

ВЯТСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ФАКУЛЬТЕТ АВТОМАТИКИ И ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКИ
КАФЕДРА ЭЛЕКТРОННЫХ ВЫЧИСЛИТЕЛЬНЫХ МАШИН

Теория автоматов (часть I)

Конспект лекций

Киров 2010

Теория автоматов (часть I). Конспект лекций /Киров, Вятский государственный университет, 2010, 56с.

В конспекте лекций по курсу «Теория автоматов» (Часть I) приведен необходимый теоретический материал, включающий основы прикладной теории цифровых автоматов, основные определения и правила синтеза абстрактных конечных автоматов, схема дискретного преобразователя В.М. Глушкова и три основных модели конечных автоматов: модель Мили, модель Мура и модель С-автомата. Далее рассмотрены методы кодирования элементов памяти, минимизация автоматов с памятью, канонический метод синтеза структурных автоматов. Особое внимание уделено правилам построения оптимальной комбинационной схемы автомата по каноническим уравнениям с учётом выбранных элементов памяти. Указана соответствующая литература.

Курс лекций предназначен для студентов очного обучения специальности 230101 - "Вычислительные машины, комплексы, системы и сети", а также бакалавров направления 230100.62 – "Информатика и вычислительная техника".

Составитель: к.т.н., доцент каф. ЭВМ В.Ю. Мельцов.

Оглавление

1. Введение.....	4
2. Задачи анализа и синтеза.....	4
3. Абстрактный управляющий автомат	7
4. Синтез абстрактных автоматов.....	9
5. Синхронный автомат	10
6. Асинхронные автоматы	11
7. Автоматы Мили и Мура	12
8. Способы задания автоматов.....	14
8.1 Табличный способ задания автоматов	15
8.2. Задание автомата с помощью графа.....	17
8.3. Матричный способ задания автоматов	19
9. Переход от автомата Мили к автомату Мура и обратно.....	19
10. Этапы синтеза автоматов.....	24
11. Минимизация числа внутренних состояний автомата.....	26
12. Минимизация полностью определённых автоматов.	27
13. Совмещённая модель автомата (С-автомата).....	31
14. Структурный синтез автомата	32
14.1. Канонический метод структурного синтеза автомата.....	32
14.2. Структурный синтез С-автомата	35
15. Кодирование состояний автомата	42
15.1. Метод противогоночного кодирования состояний автомата	43
15.2. Соседнее кодирование состояний автомата	46
15.3. Кодирование состояний автомата для минимизации комбинационной схемы	48
16. Элементарные автоматы памяти	52
17. Синтез автоматов на ПЛИС и ПЗУ	55
Рекомендуемая литература	57

1. Введение

В последние годы с большой интенсивностью ведутся исследования и работы по созданию и применению различных автоматических систем для обработки информации. Такие системы обычно реализуются в виде блоков, образующих систему управления и переработки информации. В соответствии с этим возникла новая научная дисциплина, которая получила название «Теория систем».

Целью теории систем является задача создания арсенала идей и средств, которые были бы в равной степени полезны специалистам в самых различных областях (электротехника, механика, физика, химия, социология и т.п.). Данная цель достигается путем рассмотрения системы не через ее внутреннюю структуру, а через математические законы, определяющие ее наблюдаемое поведение. При этом наиболее часто используется подход, называемый методом «черного ящика» (т.е. метод, в котором анализируются входные данные и данные, получаемые на выходе автомата, при этом внутренние процессы не рассматриваются). Было доказано, что все системы могут быть охарактеризованы в одинаковых терминах и проанализированы с помощью одного и того же набора правил.

Составной частью теории систем является теория автоматов. Она имеет дело с математическими моделями, предназначенными для в той или иной степени приближенного отображения физических явлений. Применение модели теории автоматов не ограничивается какой-либо частной областью, а возможно для решения проблем практически в любой области исследования.

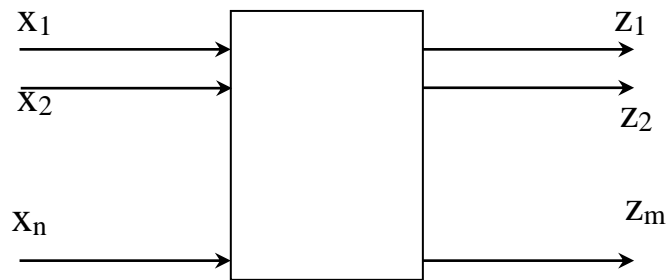
2. Задачи анализа и синтеза

Большинство проблем, встречающихся в науке и технике, можно разбить на следующие две категории: *задачи анализа* и *задачи синтеза*

Задачи анализа состоят в предсказании поведения определенной заранее заданной системы.

Задачи синтеза состоят в построении системы функций по заранее заданному алгоритму.

Любую систему (любой автомат) можно представить в виде многополюсного «черного ящика»:



Автомат имеет следующие составляющие:

1. Входные переменные, которые представляют собой воздействия, генерируемые извне и влияющие на поведение исследуемой системы.
2. Выходные переменные, называемые *реакцией системы*, представляющие собой величины, характеризующие поведение данной системы.

Входные полюсы (входные каналы) соответствуют местам поступления входных переменных, снабжаются стрелками, направленными внутрь «черного ящика». Входные и выходные переменные с точки зрения абстрактной теории автоматов не имеют какого-либо физического смысла.

Предполагается, что любая система, представимая основной моделью, управляется некоторым синхронизирующим источником. Все переменные системы изменяются в определенные дискретные моменты времени, в которые подается синхронизирующий сигнал. Эти моменты времени называются *тактами (тактовыми моментами)* и обозначаются буквой t_s . Тогда поведение системы в любой момент времени t_s не зависит от интервала времени между t_s и t_{s-1} . Кроме того, независимой величиной, относительно которой определяются все переменные системы, является не время, а порядковый номер, связанный с тактом. Системы, удовлетворяющие вышеизложенным предположениям, называются *синхронными*. Асинхронные

же системы, которые будут рассмотрены несколько позже, меняют свои сигналы, не привязываясь к синхронизирующему сигналу.

Необходимо отметить три возможных типа автоматов, отличающихся друг от друга в функциональном отношении.

1. В устройствах первого типа набор выходных сигналов, вырабатываемых в момент времени $(t + \Delta t)$, зависит только от набора входных сигналов, поданных в момент времени t , а не зависит от сигналов, поступивших на входы автомата в предшествующее время. Интервал Δt – время реакции автомата. Он остается одинаковым для исходного t при любых допустимых наборах входных сигналов. Такое однозначное и неизменное во времени соответствие между наборами входных и выходных сигналов обуславливается неизменностью внутреннего состояния автоматов и независимостью этого состояния от внешнего воздействия. Устройства такого типа называют **автоматами без памяти**.

2. В автоматах второго типа набор выходных сигналов, вырабатываемый в некоторый дискретный момент времени зависит не только от сигналов, поданных в тот же момент времени, но и от сигналов, поступивших ранее. Эти предшествующие внешние воздействия фиксируются в автомате путем изменения его внутреннего состояния. Таким образом, реакция данного автомата однозначно определяется поступившим набором входных сигналов и его внутренним состоянием на данный момент времени. Этими же факторами однозначно определяется и то состояние, в которое автомат перейдет.

3. **Конечным автоматом** называется объект, имеющий конечное число входов, конечное число внутренних состояний, работа которого носит детерминированный (от англ. - “определенный”) характер. Если конечный автомат снабдить внешней памятью и допустить ее неограниченное расширение, то такая система будет принадлежать к автоматам третьего типа, например, **машина Тьюринга**. Он показал, что с помощью автоматов третьего типа может быть смоделирована любая система, т.е. реализован любой алгоритм по переработке информации.

3. Абстрактный управляющий автомат

Автоматические дискретные устройства нашли самое широкое распространение при автоматизации управления производственных процессов. Все более сложные функции управления возлагаются на управляющие автоматы. Поэтому значительно увеличивается сложность управляющих автоматов. Управляющий автомат любой автоматической системы может рассматриваться как некое устройство, реализующее алгоритм ее функционирования. При этом отдельные функциональные блоки (ФБ) выполняют те или иные этапы алгоритмов. Для того чтобы управляющий автомат реализовал весь алгоритм, между функциональными блоками должны быть функциональные связи, определяющие порядок работы ФБ в соответствии с заданным алгоритмом в процессе функционирования системы.

Существует два основных принципа построения управляющих автоматов:

1. Управляющий автомат с *распределенными* (рассредоточенными) функциональными связями (см. рис.1). В управляющих автоматах первого типа изменение алгоритма работы (программы) требует изменения функциональной связи и иногда добавления новых ФБ (поэтому они применяются реже, чем автоматы второго типа).

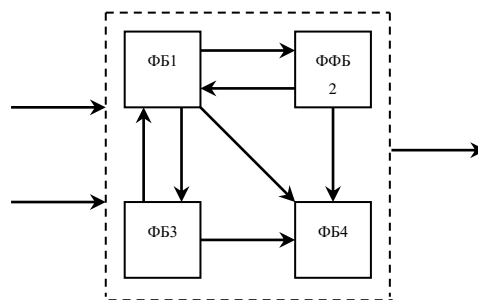


Рис.1

2. Управляющий автомат с *концентрируемыми* функциональными связями (см. рис. 2). Второй принцип создания автоматов характеризуется отсутствием функциональных связей между блоками. От центрального блока управления (ЦБУ) на функциональные блоки передается управляющая информация, от ФБ поступает осведомительная. Порядок работы ФБ определяется выдаваемой ЦБУ информацией (заложенной в ЦБУ программой) и зависит от поступающей осведомительной информации.

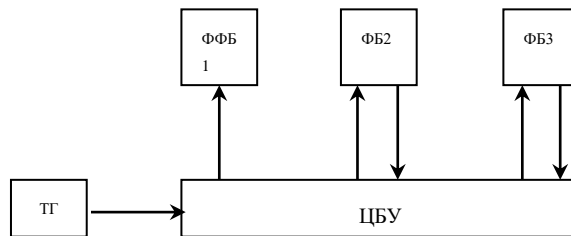


Рис.2

Программный способ управления удобен в том случае, когда в процессе эксплуатации автомата необходимо изменять режим его функционирования.

В программно управляемом автомате все ФБ можно разделить на 2 группы:

1. **Логические** функциональные блоки (ЛФБ) осуществляют проверку каких-либо условий. После завершения своей работы логический ФБ передает сигнал о значении передаваемого условия центральному блоку управления. Обычно ФБ запускается сигналом от ЦБУ.
2. **Операторные** функциональные блоки (ОФБ) осуществляют управление объектом автоматической системы. Это исполнительные приборы, механизмы. Во время выполнения определенных операций операторный ФБ может обмениваться информацией с ЦБУ.

В качестве ЛФБ и ОФБ могут использоваться схемы дискретного действия, сами имеющие достаточно сложную блочную структуру.

В зависимости от способа получения центральным БУ сигнала об окончании работы ФБ различают синхронные и асинхронные режимы работы. При **синхронном** режиме работы ЦБУ сигналы от ОФБ поступают в виде единиц; сигналы от ЛФБ - в виде нулей или единиц. При синхронном режиме сигналы обмена контролируются тактовым генератором (ТГ), при этом его частота выбирается такой, что любой ФБ успевает закончить свою работу до появления очередного импульса тактового генератора.

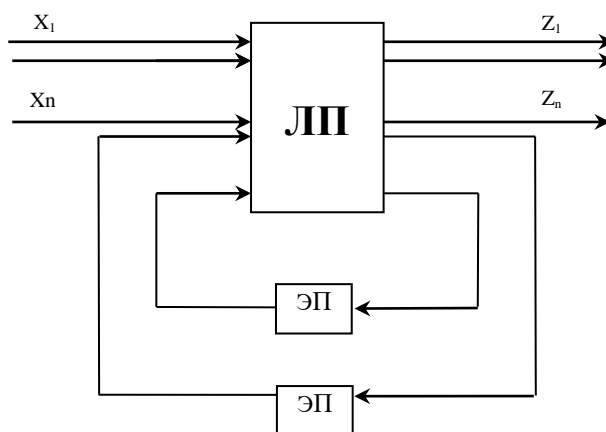
В связи с тем, что продолжительность работы одного ФБ может существенно отличаться от продолжительности работы другого, при синхронном режиме быстродействие управляющего автомата определяется наиболее медленным его функциональным блоком. Для повышения быстродействия частота ТГ может выбираться за основание

продолжительности работы большинства ФБ. Тогда при работе функциональных блоков, частота которых меньше частоты ТГ (соответственно период больше периода ТГ), операции разбиваются на несколько этапов.

Максимальное быстродействие можно достичь при сочетании *асинхронного* режима работы и оптимальной программе. В этом случае требуется усложнение ЦБУ, поскольку сигналы всех ФБ поступают в разное время.

