



Tema 1:

Introducción a la arquitectura de computadores. Análisis de prestaciones.

Arquitectura de Computadores
Grado en Ingeniería Informática

Mercedes Rodríguez García

Índice

1. Introducción a la arquitectura de computadores
 - 1.1. Niveles de descripción de un computador
 - 1.2. Ámbito de estudio
 - 1.3. Objeto de estudio
2. Clasificación de arquitecturas
 - 2.1. Taxonomía de Flynn
 - 2.1.1. Computadores SISD
 - 2.1.2. Computadores SIMD
 - 2.1.3. Computadores MIMD
 - 2.1.3.1. Tipos de computadores MIMD
 - 2.1.4. Computadores MISD
 - 2.2. Paralelismo a nivel de instrucción (ILP)
 - 2.2.1. Procesador no segmentado
 - 2.2.2. Procesador segmentado
 - 2.2.3. Procesador superescalar
 - 2.2.4. Procesador supersegmentado
3. Evaluación de prestaciones
 - 3.1. Medidas de rendimiento
 - 3.1.1. Tiempo de CPU
 - 3.1.2. MIPS
 - 3.1.3. FLOPS
 - 3.2. Benchmarck
 - 3.3. Ley de Amdahl

Bibliografía

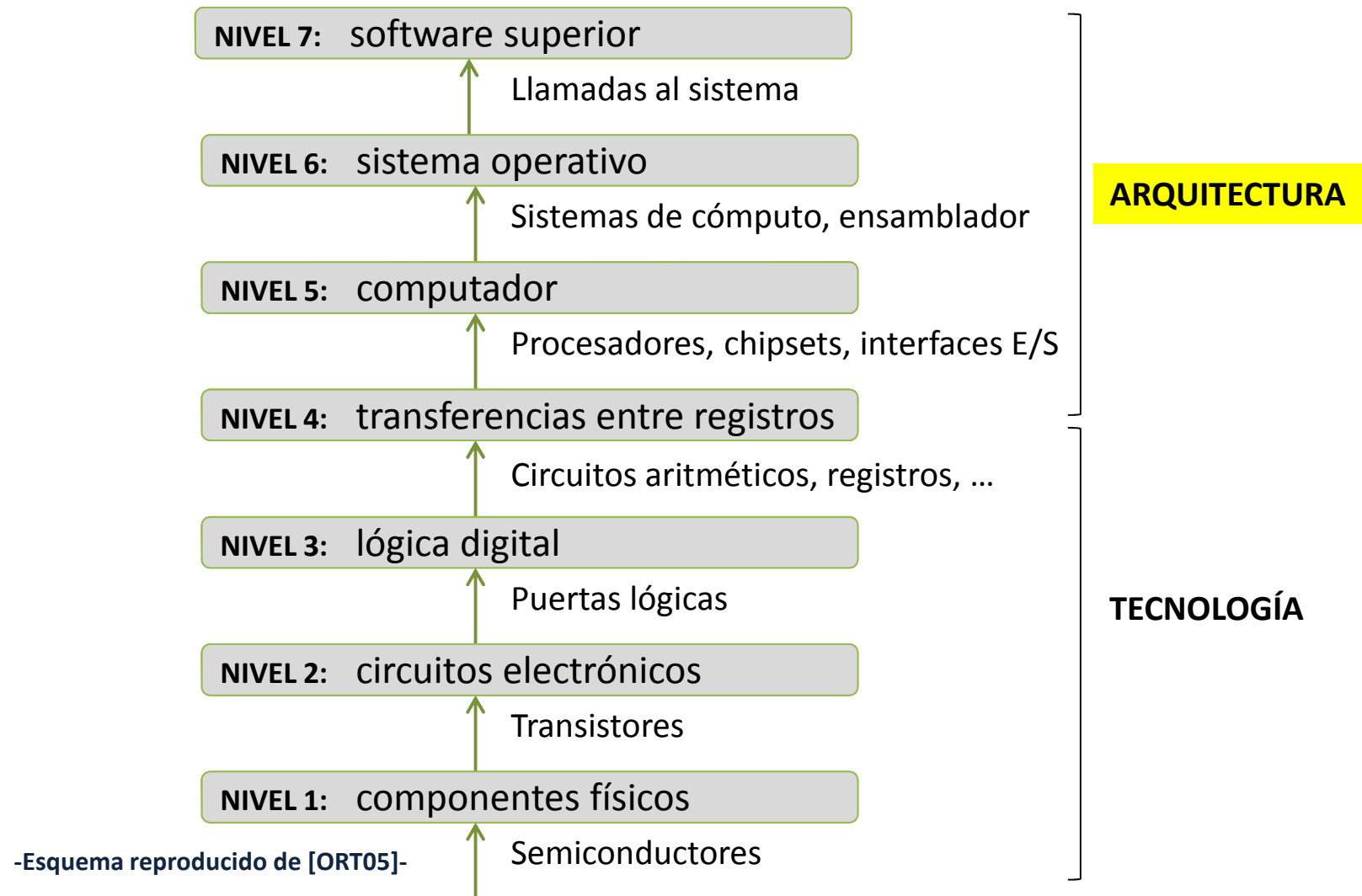
Estructura y diseño de computadores: Capítulo 1.

Patterson y Hennessy. Editorial Reverte, 2011.

1. Introducción a la arquitectura de computadores

1. Introducción a la arquitectura de computadores
 - 1.1. Niveles de descripción de un computador
 - 1.2. Ámbito de estudio
 - 1.3. Objeto de estudio
2. Clasificación de arquitecturas
 - 2.1. Taxonomía de Flynn
 - 2.1.1. Computadores SISD
 - 2.1.2. Computadores SIMD
 - 2.1.3. Computadores MIMD
 - 2.1.3.1. Tipos de computadores MIMD
 - 2.1.4. Computadores MISD
 - 2.2. Paralelismo a nivel de instrucción (ILP)
 - 2.2.1. Procesador no segmentado
 - 2.2.2. Procesador segmentado
 - 2.2.3. Procesador superescalar
 - 2.2.4. Procesador supersegmentado
3. Evaluación de prestaciones
 - 3.1. Medidas de rendimiento
 - 3.1.1. Tiempo de CPU
 - 3.1.2. MIPS
 - 3.1.3. FLOPS
 - 3.2. Benchmarck
 - 3.3. Ley de Amdahl

1.1. Niveles de descripción de un computador



1.1. Niveles de descripción de un computador

Una descripción del computador en niveles permite abordar su estudio de forma **estructurada y ordenada**.

