

GRADO

GUÍA DE ESTUDIO

INGENIERÍA DE COMPUTADORES II

2ª PARTE | PLAN DE TRABAJO Y ORIENTACIONES PARA SU DESARROLLO

Sebastián Dormido Canto
David Moreno Salinas
José Sánchez Moreno
Victorino Sanz Prat

1.- PLAN DE TRABAJO

La metodología prevista para esta asignatura incluye: estudio de contenidos teóricos utilizando la bibliografía básica de la asignatura, trabajo autónomo con los problemas propuestos en el texto básico, pruebas de autoevaluación (PAs) y realización de una prueba de evaluación a distancia (PED).

Los contenidos de esta asignatura se organizan en cuatro temas:

- Tema 1: Procesadores segmentados.
- Tema 2: Procesadores superescalares.
- Tema 3: Procesadores VLIW y procesadores vectoriales.
- Tema 4: Procesamiento paralelo.

El plan de trabajo de cada tema consiste en el estudio de un material teórico (incluido en el libro base) y en la realización de unos ejercicios prácticos.

Como ya se comentó en el apartado de evaluación en la primera parte de esta guía, los ejercicios prácticos son de dos tipos: PAs y PED. Ambos tipos de ejercicios serán publicados en el curso virtual. Las PAs no tendrán influencia en la calificación final de la asignatura y la PED ponderará un 10% en la calificación final (consultar el apartado “Evaluación” de la primera parte de la guía para más detalles).

A continuación, se presenta a modo de orientación una planificación para cada semana del curso indicando las actividades previstas, una estimación de las horas de trabajo necesarias y los resultados de aprendizaje esperados. En relación a las horas de trabajo, las hemos dividido en horas de estudio y horas de prácticas. Las horas de estudio se refieren al tiempo necesario que el alumno debe dedicar a estudiar los materiales teóricos de la asignatura. Las horas de prácticas se refieren al tiempo necesario que el alumno debe dedicar a realizar actividades prácticas (resolución de problemas del texto base, PAs y PED).

SEMANA	ACTIVIDAD	HORAS DE ESTUDIO	HORAS PRÁCTICAS	RESULTADOS DE APRENDIZAJE*
Semanas 1 y 2	Estudio Tema 1 Actividades Tema 1	12	12	RA1.1 a RA1.7 RG1 y RG2
Semanas 3, 4, 5 y 6	Estudio Tema 2 Actividades Tema 2	25	25	RA2.1 a RA2.11 RG1 y RG2
Semanas 7, 8 y 9	Estudio Tema 3 Actividades Tema 3	18	18	RA3.1 a RA3.10 RG1 y RG2
Semanas 10 y 11	Estudio Tema 4 Actividades Tema 4	10	10	RA4.1 a RA4.5 RG1 y RG2
Semanas 12 y 13	Preparación de examen	10	10	

* Los resultados de aprendizaje se explican en la siguiente sección

Prueba de Evaluación a Distancia (PED)

Esta prueba constará de uno o varios ejercicios teórico-prácticos o de programación con un simulador didáctico de un procesador segmentado y serán evaluados por el profesor-tutor y/o por el equipo docente. La prueba no es obligatoria y tiene una ponderación de un 10% en la calificación final. No es necesario acudir a un Centro Asociado para la realización de la prueba de evaluación a distancia. No se admitirán entregas de la PED fuera del plazo indicado por el equipo docente, y que estará enmarcado dentro del cuatrimestre en que se cursa la asignatura.

Pruebas de Autoevaluación (PA)

Estas pruebas estarán disponibles en el curso virtual según el calendario propuesto en el plan de trabajo. No tienen carácter obligatorio ni son evaluadas por el equipo docente pero sirven al alumno para reforzar su aprendizaje.

2.- ORIENTACIONES PARA EL ESTUDIO DE LOS CONTENIDOS

En este epígrafe se presentan unas orientaciones para el estudio de la asignatura de “Ingeniería de Computadores II” con la finalidad de servir de guía en el desarrollo del curso.

Para cada tema se presentan sus contenidos (más detallados en el índice del texto base de la asignatura), una introducción sobre los contenidos del capítulo, así como una descripción de los resultados de aprendizaje más significativos.

Hay ciertos resultados de aprendizaje que son comunes a todos los temas, tales como:

1. Saber identificar los componentes de un computador moderno (RG1).
2. Entender las razones y estrategias para la existencia de diferentes tipos de arquitectura de computador (RG2).

