



Gazi Üniversitesi
Teknoloji Fakültesi
Bilgisayar Mühendisliği Bölümü

Sayısal Elektronik Devreler Laboratuvarı
Ödevi

-Ders notunda verilen uygulamalar

Hazırlayan
Özgür Sadık Utku
181816072

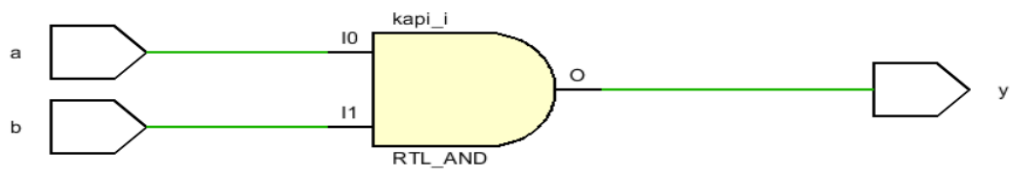
1.HAFTA

1.1 AND(VE) Kapısı

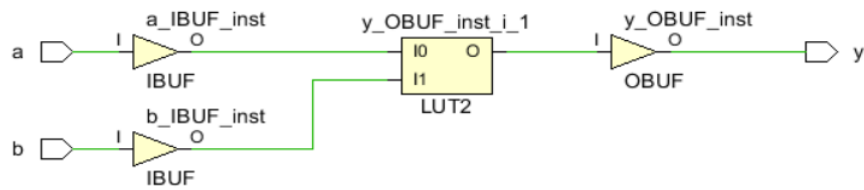
Kod

```
23 module ozgur_andkapisi(  
24     input a,  
25     input b,  
26     output y  
27 );  
28     and kapi (y,a,b);  
29 endmodule
```

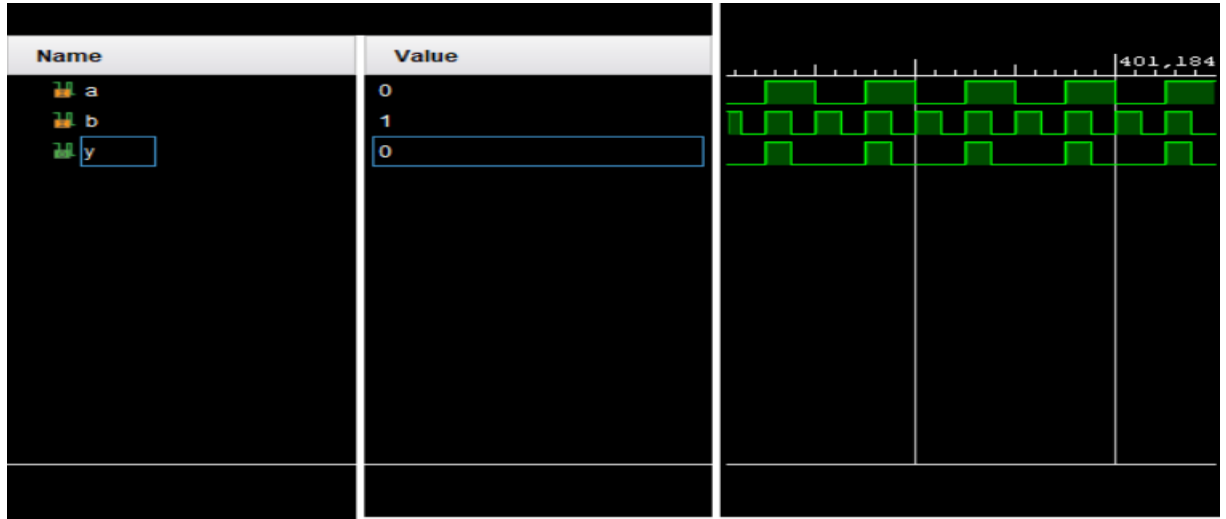
RTL Şematik



Şematik



Similasyon Sonucu

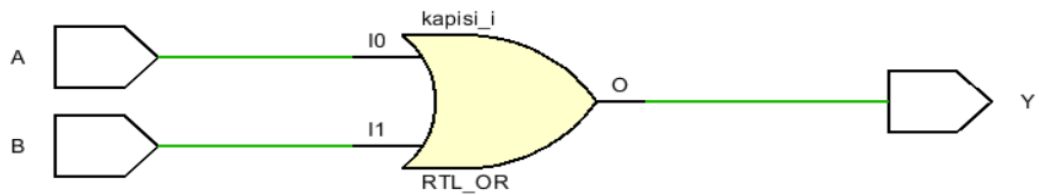


1.2 OR(VEYA) Kapısı

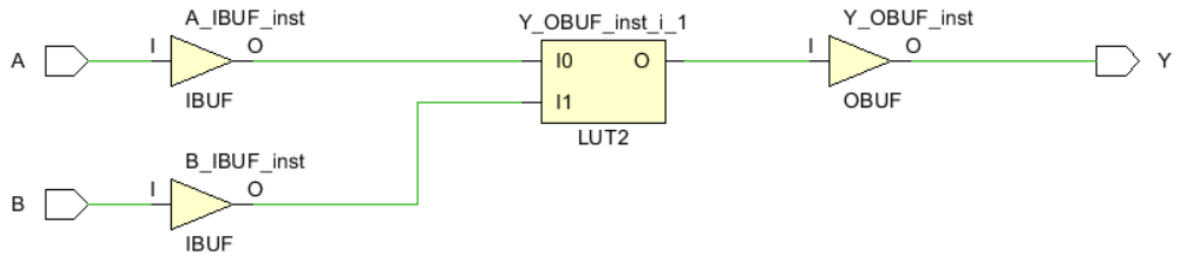
Kod

```
22 :  
23 module ozgur_orkapisi(  
24     input A,  
25     input B,  
26     output Y  
27 );  
28     or kapisi(Y,A,B);  
29 endmodule
```

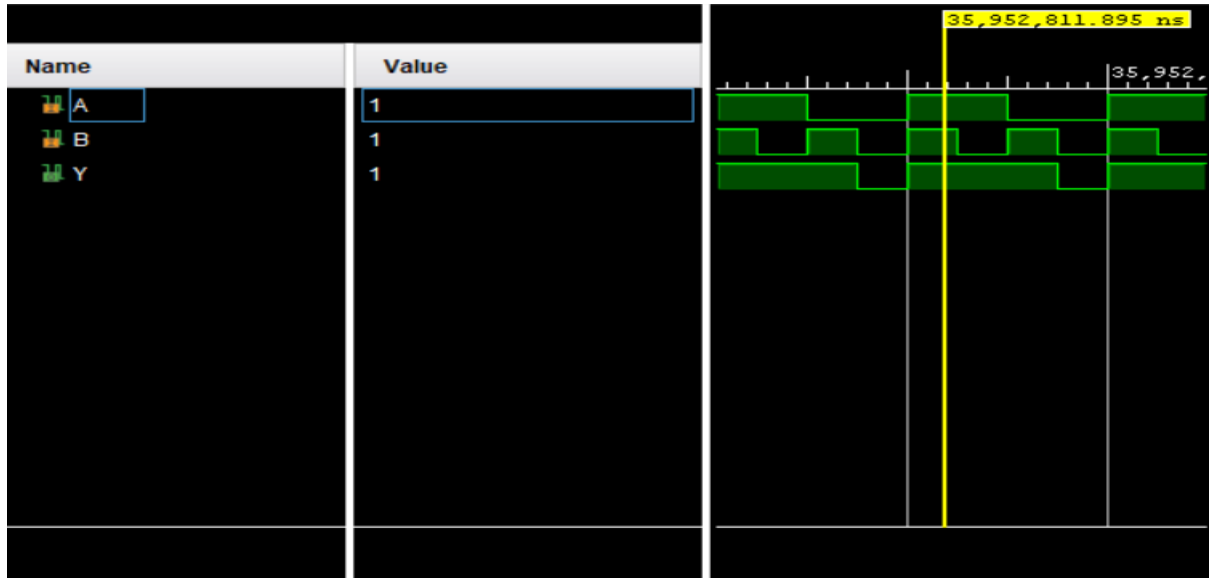
RTL Şematik



Şematik



Similasyon Sonucu

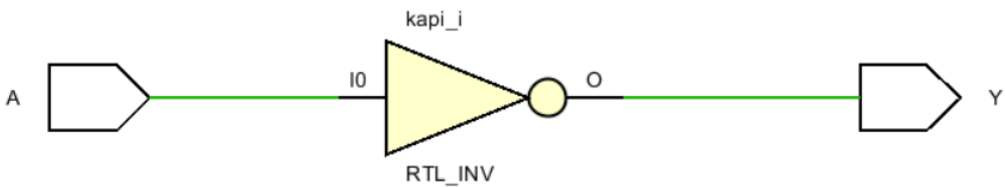


1.3 NOT(DEĞİL) Kapısı

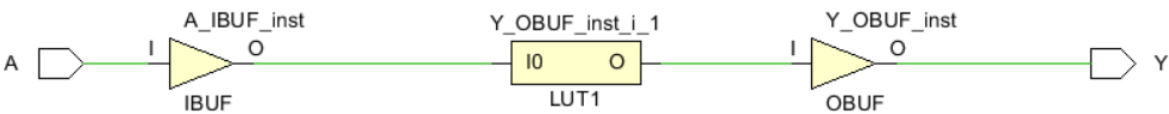
Kod

```
22 |
23 | module ozgur_notkapisi(
24 |     input A,
25 |     output Y
26 | );
27 |     not kapi(Y,A);
28 | endmodule
```

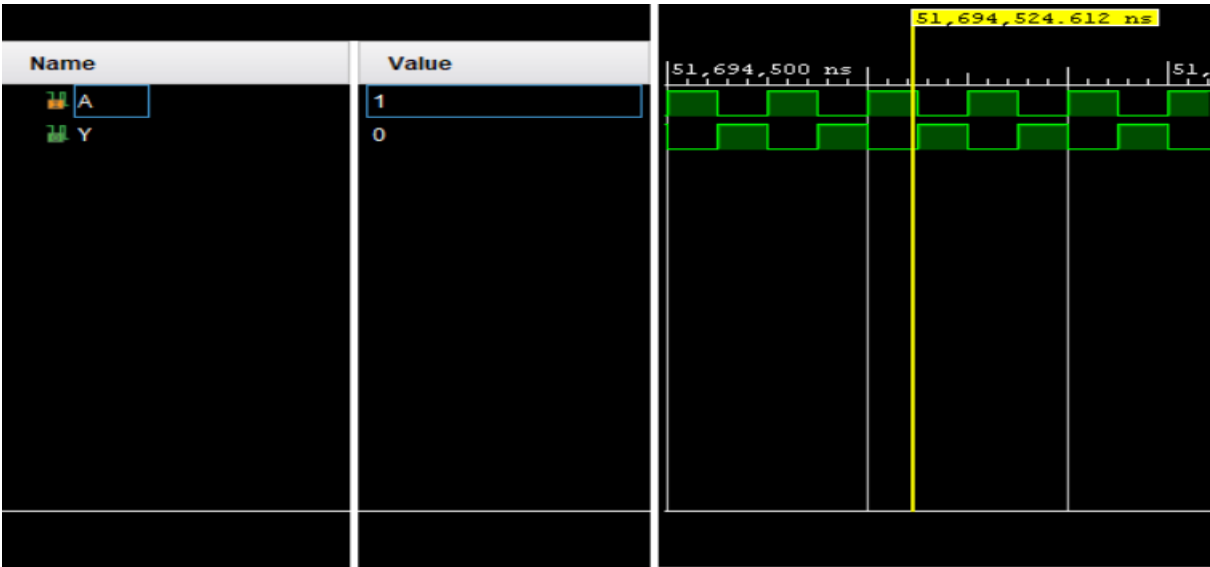
RTL Şematik



Şematik



Similasyon Sonucu



1.4 NAND(VE DEĞİL) Kapısı

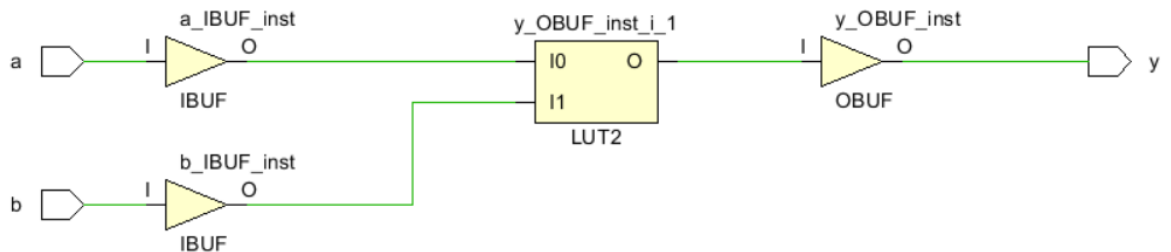
Kod

```
22 |
23 | module ozgur_nandkapisi(
24 |     input a,
25 |     input b,
26 |     output y
27 | );
28 |     nand kapi(y,a,b);
29 | endmodule
30 |
```

