

Arquitectura e Ingeniería de Computadores

Tema 1 – Introducción a las arquitecturas paralelas

Ingeniería en Informática

Departamento de Tecnología Informática y Computación

AIC – Tema 1 Introducción a las arq. paralelas

Introducción

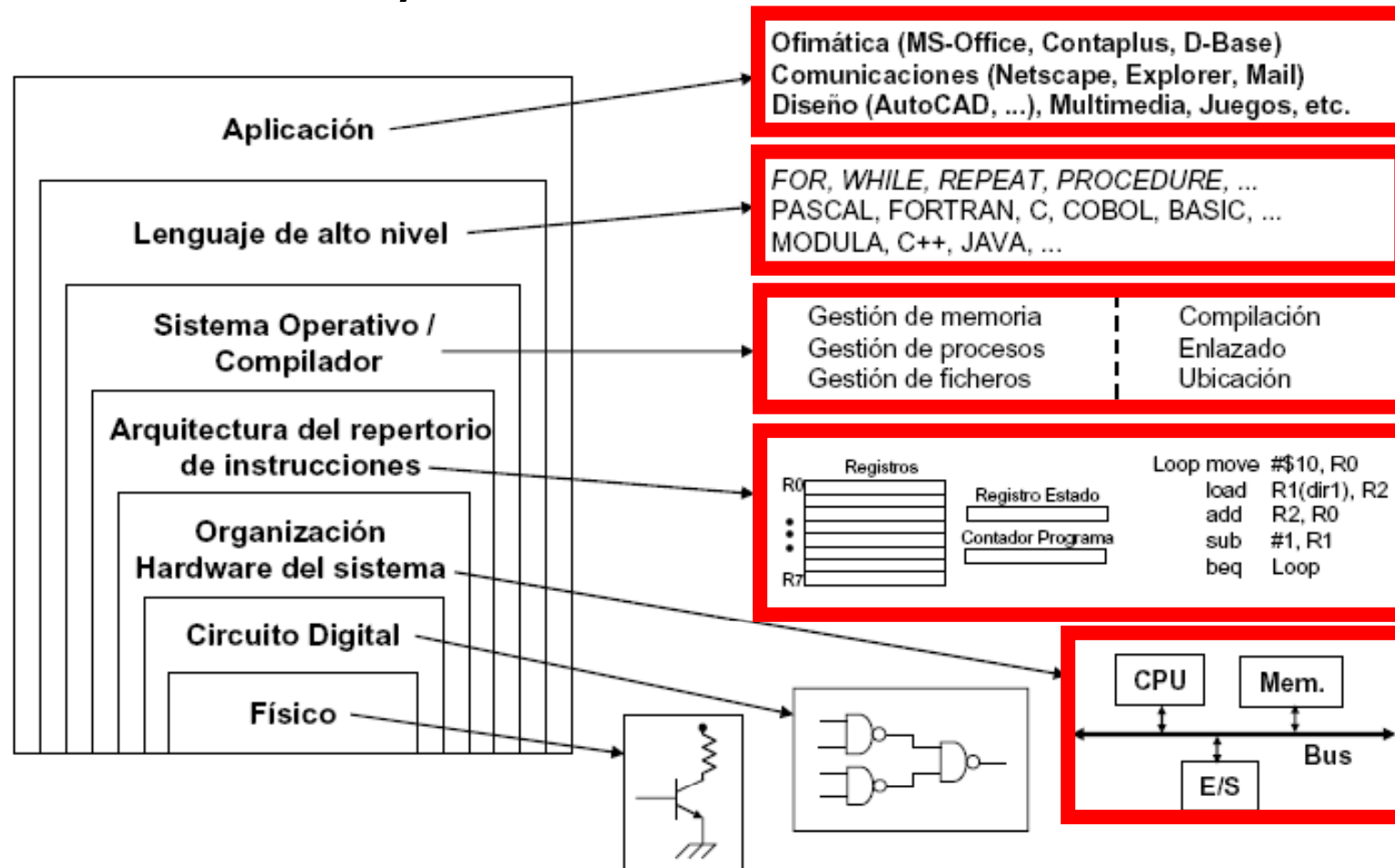
Evolución

Clasificación

Rendimiento

Problemas

◆ Marco de la asignatura: Niveles de descripción de un computador (Planteamiento inicial)



AIC – Tema 1 Introducción a las arq. paralelas

Introducción

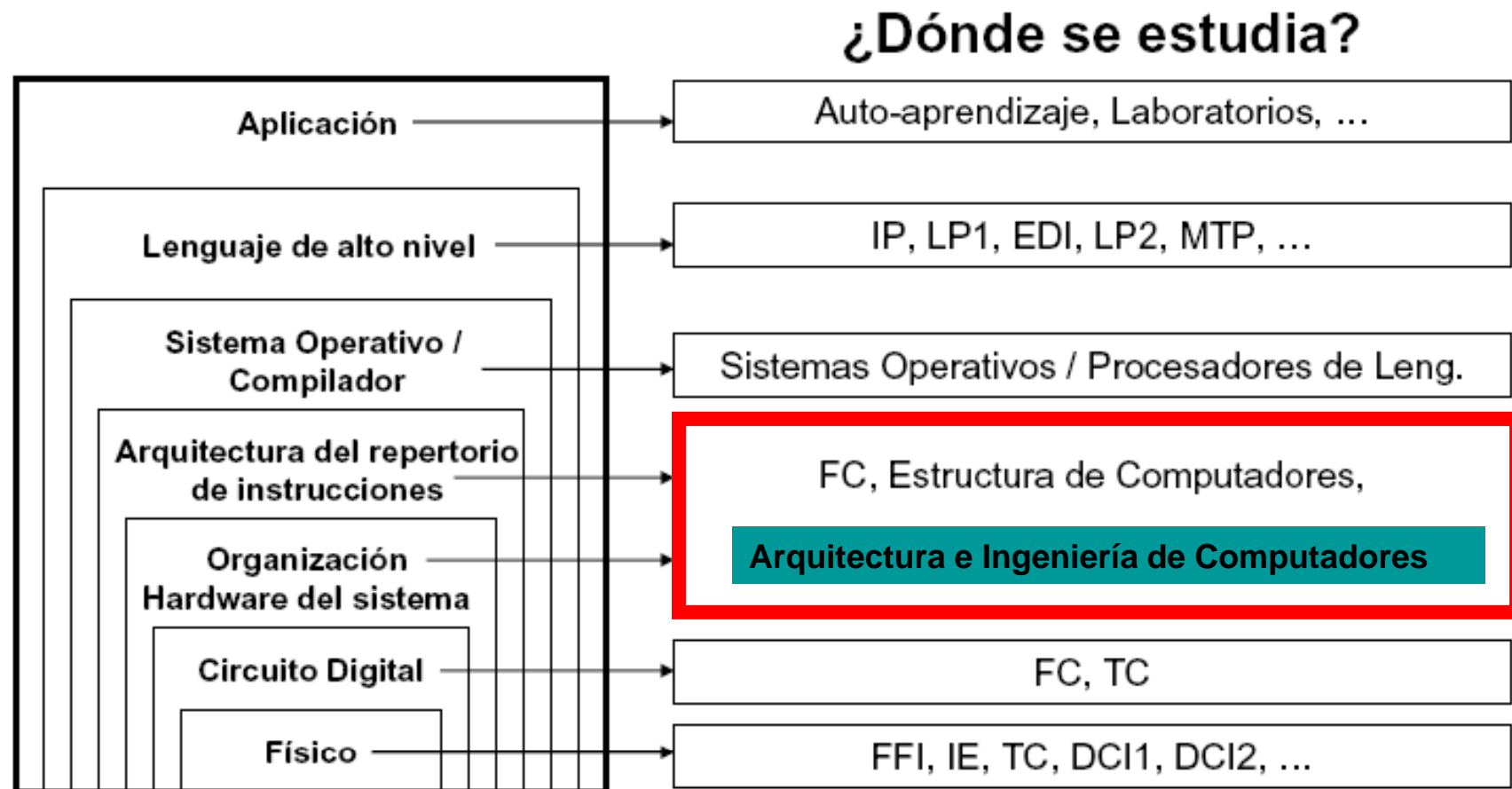
Evolución

Clasificación

Rendimiento

Problemas

◆ Marco de la asignatura:



AIC – Tema 1 Introducción a las arq. paralelas

Introducción

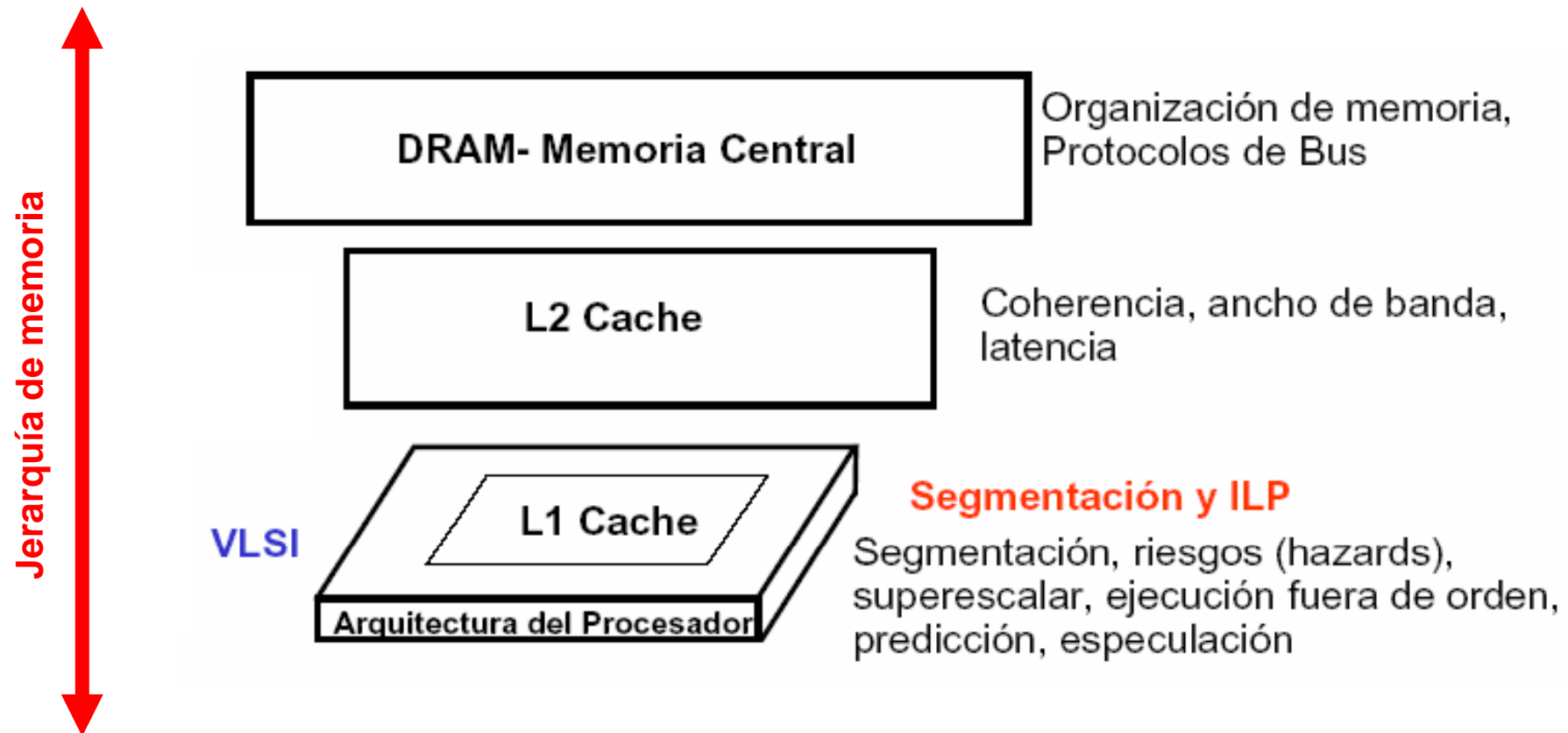
Evolución

Clasificación

Rendimiento

Problemas

◆ ¿Qué estudia la asignatura?



