Yiğit Bektaş GÜRSOY
040180063
Ödev 2

- 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;
```

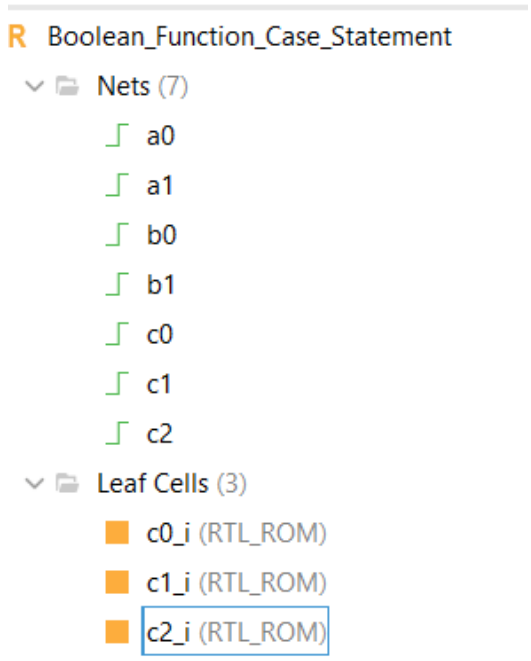
- Ardından örnek olarak verilen linklerden faydalanarak case struct oluşturdum. Kodum aşağıdaki gibidir.

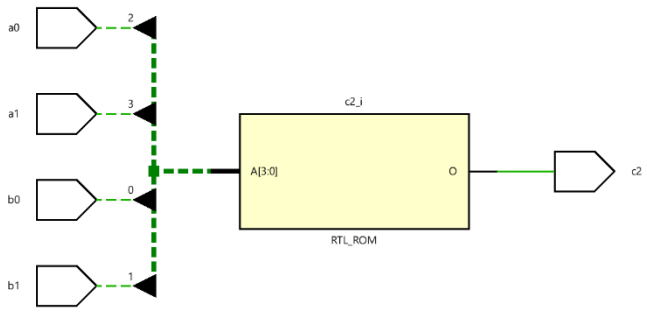
```
architecture Behavioral of
Boolean_Function_Case_Statement is

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';
```

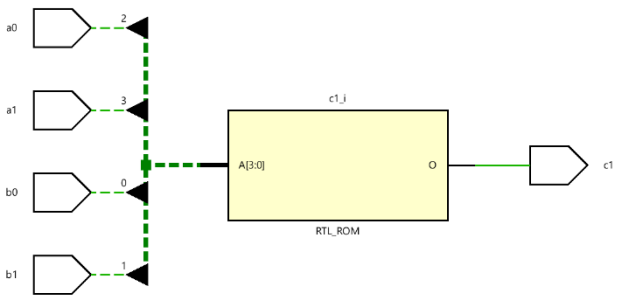
```
                when "0111" =>
                c2 <= '1';
                c1 <= '0';
                c0 <= '0';
                when "1000" =>
                c2 <= '0';
                c1 <= '1';
                c0 <= '0';
                when "1001" =>
                c2 <= '0';
                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;
```

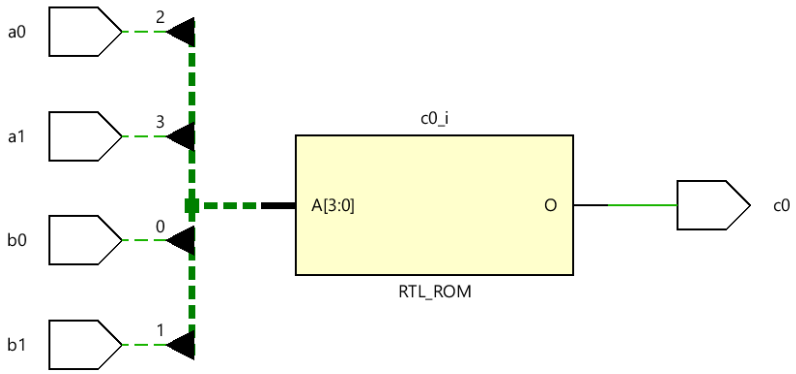
Yiğit Bektaş GÜRSOY
040180063
Ödev 2

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







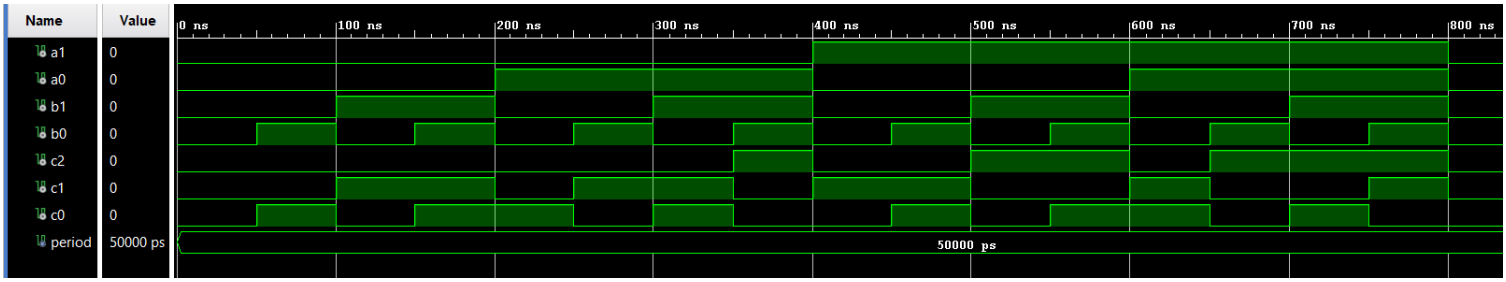


- Kodumuzun doğruluğunu kontrol etmek için testbench dosyası oluşturuyoruz. Yazılan kod ve simülasyon sonucu aşağıdadır.

Yiğit Bektaş GÜRSOY
040180063
Ödev 2

