



UNIVERSIDAD DE COLIMA

FACULTAD DE INGENIERÍA MECÁNICA Y ELÉCTRICA

CURSO: CIRCUITOS SECUENCIALES

ACTIVIDAD 18: Descripción de Registros en VHDL

Instructor: Dr. Ramón Antonio Félix Cuadras.

Esta actividad está planeada con un enfoque de aprendizaje colaborativo, en el cual lo más importante es el trabajo en equipo para lograr el objetivo de aprender significativamente. Para ello se formarán grupos de 3 personas, que tendrán que leer este documento. La calificación de esta actividad va de 0 a 100 y se asignará proporcionalmente al avance logrado al final de la sesión

El objetivo de esta Actividad es conocer la descripción en lenguaje VHDL de los registros (paralelo, serie-paralelo y paralelo-serie) y los contadores.

Un registro es un conjunto de N flip-flops que se usa para almacenar N bits de información. Para cada flip-flop de un registro se usa en común un reloj (CLK) y una señal de habilitación (CE), dado el caso una señal común de CLEAR ó RESET.

REGISTRO PARALELO

El diagrama esquemático de la Figura 18.1 es un registro paralelo de 4 bits, con Clock Enable (CE) y Clear asíncrono (CLR). Cuando el CE está en alto el CLR esté en bajo, el contenido de las entradas (D) se transfieren a las salidas (Q) durante el flanco positivo del reloj (CLK). Cuando CLR está en alto, se ignoran las demás señales y las salidas se ponen en un valor bajo. Cuando CE está desactivado, las transiciones de CLK son ignoradas, y las salidas Q conservan sus valores.

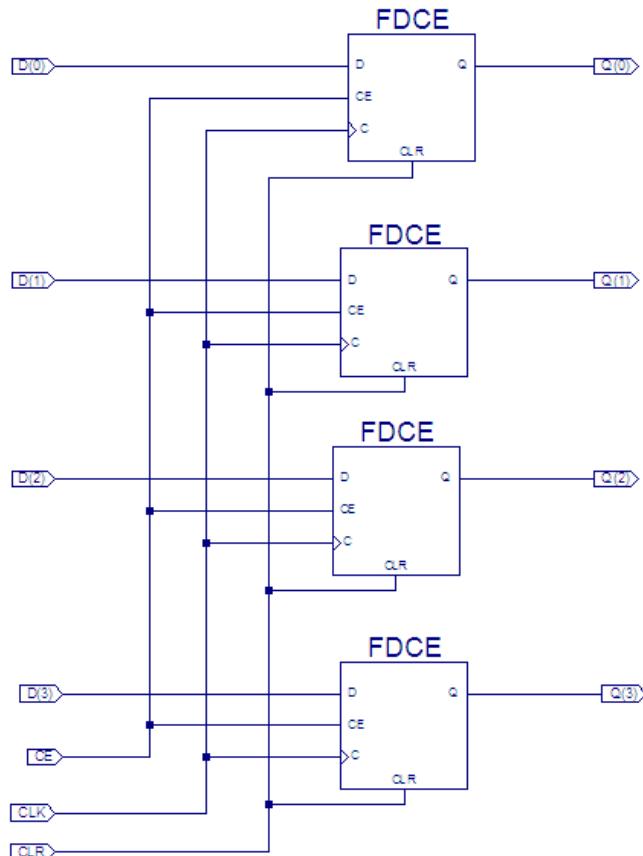


Figura 18.1: Implementación de un registro paralelo de 4 bits, a partir de FF tipo D.

La Tabla 18.1 resume el funcionamiento del un registro paralelo de la Figura 18.1.

Tabla 18.1: Tabla característica de un registro paralelo de 4 bits.

