

Arquitectura de Computadoras

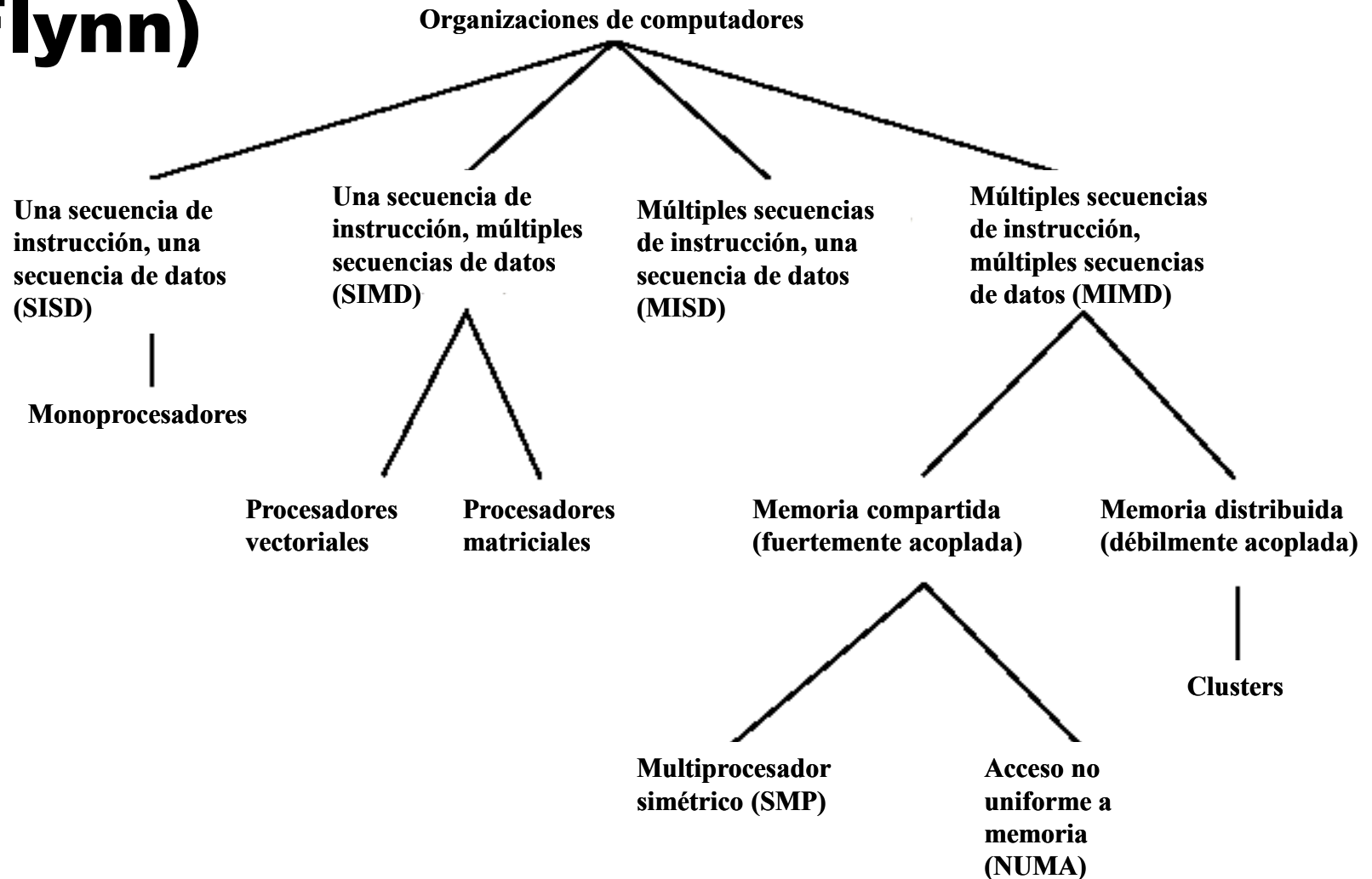
Clase 9

Procesamiento paralelo

Introducción al procesamiento paralelo

- Sea cual sea el nivel de prestaciones, la demanda de máquinas de mayor rendimiento seguirá existiendo.
 - Mejorar el rendimiento de una máquina con un solo procesador.
 - Paralelismo a nivel instrucción - ILP
 - Arquitecturas de sistemas con varios procesadores.
 - Paralelismo a nivel proceso

Taxonomía de las arquitecturas (Flynn)



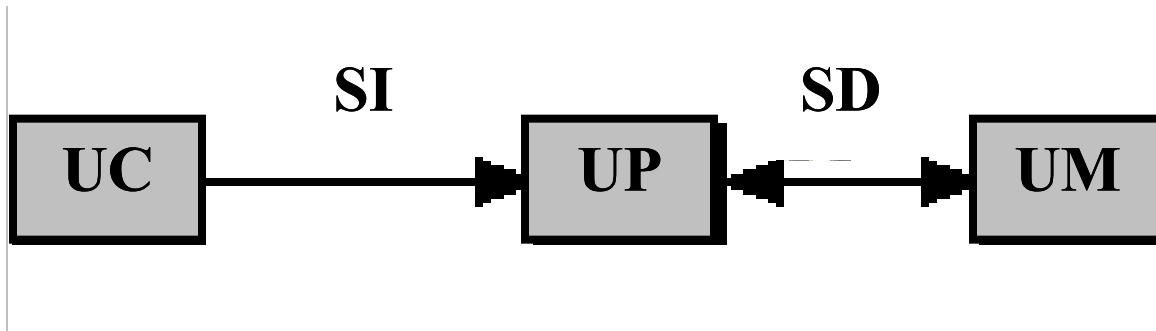
Categorías de Computadoras

- SISD
 - una secuencia de instrucciones y una secuencia de datos
- SIMD
 - una secuencia de instrucciones y múltiples secuencias de datos
- MISD
 - múltiples secuencias de instrucciones y una secuencia de datos
- MIMD
 - múltiples secuencias de instrucciones y múltiples secuencias de datos

SISD

- Un único procesador interpreta una única secuencia de instrucciones (SI).
- Datos almacenados en una única memoria(UM)
- Computadoras monoprocesador.

