

# Práctica 3: Registros

Rafael Isaac Puentes Garcia  
Arnold Torres Maldonado

*Instituto Politécnico Nacional, Unidad Profesional Interdisciplinaria de Ingeniería  
Campus Zacatecas, Fundamentos de Diseño Digital*

**Resumen**—Los circuitos pueden describirse en VHDL con instrucciones secuenciales, por lo que el tema de Registros es un tema que nos lleva para tener en cuenta la forma de implementarlos en VHDL. La programación y el diseño de sistemas digitales en el lenguaje VHDL, en función a la descripción de la lógica combinacional por lo que se realizó simulaciones para comprobar el funcionamiento de los registros, como son su manejo en VHDL.

## I. INTRODUCCIÓN

ESTE documento proporciona la implementación de las compuertas lógicas, en el lenguaje de descripción para circuitos electrónicos. Dicho lenguaje es de descripción de circuitos electrónicos digitales que utiliza distintos niveles de abstracción. El significado de las siglas VHDL es *VHSIC (Very High Speed Intergrated Circuits) Hardware Description Language*. Esto significa que VHDL permite acelerar el proceso de diseño. VHDL no es un lenguaje de programación, por ello conocer su sintaxis no implica necesariamente saber diseñar con él. Como objetivo se tiene el conocer la forma de implementar Registros en VHDL, en esta practica aprenderemos a diseñar registros en VHDL. Existen diferentes dependiendo de su entrada sea serie o paralela y su salida en serie o paralela. También los registros se pueden desplazar de izquierda a derecha o viceversa.

## II. DESARROLLO DEL ARTÍCULO

VHDL es una descripción de hardware, que permite describir circuitos síncronos y asíncronos. En particular VHDL permite tanto una descripción de la estructura del circuito (descripción a partir de subcircuitos más sencillos), como la especificación de la funcionalidad del circuito utilizando formas familiares a los lenguajes de programación. Es un lenguaje con una sintaxis amplia y flexible que permite el modelado estructural, en flujo de datos y de comportamiento de hardware. Permite el modelado preciso, en distintos estilos, del comportamiento de un sistema digital conocido y el desarrollo de modelos de simulación. Los objetivos del lenguaje VHDL son el modelado (desarrollo de un modelado para la simulación de un circuito o sistema) y la síntesis (proceso en donde se parte de una especificación de entrada o un determinado nivel de abstracción y se llega a una realización más detallada, menos abstracta) de circuitos y sistemas electrónicos y digitales.

### A. Registros

Los registros de desplazamiento son un tipo de circuitos secuenciales. Los registros se utilizan principalmente para almacenar datos digitales y, normalmente, no poseen una

secuencia característica interna de estados. Los registros de desplazamiento están formados por un conjunto de flip-flops, y son muy importantes en las aplicaciones que precisan almacenar y transferir datos dentro de un sistema digital. En general, un registro se utiliza únicamente para almacenar y desplazar datos (1s y 0s), que introduce en él una fuente externa y, normalmente, no posee ninguna secuencia característica interna de estados.

Un registro es un circuito digital con dos funciones básicas: almacenamiento de datos y movimiento de datos. La capacidad de almacenamiento de un registro le convierte en un tipo importante de dispositivo de memoria. Tenemos dos conceptos importantes: capacidad de almacenamiento y capacidad de desplazamiento. La capacidad de almacenamiento de un registro es el número total de bits (1s y 0s) de un dato digital que puede contener. Mientras que la capacidad de desplazamiento de un registro permite el movimiento de los datos de una etapa a otra dentro del registro. La Fig. 1 no muestra los tipos de registros.

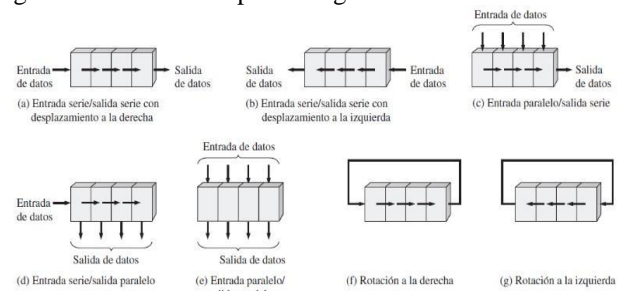


Fig. 1. Tipos de Registros.

### 1) Desarrollo de la practica

Se realizara un registro universal con las siguientes características: será un registro de 4 bits, desplazamiento de izquierda con la entrada de datos, desplazamiento a la derecha con la entrada de datos, con carga en paralelo, un botón de RESET y un botón SET síncrono.

La Fig. 2 nos muestra el registro a implementar en VHDL.

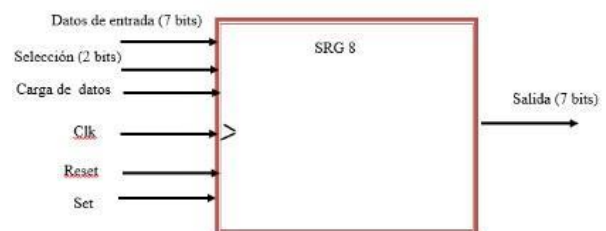


Fig. 2. Registro a implementar en VHDL.

Para implementar el registro se considerará las opciones mostradas en la Tabla 1 donde nos muestra las funciones del registro.

TABLA I  
FUNCIONES DEL REGISTRO.

