

CE4302 – Arquitectura de Computadores II

Introducción al paralelismo a nivel de hilos (TLP)

PROFESOR: ING. LUIS BARBOZA ARTAVIA

Agenda

- Introducción
- Enfoques multihilo
- Arquitecturas multihilo

Introducción

- Surge paralelismo a nivel de instrucción.
- ILP mejoraba ciertas situaciones.
- ILP bastante transparente para el programador.
- ILP es difícil de explotar en algunas aplicaciones.
 - Tipo de aplicación, tasa de instrucciones, stalls (caché, memoria).
- Surge el paralelismo a nivel de hilos.

Multihilo

Sucede cuando el flujo de instrucciones se divide en varios flujos más pequeños (hilos) de manera que los hilos se puedan ejecutar de manera paralela.

- Esta división puede o no coincidir con el concepto de hilos del SO.

Definiciones

- **Proceso:** instancia de un programa que se ejecuta en un computador.
- Características:
 - Contiene programa, datos, recursos y demás información.
 - Un proceso debe operar de manera correcta.
 - En cualquier momento debe ser posible cambiar de un proceso a otro.

Cambio de contexto

- Acción de cambiar de un proceso a otro.
- Características:
 - Se debe guardar el estado del proceso.
 - Estados: nuevo, listo, en ejecución, en espera, terminado.

Definiciones

- **Hilo:** unidad ejecutable dentro de un proceso.
- Características:
 - Contener toda la información necesaria.
 - Instrucciones, datos, PC, estado de registros.
 - Posibilidad de cambiar de hilo.
 - Oculta latencias de memoria.

¿Cuál es menos costoso?

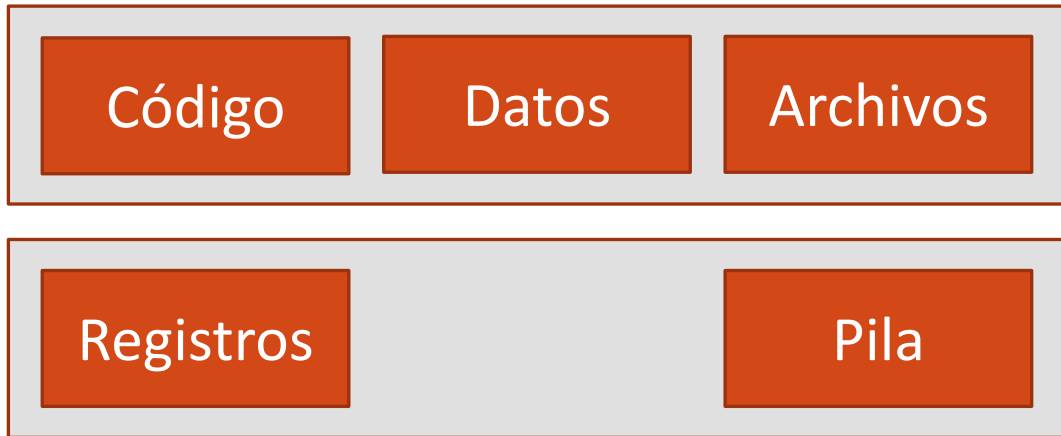
- Cambio de contexto.
- Cambio de hilo.
- Ambos son iguales en términos de costo.

Definiciones

- **Hilo:** unidad ejecutable dentro de un proceso.
- Características:
 - Contener toda la información necesaria.
 - Instrucciones, datos, PC, estado de registros.
 - Posibilidad de cambiar de hilo.
 - Oculta latencias de memoria.
- Cambio de hilo es **menos costoso** que un cambio de contexto.

¿Cuál es la diferencia entre un hilo y un proceso?

Un hilo vs Multihilo



Procesador multihilo

- Un PC por separado.
- HW adicional para ejecución concurrente.
- La búsqueda de instrucciones por hilo.
- Procesador trata cada hilo por separado.
- Puede utilizar técnicas de optimización: predicción de saltos, renombrar registros, pipelining, ejecución fuera de orden.

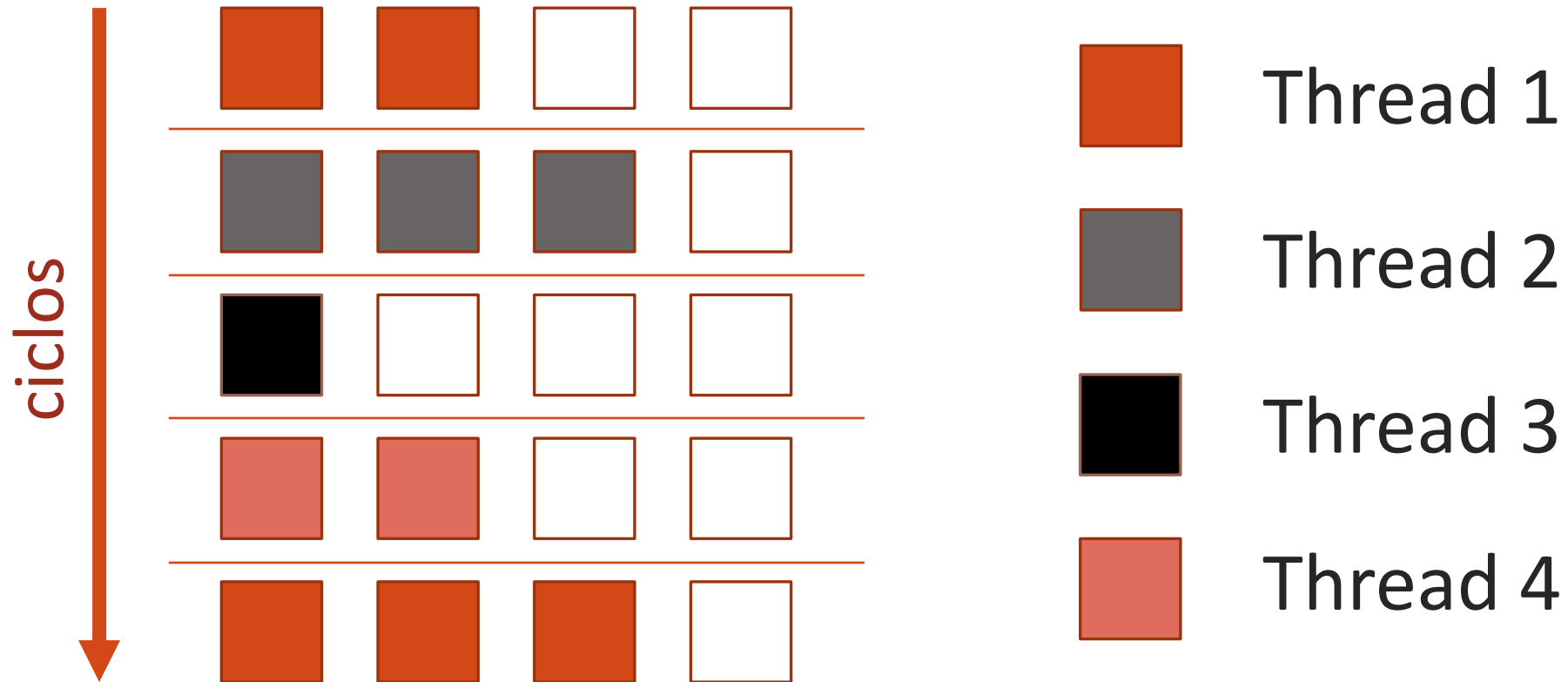
Multihilo

- Multihilo intercalado (fine-grained multithreading)
- Multihilo bloqueado (coarse-grained multithreading)
- Multihilo simultáneo (SMT)
- Multiprocesamiento en chip

Multihilo Intercalado

- Procesador tiene que lidiar con dos o más cambios de hilos por cada ciclo de reloj.
- Si hay un hilo bloqueado por alguna dependencia de datos o latencia en memoria, se salta y se ejecuta un hilo que esté listo.

