**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.

**Introducción**

En esta práctica aprenderemos a usar los corrimientos, para esto es necesario comprender diferentes conceptos muy importantes. Primero un corrimiento de registro se refiere a una operación en la que los bits de un registro (una serie de flip-flops) se desplazan o corren a través de las celdas del registro en una dirección específica. Esto se hace típicamente de manera secuencial, y es comúnmente utilizado en la electrónica digital para diversas aplicaciones, incluyendo la visualización de información en displays de 7 segmentos.

En este contexto, un "corrimiento de registro" se utiliza para mover una serie de bits que representan caracteres o números a través de un conjunto de displays de 7 segmentos para mostrar un mensaje. Para mostrar un mensaje en un display de 7 segmentos utilizando un corrimiento de registro y una tarjeta FPGA, se siguen estos pasos generales:

Se utiliza un registro de desplazamiento para almacenar los datos que se desean mostrar en el display de 7 segmentos.

Los datos del mensaje que se quiere mostrar se cargan inicialmente en el registro.

Se inicia la operación de corrimiento de registro, donde los bits almacenados en el registro se desplazan de una celda a otra en una dirección específica.

Cada conjunto de bits que se encuentra en una posición específica del registro se utiliza para encender los segmentos correspondientes en el display de 7 segmentos. Cada segmento representa un segmento de un número o carácter.

El proceso de corrimiento de registro y decodificación se repite continuamente para que el mensaje se muestre en forma secuencial en el display de 7 segmentos.

**Desarrollo**

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 este encendido y nadie este comprando se activará una grabación invitando a consumir esas bebidas.

En esta se requirió diferentes conocimientos principalmente se requirió comprender como funciona el registro de corrimiento, en específico en el código proporcionado como es que se muestra el mensaje y como este se recorre a través de los diferentes leds.

Para mostrar el mensaje mediante el uso de displays de 7 segmentos se tuvo que recordar la disposición de los displays y saber la disposición de pines a usar. Para que se muestren las palabras deseadas, usando el esquema siguiente.

Pantalla de video juego con un reloj

Descripción generada automáticamente con confianza media

Para la codificación del mensaje se usó la siguiente relación:

Para la letra: c en 7 segmentos es: 1000110

Para la letra: o en 7 segmentos es: 0100011

Para la letra: n en 7 segmentos es: 0101011

Para la letra: p en 7 segmentos es: 0001100

Para la letra: r en 7 segmentos es: 0101111

Para la letra: e en 7 segmentos es: 0000110

Para el espacio en 7 segmentos es: 1111111

Para la letra: 1 en 7 segmentos es: 1001111

Para la letra: 5 en 7 segmentos es: 0010010

Para la letra: a en 7 segmentos es: 0001000

Para la letra: f en 7 segmentos es: 0001110

Para la letra: t en 7 segmentos es: 0000111

Para la letra: s en 7 segmentos es: 0010010

Para la letra: i en 7 segmentos es: 1111011

Para la letra: 7 en 7 segmentos es: 0111000

Para la letra: u en 7 segmentos es: 1000001

Para la letra: b en 7 segmentos es: 0000011

Para la letra: l en 7 segmentos es: 1000111

Para la letra: d en 7 segmentos es: 0100001

Para la letra: 0 en 7 segmentos es: 1000000

Para la letra: 2 en 7 segmentos es: 0100100

Para la letra: y en 7 segmentos es: 0010001

El mensaje mostrado es el siguiente:

COMPREEEEEEN

Coca 15

Fanta 15

Sprite 15

7UP 15

SE ACEPTAN BILLETES DE 100 50 20

SE ACEPTAN MONEDAS DE 1 2 5 Y 10

SE DA CAMBIO

**Código del programa**

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

use ieee.std\_logic\_arith.all;

use ieee.std\_logic\_unsigned.all;

entity p9 is

Port ( clk: in std\_logic;

display0, display1, display2,display3,display4,display5: buffer std\_logic\_vector(6 downto 0)

);

end p9;

architecture Behavioral of p9 is

--declaratoria de la arquitectura del sistema registros de corrimineto en cascada

signal segundo: std\_logic;

signal Q: std\_logic\_vector(6 downto 0):="1111111";

begin

--a continuacion se muestra la parte operatoria de

--la arquitectura del sistema resgistros de corrimineto en cascada.

