

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«ОРЕНБУРГСКИЙ ГОСУДАРСТВЕННЫЙ АГРАРНЫЙ УНИВЕРСИТЕТ»**

**МЕТОДИЧЕСКИЕ МАТЕРИАЛЫ ДЛЯ ОБУЧАЮЩИХСЯ
ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ**

Б1.В.04 Теория автоматов

Спеальность 10.05.03 Информационная безопасность автоматизированных систем

**Специализация Информационная безопасность автоматизированных систем критически
важных объектов**

Форма обучения очная

СОДЕРЖАНИЕ

1. Конспект лекций	3
1.1. Лекция №1-2 Определение абстрактного автомата	3
1.2. Лекция № 3-4 Языки операторных схем алгоритма	8
1.3. Лекция № 5 Модель структурного автомата. Структурный базис. Типы триггеров. Цели и задачи структурного анализа АСУ.....	11
1.4. Лекция № 6 Асинхронные автоматы.....	15
1.5. Лекция № 7 Классификация формальных грамматик по Хомскому. Вывод. Виды топологических структур.....	19
1.6. Лекция № 8 Недетерминированный конечный автомат.....	21
2. Методические указания по проведению лабораторных работ.....	27
2.1. Лабораторная работа № 1-2 Модели Мура и Мили.....	27
2.2. Лабораторная работа № 3-5 Синтез абстрактного автомата по операторной схеме алгоритма.....	27
2.3. Лабораторная работа № 6-8 Каноничный метод структурного синтеза конечного автомата.	27
2.4. Лабораторная работа № 9-11 Асинхронные автоматы.	27
2.5. Лабораторная работа № 12-13 Язык, порождаемый грамматикой.	27
2.6. Лабораторная работа № 14-16 Недетерминированный конечный автомат.	28

1. КОНСПЕКТ ЛЕКЦИЙ

1.1. Лекция № 1-2 Тема: «Определение абстрактного автомата».

1.1.1. Вопросы лекции:

1. Понятие автомата, абстрактного автомата, конечного автомата.
2. Представление алгоритмов конечными автоматами.
3. Алгоритмы минимизации автоматов.

1.1.2. Краткое содержание вопросов:

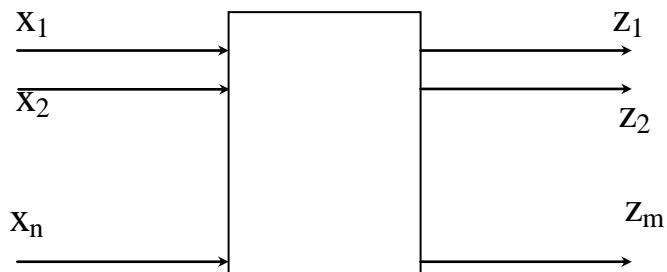
1. Понятие автомата, абстрактного автомата, конечного автомата.

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, представляемая основной моделью, управляет 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 ин-

тервала 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 автомата 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 фактами 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.

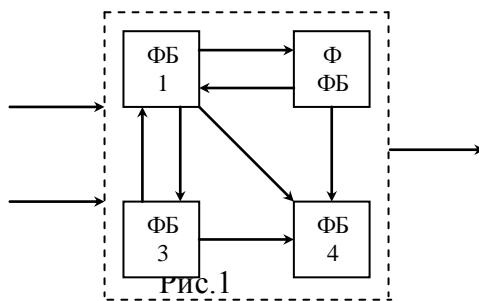
2. Представление алгоритмов конечными автоматами.

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 системе-

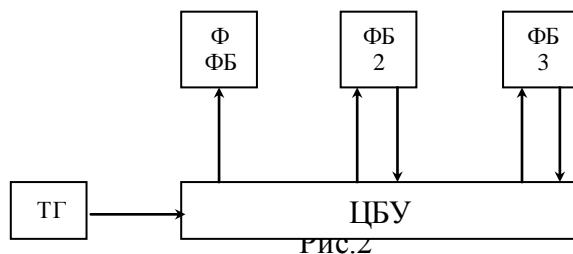
мы может рассматриваться как некое устройство, реализующее алгоритм ее функционирования. При этом отдельные функциональные блоки (ФБ) выполняют те или иные этапы алгоритмов. Для того чтобы управляющий автомат реализовал весь алгоритм, между функциональными блоками должны быть функциональные связи, определяющие 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).



2. Управляющий автомат с **концентрируемыми** функциональными связями (см. рис. 2). Второй принцип создания автомата 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 (запrogramмированной в ЦБУ программой) и зависит от поступающей осведомительной информации.



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

В программируемом 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.

3. Алгоритмы минимизации 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.Клинни, 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 од-

нотактного 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.

