Práctica 13: Decodificadores

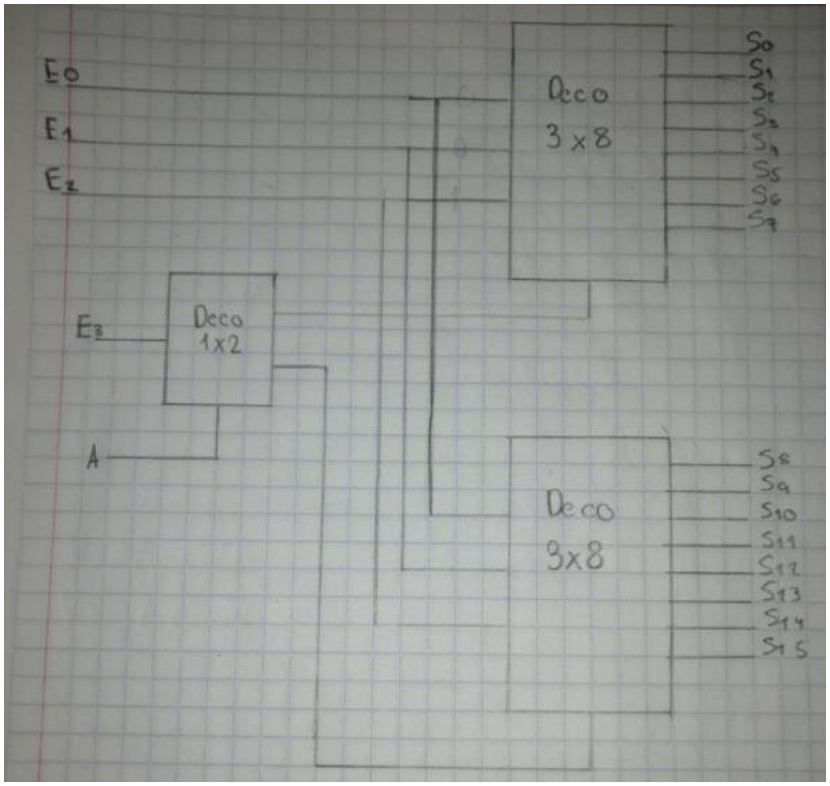
Nombre: Colín Ramiro Joel No. de lista: 3

Ejercicio 1.-

Implemente teóricamente un decodificador 4x16 mediante el uso de decodificadores de 3x8

E3	***E2***	***E1***	***E0***	***A***	***S15***	***S14***	***S13***	***S12***	***S11***	***S10***	***S9***	***S8***	***S7***	***S6***	***S5***	***S4***	***S3***	***S2***	***S1***	***S0***
0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
0	1	0	1	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
0	1	1	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
0	1	1	1	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
1	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
1	0	0	1	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
1	0	1	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
1	0	1	1	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
1	1	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
1	1	0	1	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
1	1	1	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
X	X	X	X	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

E2	*E1*	*E0*	*A*	*S7*	*S6*	*S5*	*S4*	*S3*	*S2*	*S1*	*S0*
0	*0*	*0*	*1*	*0*	*0*	*0*	*0*	*0*	*0*	*0*	*1*
0	*0*	*1*	*1*	*0*	*0*	*0*	*0*	*0*	*0*	*1*	*0*
0	*1*	*0*	*1*	*0*	*0*	*0*	*0*	*0*	*1*	*0*	*0*
0	*1*	*1*	*1*	*0*	*0*	*0*	*0*	*1*	*0*	*0*	*0*
1	*0*	*0*	*1*	*0*	*0*	*0*	*1*	*0*	*0*	*0*	*0*
1	*0*	*1*	*1*	*0*	*0*	*1*	*0*	*0*	*0*	*0*	*0*
1	*1*	*0*	*1*	*0*	*1*	*0*	*0*	*0*	*0*	*0*	*0*
1	*1*	*1*	*1*	*1*	*0*	*0*	*0*	*0*	*0*	*0*	*0*
X	*X*	*X*	*0*	*0*	*0*	*0*	*0*	*0*	*0*	*0*	*0*



Ejercicio 2.-

Realizar la simulación para obtener los resultados de la siguiente tabla:

SAL(3)	SAL(2)	SAL(1)	SAL(0)	SAL2(3)	SAL2(2)	SAL2(1)	SAL2(0)
0	0	1	0	0	1	0	0
1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1
0	0	0	0	1	0	0	0

