



UNIVERSIDAD DE ALCALÁ
D.T.O. AUTOMÁTICA

Arquitectura II
I.T. Informática de Sistemas

1

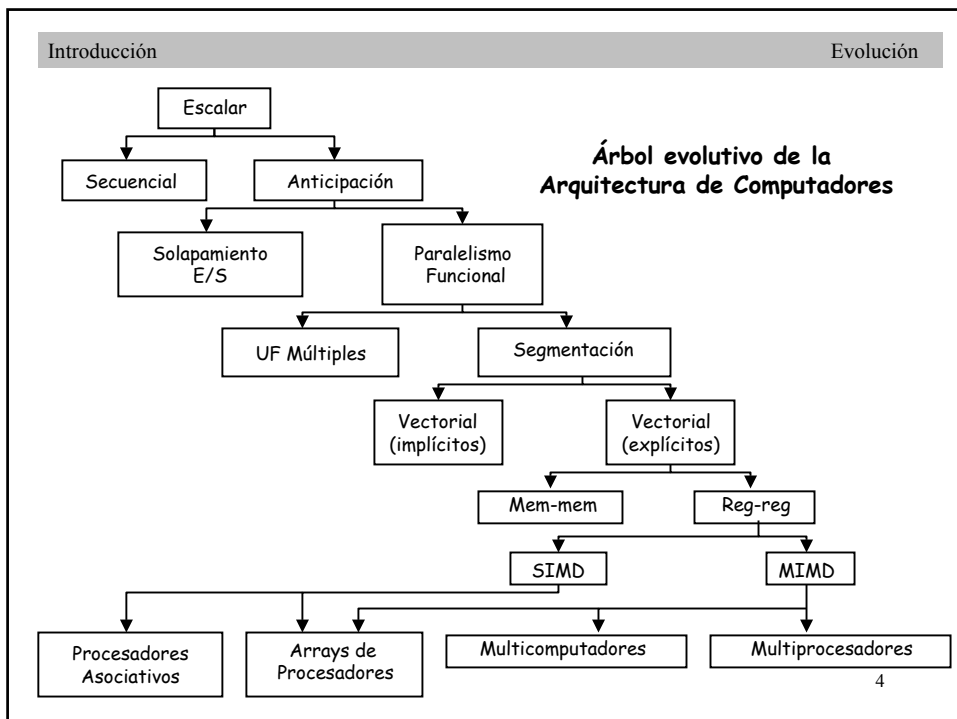
Tema 1: Introducción

- **Objetivos**
- **Introducción**
- **Rendimiento de Computadores**
- **Paralelismo**
- **Taxonomía de Flynn**
- **Tipos de paralelismo**
- **Ley de Amdahl**
- **Descripción de la asignatura**

2

Objetivos:

- Comprender la necesidad de mejorar las arquitecturas existentes
- Conocer los lugares donde se encuentran los distintos tipos de paralelismo, y sus características esenciales
- Conocer los distintos tipos de máquinas que se pueden diseñar para explotarlo



Taxonomía de Flynn:

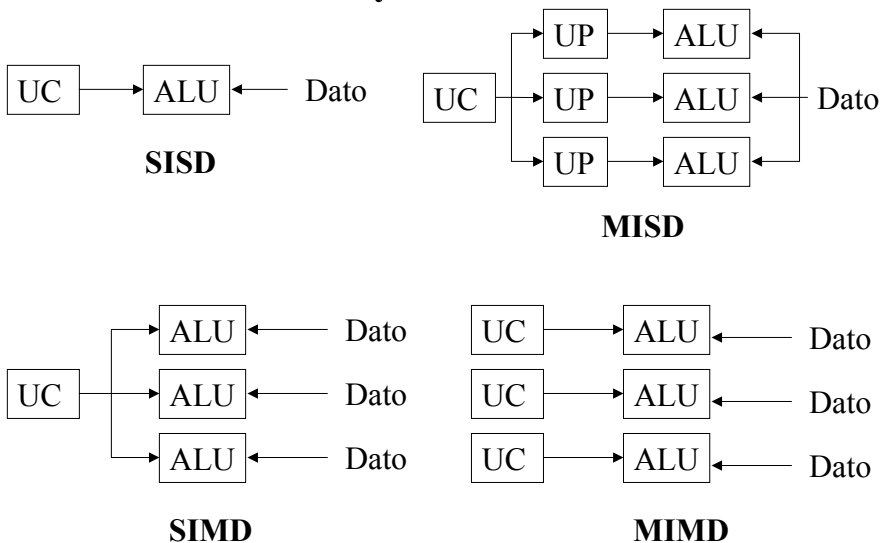
SISD: un flujo de instrucciones único trabaja sobre flujo de datos único (arquitectura clásica, superescalares)

