



Gazi Üniversitesi
Teknoloji Fakültesi
Bilgisayar Mühendisliği Bölümü

Sayısal Elektronik Devreler Laboratuvarı
Ödevi

-Ders notunda verilen uygulamalar

Hazırlayan
Özgür Sadık Utku
181816072

11.HAFTA

11.1 Moore Makinaları

Kod

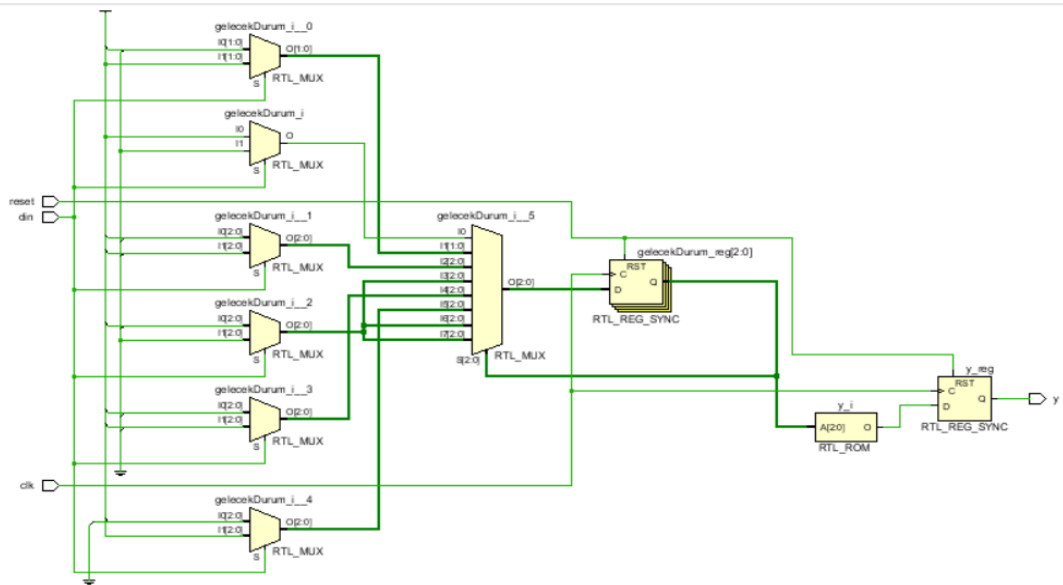
```
22 :  
23 module ozgur_Moore(din,clk,reset,y);  
24 input din,clk,reset;  
25 output reg y;  
26 parameter S0=3'b000,S1=3'b001,S2=3'b010,S3=3'b011,  
27 S4=3'b100,S5=3'b101,S6=3'b110,S7=3'b111;  
28 reg [2:0] gelecekDurum;  
29 always @(posedge clk)  
30 begin  
31     if(reset)  
32     begin  
33         gelecekDurum=S0;  
34         y=0;  
35     end  
36     else  
37     begin  
38         case(gelcekDurum)  
39         S0:begin  
40             y=0;  
41             if(din==0)  
42                 gelecekDurum=S1;  
43             else  
44                 gelecekDurum=S0;  
45         end  
46         S1:begin  
47             y=0;  
48             if(din==0)  
49                 gelecekDurum=S2;  
50             else  
51                 gelecekDurum=S3;  
52         end  
53         S2:begin  
54             y=0;  
55             if(din==0)  
56                 gelecekDurum=S4;  
57             else  
58                 gelecekDurum=S3;  
59         end  
60         S3:begin  
61             y=0;  
62             if(din==0)  
63                 gelecekDurum=S5;  
64             else  
65                 gelecekDurum=S0;  
66         end  
67         S4:begin
```

```

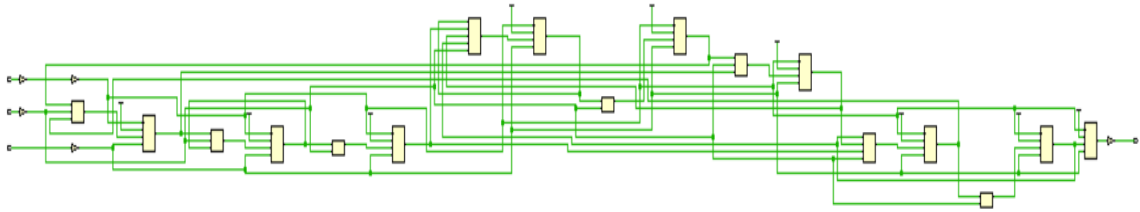
68 : y=0;
69 ⊖ if (din==0)
70 : gelecekDurum=S4;
71 : else
72 : gelecekDurum=S6;
73 : end
74 ⊖ S5:begin
75 : y=0;
76 ⊖ if (din==0)
77 : gelecekDurum=S2;
78 : else
79 : gelecekDurum=S7;
80 : end
81 ⊖ S6:begin
82 : y=0;
83 ⊖ if (din==0)
84 : gelecekDurum=S5;
85 : else
86 : gelecekDurum=S0;
87 : end
88 ⊖ S7:begin
89 : y=1;
90 ⊖ if (din==0)
91 : gelecekDurum=S5;
92 : else
93 : gelecekDurum=S0;
94 : end
95 : endcase
96 : end
97 ⊖ end
98 ⊖ endmodule

