Министерство науки высшего образования Российской Федерации

Федеральное государственное автономное образовательное учреждение высшего образования

«Казанский (Приволжский) Федеральный университет»

Институт физики

Кафедра радиофизики

Направление: - Информационная безопасность

Профиль: Безопасность автоматизированных систем

«Теория и применение микропроцессоров»

Автомат Мили

Выполнил:

студент 4 курса группы 06-751

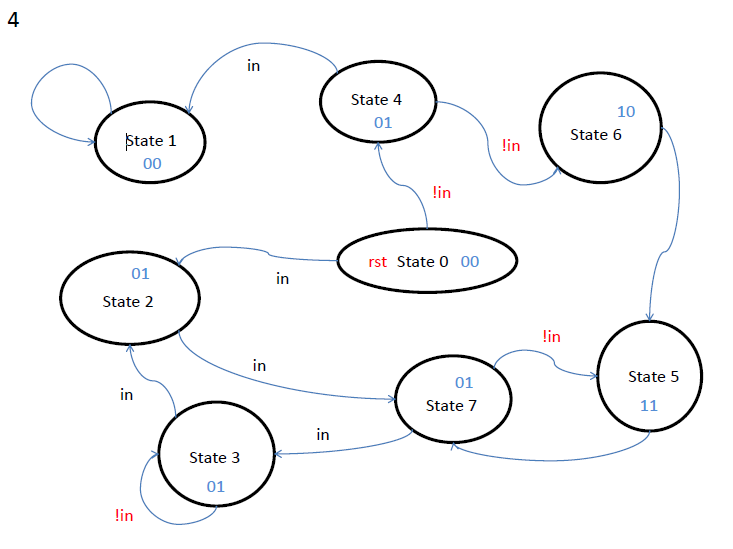
Галлямов Айрат

Казань 2020

Автомат Мили – конечный автомат, выходная последовательность которого зависит от состояния автомата и входных сигналов.

Автоматы Мили. Описывается системой уравнений:  
c(t) = δ( a(t), c(t-1) );  
b(t) = λ( a(t), c(t-1) ).

1. В данной работе мной был использован четвертый вариант графа, представленного в учебно-методическом пособии. На рис.1 представлен исходный граф автомата Мили.



*Рис.1 Исходный граф*

1. Любой автомат можно представить с помощью таблицы переходов и выходов (ТПВ). Построим ТПВ для нашего графа Мили (таблица 1):

*Таблица 1. ТПВ 4 графа Мили*

|  |  |  |
| --- | --- | --- |
|  | !in | in |
| State 0 | State 4/01 | State 2/01 |
| State 1 | State 1/00 | State 1/00 |
| State 2 | State 2/01 | State 7/01 |
| State 3 | State 3/01 | State 2/01 |
| State 4 | State 6/10 | State 1/00 |
| State 5 | State 7/01 | State 7/01 |
| State 6 | State 5/11 | State 5/11 |
| State 7 | State 5/11 | State 3/01 |
| reset | State 0 | |

1. Создадим модель нашего автомата Мили на языке Verilog, используя среду для симуляции и отладки ModelSim.
2. В проекте создадим два файла: Mealy.v и MealyTestbench.v. В первом файле будет содержаться логика переходов и состояния автомата, с помощью второго файла мы прогоним тест, он предоставит нашему автомату входные сигналы.
3. Файл Mealy.v.

Создадим модуль и определим входные и выходные порты:

**module** Mealy

**(input** clk**,** reset**,** a\_in**,** **output** **reg** **[**1**:**0**]** b\_out**,** **reg** **[**2**:**0**]** st**);**

Далее определим значения состояний:

**parameter** **[**2**:**0**]** st0**=**3'b000**,**

st1**=**3'b001**,**

st2**=**3'b010**,**

st3**=**3'b011**,**

st4**=**3'b100**,**

st5**=**3'b101**,**

st6**=**3'b110**,**

st7**=**3'b111**;**

Создадим структурные конструкции *always*, содержащую логику переходов состояний и логику выходных состояний автомата Мили:

