

Tema 1. Introducción

Arquitectura de los Computadores

Tema1. Introducción

Arquitectura

Diseño

Introducción

- 1. **Arquitectura de computadores**
- 2. **El diseño de computadores**

Tema1. Introducción

Arquitectura

Diseño

Introducción

1.Arquitectura de computadores

- ◆ 1.1 Niveles de descripción de un computador
- ◆ 1.2 Definición de arquitectura
- ◆ 1.3 Clasificación de arquitecturas

• Concepto de arquitect.
• Niveles de abstracción

→ Taxonomía de Flynn

2.El diseño de computadores

- ◆ 2.1 El proceso de diseño
 - ◆ a) Establecer requerimientos funcionales
 - ◆ b) Decisiones de implementación
 - ◆ c) Consideración de las nuevas tendencias
- ◆ 2.2 Principios de diseño
 - ◆ a) Acelerar el caso común
 - ◆ b) Ley de rendimientos decrecientes
 - ◆ c) Localidad de referencia

1.1 Niveles de descripción de un computador

Arquitectura

Diseño

Introducción

El concepto de arquitectura

- ❖ **Acuñado por IBM en 1964.** Parte del repertorio de instrucciones visible por el programador:

[Amdahl, 64]. Presentación del IBM S/360: *La arquitectura de un computador es la estructura del computador que un programador en lenguaje máquina debe conocer para escribir un programa correcto*

- ❖ **Unificar** las diferentes **divisiones de IBM** en una **arquitectura común**
- ❖ **Referencia exclusivamente** al diseño del **repertorio de instrucciones**
- ❖ **Errores de diseño** por falta de consideración de aspectos de implementación

1.1 Niveles de descripción de un computador

Arquitectura

Diseño

Introducción

El concepto de arquitectura

- 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** → Tecnología (bajo nivel)

1.1 Niveles de descripción de un computador

Arquitectura

Diseño

Introducción

Niveles estructurales de Bell y Newell [Bell, 71]:

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

1.1 Niveles de descripción de un computador

Arquitectura

Diseño

Introducción

Niveles de Interpretación de Levy [Bell 78, De Miguel 01]:

- 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 86, 99, 00].

1.1 Niveles de descripción de un computador

Arquitectura

Diseño

Introducción

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

Niv. Soft. Superiores

Llamadas al Sistema, Comandos,...

Sistema Operativo

Sistemas de cómputo, Ensamblador,...

Sist. Computador

Procesadores, Interfaces E/S, Datapath

Nivel RT

ALUs, registros, memorias, MUX,...

Digital

Puertas lógicas, inversores, biestables

Electrónica

Uniones P N, Metal, Polisilicio

Componente

1.1 Niveles de descripción de un computador

Arquitectura

Diseño

Introducción

Niveles de abstracción para un computador:

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.

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.

SW

ARQUITECTURA

TECNOLOGÍA

HW

Niv. Soft. Superiores

Llamadas al Sistema, Comandos,...

Sistema Operativo

Sistemas de cómputo, Ensamblador,...

Sist. Computador

Procesadores, Interfaces E/S, Datapath

Nivel RT

ALUs, registros, memorias, MUX,...

Digital

Puertas lógicas, inversores, biestables

Electrónica

Uniones P N, Metal, Polisilicio

Componente

1.1 Niveles de descripción de un computador

Niveles de abstracción para un computador:

Arquitectura

Diseño

Lógica digital (F.C)

- 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.

Introducción

SW

ARQUITECTURA

TECNOLOGÍA

HW

Niv. Soft. Superiores

Llamadas al Sistema, Comandos,...

Sistema Operativo

Sistemas de cómputo, Ensamblador,...

Sist. Computador

Procesadores, Interfaces E/S, Datapath

Nivel RT

ALUs, registros, memorias, MUX,...

Digital

Puertas lógicas, inversores, biestables

Electrónica

Uniones P N, Metal, Polisilicio

Componente

1.1 Niveles de descripción de un computador

Arquitectura

Diseño

Introducción

Niveles de abstracción para un computador:

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.

Construir CPU's, memorias, etc.

SW

ARQUITECTURA

TECNOLOGÍA

HW

Niv. Soft. Superiores

Llamadas al Sistema, Comandos,...

