

Tema 1

INTRODUCCIÓN

Índice

1. Concepto de Arquitectura de Computadores
2. Clases de Computadores
3. Tecnología, coste y rendimiento

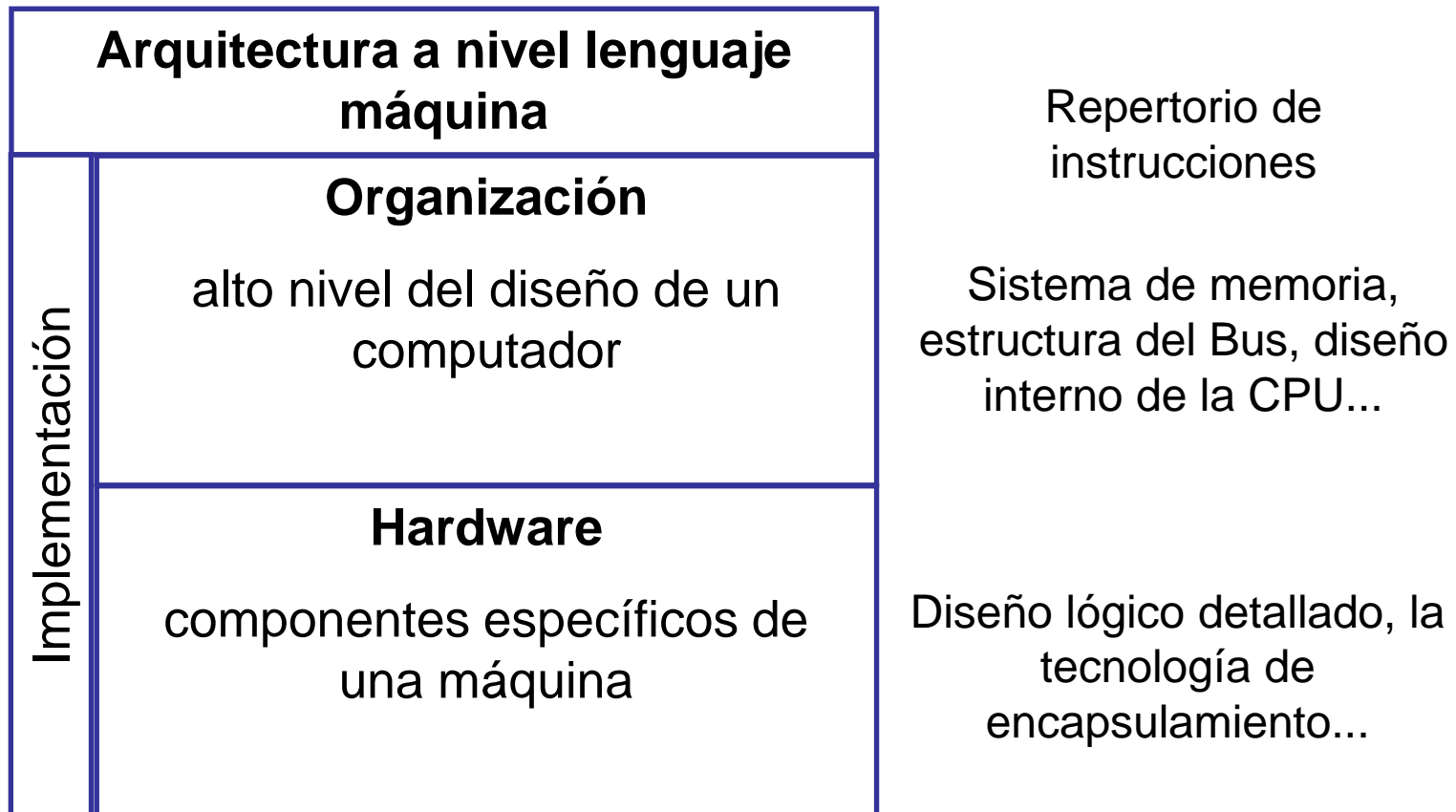
1. Concepto de Arquitectura de Computadores

