

# Estructura de Computadores

## *Módulo A.* Introducción

### Tema 1. Introducción a la estructura de computadores

#### 1. Introducción

#### 2. Niveles de descripción de un computador

Niveles de descripción. Conceptos de arquitectura y organización. Concepto de familia.

#### 3. Modelo Von Neumann

Módulos básicos. Características principales. Elementos de la CPU.

#### 4. Medidas de rendimiento

Tiempo de ejecución. Tiempo de CPU. MIPS. MFLOPS.

#### 5. Perspectiva histórica

Generaciones de computadores. Evolución en el rendimiento del computador.

Dentro de un computador real

# 1. introducción

## ¿Qué es un computador?

[Hamacher 96]: "Un computador es una máquina de cálculo electrónica de alta velocidad que acepta información digitalizada, la procesa atendiendo a una lista de instrucciones que almacena internamente, y produce la correspondiente información de salida"

### ⊗ Funciones de un computador

- **Procesamiento de datos**
- **Almacenamiento de datos**
- **Transferencias de datos entre el computador y el exterior**
- **Control de las anteriores operaciones**

### ⊗ La excesiva generalidad de estas funciones se debe a que la especialización funcional de un computador ocurre cuando se programa y no cuando se diseña

### ⊗ Sus principales componentes estructurales son:

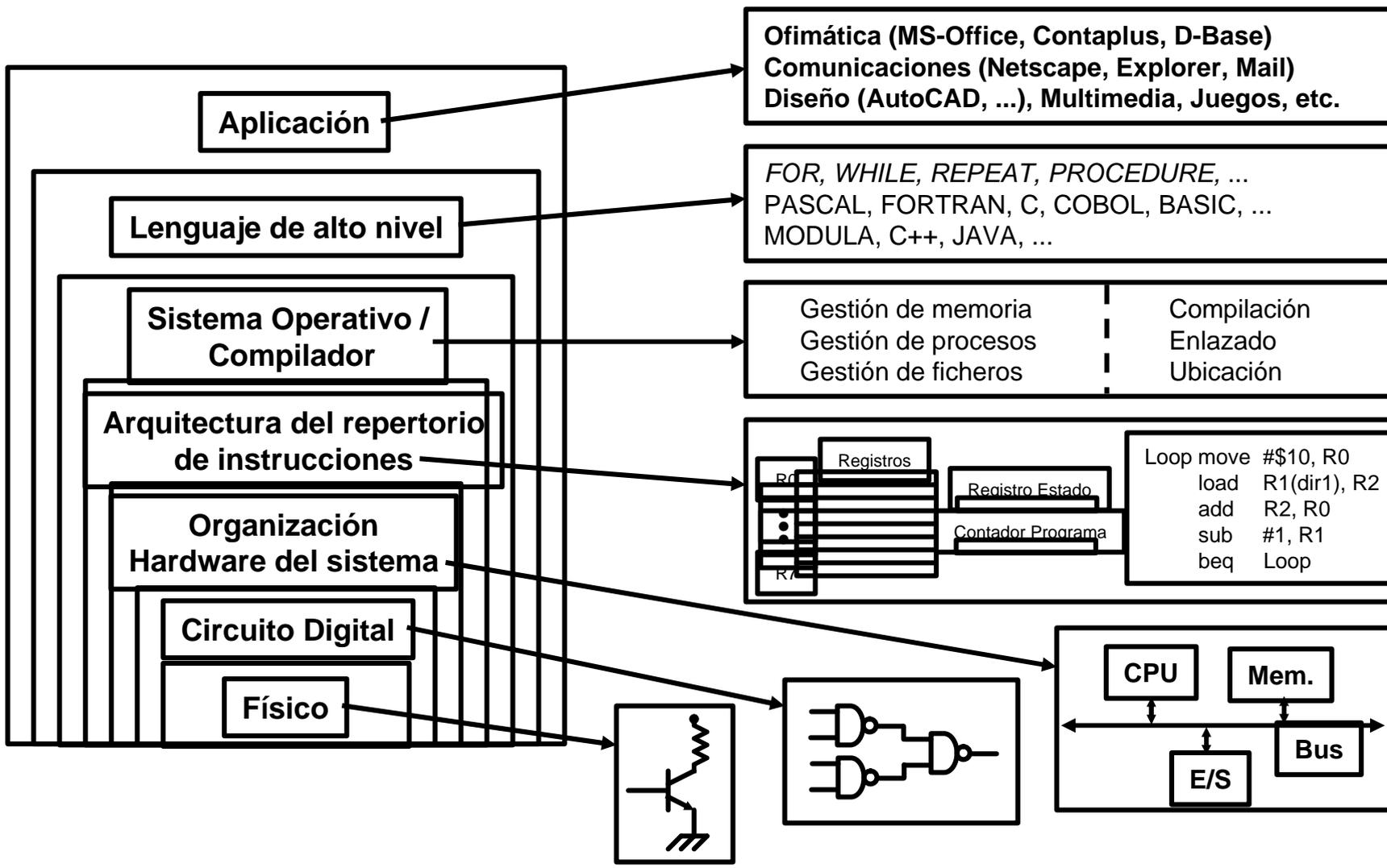
- **Procesador:** controla el funcionamiento del computador y procesa los datos
- **Subsistema de memoria:** almacena datos
- **Subsistema de entrada/salida:** transfiere datos entre el computador y el entorno externo
- **Subsistema de interconexión:** proporciona un medio de comunicación entre el procesador, la memoria y la E/S.

# 1. introducción

## Dificultades en el estudio del computador

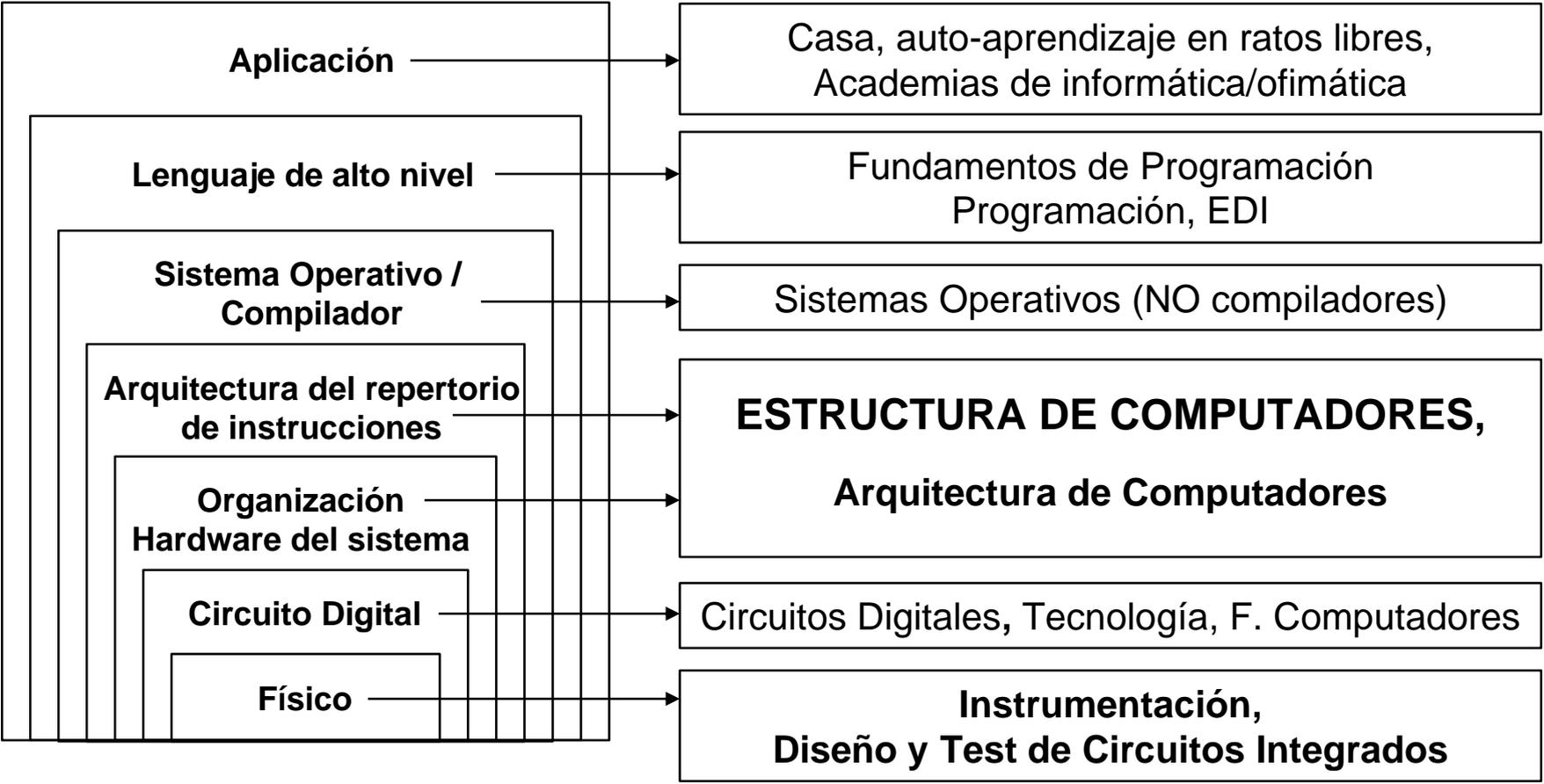
- ⊗ **Existen una gran variedad de sistemas**
  - Computadores personales
  - Estaciones de trabajo (*workstations*)
  - Mainframes
  - Supercomputadores
- ⊗ **Es un campo extremadamente cambiante ya que la tecnología avanza a pasos agigantados:**
  - **Procesador:**
    - ⇒ la densidad de integración aumenta un 30% anual
    - ⇒ las frecuencias de funcionamiento aumentan un 20% anual
    - ⇒ la potencia de cálculo aumenta un 50% anual
  - **Memoria:**
    - ⇒ la capacidad de almacenamiento (DRAMs) aumenta un 60% anual
    - ⇒ velocidad de transferencia aumenta un 10% anual
    - ⇒ coste por bit disminuye un 25% anual
  - **Discos:**
    - ⇒ la capacidad de almacenamiento aumenta un 60% anual
- ⊗ **Sin embargo, se aplican sistemáticamente ciertos conceptos fundamentales**

# 2. niveles de descripción de un computador



# 2. niveles de descripción de un computador

## ¿Dónde se estudia?



## 2. niveles de descripción de un computador

### Arquitectura del repertorio de instrucciones

- ⊗ Conjuntos de atributos de un computador que son visibles a:
  - El programador en lenguaje máquina,
  - El sistema operativo o
  - El compilador
- ⊗ Engloba los siguientes elementos
  - Conjunto de registros visibles al programador
  - Conjunto de instrucciones máquina o ensamblador
  - Tipos básicos de datos soportados por las instrucciones máquina
  - Modos de direccionamiento
  - Mecanismos de E/S

### Organización hardware del sistema

- ⊗ Conjunto de módulos básicos que componen el computador
  - CPU (ALU, unidad de control, registros, ...)
  - Memoria (memoria principal, memoria cache, memoria virtual)
  - E/S (periféricos, controladores, sistema de interrupciones, DMA, ...)
  - Buses (buses del sistema, buses de expansión, etc.)
- ⊗ Funcionamiento y alternativas de implementación de los módulos básicos
- ⊗ Interconexión e interacción entre los mismos

