

Combinación de Prácticas de Simulación y Reales para un Curso Anual de Arquitectura de Computadores

E.J. González, S. Alayón, J.D. Piñeiro, J.L. Sánchez, J.I. Estévez, L. Moreno

Dept. de Física Fundamental y Experimental, Electrónica y Sistemas

Universidad de La Laguna

38207 S/C de Tenerife

e-mail: evelio@cyc.dfis.ull.es

Resumen

Se presenta un programa de prácticas para un curso anual de la asignatura Arquitectura de Computadores del Segundo Ciclo de la Ingeniería Informática, que trata de conseguir un equilibrio entre la simulación y las prácticas reales de laboratorio.

1. Introducción

La enseñanza de asignaturas prácticas, como puede ser en nuestro caso las de Informática e Ingeniería Electrónica, debe ser una combinación apropiada de teoría, ejercicios y experimentos de laboratorio [1,2,3,4]. Respecto a la parte práctica, que es la que permite una mayor innovación, un gran número de paquetes de simulación, que proveen entornos de aprendizaje interesantes, han sido desarrollados en esta última década. En el caso particular de la Arquitectura de Computadores, existen excelentes paquetes de simulación, como SPIM [5], WinDLX, Midas [6], DLXVSim [7]... que facilitan la parte práctica de la asignatura, en especial, en áreas como la eficiencia de un conjunto particular de instrucciones, la aplicación de técnicas de segmentación, manejo de memoria, o el análisis de mejoras estructurales como la inclusión de nuevas unidades funcionales, memorias caché o la ejecución fuera de orden.

Sin embargo, esta opción no es completa y debería combinarse con experimentos reales en el laboratorio. No es fácil encontrar un equilibrio entre ambos tipos de prácticas: simulación y hardware. Se recurre más a las prácticas de simulación porque son más fáciles y baratas y los profesores no necesitan supervisar el trabajo de

los estudiantes de una manera tan directa como en el caso de las prácticas de hardware. Por el contrario, estas últimas llevan un largo tiempo de preparación y pueden llegar a ser tediosas y demasiado complicadas si no son cuidadosamente preparadas y seleccionadas. Sin embargo, existen numerosas áreas que no pueden ser enseñadas adecuadamente a través de la simulación, siendo entonces para éstas aconsejable la experimentación hardware.

Así, por ejemplo, los alumnos pueden obtener un conocimiento más profundo del paralelismo en redes de interconexión basadas en bus con prácticas tales como diseño de árbitros, memorias compartidas, accesos a zonas críticas... Estos diseños fuerzan a los estudiantes a ser cuidadosos en el diseño, formular estrategias de depuración y emplear herramientas de visualización (por ejemplo analizadores lógicos)

En esta ponencia se presenta una combinación de prácticas simulación-hardware para un curso anual de Arquitectura de Computadores (10.5 créditos) que pensamos se ajusta a los requisitos comentados. Este curso, en su primera parte incluye procesadores segmentados, procesadores vectoriales y el paralelismo a nivel de instrucciones. Una segunda parte se dedica al multiprocesamiento, y más exactamente a las redes de interconexión (buses, redes de interconexión simples y multietapa). Finalmente, una tercera parte se dedica a arquitecturas especiales, como las dedicadas al procesamiento de señales.

El resto de la ponencia estructura del siguiente modo. En los apartados 2, 3 y 4 se presentan las prácticas propuestas correspondientes a las diferentes partes de la asignatura. El apartado 5 se refiere a las conclusiones de la presente ponencia.

2. Procesadores segmentados, vectoriales y superescalares.

Como hemos indicado anteriormente, en la primera parte del curso propuesto se incluyen procesadores segmentados, procesadores vectoriales y el paralelismo a nivel de instrucciones.

Las clases teóricas se pueden ver reforzadas mediante el empleo de paquetes de simulación. En nuestro caso, los paquetes utilizados son el WinDLX, DLXVSim y Midas.

Con los dos primeros, los alumnos pueden visualizar los conocimientos adquiridos en clase sobre los aspectos relevantes de la arquitectura DLX (un repertorio RISC de instrucciones, segmentación...) y su versión vectorial: el DLXV.

La técnica es la siguiente: a los alumnos se les sugiere un código en DLX que realiza una operación sencilla (como sumar una constante a un vector) pero de una manera totalmente ineficiente (por ejemplo, con accesos a memoria en vez de emplear los registros). A partir de ese código ineficiente se proponen mejoras al mismo (eliminando dependencias, saltos...), midiendo la reducción producida en los ciclos de ejecución. De este modo, los alumnos comprueban por sí mismo la bondad de las técnicas explicadas en las clases teóricas. Los programas seleccionados además ofrecen la posibilidad de estudiar la técnica del adelantamiento (*forwarding*).

