

2º curso / 2º cuatr.  
Grado en  
Ing. Informática

# Arquitectura de Computadores

## Tema 3

# Arquitecturas con paralelismo a nivel de thread (TLP)

Material elaborado por los profesores responsables de la asignatura:  
Mancia Anguita – Julio Ortega

*Licencia Creative Commons*



ugr

Universidad  
de Granada

**ETSIIT**  
Escuela Técnica Superior  
de Ingenierías Informática  
y de Telecomunicación



**ATC**  
Departamento de Arquitectura  
y Tecnología de Computadores  
UNIVERSIDAD DE GRANADA



# Lecciones

- Lección 7. Arquitecturas TLP
  - Clasificación y estructura de arquitecturas con TLP explícito y una instancia del SO
  - Multiprocesadores
  - Multicores
  - Cores Multithread
  - Hardware y arquitecturas TLP en un chip
- Lección 8. Coherencia del sistema de memoria
- Lección 9. Consistencia del sistema de memoria
- Lección 10. Sincronización

# Objetivos Lección 7

- Distinguir entre cores multithread, multicores y multiprocesadores.
- Comparar entre cores multithread de grano fino, cores multithread de grano grueso y cores con multithread simultánea.

# Bibliografía Lección 7

## ➤ Fundamental


- T. Rauber, G. R nder. *Parallel Programming: for Multicore and Cluster Systems*. Springer 2010. Disponible en l nea (biblioteca UGR): <http://dx.doi.org/10.1007/978-3-642-04818-0>
- Theo Ungerer, Borut Robic, Jurij Silc, “A survey of Processors with Explicit Multithreading”, ACM Computing Surveys, vol. 35, no. 1, pp. 29-63, March 2003. Disponible en l nea (biblioteca UGR): <http://dl.acm.org/citation.cfm?id=641867>
- Cap tulo 3, Secci n 2. M. Anguita, J. Ortega. Fundamentos y problemas de Arquitectura de Computadores, Editorial T cnica Avicam. ESIIT/C.1 ANG fun

# Contenido Lección 7

- Clasificación y estructura de arquitecturas con TLP explícito y una instancia del SO
- Multiprocesadores
- Multicores
- Cores Multithread
- Hardware y arquitecturas TLP en un chip

# Clasificación de arquitecturas con TLP explícito y una instancia de SO

## ➤ Multiprocesador

- Ejecutan varios threads en paralelo en un **computador** con varios cores/procesadores (cada thread en un core/procesador distinto). 
- Diversos niveles de empaquetamiento: dado, encapsulado, placa, chasis y sistema.

## ➤ Multicore o multiprocesador en un chip o CMP (*Chip MultiProcessor*)

- Ejecutan varios threads en paralelo en un **chip de procesamiento** multicore (cada thread en un core distinto)

## ➤ Core multithread

- **Core** que modifica su arquitectura ILP para ejecutar threads concurrentemente o en paralelo

# Contenido Lección 7

- Clasificación y estructura de arquitecturas con TLP explícito y una instancia del SO
- Multiprocesadores
  - Ejecutan varios threads en paralelo en un **computador** con varios cores/procesadores (cada thread en un core/procesador distinto).
- Multicores
- Cores Multithread
- Hardware y arquitecturas TLP en un chip

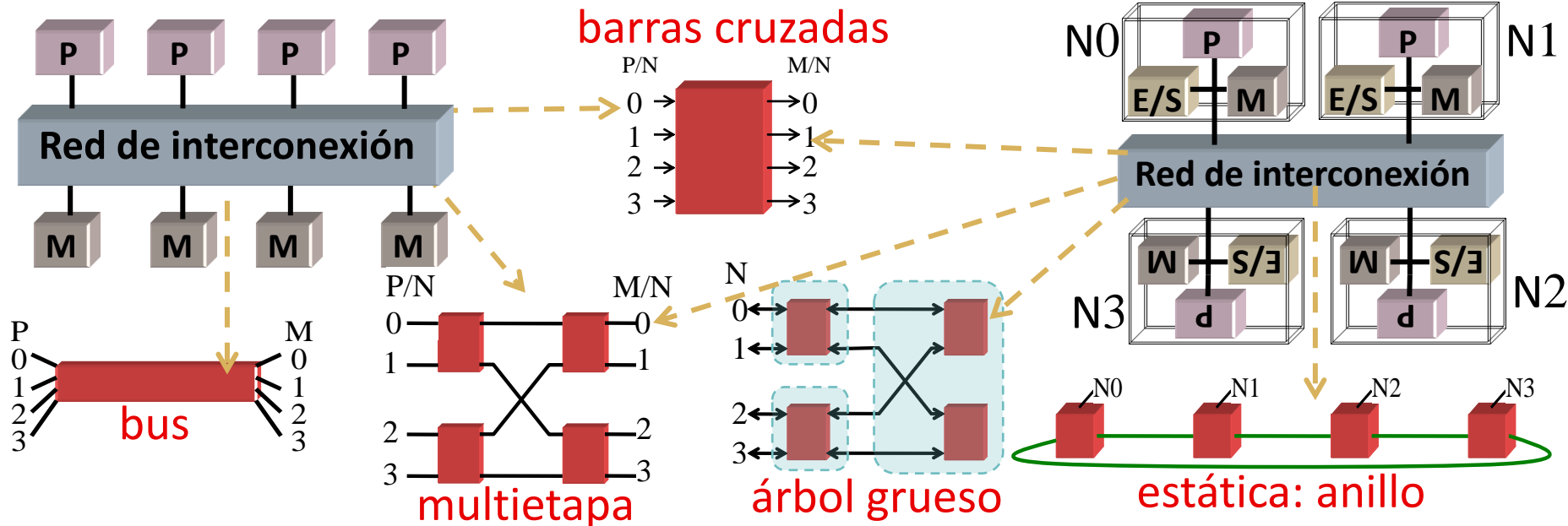
# Multiprocesadores. Criterio clasificación: sistema de memoria (Lección 1)