RTL Şematik



Şematik



Similasyon Sonucu



1.5 NOR(VEYA DEĞİL) Kapısı

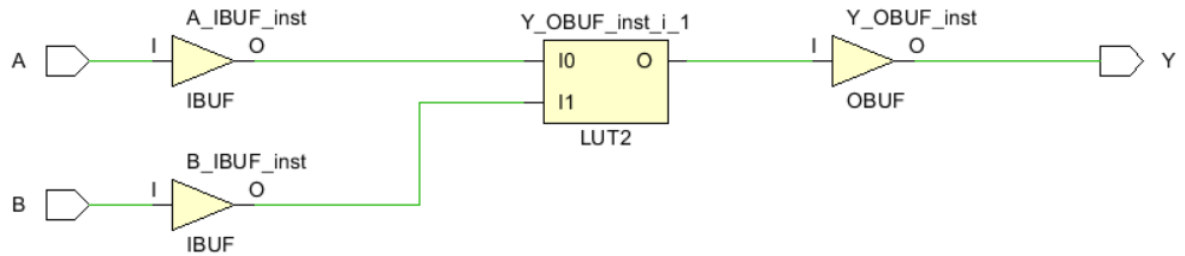
Kod

```
22 :  
23 module ozgur_norkapisi(  
24     input A,  
25     input B,  
26     output Y  
27 );  
28     nor kapi(Y,A,B);  
29 endmodule
```

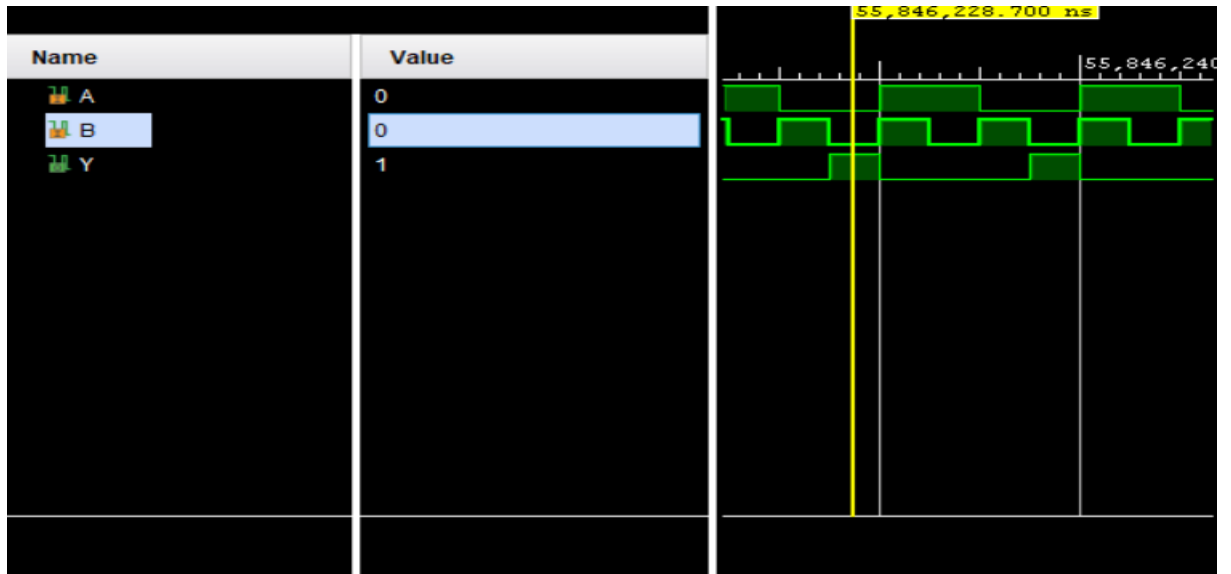
RTL Şematik



Şematik



Similasyon Sonucu

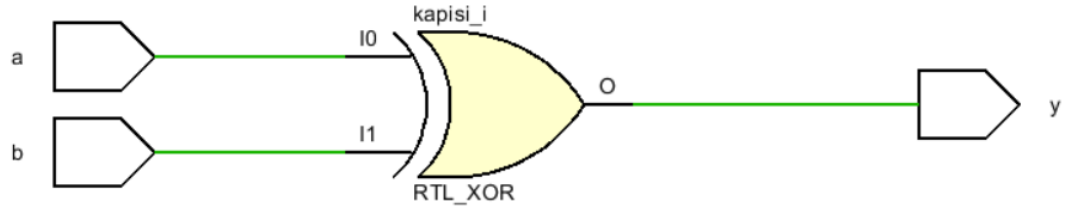


1.6 XOR(ÖZEL VEYA) Kapısı

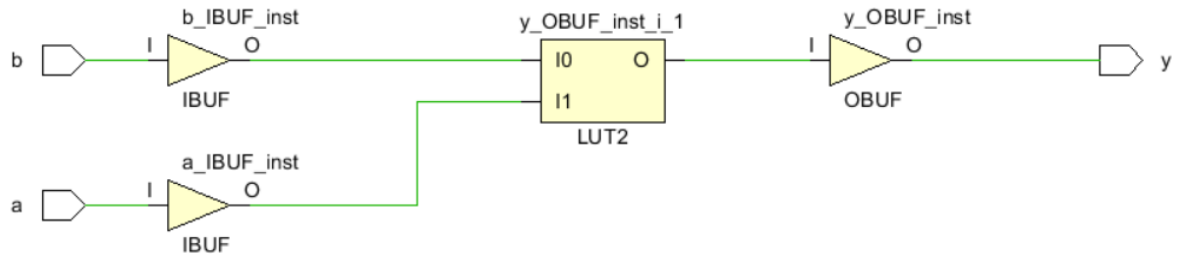
Kod

```
22 |  
23 | module ozgur_xorkapisi(  
24 |     input a,  
25 |     input b,  
26 |     output y  
27 | );  
28 |     xor kapisi (y,a,b);  
29 | endmodule
```


RTL Şematik



Şematik



Similasyon Sonucu

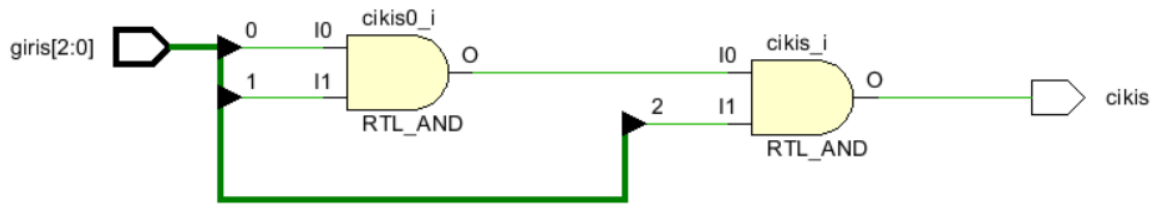


1.7 İki Girişten Fazla Girişe Sahip Kapı

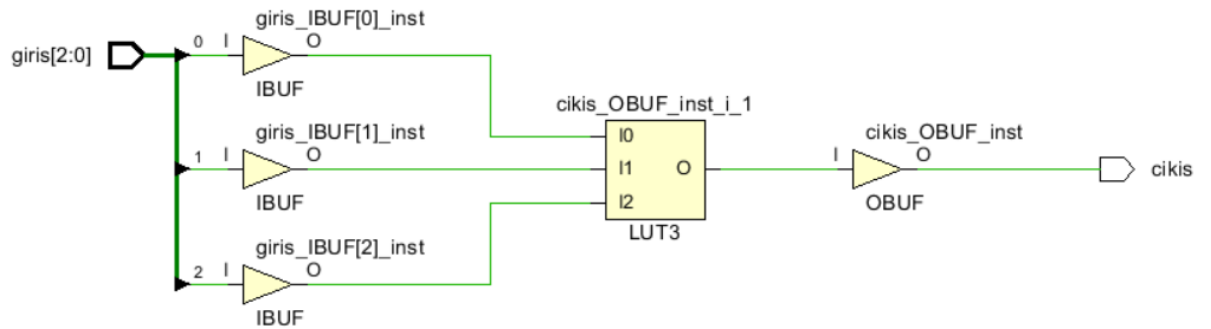
Kod

```
22 |
23 | module ozgur_ucgirisliand(
24 |     input [2:0] giris,
25 |     output cikis
26 | );
27 |     wire cikis;
28 |     assign cikis =giris[0]&giris[1]&giris[2];
29 | endmodule
```

RTL Şematik



Şematik



Similasyon Sonucu



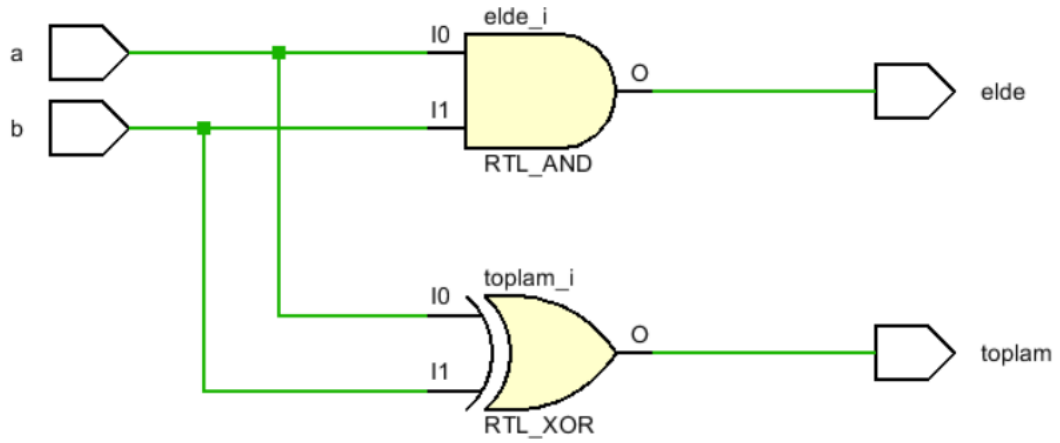
2.HAFTA

2.1 Half Adder (Yarım Toplayıcı) Devreleri

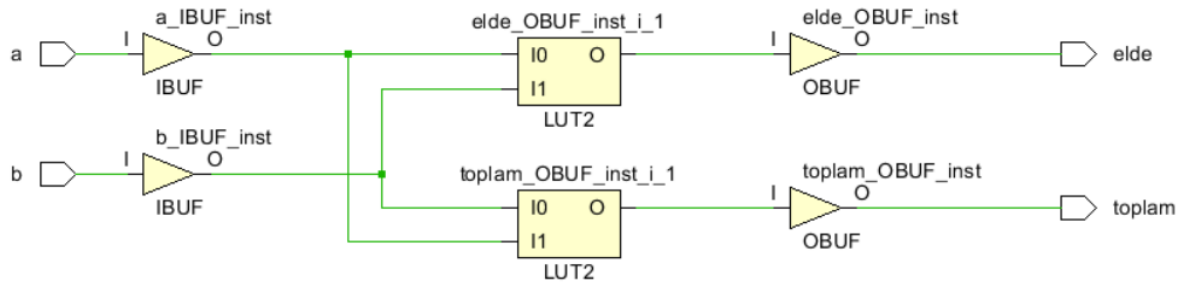
Kod

```
22 |
23 | module ozgur_YarimToplayici(
24 |     input a,
25 |     input b,
26 |     output toplam,
27 |     output elde
28 | );
29 |     xor(toplam,a,b);
30 |     and(elde,a,b);
31 | endmodule
```

