

Arquitectura de Computadores



TEMA 5

Multiprocesadores y redes de interconexión

DEPARTAMENTO DE
ARQUITECTURA DE **C**OMPUTADORES
Y **A**UTOMÁTICA

Curso 2020-2021

- ❑ Introducción
- ❑ Arquitecturas paralelas
 - o Memoria compartida
 - o Paso de mensajes
- ❑ Redes de interconexión
 - o Descripción estructural
 - o Topología de Red
 - Redes estáticas
 - Redes dinámicas
 - o Técnicas de conmutación
 - o Encaminamiento (routing)
- ❑ Bibliografía
 - o Apéndice F de Hennessy & Patterson, 5th ed., 2012
 - o Culler & Singh, "Parallel Computer Architecture: A Hardware/Software Approach", 1999
 - Cap1, Sec 3; Cap 10, Sec. 1-3 y 5

- ❑ Una definición clásica de computador paralelo
 - o Colección de elementos de procesamiento que cooperan y se comunican para resolver problemas grandes (cálculo, almacenamiento) "rápidamente". (Almansi and Gottlieb 1989)
- ❑ Objetivo
 - o Rendimiento / Eficiencia
 - o Replicación (Tolerancia a fallos)
- ❑ Esta definición da lugar a muchas preguntas:
 - o ¿Qué potencia tienen los elementos de proceso?
 - o ¿Cuánta memoria?
 - o ¿El número puede crecer de una manera sencilla: es escalable el sistema?
 - o ¿Cómo se comunican y cooperan dichos elementos?
 - o ¿Cómo se transmiten los datos entre los procesadores?
 - o ¿Qué tipo de interconexión conecta los distintos procesadores?
 - o ¿Qué abstracción (primitivas) proporciona el hardware /software al programador?

❑ Creciente interés por los multiprocesadores

"We are dedicating all of our future product development to multicore designs. We believe this is a key inflection point for the industry."

Intel President Paul Otellini,
describing Intel's future direction at the
Intel Developer Forum in 2005

❑ Algunos factores determinantes

- o Agotamiento de ILP
 - Consumo de energía y coste crecen más rápido que rendimiento
- o Creciente interés en servidores de altas prestaciones
 - Cloud computing, Software-as-a-Service (SaaS)
- o Aumento de las aplicaciones intensivas en datos (Big Data)
 - Geociencia, Biología computacional, Deep learning...
 - Disponibilidad de enormes cantidades de datos vía Internet
- o Mejora del conocimiento sobre cómo usar eficazmente los multiprocesadores
 - Entornos de grandes BD, multitud de accesos simultáneos, problemas de cálculo científico
- o Aprovechamiento de los diseños vía replicación del procesador

Arquitecturas Paralelas

- ❑ Modelo de Programación: concepto de máquina que utiliza el programador
 - o ¿Cómo se comparte la información? ¿Cómo transferir la información entre distintas partes de un programa que se ejecuta en paralelo?
 - o ¿Cómo se lleva a cabo la coordinación de actividades? ¿Qué primitivas de sincronización hay para coordinar actividades?
 - o Típicamente: el modelo de programación esta incorporado a
 - Lenguaje de Programación (compilación)
 - Librería

Son los encargados de realizar la correspondencia

 - Construcciones del lenguaje o llamadas de la librería ⇔ Primitivas disponibles a nivel usuario (Arquitectura de Comunicaciones)
 - Algunas operaciones las realiza una combinación Hw/SO
- ❑ Memoria compartida vs. paso de mensajes

Arquitecturas de Memoria Compartida

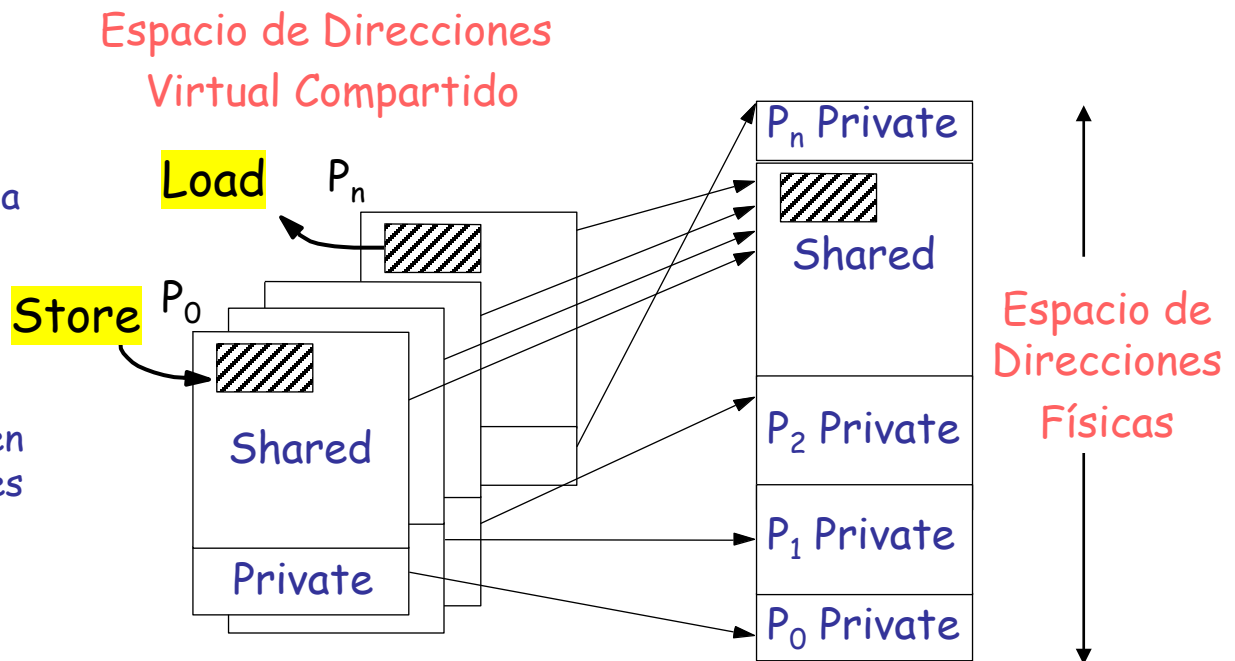
❑ Un espacio de direcciones compartido

o Referente: multiprogramación en monoprocesadores

- Objetivo: Productividad (entornos time-sharing)
- Los procesos pueden configurarse de modo que parte de su espacio de direcciones sea compartido:
 - Comunicaciones: implícitas (operaciones LOAD/STORE)
 - Sincronizaciones: SO, Operaciones Atómicas

Múltiples procesos pueden tener una región de memoria compartida proyectada sobre su espacio de direcciones virtual

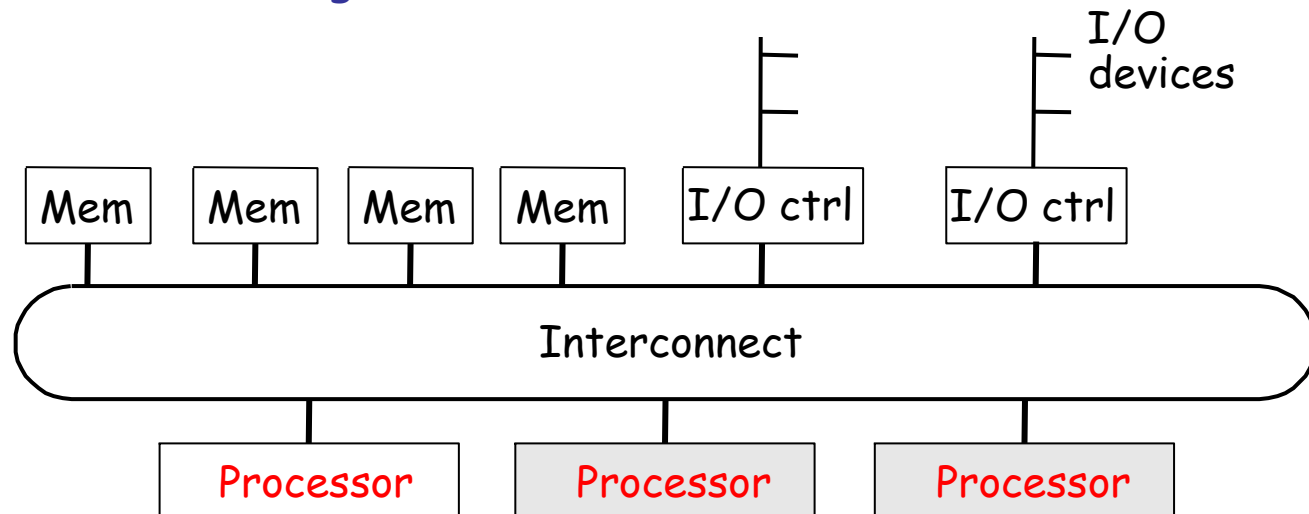
Las escrituras de variables residentes en esta región son visibles para el resto de los procesos.



Arquitecturas de Memoria Compartida

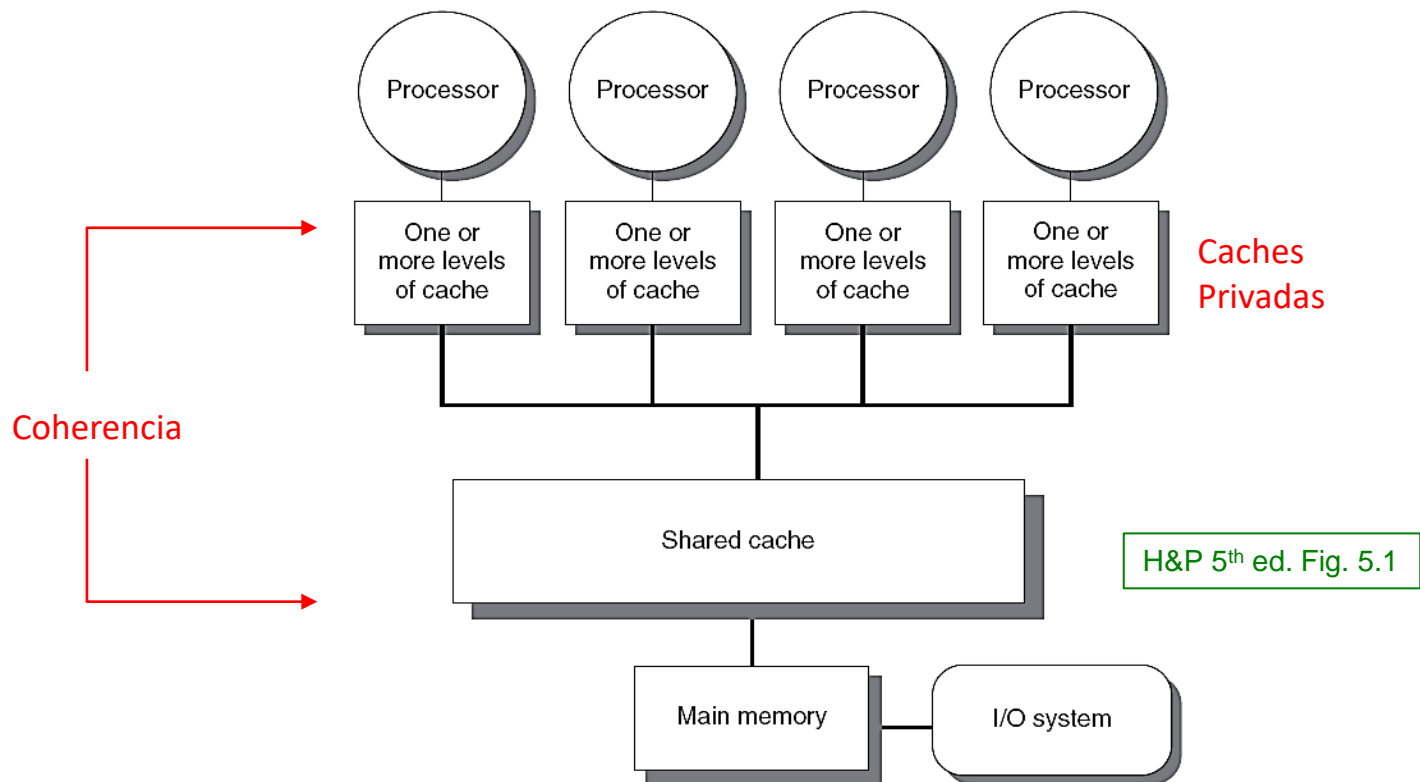
❑ Multiprocesador de memoria compartida:

- o Extensión natural de un sistema en el que diferentes procesos pueden compartir memoria → añadir uno o varios procesadores
- o Red de interconexión vs. N° de procesadores
 - Pocos procesadores: red sencilla (p. ej. Bus)
 - Muchos procesadores: red más compleja (p. ej. red multietapa)
- o Escalabilidad: ¿Cuántos procesadores se pueden conectar sin que la red se convierta en un cuello de botella que degrade el rendimiento global?



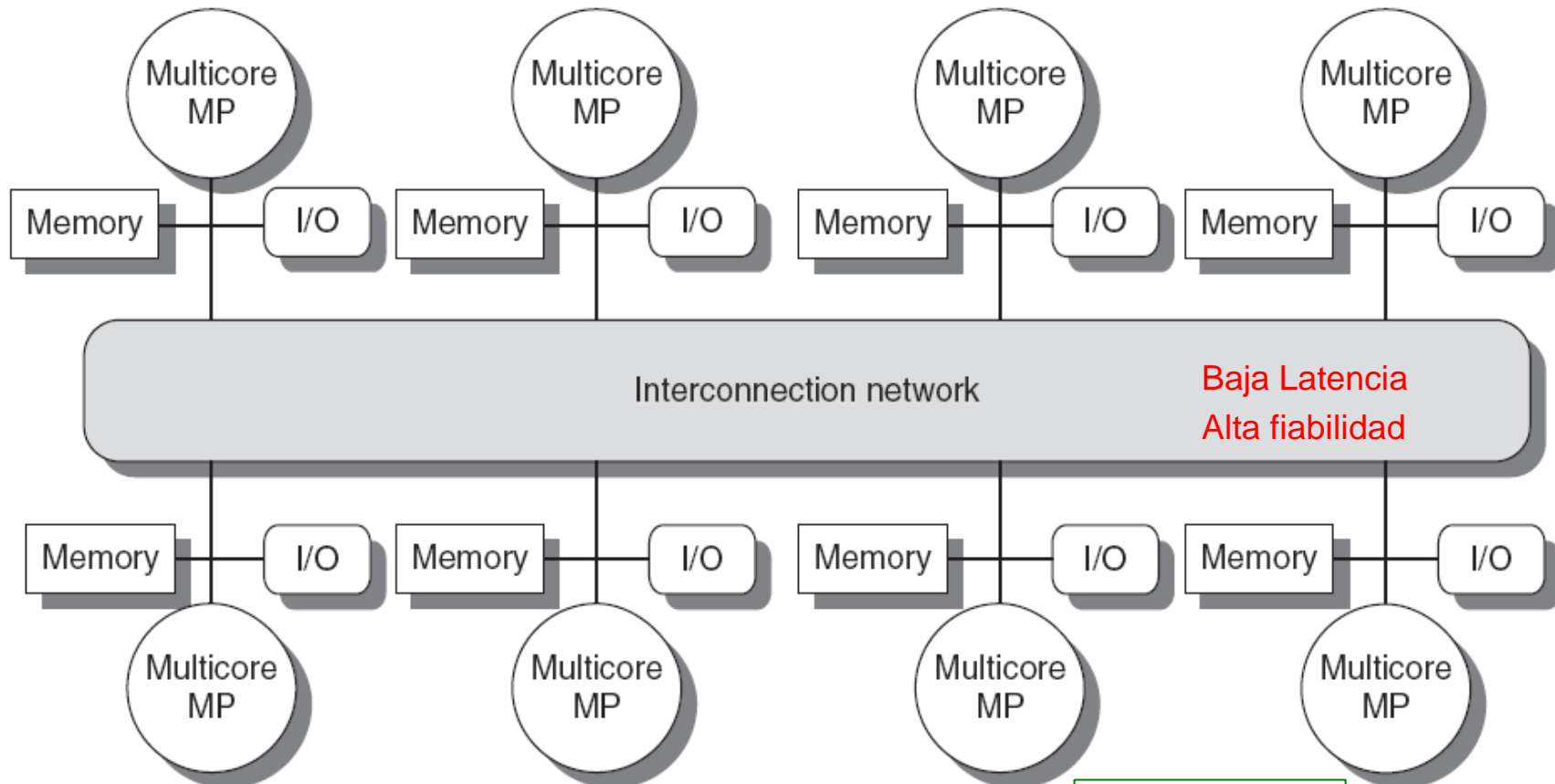
Arquitecturas de Memoria Compartida

- ❑ Diseño para un nº reducido de procesadores
 - o Symmetric (Shared-Memory) Multiprocessor (**SMP**)
 - o Memoria: **centralizada** con tiempo de acceso uniforme ("**UMA**") Interconexión vía bus común, E/S compartida
 - o Anchura de banda limitada. Limite en el numero máximo de procesadores
 - o Caches son cruciales: **Coherencia**
 - o Posibilidad de usar otras redes: crossbar, multietapa
 - o Ejemplos: Sun Enterprise 6000, SGI Challenge, Intel SystemPro, Multicores.....



Arquitecturas de Memoria Compartida

- ❑ Diseño para un número elevado de procesadores
 - o Memoria: **Distribuida** con tiempo de acceso no uniforme ("**NUMA**")
 - o Interconexión escalable.
 - o Anchura de banda crece con el numero de EP
 - o Ejemplos: Cray T3E, SGI Origin ...

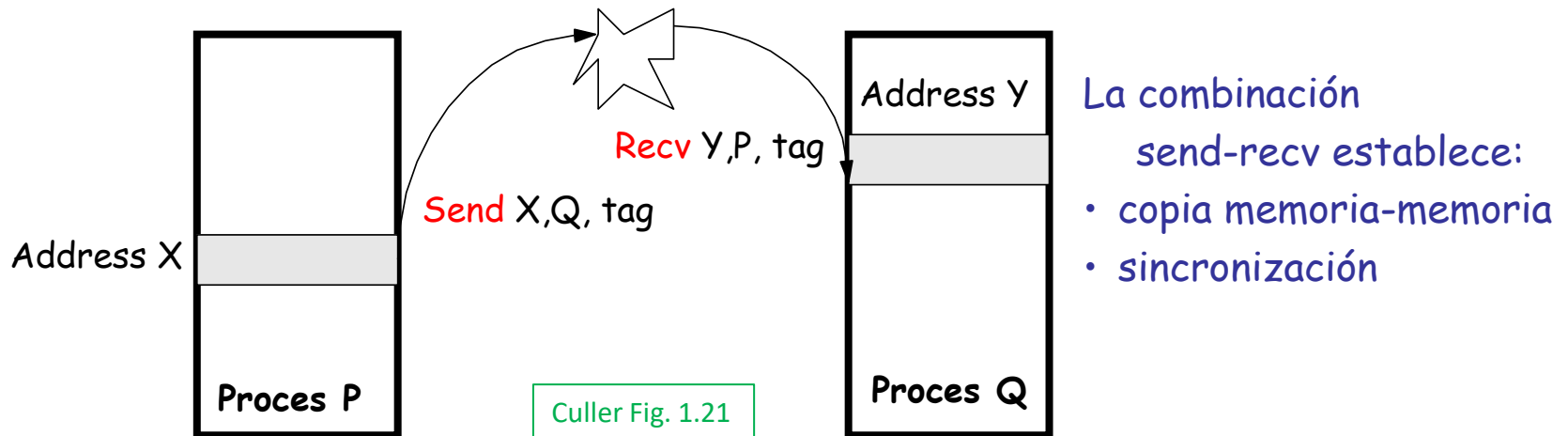


Arquitecturas de Paso de Mensajes

❑ Elemento de Proceso: Computador Completo (microproces, mem, E/S)

o Modelo de Programación

- Acceso directo al espacio de direcciones privado (memoria local)
- Comunicación explícita mediante operaciones I/O (llamadas al SO: send/recv)



❑ Sobrecarga:

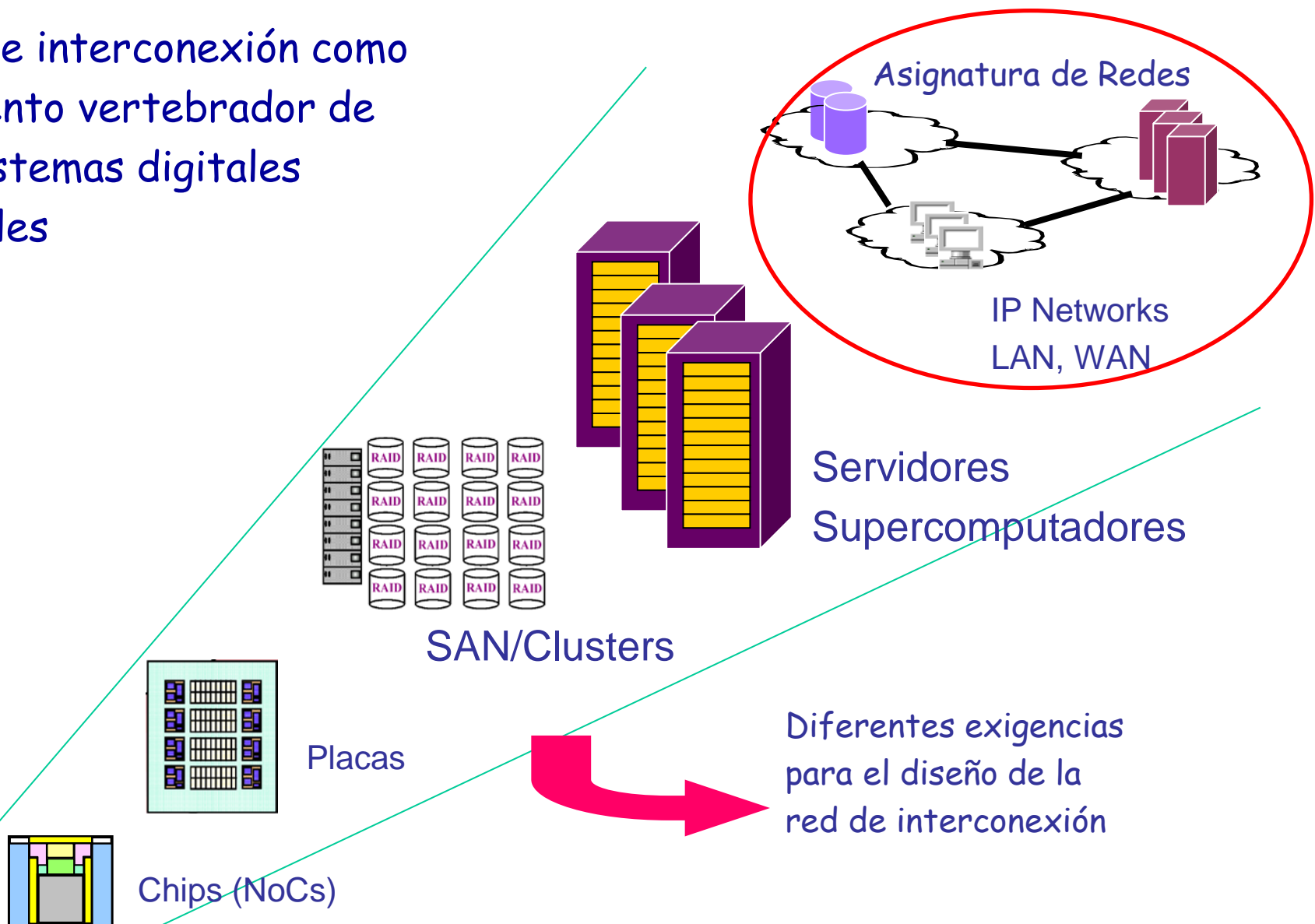
- o Construir la cabecera del mensaje; copiar los datos en el buffer de la red; enviar el mensaje; copiar los datos en el buffer receptor; copiar los datos **del espacio S.O. al espacio usuario**

❑ Diagrama de Bloques Semejante a NUMA

- o Diferencia: Las comunicaciones están **integradas en el sistema de E/S** en lugar de en el sistema de memoria

Redes de interconexión: Introducción

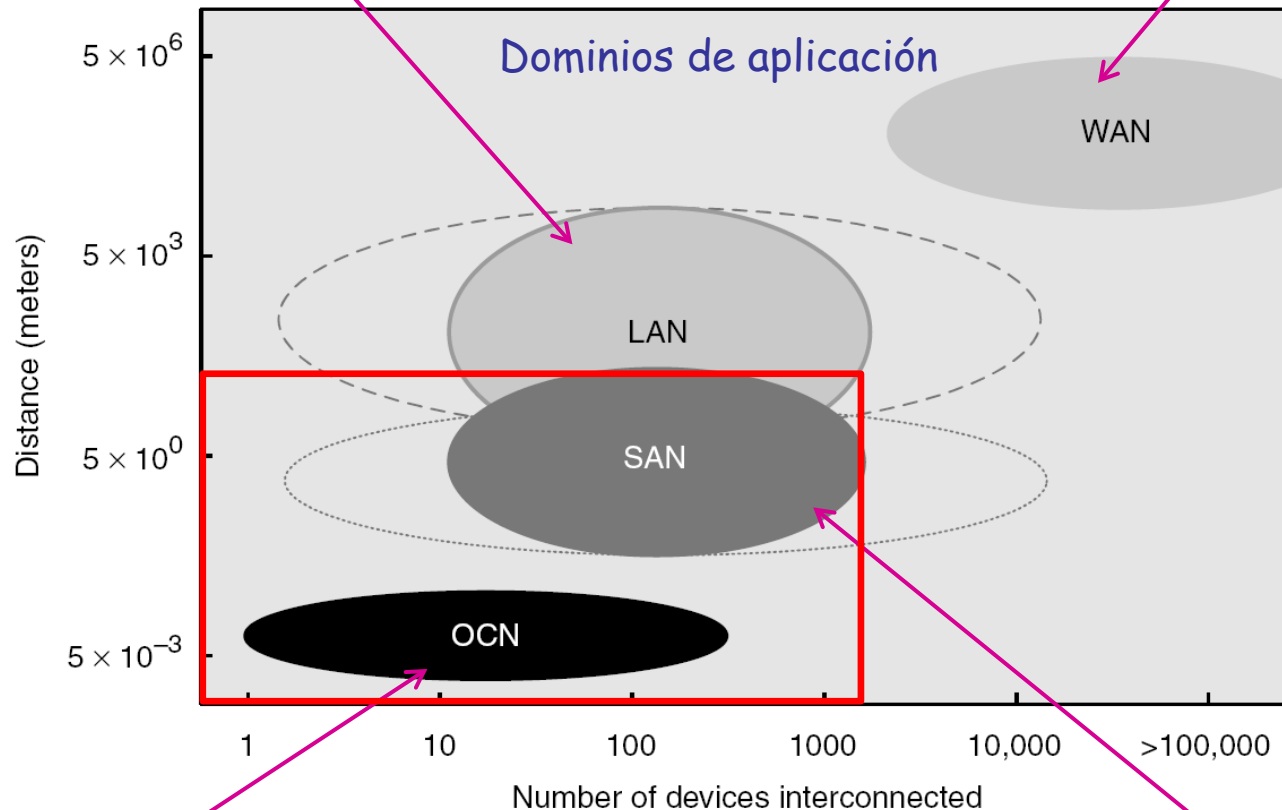
Red de interconexión como elemento vertebrador de los sistemas digitales actuales



Redes de interconexión: Introducción

- Local Area Network (LAN). Ej: Ethernet (hasta 10 Gbps sobre una distancia de 40 Km). Tb Clusters de PCs

- Wide Area Network (WAN). Conexión a escala global. Ej: ATM



H&P 5th o 6th ed.
Fig. F.2

- Network-on-Chip (NoC, OCN). Ej: AMBA de ARM, SCCC de Intel, TILE-GX de Tiler, multicores Intel (milímetros)

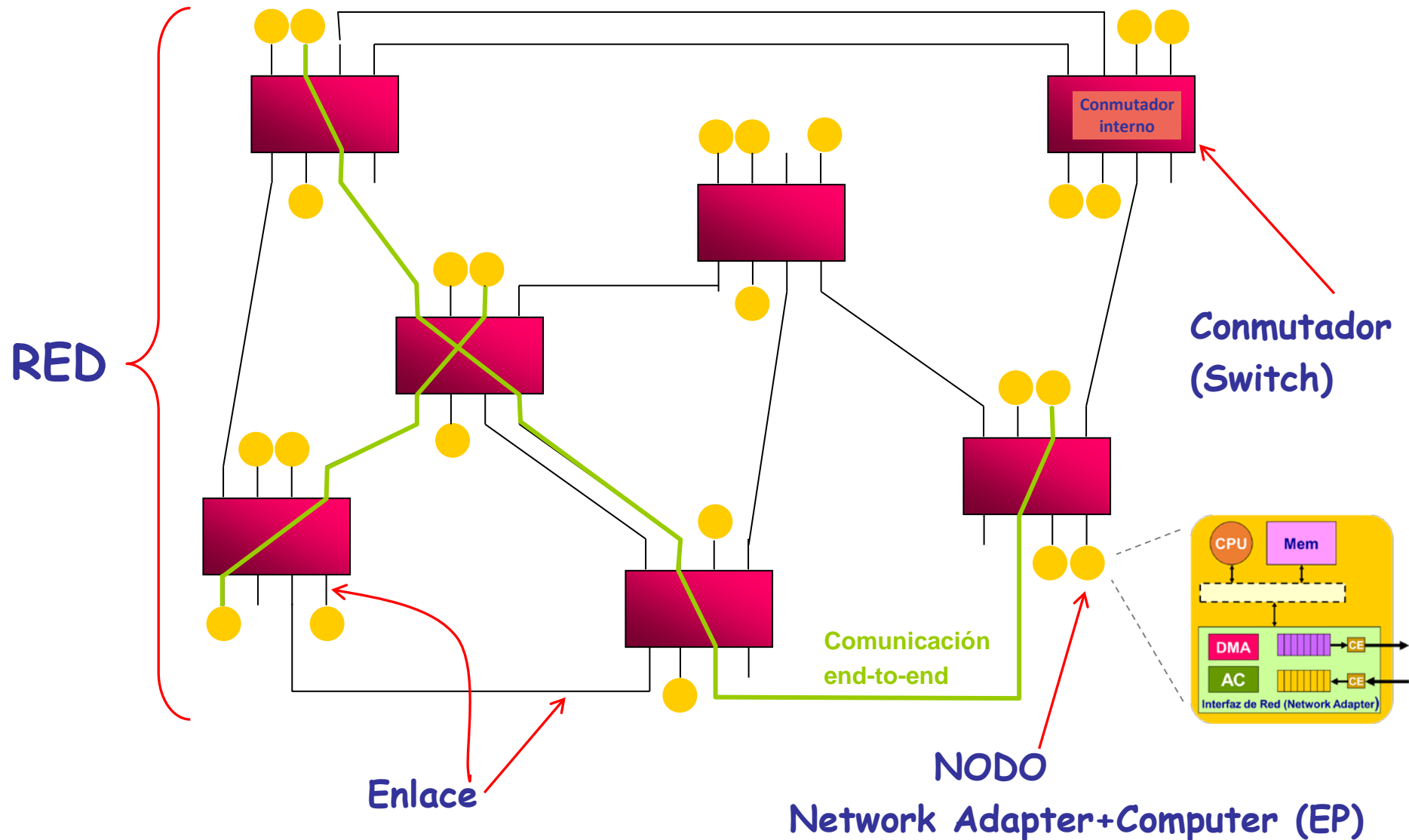
- System/Storage Area Network (SAN). Comunicación proc-mem-discos. Ej: Infiniband (hasta 120 Gbps sobre una distancia de 300 m)

- ❑ Factores de diseño: El principal factor en computadores paralelos son las prestaciones de la red:
 - o Latencia
 - o Productividad (ancho de banda)

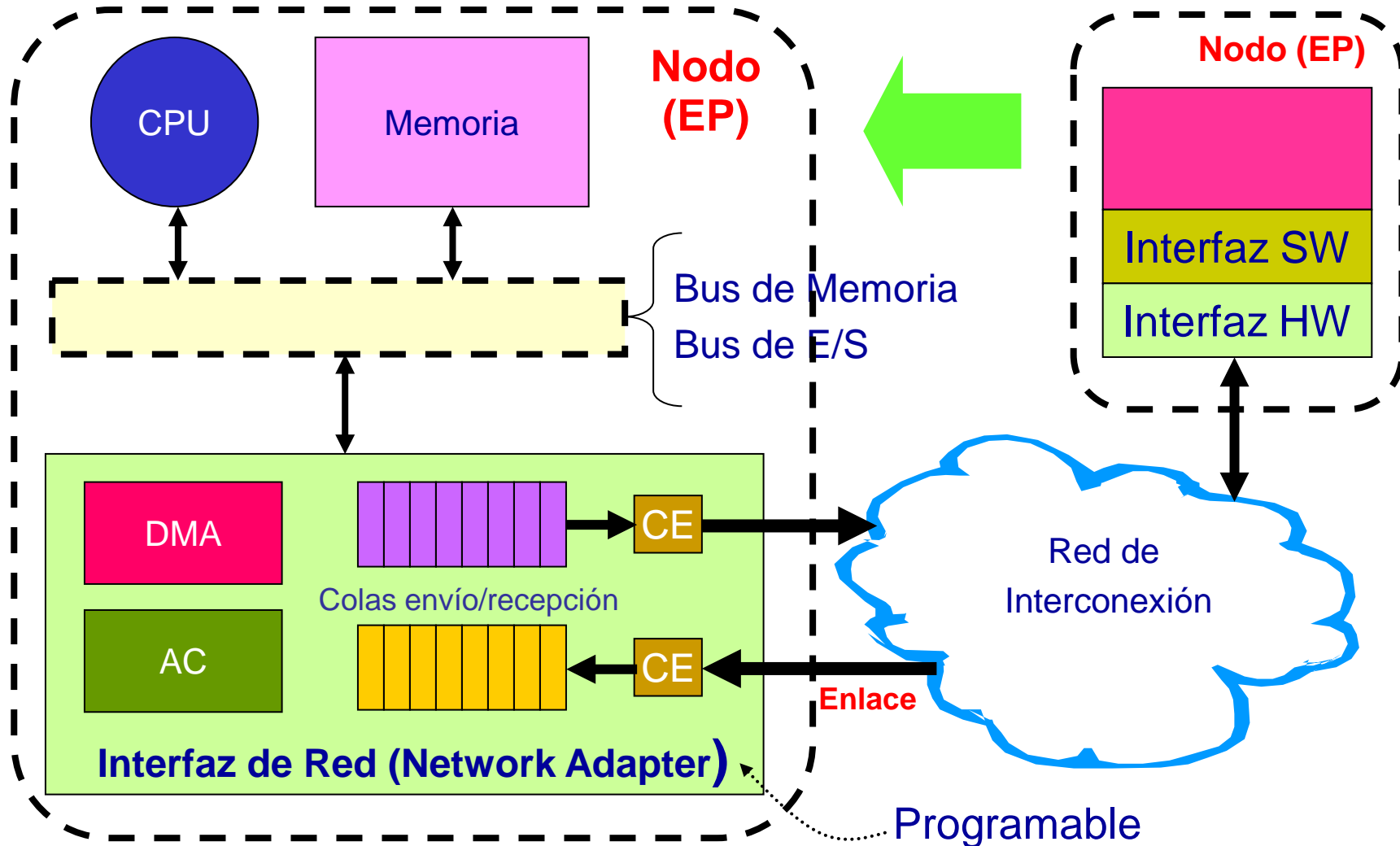
- ❑ Otros factores de diseño:
 - o Escalabilidad
 - o Fiabilidad
 - o Expansión incremental
 - o Particionado y seguridad
 - o Simplicidad y bajo coste
 - o Distancia de expansión
 - o Consumo/disipación de energía

Descripción estructural: Elementos

Lectura: Culler 10.1.1



Esquema básico de un interfaz de red



Descripción estructural: Interfaz de Red

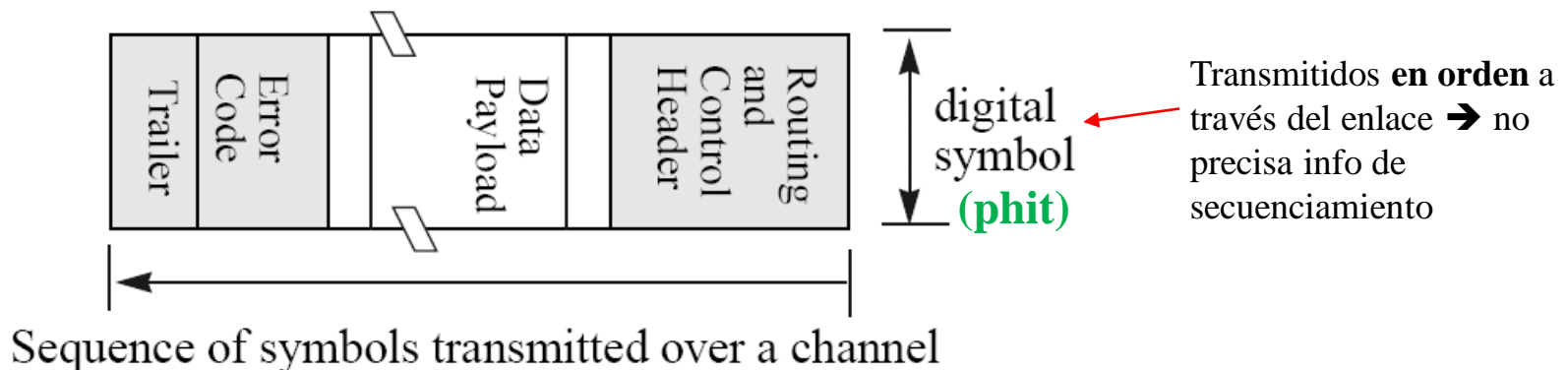
Lectura: Culler p. 680

❑ Mensaje

- o Unidad lógica de comunicación
- o Es la única unidad vista a nivel de aplicación.

❑ Paquete

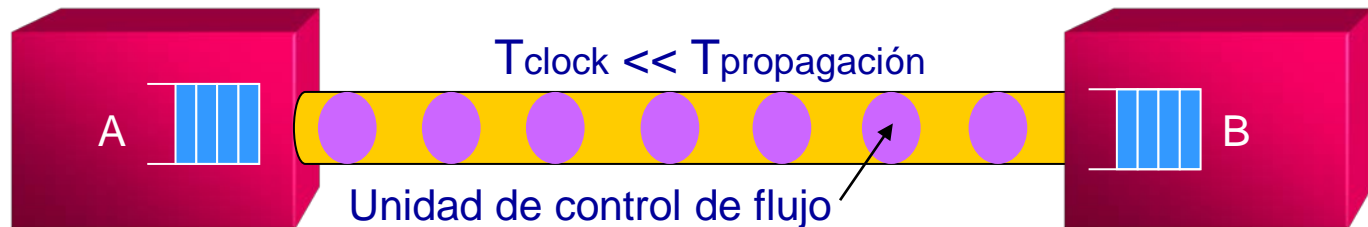
- o Unidad de transferencia entre interfaces de red (end-to-end)
- o También es la menor unidad que contiene información de encaminamiento.
- o Mecanismos de abstracción:
 - **Encapsulado:** añadir cabecera y cola a info a transmitir. P.ej. acceso a una línea de cache en un EP remoto
 - La memoria ignora los detalles de la red.
 - La red ignora la estructura de la cache
 - **Fragmentación:** un bloque muy grande de información, no suele transmitirse como un solo paquete → descomponer y encapsular trozos



Descripción estructural: Enlace

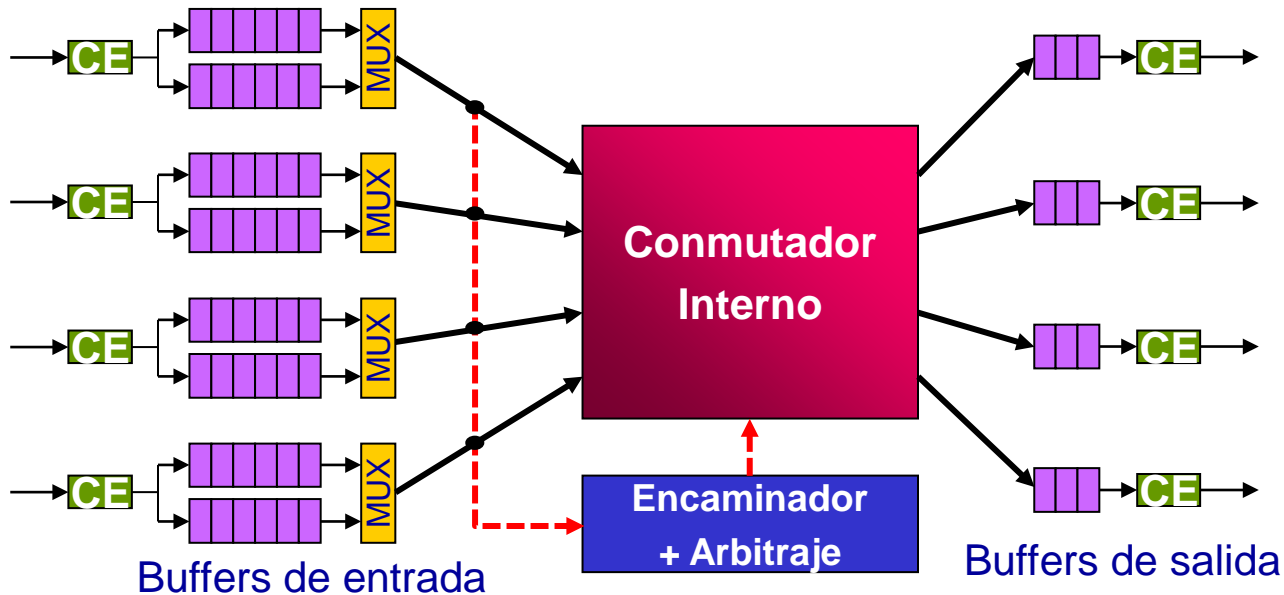
❑ Características de un enlace

- o Ancho: w
- o Velocidad de señal (frecuencia): $f = 1/\tau$
- o Ancho de banda: $B = wf$ (si w en bits y f en $s^{-1} \rightarrow AB$ en bits/s)
- o Unidad de transferencia por ciclo: **Phit (physical unit)** = w
- o Unidad de control de flujo: **Flit (flow-control unit)**
- o "Longitud"
 - Corto: un solo phit en el enlace
 - Largo: varios phit en el enlace (segmentado)



