Problema 2-01

Expresiones de circuitos

AMC

A partir de las tablas de verdad de las siguientes funciones, obtenga las expresiones algebraicas de dichas funciones y los circuitos lógicos que las realizan:

XY	F1	F2	F3
00	1	0	1
01	0	1	1
10	1	1	1
11	0	0	0

F1 = Y’

F2 = X $\oplus$ Y

F3 = (XY)’

XYZ	F4	F5	F6	F7	F8	F9
000	0	1	0	1	1	1
001	1	0	0	1	0	0
010	0	0	0	1	1	1
011	0	1	0	0	0	0
100	0	1	1	1	1	1
101	1	1	1	1	1	0
110	0	0	0	1	1	1
111	0	0	0	1	1	0

F4 = ∑(1, 5) = X'Y'Z + XY'Z = Y'Z(X' + X) = Y'Z

F5 = ∑(0, 3, 4, 5) = ...

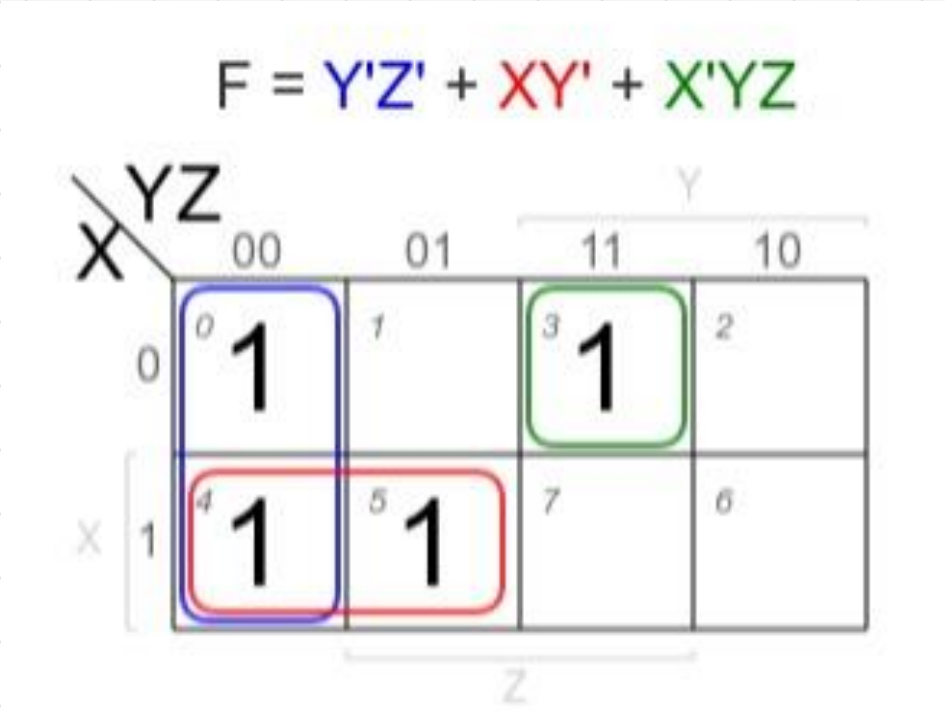
F6 = ∑(4, 5) = XY'Z' + XY'Z = XY'(Z' + Z) = XY'

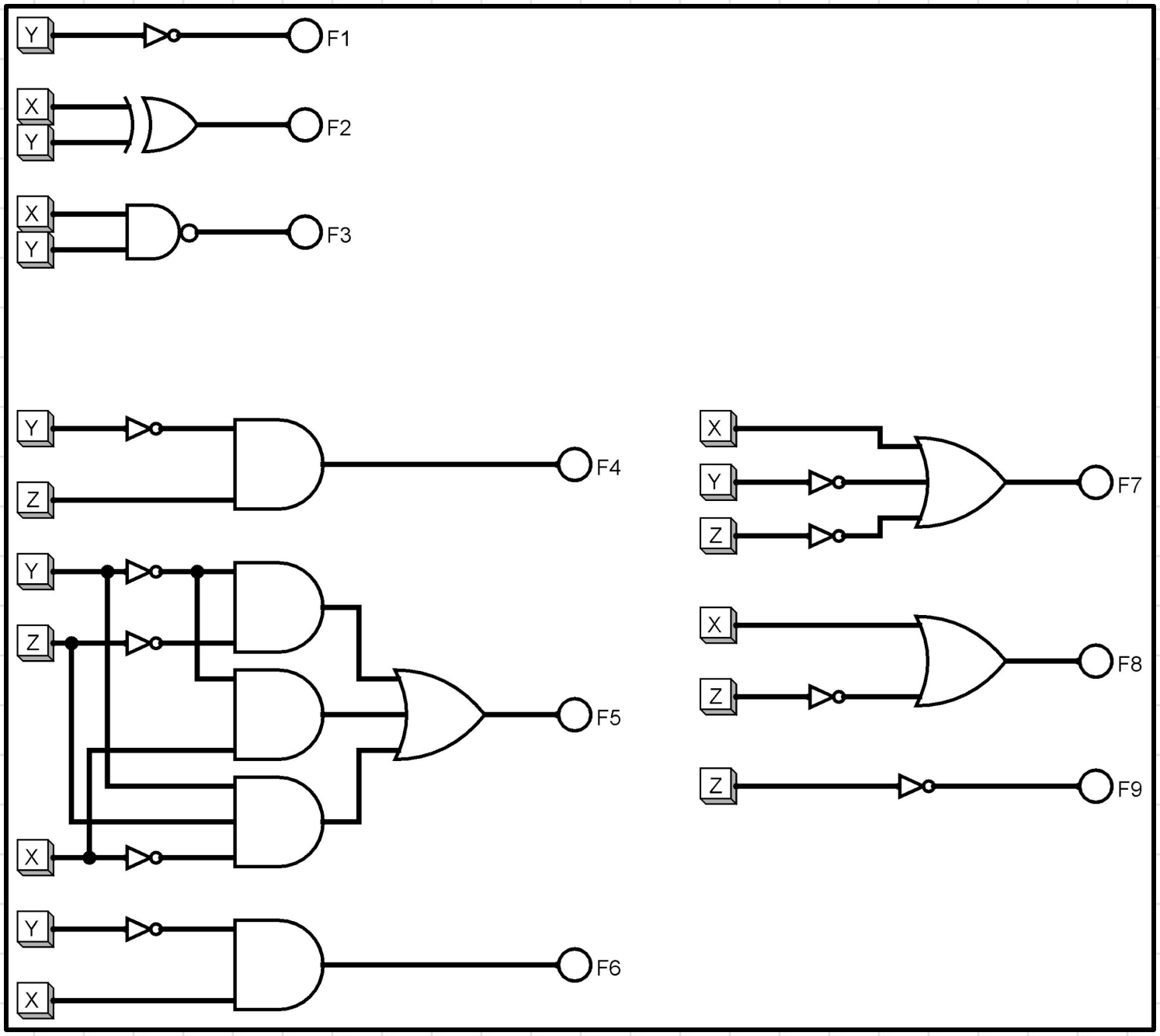
F7 = ∏(3) = (X + Y' + Z')

F8 = ∏(1, 3) = (X + Y + Z')(X + Y' + Z') = (X + Z') + ~~YY'~~

F9 = Z'

F5 = ∑(0, 3, 4, 5)

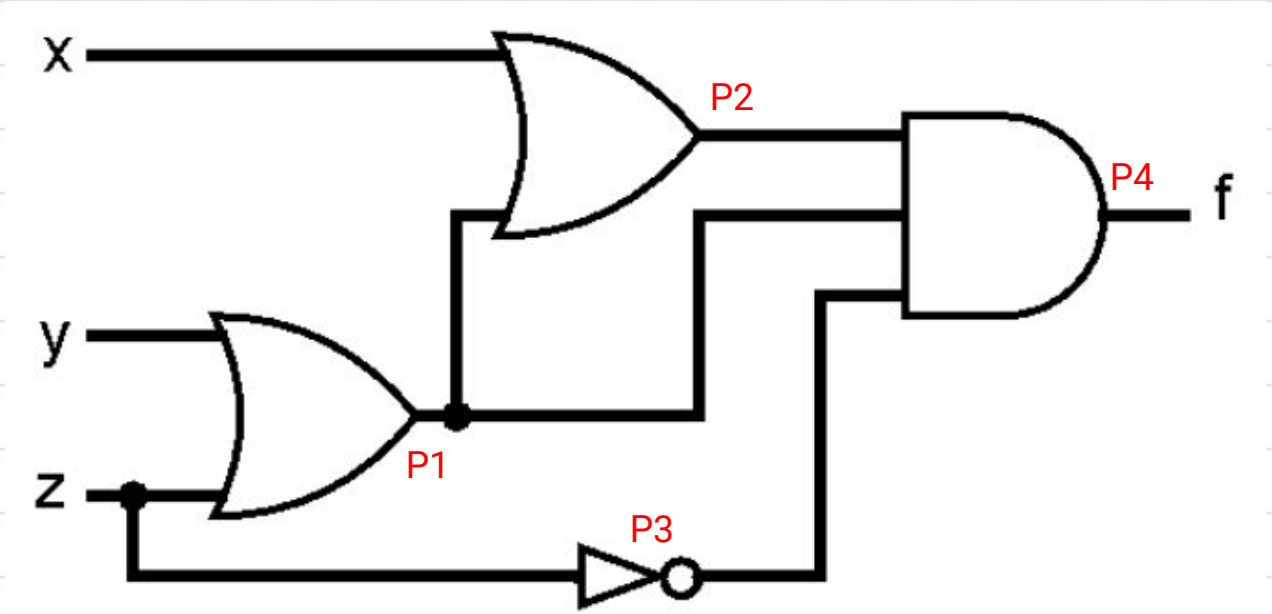



Y
F1
X
Y
F2
X
Y
F3
Y
Z
F4
Y
Z
X
F5
Y
X
F6
X
Y
Z
F7
X
Z
F8
Z
F9

Problema 2-02

Análisis

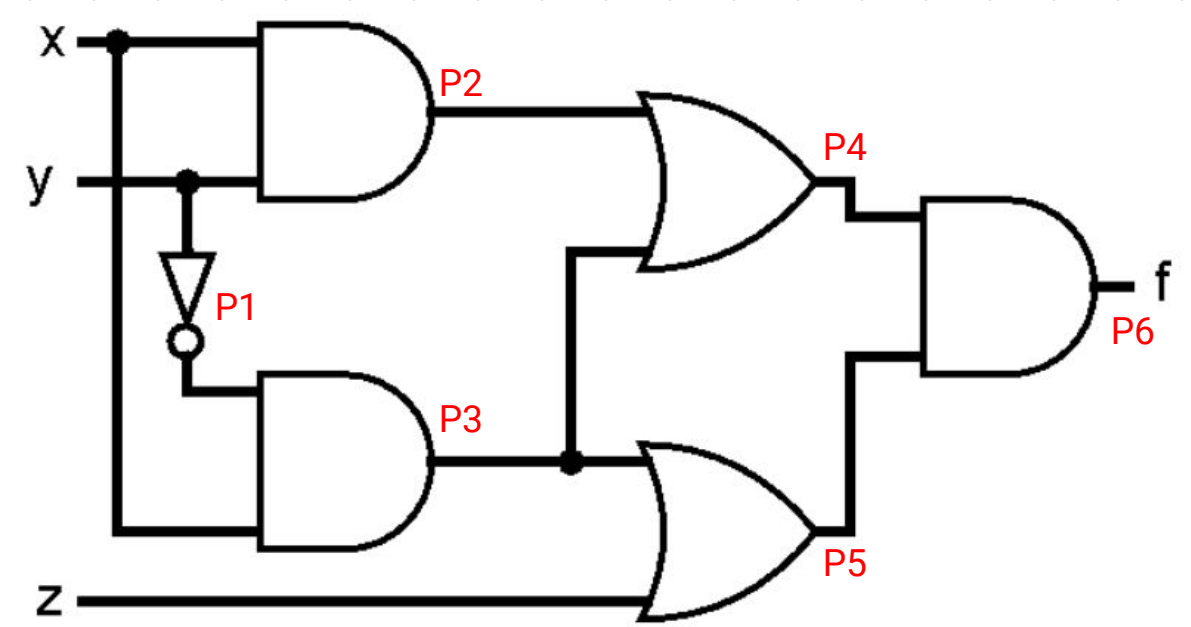
a)



$F = P4 = P1 \cdot P2 \cdot P3 = (X + P2) \cdot P2 \cdot Z' = X \cdot P2 \cdot Z' + P2 \cdot Z' =$

$= P2 \cdot Z' = (Y + Z)Z' = YZ' + ZZ' = YZ'$

b)