```

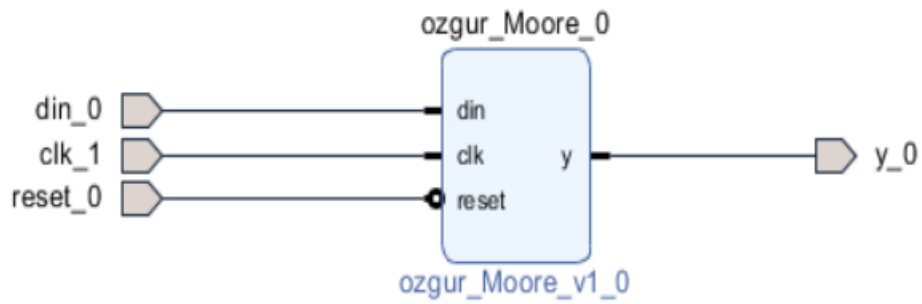
RTL Şematik



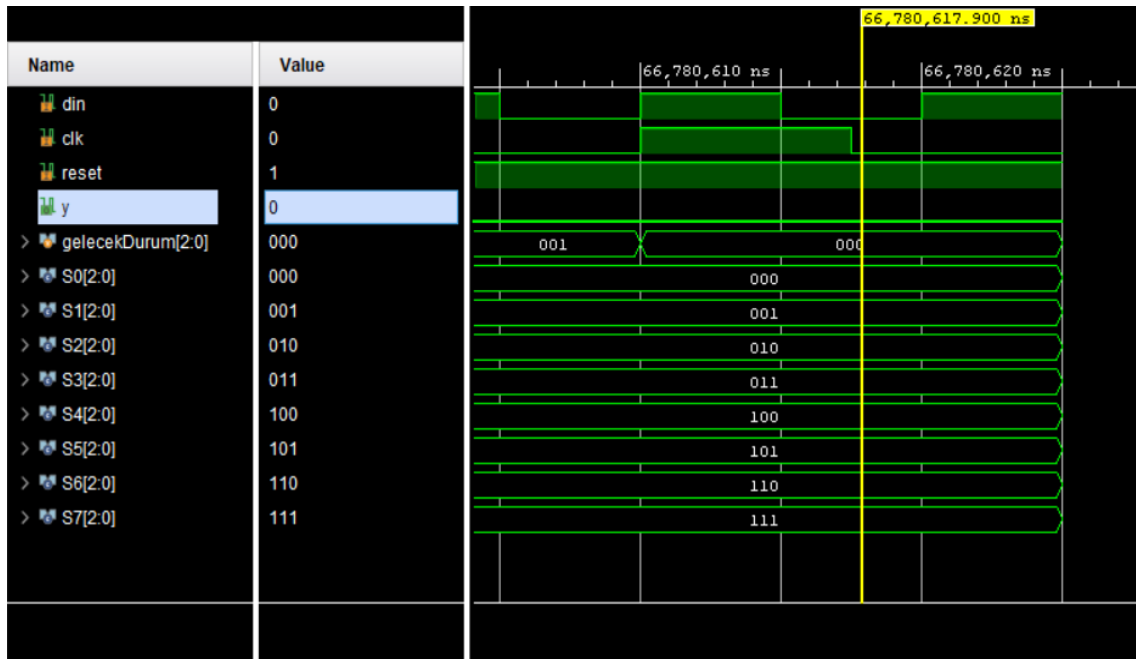
Şematik



IP Bloğu



Similasyon Sonucu



11.2 Mealy Makinaları

Kod

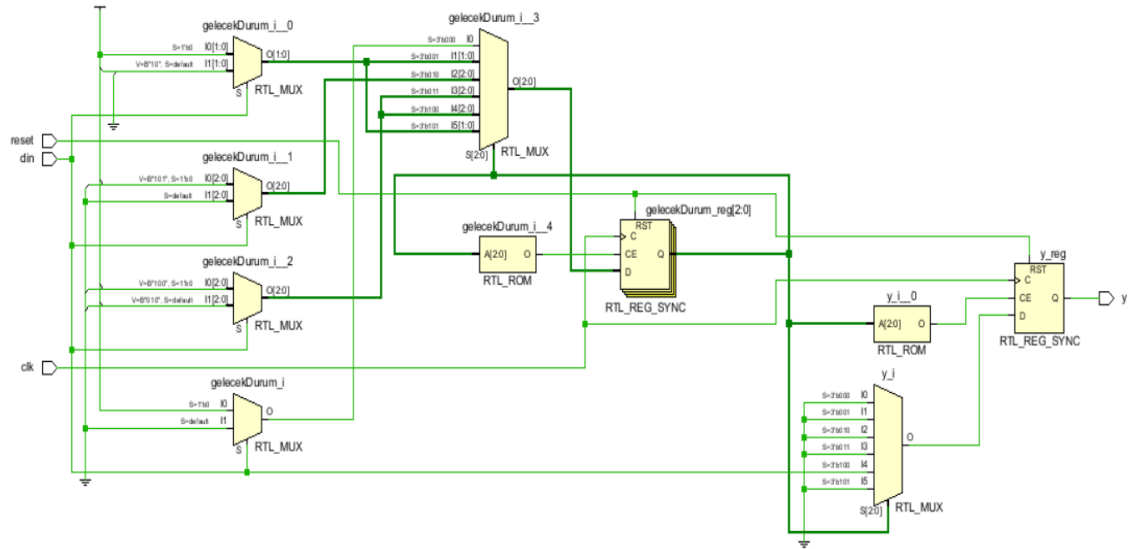
```
22 :  
23 module ozgur_Mealy(din,clk,reset,y);  
24 input din,clk,reset;  
25 output reg y;  
26 parameter S0=3'b000,S1=3'b001,S2=3'b010,S3=3'b011,  
27           S4=3'b100,S5=3'b101;  
28 reg [2:0] gelecekDurum;  
29 always @(posedge clk)  
30 begin  
31     if(reset)  
32     begin  
33         gelecekDurum=S0;  
34         y=0;  
35     end  
36     else  
37     case (gelecekDurum)  
38     S0:begin  
39         if (din==0)  
40         begin  
41             gelecekDurum=S1;  
42             y=0;  
43         end  
44         else  
45         begin  
46             gelecekDurum=S0;  
47             y=0;  
48         end  
49         end  
50     S1:begin  
51         if (din==0)  
52         begin  
53             gelecekDurum=S3;  
54             y=0;  
55         end  
56         else  