- Disciplina que trata el diseño de máquinas para ejecutar programas con criterios de optimización de rendimiento y coste
- Aspectos relacionados:
 - Diseño del repertorio de instrucciones
 - Diseño de la organización funcional
 - Diseño lógico
 - Implementación

Definición moderna

- Durante el período clásico de los computadores (hasta los años 70), cada aspecto de un computador era tratado aisladamente por un especialista distinto.
- Actualmente, la arquitectura de computadores es una disciplina vertical, que comprende todos los niveles, que diseñan máquinas programables para que ejecuten correcta y eficazmente un conjunto (previsible o no) de programas.
- Los niveles son interdependientes: las decisiones tomadas en cada uno pueden influir sobre el resto.

Ámbito de la Arquitectura de Computadores



2. Clases de Computadores

- El mercado de los computadores evoluciona según la tecnología base y los hábitos de consumo de la sociedad
- Hace viables aplicaciones novedosas
 - Computadores en coches
 - Móviles
 - Proyecto Genoma Humano
 - World Wide Web
 - Los motores de búsqueda
- Los computadores son omnipresentes

Clases de Computadores

- Dispositivos móviles personales (PMD)
 - Dependientes de batería
 - Conectados a internet
 - Incluyen teléfonos móviles, tabletas, gafas electrónicas...
 - Importancia del coste y eficiencia energética



- Computador personal
 - De propósito general, gran variedad de software (configuraciones muy diversas)
 - Optimiza precio-prestaciones
 - Potencia equilibrada de cálculo y gráficos
 - Incluye portátiles, netbooks o sobremesa



Clases de Computadores

■ Servidores

- Basados en red
- Alta capacidad, rendimiento, fiabilidad
- La disponibilidad es crítica
- Diseño escalable
- Varía desde servidores pequeños hasta servidores del tamaño de un edificio



■ Supercomputadores

- Tipo de servidor
- Cálculos científicos y de ingeniería de gama alta
- Máxima capacidad, pero representan una pequeña fracción del mercado global de ordenadores



Clases de Computadores

- Computadores embebidos
 - Ocultos como componentes de sistemas
 - Restricciones estrictas de potencia/rendimiento/coste
 - Amplio espectro coste rendimiento
 - Presentes en: microondas, lavadoras, impresoras, coches ...



- Computación en nube
 - Cluster/ Warehouse Scale Computer (WSC)
 - Software as a service (SaaS): búsquedas, redes sociales, video compartido, juegos multiusuario
 - LANs de sobremesas y servidores actuando como un gran computador.
 - Parte del software se ejecuta en un PMD y otra parte en la nube
 - Amazon, google