F = P6 = P4·P5 = (P2 + P3)(P3 + Z) =

= (XY + Y'X)(Y'X + Z) =

= XYY'X + XYZ + Y'XY'X + Y'XZ =

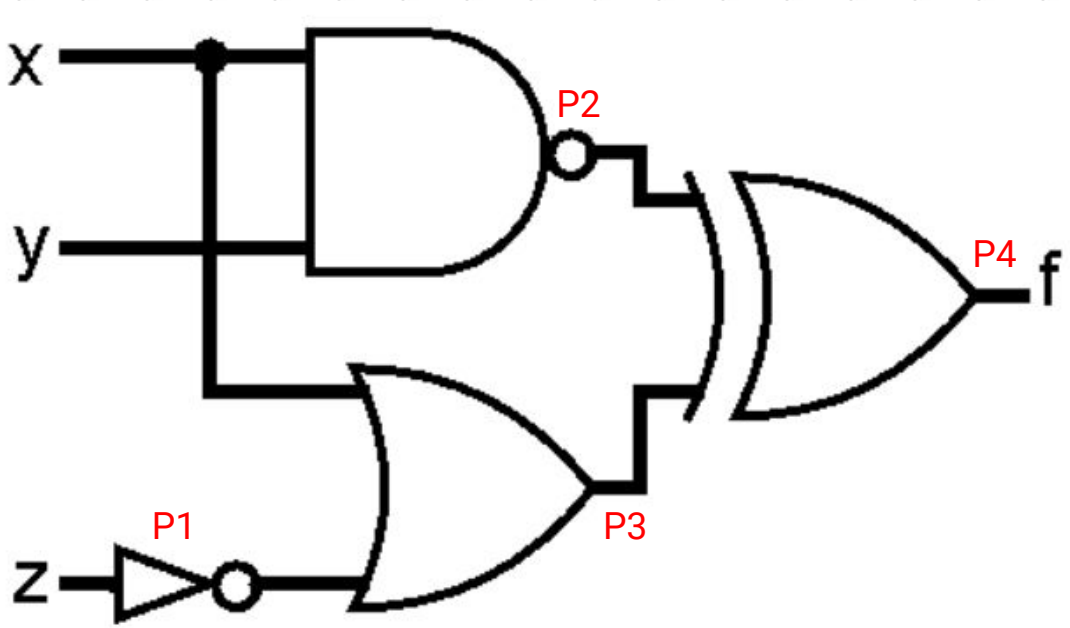
= ~~XYY'X~~ + XYZ + XY' + XY'Z =

= XYZ + XY' + XY'Z =

= XYZ + XY' = X(YZ + Y') =

= X(Z + Y') = XZ + XY'

c)



$F = P4 = P2 \oplus P3 = P2'P3 + P2P3' =$

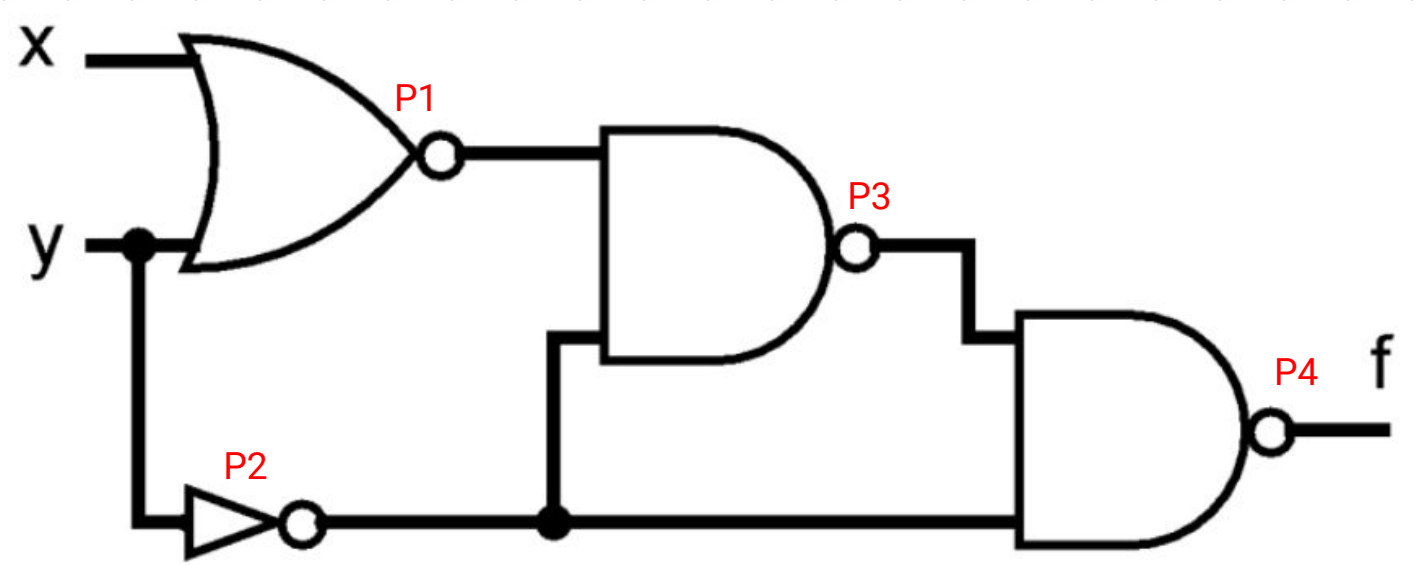
$= (XY)(X + Z') + (XY)'(X + Z')' =$

$= XYX + XYZ' + (X' + Y')(X'Z) =$

$= XYX + XYZ' + X'X'Z + Y'X'Z =$

$= XY + \cancel{XYZ'} + X'Z + \cancel{X'ZY'} = XY + X'Z$

d)



F = P4 = P2 ⊙ P3 = P2 ⊙ (P1 P2)' =

= P2 ⊙ ((X + Y)' P2)' =

= Y' ⊙ ((X + Y)' Y')' = Y' ⊙ (X'Y')' = Y' ⊙ (X + Y) =

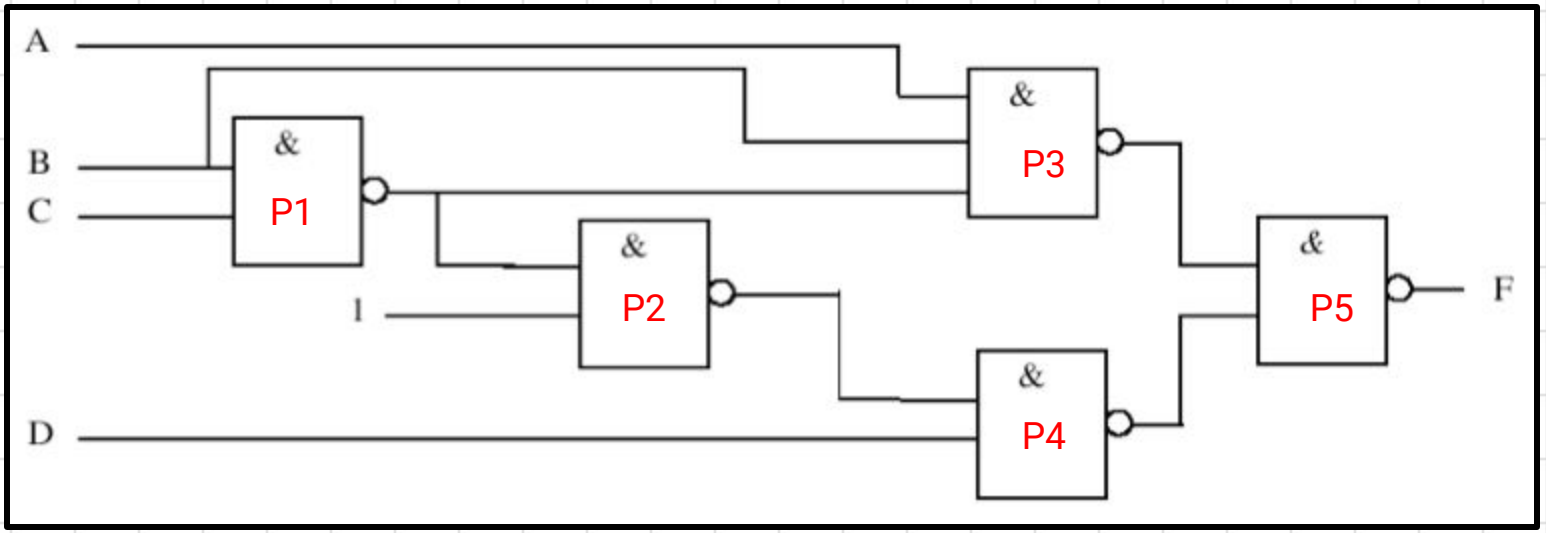
= Y'(X + Y) + Y(X + Y)' = Y'X + ~~Y'Y~~ + ~~YX'Y'~~ =

= XY'

Problema 2-03

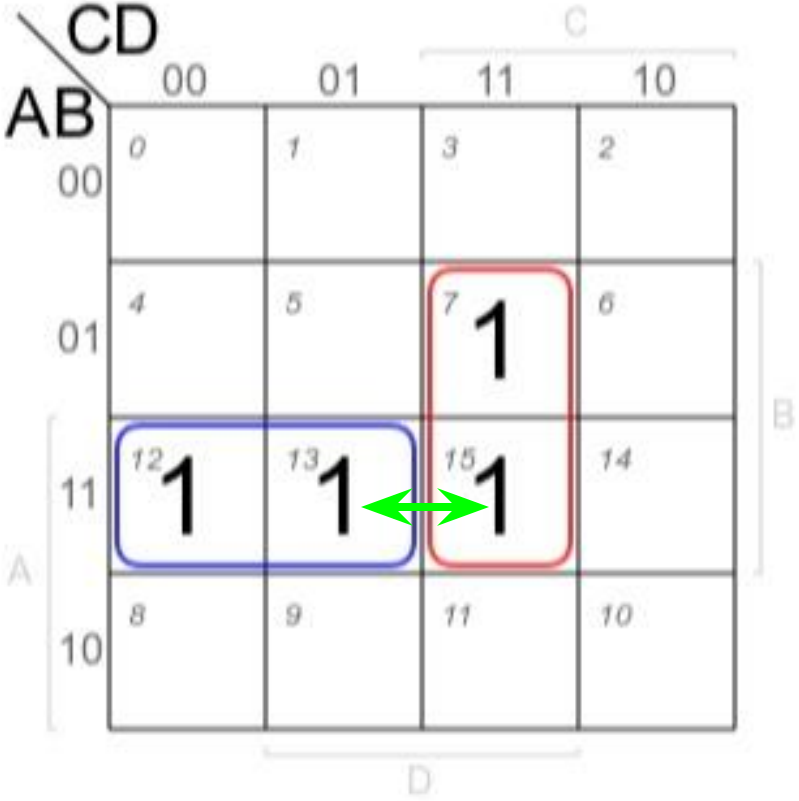
Azares

a)



$$F = P5 = (P3 \cdot P4)' = P3' + P4' = A \cdot B \cdot P1 + P2 \cdot D =$$
$$= AB(BC)' + P1'D = AB(B' + C') + BCD =$$
$$= \cancel{ABB'} + ABC' + BCD = ABC' + BCD$$

