

# Arquitectura de Computadores

## Tema 1

Introducción: Tendencias Tecnológicas  
Coste / Rendimiento/ Consumo



**D**EPARTAMENTO DE  
**A**RQUITECTURA DE **C**OMPUTADORES  
Y **A**UTOMÁTICA

Curso 2015-2016

# Contenidos

---

- o La asignatura. ¿Qué estudia?
- o El entorno tecnológico
- o Rendimiento
- o Consumo
- o Coste
- o Bibliografía

Capítulo 1 de [HePa12]

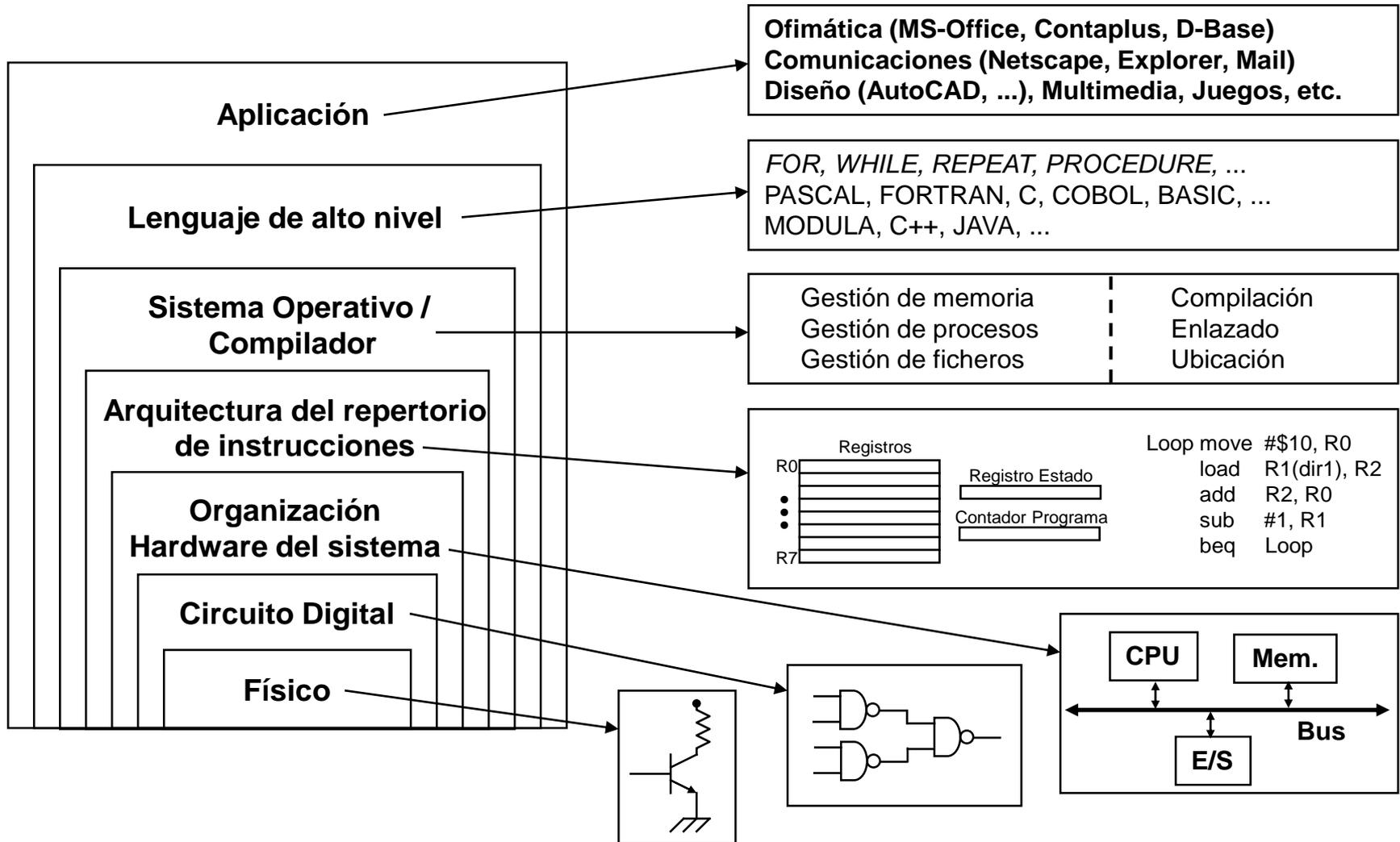
Semiconductor Industry Association. <http://public.itrs.net>

Standard Performance Evaluation Corporation. <http://www.spec.org>

Transaction Processing Council. <http://www.tpc.org>

# La asignatura

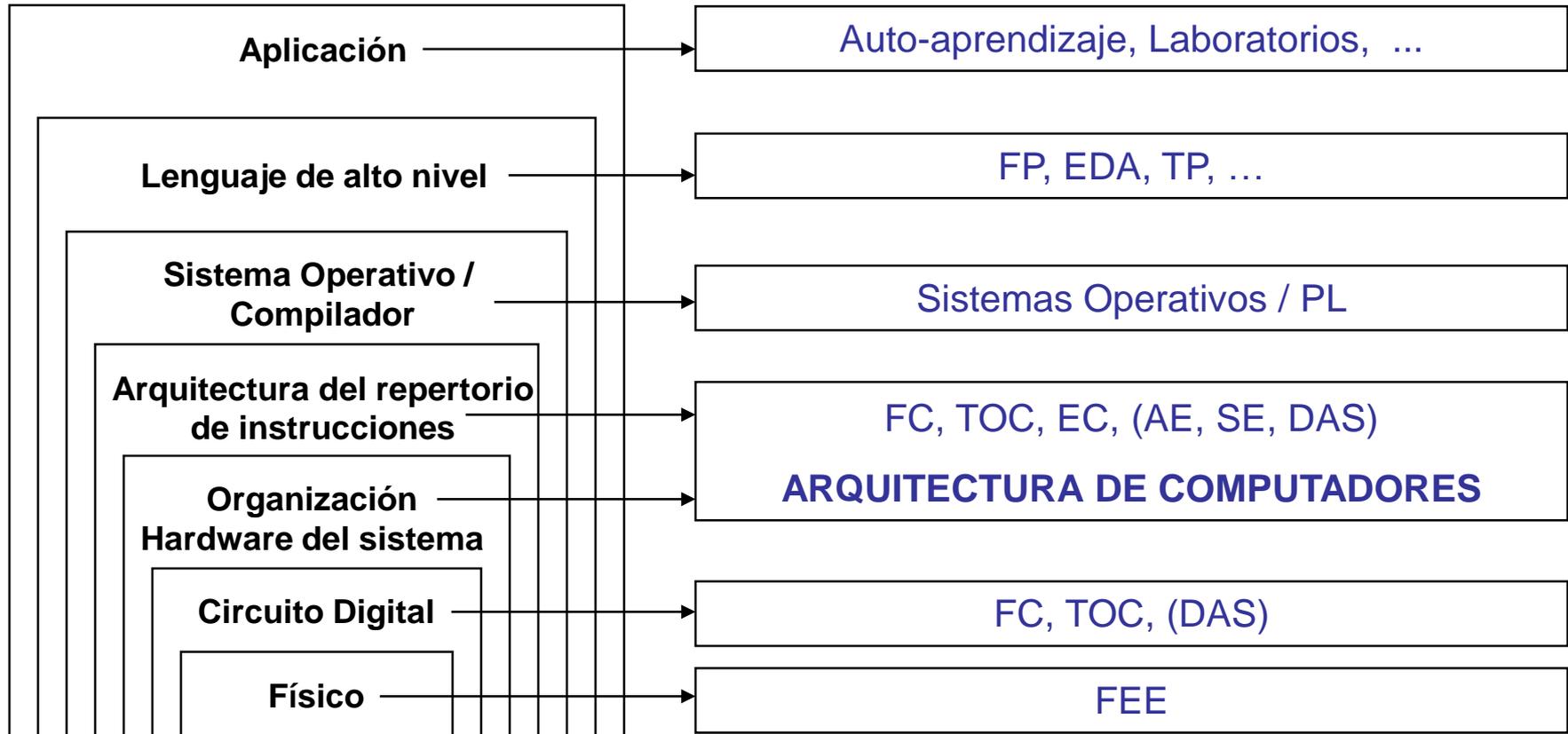
## □ Niveles de descripción y diseño de un computador



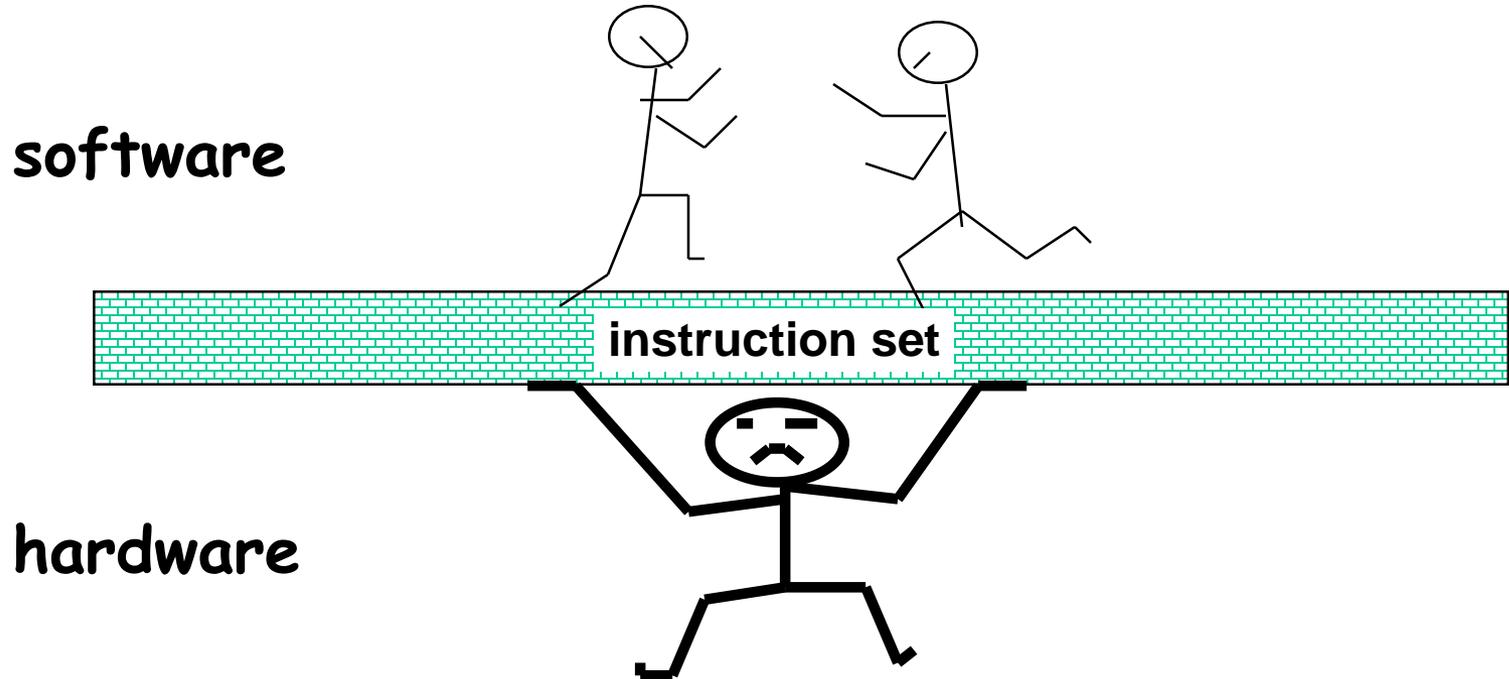
# La asignatura

## □ Niveles de descripción y diseño de un computador

### ¿Dónde se estudia?



## ISA: Interfase Critico



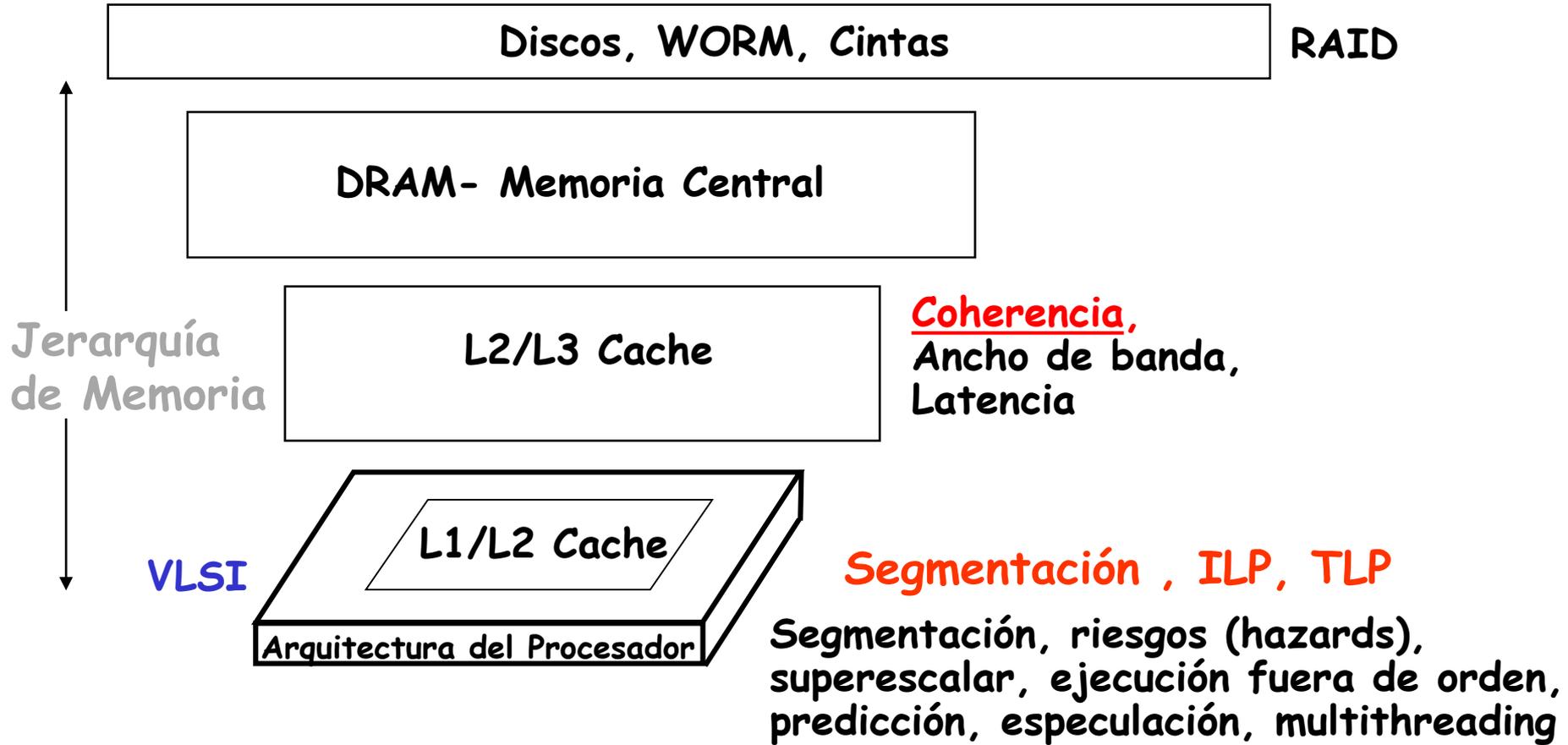
### □ Propiedades

- o Permanencia con el tiempo / tecnología (portabilidad)
- o Proporciona funcionalidad eficaz a los niveles superiores
- o Permite implementación eficiente en los niveles inferiores

# La asignatura

## □ ¿Qué estudia la asignatura?

Entrada/salida y almacenamiento



# La asignatura

- ¿Qué estudia la asignatura?



