

Laboratorio #2: Introducción a los lenguajes de descripción de hardware

Jorge Agüero Zamora, Arturo Chinchilla Sánchez, Luis Murillo Rios
georgeaz56@gmail.com mchinchilla11@gmail.com luismuelino@gmail.com

Área Académica de Ingeniería en Computadores
Instituto Tecnológico de Costa Rica

Resumen—En este laboratorio se pretende realizar los primeros pasos en el diseño y creación de circuitos digitales, esto con la ayuda de Lenguajes de Descripción de Hardware para el diseño y la FPGA como dispositivo para la creación del circuito. También se quiere hacer énfasis en los dos métodos de descripción, el estructural y por comportamiento, además del funcionamiento interno de las FPGAs.

Palabras clave—FPGA, testbench, HDL, VHDL, SystemVerilog

I. INTRODUCCIÓN

En el presente laboratorio se trabaja con Circuitos Digitales, utilizando Lenguaje de descripción de Hardware para el diseño de los circuitos y FPGAs como método de construcción de dichos circuitos.

El Lenguaje de Descripción de Hardware o HDL por sus siglas en inglés, es una herramienta que permite simular o abstraer componentes de Hardware utilizando un modelado ya sea estructural o de comportamiento. En el modelado estructural se pretende describir la estructura física del componente a modelar, como este está construido, cada una de sus entradas, salidas, etc. Mientras que en modelado de comportamiento, se describe cómo se comporta el dispositivo ante diferentes señales aplicadas a sus entradas.

En el diseño de circuitos digitales, se aplica un proceso de síntesis lógica que transforma la descripción textual de un módulo a un modelo construido mediante compuertas lógicas, es decir, transforma el código HDL en un *netlist* que describe el hardware (las compuertas lógicas a utilizar y su conexiones).

Los arreglos de compuertas programables o FPGA por sus siglas en inglés, son chips de silicio donde cada una de sus muchas compuertas pueden ser reprogramadas para ser implementadas en diversas aplicaciones. Internamente las FPGAs están compuestas por cables, compuertas lógicas biestables y puertos de entrada y salida, donde en un principio todo estos componentes no están conectados entre sí, como si fuera una bolsa con componentes listos para ensamblar un circuito digital. Estos componentes se conectarán hasta que el proceso de síntesis lógica transforme la descripción textual vista anteriormente en un *bitstream* con el modelo a construirse dentro de la FPGA, donde se unirán los componentes necesarios para construir dicho circuito digital.

II. RESULTADOS

II-A. Problema 1

En el cuadro I se muestra los resultados obtenidos para el decodificador de un display de 7 segmentos, donde se tiene una entrada de 4 bits y para cada entrada se obtiene una salida que representará los números del 0 al 9 y las letras de la A a la F utilizando para ello cada segmento del display.

Cuadro I
TABLA DE VERDAD DISPLAY 7 SEGMENTOS

Tabla de verdad							
$D_{3:0}$	S_a	S_b	S_c	S_d	S_e	S_f	S_g
0000	1	1	1	1	1	1	0
0001	1	0	0	1	1	1	1
0010	0	0	1	0	0	1	0
0011	0	0	0	0	1	1	0
0100	1	0	0	1	1	0	0
0101	0	1	0	0	1	0	0
0110	0	1	0	0	0	0	0
0111	0	0	0	1	1	1	1
1000	0	0	0	0	0	0	0
1001	0	0	0	1	1	0	0
1010	0	0	0	1	0	0	0
1011	1	1	0	0	0	0	0
1100	0	1	1	0	0	0	1
1101	1	0	0	0	0	1	0
1110	0	1	1	0	0	0	0
1111	0	1	1	1	0	0	0

En la figura II-A se muestra como sería el circuito digital construido utilizando compuertas lógicas para el display de 7 segmentos.

II-B. Problema 2

Para crear un sumador de 4 bits solamente se deben conectar la entrada C_{in} con la salida C_{out} del sumador anterior (de un 1 bit) y así sucesivamente hasta tener 4 sumadores de 1 bit conectados, los cuales formarán un sumador de 4 bits y los elementos en A y B se conectan a los bits en orden de menos significativo a mas significativo de los números que se desean sumar.

En el cuadro II se muestra la tabla de verdad obtenida de un sumador de 1 bit.

