

Laboratorio 2: Introducción a los lenguajes de descripción de hardware

Jonathan Guzmán Araya, Mariano Muñoz Masís
 Instituto Tecnológico de Costa Rica
 Área Académica Ingeniería en Computadores
 jonathana1196@gmail.com marianomm1301@gmail.com

Abstract—Falta

Index Terms—Falta

I. INTRODUCCIÓN

Los lenguajes de descripción de hardware (HDL) son herramientas extremadamente importantes para los diseñadores digitales modernos. Una vez que haya aprendido SystemVerilog o VHDL, podrá especificar sistemas digitales mucho más rápido que si tuviera que dibujar los esquemas completos. El ciclo de depuración también suele ser mucho más rápido, porque las modificaciones requieren cambios de código en lugar de un complicado recableado de esquemas. Sin embargo, el ciclo de depuración puede ser mucho más largo con HDL si no tiene una buena idea del hardware que implica su código. Los HDL se utilizan tanto para simulación como para síntesis. La simulación lógica es una forma poderosa de probar un sistema en una computadora antes de convertirlo en hardware. Los simuladores le permiten verificar los valores de las señales dentro de su sistema que podrían ser imposibles de medir en una pieza física de hardware [1]. Un bloque de hardware que posea entradas y salidas es llamado módulo. Una compuerta AND, un multiplexor y un circuito de prioridad son todos ejemplos de módulos de hardware. Existen dos estilos generales en los que se describen la funcionalidad de un módulo, estos son de comportamiento y estructurales.

A. Modelo de comportamiento

Los modelos de comportamiento se utilizan para describir el comportamiento del sistema en su totalidad, principalmente se tiene el *Modelo de Flujo de Datos*, que modela el procesamiento de los datos del sistema, y el *Modelo de Máquinas de Estado*, que modelan el sistema en función a eventos que recibe el sistema. Un ejemplo para un módulo de comportamiento se tienen una compuerta AND, la cual dependiendo de las entradas dará una salida, si cualquiera de las dos entradas es un 1 , la salida siempre será un 1 , pero cuando las dos entradas son un 0 la salida será un 0 lógico.

B. Modelo de Estructura

El Modelo de Estructura sirve para explicar los diferentes tipos de objetos de un sistema, visto desde el punto de vista de Software tenemos el *Diagrama de Clases*, básico para la construcción de cualquier programa, de la misma manera se

debe modelar los circuitos electrónicos, aplicando el concepto de Diseño Modular. Un ejemplo de esto es la construcción de un sumador completo de 1 bit, el cual por lo general se compone de varias compuertas XOR para la suma y de AND y OR para el acarreo.

Un proceso importante en el uso de HDL es la síntesis lógica. Las raíces de este proceso se remontan al tratamiento de la lógica por George Boole, en lo que ahora se denomina álgebra booleana. En los primeros días, el diseño lógico implicaba manipular las representaciones de tablas de verdad como mapas de Karnaugh. Sin embargo, la evolución de componentes lógicos discretos a matrices lógicas programables aceleró la necesidad de la automatización de la síntesis lógica. La síntesis lógica es un proceso mediante el cual una especificación abstracta del comportamiento deseado de un circuito (código HDL), normalmente a nivel de transferencia de registro (RTL), se convierte en una implementación de diseño en términos de compuertas lógicas, normalmente mediante un programa informático llamado herramienta de síntesis. Por lo que se puede decir que la síntesis lógica es un aspecto de la automatización del diseño electrónico.

Las FPGAs son el acrónimo para Field Programmable Gate Array y es una serie de dispositivos basados en semiconductores a base de matrices de Bloques Lógicos Configurables, su principal característica es que pueden ser reprogramados para un trabajo en específico [2]. Los principales componentes que posee la FPGA son terminales, Buffers, Flips – Flops, Tablas de búsqueda, Bloques de memoria, Bloques dedicados de Multiplicación, Transceptores para transmisión serie de muy alta velocidad, procesador en hardware embebido, etc [3].

Falta la 4ta pregunta

El mundo actual evoluciona alrededor de la tecnología y las ventajas que esta conlleva, por lo que se depende en gran medida del crecimiento y mejora de esta para producir comodidad y satisfacción a nuestras vidas, ya sea desde el punto de vista personal o industrial (empresarial). Los dispositivos FPGA juegan un papel importante en esta faceta, por lo que algunas de sus aplicaciones más comunes son las que se encuentran en los siguientes campos de interés:

- Procesamiento de videos e imágenes

- Telecomunicaciones y comunicación de datos
- Servidor y nube
- Defensa y espacio
- Científico y médico

Se utiliza con frecuencia y de forma extensiva en los sistemas de comunicación para mejorar la capacidad de la red, la cobertura y la calidad del servicio, al mismo tiempo que se reduce la latencia y los retrasos, especialmente cuando se trata de manipulación de datos [4].

Intel y Xilinx con ayuda de las FPGA han ido abriendo nichos en aplicaciones de centros de datos donde el bajo consumo de energía, el alto rendimiento y la capacidad de configuración pueden superar los desafíos de programación [5].

La industria automotriz utiliza sistemas electrónicos cada vez más complejos para ofrecer mayor seguridad y eficiencia al conductor. Sin embargo es difícil para las unidades de control (ECU) basadas en CPU y GPU mantenerse al día con la electrónica de consumo, debido a los largos ciclos de desarrollo de chips y los rigurosos estándares de confiabilidad y calidad aplicados a la industria automotriz. Es por ello que los arreglos de puertas programables (FPGA) pueden desempeñar un papel importante para llenar este vacío al proporcionar rendimiento y flexibilidad de vanguardia a los arquitectos de sistemas para personalizar sus proyectos con una estructura de circuito electrónico flexible (programable) [6].

II. SISTEMA DESARROLLADO Y RESULTADOS

III. ANÁLISIS DE RESULTADOS

IV. CONCLUSIONES

V. BIBLIOGRAFÍA

REFERENCES

- [1] D. Harris and S. Harris, *Digital design and computer architecture*. Morgan Kaufmann, 2010.
- [2] J. López. (2020, Sep.) Qué es un FPGA: características y utilidad de este tipo de componente. Hardzone. [Online]. Available: <https://hardzone.es/reportajes/que-es/fpga-caracteristicas-utilidad/>
- [3] M. C. Sisterna. FIELD PROGRAMMABLE GATE ARRAYS(FPGAS). Departamento de Electrónica y Automática. [Online]. Available: http://dea.unsj.edu.ar/sisdig2/FieldProgrammableGateArrays_A.pdf
- [4] HardwareBee. FPGA Advantages and Most Common Applications. HardwareBee. [Online]. Available: <https://hardwarebee.com/fpga-advantages-common-applications-today/>
- [5] K. Freund. (2018, Jul.) An Update On Data Center FPGAs. Forbes. [Online]. Available: <https://www.forbes.com/sites/moorinsights/2018/07/20/an-update-on-data-center-fpgas/?sh=4d0f53af46c2>
- [6] M. D. P. Emilio. (2017, Aug.) Maurizio Di Paolo Emilio. Electronic Design. [Online]. Available: <https://www.electronicdesign.com/markets/automotive/article/21805452/use-fpgas-to-streamline-automotive-electronics>