Switch (S) Procesador (P) Memoria (M)

**Multiprocesadores**  
**Redes de Interconexión**

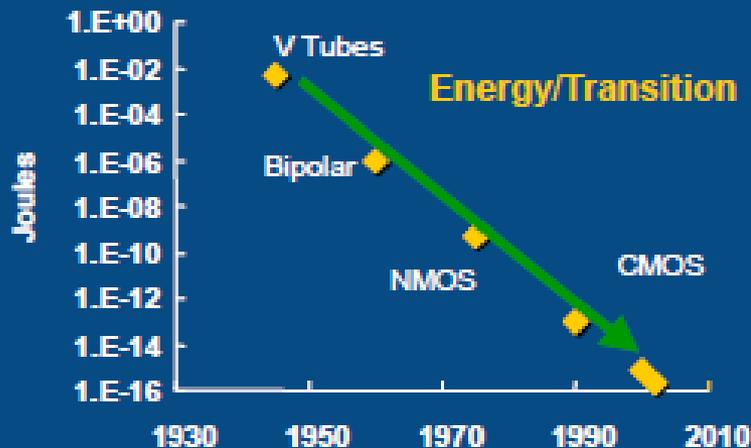
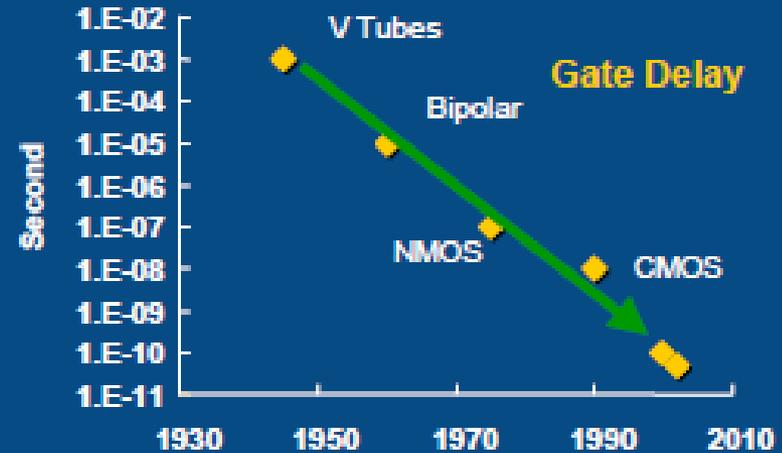
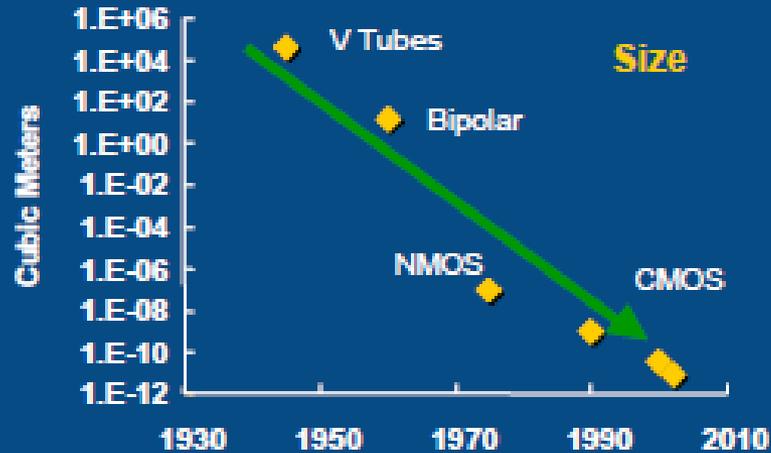
**Memoria compartida:**  
centralizada,  
distribuida,  
paralelismo de datos

**Red**

Topología,  
Routing,  
Ancho de banda,  
Latencia,

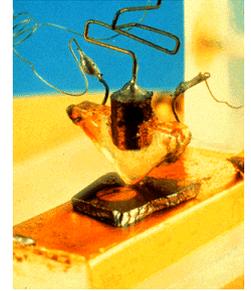
# Evolución y tendencias

## El escalado de la tecnología continua.

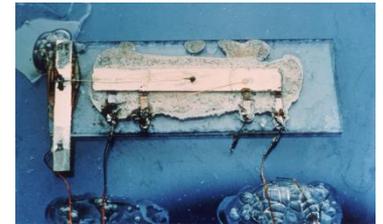


# Evolución y tendencias

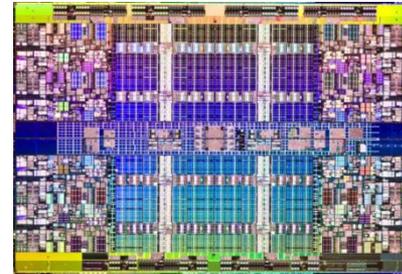
- ❑ 1949 EDSAC  $10^2$  op/seg
- ❑ 1957 Transistor: de  $10^3$  a  $10^4$  op/seg
  - DEC PDP-1 (1957)
  - IBM 7090 (1960)
- ❑ 1965 CI: de  $10^5$  a  $10^6$  op/seg
  - IBM System 360 (1965)
  - DEC PDP-8 (1965)
- ❑ 1971 Microprocesador
  - Intel 4004
- ❑ 2003 más de  $3 \times 10^{13}$  op/seg
- ❑ 2010  $> 10^{15}$  op/seg (1 Pflops)
- ❑ Jun 2014 (top 500)
  - 1° Tianhe-2 (MilkyWay-2) 3,120,000 cores, 33.9 Pflops
  - 41° Mare Nostrum 48,896 cores, 925 Tflops



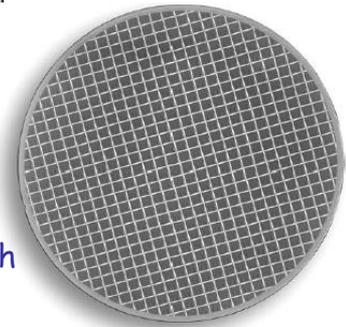
Transistor (47) PN 56



CI (58) PN2000



Intel Xeon 7500, 8c, 16Th



Oblea (Wafer)

# Evolución y tendencias

## Top 500: junio 2011

Desde lista anterior: 7 meses

Capac. total agregada: pasa de 43.6 Pflops a 58.9 Pflops

Capac. último lista: pasa de 31.1 Tflops a 40.2 Tflops



# Evolución y tendencias

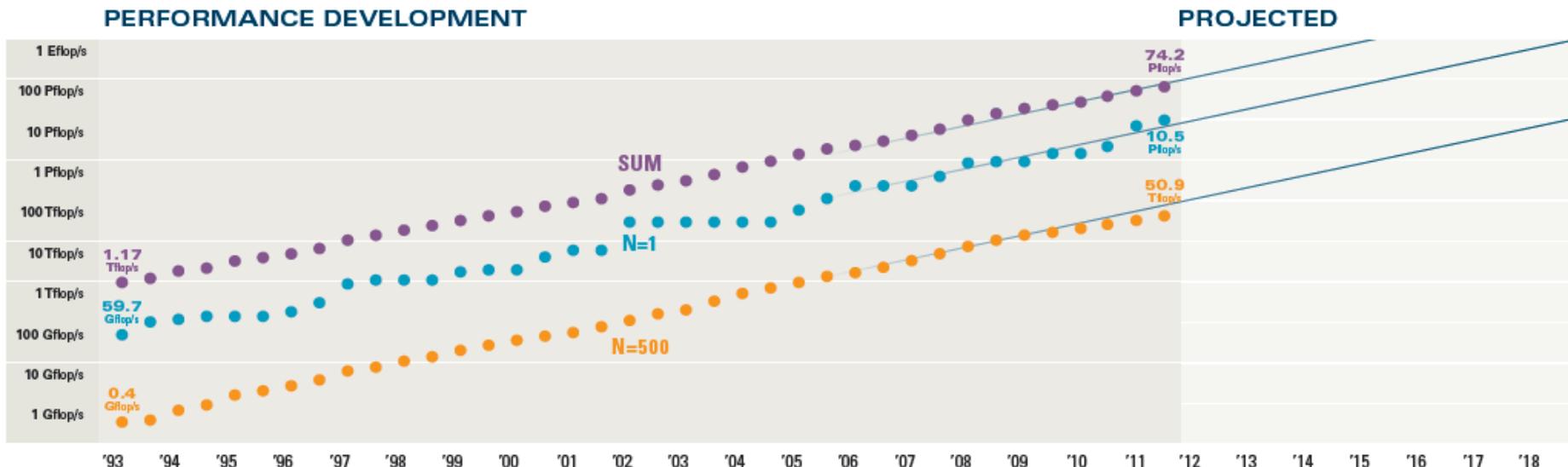
## Top 500: noviembre 2011

De lista anterior a esta: 5 meses

Capacidad Total: pasa de 58.9 Pflops a 74.2 Pflops

Capacidad último lista: pasa de 40.2 Tflops a 50.9 Tflops

	NAME/MANUFACTURER/COMPUTER	SITE	COUNTRY	CORES	R <sub>max</sub> Pflop/s
1	K computer SPARC64 VIIIfx 2.0GHz, Tofu interconnect	RIKEN	Japan	705,024	10.5
2	Tianhe-1A 6-core Intel X5670 2.93 GHz + Nvidia M2050 GPU w/custom interconnect	NUDT/NSCC/Tianjin	China	186,368	2.57
3	Jaguar Cray XT-5 6-core AMD 2.6 GHz w/custom interconnect	DOE/OS/ORNL	USA	224,162	1.76
4	Nebulae Dawning TC3600 Blade Intel X5650 2.67 GHz, NVidia Tesla C2050 GPU w/ Iband	NSCS	China	120,640	1.27
5	Tsubame 2.0 HP Proliant SL390s G7 nodes (Xeon X5670 2.93GHz) , NVIDIA Tesla M2050 GPU w/Iband	TiTech	Japan	73,278	1.19



# Evolución y tendencias

## Top 500: jun 2014

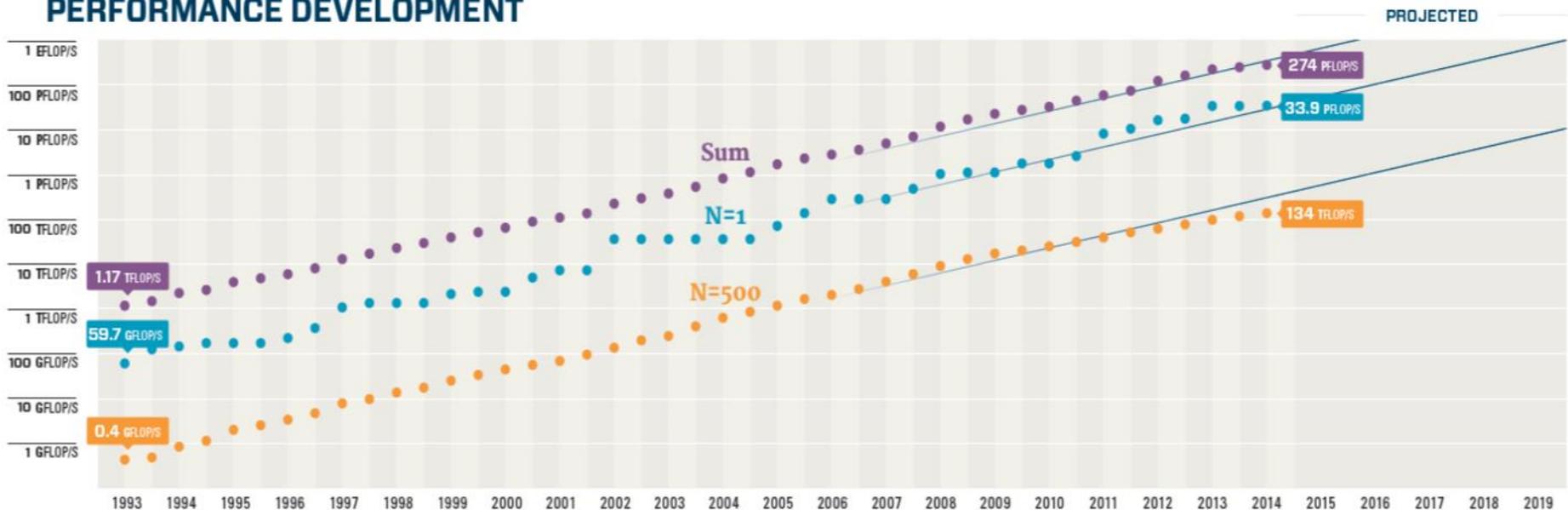
Capacidad Total: 274 Pflops

Último lista: 134 Tflops



	NAME	SPECS	SITE	COUNTRY	CORES	RMAX PFLOP/S	POWER MW
1	<b>Tianhe-2 (Milkyway-2)</b>	NUDT, Intel Ivy Bridge (12C, 2.2 GHz) & Xeon Phi (57C, 1.1 GHz), Custom interconnect	NSCC Guangzhou	China	3,120,000	33.9	17.8
2	<b>Titan</b>	Cray XK7, Operon 6274 (16C 2.2 GHz) + Nvidia Kepler GPU, Custom interconnect	DOE/SC/ORNL	USA	560,640	17.6	8.2
3	<b>Sequoia</b>	IBM BlueGene/Q, Power BQC (16C 1.60 GHz), Custom interconnect	DOE/NNSA/LLNL	USA	1,572,864	17.2	7.9
4	<b>K computer</b>	Fujitsu SPARC64 VIIIfx (8C, 2.0GHz), Custom interconnect	RIKEN AICS	Japan	705,024	10.5	12.7
5	<b>Mira</b>	IBM BlueGene/Q, Power BQC (16C, 1.60 GHz), Custom interconnect	DOE/SC/ANL	USA	786,432	8.59	3.95

## PERFORMANCE DEVELOPMENT



## □ La Ley de Moore

### Cramming More Components onto Integrated Circuits

GORDON E. MOORE, LIFE FELLOW, IEEE

*With unit cost falling as the number of components per circuit rises, by 1975 economics may dictate squeezing as many as 65 000 components on a single silicon chip.*

*The future of integrated electronics is the future of*

*Each approach evolved rapidly and converged so that each borrowed techniques from another. Many researchers believe the way of the future to be a combination of the*

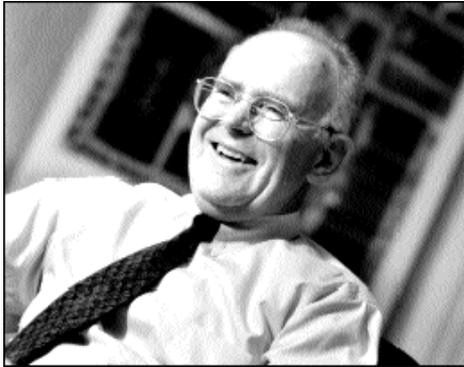


Fig. 2.

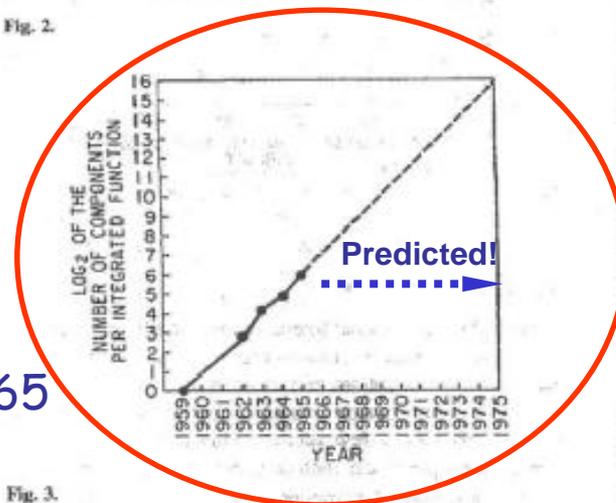


Fig. 3.

diagram to technological realization without any sp engineering.

It may prove to be more economical to build systems out of smaller functions, which are separately aged and interconnected. The availability of large functions combined with functional design and construction, shall allow the manufacturer of large systems to design construct a considerable variety of equipment both rapid and economically.

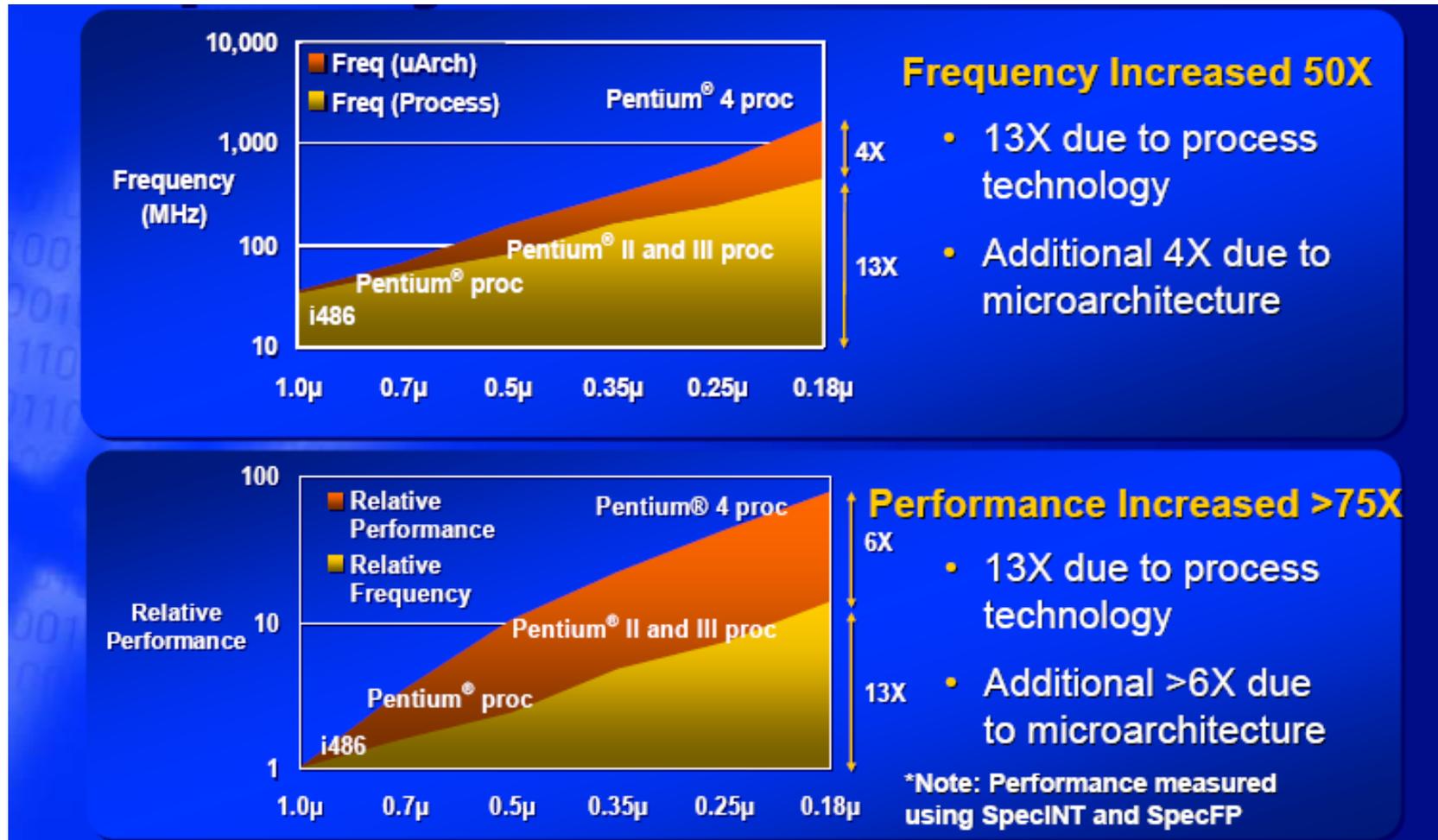
#### IX. LINEAR CIRCUITRY

Integration will not change linear systems as radical digital systems. Still, a considerable degree of integration will be achieved with linear circuits. The lack of high value capacitors and inductors is the greatest fundamental limitation to integrated electronics in the linear area.

