Flowchart

A
```
odd  ← 0    K ←
even ← 0    M ←
```

S
0   1

B
K ← K - M

0   K >= M   1

F
[   ]

0   K >= M   1

0   K = 0   1

C
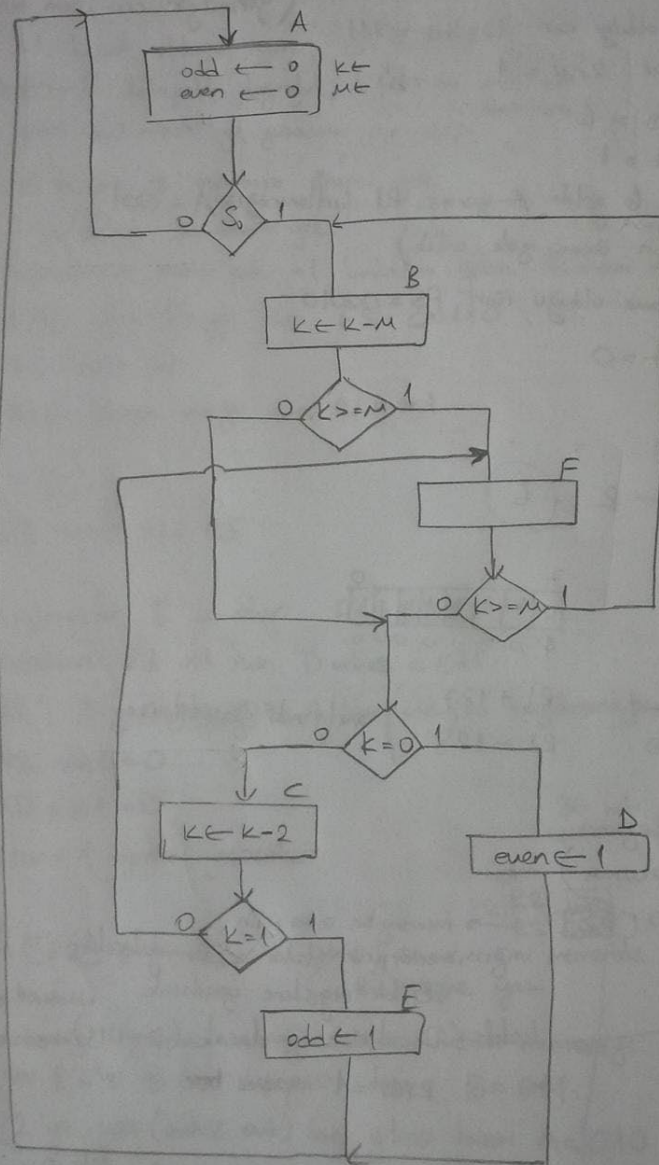K ← K - 2

D
even ← 1

0   K = 1   1

E
odd ← 1

modulo.vhd

```vhdl
library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_arith.all;
use ieee.std_logic_unsigned.all;

entity modulo is
port(
                K,M:in integer range 0 to 99;
                s0:in std_logic;
                odd, even:out std_logic;
                clk,reset:in std_logic

);
end modulo;

architecture beh of modulo is

type typ_state is (A,B,C,D,E,F);
signal state,next_state:typ_state;

signal sayi1:integer range 0 to 99;
signal sayi2:integer range 0 to 99;

begin

process(clk,reset)
begin
        if(reset='1')then
                state<=A;
        elsif(rising_edge(clk))then
                state<=next_state;
        end if;
end process;

process(s0,sayi1,sayi2,state)
begin
        case state is
                when A=>

