

PRÁCTICA 7

Diseño de divisores de frecuencia y contadores

OBJETIVO:

El alumno aprenderá a diseñar divisores de frecuencia, con el fin de visualizar cualquier sistema secuencial en tarjetas de desarrollo con FPGA, las cuales **generalmente** tienen un reloj de cristal de 50 Mhz, frecuencia muy rápida para ser detectada por el ojo humano.

Se llama divisor de frecuencia a un dispositivo que produce a su salida una frecuencia menor que la de entrada y suelen estar formados por **contadores digitales**.

Un contador es un circuito secuencial construido a partir de Flip-Flops y compuertas lógicas.

INTRODUCCIÓN:

Si a un Flip-Flop JK, todas sus entradas se le conectan a Vcc, excepto la entrada del reloj, a su salida, dicho Flip-Flop fluctúa entre cero y uno cada vez que el flanco activo del reloj se presente y si se interconecta la salida del primero al reloj del segundo y la salida del segundo al reloj del tercero, y así sucesivamente como se muestra en la figura 1, las salidas de los Flip-Flops, fluctúan en diferentes tiempos, ya que cada uno de ellos, tiene diferente entrada de reloj.

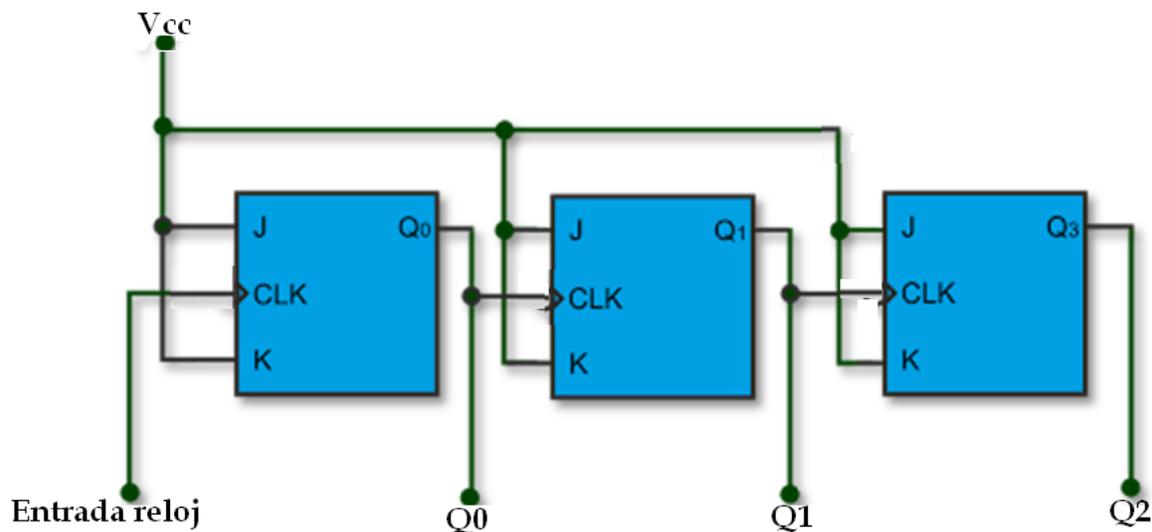
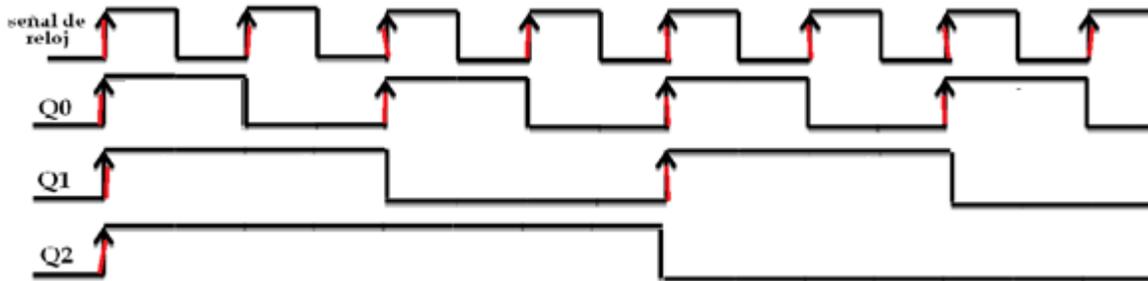


Figura 1. Flip-Flops interconectados mediante la unión de la salida de uno a la entrada de reloj del siguiente

La figura 2 muestra el diagrama de tiempos de las salidas en los Flip-Flops interconectados en la figura 1. En rojo se muestra el flanco activo.



