

Оглавление

ВВЕДЕНИЕ	3
1. Основные понятия теории автоматов.....	4
2. Классификация дискретных автоматов.....	6
3. Языки описания цифровых автоматов	10
3.1 Начальные языки описания	10
3.2 Автоматные языки описания.....	14
3.3 Описание комбинационных автоматов	18
4. Алгоритмы синтеза дискретных автоматов.....	19
4.1 Синтез комбинационных автоматов.....	19
4.1.2 Функционально полные системы логических функций. Алгоритмы получения формул в заданных базисах.....	19
4.2 Синтез схемы голосования «два из трех»	21
4.3 Синтез дискретных автоматов с памятью.....	23
4.4 Пример синтеза дискретных автоматов	25
4.4.1 Этап абстрактного синтеза	25
4.4.2 Минимизация внутренних состояний	26
4.4.3 Структурный синтез дискретных автоматов	29
4.4.4 Определение функций возбуждения и выходных функций	32
4.4.5 Функциональные схемы автоматов. Моделирование функционирования	36
5. Реализация дискретных автоматов на современной элементной базе	39
Список литературы.....	41
Приложение 1	42
Задание на синтез дискретного автомата.....	42
Приложение 2	43
Приложение 3	44

ВВЕДЕНИЕ

Современные приборы и устройства сервиса представляют собой сложные технические системы, реализованные на базе средств вычислительной техники. Цифровые устройства и микропроцессоры являются важнейшей составной частью различных объектов бытовой техники: радиоэлектронной аппаратуры, стиральных и посудомоечных машин, холодильников и климат-систем, изделий оргтехники и других устройств.

Такой широкий диапазон применения цифровых устройств определяется их высокими техническими параметрами и технико-экономическими показателями. В частности - это низкое энергопотребление, высокое быстродействие, высокая надежность и помехозащищенность, возможность реализации алгоритмов управления и обработки сигналов любой сложности, малые габариты, технологичность и низкая стоимость.

Управляющие автоматы реализуются в виде "гибкой" или программируемой логики, на основе микропроцессоров, а также в виде "жесткой логики" - на основе последовательностных цифровых устройств. Математическими моделями, используемыми при анализе и синтезе последовательностных устройств, являются конечные автоматы, имеющие конечное количество элементов памяти. В последнее время интерес к конечным автоматам возрос, что связано с развитием интегральной программируемой электроники: программируемых логических структур; программируемых логических матриц; программируемой матричной логики; базовых матричных кристаллов и др. [1, 2]. В основе синтеза таких устройств лежат методы абстрактного (логического) и структурного синтеза конечных автоматов. Это обстоятельство делает необходимым приобретения знаний и навыков применения этих методов для анализа и синтеза цифровых устройств управления объектами и устройствами сервиса студентами специальностей "Бытовая радиоэлектронная аппаратура", а также "Сервис компьютерной и микропроцессорной техники" и "Сервис спутникового и кабельного телевидения".

1. Основные понятия теории автоматов

Термин *автомат* (от греческого αὐτόματος, - самодействующий) как правило, используется в двух аспектах. С одной стороны, автомат — устройство, выполняющее некоторые функции без непосредственного участия человека. В этом смысле мы говорим, что ЭВМ — автомат, так как после загрузки программы и исходных данных ЭВМ решает заданную задачу без участия человека. С другой стороны, термин «автомат» как математическое понятие обозначает математическую модель реальных технических автоматов [3].

Цифровые устройства с точки зрения теории дискретных устройств представляются дискретными автоматами. Рассмотрим основные понятия, относящиеся к дискретным автоматам — математическим моделям цифровых устройств.

Автоматом конечным (дискретным) называется математическая модель устройства, обладающего конечным числом устойчивых состояний и осуществляющего преобразование дискретной информации

Так как дискретная информация может быть представлена в той или иной системе счисления (представлена цифрами), то часто дискретные автоматы называют *цифровыми автоматами*.

С точки зрения взаимодействия с внешней средой дискретный автомат можно представить в виде "черного ящика", который обменивается информацией с внешней средой через входные и выходные каналы (рис.1).

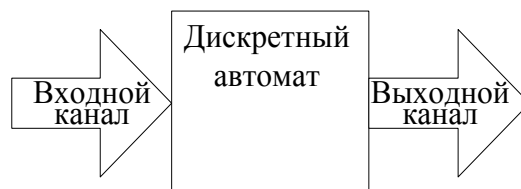


Рис. 1 Структурная схема дискретного автомата

Реальные устройства работают по *тактам*. На каждом из тактов по окончании переходного процесса устанавливается определенное внутреннее состояние автомата и соответствующие этому состоянию выходные сигналы.

В теории автоматов вводят допущения, которые упрощают рассмотрение процессов, происходящих в реальных устройствах:

- входные сигналы не имеют длительности;
- выходные сигналы изменяются мгновенно;
- переход в новое *внутреннее состояние* происходит скачкообразное, переходные процессы не учитываются
- с мгновенным изменением состояния автомата связано и мгновенное изменение выходного сигнала.

Эти допущения означают, что все изменения в дискретных автоматах происходят в дискретные моменты времени — $t = 0, 1, 2, \dots, i$, нумерующие границы между тактами. При этом говорят, что функционирование автомата свя-

зано с последовательным изменением его состояний в дискретные интервалы времени конечной длительности, называемыми *интервалами дискретности*.

Понятие *состояния автомата* используется для описания систем, выходы которых зависят не только от входных сигналов в данный момент времени, но и от некоторой предыстории, т. е. сигналов, которые поступили на входы системы ранее. Состояние автомата соответствует некоторой памяти о прошлом, позволяя устранить время как явную переменную и выразить выходные сигналы как функцию состояний и входных сигналов.

Входная и выходная информация в теории автоматов представляется сигналами, имеющими несколько уровней. Каждому уровню сопоставляется некоторый символ, который в зависимости от смысла перерабатываемой информации, может интерпретироваться как цифра, буква, признак. Промежуточные значения реальных сигналов заменяются одним из ближайших уровней. Такая информация является дискретной.

При описании цифровых автоматов внешние воздействия, выходная реакция и состояния рассматриваются как буквы трех алфавитов, называемых соответственно входным алфавитом $X = \{x_1, x_2, \dots, x_n\}$, выходным алфавитом $Y = \{y_1, y_2, \dots, y_k\}$ и алфавитом состояния $Q = \{q_1, q_2, \dots, q_s\}$.

Примером входного и выходного алфавитов является бинарный (двоичный) алфавит, содержащий только две буквы - $Q = \{0, 1\}$, $Y = \{0, 1\}$, $X = \{0, 1\}$. Различные комбинации букв алфавитов можно интерпретировать как входные (выходные) слова или дискретные сигналы и, соответственно, слова состояния. Закон функционирования автомата может быть задан двумя функциями – функцией переходов $\delta: Q \times X \rightarrow Q$, связывающей пару "входной сигнал – состояние" и отображающей множество $Q \times X$ в Q , а также функцией выходов $\lambda: Q \times X \rightarrow Y$, связывающей пару "выходная буква – состояние" и отображающей множество $Q \times X$ в Y , соответственно.

Таким образом, характерной особенностью математического описания конечного автомата является *дискретность математических моделей*, а также *конечность областей значений входных, выходных сигналов и состояний*.

Общая теория автоматов при сделанных выше допущениях разбивается на две большие части — абстрактную теорию автоматов и структурную теорию автоматов. Различие между ними заключается в том, что в абстрактной теории мы отвлекаемся от структуры, как самого автомата, так и его входных и выходных сигналов. Не интересуясь способом построения автомата, абстрактная теория изучает лишь те переходы, которые претерпевает автомат под воздействием входных сигналов, и те выходные сигналы, которые он при этом выдает. Абстрактная теория автоматов близка теории алгоритмов, является ее дальнейшей детализацией.

В общем случае математической моделью цифрового устройства является дискретный автомат, заданный совокупностью шести объектов:

- 1) конечное множество X входных сигналов:

$$X = \{x_1(t), x_2(t), \dots, x_n(t)\}$$

2) конечное множество Y выходных сигналов:

$$Y = \{y_1(t), y_2(t), \dots, y_k(t)\}$$

3) множество Q состояний автомата:

$$Q = \{q_1(t), q_2(t), \dots, q_s(t)\}$$

4) начальное состояние автомата q_0 , как элемент множества Q :

$$q_0(t) \in Q;$$

5) функция $\delta(q, x)$ (функция перехода автомата из одного состояния в другое);

6) функция $\lambda(q, x)$ (функция выходов автомата).

В начальный момент времени t_0 автомат находится в начальном состоянии q_0 . В дискретные моменты времени t на автомат поступают входные сигналы $x(t)$, автомат последовательно меняет внутренние состояния и генерирует соответствующие выходные сигналы $y(t)$.

2. Классификация дискретных автоматов

По способу формирования интервалов дискретности дискретные автоматы делят на два класса – *синхронные* автоматы и *асинхронные*. В *синхронных автоматах* моменты времени, в которых фиксируются изменения состояний автомата, задаются специальным устройством – генератором синхросигналов, выдающим импульсы через равные промежутки времени (постоянный интервал дискретности). В *асинхронных автоматах* моменты перехода автомата из одного состояния в другое заранее не определены и зависят от каких-то событий, например, от темпа поступления входных сигналов. В таких автоматах интервал дискретности является переменным.

По объему памяти дискретные автоматы подразделяются на:

- 1) дискретные автоматы с большим объемом памяти;
- 2) дискретные автоматы с малым объемом памяти;
- 3) дискретные автоматы с одним состоянием (комбинационные или функциональные автоматы).

К дискретным автоматам с большим объемом памяти относятся модели различных *запоминающих устройств*.

К дискретным автоматам цифровым устройствам с малым объемом памяти относятся модели триггеров, счетчиков, регистров, распределителей импульсов, управляющих автоматов и др.

Комбинационным автоматом называется дискретный автомат, имеющий одно внутреннее состояние, определенное типами и характером соединения элементов схемы автомата.

Это частный случай ДА. Для комбинационных автоматов характерна однозначная зависимость между входными и выходными сигналами в каждый момент времени.

К комбинационным автоматам относятся преобразователи кодов, шифраторы, дешифраторы, мультиплексоры, комбинационные сумматоры и др.

В особый класс выделяют *бесконечные автоматы*, у которых множества Q , X , и Y могут быть бесконечными. Такие автоматы являются моделями вычислительных машин, поведение которых определяется не только внутренней структурой, но и записанной программой.

По характеру функционирования автоматы могут быть *детерминированными* и *вероятностными*. Функции переходов и выходов вероятностных автоматов, а также их структура зависят от случайного выбора.

По формированию выходного сигнала дискретные автоматы подразделяются на *автоматы Мили* и *автоматы Мура*.

Дискретный автомат, для которого выходной сигнал зависит от входного сигнала и состояния, то есть $\lambda: Q \times X \rightarrow Y$, называется *автоматом Мили*.

Дискретный автомат, для которого выходной сигнал зависит только от состояния и не зависит от входного сигнала, то есть $\lambda: Q \rightarrow Y$, называется *автоматом Мура*.

Кроме того, в зависимости от способа определения выходного сигнала в синхронных автоматах существуют две возможности :

1) выходной сигнал $y(t)$ однозначно определяется входным сигналом $x(t)$ и состоянием $q(t-1)$ автомата в предшествующий момент – это автомат *первого рода*;

2) выходной сигнал $y(t)$ однозначно определяется входным сигналом $x(t)$ и состоянием $q(t)$ в данный момент времени – это автомат *второго рода*.

Следовательно, закон функционирования дискретного синхронного автомата может быть задан следующим образом:

для автомата первого рода

$$\begin{cases} q(t) = \delta(q(t-1), x(t)), \\ y(t) = \lambda(q(t-1), x(t)), \quad t = 1, 2, \dots; \end{cases} \quad (1)$$

для автомата второго рода

$$\begin{cases} q(t) = \delta(q(t-1), x(t)), \\ y(t) = \lambda(q(t), x(t)), \quad t = 1, 2, \dots; \end{cases} \quad (2)$$

Между автоматами первого и второго родов существует однозначное соответствие. Предположим, что произвольный автомат S второго рода задан уравнениями (2). Функцию выхода можно представить в следующем виде

$$\lambda(q(t), x(t)) = \lambda(\delta(q(t-1), x(t)), x(t)) = \lambda_1(q(t-1), x(t)).$$

Тогда выходной сигнал (2) $y(t)$ можно представить в виде

$$y(t) = \lambda_1(q(t - I), x(t)) \quad (3)$$

Получили новый автомат R первого рода, заданный той же функцией перехода $\delta(q, x)$, что и автомат S , а также функцией выходов $\lambda_1(q, x)$.

Таким образом, для каждого автомата S второго рода существует эквивалентный ему абстрактный автомат R первого рода, функция выходов которого получается в результате подстановки функции переходов автомата S в его сдвинутую функцию выходов:

$$\lambda_1(q, x) = \lambda(\delta(q, x), x).$$

Между моделями Мили и Мура существует соответствие, позволяющее преобразовать закон функционирования одного из них в другой или обратно. Такое преобразование порождает пару описаний законов функционирования, эквивалентных в том смысле, что им соответствует одинаковая зависимость между входной X и выходной Y последовательностями.