F = ABC' + BCD

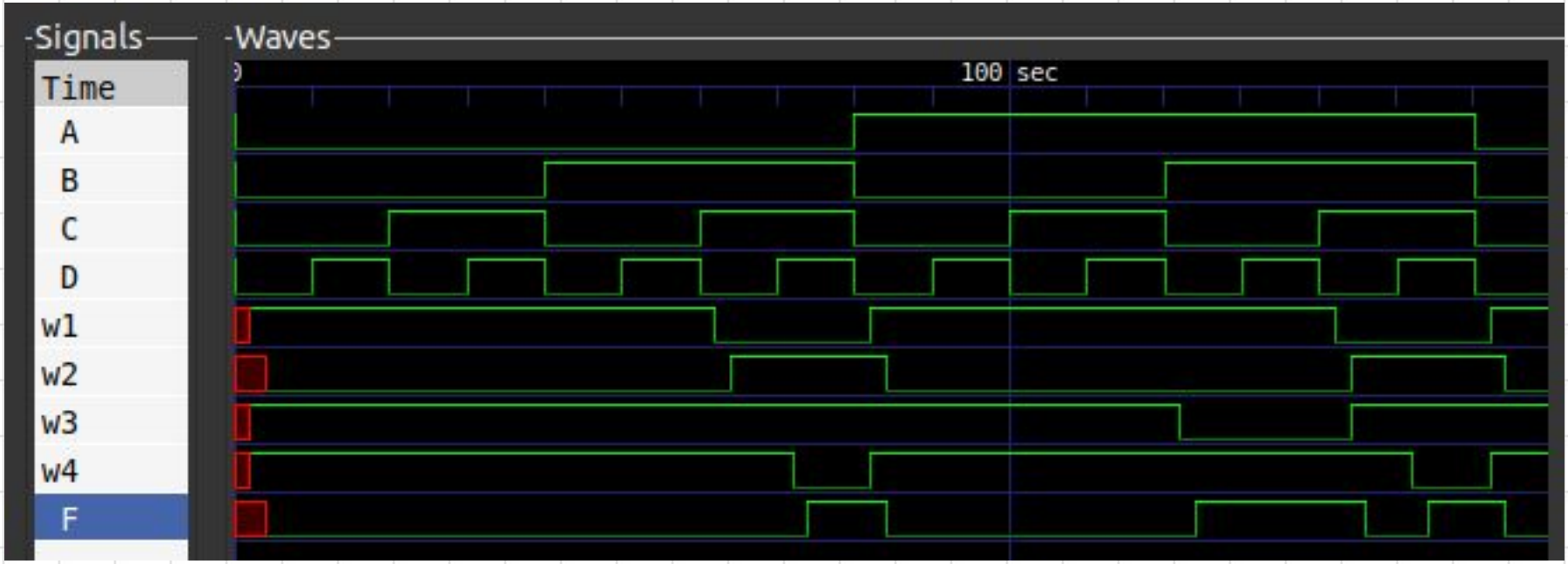


b)

```
module prob203b(input A, B, C, D, output F);

  wire w1, w2, w3, w4;

  nand #2 p1(w1, B, C);
  nand #2 p2(w2, w1, 1);
  nand #2 p3(w3, A, B, w1);
  nand #2 p4(w4, w2, D);
  nand #2 pF(F, w3, w4);

endmodule
```

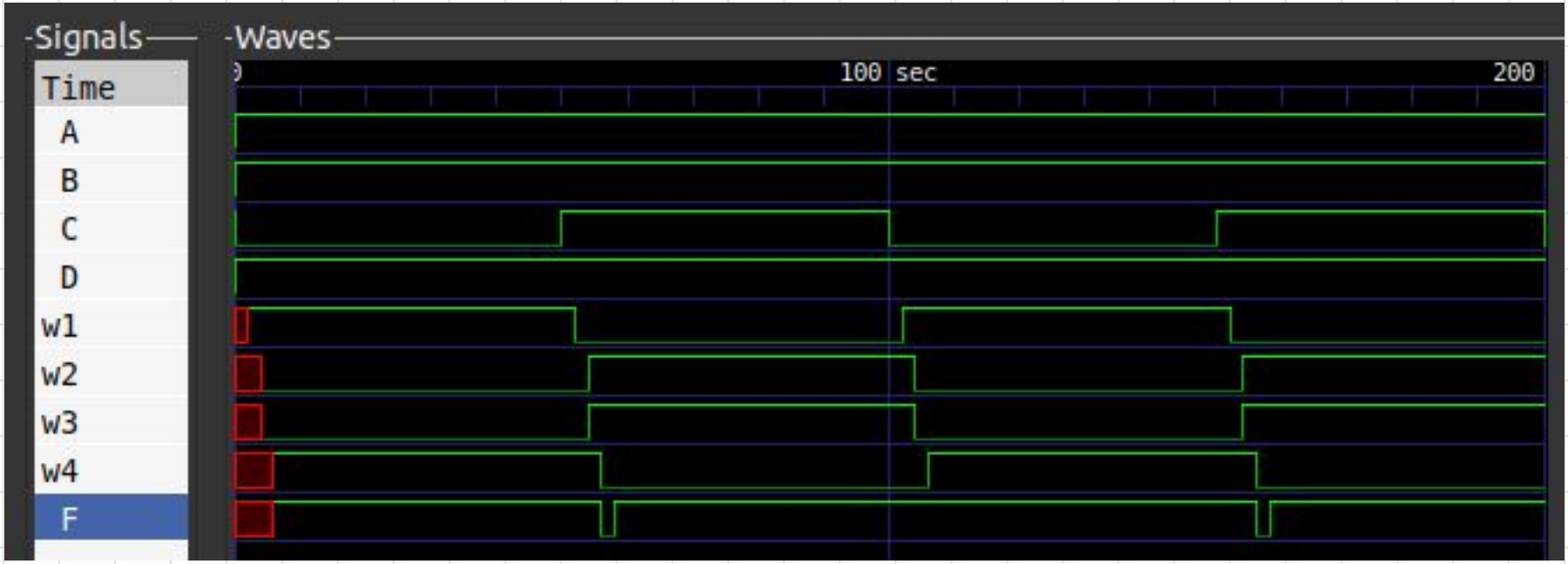
c)

```
module prob203c;

  reg A, B, C, D;
  wire F;

  prob203b dut(A, B, C, D, F);

  initial begin
    $dumpfile("prob203c.vcd");
    $dumpvars;
    {A, B, C, D} = 0;
    #170 $finish;
  end

  always #10 {A, B, C, D} = {A, B, C, D} + 1;

endmodule
```


Signals
Waves
Time
A
B
C
D
w1
w2
w3
w4
F
100 sec

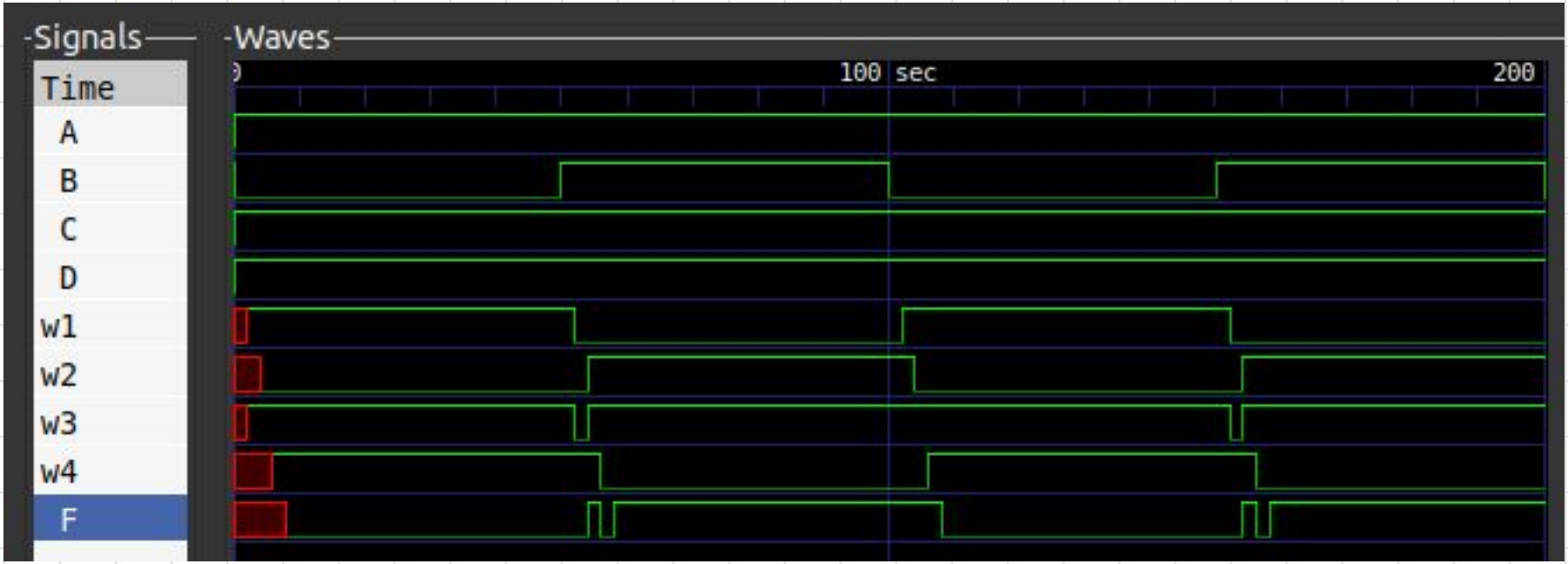
d)

```
module prob203d;

  reg A, B, C, D;
  wire F;

  prob203b dut(A, B, C, D, F);

  initial begin
    $dumpfile("prob203d.vcd");
    $dumpvars;
    {A, B, C, D} = 4'b1101;
    #200 $finish;
  end

  always #50 C = ~C;

endmodule
```

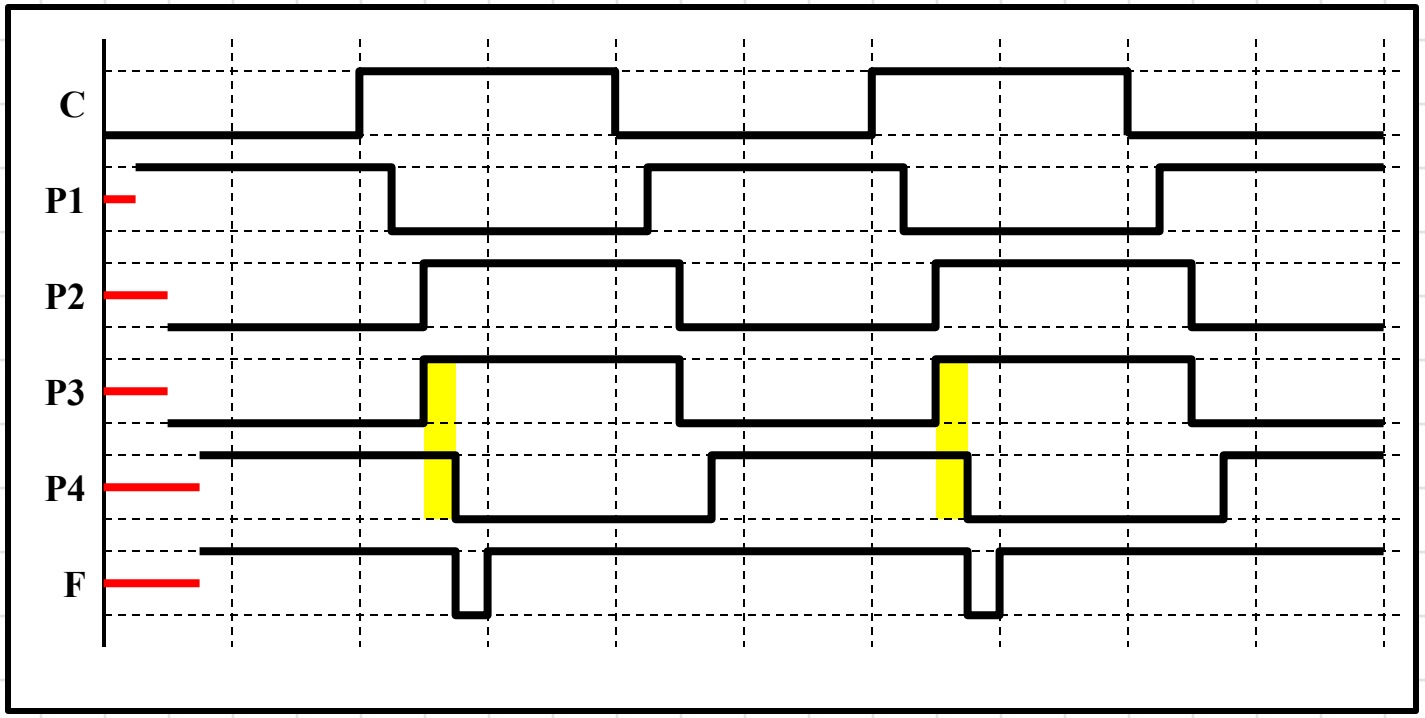

Signals
Waves
Time
A
B
C
D
w1
w2
w3
w4
F
100 sec
200

e)

```
module prob203e;

  reg A, B, C, D;
  wire F;

  prob203b dut(A, B, C, D, F);

  initial begin
    $dumpfile("prob203e.vcd");
    $dumpvars;
    {A, B, C, D} = 4'b1011;
    #200 $finish;
  end

  always #50 B = ~B;

endmodule
```