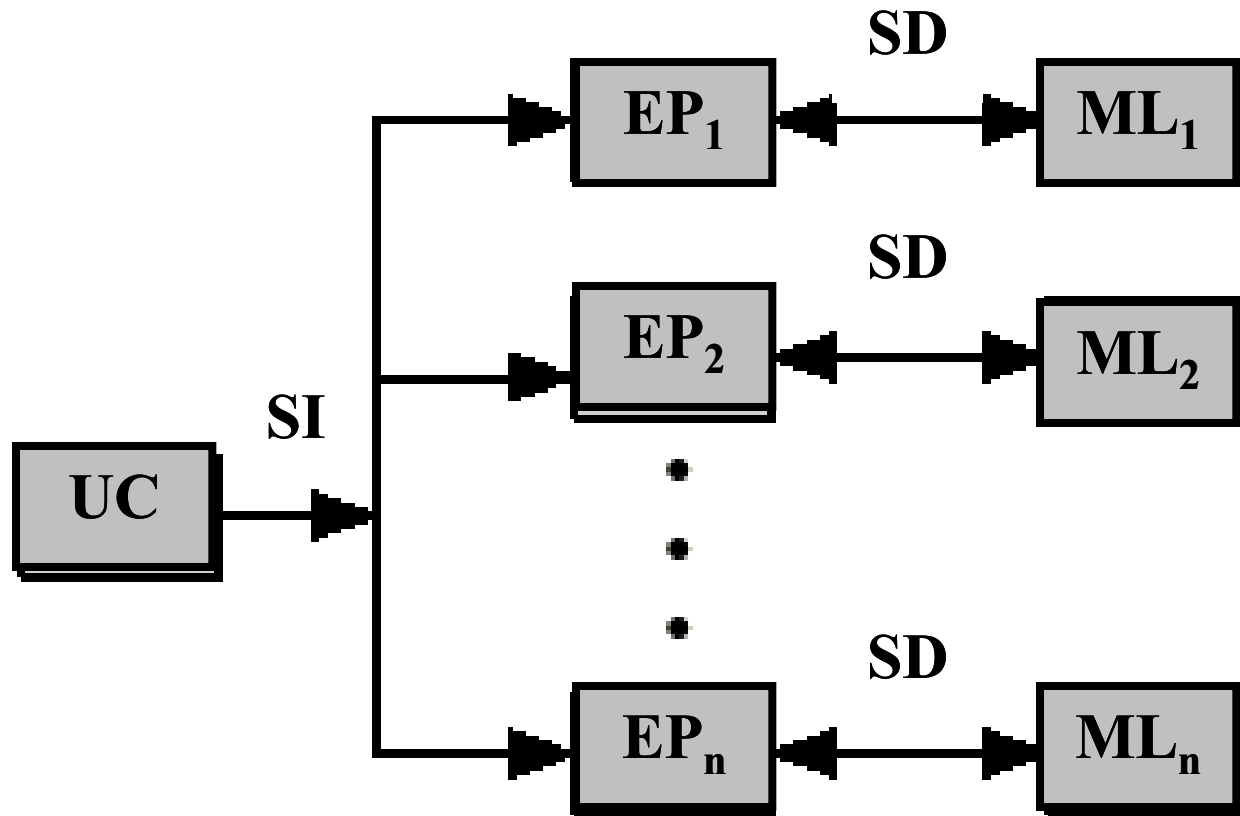
Organización SISD



SIMD

- Una única instrucción máquina controla paso a paso la ejecución simultánea de un cierto número de elementos de proceso (EP).
- Cada elemento de proceso tiene una memoria dedicada (ML).
- Cada instrucción es ejecutada por cada procesador, con un conjunto de datos diferentes.
- Procesadores vectoriales y matriciales.

Organización SIMD



Matricial?, vectorial?, paralelo?

- Computadora con una única “unidad de control” y una matriz de elementos “computacionales”.
- Tipos de instrucciones de procesador:
 - Extensiones de las instrucciones escalares:
 - Sumar, almacenar, multiplicar, etc. se convierten en operaciones vectoriales ejecutadas en todos los procesadores de modo simultáneo
 - Debe añadirse la capacidad de transferir al conjunto de instrucciones los datos escalares y vectoriales entre procesadores: atributos de un “lenguaje paralelo”.

