	Carátula para entrega de prácticas	
	Facultad de Ingeniería	Laboratorios de docencia

Laboratorio

<i>Profesor:</i>	ING. JULIO CESAR CRUZ ESTRADA
<i>Asignatura:</i>	LABORATORIO DE ORGANIZACIÓN Y ARQUITECTURA DE COMPUTADORAS
<i>Grupo:</i>	07
<i>No de Práctica:</i>	8
<i>Integrante(s):</i>	Jiménez Treviño Emilio Cristóbal Martinez Perez Brian Erik
<i>No. de Equipo de cómputo empleado:</i>	s/n
<i>Semestre:</i>	2026-1
<i>Fecha de entrega:</i>	21/11/2025
<i>Observaciones:</i>	

CALIFICACIÓN: _____

Práctica 8. Procesador RISC

Objetivo

Diseñar un microprocesador RISC, específicamente la versión de pipeline del microprocesador 68HC11 de Motorola R .

Introducción

En esta práctica se trabaja con una versión RISC en pipeline del microprocesador 68HC11 de Motorola, con el propósito de analizar su comportamiento y compararlo con la versión CISC. Para ello, se implementan y ejecutan programas sencillos en ambas arquitecturas, observando sus señales internas y midiendo los tiempos de carga de ciertos datos en los registros de trabajo. De esta forma, se busca comprender de manera experimental cómo influye el diseño del conjunto de instrucciones y el uso de pipeline en la eficiencia del procesador.

Desarrollo

Para comenzar primero descargamos el proyecto de Quartus del procesador RISC que nos proporcionó el profesor por classroom.

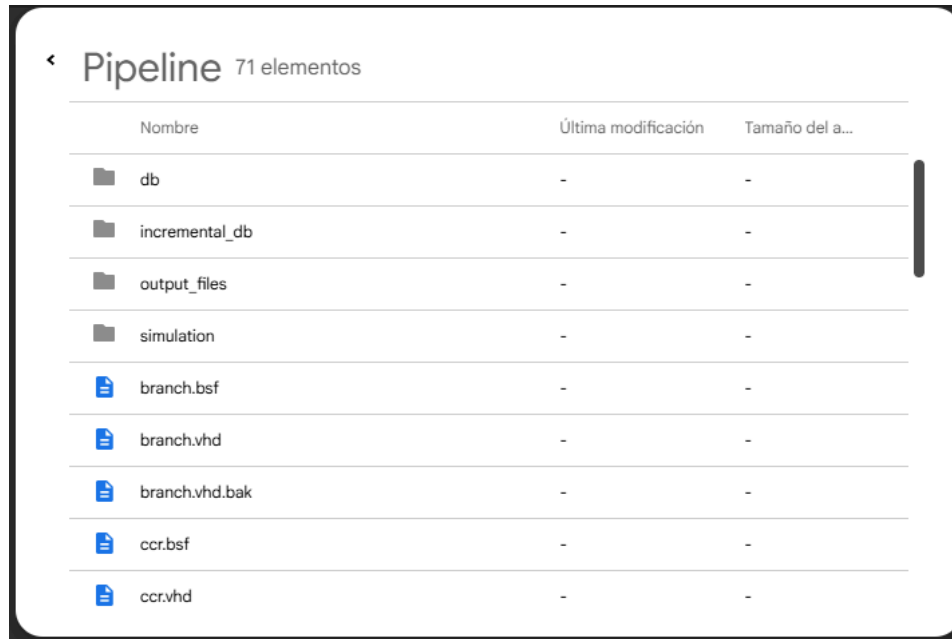


Imagen 1.1 - archivos del proyecto de Quartus “Pipeline”.

Una vez que descargamos el proyecto lo descomprimos y abrimos el proyecto para abrirlo con Quartus y compilarlo en opción de “Start Analysis & Synthesis”.



Imagen 1.2 - compilación con opción “Start Analysis & Synthesis”.

Si podemos ejecutarlo y no genera errores, significa que podemos implementar nuestra propias instrucciones en la memoria “rom” y cargar códigos para que el procesador los ejecute.