Electronics - Abril 1965



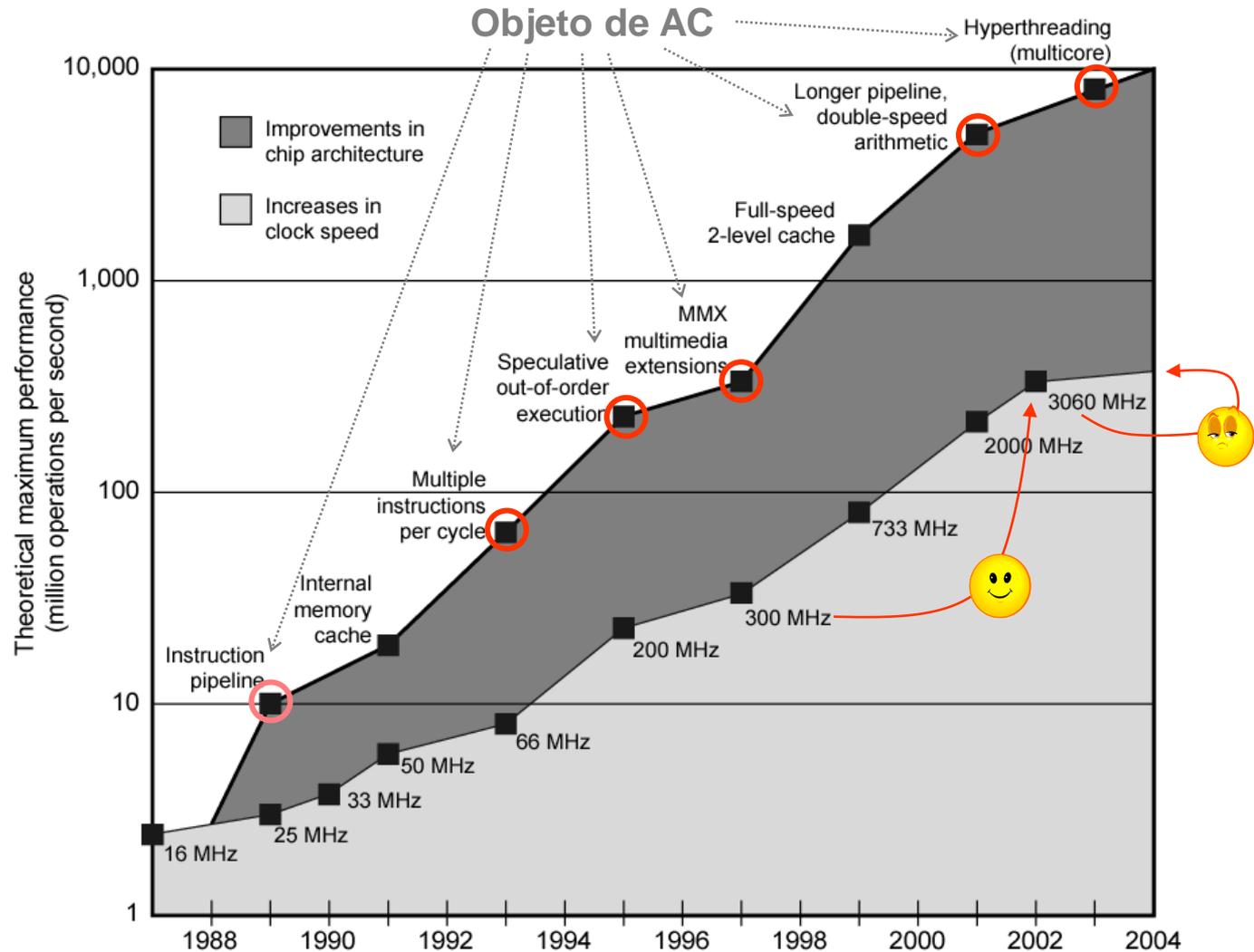
## □ Microelectrónica y microarquitectura



**50X in frequency and 75X in performance**

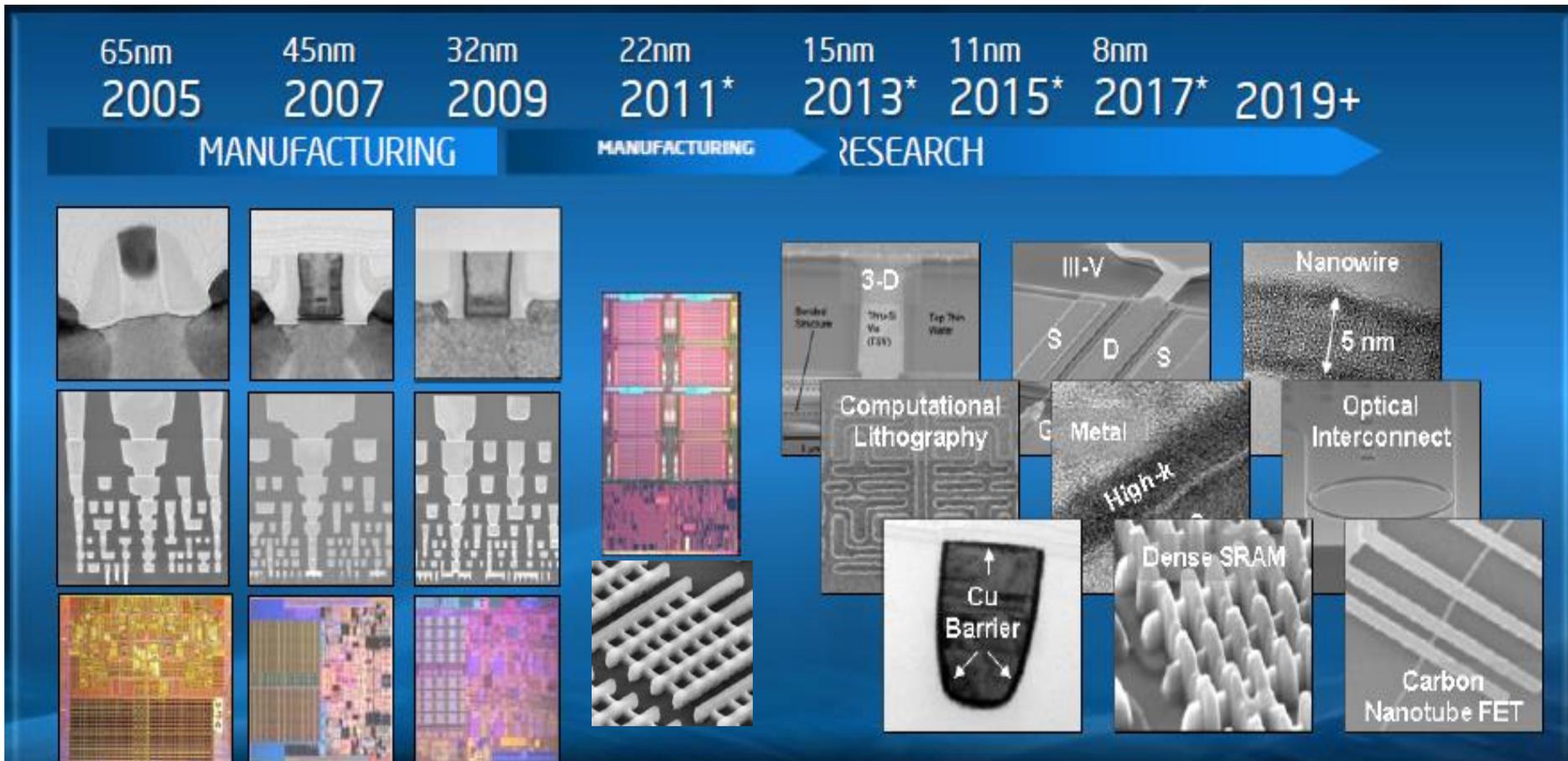
- ❑ Microelectrónica + Microarquitectura
- ❑ Una industria con un progreso que no tiene equivalente
- ❑ Doblado cada 18 meses (1982-2000):
  - Total de incremento 3,200X
  - Los coches viajarían a 176,000 MPH; y recorrerían 64,000 millas/gal.
  - El viaje: L.A. a N.Y. en 5.5 seg (MACH 3200)
- ❑ Doblado cada 24 meses (1971-2001):
  - total de incremento 36,000X
  - Los coches viajarían a 2,400,000 MPH; y recorrerían 600,000 millas/gal.
  - El viaje: L.A. a N.Y. en 0.5 seg (MACH 36,000)

## □ Microelectrónica y microarquitectura



# La Ley de Moore

El escalado de la tecnología puede acabar hacia el final de la década  
El grosor del aislante de la puerta esta limitado a 2nm



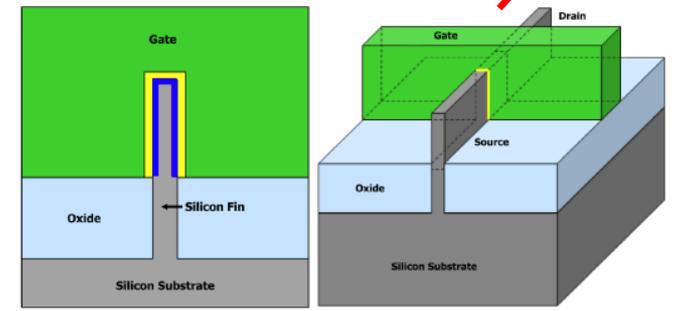
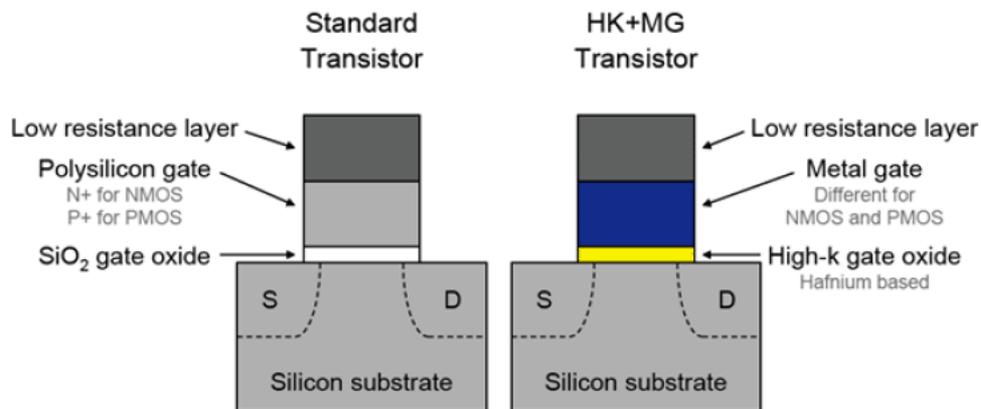
Fuente: Intel Corporation

# La Ley de Moore

Process Name	P856	P858	Px60	P1262	P1264	P1266	P1268	P1270
1 <sup>st</sup> production	1997	1999	2001	2003	2005	2007	2009	2011
Process	250 nm	180 nm	130 nm	90 nm	65 nm	45 nm	32 nm	22 nm
Wafer size	200	200	200/300	300	300	300	300	300
Inter-connect	Al	Al	Cu	Cu	Cu	Cu	Cu	?
Metal layers	5	6	6	7	8	9	9	?
Channel	Si	Si	Si	Strained Si	Strained Si	Strained Si	Strained Si	Strained Si
Gate Dielectric	SiO <sub>2</sub>	High - k	High - k	High - k				
Gate electrode	Poly-Si	Poly-Si	Poly-Si	Poly-Si	Poly-Si	Metal	Metal	Metal
Lithography	248 nm	248 nm	248 nm	193 nm	193 nm	193 nm	<b>193nm</b>	EUV 13.4 nm

(Subject to change)

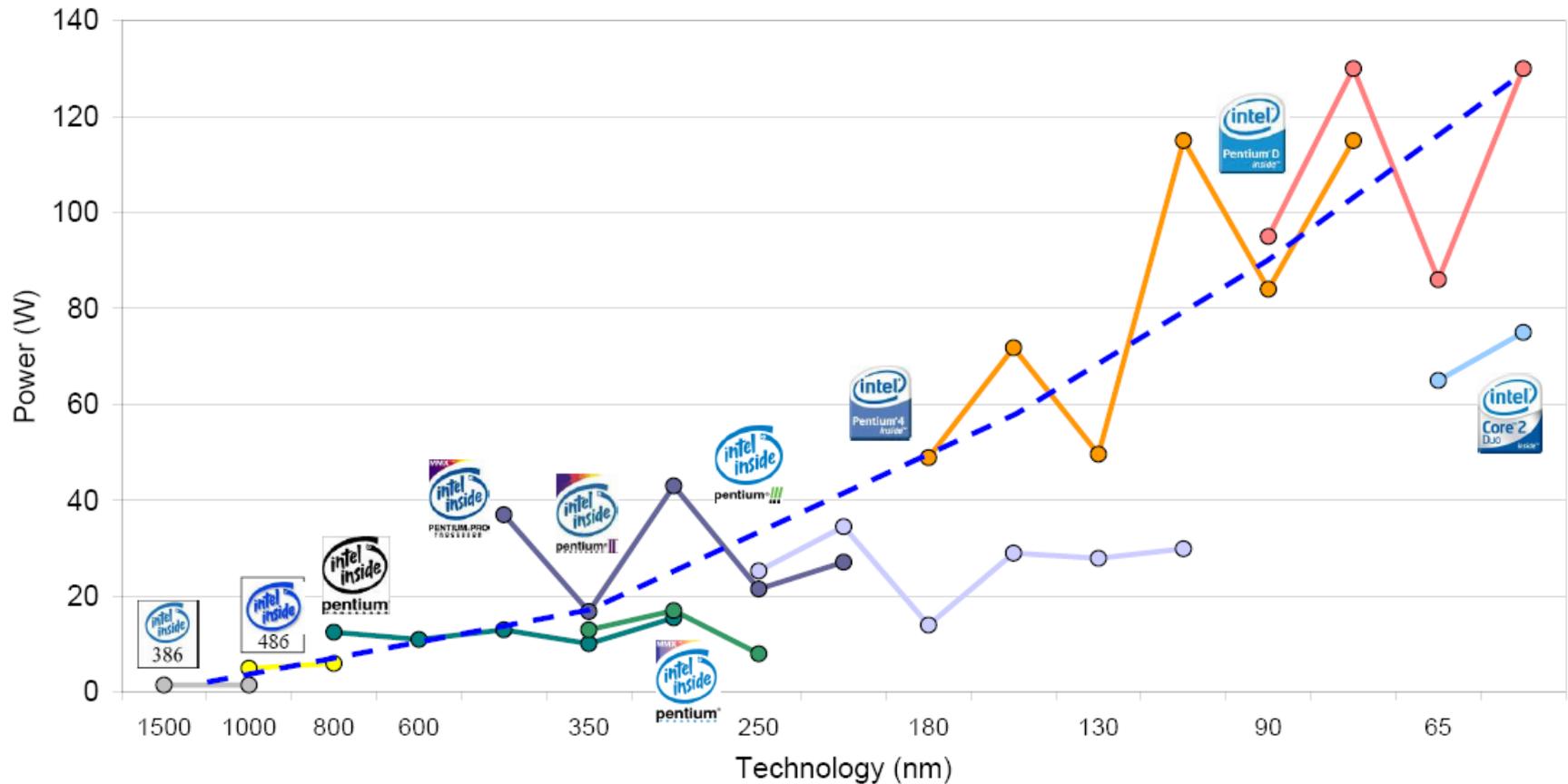
## Manufacturing process details from 1997 to 2011



Fuente: Intel Corporation

# La Ley de Moore: problemas

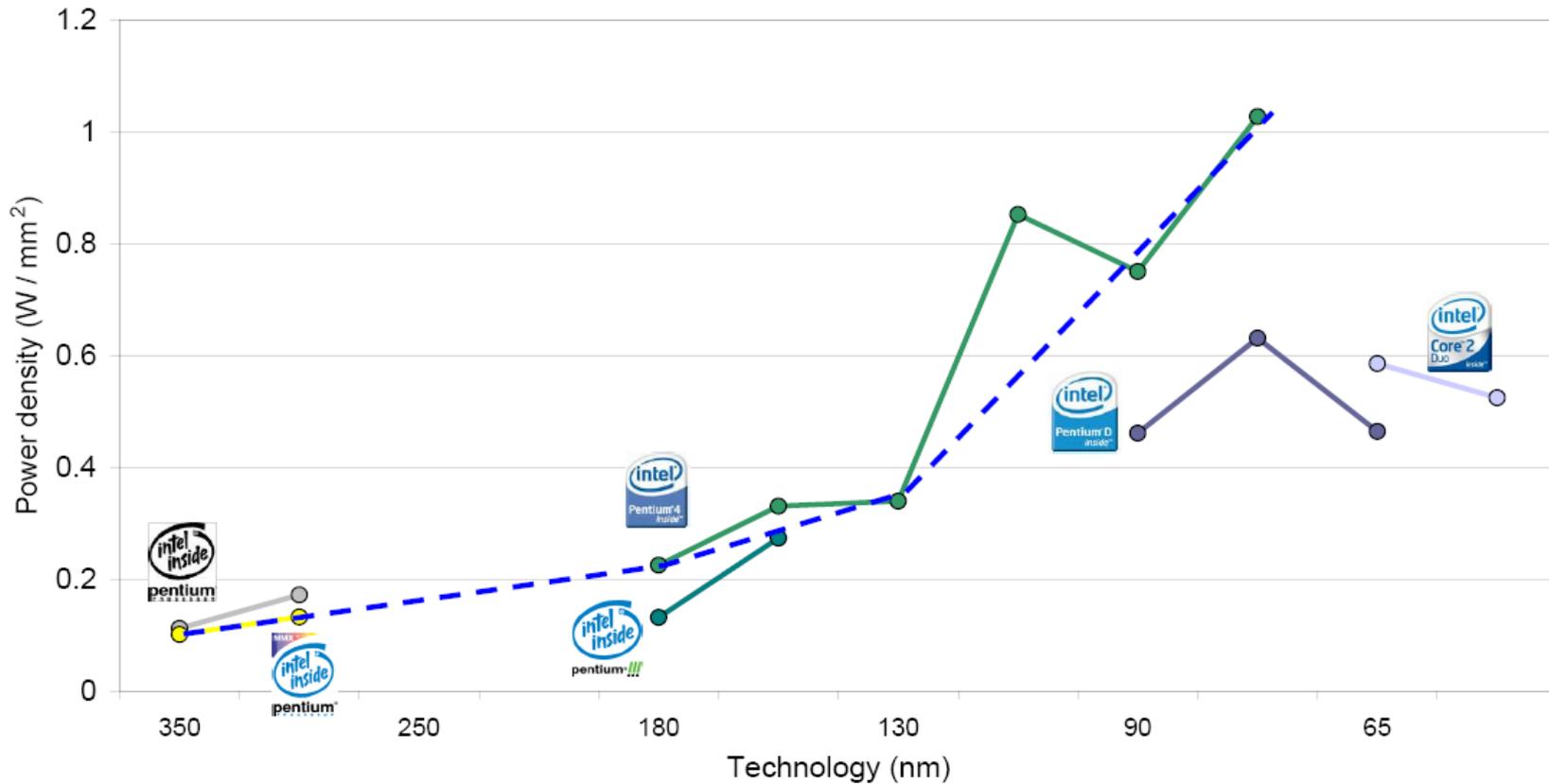
## 1er problema: consumo "Power Wall"



