ЛАБОРАТОРНАЯ РАБОТА № 2

ПОСТРОЕНИЕ ДИСКРЕТНО-ДЕТЕРМИНИРОВАННЫХ МОДЕЛЕЙ

Цель работы: получить представление о дискретно-детерминированных моделей, построении их динамических характеристик, а также проверке их на адекватность.

Используемые программы: Matlab, Simulink.

Общие сведения

Дискретно-детерминированные модели называются также конечными автоматами (англ. finite automat), или F-схемами. F-схемы характеризуются шестью элементами: конечным множеством X входных сигналов (входным алфавитом); конечным множеством Y выходных сигналов (выходным алфавитом); конечным множеством Z внутренних состояний (внутренним алфавитом или алфавитом состояний); начальным состоянием автомата; функцией переходов $\varphi(z, x)$; функцией выходов $\psi(z, x)$. Работа конечного автомата происходит по следующей схеме: в каждом i-м такте на вход автомата, находящегося в состоянии z_i подается некоторый сигнал x_i на который он реагирует переходом в новое состояние z_{i+1} и выдачей некоторого выходного сигнала y_i. Существуют разновидности конечных автоматов.

автомат Мура первого рода (рис. 1, *а*):

$$\begin{cases} z_{i+1} = \varphi(z_i, x_i), \\ y_i = \psi(z_i), i = 0,1,2,... \end{cases}$$

автомат Мура второго рода (рис. 1, *б*):

$$\begin{cases} z_{i+1} = \varphi(z_i, x_i), \\ y_{i+1} = \psi(z_{i+1}), i = 0,1,2,... \end{cases}$$

автомат Мили первого рода (рис. 2, *а*):

$$\begin{cases} z_{i+1} = \varphi(z_i, x_i), \\ y_i = \psi(z_i, x_i), i = 0,1,2,...; \end{cases}$$

автомат Мили второго рода (рис. 2, *б*):

$$\begin{cases} z_{i+1} = \varphi(z_i, x_i), \\ y_{i+1} = \psi(z_{i+1}, x_i), i = 0,1,2,...; \end{cases}$$

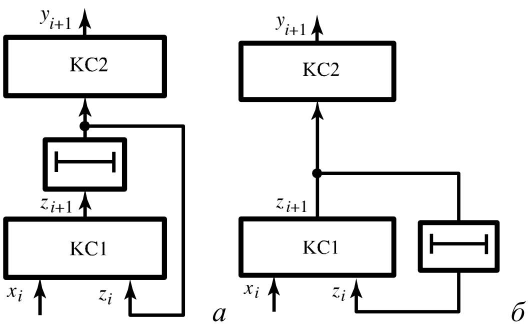


Рис. 1. Автомат Мура:
КС – комбинационная схема;
|–| - элемент задержки

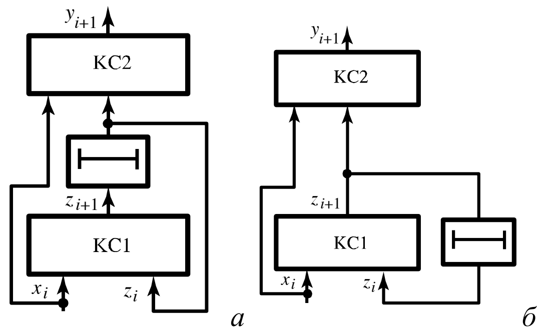


Рис. 2. Автомат Мили

Пример табличного способа задания F-автомата Мура приведен в табл. 1, а для F-автомата Мили с тремя состояниями, двумя входными и двумя выходными сигналами - в табл. 2.

Табл. 1

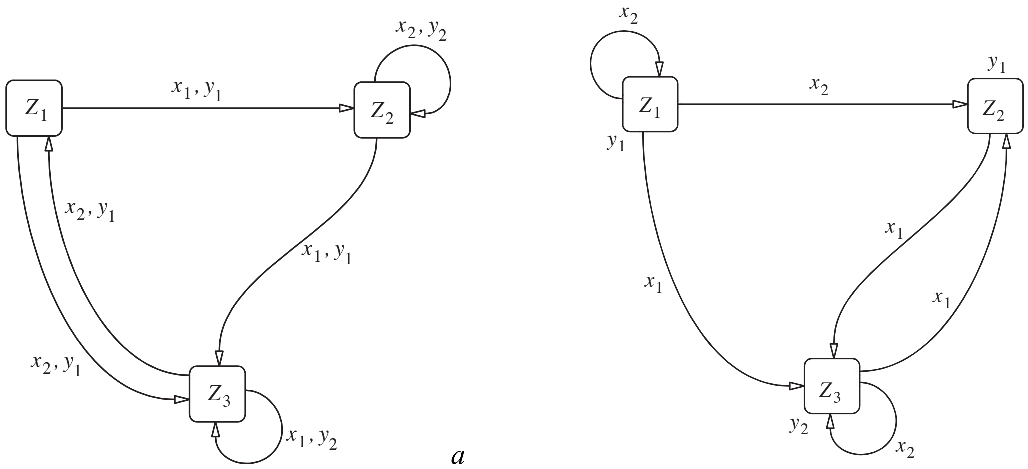
x_i	y_i		
	y_1	y_1	y_2
	z_1	z_2	z_3
x_1	z_3	z_3	z_2
x_2	z_1	z_1	z_3

Табл. 2

x_i	z_k		
	z_1	z_2	z_3
Переходы			
x_1	z_2	z_3	z_3
x_2	z_3	z_2	z_1
Выходы			
x_1	y_1	y_1	y_2
x_2	y_1	y_2	y_1

При графическом способе задания конечного автомата применяется направленный граф, вершины которого соответствуют состояниям автомата, дуги - переходам.

