

SIMULADOR DE ARQUITECTURAS SEGMENTADAS SENCILLAS CON FINES DOCENTES

Benavides, J. Ignacio; Saez, Edmundo; Montijano, Miguel; Ortiz, Manuel;
Márquez, Antonio T.; Ramírez, Antonio J.
Grupo Arquitecturas Avanzadas de Computadores, Universidad de Córdoba
E.U. Politécnica, Avda. Menéndez Pidal, s/n, 14004 – Córdoba
957 218375(6), 957 218316, e-mail: el1bebej@uco.es

RESUMEN

En este trabajo se presenta un software enmarcado en una de las líneas de investigación que viene desarrollando la Unidad Docente de Arquitectura de Ordenadores del Dpto. Electrotecnia y Electrónica de la Universidad de Córdoba, relacionada con la simulación de Arquitectura de Computadores con fines docentes. Se incide en este caso, sobre el aprendizaje de conceptos relacionados con la segmentación, a través de sencillos ejemplos, estudiando el impacto de la segmentación en el rendimiento de arquitecturas específicas. La aplicación a la enseñanza del software que aquí se implementa es inmediata, permitiendo a los alumnos el conocimiento de las técnicas actuales sobre el diseño de computadores.

1. INTRODUCCIÓN

En este trabajo vamos a presentar un software de simulación con fines docentes en el área de Arquitectura de Computadores, y más concretamente en una de las técnicas de aumento del rendimiento de los computadores, conocida con el nombre de Segmentación. Dicho software ha sido desarrollado por alumnos de la Escuela Politécnica de la Universidad de Córdoba, dentro de su trabajo de Proyecto Fin de Carrera, dirigido por profesores de la Unidad Docente de Arquitectura de Computadores del Dpto. de Electrotecnia y Electrónica.

Dicha Unidad Docente utiliza en la enseñanza de asignaturas de la citada área diferentes simuladores, algunos de ellos desarrollados por la misma Unidad. Los buenos resultados obtenidos con su utilización nos ha animado a seguir desarrollando nuevos simuladores.

La revisión de algunos simuladores ya existentes, así como las condiciones y niveles de los alumnos a los que se van a impartir, nos ha llevado a plantear un diseño muy concreto que hace más hincapié en el concepto general de Segmentación que en la aplicación a los procesadores.

El resto del trabajo lo organizamos de la siguiente forma: el apartado dos estará dedicado a una revisión de lo que se entiende por Simulación y su aplicación a la docencia. El apartado tres estará dedicado a la segmentación. En el cuatro se hará una revisión tanto de los trabajos desarrollados por nuestra Unidad Docente en la línea de Simulación, y uno de los simuladores de que se disponía. El quinto y último apartado lo dedicaremos a exponer el software desarrollado por nosotros, con las distintas unidades que lo componen.

2. SIMULACIÓN Y SU APLICACIÓN DOCENTE

Desde un punto de vista muy general podríamos decir que la Simulación consiste en la imitación del comportamiento de un sistema existente o no, independiente del tipo del sistema, y del método y herramienta utilizados para realizar la "imitación". En la actualidad la Simulación se sobreentiende que es realizado por una computadora que ejecuta un programa, donde se modela el sistema a simular.

Aunque es en el diseño Industrial y otras áreas tecnológicas donde más importancia tiene, una de las aplicaciones de la simulación más desarrollada últimamente ha sido con fines docentes, lo que ha posibilitado, sin grandes recursos de laboratorio, poder hacer comprender a los alumnos aspectos importantísimos en cada área concreta.

En nuestro caso, para el desarrollo de los simuladores hemos de priorizar aspectos como carácter didáctico, fácil uso, plataforma necesaria estándar y rápida, etc., frente a otras más de tipo comercial y de producción que se tendría en el ámbito industrial.

El uso de los Simuladores se enmarcan dentro de las actividades prácticas de las asignaturas de Arquitectura y Tecnología de Computadores de los nuevos planes de estudios de Ingeniero Técnico en Electrónica Industrial e Informática. Su correlación con el contenido teórico de esas asignaturas es fundamental para cubrir su objetivo.

3. SEGMENTACIÓN

En los ámbitos informáticos se suele entender por Segmentación como la técnica de implementación en la cual múltiples instrucciones se solapan durante su ejecución. Algunos autores[1] consideran la introducción del procesamiento segmentado o en cascada en los computadores como uno de los avances más significativos en el desarrollo de los computadores. Dado un proceso (p.e., una operación aritmética) que puede dividirse (segmentarse) en varias etapas, podrá ejecutarse en una máquina segmentada de forma que simultáneamente se ejecuten diferentes etapas, tantas como en las que se ha dividido el proceso. Una característica fundamental para que la segmentación produzca el efecto deseado es que todas las etapas en las que se ha dividido tengan el mismo tiempo de proceso.

En la figura 1 se muestra un esquema de un proceso segmentado en cinco etapas. Los datos a ser procesados van entrando y en cada etapa sufren un tratamiento, de forma que una vez atravesado las cinco etapas se obtendrá el resultado. Pensando en un sistema de procesamiento, con la información representada digitalmente y sincronizada por una señal de reloj, en cada pulso de reloj la información saliente de una etapa se convierte en la entrada de la siguiente etapa.

