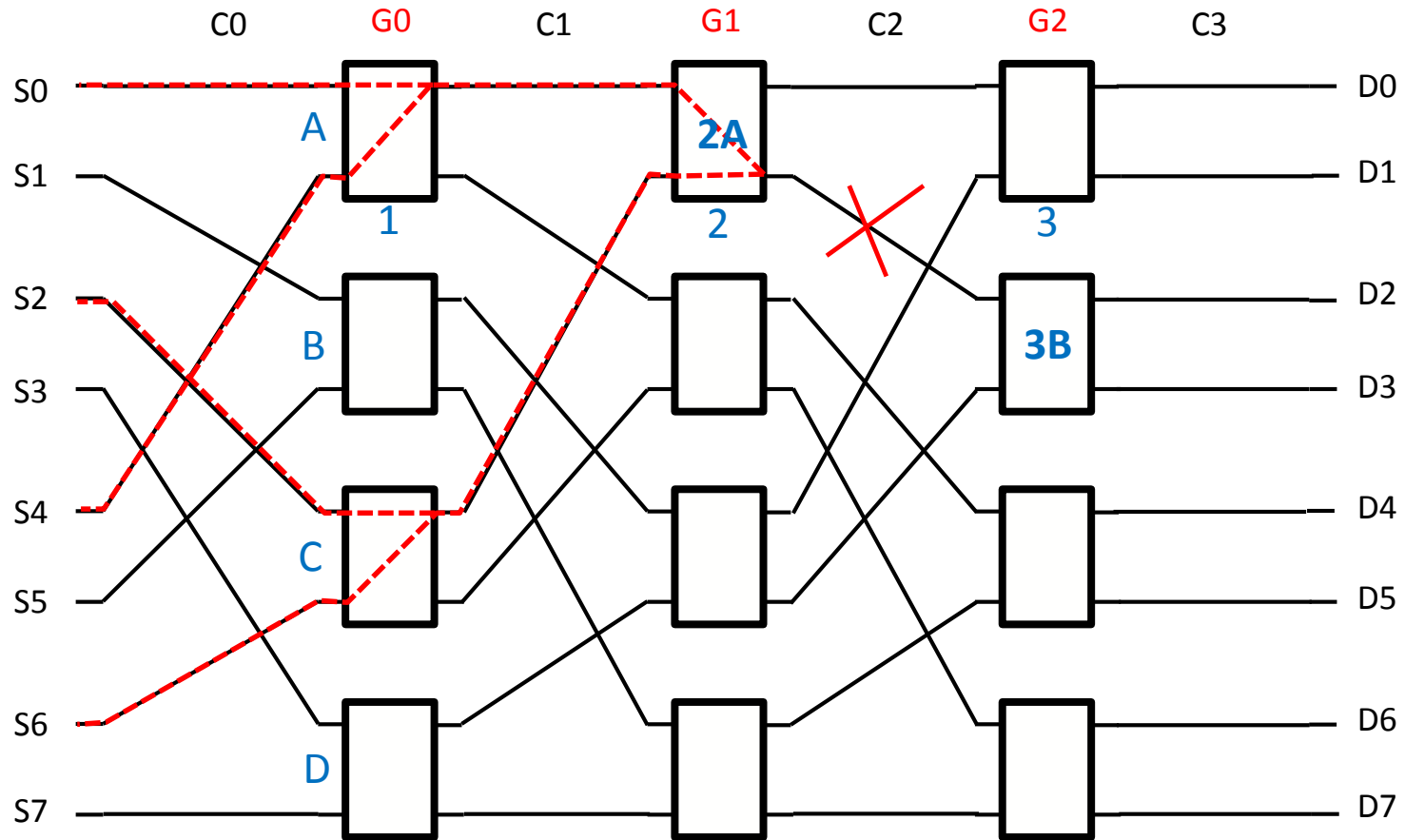


# Problema 1

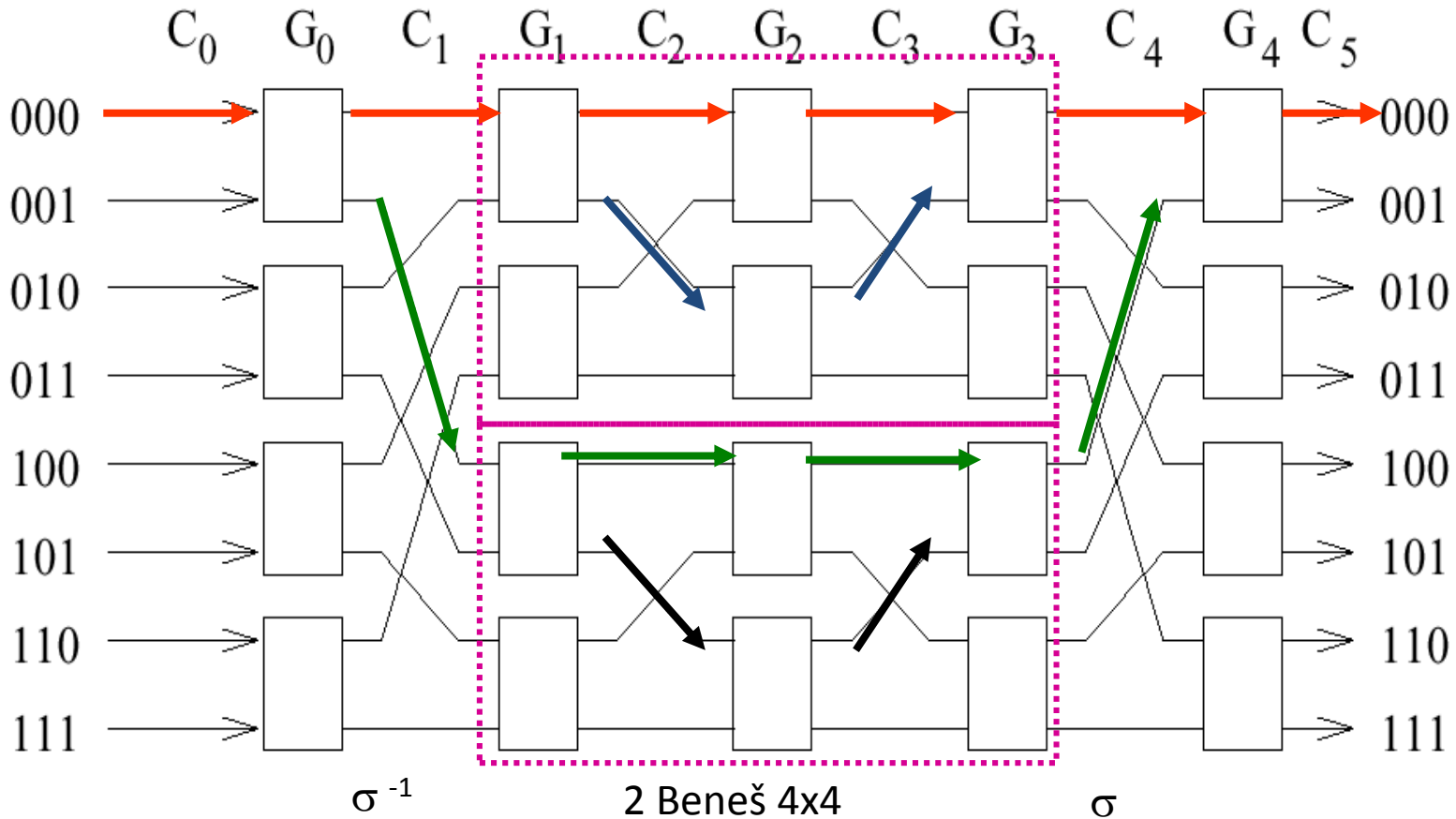
## Red Omega 8x8



Observando la figura vemos que como consecuencia de la conexión cortada:

