

# Sayı Sınıflandırıcı FSM

CSE433

Lecturer: Alp Arslan Bayrakçı

Assistant: Fatma Nur Esirci

---

<b>Özet</b>	<b>1</b>
<b>Tek-çift modülü</b>	<b>1</b>
<b>Palindrome Modülü(Binary Simetri)</b>	<b>2</b>
<b>Fibonacci Modülü</b>	<b>2</b>
C Kodu & FSM	2
Verilog Modülü	3
<b>Testler</b>	<b>4</b>
Tek-Çift Test Bench	4
Palindrome Test Bench	5
Fibonacci Test Bench	5
<b>Ekstra</b>	<b>7</b>

Muhammed Okumuş

151044017

## Özet

Sayı sınıflandırıcı makina 3 ayrı modül olarak planlandı.

- is\_odd
- is\_palindrome
- is\_fibonacci

Bu modüller ve test benchleri verilog da yazıldı. Tek-çift ve palindrome modülleri combinational olduğu için FSM'e gerek kalmadan implement edildi. Fibonacci modülü için FSM tasarlandı ve implement edildi.

Tüm modülleri birleştirici modül yazılmadı. Bu yüzden ortak testler fibonacci testi ile beraber yapıldı. Diğer modüller için bağımsız testler yapıldı.

## Tek-çift modülü

En basit modül, aldığı input'u döndürüyor.

```
1 module is_odd(  
2     input a,  
3     output o  
4 );  
5  
6     assign o = a;  
7  
8 endmodule
```

## Palindrome Modülü(Binary Simetri)

32-bit binary sayının tüm simetrik bitleri XNOR'lanıyor ve bu XNOR sonuçları AND'leniyor. Bu modül için başta loop kullanmayı düşündüm fakat verilog loopları pek sevmiyor, bu sebepten dolayı normal kombinasyonel devre olarak yazdım.

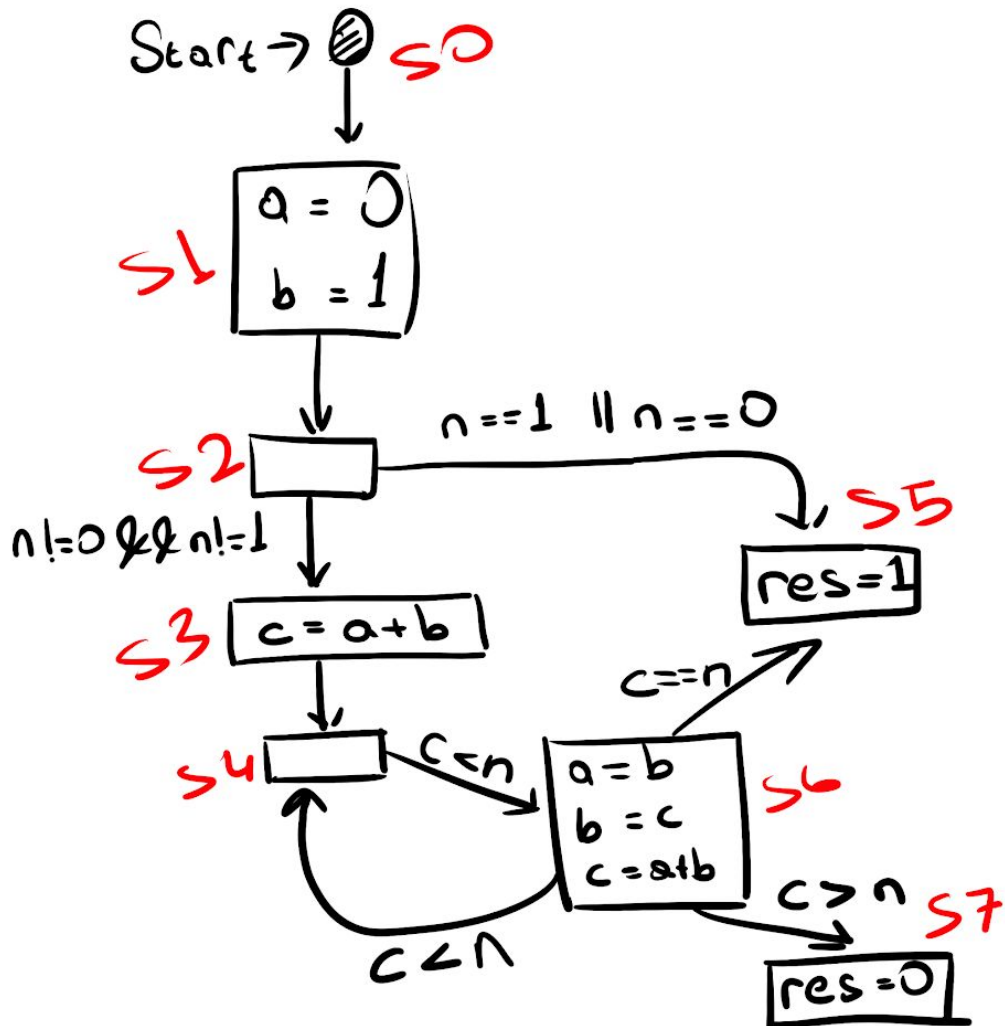
## Fibonacci Modülü

### C Kodu & FSM

```

1  int is_fibo(int n)
2  {
3      int a = 0;
4      int b = 1;
5      if (n==a || n==b) return true;
6      int c = a+b;
7      while(c<n)
8      {
9          a = b;
10         b = c;
11         c = a + b;
12         if(c == n) return true;
13     }
14     return false;
15 }

```



## Verilog Modülü

FSM 8 State'den oluştuğu için 3 bitle ifade edildi.

```

13  parameter
14      S0=3'b000,
15      S1=3'b001,
16      S2=3'b010,
17      S3=3'b011,
18      S4=3'b100,
19      S5=3'b101,
20      S6=3'b110,
21      S7=3'b111;

```

Testleri kolaylaştırmak için **reset** ve **done** sinyalleri eklendi.