4. Синтез абстрактных автоматов

Любой абстрактный автомат можно представить в виде дискретного устройства, которое имеет N входов и K выходов, кроме того, такое дискретное устройство может иметь S обратных связей, проходящих через устройства задержки (см. рис.3).



Часть устройства, в котором сосредоточены логические элементы (элементы без памяти), называют *логическим преобразователем* (ЛП). На входы элементов памяти, выступающих в роли элементов задержки, воздействуют сигналы, снимаемые с дополнительных (внутренних) выходов ЛП. Будем предполагать, что каждый элемент памяти будет находиться в двух состояниях: 0 и 1. Каждый элемент памяти может сохранить 2^s состояний.

При двузначных значениях входных сигналов (0, 1) число различных входных состояний будет равно 2^N . Говорят, что i -ый набор ($i \in [1, n]$) значений входных аргументов, воздействуют на различные входы ЛП и образует состояние входа ρ_i .

Аналогично число состояний выходов определяется значениями выходных аргументов λ_j , где $j \in [1, k]$.

Состояния всех элементов памяти определяет состояние автомата. S_M – число возможных внутренних состояний автомата.

Конечным автоматом называется устройство, определенное конечным множеством состояний входов – $P = \{\rho_1, \rho_2, \dots, \rho_N\}$, конечным множеством состояний выходов – $\Lambda = \{\lambda_1, \lambda_2, \dots, \lambda_s\}$, конечным множеством внутренних состояний $S = \{s_1, s_2, \dots, s_M\}$, а также функцией переходов, определяющей порядок смен внутренних состояний (φ) и функцией выходов, задающих выходные состояния в зависимости от P и S (ψ).

$$S = \langle P, S, \Lambda, \varphi, \psi, S_0 \rangle$$

Из множества внутренних состояний выделяется некоторое начальное состояние, называемое начальным внутренним состоянием автомата.

Будем предполагать, что автомат функционирует в некоторые дискретные моменты времени (такт). Время, в течение которого не происходит изменение входного сигнала ρ , обозначается через T и в зависимости от того, чем определяется длительность этого интервала, будем различать два класса автоматов: синхронные и асинхронные.

5. Синхронный автомат

Синхронный автомат характеризуется тем, что имеет тактовый генератор, и входные сигналы могут воздействовать на автомат только при наличии тактового сигнала.

Для синхронных автоматов характерно следующее:

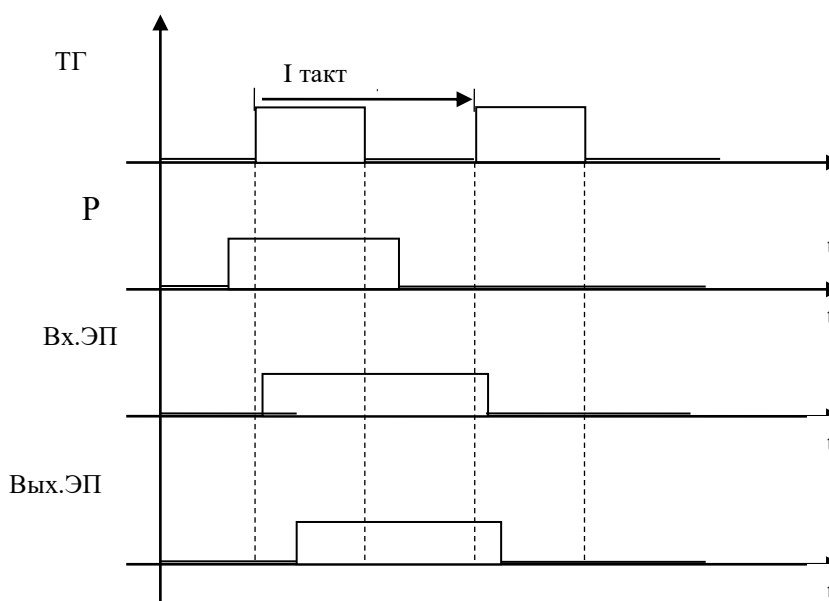
1. Входной сигнал воздействует на автомат в строго фиксированные моменты времени, то есть $T = \text{const}$.

2. Изменение внутреннего состояния автомата осуществляется в моменты времени, когда нет воздействия входных сигналов.

Таким образом, время такта (T) в синхронных автоматах строго фиксировано и определяется характеристиками тактового генератора.

Автомат может воспринимать новое состояние входа, лишь после того, как он перешел в определенное внутреннее состояние.

Следовательно, частота тактового генератора выбирается такой, чтобы до появления следующего импульса автомат успел перейти в это новое внутреннее состояние. $\Delta \ll T$



Обычно в абстрактной теории автоматов не интересуются поведением автомата, считая, что переход из одного состояния в другое происходит мгновенно.

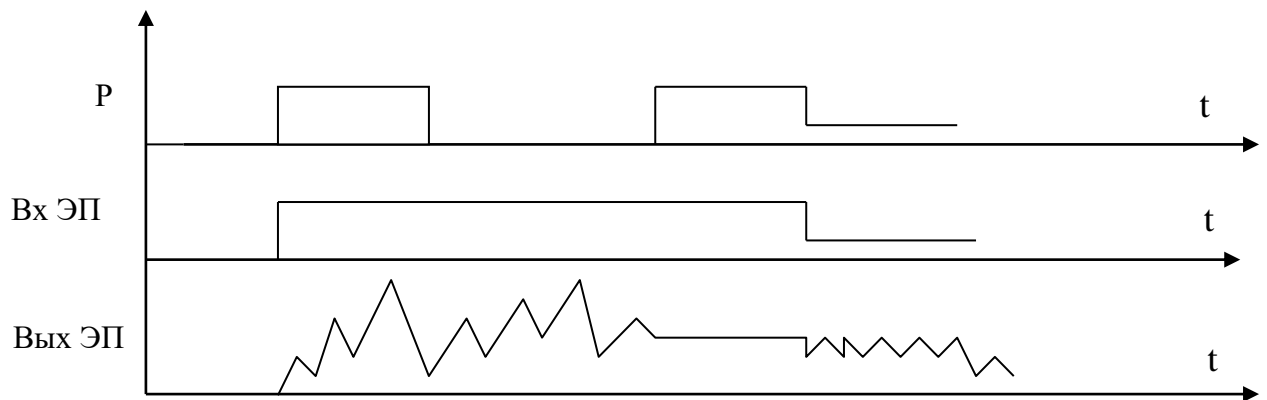
6. Асинхронные автоматы

В асинхронных автоматах длительность интервала T , в течение которого остаются неизменными входные сигналы, является величиной переменной и определяется только моментами изменения состояний входов.

Соответственно, каким бы продолжительным не был интервал времени, в течение которого остается неизменным состояние входа, он будет восприниматься автоматом как один и тот же интервал T (такт). Следовательно,

двум последовательным интервалам T_i и T_{i+1} всегда должны соответствовать различные состояния входа.

Изменение внутреннего состояния асинхронного автомата происходит при неизменном состоянии входа.



Для асинхронного автомата характерно следующее:

1. Длительность интервалов T является величиной переменной и определяется изменением состояния входов автомата.

2. Переход в новое внутреннее состояние осуществляется при неизменном состоянии входа.

Следовательно, различие синхронных и асинхронных автоматов: при асинхронном режиме работы управляющий автомат может использовать, как синхронное, так и асинхронное функциональные блоки. Синхронный автомат может использоваться в синхронном и асинхронном режимах работы.

7. Автоматы Мили и Мура

Функционирование или поведение автомата при заданных множествах (P, A, X) и начальном внутреннем состоянии x_0 полностью детерминировано, и определяется функциями переходов и выходов - φ, ψ .

Функция переходов устанавливает зависимость внутреннего состояния автомата в следующий момент времени от состояния входа и внутреннего состояния в настоящий момент времени.

Функция выходов устанавливает зависимость состояния выхода автомата от состояния входа и внутреннего состояния автомата.

Различный характер этих зависимостей для различных автоматов позволяет выделить отдельные типы автоматов в классе синхронных конечных детерминированных автоматов.

Основными являются две модели: Мили и Мура.

Автомат Мили описывается следующими формулами:

1. внутреннее состояние автомата в следующий момент времени зависит от внутреннего состояния автомата в настоящий момент времени и входного сигнала в настоящий момент времени.

$$x(t+1) = \varphi(x(t), \rho(t))$$

2. выходной сигнал автомата в настоящий момент времени зависит от входного сигнала в настоящий момент времени и внутреннего состояния автомата в настоящий момент времени.

$$\lambda(t) = \psi(x(t), \rho(t))$$

Понятие состояния автомата в момент времени t определяется внутренним состоянием автомата и состоянием входа автомата в тот же момент времени.

$$M(t) \equiv (x(t), \rho(t))$$

Автоматы, для которых функции переходов φ и функции выходов ψ определены на всех парах $\langle \rho, x \rangle$, называются *полностью определенными* или *полными автоматами*. Соответственно, автоматы, для которых функции переходов или функции выходов определены не на всех парах $\langle \rho, x \rangle$, называются *недоопределенными (не полностью определенными) автоматами*. Состояние $M(t)$ автомата недоопределенного на соответствующей паре $\langle \rho, x \rangle$, называется *неиспользованным состоянием автомата*. Если на каком-либо определенном состоянии автомата не определена только функция выходов, то говорят, что ему соответствует безразличное состояние выхода.

Автомат Мура

Для автомата Мура функции переходов и выходов выглядят следующим образом:

$$x(t+1) = \varphi(x(t), \rho(t))$$

$$\lambda(t+1) = \psi(x(t+1)) = \psi(\varphi(x(t), \rho(t))) = \psi'(x(t), \rho(t))$$

Функция выходов для автомата Мура определяется внутренним состоянием автомата.

Для асинхронного автомата.

Поведение определяется следующим уравнением:

$$x(t+1) = \varphi(x(t), \rho(t+1))$$

$$\lambda(t+1) = \psi(x(t+1), \rho(t+1))$$

В асинхронном автомате изменение состояния входа вызывает переход в следующее внутреннее состояние, т.е. внутреннее состояние автомата зависит от состояния входа в этот же момент времени, соответственно состояние выхода автомата зависит от состояния его входа.

8. Способы задания автоматов

Для задания автоматов существуют специальные формализованные языки.

Словесное описание поведения автоматов без рассмотрения внутреннего устройства позволяет описать алгоритм, который задает отображение последовательности состояний входа и состояний выхода автомата. Автомат в этом случае задается как черный ящик, и функции переходов в явном виде не описываются, такие языки получили название *начальные*.

К начальным языкам относятся: язык регулярных выражений, язык предикатных форм, язык логических схем алгоритма. Широкого применения эти языки не нашли. Для описания частного класса автомата оказался удобным начальный язык логических схем алгоритмов НЯЛСА.

Если рассматривать автомат с учетом его внутренних состояний, то необходимо определить функции перехода (из внутреннего состояния x_i во внутреннее состояние x_j не исключая $i=j$). Задание функций выходов означает, что каждой паре $\langle \rho_i, x_i \rangle$ поставлено в соответствие состояние выхода λ_i .

Языки позволяющие таким образом описать автомат называются стандартными языками. К стандартным относятся таблицы переходов, таблицы выходов, матрицы переходов, таблицы включений, графы переходов.

8.1 Табличный способ задания автоматов

Таблица переходов.

Каждая строка (столбец) таблицы переходов соответствует состоянию входов, а каждый столбец (строка) внутреннему состоянию.

	$X1$	$X2$	$X3$	$X4$
$\rho 1$	$X2$	$X1$	$X1$	$X2$
$\rho 2$	$X3$	$X3$	$X1$	$X4$
$\rho 3$	$X4$	$X2$	$X1$	$X1$

Каждая клетка соответствует внутреннему состоянию автомата, в который он должен перейти в следующий момент времени. Если в автомате какое либо состояние не определено, то в соответствующей клетке ставится прочерк. Прочерк означает, что либо это состояние запрещено, либо дальнейшее поведение не определено.

Например, задан автомат, для которого не может быть подано двух одинаковых входных сигналов следующих друг за другом.

	$\rho 1$	$\rho 2$
$X1$	$X1$	$X2$
$X2$	$X3$	-
$X3$	-	$X4$
$X4$	$X1$	-

Таблица выходов.

Функция выхода автомата также может быть задана в виде таблицы. При этом вид таблицы зависит от модели автомата (Мили, Мура).

Для автомата Мили столбцы соответствуют внутренним состояниям, строки – входным сигналам, в ячейках – выходные сигналы.

