

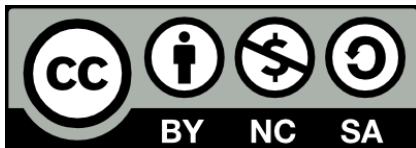
# Tema 1:

## Fundamentos de diseño y evaluación de computadores

Departament d'Arquitectura de Computadors

Facultat d'Informàtica de Barcelona

Universitat Politècnica de Catalunya





**“I think there’s a world market for about 5 computers.”**

***(Thomas J. Watson, Chairman of the Board, IBM, circa 1943)***

**“In the future, computers may weigh no more than 1.5 tonnes.”**

***– Popular mechanics, 1949***

**“There is no reason for any individual to have a computer in his home.”**

***(Ken Olson, President, Digital Equipment Corporation, 1977)***

- **Introducción**
- **Coste**
- **Rendimiento**
- **Consumo**
- **Fiabilidad**

# Evolución de los computadores

	CRAY 1	Lenovo T61	iPhone X
<b>Año (instalación)</b>	1976	2008	2017
<b>CPU</b>	Custom (circuitos discretos)	Intel Core 2 Duo T7300 (Merom)	Apple A11
<b>Características</b>	Procesador vectorial Sin Cache MP SRAM	2 cores IA32 segmentado L1I (2×32KB) & L1D(2×32KB) + L2 (4MB) MMX, SSE(1,2,3,3S),EM64T,VT-x	6 Custom ARMv8-A cores L1I(6×32KB) & L1D (6×32KB) + L2 (8MB)
<b>Consumo</b>	115.000 W	100 W (CPU 35W)	< 5 W
<b>Dimensiones</b>	Ø: 262,89 cm – 143,51 Alt: 195,58 – 48,26	3,0 × 34,0 × 24,0 cm	0,77 x 14,4 x 7,1 cm
<b>Peso</b>	5.500 Kg	2,5 Kg	0,174 Kg
<b>Coste</b>	8,86 millones dólares (IPC acumulado España 910,7%)	1.500€	1300€
<b>Memoria</b>	8 MB	2 GB	3 GB
<b>Disco</b>	2,5 GB (1 millón \$, 1976)	120 GB (50€, abril 2010)	256 GB
<b>Rendimiento</b>	160 MFLOPS (pico), 50 MFLOPS sostenido	10 GFLOPS (linpack)	120 GFLOPS (linpack)
<b>Frecuencia</b>	80 MHz	2 GHz	2,4GHz
	Refrigerado con freón	Ventilador	Pasiva



¡32 años!



¡9 años!

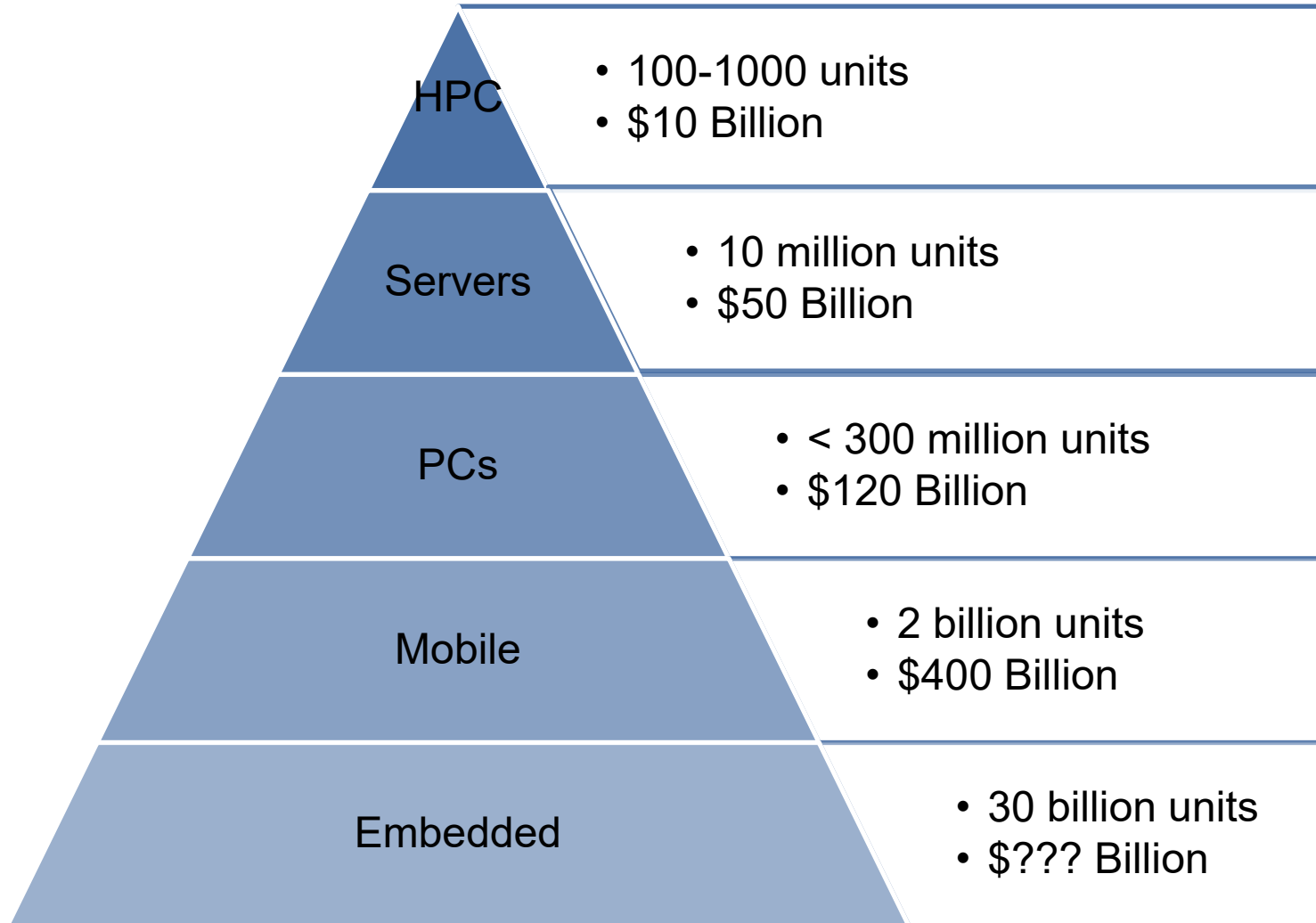


# Tipos de computadores

	Empotrados	Móviles	Sobremesa	Servidor	Supercomputador
<b>Coste del sistema</b>	1-100.000€	100-1000€	300-2500€	10K-1M€	1M-????€
<b>Precio CPU</b>	0,01-100€	10-100€	50-500€	200-10.000€	200-2.000€
<b>Puntos críticos en el diseño del sistema</b>	Precio, consumo, rendimiento en aplicaciones específicas	Precio, consumo, rendimiento en aplicaciones específicas	Coste/rendimiento, rendimiento gráficos	Throughput, disponibilidad, escalabilidad	Rendimiento en coma flotante, capacidad almacenamiento
<b>Aplicaciones</b>	Automóvil, Electrodomésticos, Lectores Blue Ray, Wearables	Teléfonos, Tablet, PDAs, ...	Portátiles, Estaciones de trabajo, Desarrollo software, Ofimática, Ocio,...	Servidor web, Bases de Datos, ...	Geofísica, Meteorología, Diseño de aviones, ...
<b>#cores</b>	1-2	1-8	2-32	8-10.000	Hasta 10.649.600 <sup>(1)</sup>
<b>Memoria</b>	Mbytes	Gbytes	Gbytes	Tbytes	Pbytes
<b>Disco</b>	Gbytes	Gbytes	Tbytes	Pbytes	Pbytes

(3) Top500 Jun/2019

# Tipos de computadores



Diseño	Program	Mant / gestión
CE EE TI	CE CS Físicos Biólogos .....	TI
CE EE TI	SE CS CE	TI
CE EE	SE CS ...	FP Primo informat ...
CE EE	SE CS CE ....	
CE EE	CE SE EE	

- Hay un cierto desconcierto a la hora de utilizar los prefijos de Medida

Nombre	Símbolo	$2^x$		$10^x$		error
Kilo	K	$2^{10}$	1024	$10^3$	1000	2,4%
Mega	M	$2^{20}$	1048576	$10^6$	1000000	4,9%
Giga	G	$2^{30}$	1073741824	$10^9$	1000000000	7,4%
Tera	T	$2^{40}$	1099511627776	$10^{12}$	1000000000000	10,0%
Peta	P	$2^{50}$	1125899906842624	$10^{15}$	1000000000000000	12,6%
Exa	E	$2^{60}$	1152921504606846976	$10^{18}$	1000000000000000000	15,3%
Zetta	Z	$2^{70}$	1180591620717411303424	$10^{21}$	1000000000000000000000	18,1%
Yotta	Y	$2^{80}$	1208925819614629174706176	$10^{24}$	1000000000000000000000000	20,1%
Xenta/Xora/Bronto		$2^{90}$	125728285239921434169442304	$10^{27}$	1000000000000000000000000000	25,7%

- Para los prefijos binarios existe la **norma ISO/CEI**, aunque no está suficientemente extendida. Por ejemplo,  $2^{20}$  se denomina **Mebi** y usa el símbolo **Mi**.



