

Instituto Tecnológico de Costa Rica
Escuela de Ingeniería en Computadores



Taller de Diseño Digital

Estudiantes:

Castro Moreno Henry Andrés – 2022026502

Blanco Coto Andrés Enrique- 2022108841

Bolaños Barboza Gabriel - 2022327511

Rivera Mora José Ignacio - 2022227827

Documentación de herramientas de ingeniería

Prof. Luis Alonso Barboza Artavia

I Semestre 2025

Introducción

Este documento tiene el propósito de analizar y sintetizar el uso de las habilidades empleadas para desarrollar, elegir e implementar técnicas y herramientas de ingeniería aplicadas al diseño de sistemas digitales. El objetivo de este proyecto es diseñar y desarrollar un computador mínimo, basado en un microprocesador con arquitectura ARMv4 de un procesador en base que permita la aplicación de una versión simple del juego PONG, donde dos jugadores controlan paletas para devolver una bola en movimiento horizontal. El sistema consta del cargamento y decodificación de instrucciones, ejecución de datos en una ALU específica, y salida de los resultados en una pantalla de monitor VGA.

A través de esta arquitectura modular, se logró un desarrollo más organizado y alineado con un proceso de depuración y prueba, tanto de cada módulo individual como del sistema en su conjunto. Además, herramientas como ModelSim y Quartus Lite fueron importantes para la correcta elaboración de un diseño eficiente con alto rendimiento. La metodología no solo cumplió con los requisitos técnicos establecidos, sino que también se enfocó en la optimización del uso de los recursos disponibles, implementando la automatización de procesos críticos.

Las herramientas utilizadas, como ModelSim y Quartus Lite, jugaron un papel crucial en la creación de un diseño eficiente con alto rendimiento. ModelSim permitió simular señales digitales y verificar la interacción entre los módulos del procesador, mientras que Quartus Lite facilitó la síntesis del diseño y la implementación del sistema en la FPGA DE10-Standard. Esta integración de herramientas permitió una transición fluida desde la simulación hasta la implementación real en hardware, lo que aseguró que el diseño estuviera alineado con los objetivos de rendimiento y funcionalidad requeridos.

La optimización de los recursos fue otro aspecto central del proyecto, donde se tomaron decisiones estratégicas para seleccionar operaciones eficientes que minimizaran el uso de memoria y ciclos de reloj durante el procesamiento. La búsqueda de la eficiencia en el uso de los recursos disponibles no solo ayudó a reducir los costos y el tiempo de desarrollo, sino que también permitió cumplir con los requisitos técnicos de rendimiento establecidos para el diseño del procesador ARMv4 y la interacción con el juego PONG.

La validación de los módulos del procesador fue realizada con ModelSim, mientras que la síntesis del diseño en la FPGA DE10-Standard se llevó a cabo utilizando Quartus Lite. Este enfoque integró tanto aspectos teóricos como prácticos, destacándose por las aplicaciones innovadoras dentro del ámbito del diseño digital, específicamente en la implementación de un procesador ARMv4 y la interactividad del juego PONG.

Selección de técnicas, recursos, herramientas o métodos acorde con las variables del problema complejo de ingeniería.

Para abarcar las necesidades del proyecto e identificar que el diseño posea las características técnicas y funcionales necesarias es necesario determinar las técnicas, instrumentos y recursos más adecuados a utilizar.

Técnicas.

La metodología de un diseño modular permite trabajar el sistema por medio de distintas etapas con funcionalidad independiente.

La metodología adoptada para el diseño del sistema es modular, lo que permite desarrollar e implementar el proyecto de manera escalonada y con funcionalidades independientes. Esta estructura facilita tanto la depuración de cada módulo como su integración al sistema completo.