TEMA 1: PROCESADORES SEGMENTADOS

Contenidos

- ⤴ Diferencias entre procesadores RISC y procesadores CISC.
- ⤴ Clasificación y características generales de las arquitecturas paralelas.
- ⤴ Medidas para evaluar el rendimiento de un computador.
- ⤴ Características de los procesadores segmentados.
- ⤴ Descripción de una arquitectura segmentada genérica.
- ⤴ Tipos de riesgos presentes en la segmentación.
- ⤴ Técnicas que evitan los riesgos en la segmentación.
- ⤴ El algoritmo de Tomasulo como técnica de planificación dinámica en la segmentación.

Introducción

En este tema se estudia el concepto de segmentación, pieza básica y fundamental sobre la que se articula el diseño de los procesadores de las últimas décadas, con independencia de su filosofía arquitectónica. La segmentación constituye una técnica de implementación de procesadores mediante la cual se solapa la ejecución de múltiples instrucciones, explotando el paralelismo entre las instrucciones que componen un programa y permitiendo descomponer la ejecución de una instrucción en un número fijo de etapas donde cada una representa una fase del procesamiento de la instrucción. El disponer de varias etapas permite que

en cada una de ellas haya una instrucción en un estadio diferente de su procesamiento y, por lo tanto, en un instante dado puedan llegar a estar ejecutándose en el procesador tantas instrucciones como etapas. Cada etapa se encuentra conectada con la anterior formando un cauce (o *pipeline*) por cuyo extremo entra la instrucción, se procesa en las sucesivas etapas de la segmentación o segmentos y, concluida la ejecución, sale por el otro extremo. Un símil muy utilizado para visualizar de forma rápida el concepto de segmentación es el de las cadenas de fabricación de automóviles. Evidentemente, la segmentación no está carente de problemas siendo los más habituales las interdependencias que existen entre las instrucciones en materia de operandos. En este tema no solo se estudia el concepto de segmentación, utilizando para ello una segmentación genérica básica pero reflejo de las reales, sino que se analizan en detalle los problemas que surgen y se estudian algunas de las soluciones adoptadas.

El Capítulo 1 del texto base de la asignatura cubre completamente todos los contenidos de este tema.

Resultados de aprendizaje

- RA 1.1. Saber las diferencias entre los procesadores RISC y los procesadores CISC.
- RA 1.2. Clasificar y conocer las características generales de las arquitecturas paralelas.
- RA 1.3. Conocer y saber utilizar las medidas para evaluar el rendimiento de un computador.
- RA 1.4. Conocer las características de los procesadores segmentados.
- RA 1.5. Conocer y saber analizar los distintos tipos de riesgos presentes en la segmentación.
- RA 1.6. Conocer las técnicas que evitan los riesgos en la segmentación.
- RA 1.7. Entender el algoritmo de Tomasulo como técnica de planificación dinámica.

TEMA 2: PROCESADORES SUPERESCALARES

Contenidos

- ⤴ Características y arquitectura genérica de un procesador superescalar.
- ⤴ Problemática de la ejecución de instrucciones fuera de orden.
- ⤴ Técnicas de prelectura y lectura de instrucciones para mejorar el ancho de banda.
- ⤴ Técnicas dinámicas para la predicción del resultado y dirección de las instrucciones de salto.
- ⤴ Técnicas de predecodificación, decodificación y traducción de instrucciones.
- ⤴ Distribución de instrucciones con y sin lectura de operandos.
- ⤴ Buffer de distribución. Estaciones de reserva centralizadas, individuales y compartidas.
- ⤴ Renombramiento de registros. Fichero de registros arquitectónicos. Fichero de registro de renombramiento.
- ⤴ Emisión y finalización de instrucciones fuera de orden.
- ⤴ Gestión de dependencias falsas de datos y de memoria.
- ⤴ Buffer de terminación. Mantenimiento de la consistencia del procesador.
- ⤴ Buffer de almacenamiento y de cargas. Mantenimiento de la consistencia de la memoria.
- ⤴ Tratamiento de interrupciones. Interrupciones precisas mediante buffer de historia, fichero de registros de futuro y buffer de terminación.
- ⤴ Limitaciones de los procesadores superescalares.

Introducción

La mayoría de los procesadores que se utilizan en los computadores actuales, ya estén orientados al mercado doméstico como al científico o al industrial, se basan en segmentaciones capaces de procesar varias instrucciones simultáneamente y de forma desordenada, de ahí su denominación de procesadores superescalares. En este tema se estudia la arquitectura de un procesador superescalar genérico, muy similar a la de los procesadores comerciales aunque más sencilla debido a su carácter didáctico. Recurriendo a esta arquitectura superescalar, se analiza cada una de las etapas que intervienen en el procesamiento de los

diferentes tipos de instrucciones (carga/almacenamiento, aritmético-lógicas, saltos, bifurcaciones) y se estudian las soluciones dadas por los investigadores y la industria a los problemas que se plantean en cada una de ellas. Además, se analizan las limitaciones que presentan este tipo de arquitecturas de procesador para crear computadores cada vez más potentes.