El último paquete, Midas, ofrece la posibilidad de que el alumno complemente sus conocimientos sobre procesadores superescalares con una herramienta que simula el comportamiento de una máquina MIPS R10000.

Todas estas herramientas son de libre uso y los alumnos pueden acceder a ellas a través de ftp.

3. Multiprocesamiento: Redes de interconexión.

Esta segunda parte se basa en el estudio del multiprocesamiento. A nuestro juicio, esta área de la Arquitectura de Computadores, es la más interesante para prácticas reales en el laboratorio. Al requerir más preparación que las prácticas

anteriores, las desglosaremos con un mayor detalle.

2.1. Prácticas de Buses. Aplicación a VME.

El conjunto de ejercicios propuestos en este apartado se basa en un prototipo basado en el microprocesador 80x86 y que ha sido diseñado completamente en nuestro laboratorio. Los principales componentes de este prototipo son:

- Procesador Intel
- Memoria Local
- Displays de 7 segmentos y teclado hexadecimal
- Conector para un subconjunto de bus ISA
- Adaptador y conector a un subconjunto de bus VME

El empleo de este prototipo resta universalidad a nuestra propuesta, pero en cambio, permite la realización de unas prácticas sobre buses que de otra manera serían muy complicadas de llevar a cabo. En efecto, el empleo de sistemas comerciales basados en el bus VME o ISA, como es el caso del sistema Miniforce VME, presenta numerosos problemas. Los estudiantes, en general, incluyen en sus diseños fallos graves (cortocircuitos entre tierra y alimentación, por ejemplo) o simplemente no toman precauciones (suelen desconectar el montaje al sistema estando éste en funcionamiento). Por tanto, si dejásemos que los alumnos conectasen directamente sus diseños al sistema Miniforce, probablemente dañarían algunos de sus componentes: CPU, interfaces...

Además el prototipo, cuyo manual de funcionamiento es suministrado a los alumnos con anterioridad a las experiencias, admite ampliaciones para nuevas prácticas, así como la posibilidad de tener un conjunto de árbitros que se puedan conectar y desconectar fácilmente al sistema. Incluso, estos árbitros pueden ser diseñados por los propios alumnos, implementando varias políticas de arbitraje.

Estos diseños se implementan en un panel como los mostrados en la Figura 1. Este tipo de panel ofrece la gran ventaja de que no se necesita soldadura cuando se diseña un nuevo dispositivo.

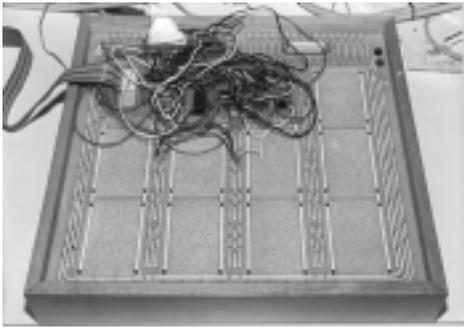


Figura 1. Panel para los montajes de los alumnos

Finalmente, el equipo de la práctica se completa con un analizador lógico Hewlett Packard 1664. Con este modelo de analizador, cuyo precio es asequible, los estudiantes pueden obtener una copia dura de los diagramas de tiempo obtenidos durante la práctica. El montaje completo puede verse en la Figura 2.

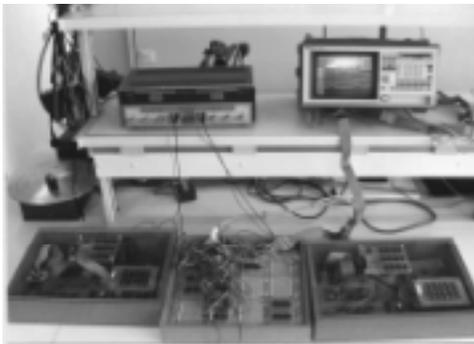


Figura 2: Material para las prácticas propuestas

Cuando los estudiantes entran en el laboratorio de Arquitectura de Computadores para llevar a cabo las experiencias, han adquirido previamente un conocimiento teórico sobre las redes de interconexión. [8,9]

Podemos dividir las prácticas en dos grandes bloques, de acuerdo con el tipo de bus (síncrono/asíncrono).

