

PART 2

1) “HW4” adlı yeni bir proje oluşturdu

2) Verilen durumların doğruluk tablosunu kullanarak tek bitlik a ve b sayıları için bir karşılaştırıcı tasarladı

$$-x = a \text{ AND } b'$$

$$-y = (a \text{ AND } b) \text{ OR } (a' \text{ AND } b')$$

$$-z = a' \text{ AND } b$$

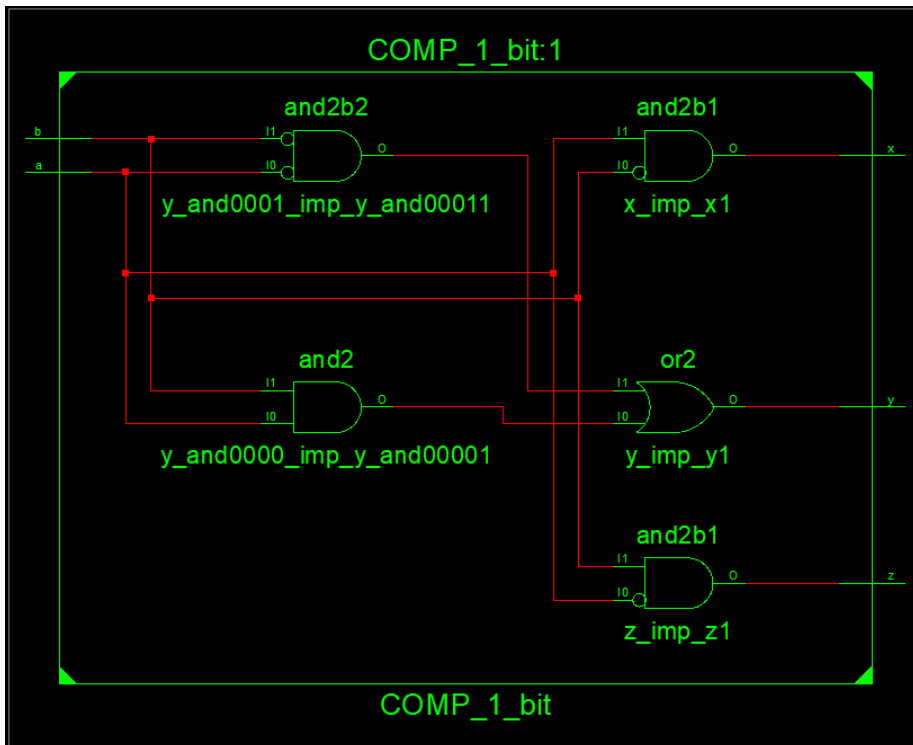
--“Yeni kaynak...”, “VHDL modülü” ile “COMP_1.vhd” adlı yeni bir kaynak oluşturuldu.

-Uygun giriş ve çıkışları eklemek için modülü düzenledi

-Tasarladığım devre için VDHL kodunu yazdım.

```
1  library IEEE;  
2  use IEEE.STD_LOGIC_1164.ALL;  
3  entity COMP_1_bit is  
4      Port ( a : in  STD_LOGIC;  
5            b : in  STD_LOGIC;  
6            x : out STD_LOGIC;  
7            y : out STD_LOGIC;  
8            z : out STD_LOGIC);  
9  end COMP_1_bit;  
10  
11  architecture Behavioral of COMP_1_bit is  
12  
13  begin  
14      x <= a AND (NOT b);  
15      y <= (a AND b) OR ((NOT a) AND (NOT b));  
16      z <= b AND (NOT a);  
17  end Behavioral;
```

