GEBZE TEKNİK

ÜNİVERSİTESİ

ELEKTRONİK

MÜHENDİSLİĞİ

ELM235

LOJİK DEVRE

TASARLABORATUVARI

LAB 7 Deney Raporu

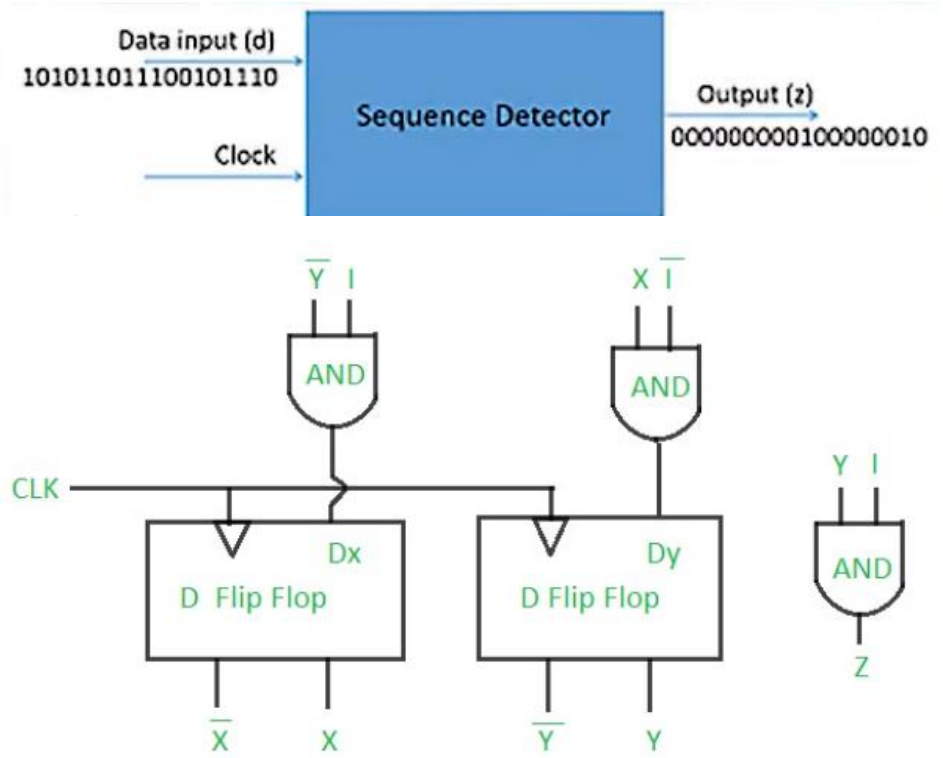
Sonlu Otomatlar-||

Hazırlayanlar
1) 200102002031 – Beyza Duran
2) 200102002043 – Senanur Ağaç

1. Teorik Araştırma

SEQUENCE DETECTOR

Bir dizi detektörü, bir dizi giriş biti alan ve hedef dizi algılandığında bir çıktı 1 üreten bir sıralı durum makinesidir. Bir Mealy makinesinde çıktı, mevcut duruma ve harici girdiye (x) bağlıdır.
Dizi algılama, önceden tanımlanmış bir dizi girdiyi tanıma eylemidir. Dizi dedektörü, temel olarak bilgi depolayabilen bir devre olan sıralı bir devredir. Sıralı devreler için iki ana model: Mealy ve Moore modeli. Bir Mealy model devresinde çıkış, girişlere ve sistemin durumuna bağlıdır, Moore modelinde, sistemin çıkışı yalnızca durumuna bağlıdır.



Bir dizi dedektörü , bir giriş bit dizisini alan ve hedef dizi algılandığında bir çıktı 1 üreten bir sıralı durum makinesidir. Bir Mealy makinesinde çıktı, mevcut duruma ve harici girdiye (x) bağlıdır. Bu nedenle, diyagramda çıktı, girdilerle birlikte durumların dışında yazılmıştır. Sıra dedektörü iki tiptir:

1.Örtüşen
2.Örtüşmeyen

Bir örtüşen dizi detektöründe, bir dizinin son biti, bir sonraki dizinin ilk biti olur. Bununla birlikte, örtüşmeyen bir dizi detektöründe, bir dizinin son biti, bir sonraki dizinin ilk biti olmaz.

1. Problemler

Problem 1- Sequence Detector

Bir girişi (A), bir çıkışı olan bir devrede, giriş sinyalinden gelen her 101 dizisi (iç içe gelenler dahil) algıladığında çıkışı (Y) 1 yapan bir FSM devresi için,

A. State Transition Diagramınızı çıkarın
B. State Table'ını oluşturun.
C. Şematik tasarımını yapın.
D. Yaptığınız devreyi Quartus'ta tasarlayın, testbench'ini oluşturun ve benzetimini yapın.

2.1.1 Problemin Çözümü

```
/* Hazırlayanlar: Senanur Ağaç – Beyza
Duran Lab6 problem 1 Sequence Detector
*/
module lab7_g15_p1 ( input logic in,clk,reset,
output logic  myout);
logic out;

typedef enum {s0,s1,s2} statetype ;
statetype state,nextstate ;

always_ff @(posedge clk)
if(reset) state<=s0;
else state<=nextstate ;

always_ff
begin

case(state)

s0: begin
if(in)begin
nextstate=s1;
end
else begin
nextstate=s0;
end
end
```

