

Tema 1

Tecnología informática y niveles de abstracción

La Revolución de los Computadores

- **Ha habido un espectacular avance en la tecnología de los computadores**
 - Sustentado por la ley de Moore
- **Ha hecho factibles nuevas aplicaciones**
 - Computadores en los automóviles
 - Teléfonos móviles
 - Proyecto genoma humano
 - World Wide Web
 - Buscadores
- **Los computadores se utilizan en todo**

Tipos de Computadores

- **Computadores de sobremesa**

- Propósito general, gran variedad de software
- Sometidos al compromiso entre coste y prestaciones

- **Servidores**

- Se apoyan sobre la red
- Altos rendimiento, capacidad y seguridad
- Desde muy pequeños hasta del tamaño de edificios

- **Computadores empotrados**

- Ocultos como componentes de sistemas
- Estrictas restricciones potencia/rendimiento/coste

Unidades de medida de la información

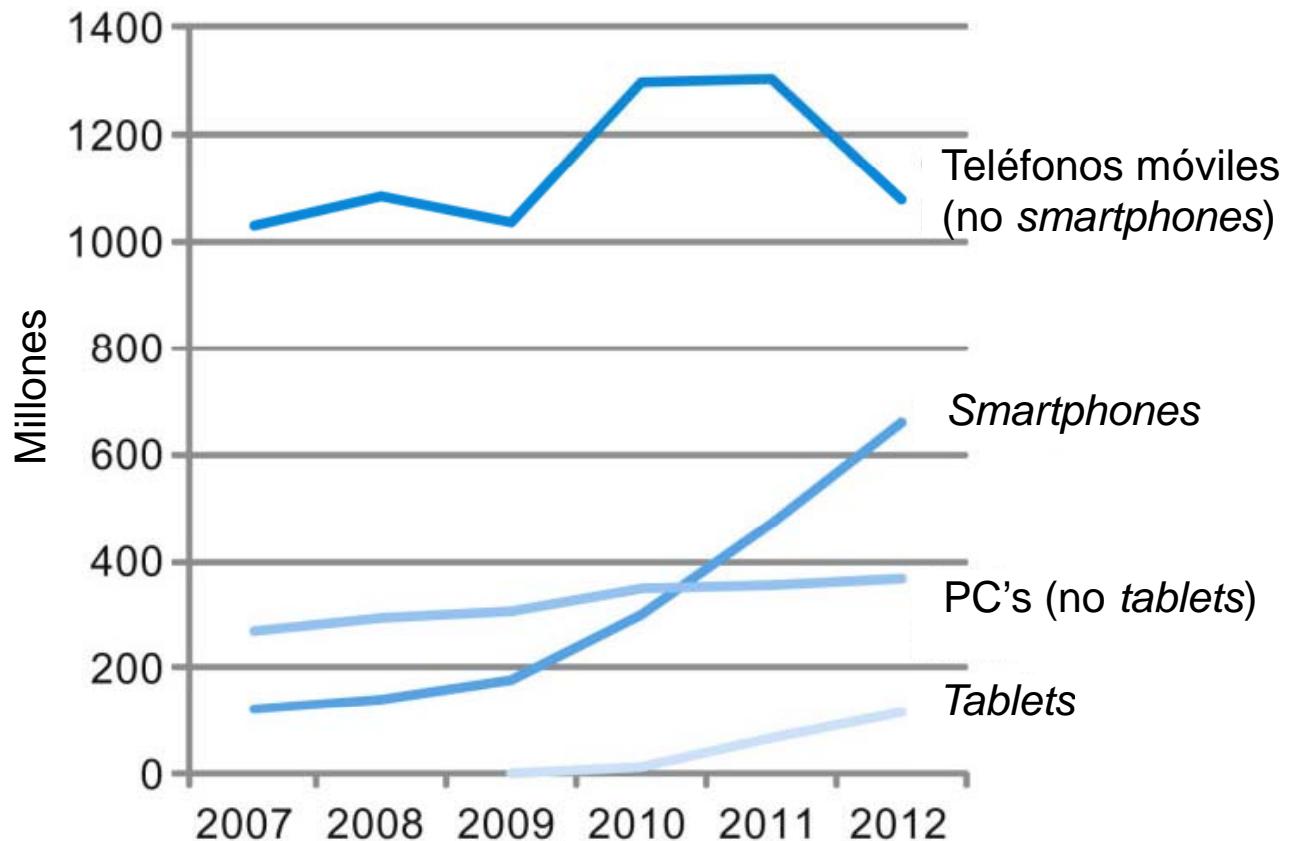
- **La unidad elemental de información es el bit (dos estados)**
 - 1 byte = 8 bits.
- **Tradicionalmente los múltiplos del byte se medían en potencias de 2.**
 - Kilobyte (KB)= 2^{10} bytes, Megabyte (MB)= 2^{20} bytes , etc.
 - Se causaba confusión por comparación con el resto de magnitudes del Sistema Internacional de Unidades que se basan en potencias de 10.
 - Se trató de resolver la confusión en 1998 definiendo el KB, MB, GB, etc. mediante potencias de 10, y con nuevos prefijos para las potencias de 2.

Múltiplos del byte

Notación decimal	Abreviatura	Valor	Notación binaria	Abreviatura	Valor	Diferencia
Kilobyte	KB	10^3	Kibibyte	KiB	2^{10}	2 %
Megabyte	MB	10^6	Mebibyte	MiB	2^{20}	5 %
Gigabyte	GB	10^9	Gibibyte	GiB	2^{30}	7 %
Terabyte	TB	10^{12}	Tebibyte	TiB	2^{40}	10 %
Petabyte	PB	10^{15}	Pebibyte	PiB	2^{50}	13 %
Exabyte	EB	10^{18}	Exbibyte	EiB	2^{60}	15 %

- En la comunidad informática no está aún plenamente aceptada la notación decimal.
- Los fabricantes de discos y proveedores de red utilizan la notación decimal ya que salen beneficiados (p.e. $10^9 < 2^{30}$)

El Mercado de Procesadores



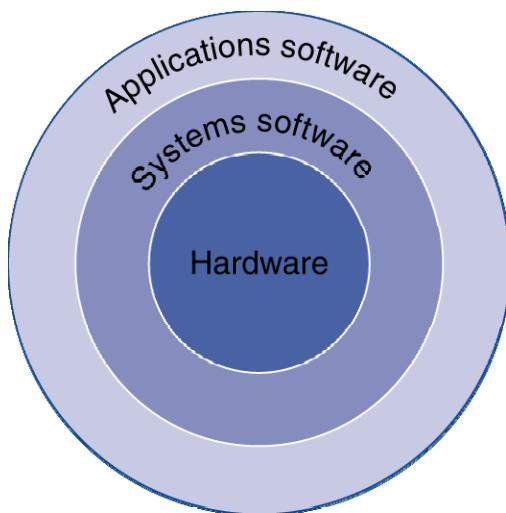
¿Qué vamos a estudiar?

