

Los chinos van a dominar el mundo.

El primo informático va a dominar el mundo.

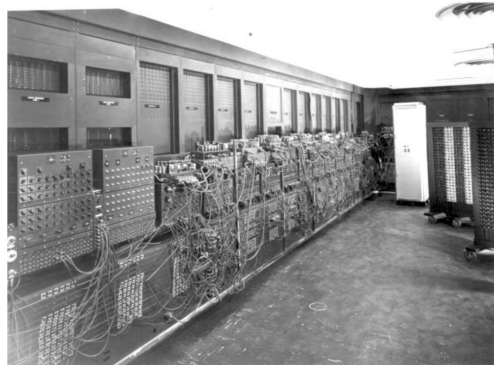
Mediciones

Nombre	Símbolo	2^x	10^x	Error
Kilo	K	$2^{10} \rightarrow 1024$	$10^3 \rightarrow 1000$	2,4%

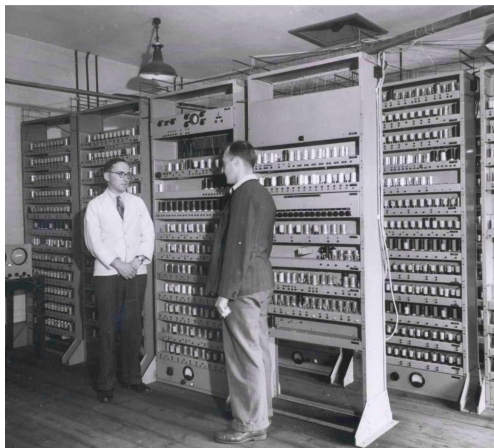
Depende de dónde esté un KB son 1000 (DISCO) o 1024 (RAM). Los vendedores no te engañan, depende de lo que compres es una medida u otra.

Von Neumann

La memoria se accede por dirección. Los programas están cargados en memoria. (...)



"Debuga esto. Mal rollo."



"Cada bombilla consume un huevo... Y un huevo por mucho... Mal rollo."

Evaluación de un sistema

Coste

El coste viene dado por:

- Tamaño del dado (el centro de un chip)
- La complejidad del dado (cuantas más capas y más transistores más caro)
- Coste ambiental y social

Rendimiento

1 / tiempo que tarda en fabricarse

Viene dado por:

- Jerarquía de memoria
- Paralelismo
- Segmentación

Consumo

Energía consumida en Watts .

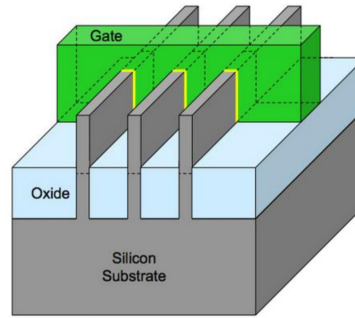
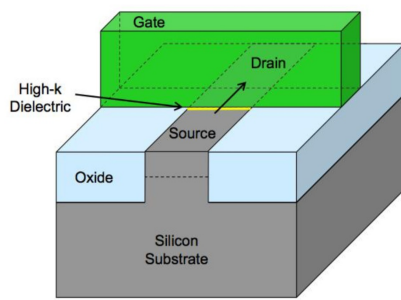
Fiabilidad