Совмещенная модель автомата (C-автомат) [4]. Абстрактный C-автомат — математическая модель дискретного устройства, для которого заданы следующие параметры:

$Q = \{q_1(t), q_2(t), \dots, q_s(t)\}$ - множество состояний;

$X = \{x_1(t), x_2(t), \dots, x_n(t)\}$ - входной алфавит;

$Y = \{y_1(t), y_2(t), \dots, y_k(t)\}$ - выходной алфавит автомата Мили;

$U = \{u_1(t), u_2(t), \dots, u_m(t)\}$ - выходной алфавит автомата Мура;

$\delta: Q \times X \rightarrow Q$ - функция переходов, реализующая отображение $D_\delta \subseteq Q \times X$ в Q ;

$\lambda_1: Q \times X \rightarrow Y$ - функция выходов, реализующая отображение $D_{\lambda_1} \subseteq Q \times X$ в Y ;

$\lambda_2: Q \rightarrow U$ — функция выходов, реализующая отображение $D_{\lambda_2} \subseteq Q$ на U ;

$q_0 \in Q$ — начальное состояние автомата.

Абстрактный C-автомат можно представить в виде устройства с одним входом, на который поступают сигналы из входного алфавита X , и двумя выходами, на которых появляются сигналы из выходных алфавитов Y и U (рис. 2).

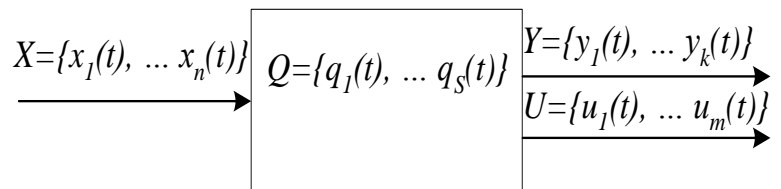


Рис. 2. Совмещенная модель автомата с одним входом и двумя выходами

Отличие C-автомата от моделей

Мили и Мура состоит в том, что он одновременно реализует две функции выходов λ_1 и λ_2 , каждая из которых характерна для этих моделей в отдельности. Этот автомат можно описать следующей системой уравнений:

$$\begin{cases} q(t+1) = \delta(q(t), x(t)), \\ y(t) = \lambda_1(q(t), x(t)), \\ u(t) = \lambda_2(q(t)). \end{cases} \quad (4)$$

Выходной сигнал $u = \lambda_2(q_i)$, $i = \overline{1, s}$, выделяется все время, пока автомат находится в состоянии q_i . Выходной сигнал $y = \lambda_1(q_i, x_j)$ выдается во время действия входного сигнала x_j при нахождении автомата в состоянии q_i .

От С-автомата легко перейти к автоматам Мили или Мура (с учетом возможных сдвигов во времени на один такт), так же как возможна трансформация автомата Мили в автомат Мура.

Таким образом, дискретный автомат полностью описывается множествами входного и выходного алфавитов и алфавитом состояний, функциями переходов и выходов $(Q, X, Y, \delta, \lambda)$. В каждый из моментов дискретного времени i автомат, находящийся в определенном состоянии, воспринимает входной сигнал - букву входного алфавита $x(i) \in X$, выдает выходной сигнал $y(i) \in Y$ - букву выходного алфавита, определяемую функцией выходов $y(i) = \lambda(q(i), x(i))$, и переходит в новое состояние, определяемое функцией переходов $q(i+1) = \delta(q(i), x(i))$.

С точки зрения структурной теории автоматов С-автомат имеет обобщенную структурную схему, представленную на рис. 3.

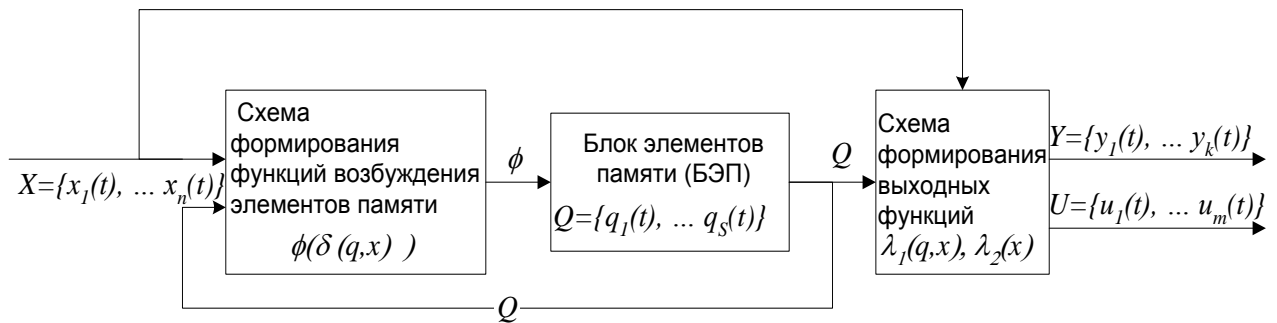


Рис. 3. Структура С-автомата

Автомат содержит три блока – две комбинационные схемы и блок элементов памяти.

Первая комбинационная схема формирует сигналы возбуждения элементов памяти, что является отображением функции переходов $q(t+1) = \delta(q(t), x(t))$ на конкретную структуру дискретного автомата.

Вторая схема формирует выходные функции автомата $y(t) = \lambda_1(q(t), x(t))$ и $u(t) = \lambda_2(q(t))$, как выходные логические функции комбинационного автомата, синтезированные с учетом используемых логических элементов.

Состояние автомата определяется состоянием элементов памяти (триггеров различных типов). Изменению состояния автомата соответствует изменение со-

стояний триггеров. Переключение триггеров происходит под воздействием функций возбуждения таким образом, что состояния автомата меняются в соответствии с функцией переходов $q(t+1) = \delta(q(t), x(t))$.

3. Языки описания цифровых автоматов

В зависимости от способа задания функцией переходов и выходов (δ и λ) в настоящее время выделяют два класса языков — *начальные языки* и *стандартные*, или *автоматные языки*. В начальных языках автомат описывается на поведенческом уровне, т. е. функции переходов и выходов обычно в явном виде не заданы. Поведение автомата описывается в терминах входных и выходных последовательностей, реализуемых операторов (отображений) или выходных управляющих последовательностей сигналов. В автоматных языках поведение автомата задается путем явного задания функций переходов и выходов.

3.1 Начальные языки описания

Среди начальных языков следует выделить *автоматное отображение*, т.е. соответствие между последовательностями входных и выходных сигналов, *язык регулярных выражений алгебры событий*, *язык логических схем алгоритмов*, *язык граф-схем алгоритмов*.

Рассмотрим кратко перечисленные начальные языки описания цифровых автоматов.

Автоматное отображение [5] реализуется посредством *автоматного оператора* T , перерабатывающего слова во входном алфавите X в слова в выходном алфавите Y

$$y = Tx; \quad x = x(1)...x(n)...; \quad x = x(1)...x(n)....$$

Автоматный оператор определяется рекуррентным соотношением:

$$\begin{aligned} q(1) &= q_0; \\ q(t+1) &= \delta(q(t), x(t)); \\ y(t) &= \lambda(q(t), x(t)). \end{aligned}$$

Если автоматный оператор T определен на всем множестве входных слов X , то соответствующий цифровой автомат является всюду определенным, а если — на некотором подмножестве входных слов X , то цифровой аппарат — *частичный*.

Например, функционирование автомата задано в виде таблицы соответствия (табл. 1), отражающей вход-выходную последовательность:

$$x_0, x_1, x_3, x_2, x_1, x_3 - y_0, y_1, y_1, y_0, y_0, y_0. \quad (5)$$

В течение шести тактов на автомат поступает последовательность входных сигналов $x_0, x_1, x_3, x_2, x_1, x_3$. Реакцией автомата на входную последовательность должна быть выходная последовательность $y_0, y_1, y_1, y_0, y_0, y_0$, формируемая на его выходе в этом же интервале времени.

Таблица 1 – Соответствие входных и выходных сигналов

	Интервалы дискретности					
	1	2	3	4	5	6
Вход. сигналы X	x_0	x_1	x_3	x_2	x_1	x_3
Выход. сигналы Y	y_0	y_1	y_1	y_0	y_0	y_0

Язык регулярных выражений алгебры событий [6]. Для заданного конечного множества входных букв $X = \{x_1, x_2, \dots, x_n\}$ регулярное выражение задается следующим образом. Для построения языка регулярных выражений используются три операции над событиями (две бинарные и одна унарная):

- 1) $A \vee B$ — объединение (дизъюнкция);
- 2) AB — умножение (конъюнкция);
- 3) $\{A\}$ — итерация (обозначается также A^*).

Выражение, построенное из букв алфавита X и из символов операций объединения, умножения и итерации с использованием соответствующим образом круглых скобок, называется регулярным выражением в алфавите X . Всякое регулярное выражение R определяет некоторое событие S (S получается в результате выполнения всех операций, входящих в выражение R). События, определяемые таким образом, называются регулярными событиями над алфавитом X . Другими словами, *регулярным событием* называется событие, полученное из элементарных событий (однобуквенных слов x_i), применением конечного числа раз операций дизъюнкции, умножения и итерации. Например, в алфавите из трех букв x, y, z регулярное выражение $x\{x \vee y \vee z\}(y \vee z)$ задает регулярное событие, состоящее из всех слов, которые начинаются буквой x и заканчиваются буквой y или z . Регулярные события, и только они, представимы в конечных автоматах.

Пусть необходимо описать автомат, выдающий сигнал ω_1 , всякий раз, когда происходит изменение входной буквы с x_1 на x_2 . Другими словами, сигнал ω_1 , должен выдаваться в ответ на любые входные последовательности, кончающиеся последовательностью x_1x_2 . Фраза «любые входные последовательности» формализуется всеобщим, или универсальным, событием, состоящим из всех возможных слов в алфавите x_1x_2 . Такое событие записывается как $\{x_1 \vee x_2\}$. Тогда событие S_1 , в ответ на которое должен выдаваться сигнал ω_1 , будет описываться регулярным выражением: $S_1/\omega_1 = \{x_1 \vee x_2\}x_1x_2$.

Язык логических схем алгоритмов [7]. В 50-х г. Прошлого века советский математик Янов Ю.И. предложил записывать алгоритмы в виде конечной строчки, состоящей из символов операторов, логических условий и верхних и нижних стрелок, которым приписаны натуральные числа.

Порядок выполнения операций в автомате определяется *программой* (или *микропрограммой*), представляющей собой совокупность микроопераций и логических условий.

Под *микрооперацией* обычно понимается элементарный процесс переработки информации в одной из частей автомата, происходящий за время такта работы автомата. При этом устройство управления вырабатывает управляющие сигналы, которые обозначим символом $V = \{v_1, v_2, \dots, v_n\}$. Ход выполнения микроопераций может нарушаться в зависимости от условий, задаваемых множеством $Z = \{z_1, z_2, \dots, z_k\}$.

Запись алгоритма, выполненная с учетом вышеизложенного, называется логической схемой алгоритма (ЛСА).

Логические схемы алгоритмов удовлетворяют следующим условиям:

- 1) содержат один начальный (v_n) и один конечный оператор (v_k);
- 2) перед оператором v_n и после оператора v_k стрелок быть не должно;
- 3) вслед за каждым логическим условием всегда стоит верхняя стрелка;
- 4) не существует двух одинаковых (с одинаковыми цифрами) нижних стрелок;
- 5) для каждой нижней стрелки должна быть, по крайней мере, одна соответствующая ей (с одинаковой цифрой) верхняя стрелка;
- 6) для каждой верхней стрелки должна быть точно одна соответствующая ей (с одинаковой цифрой) нижняя стрелка.

Рассмотрим следующую логическую схему алгоритма:

$$v_n z_1 \uparrow^1 v_1 \downarrow^1 z_2 \uparrow^2 v_2 v_3 z_3 \uparrow^2 v_4 \downarrow^2 v_5 v_k \quad (6)$$

Эта схема имеет операторы начала и конца (v_n и v_k), пять операторов ($v_1 \dots v_5$) и три логических условия (z_1, z_2, z_3). Начальному оператору соответствует некоторое начальное состояние автомата, при котором никакие микрооперации не выполняются. Если в начальном состоянии на первый вход устройства придет сигнал, равный единице ($z_1=1$), то устройство перейдет в новое состояние, в котором выполняется оператор v_1 – первый справа после логического условия z_1 , а затем проверяется логическое условие z_2 . Если же $z_1=0$, то выходим по верхней стрелке \uparrow^1 и входим по нижней стрелке с той же цифрой \downarrow^1 на проверку логического условия z_2 , минуя выполнение оператора v_1 . Если $z_2=0$, то переходим к выполнению оператора v_5 . Если же $z_2=1$, то выполняются операторы v_2 и v_3 и проверяется оператор z_3 ; если $z_3=0$, то оператор v_4 не выполняется, происходит переход к оператору v_5 . После выполнения оператора v_5 происходит переход к конечному оператору, т. е. работа дискретного устройства заканчивается.

Граф-схемы алгоритмов [8]. Другой разновидностью языка, позволяющей описывать логические схемы алгоритмов, является язык *граф-схем алгоритмов*. Граф-схема алгоритма — ориентированный связный граф, содержащий одну начальную вершину, произвольное число условных и операторных вершин и одну конечную вершину.