1

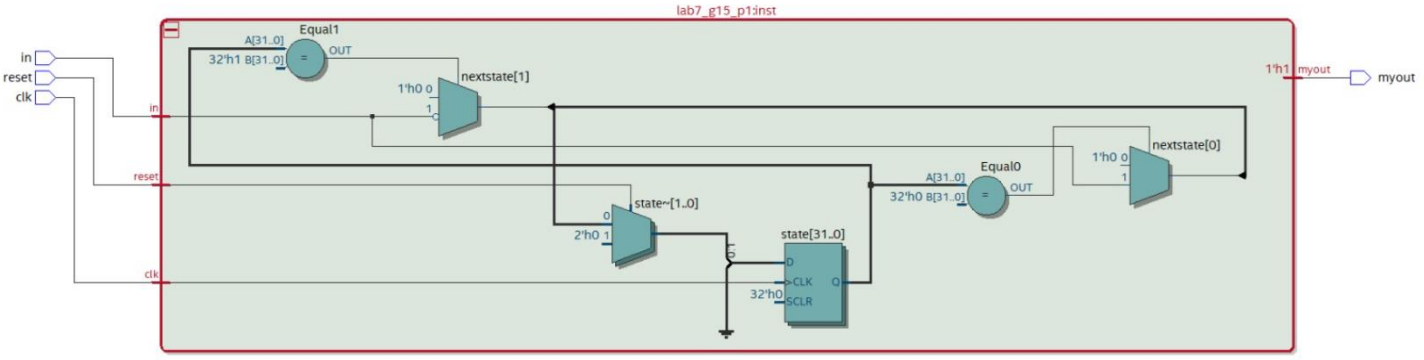
```
s1: begin
if(in)begin
nextstate=s0;
end
else begin
nextstate=s2;
end
end

s2: begin
if(in)begin
nextstate=s0;
out=1;
end
else begin
nextstate=s0;
end
end


default:
nextstate=s0;
endcase

end

always_ff
begin
myout=out;
end

endmodule
```

2

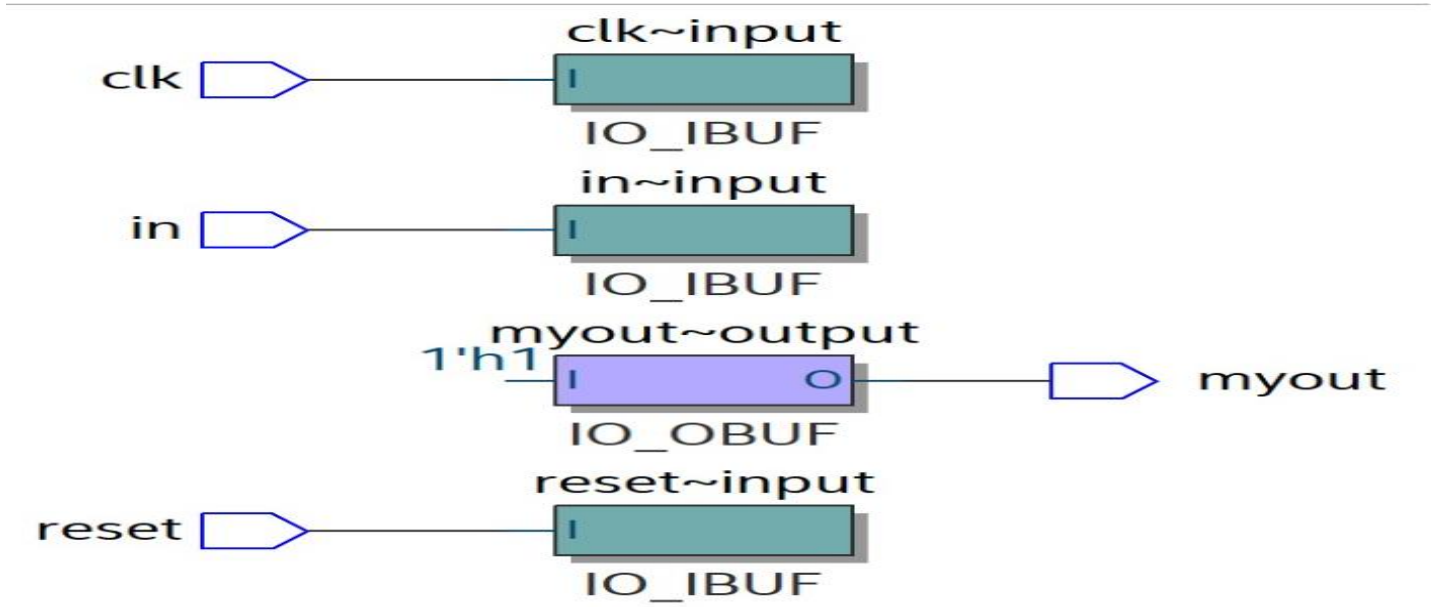
```
/* Hazırlayanlar: Senanur Ağaç – Beyza
Duran Lab6 problem 1 Sequence Detector -
Testbench*/
`timescale 1ns / 1ps
module tb_lab7_g15_p1();
logic in;
logic clk;
logic reset;
logic myout;

lab7_g15_p1 dene(in,clk,reset,myout);

integer k=0;

initial
begin

reset=0;

   for(k=0; k < 17; k=k+1)
   begin
      #5; clk=0;
      #5; clk=1;
      in=$urandom_range(0,1);
   end
 #20;
$stop;

end
endmodule
```

Yukarıda kodu yazılan Mealy sequence detectorun çalışma mantığı şu şekildedir:
Öncelikle bu bir örtüşen dizi detektörüdür. İlk statede bizden istenilen durum 1 durumunun oluşmasıdır. Eğer bu durum oluşursa next state durumunu geçilir. Yani A dan B state ine geçilir. 0 gelir ise kendi içinde döngüsü devam eder. B state ine geçildikten sonra istenilen durum 0 gelmesidir. 0 durumu oluşursa C state ine geçilir. 1 gelir ise kendi içinde 0 gelene kadar döngüde kalır. C statetinde recursive bir durum vardır. Bu durumda 1 gelirse B stateine geri gelinir. 0 gelir ise A state ine girilir ve bu döngü deveam eder.