AIC – Tema 1 Introducción a las arqu. paralelas

Introducción

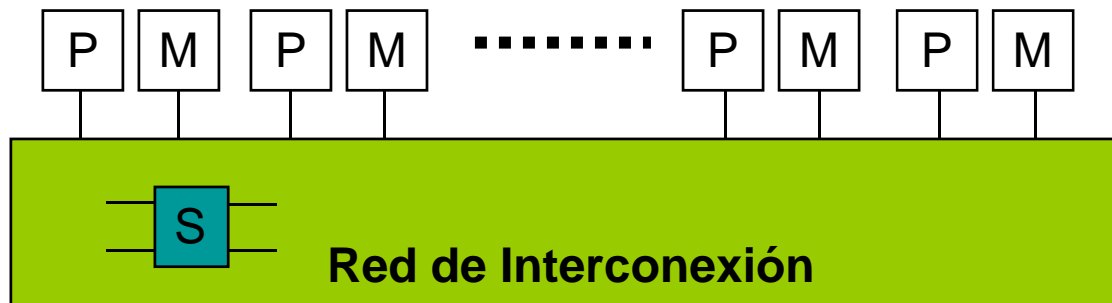
Evolución

Clasificación

Rendimiento

Problemas

◆ ¿Qué estudia la asignatura?



**Multiprocesadores
+
Redes de interconexión**

Comunicación:
Memoria compartida,
paso de mensajes,
paralelismo de datos

Elementos:

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

Red: topología,
encaminamiento,
ancho de banda,
latencia ...

AIC – Tema 1 Introducción a las arq. paralelas

Introducción

Evolución

Clasificación

Rendimiento

Problemas

◆ Definición de **arquitectura**

- ▶ “Conjunto de instrucciones, recursos y características del procesador que son visibles al software que se ejecuta en el mismo. Por tanto, la arquitectura determina el software que el procesador puede ejecutar directamente, y esencialmente define las especificaciones a las que debe ajustarse la microarquitectura” [Ortega, 2005]

◆ Definición de **microarquitectura**

- ▶ “Conjunto de recursos y métodos utilizados para satisfacer las especificaciones que establece la arquitectura. El término incluye tanto la forma en que se organizan los recursos como las técnicas utilizadas para alcanzar los objetivos de costes y prestaciones planteados. La microarquitectura define las especificaciones para la implementación lógica” [Ortega, 2005]

AIC – Tema 1 Introducción a las arq. paralelas

Introducción

Evolución

Clasificación

Rendimiento

Problemas

◆ Niveles de descripción de un computador:

▶ Niveles estructurales de Bell y Newell:

- Descripción del computador mediante una aproximación por capas.
- Cada capa utiliza los servicios que proporciona la del nivel inferior.
- Propone 5 niveles:
 - De componente
 - Electrónico
 - Digital
 - Transferencia entre registros (RT)
 - Procesador-Memoria-Interconexión (PMS)

AIC – Tema 1 Introducción a las arq. paralelas

Introducción

Evolución

Clasificación

Rendimiento

Problemas

► Niveles de Interpretación de Levy:

- Contemplan al computador desde un punto de vista funcional.
- Constituido por una serie de máquinas virtuales superpuestas.
- Cada máquina interpreta las instrucciones de su nivel, proporcionando servicios a la máquina de nivel superior y aprovechando los de la máquina de nivel inferior.
- Se distinguen 5 niveles:
 - Aplicaciones
 - Lenguajes de alto nivel
 - Sistema Operativo
 - Instrucciones máquina
 - Microinstrucciones
- Estos niveles son similares a los **niveles funcionales** de Tanenbaum.

AIC – Tema 1 Introducción a las arq. paralelas

Introducción

Evolución

Clasificación

Rendimiento

Problemas

◆ Niveles de abstracción para un computador:

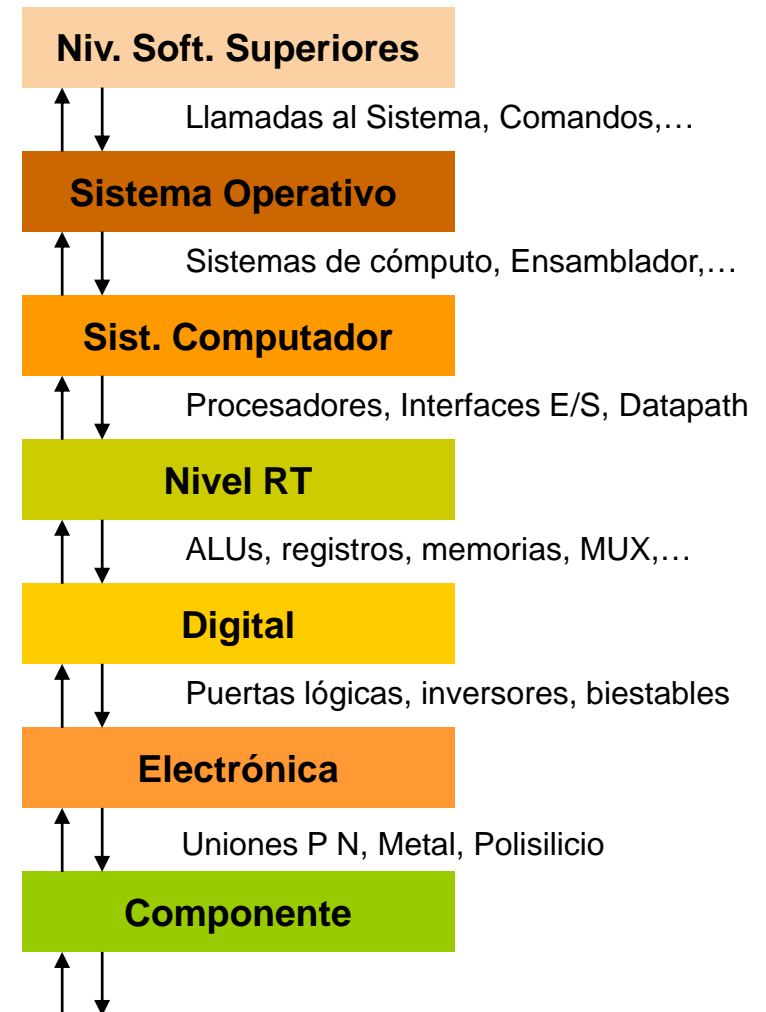
Integra la orientación **estructural** de los niveles de Bell y Newell y el punto de vista **funcional** de los niveles de Levy y Tanenbaum.

SW

ARQUITECTURA

TECNOLOGÍA

HW



AIC – Tema 1 Introducción a las arq. paralelas

Introducción

Evolución

Clasificación

Rendimiento

Problemas

◆ Componentes físicos

- ▶ Semiconductores de tipo **n** y **p**, metales, polisilicio, etc...
- ▶ A partir de estos se construyen bloques: transistores, resistencias, etc.
- ▶ Las leyes que lo rigen son las de la electrónica física.

◆ Circuito electrónico

- ▶ Puertas lógicas, biestables, etc. Utilizan componentes del nivel anterior.
- ▶ Las leyes que lo rigen son las de la electricidad, de naturaleza continua.
- ▶ El comportamiento del circuito se describe en términos de corrientes, tensiones y frecuencias.

◆ Lógica digital

- ▶ Las leyes que lo rigen son las del Álgebra de Boole.
- ▶ Se divide en 2 partes: circuitos combinacionales y secuenciales.
- ▶ **Nivel combinacional:** se utilizan como componentes las puertas NAND, NOR, NOT, etc, para generar bloques como, multiplexores, decodificadores, conversores de códigos y circuitos aritméticos.
- ▶ **Nivel secuencial:** se utilizan como componentes elementos de memoria (biestables) y bloques del nivel anterior para obtener circuitos secuenciales, como registros, contadores, memorias, etc.

AIC – Tema 1 Introducción a las arq. paralelas

Introducción

Evolución

Clasificación

Rendimiento

Problemas

◆ Transferencia entre registros (RT)

- ▶ Estudio del comportamiento de las unidades de un computador en términos de transferencia de información entre registros.
- ▶ Utiliza los componentes del nivel anterior, registros, circuitos aritméticos, memorias, etc, para crear componentes del procesador o de otros elementos del computador (interfaces).
- ▶ En este nivel se incluye como un posible subnivel la **microprogramación**.

AIC – Tema 1 Introducción a las arq. paralelas

Introducción

Evolución

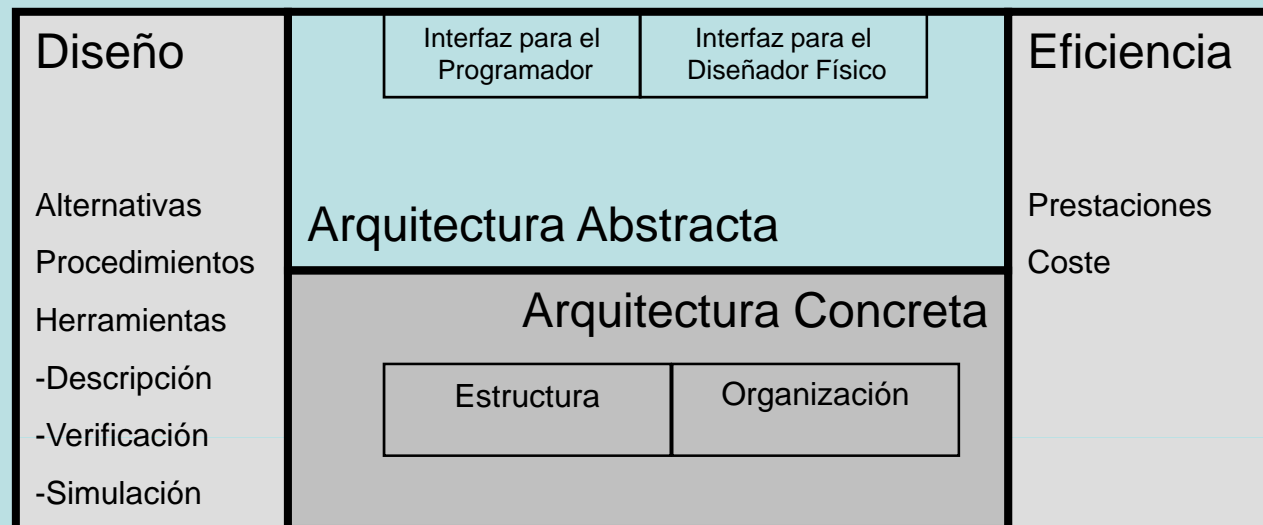
Clasificación

Rendimiento

Problemas

◆ Sistema computador

- ▶ Especificación de componentes (memorias, procesador, buses, redes de interconexión, periféricos, etc).
interconexión entre ellos y operación del sistema completo.
- ▶ Programación a bajo nivel (lenguaje máquina y ensamblador).



AIC – Tema 1 Introducción a las arq. paralelas

Introducción

Evolución

Clasificación

Rendimiento

Problemas

◆ Sistema operativo

- ▶ Interfaz entre hardware y software.
- ▶ Encargado de facilitar el uso eficiente de los recursos hardware por parte de los usuarios y de los programas de aplicación.

◆ Niveles superiores

- ▶ Niveles software: compiladores, programas escritos en lenguajes de alto nivel, etc.
- ▶ Realización de programas y compiladores eficientes requieren el conocimiento de la arquitectura del computador
→ incremento de prestaciones.

