Глава 3

Структурные модели и кодирование состояний автоматов

До построения формальной спецификации в виде графов переходов выберем структурную модель каждого автомата и вид кодирования его состояний.

В зависимости от наличия памяти, требующейся для реализации заданного алгоритма, автоматы могут быть разделены на два больших класса: автоматы без памяти (комбинационные) и автоматы с памятью (последовательностные).

3.1. Комбинационные автоматы

Автомат, значения выхода (Z) которого зависят только от значений входов *(X)* в данный момент времени и не зависят от входных воздействий в предшествующие моменты времени, называется комбинационным (однотактным), или автоматом без памяти.

В структурных моделях автоматы этого класса называются комбинационными схемами (КС).

Функционирование КС (рис. 3.1) описывается математической моделью вида

$$Z = f(X).$$

Функция, описывающая это соотношение для одного выхода, называется булевой функцией (БФУ).

БФУ задаются с помощью полностью или не полностью определенных таблиц, которые называются таблицами истинности (ТИ) и таблицами решений соответственно. Табл. 3.1 является ТИ для функции «конъюнкция» двух переменных x_1 и x_2

Более компактной формой представления является задание булевых функций в виде булевых формул, которые в отличие от таблиц всегда определены на всех 2^n *(n* — число входных переменных) наборах. Для приведенной таблицы $z = x_1x_2$ или $z = x_1\&x_2$

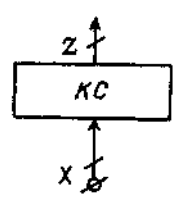

Рис. 3.1

Таблица 3.1

x_1	x_2	z
0	0	0
0	1	0
1	0	0
1	1	1

Задание БФУ в виде ГП нерационально, так как для этого класса автоматов отсутствует необходимость в отражении динамики переходов. Это, однако, не исключает принципиальную возможность отображения булевых функций в виде ГП (разд. 4.4.1).

3.2. Последовательностные автоматы

Автомат, значения выходов (Z) которого зависят не только от значений входов *(X)* в данный момент времени, но и от его состояния, называется последовательностным (многотактным), или автоматом с памятью.

Канонический метод синтеза автоматов [8] сводит задачу их структурного синтеза к реализации одной или двух комбинационных схем, первая из которых охвачена обратными связями.

Для того чтобы различить настоящий и следующий моменты времени, в модели автоматов вводятся «однотактные» элементы задержки (ЭЗ), которые по этой причине отличаются от функциональных элементов задержки, используемых в управляющих автоматах для реализации временных интервалов требуемой продолжительности.

Важной особенностью программной реализации автоматов является то, что элементы задержки (в отличие от ФЭЗ в УА) в программу в явнсм виде не вводятся, а их функции реализуют, например, при описании автомата с помощью СБФ за счет переобозначений выходных или вну - ренних переменных.

Другой особенностью такой реализации автоматов является то, что синхронность выполнения программ обеспечивает адекватность поведения автомата и используемой модели, что может не иметь места при аппаратной реализации [13], которая в системах управления обычно является асинхронной.

В зависимости от того, где в автоматах располагаются ЭЗ, они разбиваются на два подкласса (рода) [3].

В автоматах первого рода задержка располагается непосредственно в цепи «вход—выход», а в автоматах второго рода она размещается в цепи обратной связи. Поэтому если в автоматах первого рода выходные сигналы запаздывают на такт по сравнению с входными сигналами, то в автоматах второго рода запаздывание практически отсутствует.

Автоматы без выходного преобразователя (АБВП). Структурная схема АБВП первого рода представлена на рис. 3.2, второго рода — на рис. 3.3. На этих схемах символом «|—|» обозначены ЭЗ.

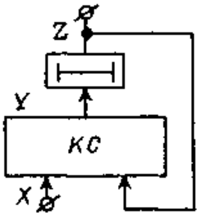

Рис. 3.2

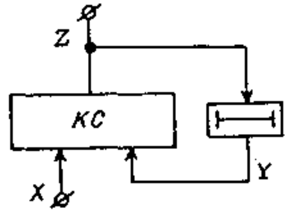

Рис. 3.3

В настоящей работе для автоматов с памятью в основном используются следующие обозначения переменных: X — входные переменные; Y — внутренние переменные; Z — выходные переменные. Эти обозначения носят мнемонический характер, так как упоминаются в том же порядке, как и в латинском алфавите.

Исходя из изложенного для автоматов рассматриваемого класса могут быть использованы следующие математические модели:

$$\begin{cases} Y = f(X, Z); \\ Z = Y; \end{cases} \qquad \begin{cases} Z = f(X, Y); \\ Y = Z. \end{cases}$$

Модели АБВП не являются универсальными, так как при их использовании коды состояний автомата принудительно должны совпадать с кодами выходных переменных, что возможно лишь в тех случаях, когда в ГП отсутствуют различные вершины с одинаковыми комбинациями значений выходных переменных.

