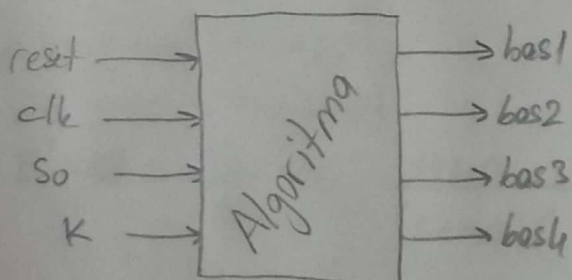
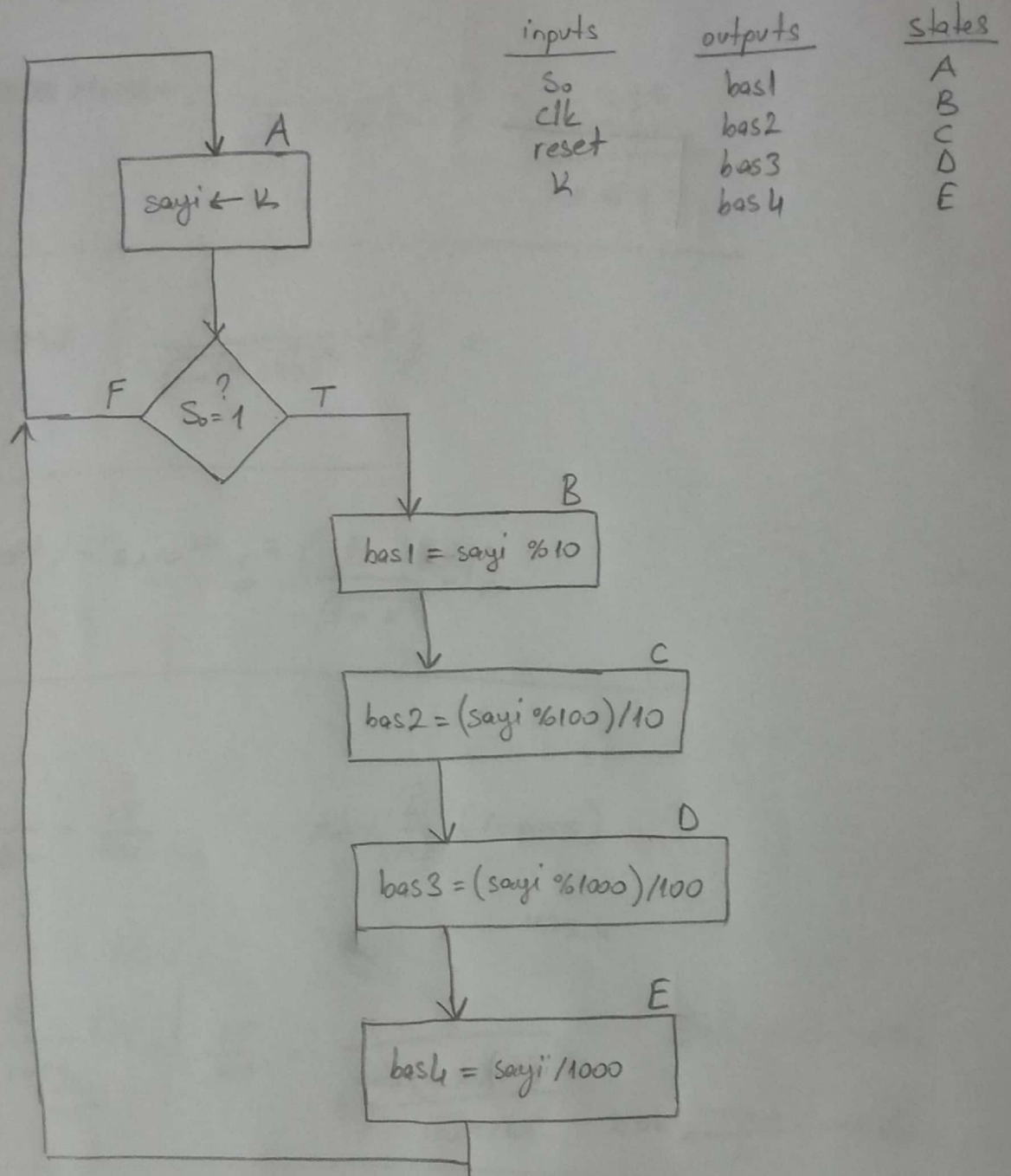


