

ENSEÑANZA SOBRE ARQUITECTURAS AVANZADAS. UNA EXPERIENCIA: ARQUITECTURA E INGENIERÍA DE COMPUTADORES

Germán Fabregat Lluca

Área de Arquitectura y Tecnología de Computadores
Departamento de Informática
Universidad Jaume I, Castellón
e-mail: fabregat@inf.uji.es

La asignatura Arquitectura y Tecnología de Computadores es la más representativa del área de ATC en el segundo ciclo de Ingeniería Informática en la Universidad Jaume I. La estructura del plan de estudios, la poca experiencia previa y la novedad y complejidad de sus contenidos le confieren una problemática especial que se analiza en el presente documento.

INTRODUCCIÓN

El año académico 1995/96, con la llegada de la primera promoción de estudiantes de Ingeniería Informática de la Universidad Jaume I al quinto (y último) curso de la carrera, comenzó a impartirse la asignatura *Arquitectura e Ingeniería de Computadores*. Actualmente, con apenas dos años de experiencia, se han detectado ya una serie de problemas típicos de la asignatura. El contraste de una primera promoción formada por un número reducido de alumnos con una preparación y capacidad por encima de la media, con la segunda de características más normales, ha puesto de manifiesto un nivel excesivo en las exigencias o complejidad de los contenidos de la asignatura y una falta de base previa en el alumnado que deben ser tenidos en consideración al impartir la asignatura en años sucesivos. Este documento pretende exponer la experiencia de estos dos años, describiendo los problemas más significativos y apuntando algunas soluciones.

En los dos apartados siguientes se describen la organización y contenidos de la asignatura y el desarrollo de las prácticas. Posteriormente se describen y analizan los problemas más importantes, intentando determinar sus causas y proponer soluciones. El trabajo se concluye exponiendo cómo se pretende modificar la asignatura para años sucesivos. Se incluye al final una lista de la bibliografía utilizada en la asignatura y el temario detallado.

CONTENIDO DE LA ASIGNATURA

Arquitectura e Ingeniería de Computadores (AIC) es una asignatura troncal de segundo ciclo en Ingeniería Informática. En la Universidad Jaume I se imparte en el último (quinto) curso de la titulación, con carácter anual. La carga docente es de 9 créditos, divididos en 4 teóricos, 3 prácticos

(de problemas) y 2 de laboratorio. Las clases teóricas y de problemas se imparten a razón de 2 horas por semana durante las 30 semanas (aproximadamente) de duración del curso. Las prácticas de laboratorio se organizan en 7 sesiones de 3 horas que se distribuyen alternando durante las 15 semanas del segundo cuatrimestre. Los 10 créditos de déficit entre el horario oficial de la asignatura y los créditos de la misma se dedican a la introducción teórica de las prácticas, y se organizan en sesiones de dos o tres horas al final del primer cuatrimestre.

AIC puede ser considerada la asignatura más importante del área de Arquitectura y Tecnología de Computadores en el segundo ciclo de la Ingeniería, pues en ella se vuelven a tratar conceptos generales sobre arquitectura de computadores. Habiéndose explicado los conceptos básicos (sistemas digitales, arquitectura Von Neumann, sistemas operativos...) a lo largo del primer ciclo, la asignatura los asume como punto de partida y se centra en lo que serían arquitecturas avanzadas o arquitecturas de altas prestaciones [1]. A continuación se indican los capítulos que componen el temario de AIC, que aparece detallado en el Anexo I.

1.-Introducción, revisión y presentación de conceptos.

PARTE I. Monoprocesadores. 2.-Segmentación encauzada (pipelines).

3.-Paralelismo a nivel de instrucción.

4.-Procesadores vectoriales y matriciales.

PARTE II. Multiprocesadores. 5.-Sistemas multiprocesadores.