AIC – Tema 1 Introducción a las arq. paralelas

Introducción

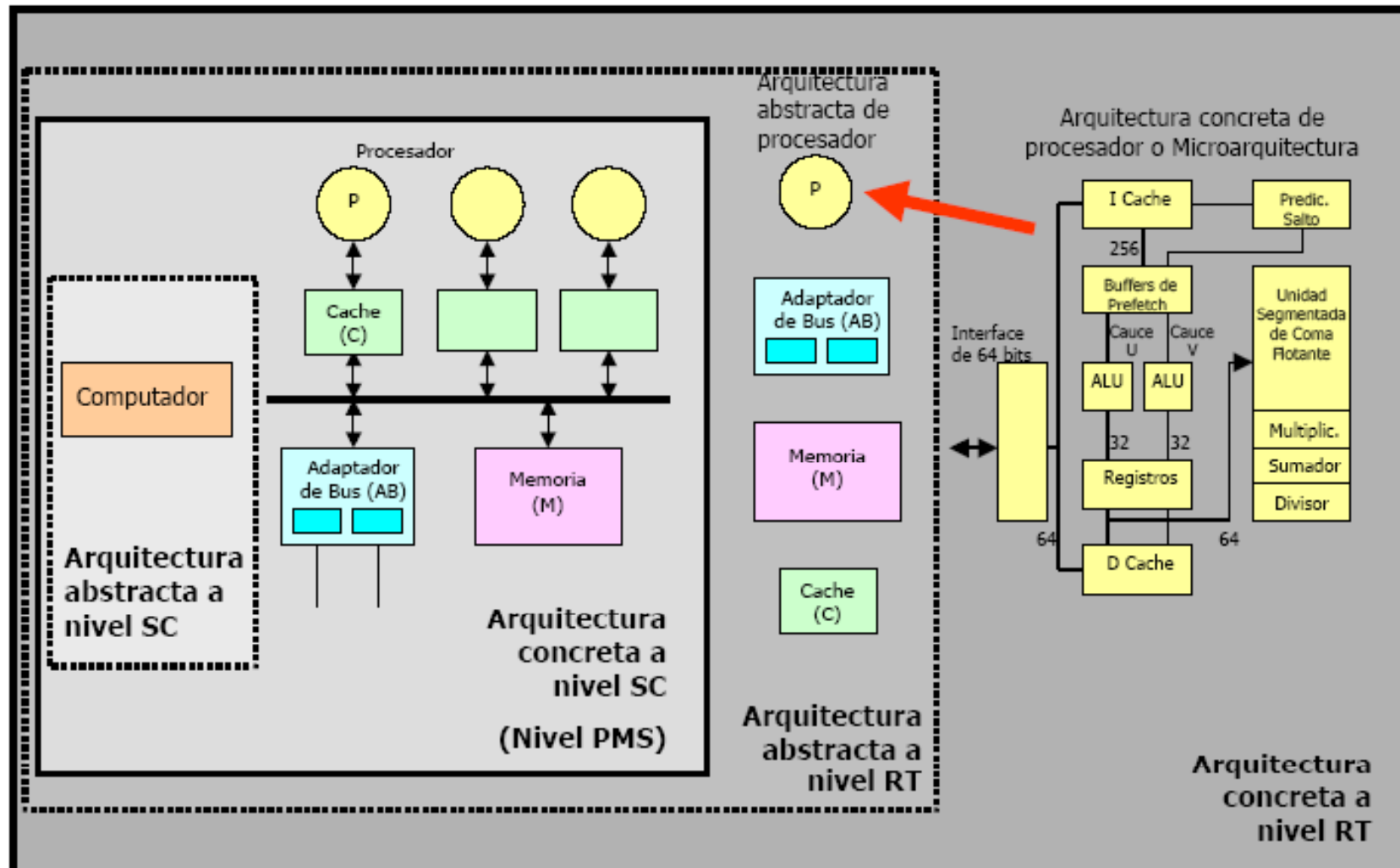
Evolución

Clasificación

Rendimiento

Problemas

◆ Ejemplo de niveles de abstracción:



AIC – Tema 1 Introducción a las arq. paralelas

Introducción

Evolución

Clasificación

Rendimiento

Problemas

◆ Una definición más amplia de arquitectura:

► “La Arquitectura de Computadores abarca las **abstracciones** que proporcionan las interfaces entre hardware/software y usuario/sistema, y la **estructura y organización** que permiten implementar dichas abstracciones, proporcionando prestaciones y costes razonables en la ejecución de los programas de computador”

AIC – Tema 1 Introducción a las arq. paralelas

Introducción

Evolución

Clasificación

Rendimiento

Problemas

◆ El ámbito de la Arquitectura de Computadores incluye:

- ▶ El lenguaje máquina del computador, la microarquitectura del procesador y la interfaz para los programas en lenguaje máquina (lenguaje máquina y arquitectura concreta del procesador).
- ▶ Los elementos del computador y como interactúan (es decir la arquitectura concreta del computador, la estructura y organización).
- ▶ La interfaz que se ofrece a los programas de alto nivel y los módulos que permiten controlar el funcionamiento del computador (sistema operativo y la arquitectura abstracta del computador).
- ▶ Los procedimientos cuantitativos para evaluar los sistemas (benchmarking).
- ▶ Las alternativas posibles y las tendencias en su evolución.

AIC – Tema 1 Introducción a las arq. paralelas

Introducción

Evolución

Clasificación

Rendimiento

Problemas

◆ Con el objeto de:

- ▶ Evaluar las prestaciones y características del computador y analizar su comportamiento, identificando los posibles cuellos de botella que limitarían las prestaciones de las aplicaciones.
- ▶ Diseñar y/o configurar un sistema para que se ajuste a los requisitos establecidos.
- ▶ Aprovechar las características del computador para escribir programas, sistemas operativos y compiladores eficaces, que permitan a las distintas aplicaciones obtener el nivel de prestaciones óptimo que requieren.

AIC – Tema 1 Introducción a las arqu. paralelas

Introducción

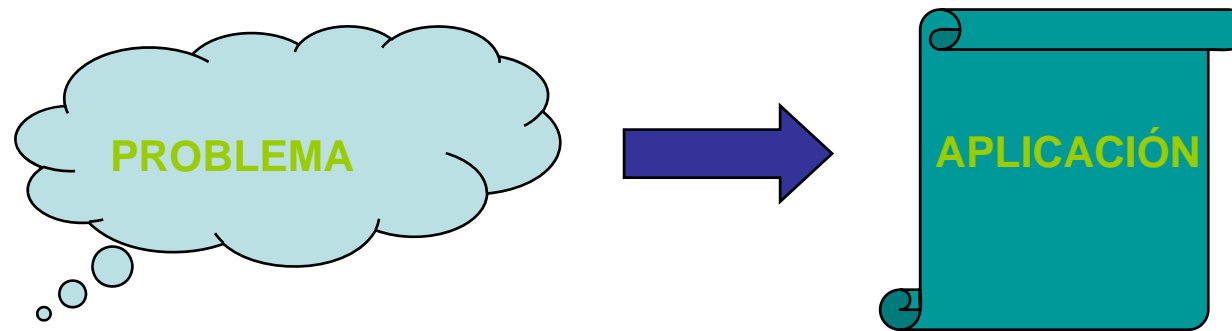
Evolución

Clasificación

Rendimiento

Problemas

◆ Aplicación de los conocimientos de arquitectura:



- ¿Se podrá ejecutar en, por ejemplo, un Pentium IV a 1 Ghz con 512 Mb de SDRAM, en 3 seg.?
- Si la plataforma está un poco ajustada ¿se puede optimizar el código para utilizar mejor los recursos?
- Si no es posible ¿cuál es la mejor opción a adquirir para implementarla: memoria, E/S, procesador, ...longevidad? ¿computador de propósito general o arquitectura específica?

AIC – Tema 1 Introducción a las arq. paralelas

Introducción

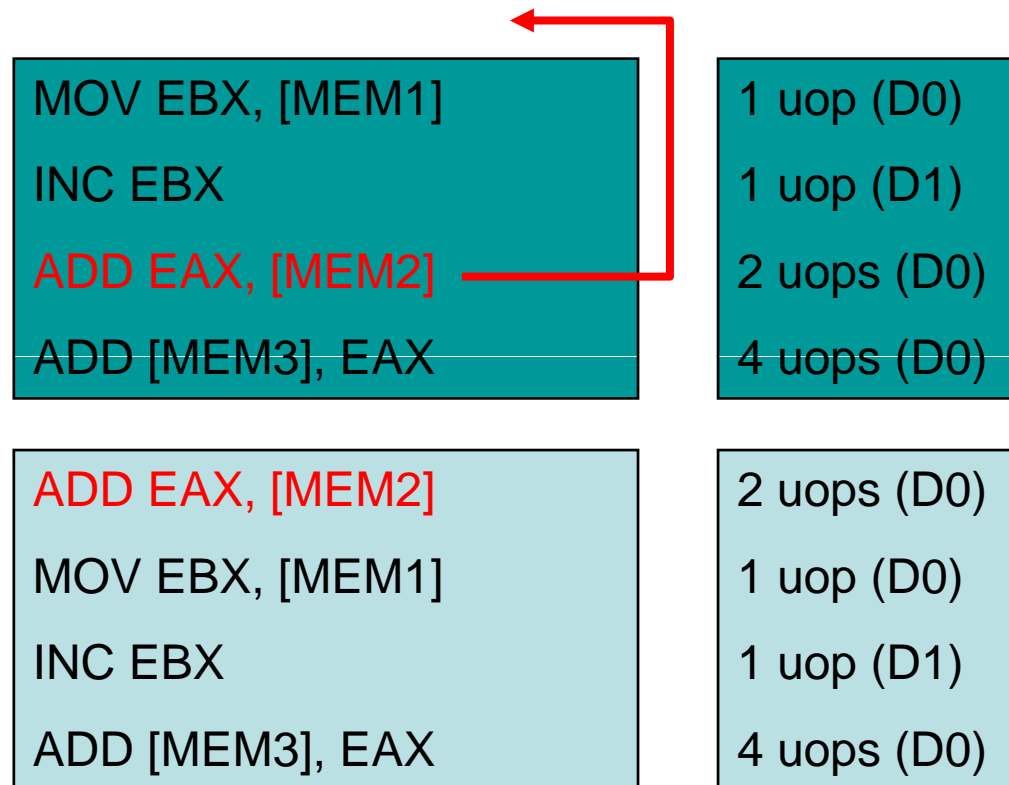
Evolución

Clasificación

Rendimiento

Problemas

Optimización de código (I)



Para **optimizar** código no es necesario utilizar el ensamblador, pero:

- Permite saber si el compilador ha realizado una optimización
- Permite realizar ciertas optimizaciones no realizadas por el compilador (no puede asumir ciertas cosas)

Necesita 2 ciclos frente a los 3 del caso anterior

AIC – Tema 1 Introducción a las arqu. paralelas

Introducción

Evolución

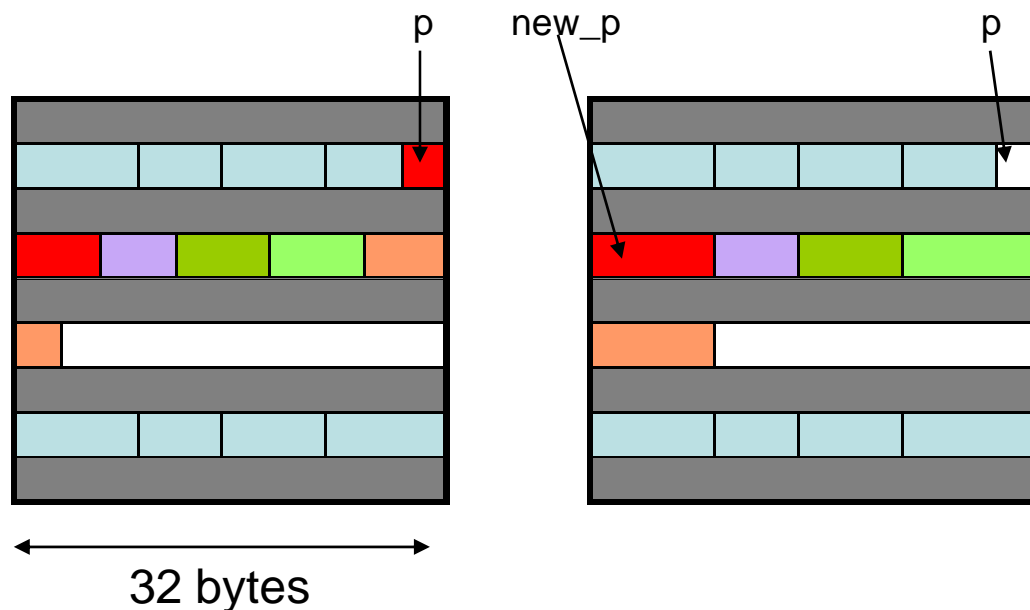
Clasificación

Rendimiento

Problemas

Optimización de código (II)

```
int *p, *new_p;  
p = ( int* ) malloc( sizeof( int ) * N + BOUND );  
new_p = ( int * ) ( ( ( int ) p + BOUND ) & ~ ( BOUND - 1 );
```



Tamaño líneas de caché

Transformaciones de código justificadas desde la arquitectura del computador:

La instrucción **malloc** sólo se entiende desde el punto de vista de la arquitectura (el programa probablemente funcionaría correctamente sin ella)

AIC – Tema 1 Introducción a las arq. paralelas

Introducción

Evolución

Clasificación

Rendimiento

Problemas

Optimización de código (III)

¿Cuál es más rápido en un Pentium?