Şekil 1: Dalga şeması



Şekil 1:RTL şeması



Şekil 3:1.soru için Post-mapping şeması

Flow Summary

<<Filter>>

Flow Status	Successful - Sun May 15 12:04:05 2022
Quartus Prime Version	19.1.0 Build 670 09/22/2019 SJ Lite Edition
Revision Name	labb7_top
Top-level Entity Name	labb7_top
Family	MAX 10
Device	10M08DAF484C8G
Timing Models	Final
Total logic elements	1 / 8,064 (< 1 %)
Total registers	0
Total pins	4 / 250 (2 %)
Total virtual pins	0
Total memory bits	0 / 387,072 (0 %)
Embedded Multiplier 9-bit elements	0 / 48 (0 %)
Total PLLs	0 / 2 (0 %)
UFM blocks	0 / 1 (0 %)
ADC blocks	0 / 1 (0 %)

Şekil 4: 1.soru için akış şeması

Analysis & Synthesis Resource Usage Summary

<<Filter>>

	Resource	Usage
1	-- 4 input functions	0
2	-- 3 input functions	0
3	-- <=2 input functions	0
4		
5	Logic elements by mode	
1	-- normal mode	0
2	-- arithmetic mode	0
6		
7	Total registers	0
1	-- Dedicated logic registers	0
2	-- I/O registers	0
8		
9	I/O pins	4
10		
11	Embedded Multiplier 9-bit elements	0
12		
13	Maximum fan-out node	myo...put
14	Maximum fan-out	1
15	Total fan-out	4
16	Average fan-out	0.50

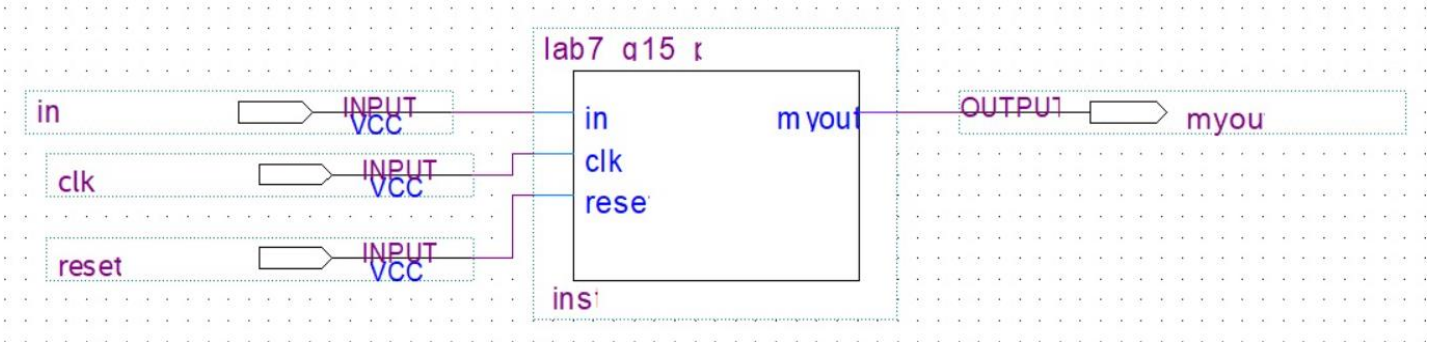
Şekil 5: Kaynak kullanımı özet şeması

Analysis & Synthesis Resource Utilization by Entity

<<Filter>>

	Compilation Hierarchy Node	Combinational ALUTs	Dedicated Logic Registers	Memory Bits	UFM Blocks	DSP Elements	DSP 9
1	|labb7_top	0 (0)	0 (0)	0	0	0	0

