Univerza v Ljubljani
Fakulteta
za računalništvo
in informatiko

Metode logičnega snovanja

Avtomati

Miha Moškon

Končni avtomat

Definicija:

$A = \{X, Y, Z, \delta, \lambda\}$,

X ... neprazna, končna množica vhodnih črk,

Y ... neprazna končna množica notranjih črk (stanj),

Z ... neprazna končna množica izhodnih črk,

δ ... funkcija naslednjega notranjega stanja (funkcija prehajanja stanj),

λ ... funkcija naslednje izhodne črke (izhodna funkcija).

Končni avtomat

Osnovna tipa avtomatov:

- Moorov avtomat,
- Mealyjev avtomat.

Sinhroni avtomati:

- prehode med stanji proži urin signal,
- najpogostejša uporaba.

Asinhroni avtomati:

- uporabljajo se zelo redko.

Moorov avtomat

Izhodna črka je določena s stanjem avtomata:

$$\lambda: Y \rightarrow Z$$

Mealyjev avtomat

Izhodna črka je določena s prehodom med stanji (s stanjem in z vhodno črko):

$$\lambda: X \times Y \rightarrow Z$$

Opisovanje

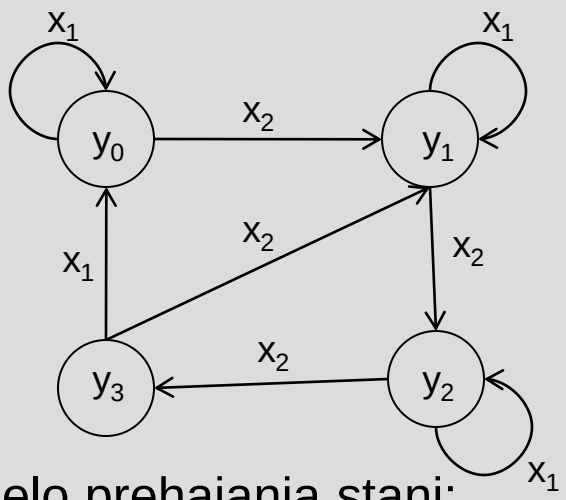
Avtomate lahko opišemo z:

- logičnimi enačbami:
  - sestavimo tabelo prehajanja stanj in izhodov,
  - Izpišemo logične enačbe.
- STL (State Transition Language) načinom:
  - trije procesi,
  - prvi proces skrbi za Set/Reset in prehode ob urinem signalu,
  - drugi proces opisuje funkcijo prehajanja stanj (δ),
  - tretji proces opisuje izhodno funkcijo (λ).

Opisovanje z logičnimi enačbami

Podan je Moorov avtomat:

Z	z_0	z_0	z_0	z_1
X \ Y	y_0	y_1	y_2	y_3
x_1	y_0	y_1	y_2	y_0
x_2	y_1	y_2	y_3	y_1



Zapišemo kodirne tabele in tabelo prehajanja stanj:

			X = 0 (x_1)		X = 1 (x_2)		
Y	Q1	Q0	D^1Q1	D^1Q0	D^1Q1	D^1Q0	Z
y_0	0	0	0	0	0	1	0 (z_0)
y_1	0	1	0	1	1	0	0 (z_0)
y_2	1	0	1	0	1	1	0 (z_0)
y_3	1	1	0	0	0	1	1 (z_1)

Opisovanje z logičnimi enačbami

Iz tabele preberemo enačbe avtomata, ki jih zapišemo v proces, ki ga proži urin signal:

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

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

Z <=    Q1 and Q0 -- zunaj procesa
```

STL opisovanje

Mooreov avtomat opišemo s tremi procesi.

Prvi proces proži ura in skrbi za Set/Reset in prehode med stanji.

Drugi proces opisuje prehode med stanji avtomata (funkcijo prehajanja stanj). Proži ga sprememba trenutnega stanja ali sprememba vhodne črke.

Tretji proces podaja izhode avtomata (izhodno funkcijo). Proži ga sprememba trenutnega stanja.

STL opisovanje

Primer (implementacija avtomata iz prosojnice 7):

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

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

architecture behavioral of moore is
   type states is (Y0, Y1, Y2, Y3);
   signal currSt, nextSt: states; -- pomozni signali
begin
```

STL opisovanje

```
process (clk)
begin
    if (rising_edge(clk)) then
        if (rst = '1') then
            currSt <= Y0;
        else
            currSt <= nextSt;
        end if;
    end if;
end process;
```

STL opisovanje

```
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;
```

STL opisovanje

```
            when Y2 =>
                    if (X = '0') then nextSt <= Y2;
                                 else nextSt <= Y3;
                    end if;
            when Y3 =>
                    if (X = '0') then nextSt <= Y0;
                                 else nextSt <= Y1;
                    end if;
            end case;
end process;
```

STL opisovanje (6)

```
output_funct:process (currSt)
begin
    case currSt is
        when Y0 =>
            Z <= '0';
        when Y1 =>
            Z <= '0';
        when Y2 =>
            Z <= '0';
        when Y3 =>
            Z <= '1';
        end case;
end process;

end architecture;
```

Naloga

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

- pritisk `BTN_W`: reset,
- pritisk `BTN_N`: clear screen,
- pritisk `BTN_S`: izpiši `A` na LCD,
- pritisk `BTN_E`: izpiši `B` na LCD.

Pri tem uporabite LCD modul, ki ga dobite na strani predmeta.

Uporaba LCD modula

Vhodi proti modulu:

- `CLK_in`, `RST_in`: urin signal in reset signal
- `DATA_TO_SEND_in`: ukaz, ki ga pošiljamo
  - `''0000000001''` – `clear screen`,
  - `''1001000001''` – izpiši `A`,
  - `''1001000010''` – izpiši `B`.
- `DATA_RDY_in`: postavimo na 1, ko so v vektorju `DATA_TO_SEND_in` veljavni podatki

Ko je LCD krmilnik prost, se izhod iz LCD modula `LCD_READY_out` postavi na `'1'`. Ko se ukaz izvaja je postavljen na `'0'`. V vektorju `DATA_TO_SEND_in` morajo biti veljavni podatki, dokler je `LCD_READY_out` enak `'0'`.

Uporaba LCD modula

Izhodi modula proti LCD krmilniku:

- `LCD_E_out`: LCD enable,
- `LCD_RS_out`: LCD register select,
- `LCD_RW_out`: LCD read write,
- `SF_D_inout`: podatkovni signali.

Predlog rešitve