CLR	CE	CLK	Q_{+1}
'1'	x	x	"0000"
'0'	'0'	x	Q
'0'	'1'	↑	D

EJEMPLO 18.1:

Este ejemplo es un registro de 4 bits, se usan instancias del componente de FF tipo D, el código del programa queda como sigue:

```
ENTITY reg4 IS
    Port(CLK : in STD_LOGIC;
          CE : in STD_LOGIC;
          CLR : in STD_LOGIC;
          D : in STD_LOGIC_VECTOR (3 DOWNTO 0);
          Q : out STD_LOGIC_VECTOR (3 DOWNTO 0));
END reg4;

ARCHITECTURE Behavioral OF reg4 IS
    SIGNAL Qaux :STD_LOGIC_VECTOR (3 DOWNTO 0):=(OTHERS=> '0'); --Señal auxiliar
para Q.
COMPONENT FFDCE IS           --Declaración del Flip-Flop FFDCE como componente
    Port(CLK : in STD_LOGIC;   --CLOCK
          CLR : in STD_LOGIC; --CLEAR
          CE: in STD_LOGIC;   --CLOCK ENABLE
          D : in STD_LOGIC;   --Entrada D
          Q : out STD_LOGIC); --Salida Q
END COMPONENT;
BEGIN
    Q<=Qaux;                --A Q se le asigna Qaux.

--Instancia FFDCE0 del componente FFDCE
FFDCE0:FFDCE PORT MAP(CLK=>CLK, CLR=>CLR, CE=>CE, D=>D(0), Q=>Qaux(0));

--Instancia FFDCE1 del componente FFDCE
FFDCE1:FFDCE PORT MAP(CLK=>CLK, CLR=>CLR, CE=>CE, D=>D(1), Q=>Qaux(1));

--Instancia FFDCE2 del componente FFDCE
FFDCE2:FFDCE PORT MAP(CLK=>CLK, CLR=>CLR, CE=>CE, D=>D(2), Q=>Qaux(2));

--Instancia FFDCE3 del componente FFDCE
FFDCE3:FFDCE PORT MAP(CLK=>CLK, CLR=>CLR, CE=>CE, D=>D(3), Q=>Qaux(3));

END Behavioral;
```

Este diseño utiliza instancias del componente FFDCE, que es un flip-flop con Clear y Clock Enable. El uso de instancias de componentes es la forma más usual de reutilizar un diseño hecho en VHDL, además da estructura al diseño final (módulo TOP).

REGISTRO SERIE-PARALELO

Un registro serie-paralelo es un registro de corrimiento con una entrada serial de 1 bit y una salida de N bits en paralelo, donde N es el número de flip-flop que contiene el registro. En la Figura 18.4 se muestra un registro serie-paralelo de 4 bits con entrada serie SLI (Shift-Left Input), salida paralela (Q) Clock Enable (CE) y Clear asíncrono (CLR).

