

УПРАЖНЕНИЯ

по дисциплината: "Проектиране на интегрални цифрови схеми и системи"
за редовните студенти от специалност ЕЛЕКТРОНИКА, модул МИКРОЕЛЕКТРОНИКА
Учебна 2019/2020 година

Упражнение 5 – Крайни автомати във VHDL. Видове. Начини за описание.

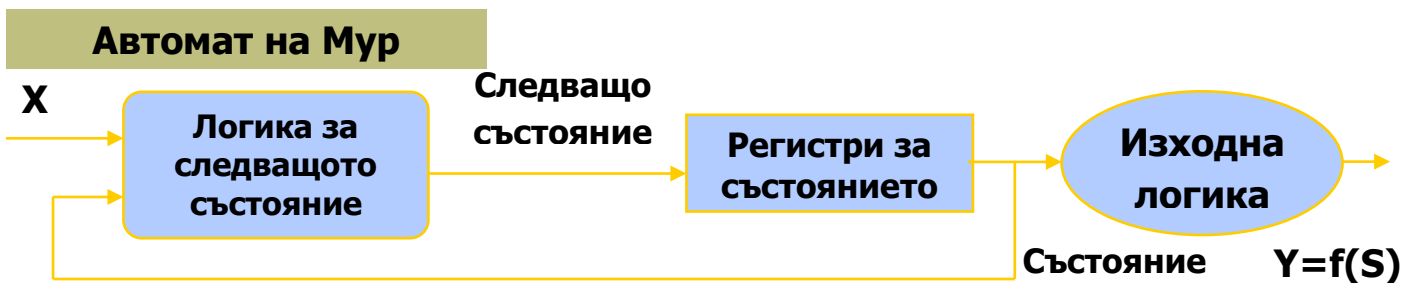
Теория:

Краен автомат е описание (модел), отговарящ на следните условия:

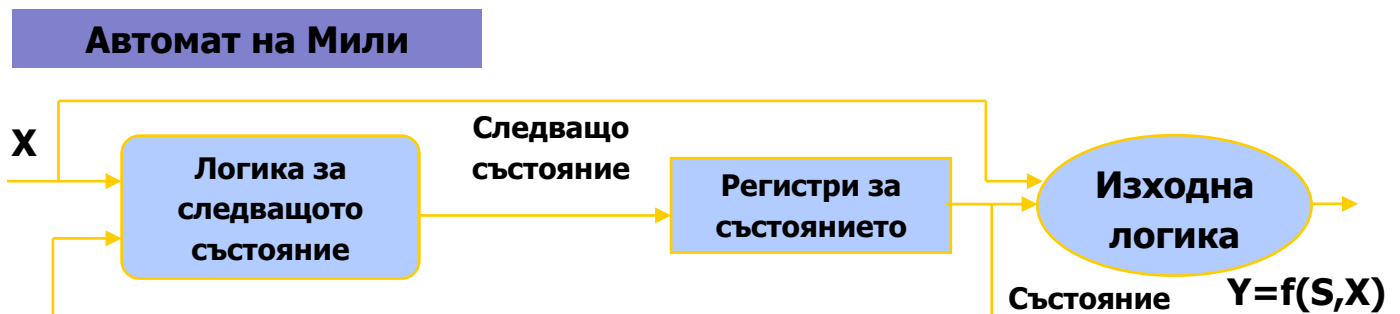
- За него е дефиниран набор (S) от краен брой състояния s ;
- Съществува начално състояние s_0 , което принадлежи към общия набор от състояния S, обикновено това е и състоянието, което се установява след Reset;
- За него е дефиниран набор от краен брой входни комбинации i ;
- За него са дефинирани функции на преход, определящи следващото възможно състояние $d=d(s,i)$;
- За него е дефинирана функция на изходите $f=f(s,i)$.

Съществуват няколко основни вида крайни автомати, в зависимост от това дали f е функция само на вътрешното състояние или зависи и от входната информация:

- Автомат на Мур (Moore machine) – стойността на изходния вектор е функция само на текущото състояние:



- Автомат на Мийли (Mealy machine) – стойността на изходния вектор зависи както от вътрешното състояние, така и от стойността на входния вектор:

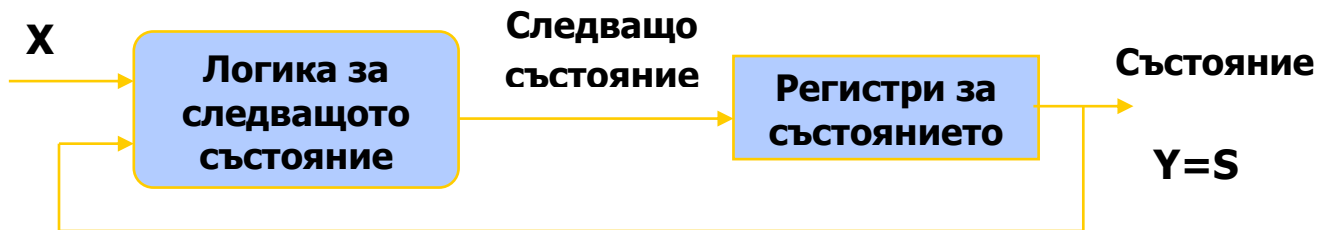


УПРАЖНЕНИЯ

по дисциплината: “Проектиране на интегрални цифрови схеми и системи”
за редовните студенти от специалност ЕЛЕКТРОНИКА, модул МИКРОЕЛЕКТРОНИКА
Учебна 2019/2020 година

