

Arquitectura de Computadores

Tema 1

Introducción: Tendencias Tecnológicas
Coste / Rendimiento/ Consumo



DEPARTAMENTO DE
ARQUITECTURA DE **C**OMPUTADORES
Y **A**UTOMÁTICA

Curso 2020-2021

- o Arquitectura del Computador
- o Evolución y tendencias
 - Procesador
 - Supercomputadores
 - Tecnología: Ley de Moore, binomio μ -electrónica – μ -arquitectura
- o Rendimiento
 - Métricas de evaluación
 - Un principio simple: Ley de Amdahl
- o Consumo
- o Coste
- o Bibliografía

Básica:

Capítulo 1 de Hennessy & Patterson 5th ed., 2012 (se usan algunas figuras de la sexta ed., 2017)

Complementaria:

500 most powerful computer systems. <http://www.top500.org>

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

Arquitectura del computador

❑ Visión “clásica”

- o Las decisiones de diseño se ocupan básicamente del repertorio de instrucciones (Instruction Set Architecture, ISA)
- o Decisiones relacionadas con:
 - registros, modos de direccionamiento, tipos de operandos (enteros, PF), tipos de instrucciones (aritméticas, transferencia, control de flujo...), codificación de instrucciones,...

❑ Visión “actual”

- o Punto de partida: Análisis de requisitos del computador objetivo → diseño para optimizar el rendimiento
 - Restricciones: coste, consumo de energía, tamaño, disponibilidad
- o Incluye (además de ISA)
 - **Microarquitectura** (u organización): estructura de la CPU, tipos de paralelismo soportado, modelo de ejecución de instrucciones, soporte de la jerarquía de memoria...
 - **Hardware**: Diseño lógico detallado, tecnología, ciclo de reloj...
- o El **papel del arquitecto**: Dominar un amplio campo de conocimientos
 - Desde compiladores y SO, hasta diseño VLSI y encapsulado de chips, pasando por todo tipo de técnicas para lograr alto rendimiento sin disparar el consumo.

Arquitectura del computador

ISA: sigue siendo un aspecto muy relevante

software



instruction set

μ -arquitectura
+ hardware



Visión de la
Arquitectura

❑ Propiedades de ISA

- o Permanencia con el tiempo / tecnología (portabilidad)
- o Proporciona funcionalidad eficaz a los niveles superiores
 - Las instrucciones ¿son realmente útiles para implementar programas?
- o Permite implementación eficiente en los niveles inferiores
 - Las instrucciones ¿se ejecutan eficientemente por el hw de procesador?

Arquitectura del computador

❑ Paralelismo

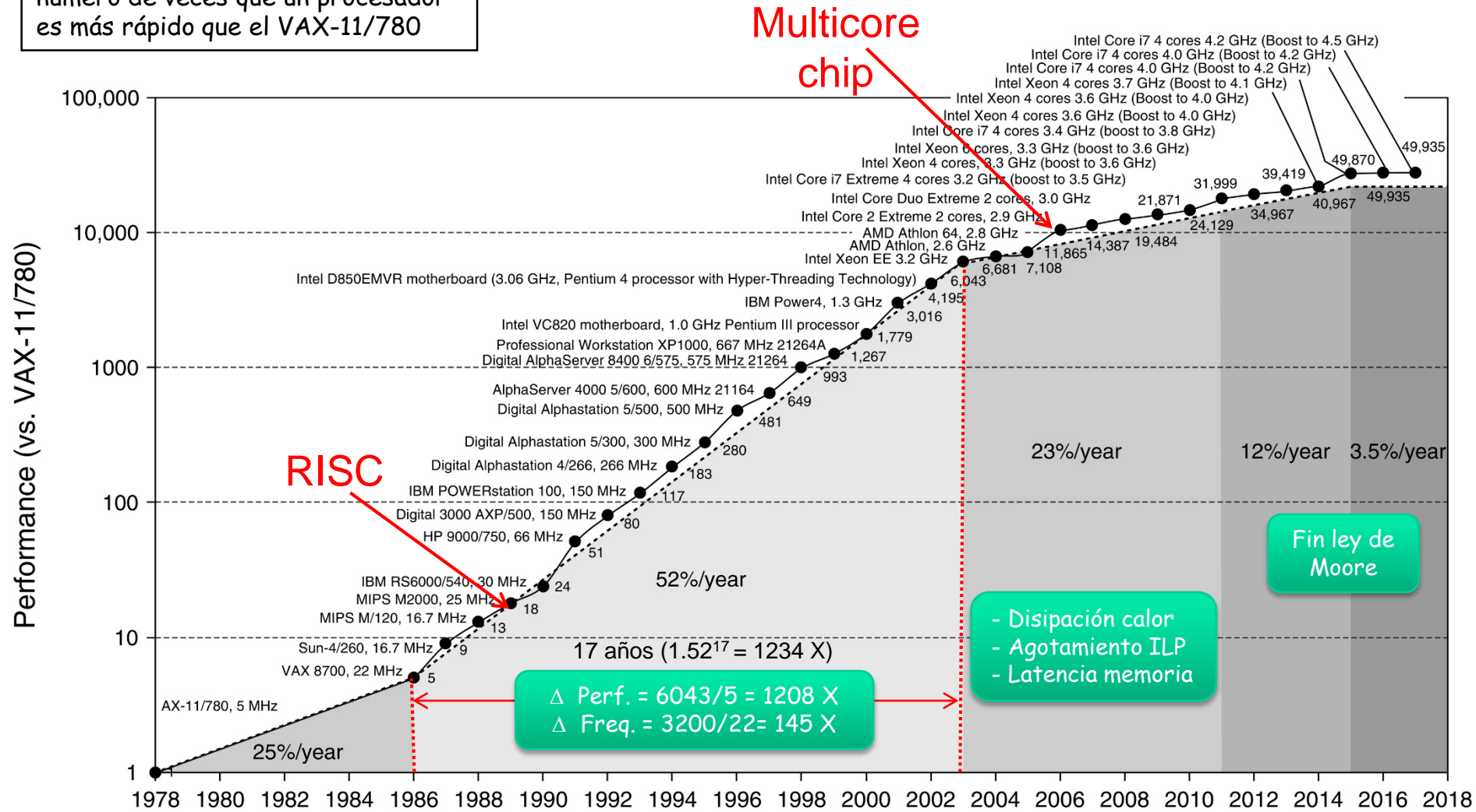
- o Propiedad inherente a las aplicaciones:
 - No todos los cálculos necesarios hay que hacerlos en un orden estricto: hay ordenaciones alternativas
 - No todos los cálculos deben hacerse en secuencia: algunos cálculos pueden hacerse simultáneamente.

❑ En Arquitectura se traduce en paralelismo...

- o a nivel de instrucciones (**Instruction-Level Parallelism, ILP**)
 - Ejecución de varias instrucciones a la vez, potencialmente en orden diferente al del programa.
 - Básicamente transparente al programador
- o a nivel de datos (**Data-Level Parallelism, DLP**)
 - Arquitectura vectorial, instrucciones multimedia, GPUs
 - Aplicación de una misma instrucción de LM a un conjunto de datos
- o a nivel de hebra (**Thread-Level Parallelism, TLP**)
 - Hebras de cálculos independientes que se ejecutan concurrentemente sobre los recursos hw disponibles (en uno o varios procesadores)
- o En este curso **tratamos los tres tipos**

Evolución y tendencias: rendimiento del procesador

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



Lectura recomendada: sección 1.1 de H&P 6th ed. (o 5th ed.)

Evolución y tendencias: supercomputadores

- ❑ Evolución del rendimiento de los computadores más potentes
- ❑ TOP500
 - o Proporciona una lista ordenada de sistemas de propósito general que se están usando para aplicaciones de alta demanda computacional (High Performance Computing, HPC).
 - Lista actualizada dos veces al año. Desde 1993!
 - o Se utiliza el test LINPACK
 - Resolver un sistema denso de n ecuaciones lineales.
 - n es seleccionado por el usuario (para obtener altos rendimientos n es típicamente > 1 millón !!)
 - Cualquier sistema específicamente diseñado para la ejecución de LINPACK queda descalificado en TOP500
 - o Rendimiento (R_{max}):
 - Op. en PF ejecutadas en test LINPACK / tiempo de ejecución: FLOPS
 - Lista ordenada por R_{max} en sentido decreciente
- ❑ GREEN500
 - o Publicada desde 2007, ordena los sistemas en sentido decreciente de eficiencia energética:
 - Eficiencia: $R_{max} / \text{Potencia consumida}$ (unidad habitual GFLOPS/W, equivale a GFLOP/J)

Evolución y tendencias: supercomputadores

TOP500: junio 2018

#1 de lista: 122.3 PFLOPS

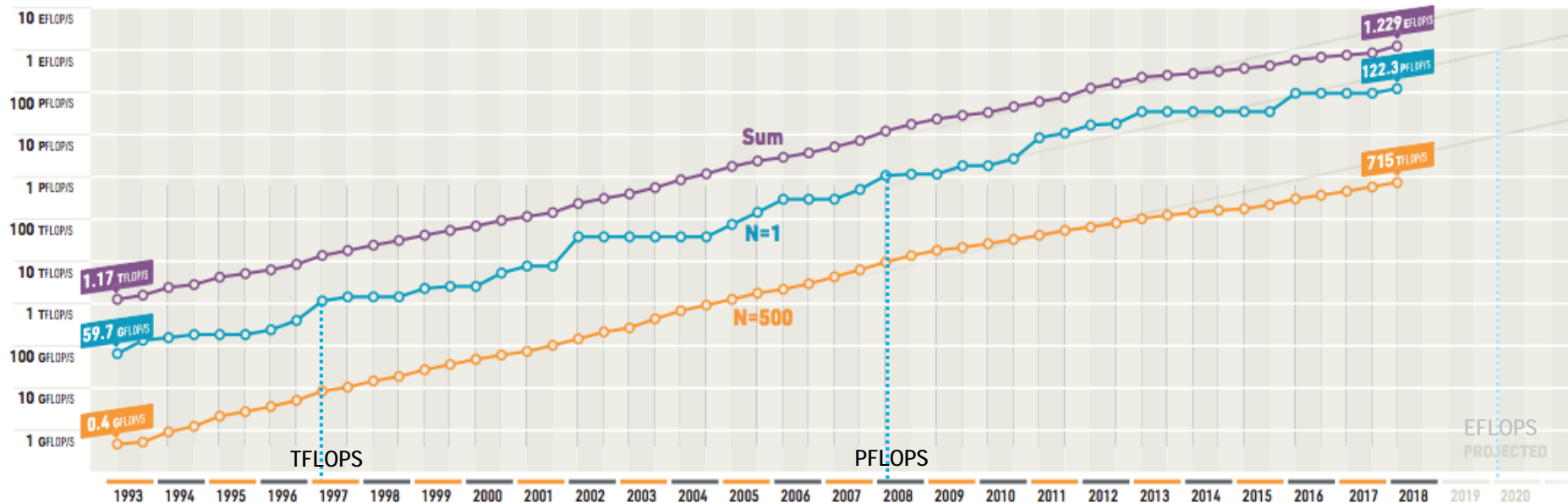
#500 de lista: 0.715 PFLOPS

Capacidad agregada de los 500 : 1229 PFLOPS



	SPECS	SITE	COUNTRY	CORES	RMAX PFLOPS	POWER MW
1	Summit IBM POWER9 (22C, 3.07GHz), NVIDIA Volta GV100 (80C), Dual-rail Mellanox EDR Infiniband	DOE/SC/ORNL	USA	2,282,544	122.3	8.8
2	Sunway TaihuLight Shenwei SW26010 (260C 1.45 GHz) Custom interconnect	NSCC in Wuxi	China	10,649,600	93.0	15.4
3	Sierra IBM POWER9 (22C, 3.1GHz), NVIDIA Tesla V100 (80C), Dual-rail Mellanox EDR Infiniband	DOE/NNSA/LLNL	USA	1,572,480	71.6	
4	Tianhe-2A (Milkyway-2A) Intel Ivy Bridge (12C 2.2 GHz) & TH Express-2, Matrix-2000	NSCC Guangzhou	China	4,981,760	61.4	18.5
5	AI Bridging Cloud Infrastructure PRIMERGY CX2550 M4, Xeon Gold 6148 (20C 2.4GHz), NVIDIA Tesla V100 (80C) SXM2, Infiniband EDR	AIST	Japan	391,680	19.9	1.6

PERFORMANCE DEVELOPMENT



Evolución y tendencias: supercomputadores

TOP500: junio 2019 (diferencias en 1 año)

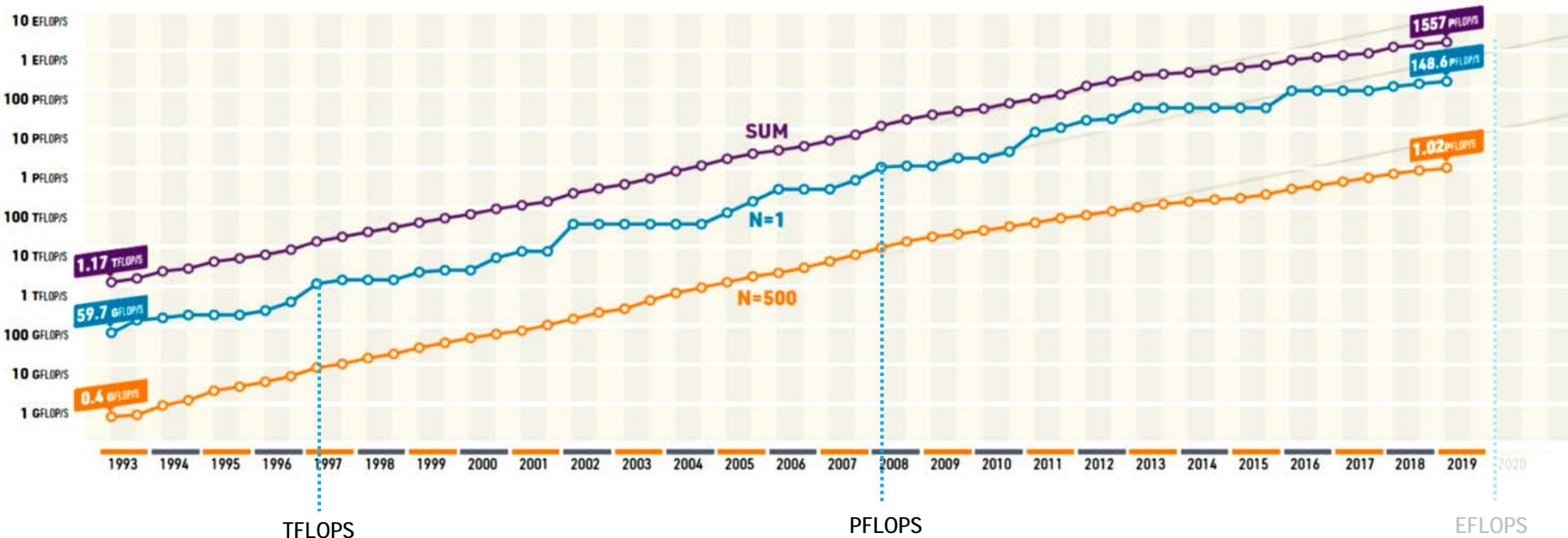
#1 de lista: 148.6 PFLOPS (+26.3) #500 de lista: 1.02 PFLOPS (+0.305)

Capacidad agregada de los 500 : 1557 PFLOPS (+328)



		SPECS	SITE	COUNTRY	CORES	RMAX PFLOPS	POWER MW
1	Summit	IBM POWER9 (22C, 3.07GHz), NVIDIA Volta GV100 (80C), Dual-Rail Mellanox EDR Infiniband	DOE/SC/ORNL	USA	2,414,592	148.6	10.1
2	Sierra	IBM POWER9 (22C, 3.1GHz), NVIDIA Tesla V100 (80C), Dual-Rail Mellanox EDR Infiniband	DOE/NNSA/LLNL	USA	1,572,480	94.6	7.44
3	Sunway TaihuLight	Shenwei SW26010 (260C, 1.45 GHz) Custom Interconnect	NSCC in Wuxi	China	10,649,600	93.0	15.4
4	Tianhe-2A (Milkyway-2A)	Intel Ivy Bridge (12C, 2.2 GHz) & TH Express-2, Matrix-2000	NSCC Guangzhou	China	4,981,760	61.4	18.5
5	Frontera	Dell C6420, Xeon Platinum 8280 28C 2.7GHz, Mellanox InfiniBand HDR	TACC/U of Texas	USA	448,448	23.5	-