Sistema Operativo

Sistemas de cómputo, Ensamblador,...

Sist. Computador

Procesadores, Interfaces E/S, Datapath

Nivel RT

ALUs, registros, memorias, MUX,...

Digital

Puertas lógicas, inversores, biestables

Electrónica

Uniones P N, Metal, Polisilicio

Componente

1.1 Niveles de descripción de un computador

Arquitectura

Diseño

Introducción

Niveles de abstracción para un computador:

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).

SW

ARQUITECTURA

TECNOLOGÍA

HW

Niv. Soft. Superiores

Llamadas al Sistema, Comandos,...

Sistema Operativo

Sistemas de cómputo, Ensamblador,...

Sist. Computador

Procesadores, Interfaces E/S, Datapath

Nivel RT

ALUs, registros, memorias, MUX,...

Digital

Puertas lógicas, inversores, biestables

Electrónica

Uniones P N, Metal, Polisilicio

Componente

1.1 Niveles de descripción de un computador

Niveles de abstracción para un computador:

Arquitectura

Diseño

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.

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.

Introducción

SW

ARQUITECTURA

TECNOLOGÍA

HW

Niv. Soft. Superiores

Llamadas al Sistema, Comandos,...

Sistema Operativo

Sistemas de cómputo, Ensamblador,...

Sist. Computador

Procesadores, Interfaces E/S, Datapath

Nivel RT

ALUs, registros, memorias, MUX,...