- **Cómo se traducen los programas al lenguaje máquina**
 - Y cómo los ejecuta después el hardware
- **La interfaz entre hardware y software**
- **Qué determina el rendimiento de un programa**
 - Y cómo puede mejorarse
- **Cómo mejoran el rendimiento los diseñadores del hardware**
- **Qué es el procesamiento paralelo**

Comprendamos el rendimiento

- **El algoritmo**
 - Determina el número de operaciones ejecutado
- **El lenguaje de programación, el compilador y la arquitectura**
 - Determinan el número de instrucciones máquina ejecutadas por operación
- **El procesador y el sistema de memoria**
 - Determinan lo rápido que se ejecutan las instrucciones
- **El sistema de E/S (I/O), incluyendo el SO**
 - Determinan lo rápido que se ejecutan las operaciones de I/O

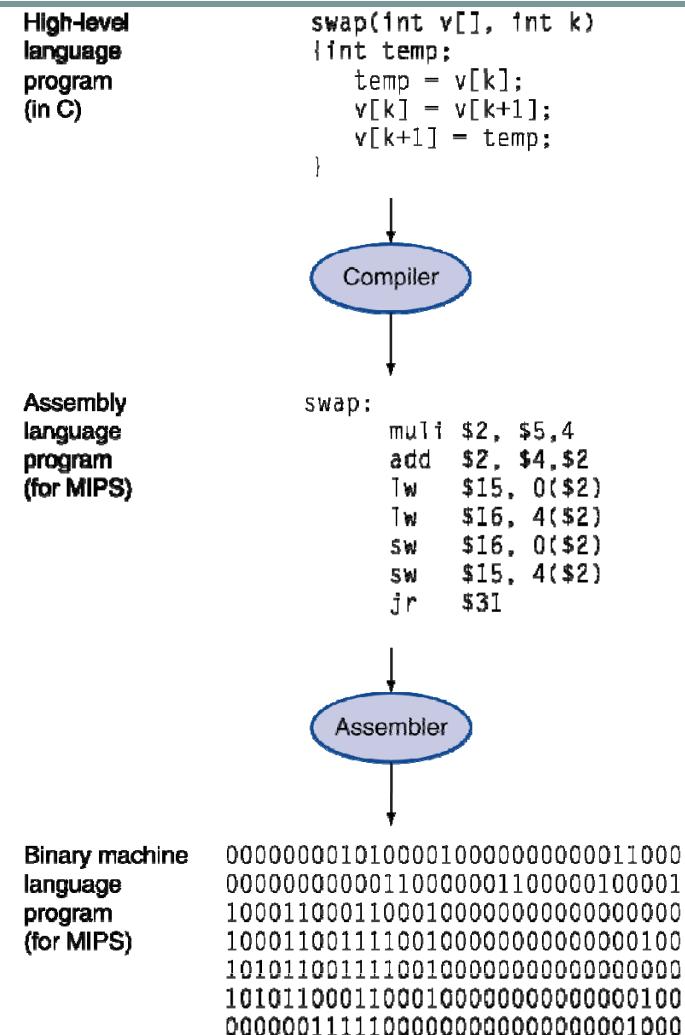
¿Qué hay por debajo de nuestro programa?



- **Software de la aplicación**
 - Escrito en un lenguaje de alto nivel (HLL, *high level language*)
- **Software de sistema**
 - Compilador: traduce el HLL a código máquina
 - Sistema operativo: código que da servicios
 - Gestionando la entrada/salida
 - Administrando la memoria, el espacio de almacenamiento y los recursos en general
 - Planificando las tareas y permitiendo la compartición de recursos
- **Hardware**
 - Procesador, memoria y controladores de I/O

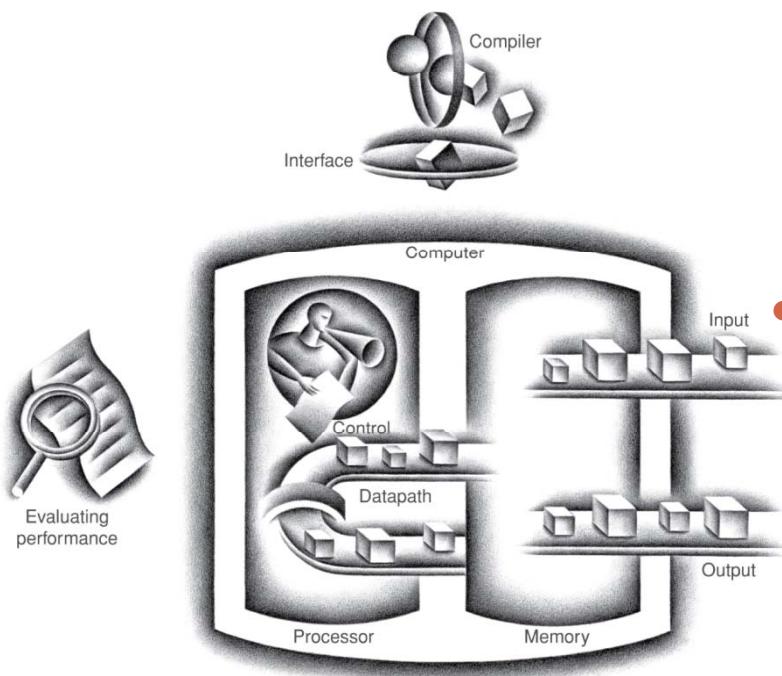
Niveles del código de los programas

- **Lenguaje de alto nivel**
 - Nivel de abstracción próximo al campo del problema
 - Pensado para ser portable y fácilmente usado
 - **Lenguaje ensamblador**
 - Representación textual de las instrucciones máquina
 - **Representación a nivel del hardware (código máquina)**
 - Dígitos binarios (bits)
 - Instrucciones y datos codificados



Los componentes de un computador

Recuadro importante



- **Los mismos componentes para todos los tipos de computadores**
 - Sobremesa, servidores, empotrados, ...
 - Control, memoria, ruta de datos, entrada/salida
- **La entrada/salida incluye:**
 - Dispositivos de interfaz de usuario
 - ✖ Pantalla, teclado, ratón
 - Dispositivos de almacenamiento
 - ✖ Disco duro, CD/DVD, flash
 - Adaptadores de red
 - ✖ Para la comunicación con otros computadores

Elementos de la memoria del computador

- **Bus de direcciones**

En él se deposita la dirección de memoria sobre la que queremos operar.

- **Células de memoria**

Son las que almacenan la información.

