



Universidad Nacional Autónoma de
México
Facultad de Ingeniería



PROYECTO 4:

“Puerta automática para personas”

Diseño Digital VLSI Grupo: 05

M.I. ELIZABETH FONSECA CHAVEZ

Hernández Jaimes Rogelio Yael

Fecha de entrega: 11/Diciembre/2022

INTRODUCCIÓN

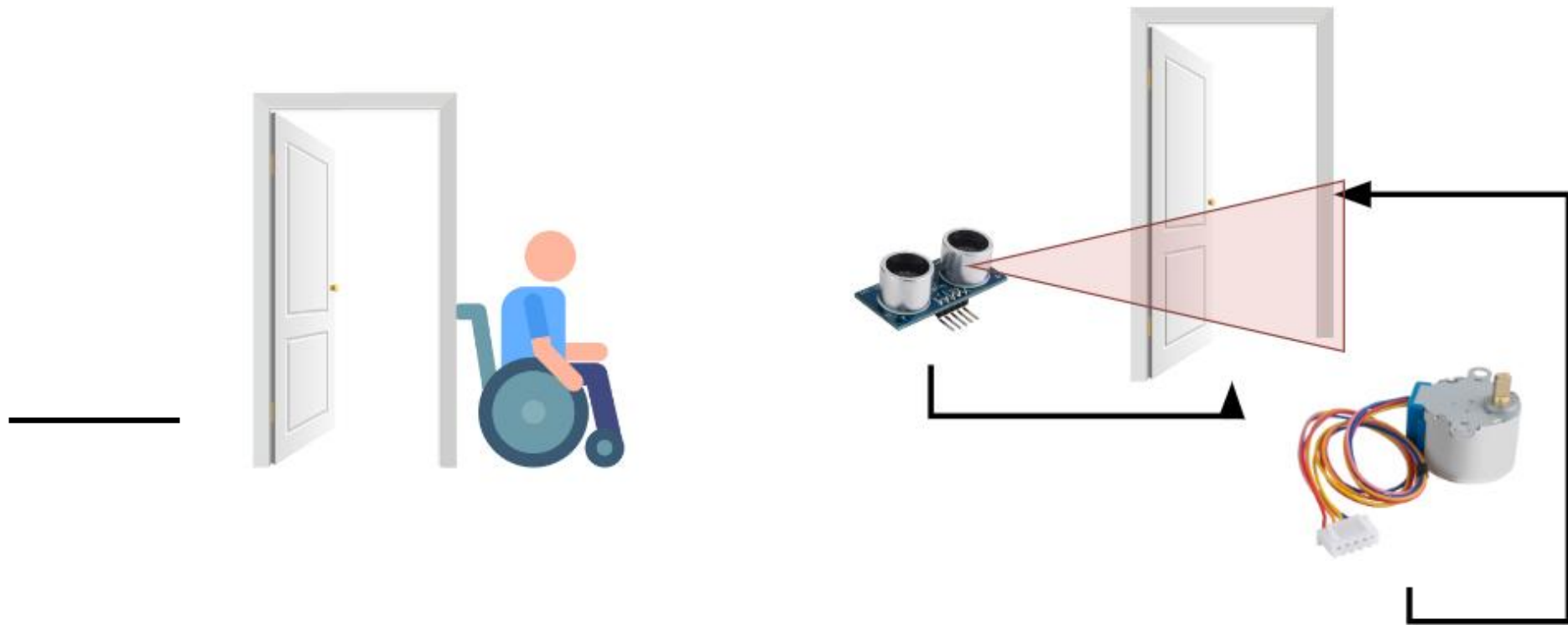
Desde hace siglos, las puertas existen. Regularn el paso de las personas, abriendo y cerrándose según se necesite. Desde hace algunos años, las puertas automáticas existen, puertas que abren y cierran por sí solas. Evitan así la necesidad de las personas de ejecutar la apertura y cierre manual.

Esta idea, permite a personas con capacidades diferentes y personas adultas mayores un paso menos complicado por las puertas, pues evitan la necesidad de abrir la puerta que en ocasiones resulta complicado.

OBJETIVO

Desarrollar un modelo de puerta automática que realice su apertura y cierre de forma independiente, pensando que las personas pueden demorar más o menos tiempo en cruzar la puerta.

MODELO





PROGRAMAS DE USO GENERAL

- Divisor de frecuencia: divf
- Regulador de ancho de pulso (PWM)

MOTOR A PASOS

```
1  library IEEE;
2  use ieee.std_logic_arith.all;
3  use IEEE.std_logic_1164.all;
4
5  --Alfaro Fernández, Azul
6  --Hernández Jaimes, Rogelio Yael
7  --Núñez Luna, Aranza Abril
8
9
10 entity MotorPasos is
11 port(clk:in std_logic;
12       sensor_disp: out std_logic;
13       sensor_eco: in std_logic;
14       rst:in std_logic;
15       mot:out std_logic_vector(3 downto 0));
16 end MotorPasos;
17
18 architecture arqMotorPasos of MotorPasos is
19 signal clk1: std_logic;
20 signal clk1b : std_logic;
21 signal clk1c : std_logic;
22 signal state: std_logic_vector(1 downto 0);
23 signal duty : integer range 0 to 1000 := 100;
24 signal direc:std_logic:='0';
25 signal pause: std_logic:='1';
26 signal inicio: std_logic;
27 begin
28 u1: entity work.divf(arqdivf) generic map(200) port map(clk,clk1);
29 u2: entity work.divf(arqdivf) generic map(25000) port map(clk,clk1c);
30 u3: entity work.senal(arqsenal) port map(clk1,duty,clk1b);
31 u4: entity work.sensor(arqsensor) port map(clk,sensor_disp,sensor_eco,inicio);
32 u5: entity work.secuencia(arqsec) port map(inicio,clk1c,direc,pause);
33 u6: entity work.estados(arqestados) port map(clk1b,pause,rst,direc,state);
34 u7: entity work.romlod(arqromlod) port map(state,pause,mot);
35 end architecture;
```

