

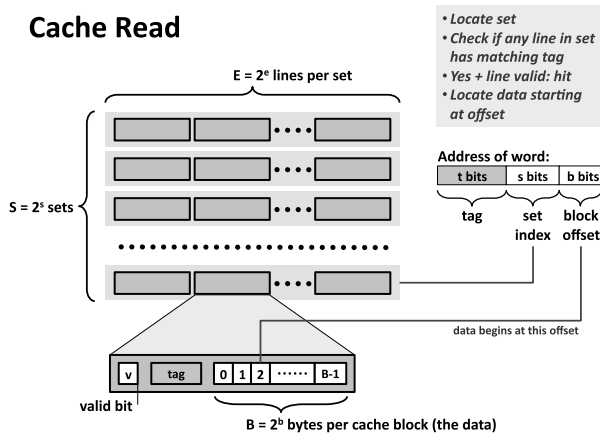
Memoria

- ▶ La memoria se puede definir como los circuitos que permiten almacenar y recuperar la información
- ▶ La unidad de almacenamiento es el *bit* (**binary element**) aunque normalmente la consideramos estructurada en *bytes* (8 bits)
- ▶ Aunque el byte es la unidad de direccionamiento, solemos hablar de *palabras*. *palabra* se refiere a la longitud de los registros del microprocesador.
 - ▶ Así hablamos de microprocesadores de 16 bits, de 32 bits ...
- ▶ Por razones históricas a veces se denomina *palabra* a 16 bits (2 bytes) y *doble palabra* a 32 bits

Acceso a la memoria

- ▶ Según la manera de acceder a las celdas de información de la memoria las podemos clasificar de
 - ▶ **memorias asociativas** Direccionables por contenidos: cuando se le pide la información almacenada en una dirección, la memoria asociativa está organizada de forma tal que puede encontrarla sencillamente inspeccionando los bits de la dirección y con una búsqueda asociativa (muy similar a una búsqueda en una tabla hash). Utilizada principalmente en las *caches*
 - ▶ **memorias convencionales** Cada celda es direccionable por un número al que llamaremos *dirección*

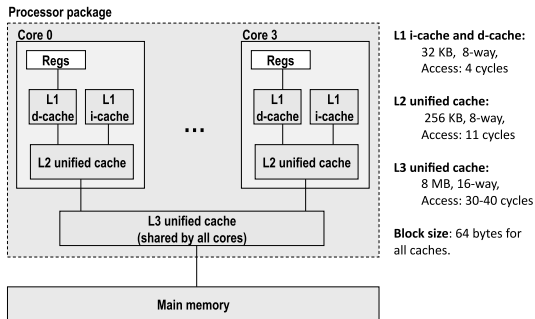
Cache Read



5

Figure: From R.E. Bryant et al. Computer Systems: A Programmer's Perspective (2nd edition), Pearson 2014

Intel Core i7 Cache Hierarchy



17

Figure: From R.E. Bryant et al. Computer Systems: A Programmer's Perspective (2nd edition), Pearson 2014

Jerarquía de la Memoria

- ▶ El tiempo de acceso es el tiempo necesario para realizar una operación de lectura/escritura, es decir, el tiempo que transcurre desde el instante en que se pone la dirección en el bus de direcciones hasta que el dato ha sido almacenado en memoria o puesto a disposición de la CPU.
- ▶ Interesa que el acceso sea lo mas rápido posible, pero ademas interesa tener la mayor capacidad sin incurrir en un costo excesivo. Por tanto se usan las memorias más rápidas y caras para donde los accesos son más frecuentes
- ▶ Surge así lo que se llama *jerarquía de la memoria* y que se establece en base a los tiempo de acceso y capacidad disponible
 1. Registros del microprocesador
 2. Memoria cache
 3. Memoria principal
 4. Unidades de disco
 5. Unidades de cinta u opticas



Examples of Caching in the Hierarchy

Cache Type	What is Cached?	Where is it Cached?	Latency (cycles)	Managed By
Registers	4-8 bytes words	CPU core	0	Compiler
TLB	Address translations	On-Chip TLB	0	Hardware
L1 cache	64-bytes block	On-Chip L1	1	Hardware
L2 cache	64-bytes block	On/Off-Chip L2	10	Hardware
Virtual Memory	4-KB page	Main memory	100	Hardware + OS
Buffer cache	Parts of files	Main memory	100	OS
Disk cache	Disk sectors	Disk controller	100,000	Disk firmware
Network buffer cache	Parts of files	Local disk	10,000,000	AFS/NFS client
Browser cache	Web pages	Local disk	10,000,000	Web browser
Web cache	Web pages	Remote server disks	1,000,000,000	Web proxy server

63

Figure: From R.E. Bryant et al. Computer Systems: A Programmer's Perspective (2nd edition), Pearson 2014

- ▶ El sistema operativo es un *asignador de recursos* lo que implica que para cada recurso
 - ▶ El S.O. debe llevar contabilidad del recurso
 - ▶ El S.O. debe tener una política de asignación del recurso
 - ▶ El S.O. debe asignar el recurso a los procesos que lo necesiten
 - ▶ El S.O. debe recuperar el recurso cuando los procesos ya no lo necesitan

Administración de memoria

- ▶ En un sistema el S.O. debe llevar contabilidad de la memoria
 - ▶ El S.O. tiene que llevar contabilidad de la memoria disponible en el sistema: la memoria no contabilizada por el S.O. no está disponible para los procesos
 - ▶ El S.O. también tiene que llevar una contabilidad por proceso
- ▶ El S.O. asigna memoria a los procesos cuando se inicia su ejecución y cuando la solicitan
- ▶ Cuando el proceso termina el S.O. RECUPERA la memoria que tenía asignada
- ▶ Si el sistema tiene memoria virtual es el S.O. quien se encarga de gestionarla

Sistemas simples: sistemas no multiprogramados

- ▶ El sistema mas simple de administración de memoria es *carecer de él*: válido en sistemas dedicados
- ▶ En un sistema operativo no multiprogramado solo dos zonas de memoria: el S.O y el proceso de usuario
- ▶ Tipicamenmte el S.O en las posiciones bajas de memoria y el resto para el proceso de usuario: concepto de monitor simple (IBSYS pra IBM 7094)
- ▶ Primeros ordenadores personales: S.O. en direcciones altas de la memoria (en ROM) y resto para procesos de usuario
- ▶ S.O. en direcciones bajas de la memoria y partes de S.O. en direcciones altas de memoria: Primeras versiones MS-DOS

Administración de memoria: Esquemas simples

- ▶ En el caso de sistemas operativos multiprogramados, el método mas simple es dividir la memoria en varias zonas
- ▶ Un proceso en cada zona
- ▶ Dos enfoques
 - ▶ zonas de tamaño fijo: El número de procesos en memoria tambien es fijo
 - ▶ zonas de tamaño variable: El número y el tamaño de las zonas puede variar