Arquitectura de Computadores

Tema 1
Introducción: Tendencias Tecnológicas
Coste / Rendimiento/ Consumo

DEPARTAMENTO DE
ARQUITECTURA DE COMPUTADORES
Y AUTOMÁTICA

Curso 2024-2025

Contenidos

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

Básica:

Capítulo 1 de Hennessy & Patterson 6th ed., 2017

Complementaria:

500 most powerful computer systems. www.top500.org

Standard Performance Evaluation Corporation. www.spec.org

International Roadmap for Devices and Systems irds.ieee.org

Arquitectura del computador

- Visión "clásica"
  - Las decisiones de diseño se ocupan básicamente del repertorio de instrucciones (Instruction Set Architecture, ISA)
  - 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"
  - Punto de partida: Análisis de requisitos del computador objetivo
  - Diseño para optimizar el rendimiento
    - Restricciones: coste, consumo de energía, tamaño, disponibilidad
  - 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...
  - 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.

ISA: sigue siendo un aspecto muy relevante

software

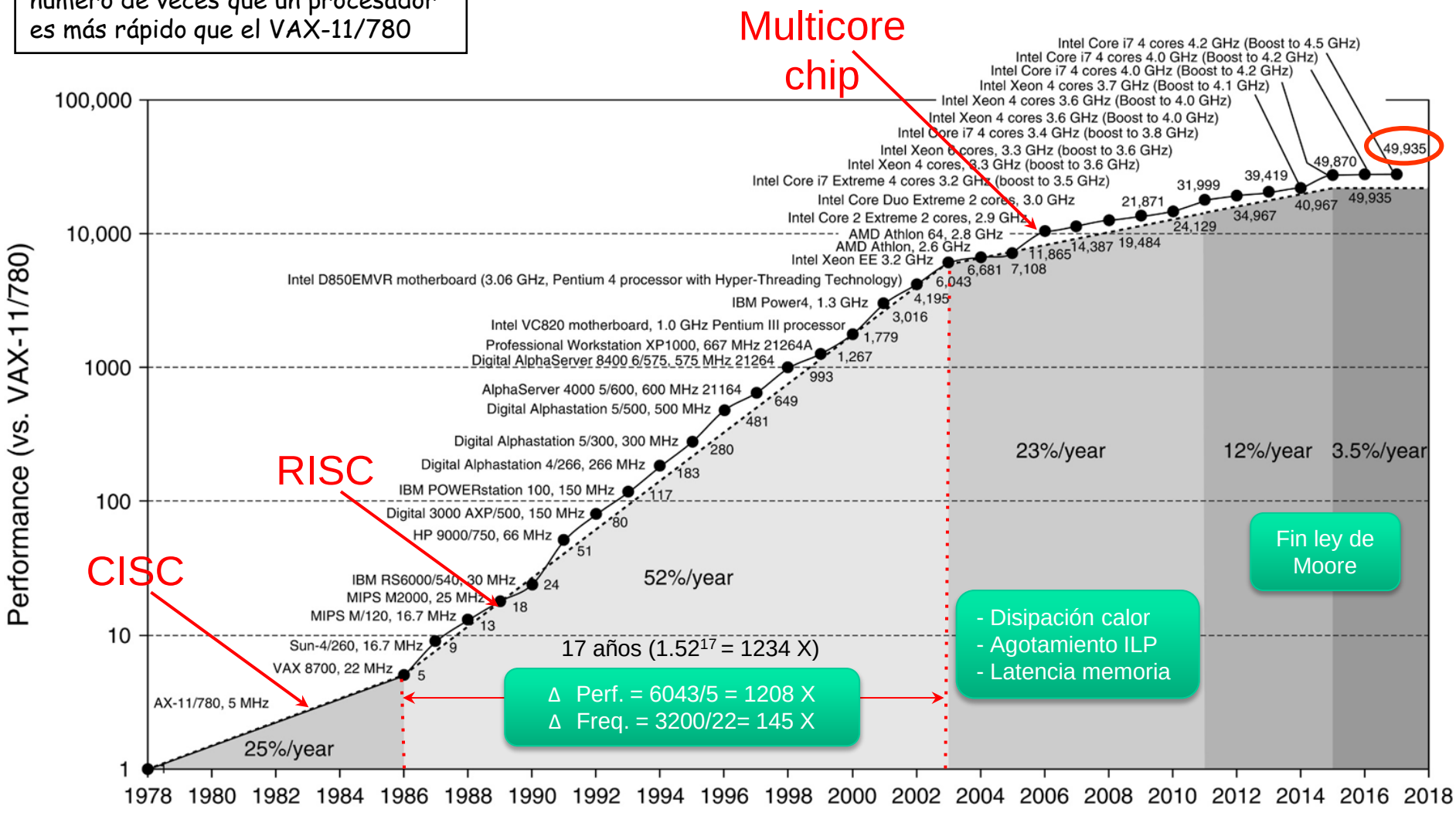
μ-arquitectura + hardware

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

- Paralelismo
  - 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...
  - 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
  - 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
  - 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)
  - 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
  - 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!
  - Se utiliza el test LINPACK (https://icl.utk.edu/hpl/)
    - 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
  - Rendimiento (Rmax):
    - Op. en PF ejecutadas en test LINPACK / tiempo de ejecución: FLOPS
    - Lista ordenada por Rmax en sentido decreciente
- GREEN500
  - Publicada desde 2007, ordena los sistemas en sentido decreciente de eficiencia energética:
    - Eficiencia: Rmax / Potencia consumida (unidad habitual GFLOPS/W, equivale a GFLOP/J)
- HPCG500 (el test es High-Performance Conjugate Gradient)

Evolución y tendencias: supercomputadores

TOP500: junio 2024

PRESENTED BY

TOP 500 The List.

ICL INNOVATIVE COMPUTING LABORATORY

BERKELEY LAB Lawrence Berkeley National Laboratory

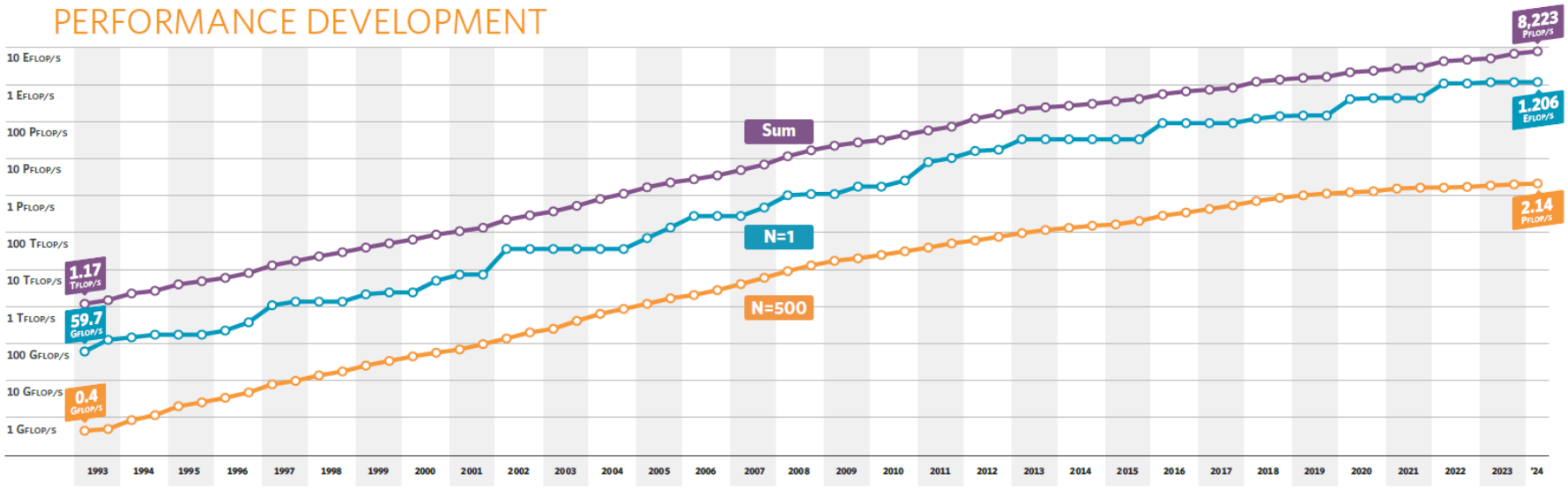
ISC GROUP Moving Forward.

FIND OUT MORE AT top500.org

MAY 2024

			SITE	COUNTRY	CORES	RMAX PFLOP/S	POWER MW
1	**Frontier**	HPE Cray EX235a, AMD Opt 3rd Gen EPYC (64C 2GHz), AMD Instinct MI250X, Slingshot-11	DOE/SC/ORNL	USA	8,699,904	1,206.0	22.7
2	**Aurora**	HPE Cray EX - Intel Exascale Compute Blade, Xeon CPU Max 9470 (52C 2.4GHz), Intel Data Center GPU Max, Slingshot-11	DOE/SC/ANL	USA	9,264,128	1,012.0	38.7
3	**Eagle**	Microsoft NDv5, Xeon Platinum 8480C (48C 2GHz), NVIDIA H100, NVIDIA Infiniband NDR	Microsoft Azure	USA	1,123,200	561.2	
4	**Fugaku**	Fujitsu A64FX (48C, 2.2GHz), Tofu Interconnect D	RIKEN R-CCS	Japan	7,630,848	442.0	29.9
5	**LUMI**	HPE Cray EX235a, AMD Opt 3rd Gen EPYC (64C 2GHz), AMD Instinct MI250X, Slingshot-11	EuroHPC/CSC	Finland	2,220,288	379.7	6.01

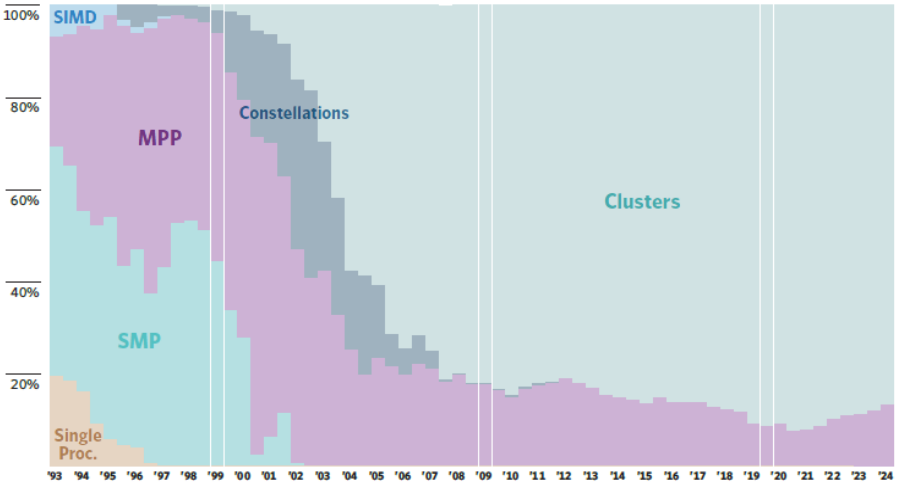
PERFORMANCE DEVELOPMENT



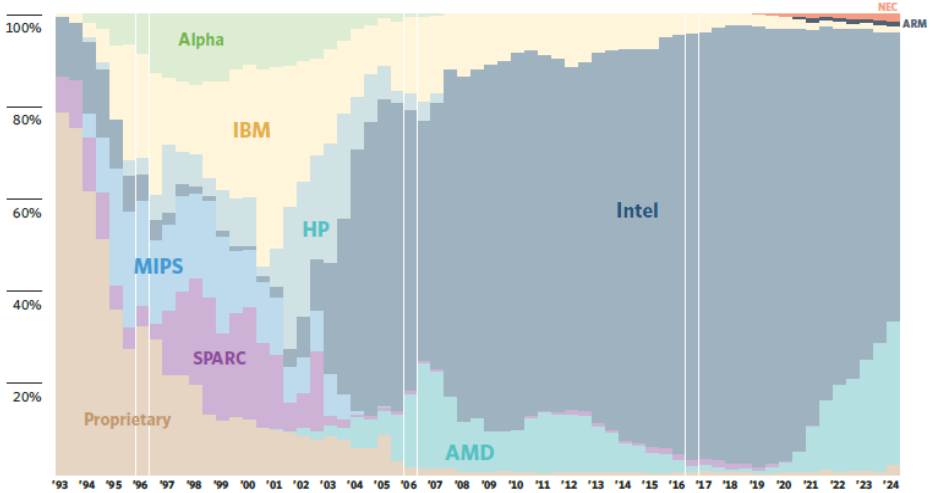
Evolución y tendencias: supercomputadores

TOP500: junio 2024. Actores principales.