- Sincronización y control de señales: Se implementaron mecanismos de sincronización entre los módulos del procesador para garantizar que todas las señales operaran de forma coordinada, reduciendo los posibles errores de temporización y mejorando la estabilidad del sistema.
- Optimización de recursos: Se seleccionaron operaciones eficientes con el fin de minimizar el uso de memoria y ciclos de reloj durante el procesamiento, garantizando un rendimiento óptimo del sistema.
- Diseño escalable: Permite trabajar en módulos independientes (Decodificador, Lógica Condicional, Ruta de Datos, ALU y Registro de Datos) y probarlos antes de la integración, facilitando una implementación más flexible y eficiente.

Herramientas

- Cputator: Elegida por su capacidad para simular y probar el procesador ARMv4 en un entorno controlado, Cputator permite ejecutar el código ensamblador directamente sobre la arquitectura ARM virtualizada, ofreciendo una plataforma sencilla para validar el comportamiento del procesador y sus interacciones sin necesidad de hardware físico.
- ModelSim: Se seleccionó ModelSim debido a su capacidad para simular señales digitales, permitiendo la verificación y validación de cada módulo de manera independiente. Esta herramienta facilita la depuración en tiempo real y proporciona una visualización detallada de las interacciones de señales, lo que mejora la precisión del proceso de diseño y asegura el funcionamiento adecuado de los módulos antes de su integración.
- Ripes: Esta herramienta se utilizó para la visualización y simulación gráfica del procesador ARM, facilitando el análisis paso a paso de las instrucciones ejecutadas por el procesador. Su interfaz gráfica permitió una comprensión más clara del flujo de control y la ejecución de las instrucciones, lo cual fue crucial para depurar el diseño y optimizar la interacción con otros módulos del sistema.

- Quartus Lite: Esta herramienta se utilizó para llevar a cabo la síntesis del diseño, generando los archivos necesarios y realizando el análisis de temporización para asegurar que el diseño cumpliera con los requisitos de rendimiento. Su integración con ModelSim permitió un flujo de trabajo continuo, desde la simulación de los módulos hasta la implementación del diseño en la FPGA, asegurando una transición fluida y sin errores entre la simulación y la implementación final.

Recursos

La selección de recursos se basó, en parte, en la disponibilidad del material facilitado por la universidad.

- FPGA DE1-Soc.
El sistema fue implementado en este dispositivo y desarrollado desde una computadora de escritorio. Asimismo, se utilizaron memorias RAM e instrucciones precargadas, lo que optimizó la ejecución de los filtros y facilitó el procesamiento eficiente del sistema.
- Controlador PS/2 (mouse o teclado).
Se usó en el proyecto planteado para el manejo de dispositivos de entrada como mouse o teclado.
- Salida por VGA.
Salida por cable VGA a monitor para la visualización del juego en la pantalla, completando así la interacción entre el usuario y el sistema.

Aplicación de técnicas, recursos, herramientas o métodos en un problema complejo de ingeniería.

Desarrollo del procesador

El procesador fue diseñado utilizando SystemVerilog y se estructuró en varias etapas funcionales. Véase figura 1.

- a. Lógica Condicional: Su rol es interpretar las instrucciones decodificadas y activar o desactivar las banderas correspondientes para ejecutar las acciones pertinentes.
- b. Registro de Datos: Se ocupa de importar y exportar los registros que se requieran para el procesamiento.
- c. Ruta de Datos: Es responsable de mantener el flujo ordenado al leer las instrucciones, dirigiéndolas hacia el módulo correspondiente, ya sea una nueva instrucción, un registro o una operación de la ALU.
- d. Controlador: Su función principal es gestionar todas las operaciones necesarias del procesador, con especial atención a las tareas del decodificador y la lógica condicional.
- e. ALU: Su tarea es realizar todas las operaciones aritméticas y lógicas necesarias, como sumas, restas, AND y OR.

- f. Decodificador: Se encarga de convertir las instrucciones en formato hexadecimal a binario, dividiéndolas en subsecciones que le indican al procesador cuál operación debe realizarse.