Stateler arası geçiş mantığı:

```

23  reg [2:0] current_state, next_state;
24
25  // sequential memory
26  always @(posedge CLK, posedge RESET)
27  begin
28      if(RESET==1)
29          current_state <= S0;
30      else
31          current_state <= next_state;
32  end
33

```

Kombinasyonel kısım(bir kısmı):

```

34  // combinational logic
35  always @(current_state, N)
36  begin
37      case(current_state)
38      S0:begin
39          //display ("S0");
40          next_state <= S1;
41      end
42      S1:begin
43          //display ("S1");
44          a = 32'b00000000000000000000000000000000;
45          b = 32'b00000000000000000000000000000001;
46          next_state <= S2;
47      end
48      S2:begin
49          //display ("S2");
50          if ( N == 32'b00000000000000000000000000000000 || N == 32'b00000000000000000000000000000001 )
51              next_state <= S5;
52          else
53              next_state <= S3;
54      end
55  end
56

```

Output mantığı:

```

96  // combinational logic to determine the output
97  always @(current_state)
98  begin
99      case(current_state)
100         S0: O <= 0;
101         S1: O <= 0;
102         S2: O <= 0;
103         S3: O <= 0;
104         S4: O <= 0;
105         S5:begin
106             O <= 1;
107             DONE <= 1;
108         end
109         S6: O <= 0;
110         S7:begin
111             O <= 0;
112             DONE <= 1;
113         end
114     endcase
115 end

```

## Testler

Testler Run-All seçeneği ile çalıştırıldı. Fibonacci kısmında tek tek çalıştırma uzun sürebilir. O yüzden bu seçenek yararlı oldu.



## Tek-Çift Test Bench

```

VSIM 18> run -all
# A: 11111111111111111111111111111111 is_odd: 1
# A: 11111111111111111111111111111110 is_odd: 0
# A: 00000000000000000000000000000000 is_odd: 0
# A: 100000000000000000000000000000001 is_odd: 1
# A: 11010010011000011000011001001011 is_odd: 1
# A: 1111111111111111100000000000000000 is_odd: 0

```

## Palindrome Test Bench

```
VSIM 20> run -all
# A: 11111111111111111111111111111111 is_palindrome: 1
# A: 11111111111111111111111111111110 is_palindrome: 0
# A: 00000000000000000000000000000000 is_palindrome: 1
# A: 10000000000000000000000000000001 is_palindrome: 1
# A: 11010010011000011000011001001011 is_palindrome: 1
# A: 11111111111111000000000000000000 is_palindrome: 0
```

## Fibonacci Test Bench

Burda birden fazla testi arka arkaya çalıştırmayı başaramadım bu yüzden her testte sayıyı tekrar giriyorum. Is\_fibonacci\_tb içerisindeki A registerı.

```
21 initial begin
22     // Initialize Inputs
23     A = 32'b10110001000110010010010011100001; // 2971215073 Largest fibonacci number in 32bit
24     reset = 1;
25     // Wait for global reset to finish
26     #30;
27     reset = 0;
28 end
```

Fibonacci 47 : 2971215073

```
VSIM 22> run -all
# A: 10110001000110010010010011100001
#     is_palindrome: 0
#     is_odd: 1
#     is_fibonacci: 1
```

Fibonacci 0 : 0

```
VSIM 26> run -all
# A: 00000000000000000000000000000000
#     is_palindrome: 1
#     is_odd: 0
#     is_fibonacci: 1
```

Fibonacci 1 : 1

```
VSIM 30> run -all
# A: 00000000000000000000000000000001
#     is_palindrome: 0
#     is_odd: 1
#     is_fibonacci: 1
```

Fibonacci 28 : 317811

```
VSIM 34> run -all
# A: 00000000000001001101100101110011
#      is_palindrome: 0
#      is_odd: 1
#      is_fibonacci: 1
```

Not Fibonacci : 999610691

```
VSIM 38> run -all
# A: 00111011100101001101100101000011
#      is_palindrome: 0
#      is_odd: 1
#      is_fibonacci: 0
```

Fibonacci sayılarının alındığı kaynak:

<http://www.maths.surrey.ac.uk/hosted-sites/R.Knott/Fibonacci/fibtable.html>



## Ekstra

Fibonacci modülü derlenirken

Warning (13012): Latch a[31] has unsafe behavior

Warning (13013): Ports D and ENA on the latch are fed by the same signal current\_state.S6

Hatası alınıyor. Bu hatanın “fed by the same signal” kısmını anlamadım, testler sorunsuz çalışıyor ama belki FSM’in done statelerinde bir mantık hatası olabilir.

Test benchlerde de fibonacci modülü done flag’i verince \$stop kullanılarak ModelSim debug break moduna geçiriliyor, başka durdurma yolunu bulamadım.

```

34 always@(*) begin
35     if (done == 1) begin
36         $display("A: %32b\n\tis_palindrome: %1b\n\tis_odd: %1b\n\tis_fibonacci: %1b", A, o_pal, o_odd, o_fibo);
37     $stop;
38     end

```

0 errors, 204 warnings