A

```
switch(val)
{
case 'A': a=val; { *95%* }
break;
case 'B' : a=1; { * 4% * }
break;
case 'C' : a=0; { * 0.5% * }
break;
case 'D' : a=val*2; { * 0.5% * }
break;
}
```

B

```
if (val=='A') a=val;
else if (val=='B') a=1;
switch(val)
{
case 'C' : a=0;
break;
case 'D' : a=val*2;
break;
}
```

AIC – Tema 1 Introducción a las arq. paralelas

Introducción

Evolución

Clasificación

Rendimiento

Problemas

Optimización de código (III)

¿Cuál es más rápido en un Pentium?

El código B se ejecuta 5 veces más rápido en un Pentium.

A

```
switch(val)
{
case 'A': a=val; { *95%* }
break;
case 'B' : a=1; { * 4% * }
break;
case 'C' : a=0; { * 0.5% * }
break;
case 'D' : a=val*2; { * 0.5% * }
break;
}
```

B

```
if (val=='A') a=val;
else if (val=='B') a=1;
switch(val)
{
case 'C' : a=0;
break;
case 'D' : a=val*2;
break;
}
```

AIC – Tema 1 Introducción a las arq. paralelas

Introducción

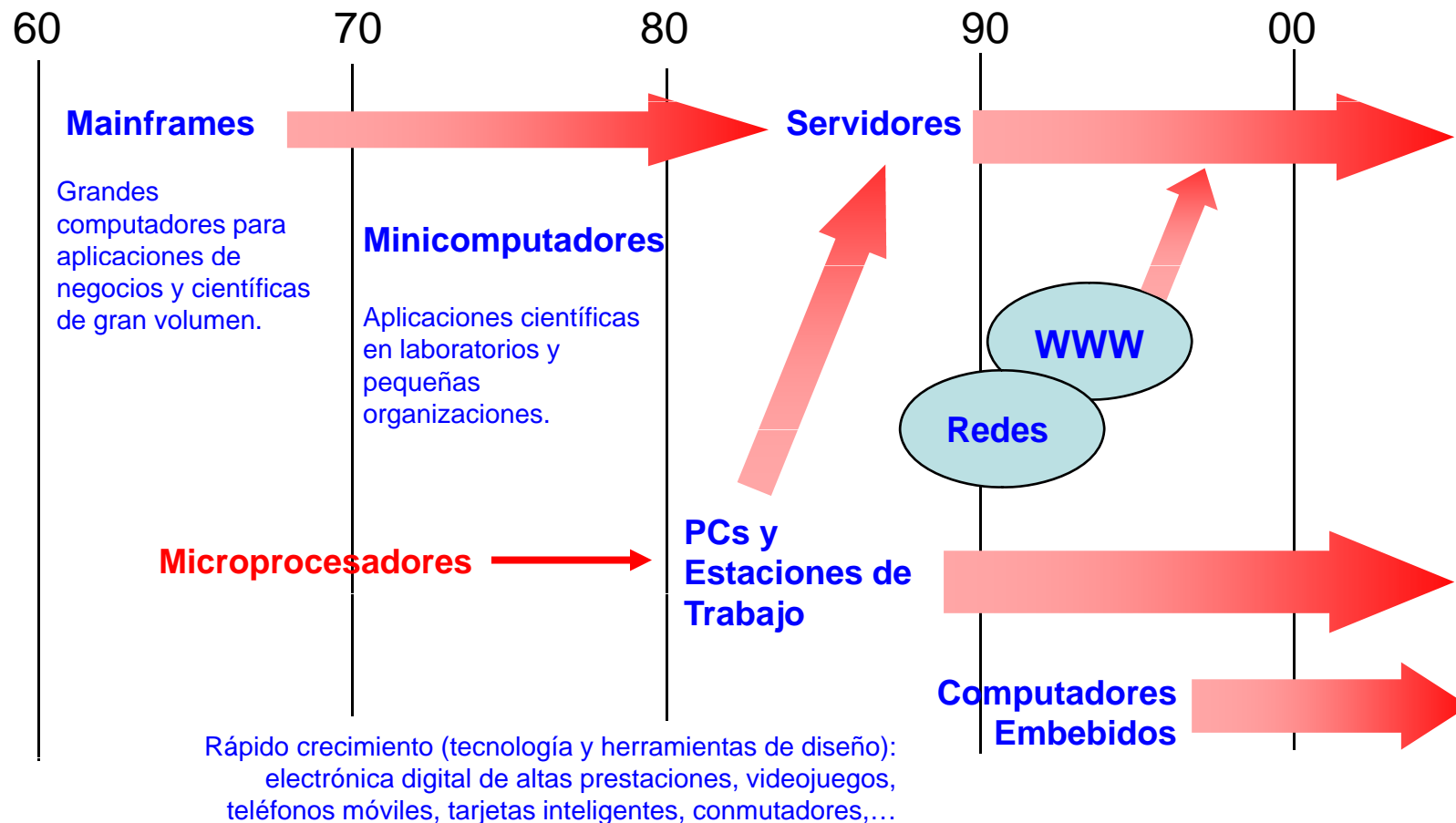
Evolución

Clasificación

Rendimiento

Problemas

◆ Evolución Histórica de los computadores:



AIC – Tema 1 Introducción a las arq. paralelas

Introducción

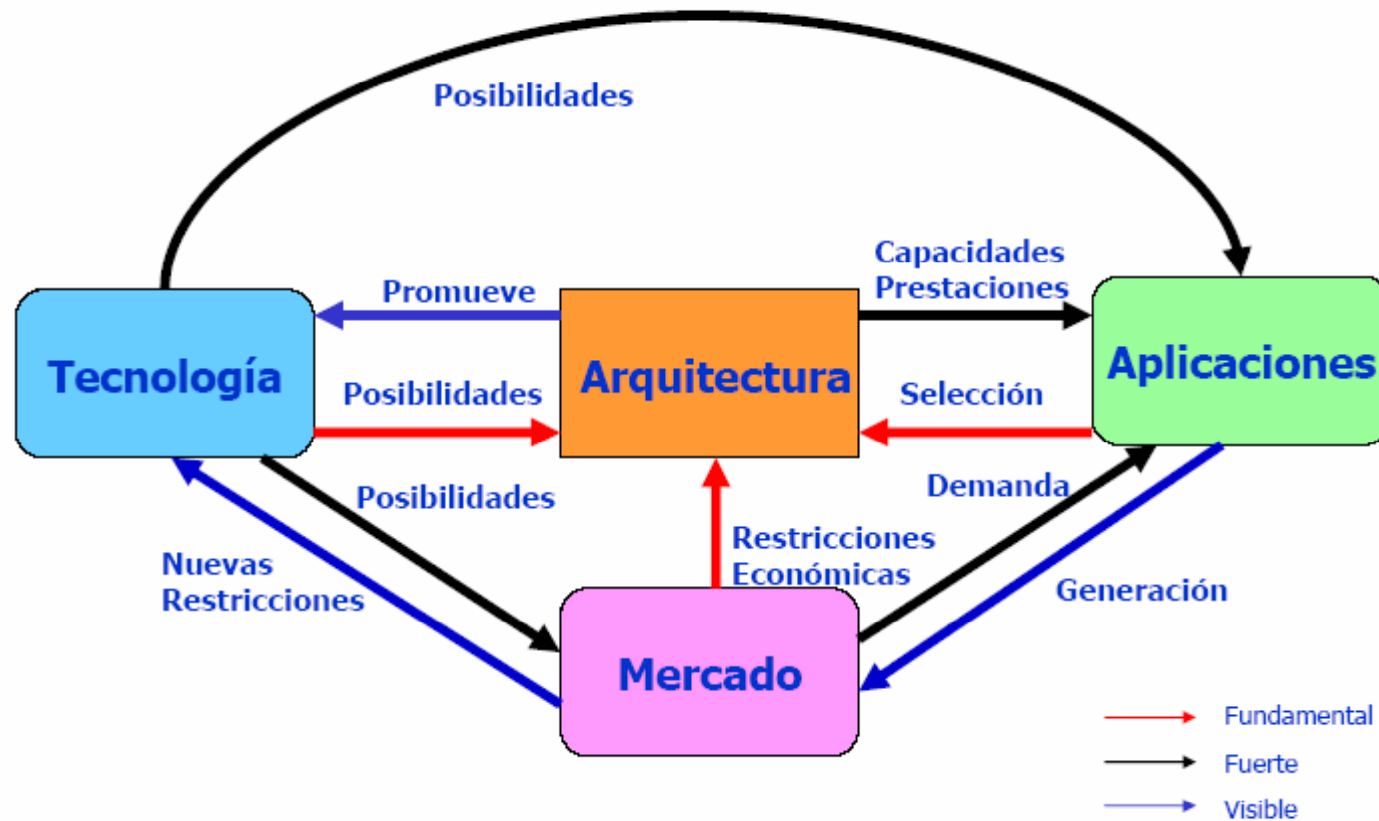
Evolución

Clasificación

Rendimiento

Problemas

❖ Interacción Tecnología-Mercado-Aplicaciones:



AIC – Tema 1 Introducción a las arq. paralelas

Introducción

Evolución

Clasificación

Rendimiento

Problemas

◆ Computadores, Aplicaciones y Mercados:

Aspecto	Sobremesa	Servidores	Empotrados
Precio (\$)	1000 – 10.000	10.000 – 10M	10 – 100.000
Precio por módulo microprocesador (\$)	100 - 1000	200 – 2000 (por procesador)	0.20 – 200 (por procesador)
Microprocesadores vendidos/año (2000)	150M	4M	300M (32 y 64 bits sólo)
Objetivos de diseño	Precio-prestaciones Prestaciones de gráficos (integrar capacidades en los procesadores) e I/O.	Integrar microprocesadores en arquitecturas multiprocesador, I/O de altas prestaciones Fiabilidad, rendimiento, escalabilidad	Incorporar técnicas de microprocesadores de altas prestaciones con las restricciones de precio, consumo. Altas prestaciones específicas para la aplicación

AIC – Tema 1 Introducción a las arqu. paralelas

Introducción

Evolución

Clasificación

Rendimiento

Problemas

◆ Evolución; existe en el mercado una demanda computacional creciente:

▶ **Aplicaciones científicas:**

- Evolución de las galaxias
- Proyecto Genoma Humano
- Proyecto “Cancer Research” de United Devices

▶ **Ingeniería:**

- Electromagnetismo y simulación de circuitos
- CAD, mecánica estructural
- Simulación de choques
- Métodos de análisis de elementos finitos

