Componente de arquitectura RISC

Maxime Ishin Montealegre Terada
Luis Ricardo Luna Lopez
Hugo Santiago Gómez Salas
Daniel Hernández Cárdenas

Departamento de ingeniería en sistemas computacionales, ESCOM IPN

maxime.montealegre@gmail.com
dany2501.dhn@gmail.com
ricardo1426@hotmail.com
hugogsalas79@gmail.com

***Resumen*— La realización de esta práctica se basa en realizar un componente de la Arquitectura RISC, la cual con diapositivas, videos, y ejemplos mostrados en clase se pudo comprender para así poder realizarla y plasmarla en el lenguaje VHDL. Se tomó la decisión de realizar una Unidad de Control Alambrada, debido a que fue una de las mejores analizadas y entendidas de las 4 opciones que se podían realizar. En este documento se hablará de dos puntos los cuales son la base de esta práctica; la Arquitectura RISC y la Unidad de Control Alambrada. Se explicara cuales son sus características, cuales son sus ventajas al momento de utilizarlos, y sobre todo cómo es que estos dos conceptos van relacionados, además de comentar su importancia al momento de implementar el uno con el otro. De igual modo se mostrara las pruebas de cómo funciona este componente, mostrado ejemplos de cómo funciona el código ya realizado y funcionando.**

***Palabras Clave* — RISC, Pipeline, Flitch.**

I. Introducción

La Arquitectura RISC empezó en la década de 1970, desarrollada por IBM y las universidades de Stanford y Berkeley. John Cocke de IBM, fue quien demostró que el 20% de las instrucciones de un ordenador podían realizar el 80% del trabajo.

La Arquitectura tipo RISC está diseñada para ejecutar un número reducido de tipos de instrucciones como las aritméticas, las lógicas y las de control de flujo, permitiendo operar a una velocidad más elevada, ocupando menos espacio en los bloques lógicos, pudiéndose hacer mucho más pequeños. Un ejemplo de esta arquitectura la podemos ver reflejada en la figura 1.

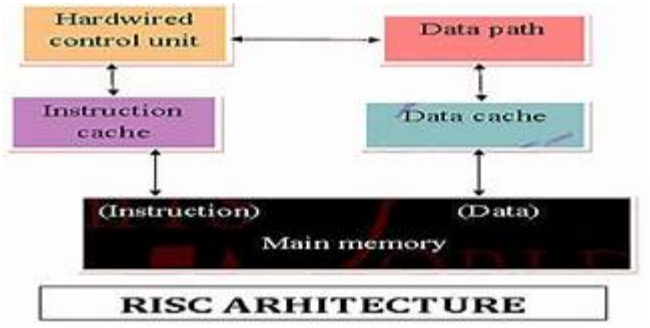


Figura 1. Arquitectura RISC.

Utiliza una tecnología del tipo pipeline muy desarrollada, es decir, que permite a los compiladores organizar las instrucciones de modo que mantengan llenos los conductos. Suelen ser una arquitectura basada en registros de propósito general que operan siempre sobre operandos que se encuentran almacenados en el procesador, cerca de la unidad de ejecución.

Las instrucciones simples que realiza este tipo de arquitectura se puedan realizar simultáneamente, de un modo más simple y eficaz. Su objetivo no es ahorrar esfuerzos externos por parte del software con sus accesos a la RAM, sino facilitar que las instrucciones sean ejecutadas lo más rápidamente posible. Sus instrucciones más breves y sencillas son capaces de ejecutarse mucho más rápido que las instrucciones más largas y complejas de un chip CISC, pero por el contrario requiere de mucha más RAM y de una tecnología de compilador . En la figura 2 podemos ver de forma gráfica las diferencias entre una Arquitectura RISC y una CISC.

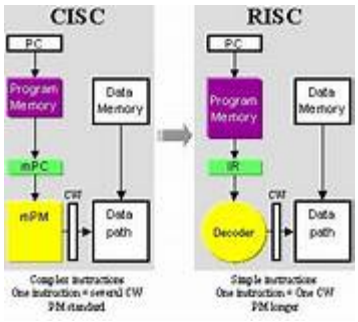


Figura 2. Comparación entre una Arquitectura RISC y CISC.

Sus principales características serían:

- La Arquitectura RISC utiliza solo un ciclo único de ejecución permitiendo sintetizar secuencias de múltiples instrucciones para operaciones menos frecuentes.
- No utiliza algún microcódigo, ya que estos lo que hacen es agregar capas de sobrecarga operativas provocando que instrucciones sencillas requieren varios ciclos de cómputo.
- Simplifican las instrucciones complejas y los modos de direccionamiento.
- Únicamente carga y almacena la memoria de acceso.
- Modelo de conjunto Cargar-Almacenar.