- **Bus de datos**

- **En lectura:**

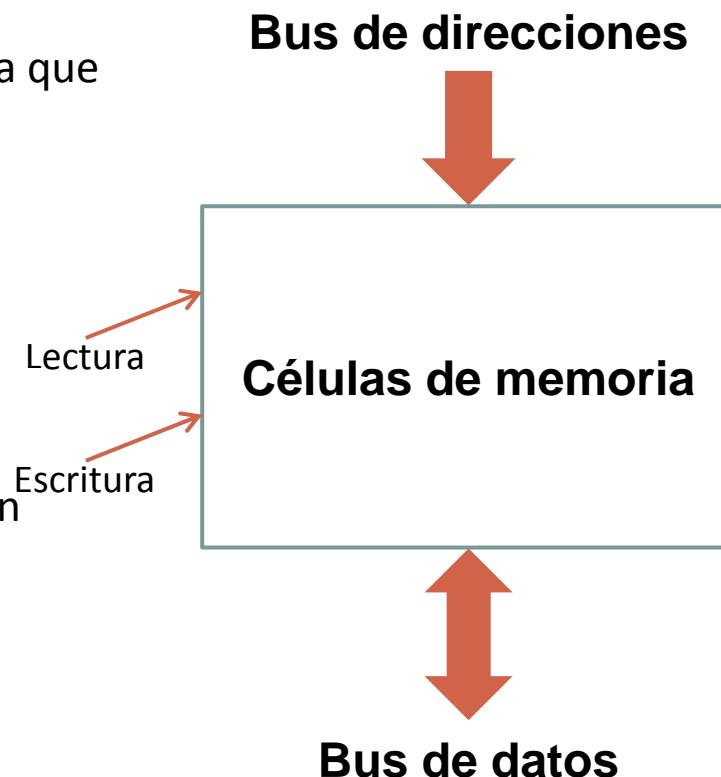
La memoria deposita en este bus la información contenida en la dirección solicitada.

- **En escritura:**

Depositamos en este bus la información que queremos escribir en la dirección solicitada.

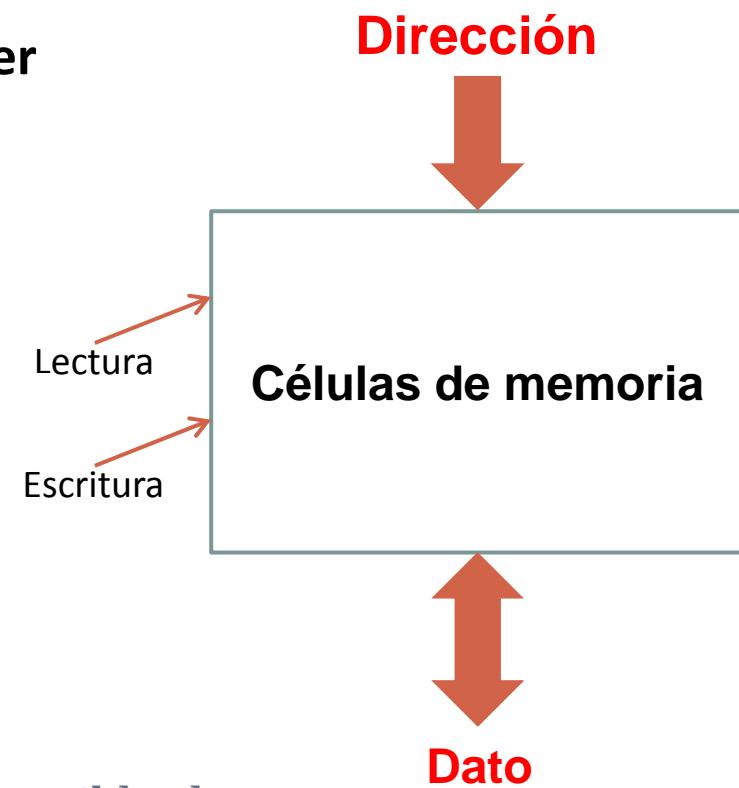
- **Líneas lógicas de control**

Lectura y escritura entre otras.



Funcionamiento de la memoria del computador (lectura)

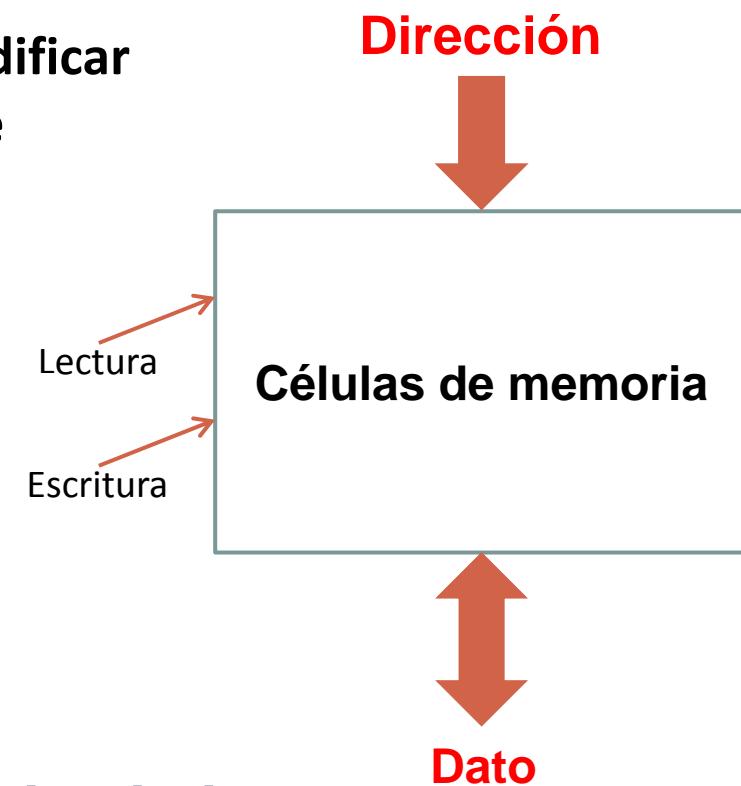
1. Se deposita en el bus de direcciones la dirección de memoria que queremos leer
2. Se activa la línea de lectura.
3. Los circuitos de la memoria comenzarán a trabajar para...
4. Situar el contenido de la dirección deseada en el bus de datos.



El tiempo que tarda la memoria en tener disponible el dato desde que se da la orden de lectura se denomina latencia de la memoria o tiempo de acceso (en lectura).

Funcionamiento de la memoria del computador (escritura)

1. Se deposita en el bus de direcciones la dirección de memoria que vamos a modificar y en el bus de datos la información que queremos escribir.
2. Se activa la línea de escritura.
3. En ese momento los circuitos de la memoria comenzarán a trabajar para...
4. Escribir en la dirección depositada en el bus de direcciones la información que está en el bus de datos.



El tiempo que tarda la memoria en escribir el dato desde que se da la orden de escritura se denomina latencia de la memoria o tiempo de acceso (en escritura).