PERFORMANCE DEVELOPMENT



Evolución y tendencias: supercomputadores

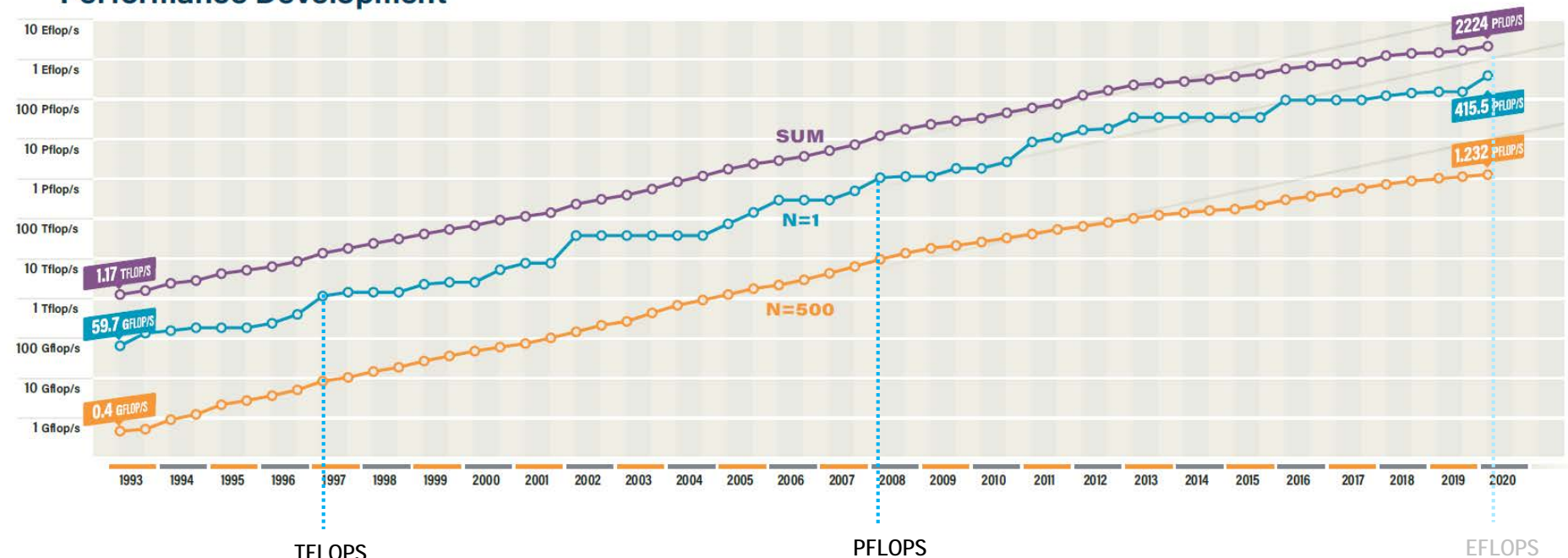
TOP500: junio 2020 (diferencias en 1 año)

#1 de lista: 415.5 PFLOPS (+266.9 !!!) #500 de lista: 1,23 PFLOPS (+0.21)

Capacidad agregada de los 500 : 2224 PFLOPS (+667 !!)

JUNE 2020	SYSTEM	SPECS	SITE	COUNTRY	CORES	RMAX PFLOP/S	POWER MW
1	Fugaku	Fujitsu A64FX (48C, 2.2GHz), Tofu Interconnect D	RIKEN R-CCS	Japan	7,288,072	415.5	28.3
2	Summit	IBM POWER9 (22C, 3.07GHz), NVIDIA Volta GV100 (80C), Dual-Rail Mellanox EDR Infiniband	DOE/SC/ORNL	USA	2,414,592	148.6	10.1
3	Sierra	IBM POWER9 (22C, 3.1GHz), NVIDIA Tesla V100 (80C), Dual-Rail Mellanox EDR Infiniband	DOE/NNSA/LLNL	USA	1,572,480	94.6	7.4
4	Sunway TaihuLight	Shenwei SW26010 (260C, 1.45 GHz) Custom Interconnect	NSCC in Wuxi	China	10,649,600	93.0	15.4
5	Tianhe-2A (Milkyway-2A)	Intel Ivy Bridge (12C, 2.2 GHz) & TH Express-2, Matrix-2000	NSCC Guangzhou	China	4,981,760	61.4	18.5

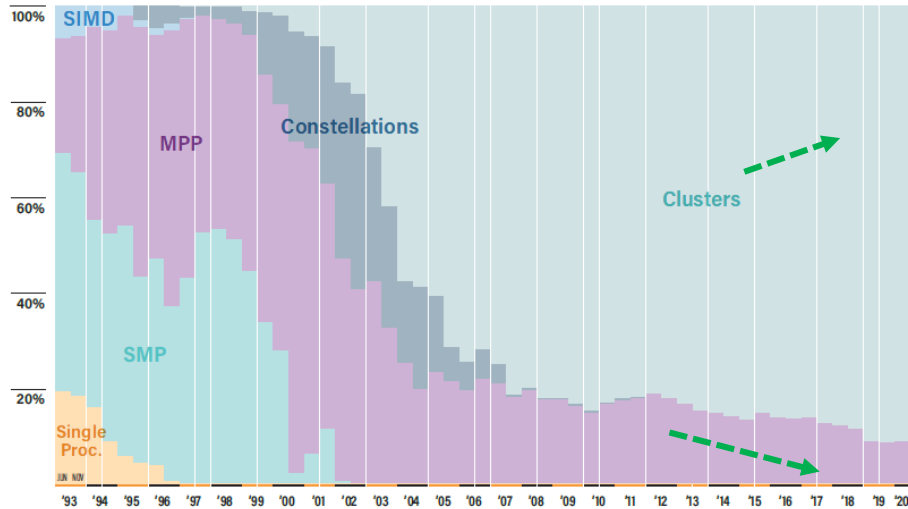
Performance Development



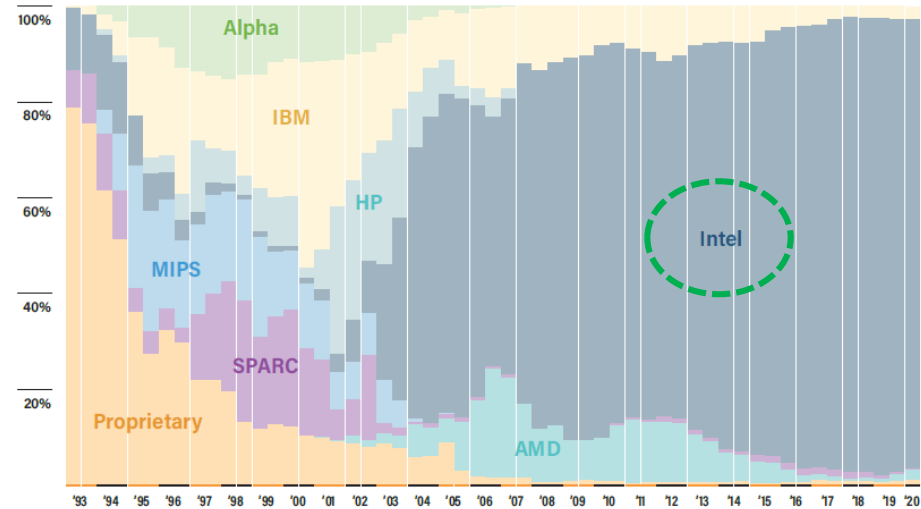
Evolución y tendencias: supercomputadores

TOP500: jun 2020. Actores principales.

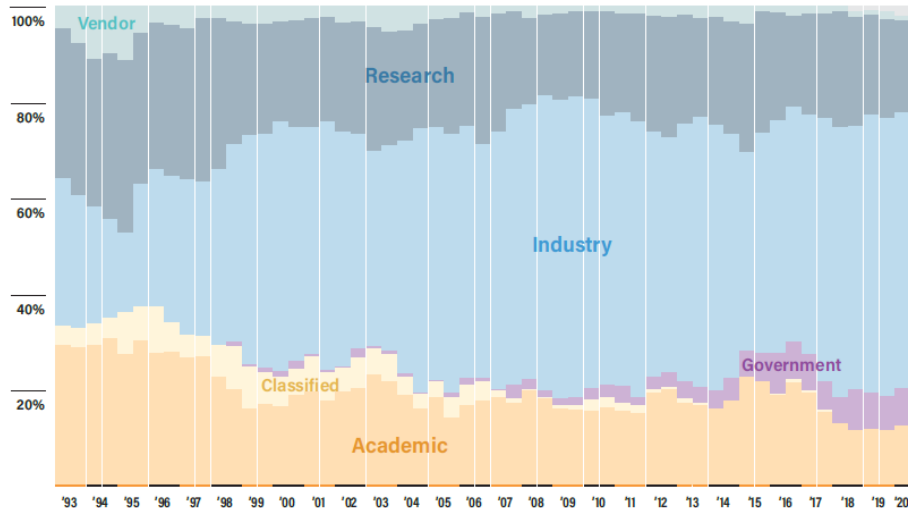
Architectures



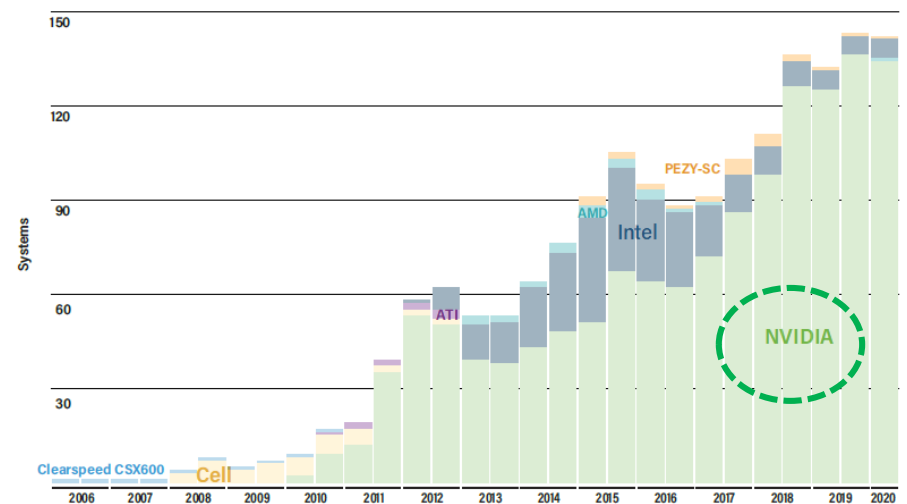
Chip Technology



Installation Type



Accelerators/Co-processors

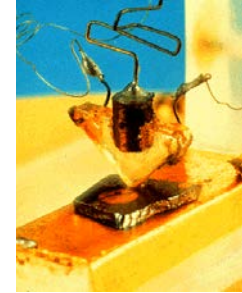


Evolución y tendencias: tecnología

❑ 1949 EDSAC 10^2 op/seg

❑ 1957 Transistor: de 10^3 a 10^4 op/s

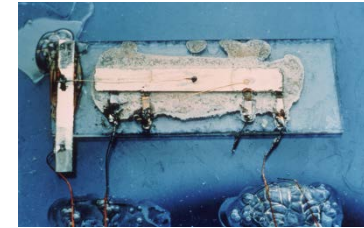
- o DEC PDP-1 (1957)
- o IBM 7090 (1960)



Transistor (47) PN 56

❑ 1965 CI: de 10^5 a 10^6 op/s

- o IBM System 360 (1965)
- o DEC PDP-8 (1965)



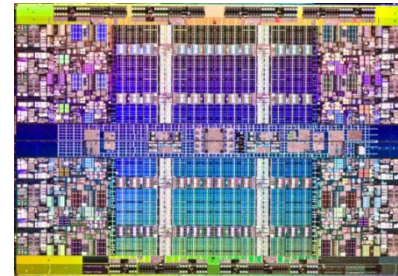
CI (58) PN2000

❑ 1971 Microprocesador

- o Intel 4004

❑ 2003 más de 3×10^{13} op/s

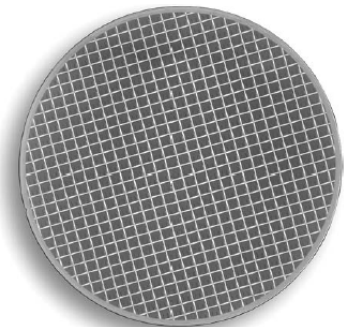
❑ 2008 $> 10^{15}$ FLOP/s (1 PFLOPS)



Intel Xeon 7500, 8c, 16Th

❑ Jun 2020 (TOP500)

- o 1º Fugaku:
7,299,072 cores; 415.5 PFLOPS (A64FX 48C 2.2 GHz)
- o 37º Mare Nostrum 4:
153,216 cores; 6.47 PFLOPS (Xeon Platinum 8160 24C 2.1GHz)



Oblea (Wafer)

Evolución y tendencias: tecnología

□ La Ley de Moore

Cramming More Components onto Integrated Circuits

GORDON E. MOORE, LIFE FELLOW, IEEE

Electronics - Abril 1965

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 various approaches.

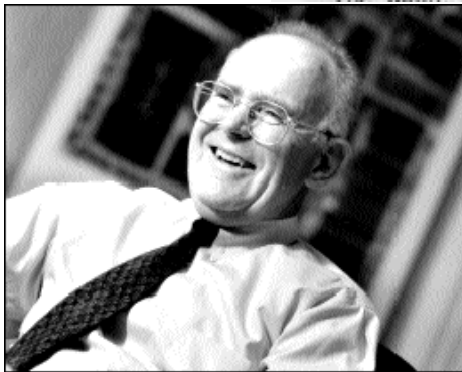


Fig. 2.

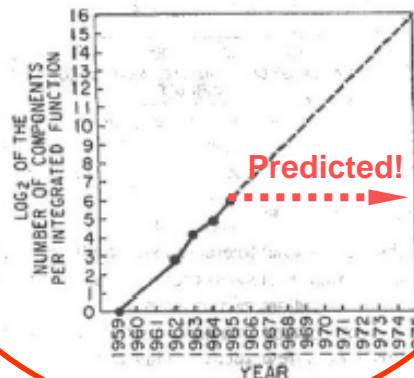


Fig. 3.

Carver A. Mead 1979

diagram to technological realization without any special engineering.

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

IX. LINEAR CIRCUITRY

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

Evolución y tendencias: tecnología

❑ Algunos aciertan... y mucho

The future of integrated electronics is the future of electronics itself. The advantages of integration will bring about a proliferation of electronics, pushing this science into many new areas.

Integrated circuits will lead to such wonders as home computers - or at least terminals connected to a central computer - automatic controls for automobiles, and personal portable communications equipment.

Cramming more components onto integrated circuits, Electronics, Volume 38, Number 8, April 19, 1965.

Gordon E. Moore, Co-founder Intel Corporation

Lectura recomendada: Artículos G Moore Campus Virtual.

Evolución y tendencias: tecnología

❑ Otros no tanto

"Hacer predicciones es muy difícil, especialmente sobre el futuro"

Niels Bohr? Samuel Goldwyn? K. K. Steincke? Robert Storm Petersen? Yogi Berra? Mark Twain? Nostradamus? Anonymous?

"No existe ninguna razón para tener un computador en casa".