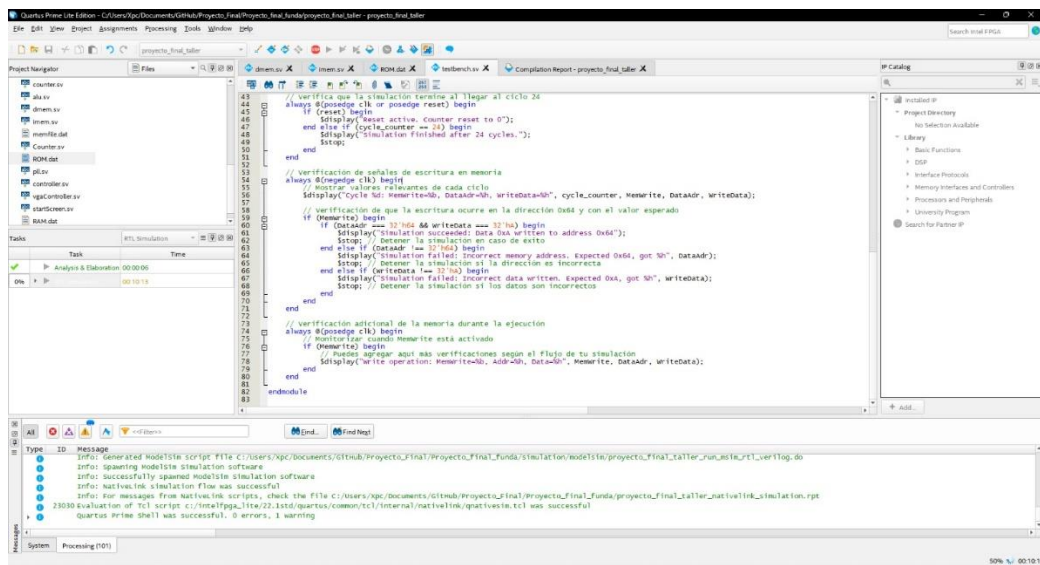


Figura 1. Captura de pantalla de la herramienta Quartus.

Simulaciones en Ripes

Ripes facilitó el análisis paso a paso de las instrucciones ejecutadas por el procesador, permitiendo a los desarrolladores observar cómo se ejecutaban las instrucciones en tiempo real y cómo interactuaban con los diferentes módulos del sistema. Se detalla más a continuación.

- Simulación Gráfica del Procesador: Ripes proporcionó una interfaz gráfica que ayudó a visualizar el comportamiento del procesador ARMv4 a medida que ejecutaba instrucciones. Esto fue crucial para entender cómo fluían los datos dentro del procesador y cómo se gestionaban las operaciones.
- Depuración y Optimización: Ripes permitió depurar el diseño del procesador al mostrar el flujo de control y la ejecución de instrucciones en un formato visual. Esto ayudó a identificar posibles errores o cuellos de botella en el diseño, lo que facilitó la optimización y la corrección de problemas antes de la implementación final.
- Análisis del Flujo de Instrucciones: La herramienta permitió observar el paso a paso de las instrucciones dentro del procesador, proporcionando una visión clara de la ejecución del código. Esto es especialmente útil para entender cómo el procesador maneja diferentes operaciones aritméticas y lógicas.

