AT5118 – Arquitectura e Ingeniería de Computadores I

Juan Antonio Maestro

Universidad Antonio de Nebrija

Principios básicos de la segmentación

Curso 2011-2012

¿Qué es la Segmentación?

- ¿Qué ocurre si una U.F. no es lo suficientemente rápida?
  - Solución tecnológica: Acelerarla con componentes más rápidos. Limitada.
  - Solución arquitectónica: Segmentación.

Definición de Segmentación

- La segmentación es una técnica por la que se *divide* una U.F. en varias etapas más rápidas, a fin de mejorar el rendimiento de la misma.
- Gracias a la segmentación, se permite la coexistencia de distintos datos en las etapas de la U.F.

Segmentación lineal

- Todas las etapas de la unidad se ejecutan en orden secuencial.
- No hay posibilidad de lazos hacia atrás.
- Cada etapa ha de producir su resultado antes del siguiente ciclo de reloj.
- Los resultados de cada etapa se guardan en registros (*latches*).

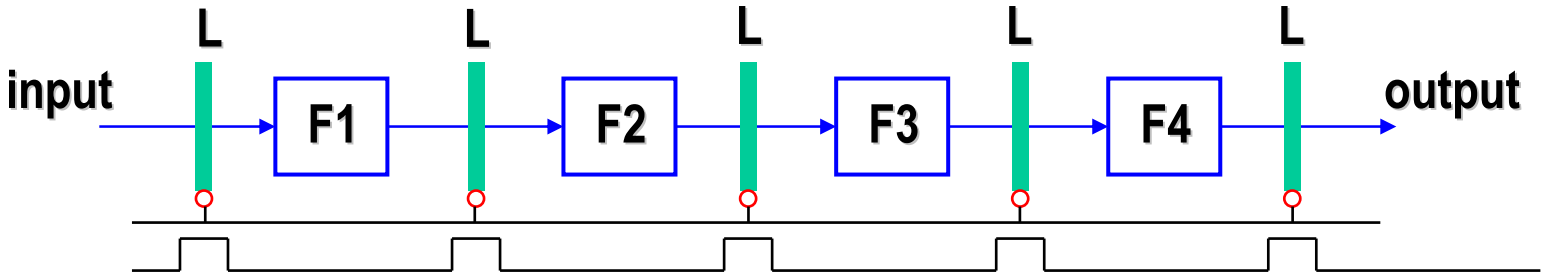
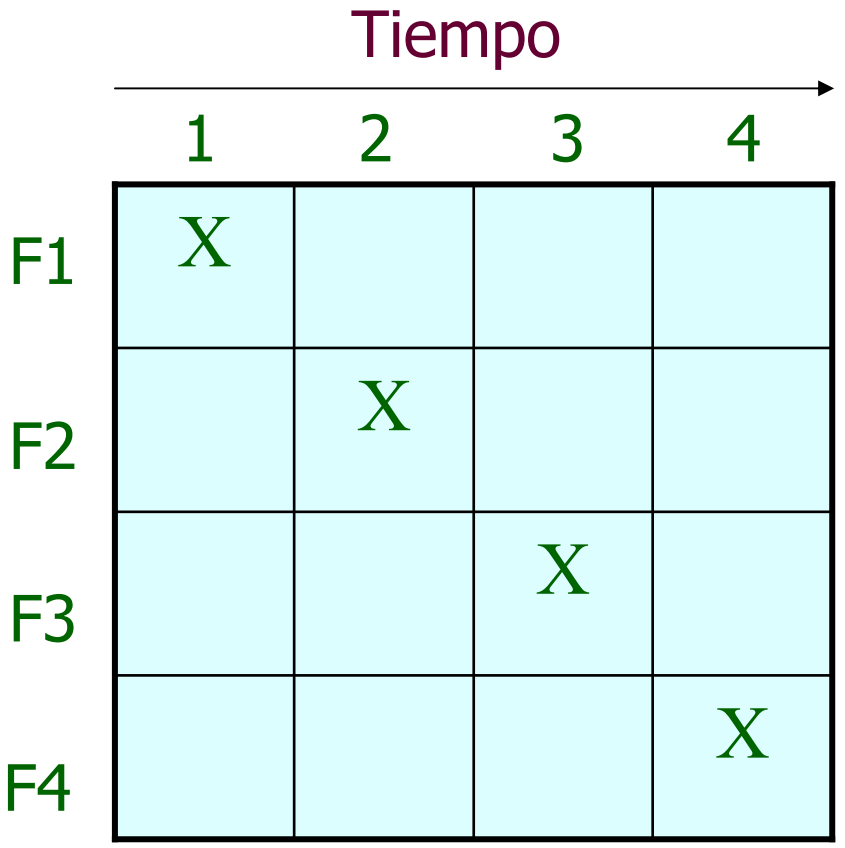


Tabla de reserva

- Tabla en la que se indica la ocupación de cada etapa frente al tiempo (ciclos de reloj).
- En una unidad con segmentación lineal, la tabla de reserva es trivial: siempre es una diagonal, ya que no hay posibilidad de vuelta atrás.