Digital

Puertas lógicas, inversores, biestables

Electrónica

Uniones P N, Metal, Polisilicio

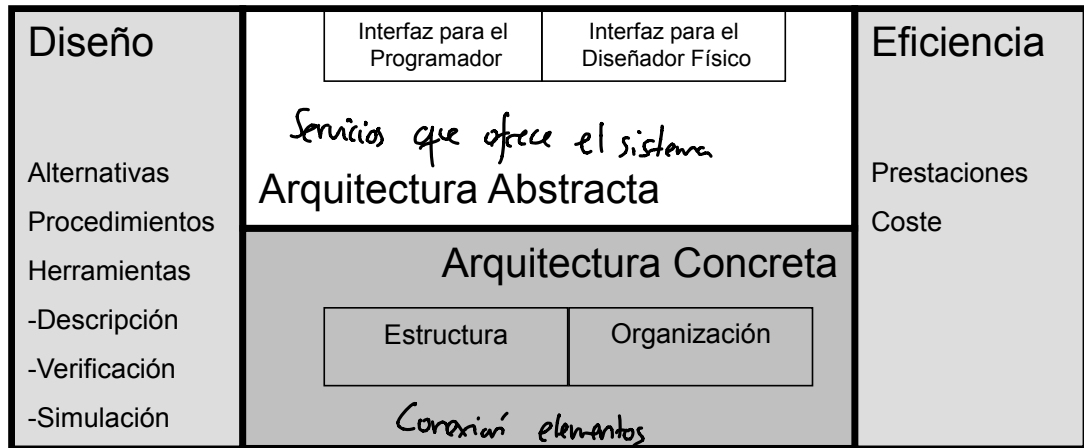
Componente

1.1 Niveles de descripción de un computador

Arquitectura

Diseño

Importante: principios de diseño → Eficiencia
Bajo coste



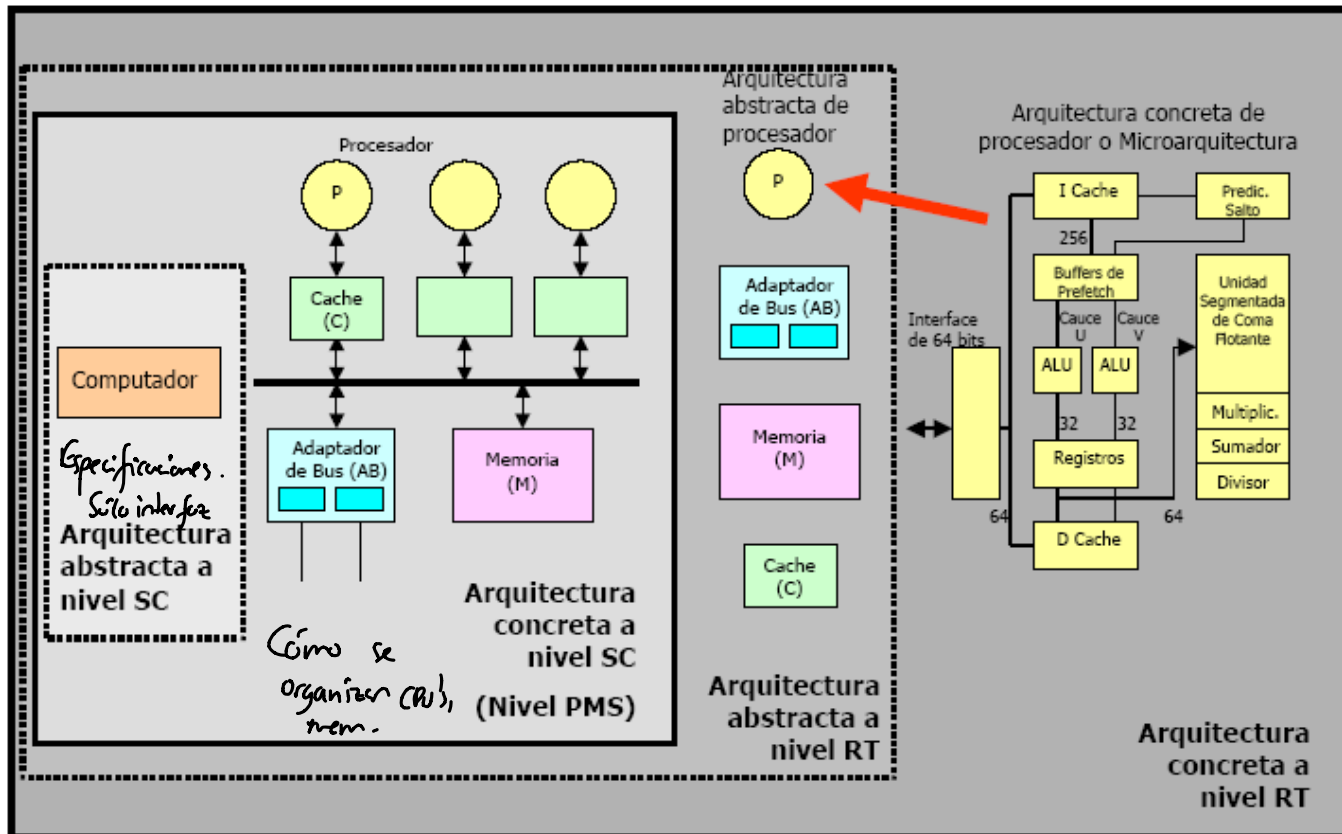
Introducción

1.2 Definición de arquitectura

Arquitectura

Diseño

Niveles que abarca la Arquitectura



Introducción

1.2 Definición de arquitectura

Arquitectura

Diseño

- ❖ **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]

Microarquitectura \longleftrightarrow Arquitectura concreta
(Se usan como sinónimos)

Introducción

1.2 Definición de arquitectura

Arquitectura

Diseño

Introducción

- ❖ **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]

1.2 Definición de arquitectura

Ámbito de la arquitectura

Arquitectura

Diseño

Introducción

Arquitectura a nivel lenguaje máquina

Repertorio de
instrucciones

Organización

alto nivel del diseño de un
computador

Sistema de
memoria, estructura
del Bus, diseño
interno de la CPU...

Hardware

componentes específicos de
una máquina

Diseño lógico
detallado, la
tecnología de
encapsulamiento...

Implementación

1.3 Clasificación de arquitecturas

Arquitectura

Diseño

- 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

(1966)

- La clasificación más extendida "**Taxonomía de Flynn**":

- SISD → Single Instruction, Single Data
- SIMD → " " Multiple Data
- MISD → Multiple " , Single Data
- MIMD → Multiple , Multiple