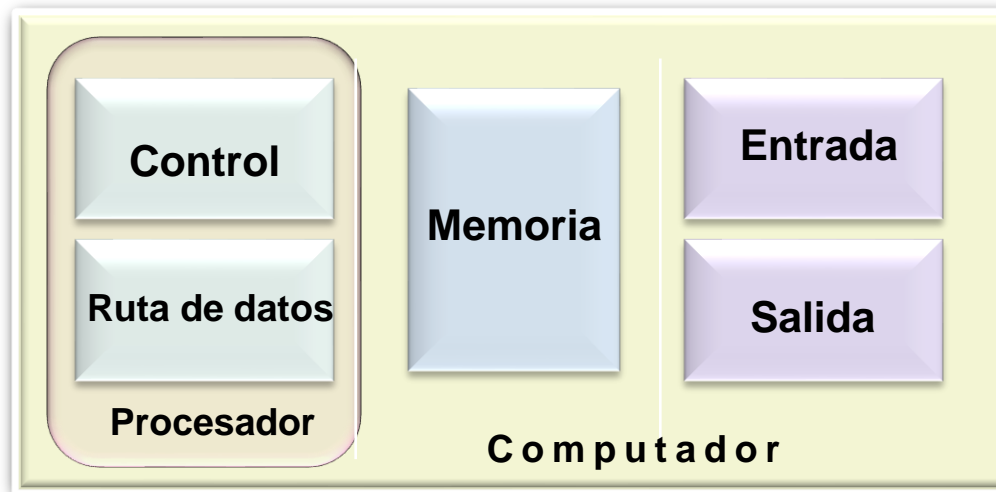


3. Tecnología, coste y rendimiento

- ¿Qué influye en el rendimiento de un programa?
 - Algoritmo
 - Determina el número de operaciones ejecutadas
 - Lenguaje de programación, compilador y arquitectura
 - Determina el número de instrucciones máquina ejecutadas por operación
 - Procesador y sistema de memoria
 - Determina la velocidad de ejecución de las instrucciones
 - Sistema de Entrada/Salida (Hardware y sistema operativo)
 - Determina la velocidad de ejecución de las operaciones de E/S

Componentes de un computador

- Los mismos componentes para todo tipo de ordenadores
 - Sobremesa, servidor, integrado
- Entrada/salida incluye
 - Dispositivos de interfaz de usuario
 - Pantalla, teclado, ratón
 - Dispositivos de almacenamiento
 - Disco duro, CD/DVD, flash
 - Adaptadores de red
 - Para comunicarse con otros ordenadores

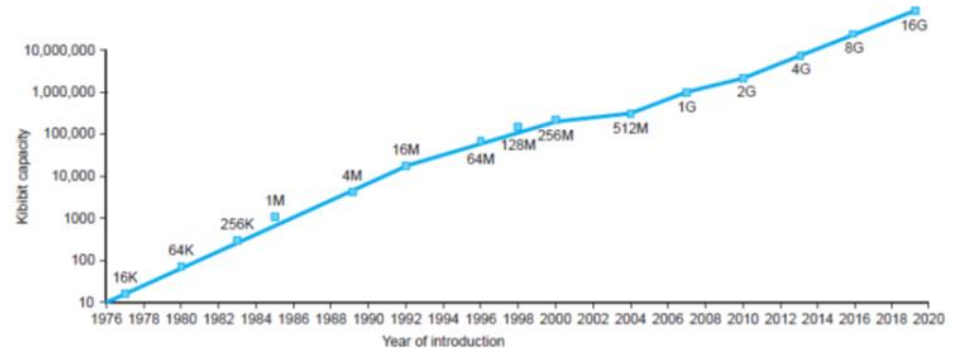


A tener en cuenta por el diseñador

- El trabajo de un diseñador ha de tener presente:
 - La tecnología disponible
 - Limitaciones en relación a potencia y energía
 - Coste de fabricación