Ken Olsen, 1977, co-fundador y CEO de DEC (Digital Equipment Corporation),

"Las máquinas serán capaces, en los próximos veinte años, de hacer cualquier trabajo que una persona pueda hacer"

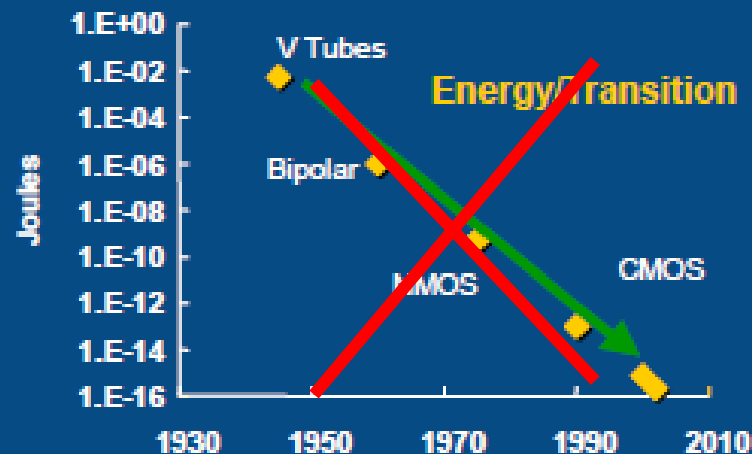
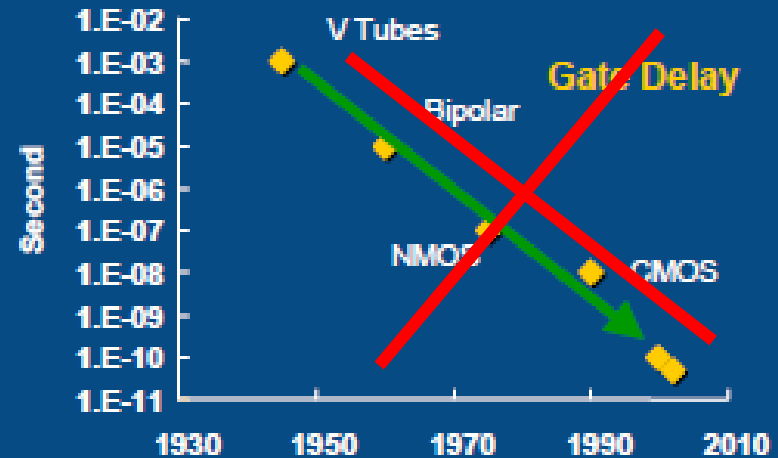
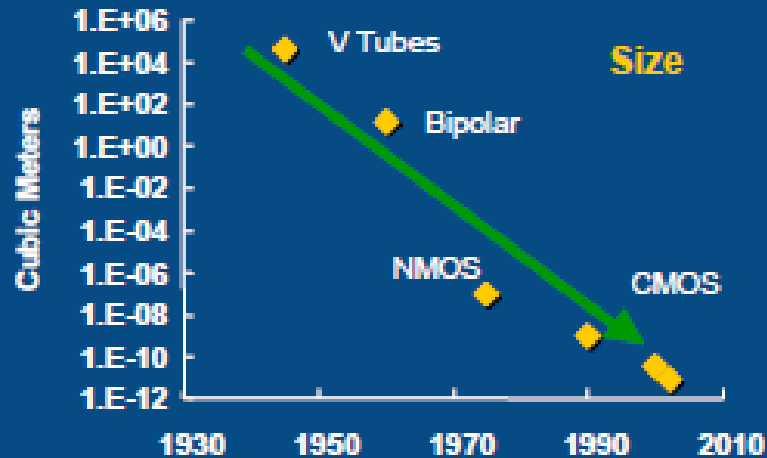
Herbert Simon, 1965, Premio Nobel Economía 1978, Premio Turing 1975

"La inteligencia artificial, ESTÁ AQUÍ" Portada del Business Week, July 9, 1984

Evolución y tendencias: tecnología

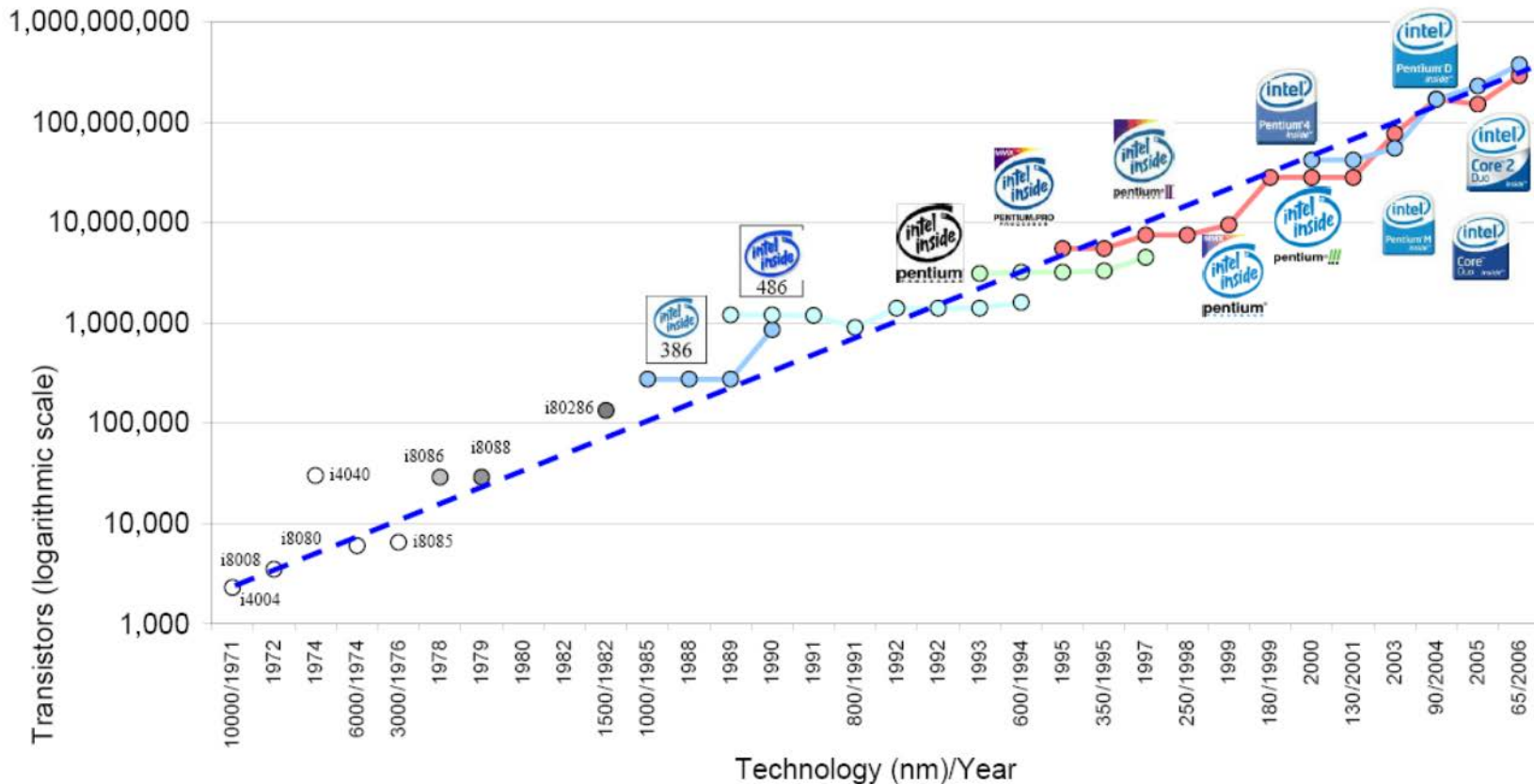
¿ El escalado de la tecnología continua ?

Después 2004



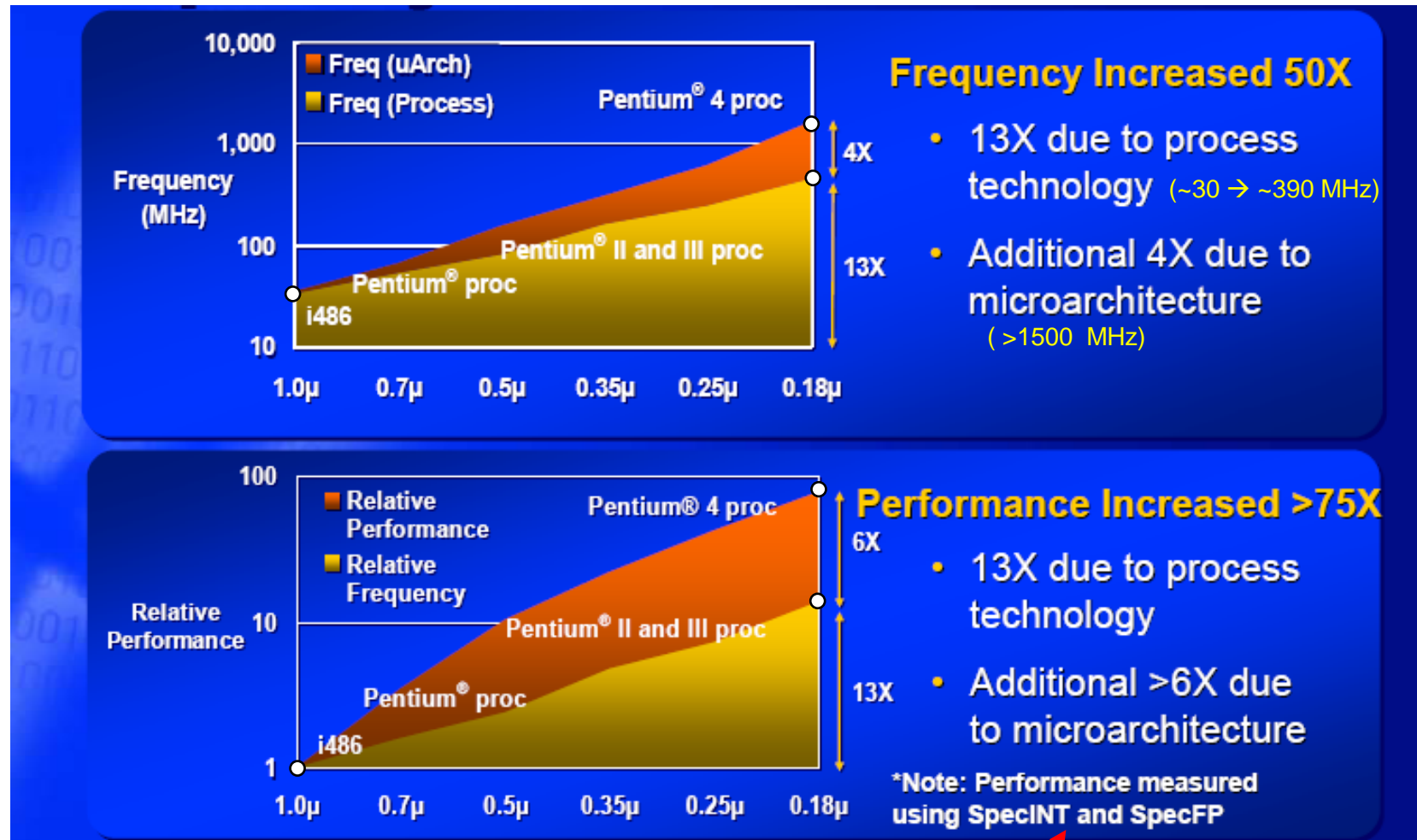
Evolución y tendencias: tecnología

- ❑ La Ley de Moore se ha cumplido...por mucho más tiempo del predicho



Fuente: Intel Corporation

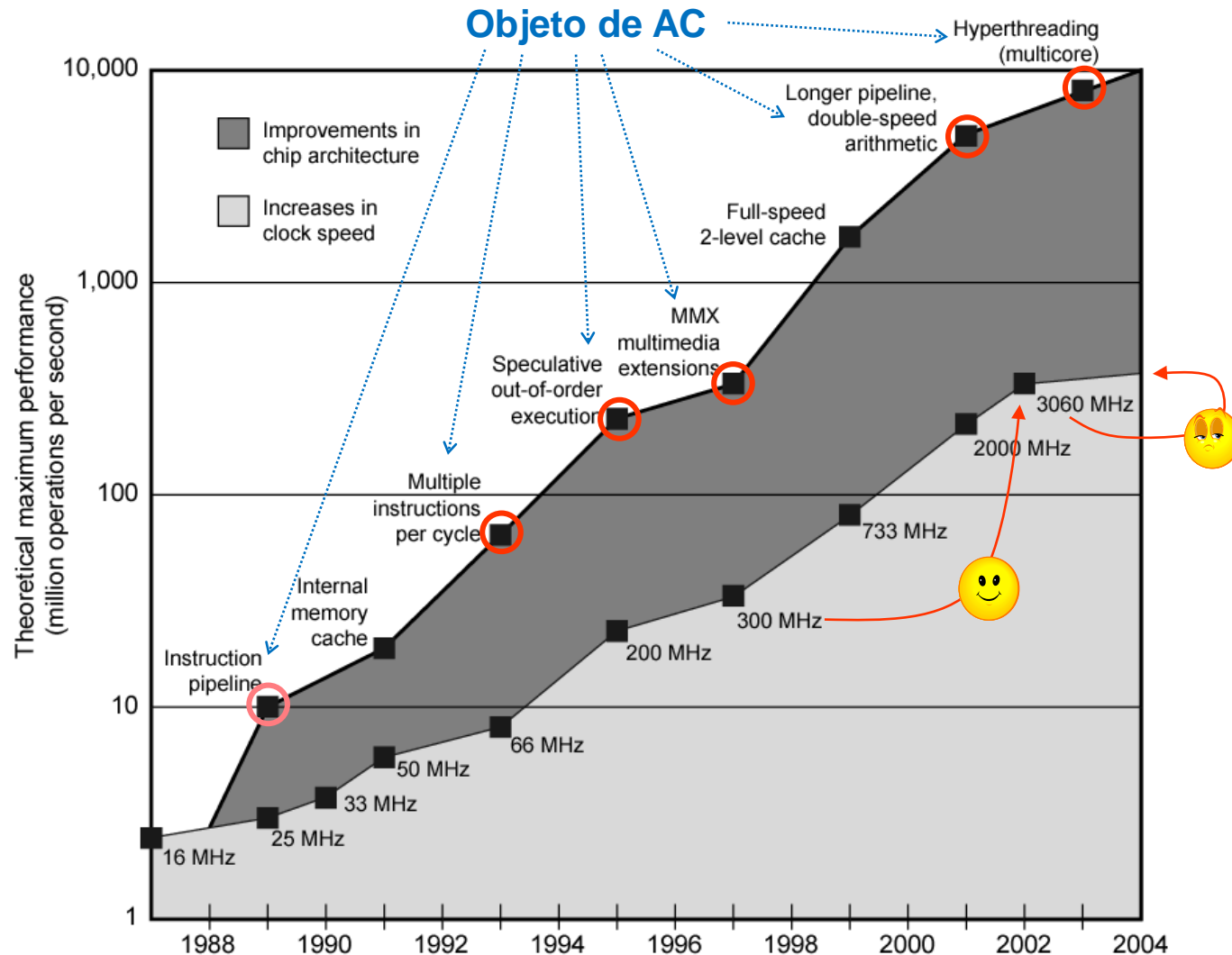
□ Microelectrónica y microarquitectura



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

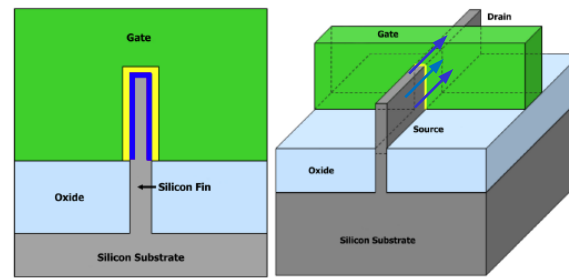
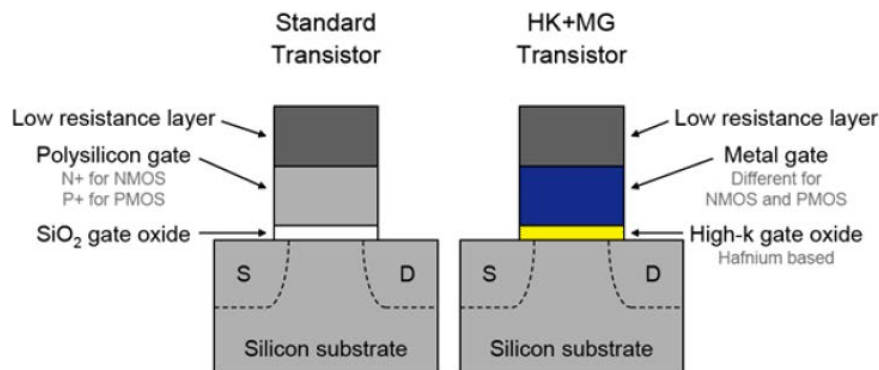


Evolución y tendencias: tecnología

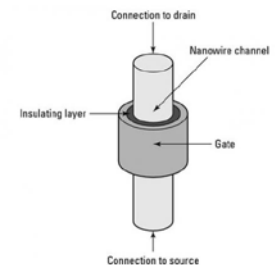
Manufacturing process details from 1997 to 2011

Process Name	P856	P858	Px60	P1262	P1264	P1266	P1268	P1270
1 st production	1997	1999	2001	2003	2005	2007	2009	2011
Process (feature size)	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 ₂	SiO ₂	SiO ₂	SiO ₂	SiO ₂	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	193nm	193nm

(Subject to change)



FinFET 3D



Gate-all around (GAAFET)

Hacia el final de la Ley de Moore...



