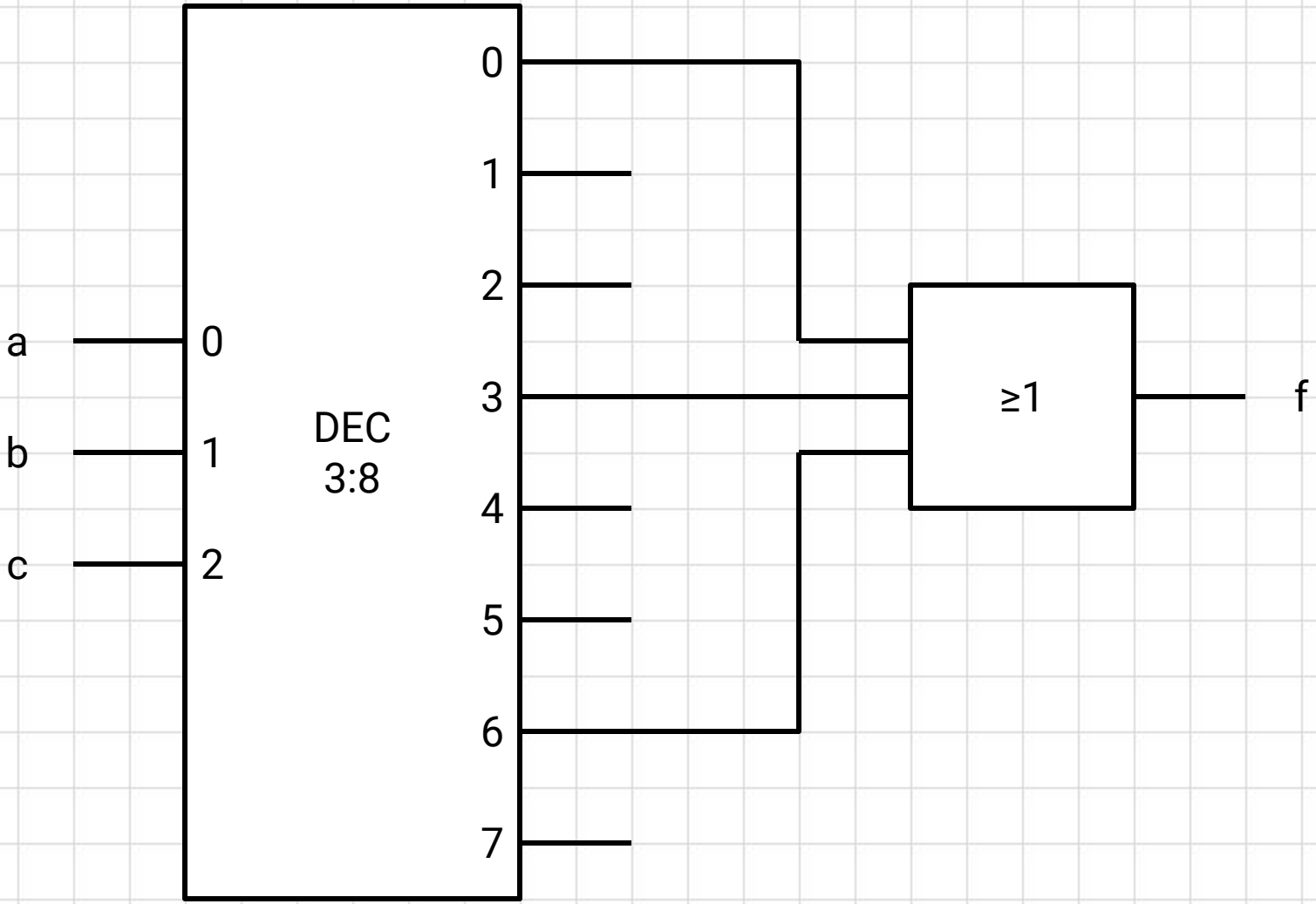
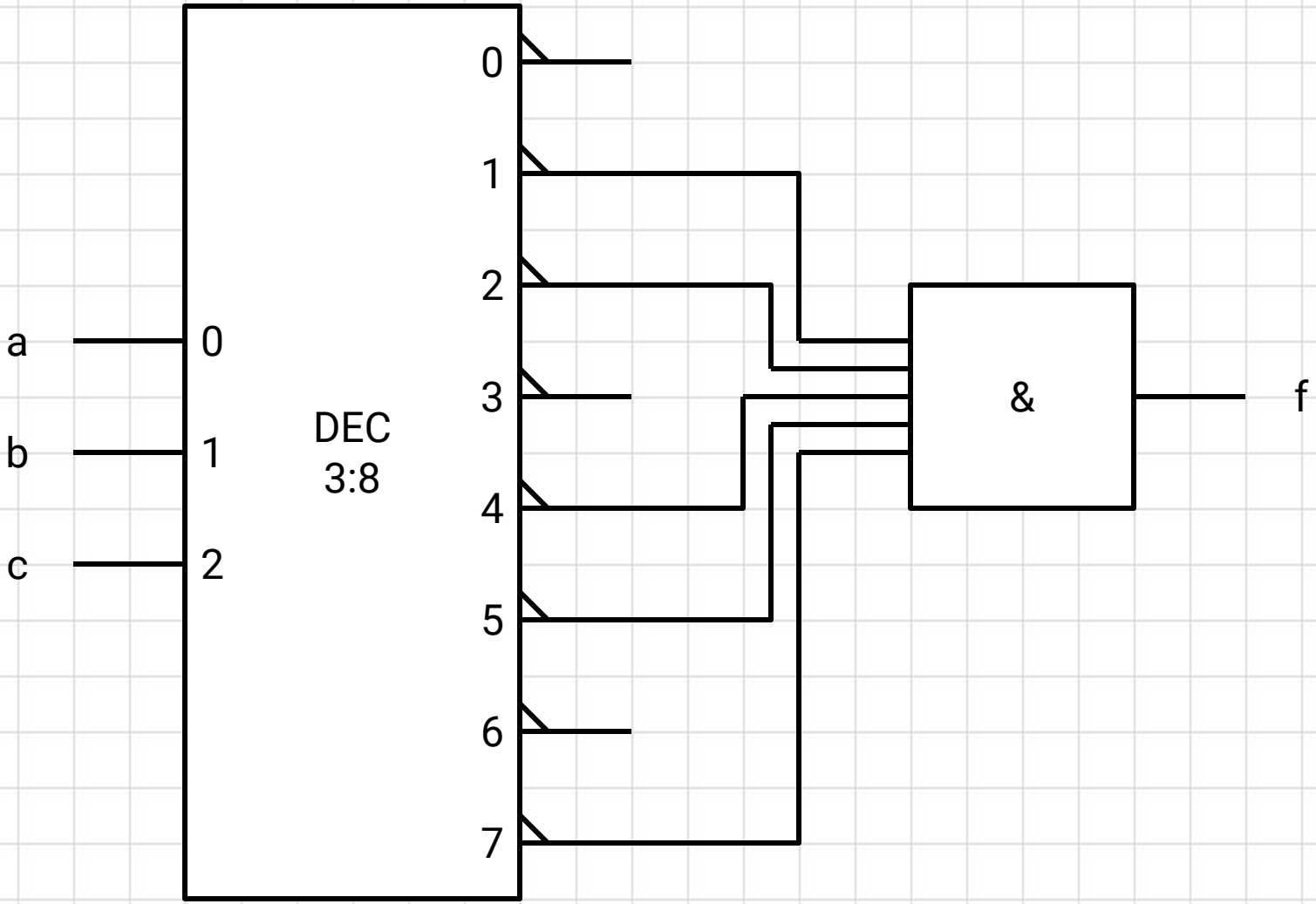
Problema 3-01