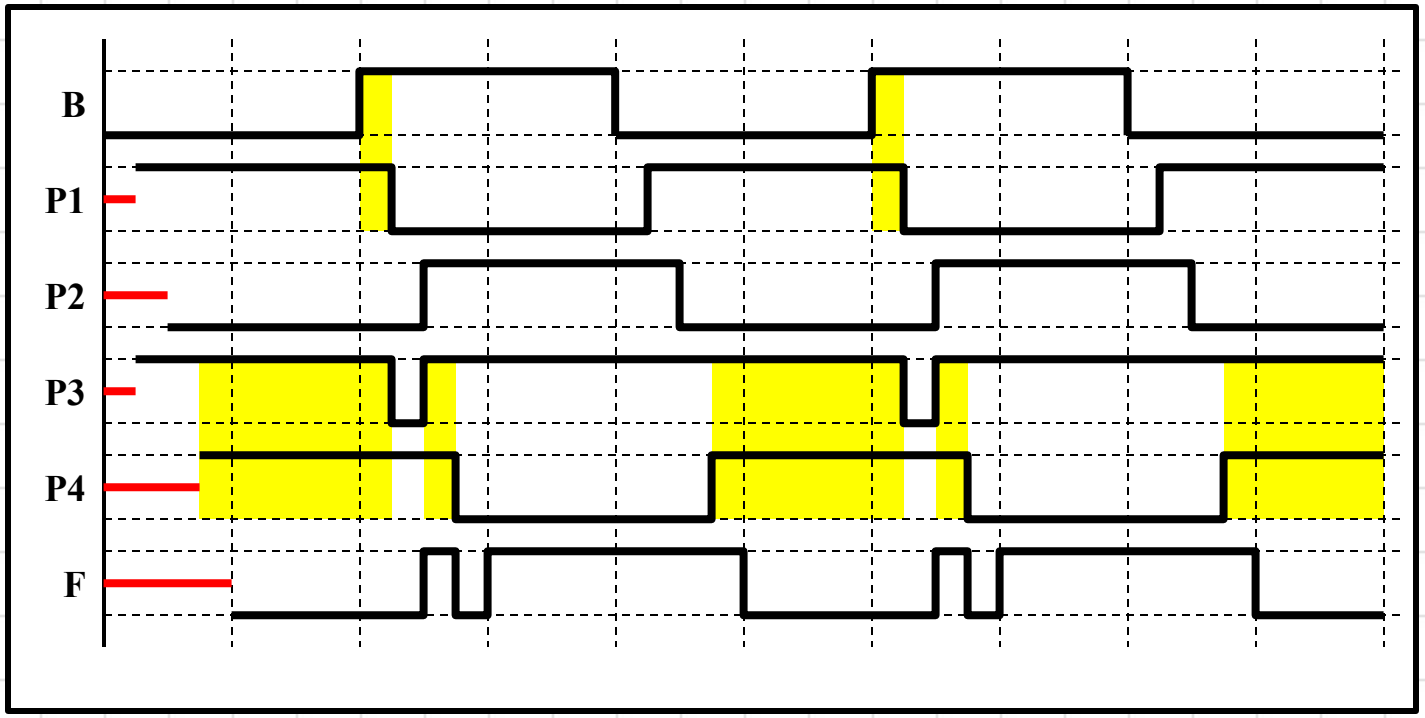

Signals
Waves
Time
A
B
C
D
w1
w2
w3
w4
F
100 sec
200

b) **A = B = D = 1**



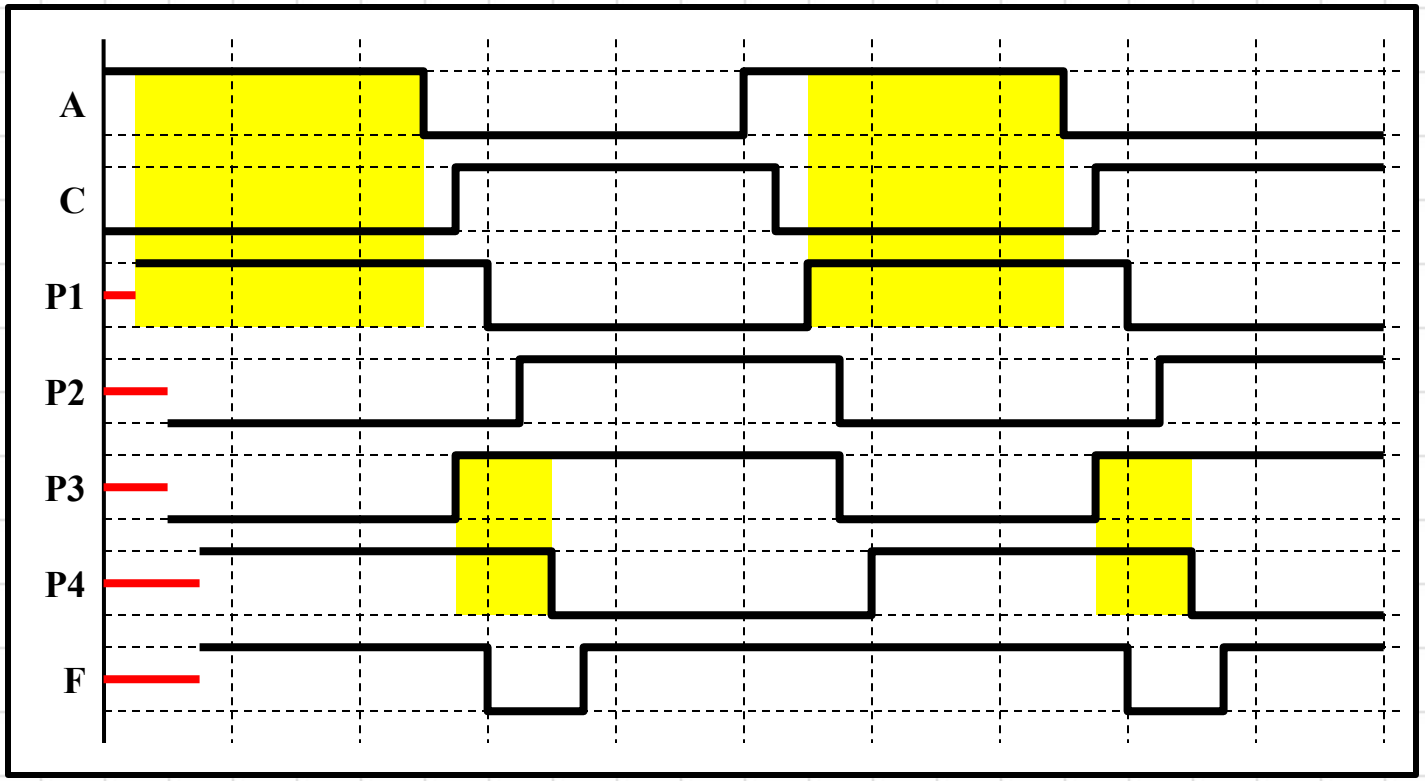
F = ABC’ + BCD = 1·1·C’ + 1·C·1 = C’ + C = 1

c) **A = C = D = 1**



F = ABC’ + BCD = 1·B·0 + B·1·1 = B

d) **B = D = 1**



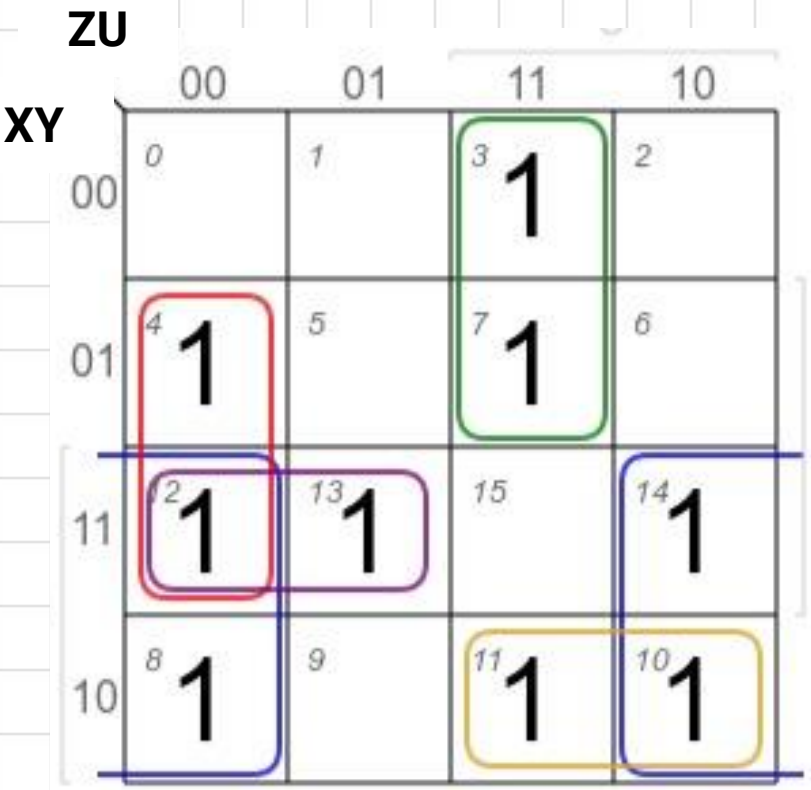
F = ABC’ + BCD = AC’ + C = A + C

Problema 2-04

Relaciones mínimas

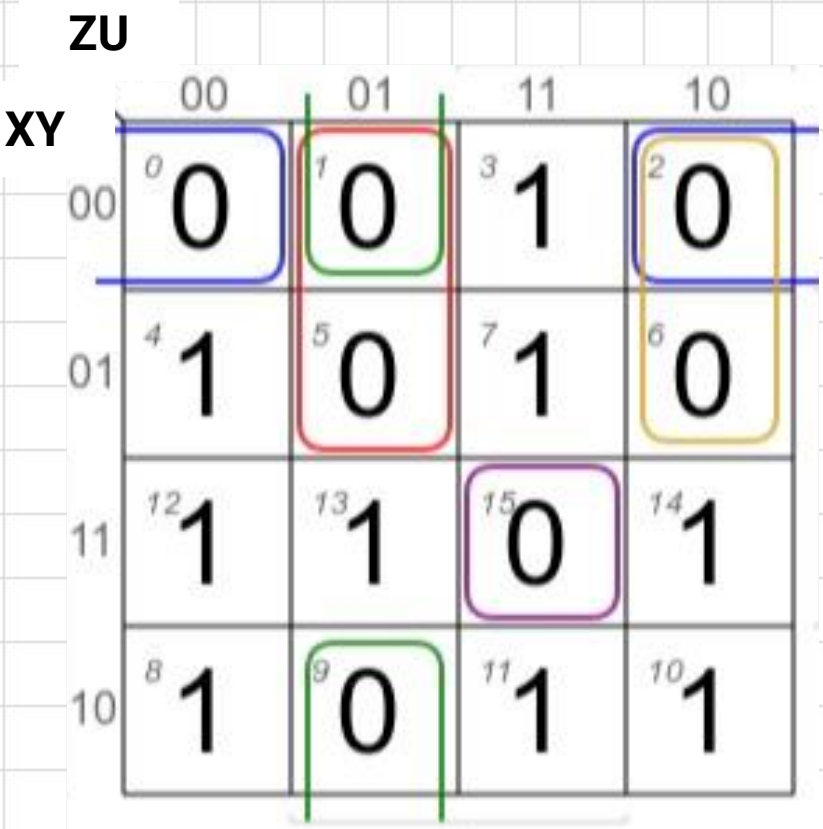
a)

Suma de Productos:



F = XU'+YZ'U'+X'ZU+XY'Z+XYZ'

Productos de sumas:



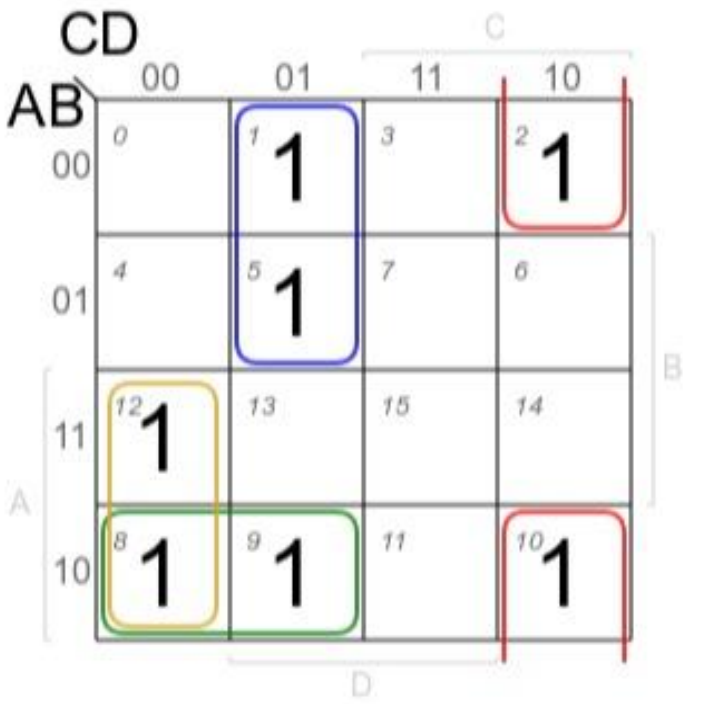
F = (X+Y+U)(X+Z+U')(Y+Z+U')·
(X+Z'+U)(X'+Y'+Z'+U')

.

f)