# Problema con las medidas

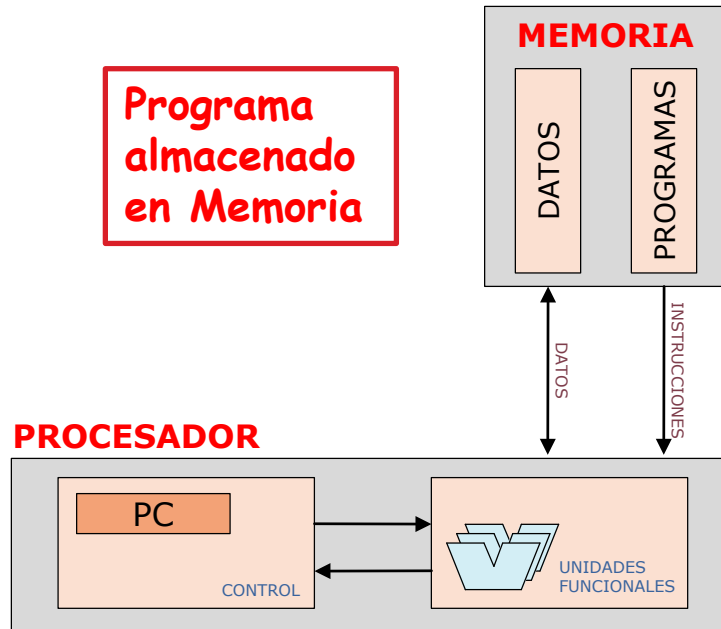
**No todo el mundo quiere decir lo mismo cuando utiliza la misma palabra:**

- Los Hercios (Hz) se miden en potencias de 10:  
un procesador a 1 GigaHercio (GHz) va a 1.000.000.000 Hz.
- La velocidad de transmisión se mide en potencias de 10:
  - un MP3 stream a 128 Kb/s transmite 128.000 bits por segundo,
  - una conexión ADSL de 12 Mb/s acepta un máximo de 12.000.000 bits por segundo.
- El ancho de banda de los buses también se mide en potencias de 10
- La Memoria RAM siempre se mide en potencias de 2: 1GB de RAM es  $2^{30}$  bytes de RAM.
- Los discos duros (HD) utilizan potencias de 10.
  - Un HD de 30GB tiene  $30 \cdot 10^9$  bytes (aproximadamente  $28 \cdot 2^{30}$ ).
  - No es marketing, sino tradición:  
la estructura física de los discos (platos, pistas, sectores) no tiene por qué ser pot. de 2.
  - Además, el SO suele indicar el tamaño del disco en potencias de 2.

**Por tanto, si compramos un portátil con 1GB de RAM y 30 GB de HD,  
Windows nos dirá que tiene 1GB de RAM y 28GB de disco duro.**



# Máquina von Neumann



**John von Neumann** (matemático húngaro) "First Draft of a report on the EDVAC" contract n. W-670-DRD-492 Moore School of Electrical Engineering, University of Pennsylvania, Philadelphia, **June 1945**

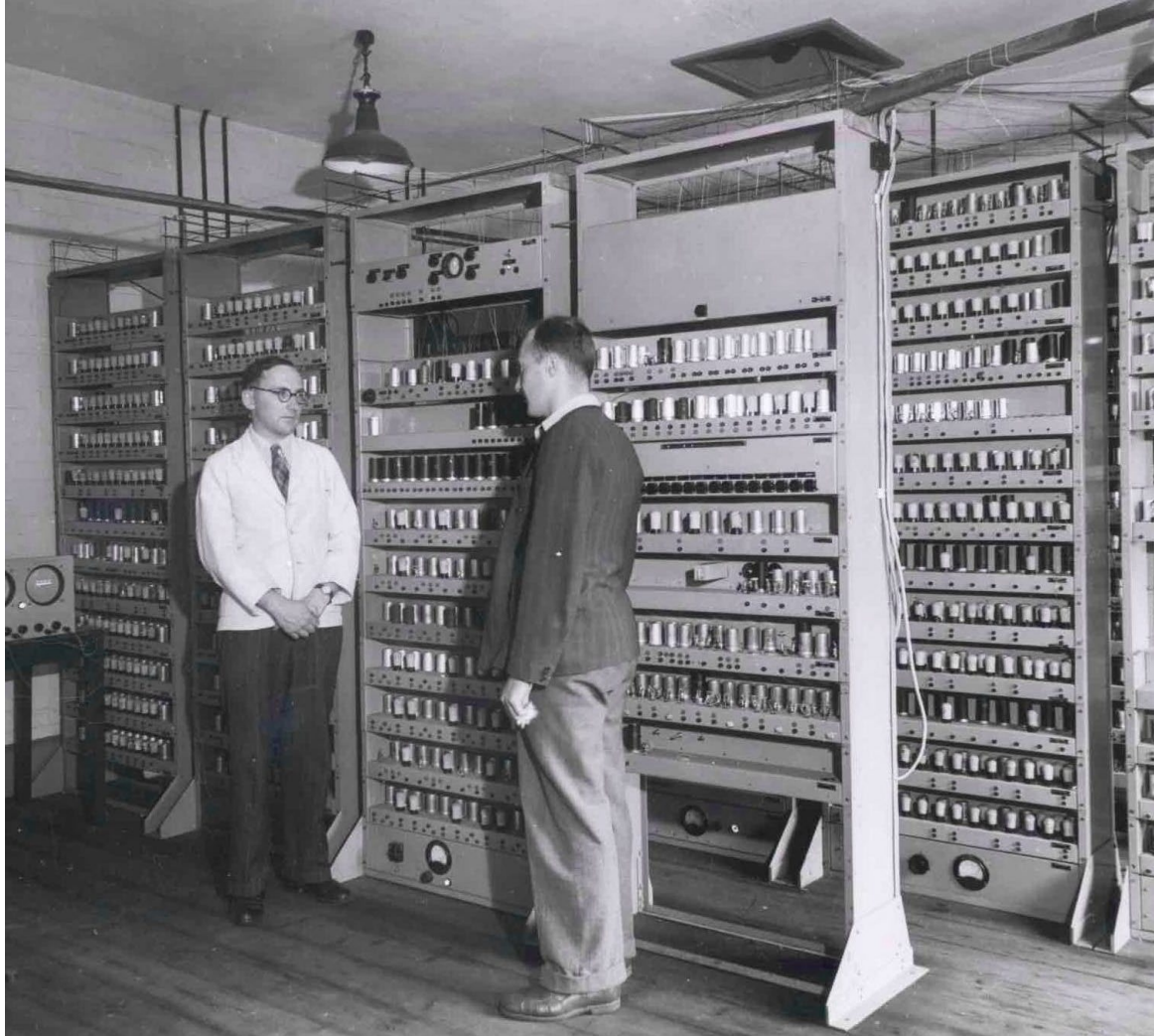
Sólo aparece un autor aunque el report es resultado de múltiples horas de discusión del grupo que diseñó el ENIAC, en el que von Neumann era sólo un visitante. Prosper Eckert y John Mauchly (diseñadores principales del ENIAC) abandonaron la Moore School por este hecho.

- Memoria accesible por dirección.
- Las posiciones de memoria se pueden leer/escribir las veces que sean necesarias.
- Tanto los datos como las instrucciones se almacenan en memoria.
- No existe ninguna señal para diferenciar en memoria datos de instrucciones.
- Las instrucciones se ejecutan en secuencia.
- Existe un registro (**PC**) que apunta siempre a la instrucción en ejecución.
- Existen instrucciones explícitas para romper el secuenciamiento.
- Las instrucciones son **imperativas**, especifican **cómo** obtener los operandos, **qué** operación hay que realizar y **dónde** dejar el resultado.



**1946: *Preliminary Discussion of the Logical Design of an Electronic Computing Instrument*, Arthur W. Burks, Herman H. Goldstine y John von Neumann**

# EDSAC 1, 1949



**Maurice Wilkes:**

**1<sup>er</sup> computador de  
programa almacenado.**

**650 instrucciones/s**



# Evaluación de un sistema informático: Métricas

## ■ Coste:

- Tamaño del *die* (dado).
- Complejidad:  
esfuerzo requerido en el diseño, validación y fabricación del procesador.
- Coste ambiental y social.