Métricas de la segmentación lineal (I)

- Tiempo de ejecución de *n* datos en una U.F. segmentada con *k* etapas:

$$T_k = [k + (n-1)] \cdot \tau$$

k corresponde al número de ciclos de iniciación. Después, se necesita un único ciclo por dato.

- Tiempo de ejecución de *n* datos en la misma U.F. sin segmentar:

$$T_1 = n \cdot k \cdot \tau$$

Métricas de la segmentación lineal (II)

- *Speed-up* de una unidad segmentada frente a una no segmentada:

$$SU_k = \frac{T_1}{T_k} = \frac{n \cdot k \cdot \tau}{[k + (n-1)] \cdot \tau} = \frac{n \cdot k}{k + (n-1)}$$

- Máximo S.U.: Cuando *n* tiende a infinito, S.U. tiende a *k*. Límite teórico difícil de conseguir.

- ¿Es conveniente que *k* sea arbitrariamente grande para maximizar el S.U.? ……

Métricas de la segmentación lineal (III)

- Eficiencia: *Speed-up* conseguido por etapa.

$$E_k = \frac{SU_k}{k} = \frac{n}{k + (n-1)}$$

- Límite inferior: $1/k$ (cuando $n=1$)
- Límite superior: 1 (cuando $n \to \infty$)

Métricas de la segmentación lineal (IV)

- *Throughput*: Número de tareas por segundo.

$$H_k = \frac{n}{[k + (n-1)] \cdot \tau}$$

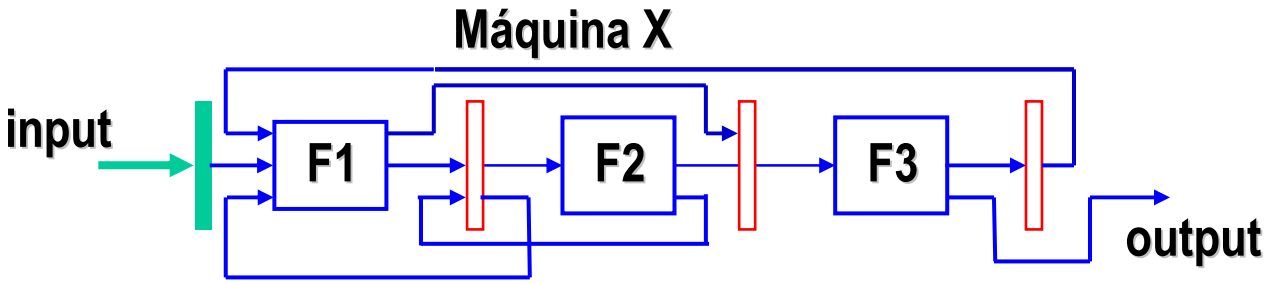
- Óptimo: 1 (cuando $n \to \infty$). En este caso, el tiempo de iniciación de la unidad es despreciable.

Segmentación no lineal

- Las etapas no se ejecutan en orden secuencial.
- Sí hay posibilidad de lazos hacia atrás.
- Cada etapa se puede visitar más de una vez en cada iniciación de datos.

- La tabla de reserva no es única en este tipo de sistemas. Depende del algoritmos de encaminamiento de datos.