Multihilo Intercalado



Multihilo Intercalado

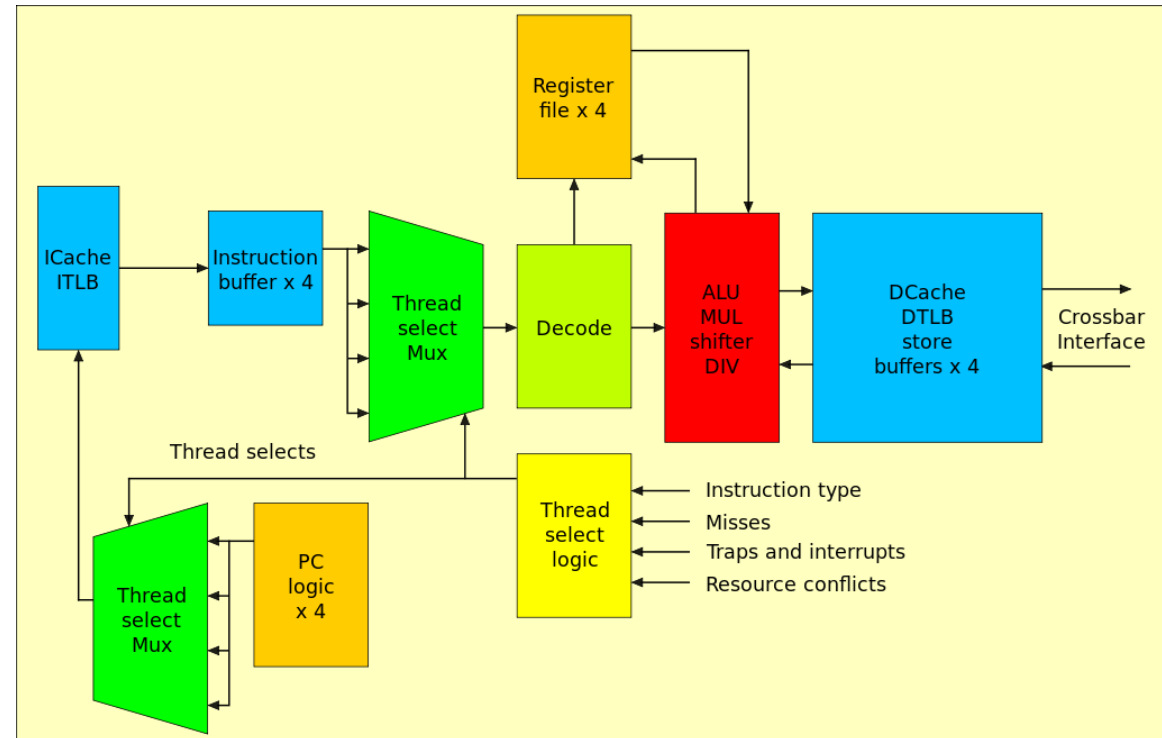
- **Ventaja:** esconder pérdidas en rendimiento por ejecutar otros hilos.
- **Desventaja:** hilos que no presentan *stalls* pueden atrasar su ejecución.

Multihilo Intercalado

- Los procesadores SPARC T1 a T5 (Oracle) lo utilizan.
- Se encuentran en servidores.
- Destinados a procesamiento de transacciones y servicios web.
- T5 soporta 16 cores y 128 hilos por procesador.
- Las GPU de NVIDIA también usan multihilo intercalado.

UltraSPARC T1 (2005)

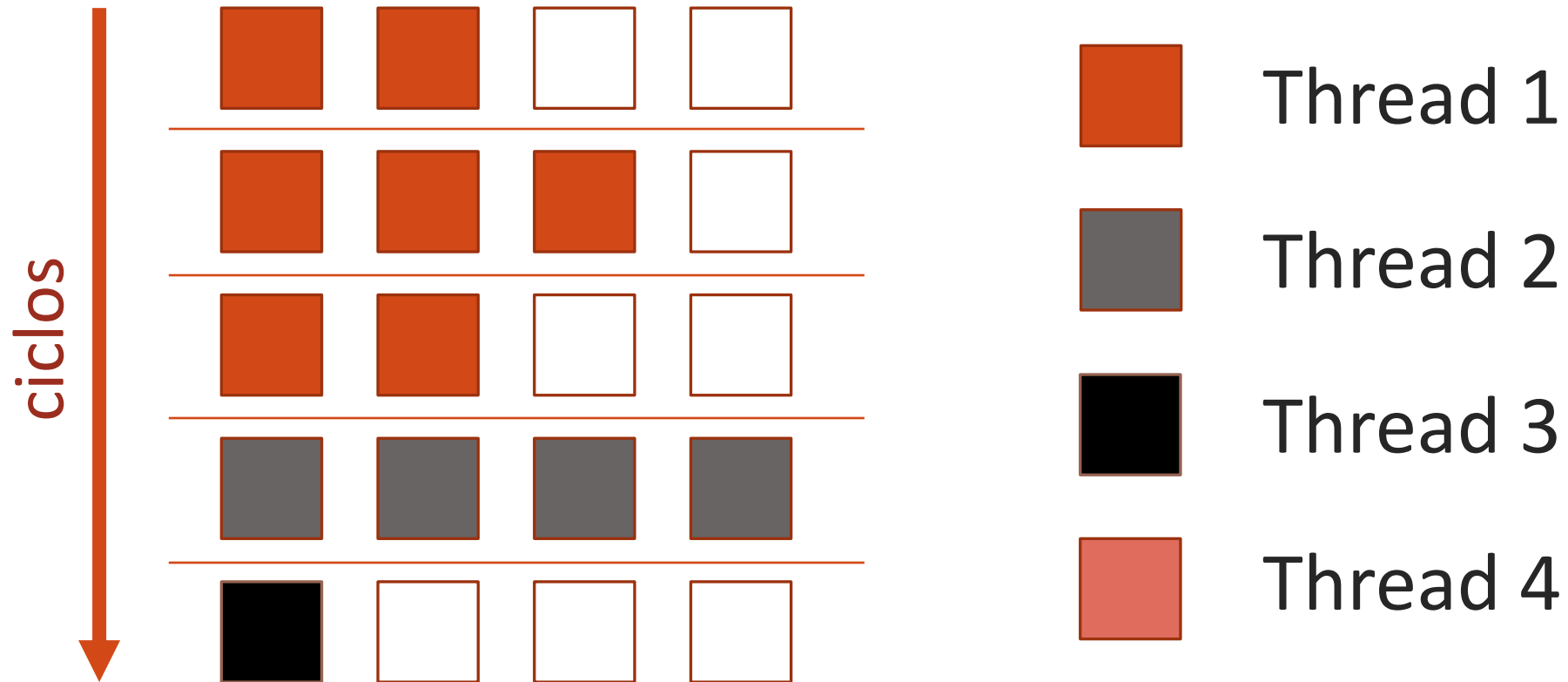
- Cada core (de 8) soporta 4 hilos concurrentes.
- Los hilos se eliminan de selección en eventos con mucha latencia.
- Esto permite 16KB de instrucciones y 8KB de cachés de datos.