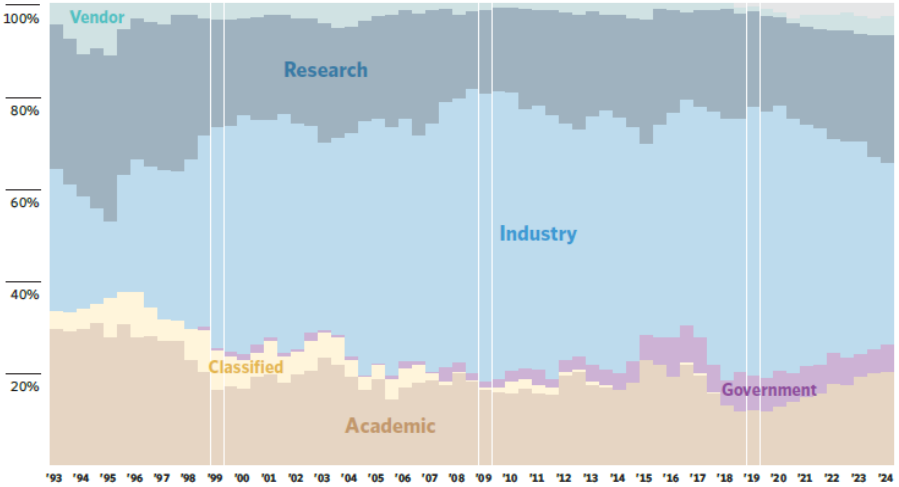
ARCHITECTURES



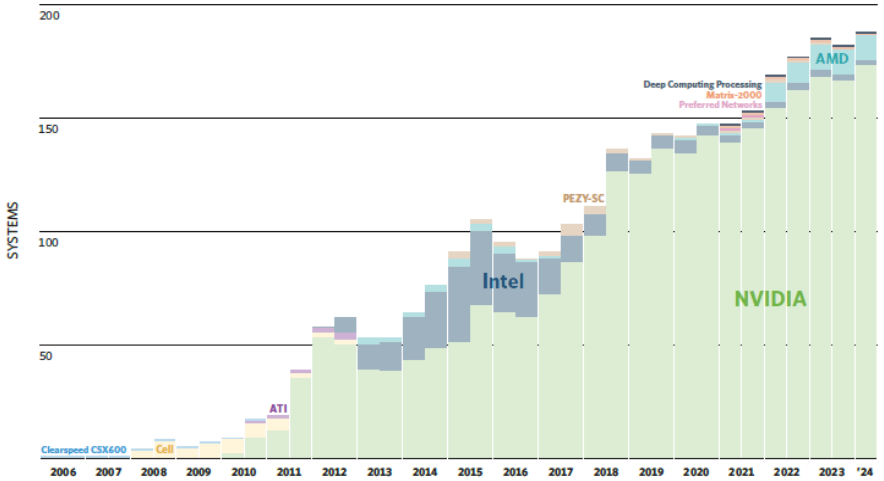
CHIP TECHNOLOGY



INSTALLATION TYPE



ACCELERATORS/CO-PROCESSORS



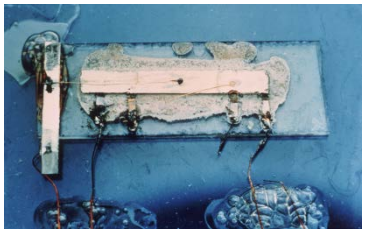
Evolución y tendencias: tecnología

- 1946 ENIAC 400 op/seg (18000 Valvulas)

Transistor (47) PN 56

- 1957 Transistor: de 10^3 a 10^4 op/s
  - DEC PDP-1 (1957)
  - IBM 7090 (1960)

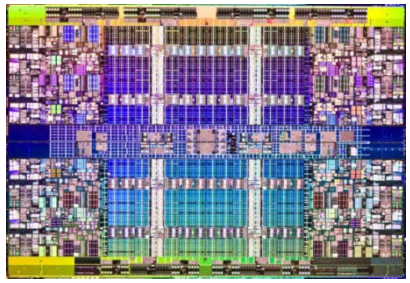
- 1965 CI: de 10^5 a 10^6 op/s
  - IBM System 360 (1965)
  - DEC PDP-8 (1965)



CI (58) PN2000

- 1971 Microprocesador
  - Intel 4004

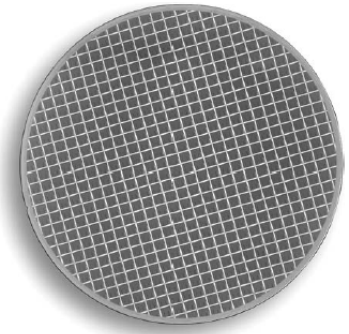
- 2003 más de $3x10^{13}$ op/s



Intel Xeon 7500, 8c, 16Th

- 2008 > 10^{15} FLOP/s (1 PFLOPS)

- Jun 2024 (TOP500)
  - 1º Frontier:
    8.699.904 cores; 1,19 EFLOPS (EPYC 64C 2 GHz + AMD MI250X)
  - 8º Mare Nostrum 4 (2023)
    663.040 cores; 175 PFLOPS (Xeon Platinum 8460Y 2.3GHz + NVIDIA H100)



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

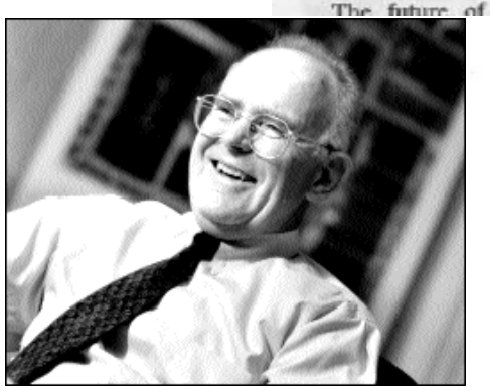


Fig. 2.

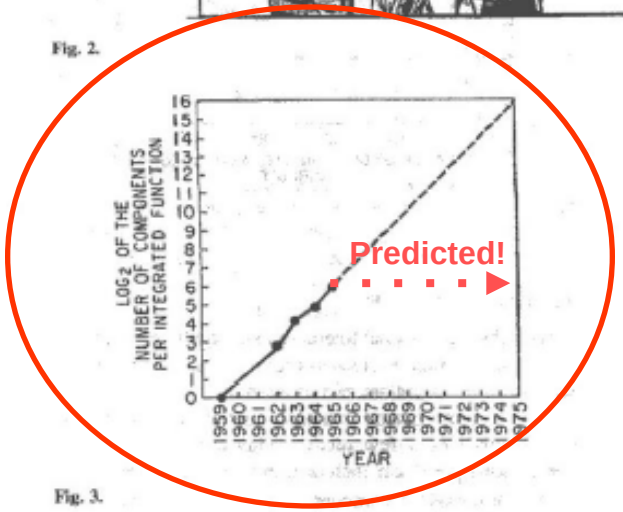


Fig. 3.

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 p
aged and interconnected. The availability of large funct
combined with functional design and construction, sh
allow the manufacturer of large systems to design
construct a considerable variety of equipment both ra
and economically.