- D2 queda desconectado de: S0, S4, S2 y S6
- D3 queda desconectado de las mismas fuentes

## Problema 2



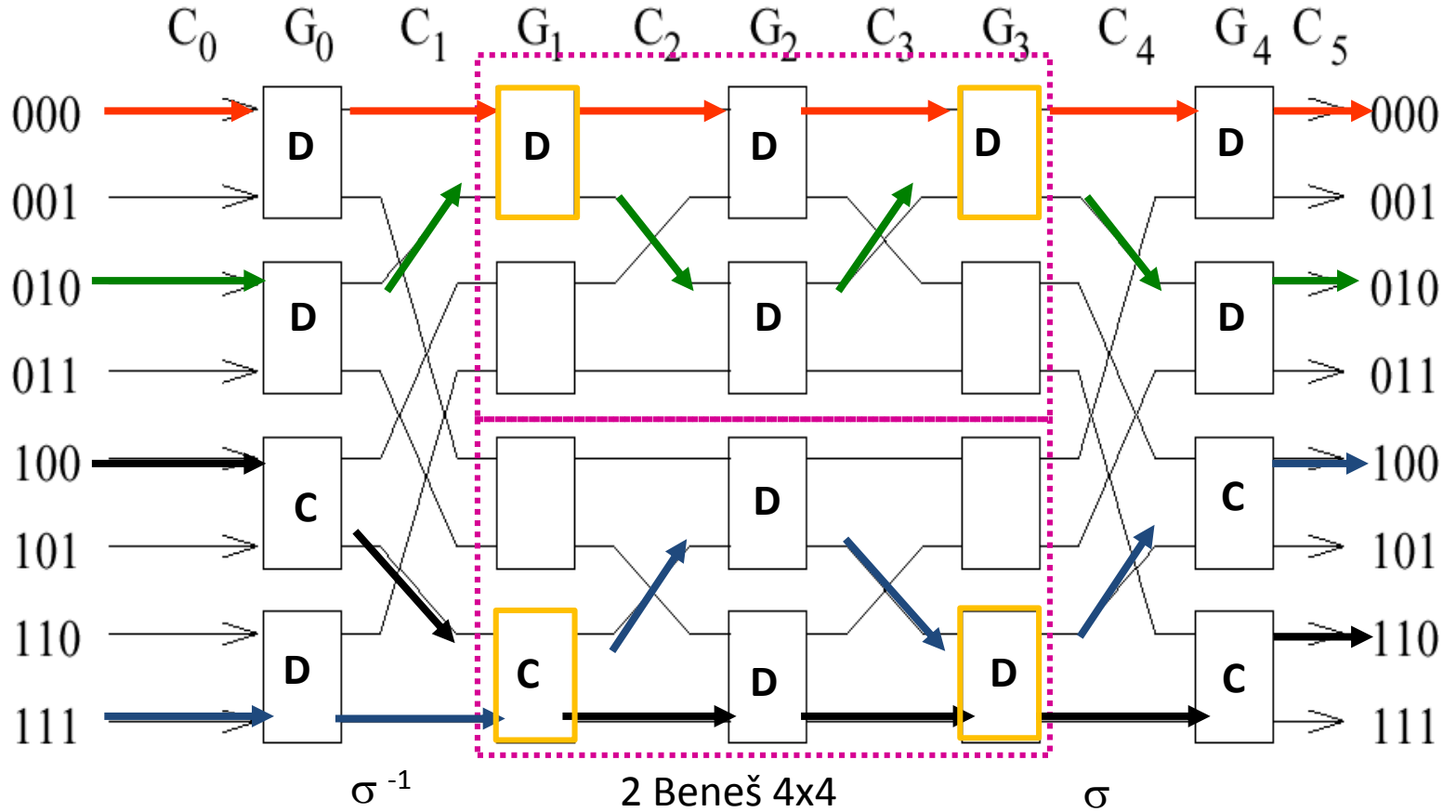
En etapa  $G_0$  hay dos caminos: directo (D) y cruzado (C).

En etapa  $G_1$ , para cada uno de los anteriores, hay dos caminos: directo y cruzado.

Con esto ya llegamos a  $G_2$  (etapa central). A partir de ahí, los caminos divergentes tienen que converger hacia D0. Por tanto los caminos se muestran en la tabla de al lado

	$G_0$	$G_1$	$G_2$	$G_3$	$G_4$
Cam 1	D	D	D	D	D
Cam 2	D	C	D	C	D
Cam 3	C	D	D	D	C
Cam 4	C	C	D	C	C

### Problema 3



Nodos con las dos entradas en uso

C: Conexión cruzada  
D: Conexión directa

## Problema 4

### a) Red Omega 4096x4096.

Calculamos el nº de etapas:

$N = K^n$ , donde  $N$ =puertos,  $k$  = base (grado del conmutador) = 2,  $n$  = etapas

$n = \log_2 4096 = 12$  etapas

$t_c = 2$  ns ;  $t_r$  (retardo conmutador) = 0,2 ns

Delay slots de un Load: viene dado por el viaje de ida y vuelta:

Procesador  $\rightarrow$  Mem (envío de la dir) + Mem  $\rightarrow$  Procesador (recepción del dato)

$t_{\text{acceso a Mem}} = 2 \times (\text{retardo conmutador}) \times (n^\circ \text{ etapas}) = 2 \times (0,2 \times 12) = 4,8$  ns.

Como 1 ciclo = 2 ns, esto implica que el delay slot debe ser un entero de ciclos mayor o igual que  $4,8/2$ . Es decir: delay slot = 3 ciclos

### b) Red Benes

Etapas =  $2 \times 12 - 1 = 23$

$t_{\text{acceso a Mem}} = 2 \times (0,2 \times 23) = 9,2$  ns.  $\rightarrow$  delay slot = 5 ciclos

## Problema 5

Red Omega de 1024 nodos con conmutadores 4x4.

En general:

$N = k^n$ , donde  $N$ =puertos,  $k$  = base (grado del conmutador),  $n$  = nº de etapas

Nº de conmutadores =  $(N/k) \times n$

En este problema:

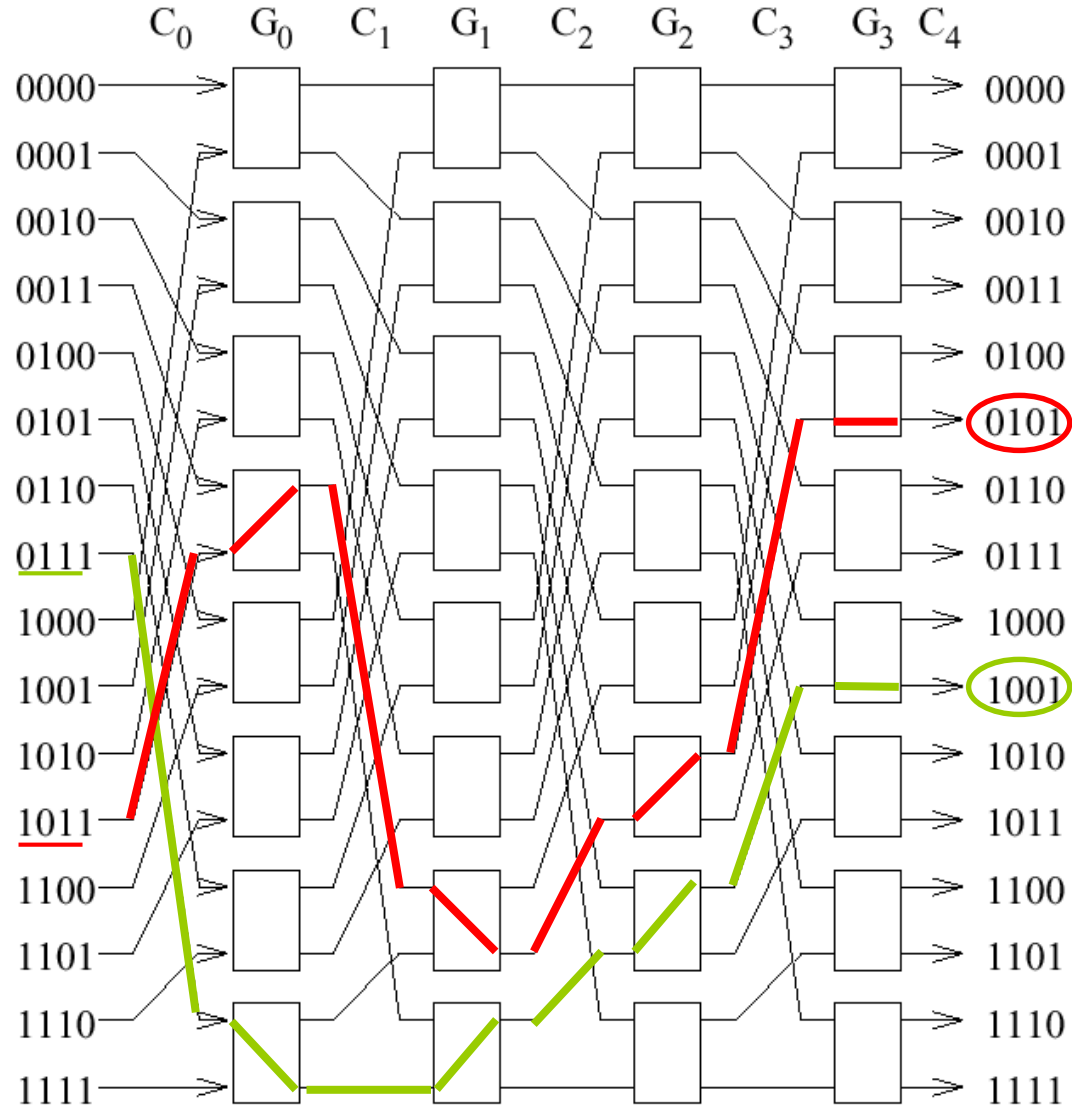
a)  $N = 1024 = 2^{10}$ ,  $k = 4 \rightarrow N = 4^n$ , Igualando  $2^{10} = 4^n$ ;  $(2^2)^5 = (2^2)^n \Rightarrow n = 5$  etapas

b) Nº de conmutadores =  $(2^{10} / 4) \times 5 = 256 \times 5 = 1280$  conmutadores

# Problema 6

Red Omega con 16 entradas, conmutadores 2x2. Conectar  $11 \rightarrow 5$  y  $7 \rightarrow 9$

- Dir destino:  $(d_3, d_2, d_1, d_0)$
- $d_i$  selecciona el funcionamiento de los conmutadores de la etapa  $n-i-1$ 
  - $d_i = 0 \rightarrow$  Salida superior
  - $d_i = 1 \rightarrow$  Salida inferior



Caminos  
Compatibles!

## Problema 7

Red Omega de 512 nodos con conmutadores 8x8.

En general:

$N = K^n$ , donde  $N$ =puertos,  $k$  = base (grado del conmutador),  $n$  = nº de etapas  
Nº de conmutadores =  $(N/k) \times n$

En este problema:

a)  $N = 512 = 2^9$ ,  $k = 8 \rightarrow N = 8^n$ , Igualando  $2^9 = 8^n$ ;  $2^9 = (2^3)^n \Rightarrow n = 3$  etapas

b) Nº de conmutadores =  $(2^9 / 8) \times 3 = 64 \times 3 = 192$  conmutadores

c)  $N = 4096 = 2^{12}$  nodos

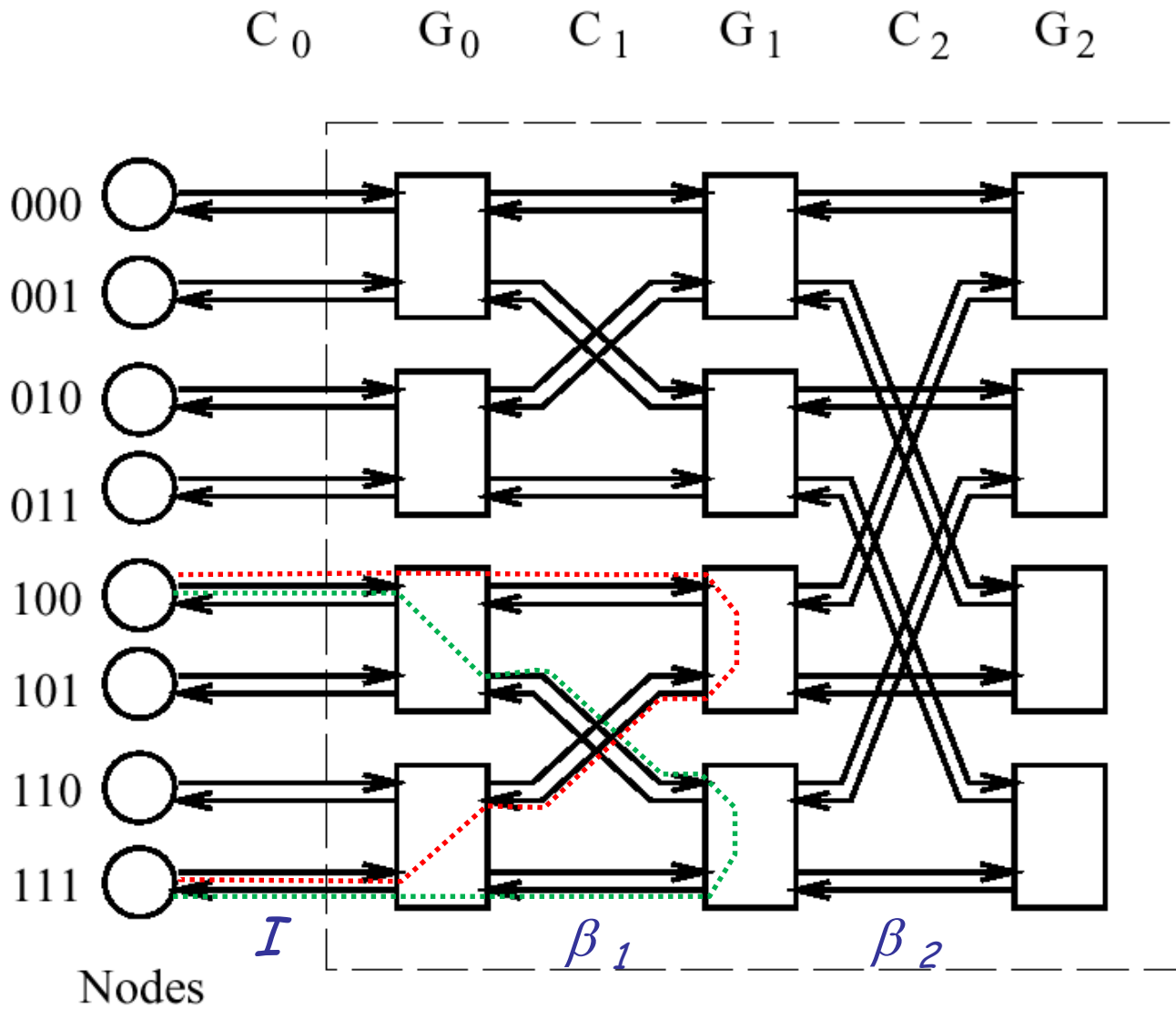
$2^{12} = (2^3)^n \Rightarrow n = 4$  etapas