	$X1$	$X2$	$X3$
$\rho 1$	$\lambda 1$	$\lambda 3$	$\lambda 1$

$\rho 2$	$\lambda 2$	$\lambda 3$	$\lambda 1$
$\rho 3$	$\lambda 1$	$\lambda 2$	$\lambda 1$

Запись $\lambda 1$ означает, что если подать на вход автомата, находящегося в состоянии $X1$, сигнал $\rho 1$, то на выходе будет $\lambda 1$.

Для автомата Мура таблица выходов не строится, т.к. от выходной сигнал не зависит от входного. Для автомата Мура строится совмещенная таблица переходов и выходов

	$\lambda 1$	$\lambda 2$	$\lambda 1$
	$X1$	$X2$	$X3$
$\rho 1$	$X1$	$X2$	$X3$
$\rho 2$	$X1$	-	$X2$
$\rho 3$	-	$X3$	-

Табличный способ для асинхронного автомата

Как и для синхронного автомата, каждая ячейка **таблицы переходов** асинхронного автомата соответствует внутреннему состоянию, в которое перейдет автомат. Это состояние определяется внутренним состоянием в предыдущий момент времени и поданным на вход сигналом.

Устойчивое состояние асинхронного автомата, т.е. состояние, соответствующее устойчивому такту обозначается в таблице переходов круглыми скобками. Неустойчивое состояние - записывается без скобок.

	$\rho 1$	$\rho 2$	$\rho 3$
$X1$	$X2$	$X2$	$X3$
$X2$	$X3$	($X5$)	($X2$)
$X3$	$X6$	-	($X1$)
$X4$	-	$X1$	$X3$

$X5$	$(X1)$	$X6$	$X4$
$X6$	$(X4)$	-	$X6$

Входной сигнал можно менять, когда автомат перешел в новое устойчивое состояние. Если устойчивого состояния нет, то происходит заикливание (неустойчивое состояние).

Если автомат переходит из одного устойчивого внутреннего состояния под воздействием входного сигнала в другое состояние, то входной сигнал можно изменять только при «попадании» автомата в устойчивое состояние. Переход автомата из одного устойчивого состояния в другое устойчивое может осуществляться через несколько неустойчивых состояний.

Таблица выходов. Особенность для асинхронных автоматов состоит в том, что записывается последний выходной сигнал (при устойчивом состоянии). В столбце внутренних состояний записываются только устойчивые состояния. Выходные сигналы соответствуют устойчивому состоянию.

	$\rho 1$	$\rho 2$
$X1$	$\lambda 1$	$\lambda 1$
$X2$	$\lambda 2$	-
$X3$	$\lambda 2$	$\lambda 3$
$X4$	-	$\lambda 1$

8.2. Задание автомата с помощью графа

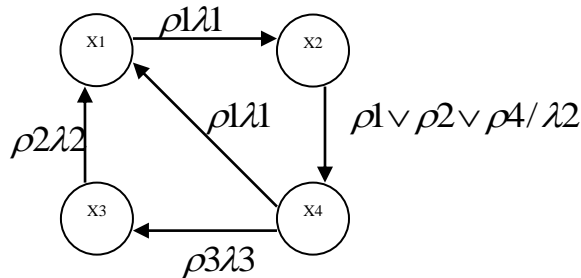
Граф автомата – это связный граф, вершины которого соответствуют внутренним состояниям автомата, а дуги определяют переходы между состояниями.

Для автомата Мили:

Две вершины графа автомата x_i и x_j соединяются дугой, направленной от x_i к x_j , если в автомате имеется переход из состояния x_i в состояние x_j . Дуге графа

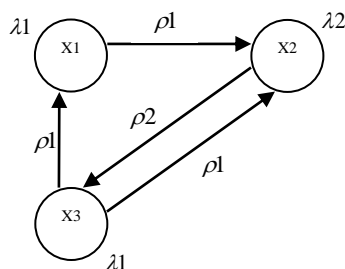
приписывается входной сигнал ρ и выходной сигнал λ . Если переход автомата из состояния x_i в состояние x_j происходит под воздействием нескольких входных сигналов, то дуге приписываются все эти сигналы через знак \vee (дизъюнкции).

Пример:



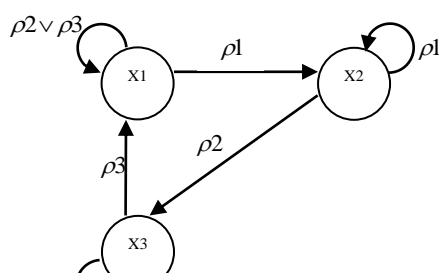
Для автомата Мура:

В автомате Мура выходной сигнал зависит только от внутреннего состояния автомата, поэтому он приписывается вершинам графа, соответствующим определенному внутреннему состоянию автомата. Все остальное как и в автомате Мили.



Для асинхронного автомата.

Особенностью графа асинхронного автомата является следующее: для того чтобы автомат поддерживал себя в устойчивом состоянии до тех пор пока не придет новый входной сигнал, необходимо указать дугу, зацикливающую вершину саму на себя с тем же входным сигналом.



8.3. Матричный способ задания автоматов

Матрица переходов представляет собой квадратную матрицу, строки и столбцы которой соответствуют внутренним состояниям автомата.

Элементы $[i,j]$ указывают состояние входа автомата ρ , при котором он переходит из внутреннего состояния x_i в состояние x_j , и состояние выхода, которое соответствует полному состоянию M . В каждой строке матрицы перехода одно и тоже состояние входа ρ не может встретиться более одного раза. Если автомат из внутреннего состояния x_i переходит в состояние x_j под воздействием нескольких входных сигналов, то в соответствующей ячейке проставляется дизъюнкция состояния входа.

	$X1$	$X2$	$X3$
$X1$	$\rho 1 \lambda 2$	$\rho 2 \lambda 1$	-
$X2$	$\rho 1 \vee \rho 3$ $/ \lambda 2$	-	$\rho 2 \lambda 3$
$X3$	-	-	$\rho 2 \lambda 1$

9. Переход от автомата Мили к автомату Мура и обратно

Абстрактный автомат может работать, как некоторый преобразователь входного слова в слова выходного алфавита

Пусть на вход этого автомата поступает входное слово — Φ (последовательность входных сигналов).

$$\Phi = \rho 1 \rho 2 \rho 1 \rho 3 \rho 2$$

Назовем переменную $\omega = \lambda(a_0; \Phi)$ реакцией автомата, находящегося в состоянии a_0 , на входное слово Φ .

Автомат Мили в ответ на входное слово длиной k выдает последовательность состояний длиной $k+1$ и выходное слово длиной k .

Зададим автомат Мура.

Найдем реакцию автомата Мура на входное слово – Φ . Начальное состояние x_1 :

$$\begin{aligned}\Phi &= \rho_1 \rho_1 \rho_2 \rho_1 \rho_2 \rho_2 \\ & \quad x_1 x_4 x_2 x_1 x_4 x_3 x_5 \\ & \quad \lambda_1 \lambda_1 \lambda_2 \lambda_1 \lambda_1 \lambda_1 \lambda_2\end{aligned}$$

Два автомата S_A и S_B с одинаковыми входным и выходным алфавитом называются - **эквивалентными**, если после установления их в начальное состояние реакции на любое входное слово совпадают.

Можно показать, что для любого автомата Мили существует эквивалентный ему автомат Мура, и наоборот. При написании алгоритма взаимной трансформации часто пренебрегают выходным сигналом, связанным с начальным состоянием.

Рассмотрим переход от автомата Мура к автомату Мили.

Пусть задан автомат Мура:

$$S_A = \langle X_A, P_A, \Lambda_A, \varphi_A, \psi_A, X_{0A} \rangle$$

Необходимо построить автомат Мили, эквивалентный автомату Мура:

$$S_B = \langle X_B, P_B, \Lambda_B, \varphi_B, \psi_B, X_{0B} \rangle$$

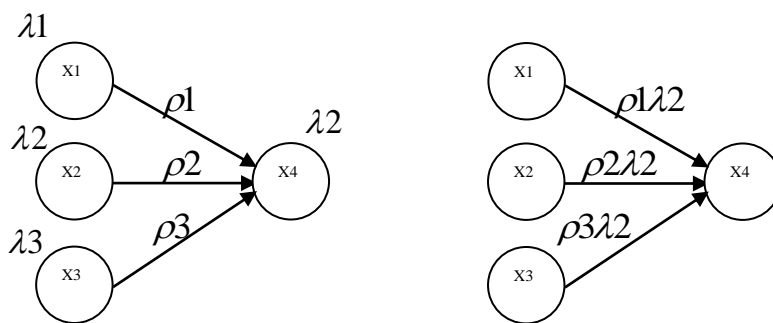
$$P_A = P_B$$

$$\Lambda_A = \Lambda_B$$

Функцию ψ_B определим следующим образом: если в автомате Мура имеются функции $\varphi_A(x_m, \rho_s) = x_s, \psi_A(x_s) = \lambda_g$, то для автомата Мили можно записать следующую функцию выхода: $\psi_B(x_m, \rho_s) = \lambda_g$.

Рассмотрим переход от автомата Мура к автомату Мили с помощью графа:

Для осуществления перехода от автомата Мура к автомату Мили выходной сигнал λ , находящийся в автомате Мура рядом с вершиной, для автомата Мили передается на все дуги, входящие в эту вершину.



Переход от автомата Мура к Мили табличным способом:

Поскольку таблица переходов автомата Мура полностью совпадает с таблицей переходов автомата Мили, то основная проблема при описании автомата Мили табличным способом – это составление таблицы выходов. Таблица выходов автомата Мили получается из таблицы переходов автомата Мура путем замены символа соответствующего внутреннему состоянию автомата Мура символом выходного сигнала.

Для автомата Мура

		$\rho 1$	$\rho 2$	$\rho 3$
$\lambda 1$	X1	X1	X2	X4
$\lambda 2$	X2	X2	X3	X1
$\lambda 1$	X3	X1	X3	X4
$\lambda 3$	X4	X1	X1	X4

Для автомата Мили

	$\rho 1$	$\rho 2$	$\rho 3$
X1	$\lambda 1$	$\lambda 2$	$\lambda 3$
X2	$\lambda 2$	$\lambda 1$	$\lambda 1$
X3	$\lambda 1$	$\lambda 1$	$\lambda 3$
X4	$\lambda 1$	$\lambda 1$	$\lambda 3$

Из самого способа построения автомата Мили очевидно, что он эквивалентен автомату Мура.

Для входной последовательности Φ поведение автоматов S_A и полностью совпадают. По индукции не трудно доказать, что любое входное слово конечной длины, поданное на входы автоматов S_A и S_B , установленных в начальное состояние x_0 вызовет

появление одинаковых выходных слов.

Переход от автомата Мили к автомату Мура. При переходе от автомата Мили к автомату Мура необходимо наложить следующие ограничения: в автомате Мили не должно быть преходящих состояний.

Преходящее состояние - это состояние, в которое при представлении автомата в виде графа не входит ни одна дуга и которое имеет хотя бы одну выходящую дугу.

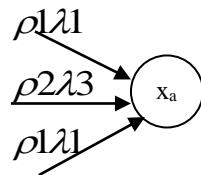
Задан автомат Мили S_A . Необходимо построить автомат Мура S_B .

Алфавиты должны совпадать:

$$P_A = P_B$$

$$\Lambda_A = \Lambda_B$$

Для определения множества X_B каждому состоянию $x_s \in X_A$ поставим соответствующее множество пар вида $x_s = \langle x_s, \lambda_g \rangle$, где λ_g - это выходной сигнал приписанный дуге, входящей в x_s .



$$X_s = \{ \langle x_a, \lambda_1 \rangle, \langle x_a, \lambda_2 \rangle, \langle x_a, \lambda_3 \rangle \}$$

Число элементов в множестве X_S будет равно числу различных выходных сигналов на дугах автомата Мили S_A , входящих в состояние x_a . Число внутренних состояний автомата Мура будет определяться объединением множеств всех X_S .

$$X_B = \bigcup X_S$$

$$X_A \Rightarrow X_B$$

Функция переходов φ_B и функция выходов ψ_B определяется так: если в автомате Мили имеется переход из состояния x_m в состояние x_s под воздействием ρ_i

$$\varphi_A(x_m, \rho_i) = x_s$$

и при этом выдается выходной сигнал λ_j

$$\psi_A(x_m, \rho_i) = \lambda_j,$$

то в автомате Мура будет переход из множества состояний X'_m , порождаемое внутренним состоянием x_m под воздействием входного сигнала ρ_i .