Конечная, операторная и условная вершины имеют по одному входу, начальная вершина входов не имеет. Начальная и операторная вершины имеют по одному выходу, конечная вершина выходов не имеет, условная вершина имеет два выхода, помеченных символами 1 и 0. Граф-схема алгоритма удовлетворяет следующим условиям:

- 1) входы и выходы вершин соединяются друг с другом с помощью дуг, направленных всегда от выхода к входу;
- 2) каждый выход соединен только с одним входом;
- 3) любой вход соединяется, по крайней мере, с одним выходом;
- 4) любая вершина графа лежит, по крайней мере, на одном пути из начальной вершины к конечной;
- 5) в каждой условной вершине записывается один из элементов множества логических условий $Z = \{z_1, z_2, \dots, z_k\}$, причем возможна запись одинаковых элементов множества Z в различные условные вершины;

6) в каждой операторной вершине записывается один из элементов множества операторов $V = \{v_1, v_2, \dots, v_n\}$, причем возможна запись одинаковых элементов множества V в различные операторные вершины.

На рис. 3 представлена граф-схема алгоритма, описанная выражением (6). Язык ГСА используется очень часто при описании алгоритмов функционирования как самого цифрового автомата, так и программ, выполняющих то или иное действие.

При проектировании различных устройств ЭВМ обычно используются содержательные граф-схемы алгоритмов, которые описывают не только формальные элементы, но и логические условия и микрооперации в содержательных терминах. Основные правила выполнения содержательных граф-схем алгоритмов и программ представлены в соответствующих ГОСТах. В качестве иллюстрации на рис. 4 пред-

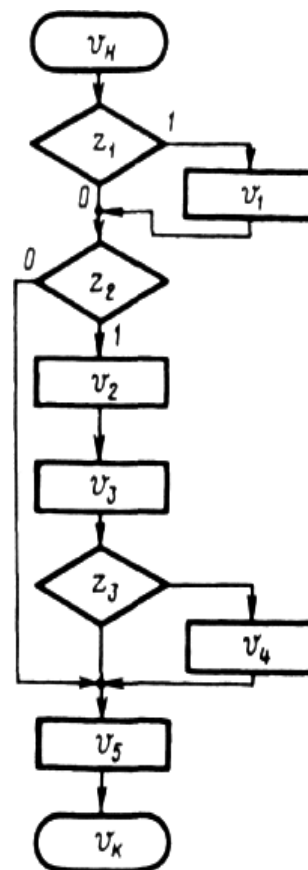


Рис. 3. Граф-схема алгоритма

ставлена содержательная граф-схема алгоритма операции сложения чисел, представленных в форме с фиксированной запятой.

3.2 Автоматные языки описания

Среди автоматных языков наиболее распространены таблицы, графы и матрицы переходов и выходов.

Таблица переходов (выходов) представляет собой таблицу с двойным входом, строки которой нумерованы входными буквами, а столбцы — состояниями. На пересечении указывается состояние, в которое переходит автомат (в таблице переходов) или выходной сигнал, выдаваемый им (в таблице выходов).

Описание работы автомата Мили таблицами переходов и выходов иллюстрируется на примере автомата S_1 , заданного множествами Q, X и Y , а также функцией перехода — $\delta: Q \times X \rightarrow Q$ и функцией выхода — $\lambda: Q \times X \rightarrow Y$ [4] (табл. 2 и табл. 3).

Таблица 2 - Таблица переходов автомата S_1 ($\delta: Q \times X \rightarrow Q$)

	q_1	q_2	q_3
x_1	q_2	q_3	q_2
x_2	q_3	q_2	q_1

Таблица 3 - Таблица выходов автомата S_1 ($\lambda: Q \times X \rightarrow Y$)

	q_1	q_2	q_3
x_1	y_1	y_3	y_3
x_2	y_2	y_1	y_1

На пересечении столбца q_m и строки x_j в таблице переходов ставится состояние $q_s = \delta(q_m, x_j)$, в которое автомат переходит из состояния q_m под действием сигнала x_j , а в таблице выходов — соответствующий этому переходу выходной сигнал $y_g = \lambda(q_m, x_j)$.

Иногда при задании автоматов Мили используют одну совмещенную таблицу переходов и выходов, в которой на пересечении столбца q_m и строки x_j записываются следующее состояние и выдаваемый выходной сигнал в виде q_s / y_g . Так, для автомата S_1 совмещенная таблица представлена таблицей 4.

В автомате Мура выходной сигнал зависит только от состояния, автомат Мура задается одной отмеченной таблицей переходов, в которой каждому ее столбцу приписан кроме состояния q_m еще и выходной сигнал $y_g = \lambda(q_m)$, соответствующий этому состоянию. Пример табличного описания автомата Мура S_2 представлен в таблице 5.

Для частичных автоматов, у которых функции δ или λ , определены не для всех пар $(q_m, x_j) \in Q \times X$, на месте неопределенных состояний и выходных сигналов ставится прочерк.

Граф автомата — ориентированный граф, вершины которого соответ-

ствуют состояниям, а дуги — переходам между ними. Две вершины графа автомата q_m и q_s (исходное состояние и состояние перехода) соединяются дугой, направленной от q_m к q_s , если в автомате имеется переход из q_m в q_s , т. е. если

$q_s = \delta(q_m, x_j)$ при некотором $x \in X$. Дуге (q_m, q_s) графа автомата приписывается входной сигнал x_j и выходной сигнал $y_g = \lambda(q_m, x_j)$.

Таблица 4 – Совмещенная таблица переходов и выходов автомата Мили S_1
 $\delta: Q \times X \rightarrow Q$,
 $\lambda: Q \times X \rightarrow Y$

	q_1	q_2	q_3
x_1	q_2/y_1	q_3/y_3	q_2/y_3
x_2	q_3/y_2	q_2/y_1	q_1/y_1

Таблица 5 – Совмещенная таблица переходов и выходов автомата Мура S_2
 $\delta: Q \times X \rightarrow Q$, $\lambda: Q \rightarrow Y$

	y_1	y_1	y_3	y_2	y_3
	q_1	q_2	q_3	q_4	q_5
x_1	q_2	q_5	q_5	q_3	q_3
x_2	q_4	q_2	q_2	q_1	q_1

При описании автомата Мура в виде графа выходной сигнал $y_g = \lambda(q_m)$ записывается внутри вершины q_m или рядом с ней. На рис. 5, а, б приведены графы автоматов S_1 и S_2 , описанных ранее табличным способом.

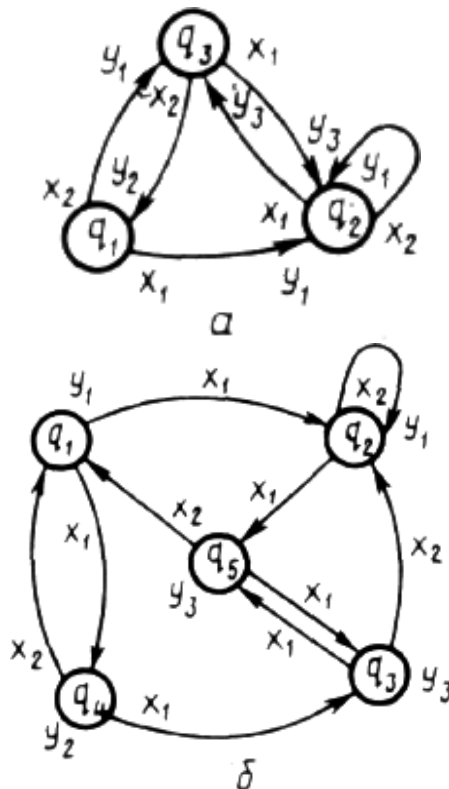


Рис. 5. Графы автоматов S_1 и S_2

Любой автомат может быть задан с помощью графа, но не всякий граф в алфавитах Q , X , Y задает автомат. В графе автомата не должно существовать двух дуг с одинаковыми входными сигналами, выходящих из одной и той же вершины (условие однозначности).

Иногда применяется способ задания автомата с помощью матрицы переходов и выходов. Строки и столбцы таблицы отмечены состояниями. Если существует переход из состояния q_m под действием входного сигнала x_j в состояние q_s с выдачей входного сигнала y_i , то на пересечении строки q_m и столбца q_s записывается пара x_j / y_i .

Для автомата Мура используется матрица, столбцы которой отмечены выходными сигналами y_i , а на пересечении ее строк и столбцов указываются только входные сигналы x_j .

Ниже приведены матрицы переходов и выходов для рассмотренных ранее автоматов S_1 и S_2 :

$$M_{S_1} = \begin{matrix} & \begin{matrix} q_1 & q_2 & q_3 \end{matrix} \\ \begin{matrix} q_1 \\ q_2 \\ q_3 \end{matrix} & \begin{bmatrix} 0 & x_1 / y_1 & x_1 / y_2 \\ 0 & x_2 / y_1 & x_1 / y_3 \\ x_2 / y_1 & x_1 / y_3 & 0 \end{bmatrix} \end{matrix},$$

$$M_{S_2} = \begin{matrix} & \begin{matrix} y_1 & y_1 & y_3 & y_2 & y_3 \end{matrix} \\ \begin{matrix} q_1 \\ q_2 \\ q_3 \\ q_4 \\ q_5 \end{matrix} & \begin{bmatrix} 0 & x_1 & 0 & x_2 & 0 \\ 0 & x_2 & 0 & 0 & x_1 \\ 0 & x_2 & 0 & 0 & x_1 \\ x_2 & 0 & x_1 & 0 & 0 \\ x_2 & 0 & x_1 & 0 & 0 \end{bmatrix} \end{matrix}$$

Рассмотрим пример составления графа автомата. Пусть задано некоторое устройство в виде вход-выходной последовательности (табл. 6).

В таблице 6 три последние набора не имеют выходного значения, т. е. выходы не определены.

Таблица 6 – Вход-выходная последовательность кодированных сигналов

Состояния	Входы			Выходы		
	x_1	x_2	x_3	y_1	y_2	y_3
q_0	0	0	0	0	0	0
q_1	0	0	1	1	0	0
q_2	0	1	0	0	1	0
q_3	0	1	1	1	1	0
	1	0	0	1	1	1*
	1	0	1	—	—	—
	1	1	0	—	—	—
	1	1	1	—	—	—

Рассмотрим последовательно входные наборы (рис. 6). В правую колонку запишем состояния, соответствующие парам входной - выходной сигналы.

1) $x_1 x_2 x_3 = 000$. Этому набору соответствует начальное состояние q_0 , кото-

рое не должно изменяться, что отображается дугой, выходящей и входящей в q_0 .

2) $x_1x_2x_3 = 001$. Состояние устройства меняется на q_1 . Значит, проводится дуга от состояния q_0 к состоянию q_1 и помечается набором 001 .

3) $x_1x_2x_3 = 010$. Новое состояние обозначим q_2 , для чего соединим q_0 и q_2 дугой и обозначим ее набором 010 .

4) $x_1x_2x_3 = 011$. Переход в состояние q_3 , которое также отличается от всех предыдущих. Соединим q_0 и q_3 дугой, отмеченной набором 011 .

5) $x_1x_2x_3 = 100$. В таблице этому входному набору соответствуют два набора: либо 111 , либо 000 , что помечено звездочкой *. Для определенности можно уточнить, что выбран набор 000 , т. е. возврат в начальное состояние. Значит, надо отметить дугу, входящую и выходящую из q_0 еще набором 100 .

Аналогичным образом поступаем с остальными внутренними состояниями q_1 , q_2 , q_3 , в результате чего получаем окончательный граф переходов для рассматриваемого автомата (рис. 6).

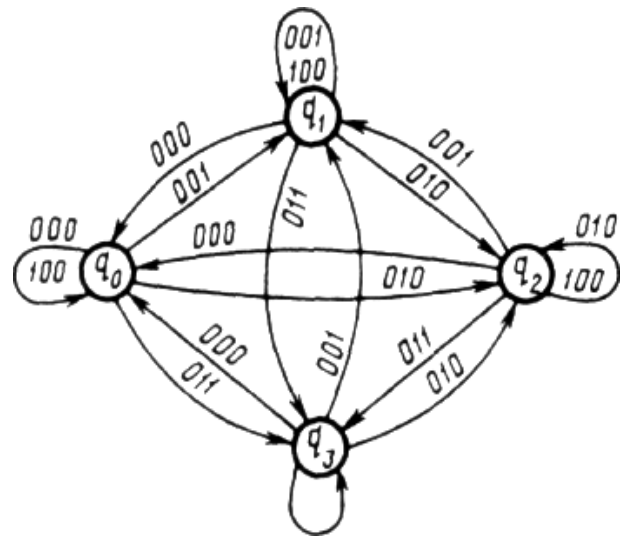


Рис. 6 Граф переходов автомата

На основании графа автомата можно составить таблицу переходов (табл. 7) или таблицу выходов.

Таблица 7 – Таблица переходов автомата

Состояния	Входы				
	000	001	010	011	100
q_0	q_0	q_1	q_2	q_3	q_0
q_1	q_0	q_1	q_2	q_3	q_1
q_2	q_0	q_1	q_2	q_3	q_2
q_3	q_0	q_1	q_2	q_3	q_3

Состояние автомата, вершина графа для которого имеет только исходящие дуги, но не имеет входящих дуг, называется *переходящим*. В такое состояние попасть нельзя, из него можно только выйти.

Состояние автомата называется *тупиковым*, если соответствующая вершина графа не содержит исходящих дуг, но имеет хотя бы одну входящую дугу.