```
--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';
```

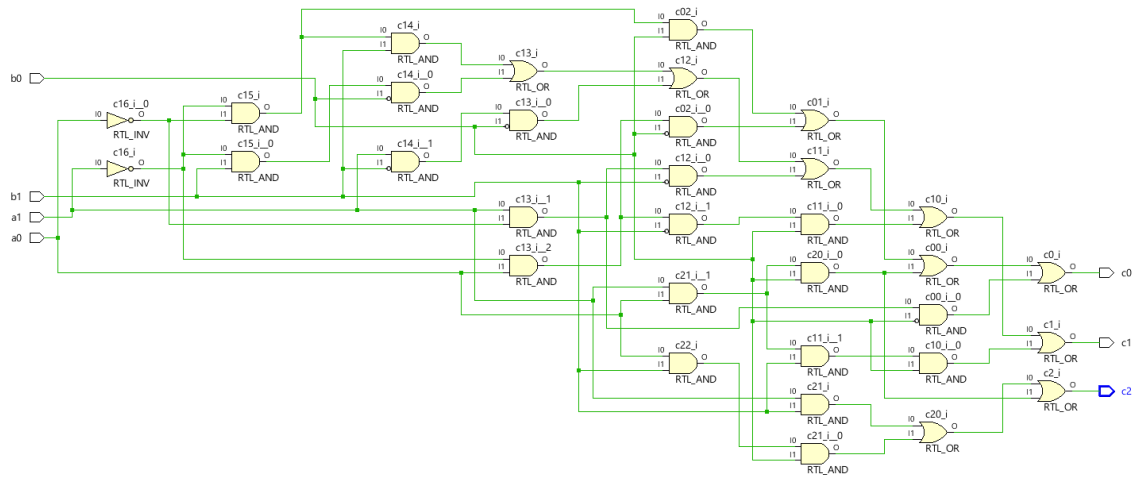
```
wait for period;
    -- 0111
    a1 <= '0';
    a0 <= '1';
    b1 <= '1';
    b0 <= '1';
    wait for period;
    -- 1000
    a1 <= '1';
    a0 <= '0';
    b1 <= '0';
    b0 <= '0';
    wait for period;
    -- 1001
    a1 <= '1';
    a0 <= '0';
    b1 <= '0';
    b0 <= '1';
    wait for period;
    -- 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;
```



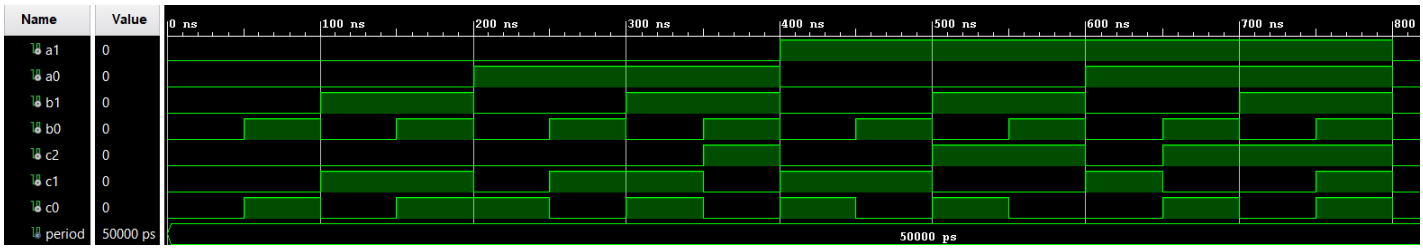
- Soruda verilen truth table ile simülasyon sonuçlarımız eşleşmektedir.

✓ Bu bölümde ise data flow yapısıyla devre oluşturuldu. İsmini "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.