▶ **Sistemas Gestores de Bases de Datos:**

- Lenguaje natural, paralelización de consultas

▶ **Inteligencia Artificial:**

- Modelado conexionista
- Análisis de imágenes, visualización y rendering

▶ **Desarrollo de películas (Toy Story) (117 Sun SparcStations)**

AIC – Tema 1 Introducción a las arq. paralelas

Introducción

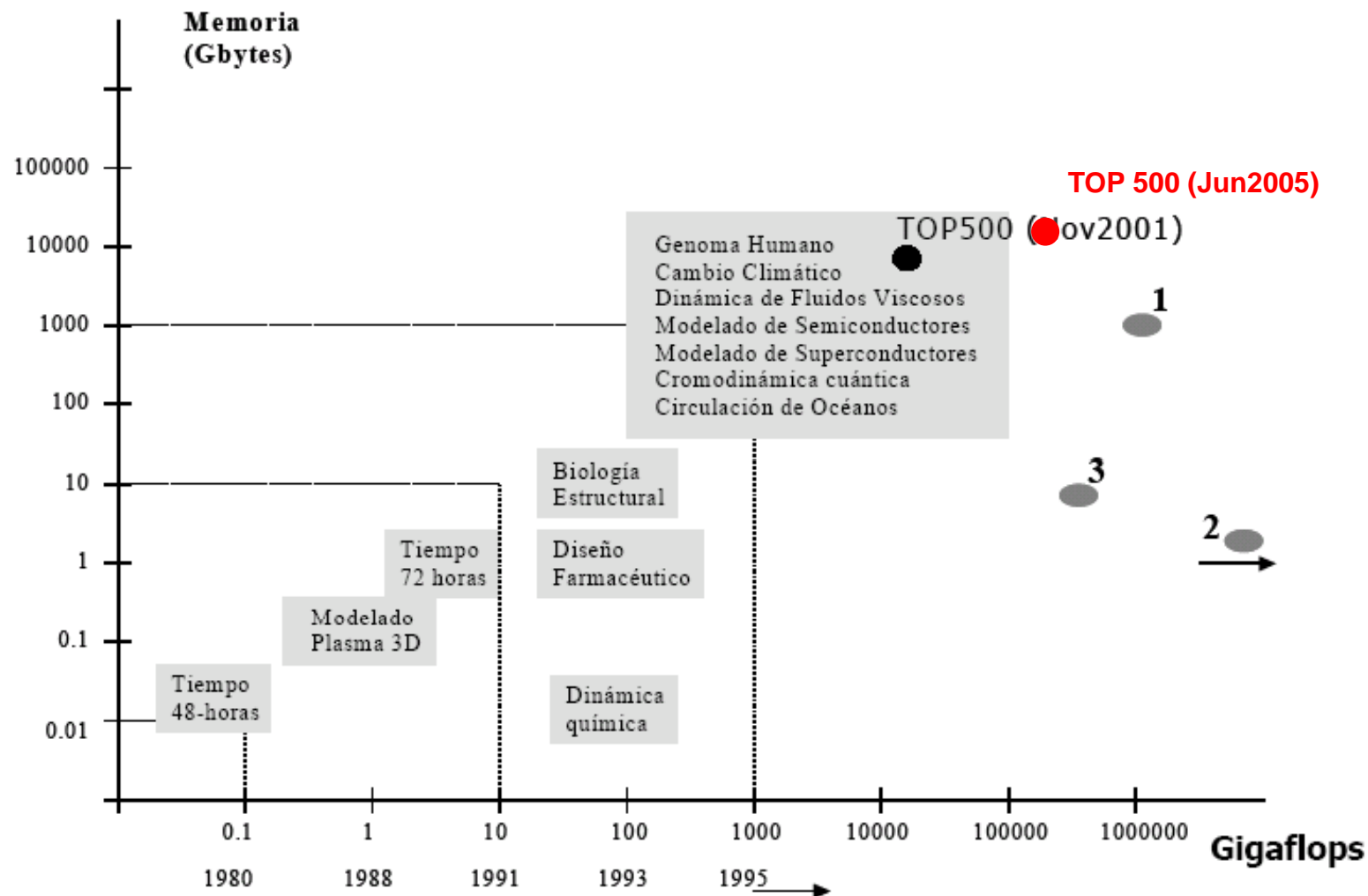
Evolución

Clasificación

Rendimiento

Problemas

◆ Evolución Cálculo Científico:



AIC – Tema 1 Introducción a las arq. paralelas

Introducción

Evolución

Clasificación

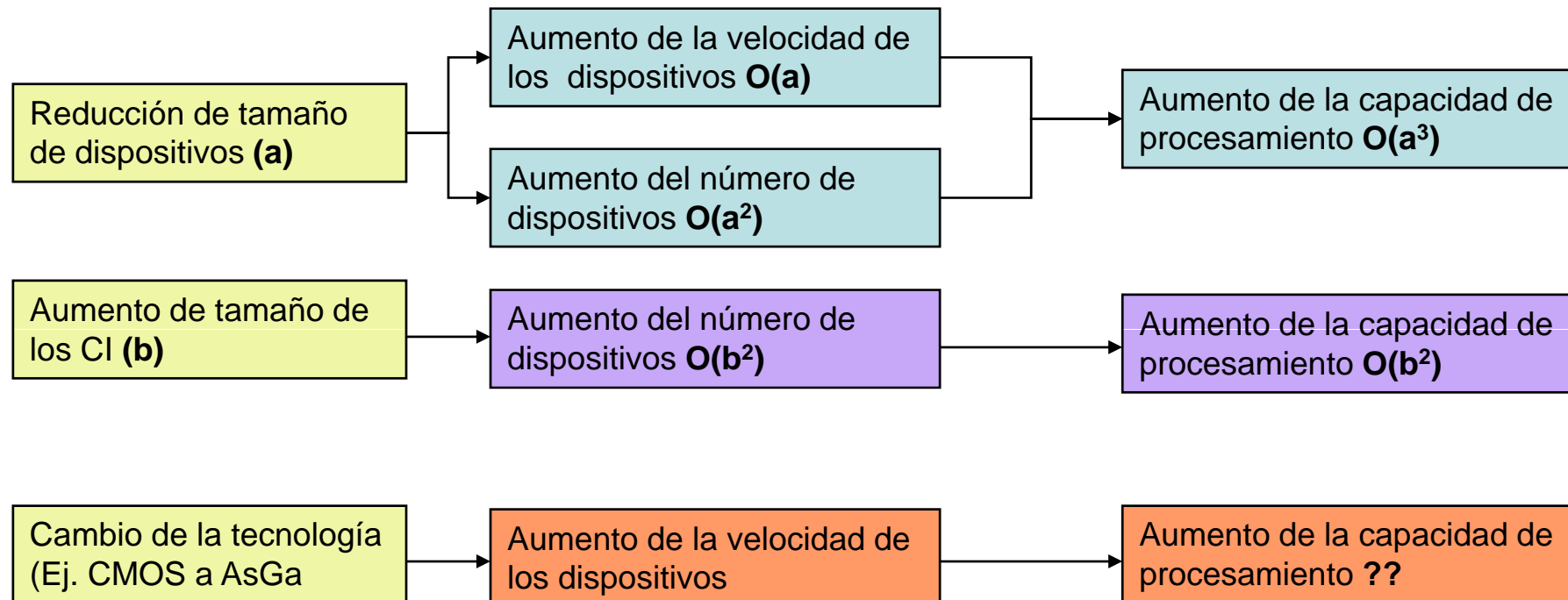
Rendimiento

Problemas

◆ Mejora de prestaciones:

► Avances en la tecnología (I):

- **Limites:** conforme se disminuye el tamaño de los transistores aumenta su velocidad, pero también aumenta la resistencia de las líneas de conexión entre ellos, y por tanto el retardo de las comunicaciones dentro del chip.



AIC – Tema 1 Introducción a las arq. paralelas

Introducción

Evolución

Clasificación

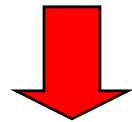
Rendimiento

Problemas

► Avances en la Tecnología (II):

¿A qué ritmo mejora la tecnología?

- **Densidad de transistores:** 35% anual (DRAMs: 40%-60% anual)
- **Velocidad de transistores:** 16% anual (DRAMs: latencia un 33% y ancho de banda por chip un 66% en 10 años)
- **Tamaño del dado:** 10%-20% anual
- **Número de Transistores/Chip:** 55% anual



Ley de Moore(1979): *“El número de transistores por circuito integrado se dobla cada 18-24 meses (asumiendo precio constante del circuito integrado)”*

Entre un 40% y un 57% de mejora anual

El aumento de la densidad de dispositivos, su velocidad, y el tamaño de los CI **permitirán hasta un 72%-88%** de mejora anual de la **capacidad de procesamiento** de un CI.

AIC – Tema 1 Introducción a las arq. paralelas

Introducción

Evolución

Clasificación

Rendimiento

Problemas

► Avances en la Tecnología (III):

Aumento de la **densidad de transistores** y el funcionamiento a **frecuencias** mayores ➡ Aumento de la **potencia disipada** por los circuitos integrados

Ejemplo: en el Alpha, la potencia consumida casi ha llegado a 30 W/cm²: *tres veces la usual de un plato caliente*

Micro	W (pico)	MHz	Mm ²	V
21064	30	200	234	3.3
21164	50	300	299	3.3
21264	72	667	302	2.0
21364	100	1000	350	1.5

El consumo es esencial en:

- Plataformas portátiles y móviles
- Servidores de altas prestaciones a base de computadores interconectados (clusters)

Dato: 8000 computadores y 2300 m² de Intel consume unos 2 Megawatios (25% del costo de mantenimiento)

Las Tecnologías de la información, suponen en USA el 8% del consumo nacional eléctrico.

AIC – Tema 1 Introducción a las arq. paralelas

Introducción

Evolución

Clasificación

Rendimiento

Problemas

► **Mejoras en la Arquitectura:** se basan en 2 principios

- **Paralelismo:**

- **Segmentación de cauces:** Cada elemento se divide en una serie de etapas por las que van pasando los operandos, instrucciones,... al ser procesados. El funcionamiento independiente de cada etapa permite que se puedan ejecutar distintas operaciones, instrucciones,.. concurrentemente.
 - Diseño de procesadores (paralelismo entre instrucciones): procesadores segmentados (procesadores superescalares y VLIW, procesadores vectoriales).
 - Diseño de unidades funcionales (sumadores segmentados).
- **Repetición de elementos:** Utilizar varias unidades funcionales, procesadores, módulos de memoria, etc. para distribuir el trabajo.
 - Nivel de sistema (multiprocesadores, controladores DMA, etc.)
 - Nivel de procesador (superescalares, vectoriales, etc.)

- **Localidad:** Acercar datos e instrucciones al lugar donde se necesitan para que el acceso a los mismos sea lo más rápido posible (jerarquía de memoria).

AIC – Tema 1 Introducción a las arq. paralelas

Introducción

Evolución

Clasificación

Rendimiento

Problemas

◆ Clasificación de las arquitecturas:

- ▶ Como toda clasificación, una clasificación (o taxonomía) de arquitecturas persigue dividir el conjunto de los computadores en una serie de clases de forma que, si se sabe la clase a la que pertenece un computador, automáticamente se conocen una serie de características interesantes del mismo.
- ▶ La clasificación más extendida es la llamada “**Taxonomía de Flynn**”:
 - **SISD**
 - **SIMD**
 - **MISD**
 - **MIMD**

AIC – Tema 1 Introducción a las arq. paralelas

Introducción

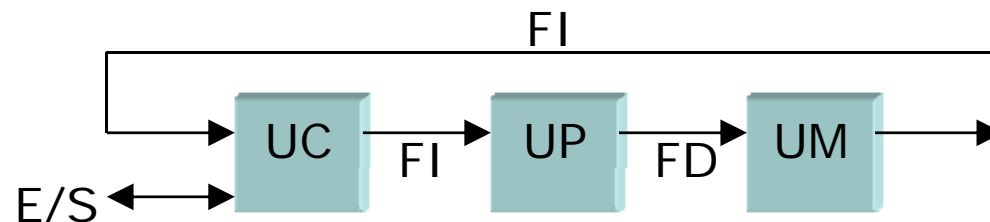
Evolución

Clasificación

Rendimiento

Problemas

- **Computadores SISD:** un único flujo de instrucciones procesa operandos y genera resultados, definiendo un único flujo de datos.



```
for i=1 to 4 do
Begin
  C[i] = A[i]+B[i];
  F[i] = D[i]-E[i];
  G[i] = K[i]*H[i];
End;
```

AIC – Tema 1 Introducción a las arq. paralelas

Introducción

Evolución

Clasificación

Rendimiento

Problemas

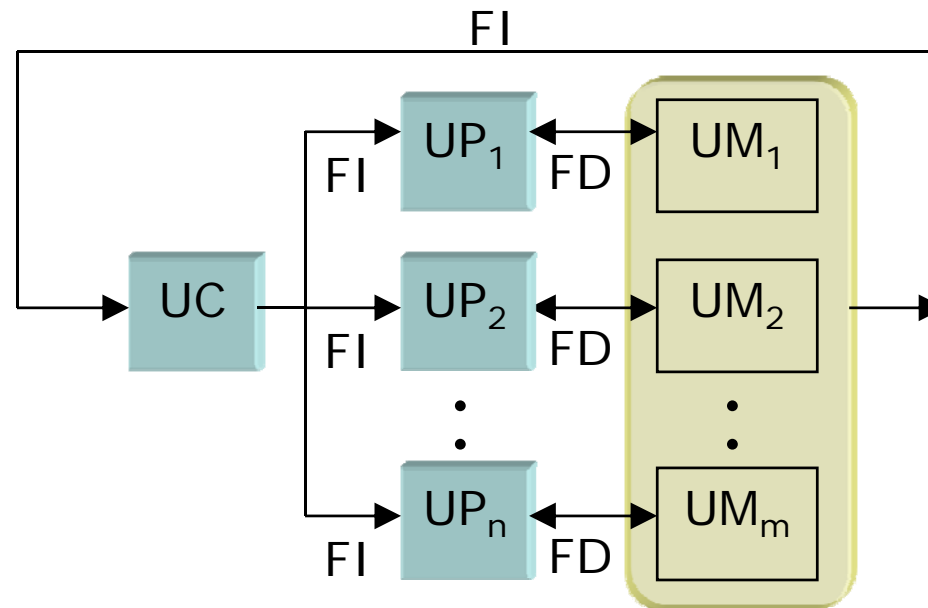
- **Computadores SIMD:** un único flujo de instrucciones procesa operandos y genera resultados, definiendo varios flujos de datos, dado que cada instrucción codifica realmente varias operaciones iguales, cada una actuando sobre operadores distintos.

```
for all Epi(i=1 to 4) do  
Begin  
  C[i] = A[i]+B[i];  
  F[i] = D[i]-E[i];  
  G[i] = K[i]*H[i];  
End;
```