Изолированным состоянием называется такое состояние, которому соответствует вершина графа, не имеющая как входящих, так и исходящих дуг.

Таким образом, с помощью графов абстрактная модель автомата записывается в виде некоторого пространственного изображения, которое помогает

при решении задач анализа или синтеза.

Любое разбиение множества состояний автомата на ряд подмножеств, объединяющих некоторые состояния, приводит к понятию *подавтомата*.

3.3 Описание комбинационных автоматов

Комбинационные автоматы задаются тремя элементами: входным алфавитом $X = \{x_1, x_2, \dots, x_n\}$, выходным алфавитом $Y = \{y_1, y_2, \dots, y_k\}$, функцией выходов, которая, учитывая единственность состояния комбинационного автомата, вырождается в однозначную зависимость между входными и выходными переменными $\lambda: X \rightarrow Y$ или $y = \lambda(x)$. Такая зависимость полностью определяется логическими функциями.

Функция $\lambda(x_1, \dots, x_n)$ называется логической (булевой), если аргументы x_1, \dots, x_n и значения функции принимают только два значения – 0 или 1.

С п о с о б ы з а д а н и я логических функций;

- Словесный.
- Табличный. Таблицы истинности.
- Цифровой. Наборы логических переменных, на которых значения логической функции принимают единичные (нулевые) значения представляются числам в двоичной или в десятичной системах счисления.
- Аналитический. Основан на представлении логической функции в виде формул, связывающих между собой переменные с помощью элементарных операций и скобок.



Рис. 7. Алгоритм синтеза комбинационных автоматов

4. Алгоритмы синтеза дискретных автоматов

Задача синтеза сводится к построению оптимальной электрической схемы функционально устойчивого устройства по заданным условиям функционирования с учетом ограничений на используемые элементы памяти и базис логических элементов.

В качестве критерия качества (оптимальности) часто выбирается минимальная сложность функциональной электрической схемы (ФЭС), определяемая числом элементов памяти (ЭП), логических элементов (ЛЭ) и связей между ними.

4.1 Синтез комбинационных автоматов

Задача этапа логического синтеза заключается в построении функциональной электрической схемы устройства, реализующего заданные условия функционирования, оптимальной с точки зрения количества логических элементов и с учетом ограничений на их тип.

Алгоритм логического синтеза комбинационных автоматов включает последовательное решение частных задач по преобразованию исходной информации. Алгоритм синтеза приведен на рис 7. В некоторых случаях отдельные задачи объединяются.

Полученная функциональная электрическая схема после решения задачи анализа по устранению сбоев, включения резервирования, технической диагностики становится исходным материалом для построения принципиальных и монтажных схем.

Условные графические обозначения элементов цифровой техники выполняются в соответствии с ГОСТ 2.743-91.

4.1.2 Функционально полные системы логических функций.

Алгоритмы получения формул в заданных базисах

Система логических функций называется *функционально-полной* или *базисом*, если любая логическая функция представима через функции этой системы.

Если функционально-полная система функций является минимальной (неизбыточной), то такая система называется *минимальным базисом*.

Доказано [9], что на множестве логических функций двух переменных можно выделить 17 различных базисов.

Наиболее распространенными являются следующие:

И, ИЛИ, НЕ (основной базис – не является минимальным);
И, НЕ;
ИЛИ, НЕ.

4.1.2.1 Алгоритм представления сложной функции в основном базисе

Заключается в последовательной замене всех формул типа ИМПЛИКАЦИЯ, РАВНОЗНАЧНОСТЬ, ЗАПРЕТ и тд. их эквивалентным представлением через операции И, ИЛИ, НЕ:

$$A \sim B = AB \vee \overline{AB}; \quad A \approx B = A\overline{B} \vee \overline{A}B; \quad A \rightarrow B = \overline{A} \vee B; \quad A \rightarrow B = \overline{\overline{A} \vee B}.$$

4.1.2.2 Алгоритм представления сложной функции в базисе И-НЕ

1. Представление логической функции в дизъюнктивной нормальной форме (ДНФ).
2. Постановка над ДНФ двойного отрицания.
3. Преобразование одного отрицания в соответствии с формулой де Моргана $\overline{A \vee B} = \overline{A} \cdot \overline{B}$.

Рассмотрим следующий пример. Требуется представить в базисе И-НЕ функцию:

$$f(x_1, x_2, x_3) = (x_1 \oplus x_2) \vee \overline{(x_1 x_2 \rightarrow x_3)}.$$

1-й этап:

$$\begin{aligned} f(x_1, x_2, x_3) &= (x_1 \oplus x_2) \vee \overline{(x_1 x_2 \rightarrow x_3)} = (x_1 \overline{x_2} \vee \overline{x_1} x_2) \vee \overline{x_1 x_2 x_3} = \\ &= (x_1 \overline{x_2} \vee \overline{x_1} x_2) \vee \overline{x_1} \vee \overline{x_2} \vee \overline{x_3} = x_1 \overline{x_2} \vee \overline{x_1} x_2 \vee \overline{x_1} \vee \overline{x_2} \vee \overline{x_3} = \overline{x_1} \vee \overline{x_2} \vee \overline{x_3}. \end{aligned}$$

2-й - 3-й этапы:

$$f(x_1, x_2, x_3) = \overline{\overline{\overline{\overline{\overline{x_1} \vee \overline{x_2} \vee \overline{x_3}}}}} = \overline{x_1 x_2 x_3}.$$

Полученное выражение соответствует логическому элементу 3И-НЕ (рис. 8).

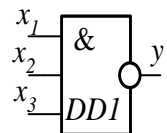


Рис. 8

4.1.2.3 Алгоритм представления сложной функции в базисе ИЛИ-НЕ

Аналогичен предыдущему алгоритму, но на первом этапе необходимо представить логическую функцию в конъюнктивной нормальной форме (КНФ).

1. Получение КНФ.
2. Постановка над КНФ двойного отрицания.
3. Преобразование одного отрицания в соответствии с формулой де Моргана $\overline{A \cdot B} = \overline{A} \vee \overline{B}$.

Рассмотрим следующий пример. Требуется представить в базисе ИЛИ-НЕ функцию

$$f(x_1, x_2, x_3) = (x_1 \oplus x_2)x_3.$$

1-й этап:

$$\begin{aligned} f(x_1, x_2, x_3) &= (x_1 \oplus x_2)x_3 = (x_1\bar{x}_2 \vee \bar{x}_1x_2)x_3 = (x_1 \vee \bar{x}_1x_2)(\bar{x}_2 \vee \bar{x}_1x_2)x_3 = \\ &= (x_1 \vee \bar{x}_1)(x_1 \vee x_2)(\bar{x}_2 \vee \bar{x}_1)(\bar{x}_2 \vee x_2)x_3 = (x_1 \vee x_2)(\bar{x}_2 \vee \bar{x}_1)x_3. \end{aligned}$$

2-й - 3-й этапы:

$$\begin{aligned} (x_1, x_2, x_3) &= \overline{\overline{(x_1 \vee x_2)(\bar{x}_2 \vee \bar{x}_1)x_3}} = \\ &= \overline{(x_1 \vee x_2) \vee (\bar{x}_2 \vee \bar{x}_1) \vee \bar{x}_3}. \end{aligned}$$

Полученному выражению соответствует схема, представленная на рис. 9.

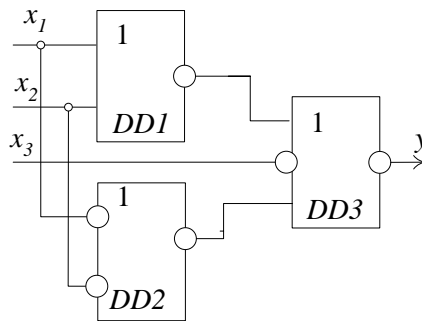


Рис. 9

4.2 Синтез схемы голосования «два из трех»

При создании высоконадежных устройств часто используется функциональная избыточность – одновременное функционирование нескольких одинаковых устройств. Так при параллельной работе трех устройств путем голосования выходных сигналов можно обеспечить безотказную работу при отказе любого одного устройства.

Постановка задачи

1 С л о в е с н о е о п и с а н и е . Синтезировать комбинационный автомат голосования трех двоичных переменных, по правилу – нуль на выходе формируется при двух или трех нулях на входе, иначе формируется единица. Базис логического элементов И-НЕ.

2. Структурная схема устройства представлена на рис. 10.

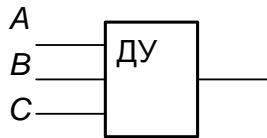


Рис. 10 Структурная схема устройства

3. В соответствии со словесным описанием строим таблицу истинности 8.).
4. Минимизируем логическую функцию картой Карно (рис. 11).

Таблица 8 – Таблица истинности

A	B	C	Y
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

	BC			
A	00	01	11	10
0	0	0	1	0
1	0	1	1	1

Рис. 11 Минимизация логической функции картой Карно

В матрице обводим три максимальных правильных контуров. Им соответствуют три простые импликанты: $y = AB \vee BC \vee AC$.

5. Осуществляем перевод в базис И-НЕ в соответствии с алгоритмом

$$y = \overline{\overline{AB} \vee \overline{BC} \vee \overline{AC}} = \overline{\overline{AB} \cdot \overline{BC} \cdot \overline{AC}}.$$

7. Вычерчиваем функциональную схему устройства (рис. 12).

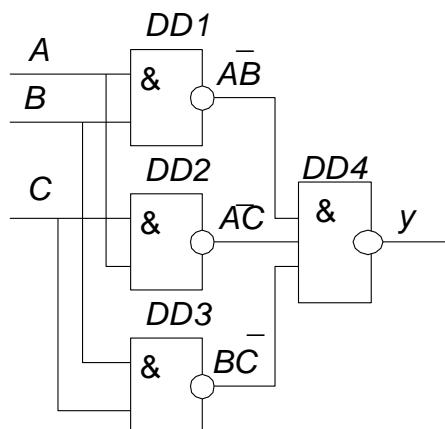


Рис. 12 Схема голосования «два из трех»

4.3 Синтез дискретных автоматов с памятью

Исходными данными для синтеза являются описание функционирования автомата одним из приведенных выше способом, а также ограничения на используемые элементы памяти и базис логических элементов.

Алгоритм синтеза включает в себя два этапа – *этап абстрактного синтеза* и *этап структурного синтеза*.

Задачей первого этапа является, во-первых, *описание* функционирования автомата, изначально представленного одним из языков начального описания, *автоматным языком* в виде таблиц переходов-выходов, и/или в виде графа автомата, и/или матрицами переходов и выходов. Другими словами, на основе начального описания автомата требуется получить его описание, представленное множествами Q, X, Y , а также функциями δ и λ .

Второй задачей абстрактного синтеза является *минимизация числа состояний* автомата. Для некоторых СБИС программируемой логики, снабженных большим количеством триггеров, необходимость решения этой задачи отпадает, так как экономия числа триггеров при построении автомата несущественна.

Основным содержанием этапа структурного синтеза является отображение автоматного описания устройства в структуру связанных необходимым образом комбинационных устройств, элементов памяти, т.е. получение функциональной схемы устройства.

В ходе структурного синтеза выполняется:

- кодирование элементов множеств Q, X, Y ;
- получение структурных таблиц переходов и выходов;
- получение таблицы возбуждения элементов памяти (учитываются ограничения на тип используемых элементов памяти);
- формирование функций возбуждения элементов памяти и выходных функций их минимизация и перевод в заданный базис элементов;
- построение схем автомата.

Кодирование состояний заключается в том, что каждому состоянию $q_i \in Q$ однозначным образом ставится в соответствие набор внутренних переменных $\gamma_1, \gamma_2, \dots, \gamma_h$. Состояния и соответствующие им коды обычно представляют в виде таблицы, которая называется таблицей кодирования состояний автомата.

Если автомат имеет s состояний, то, для того, чтобы получить однозначное соответствие, необходимо иметь не менее s различных двоичных кодов. Последнее условие можно записать в виде $s \leq 2^h$. Откуда находим

$$h = \lceil \log_2(s) \rceil, \quad (7)$$

где $\lceil a \rceil$ означает ближайшее целое, не меньшее a .

Из неравенства (1) следует, что минимальное число элементов памяти, не-

обходимое для получения однозначного кодирования, $h = \lceil \log_2(s) \rceil$.

Кодирование состояний существенно влияет на сложность комбинационной части схемы автомата. Для того чтобы упростить комбинационную схему, часто используют избыточное кодирование, выбирая h большим, чем это необходимо для получения однозначного кодирования.

При синтезе асинхронных дискретных автоматов с памятью необходимо учитывать следующие особенности переходных процессов реальных устройств:

- длительность входных сигналов намного превышают длительность изменения состояний элементов памяти, что может привести к недопустимой последовательной смене состояний автомата под воздействием одного входного сигнала;
- возникновение явления состязания элементов памяти (ЭП).

Для исключения неустойчивости перехода $q_{i+1} = \delta(q_i, x_j)$ к его конечному состоянию q_{i+1} добавляют циклический переход $q_{i+1} = \delta(q_{i+1}, x_j)$, переводящий автомат из состояния q_{i+1} в это же состояние q_{i+1} под воздействием входного сигнала x_j .

Под *состязанием элементов памяти* понимается одновременное изменение состояния нескольких элементов памяти в одном такте работы автомата при выполнении перехода из одного состояния автомата в другое. Это приводит к появлению промежуточных паразитных состояний и, соответственно, к появлению "ложных" выходных сигналов, не соответствующих функции выхода автомата.