Código ejercicio 2.-

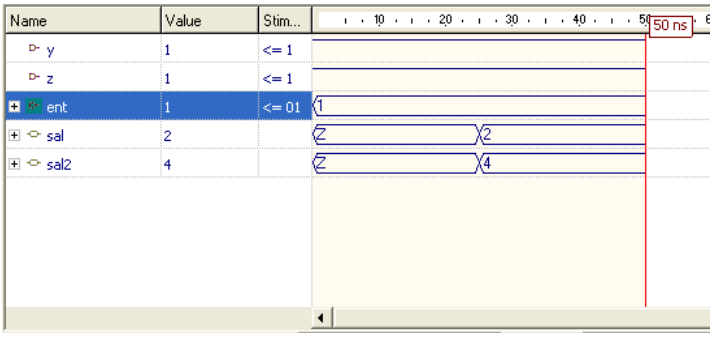
```
library ieee;
use ieee.std_logic_1164.all;
entity parte1 is
port(
	ent: in std_logic_vector(1 downto 0);
	Z,Y: in std_logic;
		sal,sal2:out std_logic_vector(3 downto 0)

);
end parte1;
architecture aparte1 of parte1 is
begin

	process(ent,Z)
		begin
			if(Z = '1')then
				if (ent = "00")then
					sal <= "0001";
				elsif (ent = "01")then
					sal <= "0010";
				elsif (ent = "10")then
					sal <= "0100";
				else
					sal <= "1000";
				end if;
		  else
					sal <= "0000";
			end if;
	end process;
	-----------------------------------------------------------------------------------------------------
	process(ent,Y)
	begin
	if (Y = '1')then
		case ent is
			when "00" => sal2 <= "0001";
			when "01" => sal2 <= "0100";
		  when "10" => sal2 <= "0010";
			when "11" => sal2 <= "1000";
			when others => sal2 <= "1XXX";
```

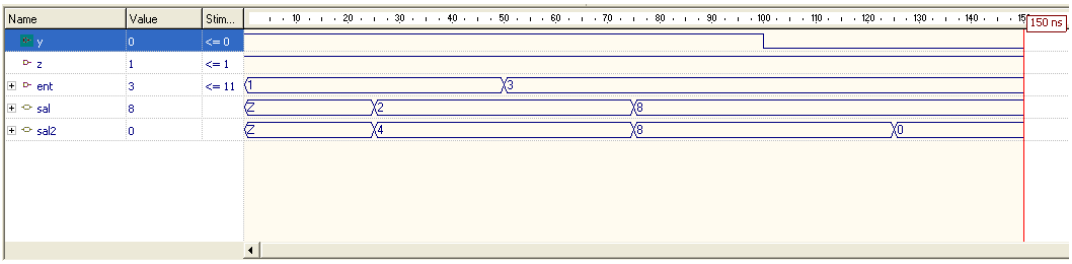
```
                end case;
        else
        sal2 <= "0000";
        end if;
        end process;
end aparte1;
```