**always@(posedge** clk **or** **posedge** reset **or** st**)**

**begin**

**if(**reset**)**

st**=**st0**;**

**else**

**case(**st**)**

st0**:**

**if(**a\_in**==**0**)**

**begin**

st**=**st4**;**

b\_out**=**2'b01**;**

**end**

**else**

**begin**

st**=**st2**;**

b\_out**=**2'b01**;**

**end**

st1**:**

**if(**a\_in**==**0**)**

**begin**

st**=**st1**;**

b\_out**=**2'b00**;**

**end**

**else**

**begin**

st**=**st1**;**

b\_out**=**2'b00**;**

**end**

st2**:**

**if(**a\_in**==**0**)**

**begin**

st**=**st2**;**

b\_out**=**2'b01**;**

**end**

**else**

**begin**

st**=**st7**;**

b\_out**=**2'b01**;**

**end**

st3**:**

**if(**a\_in**==**0**)**

**begin**

st**=**st3**;**

b\_out**=**2'b01**;**

**end**

**else**

**begin**

st**=**st2**;**

b\_out**=**2'b01**;**

**end**

st4**:**

**if(**a\_in**==**0**)**

**begin**

st**=**st6**;**

b\_out**=**2'b10**;**

**end**

**else**

**begin**

st**=**st1**;**

b\_out**=**2'b00**;**

**end**

st5**:**

**if(**a\_in**==**0**)**

**begin**

st**=**st7**;**

b\_out**=**2'b01**;**

**end**

**else**

**begin**

st**=**st7**;**

b\_out**=**2'b01**;**

**end**

st6**:**

**if(**a\_in**==**0**)**

**begin**

st**=**st5**;**

b\_out**=**2'b11**;**

**end**

**else**

**begin**

st**=**st5**;**

b\_out**=**2'b11**;**

**end**

st7**:**

**if(**a\_in**==**0**)**

**begin**

st**=**st5**;**

b\_out**=**2'b11**;**

**end**

**else**

**begin**

st**=**st3**;**

b\_out**=**2'b01**;**

**end**

**default** st**=**st0**;**

**endcase**

**end**

**endmodule**

1. Файл MealyTestbench.v.

Пропишем формирующее входное воздействие для тактовых импульсов *clk*. Установим его начальное состояние и период переключения. Так же установим значения входного сигнала *a\_in* согласно определенному порядку.