El primer tema es introductorio. El curso comienza con una reflexión sobre los avances en tecnología de fabricación de circuitos integrados y su repercusión en el aumento de prestaciones de los sistemas, incluyendo la posibilidad de incorporar en los procesadores de propósito general mecanismos reservados anteriormente a los supercomputadores [2]. A lo largo del curso se hacen algunos comentarios más al respecto de la tecnología, siempre tomando esta breve presentación como punto de referencia. El resto del tema se dedica a presentar el contenido de la asignatura y a repasar conceptos fundamentales. En particular se describe la clasificación de las unidades centrales de proceso adoptada en [3], y se explica la distinción clásica entre arquitecturas CISC y RISC [4].

El tema 2 se dedica completamente a la segmentación encauzada. Tras la presentación de los principios básicos, según [5], se sigue el enfoque de [3] describiendo los pipelines típicos de las máquinas RISC. Pese a que en un principio se eligió la arquitectura DLX presentada en [3], este último año se ha preferido utilizar un ejemplo real, describiendo el pipeline del procesador R3000 [6]. Se estudian con especial detalle las dependencias de datos analizando tipos e influencia en la ejecución de programas, y la gestión de excepciones. Con respecto a estos temas, se realizan problemas de organización de pipelines tratando de eliminar detenciones evitables y de optimización de código para máquinas segmentadas.

El estudio de pipelines más complejos, con etapas iterativas, se desarrolla según [1] (pipelines de máxima prestaciones) y [3] (unidad en coma flotante), mientras que las unidades de acceso a memoria y las técnicas de detección y gestión de dependencias son descritas de acuerdo a [7].

Aunque estrictamente, al hablar de paralelismo a nivel de instrucción, se deberían incluir tanto las máquinas escalares segmentadas como las superescalares y VLIW [3], [7], el tema tercero, *Paralelismo a nivel de instrucción*, trata exclusivamente estas dos últimas organizaciones. Se sigue fundamentalmente la referencia [7] ya que trata de forma completa y sencilla la mayor parte de los

apartados del tema. Se dedica mayor atención a los aspectos de la planificación dinámica, en donde la bibliografía se completa con [8]. Se trata también en detalle el tema de la predicción de saltos, comentado de forma menos extensa en el tema anterior, donde además de ejemplos de algunas arquitecturas actuales (obtenidos de muy diversas fuentes) se utiliza la referencia [9]. Por último, se describen brevemente las técnicas más simples de optimización del código para máquinas segmentadas, superescalares y procesadores VLIW, reforzando la bibliografía con [10].

Con respecto a este tema, se desarrollan problemas de trazas de ejecución con planificación dinámica y predicción de saltos, gestión de excepciones precisas con ciertos algoritmos de planificación, etcétera.

El tema cuarto se dedica, en el temario, a tratar los procesadores vectoriales y matriciales. La documentación seguida es fundamentalmente [3] aunque, como se comentará posteriormente, los problemas de adecuación del temario al tiempo disponible hacen que en realidad solamente se haya hecho un comentario muy breve, por lo que el capítulo está en la práctica sin desarrollar.

La segunda parte de la asignatura, dedicada íntegramente a sistemas multiprocesadores, está compuesta paradójicamente por un único tema, aunque comprende casi la mitad del curso. Se comienza con la descripción de varias posibles clasificaciones para los sistemas multiprocesador. En este punto se sigue [5] y algunas otras fuentes más o menos vagas. Lo mismo ocurre con el apartado acerca de redes de interconexión. Aunque se han consultado múltiples referencias de la bibliografía disponible [1], [5], [11], [12] no se ha encontrado ninguna suficientemente completa, por lo que se ha tenido que recurrir a una clasificación sugerida por el propio profesorado. Una parte importante del tema se centra en los conceptos de coherencia de los sistemas de memoria. En este apartado se realiza una introducción a la gestión de las memorias caché en monoprocesadores basada en [5] que se completa con apuntes de elaboración propia, para continuar con una descripción de los mecanismos basados en snooping del bus, directorios y software. La bibliografía seguida [13], [14], [15] es suficiente, y de alta calidad. El tema se completa con una extensión del problema de la coherencia de la memoria a los modelos de comportamiento, describiendo los modelos presentados en [11] y la metodología de caracterización de arquitecturas de [16]. Por último, se continúa con la referencia [11] para comentar ciertos aspectos sobre sistemas operativos y programación en multiprocesadores. Se estudian fundamentalmente los sistemas operativos basados en microkernel y el concepto de multithreading.