## 2. niveles de descripción de un computador

### Familia de computadores

- ⊗ **Conjunto de computadores con**
  - Similar arquitectura
  - Distinta organización
- ⊗ **Las familias de computadores hacen posible que**
  - Existan máquinas de la misma familia con distinta
    - ✓ Tecnología
    - ✓ Velocidad
    - ✓ Prestaciones
    - ✓ Precio
  - Las máquinas de una misma familia sean compatibles entre sí
    - ✓ Todos los miembros de una misma familia pueden ejecutar los mismos programas
- ⊗ **La compatibilidad suele ser sólo hacia arriba (upward compatibility)**
  - Las máquinas de gama alta de una familia pueden ejecutar los mismos programas que las máquinas de gama baja
  - Lo contrario (compatibilidad hacia abajo) no suele ser cierto

### Ejemplos de Familias

#### Familia PC compatible

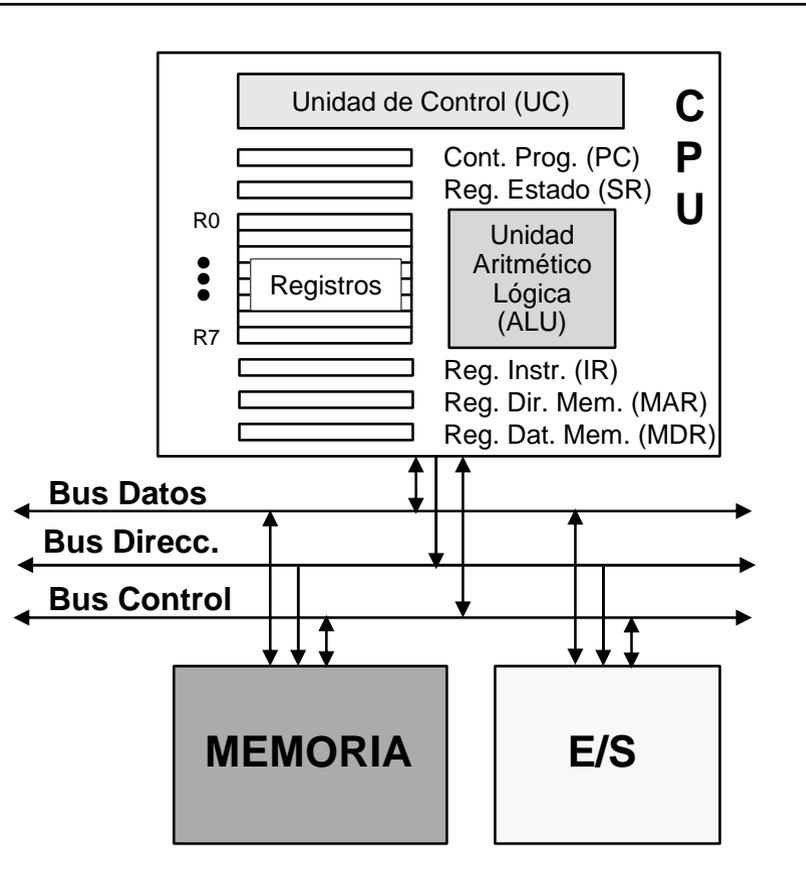
- ⊗ IBM PC-XT (8086)
- ⊗ IBM PC-AT (80286)
- ⊗ 80386
- ⊗ 80486
- ⊗ Pentium
- ⊗ Pentium Pro
- ⊗ Pentium II
- ⊗ Pentium III
- ⊗ Pentium 4

#### Familia Motorola 68XXX

- ⊗ MC68000
- ⊗ MC68010
- ⊗ MC68020
- ⊗ MC68030
- ⊗ MC68040
- ⊗ MC68060

# 3. modelo Von Neumann

## Esquema



## Módulos básicos

### CPU (Unidad Central de Proceso)

- Realiza la ejecución de las instrucciones

### Unidad de Memoria

- Almacena las instrucciones y los datos

### Unidad de E/S

- Transfiere información entre el computador y los dispositivos periféricos

## Elementos de interconexión: BUSES

### Bus de datos

- Para transferencia de datos entre la CPU y memoria o E/S

### Bus de direcciones

- Para especificar la dirección de memoria o la dirección del registro de E/S

### Bus de control

- Señales de control de la transferencia (reloj, lectura/escritura, etc.)

# 3. modelo Von Neumann

## Características principales del modelo Von Neumann

- ⊗ **Su funcionamiento se basa en el concepto de programa almacenado en memoria**
  - **La memoria principal almacena**
    - ⇒ Instrucciones: **programa que controla el funcionamiento del computador**
    - ⇒ Datos: **datos que procesa y genera dicho programa**
- ⊗ **Las palabras en memoria siguen una organización lineal**
  - **Todas las palabras de memoria tienen el mismo tamaño**
  - **No hay distinción explícita entre instrucciones y datos**
- ⊗ **La ejecución de las instrucciones es secuencial**
  - **El secuenciamiento de las instrucciones es implícito, y viene determinado por el orden en que han sido almacenadas en la memoria**
  - **Este secuenciamiento sólo puede ser modificado por instrucciones específicas de salto**
  - **El contador de programa (PC) indica en cada instante cual es la siguiente instrucción a ejecutar**
- ⊗ **Las fases que se distinguen en la ejecución de una instrucción son**
  - **Búsqueda de la instrucción en memoria (*Fetch*) y cálculo de la direcc. de instrucción siguiente**
  - **Decodificación de la instrucción por parte de la CPU**
  - **Búsqueda de los operandos de la instrucción**
  - **Ejecución de la instrucción**
  - **Escritura del resultado**

# 3. modelo Von Neumann

## Elementos de la CPU

Unidad de proceso o ruta de datos (*data-path*)

⊗ **Unidad Aritmético-Lógica (ALU)**

- Realiza las operaciones aritméticas y lógicas que indican las instrucciones del programa

⊗ **Banco de Registros**

- Conjunto de registros visibles al usuario
- Almacena los datos y los resultados con los que trabaja la ALU

⊗ **Registros especiales**

- Contador de programa, registro de estado, dirección del dato en memoria, etc.

⊗ **Buses internos**

- Caminos de interconexión entre los elementos anteriores

Unidad de control o ruta de control (*control-path*)

⊗ **Genera las señales necesarias para que la unidad de proceso ejecute las instrucciones de forma adecuada**

- Indica el tipo de operación que tiene que realizar la ALU
- Indica que registros contienen los datos y dónde se debe almacenar el resultado
- Genera señales de carga de todos los registros cuando estos tienen que almacenar información

⊗ **Es un sistema secuencial. Su complejidad depende de**

- La complejidad de la unidad de proceso y del número y tipo de instrucciones a ejecutar

## 4. medidas de rendimiento

### ¿Para qué son necesarias?

- ⊗ Permiten comparar objetivamente las prestaciones de computadores distintos:
  - Cuál ofrece mayor potencia de cálculo o mayor velocidad de procesamiento, es decir, **cuál tarda menos tiempo en ejecutar nuestras aplicaciones o programas**

### ¿Cuál es la principal métrica (desde el punto de vista del usuario individual)?

- ⊗ **Tiempo de ejecución (medido en segundos/programa)**
  - cuenta todo: tiempos de E/S, tiempos de acceso a memoria, tiempo de S.O.
  - es útil, pero depende de demasiados factores para poder ser usado eficientemente
- ⊗ **Tiempo de CPU**
  - sólo cuenta la fracción de tiempo de ejecución en la que la CPU está ocupada
  - puede ser descompuesto en:
    - ⇒ *tiempo de usuario*: tiempo empleado por la CPU para ejecutar un programa
    - ⇒ *tiempo de sistema*: tiempo empleado por la CPU para ejecutar tareas del sistema operativo en beneficio del programa

Para medir el rendimiento de un computador para un programa dado, podemos utilizar como una métrica adecuada el **tiempo de CPU de usuario**

## 4. medidas de rendimiento

### ¿Cómo puede medirse el tiempo de CPU de usuario?

- ⊗ Todos los computadores utilizan una señal periódica que determina el momento en que tienen lugar los eventos hardware: dicha señal se llama reloj
- *tiempo de ciclo*: tiempo que transcurre entre dos *pulsos de reloj* (medido en s)
  - *frecuencia de reloj*: la inversa del tiempo de ciclo (medido en  $\text{Hz}=\text{s}^{-1}$ )

Un computador funcionando a 200 MHz (frecuencia) tiene un tiempo de ciclo de:  $\frac{1}{200 \times 10^6} = 5 \cdot 10^{-9} \text{ s} = 5 \text{ ns}$

$$\begin{aligned} (\text{tiempo de CPU}) &= (\text{ciclos de reloj por programa}) \times (\text{tiempo de ciclo}) \\ &= (\text{ciclos de reloj por programa}) / (\text{frecuencia de reloj}) \end{aligned}$$

### ¿Puede asumirse que (ciclos de reloj por programa) = (instrucciones por programa)?

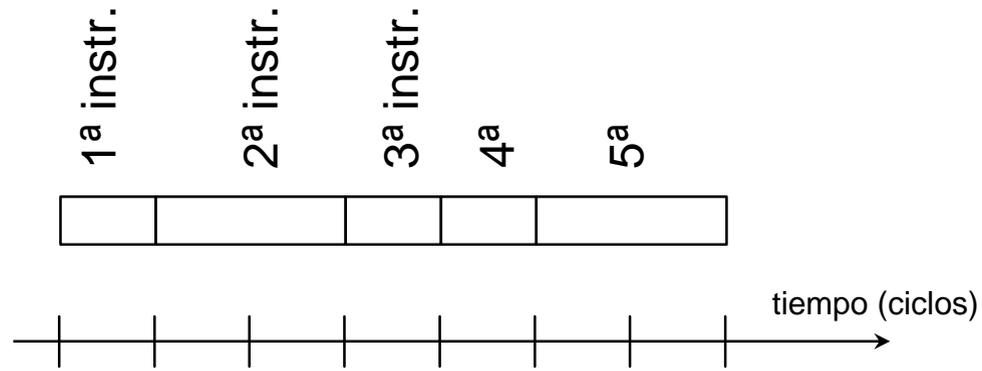
**NO:** Ya que una instrucción no siempre tarda un ciclo de reloj en ejecutarse, además diferentes instrucciones tardan diferentes cantidades de tiempo:

- ⇒ La multiplicación tarda más ciclos que la suma
- ⇒ Operaciones en punto flotante tardan más que operaciones en punto fijo
- ⇒ Instrucciones con acceso a memoria tardan más que con acceso a registros

# 4. medidas de rendimiento

## ¿Cómo puede medirse el tiempo de CPU de usuario? (Cont.)

Tiempos de ejecución distintos para diferentes instrucciones



(ciclos de reloj por programa) = (instrucciones por programa) × (ciclos promedio por instrucción)

Los “**ciclos promedio por instrucción**” (**CPI**) se calculan como una suma ponderada del número de ciclos que tarda por separado cada tipo de instrucción

<i>Operación</i>	<i>Frecuencia</i> ( $F_i$ )	<i>Ciclos</i> ( $C_i$ )	$CPI_i$ ( $F_i * C_i$ )	% tiempo ( $CPI_i / CPI$ )
ALU	50%	1	0.5	23%
Load	20%	5	1.0	45%
Store	10%	3	0.3	14%
Branch	20%	2	0.4	18%