Tiempo entre fallos / reparaciones

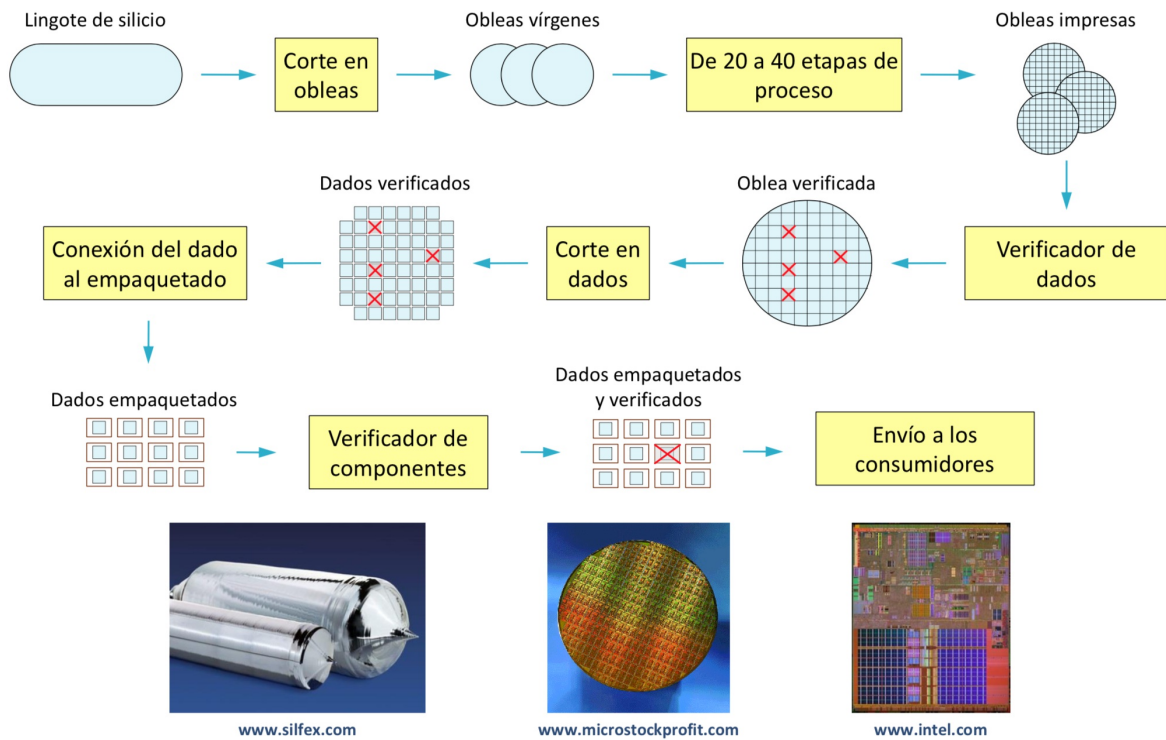
Tecnología de fabricación

Nuestros sistemas están basados en CMOS.

Traditional Planar Transistor 22 nm Tri-Gate Transistor



Proceso de fabricación de un chip



Ecuaciones

■ Coste de un circuito integrado

$$\text{Coste de un circuito integrado} = \frac{\text{Coste del die} + \text{Coste de testeo} + \text{Coste de empaquetado y test final}}{\text{Yield final (test)}}$$

■ Coste del *die* (dado)

$$\text{Coste del die} = \frac{\text{Coste del wafer}}{\text{Dies per wafer} \times \text{Die yield}}$$

■ Dies per wafer (oblea)

$$\text{Dies per wafer} = \frac{\text{Area útil}}{\text{Die area}} = \frac{\pi \times (\text{diametro}/2)^2}{\text{Die area}} - \frac{\pi \times \text{diameter}}{\sqrt{2} \times \text{Die area}}$$

Waffer area / die area

Compensación por los "dies" incompletos de los bordes

■ Die yield

$$\text{Die yield} = \text{Wafer yield} \times \left(1 + \frac{\text{defectos por unidad de area} \times \text{die area}}{\alpha} \right)^{-\alpha}$$

- α = medida de la complejidad, se aproxima al número de máscaras críticas



Ley de Moore

La razón de que no se cumpla la ley de Moore actualmente es que no sale rentable económicamente.

Latencia vs Ancho de banda

Latencia

Ping --> 10ms

Ferrari --> 3600s

Camión --> 10.800s

Ancho de banda

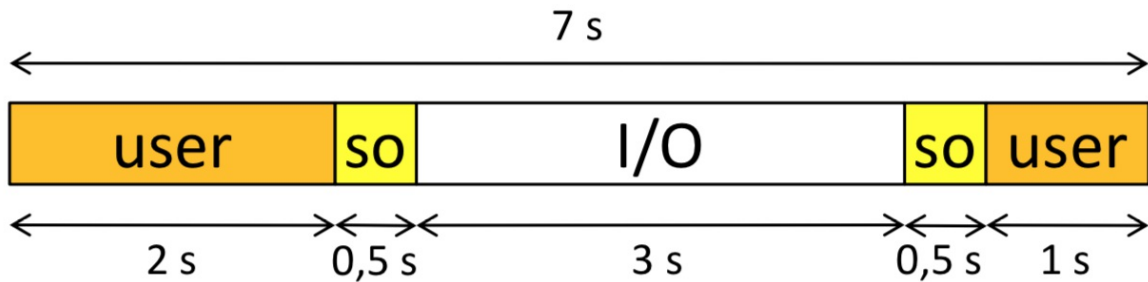
ADSL 20Mb/s --> 2,5MB/s

Ferrari --> 277,8 MB/s

Camión --> 3,15 TB/s

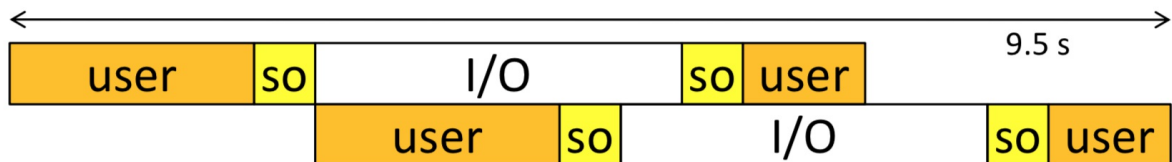
Segmentación

■ Tiempo de respuesta (wall time)



Yo puedo empezar un proceso siempre que el SO esté libre para manejarlo.