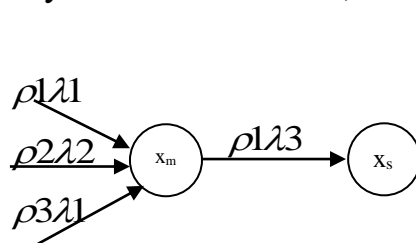
$$\varphi_B(\{X'_m\}, \rho_i) = X'_s.$$

Функция выходов автомата Мура определяется следующим образом

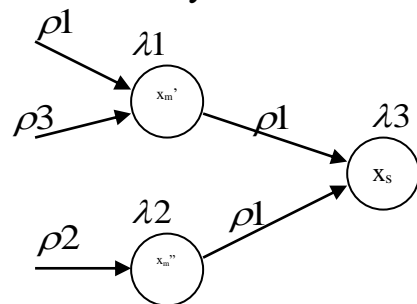
$$\psi_B(x_s') = \lambda_j$$

В качестве начального состояния x_{0B} можно взять любое состояние из множества, которое порождается начальным состоянием x_{0A} .

Т.о. получается автомат S_B , эквивалентный автомату S_A .



Автомат Мили



Автомат Мура

Изложенные методы взаимных транспозиций модели Мили и Мура показывают, что при переходе от автомата Мура к Мили число состояний автомата не меняется, а при обратном переходе число состояний, как правило, возрастает.

Вследствие транзитивности отношения эквивалентности $S_A \equiv S_B \equiv S_A' \Rightarrow S_A \equiv S_A'$. Два автомата Мили также будут эквивалентны, хотя у последнего на два состояния больше. Таким образом, эквивалентные автоматы могут иметь различное число внутренних состояний. В связи с этим возникла задача нахождения минимального автомата в классе эквивалентных между собой автоматов.

10. Этапы синтеза автоматов

В настоящее время процесс синтеза автоматов принято подразделять на следующие этапы:

На *первом* предварительном этапе формируется часто словесно условия работы автомата, то есть определяется условия его взаимодействия с другими устройствами или какими-либо объектами, выявляются необходимые входные и выходные сигналы автомата, их количество, и намечается общий закон появления выходных сигналов в зависимости от воздействия на входы автомата.

При синтезе достаточно сложного автомата его часто разбивают на отдельные блоки, поэтому первый этап иногда называют *этапом блочного синтеза автомата*.

На *втором* этапе синтеза происходит выявление законов функционирования автомата, т.е. определяются функции переходов и выходов. Формальное описание автомата должно быть представлено одним из принятых способов (матрица, граф). Этот этап принято называть *этапом абстрактного синтеза или синтезом абстрактного автомата*.

На данном этапе не интересуются теми физическими элементами, из которых должен состоять автомат.

Не рассматривается, какие конкретные числовые значения могут принимать входные и выходные сигналы и элементы памяти. Важно знать число возможных различных его внутренних состояний, состояний входов и выходов, а также законы изменения внутреннего состояния автомата и выработки выходных сигналов при поступлении той или иной последовательности входных сигналов.

Начало исследования абстрактного синтеза автоматов было положено в работе С.Клини, который предложил так называемый язык регулярных событий для описания автоматов. В дальнейшем абстрактный синтез был усовершенствован В.М. Глушковым. Абстрактный синтез автоматов был проанализирован Б.А. Трахтенбротом с использованием языков высказываний и исчисления предикатов.

Результатом второго этапа синтеза является задание автомата одним из стандартных способов. При этом выделяется объем памяти автомата.

В ряде случаев получают автомат, у которого число внутренних состояний превышает минимальное. В связи с этим, на следующем *третьем* этапе производится **минимизация числа внутренних состояний автомата**.

На *четвертом* этапе синтеза производится кодирование внутренних состояний автомата, называемое **размещением внутренних состояний**. Так же кодируются входные и выходные сигналы.

После кодирования внутреннего состояния автомата, состояний входа и выхода, составляются канонические уравнения. Четвертый этап находится на границе абстрактного и структурного синтеза автомата. На *пятом* этапе синтеза завершается выбор структуры, строится так называемая функциональная схема, состоящая из комбинационной части и автоматов памяти, т.е. происходит **структурный синтез автомата**. При этом синтез автоматов с памятью иногда сводится к синтезу автомата без памяти с помощью понятия одноконтурного эквивалента. В этом случае у автомата обрывается S обратных связей и производится синтез преобразователя с $n+S$ входами и $m+S$ выходами.

Шестой этап синтеза включает проведение электрического и других расчетов элементов схем, составление принципиальной схемы устройства и моделирование работы автомата с целью проверки его работоспособности.

На *седьмом* этапе осуществляется составление монтажных схем и технической документации.

Первые пять этапов синтеза принято называть логическим синтезом автомата (проектированием). Шестой и седьмой техническим синтезом автомата. Разделение процесса синтеза автомата на данные семь этапов, с одной стороны, облегчает процессы синтеза автомата. С другой стороны, может привести к усложнению структуры автомата. Следовательно, на каждом этапе стараются учесть его влияние на последующие этапы.

Для ускорения процесса синтеза автомата проводятся работы по его автоматизации. С этой целью разработаны специальные формальные языки, предназначенные для представления алгоритма синтеза на ЭВМ.

11. Минимизация числа внутренних состояний автомата

Под *минимизацией* числа внутренних состояний автомата понимается процесс, целью которого является получение автомата имеющего минимальное число внутренних состояний среди всех автоматов, реализующих заданные условия работы. Условиями работы автомата будем называть множество пар $\langle \rho_i, \lambda_j \rangle$ - входное слово и реакция автомата.

Будем говорить, что две последовательности $A=a_1.....a_i$ и $B=b_1.....b_i$ являются непротиворечивыми, если в них не содержится ни одной пары элементов $[a_i, b_i]$ таких, что $a_i = (\rho_i, \lambda_j)$ и $b_i = (\rho_i, \lambda_k)$.

Говорят, что автомат реализует заданные условия работы, то есть является *реализующим* автоматом, если при поступлении на его вход входной последовательности на выходе автомата будет выдаваться выходная последовательность, непротиворечивая последовательности, задаваемой условиями работы автомата. Два автомата, реализующие одни и те же условия работы, будем называть *эквивалентными автоматами*. Одни и те же условия работы могут реализовать несколько эквивалентных автоматов, имеющие различное число внутренних состояний. Задача минимизации состоит в выявлении среди этих эквивалентных автоматов минимального. Поскольку, число внутренних состояний автомата определяет ёмкость памяти, процесс минимизации числа внутренних состояний автомата приводит к минимизации числа элементов памяти.

Имеется соотношение:

$$S = \lceil \log_M N \rceil ,$$

где M – число внутренних состояний, которые может хранить ЭП,

N – количество внутренних состояний автомата.

Сокращение числа внутренних состояний целесообразно производить, т.к. это приводит (в большинстве случаев) к уменьшению числа элементов памяти и упрощению структуры логического преобразователя. Однако необходимо

помнить, что сокращение числа внутренних состояний до минимального может привести к возникновению гонок.

В настоящее время существует две группы методов построения автомата с минимальным числом внутренних состояний. Для первой группы характерно то, что сначала берётся автомат с одним внутренним состоянием, а затем производится увеличение числа его внутренних состояний до тех пор, пока он не станет реализующим автоматом.

Во второй группе методов берётся заведомо реализующий автомат с каким-то числом внутренних состояний, а затем производится уменьшение внутренних состояний до тех пор, пока автомат при числе внутренних состояний $(N-1)$ не станет нереализующим.

Методы минимизации первой группы нашли применение при задании автомата таблицами включений. Методы второй группы применяются, когда автомат задан другими стандартными способами.

Для минимизации числа внутренних состояний недоопределенного автомата в настоящее время отсутствует простой алгоритмизированный метод, дающий возможность получить минимальный недоопределенный автомат. Перебор всех возможных вариантов с целью выбора из них минимального автомата становится невозможным, даже для автомата с относительно небольшим числом внутренних состояний. В связи с трудностью синтеза минимального недоопределенного автомата в настоящее время в данной области наметились две тенденции:

1. Разработка приближённых, но алгоритмизуемых методов, которые не гарантируют построение минимального реализующего автомата, но позволяют запрограммировать данный процесс. Метод Е.А. Бутакова.

2. Разработка методов минимизации отдельных частных классов недоопределённых автоматов, для которых имеется возможность построения минимального автомата.

12. Минимизация полностью определённых автоматов.

Рассмотрим метод, предположенный Ауфенкампом и Хоном.

Основная идея этого метода состоит в разбиении всех состояний исходного абстрактного автомата на попарно непересекающиеся классы эквивалентных состояний и замене каждого класса эквивалентности одним состоянием. Получающийся в результате минимальный автомат имеет столько же состояний, на сколько классов эквивалентности разбивается множество состояний исходного автомата.

Два состояния абстрактного автомата x_m и x_s называются эквивалентными, если выходная функция $\psi(x_m, \Phi) = \psi(x_s, \Phi)$ для всех возможных входных слов Φ . Иначе состояния называются неразличимыми.

Более слабой формой эквивалентности является k -эквивалентность:

$\psi(x_m, \Phi_k) = \psi(x_s, \Phi_k)$ для всех возможных слов длиной k .

Введённые отношения эквивалентности симметричны, транзитивны и рефлексивны. Следовательно, они могут быть использованы для разбиения множества состояний абстрактного автомата на попарно непересекающиеся классы – классы эквивалентности. Соответствующие разбиения на классы эквивалентности и k -эквивалентности будем обозначать через π и π_k . Разбиение позволяет определить избыточные элементы в множестве внутренних состояний. Если каждый класс эквивалентности содержит только одно состояние, то множество внутренних состояний не сократимо.

Рассмотрим алгоритм минимизации числа внутренних состояний полностью определённого автомата:

1. Находятся последовательные разбиения π_1, π_2 множества X до тех пор, пока на каком-то $(k+1)$ шаге не окажется, что это разбиение ничем не отличается от предыдущего. Доказано, что в этом случае $(\pi_k = \pi_{k+1})$ π_k есть необходимое нам разбиение, и дальнейшее сокращение числа внутренних состояний автомата невозможно.
2. В каждом классе эквивалентности разбиения π выбирается по одному элементу, который образует множество X' .

$$S = \langle X, P, \Lambda, \varphi, \psi, X_0 \rangle$$

$$S' = \langle X', P, \Lambda, \varphi', \psi', X_0' \rangle$$

3. Функции переходов φ' и функция выходов ψ' для автомата S' определяются на множестве оставшихся внутренних состояний и множестве входных сигналов. Для этого в таблице переходов вычеркиваются столбцы, соответствующие состояниям, не вошедшим в множество X' , а в оставшихся столбцах таблицы переходов все состояния заменяются на эквивалентные из множества X' . В таблице выходов столбцы вычёркиваются.
4. В качестве начального состояния X_0' выбирается одно из состояний, эквивалентных X_0 . На практике лучше взять само X_0 .

ПРИМЕР:

Минимизация полного автомата Мили:

Таблица переходов:

	X_1	X_2	X_3	X_4	X_5	X_6	X_7	X_8	X_9	X_{10}	X_{11}	X_{12}
ρ_1	X_{10}	X_{12}	X_5	X_7	X_3	X_7	X_3	X_{10}	X_7	X_1	X_5	X_2
ρ_2	X_5	X_8	X_6	X_{11}	X_9	X_{11}	X_6	X_4	X_6	X_8	X_9	X_8

Таблица выходов:

	X_1	X_2	X_3	X_4	X_5	X_6	X_7	X_8	X_9	X_{10}	X_{11}	X_{12}
ρ_1	λ_1	λ_1	λ_2	λ_2	λ_1	λ_2	λ_1	λ_1	λ_2	λ_2	λ_2	λ_2
ρ_2	λ_2	λ_2	λ_1	λ_1	λ_2	λ_1	λ_2	λ_2	λ_1	λ_1	λ_1	λ_1

$$\pi_1 = \{X_1, X_2, X_5, X_7, X_8\} B_1,$$

$$\{X_3, X_4, X_6, X_9, X_{10}, X_{11}, X_{12}\} B_2$$

	X_1	X_2	X_5	X_7	X_8		X_3	X_4	X_6	X_9	X_{10}	X_{11}	X_{12}
ρ_1	B_2	B_2	B_2	B_2	B_2		B_1	B_1	B_1	B_1	B_1	B_1	B_1
ρ_2	B_1	B_1	B_2	B_2	B_2		B_2	B_2	B_2	B_2	B_1	B_2	B_1

$$\pi_2 = \{X_1, X_2\} C_1,$$

$$\{X_5, X_7, X_8\} C_2,$$

$$\{X_3, X_4, X_6, X_9, X_{11}\} C_3,$$

$$\{X_{10}, X_{12}\} C_4,$$

	X_1	X_2		X_5	X_7	X_8		X_3	X_4	X_6	X_9	X_{11}		X_{10}	X_{12}
ρ_1	C_4	C_4		C_3	C_3	C_4		C_2	C_2	C_2	C_2	C_2		C_1	C_1
ρ_2	C_2	C_2		C_3	C_3	C_3		C_3	C_3	C_3	C_3	C_3		C_2	C_2

$$\pi_3 = \{X_1, X_2\} D_1,$$

$$\{X_5, X_7\} D_2$$

$$\{X_8\} D_3$$

$$\{X_3, X_4, X_6, X_9, X_{11}\} D_4$$

$$\{X_{10}, X_{12}\} D_5$$

Соответственно, получили разбиение, для которого дальнейшее разбиение невозможно.