Decodificadores

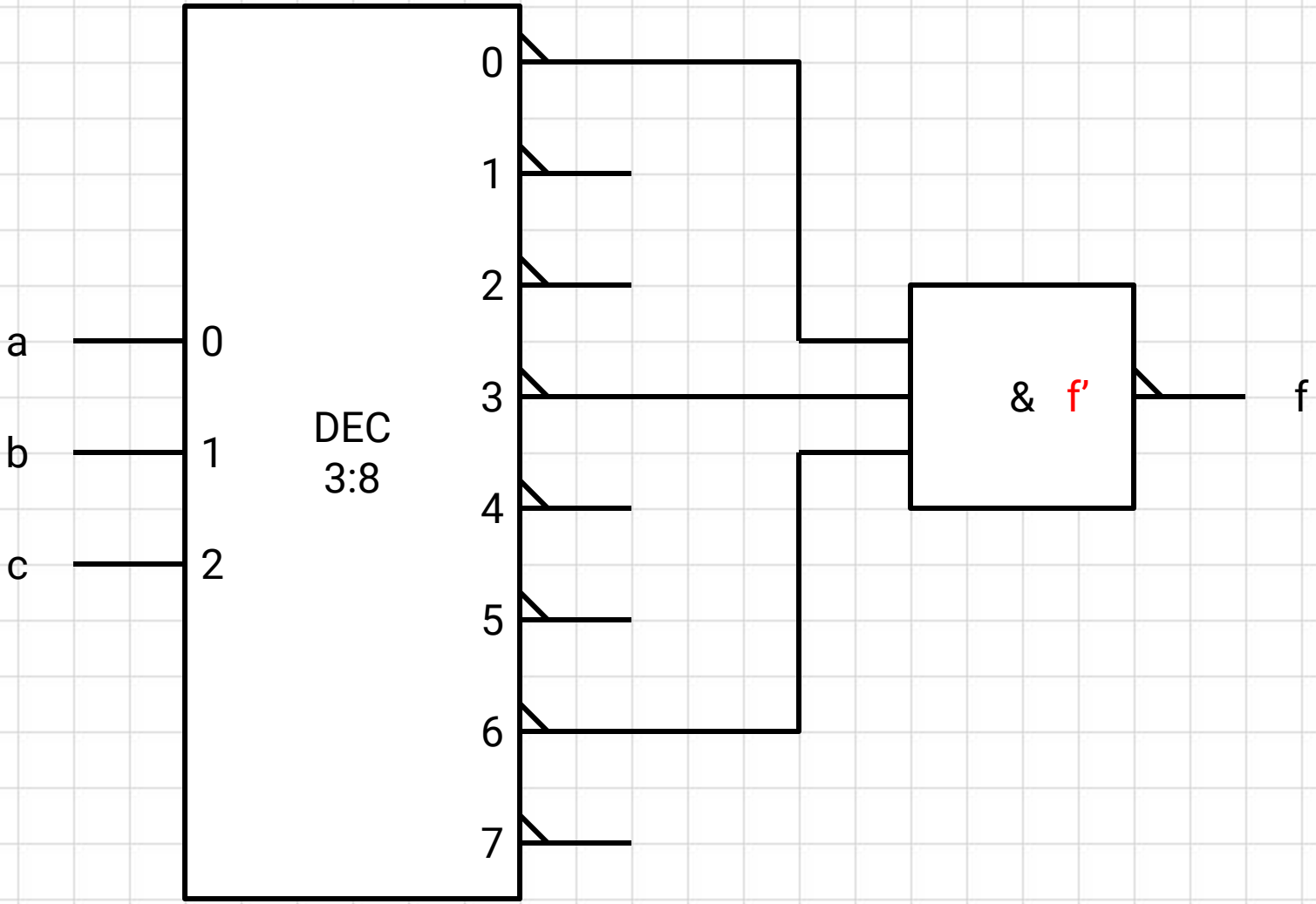
a) DEC (salidas en alto) y OR: f = ∑(0, 3, 6)



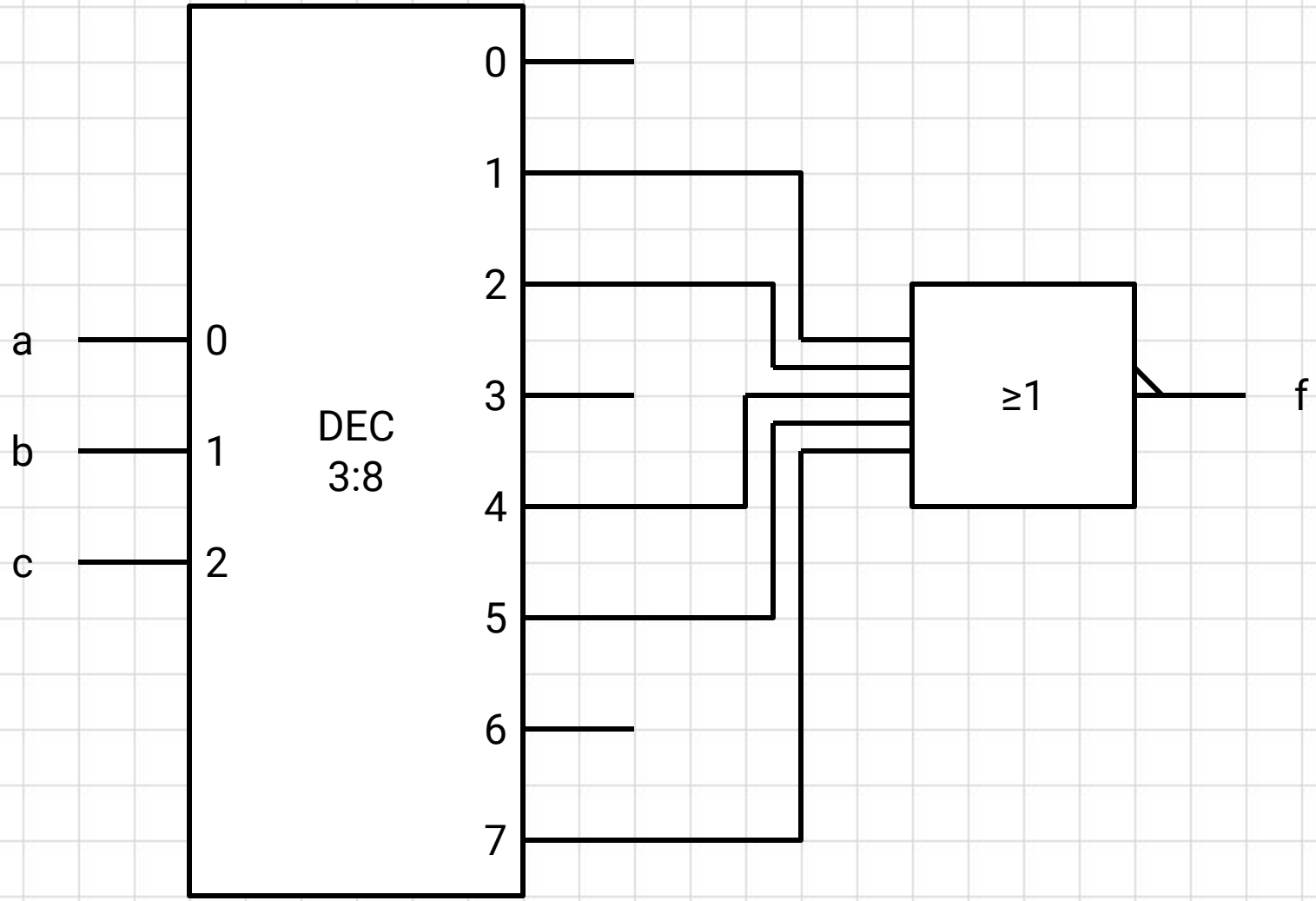
b) DEC (salidas en bajo) + AND: f = ∑(0, 3, 6) = ∏(1, 2, 4, 5, 7)



c) DEC (salidas en bajo) y NAND: f = ∑(0, 3, 6), f’ = ∏(0, 3, 6)



d) DEC (salidas en alto) + NOR: f = ∑(0, 3, 6) = ∏(1, 2, 4, 5, 7)

