



Univerza v Ljubljani

Fakulteta
za računalništvo
in informatiko



Metode logičnega snovanja

Mealyjev avtomat

Miha Moškon



Mealyjev avtomat

Izhodna črka je določena s stanjem in vhodno črko.

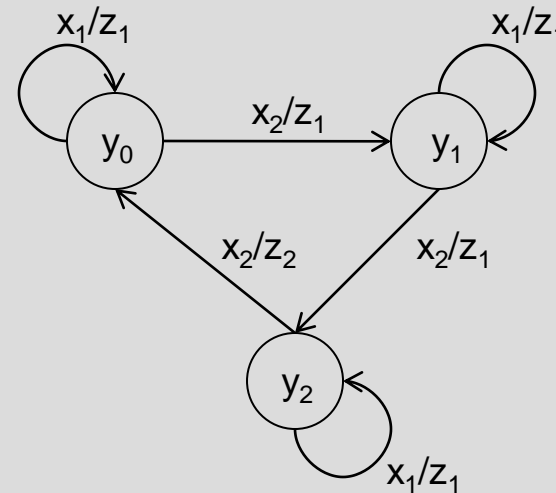
Vedno obstaja preslikava iz Moorovega avtomata v Mealyjev in obratno.

Opišemo ga lahko z logičnimi enačbami ali STL načinom.

Opisovanje z logičnimi enačbami

Podan je Mealyjev avtomat:

$X \setminus Y$	y_0	y_1	y_2
x_1	y_0/z_1	y_1/z_1	y_2/z_1
x_2	y_1/z_1	y_1/z_1	y_0/z_2



Zapišemo tabelo prehajanja stanj:

Q1 Q0		$X = 0 (x_1)$			$X = 1 (x_2)$		
		D^1Q	D^1Q	Z	D^1Q	D^1Q	Z
y_0	0 0	0	0	0 (z_1)	0	1	0 (z_1)
y_1	0 1	0	1	0 (z_1)	1	0	0 (z_1)
y_2	1 0	1	0	0 (z_1)	0	0	1 (z_2)

Opisovanje z logičnimi enačbami (2)

Iz tabele preberemo enačbe avtomata, ki jih zapišemo v proces:

```
Q1 <= (not rst and (not X and Q1 and not Q0)) or
      (not rst and ( X and not Q1 and Q0));
```

```
Q0 <= (not rst and not (X and not Q1 and Q0)) or
      (not rst and (X and not Q1 and not Q0));
```

```
z <= not rst and X and (Q1 and not Q0);
```

STL opisovanje

Poleg pomožne spremenljivke za novo stanje, potrebujemo tudi pomožno spremenljivko za novo izhodno črko.

Primer (implementaciji avtomata iz prosojnice 3):

```
-- knjiznica, ki jo uporabljamo
library IEEE;
use IEEE.STD_LOGIC_1164.all;

-- entiteta
entity mealy is
    port (clk, rst, X: in std_logic;
          Z: out std_logic);
end mealy;
```

STL opisovanje (2)

```

architecture behavioral of mealy is
    type states is (Y0, Y1, Y2);
    signal currSt, nextSt: states; -- pomocni signali
    signal nextZ: std_logic; -- izhodna crka
begin
    process (clk)
    begin
        if rising_edge(clk) then
            if (rst = '1') then
                currSt <= Y0;
                Z <= '0';

            else
                currSt <= nextSt;
                Z <= nextZ;
            end if;
        end if;
    end process;

```

STL opisovanje (3)

```

transition_funct:process (currSt, x)
begin
    case currSt is
        when Y0 =>
            if (X = '0') then nextSt <= Y0;
                else nextSt <= Y1;
            end if;
        when Y1 =>
            if (X = '0') then nextSt <= Y1;
                else nextSt <= Y2;
            end if;
        when Y2 =>
            if (X = '0') then nextSt <= Y2;
                else nextSt <= Y0;
            end if;
    end case;
end process;

```

STL opisovanje (4)

```

output_funct:process (currSt, x)
begin
    case currSt is
        when Y0 =>
            if (X = '0') then nextZ <= '0';
                else nextZ <= '0';
            end if;
        when Y1 =>
            if (X = '0') then nextZ <= '0';
                else nextZ <= '0';
            end if;
        when Y2 =>
            if (X = '0') then nextZ <= '0';
                else nextZ <= '1';
            end if;
    end case;
end process;
end architecture;

```


Naloga

S STL opisom realizirajte Mealyjev avtomat, ki bo deloval na sledeč način:

- BTN West: **reset**
- BTN North: izbira naslednjega znaka za izpis
- BTN South: izbira prejšnjega znaka za izpis
- BTN East: izpis izbranega znaka

Izbira znaka za izpis

- Implementirajte števec, ki bo štel od "`(100)0110000`" (0) do "`(100)1011010`" (Z).
- Ob pritisnjenem gumbu `BTN North` se vsebina števca poveča.
- Ob pritisnjenem gumbu `BTN South` se vsebina števca zmanjša.
- Števec se resetira na vrednost "`(100)0110000`" (0).

Detekcija pritisnjene gumba

Obstoječi modul `btn_pressed` ob enkratnem pritsku gumba občasno zazna več prehodov.

Modul `btn_pressed` dopolnite tako, da bo zaznal pritisk šele, ko bo gumb pritisnjen vsaj 100 ms.