Şekil 6: Utilization raporu



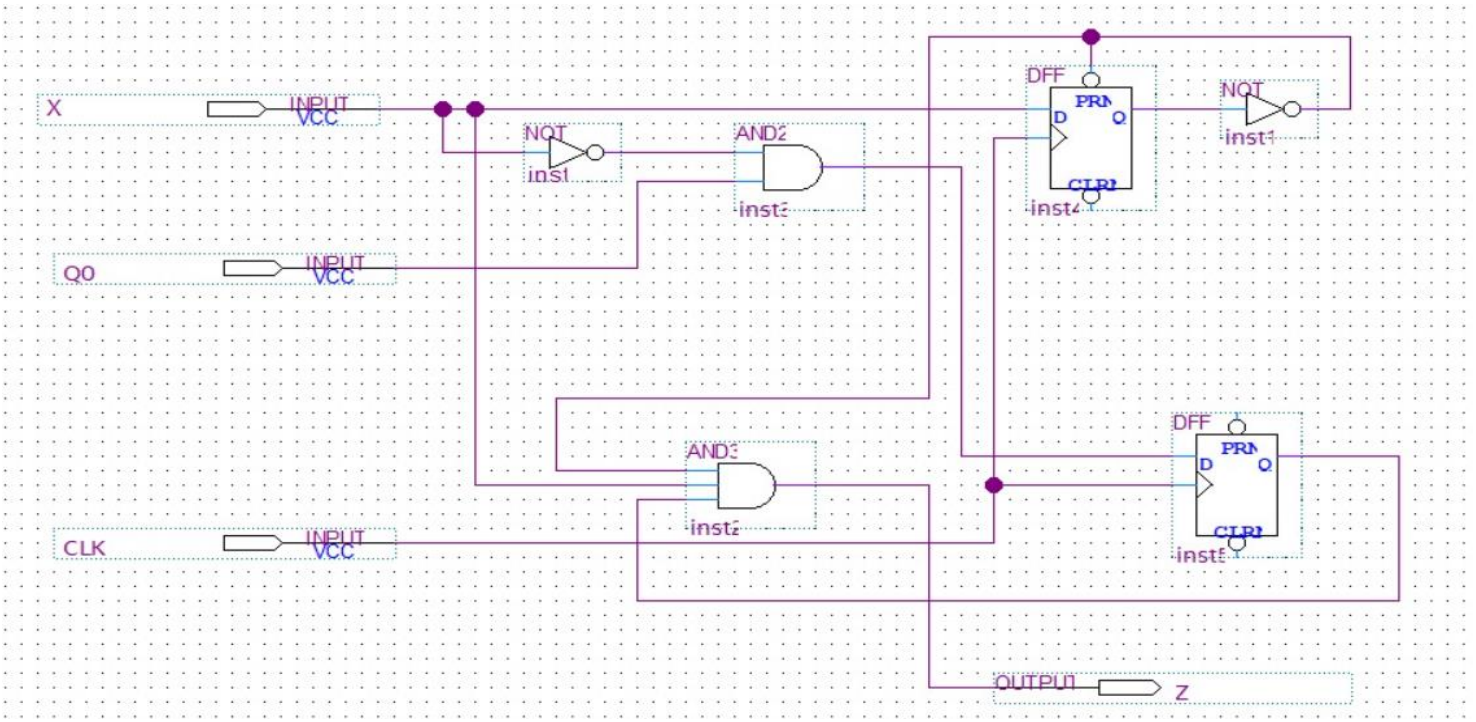
Şekil 7: 1.devre için devre şeması

Slow 1200mV 85C Model Fmax Summary

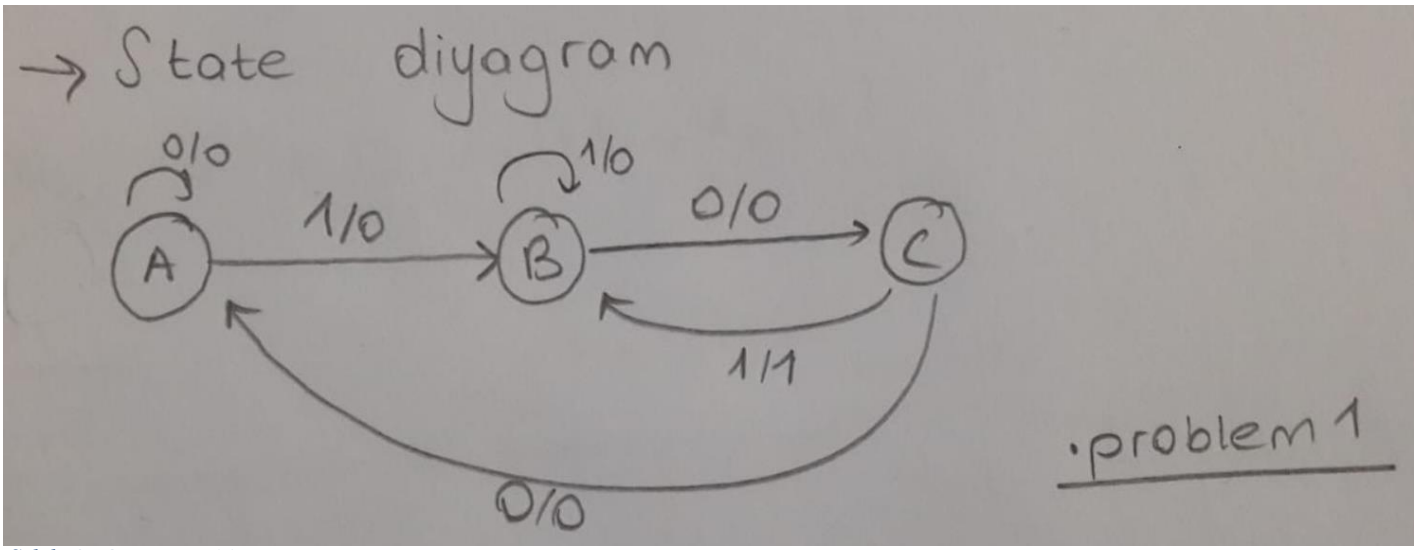
<<Filter>>

	Fmax	Restricted Fmax	Clock Name	Note
1	665.78 MHz	250.0 MHz	clk	limit due to minimum period restriction (max I/O toggle rate)