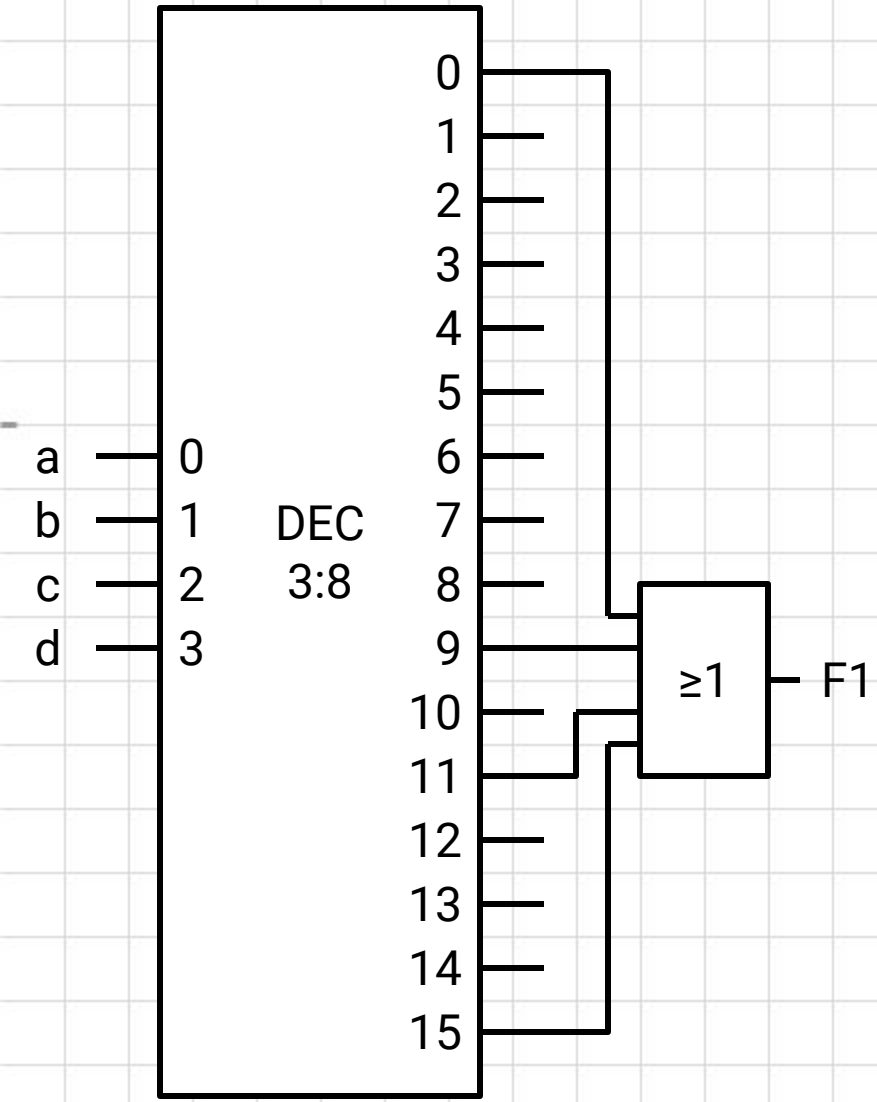


Problema 3-02

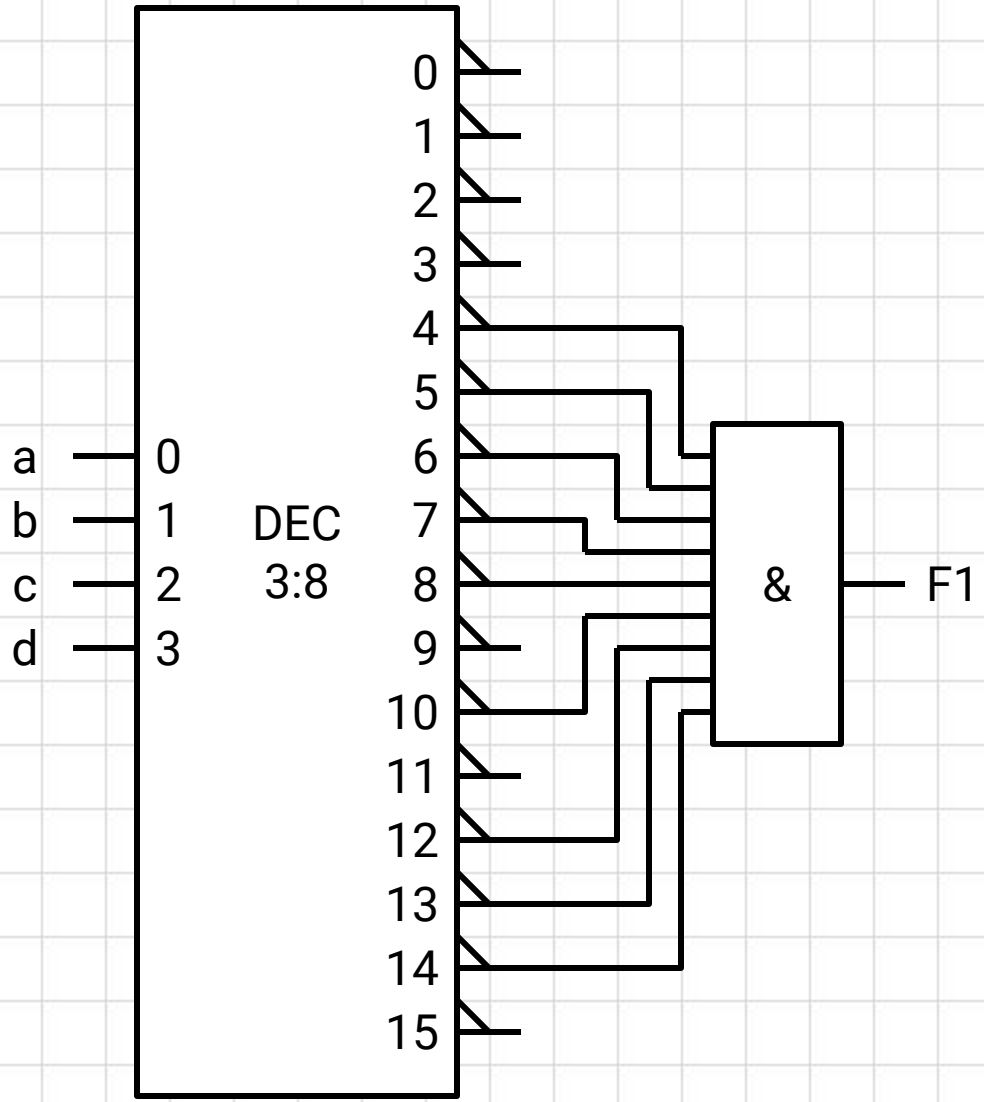
Decodificadores con indeterminaciones

$$F1 = \sum(0, 9, 11, 15) + d(1, 2, 3) = \prod(4, 5, 6, 7, 8, 10, 12, 13, 14) \cdot d(1, 2, 3)$$

a) DEC (salidas en alto) + OR:



b) DEC (salidas en bajo) + AND:



Problema 3-03

Conteo de ceros y unos

abcd	z2z1z0
0000	100
0001	100
0010	100
0011	010
0100	100
0101	010
0110	010
0111	001

abcd	z2z1z0
1000	100
1001	010
1010	010
1011	001
1100	010
1101	001
1110	001
1111	001