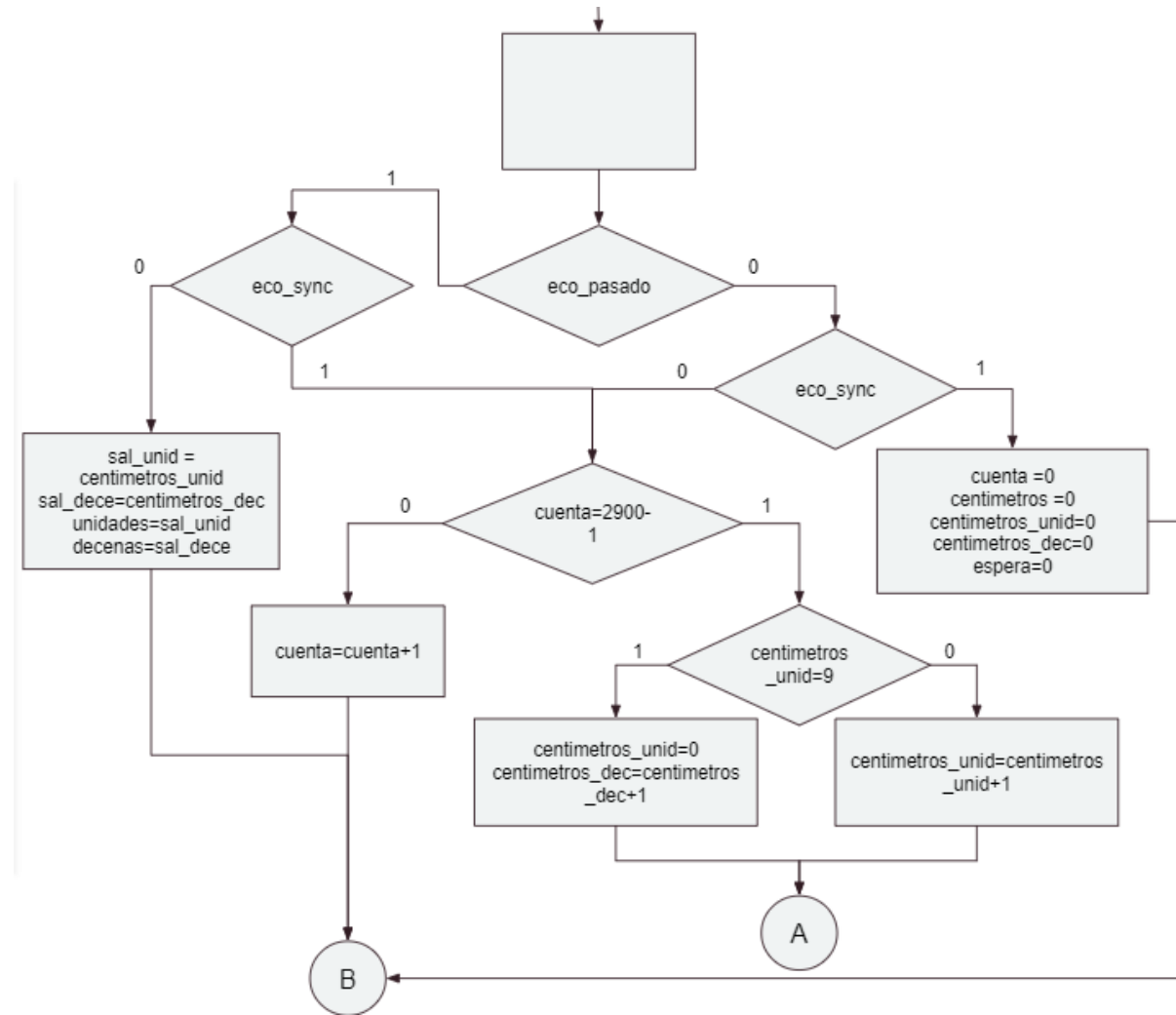
TRIGGER.VHDL

```
1  library ieee;
2  use IEEE.STD_LOGIC_1164.ALL;
3  USE IEEE.std_logic_Arith.all;
4  USE IEEE.std_logic_unsigned.all;
5
6  --Hernández Jaimes Rogelio Yael
7
8  ENTITY triger is
9  port( clk: in std_logic;
10       trigger: out std_logic;
11       espera_aux: in std_logic);
12  end entity;
13
14  architecture arqtriger of triger is
15  | signal cuenta: unsigned (16 downto 0) := (others => '0');
16  | signal espera: std_logic := '0';
17  | begin
18  |   process(clk)
19  |   begin
20  |       if rising_Edge(clk) then
21  |           espera<=espera_aux;
22  |           if espera = '0' then
23  |               if cuenta = 500 then
24  |                   trigger<= '0';
25  |                   espera <= '1';
26  |                   cuenta <= ( others => '0');
27  |               else
28  |                   trigger <= '1';
29  |                   cuenta <= cuenta+1;
30  |               end if;
31  |           end if;
32  |       end if;
33  |   end process;
34  end architecture;
```

