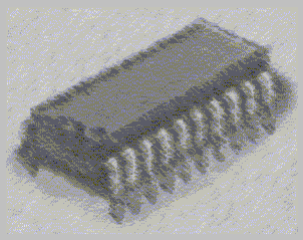
Tema 1. Introducción a la arquitectura de computadores: diseño, coste y rendimiento



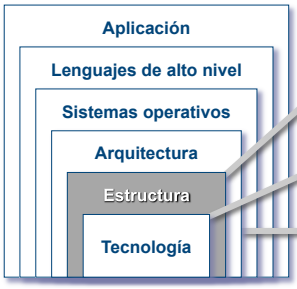
Arquitectura de Computadores I

I. T. Informática de Sistemas

Curso 2009-2010

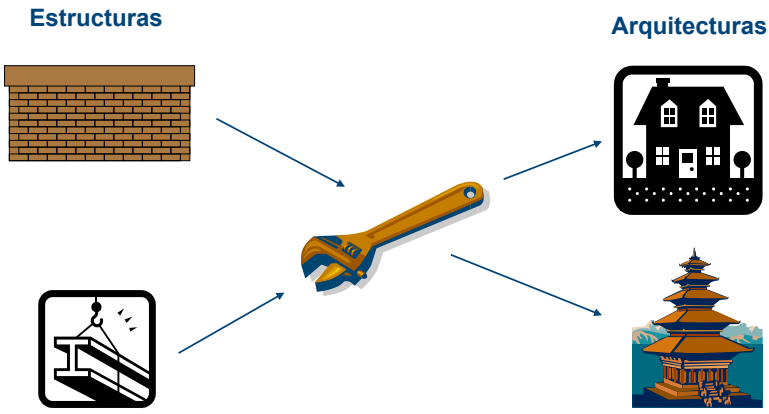
Índice

Relación de asignaturas en el plan de estudio de I.T. Informática de Sistemas

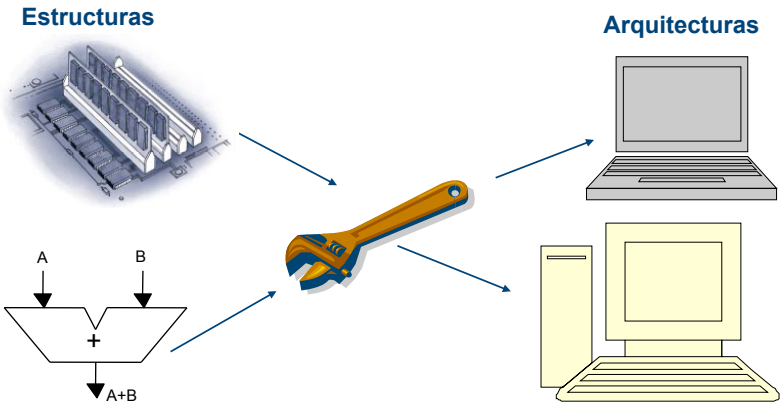


Asignatura	Contenidos
Estructura de Computadores (1er semestre)	Organización de las unidades funcionales y su interconexión para realizar las especificaciones de la arquitectura
Componentes y circuitos electrónicos (2º semestre)	Diseño de circuitos digitales
Arquitectura de Computadores I (4º semestre)	Sistema que integra tanto el hardware, como el software, y algoritmos para realizar los cálculos del computador
Arquitectura de Computadores II (5º semestre)	

Estructura vs. Arquitectura (I)



Estructura vs. Arquitectura (II)



Estructura vs. Arquitectura (y III)

Estructuras

- Son la unidades funcionales: memorias, puertas lógicas, buses, circuitos integrados, procesadores …. a partir de las cuales se formarán otras

Arquitecturas

- Dependiendo de cómo se organicen las estructuras básicas tendremos diferentes arquitecturas que responderán a las especificaciones que de coste y rendimiento se deseen obtener. PC, supercomputador, estación de trabajo …

Tareas del diseñador de computadores (I)

- La tarea de un diseñador de computadores consiste en determinar que atributos son importantes para una nueva máquina, o analizar los cuellos de botella de un sistema ya existente y lograr maximizar el rendimiento dentro de unos costes razonables de los que no puede pasarse
- Para ello debe definir una serie de requisitos funcionales que el sistema deberá cumplir
- Tendrá en cuenta: el área de aplicación, el nivel de compatibilidad software, los requisitos de los sistemas operativos y los estándares existentes