Esquema básico de un conmutador /switch/router

Lectura: Culler 10.2.2



❑ Características de un conmutador

- o Grado conmutador/nodo
 - Número de canales de entrada (salida)
- o Implementación conmutador interno: permutaciones E/S, broadcast...
- o Buffers de entrada/salida
- o Lógica de control: Arbitraje. Encaminamiento

Descripción estructural: Red

❑ Red de interconexión

- o Grafo de $V = \{\text{conmutadores y nodos}\}$ conectada por canales de comunicación $C \subseteq V \times V$

❑ Ruta

- o La secuencia de conmutadores/nodos y canales seguidos por un mensaje
 - Pensar en calles e intersecciones

❑ Características de la red

o Topología

- Estructura de la interconexión física de la red
 - Definido por su función de interconexión
 - Regular vs irregular
 - Clasificación: medio compartido, estáticas, dinámicas

o Conmutación

- Cómo atraviesan su ruta los datos de un mensaje
- Conmutación de circuitos vs. conmutación de paquetes

o Encaminamiento

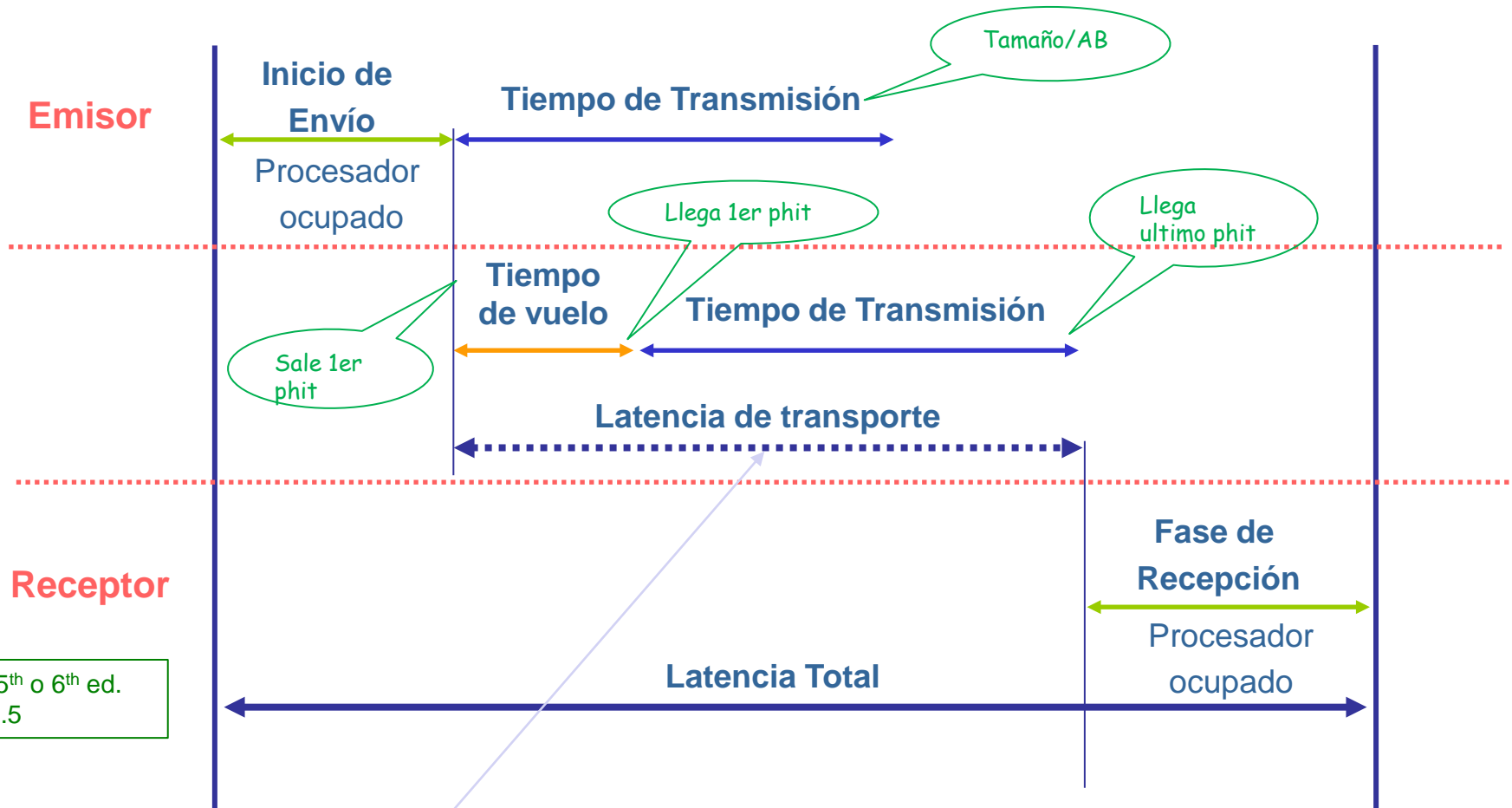
- Qué rutas pueden seguir los mensajes a través del grafo de la red
 - Encaminamiento determinista vs adaptativo

o Control de flujo (materias de Redes)

- En qué momento el mensaje, o porciones de éste, se mueven a lo largo de la ruta

Rendimiento

❑ Latencia total extremo-a-extremo (latencia y AB efectivo)



H&P 5th o 6th ed.
Fig. F.5

Latencia de Transporte: Tiempo para que el **paquete** pase a través de la red asumiendo que **no hay contención** y **sin incluir las sobrecargas** de inyectar el mensaje en la red ni de salir cuando llega al destino.

Tiempo de vuelo:

- **NoC:** ~ ps
- **SAN:** ~ ns
- **LAN:** ~ μ s
- **WAN:** ~ ms

❑ Definiciones y propiedades

o Red de interconexión

- Grafo de $V = \{\text{conmutadores y nodos}\}$ conectada por canales de comunicación $C \subseteq V \times V$

o Grado de un nodo/conmutador (d).

- Número de canales de E/S del nodo/conmutador.

o Tamaño de la red.

- No. de EP conectados.

o Número de conmutadores.

o Diámetro de la red (D).

- Máxima longitud de los caminos óptimos entre pares de EP
 - Si la topología importa, mejor cuanto D más pequeño

❑ Definiciones y propiedades

o Simetría

- Visión de la red por parte de los nodos
- Patrón de tráfico uniforme: produce una carga uniforme de los canales de la red si ésta es simétrica

o Homogeneidad

- Igualdad en las características de los nodos

o Regularidad

- Patrón de construcción/organización de la red
- Los nodos tienen el mismo grado

o Ancho de bisección

- nº canales a cortar para dividir la red en dos mitades iguales
- Ancho de banda de bisección: volumen de tráfico a través de esos canales

Clasificación topológica

❑ Tipos de clasificaciones

- o Medio compartido vs. Punto-a-Punto
- o Regulares vs. Irregulares
- o Estáticas/Directas vs. Dinámicas/Indirectas

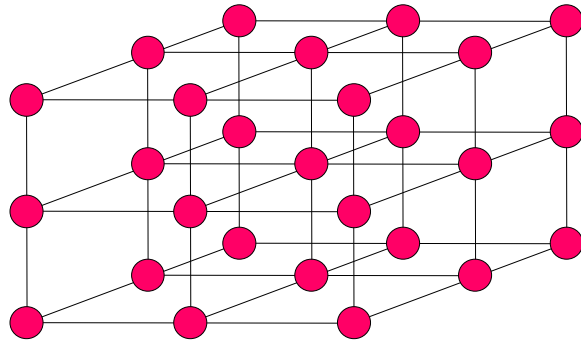
❑ Habitualmente (Duato, Yalamanchili, Ni)

- o Redes de medio compartido
 - Económicas, pero poco escalables
 - Bus, LAN (ej. Ethernet, etc)

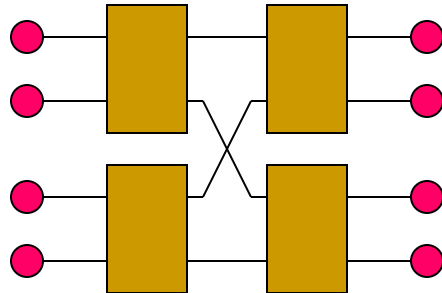
Punto a punto

- o Redes estáticas/directas
 - Malla, Toro, Hipercubo, etc
- o Redes dinámicas/indirectas
 - Crossbar
 - Redes multietapa (MIN)

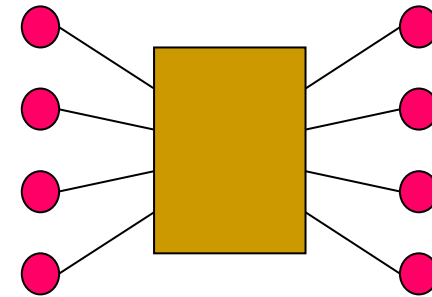
Clasificación topológica



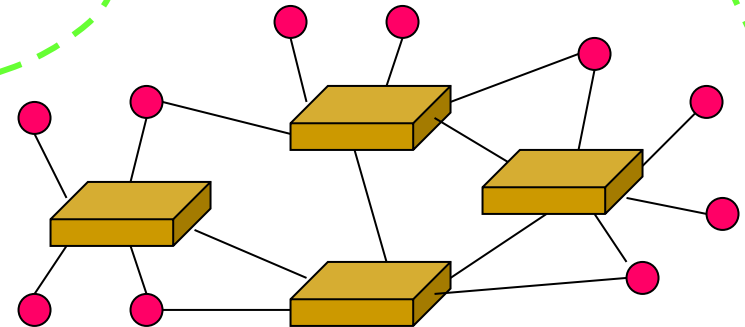
Red estática/directa
con topología regular



Multietapa



Crossbar



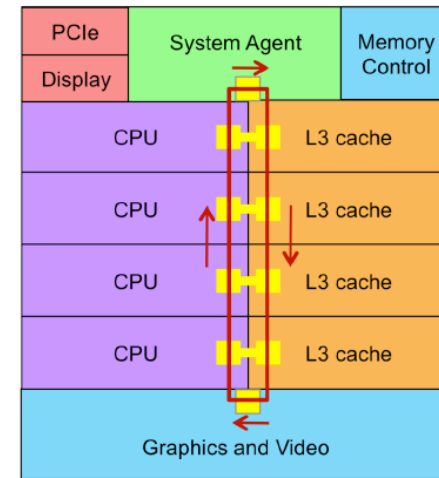
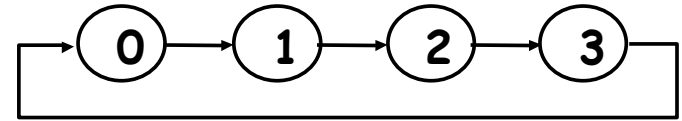
Red irregular

Redes dinámicas/indirectas

❑ Anillo unidireccional, p nodos

$$F(i) = (i+1) \bmod p, 0 \leq i < p$$

- Grado: $d = 1$ (link por nodo)
- Diámetro: $D = p - 1$ (max long)
- Simetría = Si



Ejemplo: Sandy Bridge NoC (2010):

Anillo unidireccional conecta CPUs, caches y otros módulos.

+

Conex directa CPU $_i$ -L3 $_i$

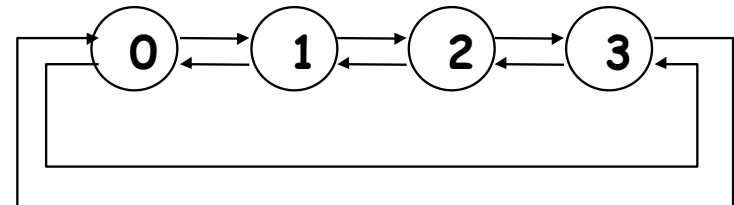
Salto (hops) para conectar cq CPU con cq j : entre 0 y 3

❑ Anillo bidireccional, p nodos

$$F_{+1}(i) = (i+1) \bmod p,$$

$$F_{-1}(i) = (i+p-1) \bmod p,$$

- $d = 2$, $D = p/2$
- Simetría = Si
- Ejemplos: FDDI, SCI, KSR1, Skylake



□ Malla 2D, $p = r \times r$ nodos

$F_{+x}(i) = (i+1)$ (existe si $i \bmod r \neq r-1$)

$F_{-x}(i) = (i-1)$ (existe si $i \bmod r \neq 0$)

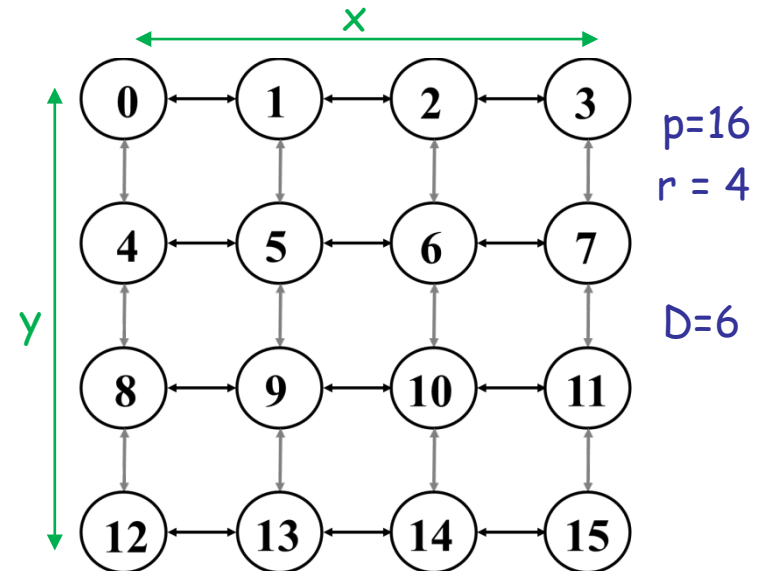
$F_{+y}(i) = (i+r)$ (existe si $i < p - r$)

$F_{-y}(i) = (i-r)$ (existe si $i \geq r$)

donde $0 \leq i \leq p-1$, $p = r \times r$

$d = 4$, $D = 2(r-1)$

o Simetría = No



□ Toro 2D, $p = r \times r$ nodos

$F_{+x}(i) = (i+1) \bmod r + (i/r) r$

$F_{-x}(i) = (i-1) \bmod r + (i/r) r$

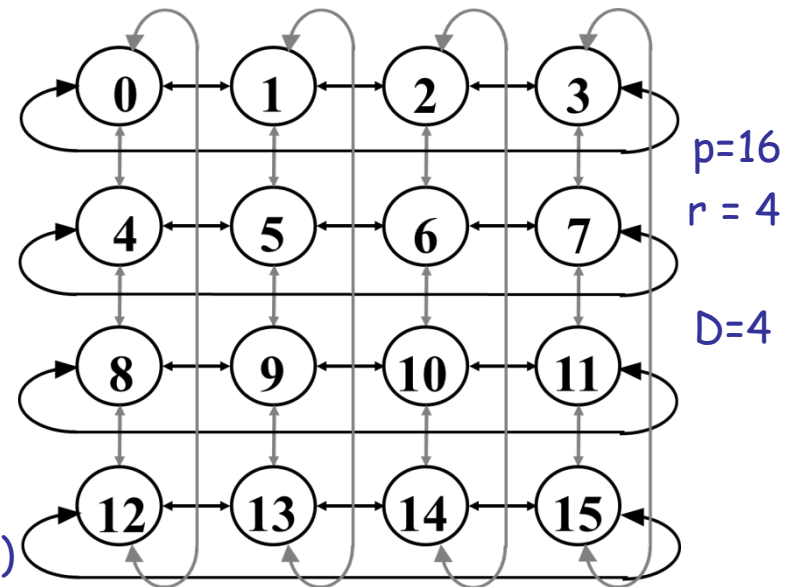
$F_{+y}(i) = (i+r) \bmod p$

$F_{-y}(i) = (i + p - r) \bmod p$

donde $0 \leq i \leq p-1$,

$d = 4$, $D = 2(r/2) \leftarrow$ división entera

o Simetría = Sí



Ejemplos: Multicores actuales (Skylake SP, malla)

❑ Hipercubo binario, $p = 2^n$ nodos

Hipercubo binario de orden $n = \log_2(p)$ (o n-cubo binario)

Cada nodo tiene una Dir. expresada mediante n dígitos binarios

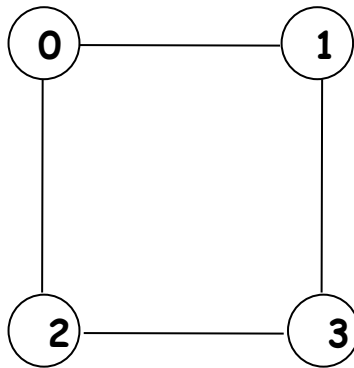
n funciones de interconexión, una por cada dimensión

$$F_i(h_{n-1}h_{n-2} \dots h_i \dots h_1 h_0) = h_{n-1}h_{n-2} \dots \bar{h}_i \dots h_1 h_0$$

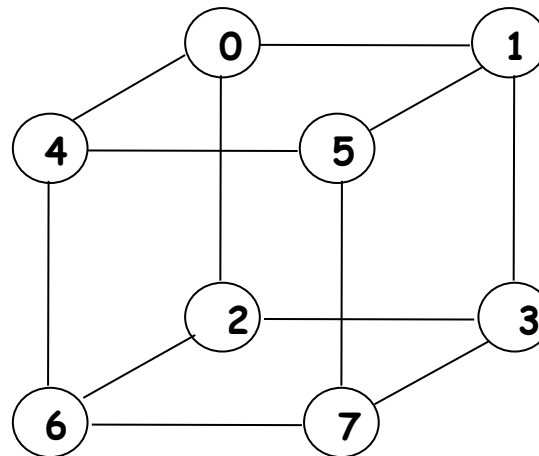
o $d = n, D = n$

o Simetría = Sí

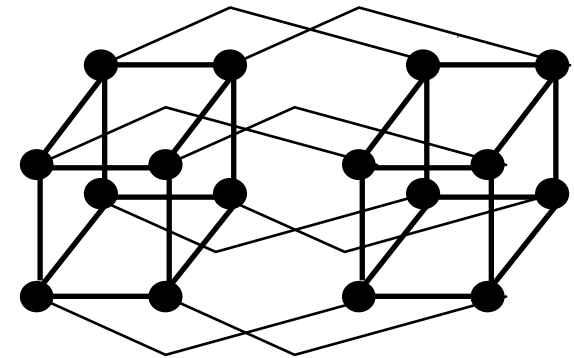
o Ejemplo: Origin 2000/3000



2-cubo $d=2, D=2$



3-cubo $d=3, D=3$



4-cubo $d=4, D=4$

❑ Idea extensible a n-cubos k-arios

□ n-cubo k-ario

- Consta de k^n nodos; cada nodo conectado con $2n$ vecinos
- Cada nodo tiene una Dir. expresada mediante n dígitos en base k
- Funciones de interconexión:

- $2n$ funciones de interconexión por nodo, 2 por cada dimensión

$$F_{i+}(h_{n-1}h_{n-2} \dots h_i \dots h_1 h_0) = h_{n-1}h_{n-2} \dots (h_i+1) \bmod k \dots h_1 h_0$$

$$F_{i-}(h_{n-1}h_{n-2} \dots h_i \dots h_1 h_0) = h_{n-1}h_{n-2} \dots (h_i+k-1) \bmod k \dots h_1 h_0$$

□ Caso particular: Toro 2D 4x4 como 2-cubo 4-ario

Cada nodo: dir con $n=2$ dígitos en base $k=4$

Funciones de interconexión: $2n=4$ por nodo

$$F_{1+}(p,q) = (p+1 \bmod 4, q) \quad F_{0+}(p,q) = (p, q+1 \bmod 4)$$

$$F_{1-}(p,q) = (p+4-1 \bmod 4, q) \quad F_{0-}(p,q) = (p, q+4-1 \bmod 4)$$

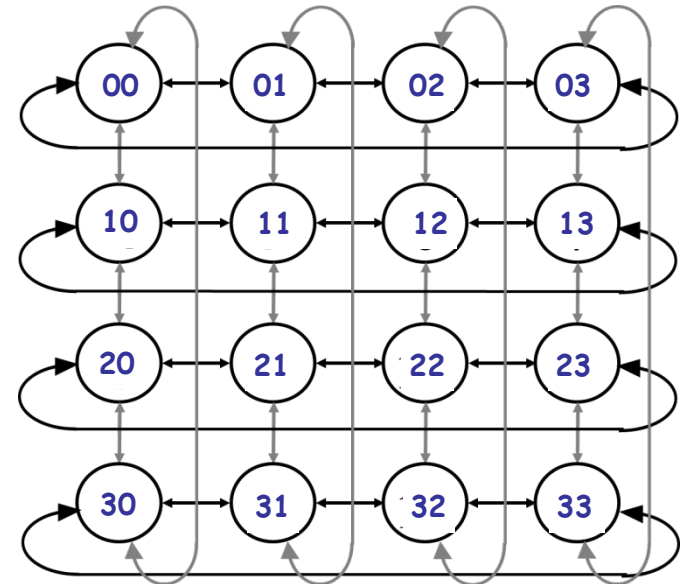
Ejemplos:

- $F_{1+}(3,0) = (0,0)$ $F_{0+}(3,0) = (3,1)$
- $F_{1-}(3,0) = (2,0)$ $F_{0-}(3,0) = (3,3)$

□ Otros casos:

Anillo bidireccional de k nodos: 1-cubo k -ario

Toro tridimensional de $k \times k \times k$ nodos: 3-cubo k -ario



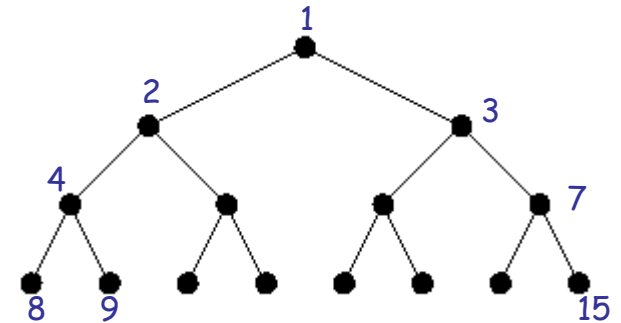
□ Árboles binario, p nodos

o Equilibrado

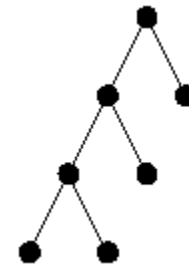
$$F_{izq}(i) = 2i, (i \leq p-1 / 2)$$

$$F_{der}(i) = 2i+1$$

donde $1 \leq i \leq p$



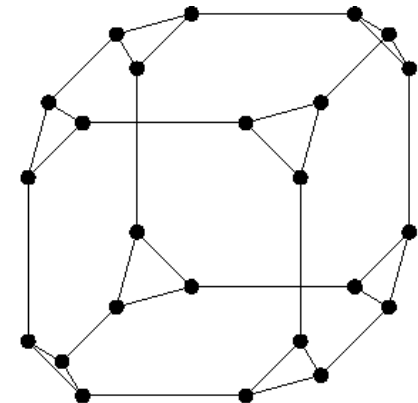
o No-equilibrado



□ Topologías híbridas

o Reducir el grado del nodo manteniendo un diámetro pequeño

o Ej: Ciclos conectados en Cubo



Algunos ejemplos

❑ Intel iPSC2 Hypercube:

- o Topología de hipercubo (6-cubo) binario, canales serie de 2.8 Mbytes/s

❑ Chaos Router:

- o Toro 2D, 8 bits de datos en cada canal, 180 MHz por canal, 360 Mbytes/s en cada dirección

❑ Cray T3D:

- o Toro 3D bidireccional, hasta 1024 nodos (8x16x8), 24 bits de datos en cada dirección (16 de datos y 8 de control, 150 MHz por canal, 300 Mbytes/s por canal)

❑ Cray T3E:

- o Toro 3D bidireccional, 14 bits de datos en cada dirección, 375 MHz por canal, 600 Mbytes/s por canal

❑ MIT M-Machine:

- o Malla 3-D, 800 Mbytes/s por cada canal

- ❑ Los EP se conectan a un conmutador, reconfigurable dinámicamente
- ❑ Modelo
 - o Grafo con conmutadores como vértices y canales como arcos
- ❑ Topologías:
 - o **Regulares**: Multiprocesadores UMA/NUMA, SIMD, ..
 - o Irregulares: Networks of Workstations (NOW), LAN conmutadas
- ❑ Diferencias con redes estáticas:
 - o 1 conmutador puede estar conectado a 0 o más EP
 - o Sólo los conmutadores conectados a EP pueden ser origen/destino de un mensaje

Redes dinámicas (Crossbar)

❑ Caso ideal:

- o Interconexión global → red de barras cruzadas (crossbar):

❑ Definición

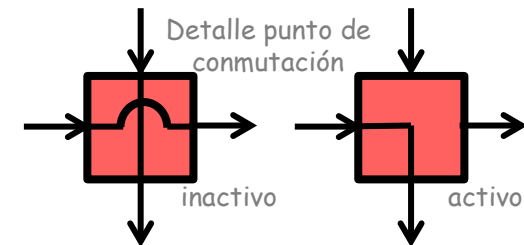
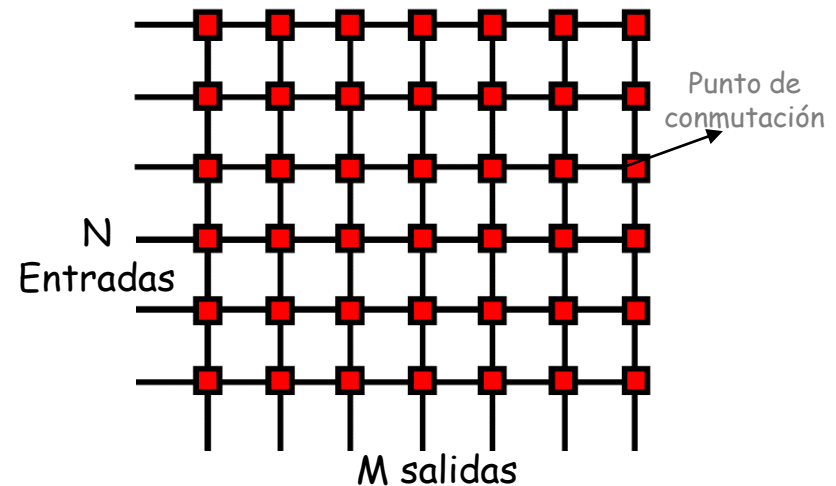
- o Red conmutada de N entradas y M salidas que permite hasta $\min(M, N)$ interconexiones punto a punto sin contención.

❑ Propiedades

- o Es posible una conexión siempre que los puertos de origen y destino estén libres
- o Coste $O(NM)$ → caro y difícil de construir si N y M grandes (nº de pines).

❑ Aplicaciones:

- o Multiprocesadores UMA de pequeña escala, multicores, para conectar procesadores con módulos de memoria
 - Todos los procesadores pueden acceder simultáneamente a memoria siempre que accedan a módulos diferentes.
- o Construcción de conmutadores para redes Dinámicas/indirectas

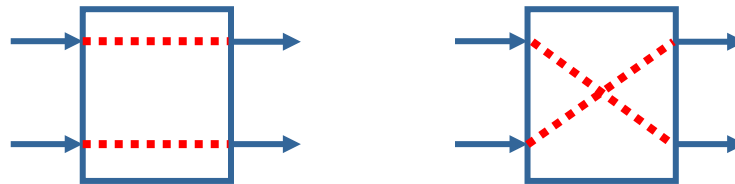


❑ Multistage Interconnection Network (MIN)

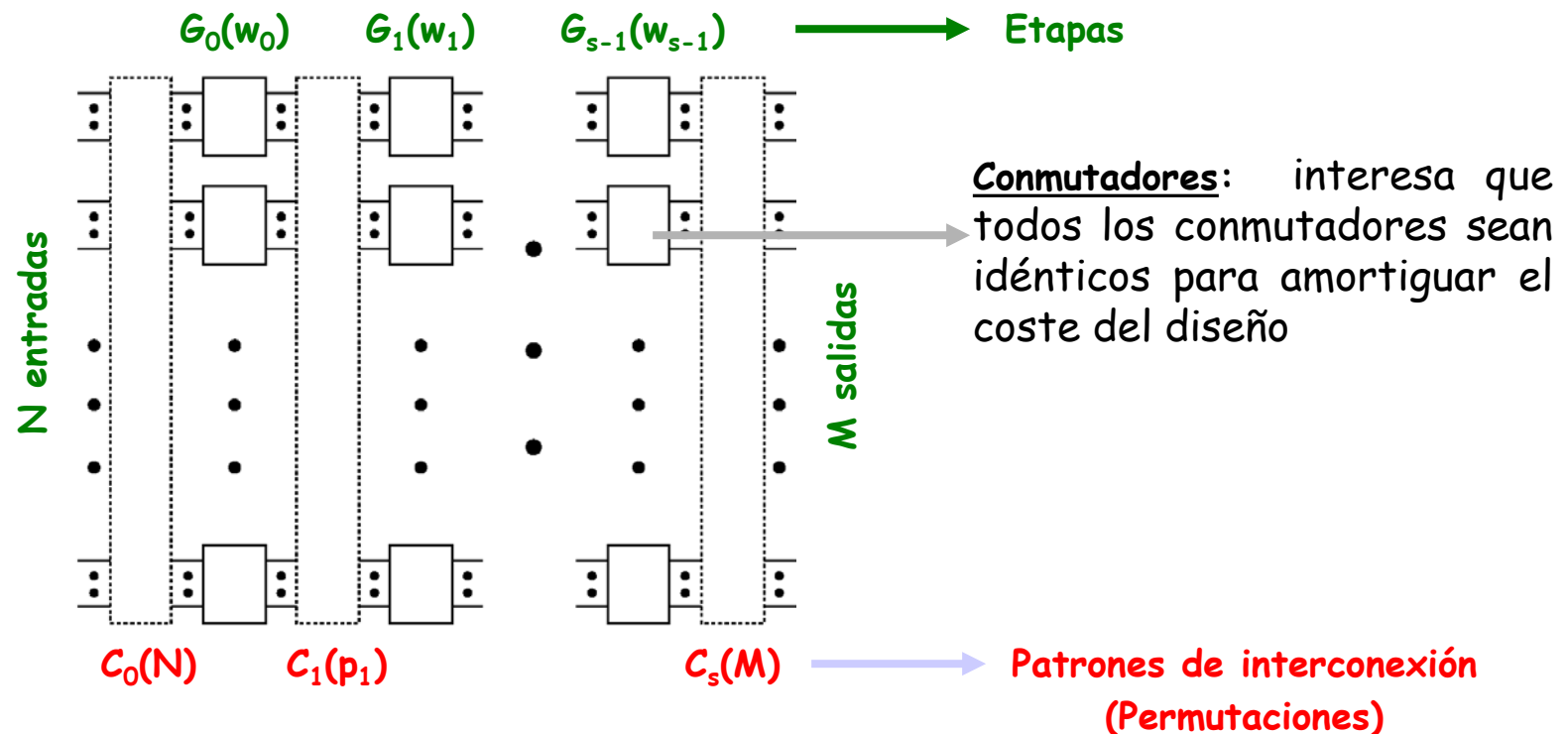
❑ Descripción

- o Conectan dispositivos de entrada a dispositivos de salida a través de etapas compuestas por conmutadores y patrones de interconexión
- o Cada conmutador puede conectar sus E con sus S en cualquier permutación deseada.

❑ Estados de un conmutador 2x2 (unidireccional)



Modelo General de MIN



- ❑ Las MINs fueron utilizadas inicialmente en redes telefónicas y posteriormente en procesadores en array.
 - o En ambos casos, un controlador central establecía el camino entre la entrada y la salida, mediante la configuración de las señales de control de cada etapa de conmutadores, G_i .

Modelo general de MIN

❑ Redes Banyan:

- o Una sola ruta entre un origen y un destino

❑ Red delta de N nodos: subclase Banyan con $N=k^n$:

- o Tiene **n etapas**, con N/k **conmutadores (crossbar)** de $k \times k$ en cada etapa
- o Es muy normal que $k=2$

❑ Permutaciones

- o Conexiones uno a uno entre los k^n puertos que definen los patrones de interconexión entre etapas

❑ Conectividad

- o Bloqueantes
 - El establecimiento de una conexión entre un nodo de entrada y otro de salida puede impedir el establecimiento de otras conexiones.
 - Ejemplo: redes Banyan
- o No bloqueantes
 - El conjunto de entradas puede conectarse con el conjunto de salidas simultáneamente en cualquier permutación.
 - Su conectividad es similar a la de un crossbar

Permutaciones y redes MIN

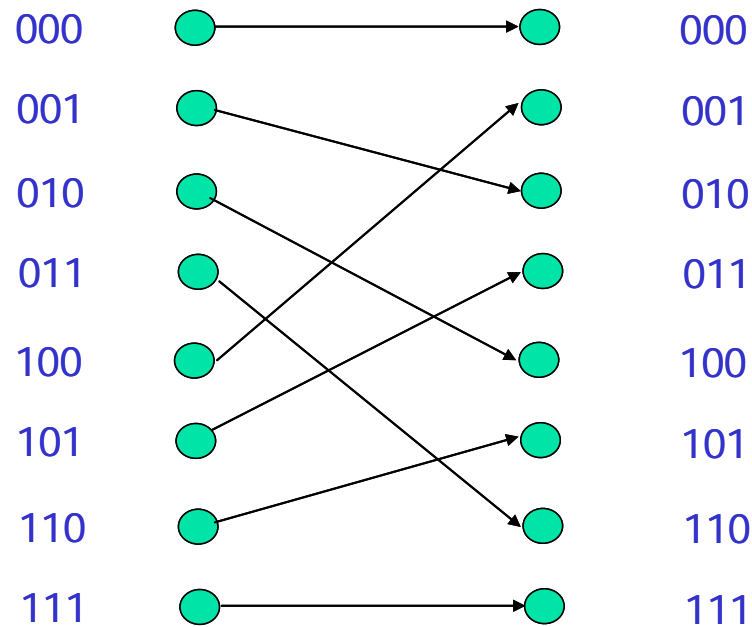
□ Perfect shuffle (σ)

Numerando la E y la S con n dígitos en base k , se define:

Permutación: $\sigma(x_{n-1} x_{n-2} \dots x_1 x_0) = (x_{n-2} \dots x_1 x_0 x_{n-1})$

e inversa: $\sigma^{-1}(x_{n-1} x_{n-2} \dots x_1 x_0) = (x_0 x_{n-1} x_{n-2} \dots x_1)$

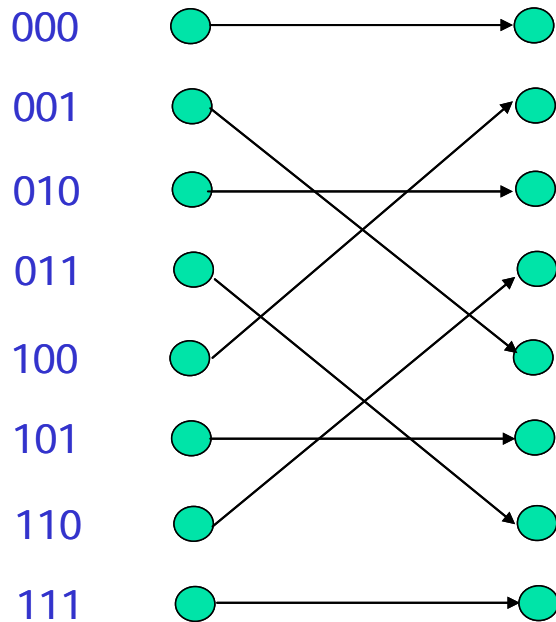
Ejemplo: $N=8, n=3, k=2$. ($N=k^n, 8=2^3$)



□ Butterfly (β)

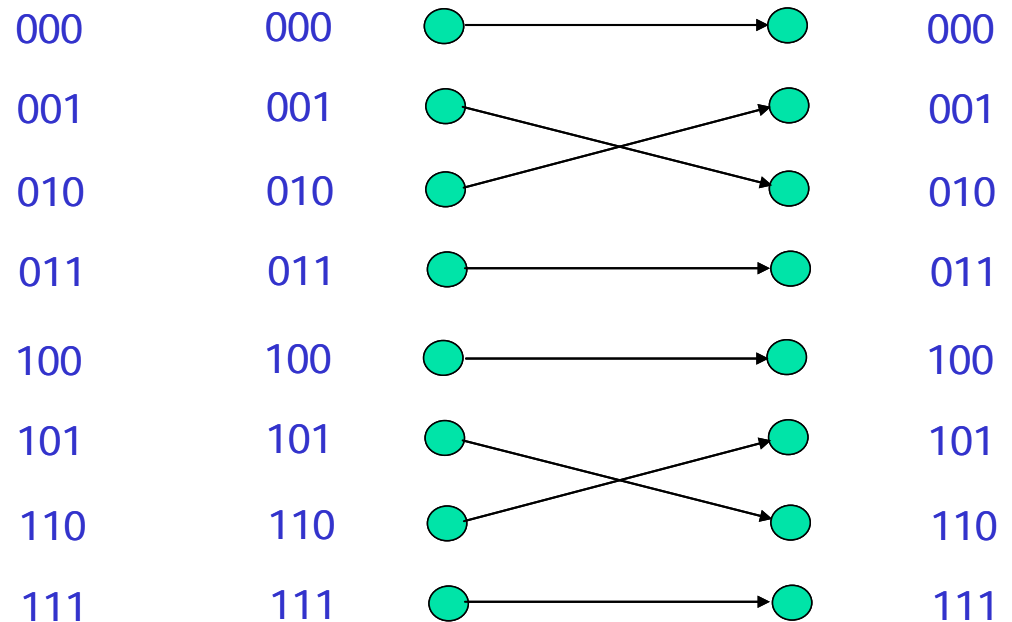
$$\beta_i(x_{n-1} \dots x_{i+1} \color{red}{x_i} x_{i-1} \dots x_1 \color{red}{x_0}) = (x_{n-1} \dots x_{i+1} \color{red}{x_0} x_{i-1} \dots x_1 \color{red}{x_i})$$

Ejemplo: $N=8, n=3, k=2, i=2$



β_2

Ejemplo: $N=8, n=3, k=2, i=1$



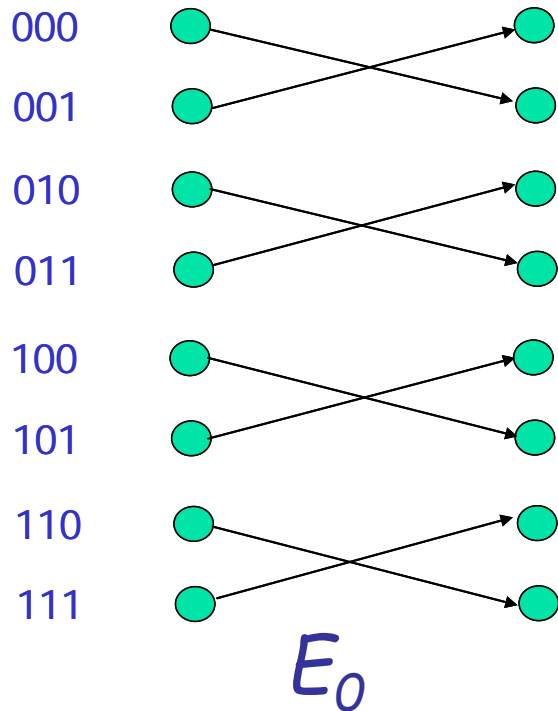
β_1

Permutaciones y redes MIN

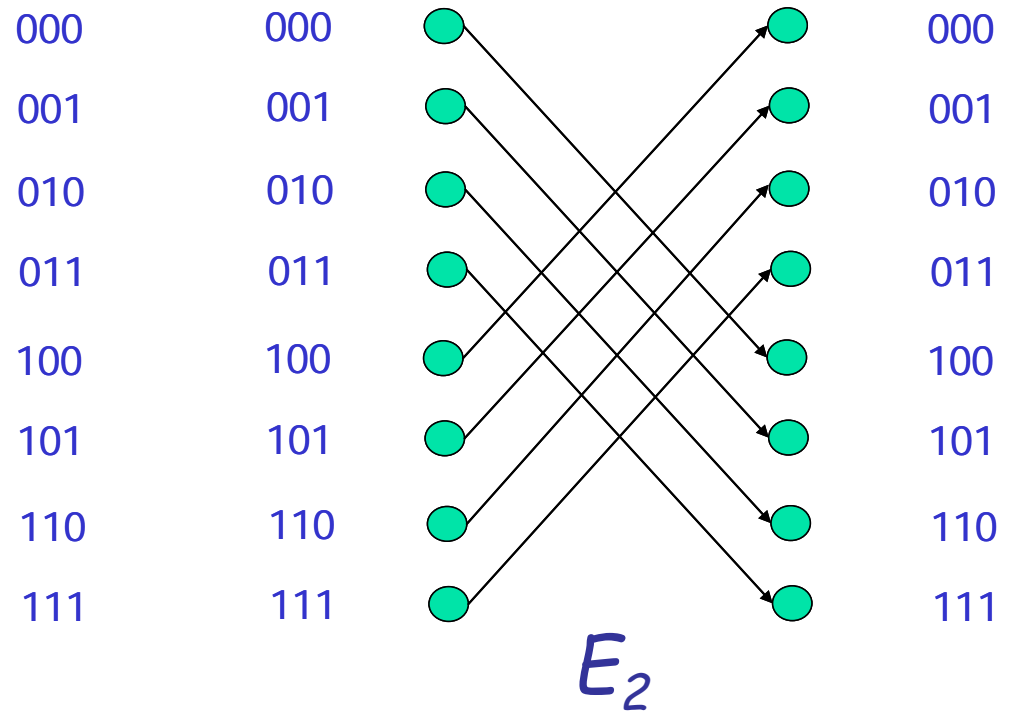
□ Exchange (E)

$$E_i(x_{n-1} \dots x_{i+1} \color{red}{x_i} x_{i-1} \dots x_1 x_0) = (x_{n-1} \dots x_{i+1} \color{red}{x'_i} x_{i-1} \dots x_1 x_0)$$