                if s0='1' then
                        next_state<=B;
                else
                        next_state<=A;
                end if;

                when B=>

                if (sayi1 >= sayi2) then
                        next_state<=F;
```

```vhdl
                elsif(sayi2 > sayi1) then
                        if(sayi1=0) then
                                next_state<=D;
                        else
                                next_state<=C;
                        end if;
        end if;

        when F=>
                if (sayi1 >= sayi2) then
                        next_state<=B;
                else
                        if(sayi1=0) then
                                next_state<=D;
                        else
                                next_state<=C;
                        end if;
                end if;

        when C=>

        if (sayi1=1) then
                next_state<=E;
        else
                next_state<=F;
        end if;

        when D=>
                next_state<=A;
        when E=>
                next_state<=A;

        end case;
end process;




process(state, K, M)
begin
        case state is
                when A=>

                        odd<='0';
                        even<='0';
                        sayi1<=K;
                        sayi2<=M;

                when B=>

                        --K<=K-M;
                        sayi1<=sayi1-sayi2;
```
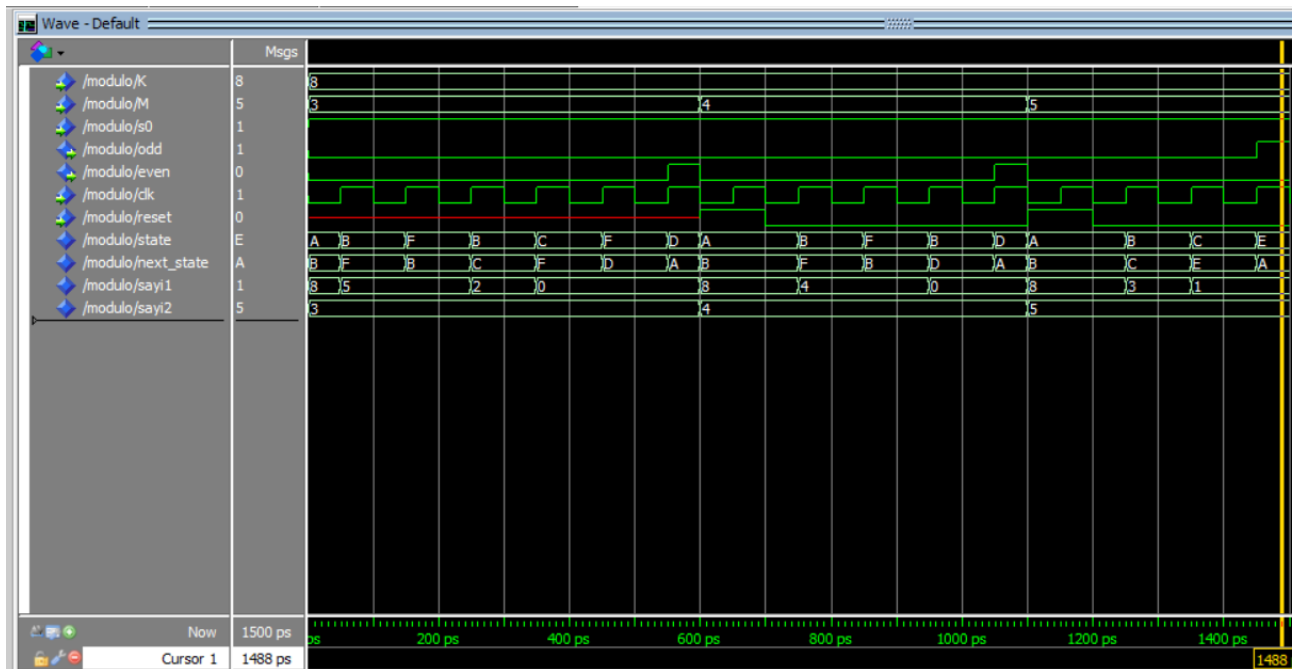
```
                when C=>

                        --K<=K-2;
                        sayi1<=sayi1-2;

                when D=>

                        even<='1';

                when E=>

                        odd<='1';

                when F=>

        end case;
end process;


end beh;
```
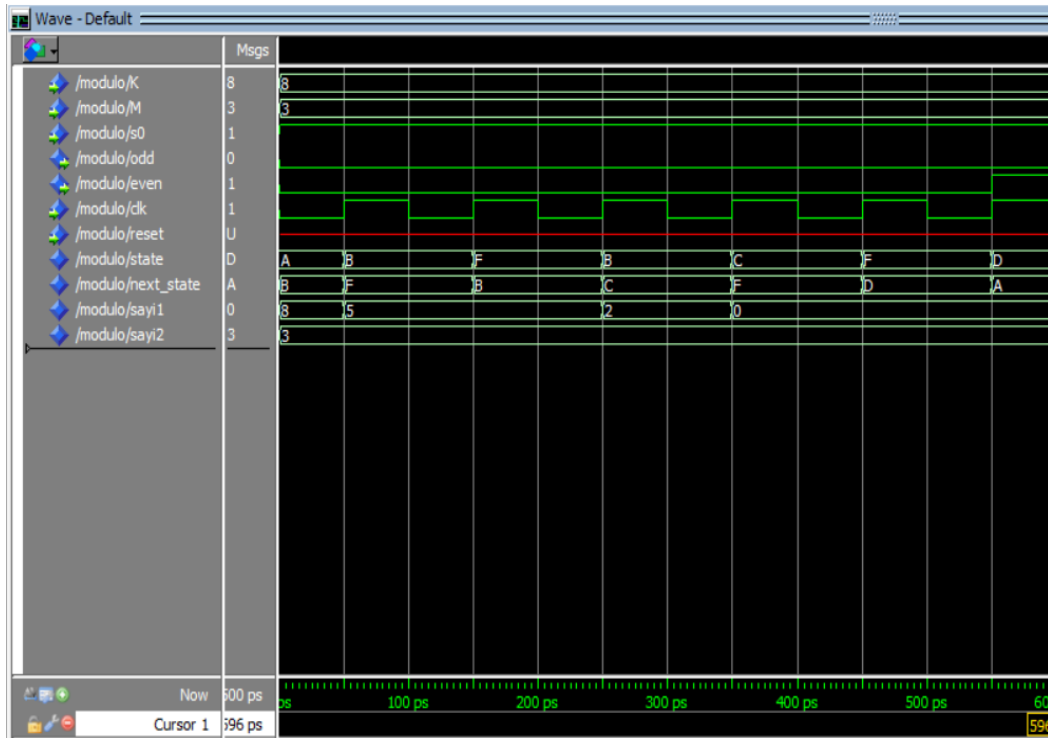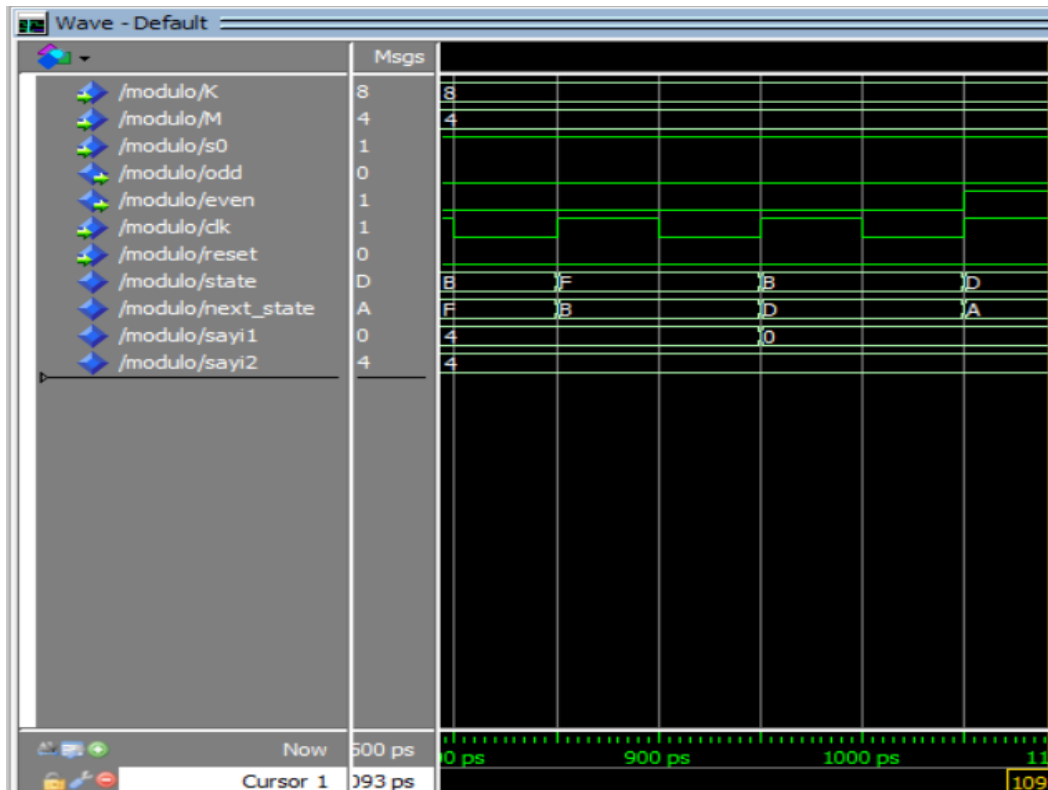
Modelsim simülasyon çıktıları

(3 denemenin tek gösterimi)

1.deneme

2.deneme

3.deneme