## ■ Rendimiento: Inversa del tiempo que tarda en completarse una tarea. Formas básicas de mejorar el rendimiento:

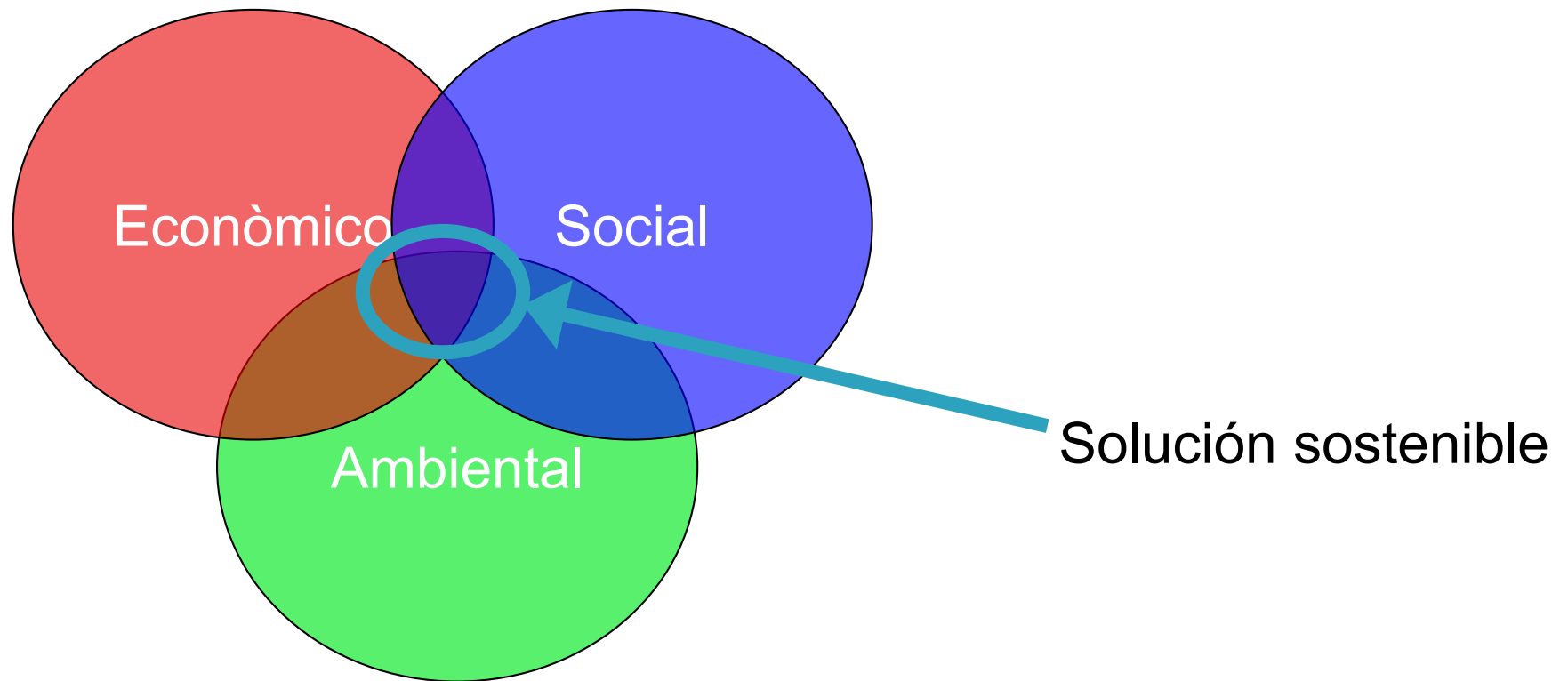
- Jerarquía de Memoria
- Paralelismo
- Segmentación

## ■ Consumo: Energía consumida por unidad de tiempo (vatios). Normalmente, mayor rendimiento requiere mayor consumo.

## ■ Fiabilidad: Tiempo entre fallos/reparaciones. Sistemas tolerantes a fallos.

# ¿Qué es la sostenibilidad?

## Competencia transversal de AC



# Coste Económico



En Méjico



Fábricas en China...



# Coste Ambiental





# Coste Social

Adicción a internet/móvil, estrés de adaptación, pérdida del contacto humano, derechos humanos y condiciones de trabajo en las fábricas en países del sur, ...



# Coste Humano

Atentados contra la salud, la dignidad, la igualdad, la libertad, o la vida misma...



# Coste de construir (huella del producto)

## ■ Ex: Un chip de 32 MB RAM (2 gr.) [Smi08]

- Electricidad generada por 1,6 Kg de combustibles fósiles
- 72 gramos de productos químicos
- 3.200 litros de agua
- 700 gramos de nitrógeno

## ■ Coste energético de fabricación: 41 MJ

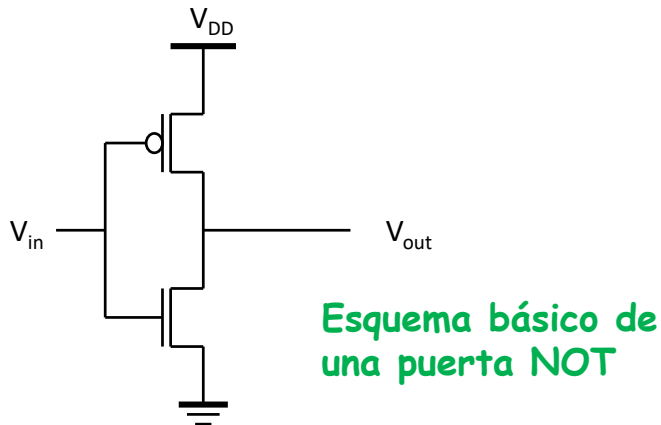
## ■ Consumo durante 4 años de vida: 15 MJ

[Smi08]: V. Smil. *Energy in nature and society: general energetic of complex systems*. The MIT Press, 2008.



# Tecnología de Fabricación

- Tecnología utilizada: **CMOS**
- Elemento básico: **el transistor**

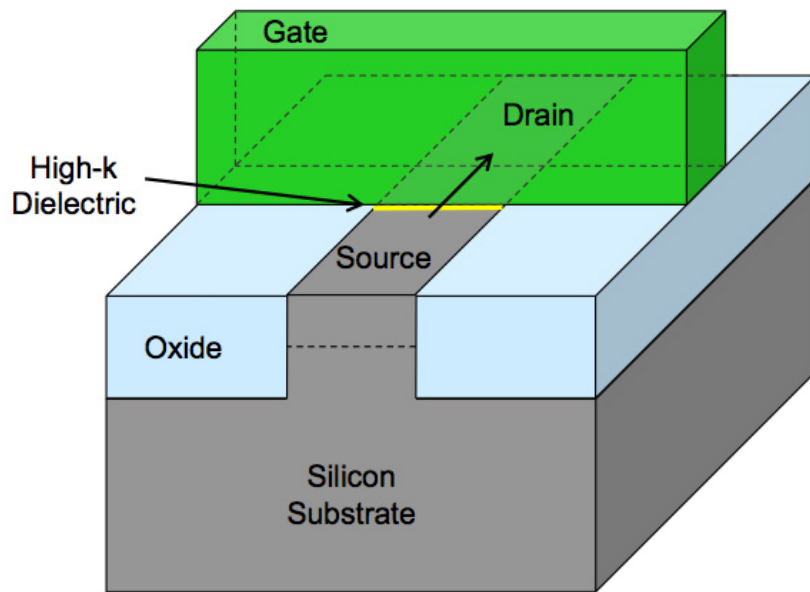


- La tecnología se identifica por la longitud de la puerta del transistor medida en micrómetros ( $\mu\text{m}$ ,  $10^{-6}$  m) o nanómetros (nm,  $10^{-9}$  m).
- Tamaño del átomo de silicio: 0,111 nm

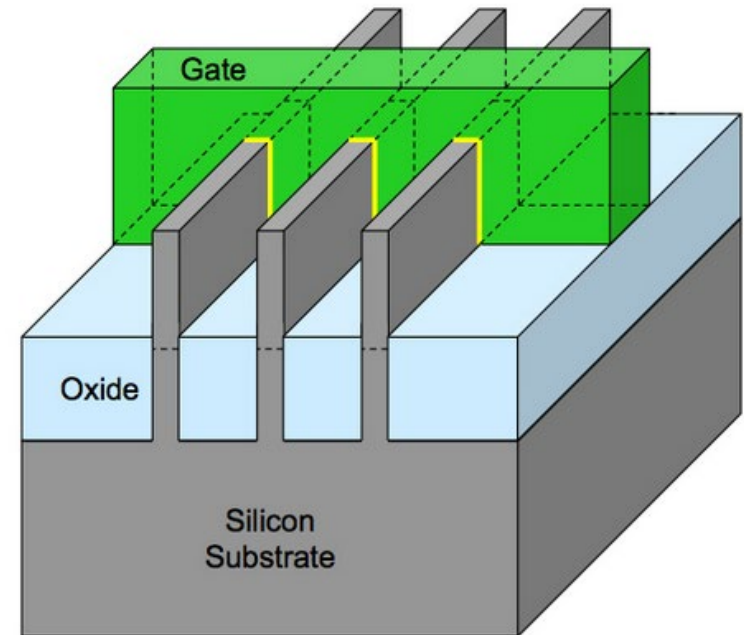
Tecnología		Ejemplo
1971	10 $\mu\text{m}$	Intel 4004
1975	3 $\mu\text{m}$	Intel 8088
1982	1,5 $\mu\text{m}$	Intel 286
1985	1 $\mu\text{m}$	Intel 386
1989	0,8 $\mu\text{m}$	Intel 486
1994	0,6 $\mu\text{m}$	Power PC 601
1995	0,35 $\mu\text{m}$	AMD K5
1998	0,25 $\mu\text{m}$	Alpha 21264
1999	180 nm	Intel Pentium III
2000	130 nm	AMD Athlon XP
2002	90 nm	Intel Pentium 4
2006	65 nm	IBM Cell (PS3)
2008	45 nm	IBM POWER 7
2010	32 nm	Intel Westmere
2011	22 nm	Intel Ivybridge
2014	14 nm	Intel Broadwell
2016	10 nm	Samsung Exynos 9
2018	7 nm	TSMC