Interpretación del esquema: por ejemplo, el nivel 1 utiliza y estudia materiales como los semiconductores para construir componentes físicos como los transistores. Por tanto, este nivel proporciona transistores al nivel superior.

La implementación de cada nivel debe perseguir la **máxima eficiencia**. Hay que considerar aspectos como coste-prestaciones, consumo de potencia, tolerancia a fallos, tamaño, etc.



Sobre el nivel 7: en el desarrollo de **compiladores** (especialmente si se tratan de compiladores para computadores paralelos) hay que tener muy en cuenta la arquitectura del computador para aprovechar todas sus características.

1.1. Ámbito de estudio



Elementos del computador



Lenguaje máquina del computador, microarquitectura del procesador



Ensamblador



Procedimientos para evaluar los computadores



Tendencias en su evolución

1.2. Objeto de estudio

Diseñar

- sistemas que alcancen los requisitos establecidos por el mercado (precio, consumo y prestaciones).

Evaluar

- las características del computador para identificar posibles cuellos de botella.

Aprovechar

- las prestaciones del computador para escribir aplicaciones, compiladores y sistemas operativos eficaces.

2. Clasificación de arquitecturas

1. Introducción a la arquitectura de computadores
 - 1.1. Niveles de descripción de un computador
 - 1.2. Ámbito de estudio
 - 1.3. Objeto de estudio
2. Clasificación de arquitecturas
 - 2.1. Taxonomía de Flynn
 - 2.1.1. Computadores SISD
 - 2.1.2. Computadores SIMD
 - 2.1.3. Computadores MIMD
 - 2.1.3.1. Tipos de computadores MIMD
 - 2.1.4. Computadores MISD
 - 2.2. Paralelismo a nivel de instrucción (ILP)
 - 2.2.1. Procesador no segmentado
 - 2.2.2. Procesador segmentado
 - 2.2.3. Procesador superescalar
 - 2.2.4. Procesador supersegmentado
3. Evaluación de prestaciones
 - 3.1. Medidas de rendimiento
 - 3.1.1. Tiempo de CPU
 - 3.1.2. MIPS
 - 3.1.3. FLOPS
 - 3.2. Benchmarck
 - 3.3. Ley de Amdahl

2.1. Taxonomía de Flynn

Es una clasificación de arquitecturas de computadores propuesta por Michael J. Flynn en 1972 (Stanford University).

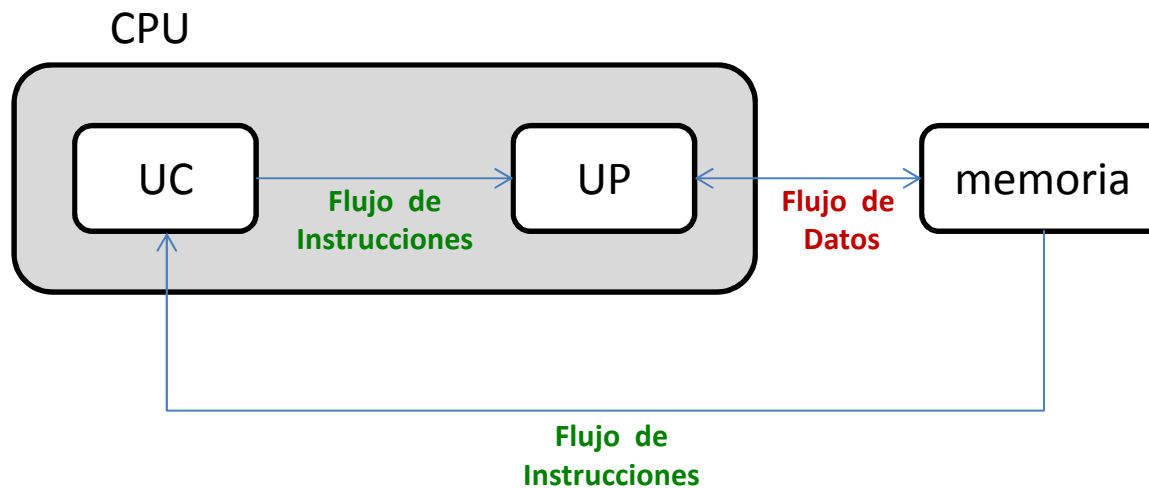
Divide los computadores en 4 clases según el número de flujos de instrucciones y flujos de datos que pueden procesarse simultáneamente.

		Flujo de Instrucciones	
		Single	Múltiple
Flujo de Datos	Single	SISD	MISD
	Multiple	SIMD	MIMD

2.1.1. Computadores SISD

Características de un computador SISD:

- Sólo tiene un procesador (CPU).
- El procesador sólo tiene una unidad de control (UC).
- El procesador sólo tiene **una** unidad de procesamiento (UP).