En relación con el tema cinco, se llevan a cabo problemas de realización de sencillos protocolos de coherencia, gestión de memoria virtual y confrontación de ejecuciones con arquitecturas, tratando de detectar violaciones del modelo de comportamiento.

REALIZACIÓN DE LAS PRÁCTICAS

El desarrollo de las prácticas de la asignatura presenta ciertos problemas, debido fundamentalmente a la falta de los medios idóneos que se tratarán en detalle en el apartado siguiente. Las soluciones elegidas han pasado por aprovechar intensivamente los medios disponibles y recurrir a la simulación en aquellos temas para los cuales no existen medios adecuados.

Se pretende, lógicamente, que las clases prácticas refuercen los conceptos explicados en teoría. Por lo tanto, realizar prácticas de análisis de los pipelines de ciertas organizaciones de procesadores no requiere de más conocimientos teóricos que los expuestos en el tema 2. Sin embargo, el único procesador RISC disponible para realizar prácticas con comodidad es el PA RISC de la estación de trabajo de la serie 700 de Hewlett Packard. Por ello, las prácticas requieren de unas clases previas de descripción de la arquitectura del PA RISC [17] y del entorno de trabajo.

Las prácticas consisten en, midiendo tiempos de ejecución de secuencias de instrucciones, extraer conclusiones acerca de la organización del pipeline del procesador. Para ello, se realizan programas en lenguaje C que llaman a rutinas en ensamblador que contiene las instrucciones problema. Desde el programa principal se mide el tiempo de ejecución y se realizan las comparaciones con un tiempo patrón calculado mediante la ejecución de instrucciones de ALU sin dependencias.

Las sesiones comienzan con una práctica introductoria de programación en ensamblador para conseguir la familiarización con el sistema, y posteriormente se estudian las instrucciones de la ALU, los accesos a memoria y los diversos saltos del conjunto de instrucciones. Las practicas finalizan con el estudio de ciertas instrucciones de control y de los permisos de acceso a memoria desde los programas de usuario.

Durante el curso pasado, algunos grupos de prácticas realizaron, para solucionar ciertos problemas sobre memorias caché planteados en las clases de pizarra, pequeños simuladores capaces de analizar tiempos de ejecución de sencillos programas que gestionaran matrices. Este año las prácticas basadas en simulación se han desestimado, pues la carga de trabajo que las anteriormente comentadas ha supuesto ocupa totalmente el tiempo disponible.

PROBLEMÁTICA DE LA DOCENCIA

Una asignatura como AIC, con los contenidos que se han descrito, plantea diversos tipos de problemas a la hora de impartirla. Se ha intentado clasificar estos problemas según su causa. A continuación se comentan estos tipos, con sus posibles causas y soluciones.

Lagunas en la formación del alumnado.

Los alumnos que llegan a AIC han cursado las asignaturas básicas sobre arquitectura de computadores en el primer ciclo, con un total de 27,5 créditos entre troncales y obligatorios y 17,5 optativos. En estas asignaturas se explican los fundamentos sobre sistemas digitales, organización de computadores y arquitectura Von Neumann, tecnología de ordenadores y sistemas operativos.

Cuando los estudiantes llegan a cursar AIC se encuentran con una asignatura sobre arquitecturas avanzadas que asume una buena base en los temas antes mencionados, requiriendo relacionar con claridad y soltura los conceptos fundamentales. En la práctica, el alumno no domina buena parte de los conocimientos necesarios para comprender con claridad la asignatura, bien por olvido bien porque en su día no fueron tratados, al menos en la profundidad necesaria. Esta falta de conocimientos básicos es más aguda sobre aspectos tecnológicos, aun que también alcanza ciertos temas puntuales de arquitectura como jerarquía de memoria (memorias caché, uso de TLBs), gestión de excepciones, protocolos de buses, etcétera.