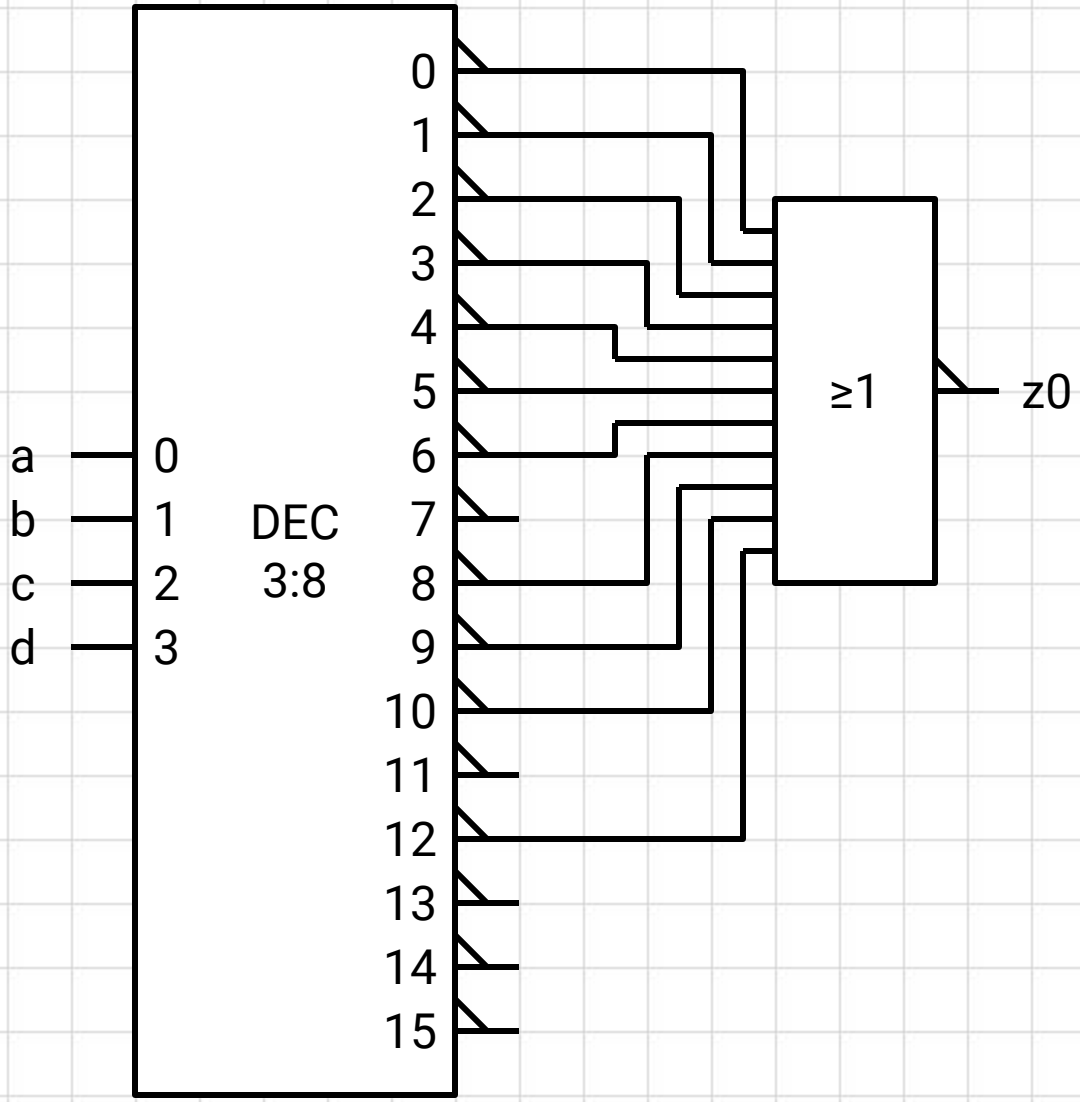
a)

