

(-2) 1. soru

a)  $(x+y+z)(xy)' = x'y'z' + x+y$

$$\overline{(x+y+z)} + x'y' = \underbrace{x'y'z'}_{\text{Absorption law}} + x'y' = x' + y'$$

$$x'y' \neq \underbrace{x'y'z' + x'y}_{x'y+z'}$$

	x	y	z	f
0	0	0	0	1
1	0	0	1	1
2	0	1	0	1
3	0	1	1	1
4	1	0	0	1
5	1	0	1	1
6	1	1	0	0
7	1	1	1	0

$\neq$

	x	y	z	f
0	0	0	0	1
1	0	0	1	0
2	0	1	0	1
3	0	1	1	1
4	1	0	0	1
5	1	0	1	1
6	1	1	0	1
7	1	1	1	1

Esit değil

b)  $x + x'y'z = (x+y)(x+z)$

Geri

	x	y	z	f = x + x'y'z
0	0	0	0	0
1	0	0	1	0
2	0	1	0	0
3	0	1	1	1
4	1	0	0	1
5	1	0	1	1
6	1	1	0	1
7	1	1	1	1

=

	x	y	z	f = (x+y)(x+z)
0	0	0	0	0
1	0	0	1	0
2	0	1	0	0
3	0	1	1	1
4	1	0	0	1
5	1	0	1	1
6	1	1	0	1
7	1	1	1	1

Esit

c)  $x(x'y+z) = xy+xz$

$xx' + xy + xz = xy+xz$   
 $xy+xz = xy+xz$  ✓  
 $x(y+z)$

	x	y	z	f = x(y'+yz)
0	0	0	0	0
1	0	0	1	0
2	0	1	0	0
3	0	1	1	0
4	1	0	0	0
5	1	0	1	1
6	1	1	0	1
7	1	1	1	1

Esit

	x	y	z	f = xy+xz
0	0	0	0	0
1	0	0	1	0
2	0	1	0	0
3	0	1	1	0
4	1	0	0	0
5	1	0	1	1
6	1	1	0	1
7	1	1	1	1

2-)

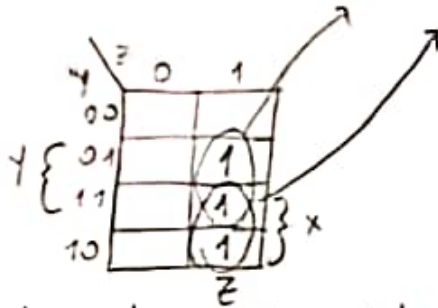
Yigit Bekir Gürsay

040180063

Yazın

$$a) ABC + A'B + ABC' = AB(C+C') + A'B = AB + A'B = B$$

$$b) x'yz + xz = yz + xz = z(y+x) \rightarrow \text{min number of literal } 3$$



$$c) (x+y)'(x'+y') = x'y'(x'+y') = x'y' + x'y = x'y'$$

$$d) xy + x(wz + wz') = xy + x(w(z+z')) = xy + xw = x(w+y)$$

$$e) (BC' + A'D)(AB' + CD) = \text{Min. number of literal} = 3$$

$$ABBC' + BCC'D + AA'B'D + A'CD' = 0$$

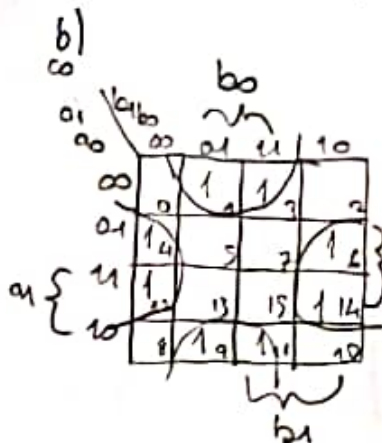
$$f) (a'+c')(a+b'+c') = \underbrace{aa'}_0 + \underbrace{ab'}_{a'b'} + \underbrace{ac'}_{c'} + \underbrace{bc'}_{c'} + \underbrace{b'c'}_{c'} + \underbrace{c'c'}_0 = a'b' + c'$$

3-)