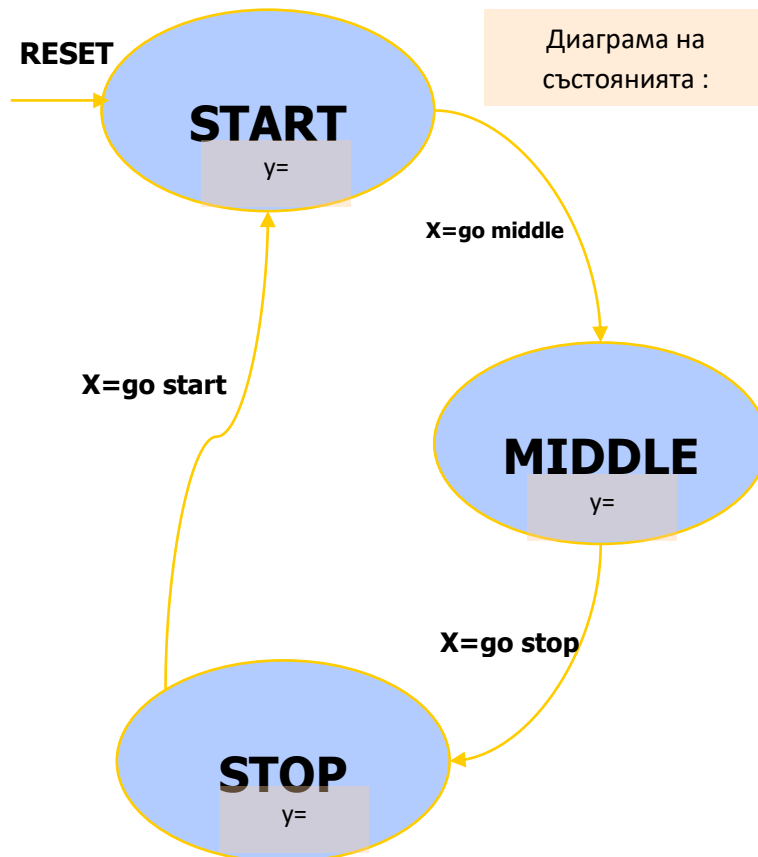
Съществува още една разновидност, която може да се разглежда като частен случай на автомат на Мур - известен е като Автомат на Медведев и при него стойността на изходния вектор съвпада с тази на вътрешния вектор на състоянията:

Автомат на Медведев



За представяне на крайните автомати се използват т.нар. диаграми на състоянието. Те представляват графично онагледяване на отделните състояния, функциите и условията за преход между тях и функциите определящи стойността на изходния вектор.

Пример за представяне на краен автомат с диаграма на състоянията:



УПРАЖНЕНИЯ

по дисциплината: “Проектиране на интегрални цифрови схеми и системи”
за редовните студенти от специалност ЕЛЕКТРОНИКА, модул МИКРОЕЛЕКТРОНИКА
Учебна 2019/2020 година

Тук началното състояние е **START**, то се установява след активен сигнал **RESET**.
Общо състоянията са три – **START**, **MIDDLE** и **STOP**. Условието за преход са дефинирани
чрез **x= ...**, а векторът на изходите чрез **y=...** за всяко едно състояние.

За описанието на крайни автомати могат да се използват различни оператори в езика
VHDL. Това може да стане и с оператор **IF ... ELSIF ...**, но е прието за описание на крайни
автомати да се използва оператор **CASE**. Синтаксисът на оператора има следния формат:

```
case израз is
  when условие за избор 1 => оператори;
  when условие за избор 2 => оператори;
  ...
  when условие за избор N => оператори;
  when others => оператори;
end case;
```

Изпълнението на оператора се състои в изчисляване на стойността на израза, задаващ
условието за избор, намиране на съответното условие за избор и изпълнение на операторите
в него. Ако не се намери съвпадение между всички зададени условия се изпълняват
операторите в условието по подразбиране **when others**. То е задължително за оператора, ако
не са зададени всички възможни условия за избор за съответния израз и отсъствието му води
до съобщение за грешка.

