


Arquitectura de los Sistemas de C  mputo

Memorias




Caracter  sticas Fundamentales I

- Ubicaci  n
 - CPU (registros / memoria de microprograma)
 - Interna o Principal (RAM - acceso aleatorio)
 - Externa o Secundaria (acceso directo o secuencial)
- Capacidad
 - Tama  o de palabra
 - Cantidad de palabras
- Unidad Direccional
 - Palabra
 - Byte
- Performance
 - Tiempo de acceso
 - Ciclo de acceso
 - Tasa de transferencia
- Unidad de transferencia
 - Palabra
 - Bloque
- M  todo de acceso
 - Secuencial
 - Directo
 - Aleatorio
 - Asociativo

Arquitectura de los Sistemas de C  mputo

2




Caracter  sticas Fundamentales II

- Soporte
 - Semiconductores
 - Superficie magn  tica
 - Superficie   ptica
- Caracter  sticas F  sicas
 - Vol  til / No vol  til
 - Borrable / No borrrable
- Organizaci  n

Arquitectura de los Sistemas de C  mputo


3



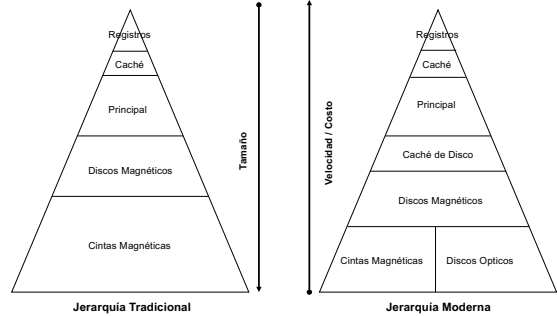
Jerarquía de memorias (I)

- En las últimas décadas la velocidad de proceso de las UCP (medida en cantidad de instrucciones por segundo), se ha duplicado cada 18 meses sin variar su precio
- Las memorias han cuadruplicado su tamaño cada 36 meses, pero han tenido un progreso menos espectacular en velocidad, con incrementos de sólo alrededor del 10% anual
- Una computadora típica contiene distintos tipos de memoria, interactuando coordinadamente de modo tal que parece tener una única memoria grande y rápida.
- La forma en la que se organizan los distintos tipos de memorias se conoce como jerarquía de memorias.


Arquitectura de los Sistemas de Cómputo
4



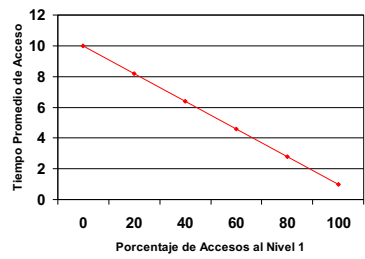
Jerarquía de Memorias



Arquitectura de los Sistemas de Cómputo
5



¿Por qué una jerarquía de memoria?




Arquitectura de los Sistemas de Cómputo
6



Comparación de los almacenamientos

Tipo de memoria	Tiempo de acceso	Costo por Mbyte	Tamaño típico	Costo típico
Registros	1 ns	Alto	1 Kb	~,-
Caché	5 – 20 ns	US\$ 100	1 Mb.	US\$ 100
Principal	60 – 80 ns	US\$ 1,10	128 Mb.	US\$ 70
Discos	10 ms	US\$ 0.05	40 Gb.	US\$ 200


Arquitectura de los Sistemas de Cómputo
7



Memoria de Semiconductores

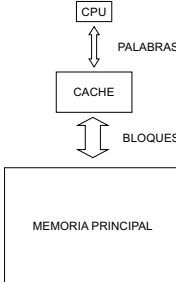
Tipo de memoria	Categoría	Mecanismo de Borrado	Mecanismo de Escritura	Volatilidad	Otras		
Random Access Memory (RAM)	Lectura / Escritura	Eléctricamente (nivel de byte)	Eléctricamente	Volátil	Estáticas Dinámicas		
Read-Only Memory (ROM)	Lectura	Imposible	Máscaras	No volátil	Estáticas		
Programmable ROM (PROM)							
Erasable PROM (EPROM)	Principalmente de lectura	Radiación ultravioleta (nivel de chip)	Eléctricamente				
Flash Memory		Eléctricamente (nivel de bloque)					
Electrically Erasable PROM (EEPROM)		Eléctricamente (nivel de byte)					