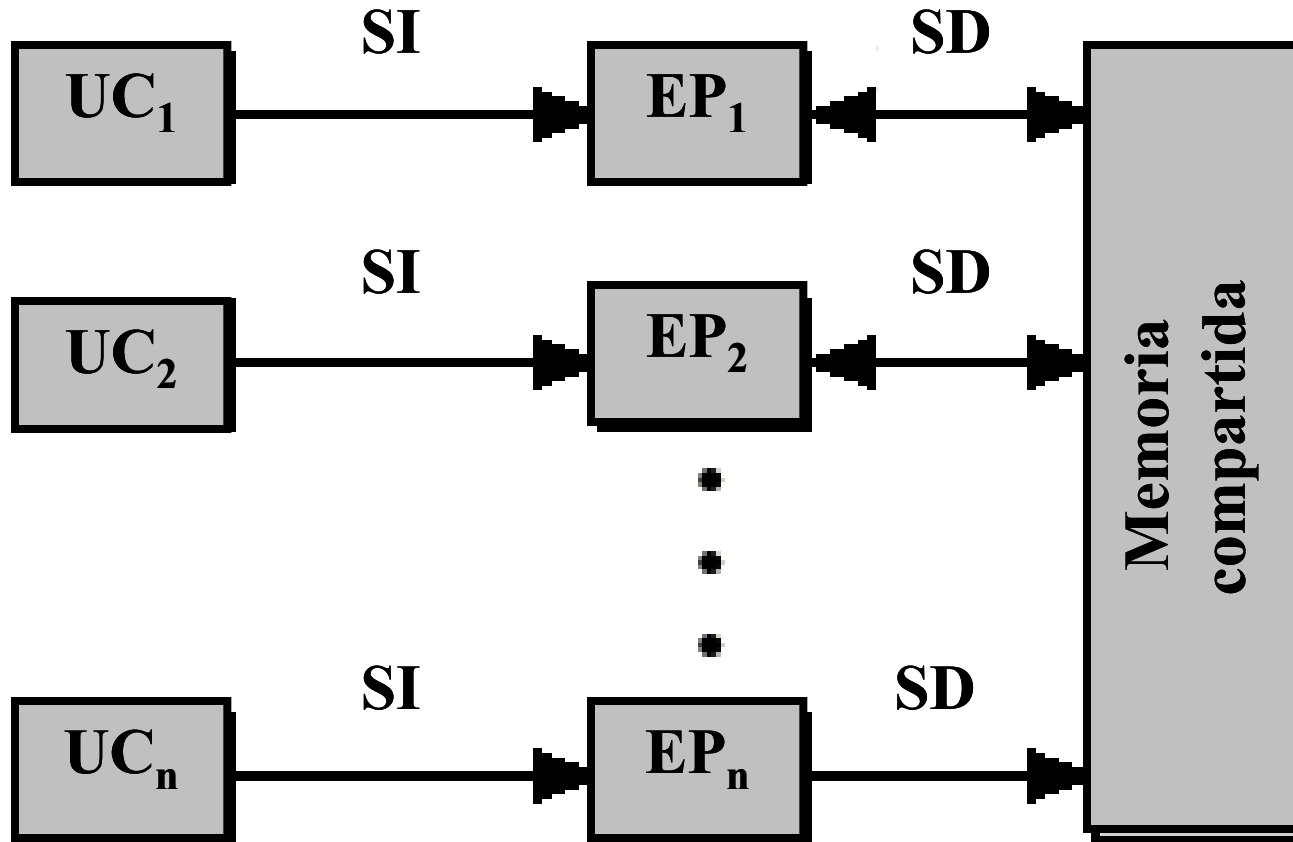
MISD

- Se transmite una secuencia de datos a un conjunto de procesadores.
- Cada procesador ejecuta una secuencia de instrucciones diferente.
- Esta estructura nunca ha sido implementada.

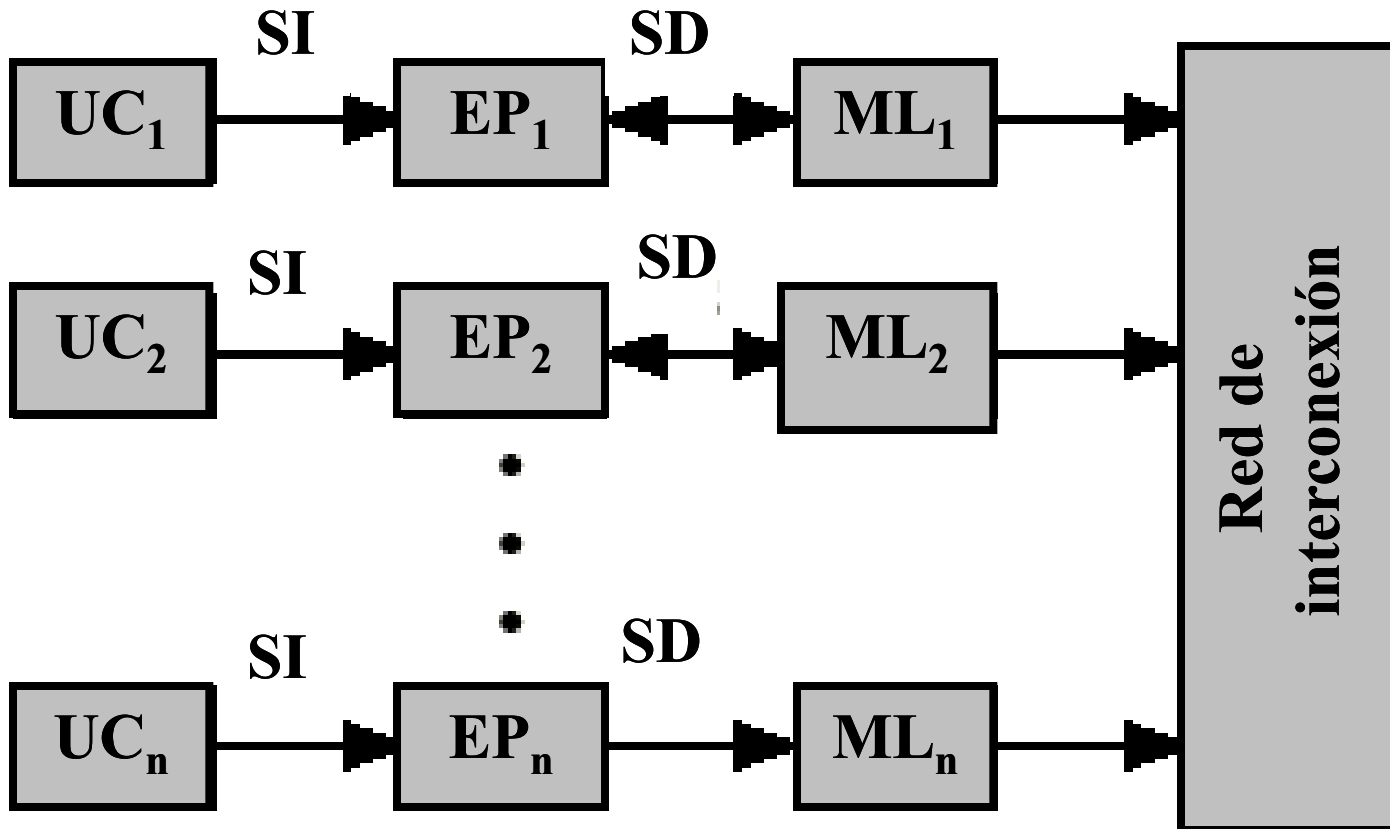
MIMD

- Un conjunto de procesadores ejecuta secuencias de instrucciones diferentes en simultáneo.
- Conjuntos de datos diferentes.
- Se pueden dividir según la forma de comunicarse
 - Memoria compartida
 - SMP (multiprocesadores simétricos) y sistemas NUMA
 - Memoria distribuida
 - Clusters

Organización MIMD de memoria compartida



Organización MIMD de memoria distribuida



Multiprocesador simétrico - SMP

- Computadora autónoma con las siguientes características:
 - Dos o más procesadores similares de capacidades comparables
 - Comparten la memoria principal y las E/S.
 - Interconectados mediante un bus u otro tipo de sistema de interconexión.
 - Tiempo de acceso a memoria similar para todos los procesadores (UMA).
 - Todos los procesadores pueden desempeñar las mismas funciones.
 - Sistema operativo integrado, que proporciona la interacción entre los procesadores y sus programas.