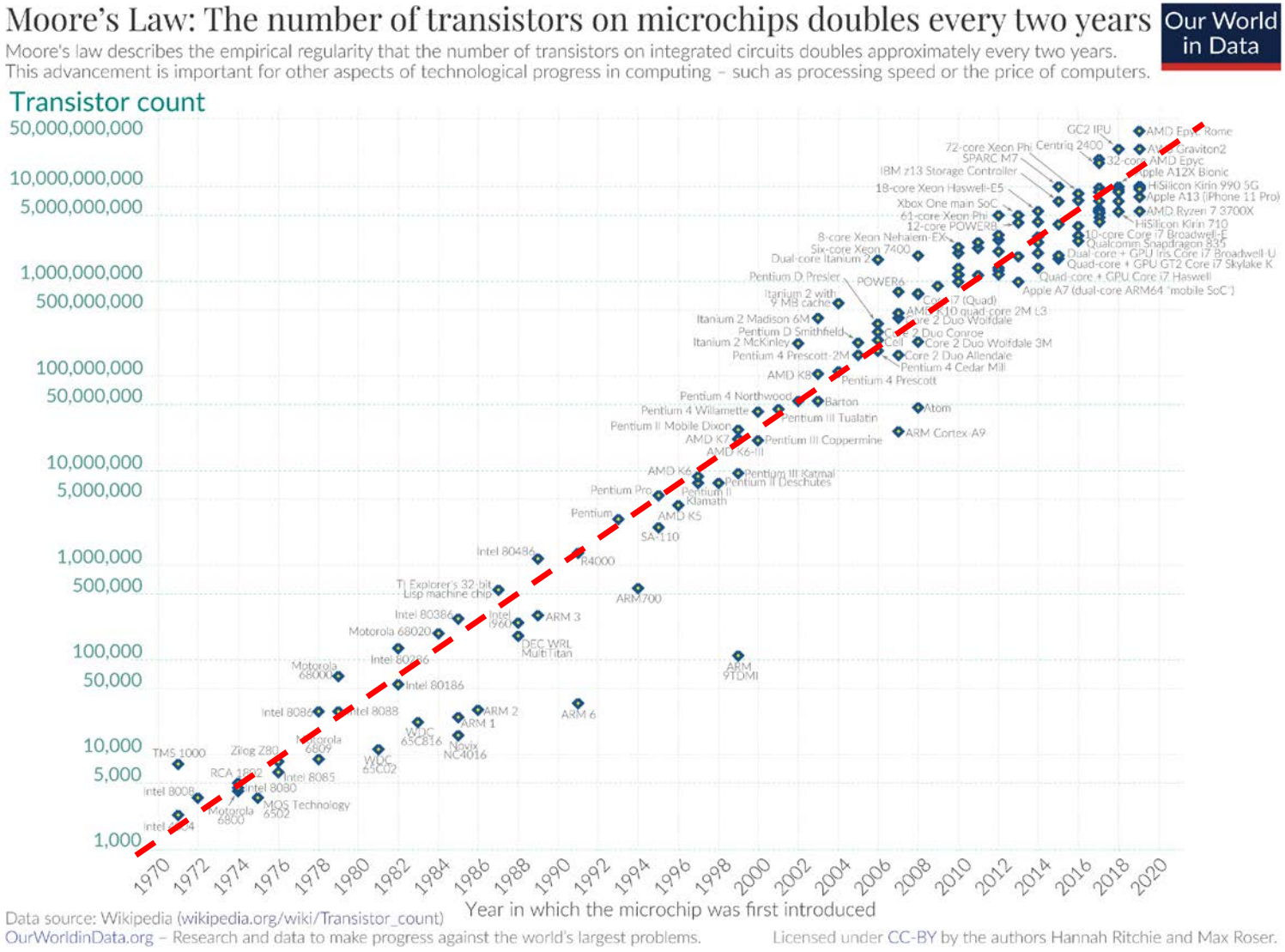
IX. Linear Circuitry

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

Carver A. Mead 1979

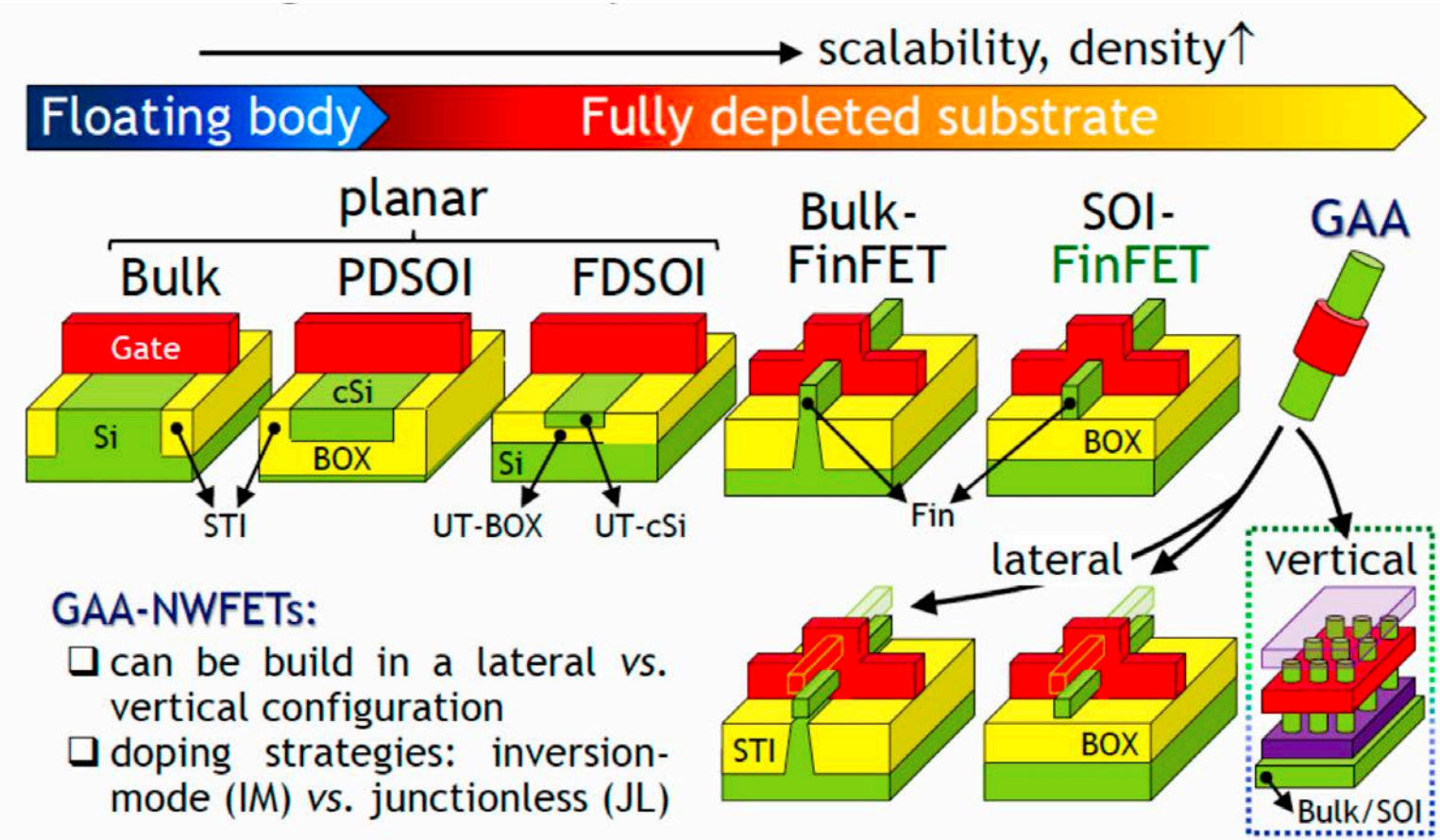
Evolución y tendencias: tecnología

- Ley de Moore se ha cumplido...por más tiempo del previsto



Evolución y tendencias: tecnología

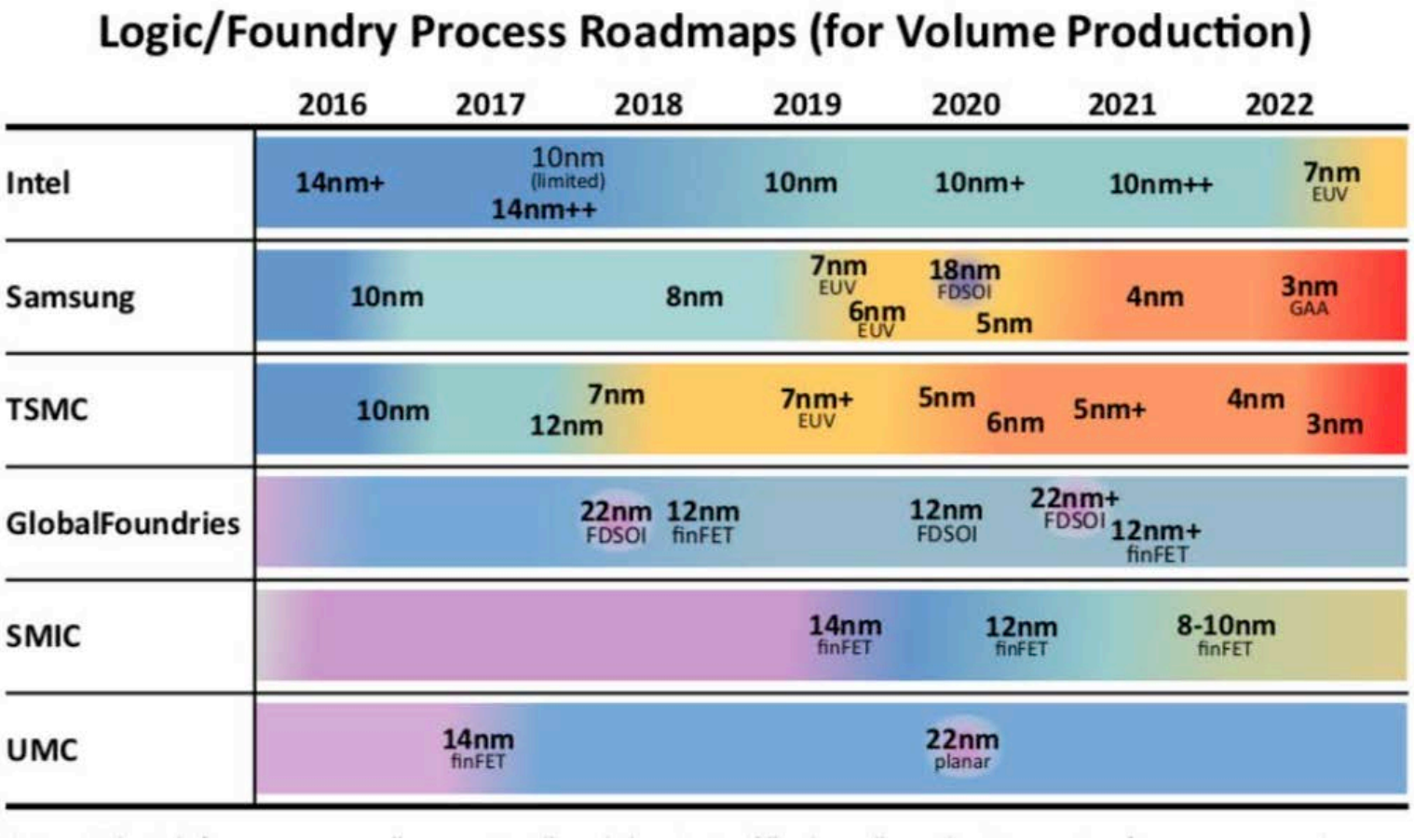
- Problema de escalado de los transistores



Fuente: IRDS 2022

Evolución y tendencias: tecnología

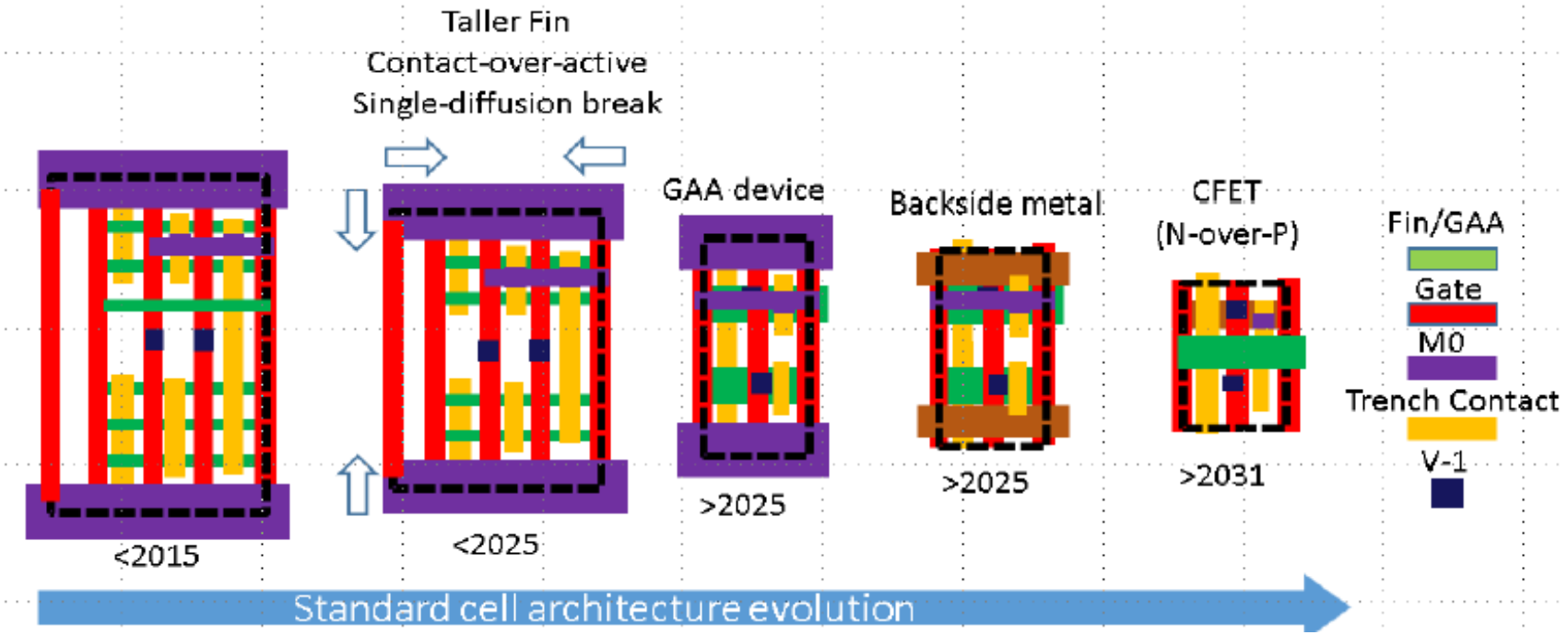
- Problema de escalado de los transistores



Fuente: THE INTERNATIONAL ROADMAP FOR DEVICES AND SYSTEMS: 2022

Evolución y tendencias: tecnología

- Todavía hay cierto margen para reducir la celda



Fuente: IRDS 2023

- Pero >2031 necesario 3D VLSI y otras soluciones
- ¿Final Ley de Moore?

