

Arquitecturas y Organización de Computadoras I

5: Memoria

Rafael Ignacio Zurita

Depto. Ingeniería de Computadoras

November 3, 2020

» Jerarquía de Memoria

Temario

- * The Memory Wall
- * Memorias
- * Estrategias de organización posibles
- * Principio de localidad
- * Jerarquía de memoria
- * DRAM vs SRAM
- * Repaso y cierre

» Jerarquía de Memoria

Cuando se quiere comprar una computadora nueva...

Si ya contamos con una PC CPU core i3 2.6Ghz, 8GB RAM, 512GB disco, ¿Qué computadora queremos adquirir?:



Procesador:

Memoria:

Disco:

» Jerarquía de Memoria

Cuando se quiere comprar una computadora nueva...

Si ya contamos con una PC CPU core i3 2.6Ghz, 8GB RAM, 512GB disco, ¿Qué computadora queremos adquirir?:



Procesador: más rápido, mas cores

Memoria:

Disco:

» Jerarquía de Memoria

Cuando se quiere comprar una computadora nueva...

Si ya contamos con una PC CPU core i3 2.6Ghz, 8GB RAM, 512GB disco, ¿Qué computadora queremos adquirir?:



Procesador: más rápido, mas cores

Memoria: más memoria

Disco:

» Jerarquía de Memoria

Cuando se quiere comprar una computadora nueva...

Si ya contamos con una PC CPU core i3 2.6Ghz, 8GB RAM, 512GB disco, ¿Qué computadora queremos adquirir?:



Procesador: más rápido, mas cores

Memoria: más memoria

Disco:

» Jerarquía de Memoria

Cuando se quiere comprar una computadora nueva...

Si ya contamos con una PC CPU core i3 2.6Ghz, 8GB RAM, 512GB disco, ¿Qué computadora queremos adquirir?:



Procesador: más rápido, mas cores

Memoria: más memoria

Disco: más disco

» Jerarquía de Memoria

Cuando se quiere comprar una computadora nueva...

Si ya contamos con una PC CPU core i3 2.6Ghz, 8GB RAM, 512GB disco, ¿Qué computadora queremos adquirir?:



GAMING
SERIES 9 PLUS

Envío gratis

OVERCLOCK

INTEL CORE I7 8700K
HEXA CORE - 3.2GHZ

NVIDIA GTX 1080TI 11GB
16GB RAM DDR4 3200 MHZ
1TB HDD | 240GB SSD

2 AÑOS DE GARANTIA
MONTAJE Y TESTEO INCLUIDO

Procesador: más rápido, mas cores

Memoria: más memoria

Disco: más disco

» Jerarquía de Memoria

"Procesador más rápido, mas cores"

- * **IMPORTANTE:** es tiempo de utilizar mayor precisión y terminología
- * Ejemplo: "Microprocesador con un rendimiento mayor que el anterior"
 - * (Es más adecuado)
 - * Buscamos que realice una mayor cantidad de trabajo en el mismo tiempo o,
 - * la misma cantidad de trabajo que antes, pero en un tiempo menor.

» Jerarquía de Memoria

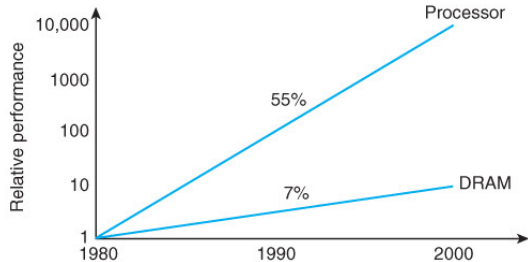
The Memory Wall

Año tras año ...

- * Los microprocesadores nuevos son más rápidos (mejoran el rendimiento).
- * Las memorias nuevas son más rápidas (mejoran el rendimiento).

FIGURE 9.4

Trends in memory performance



© Cengage Learning 2014

» Jerarquía de Memoria

Memorias

- * En una computadora de propósito general se cuenta con:
 - * Memoria con Firmware de arranque
 - * Memoria RAM (volatil, estática y dinámica/DRAM y caché)
 - * Disco rígido o memoria flash
 - * Disco de estado sólido
 - * DVD y CD
 - * Memorias SD y pendrives

¿Por qué necesitamos tantos tipos de memoria?

» Jerarquía de Memoria

Memorias

- * En una computadora de propósito general se cuenta con:
 - * Memoria con Firmware de arranque
 - * Memoria RAM (volatil, estática y dinámica/DRAM y caché)
 - * Disco rígido o memoria flash
 - * Disco de estado sólido
 - * DVD y CD
 - * Memorias SD y pendrives

¿Por qué necesitamos tantos tipos de memoria?