divisor:process(clk)

variable cuenta: std\_logic\_vector(27 downto 0):=X"0000000";

begin

if rising\_edge(clk)then

if(cuenta=X"48009E0")then

cuenta:=X"0000000";

else

cuenta:=cuenta+1;

end if;

end if;

segundo<=cuenta(23);

end process;

contador:process(clk)

begin

if rising\_edge(segundo)then

Q<=Q+1;

end if;

end process;

with Q select

display0 <= "1000110"when"0000000",

"0100011"when"0000001",

"0101011"when"0000010",

"0101011"when"0000011",

"0001100"when"0000100",

"0101111"when"0000101",

"0000110"when"0000110",

"0000110"when"0000111",

"0000110"when"0001000",

"0000110"when"0001001",

"0000110"when"0001010",

"0101011"when"0001011",

"1111111"when"0001100",

"1000110"when"0001101",

"0100011"when"0001110",

"1000110"when"0001111",

"0001000"when"0010000",

"1111111"when"0010001",

"1001111"when"0010010",

"0010010"when"0010011",

"1111111"when"0010100",

"0001110"when"0010101",

"0001000"when"0010110",

"0101011"when"0010111",

"0000111"when"0011000",

"0001000"when"0011001",

"1111111"when"0011010",

"1001111"when"0011011",

"0010010"when"0011100",

"1111111"when"0011101",

"0010010"when"0011110",

"0001100"when"0011111",

"0101111"when"0100000",

"1111011"when"0100001",

"0000111"when"0100010",

"0000110"when"0100011",

"1111111"when"0100100",

"1001111"when"0100101",

"0010010"when"0100110",

"1111111"when"0100111",

"0111000"when"0101000",

"1000001"when"0101001",

"0001100"when"0101010",

"1111111"when"0101011",

"1001111"when"0101100",

"0010010"when"0101101",

"1111111"when"0101110",

"0010010"when"0101111",

"0000110"when"0110000",

"1111111"when"0110001",

"0001000"when"0110010",

"1000110"when"0110011",

"0000110"when"0110100",

"0001100"when"0110101",

"0000111"when"0110110",

"0001000"when"0110111",

"0101011"when"0111000",

"1111111"when"0111001",

"0000011"when"0111010",

"1111011"when"0111011",

"1000111"when"0111100",

"1000111"when"0111101",

"0000110"when"0111110",

"0000111"when"0111111",

"0000110"when"1000000",

"0010010"when"1000001",

"1111111"when"1000010",

"0100001"when"1000011",

"0000110"when"1000100",

"1111111"when"1000101",

"1001111"when"1000110",

"1000000"when"1000111",

"1000000"when"1001000",

"1111111"when"1001001",

"0010010"when"1001010",

"1000000"when"1001011",

"1111111"when"1001100",

"0100100"when"1001101",

"1000000"when"1001110",

"1111111"when"1001111",

"0010010"when"1010000",

"0000110"when"1010001",

"1111111"when"1010010",

"0001000"when"1010011",

"1000110"when"1010100",

"0000110"when"1010101",

"0001100"when"1010110",

"0000111"when"1010111",

"0001000"when"1011000",

"0101011"when"1011001",

"1111111"when"1011010",

"0101011"when"1011011",

"0101011"when"1011100",

"0100011"when"1011101",

"0101011"when"1011110",

"0000110"when"1011111",

"0100001"when"1100000",

"0001000"when"1100001",

"0010010"when"1100010",

"1111111"when"1100011",

"0100001"when"1100100",

"0000110"when"1100101",

"1111111"when"1100110",

"1001111"when"1100111",

"1111111"when"1101000",

"0100100"when"1101001",

"1111111"when"1101010",

"0010010"when"1101011",

"1111111"when"1101100",

"0010001"when"1101101",

"1111111"when"1101110",

"1001111"when"1101111",

"1000000"when"1110000",

"1111111"when"1110001",

"0010010"when"1110010",

"0000110"when"1110011",

"1111111"when"1110100",

"0100001"when"1110101",

"0001000"when"1110110",

"1111111"when"1110111",

"1000110"when"1111000",

"0001000"when"1111001",

"0101011"when"1111010",

"0101011"when"1111011",

"0000011"when"1111100",

"1111011"when"1111101",

"0100011"when"1111110",

"1111111" when others; --espacios

FF1: process(segundo)

begin

if rising\_edge(segundo)then

display1<=display0;

end if;

end process;

FF2: process(segundo)

begin

if rising\_edge(segundo)then

display2<=display1;

end if;

end process;

FF3: process(segundo)

begin

if rising\_edge(segundo)then

display3<=display2;

end if;

end process;

FF4: process(segundo)

begin

if rising\_edge(segundo)then

display4<=display3;

end if;

end process;

FF5: process(segundo)

begin

if rising\_edge(segundo)then

display5<=display4;

end if;

end process;

end Behavioral;

**Videos del funcionamiento del programa:**

<https://youtu.be/VAONE-AWvt0?si=WzbUKNewiLebbIg6>

<https://youtu.be/rGdRwUptJfo>