Fuente: Intel Corporation

# La Ley de Moore: problemas

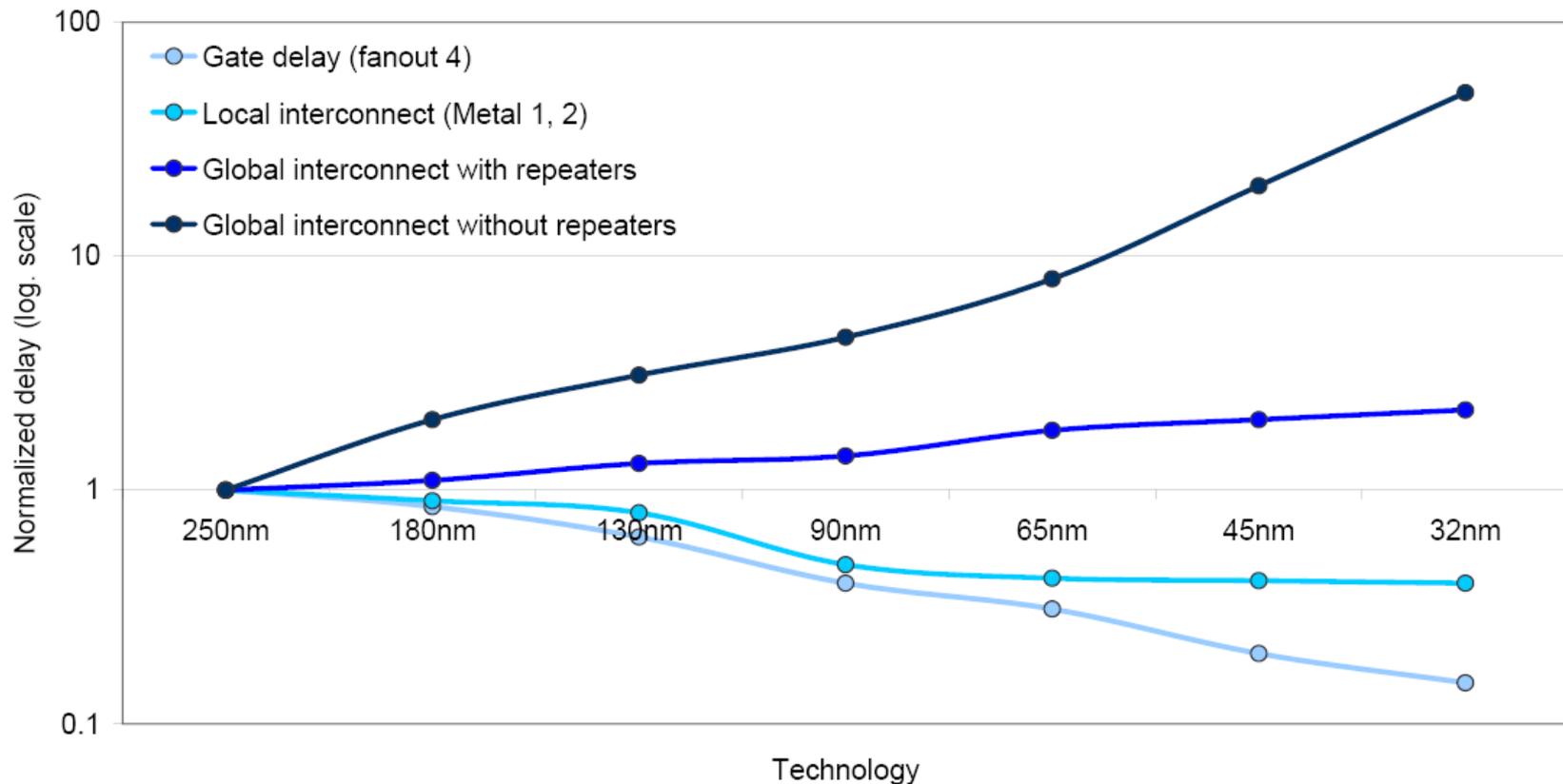
## 1<sup>er</sup> problema: consumo "Power Wall"



Fuente: Intel Corporation

## 2º problema: retardo interconexiones

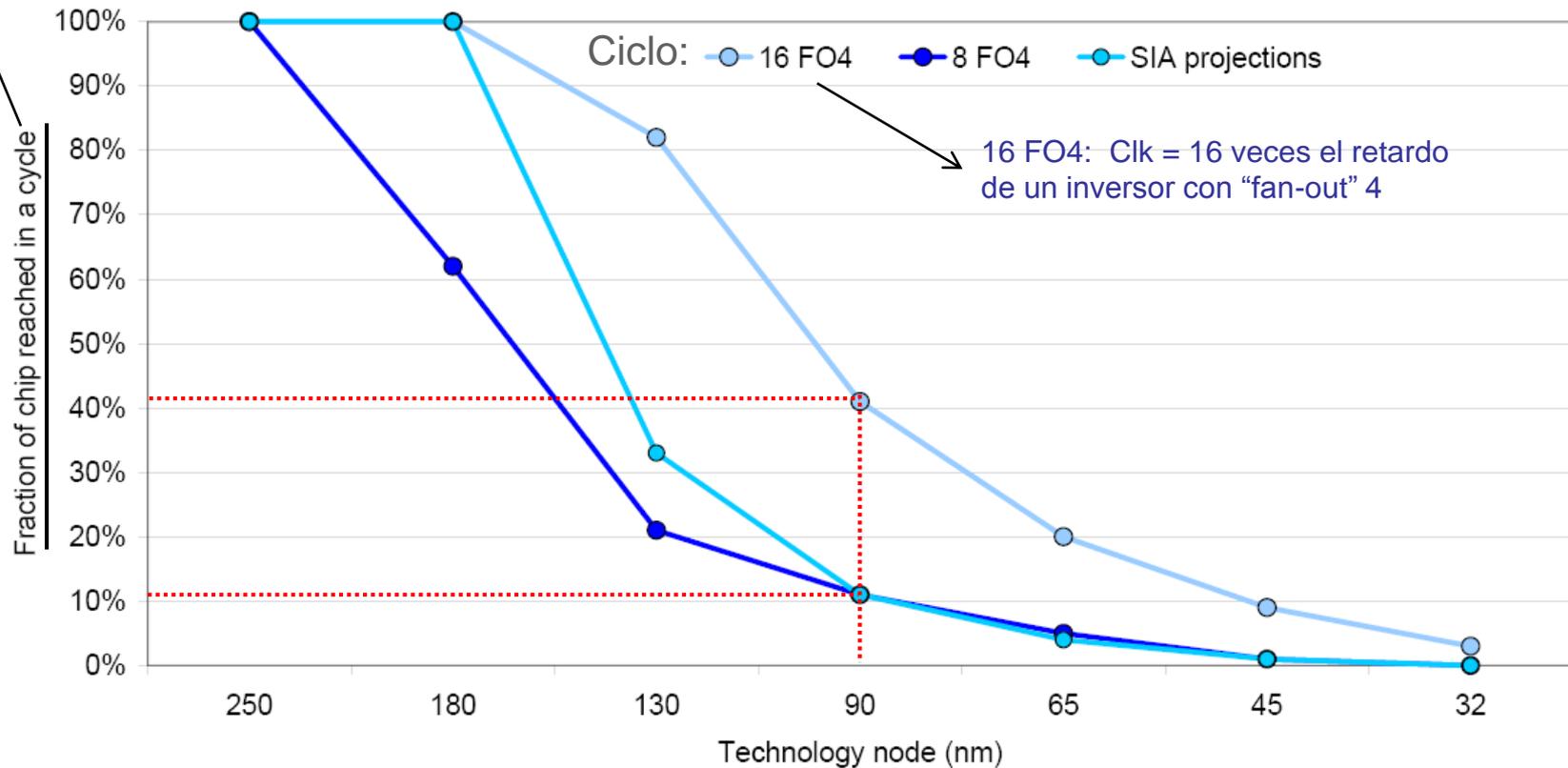
El rendimiento del transistor (retardo) escala linealmente con la tecnología  
El retardo de la interconexión no escala con la tecnología



# La Ley de Moore: problemas

## 2º problema: retardo interconexiones. ¿Qué % del área del chip se puede alcanzar en un ciclo de reloj?

Pero...¿cuánto dura un ciclo?



Multi - Many cores

# La Ley de Moore

## El punto de inflexión

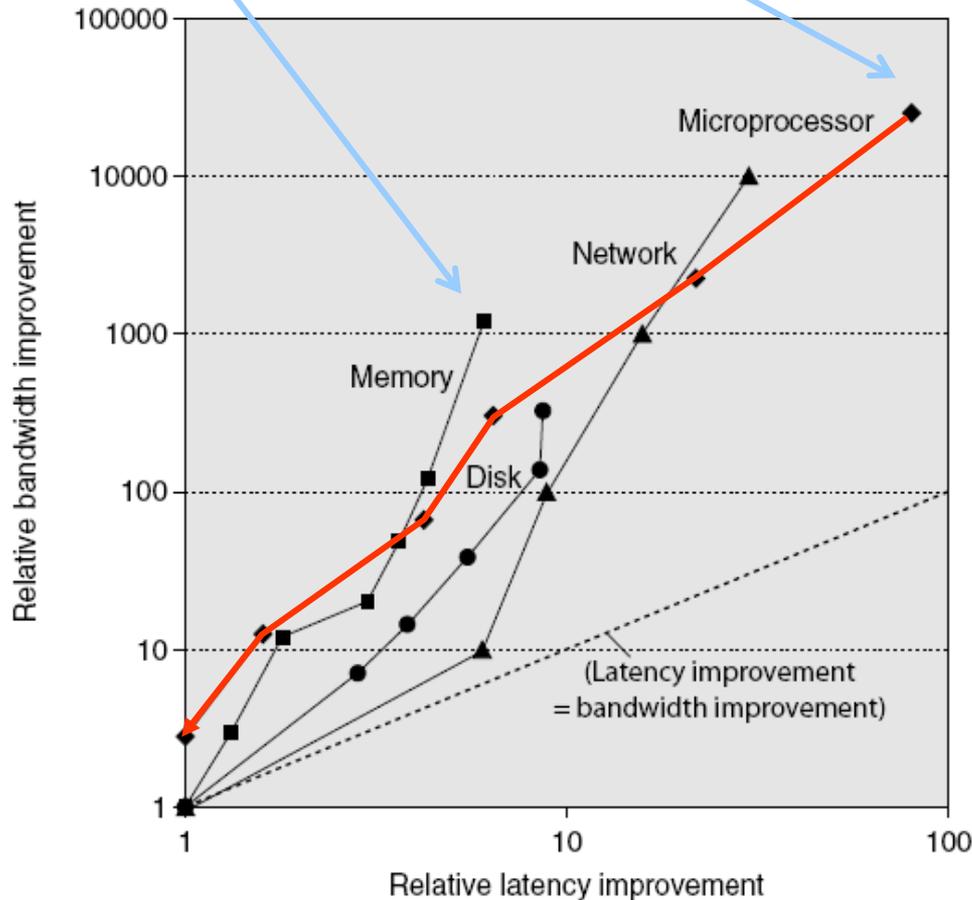
May 17, 2004 ... Intel, the world's largest chip maker, publicly acknowledged that it had hit a **"thermal wall"** on its microprocessor line. As a result, the company is changing its product strategy and disbanding one of its most advanced design groups. Intel also said that it would abandon two advanced chip development projects ... **Now, Intel is embarked on a course already adopted by some of its major rivals: obtaining more computing power by stamping multiple processors on a single chip** rather than straining to increase the speed of a single processor ... Intel's decision to change course and **embrace a "dual core" processor structure** shows the challenge of overcoming the effects of heat generated by the constant on-off movement of tiny switches in modern computers ... some analysts and former Intel designers said that *Intel was coming to terms with escalating heat problems so severe they threatened to cause its chips to fracture at extreme temperatures...*

**New York Times, May 17, 2004**

# El entorno: tendencias

- ❑ Latencia y ancho de banda en los últimos 25 años: desequilibrios

**CPU alta,  
Memoria Baja  
("Memory Wall")**



- ❑ Procesador: '286, '386, '486, Pentium, Pentium 4, Core i7 (80x,25000x)
- ❑ Ethernet: 10Mb, 100Mb, 1G/s, 10Gb/s, 100Gb/s (30x,10000x)
- ❑ Modulo de Memoria: DRAM, Page Mode DRAM, SDRAM, DDR2-3 SDRAM (6x,1200x)
- ❑ Disco : 3600, 5400, 7200, 10000, 15000 RPM (14x, 350x)

# El entorno: tendencias

## □ Latencia y ancho de banda en los últimos 25 años: Procesador

Microprocessor	16-bit address/ bus, microcoded	32-bit address/ bus, microcoded	5-stage pipeline, on-chip I & D caches, FPU	2-way superscalar, 64-bit bus	Out-of-order 3-way superscalar	Out-of-order superpipelined, on-chip L2 cache	Multicore OOO 4-way on chip L3 cache, Turbo
Product	Intel 80286	Intel 80386	Intel 80486	Intel Pentium	Intel Pentium Pro	Intel Pentium 4	Intel Core i7
Year	1982	1985	1989	1993	1997	2001	2010
Die size (mm <sup>2</sup> )	47	43	81	90	308	217	240
Transistors	134,000	275,000	1,200,000	3,100,000	5,500,000	42,000,000	1,170,000,000
Processors/chip	1	1	1	1	1	1	4
Pins	68	132	168	273	387	423	1366
Latency (clocks)	6	5	5	5	10	22	14
Bus width (bits)	16	32	32	64	64	64	196
Clock rate (MHz)	12.5	16	25	66	200	1500	3333
Bandwidth (MIPS)	2	6	25	132	600	4500	50,000
Latency (ns) *	320	313	200	76	50	15	4

x 25K  
x 80

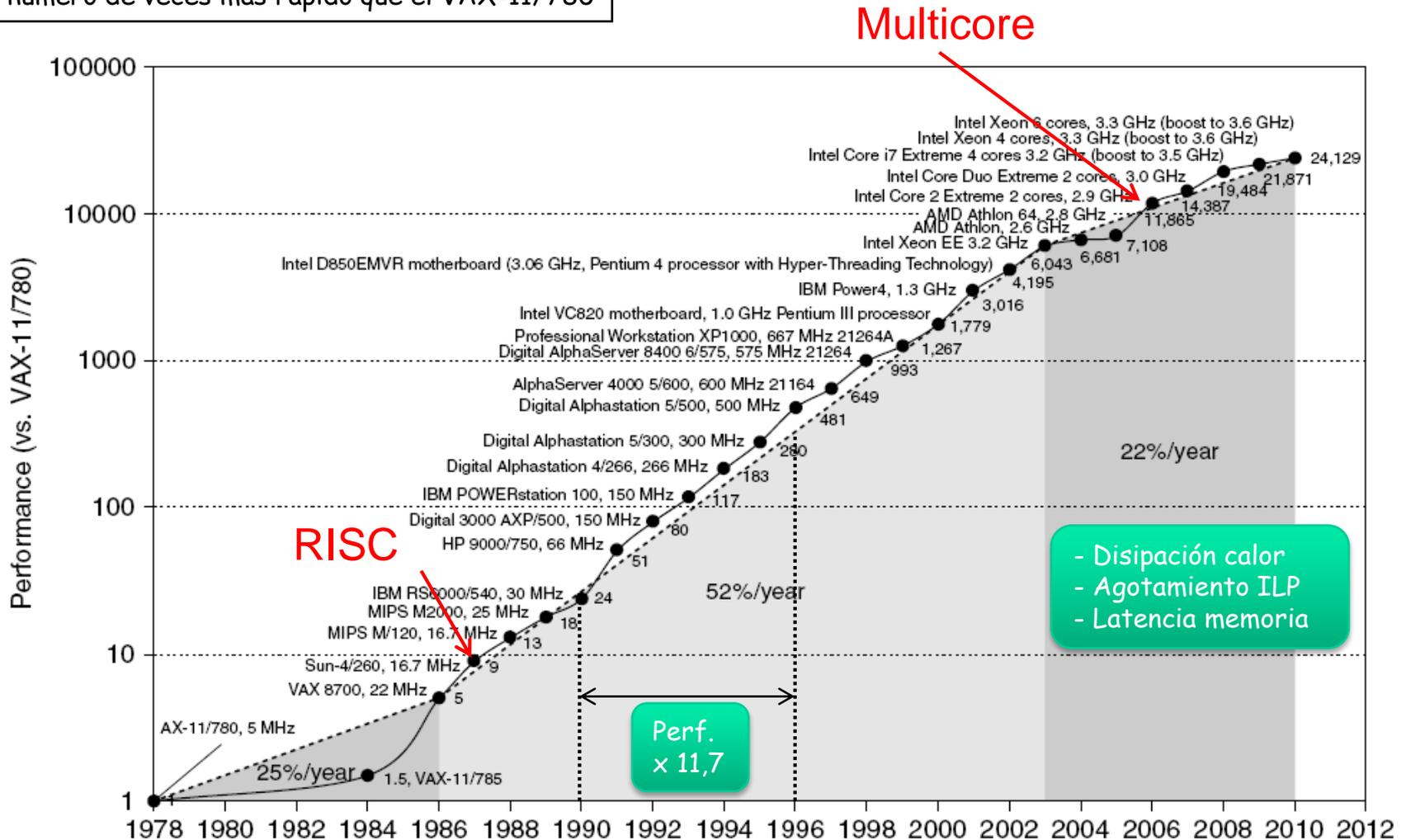
Fig 1.10 H&P 5th ed (detalle).