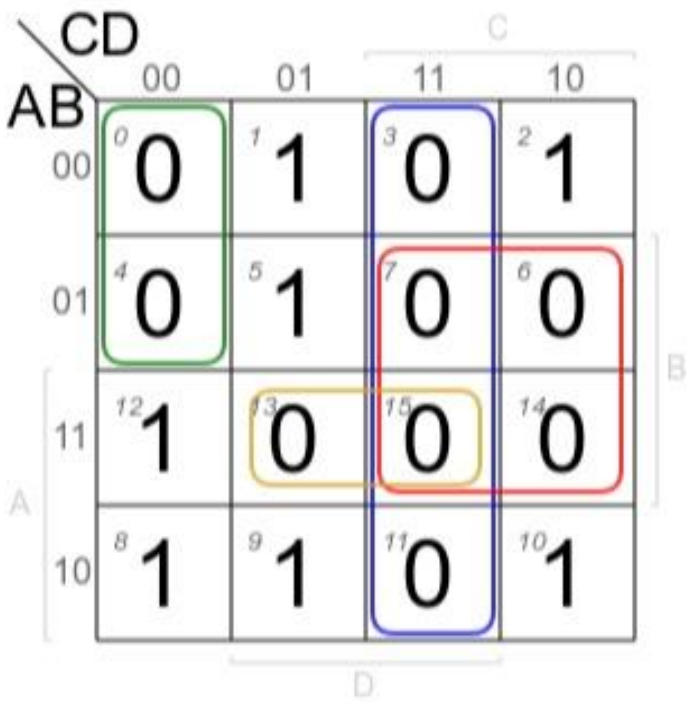
Suma de Productos:

F = A'C'D + B'CD' + AB'C' + AC'D'



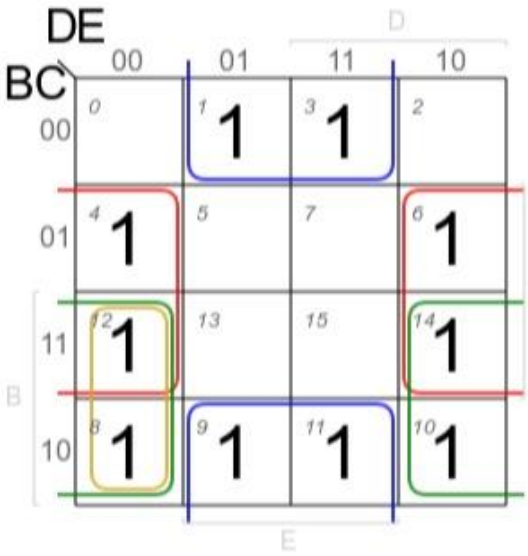
Productos de sumas:

F = (C' + D')(B' + C')(A + C + D)(A' + B' + D')

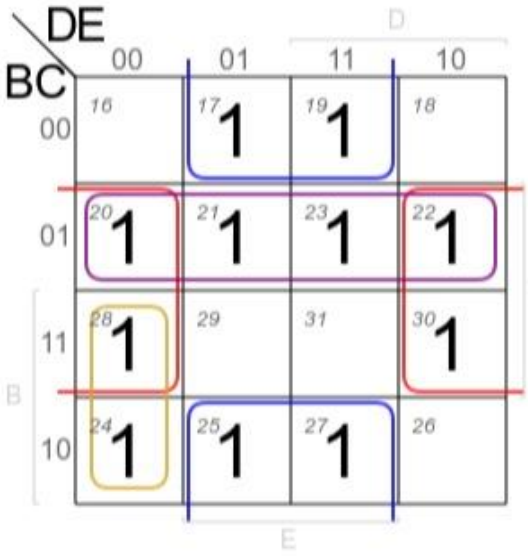


g) Suma de Productos:

F = C'E + CE' + A'BE' + BD'E' + AB'C



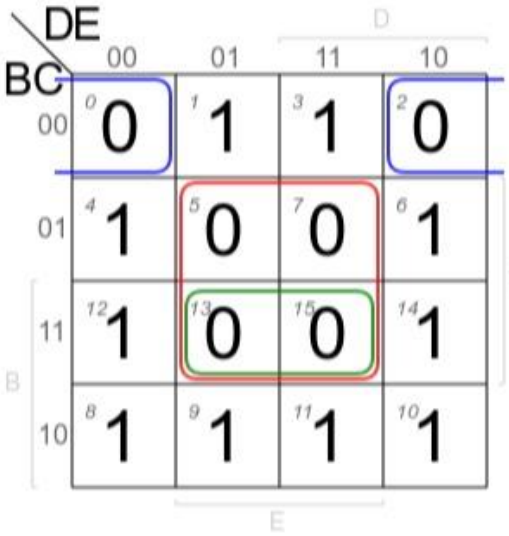
A = 0



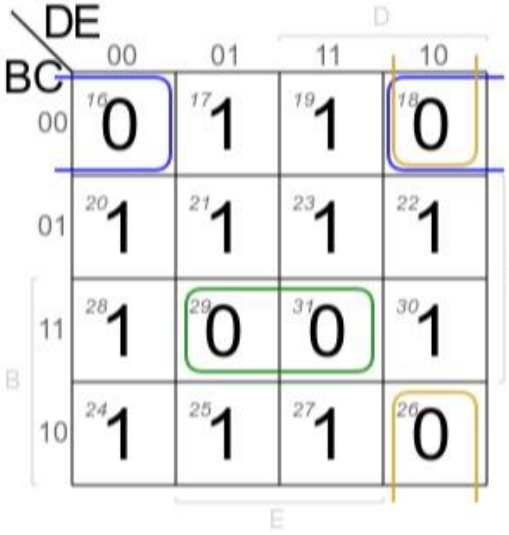
A = 1

Productos de sumas:

F = (B + C + E)(A + C' + E')(B' + C' + E')(A' + C + D' + E)



A = 0



A = 1

Problema 2-05

Valor absoluto

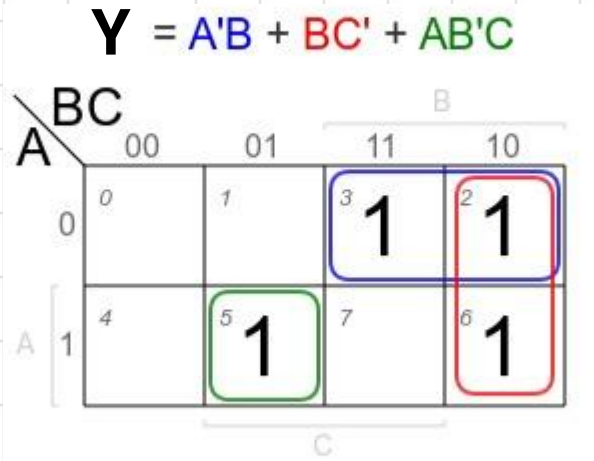
a)

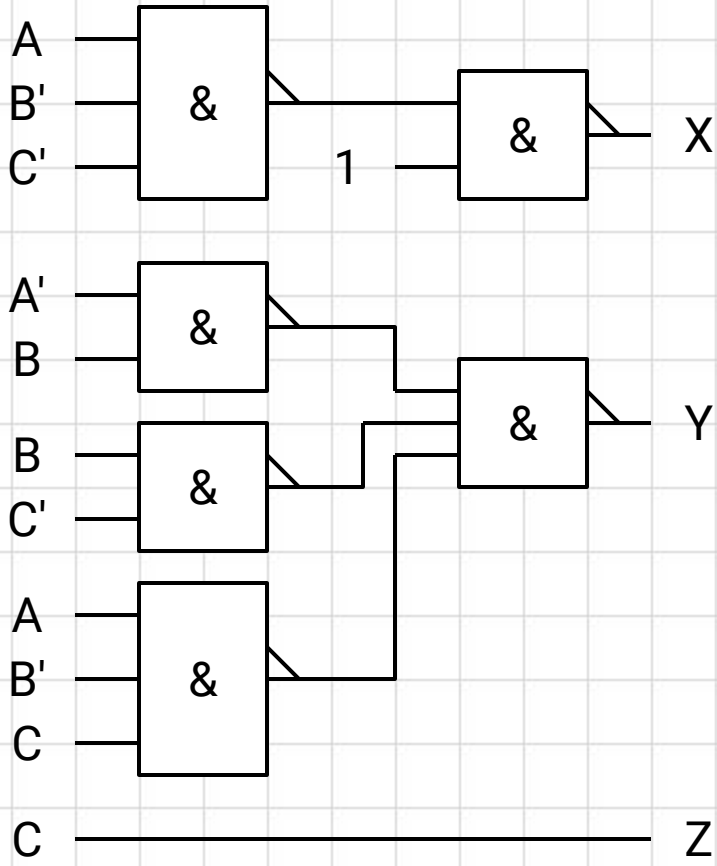
ABC	Valor	XYZ	Valor
000	0	000	0
001	1	001	1
010	2	010	2
011	3	011	3
100	-4	100	4
101	-3	011	3
110	-2	010	2
111	-1	001	1

$X = \sum(4) = AB'C'$

$Y = \sum(2,3,5,6)$

$Z = C$





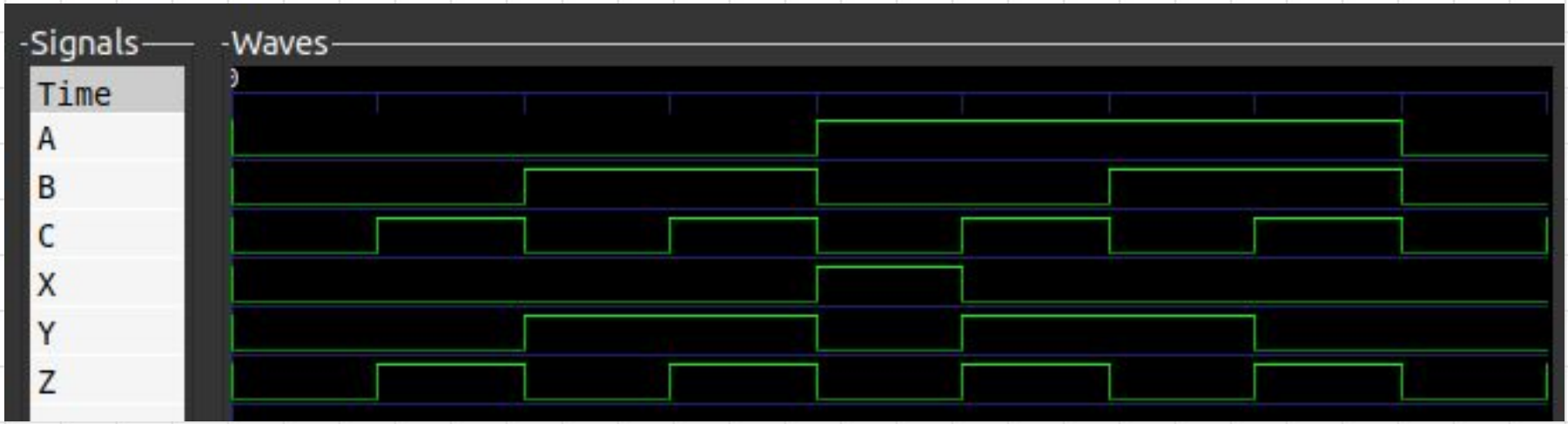
b)

```
module prob205b(input A, B, C, output X, Y, Z);

  assign X = A & ~B & ~C;
  assign Y = ~A & B | B & ~C | A & ~B & C;
  assign Z = C;

endmodule
```

c)

```
module prob205c;

  reg A, B, C;
  wire X, Y, Z;

  prob205b dut(A, B, C, X, Y, Z);

  initial begin
    $dumpfile("prob205c.vcd");
    $dumpvars;
    {A, B, C} = 0;
    #90 $finish;
  end

  always #10 {A, B, C} = {A, B, C} + 1;

endmodule
```