■ Productividad (throughput) = trabajo/tiempo



Tiempo de ejecución (rendimiento)

$$\frac{1}{\text{Rendimiento}} = \text{Tiempo de ejecución} = N \times \text{CPI} \times T_c$$

Número de instrucciones ejecutadas (points to N)
Tiempo de ciclo (points to T_c)

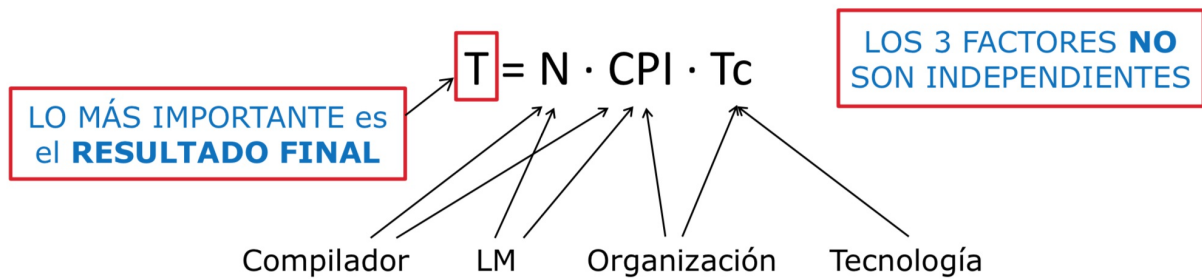
↑Tiempo ejecución \Rightarrow ↓Rendimiento

Número medio de ciclos por instrucción (points to CPI)

$$\frac{1}{\text{Rendimiento}} = \frac{\text{tiempo}}{\text{Programa}} = \frac{\text{instrucciones}}{\text{Programa}} \times \frac{\text{ciclos}}{\text{instrucción}} \times \frac{\text{tiempo}}{\text{ciclo}}$$

Rendimiento

- Reducir el tiempo de ejecución \Rightarrow Se puede actuar en cualquiera de los 3 factores



Otras métricas de Rendimiento

- MIPS: Millones de **instrucciones** por segundo
- MFLOPS: Millones de **operaciones en punto flotante** por segundo

Comparación de rendimiento

$$\text{Ganancia (Speedup)} = \frac{T_A}{T_B} \quad \begin{array}{l} > 1 \Rightarrow \text{B es más rápido que A} \\ < 1 \Rightarrow \text{B es más lento que A} \end{array}$$

Si un programa P tarda 4,5 segundos en el computador A y 2 segundos en el computador B:

$$\text{Ganancia} = \frac{T_A}{T_B} = \frac{4,5}{2} = 2,25 \Rightarrow \text{B es 2,25 veces más rápido que A, usaremos 2,25x}$$

También podemos usar porcentajes:

$$\left(\frac{T_A}{T_B} - 1 \right) \cdot 100$$

- ✓ Una ganancia del 125% \Rightarrow B es 2,25 veces más rápido.

Ley de Amdahl

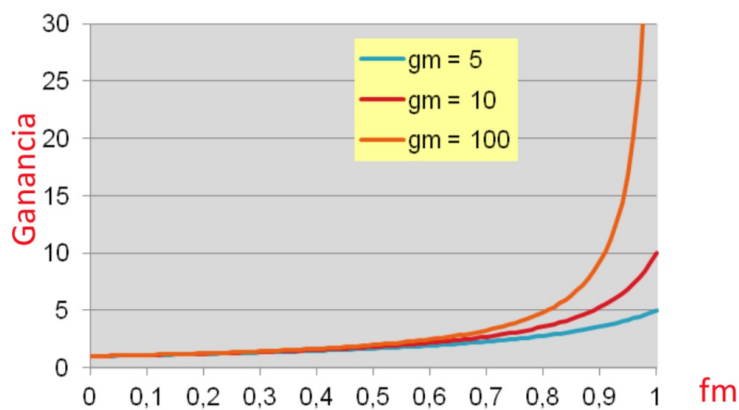
Leer capítulo 1 de H&P

Si yo solo toco una subrutina y la mejoro y todo lo demás lo dejo igual. La ganancia sería

cuanto se reduce el tiempo de la subrutina.

Si el porcentaje de lo que tarda no es grande. No vale la pena mejorarlo. Es decir, céntrate en aquello que tarda más.

Para que la ganancia obtenida sea significativa (cercana a gm) fm ha de ser prácticamente 1.