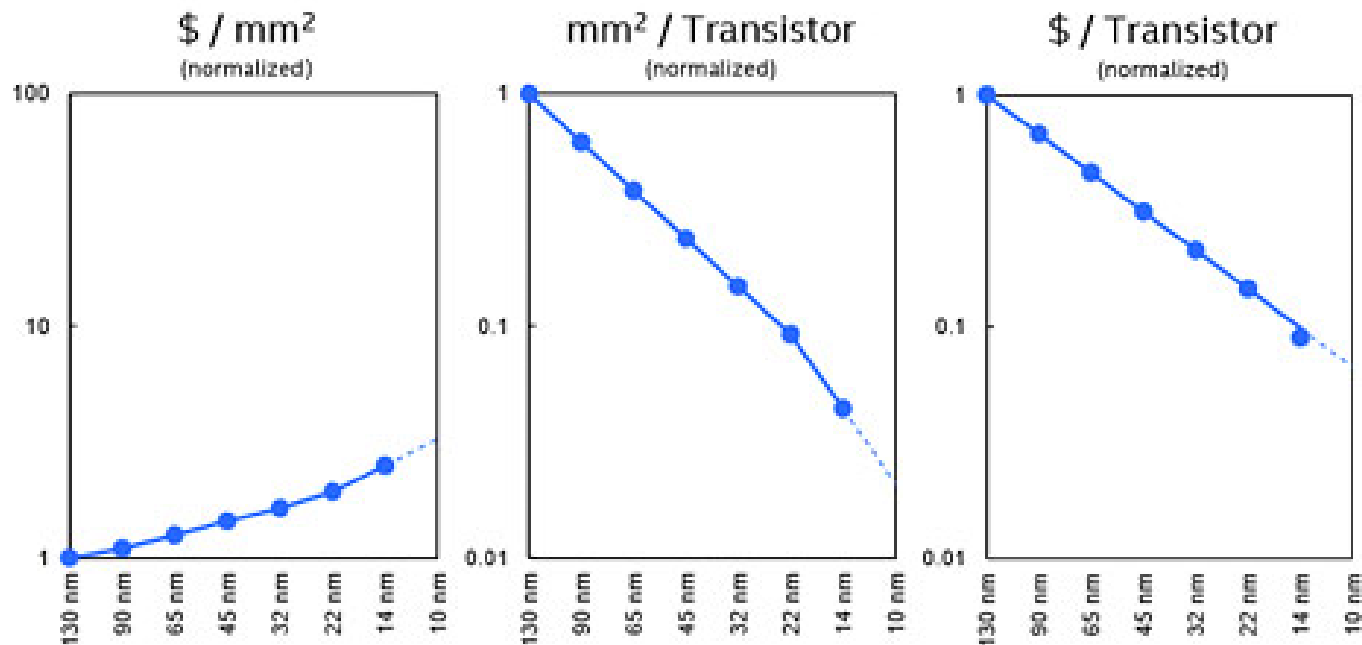
Globalfoundries 14nm 2016, ~~7nm 2018~~, Abandona la carrera (septiembre 2018)

AC — Tema 1

Evolución y tendencias: tecnología

Hacia el final de la Ley de Moore

El escalado en tamaño sigue siendo económicamente viable...



... pero...

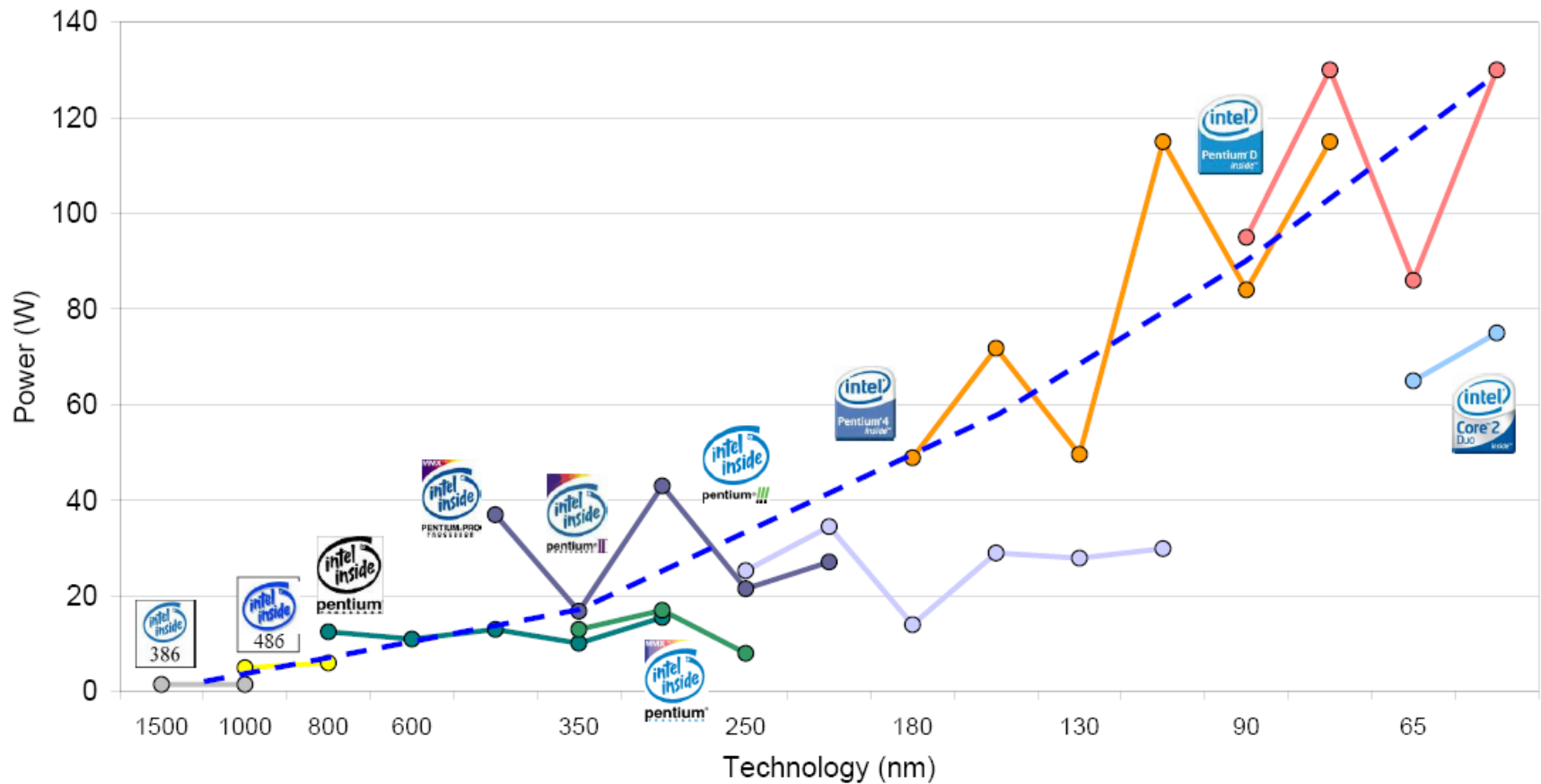
Scaling continues to provide lower cost per transistor
Cost reduction is needed to justify new technology generations



Evolución y tendencias: tecnología

La Ley de Moore: problemas...

1^{er} problema: consumo "Power Wall"

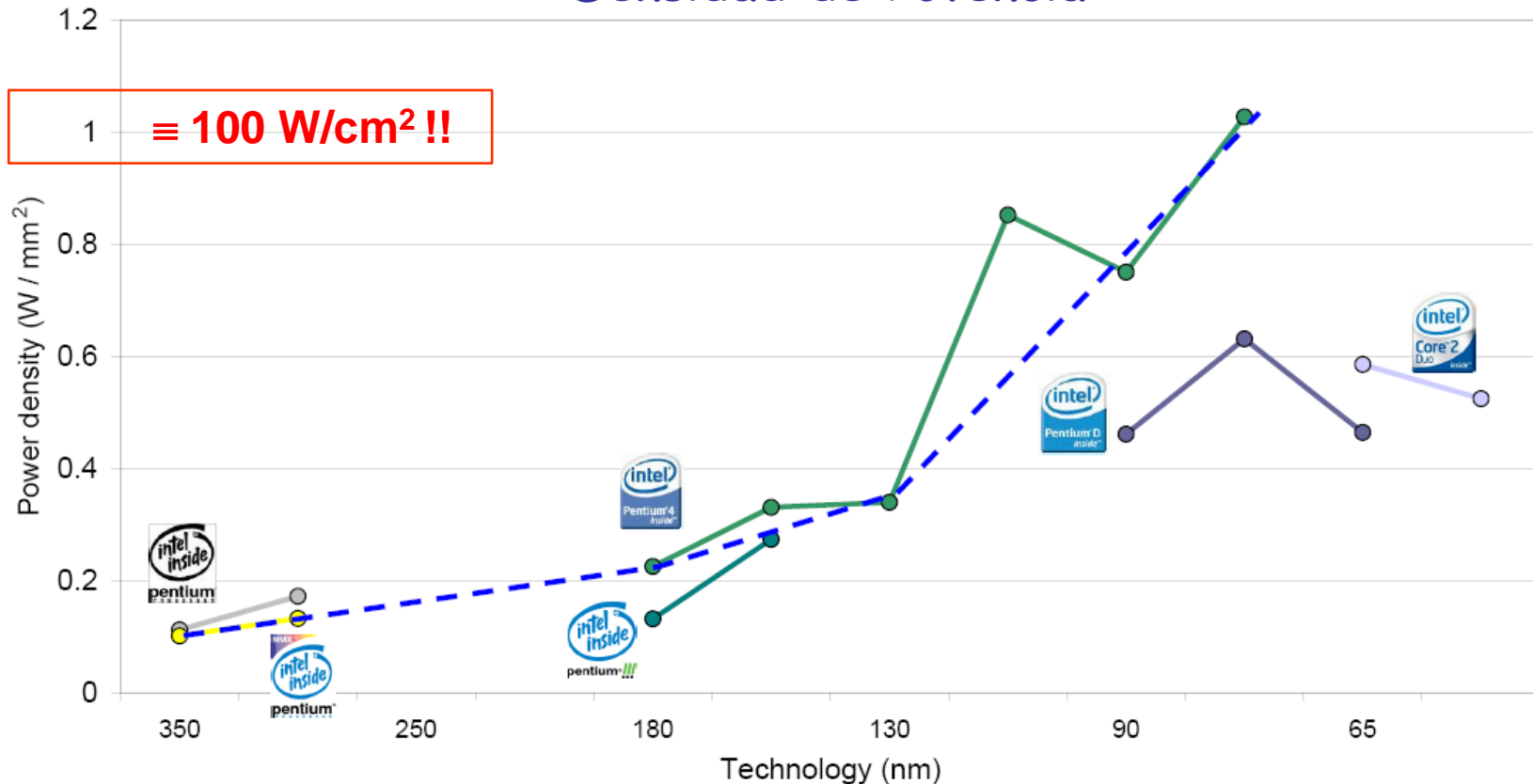


Fuente: Intel Corporation

Evolución y tendencias: tecnología

La Ley de Moore: problemas...

1^{er} problema: consumo "Power Wall" Densidad de Potencia



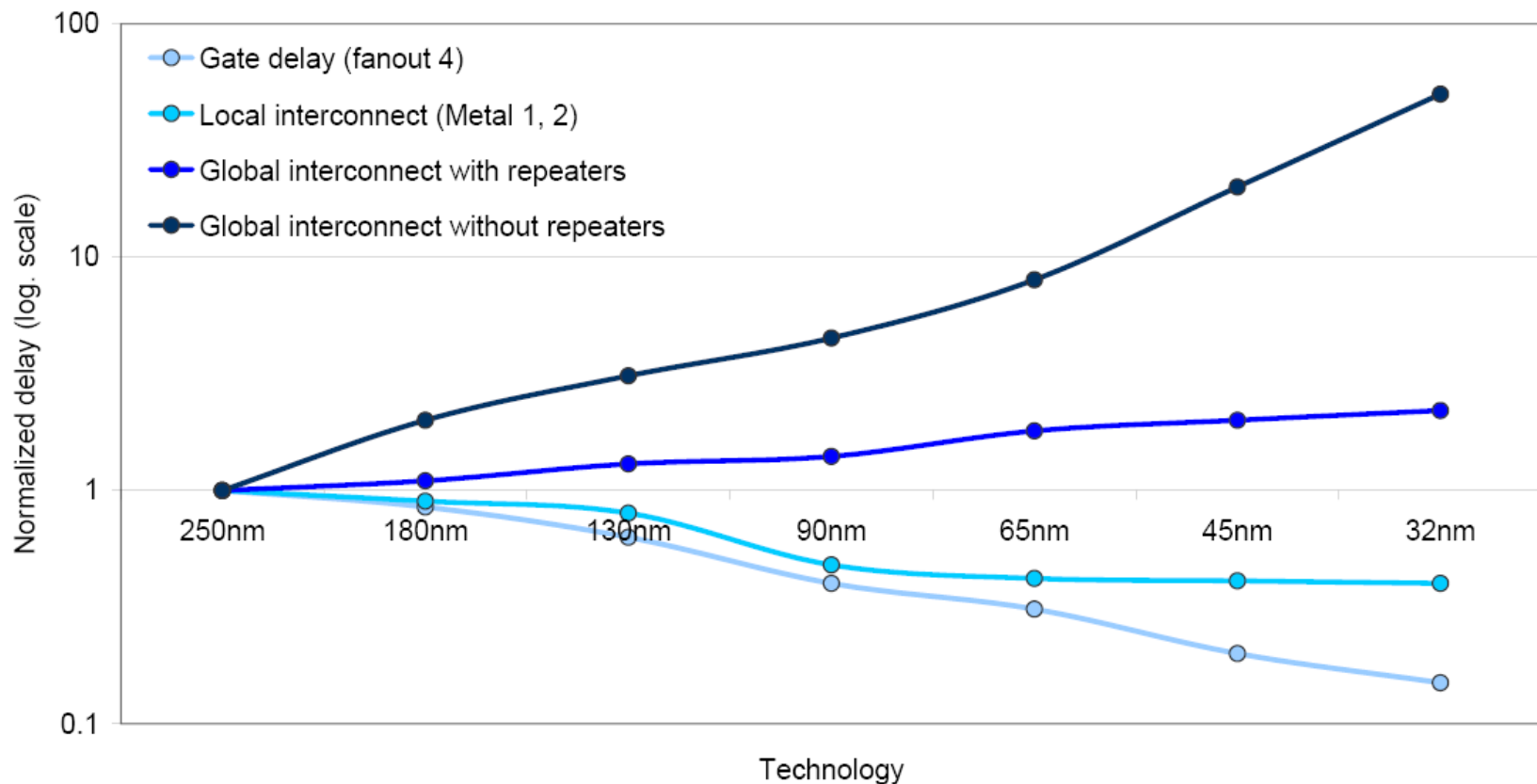
Fuente: Intel Corporation

La Ley de Moore: problemas...