Şekil 8: 1.soru için FMAX değeri



Şekil 9: 1.soru için flip-flop devresi



Şekil 10 : State Transition

Tablo 1: State1

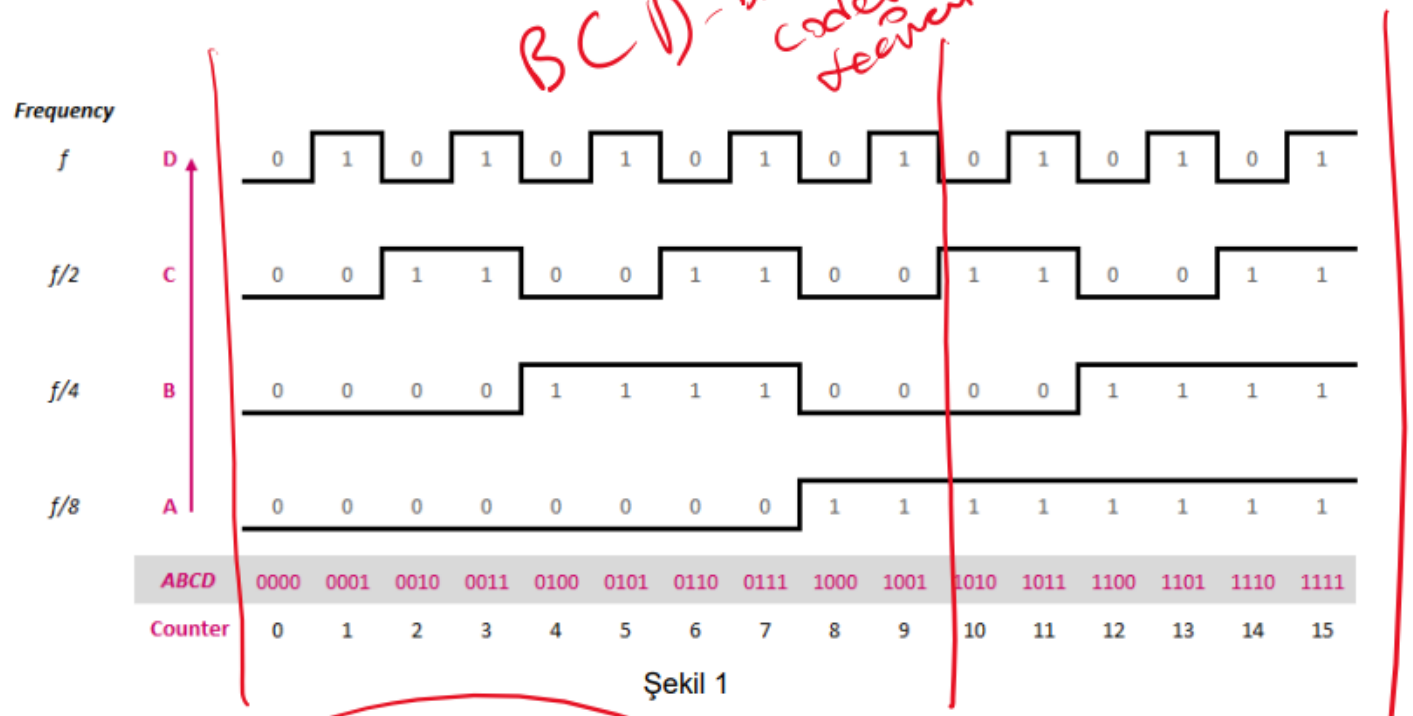
Önce	Giriş	Sonra	Çıkış
S_0	0	S_0	0
S_0	1	S_0	0
S_1	0	S_1	0
S_1	1	S_1	0
S_2	0	S_2	0
S_2	1	S_2	1

→

Önce	Giriş	Sonra	Çıkış
Q1Q0	x	Q1'Q0'	z
00	0	00	0
00	1	00	0
01	0	01	0
01	1	01	0
10	0	10	0
10	1	10	1

Problem 2 – Counters

4 bitlik Binary ve BCD senkron sayıcılar için,



Şekil 1

A. State Transition Diagramınızı çıkarın
B. Bu statelerinize göre devrenizi flip-flop kullanarak tasarlayınız.
C. Tasarladığınız devreleri şematik seviyesinde test ediniz.
D. Yaptığınız devreleri Quartus'ta tasarlayın, testbench'ini oluşturun ve benzetimini yapın.

```
/* Hazirlayanlar : Senanur Agaç - Beyza Duran */

module lab7_g15_p2 ( input logic clk,reset,
output logic [3:0] myout);

logic [3:0] count;

typedef enum
{s0,s1,s2,s3,s4,s5,s6,s7,s8,s9,s10,s11,s12,s13,s14,
s15} statetype ;
statetype state,nextstate ;

always_ff @(posedge clk)
if(reset) state<=s0;
else state<=nextstate ;

always_comb
begin
```

1

```
case(state)

s0: begin
count= 4'b0000;
nextstate=s1;
end

s1: begin
count= 4'b0001;
nextstate=s2;
end

s2: begin
count= 4'b0010;
nextstate=s3;
end
```

2

```
s3: begin
count= 4'b0011;
nextstate=s4;
end

s4: begin
count= 4'b0100;
nextstate=s5;
end

s5: begin
count= 4'b0101;
nextstate=s6;
end

s6: begin
count= 4'b0110;
nextstate=s7;
end

s7: begin
count= 4'b0111;
nextstate=s8;
end
```

3

```
s8: begin
count= 4'b1000;
nextstate=s9;
end

s9: begin
count= 4'b1001;
nextstate=s10;
end

s10: begin
count= 4'b1010;
nextstate=s11;
end

s11: begin
count= 4'b1011;
nextstate=s12;
end

s12: begin
count= 4'b1100;
nextstate=s13;
end
```

4

```
s13: begin
count= 4'b1101;
nextstate=s14;
end

s14: begin
count= 4'b1110;
nextstate=s15;
end

s15: begin
count= 4'b1111;
nextstate=s0;
end

default:
nextstate=s0;
endcase

end
```

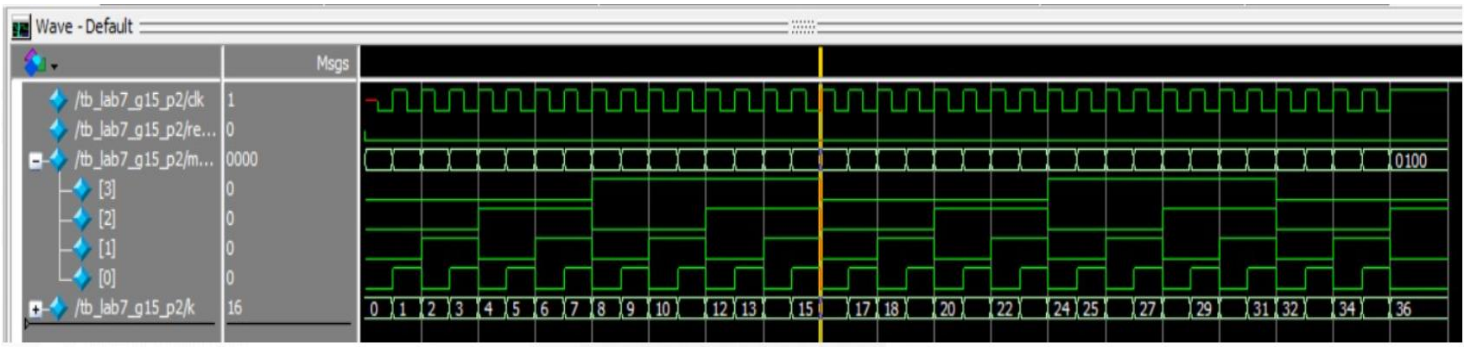
5

```
always_comb
begin
myout=count;
end



endmodule
```