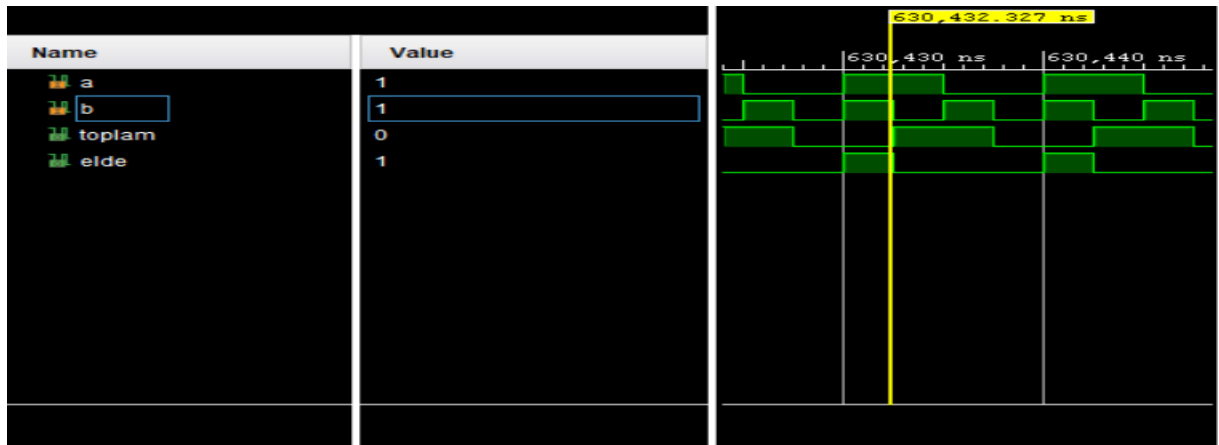
RTL Şematik



Şematik



Similasyon Sonucu

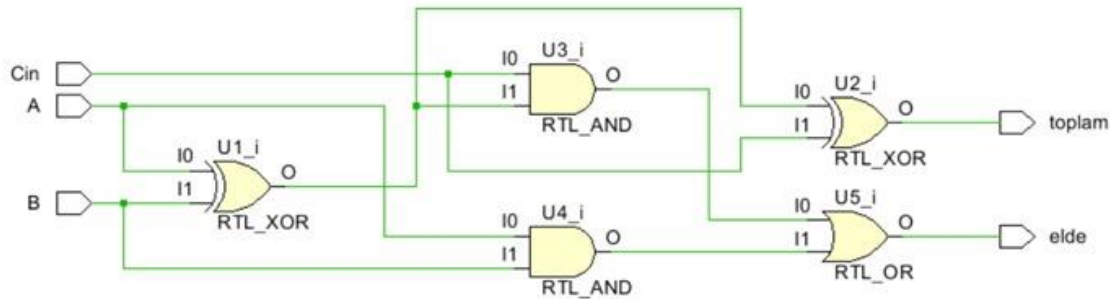


2.2 Full Adder (Tam Toplayıcı) Devreleri

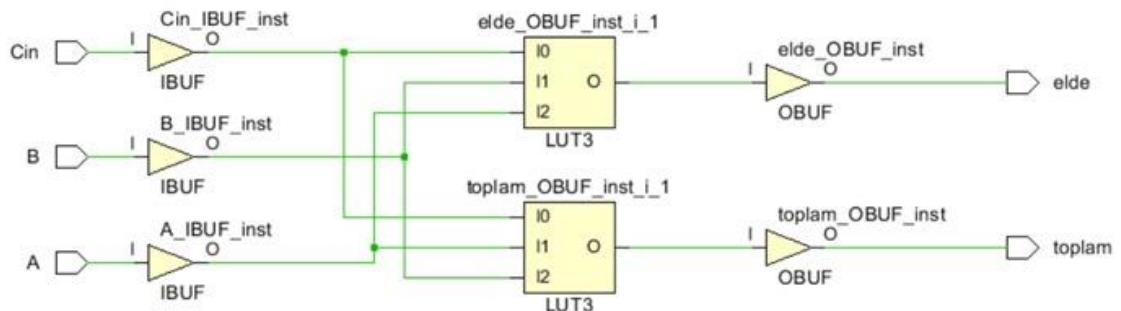
Kod

```
22 |
23 | module ozgur_TamToplayicii(
24 |     input A,
25 |     input B,
26 |     input Cin,
27 |     output toplam,
28 |     output elde
29 | );
30 |     wire t1,t2,t3;
31 |     xor U1(t1,A,B);
32 |     xor U2(toplam,t1,Cin);
33 |     and U3(t2,Cin,t1);
34 |     and U4(t3,A,B);
35 |     or U5(elde,t2,t3);
36 | endmodule
```

RTL Şematik



Şematik



Similasyon Sonucu



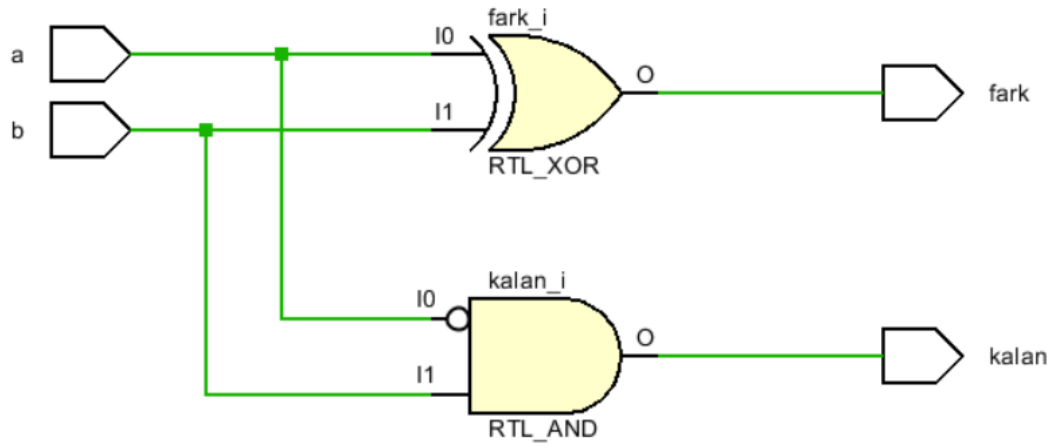
3.HAFTA

3.1 Half Subtractor (Yarım Çıkarıcı) Devreleri

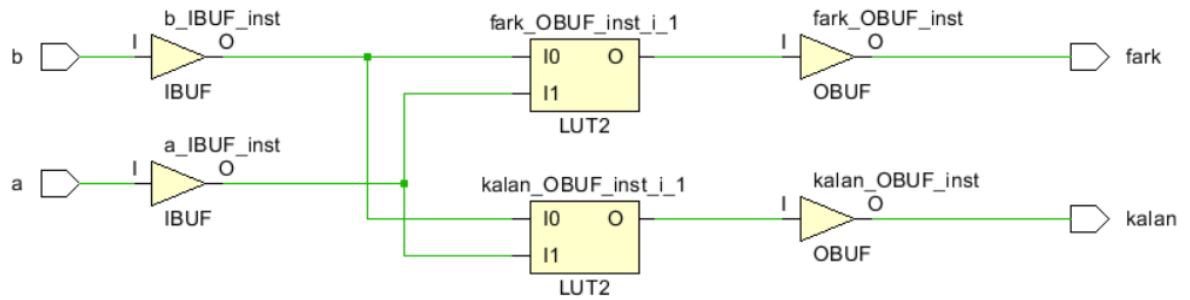
Kod

```
22 :  
23 module ozgur_YarimCikarici(  
24     input a,  
25     input b,  
26     output fark,  
27     output kalan  
28 );  
29     xor(fark,a,b);  
30     assign kalan = (~a&b);  
31 endmodule
```

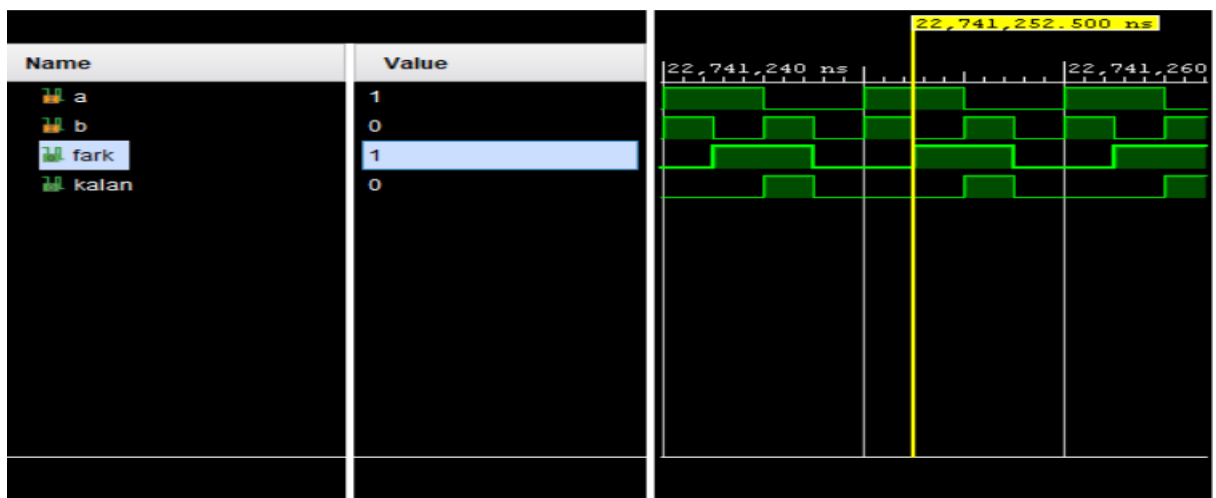
RTL Şematik



Şematik



Similasyon Sonucu

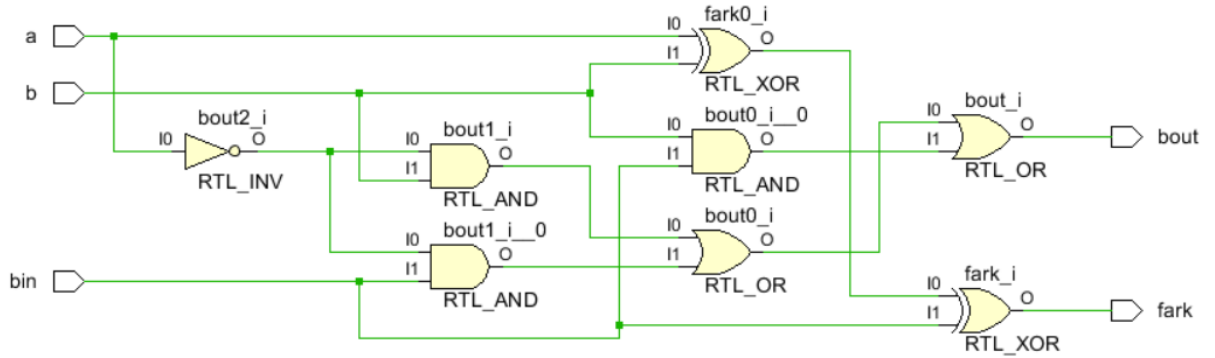


3.2 Full Subtractor (Tam Çıkarıcı) Devreleri

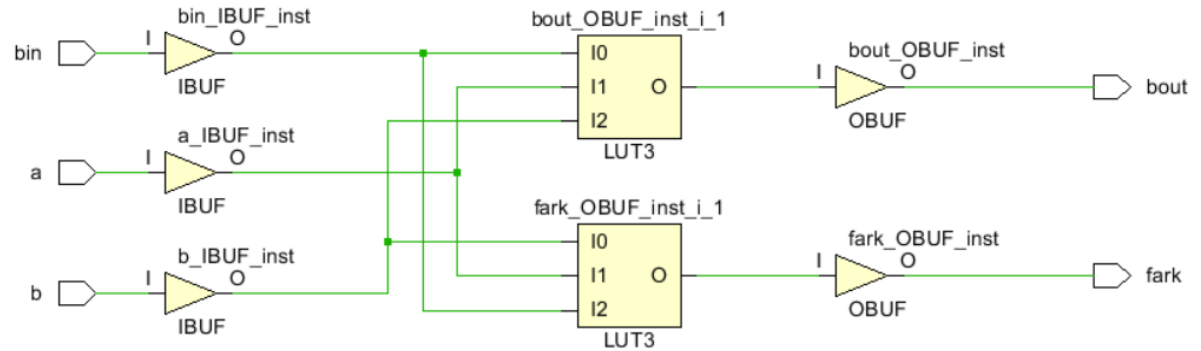
Kod

```
22
23 module ozgur_TamCikarici(
24     input a,
25     input b,
26     input bin,
27     output fark,
28     output bout
29 );
30     assign fark= (a^b^bin);
31     assign bout=(~a&b) | (~a&bin) | (b&bin);
32 endmodule
```