Tareas del diseñador de computadores (II). Área de aplicación

- El **área de aplicación** hace referencia al objetivo por el que se crea el computador o por el que se mejora el existente. Pueden ser:
  - **Computador de propósito general**: el rendimiento se equilibra entre la variedad de tareas que puede realizar
  - **Computador científico**: requerirá una alta capacidad de realizar operaciones en coma flotante
  - **Computador empresarial**: proporcionará soporte para procesar las transacciones de bases de datos, realizará cálculos simples, soportará lenguajes como Cobol, RPG I y II
  - **Computadores tolerantes a fallos**: deberán tener capacidad de detectar fallos y reconfigurarse. Normalmente tiene duplicados, o triplicados, todos los elementos que lo forman

Tareas del diseñador de computadores (III). Compatibilidad sw.

- La **compatibilidad software** hace referencia a la cantidad de aplicaciones existentes que podrán ejecutarse en la máquina nueva o mejorada. Entre las posibilidades destacan:
  - **A nivel de lenguaje de programación de alto nivel**: es mucho más flexible y cómodo para el diseñador de computadores pero requiere un nuevo compilador
  - **A nivel de compatibilidad binaria**: existen diferentes métodos pero todos tienen en común que el juego de instrucciones ya está definido, es menos flexible pero no requiere generar nuevo software o portarlo

Tareas del diseñador de computadores (III). Requisitos del S.O.

- Los **requisitos del sistema operativo** o sistemas operativos que se hayan escogido para la máquina:
  - **Tamaño del espacio de direcciones**: es una de las características más importantes a tener en cuenta ya que puede limitar las aplicaciones
  - **Manejo de memoria**: es un requisito para los sistemas operativos actuales, puede ser paginada o segmentada
  - **Memoria virtual**: decidir si la arquitectura ayudará o no al sistema operativo a manejar la memoria virtual
  - **Protección**: existen diferentes necesidades para las aplicaciones y sistemas operativos
  - **Tiempo real**: se deberá tener en cuenta si se escogerá un sistema operativo de tiempo real

Tareas del diseñador de computadores (y IV). Estándares

- La inclusión o cumplimiento de estándares hacen referencia a cuales cumplirá la arquitectura:
  - **Coma flotante**: si se empleará el estándar IEEE 754, DEC, IBM, …
  - **Buses de entrada salida**: que buses permitirá la arquitectura: SCSI, SCSI II, UltraSCSI, VME, FutureBus, …
  - **Sistemas operativos**: UNIX, Linux, Windows, DOS …
  - **Redes**: el soporte que proporcionará a determinados tipos de redes: Ethernet, ATM, Token ring, …
  - **Lenguajes de programación**: ya que afectarán al juego de instrucciones del computador: ANSI C, Fortran 77, 88, ANSI Cobol, …

Coste vs. rendimiento

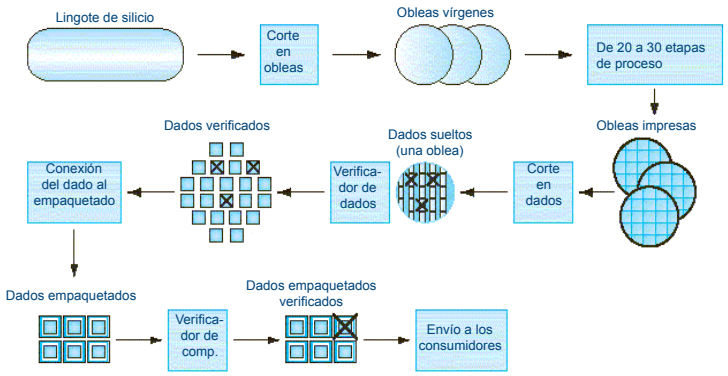
- El coste y el rendimiento suelen estar enfrentados por lo que el diseñador de computadores debe llegar a una solución de compromiso entre ambos parámetros
- Coste y rendimiento suelen ser restricciones a tener en cuenta por el diseñador de computadores
- Ejemplos de la relación entre el rendimiento y el coste:
  - **Computador personal**: es un sistema barato con rendimiento medio
  - **Estación de trabajo**: tiene un buen rendimiento pero no se dispara el precio
  - **Supercomputador**: prima el rendimiento. El coste no importa

Coste (I)