SIMD: un flujo de instrucciones único trabaja sobre un flujo de datos múltiple (computadores matriciales)

MISD: un flujo de instrucciones múltiple trabaja sobre un flujo de datos único (clase no implementada, resultado de la clasificación)

MIMD: un flujo de instrucciones múltiple trabaja sobre un flujo de datos múltiple (multiprocesadores)

Taxonomía de Flynn



Factores que impulsan la renovación en arquitectura

Necesidad de potencia de cálculo: crecimiento exponencial

- Procesos complejos en tiempo real (control de centrales, de viajes espaciales, control de tráfico, etc)
- Simulación (de moléculas, de poblaciones, predicción meteorológica, modelos mecánicos, etc)
- Problemas hasta ahora no atacables (salvo por procesos heurísticos) pero resolubles

7

Factores que impulsan la renovación en arquitectura (cont.)

Limitación de las posibilidades de la arquitectura clásica

- Presencia de múltiples cuellos de botella:
 - memoria
 - unidades funcionales
- Limitaciones físicas:
 - lento avance en las tecnologías de memoria
 - dificultad para aumentar la densidad de integración
 - crecimiento incontrolado de la disipación de calor al aumentar la frecuencia
 - límites en la frecuencia: suponiendo velocidades de transición en el silicio de $3 \cdot 10^9 \text{ cm/s}$ y distancias de 1 cm:
 $f_{\text{máx}} = 1 / (1 \text{ cm} / 3 \cdot 10^9 \text{ cm/s}) = 3 \text{ GHz}$

8

Recurso: explotar el paralelismo

Seguir investigando en mejoras tecnológicas

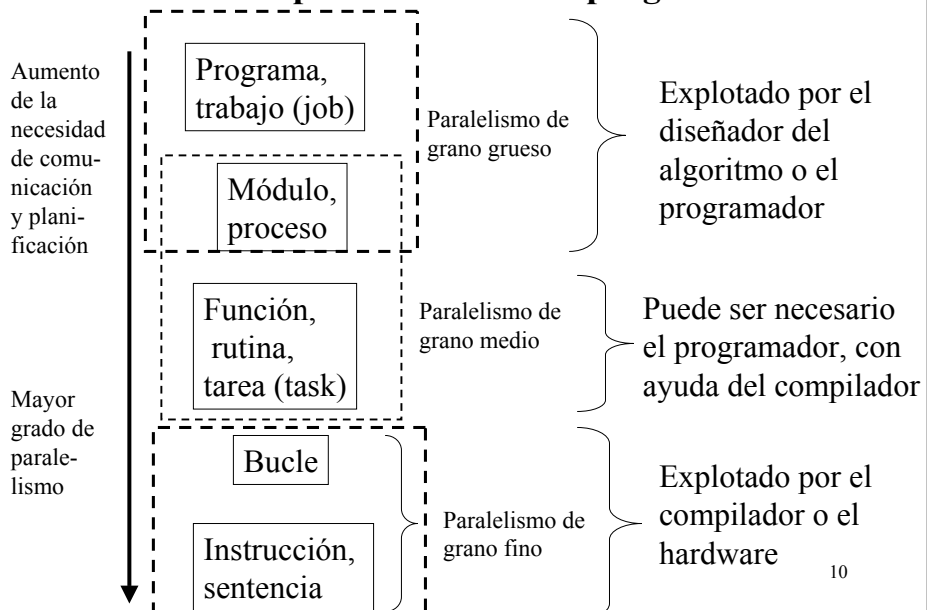
- mayores densidades de integración, técnicas de disipación de calor
- transistores moleculares
- computación cuántica

PERO: simultáneamente, buscar formas más eficientes de utilizar los recursos de un computador

Si las cosas no se pueden hacer más deprisa, hacer más cosas a la vez => Paralelismo: conjunto de tareas independientes entre sí (y que son susceptibles de ser llevadas a cabo de forma simultánea)