$z0 = \Sigma(7, 11, 13, 14, 15)$

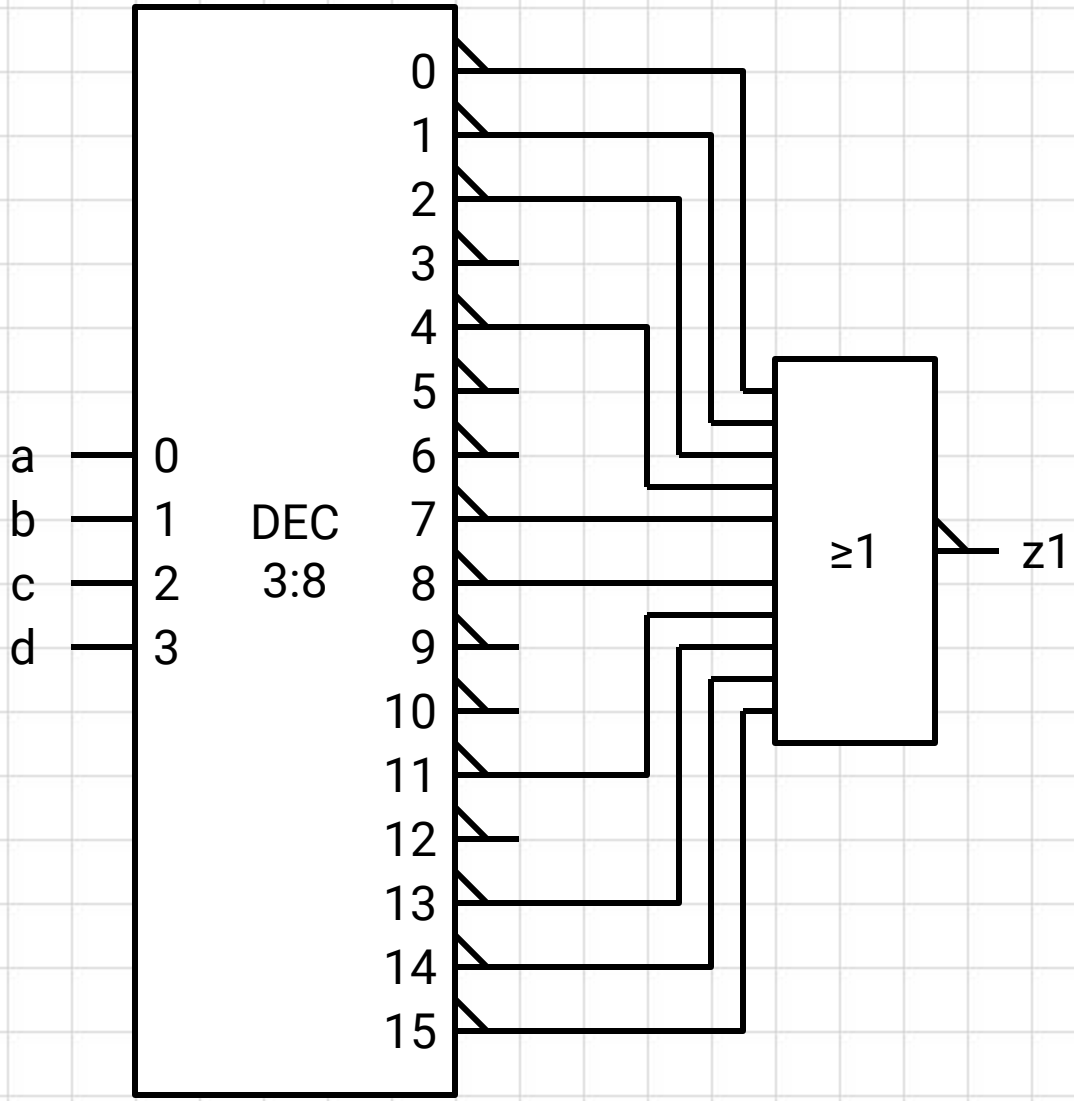
$z1 = \Sigma(3, 5, 6, 9, 10, 12)$

$z2 = \Sigma(0, 1, 2, 4, 8)$

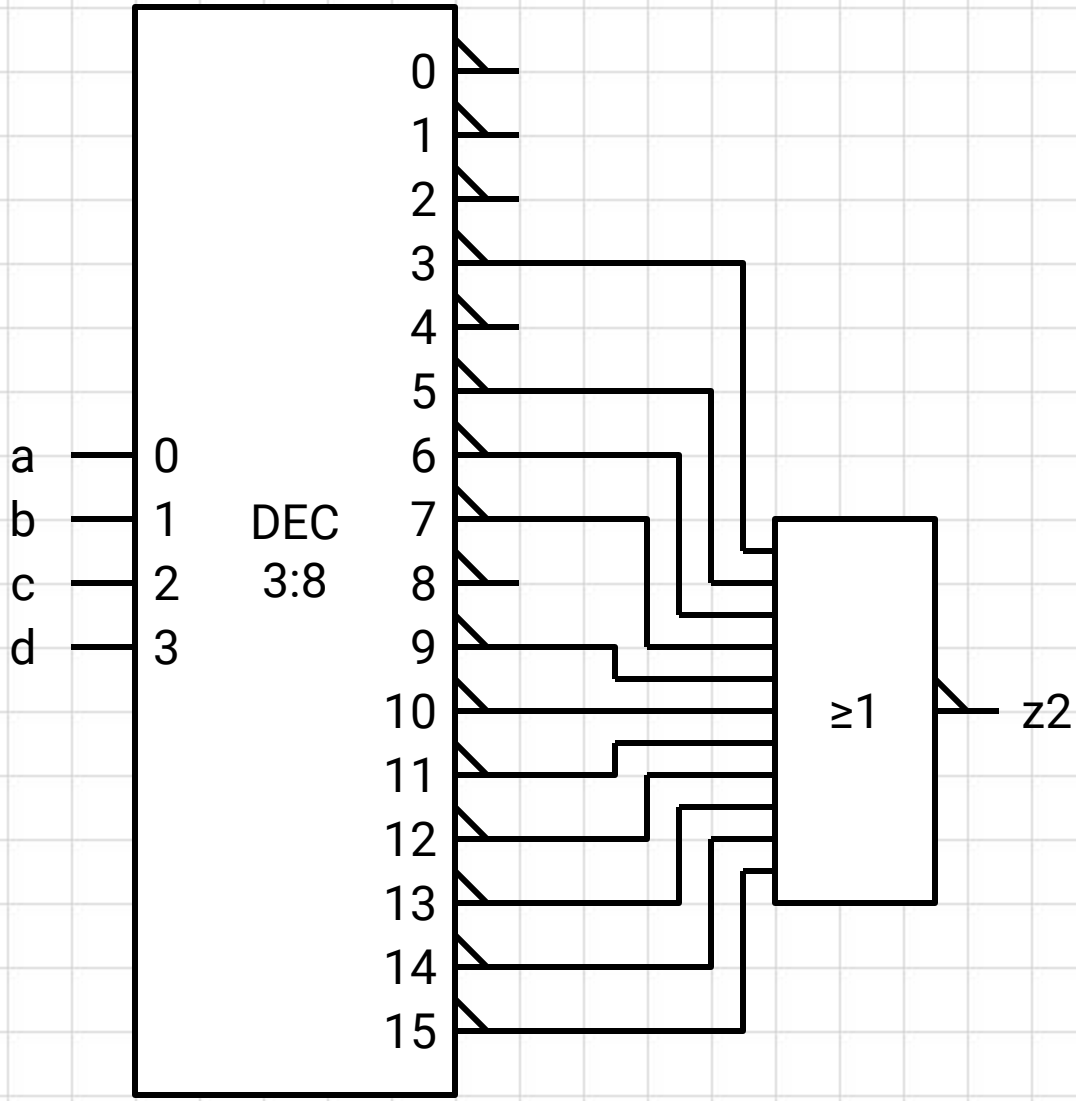
$z0 = \Sigma(7, 11, 13, 14, 15) = \Pi(0, 1, 2, 3, 4, 5, 6, 8, 9, 10, 12)$



$$z1 = \Sigma(3, 5, 6, 9, 10, 12) = \Pi(0, 1, 2, 4, 7, 8, 11, 13, 14, 15)$$



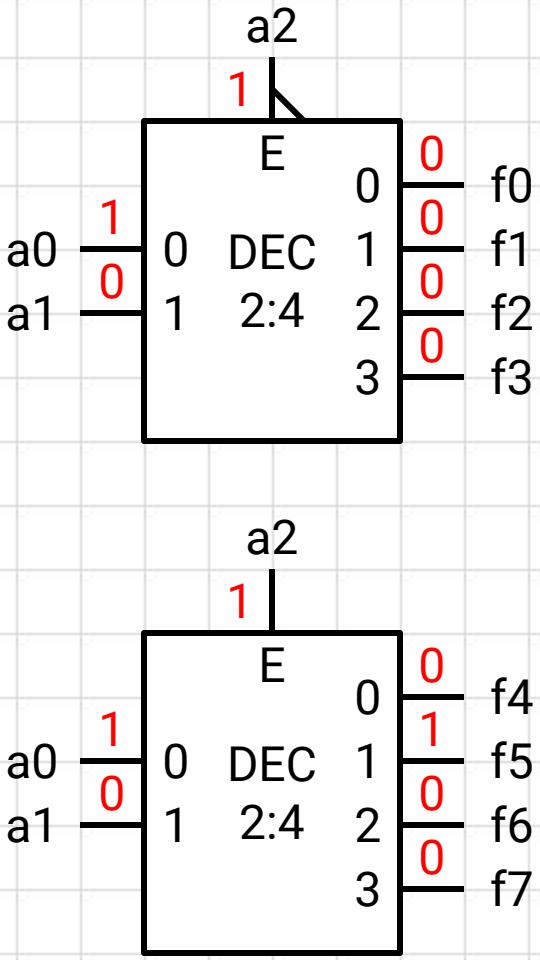
$z2 = \Sigma(0, 1, 2, 4, 8) = \Pi(3, 5, 6, 7, 9, 10, 11, 12, 13, 14, 15)$



Problema 3-04

Asociación de DEC

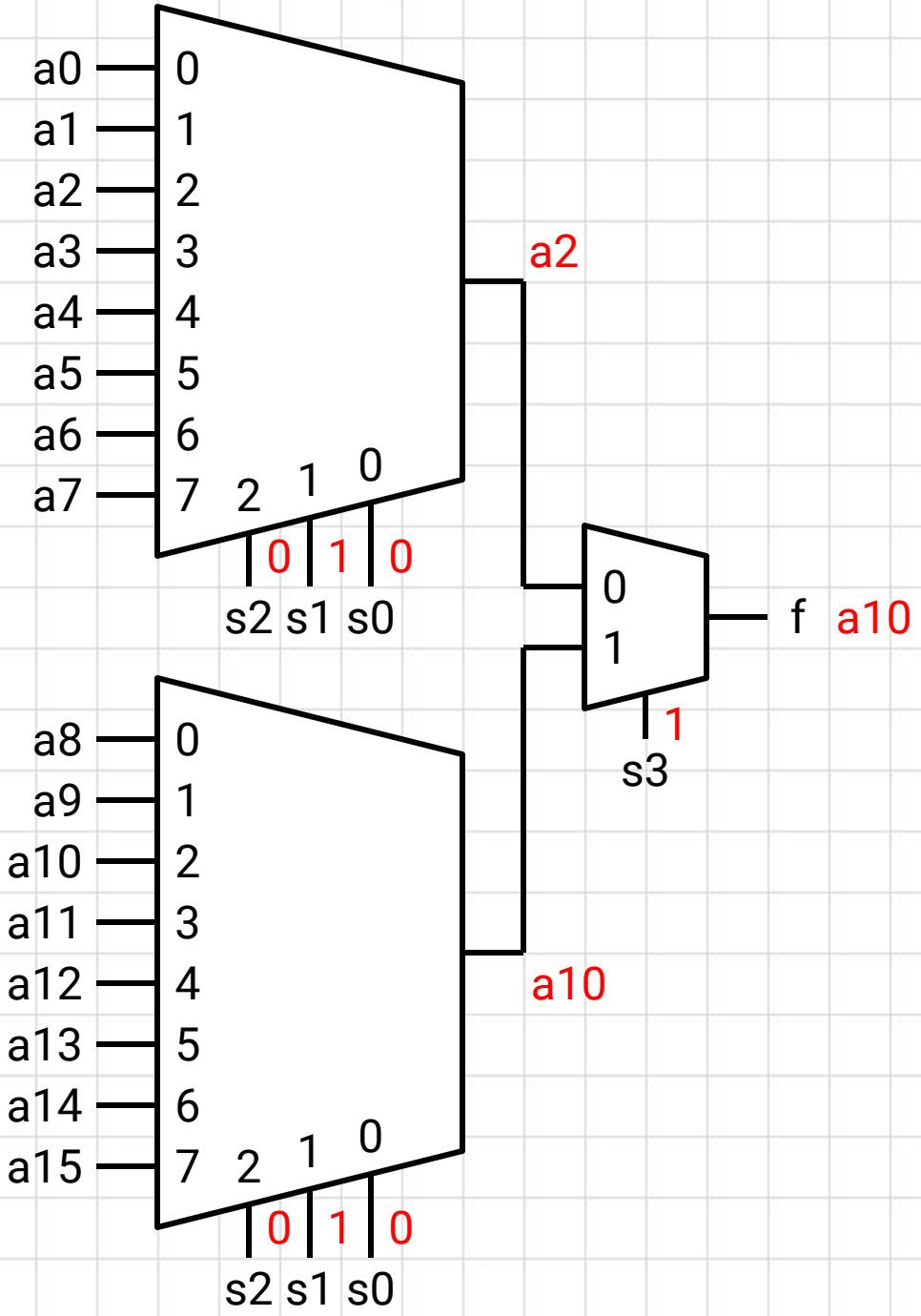
b) Un DEC 3:8


a2
1
E
a0
1
0
DEC
2:4
a1
0
1
0
1
2
3
0
0
0
0
f0
f1
f2
f3
a2
1
E
a0
1
0
DEC
2:4
a1
0
1
0
1
2
3
0
1
0
0
f4
f5
f6
f7