**module** MealyTestbench**;**

**reg** clk**,** reset**,** a\_in**;**

**wire** **[**1**:**0**]** b\_out**;**

**wire** **[**2**:**0**]** st**;**

Mealy test **(**clk**,** reset**,** a\_in**,** b\_out**,** st**);**

**initial**

**begin**

clk**=**1**;**

**forever** **#**50 clk**=!**clk**;**

**end**

**initial**

**begin**

reset**=**0**;**

**#**800 reset**=**1**;**

**#**100 reset**=**0**;**

**#**1300 reset**=**1**;**

**#**100 reset**=**0**;**

**end**

**initial**

**begin**

a\_in**=**0**;**

**#**100 a\_in**=**0**;**

**#**100 a\_in**=**0**;**

**#**100 a\_in**=**1**;**

**#**100 a\_in**=**0**;**

**#**100 a\_in**=**1**;**

**#**100 a\_in**=**0**;**

**#**100 a\_in**=**1**;**

**#**100 a\_in**=**0**;**

**#**100 a\_in**=**0**;**

**#**100 a\_in**=**1**;**

**#**100 a\_in**=**0**;**

**#**100 a\_in**=**1**;**

**#**100 a\_in**=**0**;**

**#**100 a\_in**=**1**;**

**#**100 a\_in**=**1**;**

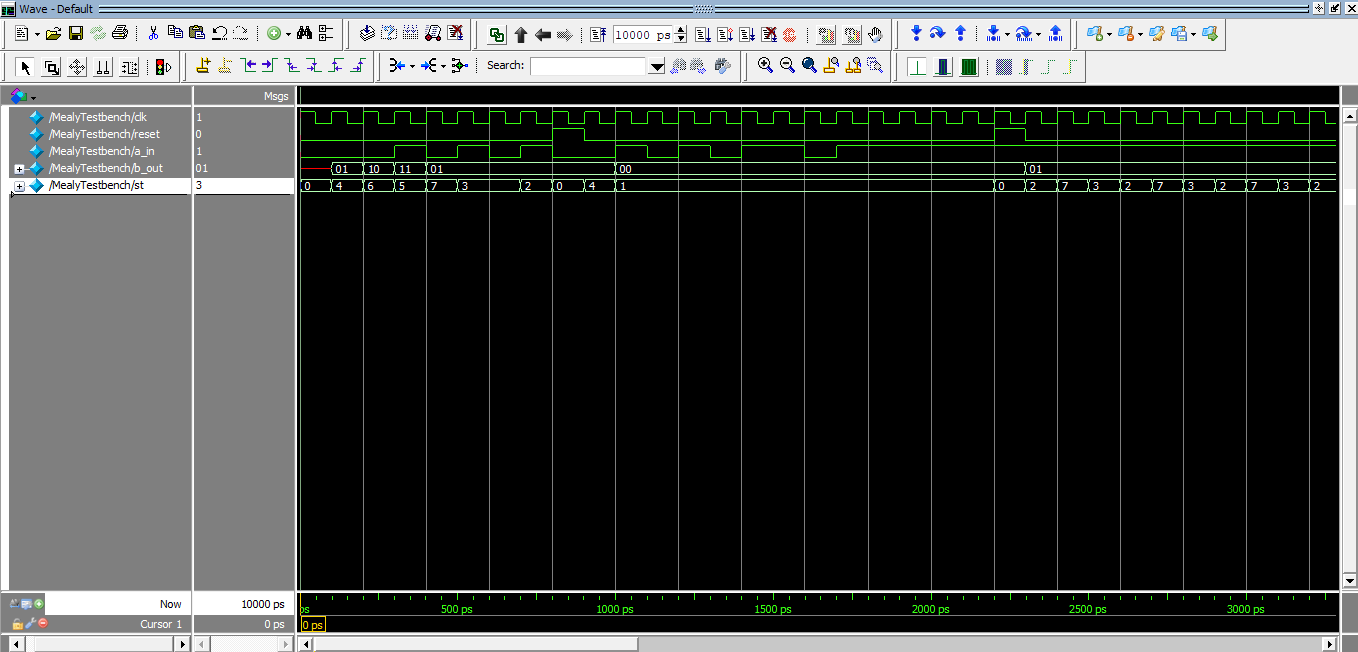
**#**100 a\_in**=**0**;**

**#**100 a\_in**=**1**;**

**end**

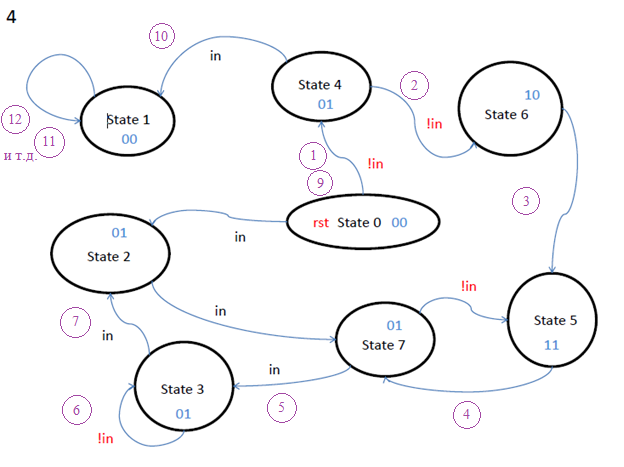
**endmodule**

1. Следующий этап – симуляция работы программы «автомат Мили». На рисунке 2 представлена визуализация работы исходного автомата.



*Рис.2. Симуляция исходного автомата Мили*

1. Переходы состояний на рис.2 происходят согласно схеме приведенной на рис.3.



*Рис.3. Схема перехода состояний*

Заключение: в данной работе была реализована модель автомата Мили по заданному графу (вариант 4) на языке Verilog в среде ModelSim, а также визуально представлены переходы состояний построенного автомата согласно схеме на рис.3.