**CPI: 2.2**

# 4. medidas de rendimiento

## ¿Cómo puede medirse el tiempo de CPU de usuario? (Cont.)

$$(\text{tiempo de CPU}) = (\text{instrucciones por programa}) \times \text{CPI} \times (\text{tiempo de ciclo})$$

⊗ ¿De qué dependen las “instrucciones por programa”?:

✓ programadores / compiladores / arquitectura del repertorio de instrucciones

⊗ ¿De qué depende el “tiempo de ciclo”?:

✓ tecnología / organización

⊗ ¿De qué dependen los “ciclos promedio por instrucción”?:

✓ organización / arquitectura del repertorio de instrucciones

ninguna de las  
variables por separado es  
medida de rendimiento

⊗ El rendimiento global de un computador se evalúa ejecutando distintos programas reales

➤ Programas “de juguete”: 10~100 líneas de código con resultado conocido

⇒ Ej.: *Criba de Eratóstenes, Puzzle, Quicksort*

➤ Programas de prueba (*benchmarks*) sintéticos: simulan la frecuencia de operaciones y operandos de un abanico de programas reales

⇒ Ej.: *Whetstone, Dhrystone*

➤ Programas reales típicos con cargas de trabajo fijas (actualmente la medida más aceptada)

⇒ *SPECint'95: LI (interprete lisp), COMPRESS (compresión de datos), GCC (compilador de C), GO (juego), PERL (intérprete PERL), VORTEX (base de datos), M88KSIM (simulador de MC88000), IJPEG (codificador de imágenes)*

# Ejemplo de medidas de rendimiento

⊗ [http://www.intel.com/es/home/index.htm?iid=esHomepage+Sites Home Research](http://www.intel.com/es/home/index.htm?iid=esHomepage+Sites_Home_Research)

## 4. medidas de rendimiento

- ⊗ Existen, además, otras métricas muy populares pero que pueden resultar engañosas por no incluir todas las variables anteriormente estudiadas.

### MIPS (Millones de instrucciones por segundo)

$$MIPS = (\text{instrucciones por programa}) / (\text{tiempo ejecución} \cdot 10^6)$$

#### ⊗ Problemas

- Depende del repertorio de instrucciones
  - ⇒ El tiempo que tarda en ejecutarse una misma instrucción y el nº de instrucciones máquina que genera el compilador puede variar de un repertorio a otro
  - ⇒ Dos programas distintos pueden tener comportamientos opuestos
- La medida de MIPS puede variar mucho de un programa a otro
  - ⇒ Existen instrucciones que tardan más tiempo en ejecutarse que otras
  - ⇒ Los programas en los que abundan instrucciones “rápidas” tardan menos en ejecutarse
  - ⇒ Los programas en los que abundan instrucciones “lentas” tardan más en ejecutarse
- Los fabricantes suelen dar medidas de MIPS muy optimistas
  - ⇒ Utilizan programas donde predominan instrucciones que tardan poco en ejecutarse

## 4. medidas de rendimiento

### MFLOPS (Millones de instrucciones en punto flotante por segundo)

$$MFLOPS = (\text{instrucciones en punto flotante por programa}) / (\text{tiempo ejecución} \cdot 10^6)$$

- ⊗ Las instrucciones en punto flotante son las que más tardan en ejecutarse
  - Son una medida algo más fiable del rendimiento real del computador
- ⊗ Problemas
  - Depende del repertorio de instrucciones en punto flotante
    - ⇒ No todos los computadores ofrecen las mismas operaciones en punto flotante
    - ⇒ Dos programas distintos pueden tener comportamientos opuestos
  - Es inútil para muchos programas
    - ⇒ Los programas enteros (sin operaciones en punto flotante) no pueden medirse en MFLOPS
  - Existen instrucciones en punto flotante de distinto tiempo de ejecución
    - ⇒ Por ejemplo: suma, resta, ... son rápidas; división, seno, exponencial, ... son lentas
    - ⇒ Los fabricantes pueden dar también medidas de MFLOPS demasiado optimistas

### Conclusión

MIPS y MFLOPS son medidas parciales poco indicativas por si solas

# 5. perspectiva histórica

## Generaciones de computadores

Generación	Fechas	Características Tecnológ.	Potencia de cálculo
Primera	1946-1957	Válvula de vacío	0,04 MIPS
Segunda	1958-1964	Transistores	0,2 MIPS
Tercera	1965-1971	Circuitos integrados	1 MIPS
Cuarta	1972-1988	Microprocesador	10 MIPS
Quinta	1988-	Sistema basados en micro	> 100 MIPS

## 1ª Generación (1946-1957)

### ⊗ ENIAC (1946)

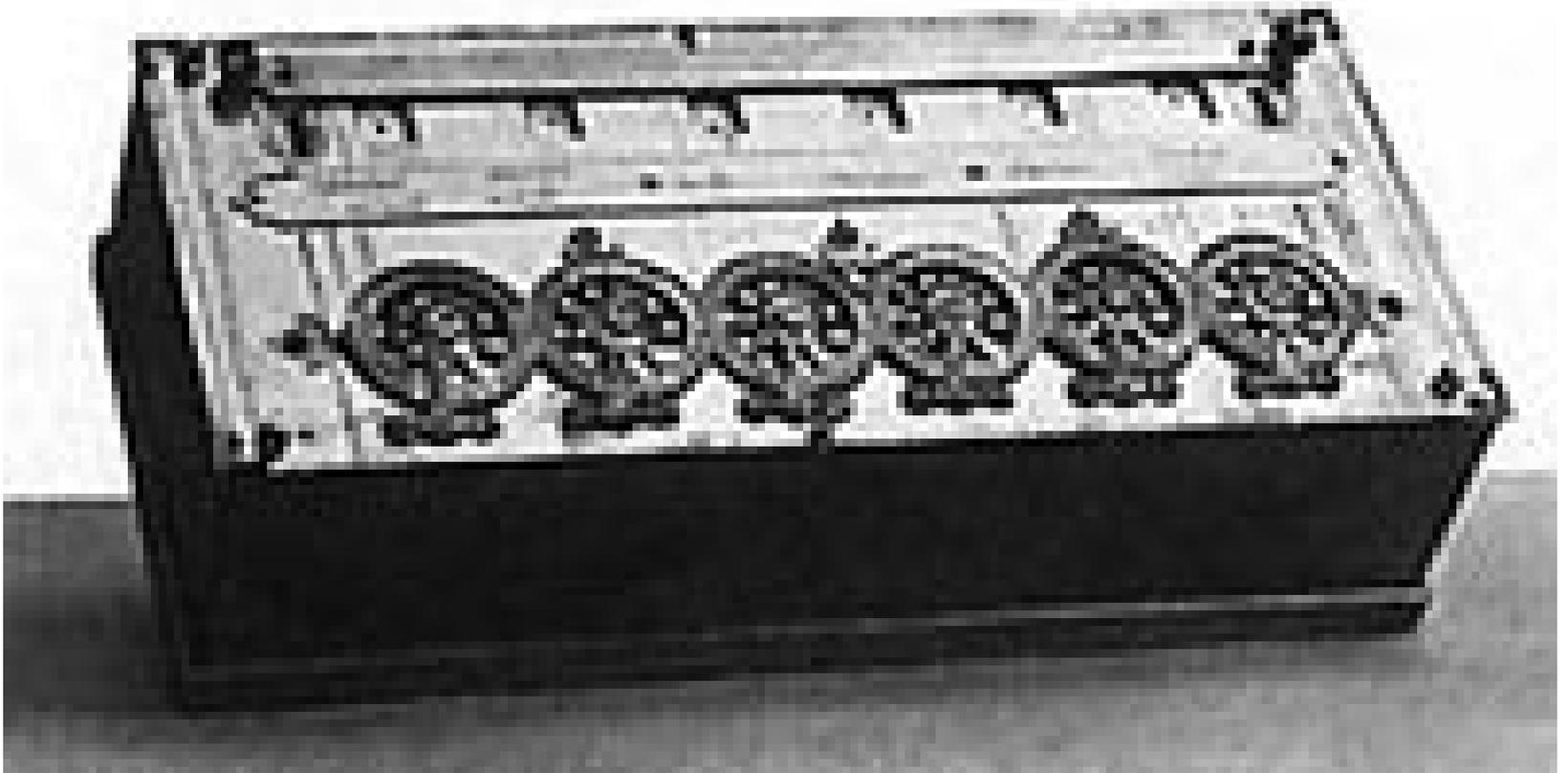
- Primer computador de propósito especial
- Peso: 30 toneladas; Superficie: 1.400 m<sup>2</sup>; Consumo: 140 KW
- Potencia cálculo: 5000 sumas/seg
- Difícil de programar. Mediante conmutadores y cables

### ⊗ Primeros computadores comerciales (Años 50)

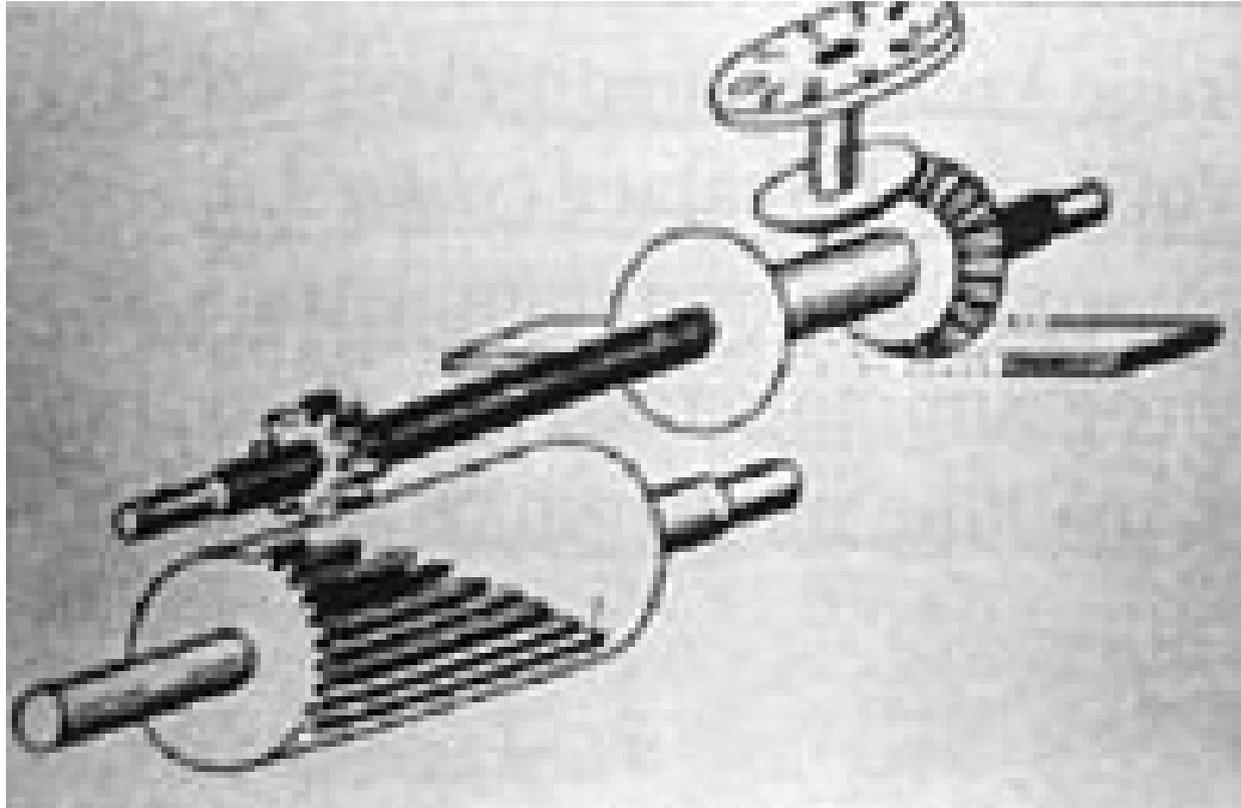
- UNIVAC I y II (Eckert & Mauchly)
- IBM Serie 700
- Programación en lenguaje máquina



