

# ARQUITECTURA DE SISTEMAS PARALELOS 1.

## PROGRAMA 2001/2002.

Facultad de Informática y Estadística. Cuarto Curso de Ingeniería Informática.

Profesores: D. Fernando Díaz del Río (coordinador) y D. Saturnino Vicente Díaz (y profesor a contratar).

### Temario de la asignatura Arquitectura de Sistemas Paralelos 1.



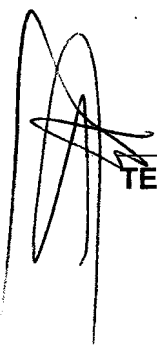
---

#### TEMA 0 PRESENTACIÓN DE LA ASIGNATURA (T: 1H).

---

#### TEMA 1 ANÁLISIS DE PRESTACIONES (T: 3H P: 1H).

---

- 
- 
- 
- 1.1 Repaso de los modelos y principales conceptos de arquitectura.
    - Arquitectura, organización y tecnología.
    - Jerarquías de memoria.
    - Modelo de von Neumann y Harvard: dificultades de paralelización.
    - Ejecución secuencial frente a ejecución paralela.
  - 1.2 Parámetros cuantitativos para la evaluación de prestaciones.
    - Rendimiento, productividad y coste.
    - Parámetros clásicos: MIPS, MFLOPS, etc.
    - Parámetros del tiempo de ejecución: CPI, frecuencia y número de instrucciones.
    - Ley de Amdahl y ley de localidad.
    - Modelado cuantitativo del acceso a memoria (bloqueos de memoria y  $CPI_{\text{bloqueo}}$ ).
    - Medias de rendimiento.
  - 1.3 Programas para la evaluación de prestaciones (*benchmarks*).
    - *Benchmarks* de juguete.
    - *Benchmarks* sintéticos.
    - Evaluación con programas reales.
    - Colecciones de programas: el caso de los SPECint y SPECfp.

---

#### TEMA 2 ARQUITECTURAS ENCADENADAS BÁSICAS (T: 12H P: 3H).

---

- 2.1 Introducción: concepto de encadenamiento de instrucciones.
- 2.2 Juegos de instrucciones orientados al encadenamiento.
  - Microinstrucciones, implementación cableada y etapas de una instrucción.
  - Características de un "buen" procesador encadenado: la importancia de la decodificación y de los accesos a memoria.
  - El problema del encadenamiento en los procesadores CISC y las ventajas de los RISC.
  - Factores involucrados en la aceleración de un procesador encadenado: CPI, periodo, número de instrucciones y bloqueos.
  - Relación entre el juego de instrucciones y la compilación.
- 2.3 Descripción del procesador ejemplo (DLX).

- Características principales: tipos de datos, juegos de registros, direccionamiento, operaciones.
- Juego y formatos de instrucciones.
- 2.4 Implementación del encadenamiento del DLX.
  - Implementación secuencial.
  - Implementación orientada al encadenamiento.
  - Detalle de la implementación.
  - Consideraciones tecnológicas del encadenamiento.
- 2.5 Dependencias, riesgos y bloqueos.
  - Dependencias en procesadores encadenados: tipos.
  - Riesgos y bloqueos.
  - Ejemplo de implementación de un bloqueo: inyección de NOP.
  - Cronograma de instrucciones.
  - Impacto en el CPI.
- 2.6 Dependencias estructurales.
  - Resolución de dependencias estructurales en el DLX.
  - Otras situaciones de riesgo estructural.
- 2.7 Dependencias de datos.
  - Tipos: RAW, WAR, WAW.
  - Desvíos o cortocircuitos (*bypasses*).
  - Implementación de desvíos.
  - Dependencias de memoria.
- 2.8 Dependencias de control.
  - Dependencias, riesgos y bloqueos de control.
  - Implementación adelantada de la resolución de los saltos.
  - Implementaciones especiales: saltos retrasados.
- 2.9 Predicción de saltos.
  - Predicción estática de salto condicionales.
  - Predicción dinámica de salto condicionales: BPB, BTB, *folding*.
  - Máquinas de estado predictivas y porcentaje de acierto en casos reales.
- 2.10 Instrucciones de duración variable.
  - Implementación.
  - Nuevos riesgos y cuantificación de bloqueos en programas científicos.

---

### TEMA 3      PLANIFICACIÓN (*SCHEDULING*) DE INSTRUCCIONES (T: 6H P: 3H).

- 3.1 Conceptos fundamentales.
  - Grafo y tipos de dependencias.
  - Bloques básicos.
  - Frecuencia de las dependencias.
- 3.2 Planificación estática.
  - Técnicas básicas de optimización del compilador. Ejemplos.
  - Límite teórico de ejecución dataflow.
- 3.3 Planificación dinámica.
  - Algoritmo del marcador (scoreboard).
  - Estaciones de reserva.
  - Renombrado dinámico de registros.
  - Implementación del Algoritmo de Tomasulo.
- 3.4 Desenrollado de bucles.
  - Bucles con paralelismo de datos.