Nº de conmutadores =  $(2^{12} / 8) \times 4 = 2^{11} = 2048$  conmutadores

Conmutadores adicionales =  $2048 - 192 = 1856$  conmutadores

Problema 9

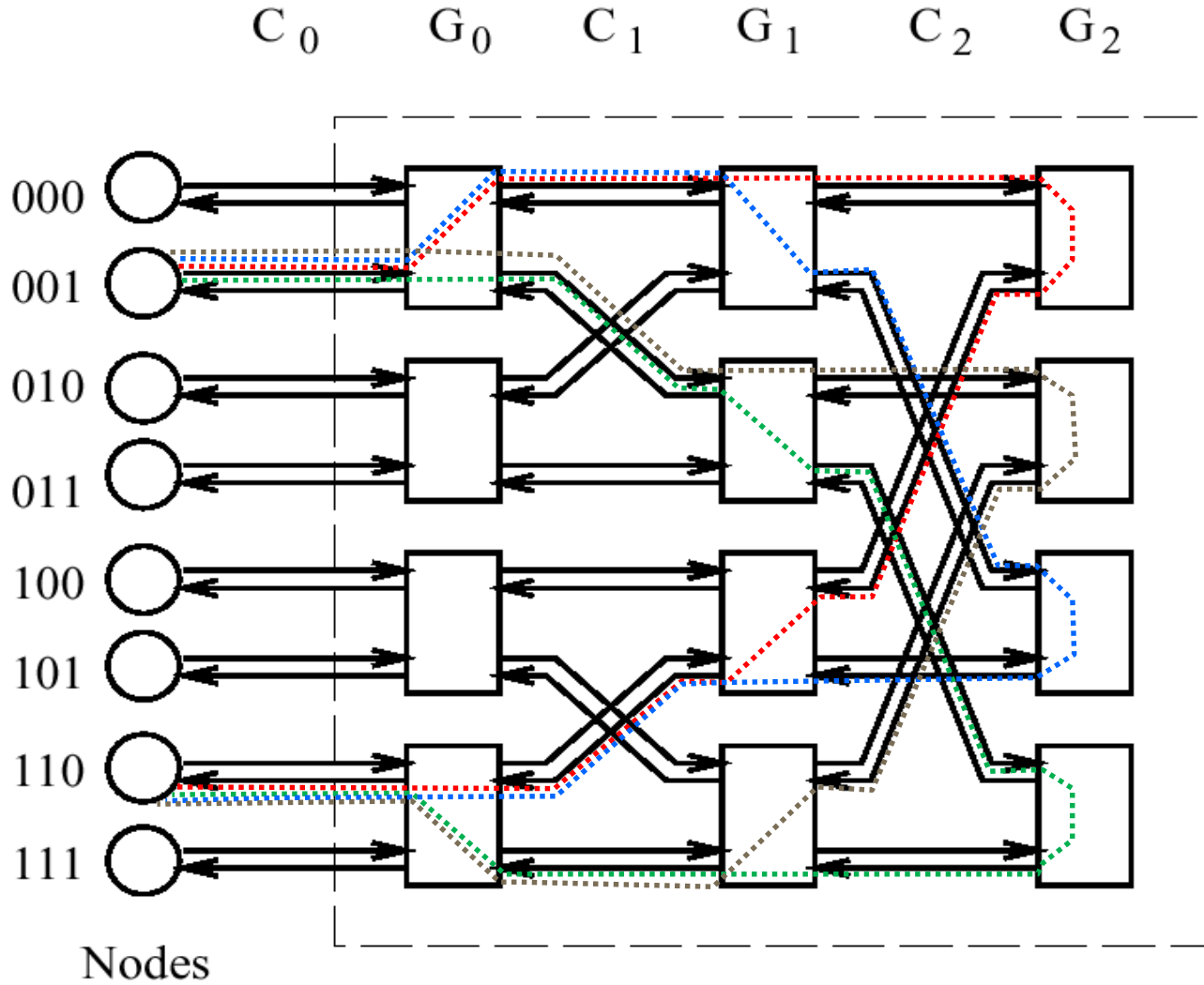
a)  $4 \rightarrow 7$





Problema 9

b)  $1 \rightarrow 6$



## Problema 10

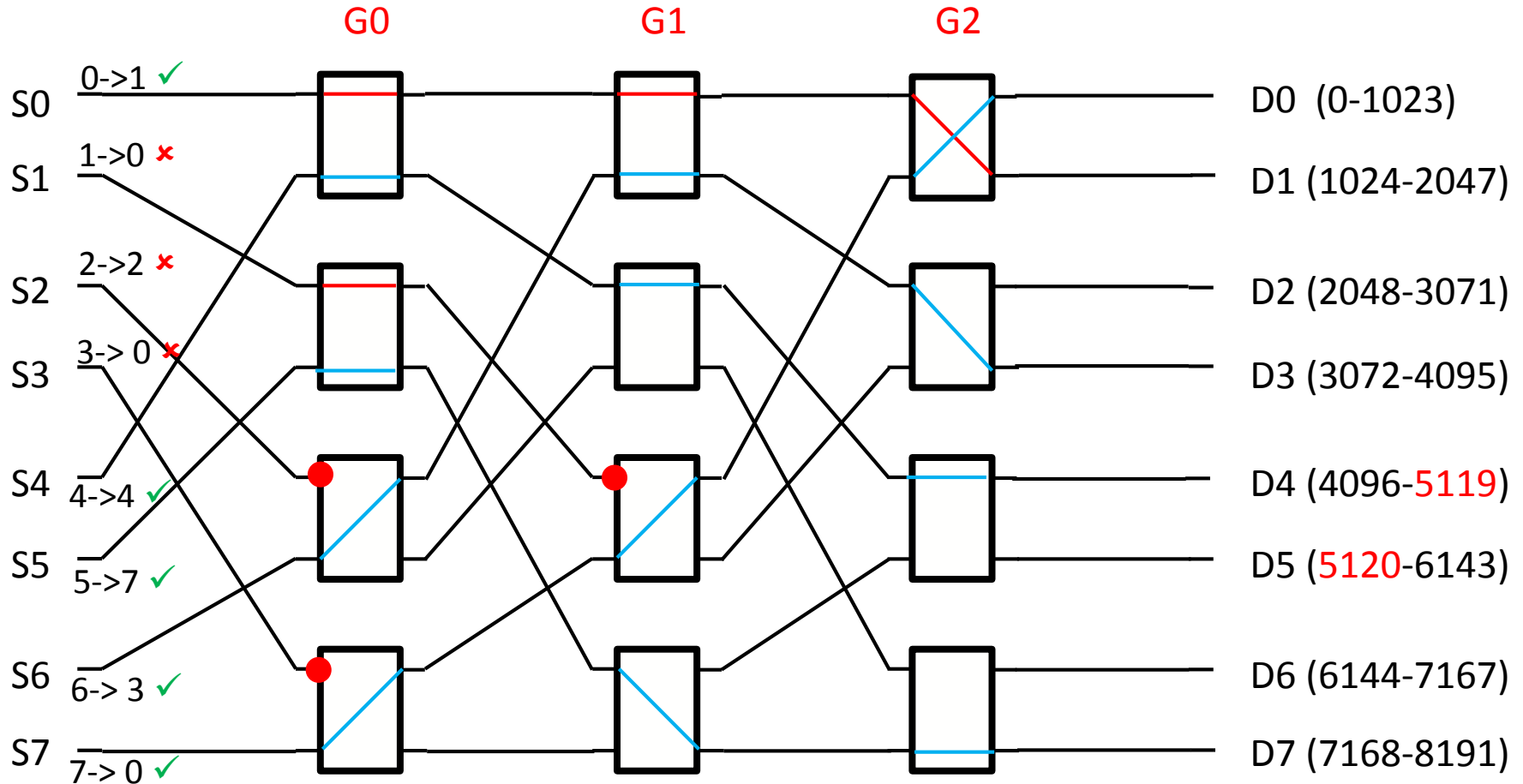
Ciclo	P0	P1	P2	P3	P4	P5	P6	P7
Intento 1	Dir: 1025 Con: 0->1	100 1->0	3000 2->2	0 3->0	5020 4->4	7800 5->7	4000 6->3	670 7->0
Exitos 1	✓	✗	✗	✗	✓	✓	✓	✓
Intento 2	Dir: 2048 Con: 0->2	100 1->0	3000 2->2	0 3->0			4001 6->3	
Exitos 2	✗	✗	✗	✓			✓	
Intento 3	Dir: 2048 Con: 0->2	100 1->0	3000 2->2	1 3->0				
Exitos 3	✗	✗	✓	✓				
Intento 4	Dir: 2048 Con: 0->2	100 1->0		2 3->0				
Exitos 4	✓	✗		✓				
Intento 5	Dir: 8000 Con: 0->7	100 1->0						
Exitos 5	✓	✓						
Intento 6		2500 1->2						
Exitos 6		✓						
Intento 7		7500 1->7						
Exitos 7		✓						

