



TEMA 1

La Revolución del Ordenador

Ley de Moore: cada 2 años → x2 el nº transistores

Clases de ordenadores:

① Ordenador de mesa: uso general

capacidad

② Servidores: gran rendimiento
fiableidad

③ (Embedded) Ordenador Embutido: oculto como componente del sistema

Prestaciones de un ordenador:

④ Algoritmo → det. nº operaciones ejecutadas

⑤ Lenguaje de programación, compilador arquitectura

↳ det. nº instrucciones de máquina ejecutadas / ejecución

⑥ Procesador y sistema de memoria

↳ det. rapidez ejecución instrucciones

⑦ Sistemas E/S (incluyendo S.O.)

↳ det. como de rápido las instrucciones E/S son ejecutadas

Tras el Programa:

⑧ Software de Aplicación → alto nivel

⑨ Software de Sistema → compilador
S.O.

⑩ Hardware → procesador, memoria, controladores
E/S

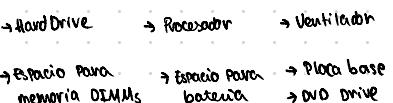
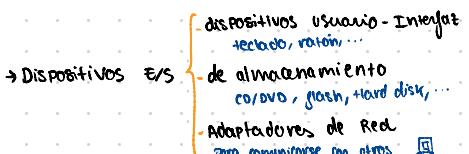
Niveles de Programación:

⑪ Alto nivel (Java)

⑫ Lenguaje Ensamblador → representación textual
de las instrucciones

⑬ Representación de hardware → bits

Componentes de un Ordenador



TIEMPO de la CPU

$$\text{Tiempo CPU} = \text{Ciclos de Reloj} \times \text{Ciclo de reloj}$$

$$T_{CPU} = \text{ciclos de reloj} \times \text{Ciclo de reloj}$$

Ejemplo: Ordenador A → f₀ = 2GHz

CPU = T_{CPU} = 10s

→ queremos convertirlo en B → Causa 12x ciclo de reloj

ordenador A

$$f_0 = 2\text{GHz} \rightarrow T_h = \frac{1}{f_0}$$

$$T_{CPU} = \text{Ciclos Reloj} \times T_h$$

$$N = \frac{10}{2 \times 10^{-9}} = 20 \times 10^9$$

ordenador B

$$T_{CPU} = \text{Ciclos } B \times T_h$$

$$12 \times N$$

$$6s = 12 \times (20 \times 10^9) \times T_h$$

$$\rightarrow T_h = \frac{1}{f_0 \times 12} \rightarrow f_0 = 4GHz$$

EJEMPLO CPI

Instruction	A	B	C
CPI for instr. type	1	2	(3)
IC in sequence 1	2	1	2
IC in sequence 2	(4)	1	1

Utiliza 4 instancias de A, 1 de B y 1 de C

Dentro del Procesador:

- ④ DataPath → operaciones con datos
- ④ Control
- ④ Memoria Cache → rápido acceso requerida, SRAM



Procesador → puede mejorar el → Respuesta-Rendimiento

Rendimiento Relativo

$$\text{Rendimiento} = \frac{1}{\text{Tiempo Ejecución}}$$

Tiempo Ejecución:

→ t transcurrido
recorriendo, tS, ...

→ t CPU → tardía en realizar una tarea en concreto

Reloj CPU → coordina las operaciones

Contador de Instrucciones (IC) y CPI

$$\text{Ciclos } B = IC \times CPI$$

"Mismo ISA" → Conjunto de Instrucciones [= IC]

Si los ≠ instrucciones → nº ≠ ciclos:

$$\text{Ciclos } B = \sum_{i=1}^n (\text{Instruction Count}_i \times CPI_i) \rightarrow IC_1 \times CPI_1 + IC_2 \times CPI_2 + \dots$$

Media de Ciclos:

$$\text{CPI} = \frac{\text{Ciclos } B}{\text{Instr. Count}} = \sum_{i=1}^n \left(CPI_i \times \frac{\text{Instruction Count}_i}{\text{Instruc. Count}} \right)$$

tarda 3 ciclos por instrucción

$$\text{seq 1} \rightarrow (2+4) + (1+2) + (2+1) = 10$$

$$\text{seq 2} \rightarrow (4+1) + (1+2) + 3 = 9 \rightarrow \text{mejor}$$

$$\left. \begin{array}{l} \text{CPI Media (seq1)} = \frac{10}{(2+4+1+2+1)} = 2 \\ \text{CPI Media (seq2)} = \frac{9}{(4+1+2)} = 1.5 \end{array} \right\}$$

TEMA 1

Resumen Prestaciones

$$Tepu = \frac{\text{Instrucciones}}{\text{Programa}} \times \frac{\text{Ciclos @}}{\text{Instrucciones}} \times \frac{\text{segundos}}{\text{Ciclos @}}$$