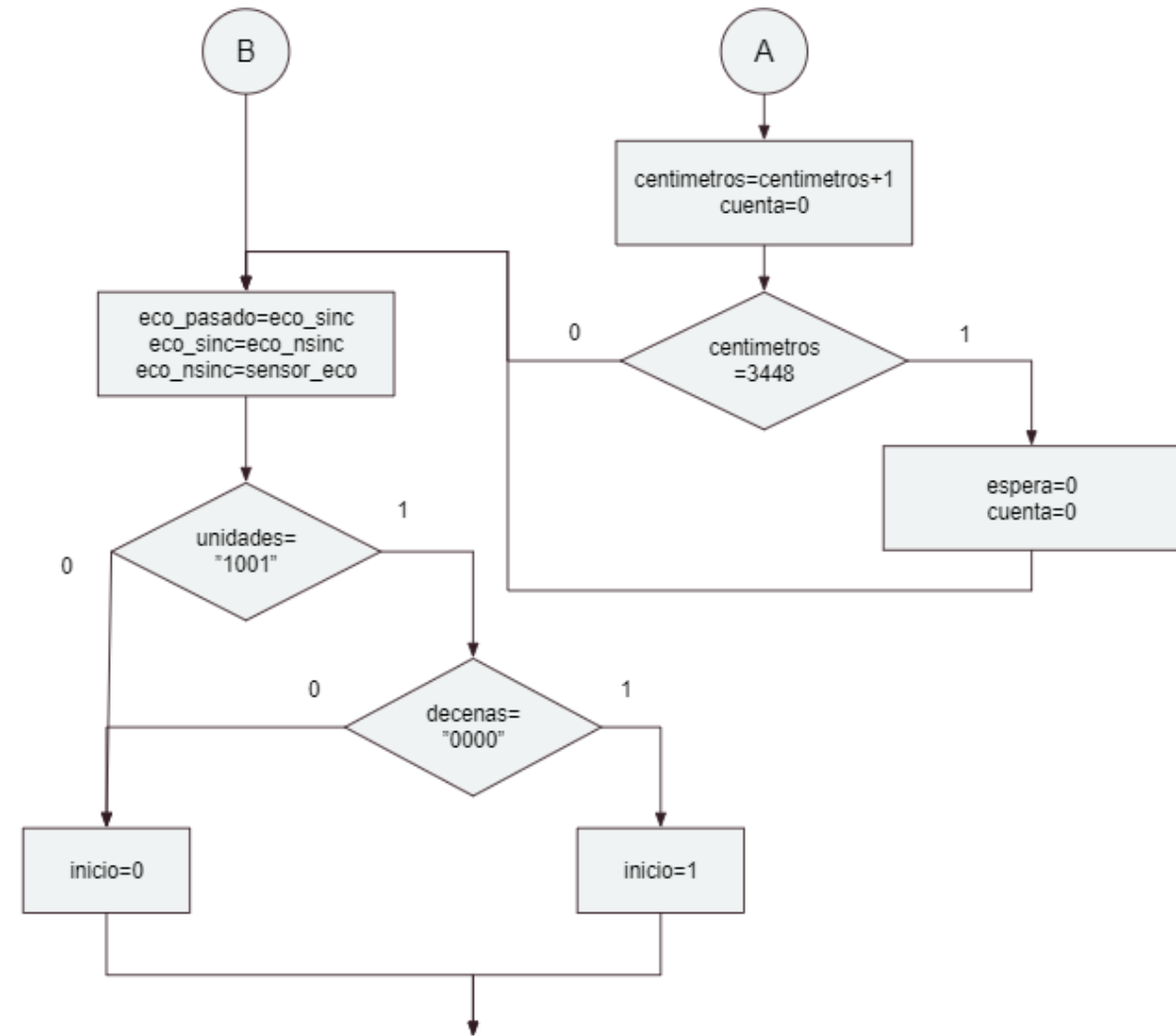
CONTADOR.VHDL

```
1 library ieee;
2 use IEEE.STD_LOGIC_1164.ALL;
3 USE IEEE.std_logic_arith.all;
4 USE IEEE.std_logic_unsigned.all;
5
6 --Hernández Jaimes Rogelio Yael
7
8 ENTITY contador is
9 port( clk: in std_logic;
10      sensor_eco: in std_logic;
11      espera: out std_logic;
12      inicio: out std_logic);
13 end entity;
14
15 ARCHITECTURE arqcont OF contador IS
16     signal eco_pasado: std_logic := '0';
17     signal eco_sinc: std_logic := '0';
18     signal eco_nsinc: std_logic := '0';
19     signal centimetros: unsigned (15 downto 0) := (others => '0');
20     signal centimetros_unid: unsigned (3 downto 0) := (others => '0');
21     signal centimetros_dec: unsigned (3 downto 0) := (others => '0');
22     signal sal_unid: unsigned (3 downto 0) := (others => '0');
23     signal sal_dece: unsigned (3 downto 0) := (others => '0');
24     signal cuenta: unsigned (16 downto 0) := (others => '0');
25     signal unidades: std_logic_vector ( 3 downto 0);
26     signal decenas: std_logic_vector ( 3 downto 0);
27 BEGIN
28     process(clk)
29     begin
30         if rising_edge(clk) then
31             if eco_pasado = '0' and eco_sinc = '1' then --Calcula la distancia
32                 cuenta <= ( others => '0' );
33                 centimetros <= ( others => '0' );
34                 centimetros_unid <= ( others => '0' );
35                 centimetros_dec <= ( others => '0' );
36                 espera <= '0';
37             elsif eco_pasado = '1' and eco_sinc = '0' then --Detecta el objeto
38                 sal_unid <= centimetros_unid;
39                 sal_dece <= centimetros_dec;
40
41                 unidades <= conv_STD_LOGIC_VECTOR(sal_unid,4); --Se guarda la unidad de centimetros en 4 bits
42                 decenas <= conv_STD_LOGIC_VECTOR(sal_dece,4); --Se guarda la unidad de centimetros en 4 bits
43             elsif cuenta = 2900-1 then
44                 if centimetros_unid = 9 then -- muestra del display del 9 al 19 o 2
45                     centimetros_unid <= ( others => '0' );
46                     centimetros_dec <= centimetros_dec+1; --Aumenta 1 en decenas
47                 else
48                     centimetros_unid <= centimetros_unid+1; --Aumenta 1 en unidades
49                 end if;
50                 centimetros <= centimetros+1; --Suma de los centimetros
51                 cuenta <= ( others => '0' );
52                 if centimetros = 3448 then
53                     espera <= '0';
54                     cuenta <= ( others => '0' );
55                 end if;
56             else
57                 cuenta <= cuenta+1;
58             end if;
59             eco_pasado <= eco_sinc;
60             eco_sinc <= eco_nsinc;
61             eco_nsinc <= sensor_eco;
62         end if;
63
64         if ((unidades>="0010" and unidades<="0110") and decenas = "0000") then
65             inicio <= '1'; -- a 5 cm de distancia
66         else
67             inicio <= '0';
68         end if;
69     end process;
70 END ARCHITECTURE;
```