Clasifica según

- N° instr. concurr.
- N° flujo datos

→ I.A

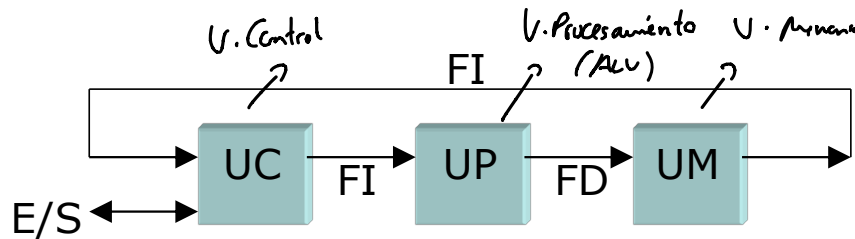
Introducción

1.3 Clasificación de arquitecturas

Arquitectura

Diseño

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;
```

1. Instrucción secuencial con sus propios datos

(Pueden ejecutarse a la vez a pesar de la estructura).

Introducción

1.3 Clasificación de arquitecturas

Arquitectura

Diseño

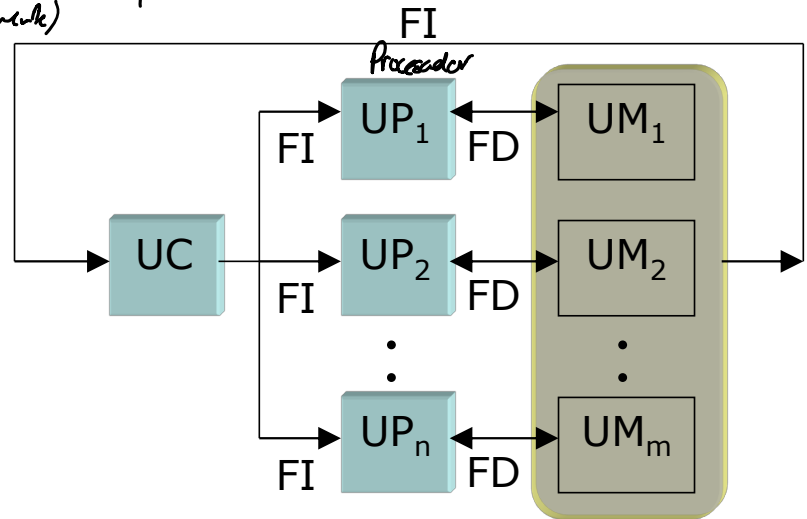
Computadores SIMD: un único flujo de instrucciones procesa operandos y genera resultados, definiendo varios flujos de datos, dado que cada instrucción codifica realmente varias operaciones iguales, cada una actuando sobre operadores distintos.

- Años 70.
- Invest. científica