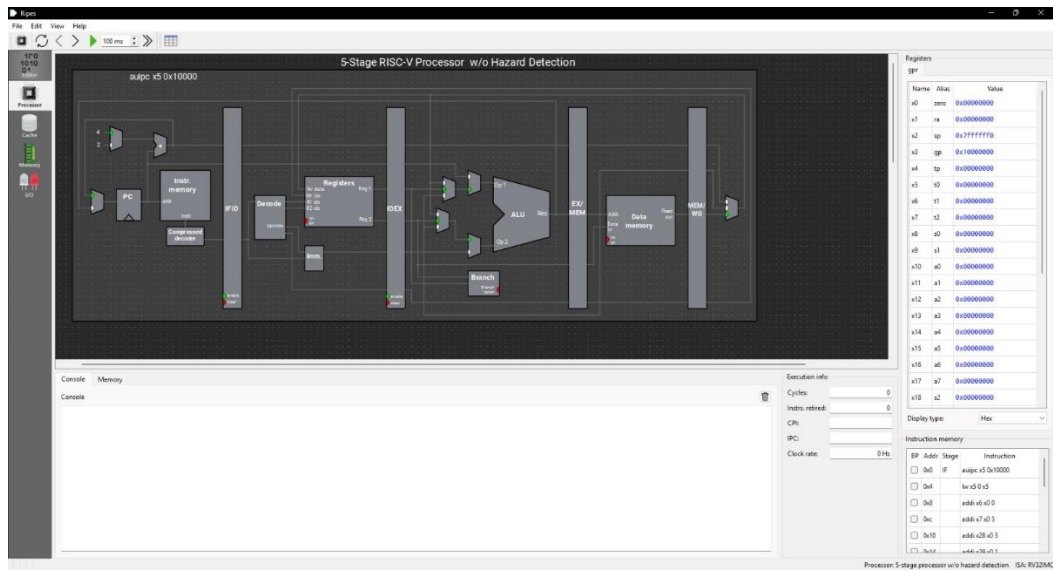


Figura 2. Captura de pantalla de la herramienta Ripes.

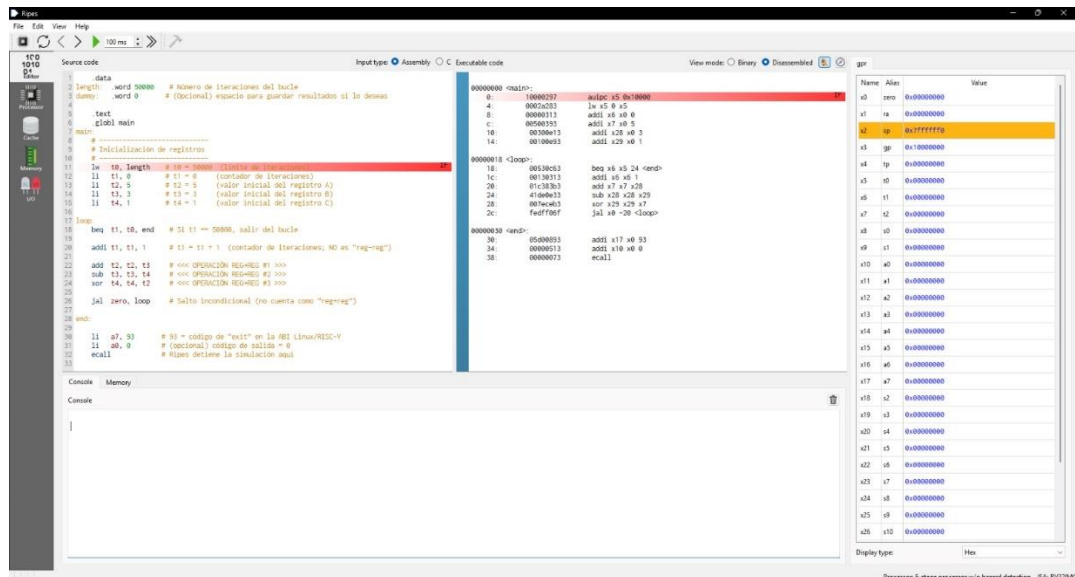


Figura 3. Captura de pantalla de la herramienta Ripes.

Testbenches en ModelSim

Cada módulo del procesador fue simulado en ModelSim utilizando testbenches específicos y modulares según haya sido el caso, como se puede observar en la figura 2 y figura 3.

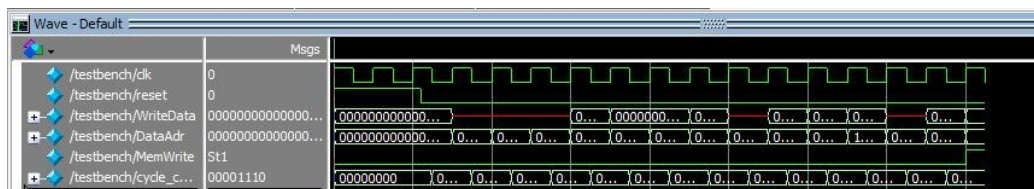


Figura 4. Muestra de los testbenches utilizados.