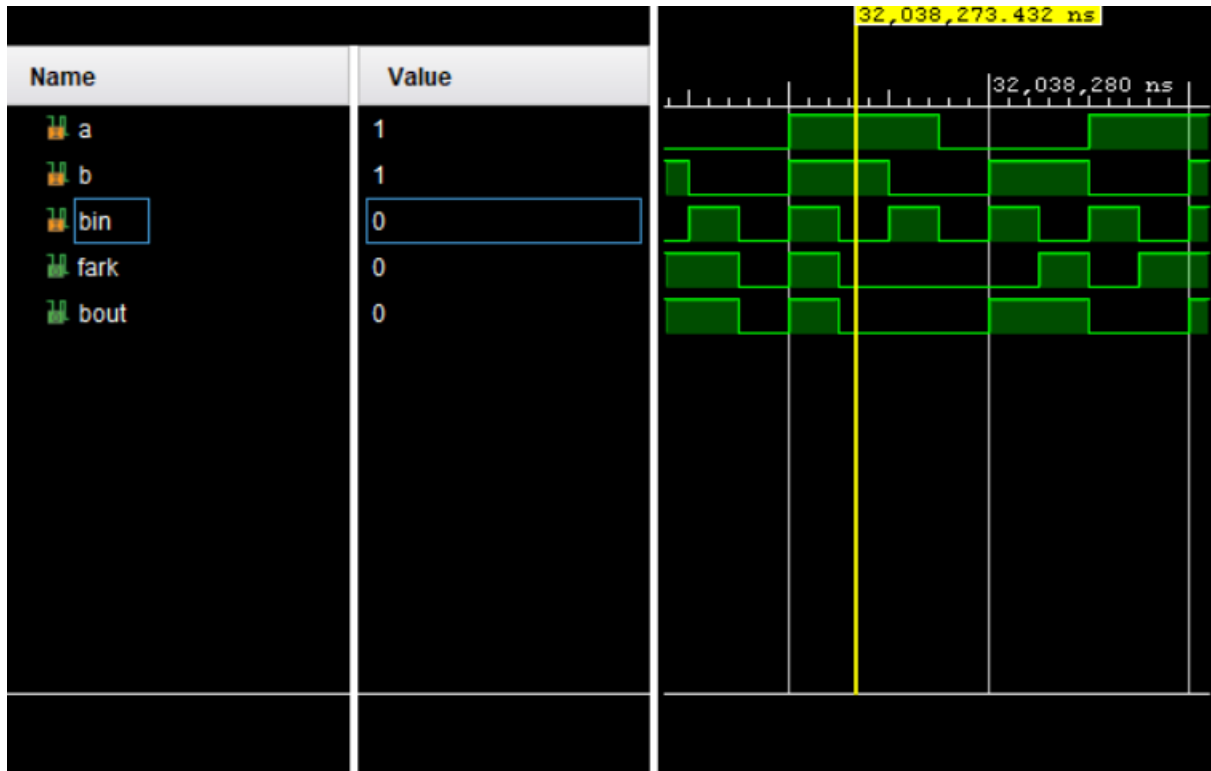
RTL Şematik



Şematik



Similasyon Sonucu



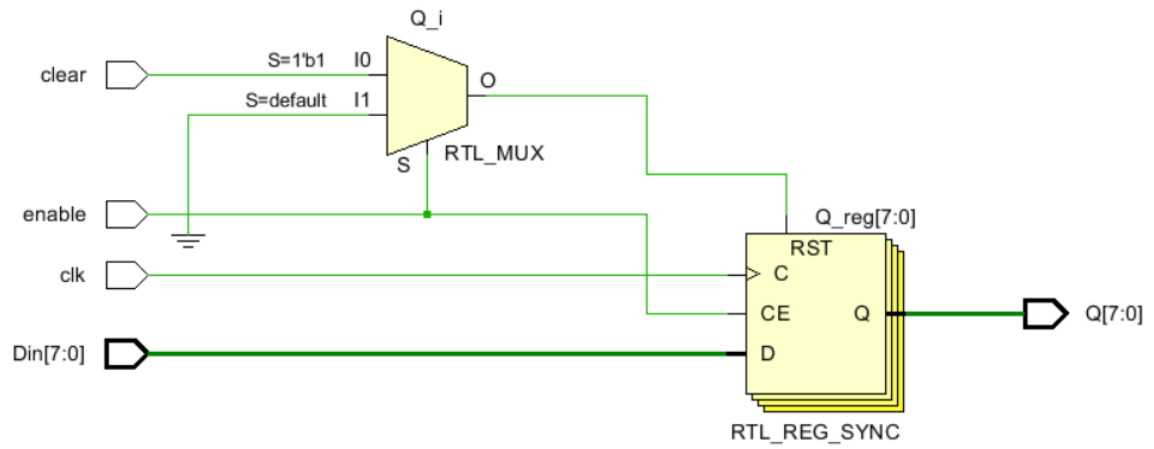
4.HAFTA

4.1 D Tipi Flip-Flop

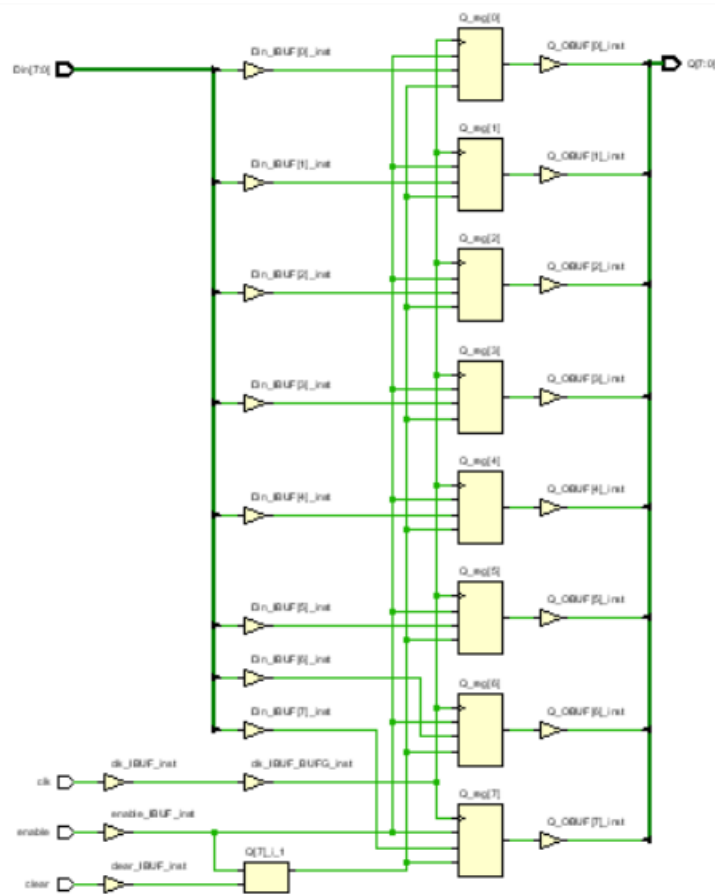
Kod

```
22 :  
23 module ozgur_DFlipflop(  
24     input [7:0] Din,  
25     input clk,  
26     input clear,  
27     input enable,  
28     output reg [7:0] Q  
29 );  
30 always@(posedge clk)  
31     if(enable)  
32     begin  
33         if(clear)  
34             Q<=0;  
35         else  
36             Q<=Din;  
37         end  
38     endmodule
```

