

ST0254 – Organización de computadores

Práctica 2: Diseño de procesadores de múltiples core

I.S. José Luis Montoya Pareja
Profesor Universidad EAFIT
Medellín, Colombia, Suramérica
email jmontoya@eafit.edu.co

Resumen

Esta práctica se orienta a la comprensión del funcionamiento de las tecnologías de procesamiento paralelo, específicamente la construcción de un procesador de tecnología SIMD (Single Instruction stream, Multiple Data stream), el cual permite efectuar varias operaciones al mismo tiempo por medio de subsistemas que almacenan resultados en la memoria general.

Palabras Clave

Paralelismo, procesador SIMD, ALU, números enteros, Logisim, Memoria.

INTRODUCCIÓN

"SIMD represents one of the earliest styles of parallel processing. The term SIMD stands for "Single-Instruction Multiple-Data," which aptly encapsulates the parallel processing model. Closely related to vector processing, the basic idea is to operate the same instruction sequence simultaneously on a large number of discrete data sets. SIMD machines are geared toward applications that exhibit massive amounts of data parallelism without complicated control flow or excessive amounts of inter-processor communication. Typical applications for SIMD machines include low-level vision and image processing, discrete particle simulation, database searches, and genetic sequence matching.

Background

The history of SIMD machines began with the ILLIAC IV project, started in 1962. The machine was the first large-scale multiprocessor, composed of 64 64-bit processors. The project itself was pretty infamous for its failure; estimated costs of \$8 million ballooned to \$31 million by 1972. The actual performance of 15 MFLOPS was far below the original estimates of 1000 MFLOPS, partially because only a quarter of the

planned machine was ever constructed. In addition, the machine took another 3 years of engineering to actually work following its delivery to NASA in 1972. Needless to say, the project slowed interest and investigation of SIMD architectures for quite a while. Eventually, Danny Hillis resurrected the SIMD architecture in 1985 with his Connection Machine. However, following a short stint in the 80's by several commercial companies such as Thinking Machines and MasPar, SIMD has once again fallen by the wayside in the arena of commercial general-purpose computing."[1]

Procesamiento de vectores

Como todos sabemos, uno de los grandes problemas de los programas son los ciclos: no hay nada más demorado en la programación que tener un ciclo para recorrer un vector y obtener un resultado. La meta es que se pueda diseñar un procesador que pueda realizar una operación que opere en paralelo.

OBJETIVOS

1. Objetivo General
Realizar el proceso completo de diseño de un procesador SIMD que pueda calcular la suma de los valores de un vector finito.
2. Objetivos Específicos
 - a. Implementar un procesador con las instrucciones que se especifican a continuación.
 - b. Implementar una instrucción básica de procesamiento paralelo.

ESPECIFICACIONES

1. Tipo de datos: Enteros de 8 bits sin signo
2. Operaciones a implementar: Ustedes decidirán cuales implementar pero como mínimo deberán tener:

- a. Operaciones de transferencia de datos
- b. Operaciones aritméticas (la suma)
- 3. Operación extra a implementar:
 - a. SUMAV Posición, Tamaño
Toma todas las posiciones de un vector que está ubicado en memoria, las suma y deja el resultado en un registro del procesador.

ANOTACIONES GENERALES DE ENTREGA E HITOS A CUMPLIR

Anotaciones generales de la práctica

1. La práctica se puede hacer individual o en grupos de hasta máximo tres personas. Los nombres de las personas que conforman cada grupo deben ser informados al profesor antes del jueves 8 de octubre.
2. La totalidad de la práctica se realizará en Logisim [2].
3. Los informes finales de la práctica se entregarán en un archivo siguiendo el formato de la IEEE para la publicación de artículos, el cual se puede descargar desde EAFIT Interactiva.
4. La entrega se realizará por EAFIT Interactiva (no se admiten por correo electrónico) adjuntando dos archivos:
 - a. El archivo con extensión .circ generado por cada grupo
 - b. Un archivo tipo documento Word o PDF con los siguientes elementos:
 - i. Los detalles de diseño encontrados durante la construcción del módulo.
 - ii. Las especificaciones del módulo entregado, es decir: los criterios de diseño del trabajo.

- iii. Dificultades encontradas durante el diseño de la práctica
- iv. Opinión personal sobre la práctica.

5. Fecha de entrega máxima: Lunes 9 de noviembre hasta las 6:00 p.m. a través de Eafit Interactiva
6. Sustentación: La sustentación se realizará por grupos, de cada grupo se elegirá un representante responsable de realizar la sustentación. Cada grupo tendrá 30 minutos para su sustentación. Al momento de la sustentación deberán estar **todos** los estudiantes del grupo presentes, de donde se elegirá el representante responsable de sustentar. El objetivo de la sustentación es determinar si los estudiantes hicieron o no la práctica.
7. Avances: Los avances se empezarán a enviar a partir del día 13 de octubre y cada martes hasta el último el día martes 3 de noviembre. El avance consta de lo que se ha hecho durante la semana de trabajo referente a la práctica.
8. Criterio de calificación:
 - a. 0.0: No se entrega nada o la sustentación demuestra que no se hizo la práctica.
 - b. 1.0 a 3.0: Se entregan las operaciones funcionando parcial o totalmente sin la operación extra a implementar.
 - c. 3.0 a 4.0: Se entregan las operaciones a y b funcionando correctamente y se entrega la operación extra sin que esta funcione.
 - d. 5.0: Se entregan las operaciones a, b y la extra funcionan correctamente y un programa funciona correctamente.

REFERENCIAS

- [1] <http://www.ai.mit.edu/projects/aries/papers/writeups/darkman-writeup.pdf> visitado el sábado 26 de septiembre de 2009
- [2] <http://ozark.hendrix.edu/~burch/logisim/> visitado el sábado 26 de septiembre de 2009