

# 66.70 Estructura del Computador

## **Memoria**

- Memoria volátil y no volátil
- RAM
- ROM
- Disco rígido
- Pendrive
- CD-ROM, DVD-ROM
- Cinta

# Clasificación por el modo de acceso

- **RAM:** M. de Acceso Aleatorio

El tiempo y procedimiento para el acceso es independiente de la dirección accedida

- RWM: M. de Lectura/Escritura, en gral. referida como RAM
- ROM: M. de Sólo Lectura

- **CAM:** M. Direccionable por Contenido (o Asociativa)

- **SAM:** M. de Acceso Secuencial

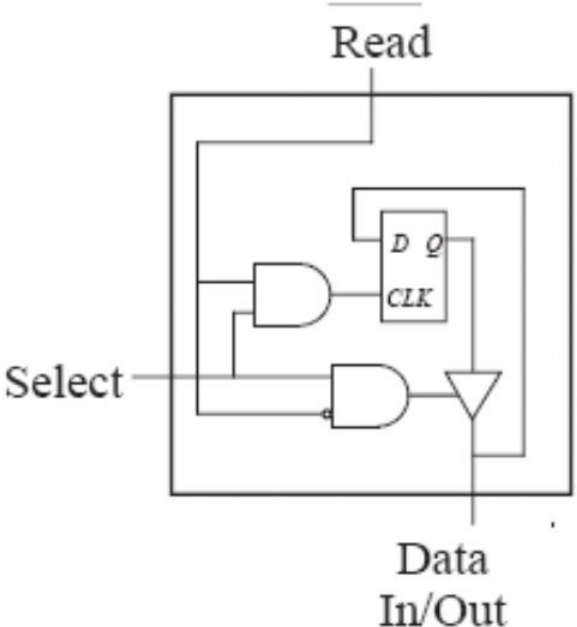
Cinta, Reg. de desplazamiento, Stack

- **DAM:** M. de Acceso Directo o Semi-Aleatorio

Disco rígido: acceso aleatorio a la pista y secuencial dentro de esta

# Memoria de lectura/escritura

## Celda de 1 bit

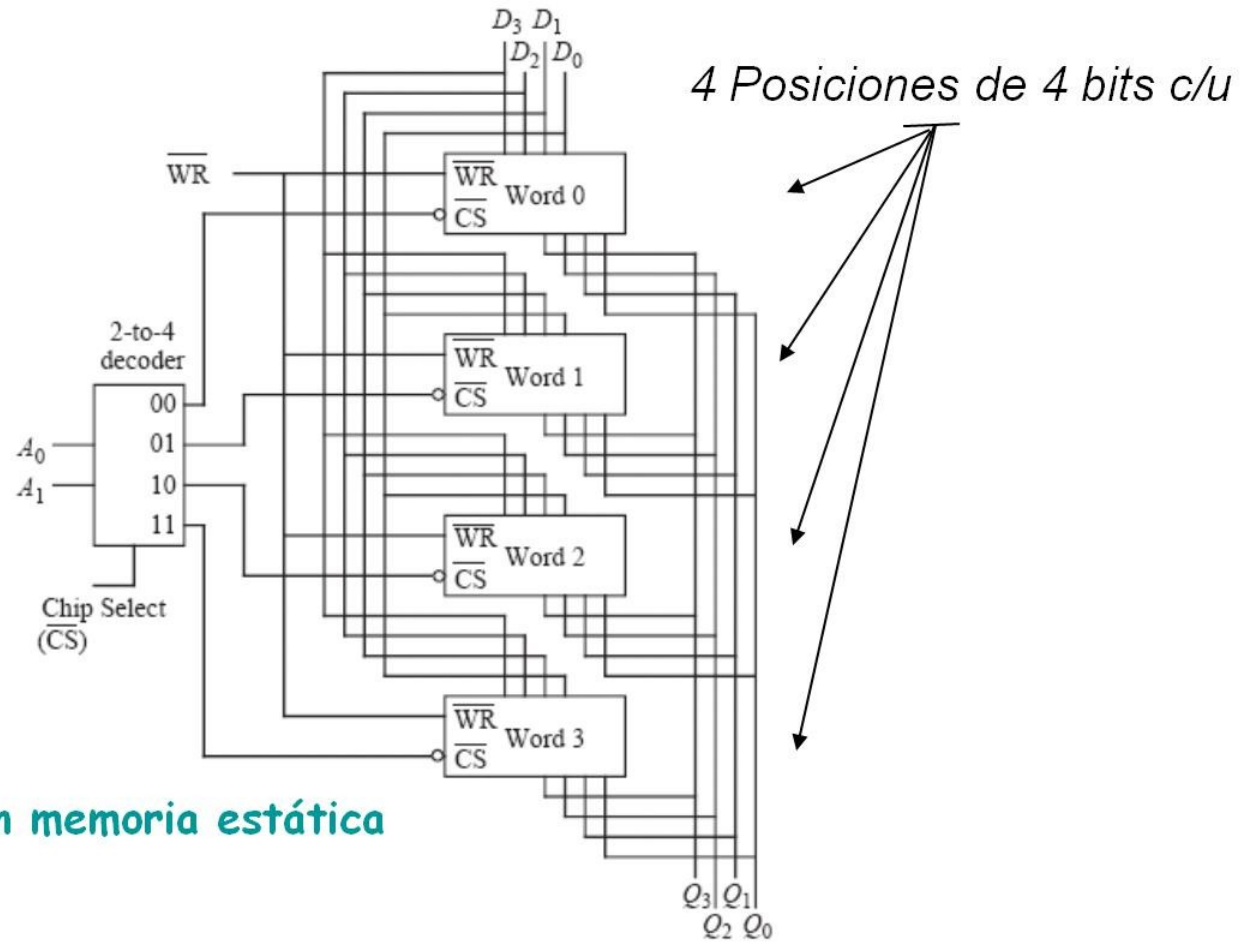
	<p>→ Memoria Estática</p> <p>→ Memoria Dinámica</p> <p><i>lect./escri.</i></p> <p><i>refresh</i></p>
<p><i>Bloque <u>funcional</u></i></p>	<p><i>Formas de implementación</i></p>

# RAM dinámica

- *DRAM fue patentada en 1968.*
- *1 o 0 lógico  $\Leftrightarrow$  capacitor cargado o descargado*
- *Significativamente más lenta que RAM estática*
- *Significativamente más barata que RAM estática:*
  - *(1 transistor + 1 capacitor) VS. (6 transistores)*
  - *Permite lograr alta densidad de elementos de memoria*
  - *apropiada para grandes volúmenes de memoria*