Sus ventajas son generar un mejor rendimiento gracias al número simple y limitado de instrucciones, además de generar un espacio libre en el encapsulado que usa para integrar otros circuitos, y por último el menor consumo energético y de calor. Por el contrario algunos de sus inconvenientes serían que su rendimiento varía según el software, su mayoría de software disponible se basa en instrucciones complejas, además de necesitar memorias muy rápidas para guardar varias instrucciones, haciendo necesaria mucha memoria caché en los ordenadores. En la figura 3 podemos analizar el proceso de instrucciones de una arquitectura RISC.

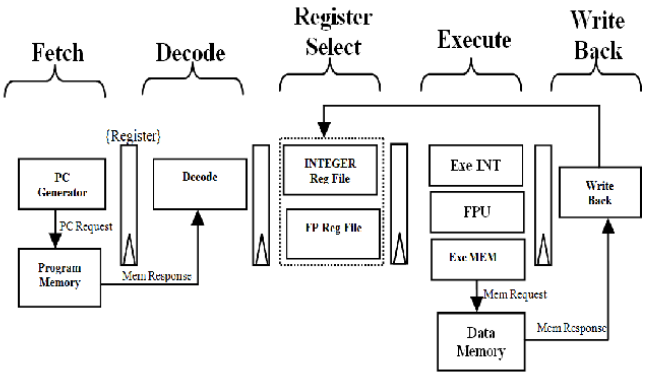


Figura 3. Diagrama de una Unidad de Control.

Una unidad de control como sabemos se encarga de sincronizar las acciones de cada una de las unidades funcionales, además de decodificar los códigos de operación y el direccionamiento de las instrucciones temporizado las operaciones necesarias. Una Unidad de Control Alambrada es un circuito secuencial el cual se basa en compuertas y componentes, caracterizándose por ser fijas, es decir, si se solicita una nueva instrucción dentro del circuito ya realizado, todo el circuito debe ser nuevamente diseñado. Esta unidad combina un control cableado con microcódigo para instrucciones complejas.

La idea general es generar señales por medio de circuitos secuenciales. Su velocidad es mayor que el micro programado, debido a esto algunos procesadores RISC la utilizan. En la figura 4 volvemos a recordar cómo es que está esquemáticamente elaborada una Unidad de Control.

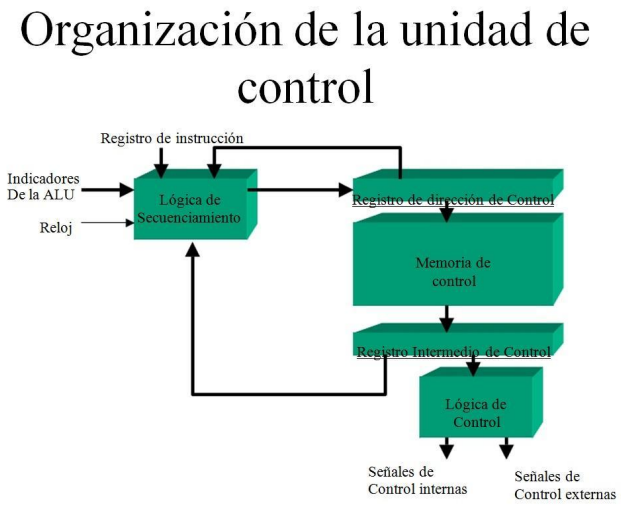


Figura 4. Unidad de Control.

Consta de un decodificador N a 2 elevado a la N, donde N es el tamaño del registro, un arreglo de compuertas, un contador Johnson y una señal de reloj.

Forma dos registros, un contador binario y uno de almacenamiento de 4 bits. En el primero almacena la dirección de la siguiente instrucción a ejecutar, mientras que en el segundo almacena la instrucción que se debe ejecutar, esta instrucción sigue hasta que termina el ciclo de ejecución. y es actualizado por el ciclo Fitch.

II. Metodología/Desarrollo

A. En ésta práctica, se reutilizará lo trabajado en la primera práctica, modificándolo para poder utilizar la Unidad de Control de forma que el programa ya no necesite del uso de un reloj para poder realizar operaciones de forma más directa.

B. Para poder almacenar datos, utilizamos una memoria FIFO (First In, First Out), el cuál nos permite manejarla de una forma más rápida y sencilla, ya que solamente se utiliza "pop y push".

C. La ingesta de datos se realizó de manera directa, de tal forma que se utilicen inmediatamente los datos para los procesos/operaciones que se quieran realizar..

D. Para ésto, la Unidad de Control se modificó para que fuera de forma alambrada,la cual se caracteriza por no requerir de un reloj como el de la práctica pasada, y todo es conectado de forma física, lo cual permite que la UC pueda controlar las operaciones.