Tendencias en la tecnología

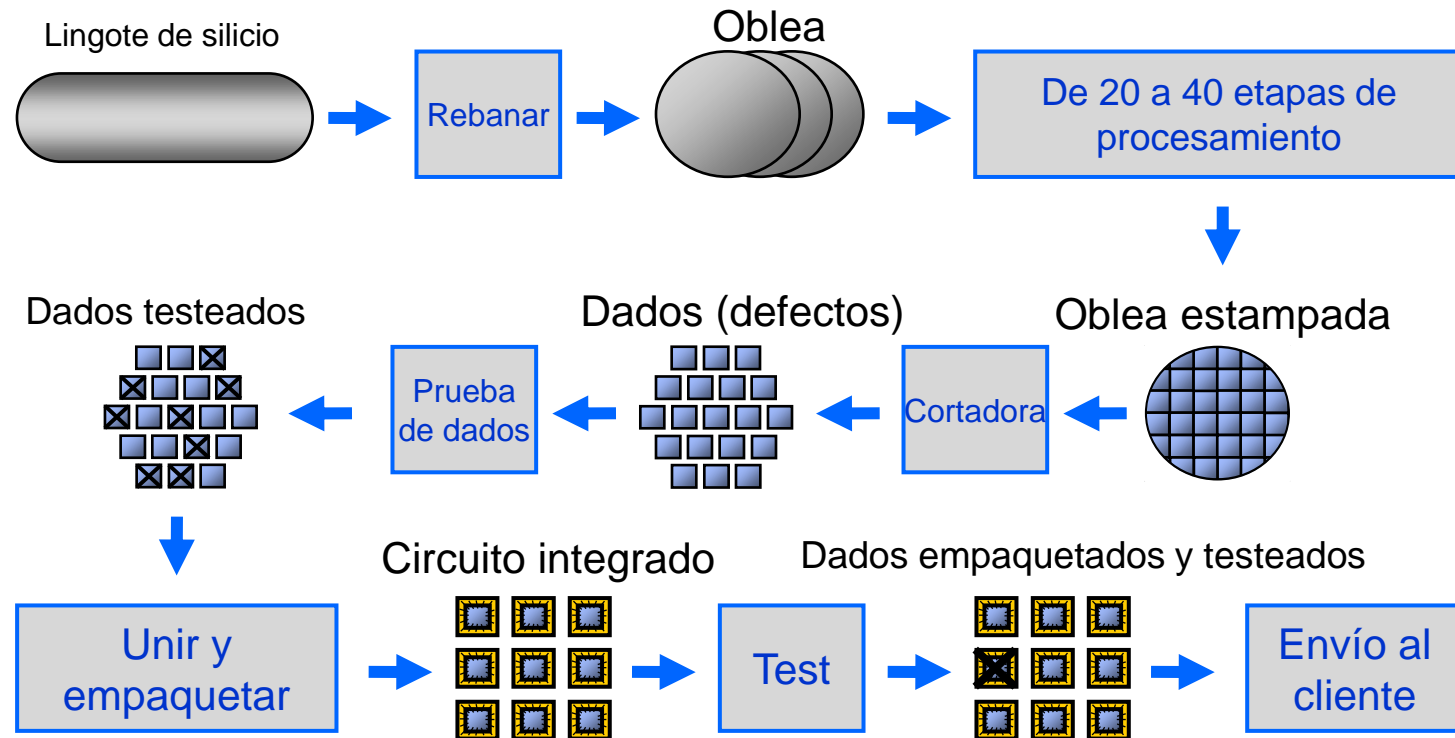
- La tecnología electrónica sigue evolucionando:
 - Se incrementa la capacidad y rendimiento
 - Se reduce el coste