# El problema del Direccionamiento

## Organización "2-D"

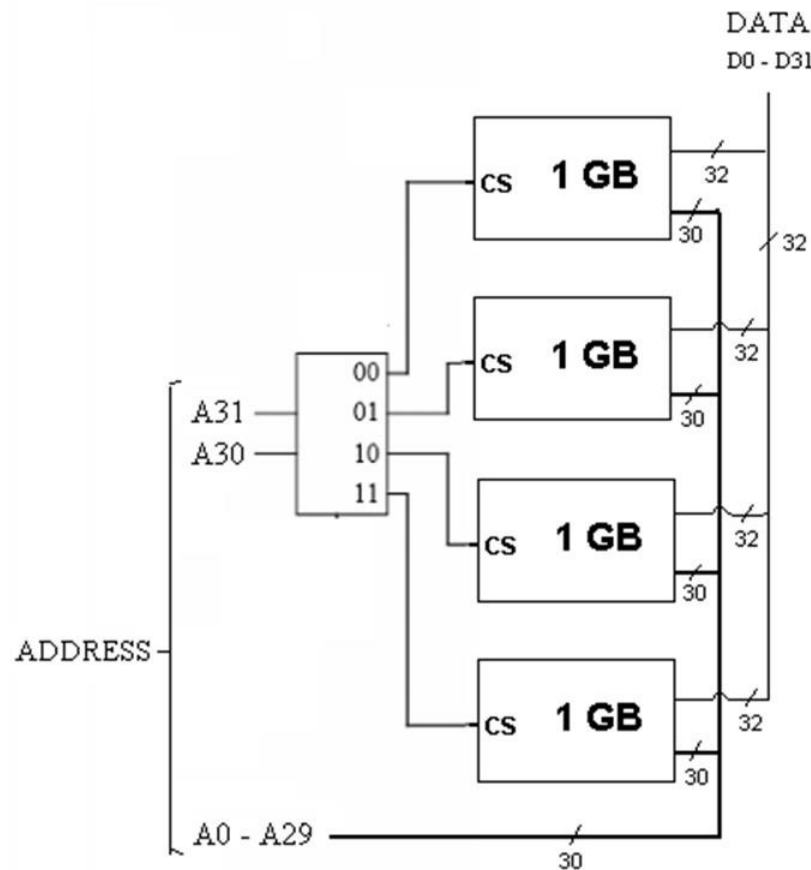


Se suele usar con memoria estática

¿Qué limitaciones aparecen con este tipo de organización?  
p.e. con 512M x 8 bits