# Problema 10

## Ciclo de acceso 1

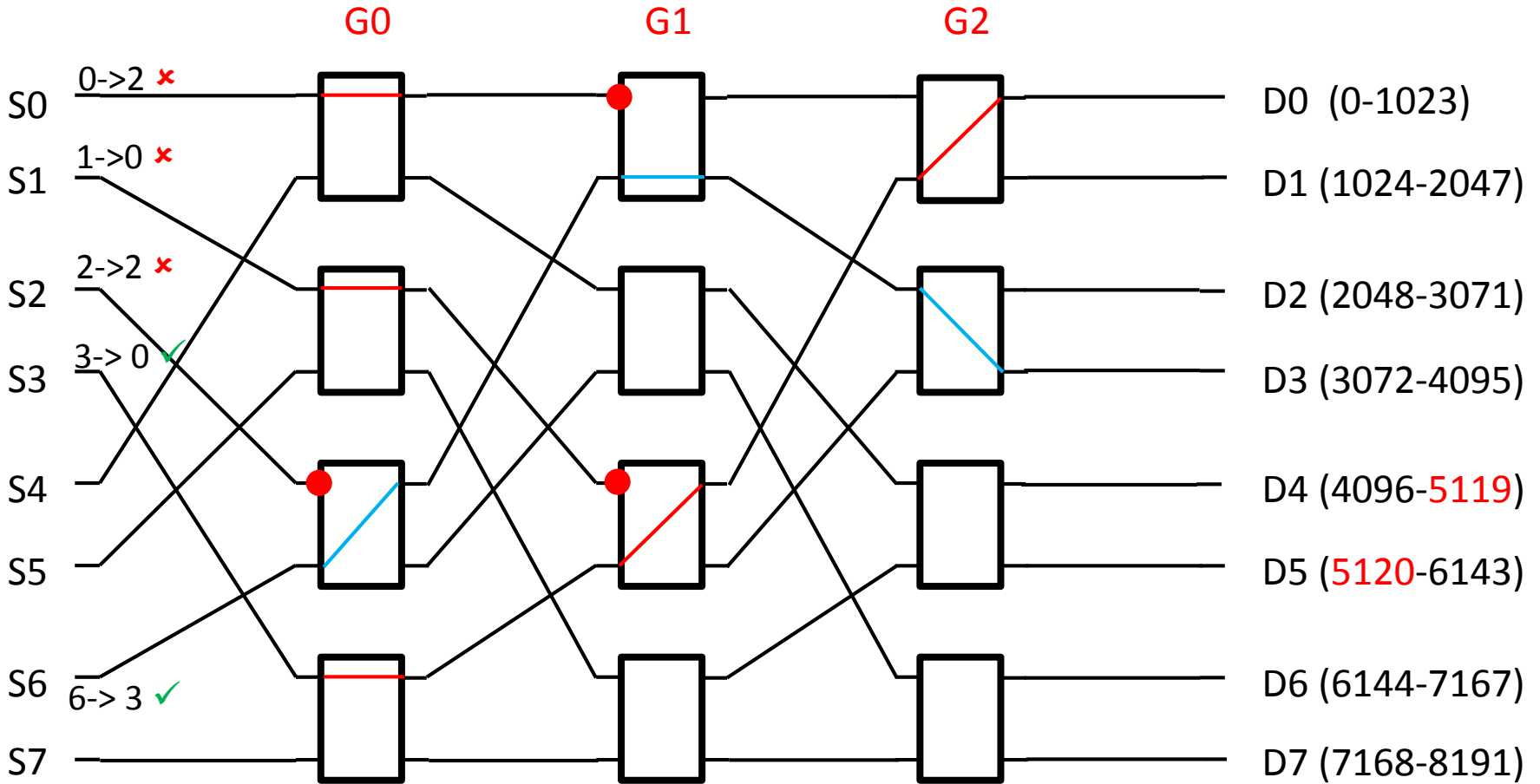
Trazamos primero los caminos que tienen prioridad (entrada 1 de cada conmutador) en color azul.

A continuación los que no tienen prioridad hasta donde son compatibles con los anteriores, en rojo.