Причинами состязания элементов памяти являются:

- конструктивные, технологические различия ЭП;
- задержки сигналов в комбинационной схеме формирования сигналов возбуждения ЭП.

Для устранения состязания ЭП необходимо, чтобы при смене состояния автомата менялось состояние только одного элемента памяти. Это может достигаться избыточным кодированием, когда смежные вершины графа автомата кодируются кодовыми комбинациями с единичным кодовым расстоянием (отличающимися только одним разрядом). В некоторых случаях для устранения состязаний и неустойчивости переходов требуется введения дополнительных вершин графа (дополнительных состояний автомата).

Кодирование состояний кажется целесообразным выполнять совместно с кодированием входных и выходных сигналов, однако такая задача оказывается весьма сложной и практически не реализуется

Для синхронных автоматов проблема состязаний элементов памяти и комбинационных схем решается путем выбора периода тактовых сигналов (длительности интервала дискретности) из условия завершения всех переходных процессов в автомате.

4.4 Пример синтеза дискретных автоматов

4.4.1 Этап абстрактного синтеза

Рассмотрим содержание и особенности отдельных этапов синтеза на примере.

Пусть требуется синтезировать асинхронный автомат Мура, начальное описание которого представлено вход-выходной последовательностью и таблицей соответствия (табл.1):

$$x_0, x_1, x_3, x_2, x_1, x_3 - y_0, y_1, y_1, y_0, y_0, y_0$$

Определяем мощность входного и выходного множеств $M_X = 4$, $M_Y = 2$.

Строим *первичную таблицу переходов-выходов* автомата Мура (табл.9). Она имеет 4 строки по числу входных сигналов и 6 столбцов. При этом считаем, что каждому такту соответствует новое состояние автомата s_i , которому ставится в соответствие выходной сигнал y_k .

Считаем, что автомат Мура перед подачей входной последовательности находится в начальном состоянии s_0 , в котором выходной сигнал не формируется. Реальное устройство может переходить в такое состояние при подключении источника питания. Отсутствие сигнала выходного алфавита на выходах устройства, когда автомат находится в начальном состоянии, обозначим символом y^* .

Таблица 9 – Первичная таблица переходов-выходов автомата Мура

X	Состояния S и выходные сигналы Y						
	y^*	y_0	y_1	y_1	y_0	y_0	y_0
	s_0	s_1	s_2	s_3	s_4	s_5	s_6
x_0	s_1	(s_1)					s_1
x_1		s_2	(s_2)		s_5	(s_5)	
x_2				s_4	(s_4)		
x_3			s_3	(s_3)		s_6	(s_6)

При подаче входной последовательности автомат последовательно переходит в шесть состояний, обозначенных $s_1...s_6$. Часть ячеек остается незаполненной. Для обеспечения устойчивой работы доопределяем автомат дополнительными переходами, в которых начальное и конечное состояния переходов совпадают (состояния в скобках). Первичный граф переходов-выходов (ГПВ) представлен на рис. 13.

Для автомата Мили первичная таблица переходов-выходов выглядит по-другому (табл. 10). Так как выходные сигналы формируются автоматами Мили только при наличии входного сигнала, то количество состояний в первичной

таблице переходов-выходов определяется количеством входных (выходных сигналов) в заданной вход-выходной последовательности. Аналогично, как и для автомата Мура, доопределяем автомат дополнительными переходами (состояния и выходные сигналы в скобках). Первичный граф переходов-выходов автомата Мили представлен на рис. 14.

Таблица 10 – Первичная таблица переходов-выходов автомата Мили

X	Состояния S и выходные сигналы Y					
	s_0	s_1	s_2	s_3	s_4	s_5
x_0	s_1 / y_0	(s_1 / y_0)				
x_1		s_2 / y_1	(s_2 / y_1)		s_5 / y_0	(s_5 / y_0)
x_2				s_4 / y_0	(s_4 / y_0)	
x_3	(s_0 / y_0)		s_3 / y_1	(s_3 / y_1)		s_0 / y_0

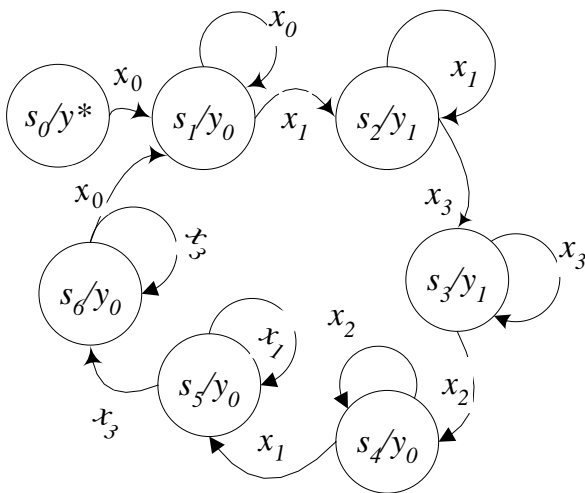


Рис. 13 Первичный граф переходов-выходов автомата Мура

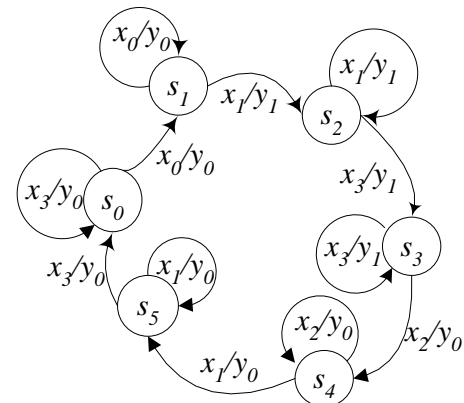


Рис. 14 Первичный граф переходов-выходов автомата Мили

4.4.2 Минимизация внутренних состояний

Задача минимизации автомата сводится к возможно максимальному уменьшению числа внутренних состояний без изменения закона их функционирования.

Решение этой задачи выполним методом Ауфенкампа и Хона, основанного на понятиях эквивалентных состояний.

Эквивалентными состояниями называются s_n и s_m такие состояния, которым, во-первых, соответствуют одинаковые выходные сигналы $y(t)$, а, во-

вторых, переход из состояний s_n и s_m под воздействием любого символа приводит к одному и тому же (или эквивалентному состоянию)

Это означает, что автомат, находящийся в этих состояниях под воздействием любой последовательности формирует одинаковые выходные последовательности.

Алгоритм оптимизации основан на разбиении множества состояний на непересекающиеся подмножества эквивалентных состояний и замена каждого подмножества эквивалентных состояний одним из его представителей.

Алгоритм минимизации:

1) методом последовательного разбиения выделяем все попарно эквивалентные состояния;

2) объединяем эквивалентные состояния в одинаковые классы $\gamma_1, \gamma_2, \dots$ и выделяем в каждом классе по одному состоянию для выполнения последующих этапов, при разбиении на классы пустые клетки во внимание не принимаются;

3) строим вторичную таблицу переходов-выходов, в которой каждый класс состояний представляем только одним эквивалентным состоянием.

Первое разбиение состояний на классы выполняем по выходным сигналам (табл. 11). Анализ таблицы 10 показывает, что в классе γ_1 состояние s_1 не является эквивалентным остальным состояниям этого класса, так как при поступлении сигнала x_1 автомат переходит из состояния s_1 в класс состояний γ_2 , в отличие от других состояний класса, переводящих автомат под действием сигнала x_1 в состояния класса γ_1 . Поэтому состояние s_1 выделяем в отдельный класс γ_3 (табл. 12).

Таблица 11 – Первое разбиение на классы состояний автомата Мура

X	Состояния S и выходные сигналы Y						
	y^*	y_0	y_0	y_0	y_0	y_1	y_1
	s_0	s_1	s_4	s_5	s_6	s_2	s_3
x_0	$s_1 (\gamma_1)$	$s_1 (\gamma_1)$			$s_1 (\gamma_1)$		
x_1		$s_2 (\gamma_2)$	$s_5 (\gamma_1)$	$s_5 (\gamma_1)$		$s_2 (\gamma_2)$	
x_2			$s_4 (\gamma_1)$				$s_4 (\gamma_1)$
x_3				$s_6 (\gamma_1)$	$s_6 (\gamma_1)$	$s_3 (\gamma_2)$	$s_3 (\gamma_2)$
Классы	γ_0	γ_1				γ_2	

На основании таблицы 12 строим вторичную таблицу переходов-выходов автомата Мура, эквивалентного исходному автомату, но содержащей только четыре состояния вместо семи (табл. 13).

Вторичный граф переходов-выходов автомата Мура представлен на рис. 15.

Таблица 12 – Второе разбиение на классы состояний автомата Мура

X	Состояния S и выходные сигналы Y						
	y^*	y_0	y_0	y_0	y_0	y_1	y_1
	s_0	s_1	s_4	s_5	s_6	s_2	s_3
x_0	$s_1 (\gamma_3)$	$s_1 (\gamma_3)$			$s_1 (\gamma_3)$		
x_1		$s_2 (\gamma_2)$	$s_5 (\gamma_1)$	$s_5 (\gamma_1)$		$s_2 (\gamma_2)$	
x_2			$s_4 (\gamma_1)$				$s_4 (\gamma_1)$
x_3				$s_6 (\gamma_1)$	$s_6 (\gamma_1)$	$s_3 (\gamma_2)$	$s_3 (\gamma_2)$
Классы	γ_0	γ_3	γ_1			γ_2	

Таблица 13 – Вторичная таблица переходов-выходов автомата Мура

X	Состояния Γ и выходные сигналы Y			
	y^*	y_0	y_0	y_1
	γ_0	γ_3	γ_1	γ_2
x_0	γ_3	γ_3	γ_3	
x_1		γ_2	γ_1	γ_2
x_2			γ_1	γ_1
x_3			γ_1	γ_2

Для автомата Мили процедура аналогична.

Будем считать, что начальным состоянием автомата Мили при включении питания является состояние s_0 . При формировании класса эквивалентности учитываем выходные сигналы в каждой строке таблицы 10. Разбиение на классы эквивалентных состояний автомата Мили представлено в таблице 14.

Таблица 14 –Разбиение на классы состояний автомата Мили

X	Состояния S и выходные сигналы Y					
	s_0	s_1	s_2	s_3	s_4	s_5
x_0	$s_1 (\gamma_1) / y_0$	$s_1 (\gamma_1) / y_0$				
x_1		$s_2 (\gamma_1) / y_1$	$s_2 (\gamma_1) / y_1$		$s_5 (\gamma_2) / y_0$	$s_5 (\gamma_2) / y_0$
x_2				$s_4 (\gamma_2) / y_0$	$s_4 (\gamma_2) / y_0$	
x_3	$s_0 (\gamma_0) / y_0$		$s_3 (\gamma_1) / y_1$	$s_3 (\gamma_1) / y_1$		$s_0 (\gamma_0) / y_0$
Классы	γ_0	γ_1			γ_2	

Таблица 16 является вторичной таблицей переходов-выходов автомата Мили.

Вторичный граф переходов-выходов автомата Мили представлен на рис. 16.

Таблица 15 – Второе разбиение на классы состояний автомата Мили

X	Состояния S и выходные сигналы Y		
	γ_0	γ_1	γ_2
x_0	γ_1 / y_0	γ_1 / y_0	
x_1		γ_1 / y_1	γ_2 / y_0
x_2		γ_2 / y_0	γ_2 / y_0
x_3	γ_0 / y_0	γ_1 / y_1	γ_0 / y_0
	α_0	α_1	α_0

Таблица 16 – Вторичная таблица переходов-выходов автомата Мили

X	Состояния A и выходные сигналы Y	
	β_0	β_1
x_0	β_1 / y_0	β_1 / y_0
x_1	β_0 / y_0	β_1 / y_1
x_2	β_0 / y_0	β_0 / y_0
x_3	β_0 / y_0	β_1 / y_1

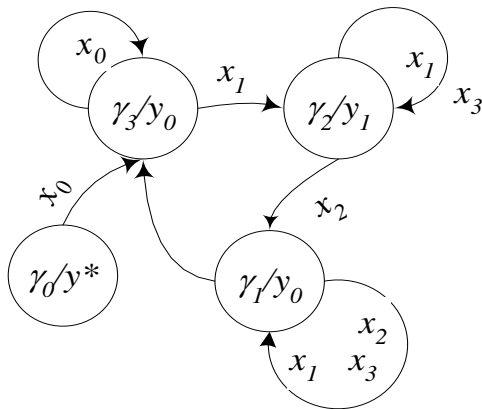


Рис. 15 Вторичный граф переходов-выходов автомата Мура

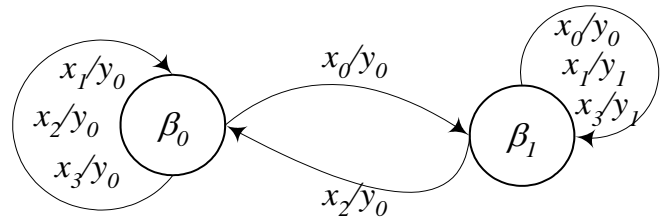


Рис. 16 Вторичный граф переходов-выходов автомата Мили

4.4.3 Структурный синтез дискретных автоматов

Структурный синтез начинается с кодирования входного и выходного алфавитов и состояний автоматов.