# Estructura básica de un transistor

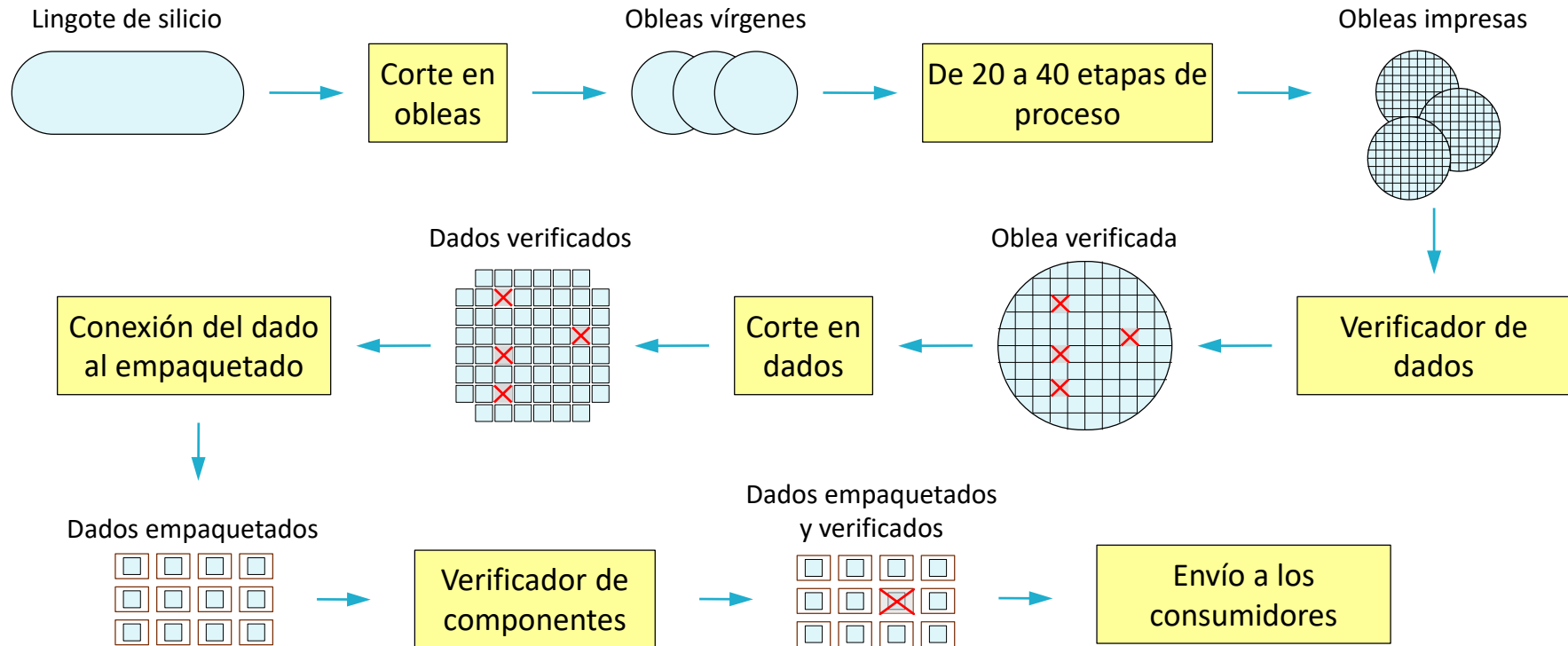
## Traditional Planar Transistor



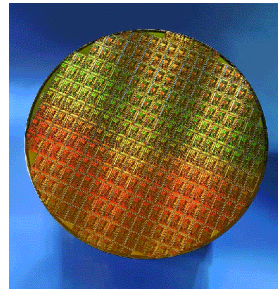
## 22 nm Tri-Gate Transistor



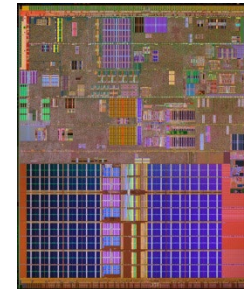
# Proceso de creación de un chip



[www.silfex.com](http://www.silfex.com)



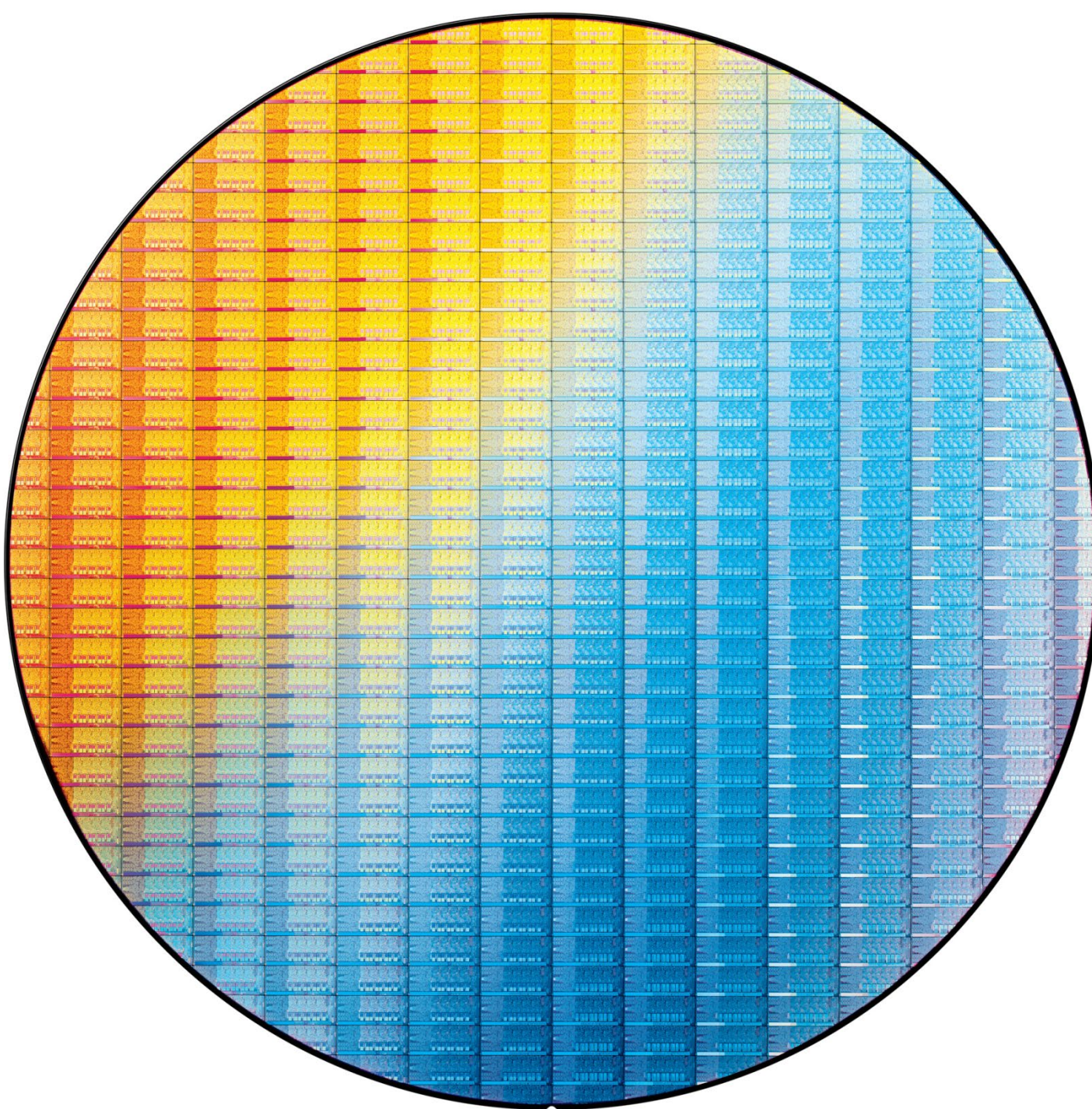
[www.microstockprofit.com](http://www.microstockprofit.com)