Funcionamiento del computador

- **Programa máquina**

Un programa en lenguaje máquina es una secuencia de números en binario que representan las instrucciones para efectuar determinada tarea

- **Registros importantes:**

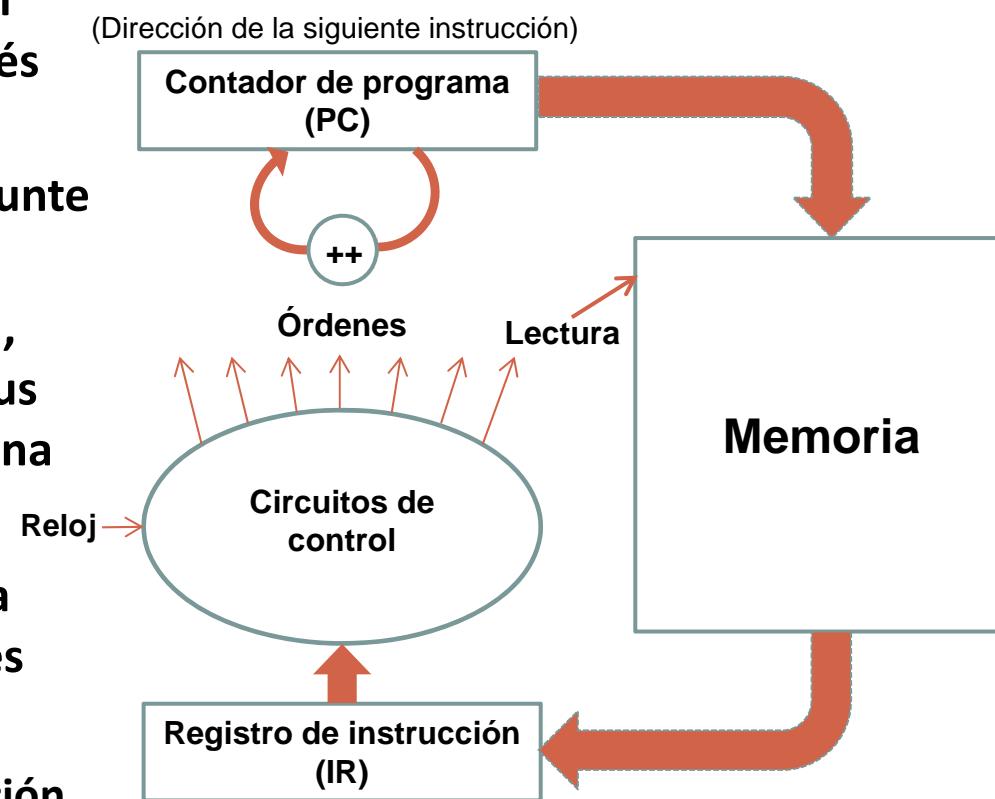
- Contador de programa (PC)
 - Contiene la dirección de la siguiente instrucción a ejecutar
- Registro de instrucción (IR)
 - Contiene la instrucción en curso

- **Proceso de ejecución de una instrucción**

- Fase de búsqueda
 - Localización de la instrucción y de sus operandos, si los tiene
- Fase de ejecución
 - Ejecución de la instrucción y escritura del resultado
 - Volver a la fase de búsqueda para ejecutar la instrucción siguiente

Esquema del proceso de ejecución de una instrucción del computador

1. Para leer la instrucción se pasa el PC al bus de direcciones y después se da la orden de lectura.
2. Se incrementa el PC para que apunte a la instrucción siguiente.
3. Como consecuencia de la lectura, tendremos la instrucción en el bus de datos desde donde se almacena en el registro de instrucción.
4. En el registro de instrucción, ésta se decodifica y se dan las órdenes oportunas para su ejecución.
5. Se repite el proceso con la instrucción siguiente.



Anatomía de un Computador



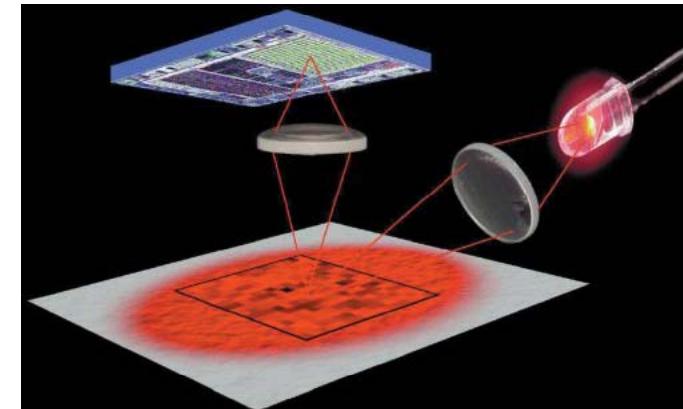
Anatomía de un Ratón

- **Ratón óptico**

- Un LED ilumina la superficie de apoyo
- Cámara de baja resolución
- Procesador de imágenes básico
 - Calcula el movimiento en *x* e *y*
- Botones y rueda

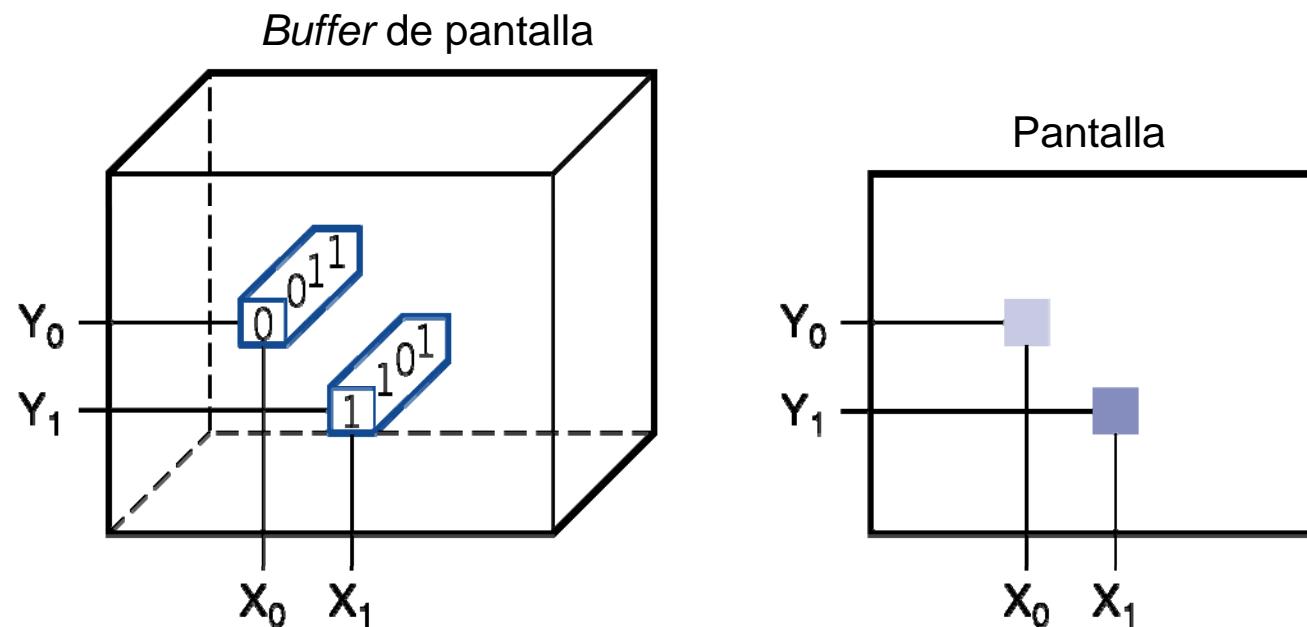
- **Dejó obsoleto al ratón mecánico de bola**