- 1.- Protocolo Síncrono: Expansión de memoria del prototipo.
- 2.- Protocolo Asíncrono: Implementación de un esclavo (expansión de memoria) a través del

interface adaptador VME. Esclavo compartido por dos masters. Diseño e implementación de dispositivos de arbitraje.

3.- Interpretación de cronogramas.

Con estas prácticas, los alumnos logran analizar protocolos síncronos/ asíncronos, observando las diferencias entre ellos, adquirir experiencia en el empleo de instrumentos como el analizador lógico, indispensables hoy cuando se trabaja en sistemas complejos como éste y asentar los conocimientos adquiridos en las clases teóricas sobre procesamiento paralelo.

A continuación detallaremos brevemente las prácticas sugeridas.

Protocolo síncrono: Ampliación de memoria.

Esta sencilla práctica permite a los estudiantes familiarizarse con el laboratorio y con las herramientas que deben emplear.

Los chips empleados son 8K x 8 Bit Fast Static RAM MCM 6264. Como estos chips son de 8 bits, mientras que el procesador 80x86 tiene 16 líneas de datos, la ampliación de memoria se organiza en parejas de chips, referentes a las direcciones pares e impares. Los módulos son seleccionados con las señales A0 y BHE* del microprocesador.

Protocolo asíncrono: Implementación de un esclavo (expansión de memoria) a través del interface adaptador VME

En esta experiencia, los alumnos asimilan el protocolo asíncrono que previamente han estudiado de una forma teórica. Deben implementar un esclavo en un sistema de un único master (un prototipo 80x86). En nuestro caso, el esclavo será una expansión de memoria con las mismas características que la práctica anterior (mismos módulos, direcciones pares/ impares).

Pero quizás el aspecto más pedagógico de la experiencia consiste en comprobar si el diseño implementado cumple con las restricciones de tiempo impuestas por el protocolo VME. Esto se realiza con el analizador lógico, el cual permite efectuar mediciones de tiempo de una manera precisa.

Protocolo Asíncrono: Esclavo compartido por dos masters.

Una vez que se ha completado la anterior práctica, el siguiente paso lógico es el de diseñar un sistema con varios masters accediendo a un recurso común. El diseño del interface hace que los estudiantes solamente tengan que conectar el nuevo master al diseño implementado, ya que pueden emplear uno de los sistemas de arbitraje contruidos anteriormente.

Sin embargo, lo más interesante desde el punto de vista pedagógico no es la conexión, sino los aspectos relativos al procesamiento paralelo. Para fomentar estos aspectos, se propone a los alumnos un problema productor/ consumidor. Los alumnos resuelven este ejercicio definiendo un área crítica mediante un semáforo binario. De esta forma, los alumnos pueden comprobar las ventajas de un bus asíncrono, controlando las líneas del bus, y probando, por ejemplo, si un dato escrito en el recurso común puede ser leído perfectamente por el otro master.

Protocolo Asíncrono: Diseño e implementación de árbitros.

Esta práctica se considera útil, ya que a las anteriores restricciones de tiempo deben añadirse las relativas al arbitraje.

Se le puede suministrar a los estudiantes un diseño previo de árbitro o, en el caso de que se desee obtener una mejor calificación, los propios alumnos pueden diseñar un árbitro que además podría estar basado en otra política.

Protocolo asíncrono: Interpretación de cronogramas (examen de los conocimientos adquiridos)

Una vez que los estudiantes han completado las prácticas anteriores, se les muestran varios cronogramas capturados con el analizador lógico, donde se hace hincapié en los diferentes pasos del protocolo. Estos diagramas de tiempo se muestran en orden creciente de complejidad.

3.2. Diseño de un sistema multiprocesador basado en transputers.

Con esta experiencia de laboratorio se pretende que el estudiante tenga un contacto real con el procesamiento paralelo en arquitecturas de paso de mensajes, tanto a nivel hardware como software.

El sistema multiprocesador que deben utilizar está implementado con Transputers del Kit Educativo de CSA [10]. La tarjeta del transputer está constituida por tres secciones: el transputer con su memoria local, la interface PC/Link y la interface externa. El transputer es un microprocesador RISC con un diseño interno distinto a microprocesadores convencionales. El concepto central es el de «proceso» y esto se refleja en su repertorio de instrucciones. La interface PC/Link hace posible la comunicación entre el transputer y el PC y el paso de datos y señales de control. La interface externa permite al transputer controlar hardware externo.

