

Aplicaciones Básicas de VHDL en Zybo z7

JOAQUÍN LA ROSA¹, MARÍA JOSÉ VÁSQUEZ¹

¹Pontificia Universidad Católica de Chile (e-mail: jalarosa@uc.cl, mjvasquez1@uc.cl)

SI Autorizo que mi proyecto (tal como ha sido entregado, sin nota ni comentarios de evaluación) sea publicado en un repositorio para pueda servir de guía y ser mejorado en proyectos de futuros estudiantes.

Este proyecto ha sido desarrollado bajo el curso IEE2463: Sistemas Electrónicos Programables.

ABSTRACT El proyecto realizado corresponde a la aplicación de procedimientos básicos de VHDL utilizando la tarjeta Zybo z7-10. Para esto, se utilizaron cuatro de los botones disponibles en la tarjeta, y los cuatro switches para definir la velocidad de parpadeo de los cuatro LEDS asociados. El objetivo principal consistió en implementar ciertos tipos de funciones en Vivado, para luego conectarlos con la tarjeta Zybo y utilizar sus componentes. Los resultados permitieron visualizar el funcionamiento correcto de los códigos mediante el encendido y apagado de los leds, que mostraban con precisión los patrones de parpadeo predeterminados, según las entradas ingresadas en los switches, que definen cuáles serán los leds a encender, y las entradas de los botones, que definen la velocidad de parpadeo de las luces.

INDEX TERMS VHDL, máquina de estados, LED, switch, botón, Zybo z7-10, clock.

I. ARQUITECTURA DE HARDWARE

El proyecto realizado sigue la estructura presentada en la Figura 1.

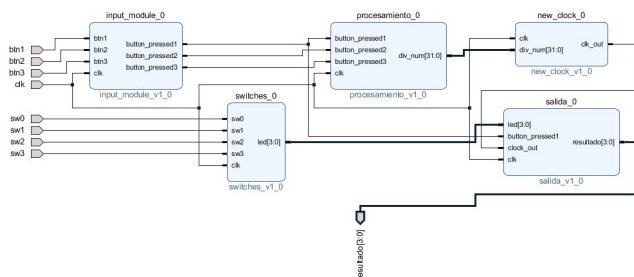


FIGURE 1: Diagrama de bloques del sistema

En primer lugar, consta de un bloque llamado *input_module* que registra la entrada de los botones cada vez que son presionados en la tarjeta. Posteriormente, la salida de este bloque se utiliza como entrada para el módulo *Procesamiento*, que, según la combinación de botones presionados por el usuario, entrega un divisor de clock llamado *div_num*. A partir de dicho valor, el siguiente bloque, llamado *new_clock* genera una salida de clock alterna que indicará la velocidad de parpadeo de los leds. Para escoger los leds a encender, el bloque *Salida* utiliza como entrada el vector

entregado por el bloque *Switches* Según los requerimientos y condicionales presentados en el código, este último bloque entrega como salida un vector de cuatro bits llamado *resultado*, que le indica a la tarjeta qué leds deben ser encendidos y con qué velocidad deben parpadear.

II. ACTIVIDADES REALIZADAS

Con respecto a las actividades obligatorias, se presenta el nivel de desempeño obtenido a continuación:

AO1 (100%): El bloque "*Salida*" cuenta con tres componentes que forman parte de su entity, representado los bloques pertenecientes al diagrama de bloques de la Figura 1. Completamente logrado, el código implementado ha compilado, se logra simular luego de la implementación, se logra generar bitstream y cargarlo en la ZYBOZ7.

AO2 (100%): Cada uno de los bloques presentados en la figura 1 se llevó a cabo en un archivo diferente de Vivado, que luego fue empaquetado (convertido en Package). Utilizando estos Packages, se creó un archivo final que reuniera todos los bloques necesarios a partir del repositorio de IP's. Esto permitió corroborar que los archivos funcionaban correctamente, que habían sido cargados sin problemas y que el sistema completo entregaba una respuesta satisfactoria.

AO3 (0%): No fue posible implementar los AXIS respectivos.