El Capítulo 2 y el Apéndice A del texto base de la asignatura cubren completamente todos los contenidos de este tema.

Resultados de aprendizaje

RA 2.1. Conocer las características de los procesadores superescalares y sus diferencias con respecto a una segmentación escalar.

RA 2.2. Entender la problemática que plantean las etapas de una segmentación superescalar.

RA 2.3. Saber qué son el *front-end*, el núcleo de ejecución fuera de orden y el *back-end* de un procesador superescalar.

RA 2.4. Conocer las técnicas dinámicas más utilizadas para la especulación del resultado y la dirección de destino de una instrucción de salto.

RA 2.5. Conocer técnicas que mejoran el ancho de banda del procesador, como, la prelectura, la predecodificación y la traducción de instrucciones.

RA 2.6. Comprender la finalidad del renombramiento de registros y conocer las diferentes formas que existen para incorporar esta técnica en un procesador superescalar.

RA 2.7. Entender la relevancia de las etapas de distribución y terminación.

RA 2.8. Comprender las técnicas para la resolución de las dependencias falsas de memoria.

RA 2.9. Entender cómo se mejora el procesamiento de las instrucciones de carga y almacenamiento.

RA 2.10. Conocer cómo se consigue que un procesador superescalar tenga precisión de excepción.

RA 2.11. Conocer otras técnicas para mantener interrupciones precisas.

TEMA 3: PROCESADORES VLIW Y PROCESADORES VECTORIALES

Contenidos

- ⤴ Características y arquitectura de un procesador VLIW (*Very Long Instruction Word*).
- ⤴ Ventajas e inconvenientes con respecto a los procesadores superescalares.
- ⤴ Evolución histórica del concepto VLIW.
- ⤴ Técnicas de planificación estática: Desenrollamiento de bucles, segmentación software y planificación de trazas.
- ⤴ El estilo arquitectónico EPIC (*Explicitly Parallel Instruction Computing*): Conceptos y características básicas.
- ⤴ Características de los procesadores vectoriales.
- ⤴ Arquitectura básica de un procesador vectorial genérico. Procesador matricial. Diferencias básicas con un procesador vectorial.
- ⤴ Características de las unidades vectoriales aritméticas. Registros vectoriales. Unidades funcionales con múltiples carriles.
- ⤴ Repertorio de instrucciones vectoriales. Instrucciones vectoriales aritméticas. Instrucciones vectoriales de carga/almacenamiento clásicas y con separación. Operaciones con máscara. Registro de longitud vectorial.
- ⤴ Vectorización de bucles de longitud desconocida mediante seccionamiento.
- ⤴ Manipulación de vectores almacenados no consecutivamente en memoria.
- ⤴ Vectorización de bucles con instrucciones condicionales.
- ⤴ Medida del rendimiento de un fragmento de código vectorial.
- ⤴ Características de una unidad de carga/almacenamiento vectorial.
- ⤴ Estimación del rendimiento de un bucle vectorizado.

Introducción

En este tema se introducen dos nuevas filosofías para el diseño de procesadores que, aunque basadas en la segmentación, introducen formas alternativas para mejorar el rendimiento: los procesadores VLIW y los procesadores vectoriales. Los procesadores VLIW se caracterizan por utilizar internamente instrucciones compuestas de varias instrucciones u operaciones básicas de forma que se maximiza el uso de las múltiples unidades funcionales que poseen. Este aspecto pone de manifiesto la gran importancia que tiene el diseño de compiladores para procesadores VLIW que tengan en cuenta la arquitectura interna del procesador y generan código objeto acorde con sus características. Esto pone de relieve los inconvenientes de esta filosofía de diseño: la necesidad de disponer de compiladores altamente especializados y los problemas de incompatibilidad ya que el código objeto que se genera para un procesador VLIW tiene en cuenta las latencias y el número de unidades funcionales de ese procesador lo que provoca que sea ineficiente si se pretende ejecutar en otro procesador VLIW con características diferentes.