Друга особеност е използването на потребителски дефинирани типове за състоянията,
с което се постига по-добра четимост на описанието. Дефинирането на потребителски тип
данни става по следния начин:

```
TYPE име на типа IS (възможни стойности);
```

Пример за деклариране на тип и на сигнали от този тип при описание на краен
автомат - светофар:

```
type t_state is (red, green, yellow); -- деклариране на нов тип state с три възможни  
стойности
```

```
signal : present_state, next_state : t_state; -- деклариране на сигнали от тип state
```

УПРАЖНЕНИЯ

по дисциплината: “Проектиране на интегрални цифрови схеми и системи”
за редовните студенти от специалност ЕЛЕКТРОНИКА, модул МИКРОЕЛЕКТРОНИКА
Учебна 2019/2020 година

Тук са декларирани два сигнала – за текущото състояние и за следващото състояние на крайния автомат.

Краен автомат може да се опише чрез един или няколко процеса, в зависимост от сложността и вижданията на проектанта. По-долу са дадени примери за крайни автомати, описани с един и с два процеса:

Краен автомат с един процес:

--One_Process_FSM:

```
process (CLK, RESET)

    begin

        if RESET = '1' then

            STATE <= START;

        elsif CLK'event and CLK = '1' then

            case STATE is --<<-- Нач. на опис. на КА.

                when START => if x = go_mid then

                    STATE <= MIDDLE;

                end if;

                when MIDDLE => if x = go_stop then

                    STATE <= STOP;

                end if;

                when STOP => if x = go_start then

                    STATE <= START;

                end if;

                when others => STATE <= START;

            end case;    --<<-- Край на опис. на КА.

        end if;

    end process One_Process_FSM;
```

Краен автомат с два процеса:

УПРАЖНЕНИЯ

по дисциплината: “Проектиране на интегрални цифрови схеми и системи”
за редовните студенти от специалност ЕЛЕКТРОНИКА, модул МИКРОЕЛЕКТРОНИКА
Учебна 2019/2020 година

Процес за смяна на състоянията:

```
process (CLK, RESET)
begin
    if RESET = '1' then
        STATE <= START;
    elsif CLK'event and CLK = '1' then
        STATE <= NEXT_STATE;
    end if;
end process;
```

Процес за определяне на следващо състояние:

```
process (STATE , x)
begin
    case STATE is
        when START=> if x = go_mid then
            NEXT_STATE <= MIDDLE;
        end if;
        when MIDDLE => if x = go_stop then
            STATE <= STOP;
        end if;
        when STOP  => if x = go_start then
            STATE <= START;
        end if;
        when others => NEXT_STATE <= START;
    end case;
end process;
```

УПРАЖНЕНИЯ

по дисциплината: “Проектиране на интегрални цифрови схеми и системи”
за редовните студенти от специалност ЕЛЕКТРОНИКА, модул МИКРОЕЛЕКТРОНИКА
Учебна 2019/2020 година

Задаването на функцията за определяне на стойността на изходния вектор може да стане вътре в процеса на определяне на състоянията:

```
when MIDDLE => if x = go_stop then  
    STATE <= STOP;  
end if;  
Y <= F(STATE);           -- Автомат на Мур  
Y <= F(STATE, INPUT);   --Автомат на Мийли
```

или извън процесите с използване на паралелни оператори от типа на **with .. select...**

with STATE select

```
y<= A  when START,  
      B  when MIDDLE,  
      C  when STOP,  
      D  when others;
```

По-долу е даден пример за описание на краен автомат – светофар с използване на два процеса и задаване на стойностите на изходите в процеса за определяне на състояние:

```
entity traffic_light is  
    port ( sensor: in std_logic;  
          clock : in std_logic;  
          red_light : out std_logic;  
          green_light : out std_logic;  
          yellow_light : out std_logic );  
end traffic_light;  
architecture simple of traffic_light is  
    type t_state is (red, green, yellow);  
    signal present_state, next_state ; t_state;  
begin
```

УПРАЖНЕНИЯ

по дисциплината: “Проектиране на интегрални цифрови схеми и системи”
за редовните студенти от специалност ЕЛЕКТРОНИКА, модул МИКРОЕЛЕКТРОНИКА
Учебна 2019/2020 година

```
process(present_state, sensor)
begin
    case present_state is
    when green =>
        next_state <= yellow;
        red_light <= '0'; yellow_light <= '0'; green_light <= '1';
    when red =>
        if (sensor = '1') then next_state <= green;
        else next_state <= red;
        end if;
        red_light <= '1'; yellow_light <= '0'; green_light <= '0';
    when yellow =>
        next_state <= red;
        red_light <= '0'; yellow_light <= '1'; green_light <= '0';
    end case;
end process;

process
begin
    wait until clock'event and clk = '1';
    present_state <= next_state;
end process;
end simple;
```

Задачи:

1. Да се опише краен автомат по задание.
2. Да се състави test_bench за симулация на описанието.
3. Да се симулира описанието и да се снемат резултати.