1.2. Лекция № 3-4 Языки операторных схем алгоритма

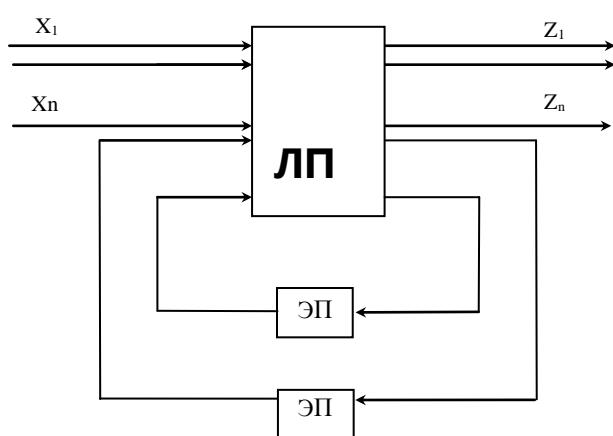
1.2.1. Вопросы лекции:

1. Синтез логических сетей в заданном базисе.
2. Алгоритмы кодирования состояний.
3. Методы проектирования алгоритмов.

1.2.2. Краткое содержание вопросов:

1. Синтез логических сетей в заданном базисе.

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.

2. Алгоритмы кодирования состояний.

Под кодированием 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 $A=a_1 \dots a_i$ и $B=b_1 \dots b_i$ 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 приводи (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 заранее известный 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 приближенных, 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.
3. Методы проектирования алгоритмов.

Для проектирования 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.

1.3. Лекция № 5 (2 часа)

Тема: «Модель структурного автомата. Структурный базис. Типы триггеров. Цели и задачи структурного анализа АСУ».

1.3.1. Вопросы лекции:

1. Минимизация конечных автоматов.

2. Противогоночное кодирование.
3. Типы триггеров.

1.3.2. Краткое содержание вопросов:

1. Минимизация конечных автоматов.

Под **минимизацией** конечных автоматов 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 $A=a_1\dots a_i$ и $B=b_1\dots b_i$ 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 приводи (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 нереализуемым.

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.

2. Противогоночное кодирование.

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 связанными.

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 - $(\alpha, \beta), (\gamma, \delta), i=1, I$.

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 неопределенным разрядом и перейти к пункту 2.
8. Пары переходов $(a_m, a_s), (a_k, a_l)$ развязаны.

3. Типы триггеров.

D-триггер:

	0	1
0	0	0
1	1	1

Закодируем состояния

$a_1 = >00, a_2 = >01, a_3 = >11.$

$z_1 = >00, z_2 = >01, z_3 = >10.$

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:

$a_1 = >01, a_2 = >10, a_3 = >00.$

	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 кодируются 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.

1.4. Лекция № 6 (2 часа)

Тема: «Асинхронные автоматы».

1.4.1. Вопросы лекции:

1. Построение асинхронных алгоритмов.

2. Синтез легкотестируемого устройства.
3. Моделирование работы асинхронного автомата.

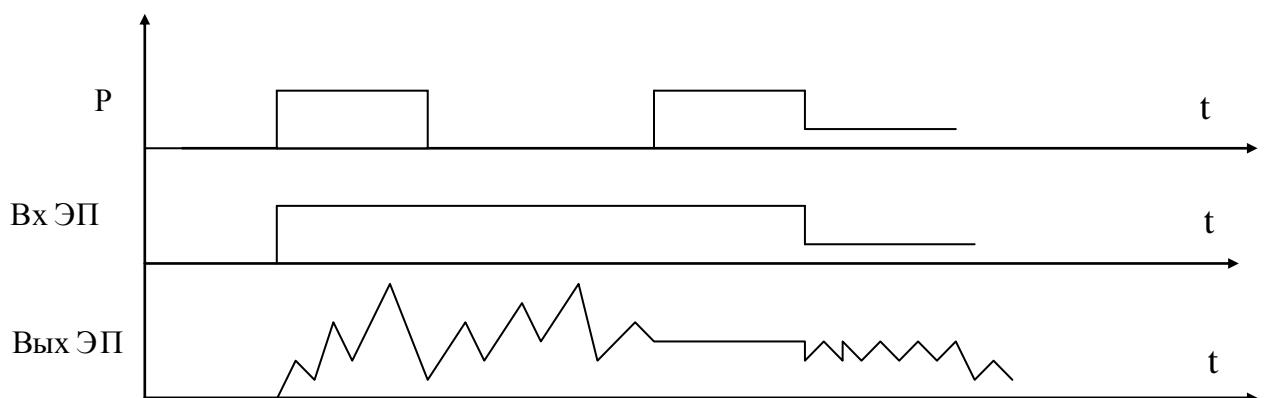
1.4.2. Краткое содержание вопросов:

1. Построение асинхронных алгоритмов.

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.

2. Синтез легкотестируемого устройства.

При синтезе легкотестируемого устройства мы не интересуемся 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.Клинни, 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 однотактного 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.

3. Моделирование работы 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.

	ρ^1	ρ^2	ρ^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 зацикливание (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.

	ρ^1	ρ^2
X1	$\lambda 1$	$\lambda 1$
X2	$\lambda 2$	-
X3	$\lambda 2$	$\lambda 3$
X4	-	$\lambda 1$

1.5. Лекция № 7 (2 часа)

Тема: «Классификация формальных грамматик по Хомскому. Вывод. Виды топологических структур».