Por otro lado, los procesadores vectoriales se caracterizan por proporcionar operadores de alto nivel que trabajan sobre vectores, no sobre valores escalares. Así, una operación vectorial tomaría dos vectores de n elementos, realizaría la correspondiente operación aritmética con ellos y generaría como resultado un vector de n elementos. Por lo tanto, una instrucción vectorial es equivalente a un bucle constituido por instrucciones secuenciales, donde en cada iteración se realiza una operación aritmética y se genera uno de los elementos del resultado. Aunque los computadores basados en procesadores vectoriales tuvieron su máximo apogeo en la década de los 80 como punta de lanza de la supercomputación científica, y habiendo sido superados hoy en día por los multiprocesadores y multicomputadores, paradójicamente actualmente se encuentran más accesibles que nunca. Ello se debe a que los actuales procesadores superescalares incorporan en su repertorio de instrucciones un conjunto de instrucciones vectoriales, conocidas como instrucciones SIMD, que permiten mejorar el rendimiento de las aplicaciones multimedia por lo que todos los procesadores incorporan una o dos unidades funcionales específicas para el tratamiento de las instrucciones SIMD.

El Capítulo 3 del texto base de la asignatura cubre todos los contenidos de este tema.

Resultados de aprendizaje

RA 3.1. Conocer otras arquitecturas de computador distintas al concepto de segmentación superescalar.

RA 3.2. Entender otras formas de explotar el paralelismo a nivel de instrucción, como son las aproximaciones VLIW y EPIC.

RA 3.3. Conocer las características de un procesador VLIW junto con sus ventajas, sus inconvenientes, sus orígenes y su proyección de futuro.

RA 3.4. Conocer las técnicas de planificación estática que se utilizan para evitar la problemática que conlleva el procesamiento VLIW: Desenrollamiento de bucles, segmentación software y planificación de trazas.

RA 3.5. Entender las aportaciones del concepto EPIC para evitar los inconvenientes de los procesadores VLIW.

RA 3.6. Conocer las características básicas de los procesadores vectoriales, su evolución y sus perspectivas de futuro.

RA 3.7. Entender las diferencias entre procesadores vectoriales y matriciales.

RA 3.8. Comprender cómo funcionan las unidades vectoriales aritmético-lógicas y de acceso a memoria para poder conseguir un flujo de datos muy elevado.

RA 3.9. Conocer un repertorio genérico de instrucciones vectoriales y las técnicas que se utilizan para resolver determinados inconvenientes como son el almacenamiento de datos en memoria con

separación superior a la unidad, los bucles con instrucciones ejecutadas condicionalmente o el procesamiento de vectores con una longitud superior a la de los registros vectoriales.

RA 3.10. Saber cómo calcular el tiempo de ejecución de un conjunto de instrucciones vectoriales y de un bucle escalar vectorizado mediante la técnica de seccionamiento.

TEMA 4: PROCESAMIENTO PARALELO

Contenidos

- ▲ Organización y principales características de las diferentes plataformas de computación paralela.
- ▲ Paradigmas de programación paralela, detallando dos casos: cliente/servidor y SPMD (*Simple Program Multiple Data*).
- ▲ Descripción de las arquitecturas basadas en comunicaciones mediante espacio de memoria compartido.
- ▲ Descripción del sistema de comunicación mediante paso de mensajes.
- ▲ Tipos de redes de intercomunicación en sistemas paralelos: estáticas y dinámicas.
- ▲ Análisis del problema de coherencia de caché en sistemas multiprocesador y sus posibles soluciones.
- ▲ Rendimiento y costes en sistemas paralelos.

Introducción

Hasta llegar a este punto, todo lo que se ha estudiado en los tres temas previos estaba orientado a la mejora de las prestaciones de computadores construidos con un único procesador. Sin embargo, hoy en día se sabe que los sistemas basados en un único procesador presentan límites en el rendimiento que pueden alcanzar, por lo que la tendencia que se ha establecido es la de construir máquinas dotadas de varios procesadores, pudiendo oscilar el número de procesadores desde unos pocos, como sucede en los computadores orientados al ámbito doméstico o empresarial, hasta alcanzar los cientos de miles en sistemas orientados a la computación de alto rendimiento. Evidentemente, esta tendencia, basada en el incremento de rendimiento en base a la suma de procesadores, va unida a toda la investigación que se continúa realizando para mejorar las características de los procesadores a nivel individual (consumo de energía, disipación de calor, tamaño, velocidad, etc.), ya se trate de procesadores superescalares, vectoriales o VLIWs. En este tema se presentan las arquitecturas más relevantes para construir computadores dotados de varios procesadores, es decir, sistemas de computación conocidos de forma coloquial como computadores paralelos. Para avanzar de forma ordenada y comprensible en la explicación de los diferentes tipos de plataforma de computación paralela que existen en la actualidad, se recurre a su clasificación en base a su organización lógica y física. La organización lógica se refiere a la visión que el programador tiene de la plataforma, es decir, las capacidades para expresar tareas paralelas (la estructura de control) y los métodos de comunicación entre dichas tareas (el modelo de comunicación). La organización física se refiere a la estructura del hardware que compone la plataforma, donde el criterio habitual es catalogar los sistemas en función de la organización del espacio de direcciones de memoria, obteniéndose así los dos tipos fundamentales: los sistemas de memoria compartida o *multiprocesadores* y los sistemas de memoria distribuida o *multicomputadores*.

