

Fundamentos del Diseño de Computadores

Arquitectura de Computadores

J. Daniel García Sánchez (coordinador)

David Expósito Singh

Javier García Blas

Óscar Pérez Alonso

J. Manuel Pérez Lobato

Grupo ARCOS

Departamento de Informática

Universidad Carlos III de Madrid



- 1 Introducción
- 2 Perspectiva histórica
- 3 Clasificación de computadores
- 4 Paralelismo
- 5 Arquitectura del Computador
- 6 Conclusión

Arquitectura de Computadores

*The term **architecture** is used here to describe the attributes of a system as seen by the programmer, i.e., the conceptual structure and functional behavior, as distinct from the organization of the data flow and controls, the logical design, and the physical implementation.*

Gene Amdahl et al.
Architecture of the IBM System.
IBM Journal of Research and Development
Vol 8 (2) pp. 87-101. 1964.

¿Qué es la arquitectura de computadores?

- **Computer Architecture** is the science and art of *selecting and interconnecting* hardware components to create computers that meet **functional, performance and cost** goals.

WWW Computer Architecture Page.

- La **Arquitectura de Computadores** no tiene nada que ver con la construcción de edificios. 😊

¿Por qué estudiar Arquitectura de Computadores?

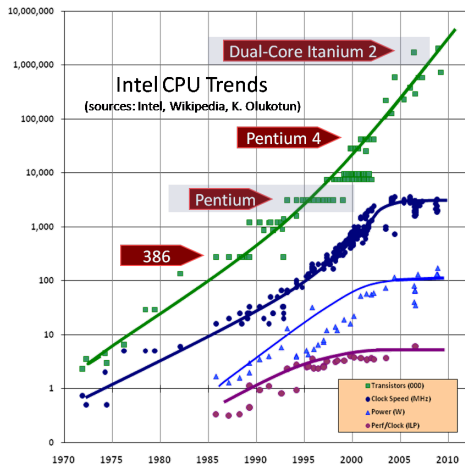
- Sin computadores no hay Ingenieros en Informática.
- Para comprender las tendencias de la próxima década.
 - ¿Como serán los computadores del futuro?
 - ¿Qué se podrá y que no se podrá hacer con ellos?
- Para comprender las limitaciones de los computadores.
 - ¿Qué se puede hacer? ¿Qué es lo que no se puede?
 - ¿Cuáles son los límites del rendimiento?
- Para aprobar esta asignatura. 😊

La ley de Moore

- El número de transistores por chip se duplica cada N meses.
 - Donde $12 < N < 24$.
 - Gordon Moore, 1965.

- Observaciones:
 - 1 Obtenida a partir de datos experimentales → Ley empírica.
 - 2 Se sigue cumpliendo todavía.
 - 3 No tiene por qué traducirse directamente en incrementos de rendimiento.

Transistores por chip



- Actividad: Lea completamente el artículo:

Fuente: **The free lunch is over.**
Herb Sutter.

<http://www.gotw.ca/publications/concurrency-ddj.htm>

Efectos de la aparición del RISC

- Mejora de la capacidad disponible
 - Un microprocesador de alta gama más potente que un supercomputador de diez años antes.

Coste frente a rendimiento

- La mejora de ratio coste/rendimiento da lugar a nuevas clases de computadores.
 - 80's: PC y *workstations*.
 - 00's:
 - *Smart-phones* y *tablets*.
 - Incremento del uso de grandes centros de datos con miles de nodos vistos como un único computador.

La revolución RISC

- Mejora continua de semiconductores ha dado lugar al dominio de computadores basados en microprocesador.
 - Desaparición de los minicomputadores.
 - *Mainframes* y supercomputadores construidos como colecciones de microprocesadores.
- Incremento sostenido del rendimiento de 1986 a 2003:
 - **52% anual.**
- **¡Ha dejado de cumplirse!**

- 1 Introducción
- 2 Perspectiva histórica
- 3 Clasificación de computadores
- 4 Paralelismo
- 5 Arquitectura del Computador
- 6 Conclusión

Primera revolución: El microprocesador

- La revolución del microprocesador.
 - Generada a partir de un único cambio.
 - Suficientes transistores (25,000) en un único chip para un procesador de 16 bits.
 - **Ventajas:**
 - **Más rápido:** Menos salidas del chip.
 - **Más barato:** Todo en un chip.
- Nuevos segmentos de mercado generados por la innovación.
 - Computadores de escritorio, CD/DVD, portátiles, consolas de videojuego, decodificadores TV, cámaras digitales, MP3, GPS, ...
- Impacto en mercados existentes:
 - Supercomputadores, *mainframes*, ...

Primer microprocesador

- Intel 4004 (1971).
 - **Dominio de aplicación:** Calculadoras.
 - **Tecnología:** 10,000 nm.