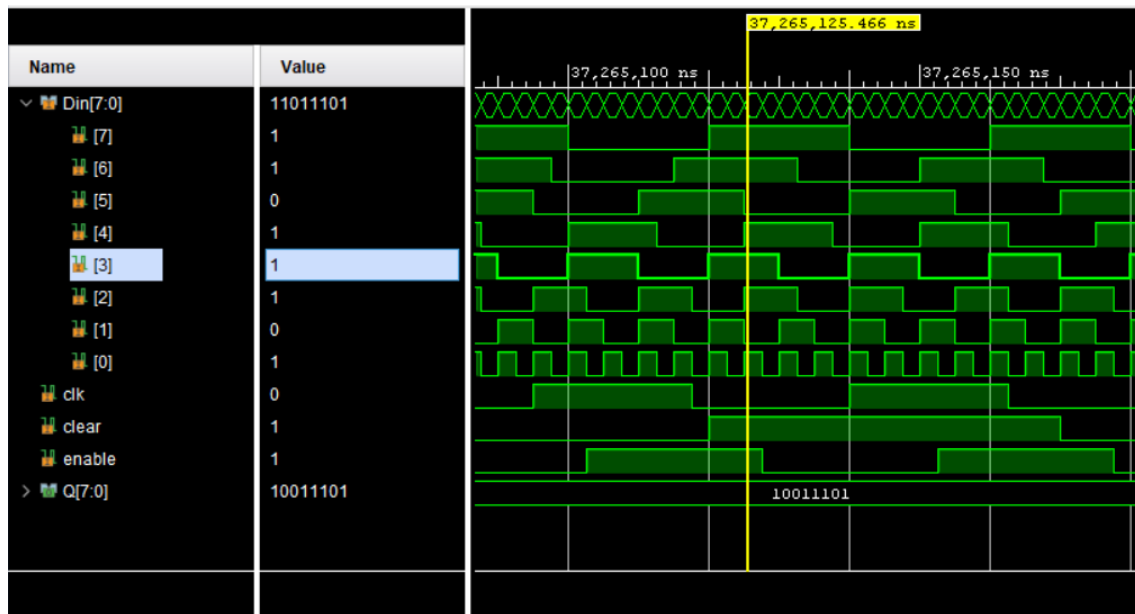
RTL Şematik



Şematik



Similasyon Sonucu

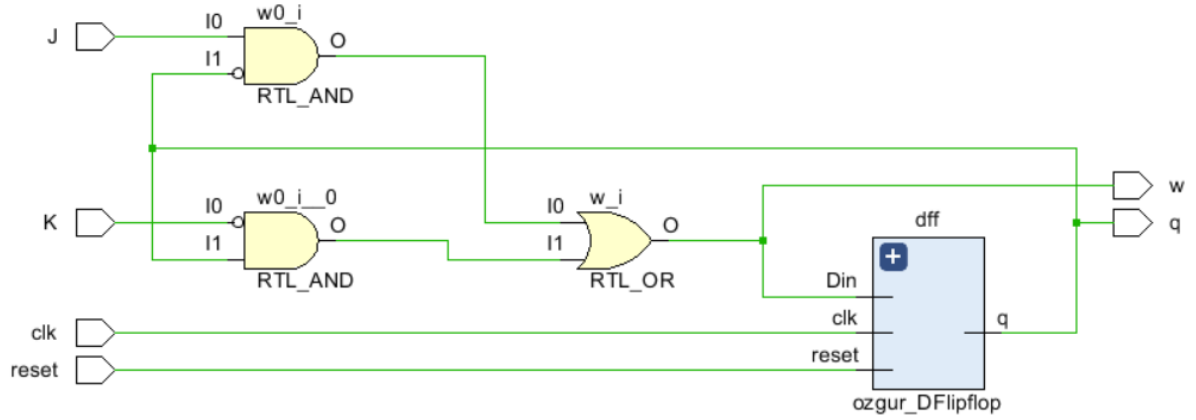


4.2 JK Tipi Flip-Flop

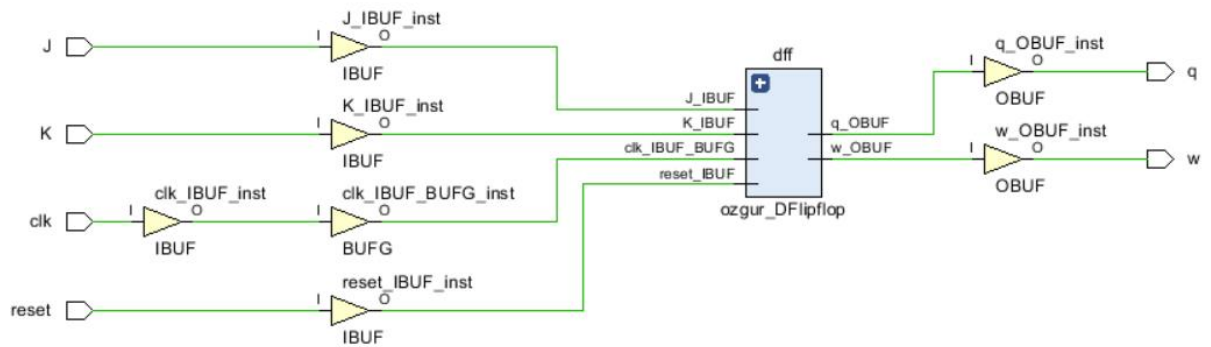
Kod

```
23 module ozgur_JKFlipFlop(  
24     input J,  
25     input K,  
26     input clk,  
27     input reset,  
28     output q,  
29     wire w  
30 );  
31 assign w = (J&~q) | (~K&q);  
32 ozgur_DFlipflop dff(w,clk,reset,q);  
33 endmodule  
34  
35 module ozgur_DFlipflop(Din,clk,reset,q);  
36     input Din,clk,reset;  
37     output reg q;  
38     always@(posedge clk)  
39     begin  
40         if(reset)  
41             q=1'b0;  
42         else  
43             q=Din;  
44         end  
45     endmodule
```

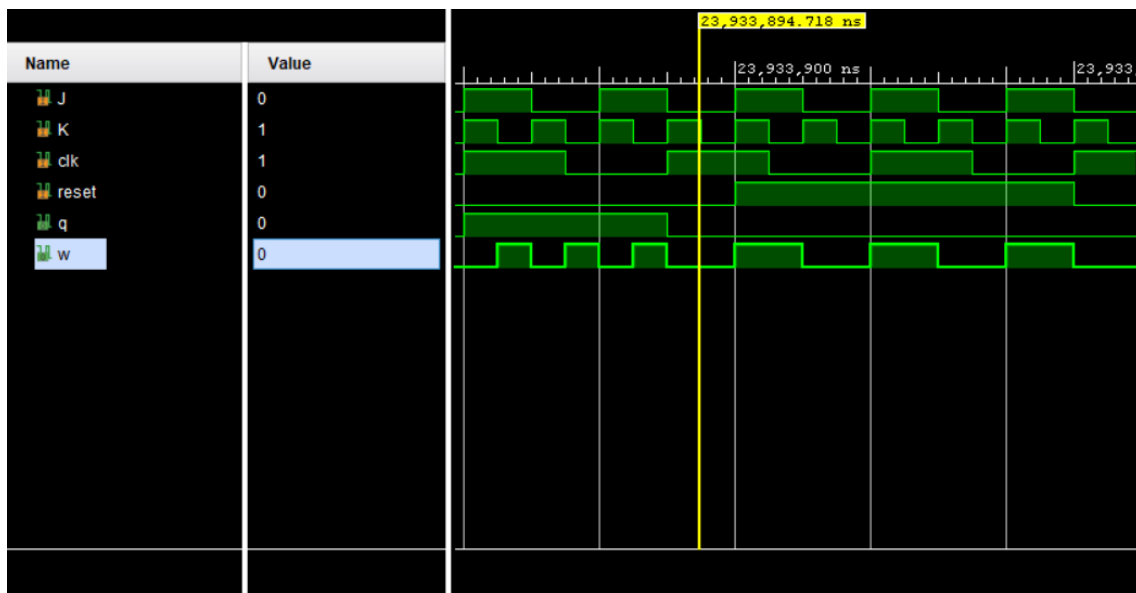
RTL Şematik



Şematik



Similasyon Sonucu



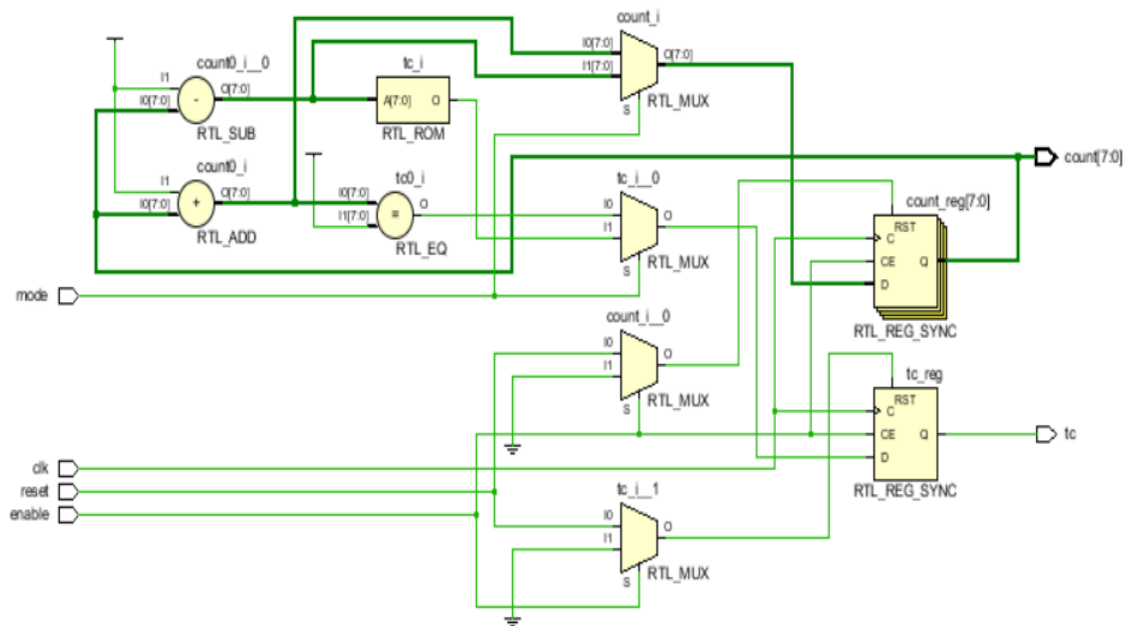
5.HAFTA

5.1 Up/Down Counter (Yukarı/Aşağı Sayıcı) Devresi

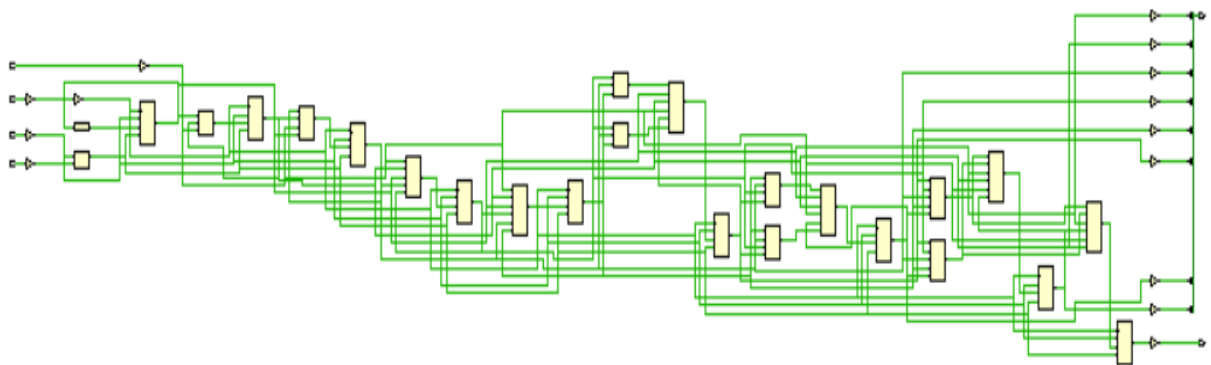
Kod

```
22 :  
23 module ozgur_AsagiYukariSayici(  
24     input clk,  
25     input enable,  
26     input reset,  
27     input mode,  
28     output reg[7:0] count,  
29     output reg tc  
30 );  
31 always@(posedge clk)  
32 begin  
33     if(enable)  
34     begin  
35         if(reset)  
36         begin  
37             count=0; tc=0;  
38         end  
39     else  
40     begin  
41         if(mode==0)  
42         begin  
43             count=count+1;  
44             if(count==255)  
45                 tc=1;  
46             else  
47                 tc=0;  
48         end  
49     else  
50     begin  
51         count=count-1;  
52         if(count==0)  
53             tc=1;  
54         else  
55             tc=0;  
56         end  
57     end  
58 end  
59 end  
60 endmodule  
--
```

RTL Şematik



Şematik



Similasyon Sonucu



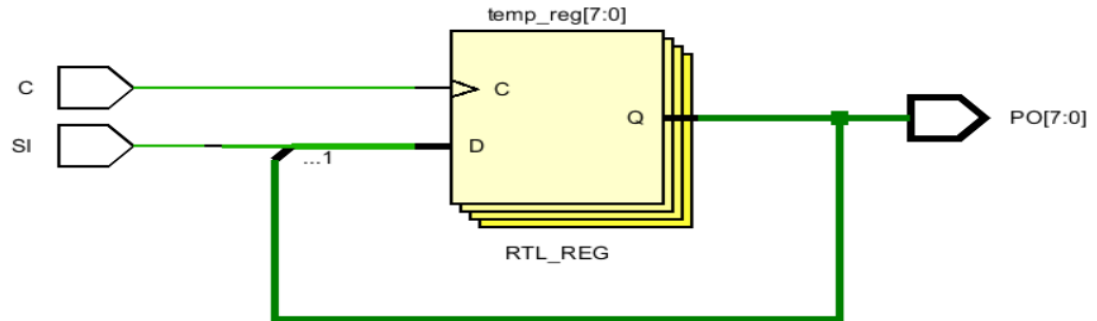
6.HAFTA

6.1 Shift Register (Kaydırma Yazmaç) Devresi

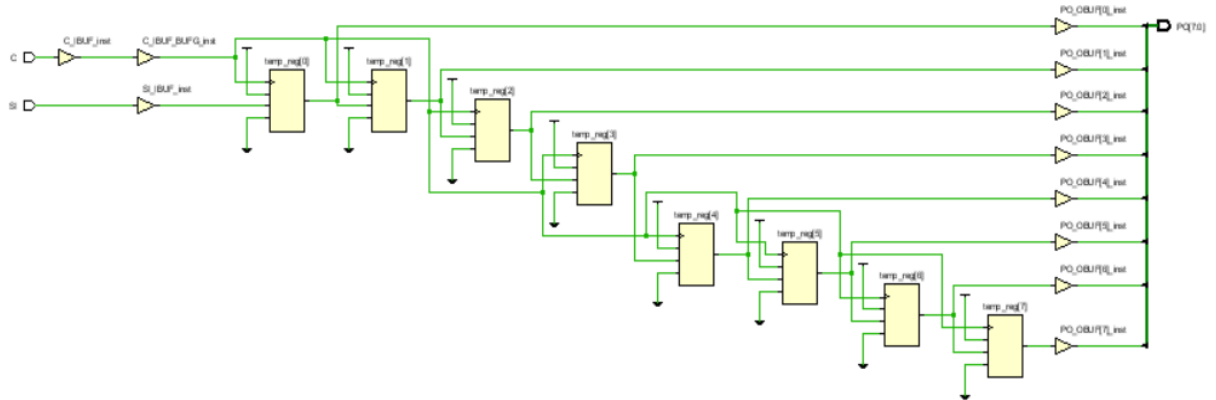
Kod

```
22 |  
23 | module ozgur_KaydirmaliYazmacSIPO(  
24 |     input C,  
25 |     input SI,  
26 |     output [7:0] PO  
27 | );  
28 |     reg [7:0] temp;  
29 |     always@(posedge C)  
30 |     begin  
31 |         temp={temp[6:0], SI};  
32 |     end  
33 |     assign PO = temp;  
34 | endmodule
```