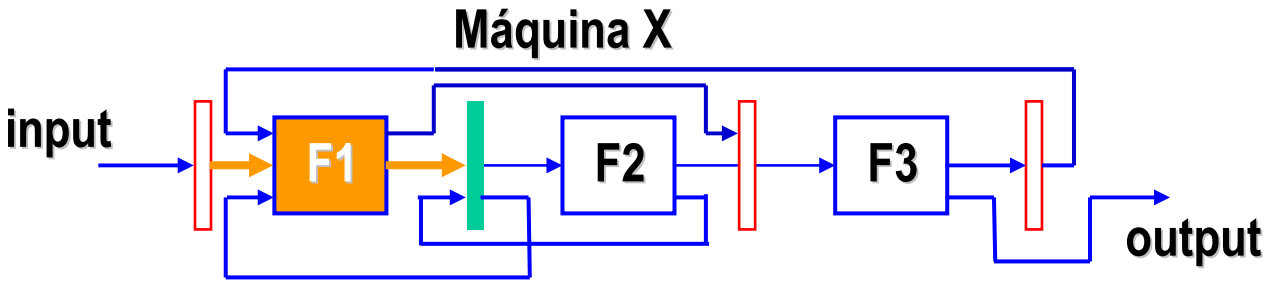
Segmentación no lineal



Tiempo →

Etapa ↓	1	2	3	4	5	6	7
F1							
F2							
F3							

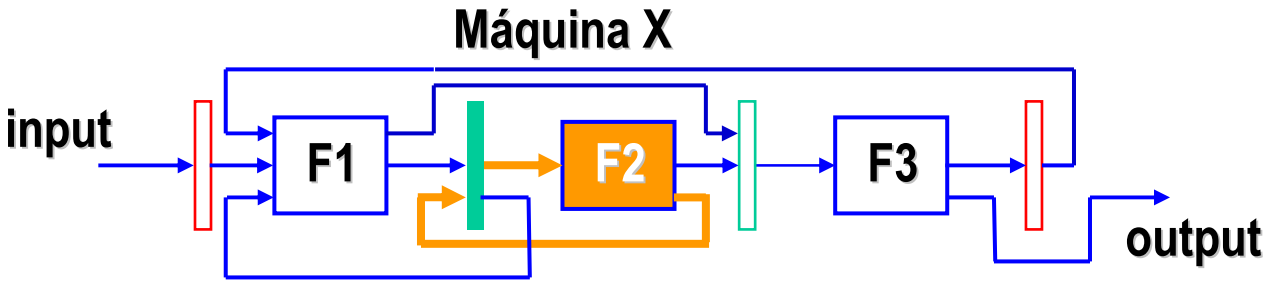
Segmentación no lineal



Tiempo →

Etapa ↓	1	2	3	4	5	6	7
F1	X						
F2							
F3							

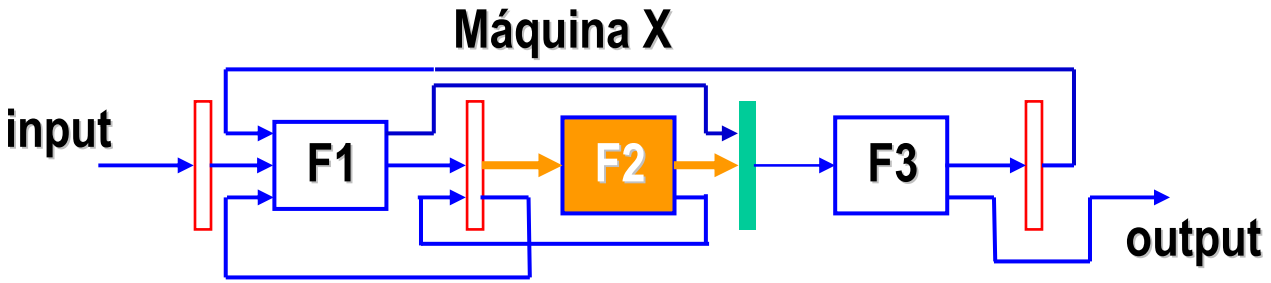
Segmentación no lineal



Tiempo →

Etapa ↓	1	2	3	4	5	6	7
F1	X						
F2		X					
F3							

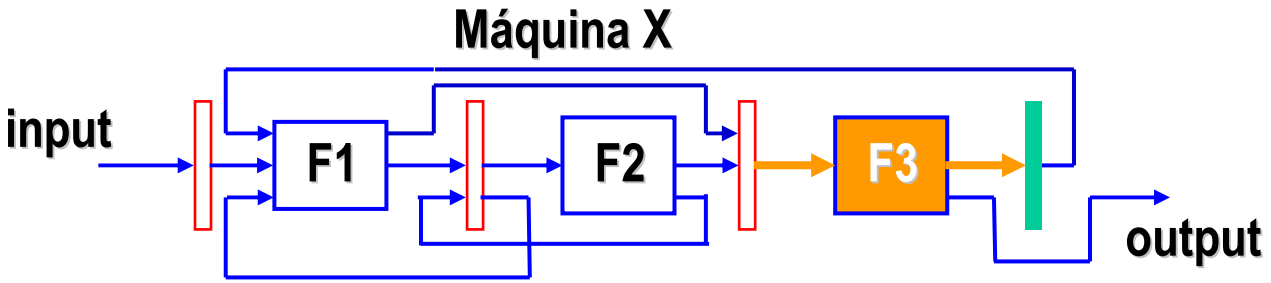
Segmentación no lineal



Tiempo →

Etapa ↓	1	2	3	4	5	6	7
F1	X						
F2		X	X				
F3							

Segmentación no lineal

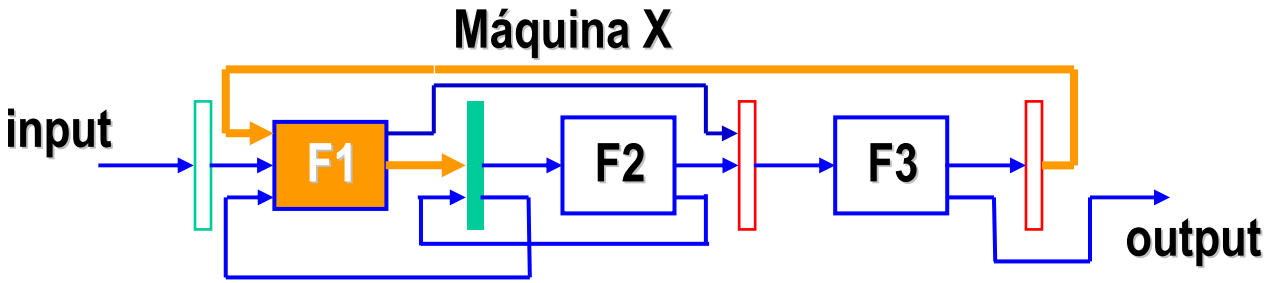


Tiempo →

Etapa ↓

	1	2	3	4	5	6	7
F1	X						
F2		X	X				
F3				X			

Segmentación no lineal

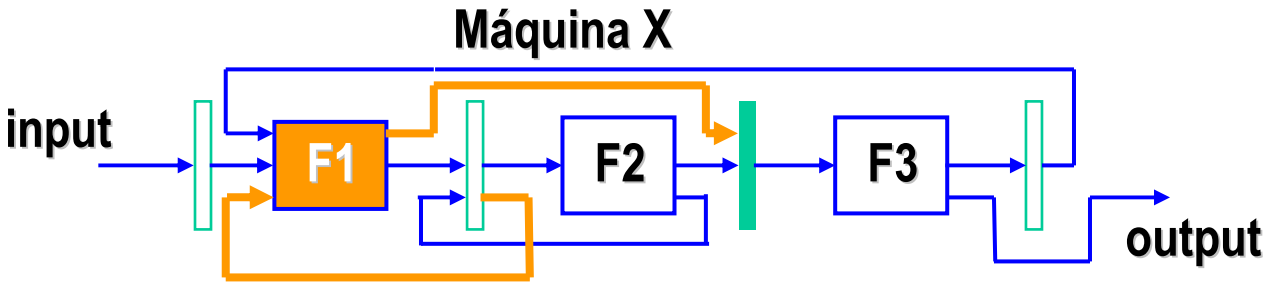


Tiempo →

Etapa ↓

	1	2	3	4	5	6	7
F1	X				X		
F2		X	X				
F3				X			

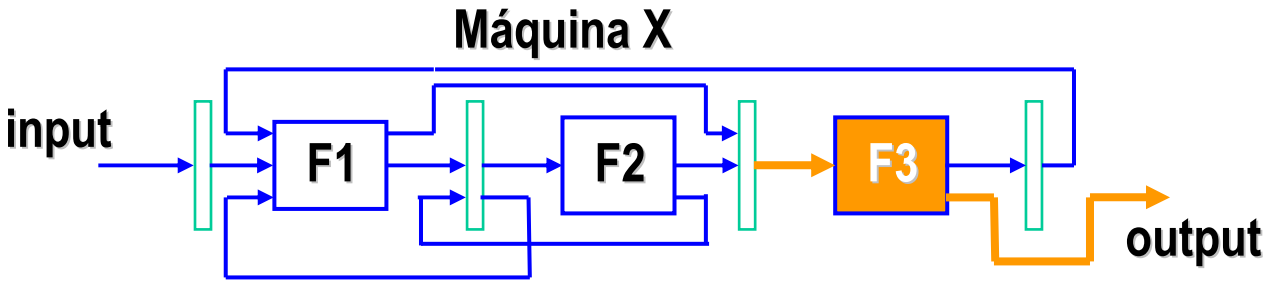
Segmentación no lineal



Tiempo →

Etapa ↓	1	2	3	4	5	6	7
F1	X				X	X	
F2		X	X				
F3				X			

Segmentación no lineal



Tiempo →

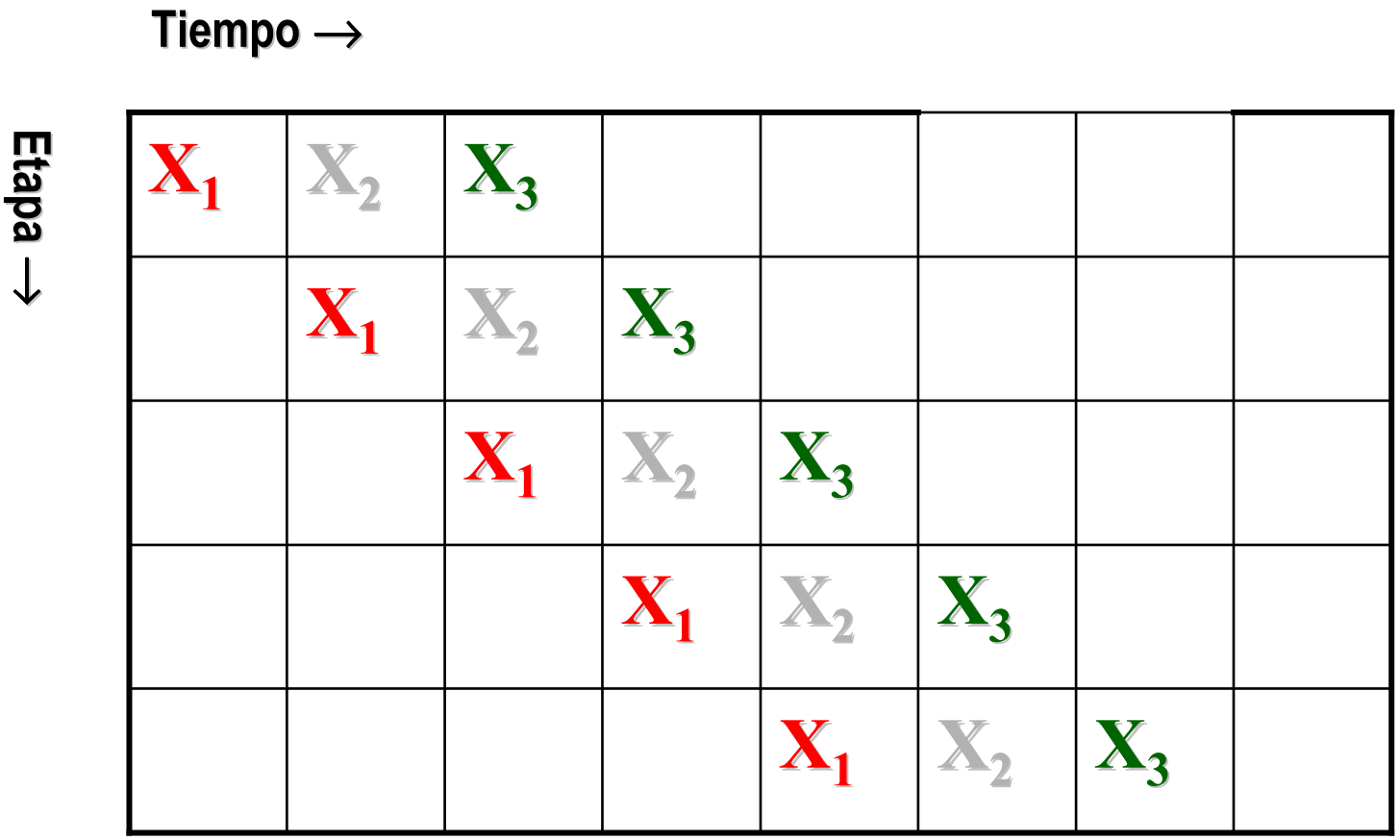
Etapa ↓

	1	2	3	4	5	6	7
F1	X				X	X	
F2		X	X				
F3				X			X

Latencia y colisión

- Latencia: Número de ciclos de reloj entre dos iniciaciones consecutivas.