57         begin  
58             gelecekDurum=S2;  
59             y=0;  
60         end  
61         end  
62     S2:begin  
63         if (din==0)  
64         begin  
65             gelecekDurum=S5;  
66             y=0;  
67         end  
68         end  
69     end  
70     end  
71 end
```

```

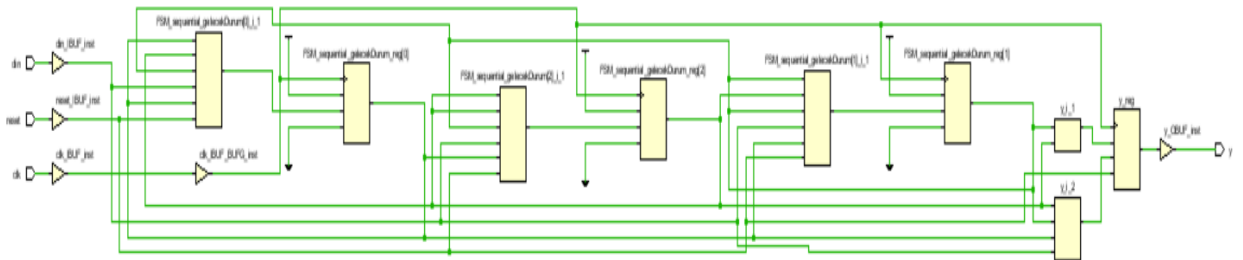
68      else
69      begin
70      gelecekDurum=S0;
71      y=0;
72      end
73      end
74      S3:begin
75      if(din==0)
76      begin
77      gelecekDurum=S4;
78      y=0;
79      end
80      else
81      begin
82      gelecekDurum=S2;
83      y=0;
84      end
85      end
86      S4:begin
87      if(din==0)
88      begin
89      gelecekDurum=S4;
90      y=0;
91      end
92      else
93      begin
94      gelecekDurum=S2;
95      y=1;
96      end
97      end
98      S5:begin
99      if(din==0)
100     begin
101     gelecekDurum=S3;
102     y=0;
103     end
104     else
105     begin
106     gelecekDurum=S2;
107     y=0;
108     end
109     end
110     endcase
111 end
112 endmodule