- **Ejemplo de aumento de calidad y disminución de coste**

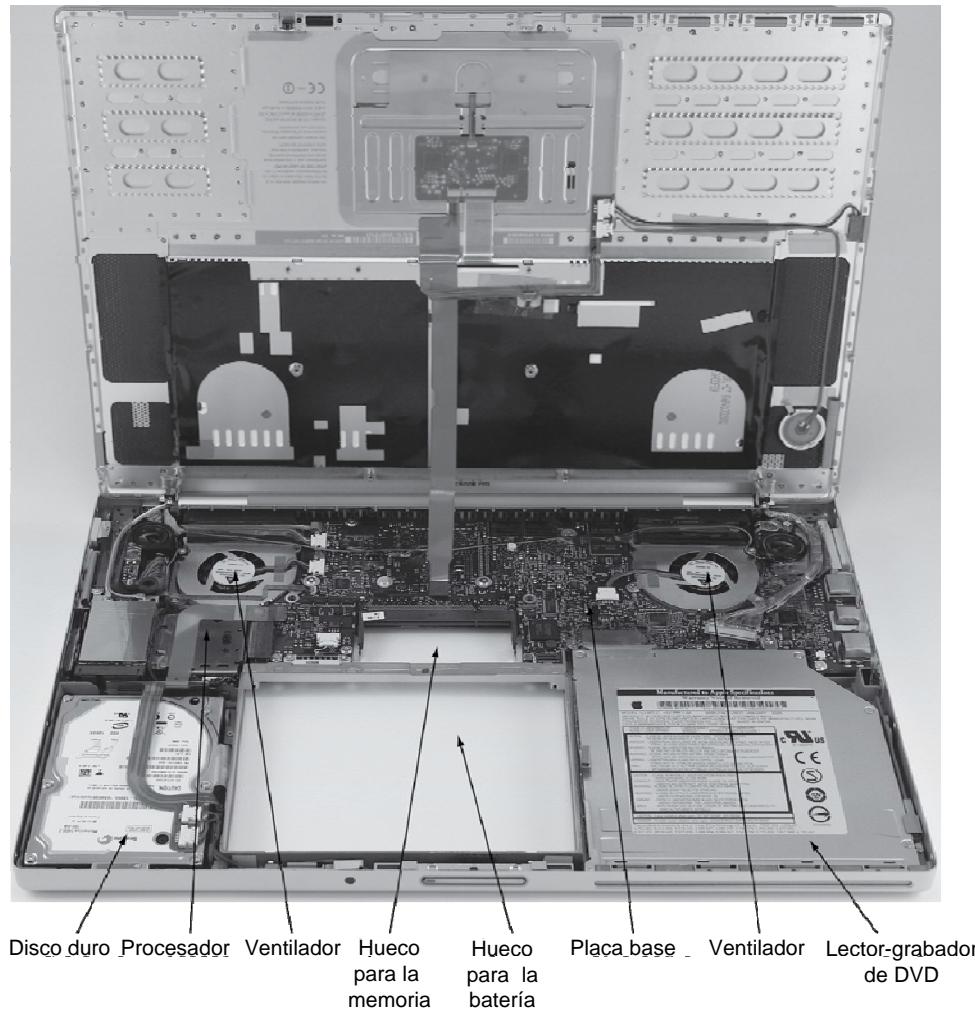


A través de la pantalla

- Pantalla LCD: los puntos que componen las imágenes (*pixels*)
 - Reflejan el contenido del almacenamiento de la imagen en la memoria (*buffer de pantalla o frame buffer*)



Abriendo la caja



Dentro del procesador (CPU)

- **Cauce o ruta de datos**
 - Ejecuta las operaciones sobre los datos.
- **Control**
 - Establece la secuencia para la ruta de datos, la memoria, ...
- **Memoria caché**
 - Memoria SRAM pequeña y rápida para tener acceso inmediato a los datos.

Dentro del procesador

- Apple A5 (tamaño original 10.1x12.1 mm)



Las abstracciones

Recuadro importante

- **La abstracción ayuda a tratar los sistemas complejos**
 - Ocultando los detalles de bajo nivel
- **Arquitectura del juego de instrucciones (*Instruction Set Architecture - ISA*)**
 - La interfaz entre el hardware y el software
- **La interfaz binaria de las aplicaciones**
 - La ISA más la interfaz de software del sistema
- **La implementación**
 - Los detalles de lo que está por debajo y su interfaz

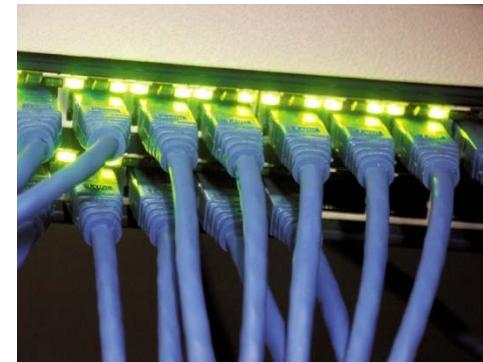
Un lugar seguro para los datos

- **Memoria principal volátil**
 - Pierde instrucciones y datos cuando se apaga
- **Memoria secundaria no volátil**
 - Disco magnético
 - Memoria *Flash*
 - Disco óptico (CDROM, DVD)



Redes

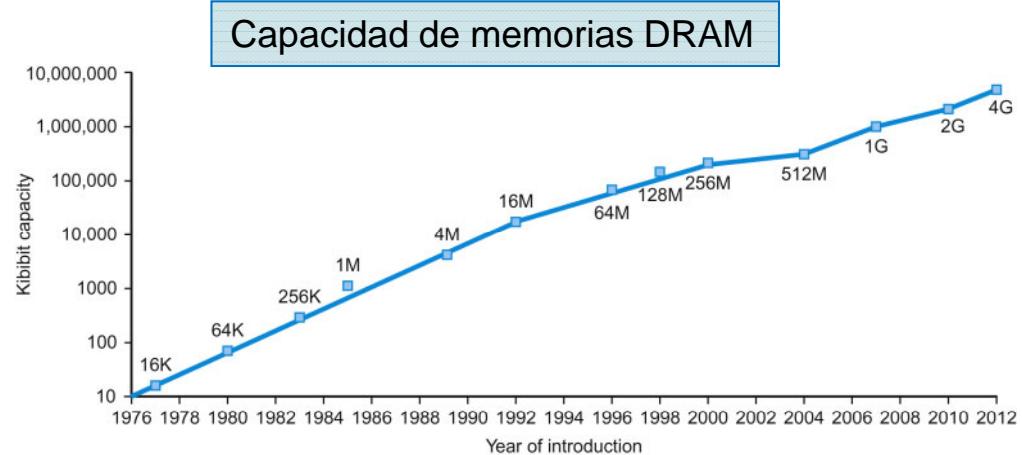
- Comunicación y compartición de recursos
- Red de área local (*Local Area Network - LAN*): Ethernet
 - Dentro de un edificio
- Red de área extensa (*Wide Area Network – WAN*): Internet
- Red inalámbrica (*Wireless*): WiFi, Bluetooth