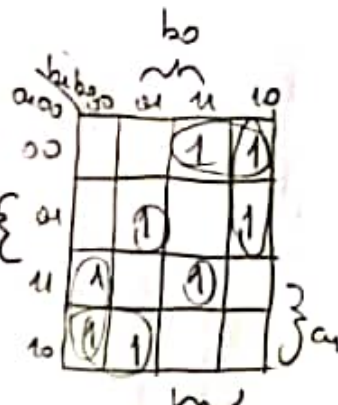
$$a) G = f_m(1, 3, 4, 6, 9, 11, 12, 14) = a_1'a_0'b_1'b_0 + a_1'a_0'b_1b_0 + a_1'a_0b_1'b_0 + a_1'a_0b_1b_0 + a_1a_0'b_1'b_0 + a_1a_0'b_1b_0 + a_1a_0b_1'b_0 + a_1a_0b_1b_0$$

$$G = f_m(2, 3, 5, 6, 8, 9, 12, 15) = a_1'a_0'b_1'b_0 + a_1'a_0'b_1b_0 + a_1'a_0b_1'b_0 + a_1'a_0b_1b_0 + a_1a_0'b_1'b_0 + a_1a_0'b_1b_0 + a_1a_0b_1'b_0 + a_1a_0b_1b_0$$

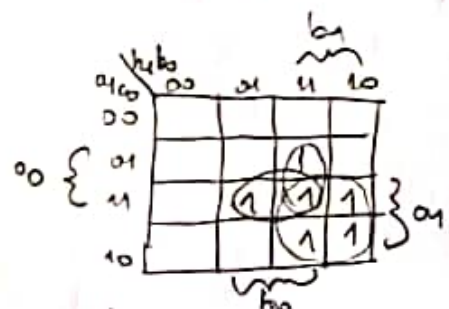
$$G_2 = f_m(7, 10, 11, 13, 14, 15) = a_1'a_0b_1b_0 + a_1a_0'b_1'b_0 + a_1a_0'b_1b_0 + a_1a_0b_1'b_0 + a_1a_0b_1b_0 + a_1a_0b_1b_0$$



$$a = a_0b_0' + a_0'b_0$$



$$G = a_1b_1b_0 + a_1\bar{a}_0b_1\bar{a}_1\bar{a}_0b_1 + a_1b_1b_0 + a_1a_0b_1b_0 + a_1a_0b_1b_0$$



$$G_2 = a_1b_1 + a_1a_0b_0 + a_1a_0b_1b_0$$

- Ödevde açıklandığı gibi proje oluşturuldu.
- Ninovadaki eklenen dosyayı indirip ilk olarak “Case Statement” yapısını kullandım.
- Case Statement kullandığım dosyanın adını “Boolean\_Function\_Case\_Statement” olarak adlandırdım.
- Ödevde söylenildiği gibi inputları ve outputlarımı tanımladım.

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity Boolean_Function_Case_Statement is

    Port ( a1 : in  STD_LOGIC;
          a0 : in  STD_LOGIC;
          b1 : in  STD_LOGIC;
          b0 : in  STD_LOGIC;

          c2: out STD_LOGIC;
          c1 : out STD_LOGIC;
          c0 : out STD_LOGIC);

end Boolean_Function_Case_Statement;
```

- Ardından örnek olarak verilen linklerden faydalanarak case struct oluşturdum. Kodum aşağıdaki gibidir.

```
architecture Behavioral of
Boolean_Function_Case_Statement is

begin
    process (a1, a0, b1, b0)
        variable input : std_logic_vector(3 downto
0);
    begin
        input := a1 & a0 & b1 & b0;
        case input is
            when "0000" =>
                c2 <= '0';
                c1 <= '0';
                c0 <= '0';

            when "0001" =>
                c2 <= '0';
                c1 <= '0';
                c0 <= '1';

            when "0010" =>
                c2 <= '0';
                c1 <= '1';
                c0 <= '0';

            when "0011" =>
                c2 <= '0';
                c1 <= '1';
                c0 <= '1';

            when "0100" =>
                c2 <= '0';
                c1 <= '0';
                c0 <= '1';

            when "0101" =>
                c2 <= '0';
                c1 <= '1';
                c0 <= '0';

            when "0110" =>
                c2 <= '0';
                c1 <= '1';
                c0 <= '1';
```