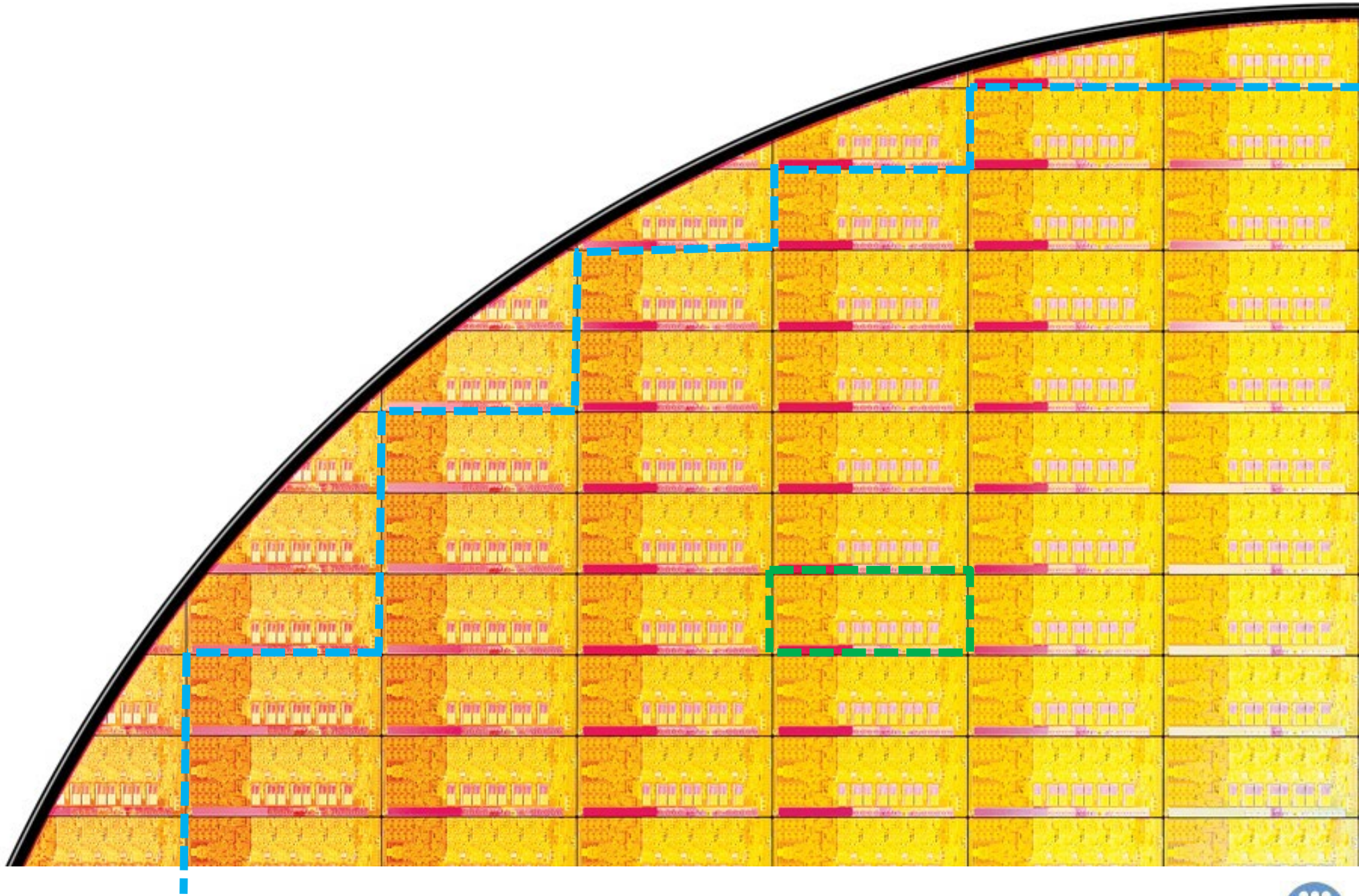
[www.intel.com](http://www.intel.com)



Intel  
Ivybridge  
22nm  
(2012)







# Evaluación del coste

## ■ Factor de yield: fracción de circuitos correctos

## ■ 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*

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

- $\alpha$  = medida de la complejidad, se aproxima al número de máscaras críticas

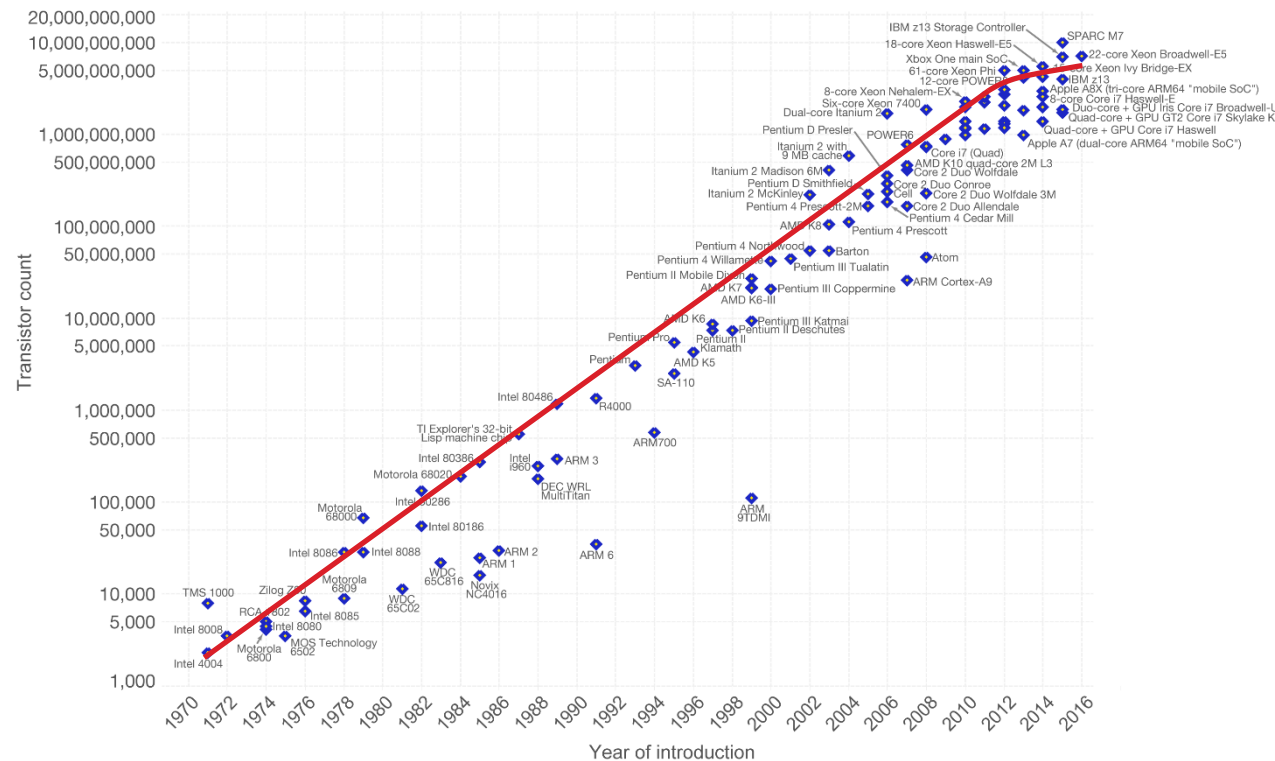


# Ley de Moore

- Gordon E. Moore, cofundador de Intel (1968) junto a Robert Noyce.
- En 1965, afirmó que el número de transistores por unidad de superficie se duplicaría cada año (1).
- En 1975, modificó su afirmación, diciendo que la capacidad de integración se duplicaría cada 24 meses.
- Actualmente se esta desacelerando. ¿Ha llegado el fin de la ley de Moore?

**Moore's Law – The number of transistors on integrated circuit chips (1971-2016)** Our World in Data

Moore's law describes the empirical regularity that the number of transistors on integrated circuits doubles approximately every two years. This advancement is important as other aspects of technological progress – such as processing speed or the price of electronic products – are strongly linked to Moore's law.



Data source: Wikipedia ([https://en.wikipedia.org/wiki/Transistor\\_count](https://en.wikipedia.org/wiki/Transistor_count))  
The data visualization is available at [OurWorldinData.org](https://www.ourworldindata.org). There you find more visualizations and research on this topic.

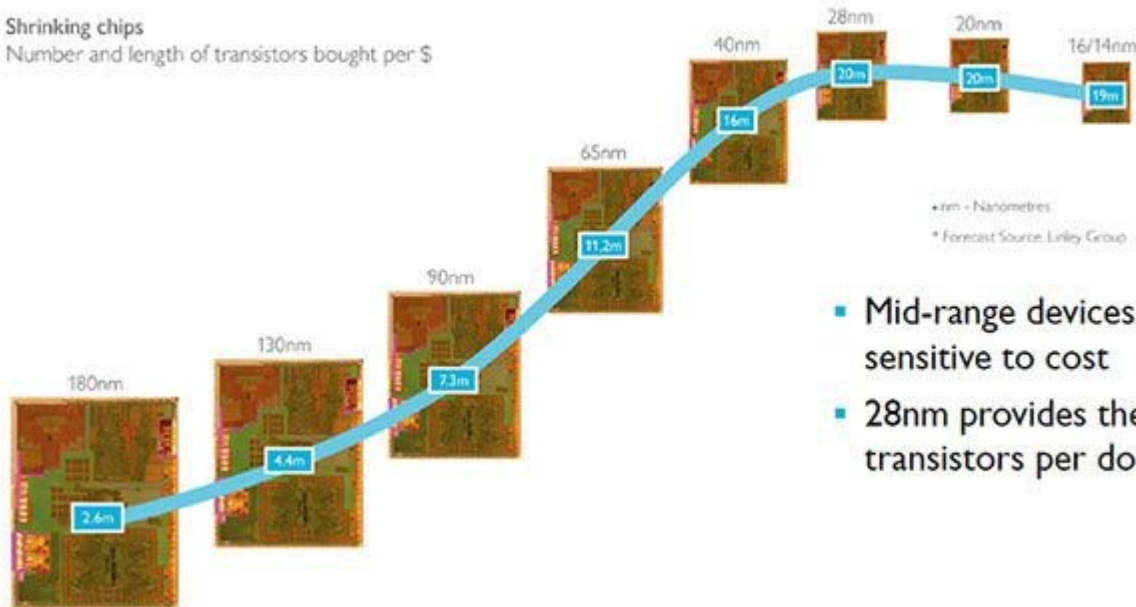
Licensed under CC-BY-SA by the author Max Roser.

