

PROYECTO UNO Y DOS CIRCUITOS DIGITALES II



Presentado por:

**Lily Yasmin Salamanca Chito
(100621011425)**

Presentado a:

Ing. Carlos Hernan Tobar Arteaga

UNIVERSIDAD DEL CAUCA

FACULTAD DE INGENIERÍA ELECTRÓNICA Y TELECOMUNICACIONES

DEPARTAMENTO DE TELECOMUNICACIONES

Popayán – 2023

RESUMEN DEL TRABAJO REALIZADO

Para la materia de Digitales 2, se llevaron a cabo dos proyectos relacionados con el diseño e implementación de compuertas lógicas, utilizando el lenguaje VHDL en el entorno de desarrollo Quartus. Estos proyectos incluyeron la creación de compuertas lógicas básicas, así como circuitos más complejos como HalfAdder (medio sumador) , FullAdder (sumas y restas), Inc16, Add16 y ALU. Además, se realizaron pruebas de cada componente y se verificó su funcionamiento tanto en ModelSim - Altera, como a través del diagrama de tiempos en Quartus. Todo el código desarrollado se gestionó y compartió a través de GitHub, organizado en una carpeta llamada "Proyectos-VHDL".

Desarrollo de Proyecto 1 y 2:

- En el primer proyecto, se diseñaron e implementaron diversas compuertas lógicas básicas utilizando VHDL, tales como:
 - Not Gate
 - And Gate
 - Or Gate
 - Xor Gate
 - Multiplexor (Mux)
 - Demultiplexor (DMux)
 - And16
 - Or16
 - Not16
 - Mux16
 - Or8Way
 - Mux4Way16
 - Mux8Way16
 - Demux4Way
 - Demux8Way

Cada componente se diseñó y probó individualmente en Quartus. Se crearon también archivos de prueba (testbench) para cada componente para verificar su correcto funcionamiento.

Se realizó un proceso exhaustivo de verificación para cada componente. Se ejecutaron pruebas utilizando los archivos de prueba en ModelSim para garantizar que cada componente funcionará según lo esperado. Además, se utilizó el diagrama de tiempos (waveform) en Quartus para verificar que las señales de entrada y salida de cada componente se comportaran correctamente.

Para una mejor comprensión de lo dicho anteriormente, a continuación se mostrará un ejemplo del diseño de una compuerta lógica NotGate, con todos los procesos descritos:

Compuerta Not Gate: La compuerta NOT (o inversora) tiene una sola entrada y produce una salida que es la inversa lógica de la entrada. Es decir, si la entrada es "1", la salida será "0", y viceversa. La siguiente figura muestra la tabla de verdad de esta compuerta.

Q	Q'
0	1
1	0

Figura 1 - Tabla de verdad compuerta Not.

Para la elaboración de la compuerta Not Gate en lenguaje VHDL, se realizó el siguiente código en Quartus, teniendo en cuenta las librerías y estructura del programa. Tenemos la parte de la Interfaz (Entity), aquí se declaran las entradas y salidas de la compuerta y en Implementación (Architecture) se construye la Not, con ayuda de la compuerta NAND (Primitiva).

```
-- Library and packages
library IEEE;
use IEEE.std_logic_1164.all;

-- Entity (Interface)
entity NotGate is
    port(
        x : in  std_logic;
        f : out std_logic
    );
end entity;

-- Architecture (Implementation)
architecture NotGate_arch of NotGate is
begin
    F <= x nand x;
end architecture;
```

Figura 2 - Código NotGate

De este proceso, se crea el diagrama esquemático de la compuerta NotGate, el cual tiene una entrada X y una salida F. Este proceso se muestra a continuación en la Figura 3:

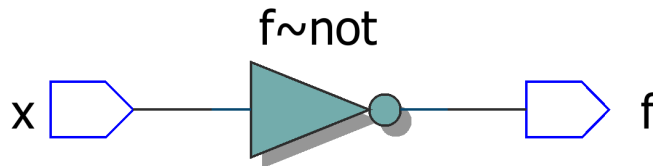


Figura 3 - Diagrama esquemático de la NotGate

Para probar si la compuerta está funcionando correctamente, se crea el diagrama de tiempo de Waterform, el cual se muestra a continuación:

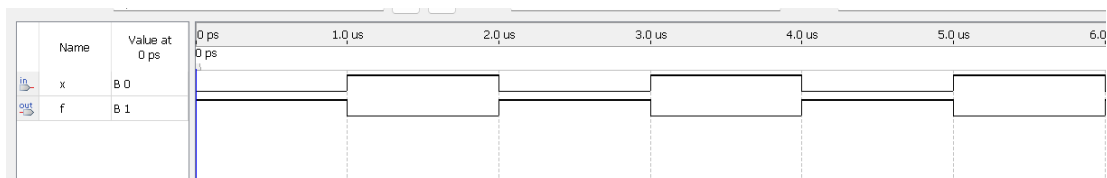


Figura 4 .-- Diagrama de tiempo de Waterform

Al comprobar que efectivamente la compuerta está realizando el proceso esperado, se procede a probar el testbench que hemos creado para la NotGate en el ModelSim-Altera.

Objects			
Name	Value	Kind	Mode
x_test	1	Signal	Internal
f_test	0	Signal	Internal

Figura 5 - Prueba del testbench en ModelSim

Como se puede observar en la figura 5, los resultados al igual que en el diagrama de tiempo de waterform son correctos y podemos dar por terminado el diseño de la compuerta NotGate.

- En el segundo proyecto, se avanzó hacia circuitos digitales más complejos, incluyendo:
 - Half Adder
 - Full Adder
 - Inc16
 - Add16
 - ALU (Arithmetic Logic Unit)

Cada uno de estos circuitos se implementó en VHDL y se sometió a un riguroso proceso de pruebas y verificación similar al proyecto 1.

Para los dos proyectos, cada componente y su correspondiente archivo de prueba se subieron a GitHub en distintas carpetas: "Proyecto1" y "Proyecto2". Se garantizó que el código estuviera bien documentado y organizado para facilitar la revisión.

Conclusiones

Se logró un progreso significativo en la comprensión y aplicación de VHDL para la implementación de compuertas lógicas y circuitos digitales. Se adquirieron habilidades valiosas en el diseño, prueba y verificación de componentes digitales, lo que sienta las bases para proyectos más complejos en el futuro.