.0

MULT/DIV

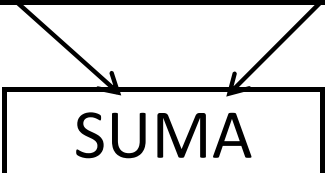
Ciclo 7: Se ejecuta RS RS 05 3/4
NO se ejecuta RS 01,02,03

FR

	bitOc.	etiqueta	dato
F0			2.0
F2	Si	01	4.5
F4	Si	03	8.0
F6	Si	02	3.0

RS

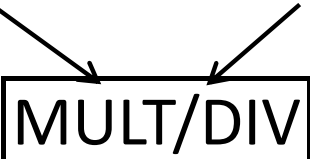
ID	etiq_1	oper_1	etiq_2	oper_2
i3: 01	05	xx	00	3.0
i4: 02	01	xx	00	3.0
i5: 03	05	xx	02	xx



SUMA

RS

ID	etiq_1	oper_1	etiq_2	oper_2
04				
i2: 05	00	13.5	00	3.0



MULT/DIV

Ciclo 8: Se ejecuta RS RS 05 4/4
Se envía RS 05: 40.5 al CDB
NO se ejecuta RS 01,02,03

FR

	bitOc.	etiqueta	dato
F0			2.0
F2	Si	01	4.5
F4	Si	03	8.0
F6	Si	02	3.0

RS

ID	etiq_1	oper_1	etiq_2	oper_2
i3: 01	05	xx	00	3.0
i4: 02	01	xx	00	3.0
i5: 03	05	xx	02	xx

SUMA

RS

ID	etiq_1	oper_1	etiq_2	oper_2
04				
i2: 05	00	13.5	00	3.0

MULT/DIV

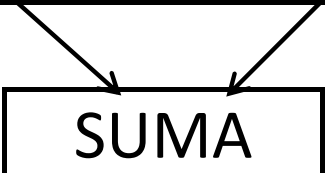
Ciclo 9: Se actualiza el valor de RS 05: 40.5
 Se vacía RS 05
 Se ejecuta RS 01 1/2
 NO se ejecuta RS 02,03

FR

	bitOc.	etiqueta	dato
F0			2.0
F2	Si	01	4.5
F4	Si	03	8.0
F6	Si	02	3.0

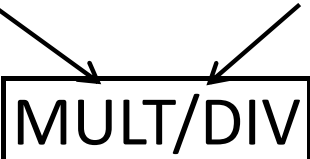
RS

ID	eti_1	oper_1	eti_2	oper_2
i3: 01	00	40.5	00	3.0
i4: 02	01	xx	00	3.0
i5: 03	00	40.5	02	xx



RS

ID	eti_1	oper_1	eti_2	oper_2
04				
05				



Ciclo 10: Se ejecuta RS 01 2/2
Se envía RS 01: 43.5 al CDB
NO se ejecuta RS 02,03

FR

	bitOc.	etiqueta	dato
F0			2.0
F2	Si	01	4.5
F4	Si	03	8.0
F6	Si	02	3.0

RS

ID	etiq_1	oper_1	etiq_2	oper_2
i3: 01	00	40.5	00	3.0
i4: 02	01	xx	00	3.0
i5: 03	00	40.5	02	xx

SUMA

RS

ID	etiq_1	oper_1	etiq_2	oper_2
04				
05				

MULT/DIV

Ciclo 11: Se actualiza el valor de RS 01: 43.5

Se vacía RS 01

Se ejecuta RS 02 1/2

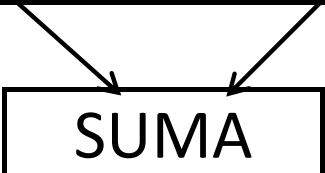
NO se ejecuta RS 03

FR

	bitOc.	etiqueta	dato
F0			2.0
F2			43.5
F4	Si	03	8.0
F6	Si	02	3.0

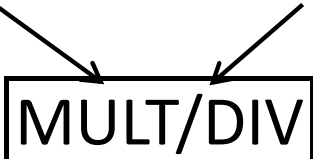
RS

ID	etiq_1	oper_1	etiq_2	oper_2
01				
i4: 02	00	43.5	00	3.0
i5: 03	00	40.5	02	xx



RS

ID	etiq_1	oper_1	etiq_2	oper_2
04				
05				



Ciclo 12: Se ejecuta RS 02 2/2
Se envía RS 02: 46.5 al CDB
NO se ejecuta RS 03

FR

	bitOc.	etiqueta	dato
F0			2.0
F2			43.5
F4	Si	03	8.0
F6	Si	02	3.0

RS

ID	etiq_1	oper_1	etiq_2	oper_2
01				
i4: 02	01	43.5	00	3.0
i5: 03	00	40.5	02	xx

SUMA

RS

ID	etiq_1	oper_1	etiq_2	oper_2
04				
05				

MULT/DIV

Ciclo 13: Se actualiza el valor de RS 02: 46.5
Se vacía RS 02
Se ejecuta RS 03 1/2