```

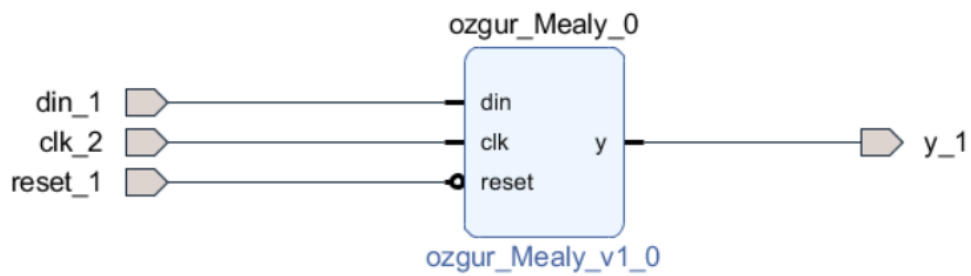
RTL Şematik



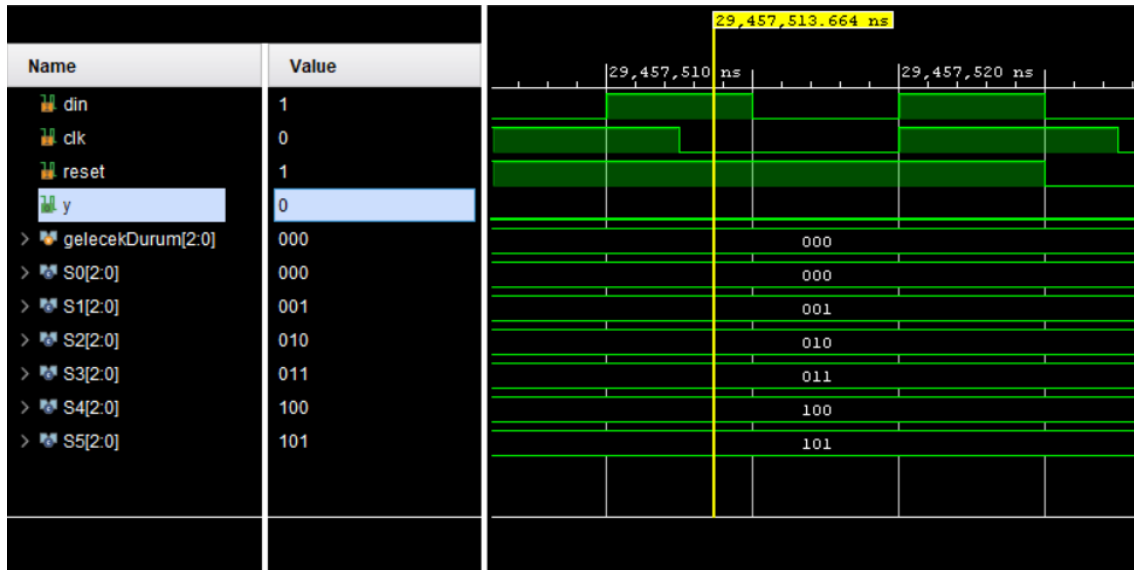
Şematik



IP Bloğu



Similasyon Sonucu



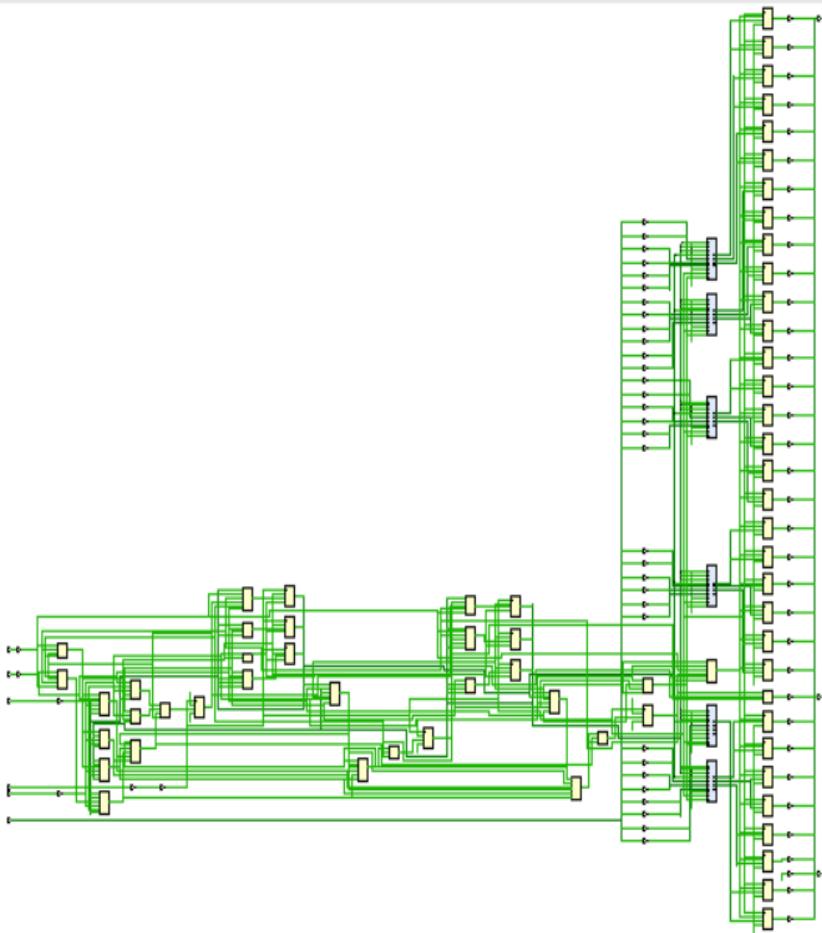
12.HAFTA

12.1 FIFO(First In First Out)

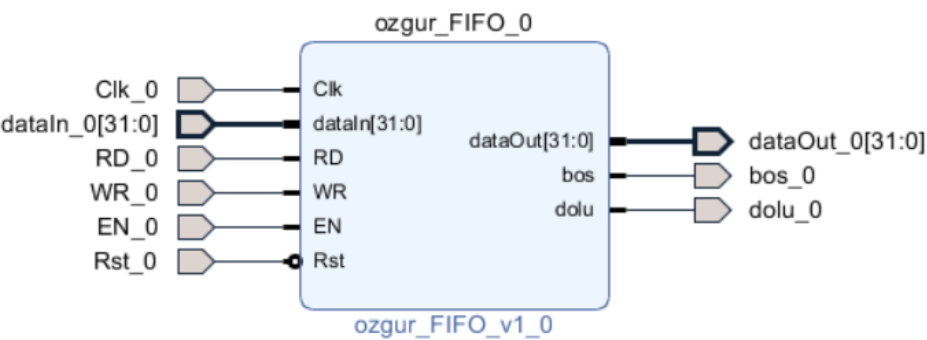
Kod

```
22 :  
23 module ozgur_FIFO(Clk,dataIn,RD,WR,EN,  
24 dataOut,Rst,bos,dolu);  
25 input Clk,RD,WR,EN,Rst;  
26 output bos,dolu;  
27 input [31:0] dataIn;  
28 output reg [31:0] dataOut;  
29 reg[2:0] Sayac=0;  
30 reg [31:0] FIFO [0:7];  
31 reg [2:0] okumaSayaci=0,yazmaSayaci=0;  
32 assign bos =(Sayac==0) ? 1'b1:1'b0;  
33 assign dolu =(Sayac==8) ? 1'b1:1'b0;  
34 always@(posedge Clk)  
35 begin  
36 if (EN==0);  
37 else begin  
38 if (Rst) begin  
39 okumaSayaci=0;  
40 yazmaSayaci=0;  
41 end  
42 else if (RD==1'b1 && Sayac!=0) begin  
43 dataOut=FIFO[okumaSayaci];  
44 okumaSayaci=okumaSayaci+1;
```