Arquitectura de los Sistemas de Cómputo
8




Memoria caché

- La velocidad del procesador y de la memoria principal evolucionan de manera dispar
- Esto afecta el rendimiento, ya que la ejecución de cada instrucción requiere varios accesos a la memoria principal
- La solución de construir la memoria principal con la misma tecnología utilizada para construir el procesador debe ser descartada por el costo
- Solución: interponer una memoria rápida pero pequeña entre la CPU y la memoria principal




Arquitectura de los Sistemas de Cómputo
9



Memoria caché - Elementos del diseño

- Tamaño de la caché
- Política de escritura (actualización)
 - Write through
 - Write back
- Función de mapeo
 - Directo
 - Asociativo
 - Asociativo por conjuntos
- Tamaño del bloque
- Algoritmo de reemplazo
 - Menos recientemente usado (LRU)
 - Primero en entrar primero en salir (FIFO)
 - Menos frecuentemente usado (LFU)
 - Aleatorio (Random)

Arquitectura de los Sistemas de Cómputo 10




Operación de la memoria caché

```

graph TD
    Iniciar([Iniciar]) --> Obtener[Obtener dirección de palabra]
    Obtener --> Decision{¿Está en la caché?}
    Decision -- Sí --> RecuperarPalabra[Recuperar la palabra]
    RecuperarPalabra --> Terminar([Terminar])
    Decision -- No --> RecuperarMP[Recuperar el bloque de la MP]
    RecuperarMP --> Asignar[Asignar un slot de la caché]
    Asignar --> RecuperarPalabraMP[Recuperar la palabra]
    RecuperarPalabraMP --> Cargar[Cargar el bloque de MP en el slot]
    Cargar --> Terminar
    
```


Arquitectura de los Sistemas de Cómputo 11



Memoria caché – Mapeo directo

- Cada bloque de memoria puede ser mapeado a un único slot de la caché
- Simple y económica
- Como hay más bloques de memoria que slots, cada slot debe tener un señalador (tag) asociado que indique que bloque de memoria está cargado en ese slot en un momento dado
- Imposibilidad de utilizar estrategias de reemplazo

Arquitectura de los Sistemas de Cómputo 12




Memoria caché – Mapeo asociativo

- Cada bloque de memoria puede ser mapeado a cualquier slot de la caché
- Cada slot de la caché debe mantener un señalador (tag) asociado que indique que bloque de memoria está cargado en ese slot de la caché en un momento dado
- Dado que la búsqueda secuencial es ineficiente se debe implementar mediante una memoria asociativa (busca por contenido del señalador y no por dirección)
- Compleja y cara
- Permite la implementación de la estrategia de reemplazo más eficiente

Arquitectura de los Sistemas de Cómputo

13




Memoria caché – Mapeo asociativo por conjuntos

- La caché se divide en I conjuntos, cada uno de los cuales contiene J slots
- Un bloque de memoria es mapeado a un conjunto de slots de la caché y puede ser almacenado en cualquier slot del conjunto
- Cada slot de la caché debe mantener un señalador (tag) asociado que indique que bloque de memoria está cargado en ese slot de la caché en un momento dado
- Compleja y cara
- Permite la implementación de estrategias de reemplazo eficientes
- Normalmente el conjunto mantiene entre dos y cuatro slots

Arquitectura de los Sistemas de Cómputo

14



Memoria caché – Funciones de mapeo

Interpretación de la dirección de una palabra en memoria principal en función de los distintos métodos de mapeo.