## Multiprocesador con memoria centralizada (UMA)

- Mayor latencia - Poco escalable

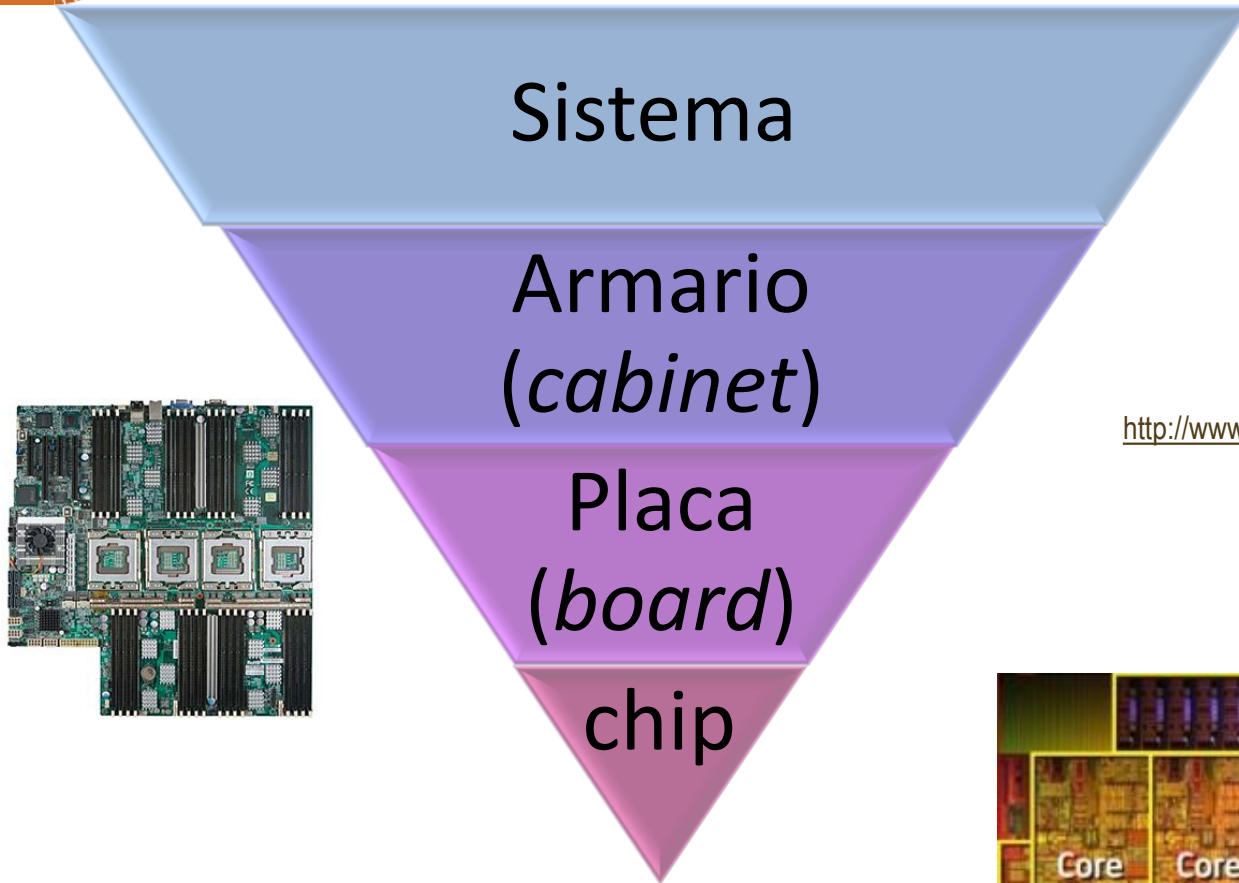
## Multiprocesador con memoria distribuida (NUMA)

- Menor latencia - escalable pero requiere para ello distribución de datos/código





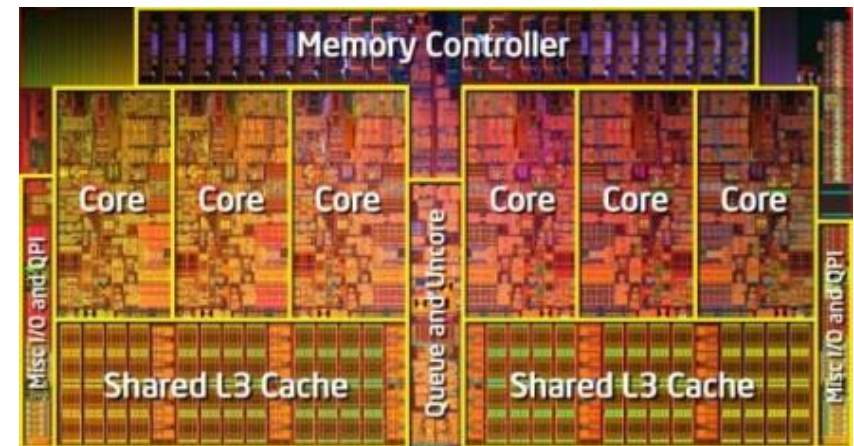
# Multiprocesadores. Criterio de clasificación: nivel de empaquet./conexión