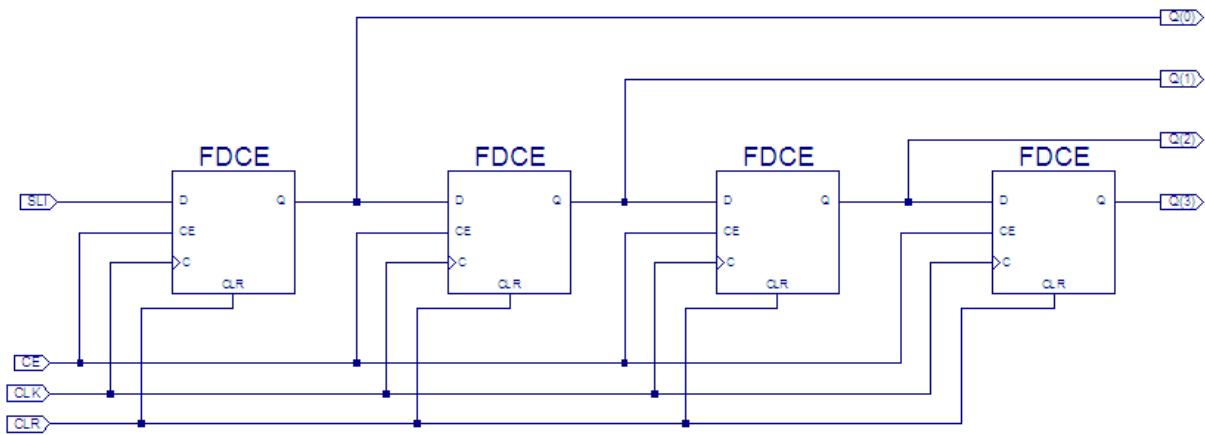


Figura 18.4: Registro Serie- Paralelo de 4 bits con CLR y CE.

Cuando CE está activo y CLR este bajo, el dato en la entrada SLI se transfiere al bit menos significativo Q(0) del registro, en el flanco de subida del reloj (CLK). Durante los subsecuentes flancos de positivos en CLK, mientras CE este alto y CLR bajo, el dato se desplazará al siguiente bit, mientras un nuevo dato se carga en Q(0), es decir, SLI \rightarrow Q(0), Q(0) \rightarrow Q(1), Q(1) \rightarrow Q(2), Q(2) \rightarrow Q(3). El registro ignora las transiciones de reloj cuando CE es bajo.

La Tabla 18.2 resume el funcionamiento del un registro serie-paralelo de la Figura 18.4.

Tabla 18.2: Tabla característica de un registro serie-paralelo de 4 bits.

CLR	CE	CLK	Q(3) ₊₁	Q(2) ₊₁	Q(1) ₊₁	Q(0) ₊₁	Q ₊₁
'1'	x	x	'0'	'0'	'0'	'0'	"0000"
'0'	'0'	x	Q(3)	Q(2)	Q(1)	Q(0)	Q
'0'	'1'	\uparrow	Q(2)	Q(1)	Q(0)	SLI	Q(2 downto 0) & SLI

REGISTRO PARALELO-SERIE

Un registro paralelo-serie tiene 2 modos de operación: “carga paralela”, en el que un dato de N bits se carga a las salidas de N flip-flop, como pasa en un registro paralelo, el otro modo es de “corrimiento”, en el cual los bits se van recorriendo del bit menos al más significativo en cada flanco (positivo ó negativo) del reloj, al igual que sucede con un registro serie.

En la Figura 18.9 se puede apreciar un registro de 4 bits paralelo-serie, con Clear y CE y corrimiento/carga (SHIFT_LOAD). Cuando CLR y SHIFT_LOAD están en bajo y CE='1', entonces un '0' (GND) se pasa al bit menos significativo (Q(0)) del registro, en el flanco de subida del reloj, al mismo tiempo el dato de la señal Q(0) se desplazará a Q(1), Q(1) se transferirá a Q(2) y por último Q(2) se pasa al puerto de salida SLO (Shift-Left Output), es decir: Q(0) \leq '0', Q(1) \leq =Q(0), Q(2) \leq =Q(1), SLO \leq =Q(2)). El registro ignora las transiciones de reloj cuando CE es bajo. Note dos cosas importantes: el dato que inicialmente estaba en SLO se pierde en el primer pulso de CLK, además después de 4 desplazamientos todos los flip-flops tendrán sus salidas en cero. Cuando CE y SHIFT_LOAD estén en '1', CLR este bajo, el dato en la entrada D se transfiere a los flip-flops que forman el registro, en el siguiente flanco positivo de CLK; pero si CE está activo.

Figura 18.9: Registro Paralelo-Serie de 4 bits con CLR y CE.

La Tabla 18.3 resume el funcionamiento del un registro paralelo-serie de la Figura 18.4.

Tabla 18.3: Tabla característica de un registro paralelo-serie de 4 bits.