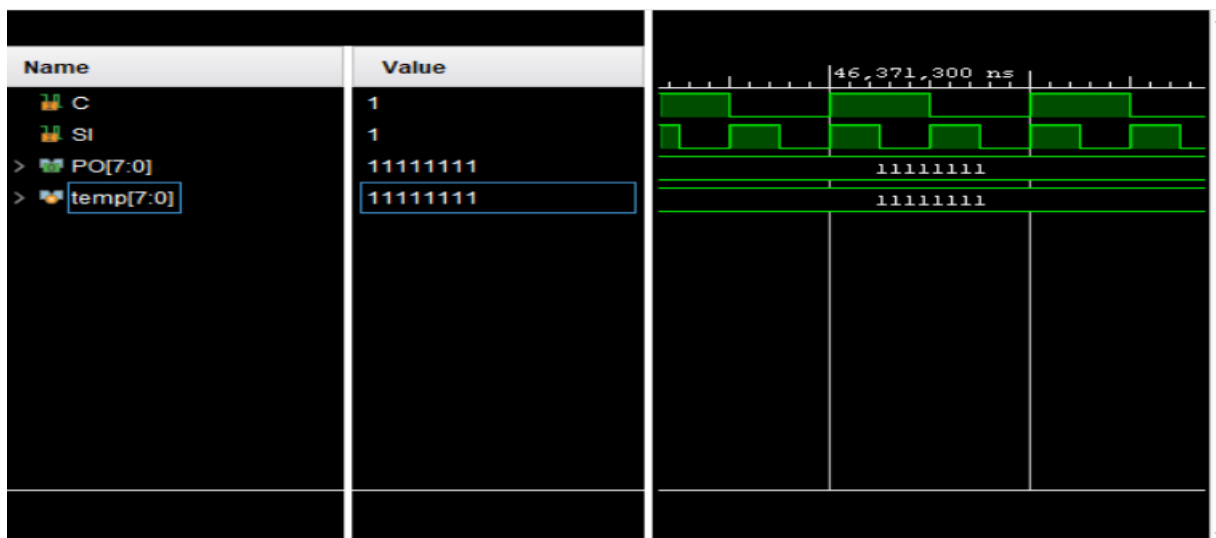
RTL Şematik



Şematik



Similasyon Sonucu



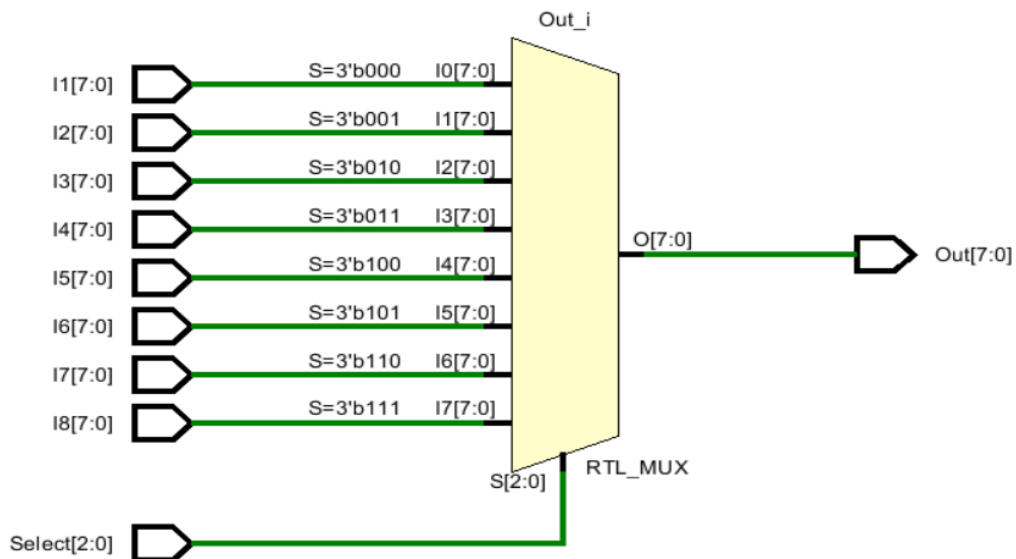
7.HAFTA

7.1 Multiplexer (Çoklayıcı) Devresi

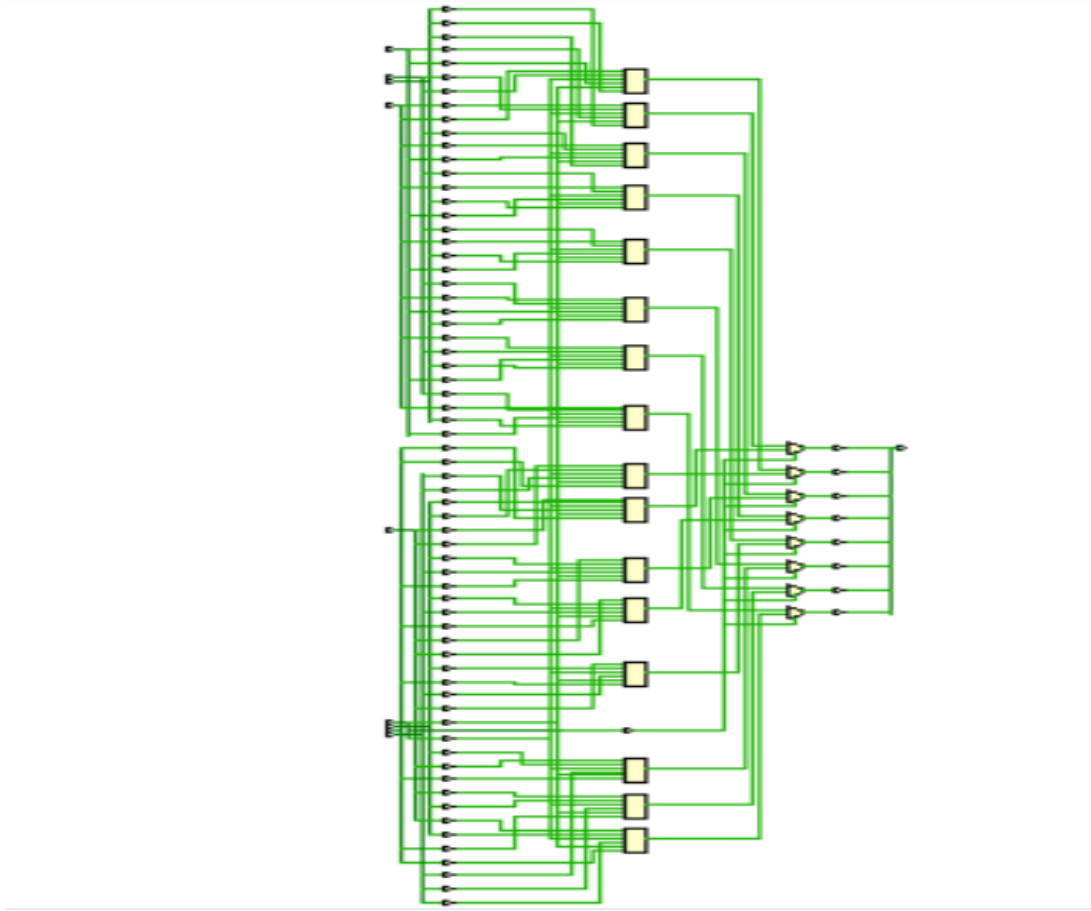
Kod

```
22 :  
23 module ozgur_mux8x1(  
24     input [7:0] I1,I2,I3,I4,I5,I6,I7,I8,  
25     input [2:0] Select,  
26     output [7:0] Out  
27 );  
28     reg[7:0] Out;  
29     always@(I1 or I2 or I3 or I4 or I5 or I6 or I7 or I8 or Select)  
30     begin  
31         case(Select)  
32             3'b000 : Out = I1;  
33             3'b001 : Out = I2;  
34             3'b010 : Out = I3;  
35             3'b011 : Out = I4;  
36             3'b100 : Out = I5;  
37             3'b101 : Out = I6;  
38             3'b110 : Out = I7;  
39             3'b111 : Out = I8;  
40             default : Out = 8'bx;  
41         endcase  
42     end  
43 endmodule
```