- **Datos:**
 - 2300 transistores.
 - 13 mm².
 - 108 KHz.
 - 12 Voltios.

- **Características:**
 - Datos de 4 bits.
 - Camino de datos en un ciclo.

Segunda revolución

- Extracción del paralelismo implícito a nivel de instrucción (ILP).
 - El hardware tiene recursos que pueden usarse en paralelo.
- **Elementos:**
 - **Segmentación:** Permitió incrementar frecuencias de reloj.
 - **Cachés:** Necesarias para incrementar las frecuencias de reloj.
 - **Coma flotante:** Integradas en el chip.
 - Incremento en la **profundidad del pipeline** y **especulación de salto**.
 - **Emisión múltiple:** Arquitecturas superescalares.
 - **Planificación dinámica:** Ejecución fuera de orden.

Culminación de procesadores de un núcleo

- Intel Pentium 4 (2003).
 - **Dominio de aplicación:** Escritorio/Servidor.
 - **Tecnología:** 90 nm (1/100x).
- **Datos:**
 - 55M transistores (20,000x).
 - 101 mm² (10x).
 - 3.4 GHz (10,000x).
 - 1.2 Volts (1/10x).
- **Características:**
 - Datos de 32/64 bits (16x).
 - Segmentación en 22 etapas (más tarde 31).
 - 3-4 instrucciones por ciclo (superescalar).
 - Dos niveles de caché en chip.
 - Paralelismo de datos (SIMD).
 - *Hyper-threading*.

Tercera revolución

- Soporte a paralelismo explícito de datos y de hilos.
 - Hardware ofrece recursos paralelos y software especifica su uso.
 - El paralelismo deja de ser ocultado por el hardware.
 - **Razón**: Beneficios cada vez menores de ILP.

- **Elementos**:
 - **Instrucciones vectoriales**: Intel SSE.
 - Soporte general para aplicaciones **multi-hilo**.

Procesadores multi-core

- Intel Core i7 (2009).
 - **Dominio de aplicación:** Escritorio / Servidor.
 - **Tecnología:** 45 nm (1/2x).
- **Datos:**
 - 774M transistores (12x).
 - 296 mm² (3x).
 - 3.2 GHz – 3.6 GHz ($\approx 1x$).
 - 0.7 – 1.4 Voltios ($\approx 1x$).
- **Características:**
 - Datos de 128 bits (2x).
 - Segmentación de 14 etapas (0.5x).
 - 4 instrucciones por ciclo ($\approx 1x$).
 - Tres niveles de caché en chip.
 - SIMD, *hyper-threading*
 - 4 cores (4x).

Tendencias arquitectónicas

- Paralelismo a nivel de instrucción:
 - Ejecución paralela de instrucciones.
 - Imposible mejorar significativamente ILP desde 2003-2005.
 - El hardware y el compilador *conspiran* para ocultar detalles al programador.
 - Programador con vista **muy simplificada** del hardware.
- Nuevos modelos para mejorar rendimiento:
 - **Data-Level** Parallelism (DLP).
 - **Thread-Level** Parallelism (TLP).
 - **Request-Level** Parallelism (RLP).
- **IMPORTANTE**: Todos ellos requieren reestructurar las aplicaciones para conseguir los incrementos de rendimiento prometidos.

- 1 Introducción
- 2 Perspectiva histórica
- 3 Clasificación de computadores**
- 4 Paralelismo
- 5 Arquitectura del Computador
- 6 Conclusión

Dispositivos móviles personales

- Dispositivos sin cables con IU multimedia.
 - Dispositivos móviles, tablets, ...
- **Precio:** \$100 – \$1000.
- **Precio de procesador:** \$10 – \$100.
- **Factores críticos:**
 - Coste.
 - Energía.
 - Rendimiento.
 - Tiempo de respuesta.

Desktop

- Diseñados para ofrecer buen rendimiento a usuarios finales.
 - Desde *ultra-books* hasta estaciones de trabajo.
 - Desde 2008 más del 50% son portátiles
- **Precio:** \$300 – \$2500.
- **Precio de procesador:** \$50 – \$500.
- **Factores críticos:**
 - Precio-Rendimiento.
 - Energía.
 - Rendimiento de gráficos.

Servidores

- Usados para ejecutar aplicaciones de gran escala y dar servicio a múltiples usuarios de forma simultánea.
 - Creciendo desde los 80.
 - Sustitución de los *mainframes*.
- **Precio:** \$5,000 – \$10,000,000.
- **Precio de procesador:** \$200 – \$2,000.
- **Factores críticos:**
 - *Throughput* (tasa de procesamiento).
 - Disponibilidad.
 - Escalabilidad.
 - Energía.