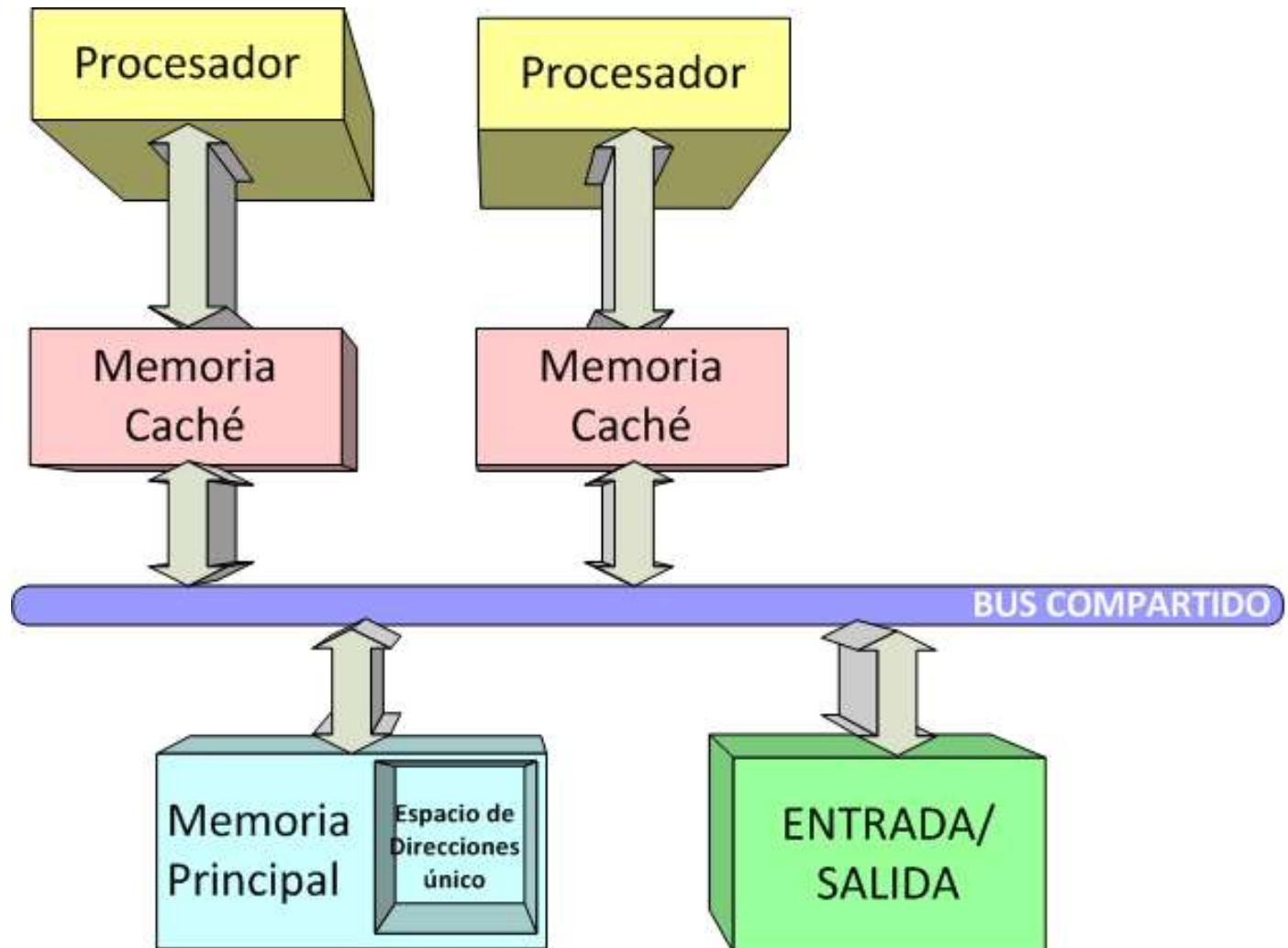
Ventajas potenciales de un SMP

- Mayores prestaciones
 - si el trabajo a realizar puede organizarse en paralelo.
- Buena disponibilidad
 - un fallo en un procesador no detendrá la computadora
- Crecimiento incremental
 - Se pueden añadir más procesadores.
- Escalado
 - En función de la cantidad de procesadores
- Cuidado: Bus compartido

Bus de tiempo compartido (desventajas)

- La prestación está limitada por el tiempo de ciclo del bus.
- Cada procesador debería estar equipado con una memoria cache para mejorar las prestaciones
 - Se reduciría el número de accesos.
- Se pueden producir problemas de coherencia de cache
 - Este problema debe ser resuelto por el hardware
 - Protocolos de sondeo y protocolos de directorio.

Arquitectura de un SMP



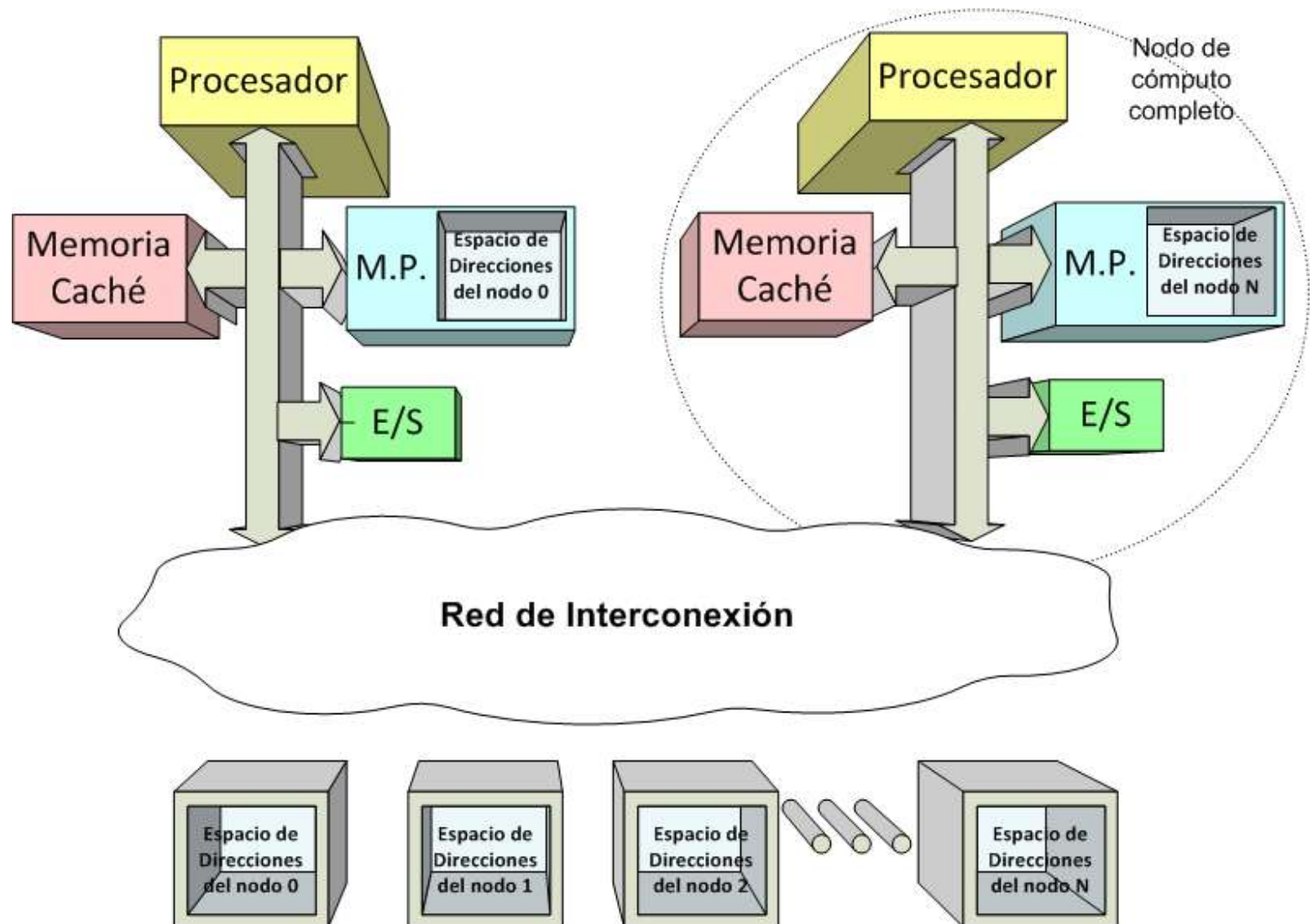
“Clusters”