Multihilo Bloqueado

- Las instrucciones de un hilo se ejecutan sucesivamente hasta que un evento (costoso) ocurra y provoque un atraso.
- Ejemplo: cache miss (L2 o L3).

Multihilo Bloqueado



Multihilo Bloqueado

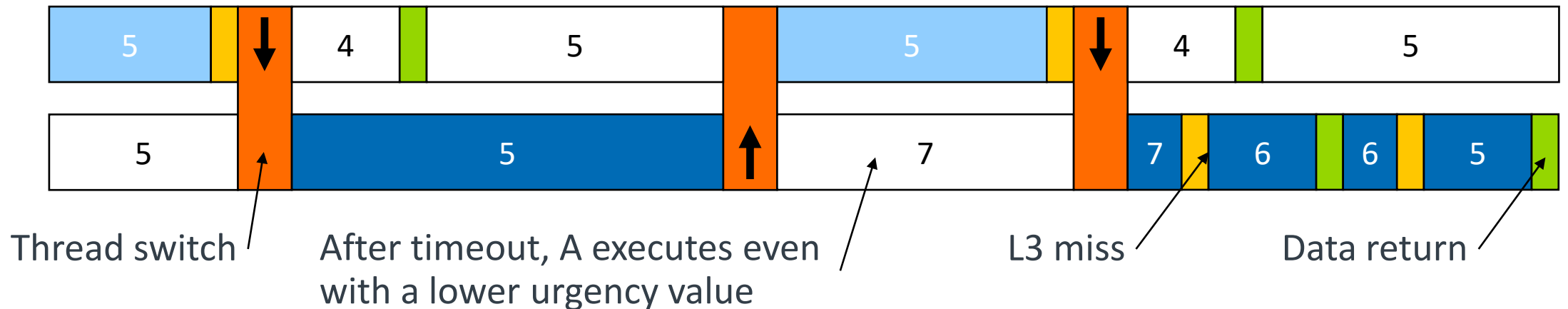
- Razones para un cambio de hilo:
 - Caché misses L1 o L2.
 - Operaciones complejas de ALU.
 - Tiempo agotado.
 - Prioridad hilo.
- ¿Cómo reducir la penalidad de cambio de hilo?
 - Pipeline más corto.
 - Agregando un buffer dedicado al cambio de hilo (prefetch).

Multihilo Bloqueado

- **Ventaja:** de utilidad para procesadores en-orden.
- **Desventaja:** costo de vaciar y llenar el pipeline es alto. No muy explorado o implementado en arquitecturas modernas.

Intel Montecito (Itanium 2)

- Soporta 2 hilos por core.
- A cada hilo se le da un valor de “urgencia”.
 - Mayor urgencia, mayor prioridad.
 - Urgencia se actualiza de forma dinámica de acuerdo con eventos del sistema.
- Cambio de hilo puede ocurrir por:
 - L3 cache miss o retorno de datos.
 - Timeout u otro evento del sistema.
- Figura muestra 2 hilos con eventos.
 - Coloreado cuando hay cambio de hilo, sin color cuando sale del cambio de hilo.
 - Urgencia dentro del cuadro.



Hay instrucciones de múltiples hilos
en el pipeline en...

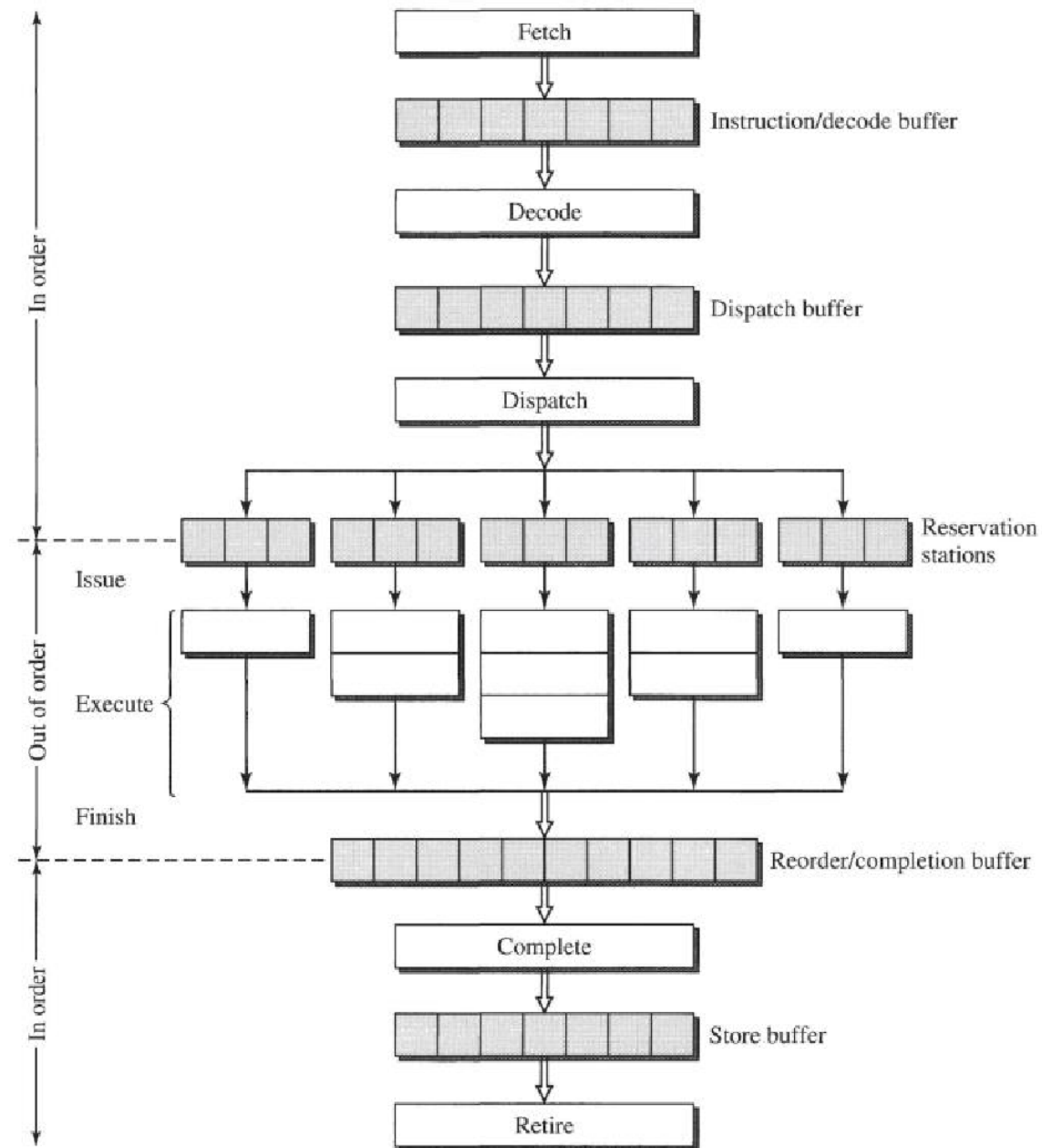
- Intercalado
- Bloqueado
- Ninguno de los anteriores