Произвольно выбираем из каждого класса по одному состоянию и формируем множество

$$X' = \{X_1, X_5, X_8, X_3, X_{10}\}$$

	X_1	X_3	X_5	X_8	X_{10}
ρ_1	X_{10}	X_5	X_3	X_{10}	X_1
ρ_2	X_5	X_3	X_3	X_3	X_8

	X_1	X_3	X_5	X_8	X_{10}
ρ_1	λ_1	λ_2	λ_1	λ_1	λ_2
ρ_2	λ_2	λ_1	λ_2	λ_2	λ_1

Особенности минимизации для автомата Мура.

	λ_1	λ_1	λ_3	λ_3	λ_3	λ_2	λ_3	λ_1	λ_2	λ_2	λ_2	λ_2
	X_1	X_2	X_3	X_4	X_5	X_6	X_7	X_8	X_9	X_{10}	X_{11}	X_{12}
ρ_1	X_{10}	X_{12}	X_5	X_7	X_3	X_7	X_3	X_{10}	X_7	X_1	X_5	X_2
ρ_2	X_5	X_7	X_6	X_{11}	X_9	X_{11}	X_6	X_4	X_6	X_8	X_9	X_8

При минимизации автомата Мура вводится понятие *0-эквивалентности* и, соответственно, разбиения множества внутренних состояний на 0-

эквивалентные классы. 0- эквивалентными называются любые одинаково отмеченные состояния автомата Мура. Если два 0- эквивалентных состояния любыми входными сигналами переводятся в два 0- эквивалентных состояния, то они называются 1-эквивалентными состояниями.

$$\pi_0 = \{X1, X2, X8\} A_1$$

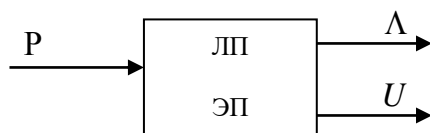
$$\{X3, X4, X5, X7\} A_2$$

$$\{X6, X9, X10, X11, X12\} A_3$$

13. Совмещённая модель автомата (С-автомата)

Под *абстрактным С-автоматом* будем понимать математическую модель С-устройства, определяемую множеством из восьми элементов.

$$S = \langle X, P, \Lambda, U, \varphi, \psi_1, \psi_2, X_0 \rangle$$



С-автомат можно представить в виде устройства с одним входом и двумя выходами. Отличие С-автомата от автоматов Мили и Мура состоит в том, что он реализует функции выходов, как для автомата Мили, так и для автомата Мура.

$$x(t+1) = \varphi(x(t), \rho(t))$$

$$\lambda(t+1) = \psi_1(x(t+1), \rho(t+1))$$

$$u(t+1) = \psi_2(x(t+1))$$

Очевидно, что от С-автомата легко перейти к автомату Мили или к автомату Мура.

В ряде случаев, оказывается, удобно использовать С-автомат для проектирования различных узлов вычислительной техники.

Для задания С-автомата также используются табличные, матричный и графовый способы. Таблица переходов С-автомата аналогична таблице переходов автомата Мили. В таблице выходов С-автомата каждое состояние отмечено соответствующим ему выходным сигналом типа U.

ПРИМЕР:

Таблица переходов:

	$X1$	$X2$	$X3$	$X4$
$\rho 1$	$X2$	$X3$	$X1$	$X2$
$\rho 2$	$X4$	$X2$	$X1$	$X3$

Таблица выходов:

	$U1$	$U2$	$U1$	$U3$
	$X1$	$X2$	$X3$	$X4$
$\rho 1$	$\lambda 1$	$\lambda 2$	$\lambda 1$	$\lambda 2$
$\rho 2$	$\lambda 2$	$\lambda 2$	$\lambda 1$	$\lambda 2$

При задании С-автомата в виде графа выходные сигналы типа λ приписываются дугам графа, а сигнал типа u приписывается вершинам состояний, которым эти сигналы соответствуют.

Нетрудно доказать, что к полностью определённом С-автомату можно применить алгоритм минимизации Ауфенкампа.

Под 1-эквивалентным состоянием С-автомата необходимо понимать состояния, которые одинаково отмечены и имеют одинаковые столбцы в таблице выходов. Дальнейшее разбиение на классы эквивалентности происходит аналогично минимизации автомата Мили.

14. Структурный синтез автомата

14.1. Канонический метод структурного синтеза автомата

Вслед за этапом абстрактного синтеза автомата, заканчивающимся минимизацией числа внутренних состояний, следует этап структурного синтеза.

Целью его является построение схемы реализующего автомата из логических элементов заданного типа. Если абстрактный автомат был лишь математической моделью дискретной устройства, то в структурном автомате необходимо учитывать структуру входных и выходных сигналов, а также

внутреннее устройство автомата на уровне структурной или функциональной (логической) схемы. Структурным синтезом занимается структурная теория автоматов, основной задачей которого является нахождение общих приёмов и методов построения структурных схем автомата на основе композиции элементарных автоматов, принадлежащих заранее заданному конечному числу типов.

Под композицией элементарных автоматов будем понимать следующее. Пусть заданы элементарные автоматы S_1, S_2, \dots, S_k . Произведем объединение элементарных автоматов в систему совместно работающих устройств. Для этого введём некоторое конечное множество узлов. Узлы разделяются на внешние и внутренние.

Композиция автомата состоит в том, что совместная работа всех элементарных автоматов от сигнала, поданного на один из внешних входных узлов, начинается одновременно. Входной сигнал запускает работу всей системы в целом, отождествляя внутренние узлы композиции автомата. После проведённых отождествлений всех узлов система автомата превращается так в называемую в схему автоматов или сеть автоматов. На внешние входные узлы подаётся набор входных сигналов (структурный входной сигнал схемы) и со всех внешних выходных узлов снимается набор выходных сигналов (структурный выходной сигнал).

При построении схемы автоматов должно выполняться условие корректности. Т.е. все входящие в композицию элементарные автоматы должны иметь одинаковые структурные входные и выходные алфавиты и должны работать в одном и том же автоматном времени.

В настоящее время наиболее распространённым структурным алфавитом является двоичная система счисления. Для двоичного алфавита разработан удобный аппарат булевых функций, позволяющий производить многочисленные операции над схемами автоматов формально.

При построении структурного автомата предварительно выбираются элементарные автоматы, из которых путем их композиции строится структурная схема автомата (Мили, Мура или С-автомата).

Если решение задачи структурного синтеза существует, то говорят, что заданная система элементарных автоматов структурно полна. В настоящее время нет сколько-нибудь эффективных методов решения основной задачи структурного синтеза при любом наборе структурно полных схем элементарных автоматов.

Обычно применяется так называемый канонический метод структурного синтеза, при этом используются элементарные автоматы специального вида.

1. Автоматы с памятью, имеющие более одного внутреннего состояния – нетривиальные автоматы.
2. Автомат без памяти – логические элементы.

Автоматы первого типа называются *элементарными автоматами памяти*. Автоматы второго типа называются *комбинационной схемой* или *логическими элементами*. Теоретическим обоснованием канонического метода структурного синтеза автомата является теорема о структурной полноте.

Всякая система элементарных автоматов, которая содержит автомат Мура с нетривиальной памятью, обладающий полной системой переходов и полной системой выходов и какую-либо функционально полную систему логических элементов является структурно-полной системой автоматов.

Существует общий конструктивный прием, позволяющий свести задачу структурного синтеза произвольного автомата к задаче синтеза комбинационной схемы. Результатом канонического метода структурного синтеза является система логических уравнений, выражающая зависимость выходных сигналов автомата и сигналов, подаваемых на входы запоминающих элементов памяти, от сигналов, приходящих на вход автомата и сигналов, снимаемых с выходов элементов памяти. Эти уравнения называются *каноническими*. Для правильной работы схемы автомата нельзя разрешать, чтобы сигналы на входе запоминающих элементов непосредственно участвовали в формировании выходных сигналов.

В связи с этим запоминающими элементами должен быть автомат Мура, а не Мили. Таким образом, структурно полная система элементов автомата должна содержать хотя бы один полный автомат Мура.

Полнота системы переходов автомата Мура означает, что для любой пары состояний $\langle b_m, b_s \rangle$ найдётся входной сигнал, переводящий автомат из состояния b_m в состояние b_s . Полнота системы выходов автомата Мура состоит в том, что каждому состоянию автомата поставлен в соответствие свой особый выходной сигнал, отличный от выходных сигналов других состояний. В связи с тем, что выходные сигналы для полного автомата Мура эквивалентны его внутренним состояниям, можно использовать одни и те же обозначения, как для внутреннего состояния автомата, так и для состояния выхода.

14.2. Структурный синтез С-автомата

Канонический метод структурного синтеза С-автомата предполагает составление структурной схемы автомата, состоящей из трех частей: модуля памяти и двух комбинационных схем.

Модуль памяти С-автомата строится из элементарных полных автоматов Мура. Т.о. после выбора элементов памяти, каждое состояние С-автомата представляется (кодируется) в структурном автомате вектором $a_m = (l_1 \dots l_i)$.

Если все элементы памяти одинаковы, то их число определяется следующим образом

$$I = \lceil \log_2 M \rceil ,$$

Переход абстрактного автомата из состояния a_m в состояние a_s под действием входного сигнала z_f с выдачей выходного сигнала λ_i

$$a_m \xrightarrow{z_f / \lambda_i} a_s$$

соответствует в структурном автомате переходу из состояния

$$(l_1 \dots l_m) \xrightarrow{(x_1 \dots x_f) / \{(\omega_1 \dots \omega_i)(u_1 \dots u_i)\}} (l_1 \dots l_s)$$

После выбора типа элементов памяти и кодирования состояния автоматов, следует этап синтеза комбинационной схемы, реализующий следующие функции:

$$\begin{aligned} Y_1 &= Y_1(\tau_1 \dots \tau_I; x_1 \dots x_L) \\ &\dots \\ Y_N &= Y_N(\tau_1 \dots \tau_I; x_1 \dots x_L) \end{aligned}$$

$$r_1 = r_1(\tau_1 \dots \tau_I)$$

...

$$r_D = r_D(\tau_1 \dots \tau_I)$$

Каноническое уравнение функции возбуждения ЭП

$$\alpha_1 = \alpha_1(\tau_1 \dots \tau_I; x_1 \dots x_L)$$

...

$$\alpha_I = \alpha_I(\tau_1 \dots \tau_I; x_1 \dots x_L)$$

При построении функции возбуждения ЭП необходимо использовать функцию выходов ЭП, ставящую в соответствие каждой паре состояний $\langle b_m, b_s \rangle$ элементарного автомата памяти входной сигнал, который переводит автомат памяти из состояния b_m в состояние b_s .

b_m	q_i	b_s
B_1	Q_1	B_1
B_1	Q_2	B_2
B_1	Q_3	B_3
B_2	Q_3	B_1
B_2	Q_1	B_2
B_2	Q_2	B_3
B_3	Q_2	B_1
B_3	Q_3	B_2
B_3	Q_1	B_3

ПРИМЕР:

Необходимо синтезировать частичный С-автомат, заданный следующей таблицей переходов и выходов.

	A_1	A_2	A_3
Z_1	A_2	-	A_1
Z_2	A_3	A_1	-
Z_3	A_2	A_3	A_3

	U_1	U_2	U_1
	A_1	A_2	A_3
Z_1	W_1	-	W_2
Z_2	W_1	W_3	-
Z_3	W_2	W_1	W_3

Таблица переходов автомата:

	a_1	a_2	a_3
z_1	a_2	-	a_1
z_2	a_3	a_1	-
z_3	a_2	a_3	a_3

Таблица выходов автомата:

	u_1	u_2	u_3
	a_1	a_2	a_3
z_1	w_1	-	w_2
z_2	w_4	w_3	-
z_3	w_2	w_1	w_3

В качестве элементарного автомата памяти будем использовать автомат Мура, заданный следующим образом:

Таблица переходов автомата памяти:

	b_1	b_2
q_1	b_1	b_2
q_2	b_2	b_1

Так как в элементарном автомате памяти (ЭАП) два входных сигнала q_1 и q_2 и два выходных сигнала b_1 и b_2 , то в структурном автомате памяти будут один входной и один выходной сигналы.

Произведем кодирование входных сигналов ЭАП (α - функция возбуждения памяти):

	α
q_1	0
q_2	1

Произведем кодирование выходных сигналов с элементов памяти:

	τ
b_1	0
b_2	1

Используя данную кодировку, заполним таблицу переходов автомата памяти:

	0	1
0	0	1
1	1	0

Поскольку у абстрактного С-автомата имеется три внутренних состояния a_1 , a_2 и a_3 , то структурный С-автомат должен иметь два элемента памяти для кодирования.