¿Cuál es la parte más rentable de mejorar? Aquella que ocupe la mayoría de tiempo.

$$\text{Ganancia} = \frac{1}{1 - fm}$$

Consumo

La energía se mide en **Joules**

La potencia se mide en **Joules/seg => Watts**

Si la potencia es constante:

$$\text{Energía} = \text{Potencia} * t$$

La potencia consumida por un CMOS depende de:

- Conmutación (cambiar de estado 1 a 0, o viceversa)
- Corriente de fugas (disipación al no ser ideales)
- Corriente de cortocircuito (...?!)

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

$$\text{Potencia}_{\text{de fuga}} = I_{\text{de fuga}} \times V$$

$$\text{Eficiencia energética} = \frac{\text{rendimiento}}{\text{watio}} = \frac{1}{\text{tiempo} \times \text{watio}} = \frac{1}{\text{Energía consumida}}$$

Fiabilidad

Fiabilidad es el tiempo de funcionamiento continuo sin fallos. Para analizarlo tenemos que tener una:

Tasa de fallos

$$\text{Tasa de fallos} = 1 / \text{tiempo medio de fallo}$$

No es difícil no? Ahora en inglés.

MTTF --> Mean Time To Failure

MTTR --> Mean Time To Repair

MTBF --> Mean Time Between Failures

$$\text{MTBF} = \text{MTTF} + \text{MTTR}$$

Disponibilidad

Disponibilidad = Tiempo medio de fallo / (Tiempo medio de fallo + Tiempo medio de R

Estadística

Este tiempo de fallo, se distribuye exponencialmente (PE be like). Por lo tanto, tenemos que:

$$p = 1 - e^{-\lambda t}$$

Calcular el MTTF de un sistema

■ Ejercicio

Sistema formado por:

- 1 CPU (incluye placa base y memoria) MTTF = 1.000.000 horas
- 2 Discos MTTF = 500.000 horas
- 1 Fuente de alimentación MTTF = 200.000 horas

Calcular el MTTF del sistema:

$$\frac{1}{\text{MTTF}_{\text{sistema}}} = \frac{1}{\text{MTTF}_{\text{CPU}}} + \frac{1}{\text{MTTF}_{\text{disco}}} + \frac{1}{\text{MTTF}_{\text{disco}}} + \frac{1}{\text{MTTF}_{\text{fuente}}} = \frac{1}{10^6} + \frac{2}{500 \cdot 10^3} + \frac{1}{200 \cdot 10^3} = \frac{1+4+5}{10^6} = \frac{1}{10^5}$$

$$\text{MTTF}_{\text{sistema}} = 100.000 \text{ horas}$$

Redundancia

Para mejorar el MTTF de un sistema se puede hacer de dos maneras.

1. Vuelvo a calcular el resultado para ver si es correcto (**gastas más tiempo**).
2. Disponer de elementos extra para reemplazar al que falla (**gastas en recursos**).

Ejemplo

- Construimos una fuente de alimentación redundante con dos fuentes de tal forma que una es suficiente para alimentar el sistema. Cuando una de las dos falla se reemplaza sin detener el sistema.
- $MTTF_{fuente}$ = tiempo medio entre fallos para 1 fuente (200.000 horas en nuestro ejemplo)
- $MTTR_{fuente}$ = tiempo de cambiar la fuente que falla (supongamos 24 horas)
- $MTTF_{2\text{ fuentes}} = MTTF_{fuente}/2$ tiempo medio entre fallos para 2 fuentes.
- Probabilidad P_2 de que falle la segunda fuente (Una vez ya ha fallado la 1a):

$$P_2 = 1 - e^{-\lambda t} = 1 - e^{-\frac{MTTR_{fuente}}{MTTF_{fuente}}}$$

- MTTR son unas pocas horas y MTTF pueden ser millones \Rightarrow (MTTR/MTTF es un valor cercano a 0)

$$\text{si } x \rightarrow 0 \text{ entonces } 1 - e^{-x} \approx x \text{ por lo que } P_2 \approx \frac{MTTR_{fuente}}{MTTF_{fuente}}$$

- $1/p$ es el numero de veces esperado que hay que repetir un proceso (falla una fuente) con probabilidad p hasta que hay éxito (falla la 2ª fuente en este caso)
- Cada vez que falla una fuente han transcurrido en media $MTTF_{2\text{ fuentes}}$ horas, por tanto:

$$MTTF_{fuente\ doble} = MTTF_{2\text{ fuentes}} \times \frac{MTTF_{fuente}}{MTTR_{fuente}} = \frac{MTTF_{fuente}^2}{2 \times MTTR_{fuente}} = \frac{200.000^2}{2 \times 24} =$$

830.000.000 horas