Частным случаем АБВП являются автоматы без выхода (АБВ). Их структура совпадает со структурой автоматов, приведенных на рис. 3.2 и 3.3, однако во внешнюю среду они выдают не выходные переменные Z, а внутренние переменные Y. Использование этого подкласса автоматов может резко упростить описание композиций автоматов.

Универсальные автоматы без выходного преобразователя. Если в различных состояниях автомат должен выдать одинаковые наборы значений выходных переменных, то для его реализации может быть использована модель универсального АБВП первого (рис. 3.4) или второго (рис. 3.5) рода.

Универсальность этих моделей достигается за счет кодирования состояний автомата, позволяющего различать состояния, в том числе и с

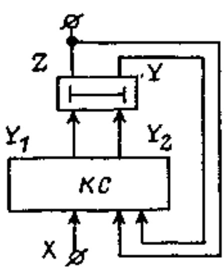

Рис. 3.4

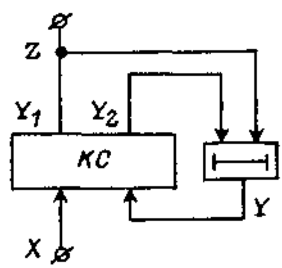

Рис. 3.5

одинаковыми значениями выходных переменных. При этом одна часть каждой кодовой комбинации принудительно задается вектором значений выходных переменных, а другая — выбирается свободно за счет значений внутренних переменных, вводимых с целью различения состояний с одинаковыми значениями выходных переменных.

Автоматы Мура (АМ). Отказаться от полного или частичного принудительного кодирования состояний возможно при введении в модель автомата второй комбинационной схемы КС2, осуществляющей преобразование свободно выбранных кодов состояний в наборы значений выходных переменных.

Структурные схемы АМ первого и второго рода приведены на рис. 3.6 и 3.7 соответственно. Эти автоматы описываются следующим образом:

$$\begin{cases} Y' = f(X, Y); \\ Z = \varphi(Y); \\ Y = Y'; \end{cases} \qquad \begin{cases} Y' = f(X, Y); \\ Z = \varphi(Y'); \\ Y = Y'. \end{cases}$$

При этом необходимо отметить, что автомат Мура второго рода всегда может быть преобразован в эквивалентный ему автомат Мура первого рода только в результате замены выходного преобразователя КС2, но не наоборот, т. е. произвольный автомат Мура первого рода не может быть заменен эквивалентным автоматом Мура второго рода путем лишь одной замены выходного преобразователя [3].

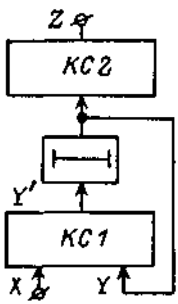

Рис. 3.6

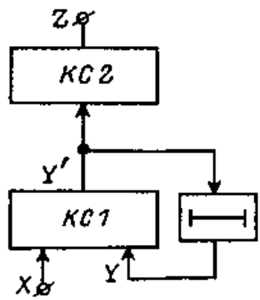

Рис. 3.7

Автоматы Мили (АМИ). Для автоматов Мура характерно, что его выходные переменные зависят только от переменных состояний и не зависят от входных переменных. Для ряда алгоритмов более естественным является реализация автомата в виде автомата Мили, для которого характерно, что его выходные переменные зависят не только от переменных состояний, но и от входных переменных (рис. 3.8, 3.9).

Эти автоматы первого и второго рода описываются следующим образом:

$$\begin{cases} Y' = f(X, Y); \\ Z = \varphi(X, Y); \\ Y = Y'; \end{cases} \qquad \begin{cases} Y' = f(X, Y); \\ Z = \varphi(X, Y'); \\ Y = Y'. \end{cases}$$

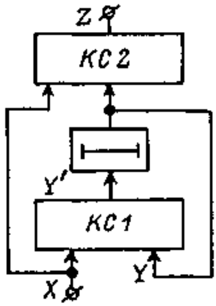

Рис. 3.8

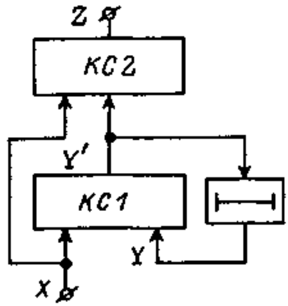

Рис. 3.9

В силу того что вторая комбинационная схема КС2 в автоматах Мили получает в отличие от автоматов Мура также и входную информацию, можно утверждать, что число состояний автомата Мили не превышает числа состояний эквивалентного ему автомата Мура [52].