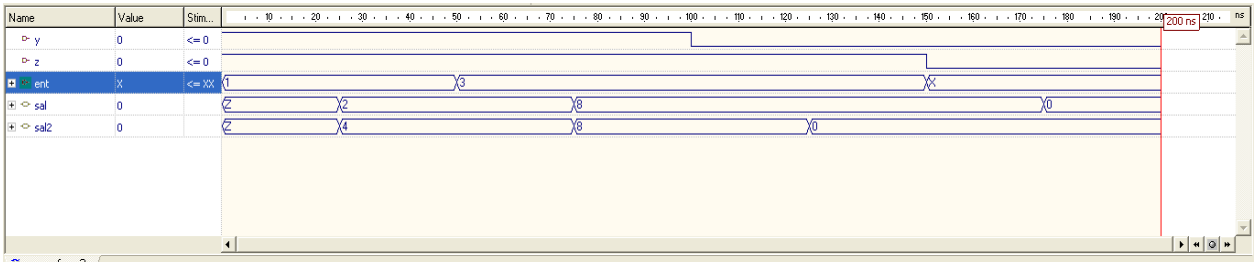
Ecuaciones ejercicio 2.-

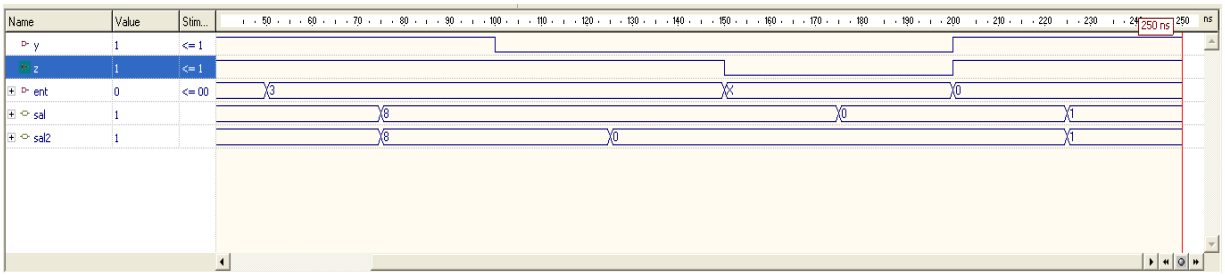
sal(0) =
/ent(0) * /ent(1) * z
sal(1) =
ent(0) * /ent(1) * z
sal(2) =
/ent(0) * ent(1) * z
sal(3) =
ent(0) * ent(1) * z
sal2(0) =
/ent(0) * /ent(1) * y
sal2(1) =
/ent(0) * ent(1) * y
sal2(2) =
ent(0) * /ent(1) * y
sal2(3) =
ent(0) * ent(1) * y