Nombre una diferencia entre el multihilo intercalado y bloqueado.

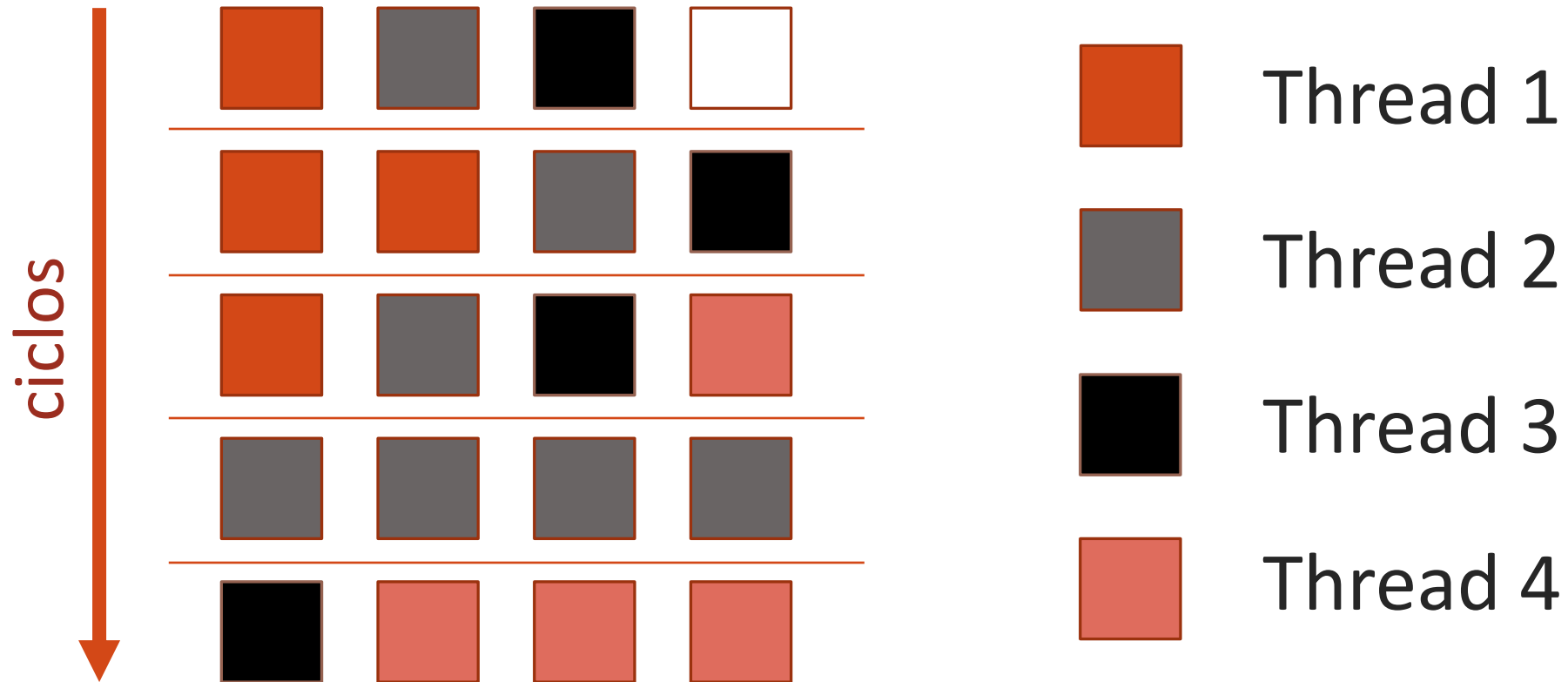
Multihilo Simultáneo

- Las instrucciones son tratadas desde múltiples hilos a las unidades de ejecución.
- Es una variación de fine-grained.