Количество разрядов для входных, выходных сигналов и состояний определяется в соответствии с формулой:

$$n = \log_2 (]M[), \quad (8)$$

где M – мощность множества соответствующего алфавита, $]x[$ – означает

минимальное целое, не меньшее числа x .

Для автомата Мура: $n_x = \log_2(4) = 2$, $n_y = \log_2(3) = 2$, $n_q = \log_2(4) = 2$.

Для автомата Мили: $n_x = \log_2(4) = 2$, $n_y = \log_2(2) = 1$, $n_q = \log_2(3) = 2$.

Кодирование входных и выходных сигналов автоматов представлено в таблице 17.

Таблица 17 – Кодирование входных и выходных алфавитов

Входные сигналы X	Коды		Выход. сиг- налы Мура	Коды		Выход. сиг- налы Мил	Коды
	a_1	a_0		d_1	d_0		d_0
x_0	0	0	y^*	0	0	y_0	0
x_1	0	1	y_0	0	1	y_1	1
x_2	1	0	y_1	1	0		
x_3	1	1					

Один из возможных вариантов кодирования состояний автоматов Мура и Мили представлен в таблице 18. Для кодирования автоматов в соответствии с табл. 18 требуется два триггера.

Таблица 18 – Кодирование состояний автоматов (рис. 9, рис. 10)

Состояния авто- мата Мура Q	Коды		Состояния авто- мата Мили	Коды
	q_1	q_0		q_0
γ_0	0	0	β_0	0
γ_1	0	1	β_1	1
γ_2	1	0		
γ_3	1	1		

Однако двухразрядное кодирование четырех состояний автомата Мура (рис. 9) не обеспечивает получение единичного кодового расстояния между кодами смежных узлов графа автомата. Следовательно, при смене состояний автомата возможны логические состязания элементов памяти. Для исключения нежелательных проявления состязаний элементов памяти увеличим разрядность кода состояния автомата Мура до трех (таблица 19) и введем один дополнительный переход для обеспечения единичного кодового расстояния между смежными вершинами графа в соответствии с рисунком 15 введем одно промежуточное неустойчивое состояние γ_1^* для перехода $\gamma_0 = \delta(\gamma_1, x_0)$, как это показано на рисунке 17. Таблица 20 является структурной таблицей автомата Мура.

Структурный граф автомата Мура представлен на рис. 18. Все смежные вершины графа обозначены кодовыми комбинациями с кодовым расстоянием, равным единице.

Два состояния автомата Мили (рис. 16) возможно представить комбинациями с кодовыми расстояниями, отличающимися на единицу и тем самым обес-

печивающими отсутствие состязаний элементов памяти.

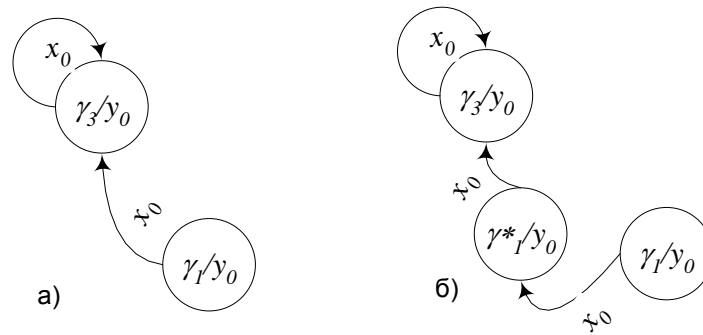


Рис. 17 Ведение дополнительного состояния $\gamma_1^* = \delta(\gamma_1, x_0)$ для автомата Мура

Таблицы 21 и 22 являются соответственно структурными таблицами переходов и выходов автомата Мили. Структурный граф автомата Мили представлен на рис. 19.

Таблица 19 – Кодирование состояний автоматов

Состояния автомата Мура Q	Коды		
	q_2	q_1	q_0
γ_0	0	1	0
γ_1	1	0	1
γ_2	1	0	0
γ_3	0	0	0
γ_1^*	0	0	1

Состояния автомата Мили	Коды
	q_1
β_0	0
β_1	1

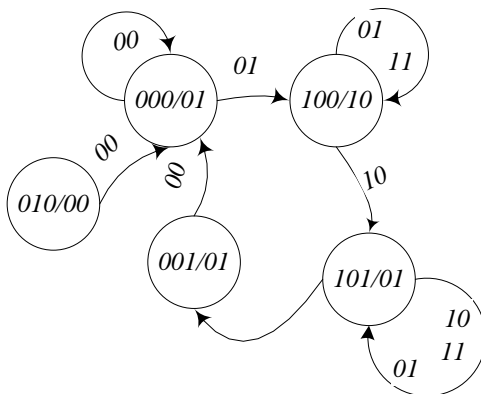


Рис. 18 Структурный граф автомата Мура

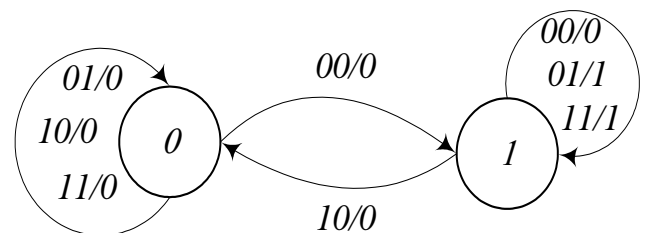


Рис. 19 Структурный граф автомата Мили

Таблица 20 – Структурная таблица переходов-выходов автомата Мура

X		y^*			y_0			y_0			y_1			y_0		
		d_1	d_0		d_1	d_0		d_1	d_0		d_1	d_0		d_1	d_0	
		0	0		0	1		0	1		1	0		0	1	
		γ_0			γ_3			γ_1			γ_2			γ_1^*		
		q_2	q_1	q_0	q_2	q_1	q_0	q_2	q_1	q_0	q_2	q_1	q_0	q_2	q_1	q_0
a_1	a_0	0	1	0	0	0	0	1	0	1	1	0	0	0	0	1
0	0	0	0	0	0	0	0	0	0	1				0	0	0
0	1				1	0	0	1	0	1	1	0	0			
1	0							1	0	1	1	0	1			
1	1							1	0	1	1	0	0			

Таблица 21 – Структурная таблица переходов автомата Мили

X		β_0	β_1
a_1	a_0	0	1
0	0	1	1
0	1	0	1
1	0	0	0
1	1	0	1

Таблица 22 – Структурная таблица выходов автомата Мили

X		β_0	β_1
a_1	a_0	0	1
0	0	0	0
0	1	0	1
1	0	0	0
1	1	0	1

4.4.4 Определение функций возбуждения и выходных функций

Следующим этапом структурного синтеза является определение функций возбуждения элементов памяти и выходных функций автомата.

Пусть автомат Мура требуется реализовать на RS-триггерах, а автомат Мили на Т-триггерах (рис. 20).

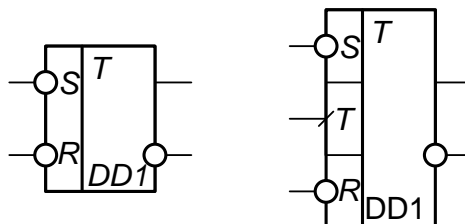


Рис. 20 Условные графические обозначения RS-триггера и Т-триггера со входами предварительной установки

Таблицу возбуждения элементов памяти строят на основе структурных таб-

лиц переходов автоматов и таблиц входов используемых триггеров (табл. 23, табл. 24).

Блок памяти синтезируемого автомата Мура должен содержать три RS-триггера, выходные сигналы которых кодируют состояние автомата - $\gamma_0 \dots \gamma_3$.

Строки таблицы возбуждения элементов памяти (табл. 26) отмечаются кодовыми комбинациями входных сигналов, а столбцы – кодовыми комбинациями состояний триггеров, кодирующих состояния автомата в соответствии со структурной таблицы переходов. На пересечениях j -й строки и i -го столбца ставятся входные сигналы триггеров, состояния которых соответствуют переходу $\delta(\lambda_i, x_j)$.

На основании таблицы возбуждения формируем шесть функций возбуждения: $\bar{R}_0, \bar{S}_0, \bar{R}_1, \bar{S}_1, \bar{R}_2, \bar{S}_2$.

Таблица возбуждения показывает, что функции возбуждения зависят от пяти логических переменных и являются частично определенными. Так как каждая функции возбуждения, за исключением \bar{S}_1 , имеет только одно нулевое значение, и активным сигналом возбуждения триггеров является сигнал логического нуля (нулевые значения функций $\bar{R}_0, \bar{S}_0, \bar{R}_1, \bar{S}_1, \bar{R}_2, \bar{S}_2$), то целесообразно доопределить эти функции единичными значениями. В результате получим, что каждая из этих функций является конститuentой нуля (макситермом). Поэтому функции возбуждения удобно представить в конъюнктивной нормальной форме в виде логических сумм пяти переменных:

$$\bar{R}_2 = a_1 \vee a_0 \vee \bar{q}_2 \vee q_1 \vee \bar{q}_0,$$

$$\bar{S}_2 = a_1 \vee \bar{a}_0 \vee q_2 \vee q_1 \vee q_0,$$

$$\bar{R}_1 = a_1 \vee a_0 \vee q_2 \vee \bar{q}_1 \vee q_0,$$

$$\bar{R}_0 = a_1 \vee a_0 \vee q_2 \vee q_1 \vee \bar{q}_0,$$

$$\bar{S}_0 = \bar{a}_1 \vee a_0 \vee \bar{q}_2 \vee q_1 \vee q_0.$$

Необходимость решения задачи минимизации полученных логических выражений отсутствует.

Следует отметить, что в начальное состояние автомат обычно переводится при включении питания сигналом одновибратора. В начальное состояние, определенное кодом 010, автомат можно перевести подачей сигнала сброса (*reset*) на входы $\bar{R}_0, \bar{S}_1, \bar{R}_2$ триггеров. Это соответствует логическому умножению выражений $\bar{R}_0, \bar{S}_1, \bar{R}_2$ на переменную, отображающую сигнал одновибратора:

$$\bar{R}_2 = (a_1 \vee a_0 \vee \bar{q}_2 \vee q_1 \vee \bar{q}_0) \cdot \overline{reset},$$

$$\bar{S}_1 = \overline{reset},$$

$$\bar{R}_0 = (a_1 \vee a_0 \vee q_2 \vee q_1 \vee \bar{q}_0) \cdot \overline{reset}.$$

Выходные функции d_1 и d_0 автомата Мура, являясь функциями состояний, зависят только от трех переменных q_2, q_1, q_0 . Пользуясь структурной таблицей переходов-выходов автомата Мура (табл. 20), построим таблицы истинности для выходных функций d_1 и d_0 .

Выходные функции также являются частично определенными. Доопределяем выходные функции, основываясь на удобстве минимизации. Доопределенные значения функций d_1 и d_0 в таблице указаны в скобках. В результате минимизации получим следующие выражения:

$$d_1 = q_2 \bar{q}_0,$$

$$d_0 = \bar{q}_2 \bar{q}_1 \vee q_2 q_0.$$

Переводим полученные логические выражения в заданный базис логических элементов и вычерчиваем функциональную схему автомата.

Таблица 25 – Таблица истинности выходных функций автомата Мура

Состояния автомата	Состояния триггеров			Выходные сигналы	Выходные функции	
	q_2	q_1	q_0		d_1	d_0
γ_3	0	0	0	y_0	0	1
γ^*_1	0	0	1	y_0	0	1
γ_0	0	1	0	y^*	0	0
-	0	1	1	-	(0)	(0)
γ_2	1	0	0	y_1	1	0
γ_1	1	0	1	y_0	0	1
-	1	1	0	-	(1)	(0)
-	1	1	1	-	(0)	(1)

Для базиса ИЛИ-НЕ и для трехвходовых логических элементов получим следующие выражения:

$$\bar{R}_2 = \overline{(a_1 \vee a_0 \vee \bar{q}_2 \vee q_1 \vee \bar{q}_0) \cdot reset} = a_1 \vee a_0 \vee \bar{q}_2 \vee q_1 \vee \bar{q}_0 \vee reset;$$

$$\bar{S}_2 = \overline{a_1 \vee \bar{a}_0 \vee q_2 \vee q_1 \vee q_0} = a_1 \vee \bar{a}_0 \vee q_2 \vee q_1 \vee q_0;$$

$$\bar{R}_1 = \overline{a_1 \vee a_0 \vee q_2 \vee \bar{q}_1 \vee q_0} = a_1 \vee a_0 \vee q_2 \vee \bar{q}_1 \vee q_0;$$

$$\bar{S}_1 = reset;$$

$$\bar{R}_0 = \overline{(a_1 \vee a_0 \vee q_2 \vee q_1 \vee \bar{q}_0) \cdot reset} = a_1 \vee a_0 \vee q_2 \vee q_1 \vee \bar{q}_0 \vee reset;$$

$$\bar{S}_0 = \overline{\bar{a}_1 \vee a_0 \vee \bar{q}_2 \vee q_1 \vee q_0} = \bar{a}_1 \vee a_0 \vee \bar{q}_2 \vee q_1 \vee q_0;$$

$$d_1 = \overline{\overline{q_2} \overline{q_0}} = \overline{\overline{q_2} \vee q_0};$$

$$d_0 = \overline{\overline{\overline{q_2} \overline{q_1}} \vee \overline{\overline{q_2} q_0}} = \overline{\overline{q_2} \vee q_1} \vee \overline{\overline{q_2} \vee \overline{q_0}} = \overline{\overline{\overline{\overline{q_2} \vee q_1} \vee \overline{q_2} \vee \overline{q_0}}}$$

4.4.5 Функциональные схемы автоматов.