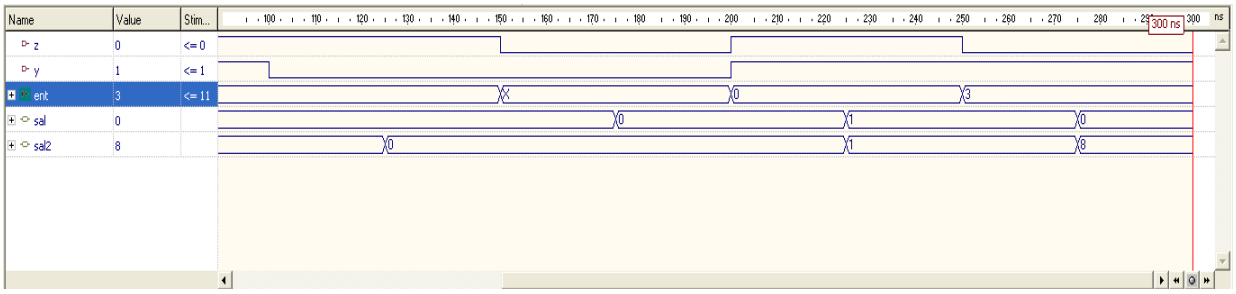
Simulación VHDL ejercicio 2.-



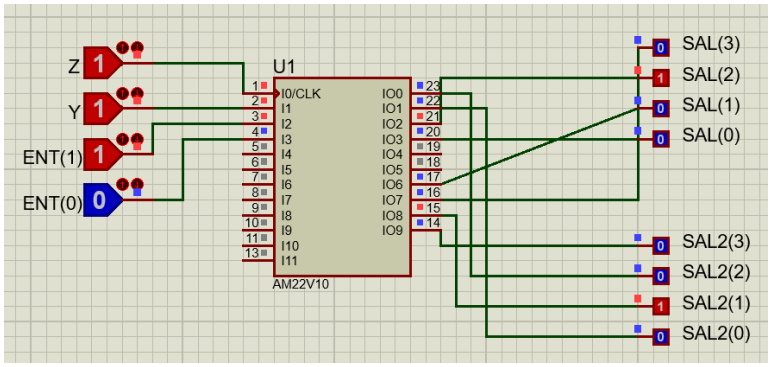


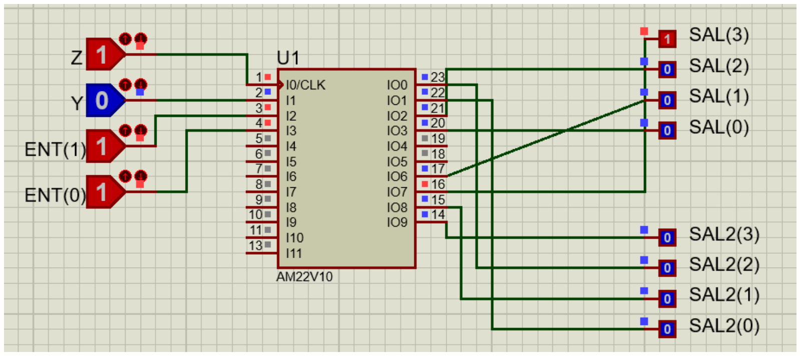


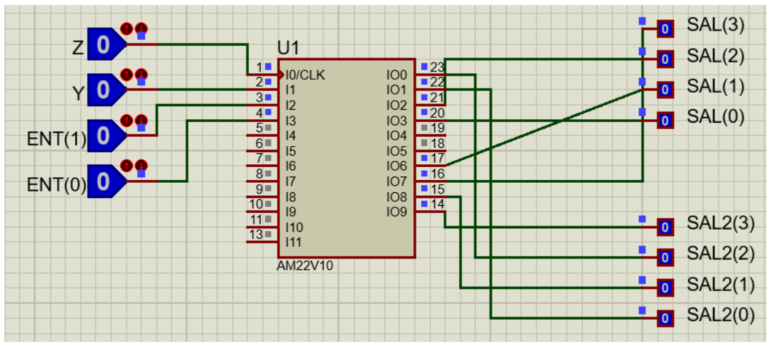


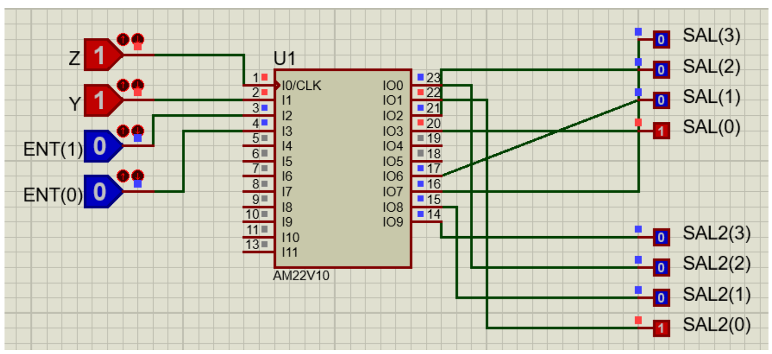


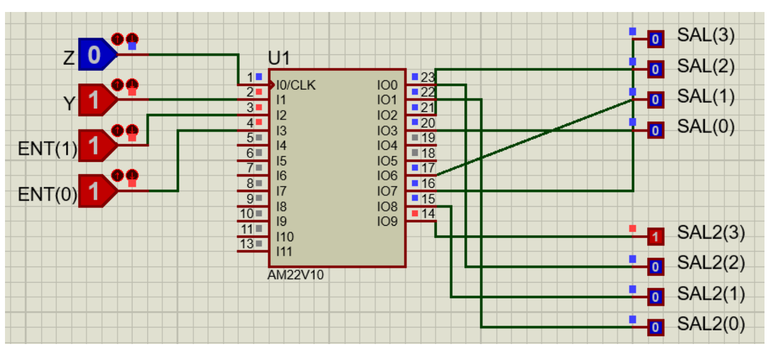
Simulación en Proteus ejercicio 2.-











Ejercicio 3: *Implementar en VHDL un decodificador 3x8 mediante decodificadores 2x4 con señal de habilitación, usando los niveles de descripción que no se usaron en el ejercicio 2.*

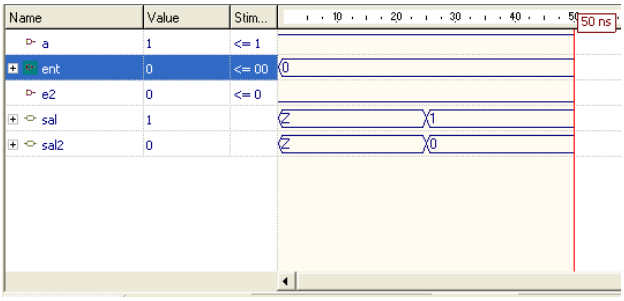
Código ejercicio 3.-

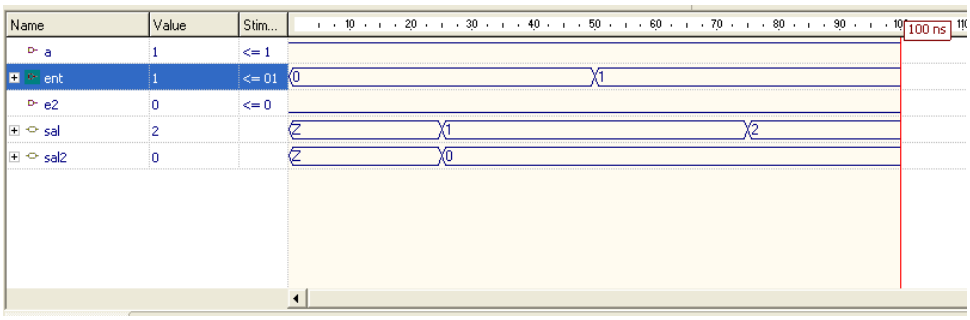
```
library ieee;
use ieee.std_logic_1164.all;
entity parte2 is
port(
        ent: in std_logic_vector(1 downto 0);
        A,E2:in std_logic;
        sal,sal2:out std_logic_vector(3 downto 0)

);
end parte2;
architecture aparte2 of parte2 is
signal C,D:std_logic;
signal sal3:std_logic_vector(3 downto 0);
begin


        C <= '1' when(E2 = '0')else
            '0';
                sal <=
                        "0001" when(ent = "00" AND C ='1')else
                        "0010" when(ent = "01" AND C ='1')else
                        "0100" when(ent = "10" AND C ='1')else
                        "1000" when(ent = "11" AND C ='1')else
                        "0000" when (C='0') else
                        "ZZZZ";
        ----------------------------------------------------------------------------------------------------

        D <= '1' when(E2 = '1')else
                    '0';
                with ent select sal3 <=
                                "0001" when "00",
                                "0010" when "01",
                                "0100" when "10",
                                "1000" when "11",
                                "0000" when others;

                        with D select sal2 <=
                                sal3 when '1',
                                "0000" when '0',
                                "XXXX" when others;
end aparte2;
```