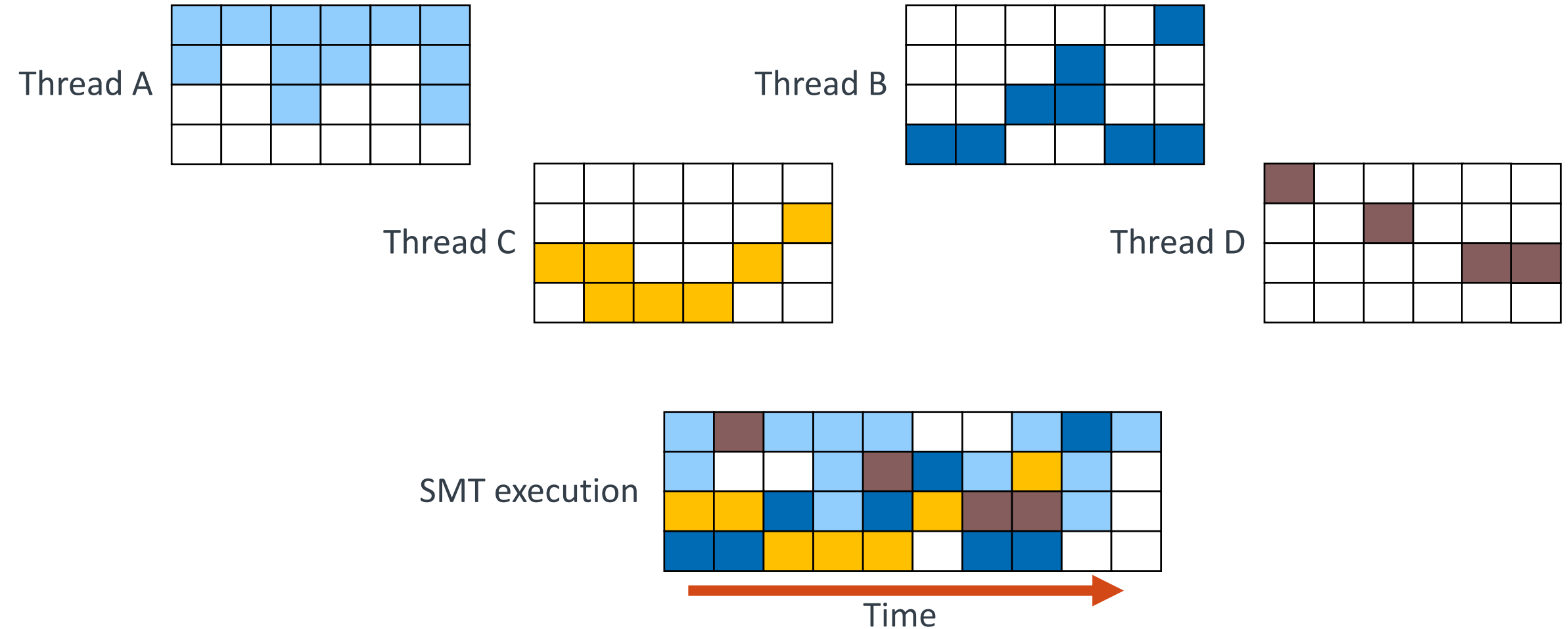
Multih



Multihilo Simultáneo



Multihilo Simultáneo



Multihilo Simultáneo

- Es utilizado para esconder grandes cantidades de latencias en un procesador.
- Incrementa el uso de unidades funcionales.
- Se logra mediante renombramiento de registros y calendarización dinámica.

Multihilo Simultáneo

Neoverse E1 CPU pipeline



- 10 stage integer pipeline
- Dual fetch/decode/rename/dispatch
- 8-instruction Reservation Station
- 40-instruction Re-Order Buffer
- Out-of-order 3-wide issue
- SMT issue across the two threads
- Dual 64-bit integer ALUs
- Dual 64-bit FP/Neon data paths
- 128-bit, non-blocking load / store pipeline
- Dual-issue load/store address and store data



Instruction fetch

Decode / rename / dispatch



Integer data path

FP / vector data path

Integer data path
(incl. mul / div / branch)

FP / vector data path

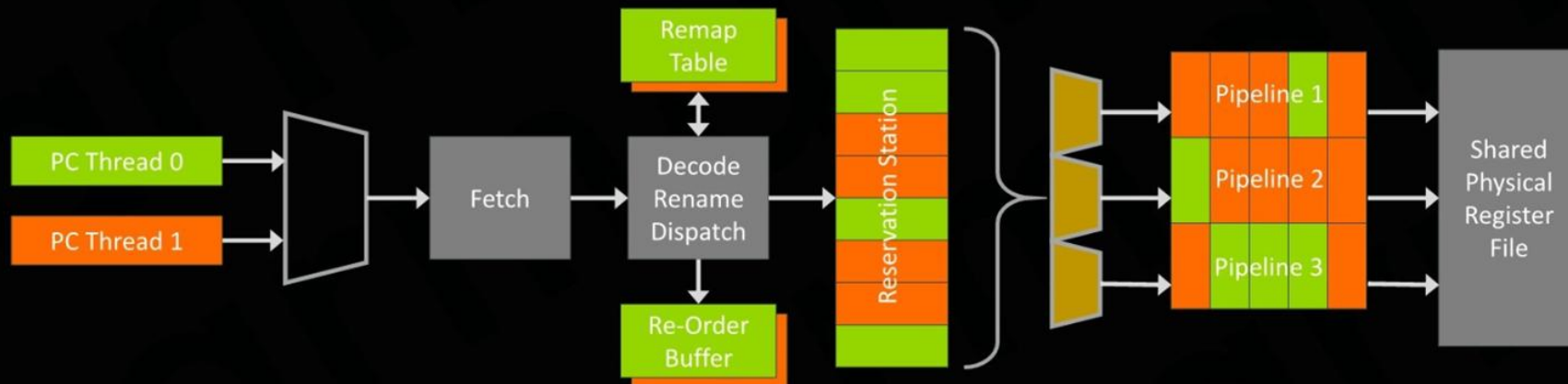
Store data

Load / store address

Multihilo Simultáneo

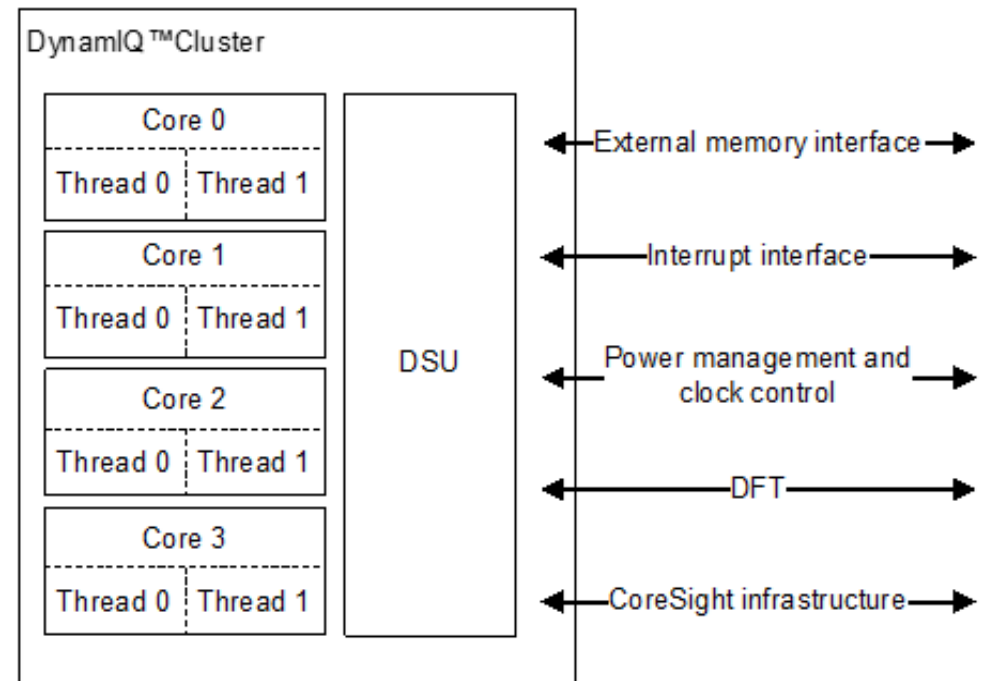
Improving throughput with multithreading

- The two threads appear to software as functionally two separate CPUs
 - Architectural state (General Purpose Registers, Vector Registers, System Registers) replicated per thread
- Each thread can be at different exception levels, running different OSes, etc.
- Simultaneous MT: instructions from both threads can be simultaneously executed
- Thread fairness: fetch aims to alternate round-robin between threads