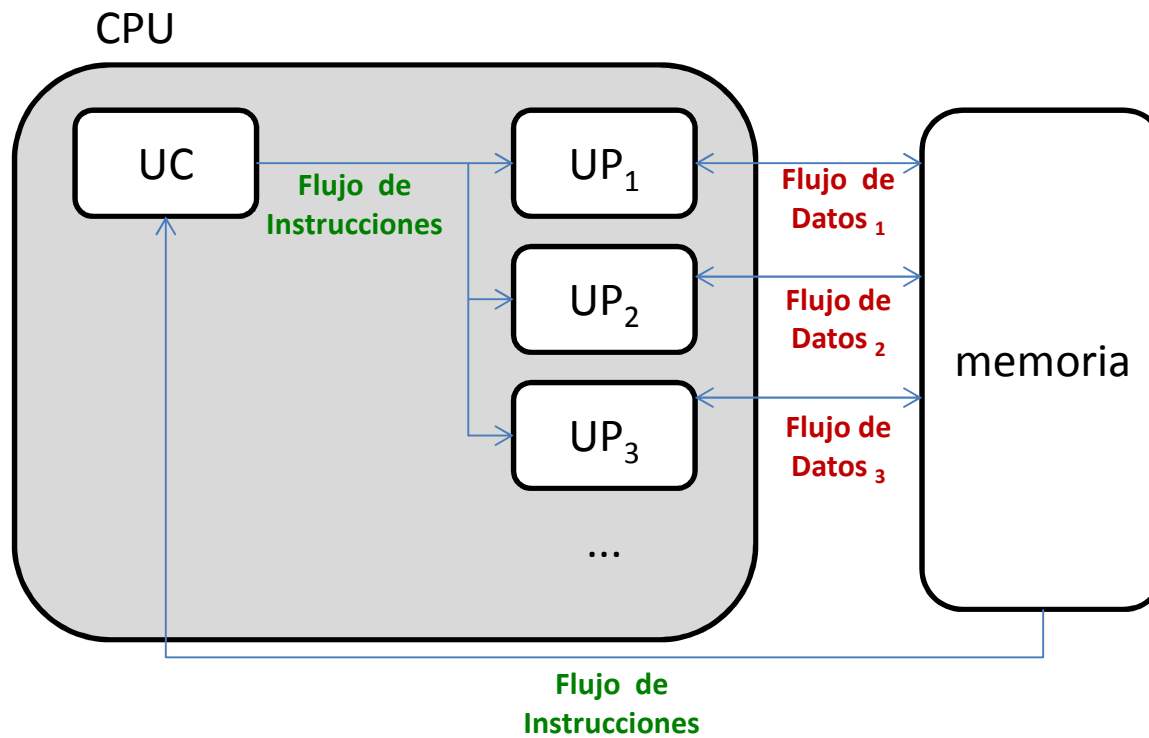


¿Podrías dar un ejemplo de computador SISD?

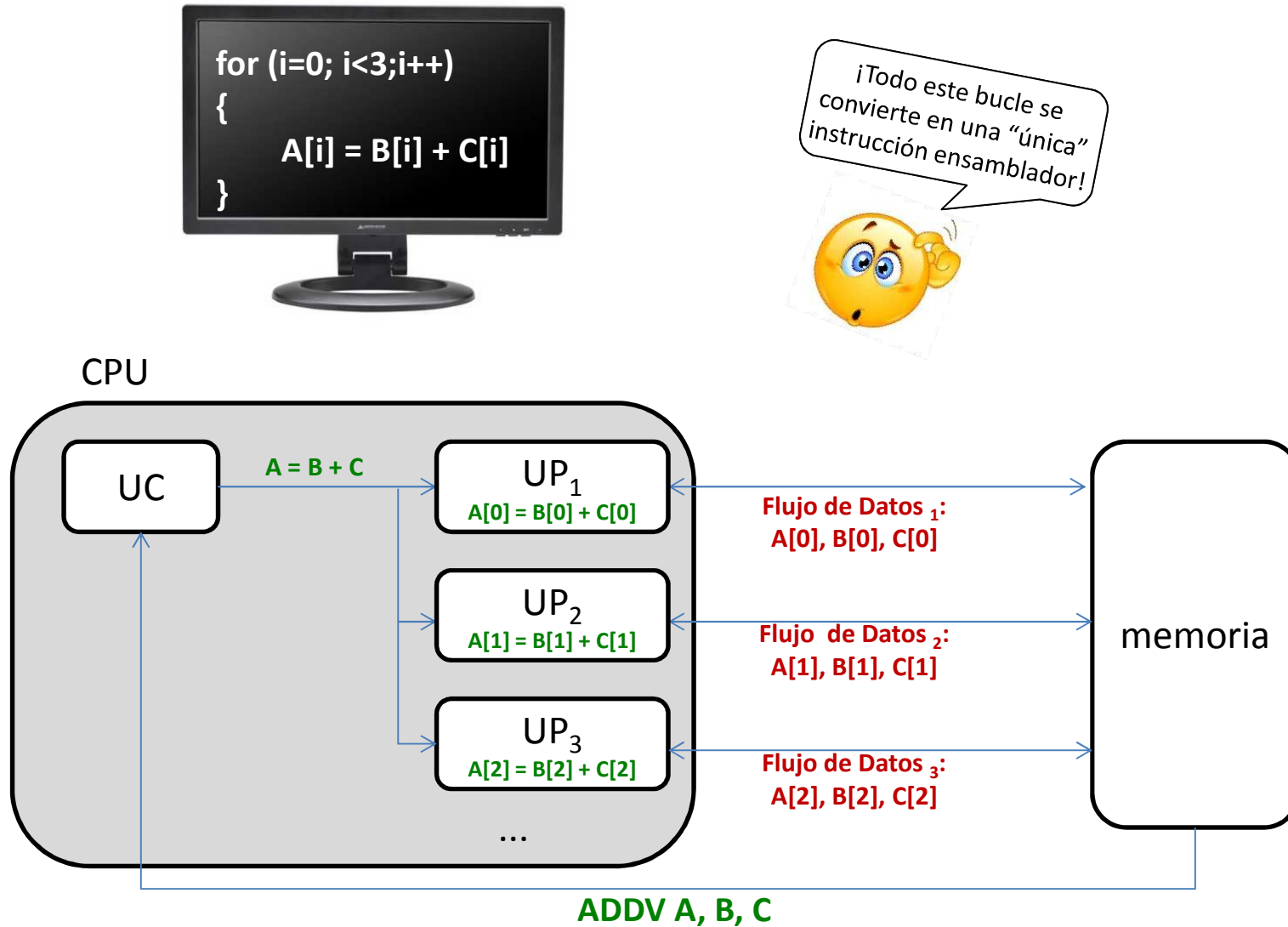
2.1.2. Computadores SIMD

Características de un computador SIMD:

- Sólo tiene un procesador (CPU).
- El procesador sólo tiene una unidad de control (UC).
- El procesador tiene **múltiples** unidades de procesamiento (UP).



2.1.2. Computadores SIMD



2.1.2. Computadores SIMD

Un procesador SIMD es capaz de ejecutar la misma instrucción sobre muchos datos de forma simultánea.

Para ello, el procesador dispone de varias unidades de procesamiento (UP) y de un conjunto de **instrucciones vectoriales**. Por ejemplo, `addv A,B,C` suma los elementos del vector B con los elementos del vector C.

Una operación vectorial “equivale” a un **bucle completo** que procesa los N elementos del vector.



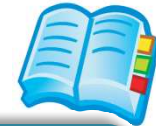
Los **compiladores** para procesadores vectoriales analizan si las instrucciones situadas dentro de los bucles pueden ser ejecutadas en paralelo y genera código objeto con instrucciones vectoriales. Cuanto mejor vectorice el código, mejor será el rendimiento.