- El coste es un parámetro a tener muy en cuenta al diseñar un nuevo procesador o al modificar uno existente
- Los factores que influyen en el coste del silicio son:
  - **El número de puertas**: influye en el número de transistores que se necesitan y por tanto un aumento de estos últimos requiere un área de silicio mayor
  - **Conexiones entre elementos**: el número y la longitud de estos
  - **Regularidad del diseño**: cuanto más regular sea el diseño, menos área ocupará

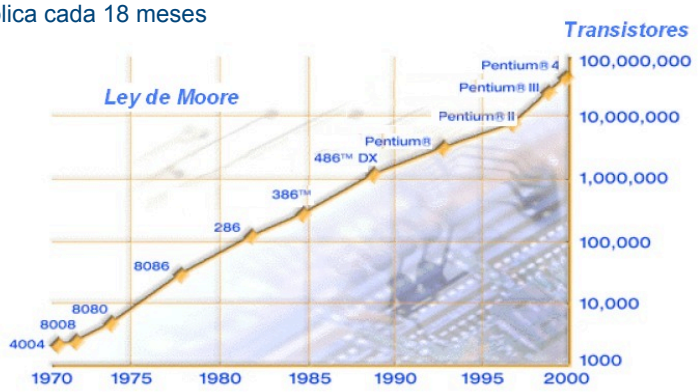
Coste (II)

- Proceso de fabricación de chips



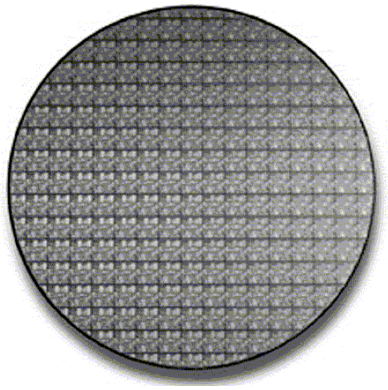
Coste (III) Ley de Moore

- La Ley de Moore postula que el número de transistores integrados se duplica cada 18 meses



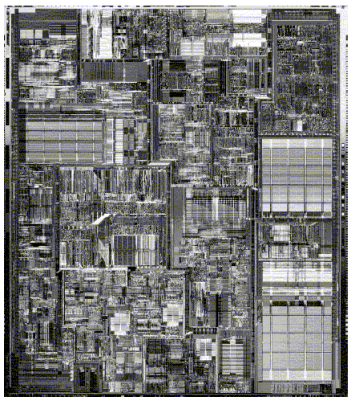
Coste (IV)

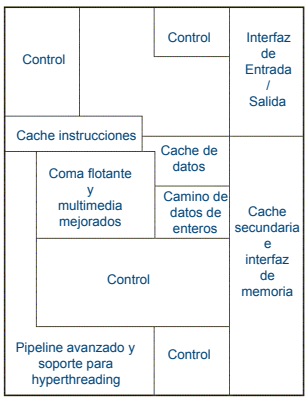
- Oblea de 200 mm. con procesadores Pentium 4



Coste (V)

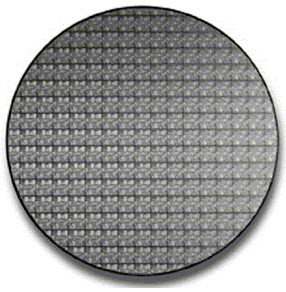
- Pentium 4 a 3.06 GHz





Coste (VI)

- Cálculo del coste por dado (procesador):



$$\text{Coste por dado} = \frac{\text{Coste por oblea}}{\text{Dados por oblea x ganancia}}$$

$$\text{Dados por oblea} = \frac{\text{Área de la oblea}}{\text{Área del dado}} = \frac{\partial \text{ x } (\text{Diametro oblea} / 2)^2}{\text{Área del dado}}$$

$$\text{Ganancia} = (1 + \frac{\text{Defectos por unidad de área x Área del dado}}{\alpha})^{-\alpha}$$

α suele ser 3 y los defectos desde 0,6 a 1,2 por cm. cuadrado

Coste (y VII)

- ***Ejemplo**:* encontrar el número de dados teóricos para una oblea de 25 cm. de diámetro y los reales mediante el cálculo de la ganancia, suponiendo una densidad de defectos de 0,8 por cm^2. El lado del dado es de 1,5 cm. y α=3

$$\text{Dados por oblea} = \frac{\text{Área de la oblea}}{\text{Área del dado}} = \frac{\partial \text{ x } (\text{Diametro oblea} / 2)^2}{\text{Área del dado}} = \frac{\partial \text{ x } (25/2)^2}{1{,}5 \text{ x } 1{,}5} = 218 \text{ dados teóricos}$$

$$\text{Ganancia} = (1 + \frac{\text{Defectos por unidad de área x Área del dado}}{\alpha})^{-\alpha} = (1 + \frac{0{,}8 \text{ x } 1{,}5 \text{ x } 1{,}5}{3})^{-3} = 0{,}24$$