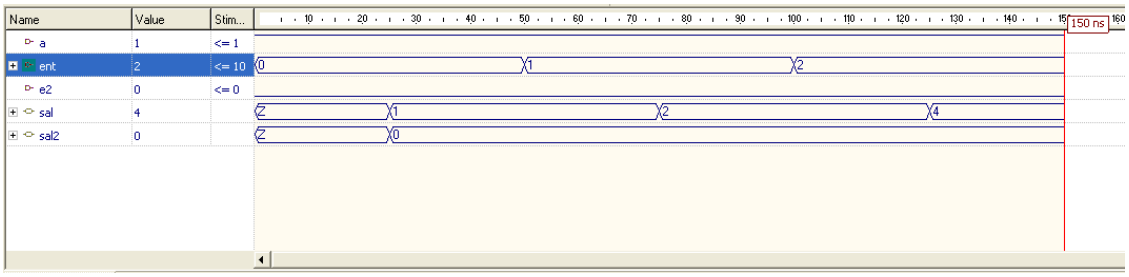
Ecuaciones ejercicio 3.-

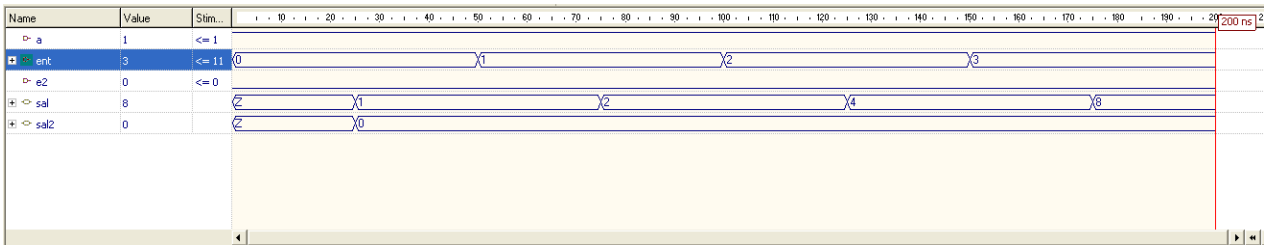
sal(0) =
/e2 * /ent(0) * /ent(1)
sal(1) =
/e2 * ent(0) * /ent(1)
sal(2) =
/e2 * /ent(0) * ent(1)
sal(3) =
/e2 * ent(0) * ent(1)
sal2(0) =
e2 * /ent(0) * /ent(1)
sal2(1) =
e2 * ent(0) * /ent(1)
sal2(2) =
e2 * /ent(0) * ent(1)
sal2(3) =
e2 * ent(0) * ent(1)