2º problema: retardo interconexiones

El rendimiento del transistor (retardo) escala linealmente con la tecnología

El retardo de la interconexión no escala con la tecnología



Evolución y tendencias: tecnología

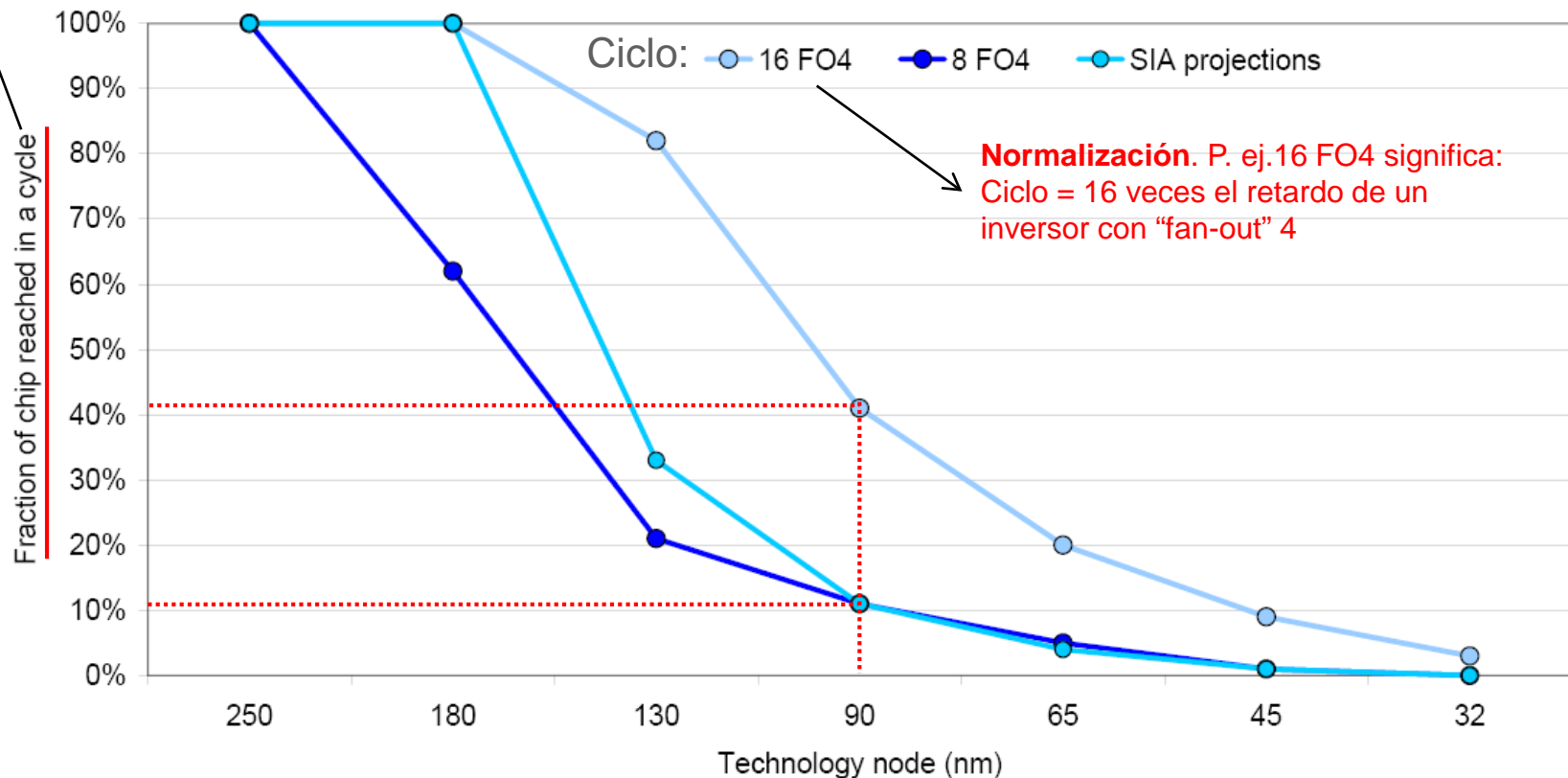
La Ley de Moore: problemas...

2º problema: retardo interconexiones.

¿Qué % del área del chip se puede alcanzar en un ciclo de reloj?

Pero...¿cuánto dura un ciclo?:

Normalizar



Multi - Many cores

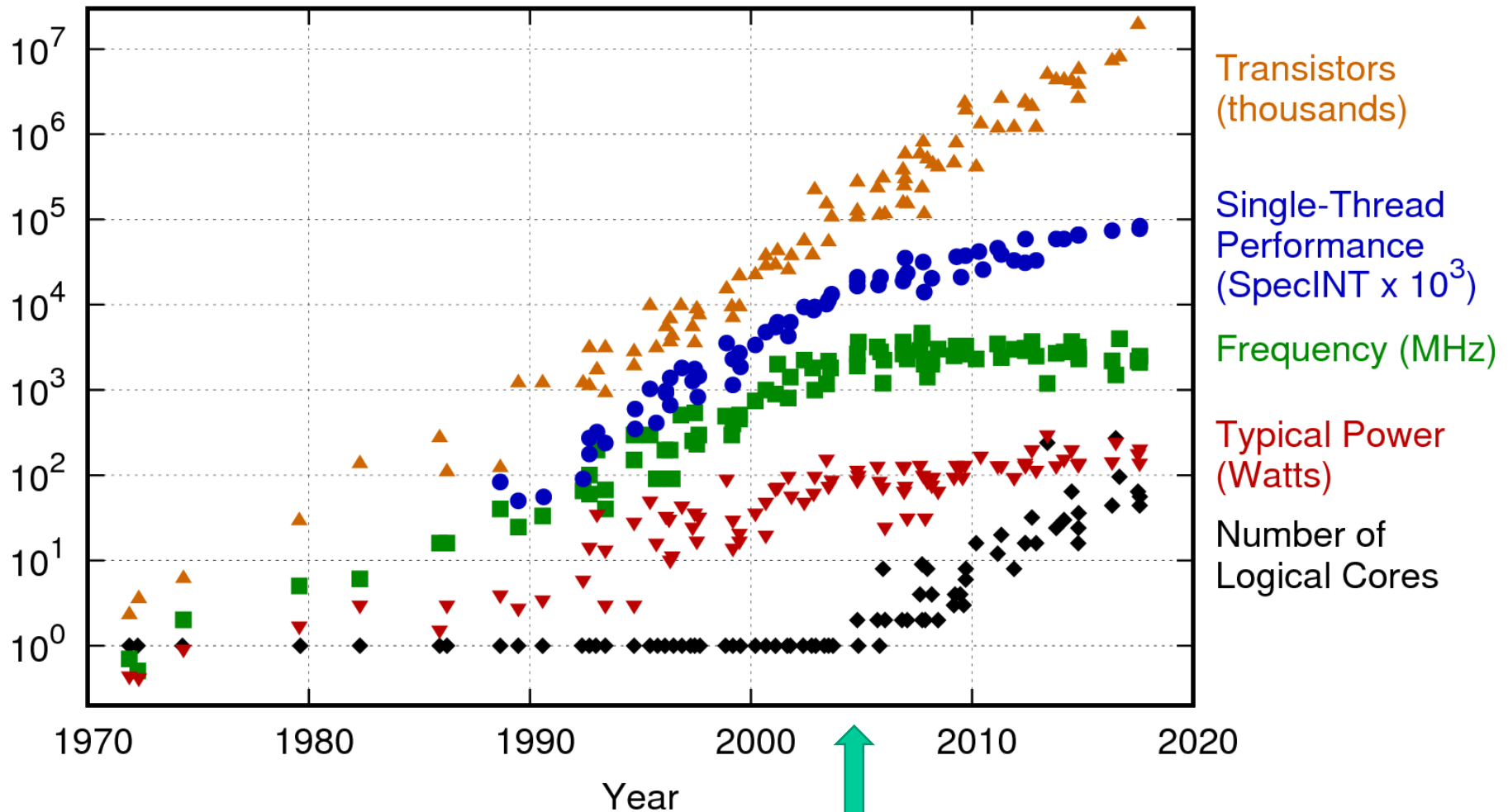
La Ley de Moore: el punto de inflexión

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

Evolución y tendencias: tecnología

42 Years of Microprocessor Trend Data



Original data up to the year 2010 collected and plotted by M. Horowitz, F. Labonte, O. Shacham, K. Olukotun, L. Hammond, and C. Batten
New plot and data collected for 2010-2017 by K. Rupp

Ancho de banda vs. Latencia (evolución ~30 años)

- ❑ Ancho de banda (o rendimiento)
 - o Cantidad total de trabajo realizada en un tiempo dado
 - o Mejora en procesadores y redes: 32,000X - 40,000X
 - o Mejora en memoria y discos: 2000X - 400X

- ❑ Latencia (o tiempo de respuesta)
 - o Tiempo entre el comienzo y la conclusión de un tarea
 - o Mejora en procesadores y redes: 80X - 50X
 - o Mejora en memoria y discos: 7X - 13X

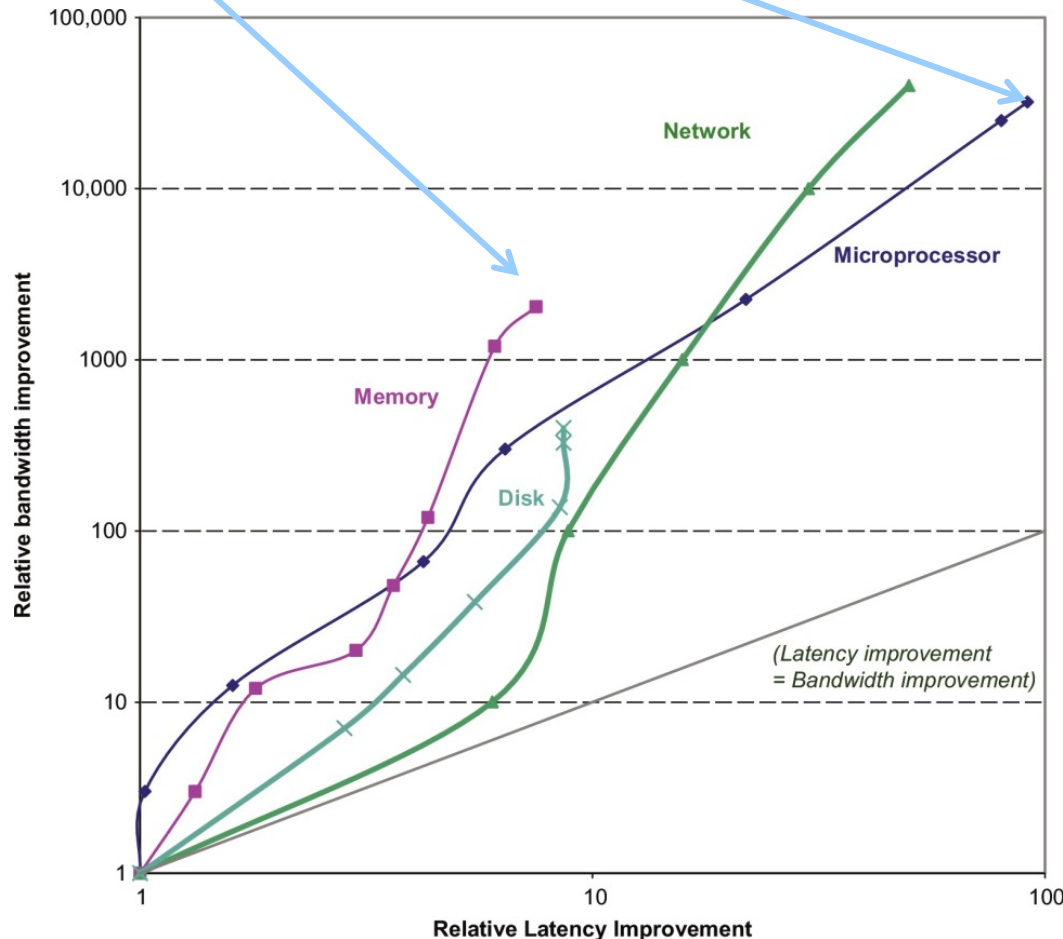
Datos obtenidos a partir de tabla 1.10 de H&P 6th ed.

- ❑ Factores que determinan este comportamiento?

Evolución y tendencias: tecnología

❑ Diferente evolución de AB y Lat para diferentes tecnologías: desequilibrios

CPU alta,
Memoria Baja
("Memory Wall")



❑ Mejora AB / Mejora Lat, siempre > 1 , pero diferentes evoluciones:

- o Procesador: '286, '386, '486, Pentium, Pentium 4, Core i7 (Lat 80x, AB 32,000x)
- o Ethernet: 10, 100 Mb/s, 1, 10, 100, 400 Gb/s (Lat 50x, AB 40,000x)
- o Modulo de Memoria: DRAM, Page Mode DRAM, SDRAM, DDR2-3-4 SDRAM (Lat 7x, AB 2000x)
- o Disco : 3600, 5400, 7200, 10000, 15000 RPM (Lat 13x, AB 400x)

Lectura recomendada: sección 1.4 de H&P 5th ed. (o 6th ed.)

Evolución y tendencias: tecnología

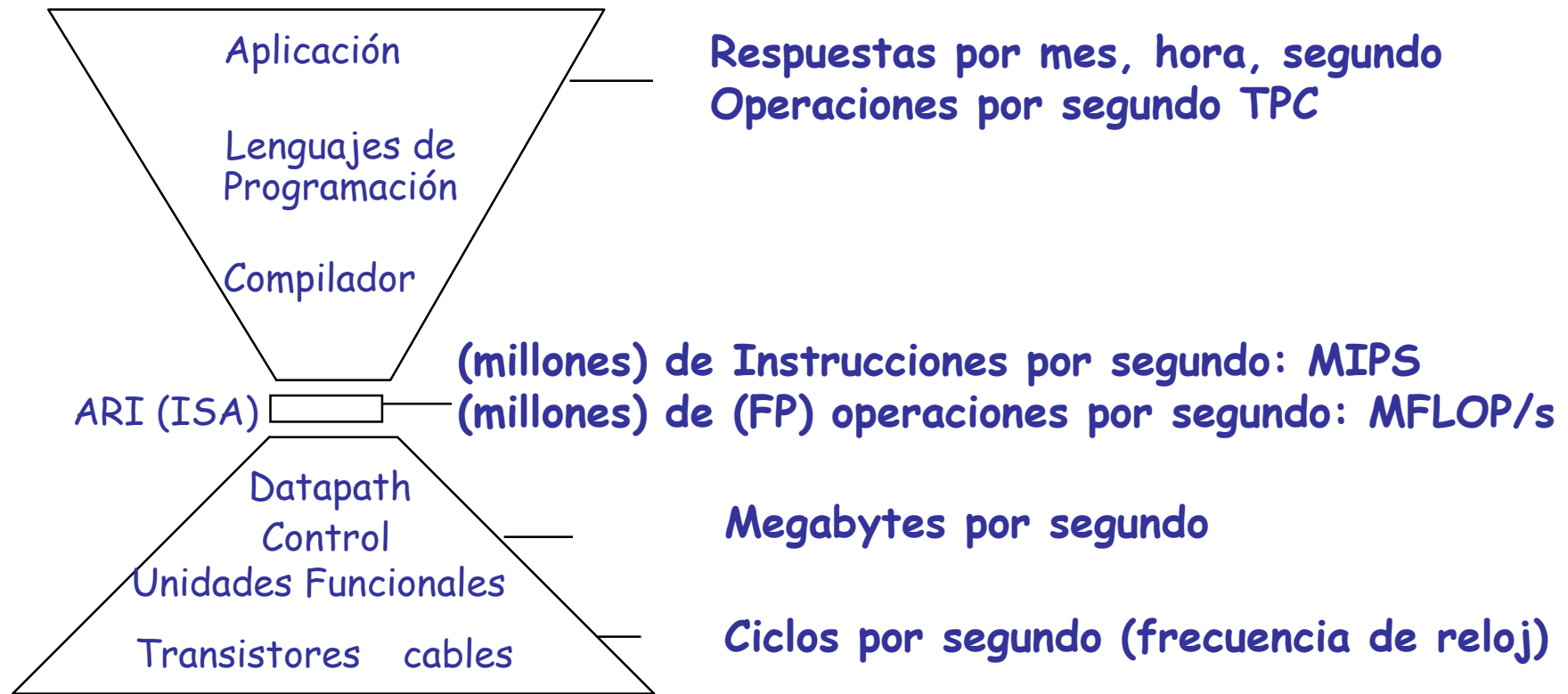
❑ Evolución de AB, Lat y otros parámetros en una ventana temporal de ~30 años

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	2015
Die size (mm ²)	47	43	81	90	308	217	122
Transistors	134,000	275,000	1,200,000	3,100,000	5,500,000	42,000,000	1,750,000,000
Processors/chip	1	1	1	1	1	1	4
Pins	68	132	168	273	387	423	1400
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	4000
Bandwidth (MIPS)	2	6	25	132	600	4500	64,000 x 32K
Latency (ns)*	320	313	200	76	50	15	4 x 80

Fig 1.10 H&P 6th ed (detalle)

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

❑ Medidas del rendimiento



La única medida fiable es el tiempo de ejecución programas reales. Dos aspectos:

- Rendimiento del procesador
- Rendimiento del computador

❑ Rendimiento del procesador

- Tiempo de ejecución de un programa

$$T_{CPU} = \text{Ciclos} \times t = \text{Ciclos} / f$$

- ✓ Ciclos: nº total de ciclos de reloj empleados
- ✓ t: período de reloj (depende de implementación, tecnología)
- ✓ f: frecuencia de reloj = $1/t$