AC1 (100%): Para el bloque *Input_module* se utilizó una máquina de estados que permitiera actualizar el estado actual

de cada uno de los botones a partir de la entrada de la Zybo y un estado anterior. Las salidas de esta máquina repercuten en el código completo, puesto que el estado de los botones define la velocidad de parpadeo de los leds.

AC2 (100%): Como se explicó anteriormente, los switches fueron usados para determinar qué leds deben prenderse y cuáles deben mantenerse apagados. Los botones y sus posibles combinaciones entregan información respecto al patrón de parpadeo lumínico, y las luces led permiten observar el funcionamiento del sistema.

AC3 (65%): A lo largo del código se utilizaron distintos operadores, tanto lógicos como aritméticos, para definir las relaciones entre entradas y salidas de cada bloque. Entre los operadores utilizados se encuentran "and", "or", "<=", ":", "*", "+", etc. Respecto a los atributos, se utilizó el atributo "x'EVENT" en más de una oportunidad para medir la entrada clk.

AC4 (0%): no se utilizaron variables.

AC5 (100%): Se utilizó código secuencial al generar la máquina de estados, ya que el sistema requería de realimentación a partir de los estados anteriores de las algunas de las señales. Por otra parte, se utilizó código concurrente para definir las salidas de AXI mediante el comando "when".

AC6 (0%): no se utilizaron *functions* ni *procedures*.

AC7 (0%): no se utilizaron funciones no vistas en clases.

III. RESULTADOS DE SIMULACIÓN

La figura 2 muestra los resultados de la implementación del bloque "salida" que recibe como entrada los leds que deben ser encendidos y la velocidad de parpadeo de los mismos, y entrega como salida las luces parpadeantes.

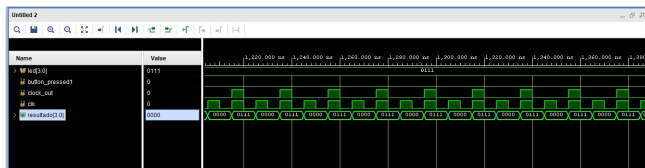


FIGURE 2: Diagrama del bloque de salida

La idea de este bloque es que para cada subida del clk_out los leds que recibe como entrada pasan de cero a uno, lo que significa que se encienden durante un ciclo de clock, para luego apagarse nuevamente. Esta relación se mantiene así mientras ninguna de las entradas cambie, lo que permite apreciar que el bloque final sí cumple con el objetivo buscado.

IV. RESULTADOS IMPLEMENTACIÓN

La implementación final no logró presentar los resultados esperados por completo, puesto que no fue posible ejecutar la simulación e implementación de los axis, lo que provocó que el código final no fuese visible en la tarjeta.

V. CONCLUSIONES

Los objetivos planteados inicialmente no fueron cumplidos del todo, puesto que si bien las simulaciones de cada uno

de los bloques presentaban los resultados esperados, al momento de combinarlos, la simulación no era capaz de responder a los ciclos de clock pedidos, dado que la comunicación con el sistema AXI fue inviable-

VI. TRABAJOS FUTUROS

El proyecto presentado sólo permite utilizar los cuatro leds principales de la tarjeta Zybo z7, a pesar de que esta cuenta con otros dos leds programables, de los cuales uno corresponde a un led RGB. Para proyectos futuros, se propone como desafío incorporar el funcionamiento de los dos leds sobrantes, ajustados a un duty cycle del clock definido por los botones de la tarjeta, tal como se hizo aquí con los primeros cuatro leds. Además, se propone utilizar las propiedades del led RGB para alternar sus colores a medida que avanza el ciclo de pestañeo, con el fin de que la tarjeta se vea lo más llamativo posible.

REFERENCES

- [1] Ayudantía 2, Sistemas Electrónicos Programables. Recuperado de https://www.myqnapcloud.com/share/74784g6inp2m2694r246673b_07981f9lk780268qr048xv559cd74f44/home/23_1S_Ayudantias
- [2] Chat GPT, comunicación personal, 29 de abril 2023.
- [3] Laboratorios varios, Sistemas Electrónicos Programables.

...