Capacidad DRAM

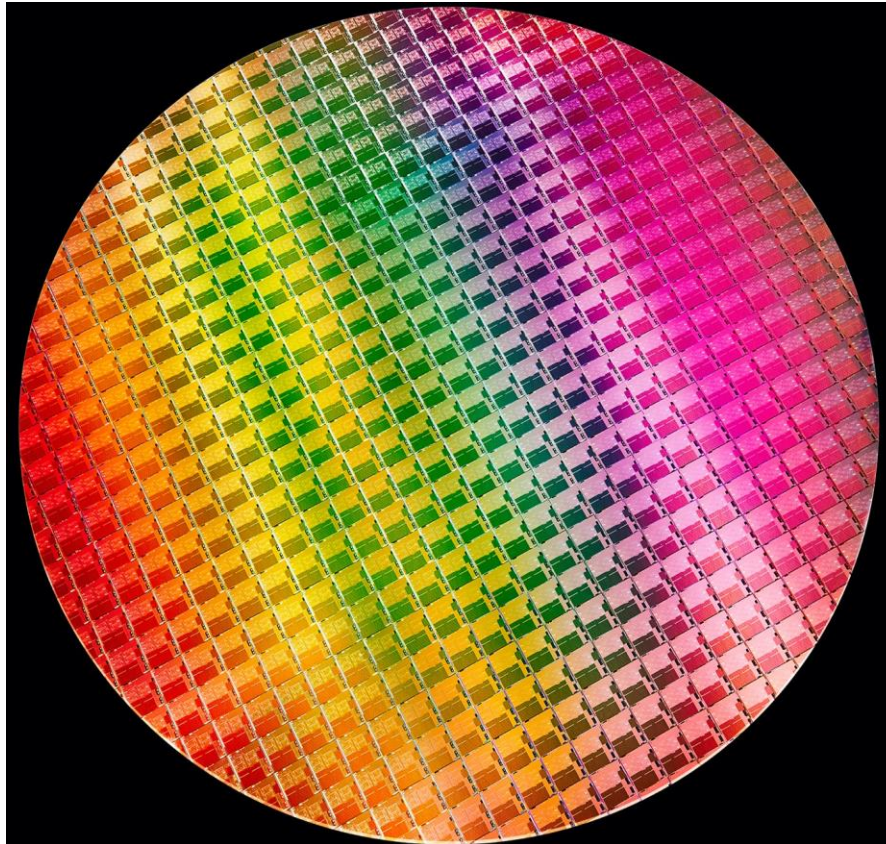
Año	Tecnología	Rendimiento/Coste relativo
1951	Tubo de Vacío	1
1965	Transistor	35
1975	Circuito integrado (CI)	900
1995	CI de Escala muy grande (VLSI)	2,400,000
2013	CI de Escala ultragrande	250,000,000,000

Fabricación de los Circuitos Integrados



$$\text{Coste final} \approx f(\text{superficie dado}^4)$$
$$\text{Superficie dado} = f(\text{complejidad del diseño})$$

Intel® Core 10ª generación



- Oblea de 300 mm, 506 chips, tecnología de 10 nm
- Cada chip mide 11,4 x 10,7 mm

Tiempo de respuesta y productividad

- Tiempo de respuesta:
 - Tiempo que se tarda en realizar una tarea
- Productividad
 - Trabajo total realizado por unidad de tiempo
 - por ejemplo, tareas/transacciones/... por hora
- Ejemplo: ¿Cómo se ven afectados?
 - Al reemplazar el procesador con una versión más rápida
 - Al añadir más procesadores
- Nos centraremos en el tiempo de respuesta por ahora...

Definición Rendimiento

$$\text{Rendimiento} = \frac{1}{\text{Tiempo de ejecución}}$$

- “X es n veces más rápido que Y”

$$\frac{\text{Rendimiento}_X}{\text{Rendimiento}_Y} = \frac{\text{Tiempo ejecución}_Y}{\text{Tiempo ejecución}_X} = n$$

Ejemplo

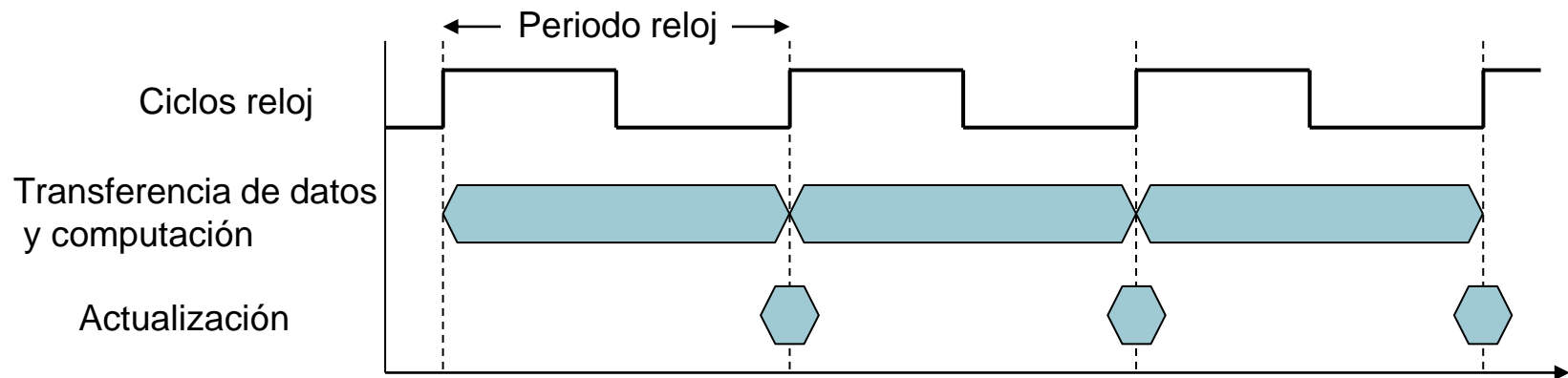
- El tiempo para ejecutar un programa en una máquina A es 10s y en una máquina B es 15s