- Computadoras completas interconectadas que trabajan conjuntamente como un único recurso
 - ilusión de que se trata de una única máquina.
- Cada computadora se denomina “nodo”.
- Prestaciones y disponibilidad elevadas.
- Aplicaciones propias de un servidor.
- Son la alternativa a los SMP.

Beneficios del “cluster”

- Escalabilidad absoluta.
- Escalabilidad incremental.
- Alta disponibilidad.
- Mejor relación precio/prestaciones.

Arquitectura de un cluster



Cluster vs SMP

- Ambos:
 - dan soporte a aplicaciones de alta demanda de recursos
 - disponibles comercialmente (SMP es mas antiguo)
- SMP:
 - Mas fácil de administrar y configurar
 - Cercano a los sistemas de un solo procesador
 - La planificación (scheduling) es la diferencia principal
 - Menos espacio físico / Menor consumo de potencia
- Cluster:
 - Superior escalabilidad incremental y absoluta
 - Superior disponibilidad
 - Redundancia

Términos UMA, NUMA, CC-NUMA

Todos los procesadores tienen acceso a toda la memoria

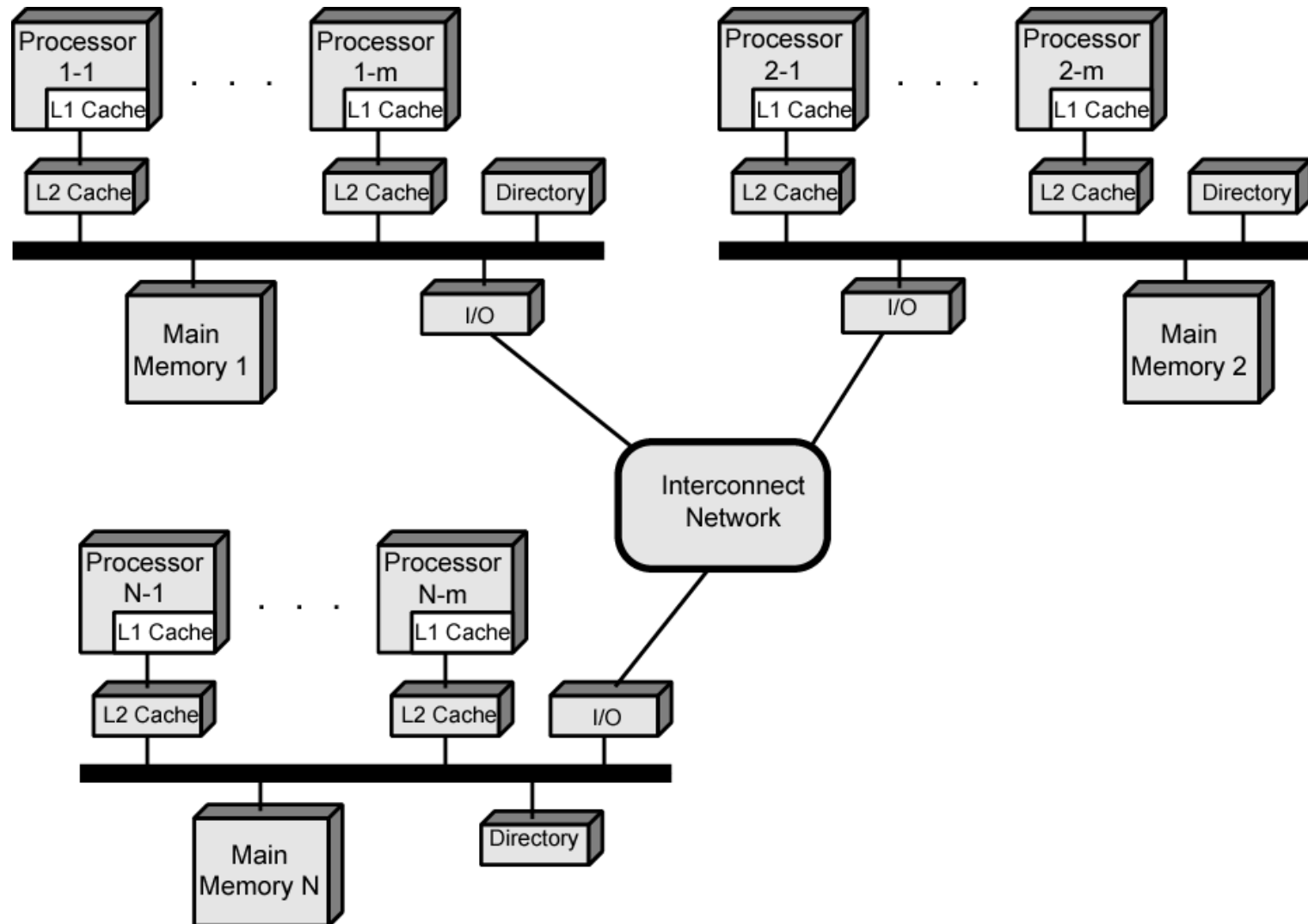
- Usan 'load' y 'store'
- UMA - Uniform memory access
 - Igual tiempo de acceso a todas las regiones de memoria
 - Igual tiempo de acceso a memoria para los diferentes procesadores
- NUMA - Nonuniform memory access
 - EL tiempo de acceso de un procesador difiere dependiendo de la región de memoria que accede
 - Diferentes procesadores acceden a diferentes regiones de memoria a diferentes velocidades
- CC-NUMA - cache coherente NUMA
 - Es un NUMA que mantiene coherencia de cache entre las cache de los distintos procesadores