RTL Şeması

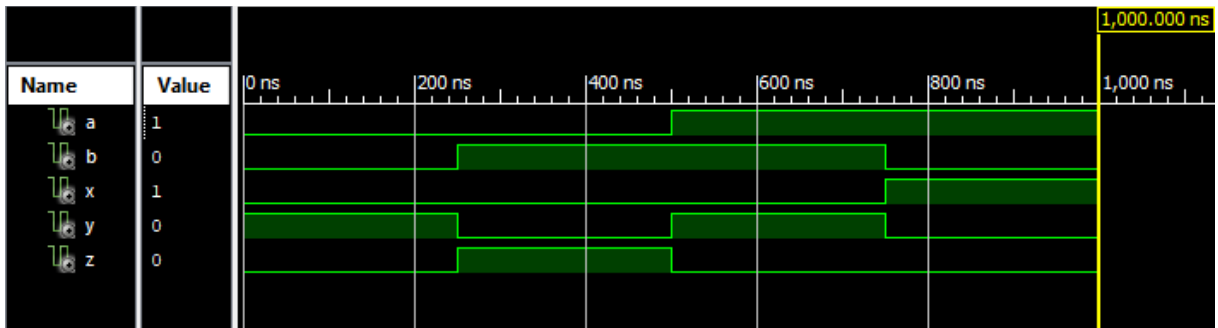


- “New source...” ile “COMP_1.tb.vhd” adlı “VHDL test bench” adlı bir test bench dosyası oluşturuldu, test modülü olarak “COMP_1.vhd” seçildi.

-Modül “COMP_1.tb.vhd” için uygun bir test tezgahı olacak şekilde düzenlendi.

```
21
22  --Inputs
23  signal a : std_logic := '0';
24  signal b : std_logic := '0';
25
26  --Outputs
27  signal x : std_logic;
28  signal y : std_logic;
29  signal z : std_logic;
30
31  BEGIN
32
33  uut: COMP_1_bit PORT MAP (
34      a => a,
35      b => b,
36      x => x,
37      y => y,
38      z => z
39  );
40
41  stim_proc: process
42  begin
43
44  -- 00
45  a <= '0';
46  b <= '0';
47  wait for 250ns;
48  -- 01
49  a <= '0';
50  b <= '1';
51  wait for 250ns;
52  -- 11
53  a <= '1';
54  b <= '1';
55  wait for 250ns;
56  -- 10
57  a <= '1';
58  b <= '0';
59  wait;
60
61  wait;
62  end process;
63
64  END;
```

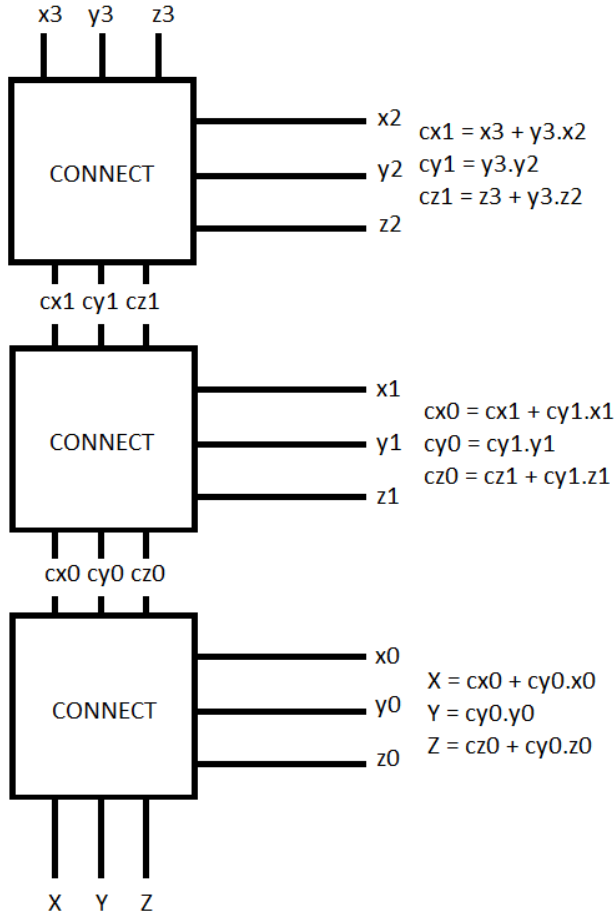
Behavioral Simulation Sonuçları



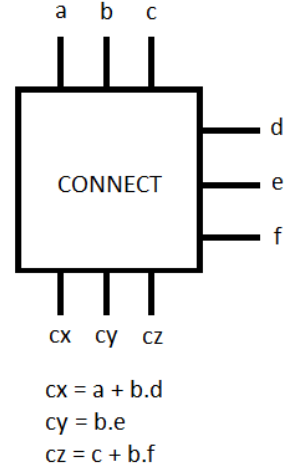
-Sonuçlar beklediğim gibi doğru çıktı.