  - La latencia óptima es 1.
  - Siempre es posible en segmentación lineal.
  - Pocas veces posible en no lineal.
- Colisión: Situación en la que datos de dos inicializaciones distintas tratan de acceder a la misma etapa.
- Latencia prohibida: Aquella que produce una colisión.

Latencia en la segmentación lineal



Segmentación lineal: Latencia 1 siempre alcanzable

Latencia en la segmentación no lineal

Latencia 1: **Prohibida**

	1	2	3	4	5	6	7
A	X	Z					
B		X	X Z	Z			
C			X	X Z	Z		
D					X	Z	X
E						X	Z
F							
G							
H							

Latencia en la segmentación no lineal

Latencia 2: **Prohibida**

	1	2	3	4	5	6	7
A	X		Z				
B		X	X	Z	Z		
C			X	X	Z	Z	
D					X		X Z
E						X	
F							
G							
H							

Latencia en la segmentación no lineal

Latencia 3: **Permitida**

	1	2	3	4	5	6	7
A	X			Z			
B		X	X		Z	Z	
C			X	X		Z	Z
D					X		X
E						X	
F							
G							
H							

Latencia en la segmentación no lineal

Lat	0	1	2	3	4	5	6
F1	1				1	1	
F2		1	1				
F3				1			1

Lat	0	1	2	3	4	5	6	7
F1	1	2			1	1,2	2	
F2		1	1,2	2				
F3				1	2		1	2

Latencia 1: **Prohibida**

Lat	0	1	2	3	4	5	6	7	8
F1	1		2		1	1	2	2	
F2		1	1	2	2				
F3				1		2	1		2

Latencia 2: **Permitida**