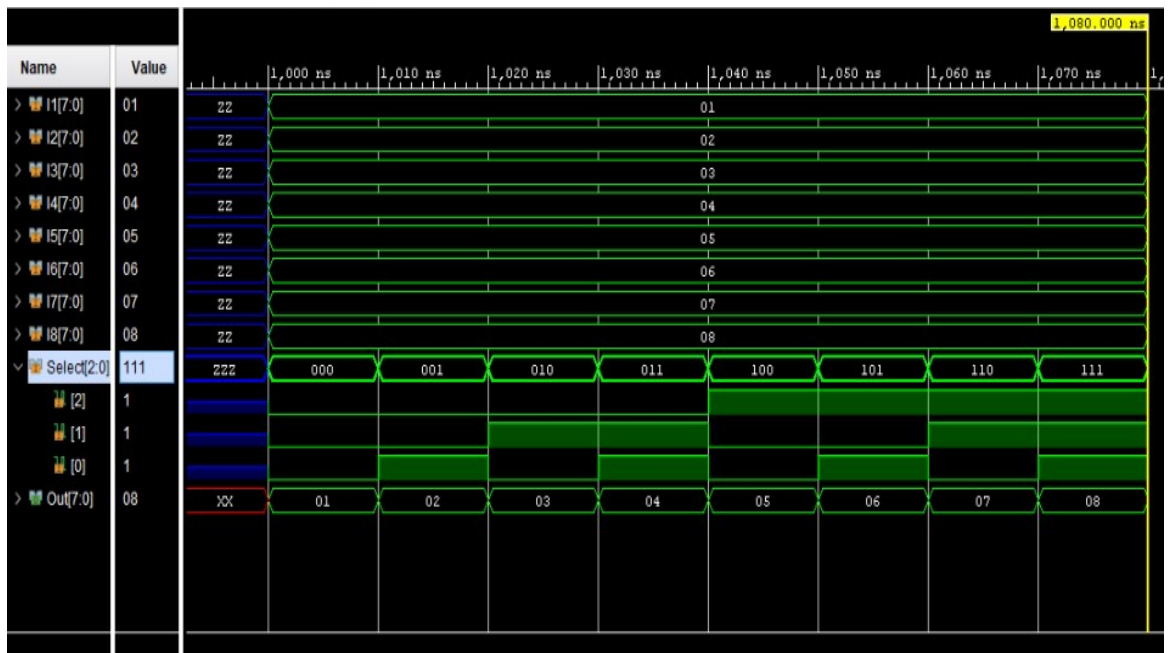
RTL Şematik



Şematik



Similasyon Sonucu



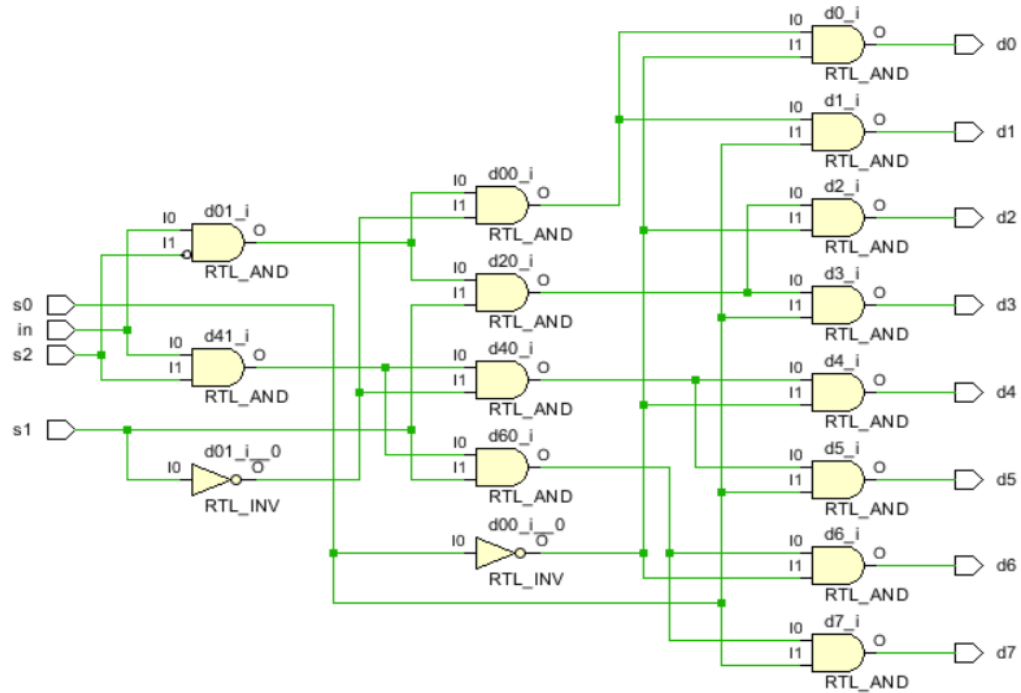
8.HAFTA

8.1 Demultiplexer (Çoklama Çözücü) Devresi

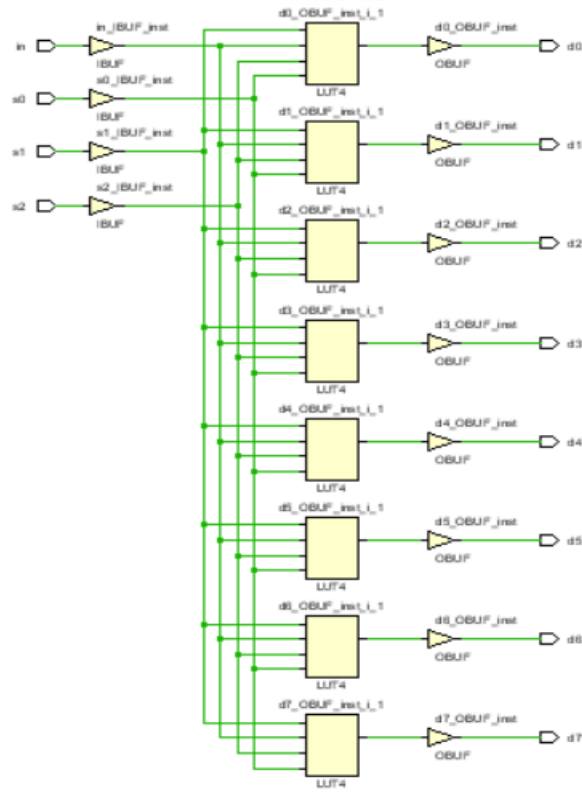
Kod

```
22 |
23 | module ozgur_Demux(
24 |     input in,
25 |     input s0,
26 |     input s1,
27 |     input s2,
28 |     output d0,d1,d2,d3,d4,d5,d6,d7
29 | );
30 | assign d0 = (in & ~s2 & ~s1 & ~s0),
31 |     d1 = (in & ~s2 & ~s1 & s0),
32 |     d2 = (in & ~s2 & s1 & ~s0),
33 |     d3 = (in & ~s2 & s1 & s0),
34 |     d4 = (in & s2 & ~s1 & ~s0),
35 |     d5 = (in & s2 & ~s1 & s0),
36 |     d6 = (in & s2 & s1 & ~s0),
37 |     d7 = (in & s2 & s1 & s0);
38 | endmodule
```

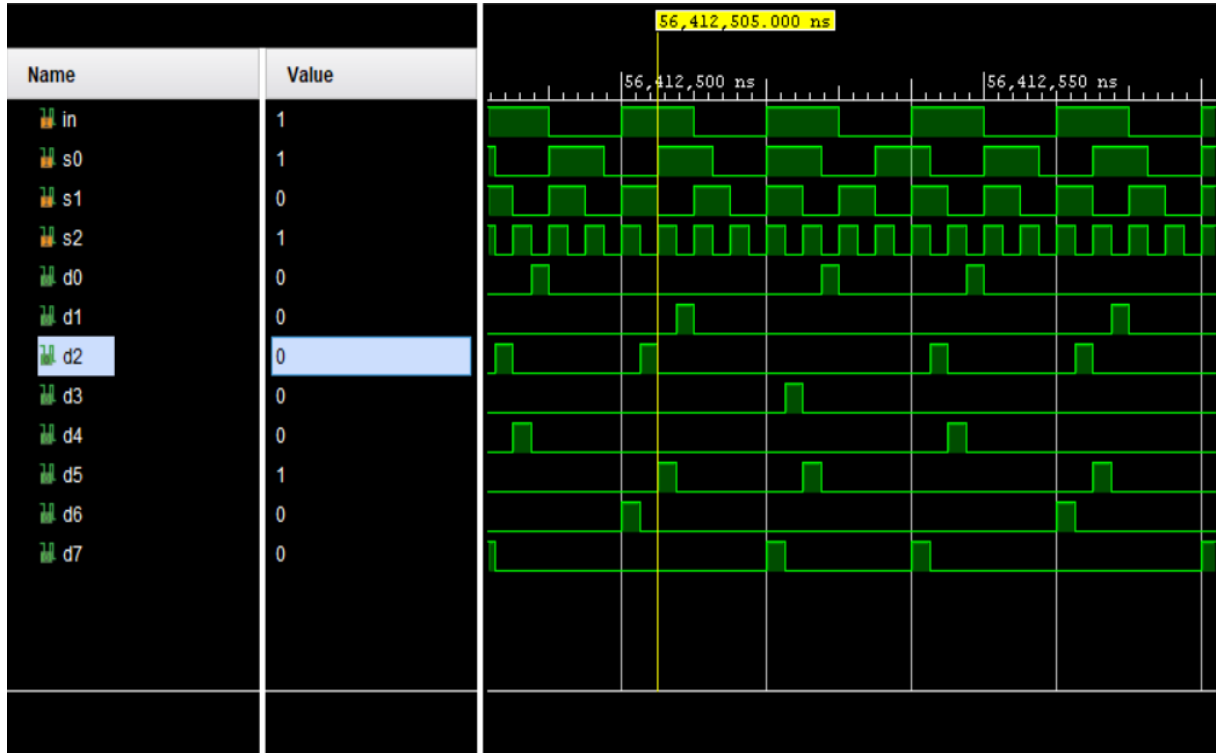
RTL Şematik



Şematik



Similasyon Sonucu



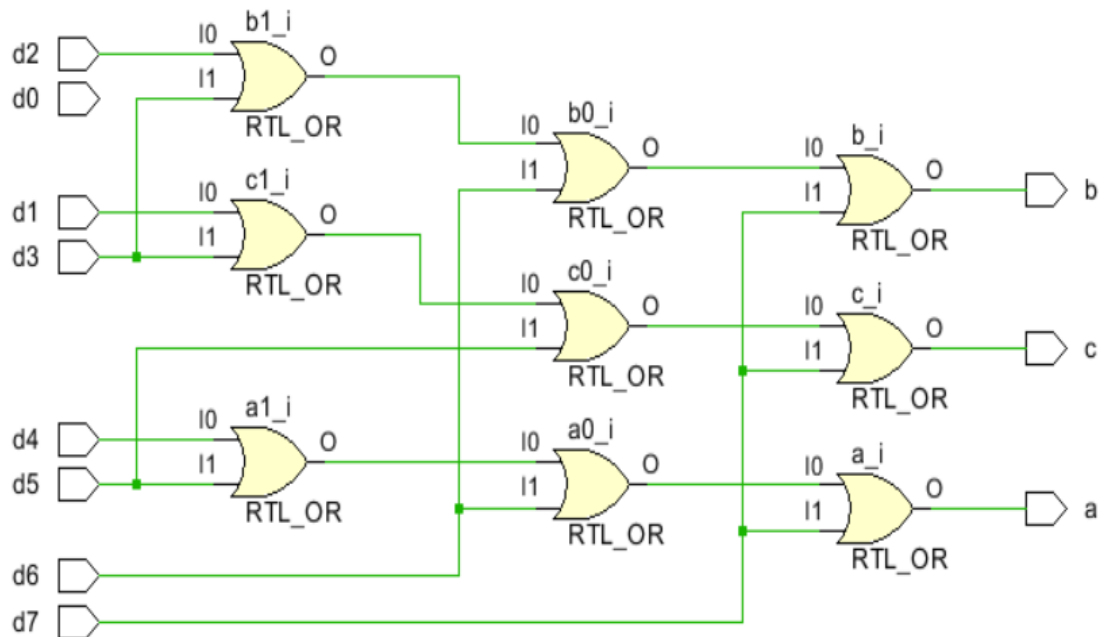
9.HAFTA

9.1 Encoder (Kodlayıcı) Devresi

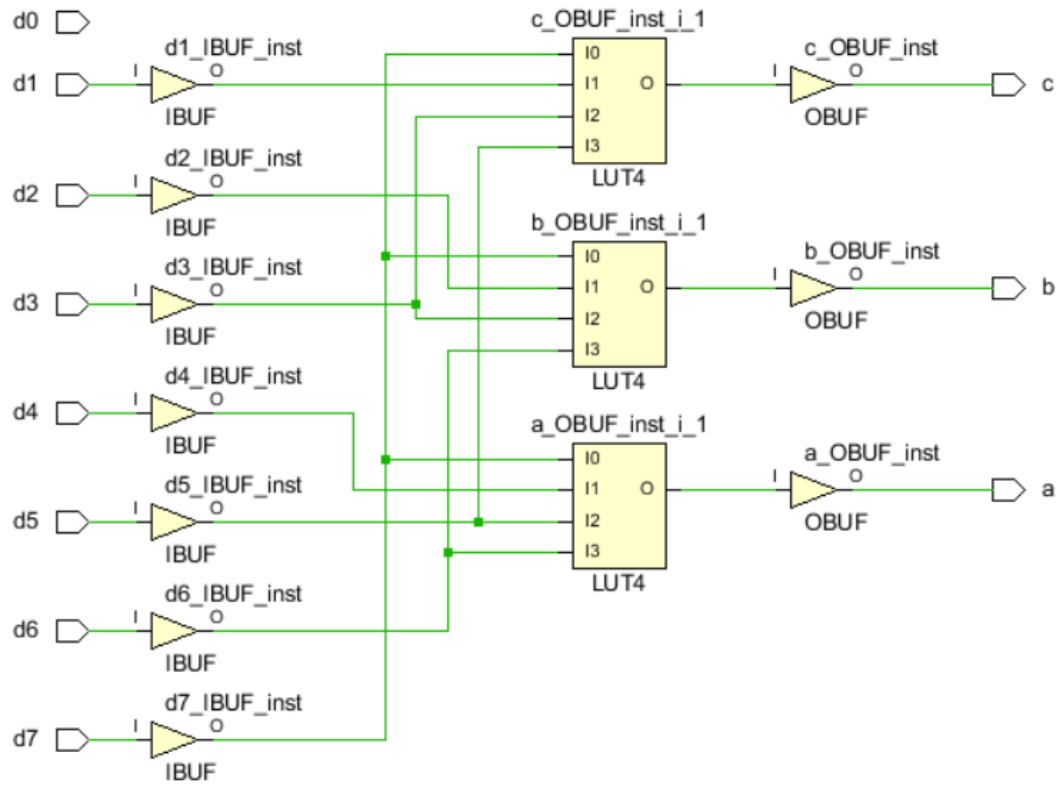
Kod

```
22 |
23 | module ozgur_Encoder(
24 |     input d0,d1,d2,d3,d4,d5,d6,d7,
25 |     output a,b,c
26 | );
27 |     or(a,d4,d5,d6,d7);
28 |     or(b,d2,d3,d6,d7);
29 |     or(c,d1,d3,d5,d7);
30 | endmodule
```