Ejemplo: $N=8, n=3, k=2, i=0$



Ejemplo: $N=8, n=3, k=2, i=2$



□ MIN Butterfly

Red delta de N nodos,
 $N = k^n$ puertos \rightarrow n etapas,
 conmutadores $k \times k$

Def: $C_i \rightarrow \beta_i$, $C_0 \rightarrow \beta_0$, $C_n \rightarrow \beta_0$

Ejemplo:

$N=16$, $k=2$, $n=4$

$C_0 = C_4 = \beta_0 = I$

$C_1 = \beta_1$

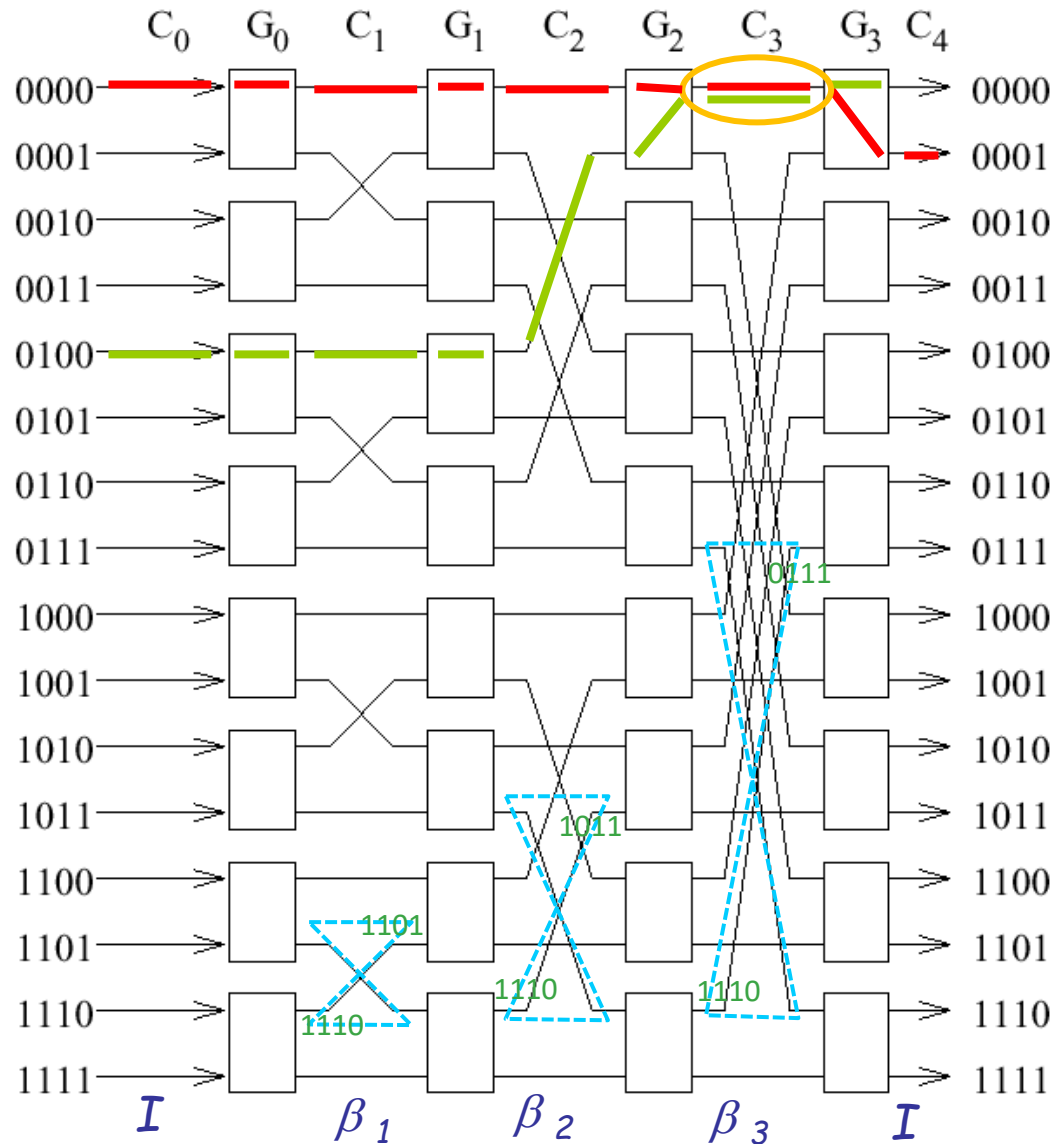
$C_2 = \beta_2$

$C_3 = \beta_3$

N/k conmutadores/etapa: $16/2=8$

4 etapas

Bloqueante: P. ej. la ruta de 0 a 1
 no es compatible con la ruta del 4
 al 0.



Redes MIN relevantes

□ MIN Omega

Red delta de N nodos,

$N = k^n$ puertos,

Def: $C_i \rightarrow \sigma$ (PS), $C_n \rightarrow I$

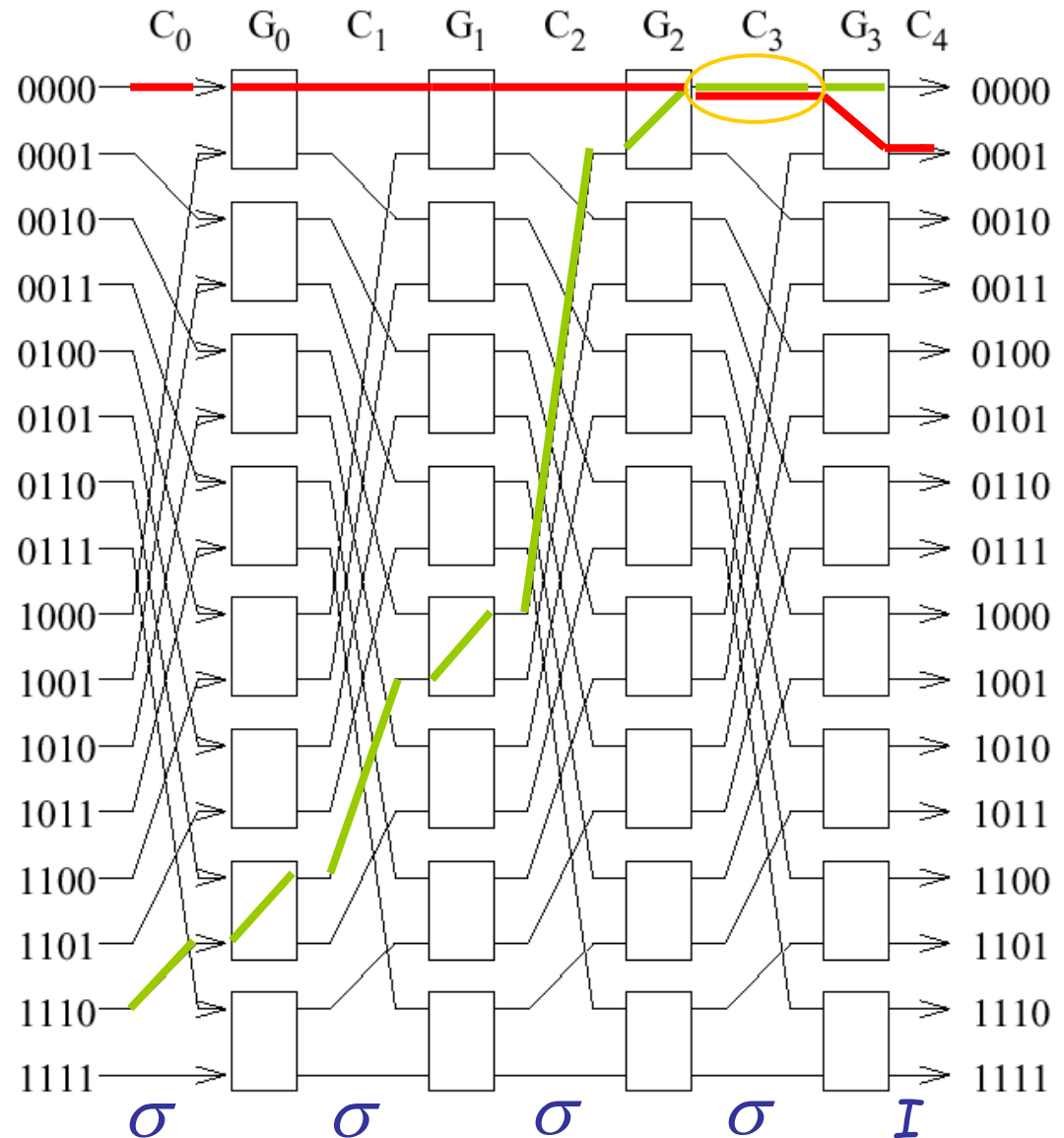
Ejemplo:

$N=16$, $k=2$, $n=4$

N/k conmutadores/etapa: $16/2=8$

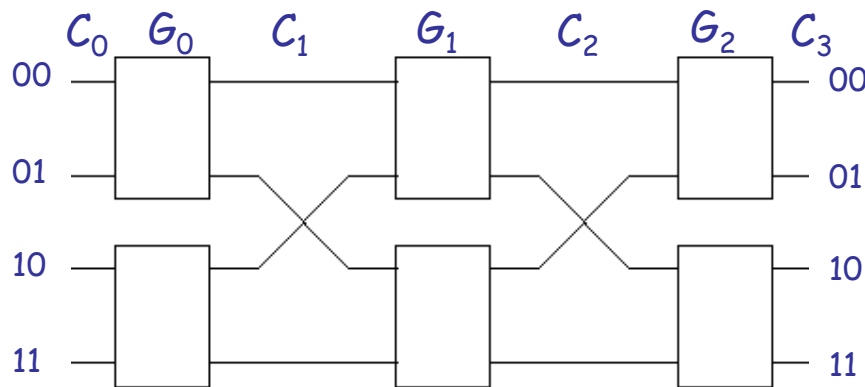
4 etapas

Bloqueante: P. ej. la ruta de 0 a 1
no es compatible con la ruta del
14 al 0.

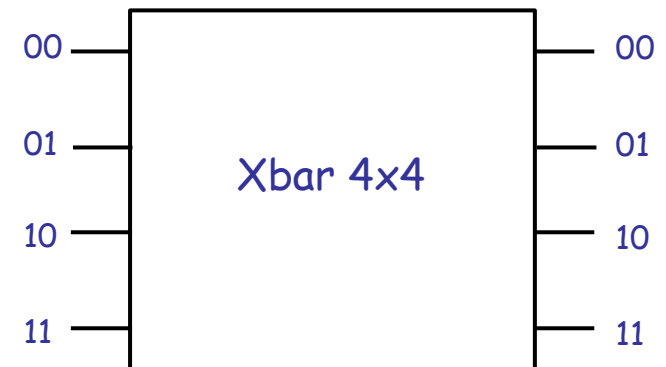


□ MIN Beneš

- o Evitar bloqueo: poner etapas de conmutadores (2x2) adicionales o usar conmutadores más complejos (p. ej. 4x4)
- o La red Beneš de N nodos (Beneš NxN) usa $2(\log_2 N) - 1$ etapas
 - $\log_2 N$ como antes + $(\log_2 N) - 1$ adicionales
- o Ejemplo: red Beneš 4x4 ($n=2, k=2 \rightarrow N=k^n=4$)
 - 3 etapas ($C_1 = \sigma^{-1}, C_{2n-2} = C_2 = \sigma$) (PS)
 - No hay bloqueo
 - Cualquier permutación E/S es posible \rightarrow equivalencia funcional con un Xbar 4x4



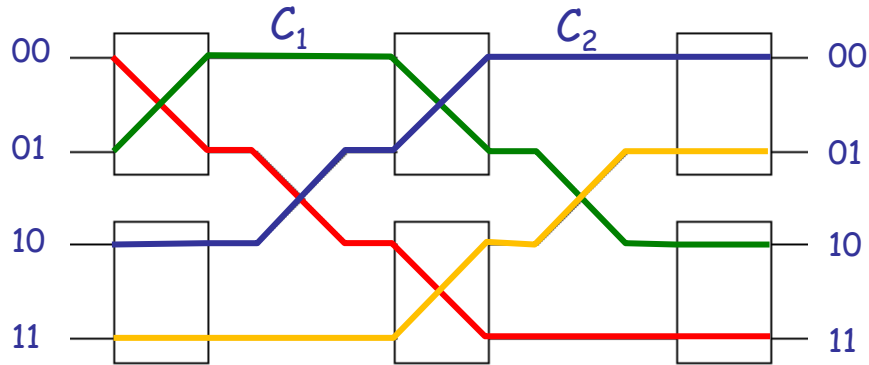
Beneš (4x4)



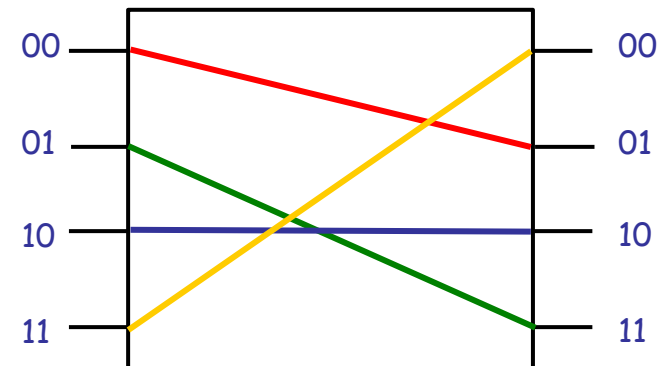
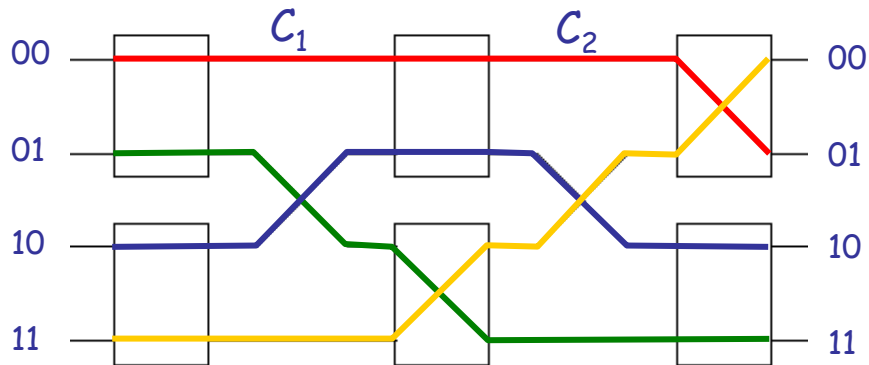
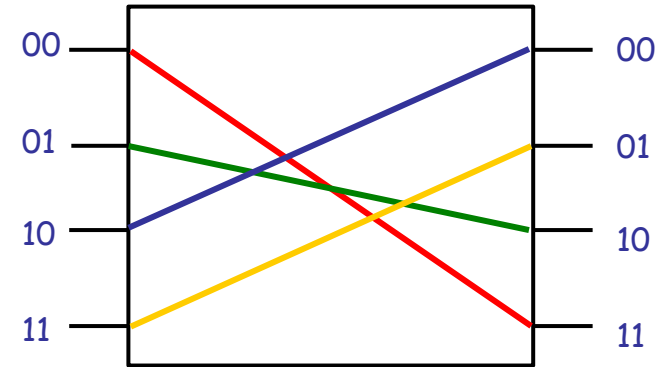
Redes MIN relevantes

□ Ejemplos

Beneš (4x4)

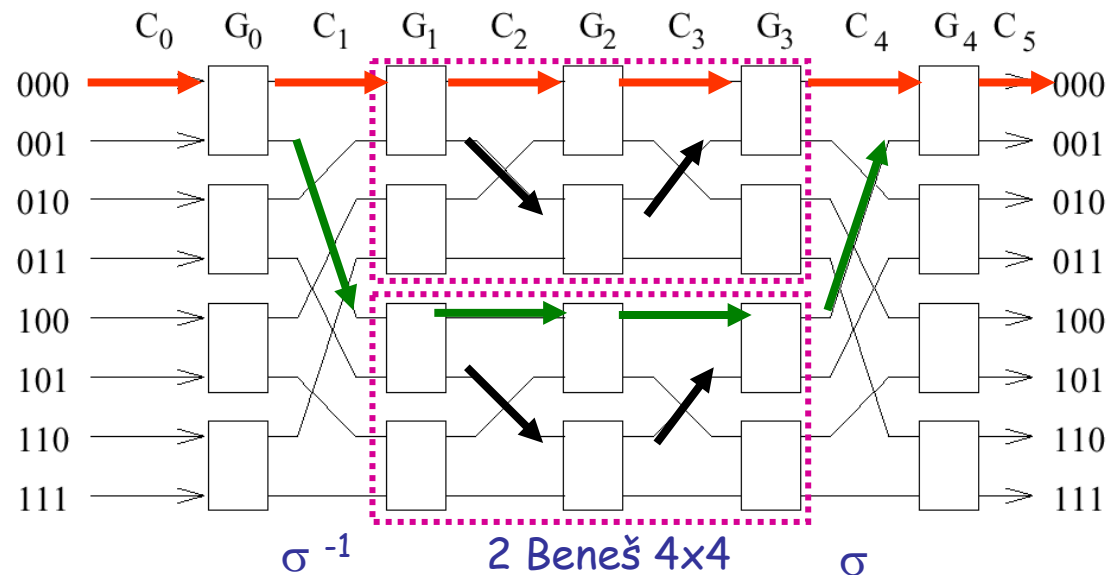


Xbar 4x4



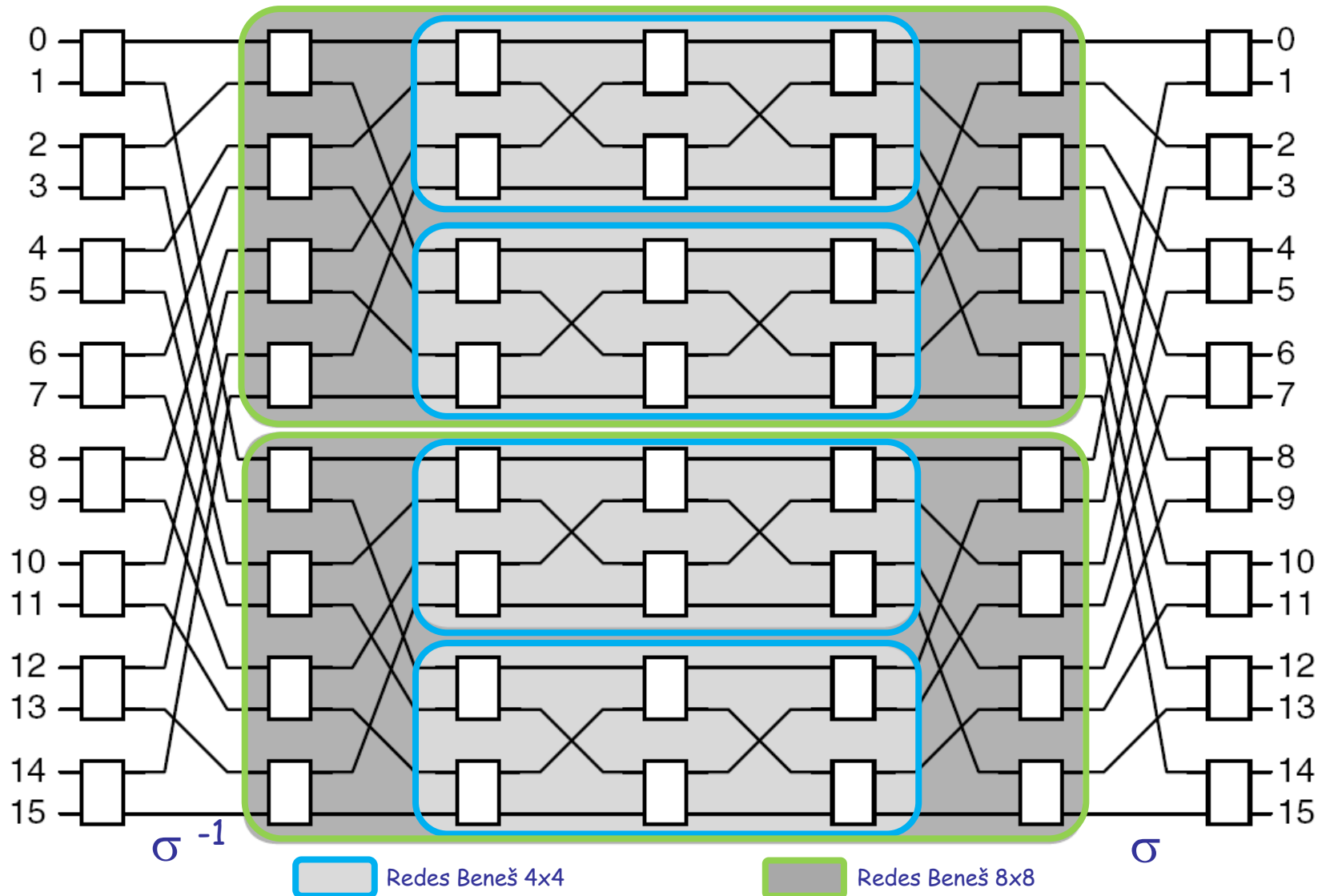
□ MIN Beneš

- o Se pueden construir redes con mayor nº de elementos a partir de redes más pequeñas
- o Una red Beneš de $N \times N$, se construye con:
 - 2 redes de Beneš de $N/2 \times N/2$
 - $C_1 \rightarrow \sigma^{-1}$, $C_{2n-2} \rightarrow \sigma$ (PS)
- o Ejemplo: Beneš 8×8 ($n=3$, etapas: $2 \cdot 3 - 1 = 5$)
 - Las flechas coloreadas muestran caminos alternativos desde la entrada 0 a la salida 0.