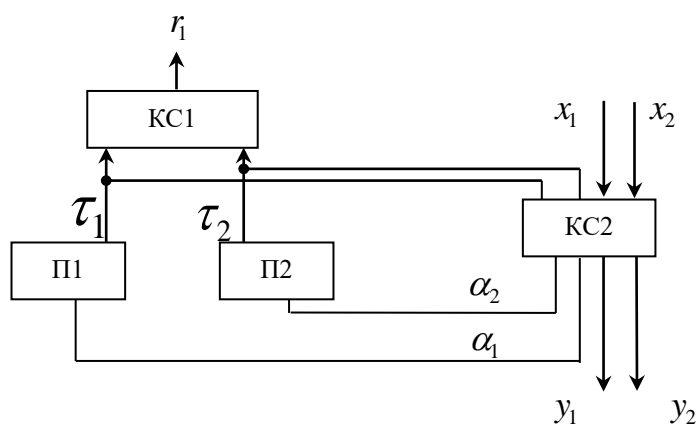
Поскольку абстрактный С-автомат имеет три входных (z_1 , z_2 и z_3) и четыре выходных сигнала типа 1 (w_1 , w_2 , w_3 и w_4), то в структурном автомате необходимо иметь два входных и два выходных канала для сигнала типа 1.

Для реализации двух выходных сигналов типа 2 необходим один выходной канал комбинационной схемы 2.

Таким образом, структурный автомат будет содержать:

1. Два элемента памяти (Π_1 и Π_2);
2. Два входных канала (x_1 и x_2);
3. Два выходных канала типа 1 (y_1 и y_2);
4. Выходной канал типа 2 (r_1).

Тогда структурная схема автомата, реализующего данные функции имеет следующий вид.



Произвольно закодируем набор входных, выходных сигналов и внутреннего состояния:

	τ_1	τ_2
a_1	0	0
a_2	0	1
a_3	1	1

	x_1	x_2
z_1	0	0
z_2	0	1
z_3	1	0

	y_1	y_2
w_1	1	0
w_2	0	0
w_3	1	1
w_4	0	1

	r
u_1	1
u_2	0

Заменяем теперь таблицы переходов и выходов абстрактного автомата с учетом принятой кодировки.

Таблица переходов структурного С-автомата:

	00	01	11
00	01	-	00
01	11	00	-
10	01	11	11

Таблица выходов:

	1	0	1
	00	01	11
00	10	-	00
01	01	11	-
10	00	10	11

Таким образом, после выбора элементов памяти и кодирования состояний автомата синтез структурного автомата сводится к синтезу комбинационных схем КС1 и КС2.

Схема КС1 должна реализовать следующие функции:

$$y_1 = y_1(\tau_1, \tau_2, x_1, x_2);$$

$$y_2 = y_2(\tau_1, \tau_2, x_1, x_2);$$

$$\alpha_1 = \alpha_1(\tau_1, \tau_2, x_1, x_2);$$

$$\alpha_2 = \alpha_2(\tau_1, \tau_2, x_1, x_2).$$

Комбинационная схема КС2 должна реализовать функцию $r_1 = r_1(\tau_1, \tau_2)$.

Функции y_1 и y_2 можно получить непосредственно из отмеченной таблицы выходов структурного С-автомата:

$$y_1 = \overline{\tau_1} \overline{\tau_2} \overline{x_1} \overline{x_2} \vee \overline{\tau_1} \overline{\tau_2} \overline{x_1} x_2 \vee \overline{\tau_1} \tau_2 \overline{x_1} \overline{x_2} \vee \tau_1 \tau_2 \overline{x_1} \overline{x_2} = 0 \vee 5 \vee 6 \vee 14$$

$$y_2 = \overline{\tau_1} \overline{\tau_2} \overline{x_1} \overline{x_2} \vee \overline{\tau_1} \tau_2 \overline{x_1} \overline{x_2} \vee \tau_1 \tau_2 \overline{x_1} \overline{x_2} = 1 \vee 5 \vee 14$$

$$r = \overline{\tau_1} \overline{\tau_2} \vee \tau_1 \tau_2 = 0 \vee 3$$

Функции возбуждения элементарных автоматов памяти П1 и П2 будут соответствовать таблице переходов структурного С-автомата с учетом выбранных ЭП.

Таблица переходов элементарного автомата памяти:

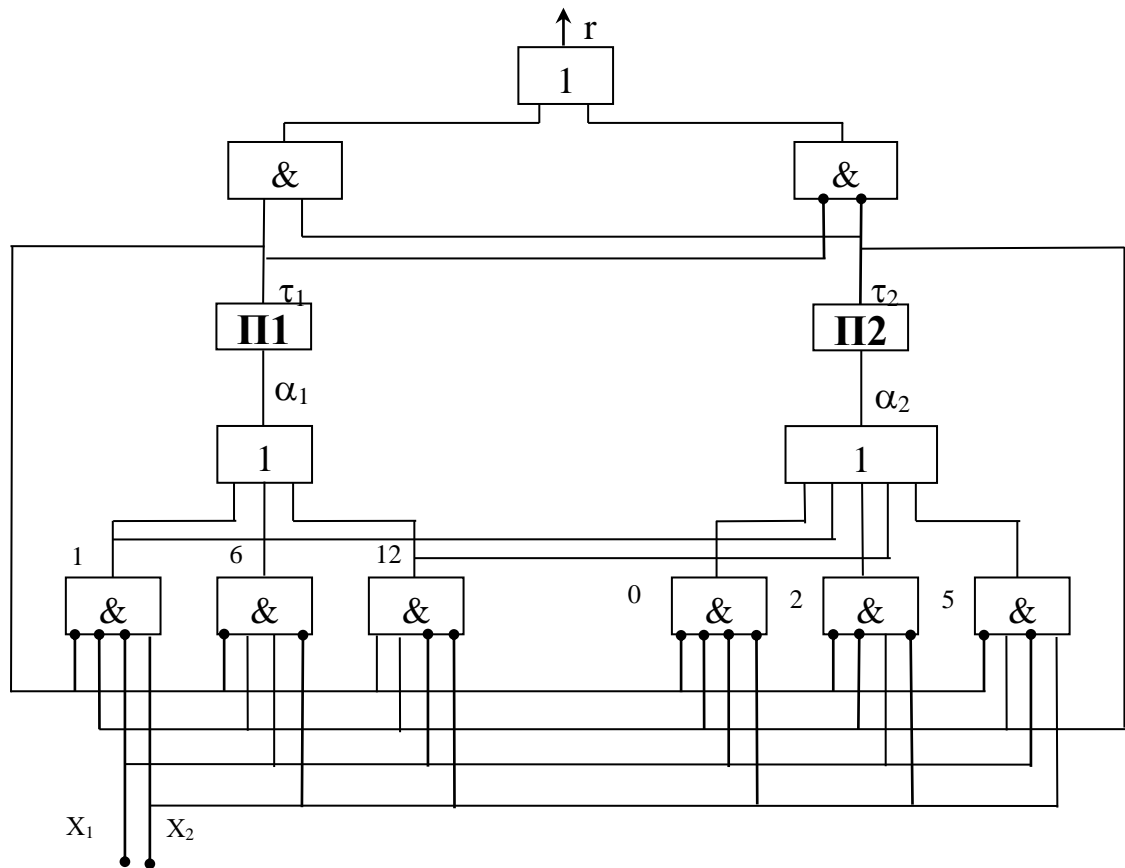
	0	1
0	0	1
1	1	0

$\tau_{\text{исх}}$	α	$T_{\text{вых}}$
0	0	0
0	1	1
1	0	1

Модифицированная таблица переходов:

	00	01	11
00	01	-	11
01	11	01	-
10	01	10	00

Строим функциональную схему структурного автомата (базис и, или, не):



Для минимизации полученных канонических уравнений можно использовать наборы переменных, на которых соответствующие функции не определены. Сформулируем правило учета недоопределенности:

1. Если некоторый код не используется для кодирования внутренних состояний автомата, то на всевозможных наборах, получаемых из этого состояния, при всех значениях входных сигналов все компоненты функции возбуждения элементов памяти и функции выхода не определены;

2. Если некоторый код не используется для кодирования входных сигналов, то на всех наборах, полученных из значений этого кода, все компоненты функции возбуждения элементов памяти и функции выхода типа 1 не определены.

3. Если для некоторой пары $\langle a_i, z_j \rangle$ функция переходов не определена, то на всех наборах всевозможных значений $\langle a_i, z_i \rangle$ все компоненты функции выходов не определены.

15. Кодирование состояний автомата

Кодирование внутренних состояний автомата заключается в сопоставлении каждому состоянию автомата набора значений соответствующих состояний элементарных автоматов памяти.

$I \geq \lceil \log_2 M \rceil$, т.е. переход автомата из одного состояния в другое осуществляется за счёт изменения внутренних состояний элементов памяти.

При функционировании автомата могут появиться так называемые состязания, которые возникают вследствие того, что

1. Элементарные автоматы памяти имеют различные, хотя достаточно близкие, времена срабатывания.
2. Различные времена задержки сигналов (функции возбуждения), поступающих на входы элементов памяти.

Если при переходе автомата из одного состояния в другое должны изменить свои состояния сразу несколько элементов памяти, то между ними начинаются состязания. Элемент памяти, который изменил своё состояние раньше других, может через цепь обратной связи (КС1) изменять сигналы на входах других запоминающих элементов. Это может привести к переходу автомата в состояние непредусмотренное графом его функционирования.

В процессе перехода из состояния a_m в состояние a_s под действием входного сигнала z_f автомат может оказаться в некотором промежуточном состоянии a_k или a_l . Если затем при том же самом входном сигнале автомат из состояния a_k или a_l перейдёт в состояние a_s , то такие состязания являются допустимыми или некритическими.

Если же под действием входного сигнала z_f автомат перейдёт из промежуточного состояния a_k в некоторое состояние a_j , то правильность работы автомата будет нарушена и такие состязания называются критическими или гонками.

Существует несколько способов ликвидации гонок. Рассмотрим лишь основные из них.

Аппаратные способы:

Первый способ состоит в тактировании (стробировании) входных сигналов автомата импульсами определенной длительности.

Предполагается, что кроме входных каналов (x_1, \dots, x_l) имеется ещё один входной канал c от генератора синхроимпульсов.

Таким образом, входным сигналом при переходе из a_m в a_s будет не z_f , а $z_f \& c$

$$a_m \xrightarrow{z_f \cdot c} a_s$$

Если длительность импульса c меньше самого короткого пути прохождения сигналов обратной связи, то к моменту перехода в промежуточное состояние a_k входной сигнал $c=0$, следовательно, переход не может осуществиться.

Другой способ ликвидации гонок заключается в использовании двойной памяти. В этом случае каждый элемент памяти дублируется, причём перезапись из первой ступени элементов памяти во вторую происходит в отсутствие синхронизирующего импульса.

Наряду с аппаратными способами устранения гонок используют специальные методы кодирования.

15.1. Метод противогоночного кодирования состояний автомата

Пусть существуют две пары двоичных кодов (α, β) и (γ, δ) . Пары называют развязанными, если i -ый разряд кода принимает одно значение в паре (α, β) и противоположное в паре (γ, δ) , иначе пары называются связными.

Доказана теорема, что в автомате состояние которого закодировано двоичными кодами, гонки отсутствуют, тогда и только тогда, когда для любых двух переходов $a_m \xrightarrow{z_f} a_s$, $a_k \xrightarrow{z_f} a_l$, происходящих под действием одного и того же входного сигнала, соответствующие пары кодов состояний развязаны.

Алгоритм противогоночного кодирования заключается в последовательном развязывании пар переходов.

Алгоритм:

Пусть имеются состояния автомата - (a_m, a_s) , (a_k, a_l)

Значение i -го разряда в данных состояниях - (α, β) , (γ, δ) , $i=1, I$.

Присвоить $i:=1$.

1. Если при некотором i значение i -ого разряда кодов $\alpha, \beta, \gamma, \delta$ образуют набор соответственно 0011 или 1100, то переходим к пункту 8, иначе к пункту 3.
2. Если при некотором i значение i -ого разряда $\alpha, \beta, \gamma, \delta$ образуют один из наборов *011, 0*11, 00*1, 001*, *01*, **11, 0**1, 00**, *0*1, 0*1*, ***1, 0***, *0**, **1*, ****, то переходим к пункту 4, иначе к пункту 5.
3. Доопределить неопределённые значения i -ого разряда до набора 0011, перейти к пункту 8.
4. Если значения i -ого разряда $\alpha, \beta, \gamma, \delta$ образуют один из наборов *100, 1*00, 110*, ..., то перейти к пункту 6. Иначе к пункту 7.
5. Доопределить неопределённые значения i -ого разряда до набора 1100 и перейти к пункту 8.
6. Дополнить коды состояния автомата одним неопределённым разрядом и перейти к пункту 2.
8. Пары переходов (a_m, a_s) , (a_k, a_l) развязаны.