Автоматы с сохранением значений выходных переменных. Автоматы Мура и Мили первого и второго рода, в которых обеспечивается запоминание выходных переменных, будем называть автоматами с сохранением значений выходных переменных.

Эти автоматы могут рассматриваться также и как суперпозиция двух простейших типов автоматов: автомата без выхода первого или второго рода и АБВП второго рода.

Возможны четыре структуры автоматов рассматриваемого класса: две — на основе автоматов Мура (рис. 3.10, 3.11) и две — на основе автоматов Мили (рис. 3.12, 3.13).

Автоматы Мура этого класса описываются соотношениями:

$$\begin{cases} Y' = f(X, Y); \\ Z = \varphi(Y, Z); \\ Y = Y'; \end{cases} \qquad \begin{cases} Y' = f(X, Y); \\ Z = \varphi(Y', Z); \\ Y = Y', \end{cases}$$

а автоматы Мили этого класса соответственно:

$$\begin{cases} Y' = f(X, Y); \\ Z = \varphi(X, Y, Z); \\ Y = Y'; \end{cases} \qquad \begin{cases} Y' = f(X, Y); \\ Z = \varphi(X, Y', Z); \\ Y = Y'. \end{cases}$$

Отметим также и то, что существуют автоматы без выходного преобразователя, принадлежащие к классу автоматов с сохранением значений выходных переменных. Для автоматов рассматриваемого класса в ГП сохраняемые значения выходных переменных могут умалчиваться и обозначаться прочерками.

Автоматы с общим выходным преобразователем. В случае если заданный алгоритм описан совокупностью автоматов, то в силу того что при программной реализации они выполняются последовательно, выходной набор при этом формируется по частям, что в общем случае может быть весьма опасным.

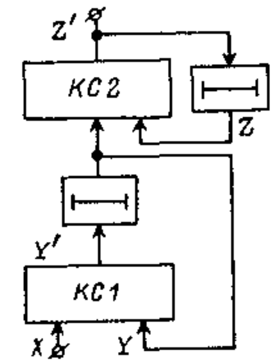

Рис. 3.10

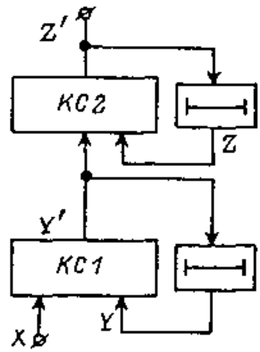

Рис. 3.11

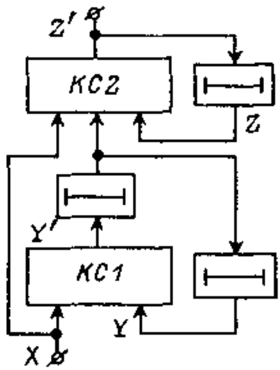

Рис. 3.12

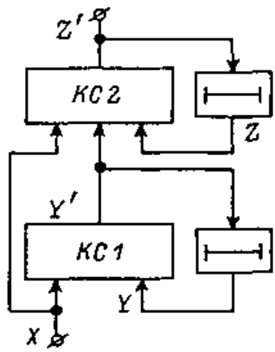

Рис. 3.13

В настоящей работе предлагается модель совокупности автоматов без выхода с общим выходным преобразователем (рис. 3.14). При этом до срабатывания последнего автомата совокупности сохраняется предыдущий выходной набор, а после его срабатывания общий выходной преобразователь формирует новый набор. Таким образом, в предлагаемой модели в течение переходного процесса выходной набор не изменяется и лишь после его завершения формируется новый, что делает поведение совокупности автоматов адекватным поведению единого автомата.

Автоматы с общим входным преобразователем. В случае если автомат задан одним ГП, то каждая булева формула, помечающая дугу ГП, может быть реализована независимо. Однако если все булевы формулы, содержащие более одной буквы, обозначать новыми переменными, предполагая, что одинаковые формулы имеют одно и то же обозначение, то появляется возможность их совместной минимизации и формирования общего входного преобразователя.

Это позволяет говорить о модели автомата с общим входным преобразователем (рис. 3.15). При этом собственно у автомата, входящего в эту

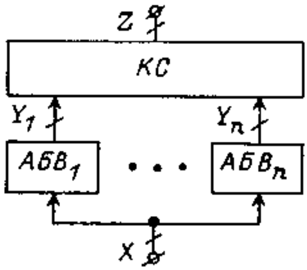

Рис. 3.14

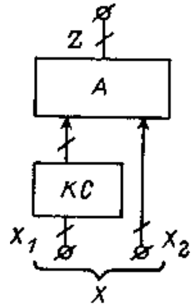

Рис. 3.15

модель, как в классической теории автоматов [50], все формулы на дугах однобуквенны. Аналогичная модель может быть использована и в случае, если алгоритм реализуется совокупностью автоматов.