CLR	CE	CLK	SHIFT_LOAD	Q(3) ₊₁	Q(2) ₊₁	Q(1) ₊₁	Q(0) ₊₁	Q ₊₁
'1'	×	×	×	'0'	'0'	'0'	'0'	"0000"
'0'	'0'	×	×	Q(3)	Q(2)	Q(1)	Q(0)	Q
'0'	'1'	↑	'0'	Q(2)	Q(1)	Q(0)	'0'	Q(2 downto 0) & '0'
'0'	'1'	↑	'1'	D(3)	D(2)	D(1)	D(0)	D

Un registro paralelo-serie funciona como un transmisor en un sistema de comunicación serie síncrona, el dato entra en paralelo al registro a través del puerto D, éste es convertido a un tren de pulsos en el puerto SLO, transmitiéndose en sincronía con el flanco de subida de CLK, en el otro extremo de la comunicación debe haber un registro serie-paralelo para decodificar el tren de pulsos y reconstruir el dato que se introdujo por el puerto D del registro paralelo-serie.

PROBLEMA 18.1

a) Escriba una versión en VHDL del registro serie-paralelo de 4 bits que se ha analizado, utilice 4 instancias del flip-flop tipo D con Clear y CE, como se hizo en el Ejemplo 18.1.

b) Escriba una versión en VHDL del registro paralelo-serie de 4 bits que se ha analizado, utilice 4 instancias del flip-flop tipo D con Clear y CE, como se hizo en el Ejemplo 18.1.

Llame al instructor para registrar el avance. (30 puntos).

DESCRIPCIÓN ALGORÍTMICA DE REGISTROS EN VHDL

El siguiente código en VHDL es la descripción algorítmica de un registro paralelo de 4 bits con Clear y CE.

```
ENTITY registro_p IS
    PORT (D :IN STD_LOGIC_VECTOR(3 DOWNTO 0);
          CLK : IN STD_LOGIC;
          CLR : IN STD_LOGIC;
          CE : IN STD_LOGIC;
          Q : OUT STD_LOGIC_VECTOR(3 DOWNTO 0));
END registro_p;

ARCHITECTURE Behavioral OF registro_p IS
BEGIN
PROCESS(CLK,CLR)
BEGIN
    IF CLR='1' THEN
        Q <= (OTHERS=>'0');
    ELSIF RISING_EDGE(CLK) AND CE='1' THEN
        Q <= D;
    END IF;
END PROCESS;
END Behavioral;
```

El siguiente código en VHDL es la descripción algorítmica de un registro serie-paralelo de 4 bits con Clear y CE.

```
ENTITY registro_sp IS
    PORT (SLI:IN STD_LOGIC;
          CLK: IN STD_LOGIC;
          CLR: IN STD_LOGIC;
          CE: IN STD_LOGIC;
          Q: OUT STD_LOGIC_VECTOR(3 DOWNTO 0));
END registro_sp;
```

```
ARCHITECTURE behavioral OF registro_sp IS
SIGNAL QAUX: STD_LOGIC_VECTOR(3 DOWNTO 0):=(OTHERS=>'0');
BEGIN
    Q<=QAUX;
PROCESS(CLK,CLR)
BEGIN
    IF CLR='1' THEN
        QAUX <= (OTHERS=>'0');
    ELSIF RISING_EDGE(CLK) AND CE='1' THEN
        --Corrimiento: por las piedritas
    END IF;
END PROCESS;
END behavioral;
```

```
--QAUX(0)<=SLI; QAUX(1)<=QAUX(0); QAUX(2)<=QAUX(1); QAUX(3)<=QAUX(2);
--Corrimiento: como se debe de hacer
    QAUX<=QAUX(2 DOWNTO 0) & SLI;
END IF;
END PROCESS;
END behavioral;
```

PROBLEMA 18.2

Diseñe en VHDL un registro paralelo-serie de 4 bits con descripción algorítmica, como si hizo en los códigos anteriores. **Llame al instructor para registrar el avance. (40 puntos).**