$$\frac{\textit{Tiempo ejecución}_B}{\textit{Tiempo ejecución}_A} = \frac{15s}{10s} = 1.5$$

- Por tanto, A es 1.5 veces más rápida que B

Ciclo de reloj

- El hardware digital funciona gobernado por un reloj de velocidad constante



- Periodo de reloj: duración del ciclo de reloj
 - Ej: $250\text{ps}=0,25\text{ns}=250\times 10^{-12}\text{s}$
- Frecuencia de reloj: ciclos por segundo
 - Ej: $4.0\text{GHz}=4000\text{MHz}=4.0\times 10^9\text{Hz}$

Tiempo de CPU

$$Tiempo\ CPU = Ciclos\ reloj\ CPU \times Tiempo\ ciclo$$

$$= \frac{Ciclos\ reloj\ CPU}{Frecuencia\ reloj}$$

- Mejorar el rendimiento
 - Reduciendo el número de ciclos
 - Incrementando la frecuencia
 - El diseñador hardware a menudo debe equilibrar el número de ciclos de reloj que requiere un programa y la duración de cada ciclo.

Ejemplo

- Un programa se ejecuta en 10s en un computador A con reloj a 2GHz
- Se quiere diseñar un computador B en el que se ejecute en 6s. Al incrementar la frecuencia se afecta al resto del diseño de la CPU, causando que requiera 1.2 veces los ciclos de reloj que necesita el computador A para dicho programa.
- ¿Qué frecuencia debería tener el computador B?

- $$Frecuencia\ reloj_B = \frac{Ciclos\ reloj_B}{Tiempo\ CPU_B} = \frac{1.2 \times Ciclos\ reloj_A}{6s}$$

- $$Ciclos\ reloj_A = Tiempo\ CPU_A \times Frecuencia\ reloj_A = 10s \times 2GHz = 20 \times 10^9$$

- $$Frecuencia\ reloj_B = \frac{1.2 \times 20 \times 10^9}{6s} = \frac{24 \times 10^9}{6s} = 4GHz$$

- B debe tener el doble de frecuencia de reloj que A para ejecutar el programa en 6s

Recuento de instrucciones y CPI

Ciclos de reloj = Recuento de instrucciones \times Ciclos por instrucción

Tiempo CPU = Recuento de instrucciones \times CPI \times Tiempo ciclo

$$= \mathbf{RI} \times \mathbf{CPI} \times \mathbf{CLK}$$

$$= \frac{\text{Recuento instrucciones} \times \text{CPI}}{\text{Frecuencia reloj}}$$

- Recuento de instrucciones para un programa (RI)
 - Determinado por programa, Repertorio de instrucciones y compilador
- Número medio de ciclos por instrucción (CPI)
 - Determinado por el hardware de la CPU y el repertorio de instrucciones
 - Si diferentes instrucciones tienen diferente CPI
 - CPI medio se verá afectado por la mezcla de instrucciones