3) -Devreyi verilen şekle göre tasarladık

-CONNECT modüllerinin modellenmesiyle başladı



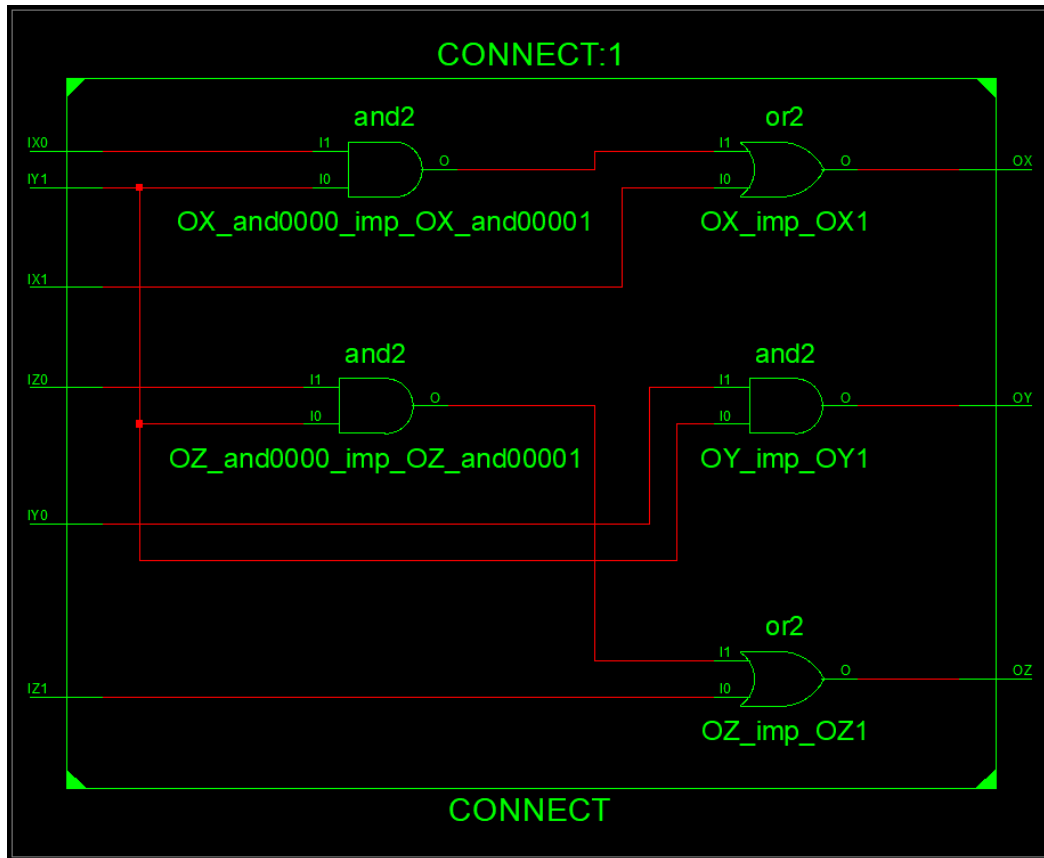
- Bu modeli kullanarak, genel CONNECT modülünü belirledi



- "CONNECT.vhd" adlı bir vhd dosyası oluşturdu ve CONNECT modülü için VHDL kodunu yazdım.

```
1 library IEEE;
2 use IEEE.STD_LOGIC_1164.ALL;
3
4
5 entity CONNECT is
6     Port ( IX1 : in  STD_LOGIC;
7           IY1 : in  STD_LOGIC;
8           IZ1 : in  STD_LOGIC;
9           IX0 : in  STD_LOGIC;
10          IY0 : in  STD_LOGIC;
11          IZ0 : in  STD_LOGIC;
12          OX : out  STD_LOGIC;
13          OY : out  STD_LOGIC;
14          OZ : out  STD_LOGIC);
15 end CONNECT;
16
17 architecture Behavioral of CONNECT is
18
19 begin
20
21 OX <= IX1 OR (IY1 AND IX0);
22 OY <= IY1 AND IY0;
23 OZ <= IZ1 OR (IY1 AND IZ0);
24
25 end Behavioral;
```