# State machine



# state.vhd

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

entity state is
port(
    start:in std_logic;
    s0:in std_logic;
    bas1,bas2,bas3,bas4:out std_logic_vector(3 downto 0);
    clk,reset:in std_logic;
    K:in std_logic_vector(13 downto 0)
);
end state;

architecture behavior of state is

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

signal sayi:std_logic_vector(13 downto 0):="000000000000000";

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,state)
begin
    case state is
        when A=>

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

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

        end case;
end case;
```

```

end process;

process(state,K)
begin
    case state is
        when A=>
            sayi<=K;
        when B=>
            bas1<=conv_std_logic_vector((conv_integer(sayi) mod 10),4);
        when C=>
            bas2<=conv_std_logic_vector(((conv_integer(sayi) mod 100)/10),4);
        when D=>
            bas3<=conv_std_logic_vector(((conv_integer(sayi) mod 1000)/100),4);
        when E=>
            bas4<=conv_std_logic_vector((conv_integer(sayi)/1000),4);
    end case;
end process;

end behavior;

```

## *VHDL kodunu yazarken yaptığımız şeyler ve kritik noktalar*

1-Gerekli kütüphaneleri ve oluşturacağımız state yapısını tanımladık

2-Oluşturduğumuz state yapısının input ve outputlarını belirledik 1 basamak en fazla 9 olacağı için bas1, bas2, bas, bas4 outputların 4 bit olması yeterlidir, giriş için de 4 basamaklı bir sayı için 14 bit bizim için yeterli olacaktır

3-A,B,C,D,E stateleri için yeni bir veri tipi tanımladık ve state, next\_state yapılarının bu veri tipi biçiminde tanımladık

4-3 tane process yapısı tanımladık bunların sensivityleri ve içerikleri şu şekilde:

process1 (clk, reset)

burda reset varsa ilk state geri dön eğer resek yoksa clockun yükselen kenarında bir sonraki state geç

process2 (s0,state)