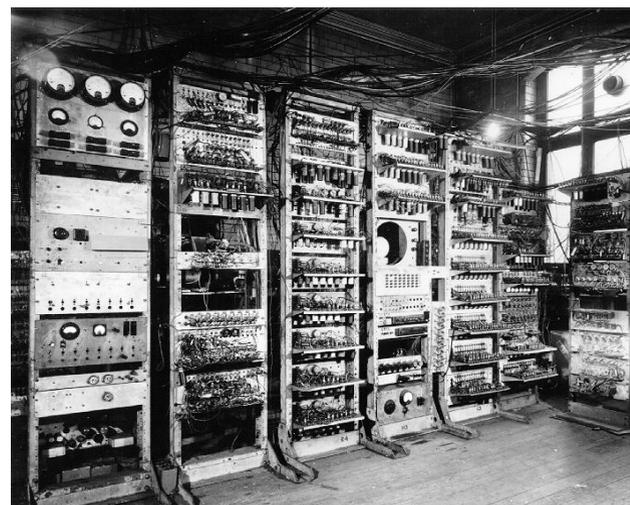
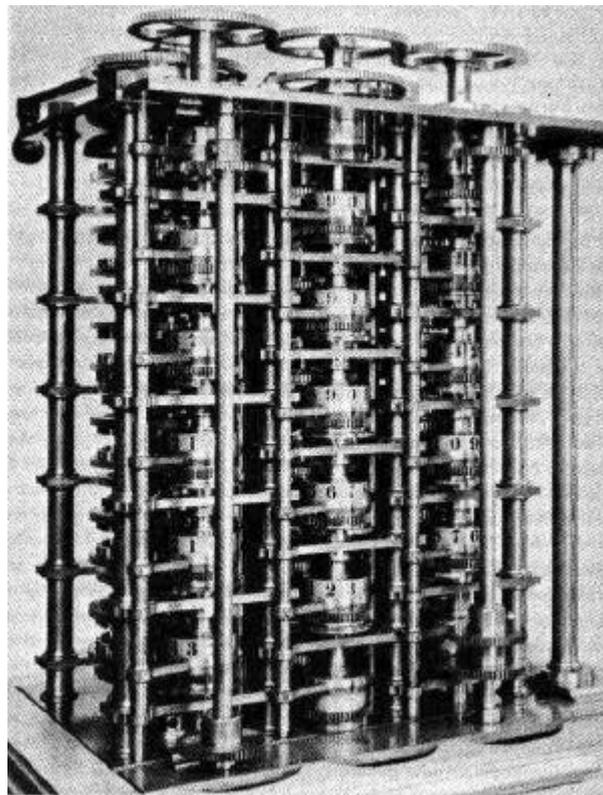
# Máquina de Pascal



# Máquina de Leibnitz

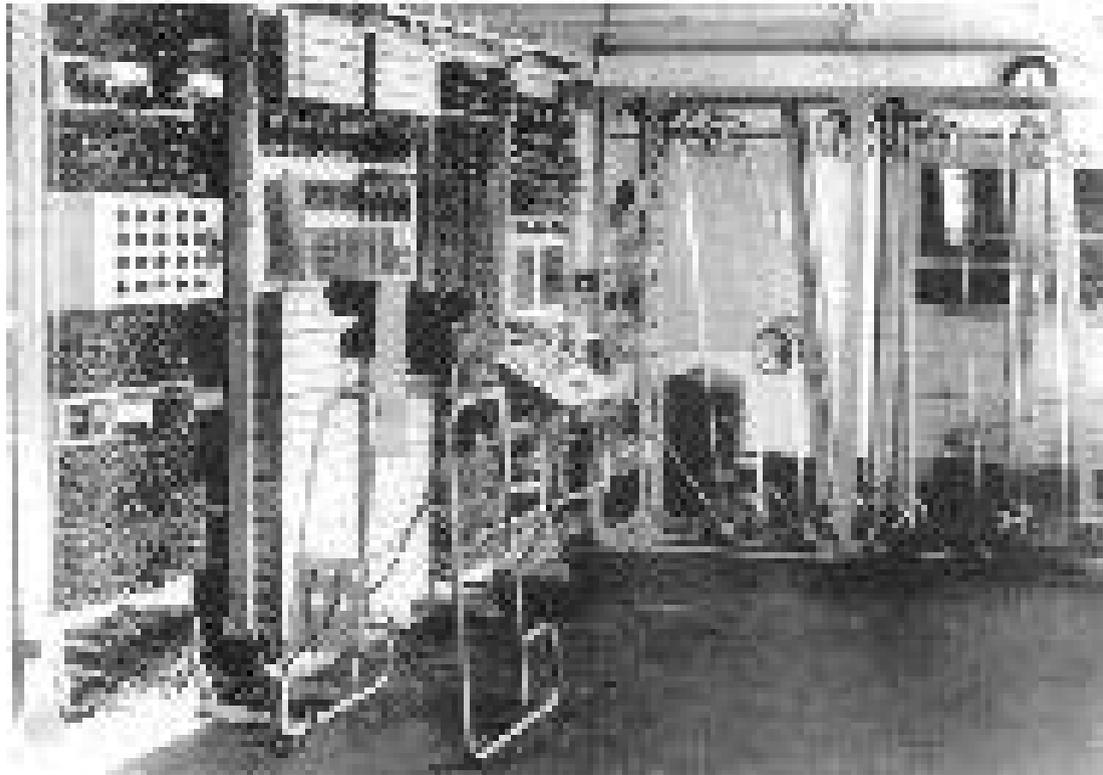


# Charles Babbage

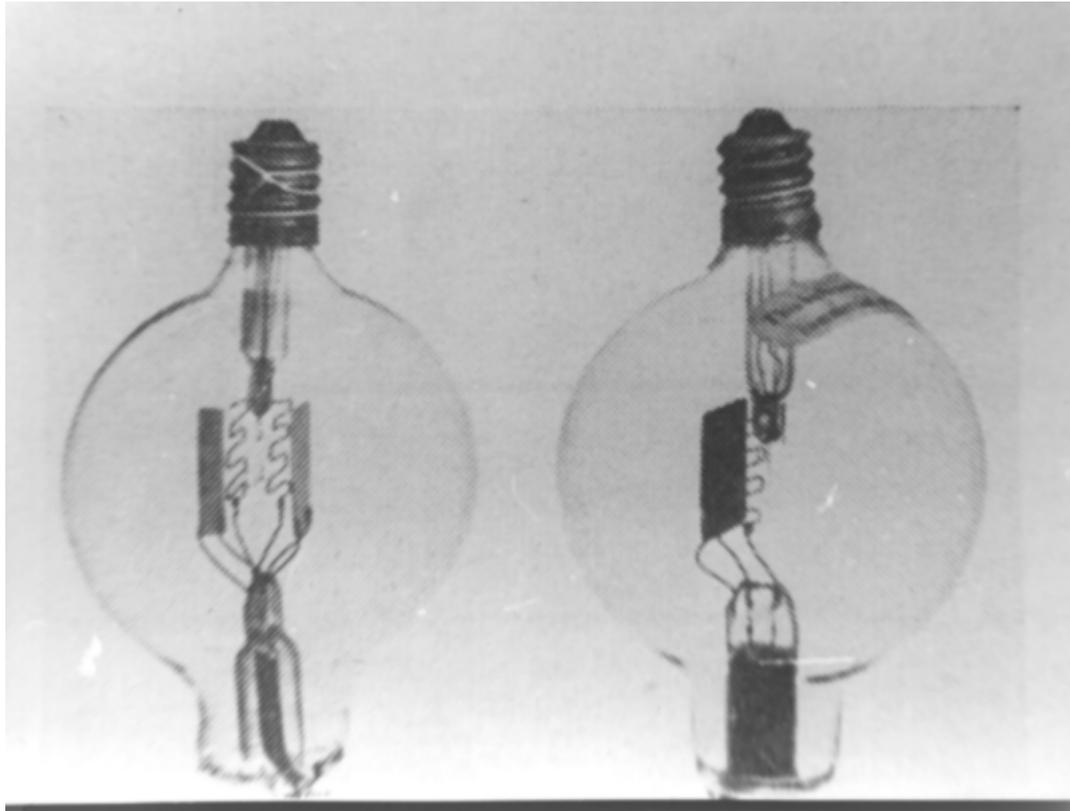


# Colossus

## Primera computadora totalmente electrónica



# Primeras válvulas de vacío



# IBM RAMAC



# 5. perspectiva histórica

## 2ª Generación (1958 - 1964)

### ⊗ El transistor

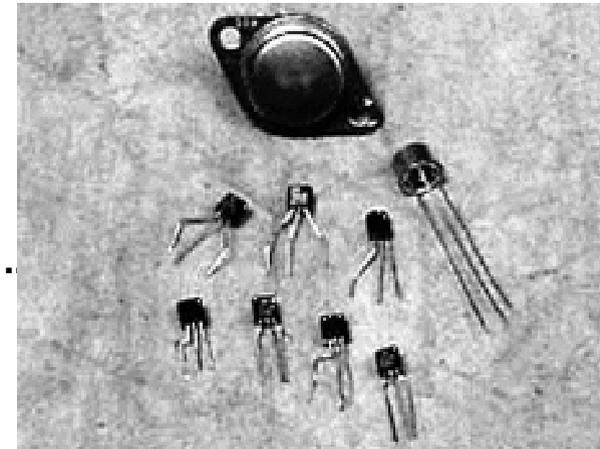
- Inventado por los laboratorios Bell en 1947
- Más pequeños, baratos y menor consumo que las válvulas de vacío
- Computadores de menor coste y tamaño, más fiables mayores prestaciones