```
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;
```

(Proc. vectoriales, proc. matriciales.)
para cada UP (simultáneamente)
Cada comp. i
sofoca muchos datos
a la vez.

Procesadores matriciales



indica que son Vectores.

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

- Replicar instrucción en varios flujos de datos → Muchas ALU
- Fuerza actual → Tarjetas gráficas (píxeles, transformación imágenes).

Introducción

1.3 Clasificación de arquitecturas

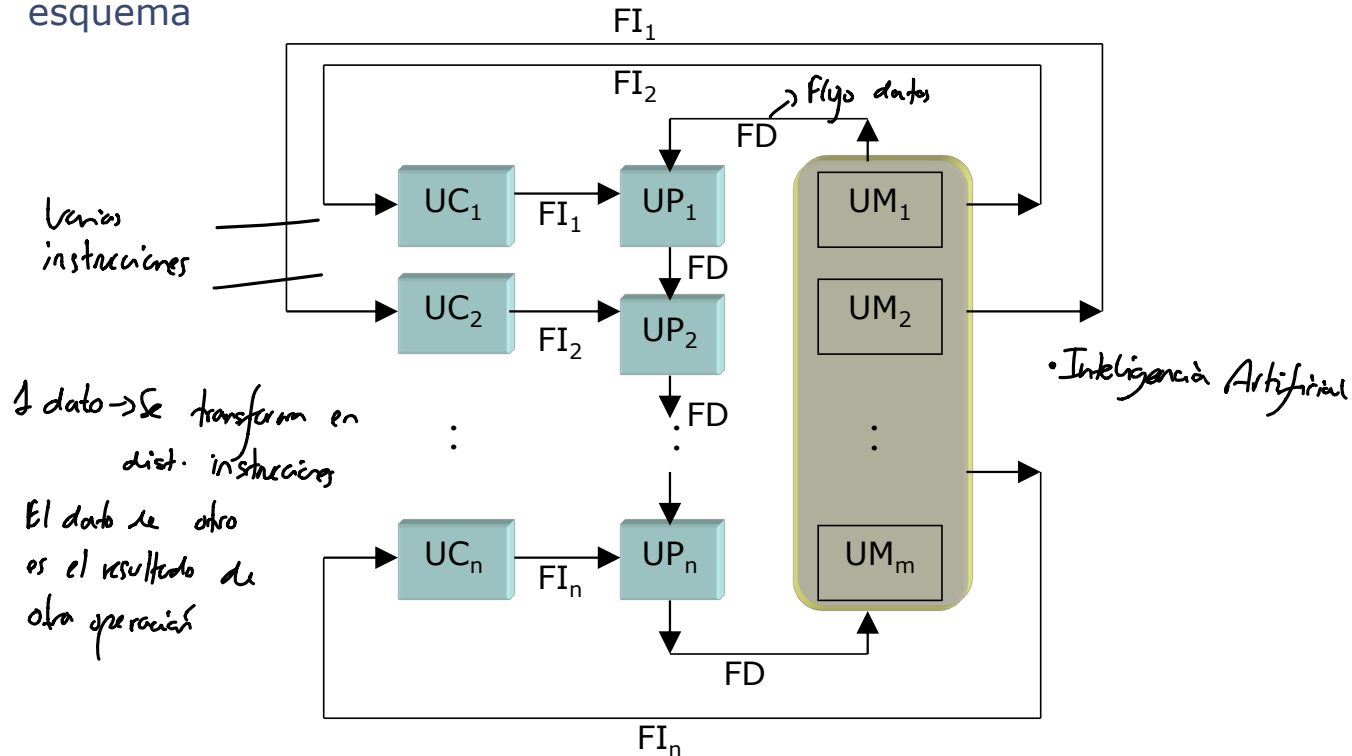
Arquitectura

Diseño

Introducción

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



TPU → . Procesador desarrollado por Google .

Tensor ↙

- Para utilizarlo en IA .
- Producto vector x matriz .

↓

Producto
Tensorial

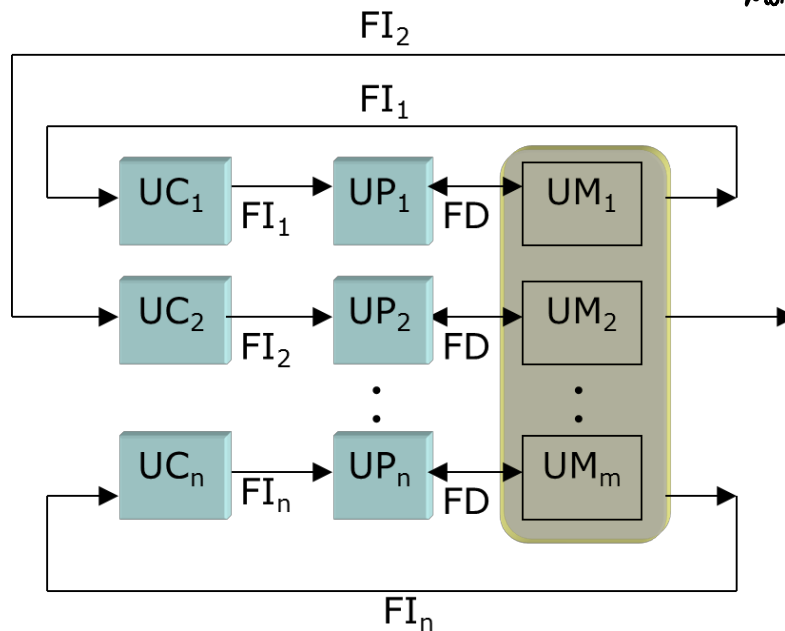
1.3 Clasificación de arquitecturas

Arquitectura

Diseño

Introducción

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.



Multi-core \rightarrow varios proces. . Cada uno sus datos . No tienen por qué tener relación.

