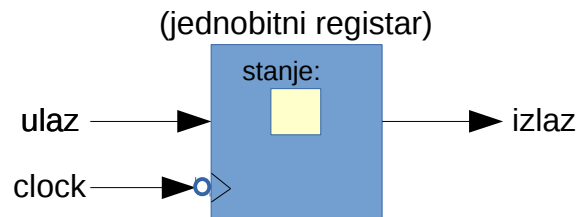


Registar: sklop koji na signal takta (primjerice padajući brid) zapisuje i pamti podatak doveden na ulaz i stavlja ga na izlaz. Primjer: registar koji pamti 1 bit:

```
entity registar is port (  
    ulaz: std_logic;  
    izlaz: std_logic);  
end registar;  
  
architecture pon of registar is  
begin  
  
    process(clock)  
        variable stanje: std_logic;  
    begin  
        if falling_edge(clock) then  
            stanje := ulaz;  
        end if;  
        izlaz <= stanje;  
    end process;  
  
end pon;
```



Efektivno, ovo je jedan bistabil D!

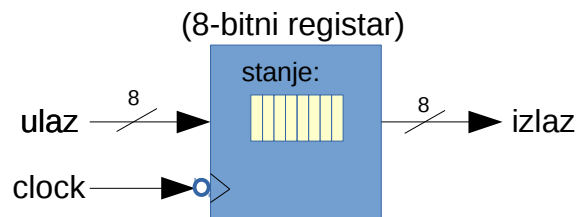
Općenito, registar može pamtit i više bitova podataka. Primjer 8-bitnog registra prikazan je u nastavku.

```
entity registar8 is port (  
    ulaz: std_logic_vector(7 downto 0);  
    izlaz: std_logic_vector(7 downto 0));  
end registar8;
```

```
architecture pon of registar8 is  
begin
```

```
    process(clock)  
        variable stanje: std_logic_vector(7 downto 0);  
    begin  
        if falling_edge(clock) then  
            stanje := ulaz;  
        end if;  
        izlaz <= stanje;  
    end process;
```

```
end pon;
```



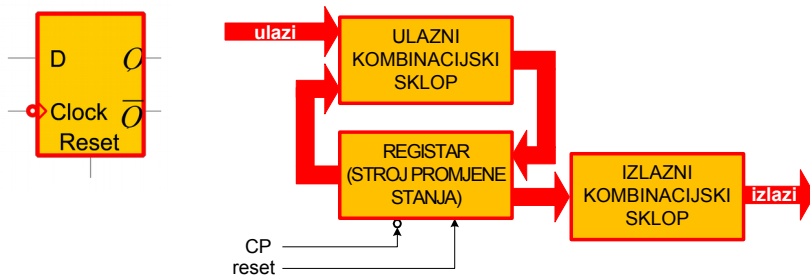
VHDL opis bistabila D s dodatnim sinkronim ulazom za resetiranje već ima jasno vidljivu strukturu Mooreovog automata – ako znamo dobro gledati. Evo VHDL opisa:

```
entity Dbistabil is port (
  D, Reset, Clock: in std_logic;
  Q, Qn: out std_logic
);
end Dbistabil;
```

```
architecture pon of Dbistabil is
begin
```

```
  process(Clock)
    variable stanje: std_logic;
  begin
    if falling_edge(Clock) then
      if Reset='1' then
        stanje := '0';
      else
        stanje := D;
      end if;
    end if;
    Q <= stanje;
    Qn <= not stanje;
  end process;
```

```
end pon;
```



Model registra: na padajući brid upiši novo stanje (zapamti podatak)

Kombinacijski sklop koji na temelju ulaza Reset, D (i trenutnog stanja) odlučuje što će se zapisati kao sljedeće stanje

Izlazni kombinacijski sklop koji na temelju stanja generira izlaze sklopa

Malo restrukturiran kod kako bi struktura Mooreovog automata postala jasnije vidljiva:

```
entity Dbistabil is port (
  D, Reset, Clock: in std_logic;
  Q, Qn: out std_logic
);
end Dbistabil;
```

```
architecture pon of Dbistabil is
begin
```

```
  process(Clock)
    variable stanje, prijedlog: std_logic;
```

```
  begin
```

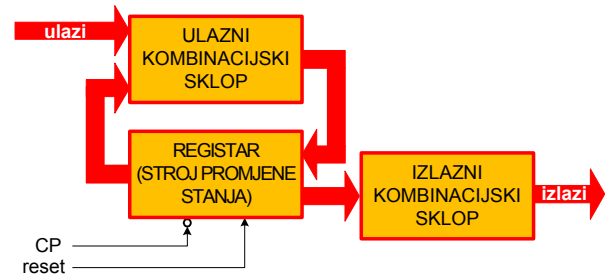
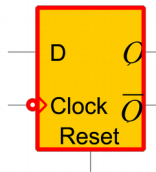
```
    if Reset='1' then
      prijedlog := '0';
    else
      prijedlog := D;
    end if;
```

```
    if falling_edge(Clock) then
      stanje := prijedlog;
    end if;
```

```
    Q <= stanje;
    Qn <= not stanje;
```

```
  end process;
```

```
end pon;
```



Kombinacijski sklop koji na temelju ulaza Reset, D (i trenutnog stanja) odlučuje što će se zapisati kao sljedeće stanje

Model registra: na padajući brid upiši novo stanje (zapamti podatak)

Izlazni kombinacijski sklop koji na temelju stanja generira izlaze sklopa

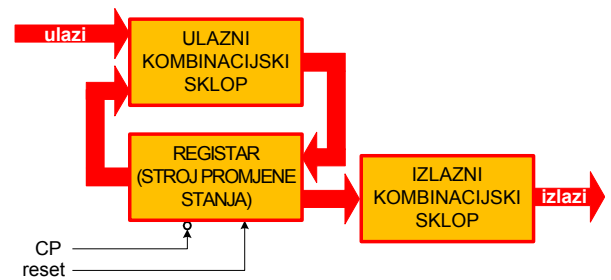
VHDL opis bistabila T: sada je jasno vidljivo da se prijedlog sljedećeg stanja računa na temelju ulaza sklopa i trenutnog stanja:

```
entity Tbistabil is port (  
    T, Reset, Clock: in std_logic;  
    Q, Qn: out std_logic  
);  
end Tbistabil;
```

```
architecture pon of Tbistabil is  
begin
```

```
    process(Clock)  
        variable stanje: std_logic;  
    begin  
        if falling_edge(Clock) then  
            if Reset='1' then  
                stanje := '0';  
            else  
                stanje := T xor stanje;  
            end if;  
        end if;  
        Q <= stanje;  
        Qn <= not stanje;  
    end process;
```

```
end pon;
```



Model registra: na padajući brid upiši novo stanje (zapamti podatak)

Kombinacijski sklop koji na temelju ulaza Reset, T i trenutnog stanja odlučuje što će se zapisati kao sljedeće stanje

Izlazni kombinacijski sklop koji na temelju stanja generira izlaze sklopa