Redes MIN relevantes

❑ Construcción iterativa de una red Beneš 16x16

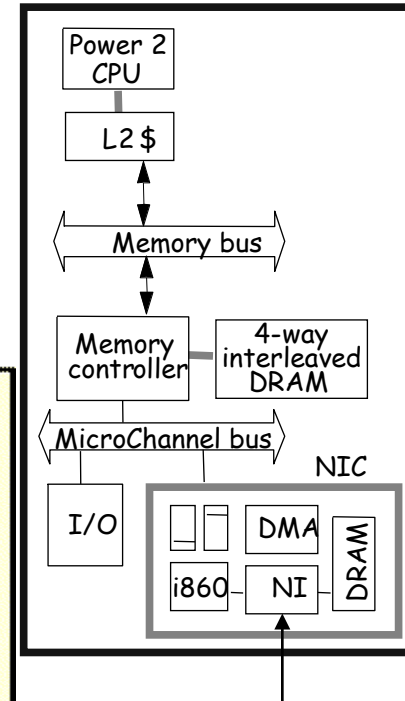


Ejemplos: IBM SP2

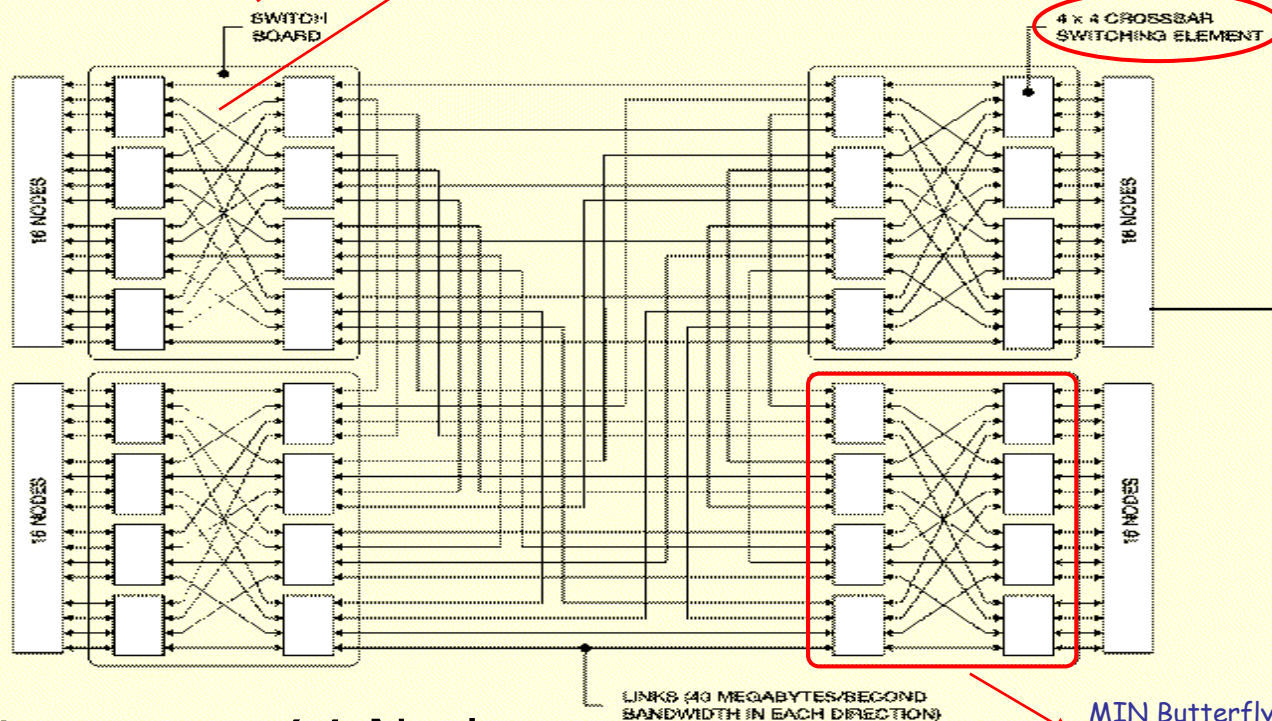


Permutación Baraje
Perfecto base 4

Nodo original IBM SP-2



Lectura: Culler 10.8.2



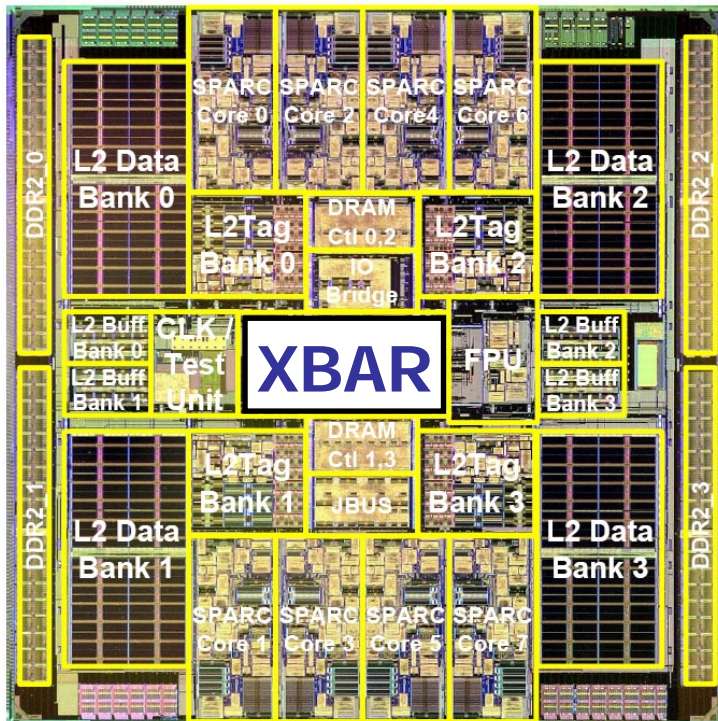
Sistema con 64 Nodos

MIN Butterfly
 $k=4, n=2 \rightarrow N=16$

Multicore Ejemplo (1)

- Sun Niagara (T1) 8 cores, 90 nm, 279 Mtrans, 378 mm² (2004)

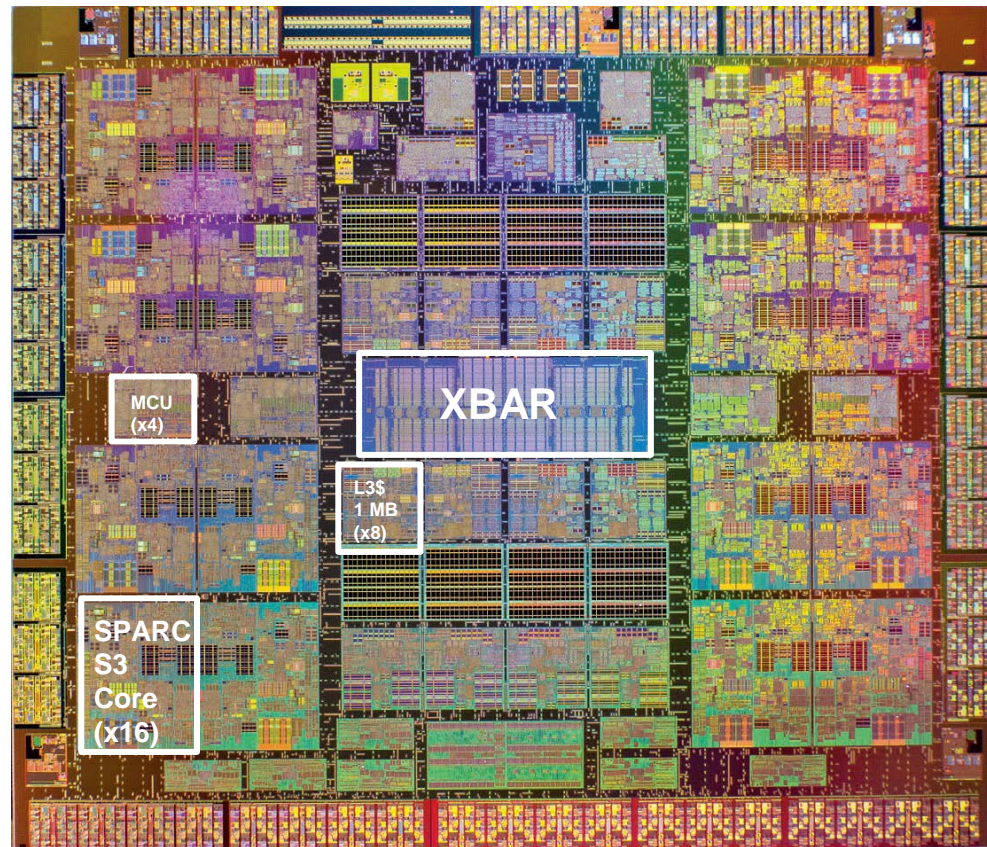
- XBAR: comunica 8 cores con 4 módulos de L2\$ + FP + I/O



Ref: IEEE Micro MARCH/APRIL 2005

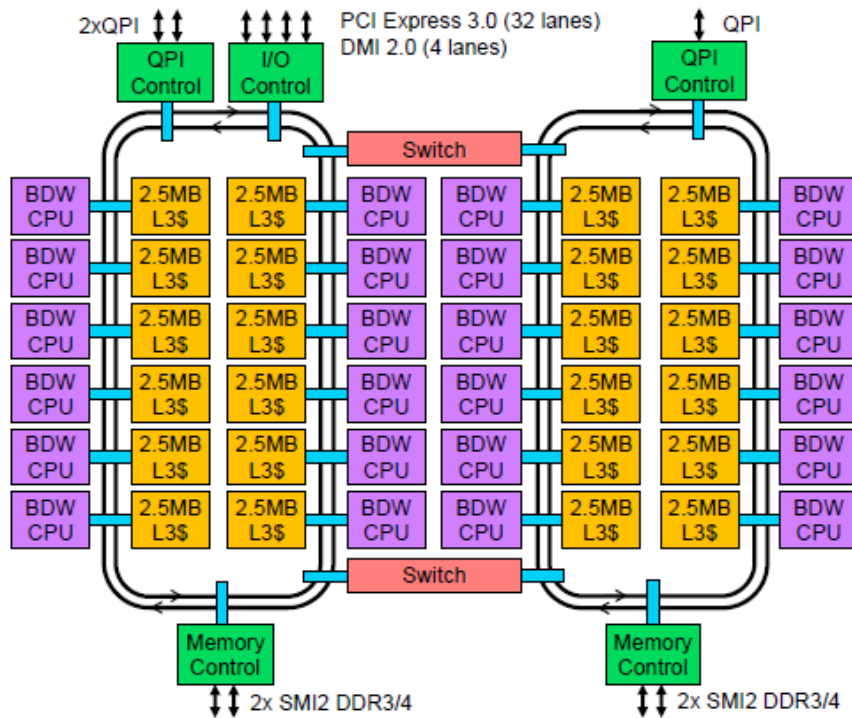
- Oracle SPARC T5 16 cores, 28 nm, 1.5 Btrans., 478 mm² (2012)

- XBAR: comunica 16 cores (2 cores comparten 1 puerto) con 8 módulos de L3\$ + I/O

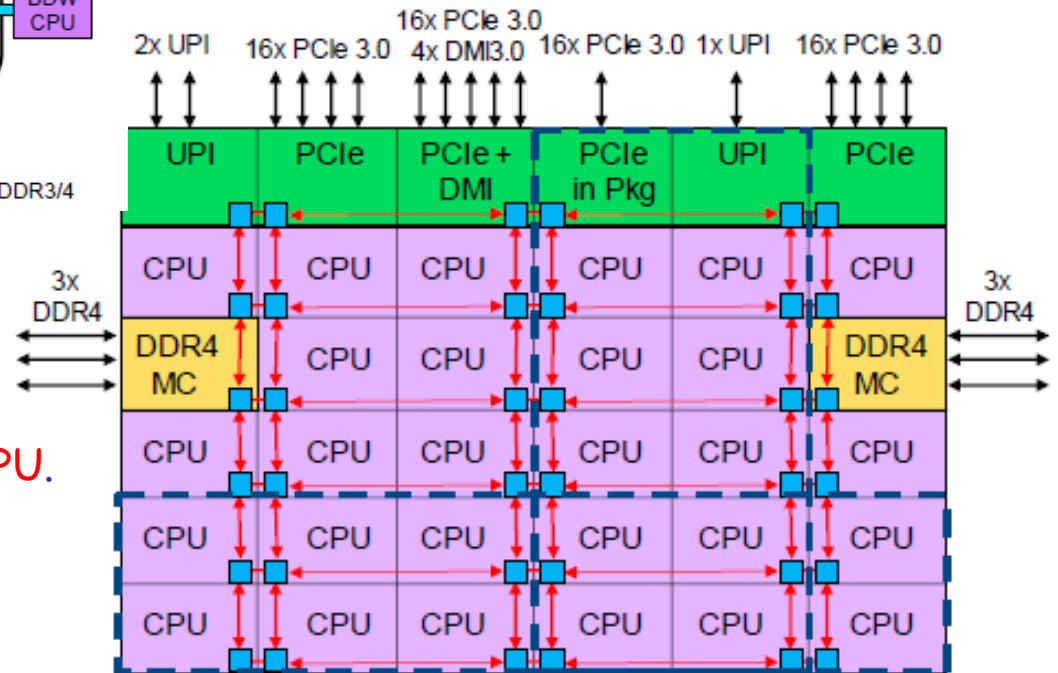


Ref: IEEE Micro MARCH/APRIL 2013

Multicore Ejemplo (2)



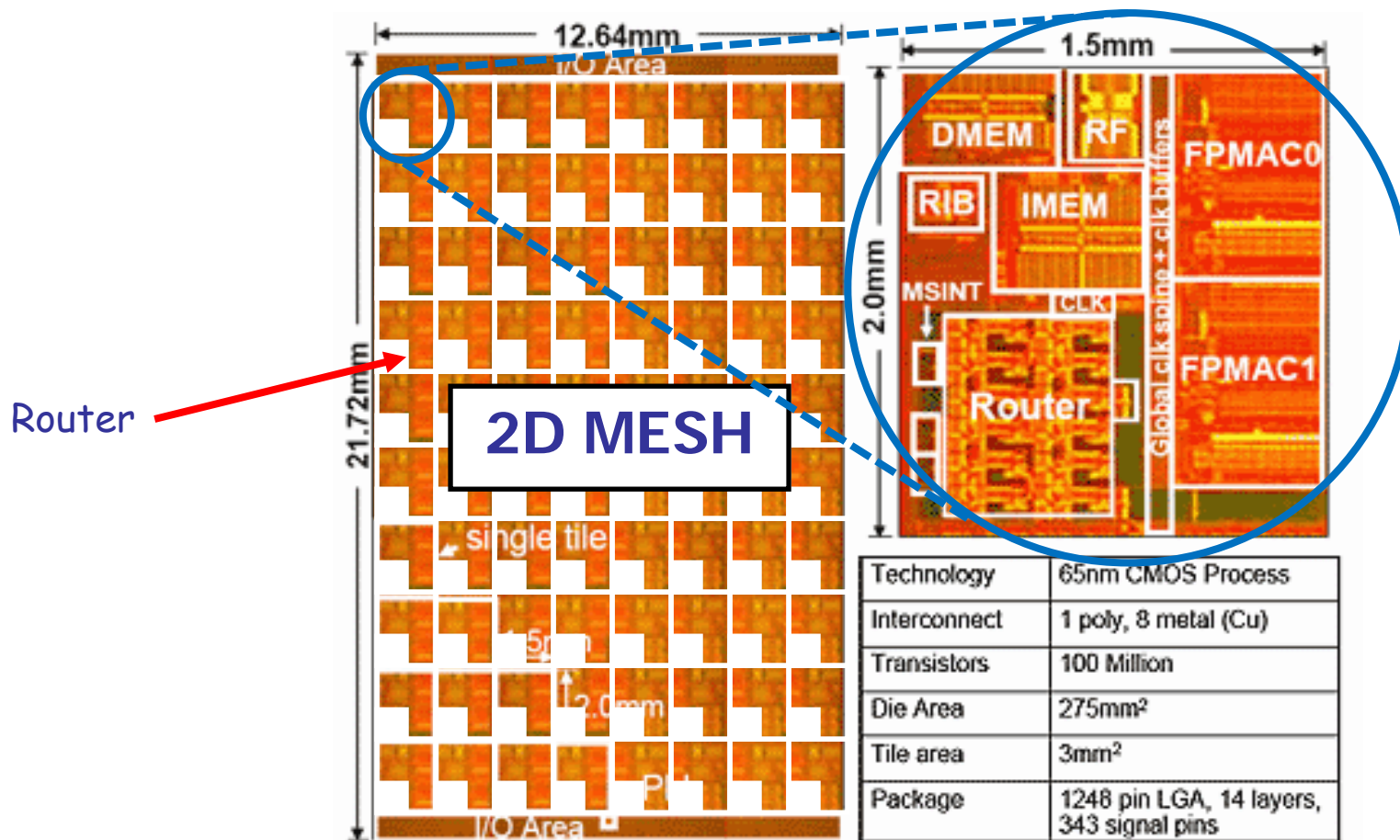
Intel Xeon E5 y E7, anillos, 1,2,...
E7 v4 dos anillos bidireccionales,
24 Cores , 60MB L3



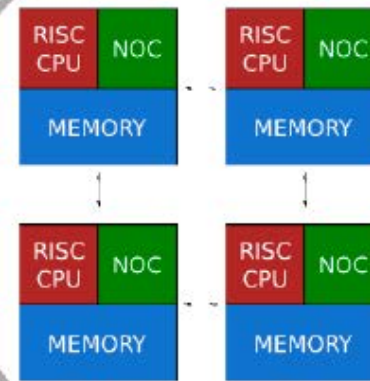
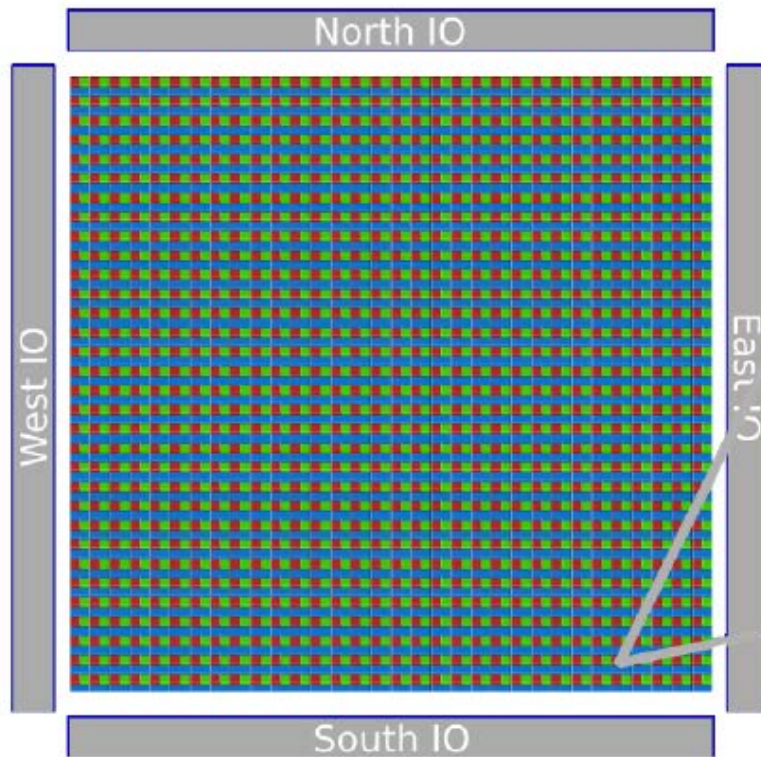
- Skylake-SP, Malla 6x6, 28 módulos CPU.
- Módulo CPU: un core SkyLake-SP y 1,375MB de L3 (Total 38,5MB)

Many Core Ejemplo (1)

- ❑ Intel Polaris (2006)
 - o 80 core prototype, VLIW architecture
 - o 2-D Mesh Topology
 - o Router 25% del área



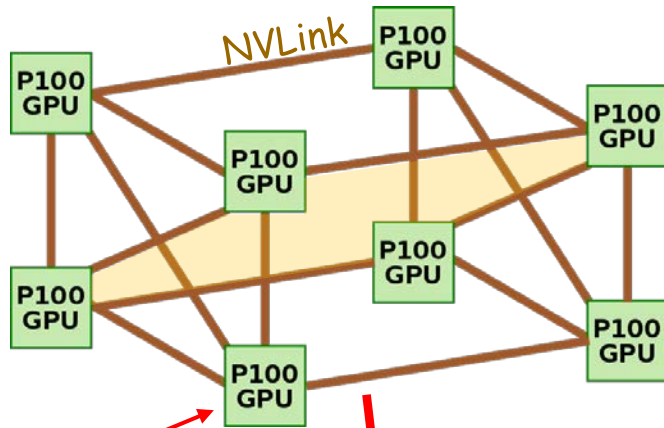
Adapteva Epiphany-V (2016)