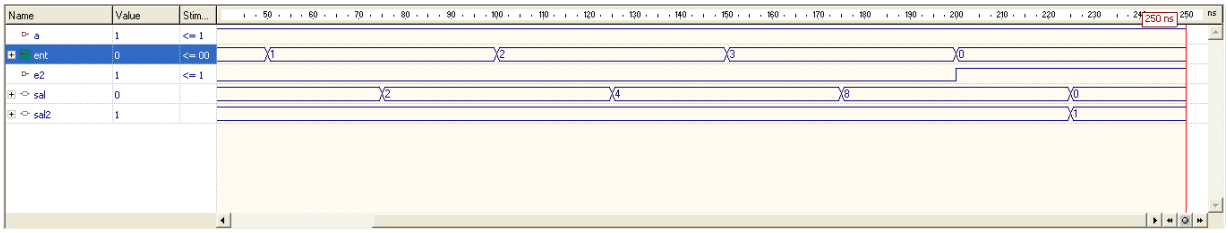
Simulación en VHDL ejercicio 3.-

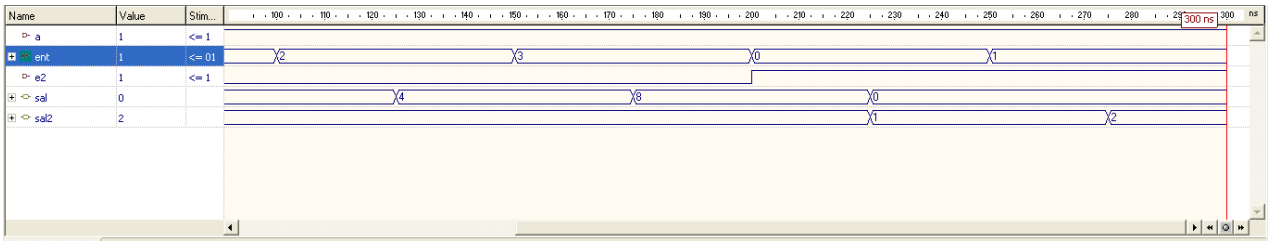


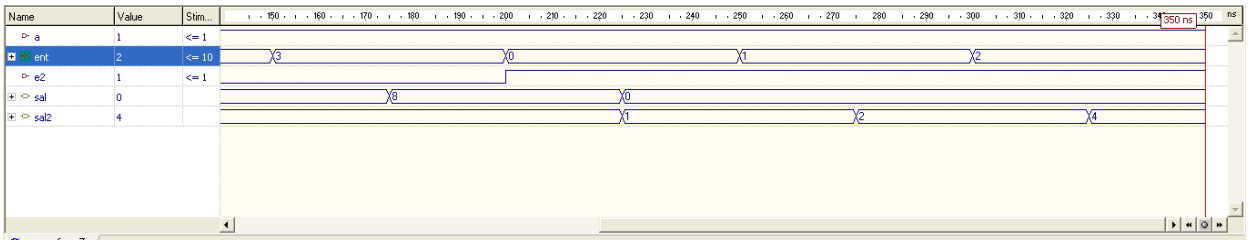


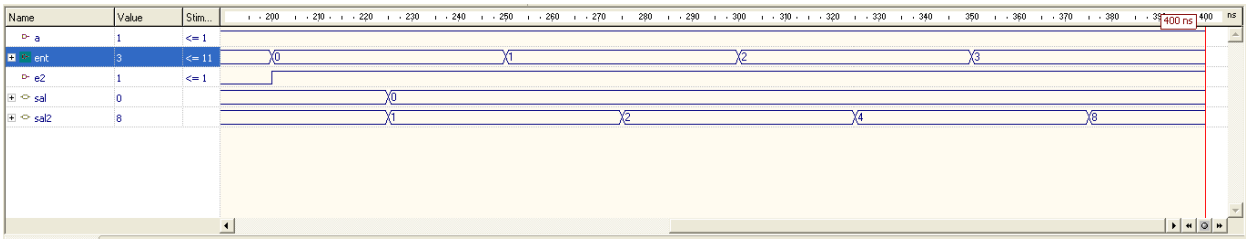

Name
Value
Stim...
a
1
<= 1
ent
2
<= 10
e2
0
<= 0
sal
4
sal2
0
150 ns

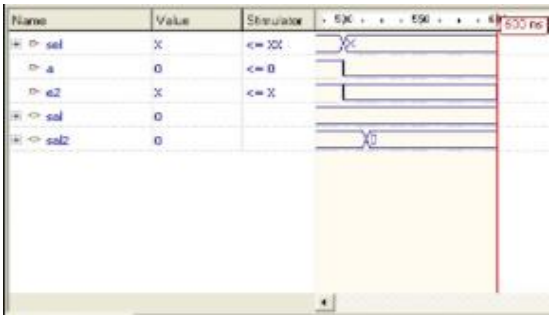

Name
Value
Stim...
a
1
<= 1
ent
3
<= 11
e2
0
<= 0
sal
8
sal2
0
200 ns


Name
Value
Stim...
a
1
<= 1
ent
0
<= 00
e2
1
<= 1
sal
0
sal2
1
250 ns

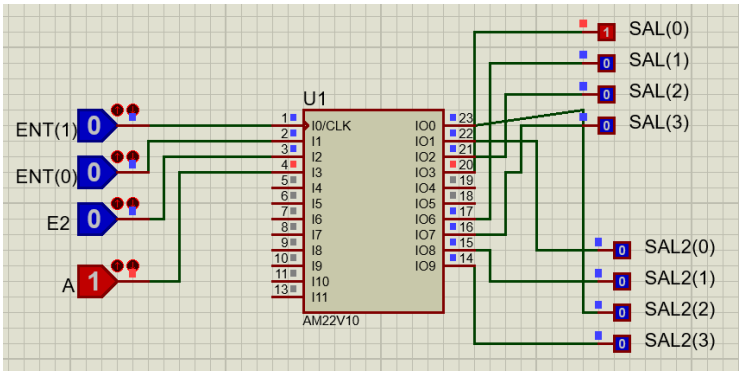

Name
Value
Stim...
a
1
<= 1
ent
1
<= 01
e2
1
<= 1
sal
0
sal2
2