Observando la figura 2 y poniendo leds a las salidas Q2 Q1 Q0 (en ese orden) se tiene un contador descendente, que cuenta del 7 al cero.

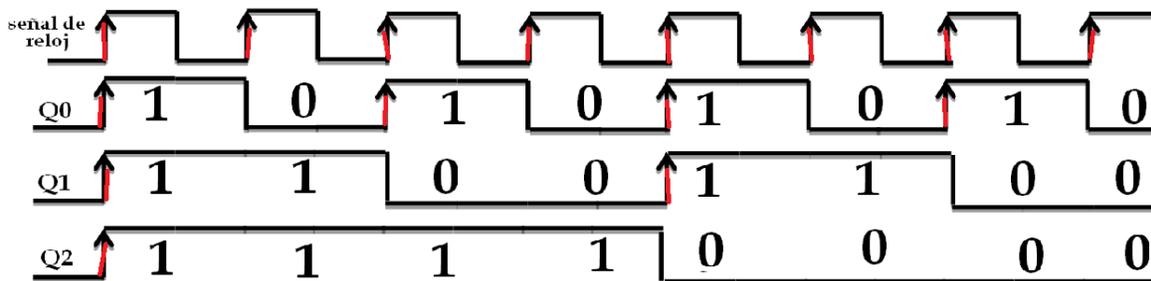


Figura 2. Diagrama de tiempos de las salidas en los Flip-Flops

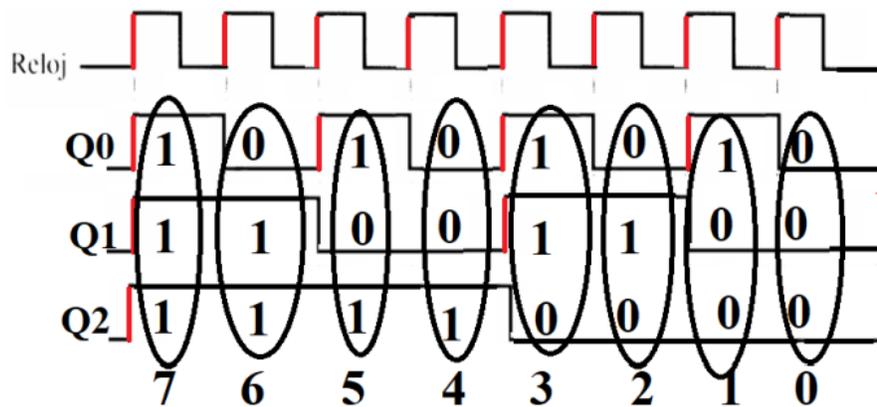


Figura 3. Contador descendente, que cuenta del 7 al cero.

También se observa que si la salida Q2, se selecciona como entrada de reloj para cualquier sistema digital, se obtuvo un divisor de frecuencia respecto a la entrada del reloj original. Como se observa en la figura 4.

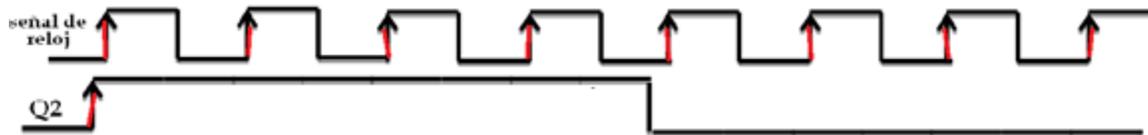


Figura 4. Divisor de frecuencia.

ESPECIFICACIONES:

Diseñar un contador de 28 bits, del cual únicamente se tomará en cuenta la salida del bit 26 para utilizarla como el reloj de cualquier diseño secuencial implementado en tarjetas con reloj de 50 Mhz.

Esto hará que la entrada del reloj, tenga un retraso de 226 bits, dando un valor en hexadecimal de **48009E0** asegurando con esto un tiempo de 1.51 segundos entre cada pulso del reloj.

Posteriormente diseñar un contador binario de cuatro bits que cuente del 9 al cero.