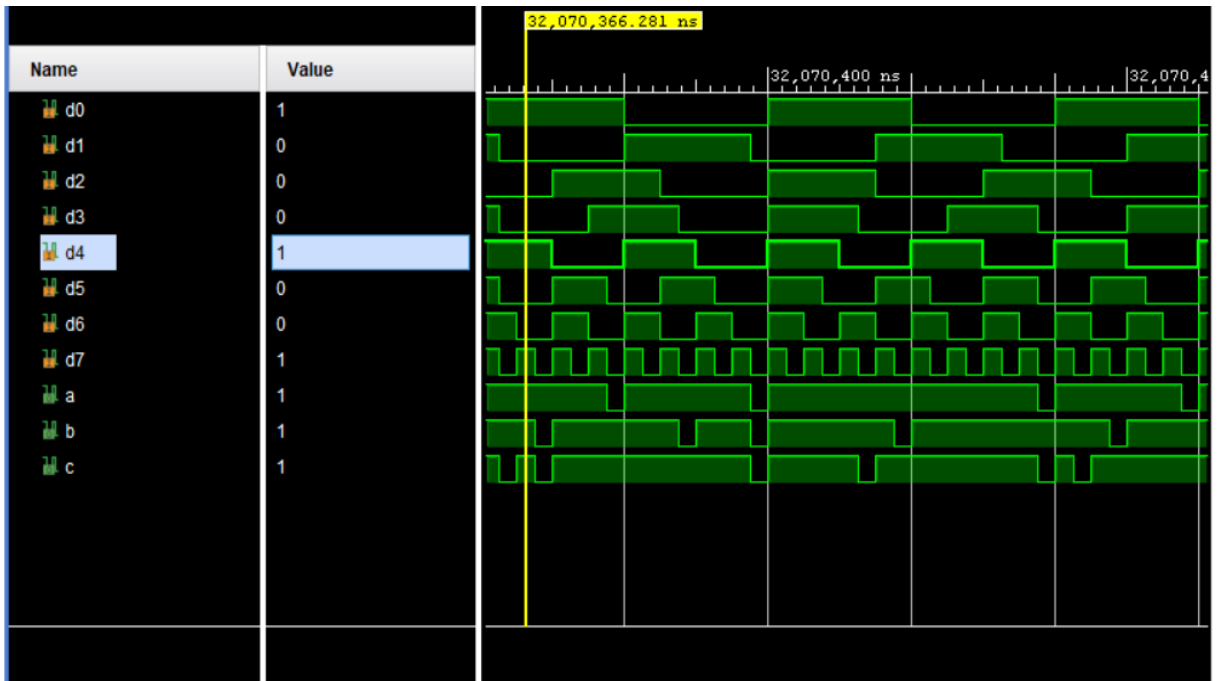
RTL Şematik



Şematik



Similasyon Sonucu

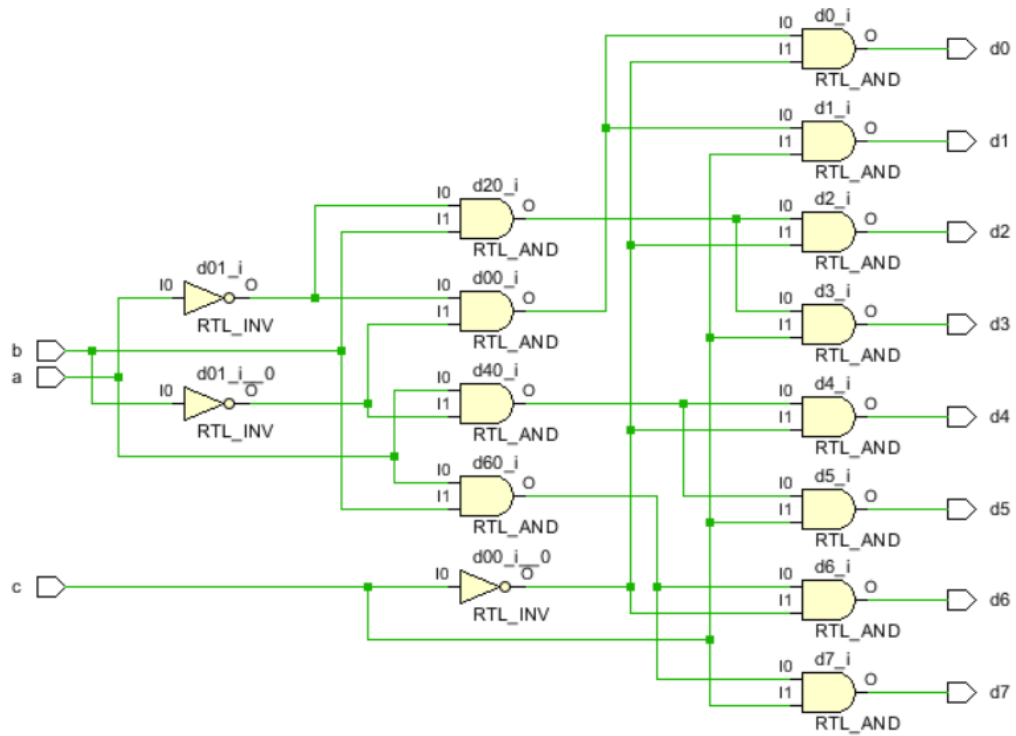


9.2 Decoder (Kod Cözücü) Devresi

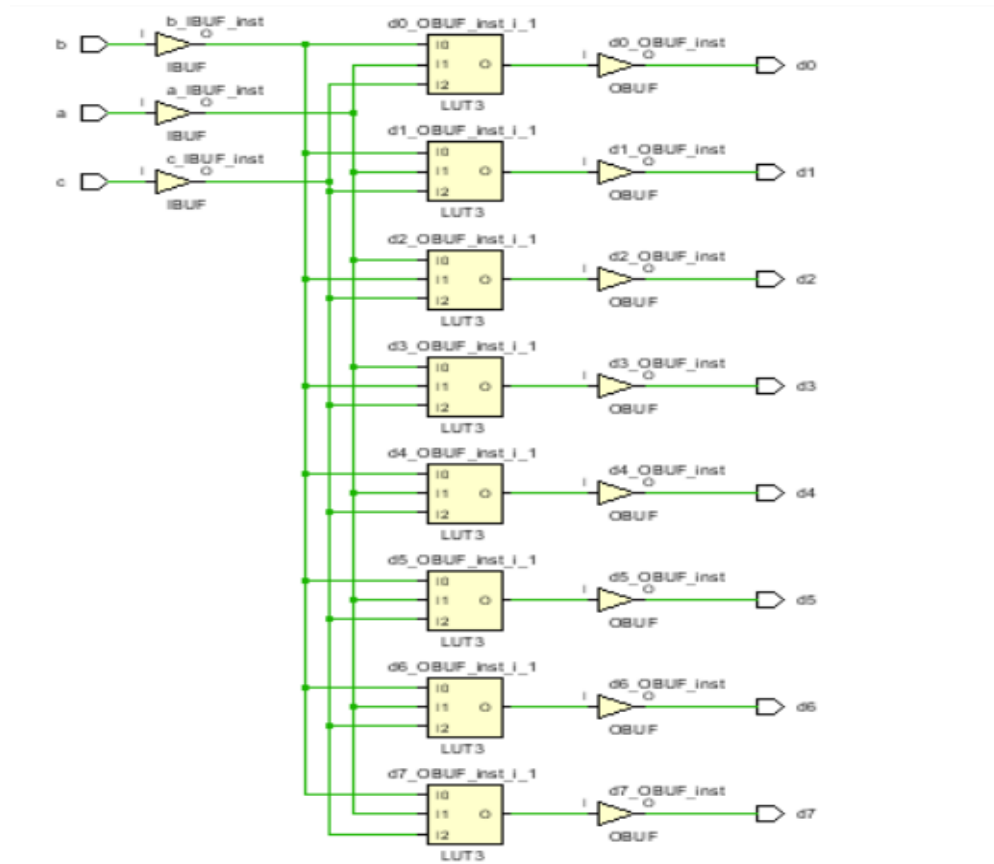
Kod

```
22
23 module ozgur_Decoder(
24     input a,b,c,
25     output d0,d1,d2,d3,d4,d5,d6,d7
26 );
27     assign d0 = (~a&~b&~c),
28            d1 = (~a&~b&c),
29            d2 = (~a&b&~c),
30            d3 = (~a&b&c),
31            d4 = (a&~b&~c),
32            d5 = (a&~b&c),
33            d6 = (a&b&~c),
34            d7 = (a&b&c);
35 endmodule
```

RTL Şematik



Şematik



Similasyon Sonucu



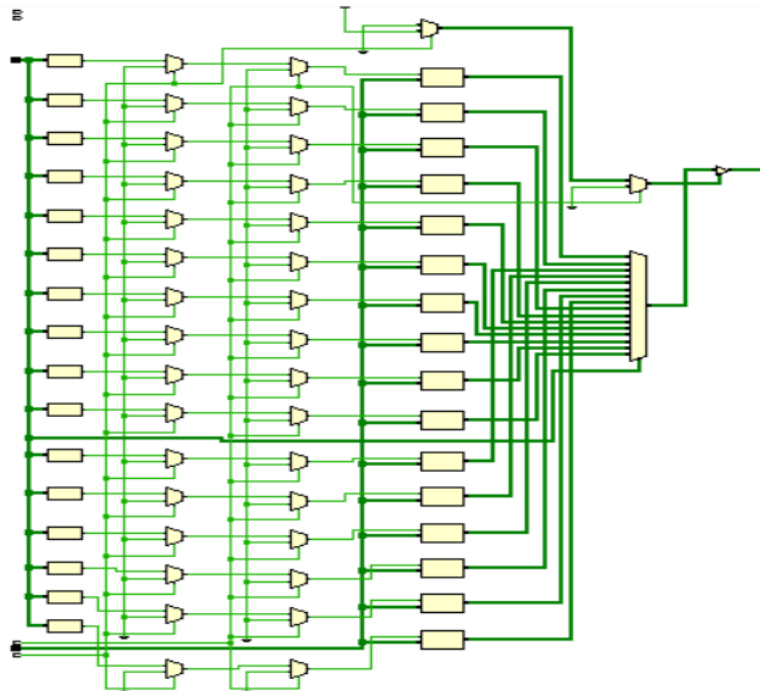
10.HAFTA

10.1 Random Access Memory (RAM) Devresi

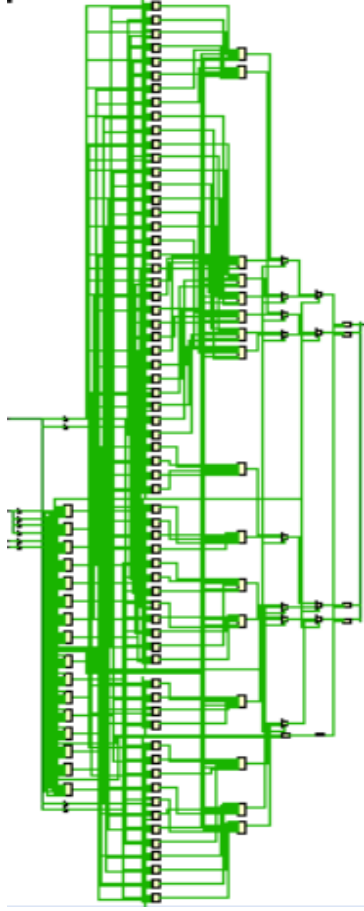
Kod

```
22 :  
23 module ozgur_ram(  
24     input [3:0] data,ekleRam,  
25     input clk,reset,cs,we,  
26     output reg [3:0] cikis  
27 );  
28     reg [3:0] ram[0:15];  
29     always@(*)  
30     begin  
31         if(cs)  
32             if(we)  
33                 begin  
34                     ram[ekleRam]=data;  
35                     cikis=4'bzzzz;  
36                 end  
37             else  
38                 cikis=ram[ekleRam];  
39             else  
40                 cikis=4'bzzzz;  
41             end  
42     endmodule
```

RTL Şematik



Şematik



Similasyon Sonucu