Multihilo Simultáneo

- Arm's first SMT Cortex-A core
- Cortex-A65 implements a 64-bit ISA only
- Built on DynamIQ technology
 - DSU contains all the interfaces to connect to the system on chip (SoC).
- Dual-threaded, out-of-order execution
 - Each thread is a PE.
- Separate L1 instruction and data cache
 - L2 cache is optional.
 - L3 cache in DSU

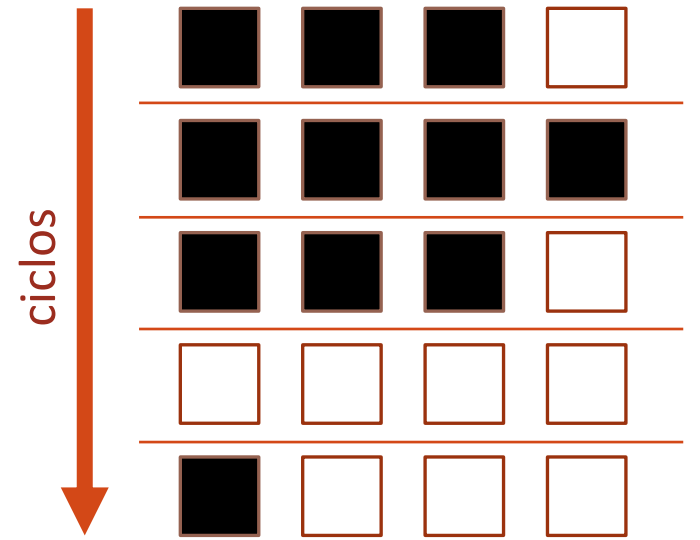
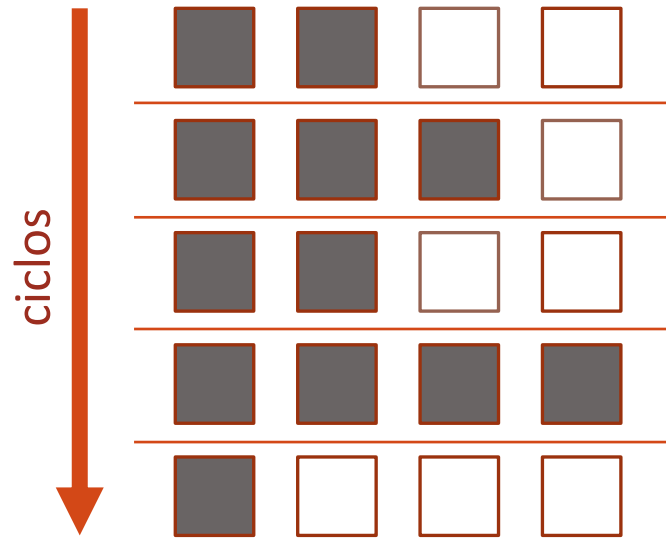
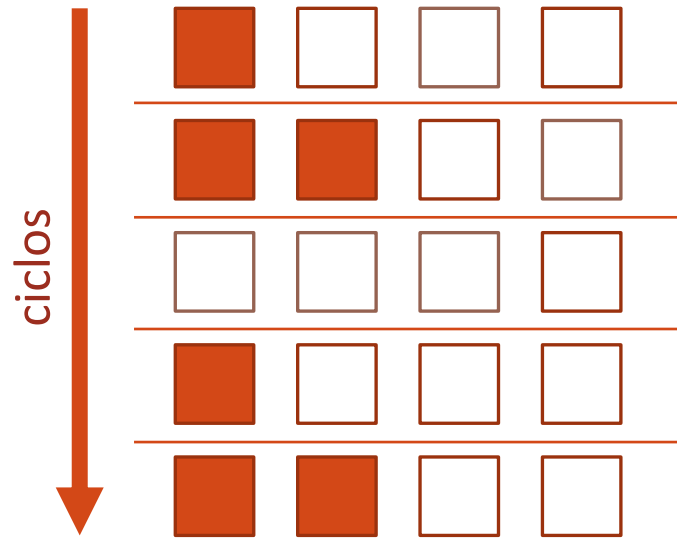


ARM DynamIQ Shared Unit (DSU) integrates one or more cores with an L3 memory system, control logic and external interfaces to form a multicore cluster.

Multiprocesamiento en chip

- Se implementan múltiples núcleos en un único chip y cada núcleo maneja diferentes hilos.

Multiprocesamiento en chip



Multiprocesamiento en chip

- El hardware es usado de una manera más eficiente sin depender de la complejidad del pipeline.

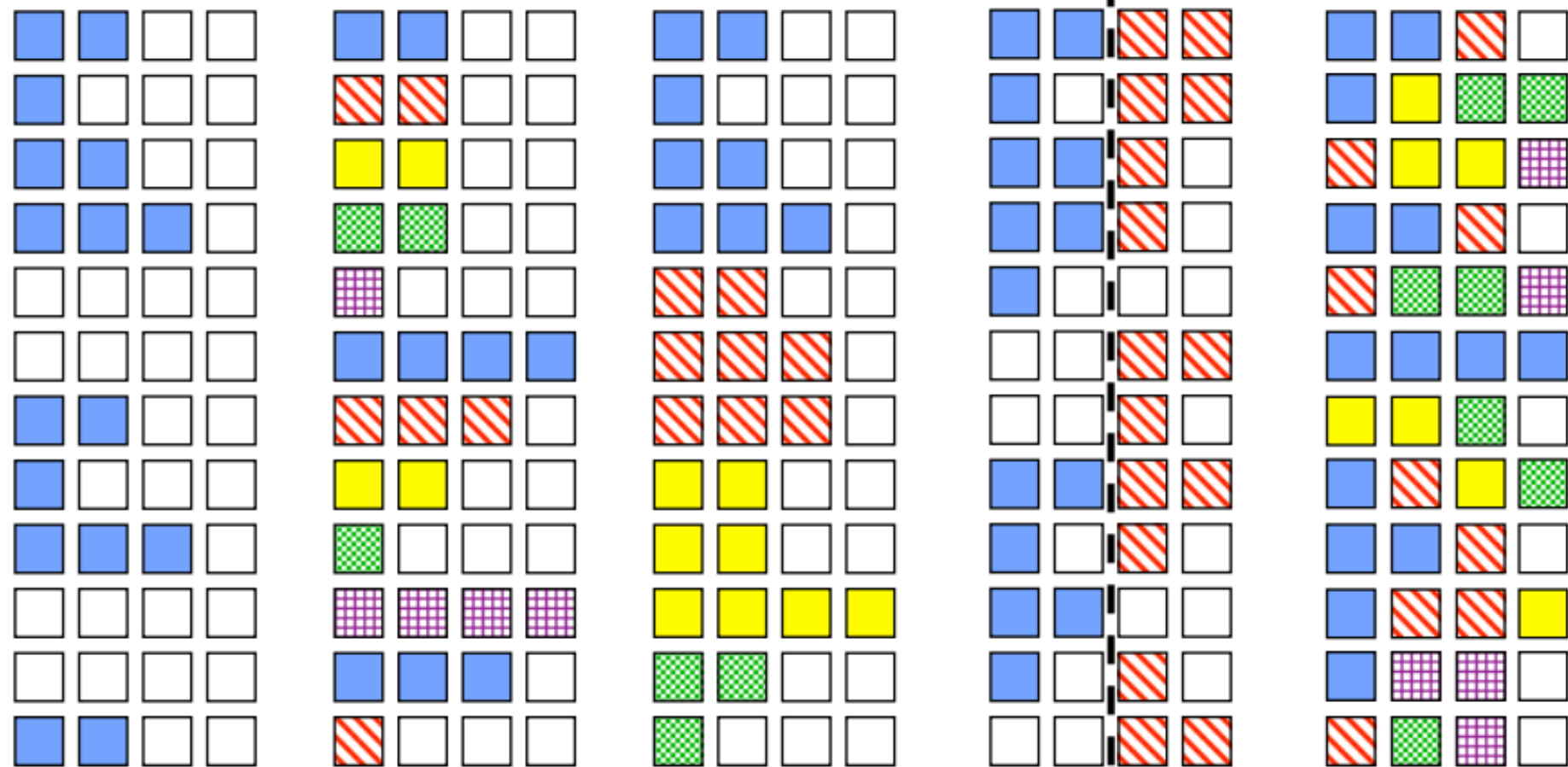
Time (processor cycle)