El Capítulo 4 del texto base de la asignatura cubre completamente todos los contenidos de este tema.

Resultados de aprendizaje

RA 4.1. Analizar y distinguir las características de las principales plataformas de computación paralela, ya sea desde el punto de vista del programador o de la arquitectura hardware.

RA 4.2. Conocer los diferentes tipos de redes para la interconexión de componentes en sistemas de memoria compartida, analizando las ventajas e inconvenientes de cada tipo.

RA 4.3. Estudiar las técnicas más comunes para el mantenimiento de la coherencia en sistemas de caché compartida.

RA 4.4. Estudiar las características y arquitecturas más comunes de los sistemas tipo clúster.

RA 4.5. Analizar y conocer el rendimiento y las ventajas de uso de los sistemas de computación paralela.

Otra información de tipo general sobre la asignatura que también resulta útil como orientación en su estudio es la siguiente:

Contextualización

Todos los temas de la asignatura están relacionados entre sí y sirven para cubrir el siguiente objetivo básico:

Dar una visión, lo más completa posible, de las diferentes filosofías arquitectónicas en que se basa el diseño y la construcción de los computadores actuales para mejorar sus prestaciones.

De acuerdo con sus contenidos y en función del posicionamiento de la asignatura, sus contenidos se pueden ver como estructurados en dos bloques. El primer bloque, que podría titularse *mejora de prestaciones en arquitecturas monoprocesador*, está formado por los tres primeros temas y es una continuación natural y progresiva de los conceptos estudiados en *Ingeniería de Computadores I*.

El segundo bloque, que podría denominarse *arquitecturas y procesamiento paralelo*, consta del tema 4 y en él se abordan los conceptos necesarios para establecer el punto de partida para los temas de procesamiento y programación paralela que se estudian en el tercer curso de los grados que oferta la UNED en materia de ciencias de la computación. .

Materiales requeridos para el estudio

El material que se requiere para el estudio de la asignatura “Ingeniería de Computadores II” es la bibliografía básica recomendada:

Título: INGENIERÍA DE COMPUTADORES II

Autores: Dormido Canto, Sebastián; Sánchez Moreno, José; Sanz Prat, Victorino

Editorial: SANZ Y TORRES

ISBN: 978-84-92948-60-4

Dado el perfil del alumno para el que se ha escrito este libro, se ha tratado de cuidar de manera muy especial los aspectos específicos de la enseñanza a distancia. Los conceptos se introducen de forma progresiva, tratando de que el estudio se realice de forma incremental y asentando los conceptos vistos con anterioridad. La gran cantidad de figuras, tablas y ejemplos que presenta el texto (más de 200) tienen como objetivo facilitar su estudio sin la ayuda directa de un profesor.

La estructuración de todos los capítulos es uniforme. Todos contienen, además de las secciones específicas de desarrollo del tema, una sección guión-esquema donde se enumeran los temas tratados y una introducción en la que se exponen los objetivos a cubrir y se dan algunas reseñas históricas, una sección final de conclusiones en la que se resumen los conceptos introducidos y se proporciona una visión global y de futuro, una sección de preguntas de autoevaluación (alrededor de 250) y una sección de problemas (más de 40) con diferentes grados de dificultad que pretenden cubrir todos los aspectos tratados a lo largo de cada capítulo.