¿Cómo identificar todas las latencias prohibidas?

Identificación de latencias prohibidas

- Latencias prohibidas: Distancias entre cada uno de los pares de marcas en la tabla de reserva.

- Toda latencia prohibida es siempre menor que el número de columnas en la tabla de reserva.

Ciclo de Latencia y Latencia Media

- Ciclo de latencia: Secuencia de latencias que se repiten sin que exista colisión. No puede contener latencias prohibidas. Ej.: (1,3)=1,3,1,3,1,3,1,….
- Ciclo constante: Aquel que sólo tiene una latencia. Ej.: (3)=3,3,3,3,3,3,3,….
- Latencia media: Suma de las latencias de un ciclo, entre el número de ciclos. Ej.:
  - AL(1,3)=(1+3)/2=2
  - AL(3)=3/1=3

Mínima Latencia Media (MAL)

- Objetivo: Encontrar el ciclo de latencia con la mínima latencia media (*MAL*).
- Al minimizar la latencia se consigue un mayor rendimiento de la unidad funcional.

Vector de Colisión

- Sea *m* la máxima latencia prohibida de una tabla de reserva.
- Se define Vector de Colisión, al vector $C=(C_m, C_{m-1}, \ldots, C_2, C_1)$, siendo C_n igual a 1 si la latencia *n* es prohibida, y 0 si es permitida.
- Ej.: Sistema con latencias prohibidas 2,4,5,7 $\Rightarrow$ C=(1011010).