Superscalar


Fine-Grained Coarse-Grained

Multiprocessing


Simultaneous Multithreading




■ Thread 1

 Thread 2

Thread 3

 Thread 4

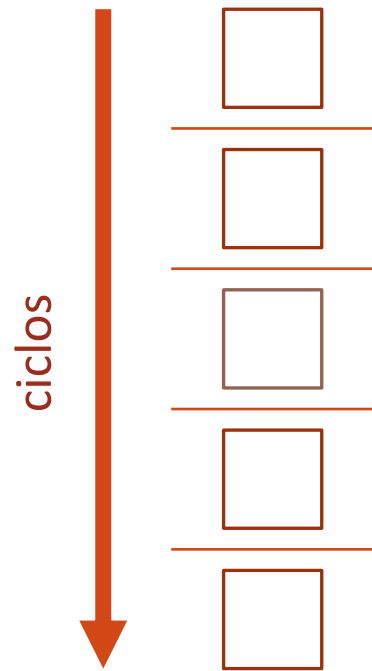
 Thread 5

☐ Idle slot

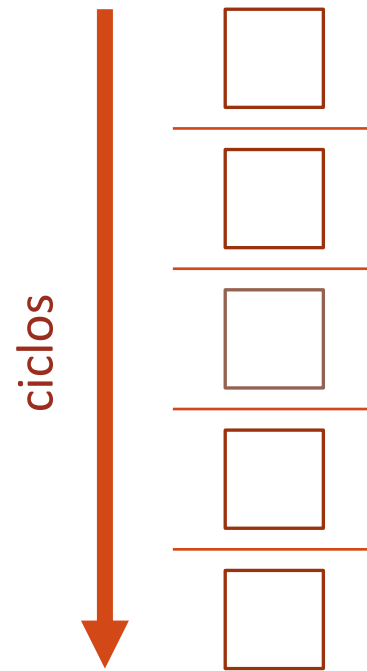
Implementaciones multihilo

- Escalares.
- Superescalares.