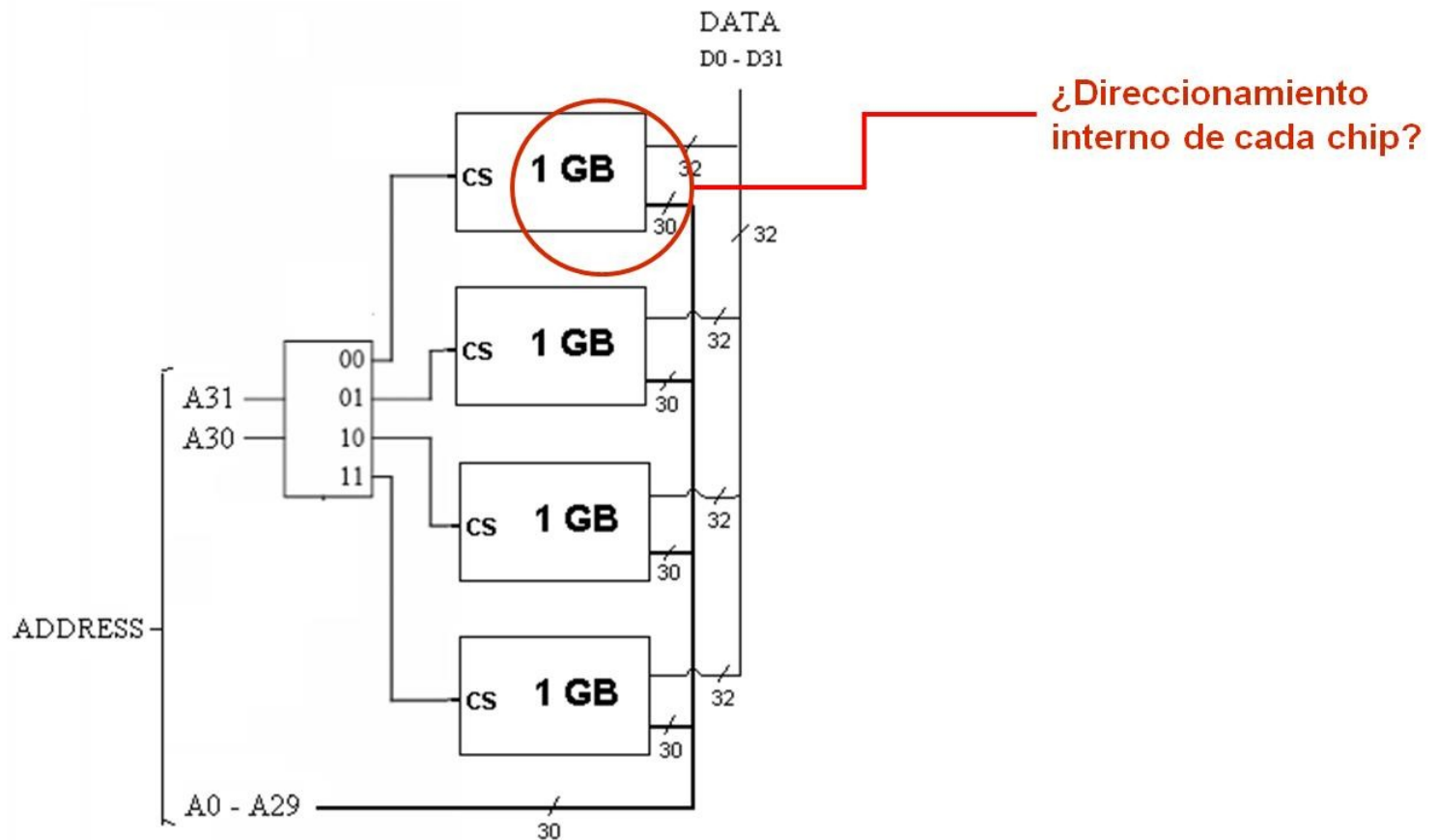
# El problema del Direcccionamiento

## Organización en “2-D” con varios chips de memoria



# El problema del Direcccionamiento

## Organización en “2-D” con varios chips de memoria

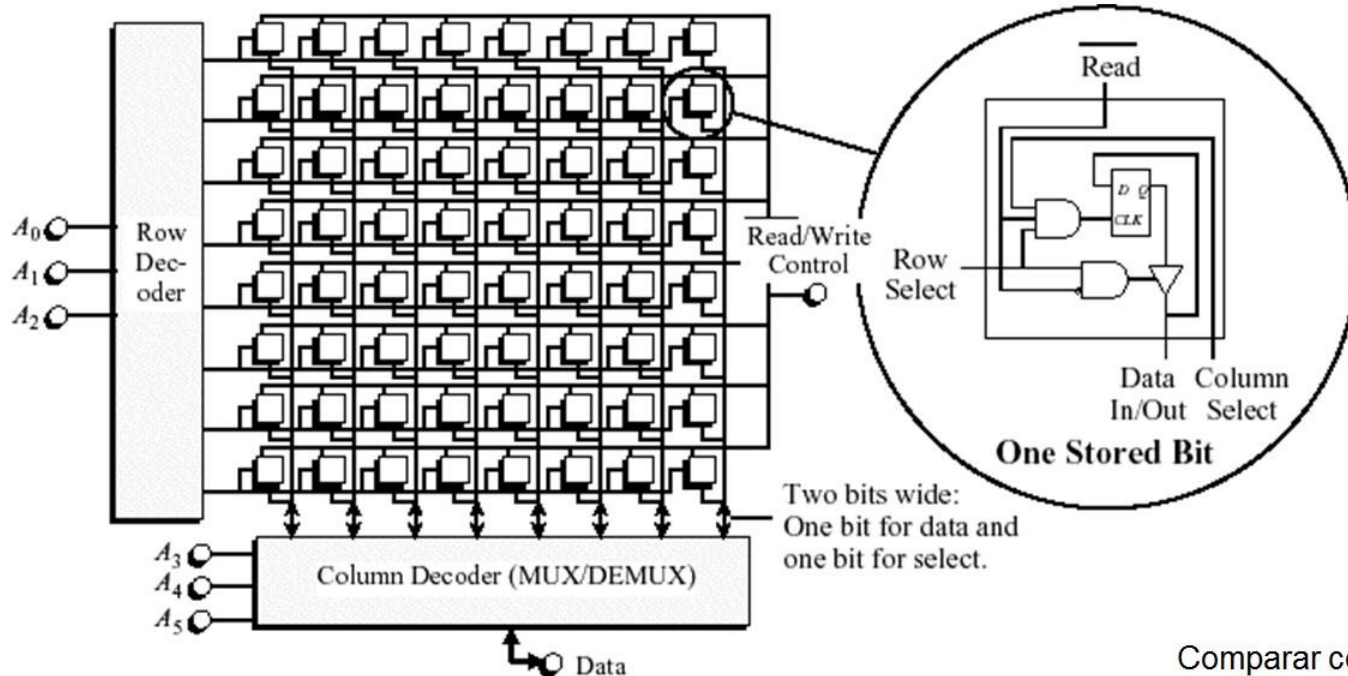




# El problema del Direcccionamiento

## Organización “2-1/2 D”

Módulo de 64 posiciones de **1 bit**



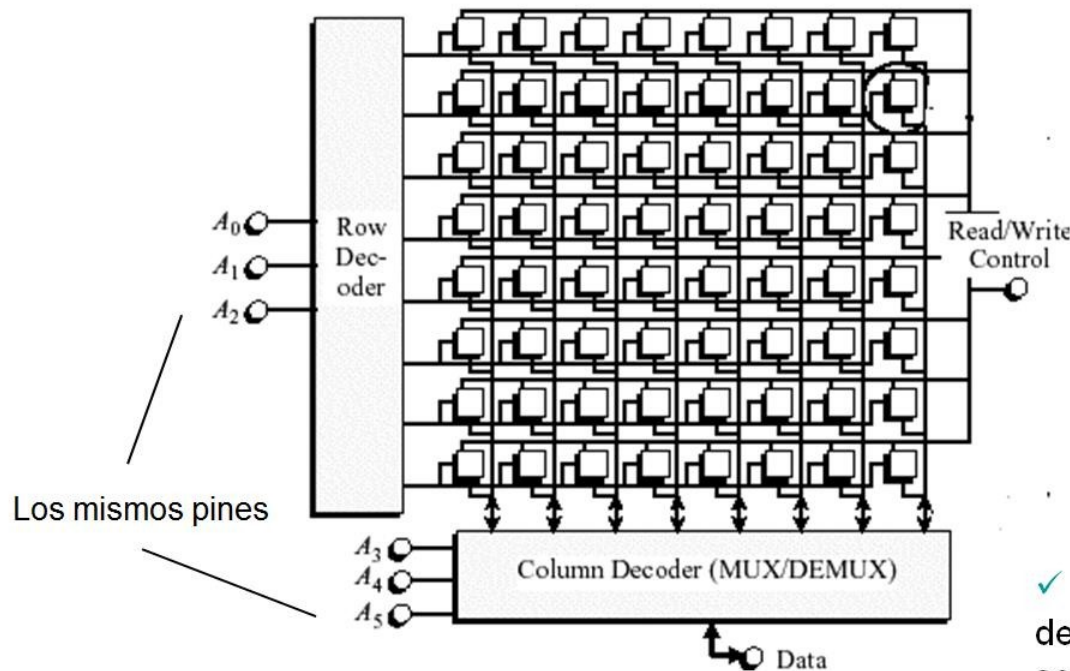
Comparar con celda de 1 bit en transparencia anterior

- $n$  bits de address  $\Rightarrow n/2$  bits en filas y  $n/2$  bits en columnas

# El problema del Direcccionamiento

## Organización “2-1/2 D”

Módulo de 64 posiciones de **1 bit**



- ✓ Menor número de pines
- ✓ Menor tiempo de acceso de palabras sucesivas (decodifica sólo columna)

- ✓ Los 8 bits correspondientes una posición de memoria están en una “3ra dimensión” y son seleccionados simultáneamente

- $n$  bits de address  $\Rightarrow n/2$  bits en filas y  $n/2$  bits en columnas **+ multiplexado en el tiempo**

Direccionar 1 bit  $\Rightarrow$  (a) Seleccionar fila (b) Seleccionar columna

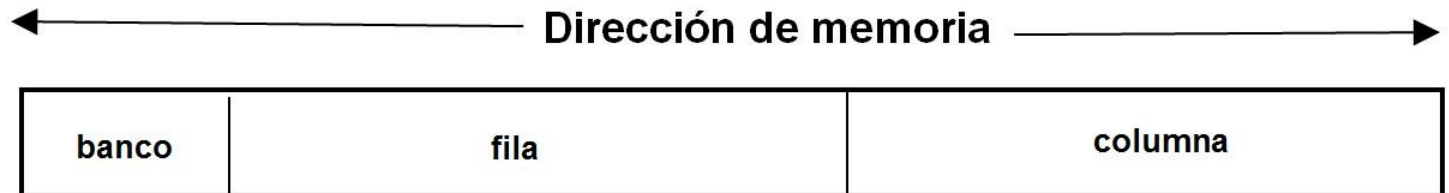
n° de fila en guardado en registro y posterior decodificación de columna



# Mapeo de memoria dentro del módulo de DRAM

## Arquitectura multibanco

- Un sólo módulo de memoria está conformado por varios chips
- Todos los bits que definen una dirección de memoria dentro del módulo se dividen según:



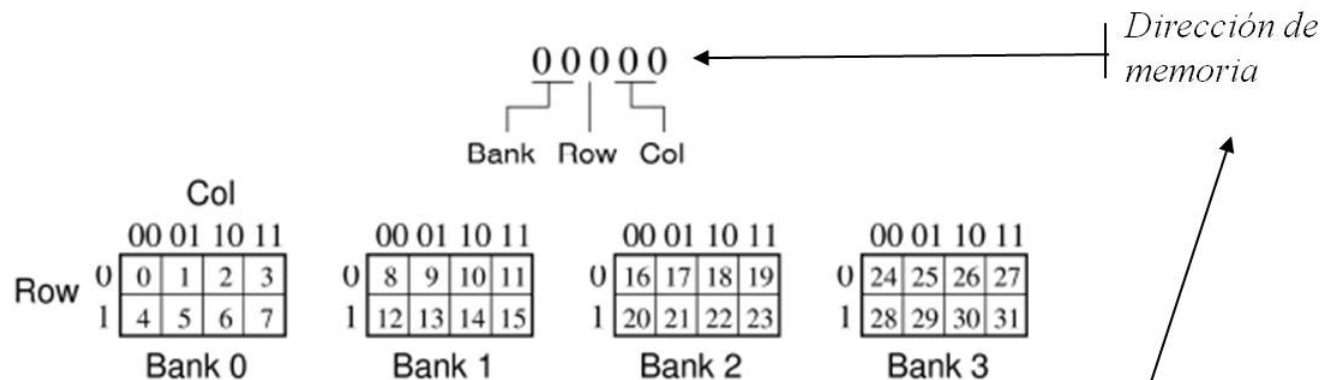
- Cada chip es especificado por los bits “banco”
- Dentro de cada chip cada bit es especificado por su fila y su columna



# Mapecto de memoria dentro del m3dulo de DRAM

## Arquitectura multibanco

Ejemplo tutorial de un m3dulo de memoria compuesto por  
4 m3dulos de 8 bits cada uno