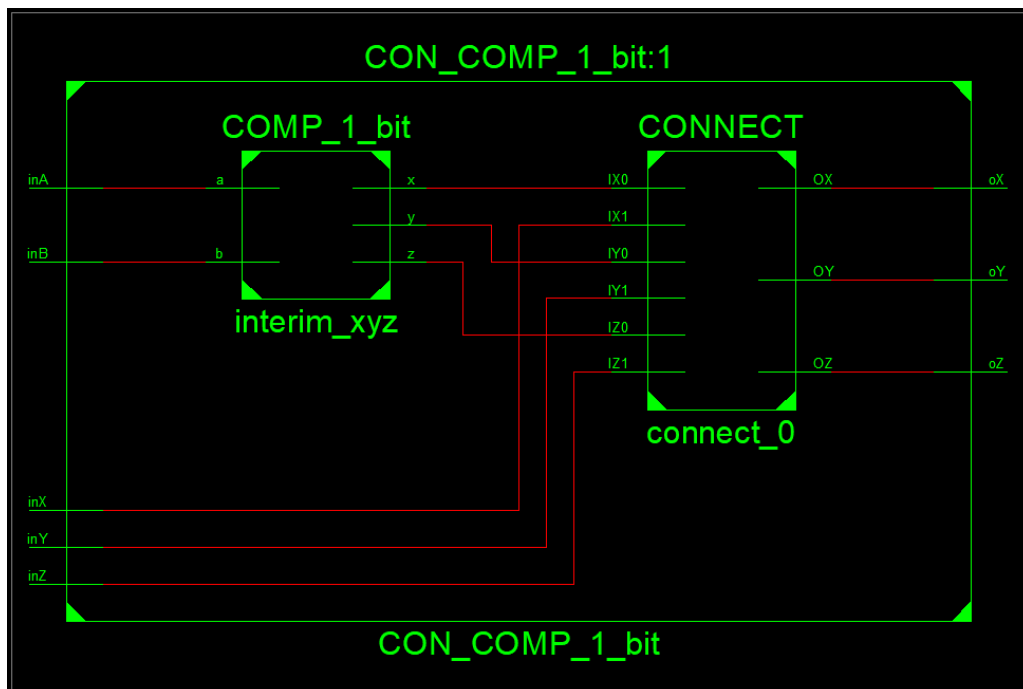
-RTL Şeması



- “CON_COMP_1bit.vhd” adında yeni bir vhd dosyası oluşturdum ve CONNECT ve COMP_1_bit modüllerini kullanarak VHDL kodunu yazdım.

```
7  entity CON_COMP_1_bit is
8      Port ( inA : in  STD_LOGIC;
9            inB : in  STD_LOGIC;
10           inX : in  STD_LOGIC;
11           inY : in  STD_LOGIC;
12           inZ : in  STD_LOGIC;
13           oX : out  STD_LOGIC;
14           oY : out  STD_LOGIC;
15           oZ : out  STD_LOGIC);
16  end CON_COMP_1_bit;
17
18  architecture Behavioral of CON_COMP_1_bit is
19
20  component COMP_1_bit is
21      Port ( a : in  STD_LOGIC;
22            b : in  STD_LOGIC;
23            x : out  STD_LOGIC;
24            y : out  STD_LOGIC;
25            z : out  STD_LOGIC);
26  end component;
27
28  component CONNECT is
29      Port ( IX1 : in  STD_LOGIC;
30            IY1 : in  STD_LOGIC;
31            IZ1 : in  STD_LOGIC;
32            IX0 : in  STD_LOGIC;
33            IY0 : in  STD_LOGIC;
34            IZ0 : in  STD_LOGIC;
35            OX : out  STD_LOGIC;
36            OY : out  STD_LOGIC;
37            OZ : out  STD_LOGIC);
38  end component;
39
40  signal mx, my, mz : STD_LOGIC;
41
42  begin
43
44  interim_xyz : COMP_1_bit port map (inA,inB,mx,my,mz);
45  connect_0 : CONNECT port map (inX,inY,inZ,mx,my,mz,oX,oY,oZ);
46
47  end Behavioral;
```