300 ns

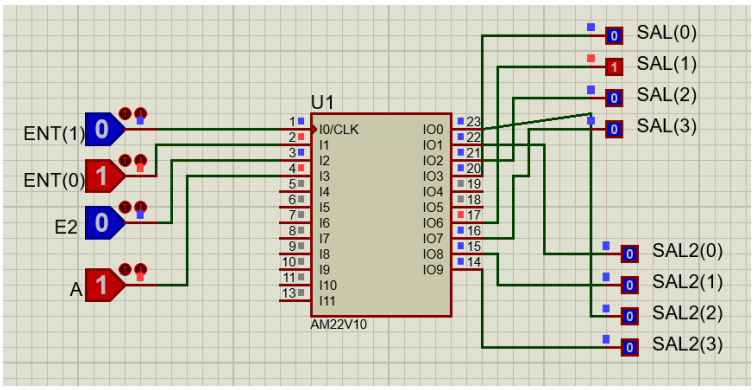

Name
Value
Stim...
a
1
<= 1
ent
2
<= 10
e2
1
<= 1
sal
0
sal2
4
350 ns

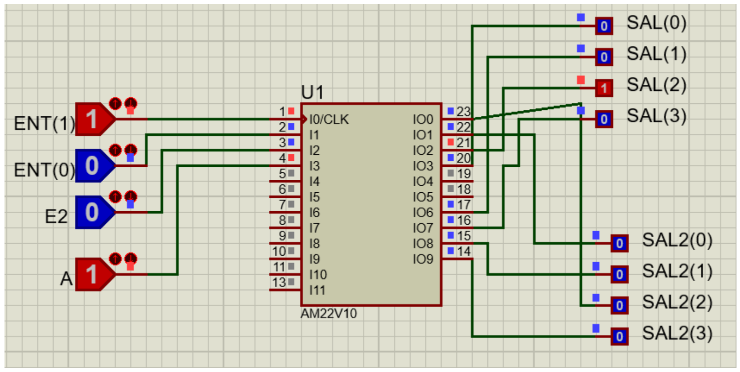


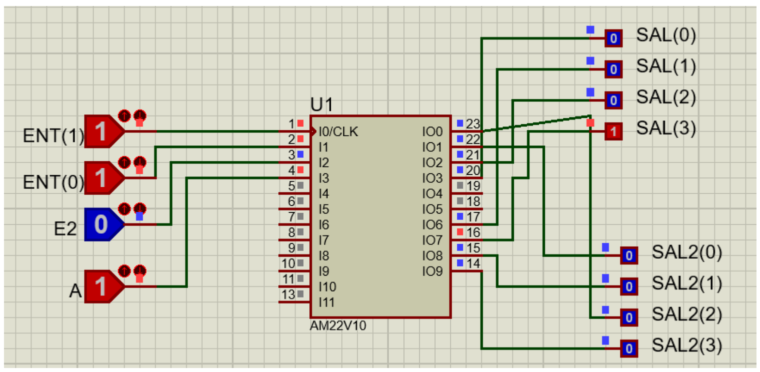


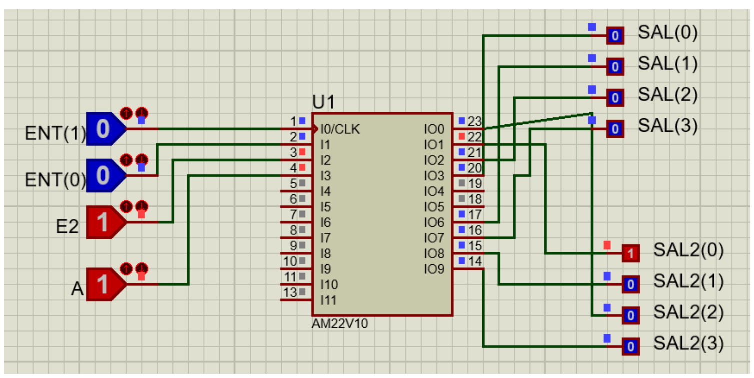
Simulación en Proteus ejercicio 3.-

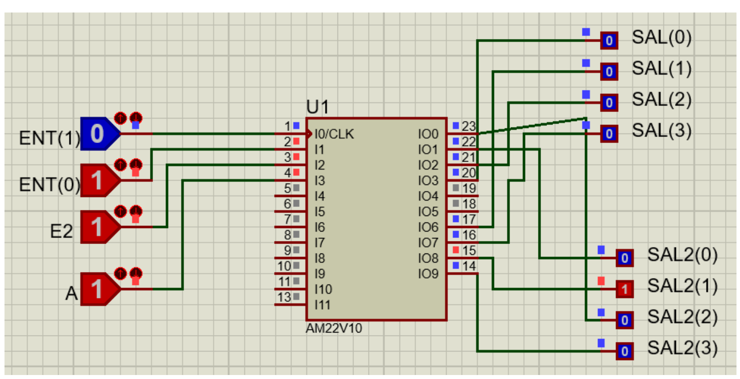


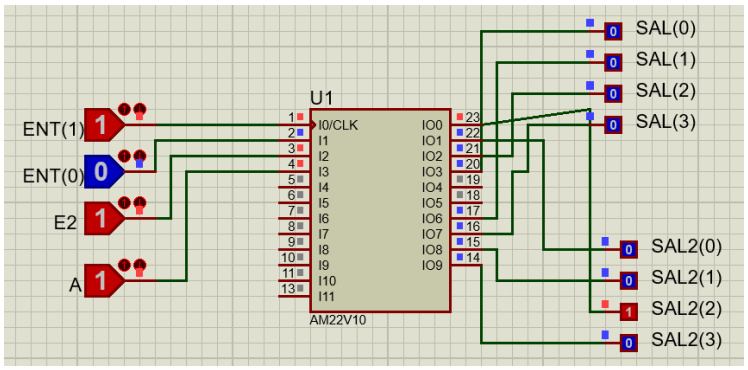


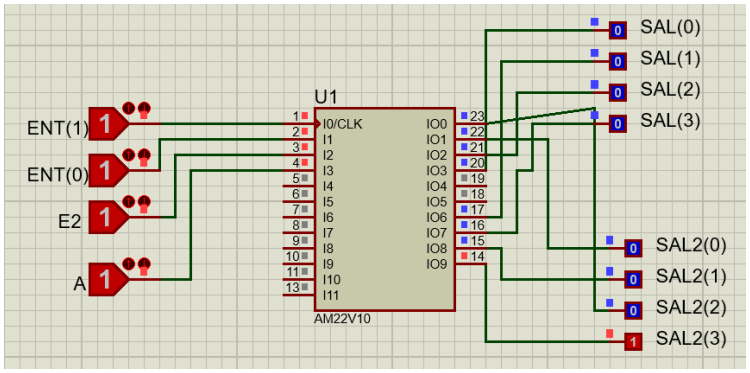

U1
ENT(1)
ENT(0)
E2
A
AM22V10
SAL(0)
SAL(1)
SAL(2)
SAL(3)
SAL2(0)
SAL2(1)
SAL2(2)
SAL2(3)


U1
ENT(1)
ENT(0)
E2
A
AM22V10
SAL(0)
SAL(1)
SAL(2)
SAL(3)