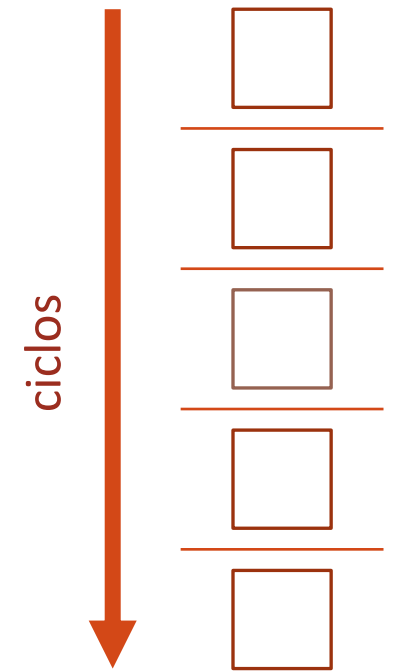
Arquitecturas escalares



Escalar
Único hilo

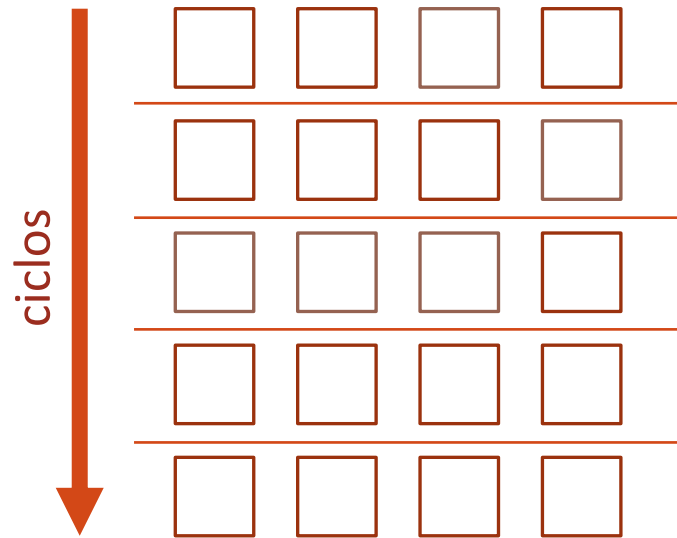


Intercalado

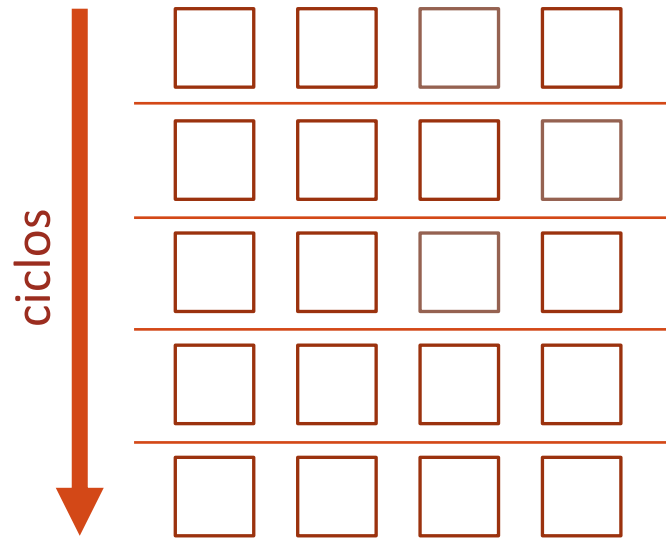


Bloqueado

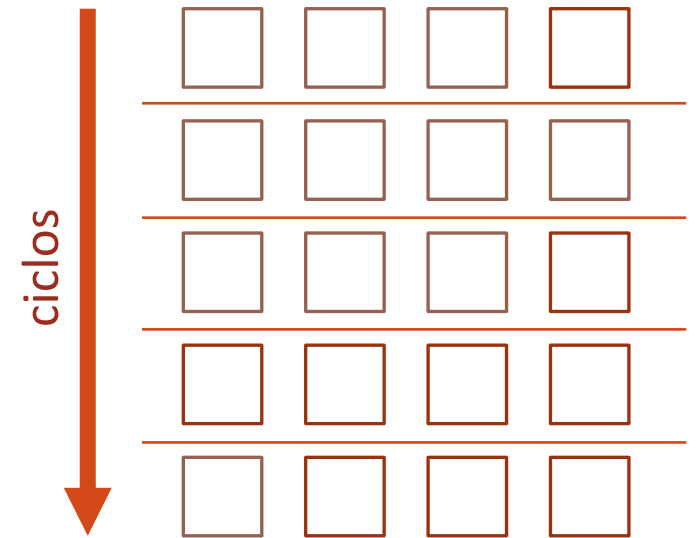
Arquitecturas superescalares



Superescalar
Único hilo

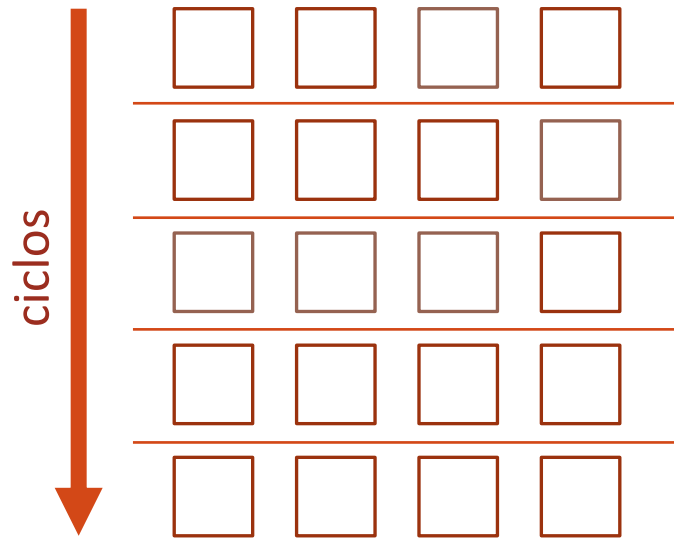


Intercalado

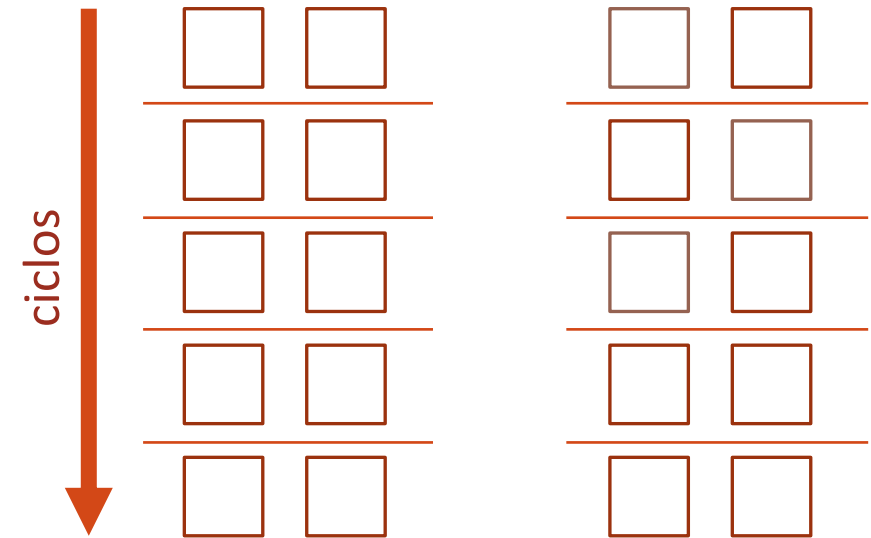


Bloqueado

SMT y Multiprocesador en chip



SMT



Multicore

Seleccione las desventajas del SMT:

- Diseño más difícil.
- Hilos sin stalls atrasan su ejecución.
- Se utiliza más HW.
- Costo de vaciar pipeline es alto.
- Incrementa uso de FU.

Papers

Thread level parallelism of desktop applications

University of Michigan
1301 Beal Ave.
Ann Arbor, MI
48109-2122
+1-734-764-0203

Intel Microprocessor Research Lab
5350 NE Elam Young Parkway
Hillsboro, OR
97123
+1-503-696-3154

Krisztián Flautner	manowar@engin.umich.edu
Rich Uhlig	richard.a.uhlig@intel.com
Steve Reinhardt	stever@eecs.umich.edu
Trevor Mudge	tmm@eecs.umich.edu

¿Por qué TLP en aplicaciones desktop?
¿Qué encontraron los autores?

Papers

Evolution of Thread-Level Parallelism in Desktop Applications

Geoffrey Blake, Ronald G. Dreslinski, Trevor Mudge
University of Michigan, Ann Arbor
[blakeg,rdreslin,tnm]@umich.edu

Krisztián Flautner
ARM
krisztian.flautner@arm.com

- To what degree does the overall system leverage concurrency, and how has that changed from 10 years ago?
- What impact does SMT have on parallel performance?
- How are GPU's being used to improve system performance, and do opportunities exist to further exploit them?
- How does architectural sophistication and clock frequency impact TLP?

Investigar

- 1- Qué es *thread level speculation (TLS)*?
- 2- Que son algoritmos irregulares en el contexto de ***parallel computing***.
- 3- Qué herramientas existen en la industria para analizar el ***TLP***.
- 4- Investigue sobre los siguientes términos: ***logical core, physical core, risc-v hart***
- 5- Sección 5.1 John L Hennessy y David A Patterson. *Computer architecture: a quantitative approach*. Elsevier, 2017.
- 6- Investigue que es el thread director en las nuevas arquitecturas de Intel y como se relaciona con el OS

Converting Thread-Level Parallelism to Instruction-Level Parallelism via Simultaneous Multithreading

JACK L. LO and SUSAN J. EGGERS
University of Washington
JOEL S. EMER
Digital Equipment Corporation
HENRY M. LEVY
University of Washington
REBECCA L. STAMM
Digital Equipment Corporation
and
DEAN M. TULLSEN
University of California, San Diego

Referencias

- Stallings, W. (2003). Computer organization and architecture: designing for performance. Pearson Education India.
- Hennessy, J., & Patterson, D. (2012). Computer Architecture: A Quantitative Approach (5th ed.). Elsevier Science..

CE4302 – Arquitectura de Computadores II

Introducción al paralelismo a nivel de hilos (TLP)

PROFESOR: ING. LUIS BARBOZA ARTAVIA