Procesadores matriciales

```
ADDV C,A,B  
SUBV F,D,E  
MULV G,K,H
```

Procesadores vectoriales



AIC – Tema 1 Introducción a las arq. paralelas

Introducción

Evolución

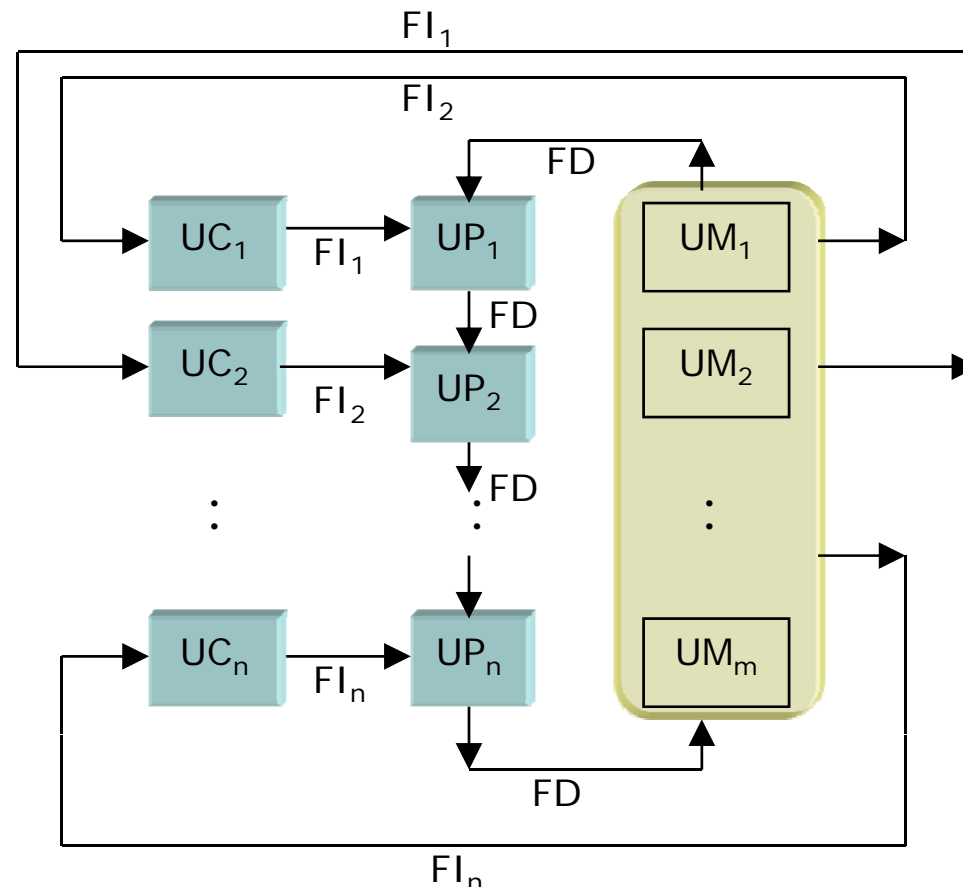
Clasificación

Rendimiento

Problemas

► **Computadores MISD:** se ejecutan varios flujos distintos de instrucciones (MI) aunque todos actúan sobre el mismo flujo de datos.

- *Actualmente no existen computadores que funcionen bajo este esquema.*



AIC – Tema 1 Introducción a las arq. paralelas

Introducción

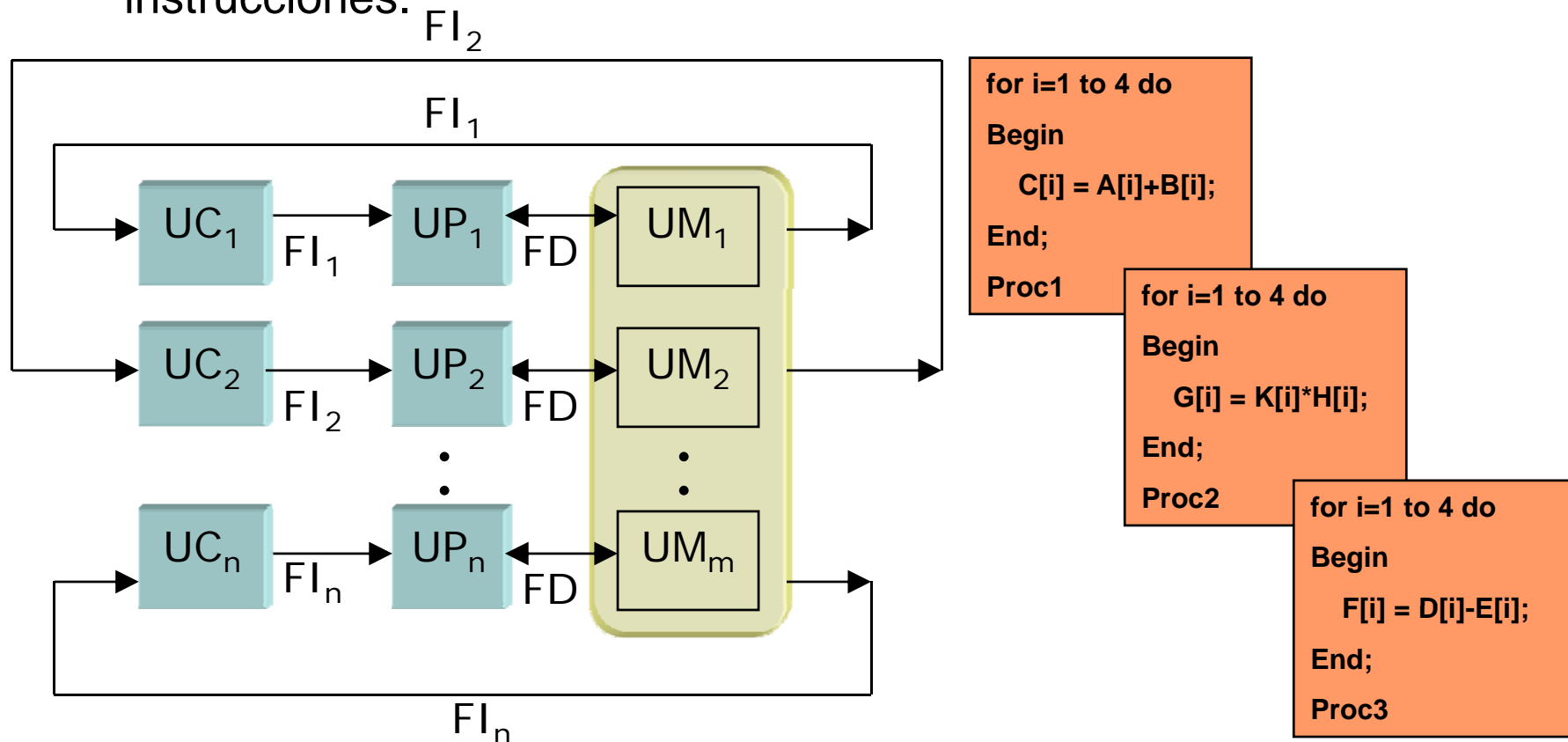
Evolución

Clasificación

Rendimiento

Problemas

- **Computadores MIMD:** el computador ejecuta varias secuencias o flujos distintos de instrucciones, y cada uno de ellos procesa operandos y genera resultados definiendo un único flujo de instrucciones, de forma que existen también varios flujos de datos uno por cada flujo de instrucciones.



AIC – Tema 1 Introducción a las arq. paralelas

Introducción

Evolución

Clasificación

Rendimiento

Problemas

Taxonomía de Flynn	Paralelismo Funcional	Paralelismo de Datos
SISD	Proc. Segmentados Proc. Superescalares Proc. VLIW	
SIMD		Proc. Vectoriales
		Proc. Matriciales Proc. Sistólicos
MIMD	Memoria Compartida: Acceso Uniforme (SMPs), Acceso no Uniforme o Distribuida (NUMA, COMA). Paso de Mensajes (NORMA): Multicomputadores, Clusters.	

AIC – Tema 1 Introducción a las arq. paralelas

Introducción

Evolución

Clasificación

Rendimiento

Problemas

◆ Tipos de paralelismo:

▶ **Paralelismo de datos:** La misma función, instrucción, etc. se ejecuta en paralelo pero en cada una de esas ejecuciones se aplica sobre un conjunto de datos distinto.

▶ **Paralelismo funcional:** Varias funciones, tareas, instrucciones, etc. (iguales o distintas) se ejecutan en paralelo. Se distinguen los siguientes niveles (según el tipo de entidades funcionales que se ejecutan en paralelo):

- **Nivel de instrucción (ILP)** – se ejecutan en paralelo las instrucciones de un programa. Granularidad fina.
- **Nivel de bucle o hebra (Thread)** – *se ejecutan en paralelo distintas iteraciones de un bucle o secuencias de instrucciones de un programa. Granularidad fina/media.*
- **Nivel de procedimiento (Proceso)** – los distintos procedimientos que constituyen un programa se ejecutan simultáneamente. Granularidad media.
- **Nivel de programa** – la plataforma ejecuta en paralelo programas diferentes que pueden corresponder, o no, a una misma aplicación. Granularidad gruesa.

AIC – Tema 1 Introducción a las arqu. paralelas

Introducción

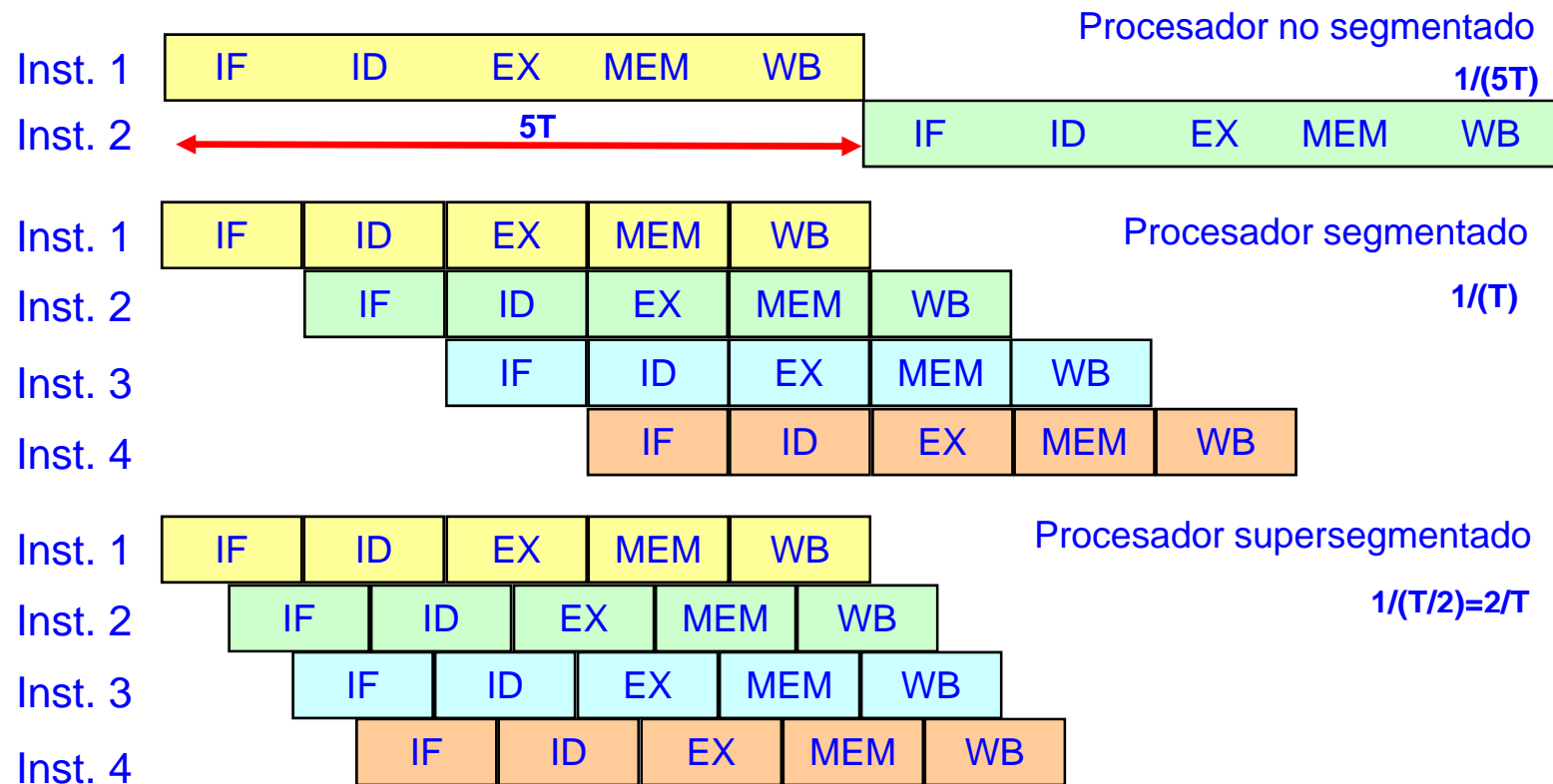
Evolución

Clasificación

Rendimiento

Problemas

◆ Ejemplo: paralelismo entre instrucciones ILP (I)