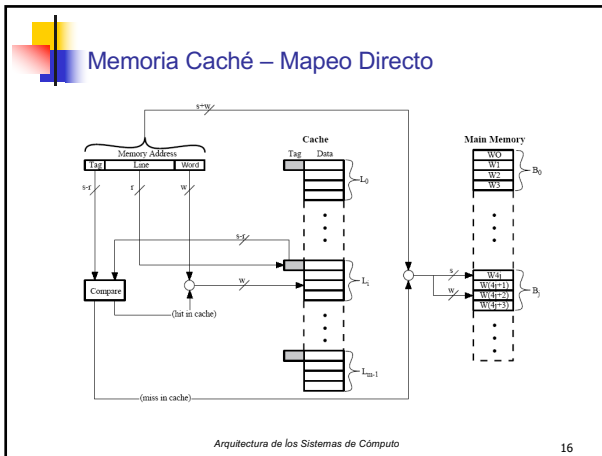
SUPUESTOS

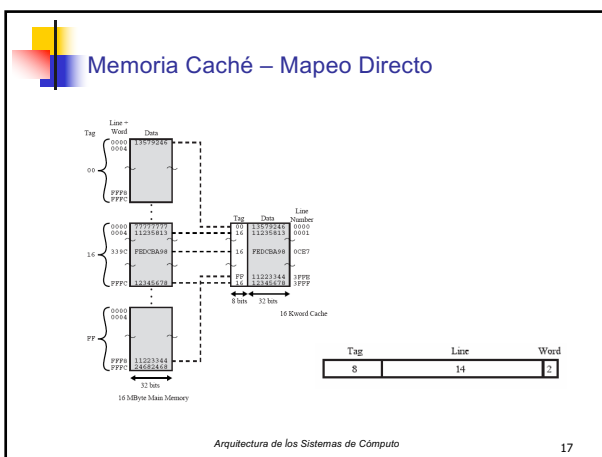
- Caché de 64 Kb.
- Slots de 4 bytes cada uno
- 16 K slots
- Memoria principal de 16 Mb.
- Direcciones de 24 bits
- Bloques de 4 bytes cada uno
- 4 M bloques

Mapeo Directo	Tag (8)	Slot (14)	W (2)
Mapeo Asociativo	Tag (22)		W (2)
Mapeo Asociativo por conjuntos (2 slots por conjunto)	Tag (9)	Set (13)	W (2)

Arquitectura de los Sistemas de Cómputo

15





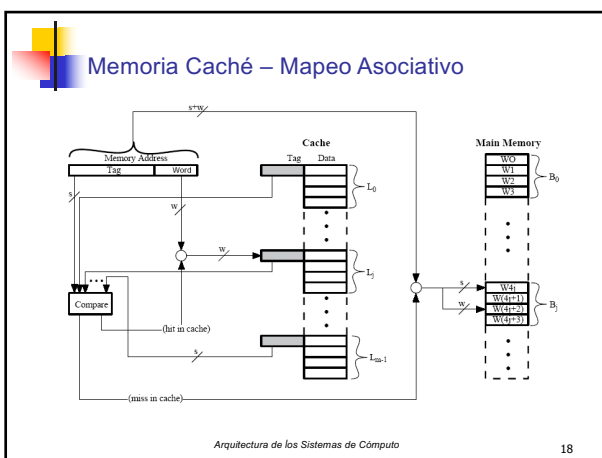


Diagram illustrating the mapping of memory addresses to cache lines in a 16 MByte Main Memory and a 16 Kword Cache.

Memory Address (16 bits):