Clusters / Warehouse Scale Computers (WSC)

- Una colección de computadores conectados mediante LAN que actúa como un computador más grande.
 - Alcanza más popularidad debido a crecimiento de **SaaS** (*Software as a Service*).
 - Cada nodo ejecuta su propio sistema operativo.
 - WSC → 10,000+ nodos.
- **Precio**: \$100,000 – \$200,000,000.
- **Precio de procesador**: \$50 – \$250.
- **Factores críticos**:
 - Precio-Rendimiento.
 - *Throughput* (tasa de procesamiento).
 - Proporcionalidad en energía.

Empotrados

- Computador dentro de otro sistema que ejecuta aplicaciones pre-establecidas.
 - Lavaplatos, consola de videojuegos, MP3, ...
- **Precio de procesador:** \$0.01 – \$100.
- **Factores críticos:**
 - Precio.
 - Energía.
 - Rendimiento de aplicación específica.

Ventas (2010)

Clase	Unidades vendidas
Dispositivos móviles personales	1,800,000,000
Escritorio	350,000,000
Servidores	20,000,000
Empotrados	19,000,000,000

- 1 Introducción
- 2 Perspectiva histórica
- 3 Clasificación de computadores
- 4 Paralelismo
- 5 Arquitectura del Computador
- 6 Conclusión

Velocidad de máquina secuencial

- Una **máquina secuencial** de 1 TFLOP (10^{12} FLOPS):
 - Los datos deben viajar una cierta **distancia** (r) desde memoria a CPU.
 - 1 **dato elemental** por ciclo:
 - $\Rightarrow 10^{12}$ veces por segundo $\Rightarrow 10^{-12}$ s. por ciclo.
 - Datos viajando a la **velocidad de la luz**: $c = 3 \cdot 10^8$ m/s.
 - $r = 3 \cdot 10^8 \cdot 10^{-12} = 0.3\text{mm}$

- 1 TB de datos en **superficie** de 0.3 mm^2 :
 - Cada dato debería almacenarse en 3 Angstroms (aprox.).
 - ¡El tamaño de un átomo pequeño!

- **CONCLUSIÓN**: Aproximación secuencial no factible.

Tipos de paralelismo

- Todos los computadores presentan restricciones de coste y consumo de energía.
- El **paralelismo** aparece como el principal mecanismo de diseño de computadores.
- **Tipos de paralelismo en las aplicaciones:**
 - **Paralelismo de datos:** Una operación aplicada a muchos datos.
 - **Paralelismo de tareas:** Tareas operan independientemente y en paralelo.

Paralelismo hardware

- **ILP**: *Instruction-Level Parallelism*.
 - Explota paralelismo de datos con ayuda del compilador (segmentación, ejecución especulativa, ...).
- Arquitecturas **Vectoriales** y **GPUs**.
 - Explota paralelismo de datos aplicando la misma operación a varios datos en paralelo.
- **TLP**: *Thread-Level Parallelism*.
 - Explota paralelismo de datos o tareas en hardware altamente acoplado.
 - Permite interacciones entre hilos.
- **RLP**: Request-Level Parallelism.
 - Explota paralelismo entre tareas altamente desacopladas.

Taxonomía de Flynn (1966)

- Una clasificación de arquitecturas paralelas posibles.
- **SISD**: *Single Instruction / Single Data Stream*.
 - Mono-procesador.
 - Puede usar técnicas de ILP.
- **SIMD**: *Single Instruction / Multiple Data Stream*.
 - Las mismas instrucciones ejecutadas por procesadores diferentes sobre datos distintos.
 - **Alternativas**: Procesadores vectoriales, extensiones multimedia y *GPUs*.
- **MISD**: *Multiple Instructions / Single Data Stream*.
 - No se conocen implementaciones comerciales.
- **MIMD**: *Multiple Instructions / Multiple Data Stream*.
 - Cada procesador opera sobre sus propios datos ⇒ Paralelismo de tareas.

Más sobre MIMD

- Variedad de arquitecturas **MIMD**:
 - Arquitecturas altamente acopladas.
 - **TLP** (*Thread-Level Parallelism*): Arquitecturas *Multi/Many-core*.
 - Arquitecturas débilmente acopladas:
 - **RLP** (*Request-Level Parallelism*): *Clusters y WSCs*.

- **MIMD** es:
 - Más flexible y general que SIMD.
 - Más caro que SIMD.
 - Requiere suficiente granularidad de tareas.

- 1 Introducción
- 2 Perspectiva histórica
- 3 Clasificación de computadores
- 4 Paralelismo
- 5 Arquitectura del Computador**
- 6 Conclusión

Visiones de la arquitectura

- El diseño de computadores es complejo:
 - **Determinar** atributos importantes.
 - **Maximizar** rendimiento y eficiencia energética con control sobre coste, potencia y disponibilidad.