Las comunicaciones entre procesos se llevan a cabo mediante «canales» por «paso de mensajes». Un canal entre dos procesos que se ejecutan en distintos transputers se implementa con «links». Los links son los encargados de realizar la transferencia con DMA sin intervención del procesador. Se implementan con cables que permiten conectar tarjetas entre sí y establecer una comunicación serie y bidireccional. Existe una gran flexibilidad a la hora de conectar las tarjetas lo que hace posible formar distintas topologías de redes. Estas redes pueden estar localizadas dentro de un PC o distribuidas en varios PC's [2].

El sistema diseñado en el laboratorio implementa un anillo de cuatro transputers: un T800 y tres T400. Las diferencias principales entre ambas series son que la serie 800 contiene una FPU y cuatro links, mientras que la serie 400 carece de FPU y sólo consta de dos links. El esquema se muestra en la siguiente figura.

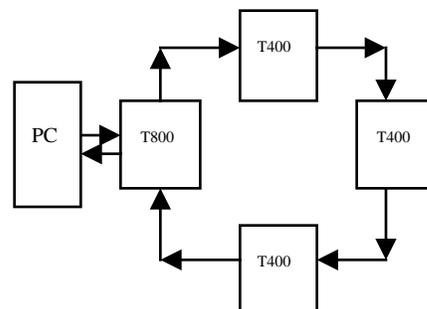


Figura 3. Anillo implementado por el sistema.

Los transputers que constituyen el anillo colaboran en la realización de una tarea global,

ocupándose cada uno de una subtarea. La aplicación concreta que se ha preparado para el laboratorio es la implementación de un filtro decimador multietapa. Se ha escogido esta aplicación por las siguientes razones:

- La importancia que este campo tiene dentro del procesamiento de señales, conocido en la nomenclatura inglesa como *multirate processing*.
- Porque los fundamentos teóricos de la experiencia son conocidos por los alumnos, ya que han sido estudiados en una asignatura de procesamiento de señales que se imparte al mismo tiempo que la asignatura de Arquitectura de Computadores objeto del presente artículo.
- Porque la arquitectura más adecuada para la realización de este filtro decimador es la topología de anillo.

El factor de decimación total es 96 y el filtro se divide en tres etapas cuyos factores de decimación son 8, 6 y 2 respectivamente [11].

La tarjeta host (T800) ejecuta dos procesos concurrentes: el primero genera muestras de una señal sinusoidal a una determinada frecuencia y las envía en paquetes al resto del anillo y el segundo está a la espera de que lleguen los paquetes de muestras procesadas por el anillo para finalmente guardarlos en el PC.

Las tarjetas T400 implementan cada una de las tres etapas en las que se ha dividido el filtro. Cada una de ellas ejecuta tres procesos distintos: proceso buffer de entrada (proceso que está a la espera del paquete de datos que le llega del nodo anterior del anillo), proceso que implementa el algoritmo del filtro de la etapa y proceso buffer de salida (envía el paquete procesado al siguiente nodo de la red).

El estudiante deberá cumplir los siguientes objetivos:

- Dados los coeficientes y algoritmos de los filtros, realizar los programas de las distintas tarjetas, con especial cuidado en la comunicación entre procesos y uso de los links. El software que va a utilizar es una versión especial de C para transputers.
- Comprobar el buen funcionamiento del sistema y realizar medidas de tiempos.

- Implementar el mismo filtro multietapa en un solo procesador de modo secuencial y realizar medidas de tiempos.
- Comprobar la eficiencia de las estrategias de procesamiento paralelo frente al secuencial: comparación de las medidas de tiempos.

4. Arquitecturas de Procesamiento de Señales. Aplicación a DSP.

El objetivo de esta parte no es tanto que los alumnos realicen una aplicación de envergadura en un Procesador de Señales Digitales (DSP) como que realicen varios pequeños ejemplos en los que puedan resolver el mismo problema y comparar por un lado la velocidad de ejecución y por otro la extensión del código. Estas prácticas están implementadas sobre un DSP TMS320C40 de Texas Instruments [12]. Se persigue, además, que el alumno pueda utilizar el DSP y aprender a programarlo en ensamblador, como complemento a las clases teóricas sobre la arquitectura del mismo.

Para la implementación de la práctica se pueden considerar dos niveles:

- Nivel de simulación: El alumno debe hacer el programa, pero lo ejecuta sobre un simulador del DSP, de forma que no tiene problemas debidos al hardware.
- Nivel de ejecución sobre DSP: Una vez superado el nivel anterior, el alumno está capacitado para ejecutar esta práctica sobre una tarjeta que contenga el DSP y comprobar su funcionamiento en tiempo real.

