

Coherencia de Caché

IIC2343 - Arquitectura de Computadores

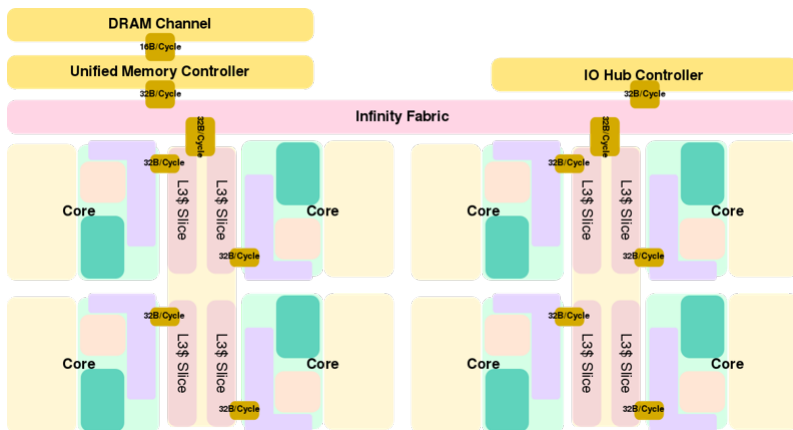
Nicolás Elliott B. (nicolas.elliott@uc.cl)



DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN
ESCUELA DE INGENIERÍA
PONTIFICIA UNIVERSIDAD CATÓLICA DE CHILE

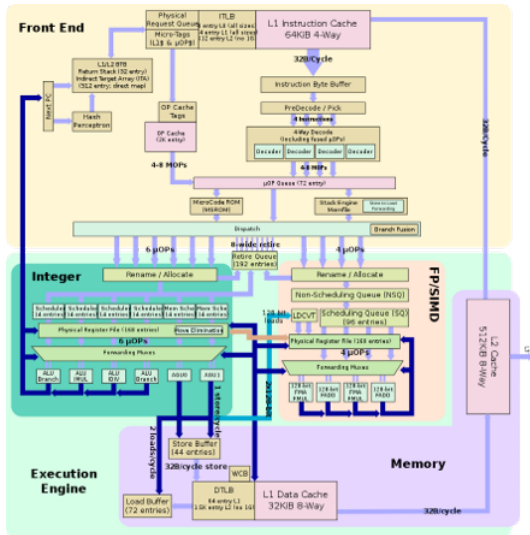
(II/2019)

Situación de los procesadores modernos



Fuente: https://en.wikipedia.org/wiki/File:zen_soc_block.svg

Situación de los procesadores modernos



Paralelismo MIMD es utilizado en casi todos los procesadores actuales

- Nace de la necesidad de ejecutar más threads en paralelo
- También del hecho de que hace un tiempo a la fecha el incremento de rendimiento de un núcleo individual no ha sido tan agresivo como hace años atrás
- Cada vez se incorporan más y más núcleos e hilos de ejecución

Paralelismo MIMD plantea nuevos desafíos a la hora de diseñar un procesador

- Se sabe que diseños UMA tienen problemas de rendimiento con más de 20 núcleos
- Profundidad de pipeline hace que el problema de disipación de calor no sea trivial para algunos diseños
- Los diferentes núcleos deben ser capaces de comunicarse entre si

El acceso a memoria se debe especificar

- Todos los núcleos comparten la misma memoria
- De algún modo se debe coordinar el acceso de los distintos núcleos a la memoria para poder atender las peticiones de todos
- Sistema Operativo interviene seleccionando qué procesos ejecutan en cada momento en qué núcleo
- Problema: dos procesos que comparten memoria quieren escribir en la misma dirección al mismo tiempo

El acceso a memoria se debe especificar

- Estos problemas se solucionan de diversas formas:
 - Instrucciones atómicas de compare and swap en el procesador
 - Mediante uso de locks en el software
 - Incorporando buffers y una cola de peticiones en el controlador de memoria
- ¿Cómo solucionamos fácilmente el hazard estructural de acceso a memoria en una arquitectura Von Neumann?
- Hay un problema que sigue sin resolverse: La caché

Problema de la caché

- Una posible solución es hacer todas las cachés write-through, pero no soluciona completamente el problema (¿Por qué?)
- Otra es adoptar un protocolo para asegurar que las cachés se mantienen coherentes en el procesador

Protocolo MESI

- Desarrollado por la universidad de Illinois
- Basado en invalidación
- Consiste en asignar estados a las líneas de la caché para comunicar eventos sobre estas, o saber cómo actuar

Estados MESI

- M: Modificado, línea presente sólo en la caché actual y ha sido modificada respecto a la memoria RAM
- E: Exclusiva, línea presente sólo en la caché actual y no ha sido modificada respecto a la RAM
- S: Compartida, línea se puede encontrar en otras cachés y no ha sido modificada respecto a la RAM
- I: Inválida, Indica que la línea es inválida

Protocolo MESI

- Este protocolo requiere que el controlador de caché verifique no sólo las peticiones hacia él, si no que las peticiones de otras cachés hacia la RAM. Caché Snooping
- Siempre se considera la RAM como la fuente última de la verdad
- Además, cachés deben tener un canal entre sí por el que comunicarse entre ellas. ¿Lo tienen?

Escribir en una línea S

- Líneas en estado compartido están presentes en más de una caché
- Si la caché es write-back, podría no enterarme de que una de las cachés cambió el dato
- ¿Cómo podemos solucionarlo?

Escribir en una línea S

- Líneas en estado compartido están presentes en más de una caché
- Si la caché es write-back, podría no enterarme de que una de las cachés cambió el dato
- ¿Cómo podemos solucionarlo?
 - Request for Ownership