- Address 000000 to 000004
- Address 163398 to 1633A7
- Address 1633B0 to 1633B7
- Address 1633C0 to 1633C7
- Address 1633D0 to 1633D7
- Address 1633E0 to 1633E7
- Address 1633F0 to 1633F7
- Address 163400 to 163407
- Address 163410 to 163417
- Address 163420 to 163427
- Address 163430 to 163437
- Address 163440 to 163447
- Address 163450 to 163457
- Address 163460 to 163467
- Address 163470 to 163477
- Address 163480 to 163487
- Address 163490 to 163497
- Address 1634A0 to 1634A7
- Address 1634B0 to 1634B7
- Address 1634C0 to 1634C7
- Address 1634D0 to 1634D7
- Address 1634E0 to 1634E7
- Address 1634F0 to 1634F7
- Address 163500 to 163507
- Address 163510 to 163517
- Address 163520 to 163527
- Address 163530 to 163537
- Address 163540 to 163547
- Address 163550 to 163557
- Address 163560 to 163567
- Address 163570 to 163577
- Address 163580 to 163587
- Address 163590 to 163597
- Address 1635A0 to 1635A7
- Address 1635B0 to 1635B7
- Address 1635C0 to 1635C7
- Address 1635D0 to 1635D7
- Address 1635E0 to 1635E7
- Address 1635F0 to 1635F7
- Address 163600 to 163607
- Address 163610 to 163617
- Address 163620 to 163627
- Address 163630 to 163637
- Address 163640 to 163647
- Address 163650 to 163657
- Address 163660 to 163667
- Address 163670 to 163677
- Address 163680 to 163687
- Address 163690 to 163697
- Address 1636A0 to 1636A7
- Address 1636B0 to 1636B7
- Address 1636C0 to 1636C7
- Address 1636D0 to 1636D7
- Address 1636E0 to 1636E7
- Address 1636F0 to 1636F7
- Address 163700 to 163707
- Address 163710 to 163717
- Address 163720 to 163727
- Address 163730 to 163737
- Address 163740 to 163747
- Address 163750 to 163757
- Address 163760 to 163767
- Address 163770 to 163777
- Address 163780 to 163787
- Address 163790 to 163797
- Address 1637A0 to 1637A7
- Address 1637B0 to 1637B7
- Address 1637C0 to 1637C7
- Address 1637D0 to 1637D7
- Address 1637E0 to 1637E7
- Address 1637F0 to 1637F7
- Address 163800 to 163807
- Address 163810 to 163817
- Address 163820 to 163827
- Address 163830 to 163837
- Address 163840 to 163847
- Address 163850 to 163857
- Address 163860 to 163867
- Address 163870 to 163877
- Address 163880 to 163887
- Address 163890 to 163897
- Address 1638A0 to 1638A7
- Address 1638B0 to 1638B7
- Address 1638C0 to 1638C7
- Address 1638D0 to 1638D7
- Address 1638E0 to 1638E7
- Address 1638F0 to 1638F7
- Address 163900 to 163907
- Address 163910 to 163917
- Address 163920 to 163927
- Address 163930 to 163937
- Address 163940 to 163947
- Address 163950 to 163957
- Address 163960 to 163967
- Address 163970 to 163977
- Address 163980 to 163987
- Address 163990 to 163997
- Address 1639A0 to 1639A7
- Address 1639B0 to 1639B7
- Address 1639C0 to 1639C7
- Address 1639D0 to 1639D7
- Address 1639E0 to 1639E7
- Address 1639F0 to 1639F7
- Address 163A00 to 163A07
- Address 163A10 to 163A17
- Address 163A20 to 163A27
- Address 163A30 to 163A37
- Address 163A40 to 163A47
- Address 163A50 to 163A57
- Address 163A60 to 163A67
- Address 163A70 to 163A77
- Address 163A80 to 163A87
- Address 163A90 to 163A97
- Address 163AA0 to 163AA7
- Address 163AB0 to 163AB7
- Address 163AC0 to 163AC7
- Address 163AD0 to 163AD7
- Address 163AE0 to 163AE7
- Address 163AF0 to 163AF7
- Address 163B00 to 163B07
- Address 163B10 to 163B17
- Address 163B20 to 163B27
- Address 163B30 to 163B37
- Address 163B40 to 163B47
- Address 163B50 to 163B57
- Address 163B60 to 163B67
- Address 163B70 to 163B77
- Address 163B80 to 163B87
- Address 163B90 to 163B97
- Address 163BA0 to 163BA7
- Address 163BB0 to 163BB7
- Address 163BC0 to 163BC7
- Address 163BD0 to 163BD7
- Address 163BE0 to 163BE7
- Address 163BF0 to 163BF7
- Address 163C00 to 163C07
- Address 163C10 to 163C17
- Address 163C20 to 163C27
- Address 163C30 to 163C37
- Address 163C40 to 163C47
- Address 163C50 to 163C57
- Address 163C60 to 163C67
- Address 163C70 to 163C77
- Address 163C80 to 163C87
- Address 163C90 to 163C97
- Address 163CA0 to 163CA7
- Address 163CB0 to 163CB7
- Address 163CC0 to 163CC7
- Address 163CD0 to 163CD7
- Address 163CE0 to 163CE7
- Address 163CF0 to 163CF7
- Address 163D00 to 163D07
- Address 163D10 to 163D17
- Address 163D20 to 163D27
- Address 163D30 to 163D37
- Address 163D40 to 163D47
- Address 163D50 to 163D57
- Address 163D60 to 163D67
- Address 163D70 to 163D77
- Address 163D80 to 163D87
- Address 163D90 to 163D97
- Address 163DA0 to 163DA7
- Address 163DB0 to 163DB7
- Address 163DC0 to 163DC7
- Address 163DD0 to 163DD7
- Address 163DE0 to 163DE7
- Address 163DF0 to 163DF7
- Address 163E00 to 163E07
- Address 163E10 to 163E17
- Address 163E20 to 163E27
- Address 163E30 to 163E37
- Address 163E40 to 163E47
- Address 163E50 to 163E57
- Address 163E60 to 1

