**UNIVERSIDAD DE GUADALAJARA**

**CUCEI**

CARRERA: **INGENERIA EN COMPUTACION**

SIMULACION 9: **PROGRAMACIÓN DE SISTEMAS DIGITALES EN VHDL**

ALUMNO: **EFRAIN ROBLES PULIDO**

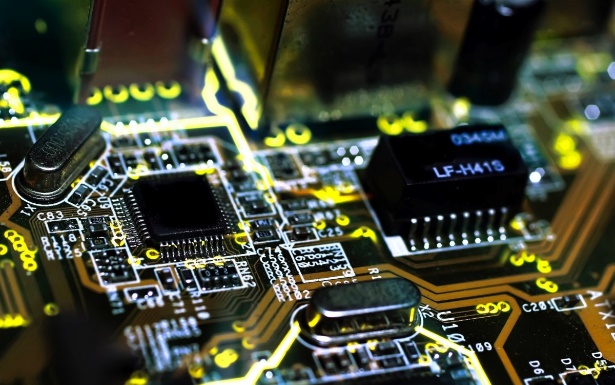
CODIGO: **221350095**

*NOMBRE DE LA MATERIA:* **ARQUITECTURA DE COMPUTADORAS**

*SECCIÓN:*  **D10** *CALENDARIO:* **2021B**

*NOMBRE DE LA PROFESORA:* **THELMA ISABEL MORALES RAMIREZ**





**Simulación 9\_Codigo VHDL**

1. Realice un programa de un multiplexor de 1 bit de salida y 8 entradas, como el que se muestra en la figura. Desarrolle su código de programación con base a la tabla de verdad.

Diagrama, Esquemático

Descripción generada automáticamenteTexto, Tabla

Descripción generada automáticamente

* **Ynot = ~Y**

**Codigo VHDL:**

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

use IEEE.STD\_LOGIC\_UNSIGNED.ALL;

entity Multiplexor is

Port ( E0 : in STD\_LOGIC;--Entradas

E1 : in STD\_LOGIC;

E2 : in STD\_LOGIC;

E3 : in STD\_LOGIC;

E4 : in STD\_LOGIC;

E5 : in STD\_LOGIC;

E6 : in STD\_LOGIC;

E7 : in STD\_LOGIC;

--Selectores (A,B,C, EN)

sel: in STD\_LOGIC\_VECTOR (3 downto 0);

--Salidas

Y : inout STD\_LOGIC;

Ynot : out STD\_LOGIC);

end Multiplexor;

architecture Behavioral of Multiplexor is

begin

with sel select

Y <= E0 when "0000",

E1 when "0001",

E2 when "0010",

E3 when "0011",

E4 when "0100",

E5 when "0101",

E6 when "0110",

E7 when "0111",

'0' when others;

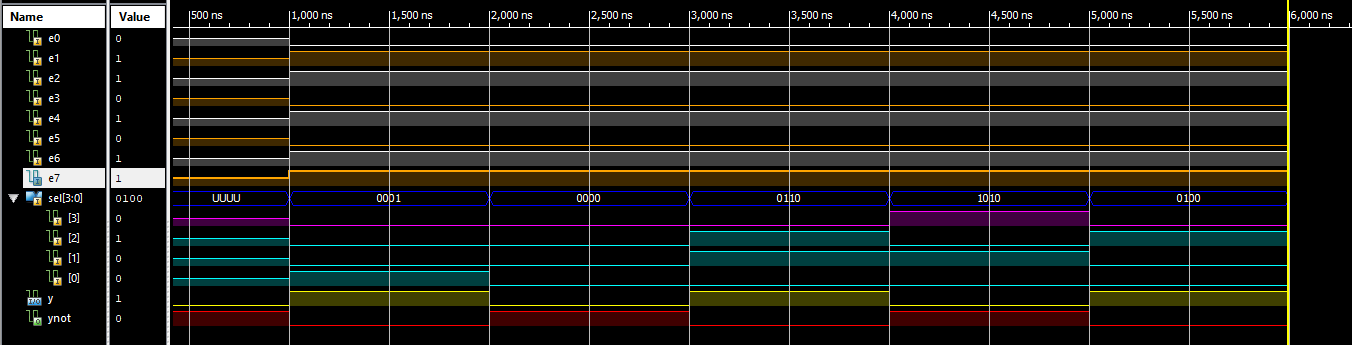
Ynot<= not Y;

end Behavioral;

Texto

Descripción generada automáticamente con confianza media

**Cronograma:**



1. Diseñe un registro de 4 bits como el mostrado en la figura, cuyo funcionamiento se encuentra regulado por las señales de control S0 y S1, tal y como se muestra en la siguiente tabla.