-RTL Şeması

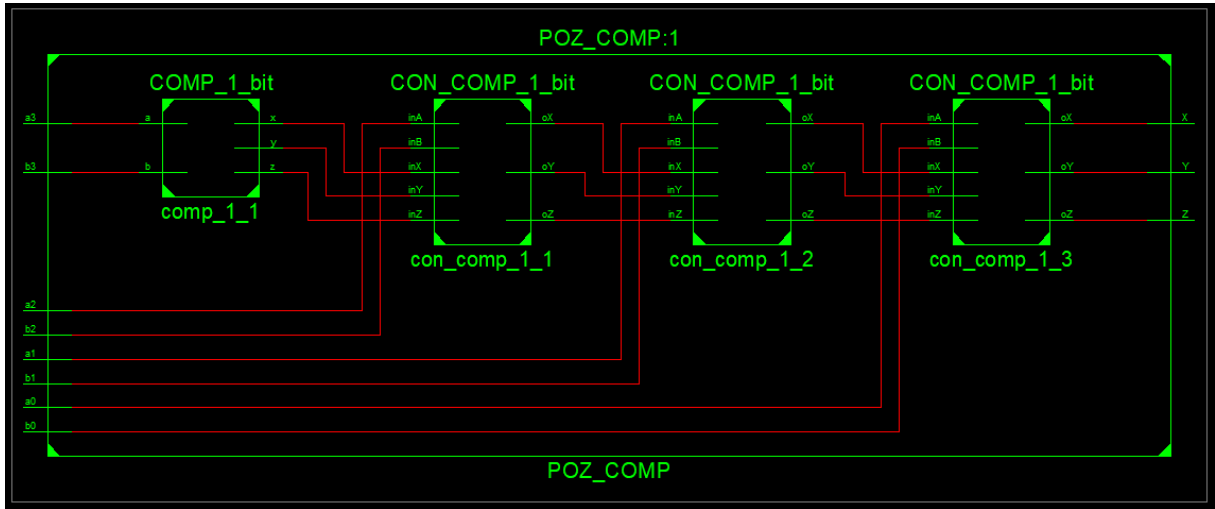


-“POZ_COMP.vhd” adlı yeni bir vhd dosyası oluşturdu ve Şekil 1'e göre VHDL kodunu yazdı.

-Yapı taşları olarak CON_COMP_1_bit ve COMP_1_bit modülleri kullanıldı

```
5  entity POZ_COMP is
6      Port ( a0 : in  STD_LOGIC;
7             a1 : in  STD_LOGIC;
8             a2 : in  STD_LOGIC;
9             a3 : in  STD_LOGIC;
10            b0 : in  STD_LOGIC;
11            b1 : in  STD_LOGIC;
12            b2 : in  STD_LOGIC;
13            b3 : in  STD_LOGIC;
14            X : out  STD_LOGIC;
15            Y : out  STD_LOGIC;
16            Z : out  STD_LOGIC);
17 end POZ_COMP;
18
19 architecture Behavioral of POZ_COMP is
20
21 component COMP_1_bit is
22     Port ( a : in  STD_LOGIC;
23           b : in  STD_LOGIC;
24           x : out  STD_LOGIC;
25           y : out  STD_LOGIC;
26           z : out  STD_LOGIC);
27 end component;
28
29 component CON_COMP_1_bit is
30     Port ( inA : in  STD_LOGIC;
31           inB : in  STD_LOGIC;
32           inX : in  STD_LOGIC;
33           inY : in  STD_LOGIC;
34           inZ : in  STD_LOGIC;
35           oX : out  STD_LOGIC;
36           oY : out  STD_LOGIC;
37           oZ : out  STD_LOGIC);
38 end component;
39
40 signal mx2,my2,mz2 : STD_LOGIC;
41 signal mx1,my1,mz1 : STD_LOGIC;
42 signal mx0,my0,mz0 : STD_LOGIC;
43
44 begin
45
46 comp_1_1: COMP_1_bit port map (a3,b3,mx2,my2,mz2);
47 con_comp_1_1: CON_COMP_1_bit port map (a2,b2,mx2,my2,mz2,mx1,my1,mz1);
48 con_comp_1_2: CON_COMP_1_bit port map (a1,b1,mx1,my1,mz1,mx0,my0,mz0);
49 con_comp_1_3: CON_COMP_1_bit port map (a0,b0,mx0,my0,mz0,X,Y,Z);
50
51 end Behavioral;
```

RTL Şeması



- "New source..." ile "POZ_COMP.tb.vhd" adlı "VHDL test bench" adlı bir test bench dosyası oluşturdum, test modülü olarak "POZ_COMP.vhd"yi seçtim.