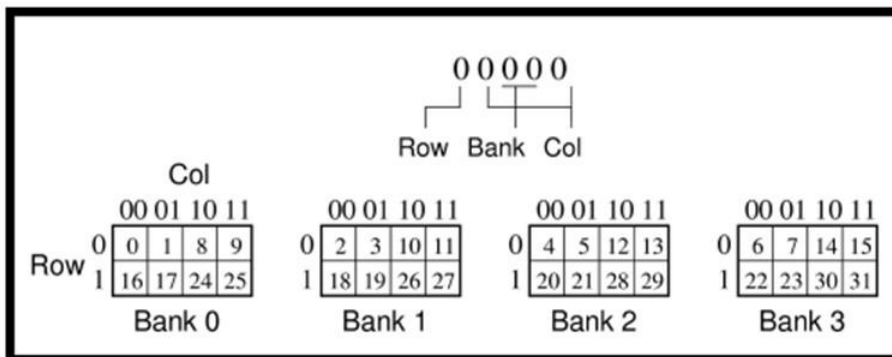
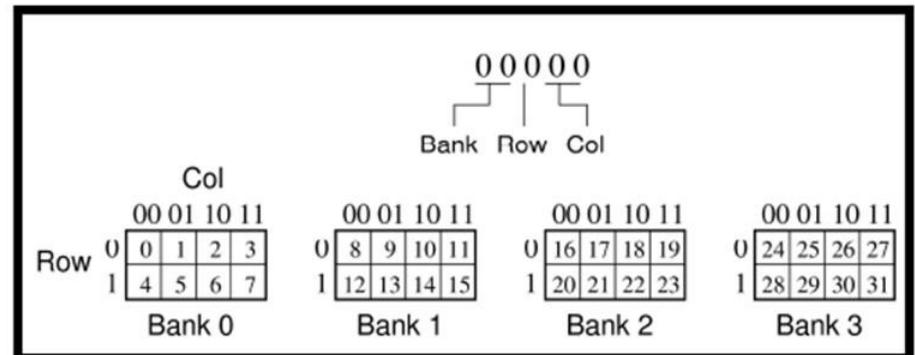
- Total de memoria del m3dulo 32 bits => la direcci3n es definida por 5 bits
- La direcci3n de memoria se interpreta como dividida en los campos Banco, Fila y Columna
- Una direcci3n de memoria espec3fica identifica uno de los 32 bits (0..31) y este puede estar en cualquiera de los cuatro m3dulos



# Mapeo de memoria dentro del módulo de DRAM

## Arquitectura multibanco

*Acceso secuencial a los bancos*



*Bancos entrelazados*

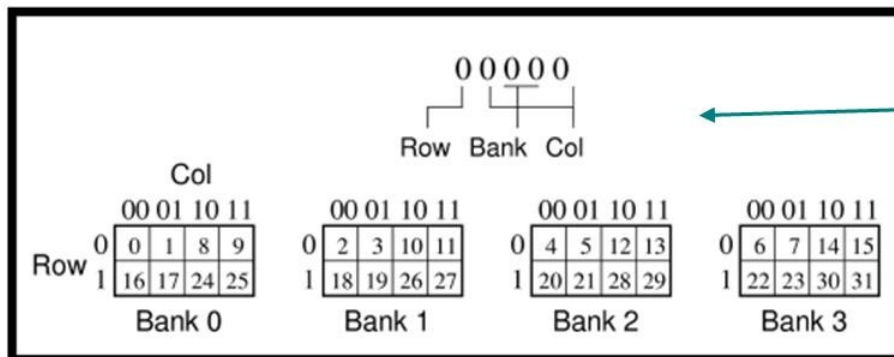
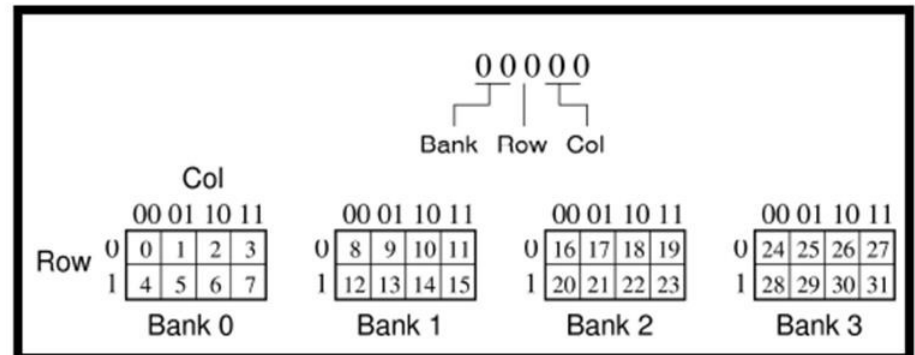




# Mapeo de memoria dentro del módulo de DRAM

## Arquitectura multibanco

*Acceso secuencial a los bancos*



- ✓ Lee 2 bits consecutivos de un banco mientras el otro es accedido para refrescar la información (enmascara el tiempo de refresco)
- ✓ Mejora rendimiento si pos. sucesivas están en bancos diferentes (ver "localidad espacial")

*Bancos entrelazados*

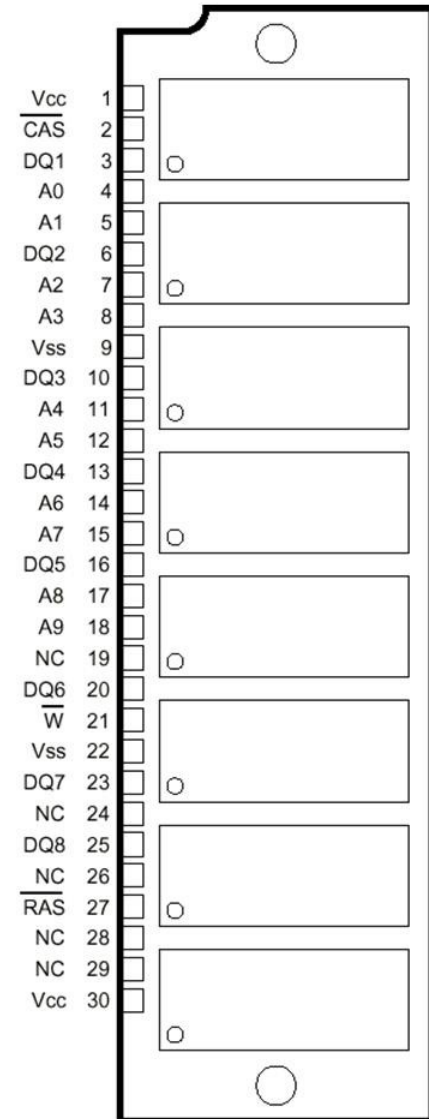
# Módulos de memoria comerciales

PIN NOMENCLATURE	
A0-A9	Address Inputs
$\overline{\text{CAS}}$	Column-Address Strobe
DQ1-DQ8	Data In/Data Out
NC	No Connection
$\overline{\text{RAS}}$	Row-Address Strobe
$V_{\text{CC}}$	5-V Supply
$V_{\text{SS}}$	Ground
$\overline{\text{W}}$	Write Enable

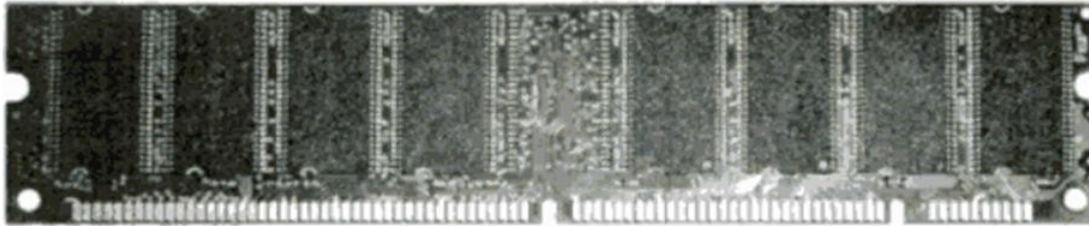
- ✓ 8 chips de 1 Mbit
- ✓ DRAM
- ✓ SIMM 30 pines
- ✓  $V_{\text{CC}}$  3.3 - 2.5 - 1.8 V (menor  $V_{\text{CC}}$ , mayor velocidad-menor consumo)

- Señales RAS y CAS

¿Cuál es su capacidad de memoria?  
 ¿Cómo direcciona cada posición de memoria?