Tendencias de la tecnología

- **La tecnología electrónica continúa avanzando**

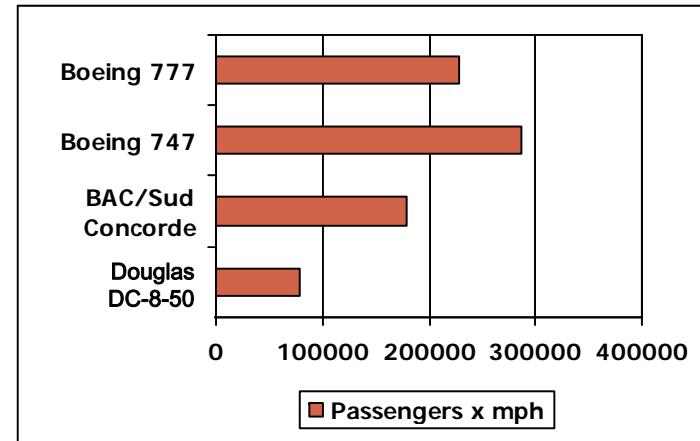
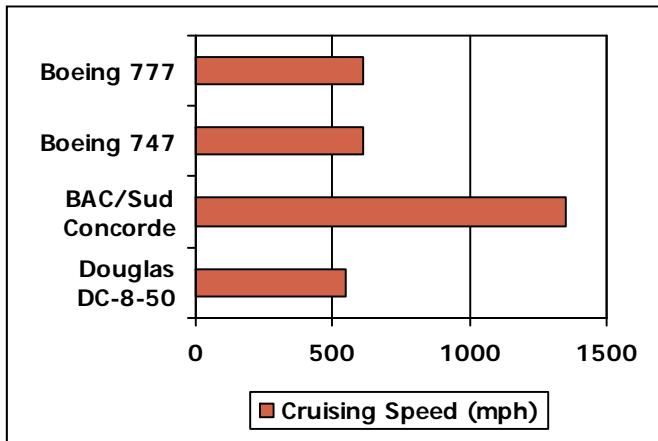
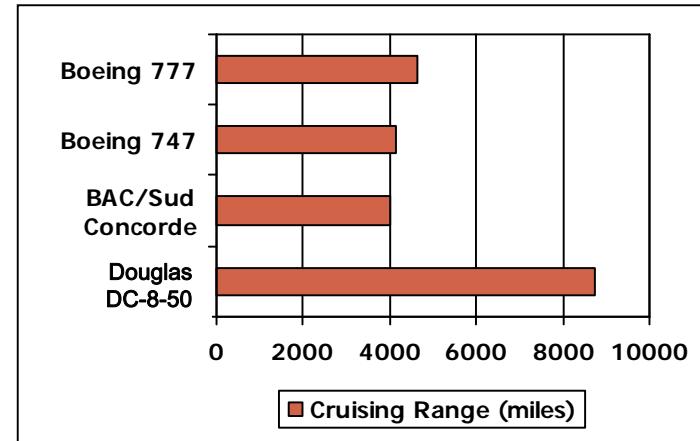
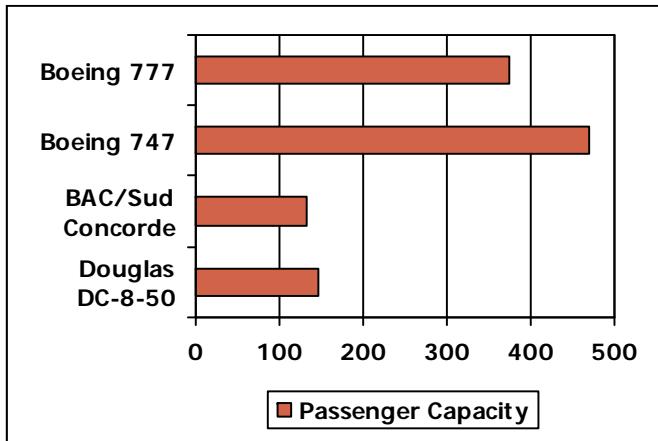
- Se incrementa la capacidad y el rendimiento
- Se reduce el coste



Año	Tecnología	Relación rendimiento/coste
1951	Tubo de vacío	1
1965	Transistor	35
1975	Circuito integrado (IC)	900
1995	“Very large scale IC” (VLSI)	2.400.000
2013	“Ultra large scale IC”	250.000.000.000

Definamos el rendimiento

- ¿Qué avión tiene el mejor rendimiento?



Tiempo de respuesta y productividad

- **Tiempo de respuesta**
 - Cuánto tiempo le lleva terminar una tarea
- **Productividad**
 - Trabajo total realizado por unidad de tiempo
 - Por ejemplo, tareas/transacciones/... por hora /por segundo
- **¿Cómo se ven afectados el tiempo de respuesta y la productividad por ...**
 - Reemplazar el procesador por una versión más rápida?
 - Añadir más procesadores?
- **Nos concentraremos en el tiempo de respuesta, por ahora.**

Rendimiento relativo

- **Definimos**

Rendimiento = 1/Tiempo de ejecución

- **“X es n veces más rápido que Y”**

Rendimiento_X/Rendimiento_Y =

= Tiempo de ejecución_Y/Tiempo de ejecución_X =

= $t_Y/t_X = n$ (**X es n veces más rápido que Y**)

- Ejemplo: tiempo invertido en ejecutar un programa

- 10 s en A, 15 s en B

- $t_B/t_A = 15 \text{ s} / 10 \text{ s} = 1.5$

- A es 1.5 veces más rápido que B

La medida del tiempo de ejecución

- **Tiempo invertido (*elapsed time*)**

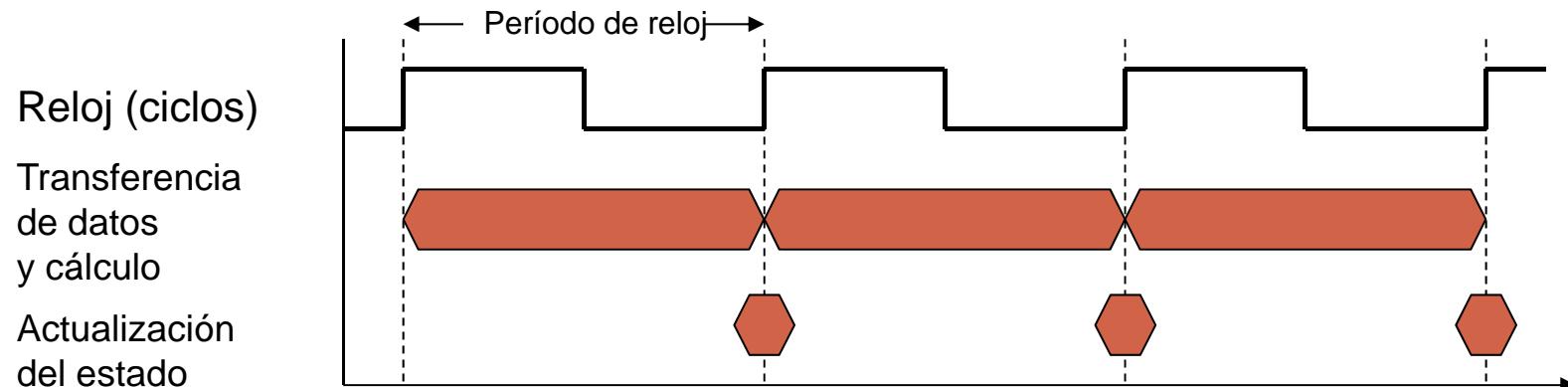
- Tiempo de respuesta total, incluyendo todos los conceptos
 - ✖ Procesamiento, I/O, labores del SO, tiempo de espera
- Determina el rendimiento del sistema