- Ciclos promedio por instrucción:

$$CPI = \text{Ciclos} / NI$$

- ✓ NI: nº de instrucciones ejecutadas (depende de compiladores y LM)
- ✓ CPI: (depende de LM, implementación, paralelismo)

- Sustituyendo:

$$T_{CPU} = NI \times CPI \times t = NI \times CPI / f$$

□ Rendimiento del procesador

- Si asumimos n tipos de instrucciones, cada tipo con una duración de CPI_i ciclos:

$$Ciclos = \sum_{i=1}^n CPI_i \times NI_i$$

✓ NI_i : n° de instrucciones del tipo i ejecutadas

- Dividiendo por NI ,

$$CPI = \frac{1}{NI} \times \sum_{i=1}^n CPI_i \times NI_i = \sum_{i=1}^n CPI_i \times F_i$$

✓ $F_i = NI_i / NI$: fracción de instrucciones del tipo i ejecutadas

Invertir recursos donde se gasta el tiempo

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

$$CPI = 1 \cdot 0,5 + 2 \cdot 0,2 + 2 \cdot 0,1 + 2 \cdot 0,2 = 1.5$$

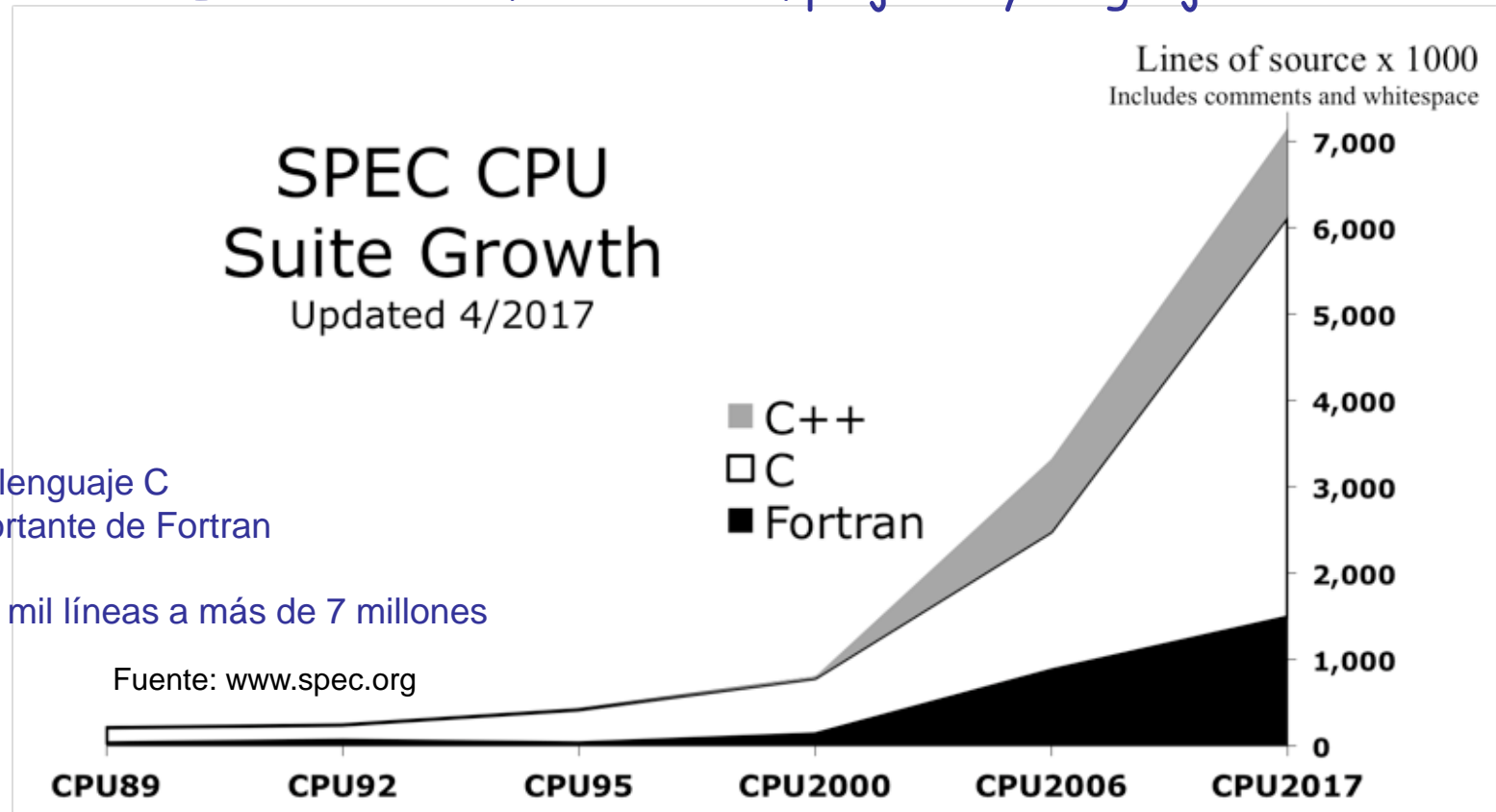
- ❑ Rendimiento global del computador: programas de prueba (benchmarks)
 - o La única forma fiable es ejecutando distintos programas reales.
 - Programas “de juguete”: 10~100 líneas de código con resultado conocido. Ej: Criba de Eratóstenes, Puzzle, Quicksort
 - Programas de prueba sintéticos: simulan la frecuencia de operaciones y operandos de un abanico de programas reales. Ej: Whetstone, Dhrystone
 - ¿Por qué estos benchmarks han perdido fiabilidad?
 - o Programas reales típicos con cargas de trabajo fijas (actualmente es la medida más aceptada): **SPEC CPU**
 - o Otros
 - HPC: LINPACK, SPEChpc96, Nas Parallel Benchmark
 - Servidores: ~~SPECweb~~, SPEC_Virt_SC, SPECsfs(File servers), TPC-C, SPECjbb (Java)
 - Graficos: SPECviewperf(OpenGL), SPECapc(aplicaciones 3D)
 - Winbench, EEMBC

Lectura recomendada: sección 1.8 de H&P 5th ed. (o 6th ed.)
y <https://www.spec.org/cpu2017/Docs/overview.html>

Rendimiento: SPEC

- ❑ Rendimiento global del computador: SPEC (Standard Performance Evaluation Corporation)
 - o Desde 1989 define series de programas de prueba ("benchmark suites") que representan aplicaciones reales de diferentes tipos.
 - o 6 generaciones de SPEC CPU: 89, 92, 95, 2000, 2006 y 2017.
 - Cada una más compleja que la anterior.
 - o En cada generación los programas que integran la suite SPEC son diferentes de los de la generación anterior (aunque alguno puede permanecer)
 - o ¿Por qué es preciso definir una nueva suite cada cierto tiempo?
- ❑ ¿Qué se mide en SPEC CPU 2017?
 - o Rendimiento en computación intensiva (similar a sus predecesores) . Se enfatiza el rendimiento de:
 - Procesador: Chip(s) de la CPU
 - Memoria: Jerarquía de memoria, incluyendo caches y memoria principal
 - Compiladores: C, C++ y Fortran, incluyendo optimizadores
 - o No se mide el rendimiento de:
 - redes, hw gráfico, bibliotecas Java, E/S... Otros SPEC para ello
 - o Dos tipos de medidas:
 - **SPECspeed**: Rendimiento del computador en la ejecución de una tarea (para cada benchmark, se mide su tiempo de ejecución)
 - **SPECrate**: Rendimiento del computador en la ejecución de varias tareas en paralelo (para cada benchmark, se mide el tiempo necesario para ejecutar varias copias del mismo a la vez).

❑ Evolución de SPEC CPU: incremento de complejidad y lenguajes usados



- Predominio del lenguaje C
- Presencia importante de Fortran
- Total código:
De unas 250 mil líneas a más de 7 millones

Nº Progr INT		6	8	12	12	10
Nº Progr FP	10	14	10	14	17	10/13*

* SPECspeed/SPECrate

Rendimiento: SPEC

❑ ¿Qué tipos de programas se usan como benchmarks?

- o Enteros: Compiladores (GNU C) e intérpretes (Perl), simulación de eventos discretos, compresión de vídeo y datos, IA (p.ej. Sudoku)...
- o Punto flotante: Dinámica molecular, dinámica de fluidos, predicción meteorológica, imagen biomédica, modelado oceánico...

❑ Anatomía de SPEC2017

Fuente: www.spec.org

Suite	Contents	Metrics	How many copies? What do Higher Scores Mean?
SPECspeed 2017 Integer	10 integer benchmarks	SPECspeed2017_int_base* SPECspeed2017_int_peak**	SPECspeed suites always run one copy of each benchmark. Higher scores indicate that less time is needed.
SPECspeed 2017 Floating Point	10 floating point benchmarks	SPECspeed2017_fp_base SPECspeed2017_fp_peak	
SPECrate 2017 Integer	10 integer benchmarks	SPECrate2017_int_base SPECrate2017_int_peak	SPECrate suites run multiple concurrent copies of each benchmark. The tester selects how many. Higher scores indicate more throughput (work per unit of time).
SPECrate 2017 Floating Point	13 floating point benchmarks	SPECrate2017_fp_base SPECrate2017_fp_peak	

* base: resultado con opciones de compilación que se ajustan a reglas estrictas fijadas por SPEC

** peak: resultado con opciones de compilación elegidas por el usuario

❑ ¿Cómo se calculan las métricas SPECspeed?

- o Para cada benchmark, i , se mide el tiempo de ejecución en el sistema a evaluar (SAE) y se calcula la razón:

$$r_i = \frac{\text{Tiempo de ejecución en la "máquina de referencia"}}{\text{Tiempo de ejecución en el SAE}}$$

❑ ¿Y en SPECrate?

- o Para cada benchmark, i , se lanza a ejecutar un nº de copias determinado por el usuario, nc , y se calcula:

$$r_i = \frac{T.\text{de ejecución de 1 copia en la "máquina de referencia"}}{T.\text{de ejecución de las } nc \text{ copias el SAE}} \times nc$$

❑ ¿Qué es la máquina de referencia?

- o Un computador elegido para comparar con los demás
 - En SPEC CPU 2017: Sun Fire V490 con chips UltraSPARC-IV+ a 2100 MHz
- o ¿Dónde encuentro los tiempos de ejecución en la máquina de referencia?
 - En la web de SPEC
- o ¿Es crucial la elección de la máquina de referencia?
 - Si comparamos un par de computadores, el resultado no depende de la maq. ref.

❑ ¿Cómo se calculan los resultados globales en SPEC?

- o Para cada suite, se calculan las razones r_i ($i=1..n$) correspondientes a los n benchmarks que la integran
- o Para dicha suite se reporta un único valor, dado por la media geométrica (MG) de las n razones, es decir

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

- o Ejemplo: Para calcular "SPECspeed 2017_int_base" de un cierto SAE:

1. Obtener los tiempos de ejecución de cada uno de los 10 benchmarks enteros en el SAE, con las opciones de compilación "base"
2. Calcular las razones r_i ($i=1..10$) entre los t de ejecución en la maq. ref. y los t de ejecución en el SAE
3. Resultado (mejor rendimiento cuanto mayor):

$$SPECspeed2017_int_base = \sqrt[10]{r_1 \times r_2 \times \dots \times r_{10}}$$

❑ ¿Qué ventaja aporta el uso de la MG?

- o Ejemplo simplificado: Supongamos una suite con solo dos benchmarks y comparemos dos computadores A y B

Tiempos de ejecución en cada computador y razones

Benchmark		Maq. Ref.	Comp. A	Razón	Comp. B	Razón
	P1	R1	A1	R1/A1	B1	R1/B1
	P2	R2	A2	R2/A2	B2	R2/B2

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

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

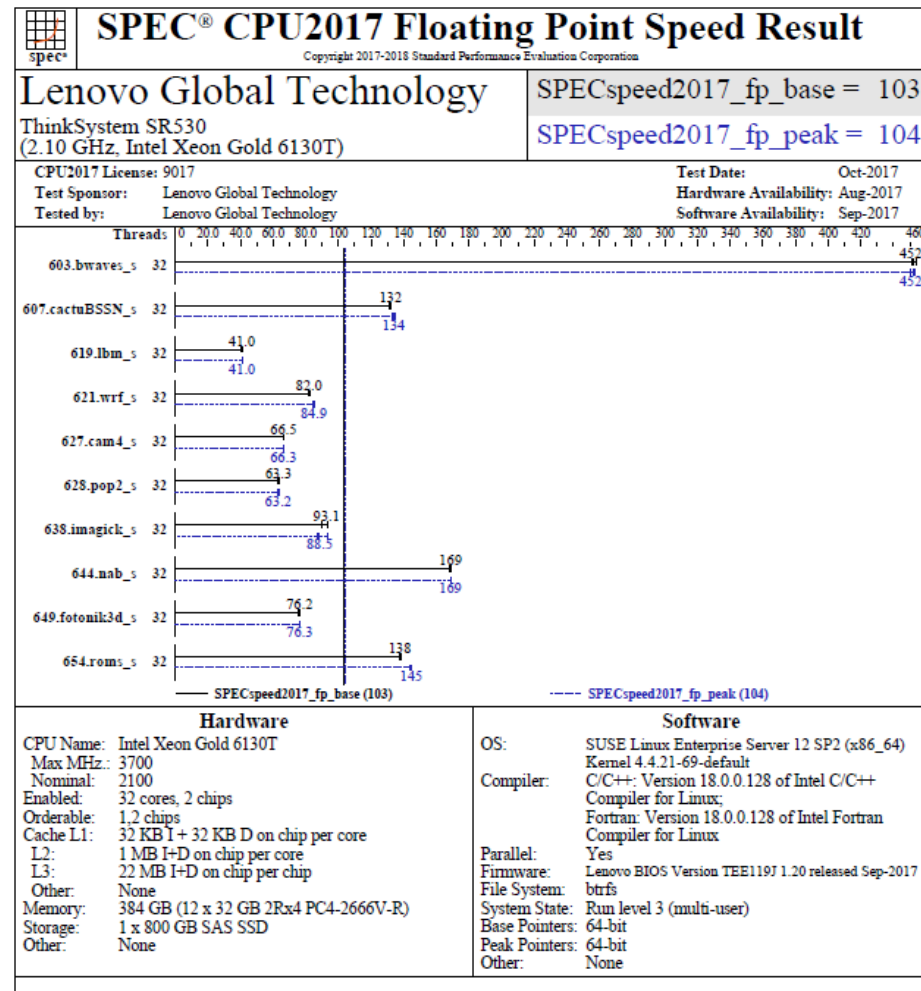
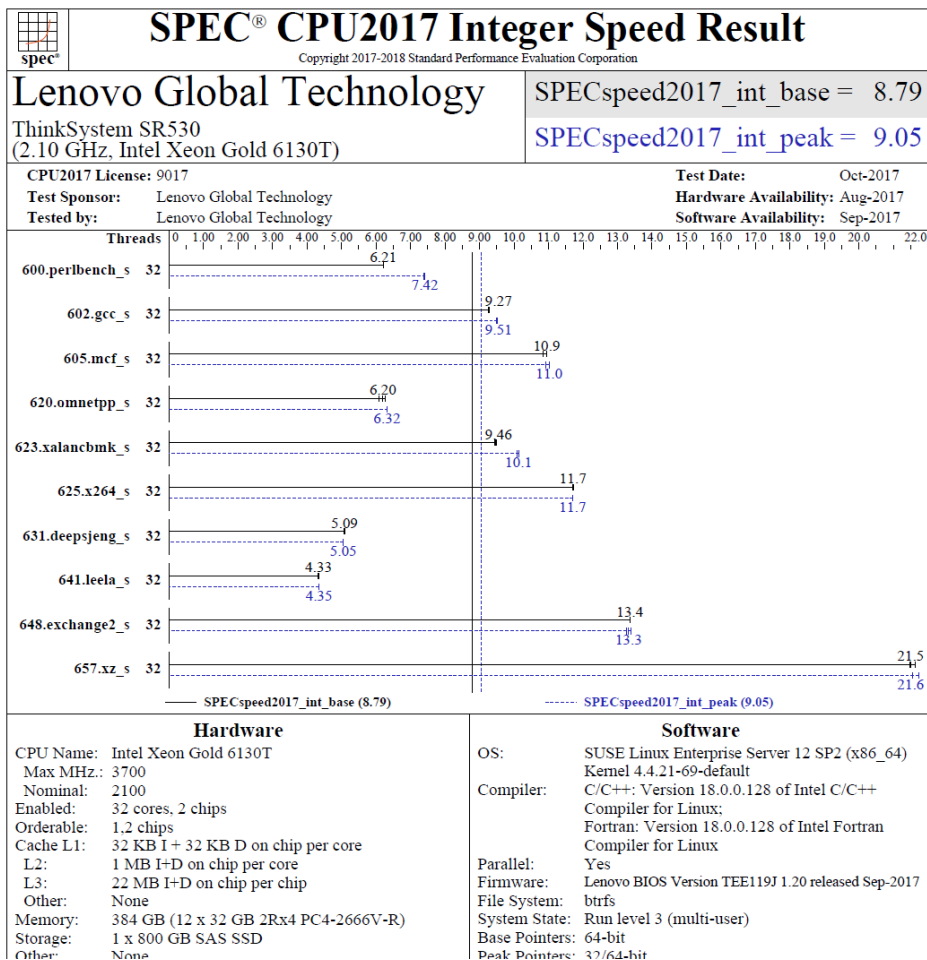
- o El uso de la MG garantiza la neutralidad de la máquina de referencia

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

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

Rendimiento: SPEC

¿Cómo se presentan los resultados?