- * Memoria ideal: ilimitada (siempre suficiente), veloz, densa (tamaño físico), de bajo consumo, robusta, barata, persistente. Pero, la tecnología es insuficiente.

» Jerarquía de Memoria

Memorias

¿Por qué necesitamos tantos tipos de memoria?

- * Memoria ideal: ilimitada (siempre suficiente), veloz, densa (tamaño físico), de bajo consumo, robusta, barata, persistente. Pero, la tecnología es insuficiente.

	Capacity	Latency	Cost/GB
Register	1000s of bits	20 ps	\$\$\$\$
SRAM	~10 KB-10 MB	1-10 ns	~\$1000
DRAM	~10 GB	80 ns	~\$10
Flash	~100 GB	100 us	~\$1
Hard disk	~1 TB	10 ms	~\$0.10

» Jerarquía de Memoria

Memorias

¿Por qué necesitamos tantos tipos de memoria?

- * Memoria ideal: ilimitada (siempre suficiente), veloz, densa (tamaño físico), de bajo consumo, robusta, barata, persistente. Pero, la tecnología es insuficiente.

	Capacity	Latency	Cost/GB
Register	1000s of bits	20 ps	\$\$\$\$
SRAM	~10 KB-10 MB	1-10 ns	~\$1000
DRAM	~10 GB	80 ns	~\$10
Flash	~100 GB	100 us	~\$1
Hard disk	~1 TB	10 ms	~\$0.10

- * **IDEA 1:** Exponer todos los tipos de memoria y que el programador la utilice con el mejor rendimiento posible.

» Jerarquía de Memoria

Localidad de las Referencias

- * **IDEA 2:** mantener los datos más frecuentemente utilizados por el procesador en una pequeña memoria veloz SRAM (cercana a la CPU) de manera transparente.
- * Hacer referencia a la memoria principal únicamente de vez en cuando.
 - * Esta idea libera al programador
 - * ¿Qué hace posible esta idea?

» Jerarquía de Memoria

Localidad de las Referencias

- * **IDEA 2:** mantener los datos más frecuentemente utilizados por el procesador en una pequeña pero veloz memoria SRAM (cercana a la CPU) de manera transparente.
- * Hacer referencia a la memoria principal unicamente de vez en cuando.
 - * Esta idea libera al programador
 - * ¿Qué hace posible esta idea?
- * **Comportamiento de los programas:**
 - * En un intervalo corto de tiempo se utiliza una pequeña fracción del total de la memoria.
 - * Este comportamiento de acceso ha sido llamado principio de localidad de las referencias[DENN68], o simplemente **principio de localidad**.

» Jerarquía de Memoria

Localidad de las Referencias

```
while (save[i]==k)
    i = i + j;
```

i: \$s3, j: \$s4, k: \$s5, base of save[]: \$s6

```
Loop:  add $t1,$s3,$s3
        add $t1,$t1,$t1
        add $t1,$t1,$s6
        lw  $t1,0($t1)
        bne $t1,$s5,Exit

        add $s3,$s3,$s4
        j   Loop

Exit:
```

Repaso de IC: Ciclo de instrucción

El procesador ejecuta las mismas instrucciones varias veces,

y también accede a elementos del vector `save[]`.

» Jerarquía de Memoria

Localidad de las Referencias

```
while (save[i]==k)
    i = i + j;
```

i: \$s3, j: \$s4, k: \$s5, base of save[]: \$s6

```
Loop:  add $t1,$s3,$s3
        add $t1,$t1,$t1
        add $t1,$t1,$s6
        lw  $t1,0($t1)
        bne $t1,$s5,Exit

        add $s3,$s3,$s4
        j   Loop

Exit:
```

Repaso de IC: Ciclo de instrucción

El procesador ejecuta las mismas instrucciones varias veces (**localidad temporal**),

y también accede a elementos del vector `save[]` (**localidad espacial**).

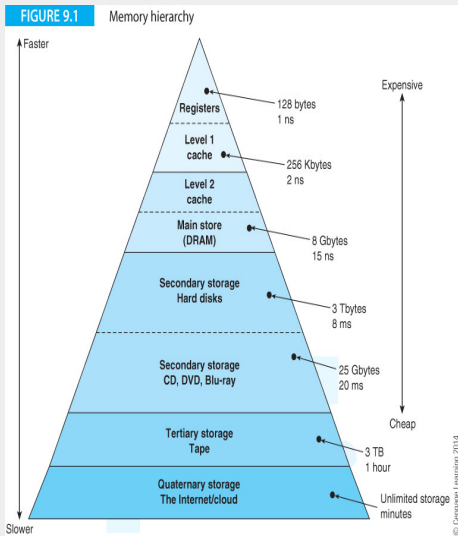
» Jerarquía de Memoria

Resumen

- * **Jerarquía de Memoria:** Organización de la memoria de una computadora que permita
 - * obtener el **rendimiento** de una memoria de alta velocidad
 - * al **costo** (precio) de una memoria grande y lenta.
- * Idea clave: tener los datos correctos, en el lugar preciso, en el momento adecuado (principio de localidad)

» Jerarquía de Memoria

Jerarquía de Memoria



Organizar el sistema de memoria dentro de una **jerarquía de niveles**.