9

Niveles de paralelismo en un programa



10

Tipos de paralelismo:

- **En monoprocesadores:**
 - **Segmentación**
 - **División funcional.**
- **En multiprocesadores**
 - **SIMD**
 - **MIMD**
 - **Acoplo fuerte**
 - **Acoplo moderado**
 - **Acoplo débil**

En cuanto a programación:

- **SPMD** (Simple programa múltiple dato). El mismo programa es cargado en múltiples procesadores y se ejecuta sobre conjuntos de datos distintos.
- **MPMD** (Múltiple programa múltiple dato). Cada procesador ejecuta programas distintos. Esta estructura suele ser del tipo maestro esclavo en la que un procesador coordina el trabajo del resto.

El aprovechamiento de un computador paralelo depende, en gran medida, del entorno de programación en dos facetas:

- **Las herramientas de programación**
- **El sistema operativo**

Enfoques del paralelismo en computadores paralelos:

Paralelismo implícito: Se programa en lenguaje secuencial y el compilador se encarga de paralelizar y asignar recursos.

- Pequeño aprovechamiento (depende de la inteligencia del Compilador)
- El trabajo del programador es fácil
- Aprovecha todo el código secuencial existente

Paralelismo explícito: se usan dialectos paralelos de programación.

- Mejor aprovechamiento de las posibilidades paralelas de la máquina
- Más trabajo para el programador

Rendimiento de computadores

$$T = I_c \times CPI \times \tau$$

I_c (N° de instrucciones del programa)
 CPI (Ciclos por instrucción)
 τ (Ciclo de reloj)

Desarrollando el término de ciclos por instrucción

$$T = I_c \times (p + m \times k) \times \tau$$

p número de ciclos del procesador para la ejecución de la instrucción (decodificación, ejecución etc.)
 m número de referencias a memoria.
 K relación en ciclos que existe entre las operaciones del procesador y las operaciones de acceso a memoria.

Rendimiento de computadores

	C P I				
	I_c Instrucciones por programa	p ciclos de procesador	m Referencias a memoria	k Latencia de la memoria	τ ciclo de reloj
Cjto. de instrucciones	X	X			
Tecnología del compilador	X	X	X		
Implementación del computador y control		X			X
Cache y jerarquía de memoria				X	X

15

Rendimiento de computadores

RENDIMIENTO (MIPS)

- **Rendimiento teórico.-** Es el máximo rendimiento que se puede alcanzar (rendimiento de pico).
- **Rendimiento real.-** Es el que se obtiene en un programa determinado.
- **Rendimiento sostenido.-** Es el más indicativo, representa la media de rendimiento para diversas tareas

$$MIPS = \frac{I_c}{T \times 10^6}$$

$$MIPS = \frac{f}{CPI \times 10^6} = \frac{f \times I_c}{C \times 10^6}$$

$$CPI = \frac{C}{I_c}$$

16

Productividad (Throughput)

Indica el número de programas que el sistema puede ejecutar por unidad de tiempo:

$$W_p = \frac{MIPS \times 10^6}{I_c} = \frac{f}{I_c \times CPI}$$

Ejemplo:

	f	Rendimiento	Tiempo de CPU
VAX 11/780	5 MHz.	1MIPS	12x
IBM RS/6000	25 Mhz	18 MIPS	x

VAX 11/780

CPI = 5/1 = 5

IBM RS/6000

CPI = 25/18 = 1.3

Relación entre la longitud de los programa: el programa ejecutado en el IBM es 1.5 veces mayor que el ejecutado en el VAX.

$$I_c = MIPS \times T \times 10^6$$

$$r = \frac{18 \times 1 \times 10^6}{1 \times 12 \times 10^6} = 1.5$$

17

Rendimiento de computadores

Nombre	Valor	Rango
MFLOPS	megaFLOPS = 10^6	Rango de las workstations actuales
GFLOPS	gigaFLOPS = 10^9	Rango de los actuales supercomputadores
TFLOPS	teraFLOPS = 10^{12}	Rango de los supercomputadores que están apareciendo en la actualidad
PFLOPS	petaFLOPS = 10^{15}	El sueño, los ingenieros se preguntan que problemas se podrán resolver con esta potencia
EFLOPS	exaFLOPS = 10^{18}	Probablemente necesite una tecnología nueva