Desde un punto de vista pedagógico, el alumno puede aprender a programar el DSP sin enfrentarse a los problemas de ejecución sobre dicho procesador. Por ejemplo, características de la tarjeta en la que está insertado el procesador, condiciones hardware necesarias para su buen funcionamiento, etc. De esta manera el aprendizaje es más rápido al discriminar entre las fuentes de posible error (hardware o software) y el alumno puede reconocer los problemas típicos de cada fase.

Una vez que el alumno ha recibido una adecuada información sobre los aspectos básicos del DSP (su arquitectura, software a emplear, estructura de un programa en ensamblador, juego

de instrucciones...), se plantea una práctica que consiste en resolver un problema presentado bajo diferentes aspectos, de forma que los alumnos se vean obligados a utilizar instrucciones diferentes del repertorio, tipos de direccionamiento, instrucciones paralelas... Como práctica base se puede considerar el problema de cargar dos vectores de punto flotante en memoria utilizando las instrucciones de carga. Se deberían emplear para su realización el direccionamiento indirecto con pro o postincremento (o decremento), instrucciones en un bloque de repetición, instrucciones paralelas, direccionamiento circular...

5. Conclusión

Se ha presentado un curso anual de Arquitectura de Computadoras, tratando de encontrar un equilibrio entre la simulación y las prácticas reales de laboratorio.

La simulación nos permite reforzar los conocimientos sobre procesadores segmentados, vectoriales y superescalares y las arquitecturas de procesamiento de señales.

Por otro lado, las prácticas reales de laboratorio han sido cuidadosamente escogidas, valorando los aspectos pedagógicos de las mismas sobre multiprocesamiento, y especialmente, redes de interconexión. Requieren una adecuada preparación por parte del profesor, así como una completa documentación y una labor de seguimiento en el laboratorio. No obstante, se ha tratado de evitar que pudieran resultar tediosas o implicar demasiado tiempo.

Aparte de reforzar los conocimientos adquiridos en las clases teóricas, estas prácticas reales aportan al alumno experiencia en la depuración de prototipos y en la utilización de instrumentación especializada, como por ejemplo analizadores lógicos.

Es importante resaltar que hay conocimientos teóricos, como el procesamiento paralelo o el empleo de buses, que el alumno no puede asimilar completamente si no se le ofrece la posibilidad de trabajar sobre ellos en prácticas reales, debido a la complejidad de sus contenidos. Éste es el objetivo principal que pretendemos alcanzar con las prácticas diseñadas en el laboratorio.

Referencias

- [1] J. Fulcher. *Experience with Teaching Computer Architecture*. Int. J. Elect. Enging Educ., 30, 329-342, 1993.
- [2] L. Philips, P. Mechan, M.C. Cavenor. *Laboratory-Based Coursework in Parallel Processing Using the Transputer*. IEEE Transaction on Education, 37, N° 3, 299-302. August 1994.
- [3] L. Moreno, L. Acosta, A. Hamilton, J.L. Sánchez, J.D. Piñeiro, J.J. Merino, R.M. Aguilar. *Experiments on a DC motor Based System Digital Control Course*. IJEEE, 32, 163-178, 1995.
- [4] L. Moreno, I. Roca, E. Luque. *Floppy Disk Controllers for a Computer Architecture Course*. IEEE Transactions on Education, Vol 32, No. 2, 1989.
- [5] SPIM ha sido desarrollado por J.R.Larus (Computer Science Department. University of Winsconsin-Madison).
- [6] WinDLX y Midas han sido desarrollados por H. Grünbacher (Vienna University of Technology).
- [7] WinDLXV ha sido desarrollado por Roberto Calpé y Pedro López (Universidad Politécnica de Valencia).
- [8] J. Duato, S. Yalamanchili, L. Ni. *Interconnection Networks: An Engineering Approach*. IEEE Computer Society, 1997.
- [9] A.L. Decegama, «The Technology of Parallel Processing». Prentice Hall. 1989.
- [10] CSA Computer System Architects. *Transputer Education Kit. User Guide*, 1994
- [11] P.P. Vaidyanathan, *Multirate Systems and Filter Banks*. Prentice Hall, 1993.
- [12] *TMS320C4X C Source Debugger. User Guide*. Texas Instruments, 1992.