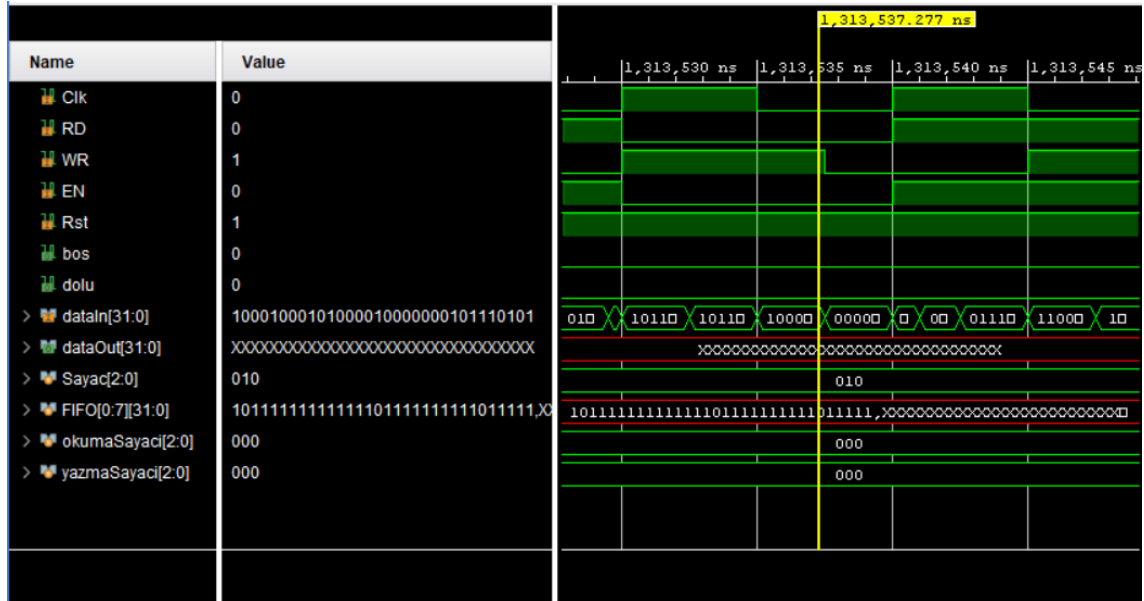

Şematik



IP Bloğu



Similasyon Sonucu



13.HAFTA

13.1 Trafik Işığı

Kod

```
22 |
23 | module ozgur_trafik_isingi(light_highway, light_farm, C, clk, rst_n);
24 |     parameter HGRE_FRED=2'b00, // Highway green and farm red
25 |         HYEL_FRED = 2'b01, // Highway yellow and farm red
26 |         HRED_FGRE=2'b10, // Highway red and farm green
27 |         HRED_FYEL=2'b11; // Highway red and farm yellow
28 |     input C, // sensor
29 |         clk, // clock = 50 MHz
30 |         rst_n; // reset active low
31 |     output reg[2:0] light_highway, light_farm; // output of lights
32 |     // fpga4student.com FPGA projects, VHDL projects, Verilog projects
33 |     reg[27:0] count=0,count_delay=0;
34 |     reg delay10s=0, delay3s1=0,delay3s2=0,RED_count_en=0,YELLOW_count_en1=0,YELLOW_count_en2=0;
35 |     wire clk_enable; // clock enable signal for 1s
36 |     reg[1:0] state, next_state;
37 |     // next state
38 |     always @(posedge clk or negedge rst_n)
39 |     begin
40 |         if(~rst_n)
41 |             state <= 2'b00;
42 |         else
43 |             state <= next_state;
44 |     end
```

```

45 // FSM
46 always @(*)
47 begin
48 case(state)
49 HGRE_FRED: begin // Green on highway and red on farm way
50     RED_count_en=0;
51     YELLOW_count_en1=0;
52     YELLOW_count_en2=0;
53     light_highway = 3'b001;
54     light_farm = 3'b100;
55     if(C) next_state = HYEL_FRED;
56     // if sensor detects vehicles on farm road,
57     // turn highway to yellow -> green
58     else next_state =HGRE_FRED;
59 end
60 HYEL_FRED: begin// yellow on highway and red on farm way
61     light_highway = 3'b010;
62     light_farm = 3'b100;
63     RED_count_en=0;
64     YELLOW_count_en1=1;
65     YELLOW_count_en2=0;
66     if(delay3s1) next_state = HRED_FGRE;
67     // yellow for 3s, then red
68     else next_state = HYEL_FRED;
69 end
70 HRED_FGRE: begin// red on highway and green on farm way
71     light_highway = 3'b100;
72     light_farm = 3'b001;
73     RED_count_en=1;
74     YELLOW_count_en1=0;
75     YELLOW_count_en2=0;
76     if(delay10s) next_state = HRED_FYEL;
77     // red in 10s then turn to yello -> green again for high way
78     else next_state =HRED_FGRE;
79 end