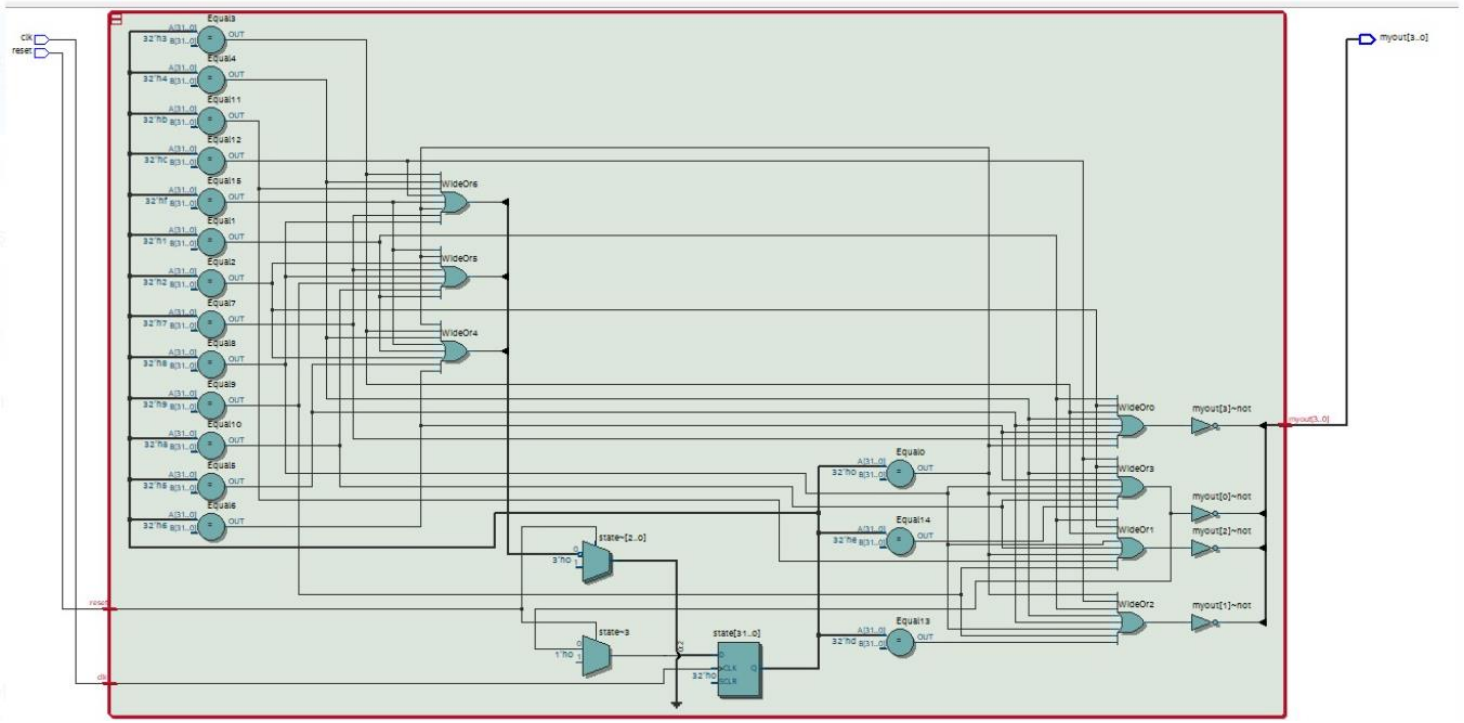
6

```
/* Hazirlayanlar : Senanur Agaç - Beyza Duran */
`timescale 1ns / 1ps

module tb_lab7_g15_p2();
logic clk;
logic reset;
logic [3:0]myout;

lab7_g15_p2 dene(clk,reset,myout);


integer k=0;

initial
begin

reset=0;

   for(k=0; k < 36; k=k+1)
   begin
      #5; clk=0;
      #5; clk=1;
   end


  #20;

$stop;

end
endmodule
```