ПРИМЕР:

	a1	a2	a3	a4	a5	a6	a7
Z1	a2	a2	a4	a4	a6	a6	-
Z2	a1	a3	a3	a1	a3	-	-
Z3	-	a5	a7	-	a5	-	a7

Z1: (a1,a2) (a2,a2) (a3,a4) (a4,a4) (a5,a6) (a6,a6)

(a1,a2) (a2,a2) – Состязания некритические

(a1,a2) (a3,a4) – Состязания критические

Рассмотрим всевозможные пары Z1.

Вводим код длиной 1.

	$\tau 1$	$\tau 2$	$\tau 3$	$\tau 4$
a1	0	1	1	-
a2	0	0	0	0
a3	1	0	0	1
a4	1	0	1	-
a5	1	1	0	0
a6	1	1	-	-
a7	-	-	-	1

Сейчас все пары развязаны для z1.

Аналогично проверяем все пары Z2, Z3.

Z2=(a1,a1) (a2,a3) (a3,a3) (a4,a1) (a5,a3)

Z3=(a2,a5) (a3,a7) (a5,a5) (a7,a7)

Длина кода, полученная в результате применения алгоритма противогоночного кодирования, в большинстве случаев оказывается не минимальной, поскольку при введении нового разрядного кода могут развязаться пары переходов, которые уже были развязаны ранее. В связи с этим необходимо минимизировать длину получаемых кодов. Для этого исключается один из разрядов кодов, в результате чего некоторые пары могут оказаться связанными, поэтому применим алгоритм развязывания переходов и пробуем

исключить еще один разряд и т.д. до тех пор, пока изменить длину кода возможно.

	$\tau 1$	$\tau 2$	$\tau 3$	$\tau 4$	$\tau 5$
a1	0	1	1	-	0
a2	0	0	0	0	0
a3	1	0	0	1	-
a4	1	0	1	-	-
a5	1	1	0	0	1
a6	1	1	-	-	1
a7	-	0	-	1	-

Дальнейшее вычёркивание приведёт к добавлению $\tau 6$. Получим код, в котором все пары развязаны и длина этого кода минимальна.

15.2. Соседнее кодирование состояний автомата

Соседними состояниями первого порядка называются состояния, которые можно соединить ребром.

Соседними состояниями первого рода называются состояния, которые под действием одного и того же входного сигнала переходят в одно и то же состояние.

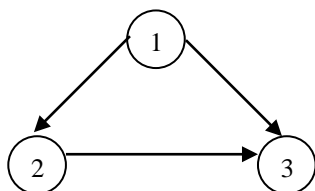
Два состояния, в которые переходят под действием одного и того же входного сигнала состояния, являющиеся соседями первого рода, называются соседями второго рода.

Соседними кодами называются наборы значений, имеющие противоположные значения только в одном разряде, т.е. коды с кодовым расстоянием 1.

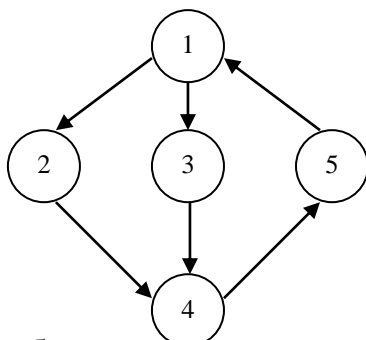
При соседнем кодировании любые два состояния, связанные дугой на графе автомата кодируются наборами, отличающимся состоянием лишь одного элемента памяти.

Существуют ограничения соседнего кодирования:

1. В графе автомата не должно быть циклов с нечётным числом вершин.



2. Два соседних состояния второго порядка не должны иметь более двух состояний, лежащих между ними.



Наиболее просто определить соседей, как первого, так и второго рода, используя обратную таблицу переходов, в отличие от прямой таблицы в её ячейки проставляются состояния, из которых автомат переходит в состояние, указанное в заголовке столбца.

	<i>a1</i>	<i>a2</i>	<i>a3</i>	<i>a4</i>	<i>a5</i>	<i>a6</i>
<i>Z1</i>	<i>a2</i> <i>a6</i>	<i>a1</i> <i>a4</i>	--	<i>a3</i>	--	<i>a5</i>
<i>Z2</i>	-	-	<i>a1</i> <i>a3</i> <i>a4</i>	--	<i>a2</i> <i>a5</i> <i>a6</i>	--

Соседями первого рода в обратной таблице переходов оказываются попарно состояния, находящиеся в одной ячейке таблицы:

$(a2, a6), (a1, a4), (a1, a3), (a3, a4), (a2, a5), (a5, a6).$

Соседи второго рода определяются так: если состояния являющиеся соседями первого рода оказались в одной строке, но в разных столбцах, то состояния, соответствующие этим столбцам, будут соседями второго рода.

Часто для кодирования состояний, близких к соседним, используются диаграммы Вейча–Карно. В этом случае первым кодируется состояние, которое в списке соседей встречается наибольшее число раз.

Диаграмма :

	0	0	1	1
	0	1	1	0
0	<i>a1</i>	<i>a4</i>	<i>a2</i>	<i>a6</i>
1	<i>a3</i>	-	-	<i>a5</i>

15.3. Кодирование состояний автомата для минимизации комбинационной схемы

Анализ канонического метода синтеза автомата показывает, что различные варианты кодирования состояний автомата приводят к различным вариантам формирования выражения для функции возбуждения элементов памяти и функции выхода. В результате оказывается, что сложность комбинационной схемы автомата существенно зависит от выбранного кодирования.

	<i>a1</i>	<i>a2</i>	<i>a3</i>
<i>z1</i>	<i>a2</i>	-	<i>a1</i>
<i>z2</i>	<i>a3</i>	<i>a1</i>	-
<i>z3</i>	<i>a2</i>	<i>a3</i>	<i>a3</i>

D-триггер:

	0	1
0	0	0
1	1	1

Закодируем состояния

$a1 \Rightarrow 00, a2 \Rightarrow 01, a3 \Rightarrow 11.$

$z1 \Rightarrow 00, z2 \Rightarrow 01, z3 \Rightarrow 10.$

Получим отмеченную таблицу переходов структурного автомата:

	01	01	11
00	01	-	00
01	11	00	-
10	01	11	11

$$\alpha_1 = 1 \vee 6 \vee 14$$

$$\alpha_2 = 0 \vee 1 \vee 2 \vee 6 \vee 14$$

Возьмём другое кодирование:

$a1 \Rightarrow 01, a2 \Rightarrow 10, a3 \Rightarrow 00.$

	01	10	00
00	10	-	01
01	00	01	-
10	10	00	00

$$\alpha_1 = 4 \vee 6$$

$$\alpha_2 = 0 \vee 9$$

При кодировании состояний автомата используются алгоритмы, позволяющие упростить функцию возбуждения элементов памяти, если при синтезе автомата в качестве элементарных автоматов памяти используются D-триггеры.

Алгоритм минимизации при использовании D-триггера.

1. Каждому состоянию ставится в соответствие целое число N_m , равное числу переходов в состояние a_m .

2. Числа N_m сортируются по убыванию.
3. Состояние с наибольшим N кодируется 00...00.
4. Следующие I состояний (I -число ЭП) кодируются 00..01, 00..10..0
5. Для кодирования оставшихся состояний используются коды, содержащие 2, затем 3 единицы и т.д., пока все состояния не будут закодированы.

В результате получаем кодирование, при котором чем больше переходов имеется в некоторое состояние a_m , тем меньше единиц содержится в его коде.

Большое число работ было посвящено получению такого кодирования, при котором уменьшается зависимость функции возбуждения ЭП от переменных обратной связи (τ_1, \dots, τ_n) .

В то же время многие авторы отмечают сложность этих методов кодирования, а также трудности одновременной минимизации функции возбуждения элементов памяти и функции выходов. Поэтому на практике чаще всего используется эвристический алгоритм кодирования состояний автомата, минимизирующий суммарное число переключений элементов памяти на всех переходах автомата. При таком подходе уменьшается сложность схем, реализующих дизъюнкцию на входе ЭП, а, следовательно, минимизирующих и комбинационную схему.

Основной алгоритм:

1. Строим матрицу, состоящую из различных пар номеров таких, что в автомате S есть переход $a_{\alpha k} \rightarrow a_{\beta k}$

$$M = \begin{pmatrix} \delta 1 & \beta 1 \\ \delta k & \beta k \end{pmatrix}$$

2. Переставим строки матрицы так, чтобы выполнялось условие:

$(\delta \beta) \cap ((\delta 1 \beta 1), \dots, (\delta - 1 \beta - 1)) \neq 0$. Такую матрицу можно построить только для связного графа автомата.

3. Закодируем состояние первой строки:

$$k_{\delta 1} = 00 \dots 00$$

$$k_{\beta 1} = 00 \dots 01$$

4. Вычёркиваем из матрицы M первую строку. Получим матрицу M' .
5. В начальной строке матрицы M' один элемент уже закодирован.
6. Выберем незакодированный элемент первой строки матрицы и обозначим его γ .

Построим матрицу M_γ , выбрав из M' все строки содержащие элемент γ .

7. Пусть множество $B_\gamma = \{\gamma_1, \dots, \gamma_f, \dots, \gamma_F\}$ - множество всех элементов матрицы M_γ , которые уже закодированы. Они известны.

Для каждого кода $k\gamma_f$ найдём $C^1\gamma_f$ - множество кодов, соседних с кодом $k\gamma_f$ и ещё не занятых для кодирования состояний автомата. Построим множество всех возможных кодов, соседних и еще незакодированных:

$$D^1\gamma = \bigcup_1^F C^1\gamma_f$$

если $D^1\gamma = 0$, то строим множество $D^2\gamma = \bigcup_1^F C^2\gamma_f$.

Если нет ни одного множества с незакодированными элементами, то количество ЭП выбрано неправильно.

8. Находим $Wgf = |k_{\gamma g} - k_{\gamma f}|^2$ - кодовое расстояние

9. Находим сумму всех кодовых расстояний

$$Wg = \sum_1^F Wgf$$

10. Выбираем код для состояния γ , у которого сумма кодовых расстояний Wg минимальна.

11. Из матрицы M' вычеркиваем строки, в которых оба элемента закодированы, получаем матрицу M'' . Если матрица M' - пустая, переходим к пункту 12, иначе к пункту 5.

12. Вычисляем $W = \sum t_{ms}$, сумму всех кодовых расстояний.

Оценкой качества кодирования рассмотренного алгоритма может служить число K

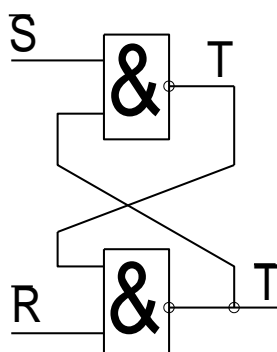
$$K = \frac{W}{p},$$

где p - число переходов данного автомата. Чем меньше K , тем ближе полученное кодирование к соседнему.

Эксперименты показали, что K при хорошем кодировании лежит в пределах $1,4 \leq K \leq 2,1$.

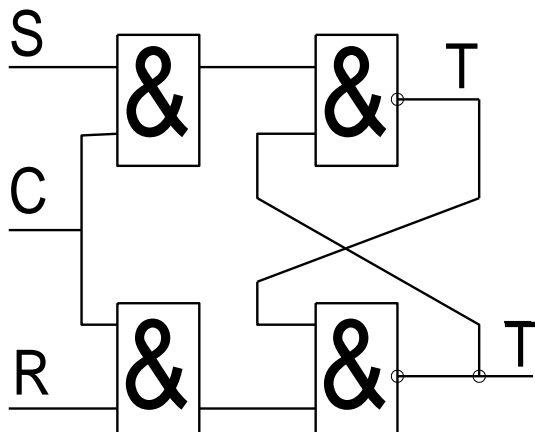
16. Элементарные автоматы памяти

1. RS-триггер – элемент памяти с двумя входами S – set, R – reset.



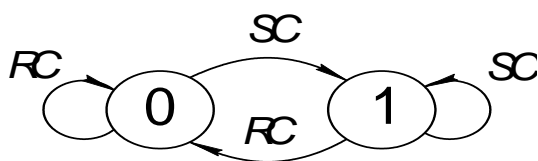
Основной недостаток - инверсное значение входных и выходных сигналов, нет защиты от аппаратных гонок.

Для того чтобы использовать данный триггер и устранить гонки применяют систему двухтактную, т.е. добавляют ещё один каскад.