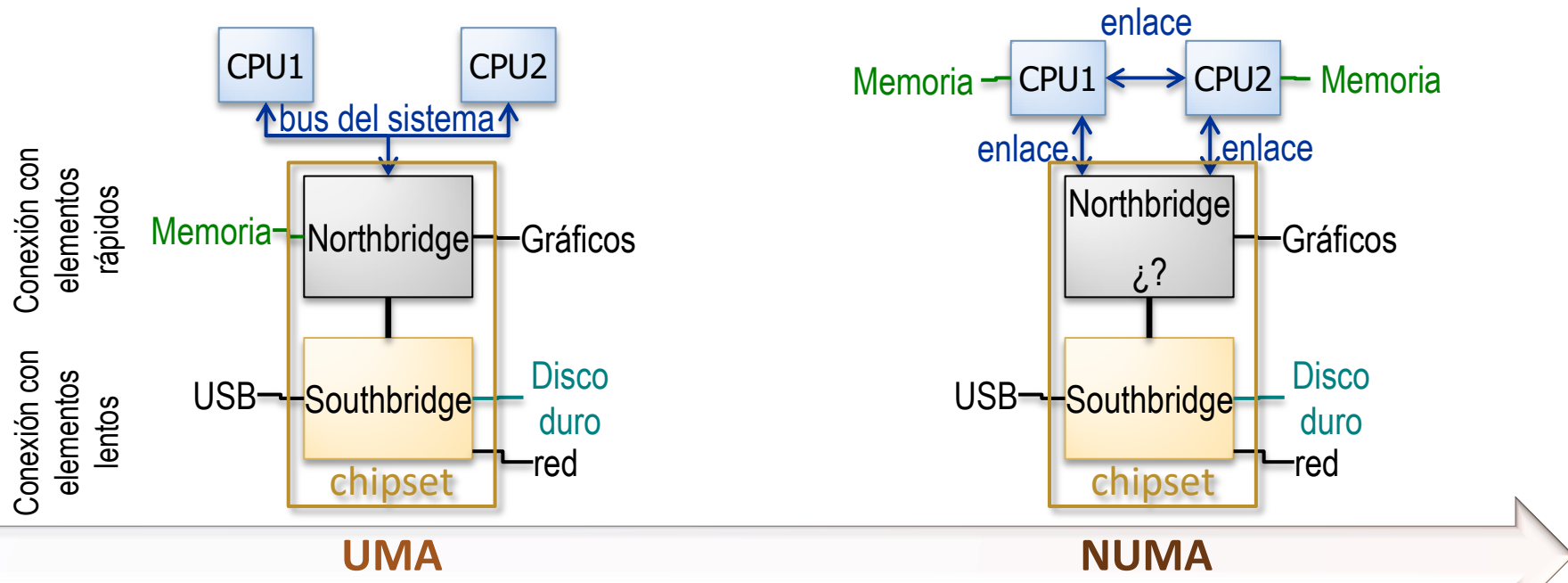
SGI Altix 4700

<http://www.sgi.com/products/remarketed/servers/altix4700.html>

Multicore

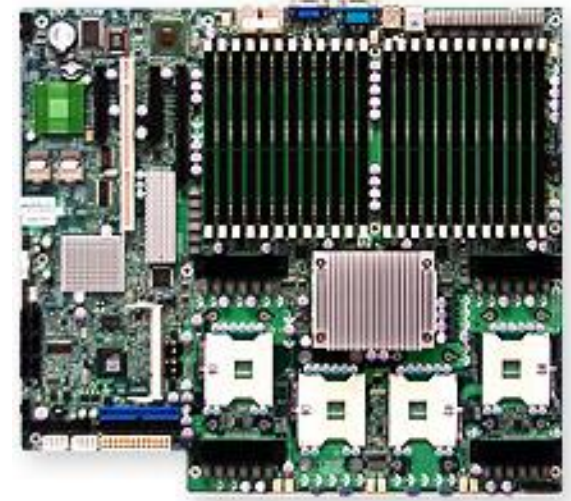
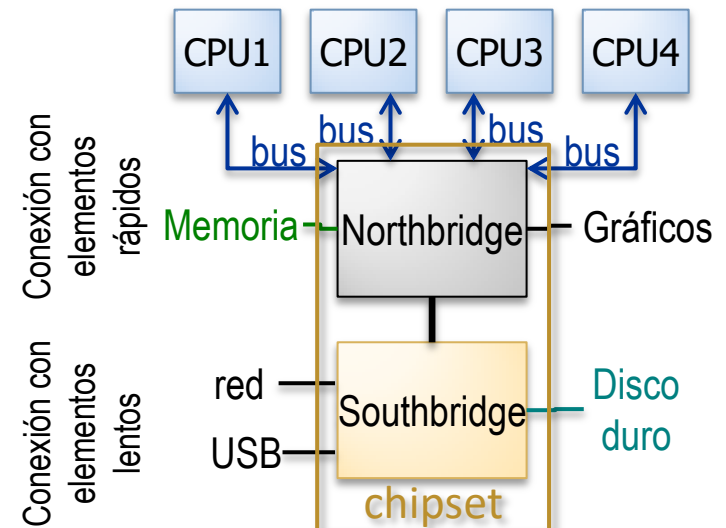
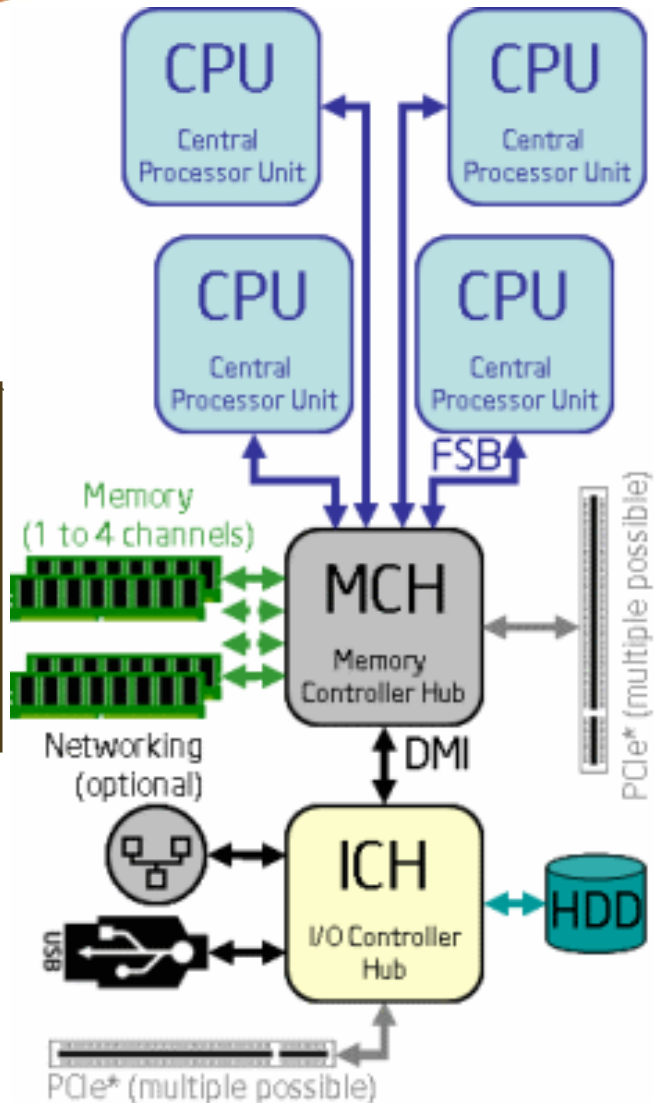


# Multiprocesador en una placa: evolución de UMA a NUMA



- ❖ Controlador de memoria en **chipset** (*Northbridge* chip)
- ❖ Red: bus (medio compartido)
- ❖ Controlador de memoria en chip del procesador
- ❖ Red: enlaces (conexiones punto a punto) y conmutadores (en el chip del procesador)
- ❖ Ejemplos en servidores:
  - AMD Opteron (2003): enlaces HyperTransport (2001)
  - Intel (Nehalem) Xeon 7500 (2010): enlaces QPI (*Quick Path Interconnect*, 2008)

# Multiprocesador en una placa: UMA con bus (Intel Xeon 7300)



# Multiprocesador en una placa: CC-NUMA con red estática (Intel Xeon 7500)

[http://ark.intel.com/products/46490/Intel-Xeon-Processor-E7520-18M-Cache-1\\_86-GHz-4\\_80-GTs-Intel-QPI](http://ark.intel.com/products/46490/Intel-Xeon-Processor-E7520-18M-Cache-1_86-GHz-4_80-GTs-Intel-QPI)