0 1 2 3 -4 -3 -2 -1 0

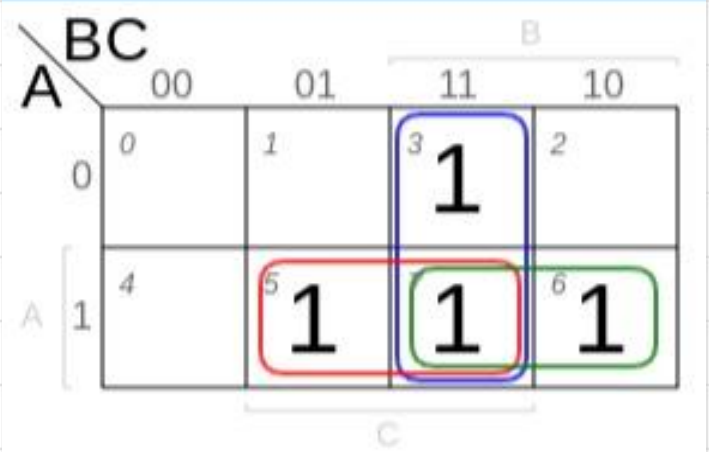


0 1 2 3 4 3 2 1 0

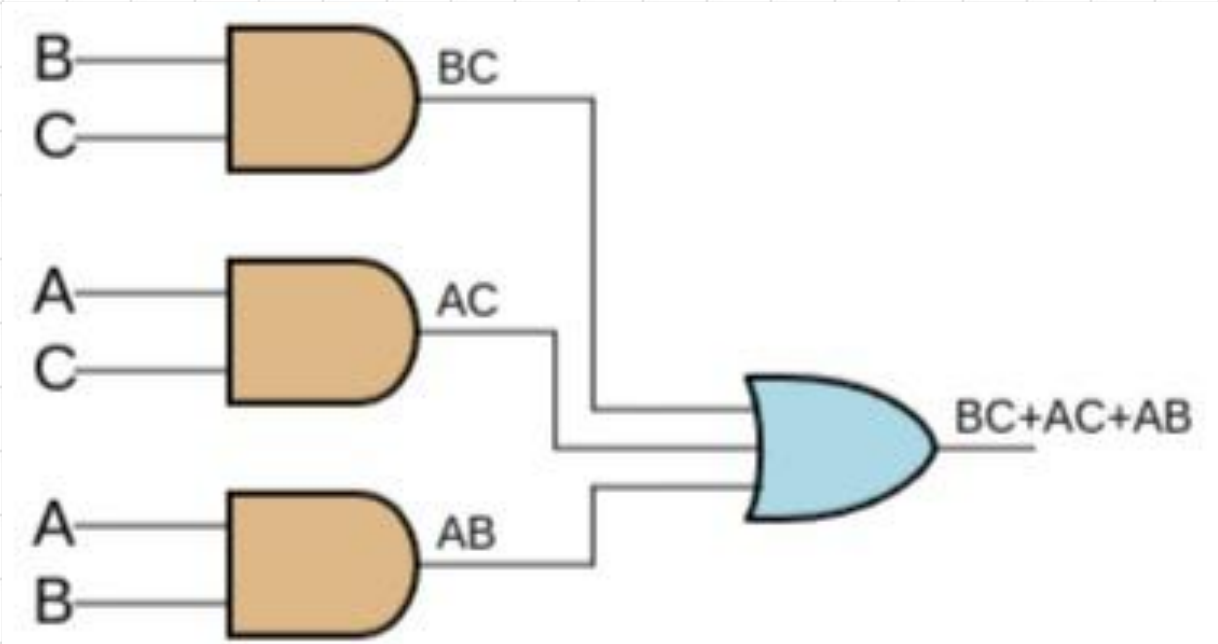
Problema 2-06

Circuito votador

a)



$$F = BC + AC + AB$$

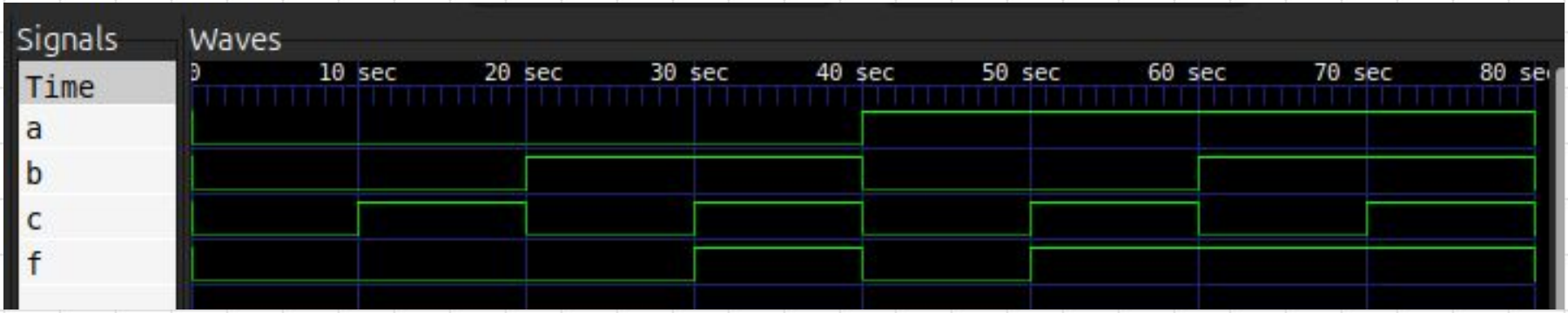


b)

```
module votador (
    input  a,
    input  b,
    input  c,
    output f
);
  assign f = b & c | a & c | a & b;
endmodule
```

c)

```
module votador_tb;
  reg a, b, c;
  wire f;
  votador dut (
      a,
      b,
      c,
      f
  );
  initial begin
    $dumpfile("sim.vcd");
    $dumpvars;
    {a, b, c} = 0;
    #80 $finish;
  end
  always #10 {a, b, c} = {a, b, c} + 1;
endmodule
```

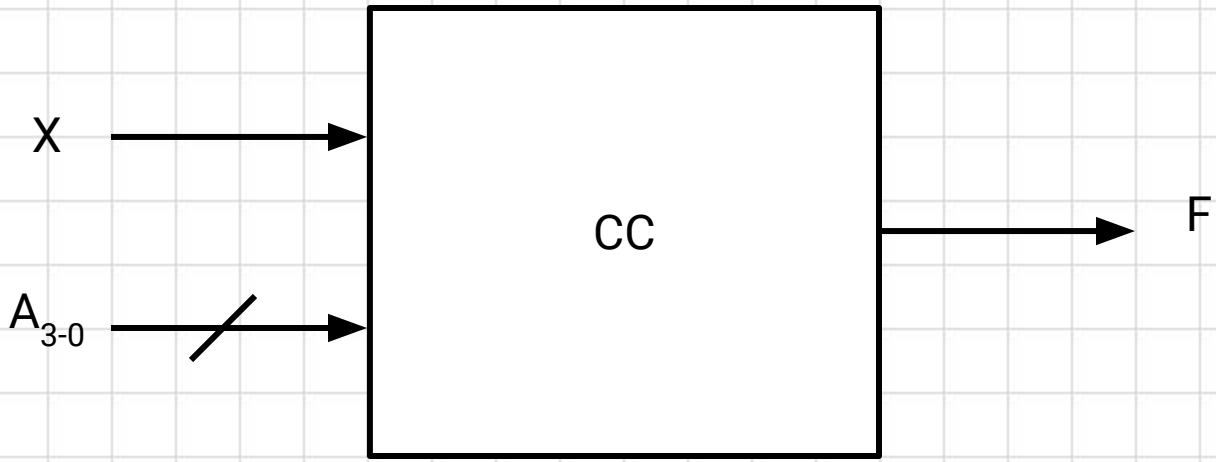

Signals
Waves
Time
a
b
c
f
0
10 sec
20 sec
30 sec
40 sec
50 sec
60 sec
70 sec
80 se

Problema 2-07

Función a partir de BCD

AMC

a)



Posición	X A	F	Posición	X A	F
0	0 0000	0	16	1 0000	0
1	0 0001	1	17	1 0001	0
2	0 0010	1	18	1 0010	0
3	0 0011	0	19	1 0011	1
4	0 0100	1	20	1 0100	0
5	0 0101	0	21	1 0101	0
6	0 0110	0	22	1 0110	1
7	0 0111	1	23	1 0111	0
8	0 1000	1	24	1 1000	0
9	0 1001	0	25	1 1001	1

$$F = \sum(1,2,4,7,8,19,22,25) + d(10,11,12,13,14,15,26,27,28,29,30,31)$$

$$F = \sum(1,2,4,7,8,19,22,25) + d(10,11,12,13,14,15,26,27,28,29,30,31)$$

F	A2,A1,A0							
	000	001	011	010	110	111	101	100
X,A3 00	0	1	0	1	0	1	0	1
01	1	0	-	-	-	-	-	-
11	0	1	-	-	-	-	-	-
10	0	0	1	0	1	0	0	0