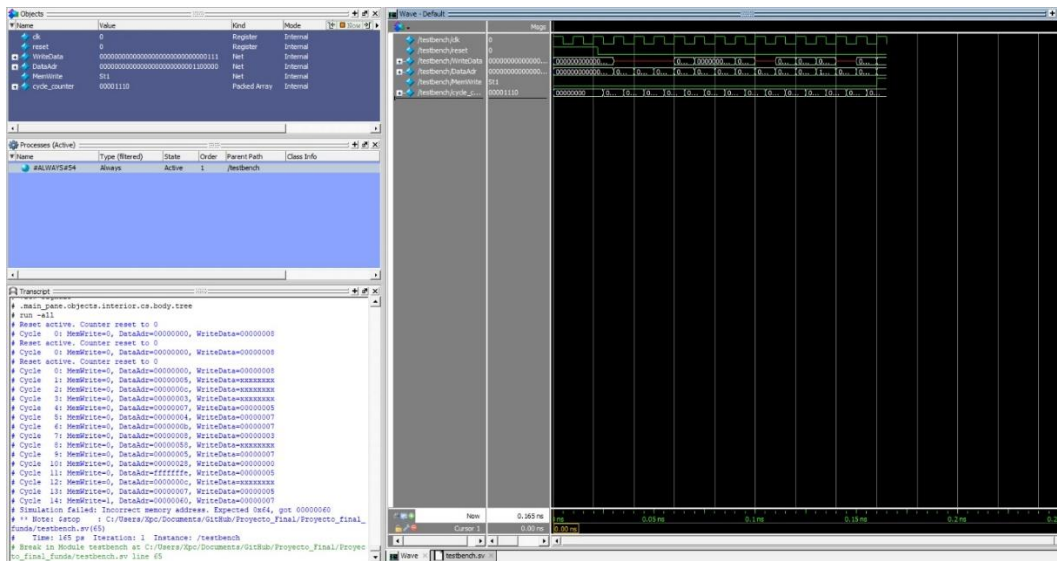


Figura 5. Captura de pantalla de la herramienta ModelSim.

Creación o adaptación de técnicas, recursos, herramientas o métodos en problemas complejos de ingeniería.

Diseño del módulo VGA

Se desarrolló un módulo personalizado en SystemVerilog para manejar la salida de datos hacia el monitor VGA. Este módulo permitió visualizar el juego, garantizando una evaluación clara y precisa de todo el sistema, y facilitando la validación de la interacción entre los módulos de procesamiento y la visualización. Sin embargo, una limitación clave al utilizar este enfoque fue la resolución limitada de la salida VGA, lo que obligó a optimizar el diseño para ajustarse a la capacidad de visualización disponible sin perder claridad en los datos procesados. A pesar de esta limitación, la validación de la interacción entre los módulos de procesamiento y la visualización se facilitó al trabajar con un entorno controlado, lo que permitió realizar ajustes finos y asegurar que las señales se representaran correctamente en la pantalla.

Simulación integrada en ModelSim

Aunque los módulos fueron validados individualmente, se realizó una simulación completa del sistema para verificar el flujo de datos desde la carga de las instrucciones hasta la salida final por VGA. Una limitación importante en el uso de ModelSim es que las simulaciones pueden ser más lentas y costosas en términos de recursos cuando se simula el sistema completo en lugar de módulos individuales. Para mitigar este problema, se usaron testbenches eficientes y modulares a la vez que se optimizó la configuración de las simulaciones, dividiendo los procesos en pasos que permitieran manejar los recursos de manera más eficaz. Esta simulación integrada no solo garantizó que todos los módulos interactuaran correctamente, sino que también permitió ajustar el rendimiento del sistema para cumplir con los requisitos sin sobrecargar la simulación.