2.1.2. Computadores SIMD

Tipos de computadores SIMD:

- **Procesadores de arrays** (ej. el antiguo Cray-1 año 1976).
- **Procesadores con extensiones SIMD**, también denominadas extensiones multimedia (ej. algunos Pentium, núcleos del Corei7, etc).
- **Procesadores vectoriales** (ej. Procesador Cell año 2005).

Cuestión 1



Estudia las extensiones SIMD de los procesadores x86 y cómo se implementan.

Recomendable utilizar el Patterson para trabajar el último punto.

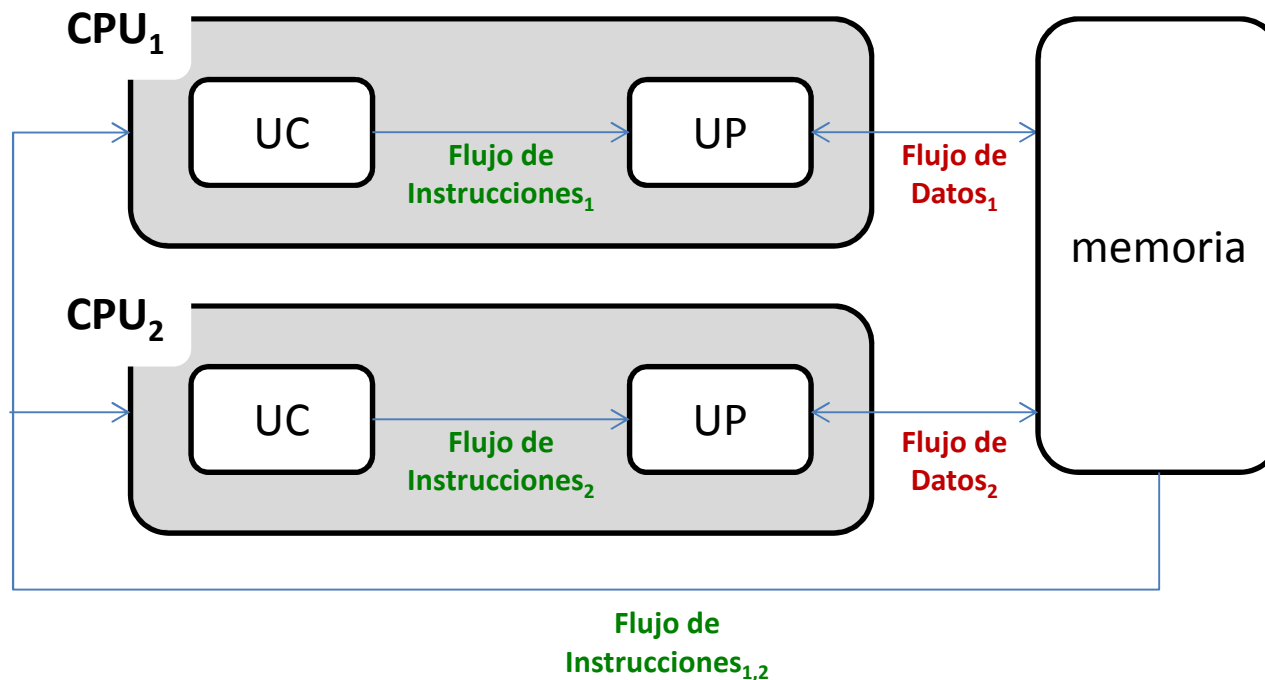
2.1.3. Computadores MIMD

Características de un computador MIMD:

- Tiene **varios** procesadores (CPU).
- Cada procesador tiene su unidad de control (UC) y su unidad de procesamiento (UP).



¿Podrías dar un ejemplo de computador MIMD?

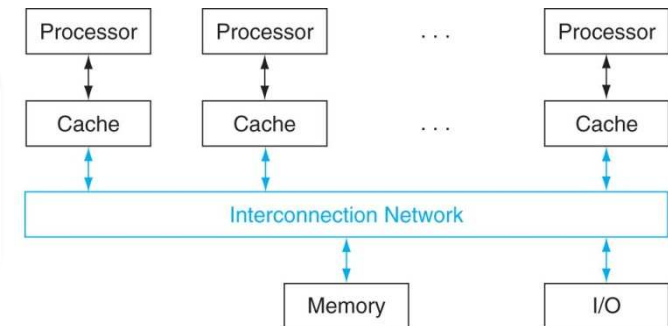


2.1.3.1. Tipos de computadores MIMD

1

Memoria compartida (multiprocesadores)

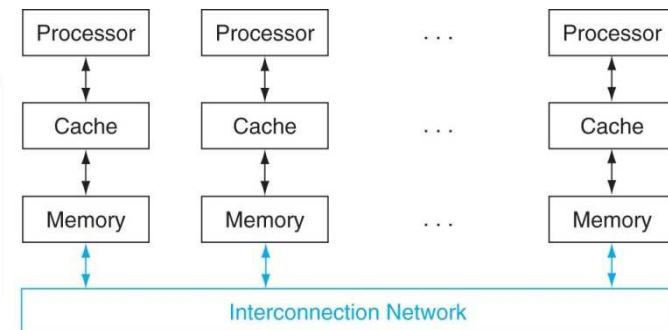
- Existe un único espacio de direcciones que es compartido por todos los procesadores.
- Normalmente sistema formado por una única computadora.
- Ejemplo: procesador multinúcleo



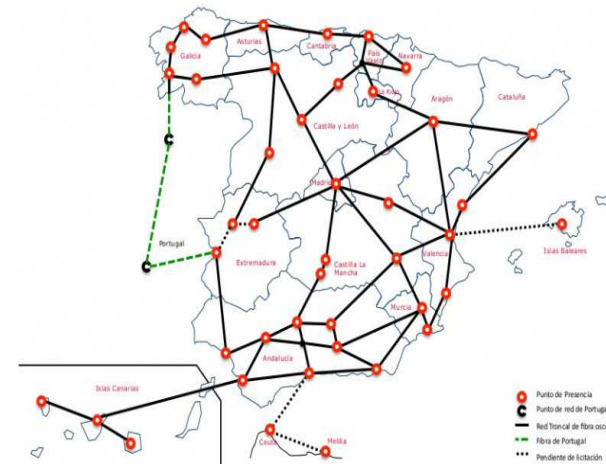
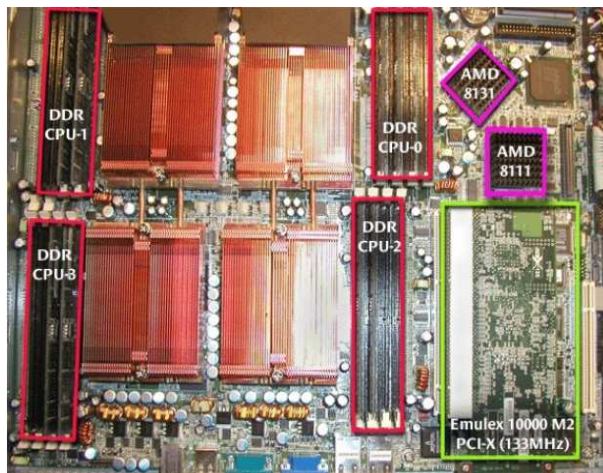
2