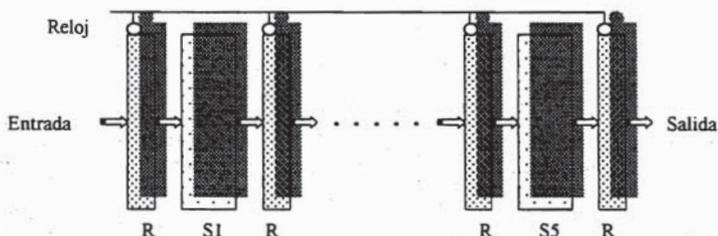


Figura 1. Esquema de un proceso segmentado en cinco etapas

En un caso teórico y suponiendo que los subsistemas intermedios de almacenamiento que mantienen la información en la entrada de cada etapa no implican retardo, se conseguiría un aumento de hasta cinco veces el número de operaciones realizables por unidad de tiempo.

La segmentación se puede aplicar en diferentes niveles: nivel de operación aritmética, nivel de instrucción, y nivel de procesadores. En nuestro caso sólo vamos a tratar de simular los dos primeros aspectos, viendo primero la segmentación de un conjunto de operaciones aritméticas de tratamiento de señal, y posteriormente la ejecución segmentada de instrucciones en un procesador simple.

4. ANTECEDENTES

Vamos a hacer una distinción entre los trabajos anteriores de nuestro grupo y algunos en el campo de la simulación de la Segmentación, disponibles como software de libre uso.

4.1. Desarrollos nuestros.

Nuestra Unidad viene trabajando en el desarrollo de simuladores de Arquitectura de Computadores. El más reciente de ellos fue un simulador de una arquitectura tipo CISC que incorpora un microensamblador[2] que se está usando en la enseñanza de las asignaturas "Arquitectura de Computadores" y "Diseño de Computadores" de los estudios de Ingeniería Técnica de Informática y Electrónica Industrial. Este simulador permite hacer un análisis detallado de una arquitectura CISC y puede ejecutar programas en tres diferentes niveles: nivel de programa, nivel de instrucción, y nivel de microinstrucción.

La buena experiencia de la aplicación de este simulador en el pasado curso nos animó a desarrollar nuevos simuladores.

4.2. Simuladores de otros Grupos.

Aunque existe muchos (p.e., MipSim32, WinDLX[3], SimMips) vamos a ver uno solo de ellos.

Simulador SimMips

Es un simulador desarrollado por la Facultad de Informática de la Universidad de Deusto. La aplicación se ejecuta en una ventana de diálogo y es muy simple de manejar, siendo ideal para el aprendizaje de la segmentación en los procesadores MIPS.

La simulación del procesador presenta tres modos diferentes: *monociclo*, la instrucción se ejecuta en un ciclo; *multiciclo*, una instrucción se ejecuta en varios ciclos pero sólo una vez; y *segmentado*, puede ejecutar varias instrucciones a la vez en un mismo procesador multiciclo.

En la pantalla principal del simulador MIPS segmentado, aparece además de la estructura del procesador ventanas de banco de registro, memoria de datos, memoria de instrucciones e instrucción ejecutándose.

Como conclusión añadiremos que el programa SimMips simula el procesamiento en cascada de instrucciones tal y como se haría sobre un procesador de tipo MIPS, además de otros modos de procesamiento. Es fácil de manejar, sin embargo, no muestra bien cómo las instrucciones se propagan por el procesador.

5. NUESTRA PROPUESTA

Como ya se comentó, nuestro objetivo es la implementación de un simulador que permita a los alumnos de las asignaturas relacionadas con Arquitectura de Computadores asimilar los conceptos básicos de segmentación. Por ello, nuestra propuesta se centra en primer lugar en la simulación a nivel de operaciones y, posteriormente, a nivel de instrucción en un procesador simple, dejando los aspectos de la simulación de procesadores segmentados más avanzados para otro simulador, como por ejemplo la segmentación en procesadores RISC.

El programa del simulador se ha realizado en Visual C++ por las ventajas inherentes que posee como: entorno gráfico amigable, disponibilidad de múltiples ventanas, menús desplegados, fácil ampliación de los sistemas a simular, etc. Éste puede dividirse en siete módulos, cada uno de los cuales simula una operación de cálculo concreto, y un octavo módulo que simula un procesador simple segmentado. Cada uno de estos módulos se ha denominado ejemplo y han sido elegidos operaciones con características básicas en punto flotante (suma, resta, multiplicación y división), así como algunas operaciones más usadas en el cálculo matemático (producto de un escalar por un vector, producto vectorial y una rutina de tratamiento de gráficos por computador como es la conocida con el nombre de Z-buffer).

Los ejemplos desarrollados han sido:

Ejemplo 1. $A = A + cte \cdot Vect(i)$, donde $i = 0, \dots, n-1$

Acumular en A el resultado de multiplicar una constante entera (cte) por los diferentes componentes de un vector.