CONTADOR.VHDL



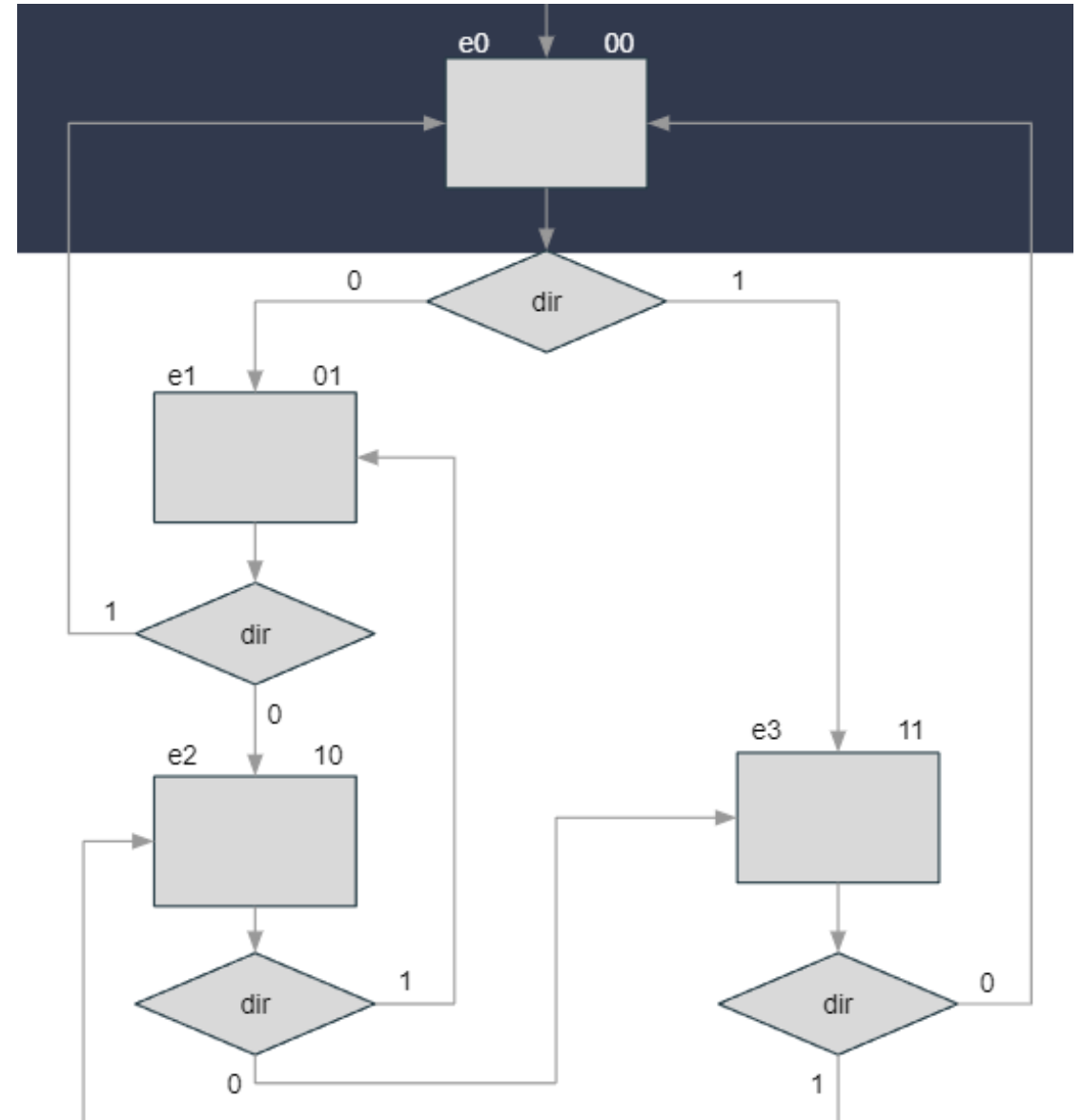
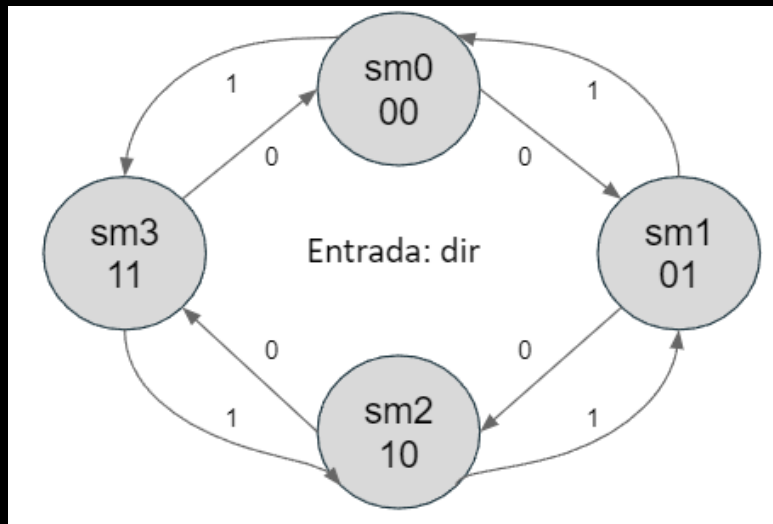
CONTADOR.VHDL