El índice completo de este texto se presenta a continuación:

Capítulo 1. Procesadores segmentados

- 1.1. Guión-esquema
- 1.2. Introducción
- 1.3. Procesadores RISC frente a procesadores CISC
- 1.4. Clasificación de las arquitecturas paralelas
- 1.5. Evaluación y mejora del rendimiento de un computador
- 1.6. Características de los procesadores segmentados
- 1.7. Arquitectura segmentada genérica
 - 1.7.1. Repertorio de instrucciones de la ASG
 - 1.7.2. Implementación de la segmentación de instrucciones en la ASG
- 1.8. Riesgos en la segmentación
 - 1.8.1. Riesgos estructurales
 - 1.8.2. Riesgos por dependencias de datos
 - 1.8.2.1. La reorganización de código
 - 1.8.2.2. El interbloqueo entre etapas
 - 1.8.2.3. El adelantamiento (camino de *bypass* o *forwarding*)
 - 1.8.3. Riesgos de control
- 1.9. Planificación dinámica: Algoritmo de Tomasulo
- 1.10. Resumen
- 1.11. Referencias
- 1.12. Preguntas de autoevaluación
- 1.13. Actividades

Capítulo 2. Procesadores superescalares

- 2.1. Guión-esquema
- 2.2. Introducción
- 2.3. Características de los procesadores superescalares
- 2.4. Arquitectura de un procesador superescalar genérico
- 2.5. Lectura de instrucciones
 - 2.5.1. Falta de alineamiento
 - 2.5.2. Rotura de la secuencialidad
 - 2.5.3. Tratamiento de los saltos
 - 2.5.4. Estrategias de predicción dinámica
 - 2.5.4.1. Predicción de la dirección de destino de salto mediante BTAC
 - 2.5.4.2. Predicción de destino de salto mediante BTB con historial de salto
 - 2.5.4.3. Predictor de Smith o predictor bimodal
 - 2.5.4.4. Predictor de dos niveles basado en el historial global
 - 2.5.4.5. Predictor de dos niveles basado en el historial local
 - 2.5.4.6. Predictor de dos niveles de índice compartido gshare
 - 2.5.4.7. Predictores híbridos
 - 2.5.5. Pila de dirección de retorno
 - 2.5.6. Tratamiento de los errores en la predicción de los saltos
- 2.6. Decodificación
 - 2.6.1. Predecodificación
 - 2.6.2. Traducción de instrucciones
- 2.7. Distribución
 - 2.7.1. Organización de la ventana de instrucciones
 - 2.7.2. Operativa de una estación de reserva individual
 - 2.7.2.1. Fase de distribución

- 2.7.2.2. Fase de supervisión
- 2.7.2.3. Fase de emisión
- 2.7.3. Lectura de los operandos
- 2.7.4. Renombramiento de registros
 - 2.7.4.1. Organización independiente del RRF con acceso indexado
 - 2.7.4.2. Organización independiente del RRF con acceso asociativo
 - 2.7.4.3. Organización del RRF como parte del buffer de reordenamiento
- 2.7.5. Ejemplo de procesamiento de instrucciones con renombramiento
- 2.8. Terminación
- 2.9. Retirada
- 2.10. Mejoras en el procesamiento de las instrucciones de carga/almacenamiento
 - 2.10.1. Reenvío de datos entre instrucciones de almacenamiento y de carga
 - 2.10.2. Terminación adelantada de las instrucciones de carga
- 2.11. Tratamiento de interrupciones
 - 2.11.1. Excepciones precisas con buffer de reordenamiento
- 2.12. Limitaciones de los procesadores superescalares
- 2.13. Resumen: Una visión global del núcleo de ejecución dinámica
- 2.14. Referencias
- 2.15. Preguntas de autoevaluación
- 2.16. Actividades

Capítulo 3. Procesadores VLIW y procesadores vectoriales

- 3.1. Guión-esquema
- 3.2. Introducción
- 3.3. El concepto arquitectónico VLIW
- 3.4. Arquitectura de un procesador VLIW genérico
- 3.5. Planificación estática o basada en el compilador
- 3.6. Desenrollamiento de bucles
- 3.7. Segmentación software
- 3.8. Planificación de trazas
- 3.9. Operaciones con predicado
- 3.10. Tratamiento de excepciones
- 3.11. El enfoque EPIC
- 3.12. Procesadores vectoriales
- 3.13. Arquitectura vectorial básica
- 3.14. Repertorio genérico de instrucciones vectoriales
- 3.15. Medida del rendimiento de un fragmento de código vectorial
- 3.16. La unidad funcional de carga/almacenamiento vectorial
- 3.17. Medida del rendimiento de un bucle vectorizado
- 3.18. Resumen: Visión global y perspectivas de futuro
- 3.19. Referencias
- 3.20. Preguntas de autoevaluación
- 3.21. Actividades

Capítulo 4. Procesamiento paralelo

- 4.1. Guión-Esquema
- 4.2. Introducción
- 4.3. Tipos de plataformas de computación paralela
 - 4.3.1. Basado en la estructura de control