```
for i=1 to 4 do
Begin
  C[i] = A[i]+B[i];
End;
Proc1
```

```
for i=1 to 4 do
Begin
  G[i] = K[i]*H[i];
End;
Proc2
```

```
for i=1 to 4 do
Begin
  F[i] = D[i]-E[i];
End;
Proc3
```


1.3 Clasificación de arquitecturas

Arquitectura

Diseño

Introducción

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

1.3 Clasificación de arquitecturas

Arquitectura

Diseño

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):

Granularidad → Nivel de especificidad

- **Nivel de instrucción (ILP)** – se ejecutan en paralelo las instrucciones de un programa. Granularidad fina.
 - Distintas instr. a la vez.
 - Ejecución especulativa (garantizar tiempo) → if else
- **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.
 - Ejecución forma paralela (MIMD, MISD)
 - Programar bucle en unidades de procesamiento.
- **Nivel de procedimiento (Proceso)** – los distintos procedimientos que constituyen un programa se ejecutan simultáneamente. Granularidad media.
 - Distintas rutinas (opera E/S) ejecutando a la vez.
- **Nivel de programa** – la plataforma ejecuta en paralelo programas diferentes que pueden corresponder, o no, a una misma aplicación. Granularidad gruesa.

Multi-core → Distintos programas en distintos proc. (Paralelismo).

Espectro
y métodos

Introducción

1.3 Clasificación de arquitecturas

Tipos de computadores

no proc. demasiado rápido (refrigeración)

- **Dispositivos móviles personales**

- Teléfonos móviles, tablets, ... Coste y eficiencia energética

- **Ordenadores sobremesa**

- Precio-rendimiento

- **Servidores** → Ordenadores altas prestaciones. Lo importante → Disponibilidad

- Disponibilidad, escalabilidad, rendimiento

- **Clusters** → Conjunto de computadores conectados → más computadores en paralelo para distribuir carga de peticiones

- LANs de sobremesas y servidores actuando como un gran computador. SaaS: búsquedas, redes sociales, video compartido, juegos multiusuario Warehouse - scale computer. SaaS → Software as a Service.

↓
+ dinero
en refrigeración

- **Embebidos** → Realiza una o algunas pocas funciones dedicadas

- Presentes en máquinas: microondas, lavadoras, impresoras, switches, coches ... Amplio espectro coste rendimiento

Arquitectura

Diseño

Intel:

Xeon → Servidores

i7-i9 → Para gráfico

Introducción

2. Diseño de computadores

Arquitectura

Diseño

Introducción

• Tarea de diseño

→ *Association for Computing Machinery*

• Propuesta conjunta ACM-IEEE

- Informática basada en tres paradigmas: teoría, abstracción y diseño (no prima ninguno sobre el otro)
- Teoría: Fuerte base matemática. Ciencias formales
 - Definición
 - Teorema
 - Demostración
 - Interpretación

2. Diseño de computadores

Arquitectura

Diseño

Introducción

• Tarea de diseño

• Propuesta conjunta ACM-IEEE

- Informática basada en tres paradigmas: teoría, abstracción y diseño (no prima ninguno sobre el otro)
- **Abstracción: Ciencias experimentales. (Física, química)**
 - Hipótesis
 - Construcción de un modelo y realización de predicciones
 - Diseño de experimentos y recogida de resultados
 - Análisis de resultados

2. Diseño de computadores

Arquitectura

Diseño

• Tarea de diseño

• Propuesta conjunta ACM-IEEE

- Informática basada en tres paradigmas: teoría, abstracción y diseño (no prima ninguno sobre el otro)

• Diseño: Ingenierías. Ciencias aplicadas

Pasos

- Establecer requerimientos → *Aquello que se pide (demanda marcada)*
- Especificar → *Construir*
- Realización del sistema
- Prueba del sistema

Introducción

2.1 El proceso de diseño de computadores

Arquitectura

Diseño

Introducción