Ejemplo 2. $A = A + Vect1(i) \cdot Vect2(i)$, donde $i = 0, \dots, n-1$

Acumular en A el resultado de multiplicar dos vectores componente a componente.

Ejemplo 3. $A(k) = A(k) + MA(k, i) \cdot MB(i)$, donde $k = 0, \dots, m-1$, y $i = 0, \dots, n-1$

En este caso, se realiza el producto de un vector de dimensión $[n]$ por una matriz de dimensión $[n \times m]$, y el resultado lo almacenamos en un vector de dimensión $[m]$.

Ejemplo 4. *Z-Buffer*

Es una operación que se aplica en la renderización de imágenes por computador que consiste en: dado una matriz bidimensional que representa el resultado de una renderización, el valor de un pixel (i, j) es sustituido por un nuevo valor proyectado si la profundidad del nuevo es menor que la del anterior. En definitiva, es una comparación de dos valores de profundidad (z) y se realiza una sustitución del valor del pixel si procede.

Ejemplos 5, 6, y 7. *Operaciones en punto flotante*

Estos ejemplos están dedicados a realizar las operaciones de suma, resta, multiplicación y división en punto flotante respectivamente.

En todos estos ejemplos se segmentan las operaciones en un número determinado de etapas y se puede observar el flujo de información a lo largo de la etapa en su ejecución. Para ver como funciona vamos a comentar con más profundidad la organización hardware de uno de los ejemplos.

5.1. Suma y resta en punto flotante.

La simulación de esta operación se basa en el algoritmo clásico de suma y resta de números en punto flotante, y suponiendo que se parte de una representación estándar normalizado. Consta de los siguientes pasos: verificación de ceros, alineación de mantisa, suma o resta de la mantisa, y normalización del resultado. De acuerdo con esto la operación se ha segmentado en 6 etapas, tal y como se muestra en la figura 2.

En la etapa inicial se seleccionan los datos con los que se va a operar, que se supone almacenados en una memoria. En la segunda etapa, además de verificar si alguno de los operandos es cero, realiza la comparación y resta de los exponentes para aplicarlo a la etapa siguiente, que alinea las mantisas realizando el desplazamiento correspondiente. En la cuarta etapa realiza la suma o resta, mientras que en la quinta, normaliza el resultado. La última etapa

está dedicada al almacenamiento del resultado. Salvo la primera y la última etapa, el resto se corresponde con las fases definidas en el algoritmo clásico. Cada etapa está separada de la anterior por un registro buffer.

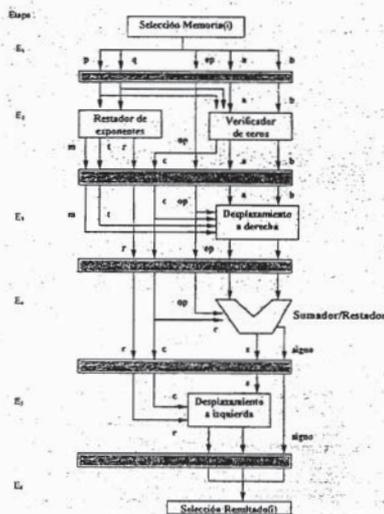


Figura 2. Etapas del algoritmo de suma y resta

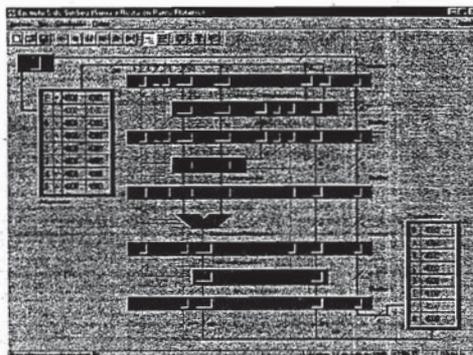


Figura 3. Simulación de la operación de suma y resta

En la figura 3 se muestra la pantalla correspondiente a la simulación de esta operación. En ella aparecen una serie de componentes como son:

- * *Registro de direcciones de memoria:* donde se indica la dirección que se está accediendo para la búsqueda de los datos.
- * *Memoria:* bloque de 16 posiciones de longitud $2n+1$, ya que en cada posición almacena los dos operandos y la operación a realizar; y bloque de 16 posiciones para almacenar el resultado.
- * *Bloques de unidades funcionales:* sumador, verificación de ceros, desplazamiento, etc.

6. REFERENCIAS

- [1] W. Stallings. "Organización y Arquitectura de Computadoras". Ed. Limusa. S.A. de C.V. Grupo Noriega Editores. 1995.
- [2] S. Gracia, A. Peñas. "Herramienta para el estudio interno de un Computador: Control simulado de ejecución. Sistema Microensamblador para la ejecución de nuevas Microinstrucciones". Proyec. Fin de Carrera de I.T.I. E.U. Politec. U. Córdoba. 1997.
- [3] H. Grünbacher. "WinDLx". Dpto. of VLSI-Design. Vienna University of Technology.