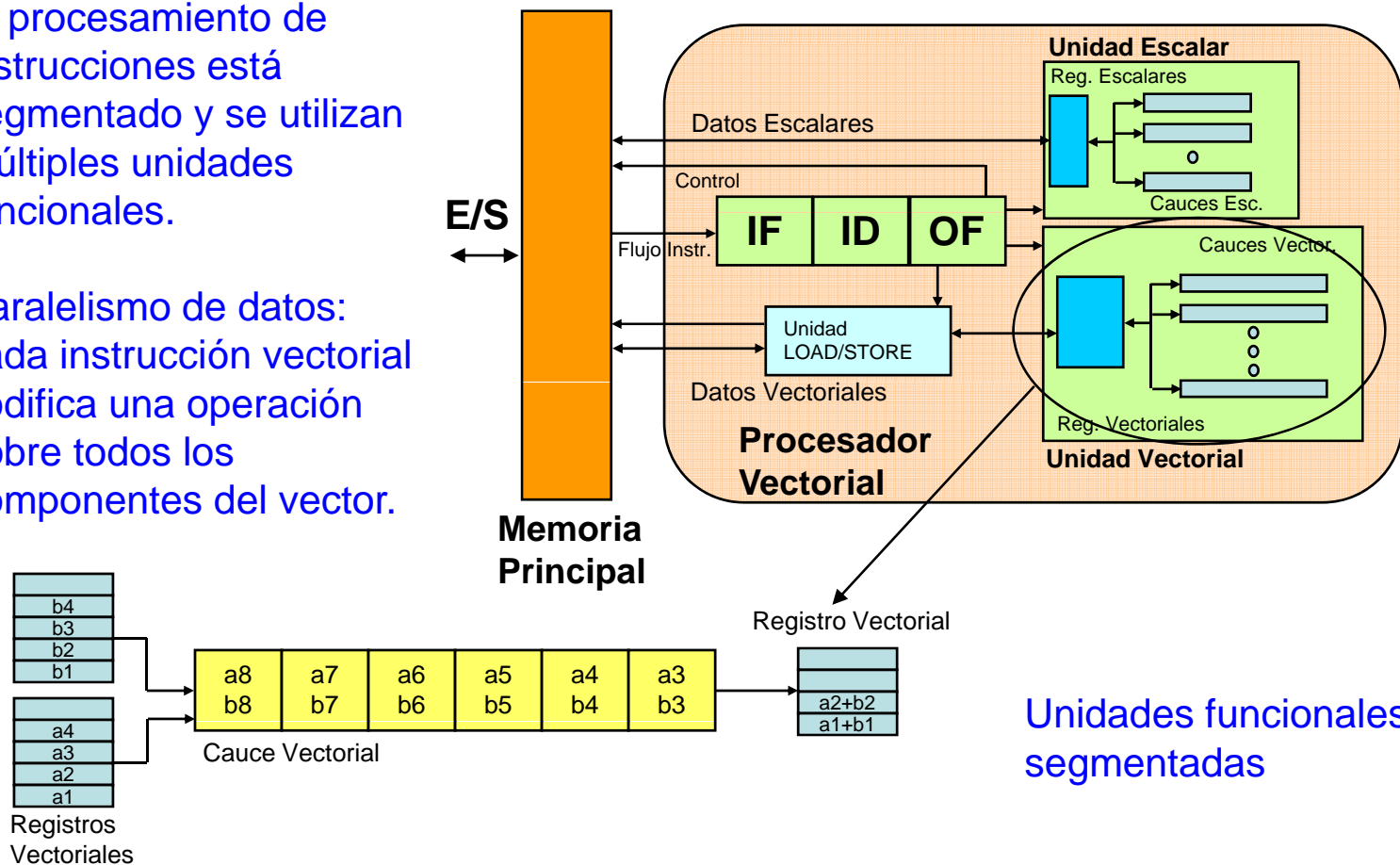
AIC – Tema 1 Introducción a las arq. paralelas

Introducción Evolución **Clasificación** Rendimiento Problemas

◆ Ejemplo: Procesadores Vectoriales: ILP y Paralelismo de Datos.

El procesamiento de instrucciones está segmentado y se utilizan múltiples unidades funcionales.

Paralelismo de datos: cada instrucción vectorial codifica una operación sobre todos los componentes del vector.



AIC – Tema 1 Introducción a las arq. paralelas

Introducción

Evolución

Clasificación

Rendimiento

Problemas

◆ Tiempo de ejecución:

- ▶ Tiempo necesario para completar la ejecución de un programa (*response time, elapsed time*).
- ▶ Está formado por:

- **Tiempo de CPU de usuario:** tiempo que la CPU invierte en ejecutar el programa.
- **Tiempo de CPU de sistema:** tiempo que la CPU invierte en ejecutar las tareas de SO necesarias para el programa.
- **Tiempo asociado a:** esperas debidas a **E/S** o a la **ejecución** de otros programas.

- ▶ Ejemplo: Comando **time** (Unix): 63.0u 15.0s 2:10 60%
 - 63.0u = tiempo CPU de usuario
 - 15.0s = tiempo de CPU de sistema
 - 63.0 + 15.0 es el 60% del total (2:10 = 130seg)

AIC – Tema 1 Introducción a las arq. paralelas

Introducción

Evolución

Clasificación

Rendimiento

Problemas

$$\text{Tiempo de CPU (T}_{\text{CPU}}\text{)} = \text{Ciclos_del_Programa} \times T_{\text{CICLO}} = \frac{\text{Ciclos_del_Programa}}{\text{Frecuencia_de_Reloj}}$$

$$\text{Ciclos por Instrucción (CPI)} = \frac{\text{Ciclos_del_Programa}}{\text{Número_de_Instrucciones (NI)}}$$

$$T_{\text{CPU}} = \text{NI} \times \text{CPI} \times T_{\text{CICLO}}$$

$$T_{\text{CICLO}} = 1/F$$

F=frecuencia

AIC – Tema 1 Introducción a las arq. paralelas

Introducción

Evolución

Clasificación

Rendimiento

Problemas

- ◆ Para **procesadores** que pueden empezar la ejecución (**emitir**) varias instrucciones al mismo tiempo:

$$T_{\text{CPU}} = NI \times \underbrace{(CPE / IPE)}_{\text{CPI}} \times T_{\text{CICLO}}$$

CPI

CPE = ciclos entre inicio de emisión de instrucciones.

IPE = instrucciones que pueden emitirse (empezar la ejecución) cada vez que se produce ésta.

AIC – Tema 1 Introducción a las arq. paralelas

Introducción

Evolución

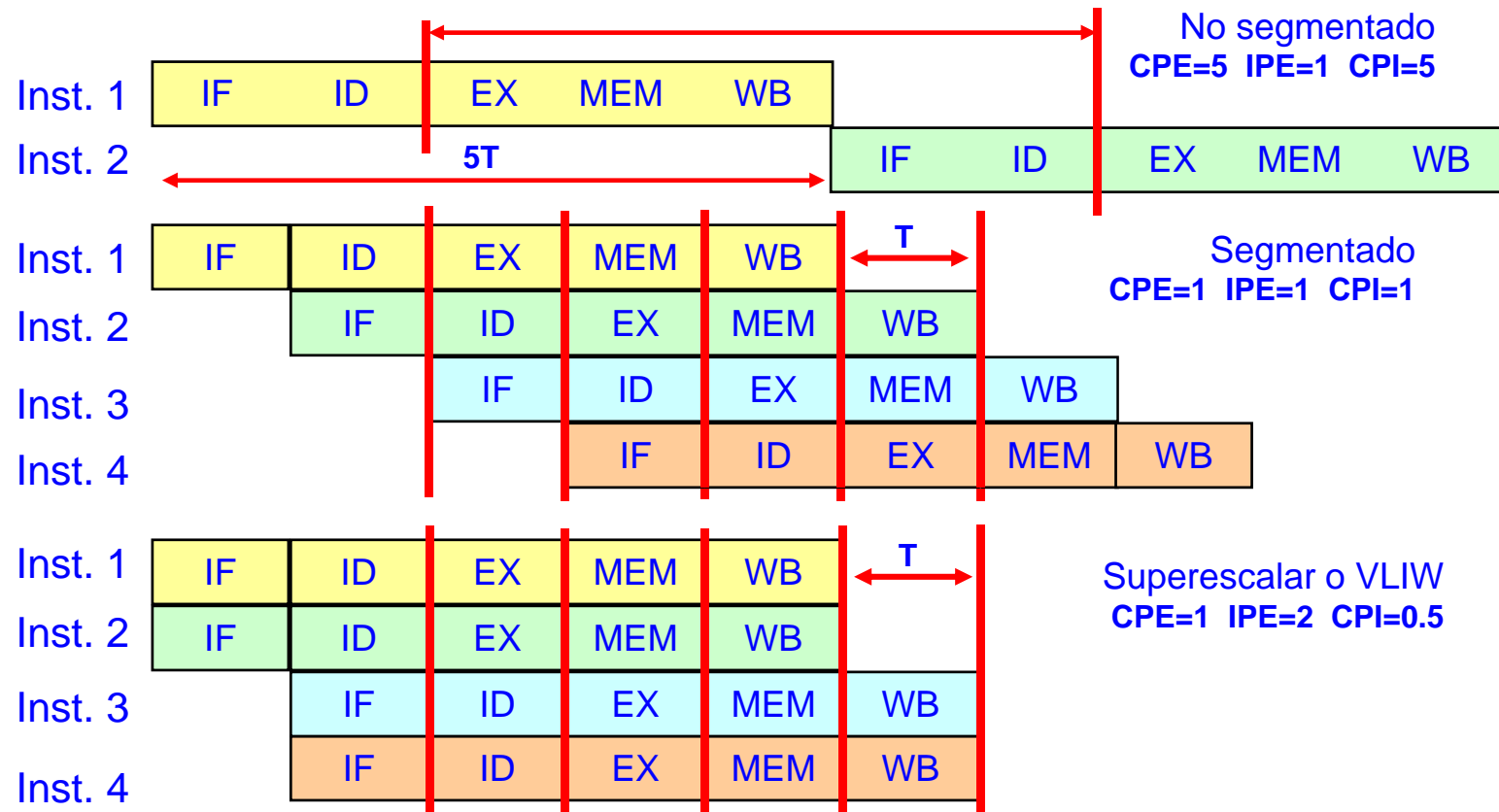
Clasificación

Rendimiento

Problemas

Ejemplo:

$$\text{CPI} = \text{CPE} / \text{IPE}$$



AIC – Tema 1 Introducción a las arq. paralelas

Introducción Evolución Clasificación Rendimiento Problemas

- ◆ Procesadores que pueden codificar varias operaciones en una instrucción:

$$T_{\text{CPU}} = (\text{Noper} / \text{Op_instr}) \times \text{CPI} \times T_{\text{CICLO}}$$

NI

Noper = número de operaciones que realiza el programa.