# Ley de Moore

- La ley de Moore es realmente una observación económica y no tecnológica.
- Lo que realmente Moore afirmó es que el número de transistores **(que de forma económica)** se pueden integrar en un circuito se duplicaría cada cierto tiempo.

## 28nm: Optimal Balance of Cost and Power for 2015 Devices

Shrinking chips  
Number and length of transistors bought per \$



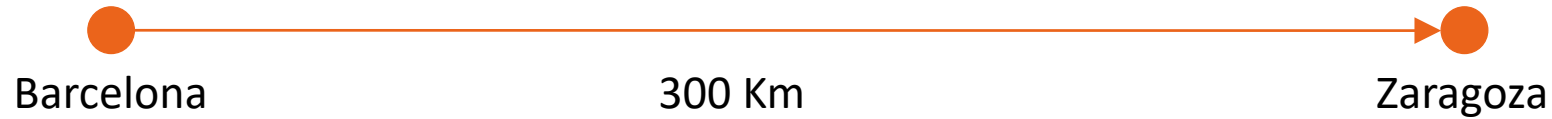
- Mid-range devices are highly sensitive to cost
- 28nm provides the most transistors per dollar

# Latencia y Ancho de Banda

- **LATENCIA:** tiempo que transcurre entre la solicitud de un dato (a memoria por ejemplo) y la disponibilidad del mismo. Se mide en ciclos o unidades de tiempo (s).
- **ANCHO de BANDA:** número de bytes transmitidos por unidad de tiempo. Se mide en KB/s, MB/s, GB/s (siempre potencias de 10).

	Latencia	Ancho de Banda
Memoria DDR3-1600	8,75 ns ( $10^{-9}$ s)	12,8 GB/s
Gigabit Ethernet	190 $\mu$ s ( $10^{-6}$ s)	1 Gb/s
Disco Duro SATA -600	7 ms ( $10^{-3}$ s)	145 MB/s

# Ejemplo de Latencia vs Ancho de Banda

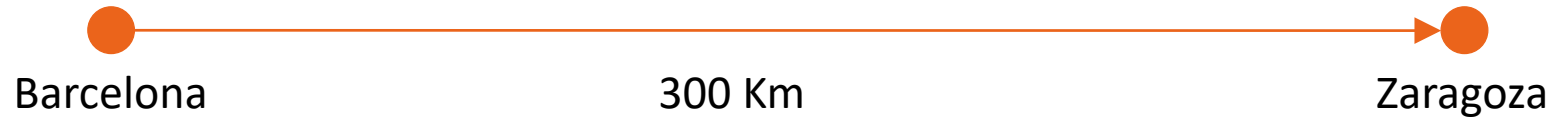


## LATENCIA

- Ping a [www.unizar.es](http://www.unizar.es): **10 ms**
- Ferrari F1 (a 300 Km/h): **3.600 s**
- Camión Volvo FH16 (a 100 Km/h): **10.800 s**

¿Qué método permite transportar más información por unidad de tiempo?

# Ejemplo de Latencia vs Ancho de Banda



## LATENCIA

- Ping a [www.unizar.es](http://www.unizar.es): **10 ms**
- Ferrari F1 (a 300 Km/h): **3.600 s**
- Camión Volvo FH16 (a 100 Km/h): **10.800 s**

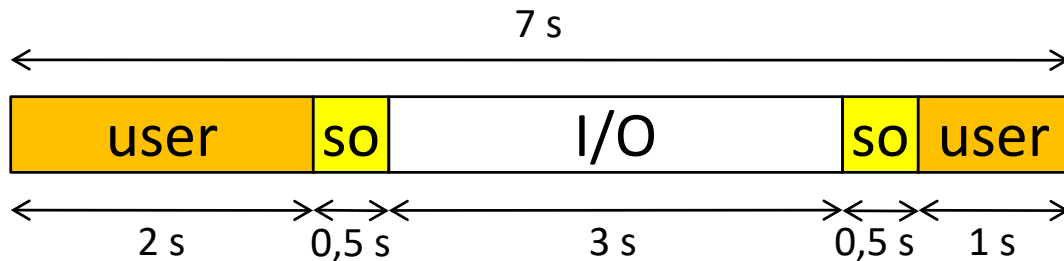
## ANCHO de BANDA

- ADSL a 20 Mb/s: **2,5 MB/s**
- Ferrari F1 (transportando 1 HD de 1 TB): **277,8 MB/s**
- Camión Volvo FH16 (transportando 34.000 HD de 1 TB): **3,15 TB/s**



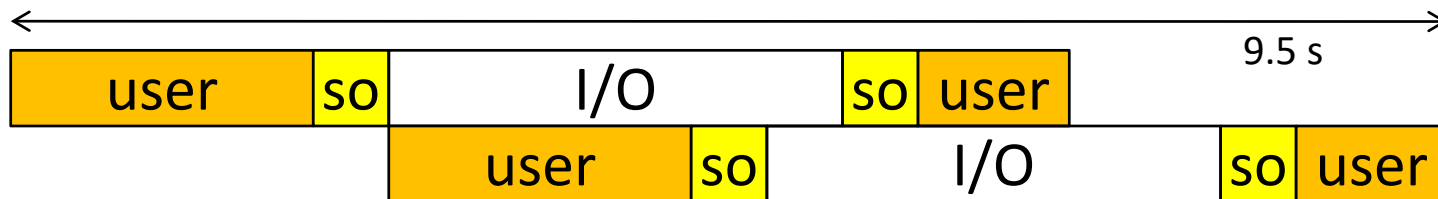
# Productividad vs Tiempo de respuesta

## ■ Tiempo de respuesta (wall time)



- Tiempo de usuario: 3s
- Tiempo de sistema: 1s
- Tiempo de CPU: 4s
- Tiempo de respuesta: 7s
- Throughput:  $1 \text{ proceso} / 7 \text{ s} = 0,14 \text{ procesos/segundo}$

## ■ Productividad (throughput) = trabajo/tiempo



- Throughput =  $2 \text{ procesos} / 9,5 \text{ s} = 0,21 \text{ procesos/segundo}$

# Rendimiento de un procesador

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

Número de instrucciones ejecutadas

Tiempo de ciclo

Número medio de ciclos por instrucción

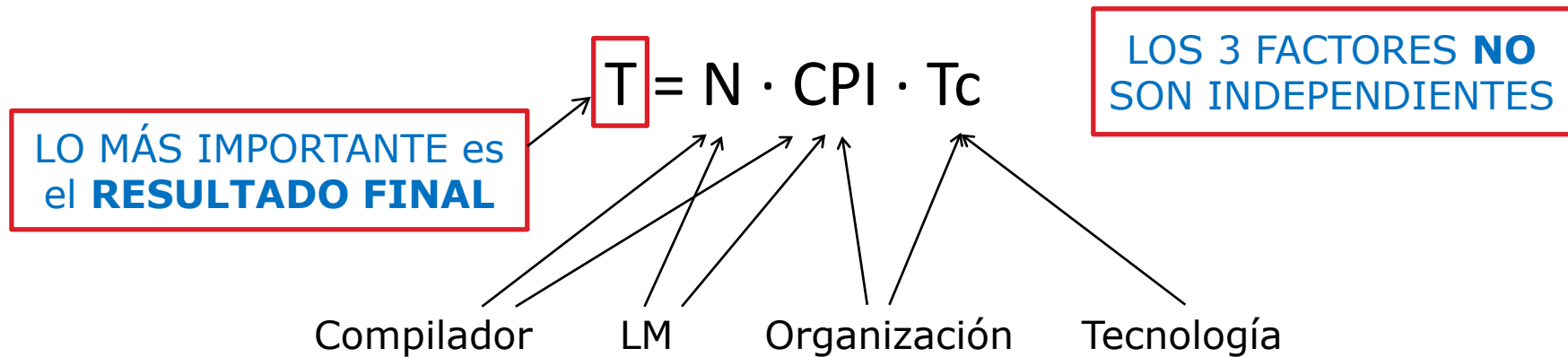
↑Tiempo ejecución ⇒ ↓Rendimiento

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

# Métricas de Rendimiento

## ■ Rendimiento de un procesador

- 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 Rendimientos

- Para comparar el rendimiento de 2 computadores usaremos el tiempo de ejecución:

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

- 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:

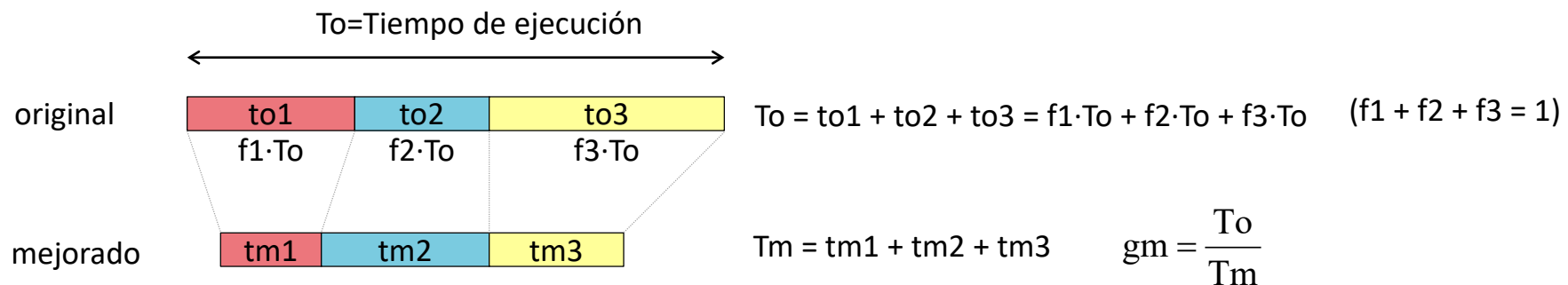
$$\left( \frac{T_A}{T_B} - 1 \right) \cdot 100$$

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

# Límites en la mejora del Rendimiento

## ■ Ley de Amdahl (leed el capítulo correspondiente del H&P)

- Expresión formal del sentido común.



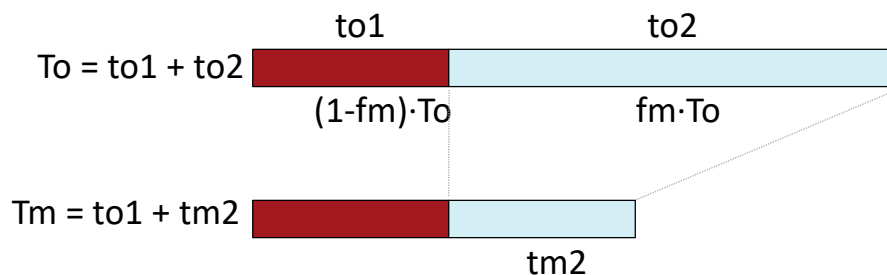
$$g_1 = \frac{to1}{tm1} = \frac{f_1 \cdot T_o}{tm1} \Rightarrow tm1 = \frac{f_1 \cdot T_o}{g_1}$$

$$\text{Ganancia} = \frac{T_o}{T_m} = \frac{to1 + to2 + to3}{tm1 + tm2 + tm3} = \frac{f_1 \cdot T_o + f_2 \cdot T_o + f_3 \cdot T_o}{\frac{f_1 \cdot T_o}{g_1} + \frac{f_2 \cdot T_o}{g_2} + \frac{f_3 \cdot T_o}{g_3}} = \frac{1}{\frac{f_1}{g_1} + \frac{f_2}{g_2} + \frac{f_3}{g_3}}$$

# Mejora del Rendimiento

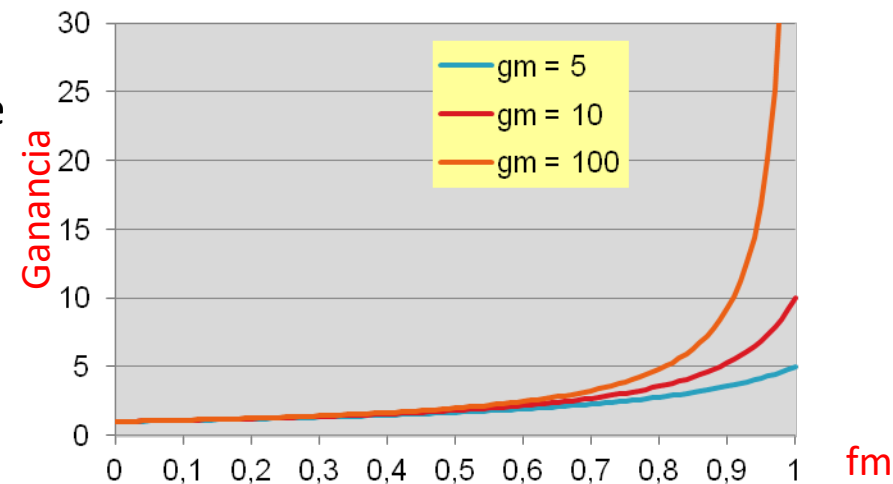
## ■ Ley de Amdahl

- Caso particular: sólo mejoramos una fracción (fm)
  - ✓ Fracción 1 (1-fm): sin mejorar (g1 = 1)
  - ✓ Fracción 2 (fm): mejorada (g2 = gm > 1)
- Si aplicamos una mejora, la ganancia obtenida depende de la fracción del tiempo original donde se usa esa mejora.



$$\text{Ganancia} = \frac{T_o}{T_m} = \frac{1}{1 - f_m + \frac{f_m}{g_m}}$$

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



gm	5	10	50	100	1.000
fm	0,75	0,88	0,979	0,984	0,9989
Ganancia	2,5	5	25	50	500

# Mejora del Rendimiento

## ■ Ley de Amdahl

- Regla de diseño: Optimizar el caso frecuente.
- Ejercicio, **¿Qué ocurre cuando  $gm \rightarrow \infty$ ?**  $\Rightarrow \text{Ganancia} = \frac{1}{1 - fm}$

1-fm : fm	0,8 : 0,2	0,5 : 0,5	0,25 : 0,75	0,1 : 0,9	0,01 : 0,99
Ganancia	1,25	2	4	10	100

- Pero no hay que mejorar excesivamente el caso común. Es más eficaz aumentar la fracción a la que se aplica la mejora, aunque sea con ganancias pequeñas, que obtener ganancias muy grandes sobre una fracción muy pequeña.
  - ✓ Paralelizar el 20% de un programa y ejecutarlo en 1000 CPUs: Ganancia = 1,25
  - ✓ Paralelizar el 99% de un programa y ejecutarlo en 2 CPUs: Ganancia = 1,98
- La ley de Amdahl se puede aplicar a múltiples situaciones.



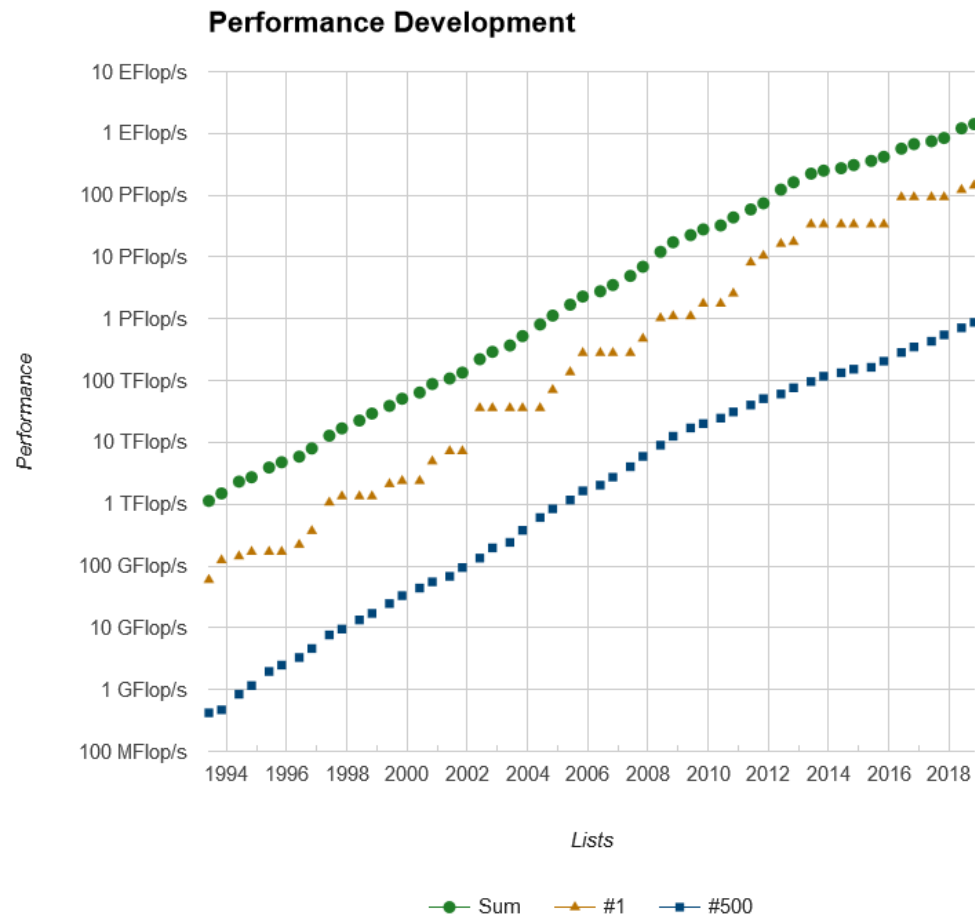
Rank	Name	Computer	Cores	TFlop/s	Power (kW)
1	Summit	IBM Power System AC922, IBM POWER9 22C 3.07GHz, NVIDIA Volta GV100, Dual-rail Mellanox EDR Infiniband	2397824	143500	9783
2	Sierra	IBM Power System S922LC, IBM POWER9 22C 3.1GHz, NVIDIA Volta GV100, Dual-rail Mellanox EDR Infiniband	1572480	94640	7438
3	Sunway TaihuLight	Sunway MPP, Sunway SW26010 260C 1.45GHz, Sunway	10649600	93014	15371
4	Tianhe-2A	TH-IVB-FEP Cluster, Intel Xeon E5-2692v2 12C 2.2GHz, TH Express-2, Matrix-2000	4981760	61444	18482
5	Frontera	Dell C6420, Xeon Platinum 8280 28C 2.7GHz, Mellanox InfiniBand HDR	448448	23516	¿?
6	Piz Daint	Cray XC50, Xeon E5-2690v3 12C 2.6GHz, Aries interconnect , NVIDIA Tesla P100	387872	21230	2384
29	MareNostrum	Lenovo SD530, Xeon Platinum 8160 24C 2.1GHz, Intel Omni-Path	153216	6470	1632