- 4.3.1.1. Paradigma Maestro/Esclavo
- 4.3.1.2. Paradigma SPMD (*Single Program Multiple Data*)
- 4.3.2. Basado en el modelo de comunicación
 - 4.3.2.1. Espacio de direcciones compartido
 - 4.3.2.2. Paso de mensajes
- 4.4. Sistemas de memoria compartida
 - 4.4.1. Redes de interconexión
 - 4.4.1.1. Redes estáticas
 - 4.4.1.2. Caracterización de redes estáticas
 - 4.4.1.3. Redes dinámicas
 - 4.4.1.4. Comparación del rendimiento de redes dinámicas
 - 4.4.2. Protocolos de coherencia de caché
- 4.5. Sistemas de memoria distribuida
 - 4.5.1. Consideraciones generales sobre los clusters
 - 4.5.2. ¿Por qué clusters?
 - 4.5.3. ¿Cuándo y cómo utilizar un cluster?
 - 4.5.4. Programación de clusters
- 4.6. Rendimiento y costes en sistemas paralelos
 - 4.6.1. Factores que influyen en la velocidad computacional
 - 4.6.1.1. Granularidad de los procesos
 - 4.6.1.2. Factor de aceleración (*speedup*)
 - 4.6.1.3. Ley de Amdahl
 - 4.6.1.4. Eficiencia
 - 4.6.1.5. Coste
 - 4.6.1.6. Escalabilidad
 - 4.6.1.7. Balance de carga
 - 4.6.2. Costes de la comunicación mediante paso de mensajes
 - 4.6.3. Costes de la comunicación mediante memoria compartida
- 4.7. Resumen
- 4.8. Referencias
- 4.9. Preguntas de autoevaluación
- 4.10. Actividades

Orientaciones concretas para el estudio de los contenidos

El texto básico de la asignatura ha sido desarrollado siguiendo las pautas de elaboración de textos didácticos para la educación a distancia. En él se refleja en cada uno de los capítulos los contenidos que se han de conocer para abordar su estudio con éxito, así como los contenidos fundamentales de cada capítulo.

Como estrategias de aprendizaje recomendadas para el estudio de la asignatura cabe destacar lo siguiente:

- ▲ La asignatura es sumamente práctica.
- ▲ Hay que evitar estudiar de memoria e intentar comprender bien los conceptos.
- ▲ Se recomienda hacer las actividades prácticas así como plantearse sus propios ejercicios.
- ▲ Evitar consultar excesivamente los libros permitidos durante la realización de la prueba presencial. El examen está pensado para que se pueda solucionar sin realizar ningún tipo de consulta, por lo que los libros deben estar, únicamente, para pequeñas comprobaciones de última hora. Si no, lo más probable es que no le dé tiempo suficiente para acabar.

Orientaciones sobre los ejercicios de autoevaluación

Como ya se mencionó en el enlace de Evaluación de los Aprendizajes disponible en la primera parte de la guía, se publicará en el curso virtual un conjunto de Pruebas de Autoevaluación. La finalidad de estas pruebas es la de permitir al estudiante evaluarse, mientras realiza el estudio de los materiales y detectar si está asimilando adecuadamente los contenidos de la asignatura. Dispondrán de un solucionario, de forma que el estudiante pueda conocer su nivel de preparación y solventar lagunas, y no tendrán influencia en la calificación final de la asignatura. La publicación de estas pruebas será secuenciada temporalmente según el calendario propuesto en el plan de trabajo. Para realizar un uso adecuado de estas pruebas se recomienda al alumno el haber estudiado previamente los contenidos teóricos correspondientes.

Actividades complementarias

Puesto que el curso 2011-2012 es el primero en el que se imparte esta asignatura, no se disponen de exámenes de años anteriores. Sin embargo, la utilización de las preguntas de autoevaluación y la realización de las actividades prácticas situadas al final de cada capítulo del texto base son una ayuda excelente para preparar los exámenes.

En los cursos venideros se proporcionará al alumno una colección de exámenes de cursos anteriores que le ayudarán en su preparación. Estos exámenes estarán disponibles en el curso virtual, que se irá actualizando tras la realización de cada prueba presencial.

3.- ORIENTACIONES PARA LA REALIZACIÓN DEL PLAN DE ACTIVIDADES

En esta asignatura está prevista la realización de una única PED. En este apartado se proporciona algunas orientaciones para el desarrollo de esta prueba de carácter voluntario, la cual tiene un peso del 10% en la calificación final. El objetivo de la PED es el de afianzar y ampliar algunos de los conocimientos estudiados en la teoría. Tal y como se ha indicado, la PED podrá constar de uno o varios ejercicios prácticos o consistir en el uso de un simulador didáctico de un procesador segmentado.

