

# Entrega final práctica 2

## Organización de Computadores

Sebastián Arcila Valenzuela (*sarcilav@eafit.edu.co*), Sergio Botero Uribe (*sbotero2@eafit.edu.co*).

**Resumen**—Entrega final de la práctica 2 de Organización de computadores, el informe contiene los detalles de la construcción del procesador así como las especificaciones de la solución planteada, las dificultades encontradas y los comentarios por cada uno de los integrantes del equipo sobre la práctica.

### I. INTRODUCCIÓN

Para esta práctica no se pudieron completar con todos los requerimientos ya que con la aproximación que se decidió implementar no se pudo terminar la implementación de la sumatoria de vectores. El problema parece estar en el intercambio de las señales entre los componentes que se propusieron para la solución.

Las operaciones básicas funcionan sin inconvenientes pero la instrucción de suma de vectores crea oscilaciones en el simulador, oscilaciones que se esperan resolver para el día de la sustentación con la intención de no dejar el montaje de los componentes sin uso.

### II. DETALLES DE DISEÑO DE CONSTRUCCIÓN

Para esta práctica se tuvieron varias ideas principales, algunas de ellas se basaban en un procesador completamente diferente, otras tenían como idea principal el uso de un BUS más amplio, otras sólo pensaban sobrecargar una unidad aritmético-lógica mientras que otras mucho más arriesgadas trataban de incorporar algo de todas.

Algunas aproximaciones fueron:

- La primera aproximación fue la de crear una ALU que tuviera cuatro entradas y dos salidas de registro, esto llevaría a usar unos 8 registros para poder tener algunos para propósito general y evitar el hecho de que todo el procesador se viera involucrado en la operación de sumatoria de los valores del vector. Esta idea empezó a demostrar que era bastante complicada ya que el número de señales de la unidad de control se incrementaba para el manejo de las entradas y salidas de los registros. Hasta este momento no habíamos encontrado un error que después haría que se cambiara todo el procesador.
- Para atacar el problema de las señales de la unidad de control, se pensó en sobrecargar una ALU, hacer que la operación de suma de vector se manejara completamente allí, que se tuvieran un conjunto de registros exclusivos para esa operación y que las señales fueran totalmente independientes de la microprogramación. El error ahora sería que las señales de la unidad de control se interrumpirían con las que crearía la ALU. Hasta aquí todavía nos estábamos centrando en la ALU como el componente que haría la diferencia en el paralelismo.

- Del resto de las ideas y los diseños, la que parecía ser la que daría mejores resultados fue la de implementar una ALU con ciertos registros independiente de la normal que además sería controlada por una unidad de control independiente. Esta unidad de control debía ser activada por la unidad de control del procesador y que luego esta le pasara el 'control' nuevamente a la unidad del procesador. A este punto ya nos habíamos dado cuenta que los diseños anteriores no estaban atacando el problema para lograr paralelismo sino que sólo se estaban haciendo muchas sumas en un Tick de reloj, pero nada de paralelismo.

### III. ESPECIFICACIONES DEL PROCESADOR

El procesador implementado tiene dos unidades de control, una que controla el procesador para operaciones normales y otra unidad de control que se encarga específicamente de la ALU sobrecargada, el cambio de control de las unidades se hace mediante las señales que salen de la instrucción de suma del vector.

Una vez que la unidad de control del procesador pasa el control a su homólogo de la ALU, esta se encarga de traer los valores del vector de forma consecutiva para guardarlos en los registros de la ALU y cada vez que se traen 4 posiciones del vector se hace la suma de todos los valores, esto se hace en dos peticiones a la RAM.

### IV. DIFICULTADES ENCONTRADAS DURANTE LA CREACIÓN DEL PROCESADOR

El concepto de paralelismo para la suma de los elementos de un vector tiene ciertas implicaciones a nivel de implementación que desde el principio no se tuvieron claras. También hay que tener en cuenta que el diseño de un procesador no es como normalmente tengo la costumbre de empezar las prácticas (tipo Extreme Programming) sino que se debe diseñar primero una aproximación y empezar a crear unidades por partes separadas para luego unir las todas. Ese fue el error que tuvo en general el equipo de trabajo para poder cumplir con todos los puntos de la práctica.

-Sergio Botero Uribe

Logisim es una herramienta aunque de código abierto que ha tenido un desarrollo muy centralizado y que no es muy flexible y además muy poco usable, por lo que no nos rindió mucho el trabajo.

-Sebastián Arcila Valenzuela

## V. OPINIONES SOBRE LA PRÁCTICA

Toda la forma de pensar que se requiere para modelar el procesador y para sincronizar todas las señales en una unidad de control microprogramada considero que es parte importante en la lógica mínima y general que debe tener un ingeniero de sistemas.

Aunque caímos en el error de dejar la parte de diseño para lo último, se hubiera podido sacar una de tantas aproximaciones, me parecería muy interesante conocer todas esas formas, para saber cual de todas las ideas de implementación podían resultar y cuales ofrecen mejor rendimiento.

Por último, estoy cansado de usar el mouse y de hacer tanto click.

*-Sergio Botero Uribe*

Las herramientas en Java una vez mas me volvieron a sorprender con su increíble usabilidad y velocidad, es más después de esta práctica quedamos listo para dictar un curso de paciencia hacia Java. Espero que esta sea mi ultima experiencia con Java, que invento quien dijo que Java era un lenguaje de propósito general. Por otro lado nunca me han gustado los trabajos que sean de desarrollo en aplicaciones que sean puro click y más clicks.

*-Sebastián Arcila Valenzuela*