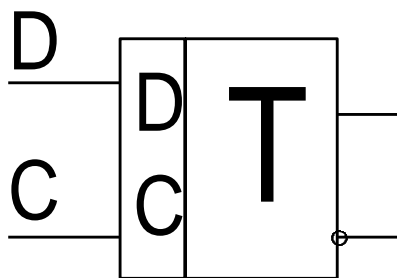


R	S	T_n	T_{n+1}
b1	0	0	0
0	1	0	1
1	0	1	0
0	b2	1	1
1	1	b3	?

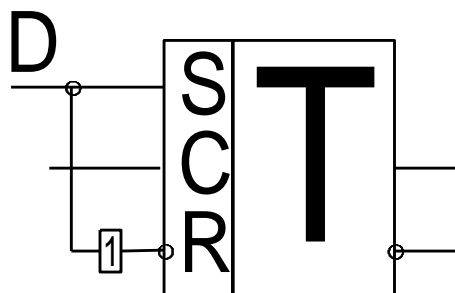
Граф такого автомата:



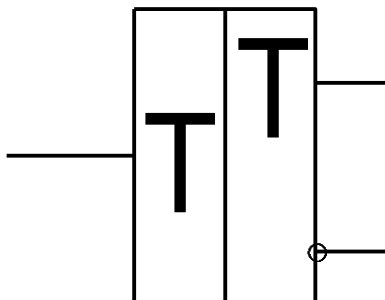
2. D-триггер.



D	T_n	T_{n+1}
0	0	0
0	1	0
1	0	1
1	1	1



3. T-триггер



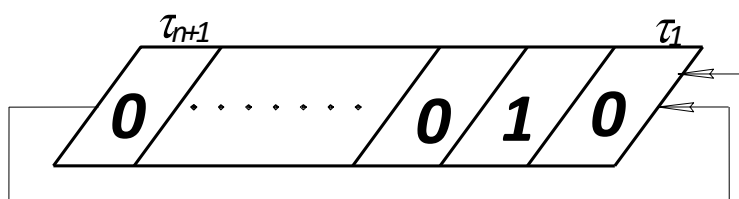
T	T_n	T_{n+1}
0	0	0
0	1	1
1	0	1
1	1	0

Методы унитарного кодирования

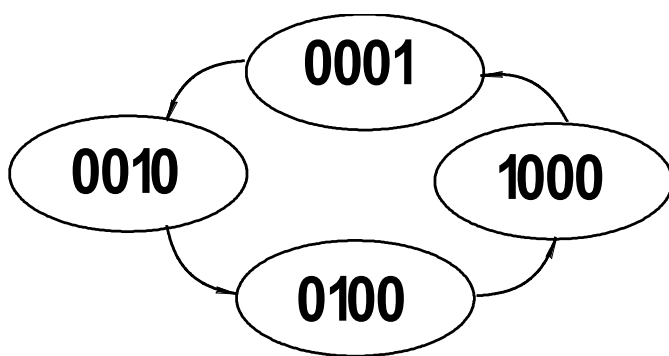
При унитарном кодировании в любом коде - только 1 единица.

При этом коды 00..001 и 00..010 являются соседними

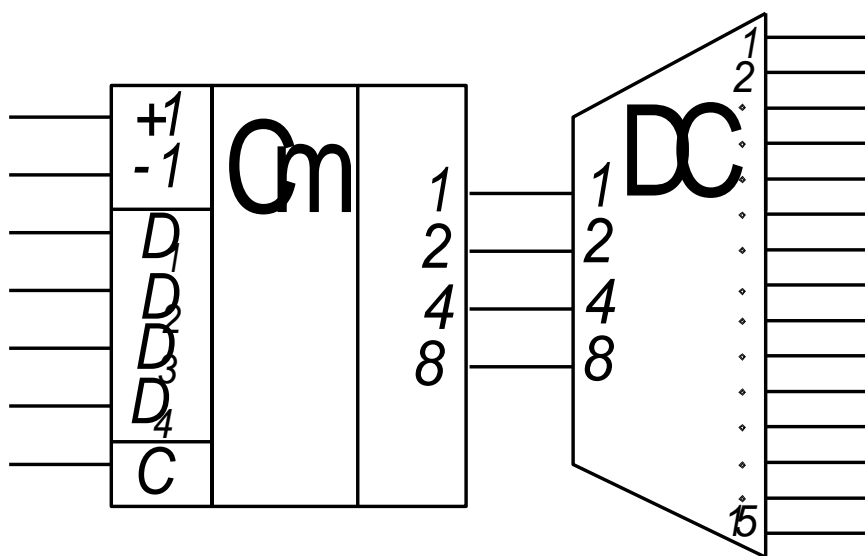
Сдвиговый регистр:



Граф сдвигового регистра:



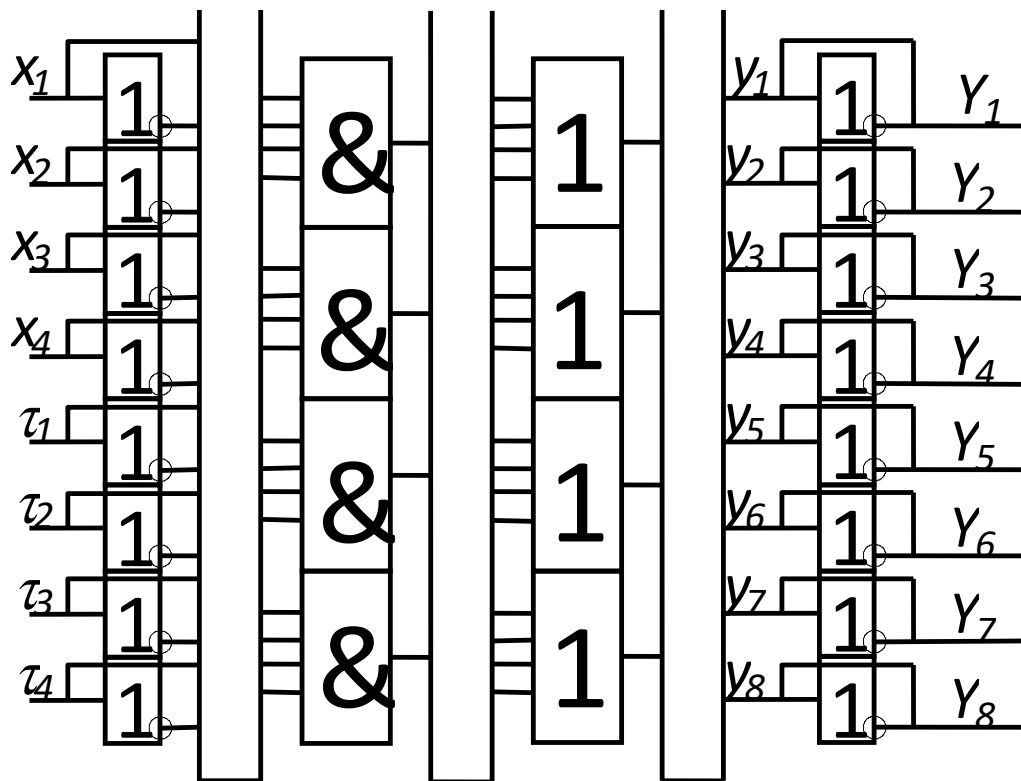
Использование схем на основе двоичного счетчика и дешифратора



17. Синтез автоматов на ПЛМ и ПЗУ

ПЛМ – программируемая логическая матрица.

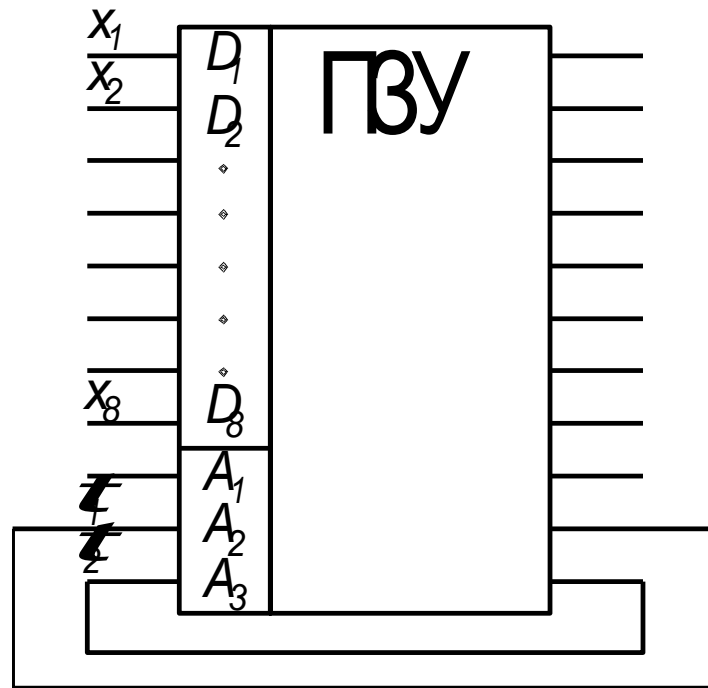
Рассмотрим структуру микросхемы ПЛМ.



Данная микросхема легко программируется на основе канонических уравнений комбинационной схемы структурного автомата.

ПЗУ – постоянное запоминающее устройство.

Микросхема ПЗУ обычно содержит следующие входы и выходы.



Программа ("прошивка") для реализации автомата на основе ПЗУ требуемого алгоритма имеет вид таблицы соответствия входных и выходных сигналов автомата. Программа записывается электрическим способом и обычно имеется возможность перезаписи программ в микросхему – ЭППЗУ.

Рекомендуемая литература

Основная литература

1. Карпов, Юрий Глебович. Теория автоматов: Учеб. для вузов / Карпов, Юрий Глебович. - М.: Питер, 2003, 2002. - 208с.: ил. - Библиогр.: с. 204. (количество в библиотеке - 24 экз).

2. Молчанов, Алексей Юрьевич. Системное программное обеспечение: Учеб. / Молчанов, Алексей Юрьевич. - СПб.: Питер, 2003. - 396с.: ил. - Библиогр.: с. 386. (количество в библиотеке - 20 экз).

3. Мельцов, Василий Юрьевич. Синтез микропрограммных управляющих автоматов. Учебное пособие / Мельцов, Василий Юрьевич, Фадеева, Татьяна Рафаиловна. – Киров: ВятГУ, 2000,-56с. (количество в библиотеке - 55 экз).

Дополнительная литература

4. Акулов, Олег Анатольевич. Информатика: базовый курс: Учеб. / Акулов, Олег Анатольевич, Медведев, Николай Викторович. - М.: Омега-Л, 2004. - 552с. - (Учебник для технических вузов). - Библиогр.: с. 545-546. (количество в библиотеке - 3 экз).

5. Калиш, Георгий Германович. Основы вычислительной техники: Учеб. пособие / Калиш, Георгий Германович. - М.: Высш. шк., 2000. - 271с. (количество в библиотеке - 2 экз).

6. Лысиков, Борис Григорьевич. Арифметические и логические основы цифровых автоматов: Учеб. / Лысиков, Борис Григорьевич. - 2-е изд., перераб. и доп. - Минск: Вышейш. шк., 1980. - 335с. (количество в библиотеке - 24 экз).

7. Савельев, Александр Яковлевич. Прикладная теория цифровых автоматов: Учеб. / Савельев, Александр Яковлевич. - М.: Высш. шк., 1987. - 272с. (количество в библиотеке - 24 экз).

8. Лазарев, Владимир Георгиевич. Синтез управляющих автоматов / Лазарев, Владимир Георгиевич, Пийль, Елена Ивановна. - 3-е изд., перараб. и доп. - М.: Энергия, 1989. - 328с. (количество в библиотеке - 2 экз).

9. Сапожников, Валерий Владимирович. Методы синтеза надежных автоматов / Сапожников, Валерий Владимирович, Сапожников, Владимир Владимирович. - Л.: Энергия, 1980. - 93с. (количество в библиотеке - 11 экз).

10. Соловьев, Валерий Васильевич. Проектирование цифровых систем на основе программируемых логических интегральных схем / Соловьев, Валерий Васильевич. - М.: Горячая линия-Телеком, 2001. - 63бс. (количество в библиотеке - 3 экз).

11. Глушков, Виктор Михайлович. Логическое проектирование дискретных устройств / Глушков, Виктор Михайлович, Капитонова, Юлия Владимировна, Мищенко, Андрей Тихонович; АН УССР, Ин-т кибернетики. - Киев: Наук. думка, 1987. - 263с.: ил. (количество в библиотеке - 4 экз).

13. Рейуорд-Смит, В. Дж. Теория формальных языков: Вводный курс / Рейуорд-Смит, В. Дж. - М.: Радио и связь, 1988. - 127с. (количество в библиотеке - 1 экз).