Ejemplo

- Dos implementaciones para el mismo repertorio de instrucciones:
- Computador A: Tiempo de ciclo=250ps, CPI=2.0
- Computador B: Tiempo de ciclo=500ps, CPI= 1.2
- ¿Qué computador es más rápido y en cuánto?

$$Tiempo\ CPU_A = RI \times CPI_A \times CLK_A = RI \times 2.0 \times 250ps = RI \times 500ps$$

$$Tiempo\ CPU_B = RI \times CPI_B \times CLK_B = RI \times 1.2 \times 500ps = RI \times 600ps$$

- El computador A es más rápido que el B

$$\frac{Tiempo\ CPU_B}{Tiempo\ CPU_A} = \frac{RI \times 600ps}{RI \times 500ps} = 1.2$$

- El computador A es 1.2 veces más rápido que B

Profundizando en el CPI

- Si distintas clases de instrucciones tardan distinto número de ciclos

$$\text{Ciclos de reloj} = \sum_{i=1}^n (CPI_i \times \text{Recuento de instrucciones}_i)$$

- Media ponderada del CPI

$$CPI = \frac{\text{Ciclos de reloj}}{\text{Recuento de instrucciones}} = \sum_{i=1}^n CPI_i \times \frac{\text{Recuento instrucciones}_i}{\text{Recuento instrucciones}}$$

Frecuencia relativa

Ejemplo

- Supongamos que un diseñador de compiladores está considerando distintas alternativas de código compilado utilizando instrucciones de las clases A, B, C

Clase	A	B	C
CPI por clase	1	2	3
RI en secuencia 1	2	1	2
RI en secuencia 2	4	1	1

- Secuencia 1: RI = 5

- Ciclos de reloj
 $= 2 \times 1 + 1 \times 2 + 2 \times 3$
 $= 10$
- CPI medio $= 10/5 = 2.0$

- Secuencia 2: RI = 6

- Ciclos de reloj
 $= 4 \times 1 + 1 \times 2 + 1 \times 3$
 $= 9$
- CPI medio $= 9/6 = 1.5$