# Módulos de memoria comerciales



- ✓ DRAM
- ✓ DIMM 168 pines (84 x 2)
- ✓ Organizada en palabras de 64 bits
- ✓  $DQ_0 \dots DQ_{63}$ : 8 bytes leídos en paralelo
- ✓  $A_0 \dots A_{12}$ : direccionamiento
- ✓ WE: habilita escritura
- ✓ 16 pines con  $V_{SS}$  y 16 pines con  $V_{DD}$

PIN NO.	PIN NAME	PIN NO.	PIN NAME	PIN NO.	PIN NAME	PIN NO.	PIN NAME
1	$V_{SS}$	43	$V_{SS}$	85	$V_{SS}$	127	$V_{SS}$
2	DQ0	44	OE2	86	DQ32	128	NC
3	DQ1	45	RAS2	87	DQ33	129	RAS3
4	DQ2	46	CAS2	88	DQ34	130	CAS6
5	DQ3	47	CAS3	89	DQ35	131	CAS7
6	$V_{DD}$	48	WE2	90	$V_{DD}$	132	NC
7	DQ4	49	$V_{DD}$	91	DQ36	133	$V_{DD}$
8	DQ5	50	NC	92	DQ37	134	NC
9	DQ6	51	NC	93	DQ38	135	NC
10	DQ7	52	NC	94	DQ39	136	NC
11	DQ8	53	NC	95	DQ40	137	NC
12	$V_{SS}$	54	$V_{SS}$	96	$V_{SS}$	138	$V_{SS}$
13	DQ9	55	DQ16	97	DQ41	139	DQ48
14	DQ10	56	DQ17	98	DQ42	140	DQ49
15	DQ11	57	DQ18	99	DQ43	141	DQ50
16	DQ12	58	DQ19	100	DQ44	142	DQ51
17	DQ13	59	$V_{DD}$	101	DQ45	143	$V_{DD}$
18	$V_{DD}$	60	DQ20	102	$V_{DD}$	144	DQ52
19	DQ14	61	NC	103	DQ46	145	NC
20	DQ15	62	NC	104	DQ47	146	NC
21	NC	63	NC	105	NC	147	NC
22	NC	64	$V_{SS}$	106	NC	148	$V_{SS}$
23	$V_{SS}$	65	DQ21	107	$V_{SS}$	149	DQ53
24	NC	66	DQ22	108	NC	150	DQ54
25	NC	67	DQ23	109	NC	151	DQ55
26	$V_{DD}$	68	$V_{SS}$	110	$V_{DD}$	152	$V_{SS}$
27	WE0	69	DQ24	111	NC	153	DQ56
28	CAS0	70	DQ25	112	CAS4	154	DQ57
29	CAS1	71	DQ26	113	CAS5	155	DQ58
30	RAS0	72	DQ27	114	RAS1	156	DQ59
31	OE0	73	$V_{DD}$	115	NC	157	$V_{DD}$
32	$V_{SS}$	74	DQ28	116	$V_{SS}$	158	DQ60
33	A0	75	DQ29	117	A1	159	DQ61
34	A2	76	DQ30	118	A3	160	DQ62
35	A4	77	DQ31	119	A5	161	DQ63
36	A6	78	$V_{SS}$	120	A7	162	$V_{SS}$
37	A8	79	NC	121	A9	163	NC
38	A10	80	NC	122	A11	164	NC
39	A12	81	NC	123	NC	165	SA0
40	$V_{DD}$	82	SDA	124	$V_{DD}$	166	SA1
41	NC	83	SCL	125	NC	167	SA2
42	NC	84	$V_{DD}$	126	NC	168	$V_{DD}$



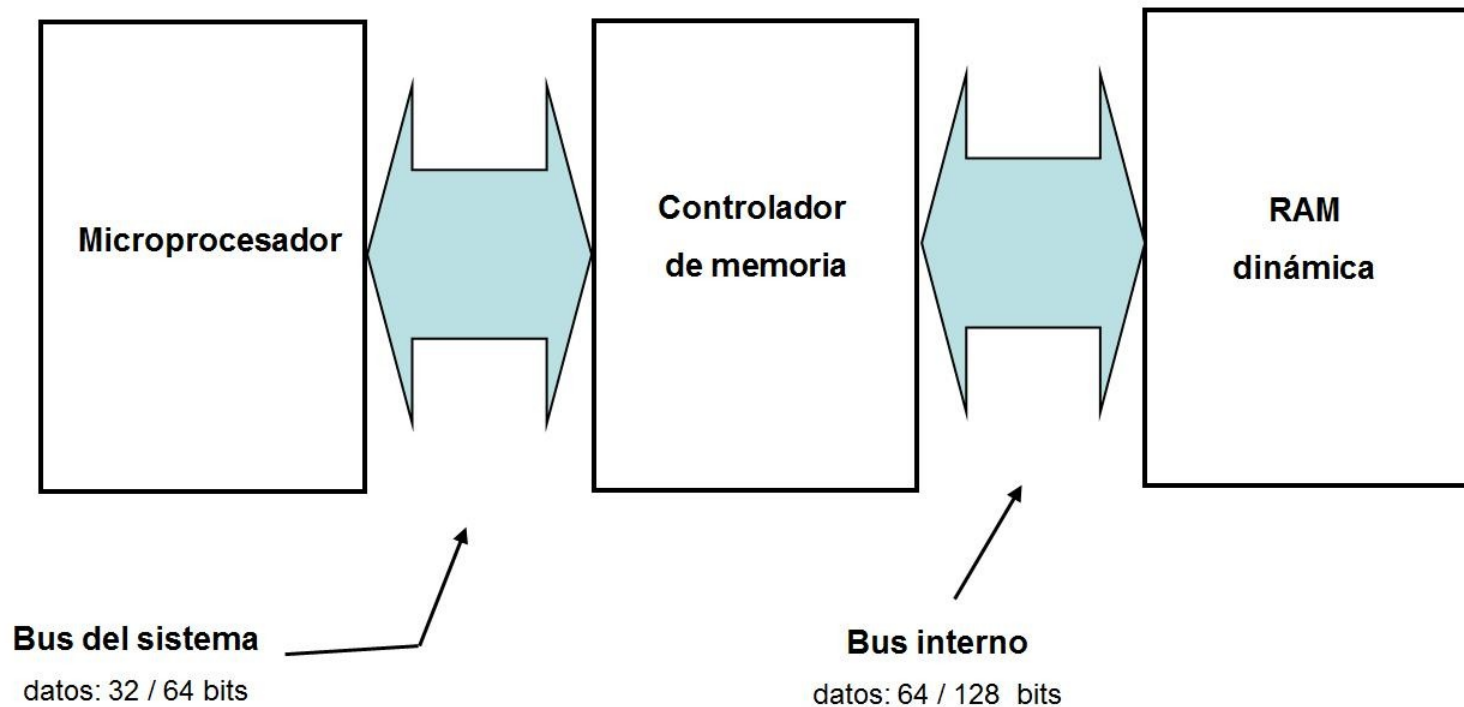
# Criterios para reducir el tiempo de acceso