- Desenrollado y entrelazado sistemáticos de iteraciones.
- Impacto en las prestaciones y reducción de instrucciones.
- Dificultades de aplicación en CISC.
- 3.5 Técnicas software avanzadas.
  - Segmentación software.
  - Planificación de trazas.
- 3.6 Ejemplos reales: planificación estática vs. dinámica.
  - El problema de la compatibilidad de las aplicaciones.
  - Casos reales: Evolución de algunas familias en cuanto a la planificación

## **Prácticas de Arquitectura de Sistemas Paralelos 1.**

Todas las prácticas incluyen una explicación previa (aproximadamente 0.5 horas) de los conceptos necesarios y de la metodología a seguir. La duración completa, incluyendo la explicación previa, elaboración del software y realización de pruebas es de alrededor de dos horas, aunque en función de la disponibilidad de laboratorios, las prácticas se podrán subdividir en varias sesiones.

Se impartirán las siguientes prácticas a lo largo del cuatrimestre:

*Práctica 1. Análisis de prestaciones y estudio de la memoria cache en distintas máquinas.*

*Práctica 2. Estudio del encadenamiento en DLX.*

*Práctica 3. Estudio de los bloqueos estructurales, de datos y de control en el DLX.*

*Práctica 4. Impacto en las prestaciones de la planificación estática y del desenrollado.*

*Práctica 5. Algoritmo de Tomasulo.*

## **Sistema de Evaluación.**

El examen de la asignatura constará de una parte teórica, que tendrá un valor en torno al 30% de la nota del examen, y otra de problemas (aproximadamente el 70% de la puntuación final). La correcta realización de las prácticas es condición indispensable para aprobar la asignatura.

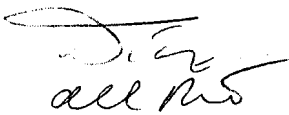
## **BIBLIOGRAFIA BASICA.**

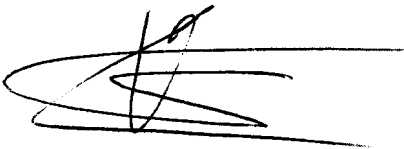
- [H&P96] J.L. Hennessy, D.A. Patterson "Computer Architecture. A Quantitative Approach". Morgan-Kaufmann (Second Edition), 1996.
- [MUELL00] S. M. Mueller, W. J. Paul. *Computer Architecture: Complexity and Correctness*. Edit. Springer. 2000.
- [H&P2000] D. A. Patterson, J. L. Hennessy. *Estructura y Diseño de Computadores. Interfaz Circuitería-Programación*. Edit. Reverté. 2000.
- [STON90] H.S. Stone, "High-Performance Computer Architecture". Addison-Wesley (Second Edition) 1990.
- [HWAN93] K. Hwang "Advanced Computer Architecture: Parallelism, Scalability, Programmability". McGraw-Hill, 1993.
- [KOGG81] P. M. Kogge "The Architecture of Pipelined Computers". McGraw-Hill, 1981.
- [LILJ91] D.J. Lilja, de., "Architectural Alternatives for Exploiting Parallelism". IEEE Comp. Soc. Press, 1991.
- [TANE99] A. S. Tanenbaum. *Structured Computer Organization*. Prentice Hall. 1999. Fourth Edition.
- [ALMA94] G.S. Almansí, A. Gottlieb "Highly Parallel Computing". Benjamin/Cummings (Second Edition), 1994.

## **BIBLIOGRAFIA COMPLEMENTARIA.**

- [TRED95] N. Tredennick, "Technology and Business: Forces Driving Microprocessor Evolution". *Proc. Of the IEEE*, vol. 83, no. 12, Diciembre 1995.

- [H&P95] D.A. Patterson, J.L. Hennessy "Organización y Diseño de Computadores. La Interfaz Hardware/Software". McGraw-Hill, 1995.
- [STAL96] W. Stallings, "Computer Organization and Architecture. Designing for Performance". Prentice Hall, 1996 (Fourth Edition).
- [JAIN91] R. Jain, "The Art of Computer Systems Performance Analysis: Techniques for Experimental Design, Measurement, Simulation and Modeling". John Wiley & Sons, 1991.
- [GILA95] R. Giladi, N. Ahituv "SPEC as a Performance Evaluation Measure". *IEEE Computer*, August 1995 (pp. 33-42).
- [KAIN96] R.Y. Kain, "Advanced Computer Architecture: A Systems Design Approach". Prentice-Hall, 1996.

  
all ms  
Fernando Díaz

  
Sergio Vento Díaz