- 1024 64-bit RISC processors
- 64 MB of distributed on-chip SRAM
- Three 136-bit wide 2D mesh NOCs
- 4,560 Mtransistors, 117mm², 16nm

Red de interconexión GPUs (Ejemplo: NVLink)

Posible topología híbrida cubo malla

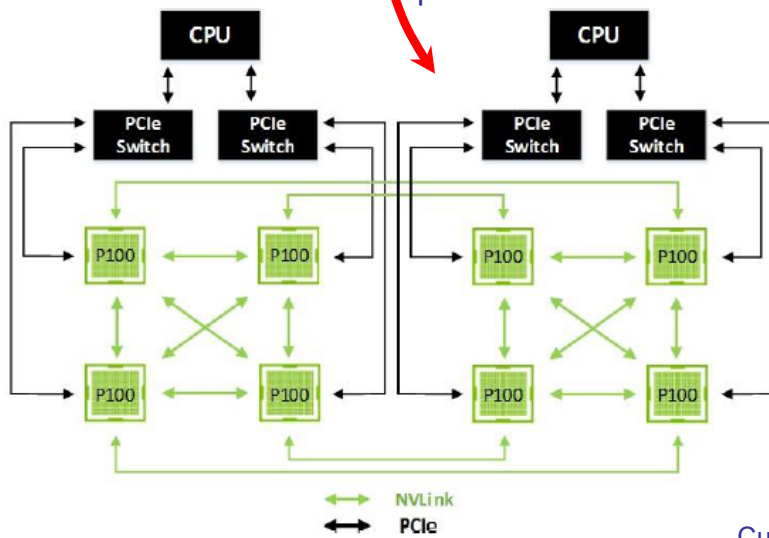


NVLink:

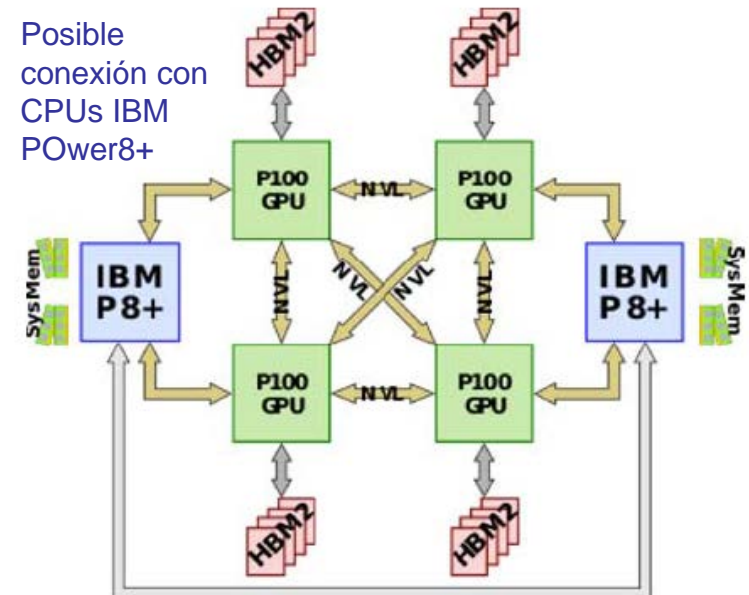
- 40 GB/s bidir por enlace
- 8 hilos/enlace
- 4 enlaces/chip (6 en NVLink 2.0)
- Directamente compatible con CPUs IBM Power 8+, 9

Pascal GP100 GPU

Posible conexión con CPUs tipo Intel



Posible conexión con CPUs IBM Power8+



Técnicas de conmutación

- ❑ Enfoques básicos: Conmutación de circuitos vs. conmutación de paquetes
- ❑ Conmutación de circuitos.
 - o Antes del comienzo del envío del mensaje se establece cuál será la secuencia de enlaces que atravesarán los datos desde el origen al destino.
 - Un mensaje sin datos (cabecera) recorre el camino completo y va reservando los enlaces.
 - Cuando se alcanza el destino se devuelve un ACK al origen por el mismo camino ya establecido.
 - o Una vez establecido el camino todos los datos lo siguen.
 - o Los canales que componen el camino quedan reservados desde el comienzo hasta el final de la transmisión → durante ese tiempo no puede usarse para otras comunicaciones.
- ❑ Conmutación de paquetes.
 - o Cada decisión de rutado se va tomando según el mensaje progresa
 - o Los enlaces que no estén siendo utilizados en un momento dado, pueden usarse para otra comunicación .
 - o Variantes:
 - Store-and-Forward
 - Cut-Through

Técnicas de conmutación: latencia

❑ Suposiciones

- o No hay congestión en la red: latencia básica
- o Se quiere transmitir un mensaje que atravesará h enlaces (hops)
- o El tamaño del mensaje es n bits y el ancho de banda es B
- o El tiempo para decidir cuál deber ser el enlace siguiente a recorrer (rutado) y establecer la conexión con ese enlace (conmutación) es Δ

❑ Latencia con conmutación de circuitos

- o $T_{cc}(n,h) = h\Delta + n/B$
 - tiempo previo para decidir la ruta, más tiempo de transmitir el mensaje completo de fuente a destino
 - Ventajas: una vez seleccionada ruta (circuito) la transmisión se produce a la máxima velocidad (determinada por el AB)
 - Inconvenientes: reserva de los canales (desde antes de comenzar la transmisión hasta que el último dato llega al destino)

❑ Latencia con Store-and-Forward

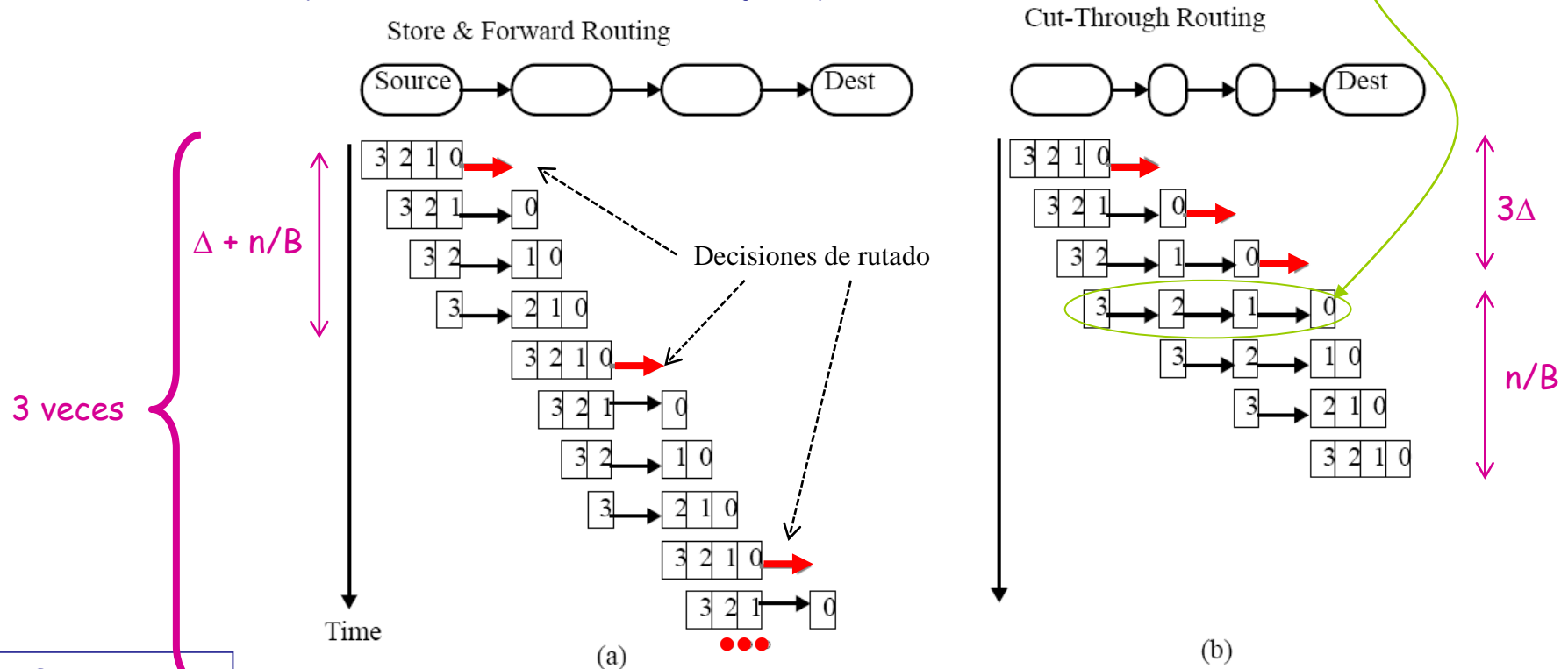
- El mensaje completo se transmite de un nodo al siguiente.
- Cuando se ha recibido completamente, se decide el siguiente enlace a seguir y se vuelve a transmitir $\rightarrow h$ transmisiones + h decisiones
- o $T_{sf}(n,h) = h\Delta + h(n/B) = h(\Delta + n/B)$
 - Menos enlaces reservados que en CC, pero ...
 - Latencia mucho peor: Proporcional a h (topología)
- o ¿Hay alguna forma de no reservar enlaces y tener mejor latencia?

Técnicas de conmutación: latencia

Latencia con Cut-Through

- El paquete está dividido en trozos de forma natural: flits
- Cuando se recibe el primer flit decidir el siguiente enlace de la ruta y comenzar la retransmisión, sin esperar a la recepción de los restantes flits
- $T_{ct}(n,h) = h\Delta + n/B$
 - La transmisión es similar a CC, pero de forma segmentada.
 - No hay reserva de circuitos, pero tiempo básicamente comparable a CC

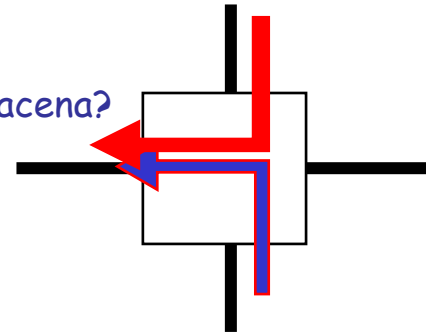
Comparación SF vs. CT (ejemplo: $h=3$)



Técnicas de conmutación: contención

❑ El problema de la contención

- o Dos paquetes intentan usar el mismo canal al mismo tiempo
 - ¿Quién pasa? Arbitraje
 - ¿Qué ocurre con el paquete que se bloquea? ¿Dónde se almacena?
 - El tamaño de la memoria es limitado
 - Cómo afecta a las prestaciones → latencia
 - Caso extremo → Saturación e Interbloqueos (Deadlock)



❑ Store&Forward

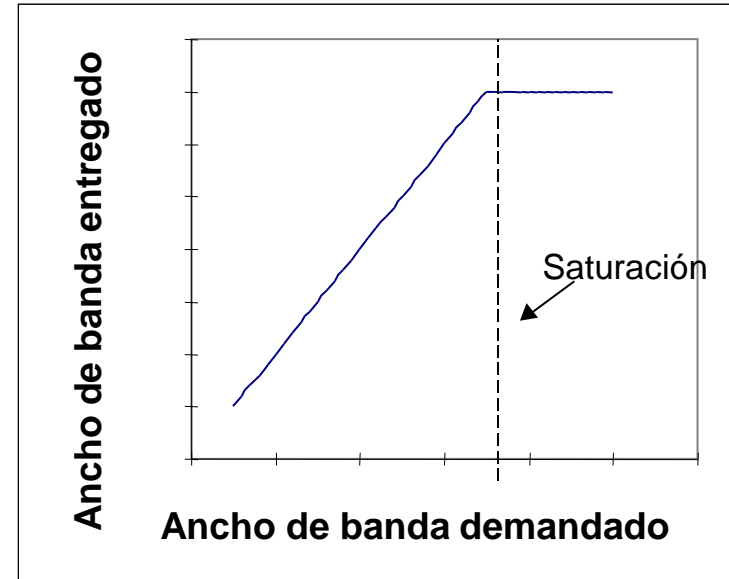
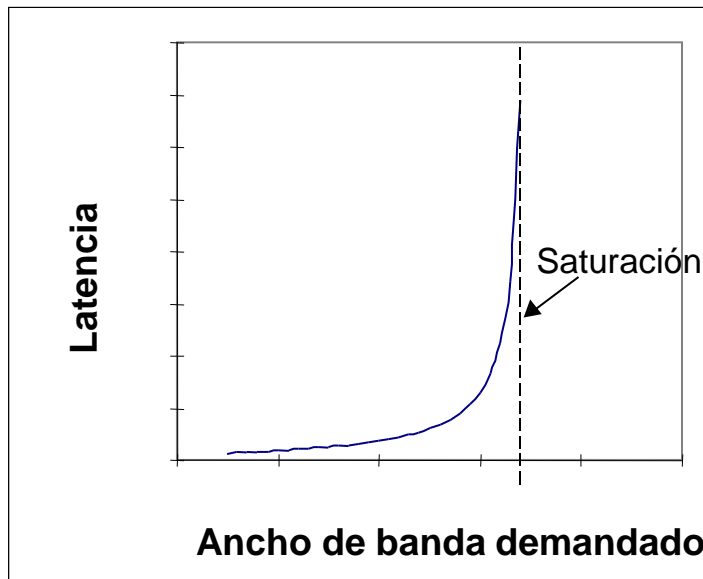
- o El paquete bloqueado se recibe y se almacena completo en buffers del nodo/encaminador

❑ Cut-through

- o Virtual Cut-Through. Los flits del paquete bloqueado se van recibiendo y almacenando en buffers del encaminador
 - En situaciones de contención → comportamiento se degrada hasta ser similar a SF
 - Necesitamos buffers con tamaño suficiente para almacenar un paquete
- o Wormhole. Los buffers del encaminador sólo almacenan unos pocos flits del mensaje bloqueado. Los restantes flits permanecen en los encaminadores previos sin avanzar.
 - Buffers más pequeños
 - Un paquete bloqueado en la red puede estar repartido entre varios encaminadores a lo largo de la ruta → Propagación hacia atrás (back-pressure) de la contención (mecanismo de control del flujo)
 - Cuidado con interbloqueos --> ocupación extensiva de los canales

Latencia y AB

- ❑ La latencia y el AB entregado por la red son dependientes del AB demandado
 - o Mientras que el AB demandado está claramente por debajo de la capacidad máxima de la red (poca contención) la latencia se mantiene baja (básicamente cte)
 - o A partir de cierto punto, si se demanda más AB (los EP producen más paquetes por ut), se produce la saturación
 - La latencia aumenta rápidamente
 - El AB entregado no se incrementa
 - o Recordar! Un multiprocesador es un sistema cerrado
 - Aumento AB demandado → Aumento latencia comunicaciones → Reducción avance programas → Reducción paquetes generados → Reducción AB demandado.



❑ Objetivo

- o Determinar la ruta que debe seguir un mensaje desde el origen hasta el destino
- o Comportamiento ideal
 - Baja latencia
 - Distribución de la carga
 - Tolerancia a fallos
 - Libre de interbloqueos

❑ Mecanismo de encaminamiento

- o Aritmético: determinar el siguiente tramo de la ruta mediante una op aritmética simple
- o Selección de la ruta en origen: la información de ruta acompaña al paquete. Problema: incremento de longitud.
- o Mediante tablas: Cada conmutador contiene una tabla de rutas. Cada paquete contiene un índice de acceso a la tabla de cada conmutador atravesado.

❑ Propiedades del encaminamiento

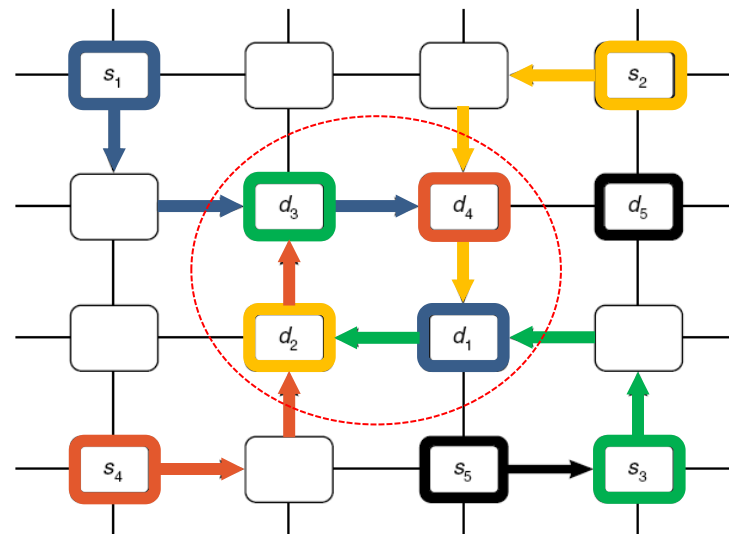
- o Mínimo / No mínimo
- o Determinista / Adaptativo

Problemas del encaminamiento

- ❑ Interbloqueo (deadlock)

- o Un paquete está a la espera de un evento que no puede ocurrir
 - P. ej. Los paquetes no pueden avanzar hacia su destino porque cada uno está esperando en un enlace a que otro paquete deje disponible un buffer de entrada. Pero eso no puede ocurrir porque los enlaces bloqueados forman un ciclo
- o Una red bien diseñada debe estar libre de interbloqueos

Ejemplo (interbloqueo en una malla 2D): 4 mensajes simultáneos de s_i a d_i ($i=1,4$)



Cada mensaje hace tres saltos y queda interbloqueado en el ciclo (d1, d2, d3, d4)

El mensaje de s5 a d5 tb se ve afectado por el interbloqueo

- ❑ Espera indefinida (starvation)

- o Un paquete está a la espera de un evento que puede ocurrir, pero no ocurre. Puede deberse a una mala planificación del sistema (fairness)

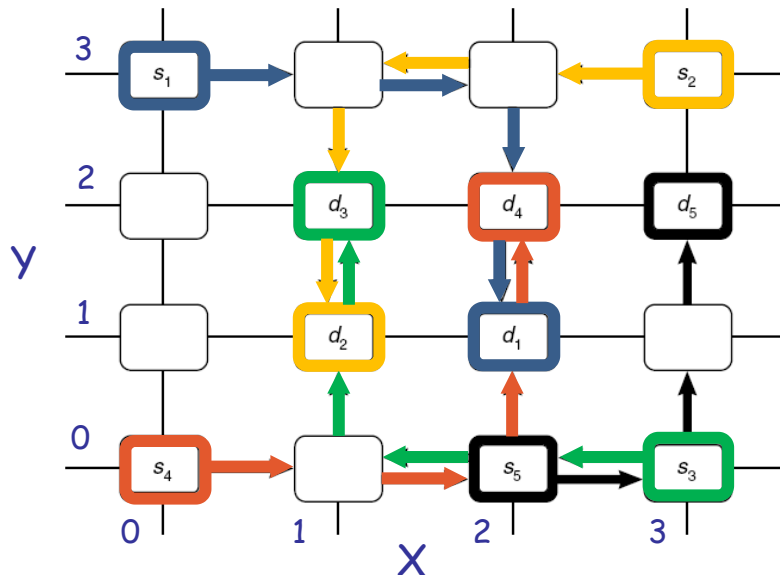
- ❑ Ciclos infinitos (livelock)

- o Un paquete está recorriendo nodos sin llegar nunca a su destino. Sólo puede ocurrir con encaminamiento adaptativo no mínimo

Mecanismos de encaminamiento

□ Orden dimensional (Dimension-order routing)

- o En la propagación del paquete las diferentes dimensiones de la red (X, Y, Z, ...) se siguen en un orden estricto
- o Los enlaces en una cierta dimensión no pueden ser utilizados por un paquete hasta que no ha recorrido todos los enlaces necesarios (para alcanzar el destino) en todas las dimensiones precedentes.
- o Aplicación en mallas, hipercubos, ...
- o Ejemplo en una malla 2D: los paquetes siguen el orden dimensional X,Y.
 - El anterior interbloqueo no existe
 - Problema potencial: sólo existe una ruta posible para cada comunicación (determinista)
 - Saturación, no tolerancia a fallos, desaprovechamiento de enlaces poco usados, ...