В последующих разделах рассматриваются также и другие модели автоматов, которые целесообразно использовать при построении систем логического управления, например смешанные автоматы или автоматы с флагами (гл. 9).

3.3. Кодирование состояний автоматов

При построении графа переходов каждой вершине (состоянию) присваивается номер — некоторое целое десятичное положительное число.

Принудительное кодирование. Если во всех состояниях ГП коды выходных переменных различаются, то каждое число, обозначающее номер состояния, может быть закодировано двоичным вектором, совпадающим с вектором значений выходных переменных в этом состоянии. Такой вид кодирования будем называть «принудительным». Структурная модель автомата, использующего этот вид кодирования, соответствует автомату без выходного преобразователя. Однако если в ГП в различных состояниях должны формироваться одинаковые значения выходных переменных, то при использовании этого вида кодирования такой ГП автоматом указанного типа не может быть реализован.

Принудительно-свободное кодирование. Универсальность автомата без выходного преобразователя при реализации ГП, в которых векторы значений выходных переменных совпадают в некоторых состояниях, обеспечивается принудительно-свободным кодированием состояний.

Кодирование в этом случае осуществляется следующим образом: разряды принудительной части вектора каждого состояния совпадают со значениями выходных переменных рассматриваемого состояния, а количество разрядов в свободной части вектора равно двоичному логарифму числа состояний, в которых выходные векторы совпадают. Значения разрядов в свободной части кодов состояний, в которых значения выходных переменных совпадают, должны различаться между собой, а для остальных состояний выбор свободной части векторов произволен. Пусть, например, автомат с тремя состояниями имеет следующие векторы значений выходных переменных: 00, 01, 01. Они образуют принудительную часть векторов состояний. В данном случае в двух состояниях значения выходных переменных совпадают и поэтому в свободной части векторов достаточно иметь $\log_2 2 = 1$ разряд, различающий эти состояния. Таким образом, может использоваться один из следующих вариантов кодирования: 00*, 010, 011 или 00*, 011, 010, где * — символ, обозначающий произвольное значение двоичного разряда. Этот вид кодирования обеспечивает универсальность автомата без выходного преобразователя.

Разновидности свободного кодирования. Если автомат содержит выходной преобразователь, как это имеет место, например, в автоматах Мура, то векторы кодов состояний автомата могут выбираться независимо (свободно) от векторов значений выходных переменных, так как именно с помощью этого преобразователя и может быть осуществлено перекодирование произвольного вектора состояния в требуемый вектор выходных переменных.

Свобода кодирования во многом связана с программной реализацией алгоритмов, так как в данном случае в отличие от схем в применении противогоночных или помехоустойчивых кодов [27] нет необходимости.

Рассмотрим четыре вида кодирования, которые будем использовать для автоматов с выходным преобразователем: логарифмическое, унитарное (единичное), двоичное и многозначное.

Логарифмическое кодирование. Если десятичный номер каждого состояния закодировать числом, равным двоичному эквиваленту номера, или сопоставить десятичному номеру некоторое двоичное число, количество разрядов которого равно $]\log_2 s[$ (где s — число состояний автомата;] [— символ округления до ближайшего большего целого), то такой вид кодирования называется логарифмическим кодированием.

Унитарное (единичное) кодирование. При использовании унитарного кодирования в его классической трактовке [13] i-й вершине сопоставляется 5-местный вектор, в котором только на i-й позиции размещается единица, а на остальных позициях — нули.

Двоичное кодирование. Присвоим i-й вершине ГП только одну переменную Y_i. Эта переменная равна единице в случае, когда автомат находится в этой вершине, и нулю — в противном случае.

Такой вид кодирования связан с присвоением каждому состоянию 2^{s-1} векторов, каждый из которых s-местный, причем различным состояниям присваиваются в том числе и одинаковые векторы. В этом случае, как и в предыдущем, рабочими являются векторы с одной единицей, в то время как все остальные векторы рабочими не являются.

Наличие только одной единицы в каждом рабочем векторе является признаком для обнаружения факта перехода автомата в нерабочее состояние, что невозможно при логарифмическом кодировании. Этот вид кодирования обеспечивает построение весьма простых булевых формул, бесповторных по переменным состояний.

Многозначное кодирование. Если в качестве кода i-го состояния использовать десятичный номер, присвоенный i-й вершине ГП, то это означает, что в автомате используется многозначное кодирование, т. е. все состояния автомата кодируются одной переменной Y, которая может принимать s значений.