- **Establecer requerimientos funcionales**
- **Especificar el sistema**
- **Realización del sistema**
- **Prueba del sistema**

2.1 El proceso de diseño de computadores

Arquitectura

Diseño

Introducción

a. Establecer requerimientos funcionales y especificar

- **Funcionalidades inspiradas por el mercado y el software de aplicación que determinan características específicas del sistema**
- **Especificación en base a criterios de coste, rendimiento, consumo y disponibilidad para el mercado pensado**

2.1 El proceso de diseño de computadores

Diferencias en función del requerimiento funcional

a. Establecer requerimientos funcionales y especificar

Arquitectura

Diseño

Compiladores
&
ejecutables

Introducción

Requerimientos funcionales	Características típicas requeridas o soportadas
Área de aplicación <ul style="list-style-type: none">• Móviles• PCs• Servidores• Clusters/Warehouse-Scale Computers• Computación embebida	Objetivo del computador <ul style="list-style-type: none">• Rendimiento en tiempo real para muchas tareas, incluyendo gráficos, video y audio; eficiencia energética.• Rendimiento equilibrado para muchas tareas, incluyendo gráficos, video, y audio.• Soporte para BB.DD. y transacciones; alta fiabilidad y disponibilidad; escalabilidad• Alta productividad para tareas independientes; corrección de errores en memoria; proporcionalidad energética• A menudo requiere soporte especial para video y audio (y otras extensiones específicas de aplicaciones); limitaciones en consumo y se puede requerir control de potencia; limitaciones de tiempo real.
Nivel de compatibilidad software En lenguaje de programación Código binario compatible	Determina la cantidad de software existente para la máquina Más flexible para el diseñador, necesita nuevo compilador La arquitectura está completamente definida (poca flexibilidad), pero no necesita invertir en software ni en portar programas
Requerimientos del S.O. Tamaño del espacio de direcciones Gestión de memoria Cambio de contexto Interrupciones Protección	Características necesarias para soportar el S.O. requerido Muy importante, puede limitar aplicaciones Para S.O. modernos; puede ser plana, paginada, segmentada. Requerido para interrumpir y recomenzar un programa Tipos de soporte impactan en el diseño hardware y S.O. Diferentes S.O. y necesidades de aplicación: protección de páginas frente a protección de segmentos.
Estándares Punto flotante Bus E/S Sistema operativo Redes Lenguajes de programación	Ciertos estándares pueden ser requeridos por el mercado Formato y aritmética: IEEE 754, aritmética especial para gráficos o procesamiento de señal Dispositivos E/S: Serial ATA, Serial Attach SCSI, PCI Express UNIX, Windows, Linux Distintas redes: Ethernet, Lenguajes (ANSI C, C++, Java, Fortran) afectan al repertorio de instrucciones.

2.1 El proceso de diseño de computadores

b. Decisiones de implementación

¿Como se implementa mejor una funcionalidad requerida?

La decisión de implementación software o hardware

❖ Ventajas implementación software

- ❖ El bajo coste errores → Todo es modificable (más sencillo)
- ❖ Facilidad de diseño
- ❖ Actualización simple → Actualizar computadores, etc.

❖ Ventajas implementación hardware

- ❖ Rendimiento → Velocidad (y que tenga uso)
(Ley de Amdahl)

2.1 El proceso de diseño de computadores

Arquitectura

Diseño

Introducción

c. Consideración de las nuevas tendencias

◆ **Diseñador consciente de tendencias:**

- ◆ Utilización del computador
- ◆ Tecnología de computadores
- ◆ Una arquitectura a nivel lenguaje máquina con éxito puede durar decenas de años (núcleo de la IBM 360 desde 1964, 50 años)