(\*) Tiempo de una op. sencilla, asumiendo que no hay contención

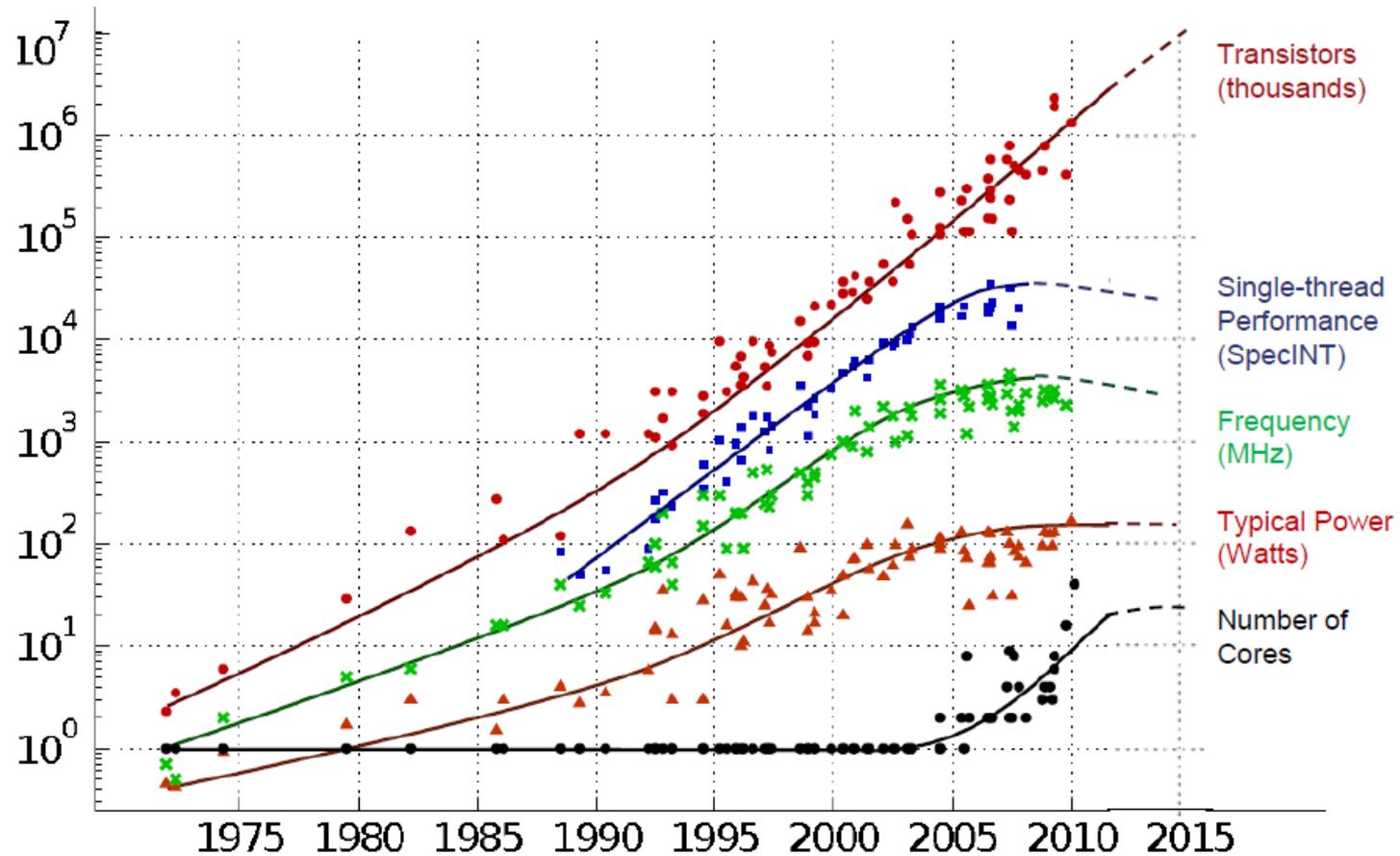
# Rendimiento

## Evolución del rendimiento de los procesadores

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



# 30 años de evolución



Original data collected and plotted by M. Horowitz, F. Labonte, O. Shacham, K. Olukotun, L. Hammond and C. Batten  
Dotted line extrapolations by C. Moore

## □ La Regla de Pollack (Intel)

o Regla empírica: El rendimiento obtenido con un conjunto de recursos,  $R$ , crece de forma proporcional a la raíz cuadrada de  $R$ .

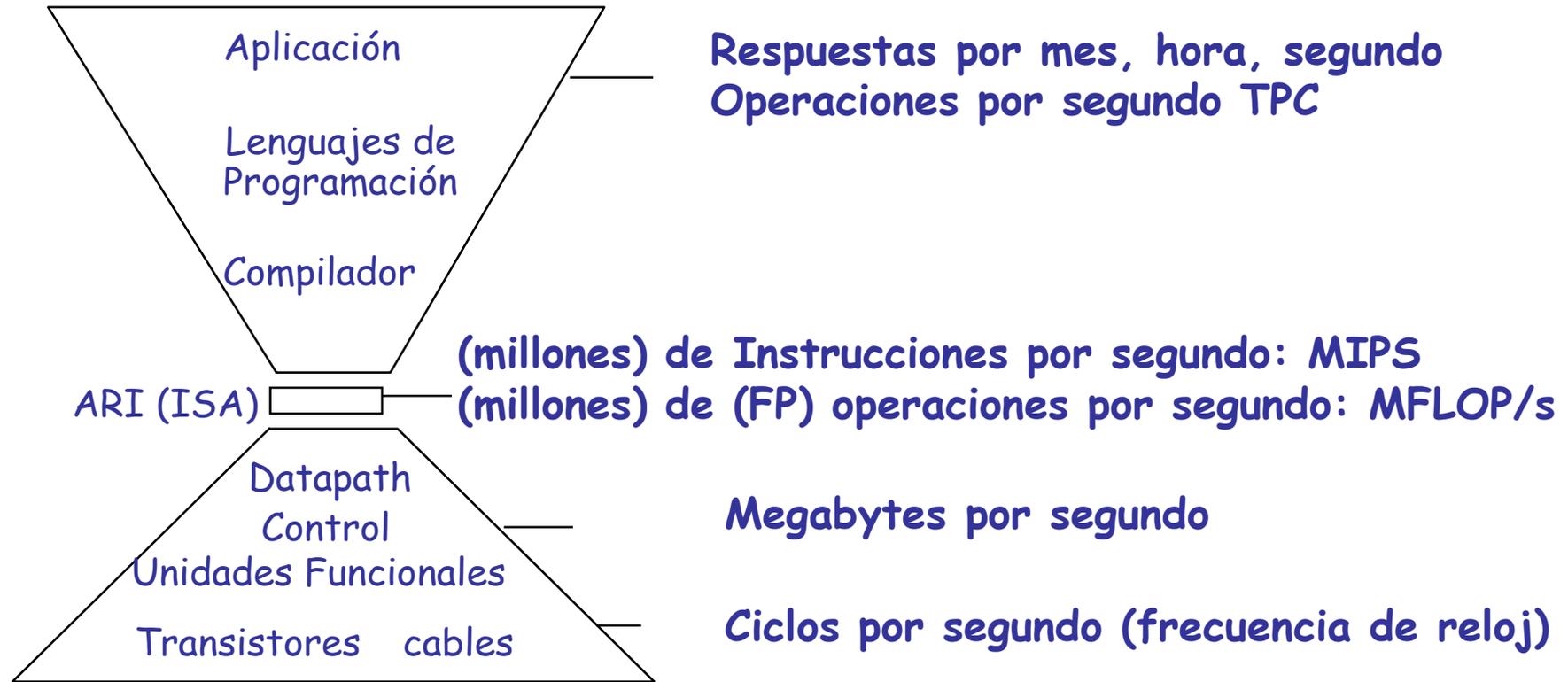
$$\text{Rend}(R) \sim \sqrt{R}$$

- Transistores/Consumo 2x → Rendimiento 1.4x
- Transistores/Consumo 4x → Rendimiento 2x



# Rendimiento

## Medidas del rendimiento



La única medida fiable es el tiempo de ejecución programas reales  
Dos aspectos: Rendimiento del procesador, Rendimiento del computador

# Rendimiento

## □ Rendimiento del procesador

$$T_{CPU} = N * CPI * t$$

- ✓ N: n° de instrucciones (Compiladores y LM)
- ✓ CPI: (LM, implementación, paralelismo)
- ✓ t: período de reloj (implementación, tecnología)

## □ Ciclos medios por instrucción (CPI)

$$\begin{aligned} CPI &= (T_{CPU} * \text{Frecuencia de reloj}) / \text{Numero de Instrucciones} \\ &= \text{Ciclos} / \text{Numero de Instrucciones} \end{aligned}$$

Si asumimos que existen n tipos de instrucciones:

$$T_{CPU} = t * \sum_{j=1}^n (CPI_j * I_j) \quad (I_j = n^{\circ} \text{ instrucciones tipo } j \text{ ejecutadas})$$

Dividiendo por (t.N)

$$CPI = \sum_{j=1}^n CPI_j * F_j \quad (\text{donde } F_j \text{ es la frecuencia de aparición de la instrucción tipo } j)$$

**Ejemplo :** ALU 1 ciclo( 50%), Ld 2 ciclos(20%), St 2 ciclos(10%), saltos 2 ciclos(20%)

$$CPI = 1*0,5 + 2*0,2 + 2*0,1 + 2*0,2 = 1.5$$

Invertir recursos donde se gasta el tiempo

## □ Rendimiento global del computador : Benchmarks

- ✓ La única forma fiable es ejecutando distintos programas reales.
  - ✓ Programas "de juguete": 10~100 líneas de código con resultado conocido. *Ej.: Criba de Erastótenes, 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) SPEC
- ✓ Otros
  - ✓ HPC: LINPACK, SPEChpc96, Nas Parallel Benchmark
  - ✓ Servidores: SPECweb, SPECsfs( File servers), TPC-C, SPECjbb ( Java )
  - ✓ Graficos: SPECviewperf(OpenGL), SPECapc( aplicaciones 3D)
  - ✓ Winbench, EEMBC

## □ Rendimiento global del computador : SPEC

- ✓ Programas reales típicos con cargas de trabajo fijas (actualmente la medida más aceptada)
  - ✓ **SPEC89**: 10 programas proporcionando un único valor.
  - ✓ **SPEC92**: 6 programas enteros (SPECint92) y 14 en punto flotante (SPECfp92). Sin límites en opciones de compilación
  - ✓ **SPEC95**: 8 programas enteros (SPECint95) y 10 en punto flotante (SPECfp95). Dos opciones en compilación: la mejor para cada programa y la misma en todos (base)
  - ✓ **SPEC2000** 12 programas enteros y 14 en punto flotante. Dos opciones de compilación ( la mejor: spec--, la misma spec--\_base
  - ✓ **SPEC2006** 12 programas enteros y 17 en punto flotante. Dos opciones de compilación ( la mejor: spec--, la misma spec--\_base

# Rendimiento

## ➤ SPEC2006 versus SPEC2000

Evolución de la jerarquía de memoria, ampliación de capacidad  
Más programas más complejos

Benchmark Description	CPU2000			CPU2006		
	Integer	Lng	RT	Integer	Lng	RT
GNU C compiler	176.gcc	C	1,100	403.gcc	C	8,050
Manipulates strings & prime numbers in Perl language	253.perlbnk	C	1,800	400.perlbench	C	9,766
Minimum cost network flow solver (combinatorial optimization)	181.mcf	C	1,800	429.mcf	C	9,120
Data compression utility	256.bzip2	C	1,500	401.bzip2	C	9,644
Data compression utility	164.gzip	C	1,400			
Video compression & decompression				464.h264ref	C	22,235
Artificial intelligence, plays game of Chess	186.crafty	C	1,000	458.sjeng	C	12,141
Artificial intelligence, plays game of Go				445.gobmk	C	10,489
Artificial intelligence used in games for finding 2D paths across terrains				473.astar	C++	7,017
Natural language processing	197.parser	C	1,800			
XML processing				483.xalancbnk	C++	6,869
FPGA circuit placement and routing	175.vpr	C	1,400			
EDA place and route simulator	300.twolf	C	3,000			
Search gene sequence				456.hmmer	C	9,333
Ray tracing	252.eon	C++	1,300			
Computational group theory	254.gap	C	1,100			
Database program	255.vortex	C	1,900			
Library for simulating a quantum computer				462.libquantum	C	20,704
Discrete event simulation				471.omnetpp	C++	6,270
	hours	5.3	19,100	hours	36.6	131,638

# Rendimiento

## ➤ SPEC2006 versus SPEC2000

Benchmark Description	CPU2000			CPU2006			
	Floating Pnt	Lng	RTime	Floating Pnt	Lng	RTime	
Weather prediction, shallow water model	171.swim	F77	3,100				
Velocity & distribution of pollutants based on temperature, wind	301.apsi	F77	2,600				
Weather modeling (30km area over 2 days)				481.wrf	C/F	11,215	
Physics, particle accelerator model	200.sixtrack	F77	1,100				
Parabolic/elliptic partial differential equations	173.applu	F77	2,100				
Multi-grid solver in 3D potential field	172.mgrid	F77	1,800				
General relativity, solves Einstein evolution equations				436.cactusADM	C/F	11,927	
Computational electromagnetics (solves Maxwell equations in 3D)				459.GemsFDTD	F	10,583	
Quantum chromodynamics	168.wupwise	F77	1,600				
Quantum chromodynamics, gauge field generation with dynamical quarks				433.milc	C	9,180	
Fluid dynamics, analysis of oscillatory instability	178.galgel	F90	2,900				
Fluid dynamics, computes 3D transonic transient laminar viscous flow				410.bwaves	F	13,592	
Computational fluid dynamics for simulation of astrophysical phenomena				434.zeusmp	F	9,096	
Fluid dynamics, large eddy simulations with linear-eddy model in 3D				437.leslie3d	F	9,358	
Fluid dynamics, simulates incompressible fluids in 3D				470.lbm	C	13,718	
Molecular dynamics (simulations based on newtonian equations of motion)				435.gromacs	C/F	7,132	
Biomolecular dynamics, simulates large system with 92,224 atoms				444.namd	C++	8,018	
Computational chemistry	188.amp	C	2,200				
Quantum chemistry package (object-oriented design)				465.tonto	F	9,822	
Quantum chemistry, wide range of self-consistent field calculations				416.gamess	F	19,575	
Computer vision, face recognition	187.facerec	F90	1,900				
Speech recognition system				482.sphinx3	C	19,528	
3D graphics library	177.mesa	C	1,400				
Neural network simulation (adaptive resonance theory)	179.art	C	2,600				
Earthquake modeling (finite element simulation)	183.quake	C	1,300				
Crash modeling (finite element simulation)	191.fma3d	F90	2,100				
Number theory (testing for primes)	189.lucas	F90	2,000				
Structural mechanics (finite elements for linear & nonlinear 3D structures)				454.calculix	C/F	8,250	
Finite element analysis (program library)				447.dealll	C++	11,486	
Linear programming optimization (railroad planning, airlift models)				450.soplex	C++	8,338	
Image ray tracing (400x400 anti-aliased image with abstract objects)				453.povray	C++	5,346	
		hours	8.0	28,700	hours	52	186,164

# Rendimiento

## ➤ Evolución de los SPEC

Year	Iteration	Suites	Languages	Measures	Reference Machine
1989	SPEC CPU	10 SPEC programs RT: 18.66 hours (scores not rounded)	C(4) & Fortran(5) & C/Fortran(1)	SPECmark SPECthruput	Vax 11/780 5 MHz 8K cache off-chip memory N/A
1992	SPEC CPU92	6 CINT92 programs RT: 6.21 hours 14 CFP92 programs RT: 41.27 hours (scores rounded to 10s place)	C(6)  C(2) & Fortran(12)	SPECint92 SPECfp92 SPECint_rate92 SPECfp_rate92	same as SPEC89
1995	SPEC CPU95	8 CINT95 programs RT: 5.25 hours 10 CFP95 programs RT: 11.00 hours (scores rounded to 100s place)	C(8)  Fortran(10)	SPECint95 SPECint_base95 SPECfp95 SPECfp_base95 SPECint_rate95 SPECint_rate_base95 SPECfp_rate95 SPECfp_rate_base95	SPARCstation 10/40 40 MHz SuperSPARC I 20K/16K I/D L1 on-chip no L2 cache 128MB memory
2000	SPEC CPU2000	12 CINT2000 programs RT: 5.31 hours 14 CFP2000 programs RT: 7.97 hours (scores rounded to 100s place)	C(11) & C++(1)  C(4) & Fortran77(6) & Fortran90(4)	same set of 8 measures defined for SPEC CPU95	Ultra 5 model 10 300 MHz UltraSPARC Iii 16K/16K I/D L1 on-chip 2MB L2 cache off-chip 256MB memory
2006	SPEC CPU2006	12 CINT2006 programs RT: 36.57 hours 17 CFP2006 programs RT: 51.71 hours (scores not rounded)	C(9) & C++(3)  C(3) & C++(4) & Fortran(6) & C/Fortran(4)	same set of 8 measures defined for SPEC CPU95	Ultra Enterprise 2 296 MHz UltraSPARC II 16K/16K I/D L1 on-chip 2MB L2 cache off-chip 1GB memory

## □ ¿Cómo se calculan los resultados en SPEC?

o Supongamos que la "suite" SPECxx está formada por n programas patrón

o Sea:

$$r_i = \frac{T \text{ ejecución patrón } i \text{ en máquina de referencia}}{T \text{ ejecución patrón } i \text{ en máquina evaluada}}, \forall i = 1, n$$

Entonces, se define

$$SPECxx = \sqrt[n]{r_1 \times r_2 \times \dots \times r_n}$$

o sea, la media geométrica (MG) de los n ratios.