80 HRED_FYEL:begin// red on highway and yellow on farm way
81     light_highway = 3'b100;
82     light_farm = 3'b010;
83     RED_count_en=0;
84     YELLOW_count_en1=0;
85     YELLOW_count_en2=1;
86     if(delay3s2) next_state = HGRE_FRED;
87     // turn green for highway, red for farm road
88     else next_state =HRED_FYEL;
89 end
90 default: next_state = HGRE_FRED;

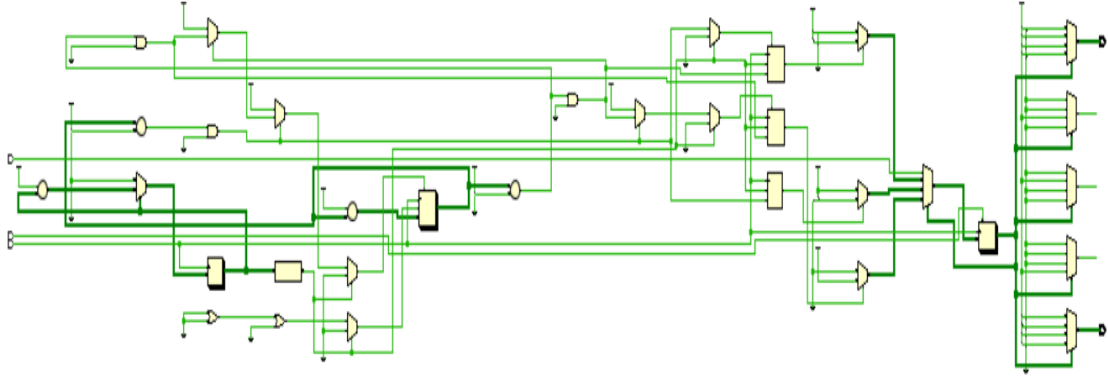
```

```

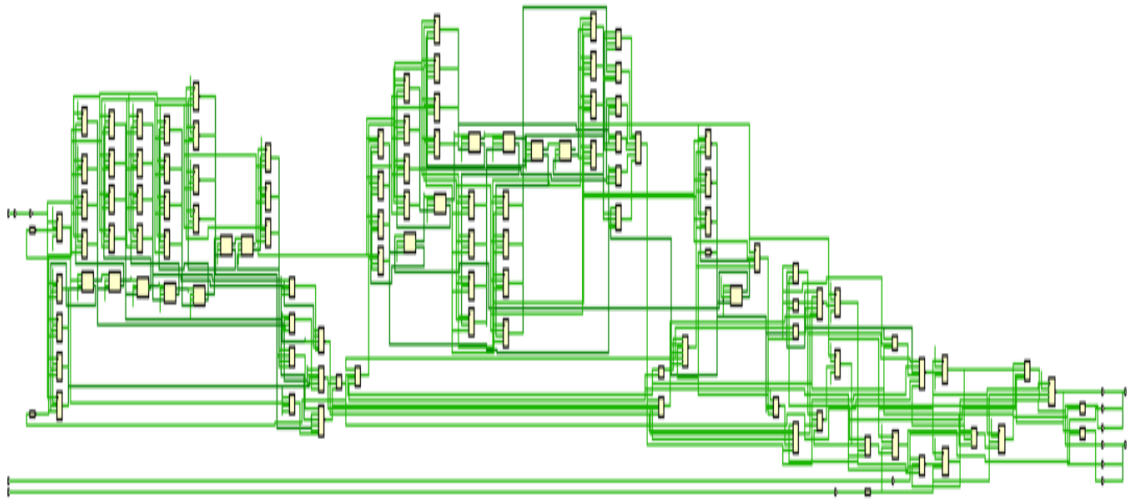
91 | endcase
92 | end
93 | // fpga4student.com FPGA projects, VHDL projects, Verilog projects
94 | // create red and yellow delay counts
95 | always @(posedge clk)
96 | begin
97 |     if(clk_enable==1) begin
98 |         if(RED_count_en||YELLOW_count_en1||YELLOW_count_en2)
99 |             count_delay <=count_delay + 1;
100 |             if((count_delay == 9)&&RED_count_en)
101 |                 begin
102 |                     delay10s=1;
103 |                     delay3s1=0;
104 |                     delay3s2=0;
105 |                     count_delay<=0;
106 |                 end
107 |             else if((count_delay == 2)&&YELLOW_count_en1)
108 |                 begin
109 |                     delay10s=0;
110 |                     delay3s1=1;
111 |                     delay3s2=0;
112 |                     count_delay<=0;
113 |                 end
114 |             else if((count_delay == 2)&&YELLOW_count_en2)
115 |                 begin
116 |                     delay10s=0;
117 |                     delay3s1=0;
118 |                     delay3s2=1;
119 |                     count_delay<=0;
120 |                 end
121 |             else
122 |                 begin
123 |                     delay10s=0;
124 |                     delay3s1=0;
125 |                     delay3s2=0;
126 |                 end
127 |             end
128 |         end
129 |         // create 1s clock enable
130 |         always @(posedge clk)
131 |         begin
132 |             count <=count + 1;
133 |             //if(count == 50000000) // 50,000,000 for 50 MHz clock running on real FPGA
134 |             if(count == 3) // for testbench
135 |                 count <= 0;
136 |         end
137 |         assign clk_enable = count==3 ? 1: 0; // 50,000,000 for 50MHz running on FPGA
138 |     endmodule