Otras medidas de rendimiento

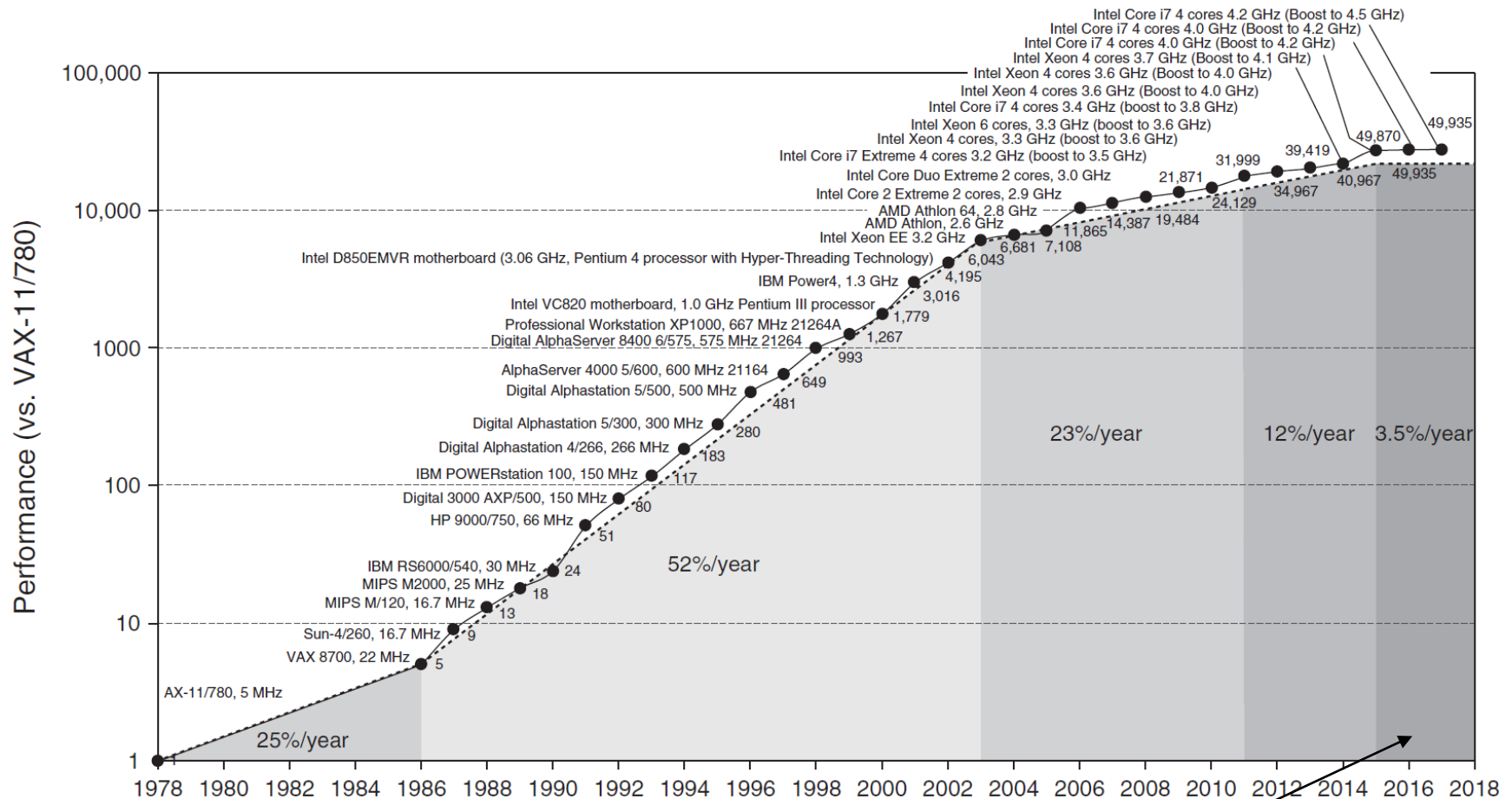
■ **MIPS:** Millones de instrucciones por segundo

- No es adecuado para:
 - Computadores con repertorios de instrucciones distintas
 - Diferencias de complejidad entre instrucciones
- $$MIPS = \frac{\text{Recuento de Instrucciones}}{\text{Tiempo de ejecución} \times 10^6} = \frac{\text{Recuento de instrucciones}}{\frac{\text{Recuento de instrucciones} \times CPI}{f} \times 10^6} = \frac{f}{CPI \times 10^6}$$
- Depende del programa que se considere

■ **MFLOPS :** Millones de operaciones en coma flotante por segundo.

- Contabiliza operaciones en lugar de instrucciones. Un mismo programa ejecutándose en distintas arquitecturas tendrá distinto número de instrucciones, pero igual número de operaciones en CF
- $$MFLOPS = \frac{\text{Número operaciones en coma flotante en programa}}{\text{Tiempo ejecución} \times 10^6}$$
- No aplicable a programas sin usos de coma flotante (compiladores, procesadores de texto)
- Depende del repertorio de instrucciones en CF de la máquina, no siempre el mismo
- Programas diferentes ejecutan distintas operaciones en CF y no todas con el mismo coste.

Rendimiento del monoprocesador



Limitado por la potencia, el paralelismo a nivel de instrucción y la latencia de la memoria.

Multiprocesadores

- Microprocesadores multinúcleo (*Multicore*)
 - Más de un procesador por chip
- Requiere programación explícitamente paralela
 - Comparación con el paralelismo a nivel de instrucción
 - El hardware ejecuta varias instrucciones a la vez
 - Oculto al programador
- Difícil de realizar
 - Programación en función del rendimiento
 - Equilibrio de la carga
 - Optimización de la comunicación y la sincronización