La Ley de Moore: final del escalado de Dennard * (2004)

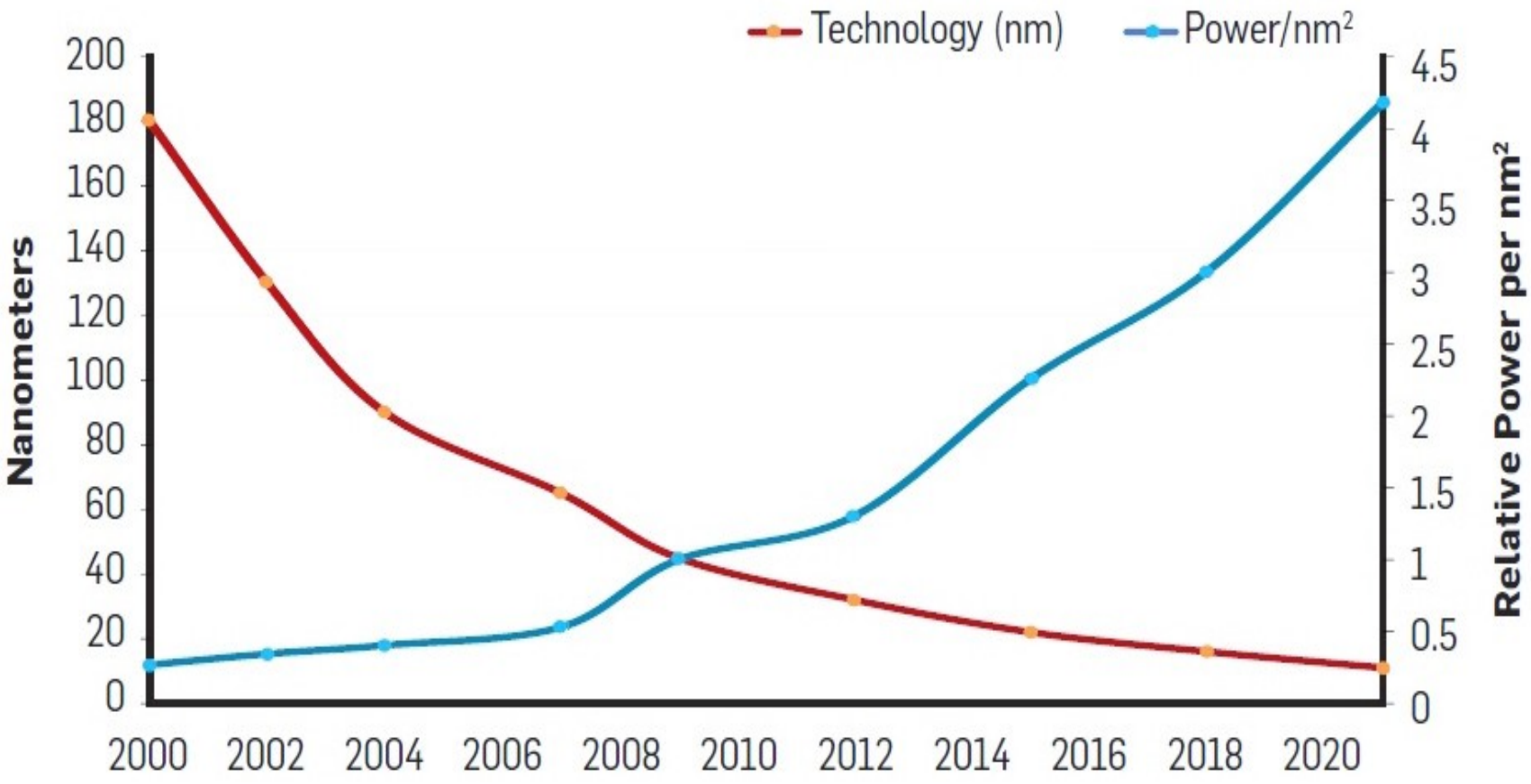
Parameter (scale factor = s)	Classic Scaling	Current Scaling
Dimensions	1/s	1/s
Voltage	1/s	1
Current	1/s	1/s
Power	$1/s^2$	1/s
Power Density	1	**s**
Delay R*C = V/I * C	1/s	~1

Por debajo de 65nm
Transistores más pequeños,
pero no más rápidos
ni más eficientes

* This is the MOSFET scaling theory by Robert Dennard et al. (IEEE JSCC 74) proposing to maintain a constant electric field throughout the device as it shrinks in size . Table source: Krisztián Flautner “From niche to mainstream: can critical systems make the transition?”

La Ley de Moore: problemas...

1er problema: consumo "Power Wall"



Punto de inflexión[2]: 65 nm

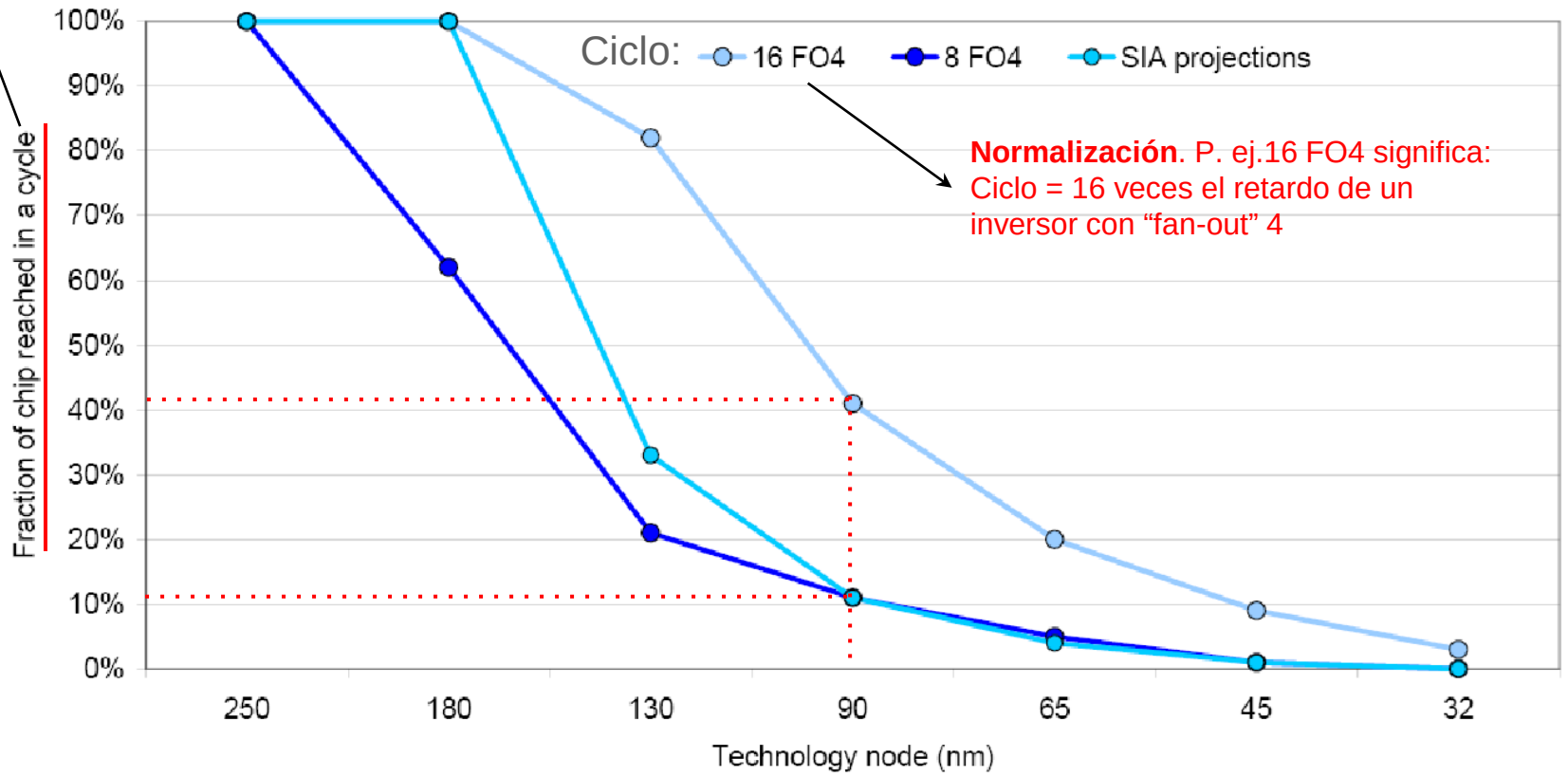
Fuente: Hennesy, J., Pattersson, D.A. *A new Golden Age for Computer Architecture.* Communications of the ACM. 2019

La Ley de Moore: problemas...

2º problema: retardo interconexiones.

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

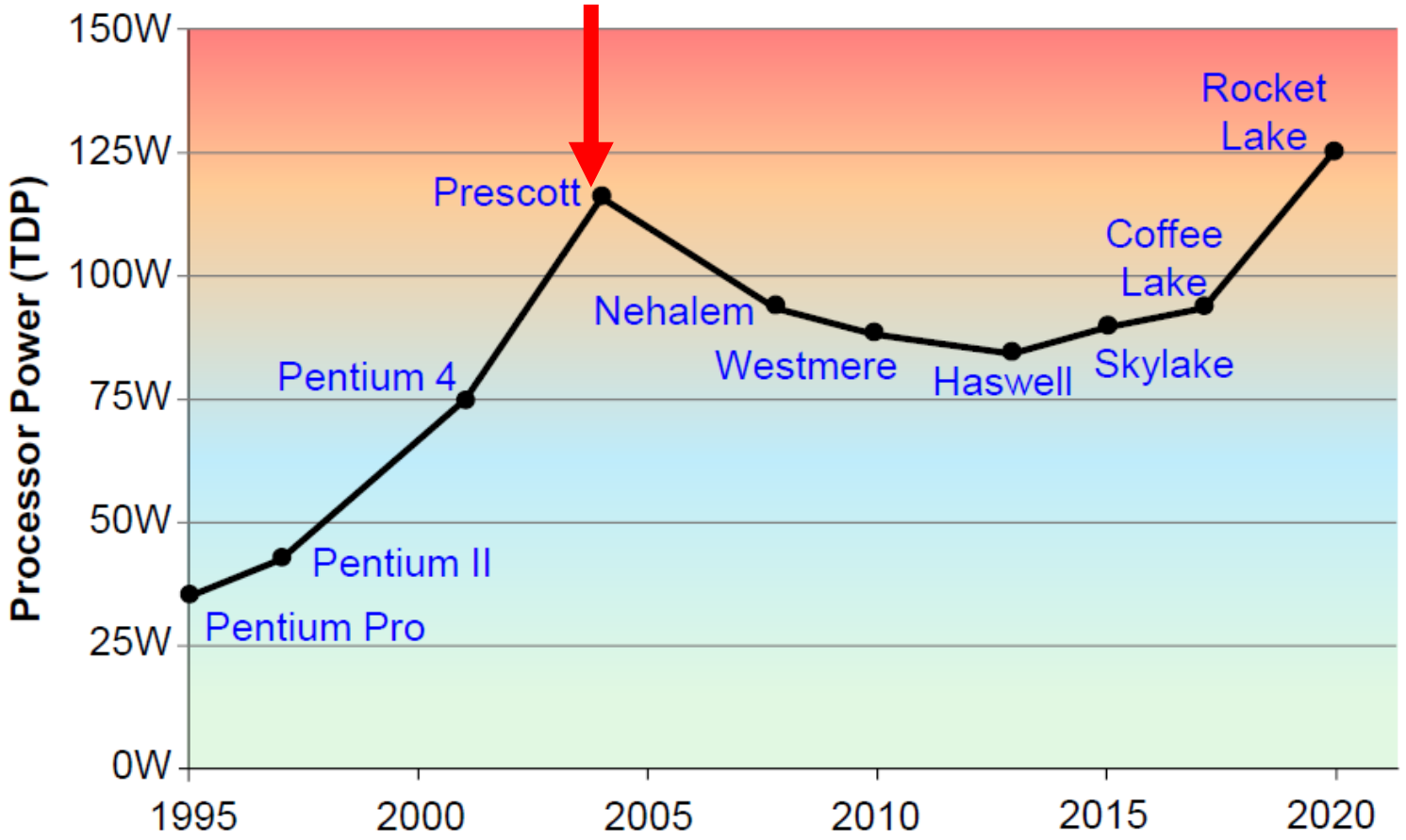
Pero...¿cuánto dura un ciclo?: **Normalizar**