Procesador CISC

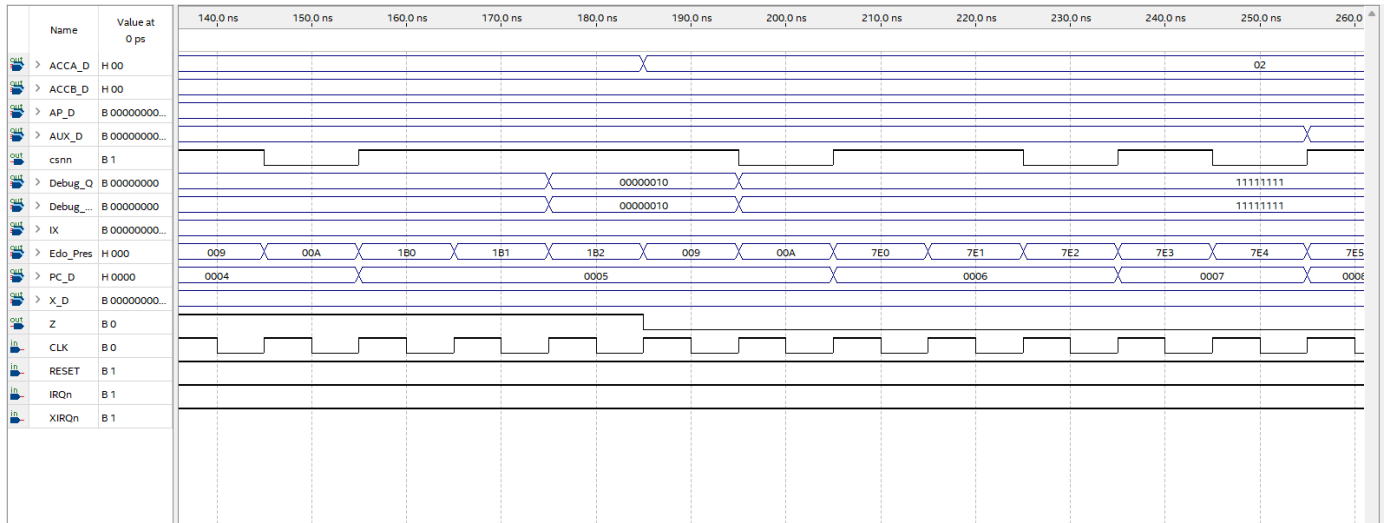


Imagen 1.3 - cargado del dato "02" con procesador CISC.