FR

	bitOc.	etiqueta	dato
F0			2.0
F2			43.5
F4	Si	03	8.0
F6			46.5

RS

ID	etiq_1	oper_1	etiq_2	oper_2
01				
02				
i5: 03	00	40.5	00	46.5

SUMA

RS

ID	etiq_1	oper_1	etiq_2	oper_2
04				
05				

MULT/DIV

Ciclo 14: Se ejecuta RS 03 2/2
Se envía RS 03: 87 al CDB

FR

	bitOc.	etiqueta	dato
F0			2.0
F2			43.5
F4	Si	03	8.0
F6			46.5

RS

ID	etiq_1	oper_1	etiq_2	oper_2
01				
02				
i5: 03	00	40.5	00	46.5

SUMA

RS

ID	etiq_1	oper_1	etiq_2	oper_2
04				
05				

MULT/DIV

Ciclo 15: Se actualiza el valor de RS 03: 87
Se vacía RS 03

FR

	bitOc.	etiqueta	dato
F0			2.0
F2			43.5
F4			87.0
F6			46.5

RS

ID	eti_1	oper_1	eti_2	oper_2
01				
02				
03				

SUMA

RS

ID	eti_1	oper_1	eti_2	oper_2
04				
05				

MULT/DIV

Problema 1 Sep. 2012

Ingeniería de Computadores II

Tutor: Antonio Rivero Cuesta

Problema 1 - 2012 septiembre

Mostrar la evolución de los Registros en coma flotante (FR) y las estaciones de Reserva (RS) para todos los ciclos que sean necesarios en la ejecución del siguiente fragmento de código utilizando el algoritmo de Tomasulo.

```
i1: ADDD    F6,F4,F2  
i2: MULTD  F0,F4,F6  
i3: MULTD  F6,F6,F2  
i4: ADDD    F2,F6,F0
```

- Considere las siguientes hipótesis de partida:
- Para reducir el número de ciclos máquina se permite que la FLOS distribuya hasta dos instrucciones en cada ciclo según el orden del programa.
- Una instrucción puede comenzar su ejecución en el mismo ciclo en que se distribuye a una estación de reserva.

- La operación **suma** tiene una latencia de **un ciclo** y la de **multiplicación** de **dos ciclos**.
- Se permite que una instrucción reenvíe su resultado a instrucciones dependientes durante su último ciclo de ejecución. De esta forma una instrucción a la espera de un resultado puede comenzar su ejecución en el siguiente ciclo si se detecta una coincidencia.

- Los valores de etiqueta 01, 02 y 03 se utilizan para identificar las tres estaciones de reserva de la unidad funcional de suma, mientras que 04 y 05 se utilizan para identificar las dos estaciones de reserva de la unidad funcional de multiplicación/división. Estos valores de etiqueta son los ID de las estaciones de reserva.
- Inicialmente, el valor de los registros es $F0=8.0$, $F2=3.5$, $F4=2.0$ y $F6=3.0$.

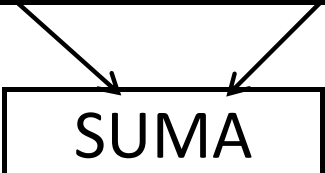
Ciclo 1: Se distribuye i1 e i2
 i1: ADDD F6,F4,F2
 i2: MULTD F0,F4,F6
 Se ejecuta RS 01 1/1
 Se envía **RS 01: 5.5** al CDB
 No se ejecuta RS 04

FR

	bitOc.	etiqueta	dato
F0	Si	04	8.0
F2			3.5
F4			2.0
F6	Si	01	3.0

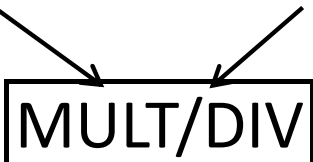
RS

ID	etiq_1	oper_1	etiq_2	oper_2
i1:01	00	2.0	00	3.5
02				
03				



RS

ID	etiq_1	oper_1	etiq_2	oper_2
i2:04	00	2.0	01	xx
05				



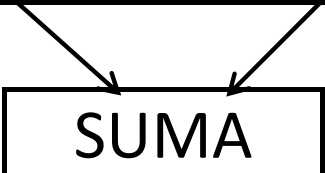
Ciclo 2: Se distribuye i3 e i4
i3: MULTD F6,F6,F2
i4: ADDD F2,F6,F0
Se actualiza el valor de **RS 01: 5.5**
Se vacía RS 01
Se ejecuta RS 04 1/2
Se ejecuta RS 05 1/2
NO se ejecuta RS 02