Problema 3-05

Asociación de MUX

b) Un MUX16


a0
a1
a2
a3
a4
a5
a6
a7
0
1
2
3
4
5
6
7
2
1
0
0
1
0
s2
s1
s0
a2
0
1
f
a10
1
s3
a8
a9
a10
a11
a12
a13
a14
a15
0
1
2
3
4
5
6
7
2
1
0
0
1
0
s2
s1
s0
a10

Problema 3-06

Implementación de funciones con MUX

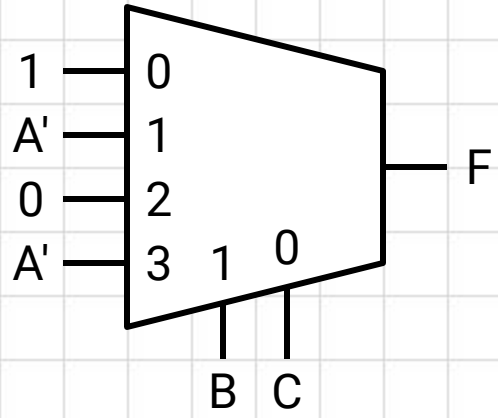
Realice las funciones de conmutación siguientes utilizando multiplexores de **4** canales (MUX 4:1). Considere las variables en doble rail.

- a) $F = \sum(0, 1, 3, 4)$
- b) $F = \sum(2, 4, 5, 7)$
- c) $F = \sum(0, 3, 4)$
- d) $F = \sum(1, 2, 3, 6, 7)$

a) F = ∑(0, 1, 3, 4)

A \ BC	00	01	11	10
0	1	1	1	0
1	1	0	0	0

F



Problema 3-08

DEC y MUX en Verilog

a)

E	A	F
0	**	1111
1	00	1110
1	01	1101
1	10	1011
1	11	0111

F0 = E' + A1 + A0

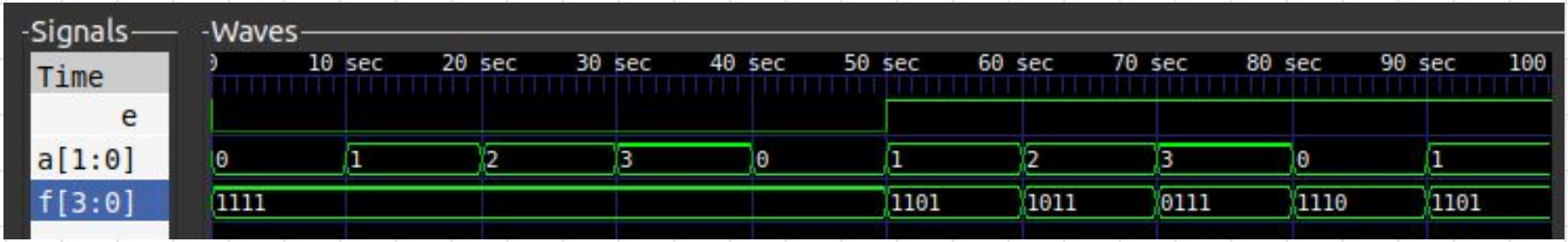
F1 = E' + A1 + A0'

F2 = E' + A1' + A0

F3 = E' + A1' + A0'

b)

```
module prob308b (input e, input [1:0] a, output [3:0] f);
  assign f = {4{~e}} | ~(1 << a);
endmodule
```



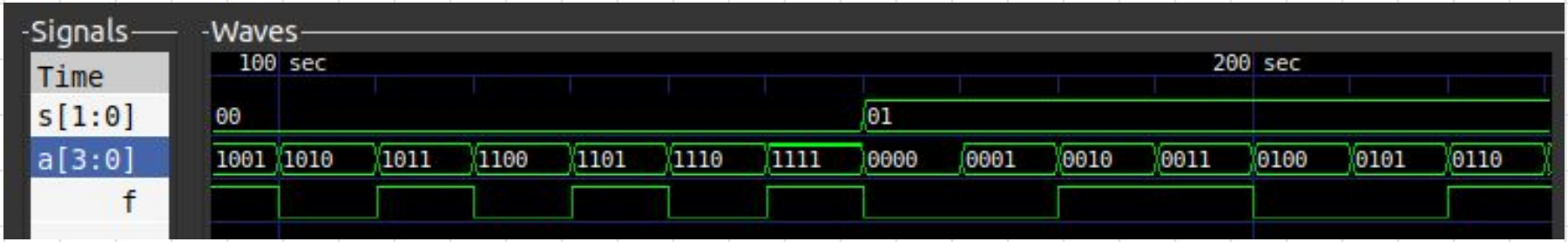
c)

S	A	F
00	***0	0
00	***1	1
01	**0*	0
01	**1*	1
10	*0**	0
10	*1**	1
11	0***	0
11	1***	1

F = S1'S0'A0 + S1'S0A1 + S1S0'A2 + S1S0A3

d)

```
module prob308d (input [1:0] s, input [3:0] a, output f);
  assign f = a[s];
endmodule
```