Memoria distribuida (multicomputadores)

- Cada procesador tiene su propio espacio de direcciones.
- Conjunto de computadoras conectadas entre sí por una red de comunicaciones. Percibido por el usuario como un solo sistema.
- Ejemplos: cluster, grid.



2.1.3.1. Tipos de computadores MIMD



2.1.3.1. Tipos de computadores MIMD



El Supercomputador de la UCA “CAI 2”

<http://supercomputacion.uca.es/>

Cuestión 2



1. ¿Qué es una supercomputadora? Explícalo en detalle.
2. En 2015, ¿cuál fue la mejor supercomputadora?
3. En la lista TOP500, ¿hay alguna española?

<http://en.wikipedia.org/wiki/Supercomputer>

<http://es.wikipedia.org/wiki/TOP500>

<http://www.top500.org/>

2.1.3.1. Tipos de computadores MIMD

Documentación adicional

Grid

Concepto:

http://es.wikipedia.org/wiki/Computaci%C3%B3n_grid

http://en.wikipedia.org/wiki/Grid_computing

<http://www.gridcomputing.com/>

Ejemplo: búsqueda de inteligencia extraterrestre

<http://es.wikipedia.org/wiki/SETI@home>

Ejemplo: irisgrid

<http://www.irisgrid.es/>

<http://www.irisgrid.es/doc/Folleto.pdf>

Ejemplo: Red Española de Supercomputación

http://es.wikipedia.org/wiki/Red_Espa%C3%B1ola_de_Supercomputaci%C3%B3n

<http://www.iac.es/servicios.php?op1=30>

2.1.4. Computadores MISD

- Constituyen una clase de computadores cuyo comportamiento se puede implementar con una arquitectura MIMD.
- Por el motivo anterior, no existen computadores MISD específicos.
- USO: Un ejemplo sería un conjunto de equipos que trata de factorizar un número primo muy grande utilizando diferentes algoritmos.

2.2. Paralelismo a nivel de instrucción (ILP)

IDEA: Si la ejecución de una instrucción la dividimos (segmentamos) en varias etapas y conseguimos que cada etapa se lleve a cabo en una sección concreta del procesador sin interferir con las demás secciones,

¿Qué podríamos conseguir?

2.2. Paralelismo a nivel de instrucción (ILP)

E T A P A S

IF (Instruction Fetch): captar de memoria la instrucción a ejecutar.

Camino a
memoria

ID (Instruction Decode): decodificar la instrucción.

U.C.

EX (Execution): ejecución de la instrucción (calcular una dirección de memoria, realizar una operación aritmético/lógica con los operandos o calcular la dirección de un salto).

A.L.U.

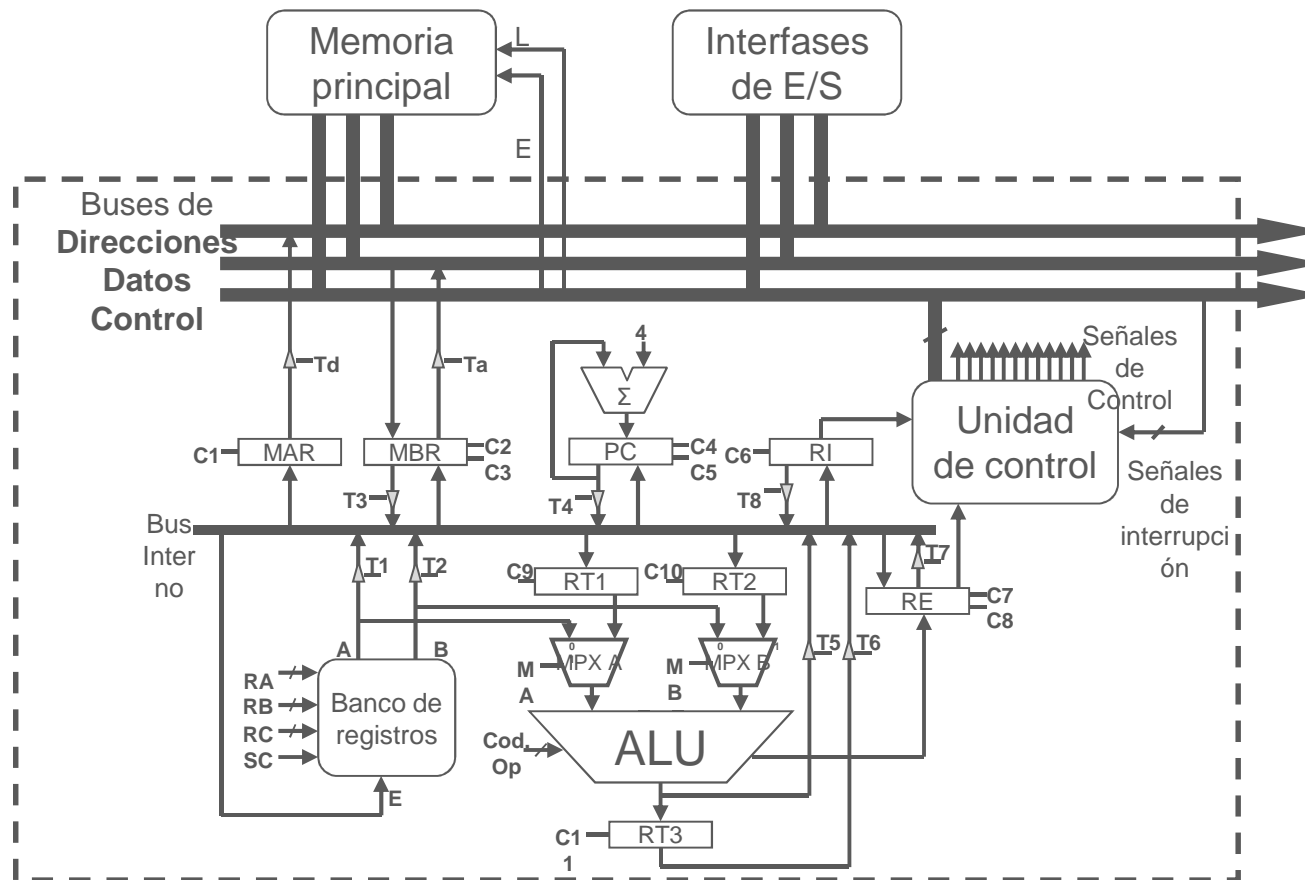
MEM (Memory): acceso a memoria para leer/escribir un dato (esta etapa sólo tiene lugar en instrucciones de carga/almacenamiento).