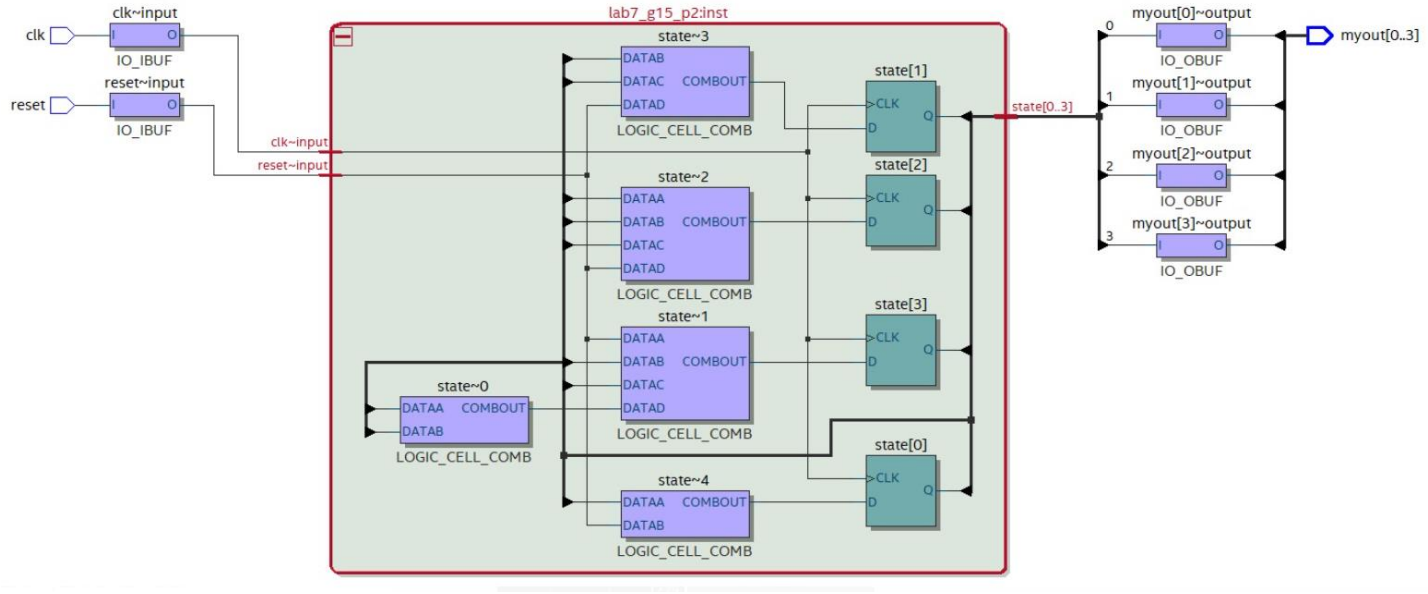
1



Şekil 11:2.soru için dalga şeması



Şekil 12: 2.Soru için RTL şeması



Şekil 13:2.Soru için post-mapping şeması

Flow Summary

<<Filter>>

Flow Status	Successful - Sun May 15 12:24:19 2022
Quartus Prime Version	19.1.0 Build 670 09/22/2019 SJ Lite Edition
Revision Name	labb7p2_top
Top-level Entity Name	labb7p2_top
Family	MAX 10
Device	10M08DAF484C8G
Timing Models	Final
Total logic elements	6 / 8,064 (< 1 %)
Total registers	4
Total pins	6 / 250 (2 %)
Total virtual pins	0
Total memory bits	0 / 387,072 (0 %)
Embedded Multiplier 9-bit elements	0 / 48 (0 %)
Total PLLs	0 / 2 (0 %)
UFM blocks	0 / 1 (0 %)
ADC blocks	0 / 1 (0 %)

Şekil 14: 2.soru için akış şeması

Analysis & Synthesis Resource Usage Summary

<<Filter>>

	Resource	Usage
2	-- 3 input functions	1
3	-- <=2 input functions	2
5		
6	Logic elements by mode	
1	-- normal mode	5
2	-- arithmetic mode	0
7		
8	Total registers	4
1	-- Dedicated logic registers	4
2	-- I/O registers	0
9		
10	I/O pins	6
11		
12	Embedded Multiplier 9-bit elements	0
13		
14	Maximum fan-out node	lab...[0]
15	Maximum fan-out	5
16	Total fan-out	33
17	Average fan-out	1.57

Şekil 15:2.soru için kaynak kullanımı özet şeması

Analysis & Synthesis Resource Utilization by Entity

<<Filter>>

	Compilation Hierarchy Node	Combinational ALUTs	Dedicated Logic Registers	Memory Bits	UFM Blocks	DSP Elements	DSP 9
1	|labb7p2_top	5 (0)	4 (0)	0	0	0	0
1	|lab7_g15_p2:inst|	5 (5)	4 (4)	0	0	0	0