1.5.1. Вопросы лекции:

1. Топологические структуры.
2. Анализ формальных грамматик.

1.5.2. Краткое содержание вопросов:

1. Топологические структуры.

Под топологической структурой 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.

2. Анализ формальных грамматик.

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.

1.6. Лекция № 8 (2 часа)

Тема: «Недетерминированный конечный автомат».

1.6.1. Вопросы лекции:

1. Самопроверяемые детекторы неупорядоченных кодов.
2. Стратегии синтаксического анализа.

1.6.2. Краткое содержание вопросов:

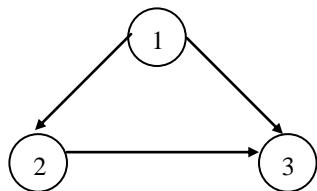
1. Самопроверяемые детекторы неупорядоченных кодов.

Самопроверяемым детектором 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.

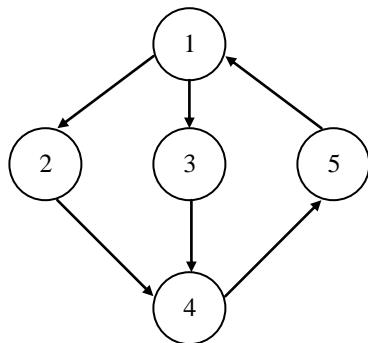
Два состояния, в которые переходят под действием одного и того же входного сигнала состояния, являющиеся соседями первого рода, называются соседями второго рода. Соседними кодами называются наборы значений, имеющие противоположные значения только в одном разряде, т.е. коды с кодовым расстоянием 1. При соседнем кодировании любые два состояния, связанные дугой на графе автомата кодируются наборами, отличающимися 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.

	$a1$	$a2$	$a3$	$a4$	$a5$	$a6$
$Z1$	$a2$	$a1$	--	$a3$	--	$a5$
	$a6$	$a4$				
$Z2$	-	-	$a1$	--	$a2$	--
			$a3$		$a5$	
			$a4$		$a6$	

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	a1	a4	a2	a6
1	a3	-	-	a5

2. Стратегии синтаксического анализа.

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.

	a1	a2	a3
z1	a2	-	a1
z2	a3	a1	-
z3	a2	a3	a3

D-триггер:

	0	1
0	0	0
1	1	1

Закодируем состояния

$a_1 = >00, a_2 = >01, a_3 = >11.$

$z_1 = >00, z_2 = >01, z_3 = >10.$

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:

$a_1 = >01, a_2 = >10, a_3 = >00.$

	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 кодируются 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, минимизирующую комбинационную схему.

Основной алгоритм:

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

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

2. Переставим строки матрицы так, чтобы выполнялось условие:

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

3. Закодируем состояние первой строки:

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

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

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 $Wgf = |k_g - k_f|^2$ - 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 $1,4 \leq K \leq 2,1$

2. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО ПРОВЕДЕНИЮ ЛАБОРАТОРНЫХ РАБОТ

2.1. Лабораторная работа №1-2 (4 часа).

Тема: «Модели Мура и Мили».

1. Построить таблицу истинности для функции $f = z \wedge y + x$
2. Построить таблицу истинности для функции $f = z + y + x$
3. Построить таблицу истинности для функции $f = z \wedge y \wedge x$

2.2. Лабораторная работа № 3-5 (6 часов).

Тема: «Синтез абстрактного автомата по операторной схеме алгоритма».

1. Построить синхронный D-триггер на элементах И-НЕ
2. Построить синхронный RS-триггер на элементах И-НЕ
3. Построить синхронный JK-триггер на элементах ИЛИ-НЕ

2.3. Лабораторная работа № 6-8 (6 часов).

Тема: «Канонический метод структурного синтеза конечного автомата».

1. Построить синхронный T-триггер на элементах И-ИЛИ-НЕ
2. Построить асинхронный JK-триггер на элементах ИЛИ-НЕ
3. Построить асинхронный T-триггер на элементах И-НЕ

2.4. Лабораторная работа № 9-11

Тема: «Асинхронные автоматы».

1. Построить счётчик от 0 до 9 на D-триггере
2. Построить счётчик от 2 до 12 на T-триггере
3. Построить счётчик от 13 до 5 на RS-триггере

2.5. Лабораторная работа № 12-16

Тема: «Язык, порождаемый грамматикой».

1. Построить мультиплексор 6 1 на элементах И-НЕ
2. Построить дешифратор на элементах ИЛИ-НЕ
3. Построить схему преобразования прямого кода в код Грэя на элементах И-НЕ

2.6. Лабораторная работа № 14-16 (интерактивная форма – 6ч)

Тема: «Недетерминированный конечный автомат».

1. Построить шифратор на элементах ИЛИ-НЕ
2. Построить схему преобразования кода Джонсона в обратный код на элементах ИЛИ-НЕ
3. Построить схему преобразования кода Грея в код Джонсона на элементах И-НЕ