### ⊗ Principales computadores comerciales

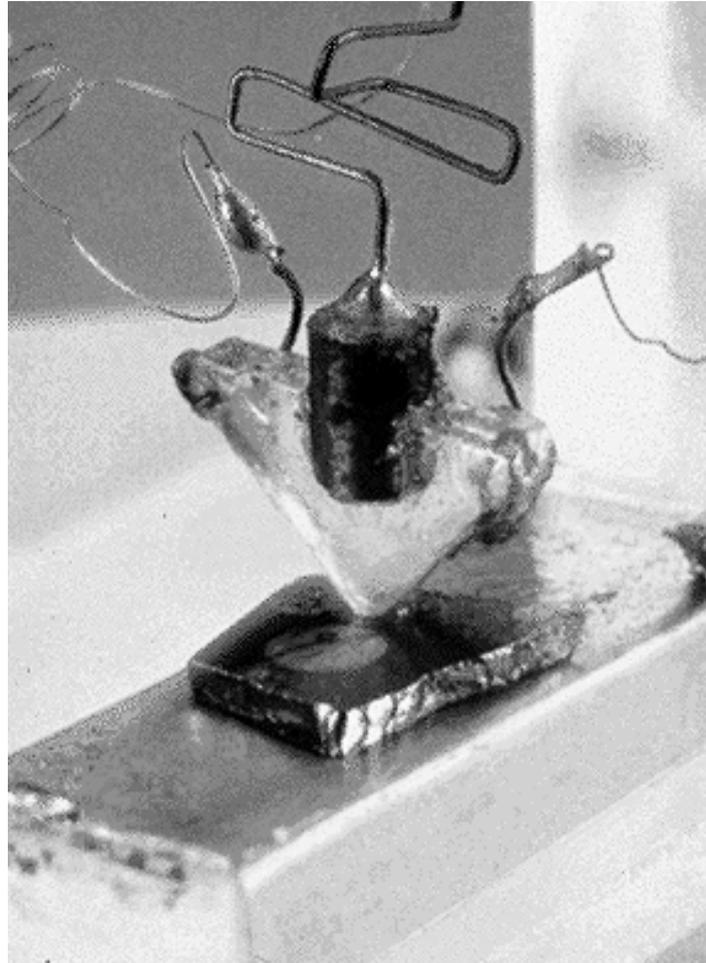
- DEC PDP-1
- IBM Serie 7000
- UNIVAC 1100

### ⊗ Otras innovaciones

- Programación en lenguajes de alto nivel (Fortran, Cobol, ...)
- Canales de E/S
- Memoria virtual
- Interrupciones para la E/S



# Transistor



# IBM 7030



# 5. perspectiva histórica

## 3ª Generación (1965 - 1971)

- ⊗ **El circuito integrado (CI)**
  - Integración de múltiples componentes (transistores, resistencias, condensadores, .. ) en un mismo CI de silicio
  - Reduce enormemente el tiempo de fabricación, el tamaño y el coste del computador
  - Aumenta aún más la fiabilidad y las prestaciones
- ⊗ **Principales computadores comerciales**
  - DEC PDP-8
  - IBM Series 360 y 370
  - UNIVAC 1108
  - CDC series 6600, 7800 y Cyber
- ⊗ **Otras innovaciones**
  - Microprogramación
  - Interrupciones con prioridades
  - Memoria cache
  - Controladores DMA
  - Nuevos lenguajes de programación (Basic, APL, Pascal)
  - Sistemas operativos robustos (MVS de IBM, VMS de DEC)

# IBM 360



# 5. perspectiva histórica

## 4ª Generación (1972 - 1988)

### ⊗ El microprocesador

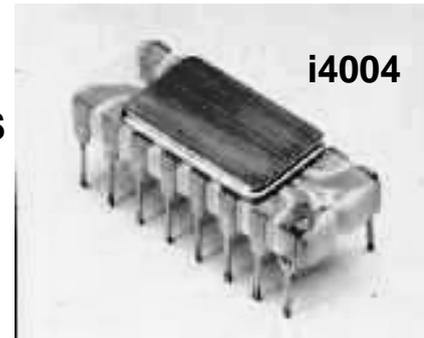
- Aumento constante en escala de integración (LSI, VLSI): se duplica cada año
- La escala LSI permite integrar todos los componentes de un procesador en un mismo chip
  - ⇒ Nace el MICROPROCESADOR
- Primer microprocesador: intel 4004 (procesador de 4 bits, 1971)

### ⊗ Aparece el computador personal (IBM/PC)

- Microsoft desarrolla el primer sistema operativo para PC: MS-DOS

### ⊗ Principales procesadores de esta generación

- De 8 bits: intel 8008, 8080; Motorola 6502, 6800
- De 16 bits (año 78): intel 8086, 80286; Motorola 68000, 68010
- De 32 bits (años 85): intel 80386; Motorola 68020, 68030



### ⊗ Surge el procesador con conjunto reducido de instrucciones (RISC)

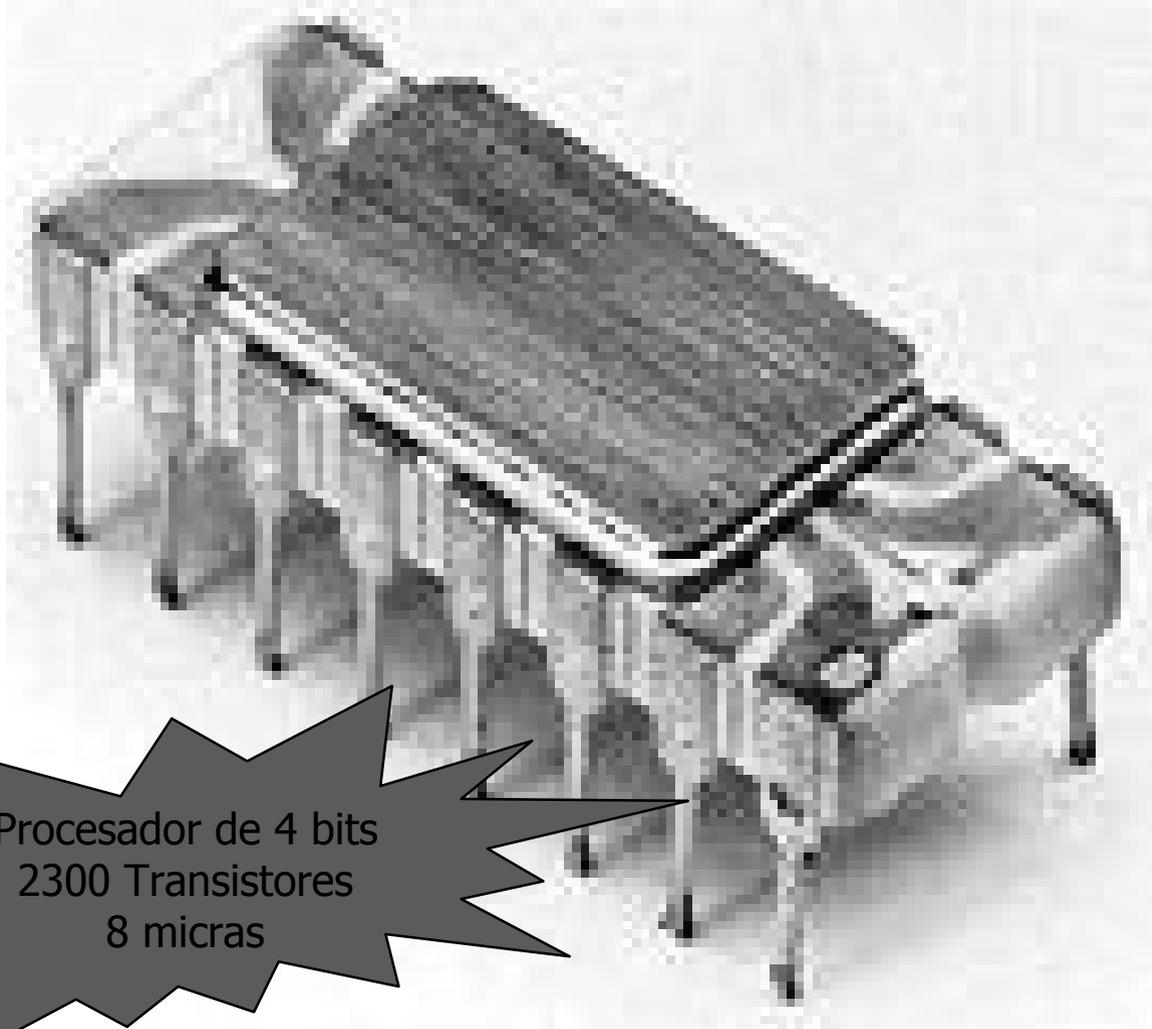
- MIPS R2000 (32 bits, año 1986)

### ⊗ Supercomputadores vectoriales y paralelos

- Basados en tecnología ECL
- Muy caros de fabricar y mantener
- Cray, NEC, Hitachi, Fujitsu

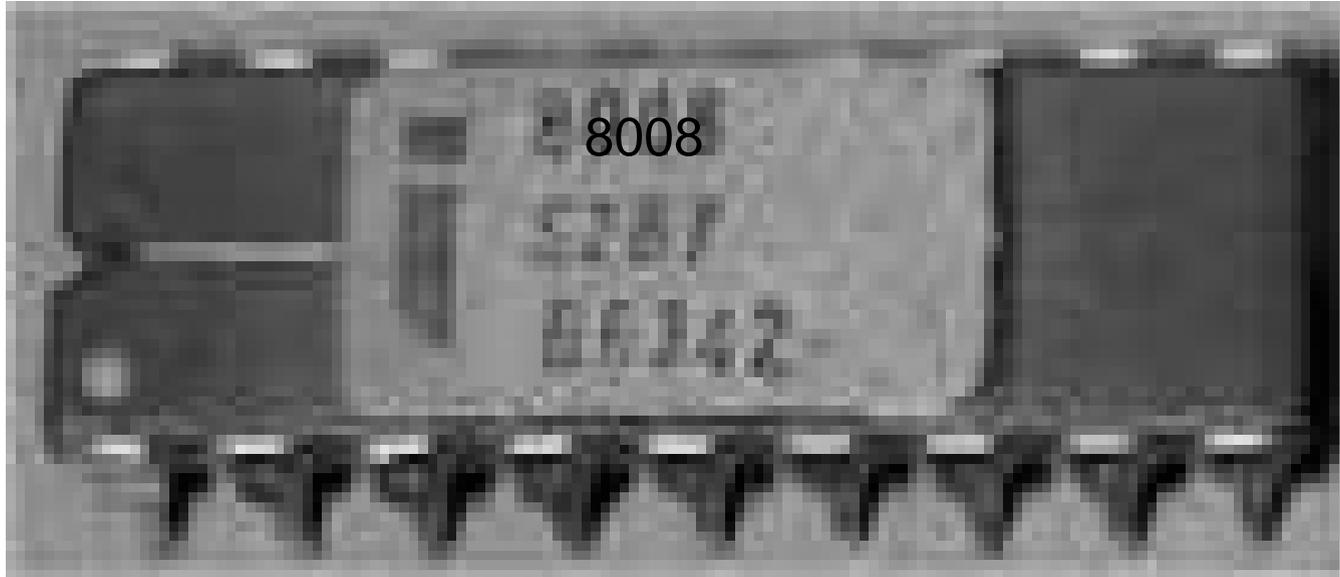
### ⊗ Aparecen las redes de computadores

