

INSTITUTO POLITECNICO NACIONAL
ESCUELA SUPERIOR DE CÓMPUTO

Diseño de Sistemas Digitales
Registro Universal de 4 bits

Alcaraz Fraga Ricardo
Cruz Contreras Karen Tiffany
Montaño Morales Angeles Aranza

Grupo 2CM9
Profesor Testa Nava Alexis

Diseño de la Práctica

El objetivo de esta práctica es realizar un registro universal de 4 bits, con las siguientes operaciones:

- Carga
- Retención
- Limpieza de Registro
- Desplazamiento a la Izquierda
- Desplazamiento a la Derecha

Para las operaciones de desplazamiento; en los bits desplazados se mostrará el bit cargado en la entrada ES, por lo que al final, el registro tendrá 9 entradas:

- D0, D1, D2, D3 (los datos a cargar)
- CLK (pulso de reloj)
- CLEAR (para la limpieza del registro)
- OPER0, OPER1 (para la selección de la operación a realizar)
- ES (el bit a cargar en los bits desplazados)

Para la elaboración del registro se hizo uso de flip flops de tipo D, ya que son (en nuestra opinión) los de implementación y manejo más sencillo.

Nuestro bit menos significativo es D0.

Diseño del Circuito

Registro Universal de 4 bits

Oper1	Oper0	Operación	Oper1	Oper0	Operación
0	0	Retención	1	0	Corr a la izquierda
0	1	Carga	1	1	Corr a la derecha