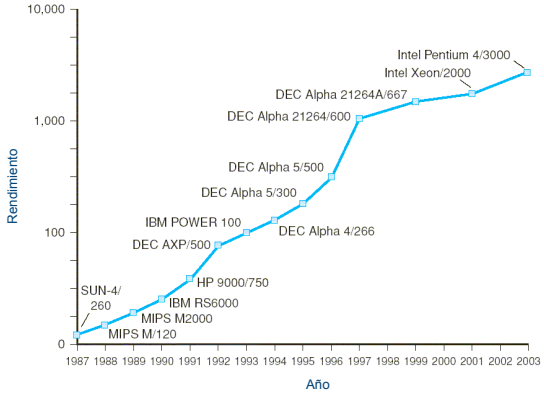
$$\text{Dados reales} = \text{Dados por oblea teóricos x Ganancia} = 218 \text{ x } 0{,}24 = 52 \text{ dados reales}$$

Rendimiento (I). Introducción

- Los computadores se construyen para realizar un gran número de cálculos en el menor tiempo posible
- Para medir la rapidez y la potencia de cálculo se emplean:
  - El **tiempo de ejecución de un trabajo**: cuanto menos tarde, mejor
  - La **productividad, o throughput**: que es la cantidad de trabajo que es capaz de ejecutar en un tiempo determinado. Cuanto más realice, mejor
- El tiempo que tarde en ejecutar un programa depende de varios factores: de las operaciones de entrada/salida, del acceso a memoria, del tiempo de CPU consumido por el sistema operativo y del tiempo de CPU del usuario
- El tiempo de ejecución hace referencia al tiempo de CPU de usuario

Rendimiento (II). Evolución

- Evolución del rendimiento de los procesadores



Rendimiento (III). Métricas

- **MIPS**: Millones de instrucciones por segundo
- **MFLOPS**: Millones de operaciones en coma flotante por segundo
- Existen una serie de test que permiten medir el rendimiento de todo el sistema en conjunto
- Una forma de medir el rendimiento es por el tiempo de ejecución de los programas

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

- Con lo que se puede decir que una máquina X es n veces más rápida que otra Y si:

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

Rendimiento (IV). Métricas

- Rendimiento de la CPU

Tiempo de ejecución de CPU = Ciclos de reloj de la CPU x Tiempo de
para un programa para un programa ciclo de reloj

$$\text{Tiempo de ejecución de CPU para un programa} = \frac{\text{Ciclos de reloj de la CPU para programa}}{\text{Frecuencia de reloj}}$$

Tiempo de ejecución = Número de instrucciones x CPI x tiempo de ciclo

$$\text{Tiempo de ejecución} = \frac{\text{Número de instrucciones x CPI}}{\text{Frecuencia de reloj}}$$

Rendimiento (V). Métricas

- Resumiendo: los componentes básicos del rendimiento y la unidad de medida son:

Componentes del rendimiento	Unidad de medida
Tiempo de ejecución de CPU para un programa	Segundos por programa
Número de instrucciones	Número de instrucciones ejecutadas por el programa
Ciclos por instrucción (CPI)	Número medio de ciclos por instrucción
Tiempo de ciclo de reloj	Segundos por ciclo de reloj

Rendimiento (VI). Métricas

- **Rendimiento de la memoria**
  - **Tamaño en número de palabras.** Aumenta la funcionalidad del sistema
  - **Latencia o tiempo de respuesta en ns**. Ligada a la rapidez de ejecución de los programas
  - A mayor tamaño de memoria menor rapidez

- **Rendimiento de los buses**
  - Número de líneas de comunicación.
  - Frecuencia de transmisión de los datos en MHz
  - **Ancho de banda.** Megabytes por segundo

Rendimiento (VII). Benchmarks

- Los programas de evaluación recogen un amplio espectro de funciones para las que son normalmente empleados los ordenadores
- Los programas de evaluación son conocidos como Benchmark
- Características de los programas de evaluación:
  - Portables a cualquier tipo de máquina: PC, estación de trabajo, multiprocesador
  - Informe de resultados fáciles de interpretar
  - Tienen en cuenta el efecto del compilador
- Programas de evaluación más empleados:
  - SPEC CPU 2000
  - iCOMP