ESTADOS.VHDL

```
1 library IEEE;
2 use IEEE.std_logic_1164.all;
3 use IEEE.std_logic_arith.all;
4 use IEEE.std_logic_unsigned.all;
5
6 --Alfaro Fernández, Azul
7 --Hernández Jaimes, Rogelio Yael
8 --Núñez Luna, Aranza Abril
9
10 entity estados is
11 port (clk: in std_logic;
12       UD: in std_logic; --paro
13       rst: in std_logic;
14       dir: in std_logic; --0: sentido antihorario, 1: sentido horario
15       state: out std_logic_vector(1 downto 0));
16 end estados;
17 architecture arqestados of estados is
18     subtype estado is std_logic_vector(1 downto 0);
19     constant sm0:estado:="00";
20     constant sm1:estado:="01";
21     constant sm2:estado:="10";
22     constant sm3:estado:="11";
23     signal pres_S,next_S:estado;
24 begin
25     process(clk)
26     begin
27         if rising_edge(clk) then
28             if UD='0' then
29                 if rst='1' then
30                     pres_S<=sm0;
31                 else
32                     pres_S<=next_S;
33                 end if;
34             end if;
35             state<=pres_S;
36         end process;
37
38     process(pres_S, dir)
39     begin
40         case(pres_S) is
41             when sm0 =>
42                 if dir='0' then
43                     next_S<=sm1;
44                 else
45                     next_S<=sm3;
46                 end if;
47             when sm1 =>
48                 if dir='0' then
49                     next_S<=sm2;
50                 else
51                     next_S<=sm0;
52                 end if;
53             when sm2 =>
54                 if dir='0' then
55                     next_S<=sm3;
56                 else
57                     next_S<=sm1;
58                 end if;
59             when others => --Estado 3
60                 if dir='0' then
61                     next_S<=sm0;
62                 else
63                     next_S<=sm2;
64                 end if;
65         end case;
66     end process;
67 end architecture;
```