```
            when "0111" =>
                c2 <= '1';
                c1 <= '0';
                c0 <= '0';
            when "1000" =>
                c2 <= '0';
                c1 <= '1';
                c0 <= '0';
            when "1001" =>
                c2 <= '0';
                c1 <= '1';
                c0 <= '1';
            when "1010" =>
                c2 <= '1';
                c1 <= '0';
                c0 <= '0';
            when "1011" =>
                c2 <= '1';
                c1 <= '0';
                c0 <= '1';
            when "1100" =>
                c2 <= '0';
                c1 <= '1';
                c0 <= '1';
            when "1101" =>
                c2 <= '1';
                c1 <= '0';
                c0 <= '0';
            when "1110" =>
                c2 <= '1';
                c1 <= '0';
                c0 <= '1';
            when "1111" =>
                c2 <= '1';
                c1 <= '1';
                c0 <= '0';
            when others =>
                c2 <= 'U';
                c1 <= 'U';
                c0 <= 'U';
        end case;
    end process;
end Behavioral;
```

- Yukarıdaki kodun RTL şeması aşağıdaki gibi çıkmıştır. Her çıkış için , c2,c1 ve c0, ayrı bir ROM bloğu oluşturularak gerçekleştirilmiştir.

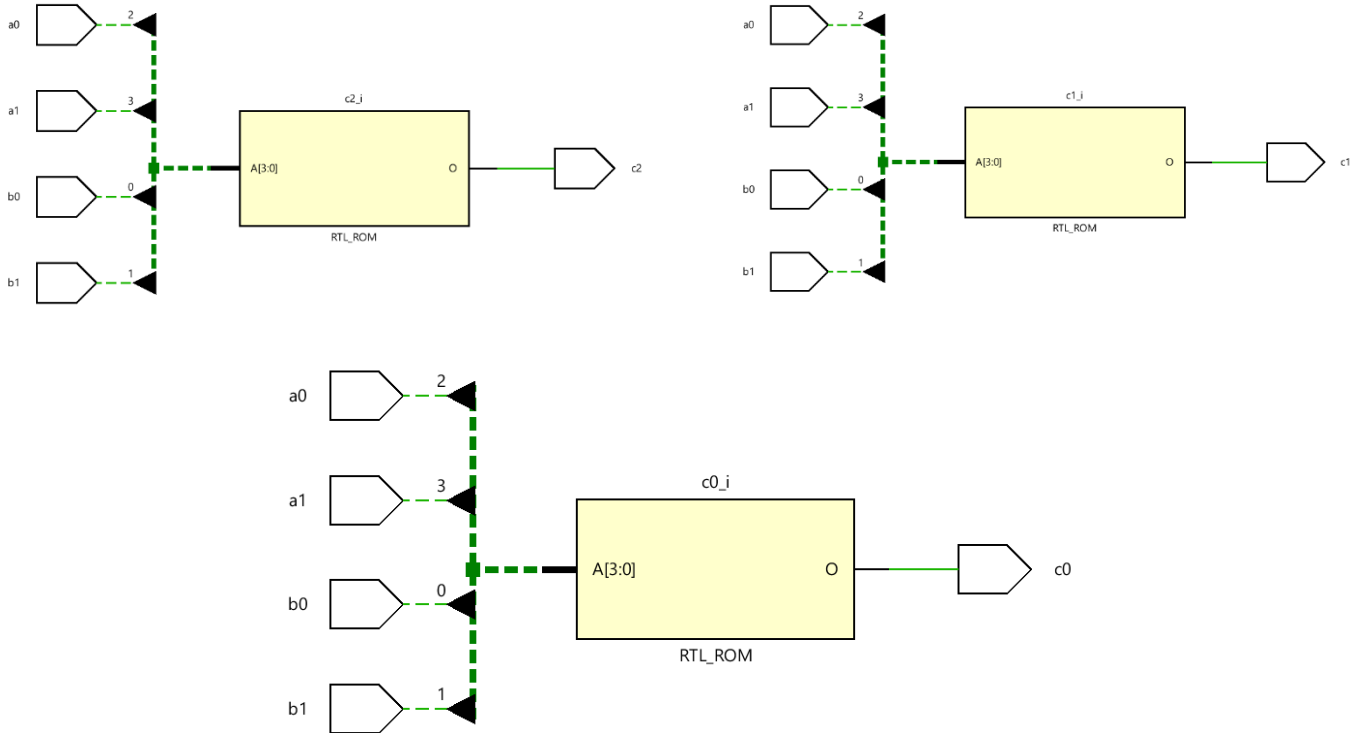
#### R Boolean\_Function\_Case\_Statement

##### ▼ Nets (7)

a0  
a1  
b0  
b1  
c0  
c1  
c2

##### ▼ Leaf Cells (3)

c0\_i (RTL\_ROM)  
c1\_i (RTL\_ROM)  
c2\_i (RTL\_ROM)



- Kodumuzun doğruluğunu kontrol etmek için testbench dosyası oluşturuyoruz. Yazılan kod ve simülasyon sonucu aşağıdadır.