- **Tiempo de CPU (*CPU time o runtime*)**

- Tiempo consumido procesando un determinado trabajo
 - ✖ Se descuenta el tiempo de I/O y otros trabajos con recursos compartidos
- Incluye el tiempo de CPU de usuario y el tiempo de CPU del SO
- Los distintos programas se ven afectados de forma diferente por los rendimientos de la CPU y del sistema

Temporización de la CPU

- El funcionamiento del hardware digital está gobernado por un reloj de frecuencia constante



- Período de reloj (T): duración de un ciclo de reloj
 - Por ejemplo, $250 \text{ ps} = 0.25 \text{ ns} = 250 \times 10^{-12} \text{ s}$
- Frecuencia de reloj (f): ciclos por segundo
 - Por ejemplo, $4.0 \text{ GHz} = 4000 \text{ MHz} = 4.0 \times 10^9 \text{ Hz}$

Tiempo de CPU

$t = \text{Tiempo de CPU} =$

$= \text{Número de ciclos} \times \text{Duración del ciclo} = N \times T =$

$$= \frac{\text{Número de ciclos}}{\text{Frecuencia de reloj}} = \frac{N}{f}$$

- **El rendimiento mejora al ...**

- Rebajar el número de ciclos de reloj
- Aumentar la frecuencia de reloj
- El diseñador de hardware frecuentemente establece un compromiso entre frecuencia de reloj frente a número de ciclos

Ejemplo de tiempo CPU

- **Computador A: reloj 2 GHz , tiempo CPU 10 s**
- **Al diseñar el Computador B**
 - Se pretende un tiempo CPU de 6 s
 - Se puede acelerar el reloj, pero causa $1.2 \times$ número de ciclos
- **¿Cuál debe ser la frecuencia de reloj de B?**

$$f_B = \frac{N_B}{t_B} = \frac{1.2 \times N_A}{6 \text{ s}}$$

$$N_A = t_A \times f_A = 10 \text{ s} \times 2 \text{ GHz} = 20 \times 10^9 \text{ instrucciones}$$

$$f_B = \frac{1.2 \times N_A}{6 \text{ s}} = \frac{1.2 \times 20 \times 10^9}{6} = \frac{24 \times 10^9}{6} = 4 \text{ GHz}$$

Número de instrucciones y CPI

N = Número de ciclos =

= Número de instrucciones x Ciclos por instrucción = $Ic \times CPI$

$$t = N \times T = Ic \times CPI \times T = \frac{Ic \times CPI}{f}$$

- **Número de instrucciones para un programa (Ic)**
 - Está determinado por el programa, ISA y el compilador
- **Promedio de ciclos por instrucción**
 - Determinado por el hardware de la CPU
 - Si distintas instrucciones tienen distintos CPI
 - El CPI promedio estará afectado por la combinación de instrucciones

Ejemplo de CPI

- Computador A: Tiempo de ciclo = 250 ps, CPI = 2.0
- Computador B: Tiempo de ciclo = 500 ps, CPI = 1.2
- Misma ISA (mismo número de instrucciones)
- **¿Cuál es más rápido y cuánto más?**

$$t_A = Ic \times CPI_A \times T_A =$$

$$= Ic \times 2.0 \times 250 \text{ ps} = Ic \times 500 \text{ ps}$$

A es más rápido...

$$t_B = Ic \times CPI_B \times T_B =$$

$$= Ic \times 1.2 \times 500 \text{ ps} = Ic \times 600 \text{ ps}$$

$$\frac{t_B}{t_A} = \frac{Ic \times 600 \text{ ps}}{Ic \times 500 \text{ ps}} = 1.2$$

...esto más:
(Ganancia de velocidad
o aceleración)

CPI en más detalle

- Si diferentes tipos de instrucciones implican distintos números de ciclos:

$$N = \sum_{i=1}^n (CPI_i \times Ic_i)$$

- CPI: media ponderada

$$CPI = \frac{N}{Ic} = \sum_{i=1}^n \left(CPI_i \times \frac{Ic_i}{Ic} \right) = \frac{1}{Ic} \sum_{i=1}^n (CPI_i \times Ic_i)$$

↑
Frecuencia relativa
instrucciones de tipo i

Ejemplo de CPI

- Diferentes secuencias de código compilado que utilizan instrucciones de tipos A, B y C

Tipo	A	B	C
CPI para ese tipo	1	2	3
l_c (nº) en la secuencia 1	2	1	2
l_c (nº) en la secuencia 2	4	1	1

- Secuencia 1: $l_c = 5$
 - Ciclos de reloj:
 $2 \times 1 + 1 \times 2 + 2 \times 3 = 10$
 - CPI prom.= $10/5 = 2.0$
- Secuencia 2: $l_c = 6$
 - Ciclos de reloj:
 $4 \times 1 + 1 \times 2 + 1 \times 3 = 9$
 - CPI prom.= $9/6 = 1.5$

Resumen sobre rendimiento

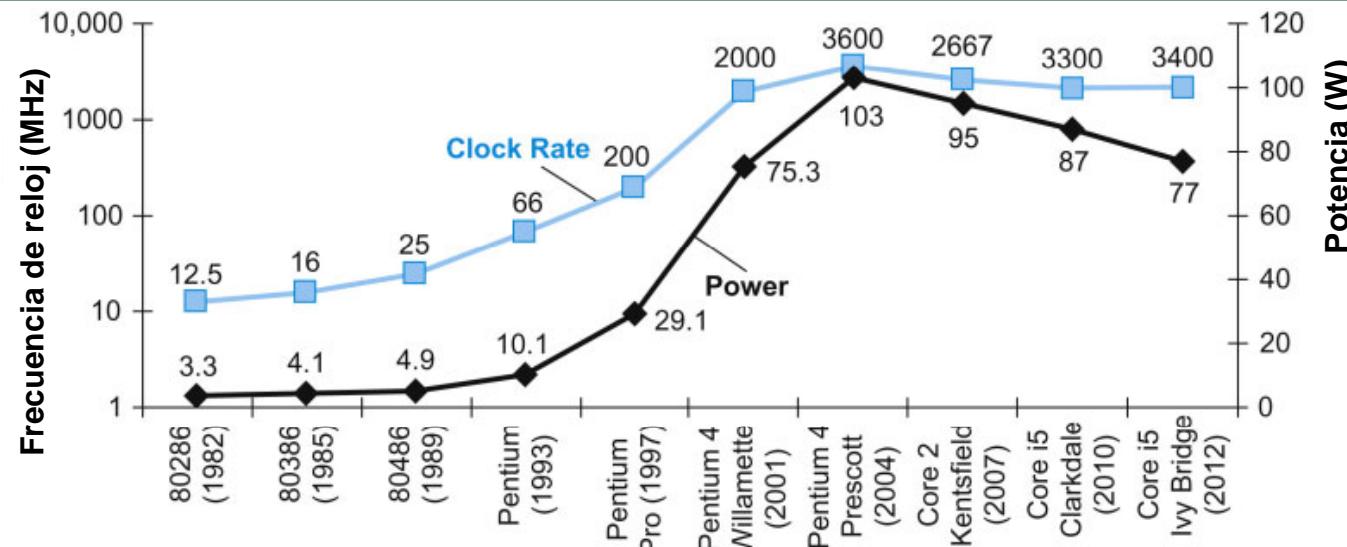
Recuadro importante

$$t = \frac{\text{Instrucciones}}{\text{Programa}} \times \frac{\text{Ciclos}}{\text{Instrucción}} \times \frac{\text{Tiempo}}{\text{Ciclo de reloj}}$$