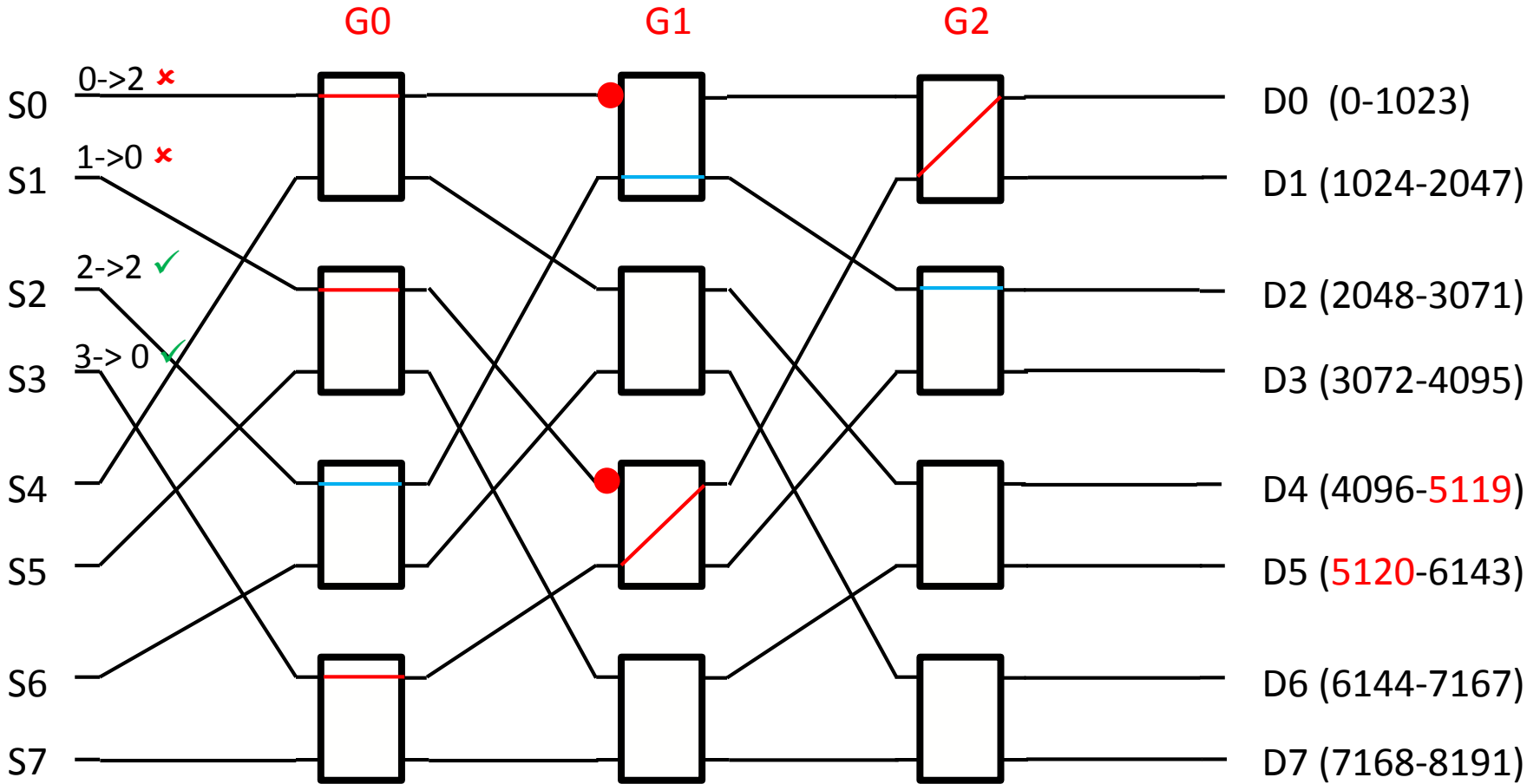
Problema 10

Ciclo de acceso 2



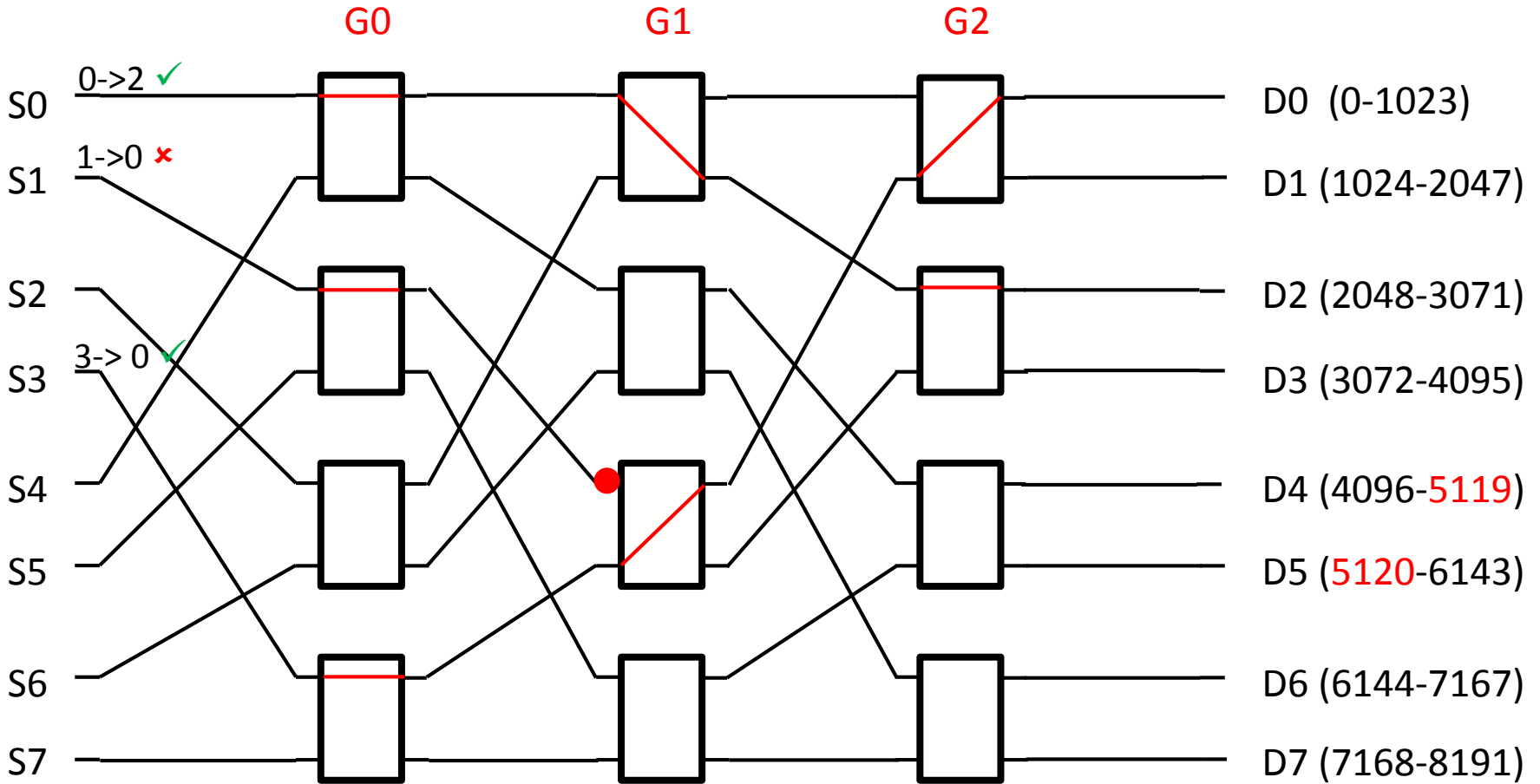
Problema 10

Ciclo de acceso 3



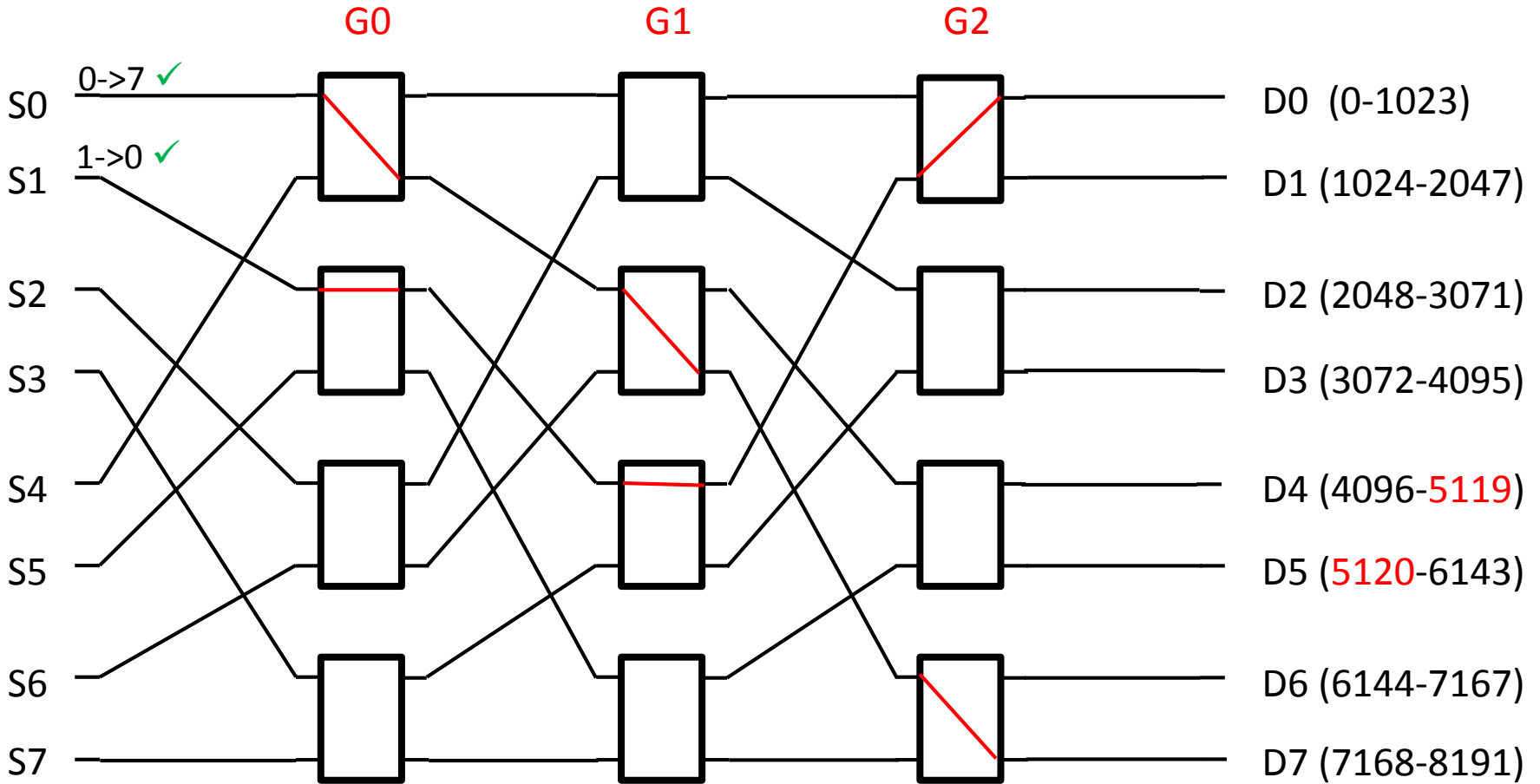
Problema 10

Ciclo de acceso 4



Problema 10

Ciclo de acceso 5



## Problema 10

$$a) \quad T_{RC} = 0,5 \text{ ns} \quad T_{mem} = 2 \text{ ns} \quad T_{com} = N^{\circ} \text{ de etapas} \times T_{RC} = 3 \times 0,5 = 1,5 \text{ ns}$$

$$T_{ciclo} = 2 \times T_{com} + T_{mem} = 2 \times 1,5 + 2 = 5 \text{ ns}$$

Tiempo total para cada procesador = (nº ciclos  $P_i$ ) x  $T_{ciclo}$

$$P0: T_{total} = 5 \times 5 = 25 \text{ ns}$$

$$P1: T_{total} = 7 \times 5 = 35 \text{ ns}$$

$$P2: T_{total} = 3 \times 5 = 15 \text{ ns}$$

.....

.....

Caso más lento P1: 35 ns



## Problema 10 (Con Crossbar. En accesos a mismo destino se anulan las operaciones procedentes de puertos inferiores)

Ciclo	P0	P1	P2	P3	P4	P5	P6	P7
Intento 1	<b>Direc:</b> 1025 <b>Conex:</b> 0->1	100 1->0	3000 2->2	0 3->0	5020 4->4	7800 5->7	4000 6->3	670 7->0
Exitos 1	✓	✗	✓	✗	✓	✓	✓	✓
Intento 2	<b>Direc:</b> 2048 <b>Conex:</b> 0->2	100 1->0		0 3->0			4001 6->3	
Exitos 2	✓	✗		✓			✓	
Intento 3	<b>Dir:</b> 8000 <b>Con:</b> 0->7	100 1->0		1 3->0				
Exitos 3	✓	✗		✓				