Normalización. P. ej.16 FO4 significa: Ciclo = 16 veces el retardo de un inversor con “fan-out” 4

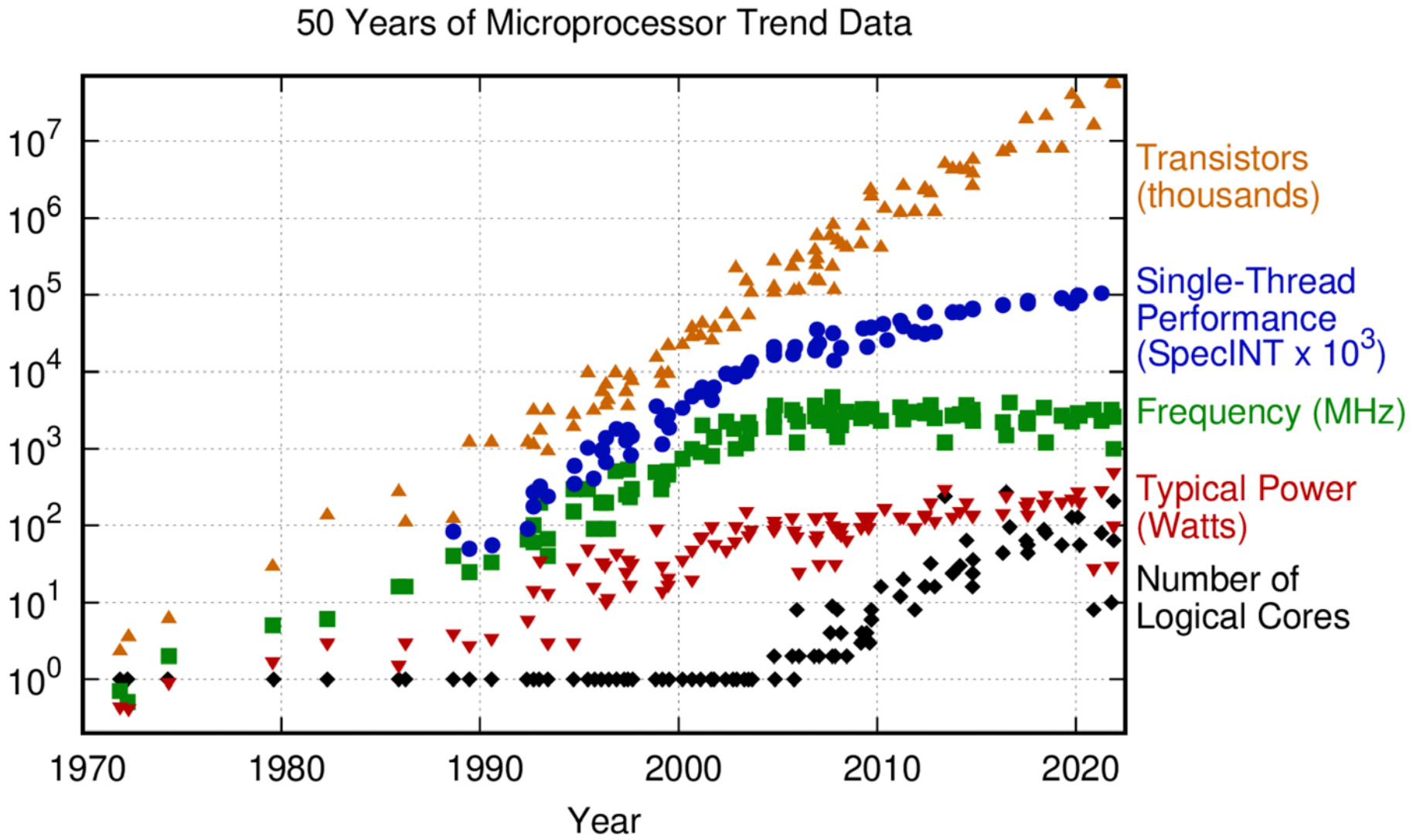
Multi - Many cores

La Ley de Moore: el punto de inflexión



Decktop/laptop Alder Lake (12G), Raptor Lake (8+8) (13G) (253W)
Server Xeon Sapphire Rapids 360w.Gpu Ponte Vecchio 600w

Evolución y tendencias: tecnología



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

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

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

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

- 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
Latency (ns) *	320	313	200	76	50	15	4

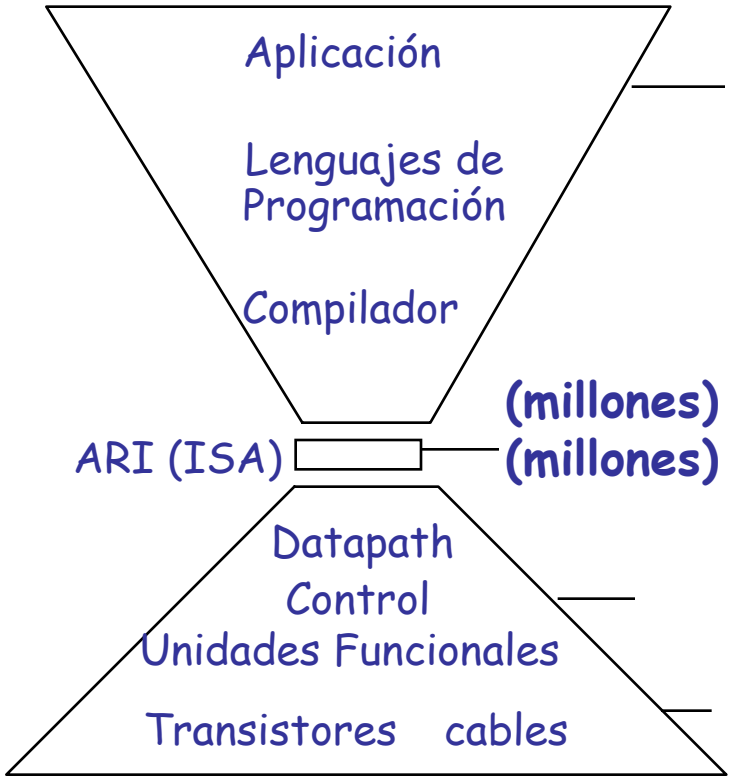
x 32K

x 80

Fig 1.10 H&P 6th ed (detalle)

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

❑ Medidas del rendimiento



Respuestas por mes, hora, segundo
Operaciones por segundo TPC

(millones) de Instrucciones por segundo: MIPS
(millones) de (FP) operaciones por segundo: MFLOP/s

Megabytes por segundo

Ciclos por segundo (frecuencia de reloj)

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

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

✓ 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:

$$CPI = Ciclos / NI$$

✓ 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 : nº 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*0,5 + 2*0,2 + 2*0,1 + 2*0,2 = 1.5

Rendimiento

- Rendimiento global del computador: programas de prueba (benchmarks)
  - 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?
  - Programas reales típicos con cargas de trabajo fijas (actualmente es la medida más aceptada): **SPEC CPU**
  - Otros
    - HPC: LINPACK, SPEChpc, Nas Parallel Benchmark
    - Servidores: ~~SPECweb~~, SPEC_Virt_SC, SPECSFS(File servers), TPC-C, SPECjbb (Java)
    - Graficos: SPECviewperf(OpenGL), SPECapc(aplicaciones 3D)
    - Winbench, EEMBC, Embench
    - MLPerf

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 Perfomance Evaluation Corporation)
  - Desde 1989 define series de programas de prueba ("benchmark suites") que representan aplicaciones reales de diferentes tipos.
  - 6 generaciones de SPEC CPU: 89, 92, 95, 2000, 2006 y 2017.
    - Cada una más compleja que la anterior.
  - En cada generación los programas que integran la suite SPEC son diferentes de los de la generación anterior (aunque alguno puede permanecer)
  - ¿Por qué es preciso definir una nueva suite cada cierto tiempo?
- ¿Qué se mide en SPEC CPU 2017?
  - 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
  - No se mide el rendimiento de:
    - redes, hw gráfico, bibliotecas Java, E/S... Otros SPEC para ello
  - 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).

Rendimiento: SPEC

- ¿Qué tipos de programas se usan como benchmarks?
  - Enteros: Compiladores (GNU C) e intérpretes (Perl), simulación de eventos discretos, compresión de vídeo y datos, IA (p.ej. Sudoku)...
  - 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

Rendimiento: SPEC

- ¿Cómo se calculan las métricas SPECspeed?
  - Para cada bechmark, i, se mide el tiempo de ejecución en el sistema a evaluar (SAE) y se calcula la razón:

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

- ¿Y en SPECrate?
  - Para cada benchmark, i, se lanza a ejecutar un nº de copias determinado por el usuario, nc, y se calcula:

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

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

Rendimiento: SPEC

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

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

  - 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):

$$SPECspeed2017\_int\_base = \sqrt[10]{r_1 \times r_2 \times \ldots \times r_{10}}$$

Rendimiento: SPEC

- ¿Qué ventaja aporta el uso de la MG?
  - 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

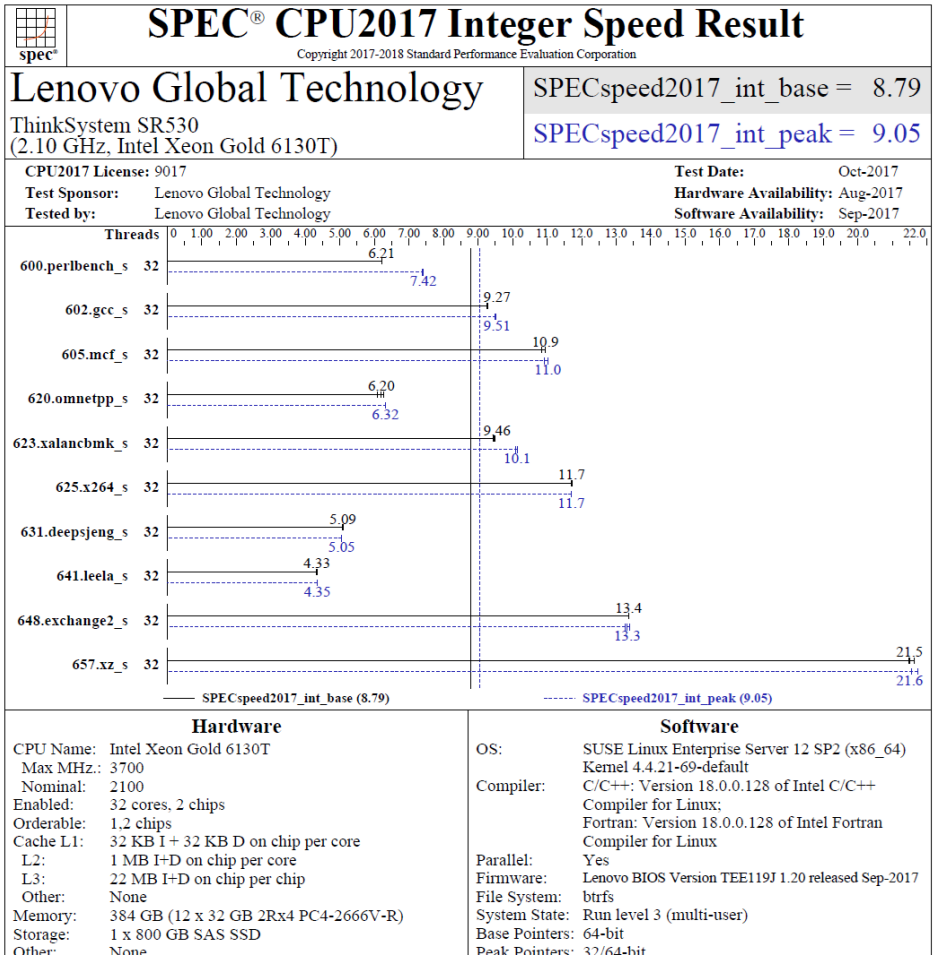
$$Rendimiento\ A = \sqrt[2]{\frac{R1}{A1} \times \frac{R2}{A2}} \qquad Rendimiento\ B = \sqrt[2]{\frac{R1}{B1} \times \frac{R2}{B2}}$$

  - 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:

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

Rendimiento: SPEC

¿Cómo se presentan los resultados?