ESTADOS.VHDL

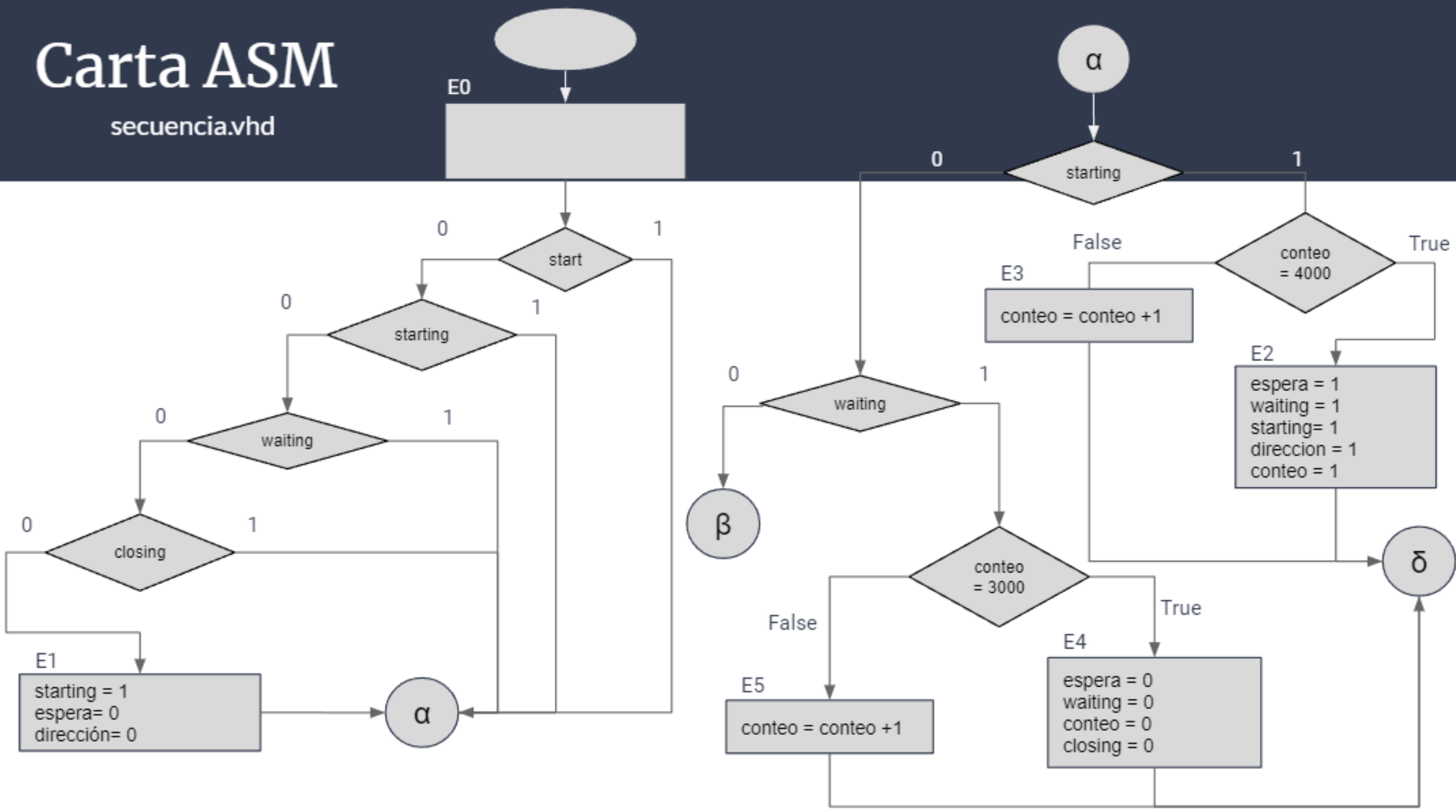


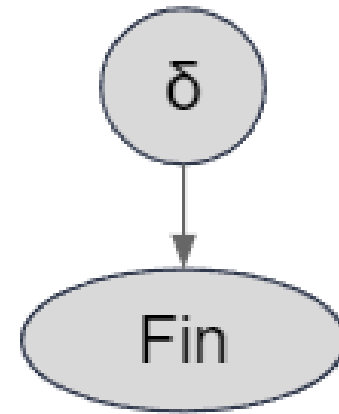
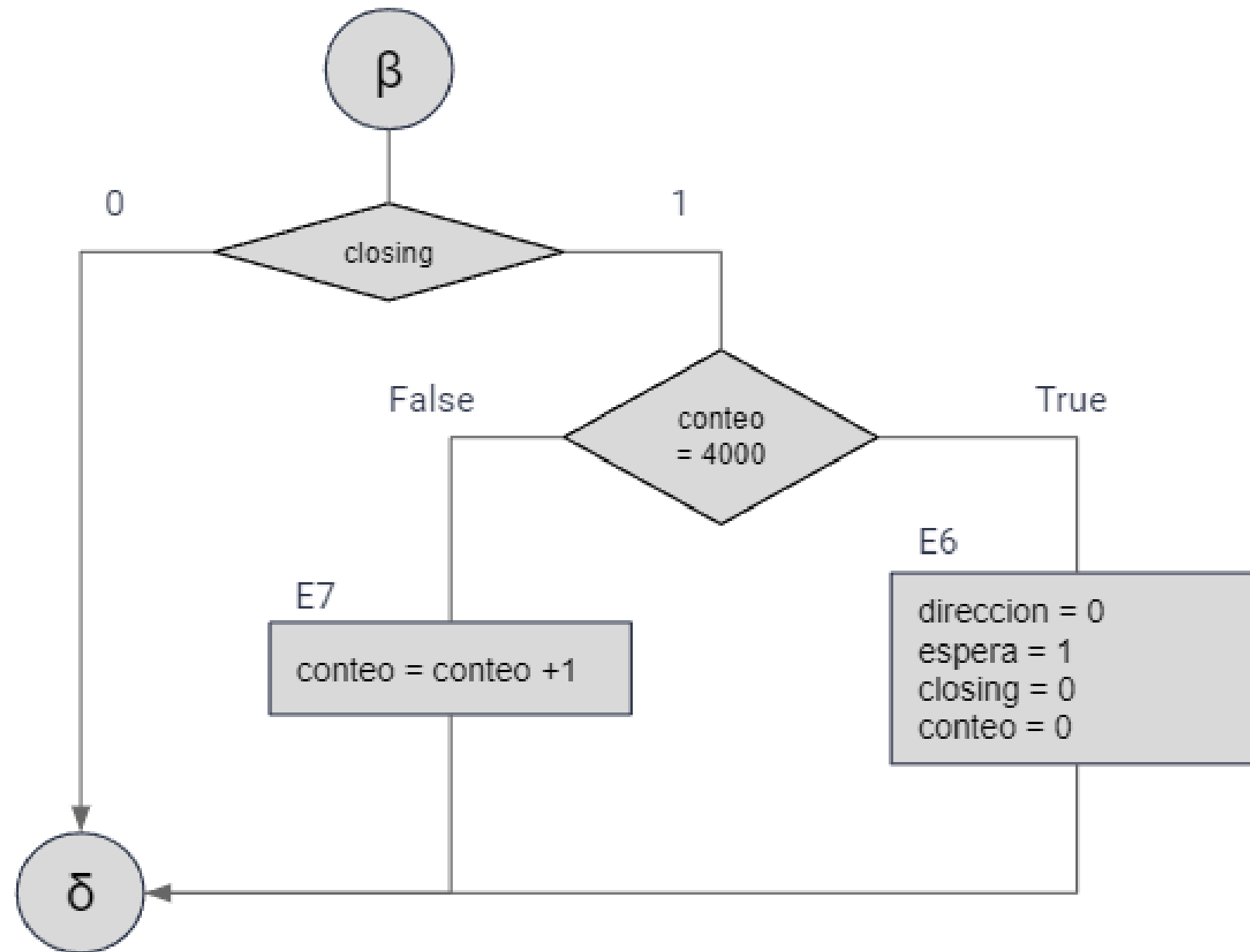
SECUENCIA.VHDL

```
1  library ieee;
2  use ieee.std_logic_1164.all;
3
4  --Alfaro Fernández, Azul
5  --Hernández Jaimes, Rogelio Yael
6  --Núñez Luna, Aranza Abril
7
8  entity secuencia is
9  port ( start: in std_logic;
10        clk: in std_logic;
11        direccion: out std_logic:= '0';
12        espera: out std_logic);
13  end entity;
14
15  architecture arqsec of secuencia is
16  signal conteo: integer range 0 to 10000:=0;
17  signal starting: std_logic:= '0'; --puerta abriendose
18  signal waiting: std_logic:= '0'; --puerta abierta
19  signal closing: std_logic:= '0'; -- puerta cerrandose
20  begin
21      process(clk)
22      begin
23          if rising_edge(clk) then
24              if start= '1' and starting= '0' and waiting= '0' and closing= '0' then
25                  starting<= '1';
26                  espera<= '0';
27                  direccion<= '0';
28              end if;
29              --abriendo puerta
30              if starting= '1' then
31                  if conteo=4800 then
32                      espera<= '1';
33                      waiting<= '1';
34                      starting<= '0';
35                      direccion<= '1';
36                      conteo<= 0;
37                  else
38                      conteo<= conteo+1;
39                  end if;
40              elsif waiting= '1' and start= '0' then
41                  --puerta abierta
42                  if conteo=4500 then
43                      espera<= '0';
44                      conteo<= 0;
45                      waiting<= '0';
46                      closing<= '1';
47                  else
48                      conteo<= conteo+1;