burda s0 sıfır anahtarı 0 iken tekrar A statine geç , 1 olduğunda bir sonraki state geç sonrasında zaten koşul olmadan stateleri tek tek gezecek

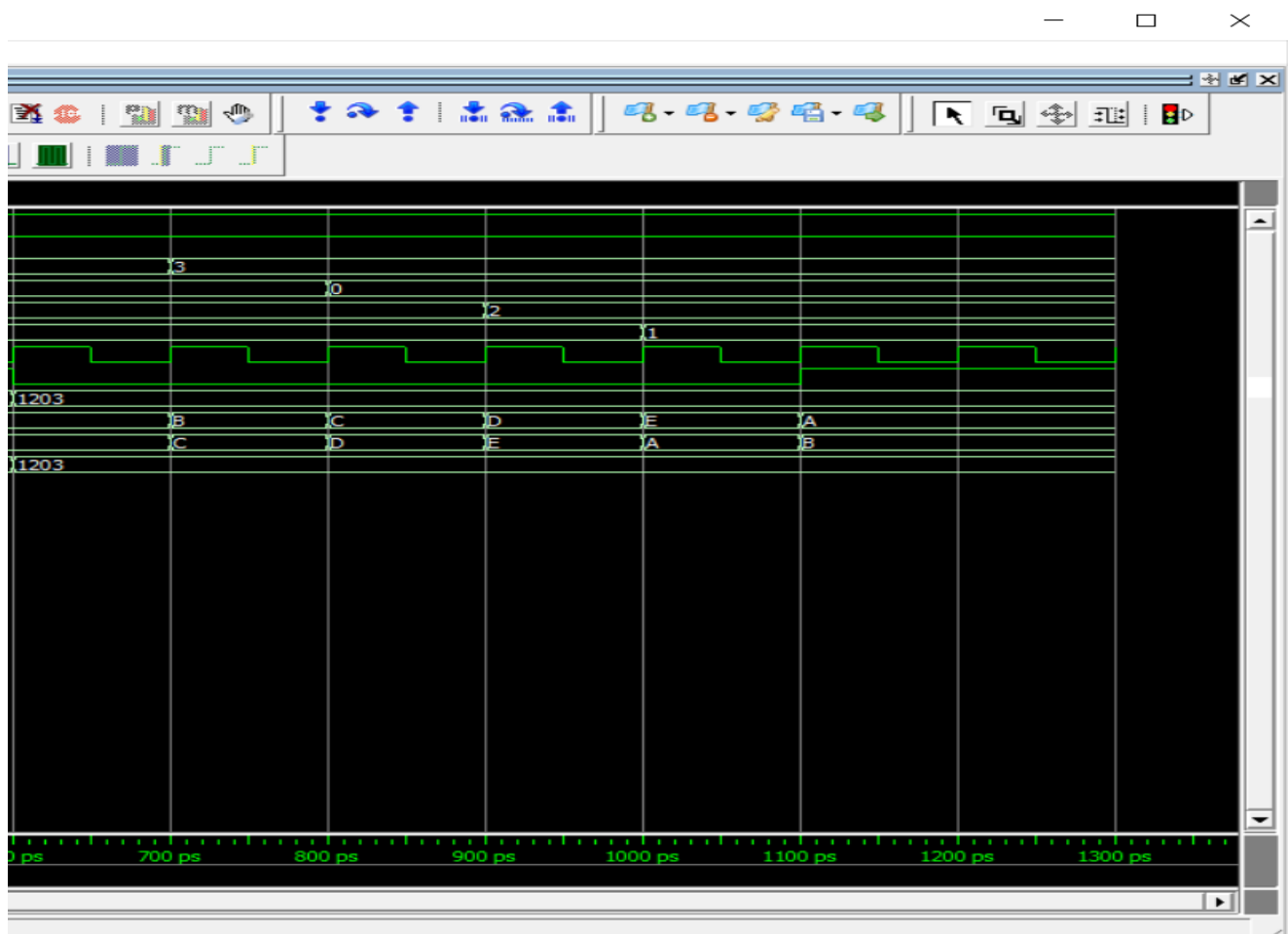
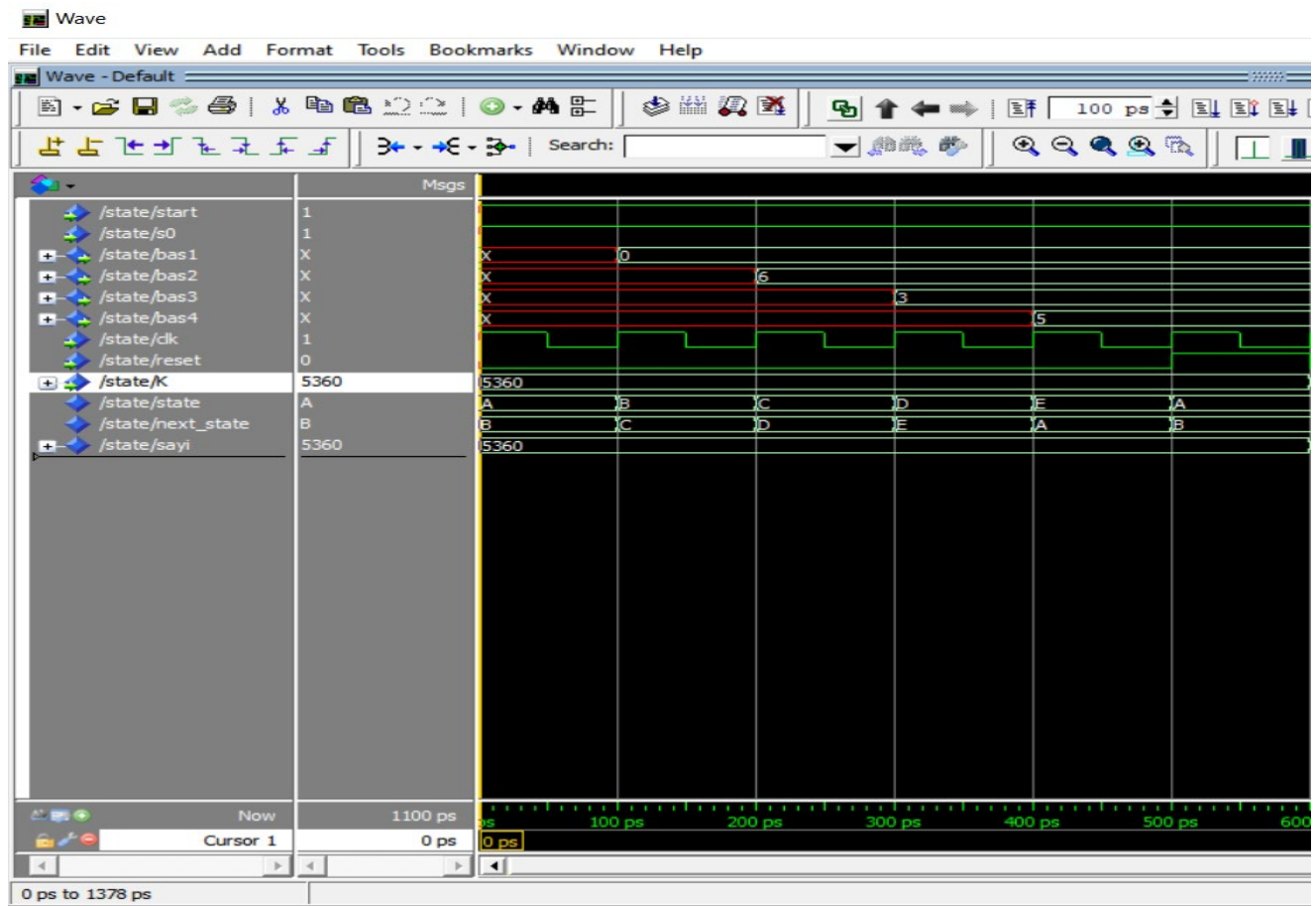
process3 (state,K)

burda statelerin içinde gerçekleşen işlemleri tanımladık

A statetinde arasyal olan sayımızı dışardan K inputuyla girdiğimiz sayıya eşitliyoruz

B,C,D,E statelerindeki önemli nokta ise binary olan bir sayıyı ilk önce integer forma dönüştürüyoruz gerekli mod alma bölme işlemlerinden sonra da bas1, bas2, bas3, bas4 outputlarımızı integer formdan tekrar binary forma çeviriyoruz

## Modelsim simülasyon çıktıları



---

## *Modesim simülasyonunda yaptığımız şeyler*

### *1.deneme*

s0=1

clk=clock

reset=0

K=5360 (01010011110000) bunu force yaparken 14 bit olarak tanımladığımız için bit bit forceluyoruz ama ben simülasyonda gösterirken decimal formda gösterdim

bas1=0

bas2=6

bas3=3

bas4=5

### *2.deneme*

s0=1

clk=clock

reset=0

K=1203 (00010010110011)

bas1=3

bas2=0

bas3=2

bas4=1

---