

Práctica 2.

DISEÑO DE REGISTROS DE CORRIMIENTO EN CASCADA

OBJETIVO:

Demostrar a los estudiantes que las declaraciones secuenciales requieren de un orden para ser ejecutadas. Diseñar registros de corrimiento en cascada utilizando las estructuras de control *if-then-else* o *case* dentro de un proceso.

ESPECIFICACIONES:

Utilizando un FPGA y 8 displays de 7 segmentos, diseñar un sistema digital que despliegue un mensaje que se desplace en los displays.

La figura 2.1 muestra el diagrama de bloques del sistema registros de corrimiento en cascada.

DIAGRAMA DE BLOQUES:

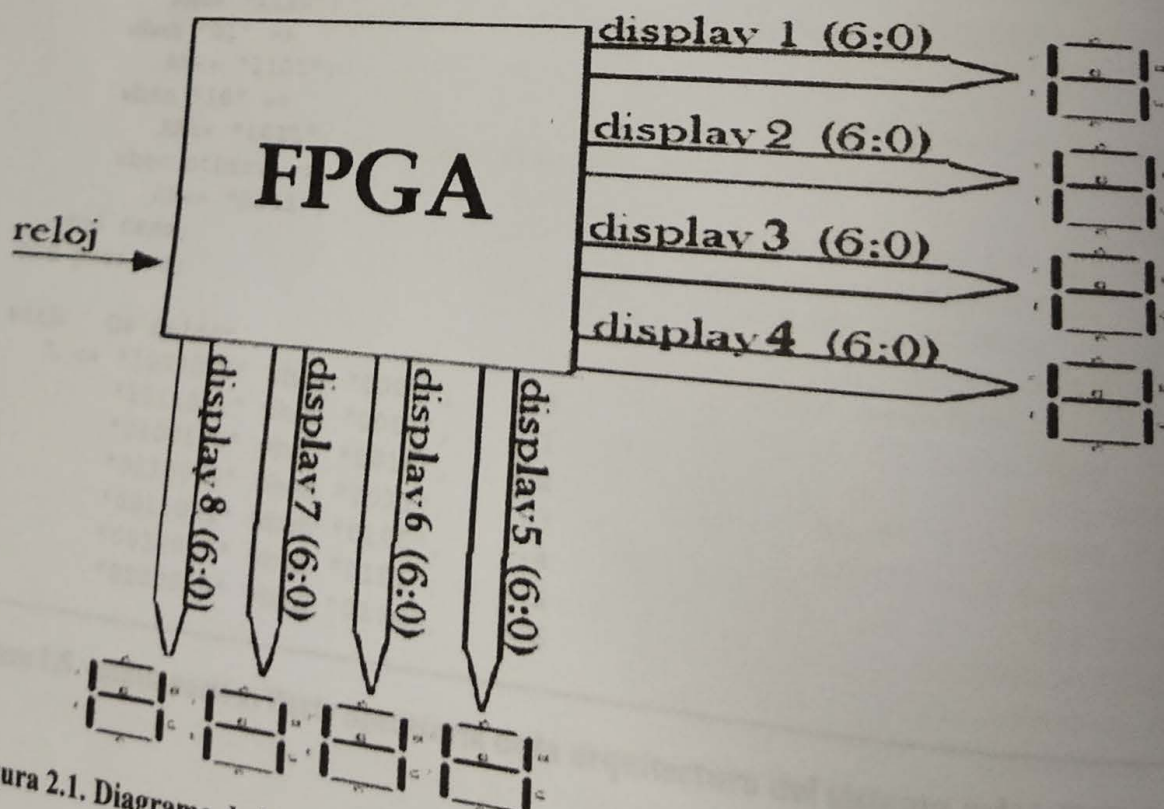


Figura 2.1. Diagrama de bloques del sistema registros de corrimiento en cascada. [2]

Dentro del sistema digital registros de corrimiento en cascada se tienen varios bloques funcionales, los cuales internamente ejecutan instrucciones en forma secuencial. La figura 2.2 muestra los bloques funcionales del sistema.

BLOQUES FUNCIONALES:

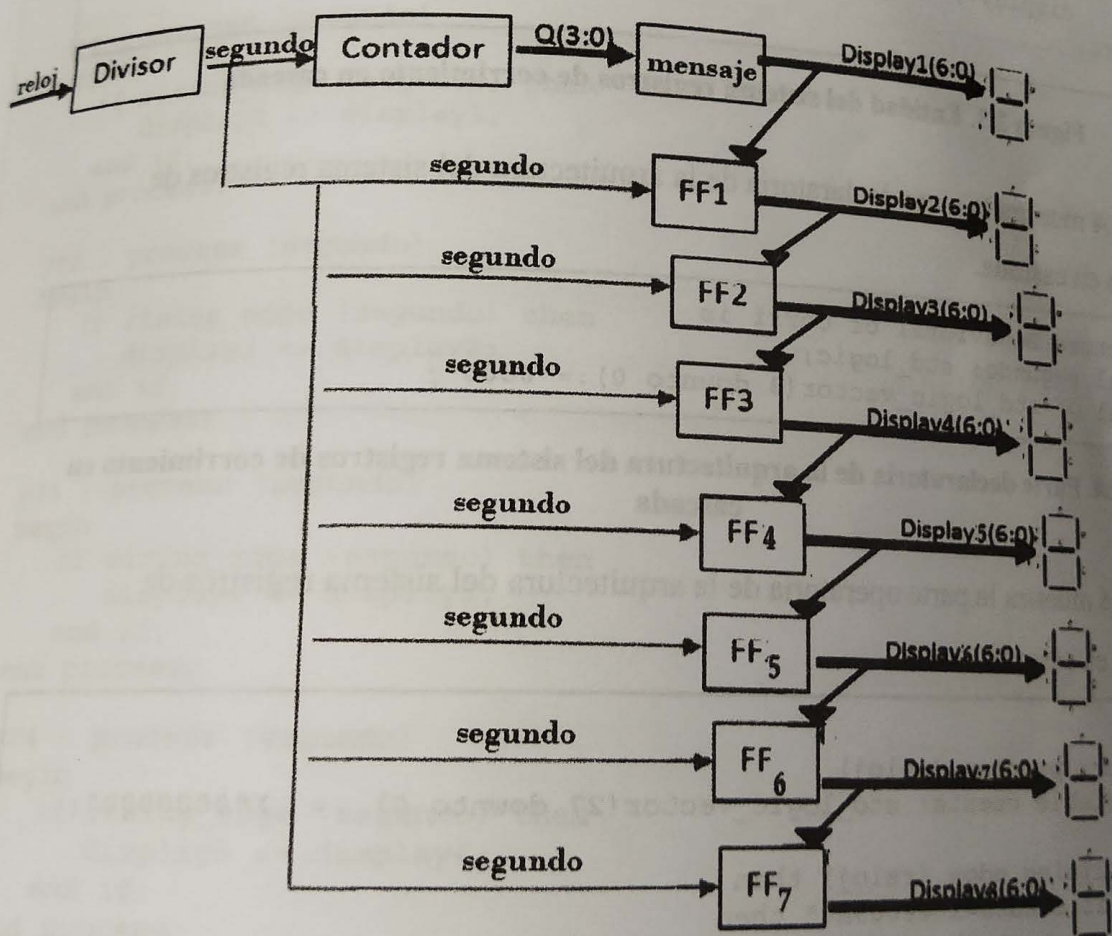


Figura 2.2. Diagrama de bloques funcionales del sistema registros de corrimiento en cascada

La figura 2.3 muestra la entidad del sistema registros de corrimiento en cascada.

```

library IEEE;
use IEEE.std_logic_1164.all;
use IEEE.std_logic_arith.all;
use IEEE.std_logic_unsigned.all;

entity corri is
  Port (reloj: in std_logic;
        display1, display2, display3, display4, display5, display6,
        display7, display8: buffer std_logic_vector (6 downto 0));
end corri;
  
```

Figura 2.3. Entidad del sistema registros de corrimiento en cascada

La figura 2.4 muestra la parte declaratoria de la arquitectura del sistema registros de corrimiento en cascada.

```

architecture Behavioral of corri is
  signal segundo: std_logic;
  signal Q: std_logic_vector(3 downto 0) := "0000";
end architecture Behavioral of corri;
  
```

Figura 2.4. Parte declaratoria de la arquitectura del sistema registros de corrimiento en cascada

La figura 2.5 muestra la parte operatoria de la arquitectura del sistema registros de corrimiento en cascada.

```

begin
  divisor: process (reloj)
    variable cuenta: std_logic_vector(27 downto 0) := X"00000000";
  begin
    if rising_edge (reloj) then
      if cuenta=X"48009E0" then
        cuenta:= X"00000000";
      else
        cuenta:=cuenta+1;
      end if;
    end if;
    segundo <= cuenta(22);
  end process;

  contador: process (segundo)
  begin
    if rising_edge (segundo) then
      Q <= Q +1;
    end if;
  end process;
end;
  
```

Figura 2.5. Parte operatoria de la arquitectura del sistema registros de corrimiento en cascada

```

with Q select
    display1 <= "0000110" when "0000", -- E
                "0101011" when "0001", -- n
                "1111111" when "0010", -- espacio
                "1000111" when "0011", -- L
                "0001000" when "0100", -- A
                "1111111" when "0101", -- espacio
                "1000000" when "0110", -- O
                "1000111" when "0111", -- L
                "0001000" when "1000", -- A
                "1111111" when others; -- espacios

FF1 : process (segundo)
begin
    if rising_edge (segundo) then
        display2 <= display1;
    end if;
end process;

FF2 : process (segundo)
begin
    if rising_edge (segundo) then
        display3 <= display2;
    end if;
end process;

FF3 : process (segundo)
begin
    if rising_edge (segundo) then
        display4 <= display3;
    end if;
end process;

FF4 : process (segundo)
begin
    if rising_edge (segundo) then
        display5 <= display4;
    end if;
end process;

FF5 : process (segundo)
begin
    if rising_edge (segundo) then
        display6 <= display5;
    end if;
end process;

FF6 : process (segundo)
begin
    if rising_edge (segundo) then
        display7 <= display6;
    end if;
end process;
    
```


Figura 2.5. (continuación) Parte operatoria de la arquitectura del sistema registros de corrimiento en cascada

```

FF7 : process (segundo)
begin
  if rising_edge (segundo) then
    display8 <= display7;
  end if;
end process;
end behavioral;

```

Figura 2.5. (continuación) Parte operatoria de la arquitectura del sistema registros de corrimiento en cascada

ACTIVIDAD COMPLEMENTARIA:

Diseñar un sistema que realice la venta de bebidas de 4 diferentes sabores, cada bebida vale \$15, se aceptan billetes de \$100, \$50, \$20 y monedas de \$1, \$2 \$5 y \$10 y da cambio. Cuando el sistema esté encendido y nadie esté comprando se activará una grabación invitando a consumir esas bebidas.