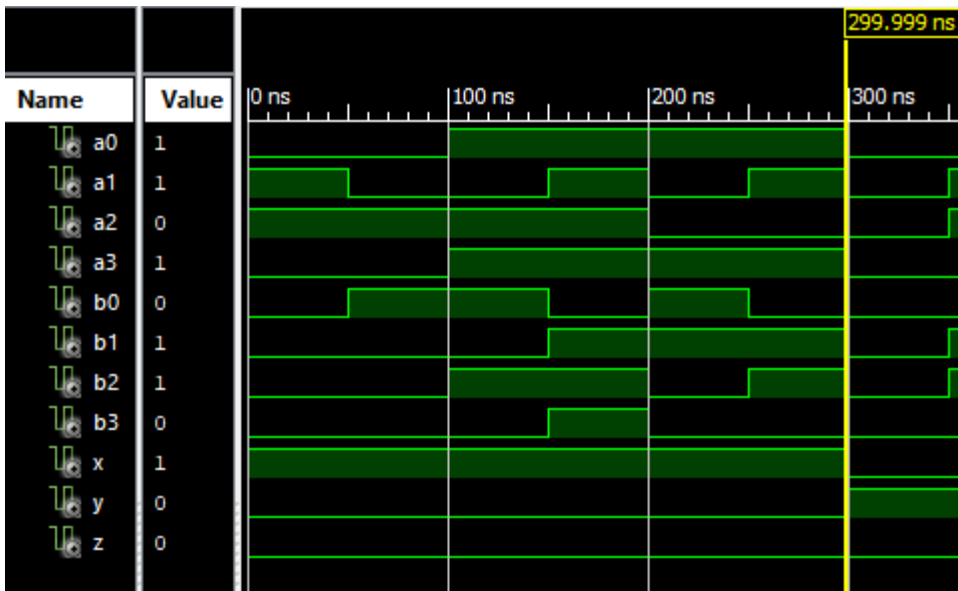
-300ns aralıklarla üç durum için test bench yazdım.

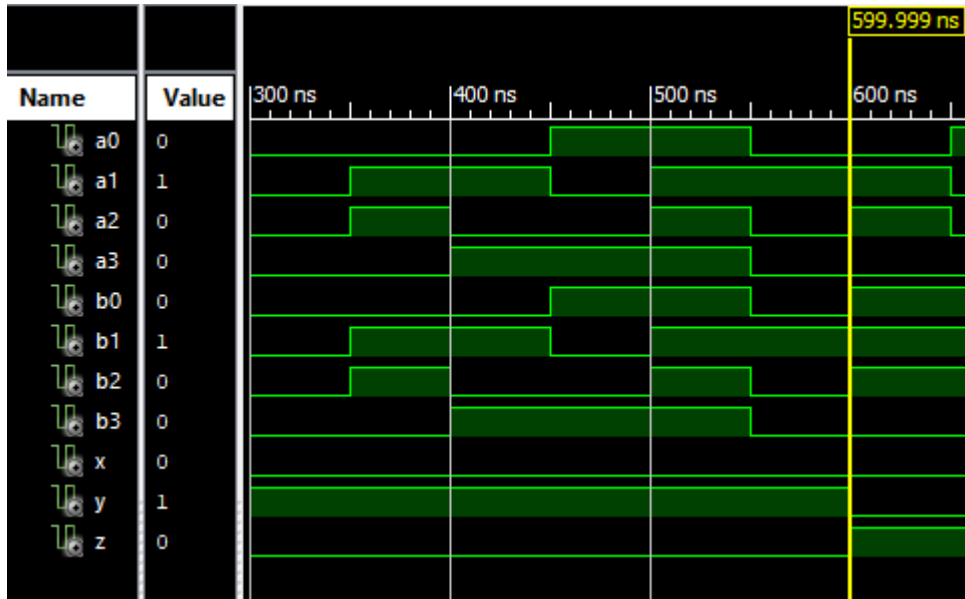
A>B için -0-300 ns

A=B için -300-600 ns

A<B için -600-900 ns

Behavioral Simulation Sonuçları





Sonuçlar beklediğim gibi hesaplarımla eşleşti.