Selección 1	Selección 2	Simulación
0	0	No operación
0	1	Desplazamiento a la izquierda
1	0	Desplazamiento a la derecha
1	1	Carga paralela

#### a) Desarrollo de la Practica 1)

- 1) Crear nuevo proyecto.
- 2) Crear un archivo VHDL.
- 3) Editar Código.
- 4) Simule el funcionamiento del registro.

#### b) Resultados

- 1) Registro
- 2) Código VHDL

```

---- library IEEE; use
IEEE.STD_LOGIC_1164.ALL;

entity Registro is
  Port ( Entrada_datos : in STD_LOGIC_VECTOR (7 downto
0);
        Seleccion : in STD_LOGIC_VECTOR (1 downto 0);
        Carga : in STD_LOGIC;
        Clk : in STD_LOGIC;
        Reset : in STD_LOGIC;
        Set : in STD_LOGIC;
        Salida : out STD_LOGIC_VECTOR (7 downto
0)); end Registro; architecture Behavioral of Registro is

  signal clk1 : STD_LOGIC; signal cuenta :
integer range 0 to 50000000;
  signal selec1 : integer; signal registro_aux :
STD_LOGIC_VECTOR(7 downto 0);

begin
  Salida <= registro_aux; reloj: process (Clk)--se
declara un reloj de 1 segundo begin
    if (Clk'event and Clk='1') then
      if (Reset = '1') then
        registro_aux <= "00000000";
      else
        if (Seleccion = "01") then
          registro_aux <= Entrada_datos(6 downto 0) &
Carga;
        elsif (Seleccion = "10") then
          registro_aux <= Carga & Entrada_datos(7
downto 1);
        elsif (Seleccion = "00") then
          registro_aux <= Carga & Entrada_datos(7 downto
1); end if; end if;
        if (cuenta=50000000) then --se verifica cada vez que ocurre

```

un cambio en el reloj cuenta

```

    <= 0;
    clk1 <= '1';
  else
    cuenta <= cuenta + 1;--el proceso de contar aumenta
    la salida en 1 clk1 <= '0';
  end if;
end if;
end process;

selec: process (Seleccion)
begin
  case Seleccion is
    when "00" => selec1 <= 0;
    when "01" => selec1 <= 1;
    when "10" => selec1 <= 2;
    when "11" => selec1 <= 3;
    when others => selec1 <= 0;
  end case; end
process;
end Behavioral;

```

#### 3) Simulaciones

La Fig. 3 nos muestra la simulación con la combinación de entrada **01011100** siendo así la entrada de 8 bits con las combinaciones de selección **11**, la **carga** de **0**, **clk** de **1**, **reset** de **1**.

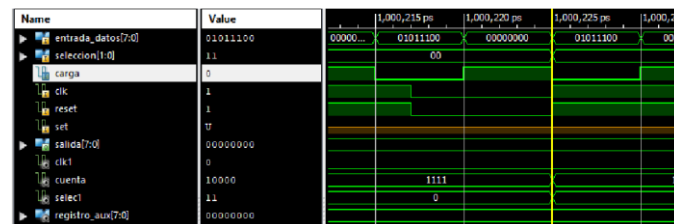


Fig. 3. Simulación 1 con entrada 01011100 y selección de 11.

La Fig. 4 nos muestra la simulación con la combinación de entrada **01011100** siendo así la entrada de 8 bits con las combinaciones de selección **00**, la **carga** de **0**, **clk** de **1**, **reset** de **1**.

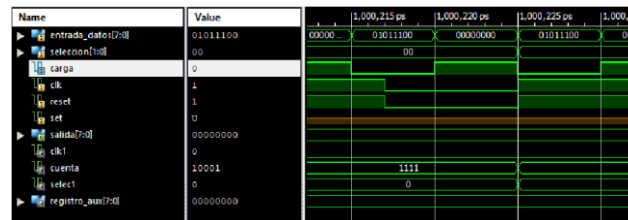


Fig. 4. Simulación 1 con entrada 01011100 y selección de 00.

La Fig. 5 nos muestra la simulación con la combinación de entrada **00000000** siendo así la entrada de 8 bits con las combinaciones de selección **11**, la **carga** de **1**, **clk** de **1**, **reset** de **1**.

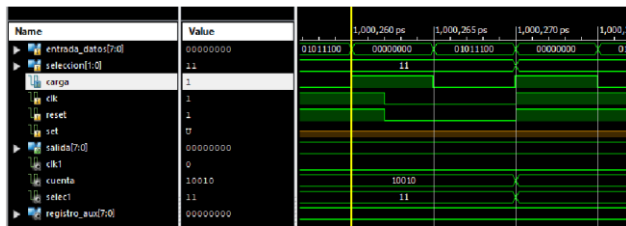


Fig. 5. Simulación 1 con entrada 00000000 y selección de 11.

## CONCLUSIONES

La practica en donde consistía en implementar un registro en VHDL, fue de gran ayuda para poder comprender de una manera mas eficiente el funcionamiento y el manejo de los registros. Los registros como sabemos se hacen registros de derecha a izquierda y viceversa, por lo que cuando se realiza un desplazamiento hacia la izquierda el MSB, es decir, el bit mas significativo y este se pierde por lo que debemos de tener en cuenta el cuidado de los desplazamientos en las tareas que tengamos que realizar.

Los registros son útiles elementos de memoria que en esta ocasión fueron fáciles de programar; fue interesante hacerlo mediante sentencias lógicas en vez de programar una función lógica, esta vez analizamos el comportamiento del circuito y en vez de pasarlo a una tabla de verdad, minimizarlo y sacar la función lógica, lo describimos tal y como lo percibimos.