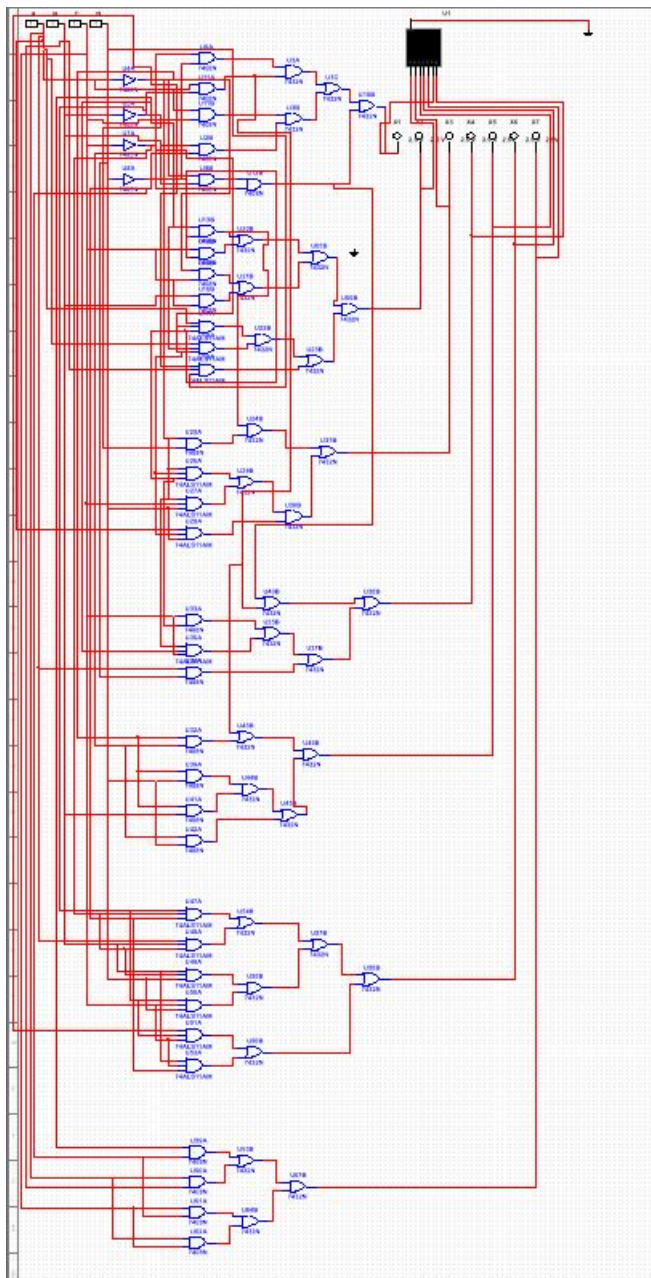


Figura 1. Circuito compuesto por compuertas lógicas para decodificador para el display 7 segmentos

Cuadro II
TABLA DE VERDAD SUMADOR 1 BIT

Tabla de Verdad				
Entradas			Salidas	
A	B	C _{in}	Salida	C _{out}
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Además, en la figura II-B se muestra el diagrama para un

sumador de 1 bit hecho con compuertas lógicas.

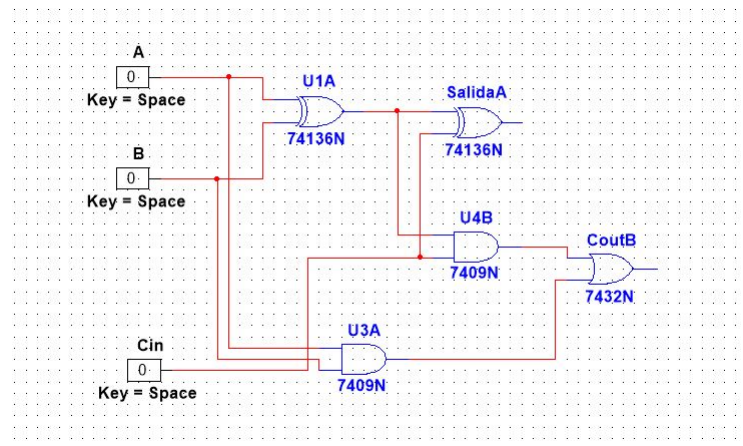


Figura 2. Sumador de 1 bit

Se probó el sumador de un bit utilizando multisim, esto con el objetivo de verificar los resultados.

II-C. Problema 3

Un contador es un dispositivo que va aumentando (o disminuyendo) su valor de salida conforme pasa una señal de reloj. La cantidad de valores que puede contar depende de la cantidad de salidas que este tenga. Los contadores permiten reiniciar el conteo mediante una señal de entrada conocida como "Reset", y estos son síncronos o asíncronos dependiendo de si deben esperar la señal de reloj para reiniciar el conteo (síncronos), o si lo reinician al instante de recibir la señal de Reset" (asíncronos).

Para el diseño del contador de N bits con Reset" síncrono se utilizaron Flip Flops JK y algunas compuertas AND para lograr el comportamiento deseado de contador. La patilla de Reset" debe mantenerse en ALTO (Logic 1) y cuando se desea reiniciar el conteo, esta se pone en BAJO (Logic 0).

En la figura 3 se muestra el diagrama de un contador de 4 bits, pero este se puede extender a N bits con mas flip flops. Por necesidad de dimensionar las salidas del contador y por simplicidad se eligió representar uno con 4 bits de salida, lo cual permite un conteo de 0 hasta 15.