# Intel 4004



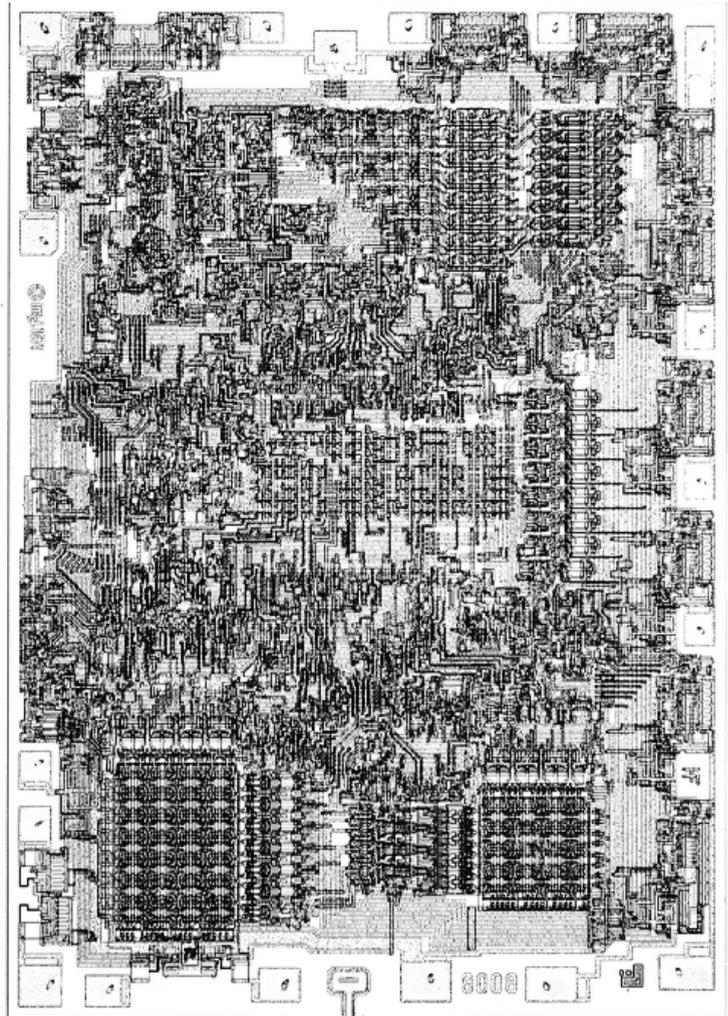
Procesador de 4 bits  
2300 Transistores  
8 micras

# Intel 8008

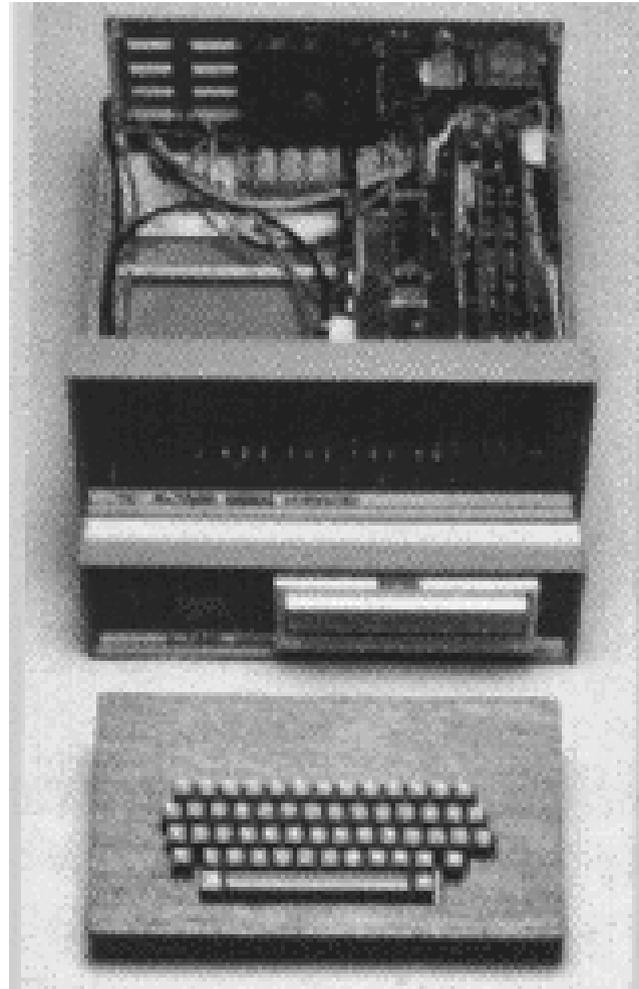


**Procesador de 8 bits**  
**3500 Transistores**  
**16 K bytes de memoria**  
**0,5 MHz**

# Intel 8008



# MITS Altair 8800



2 MHz

# Apple II (MC6502)



# CRAY 1

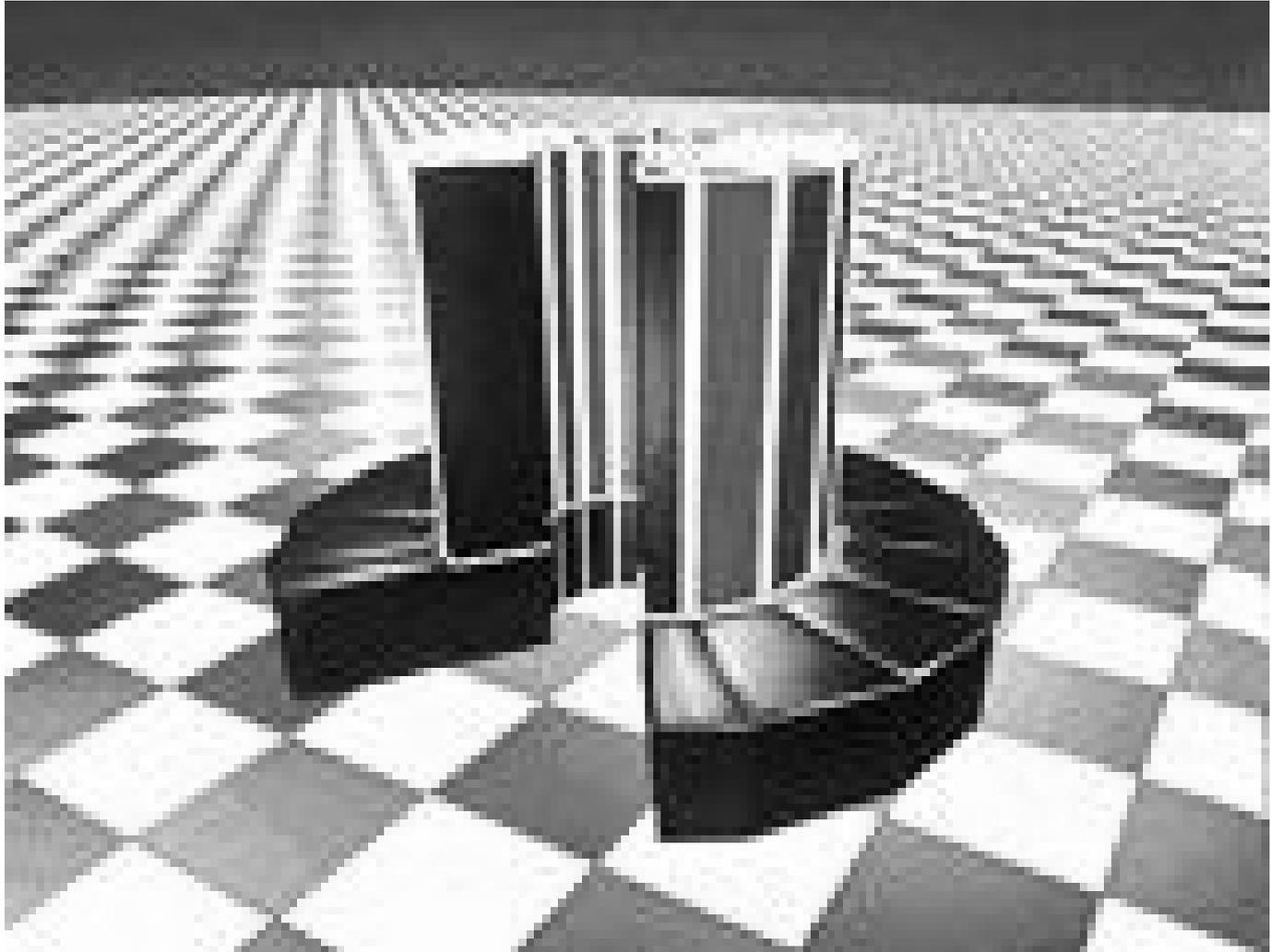


Figura 1.1: Cray 1

# 5. perspectiva histórica

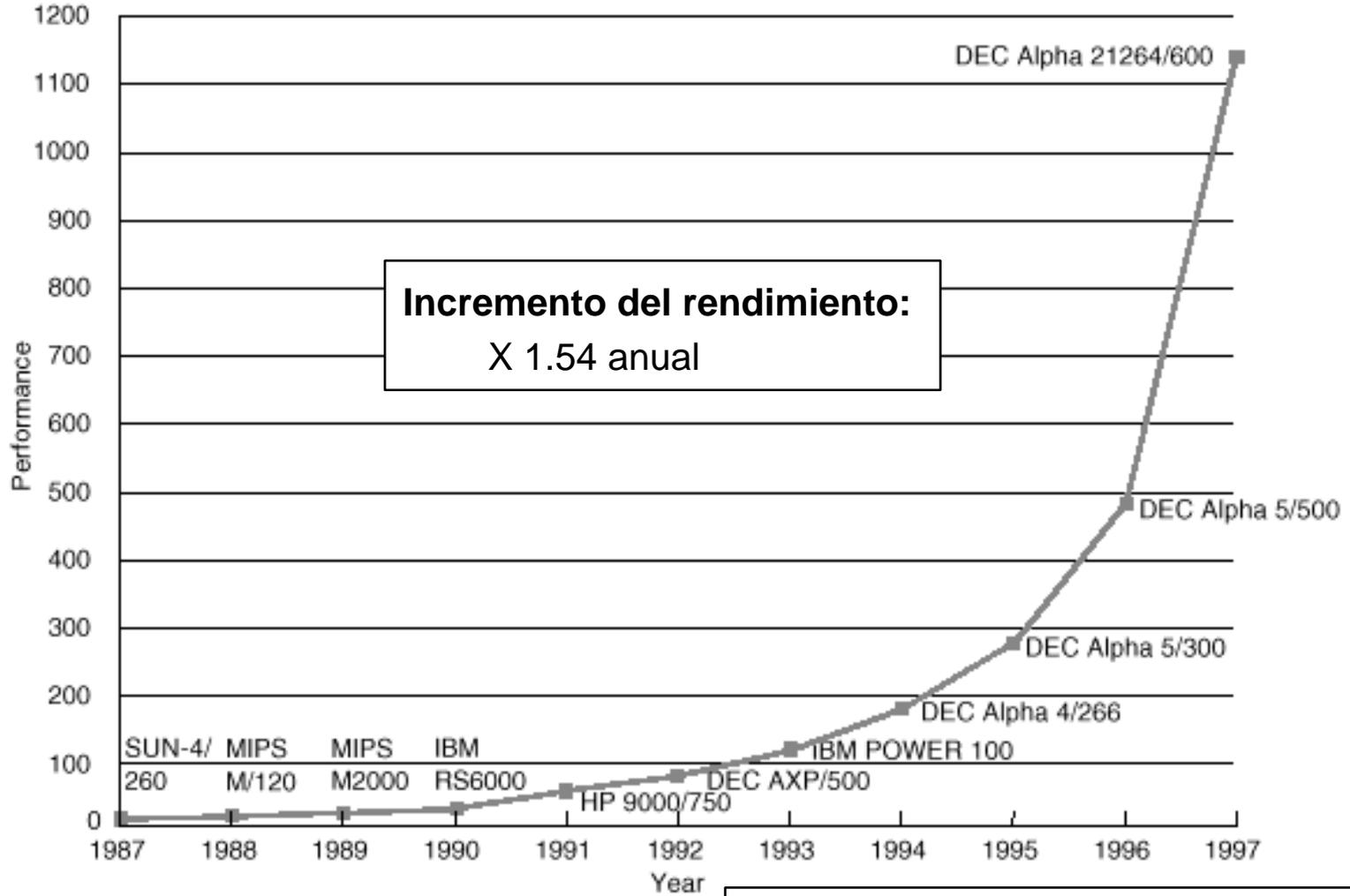
## 5ª Generación (1988 - ...)