[www.top500.org](http://www.top500.org)

**Datos de junio de 2019**

- Ranking de los 500 supercomputadores más potentes del mundo.
- La lista se actualiza 2 veces al año: junio (ICS) y noviembre (SC)
- LINPACK es la aplicación utilizada para hacer este ranking.

# Top 500

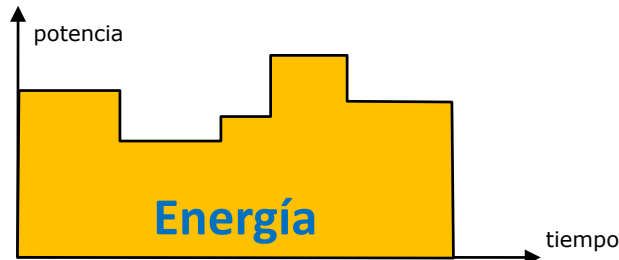


¡ Escala logarítmica !

<https://www.top500.org>

## ■ Potencia y Energía

- Potencia es el trabajo realizado por unidad de tiempo
- Energía = Integral de la potencia en el tiempo



Potencia es el nivel de consumo

Energía es el área (nivel·tiempo)

- La Energía se mide en unidades de trabajo: **Julios**
- Potencia es la energía consumida por unidad de tiempo: **Watios** (Julios/seg)
- Si la potencia es constante:  $\text{Energía} = \text{Potencia} \times t$
- La Potencia es importante por razones de disipación térmica
- La Energía consumida es importante por el coste (económico y/o ambiental) y/o para incrementar la duración de la batería que alimenta al computador.
  - ✓ Batería = Energía

## ■ Energía y potencia eléctricas

Potencia =  $I \times V$  = amperios  $\times$  voltios = watios

Energía =  $P \times t$  =  $I \times V \times t$  = amperios  $\times$  voltios  $\times$  segundo = watios  $\times$  segundo = julios

## ■ La Potencia consumida por un circuito CMOS tiene 3 componentes:

- **Conmutación**: debido a la conmutación entre niveles de tensión en la carga capacitiva efectiva de todo el chip.
- **Corriente de fugas**: los transistores no son ideales.
- **Corriente de cortocircuito**: los dos transistores del inversor están activos cuando la entrada cambia de tensión.

- La potencia debida a conmutación es la más importante, aunque la de fugas representa un porcentaje cada vez mayor debido a las reducidas dimensiones de los transistores.



## ■ Potencia y energía de conmutación:

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

$$\text{Energía} = C \times V^2 \text{ [energía consumida en 1 ciclo de reloj]}$$

siendo,

- $f$ , frecuencia
- $C$ , capacidad efectiva equivalente de todo el chip en 1 ciclo (faradios)
- $V$ , tensión de alimentación

## ■ Potencia de fugas:

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

### Valores típicos

Disco Duro	15 W
Bombilla de 50W	50 W
portátil	75 W
PC de sobremesa	400 W
Lavadora	1.500 W
Potencia contratada hogar estándar	5,5 KW
Ferrari F1(2004) (900 CV, 1 CV = 735 W)	661 KW
Locomotora AVE Madrid-Barcelona	8,8 MW
Central nuclear (producción)	1 GW

# Métricas que relacionan Rendimiento y Potencia

## ■ Métricas de eficiencia

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

- Aproximaciones según el tipo de computador
  - ✓ Portátiles (bajo consumo) →  $1/\text{Energía} = \text{Duración de la batería}$
  - ✓ Supercomputadores (green 500) →  $\text{Mflops} / \text{W}$

Rank	TOP500	Name	Computer	Power (Kw)	GFlops/Watt
1	469	DGX SaturnV Volta	NVIDIA DGX-1 Volta36, Xeon E5-2698v4 20C 2.2GHz, Infiniband EDR, NVIDIA Tesla V100	97	15.1
2	1	Summit	IBM Power System AC922, IBM POWER9 22C 3.07GHz, NVIDIA Volta GV100, Dual-rail Mellanox EDR Infiniband	10096	14.7
3	8	AI Bridging Cloud Infrastructure (ABCI)	PRIMERGY CX2570 M4, Xeon Gold 6148 20C 2.4GHz, NVIDIA Tesla V100 SXM2, Infiniband EDR	1649	14.4
4	393	MareNostrum	Lenovo SD530, Xeon Platinum 8160 24C 2.1GHz, Intel Omni-Path	81	14,13
5	22	TSUBAME3.0	SGI ICE XA, IP139-SXM2, Xeon E5-2680v4 14C 2.4GHz, Intel Omni-Path, NVIDIA Tesla P100 SXM2	792	13.7

[www.green500.org](http://www.green500.org)

Datos de junio de 2019

- Ranking de los 500 supercomputadores más potentes del mundo ordenados según eficiencia energética (MFLOPS/watio).
- El supercomputador que más consume (top2 nov 2017): 17808 kW. = 17.808 MW
- Potencia de ASCO 2: 1027,2 MW (los 500 supercomputadores del top500 consumían todos juntos alrededor de 300 MW en nov 2010)

## ■ Métricas para caracterizar la fiabilidad:

- Fiabilidad: tiempo de funcionamiento continuo sin fallos

- ✓ MTTF = Mean Time To Failure

- Tasa de fallos (Failure rate)

$$\lambda = \frac{1}{\text{MTTF}}$$

- Interrupción del servicio se mide como el tiempo medio necesario para restablecerlo

- ✓ MTTR = Mean Time To Repair

- Tiempo medio entre fallos (Mean Time Between Failures)

- ✓ MTBF = MTTF+MTTR

- Disponibilidad (availability): Fracción del tiempo en que un sistema está funcionando.

$$\text{Availability} = \frac{\text{MTTF}}{\text{MTTF} + \text{MTTR}}$$



## ■ El tiempo entre fallos se aproxima a una distribución exponencial donde:

- $p$  = probabilidad de que se produzca un fallo
- $\lambda = 1/\text{MTTF}$  (failure rate)
- $t$  = tiempo transcurrido

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

## ■ ¿Cómo calcular el MTTF de un sistema, dado el MTTF de los componentes?

- Dados dos componentes con fallos independientes y distribución exponencial
  - ✓ Probabilidades de fallo  $p_1$  y  $p_2$
  - ✓ Tasas de fallo  $\lambda_1$  y  $\lambda_2$
  - ✓ Tiempos medios entre fallos  $\text{MTTF}_1$  y  $\text{MTTF}_2$
- probabilidad de que se produzca un fallo es  $1 -$  “probabilidad que no falle ninguno”, o sea:

$$p = 1 - (1 - p_1) \times (1 - p_2) = 1 - e^{-\lambda_1 t} \times e^{-\lambda_2 t} = 1 - e^{-(\lambda_1 + \lambda_2) t}$$

- Que sigue una distribución exponencial con  $\lambda = \lambda_1 + \lambda_2$  de donde se deduce que :

$$\frac{1}{\text{MTTF}} = \frac{1}{\text{MTTF}_1} + \frac{1}{\text{MTTF}_2}$$

## ■ 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:

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

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

## ■ Una forma de mejorar la fiabilidad es mediante redundancia.

- ✓ En tiempo: Repetir un cálculo para comprobar si es erróneo
- ✓ En recursos: Disponer de componentes extra que reemplazan al que falla

- Ver ejemplos páginas 26 y 27 de H&P

## ■ Redundancia: 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