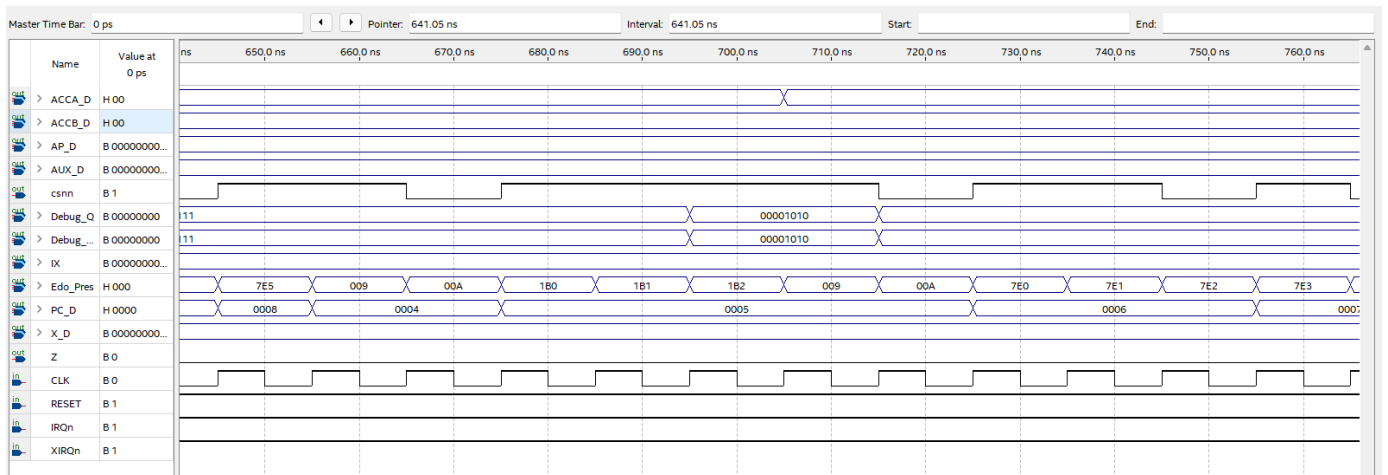


Imagen 1.4 - cargado del dato "0A" con procesador CISC.

Procesador RISC

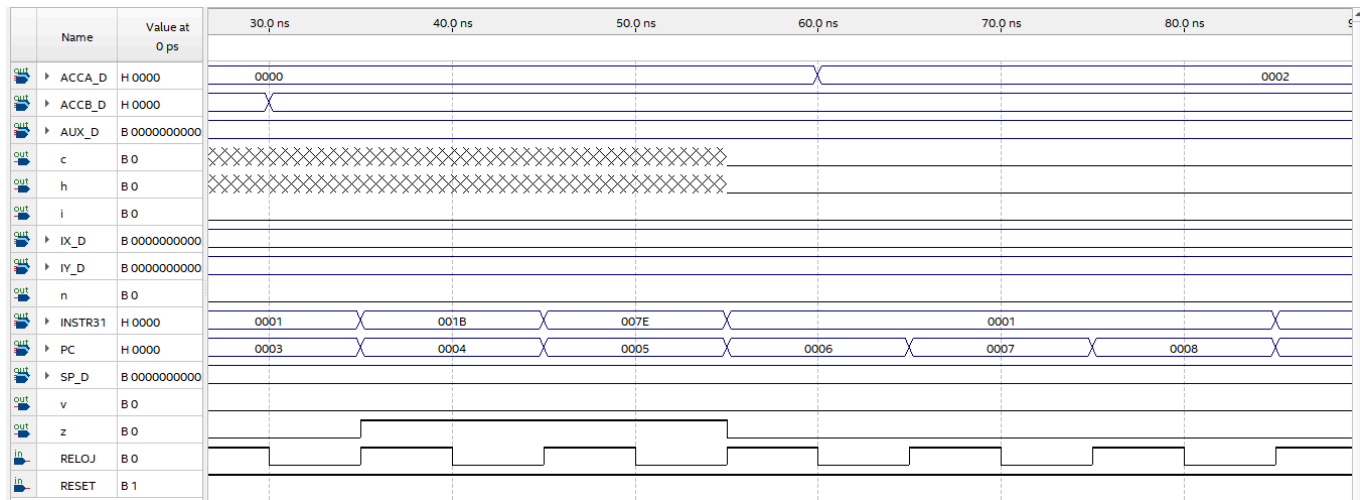


Imagen 1.5 - cargado del dato “02” con procesador RISC.