Моделирование функционирования

Функциональная схема автомата Мура на RS-триггерах и трехвходовых элементах ИЛИ-НЕ, реализующего вход-выходную последовательность, представлена в Приложении 2.

Для проверки соответствия синтезированной схемы заданным условиям функционирования автомата выполним функциональное моделирование полученной схемы при помощи программы функционального моделирования Electronics Workbench [10].

Модель автомата, собранная на рабочем поле Electronics Workbench, представлена на рис. 21. Источником входных сигналов является генератор двоичных слов, а выходные сигналы модели автомата индицируются при помощи логического анализатора. Настройка генератора слов представлена на и временные диаграммы функционирования автомата, полученные при помощи логического анализатора, представлены на рис. 22.

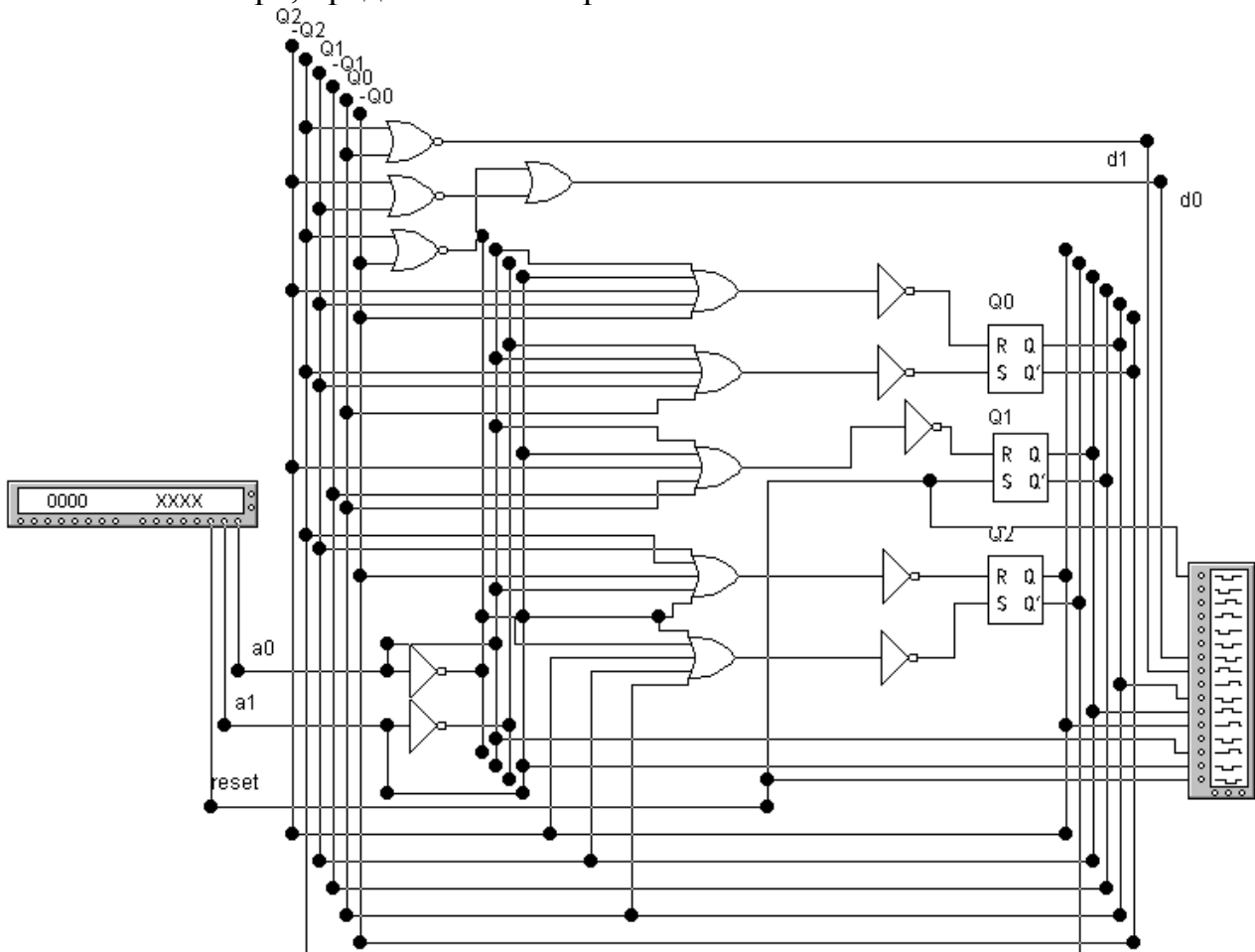


Рис. 21 Модель автомата Мура (Electronics Workbench)

Анализ входных и выходных сигналов позволяет сделать вывод, что синтезируемая схем соответствует заданным условиям функционирования – получена требуемая вход-выходная последовательность.

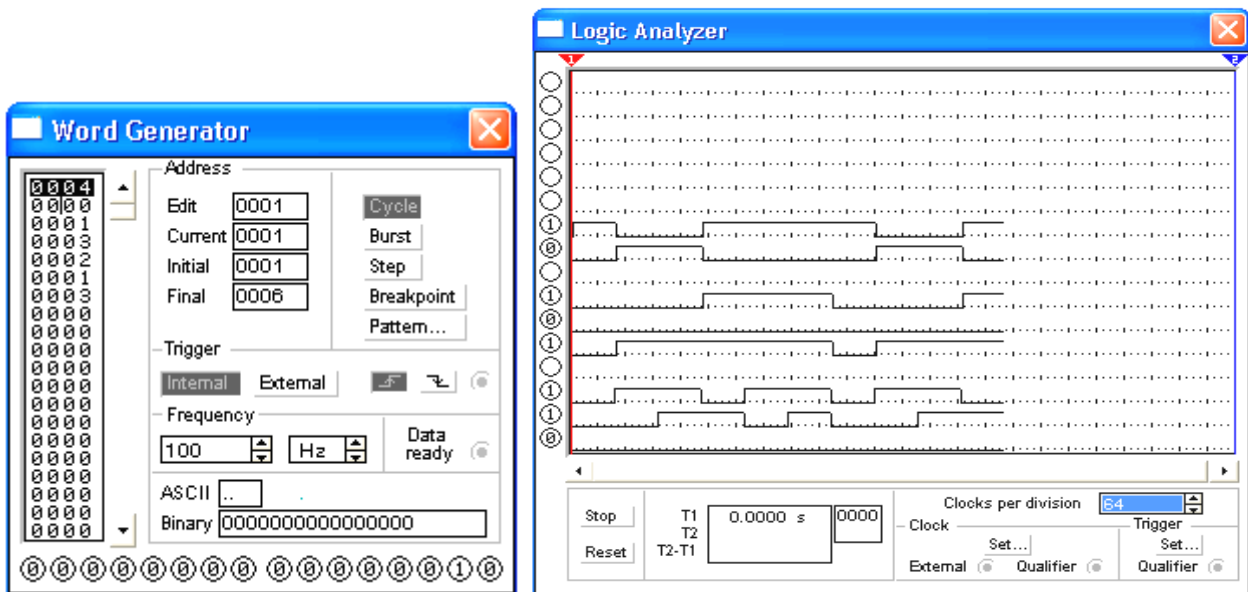


Рис. 22 Настройка генератора слов и временные диаграммы функционирования автомата Мура

Блок памяти синтезируемого автомата Мили содержит один Т-триггер, выходной сигнал которого кодирует состояния автомата - β_0, β_1 .

На основании таблицы возбуждения формируем функцию возбуждения T :

$$T = \bar{a}_1 \bar{a}_0 \bar{Q} \vee a_1 \bar{a}_0 Q,$$

где Q - состояние T -триггера.

Выходную функцию d_0 автомата Мили удобно представить в дизъюнктивной форме, так как, являясь функцией трех переменных a_1, a_0 и Q , функция определена единичным значением только на двух точках своего определения.

$$d_0 = \bar{a}_1 a_0 Q \vee a_1 a_0 Q = a_0 Q.$$

Представим функцию возбуждения триггера и выходную функцию в базисе двухвходовых элементов И-НЕ.

$$T = \overline{\bar{a}_1 \bar{a}_0 \bar{Q} \vee a_1 \bar{a}_0 Q} = \overline{\bar{a}_1 \bar{a}_0 \bar{Q}} \cdot \overline{a_1 \bar{a}_0 Q} = \overline{\bar{a}_1 \bar{a}_0 \bar{Q}} \cdot \overline{a_1 \bar{a}_0 Q}$$

$$d_0 = \overline{\overline{a_0 Q}}$$

Функциональная схема автомата приведена в Приложении 3.

Модель автомата, собранная на рабочем поле Electronics Workbench, представлена на рис. 23. Источником входных сигналов является генератор двоичных слов, а выходные сигналы модели автомата индицируются при помощи логического анализатора. В качестве Т-триггера используется JK-триггер, включенный по схеме счетного режима. Настройка генератора слов представлена на и временные диаграммы функционирования автомата, полученные при помощи

логического анализатора, представлены на рис. 24.

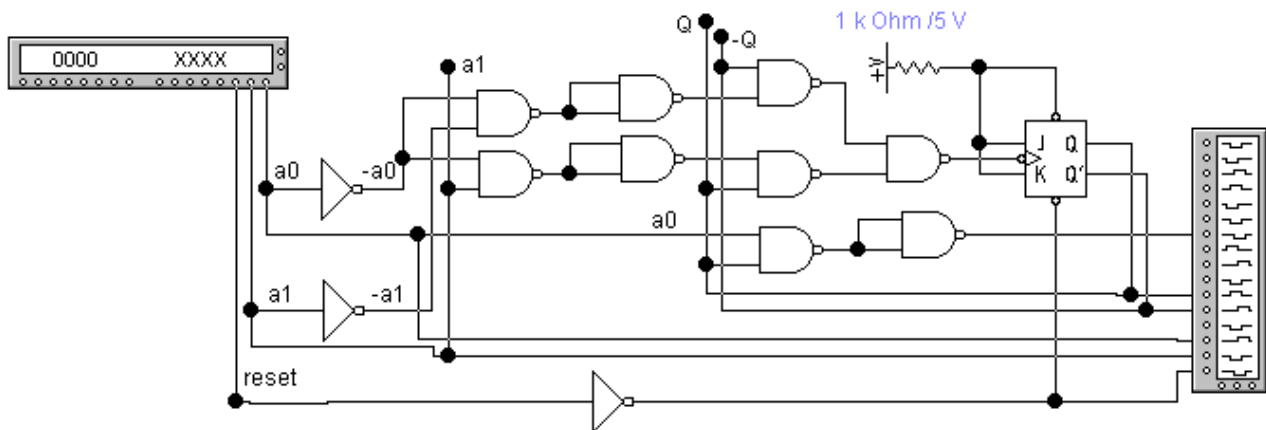


Рис. 23 Модель автомата Мили (*Electronics Workbench*)

Анализ входных и выходных сигналов позволяет сделать вывод, что синтезируемая схем соответствует заданным условиям функционирования – получена требуемая вход-выходная последовательность.

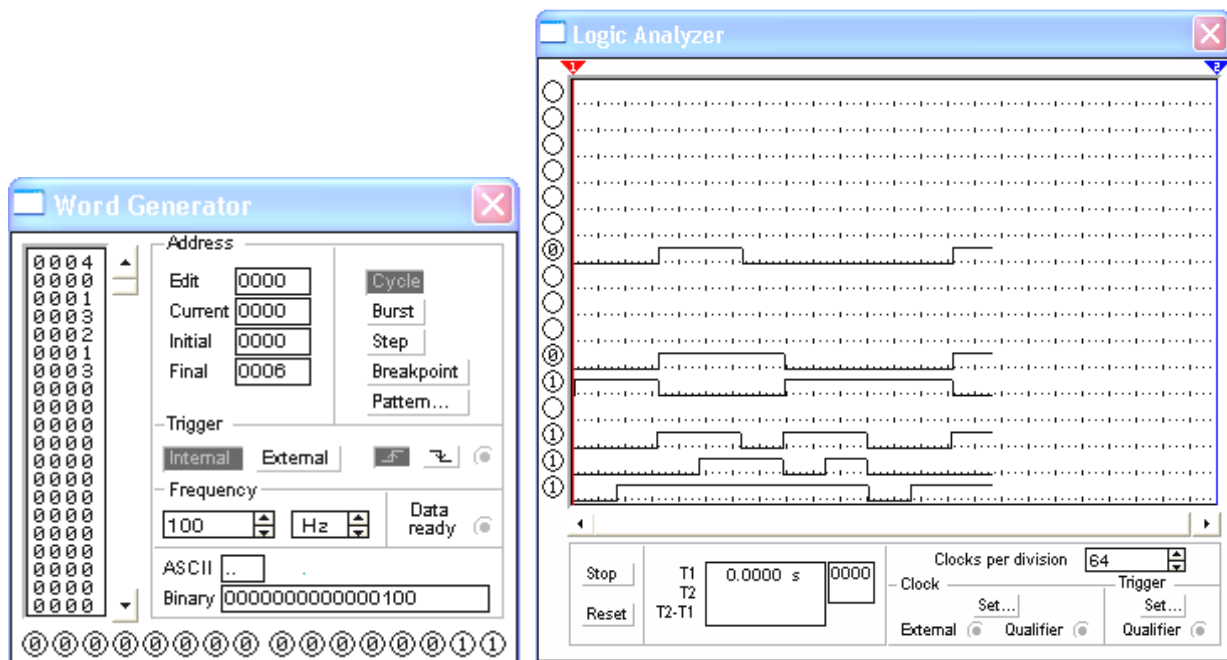


Рис. 24 Настройка генератора слов и временные диаграммы функционирования автомата Мили

5. Реализация дискретных автоматов на современной элементной базе

Для реализации дискретных управляющих автоматов традиционно использовались микросхемы малой и средней степени интеграции. Известно, что применение МИС и СИС влечет за собой рост числа корпусов ИС, усложнение монтажа, снижение надежности системы. Избежать этих недостатков позволяет использование программируемых логических интегральных микросхем (ПЛИС), которые представляют собой БИС с программируемой и перепрограммируемой структурой. С их помощью возможно создание цифрового автомата на одном кристалле.