```

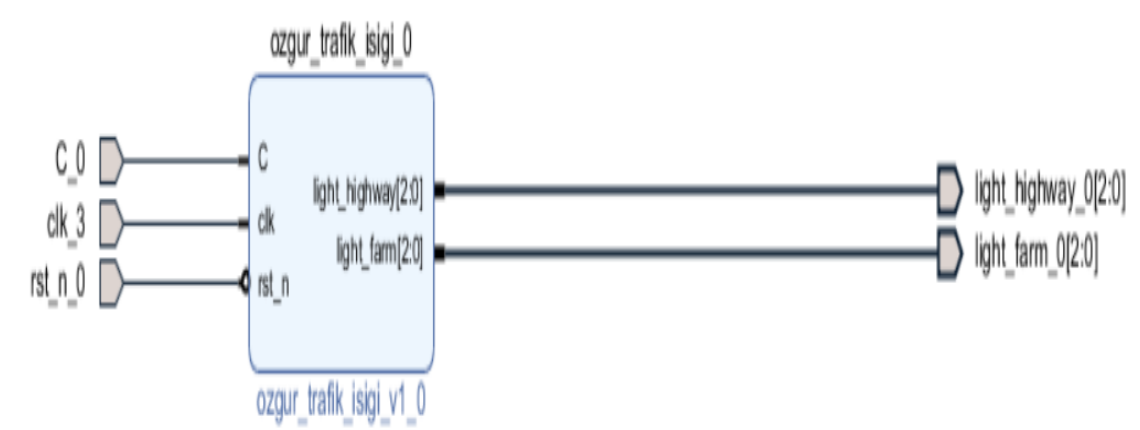
RTL Şematik



Şematik



IP Bloğu

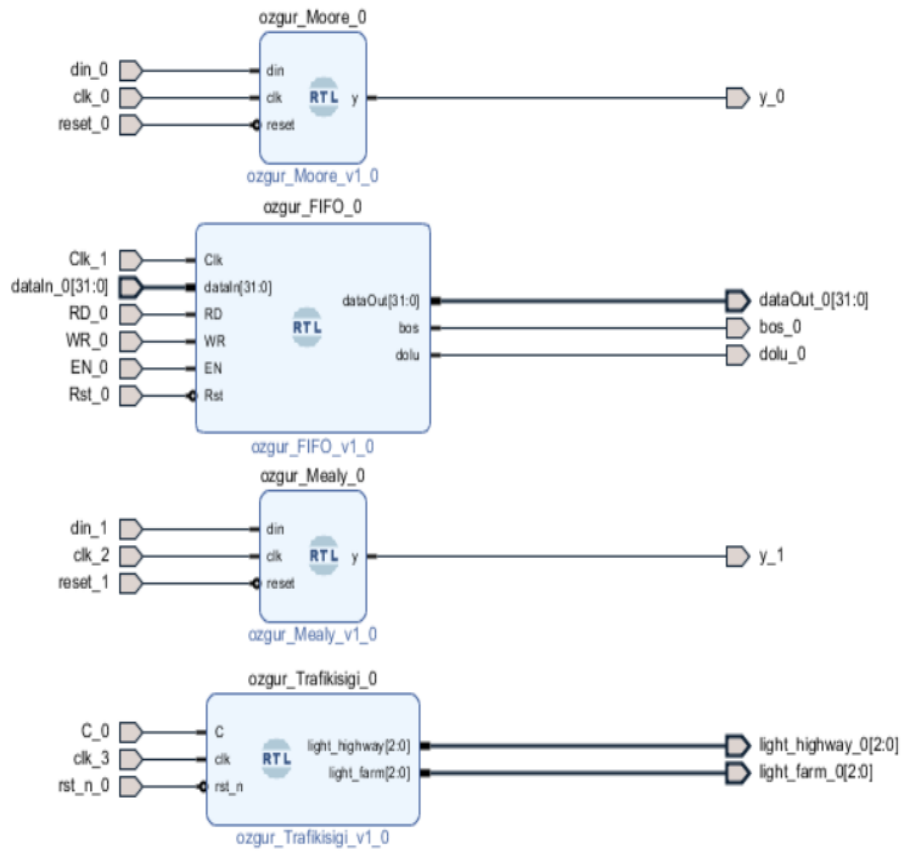


Similasyon Sonucu



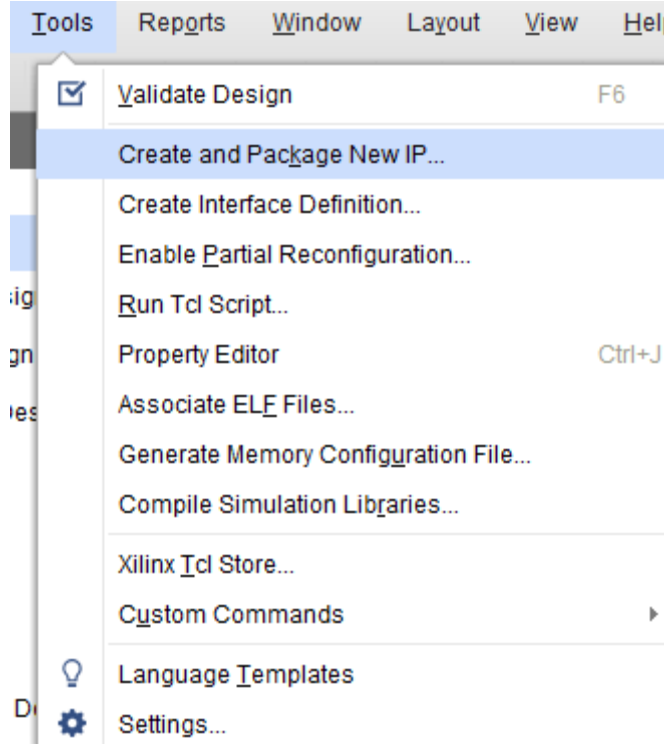
15.HAFTA

IP Bloğu

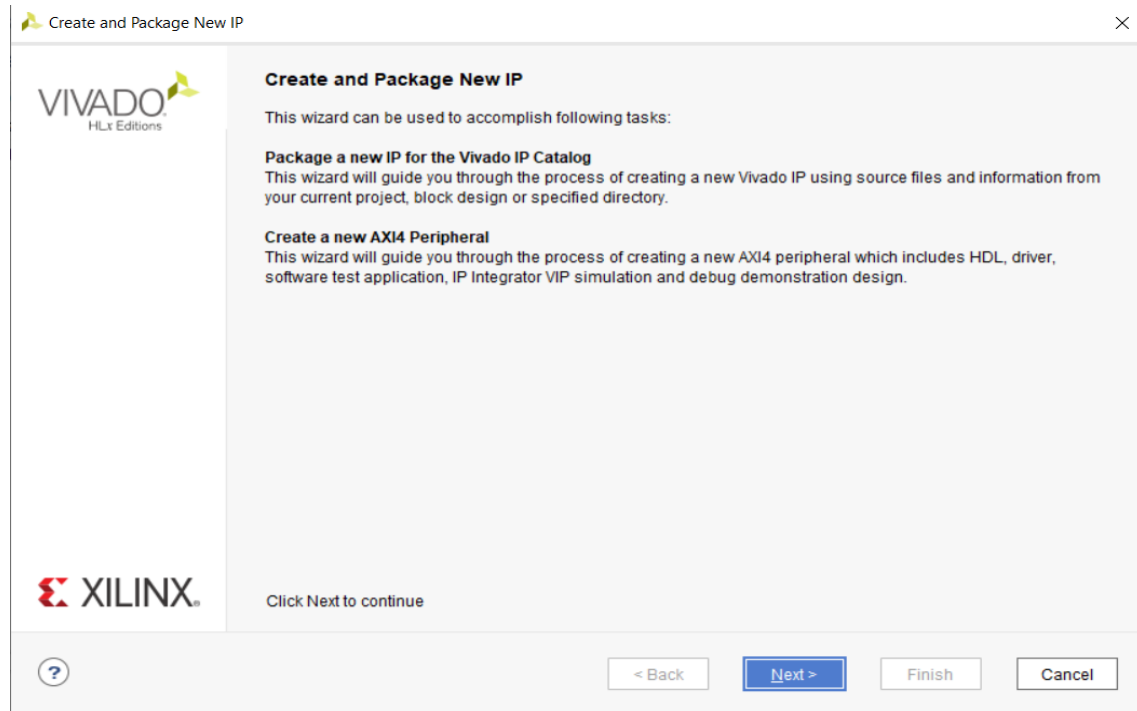


IP Nasıl Kaydedilir


Yeni IP paketi oluşturulur.



Next diyerek devam edilir.



Package your current project seçilir ve next diyerek devam edilir

 Create and Package New IP ×

Create Peripheral, Package IP or Package a Block Design
Please select one of the following tasks.

Packaging Options

☒ Package your current project
Use the project as the source for creating a new IP Definition.

☐ Package a block design from the current project
Choose a block design as the source for creating a new IP Definition.
Select a block design:


☐ Package a specified directory