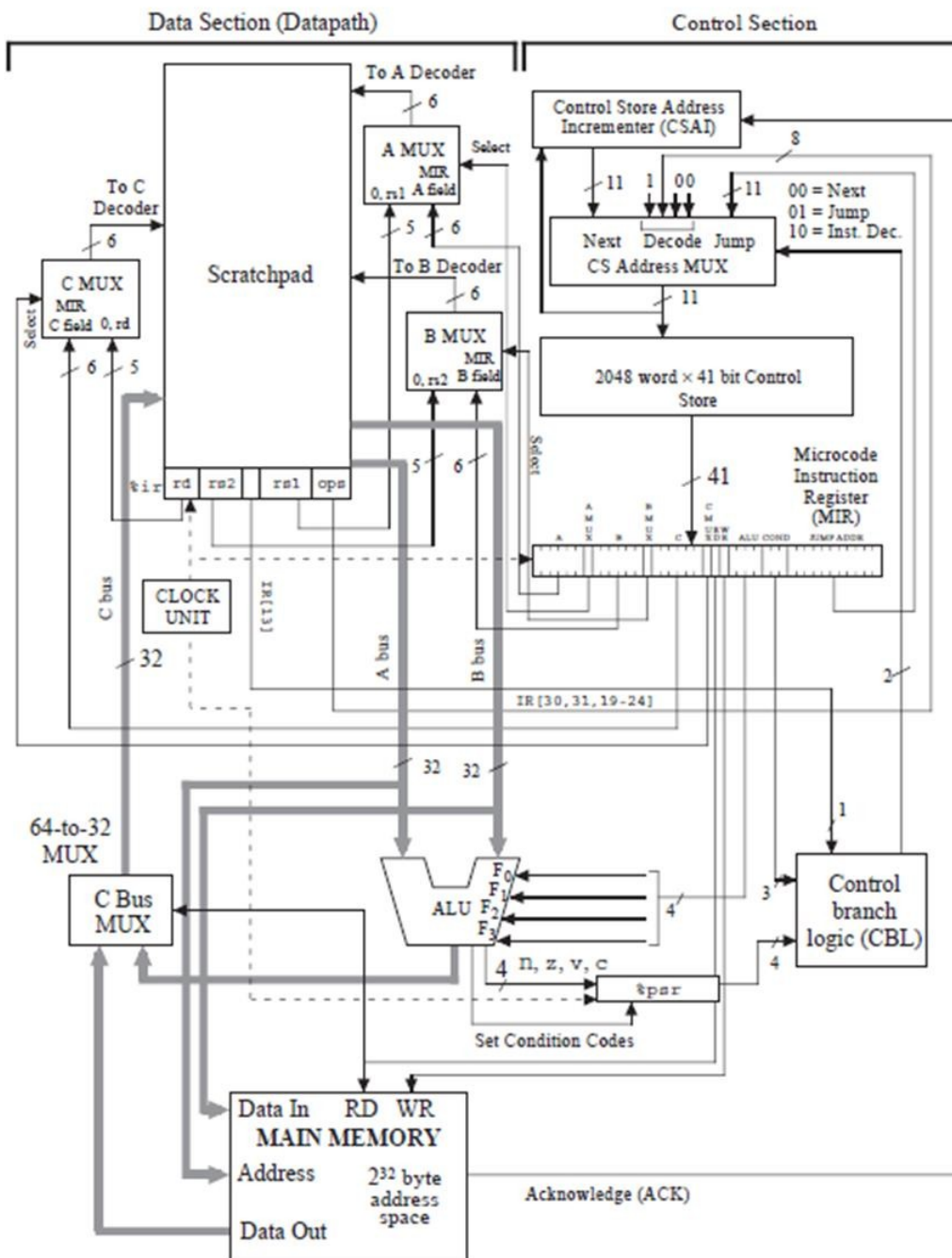
- ✓ Bancos entrelazados
  - Accede a un banco mientras en el otro se refresca la información
  - Enmascara el tiempo de refresco
  - Mejora rendimiento si pos. sucesivas están en bancos diferentes
- ✓ Mayor cantidad de bits leídos en paralelo (*ancho del bus*)
- ✓ Mayor velocidad del clock (*memoria sincrónica SDRAM*)
  - Mayor consumo de potencia
  - Proclive a errores de almacenamiento de bits
- ✓ Con igual velocidad de clock leer en flanco ascendente y también en el descendente (DDR)

---

La RAM dinámica es relativamente similar en todas las tecnologías, las principales diferencias están en su conexionado, direccionamiento y mejoras con circuitos adicionales on-chip