Базовая структура ПЛИС показана на рис. 25. Ее основой служит последовательность программируемых матриц элементов *И* и элементов *ИЛИ*. В матрицах имеются системы горизонтальных и вертикальных связей, в узлах пересечения которых в процессе программирования создаются или разрушаются элементы связи. В структуру входят также блоки входных (БВх) и выходных (БВых) буферных каскадов.

Переменные $X_1 \dots X_m$ подаются через БВх на входы элементов *И* (конъюнкторов), и в матрице *И* образуются k термов. Под термом понимается конъюнкция, связывающая входные переменные, которые могут быть представлены как в прямой, так и в инверсной форме. Число формируемых термов равно числу конъюнкторов или, что то же самое, числу выходов матрицы *И*.

Термы подаются далее на входы матрицы *ИЛИ*, т. е. на входы дизъюнкторов, формирующих выходные функции. Число дизъюнкторов равно числу вырабатываемых функций n .

Таким образом, ПЛИС реализует дизъюнктивную нормальную форму воспроизводимых функций (двухуровневую логику). ПЛИС способна реализовать систему n логических функций от m аргументов, содержащую не более k термов. Воспроизводимые функции являются комбинациями из любого числа термов, формируемых матрицей *И*. Какие именно термы будут выработаны и какие комбинации этих термов составят выходные функции, определяется программированием ПЛИС. В микросхеме ПЛИС можно запрограммировать комбинационную часть дискретного автомата.

Существуют и более сложные схемы ПЛИС, например, схемы с двунаправленными выводами, а также схемы с памятью. Последние наиболее удобны для построения автоматов, т. к. помимо комбинационной части содержат на кристалле триггеры обычно D типа, собранные в регистры. Структура ПЛИС с памятью показана на рис. 26. Кроме трех обычных параметров (m , n , k), она имеет и параметр s – число элементов памяти (разрядов регистра). Структура БИС совпадает с канонической схемой автомата. Результат текущего шага обработки информации зависит от результатов предыдущих шагов, что обеспечивается обратной связью с регистра на вход ПЛИС. Максимальное число внутренних состояний автомата – 2^s . Автомат рассматривается как синхронный, т. к. петля

обратной связи активизируется только по разрешению тактовых сигналов ТС.

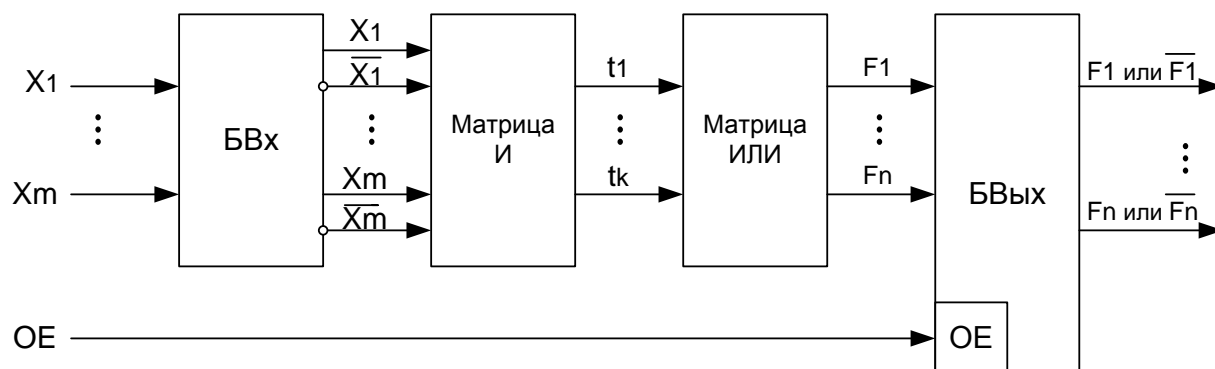


Рис. 25 Базовая структура ПЛИС

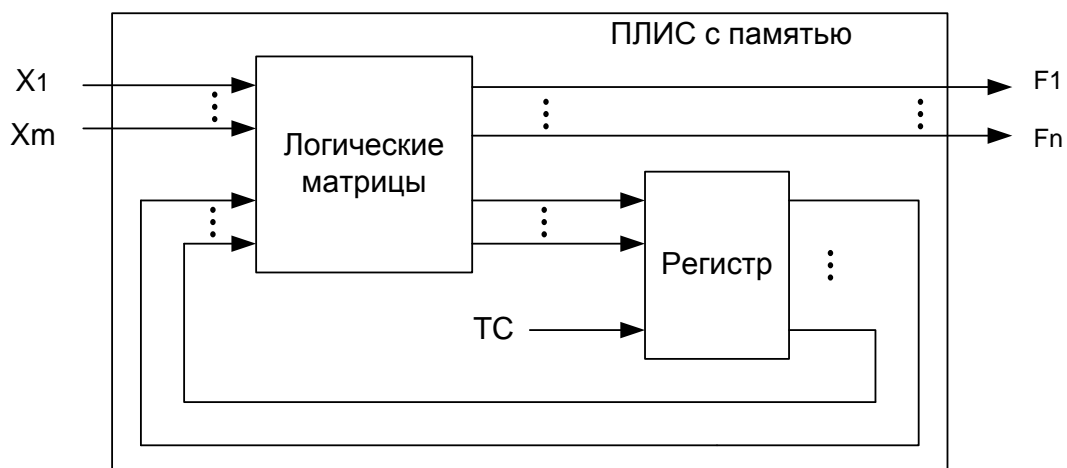


Рис. 26 Структура ПЛИС с памятью

Список литературы

1. Соловьев В.В., Булатова И.Р. Стандартные программируемые логические устройства // Зарубежная радиоэлектроника. 2000. № 4. С. 66–76.
2. Соловьев В.В., Булатова И.Р. Архитектуры сложных программируемых логических интегральных схем // Зарубежная радиоэлектроника. 2000. № 5. С. 62–78.
3. Энциклопедия кибернетики / Под ред. В.М. Глушкова. – Киев: Главная редакция украинской советской энциклопедии, 1974
4. Соловьев А.Я. Основы информатики: Учебник для вузов. – М.: Изд-во МГТУ им.Н.Э. Баумана, 2001.
5. Трахтенберг Б.А., Барадин Я.М. Конечные автоматы: Поведение и синтез. М.: Наука, 1970.
6. Глушков В.М. Синтез цифровых автоматов. М.: Наука, 1962.
7. Янов Ю.И. О логических схемах алгоритмов //Проблемы кибернетики. М.: Физматгиз, 1958. Вып.1. С.29-53
8. Ершов А.П., Ляпунов А.А. О формализации понятия программы. // Кибернетика, 1967, №5.
9. Дискретная математика и математические основы кибернетики. / Под ред. С.В. Яблонского и О.Б. Лупанова. Т.1. – М.: Наука, 1974.
10. Карлащук В.И. Электронная лаборатория на IBM PC: Программа Electronics Workbench и ее применение. – М.: СОЛОН-Р, 2001.

Приложение 1

Задание на синтез дискретного автомата

Вариант	Входная последовательность	Выходная последовательность	Тип автомата	Тип элементов памяти	Реализация функций возбуждения и выходных функций
1.	$x_0, x_1, x_2, x_3, x_4, x_1, x_2, x_4, x_2$	$y_0, y_1, y_2, y_0, y_1, y_2, y_3, y_1, y_2$	Мура	RS-триггеры	И-НЕ
2.	$x_0, x_1, x_2, x_3, x_4, x_1, x_2, x_4, x_1$	$y_0, y_1, y_2, y_0, y_1, y_2, y_3, y_1, y_2$	Мили	D-триггеры	ИЛИ-НЕ
3.	$x_0, x_1, x_2, x_3, x_4, x_1, x_2, x_4, x_2$	$y_0, y_1, y_2, y_0, y_1, y_2, y_3, y_1, y_2$	Мура	JK-триггеры	И-ИЛИ-НЕ
4.	$x_0, x_1, x_2, x_3, x_4, x_3, x_1, x_4, x_3$	$y_0, y_1, y_2, y_0, y_1, y_2, y_3, y_1, y_2$	Мили	T-триггеры	И-НЕ
5.	$x_0, x_1, x_2, x_3, x_2, x_1, x_0, x_3, x_1$	$y_0, y_1, y_2, y_1, y_2, y_2, y_1, y_3, y_0$	Мура	RS-триггеры	мультиплексоры
6.	$x_0, x_1, x_2, x_3, x_0, x_2, x_1, x_4, x_1$	$y_0, y_1, y_2, y_1, y_2, y_2, y_1, y_3, y_0$	Мили	D-триггеры	И-НЕ
7.	$x_0, x_1, x_2, x_3, x_1, x_2, x_3, x_4, x_3$	$y_0, y_1, y_2, y_1, y_2, y_2, y_1, y_3, y_0$	Мура	JK-триггеры	ИЛИ-НЕ
8.	$x_0, x_1, x_2, x_3, x_1, x_2, x_3, x_4, x_1$	$y_0, y_1, y_2, y_1, y_2, y_2, y_1, y_3, y_0$	Мили	T-триггеры	И-ИЛИ-НЕ
9.	$x_0, x_1, x_2, x_3, x_1, x_2, x_3, x_4, x_2$	$y_0, y_1, y_2, y_0, y_1, y_2, y_0, y_1, y_3$	Мура	RS-триггеры	И-НЕ
10.	$x_0, x_1, x_2, x_3, x_1, x_2, x_3, x_1, x_4$	$y_0, y_1, y_2, y_0, y_1, y_2, y_0, y_1, y_3$	Мили	D-триггеры	мультиплексоры
11.	$x_0, x_1, x_2, x_3, x_1, x_2, x_3, x_2, x_4$	$y_0, y_1, y_2, y_0, y_1, y_2, y_0, y_1, y_3$	Мура	JK-триггеры	И-НЕ
12.	$x_0, x_1, x_2, x_3, x_1, x_2, x_3, x_1, x_3$	$y_0, y_1, y_2, y_0, y_1, y_2, y_0, y_1, y_3$	Мили	T-триггеры	ИЛИ-НЕ
13.	$x_0, x_1, x_2, x_3, x_4, x_2, x_3, x_4, x_1$	$y_0, y_1, y_2, y_3, y_1, y_2, y_0, y_2, y_1$	Мура	RS-триггеры	И-ИЛИ-НЕ
14.	$x_0, x_1, x_2, x_3, x_4, x_2, x_3, x_4, x_3$	$y_0, y_1, y_2, y_3, y_1, y_2, y_0, y_2, y_1$	Мили	D-триггеры	И-НЕ
15.	$x_0, x_1, x_2, x_3, x_4, x_2, x_3, x_4, x_1$	$y_0, y_1, y_2, y_3, y_1, y_2, y_0, y_2, y_1$	Мура	JK-триггеры	мультиплексоры

Приложение 2

Приложение 3

1 Соловьев В.В., Булатова И.Р. Стандартные программируемые логические устройства // Зарубежная радиоэлектроника. 2000. № 4. С. 66–76.

2 Соловьев В.В., Булатова И.Р. Архитектуры сложных программируемых логических интегральных схем // Зарубежная радиоэлектроника. 2000. № 5. С. 62–78.

3 Энциклопедия кибернетики / Под ред. В.М. Глушкова. – Киев: Главная редакция украинской советской энциклопедии, 1974

4 Соловьев А.Я. Основы информатики: Учебник для вузов. – М.: Изд-во МГТУ им.Н.Э. Баумана, 2001.

5 Трахтенберг Б.А., Барадин Я.М. Конечные автоматы: Поведение и синтез. М.: Наука, 1970.

6 Глушков В.М. Синтез цифровых автоматов. М.: Наука, 1962.

7 Янов Ю.И. О логических схемах алгоритмов //Проблемы кибернетики. М.: Физматгиз, 1958. Вып.1. С.29-53

8 Ершов А.П., Ляпунов А.А. О формализации понятия программы. // Кибернетика, 1967, №5.

9 Дискретная математика и математические основы кибернетики. / Под ред. С.В. Яблонского и О.Б. Лупанова. Т.1. – М.: Наука, 1974.

10 Карлащук В.И. Электронная лаборатория на IBM PC: Программа Electronics Workbench и ее применение. – М.: СОЛОН-Р, 2001.