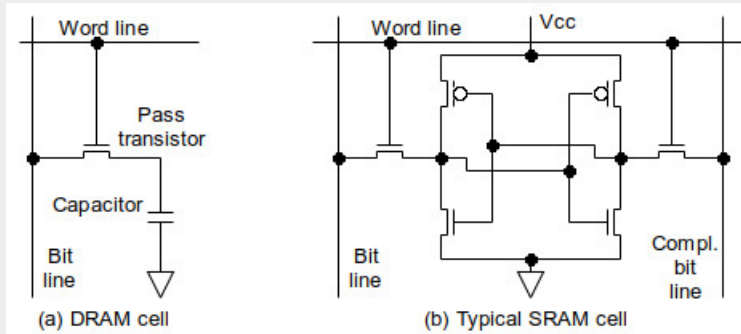
Cada nivel está compuesto por un tipo de memoria (tecnología)

Objetivo: *obtener el rendimiento de una memoria de gran velocidad al coste de una memoria de baja velocidad, y de tamaño casi ilimitado.*

» Jerarquía de Memoria

DRAM vs SRAM

- * Factores: densidad, costos, consumo, rendimiento.
- * ¿Densidad en cachés?



DRAM vs SRAM

» Jerarquía de Memoria

Repaso

Organizando las tecnologías de memoria dentro de una **jerarquía de niveles** se puede construir un sistema de memoria de

- * bajo costo y
- * con un rendimiento similar al de una memoria cara y de alta velocidad

Si este sistema es transparente al programador

- * ¿Para qué interesa su estudio?

» Jerarquía de Memoria

Verifique su comprensión del tema

1. ¿Qué significa el principio de localidad?.
2. Intente componer y redactar una analogía no técnica.
3. Evaluar el tiempo de ejecución de los dos programas a continuación.
4. Debatir en grupo por qué es posible que existe una diferencia.
5. En base a la respuesta anterior: ¿Considera que es importante el estudio de Jerarquía de Memoria?.

» Jerarquía de Memoria

Verifique su comprensión del tema

Recorre una matriz- version 1

```
main () {  
    int i, j, k;  
    int n[10000][100]; /* definicion de una matriz */  
  
    for (k=0; k<100; k++) {  
  
        for (i=0; i<10000; i++)  
        for (j=0; j<100; j++)  
            n[i][j] +=3;  
  
        for (i=0; i<10000; i++)  
        for (j=0; j<100; j++)  
            n[i][j] +=5;  
    }  
}
```

Recorre una matriz- version 2|

```
main () {  
    int i, j, k;  
    int n[10000][100]; /* definicion de una matriz */  
  
    for (k=0; k<100; k++) {  
  
        for (j=0; j<100; j++)  
        for (i=0; i<10000; i++)  
            n[i][j] +=3;  
  
        for (j=0; j<100; j++)  
        for (i=0; i<10000; i++)  
            n[i][j] +=5;  
    }  
}
```


» Jerarquía de Memoria

- * ¿Preguntas?
- * Próximas clases: Caché y Memoria Virtual

» Bibliografía

Material complementario de estudio

Apunte de cátedra

- * **Memoria**, Rafael Ignacio Zurita 2017 (disponible en PEDCO). Versión en español ampliada (con permiso escrito de Prof. Alan Clements y Prof. Hank Levy) de los libros:
 - * Computer Organization and Architecture: Themes and Variations, Alan Clements, Cengage Learning, 2013, ISBN: 1285415426, 9781285415420
 - * Computer Programming and Architecture the VAX-11, Henry Levy, Digital Press 1980

Libros

- * Andrew S. Tanenbaum (2000), ORGANIZACIÓN DE COMPUTADORAS un enfoque estructurado, Editorial Prentice Hall. (10 copias en biblioteca)
- * David. Patterson John L. Hennessy (1995), ORGANIZACIÓN Y DISEÑO DE COMPUTADORES La interfaz hardware/software, McGraw-Hill (8 copias en biblioteca).