FR

	bitOc.	etiqueta	dato
F0	Si	04	8.0
F2	Si	02	3.5
F4			2.0
F6	Si	05	3.0

RS

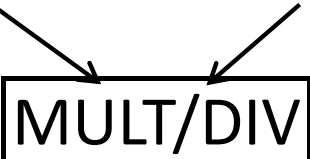
ID	etiq_1	oper_1	etiq_2	oper_2
01				
i4: 02	05	xx	04	xx
03				



SUMA

RS

ID	etiq_1	oper_1	etiq_2	oper_2
i2: 04	00	2.0	00	5.5
i3: 05	00	5.5	00	3.5



MULT/DIV

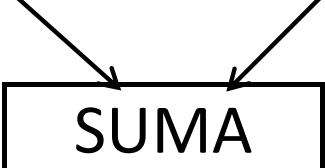
Ciclo 3: Se ejecuta RS 04 2/2
 Se ejecuta RS 05 2/2
 Se envía RS 04: 11.0 al CDB
 Se envía RS 05: 19.25 al CDB
 NO se ejecuta RS 02

FR

	bitOc.	etiqueta	dato
F0	Si	04	8.0
F2	Si	02	3.5
F4			2.0
F6	Si	05	3.0

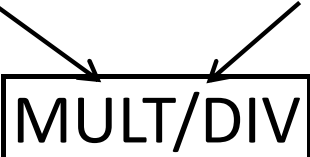
RS

ID	etiq_1	oper_1	etiq_2	oper_2
01				
i4: 02	05	xx	04	xx



RS

ID	etiq_1	oper_1	etiq_2	oper_2
i2: 04	00	2.0	00	5.5
i3: 05	00	5.5	00	3.5



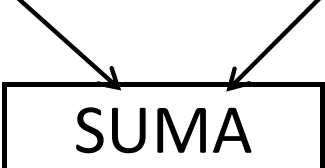
Ciclo 4: Se actualiza el valor de RS 04: 11.0
 Se actualiza el valor de RS 05: 19.25
 Se vacía RS 04
 Se vacía RS 05
 Se ejecuta RS 02 1/1
 Se envía RS 02: 30.25 al CDB

FR

	bitOc.	etiqueta	dato
F0			11.0
F2	Si	02	3.5
F4			2.0
F6			19.25

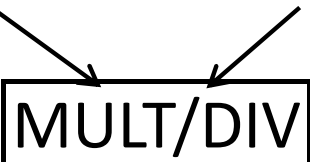
RS

ID	etiq_1	oper_1	etiq_2	oper_2
01				
i4: 02	05	19.25	04	11.0



RS

ID	etiq_1	oper_1	etiq_2	oper_2
04				
05				



Ciclo 5: Se actualiza el valor de RS 02: 30.25
Se vacía RS 02

FR

	bitOc.	etiqueta	dato
F0			11.0
F2			30.25
F4			2.0
F6			19.25

RS

ID	etiq_1	oper_1	etiq_2	oper_2
01				
02				

SUMA

RS

ID	etiq_1	oper_1	etiq_2	oper_2
04				
05				

MULT/DIV

Problema 1 Feb. 2013

Ingeniería de Computadores II

Tutor: Antonio Rivero Cuesta

Problema 1 - 2013 febrero

Mostrar la evolución de los Registros en coma flotante (FR) y las estaciones de Reserva (RS) para todos los ciclos que sean necesarios en la ejecución del siguiente fragmento de código utilizando el algoritmo de Tomasulo.

```
i1: MULTD F0,F6,F2  
i2: ADDD   F4,F2,F6  
i3: ADDD   F6,F4,F0  
i4: ADDD   F2,F6,F0
```

- Considere las siguientes hipótesis de partida:
- Para reducir el número de ciclos máquina se permite que la FLOS distribuya hasta dos instrucciones en cada ciclo según el orden del programa.
- Una instrucción puede comenzar su ejecución en el mismo ciclo en que se distribuye a una estación de reserva.

- La operación **suma** tiene una latencia de **un ciclo** y la de **multiplicación** de **dos ciclos**.
- Se permite que una instrucción reenvíe su resultado a instrucciones dependientes durante su último ciclo de ejecución. De esta forma una instrucción a la espera de un resultado puede comenzar su ejecución en el siguiente ciclo si se detecta una coincidencia.