Pasos para calcular el ciclo de latencia óptimo

- Calcular el Vector Inicial de Colisión.
- Calcular el Diagrama de Estado de la Tabla de Reserva.
- Sobre el Diagrama, identificar todos los ciclos de latencia que sean *greedy* (aquellos en los que se elige la menor latencia posible en cada estado).
- Calcular la latencia media de cada uno de estos ciclos. La menor es la *MAL*.

Algoritmo para calcular el Diagrama de Estados

- Se parte del Vector de Colisión Inicial, que es el Estado inicial del Diagrama. Se identifican las latencias permitidas.
- Cada latencia da lugar a un arco. El estado destino de cada arco se calcula mediante la operación lógica:

$$C_{final} = (\vec{C}^{n}{}_{actual}) + (C_{inicial})$$

$\vec{C}^{n}$ es el desplazamiento lógico a la derecha de C (n posiciones, siendo n la latencia del arco).

$+$ es la operación O-lógica.

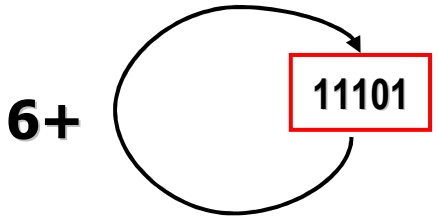
- Si C_{final} da lugar a un nuevo estado, se añade al diagrama y se repite el proceso.

Ejemplo de cálculo de ciclos de latencia

Lat	0	1	2	3	4	5	6
F1	**1**				**1**	**1**	
F2		**1**	**1**				
F3				**1**			**1**

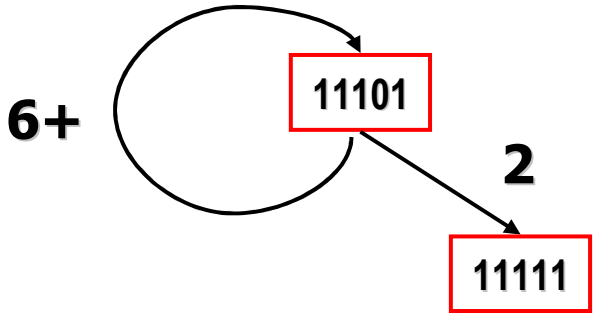
- Latencias prohibidas: 1, 3, 4, 5
- Latencias permitidas: 2, 6+
- Vector de colisión inicial: C=(1,1,1,0,1)

Ejemplo de cálculo de ciclos de latencia



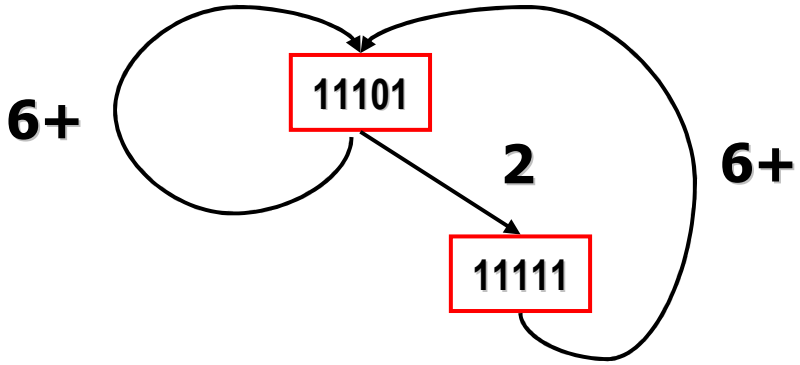
- Toda latencia mayor que C_m es siempre permitida, y genera un arco hacia el estado inicial

Ejemplo de cálculo de ciclos de latencia



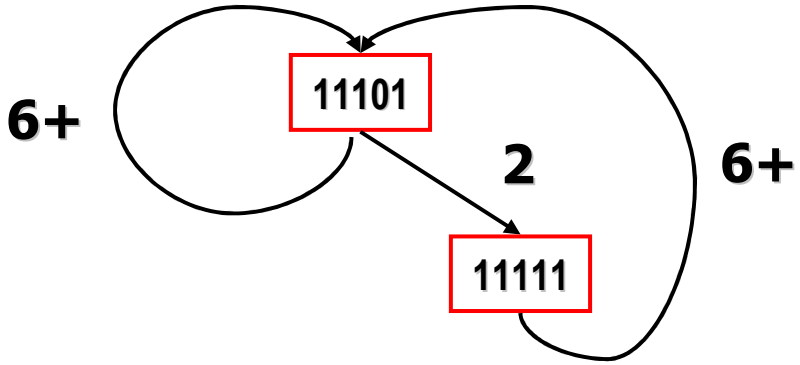
- Latencia 2:
- $C_{final} = (00111) + (11101) = (11111)$

Ejemplo de cálculo de ciclos de latencia



- Desde (11111) no hay latencias permitidas, excepto 6+, que retorna al estado inicial.
- Todos los estados han sido explorados: Diagrama de Estados final.