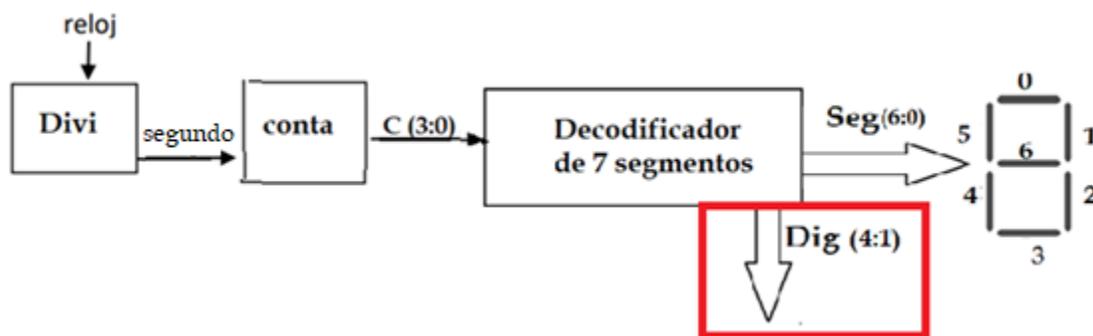
Para contar del nueve al cero se requiere de un contador de 4 bits, con el que se puede obtener la cuenta del 15 al cero por lo que se requiere que de cada vez que se llegue al número cero se reinicie la cuenta.

Finalmente se requiere visualizar la cuenta en un display de 7 segmentos.

Nota:

Para tarjetas de desarrollo con los displays de 7 segmentos conectados en paralelo, se requiere adicionar un vector de salida y código extra los cuales se muestran en rojo.

DIAGRAMA DE BLOQUES:



Bloque Divi:

Internamente generaremos un contador de 28 bits de salida, del cual únicamente se tomará en cuenta la salida 26 para utilizarla como el reloj del contador.

Esto hará que la entrada del reloj de 50 Mhz tenga un retraso de 2^{26} bits, dando un valor aproximado en hexadecimal de 48009E0 asegurando con esto un tiempo de 1.51 segundos entre cada pulso del reloj. A esa salida la llamaremos Segundo

Bloque Conta:

Internamente se genera un contador de 4 bits, del cual se reinicia, cada vez que llegue al número cero, debido a que con 4 bits, es capaz de contar hasta el 15.

Bloque Deco:

En este bloque se requiere hacer una tabla de verdad la cual tiene por entrada un vector C de cuatro bits y a su salida requiere decodificar a 7 segmentos cada combinación ese vector.

Código en VHDL

```
library IEEE;
use IEEE.std_logic_1164.ALL;
use IEEE.std_logic_arith.ALL;
use IEEE.std_logic_unsigned.ALL;

entity contador is
  Port (reloj : in  STD_LOGIC;
        Dig : out STD_LOGIC_VECTOR (4 downto 1);
        Seg : out STD_LOGIC_VECTOR (6 downto 0);
end contador;
```



```
architecture Behavioral of contador is
  signal segundo: std_logic;
  signal C: std_logic_vector (3 downto 0);
begin

  Divi: process (reloj)
    variable cuenta: std_logic_vector(27 downto 0) := x"0000000";
  begin
    if rising_edge (reloj) then
      if cuenta = x"48009e0" then
        cuenta := x"0000000";
      else
        cuenta := cuenta + 1;
      end if;
    end if;
    segundo <= cuenta(24);
  end process;

  conta: process (segundo)
    variable cuenta : std_logic_vector(3 downto 0) := "1001";
  begin
    if rising_edge (segundo) then
      if cuenta = "0000" then
        cuenta := "1001";
      else
        cuenta := cuenta - 1;
      end if; end if;
      C <= cuenta;
    end process;
```



with C select

```
Seg <= "1000000" when "0000", --0
      "1111001" when "0001", --1
      "0100100" when "0010", --2
      "0110000" when "0011", --3
      "0011001" when "0100", --4
      "0010010" when "0101", --5
      "0000010" when "0110", --6
      "1111000" when "0111", --7
      "0000000" when "1000", --8
      "0010000" when "1001", --9
      "1000000" when others; --cero
```

with C select

```
Dig <= "1110" when "0000",
      "1110" when others;
```

end Behavioral;



<https://www.youtube.com/watch?v=hFA8aQjvcqQ>



Prácticas en lenguaje VHDL

M.I. Norma Elva Chávez Rodríguez

Division de Ingeniería Eléctrica

Departamento de Computacion