En caso de tratarse de ejercicios prácticos, la PED constará de ejercicios similares a los que pueden aparecer en la prueba presencial. Ejemplos de este tipo de ejercicios se pueden encontrar al final de cada capítulo del texto base de la asignatura.

Si la PED se basase en el uso del simulador, la prueba consistirá en la programación de un procesador segmentado, muy similar al descrito en el capítulo 1 del texto base, para desarrollar un conjunto de programas en ensamblador que pongan de manifiesto algunas de las características de este procesador y de las técnicas estudiadas para mejorar el rendimiento mediante incremento del paralelismo a nivel de instrucción (por ejemplo, el desenrollamiento de bucles o la segmentación software).

El enunciado de la PED recogerá el tipo de actividad a realizar, los materiales necesarios, los plazos y la forma de presentación de la prueba. En el caso del uso del simulador, los materiales necesarios para realizar la PED, además del enunciado, serán el software de simulación y el manual de uso.

La evaluación de la PED correrá a cargo de los profesores tutores y/o del equipo docente. Un ejemplo de protocolo de evaluación se muestra en la tabla 1 cuando la PED está basada en el uso del simulador didáctico. En esta tabla (que se suele conocer como rúbrica), la columna "Categoría" muestra el objetivo de la evaluación. De entre las columnas tituladas "4", "3", "2", y "1" el tutor elegirá lo que corresponda para cada

alumno y para cada ejercicio. Por último, en la columna de observaciones el tutor podrá realizar cualquier comentario que considere oportuno. La máxima calificación es 10, siendo necesario obtener un 5 para superar la prueba.

Tabla 1 Ejemplo de protocolo de evaluación de la PED cuando se trata del uso del simulador para desarrollar las técnicas de desenrollamiento de bucles y segmentación software: (Rúbrica general para la Prueba de Evaluación a Distancia)

CATEGORÍA	4	3	2	1	OBSERVACIONES
Programar el bucle en ensamblador (P1:1)	Sabe programar el bucle en ensamblador	Sabe programar el bucle en ensamblador pero comete errores leves	Sabe programar el bucle en ensamblador pero comete errores graves	No sabe	
Explicar las causas de las detenciones al ejecutar el bucle en el simulador (P1:2)	Explica perfectamente las causas de las detenciones	No explica correctamente todas las detenciones, solo algunas	No explica correctamente las detenciones	No sabe	
Calcular el CPI del bucle original y explicar cómo lo realiza (P1:3)	Lo calcula y explica correctamente	Lo calcula correctamente pero no lo explica	Lo calcula pero con ligeros errores de concepto	No sabe	
Saber desenrollar y programar el bucle sin planificación (P1:4)	Programa correctamente el bucle desenrollado sin planificación	Desenrolla y programa el bucle con algunos errores	Desenrolla y programa el bucle con muchos errores	No sabe	
Calcular el CPI del bucle desenrollado y sin planificar y explicar cómo lo realiza (P1:5)	Lo calcula y explica correctamente	Lo calcula correctamente pero no lo explica	Lo calcula pero con ligeros errores de concepto	No sabe	
Planificar y programar el bucle desenrollado para reducir las detenciones (P1:6)	Planifica y programa correctamente el bucle desenrollado	Planifica y programa el bucle desenrollado con algunos errores	Planifica y programa el bucle desenrollado con muchos errores	No sabe	
Calcular el CPI del bucle desenrollado y planificado y explicar cómo lo realiza (P1:7)	Lo calcula y explica correctamente	Lo calcula correctamente pero no lo explica	Lo calcula pero con ligeros errores de concepto	No sabe	
Programar la segmentación software del bucle original y explicarla (P2:1)	Programa la segmentación software y la explica correctamente	Programa la segmentación software pero no la explica correctamente	Realiza la programación de la segmentación software pero comete errores	No sabe	
Calcular el CPI del bucle obtenido mediante segmentación software (P2:2)	Lo calcula y explica correctamente	Lo calcula correctamente pero no lo explica	Lo calcula pero con ligeros errores de concepto	No sabe	

INGENIERÍA DE COMPUTADORES II

Extraer conclusiones (P3:1)	Las conclusiones que obtiene son correctas	Las conclusiones que obtiene están incompletas o son obvias	Las conclusiones que obtiene son incorrectas	No sabe	
---------------------------------------	--	---	--	---------	--