19

[illegible][illegible]

20

[illegible]

Memoria Caché – Mapeo Asociativo por Conjuntos

The diagram illustrates a 16-Byte Main Memory and a 16-Kword Cache. The memory is divided into three sections: 0000-0004 (Set=Word), 0004-000C (Set=Word, Data), and 000C-0010 (Set=Word, Data). The cache is divided into two sections: 0000-0004 (Set=Word, Data) and 0004-000C (Set=Word, Data). The diagram shows the mapping of memory addresses to cache entries and the resulting data flow.

Memory Layout:

- 0000-0004 (Set=Word):** Contains 5 words (0000, 0004, 0008, 000C, 0010). Address 0000 points to 12345678.
- 0004-000C (Set=Word, Data):** Contains 5 words (0004, 0008, 000C, 0010, 0014). Address 0004 points to 12345678. Address 0008 points to FEDCBA98. Address 000C points to 12345678.
- 000C-0010 (Set=Word, Data):** Contains 5 words (000C, 0010, 0014, 0018, 001C). Address 000C points to 12345678. Address 0010 points to 12345678. Address 0014 points to 12345678. Address 0018 points to 12345678. Address 001C points to 12345678.

Cache Layout:

- 0000-0004 (Set=Word, Data):** Contains 5 words (0000, 0004, 0008, 000C, 0010). Address 0000 points to 12345678. Address 0004 points to 12345678. Address 0008 points to FEDCBA98. Address 000C points to 12345678. Address 0010 points to 12345678.
- 0004-000C (Set=Word, Data):** Contains 5 words (0004, 0008, 000C, 0010, 0014). Address 0004 points to 12345678. Address 0008 points to FEDCBA98. Address 000C points to 12345678. Address 0010 points to 12345678. Address 0014 points to 12345678.

Address Mapping:

- 0000: 12345678
- 0004: 12345678
- 0008: FEDCBA98
- 000C: 12345678
- 0010: 12345678
- 0014: 12345678
- 0018: 12345678
- 001C: 12345678

Cache Mapping:

- 0000: 12345678
- 0004: 12345678
- 0008: FEDCBA98
- 000C: 12345678
- 0010: 12345678
- 0014: 12345678
- 0018: 12345678
- 001C: 12345678

Address Flow:

- 0000: 12345678
- 0004: 12345678
- 0008: FEDCBA98
- 000C: 12345678
- 0010: 12345678
- 0014: 12345678
- 0018: 12345678
- 001C: 12345678

Cache Flow:

- 0000: 12345678
- 0004: 12345678
- 0008: FEDCBA98
- 000C: 12345678
- 0010: 12345678
- 0014: 12345678
- 0018: 12345678
- 001C: 12345678

Address Flow:

- 0000: 12345678
- 0004: 12345678
- 0008: FEDCBA98
- 000C: 12345678
- 0010: 12345678
- 0014: 12345678
- 0018: 12345678
- 001C: 12345678

Cache Flow:

- 0000: 12345678
- 0004: 12345678
- 0008: FEDCBA98
- 000C: 12345678
- 0010: 12345678
- 0014: 12345678
- 0018: 12345678
- 001C: 12345678

Address Flow:

- 0000: 12345678
- 0004: 12345678
- 0008: FEDCBA98
- 000C: 12345678
- 0010: 12345678
- 0014: 12345678
- 0018: 12345678
- 001C: 12345678

Cache Flow:

- 0000: 12345678
- 0004: 12345678
- 0008: FEDCBA98
- 000C: 12345678
- 0010: 12345678
- 0014: 12345678
- 0018: 12345678
- 001C: 12345678

Address Flow:

- 0000: 12345678
- 0004: 12345678
- 0008: FEDCBA98
- 000C: 12345678
- 0010: 12345678
- 0014: 12345678
- 0018: 12345678
- 001C: 12345678

Cache Flow:

- 0000: 12345678
- 0004: 12345678
- 0008: FEDCBA98
- 000C: 12345678
- 0010: 12345678
- 0014: 12345678
- 0018: 12345678
- 001C: 12345678

Address Flow:

- 0000: 12345678
- 0004: 12345678
- 0008: FEDCBA98
- 000C: 12345678
- 0010: 12345678
- 0014: 12345678
- 0018: 12345678
- 001C: 12345678

Cache Flow:

- 0000: 12345678
- 0004: 12345678
- 0008: FEDCBA98
- 000C: 12345678
- 0010: 12345678
- 0014: 12345678
- 0018: 12345678
- 001C: 12345678

Address Flow:

- 0000: 12345678
- 0004: 12345678
- 0008: FEDCBA98
- 000C: 12345678
- 0010: 12345678
- 0014: 12345678
- 0018: 12345678
- 001C: 12345678

Cache Flow:

- 0000: 12345678
- 0004: 12345678
- 0008: FEDCBA98
- 000C: 12345678
- 0010: 12345678
- 0014: 12345678
- 0018: 12345678
- 001C: 12345678

Address Flow:

- 0000: 12345678
- 0004: 12345678
- 0008: FEDCBA98
- 000C: 12345678
- 0010: 12345678
- 0014: 12345678
- 0018: 12345678
- 001C: 12345678

Cache Flow:

- 0000: 12345678
- 0004: 12345678
- 0008: FEDCBA98
- 000C: 12345678
- 0010: 12345678
- 0014: 12345678
- 0018: 12345678
- 001C: 12345678


Address Flow:

- 0000: 12345678
- 0004: 12345678
- 0008: FEDCBA98
- 000C: 12345678
- 0010: 12345678
- 0014: 12345678
- 0018: 12345678
- 001C: 12345678

Cache Flow:

- 0000: 12345678
- 0004: 12345678
- 0008: FEDCBA98
- 000C: 12345678
- 0010: 12345678
- 0014: 12345678
- 0018: 12345678


21



Memoria caché – Estrategias de reemplazo

- Mapeo Directo
 - Al haber un único slot en el que puede cargarse el bloque no hay posibilidad de implementar estrategias de reemplazo.
- Mapeos Asociativo y Asociativo por Conjuntos
 - **LRU (Least-recently used)**: Reemplaza el bloque que no ha sido referenciado por más tiempo
 - Se agrega a cada slot un bit de uso que se pone a uno cada vez que el slot es referenciado. Periódicamente todos se ponen a cero.
 - **FIFO (First-in-first-out)**: Reemplaza el bloque que hace más tiempo que está en la caché
 - Puede implementarse fácilmente con una técnica de buffer circular
 - **LFU (Least-frequently used)**: Reemplaza el bloque que ha sido usado menos veces
 - Se debe asignar un contador a cada slot que se pone a cero cuando se carga un nuevo bloque y se incrementa con cada referencia al bloque
 - **Random**: Reemplaza un bloque elegido al azar


Arquitectura de los Sistemas de Cómputo
22



Memoria caché – Políticas de escritura

- Se refiere a la política a seguir cuando un bloque que está en la caché es modificado
- Los problemas surgen en un equipo con DMA y en un contexto con múltiples procesadores
- Las políticas que normalmente se siguen son:
 - **Write through**: Cualquier actualización hecha a un bloque en la caché es inmediatamente realizada en el bloque de la memoria principal. Se asegura así que el contenido de la memoria principal siempre es válido.
 - **Write back**: Las actualizaciones son hechas exclusivamente en la caché y sólo son transferidas a la memoria principal cuando el bloque es reemplazado en la caché. Por momentos el contenido de la memoria principal puede ser inválido.


Arquitectura de los Sistemas de Cómputo
23



Memoria caché – Coherencia

- Es un problema que surge en una organización en bus cuando los dispositivos conectados al bus tienen cada uno su propia caché.
- Si una palabra en una caché es modificada, se invalida no sólo el contenido de la memoria principal sino también el de otras cachés.
- Las políticas que normalmente se siguen son:
 - **Bus Watching with Write Through**: Cada caché monitorea el bus para detectar referencias de escritura a palabras de memoria que están en ellas e invalidarlas cuando así ocurre
 - **Hardware transparency**: Se agrega hardware para asegurar que las actualizaciones a la memoria principal vía una caché se reflejen en todas las cachés
 - **Non-cacheable Memory**: Sólo una porción de la memoria principal es compartida por más de un procesador y esa porción de memoria no puede llevarse a la caché.