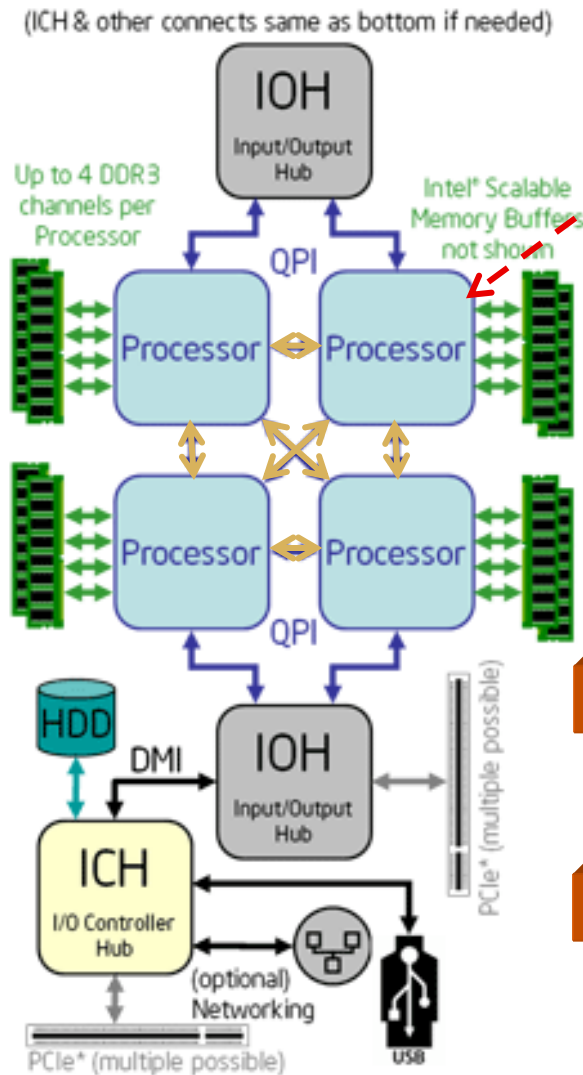
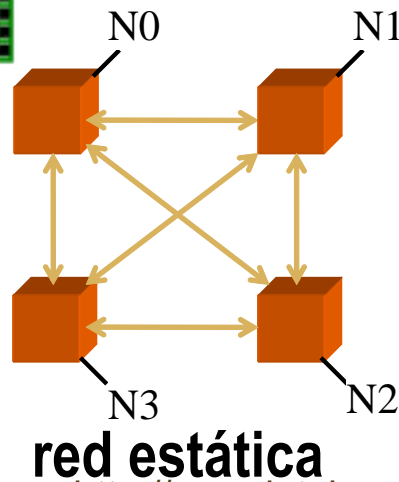


Diagrama de bloques de la placa



<http://www.intel.com/technology/quickpath/demo/demo.htm>



Placa para Intel Xeon 7500

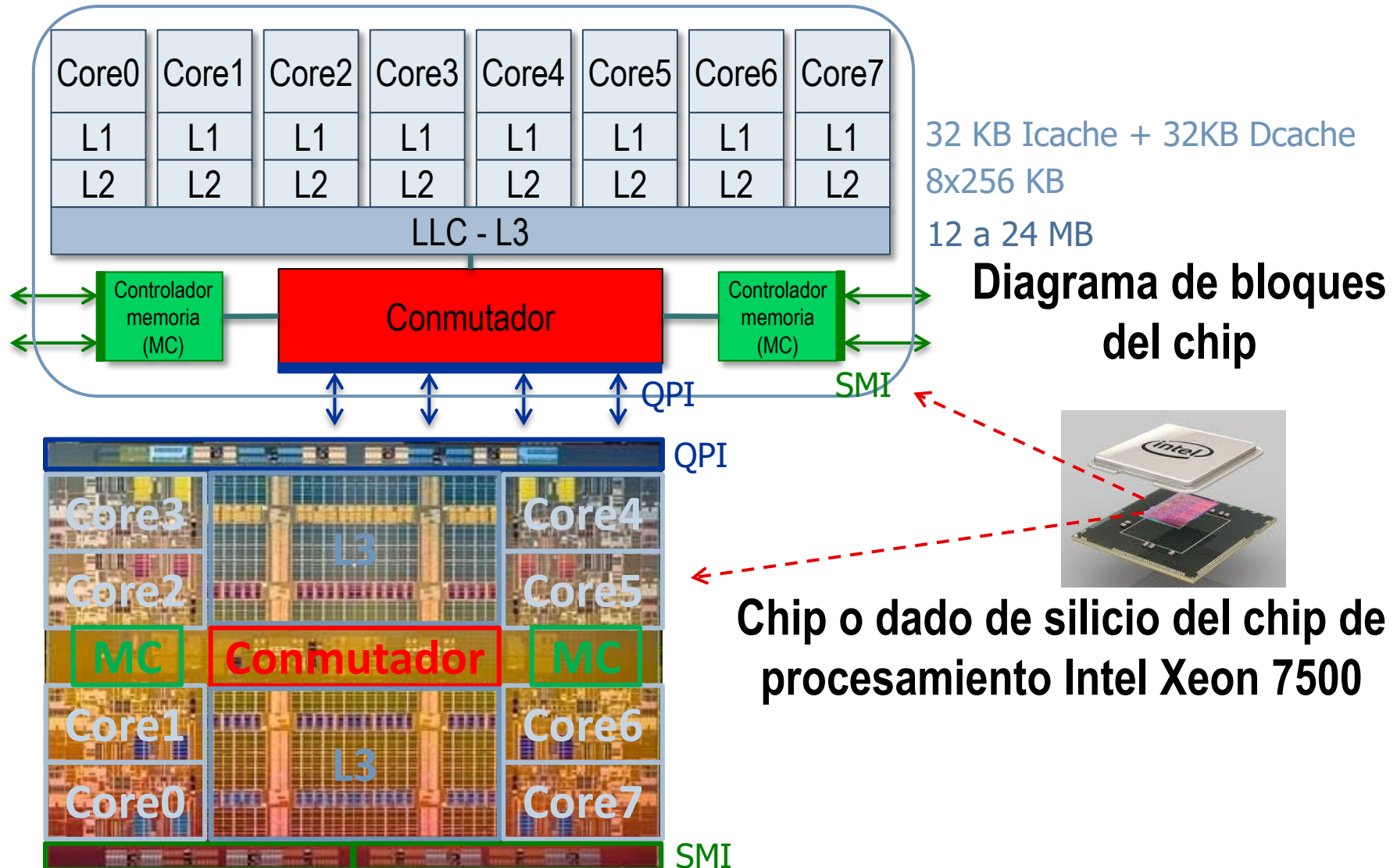
<http://www.supermicro.es/?opcion=contenido&plt=Xeon7500>



