Bitácora de trabajo del proyecto 1

Participantes

1. Fabian Astorga
2. Javier Francisco Sancho
3. Oscar Ulate Alpízar

Tabla de bitácora

Fecha	Horas	Integrantes	Descripción
12/9/17	5	Javier, Oscar	1. Elegir los elementos del ISA: memoria, cantidad de registros, tipos de registros, instrucciones: 1. Entre otras cosas addressability y cantidad de bits por registro. 2. Comprender el algoritmo de fuerza bruta y Karp-Rabin para ver qué instrucciones se necesitan en la arquitectura. 3. Surgieron muchas dudas, como: 1. Manejo de labels y saltos 2. División y multiplicación en la arquitectura necesario para los algoritmos de búsqueda de patrones.
12/9/17	5	Fabián	1. Investigación sobre Flex para la creación del análisis sintáctico del compilador. 2. Implementación de analizador básico en C.
16/9/17	2	Oscar	1. Intento de programar el algoritmo de comparación de textos mediante Fuerza Bruta en ARM para de esa manera conoce qué instrucciones debe soportar la microarquitectura.
18/9/17	3	Javier, Oscar	1. Discusión sobre qué tipo de memoria se va a utilizar para las instrucciones en el Fetch. Además se analizó el tipo de recorrido que debe hacer el PC en una instrucción básica tipo R.
18/9/17	4	Fabián	1. Implementación del compilador: Se creó un archivo lex con todas las reglas que iba a tener el compilador

			para detectar los parámetros. Al detectar el compilador una regla, se retorna un calor constante en un archivo header.
23/9/17	4	Fabián	1. Implementación del compilador: Continuación de implementación del compilador.
26/9/17	2	Fabián	1. Implementación del compilador: Continuación de implementación del compilador.
2/10/17	3,5	Oscar	1. Implementación de la microarquitectura: Comienzo de programación de módulos básicos del Fetch. Se decide dejar la unidad de control para el final de la programación. 2. Se investiga sobre implementaciones que puedan funcionar de base sobre todo pensando en las memorias que no se tiene claro cómo resolver.
3/10/17	4	Oscar	1. Implementación completa del Fetch y Decodificador de la microarquitectura. Aún hay ciertos problemas y dudas con respecto a las banderas y los ciclos de reloj.
4/10/17	6	Fabián, Javier, Oscar	1. Revisión del Fetch y Decodificador: Depuración de etapas en cuanto a conexión entre los módulos. 2. Se discute sobre cómo se va a solucionar los problemas con el espacio de memoria pensando en la memoria de instrucciones y la de registros. 3. Se comienza a realizar un documento formal con las principales ideas e instrucciones.
5/10/17	1,5	Fabian	1. Implementación de módulos sobrantes para el Decodificador: Sign Extension, bit adder.
6/10/17	3	Oscar	1. Se implementó la ALU del microprocesador.

6/10/17	5	Javier	1. Se crearon los módulos “top” del microprocesador, donde se instancian los módulos de los componentes más básicos y se le da unidad al módulo.
8/10/17	6	Fabián, Javier, Oscar	1. Se trabaja y se completan los documentos que se deben entregar el lunes 9 de octubre.
9/10/17	2	Javier, Oscar	1. Implementación de los registros IF/ID y ID/EX.
11/10/17	4	Fabián	1. Trabajo en el compilador para dar funcionamiento a las etiquetas de direcciones de memoria.
11/10/17	5	Javier, Oscar	1. Implementación de etapas de Memoria y Write Back. 2. Implementación de registros EX/MEM y MEM/WB.
13/10/17	2	Javier, Oscar	1. Discusión y análisis sobre el manejo de los ciclos de reloj y banderas de control. 2. Oscar realizó una tabla con todas las banderas y sus valores para todas las instrucciones de la microarquitectura.
13/10/17	3	Javier	1. Implementación de la unidad de control tomando en cuenta todo lo analizado y discutido en conjunto.
14/10/17	3	Oscar	1. Implementación del algoritmo de “Fuerza Bruta” para que fuera corrido en el procesador creado.
16/10/17	7	Javier	1. Arreglos de errores y warnings de compilación para que fuera posible simular el procesador y hacerle pruebas. Además de hacer pequeños cambios requeridos en la lógica de programación.
16/10/17	2	Fabián	1. Creación de algoritmo para convertir código de binario a Intel-Hex para que pueda ser leído por la FPGA.
16/10/17		Oscar	1. Implementación del código de 7 segmentos para mostrar resultados en la FPGA.