Retención \rightarrow

$$Q_0 \leftarrow V_0$$

$$Q_1 \leftarrow V_1$$

$$Q_2 \leftarrow V_2$$

$$Q_3 \leftarrow V_3$$

Carga \rightarrow

$$V_0 = D_0 \quad Q_0 \leftarrow V_0$$

$$V_1 = D_1 \quad Q_1 \leftarrow V_1$$

$$V_2 = D_2 \quad Q_2 \leftarrow V_2$$

$$V_3 = D_3 \quad Q_3 \leftarrow V_3$$

Corr a la izq \rightarrow

$$V_0 = V_1 \quad Q_0 \leftarrow V_0$$

$$V_1 = V_2 \quad Q_1 \leftarrow V_1$$

$$V_2 = V_3 \quad Q_2 \leftarrow V_2$$

$$V_3 = E_5 \quad Q_3 \leftarrow V_3$$

Corr a la der \rightarrow

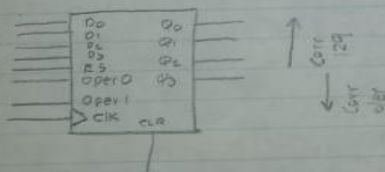
$$V_0 = E_5 \quad Q_0 \leftarrow V_0$$

$$V_1 = V_0 \quad Q_1 \leftarrow V_1$$

$$V_2 = V_1 \quad Q_2 \leftarrow V_2$$

$$V_3 = V_2 \quad Q_3 \leftarrow V_3$$

E_5 - bit que se carga en el corrimiento



Clear \rightarrow

$$V_0 = 0 \quad Q_0 \leftarrow V_0$$

$$V_1 = 0 \quad Q_1 \leftarrow V_1$$

$$V_2 = 0 \quad Q_2 \leftarrow V_2$$

$$V_3 = 0 \quad Q_3 \leftarrow V_3$$

Código Empleado

```
LIBRARY IEEE;
USE IEEE.STD_LOGIC_1164.ALL;

ENTITY RU IS
    PORT (
        D0, D1, D2, D3: IN STD_LOGIC;           -- Entradas en paralelo
        ES: IN STD_LOGIC;                       -- Bit que se cargará en los
flip flops desplazados
        OPER0, OPER1: IN STD_LOGIC;             -- Selectores de las
operaciones a realizar
        CLK, CLR: IN STD_LOGIC;                 -- Reloj y Clear
        Q0, Q1, Q2, Q3: OUT STD_LOGIC           -- Salidas
    );

    ATTRIBUTE pin_numbers OF RU: ENTITY IS
        "D0:2 D1:3 D2:4 D3:5 OPER0:6 OPER1:7 ES:8 CLR:9";

END RU;

ARCHITECTURE BEHAVE OF RU IS
BEGIN
    PROCESS (CLK)
        VARIABLE V0: STD_LOGIC;                 -- Variables para almacenar
los valores introducidos
        VARIABLE V1: STD_LOGIC;
        VARIABLE V2: STD_LOGIC;
        VARIABLE V3: STD_LOGIC;
    BEGIN
        IF (CLK='1' AND CLK'EVENT) THEN
            IF (CLR='1') THEN                     -- Operación de limpieza
                (todos los valores se van a 0)
                V0:='0';
                V1:='0';
                V2:='0';
                V3:='0';

            ELSIF (OPER1='0' AND OPER0='0') THEN    -- Operación de
retención (se mantienen los valores)
                V0:=V0;
                V1:=V1;
                V2:=V2;
                V3:=V3;

            ELSIF (OPER1='0' AND OPER0='1') THEN    -- Operación de
carga (se guardan las entradas actuales)
                V0:=D0;
                V1:=D1;
                V2:=D2;
                V3:=D3;

            ELSIF (OPER1='1' AND OPER0='0') THEN    -- Operación de
Corrimiento a la Izquierda
                V0:=V1;
                V1:=V2;
                V2:=V3;
```

```

        V3:=ES;

        ELSIF (OPER1='1' AND OPER0='1') THEN      -- Operación de
Corrimiento a la Derecha
            V3:=V2;
            V2:=V1;
            V1:=V0;
            V0:=ES;

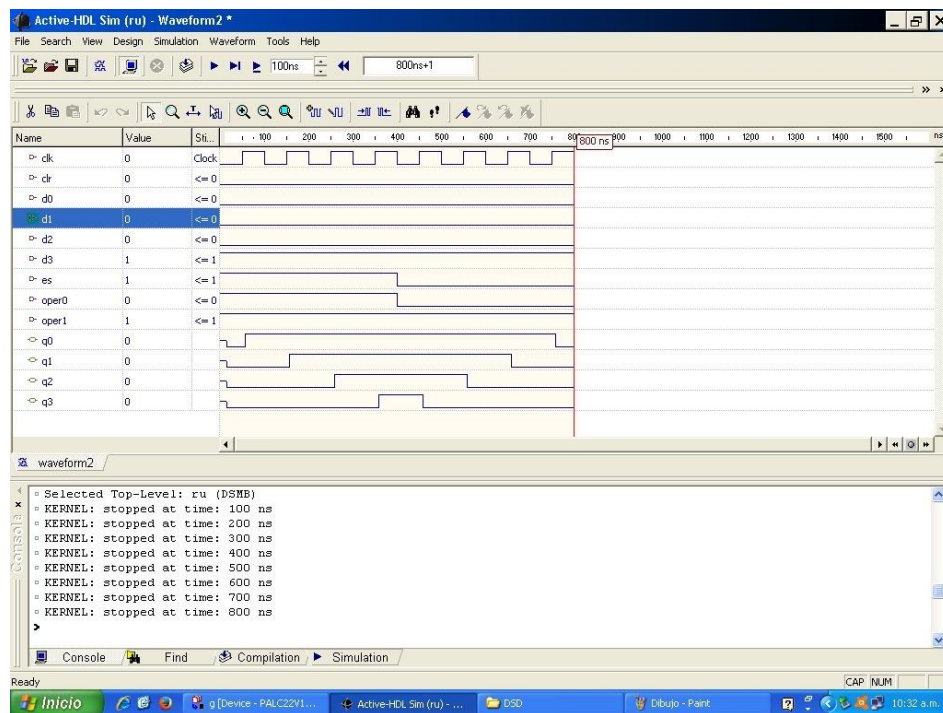
        END IF;
    END IF;

    Q0<=V0;          -- Salidas del registro
    Q1<=V1;
    Q2<=V2;
    Q3<=V3;

END PROCESS;
END BEHAVE;

```

Simulaciones



Conclusiones

El registro realizaba correctamente las operaciones de carga, retención y limpieza, pero con las operaciones de desplazamiento había un problema; no desplazaba los bits de manera sincronizada a los pulsos de reloj. Esto fue consecuencia de la manera en la que se diseñó el código, sin embargo, no nos pareció un problema ya que en las simulaciones el desplazamiento sí se realizaba de manera sincronizada con los pulsos de reloj, por lo que creímos que el diseño era correcto. No sabemos si se debe a un error del simulador, pero debido a esto nuestro circuito no realizó de manera adecuada los desplazamientos.

Los registros son importantes ya que, como hemos visto a lo largo de la asignatura, los dispositivos de almacenamiento son de suma importancia para la lógica secuencial.