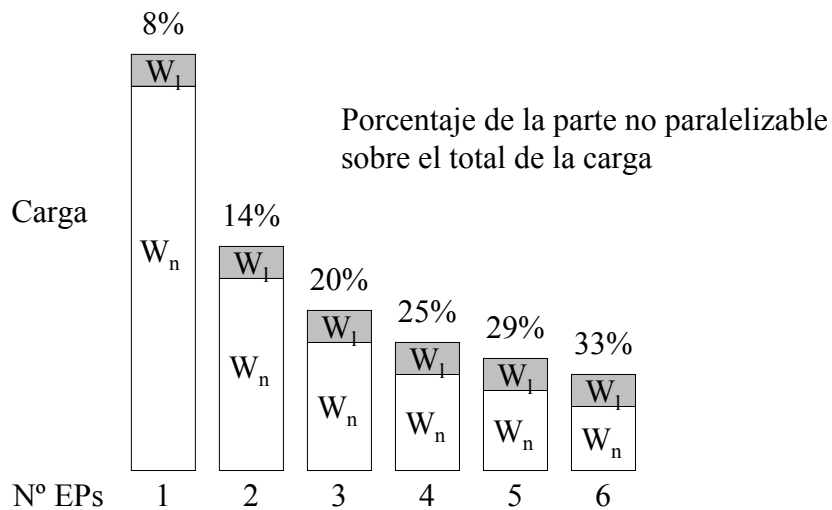
18

Ley de Amdahl

$$S_n = \frac{W_l + W_n}{W_l + W_n/n}$$

- S_n : mejora del rendimiento (speedup)
 W_l : parte no paralelizable
 W_n : parte paralelizable
 n : número de procesadores

Ley de Amdahl, versión gráfica



OJO! A la hora de introducir modificaciones en la arquitectura, no hay que olvidar los factores que influyen en el rendimiento:

- Recuento de instrucciones del programa
- Ciclos por Instrucción (CPI)
- Accesos a memoria
- Ciclo de reloj

Y tener en cuenta la relación:

$$\text{Tiempo ejecución} = \text{N}^\circ \text{ Inst.} * \text{CPI} * \text{Ciclo}$$

21

Descripción de la asignatura

La asignatura trata varios temas importantes en el diseño actual de computadores (pero no todos):

- Las técnicas de aprovechamiento del paralelismo entre y dentro de las instrucciones (de utilidad principalmente en el diseño de microprocesadores): segmentación
- Las técnicas de aprovechamiento del paralelismo en los datos: máquinas SIMD
- Las técnicas de aprovechamiento del paralelismo de grano grueso en las aplicaciones: multiprocesadores

22

Contenidos:

- Segmentación:** descripción del concepto de segmentación, y su aplicación genérica (centrándose en cómo se aplica, en lugar de a qué)
- Segmentación de instrucciones:** aplicación de las técnicas vistas (y de bastantes más que surgen en el caso de las instrucciones) a la ejecución de instrucciones
- Segmentación II:** continuación del tema anterior, centrándose en la ampliación de las posibilidades de la segmentación
- Procesadores vectoriales y máquinas SIMD:** estudio de máquinas pensadas especialmente para proporcionar gran potencia en cálculo numérico
- Multiprocesadores:** diseño de máquinas que constan de varios procesadores que cooperan para ejecutar las aplicaciones, y de los problemas que se plantean en ellas

23

Bibliografía:

- Hennessy-Patterson: Arquitectura de computadores: un enfoque cuantitativo. 2ª edición (OJO!). (“El Hennessy-Patterson”)
- Kai Hwang: Advanced Computer Architecture. McGraw-Hill, 1993

Para quien el inglés sea un problema:

- Kai Hwang, Fayé Briggs: Arquitectura de computadoras y procesamiento paralelo. McGraw-Hill, 1988

Es la versión antigua (a veces demasiado), pero muchas cosas sirven. De todas formas, mejor el más reciente.

24

Bibliografía (cont.):

En internet hay muchas cosas. Por ejemplo:

- Página de David Patterson, de la U. de Berkeley:
<http://www.cs.berkeley.edu/~pattnsn/>
- Página de Al Davis de la Universidad de Utah:
<http://www.cs.utah.edu/~ald/>
- Página de Arvind, MIT. <http://www.csg.lcs.mit.edu/6.823/>
- Computer architecture home page:
<http://www.cs.wisc.edu/~arch/www>