Rendimiento: Un principio simple

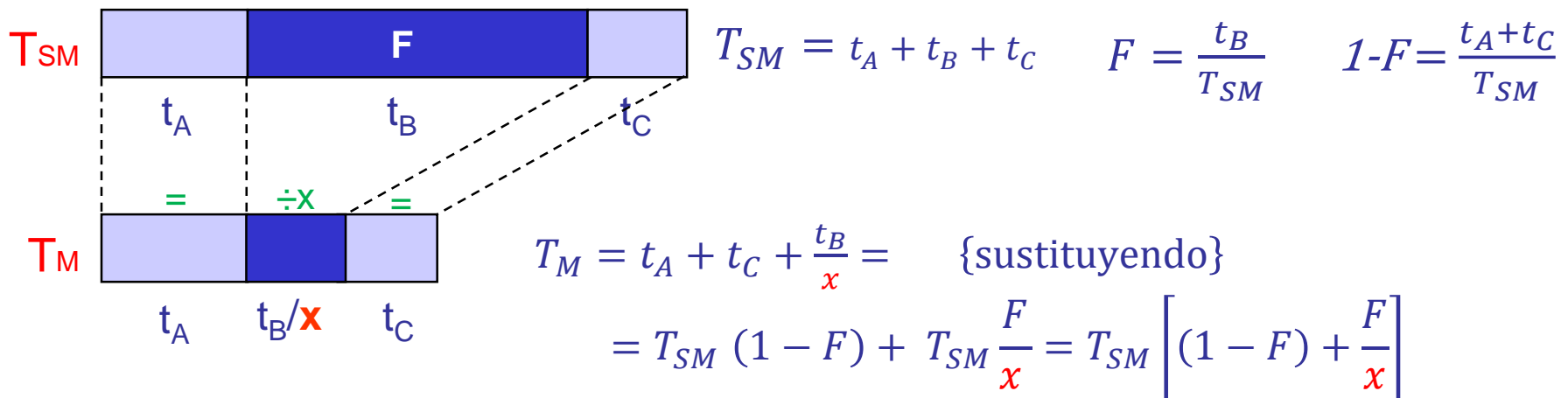
□ Un principio básico: Hacer rápidas las funciones frecuentes.

"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 fracción, F , del código x veces más rápido.

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



Rendimiento: Un principio simple

□ La Ley Amdahl

$$T_M = T_{SM} \left[(1 - F) + \frac{F}{x} \right]$$

$$\text{Speedup} \{def\} = \frac{T_{SM}}{T_M} = \frac{1}{(1 - F) + \frac{F}{x}}$$

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

$$T_M = T_{SM} \times (0.9 + 0.1 / 2) = 0.95 \times T_{SM}$$
$$\text{Speedup} = \frac{1}{0.95} = 1.053$$

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.

$$T_M = T_{SM} \times (0.1 + 0.9 / 100) = 0.109 \times T_{SM}$$
$$\text{Speedup} = \frac{1}{0.109} = 9.17$$

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:

Proesadores (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%)

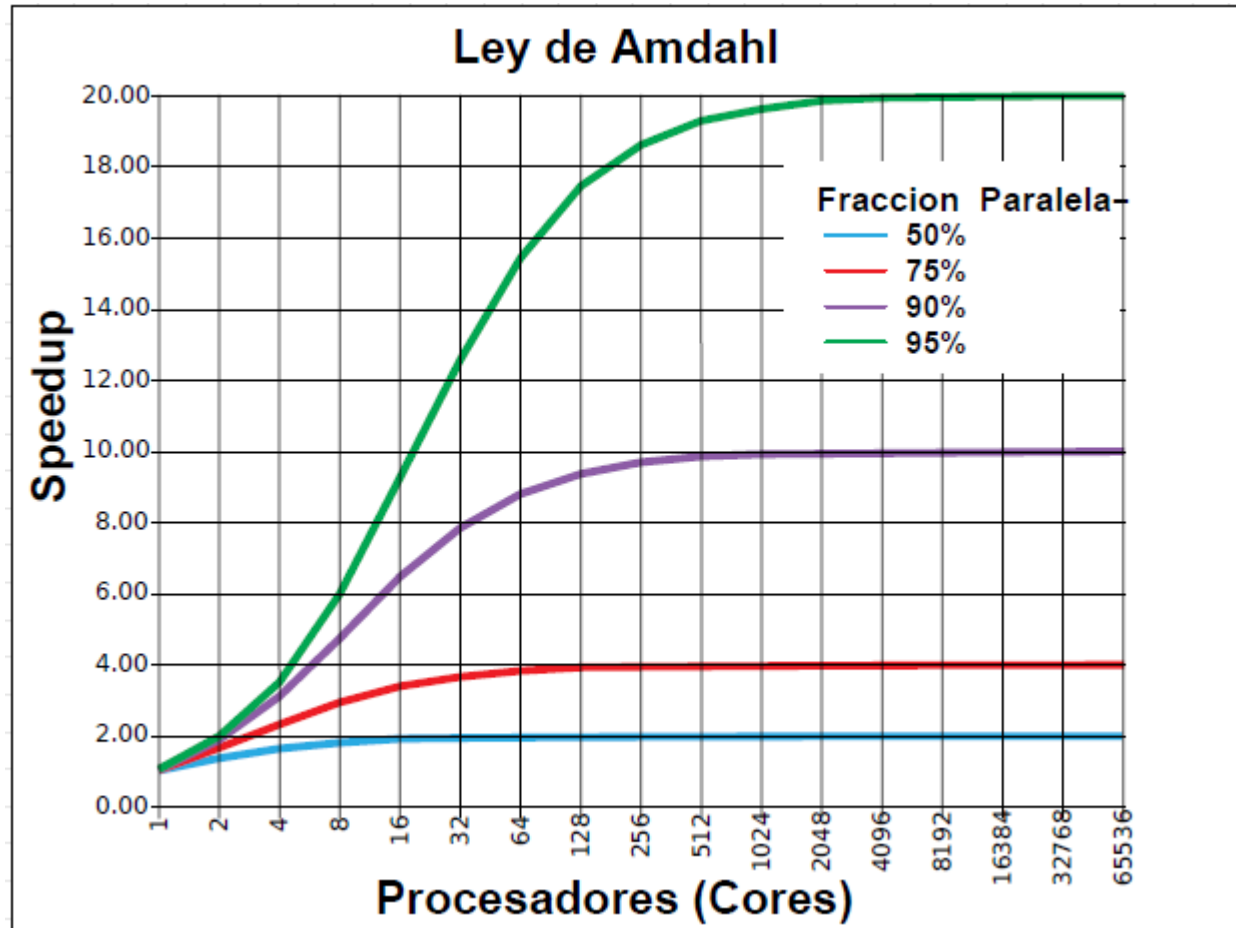
Observaciones:

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 nº de procesadores apenas mejora el Speedup, por lo que se degradada mucho la Eficiencia.

Rendimiento: Un principio simple

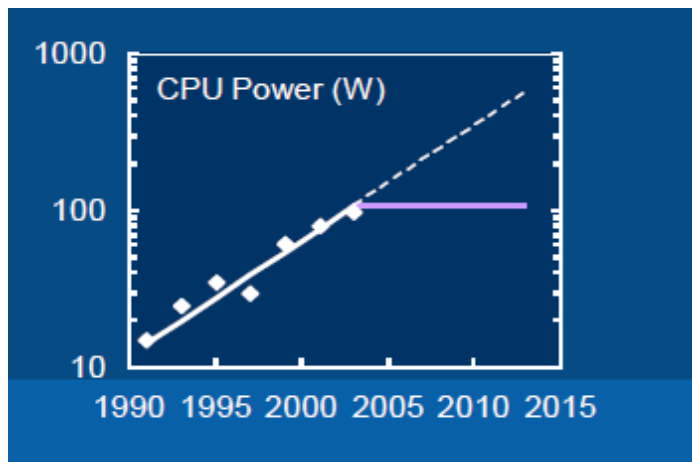
Everyone knows Amdahl's Law but quickly forgets it!

Thomas Puzak (IBM's T. J. Watson Research Center)



Consumo: 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
- ❑ Aspectos a considerar:
 - o Consumo máximo de potencia posible (pico). Diseño para el máximo?
 - o Consumo de potencia sostenido. Thermal Design Power (TDP).
 - o Determina la potencia del sistema de refrigeración y de la fuente de alimentación
 - o Está por debajo de la potencia "pico" y por encima de la "media"
 - o Energía y eficiencia energética: Cuál es la métrica más adecuada para comparar el rendimiento energético de dos procesadores?
Energía vs. Potencia.



- ❑ Objetivos
 - o Autonomía
 - o Tamaño compacto
 - o Reducir coste y consumo del sistema de refrigeración

Lectura recomendada: sección 1.5 de H&P 5th ed.

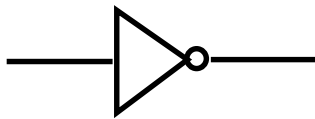
Consumo: Potencia y Energía

- ❑ Reto: “seguir incrementando en rendimiento sin incrementar el consumo”
- ❑ Energía dinámica: Consumida para la conmutación de los transistores.
 - o Transistores más pequeños, rápidos y de menor consumo (menor energía por conmutación). Pero... mayor cantidad en el chip!
 - o Evitar conmutaciones innecesarias (clock gating, glitch reduction)
- ❑ Energía estática: se consume aunque los transistores no estén conmutando (leakage current)
 - o Reducir leakage current: aislante (gate dielectric) Hi-K
 - o Evitar leakage en módulos que no se están usando. Suprimir alimentación (power gating)

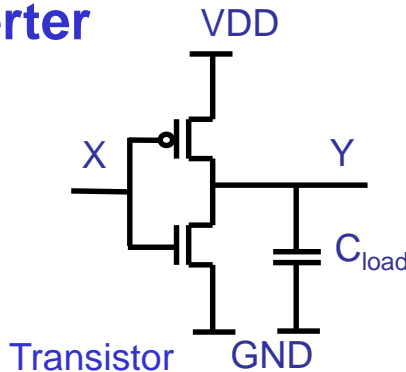
Consumo: Potencia y Energía

- E. dinámica consumida en transiciones de la entrada $1 \rightarrow 0 \rightarrow 1$

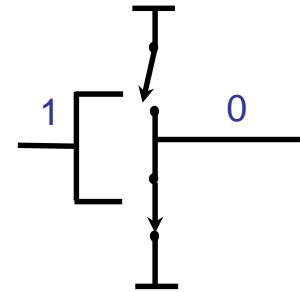
Static CMOS inverter



Schematic

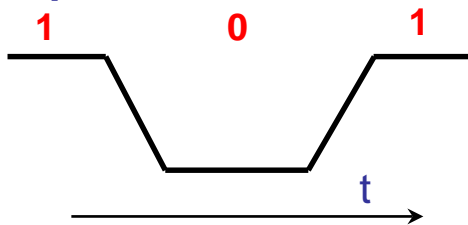


Transistor



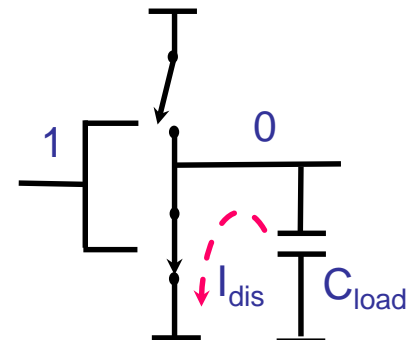
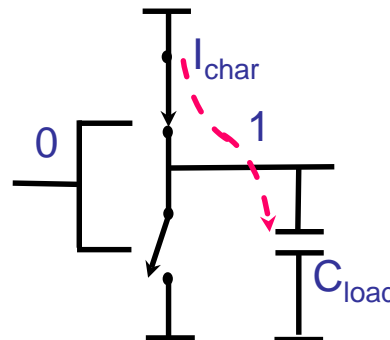
Ideal Switch

Input



$$E_{1 \rightarrow 0} = \frac{1}{2} V_{DD}^2 \times C_{load}$$

$$E_{0 \rightarrow 1} = 0$$



$$\text{Consumo medio por transición: } E = \frac{1}{4} V_{DD}^2 \times C_{load}$$

Consumo: Potencia y Energía

- ❑ Factores que influyen en la $E_{\text{dinámica}}$

$$E_{\text{dinámica}} \propto VDD^2 * C_{\text{load}}$$

- Reducir VDD. Impacto:

Cuadrático sobre el consumo de energía!

Pero además...

Lineal sobre frecuencia máxima y rendimiento (aprox.)

- Reducir C_{load} : Microarquitectura

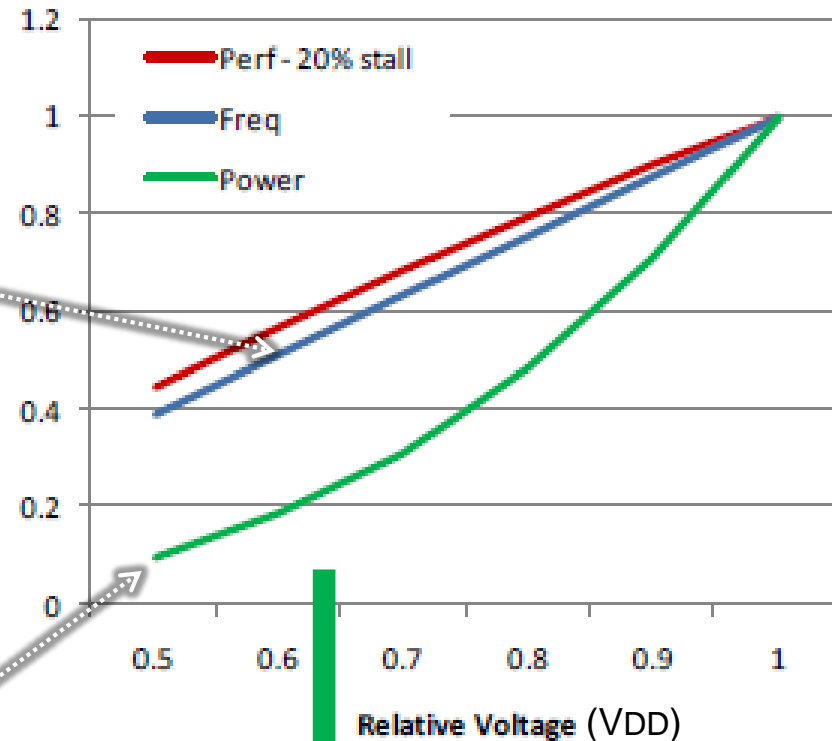
- ❑ Potencia $\text{dinámica} \propto VDD^2 * C_{\text{load}} * \text{Frecuencia}$

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

▪ Técnicas de Escalado Dinámico de Voltaje y Frecuencia (DVFS)

Impacto cúbico de VDD sobre la potencia dinámica: debido a la reducción adicional de la frecuencia máxima (En figura, aprox.: $0,125 = 0,5^2 \times 0,5$)