Ejemplo de cálculo de ciclos de latencia



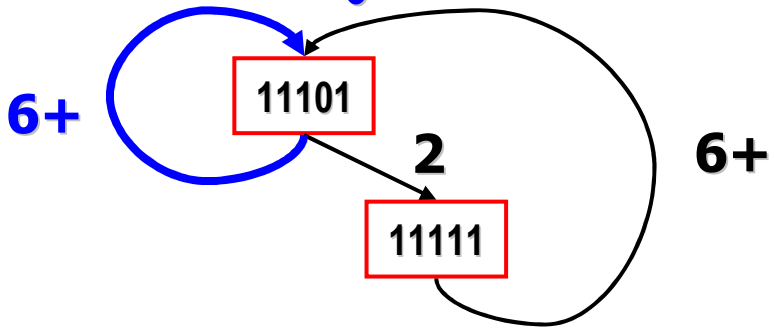
- Ciclos de latencia:
  - Desde (11101):
    - (6): No *greedy;* (2,6): *Greedy*
  - Desde (11111):
    - (6,2): *Greedy*. Igual que (2,6)

Único ciclo *greedy*: (2,6)
AL(2,6)=4 $\Rightarrow$ **MAL**

Ejemplo de cálculo de ciclos de latencia

Lat	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
F1	1				1	1	2				2	2	3				3	3	4		
F2		1	1					2	2					3	3					4	4
F3				1			1			2			2			3			3		

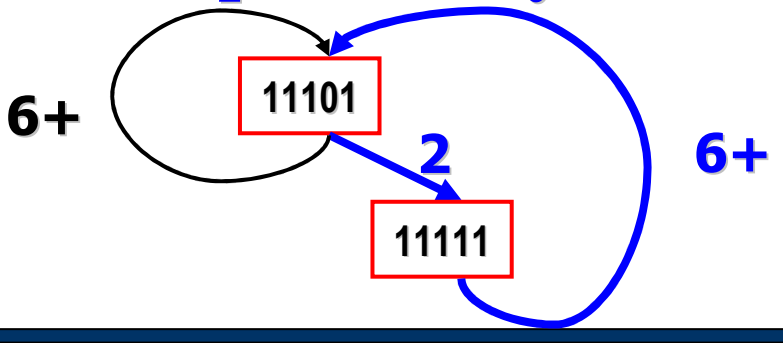
6 6 6



Planificación 1:
Latencias= <6, 6, 6, 6, 6, ..>
Ciclo de latencias= (6)
AL = 6

Lat	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
F1	1		2		1	1	2	2	3		4		3	3	4	4	5		6		5
F2		1	1	2	2					3	3	4	4					5	5	6	6
F3				1		2	1		2			3		4	3		4			5	

2 6 2 6 2



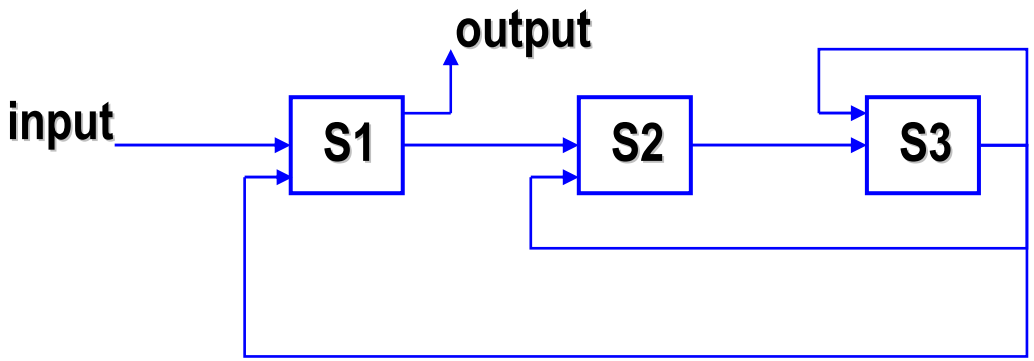
Planificación 2: (Greedy)
Latencias= <2, 6, 2, 6, 2, 6, ..>
Ciclo de latencias = (2, 6)
AL = (2+6)/2 = 4 = MAL

¿Es este el mejor rendimiento alcanzable?

- La mínima latencia media (MAL) asociada al mejor ciclo *greedy* es óptima para la tabla de reserva inicial.
- Sin embargo, puede no ser óptima para el sistema.

- Solución: Calcular otra tabla de reserva para el mismo sistema que derive en una mejor latencia media.
- En general, MAL tiene como:
  - límite inferior el máximo número de marcas en cualquier fila de la tabla de reserva
  - límite superior la latencia media del mejor ciclo *greedy*.