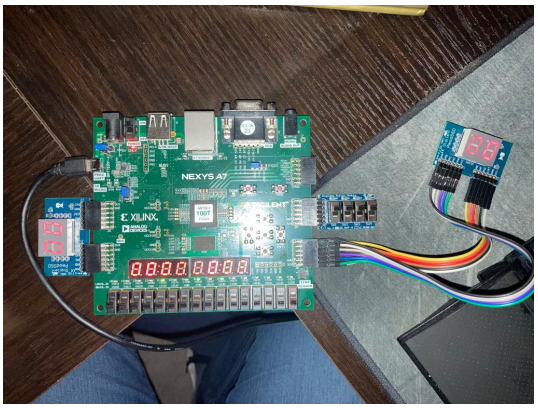
E. Se desconectó entonces el reloj, para que de esta forma, dejara de manejarse de manera síncrona.

F. Una de las grandes ventajas de utilizar éste método, es que es más sencillo de programar, además de que es más rápido para tareas sencillas, sin embargo, éste tipo de UC alambrado, no nos permite optimizar los procesos como podríamos hacerlo si utilizamos una UC.

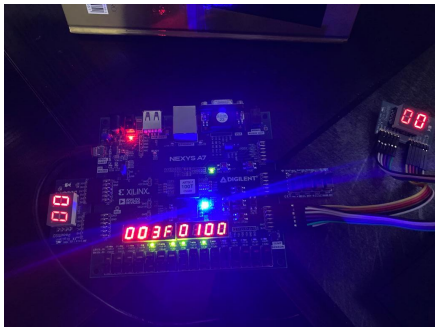
III. Resultados y discusión

Para poder observar EL funcionamiento de los resultados de la práctica se tomó como evidencia la serie de pasos a seguir para comprender cómo funciona y se plasmó en las siguiente imágenes.

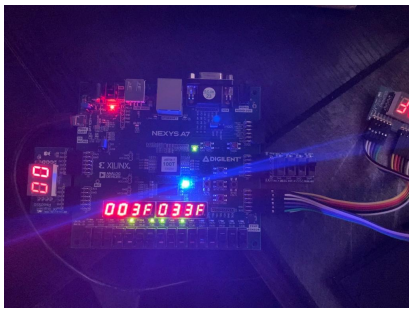
1. El sistema corriendo la Unidad de control.



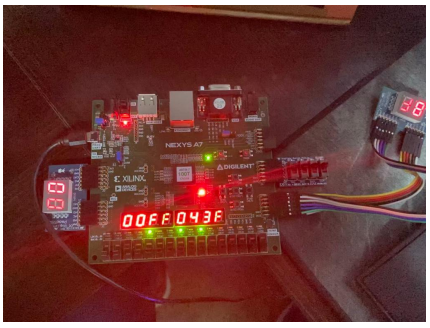
2. Primera instrucción. Carga la entrada del registro AX.



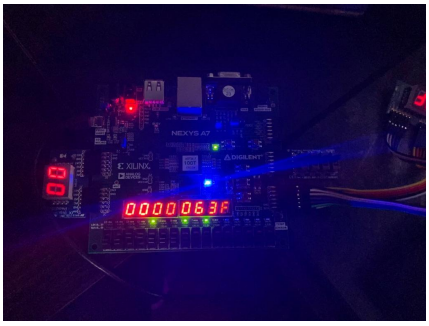
3. Mueve el registro AX a la memoria RAM.



4. hace uso de la instrucción suma. Suma la entrada de datos y el registro AX. El resultado se colocará en AX.



5. Corrimiento hacia la izquierda del registro AX.



IV. Conclusiones

Durante la realización de la práctica, pudimos observar principalmente los distintos tipos de UC que podemos utilizar, dependiendo del tipo de procesos u operaciones que queremos llevar a cabo, las alambradas y las microprogramadas, enfocándonos en esta ocasión en el tipo alambrado, el cuál es más sencillo de implementar y suele ser más rápido para realizar tareas sencillas, durante cada sesión con el profesor, fuimos estudiando y analizando las características de éste tipo de UC para poder implementarlo en ésta práctica, como el manejo de una memoria FIFO.

Referencias

[1] Arquitectura del Computador (2014). “Diseño de Logica de Control”. Disponible en: https://laarquitecturadecomputadoras.blogspot.com/2014/11/diseno-de-logica-de-control_19.html

[2] M. en C. Najera Medina Gabriela (1996). “Principios de una Arquitectura RISC”. Disponible en: http://cidecame.uaeh.edu.mx/lcc/mapa/PROYECTO/libro20/13_principios_de_arquitectura_risc.html

[3] Roberto Soler (2021). “RISC: La arquitectura de procesadores usada por ARM para cambiar el mercado”. Disponible en: https://www.profesionalreview.com/2021/07/17/que-es-risc/