Motivación NUMA

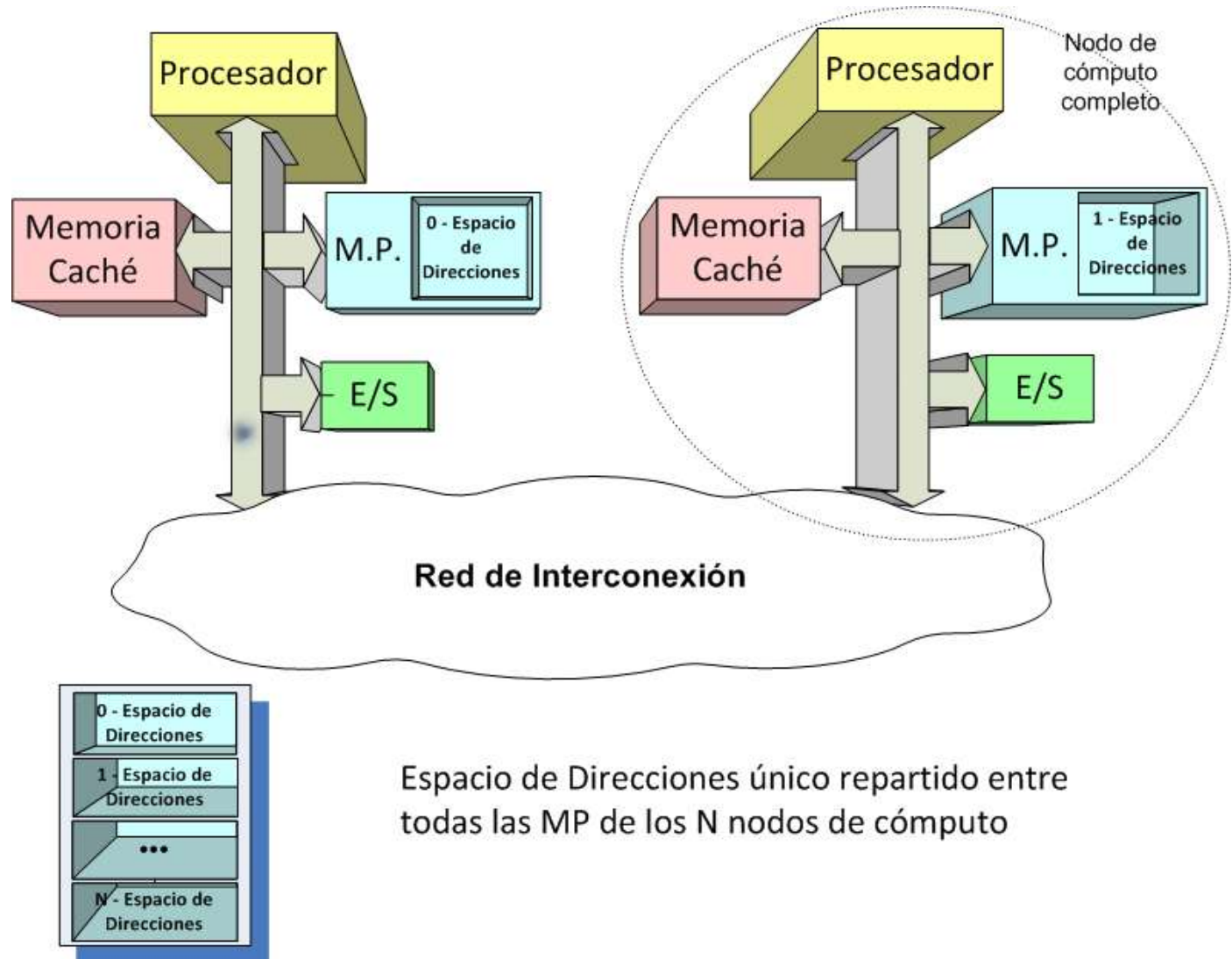
- SMP tiene límite práctico en su número de procesadores
 - entre 16 y 64 por degradación de prestaciones
- En clusters cada nodo tiene su propia memoria principal
 - Aplicaciones no 'ven' la memoria global
 - Coherencia de cache mantenida por software no por hardware
- NUMA retiene las características tipo SMP y brinda multi-procesamiento a gran escala
 - ej. SGI Origin de Silicon Graphics es NUMA con 1024 MIPS R10000

Objetivo NUMA: tener una memoria transparente del sistema y permitir nodos, cada uno con su propio bus o sistema de conexión interna

Organización CC-NUMA



Arquitectura de memoria compartida-distribuida



Operación CC-NUMA

- Cada procesador tiene cache L1 y L2
- Cada nodo tiene su propia memoria principal
- Nodos conectados por algún tipo de red
- Cada procesador 've' un único espacio de direcciones de memoria
- Orden de acceso a memoria:
 - cache L1 (local al procesador)
 - cache L2 (local al procesador)
 - Memoria principal (local al nodo)
 - Memoria remota
 - Petición por red
- Automático y transparente

Procesamiento Multihebra (Multithreading)

- Aumento de paralelismo de instrucciones
 - Sin el aumento de complejidad y consumo de potencia de la segmentación de cauce y los superescalares
- La secuencia de instrucciones se divide en secuencias más pequeñas llamadas hebras (threads) que pueden ejecutarse en paralelo
- Amplia variedad de diseños multihebra.

Términos: Hebra y Proceso

- Concepto de Hebra en procesadores multihebra puede no ser el de S.O. multiprogramados.
- Proceso
 - Un programa 'corriendo' en una computadora
 - Propiedad de Recursos
 - Espacio de direcciones virtuales para almacenar la imagen de un proceso (code, data, stack, etc)
 - Planificación/ejecución
 - Hay camino de ejecución (traza)
- Conmutación de Proceso (process switch)

Términos: Hebra y Procesos (2)

- Hebra (thread)
 - Unidad de trabajo de un proceso que puede asignarse
 - Incluye un contexto de procesador (incluido PC y SP) y área de datos para su pila (stack)
 - Se ejecuta secuencialmente.
 - Interrumpible. El procesador cambiaría a otra hebra
- Conmutación de hebra (thread switch)
 - Cambio de control del procesador entre hebras de un mismo proceso
 - Usualmente menos costosa que la conmutación de proceso