III. ANÁLISIS DE RESULTADOS

III-A. Problema 1

Para el circuito digital del decodificador para la display 7 segmentos se utilizó un modelado de comportamiento, donde para una entrada de 4 bits se describió como debía comportarse el dispositivo en su salida.

Al ejecutar el testbench el circuito funciona de manera correcta y de la misma forma al pasarlo a la FPGA, de manera que se pudieron representar los números del 0 al 9 y las letras de la A a la F satisfactoriamente, encendiendo los segmentos como se mostró en el cuadro I.

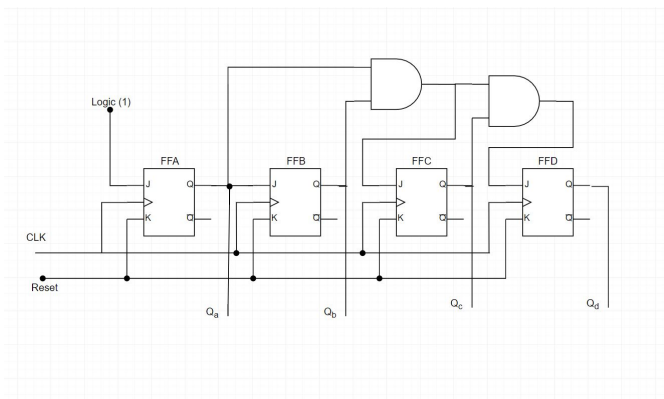


Figura 3. Diagrama contador de 4 bits sincrónico

III-B. Problema 2

A la hora de utilizar el lenguaje de descripción de hardware system verilog, para montar el sumador en la FPGA y analizar los resultados, se tuvo que tomar en cuenta que al ser un sumador de 4 bits formado de sumadores de 1 bit el acarreo de cada sumador tuvo que ir conectado correctamente a los acarreos de los otros componentes (sumador de 1 bit).

Con esto en cuenta se tiene que saber que el acarreo de entrada del primer sumador debe ser de "0" lógico, ya que si se toma como uno esto va a interferir con el resultado de la suma, esto porque se le estaría sumando un bit mas el cual no pertenece a la suma inicial, esto puede ser útil a la hora de realizar una resta y utilizar el complemento a base 2 para este propósito.

Otro dato a tomar en cuenta es el que el acarreo de salida del último sumador de 1 bit debe ser sumado al resultado final ya que este es un bit, sino este bit de la suma no se toma en cuenta para el valor final, lo que puede generar un resultado erróneo.

A la hora de montar el sumador en la FPGA se utilizaron 8 switches para los valores respectivos de "a" y "b", además de uno extra para el acarreo de entrada, esto se unió al 7 segmentos del problema 1 para mostrar los resultados.

III-C. Problema 3

SystemVerilog permite definir entradas y salidas parametrizables, lo cual permite definir el comportamiento de un módulo en general que se puede utilizar para cualquier cantidad deseada de entradas y/o salidas. Se aprovechó de esta característica del lenguaje para definir el contador.

Usando la función de *always_ff* se modela el comportamiento de un flip flop el cuál recibe de entrada únicamente la señal de reloj(CLK), por lo tanto solo se hace el incremento o el reinicio del conteo conforme entra una señal de reloj, obteniendo así el comportamiento sincrónico del "Reset" deseado.

Los testbenches de auto-chequeo utilizan la función *assert* para probar que la salida de un módulo ante una entrada sea la esperada, y de lo contrario muestra un error, pudiendo así indicar fácilmente donde sucede el error, ayudando así a poder arreglar errores más fácilmente.

IV. CONCLUSIONES

Se logró satisfactoriamente la representación de los números del 0 al 9 y las letras de la A a la F en el display de 7 segmentos utilizando el decodificador diseñado, la comprobación del correcto funcionamiento se llevó a cabo con la ayuda del testbench y al pasar el diseño a la FPGA.

La conexión entre el modulo del 7 segmentos para mostrarlos en pantalla los resultados de la suma mostró un problema. El sumador se encuentra programado en system verilog y el 7 segmentos en VHDL por lo que dio un conflicto a la hora de leer la configuración de bits enviados por el sumador para su representación el cual no se pudo resolver en el laboratorio. Por lo que se concluye que pese a que ambos lenguajes son compatibles pero para unirlos hay que tener consideraciones en cuenta.

El uso de un HDL para la descripción de un módulo de hardware provee un nivel de abstracción elevado en el diseño de estos, lo cual es de gran comodidad, puesto que permiten que el diseñador se enfoque más en el funcionamiento del dispositivo que en los componentes que terminaran formando su diseño.

Durante el uso y la manipulación de la FPGA se debe tener especial cuidado con los pines para evitar dañar el equipo.

REFERENCIAS

- [1] S. Harris and D. Harris, *Digital design and computer architecture*. Amsterdam [i 11 pozostałych]: Elsevier / Morgan Kaufmann Publishers, 2016.