SPEC® CPU2017 Integer Speed Result

Copyright 2017-2018 Standard Performance Evaluation Corporation

Lenovo Global Technology

ThinkSystem SR530
(2.10 GHz, Intel Xeon Gold 6130T)

SPECspeed2017_int_base = 8.79
SPECspeed2017_int_peak = 9.05

CPU2017 License: 9017
Test Sponsor: Lenovo Global Technology
Tested by: Lenovo Global Technology

Test Date: Oct-2017
Hardware Availability: Aug-2017
Software Availability: Sep-2017

Threads		base	peak
600.perlbench_s	32	6.21	7.42
602.gcc_s	32	9.27	9.51
605.mcf_s	32	10.9	11.0
620.omnetpp_s	32	6.20	6.32
623.xalancbmk_s	32	9.46	10.1
625.x264_s	32	11.7	11.7
631.deepsjeng_s	32	5.09	5.05
641.leela_s	32	4.33	4.35
648.exchange2_s	32	13.4	13.3
657.xz_s	32	21.5	21.6

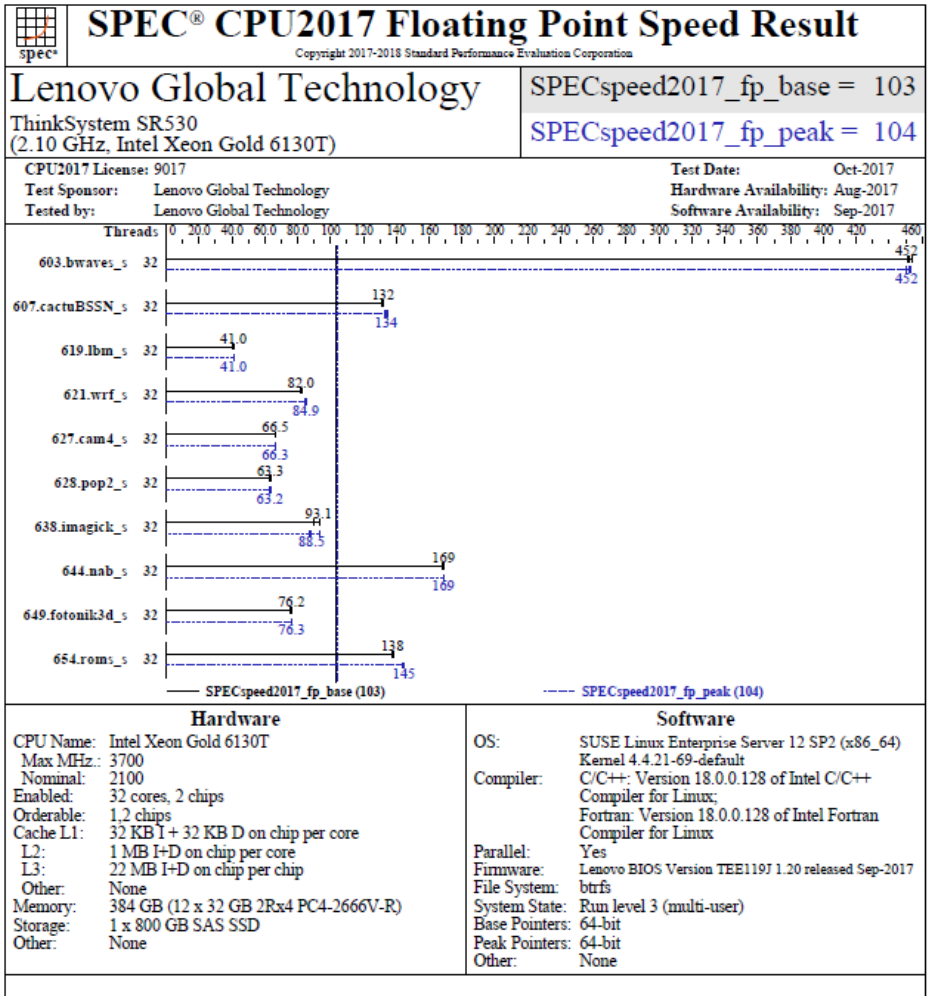
SPECspeed2017_int_base (8.79) — SPECspeed2017_int_peak (9.05)

Hardware

CPU Name:	Intel Xeon Gold 6130T
Max MHz.:	3700
Nominal:	2100
Enabled:	32 cores, 2 chips
Orderable:	1,2 chips
Cache L1:	32 KB I + 32 KB D on chip per core
L2:	1 MB I+D on chip per core
L3:	22 MB I+D on chip per chip
Other:	None
Memory:	384 GB (12 x 32 GB 2Rx4 PC4-2666V-R)
Storage:	1 x 800 GB SAS SSD
Other:	None

Software

OS:	SUSE Linux Enterprise Server 12 SP2 (x86_64) Kernel 4.4.21-69-default
Compiler:	C/C++: Version 18.0.0.128 of Intel C/C++ Compiler for Linux; Fortran: Version 18.0.0.128 of Intel Fortran Compiler for Linux
Parallel:	Yes
Firmware:	Lenovo BIOS Version TEE119J 1.20 released Sep-2017
File System:	btrfs
System State:	Run level 3 (multi-user)
Base Pointers:	64-bit
Peak Pointers:	32/64-bit



SPEC® CPU2017 Floating Point Speed Result

Copyright 2017-2018 Standard Performance Evaluation Corporation

Lenovo Global Technology

ThinkSystem SR530
(2.10 GHz, Intel Xeon Gold 6130T)

SPECspeed2017_fp_base = 103
SPECspeed2017_fp_peak = 104

CPU2017 License: 9017
Test Sponsor: Lenovo Global Technology
Tested by: Lenovo Global Technology

Test Date: Oct-2017
Hardware Availability: Aug-2017
Software Availability: Sep-2017

Threads		base	peak
603.bwaves_s	32	452	452
607.cactuBSSN_s	32	132	134
619.lbm_s	32	41.0	41.0
621.wrf_s	32	82.0	84.9
627.cam4_s	32	66.5	66.3
628.pop2_s	32	63.3	63.2
638.imagick_s	32	93.1	88.5
644.nab_s	32	169	169
649.fotonik3d_s	32	76.2	76.3
654.roms_s	32	138	145

SPECspeed2017_fp_base (103) — SPECspeed2017_fp_peak (104)

Hardware

CPU Name:	Intel Xeon Gold 6130T
Max MHz.:	3700
Nominal:	2100
Enabled:	32 cores, 2 chips
Orderable:	1,2 chips
Cache L1:	32 KB I + 32 KB D on chip per core
L2:	1 MB I+D on chip per core
L3:	22 MB I+D on chip per chip
Other:	None
Memory:	384 GB (12 x 32 GB 2Rx4 PC4-2666V-R)
Storage:	1 x 800 GB SAS SSD
Other:	None

Software

OS:	SUSE Linux Enterprise Server 12 SP2 (x86_64) Kernel 4.4.21-69-default
Compiler:	C/C++: Version 18.0.0.128 of Intel C/C++ Compiler for Linux; Fortran: Version 18.0.0.128 of Intel Fortran Compiler for Linux
Parallel:	Yes
Firmware:	Lenovo BIOS Version TEE119J 1.20 released Sep-2017
File System:	btrfs
System State:	Run level 3 (multi-user)
Base Pointers:	64-bit
Peak Pointers:	64-bit
Other:	None

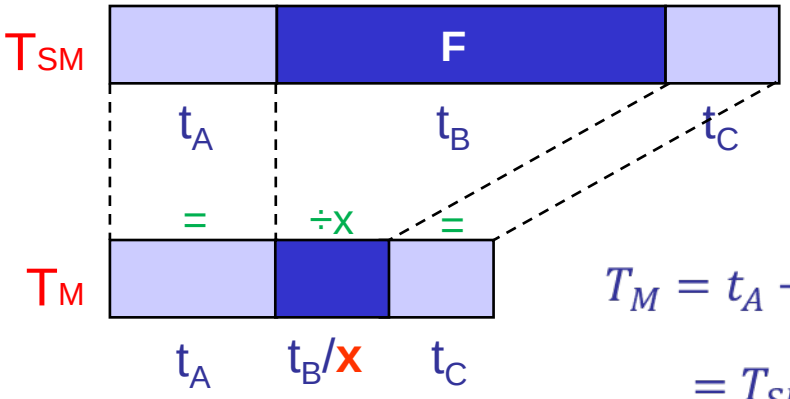
Rendimiento: Un principio simple

- 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 x veces más rápido una parte del código que consume una fracción F del tiempo total de ejecución (ver gráfica).

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



$T_{SM} = t_A + t_B + t_C \qquad F = \dfrac{t_B}{T_{SM}} \qquad 1\text{-}F = \dfrac{t_A + t_C}{T_{SM}}$

$$T_M = t_A + t_C + \frac{t_B}{x} = \quad \{\text{sustituyendo}\}$$

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

Rendimiento: Un principio simple

- **La Ley *Amdahl***

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

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

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

$T_M = T_{SM}$ x (0.9 + 0.1 / **2**) = 0.95 x T_{SM}

$$Speedup = \frac{1}{0.95} = 1.053$$

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.

$T_M = T_{SM}$ x (0.1 + 0.9 / **100**) = 0.109 x T_{SM}

$$Speedup = \frac{1}{0.109} = 9.17$$

El uso de 100 procesadores sólo multiplica la velocidad por 9.17

Rendimiento: Un principio simple

❑ Concepto de eficiencia (E)

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

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:

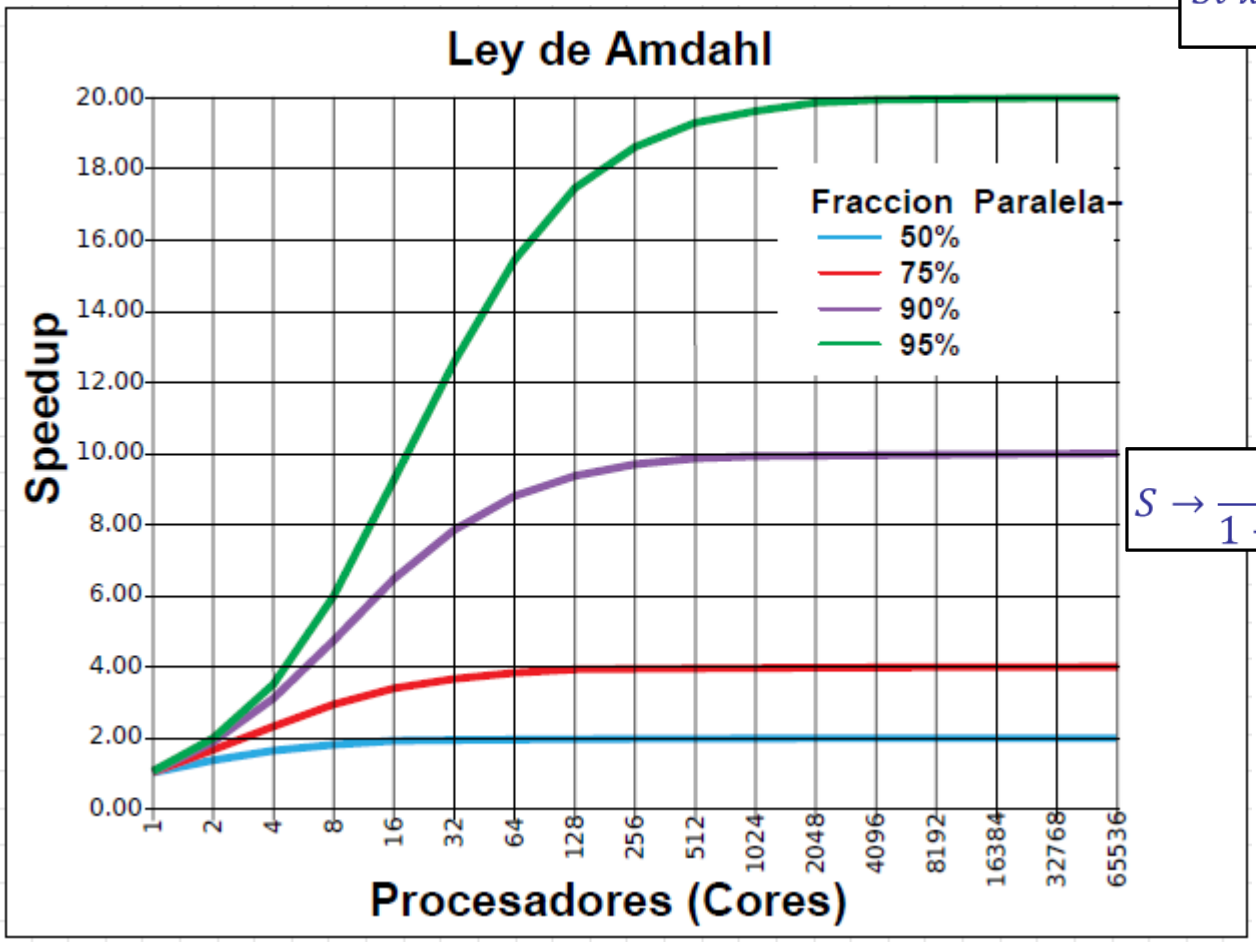
1. La fracción no paralelizable de un cálculo, (1-F), limita seriamente el Speedup, incluso cuando esta fracción es pequeña.
2. A partir de cierto punto, aumentar mucho el nº de procesadores apenas mejora el Speedup, por lo que se degradada mucho la Eficiencia.

Rendimiento: Un principio simple

Everyone knows Amdahl's Law but quickly forgets it!

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

$$Si\ x \to \infty \ \Rightarrow S \to \frac{1}{1-F}$$



Ejemplo: F=0,9

$$S \to \frac{1}{1-F} = \frac{1}{1-0{,}9} = 10$$

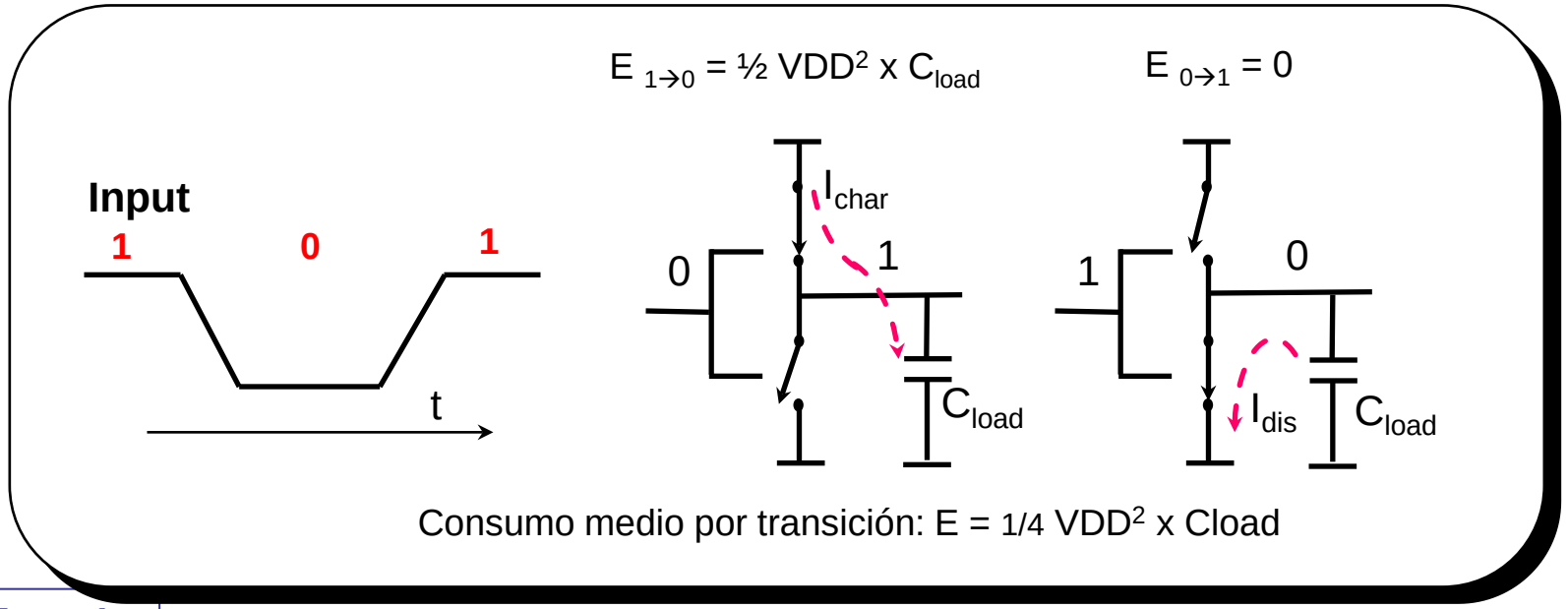
Consumo: Potencia y Energía

- El consumo de potencia eléctrica: uno de los principales retos en el diseño actual de computadores
  - Problema: El consumo se transforma en calor a disipar
- Energía:
  - La energía total que consume el sistema.
- Potencia: energía consumida por unidad de tiempo
  - Consumo máximo de potencia posible (pico). Diseño para el máximo?
  - Consumo de potencia sostenido. Thermal Design Power (TDP).
    - Determina la potencia del sistema de refrigeración y de la fuente de alimentación
    - Está por debajo de la potencia "pico" y por encima de la "media"
- Energía y eficiencia energética: ¿Cuál es la métrica más adecuada para comparar el rendimiento energético de dos procesadores? Energía vs. Potencia.

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

Consumo: Energía

- Energía estática: se consume aunque los transistores no estén conmutando (leakage current)
- Energía dinámica: Consumida para la conmutación de los transistores.
  - Transistores más pequeños, rápidos y de menor consumo (menor energía por conmutación). Pero... mayor cantidad en el chip!
- E. dinámica consumida en transiciones de la entrada 1→0→1



Consumo: Potencia y Energía

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

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

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

- Técnicas de Escalado Dinámico de Voltaje y Frecuencia (DVFS). Reducir VDD. Impacto:

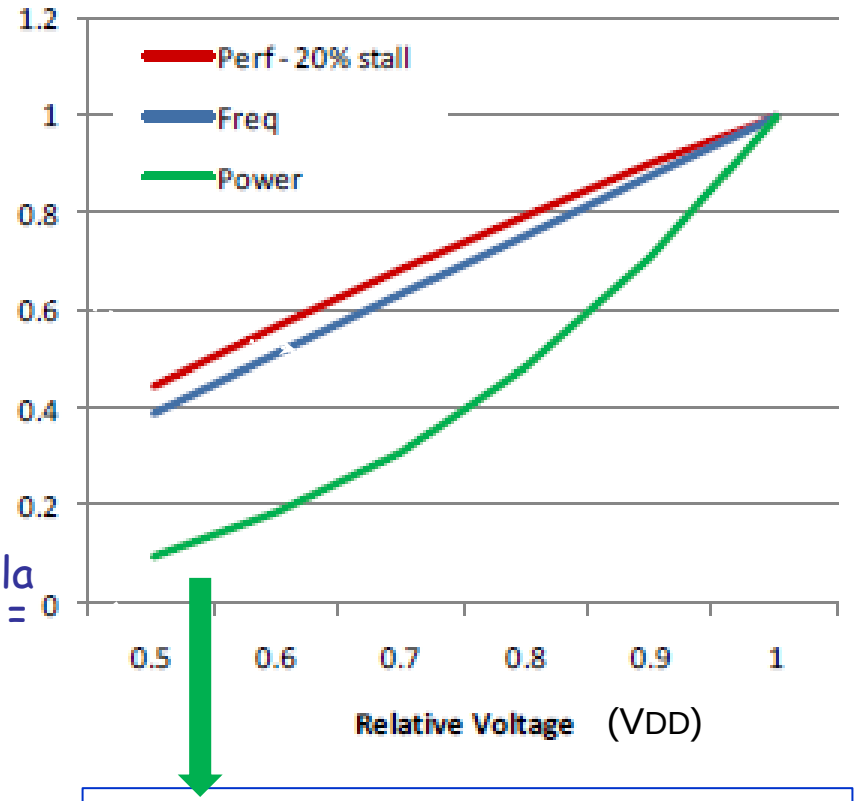
  Cuadrático sobre el consumo de energía!

  Pero además...

  Lineal sobre frecuencia máxima y rendimiento (aprox.)

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

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



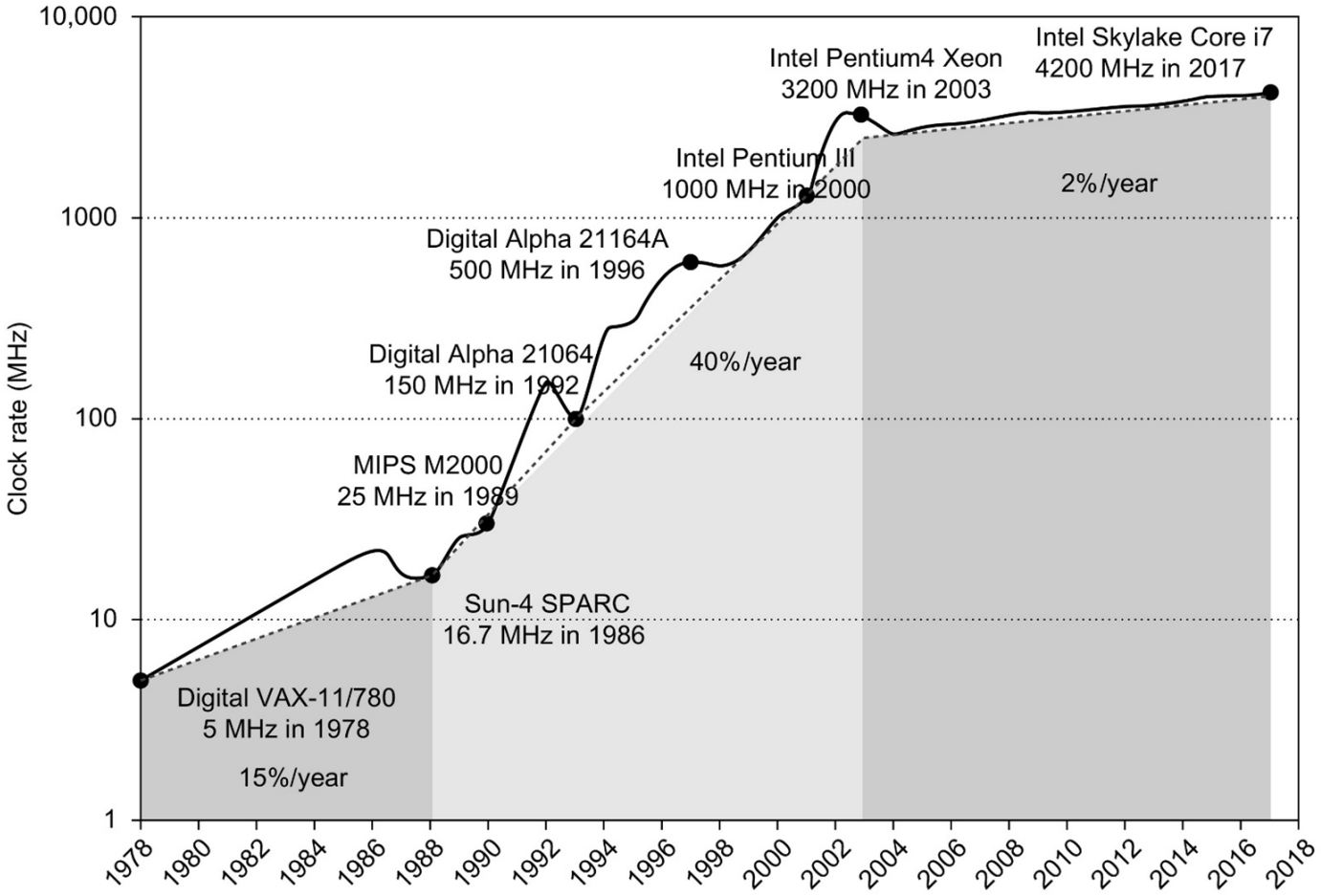
Incialmente: VDD=1. Freq=1, Perf=1, Power=1