Camino a
memoria

WB (WriteBack): escribir el resultado en los registros.

Camino a
registros

2.2. Paralelismo a nivel de instrucción (ILP)



¿Para conseguir ILP hay que duplicar elementos en el procesador?



2.2. Paralelismo a nivel de instrucción (ILP)

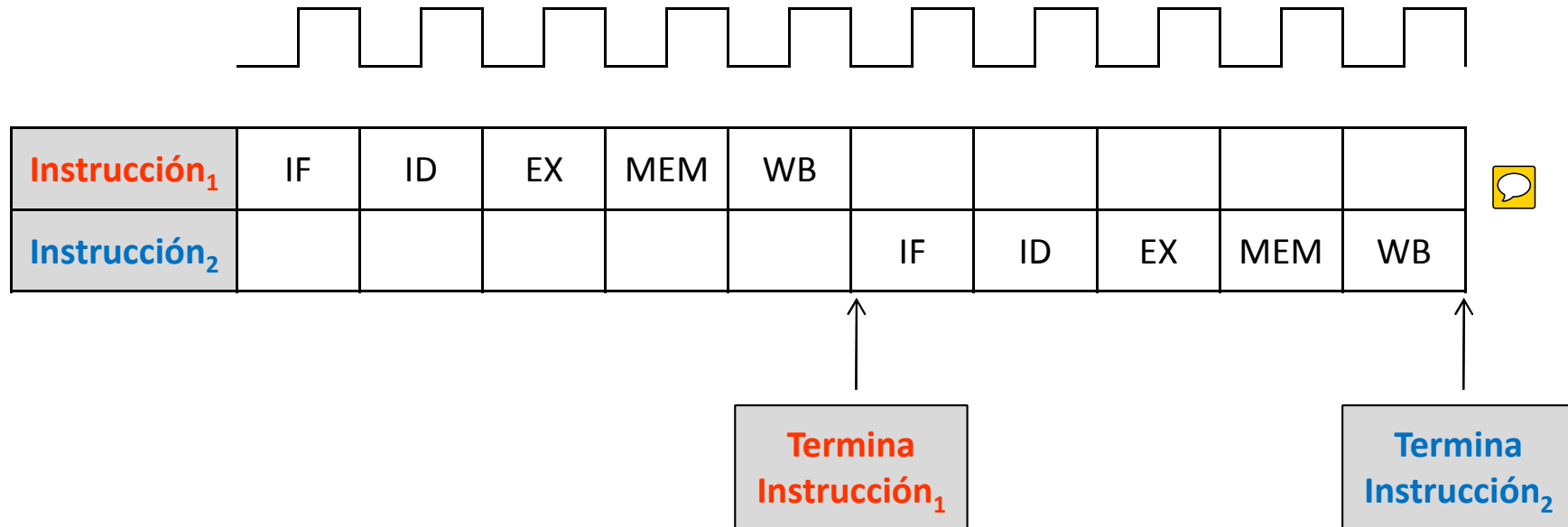
Para conseguir el paralelismo a nivel de instrucción hay que implementar en el procesador un camino de datos segmentado (*pipeline*).

El camino de datos se segmenta en “zonas”. Cada zona se dedica a ejecutar una etapa concreta de la instrucción.



Instrucción 1
Instrucción 2
Instrucción 3
Instrucción 4
Instrucción 5
Instrucción 6
...

2.2.1. Procesador no segmentado

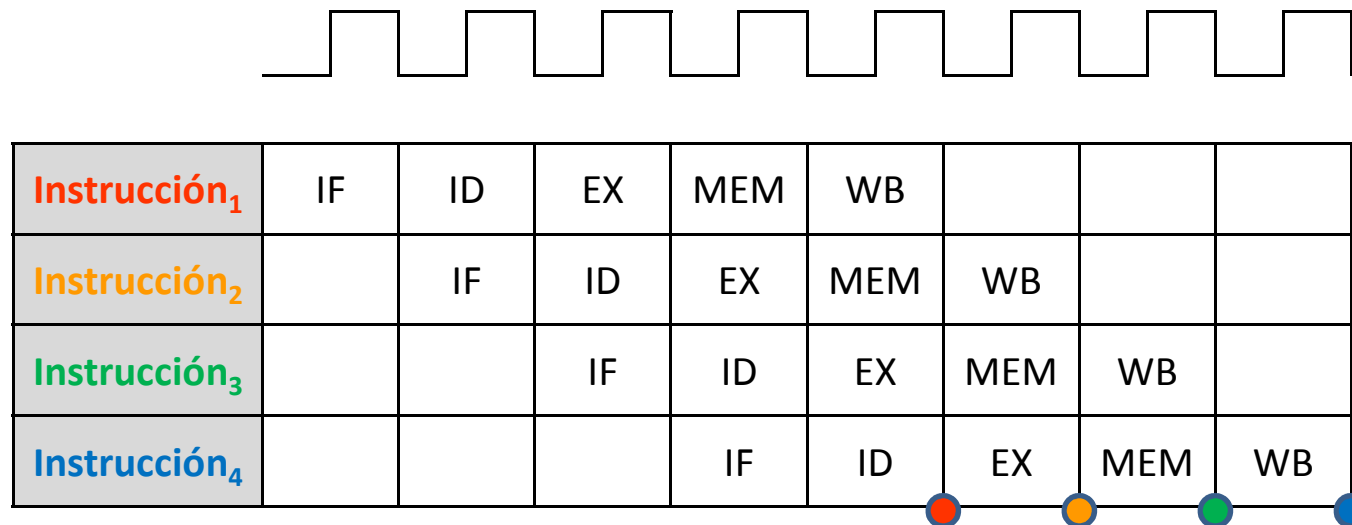


¿Cada cuántos ciclos se termina de ejecutar una instrucción?