2.1 El proceso de diseño de computadores

Arquitectura

Diseño

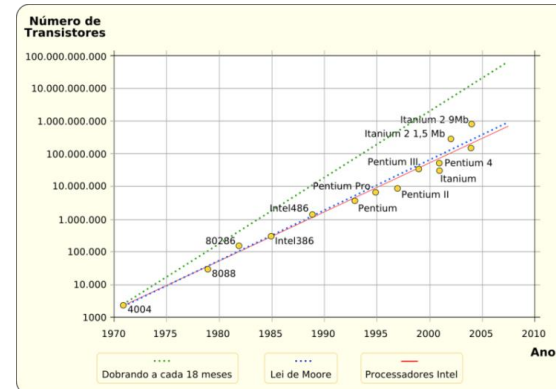
c. Consideración de las nuevas tendencias

◆ Tendencias en tecnologías hardware [Agarwall, 00]

Tecnología	Tendencias de rendimiento y densidad
Tecnología de CI	<p>El número de transistores en un chip aumenta aproximadamente el 35% por año, x4 en 4 años.</p> <p>La velocidad de los dispositivos aumenta casi a esa rapidez.</p> <p>[Agarwall, 00] disminución tasa crecimiento a 12% anual (consecuencia de procesos de 0,035 micras previstos para 2014)</p>
DRAM semiconductora	<p>La densidad aumenta en un 60% por año, cuadruplicándose en tres años. 2011 25-40% x2 cada 2-3 años [Kim, 2005]</p> <p>La duración del ciclo ha mejorado muy lentamente, decreciendo aproximadamente una tercera parte en diez años.</p>
Tecnología de almacenamiento secundario	<p>La densidad del disco magnético aumenta desde 2004 40% año x2 cada 3 años.</p> <p>El tiempo de acceso ha mejorado un tercio en diez años. Crecimiento de los discos SSD (NAND SLC, MLC y TLC)</p>

◆ Ley de Moore → A nivel transistores

- ◆ Las velocidades de cómputo y las densidades de almacenamiento se duplican cada 18 meses



Introducción

Licensed under [CC-BY-SA](#) by the author Max Roser.

2.1 El proceso de diseño de computadores

Field Programmable Gate Array → Programar conexiones entre puertos
↳ Probar hardware

c. Consideración de las nuevas tendencias

◆ Tendencias software

- ◆ Creciente cantidad de memoria utilizada por los programas y sus datos
- ◆ Sustitución del lenguaje ensamblador por los lenguajes de alto nivel.
- ◆ Reorientación de las arquitecturas hacia el soporte de los compiladores

• Juegos de instrucciones pensando en el compilador.

Arquitectura

Diseño

Introducción

2.2 Principios de diseño de computadores

Arquitectura

Diseño

Introducción

a. Acelerar el caso común

◆ Favorecer el caso frecuente

- ◆ **Ejemplo:** El desbordamiento de la suma es poco frecuentemente.
- ◆ El principio indicaría la optimización del caso sin desbordamiento
- ◆ La cuantificación de este principio se conoce como la ley de Amdahl

◆ Ley de Amdahl

- ◆ Define la ganancia de rendimiento o aceleración que puede obtenerse al mejorar alguna característica de un computador
- ◆ La mejora obtenida en el rendimiento al utilizar algún modo de ejecución más rápido está limitada por la fracción de tiempo en que se puede utilizar ese modo más rápido

$$\text{Aceleración} = \frac{\text{Rendimiento con mejora}}{\text{Rendimiento sin mejora}} = \frac{\text{Tiempo ejecución sin mejora}}{\text{Tiempo ejecución con mejora}}$$

2.2 Principios de diseño de computadores

Arquitectura

Diseño

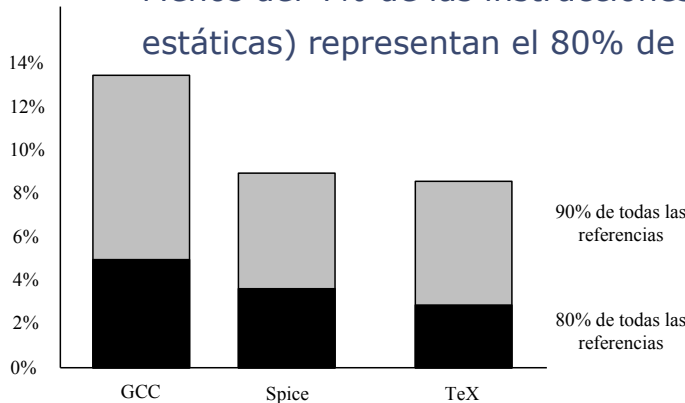
Introducción

b. Ley de rendimientos decrecientes

- La mejora incremental en la aceleración conseguida por una mejora adicional en el rendimiento de una parte del cálculo disminuye tal y como se van añadiendo mejoras.

c. Localidad de referencia → Programas usan pequeño rango de código mucho.

- Tendencia de los programas a reutilizar los datos e instrucciones usados recientemente. Los programas suelen emplear el 90% de su tiempo de ejecución en el 10% del código.
- Menos del 4% de las instrucciones del programa Spice (instrucciones estáticas) representan el 80% de las instrucciones dinámicas.



- Localidad temporal:** Los elementos accedidos recientemente probablemente serán accedidos en un futuro próximo.
- Localidad espacial:** Los elementos cuyas direcciones son próximas tienden a ser referenciados juntos en el tiempo.