### ⊗ Sistemas basados en microprocesador

- **Continúa crecimiento en la escala de integración:**
  - ⇒ Más de 15 millones de transistores por chip
- **Se abaratan los precios de los microprocesadores y aumentan sus prestaciones**
- **Aparece el procesador superescalar, capaz de lanzar varias instrucciones por ciclo de reloj**
- **La increíble relación coste-prestaciones del microprocesador hace de éste el elemento básico de:**
  - ⇒ **Computadores personales**
    - ⇒ PC (i486, Pentium, Pentium II, Pentium III, Pentium 4, Pentium M)
    - ⇒ Machintosh (Motorola 68030, 40, PowerPC4)
  - ⇒ **Estaciones de trabajo (Work-Stations)**
    - ⇒ Sun (SuperSparc, UltraSparc I, UltraSparc II, UltraSparc III)
    - ⇒ DEC (Alpha 21064, 21164, 21264, 21364)
    - ⇒ Silicon Graphics (MIPS R4000, R5000, R8000, R10000, R12000)
    - ⇒ IBM (RS/6000)
  - ⇒ **Supercomputadores paralelos basados en microprocesadores**
    - ⇒ Multicomputadores de de memoria distribuida (Cray T3E, IBM SP2)
    - ⇒ Multiprocesadores de memoria compartida (SG Origin 2000, Sun SparcServer, HP PA8000)

# 5. perspectiva histórica

## Evolución del rendimiento de los computadores



**Incremento del rendimiento:**  
X 1.54 anual

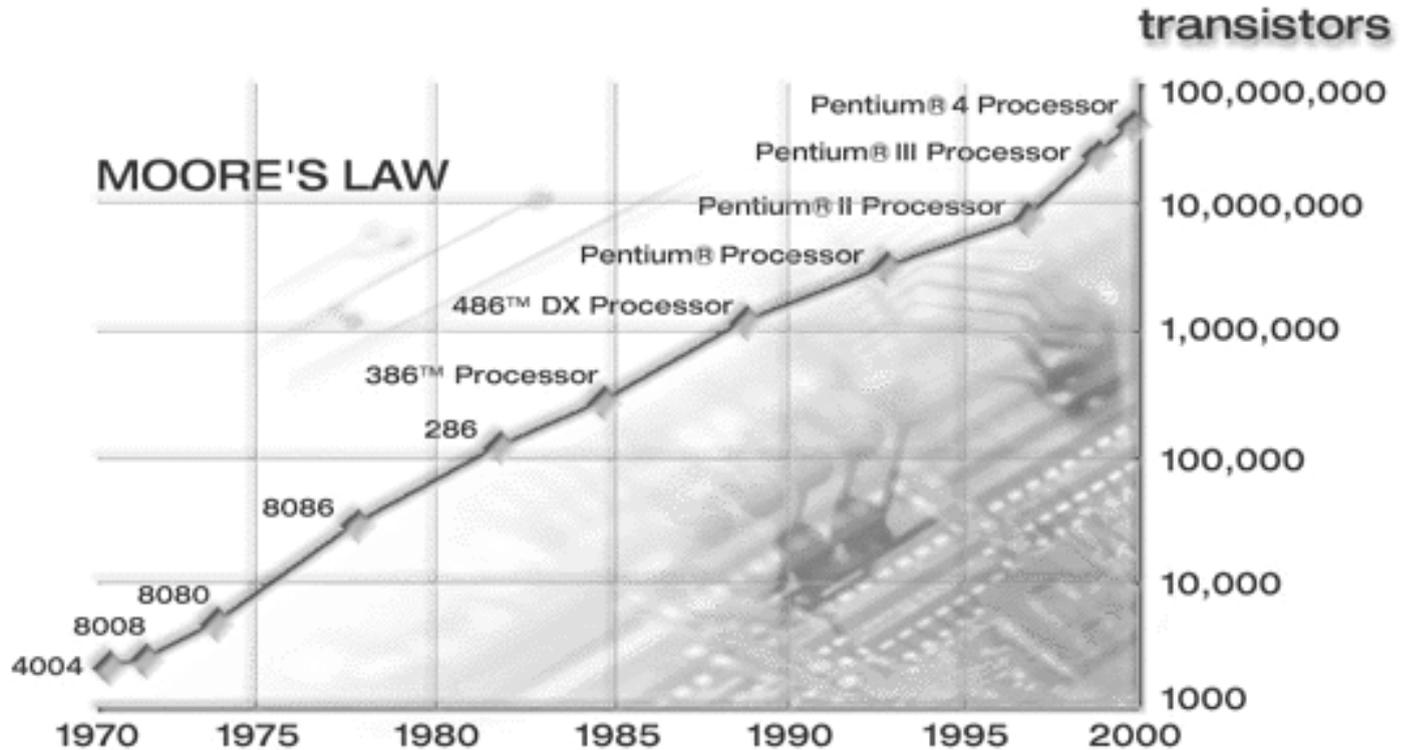
**Medida de rendimiento utilizada:**  
número de veces más rápido que el VAX-11/780

Evolution of Computer Performance

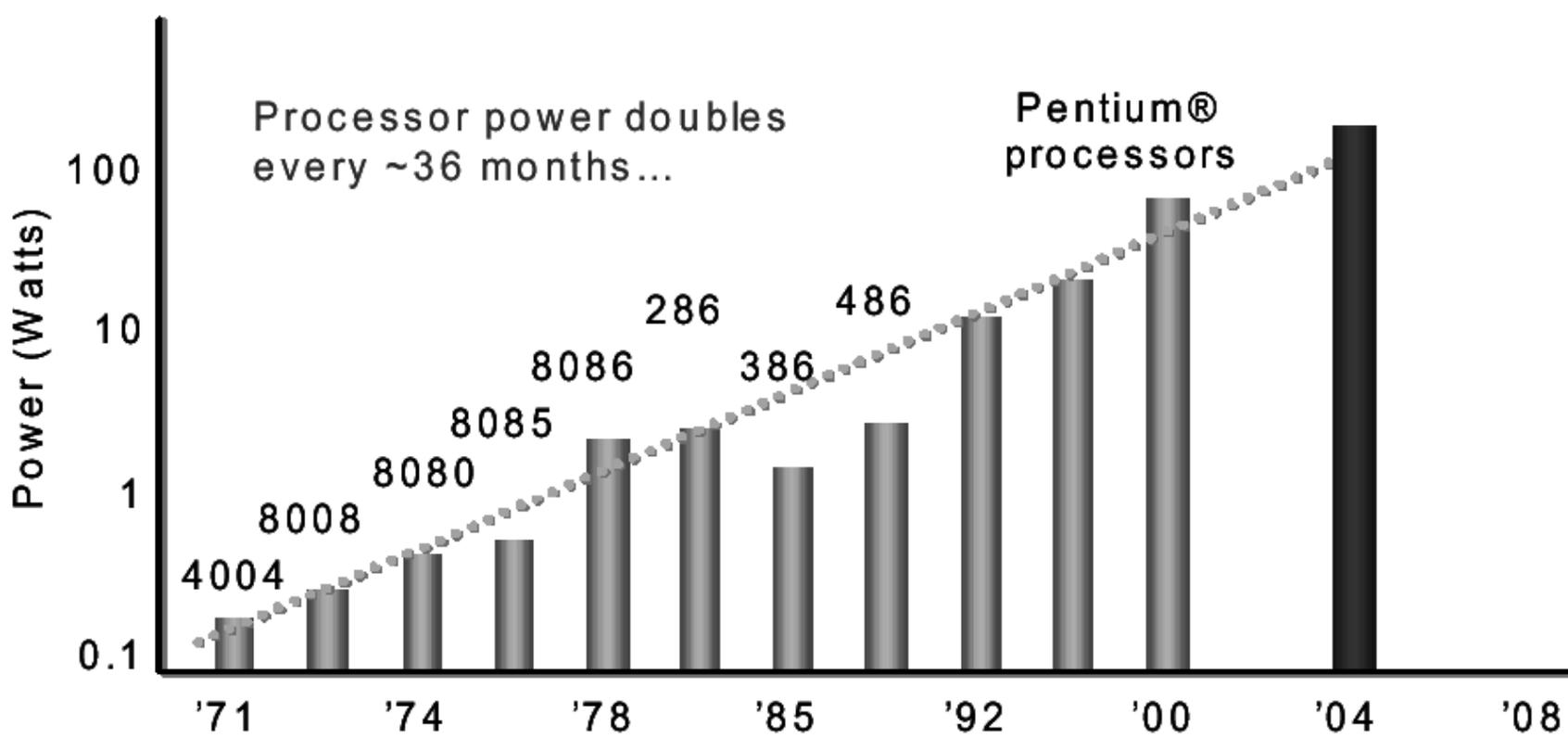
# Cray XP



# Ley de Moore Según Intel



# Consumo de Potencia



# Predicciones

	1997	1999	2001	2003	2006	2009	2012
<b>Tamaño característico (micras)</b>	<b>0.25</b>	<b>0.18</b>	<b>0.15</b>	<b>0.13</b>	<b>0.1</b>	<b>0.07</b>	<b>0.05</b>
<b>Voltaje de alimentación (V)</b>	<b>1.8-2.5</b>	<b>1.5-1.8</b>	<b>1.2-1.5</b>	<b>1.2-1.5</b>	<b>0.9-1.2</b>	<b>0.6-0.9</b>	<b>0.5-0.6</b>
<b>Transistores por chip (M)</b>	<b>11</b>	<b>21</b>	<b>40</b>	<b>76</b>	<b>200</b>	<b>520</b>	<b>1,400</b>
<b>Bits DRAM por chip (M)</b>	<b>167</b>	<b>1,070</b>	<b>1,700</b>	<b>4,290</b>	<b>17,200</b>	<b>68,700</b>	<b>275,000</b>
<b>Tamaño del dado (mm<sup>2</sup>)</b>	<b>300</b>	<b>340</b>	<b>385</b>	<b>430</b>	<b>520</b>	<b>620</b>	<b>750</b>
<b>Dimensión máxima del chip (mm)</b>	<b>17.3</b>	<b>18.4</b>	<b>19.6</b>	<b>20.7</b>	<b>22.8</b>	<b>24.9</b>	<b>27.4</b>
<b>Frecuencia de reloj local (MHz)</b>	<b>750</b>	<b>1,250</b>	<b>1,500</b>	<b>2,100</b>	<b>3,500</b>	<b>6,000</b>	<b>10,000</b>
<b>Frecuencia de reloj global (MHz)</b>	<b>750</b>	<b>1,200</b>	<b>1,400</b>	<b>1,600</b>	<b>2,000</b>	<b>2,500</b>	<b>3,000</b>
<b>Máxima pot por chip (W)</b>	<b>70</b>	<b>90</b>	<b>110</b>	<b>130</b>	<b>160</b>	<b>170</b>	<b>175</b>