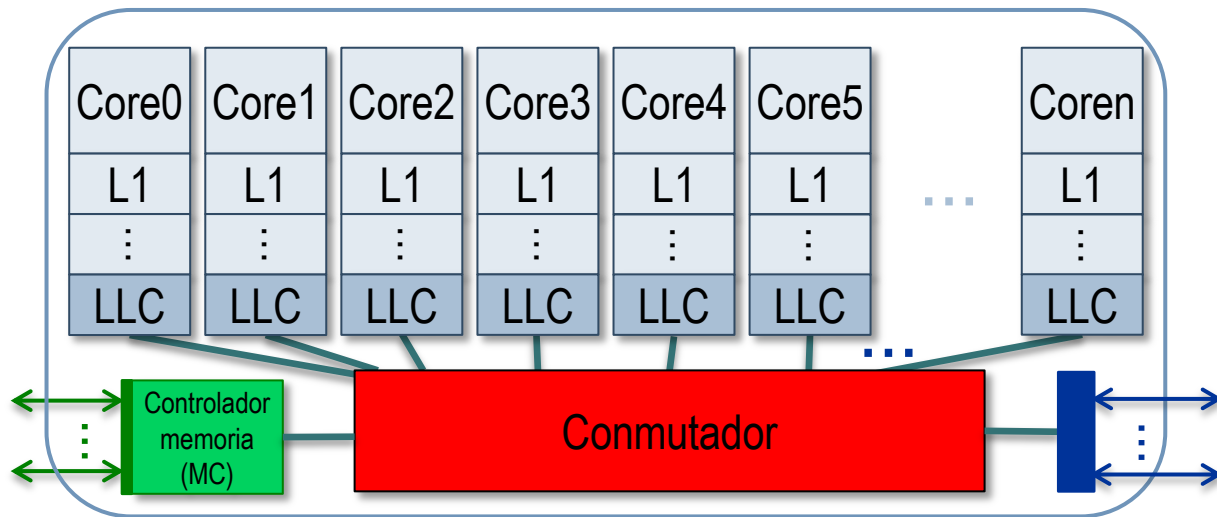
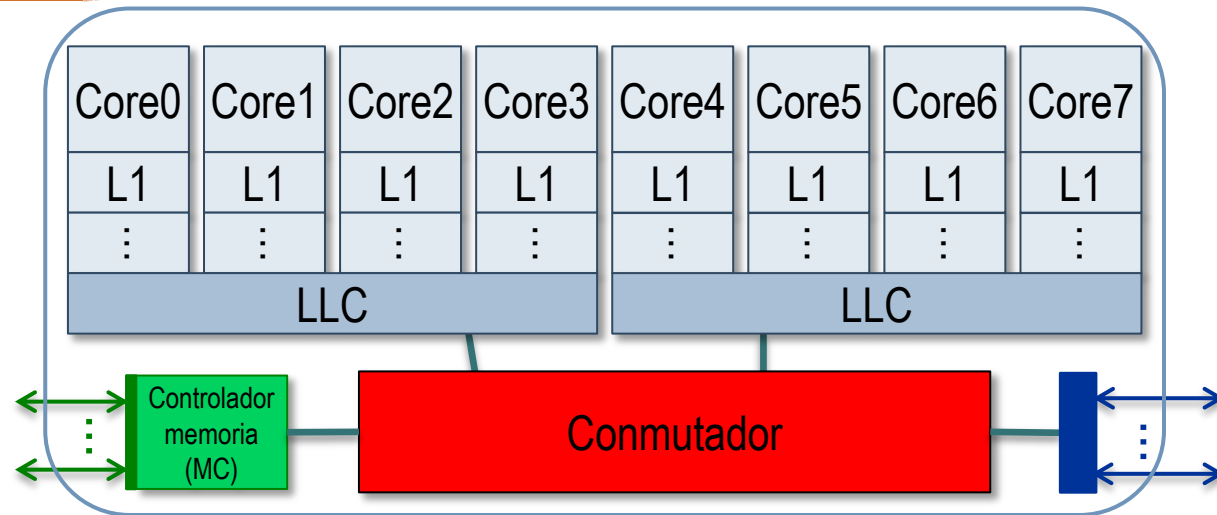
# Contenido Lección 7

- Clasificación y estructura de arquitecturas con TLP explícito y una instancia del SO
- Multiprocesadores
- **Multicores**
  - Ejecutan varios threads en paralelo en un **chip de procesamiento** multicore (cada thread en un core distinto)
- Cores Multithread
- Hardware y arquitecturas TLP en un chip

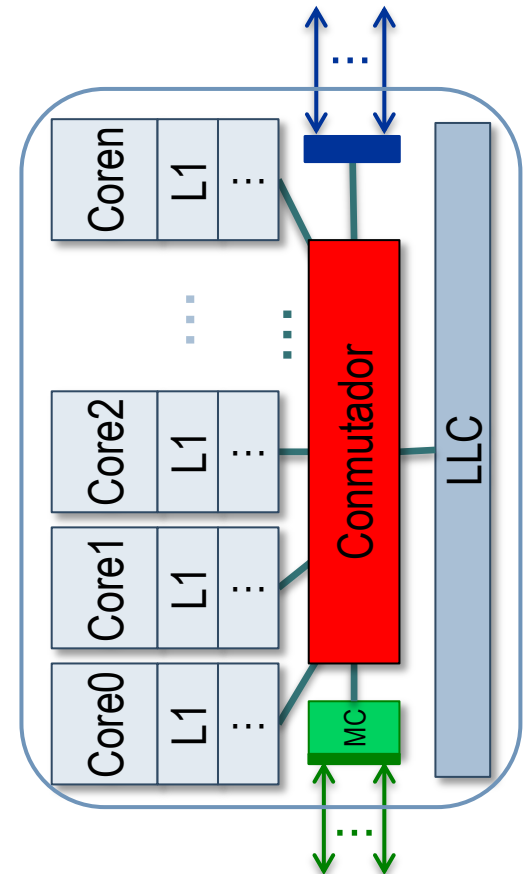
# Multiprocesador en un chip o Multicore o CMP (*Chip MultiProcessor*)



# Multicore: otras posibles estructuras



Link Controller (LC)