SAL2(0)
SAL2(1)
SAL2(2)
SAL2(3)


U1
ENT(1)
ENT(0)
E2
A
AM22V10
SAL(0)
SAL(1)
SAL(2)
SAL(3)
SAL2(0)
SAL2(1)
SAL2(2)
SAL2(3)


U1
ENT(1)
ENT(0)
E2
A
AM22V10
SAL(0)
SAL(1)
SAL(2)
SAL(3)
SAL2(0)
SAL2(1)
SAL2(2)
SAL2(3)


U1
ENT(1)
ENT(0)
E2
A
I0/CLK
I1
I2
I3
I4
I5
I6
I7
I8
I9
I10
I11
IO0
IO1
IO2
IO3
IO4
IO5
IO6
IO7
IO8
IO9
AM22V10
SAL(0)
SAL(1)
SAL(2)
SAL(3)
SAL2(0)
SAL2(1)
SAL2(2)
SAL2(3)


U1
ENT(1)
ENT(0)
E2
A
I0/CLK
I1
I2
I3
I4
I5
I6
I7
I8
I9
I10
I11
IO0
IO1
IO2
IO3
IO4
IO5
IO6
IO7
IO8
IO9
AM22V10
SAL(0)
SAL(1)
SAL(2)
SAL(3)
SAL2(0)
SAL2(1)
SAL2(2)
SAL2(3)