La forma más natural de solucionar estas deficiencias es, lógicamente, detenerse a explicar aquellos conceptos que en la práctica no son conocidos por los alumnos. En otros temas, donde se tiene una formación incompleta o vaga, se indica la necesidad de profundizar para poder seguir la asignatura. Por último, se procura explicar algunos ejemplos aplicados al hilo de los conceptos teóricos haciendo comentarios para mostrar cómo se tienen en cuenta ciertos aspectos prácticos que ayuden a aclarar ideas.

Complejidad y extensión de la asignatura.

AIC, por su temario y número de créditos, es una asignatura larga, que incluye una gran diversidad de conceptos. Muchos de ellos, además, por su complejidad en relación a la formación de los alumnos (que parece más orientada hacia la programación) se hacen particularmente difíciles al alumnado, que afronta la asignatura con cierto miedo desde el inicio.

Hasta el momento, la única forma de tratar la falta de seguridad de los estudiantes frente a la asignatura ha sido dedicar tiempo hacia el final del curso para repasar los conceptos fundamentales del temario, mediante la realización de problemas y la explicación de aquellos apartados que pudieran haber quedado poco claros a lo largo del curso. Esta práctica obliga a recortar el temario establecido a priori, no impartiendo el tema 4 por parecer el menos importante.

Falta de bibliografía adecuada y estable.

En AIC se intenta dar una visión de las arquitecturas avanzadas que incorpore los cambios que día a día se van produciendo tanto a nivel académico-científico como comercial. Con esta dinámica, es difícil encontrar una bibliografía estable pues muchos temas cambian o incorporan novedades constantemente (predicción de saltos, gestión de memorias caché, coherencia de memoria) mientras que sobre otros todavía no hay unos conceptos establecidos de común acuerdo entre la comunidad científica (clasificación de multiprocesadores y topologías, sistemas operativos para multiprocesadores). Algunos puntos del temario deben ser completados entonces sin el respaldo de una buena bibliografía, al albedrío del profesorado de la asignatura.

En la actualidad, de algunos puntos que se renuevan constantemente se busca información en revistas científicas o en grupos de debate en internet (comp.arch, por ejemplo). Para otras carencias, se ha pensado buscar ayuda en la comunidad docente a nivel estatal, por ejemplo asistiendo a jornadas como las presentes JENUI97.

Orientación de la formación del profesorado.

Los temas que podrían ser incluidos en una asignatura sobre arquitecturas avanzadas, y fundamentalmente el enfoque y la extensión de cada uno de ellos, se ven necesaria aunque involuntariamente influidos por la actividad investigadora del profesorado que imparte la asignatura, de una forma más aguda que otras asignaturas de base más establecida y menos dinámica. En el caso de la asignatura objeto del presente artículo, la investigación del profesorado se orienta, en el campo de los multiprocesadores, hacia la coherencia de memoria y los modelos de comportamiento.

La única solución de este problema, por su subjetividad, pasa por un debate sobre los contenidos de la asignatura que se espera comience en el marco de JENUI97.

Falta de medios para las prácticas de laboratorio.

AIC, como se ha dicho antes, posee un temario extenso y diverso. Realizar actividades prácticas que permitan a todos los alumnos reforzar la mayor parte de los capítulos del temario parece poco viable. Preparar prácticas diferentes para que, al menos por grupos, se cubra todo el contenido de la asignatura requiere una diversidad de medios que, en muchos casos, son de elevado coste y difíciles de conseguir.

Hasta el momento, las prácticas se han centrado en un único aspecto de la asignatura: análisis del pipeline de procesadores RISC segmentados. Para el futuro se intentará utilizar un multiprocesador (diseñado y construido en la propia Universidad Jaume I como resultado de un proyecto) basado en controladores RISC de la familia del R3000. Otra iniciativa más ambiciosa, propuesta como proyecto final de carrera, incluye el diseño e implementación del UJI RISC, una unidad central de proceso RISC que permita, en un futuro, disponer de una base bien conocida y fácilmente modificable para poder realizar un conjunto más completo de prácticas.

Contraste entre promociones de estudiantes.