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

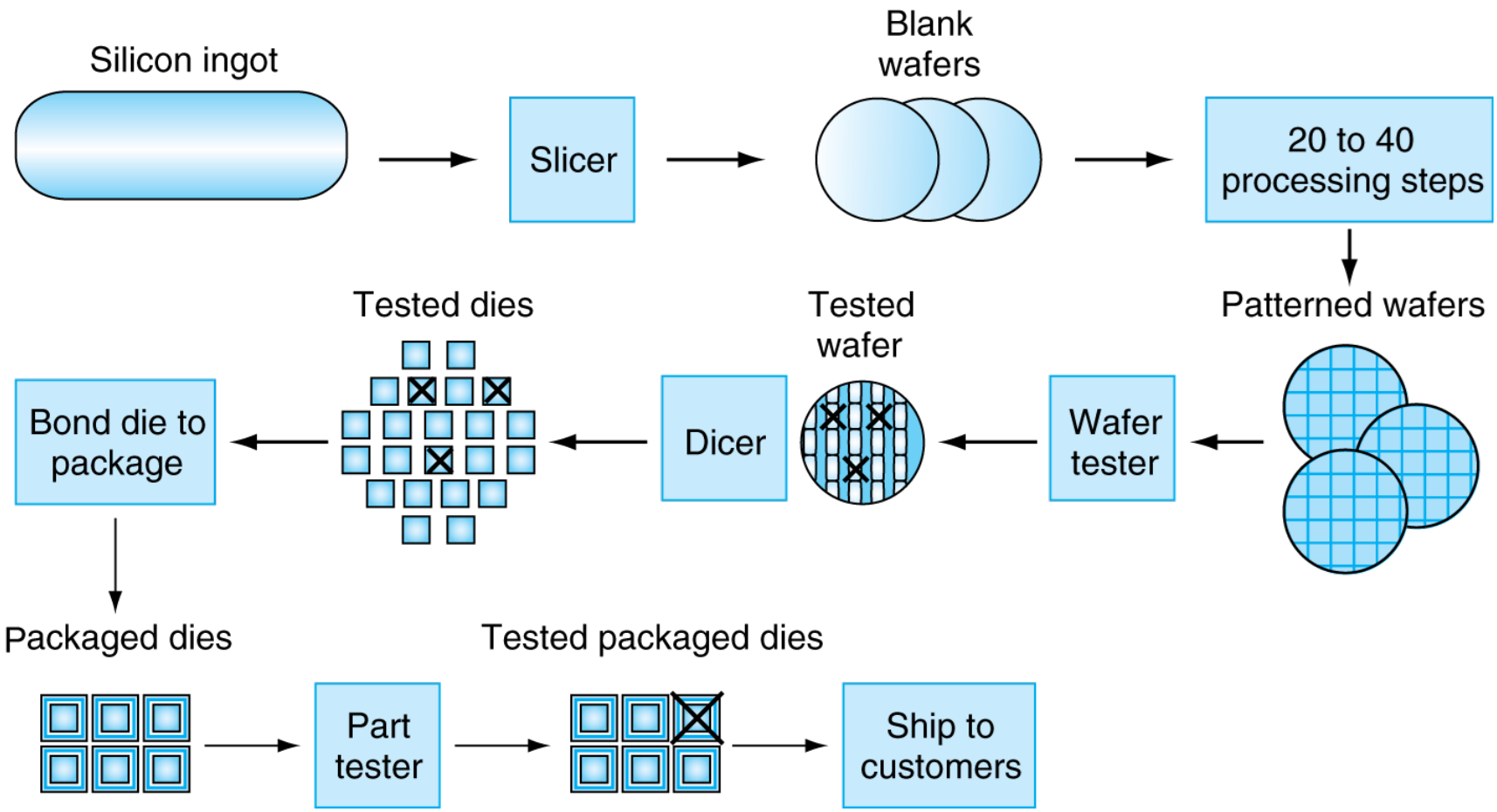
Añadir 7 cores ➔ Power ≈ 1, Perf ≈ 4 !!

Consumo: Potencia y Energía

- Evolución del "clock"
  - Intel 80386 2W
  - Intel Core i7 6700K / 4 GHz processor, 95 W (a disipar en 1.5 x 1.5 cm)
  - >100W problema 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

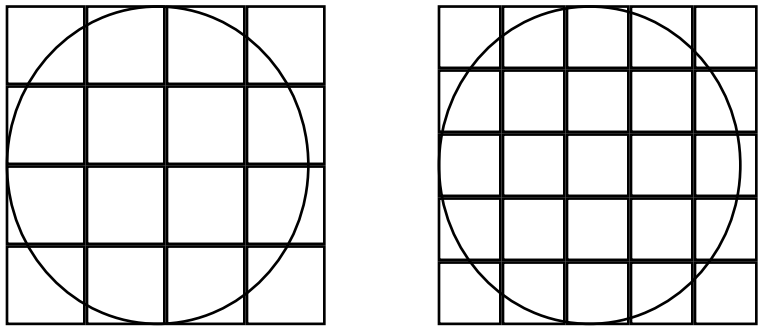
$$\text{coste de CI} = \frac{\text{Die coste} + \text{Testing coste} + \text{Packaging coste}}{\text{Final test yield}}$$

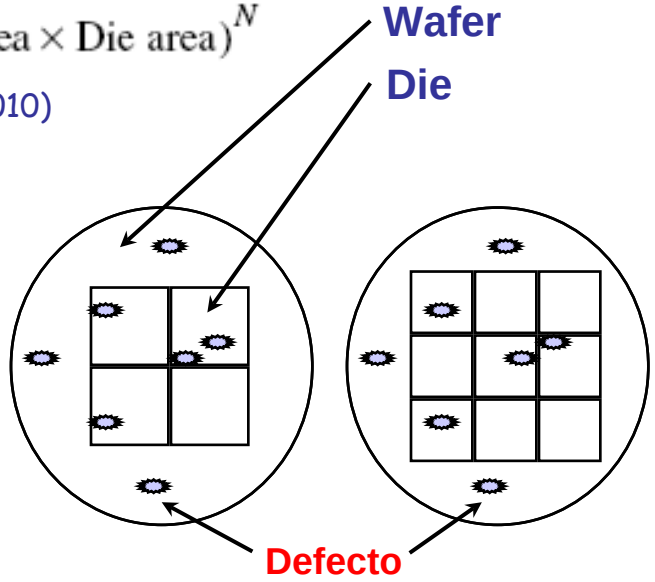
$$\text{Die coste} = \frac{\text{coste del Wafer}}{\text{Dies por Wafer} * \text{Die yield}}$$

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 (32 nm, 2009)





El coste de CI (Die) ≈ f(área del die)²

Coste-Rendimiento-Consumo

- Tres servidores DELL PowerEdge
  - SW: IBM Java VM / MS Windows 2008 EE X64

Lectura: Sección 1.10 de H&P 5th ed

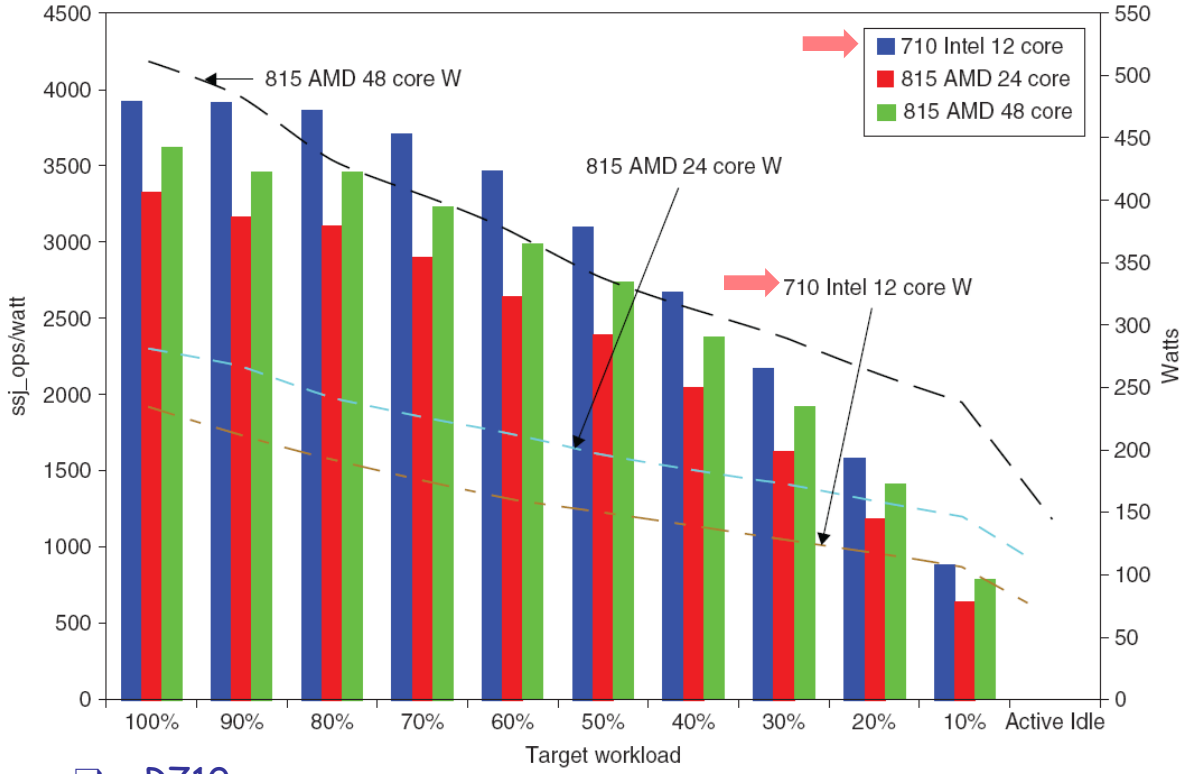
	System 1		System 2		System 3	
Component		Cost (% Cost)		Cost (% Cost)		Cost (% Cost)
Base server	PowerEdge R710	$653 (7%)	PowerEdge R815	$1437 (15%)	PowerEdge R815	$1437 (11%)
Power supply	570 W		1100 W		1100 W	
Processor	Xeon X5670	$3738 (40%)	Opteron 6174	$2679 (29%)	Opteron 6174	$5358 (42%)
Clock 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: ssj_ops/watt
  - $\frac{\text{server side java op /s}}{\text{watt}}$

    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:

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

- 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