- Algoritmo → afecta IC, visible CPI
- Lenguaje Program → IC, CPI
- Compilador → IC, CPI
- ISA (conjunto Instrucciones) → IC, CPI, Tp, To

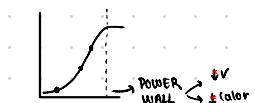
Tendencia de consumo

en tecnología CMOS IC:

$$\text{Consumo} = C \times V^2 \times f$$

capacidad voltaje

versión	tensión	frec.
1	1.4V	3GHz
2	0.8V	4GHz



- La Potencia dinámica se ha reducido un 10%
 - ¿cuanto es la carga capacitiva?
- $$C_{eq} = V \cdot f$$

$$P_B \rightarrow \text{nueva}$$

$$P_A \rightarrow \text{antigua} \rightarrow P_A = 0.9 \cdot P_B$$

$$\hookrightarrow \frac{P_A}{P_B} = \frac{C_1 \cdot (1.4V)^2 \cdot 3GHz}{C_2 \cdot (0.8V)^2 \cdot 4GHz} = 0.9 \rightarrow C_2 = 1.2 C_1$$

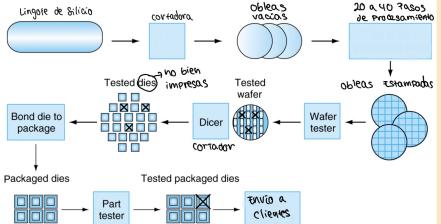
Multiprocesadores

$\Delta f = \Delta \text{consumo} + \Delta T \rightarrow$ transistores no funcionan bien
→ aumentamos los núcleos (cores): tenemos varios procesadores en //

Procesadores varios núcleos (multicore)

Programar las prestaciones difícil
equivalente de carga optimización < optimización < sincronización

Fabricación IC's



Rendimiento (yield): proporción de dies por obtura

SPEC CPU Benchmark

"seuelas"

- Programa utilizado para Prestaciones
- Standard Performance Evaluation Corp (SPEC)
mejora seuelas o puntos de ref para CPU, E/S, web...

para algunos Benchmark (puntos de referencia), core i3:

- al 100% → Carga 295 W → consumo & no está trabajando es alto
- al 50% → Carga 246 W (85%)
- al 10% → Carga 180 W (61%)

Solución: añadir cores mientras no trabaja

MIPS : Million of Instruction Per Second

No considera #s en ISA
y/o en complejidad de instrucciones.
La CPU varía entre programas de la CPU

$$\text{MIPS} = \frac{\text{Instrucción ejecutada}}{\text{CPI}}$$

$$\text{MIPS} = \frac{\text{frecuencia}}{\text{CPI} \times 10^6}$$

Ejemplo:

operaciones @ clases	ciclos/clases para c1	ciclos/clases para c2	frecuencia (en la q aparece)
# A	1	2	60%
+ B	2	3	30%
- C	4	4	10%

¿Cuál de los dos ordenadores tiene mejor rendimiento?

$$\text{CPI}_1 = 1 \cdot 0.6 + 2 \cdot 0.3 + 4 \cdot 0.1 = 1.6$$

$$\text{CPI}_2 = 2 \cdot 0.6 + 3 \cdot 0.3 + 4 \cdot 0.1 = 2.5$$

$$\text{MIPS}_1 = \frac{80 \times 10^6}{1.6 \times 10^6} = 50$$

$$\text{MIPS}_2 = \frac{100 \times 10^6}{2.5 \times 10^6} = 40$$

PEORES PRESTACIONES

Ejemplo (2):

- queremos x 1.4 ciclos de reloj
- Opciones
- *: mejorar factor 8
 - +: mejorar factor 3 (x 3 más rápido)

② CPU nuevo = Tcpu n = Tcpu a + 1/4

N ciclos mijo + 1/4 ciclos
no seuelas → cambia

Probaremos con multiplicación:

$$\text{Tcpu antiguo} = Tcpu \left(\frac{0.5T}{8} + \frac{0.2T}{8} + \frac{0.8T}{8} \right) \rightarrow Tcpu \text{ nuevo} = 0.5T + 0.2T + 0.8T$$

$$\frac{\text{Tcpu nuevo}}{\text{Tcpu antiguo}} \rightarrow \frac{1.4}{1} \Rightarrow \frac{\text{Tcpu a}}{\text{Tcpu n}} = \frac{0.5T + 0.2T + 0.8T}{0.5T + 0.2T + 0.8T} = 1/16 \quad \text{luego de !!}$$

→ Puedes hacer q el
Programa sea más de 100 veces más rápido con
solo una mejora?