На рис. 3, *а*, *б* приведены заданные ранее таблицами *F*-автоматы Мили и Мура соответственно.



а *б*

Рис. 3. Графический способ задания *F*- автомата

Пример

В качестве примера рассмотрим автомат Мура первого рода. Пусть, например, требуется, чтобы некий объект выполнял команды **направо**, **налево** и **кругом**, поворачиваясь в соответствующую сторону. Вид этого объекта в результате выполнения команды зависит не только от самой команды, но и от состояния объекта, в котором он находился.

Пусть состояние определяет, какой стороной объект обращен к нам: **анфас**, **левая**, **правая**, **зад**. Тогда функцию переходов состояний автомата, моделирующего указанное поведение, можно представить в виде графа или таблицы. Так, если объект находится в состоянии **анфас**, то при выполнении команды **направо** он должен показать нам свой левый бок, т.е. перейти в состояние **левая**. Если повторить ту же команду, то объект перейдет в состояние **зад**.

В рассматриваемом примере в качестве функции выхода автомата можно взять такую, которая сопоставляет каждому состоянию соответствующее графическое изображение объекта. Возможный вариант показан ниже. Вы можете выбрать команду чтобы изменить положение автомобиля (рис. 4).

z_1 z_2 z_3 z_4

Рис. 4. Автомобиль, повернись.

Обозначим состояние «анфас» - z_1, «левая» - z_2, «правая» - z_3, «зад» - z_4, а команду «направо» - x_1, «налево» - x_2, «кругом» - x_3. Составим таблицу состояний (табл. 3) и соответствующий ей граф (рис. 4)

Табл. 3

x_i	y_1	y_2	y_3	y_4
	z_1	z_2	z_3	z_4
x_1	z_2	z_4	z_1	z_3
x_2	z_3	z_1	z_4	z_2
x_3	z_4	z_3	z_2	z_1

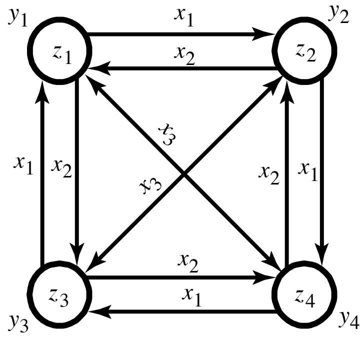


Рис. 4.

Запишем в каком случае возникает каждое состояние как сумму произведений предыдущего состояния на команду, вызвавшую переход:

$$z_1 = x_1 z_3 \vee x_2 z_2 \vee x_3 z_4;$$

$$z_2 = x_1 z_1 \vee x_2 z_4 \vee x_3 z_3;$$

$$z_3 = x_1 z_4 \vee x_2 z_1 \vee x_3 z_2;$$

$$z_4 = x_1 z_2 \vee x_2 z_3 \vee x_3 z_1.$$

Первоначальное состояние автомата примем z_1.

Для рассмотренного автомата Мура отрывок программной реализации может иметь вид:

```
switch z
   case 1,
      if (x==1) y=2;
      end
      if (x==2) y=3;
      end
      if (x==3) y=4;
      end
   case 2,
      if (x==1) y=4;
      end
      if (x==2) y=1;
      end
      if (x==3) y=3;
      end
   case 3,
      if (x==1) y=1;
      end
      if (x==2) y=4;
      end
      if (x==3) y=2;
      end
   case 4,
      if (x==1) y=3;
      end
      if (x==2) y=2;
      end
      if (x==3) y=1;
      end
end
```

Порядок работы

1. Придумать F-автомат, представив его в табличном и графическом видах, и составить его модель. Начальное состояние автомата и входное слово генерируются самостоятельно.
2. Проверить модель на адекватность.

В отчете представить:

- теоретическое введение;
- номер индивидуального варианта;
- исходные данные;
- структурную схему и порядок синтеза модели;
-результаты проверки на адекватность;
- временные диаграммы работы автомата при заданном входном воздействии;
- выводы по проделанной работе.

ВАРИАНТЫ ЗАДАНИЙ

Тип автомата и размерность входного алфавита x_n, выходного алфавита y_m и состояний z_p выбрать согласно варианту:

№ п/п	Тип автомата	*n*	*m*	*p*
1	Мили (1-го рода)	2	2	4
2	Мили (2-го рода)	3	2	4
3	Мура (1-го рода)	4	2	4
4	Мура (2-го рода)	4	3	3
5	Мили (1-го рода)	3	3	3
6	Мили (2-го рода)	2	3	3
7	Мура (1-го рода)	3	4	2
8	Мура (2-го рода)	4	4	2
9	Мили (1-го рода)	2	2	2
10	Мили (2-го рода)	3	5	3