Diagrama

Descripción generada automáticamenteInterfaz de usuario gráfica, Texto, Tabla

Descripción generada automáticamente

**Codigo VHDL:**

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

use IEEE.STD\_LOGIC\_UNSIGNED.ALL;

entity Registro is

Port ( D : in STD\_LOGIC\_VECTOR (3 downto 0);

rst : in STD\_LOGIC;

CLK : in STD\_LOGIC;

S : in STD\_LOGIC\_VECTOR (1 downto 0);

Q : inout STD\_LOGIC\_VECTOR (3 downto 0));

end Registro;

architecture Behavioral of Registro is

begin

process(CLK,D,S,rst)

begin

if (rst='1') then

Q<="0000";

elsif (CLK' event and CLK='1') then

if (S="00") then

Q<=Q;

elsif (S="01") then

Q(3)<=Q(2);

Q(2)<=Q(1);

Q(1)<=Q(0);

Q(0)<=Q(3);

elsif (S="10") then

Q<=D;

elsif (S="11") then

Q(3)<=Q(0);

Q(2)<=Q(3);

Q(1)<=Q(2);

Q(0)<=Q(1);

end if;

end if;

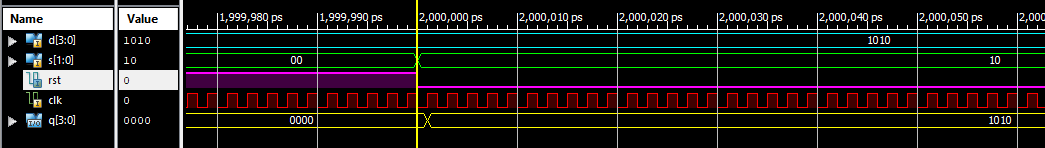
end process;

end Behavioral;

Imagen que contiene Tabla

Descripción generada automáticamente

**Cronograma:**



Interfaz de usuario gráfica

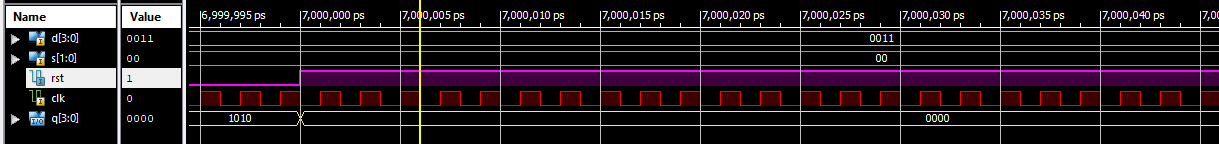
Descripción generada automáticamente

Interfaz de usuario gráfica

Descripción generada automáticamente

Interfaz de usuario gráfica

Descripción generada automáticamente con confianza media



1. Diseñe y programe un contador de 0 a 59, ascendente (1) y descendente (0). El circuito debe contar con una señal de reset activo en bajo, que coloca las salidas Q en estado bajo.

Gráfico, Esquemático, Gráfico de cajas y bigotes

Descripción generada automáticamente

**Codigo 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 ( RST : in STD\_LOGIC;

U\_D : in STD\_LOGIC;

CLK : in STD\_LOGIC;

Q : inout STD\_LOGIC\_VECTOR (5 downto 0));

end Contador;

architecture Behavioral of Contador is

begin

process (CLK,RST,U\_D)

begin

if(CLK'event and CLK='1') then

if (RST='0') then

Q<="000000";

else

if (U\_D='1') then

Q<=Q+1;

if (Q="111011") then --59

Q<="000000";--0

end if;

else

Q<=Q-1;

if (Q="000000") then --0

Q<="111011";--59

end if;

end if;

end if;

end if;

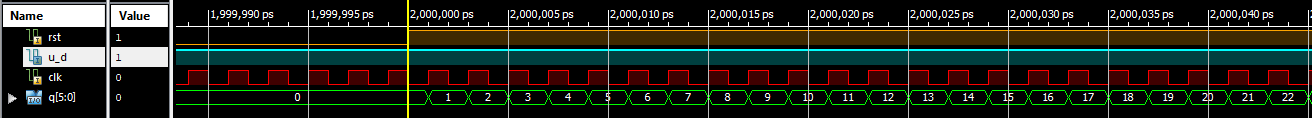
end process;