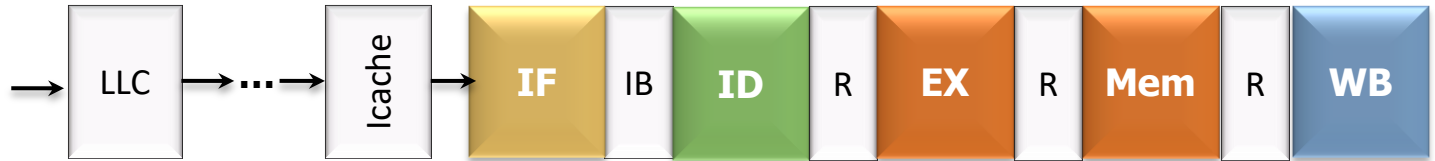
# Contenido Lección 7

- Clasificación y estructura de arquitecturas con TLP explícito y una instancia del SO
- Multiprocesadores
- Multicores
- Cores Multithread
  - Modifican su arquitectura ILP (segmentada, escalar o VLIW) para ejecutar threads concurrentemente o en paralelo
- Hardware y arquitecturas TLP en un chip

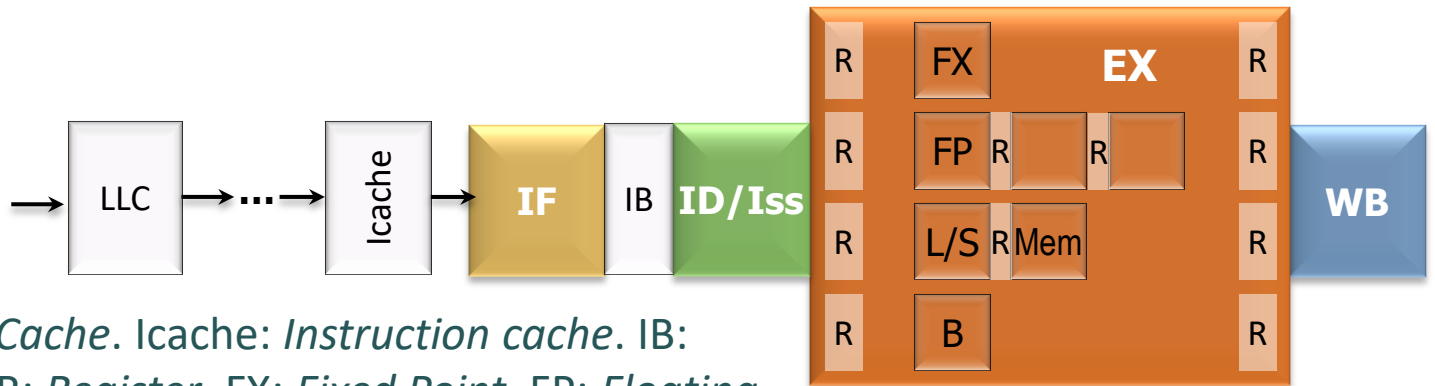


# Arquitecturas ILP

Escolar  
segmentada



VLIW y  
superescalar



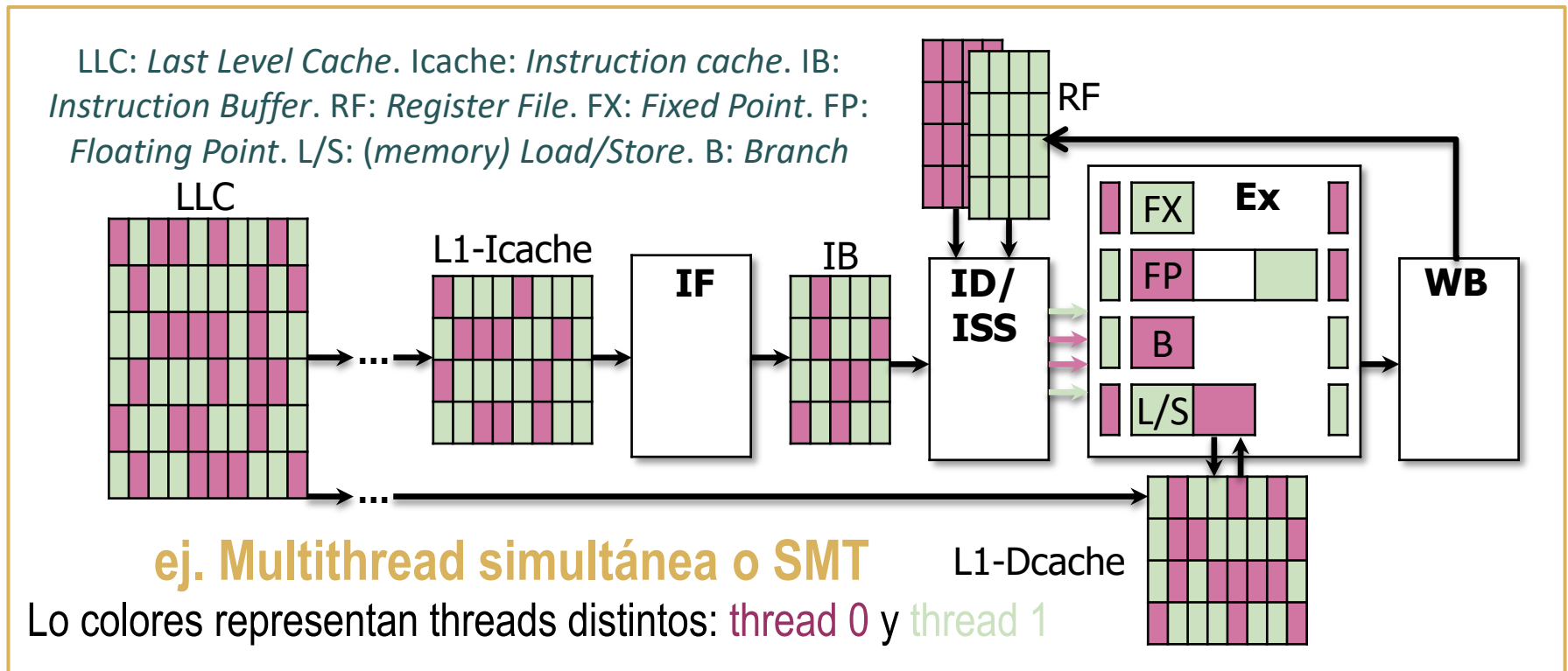
LLC: *Last Level Cache*. Icache: *Instruction cache*. IB: *Instruction Buffer*. R: *Register*. FX: *Fixed Point*. FP: *Floating Point*. L/S: *(memory) Load/Store*. B: *Branch*

- Etapa de captación de instrucciones (*Instruction Fetch*)
- Etapa de decodificación de instrucciones y emisión a unidades funcionales (*Instruction Decode/Instruction Issue*)
- Etapas de ejecución (*Execution*). Etapa de acceso a memoria (*Memory*)
- Etapa de almacenamiento de resultados (*Write-Back*)