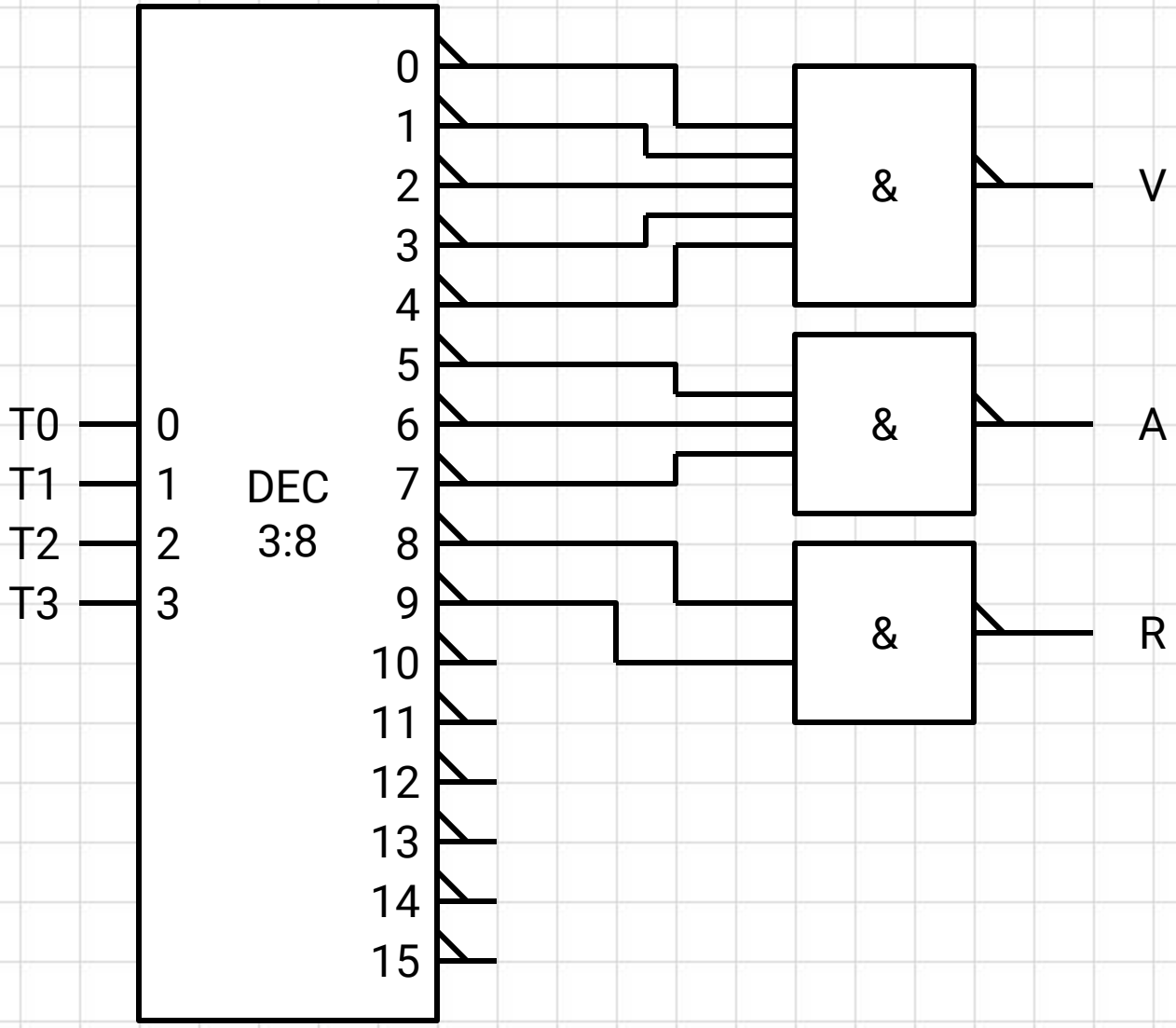
Problema 3-09

Sensor de toxicidad

a) $V = \sum(0, 1, 2, 3, 4) + d(10, 11, 12, 13, 14, 15)$

$A = \sum(5, 6, 7) + d(10, 11, 12, 13, 14, 15)$

$R = \sum(8, 9) + d(10, 11, 12, 13, 14, 15)$

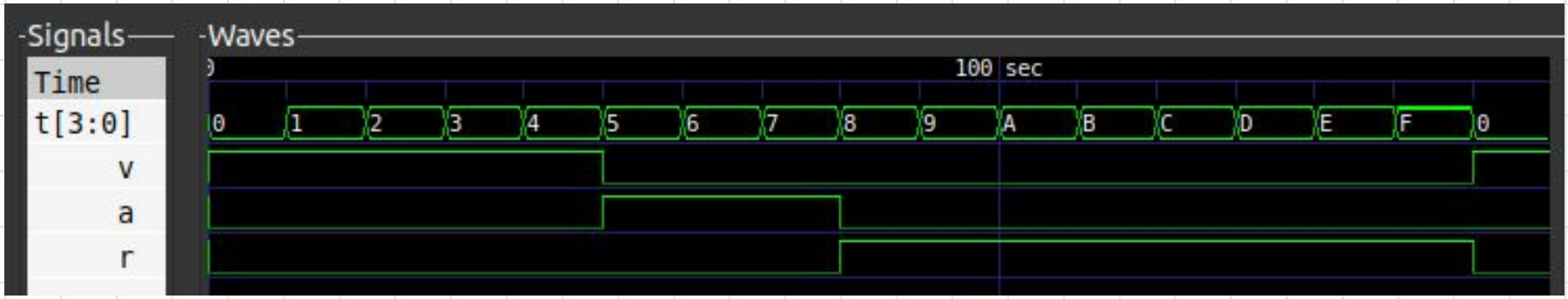


b)

```
module circuitox (
  input wire t3, t2, t1, t0,
  output reg r, a, v);

  always @* begin
    {r, a, v} = 0;
    if ({t3, t2, t1, t0} <= 4)
      v = 1;
    if ({t3, t2, t1, t0} >= 5 && {t3, t2, t1, t0} <= 7)
      a = 1;
    if ({t3, t2, t1, t0} >= 8)
      r = 1;
  end

endmodule
```

c)

```
module circuitox_tb;
  reg [3:0] t;
  wire r, a, v;
  circuitox dut(t[3], t[2], t[1], t[0], r, a, v);
  initial begin
    $dumpfile("sim.vcd"); $dumpvars;
    t = 0;
    #170 $finish;
  end
  always #10 t = t + 1;
endmodule
```



Problema 3-10

MUX4 en Verilog

a)

```
module prob310a (
    input [1:0] s,
    input [3:0] a,
    output f
);

  assign f = a[s];

endmodule
```

b)

```
module prob310b (
    input [1:0] s,
    input [3:0] a,
    output reg f
);

  always @*
    f = a[s];

endmodule
```

Problema 3-11

DEMUX4 en Verilog

a)

```
module demux4 (
    input [1:0] s,
    input a,
    output [3:0] f
);

  assign f = a << s;

endmodule
```

Problema 3-12

DEC 4:16 en Verilog

a)

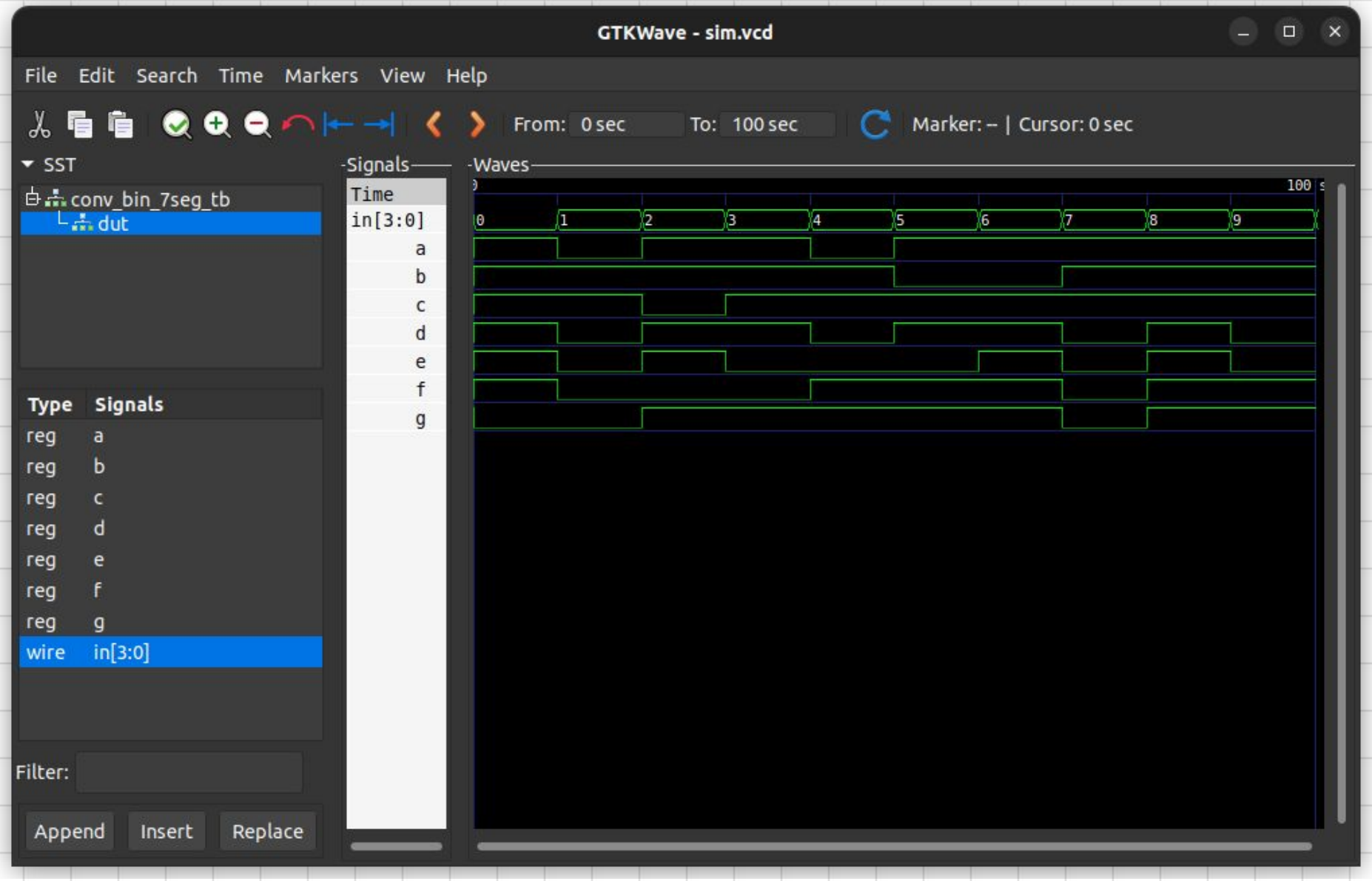
```
module dec4a16 (
    input  [ 3:0] a,
    output [15:0] f
);

  assign f = 1 << a;

endmodule
```

Problema 3-13

Convertidor BCD a 7-Segment en Verilog

a)