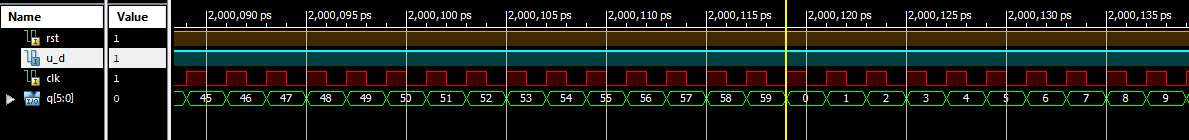
end Behavioral;

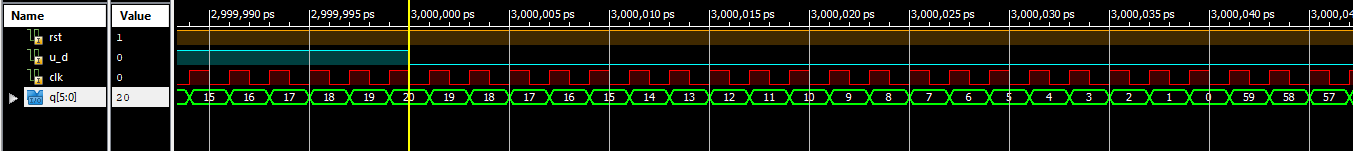
Texto

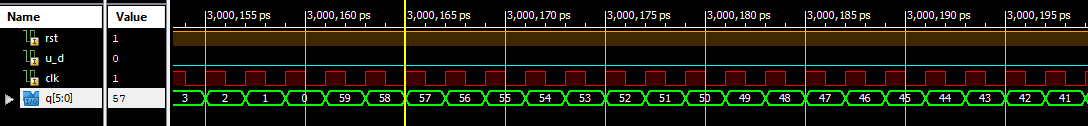
Descripción generada automáticamente con confianza media

**Cronograma:**









Pantalla de un video juego

Descripción generada automáticamente con confianza media

1. El siguiente circuito muestra el control de encendido de un motor a pasos en secuencia de medio tiempo.

* Tiene una señal de reset que permite restablecer la secuencia de conteo inicial.
* Independiente a su control de fases, cuenta con señales de monitoreo que indican en que paso se encuentra el proceso.

Imagen que contiene Tabla

Descripción generada automáticamenteDiagrama

Descripción generada automáticamente

**Codigo VHDL:**

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

use IEEE.STD\_LOGIC\_ARITH.ALL;

use IEEE.STD\_LOGIC\_UNSIGNED.ALL;

entity Motor is

Port ( RST : in STD\_LOGIC;

CLK : in STD\_LOGIC;

D : inout STD\_LOGIC\_VECTOR (3 downto 0);

DIS : out STD\_LOGIC\_VECTOR (6 downto 0);

Q : inout STD\_LOGIC\_VECTOR (1 downto 0);

F : out STD\_LOGIC\_VECTOR (3 downto 0));

end Motor;

architecture Behavioral of Motor is

begin

cont: process (CLK, RST, Q)--circuito secuencial

begin

if (RST='1') then

Q<="00";

elsif(CLK'event and CLK='1') then

Q<=Q+1;

end if;

end process cont;

deco: process (Q)--circuito combinacional

begin

case Q is

when "00" => D <= "1000";

when "01" => D <= "0100";

when "10" => D <= "0010";

when others => D <= "0001";

end case;

end process deco;

fase: process (D)--circuito combinacional FASES

begin

case D is

when "1000" => F <= "1000";

when "0100" => F <= "1100";

when "0010" => F <= "0100";

when "0001" => F <= "0110";

when others => F <= "0000";

end case;

end process fase;

disp: process (D)--circuito combinacional DISPLAY

begin

case D is-----------------------------abcdefg

when "1000" => DIS <= "1111111";--8

when "0100" => DIS <= "1001110";--12-c

when "0010" => DIS <= "0110011";--4

when "0001" => DIS <= "1011111";--6

when others => DIS <= "0000000";--nada

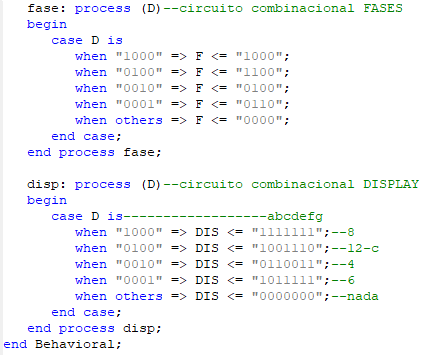
end case;

end process disp;

end Behavioral;

Texto

Descripción generada automáticamente



**Cronograma:**