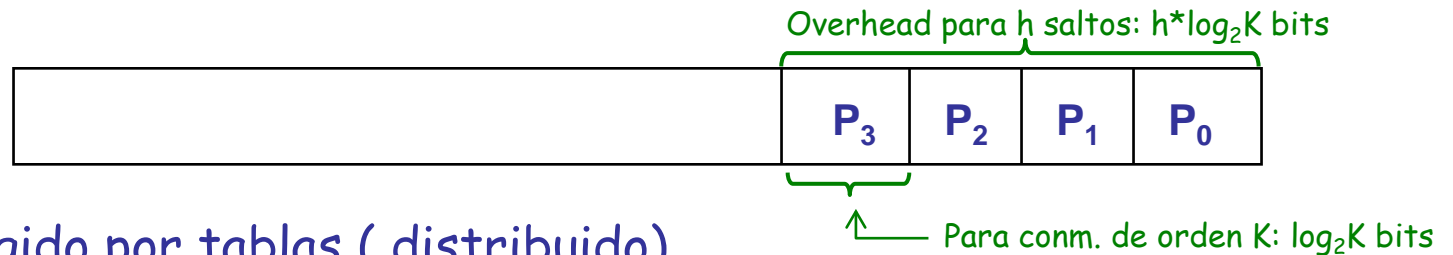
Algoritmo: paquete desde (x_1, y_1) hasta (x_2, y_2)

- o Nodo actual: (x, y) . Inicialmente $(x, y) = (x_1, y_1)$
- o $(Dx, Dy) = (x_2 - x, y_2 - y)$
- o Si $Dx > 0$, incrementar x (este)
- o Si $Dx < 0$, decrementar x (oeste)
- o Si $Dx = 0$
 - Si $Dy > 0$, incrementar y (norte)
 - Si $Dy < 0$, decrementar y (sur)

Mecanismos de encaminamiento

❑ Selección de ruta en origen

- o La cabecera del mensaje contiene la serie de puertos seleccionados
- o Utilizado y dividido en ruta: Cada conmutador usa el primer puntero y lo elimina de la lista.
- o Ejemplos: Meiko CS-2, Myrinet, MIT Artic



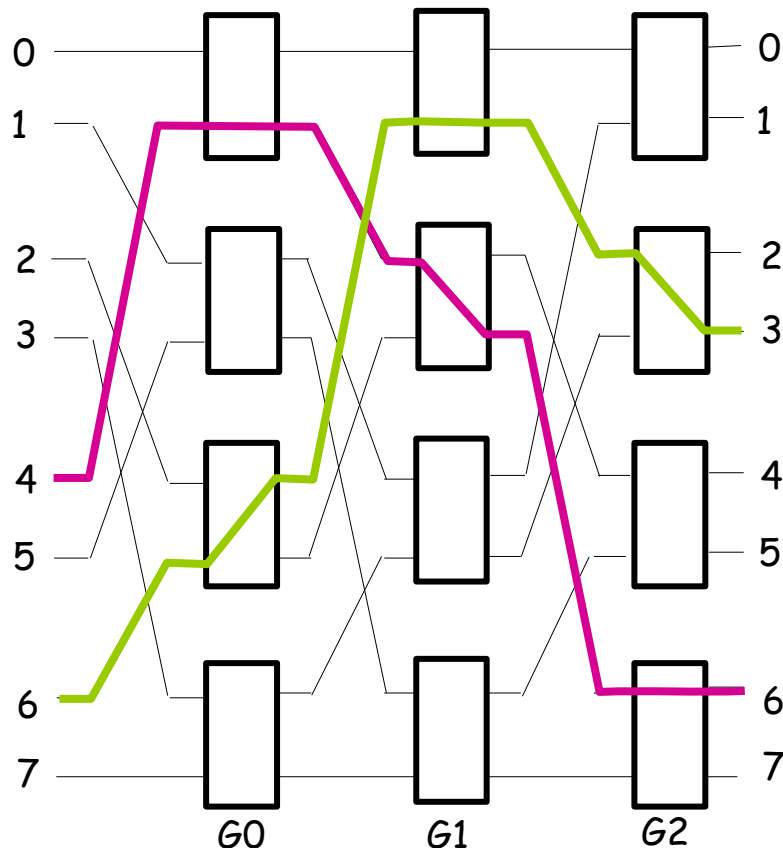
❑ Dirigido por tablas (distribuido)

- o Cada conmutador contiene una tabla de rutado, R
- o La cabecera del mensaje contiene un índice, i , que determina la entrada de la tabla de rutado que especifica el puerto de salida.
 - $\text{Out_port} = R[i]$; ($R[i]$ = entrada i -ésima de tabla de rutado)
- o Generalmente la tabla también proporciona el índice, i' , para el siguiente switch
 - $(\text{Out_port}, i') = R[i]$
- o Ejemplo: ATM

Mecanismos de encaminamiento

- MIN: Uso de la dirección de destino

Red Omega



□ Funcionamiento:

- o Etapas: $n=3$
- o Dir destino: (d_2, d_1, d_0)
- o El bit d_i se usa para seleccionar el funcionamiento de los conmutadores de la etapa $n-i-1$
 - $d_i = 0 \rightarrow$ Salida superior
 - $d_i = 1 \rightarrow$ Salida inferior

□ Ejemplo 1: paquete de 4 a 6

- o La dir destino es (110)
 - $d_2=1 \rightarrow$ Salida inferior en etapa 0
 - $d_1=1 \rightarrow$ Salida inferior en etapa 1
 - $d_0=0 \rightarrow$ Salida superior en etapa 2

□ Ejemplo 2: paquete de 6 a 3

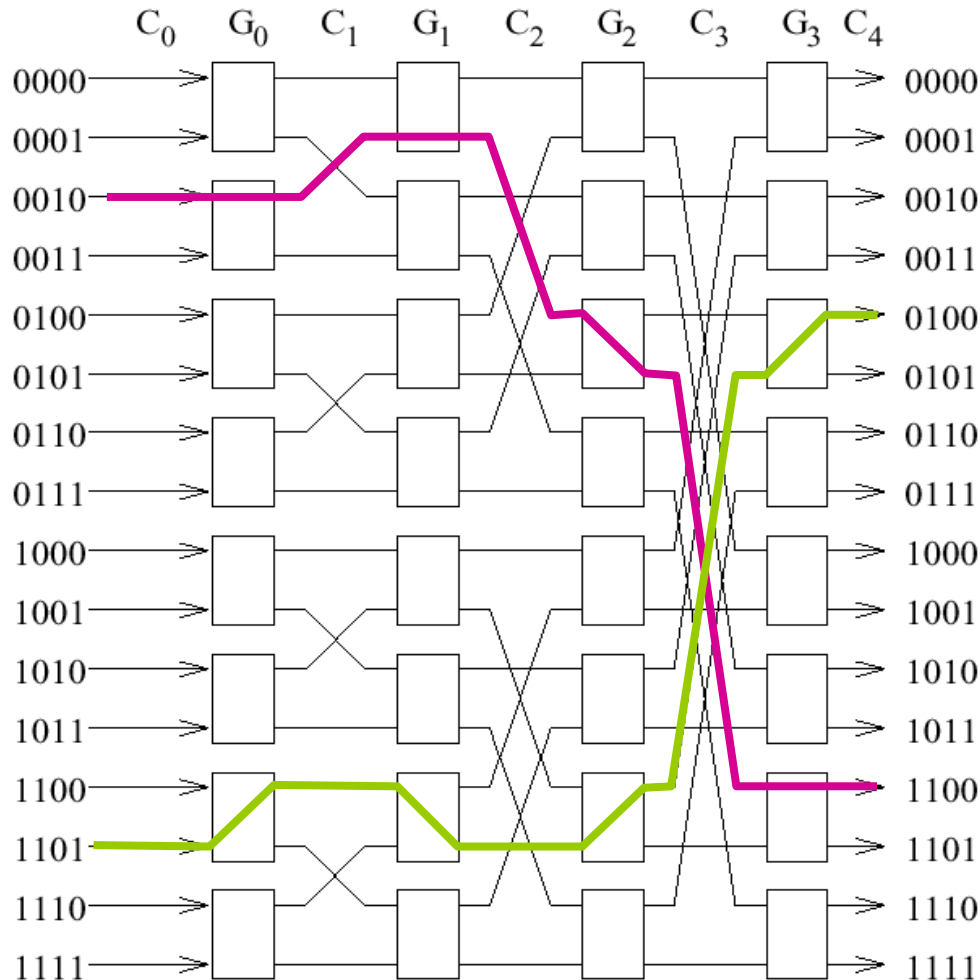
- o La dir destino es (011)
 - $d_2=0 \rightarrow$ Salida superior en etapa 0
 - $d_1=1 \rightarrow$ Salida inferior en etapa 1
 - $d_0=1 \rightarrow$ Salida inferior en etapa 2

- La configuración de los conmutadores no depende del nodo de entrada

Mecanismos de encaminamiento

□ MIN: Uso de la dirección destino

Red Butterfly



□ Funcionamiento:

- o Etapas: $n=4$
- o Dir destino: (d_3, d_2, d_1, d_0)
- o El bit d_i se usa para seleccionar el funcionamiento de los conmutadores de la etapa $i-1$ (d_n determina etapa $n-1$)
 - $d_i = 0 \rightarrow$ Salida superior
 - $d_i = 1 \rightarrow$ Salida inferior

□ Ejemplo 1: paquete de 2 a 12

- o La dir destino es (1100)
 - $d_1=0 \rightarrow$ Salida superior en etapa 0
 - $d_2=1 \rightarrow$ Salida inferior en etapa 1
 - $d_3=1 \rightarrow$ Salida inferior en etapa 2
 - $d_0=0 \rightarrow$ Salida superior en etapa 3

□ Ejemplo 2: paquete de 13 a 4

- o La dir destino es (0100)
 - $d_1=0 \rightarrow$ Salida superior en etapa 0
 - $d_2=1 \rightarrow$ Salida inferior en etapa 1
 - $d_3=0 \rightarrow$ Salida superior en etapa 2
 - $d_0=0 \rightarrow$ Salida superior en etapa 3

Interbloqueo: evitación

❑ Interbloqueo: ¿Cómo puede producirse?

- o Condiciones necesarias:
 - Recurso compartido
 - Adquirido incrementalmente
 - Sin mecanismo de expulsión
 - Los agentes en espera forman una cola circular
- o Canales como recurso compartido adquirido incrementalmente
 - Buffer fuente unido a buffer destino
 - Canales forman una ruta

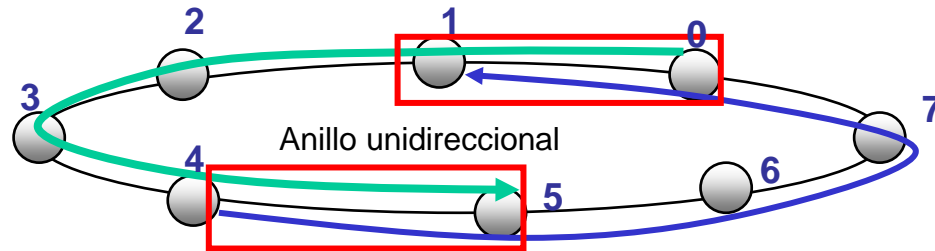
❑ ¿Cómo puede evitarse?

- o Restringiendo cómo se adquirieren los canales
 - Ejemplo: orden dimensional
- o Aportando recursos adicionales
 - Canales virtuales

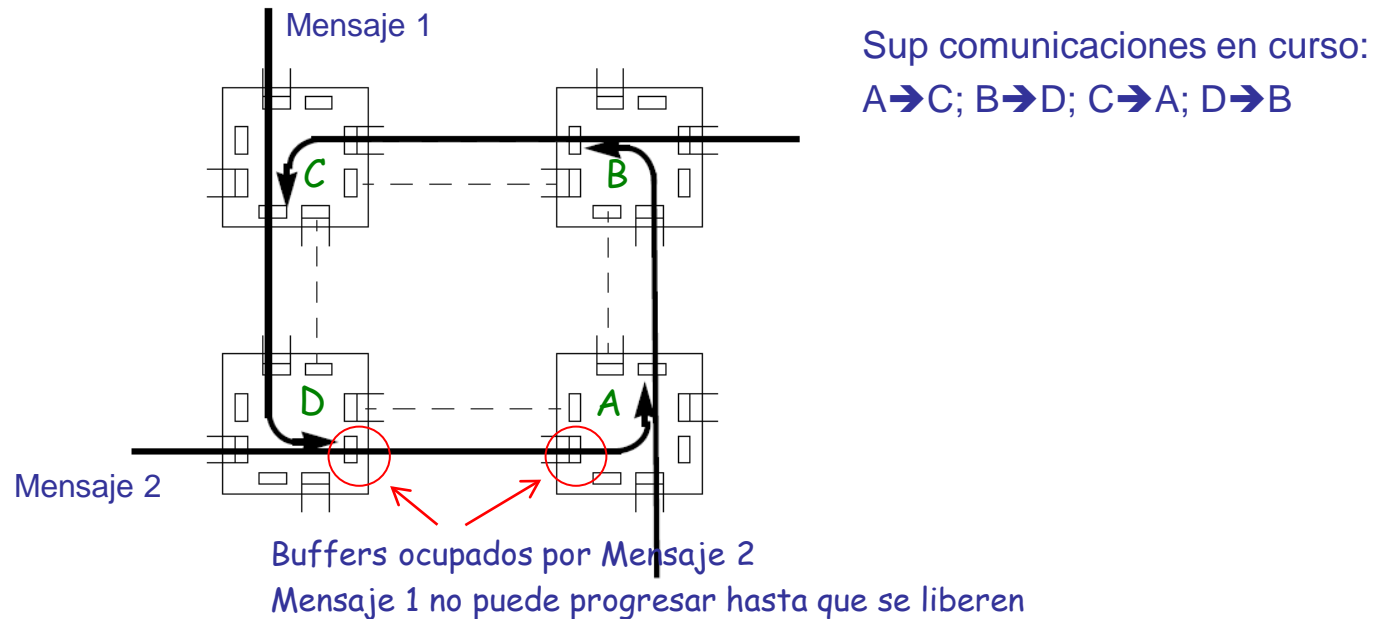
Interbloqueo: evitación

□ Ejemplos de situaciones de interbloqueo

1)



2)

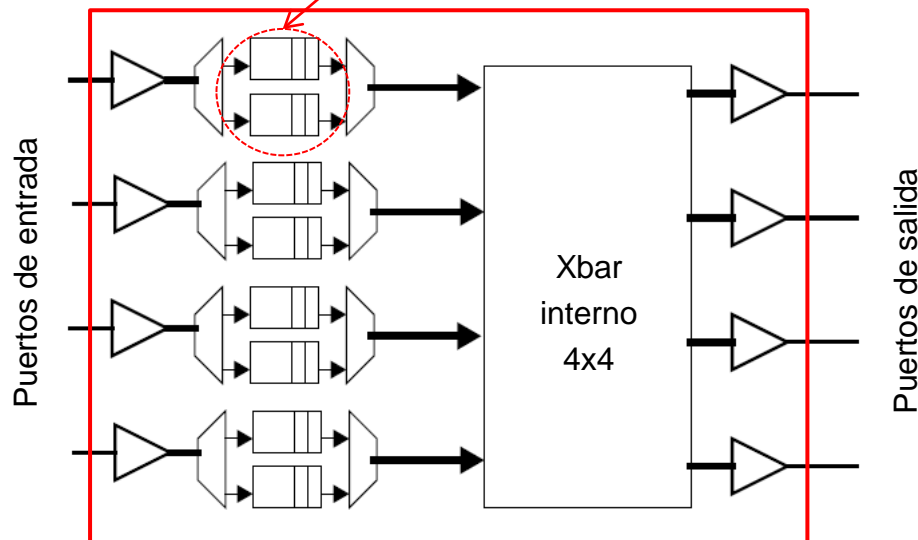


Interbloqueo: evitación

❑ Canales virtuales:

- o Mecanismo básico para construir redes libres de interbloqueos con encaminamiento worm-hole
 - Back-pressure puede implicar la reserva incremental de muchos enlaces en la red
- o Técnica: asociar con cada enlace físico más de un buffer de entrada, que se comportan como canales adicionales (virtuales)
 - Proporcionan una vía de escape en situaciones que provocarían interbloqueo
 - No aumenta el nº de enlaces en la red
 - No aumenta el tamaño del crossbar interno del conmutador: los buffers asociados a cada canal físico, están conectados al crossbar interno mediante mux

Conmutador con 2 canales virtuales por puerto de entrada



Lectura: Culler 10.5.4

Interbloqueo: evitación

❑ Canales virtuales: ejemplo malla 2D

- o Sup encaminamiento: en caso de congestión del canal "natural", se puede usar un canal alternativo.
 - Ventaja: mejor aprovechamiento de los recursos de la red
 - Problema: posibilidad de interbloqueo (ver tr. 57)
 - Solución: usar dos canales virtuales (Hi, Lo) por enlace físico
 - Un paquete que entra por Hi/Lo usa el mismo canal virtual (Hi/Lo) en el siguiente salto, excepto si hace un giro norte→oeste, en cuyo caso usa el canal virtual contrario (Lo/Hi)

Sup comunicaciones en curso:

$A \rightarrow C$; $B \rightarrow D$; $C \rightarrow A$; $D \rightarrow B$

Sin CV: interbloqueo

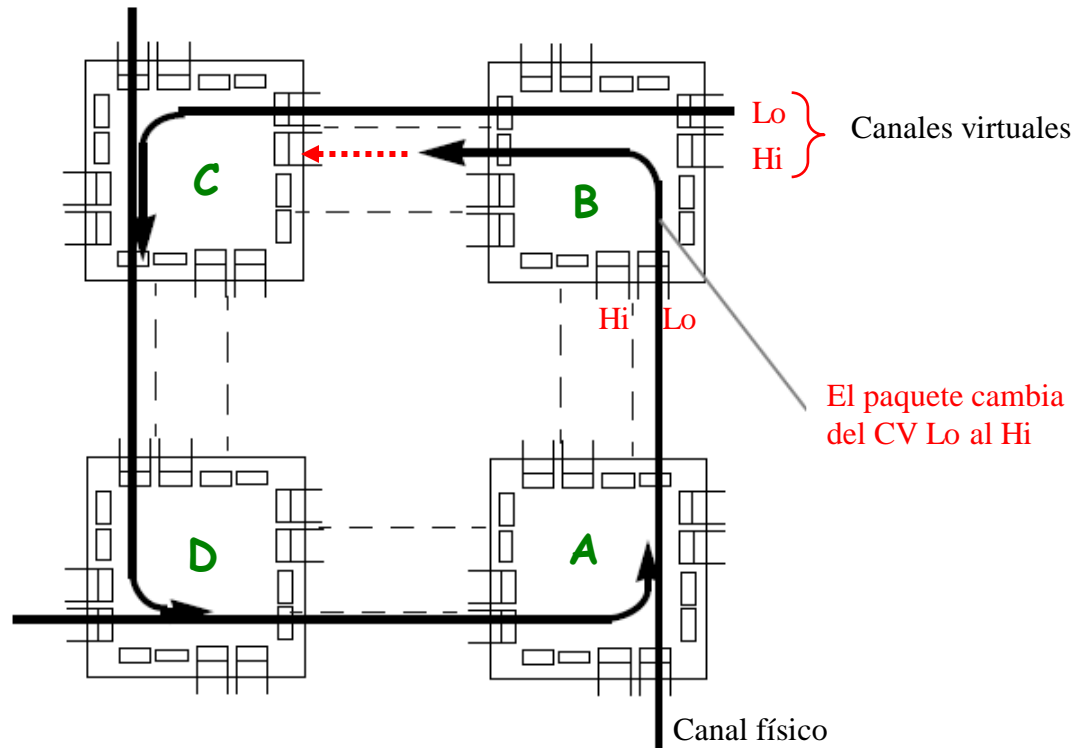
Con CV:

A alcanza C, libera A-B

D alcanza B, libera D-A

C alcanza A, libera C-D

B alcanza D, libera B-C



Ejemplos: topología, routing, conmutación en sistemas comerciales

Company	System [network] name	Max. number of nodes [\times # CPUs]	Basic network topology	Switch queuing (buffers)	Network routing algorithm	Switch arbitration technique	Network switching technique
Intel	ASCI Red Paragon	4510 [\times 2]	2D mesh (64×64)	Input buffered (1 flit)	Distributed dimension-order routing	2-phased RR, distributed across switch	Wormhole with no virtual channels
IBM	ASCI White SP Power3 [Colony]	512 [\times 16]	Bidirectional MIN with 8-port bidirectional switches (typically a fat tree or Omega)	Input and central buffer with output queuing (8-way speedup)	Source-based LCA adaptive, shortest-path routing, and table-based multicast routing	2-phased RR, centralized and distributed at outputs for bypass paths	Buffered wormhole and virtual cut-through for multicasting, no virtual channels
Intel	Thunder Itanium2 Tiger4 [QsNet ^{II}]	1024 [\times 4]	Fat tree with 8-port bidirectional switches	Input buffered	Source-based LCA adaptive, shortest-path routing	2-phased RR, priority, aging, distributed at output ports	Wormhole with 2 virtual channels
Cray	XT3 [SeaStar]	30,508 [\times 1]	3D torus ($40 \times 32 \times 24$)	Input with staging output	Distributed table-based dimension-order routing	2-phased RR, distributed at output ports	Virtual cut-through with 4 virtual channels
Cray	X1E	1024 [\times 1]	4-way bristled 2D torus ($\sim 23 \times 11$) with express links	Input with virtual output queuing	Distributed table-based dimension-order routing	2-phased wavefront (pipelined) global arbiter	Virtual cut-through with 4 virtual channels
IBM	ASC Purple pSeries 575 [Federation]	>1280 [\times 8]	Bidirectional MIN with 8-port bidirectional switches (typically a fat tree or Omega)	Input and central buffer with output queuing (8-way speedup)	Source and distributed table-based LCA adaptive, shortest-path routing, and multicast	2-phased RR, centralized and distributed at outputs for bypass paths	Buffered wormhole and virtual cut-through for multicasting with 8 virtual channels
IBM	Blue Gene/LeServer Solution [Torus Net.]	65,536 [\times 2]	3D torus ($32 \times 32 \times 64$)	Input-output buffered	Distributed, adaptive with bubble escape virtual channel	2-phased SLQ, distributed at input and output	Virtual cut-through with 4 virtual channels