Integración de módulos y pruebas continuas

Durante el proceso de integración, cada módulo fue evaluado en el contexto del sistema completo para asegurar que la comunicación entre ellos fuera fluida y sin fallos. Sin embargo, la integración de módulos en un sistema complejo como este presenta el desafío de las dependencias entre módulos, lo que puede resultar en errores de comunicación difíciles de detectar. Para superar esta limitación, se implementaron pruebas continuas y revisiones iterativas, lo que permitió verificar el flujo de información entre módulos de forma constante. A pesar de estas dificultades, el enfoque modular y la integración escalonada ayudaron a resolver los problemas de comunicación, garantizando que el sistema operara de manera estable.

Ajustes en la optimización de recursos

Se implementaron ajustes adicionales para optimizar el uso de recursos, como la memoria y los ciclos de reloj, asegurando que el sistema pudiera operar de manera eficiente incluso bajo condiciones de carga elevada. Sin embargo, la optimización de recursos en un sistema de procesamiento digital siempre presenta limitaciones inherentes, como el trade-off entre la eficiencia de procesamiento y el consumo de energía o la complejidad del diseño. Para superar estas limitaciones, se realizaron ajustes en la asignación de recursos, como la reducción de la latencia y el uso eficiente de las instrucciones precargadas. A pesar de las restricciones en el hardware y los recursos, se logró optimizar el rendimiento del sistema y reducir la carga sobre la FPGA, permitiendo que el diseño mantuviera un alto nivel de rendimiento en condiciones de uso constante.

Resultados

La ejecución del proyecto se observa en la figura 4.

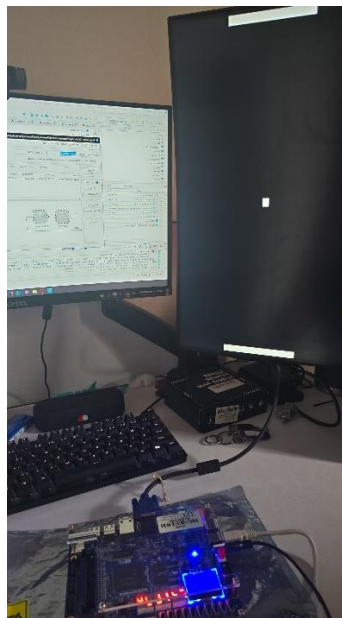


Figura 6. Imagen del juego.

Conclusiones

El desarrollo e implementación del procesador basado en arquitectura ARMv4 fue completado con éxito, cumpliendo con los principales objetivos establecidos en el proyecto. La división modular del sistema permitió una estructura ordenada y eficiente para el diseño, lo que facilitó tanto la depuración de cada módulo individual como la integración del sistema en su conjunto. El uso de herramientas avanzadas como ModelSim y Quartus Lite fue fundamental para asegurar la validación de los módulos y la implementación efectiva del diseño en la FPGA DE1-Soc.

El sistema diseñado no solo cumplió con los requerimientos técnicos del enunciado, sino que también mostró un alto potencial para ser ampliado y adaptado a diferentes aplicaciones en el futuro. La capacidad de integrar múltiples módulos y periféricos, como el controlador PS/2 y la salida VGA, refuerza las posibilidades de expansión del sistema, permitiendo su integración con nuevos dispositivos o funcionalidades.

El proyecto ha demostrado ser una base sólida para el desarrollo de sistemas embebidos interactivos, con gran potencial para futuras mejoras y ampliaciones. A pesar de los desafíos presentados, se ha alcanzado un avance significativo en la creación de un procesador ARMv4 funcional y en la implementación de un sistema que procesa y visualiza imágenes de manera eficiente.

Bibliografía

- [1] Harris, D., & Harris, S. Digital Design and Computer Architecture. Morgan Kaufmann, 2016.
- [2] ARMArchitecture Reference Manual. ARM Ltd., 2005.
- [3] «CPULator Computer System Simulator». [En línea]. Available: <https://cpulator.01xz.net/>.