Intento 4		100 1->0		2 3->0				
Exitos 4		✗		✓				
Intento 5		100 1->0						
Exitos 5		✓						
Intento 6		2500 1->2						
Exitos 6		✓						
Intento 7		7500 1->7						
Exitos 7		✓						

## Problema 10

b) Con un Crossbar.

$$T_{\text{ciclo}} = 2 \times T_{\text{com}} + T_{\text{mem}} = 2 \times (4 \times 0,5) + 2 = 6 \text{ ns}$$

El Xbar tiene conectividad total, pero en un ciclo no puede haber varios procesadores accediendo a un mismo módulo de memoria. La transparencia anterior muestra el secuenciamiento de accesos, suponiendo que en caso de conflicto, la política de prioridades anula, igual que en el apartado a), las peticiones procedentes de las entradas con los números inferiores.

Por ejemplo, en el ciclo 1, P1, P3 y P7 intentan acceder al Módulo 0, pero solo P7 lo consigue.

Como muestra la tabla, con esta política de prioridades, hacen falta 7 ciclos. Con otras políticas el nº de ciclos puede variar (ver transp. siguiente). Luego:

$$T_{\text{total}} = 7 \times 6 = 42 \text{ ns} \quad \text{Speedup} = 35/42 = 0,83$$

Es decir, la velocidad no mejora, sino que empeora.

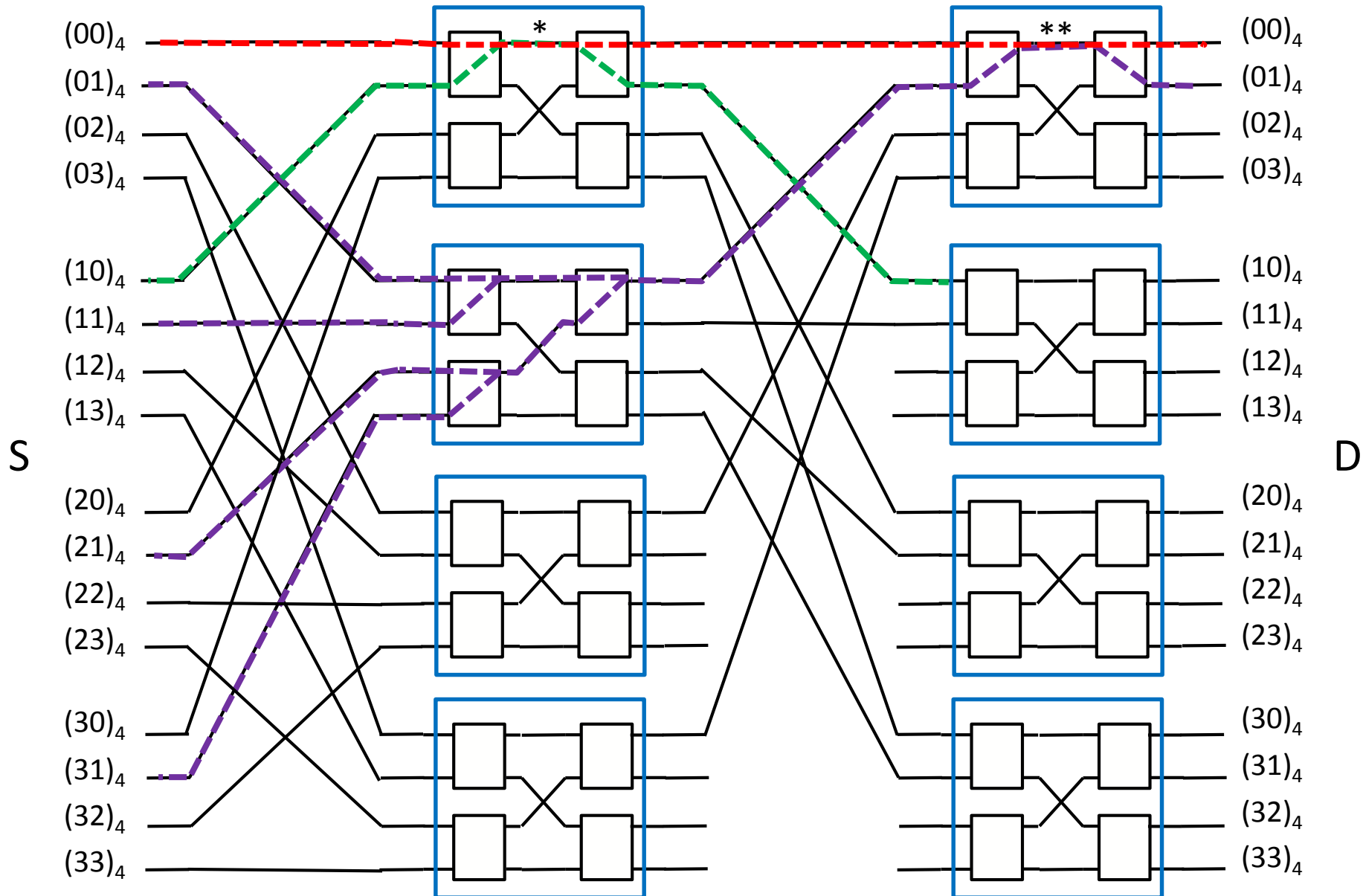
c) Posterga durante mucho tiempo los accesos de algunos procesadores. Vía de solución: alternar la política de prioridades.