```
--import std_logic from the IEEE library

library ieee;
use ieee.std_logic_1164.all;

--ENTITY DECLARATION: no inputs, no outputs
entity Boolean_Function_Case_Statement_tb is
end Boolean_Function_Case_Statement_tb;

-- Describe how to test the AND Gate
architecture tb of Boolean_Function_Case_Statement_tb is
--pass andGate entity to the testbench as component
component Boolean_Function_Case_Statement is
    Port ( a1      : in  STD_LOGIC ;
          a0      : in  STD_LOGIC ;
          b1      : in  STD_LOGIC ;
          b0      : in  STD_LOGIC ;

          c2      : out STD_LOGIC ;
          c1      : out STD_LOGIC ;
          c0      : out STD_LOGIC );
end component;

signal a1 : STD_LOGIC := '0';
signal a0 : STD_LOGIC := '0';
signal b1 : STD_LOGIC := '0';
signal b0 : STD_LOGIC := '0';
signal c2 : STD_LOGIC := '0';
signal c1 : STD_LOGIC := '0';
signal c0 : STD_LOGIC := '0';

constant period : time := 50ns;

begin

    uut: Boolean_Function_Case_Statement PORT MAP(
        a1 => a1,
        a0 => a0,
        b1 => b1,
        b0 => b0,
        c2 => c2,
        c1 => c1,
        c0 => c0
    );


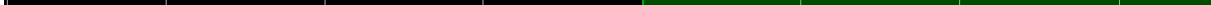

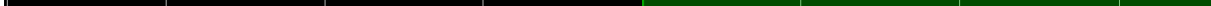



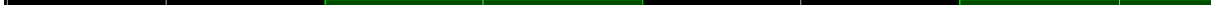









    stimulus : process
    begin

        -- 0000
        a1 <= '0';
        a0 <= '0';
        b1 <= '0';
        b0 <= '0';
        wait for period;
        -- 0001
        a1 <= '0';
        a0 <= '0';
        b1 <= '0';
        b0 <= '1';
        wait for period;
        -- 0010
        a1 <= '0';
        a0 <= '0';
        b1 <= '1';
        b0 <= '0';
        wait for period;
        -- 0011
        a1 <= '0';
        a0 <= '0';
        b1 <= '1';
        b0 <= '1';
        wait for period;
        -- 0100
        a1 <= '0';
        a0 <= '1';
        b1 <= '0';
        b0 <= '0';
```

```
wait for period;
        -- 0111
        a1 <= '0';
        a0 <= '1';
        b1 <= '1';
        b0 <= '1';
        wait for period;
        -- 1000
        a1 <= '1';
        a0 <= '0';
        b1 <= '0';
        b0 <= '0';
        wait for period;
        -- 1001
        a1 <= '1';
        a0 <= '0';
        b1 <= '0';
        b0 <= '1';
        wait for period;
        -- 1010
        a1 <= '1';
        a0 <= '0';
        b1 <= '1';
        b0 <= '0';
        wait for period;
        -- 1011
        a1 <= '1';
        a0 <= '0';
        b1 <= '1';
        b0 <= '1';
        wait for period;
        -- 1100
        a1 <= '1';
        a0 <= '1';
        b1 <= '0';
        b0 <= '0';
        wait for period;
        -- 1101
        a1 <= '1';
        a0 <= '1';
        b1 <= '0';
        b0 <= '1';
        wait for period;
        -- 1110
        a1 <= '1';
        a0 <= '1';
        b1 <= '1';
        b0 <= '0';
        wait for period;
        -- 1111
        a1 <= '1';
        a0 <= '1';
        b1 <= '1';
        b0 <= '1';
        wait for period;
        a1 <= '0';
        a0 <= '0';
        b1 <= '0';
        b0 <= '0';
        wait;

    end process;

end tb;
```

Name	Value	0 ns	100 ns	200 ns	300 ns	400 ns	500 ns	600 ns	700 ns	800 ns
 a1	0									
 a0	0									
 b1	0									
 b0	0									
 c2	0									
 c1	0									
 c0	0									
 period	50000 ps									
										

- Soruda verilen truth table ile simülasyon sonuçlarımız eşleşmektedir.
- 
- ✓ Bu bölümde ise data flow yapısıyla devre oluşturuldu. İsmi “Boolean\_Function\_Data\_Flow” olarak adlandırdım.
  - ✓ Kodun RTL şeması ve kodun kendisi aşağıdaki gibidir.