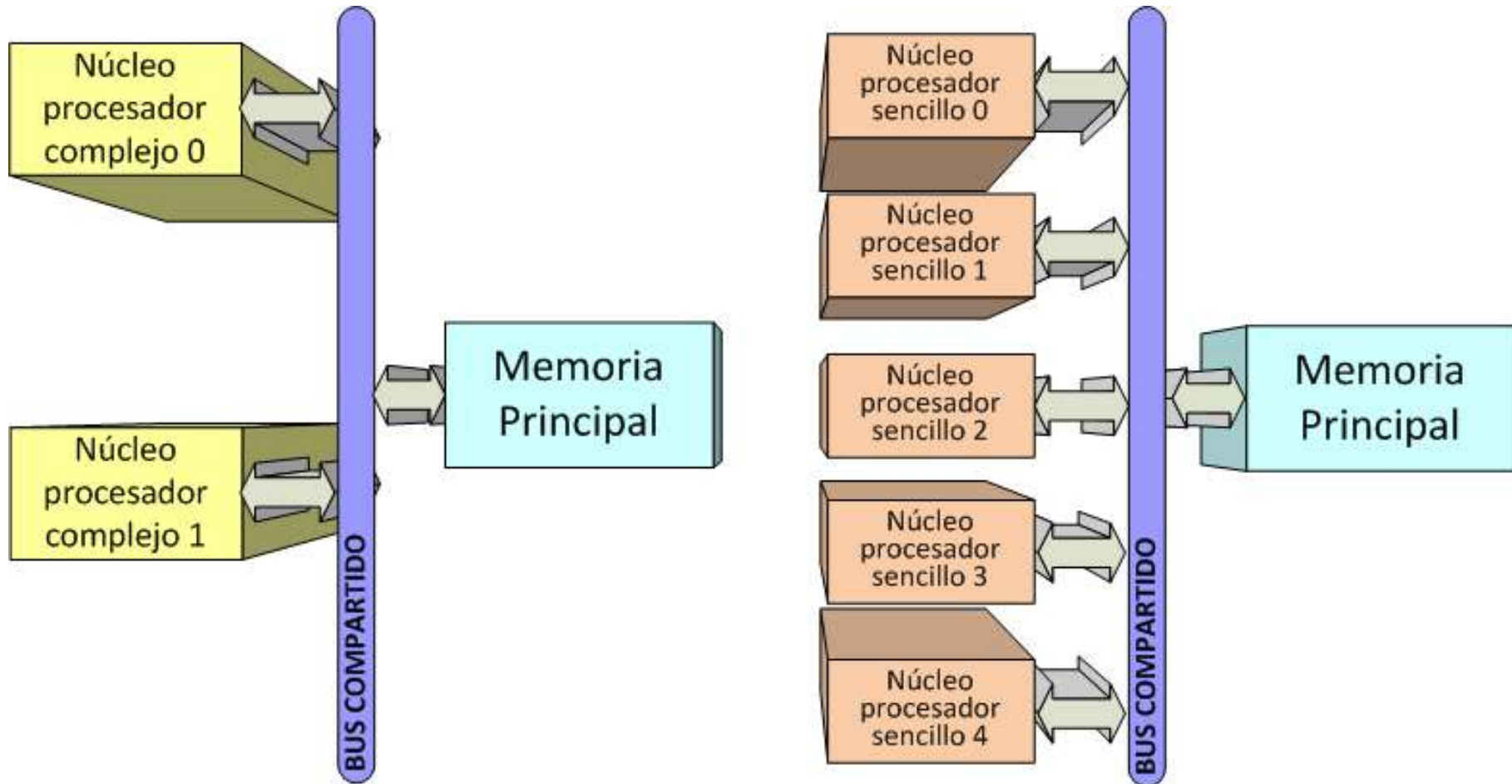
Multihebra implícito y explícito

- Multihebra explícito
 - Ejecución concurrente de instrucciones de diferentes hebras explícitas
 - Mezcla de instrucciones de diferentes hebras en cauces compartidos
 - Ó por ejecución paralela en cauces paralelos
 - Todos los procesadores comerciales lo usan
- Multihebra implícito
 - Ejecución concurrente de varias hebras extraídas de un único programa secuencial.
 - Definidas estáticamente por el compilador ó dinámicamente por el hardware