# Rendimiento

## □ ¿Qué ventaja aporta el uso de la MG?

- o Ejemplo simplificado: Supongamos una suite con dos programas patrón y comparemos dos máquinas A y B

Tiempos de ejecución en cada computador y ratios

		Comp. Ref	Comp. A	Ratio	Comp. B	Ratio
Programas patrón	P1	R1	A1	R1/A1	B1	R1/B1
	P2	R2	A2	R2/A2	B2	R2/B2

$$\text{Rendimiento } A = \sqrt[2]{\frac{R1}{A1} \times \frac{R2}{A2}}$$

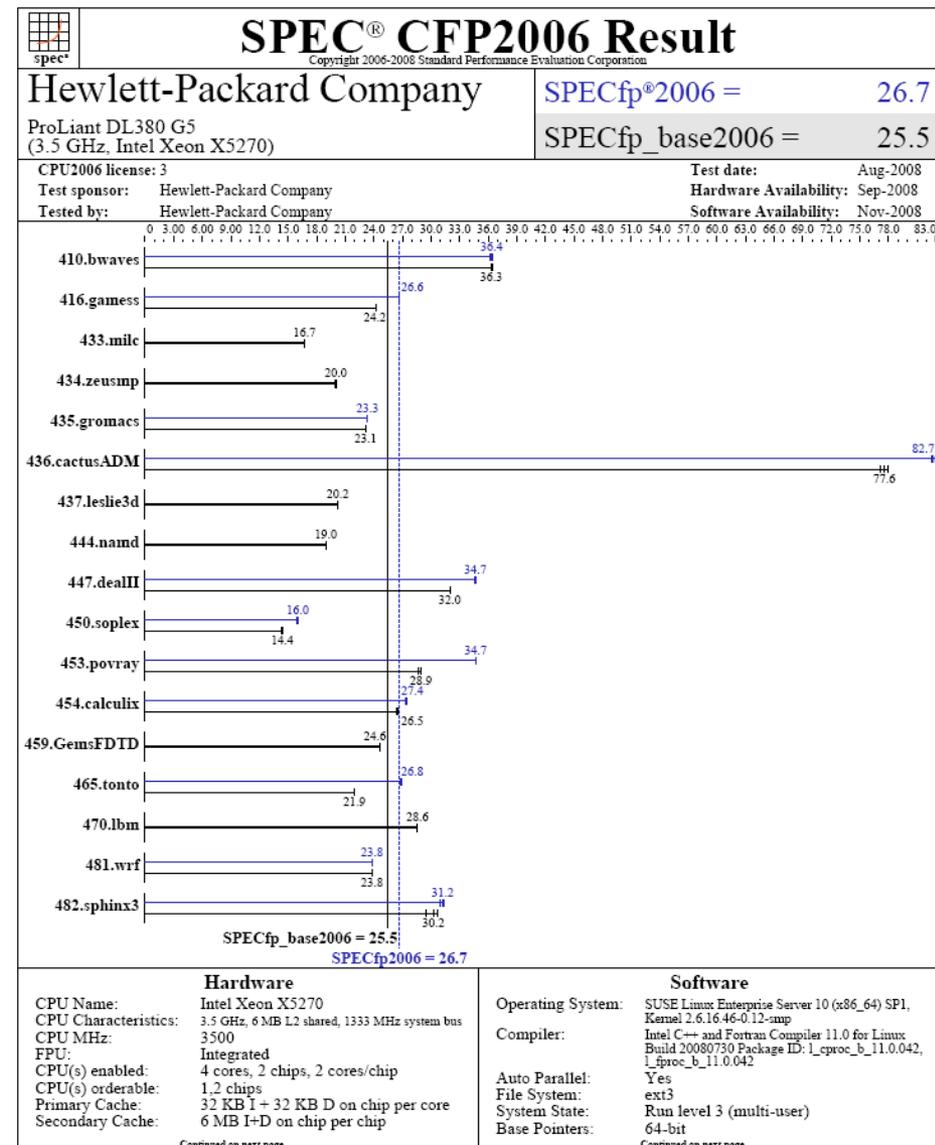
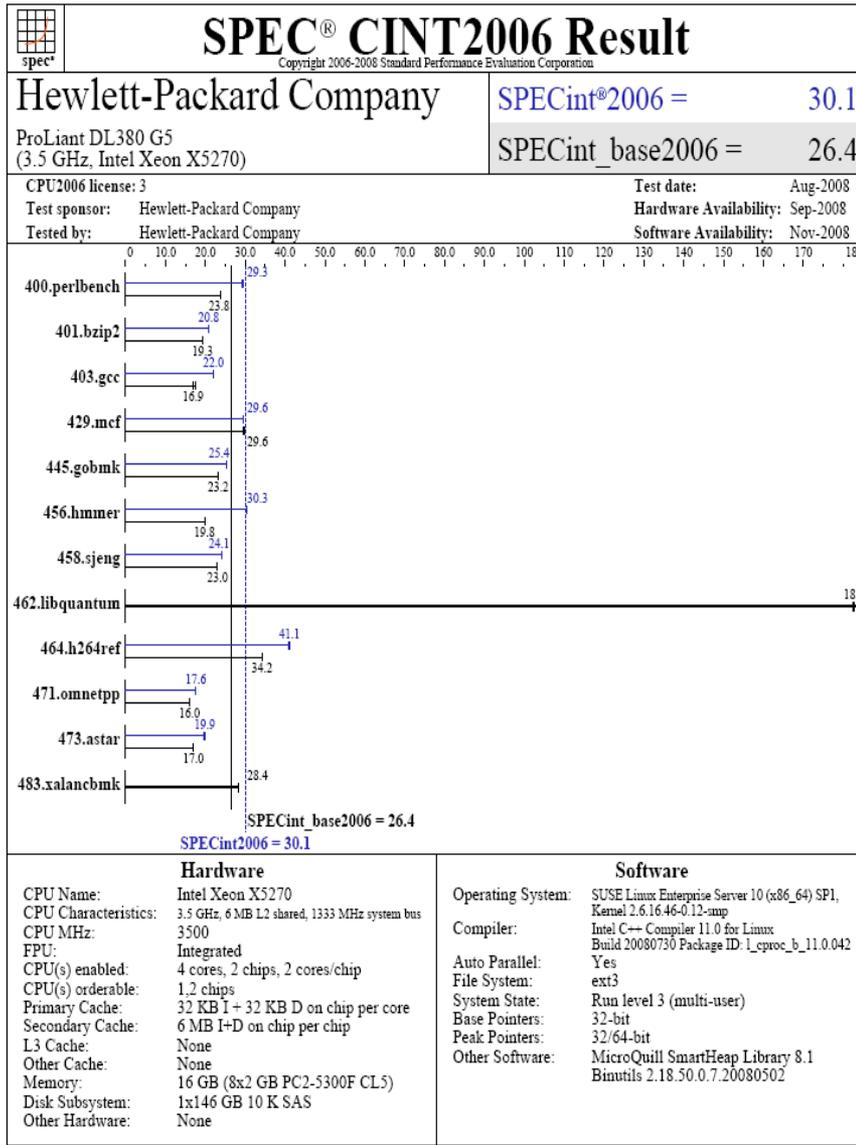
$$\text{Rendimiento } B = \sqrt[2]{\frac{R1}{B1} \times \frac{R2}{B2}}$$

- o Neutralidad de la máquina de referencia. La relación de rendimientos entre A y B es independiente de la máquina de referencia elegida. Efectivamente, operando:

$$\frac{\text{Rendimiento } A}{\text{Rendimiento } B} = \sqrt[2]{\frac{B1 \times B2}{A1 \times A2}} = \frac{\text{MG tiempos ejec en B}}{\text{MG tiempos ejec en A}}$$

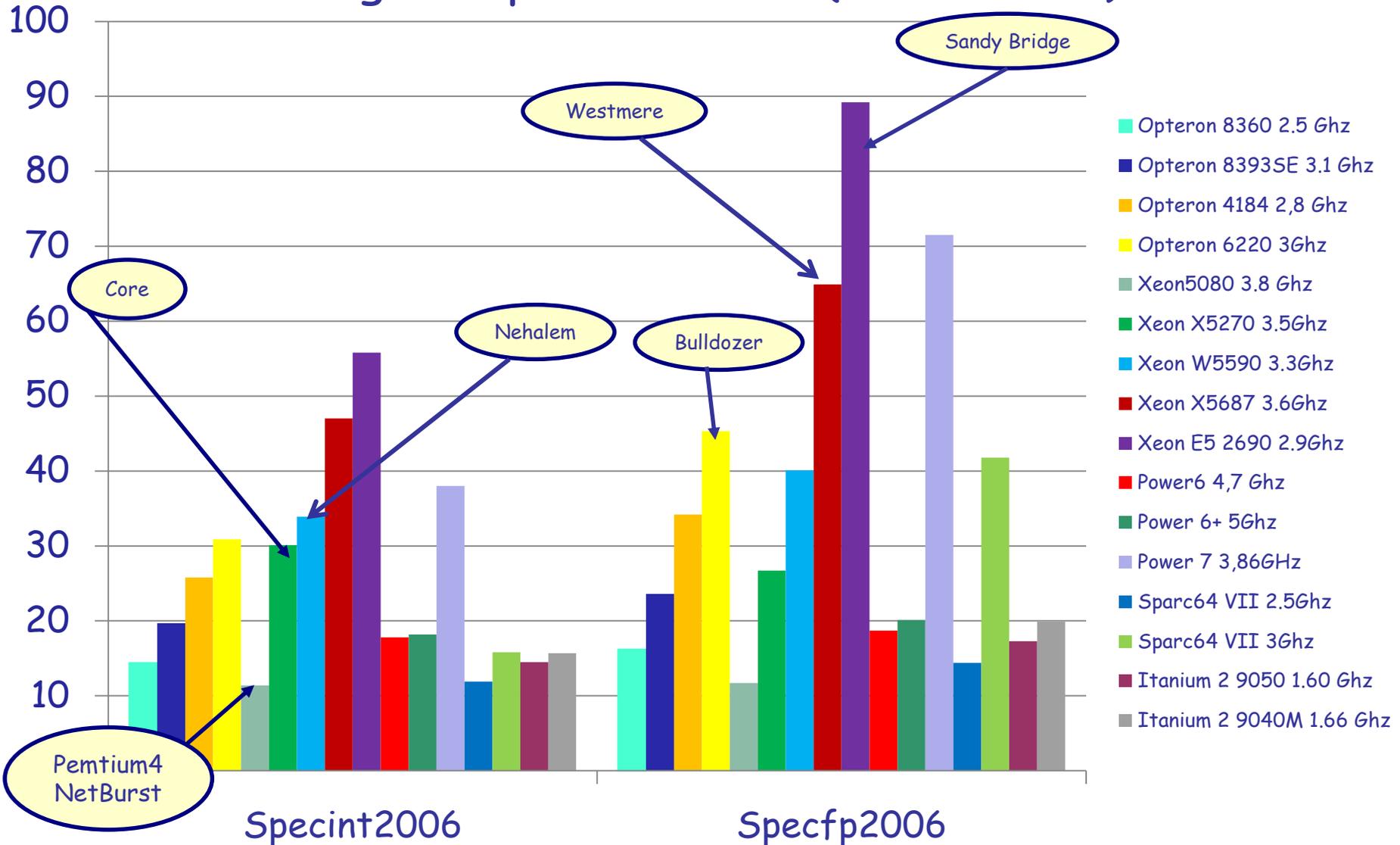
# Rendimiento

## ¿Cómo se presentan los resultados en SPEC?



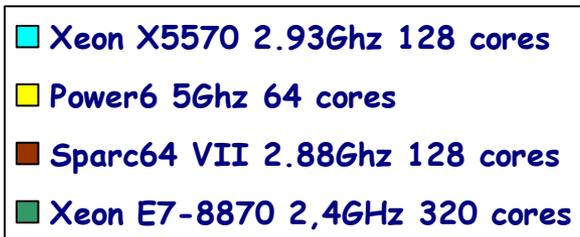
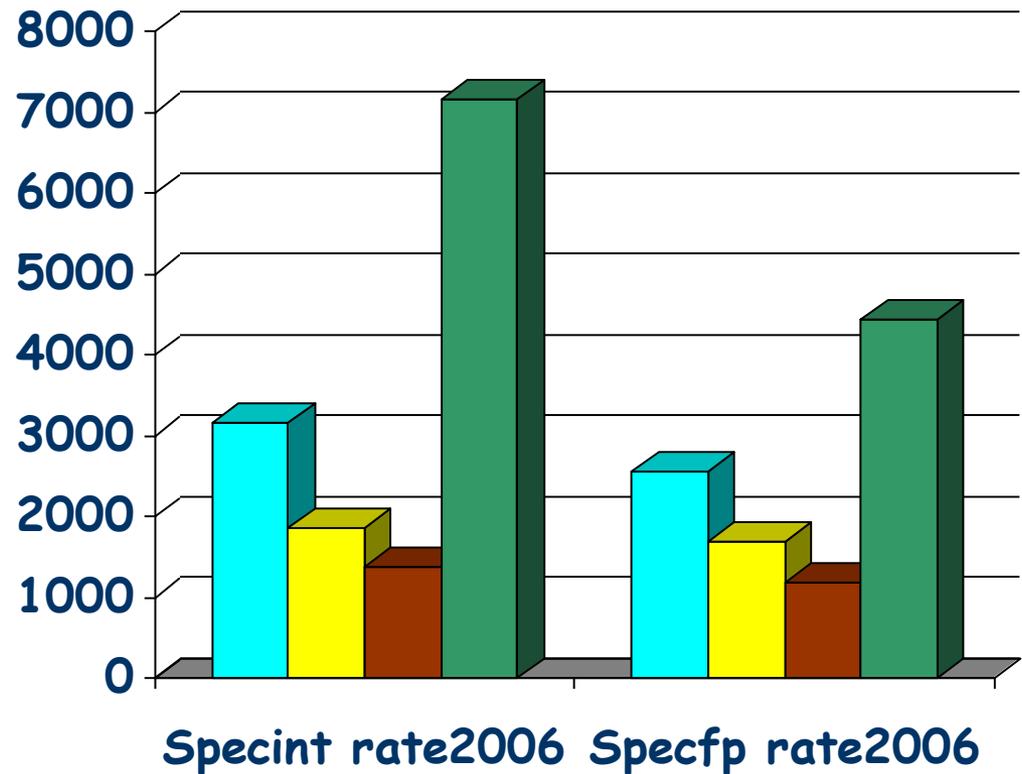
# Rendimiento

## □ SPEC de algunos procesadores( SPEC2006)



# Rendimiento

□ SPEC rate (\*) de sistemas de 32 chips ( SPEC2006)

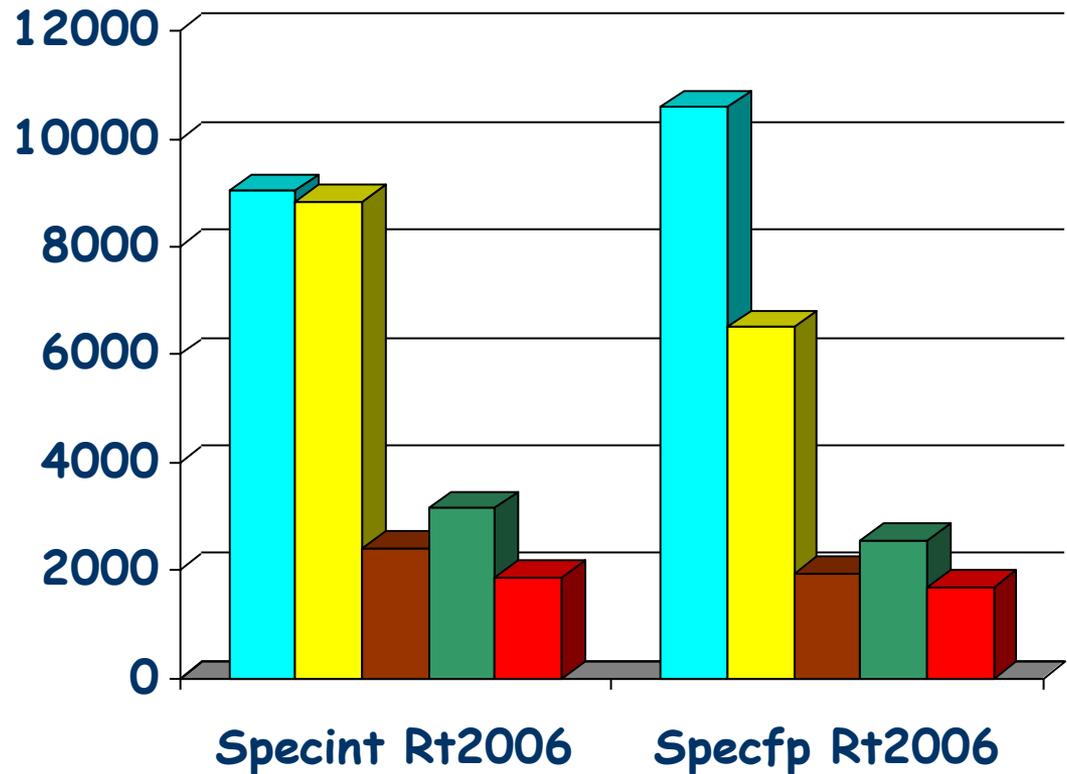


(\*) Analiza el rendimiento de una máquina en la ejecución de múltiples tareas a la vez. En la máquina evaluada se ponen en ejecución simultánea tantas copias del benchmark como cores existan

# Rendimiento

□ SPEC rate: comparación con sistemas mayores (SPEC2006)

■ Itanium 9040 1024 cores 512 chips
■ Opteron 8384 2.7Ghz 768 cores 192 chips
■ Sparc64 VII 2.88Ghz 256 cores 64 chips
■ Xeon X5570 2.93Ghz 128 cores 32 chips
■ Power6 5Ghz 64 cores 32 chips



# Rendimiento: Un principio simple

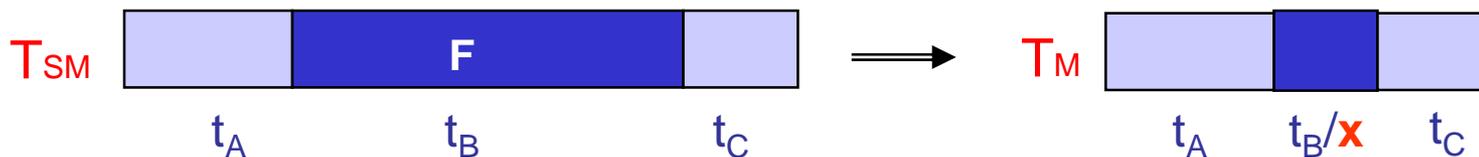
- Un principio básico: Hacer rápidas las funciones frecuentes.