¿Cuántas instrucciones se ejecutan en un ciclo?



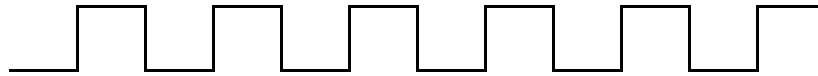
2.2.2. Procesador segmentado *-pipeline-*



Una vez que se ha llenado el pipeline, ¿Cuántas instrucciones se ejecutan en un ciclo?

Termina Instrucción₁

2.2.3. Procesador superescalar



Instrucción ₁	IF	ID	EX	MEM	WB	
Instrucción ₂	IF	ID	EX	MEM	WB	
Instrucción ₃		IF	ID	EX	MEM	WB
Instrucción ₄		IF	ID	EX	MEM	WB

- En cada etapa se ejecuta más de una instrucción. El número máximo de instrucciones que se pueden ejecutar en una etapa se denomina **grado**.
- Para conseguir esta concurrencia hay que replicar elementos del procesador.
- En cada ciclo de reloj se inicia la ejecución de varias instrucciones.



Una vez que se ha llenado el pipeline,
¿Cuántas instrucciones se ejecutan en un ciclo?
En este ejemplo,
¿De qué grado es el procesador superescalar?

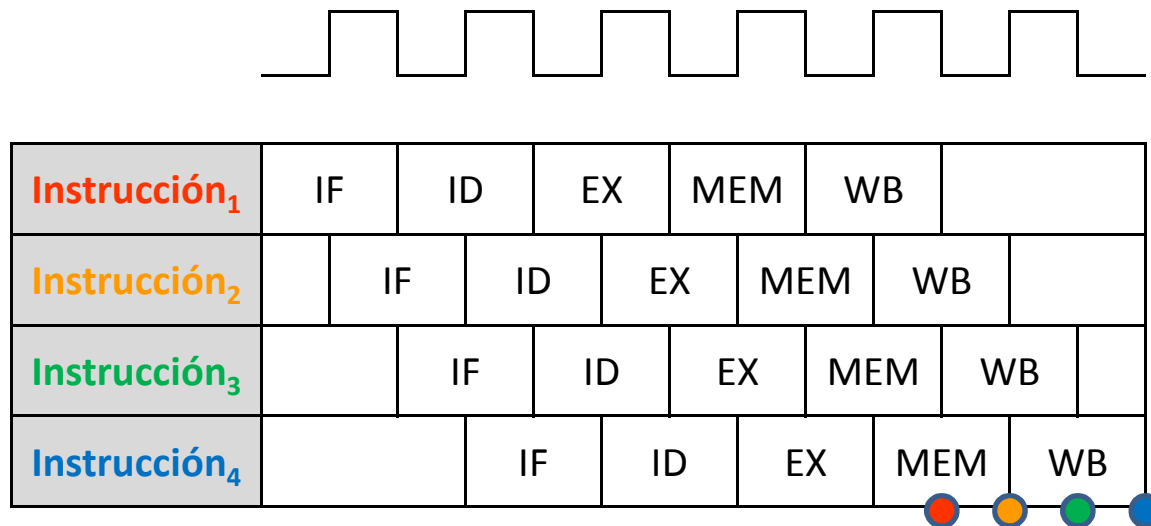


Cuestión 3

1. ¿Qué es un procesador VLIW?
2. Diferencias con un procesador superescalar
3. Ejemplos actuales

<http://es.wikipedia.org/wiki/VLIW>
<http://es.wikipedia.org/wiki/Superescalar>
Necesario consultar el Patterson

2.2.4. Procesador supersegmentado



- Cada etapa se segmenta en varias subetapas, dando como resultado un pipeline de **numerosas etapas** (algunos procesadores tienen hasta 31 etapas como el Pentium 4 Prescott, año 2004).

- Como consecuencia, se pueden ejecutar en paralelo más instrucciones de lo habitual sin tener que replicar elementos del procesador.



Una vez que se ha llenado el pipeline, ¿Cuántas instrucciones se ejecutan en un ciclo?
En este ejemplo, ¿En cuántas subetapas se divide una etapa?


2.3. Paralelismo a nivel de Instrucción (ILP) vs. Paralelismo a nivel de Procesos (PLP)



¿Qué diferencias existen? Inicia un Debate

Cuestión 4



1. ¿Qué es el Hyperthreading? 
2. ¿Dónde se encuadra? ¿en ILP? o ¿en PLP?
3. Ejemplos actuales

Recomendable utilizar el Patterson para trabajar los puntos 1 y 2.

3. Evaluación de prestaciones

1. Introducción a la arquitectura de computadores
 - 1.1. Niveles de descripción de un computador
 - 1.2. Ámbito de estudio
 - 1.3. Objeto de estudio
2. Clasificación de arquitecturas
 - 2.1. Taxonomía de Flynn
 - 2.1.1. Computadores SISD
 - 2.1.2. Computadores SIMD
 - 2.1.3. Computadores MIMD
 - 2.1.3.1. Tipos de computadores MIMD
 - 2.1.4. Computadores MISD
 - 2.2. Paralelismo a nivel de instrucción (ILP)
 - 2.2.1. Procesador no segmentado
 - 2.2.2. Procesador segmentado
 - 2.2.3. Procesador superescalar
 - 2.2.4. Procesador supersegmentado
3. Evaluación de prestaciones
 - 3.1. Medidas de rendimiento
 - 3.1.1. Tiempo de CPU
 - 3.1.2. MIPS
 - 3.1.3. FLOPS
 - 3.2. Benchmarck
 - 3.3. Ley de Amdahl

3.1.1. Tiempo de CPU

$$T_{\text{CPU}} = NI * CPI_{\text{global}} * T = \frac{NI * CPI_{\text{global}}}{F}$$

Es el tiempo que tarda la CPU en ejecutar un programa determinado.

NI es el número total de instrucciones máquina que ejecuta el programa.