Op_instr = número de operaciones que puede codificar una instrucción.

AIC – Tema 1 Introducción a las arq. paralelas

Introducción

Evolución

Clasificación

Rendimiento

Problemas

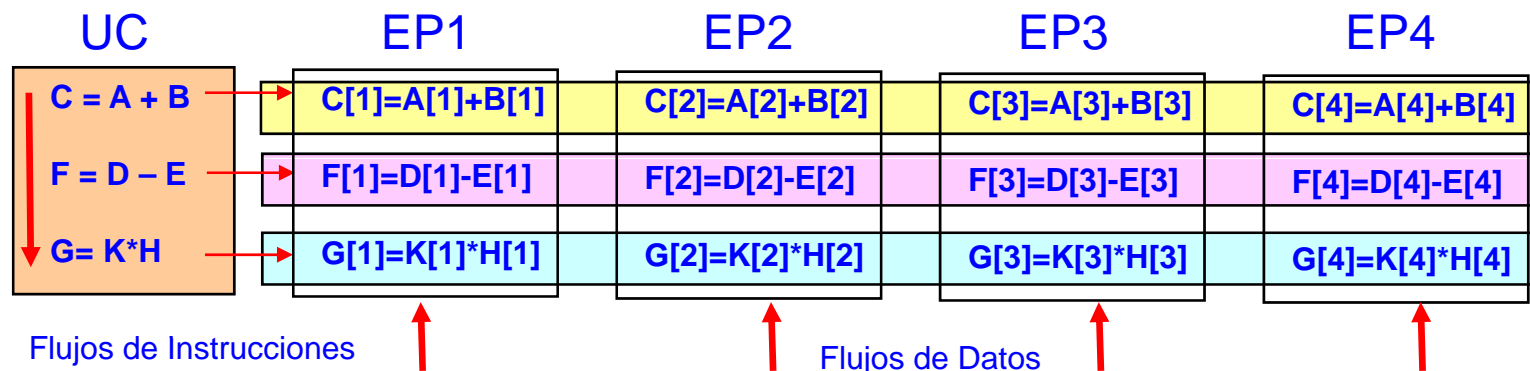
Ejemplo:

NI = Noper/Op_instr

Ejemplo paralelismo datos

Procesador Matricial

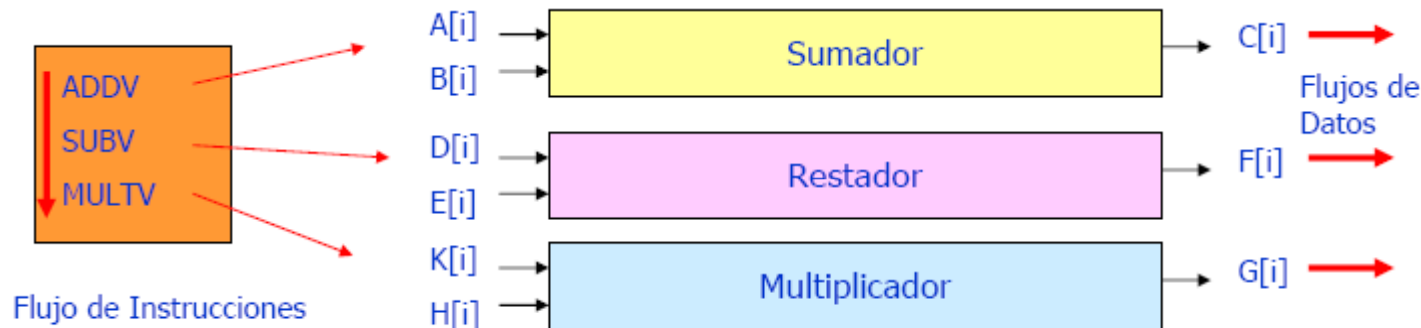
Noper=12 Op_instr=4 NI=3



Ejemplo paralelismo instrucciones

Procesador Vectorial

Noper=12 Op_instr=4 NI=3



AIC – Tema 1 Introducción a las arq. paralelas

Introducción

Evolución

Clasificación

Rendimiento

Problemas

◆ Mejora del tiempo de CPU:

▶ *Tciclo*

- **Tecnología:** Dispositivos más rápidos. Integración procesador + On-chip caches.
- **Estructura y organización:** Jerarquía de memoria, diseño de menos niveles de lógica por etapas, supersegmentación.

▶ *CPI*

- **Tecnología:** Mayor integración.
- **Estructura y organización:** ILP (segmentación de cauce, VLIW, superescalares). En superescalares y VLIW $\rightarrow CPI = CPE / IPE$
- **Repertorio de instrucciones:** Instrucciones con implementación sencilla.
- **Compilador:** Extracción del máximo ILP.

▶ *NI*

- **Repertorio de instrucciones:** CISC \rightarrow RISC, repertorio VLIW, repertorio vectorial.

AIC – Tema 1 Introducción a las arq. paralelas

Introducción

Evolución

Clasificación

Rendimiento

Problemas

◆ Medidas de prestaciones:

► **MIPS:** millones de instrucciones por segundo.

$$\text{MIPS} = \frac{\text{NI}}{T_{\text{CPU}} \times 10^6} = \frac{F(\text{frecuencia})}{\text{CPI} \times 10^6}$$

- Depende del repertorio de instrucciones (difícil comparar entre máquinas diferentes).
- Puede variar con el programa (no caracteriza la máquina).
- Puede variar inversamente con las prestaciones (mayor valor de MIPS corresponde a peores prestaciones).

AIC – Tema 1 Introducción a las arq. paralelas

Introducción

Evolución

Clasificación

Rendimiento

Problemas

- ▶ **MFLOPS:** millones de operaciones en coma flotante por segundo.

$$\text{MFLOPS} = \frac{\text{Operaciones_en_Coma_Flotante}}{T_{\text{CPU}} \times 10^6}$$

- No es adecuado para todos los programas (sólo tiene en cuenta las operaciones en coma flotante).
- El conjunto de operaciones en coma flotante no es constante en todas las máquinas.

AIC – Tema 1 Introducción a las arq. paralelas

Introducción

Evolución

Clasificación

Rendimiento

Problemas

- ◆ **Lo más adecuado:** utilizar **benchmarks** (conjuntos de programas) seleccionados que representen la carga de trabajo usual de las máquinas a evaluar.
- ◆ **Tipos de benchmarks:**
 - ▶ **Aplicaciones reales.** Compiladores, Word, MathLab, ...
 - ▶ **Kernels.** Trozos de aplicaciones reales seleccionados para evaluar características específicas.
 - ▶ **Simples (Toys).** Pequeños programas fáciles de programar y cuyo resultado es conocido (Quicksort).
 - ▶ **Sintéticos.** Reproducen porcentajes de instrucciones y situaciones de carga reales.
- ◆ **Benchmarks suites (SPEC).** Cjts. de Benchmarks que miden las prestaciones de los computadores a través de un conjunto de aplicaciones distintas. Las limitaciones de un benchmark se suplen con la presencia de otros. Se cambian periódicamente para evitar optimizaciones realizadas con el único objetivo de mejorar los resultados del conjunto de benchmark.

AIC – Tema 1 Introducción a las arq. paralelas

Introducción

Evolución

Clasificación

Rendimiento

Problemas

◆ Conjunto de benchmarks del SPEC2000.

- ▶ 11 benchmarks enteros
- ▶ 14 benchmarks coma-flotante

CINT2000 (en C y en C++ el 252.eon)

164.gzip 1400 Data compression utility
175.vpr 1400 FPGA circuit placement and routing
176.gcc 1100 C compiler
181.mcf 1800 Minimum cost network flow solver
186.crafty 1000 Chess program
197.parser 1800 Natural language processing
252.eon 1300 Ray tracing
253.perlbmk 1800 Perl
254.gap 1100 Computational group theory
255.vortex 1900 Object Oriented Database
256.bzip2 1500 Data compression utility
300.Twof 3000 Place and route simulator

CPF2000 (6 Fortran 77, 4 Fortran 90, y 4 C)

168.wupwise 1600 Quantum chromodynamics
171.swim 3100 Shallow water modeling
172.mgrid 1800 Multi-grid solver in 3D potential field
173.aplu 2100 Parabolic/elliptic differential equations
177.mesa 1400 3D Graphics library
178.galgel 2900 Fluid dynamics: oscillatory instability
179.art 2600 Neural network simulation (ART)
183.quake 1300 Finite element simulation: earthquake
187.facerec 1900 Computer vision: recognizes faces
188.amp 2200 Computational chemistry
189.lucas 2000 Number theory: primality testing
191.fma3d 2100 Finite element crash simulation
200.sixtrack 1100 Particle accelerator model
301.apsi 2600 Solves problems regarding pollutants

<http://www.specbench.org/osg/cpu2000/docs/readme1st.html>

AIC – Tema 1 Introducción a las arq. paralelas

Introducción

Evolución

Clasificación

Rendimiento

Problemas

◆ Benchmarks suites para la evaluación de computadores de altas prestaciones:

▶ SPEC HPC :

- Predicción de tiempo atmosférico.
- Aplicación de química cuántica.
- Localización de yacimientos de petróleo y gas.

▶ LINPACK:

- Conjunto de subrutinas de resolución y análisis de ecuaciones lineales, sistemas de mínimos cuadrados, etc.
- La lista TOP500 que evalúa los 500 computadores más potentes utiliza LINPACK como benchmark.

AIC – Tema 1 Introducción a las arq. paralelas

Introducción

Evolución

Clasificación

Rendimiento

Problemas

◆ Ejemplo: Earth Simulator (TOP 500: número 1 en 2002)

El núcleo de este programa es una rutina denominada *daxpy* que multiplica un vector por una constante y los suma a otro vector:

```
for (i=0 ; i<N ; i++)  
  y[i] = y[i] + a*x[i];
```

Valor de **N** para alcanzar la mitad de las prestaciones

- Basado en la arquitectura NEC SX (vectorial).
- Dispone de 640 nodos, con 8 procesadores vectoriales (8 Gflop/s pico por procesador), 2 ns de tiempo de ciclo, 16GB de memoria principal compartida por nodo.
- Total: 5120 procesadores, 40 Tflop/s pico, y 10 TB de memoria.
- Una única red de barras cruzadas (1800 millas de cable) 83,000 cables de cobre, 16 GB/s cross section bandwidth (ancho de banda de sección).
- 700 TB de disco
- 1.6 PB almacenamiento masivo
- Dimensiones: 4 pistas de tenis, 3 plantas
- Linpack Benchmark = 35.6 Tflop/s
- Tamaño **N** = 1,041,216; (8.7 TB de memoria)
- Mitad del máximo **N**_{1/2} = 265,408
- Tiempo de ejecución del benchmark: 5.8 horas
- Software: MPI/Fortran

AIC – Tema 1 Introducción a las arq. paralelas

Introducción

Evolución

Clasificación

Rendimiento

Problemas

◆ Rendimiento:

- ▶ **Ganancia de velocidad:** incremento de velocidad que se consigue gracias al aumento de recursos de una máquina respecto a la máquina base.

$$G_P = \frac{T_1}{T_P}; \quad G_P \leq P$$

- ▶ **Eficiencia:** se obtiene dividiendo la ganancia entre el número de recursos (permite evaluar en qué medida las prestaciones de un programa se acercan al máximo).

$$E_P = \frac{G_P}{P}; \quad E_P \leq 1$$

- ▶ **Productividad:** número de operaciones que se ejecutan por unidad de tiempo.

AIC – Tema 1 Introducción a las arq. paralelas

Introducción

Evolución

Clasificación

Rendimiento

Problemas

Kilobyte (KB)	10^3 bytes	Kibibyte (KiB)	1024 bytes (2^{10} bytes)
Megabyte (MB)	10^6 bytes	Mebibyte (MiB)	2^{20} bytes
Gigabyte (GB)	10^9 bytes	Gibibyte (GiB)	2^{30} bytes
Terabyte (TB)	10^{12} bytes	Tebibyte (TiB)	2^{40} bytes