“Gastar recursos donde se invierte la mayor cantidad de tiempo”

- **Ley de Amdahl:** Permite caracterizar este principio

Permite la evaluación del speedup que se obtendrá al aplicar una cierta mejora,  $M$ , que permite ejecutar una parte del código  $x$  veces más rápido.

**Def:** 
$$\text{Speedup}(E) = \frac{\text{TEj sin } M}{\text{TEj con } M} = \frac{\text{Performance con } M}{\text{Performance sin } M}$$



Si la mejora sólo acelera la ejecución de un fracción  $F$  de la tarea, el tiempo de ejecución del resto permanece sin modificación. Por tanto es muy importante el porcentaje de la tarea que es acelerada.

$$F = \frac{t_B}{t_A + t_B + t_C}$$

# Rendimiento: Un principio simple

## □ La Ley Amdahl

$$TE_{j_{nuevo}} = TE_{j_{antiguo}} \times \left[ (1 - \text{Fraccion}_{mejora}) + \frac{\text{Fraccion}_{mejora}}{X} \right]$$

$$\text{Speedup} = \left\{ \text{def} \right\} \frac{TE_{j_{antiguo}}}{TE_{j_{nuevo}}} = \frac{1}{(1 - \text{Fraccion}_{mejora}) + \frac{\text{Fraccion}_{mejora}}{X}}$$

**Ejemplo 1:** El 10% del tiempo de ejecución de mi programa es consumido por operaciones en PF. Se mejora la implementación de la operaciones PF reduciendo su tiempo a la mitad

$$TE_{j_{nuevo}} = TE_{j_{antiguo}} \times (0.9 + 0.1 / 2) = 0.95 \times TE_{j_{antiguo}} \quad \text{Speedup} = \frac{1}{0.95} = 1.053$$

Mejora de sólo un 5.3%

**Ejemplo 2:** Para mejorar la velocidad de una aplicación, se ejecuta una parte que consumía el 90% del tiempo sobre 100 procesadores en paralelo. El 10% restante no admite la ejecución en paralelo.

$$TE_{j_{nuevo}} = TE_{j_{antiguo}} \times (0.1 + 0.9 / 100) = 0.109 \times TE_{j_{antiguo}} \quad \text{Speedup} = \frac{1}{0.109} = 9.17$$

El uso de 100 procesadores sólo multiplica la velocidad por 9.17

# Rendimiento: Un principio simple

## □ Concepto de eficiencia (E)

$$E = \frac{\text{Speedup}}{x} = \frac{\frac{1}{(1-F) + \frac{F}{x}}}{x} = \frac{1}{x(1-F) + F} = \frac{1}{x + F(1-x)}$$

El valor máximo posible de E es 1 (para lo que se necesitaría que F=1)

## □ Ampliación del Ejemplo 2:

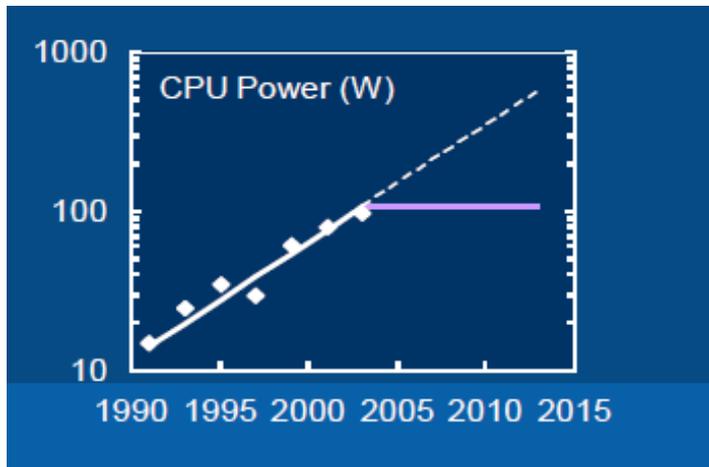
Procesadores (x)	F	Speedup	Eficiencia
10	0.9	5.26	0,526 (52.6%)
100	0.9	9.17	0,0917 (9.17%)
1000	0.9	9.91	0.00991 (0.99%)

### Observaciones:

1. La fracción no paralelizable de un cálculo, (1-F), limita seriamente el Speedup, incluso cuando esta fracción es pequeña.
2. A partir de cierto punto, aumentar mucho el n° de procesadores apenas mejora el Speedup, por lo que se degrada mucho la Eficiencia.

# Potencia y Energía

- ❑ El consumo de potencia eléctrica: uno de los principales retos en el diseño de computadores
- ❑ Problema: El consumo se transforma en calor a disipar
- ❑ La potencia debe ser similar en cada segmento
- ❑ ¿Cuánta potencia es preciso disipar?
  - o ¿Qué consumo de potencia puede mantenerse de forma sostenida? TDP (Thermal Design Power). Determina la potencia del sistema de refrigeración y de la fuente de alimentación
  - o Está por debajo de la potencia "pico" (peak) y por encima de la "media" (average)
- ❑ Cuál es la métrica más adecuada para comparar el rendimiento energético de dos procesadores? Energía vs. Potencia.



- ❑ Autonomía
- ❑ Tamaño compacto
- ❑ Coste y consumo del sistema de refrigeración

# Potencia y Energía

- ❑ Reto: "seguir incrementando en rendimiento sin incrementar el consumo"
- ❑ Tecnología. Transistores mas pequeños, rápidos y de menor consumo. Pero... mayor cantidad!
- ❑ Arquitectura;

o Potencia = Energía por instrucción x Instrucciones por segundo

o Reducir

$$EPI = E_{\text{dinámica}} + E_{\text{estática}}$$

Power gating:  
Suprimir alimentación

# Potencia y Energía

- ❑ Reducir EPI reduciendo  $E_{\text{dinámica}}$

$$E_{\text{dinámica}} \propto \frac{1}{2} VDD^2 * C_{\text{dyn}}$$

- Reducir VDD.

Impacto lineal sobre frecuencia y performance.

Cuadrático sobre el consumo de energía.

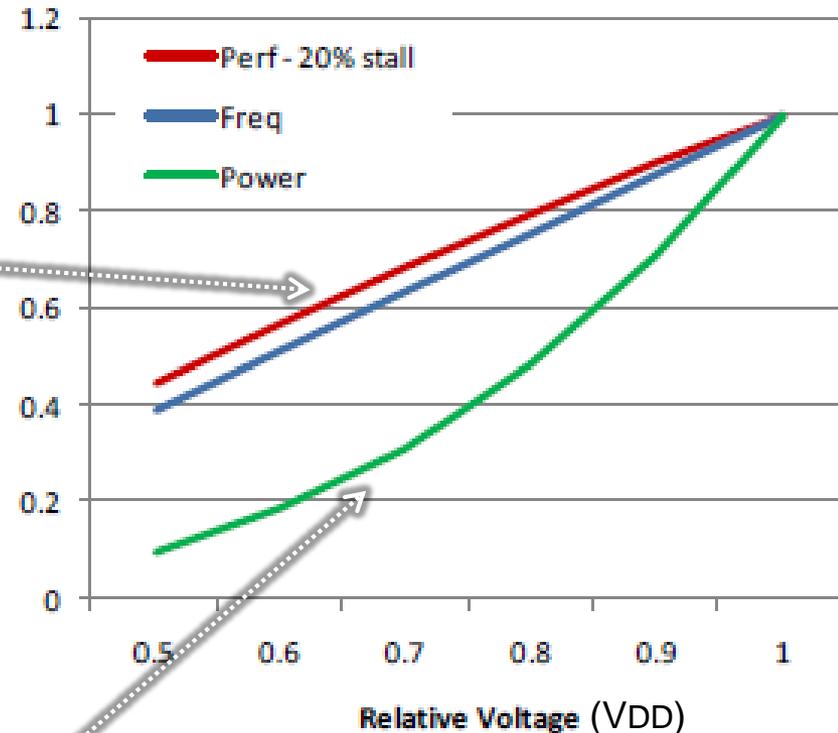
- Reducir  $C_{\text{dyn}}$  : Microarquitectura

- ❑ Potencia dinámica  $\propto \frac{1}{2} VDD^2 * C_{\text{dyn}} * \text{Frecuencia}$

Ojo! Reducir la frecuencia del reloj reduce la potencia, pero no la energía ( DVFS)

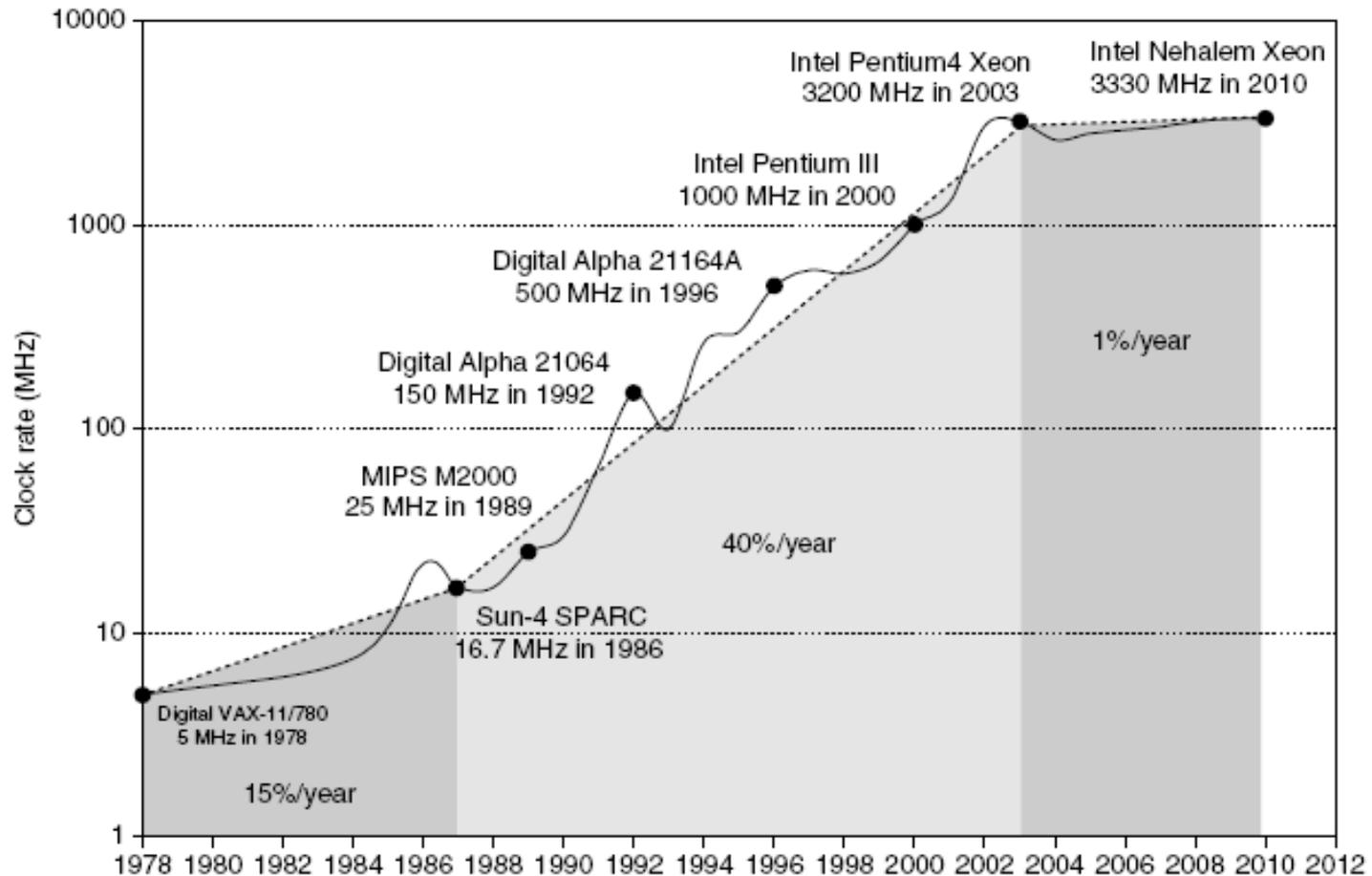
Reducción VDD (de 5V a <1V en 20 años):

- Impacto cúbico sobre la potencia dinámica: debido a la reducción adicional de la frecuencia máxima

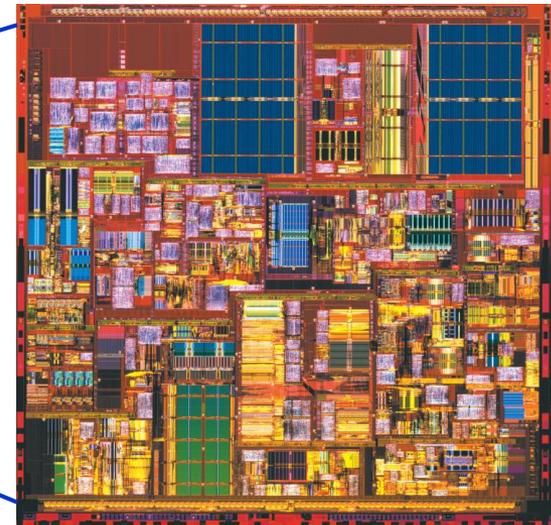
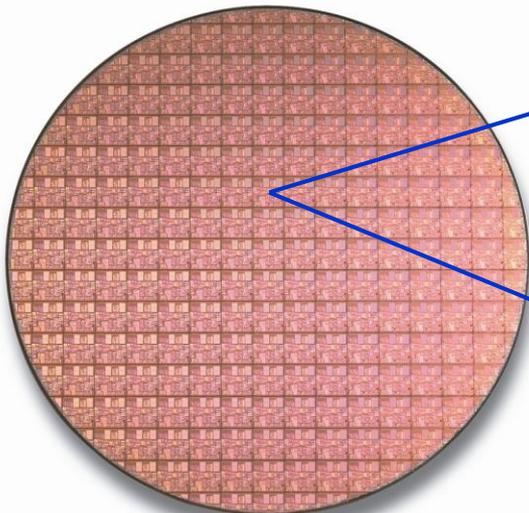
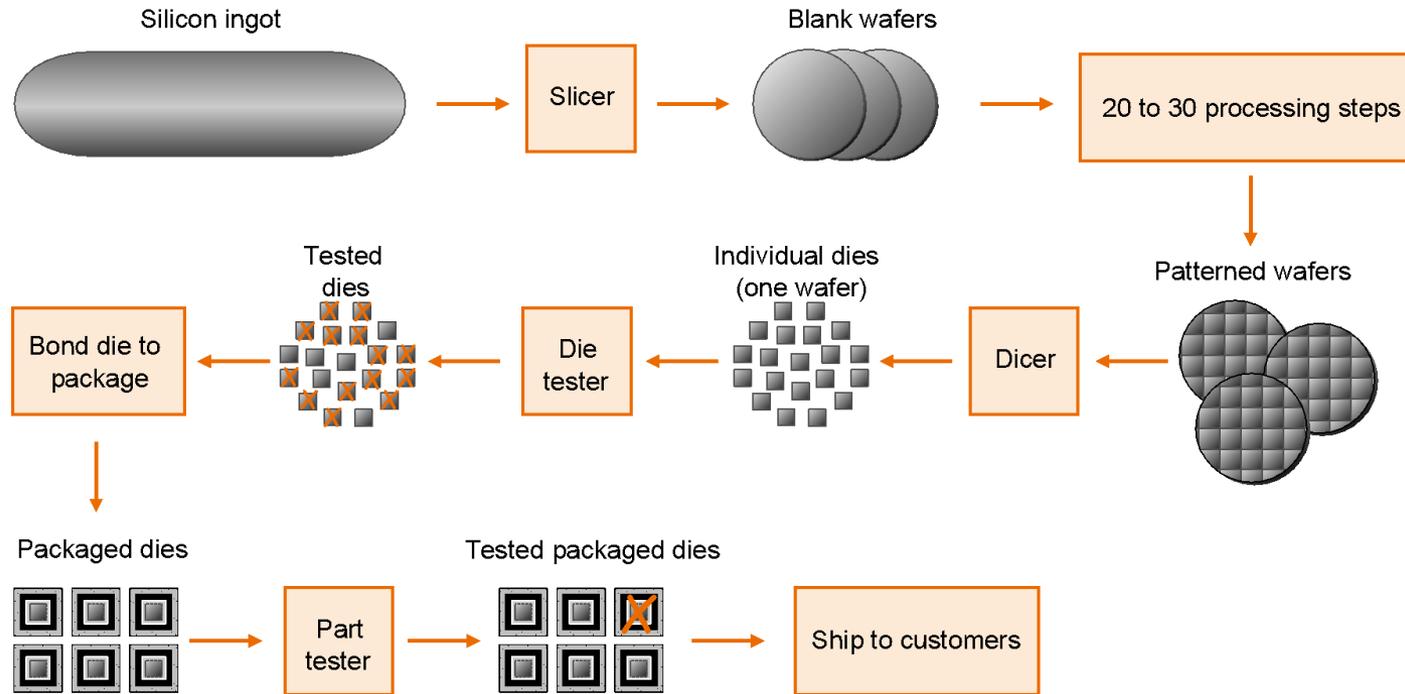


## □ Evolución del "clock"

Intel 80386 2W, Intel Core i7 3.3GHz, 130W.  
Limite para refrigeración con aire forzado.