CPI_{global} es el número medio de ciclos en los que se ejecuta una instrucción máquina del programa.

$$CPI_{\text{global}} = \frac{\sum_{i=1}^n NI_i * CPI_i}{NI}$$

donde *i* identifica los distintos tipos de instrucción.

T es el periodo del reloj del procesador, es decir, el tiempo que dura un ciclo. También se puede ver como el inverso de la frecuencia (**F**).

3.1.1. Tiempo de CPU

$$T_{\text{CPU}} = \text{CPP} * T = \frac{\text{CPP}}{F}$$

CPP es el número de ciclos que tarda en ejecutarse un programa.

$$\text{CPP} = \sum_{i=1}^n NI_i * CPI_i$$

donde i identifica los distintos tipos de instrucción.

T es el periodo del reloj del procesador, es decir, el tiempo que dura un ciclo. También se puede ver como el inverso de la frecuencia (**F**).

3.1.1. Tiempo de CPU

Prestaciones: relaciona las prestaciones y el tiempo de ejecución de un computador

$$\text{Prestaciones} = \frac{1}{T_{\text{CPU}}}$$

Relación de prestaciones: relaciona las prestaciones de dos máquinas diferentes

$$\frac{\text{Prestaciones}_A}{\text{Prestaciones}_B} = \frac{T_{\text{CPU } B}}{T_{\text{CPU } A}} = n$$

← “La máquina A
es **n veces más rápida**
que la máquina B”



¿Cómo será el CPI de un procesador segmentado?

¿Cómo será el CPI de un procesador superescalar?

3.1.1. Tiempo de CPU

Problema 1

Un mismo programa se ejecuta en dos computadores diferentes,

1.- ¿Qué CPU ejecuta el programa más rápido?

2.- ¿Cuánto más rápida es?

	T	CPI
CPU_A	250 ps	2
CPU_B	500 ps	1,2

Se supone que el programa ejecuta el mismo número de instrucciones máquina en las dos CPUs.

-Problema obtenido de [PATT11]-

3.1.1. Tiempo de CPU

Problema 2

Un mismo programa se ejecuta en dos máquinas diferentes. Sabemos que la máquina B necesita 1,2 veces más ciclos de reloj que la máquina A para ejecutarlo.

1.- ¿Qué frecuencia de reloj tendría que tener la máquina B?

	F	T_{CPU}
CPU_A	2 GHz	10 s
CPU_B	?	6 s

-Problema obtenido de [PATT11]-

3.1.1. Tiempo de CPU

Problema 3

Un diseñador de compiladores tiene que decidir entre dos secuencias de código para una máquina en particular. Esta máquina tiene tres tipos de instrucciones máquina diferentes: A, B, C. Las secuencias requieren el siguiente número de instrucciones:

	A	B	C
Secuencia ₁	2	1	2
Secuencia ₂	4	1	1

Por otro lado, el CPI para cada tipo de instrucción es el siguiente:

	A	B	C
CPI	1	2	3

- 1.- ¿Qué secuencia de código ejecuta mayor número de instrucciones?
- 2.- ¿Qué secuencia de código es más rápida?
- 3.- ¿Cuál es el CPI_{MEDIO} para cada secuencia?

-Problema obtenido de [PATT11]-

3.1.2. MIPS

MIPS = MILLION INSTRUCTIONS PER SECOND

$$\text{MIPS} = \frac{NI}{T_{\text{CPU}} * 10^6} = \frac{F}{\text{CPI}_{\text{global}} * 10^6}$$

INCONVENIENTE:

Se desea contrastar un programa escrito en lenguaje de alto nivel en dos computadoras de distinta arquitectura, por ejemplo, una es MIPS y otra es x86.

En cada computadora se genera un ejecutable con código máquina muy distinto, específico de cada arquitectura. Decimos que es muy distinto en cuanto a número de instrucciones y número de ciclos que requiere cada instrucción para ejecutarse.

Con la medida MIPS, ¿¿¿siempre obtendremos resultados coherentes??? ...debate...

Por tanto, la medida MIPS sólo es útil para comparar procesadores con el mismo juego de instrucciones.



Cuestión 5

1. CISC/RISC.
2. Ejemplos actuales.

<http://es.wikipedia.org/wiki/RISC>

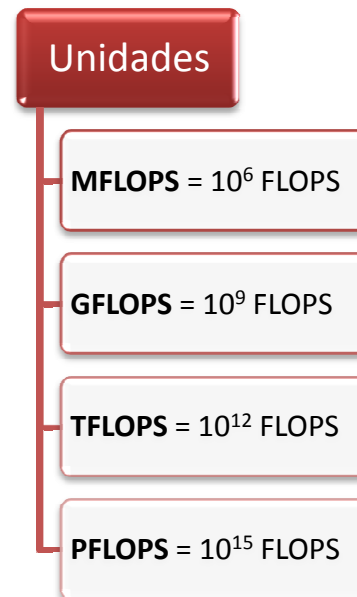
<http://es.wikipedia.org/wiki/CISC>

3.1.3. FLOPS

FLOPS = FLOATING POINT OPERATIONS PER SECOND

IDEA: Un programa en computadoras diferentes puede ejecutar un número distinto de instrucciones máquina pero el mismo número de operaciones en punto flotante.

- 1 Sólo es aplicable a operaciones en punto flotante.
- 2 La precisión en punto flotante no es la misma en todas las máquinas.



3.2. Benchmark *-Comparativa-*

Herramientas software utilizadas para medir el rendimiento general del computador o de un componente concreto (CPU, RAM, tarjeta gráfica, etc).

1

SPEC (Standard Performance Evaluation Corporation)

Muy utilizado para evaluar computadoras personales y servidores.

<http://www.spec.org/benchmarks.html>

2

Linpack

Muy utilizado para evaluar supercomputadoras (TOP500).

<http://www.top500.org/project/linpack>

3.3. Ley de Amdahl

Mide el incremento de rendimiento del computador cuando uno de sus componentes ha sido mejorado.

$$\text{Incremento de rendimiento} = \frac{1}{(1-T) + \frac{T}{M}}$$

T es el porcentaje de tiempo que el computador utiliza el componente que se ha mejorado

M es el factor de mejora



¿Qué componente será más interesante mejorar?

3.3. Ley de Amdahl

Problema 4

En un computador añadimos una caché 5 veces más rápida. La caché es utilizada el 90% del tiempo.

¿En qué factor ha mejorado el rendimiento?