$$F(X, A3, A2, A1, A0) = X'A3'A2'A1'A0 + X'A2'A1A0' + X'A2A1'A0' + X'A2A1A0 + X'A3A0' + XA2'A1A0 + XA2A1A0' + XA3A0$$

F(X, A3, A2, A1, A0) =

X'A3'A2'A1'A0 +

X'A2'A1A0' +

X'A2A1'A0' +

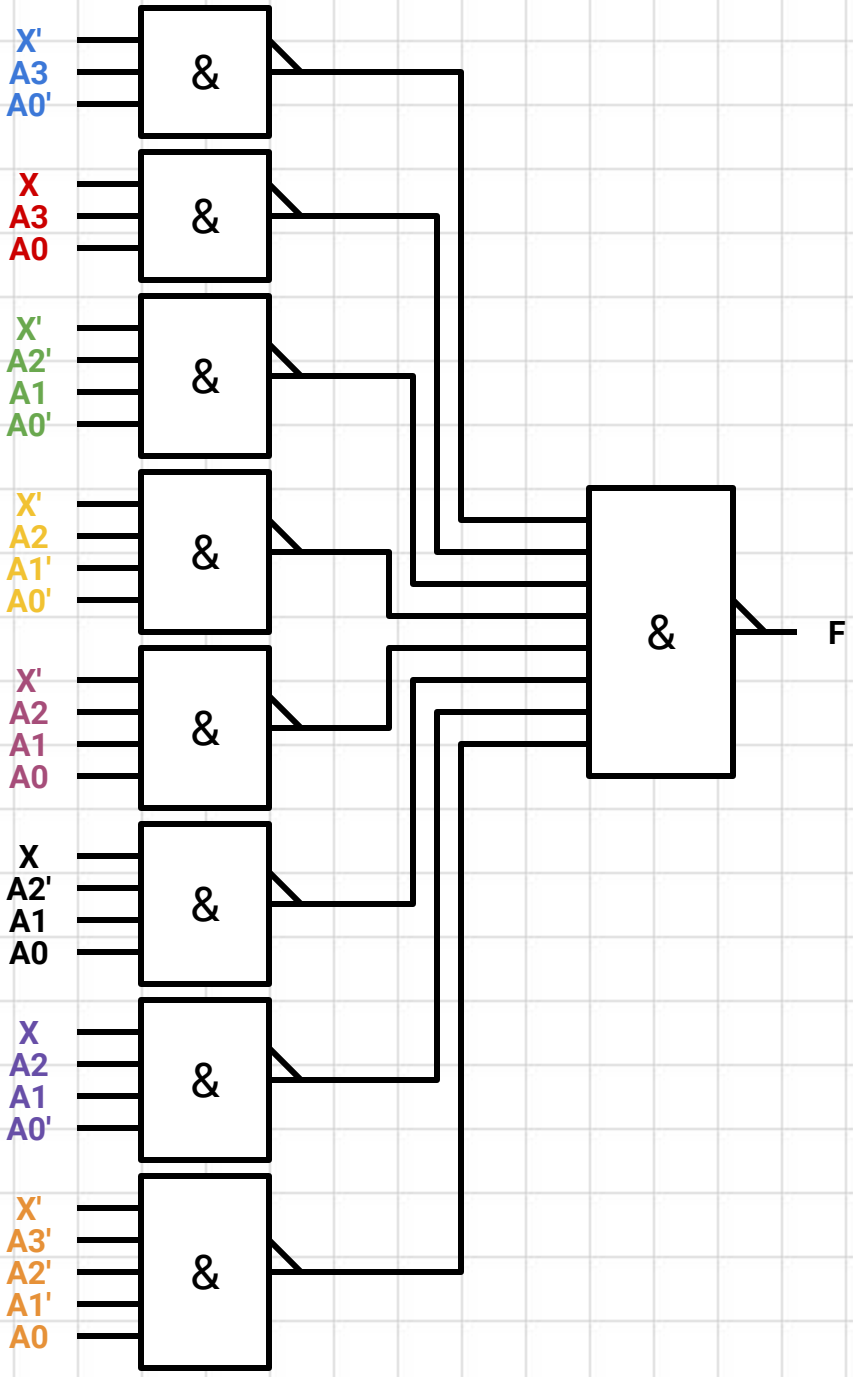
X'A2A1A0 +

X'A3A0' +

XA2'A1A0 +

XA2A1A0' +

XA3A0

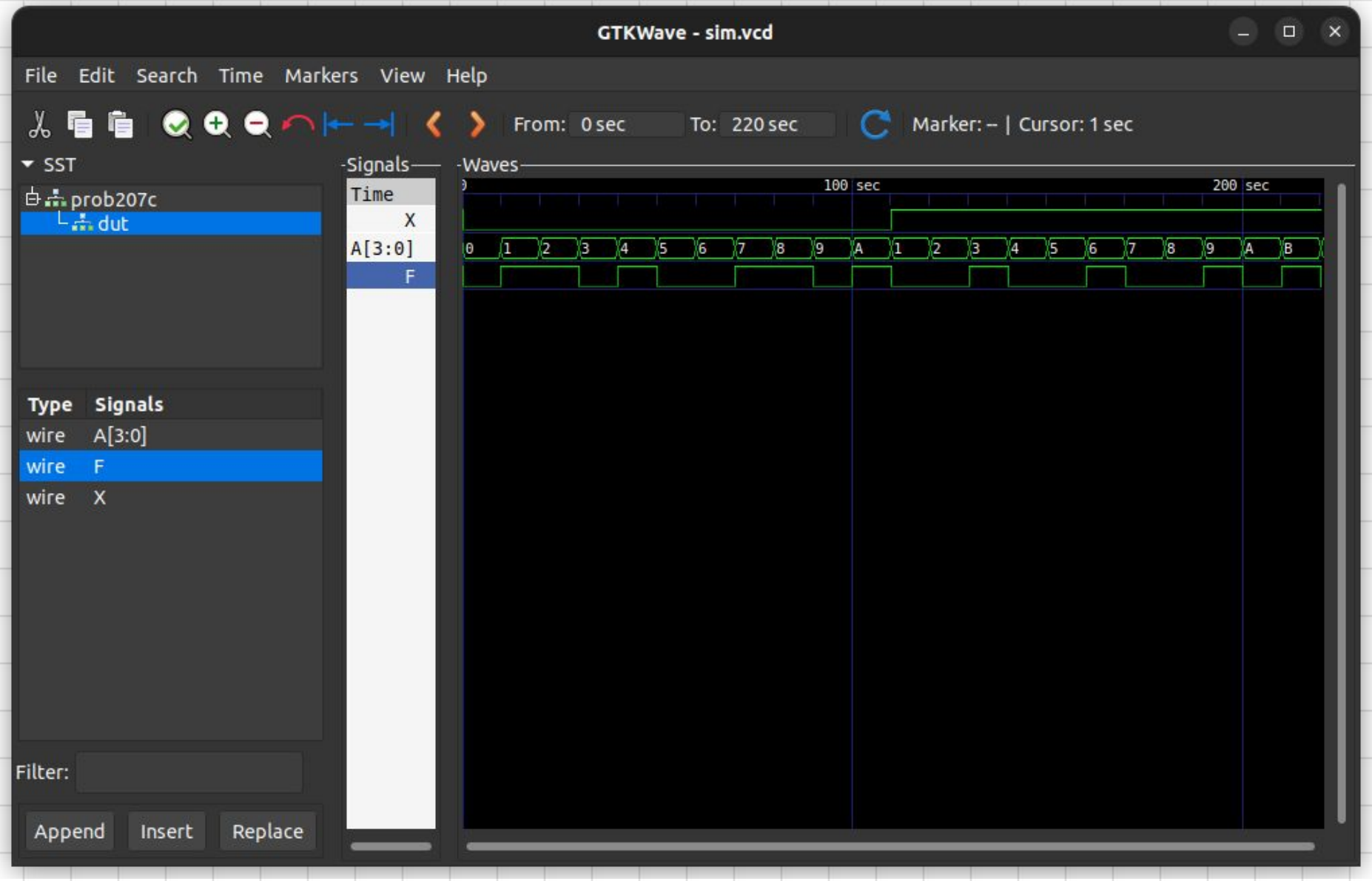


b)

```
module prob207b(input X, input [3:0] A, output F);

  assign F = (~X & ~A[3] & ~A[2] & ~A[1] & A[0])
           | (~X & ~A[2] &  A[1] & ~A[0]       )
           | (~X &  A[2] & ~A[1] & ~A[0]       )
           | (~X &  A[2] &  A[1] &  A[0]       )
           | (~X &  A[3] & ~A[0]               )
           | ( X & ~A[2] &  A[1] &  A[0]       )
           | ( X &  A[2] &  A[1] & ~A[0]       )
           | ( X &  A[3] &  A[0]               );

endmodule
```

c)

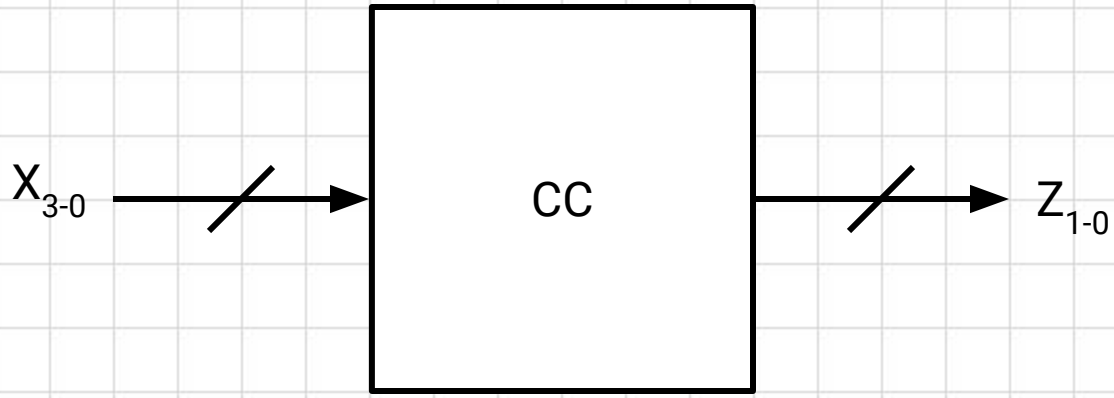
```
module prob207c;

  reg X;
  reg [3:0] A;
  wire F;

  prob207b dut(X, A, F);

  initial begin
    $dumpfile("sim.vcd"); $dumpvars;
    X = 0; A = 0;
    #110 X = 1; A = 0;
    #110 $finish;
  end
  always #10 A = A + 1;

endmodule
```


GTKWave - sim.vcd
File Edit Search Time Markers View Help
From: 0 sec
To: 220 sec
Marker: – | Cursor: 1 sec
SST
prob207c
dut
Type Signals
wire A[3:0]
wire F
wire X
Filter:
Append Insert Replace
Signals
Time
X
A[3:0]
F
Waves
100 sec
200 sec
0 1 2 3 4 5 6 7 8 9 A 1 2 3 4 5 6 7 8 9 A B

Problema 2-08

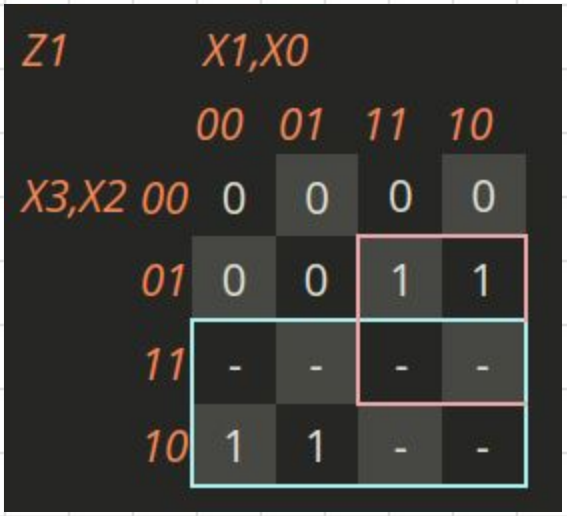
División BCD

a) Interfaz:

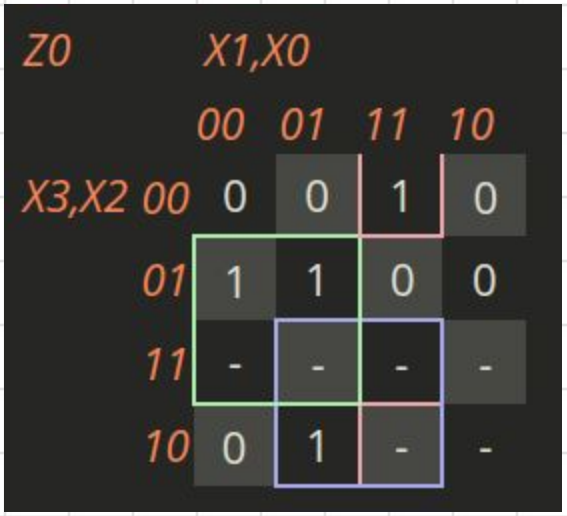


X	X3X2X1X0	Z	Z1Z0
0	0000	0	00
1	0001	0	00
2	0010	0	00
3	0011	1	01
4	0100	1	01
5	0101	1	01
6	0110	2	10
7	0111	2	10
8	1000	2	10
9	1001	3	11
	en otro caso		**

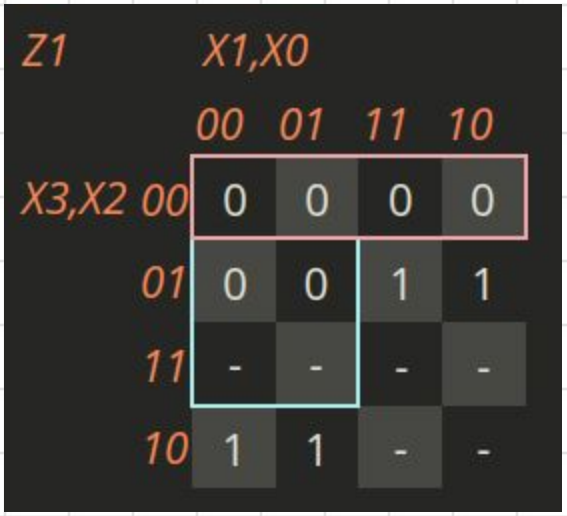
$Z1 = \sum(6,7,8,9) + d(10,11,12,13,14,15)$
$Z0 = \sum(3,4,5,9) + d(10,11,12,13,14,15)$



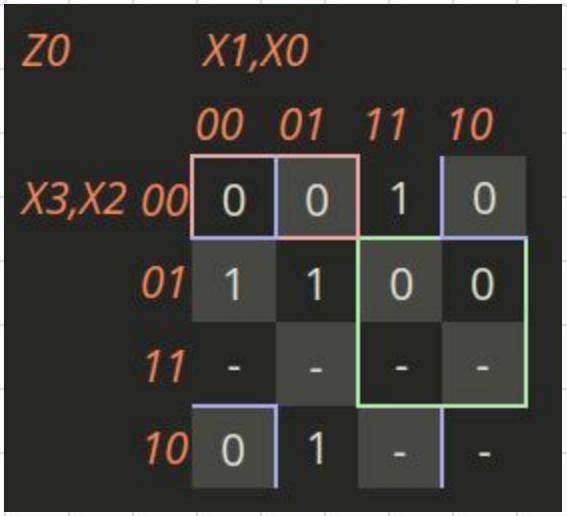
Z1 = X2X1 + X3



Z0 = X2'X1X0 + X2X1' + X3X0



Z1 = (X3 + X2)(X2' + X1)



Z0 = (X3 + X2 + X1)·
(X2' + X1')(X2 + X0)

b)

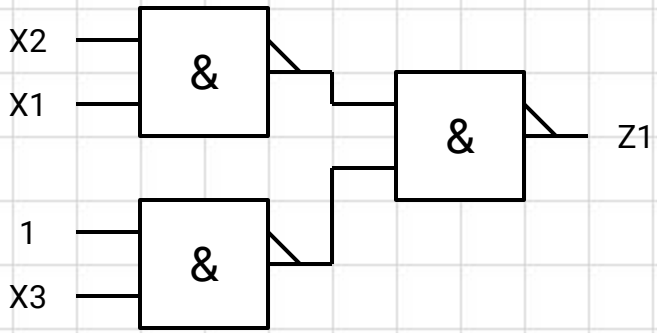
Z1 = X2X1 + X3
= [X2X1 + X3]"
= **[(X2X1)'(1·X3)']'**

Z0 = X2'X1X0 + X2X1' + X3X0
= [X2'X1X0 + X2X1' + X3X0]"
= **[(X2'X1X0)'(X2X1')'(X3X0)']'**

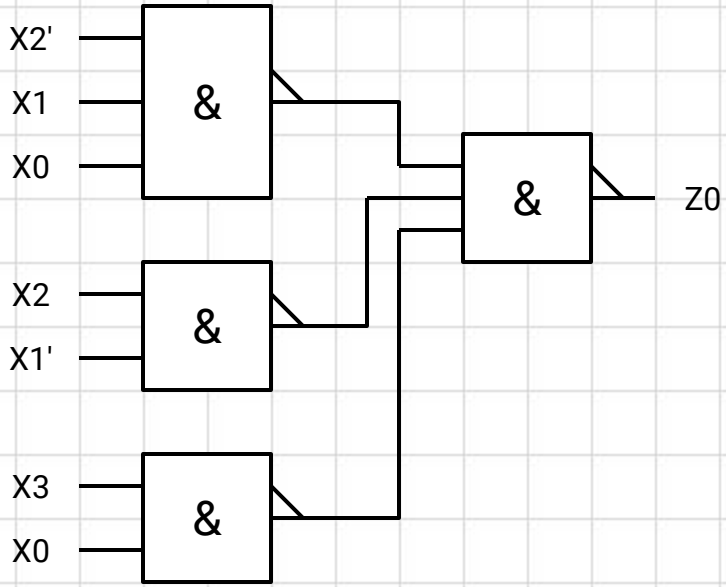
Z1 = (X3 + X2)(X2' + X1)
= [(X3 + X2)(X2' + X1)]"
= [(X3 + X2)' + (X2' + X1)']'