Rendimiento (VIII). Benchmarks

- SPEC CPU 2000

Programas del CINT2000		Programas del CFP 2000	
Nombre	**Descripción**	**Nombre**	**Descripción**
Bzip2	Compresión	Facerec	Reconocimiento de caras
Crafty	Juego de ajedrez	Applu	Ecuaciones diferenciales
Eon	Visualización	Apsi	Predicción meteorológica
Gap	Intérprete de teoría de grupo	Sixtrack	Acelerador de partículas
Gcc	Compilador C de GNU	Wupwise	Física cuántica
Gzip	Compresión	Art	Reconocimiento de imágenes
Mcf	Optimización combinatoria	Mgrid	Multimalla 3D
Parser	Procesador de texto	Galgel	Dinámica de fluidos
Perbmk	Entorno lenguaje PERL	Ammp	Química computacional
Twolf	Simulador de circuitos	Lucas	Teoría de números
Vortex	Base de datos OO	Fma3d	Simulación de colisiones
Vpr	Enrutado de chips FPGA	Swin	Modelado de superficie del agua
		Equake	Ondas sísmicas
		Mesa	Libreía gráficos 3D

Rendimiento (IX). Benchmarks

- iCOMP 3.0

Benchmark	Peso porcentual	Aplicaciones a las que caracteriza
CPUmark99	20	Computación intensiva de números enteros
Wintune98 Advanced CPU integer test	20	Computación intensiva de números enteros
MulitmediaMark 99	25	Aplicaciones A / V en Internet
3D Winbench99-3Dlighting and Transf. Test	20	Geometría e iluminación 3D
Winbench99-FPU WinMak	5	Computación intensiva en punto flotante
Jmark 2.0 Processor test	10	Aplicaciones java

Rendimiento (X). Ley de Amdahl

- **Cuello de botella**. Subsistema o subsistemas que degradan el rendimiento del equipo en general. Mejorar el caso común
- La ley de Amdahl mide el impacto en el rendimiento del cambio en un subsistema
- **Ley de Amdahl**:

$$A = \frac{\text{Tiempo sin mejora}}{\text{Tiempo con mejora}} = \frac{1}{(1 - F_m) + \frac{F_m}{A_m}}$$

- Am factor de mejora que ha introducido el subsistema alterado
- Fm. Fracción de tiempo que el sistema emplea el subsistema alterado

Rendimiento (XI). Ley de Amdahl

- Si Am = ∞ $A = \frac{1}{(1 - F_m)}$
  - El porcentaje máximo que un subsistema puede acelerarse actuando sobre uno de los componentes está acotado en función de cuánto se use
- Si Fm = 0 $A = 1$
  - La mejora sobre un componente no tiene efecto sobre el sistema total si ese componente no se emplea
- Si Fm = 1 $A = A_m$
  - Todo el tiempo de ejecución del programa se dedica a emplear el componente mejorado, con o que la ganancia en velocidad experimentado por el subsistema se trasladará al sistema general

Rendimiento (y XII). Ley de Amdahl

- ***Ejemplo:***
  - Se desea mejorar el rendimiento de un computador introduciendo un coprocesador matemático que realice las operaciones en la mitad de tiempo. Calcular la ganancia en velocidad del sistema para la ejecución de un programa si el 60% del mismo se dedica a operaciones aritméticas. Si el programa tarda 12 segundos en ejecutarse sin la mejora. ¿cuánto tardará con la mejora?
  - Am = 2 y Fm = 0,6

$$A = \frac{1}{(1-0{,}6) + \frac{0{,}6}{2}} = 1{,}42$$

  - Con lo que el sistema es un 42% más rápido

$$A = \frac{TiempoEjecuciónSinMejora}{TiempoEjecuciónConMejora} \Rightarrow 1{,}42 = \frac{12}{TiempoEjecuciónConMejora}$$

  - Lo que hace que el programe tarde 8,45 segundos

Bibliografía

- Estructura y diseño de computadores
  David A. Patterson y John L. Hennessy. Reverté, 2000
  Capítulo 2

- Arquitectura de computadores. Un enfoque cuantitativo
  John L. Hennessy y David A. Patterson. Mc Graw Hill, 3ª ed, 2002
  Capítulos 1 y 2