Choose a directory as the source for creating a new IP Definition.

Create AXI4 Peripheral

☐ Create a new AXI4 peripheral
Create an AXI4 IP, driver, software test application, IP Integrator AXI4 VIP simulation and debug demonstration design.

? < Back Next > Finish Cancel

Include xci files seçeneği seçilir ve next diyerek devam edilir.

 Create and Package New IP ×

Package Your Current Project
Select the directory where the IP Definition will be created and the associated options for packaging the current project.

IP location:

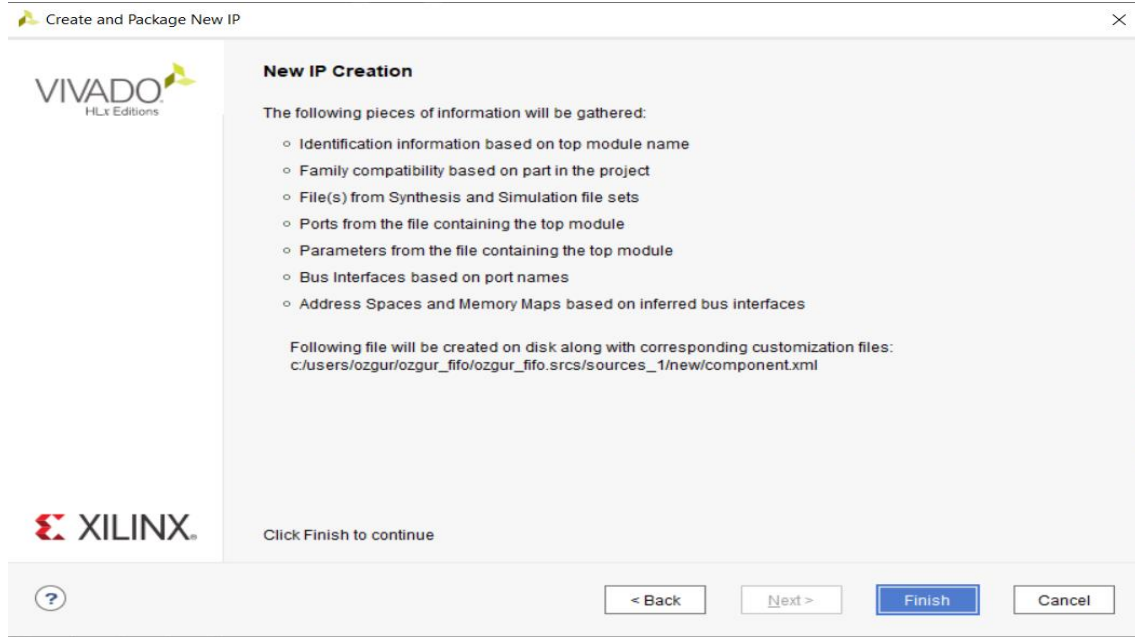
Packaging IP in the project

☒ Include .xci files

☐ Include IP generated files

? < Back Next > Finish Cancel

Finish butonuna basilarak IP oluřturulması tamamlanır.



Açılan pencerede “Review and Package” bölümünden “Package IP” yapılır ve kaydedilir.