Modificación de la Tabla de Reserva

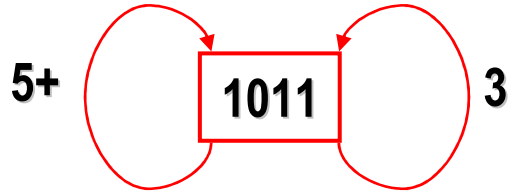


	1	2	3	4	5
S1	X				X
S2		X		X	
S3			X	X	

Vector de Colisión Inicial

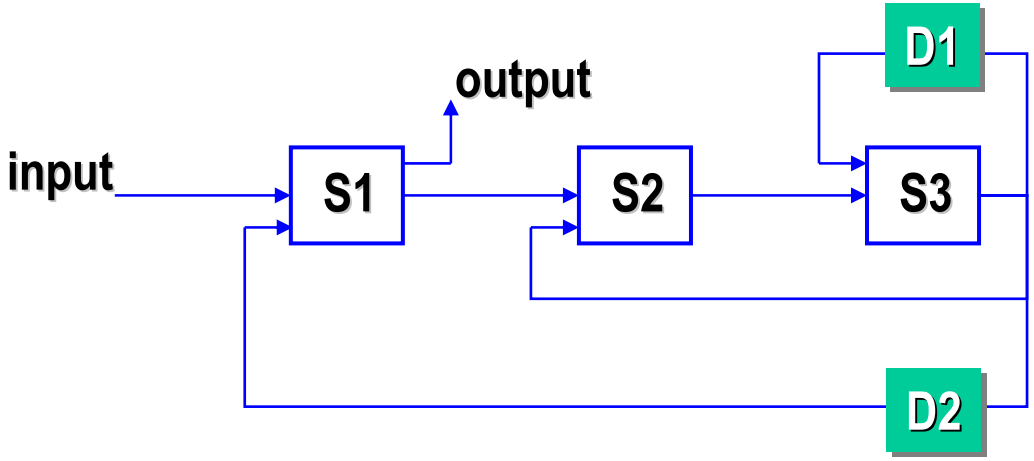
1011

Modificación de la Tabla de Reserva



- Ciclo *greedy*: (3) ⇒ MAL=3
- ¿Es posible mejorarlo?
  - El número máximo de marcas en una fila de la tabla de reservas es 2.
  - ¿Se puede alcanzar MAL=2 modificando la tabla?

Modificación de la Tabla de Reserva

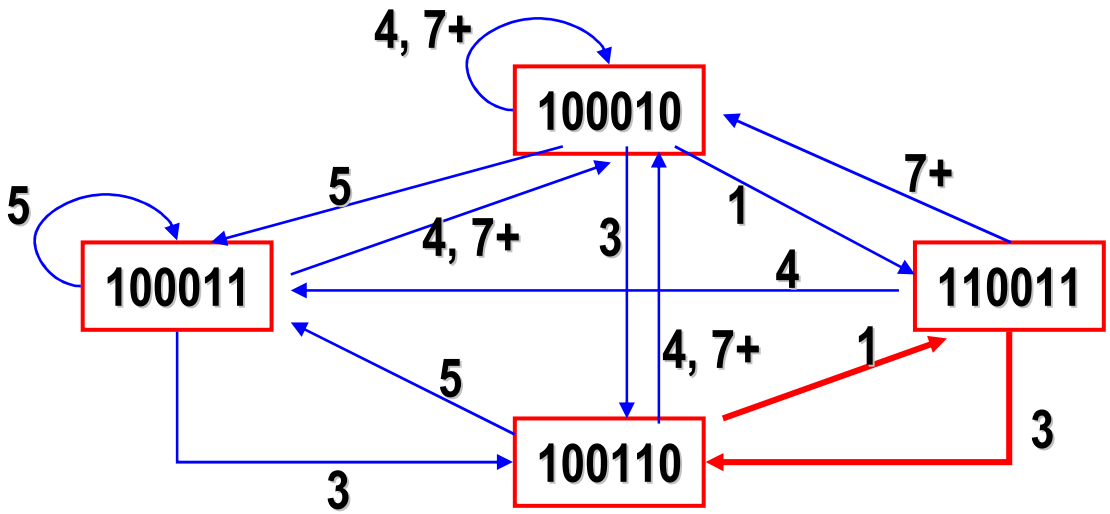


	1	2	3	4	5	6	7
S1	X				X	D2	X
S2		X		X			
S3			X	X D1	X		

Vector de Colisión Inicial

100010

Modificación de la Tabla de Reserva



- Ciclo *greedy*: (1,3) ⇒ MAL=2
- Introduciendo retrasos en el sistema se ha modificado la tabla de reserva, y se ha conseguido un ciclo mejor.
- ¿Contradictorio?

Métricas de la segmentación no lineal (I)

- Throughput: Número de tareas iniciadas por ciclo de reloj. Depende de la latencia:
  - H(3) = 1/3 x1 - - x2 - - x3 ….
  - H(1,3) = 2/4 = 1/2 x1x2 - - x3x4 - - x5 ….

Métricas de la segmentación no lineal (II)

- Eficiencia: % de utilización de las etapas en estado estacionario (tras las inicialización).

	1	2	3	4	5	6	7	8
S1	1			2	1		3	2
S2		1		1	2		2	3
S3			1	1		2	2	

Estacionario

E(3) = 6/9 = 66.7%

	1	2	3	4	5	6	7	8	9	10	11
S1	1	2			3	4	1	2	5	6	3
S2		1	2	1	2	3	4	3	4	5	6
S3			1	2	1	2	3	4	3	4	5

Estacionario

E(1,3) = 12/12 = 100%