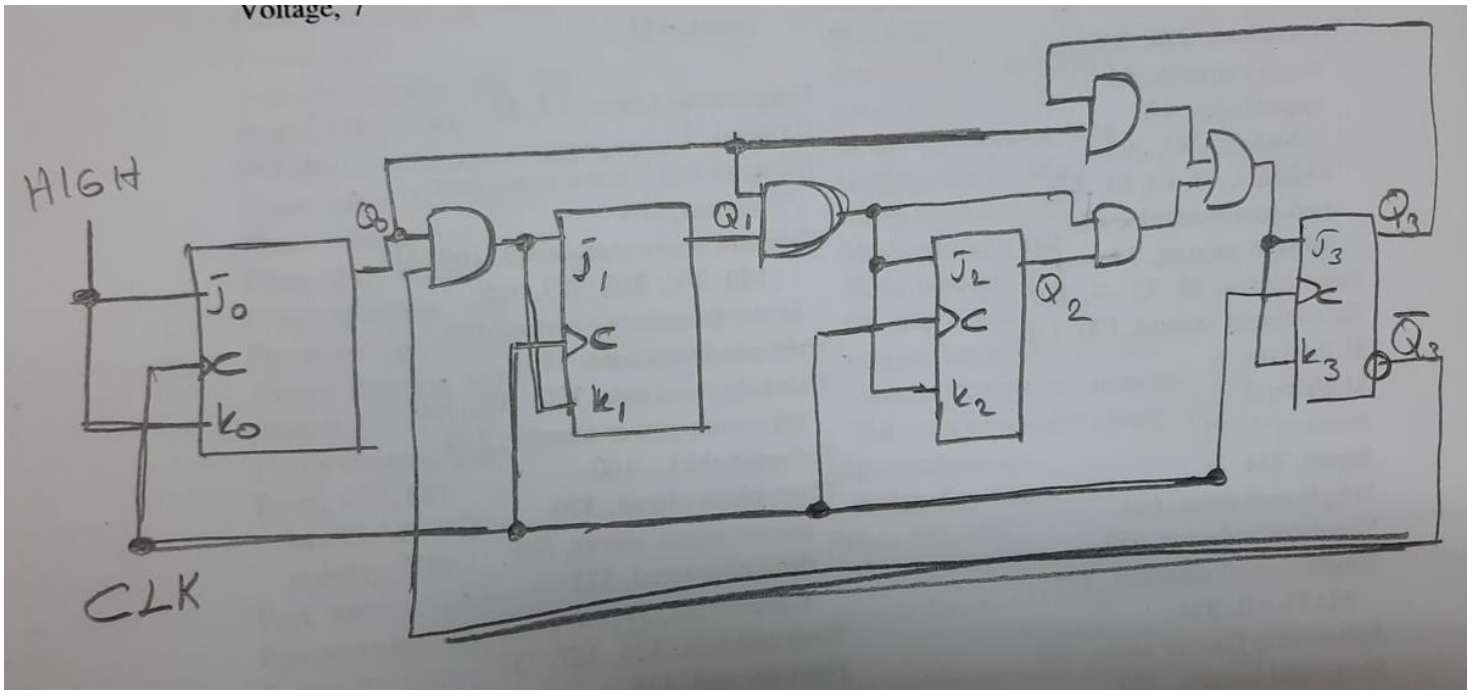
Şekil 16 :2.sou için utilization raporu

Slow 1200mV 0C Model Fmax Summary

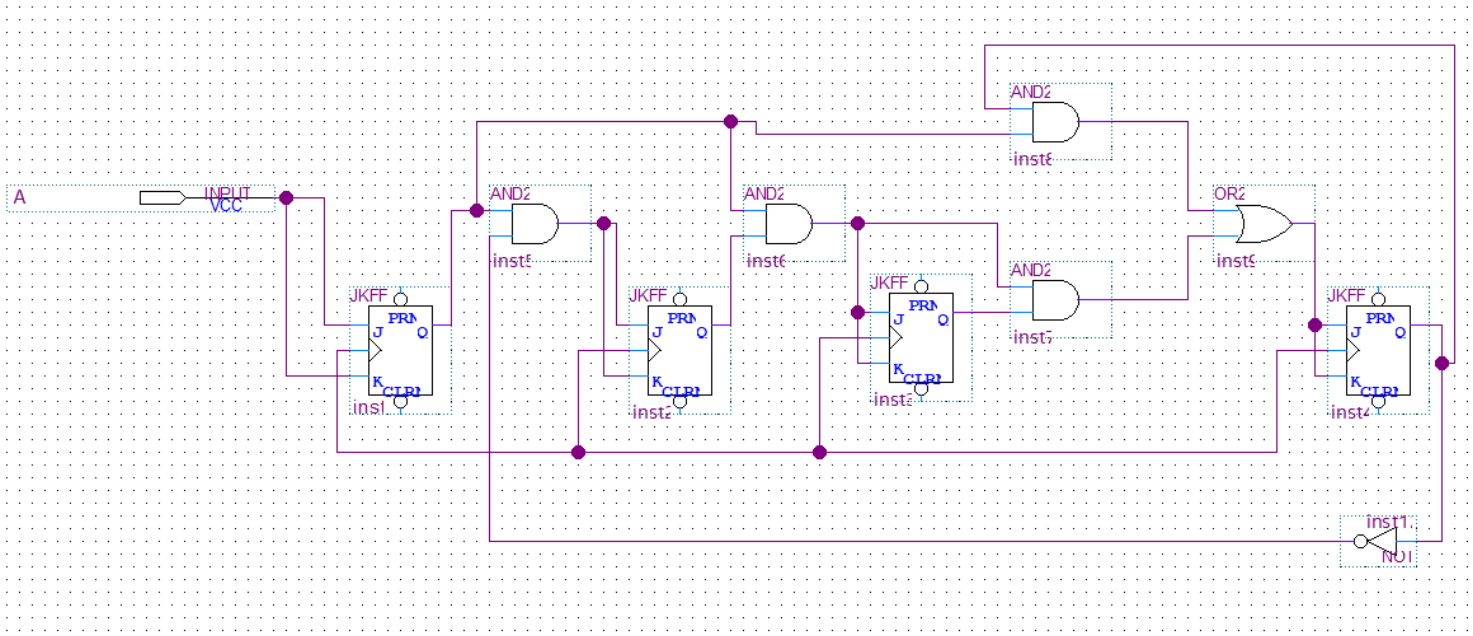
<<Filter>>

	Fmax	Restricted Fmax	Clock Name	Note
1	715.82 MHz	250.0 MHz	clk	limit due to minimum period restriction (max I/O toggle rate)

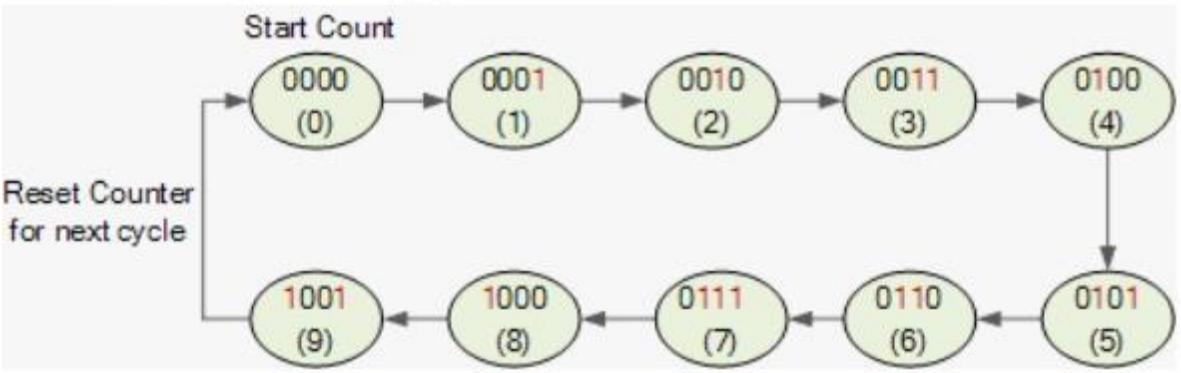
Şekil17:2.soru için FMAX değeri



Şekil 18: 2.Soru için devre şeması



Şekil 19:Devre şeması



Şekil 20:2.soru için state transition şeması