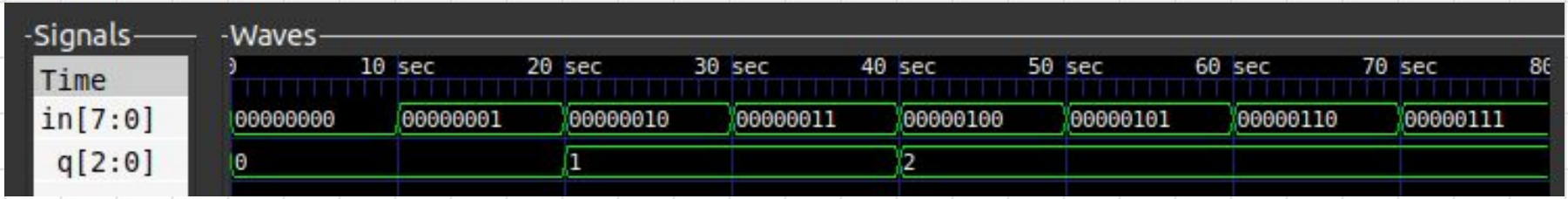
```
module conv_bin_7seg (
  input [3:0] in,
  output reg a, b, c, d, e, f, g);
  always @*
    case (in)
      0:       {a, b, c, d, e, f, g} = 7'b1111110;
      1:       {a, b, c, d, e, f, g} = 7'b0110000;
      2:       {a, b, c, d, e, f, g} = 7'b1101101;
      3:       {a, b, c, d, e, f, g} = 7'b1111001;
      4:       {a, b, c, d, e, f, g} = 7'b0110011;
      5:       {a, b, c, d, e, f, g} = 7'b1011011;
      6:       {a, b, c, d, e, f, g} = 7'b1011111;
      7:       {a, b, c, d, e, f, g} = 7'b1110000;
      8:       {a, b, c, d, e, f, g} = 7'b1111111;
      default: {a, b, c, d, e, f, g} = 7'b1110011;
    endcase
endmodule
```


GTKWave - sim.vcd
File Edit Search Time Markers View Help
From: 0 sec
To: 100 sec
Marker: – | Cursor: 0 sec
SST
conv_bin_7seg_tb
dut
Type Signals
reg a
reg b
reg c
reg d
reg e
reg f
reg g
wire in[3:0]
Filter:
Append Insert Replace
Signals
Time
in[3:0]
a
b
c
d
e
f
g
Waves
100
0 1 2 3 4 5 6 7 8 9

Problema 3-14

Codificador prioridad 8:3 en Verilog

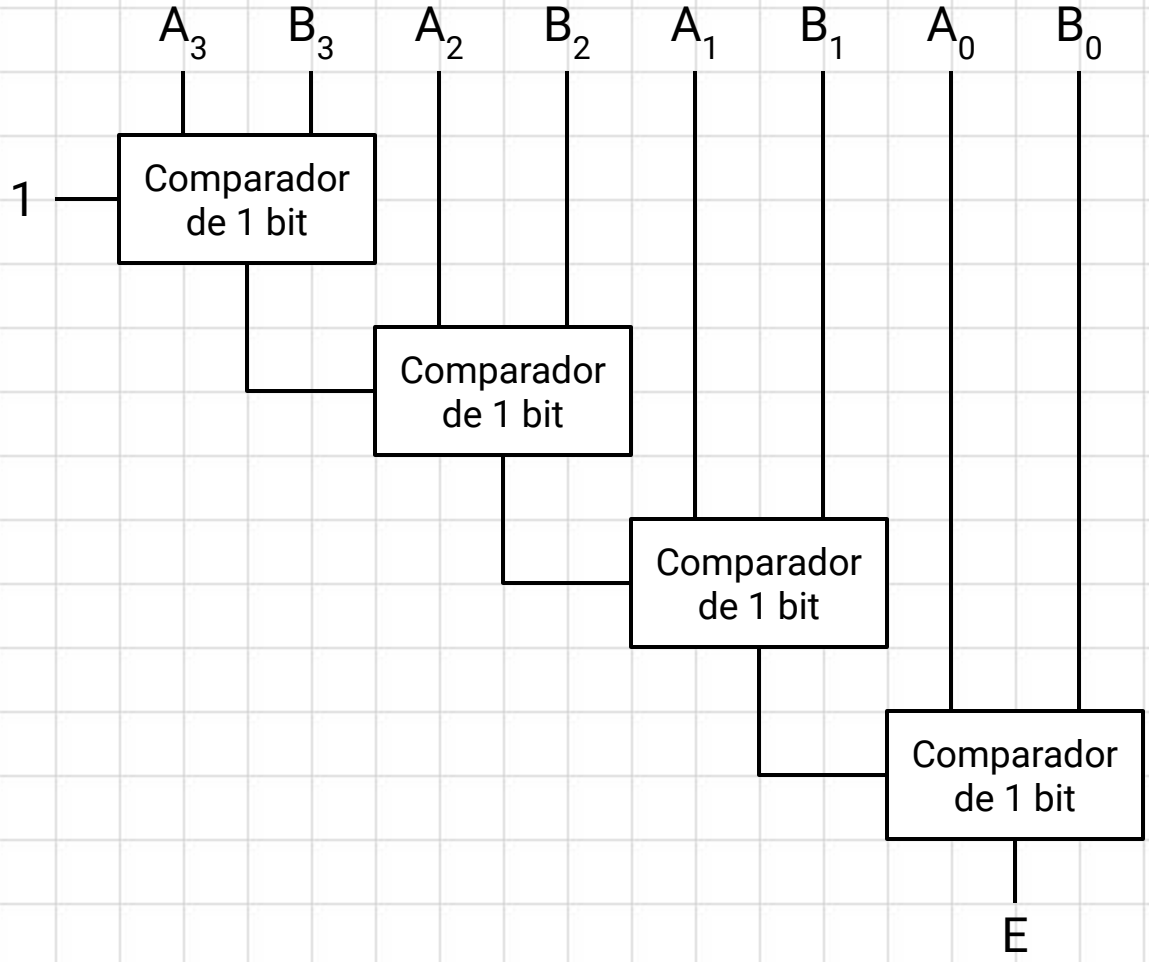
```
module cod_pri_8_3 (input [7:0] in, output reg [2:0] q);

  always @* begin
    q = 0;
    for (integer n = 1; n < 8; n = n + 1)
      if (in[n]) q = n;
  end

endmodule
```



Problema 3-15

Comparador

a)

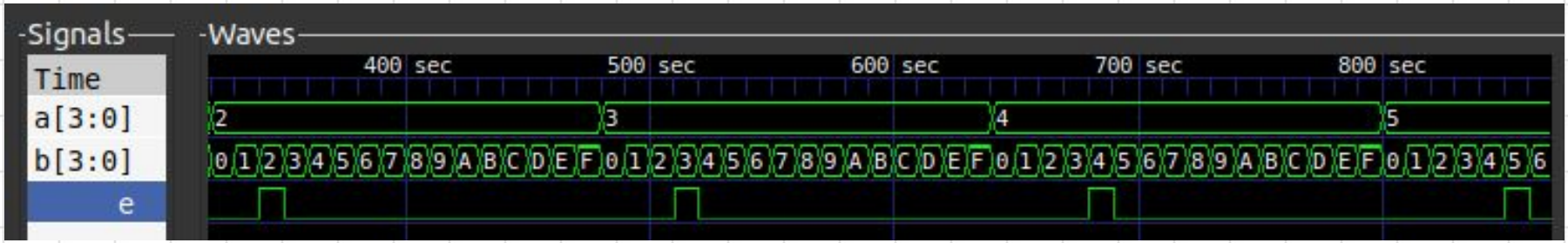

A3
B3
A2
B2
A1
B1
A0
B0
1
Comparador de 1 bit
Comparador de 1 bit
Comparador de 1 bit
Comparador de 1 bit
E

b)

```
module comp1bit(input a, b, c, output e);
  assign e = c && (a == b);
endmodule
```

c)

```
module comp4bit(input [3:0] a, input [3:0] b, output e);
  wire w1, w2, w3;
  comp1bit c3(a[3], b[3], 1'b1, w3);
  comp1bit c2(a[2], b[2],   w3, w2);
  comp1bit c1(a[1], b[1],   w2, w1);
  comp1bit c0(a[0], b[0],   w1,  e);
endmodule
```



d)

```
module comp4bit_proc(input [3:0] a, input [3:0] b,
                     output reg e);
  always @*
    e = (a == b);

endmodule
```