$$t = Ic \times CPI \times T = \frac{Ic \times CPI}{f}$$

- **El rendimiento depende del ...**
 - Algoritmo: afecta a Ic , posiblemente a CPI
 - Lenguaje de programación: afecta a Ic , CPI
 - Compilador: afecta a Ic , CPI
 - ISA: afecta a Ic , CPI , T

Tendencia en la potencia



- Para la tecnología de IC CMOS:

Potencia dinámica Factor de actividad Carga capacitiva Voltaje Frecuencia
 ↓ ↓ ↓ ↓ ↓

$$W_{din} = \alpha CV^2 f$$

 Potencia estática Corrientes de fuga
 ↓ ↓

$$W_{st} = VI_{fuga}$$

Reducción de la potencia

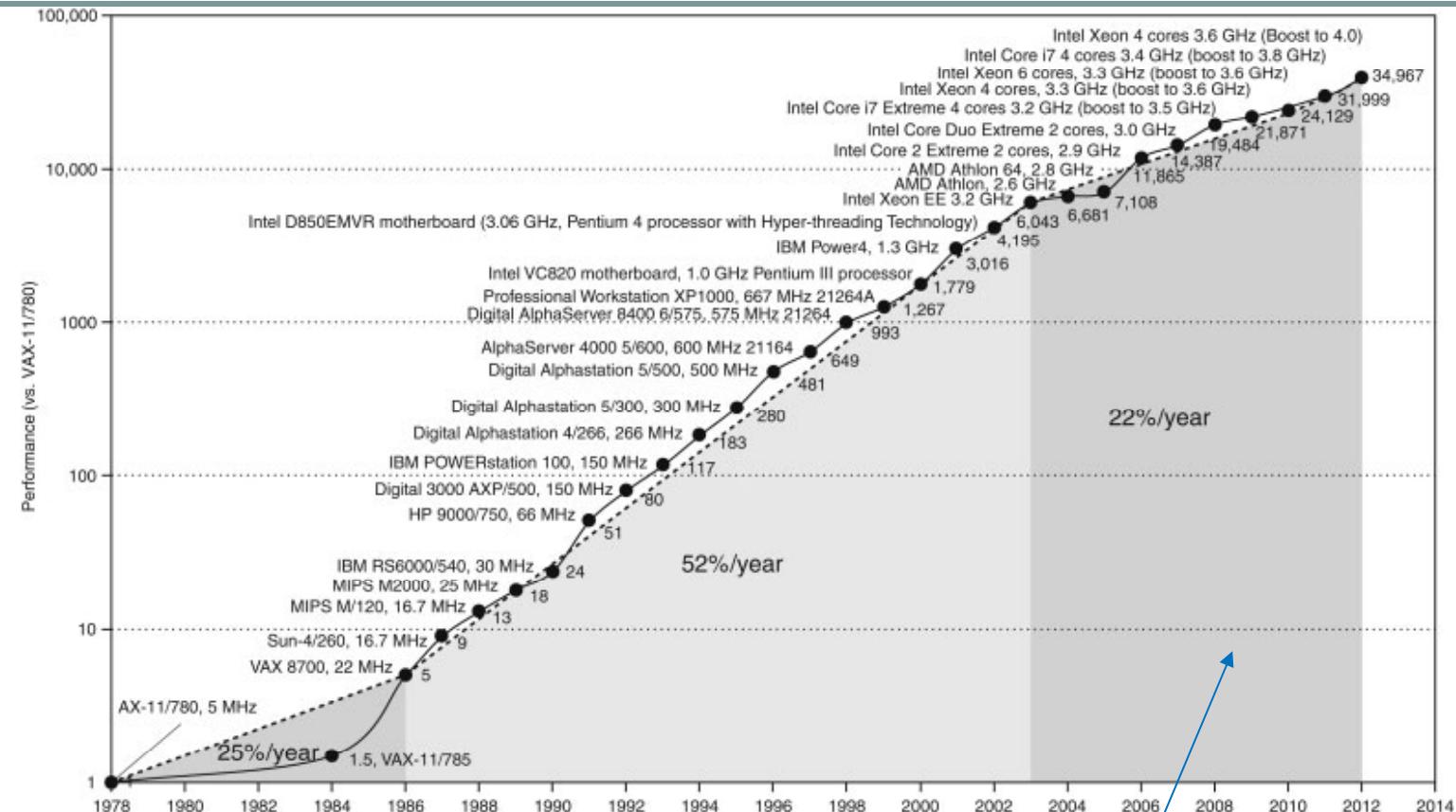
- **Supongamos que una nueva CPU tiene**

- 85 % de la carga capacitiva de la antigua CPU
- Reducción de 20 % en voltaje y 10 % en frecuencia

$$\frac{W_{nueva}}{W_{anterior}} = \frac{\alpha C_{ant.} \times 0.85 \times (V_{ant.} \times 0.80)^2 \times f_{ant.} \times 0.90}{\alpha C_{ant.} V_{ant.}^2 f_{ant.}} = \\ = 0.85 \times 0.8^2 \times 0.9 = 0,49$$

- Muro de la potencia:
 - No podemos reducir más el voltaje
 - No podemos eliminar más calor
- ¿Cómo podemos así mejorar rendimiento?

Rendimiento monoprocesador



Limitado por la potencia, el paralelismo a nivel de instrucciones y la latencia de memoria

Multiprocesadores

- **Microprocesadores multinúcleo**
 - Más de un núcleo procesador en cada chip
- **Requieren programación explícitamente paralela**
 - En comparación en el paralelismo de instrucciones ...
 - El hardware ejecuta varias instrucciones a la vez
 - El programador no lo ve
 - Es difícil
 - Programar obteniendo buen rendimiento
 - Equilibrar las cargas
 - Optimizar comunicación y sincronización

Conclusiones

- **Está mejorando la relación coste/rendimiento**
 - Debido al desarrollo de la tecnología subyacente
- **Se establecen niveles jerárquicos de abstracción**
 - Tanto para el hardware como el software
- **Arquitectura del juego de instrucciones (ISA)**
 - Es la interfaz entre hardware y software
- **La mejor medida del rendimiento es el inverso del tiempo de ejecución**
- **La potencia es un factor limitador**
 - Se utiliza el paralelismo para mejorar el rendimiento