Un último problema, coyuntural debido a la poca experiencia de AIC en la Universidad Jaume I, es el contraste entre promociones. La primera, con muy pocos alumnos, estaba formada por aquellos estudiantes que habían seguido año a año una carrera en formación en una universidad en formación.

No es por tanto de extrañar que su capacidad fuera más alta de lo normal. La segunda promoción, más numerosa y menos preparada, ha puesto de manifiesto la dificultad de la asignatura.

Se espera, con el paso de los años, ir adecuando los contenidos y el nivel de exigencia de AIC al tiempo de que dispone el alumnado para dedicar a su estudio.

CONCLUSIONES

Se ha presentado la asignatura *Arquitectura e Ingeniería de Computadores*, con su contenido y los problemas hallados durante su experiencia docente. La asignatura debe estar en proceso de evolución y mejora constante, y el presente trabajo y la discusión que esperamos suscite debe contribuir a ello. Este ha sido al menos nuestro objetivo.

BIBLIOGRAFÍA

- [1] H. S. Stone. *High Performance Computer Architecture*. Addison-Wesley. 1993.
- [2] J. L. Hennessy y N. P. Jouppi. *Computer Technology and Architecture: An Evolving Interaction*. IEEE Computer. Septiembre 1991.
- [3] J. L. Hennessy y D. A. Patterson. *Arquitectura de Computadores. Un Enfoque Cuantitativo*. McGraw-Hill. 1993.
- [4] W. Stallings. *Reduced Instruction Set Computers*. IEEE Computer Society Press. 1990.
- [5] K. Hwang y F. A. Briggs. *Arquitectura de Computadoras y Procesamiento Paralelo*. McGraw-Hill. 1987.
- [6] G. Kane y J. Heinrich. *MIPS RISC Architecture*. Prentice Hall. 1992.
- [7] M. Johnson. *Superescalar Microprocessor Design*. Prentice-Hall. 1991.
- [8] R. M. Tomasulo. *An Efficient Algorithm for Exploiting Multiple Arithmetic Units*. IBM Journal of Research and Development. Enero 1967.
- [9] H. G. Cragon. *Branch Strategy Taxonomy and Performance Models*. IEEE Computer Society Press. 1992.
- [10] K. Dowd. *High Performance Computing*. O'Reilly & Associates, Inc. 1993.
- [11] A. S. Tanenbaum. *Distributed Operating Systems*. Prentice-Hall. 1995.
- [12] P. Tvrđik. *ISIPCALA '93 Drafts of Papers*. International Summer Institute on Parallel Computer Architectures, Lenguajes and Algorithms. 1993.
- [13] J. Handy. *The Cache Memory Book*. Academic Press. 1993.
- [14] M. Tomasevic y V. Milutinovic. *The Cache Coherence Problem in Shared Memory Multiprocessors: Hardware Solutions*. IEEE Computer Society Press. 1993.
- [15] M. Dubois y S. S. Thakkar. *Cache and Interconnect Architectures in Multiprocessors*. Kluwer Academic Publishers. 1990.
- [16] W. W. Collier. *Reasoning About Parallel Architectures*. Prentice-Hall. 1992.
- [17] *PA RISC 1.1 Architecture and Instruction Set Reference Manual*. Hewlett Packard. 1994.

ANEXO I. TEMARIO DE LA ASIGNATURA

Tema 1. Introducción, revisión y presentación de conceptos.

- 1.1. Aumento de prestaciones. Tecnología y arquitectura.
- 1.2. Revisión de conceptos.
 - 1.2.1. Arquitectura Von Neumann.
 - 1.2.2. Jerarquía de memoria.
 - 1.2.3. Subsistema de entrada/salida.
 - 1.2.4. Buses y redes de interconexión.
 - 1.2.5. Sistema operativo.
- 1.3. Arquitecturas monoprocesador.
 - 1.3.1. Arquitecturas no Von Neumann.
 - 1.3.2. Máquinas de pila, de registros y de memoria.
 - 1.3.3. Procesadores CISC y RISC.
- 1.4. Paralelismo en monoprocesadores.
 - 1.4.1. Coprocesadores.
 - 1.4.2. Procesadores vectoriales y matriciales.
 - 1.4.3. Procesadores superescalares y VLIW.
- 1.5. Arquitecturas multiprocesador.
 - 1.5.1. Clasificación de los multiprocesadores.
 - 1.5.2. Memoria y redes de interconexión.
 - 1.5.3. Memoria compartida y distribuida.
 - 1.5.4. Sistemas fuertemente acoplados, débilmente acoplados y distribuidos.