- Diferentes visiones sobre diseño de la arquitectura:
 - Diseño de juego de instrucciones.
 - Organización funcional.
 - Diseño lógico.
 - Implementación: diseño de circuitos, empaquetado, refrigeración, . . .
 - Integración con compiladores, sistemas operativos, . . .

ISA: Instruction Set Architecture

- **ISA**: Parte de la arquitectura visible al programador.
 - Instrucciones disponibles.
 - Número y tipo de registros.
 - Formato de instrucciones.
 - Modos de direccionamiento.
 - Condiciones de excepción.

Tipos ISA

- Casi todas las ISA actuales usan registros de propósito general.
- Versiones populares:
 - ISA **Registro-Memoria**:
 - Muchas instrucciones pueden acceder a memoria principal.
 - Ej.: Intel 80x86.
 - ISA **Load-Store**:
 - Solamente instrucciones **load/store** pueden acceder a memoria principal.
 - Otras instrucciones operan sobre registros.
 - Ej.: ARM, MIPS.
- Las ISAs más recientes son *load-store*.

Direccionamiento de memoria

■ Tipos:

- **Direccionamiento por bytes**: Dirección identifica un byte de memoria.
- **Direccionamiento por palabras**: Dirección identifica una palabra de memoria.

■ **Direccionamiento más popular**: Direccionamiento por bytes.

■ **Variantes**:

- Todos los accesos debe estar alineados (Ej. MIPS).
- Los accesos alineados son más rápidos (Ej. 80x86).

Modos de direccionamiento

■ MIPS:

- Registro.
- Inmediato.
- Relativo a registro.

■ 80x86:

- Registro, inmediato, relativo a registro.
- Absoluto.
- Registro base con desplazamiento (2 registros).
- Registro base con índice escalado y desplazamiento. . . .

Tipos y tamaños de operandos

■ **Enteros:**

- 8 bits (carácter).
- 16 bits (carácter Unicode).
- 32 bits (entero / palabra).
- 64 bits (entero largo / doble palabra).

■ **Coma flotante:**

- 32 bits (simple precisión).
- 64 bits (doble precisión).
- 80 bits (80x86 doble precisión extendida).

Operaciones

■ Categorías:

- Transferencia de datos, aritmética, lógica, control, y coma flotante.

■ MIPS:

- Instrucciones simples, fáciles de implementar en *pipeline* (RISC).

■ 80x86:

- Muchas más instrucciones.
- Complejidad variable.

Instrucciones de flujo de control

■ Categorías:

- Bifurcaciones condicionales, saltos, llamadas y retorno de subrutina.

■ MIPS:

- Direccionamiento relativo a PC.
- Condiciones sobre valores de registros.
- Subrutinas colocan valor de retorno en registro.

■ 80x86:

- Direccionamiento relativo a PC.
- Condiciones sobre bits con códigos de condición.
- Llamadas a subrutina mediante pila.

Codificación de instrucciones

■ Longitud fija:

- MIPS.
- Todas las instrucciones de 32 bits.
- Decodificación de instrucción simplificada.

■ Longitud variable:

- 80x86.
- Longitud de 1 a 18 bytes.
- Reduce tamaño de programa.



- 1 Introducción
- 2 Perspectiva histórica
- 3 Clasificación de computadores
- 4 Paralelismo
- 5 Arquitectura del Computador
- 6 Conclusión

Resumen

- La **ley de Moore** todavía se mantiene.
 - Pero *“The free lunch is over”*.
- Nuevos modelos mejoran rendimiento (**DLP**, **TLP**, **RLP**), pero requieren reestructurar las aplicaciones.
- Diversidad en clases de computadores con variación en propiedades y requisitos.
 - Móviles, Desktop, Servidores, WSC, Empotrados.
- Arquitecturas emergentes combinan **SIMD** y **MIMD**.
- Diferencia entre **ISA** y *arquitectura*.

Referencias

- **Computer Architecture. A Quantitative Approach**

5th Ed.

Hennessy and Patterson.

Secciones 1.1, 1.2, and 1.3.

- **The Free Lunch is over.**

Herb Sutter.

<http://www.gotw.ca/publications/concurrency-ddj.htm>

- **Welcome to the Jungle.**

Herb Sutter.

<http://herbsutter.com/welcome-to-the-jungle/>

Fundamentos del Diseño de Computadores

Arquitectura de Computadores

J. Daniel García Sánchez (coordinador)

David Expósito Singh

Javier García Blas

Óscar Pérez Alonso

J. Manuel Pérez Lobato

Grupo ARCOS

Departamento de Informática

Universidad Carlos III de Madrid