- Los valores de etiqueta 01, 02 y 03 se utilizan para identificar las tres estaciones de reserva de la unidad funcional de suma, mientras que 04 y 05 se utilizan para identificar las dos estaciones de reserva de la unidad funcional de multiplicación/división. Estos valores de etiqueta son los ID de las estaciones de reserva.
- Inicialmente, el valor de los registros es $F0=2.0$, $F2=2.5$, $F4=8.0$ y $F6=4.0$.

i1: MULTD F0,F6,F2
i2: ADDD F4,F2,F6
i3: ADDD F6,F4,F0
i4: ADDD F2,F6,F0

FR

	bitOc.	etiqueta	dato
F0			2.0
F2			2.5
F4			8.0
F6			4.0

RS

ID	eti_1	oper_1	eti_2	oper_2
01				
02				
03				

SUMA

RS

ID	eti_1	oper_1	eti_2	oper_2
04				
05				

MULT/DIV

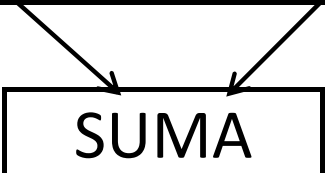
Ciclo 1: Se distribuye i1 e i2
 i1: MULTD F0,F6,F2
 i2: ADDD F4,F2,F6
 Se ejecuta RS 01 1/1
 Se ejecuta RS 04 1/2
 Se envía **RS 01: 6.5** al CDB

FR

	bitOc.	etiqueta	dato
F0	Si	04	2.0
F2			2.5
F4	Si	01	8.0
F6			4.0

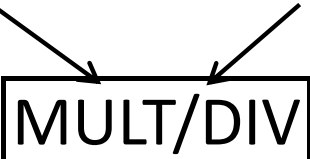
RS

ID	etiq_1	oper_1	etiq_2	oper_2
i2:01	00	2.5	00	4.0
02				
03				



RS

ID	etiq_1	oper_1	etiq_2	oper_2
i1:04	00	4.0	00	2.5
05				



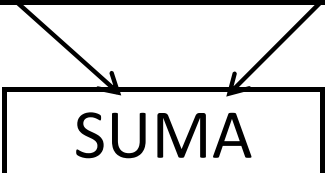
Ciclo 2: Se distribuye i3 e i4
i3: ADDD F6,F4,F0
i4: ADDD F2,F6,F0
Se actualiza el valor de **RS 01: 6.5**
Se vacía RS 01
Se ejecuta RS 04 2/2
Se envía **RS 04: 10.0** al CDB
NO se ejecuta RS 02, 03

FR

	bitOc.	etiqueta	dato
F0	Si	04	2.0
F2	Si	03	2.5
F4			6.5
F6	Si	02	4.0

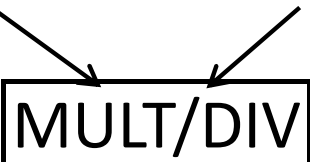
RS

ID	etiq_1	oper_1	etiq_2	oper_2
01				
i3: 02	04	xx	00	6.5
i4: 03	02	xx	00	6.5



RS

ID	etiq_1	oper_1	etiq_2	oper_2
i1: 04	00	4.0	00	2.5
05				



Ciclo 3: Se actualiza el valor de RS 04: 10.0

Se vacía RS 04

Se ejecuta RS 02 1/1

Se envía RS 02: 16.5 al CDB

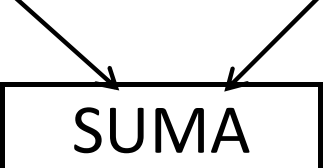
NO se ejecuta RS 03

FR

	bitOc.	etiqueta	dato
F0			10.0
F2	Si	03	2.5
F4			6.5
F6	Si	02	4.0

RS

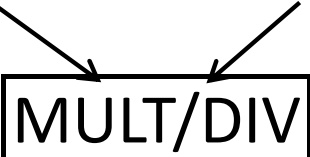
ID	etiq_1	oper_1	etiq_2	oper_2
01				
i3: 02	04	10.0	00	6.5
i4: 03	02	xx	00	6.5



SUMA

RS

ID	etiq_1	oper_1	etiq_2	oper_2
04				
05				



MULT/DIV

Ciclo 4: Se actualiza el valor de RS 02: 16.5
 Se vacía RS 02
 Se ejecuta RS 03 1/1
 Se envía RS 03: 23.0 al CDB

FR

	bitOc.	etiqueta	dato
F0			10.0
F2	Si	03	2.5
F4			6.5
F6			16.5

RS

ID	eti_1	oper_1	eti_2	oper_2
01				
02				
i4: 03	02	16.5	00	6.5

RS

ID	eti_1	oper_1	eti_2	oper_2
04				
05				

Ciclo 5: Se actualiza el valor de RS 03: 23.0
Se vacía RS 03

FR

	bitOc.	etiqueta	dato
F0			10.0
F2			23.0
F4			6.5
F6			16.5

RS

ID	eti_1	oper_1	eti_2	oper_2
01				
02				
03				

SUMA

RS

ID	eti_1	oper_1	eti_2	oper_2
04				
05				

MULT/DIV