# Revisitando la conexión memoria-procesador

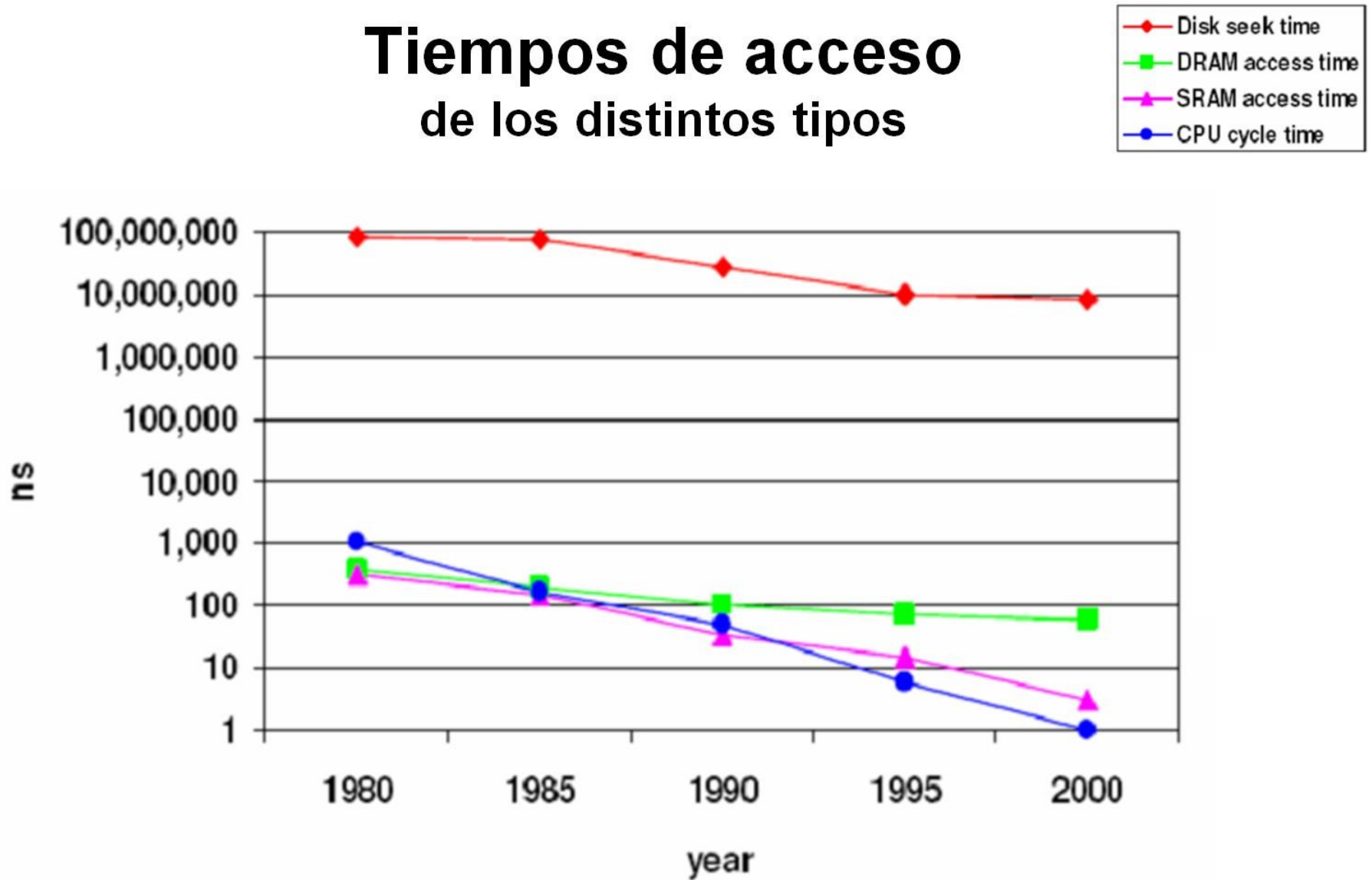




Revisitando la  
conexión  
memoria-procesador

*¿Qué debería modificarse?*

## Tiempos de acceso de los distintos tipos



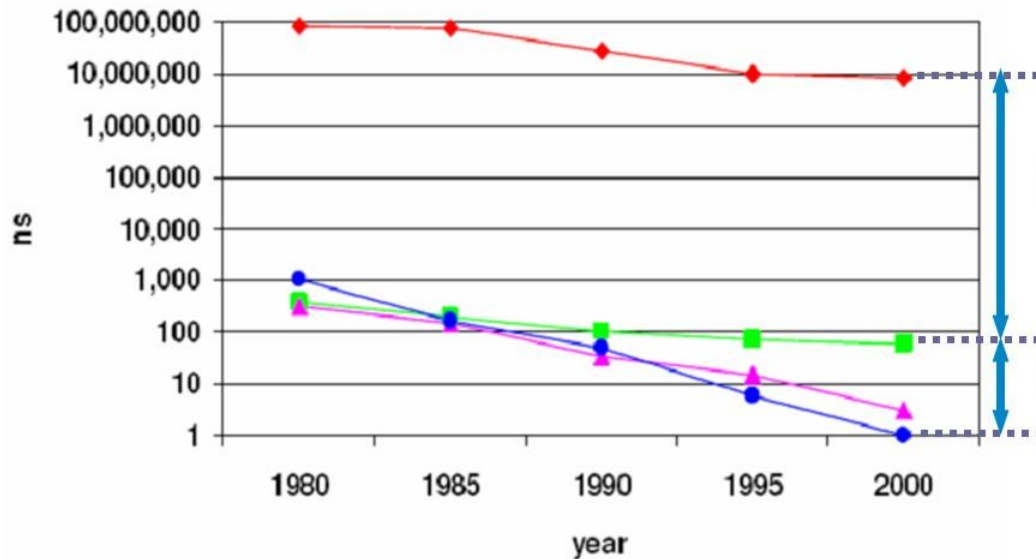
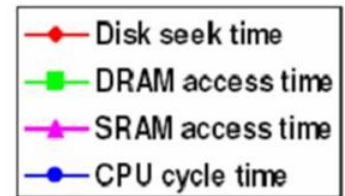
## Capacidad vs. Velocidad

- **Menor** tiempo de acceso => **mayor** costo por bit
- **Mayor** capacidad => **menor** costo por bit
- **Mayor** capacidad => **mayor** tiempo de acceso

una computadora ...

...requiere de mucha memoria y de memoria muy rápida

# Tiempos de acceso



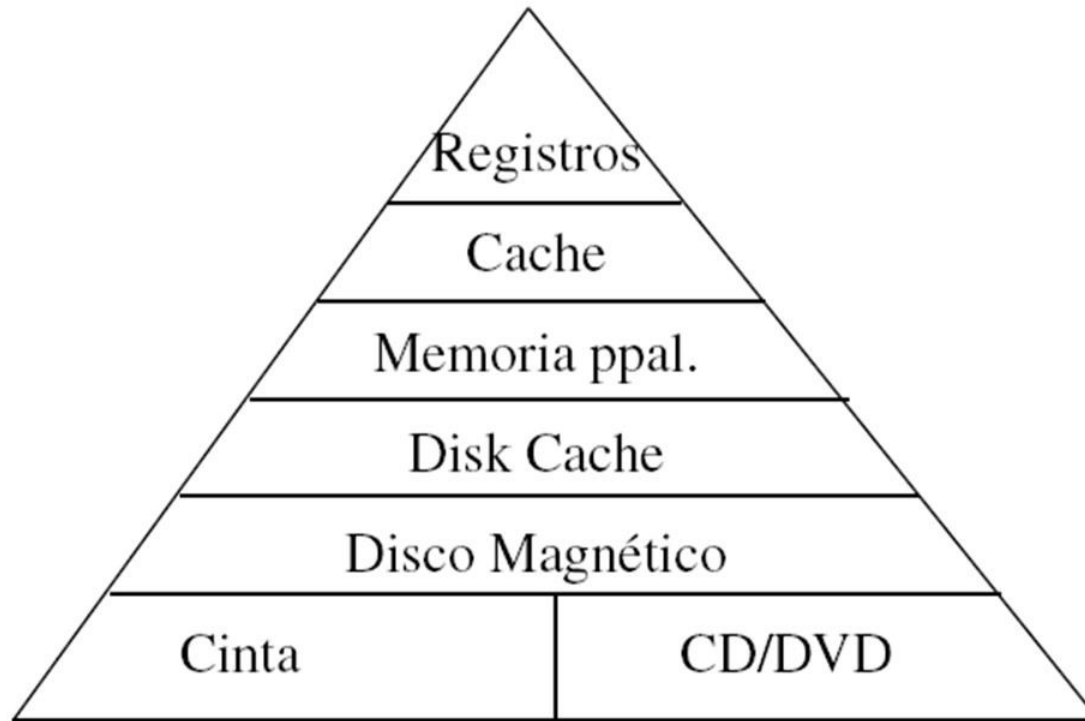
El S.O. busca compensar esta gran diferencia y usa el HD para aumentar la memoria disponible

~~El procesador debe esperar a la RAM~~

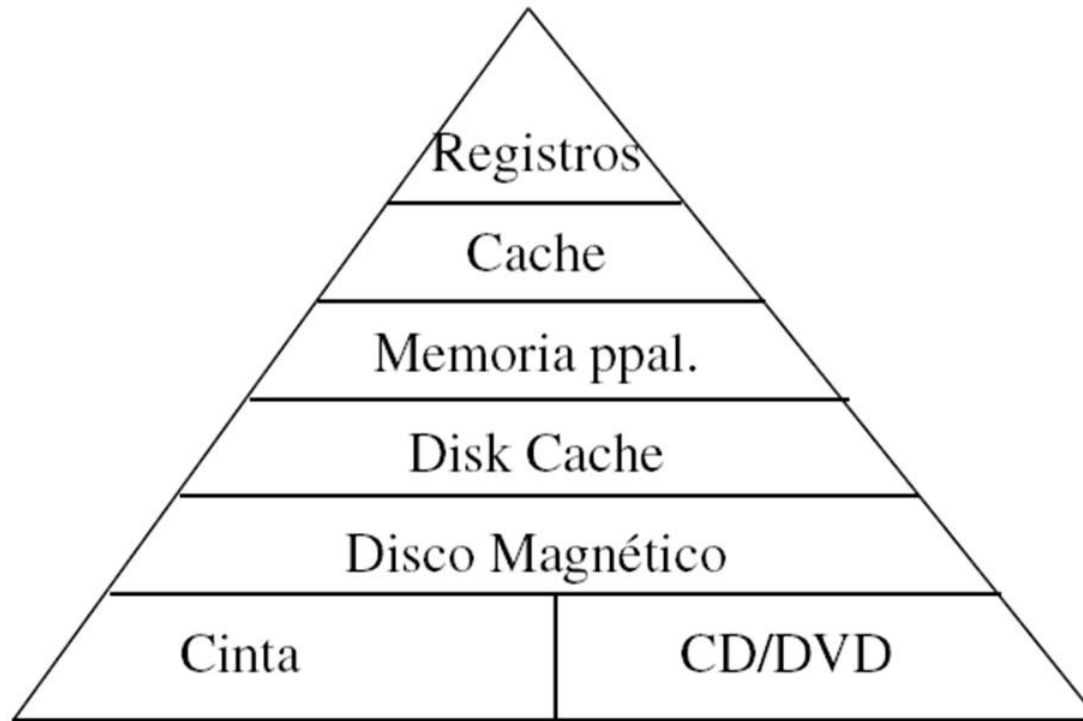
Organizar el funcionamiento de los distintos tipos de memoria (SRAM, DRAM, HD)