## Problema 12

Línea roja. Las conexiones \* y \*\* quedan ocupadas al conectar  $0 \rightarrow 0$

Línea verde. Conexiones bloqueadas por el uso de \*: S4 con D4, D5, D6 y D7

Línea morada. Conexiones bloqueadas por el uso de \*\*: S1, S5, S9 y S13 con D1



## Problema 13

Red Omega de 16 entradas

Tamaño mensaje:  $n=128$  bits; Ancho de banda:  $B= 1\text{GB/s} = 8 \times 10^9$  bits/s

Retardo conmutador grado 2:  $\Delta_2 = 4 \times 10^{-9}$  s ; Retardo conmutador grado 4:  $\Delta_4 = 6 \times 10^{-9}$  s

Con conmutadores de grado 2 => Nº etapas (hops):  $h=4$

Con cut-through:  $T_{CT2} = h \Delta_2 + n/B = 4 \times 4 \times 10^{-9} \text{ s} + (128 \text{ bits} / 8 \times 10^9 \text{ bits/s}) = 32 \times 10^{-9} \text{ s} = 32 \text{ ns}$

Con store-and-forward:  $T_{SF2} = h ( \Delta_2 + n/B ) = 4 \times [ 4 \times 10^{-9} + (128 / 8 \times 10^9) ] = 80 \text{ ns}$

Con conmutadores de grado 4 => Nº etapas:  $h=2$

Con cut-through:  $T_{CT4} = h \Delta_4 + n/B = 2 \times 6 \times 10^{-9} + (128 / 8 \times 10^9) = 28 \text{ ns}$

Con store-and-forward:  $T_{SF4} = h ( \Delta_4 + n/B ) = 2 \times [ 6 \times 10^{-9} + (128 / 8 \times 10^9) ] = 44 \text{ ns}$

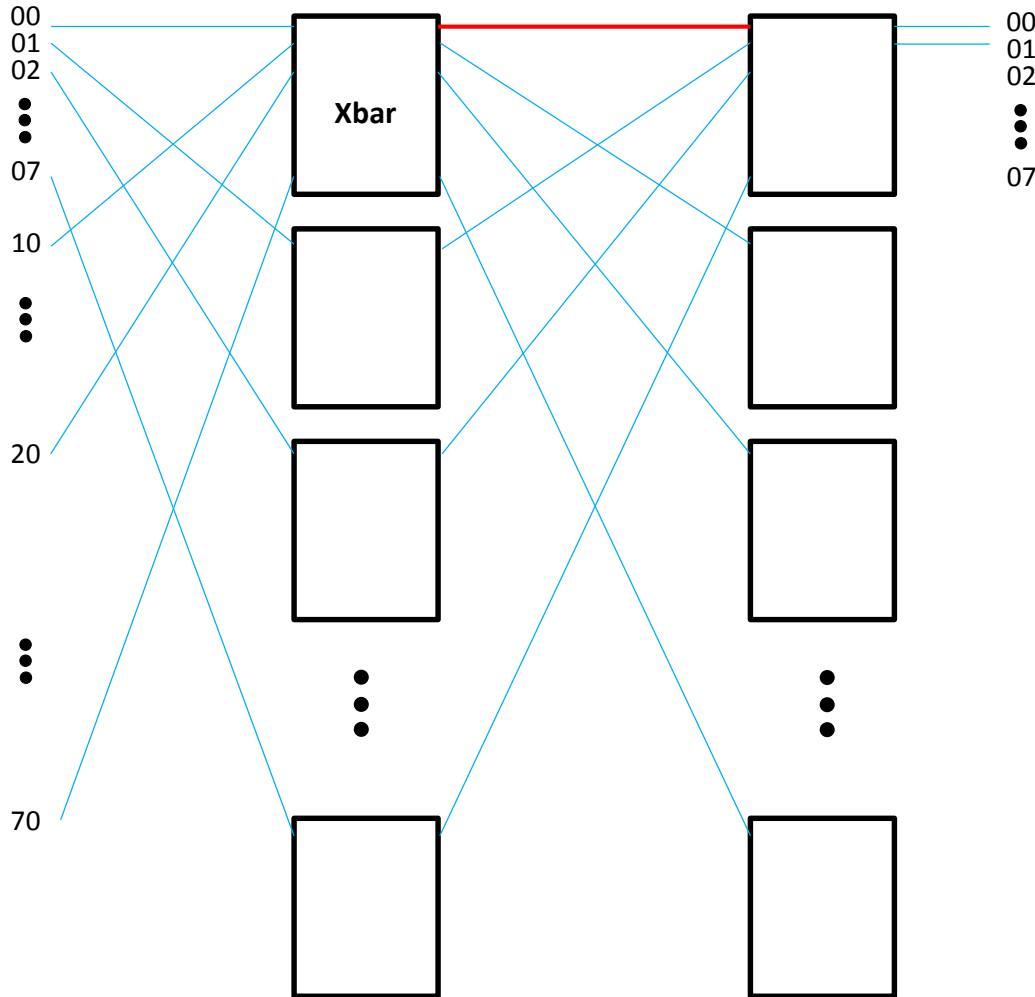
Análisis:

- La opción más rápida es CT4.
- Se observa que con CT el nº de etapas de la red no influye excesivamente.
- Sin embargo, con SF el nº de etapas es muy determinante.

# Problema 14

$N = 64$ ; Base  $k = 8$ ;  $N = k^n \rightarrow N^{\circ}$  etapas =  $n = 2$

Red Omega:  $C_0 = C_1 = \sigma$ ;  $C_2 = I$ ;  $\sigma(a,b) = (b,a)$  siendo  $a,b \in \{0..7\}$



Al conectar  $S_0 \rightarrow D_0$  queda bloqueada cq comunicación que atraviese el enlace rojo. Es decir, cq comunicación que conecte las fuentes:

$S_8, S_{16}, \dots, S_{56}$

con los destinos:

$D_1, D_1, \dots, D_7$