Arquitectura de los Sistemas de Cómputo
24




Memoria caché – Tamaño del bloque

- Bloques más grandes reducen el número de bloques que entran en la caché, lo que puede obligar a su reemplazo poco después de haber sido cargados.
- Cuando el tamaño del bloque crece cada palabra adicional está más lejos de la palabra requerida que provocó la carga en la caché y, en consecuencia, disminuye la probabilidad de que sea utilizada en el futuro cercano.
- No hay un tamaño óptimo, ya que éste depende de las características de localidad de cada proceso particular
- Normalmente se usa un tamaño de entre 4 y 8 unidades direccionables (palabras o bytes)

Arquitectura de los Sistemas de Cómputo

25




Discos Magnéticos

- Son platos contruidos de metal o plástico recubiertos con un material magnetizable.
- Los datos son grabados o leídos mediante una cabeza de lectura escritura
- Durante las operaciones de lectura la cabeza está estacionaria y el plato gira debajo de ella

- Conceptos
 - Pista (track)
 - Cilindro (cylinder)
 - Inter Track Gap (ITG)
 - Sectores o bloques
 - Longitud fija
 - Longitud variable
 - Inter Block Gap (IBG)
 - ID Field
 - SYNCH byte
 - Densidad (bits per inch)
 - Tiempo de acceso
 - Demora de rotación
 - Velocidad de transferencia
 - Factor de intercalado

Arquitectura de los Sistemas de Cómputo

26



Discos Magnéticos - Características

- Movilidad de las cabezas de lectura / escritura
 - Fijas
 - Móviles
- Portabilidad
 - Removibles
 - No removibles (fijos)
- Mecánica de la cabezas de lectura / escritura
 - Contacto (floppy)
 - Separación fija
 - Separación aerodinámica

- Platos
 - Uno
 - Varios (disk pack)
- Lados
 - Simple (uno solo)
 - Doble (ambos)

Arquitectura de los Sistemas de Cómputo

27

Discos Magnéticos – Estructura general

The diagram illustrates the general structure of a magnetic disk. A central vertical spindle is labeled 'Spindle'. Ten circular disk surfaces are mounted on the spindle, labeled 'Surface 0' at the bottom to 'Surface 9' at the top. Each surface is separated by a thin layer of 'Plaster'. A 'Read/write head (1 per surface)' is positioned above the surfaces, with an arrow indicating the 'Direction of arm motion' (left and right). A vertical 'Boom' supports the head assembly. The surfaces are numbered 0 through 9 from bottom to top.

Arquitectura de los Sistemas de Cómputo

28

28

[illegible]

Discos Magnéticos – Estructura de la superficie

El diagrama ilustra la estructura física de un disco magnético. Se muestran tres pistas concéntricas, cada una dividida en sectores. Los sectores están separados por huecos intersectoriales (Inter-sector gap). Las pistas están separadas por huecos interpistas (Inter-track gap). Las pistas están etiquetadas como S_0 , S_1 y S_2 . Los sectores están etiquetados como T_0 , T_1 y T_2 . Los huecos intersectoriales están etiquetados como G_0 , G_1 y G_2 . Los huecos interpistas están etiquetados como H_0 , H_1 y H_2 .

Arquitectura de los Sistemas de Cómputo

29

29

Discos Magnéticos – Formato de pista

The diagram illustrates the format of a magnetic disk track, showing sectors 0, 1, and 29. Each sector is divided into five fields: Gap, ID Field, Cap, Data Field, and Gap. The sizes of these fields are specified in bytes. Sectors 0 and 1 are expanded to show their internal structure, including Sync Byte, Track #, Head #, Sector #, and CRC. The total size of the track is 600 Bytes/Sector.

Sector	Field	Bytes
Physical Sector 0	Cap 1	17
	ID Field 0	7
	Cap 2	41
	Data Field 0	515
	Cap 3	20
Physical Sector 1	Cap 1	17
	ID Field 1	7
	Cap 2	41
	Data Field 1	515
	Cap 3	20
Physical Sector 29	Cap 1	17
	ID Field 29	7
	Cap 2	41
	Data Field 29	515
	Cap 3	20

600 Bytes/Sector

Expanded Sector 0 Structure:

Field	Bytes
Sync Byte	1
Track #	2
Head #	1
Sector #	1
CRC	2

Expanded Sector 1 Structure:

Field	Bytes
Sync Byte	1
Data	512
CRC	2

30