# Arquitecturas ILP

- Procesadores/cores segmentados:
  - Ejecutan instrucciones **concurrentemente** segmentando el uso de sus componentes
- Procesadores/cores VLIW (*Very Large Instruction Word*) y superescalares:
  - Ejecutan instrucciones **concurrentemente** (segmentación) y en **paralelo** (tienen múltiples unidades funcionales y emiten múltiples instrucciones en paralelo a unidades funcionales)
    - VLIW:
      - Las instrucciones que se ejecutan en paralelo se captan juntas de memoria.
      - Este conjunto de instrucciones conforman la palabra de instrucción muy larga a la que hace referencia la denominación VLIW
      - El hardware presupone que las instrucciones de una palabra son independientes: no tiene que encontrar instrucciones que pueden emitirse y ejecutarse en paralelo
    - Superescalares:
      - Tiene que encontrar instrucciones que puedan emitirse y ejecutarse en paralelo (tiene hardware para extraer paralelismo a nivel de instrucción)

# Modificación de la arquitectura ILP en Core Multithread (ej. SMT)



- Almacenamiento: se multiplexa, se reparte o comparte entre threads, o se replica
  - Con SMT: repartir, compartir o replicar
- Hardware dentro de etapas: se multiplexa, o se reparte o comparte entre threads
  - Con SMT: unidades funcionales (etapa Ex) compartidas, resto etapas repartidas o compartidas; multiplexación es posible (p. ej. predicción de saltos y decodificación)

# Clasificación de cores multithread

## ➤ *Temporal Multithreading* (TMT)

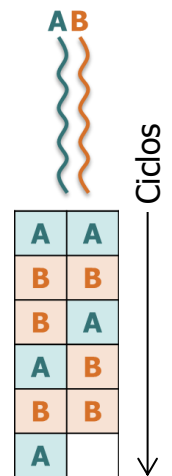
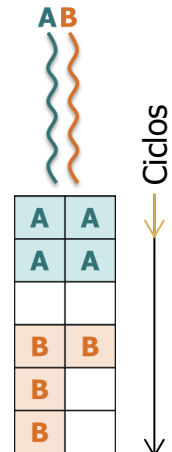
- Ejecutan varios threads **concurrentemente** en el mismo core

- La conmutación entre threads la decide y controla el hardware

- Emite instrucciones de **un único thread** en un ciclo

## ➤ *Simultaneous MultiThreading* (SMT) o multihilo simultáneo o *horizontal multithread*

- Ejecutan, en un core superescalar, varios threads en **paralelo**
- Pueden emitir (para su ejecución) instrucciones de **varios threads** en un ciclo



# Clasificación de cores con TMT



Ciclos



Ciclos



## ➤ *Fine-grain multithreading (FGMT) o interleaved multithreading*

### ➤ La conmutación entre threads la decide el hardware **cada ciclo** (coste 0)

- por turno rotatorio (*round-robin*) o
- por eventos de cierta latencia combinado con alguna técnica de planificación (ej. thread menos recientemente ejecutado)
  - Eventos: dependencia funcional, acceso a datos a cache L1, salto no predecible, una operación de cierta latencia (ej. div), ...

## ➤ *Coarse-grain multithreading (CGMT) o blocked multithreading*

### ➤ La conmutación entre threads la decide el hardware (coste de 0 a varios ciclos)

- tras intervalos de tiempo prefijados (*timeslice multithreading*) o
- por eventos de cierta latencia (*switch-on-event multithreading*).

# Clasificación de cores con CGMT con conmutación por eventos

## ➤ Estática:

### ➤ Conmutación

- Explícita: instrucciones explícitas para conmutación (instrucciones añadidas al repertorio)
- Implícita: instrucciones de carga, almacenamiento, salto

### ➤ Ventaja/Inconveniente:

- coste cambio contexto bajo (0 o 1 ciclo) / cambios de contextos innecesarios

## ➤ Dinámica:

### ➤ Conmutación típicamente por:

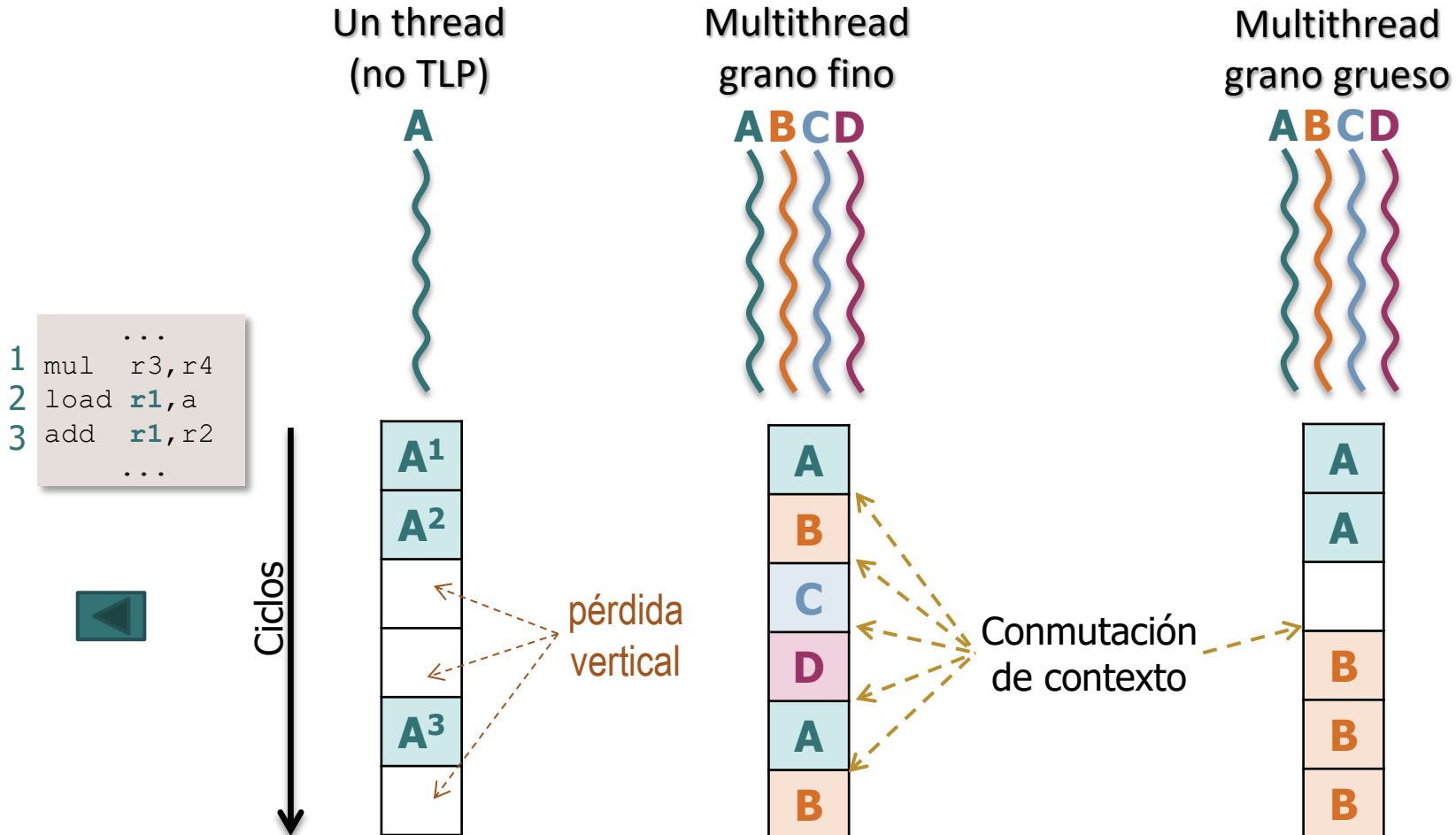
- fallo en la última cache dentro del chip de procesamiento (conmutación por fallo de cache), interrupción (conmutación por señal), ...