Rendimiento del sistema: como si tuviera mucha memoria y sólo memoria rápida

# Organización en jerarquías



# Organización en jerarquías



Memory type	Access time	Cost/MB	Typical amount used	Typical cost
Registers	0.5 ns	High	2 KB	–
Cache	5–20 ns	\$80	2 MB	\$160
Main memory	40–80ns	\$0.40	512 MB	\$205
Disk memory	5 ms	\$0.005	40 GB	\$200



---

# El porqué de la Memoria Cache

90% del tiempo de ejecución corresponde al 10% del código

## **“Principio de localidad”**

- Localidad temporal

Si accedo a una dirección, en poco tiempo volveré a accederla

- Localidad espacial

Si accedo a una dirección, las direcciones cercanas tienen mayor probabilidad de ser accedidas

---

# El porqué de la Memoria Cache

90% del tiempo de ejecución corresponde al 10% del código

## “Principio de localidad”

- Localidad temporal

Si accedo a una dirección, en poco tiempo volveré a accederla

- Localidad espacial

Si accedo a una dirección, las direcciones cercanas tienen mayor probabilidad de ser accedidas

Datos almacenados en posiciones contiguas

Iteraciones, procedimientos recursivos

# Memoria Cache

**Evita el cuello de botella producido por la marcada diferencia entre la velocidad del CPU y la velocidad de memoria principal**

**Físicamente el cache es:**

- Memoria muy rápida
- de poca capacidad
- “Cercana” al CPU

**Cómo funciona:**

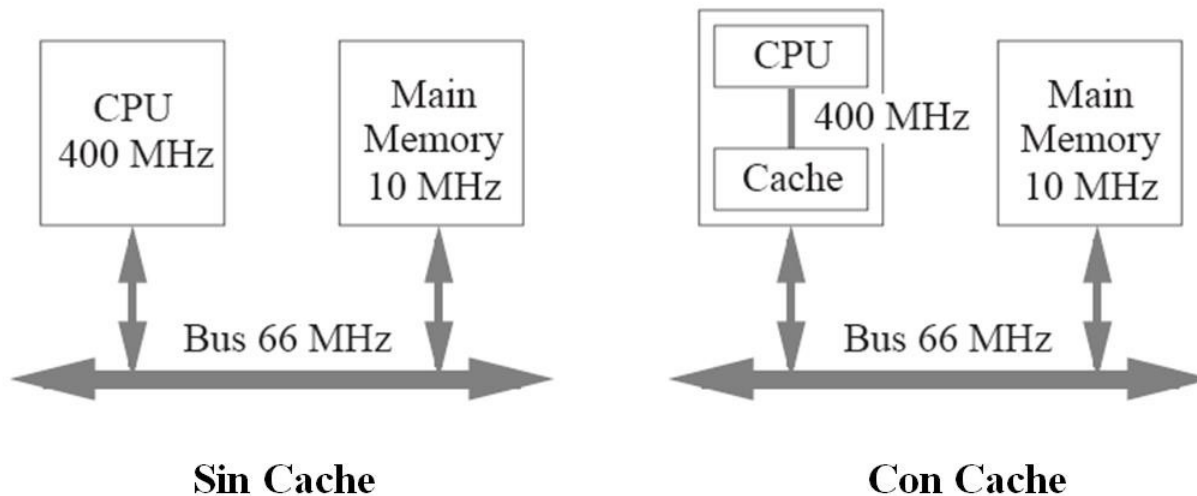
- Memoria dividida en bloques
- Al acceder un dato de mem. principal: bajo bloque completo al cache
- En el próximo acceso verifico si la posición buscada esta en cache,  
si no cargo otro bloque

**Agrega pasos al proceso de lectura/escritura pero:**

- *Cache + Principio de localidad => Aumenta el rendimiento*

# Memoria Cache

- ✓ **Visibilidad desde el punto de vista del programador**
- ✓ **Porqué la mem. cache es más rápida que la mem. princ.?**
  - Construida con electrónica más rápida (SRAM)
    - es más cara, ocupa más espacio y disipa más potencia
    - pero es escasa
  - Por ser escasa su árbol de decodificación es pequeño
  - Su cercanía al CPU es física y lógica, **no** se comunica por un bus compartido



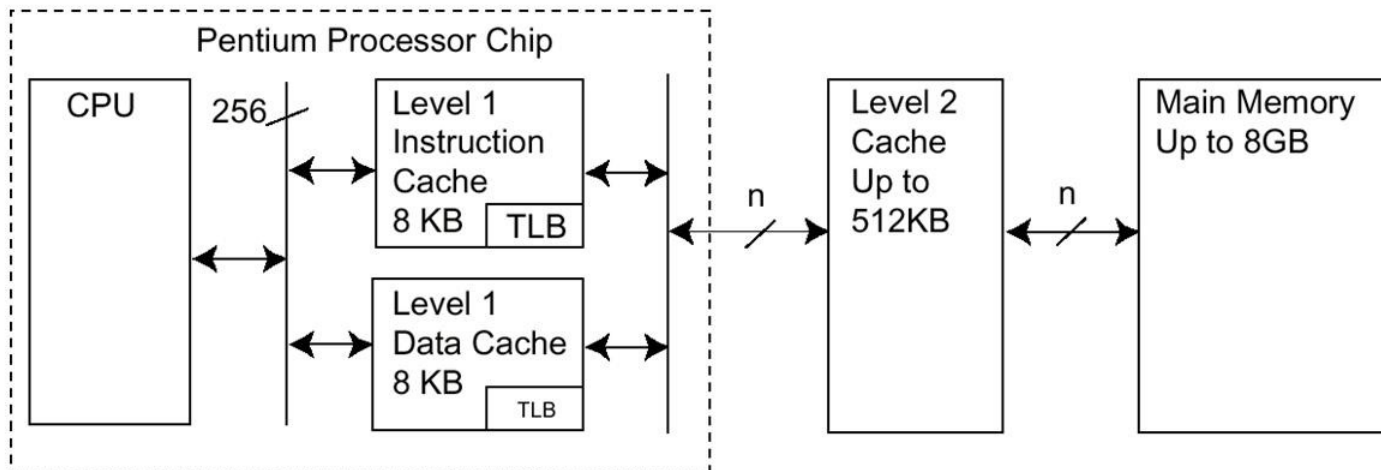
# Estructuras del Cache

## ✓ Cache especializado

- Cache de datos
- Cache de instrucciones

## ✓ Cache multinivel

- Cache más grandes son más **lentos**
  - Cache grandes, mayor **índ. de aciertos**
- => varios niveles



# Historia del cache en procesadores Intel

- **80386** – sin caché on-chip.
- **80486** – 8kB de caché unificada. Bloques de 16 bytes
- **Pentium** – 2 cachés on-chip. 8kB para datos y 8kB para instrucciones.
- **Pentium II** – caché L2
- **Pentium 4**
  - Caché L1: 8kBytes (4k+4k)
  - Caché L2 256kB
  - Caché L3
- **Core 2 duo**
  - Cache L1 (32k + 32 k) L2 6 Mb
- **Core i7**
  - Cache L1 (32k + 32 k) L2 256k L3 8Mb