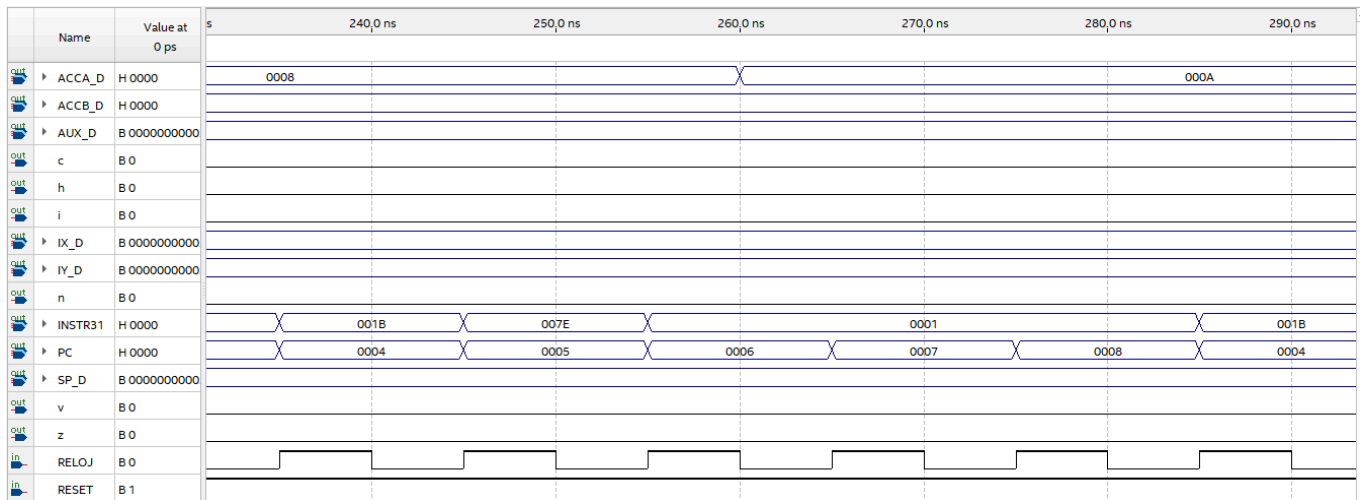


Imagen 1.6 - cargado del dato “0A” con procesador RISC.

El objetivo de esta actividad es evaluar la eficiencia de las arquitecturas RISC y CISC. Para ello, la actividad se dividió en dos casos: el caso 1, que mide el tiempo que tarda cada procesador en cargar el dato 02; y el caso 2, que mide el tiempo de carga del dato 0A, correspondiente al resultado final del algoritmo de suma de 2 en 2. Los resultados obtenidos se analizarán desde dos perspectivas: el porcentaje de velocidad en comparación con RISC, y la relación de cuántas veces RISC es más rápido que CISC.

CASO 1:

$$EF_{CISC} = (60ns/185ns) * 100$$

$$EF_{CISC} = 34.43\% \text{ de la velocidad del RISC}$$

$$EF_{RISC} = (185ns/60ns) * 100$$

$$EF_{RISC} = 3.08 \text{ veces más rápido que CISC}$$

CASO 2:

$$EF_{CISC} = (260ns/705ns) * 100$$

$$EF_{CISC} = 36.87\% \text{ de la velocidad del RISC}$$

$$EF_{RISC} = (705ns/260ns) * 100$$

$$EF_{RISC} = 271.15 \text{ veces más rápido que CISC}$$

Conclusiones

Jiménez Treviño Emilio Cristóbal

En esta práctica pude observar como el manejo de las instrucciones de cada arquitectura del procesador afecta muchísimo al eficiencia del programa, durante el desarrollo tuve problemas al correr ambos programas por que mi version del procesador CISC tenia unos problemas con la carga de instrucciones, lo solucione viendo que no estaba cargando bien la memoria entonces se pudo arreglar, al momento de hacer la actividad note como la eficiencia del RISC muy superior a la CISC por que esta usa pipeline permitiendo que las instrucciones se ejecuten en paralelo mejorando así el rendimiento.

Martinez Perez Brian Erik

Esta práctica me ayudó a comprender el concepto de *pipeline* en el procesador RISC como estrategia para aumentar el rendimiento del procesador, permitiendo que múltiples instrucciones se ejecuten simultáneamente en diferentes etapas. aunque no lo hace de forma paralela ya que este comienza a ejecutar otra instrucción terminando alguna de las demás etapas del ciclo fetch. En clase solo hicimos un ejemplo en el que comparamos la ejecución de un mismo programa pero en procesadores con arquitectura distinta. La arquitectura CISC ejecuta las instrucciones de forma secuencial, mientras que la arquitectura RISC ejecuta las instrucciones con la técnica de pipeline. Por último comparamos el tiempo en la carga de operandos y el procesador con la arquitectura RISC tuvo un mejor rendimiento en tiempo.

Bibliografía

Laboratorio de Organización y Arquitectura de Computadoras. (2019, octubre 29). *Práctica No. 8 Procesador RISC*.