### ➤ Ventaja/Inconveniente:

- reduce cambios de contexto innecesarios / mayor sobrecarga al cambiar de contexto

# Alternativas en un core escalar segmentado

- En un core escalar se emite una instrucción cada ciclo de reloj



# Alternativas en un core con emisión múltiple de instrucciones de un thread

- En un core superescalar o VLIW se emiten más de una instrucción cada ciclo de reloj; en las alternativas de abajo, de un único thread



Un thread  
(no TLP)

FX	FP	M	BR
----	----	---	----

```

1 mulf f1, f2
2 mul  r3, r4
3 load r1, a
4 add  r1, r2
5 addf f1, f3
...
    
```

pérdida horizontal

Ciclos

A	A		
A <sup>2</sup>	A <sup>1</sup>	A <sup>3</sup>	
A <sup>4</sup>			A <sup>6</sup>
A <sup>8</sup>	A <sup>5</sup>		A <sup>7</sup>
		A	

13 huecos

pérdida vertical

Multithread  
grano fino



A	A		
B	B		B
C			
D	D	D	D
A <sup>2</sup>	A <sup>1</sup>	A <sup>3</sup>	
B		B	B

8 huecos

Multithread  
grano grueso



Comutación de contexto

A	A		
A <sup>2</sup>	A <sup>1</sup>	A <sup>3</sup>	
B	B		B
B		B	B
B		B	B

10 ← ancho emisión →



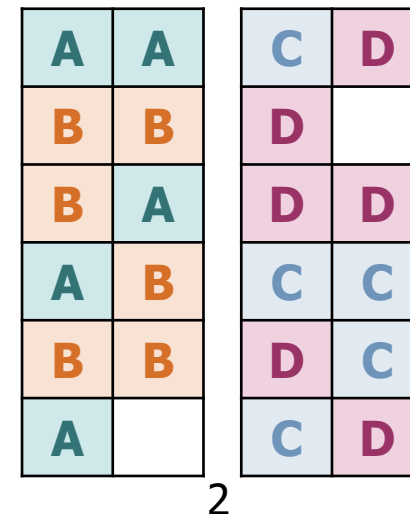
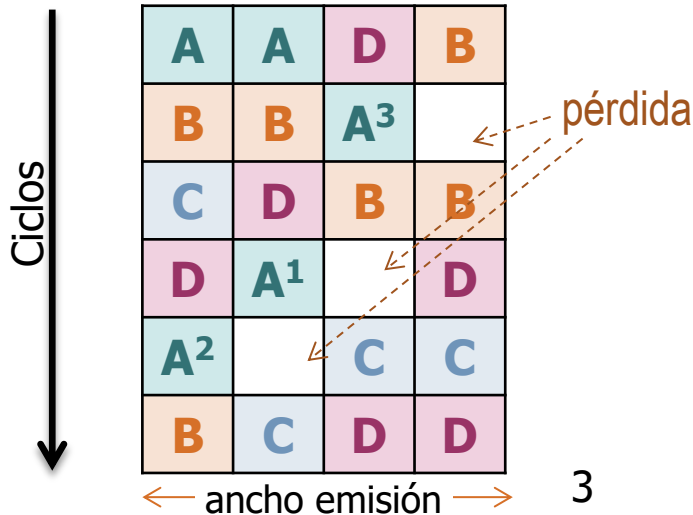
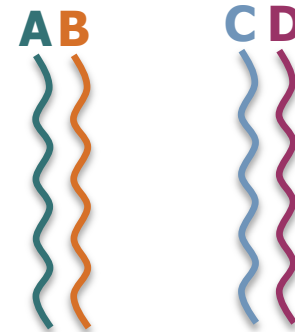
# Core multithread simultánea y multicores

- En un multicore y en un core superescalar con SMT (*Simultaneous MultiThread*) se pueden emitir instrucciones de distintos threads cada ciclo de reloj

Core con multithread simultánea (4 thread por core)



Multicore con 2 cores con multithread simultánea (2 threads por core)



# Contenido Lección 7

- Clasificación y estructura de arquitecturas con TLP explícito y una instancia del SO
- Multiprocesadores
- Multicores
- Cores Multithread
- Hardware y arquitecturas TLP en un chip

# Hardware y arquitecturas TLP en un chip



Hardware	CGMT	FGMT	SMT	CMP
Registros de la arquitectura	replicado (al menos PC)	replicado	replicado	replicado
Almacenamiento	multiplexado	multiplexado, repartido, compartido o replicado	repartido, compartido o replicado	replicado
Otro hardware de las etapas del cauce	multiplexado	Captación: repartida o compartida; <u>Resto:</u> multiplexadas	<u>UF:</u> compartidas; <u>Resto:</u> repartidas o compartidas	replicado
Etiquetas para distinguir el thread de una instr.	Sí	Sí	Sí	No
Hardware para conmutar entre threads	Sí	Sí	No	No

# Para ampliar ...

## ➤ Webs

- An Introduction to the Intel® QuickPath Interconnect,  
<http://www.intel.com/content/www/us/en/io/quickpath-technology/quick-path-interconnect-introduction-paper.html>
- Intel® QuickPath Technology Animated Demo [119 K]  
<http://www.intel.com/content/www/us/en/performance/performance-quickpath-architecture-demo.html>

## ➤ Artículos en revistas

- Kongetira, P.; Aingaran, K.; Olukotun, K.; , "Niagara: a 32-way multithreaded Sparc processor," *Micro, IEEE* , vol.25, no.2, pp. 21-29, March-April 2005. Disponible en línea (biblioteca UGR): <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=1453485&isnumber=31213>