# Coste: Fabricación de un CI



# Coste

□ Coste : El fundamental, el coste del CI

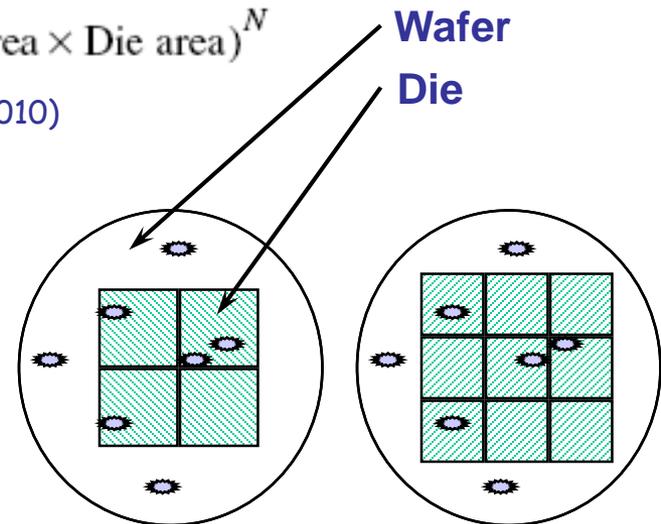
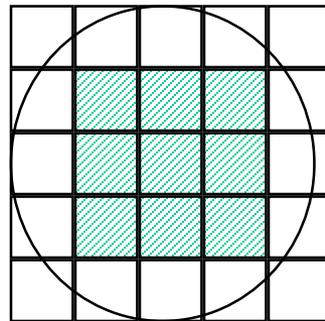
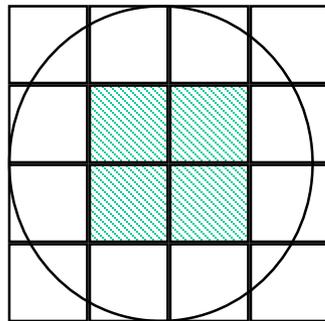
$$\text{coste de CI} = \frac{\text{Die coste} + \text{Testing coste} + \text{Packaging coste}}{\text{Final test yield}}$$

$$\text{Die coste} = \frac{\text{coste del Wafer}}{\text{Dies por Wafer} * \text{Die yield}}$$

Modelo empírico

$$\text{Die yield} = \text{Wafer yield} \times 1 / (1 + \text{Defects per unit area} \times \text{Die area})^N$$

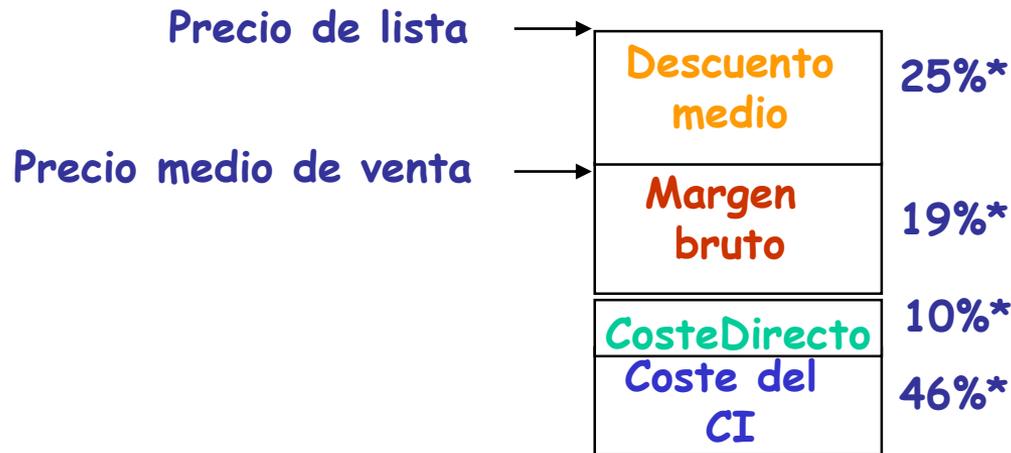
Defects per unit area = 0.016-0.057 defects per square cm (2010)  
N = process-complexity factor = 11.5-15.5 (40 nm, 2010)



**El coste de CI (Die)  $\approx f(\text{área del die})^2$**

## □ Componentes del coste final ( Precio )

- o Coste del CI
- o Costo Directo: costes recurrentes: mano de obra, compras,
- o Margen bruto: costes no recurrentes, I&D, marketing, ventas, equipamiento, costes financieros, beneficio, impuestos
- o Descuento



\* Valores para un PC típico a comienzos de la pasada década

- o Para sistemas grandes (e.g. data centers): Creciente importancia del coste de operación (alimentación, refrigeración, eficiencia energética)

# Coste-Rendimiento

## □ Tres servidores DELL PowerEdge

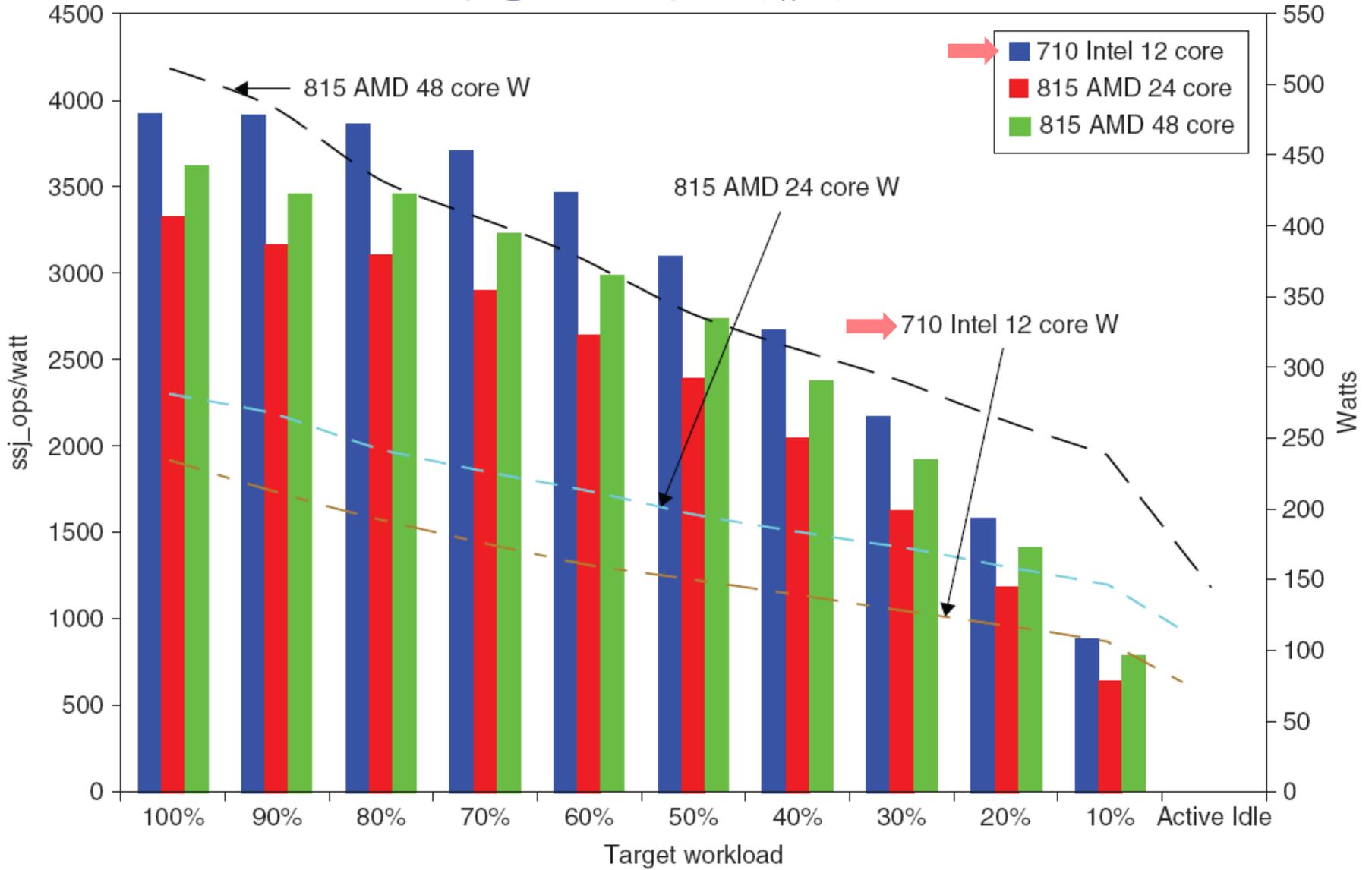
	System 1		System 2		System 3	
Component	Cost (% Cost)		Cost (% Cost)		Cost (% Cost)	
Base server	PowerEdge R710	\$653 (7%)	PowerEdge R815	\$1437 (15%)	PowerEdge R815	\$1437 (11%)
Power supply	570 W		1100 W		1100 W	
Processor	Xeon X5670	\$3738 (40%)	Opteron 6174	\$2679 (29%)	Opteron 6174	\$5358 (42%)
Clock rate	2.93 GHz		2.20 GHz		2.20 GHz	
Total cores	12		24		48	
Sockets	2		2		4	
Cores/socket	6		12		12	
DRAM	12 GB	\$484 (5%)	16 GB	\$693 (7%)	32 GB	\$1386 (11%)
Ethernet Inter.	Dual 1-Gbit	\$199 (2%)	Dual 1-Gbit	\$199 (2%)	Dual 1-Gbit	\$199 (2%)
Disk	50 GB SSD	\$1279 (14%)	50 GB SSD	\$1279 (14%)	50 GB SSD	\$1279 (10%)
Windows OS		\$2999 (32%)		\$2999 (33%)		\$2999 (24%)
Total		\$9352 (100%)		\$9286 (100%)		\$12,658 (100%)
Max ssj_ops *	910,978		926,676		1,840,450	Mejor rendimiento
Max ssj_ops/\$	97		100		145	Mejor rendimiento / \$

\* ssj\_ops: server side Java operations / s

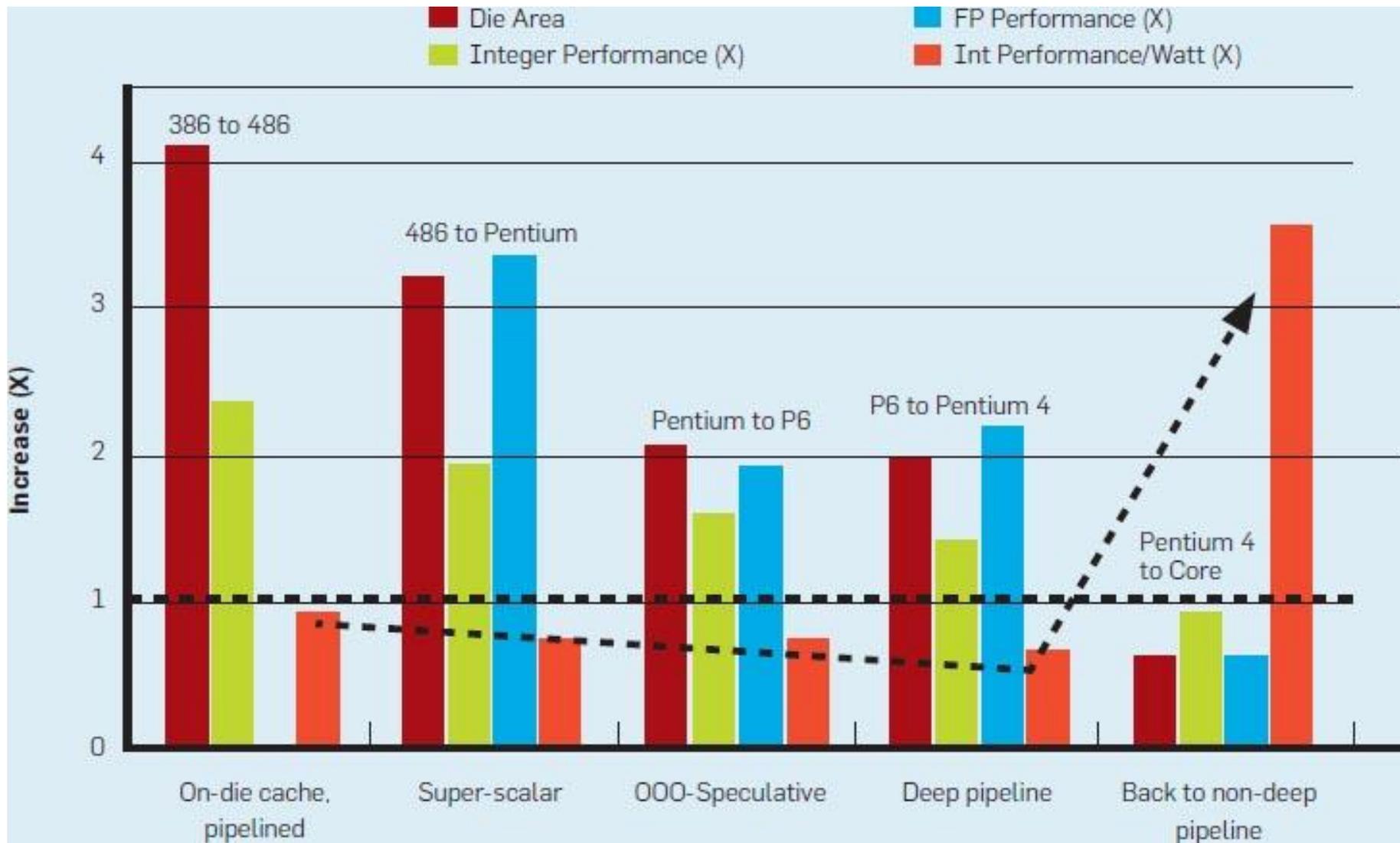
...¿Y el consumo? → SPECpower

# Coste-Rendimiento

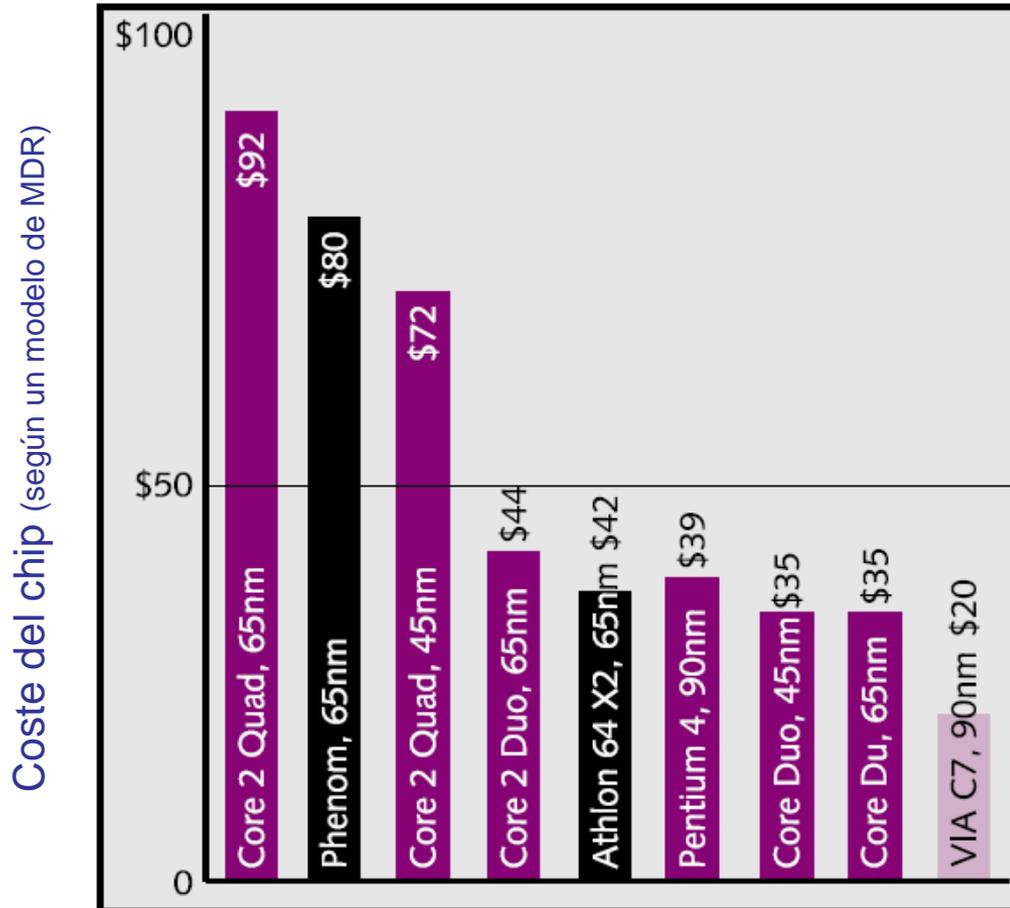
## SPEC Power Benchmark



# 30 años de evolución



## Algunos ejemplos reales



El precio de los microprocesadores depende del volumen. 10% de reducción cuando se dobla el volumen de producción



Fuente: Microprocessor Data Report, 2008