49                  end if;
50              elsif closing= '1' then
51                  --cerrando puerta
52                  if conteo=4800 then
53                      direccion<= '0';
54                      espera<= '1';
55                      closing<= '0';
56                      conteo<= 0;
57                  else
58                      conteo<= conteo+1;
59                  end if;
60              end if;
61          end if;
62      end process;
63  end architecture;
```

Carta ASM

secuencia.vhd





ROMLOD.VHDL

```
1  library ieee;
2  use ieee.std_logic_1164.all;
3  use ieee.std_logic_UNSIGNED.all;
4  --Alfaro Fernández, Azul
5  --Hernández Jaimes, Rogelio Yael
6  --Núñez Luna, Aranza Abril
7
8  entity romlod is
9  port(
10     bus_dir: in std_logic_vector (1 downto 0);
11     cs: in std_logic;
12     bus_datos: out std_logic_vector (3 downto 0));
13  end romlod;
14
15  architecture arqromlod of romlod is
16     --formato de orden de bits: Direccion,
17     constant L1: std_logic_vector (3 downto 0):="1100";
18     constant L2: std_logic_vector (3 downto 0):="1001";
19     constant L3: std_logic_vector (3 downto 0):="0011";
20     constant L4: std_logic_vector (3 downto 0):="0110";
21     type memoria is array (3 downto 0) of std_logic_vector (3 downto 0);
22     constant mem_rom: memoria:=(L1,L2,L3,L4);
23     signal dato: std_logic_vector (3 downto 0);
24  begin
25     prom: process(bus_dir)
26     begin
27         dato <= mem_rom(conv_integer(bus_dir));
28     end process prom;
29
30     pbuf: process (dato,cs)
31     begin
32         if(cs='0') then
33             bus_datos <= dato;
34         else
35             bus_datos <= (others => '0');
36         end if;
37     end process pbuf;
38  end arqromlod;
```