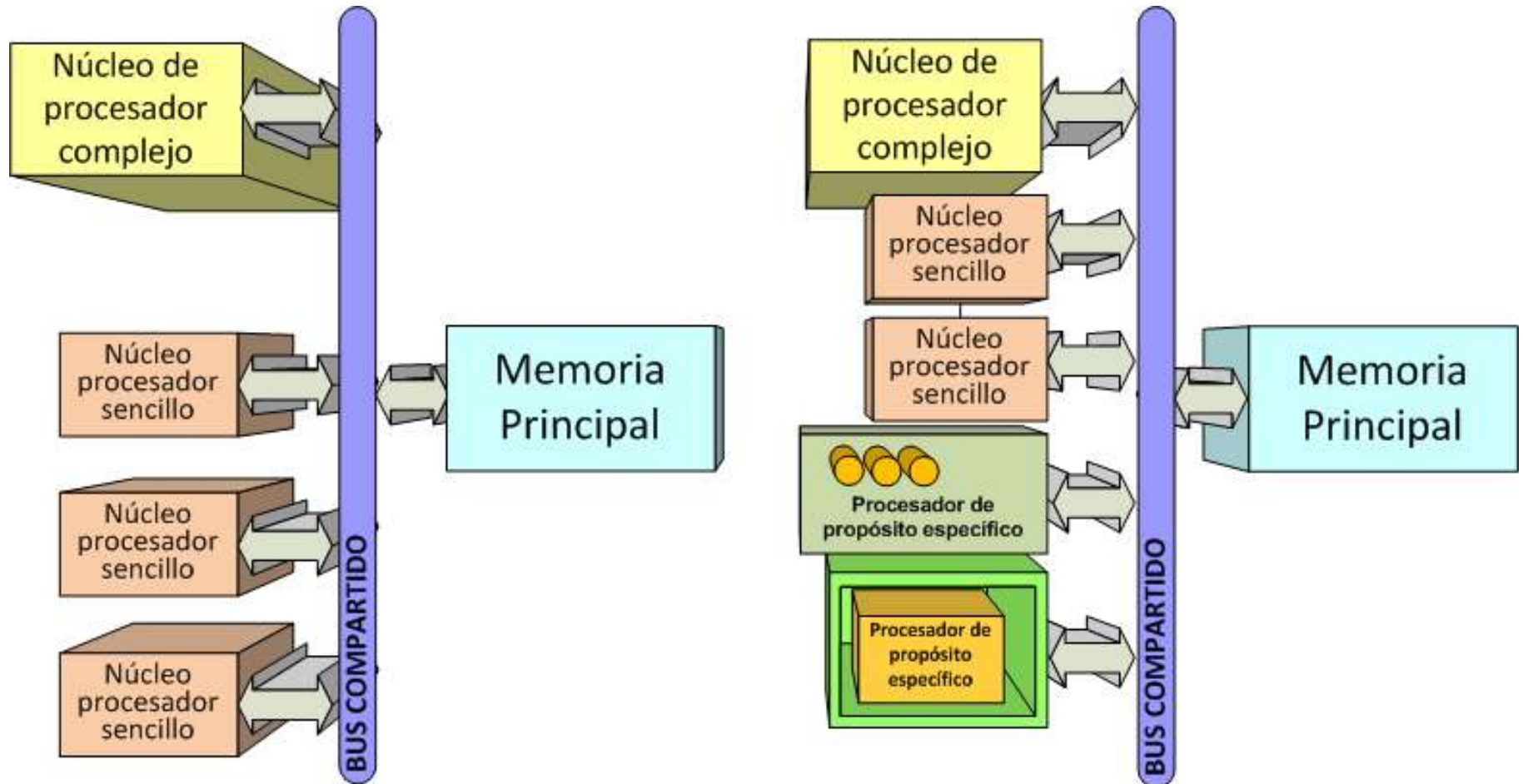
Procesador multihebra

- PC (contador de programa) distinto para cada hebra que pueda ejecutarse concurrentemente.
 - Cantidad y tipo de HW para ejecución concurrente
- Se trata cada hebra separadamente
 - Predicción de saltos, renombre de registros y etc para optimizar ejecución.
- Paralelismo entre hebras
- Aproximaciones con ejecución simultánea real
 - Multihebra simultánea (SMT) – Pentium 4 HT
 - Multiprocesador monochip

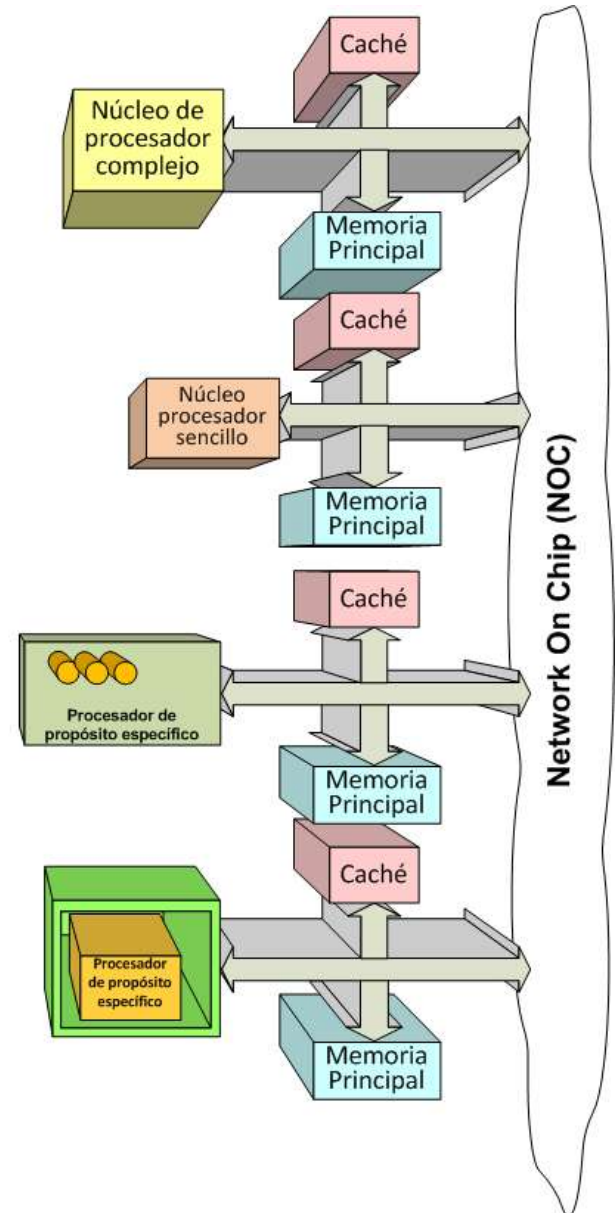
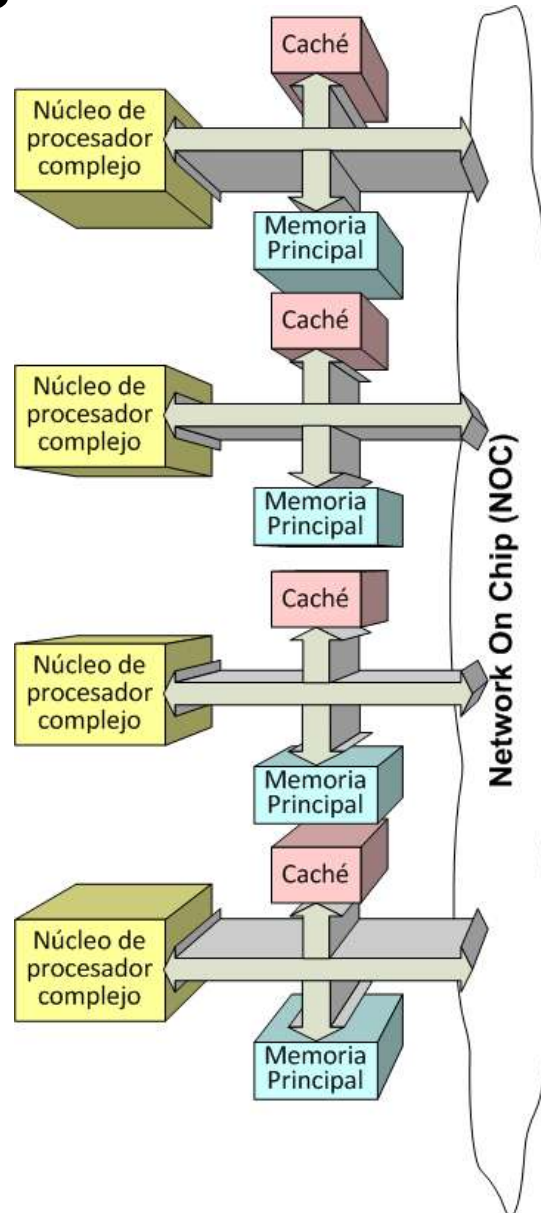
Arquitecturas on chip (memoria compartida)



Arquitecturas on chip (memoria compartida) (2)



Arquitecturas on chip (memoria distribuida) (3)



Lecturas recomendadas

- *Organización y Arquitectura de Computadoras*, William Stallings, Capítulo 16 de 5^{ta} edición ó Capítulo 18 de 7^{ma} edición.
- *Diseño y evaluación de arquitecturas de computadoras*, M. Beltrán y A. Guzmán, Capítulo 5 de 1^{ra} edición.