```

library IEEE;

use IEEE.STD_LOGIC_1164.ALL;

entity Boolean_Function_Data_Flow is

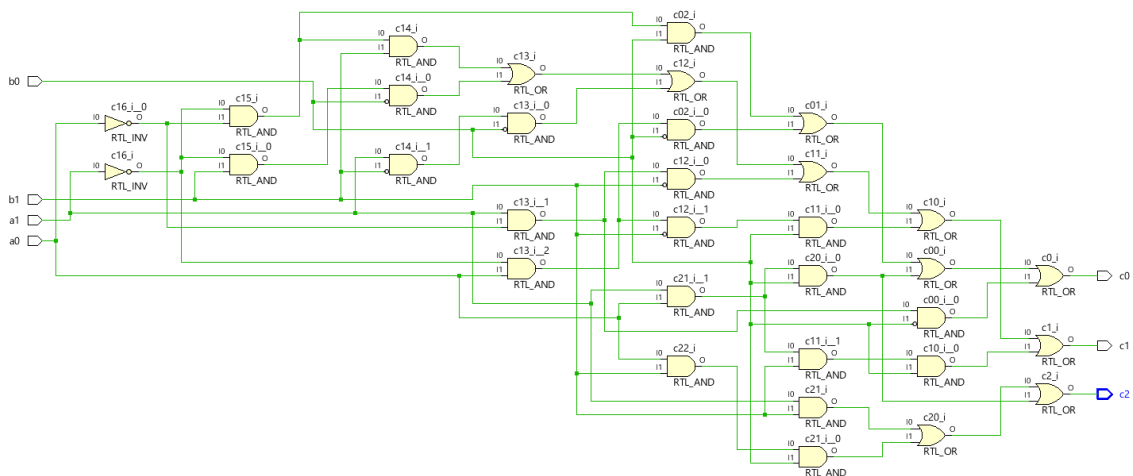
    Port ( a1 : in  STD_LOGIC;
           a0 : in  STD_LOGIC;
           b1 : in  STD_LOGIC;
           b0 : in  STD_LOGIC;

           c2 : out STD_LOGIC;
           c1 : out STD_LOGIC;
           c0 : out STD_LOGIC);

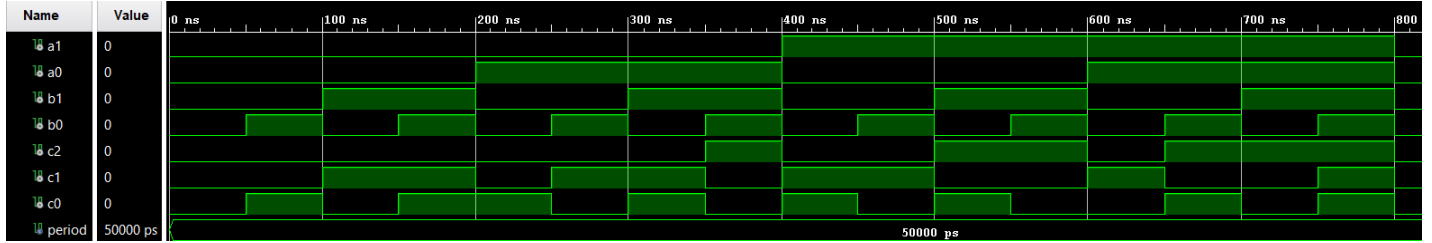
end Boolean_Function_Data_Flow;

architecture Behavioral of Boolean_Function_Data_Flow is
begin
    c2 <= (a1 AND b1) OR (a0 AND b1 AND b0) OR (a1 AND a0 AND b0);
    c1 <= ((NOT a1) AND (NOT a0) AND b1) OR ( (NOT a1) AND b1 AND (NOT b0))
    OR (a1 AND (NOT b1) AND (NOT b0)) OR (a1 AND (NOT a0) AND (NOT b1))
    OR ( (NOT a1) AND a0 AND (NOT b1) AND b0) OR ( a1 AND a0 AND b1 AND b0);
    c0 <= ( (NOT a1) AND (NOT a0) AND b0) OR ( (NOT a1) AND a0 AND (NOT b0))
    OR (a1 AND a0 AND b0) OR (a1 AND (NOT a0) AND (NOT b0));
end Behavioral;

```



- ✓ Test bench kodları aynı kaldı sadece “Boolean\_Function\_Case\_Statement” olan yerleri “Boolean\_Function\_Data\_Flow” olarak , “Boolean\_Function\_Case\_Statement\_tb” olan yerleri de “Boolean\_Function\_Data\_Flow\_tb” olarak değiştirdim. Yeni simülasyon dosyası oluşturarak bu kodu yapıştırdım. Kodun çıktısı aşağıdaki gibidir.



- ✓ Beklenildiği gibi simülasyon sonucumuz doğruluk tablosu ile aynı çıktı. Ek olarak bu sonucumuz case statement şeklinde yaptığımız sonuçla da aynı çıktı, bunun olması yapıyı doğru tasarladığımızı gösteriyor.