ROMLOD.VHDL

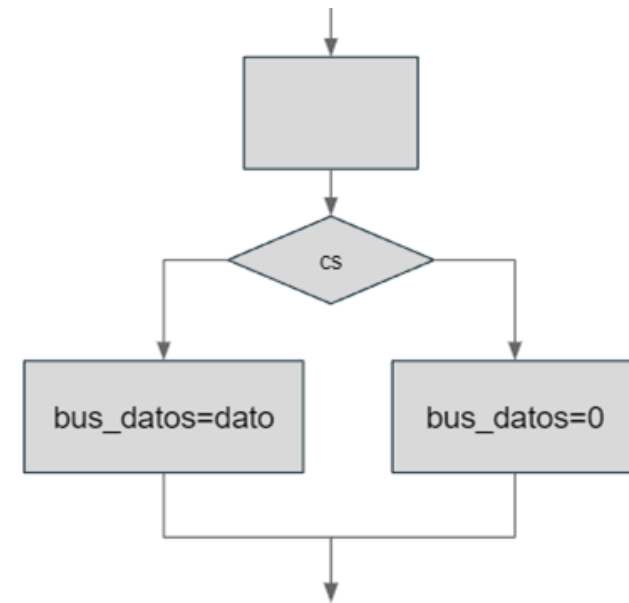
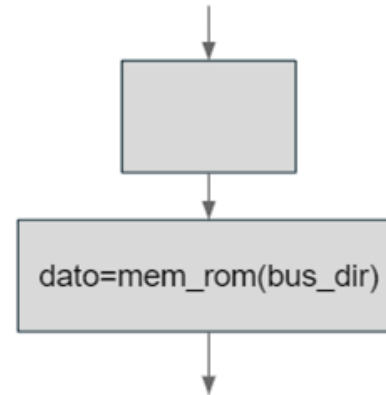
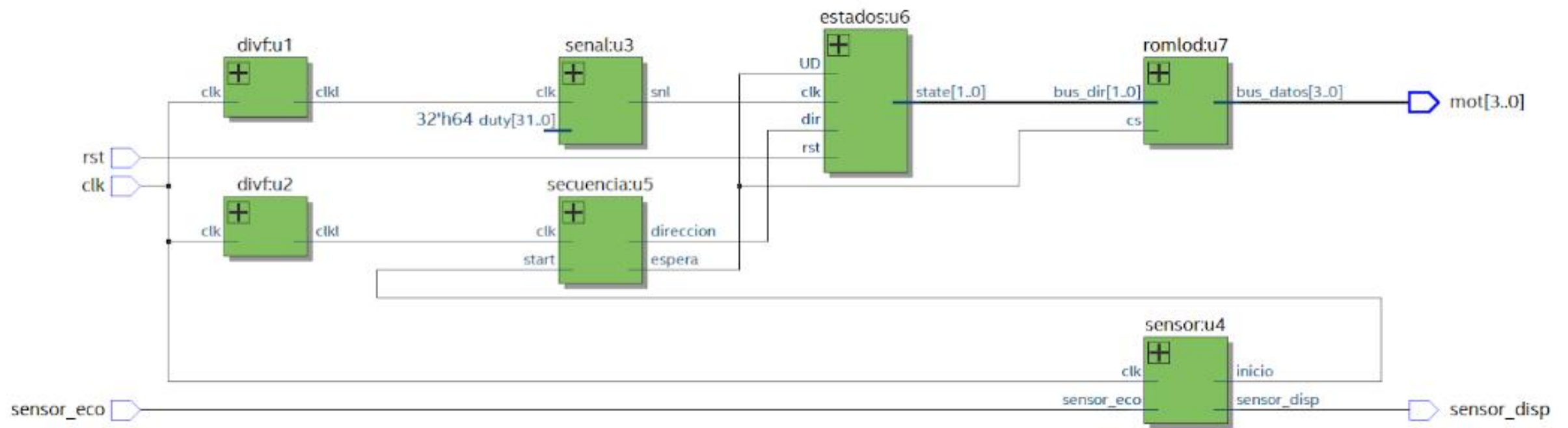


DIAGRAMA RTL



FUNCIONAMIENTO



CONCLUSIÓN

Los conocimientos adquiridos a lo largo del curso se pueden utilizar en aplicaciones de la vida real con relativa facilidad. Esto a través de la variedad de componentes que la tarjeta FPGA nos permite utilizar.

Asimismo, es importante comprender cada parte necesaria para la construcción de un sistema, pues se requiere armar cada una de las partes una por una y considerar a profundidad el funcionamiento que se busca conseguir.

REFERENCIAS

- Fonseca, E. *Prácticas de diseño digital VLSI 2022 para tarjetas FPGA Altera-Intel (DE10-LITE)*. https://drive.google.com/file/d/1npB8Hr-OpyAwio8iXaHZ_ox8dPFAgdig/view
- Profesora Elizabeth Fonseca. (2020, 25 marzo). ROM VHDL [Vídeo]. YouTube. <https://www.youtube.com/watch?v=X-cDHDVHWL8>