Tema 2. Segmentación encauzada (pipelines).

- 2.1. Principios básicos de la segmentación encauzada.
- 2.2. Ejecución de instrucciones.
 - 2.2.1. Pipeline fundamental para enteros.
 - 2.2.2. Riesgos: detenciones, desplazamientos y vaciados.
 - 2.2.2.1. Bifurcaciones.
 - 2.2.2.2. Dependencias estructurales.
 - 2.2.2.3. Dependencias de datos.
 - 2.2.3. Excepciones precisas e imprecisas.
- 2.3. Pipelines con etapas iterativas.
 - 2.3.1. Pipeline para operaciones en punto flotante.
 - 2.3.2. Ejecución solapada de instrucciones.
 - 2.3.2.1. Dependencias estructurales.
 - 2.3.2.2. Dependencias de datos.
 - 2.3.3. Excepciones.
- 2.4. Accesos a memoria.

Tema 3. Paralelismo a nivel de instrucción.

- 3.1. Explotación del paralelismo a nivel de instrucción.
 - 3.1.1. Identificación de paralelismo en bloques y fuera de ellos.
 - 3.1.2. Organizaciones para explotar el paralelismo.
 - 3.1.2.1. Superpipelines.
 - 3.1.2.2. Procesadores VLIW.
 - 3.1.2.3. Procesadores superescalares.

- 3.2. **Ordenación de las instrucciones.**
 - 3.2.1. Ejecución fuera de orden y ejecución especulativa.
 - 3.2.2. Algoritmos para la planificación dinámica.
 - 3.2.3. Dependencias de datos.
 - 3.2.4. Predicción de saltos.
 - 3.2.5. Tratamiento de las excepciones.
- 3.3. Técnicas de optimización del código.
 - 3.3.1. Desdoblamiento de bucles.
 - 3.3.2. Segmentación del software.
 - 3.3.3. Planificación basada en trazas.

Tema 4. Procesadores vectoriales y matriciales.

- 4.1. Aplicaciones de cálculo intensivo.
- 4.2. Procesadores vectoriales.
 - 4.2.1. Estructura interna.
 - 4.2.2. Accesos a memoria.
 - 4.2.3. Unidades funcionales.
- 4.3. Procesadores matriciales.
 - 4.3.1. Organización interna.
 - 4.3.2. Clasificación.

Tema 5. Sistemas multiprocesadores.

- 5.1. Introducción.
 - 5.1.1. Sistemas monoprocesador y sistemas multiprocesador.
 - 5.1.2. Clasificación de los sistemas MIMD.
- 5.2. Redes de interconexión.
 - 5.2.1. Criterios y clasificación.
 - 5.2.2. Redes de comunicación paralela y buses.
 - 5.2.3. Redes de comunicación serie.
- 5.3. Jerarquía de memoria.
 - 5.3.1. Problemática de los subsistemas de memoria.
 - 5.3.1.1. Memoria compartida y distribuida.
 - 5.3.1.2. Compartición virtual y lógica.
 - 5.3.2. Mecanismos de coherencia de las memorias caché.
 - 5.3.2.1. Snooping del bus.
 - 5.3.2.2. Mecanismos basados en directorio.
 - 5.3.2.3. Soluciones software.
 - 5.3.3. Memoria virtual.
- 5.4. Modelos de comportamiento.
 - 5.4.1. Introducción.
 - 5.4.2. Clasificación y ejemplos.
- 5.5. Otros aspectos de los multiprocesadores.
 - 5.5.1. Subsistemas de Entrada/Salida.
 - 5.5.2. Sistemas operativos.
 - 5.5.3. Programación.