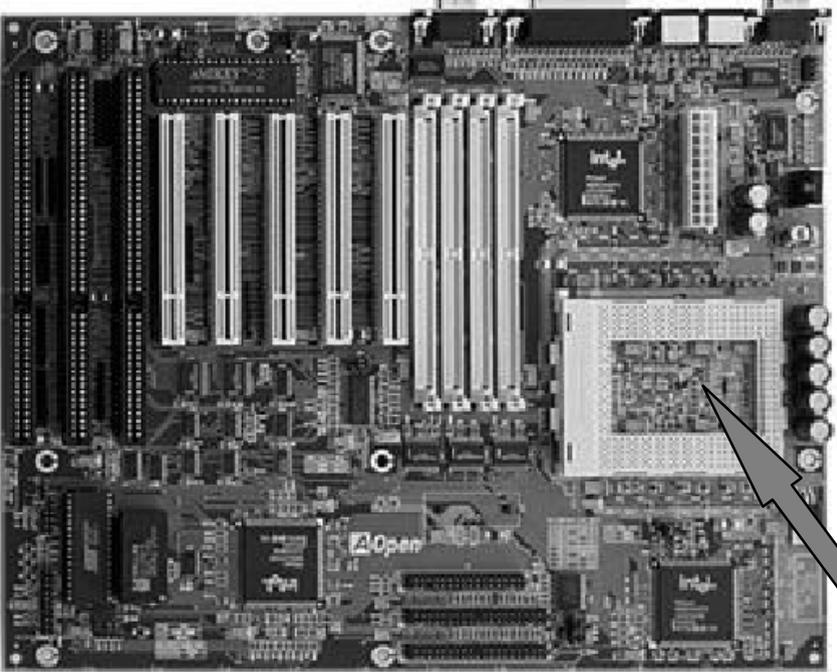
# 5. perspectiva histórica

## Evolución del rendimiento de los computadores (cont.)

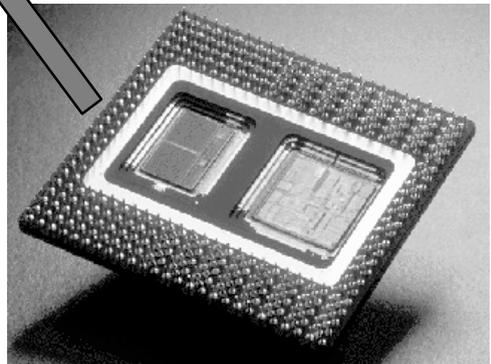
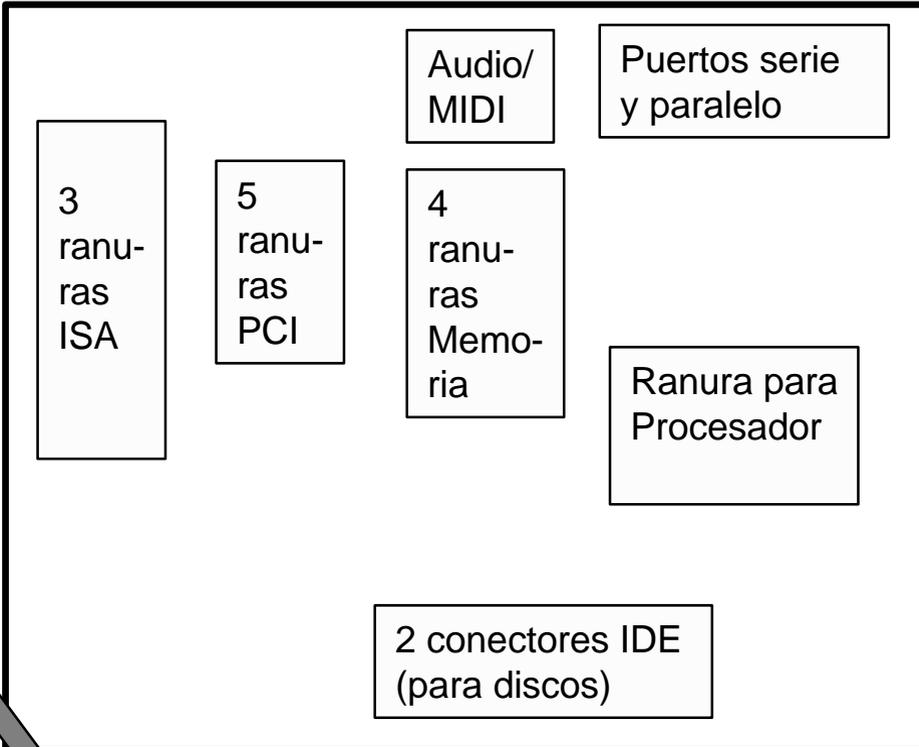
- ⊗ Incremento del rendimiento: **X 1.54 anual**
- ⊗ Incremento en la frecuencia del reloj: **X 1.25 anual**
- ⊗ **La diferencia entre el incremento en la frecuencia del reloj y el incremento real del rendimiento se debe a la introducción de mejoras en el diseño y la organización del propio computador**
  - **Mejoras en el procesador**
    - ⇒ Procesadores superescalares (lanzan varias instrucciones por ciclo)
    - ⇒ Ejecución fuera de orden y técnicas de ejecución especulativa (eliminan muchas de las dependencias impuestas por el programa)
  - **Mejoras en la memoria**
    - ⇒ Uso de jerarquía de memoria (memoria cache, memoria principal, memoria virtual)
  - **Mejoras en los elementos de interconexión**
    - ⇒ Uso de jerarquía de buses (buses del sistema, buses de expansión, buses externos)
  - **Mejoras en la gestión de la entrada/salida**
    - ⇒ Acceso directo a memoria (DMA), procesadores de E/S
  - **Mejoras en los dispositivos de entrada/salida**
    - ⇒ Dispositivos de almacenamiento de alta velocidad (discos, CD-ROM, etc.)
    - ⇒ Redes de alta velocidad

# 5. perspectiva histórica

## Dentro de un computador moderno



Acer AX64 Pentium Pro Motherboard

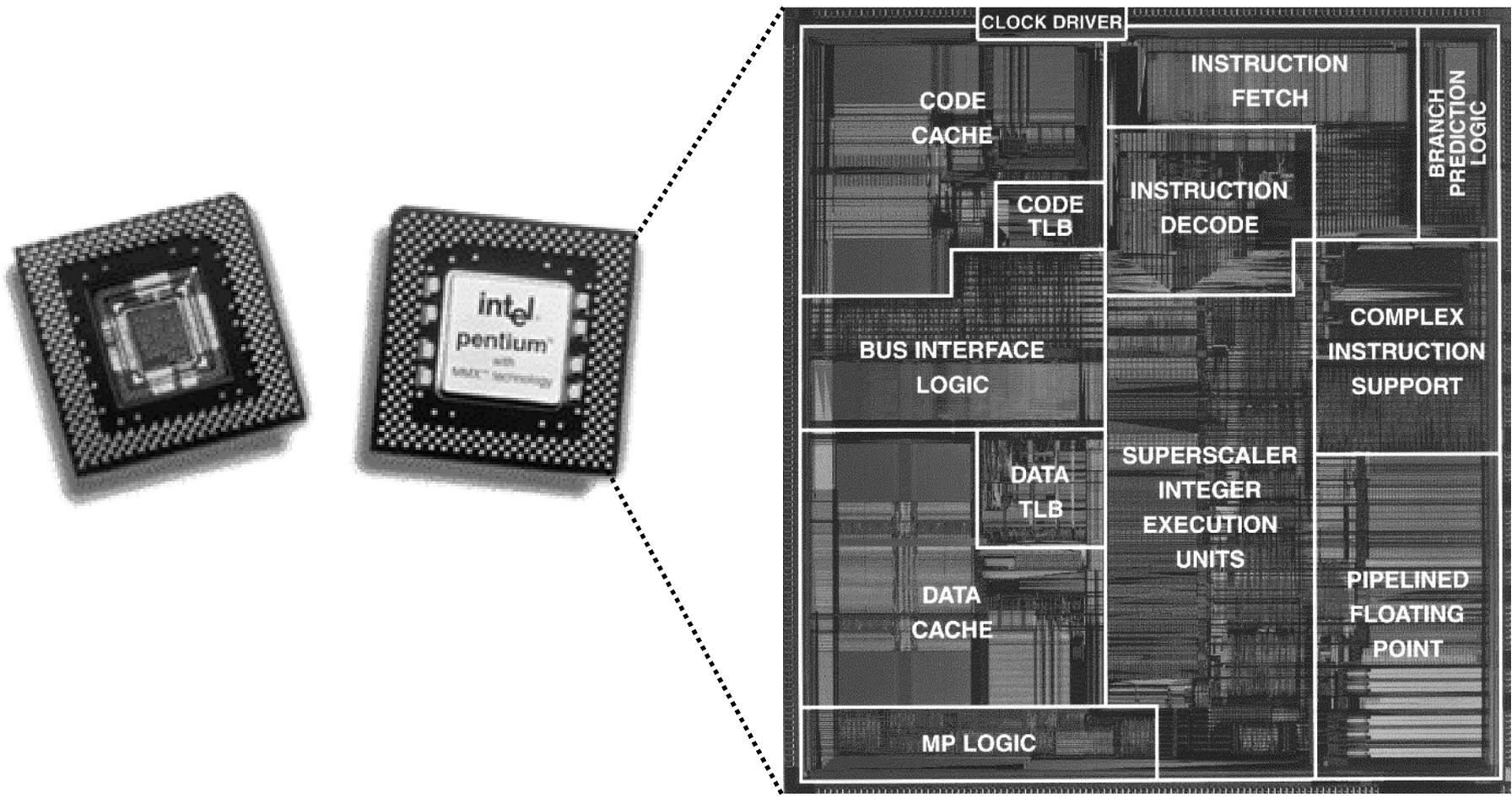


Procesador Pentium Pro

Entorno de Computación

# 5. perspectiva histórica

Dentro de un computador moderno: el procesador



ASN

Numero de Pagina Virtual <35>

Despl <13>

ASN	Prot	V	Tag	Way	Line	Datos

I CACHE

ASN	Prot	V	Tag	DF

D TLB

Mux 128:1

ASN	Prot	V	Tag	DF

I TLB

Mux 128:1

Valid <1>	Tag<29>	Datos
		<512>

Valid <1>	Tag<29>	Datos
		<512>

D CACHE

Etiqueta <38>	Datos<512>

I PREFETCH

Address

Mux 4:1

Address<38>	Datos<512>

VICTIM

Tag	Index	V	D	Tag	Datos
<1>		<1>	<1>	<21>	<512>

L2

System chip Memory

# Bibliografía

- ⊗ <http://www.top500.org/>
- ⊗ <http://www.intel.com>
  
- ⊗ **Bibliografía**
- ⊗ **Stallings Pág. 1 y 2**
- ⊗ **Patterson & Hennessy Pág. 1 y 2**