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



Inicialmente: VDD=1. Freq=1, Perf=1, Power=1

Si: VDD=0.5 → Freq≈0.4, Perf≈0.5, Power≈0.125

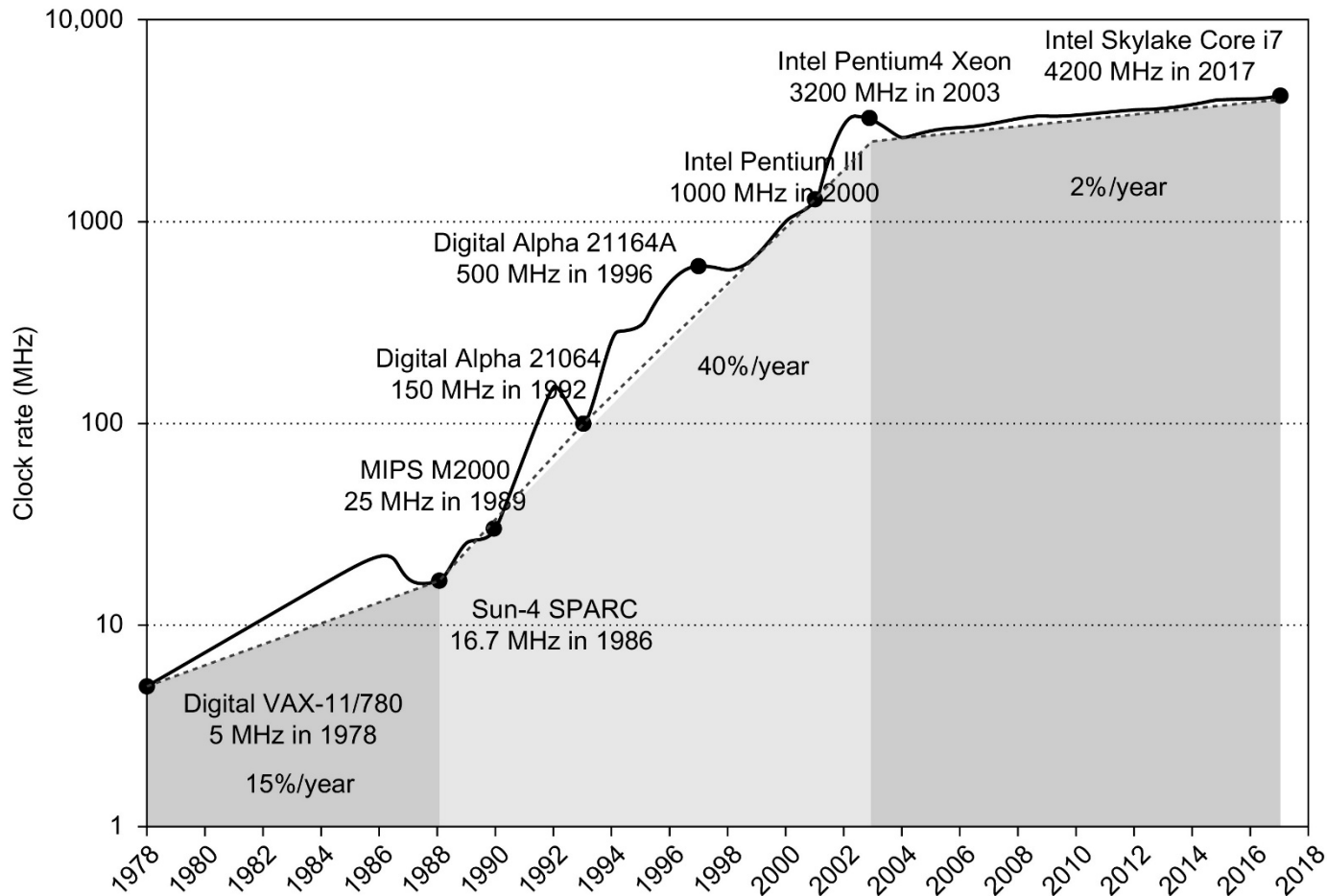
Añadir 7 cores → Power ≈ 1, Perf ≈ 4 !!

❑ Evolución del "clock"

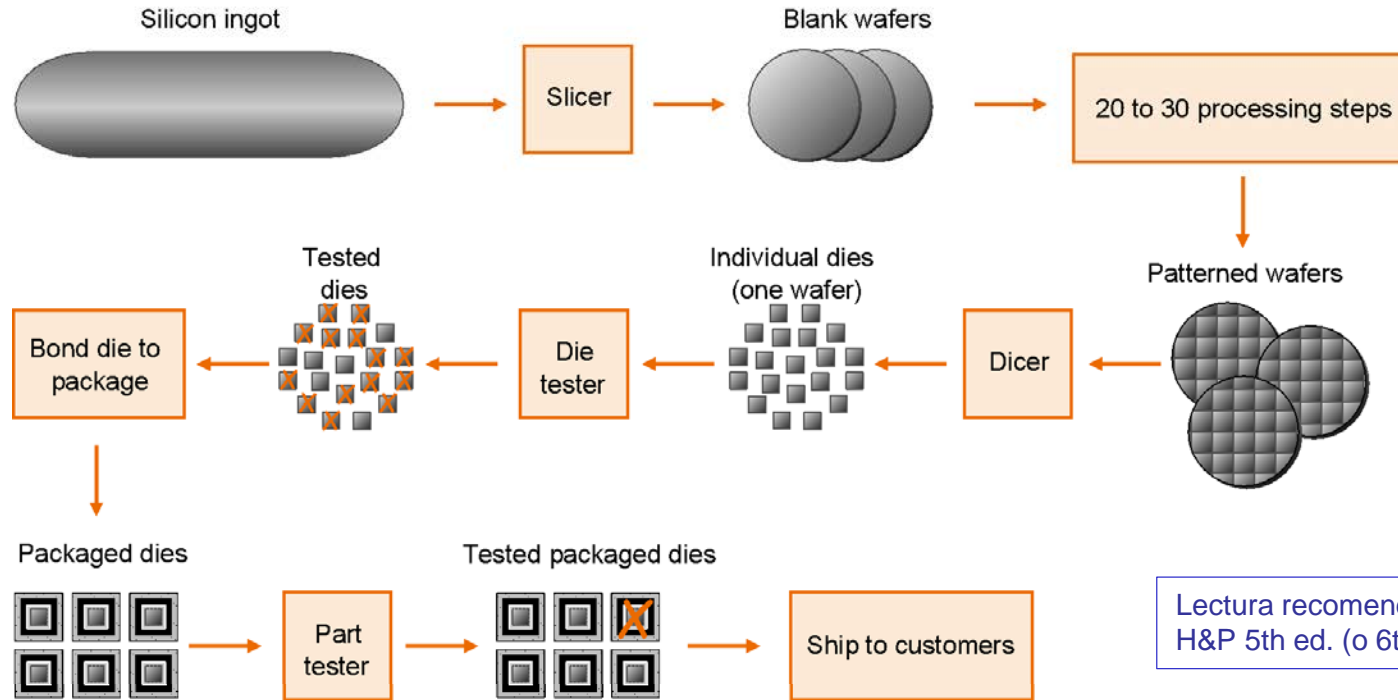
Intel 80386 2W

Intel Core i7 6700K / 4 GHz processor, 95 W (a disipar en 1.5 x 1.5 cm)

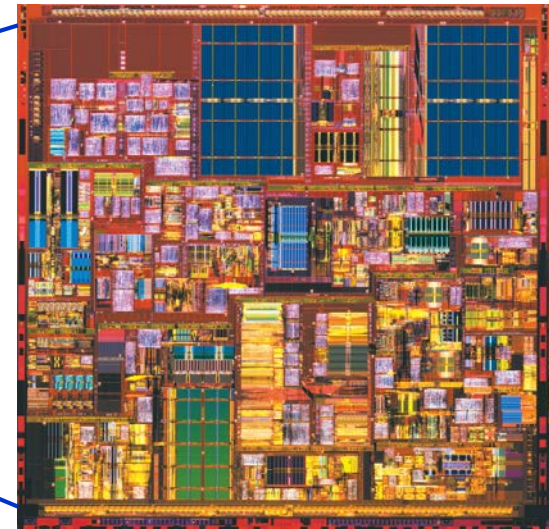
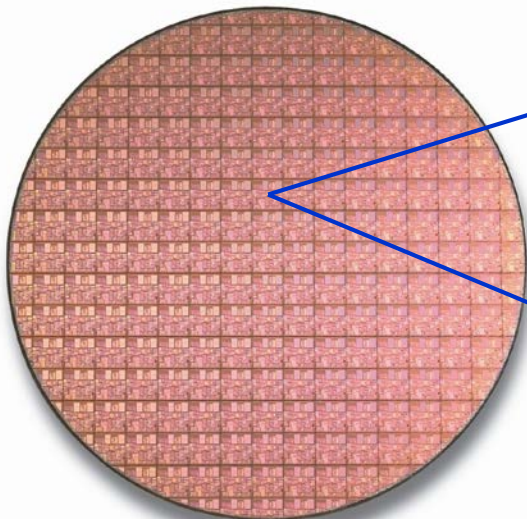
Limite para refrigeración con aire forzado



Coste: Fabricación de un CI



Lectura recomendada: sección 1.6 de H&P 5th ed. (o 6th ed.)



❑ Coste : El fundamental, el coste del CI

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

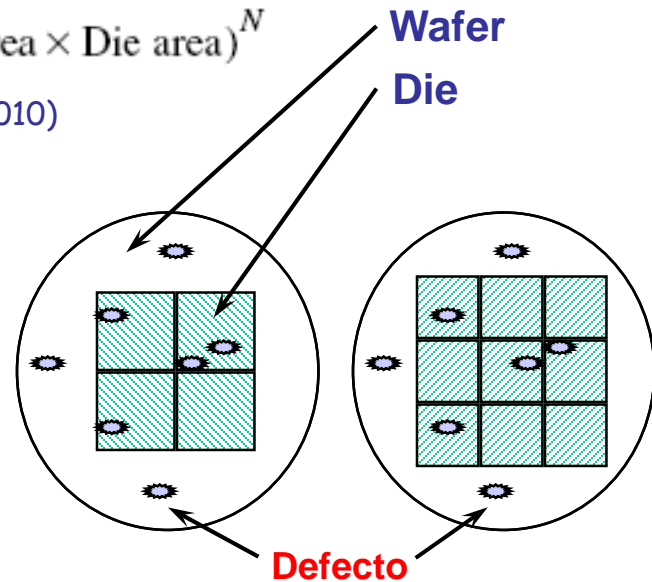
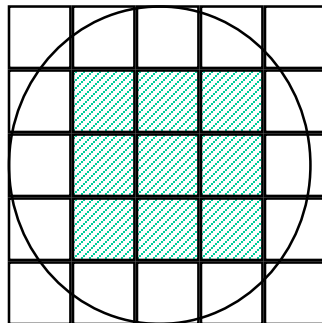
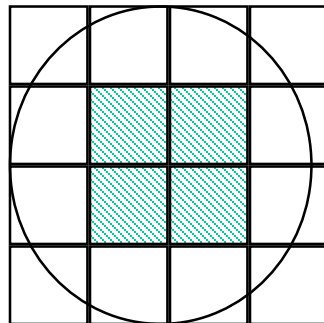
$$\text{Die coste} = \frac{\text{coste del Wafer}}{\text{Dies por Wafer} * \text{Die yield}}$$

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)



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

Factores que influyen en el binomio coste-precio

- ❑ Curva de aprendizaje (experiencia)
 - o Mejora del "yield": fracción de unidades fabricadas (procesadores, placas, sistemas...) que pasan los test de calidad.
 - o Abarata el coste por unidad, permite disminuir el precio
- ❑ Volumen de producción
 - o Estimación para coste μ -procesadores: 10% menos al duplicar el volumen
 - o Curva de aprendizaje más rápida.
- ❑ Paso de equipos especializados a "commodities" (artículos de consumo)
 - o Sector muy competitivo: margen beneficio↓, precio↓

Coste-Rendimiento-Consumo

❑ Tres servidores DELL PowerEdge

Lectura: Sección 1.10 de H&P 5th ed

o SW: IBM Java VM / MS Windows 2008 EE X64

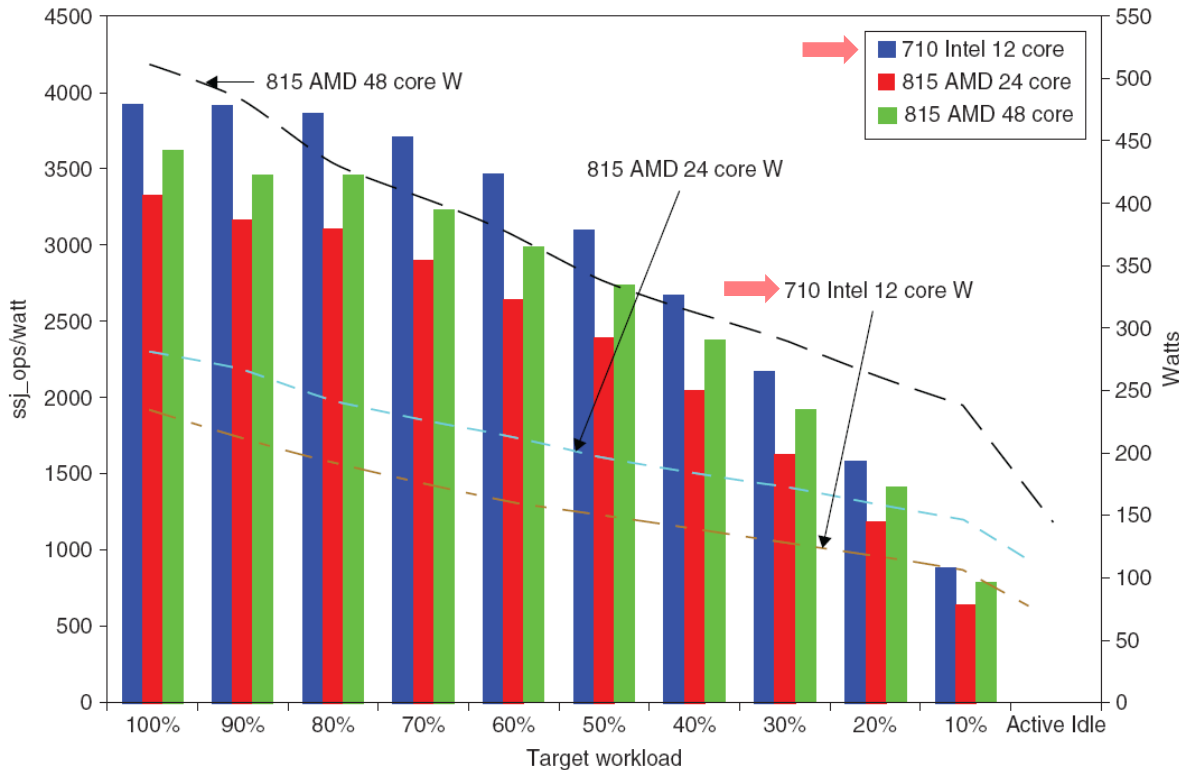
	System 1		System 2		System 3	
Component	Cost (% Cost)		Cost (% Cost)		Cost (% Cost)	
Base server	PowerEdge R710	\$653 (7%)	PowerEdge R815	\$1437 (15%)	PowerEdge R815	\$1437 (11%)
Power supply	570 W		1100 W		1100 W	
Processor	Xeon X5670	\$3738 (40%)	Opteron 6174	\$2679 (29%)	Opteron 6174	\$5358 (42%)
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-Consumo

SPEC Power Benchmark



- Usa SW escrito en Java
- Medida: $\frac{\text{ssj_ops}}{\text{watt}}$
 - $\frac{\text{server side java op}}{\text{watt}}$
- Interpretación alternativa: $\frac{\text{server side java op}}{\text{jul}}$
- Valores medidos a distintas cargas de trabajo (0-100%)
- El resultado de Spec Power es un escalar que "resume" todas las medidas:

$$\text{ssj_ops/w (global)} = \frac{\sum_{i=0}^{10} \text{ssj_ops}}{\sum_{i=0}^{10} \text{power}}$$

- R710
 - Mejor valor de ssj_ops/watt a todas las cargas
 - Mejor relación rendimiento / potencia
 - Además, es casi el más barato. Luego, mejor relación rendimiento/potencia/dólar
- Al tener en cuenta la eficiencia energética cambia la relación entre los tres sistemas!

System	Rendimiento: ssj_ops/w (global)	Rendim/\$
R710	3034	0.324
R815 (24 core)	2357	0.254
R815 (48 core)	2696	0.213