Z0 = (X3 + X2 + X1)·
(X2' + X1')(X2 + X0)
= [(X3 + X2 + X1)·
(X2' + X1')(X2 + X0)]"
= [(X3 + X2 + X1)' +
(X2' + X1')' + (X2 + X0)']'

Z1 = [(X2X1)'(1·X3)']'



Z0 = [(X2'X1X0)'(X2X1')'(X3X0)']'

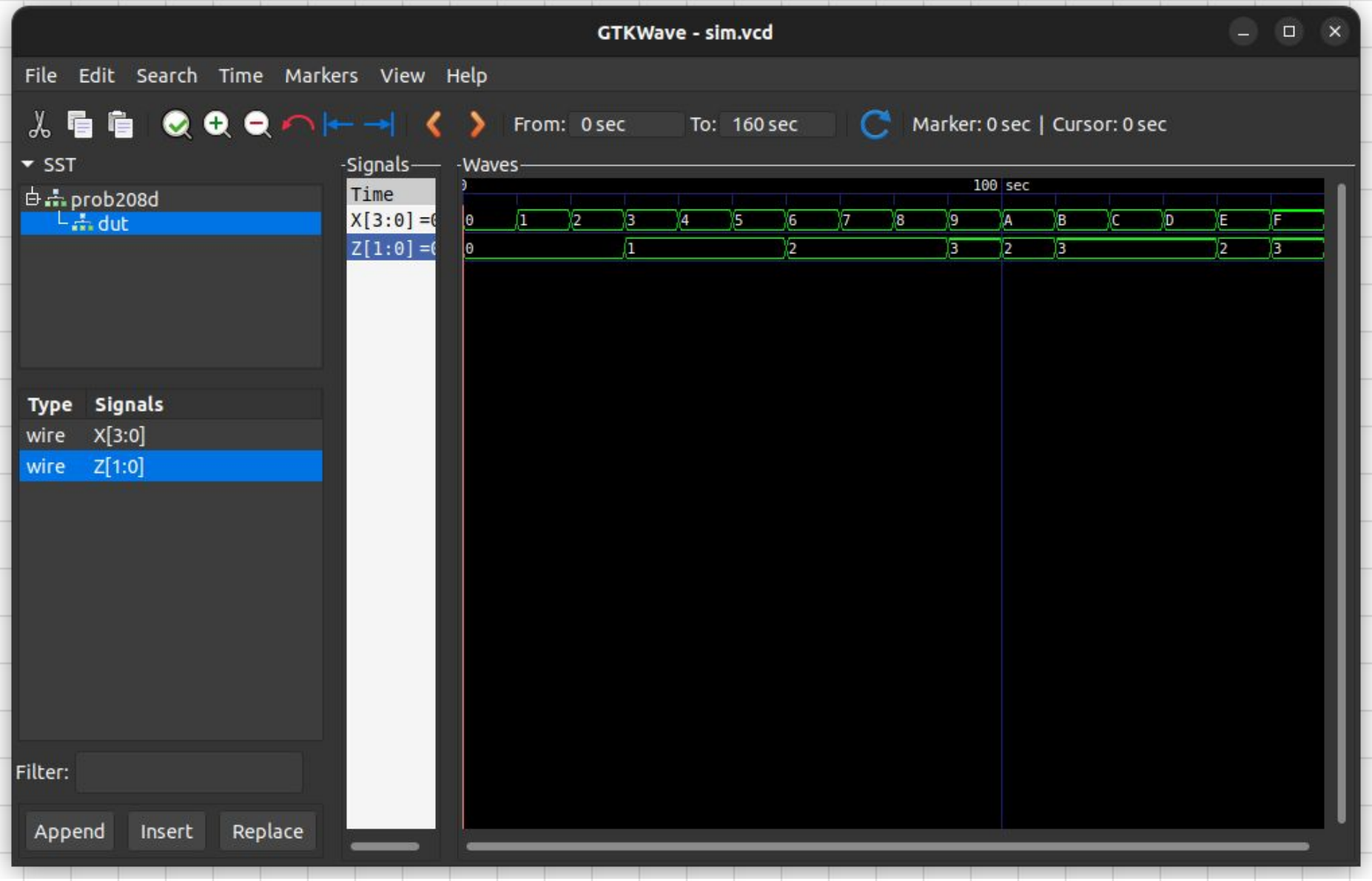


c)

```
module prob208c(input [3:0] X, output [1:0] Z);

  assign Z[1] = (X[2] & X[1]) | (X[3]);
  assign Z[0] = (~X[2] & X[1] & X[0])
              | (X[2] & ~X[1]) | (X[3] & X[0]);

endmodule
```

d)

```
module prob208d;

  reg [3:0] X;
  wire [1:0] Z;

  prob208c dut(X, Z);

  initial begin
    $dumpfile("sim.vcd"); $dumpvars;
    X = 0;
    #160 $finish;
  end

  always #10 X = X + 1;

endmodule
```

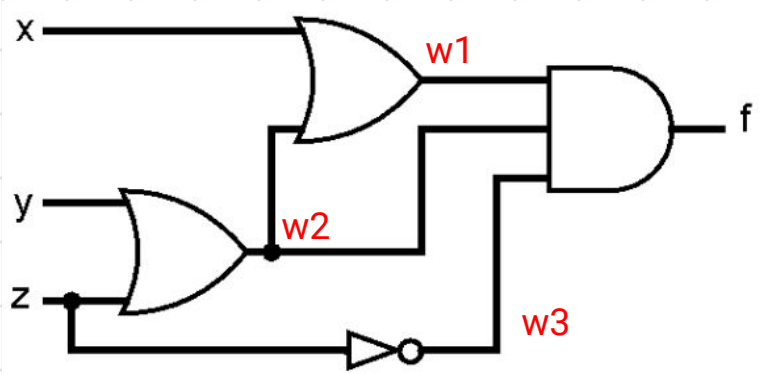

GTKWave - sim.vcd
File Edit Search Time Markers View Help
From: 0 sec
To: 160 sec
Marker: 0 sec | Cursor: 0 sec
SST
prob208d
dut
Type Signals
wire X[3:0]
wire Z[1:0]
Filter:
Append Insert Replace
Signals
Time
X[3:0] =
Z[1:0] =
Waves
100 sec
0 1 2 3 4 5 6 7 8 9 A B C D E F
0 1 2 3 2 3 2 3

Problema 2-09

Descripción estructural

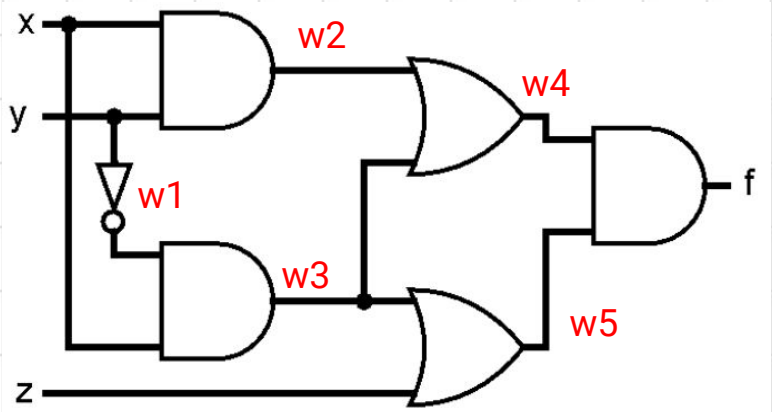
a)

```
module prob209a(input x, y, z, output f);
  wire w1, w2, w3;
  or p1(w1, x, w2);
  or p2(w2, y, z);
  not p3(w3, z);
  and pF(f, w1, w2, w3);
endmodule
```



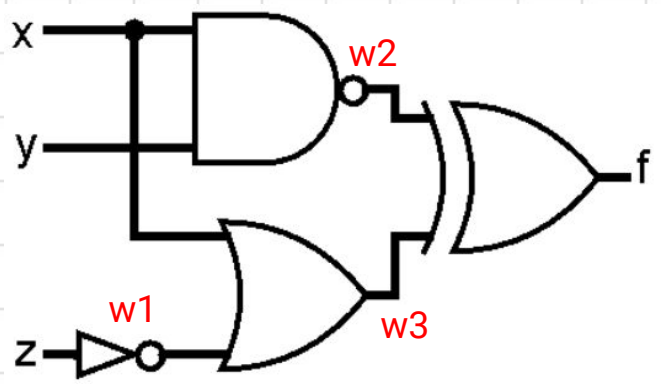
b)

```
module prob209b(input x, y, z, output f);
  wire w1, w2, w3, w4, w5;
  not p1(w1, y);
  and p2(w2, x, y);
  and p3(w3, w1, x);
  or p4(w4, w2, w3);
  and pF(f, w4, w5);
endmodule
```



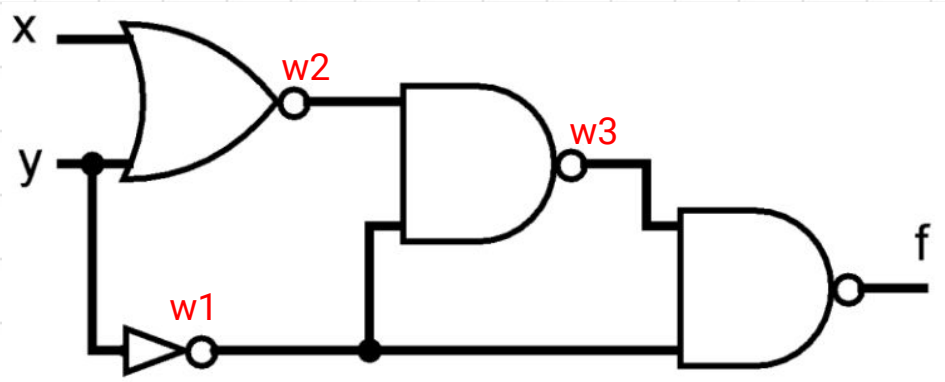
c)

```
module prob209c(input x, y, z, output f);
  wire w1, w2, w3;
  not p1(w1, z);
  nand p2(w2, x, y);
  or p3(w3, x, w1);
  xor pF(f, w2, w3);
endmodule
```



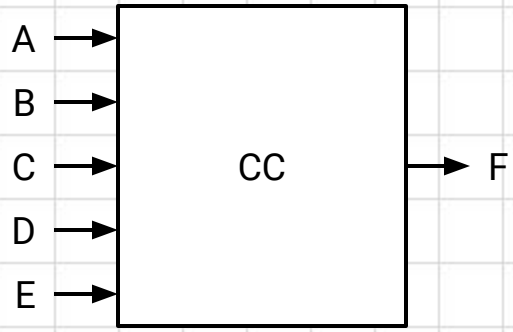
d)

```
module prob209d(input x, y, z, output f);
  wire w1, w2, w3;
  not p1(w1, y);
  nor p2(w2, x, y);
  nand p3(w3, w2, w1);
  nand pF(f, w3, w1);
endmodule
```



Problema 2-18

Directiva de fútbol



F	C,D,E							
	000	001	011	010	110	111	101	100
A,B 00	0	1	0	1	0	1	0	1
01	0	1	0	1	1	0	0	1
11	1	0	1	0	1	0	1	0
10	0	1	1	0	1	0	1	0

F = A'C'DE' + A'CD'E' + A'B'CDE + A'C'D'E + ACDE' + ABC'D'E' + AC'DE + ACD'E + A'BDE' + B'C'D'E

Problema 2-20

BCD x5

AMC

A3-A0	X3-X0	Y3-Y0
0000	0000	0000
0001	0000	0101
0010	0001	0000
0011	0001	0101
0100	0010	0000
0101	0010	0101
0110	0011	0000
0111	0011	0101
1000	0100	0000
1001	0100	0101
en otro caso	----	----

X3 = 0

X2 = A3

X1 = A2

X0 = A1

Y3 = 0

Y2 = A0

Y1 = 0

Y0 = A0